Informationskopie !
Wird bei Änderung nicht
ausgetauscht
for Information only !

Document Name: **Engine Management System**

Project: **MSD80 6 Cyl**

Engineering Change Number:

Version: 48.0

Release Date: 2008-07-01

Chapter Contents of All Chapters	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 1 of 9643

Version history

Document version	Previous version	Change description (including number)
48.0		ECM

8 Driver request pedals determination 1641

9 Driver request 1691

10 Driveability 1706

12 ECU Processor Monitoring 2071

59 Suspension system 8989

60 System Configuration 9011

67 Vehicle speed control 9447

Index of All Chapters

1 Air temperature

Table of Contents

Index

1.1 AIRT – Requirements to infrastructure interface

Input data:

LV_VAR_TCHA			

Export actions:

ACTION_INFR_GetVpTia (OUT <Vp_tia_sens**>)**
This action provides the digitized voltage value of the available intake air temperature sensors in a single value.
ACTION_INFR_GetVpTiaTcha (OUT <Vp_tia_tcha_sens**>)**
This action provides the digitized voltage value of the available temperatur up turbocharger sensor in a single value.

Description for actions:

ACTION_INFR_GetVpTia (Vp_tia_sens**)**					
This action provides the digitized voltage value of the available intake air temperature sensors in a single value delivered from the infrastructure. The AD conversion is performed autonomously by the infrastructure. When the action is called the gathered information will be provided to the application software level. Port-Channel: if LV_VAR_TCHA = 0 (N53) PIN 5_28 if LV_VAR_TCHA = 1 (N54) PIN 5_17					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Vp_tia_sens	OUT	0...7FFFH	0...4.999847412	1.5258E-04	V
Intake air temperature sensor voltage value					

ACTION_INFR_GetVpTiaTcha (Vp_tia_tcha_sens**)**					
This action provides the digitized voltage value of the available temperatur sensor up turbocharger in a single value delivered from the infrastructure. The AD conversion is performed autonomously by the infrastructure. When the action is called the gathered information will be provided to the application software level. Port-Channel: if LV_VAR_TCHA = 0 (N53) PIN 5_17 if LV_VAR_TCHA = 1 (N54) PIN 5_28					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Vp_tia_tcha_sens	OUT	0...7FFFH	0...4.999847412	1.5258E-04	V
Voltage for temperatur sensor up turbocharger					

FUNCTION DESCRIPTION:

General information:

The following action is used for the acquisition of the digitized voltage value of the adressed intake air temperature sensor and the "up turbocharger" air temperature sensor in two single values. The action delivers the intake air temperature sensor and the "up turbocharger" air temperature sensor voltage values to the application software level from the standard AD converter queue.

- The AD conversion is performed autonomously by the infrastructure, the returned value is not older than 1000 ms (in case of TIA sensor).

- The voltage values are gathered in the infrastructure until the application reads out the information by calling the action, old values are replaced by new values.

Requirements for ACTION_INFR_GetVpTia:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Vp_tia_sens			10 Bit		No comment

Diagnosis: no electrical diagnosis done here

Coincidence requirements: When calling this action, all returned voltages must be within the last 100 ms. Port-Channel:

```
If          LV_VAR_TCHA = 0
      then  Port-Channel: PIN_5_28 (N53)
      else  Port-Channel: PIN_5_17 (N54)
endif
```

Requirements for ACTION_INFR_GetVpTiaTcha:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Vp_tia_tach_sens			10 Bit		No comment

Diagnosis: no electrical diagnosis done here

Coincidence requirements: When calling this action, all returned voltages must be within the last 100 ms. Port-Channel:

```
If          LV_VAR_TCHA = 0
      then  Port-Channel: PIN_5_17 (N53)
      else  Port-Channel: PIN_5_28 (N54)
endif
```

1.2 AGGR AIRT adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TIA_IM	O/V	0...1H	0...1	1	-
Boolean for error currently present which might affect temperature TIA_IM (after debounce)					
LV_ERR_TIA_THR	O/V	0...1H	0...1	1	-
Boolean for error currently present which might affect temperature TIA_THR (after debounce)					
TIA_IM	O/V	0...FEH	-48...142.5	0.75	°C
Air temperature in the intake manifold					
TIA_THR	O/V	0...FEH	-48...142.5	0.75	°C
Air temperature at the throttle body					
TIA_THR_ST	O/V	0...FEH	-48...142.5	0.75	°C
Air temperature at the throttle body at start					

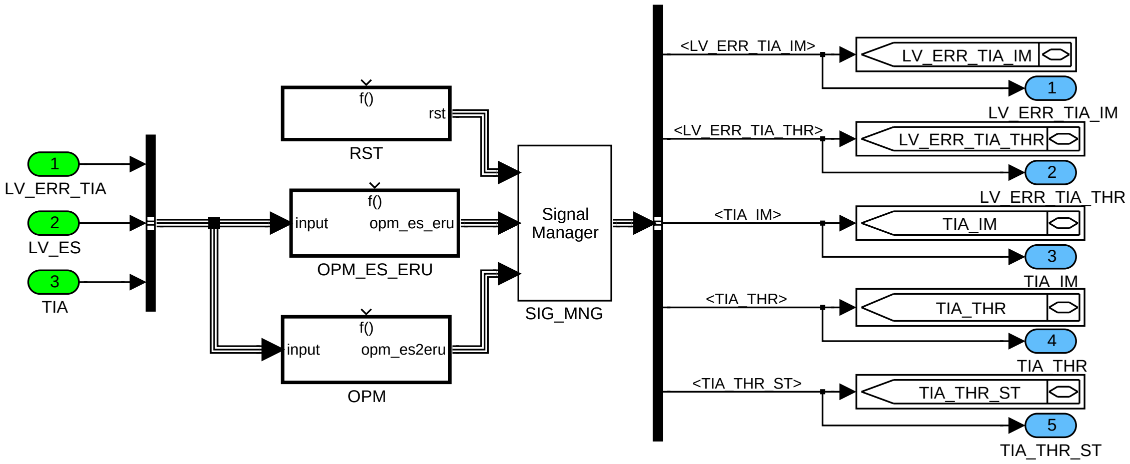
Input data:

LV_ERR_TIA	LV_ES	TIA	

1.2.1 Function Description

Adaptation of AIRT variables to MSV70 environment.

Function Description



SDA_SRS / SDA 4.0 29−Jul−2005

Figure 1 AIRT_M300B

1.2.1.1 Calculation at reset

Initialization at reset

TIA_THR_ST is initialised at engine start.

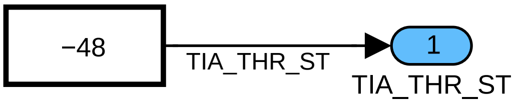


Figure 2 AIRT_M300B/ RST/ INI

1.2.1.2 Calculations at Engine stop to engine run

Initialization

TIA is assigned to output variable TIA_THR_ST

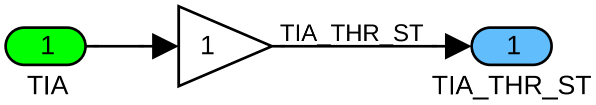


Figure 3 AIRT_M300B/ OPM_ES_ERU/ INI

1.2.1.3 Formula section

Calculation of outputs

Output variables TIA_IM, TIA_THR, LV_ERR_TIA_IM and LV_ERR_TIA_THR are calculated here. TIA is assigned to TIA_IM and TIA_THR , LV_ERR_TIA is assigned to LV_ERR_TIA_THR and LV_ERR_TIA_IM.

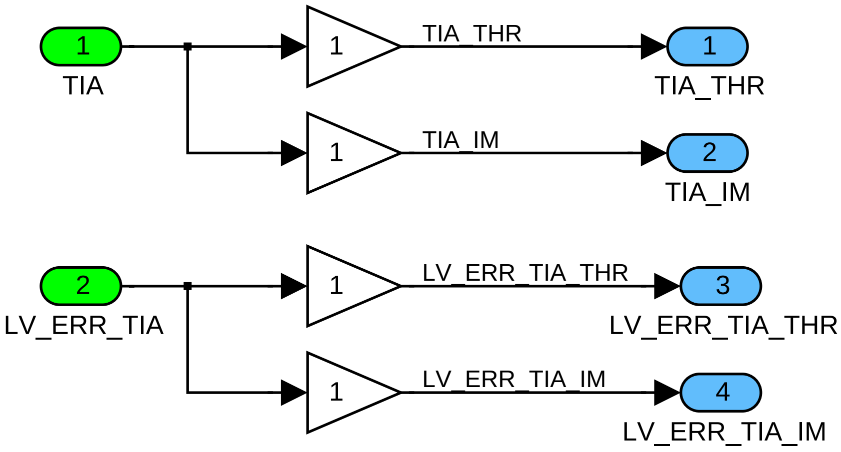


Figure 4 AIRT_M300B/ OPM/ SUB_CLC

1.3 Ambient Air Temperature TAM

Overview

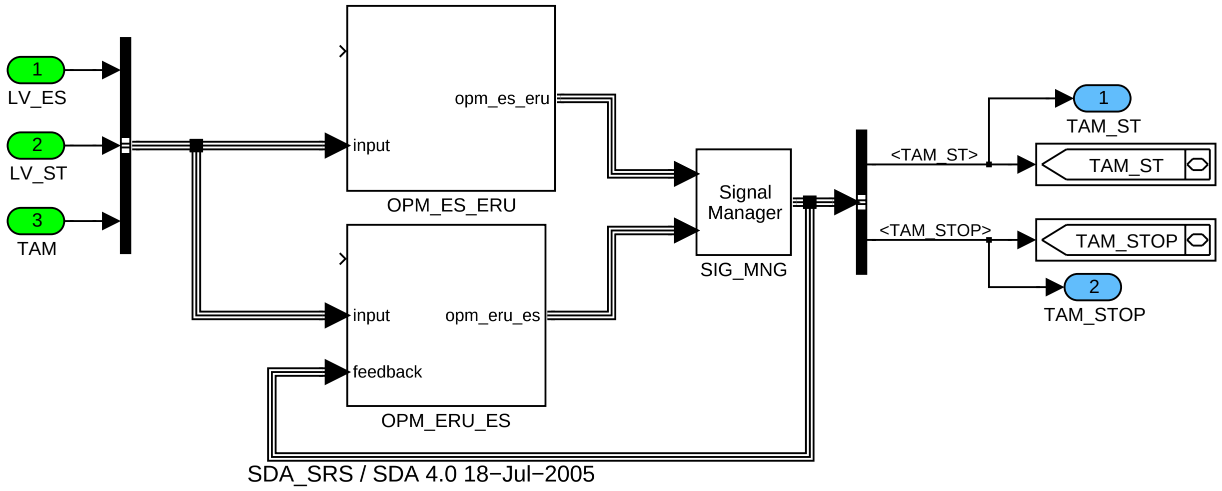


Figure 5 AIRT_M4024

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TAM_ST	O/V	0...FEH	-48...142.5	0.75	°C
Ambient air temperature at Start					
TAM_STOP	O/V/S	0...FEH	-48...142.5	0.75	°C
Ambient air temperature at engine stop					

Input data:

LV_ES	LV_ST	TAM	

1.3.1 Calculation of ambient temperature at start(ES to ERU)

General information:

TAM is received via CAN (see chapter CAN-messages).

TAM_ST is determined at each transition from ES to ST.

Function Description

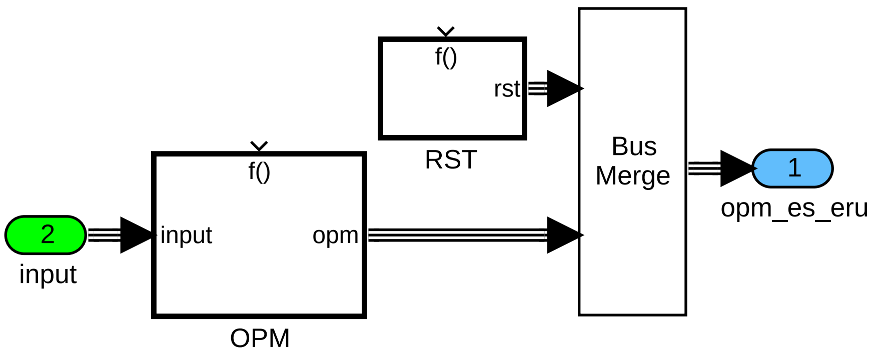


Figure 6 AIRT_M4024/ OPM_ES_ERU

1.3.1.1 Initialisation at reset

Initialisation of TAM_ST to 0 at reset

Figure 7 AIRT_M4024/ OPM_ES_ERU/ RST

1.3.1.2 Formula section

Aissgnment of TAM_ST

TAM is assigned to TAM_ST at start.

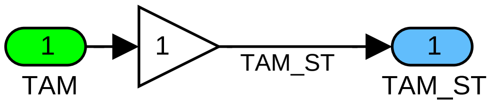


Figure 8 AIRT_M4024/ OPM_ES_ERU/ OPM/ CLC

1.3.2 Calculation of ambient temperature at Engine stop(ERU to ES)

General information:

TAM is received via CAN (see chapter CAN-messages).

TAM_STOP is determined at each transition from LV_ES 0 --> 1.

Function Description

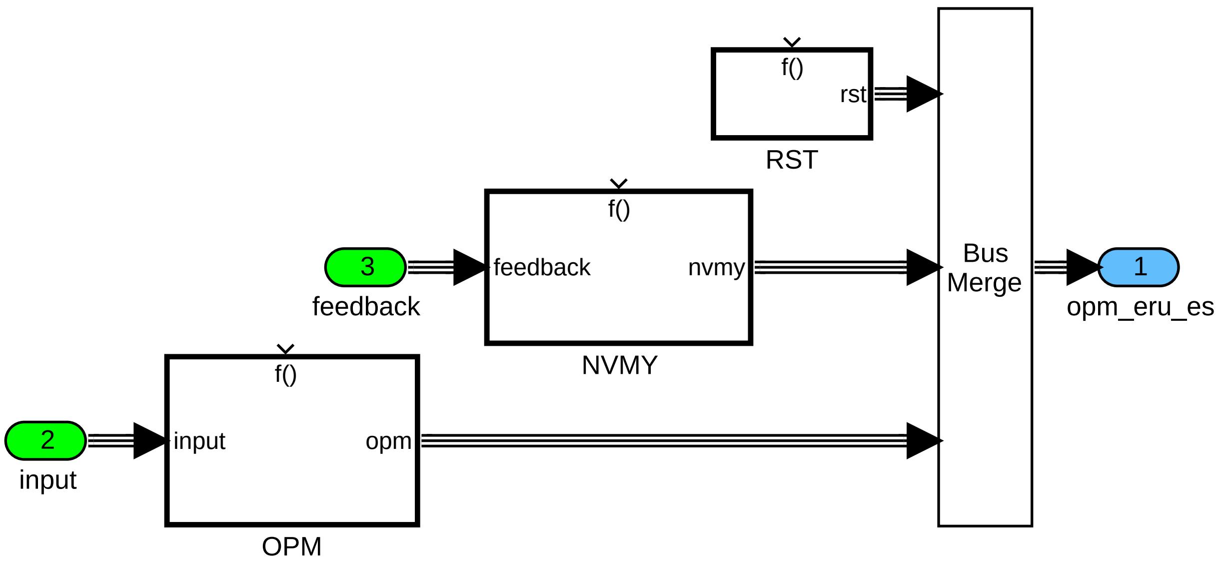


Figure 9 AIRT_M4024/ OPM_ERU_ES

1.3.2.1 At reset

Initialisation

Initialisation of TAM_STOP to 0 at reset.

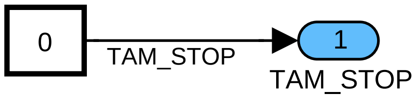


Figure 10 AIRT_M4024/ OPM_ERU_ES/ RST/ INI

1.3.2.2 Initialisations at non volatile memory system event(NVMY).

At system event NVMY the initialisation, restore and store action to and from the non volatile memory takes place.

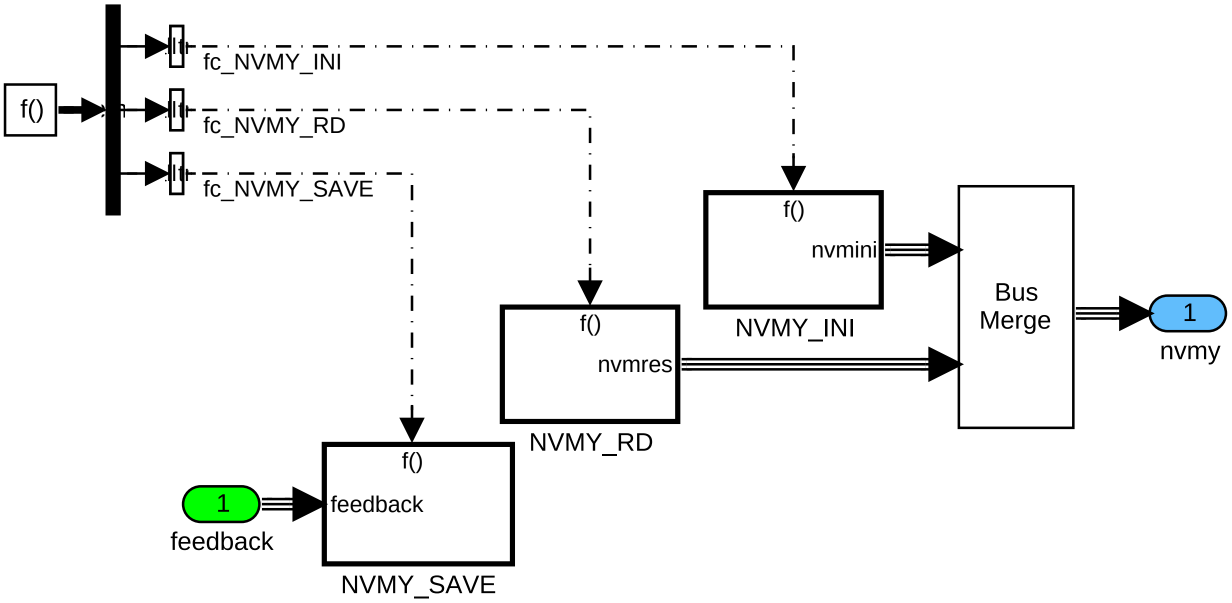


Figure 11 AIRT_M4024/ OPM_ERU_ES/ NVMY

1.3.2.3 Formula section

Assignment to TAM_STOP

TAM is assigned to TAM_STOP

Figure 12 AIRT_M4024/ OPM_ERU_ES/ OPM/ CLC

1.4 Intake air temperature TIA

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TIA	O/V	0...FEH	-48...142.5	0.75	°C
Intake air temperature					
TIA_MES	O/V	0...FEH	-48...142.5	0.75	°C
Intake air temperature raw value					
TIA_ST	O/V	0...FEH	-48...142.5	0.75	°C
Intake air temperature at ST					
TIA_SUB	O/V	0...FEH	-48...142.5	0.75	°C
Substitute value intake air temperature					
TIA_TCHA	O/V	0...FEH	-48...142.5	0.75	°C
Air temperature up turbo charger					
VP_TIA	O/V	0...7FFFH	0...4.99984741	1.52588E-4	V
Voltage value of the intake air temperature sensor					
VP_TIA_TCHA	O/V	0...7FFFH	0...4.99984741	1.52588E-4	V
Voltage for temperatur sensor up turbocharger					

Input data:

C_VP_TIA_MAX_DIAG	C_VP_TIA_MIN_DIAG	LV_ERR_TCO	LV_ERR_TIA
LV_ERR_TIA_TCHA_UP	LV_ST	NC_CYL_NR	TCO
LV_VAR_TCHA			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_DIAG	1	0...FEH	-48...142.5	0.75	°C
TCO threshold for switching TIA substitute					
C_TIA_GRD_MAX_DIAG	1	0...FEH	0...190.5	0.75	°C/500ms
Maximum permissible gradient for TIA					
C_TIA_SUB_BOL	1	0...FEH	-48...142.5	0.75	°C
Lower TIA substitute value					
C_TIA_SUB_TOL	1	0...FEH	-48...142.5	0.75	°C
Upper TIA substitute value					
IP_TIA_TCHA__VP_TIA_TCHA	16	0...FEH	-48...142.5	0.75	°C
LDP_VP_TIA_TCHA	16	0...7FFFH	0...4.99984741	1.52588E-4	V
NTC characteristic up turbo charger					
IP_TIA__VP_TIA	16	0...FEH	-48...142.5	0.75	°C
LDP_VP_TIA_TIA	16	0...7FFFH	0...4.99984741	1.52588E-4	V
NTC characteristic					

Import actions:

ACTION_INFR_GetVpTia(OUT <VP_TIA>)

This action provides the digitized voltage value of the available intake air temperature sensors in a single value.

ACTION_INFR_GetVpTiaTcha(OUT <VP_TIA_TCHA>)

This action provides the digitized voltage values of the available temperature of turbocharger sensor in a single value.

1.4.1 Function description

Function Description

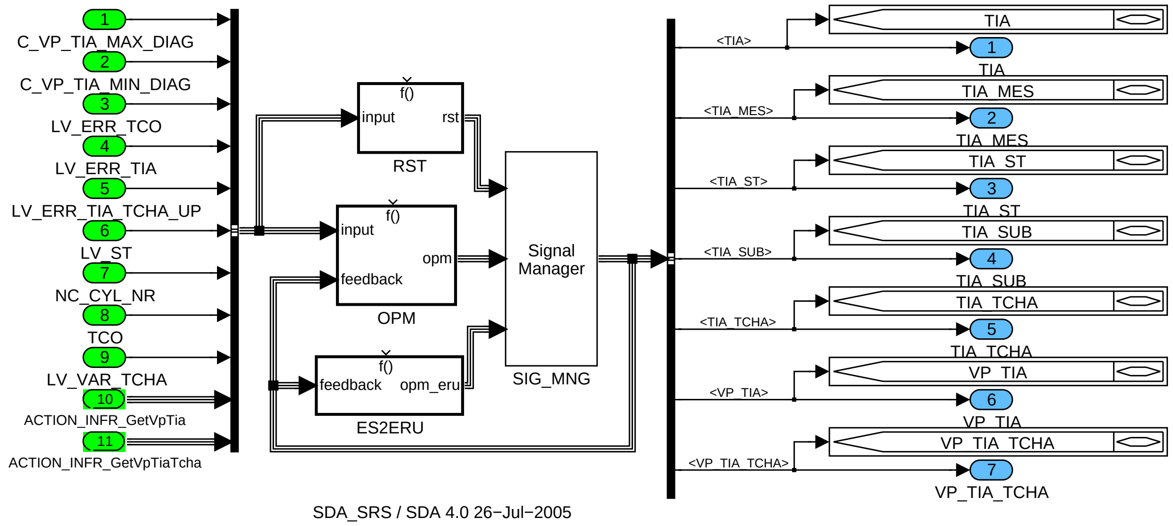


Figure 13 AIRT_M4007

1.4.1.1 Temperature determination

General information:

The air temperatures are measured through an A/D converter.

The conversion into physical values is made through the non-linear characteristic IP_TIA__VP_TIA and IP_TIA__V_TANS_TCHA. In case of a present fault of the TIA-sensor a substitute value will be used.

Calculation at reset

Calculation of TIA, TIA_MES, VP_TIA and VP_TIA_TCHA at reset

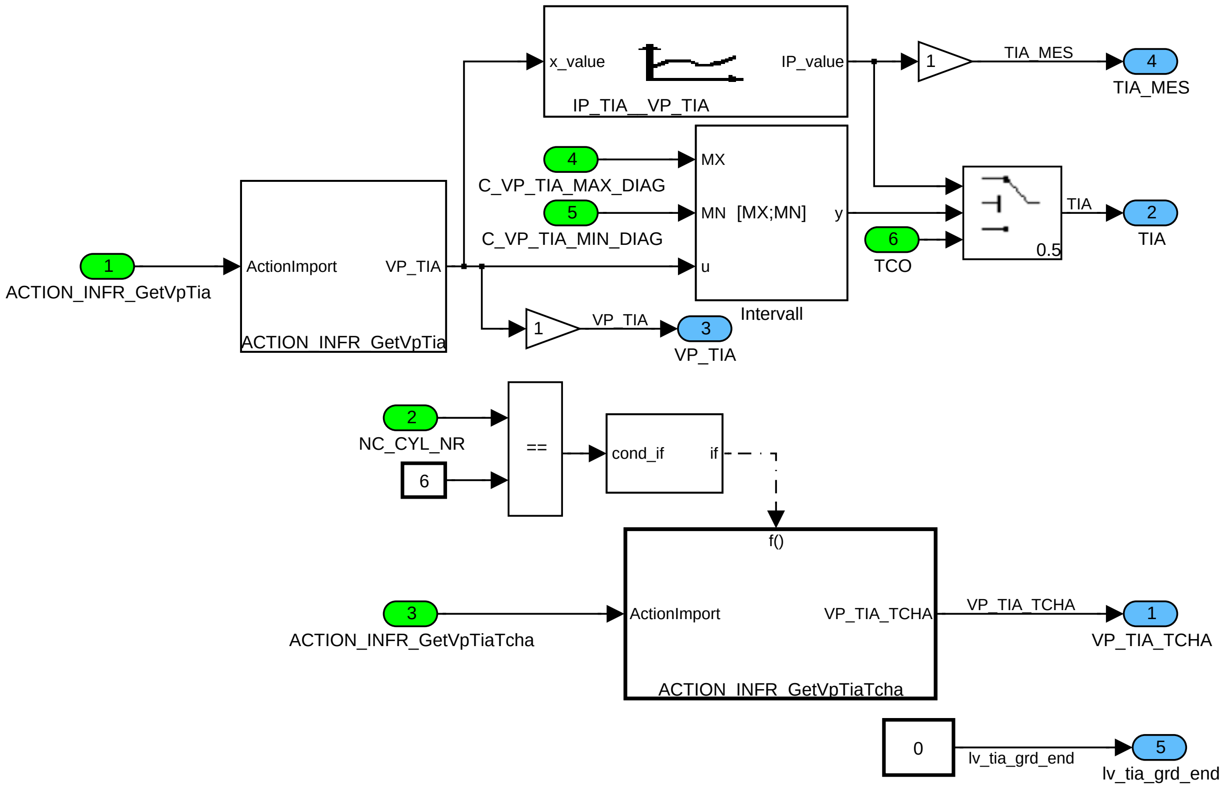


Figure 14 AIRT_M4007/ RST/ CLC

1.4.1.2 Formula section

Calculation at 100ms recurrence

Calculation of TIA, TIA_MES, TIA_SUB, VP_TIA and VP_TIA_TCHA

Action infrastructure outputs

The outputs of the air temperature sensor (VP_TIA and VP_TIA_TCHA) are calculated here using action infrastructure.

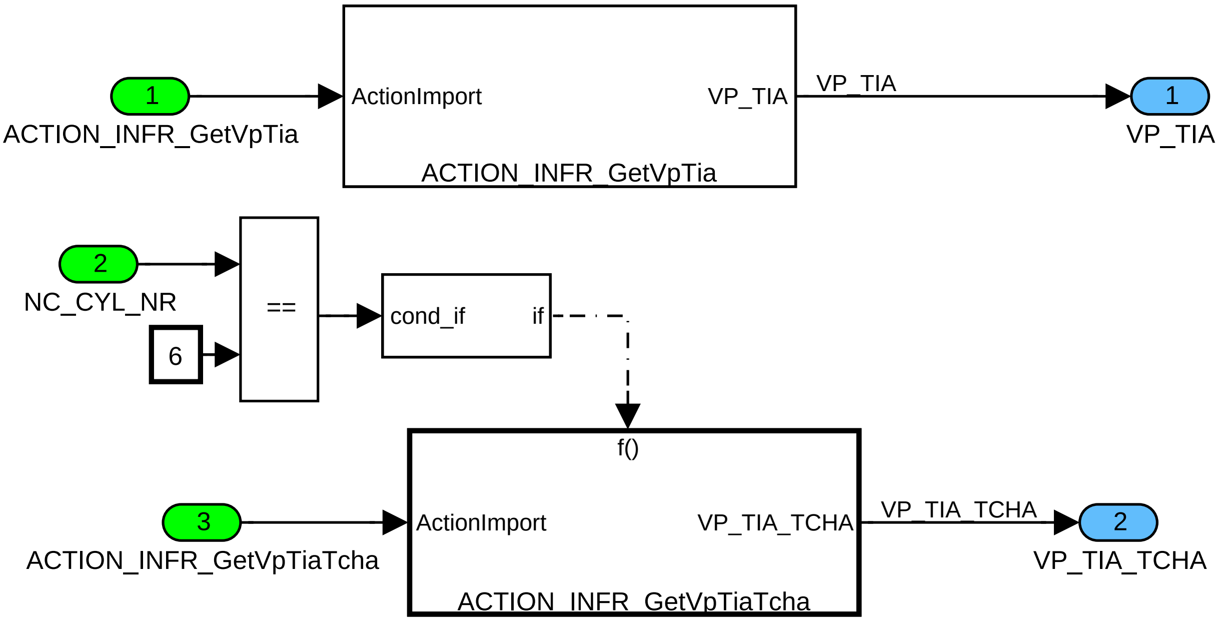


Figure 15 AIRT_M4007/ OPM/ CLC/ CLC_1_VP_TIA

Determination of TIA , TIA_TCHA, TIA_MES and TIA_SUB

Calculation of TIA is done using TIA_SUB, Output from temperature gradient monitoring scheme and old value of TIA .TIA_TCHA is calculated using TIA and IP MAP.TIA_MES is calculated from the IP table.

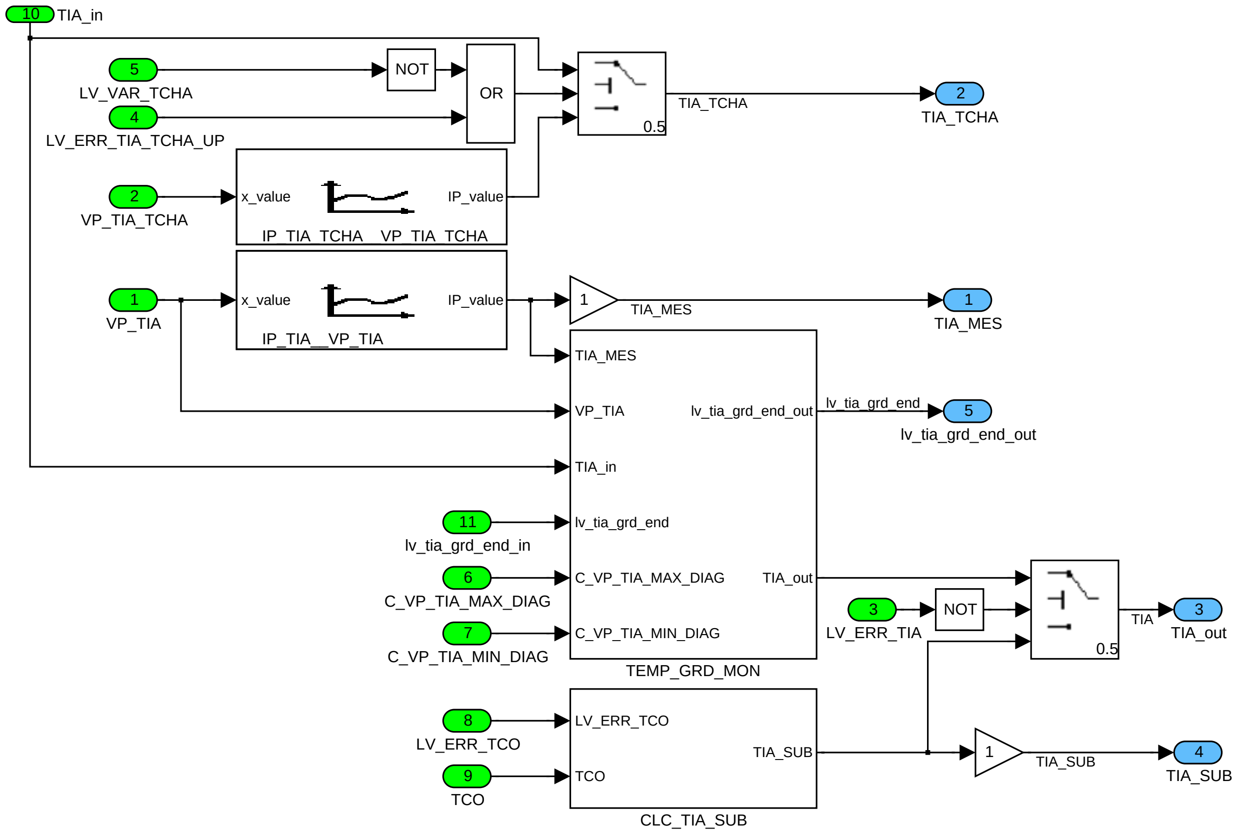


Figure 16 AIRT_M4007/ OPM/ CLC/ CLC_2_TIA_ERR

Determination TIA_SUB

TIA_SUB is determined from lower TIA substitute value (C_TIA_SUB_BOL) and upper TIA substitute value (C_TIA_SUB_TOL).

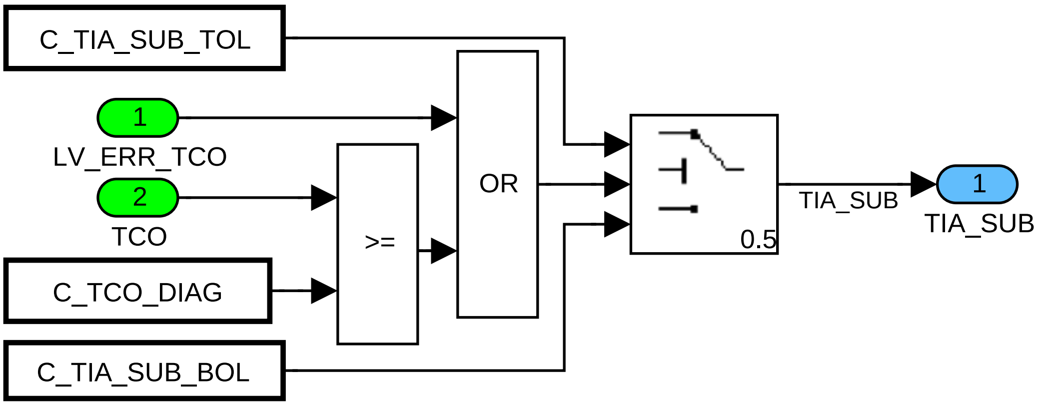


Figure 17 AIRT_M4007/ OPM/ CLC/ CLC_2_TIA_ERR/ CLC_TIA_SUB

Temperature gradient monitoring

General information:

The temperature TIA is monitored for unplausible gradients. Previously there is a check that the measured value is within the permissible value range (see chapter Diagnosis). If yes, a check is made that the difference between the old and the new measured value exceeds the permissible gradient

C_TIA_GRD_MAX_DIAG for TIA.

If this is the case, then the old measured value remains unchanged. If it is determined once again at the next measurement that the maximum gradient was exceeded, then the new measured value is taken over into the working memory. A measured value can therefore only be masked out if the gradient was exceeded for the first time.

Gradient monitoring does not lead to any error storage, but serves only for masking out not plausible measured values (disturbances).

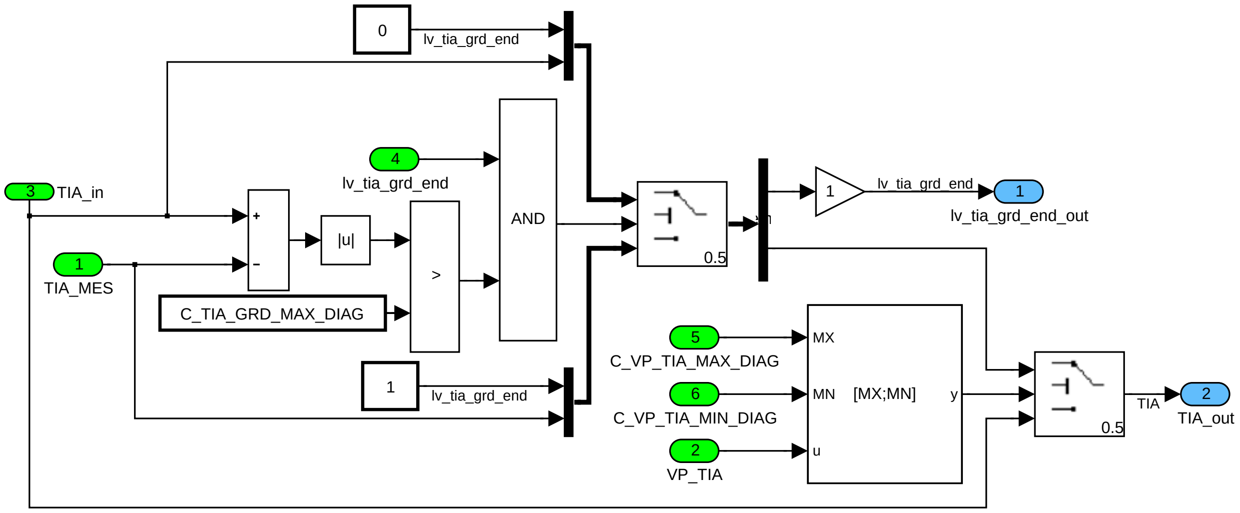


Figure 18 AIRT_M4007/ OPM/ CLC/ CLC_2_TIA_ERR/ TEMP_GRD_MON

1.4.1.3 Calculation at Engine stall to engine run event

Assignment of TIA to TIA_ST

TIA is assigned to TIA_ST

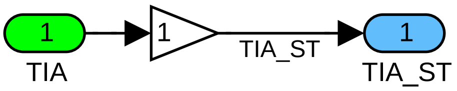


Figure 19 AIRT_M4007/ ES2ERU/ OPM

1.5 Electrical TIA diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_TIA_EL	O/V	0...1H	0...1	1	-
End of Diagnosis cycle TIA_EL					
LV_ERR_TIA_EL	O/V	0...1H	0...1	1	-
Electrical TIA error					
LV_END_DIAG_TIA_GRD	O/V	0...1H	0...1	1	-
End of Diagnosis cycle TIA_GRD					
LV_ERR_TIA_GRD	O/V	0...1H	0...1	1	-
Boolean for intake air temperature signal gradient error					
ERR_SYM_TIA_GRD	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom TIA gradient					
LV_CDN_DIAG_TIA_EL	V	0...1H	0...1	1	-
Diagnosis condition TIA_EL					
ERR_SYM_TIA_EL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom TIA_EL					
LV_CDN_DIAG_TIA_GRD	V	0...1H	0...1	1	-
Diagnosis condition TIA gradient					

Input data:

LV_IGK	T_AST_DIAG	VP_TIA	NC_IDX_DIAG_TIA_EL
NC_IDX_DIAG_TIA_GRD	TIA_MES	TCO	TAM

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_TIA_GRD	1	0...FFH	0...255	1	-
Anti-bounce decrement					
C_ABC_INC_TIA_EL	1	0...FFH	0...255	1	-
Anti-bounce increment					
C_ABC_INC_TIA_GRD	1	0...FFH	0...255	1	-
Anti-bounce increment					
C_ABC_MAX_TIA_EL	1	1...FFH	1...255	1	-
Anti-bounce maximum					
C_ABC_MAX_TIA_GRD	1	1...FFH	1...255	1	-
Anti-bounce maximum					
C_TIA_MAX_DIAG_GRD_1	1	0...FEH	0...190.5	0.75	°C
Maximum intake temperature gradient for TIA diagnosis part 1					
C_TIA_MAX_DIAG_GRD_2	1	0...FEH	0...190.5	0.75	°C
Maximum intake temperature gradient for TIA diagnosis part 2					
C_TIA_RNG_TCO_DIAG_H	1	0...FEH	-48...142.5	0.75	°C
High range for the detection of engine at cold operating temperature					
C_TIA_RNG_TCO_DIAG_L	1	0...FEH	-48...142.5	0.75	°C
Low range for the detection of engine at cold operating temperature					
C_TIA_RNG_TCO_DIAG_THD	1	0...FEH	-48...142.5	0.75	°C
Threshold range for the detection of tuning manipulation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_TIA_DIAG	1	0...FFFFH	0...3.27675E+4	0.5	s
Delay time for diagnosis activation					
C_T_DLY_TIA_GRD	1	0...FFFFH	0...3.27675E+4	0.5	s
Delay time for intake air temperature gradient					
C_VP_TIA_MAX_DIAG	1	0...7FFFH	0...4.99984741	1.52588E-4	V
upper threshold for range check of VP_TIA					
C_VP_TIA_MIN_DIAG	1	0...7FFFH	0...4.99984741	1.52588E-4	V
lower threshold for range check of VP_TIA					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the status of the failure availability
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

1.5.1 Function description

General information:

The purpose of the diagnosis shall be to detect electrical faults as defined by OBD I requirements. The input signal is analog from a NTC.

Description:

Error symptoms are defined for this diagnosis function as:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘

│ │ │ └─> Short circuit to Vbatt or open load        (= SYM_0)
│ │ └───> Short circuit to GND                       (= SYM_1)
│ └─────> -                                    (= SYM_2)
└───────> -                                    (= SYM_3)
```

Application Condition

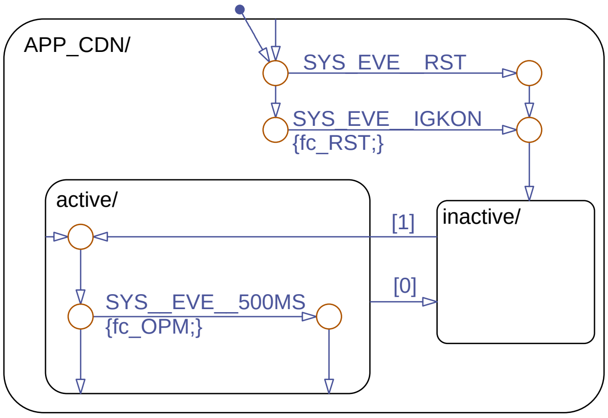
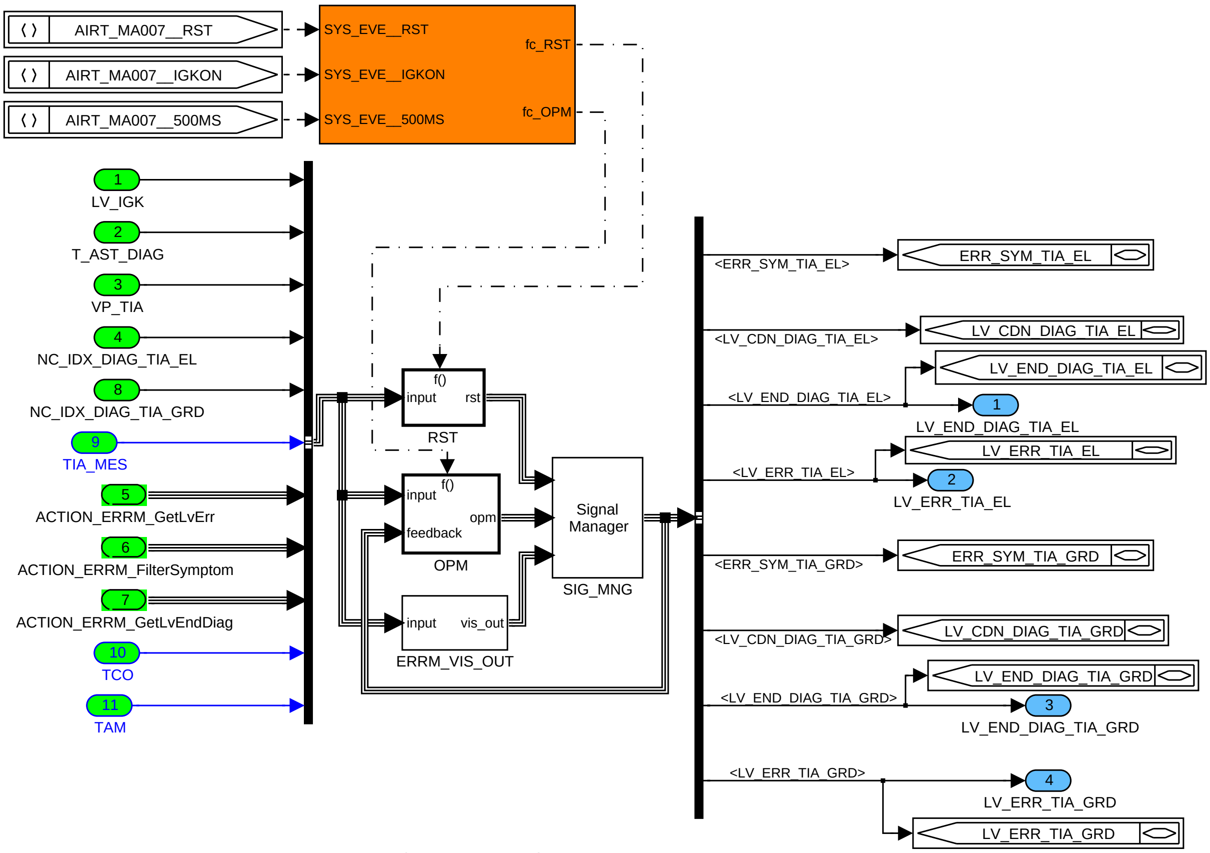


Figure 20 AIRT_MA007/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 05−Feb−2007

Figure 21 AIRT_MA007

1.5.1.1 Reset

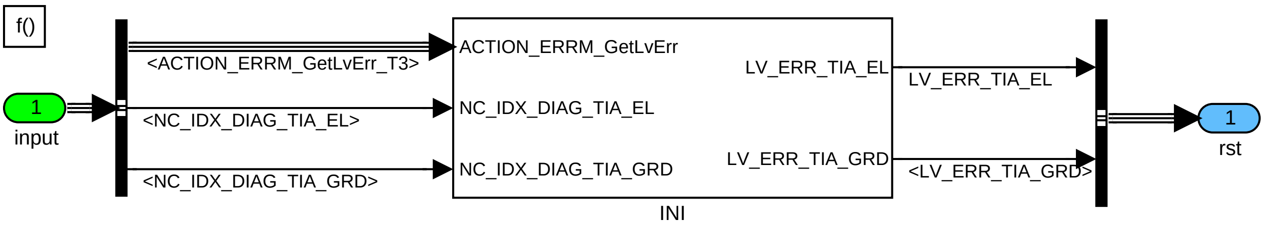


Figure 22 AIRT_MA007/ RST

Calculation at reset

All output data are according to ABC configuration "STD_INI"

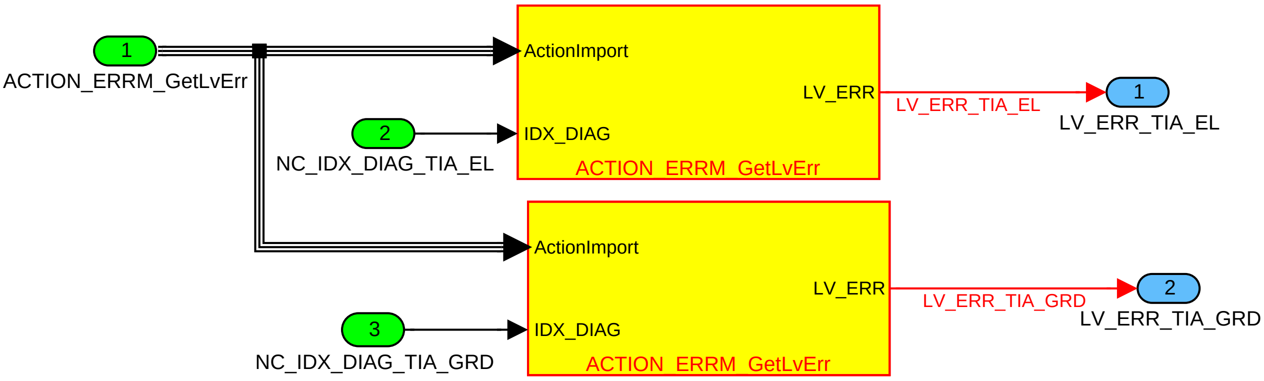


Figure 23 AIRT_MA007/ RST/ INI

1.5.1.2 Formula section

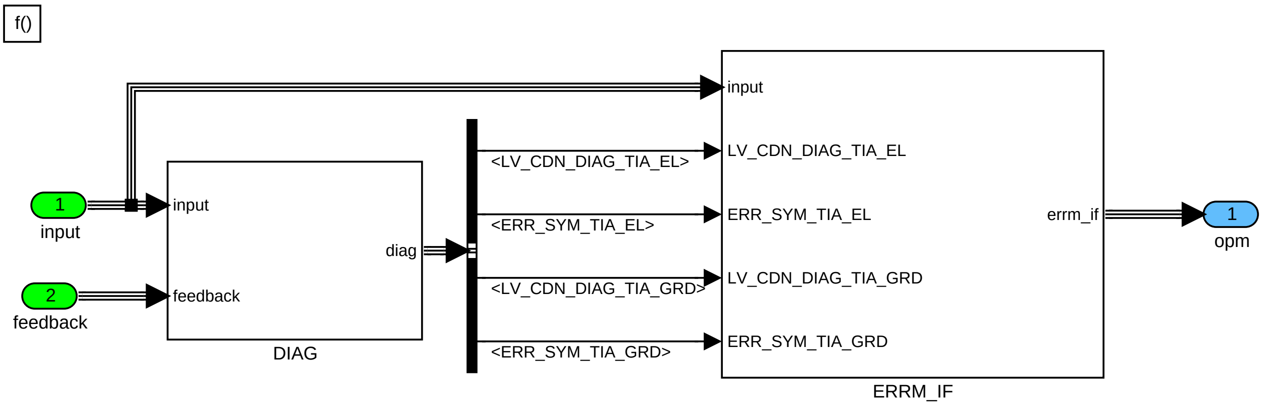


Figure 24 AIRT_MA007/ OPM

Diagnosis

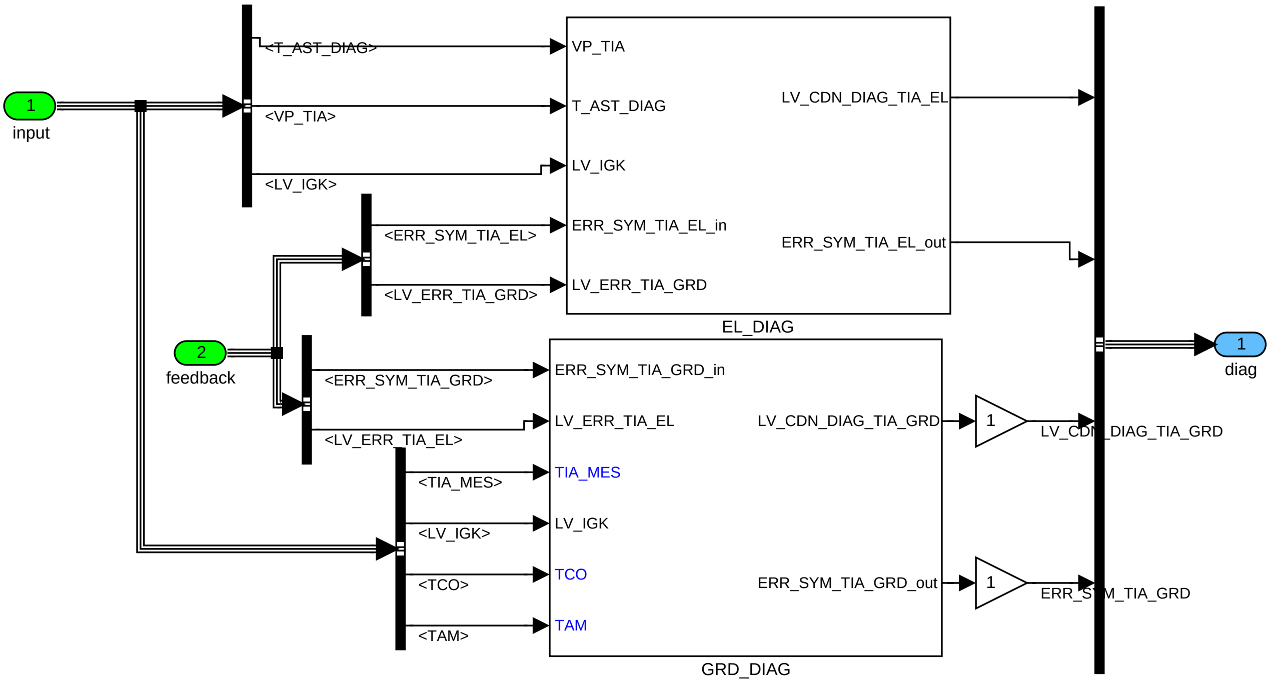


Figure 25 AIRT_MA007/ OPM/ DIAG

Electrical diagnosis

Short circuit to ground, short circuit to battery or open load can be detected

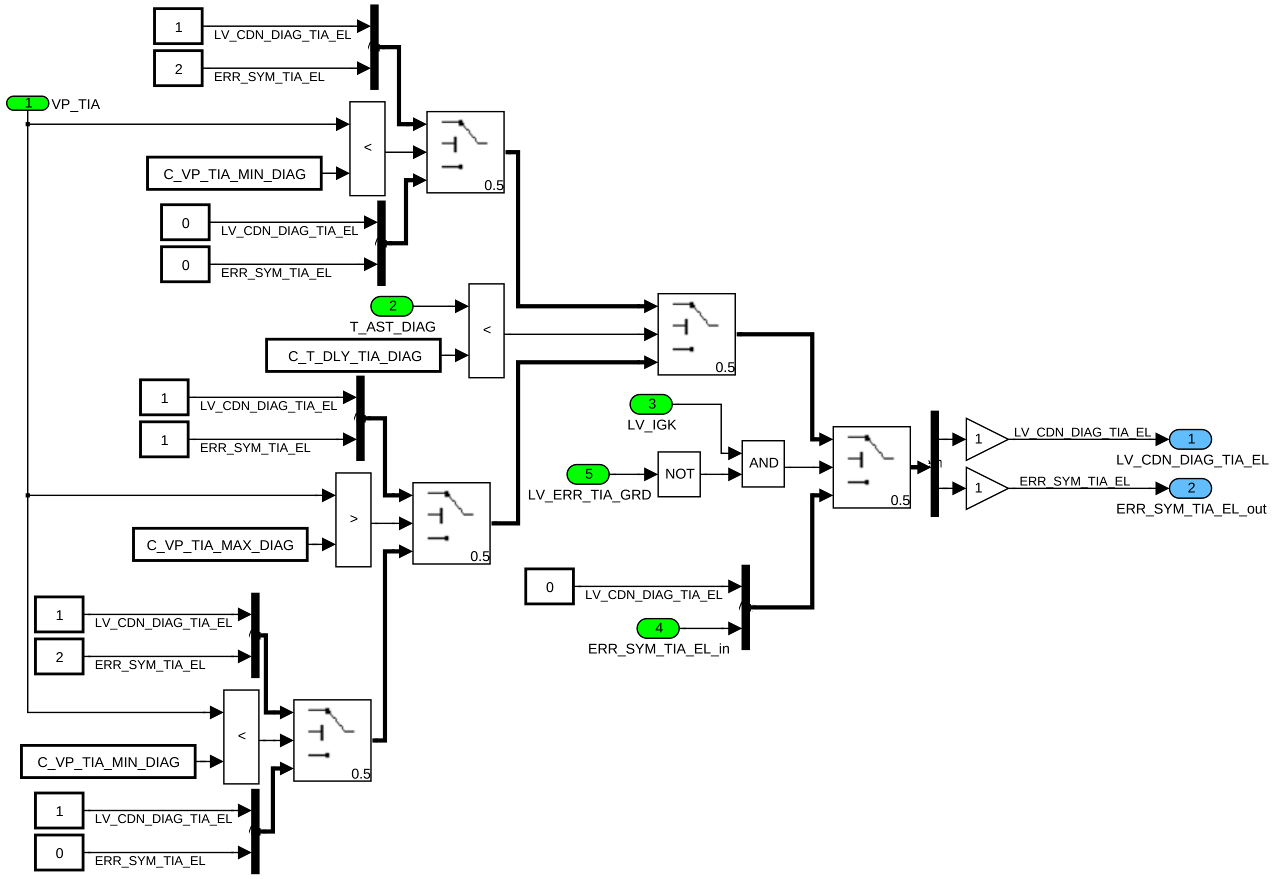


Figure 26 AIRT_MA007/ OPM/ DIAG/ EL_DIAG

Intake air temperature signal gradient diagnosis

The purpose of the signal gradient diagnosis is to detect a not plausible gradient on the intake air temperature signal. The monitoring is based on a comparison between the actual and the last intake air temperature value. The symptom of the error code is handled by anti-bouncing.

Error-symptoms are defined to this diagnosis function as following:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 | | | └─>                                              (= SYM_0)
 | | └───>                                              (= SYM_1)
 | └──────> -Intake air temperature range check error   (= SYM_2)
 └────────> -Intake air temperature signal gradient error     (= SYM_3)
```

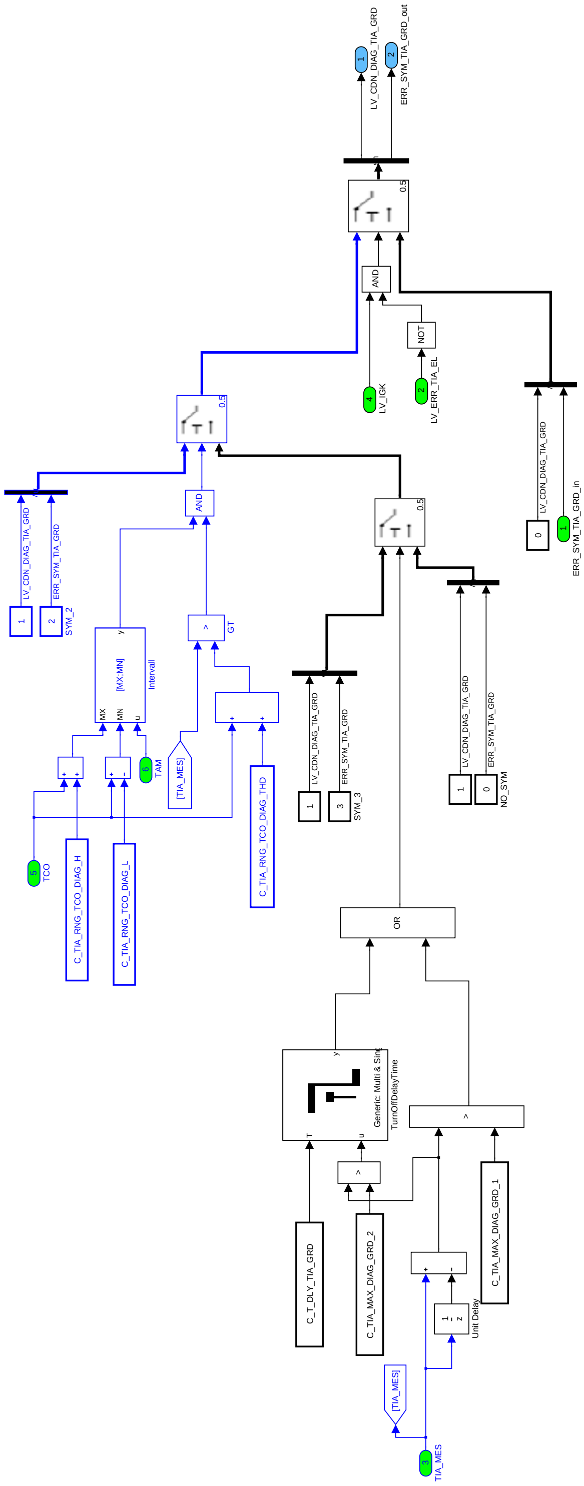


Figure 27 AIRT_MA007/ OPM/ DIAG/ GRD_DIAG

ERRM_IF

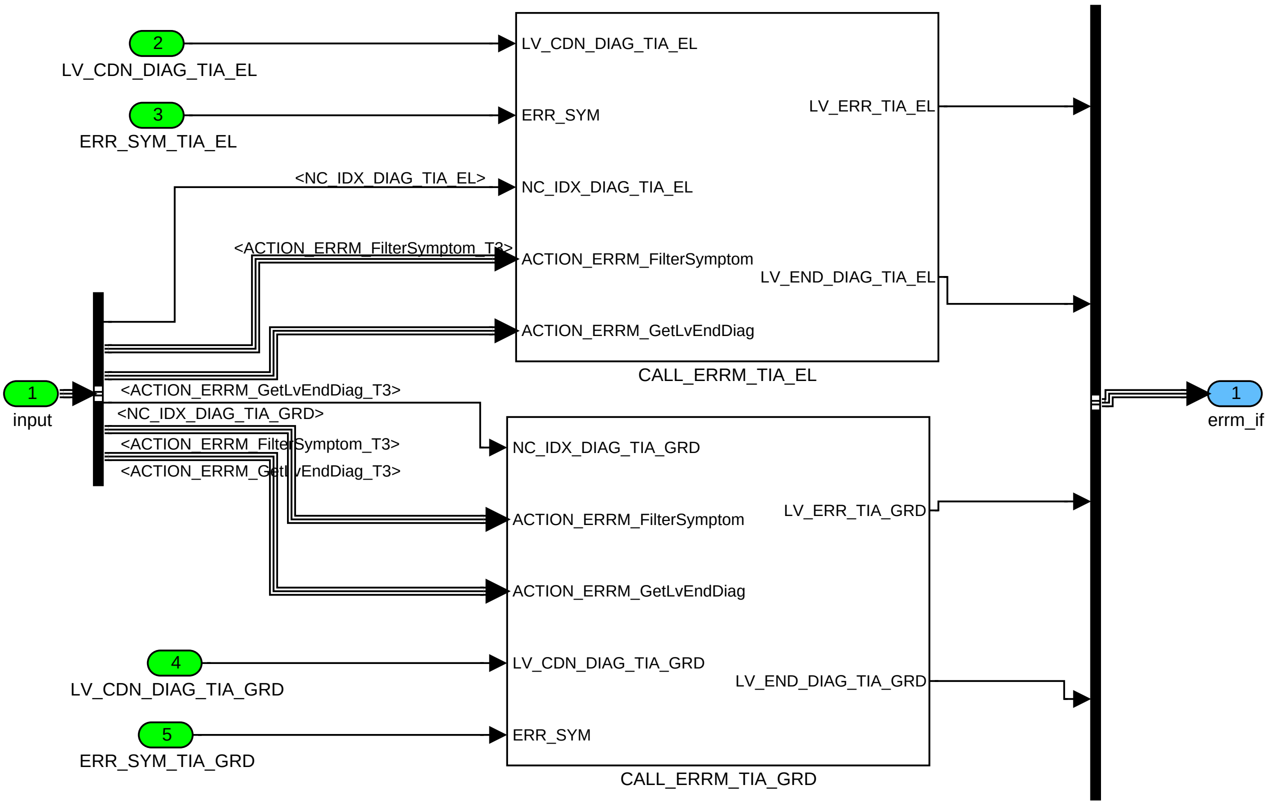


Figure 28 AIRT_MA007/ OPM/ ERRM_IF

ERRM inteface

Calculation of present error and of diagnosis LV_ERR_TIA_EL and LV_END_DIAG_TIA_EL are calculed here by error management

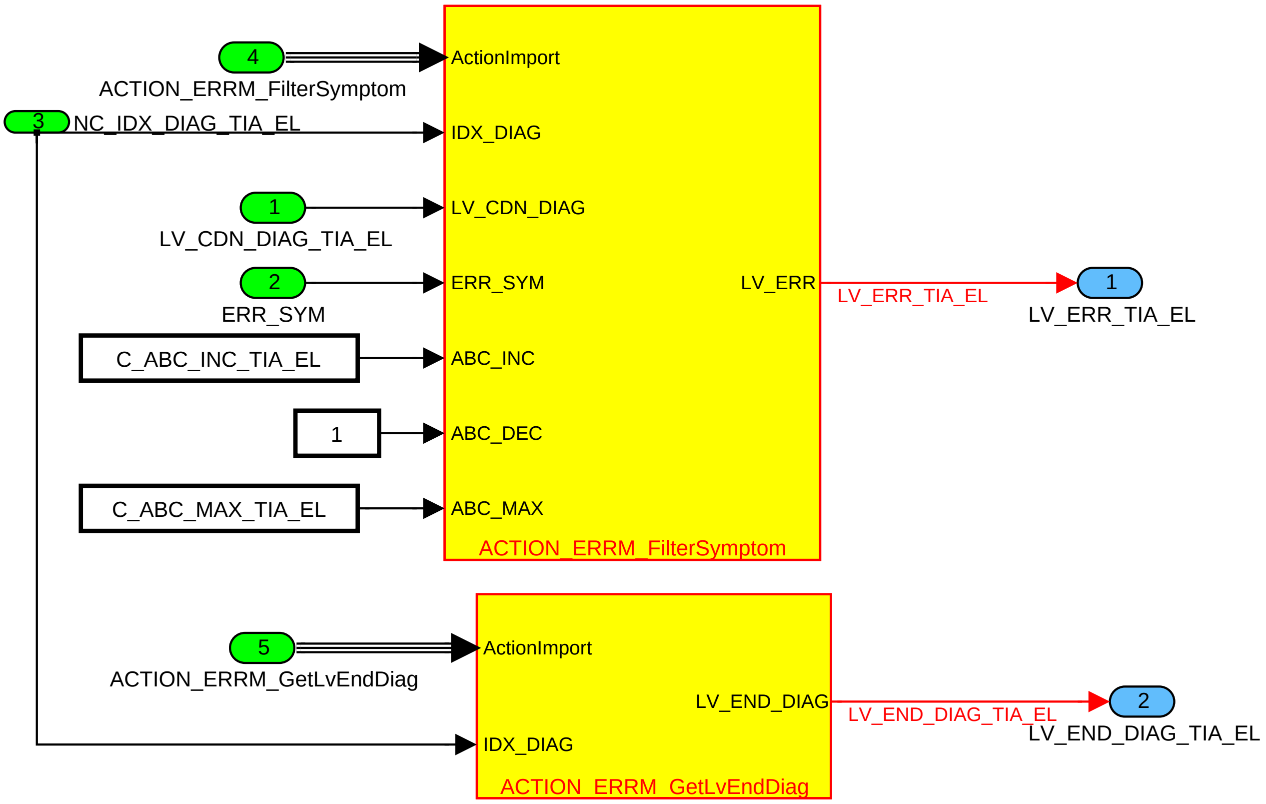


Figure 29 AIRT_MA007/ OPM/ ERRM_IF/ CALL_ERRM_TIA_EL

ERRM interface

Calculation of present error and of diagnosis LV_ERR_TIA_GRD and LV_END_DIAG_TIA_GRD are calculed here by error management

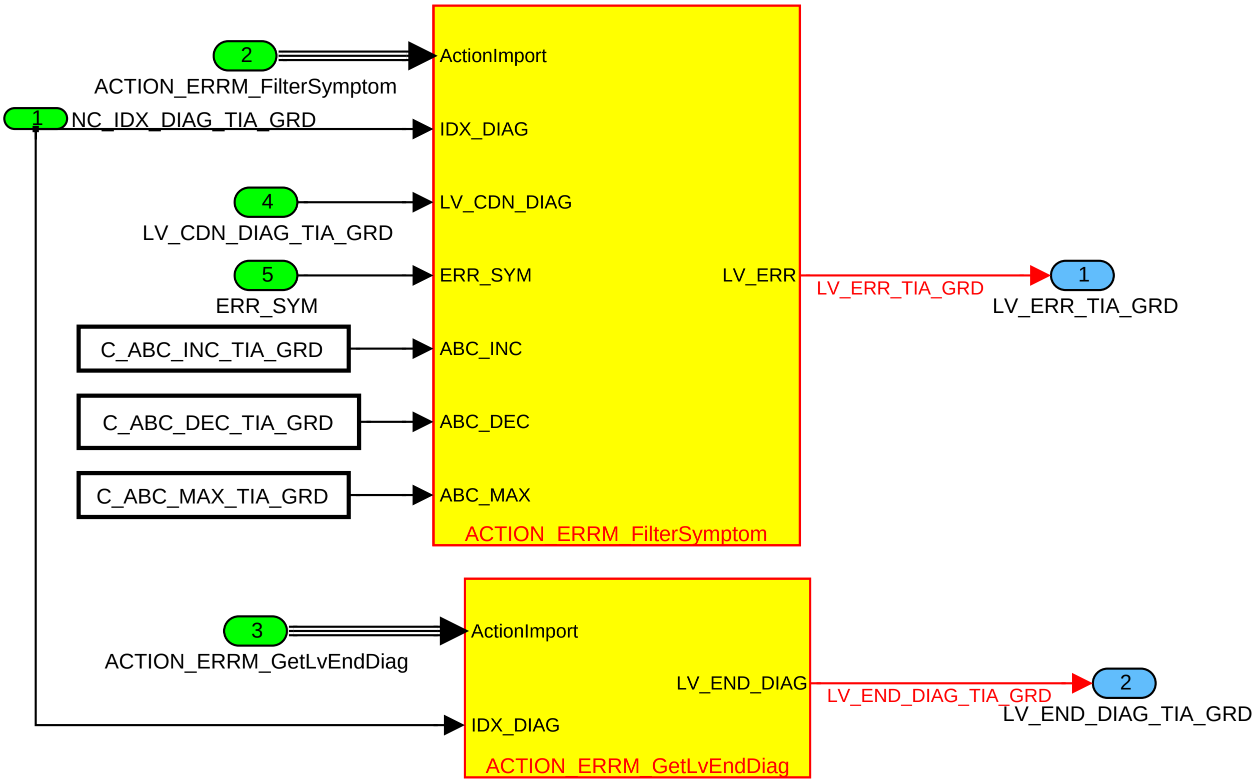


Figure 30 AIRT_MA007/ OPM/ ERRM_IF/ CALL_ERRM_TIA_GRD

1.5.1.3 ERRM_VIS_OUT

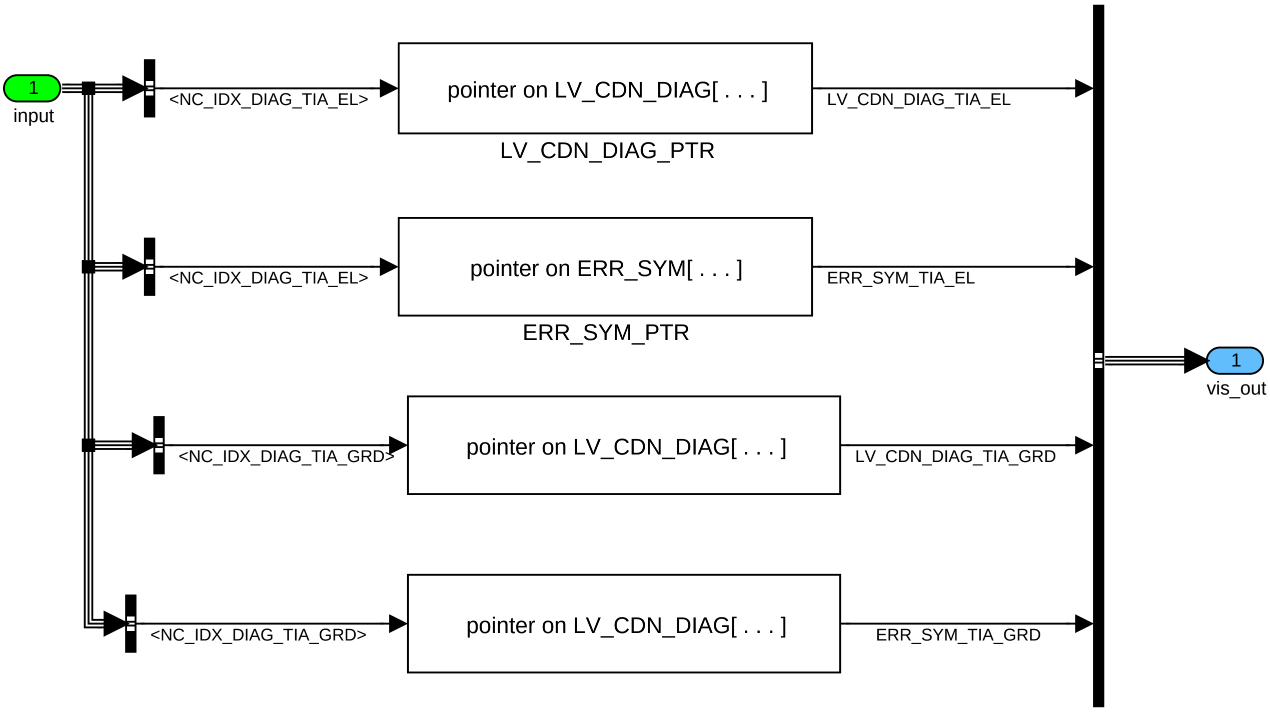


Figure 31 AIRT_MA007/ ERRM_VIS_OUT

1.6 Global TIA system error (LV_ERR_TIA)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TIA	O/V	0...1H	0...1	1	-
TIA error present					

Input data:

LV_ERR_TIA_EL	LV_ERR_TIA_PLAUS	LV_IGK	LV_ERR_TIA_GRD

1.6.1 Function Description

General information:

The global error flag LV_ERR_TIA can be used for inhibition. It is set if a electrical or a plausibility TIA error occur.

Application Condition

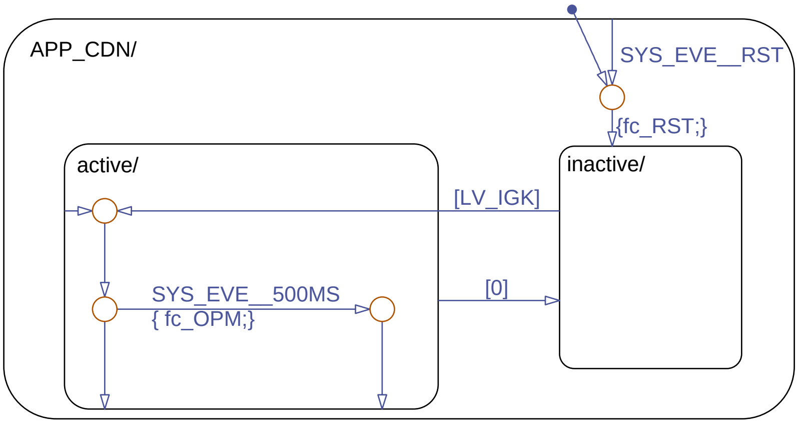


Figure 32 AIRT_MA025/ APP_CDN/ Chart

Function Description

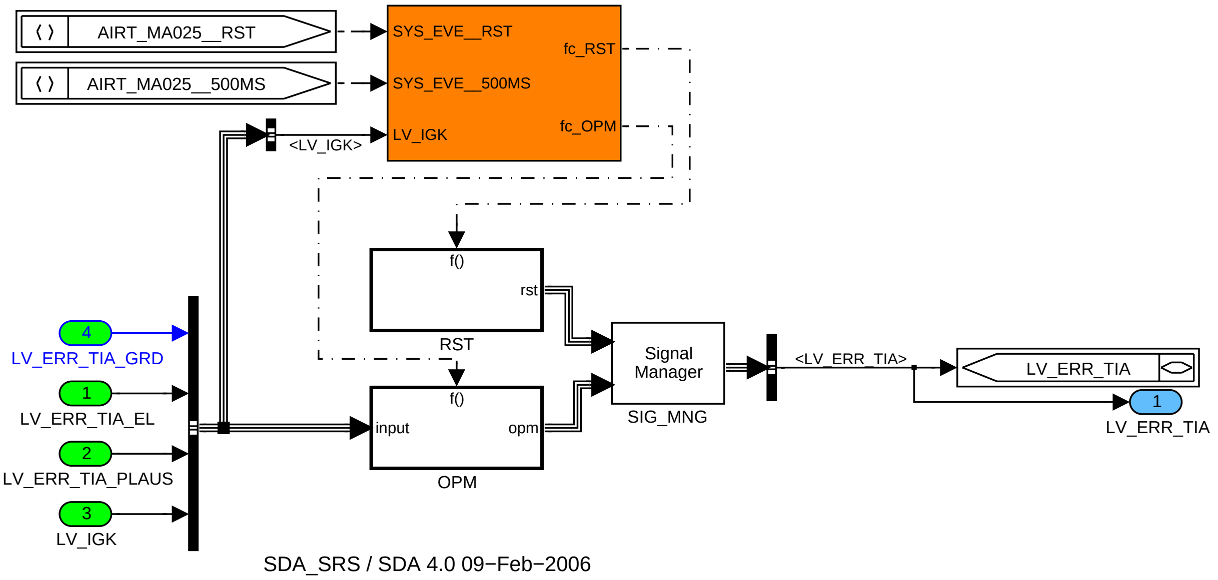


Figure 33 AIRT_MA025

1.6.1.1 Reset calculation

Initialisation of LV_ERR_TIA

LV_ERR_TIA is initialised to 0.

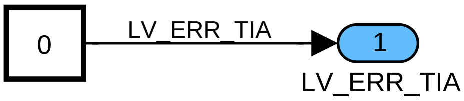


Figure 34 AIRT_MA025/ RST/ INI

1.6.1.2 Formula Section

Output is calculated at 500ms recurrence.

Calculation of LV_ERR_TIA

If LV_ERR_TIA_PLAUS or LV_ERR_TIA_EL is set, then LV_ERR_TIA flag is set.

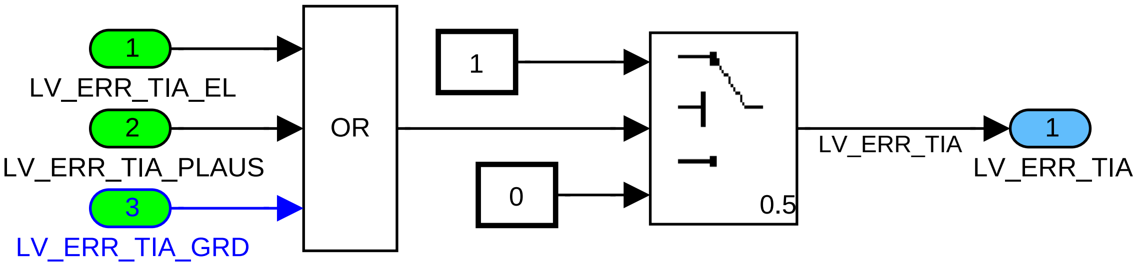


Figure 35 AIRT_MA025/ OPM/ CLC_INH_ERR_TIA

1.7 Electrical TIA_TCHA_UP diagnosis (LV_ERR_TIA_TCHA_UP)

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
LV_END_DIAG_TIA_TCHA_UP	O/V	0...1H	0...1	1	-
End of Diagnosis cycle TIA_TCHA_UP					
LV_ERR_TIA_TCHA_UP	O/V	0...1H	0...1	1	-
Electrical TIA_TCHA_UP error					
ERR_SYM_TIA_TCHA_UP	V	0H	NO_SYM	1	-
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Detected symptom TIA_TCHA_UP					
LV_CDN_DIAG_TIA_TCHA_UP	V	0...1H	0...1	1	-
Diagnosis condition TIA_TCHA_UP					

Input data:

C_T_DLY_TIA_DIAG	LV_IGK	LV_VAR_TCHA	T_AST_DIAG
VP_TIA_TCHA	NC_IDX_DIAG_TIA_TCHA_UP		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TIA_TCHA_UP	1	0...FFH	0...255	1	-
Anti-bounce increment					
C_ABC_MAX_TIA_TCHA_UP	1	1...FFH	1...255	1	-
Anti-bounce maximum					
C_VP_TIA_TCHA_UP_MAX_DIAG	1	0...7FFFH	0...4.99984741	1.52588E-4	V
TIA threshold for short to ground ¿ TIA_TCHA_UP diagnosis					
C_VP_TIA_TCHA_UP_MIN_DIAG	1	0...7FFFH	0...4.99984741	1.52588E-4	V
TIA threshold for short to VB or line break ¿ TIA_TCHA_UP diagnosis					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the status of the failure availability
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

1.7.1 Function Description

General information:

The purpose of the diagnosis shall be to detect electrical faults as defined by OBD I requirements. The input signal is analog from a NTC located at the Mass air flow meter.

Description:

Error symptoms are defined for this diagnosis function as:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘

 |  |  |  └─> Short circuit to Vbatt or open load        (= SYM_0)
 |  |  └───> Short circuit to GND                    (= SYM_1)
 |  └─────> -                                       (= SYM_2)
 └───────> -                                       (= SYM_3)
```

Function Description

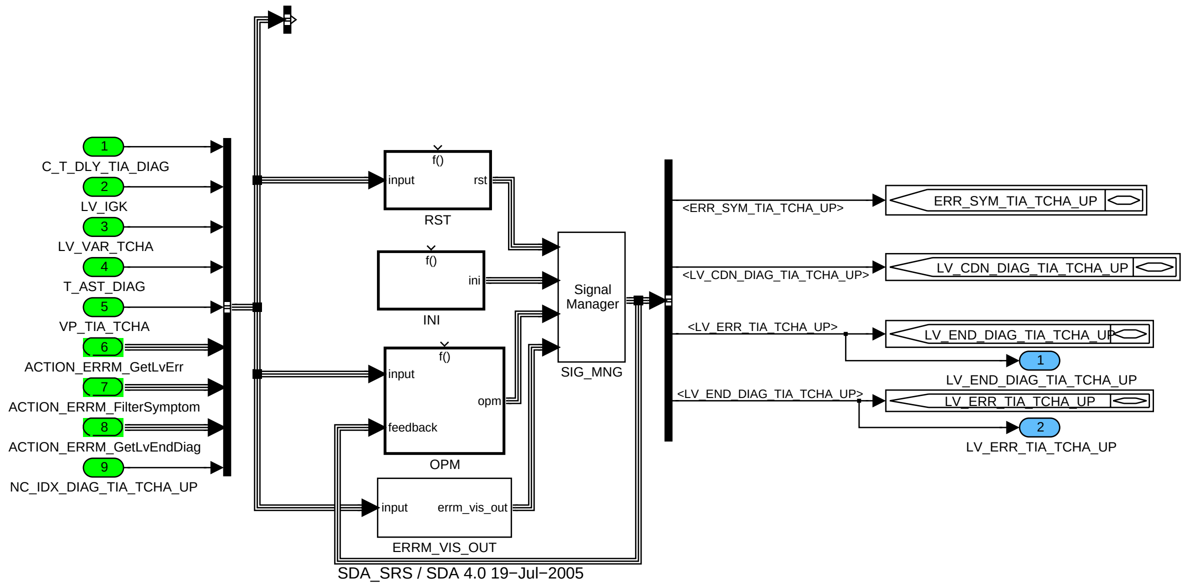


Figure 36 AIRT_MA0C8

1.7.1.1 Initialisation at system event RESET and IGKON

Initialisation

LV_ERR_TIA_TCHA_UP is initialised.

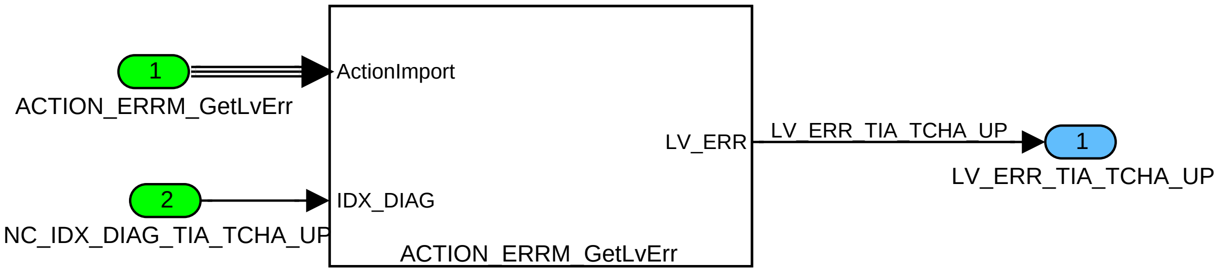


Figure 37 AIRT_MA0C8/ RST/ INI

1.7.1.2 At deactivation

At deactivation LV_END_DIAG_TIA_TCHA_UP is initialised to 1.

Figure 38 AIRT_MA0C8/ INI

1.7.1.3 Formula section

Diagnosis

Short circuit to ground , short circuit to Vbatt or open load are detected.

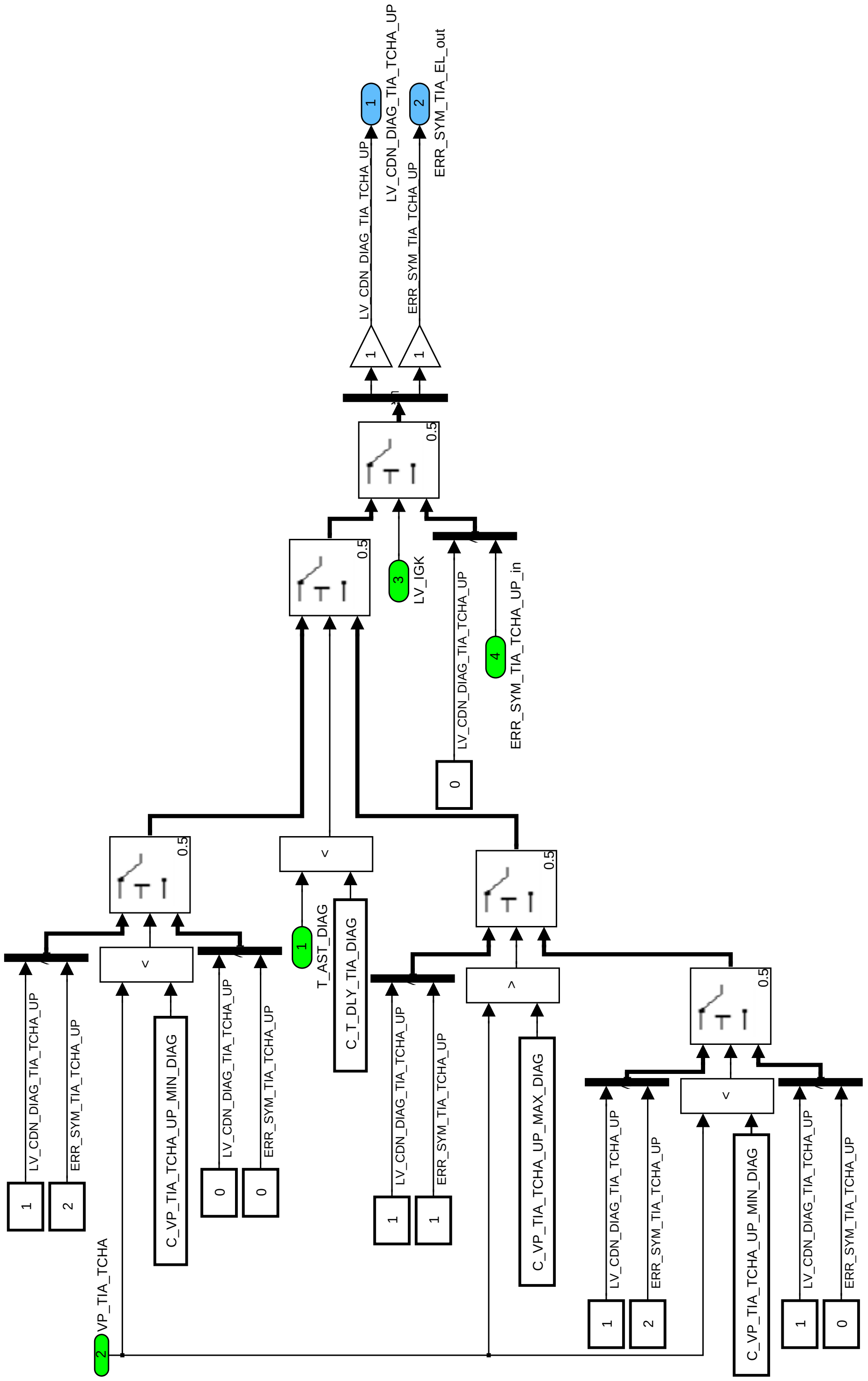


Figure 39 AIRT_MA0C8/ OPM/ DIAG

ERRM interface

Calculation of present error and end of diagnosis. LV_ERR_TIA_TCHA_UP and LV_END_DIAG_TIA_TCHA_UP are calculated by error management.

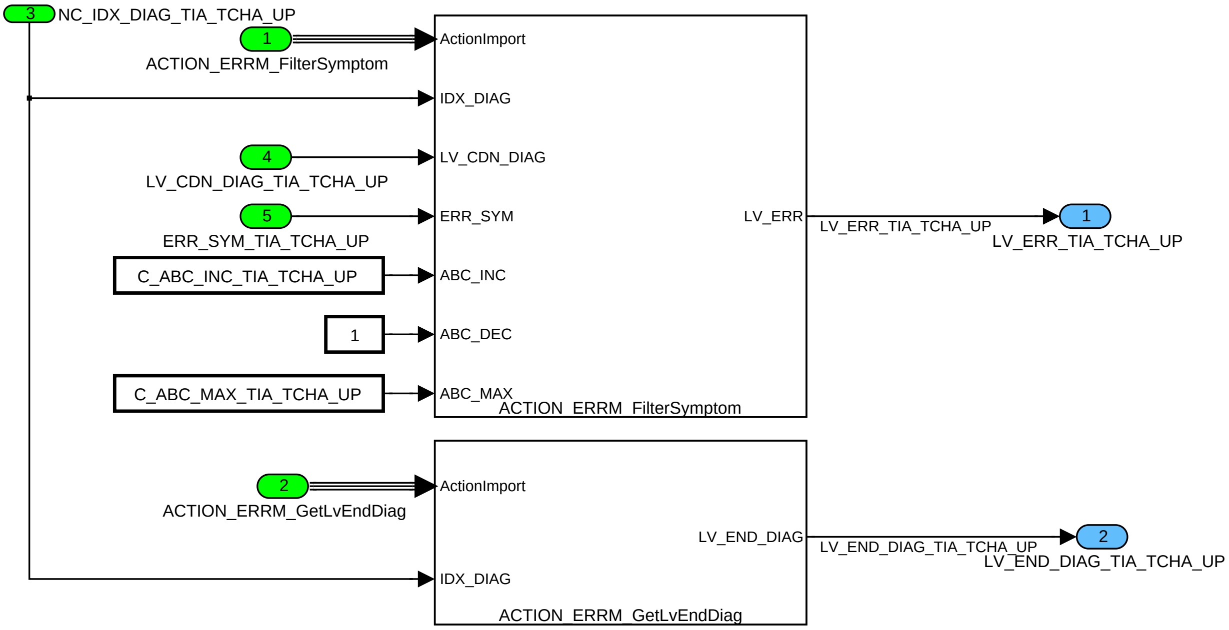


Figure 40 AIRT_MA0C8/ OPM/ ERRM_IF

1.7.1.4 Visual mode output calculation

Error symptom and condition calculation

ERR_SYM_TIA_TCHA_UP and LV_CDN_DIAG_TIA_TCHA_UP are calculated by error managerment in visual mode.

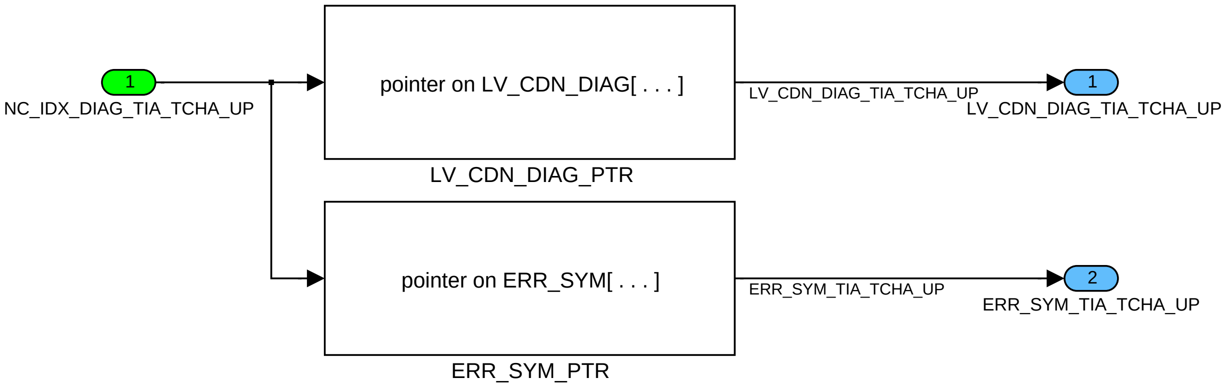


Figure 41 AIRT_MA0C8/ ERRM_VIS_OUT/ SUB_CLC

1.8 TIA plausibility check (LV_ERR_TIA_PLAUS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_TIA_PLAUS	O/V	0...1H	0...1	1	-
End of diagnosis cycle TIA_PLAUS					
LV_ERR_TIA_PLAUS	O/V	0...1H	0...1	1	-
temperature air sensor plausibility error					
CTR_IS_CDN_TIA_PLAUS	V	0...FFH	0...255	1	-
Cycle counter for T_IS condition					
TIA_DIF_DIAG	V	0...FEH	0...190.5	0.75	°C
Difference between TIA and TIA_ST					
TIA_DIF_MAX_DIAG	V	0...FEH	-48...142.5	0.75	°C
Maximum reached TIA					
TIA_DIF_MIN_DIAG	V	0...FEH	-48...142.5	0.75	°C
Minimum reached TIA					
TIA_PLAUS_MAX_DIAG	V	0...FEH	-48...142.5	0.75	°C
Maximum expected TIA					
TIA_PLAUS_MIN_DIAG	V	0...FEH	-48...142.5	0.75	°C
Minimum expected TIA					
T_IS_TIA_PLAUS_DIAG	V	0...FFFFH	0...3.27675E+4	0.5	s
Time period - vehicle is driven in Idle					
T_LOAD_IS_CDN_TIA_PLAUS	V	0...FFFFH	0...3.27675E+4	0.5	s
Time period - vehicle is driven with high load					
T_VS_TIA_PLAUS_DIAG	V	0...FFFFH	0...3.27675E+4	0.5	s
Time period - vehicle is driven with certain speed					
ERR_SYM_TIA_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected error symptom TIA_PLAUS					
LV_CDN_DIAG_TIA_PLAUS	V	0...1H	0...1	1	-
Diagnosis condition TIA_PLAUS					
LV_DIF_CDN_TIA_PLAUS_DIAG	V	0...1H	0...1	1	-
Flag indicating that TIA signal was moving					
LV_IS_CDN_TIA_PLAUS_DIAG	V	0...1H	0...1	1	-
Flag indicating that vehicle idle condition is reached					
LV_LOAD_IS_CDN_TIA_PLAUS	V	0...1H	0...1	1	-
Flag indicating that vehicle was driven with high load to enter in idle condition					
LV_VS_CDN_TIA_PLAUS_DIAG	V	0...1H	0...1	1	-
Flag indicating that vehicle speed condition is reached					

Input data:

LV_IGK	LV_INH_DIAG_TIA_PLAUS	LV_IS	LV_ST_END
MAF	TAM	TAM_ST	TCO
TIA[NC_SENS_NR_TIA]	T_AST_DIAG	VS	NC_IDX_DIAG_TIA_PLAUS
TIA_ST	LV_END_DIAG_TIA_EL		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TIA_PLAUS	1	0...FFH	0...255	1	-
Anti bounce increment TIA_PLAUS diagnosis					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_MAX_TIA_PLAUS	1	1...FFH	1...255	1	-
Anti bounce maximum TIA_PLAUS diagnosis					
C_CTR_IS_CDN_TIA_PLAUS	1	0...FFH	0...255	1	-
Number of valid IS conditions cycles					
C_MAF_MIN_INT_CDN_TIA_PLAUS	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Threshold MAF to enable MAF_CYL integral					
C_TAM_MAX_TIA_PLAUS_DIAG	1	0...FEH	-48...142.5	0.75	°C
Maximum ambient temperture to activate the diagnosis condition					
C_TAM_MIN_TIA_PLAUS_DIAG	1	0...FEH	-48...142.5	0.75	°C
Minimum ambient temperture to activate the diagnosis condition					
C_TCO_MIN_IS_CDN_TIA_PLAUS	1	0...FEH	-48...142.5	0.75	°C
TIA offset to MAX/MIN range to detect a error					
C_TIA_DIF_DIAG	1	0...FEH	0...190.5	0.75	°C
Minimum temperature difference to detect a non-stucking TIA signal					
C_TIA_PLAUS_DIF_ERR	1	0...FEH	0...190.5	0.75	°C
TIA offset to MAX/MIN range to detect a error					
C_T_LOAD_IS_CDN_TIA_PLAUS	1	0...FFFFH	0...3.27675E+4	0.5	s
Threshold for load timer to enter IS condition					
C_T_VS_MIN_TIA_PLAUS_DIAG	1	0...FFFFH	0...3.27675E+4	0.5	s
Minimum time period with a certain vehicle speed to expect a moving TIA signal					
C_VS_MAX_IS_CDN_TIA_PLAUS	1	0...FFH	0...255	1	km/h
Maximum VS for valid IS condition					
C_VS_MIN_TIA_PLAUS_DIAG	1	0...FFH	0...255	1	km/h
Minimum VS for valid VS conditon cycle					
ID_T_IS_MIN_TIA_PLAUS_DIAG	4	0...FFFFH	0...3.27675E+4	0.5	s
LDP_TAM_TIA_PLAUS_DIAG	4	0...FEH	-48...142.5	0.75	°C
Minimum time period in IDLE to expect a moving TIA signal					

Import actions:

ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure
ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the status of the failure availability

1.8.1 Function Description

General information:

This diagnosis is performed in order to detect a stucking or not plausible TIA signal which cannot be detected by electrical range diagnosis.

Description:

The diagnosis can be inhibited in cold conditions (If TAM_ST <= C_TAM_MIN_TIA_PLAUS_DIAG).

The error detection is based on two different diagnosis:

Plausible range detection:

The plausible TIA must be between TCO and TAM. If TIA is outside of the range plus offset the error symptom MAX/MIN is set and the error is debounced.

Stucking signal:

If the vehicle was driven with a certain vehicle speed for a calibratable time (cool down of hot TIA) or vehicle was in IDLE for a calibratable time and cycles (hot up of cold TIA) and the TIA signal was not moving then a stucking TIA signal is detected and the error is debounced.

Error symptoms are defined for this diagnosis function as:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 |  |  |  └─> Signal to high          (= SYM_0)
 |  |  └────> Signal to low           (= SYM_1)
 |  └───────> -                       (= SYM_2)
 └──────────> Signal is not plausible (= SYM_3)
```

Application Condition

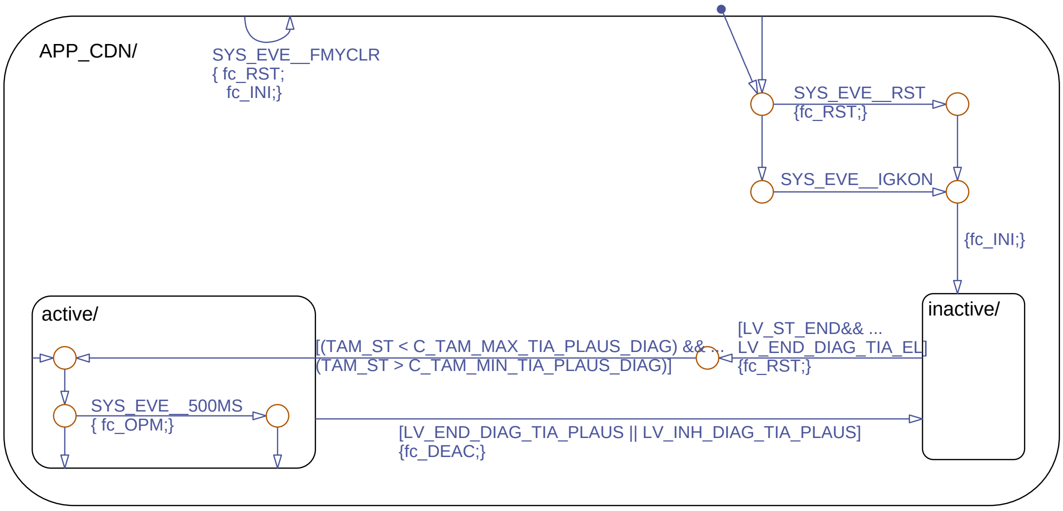


Figure 42 AIRT_MB051/ APP_CDN/ Chart

Function Description

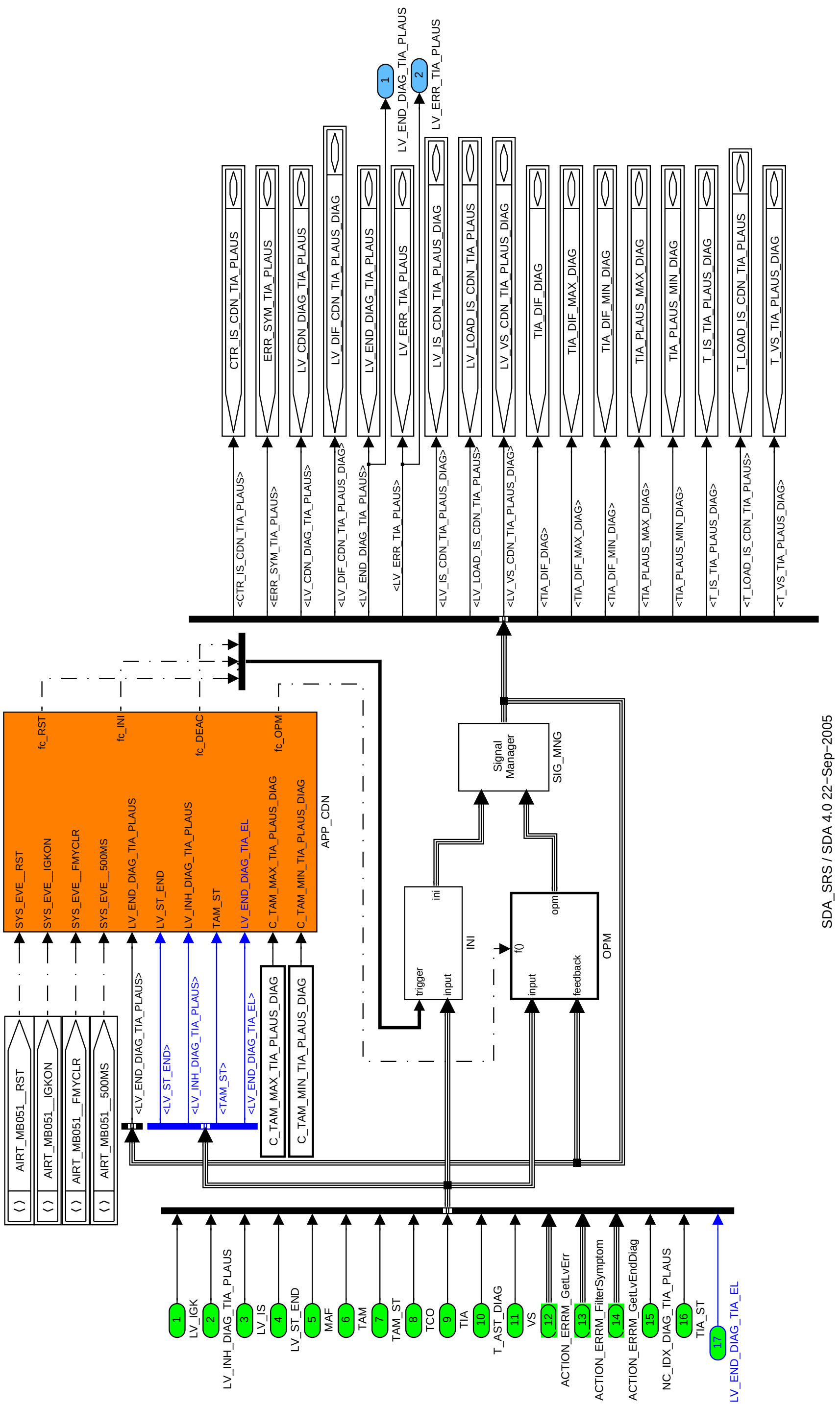


Figure 43 AIRT_MB051

1.8.1.1 Initialisations

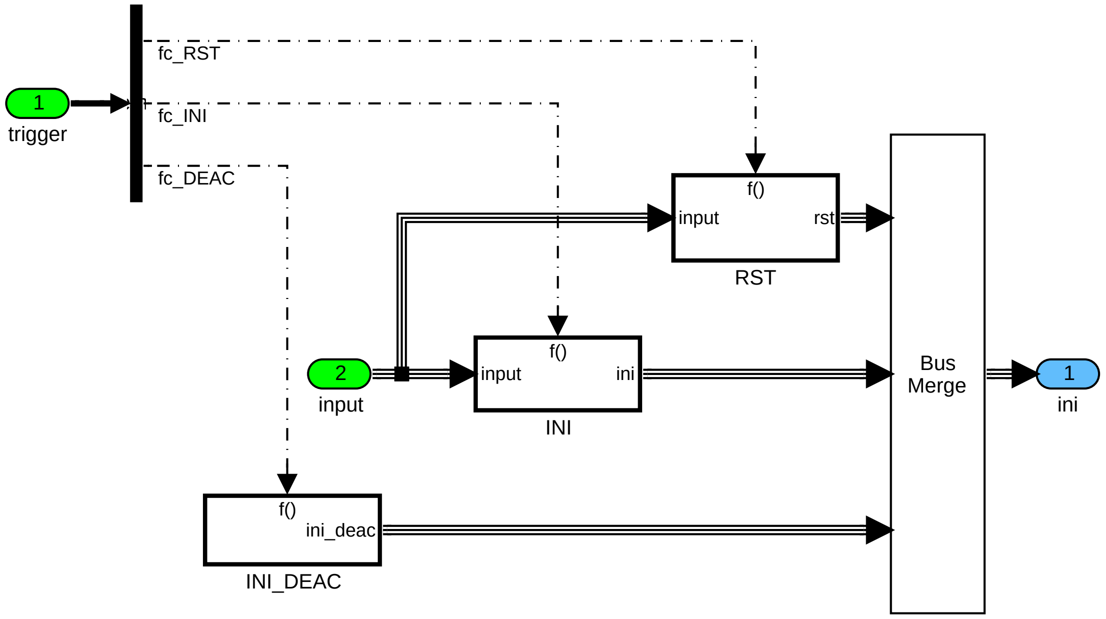


Figure 44 AIRT_MB051/ INI

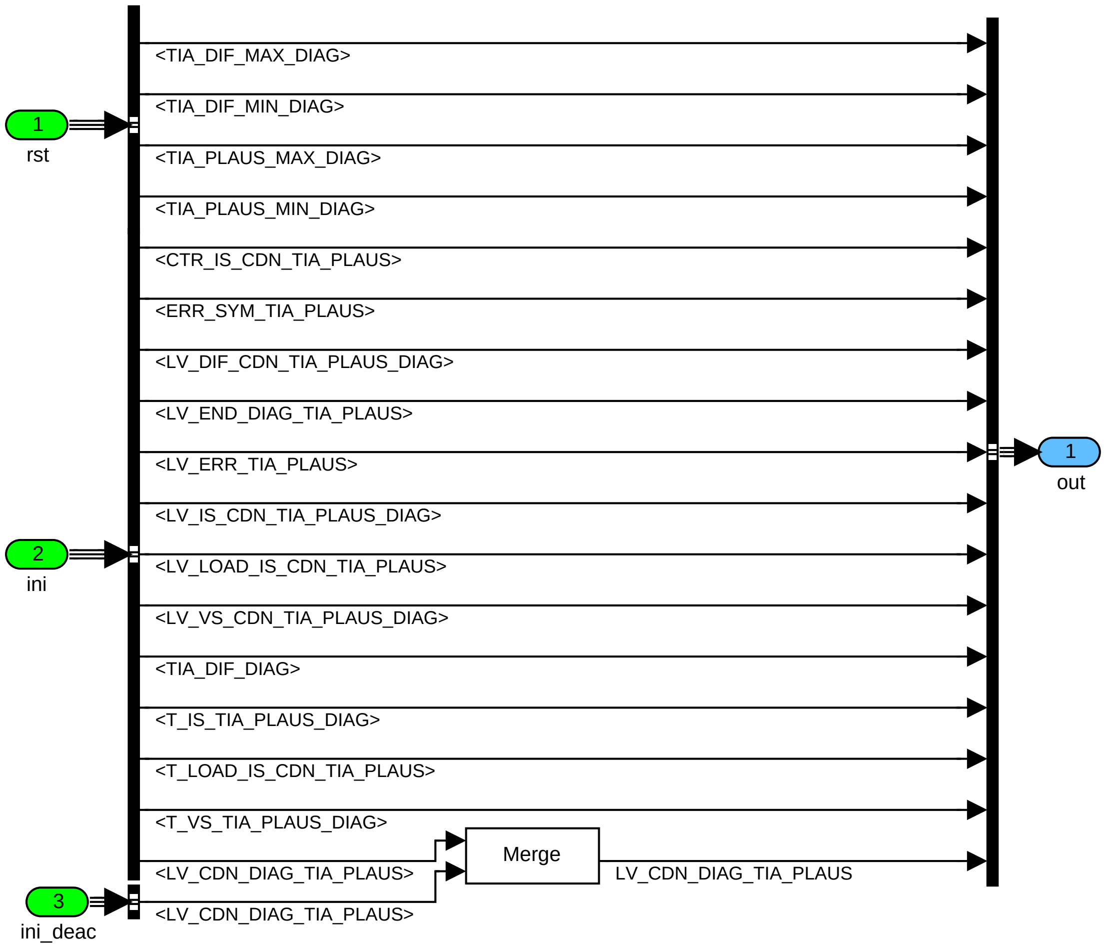


Figure 45 AIRT_MB051/ INI/ BUS_MERGE

Initialisation at LV_IGK 0 --> 1 or reset or clearing error memory

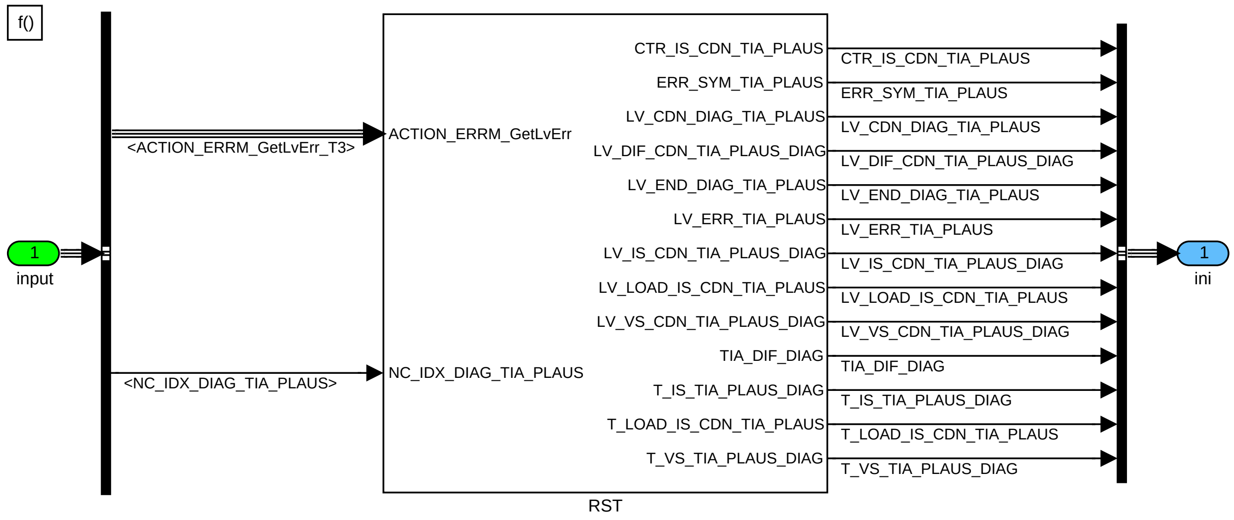


Figure 46 AIRT_MB051/ INI/ INI

Initialisation

All variables except TIA_DIF_MAX/MIN_DIAG and TIA_PLAUS_MAX/MIN_DIAG are initialised

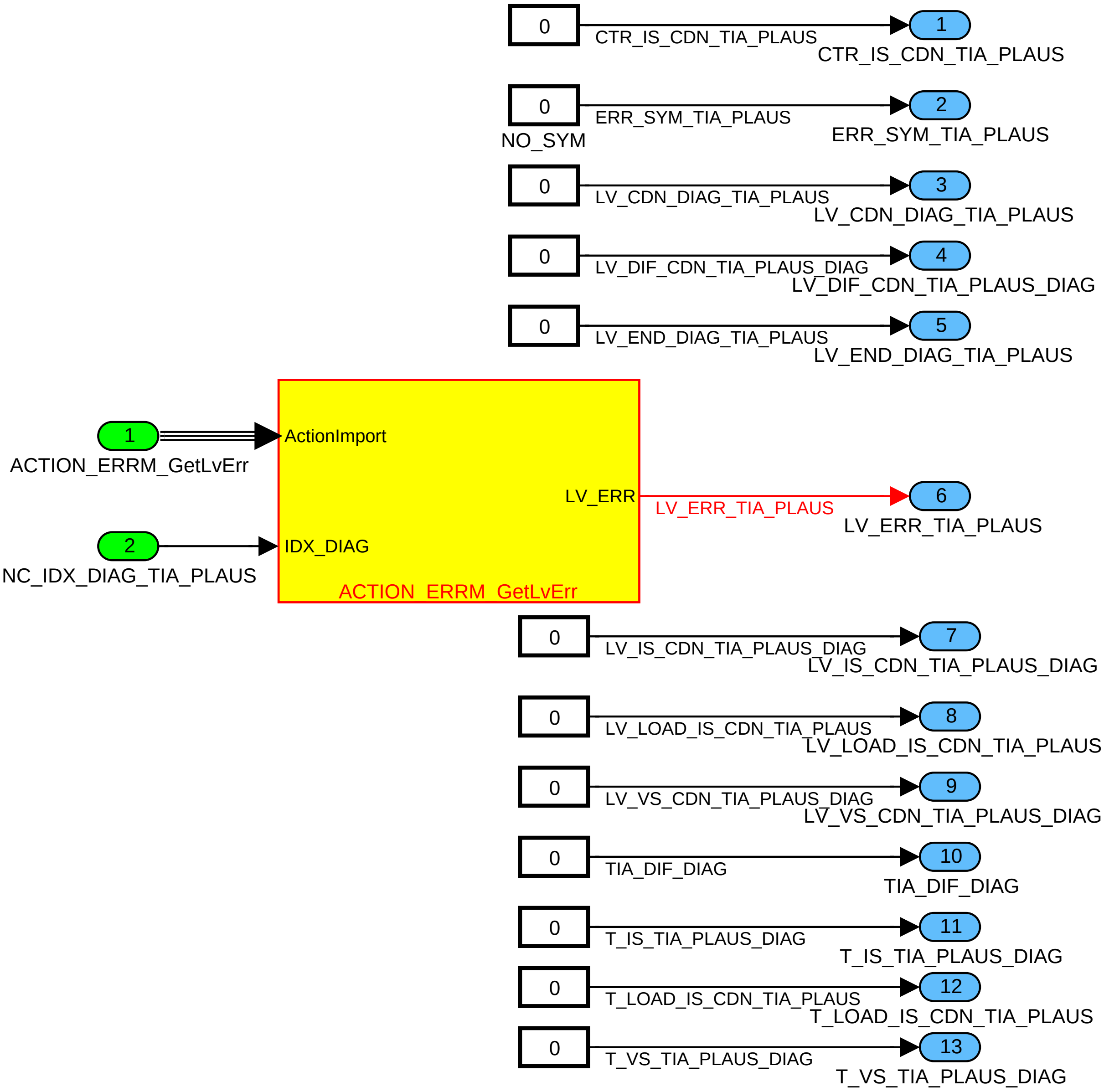


Figure 47 AIRT_MB051/ INI/ INI/ RST

Resetting the value of LV_CDN_DIAG_TIA_PLAUS at deactivation

LV_CDN_DIAG_TIA_PLAUS value is set to 0 when Inhibition bit or LV_END_DIAG_TIA_PLAUS bit goes low.

Figure 48 AIRT_MB051/ INI/ INI_DEAC

Reset at LV_ST_END = 1or reset or clearing error memory

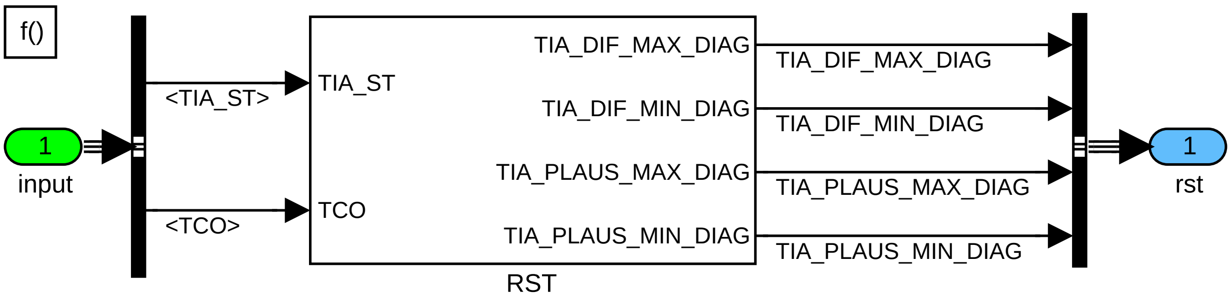


Figure 49 AIRT_MB051/ INI/ RST

Initialisation at reset

Initialisation of TIA_DIF_MAX/MIN_DIAG and TIA_PLAUS_MAX/MIN_DIAG.

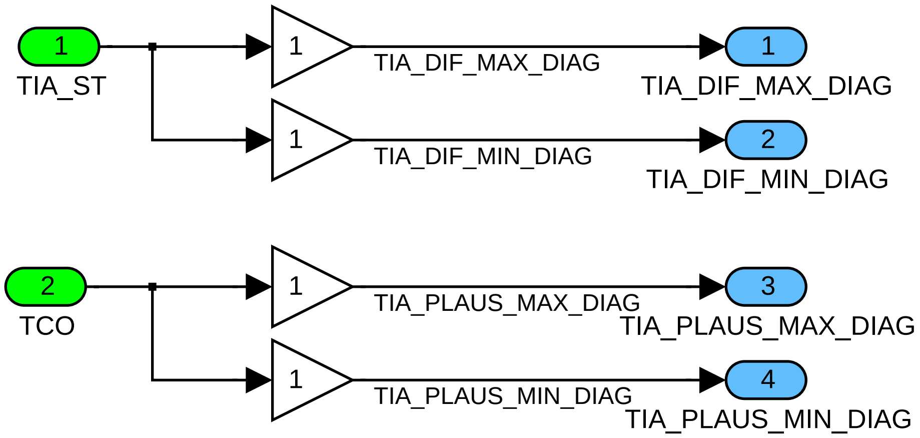


Figure 50 AIRT_MB051/ INI/ RST/ RST

1.8.1.2 Formula section

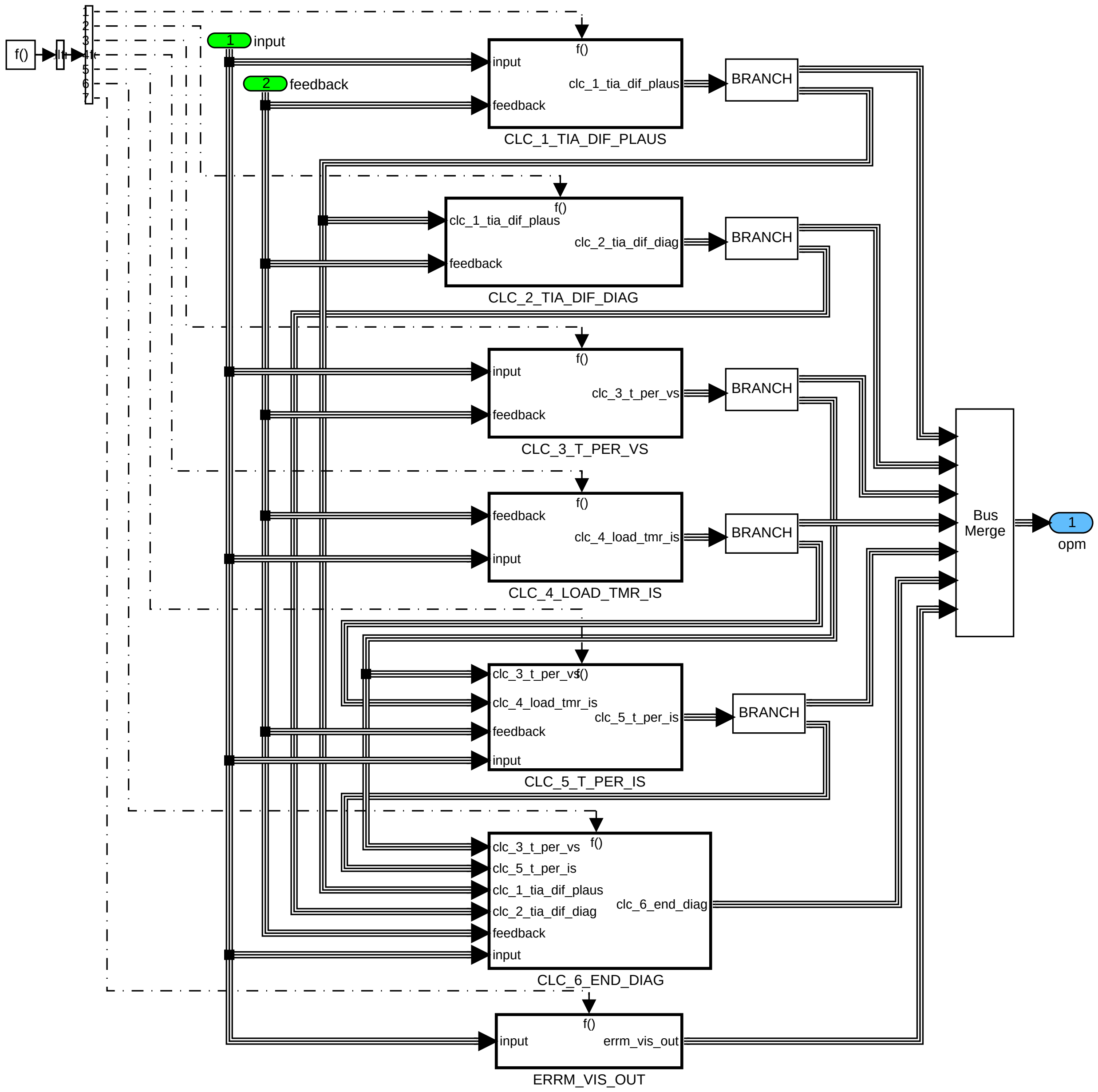


Figure 51 AIRT_MB051/ OPM

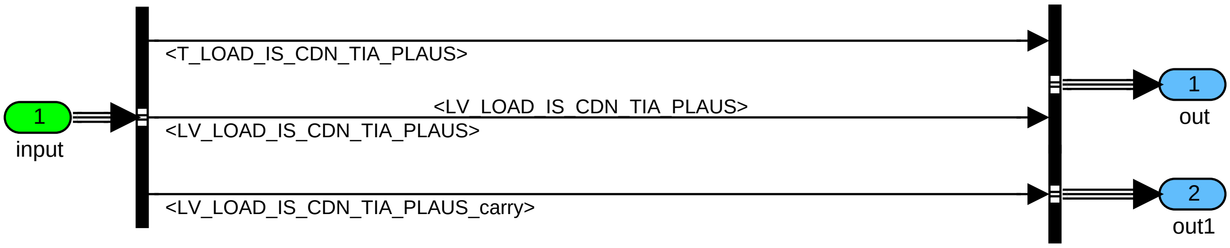


Figure 52 AIRT_MB051/ OPM/ BRANCH

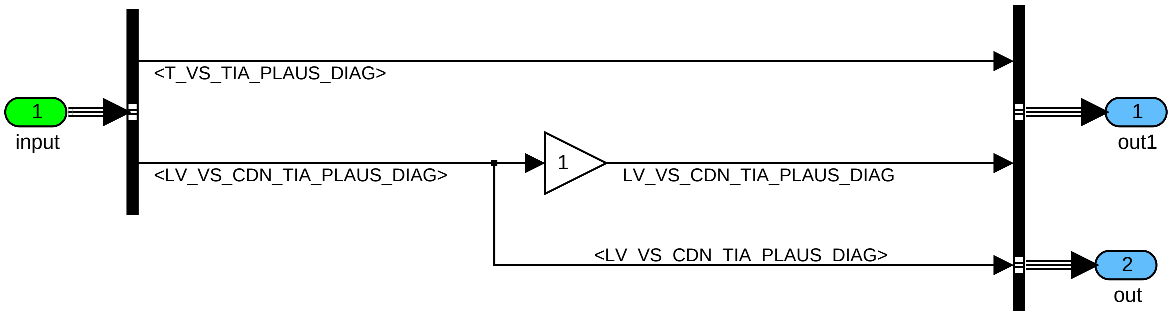


Figure 53 AIRT_MB051/ OPM/ BRANCH1

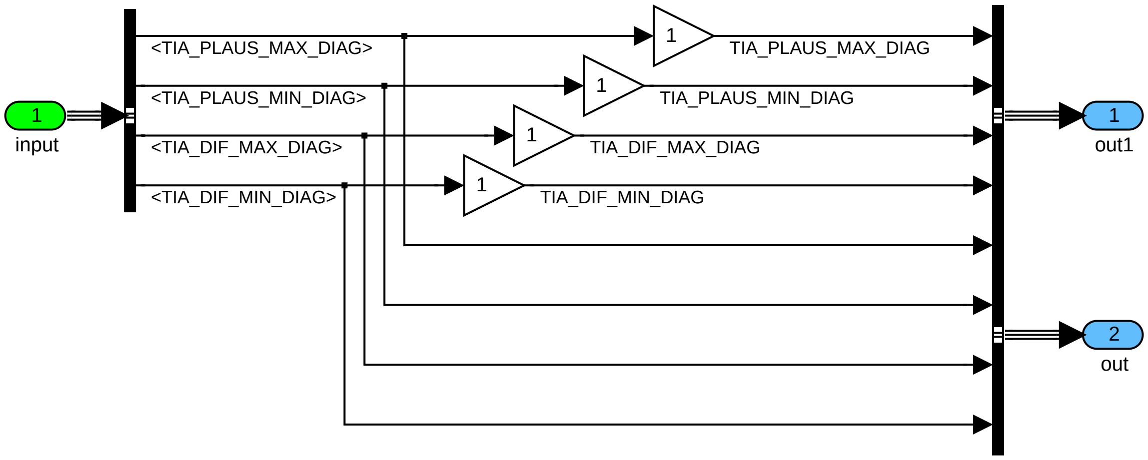


Figure 54 AIRT_MB051/ OPM/ BRANCH2

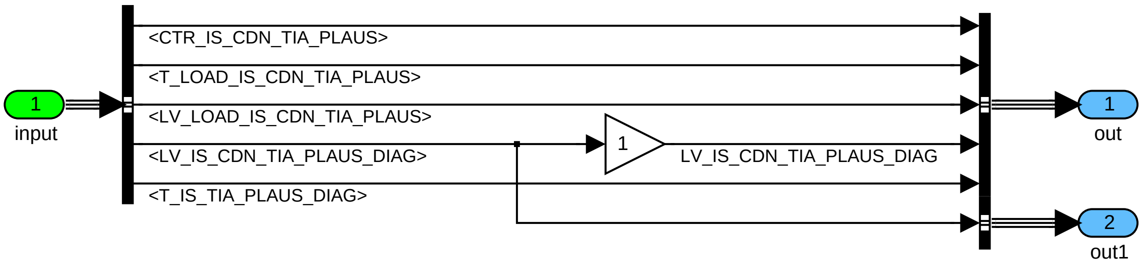


Figure 55 AIRT_MB051/ OPM/ BRANCH3

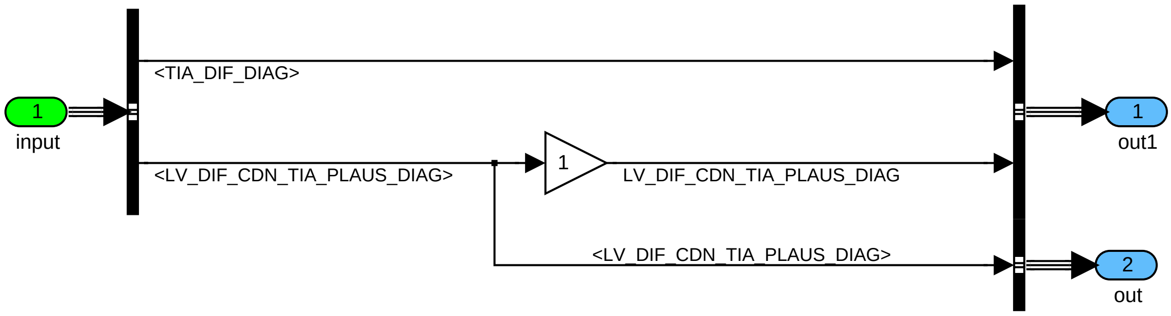


Figure 56 AIRT_MB051/ OPM/ BRANCH4

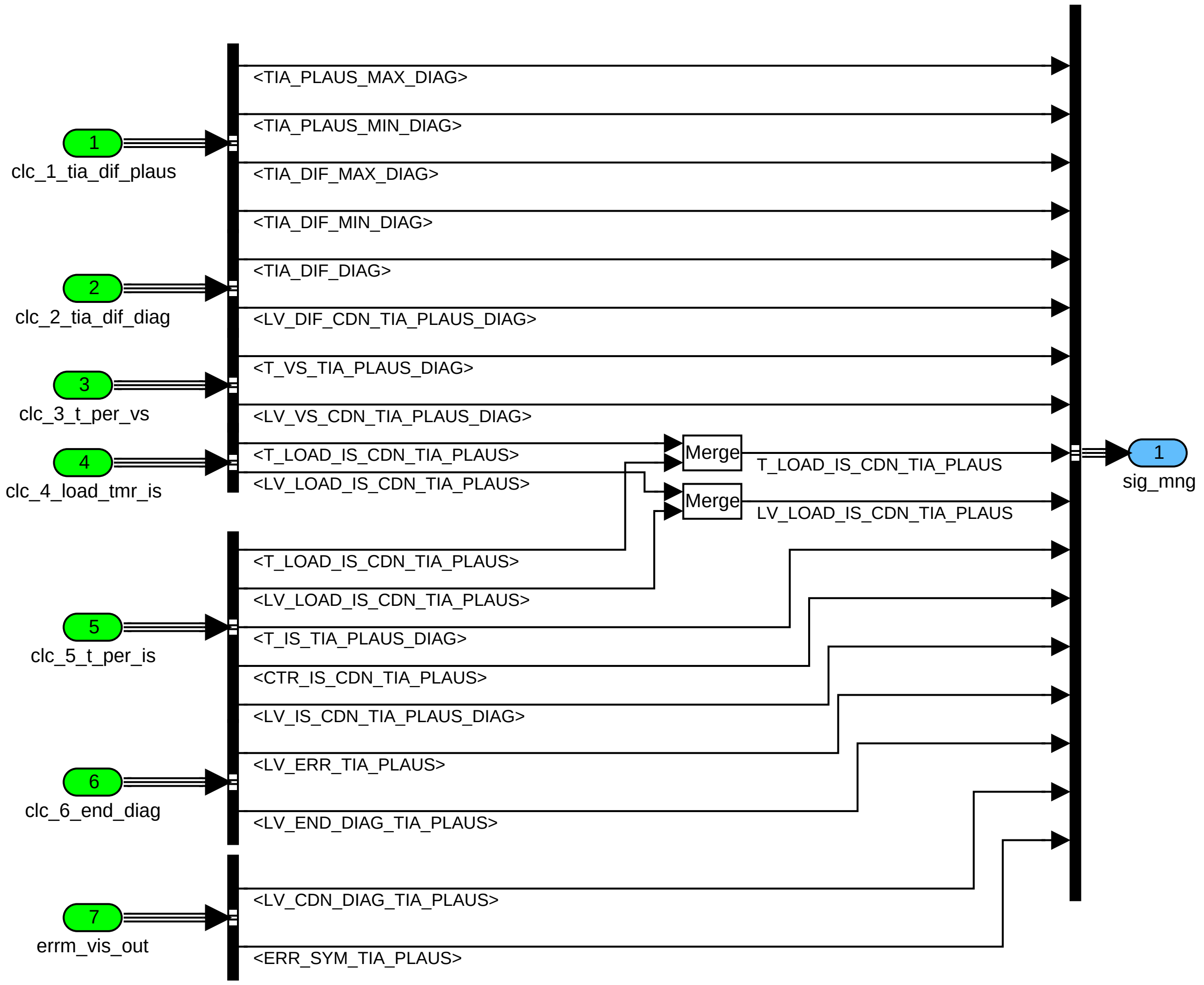


Figure 57 AIRT_MB051/ OPM/ BUS_MERGE

Calculation of TIA_PLAUS_MAX/MIN_DIAG and TIA_DIF_MAX/MIN_DIAG

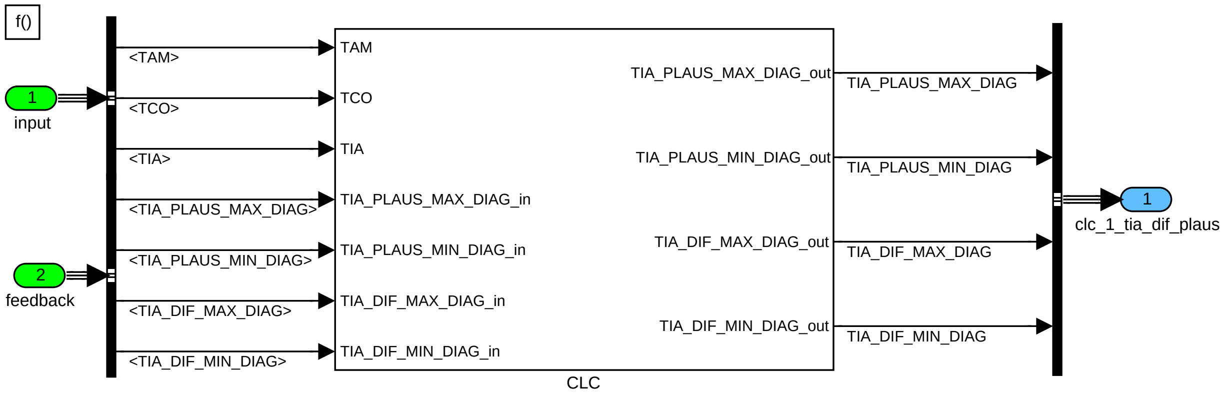


Figure 58 AIRT_MB051/ OPM/ CLC_1_TIA_DIF_PLAUS

Calculation

Calculation of TIA_PLAUS_MAX/MIN_DIAG and TIA_DIF_MAX/MIN_DIAG from the inputs TIA,TAM and TCO and past value of TIA_PLAUS_MAX/MIN_DIAG and TIA_DIF_MAX/MIN_DIAG.

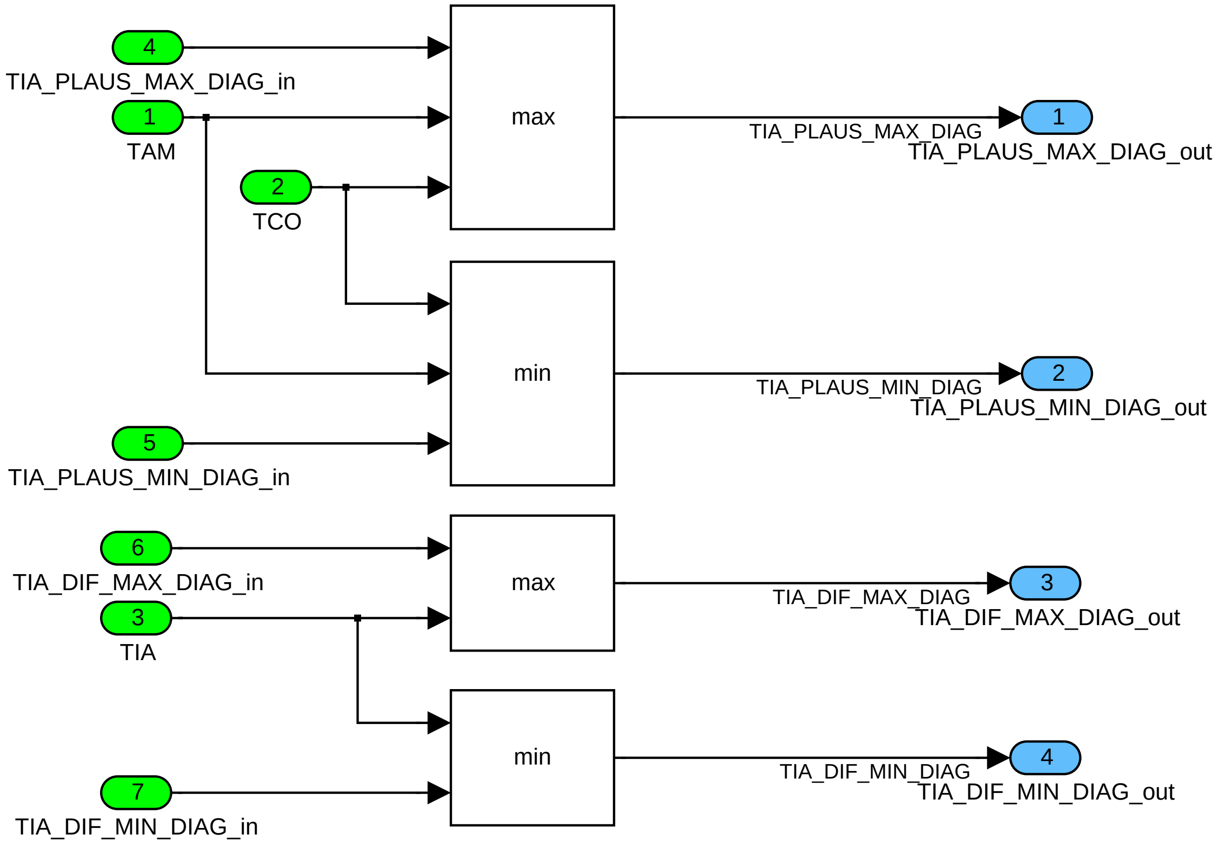


Figure 59 AIRT_MB051/ OPM/ CLC_1_TIA_DIF_PLAUS/ CLC

Calculation of TIA_DIF_DIAG

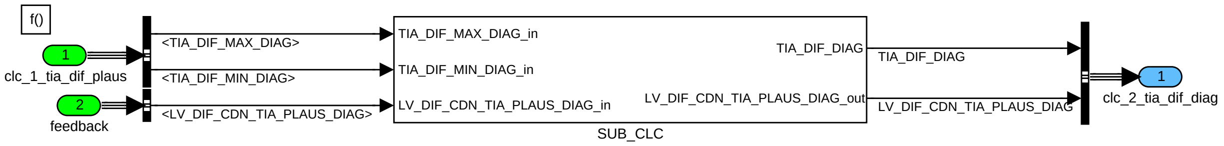


Figure 60 AIRT_MB051/ OPM/ CLC_2_TIA_DIF_DIAG

Calculation of the TIA_DIF_DIAG for stucking detection

Calculation of the TIA_DIF_DIAG for stucking detection. If TIA_DIF_DIAG is greater than C_TIA_DIF_DIAG (*e.g. 3°C*) LV_DIF_CDN_TIA_PLAUS_DIAG is set to 1 (set irreversible)

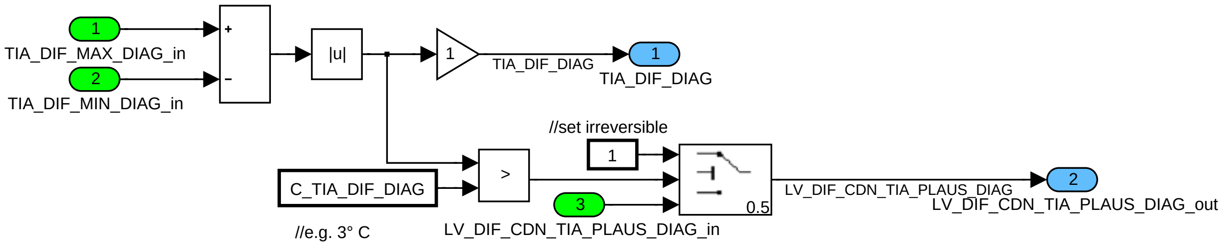


Figure 61 AIRT_MB051/ OPM/ CLC_2_TIA_DIF_DIAG/ SUB_CLC

Calculation at vehicle speed (VS)

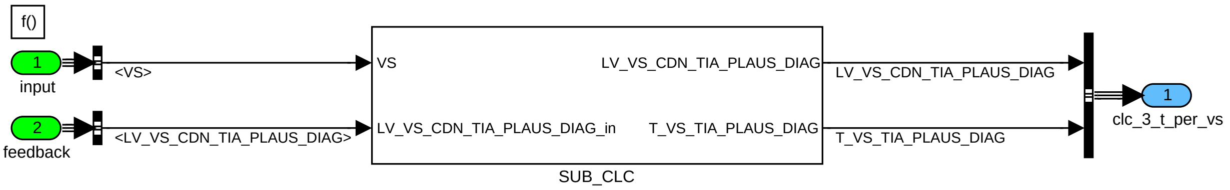


Figure 62 AIRT_MB051/ OPM/ CLC_3_T_PER_VS

Calculation of time period when vehicle is moving(Vehicle with speed)

When VS is greater then C_VS_MIN_TIA_PLAUS_DIAG (e.g. 40km/h) then timer (T_VS_TIA_PLAUS_DIAG) starts incrementing and if T_VS_CDN_TIA_PLAUS_DIAG is greater than threshold for load timer ,C_T_VS_MIN_TIA_PLAUS_DIAG (e.g. 180 s) then LV_VS_CDN_TIA_PLAUS_DIAG is set to 1(set irreversible). If VS is less than minimum VS then timer is reset.

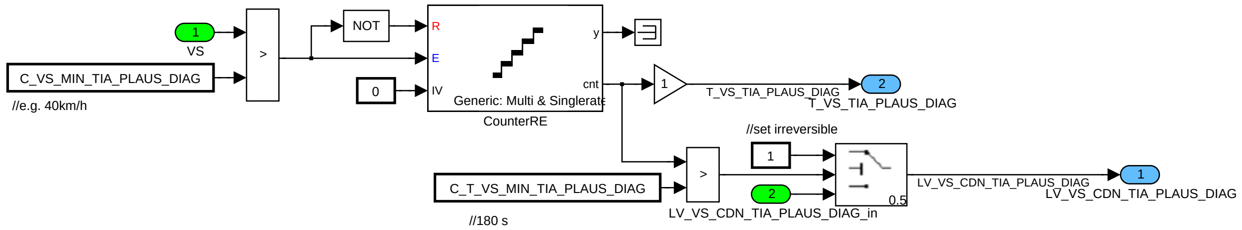


Figure 63 AIRT_MB051/ OPM/ CLC_3_T_PER_VS/ SUB_CLC

Calculation at idle speed condition

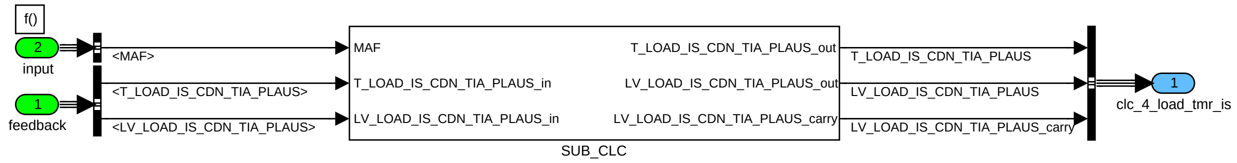


Figure 64 AIRT_MB051/ OPM/ CLC_4_LOAD_TMR_IS

Calculation of load timer to enter in idle speed condition

If MAF is greater than C_MAF_MIN_INT_CDN_TIA_PLAUS (e.g. 200 mg/stk), time period of the vehicle driven with high load (T_LOAD_IS_CDN_TIA_PLAUS) and flag indication that vehicle was driven with high load to enter in idle condition (LV_LOAD_IS_CDN_TIA_PLAUS) is calculated.

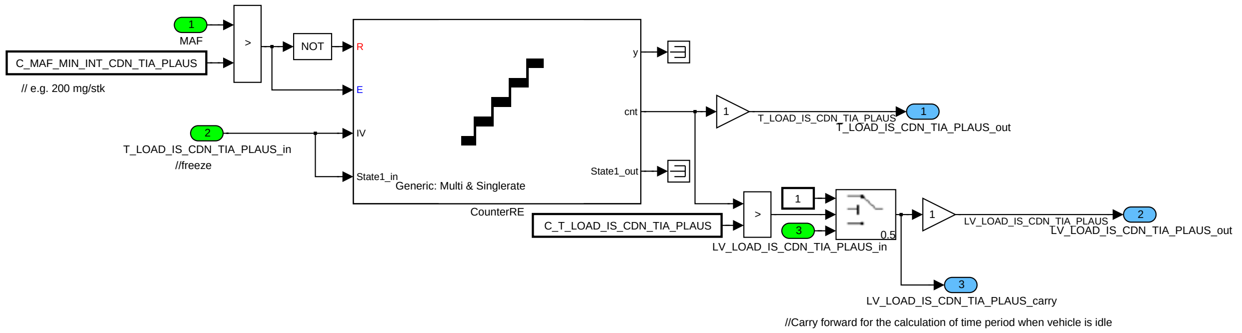


Figure 65 AIRT_MB051/ OPM/ CLC_4_LOAD_TMR_IS/ SUB_CLC

Calculation at vehicle idle

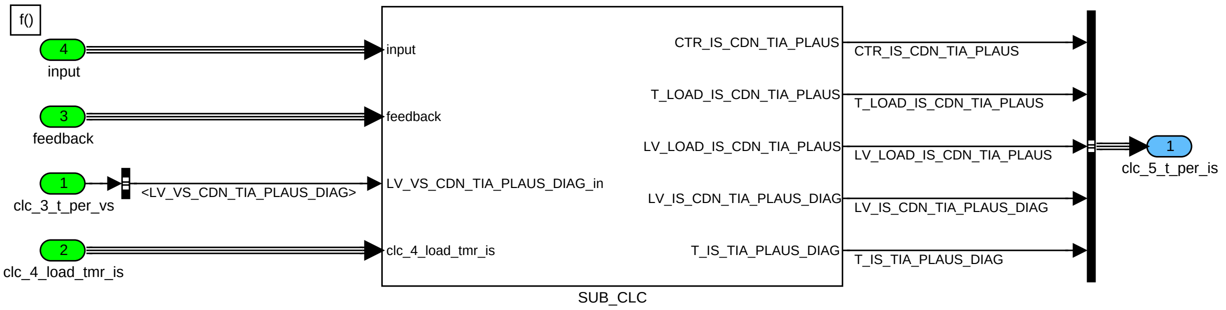


Figure 66 AIRT_MB051/ OPM/ CLC_5_T_PER_IS

Condition check for LV_VS_CDN_TIA_PLAUS_DIAG

Calculation starts when LV_VS_CDN_TIA_PLAUS_DIAG is 1.

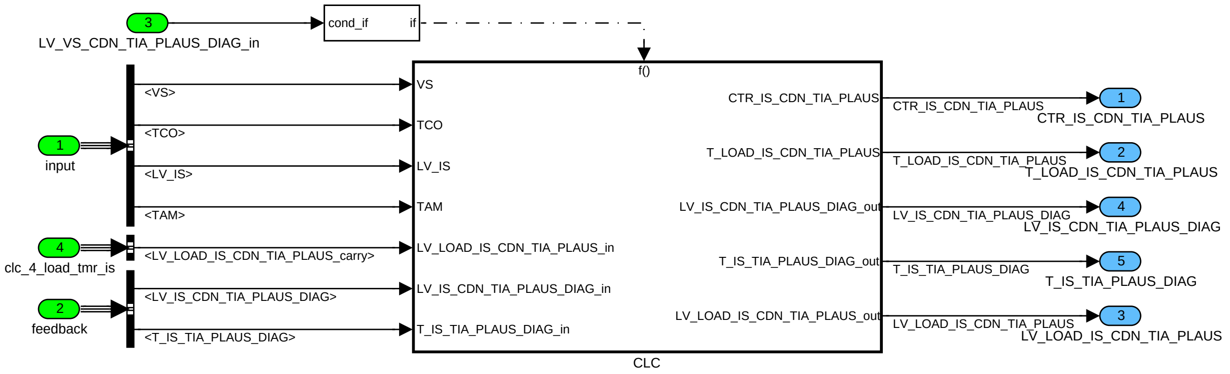


Figure 67 AIRT_MB051/ OPM/ CLC_5_T_PER_IS/ SUB_CLC

Condition check and Calculation of T_IS_TIA_PLAUS_DIAG

If LV_IS is true and VS is greater than maximum VS, C_VS_MAX_IS_CDN_TIA_PLAUS (e.g. 3km/h) and coolant temperature is greater than TIA offset to detect error, C_TCO_MIN_IS_CDN_TIA_PLAUS (e.g. 85 ° C) and LV_LOAD_IS_CDN_TIA_PLAUS is true, then timer, T_IS_TIA_PLAUS_DIAG increment else resets. If the timer value exceeds the threshold value evaluated from ID map (e.g. 20s), then also the timer resets.

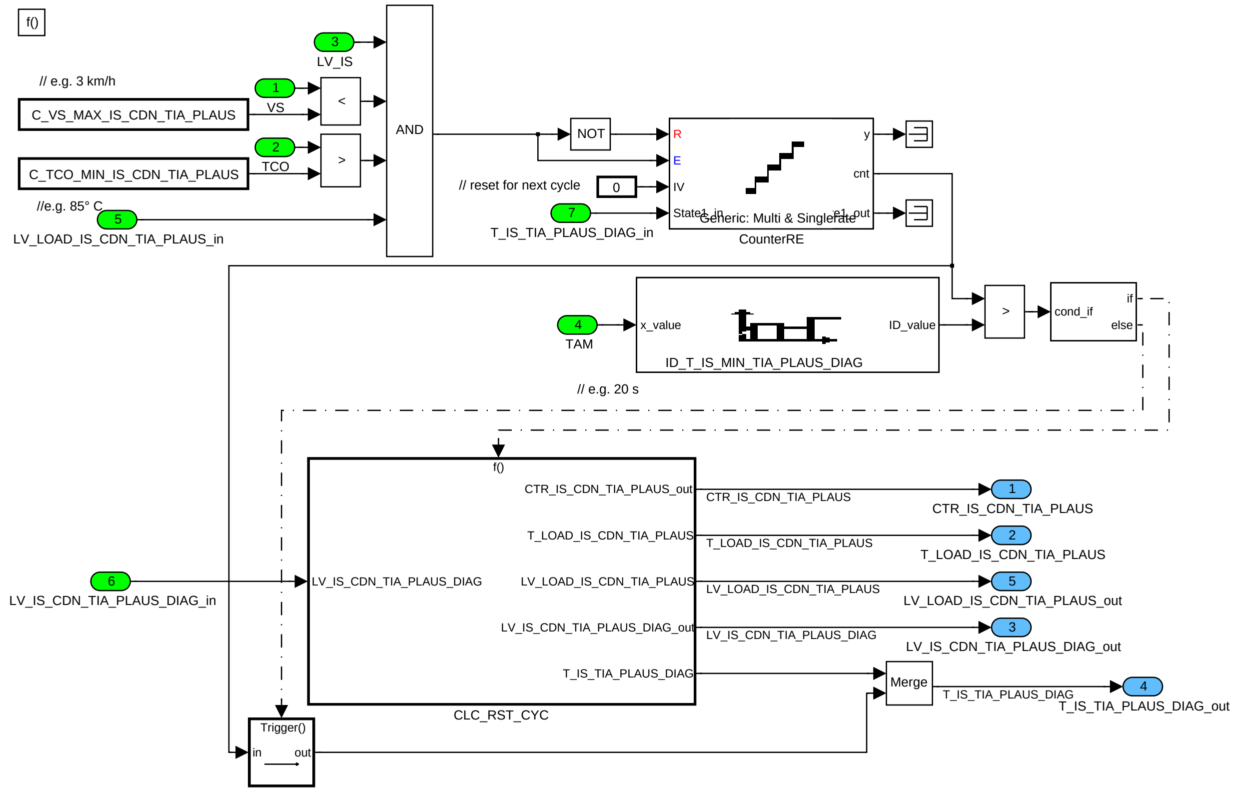


Figure 68 AIRT_MB051/ OPM/ CLC_5_T_PER_IS/ SUB_CLC/ CLC

Calculation of reset for next cycle

T_IS_TIA_PLAUS_DIAG, T_LOAD_IS_CDN_TIA_PLAUS, LV_LOAD_IS_CDN_TIA_PLAUS are reset to 0. If the incremented value of cycle counter for T_IS condition (CTR_IS_CDN_TIA_PLAUS) is greater than number of valid IS condition cycles (C_CTR_IS_CDN_TIA_PLAUS) then LV_IS_CDN_TIA_PLAUS_DIAG is set to 1(set irreversible).

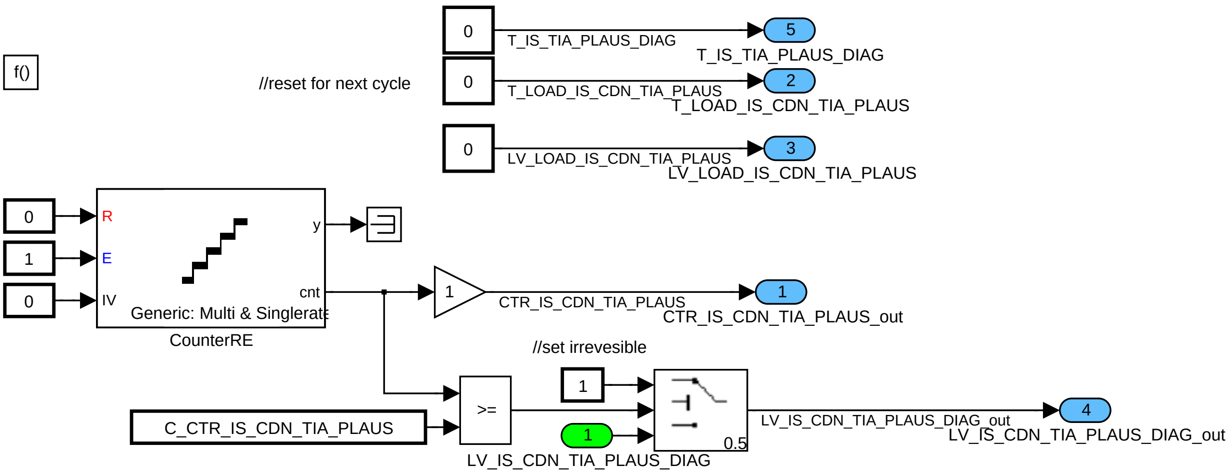


Figure 69 AIRT_MB051/ OPM/ CLC_5_T_PER_IS/ SUB_CLC/ CLC/ CLC_RST_CYC

ERRM diagnosis and interface

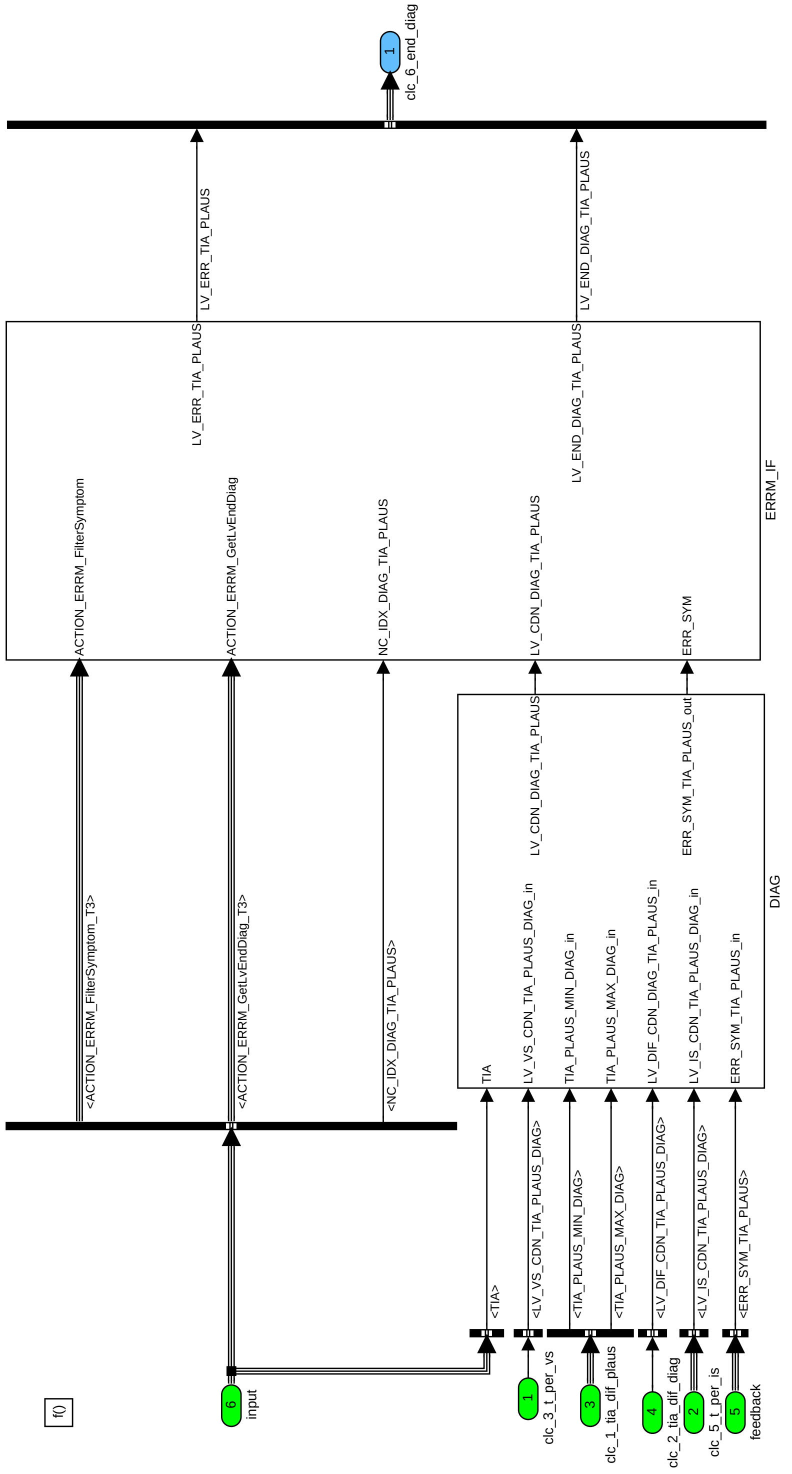


Figure 70 AIRT_MB051/ OPM/ CLC_6_END_DIAG

ERRM diagnosis

Signal to high, signal to low and signal is not plausible error symptoms are calculated here.

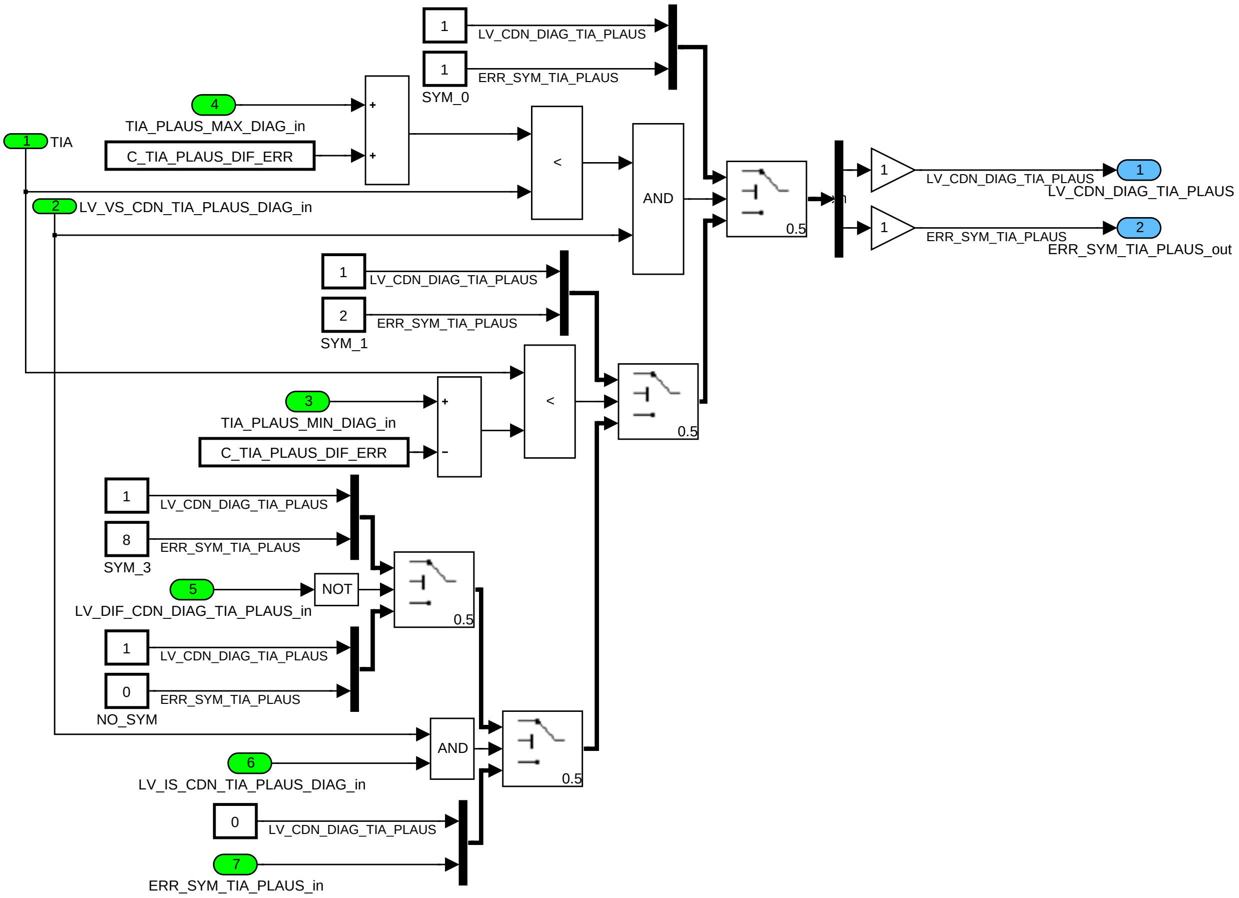


Figure 71 AIRT_MB051/ OPM/ CLC_6_END_DIAG/ DIAG

ERRM interface

LV_ERR_TIA_PLAUS and LV_END_DIAG_TIA_PLAUS are calculated.

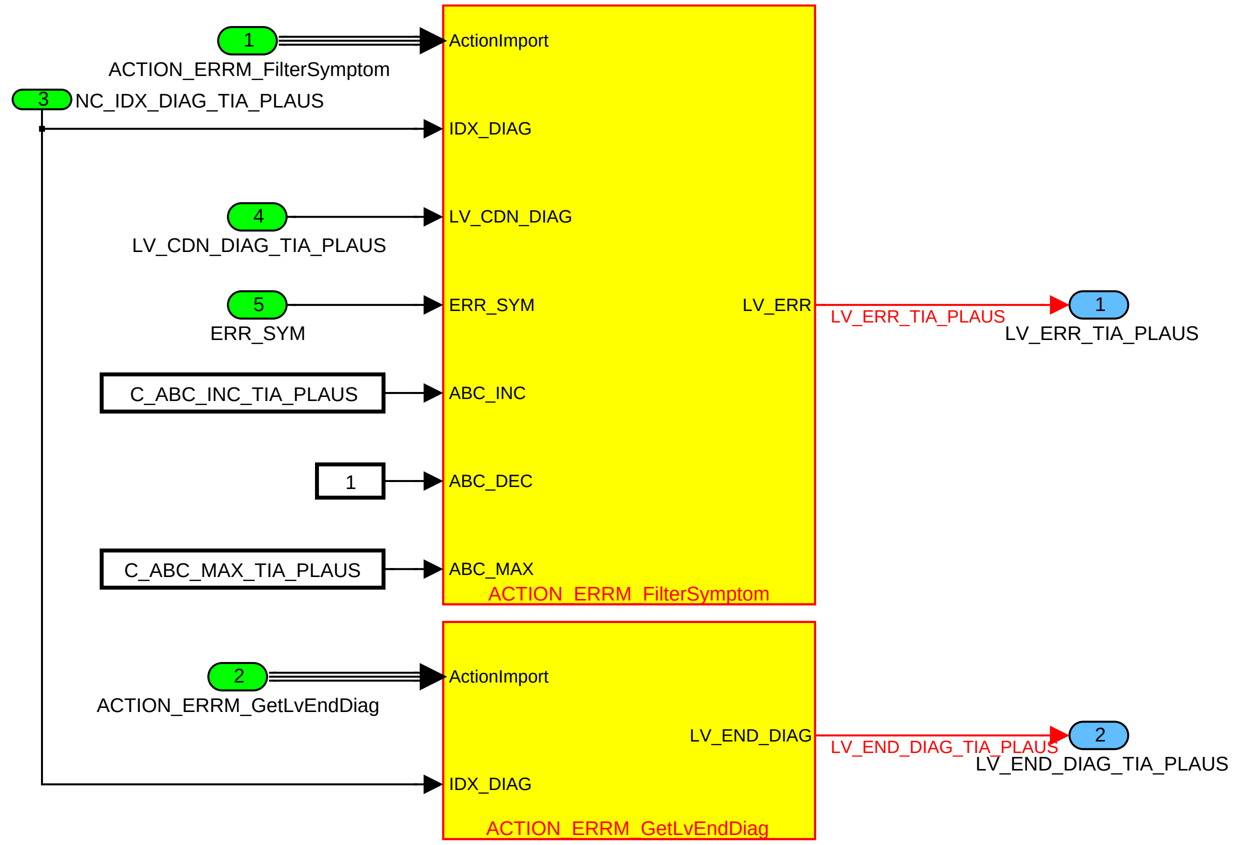


Figure 72 AIRT_MB051/ OPM/ CLC_6_END_DIAG/ ERRM_IF

Visual mode ouputs

LV_CDN_DIAG_TIA_PLAUS and ERR_SYM_TIA_PLAUS are calculated.

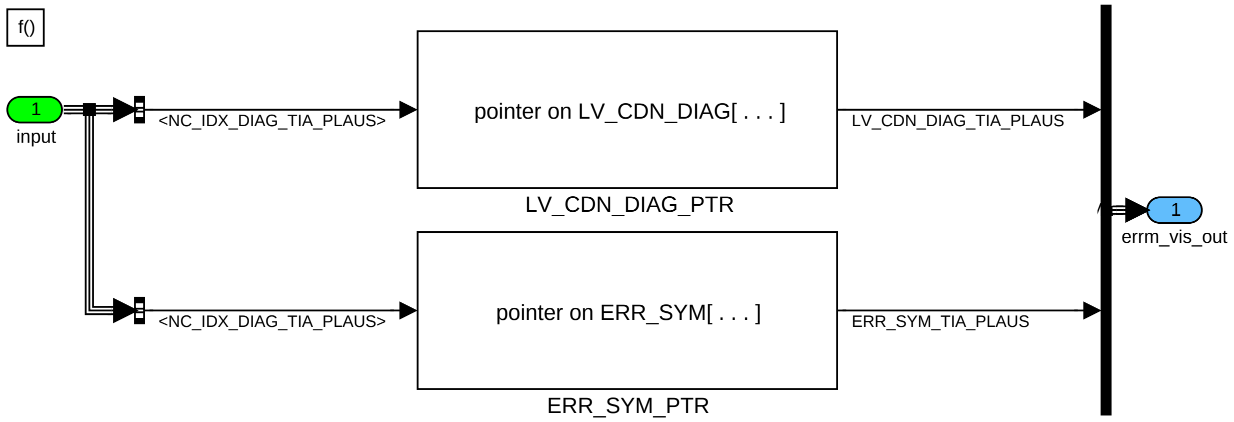


Figure 73 AIRT_MB051/ OPM/ ERRM_VIS_OUT

1.8.1.3 SUBFUNCTION: SIG_MNG

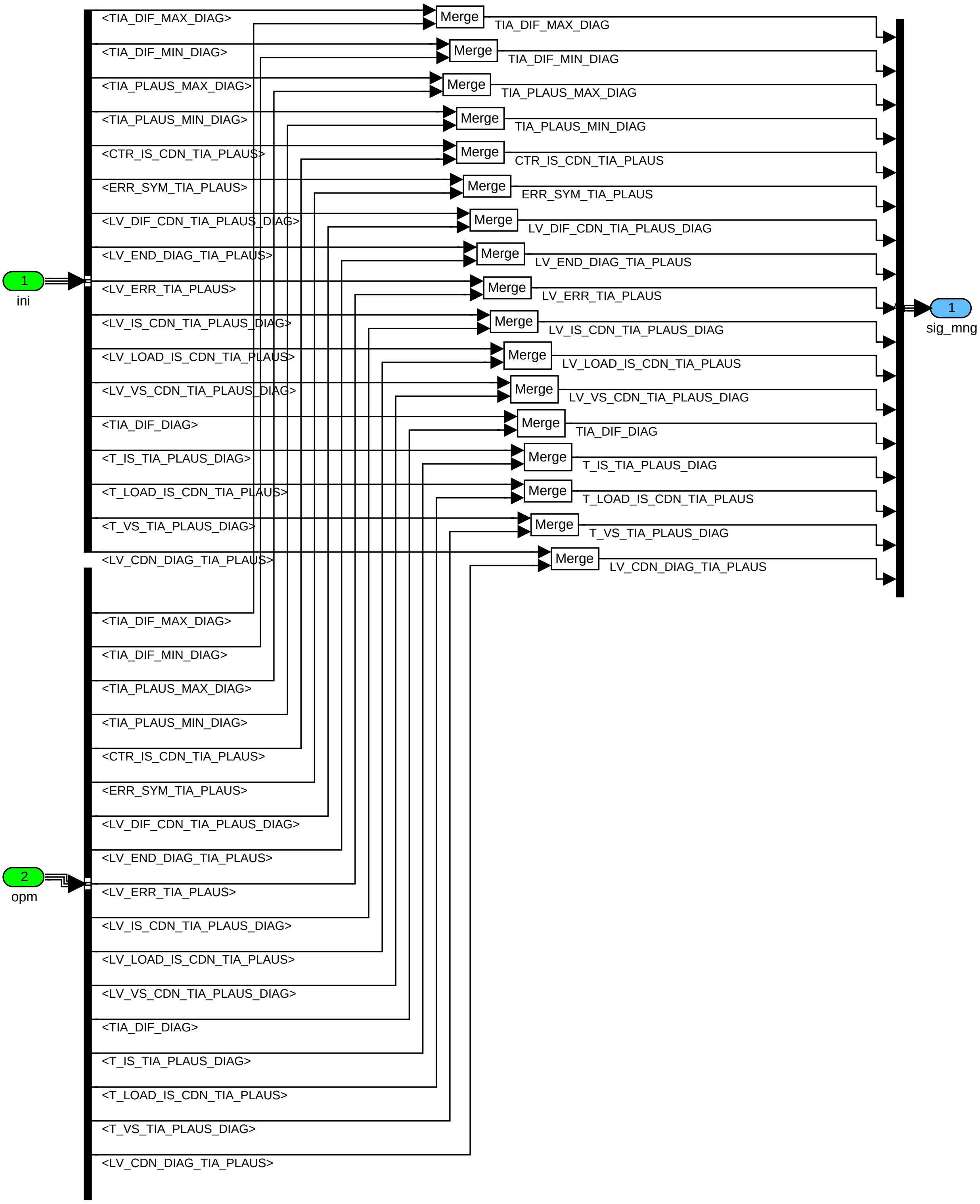


Figure 74 AIRT_MB051/ SIG_MNG

1.9 Intake air temperature sensor diagnosis (Applic. Inc.)

Overview

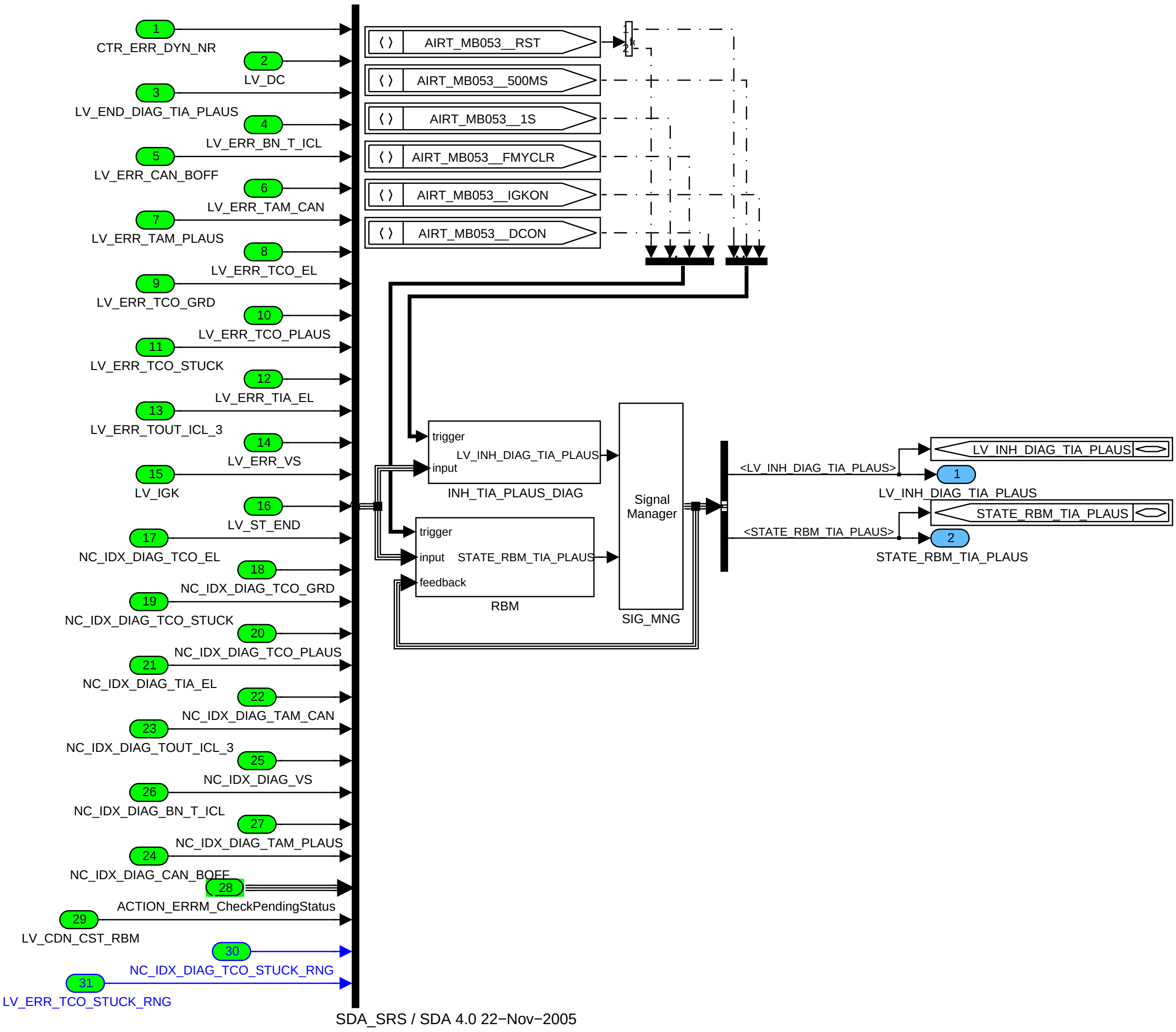


Figure 75 AIRT_MB053

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_TIA_PLAUS	O/V	0...1H	0...1	1	-
Inhibition of TIA PLAUS diagnosis					
STATE_RBM_TIA_PLAUS	O/V	0...7H	0...7	1	-
Interface of TIA_PLAUS monitor with the Rate-Based Monitoring statisticsBit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1)Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1)Bit 2: individual RBM co					

Input data:

CTR_ERR_DYN_NR	LV_DC	LV_END_DIAG_TIA_PLAUS	LV_ERR_BN_T_ICL
LV_ERR_CAN_BOFF	LV_ERR_TAM_CAN	LV_ERR_TAM_PLAUS	LV_ERR_TCO_EL
LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS	LV_ERR_TCO_STUCK	LV_ERR_TIA_EL

LV_ERR_TOUT_ICL_3	LV_ERR_VS	LV_IGK	LV_ST_END
NC_IDX_DIAG_TCO_EL	NC_IDX_DIAG_TCO_GRD	NC_IDX_DIAG_TCO_STUCK	NC_IDX_DIAG_TCO_PLAUS
NC_IDX_DIAG_TIA_EL	NC_IDX_DIAG_TAM_CAN	NC_IDX_DIAG_TOUT_ICL_3	NC_IDX_DIAG_CAN_BOFF
NC_IDX_DIAG_VS	NC_IDX_DIAG_BN_T_ICL	NC_IDX_DIAG_TAM_PLAUS	LV_CDN_CST_RBM
NC_IDX_DIAG_TCO_STUCK_RNG	LV_ERR_TCO_STUCK_RNG		

Import actions:

ACTION_ERRM_CheckPendingStatus(IN <PRM_IDX_DIAG>, OUT <PRM_LV_ERR_PND>)
This API shall be used to verify if a failure stored in the dynamic memory has the pending status or not. When it impossible to determine if the fault is pending or not (failure not store because the dynamic memory is full), this failure should be considered as pending anyway.

1.9.1 Inhibition of TIA_PLAUS diagnosis due to present error

General information:

Inhibiton flag to inhibit the TIA_PLAUS diagnosis because of present error.

Application Condition

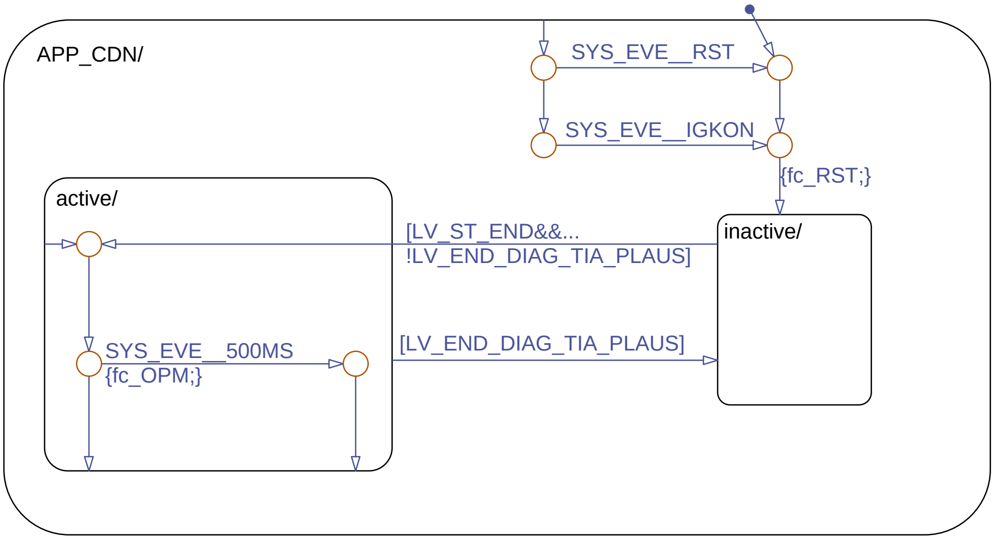


Figure 76 AIRT_MB053/ INH_TIA_PLAUS_DIAG/ APP_CDN/ Chart

Function Description

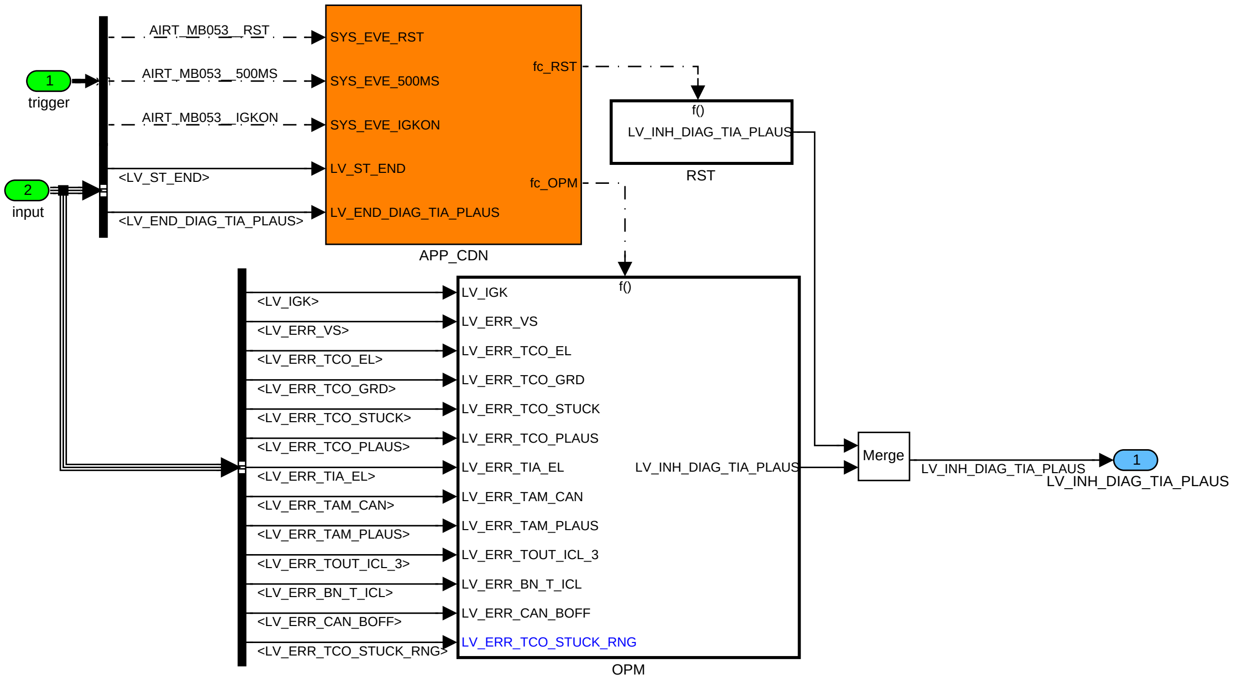


Figure 77 AIRT_MB053/ INH_TIA_PLAUS_DIAG

1.9.1.1 Initialisation

Initialisation of LV_INH_DIAG_TIA_PLAUS

Figure 78 AIRT_MB053/ INH_TIA_PLAUS_DIAG/ RST

1.9.1.2 Formula Section

Ireversible inhibition of diagnosis. LV_INH_DIAG_TIA_PLAUS bit is set to 1, if any one of the inputs is 1.

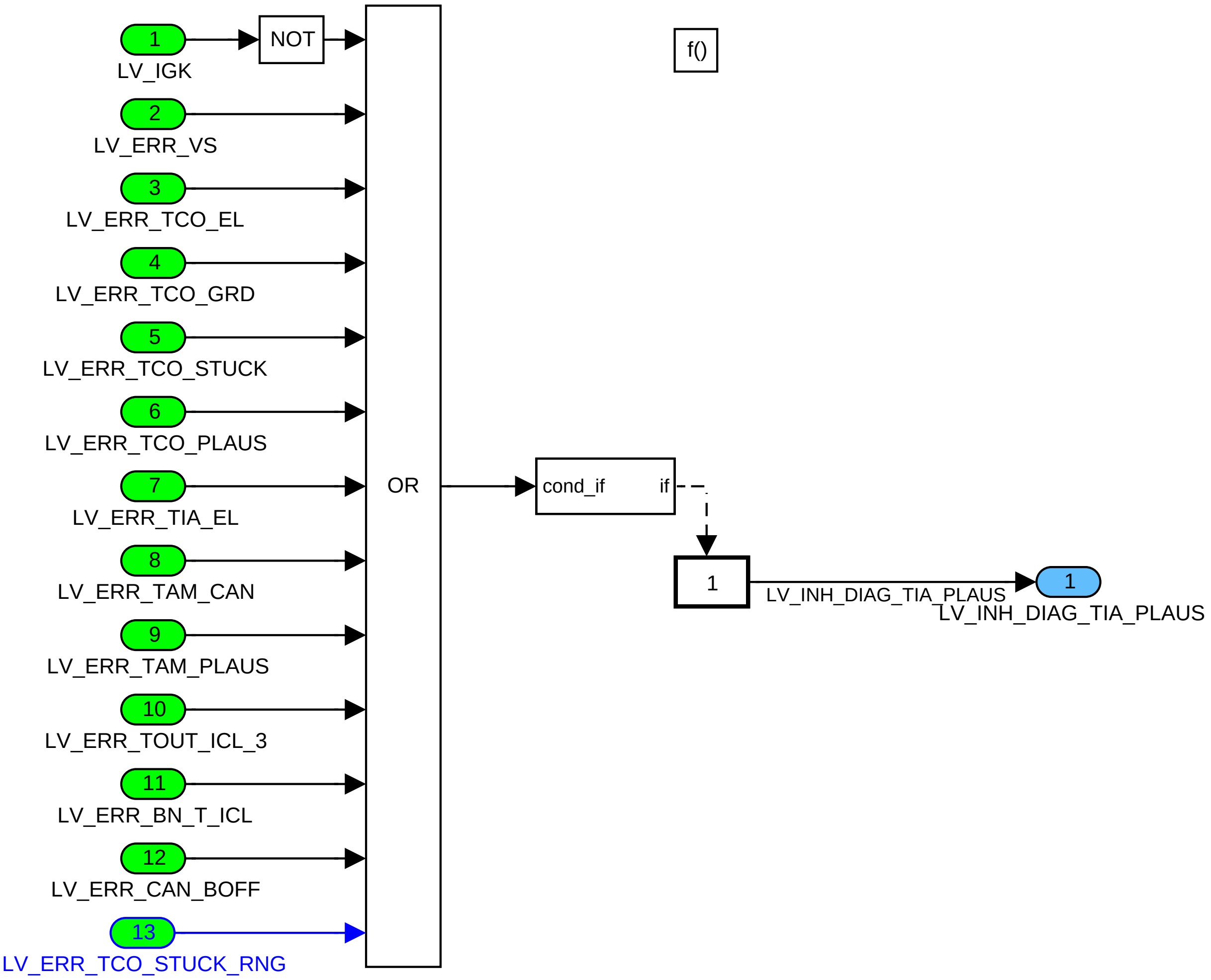


Figure 79 AIRT_MB053/ INH_TIA_PLAUS_DIAG/ OPM

1.9.2 Interface for Rate – Based – Monitoring

General information:

With this module the interface between the TIA_PLAUS monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TIA_PLAUS data.

Within STATE_RBM_TIA_PLAUS, three different information are defined:

Conditions for monitoring are met long enough to detect malfunction (bit 0)

(no intrusive operation, no short trip)

Monitor disabled because of system malfunction (bit 1)

(depending on failure status: pending)

Monitor individual RBM conditions encountered within this DC (bit 2)

(not valid for TIA_PLAUS diagnosis)

Application Condition

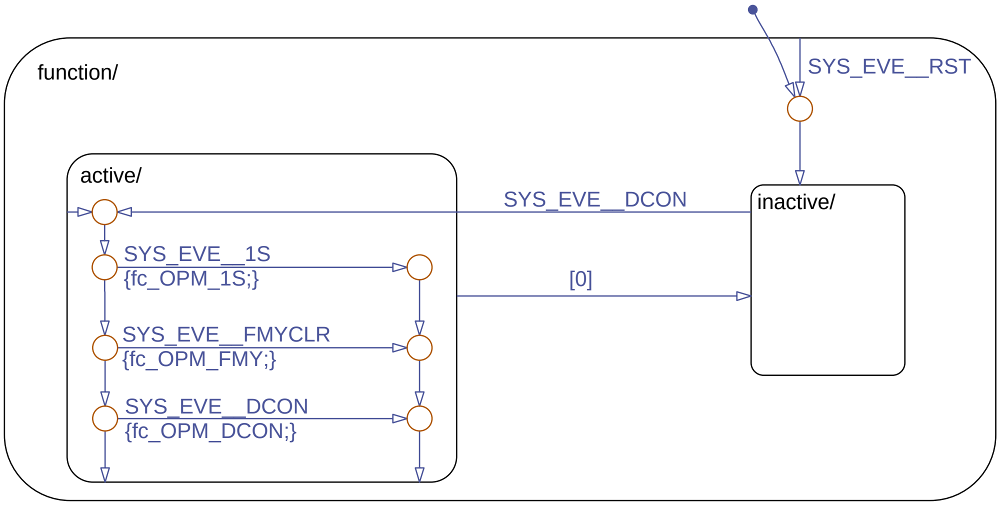


Figure 80 AIRT_MB053/ RBM/ APP_CDN/ Chart

Function Description

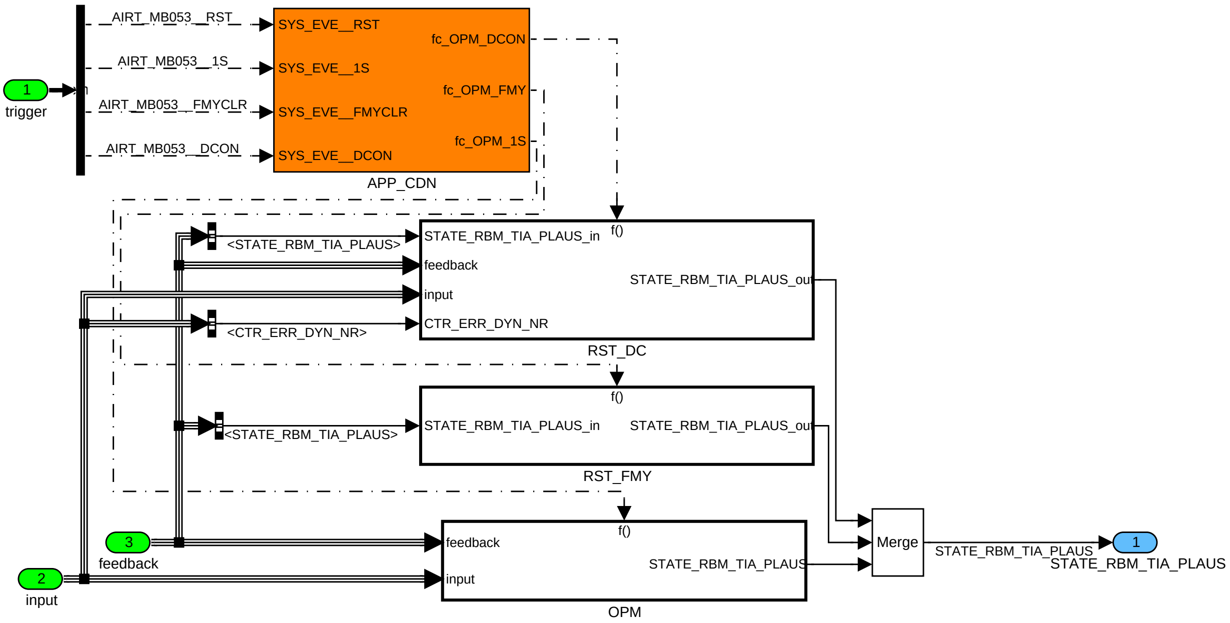


Figure 81 AIRT_MB053/ RBM

1.9.2.1 Formula Section at reset on driving cycle on

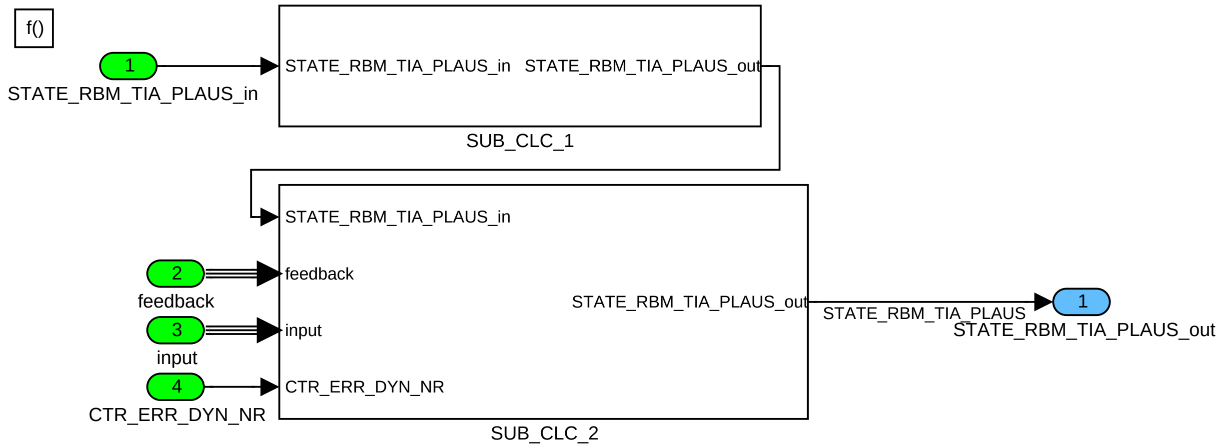


Figure 82 AIRT_MB053/ RBM/ RST_DC

Initialisation at DCON

Bits 0, 1 and 2 of STATE_RBM_TIA_PLAUS are reset to 0, whenever there is transition of LV_DC from 0 to 1.

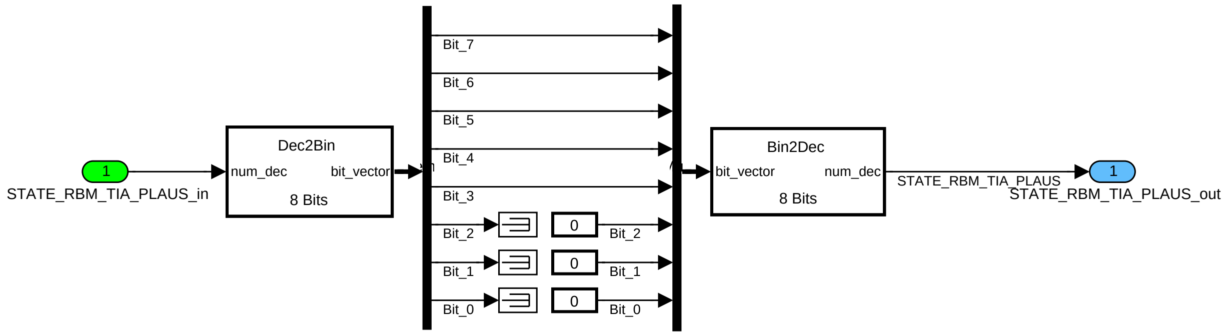


Figure 83 AIRT_MB053/ RBM/ RST_DC/ SUB_CLC_1

Condition check for CTR_ERR_DYN_NR

Activation condition at LV_DC transition from 0 to 1 and the dynamic failure memory is not empty(CTR_ERR_DYN_NR is not equal to 0)

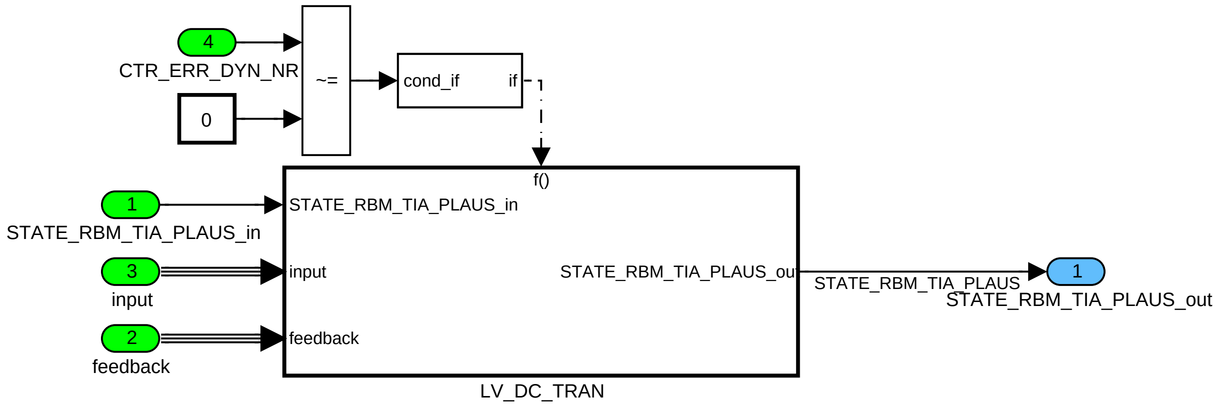


Figure 84 AIRT_MB053/ RBM/ RST_DC/ SUB_CLC_2

Condition check for bit 1 of STATE_RBM_TIA_PLAUS

While bit 1 of STATE_RBM_TIA_PLAUS is 0 the pending status for each of the failure will be checked.

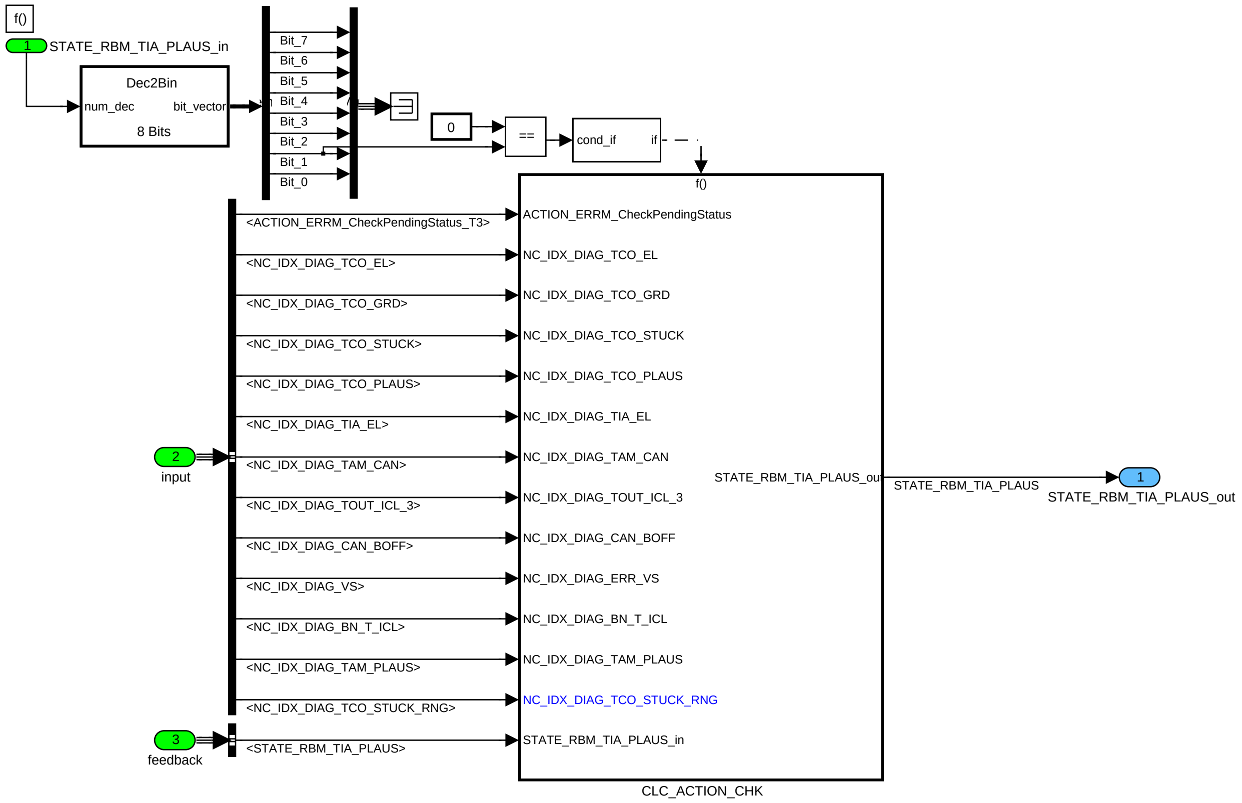


Figure 85 AIRT_MB053/ RBM/ RST_DC/ SUB_CLC_2/ LV_DC_TRAN

Pending status check

Pending status for each of the failure is checked using Check Pending Status Action call from ERRM.

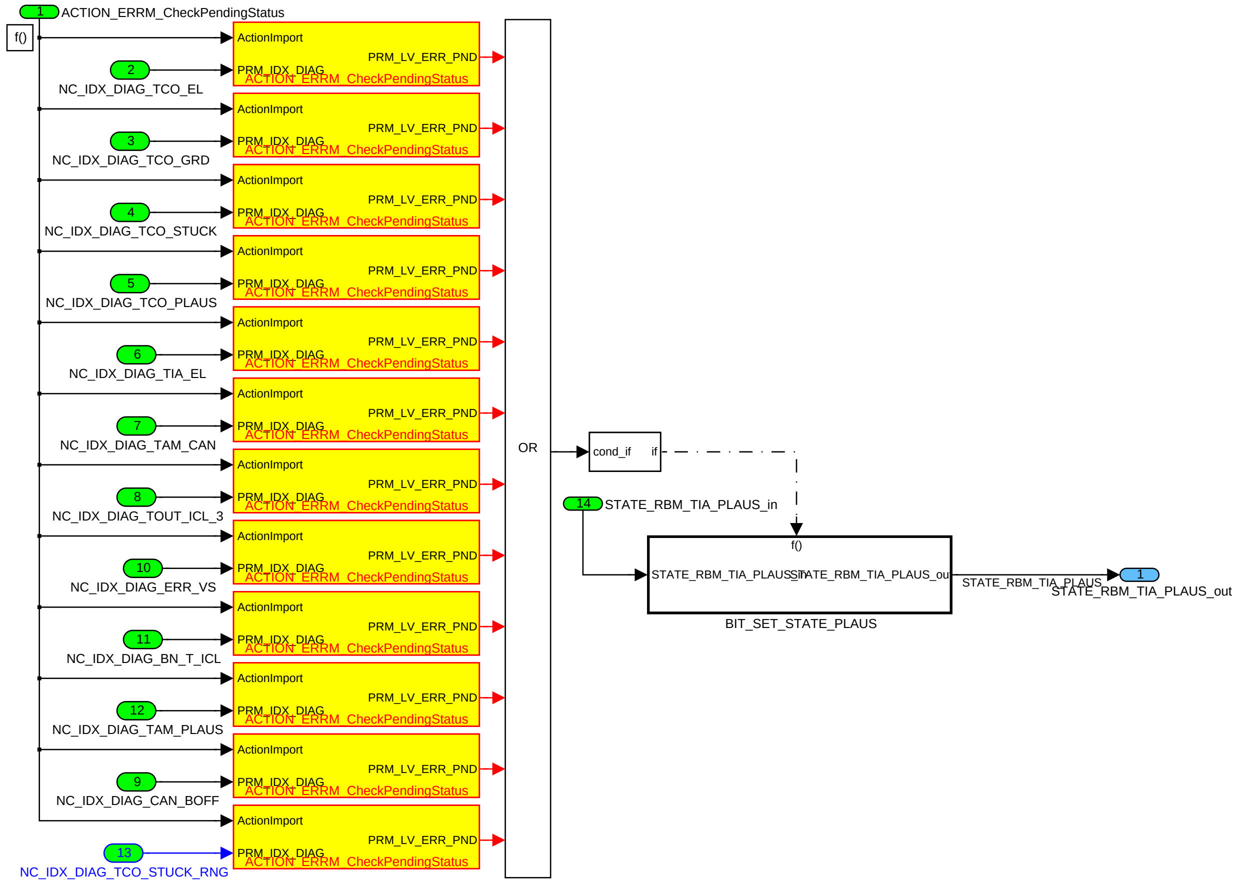


Figure 86 AIRT_MB053/ RBM/ RST_DC/ SUB_CLC_2/ LV_DC_TRAN/ CLC_ACTION_CHK

Bit 1 setting for STATE_RBM_TIA_PLAUS

If any of the failures has pending status the bit 1 of STATE_RBM_TIA_PLAUS will be set to 1

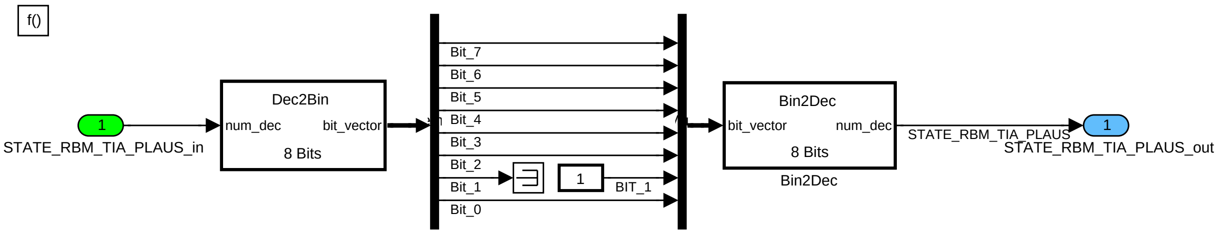


Figure 87 AIRT_MB053/ RBM/ RST_DC/ SUB_CLC_2/ LV_DC_TRAN/ CLC_ACTION_CHK/ BIT_SET_STATE_PLAUS

1.9.2.2 Initialisation at Failure Memory Reset

Bit 1 of STATE_RBM_TIA_PLAUS is reset to 0 whenever there is Failure Memory reset event.

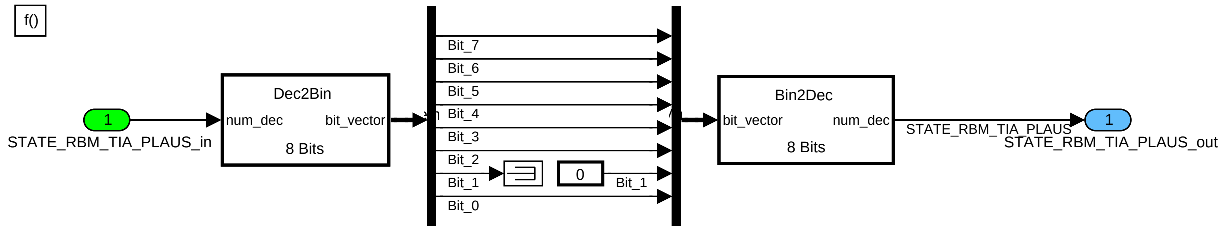


Figure 88 AIRT_MB053/ RBM/ RST_FMY

1.9.2.3 Formula section for 1 second recurrence

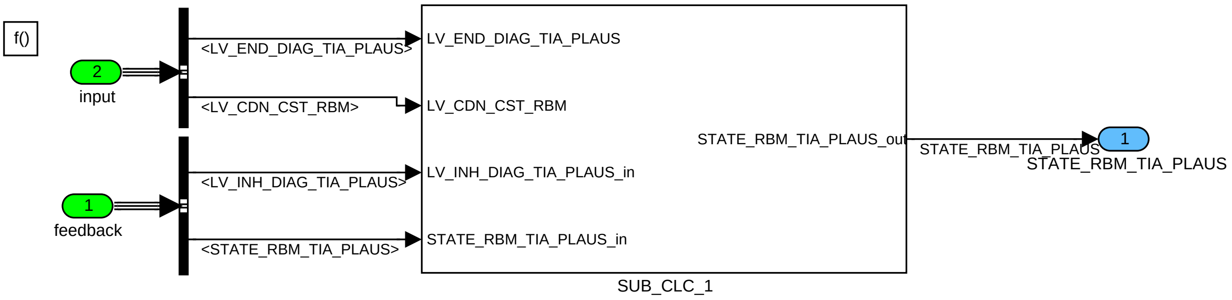


Figure 89 AIRT_MB053/ RBM/ OPM

Bit set operation

Calculation of status of bits 0, 1 and 2 of STATE_RBM_TIA_PLAUS at 1 second recurrence.

If bit 0 of STATE_RBM_TIA_PLAUS is 0 and LV_END_DIAG_TIA_PLAUS is 1 then bit 0 of STATE_RBM_TIA_PLAUS is set to 1.

If bit 1 of STATE_RBM_TIA_PLAUS is 0 and LV_INH_DIAG_TIA_PLAUS is 1 then bit 1 of STATE_RBM_TIA_PLAUS is set to 1

If bit 2 of STATE_RBM_TIA_PLAUS is 0 and LV_CDN_CST_RBM is 1 then bit 2 of STATE_RBM_TIA_PLAUS is set to 1

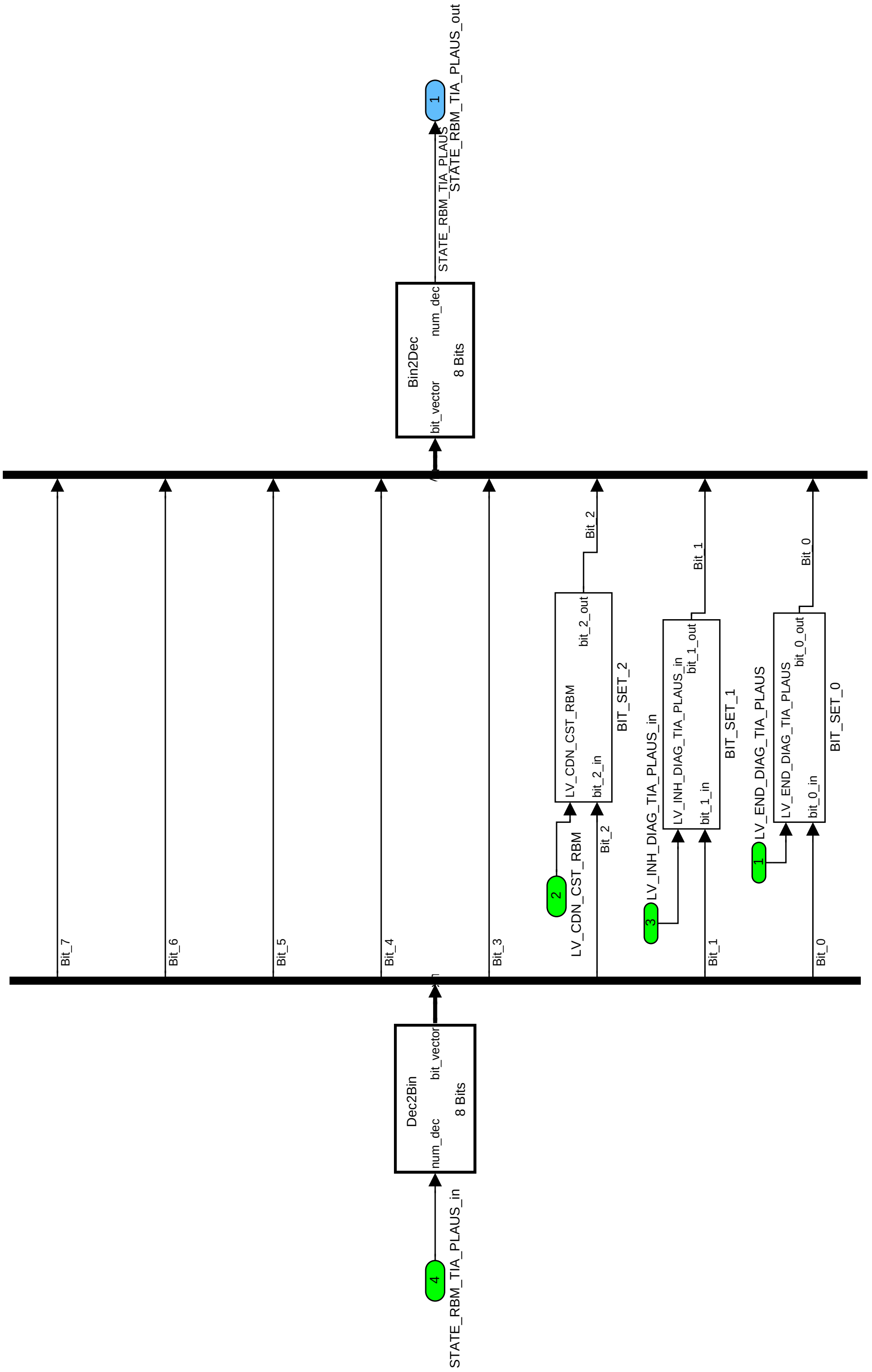


Figure 90 AIRT_MB053/ RBM/ OPM/ SUB_CLC_1

BIT_SET_0

Bit 0 of STATE_RBM_TIA_PLAUS is set to 1, if bit 0 of STATE_RBM_TIA_PLAUS is 0 and LV_END_DIAG_TIA_PLAUS is 1

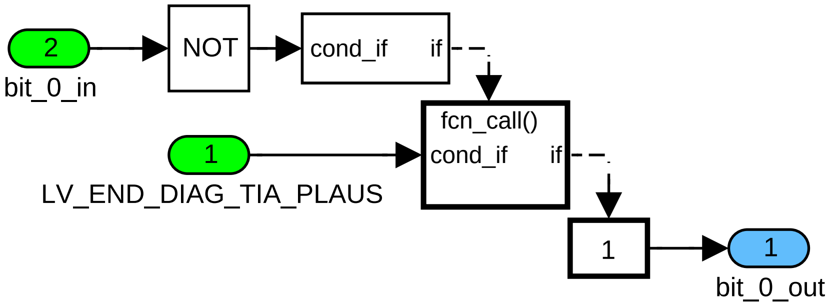


Figure 91 AIRT_MB053/ RBM/ OPM/ SUB_CLC_1/ BIT_SET_0

BIT_SET_1

Bit 1 of STATE_RBM_TIA_PLAUS is set to 1, if bit 0 of STATE_RBM_TIA_PLAUS is 0 and LV_INH_DIAG_TIA_PLAUS is 1

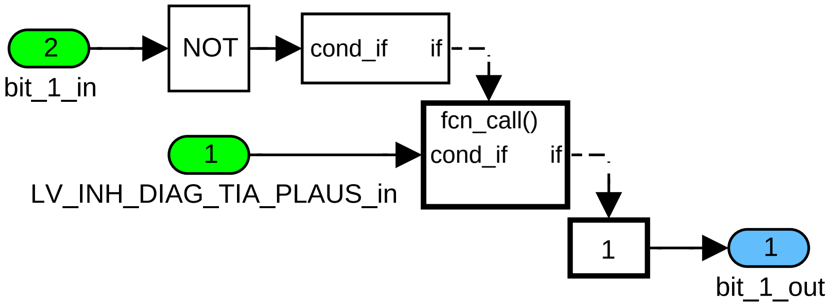


Figure 92 AIRT_MB053/ RBM/ OPM/ SUB_CLC_1/ BIT_SET_1

BIT_SET_2

If bit 2 of STATE_RBM_TIA_PLAUS is 0 and LV_CDN_CST_RBM is 1 then bit 2 of STATE_RBM_TIA_PLAUS is set to 1.

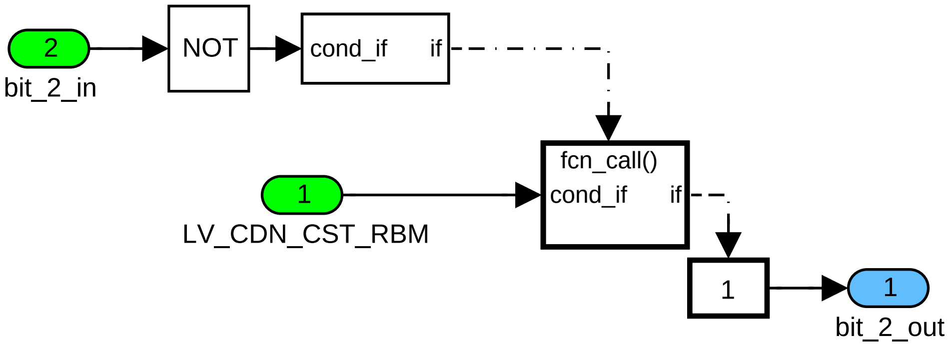


Figure 93 AIRT_MB053/ RBM/ OPM/ SUB_CLC_1/ BIT_SET_2

1.10 Ambient temperature sensor diagnosis (TAM)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TAM	V/O	0...1H	0...1	1	[-]
Present failure on TIA sensor(s) which might affect temperature TAM					
LV_ERR_TAM_CAN	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on TAM sensor					
LV_CDN_DIAG_TAM_CAN	V	0...1H	0...1	1	[-]
Diagnosis condition TAM_CAN					
ERR_SYM_TAM_CAN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom TAM_CAN diagnosis					
LV_END_DIAG_TAM_CAN	V	0...1H	0...1	1	[-]
End of Diagnosis cycle TAM_CAN					
LV_ERR_TAM_PLAUS	V/O	0...1H	0...1	1	[-]
TAM sensor signal is not plausible					
LV_CDN_DIAG_TAM_PLAUS	V	0...1H	0...1	1	[-]
Diagnosis condition TAM_PLAUS					
ERR_SYM_TAM_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom TAM_PLAUS					
LV_END_DIAG_TAM_PLAUS	V	0...1H	0...1	1	[-]
End of diagnosis cycle TAM_PLAUS					
TAM_DIF_PLAUS	V	0...FEH	0...190.5	0.75	[°C]
TAM difference to detect a moving signal					
TAM_DIF_DIAG	V	0...FEH	0...190.5	0.75	[°C]
TAM difference to detect a TAM error					
T_TAM_DIF_DIAG	V	0...FFH	0...25.5	0.1	[s]
Timer for anti-bounce the temperature difference					
T_VS_MIN_VLD_TAM_ERR	V	0...FFFFH	0...6553.5	0.1	[s]
Timer of vehicle speed period					
LV_DET_ERR_TAM_PLAUS	V	0...1H	0...1	1	[-]
TAM-plaus error is detected but not validated					
LV_VS_CDN_TAM_ERR	V	0...1H	0...1	1	[-]
Bit information of status vehicle was driven with speed to validate the diagnosis result					
TAM_DIF_MAX_DIAG	V	0...FEH	-48...142.5	0.75	[°C]
Maximum reached TAM in diagnosis cycle					
TAM_DIF_MIN_DIAG	V	0...FEH	-48...142.5	0.75	[°C]
Minimum reached TAM in diagnosis cycle					
LV_END_RBM_TAM_PLAUS	V/O	0...1H	0...1	1	[-]
End of diagnosis cycle TAM_PLAUS for RBM statistics					
LV_CDN_RBM_TAM_PLAUS	V	0...1H	0...1	1	[-]
Diagnosis condition for RBM statistics TAM_PLAUS					
LV_CNL_DIAG_TAM_PLAUS	V	0...1H	0...1	1	[-]
Irreversible stop of diagnosis TAM_PLAUS					

Input data:

LV_ST_END	TAM	TAM_ST	TAM_STOP
LV_INH_DIAG_TAM_PLAUS	T_ES_CUS	TCO_2_ST	T_AST
TIA	TCO_2	VS	LV_DIAG_CDN_TAM_PLAUS
LV_ERR_TOUT_ICL_3	LV_ERR_CAN_BOFF	LV_IGK	LV_ERR_BN_T_ICL
LV_TAM_CAN_ERR	TCO_ST	TCO	LV_VAR_TCO_2
STATE_ERR_SYM_TAM_CAN	LV_ERR_TOUT_ICL_7	LV_STATE_ERR_CAN_TOUT	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TAM_CAN	1	0...FFH	0...255	1	[-]
Anti-bounce increment TAM_CAN diagnosis					
C_ABC_MAX_TAM_CAN	1	1...FFH	1...255	1	[-]
Anti-bounce maximum TAM_CAN diagnosis					
C_TAM_DIF_PLAUS_MIN	1	0...FEH	0...190.5	0.75	[°C]
Minimum TAM difference to detect a moving signal during engine run					
C_TAM_DIF_ST_END	1	0...FEH	0...190.5	0.75	[°C]
Minimum TAM difference to detect a moving signal after start					
C_T_AST_MAX_TAM_DIAG	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum time after start to deactivate the diagnosis condition					
C_T_ES_MIN_TAM_DIAG	1	0...FFFFH	0...65535	1	[min]
Minimum engine stop time to activate the diagnosis condition					
C_TCO_2_ST_MIN_TAM_DIAG	1	0...FEH	-48...142.5	0.75	[°C]
Minimum exhaust engine cooling temperature to activate the diagnosis condition					
C_TAM_MIN_TAM_DIAG	1	0...FEH	-48...142.5	0.75	[°C]
Minimum ambient temperature to activate the diagnosis condition					
C_TAM_MAX_TAM_DIAG	1	0...FEH	-48...142.5	0.75	[°C]
Maximum ambient temperature to activate the diagnosis condition					
C_TAM_DIF_MAX_ERR	1	0...FEH	0...190.5	0.75	[°C]
Temperature difference to detect a TAM error					
C_T_TAM_DIF_MAX_ERR	1	0...FFH	0...25.5	0.1	[s]
Anti ¿ bounce time to detect a TAM error					
C_VS_MIN_VLD_TAM_ERR	1	0...FFH	0...255	1	[km/h]
Minimum vehicle speed to count time period for error validation					
C_T_VS_MIN_VLD_TAM_ERR	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time perid of vehicle with speed					
C_TCO_2_TIA_DIF_MAX	1	0...FEH	0...190.5	0.75	[°C]
Maximum deviation between TCO_2 and TIA to start TAM_PLAUS diagnosis					
C_TCO_ST_MIN_TAM_DIAG	1	0...FEH	-48...142.5	0.75	[°C]
Minimum engine cooling temperature to activate the diagnosis condition					
C_TCO_TIA_DIF_MAX	1	0...FEH	0...190.5	0.75	[°C]
Maximum deviation between TCO and TIA to start TAM_PLAUS diagnosis					

1.10.1 Global TAM system error (LV_ERR_TAM)

General information:

The global error flag LV_ERR_TAM can be used for inhibition. It is set if a CAN or a plausibility TIA error occure.

Application conditions:

Initialisation: 0 at reset

Recurrence: 100ms

Activation: LV_IGK = 1

Formula section:

```
If      LV_ERR_TAM_CAN = 1 or LV_ERR_TAM_PLAUS = 1
Then    LV_ERR_TAM = 1
Else    LV_ERR_TAM = 0
Endif
```

1.10.2 CAN based TAM – signal diagnosis (LV_ERR_TAM_CAN)

General information:

The purpose of the diagnosis shall be to detect electrical faults as defined by OBD I requirements. The input signals are from CAN.

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
│ │ │ └─> short circuit to vbatt or open line      (= SYM_0)
│ │ └───> short circuit to ground                  (= SYM_1)
│ └─────> signal error                             (= SYM_2)
└───────> -                                        (= SYM_3)
```

Application conditions:

Initialization: according ABC configuration **STD_INI**
(ABC datas are initialized at transition LV_IGK 0->1)

Recurrence: 100 ms

Activation: LV_IGK = 1

Formula section:

```
LV_CDN_DIAG_TAM_CAN = 1
ERR_SYM_TAM_CAN = NO_SYM
If      LV_ERR_CAN_BOFF         = 0   and
        LV_ERR_TOUT_ICL_7       = 0   and
        LV_STATE_ERR_CAN_TOUT = 0     and
        STATE_ERR_SYM_TAM_CAN = SCP        // error symptom get from CAN
Then    ERR_SYM_TAM_CAN = SYM_0
Elseif  LV_ERR_CAN_BOFF         = 0   and
        LV_ERR_TOUT_ICL_7       = 0   and
```

```
         LV_STATE_ERR_CAN_TOUT = 0  and
         STATE_ERR_SYM_TAM_CAN = SCG        // error symptom get from CAN
Then     ERR_SYM_TAM_CAN = SYM_1
Elseif   LV_TAM_CAN_ERR      = 1   and
         LV_ERR_TOUT_ICL_3   = 0   and
         LV_ERR_BN_T_ICL     = 0   and
         LV_ERR_CAN_BOFF     = 0
Then     ERR_SYM_TAM_CAN = SYM_2
Endif
```

Calculation of present error and end of diagnosis:

LV_ERR_TAM_CAN and LV_END_DIAG_TAM_CAN are calculated by error management

1.10.3 Ambient temperature sensor plausibility check (LV_ERR_TAM_PLAUS)

FUNCTION DESCRIPTION:

General information:

This diagnosis is performed in order to detect a stucking or not plausible TAM signal which cannot be detected by electrical range diagnosis.

Description:

Diagnosis condition:

The diagnosis should only be activated:

- after a long engine stop time, e.g. > 10h in order to get similar temperatures of TAM / TIA / TCO_2 or TCO (depending on variant coding LV_VAR_TCO_2)
- if TAM is in a "critical" range e.g. fade out temperature of other OBD2 diagnosis
- just after start to avoid warming-up effects, e.g. within 5 s
- not in cold environment, because it's not requested by law

If these conditions are not fulfilled the diagnosis is switched off if C_T_AST_MAX_TAM_DIAG is elapsed after start.

Error detection

The TAM_PLAUS error is detected if the temperature-difference TAM_DIF_DIAG is over the threshold C_TAM_DIF_MAX_ERR for the anti-bounce time C_T_TAM_DIF_MAX_ERR.

End of diagnosis cycle with positive result

The positive diagnosis result is based on the monitoring of the alteration of TAM at and can be reached even if a error is detected or the validation phase is running.

- Just after start, comparing the start and stop temperature
- During warm up phase

If a alteration is detected the diagnosis is finished.

Error validation

The validation is only performed if all el. diagnosis for TIA / TCO_2 or TCO and finished (LV_DIAG_CDN_TAM_PLAUS = 1) and the vehicle was driven with a certain vehicle speed (C_VS_MIN_VLD_TAM_ERR for C_T_VS_MIN_VLD_TAM_ERR).

If both conditions are true and a error was detected then the error LV_ERR_TAM_PLAUS is set for this driving cylce and the diagnosis is switched off.

Application conditions:

Initialisation: 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory
TAM_DIF_MAX / MIN_DIAG = TAM_ST
at LV_ST_END = 1 **or** reset **or** at clearing error memory

Recurrence: 100ms

Activation: LV_ST_END = 1

Deactivation: LV_END_DIAG_TAM_PLAUS = 1 **or**
LV_INH_DIAG_TAM_PLAUS = 1 **or**
LV_CNL_DIAG_TAM_PLAUS = 1
(set LV_CDN_DIAG_TAM_PLAUS = 0 at deactivation)

Formula section:

Calculation of TAM_DIF_PLAUS to end the diagnosis

```
TAM_DIF_MAX_DIAG = MAX [ TAM, TAM_DIF_MAX_DIAG(n-1)]
TAM_DIF_MIN_DIAG = MIN [ TAM, TAM_DIF_MIN_DIAG(n-1)]
TAM_DIF_PLAUS = ABS (TAM_DIF_MAX_DIAG - TAM_DIF_MIN_DIAG)
If      TAM_DIF_PLAUS > C_TAM_DIF_PLAUS_MIN     or      //e.g. 3°C
        ABS ( TAM_STOP – TAM_ST ) > C_TAM_DIF_ST_END    //e.g. 5°C
Then    LV_CDN_DIAG_TAM_PLAUS = 1
        LV_END_DIAG_TAM_PLAUS = 1
Endif
```

Calculation of diagnosis condition / error detection

```
If(1) T_AST < C_T_AST_MAX_TAM_DIAG        and     //e.g. 5s
    LV_DET_ERR_TAM_PLAUS = 0
Then(1)
    If(2) T_ES_CUS > C_T_ES_MIN_TAM_DIAG        and     //e.g. >10h
       [ TAM  >  C_TAM_MAX_TAM_DIAG             or      //e.g.  >35°C or  < -10°C
       TAM  <  C_TAM_MIN_TAM_DIAG ]
   Then(2)          // conditions to calculate TAM_DIF_DIAG fulfilled
       If(3) LV_VAR_TCO_2 = 1
       Then(3)
          If(4)
          TCO_2_ST > C_TCO_2_ST_MIN_TAM_DIAG      and     //e.g. > -7°C
          ABS[ TCO_2 – TIA ] < C_TCO_2_TIA_DIF_MAX  and   //e.g. < 10°C
          Then(4)
          LV_CDN_RBM_TAM_PLAUS = 1
```

```
            TAM_DIF_DIAG = ABS [ (TIA + TCO_2) * 0.5 – TAM ]
            Else(4)
            TAM_DIF_DIAG = TAM_DIF_DIAG          //freeze
            Endif(4)
        Else(3)
            If(4a)
            TCO_ST > C_TCO_ST_MIN_TAM_DIAG       and   //e.g. > -7°C
            ABS[ TCO – TIA ] < C_TCO_TIA_DIF_MAX and   //e.g. < 10°C
            Then(4a)
            LV_CDN_RBM_TAM_PLAUS = 1
            TAM_DIF_DIAG = ABS [ (TIA + TCO) * 0.5 – TAM ]
            Else(4a)
            TAM_DIF_DIAG = TAM_DIF_DIAG          //freeze
            Endif(4a)
        Endif(3)
    Endif(2)
Else(1)     If    LV_DET_ERR_TAM_PLAUS = 0   and
                  LV_CDN_RBM_TAM_PLAUS = 0
            Then  LV_CNL_DIAG_TAM_PLAUS = 1         //deactivation of diagnosis
            Else  stop diagnosis condition calculation, waiting for LV_END_DIAG_TAM_PLAUS
            Endif
Endif(1)
```

Calculation of debounce-timer for error detection

```
If    TAM_DIF_DIAG > C_TAM_DIF_MAX_ERR
Then        T_TAM_DIF_DIAG ++
    If      T_TAM_DIF_DIAG > C_T_TAM_DIF_MAX_ERR  //e.g.1s
    Then    LV_DET_ERR_TAM_PLAUS = 1
    Endif
Else        T_TAM_DIF_DIAG - -
Endif
```

Calculation of minimum time period of vehicle with speed

```
If          VS > C_VS_MIN_VLD_TAM_ERR                          //e.g. 50km/h
Then        T_VS_MIN_VLD_TAM_ERR ++
    If            T_VS_MIN_VLD_TAM_ERR > C_T_VS_MIN_VLD_TAM_ERR
    Then          LV_VS_CDN_TAM_ERR = 1
    Endif
Else        T_VS_MIN_VLD_TAM_ERR = T_VS_MIN_VLD_TAM_ERR        //freeze
```

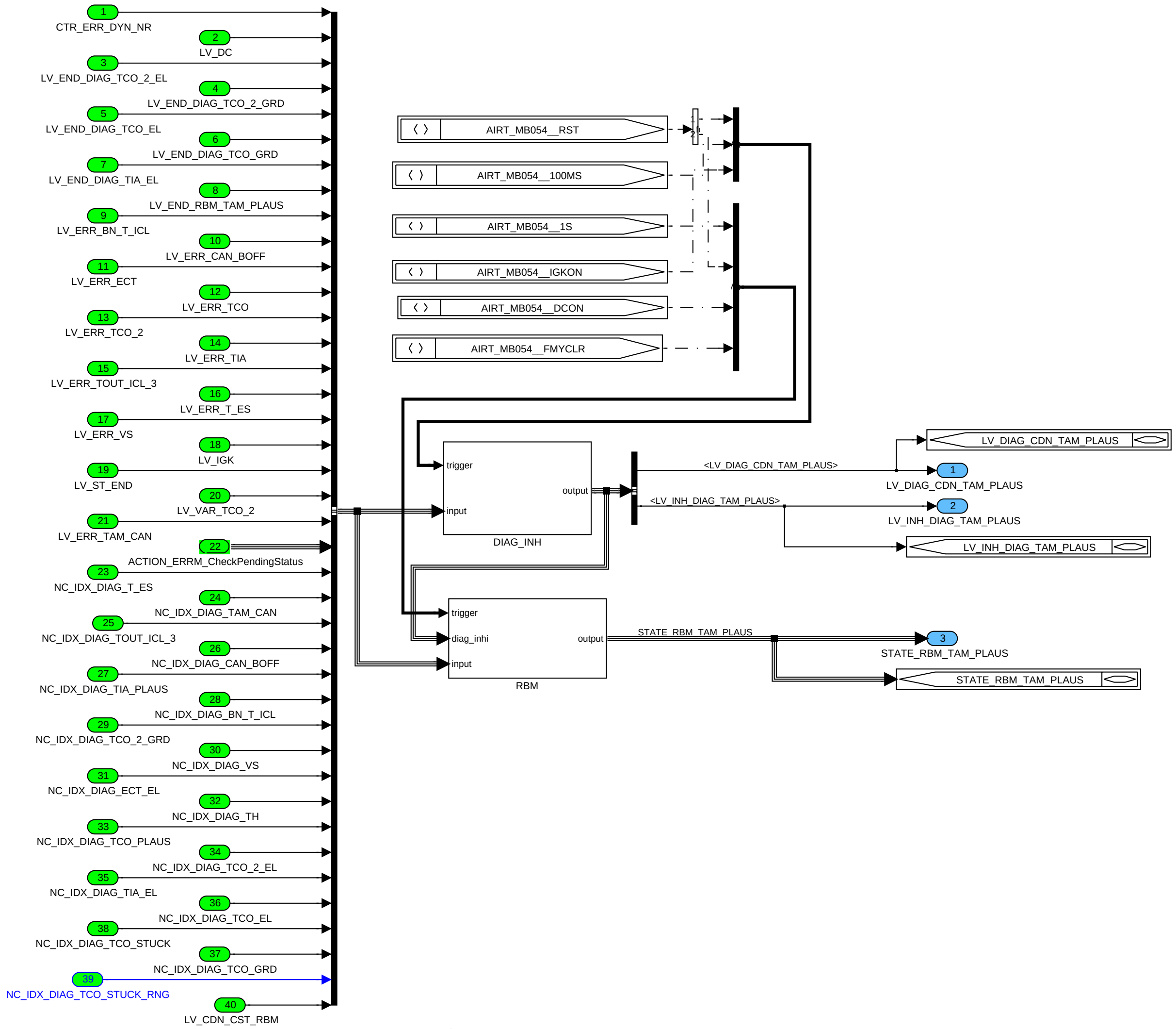
```
Endif
```

End of diagnosis cycle

```
If      LV_DIAG_CDN_TAM_PLAUS = 1   and    //TIA / TCO_2 or TCO signal is valid
        LV_VS_CDN_TAM_ERR = 1                      //Vehicle was driven
Then    LV_CDN_DIAG_TAM_PLAUS = 1
        ERR_SYM_TAM_PLAUS = NO_SYM
        LV_END_DIAG_TAM_PLAUS = 1
        LV_END_RBM_TAM_PLAUS = 1
    If      LV_DET_ERR_TAM_PLAUS = 1
    Then    ERR_SYM_TAM_PLAUS = SYM_3
            LV_ERR_TAM_PLAUS = 1
    Endif
Endif
```

1.11 Ambient temperature sensor plausibility check (Applic. Inc.)

Overview



SDA_SRS / SDA 4.0 22−Nov−2005

Figure 94 AIRT_MB054

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_CDN_TAM_PLAUS	O/V	0...1H	0...1	1	-
Flag for status ¿all TCO / TCO_2 diagnosis cycle are finished¿					
LV_INH_DIAG_TAM_PLAUS	O/V	0...1H	0...1	1	-
Inhibition of TAM plaus diagnosis due to present error					
STATE_RBM_TAM_PLAUS	O/V	0...7H	0...7	1	-
Interface of TAM_PLAUS monitor with the Rate-Based Monitoring statisticsBit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1)Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1)Bit 2: individual RBM co					

Input data:

CTR_ERR_DYN_NR	LV_DC	LV_END_DIAG_TCO_2_E	LV_END_DIAG_TCO_2_G

		L	RD
LV_END_DIAG_TCO_EL	LV_END_DIAG_TCO_GRD	LV_END_DIAG_TIA_EL	LV_END_RBM_TAM_PLAUS
LV_ERR_BN_T_ICL	LV_ERR_CAN_BOFF	LV_ERR_ECT	LV_ERR_TCO
LV_ERR_TCO_2	LV_ERR_TIA	LV_ERR_TOUT_ICL_3	LV_ERR_T_ES
LV_ERR_VS	LV_IGK	LV_ST_END	LV_VAR_TCO_2
LV_ERR_TAM_CAN	NC_IDX_DIAG_T_ES	NC_IDX_DIAG_TAM_CAN	NC_IDX_DIAG_TOUT_ICL_3
NC_IDX_DIAG_CAN_BOFF	NC_IDX_DIAG_TIA_PLAUS	NC_IDX_DIAG_BN_T_ICL	NC_IDX_DIAG_TCO_2_GRD
NC_IDX_DIAG_VS	NC_IDX_DIAG_ECT_EL	NC_IDX_DIAG_TH	NC_IDX_DIAG_TCO_PLAUS
NC_IDX_DIAG_TCO_2_EL	NC_IDX_DIAG_TIA_EL	NC_IDX_DIAG_TCO_EL	NC_IDX_DIAG_TCO_GRD
NC_IDX_DIAG_TCO_STUCK	NC_IDX_DIAG_TCO_STUCK_RNG	LV_CDN_CST_RBM	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_DIAG_TAM_PLAUS	1	0...1H	0...1	1	-
Switch to shut off the TAM plaus diagnosis (=1)					

Import actions:

ACTION_ERRM_CheckPendingStatus(IN <PRM_IDX_DIAG>, OUT <PRM_LV_ERR_PND>)
This API shall be used to verify if a failure stored in the dynamic memory has the pending status or not. When it impossible to determine if the fault is pending or not (failure not store because the dynamic memory is full), this failure should be considered as pending anyway.

1.11.1 Inhibition of diagnosis

Application Condition

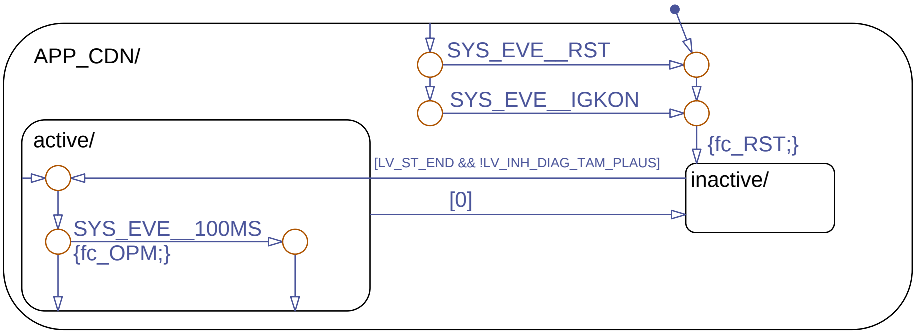


Figure 95 AIRT_MB054/ DIAG_INH/ APP_CDN/ Chart

Function Description

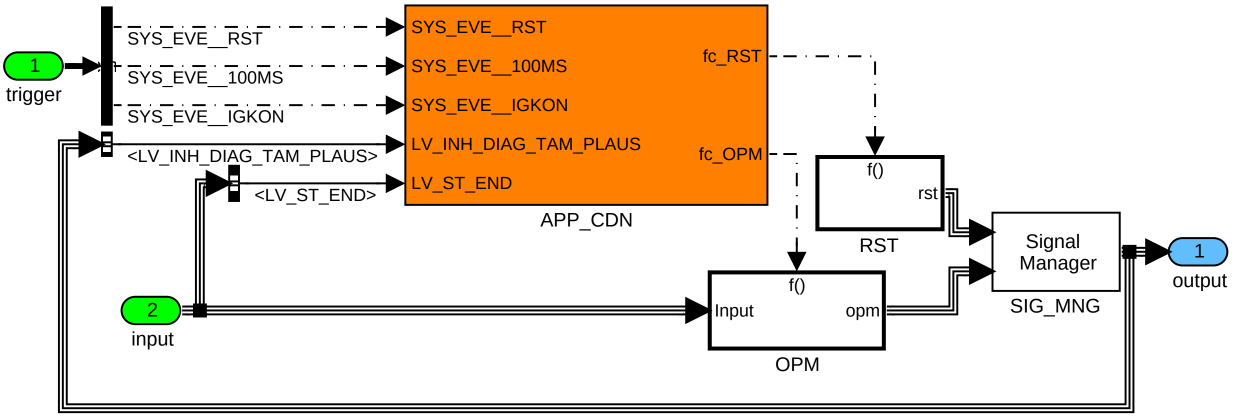


Figure 96 AIRT_MB054/ DIAG_INH

1.11.1.1 Formula Section

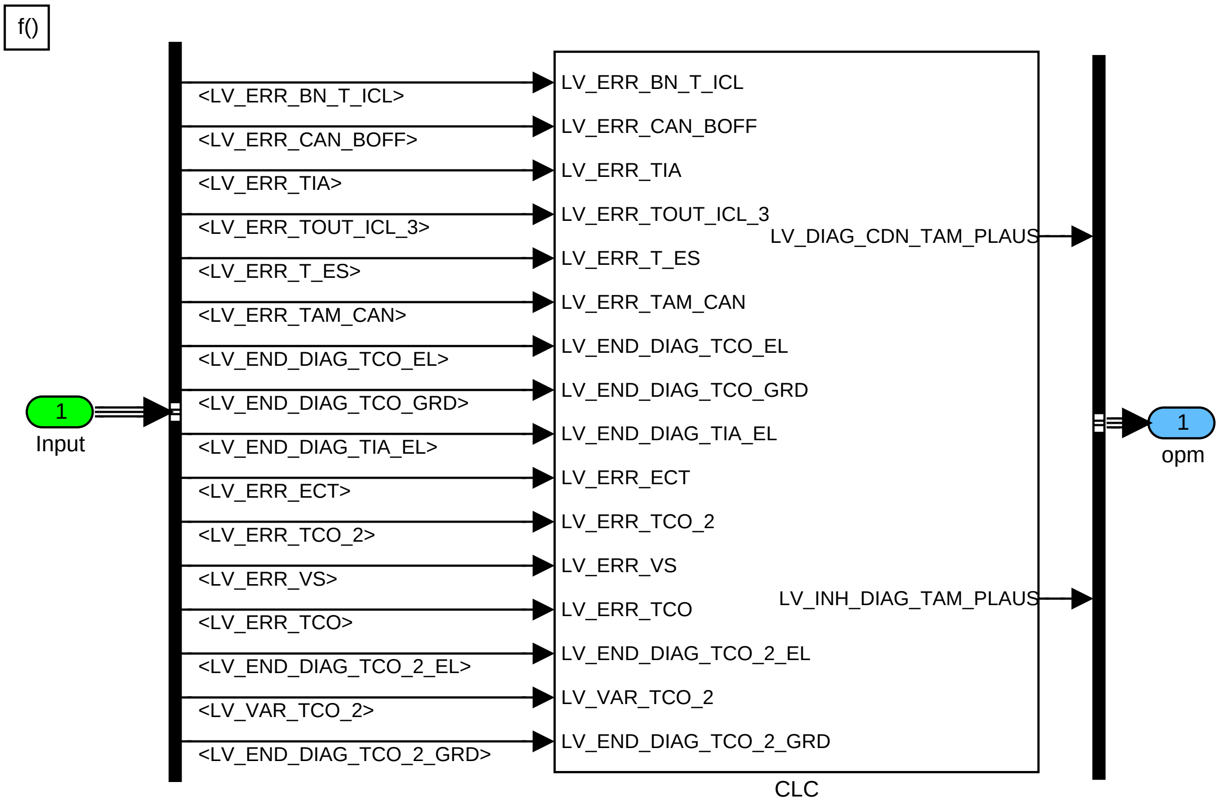


Figure 97 AIRT_MB054/ DIAG_INH/ OPM

Calculation

All the calculation within formula section will be done when this block is triggered. The output variables will be set to one if the respective conditions are true.

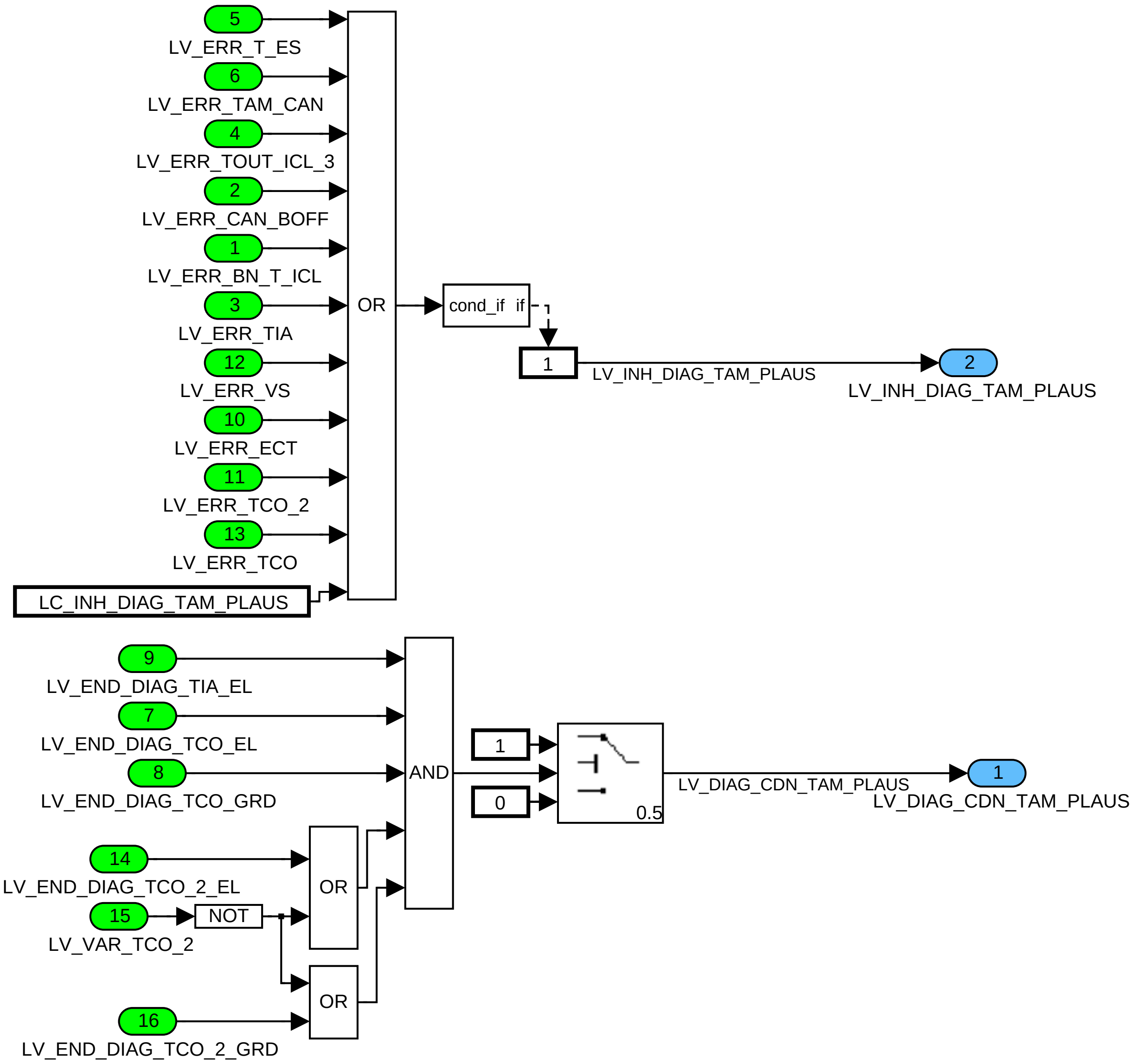


Figure 98 AIRT_MB054/ DIAG_INH/ OPM/ CLC

1.11.1.2 Initialization

The output variables are reset to zero.

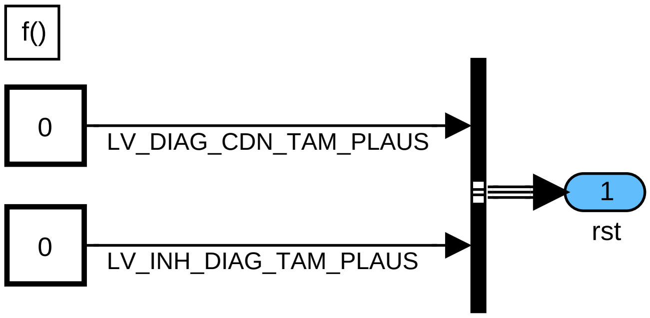


Figure 99 AIRT_MB054/ DIAG_INH/ RST

1.11.1.3 SUBFUNCTION: SIG_MNG

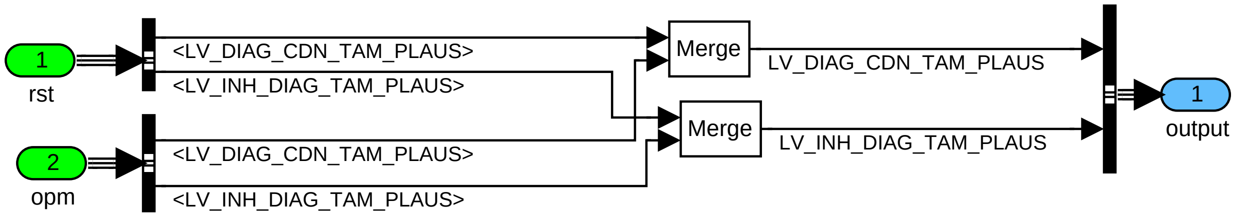


Figure 100 AIRT_MB054/ DIAG_INH/ SIG_MNG

1.11.2 Interface for Rate – Based – Monitoring

General information:

With this module the interface between the TAM_PLAUS monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TAM_PLAUS data.

Within STATE_RBM_TAM_PLAUS, three different information are defined:

Conditions for monitoring are met long enough to detect malfunction (bit 0)

(no intrusive operation, no short trip)

Monitor disabled because of system malfunction (bit 1)

(depending on failure status: pending)

Application Condition

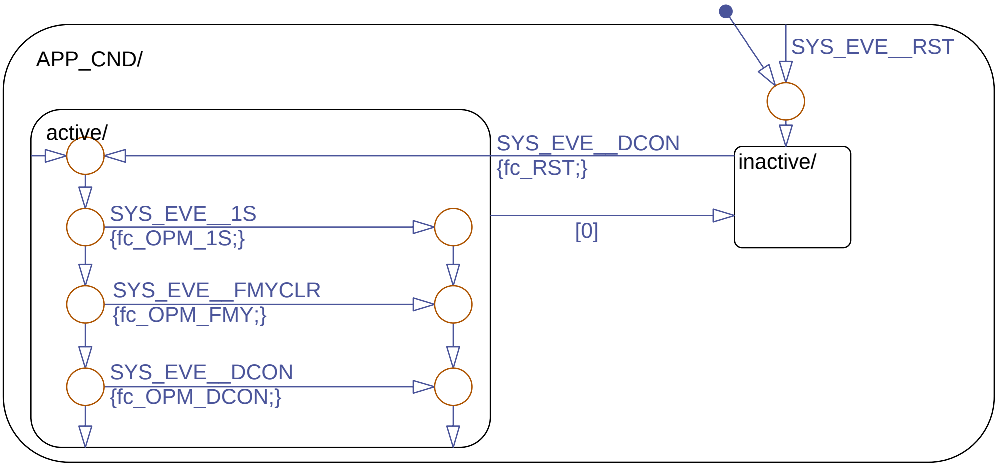


Figure 101 AIRT_MB054/ RBM/ APP_CDN/ Chart

Function Description

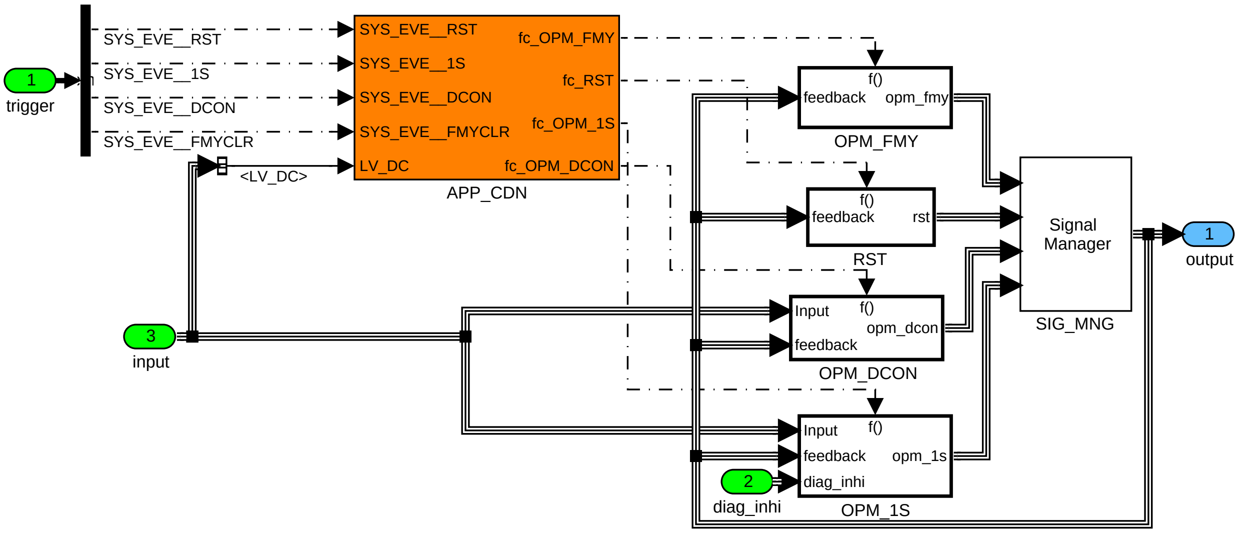


Figure 102 AIRT_MB054/ RBM

1.11.2.1 Reset at FMY_CLR EVENT

On failure memory reset the bit 1 of STATE_RBM_TAM_PLAUS.

f()

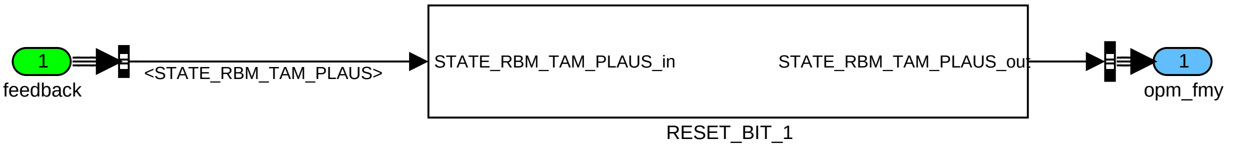


Figure 103 AIRT_MB054/ RBM/ OPM_FMY

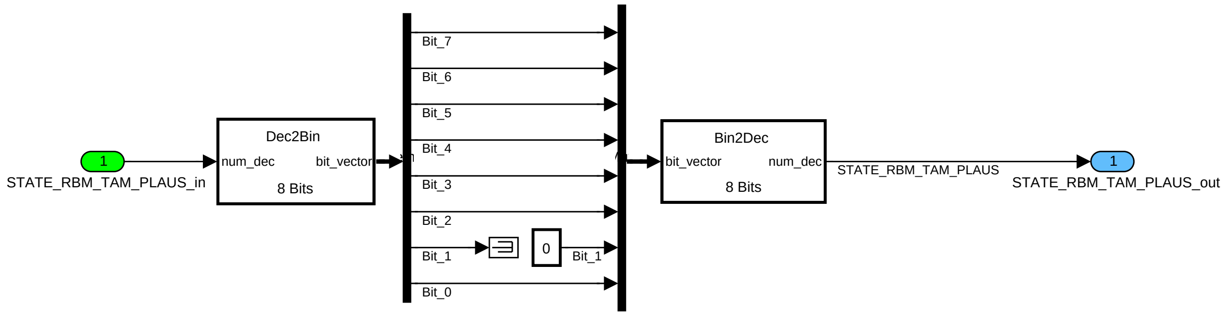


Figure 104 AIRT_MB054/ RBM/ OPM_FMY/ RESET_BIT_1

1.11.2.2 Initialization at DCON

At 0->1 transition of LV_DC, the bit 0, bit 1 and bit 2 of STATE_RBM_TAM_PLAUS are reset to zero.

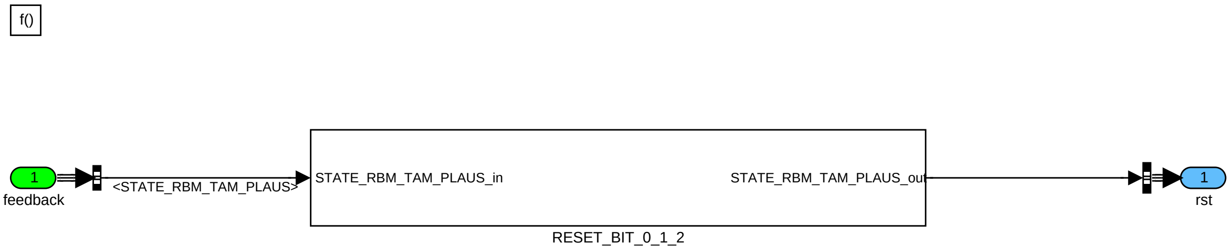


Figure 105 AIRT_MB054/ RBM/ RST

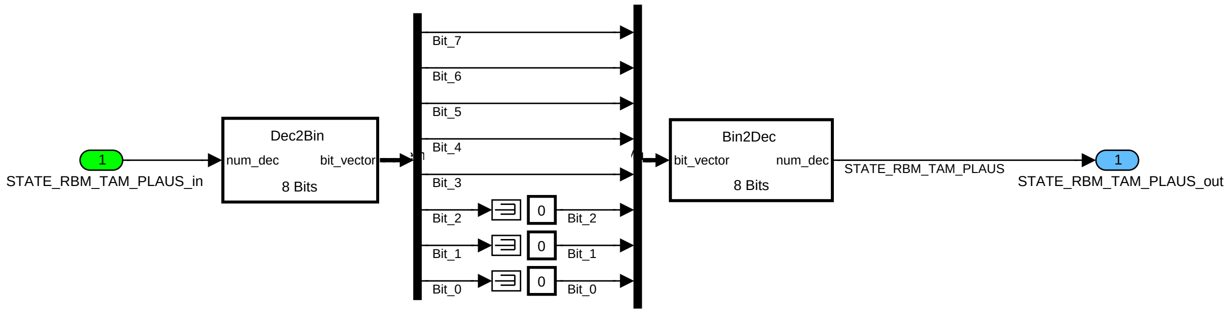


Figure 106 AIRT_MB054/ RBM/ RST/ RESET_BIT_0_1_2

1.11.2.3 Formula Section for SYSTEM EVENT DCON

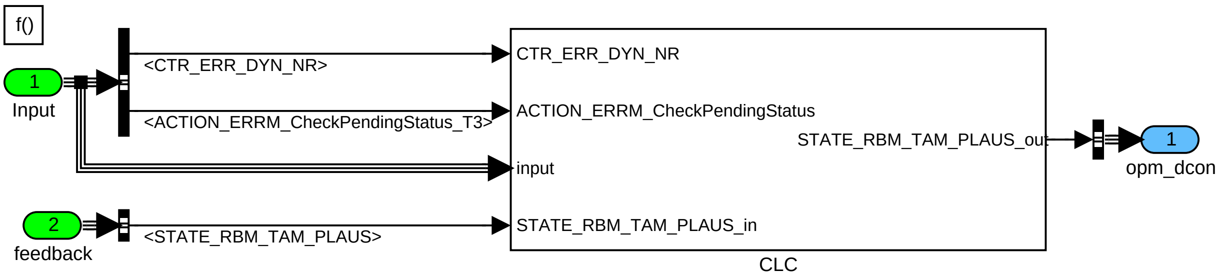


Figure 107 AIRT_MB054/ RBM/ OPM_DCON

If_Then

All the calculation within formula section will be done when this block is triggered. The pending status check function will be called if CTR_ERR_DYN_NR is greater than zero.

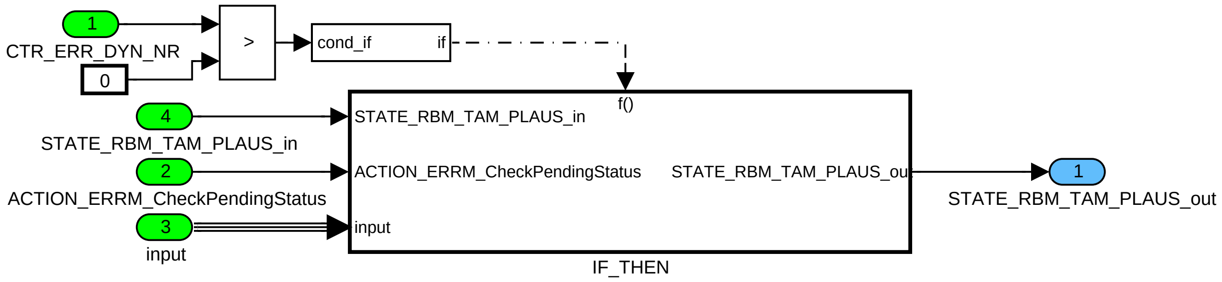


Figure 108 AIRT_MB054/ RBM/ OPM_DCON/ CLC

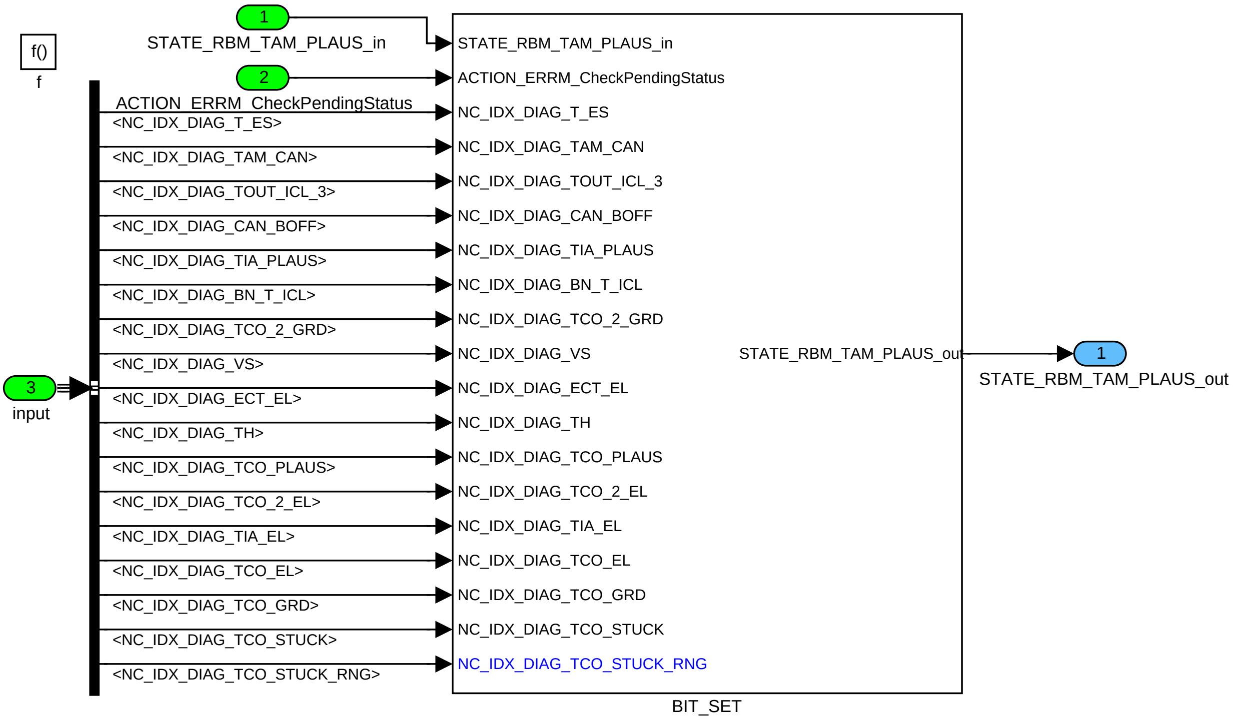


Figure 109 AIRT_MB054/ RBM/ OPM_DCON/ CLC/ IF_THEN

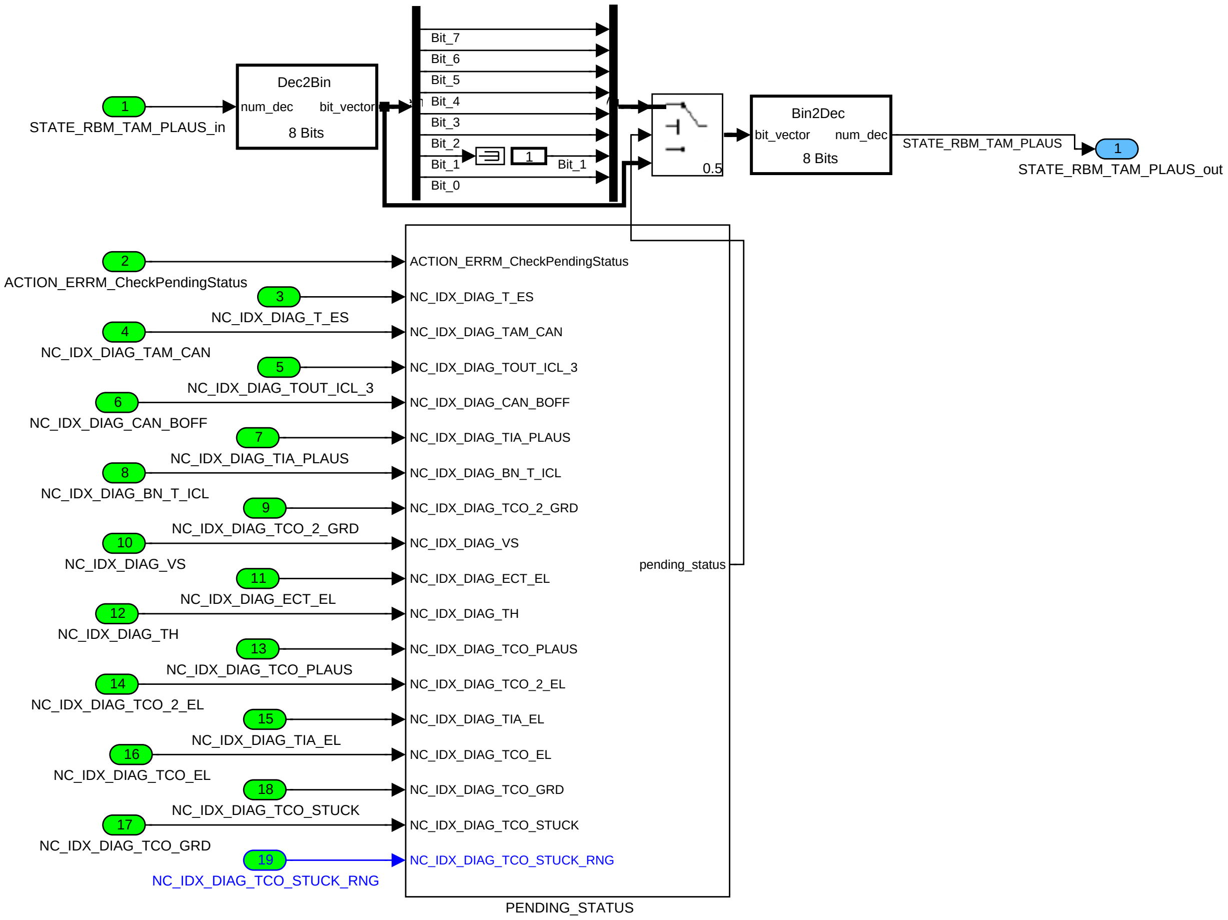


Figure 110 AIRT_MB054/ RBM/ OPM_DCON/ CLC/ IF_THEN/ BIT_SET

Pending_Status Check

This API shall be used to verify if a failure stored in the dynamic memory has the pending status or not. When it is impossible to determine if the fault is pending or not (failure not store because the dynamic memory is full), this failure should be considered as pending anyway.

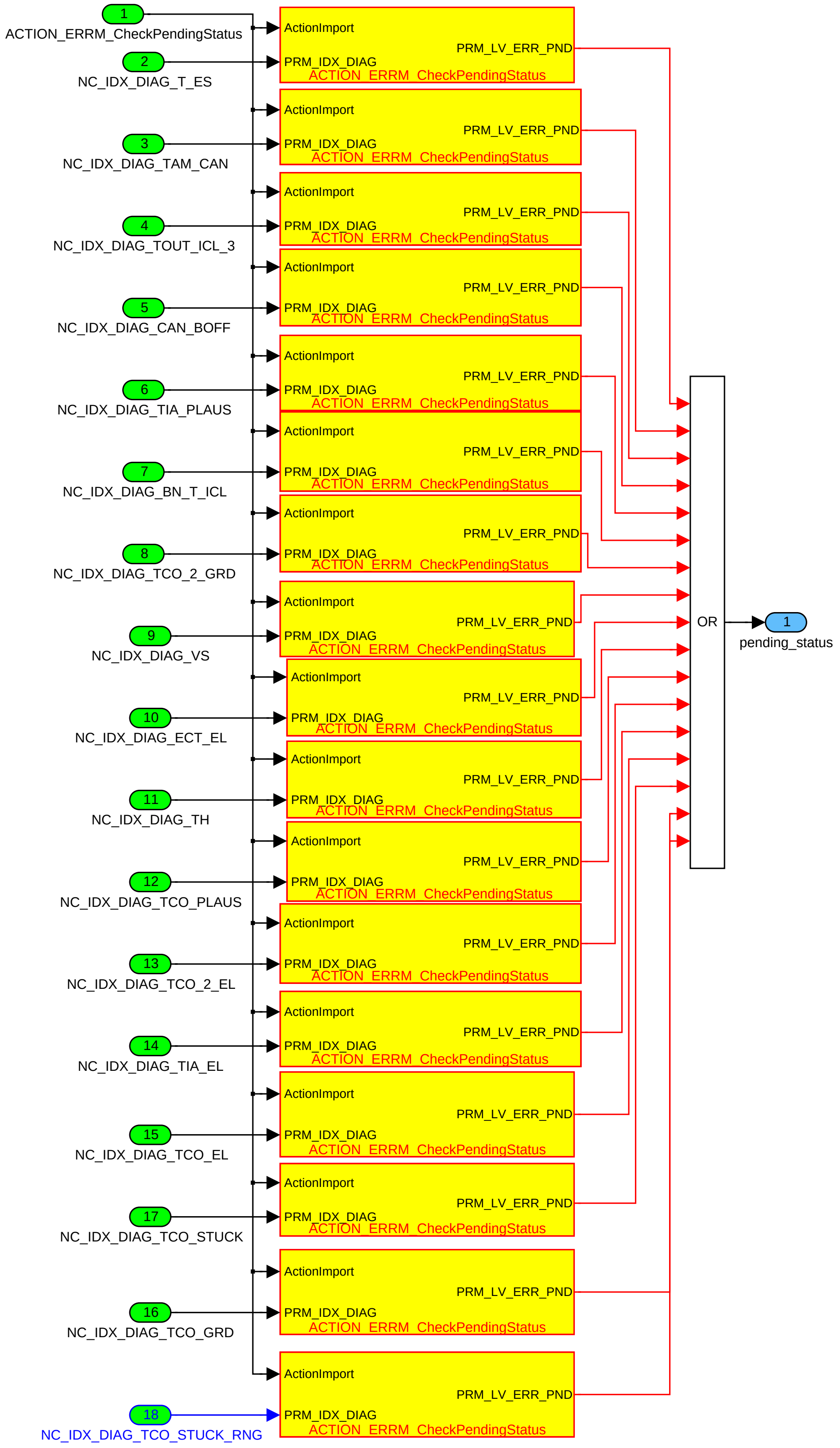


Figure 111 AIRT_MB054/ RBM/ OPM_DCON/ CLC/ IF_THEN/ BIT_SET/ PENDING_STATUS

1.11.2.4 Formula Section for SYSTEM EVENT 1S

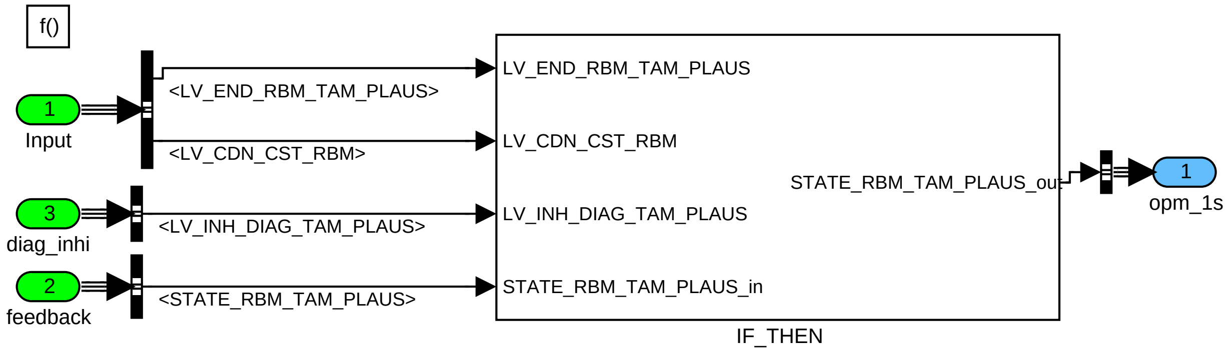


Figure 112 AIRT_MB054/ RBM/ OPM_1S

If_Then

All the calculation within formula section will be done when this block is triggered. If the condition is fulfilled then the bit 0 and bit 1 of STATE_RBM_TAM_PLAUS will be set to 1.

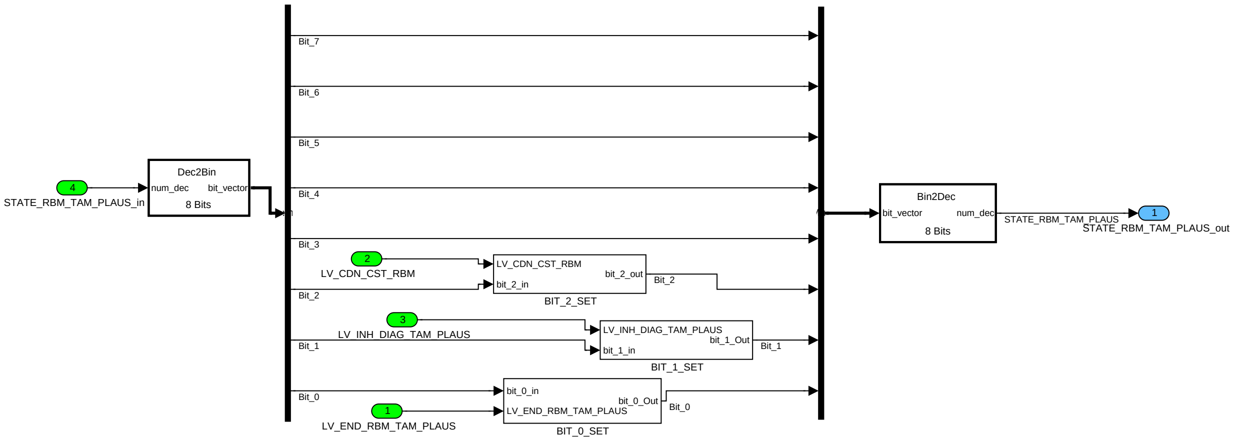


Figure 113 AIRT_MB054/ RBM/ OPM_1S/ IF_THEN

Bit_0_Set

Bit 0 of STATE_RBM_TAM_PLAUS is set to 1 if the condition is true.

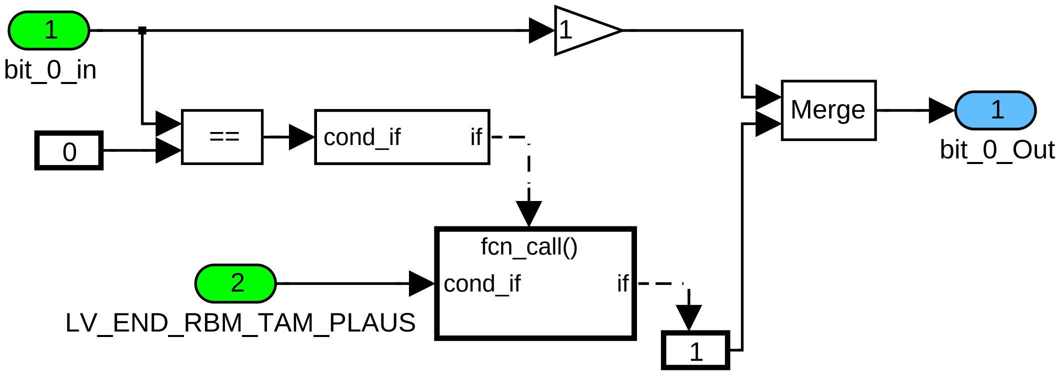


Figure 114 AIRT_MB054/ RBM/ OPM_1S/ IF_THEN/ BIT_0_SET

Bit_1_Set

Bit 1 of STATE_RBM_TAM_PLAUS is set to 1 if the condition is true.

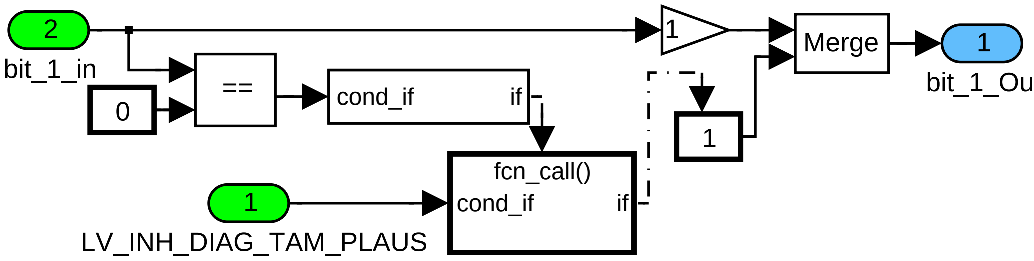


Figure 115 AIRT_MB054/ RBM/ OPM_1S/ IF_THEN/ BIT_1_SET

Bit_2_Set

Bit 2 of STATE_RBM_TAM_PLAUS is set to 1 if the condition is true.

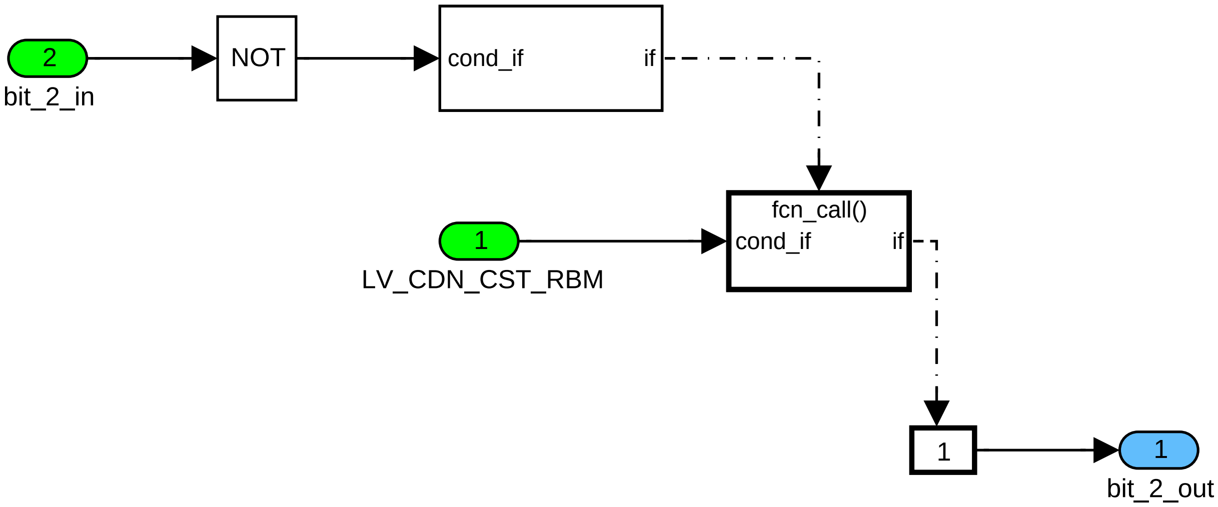


Figure 116 AIRT_MB054/ RBM/ OPM_1S/ IF_THEN/ BIT_2_SET

1.12 Customer adaptation module: AIRT

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Tans	O/V	FE0C...5DCH	-50...150	0.1	°C
intake air temperature					
Tumg	O/V	FE0C...5DCH	-50...150	0.1	°C
ambient temperature (modelled)					
Tumg_can	O/V	FE0C...5DCH	-50...150	0.1	°C
ambient temperature					
Tvld	O/V	FE0C...5DCH	-50...150	0.1	°C
Air temperature up turbo charger					
B_tumers	O/V	0..1H	0..1	1	[-]
Ambient temperature via CAN faulty, substitute value is taken					

Input data:

TIA	TAM	TIA_TCHA	LV_TAM_CAN_ERR

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: *at reset:* 0: Tumg_can, B_tumers
first measured value: Tumg, Tans, Tvld
at PWL: last measured value: Tumg, Tans, Tvld
last valid value: Tumg_can
0: B_tumers

Recurrence: 100ms: Tumg, Tumg_can, Tans, Tvld, B_tumers

Activation: every engine state

Deactivation:

Formula section:

Remark: all formulas are valid in a **physical** meaning

Tans = TIA

Tumg = TAM

Tumg_can = TAM

Tvld = TIA_TCHA

B_tumers = LV_TAM_CAN_ERR

2 Combustion modes

Table of Contents

Index

2.1 Lambda testing for Stratified Mode

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_SP_S_RAW	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Raw value lambda setpoint for stratified mode					
LAMB_SP_S	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda setpoint for stratified mode					

Input data:

MAF_SP_S	MFF_CP	N_32	
LV_S_CLC	MFF_SP_S	LAMB_SP_S_EXT	

FUNCTION DESCRIPTION:

General information:

The lambda LAMB_SP_S in stratified has to be limited to a minimum lambda value. If the lambda setpoint LAMB_SP_S is below the minimum threshold LAMB_MIN_S then the stratified combustion is not authorized.

Description:

Application conditions:

Activation: LV_S_CLC = 1

Deactivation: LV_S_CLC = 0

Initialization: at reset:

LAMB_SP_S(n-1) = C_LAMB_SP_S_INI

at the transition of LV_S_CLC from 0 to 1:

LAMB_SP_S_RAW has to be calculated for the initialization.

LAMB_SP_S(n-1) = LAMB_SP_S_RAW.

Update rate: 10 ms

Formula section:

```
IF      LC_LAMB_SP_S_RAW_EXT = 0
Then    LAMB_SP_S_RAW   =
                     MAF_SP_S / (C_FAC_LAMB_AFS * (MFF_SP_S + MFF_CP))
ELSE    LAMB_SP_S_RAW   = LAMB_SP_S_EXT ("from Layer")
endif
LAMB_SP_S(n) =
C_CRLC_LAMB_SP_S * (LAMB_SP_S_RAW – LAMB_SP_S(n-1)) + LAMB_SP_S(n-1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_LAMB_AFS	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Air fuel stoichometric factor (typical value 14.71)					
C_CRLC_LAMB_SP_S	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation filter constant for LAMB_SP_S filtering					
C_LAMB_SP_S_INI	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Initialization value for LAMB_SP_S after reset.					
LC_LAMB_SP_S_RAW_EXT	1	0...1H	0...1	1	[-]
Switch between Layer and internal calculatiopn of LAMB_SP_S					

2.2 Combustion State Manager (Appl. Incidences)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_S	V/O	0...1	0...1	1	-
Inhibit of stratified mode due to project specific requests.					
LV_INH_AFL	V/O	0...1	0...1	1	-
Inhibit of AFL mode due to project specific requests.					
LV_S_CLC	V/O	0...1	0...1	1	-
Logical variable for the deactivation of all straified mode relevant modules.					
LV_AFL_CLC	V/O	0...1	0...1	1	-
Logical variable for the deactivation of all AFL- mode relevant modules.					

Input data:

LV_INH_S_MAN	LV_BRAKE_REQ	LV_ERR_EGR	LV_TEMP_BOL
STATE_CMB_CTL	N_32	LV_MIS_STATE_A	LV_MIS_STATE_B
LV_SAWUP	LV_ERR_PORT	LV_ERR_FPS	
LV_ERR_NOX_SENS_CAN_BOFF	LV_FPAPWM_DIAG_AFS_REQ		

FUNCTION DESCRIPTION:

There are several project specific requests to inhibit the stratified and AFL mode. Within this function all these requests lead to the bits LV_INH_S and LV_INH_AFL to simpify the combustion manager.

To reduce the processor load all stratified mode related modules shall no longer be calculated, if LV_S_CLC = 0. All AFL-mode relevant modules including the combustion manager shall no longer be calculated, if LV_AFL_CLC = 0.

Application conditions:

Initialisation: at reset: LV_INH_S = 1
LV_INH_AFL = 1
LV_S_CLC = 1
LV_AFL_CLC = 1

Recurrence: 10 ms

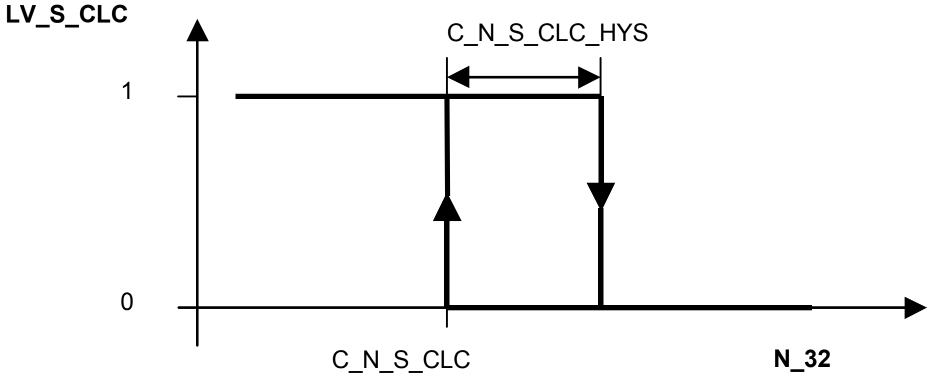
Activation: every engine state

Deactivation: -

Formula section:

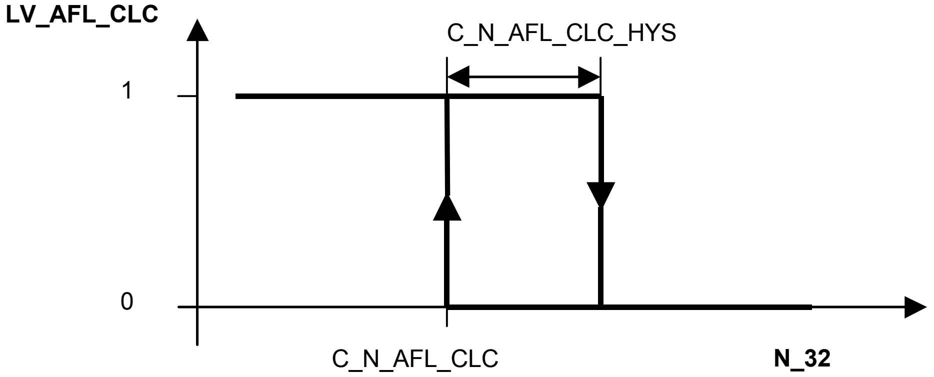
Definition of LV_S_CLC:

This calculation will only be done if the STATE_CMB_CTL = HOM_AFS.



Definition of LV_AFL_CLC:

This calculation will only be done if the STATE_CMB_CTL = HOM_AFS.



```
IF  LV_INH_S_MAN               or
    LV_MIS_STATE_A             or
    LV_MIS_STATE_B             or
    LV_ERR_PORT                or
    LV_ERR_EGR                 or
    LV_ERR_NOX_SENS_CAN_BOFF             or
    LV_ERR_FPS          or
    LV_ERR_NOX_SENS_CAN_MSG_LOST_i                    or
    LV_SAWUP            or
    LV_FPAPWM_DIAG_AFS_REQ               or
    LV_BRAKE_REQ               or
    LV_TEMP_BOL                or
    Not LV_S_CLC               or
Then        LV_INH_S = 1
Else        LV_INH_S = 0
```

```
IF  LV_MIS_STATE_A             or
    LV_MIS_STATE_B             or
    LV_ERR_PORT                or
    LV_ERR_EGR                 or
    LV_ERR_NOX_SENS_CAN_BOFF             or
    LV_ERR_FPS          or
    LV_ERR_NOX_SENS_CAN_MSG_LOST_i                    or
    LV_SAWUP            or
    LV_FPAPWM_DIAG_AFS_REQ               or
    LV_BRAKE_REQ               or
    LV_TEMP_BOL                or
    Not LV_AFL_CLC             or
Then        LV_INH_AFL = 1
Else        LV_INH_AFL = 0
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_S_CLC	1	0...FFH	0...8160	32	rpm
Engine speed limit for the deactivation of the stratified mode related modules.					
C_N_S_CLC_HYS	1	0...FFH	0...8160	32	rpm
Engine speed hysteresis for the deactivation of the stratified mode related modules.					
C_N_AFL_CLC	1	0...FFH	0...8160	32	rpm
Engine speed limit for the deactivation of the AFL mode related modules.					
C_N_AFL_CLC_HYS	1	0...FFH	0...8160	32	rpm
Engine speed hysteresis for the deactivation of the AFL mode related modules.					

2.2.1 Variables of Non implemented Functions for combustion management

Initialisation:

NT_AGI_S_RED = 0 (if functionallity is not implemented yet)
LV_NT_STC_MAX_ACT = 0 (if functionallity is not implemented yet)
LV_PUC_INH_TEMP_CAT = 0 (if functionallity is not implemented yet)
MFF_CP = 0 mg/stk (if functionallity is not implemented yet)

2.3 Customer adaptation module: AGGR CBMD

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_ase_abgas	V/O	0...1H	0...1	1	[-]
Logical variable for CARB A misfire criterion confirmed					
B_ase_kat	V/O	0...1H	0...1	1	[-]
Logical variable for CARB B misfire criterion confirmed					
B_vb_hmm	V/O	0...1H	0...1	1	[-]
Inhibit of AFL mode due to project specific requests.					
B_vb_sch	V/O	0...1H	0...1	1	[-]
Inhibit of stratified mode due to project specific requests.					
B_hmm_clc	V/O	0...1H	0...1	1	[-]
Logical variable for the deactivation of all AFL- mode relevant modules.					
B_sch_clc	V/O	0...1H	0...1	1	[-]
Logical variable for the deactivation of all S- mode relevant modules.					
B_sk_homla1	V/O	0...1H	0...1	1	[-]
flag for switching to HOM_AFS due to faults in non-HOM_AFS					
LV_S_RUN	V/O	0...1H	0...1	1	[-]
Flag to activate stratified calculations					
LV_HOM_RUN	V/O	0...1H	0...1	1	[-]
Flag to activate homogeneous calculations					
LV_HOM_AFS_REQ	V/O	0...1H	0...1	1	[-]
Request for AFS mode					
OPM_AV	V/O	0H 1H 2H 3H 8H	- S AFS AFL LIH	1	[-]
actual engine operation mode					
OPM_REQ	V/O	0H 1H 2H 3H 8H	- S AFS AFL LIH	1	[-]
Requested engine operation mode (for operation switch manager)					
OPM_REQ_CUS	V/O	0...FFFFH	0...65535	1	[-]
Actual engine operation mode (16bit)					
LV_S_ACT	V/O	0...1H	0...1	1	[-]
Stratified ignition and injection parameters are applied to the engine					
LV_HOM_ACT	V/O	0...1H	0...1	1	[-]
Homogeneous ignition and injection parameters are applied to the engine					
STATE_CMB_CTL	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H	HOM_AFS AFS_TO_AFL HOM_AFL AFL_TO_AFS HOM_TO_S BACKS S S_TO_HOM BACKHOM	1	[-]
States of the combustion management					
LV_IGA_AND_INJ_SWI_1	V/O	0...1H	0...1	1	[-]
Logical variable controlling igintion and injection mode (stratified or homogeneous).					
LV_IGA_AND_INJ_SWI_2	V/O	0...1H	0...1	1	[-]
Logical variable controlling igintion and injection mode (stratified or homogeneous).					
LV_IGA_AND_INJ_SWI	V/O	0...1H	0...1	1	[-]
Logical variable controlling igintion and injection mode (stratified or homogeneous).					
LV_IGA_AND_INJ_SWI_HOMS_1	V/O	0...1H	0...1	1	[-]
Logical variable controlling igintion and injection mode (stratified or homogeneous-stratified), for bank 1					

LV_IGA_AND_INJ_SWI_HOMS_2	V/O	0...1H	0...1	1	[-]
Logical variable controlling igintion and injection mode (stratified or homogeneous-stratified)., for bank 2					
LV_IGA_AND_INJ_SWI_HOMS	V/O	0...1H	0...1	1	[-]
Logical variable controlling igintion and injection mode (stratified or homogeneous-stratified).					
LV_FUP_SP_SWI	V/O	0...1H	0...1	1	[-]
Flag for fuel pressure setpoint switch					
LV_HOM_AFL_ACT	V/O	0...1H	0...1	1	[-]
Homogeneous air fuel lean mode active					
LV_HOM_AFS_ACT	V/O	0...1H	0...1	1	[-]
Logical variables indicates active homogeneous stoichiometric state					
LV_SP_RATE_CYL_EGR_SWI	V/O	0...1H	0...1	1	[-]
Flag for EGR setpoint rate					
LV_S_ENA	V/O	0...1H	0...1	1	[-]
catalyst heating home					
LV_HOM_ENA	V/O	0...1H	0...1	1	[-]
catalyst heating home					
LAMB_SP_S_EXT	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
lambda setpoint for stratified mode					

Input data:

B_esp_h1	B_esp_h2	B_esp_hs1	B_esp_hs2
B_hmm_akt	B_hom_akt	B_hstoech	B_prsoll_h
B_sch_akt	B_schicht	B_verb_h	B_verb_s
Ba_ist	Ba_soll	Baw_ist	ECU_STATE
ECU_STATE	La_abgas1	La_abgas2	LV_AFL_CLC
LV_INH_AFL	LV_INH_S	LV_MIS_STATE_A	LV_S_CLC
LV_SWI_AFS_MON			

Application conditions:

Initialisation at reset:

0, except LAMB_SP_S_EXT = 1.00

Recurrence : 10 ms

Activation: at every engine state

Deactivation: ---

2.3.1 Outputs for BMW which are defined as CBMD exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Formula section:

if ECU_STATE = "PWL"

then

```
        B_ase_abgas  = 0
        B_ase_kat     = 0
        B_vb_hmm     = 0
        B_vb_sch      = 0
        B_hmm_clc    = 0
        B_sch_clc     = 0


    else
        B_ase_abgas        =     LV_MIS_STATE_B
        B_ase_kat          =     LV_MIS_STATE_A
        B_vb_hmm           =     LV_INH_AFL
        B_vb_sch           =     LV_INH_S
        B_hmm_clc          =     LV_AFL_CLC
        B_sch_clc          =     LV_S_CLC
        B_sk_homla1        =     LV_SWI_AFS_MON
    Endif
```

2.3.2 Outputs for SV aggregates

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes.

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
LV_S_RUN                          =     B_sch_akt
LV_HOM_RUN                        =     B_hom_akt
LV_HOM_AFS_REQ                    =     B_hstoech
OPM_AV                            =     Ba_ist
OPM_REQ                           =     Ba_soll
OPM_REQ_CUS                       =     Baw_ist
LV_ S_ACT                         =     B_schicht
LV_HOM_ACT                        =     ! B_schicht
LV_IGA_AND_INJ_SWI_1              =     B_esp_h1
LV_IGA_AND_INJ_SWI                =     B_esp_h1
LV_IGA_AND_INJ_SWI_2              =     B_esp_h2
LV_IGA_AND_INJ_SWI_HOMS_1         =     B_esp_hs1
LV_IGA_AND_INJ_SWI_HOMS           =     B_esp_hs1
LV_IGA_AND_INJ_SWI_HOMS_2         =     B_esp_hs2
LV_FUP_SP_SWI          =     B_prsoll_h
LV_HOM_AFL_ACT         =     B_hmm_akt

if   Ba_ist = 2  and  LV_HOM_AFL_ACT = 0
then      STATE_CMB_CTL=0Hex                      “HOM_AFS”
else      if Ba_ist = 2  and  LV_HOM_AFL_ACT = 1
          then        STATE_CMB_CTL=2Hex          “HOM_AFL”
          else        if Ba_ist=1 or Ba_ist=3
```

```
                    then    STATE_CMB_CTL=6Hex    "S"
                    endif
            endif
    endif

    If      Ba_ist = 1
    then        LV_SP_RATE_CYL_EGR_SWI = 0
    else        LV_SP_RATE_CYL_EGR_SWI = 1
    endif

    If          STATE_CMB_CTL = 0Hex
    Then        LV_HOM_AFS_ACT  =       1
    Else        LV_HOM_AFS_ACT  =       0
    Endif
    LV_S_ENA= ! B_verb_s
    LV_HOM_ENA      =! B_verb_h
    LAMB_SP_S_EXT = 0,5*(La_abgas1 + La_abgas2)
```

3 Charger

Table of Contents

Index

3.1 Waste gate driver diagnosis (LV_ERR_WG_x_DR)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_WG_1_DR	V/O	0...1H	0...1	1	[-]
Waste gate driver error cylinder bank 1 detected					
LV_CDN_DIAG_WG_1_DR	V/O	0...1H	0...1	1	[-]
Diagnosis condition Waste gate driver cylinder bank 1					
ERR_SYM_WG_1_DR	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom Waste gate driver cylinder bank 1					
LV_END_DIAG_WG_1_DR	V/O	0...1H	0...1	1	[-]
End of diagnosis Waste gate driver cylinder bank 1					
LV_INH_DIAG_WG_1_DR	V/O	0...1H	0...1	1	[-]
Waste gate driver diagnosis inhibition cylinder bank 1					
CDN_DIAG_WG_1_DR	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of WG_1_DR bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_WG_1_DR	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for WG_1_DR (only parameter)					
LV_ERR_WG_2_DR	V/O	0...1H	0...1	1	[-]
Waste gate driver error cylinder bank 2 detected					
LV_CDN_DIAG_WG_2_DR	V/O	0...1H	0...1	1	[-]
Diagnosis condition Waste gate driver cylinder bank 2					
ERR_SYM_WG_2_DR	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom Waste gate driver					
LV_END_DIAG_WG_2_DR	V/O	0...1H	0...1	1	[-]
End of diagnosis Waste gate driver cylinder bank 2					
LV_INH_DIAG_WG_2_DR	V/O	0...1H	0...1	1	[-]
Waste gate driver diagnosis inhibition cylinder bank 2					
CDN_DIAG_WG_2_DR	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of WG_2_DR bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_WG_2_DR	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for WG_2_DR (only parameter)					

Input data:

LV_CDN_VB_OBD1	LV_IGK	WGPWM[NC_CBK_EX_NR]	LV_VAR_TCHA
LV_ERR_SPI_MPS			

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis is to detect electrical faults in the waste gate. There are one for each exhaust bank (x = 1 or 2). All calibration data will be copyed into SW internal CBK spezific calibration data. Example:

C_ABC_INC_WG_DR = C_ABC_INC_WG_1_DR = C_ABC_INC_WG_2_DR

The input signal is a modulated control pulse (PWM).

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.
The diagnosis condition calculation is done according **“Open range”** duty cycle, thus a definition of the PWM diagnosis windows is neccessary (see calibration data).

Error-symptoms are defined to this diagnosis function as following :

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 |  |  |  └─> Short circuit to VBatt          (= SYM_0)
 |  |  └────> Short circuit to GND            (= SYM_1)
 |  └───────> Open load                       (= SYM_2)
 └──────────> -                               (= SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**
(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 100ms

Activation: LV_VAR_TCHA =1 at every engine operating state

```
Deactivation:    If      LV_VAR_TCHA =0
                 Then    LV_END_DIAG_WG_X_DR = 1
                 Endif
```

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_WG_X_DR = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_WG_X_DR.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_WG_X_DR.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_WG_X_DR and ERR_DIAG_WG_X_DR.

This algorithm determines:

ERR_SYM_WG_X_DR	(detected error symptom)
LV_ERR_WG_X_DR	(Error flag for debounced error)

CDN_DIAG_WG_X_DR (Diagnosis condition information)

LV_END_DIAG_WG_X_DR (End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_IGK = 1              AND
        LV_CDN_VB_OBD1 = 1      AND
        LV_ERR_SPI_MPS = 0
THEN    LV_INH_DIAG_WG_X_DR = 0
ELSE    LV_INH_DIAG_WG_X_DR = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_WG_DR	1	0...FFH	0...255	1	[-]
Debounce counter increment - Waste gate 1					
C_ABC_MAX_WG_DR	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - Waste gate					
C_WG_DR_PWM_DIAG_MIN_SCB	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Minimum threshold for SCB diagnosis window					
C_WG_DR_PWM_DIAG_MAX_SCG	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Maximum threshold for SCG diagnosis window					
C_WG_DR_PWM_DIAG_MIN_OC	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Minimum threshold for OC diagnosis window					

3.2 Recirculation flap diagnosis (LV_ERR_RFP_DR)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_RFP_DR	V/O	0...1H	0...1	1	[-]
recirculation flap driver error detected					
LV_CDN_DIAG_RFP_DR	V/O	0...1H	0...1	1	[-]
Diagnosis condition recirculation flap driver fullyfied					
ERR_SYM_RFP_DR	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom recirculation flap driver					
LV_END_DIAG_RFP_DR	V/O	0...1H	0...1	1	[-]
End of diagnosis recirculation flap driver reached					
LV_INH_DIAG_RFP_DR	V/O	0...1H	0...1	1	[-]
recirculation flap driver diagnosis inhibition activ					
CDN_DIAG_ RFP_DR	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of RFP_DR bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_ RFP_DR	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for RFP_DR(only parameter)					

Input data:

LV_CDN_VB_OBD1	LV_IGK	RFPPWM	LV_VAR_TCHA
LV_ERR_SPI_MPS			

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis is to detect electrical faults in the recirculation flap.

The input signal is a modulated control pulse (PWM).

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.
The diagnosis condition calculation is done according **“Open range”** duty cycle, thus a definition of the PWM diagnosis windows is neccessary (see calibration data).

Error-symptoms are defined to this diagnosis function as following :

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │  │  │  └─> Short circuit to VBatt          (= SYM_0)
  │  │  └────> Short circuit to GND            (= SYM_1)
  │  └───────> Open load                       (= SYM_2)
  └──────────> -                               (= SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 100ms

```
Activation:          LV_VAR_TCHA = 1
Deactivation:  If    LV_VAR_TCHA = 0
               Then  LV_END_DIAG_RFP_DR = 1
               Endif
```

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_RFP_DR = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_RFP_DR.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_RFP_DR.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_RFP_DR and ERR_DIAG_RFP_DR.

This algorithm determines:

ERR_SYM_RFP_DR	(detected error symptom)
LV_ERR_RFP_DR	(Error flag for debounced error)
LV_CDN_DIAG_RFP_DR	(Diagnosis condition information)
LV_END_DIAG_RFP_DR	(End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_IGK = 1               AND
        LV_CDN_VB_OBD1 = 1       AND
        LV_ERR_SPI_MPS = 0
THEN    LV_INH_DIAG_RFP_DR = 0
ELSE    LV_INH_DIAG_RFP_DR = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_RFP_DR	1	0...FFH	0...255	1	[-]
Debounce counter increment - recirculation flap driver					
C_ABC_MAX_RFP_DR	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - recirculation flap driver					
C_RFP_DR_PWM_DIAG_MIN_SCB	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Minimum threshold for SCB diagnosis window recirculation flap driver					
C_RFP_DR_PWM_DIAG_MAX_SCG	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Maximum threshold for SCG diagnosis window recirculation flap driver					
C_RFP_DR_PWM_DIAG_MIN_OC	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Minimum threshold for OC diagnosis window recirculation flap driver					

3.3 Charge Air Pressure Sensor Diagnosis (PUT)

Output data:

Name	Mod.	Hex. Limit	Phys. Limit	Resol.	Unit
LV_ERR_PUT_EL	V/O	0...1H	0...1	1	[-]
electrical PUT sensor error detected					
LV_CDN_DIAG_PUT_EL	V/O	0...1H	0...1	1	[-]
Status of diagnosis for electrical PUT sensor					
ERR_SYM_PUT_EL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom electrical PUT sensor					
LV_END_DIAG_PUT_EL	V/O	0...1H	0...1	1	[-]
End of pressure up throttle sensor diagnosis					

Input data:

V_PUT_MV_MAX_DIAG		V_PUT[NC_MAP_SENS_NR]	V_PUT_MV_MIN_DIAG
LV_VB_CDN_OBD_1	LV_VAR_TCHA		

FUNCTION DESCRIPTION:

General Information:

Analog input signal in the A/D-Input from the Microprocessor.

Description

Error-symtoms are defined to this diagnosis function as following :

```
□□□□
| | | └─> Circuit high input          (= SYM_0)
| | └───> Circuiit low input          (= SYM_1)
| └─────> -                           (= SYM_2)
└───────> -                           (= SYM_3)
```

Remark: Calculation of LV_END_DIAG_PUT see generic calculation “End of diagnosis” in anti bounce algorithm.

Application conditions:

Activation: LV_VAR_TCHA = 1

Deactivation: **If** LV_VAR_TCHA = 0

Then LV_END_DIAG_PUT_EL = 1

Endif

Initialization: -

Calcualtion recurrence: 10ms

Formula section:

```
if      LV_VB_CDN_OBD_1     = 1
then    LV_CDN_DIAG_PUT_EL = 1   (diagnosis is active)
else    LV_CDN_DIAG_PUT_EL = 0    (diagnosis inactive)
----
If      V_PUT > V_PUT_MV_MAX_DIAG
Then    Symptom "Circuit high input" is active, anti-bounce counter increment
        ERR_SYM_ PUT_EL = 1H
        LV_ERR_PUT_EL= 1          (after debounce)
        //Short circuit to battery or open circuit//
Else

    If      V_PUT         <        V_PUT_MV_MIN_DIAG
    Then    Symptom "Circuit low input" is active, anti-bounce counter increment   ERR_SYM_ PUT_EL = 2H
            LV_ERR_PUT_EL= 1          (after debounce)
            //Short circuit to ground//

    Else    Symptom "Circuit high input" and  "Circuit low input" are inactive, anti-bounce counter decrement
            ERR_SYM_PUT_EL = 0H
            LV_ERR_PUT_EL = 0         (after debounce)

    Endif

Endif
```

3.4 PUT sensor diagnosis (Appl. Inc.)

Output data:

Name	Mod.	Hex. Limit	Phys. Limit	Resol.	Unit
V_PUT_MV_MAX_DIAG	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold value of V_PUT_MV to detect Short circuit in signal wire to VB					
V_PUT_MV_MIN_DIAG	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold value of V_PUT_MV to detect Short circuit in signal wire to earth or wire break					
LV_PUT_DIAG_ENG_RUN	-	0...1H	0...1	1	[-]
thresholds for running engine were once written function must no longer be calculated					

Input data:

	LV_ES		LV_VAR_TCHA

FUNCTION DESCRIPTION:

General Information:

Defined the thresholds to detect either Short circuit in signal wire to battery or Short circuit in signal wire to earth or wire break. PUT = Pressure upstream throttle

Application conditions:

Activation : LV_VAR_TCHA = 1 **and** LV_PUT_DIAG_ENG_RUN = 0

Deactivation: LV_PUT_DIAG_ENG_RUN = 1

Initialization: LV_PUT_DIAG_ENG_RUN = 0 at reset

Calcualtion recurrence: 10ms

Formula section:

If LV_PUT_DIAG_ENG_RUN = 0 and LV_ES = 0
Then LV_PUT_DIAG_ENG_RUN = 1
Short circuit in signal wire to VB:
V_PUT_MV_MAX_DIAG = C_V_PUT_MV_MAX_DIAG
Short circuit in signal wire to earth or wire break:
V_PUT_MV_MIN_DIAG = C_V_PUT_MV_MIN_DIAG
else *thresholds for not running engine*

V_PUT_MV_MAX_DIAG = C_V_PUT_MV_MAX_DIAG_ES

V_PUT_MV_MIN_DIAG = C_V_PUT_MV_MIN_DIAG_ES

Endif

Error treatment:

Error debounce:

- Debounce counter increment: C_ABC_INC_PUT_EL
- Debounce counter decrement: 1
- Debounce counter maximum value: C_ABC_MAX_PUT_EL

Configuration of diagnostic syntoms :

Diagnosis	**Symptom**	**Nr**	**ABC type**
Pressure upstream throttle sensor diagnosis with short circuit signal wire to VB	Short circuit in signal wire to VB	*0*	MEM
Pressure upstream throttle sensor diagnosis short circuit in signal wire to earth or wire break	short circuit in signal wire to earth or wire break	*1*	MEM

Calibration data:

Name	**Dim**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
C_ABC_INC_PUT_EL	1	0...FFH	0...255	1	[-]
Increment debounce counter					
C_ABC_MAX_PUT_EL	1	1...FFH	1...255	1	[-]
Maximum value debounce counter					
C_V_PUT_MV_MAX_DIAG	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold value V_PUT_MV to detect Short circuit in signal wire to VB					
C_V_PUT_MV_MIN_DIAG	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold value V_PUT_MV to detect Short circuit in signal wire to earth or wire break					
C_V_PUT_MV_MAX_DIAG_ES	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold value V_PUT_MV to detect Short circuit in signal wire to VB for engine Stop					
C_V_PUT_MV_MIN_DIAG_ES	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold value V_PUT_MV to detect Short circuit in signal wire to earth or wire break for engine Stop					

3.5 Customer adaptation module: AGGR CHRG

3.5.1 Outputs for SV Aggregates

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
WGPWM[NC_CBK_EX_NR]	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
PWM output wastegate[NC_CBK_EX_NR]					
RFPPWM	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
PWM output recirculation flap					

Input data:

Tvwg	Tvwg2	Tvulv	LV_WGPWM_EXT_ADJ[NC_CBK_EX_NR]
WGPWM_EXT_ADJ[NC_CBK_EX_NR]	LV_ERR_WG_DR[NC_CBK_EX_NR]		

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: 0 at reset

Recurrence: all 10 ms

Activation: every engine state

In power latch phase:
WGPWM[NC_CBK_EX_NR] = 0
RFPPWM = *last calcuclated value*

Formula section:

```
If      LV_WGPWM_EXT_ADJ[1] = 0
Then
        IF      LV_ERR_WG_DR[1]  =  0
        Then    WGPWM[1]   = Tvwg
        Else    WGPWM[1]   = C_WGPWM _SUB_DIAG
Else
                WGPWM[1]   =  WGPWM_EXT_ADJ[1]
Endif
```

```
If          LV_WGPWM_EXT_ADJ[2] = 0
Then
        IF      LV_ERR_WG_DR[2]  =  0
        Then    WGPWM[2]    = Tvwg2
        Else    WGPWM[2]    = C_WGPWM _SUB_DIAG
Else
                WGPWM[2]    =  WGPWM_EXT_ADJ[2]
Endif
------

RFPPWM           =        Tvulv
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_WGPWM_SUB_DIAG	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
PWM output wastegate in case of electrical failure					

4 Cylinder individual lambda control

Index

4.1 Cylinder individual Lambda Control based on a WRAF Sensor

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
FAC_CYL_LAM_COR[NC_CYL_NR]	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Factor for the injection time correction of cylinder x					
MFF_ADD_CYL_LAM_COR[NC_CYL_NR]	O/V/S	8000...7FFFH	-694.51059...694.48940	0.0211948	[mg/stk]
Cylinder individual fuel mass offset correction					
MFF_ADD_CYL_LAM_COR_OUT[NC_CYL_NR]	O/V	8000...7FFFH	-694.51059...694.48940	0.0211948	[mg/stk]
Cylinder individual fuel mass offset correction - weighted output					
MFF_ADD_CYL_LAM_COR_MV[NC_CBK_EX_NR]	V	8000...7FFFH	-694.51059...694.48940	0.0211948	[mg/stk]
Mean value of the fuel mass offset correction - calculated out of the cylinder individual correction values					
MFF_SP_SAVE_LAM_CYL_SEL	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Stored fuel mass setpoint at CILC deactivation moment					
MFF_DELTA_LAM_CYL_SEL_REAC	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Absolute fuel mass difference at deactivation and reactivation interval					
FAC_VAL_LAM_CYL_SEL_REAC	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Valuated factor for internal states at reactivation moment					
FAC_CYL_LAM_INT[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual controller output - I component					
FAC_CYL_LAM_P[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual controller output - P component					
FAC_CYL_LAM_I_MV_SHIFT[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual controller output - I component, shifted by its mean value					
FAC_CYL_LAM_P_MV_SHIFT[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual controller output - P component, shifted by its mean value					
FAC_CYL_LAM[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Output of the cylinder individual controller					
FAC_CYL_LAM_ABSV_SUM[NC_CBK_EX_NR]	O/V	0...7FFFH	0...49.99847	1.5259e-3	[%]
Sum of the absolute cylinder individual controller outputs					
FAC_LAM_CYL_SEL_ADJ[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual adaptation correction factor					
FAC_LAM_CYL_SEL_ADJ_L_RNG[NC_CYL_NR]	O/V/S	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual adaptation correction factor (low-range)					
FAC_LAM_CYL_SEL_ADJ_L_RNG_MV[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Mean value of low-range adaptation - calculated out of the cylinder individual correction values					
FAC_LAM_CYL_SEL_ADJ_H_RNG[NC_CYL_NR]	O/V/S	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual adaptation correction factor (high-range)					
FAC_LAM_CYL_SEL_ADJ_H_RNG_MV[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Mean value of high-range adaptation - calculated out of the cylinder individual correction values					
FAC_LAM_CYL_SEL_ADJ_CST[NC_CYL_NR]	O/V/S	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual adaptation correction factor under engine-cold condition					
FAC_LAM_CYL_SEL_ADJ_CST_MV[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Mean value of cold-range adaptation - calculated out of the cylinder individual correction values					
FAC_LAM_CYL_SEL_ADJ_INTER[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual adaptation correction factor (cold-nominal-interpolation)					

FAC_LAM_CYL_SEL_ADJ_FIL[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual adaptation filter output					
FAC_LAM_CYL_SEL_ADJ_CST_FIL[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual adaptation filter output under engine cold condition					
DELTA_LAMB_AV[NC_CBK_EX_NR]	V	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Signal value of the WRAF Sensor, sampled with an appropriate resolution and shifted by its mean value					
DELTA_LAMB_MDL[NC_CBK_EX_NR]	V	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Estimated value of the WRAF Sensor					
DELTA_LAMB_ERR_RAW[NC_CBK_EX_NR]	V	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Difference between the measured and the estimated value of the WRAF sensor					
DELTA_LAMB_ERR[NC_CBK_EX_NR]	V	8000...7FFFH	-8...7.99975	0.2441e-3	[1/s]
Weighted estimation-error					
DELTA_LAMB_ERR_CYL_SEL[NC_CYL_NR]	V	8000...7FFFH	-8...7.99975	0.2441e-3	[1/s]
Weighted estimation-error (split)					
DELTA_LAMB_CYL_SEL[NC_CBK_EX_NR]	V	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Reassembled signal, made of individual adjusted signals					
DELTA_LAMB_DIF_CYL[NC_CYL_NR]	V	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Control error for each cylinder					
DELTA_LAMB_CYL[NC_CYL_NR]	V	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Individual adjusted signal					
LAMB_CYL_SEL[NC_CYL_NR]	V	0...7FFFH	0...7.99975	0.2441e-3	[-]
Cylinder individual lambda value					
VLS_CYL_SEL[NC_CYL_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Upstream oxygen sensor acquisition (cylinder Individual)					
DELTA_LAMB_CYL_SEL_CQ[NC_CBK_EX_NR]	O/V	0...FFFFH	0...7.99987	0.1221e-3	[-]
Characteristic value of the estimated values					
DELTA_LAMB_CYL_SEL_CQ_FIL[NC_CBK_EX_NR]	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Filter value of characteristic quantity of estimated cylinder individual lambda deviation					
LAMB_CYL_SEL_CQ_REF[NC_CBK_EX_NR]	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Reference CILC - Characteristic Quantity stored at virtual limit transition					
LAMB_CYL_SEL_CQ_DIF_REF[NC_CBK_EX_NR]	V	8000...7FFFH	-0.5...0.49998	0.0153e-3	[-]
Deviation of CILC - Characteristic Quantity from the stored reference value					
CRK_CYL_LAM[NC_CYL_NR]	O/V	0...F0H	0...1440	6	[°CRK]
Value of phase displacement					
FAC_T[NC_CBK_EX_NR]	V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Time factor of the exponential function for the sensor model					
LV_FAC_CYL_LAM_LIM_MIN[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Flag indicating that the cylinder individual lambda controller output is out of lower limit					
LV_FAC_CYL_LAM_LIM_MAX[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Flag indicating that the cylinder individual lambda controller output is out of upper limit					
LV_LAM_CYL_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Cylinder individual lambda control is active					
LV_LAM_CYL_SEL_SEG_REF_CLL[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Segment based reference flag to close the control loop					
LV_LAM_CYL_SEL_CTL_DRIFT_PRED[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Drift-away behaviour of cylinder individual lambda control predicted					
LV_LAMB_CYL_SEL_CQ_DRIFT[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Drift-away behaviour of cylinder individual lambda control detected - based on CQ value					

LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag indicating that the value of cylinder individual lambda adaptation in high-range is valid					
LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag indicating that the value of cylinder individual lambda adaptation in low-range is valid					
LV_LAM_CYL_SEL_ADJ_VLD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag indicating that the value of cylinder individual lambda adaptation is valid for the current adaptation state					
LV_MFF_ADD_CYL_LAM_COR_LIM[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag indicating that at least one of the fuel mass offset correction values is out of range					
LV_FAC_LAM_CYL_ADJ_CST_LIM[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag indicating that at least one of the cold-adaptation values is out of range					
LV_FAC_LAM_CYL_SEL_ADJ_LIM[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag indicating that at least one of the nominal-adaptation values is out of range					
LV_FAC_CYL_LAM_VIRT_LIM[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag indicating that the cylinder individual lambda control is out of virtual limit					
CTR_T_LAM_CYL_ADJ[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[s]
Timer of cylinder individual lambda control adaptation					
CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[NC_CBK_EX_NR]	O/V	0...FFFFH	0...65535	1	[-]
CILC- drift-away counter based on CQ value					
CTR_LAMB_CYL_SEL_CQ_STAB[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
CILC- stability counter based on CQ value					
CTR_LAM_CYL_SEL_REF_CLL[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Segment based reference counter to close the control loop					
STATE_LAM_CYL_SEL[NC_CBK_EX_NR]	O/V	0H 1H 2H 3H	PAS OPL WAIT CLL	1	[-]
State of cylinder individual lambda control					
STATE_LAMB_CYL_SEL_CQ_SLOP[NC_CBK_EX_NR]	O/V/S	0H 1H 2H	NEUT RISE RISE_CYC	1	[-]
Slope state of CILC - Characteristic Quantity					
STATE_LAM_CYL_SEL_ADJ[NC_CBK_EX_NR]	O/V	0H 1H 2H 3H 4H 5H 6H	PAS WAIT_ENG_COLD ADJ_ENG_COLD WAIT_ENG_NOM ADJ_NOM_L_RNG ADJ_NOM_H_RNG ADJ_NOM_OFS	1	[-]
State of cylinder individual lambda control adaptation					

Input data:

SEG_NR	N	MAF	N_32
TCO	LV_DC	NC_CYL_NR	AMP
LV_LAM_LSCL[NC_CBK_EX_NR]	MFF_SP[NC_CBK_EX_NR]	VLS_CYL[NC_CYL_NR]	
NC_CBK_EX_NR	VLS_CYL_TRIG[NC_CYL_NR]		
DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]	PHA_SHIFT_CAM_EX[NC_CBK_EX_NR]		

LV_INH_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]	LV_LAM_CYL_ENA[NC_CBK_EX_NR]
LV_LAM_CYL_SEL_CTL_FAST_REQ	LF_LS_CBK_EX_LAM_CYL_SEL_CONF
LV_LAM_CYL_SEL_ADJ_OFS_REQ[NC_CBK_EX_NR]	LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[NC_CBK_EX_NR]
STATE_DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]	MFF_SP_LAM_CYL_SEL
TEMP_LAM_CYL_SEL	LV_FAC_CYL_LAM_COR_OSC[NC_CBK_EX_NR]

DESCRIPTION:

General information:

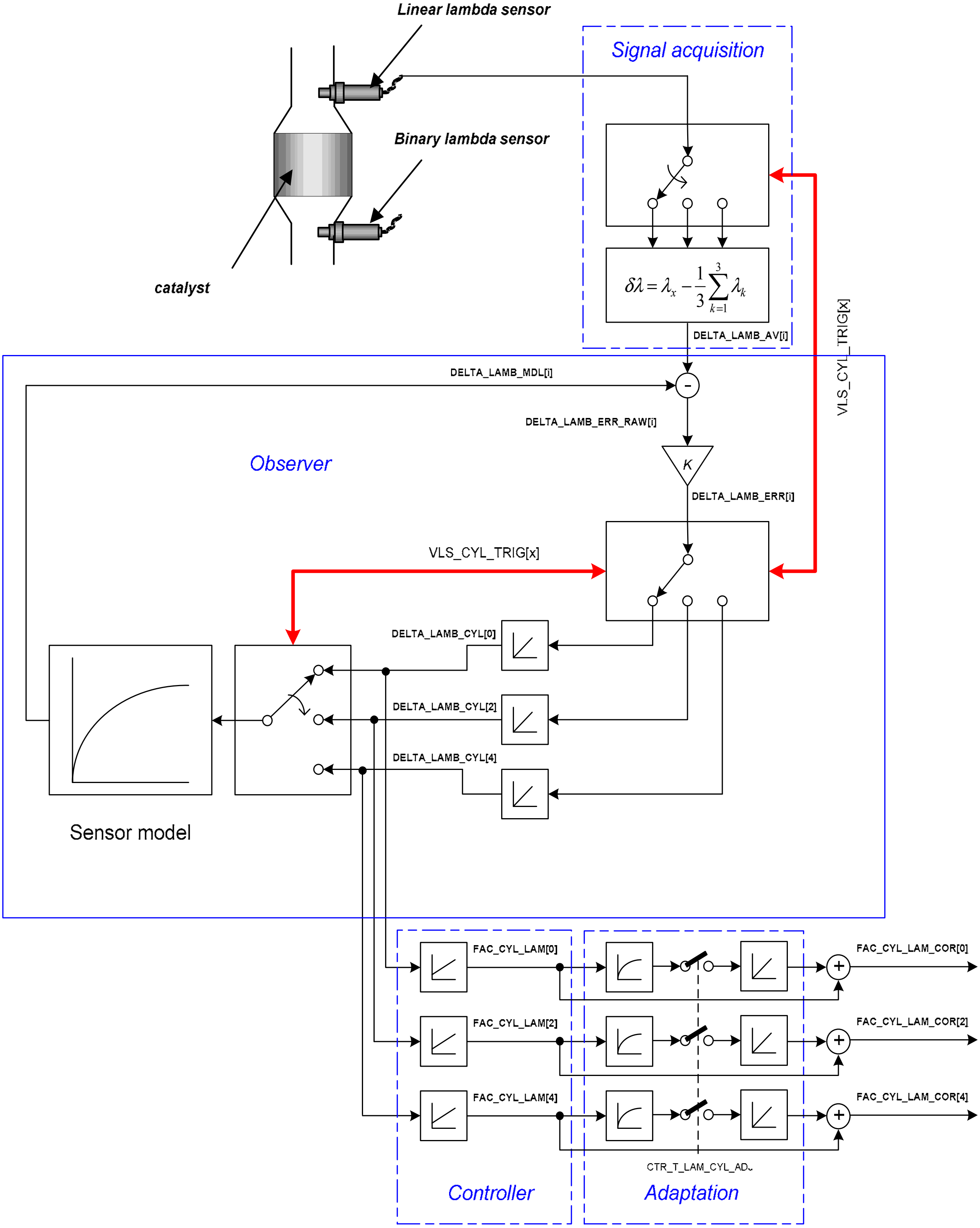
```
if(1) two separate cylinder banks are concerned
then
    i = 1, for cylinder bank 1 ( x = 0, 2, 4, ... )
    i = 2, for cylinder bank 2 ( x = 1, 3, 5, ... )
else(1)
    i = 1 ( x = 0 ... 3 or 4 )
endif(1)
```

Description of index "[i]" and "[x]":

x: logical cylinder
*use **NC_CYL_NR** : number of engine cylinders – index "[x]"*
*use **LF_LS_CBK_EX_LAM_CYL_SEL_CONF**: pattern for allocation of physical cylinders to exhaust bank – index "[i]"*

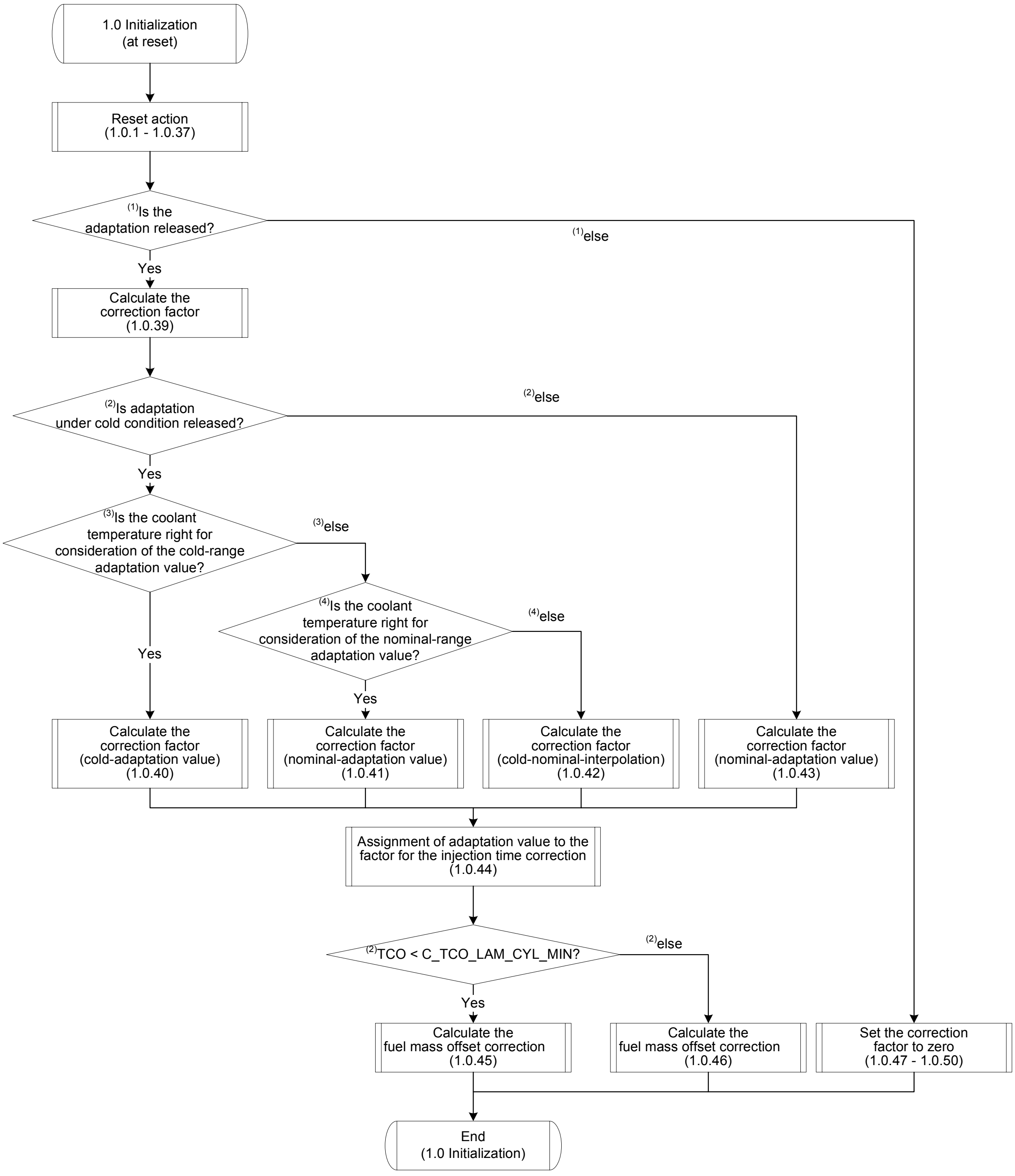
The purpose of this function is detection and control the cylinder to cylinder air-fuel ratio deviation in an internal combustion engine using an observer to estimate the deviation of each cylinder that is not measurable and a controller for each cylinder separately. In this connection the periodic signal of the WRAF sensor is reconstructed using a sensor model and adapted to the measured value. The estimated state describes the deviation in percentage for each cylinder. The mentioned value is an input to a control system that balances the air- /fuel ratio in each cylinder separately (Figure 1).

Figure 1: Structure of the cylinder individual lambda control for a 3-cylinder-exhaust-bank



Application conditions:

4.1.1 Initialization (at reset):



DELTA_LAMB_CYL_SEL_CQ[i] = 0 **(4.1.1.1)**

DELTA_LAMB_CYL_SEL_CQ_FIL[i] = 0 **(4.1.1.2)**

LAMB_CYL_SEL_CQ_REF[i] = 0 **(4.1.1.3)**

LAMB_CYL_SEL_CQ_DIF_REF[i] = 0 **(4.1.1.4)**

DELTA_LAMB_AV[i] = 0 **(4.1.1.5)**

DELTA_LAMB_ERR_RAW[i] = 0 **(4.1.1.6)**

DELTA_LAMB_ERR[i] = 0 **(4.1.1.7)**

DELTA_LAMB_ERR_CYL_SEL[x] = 0 **(4.1.1.8)**

DELTA_LAMB_CYL[x] = 0 **(4.1.1.9)**

DELTA_LAMB_CYL_SEL[i] = 0 **(4.1.1.10)**

DELTA_LAMB_MDL[i] = 0 **(4.1.1.11)**

DELTA_LAMB_DIF_CYL[x] = 0 **(4.1.1.12)**

FAC_CYL_LAM_INT[x] = 0 **(4.1.1.13)**

FAC_CYL_LAM_P[x] = 0 **(4.1.1.14)**

FAC_CYL_LAM_I_MV_SHIFT[x] = 0 **(4.1.1.15)**

FAC_CYL_LAM_P_MV_SHIFT[x] = 0 **(4.1.1.16)**

FAC_CYL_LAM[x] = 0 **(4.1.1.17)**

FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0 **(4.1.1.18)**

FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0 **(4.1.1.19)**

LV_FAC_CYL_LAM_LIM_MIN[x] = 0 **(4.1.1.20)**

LV_FAC_CYL_LAM_LIM_MAX[x] = 0 **(4.1.1.21)**

LV_LAM_CYL_SEL_SEG_REF_CLL[i] = 0

(4.1.1.22)

LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0

(4.1.1.23)

STATE_LAM_CYL_SEL[i] = 0 [PAS]

(4.1.1.24)

STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]

(4.1.1.25)

CTR_T_LAM_CYL_ADJ[i] = 0

(4.1.1.26)

MFF_ADD_CYL_LAM_COR_MV[i] = 0

(4.1.1.27)

MFF_DELTA_LAM_CYL_SEL_REAC = 0

(4.1.1.28)

FAC_VAL_LAM_CYL_SEL_REAC = 0

(4.1.1.29)

FAC_CYL_LAM_ABSV_SUM[i] = 0

(4.1.1.30)

MFF_SP_SAVE_LAM_CYL_SEL = 0

(4.1.1.31)

CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] = 0

(4.1.1.32)

LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 0

(4.1.1.33)

CTR_LAM_CYL_SEL_REF_CLL[i] = 0

(4.1.1.34)

LV_LAM_CYL_SEL_ADJ_VLD[i] = 0

(4.1.1.35)

LV_MFF_ADD_CYL_LAM_COR_LIM[i] = 0

(4.1.1.36)

LV_FAC_CYL_LAM_VIRT_LIM[i] = 0

(4.1.1.37)

CTR_LAMB_CYL_SEL_CQ_STAB[i] = 0

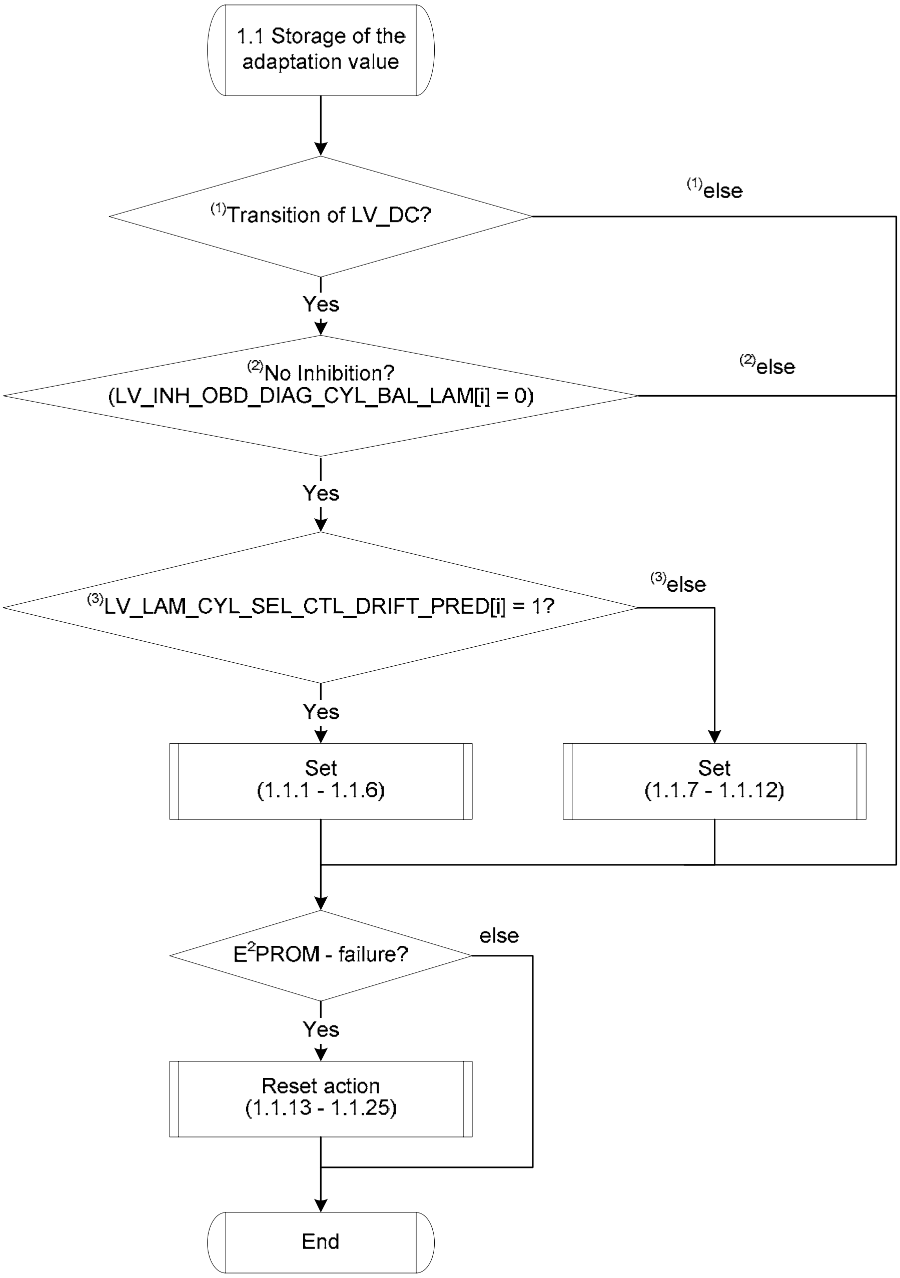
(4.1.1.38)

```
if(1) LC_ADAPT_COR_OUT = 1
then
    FAC_LAM_CYL_SEL_ADJ[x] = FAC_LAM_CYL_SEL_ADJ_L_RNG[x]
                                                                  (4.1.1.39)
    if(2)   LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 1
                [adaptation under cold condition released]
    then
            if(3)   TEMP_LAM_CYL_SEL ≤ C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG
            then
                    FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                        FAC_LAM_CYL_SEL_ADJ_CST[x] (4.1.1.40)
            else(3)
                    if(4)   TEMP_LAM_CYL_SEL ≥
                                        C_TEMP_THD_CYL_SEL_ADJ_NOM_RNG
                    then
                            FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                            FAC_LAM_CYL_SEL_ADJ[x] (4.1.1.41)
                    else(4)
```

```
                    FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                        FAC_LAM_CYL_SEL_ADJ_CST[x] +
                                                 (TEMP_LAM_CYL_SEL –
                                  C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG)*
                                              (FAC_LAM_CYL_SEL_ADJ[x] –
                                        FAC_LAM_CYL_SEL_ADJ_CST[x])/
                                  (C_TEMP_THD_CYL_SEL_ADJ_NOM_RNG –
                             C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG) (4.1.1.42)
                endif(4)
            endif(3)
        else(2)
            FAC_LAM_CYL_SEL_ADJ_INTER[x] = FAC_LAM_CYL_SEL_ADJ[x]
                                                                 (4.1.1.43)
        endif(2)

        FAC_CYL_LAM_COR[x] = FAC_LAM_CYL_SEL_ADJ_INTER[x]
                                                                 (4.1.1.44)

        if(2)   TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_MIN
        then
            MFF_ADD_CYL_LAM_COR_OUT[x] = MFF_ADD_CYL_LAM_COR[x]*
                                    IP_FAC_WUP_MFF_ADD_CYL_LAM (4.1.1.45)
        else(2)
            MFF_ADD_CYL_LAM_COR_OUT[x] = MFF_ADD_CYL_LAM_COR[x]

                                                                 (4.1.1.46)
        endif(2)
    else(1)
        FAC_LAM_CYL_SEL_ADJ[x] = 0
                                                                 (4.1.1.47)
        FAC_LAM_CYL_SEL_ADJ_INTER[x] = 0
                                                                 (4.1.1.48)
        FAC_CYL_LAM_COR[x] = 0
                                                                 (4.1.1.49)
        MFF_ADD_CYL_LAM_COR_OUT[x] = 0
                                                                 (4.1.1.50)
    endif(1)
```

4.1.2 Storage of the adaptation value:

The value of FAC_LAM_CYL_SEL_ADJ_L_RNG[x], FAC_LAM_CYL_SEL_ADJ_H_RNG[x] , FAC_LAM_CYL_SEL_ADJ_CST[x] and MFF_ADD_CYL_LAM_COR[x] shall be stored in the non-volatile memory at DC- transition (LV_DC 1 → 0). Should the value of cylinder individual injection time fall into the lower/upper limit, the adaptation value shall not be stored in the non-volatile memory. I.e., if one of the exhaust-bank-related cylinders is out of range, the adaptation values for all cylinders of the concerned exhaust bank shall be taken from the last driving cycle (LDC).



```
if$^{(1)}$ LV_DC 1 → 0
then
    if$^{(2)}$ LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 0
    then
        if$^{(3)}$ LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 1
        then
            FAC_LAM_CYL_SEL_ADJ_L_RNG[x] =
                    FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ (4.1.2.1)
            FAC_LAM_CYL_SEL_ADJ_H_RNG[x] =
                    FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ (4.1.2.2)
            FAC_LAM_CYL_SEL_ADJ_CST[x] =
                    FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$ (4.1.2.3)
            MFF_ADD_CYL_LAM_COR[x] = MFF_ADD_CYL_LAM_COR[x]$_{LDC}$
                                                            (4.1.2.4)
            LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] =
                    LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i]$_{LDC}$ (4.1.2.5)
            LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] =
                    LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i]$_{LDC}$ (4.1.2.6)
        else$^{(3)}$
            FAC_LAM_CYL_SEL_ADJ_L_RNG[x] =
                    FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{CDC}$ (4.1.2.7)
            FAC_LAM_CYL_SEL_ADJ_H_RNG[x] =
                    FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{CDC}$ (4.1.2.8)

            FAC_LAM_CYL_SEL_ADJ_CST[x] =
                    FAC_LAM_CYL_SEL_ADJ_CST[x]$_{CDC}$ (4.1.2.9)
            MFF_ADD_CYL_LAM_COR[x] = MFF_ADD_CYL_LAM_COR[x]$_{CDC}$
                                                            (4.1.2.10)
            LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] =
                    LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i]$_{CDC}$ (4.1.2.11)
            LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] =
                    LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i]$_{CDC}$ (4.1.2.12)
        endif$^{(3)}$
    endif$^{(2)}$
        [Remark: the cylinder based limitation (index x) shall be
        assigned to the corresponding exhaust bank (index i)]
endif$^{(1)}$
```

At E²PROM – failure:

FAC_LAM_CYL_SEL_ADJ_L_RNG[x] = 0

(4.1.2.13)

FAC_LAM_CYL_SEL_ADJ_H_RNG[x] = 0

(4.1.2.14)

FAC_LAM_CYL_SEL_ADJ_CST[x] = 0

(4.1.2.15)

MFF_ADD_CYL_LAM_COR[x] = 0

(4.1.2.16)

$$LV\_LAM\_CYL\_SEL\_ADJ\_H\_RNG\_VLD[i] = 0 \tag{4.1.2.17}$$

$$LV\_LAM\_CYL\_SEL\_ADJ\_L\_RNG\_VLD[i] = 0 \tag{4.1.2.18}$$

$$FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_{LDC} = 0 \tag{4.1.2.19}$$

$$FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG[x]_{LDC} = 0 \tag{4.1.2.20}$$

$$FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_{LDC} = 0 \tag{4.1.2.21}$$

$$MFF\_ADD\_CYL\_LAM\_COR[x]_{LDC} = 0 \tag{4.1.2.22}$$

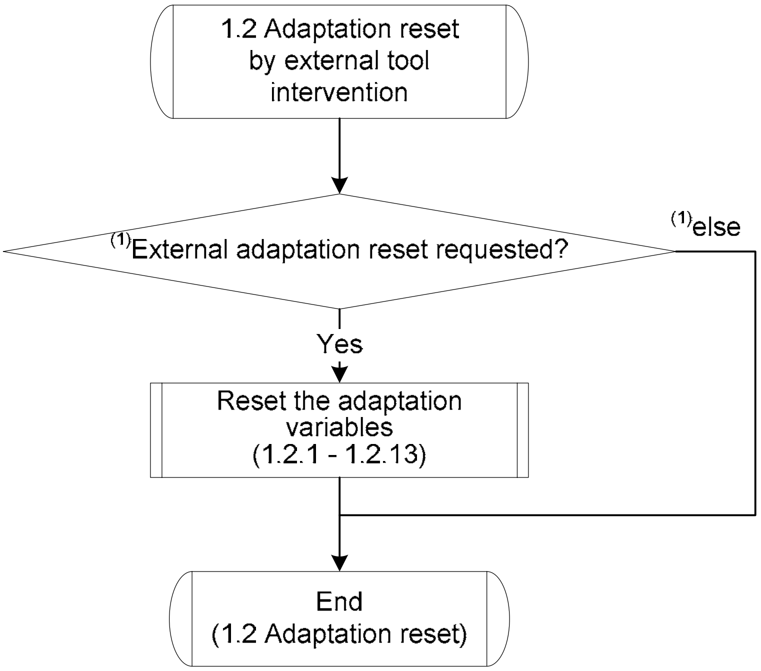
$$LV\_LAM\_CYL\_SEL\_ADJ\_H\_RNG\_VLD[i]_{LDC} = 0 \tag{4.1.2.23}$$

$$LV\_LAM\_CYL\_SEL\_ADJ\_L\_RNG\_VLD[i]_{LDC} = 0 \tag{4.1.2.24}$$

$$STATE\_LAMB\_CYL\_SEL\_CQ\_SLOP[i] = 0 \tag{4.1.2.25}$$

4.1.3 Adaptation reset by external tool intervention

Following calculation shall be carried out independently of the activation condition (time recurrence: 100 ms):



if$^{(1)}$ LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[i] = 1
then

[*Remark: the exhaust bank based calculation shall be executed for all of the exhaust-bank-related cylinders (index "[i]" -> index "[x]")*]

FAC_LAM_CYL_SEL_ADJ_L_RNG[x] = 0 **(4.1.3.1)**

FAC_LAM_CYL_SEL_ADJ_H_RNG[x] = 0 **(4.1.3.2)**

FAC_LAM_CYL_SEL_ADJ_CST[x] = 0 **(4.1.3.3)**

MFF_ADD_CYL_LAM_COR[x] = 0 **(4.1.3.4)**

LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] = 0 **(4.1.3.5)**

LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] = 0 **(4.1.3.6)**

FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ = 0 **(4.1.3.7)**

FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ = 0 **(4.1.3.8)**

FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$ = 0

(4.1.3.9)

$MFF\_ADD\_CYL\_LAM\_COR[x]_{LDC} = 0$

(4.1.3.10)

$LV\_LAM\_CYL\_SEL\_ADJ\_H\_RNG\_VLD[i]_{LDC} = 0$

(4.1.3.11)

$LV\_LAM\_CYL\_SEL\_ADJ\_L\_RNG\_VLD[i]_{LDC} = 0$

(4.1.3.12)

STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0

(4.1.3.13)

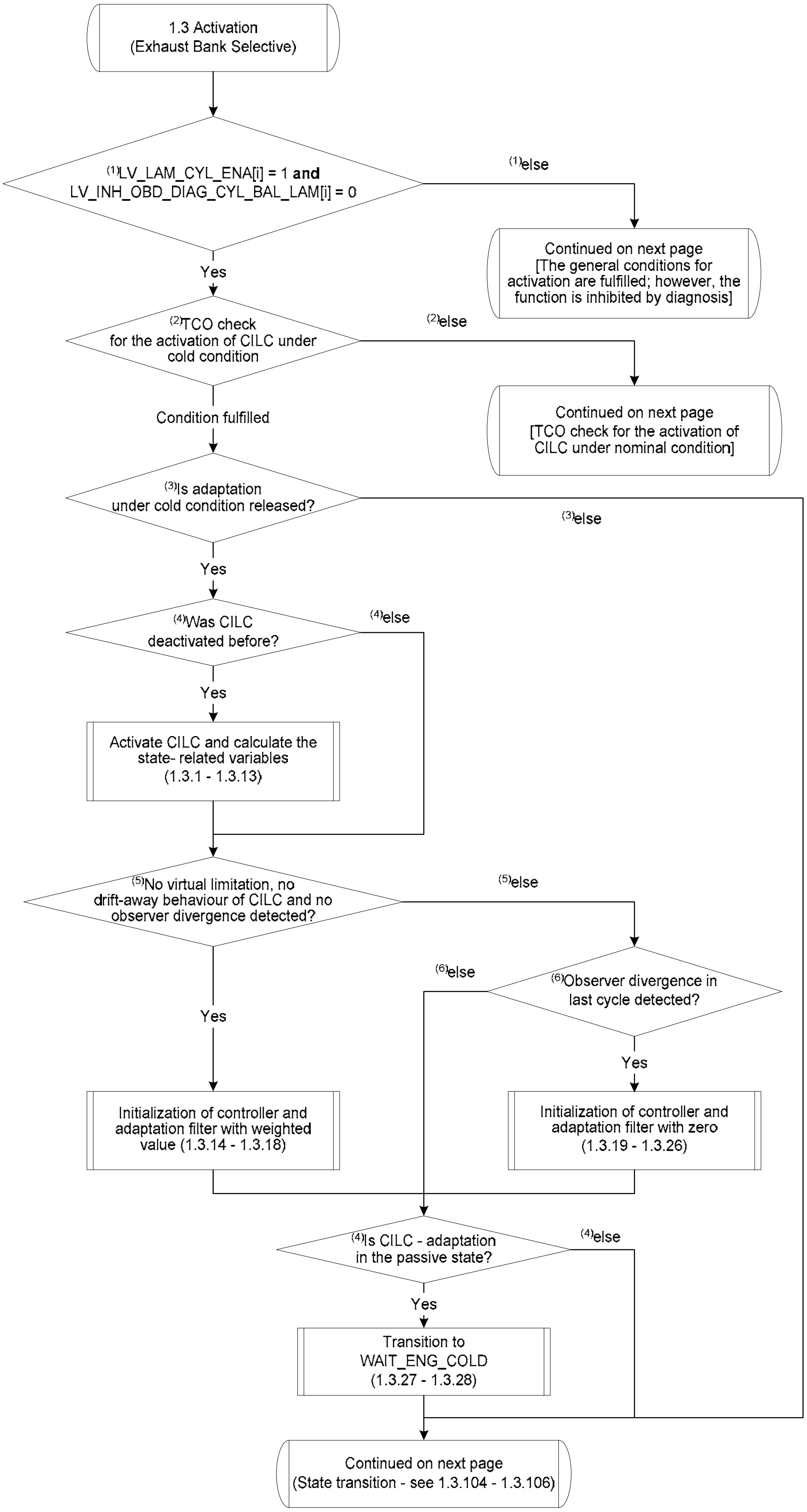
endif(1)

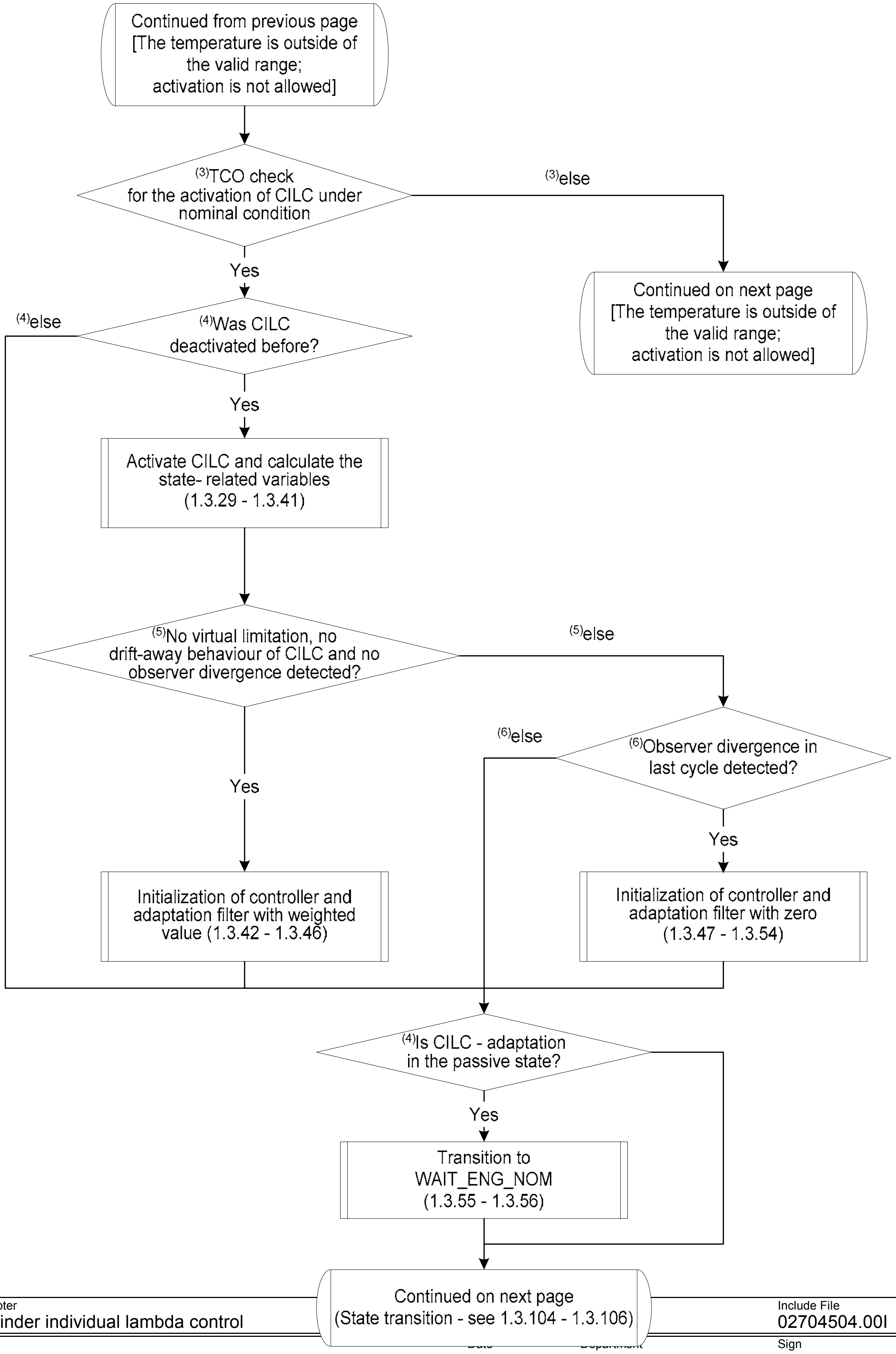
Time recurrence:

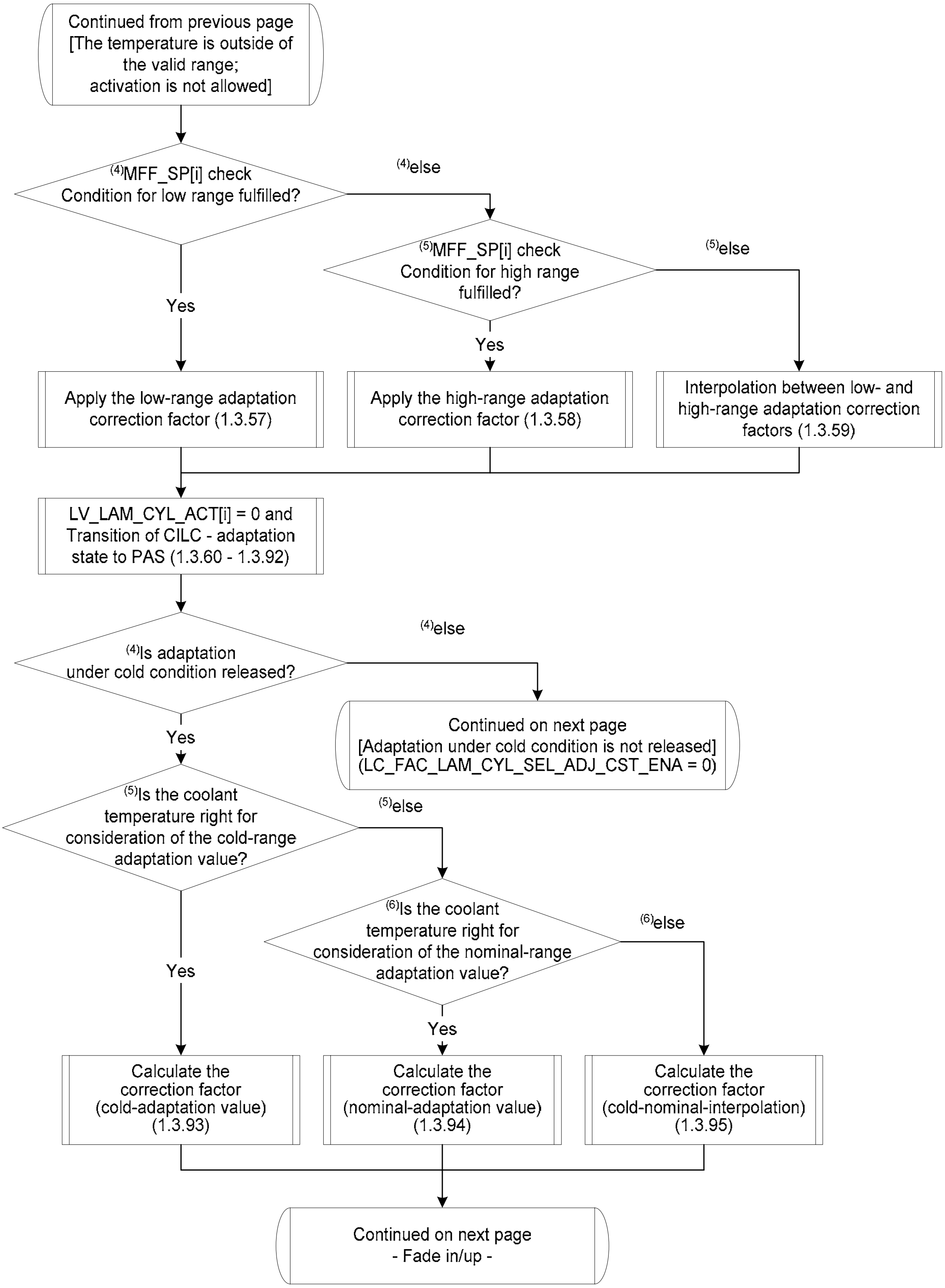
every TDC; 100 ms **(4.1.3)**; every C_T_LAM_CYL_ADJ_CST **(4.1.13.9)**;

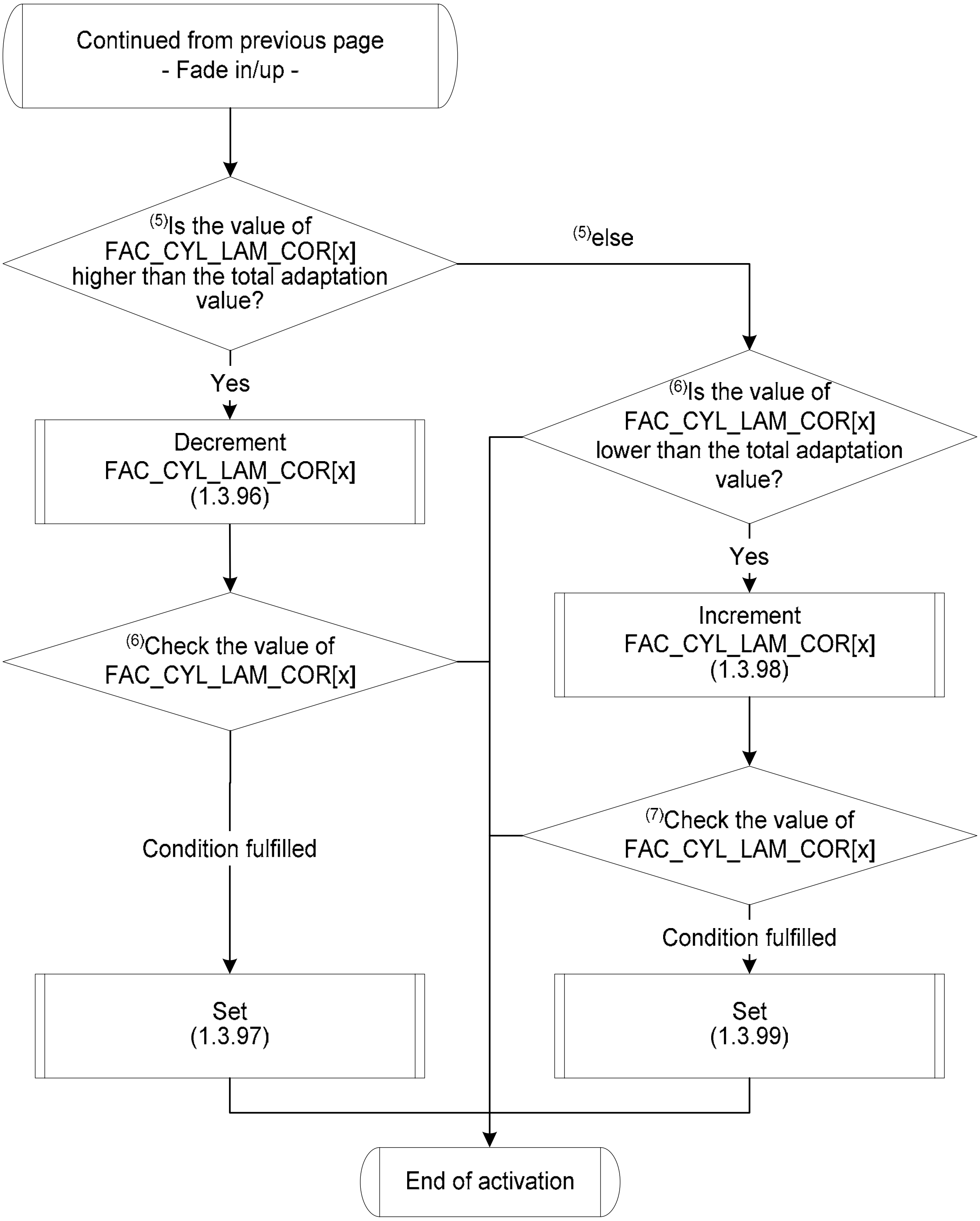
every C_T_LAM_CYL_ADJ **(4.1.13.9)**;

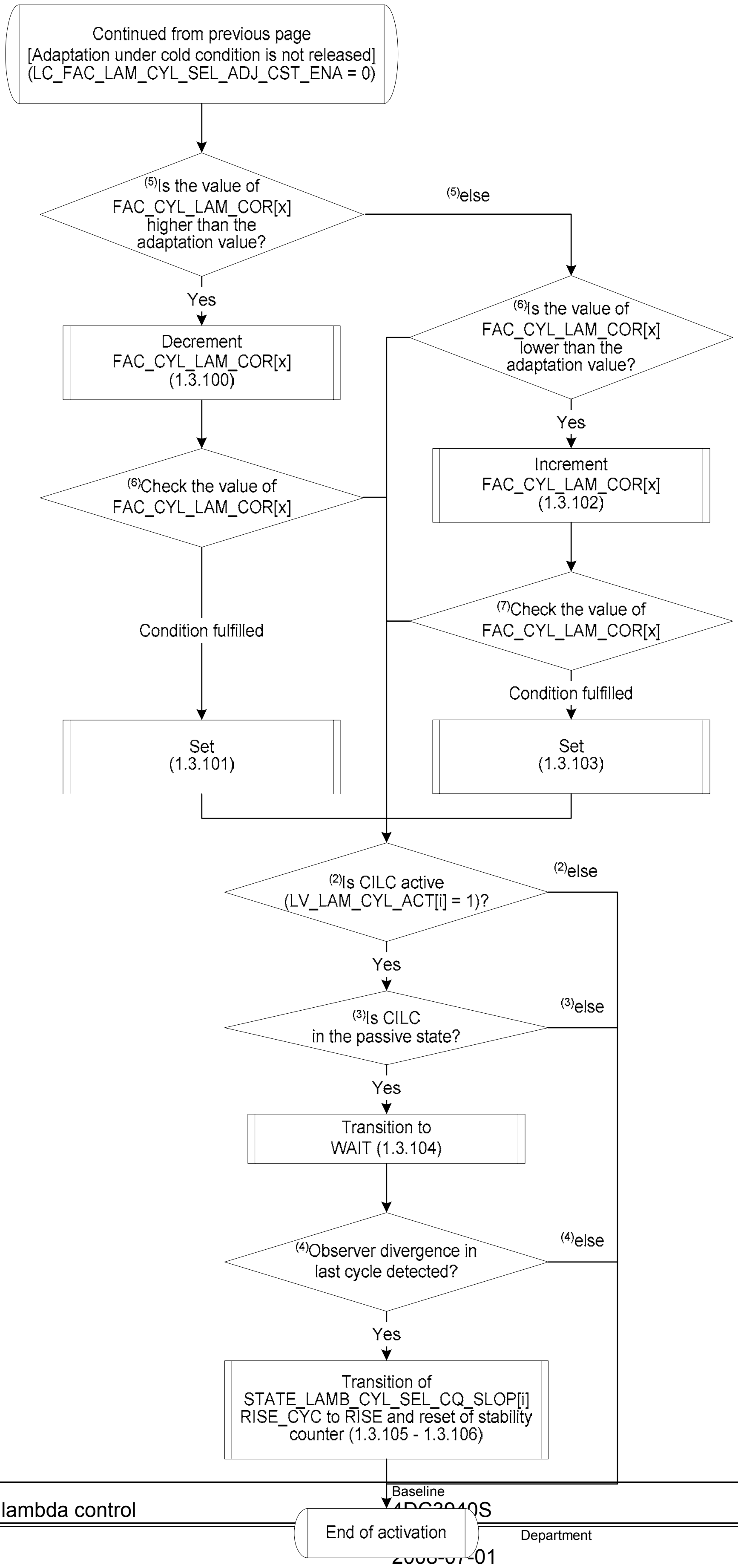
4.1.4 Activation (exhaust bank selective):

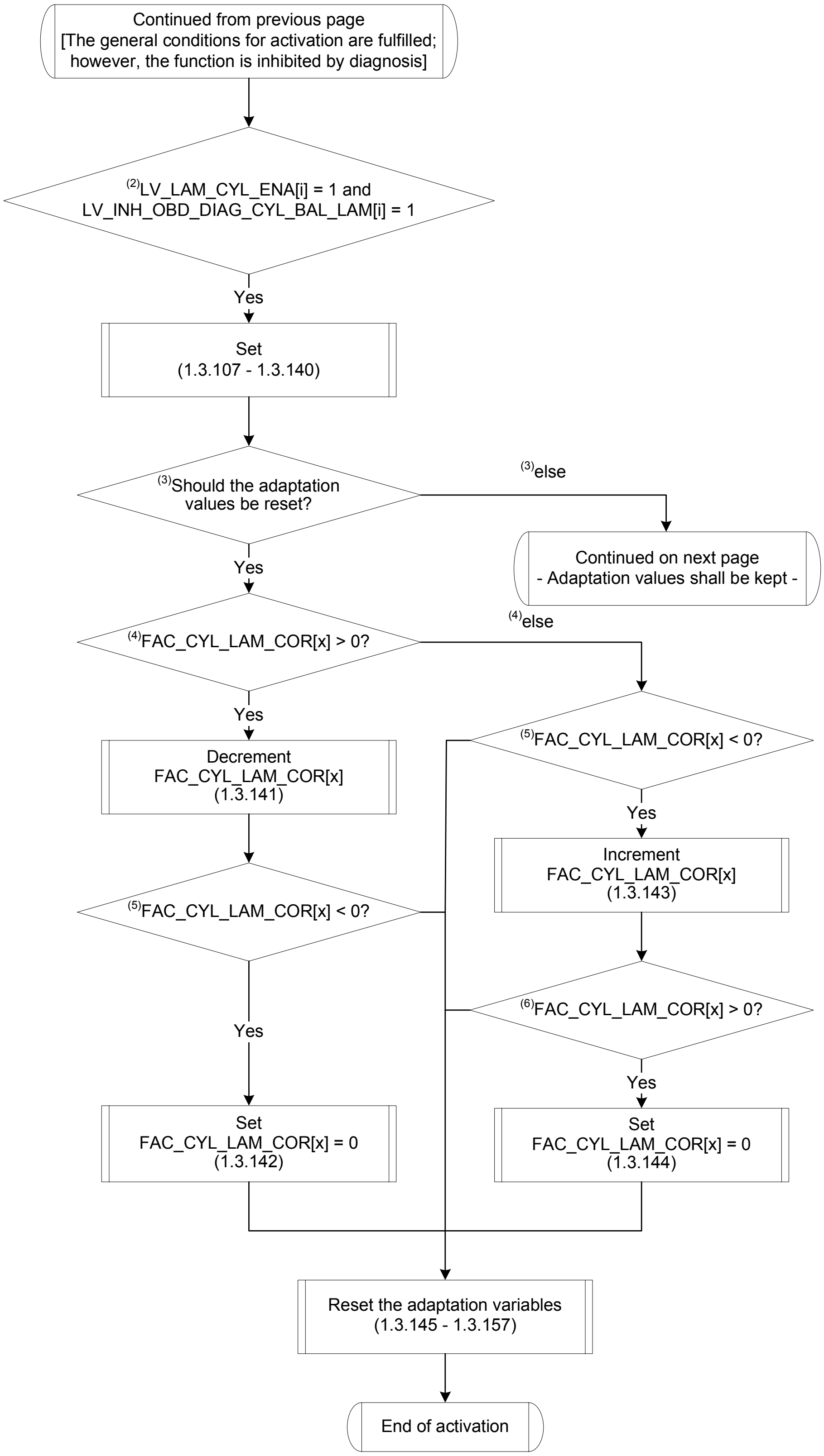

1.3 Activation
(Exhaust Bank Selective)
(1)LV_LAM_CYL_ENA[i] = 1 and
LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 0
(1)else
Continued on next page
[The general conditions for activation are fulfilled; however, the function is inhibited by diagnosis]
Yes
(2)TCO check
for the activation of CILC under
cold condition
(2)else
Continued on next page
[TCO check for the activation of CILC under nominal condition]
Condition fulfilled
(3)Is adaptation
under cold condition released?
(3)else
Yes
(4)Was CILC
deactivated before?
(4)else
Yes
Activate CILC and calculate the state- related variables
(1.3.1 - 1.3.13)
(5)No virtual limitation, no
drift-away behaviour of CILC and no
observer divergence detected?
(5)else
(6)else
(6)Observer divergence in
last cycle detected?
Yes
Yes
Initialization of controller and adaptation filter with weighted value (1.3.14 - 1.3.18)
Initialization of controller and adaptation filter with zero
(1.3.19 - 1.3.26)
(4)Is CILC - adaptation
in the passive state?
(4)else
Yes
Transition to
WAIT_ENG_COLD
(1.3.27 - 1.3.28)
Continued on next page
(State transition - see 1.3.104 - 1.3.106)

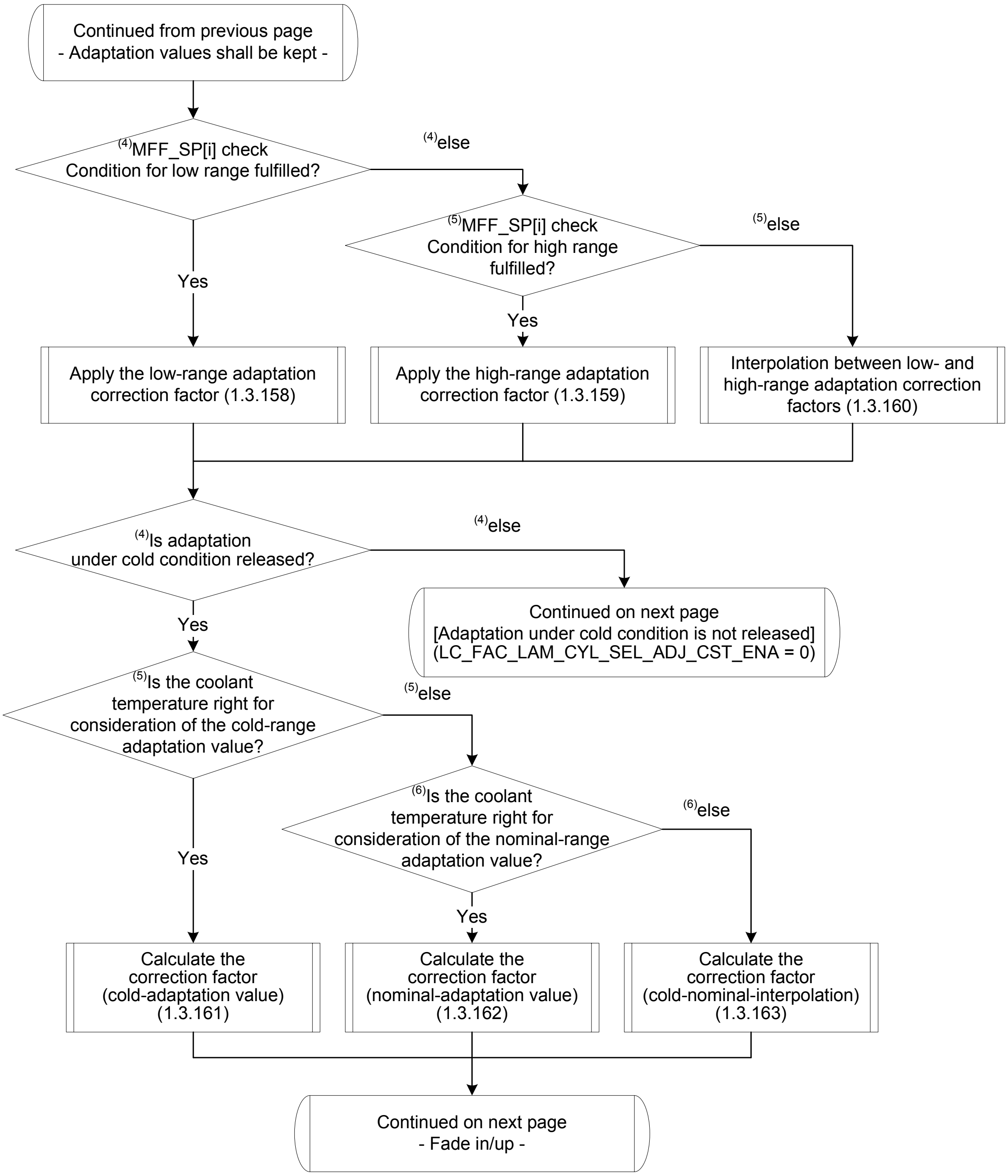

Continued from previous page [The temperature is outside of the valid range; activation is not allowed]
(3)TCO check for the activation of CILC under nominal condition
(3)else
Yes
Continued on next page [The temperature is outside of the valid range; activation is not allowed]
(4)else
(4)Was CILC deactivated before?
Yes
Activate CILC and calculate the state- related variables (1.3.29 - 1.3.41)
(5)No virtual limitation, no drift-away behaviour of CILC and no observer divergence detected?
(5)else
(6)else
(6)Observer divergence in last cycle detected?
Yes
Yes
Initialization of controller and adaptation filter with weighted value (1.3.42 - 1.3.46)
Initialization of controller and adaptation filter with zero (1.3.47 - 1.3.54)
(4)Is CILC - adaptation in the passive state?
Yes
Transition to WAIT_ENG_NOM (1.3.55 - 1.3.56)
Continued on next page (State transition - see 1.3.104 - 1.3.106)

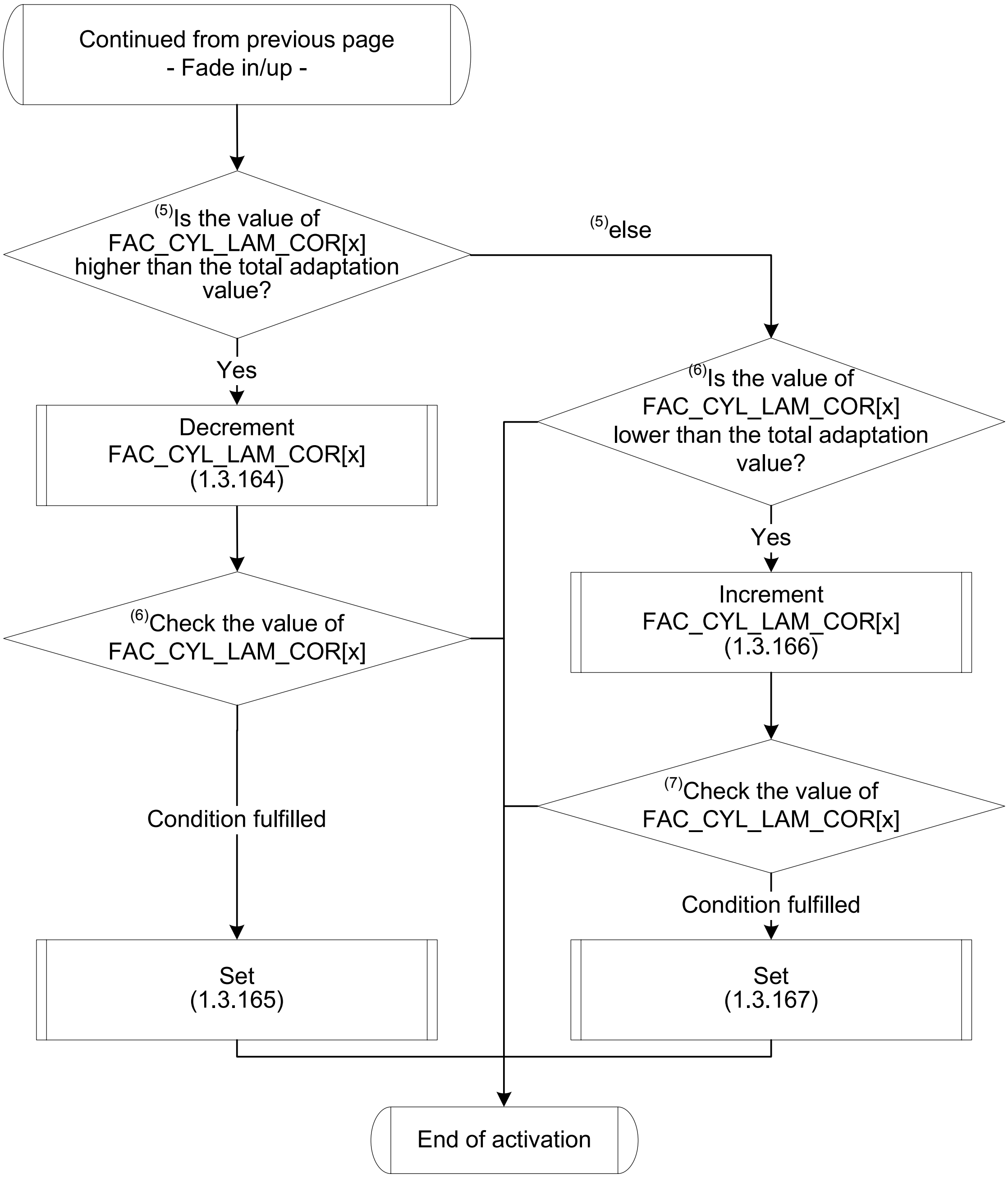

Continued from previous page
[The temperature is outside of the valid range; activation is not allowed]
(4)MFF_SP[i] check Condition for low range fulfilled?
(4)else
(5)MFF_SP[i] check Condition for high range fulfilled?
(5)else
Yes
Yes
Apply the low-range adaptation correction factor (1.3.57)
Apply the high-range adaptation correction factor (1.3.58)
Interpolation between low- and high-range adaptation correction factors (1.3.59)
LV_LAM_CYL_ACT[i] = 0 and Transition of CILC - adaptation state to PAS (1.3.60 - 1.3.92)
(4)Is adaptation under cold condition released?
(4)else
Continued on next page
[Adaptation under cold condition is not released]
(LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 0)
Yes
(5)Is the coolant temperature right for consideration of the cold-range adaptation value?
(5)else
(6)Is the coolant temperature right for consideration of the nominal-range adaptation value?
(6)else
Yes
Yes
Calculate the correction factor (cold-adaptation value) (1.3.93)
Calculate the correction factor (nominal-adaptation value) (1.3.94)
Calculate the correction factor (cold-nominal-interpolation) (1.3.95)
Continued on next page
- Fade in/up -

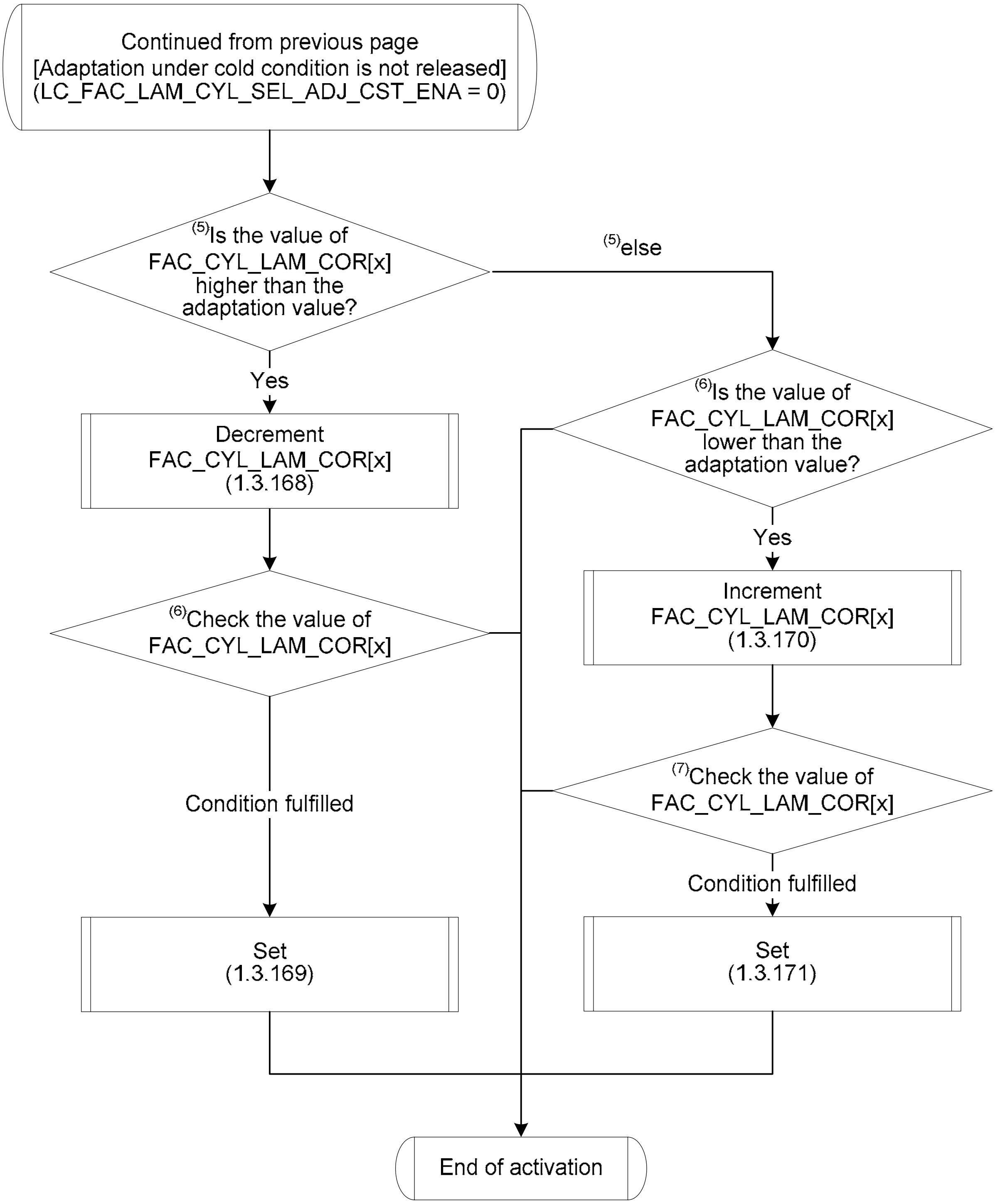

Continued from previous page
- Fade in/up -
(5)Is the value of FAC_CYL_LAM_COR[x] higher than the total adaptation value?
(5)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.3.96)
(6)Is the value of FAC_CYL_LAM_COR[x] lower than the total adaptation value?
Yes
Increment FAC_CYL_LAM_COR[x] (1.3.98)
(6)Check the value of FAC_CYL_LAM_COR[x]
(7)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
Condition fulfilled
Set (1.3.97)
Set (1.3.99)
End of activation


Continued from previous page
[Adaptation under cold condition is not released]
(LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 0)
(5)Is the value of FAC_CYL_LAM_COR[x] higher than the adaptation value?
(5)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.3.100)
(6)Is the value of FAC_CYL_LAM_COR[x] lower than the adaptation value?
Yes
Increment FAC_CYL_LAM_COR[x] (1.3.102)
(6)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
(7)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
Set (1.3.101)
Set (1.3.103)
(2)Is CILC active (LV_LAM_CYL_ACT[i] = 1)?
(2)else
Yes
(3)Is CILC in the passive state?
(3)else
Yes
Transition to WAIT (1.3.104)
(4)Observer divergence in last cycle detected?
(4)else
Yes
Transition of STATE_LAMB_CYL_SEL_CQ_SLOP[i] RISE_CYC to RISE and reset of stability counter (1.3.105 - 1.3.106)
End of activation


Continued from previous page
[The general conditions for activation are fulfilled; however, the function is inhibited by diagnosis]
(2)LV_LAM_CYL_ENA[i] = 1 and LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 1
Yes
Set (1.3.107 - 1.3.140)
(3)Should the adaptation values be reset?
(3)else
Continued on next page - Adaptation values shall be kept -
Yes
(4)FAC_CYL_LAM_COR[x] > 0?
(4)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.3.141)
(5)FAC_CYL_LAM_COR[x] < 0?
Yes
Increment FAC_CYL_LAM_COR[x] (1.3.143)
(5)FAC_CYL_LAM_COR[x] < 0?
(6)FAC_CYL_LAM_COR[x] > 0?
Yes
Yes
Set FAC_CYL_LAM_COR[x] = 0 (1.3.142)
Set FAC_CYL_LAM_COR[x] = 0 (1.3.144)
Reset the adaptation variables (1.3.145 - 1.3.157)
End of activation


Continued from previous page
- Adaptation values shall be kept -
(4)MFF_SP[i] check
Condition for low range fulfilled?
(4)else
(5)MFF_SP[i] check
Condition for high range fulfilled?
(5)else
Yes
Yes
Apply the low-range adaptation correction factor (1.3.158)
Apply the high-range adaptation correction factor (1.3.159)
Interpolation between low- and high-range adaptation correction factors (1.3.160)
(4)Is adaptation under cold condition released?
(4)else
Continued on next page
[Adaptation under cold condition is not released]
(LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 0)
Yes
(5)Is the coolant temperature right for consideration of the cold-range adaptation value?
(5)else
(6)Is the coolant temperature right for consideration of the nominal-range adaptation value?
(6)else
Yes
Yes
Calculate the correction factor (cold-adaptation value) (1.3.161)
Calculate the correction factor (nominal-adaptation value) (1.3.162)
Calculate the correction factor (cold-nominal-interpolation) (1.3.163)
Continued on next page
- Fade in/up -


Continued from previous page
- Fade in/up -
(5)Is the value of FAC_CYL_LAM_COR[x] higher than the total adaptation value?
(5)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.3.164)
(6)Is the value of FAC_CYL_LAM_COR[x] lower than the total adaptation value?
Yes
Increment FAC_CYL_LAM_COR[x] (1.3.166)
(6)Check the value of FAC_CYL_LAM_COR[x]
(7)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
Condition fulfilled
Set (1.3.165)
Set (1.3.167)
End of activation


Continued from previous page
[Adaptation under cold condition is not released]
(LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 0)
(5)Is the value of FAC_CYL_LAM_COR[x] higher than the adaptation value?
(5)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.3.168)
(6)Is the value of FAC_CYL_LAM_COR[x] lower than the adaptation value?
Yes
Increment FAC_CYL_LAM_COR[x] (1.3.170)
(6)Check the value of FAC_CYL_LAM_COR[x]
(7)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
Condition fulfilled
Set (1.3.169)
Set (1.3.171)
End of activation

```
if(1) (LV_LAM_CYL_ENA[i] = 1 and LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 0)
then
    if(2)   (C_TEMP_LAM_CYL_CST_MIN < TEMP_LAM_CYL_SEL <
                                               C_TEMP_LAM_CYL_CST_MAX and
                               TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_MIN and
                                                         LV_LAM_LSCL[i] = 1)
    [the temperature is inside of the valid range for activation under engine-cold
    condition]
    then
        if(3)   LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 1
                [adaptation under cold condition released]
        then
            if(4)   LV_LAM_CYL_ACT[i] = 0
            then
                LV_LAM_CYL_ACT[i] = 1
                                                                  (4.1.4.1)
                MFF_DELTA_LAM_CYL_SEL_REAC =
                                     abs(MFF_SP_SAVE_LAM_CYL_SEL –
                                         MFF_SP_LAM_CYL_SEL) (4.1.4.2)
                FAC_VAL_LAM_CYL_SEL_REAC =
                                     IP_FAC_VAL_LAM_CYL_SEL_REAC
                              (MFF_DELTA_LAM_CYL_SEL_REAC) (4.1.4.3)
                DELTA_LAMB_CYL_SEL_CQ[i] =
                                           DELTA_LAMB_CYL_SEL_CQ[i]*
                                FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.4)
                DELTA_LAMB_CYL_SEL_CQ_FIL[i] =
                                       DELTA_LAMB_CYL_SEL_CQ_FIL[i]*
                                FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.5)
                DELTA_LAMB_AV[i] = DELTA_LAMB_AV[i]*
                                FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.6)
                DELTA_LAMB_ERR_RAW[i] = DELTA_LAMB_ERR_RAW[i]*
                                FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.7)
                DELTA_LAMB_ERR[i] = DELTA_LAMB_ERR[i]*
                                FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.8)
                DELTA_LAMB_ERR_CYL_SEL[x] =
                                      DELTA_LAMB_ERR_CYL_SEL[x]*
                                FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.9)
                DELTA_LAMB_CYL[x] = DELTA_LAMB_CYL[x]*
                               FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.10)
                DELTA_LAMB_CYL_SEL[i] = DELTA_LAMB_CYL_SEL[i]*
                               FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.11)
                DELTA_LAMB_MDL[i] = DELTA_LAMB_MDL[i]*
                               FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.12)
                DELTA_LAMB_DIF_CYL[x] = DELTA_LAMB_DIF_CYL[x]*
```

```
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.13)
if(5)   (LV_FAC_CYL_LAM_VIRT_LIM[i] = 0 and
            LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0 and
            STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0)
then
    FAC_CYL_LAM_INT[x] = FAC_CYL_LAM_INT[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.14)
    FAC_CYL_LAM_I_MV_SHIFT[x] =
                    FAC_CYL_LAM_I_MV_SHIFT[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.15)
    FAC_CYL_LAM[x] = FAC_CYL_LAM[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.16)
    FAC_CYL_LAM_ABSV_SUM[i] =
                    FAC_CYL_LAM_ABSV_SUM[i]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.17)
    FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] =
                    FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.18)
else(5)
    if(6)   STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 2
    then
        FAC_CYL_LAM_INT[x] = 0
                                        (4.1.4.19)
        FAC_CYL_LAM_I_MV_SHIFT[x] = 0
                                        (4.1.4.20)
        FAC_CYL_LAM[x] = 0
                                        (4.1.4.21)
        FAC_CYL_LAM_ABSV_SUM[i] = 0
                                        (4.1.4.22)
        FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0
                                        (4.1.4.23)
        LV_FAC_CYL_LAM_LIM_MIN[x] = 0
                                        (4.1.4.24)
        LV_FAC_CYL_LAM_LIM_MAX[x] = 0
                                        (4.1.4.25)
        LV_FAC_CYL_LAM_VIRT_LIM[i] = 0
```

```
                                                        (4.1.4.26)
            endif$^{(5)}$
        endif$^{(4)}$
        if$^{(4)}$   STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]
        then
            STATE_LAM_CYL_SEL_ADJ[i] = 1 [WAIT_ENG_COLD]
                                                        (4.1.4.27)

            CTR_T_LAM_CYL_ADJ[i] = C_T_LAM_CYL_ADJ_CST
                                                        (4.1.4.28)
        endif$^{(4)}$
    endif$^{(3)}$
else$^{(2)}$ [the temperature is outside of the valid range for activation under engine-cold condition]
    if$^{(3)}$   TEMP_LAM_CYL_SEL $\geq$ C_TEMP_LAM_CYL_MIN
    then
        if$^{(4)}$   LV_LAM_CYL_ACT[i] = 0
        then
            LV_LAM_CYL_ACT[i] = 1
                                                        (4.1.4.29)
            MFF_DELTA_LAM_CYL_SEL_REAC =
                    abs(MFF_SP_SAVE_LAM_CYL_SEL – MFF_SP_LAM_CYL_SEL) (4.1.4.30)
            FAC_VAL_LAM_CYL_SEL_REAC =
                    IP_FAC_VAL_LAM_CYL_SEL_REAC
                    (MFF_DELTA_LAM_CYL_SEL_REAC)        (4.1.4.31)
            DELTA_LAMB_CYL_SEL_CQ[i] =
                    DELTA_LAMB_CYL_SEL_CQ[i]*
                    FAC_VAL_LAM_CYL_SEL_REAC            (4.1.4.32)
            DELTA_LAMB_CYL_SEL_CQ_FIL[i] =
                    DELTA_LAMB_CYL_SEL_CQ_FIL[i]*
                    FAC_VAL_LAM_CYL_SEL_REAC            (4.1.4.33)
            DELTA_LAMB_AV[i] = DELTA_LAMB_AV[i]*
                    FAC_VAL_LAM_CYL_SEL_REAC            (4.1.4.34)
            DELTA_LAMB_ERR_RAW[i] = DELTA_LAMB_ERR_RAW[i]*
                    FAC_VAL_LAM_CYL_SEL_REAC            (4.1.4.35)
            DELTA_LAMB_ERR[i] = DELTA_LAMB_ERR[i]*
                    FAC_VAL_LAM_CYL_SEL_REAC            (4.1.4.36)
            DELTA_LAMB_ERR_CYL_SEL[x] =
                    DELTA_LAMB_ERR_CYL_SEL[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC            (4.1.4.37)
            DELTA_LAMB_CYL[x] = DELTA_LAMB_CYL[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC            (4.1.4.38)
            DELTA_LAMB_CYL_SEL[i] = DELTA_LAMB_CYL_SEL[i]*
```

```
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.39)
DELTA_LAMB_MDL[i] = DELTA_LAMB_MDL[i]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.40)
DELTA_LAMB_DIF_CYL[x] = DELTA_LAMB_DIF_CYL[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.41)
if(5)   (LV_FAC_CYL_LAM_VIRT_LIM[i] = 0 and
             LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0 and
                  STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0)
then
        FAC_CYL_LAM_INT[x] = FAC_CYL_LAM_INT[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.42)
        FAC_CYL_LAM_I_MV_SHIFT[x] =
                          FAC_CYL_LAM_I_MV_SHIFT[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.43)
        FAC_CYL_LAM[x] = FAC_CYL_LAM[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.44)
        FAC_CYL_LAM_ABSV_SUM[i] =
                           FAC_CYL_LAM_ABSV_SUM[i]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.45)
        FAC_LAM_CYL_SEL_ADJ_FIL[x] =
                          FAC_LAM_CYL_SEL_ADJ_FIL[x]*
                    FAC_VAL_LAM_CYL_SEL_REAC (4.1.4.46)
else(5)
        if(6)   STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 2
        then
                FAC_CYL_LAM_INT[x] = 0
                                             (4.1.4.47)
                FAC_CYL_LAM_I_MV_SHIFT[x] = 0
                                             (4.1.4.48)
                FAC_CYL_LAM[x] = 0
                                             (4.1.4.49)
                FAC_CYL_LAM_ABSV_SUM[i] = 0
                                             (4.1.4.50)
                FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0
                                             (4.1.4.51)
                LV_FAC_CYL_LAM_LIM_MIN[x] = 0
```

```
                                                                        (4.1.4.52)
                              LV_FAC_CYL_LAM_LIM_MAX[x] = 0
                                                                        (4.1.4.53)
                              LV_FAC_CYL_LAM_VIRT_LIM[i] = 0
                                                                        (4.1.4.54)
                      endif(6)
                  endif(5)
              endif(4)
              if(4)   STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]
              then
                  STATE_LAM_CYL_SEL_ADJ[i] = 3 [WAIT_ENG_NOM]
                                                                        (4.1.4.55)


                  CTR_T_LAM_CYL_ADJ[i] = C_T_LAM_CYL_ADJ
                                                                        (4.1.4.56)
              endif(4)
          else(3) [the temperature is outside of the valid range; activation is not allowed]
              if(4)   MFF_SP[i] ≤ C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG
              then
                  FAC_LAM_CYL_SEL_ADJ[x] =
                            FAC_LAM_CYL_SEL_ADJ_L_RNG[x] (4.1.4.57)
              else(4)
                  if(5)   MFF_SP[i] ≥ C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG
                  then
                      FAC_LAM_CYL_SEL_ADJ[x] =
                            FAC_LAM_CYL_SEL_ADJ_H_RNG[x] (4.1.4.58)
                  else(5)
                      FAC_LAM_CYL_SEL_ADJ[x] =
                                      FAC_LAM_CYL_SEL_ADJ_L_RNG[x] +
                            (MFF_SP[i] - C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG)*
                                      (FAC_LAM_CYL_SEL_ADJ_H_RNG[x] –
                                       FAC_LAM_CYL_SEL_ADJ_L_RNG[x])/
                                (C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG –
                          C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG) (4.1.4.59)
                  endif(5)
              endif(4)
              LV_LAM_CYL_ACT[i] = 0
                                                                        (4.1.4.60)
              MFF_DELTA_LAM_CYL_SEL_REAC = 0
                                                                        (4.1.4.61)
              MFF_SP_SAVE_LAM_CYL_SEL = 0
                                                                        (4.1.4.62)
              FAC_VAL_LAM_CYL_SEL_REAC = 0
                                                                        (4.1.4.63)
              DELTA_LAMB_CYL_SEL_CQ[i] = 0
                                                                        (4.1.4.64)
```

DELTA_LAMB_CYL_SEL_CQ_FIL[i] = 0

(4.1.4.65)

LAMB_CYL_SEL_CQ_REF[i] = 0

(4.1.4.66)

LAMB_CYL_SEL_CQ_DIF_REF[i] = 0

(4.1.4.67)

CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] = 0

(4.1.4.68)

CTR_LAMB_CYL_SEL_CQ_STAB[i] = 0

(4.1.4.69)

LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 0

(4.1.4.70)

LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0

(4.1.4.71)

DELTA_LAMB_AV[i] = 0

(4.1.4.72)

DELTA_LAMB_ERR_RAW[i] = 0

(4.1.4.73)

DELTA_LAMB_ERR[i] = 0

(4.1.4.74)

DELTA_LAMB_ERR_CYL_SEL[x] = 0

(4.1.4.75)

DELTA_LAMB_CYL[x] = 0

(4.1.4.76)

DELTA_LAMB_CYL_SEL[i] = 0

(4.1.4.77)

DELTA_LAMB_MDL[i] = 0

(4.1.4.78)

DELTA_LAMB_DIF_CYL[x] = 0

(4.1.4.79)

FAC_CYL_LAM_INT[x] = 0

(4.1.4.80)

FAC_CYL_LAM_I_MV_SHIFT[x] = 0

(4.1.4.81)

FAC_CYL_LAM[x] = 0

(4.1.4.82)

FAC_CYL_LAM_ABSV_SUM[i] = 0

(4.1.4.83)

FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0

```
                                                        (4.1.4.84)
FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0
                                                        (4.1.4.85)
STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]
                                                        (4.1.4.86)
STATE_LAM_CYL_SEL[i] = 0 [PAS]
                                                        (4.1.4.87)
CTR_LAM_CYL_SEL_REF_CLL[i] = 0
                                                        (4.1.4.88)
LV_LAM_CYL_SEL_ADJ_VLD[i] = 0
                                                        (4.1.4.89)
LV_FAC_CYL_LAM_LIM_MIN[x] = 0
                                                        (4.1.4.90)
LV_FAC_CYL_LAM_LIM_MAX[x] = 0
                                                        (4.1.4.91)
LV_FAC_CYL_LAM_VIRT_LIM[i] = 0
                                                        (4.1.4.92)
if(4)   LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 1
        [adaptation under cold condition released]
then
        if(5)   TEMP_LAM_CYL_SEL ≤
                            C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG
        then
                FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                  FAC_LAM_CYL_SEL_ADJ_CST[x]
                                                        (4.1.4.93)
        else(5)
                if(6)   TEMP_LAM_CYL_SEL ≥
                             C_TEMP_THD_CYL_SEL_ADJ_NOM_RNG
                then
                        FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                      FAC_LAM_CYL_SEL_ADJ[x]
                                                        (4.1.4.94)
                else(6)
                        FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                 FAC_LAM_CYL_SEL_ADJ_CST[x] +
                                        (TEMP_LAM_CYL_SEL –
                          C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG)*
                                    (FAC_LAM_CYL_SEL_ADJ[x] –
                                FAC_LAM_CYL_SEL_ADJ_CST[x])/
                          (C_TEMP_THD_CYL_SEL_ADJ_NOM_RNG –
                 C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG) (4.1.4.95)
                endif(6)
        endif(5)
        if(5)   FAC_CYL_LAM_COR[x]_k >
                               FAC_LAM_CYL_SEL_ADJ_INTER[x]
```

```
    then
        FAC_CYL_LAM_COR[x]$_k$ = FAC_CYL_LAM_COR[x]$_{k-1}$ -
                                  C_FAC_CYL_LAM_ABSV_DEC (4.1.4.96)
        if$^{(6)}$ FAC_CYL_LAM_COR[x]$_k$ <
                                  FAC_LAM_CYL_SEL_ADJ_INTER[x]
        then
            FAC_CYL_LAM_COR[x]$_k$ =
                                  FAC_LAM_CYL_SEL_ADJ_INTER[x] (4.1.4.97)
        endif$^{(6)}$
    else$^{(5)}$
        if$^{(6)}$ FAC_CYL_LAM_COR[x]$_k$ <
                                  FAC_LAM_CYL_SEL_ADJ_INTER[x]
        then
            FAC_CYL_LAM_COR[x]$_k$ =
                                  FAC_CYL_LAM_COR[x]$_{k-1}$ +
                                  C_FAC_CYL_LAM_ABSV_DEC (4.1.4.98)
            if$^{(7)}$ FAC_CYL_LAM_COR[x]$_k$ >
                                  FAC_LAM_CYL_SEL_ADJ_INTER[x]
            then
                FAC_CYL_LAM_COR[x]$_k$ =
                                  FAC_LAM_CYL_SEL_ADJ_INTER[x] (4.1.4.99)
            endif$^{(7)}$
        endif$^{(6)}$
    endif$^{(5)}$
else$^{(4)}$ [adaptation under cold condition not released]
    if$^{(5)}$ FAC_CYL_LAM_COR[x]$_k$ > FAC_LAM_CYL_SEL_ADJ[x]
    then
        FAC_CYL_LAM_COR[x]$_k$ = FAC_CYL_LAM_COR[x]$_{k-1}$ –
                                  C_FAC_CYL_LAM_ABSV_DEC (4.1.4.100)
        if$^{(6)}$ FAC_CYL_LAM_COR[x]$_k$ <
                                  FAC_LAM_CYL_SEL_ADJ[x]
        then
            FAC_CYL_LAM_COR[x]$_k$ =
                                  FAC_LAM_CYL_SEL_ADJ[x] (4.1.4.101)
        endif$^{(6)}$
    else$^{(5)}$
        if$^{(6)}$ FAC_CYL_LAM_COR[x]$_k$ <
                                  FAC_LAM_CYL_SEL_ADJ[x]
        then
            FAC_CYL_LAM_COR[x]$_k$ =
                                  FAC_CYL_LAM_COR[x]$_{k-1}$ +
                                  C_FAC_CYL_LAM_ABSV_DEC (4.1.4.102)
            if$^{(7)}$ FAC_CYL_LAM_COR[x]$_k$ >
                                  FAC_LAM_CYL_SEL_ADJ[x]
            then
                FAC_CYL_LAM_COR[x]$_k$ =
                                  FAC_LAM_CYL_SEL_ADJ[x] (4.1.4.103)
            endif$^{(7)}$
        endif$^{(6)}$
```

```
                endif(5)
            endif(4)
        endif(3)
    endif(2)
    if(2)   LV_LAM_CYL_ACT[i] = 1
    then
        if(3)   STATE_LAM_CYL_SEL[i] = 0 [PAS]
        then
            STATE_LAM_CYL_SEL[i] = 2 [WAIT]
                                                                  (4.1.4.104)
            if(4)   STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 2
            then
                STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 1
                                                                  (4.1.4.105)
                CTR_LAMB_CYL_SEL_CQ_STAB[i] = 0
                                                                  (4.1.4.106)
            endif(4)
        endif(3)
    endif(2)
else(1)     [the general conditions for activation are fulfilled; however, the function is
            inhibited by diagnosis]
    if(2)   (LV_LAM_CYL_ENA[i] = 1 and LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 1)
    then
        LV_LAM_CYL_ACT[i] = 0
                                                                  (4.1.4.107)
        LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0
                                                                  (4.1.4.108)
        DELTA_LAMB_CYL_SEL_CQ[i] = 0
                                                                  (4.1.4.109)
        DELTA_LAMB_CYL_SEL_CQ_FIL[i] = 0
                                                                  (4.1.4.110)
        LAMB_CYL_SEL_CQ_REF[i] = 0
                                                                  (4.1.4.111)
        LAMB_CYL_SEL_CQ_DIF_REF[i] = 0
                                                                  (4.1.4.112)
        CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] = 0
                                                                  (4.1.4.113)
        CTR_LAMB_CYL_SEL_CQ_STAB[i] = 0
                                                                  (4.1.4.114)
        LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 0
                                                                  (4.1.4.115)
        DELTA_LAMB_AV[i] = 0
                                                                  (4.1.4.116)
        DELTA_LAMB_ERR_RAW[i]          = 0
                                                                  (4.1.4.117)
```

DELTA_LAMB_ERR[i] = 0

(4.1.4.118)

DELTA_LAMB_ERR_CYL_SEL[x] = 0

(4.1.4.119)

DELTA_LAMB_CYL[x] = 0

(4.1.4.120)

DELTA_LAMB_CYL_SEL[i] = 0

(4.1.4.121)

DELTA_LAMB_MDL[i] = 0

(4.1.4.122)

DELTA_LAMB_DIF_CYL[x] = 0

(4.1.4.123)

FAC_CYL_LAM_INT[x] = 0

(4.1.4.124)

FAC_CYL_LAM_P[x] = 0

(4.1.4.125)

FAC_CYL_LAM_I_MV_SHIFT[x] = 0

(4.1.4.126)

FAC_CYL_LAM_P_MV_SHIFT[x] = 0

(4.1.4.127)

FAC_CYL_LAM[x] = 0

(4.1.4.128)

FAC_CYL_LAM_ABSV_SUM[i] = 0

(4.1.4.129)

FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0

(4.1.4.130)

FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0

(4.1.4.131)

LV_LAM_CYL_SEL_SEG_REF_CLL[i] = 0

(4.1.4.132)

STATE_LAM_CYL_SEL[i] = 0 [PAS]

(4.1.4.133)

STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]

(4.1.4.134)

CTR_T_LAM_CYL_ADJ[i] = 0

(4.1.4.135)

CTR_LAM_CYL_SEL_REF_CLL[i] = 0

(4.1.4.136)

LV_LAM_CYL_SEL_ADJ_VLD[i] = 0

(4.1.4.137)

LV_FAC_CYL_LAM_LIM_MIN[x] = 0

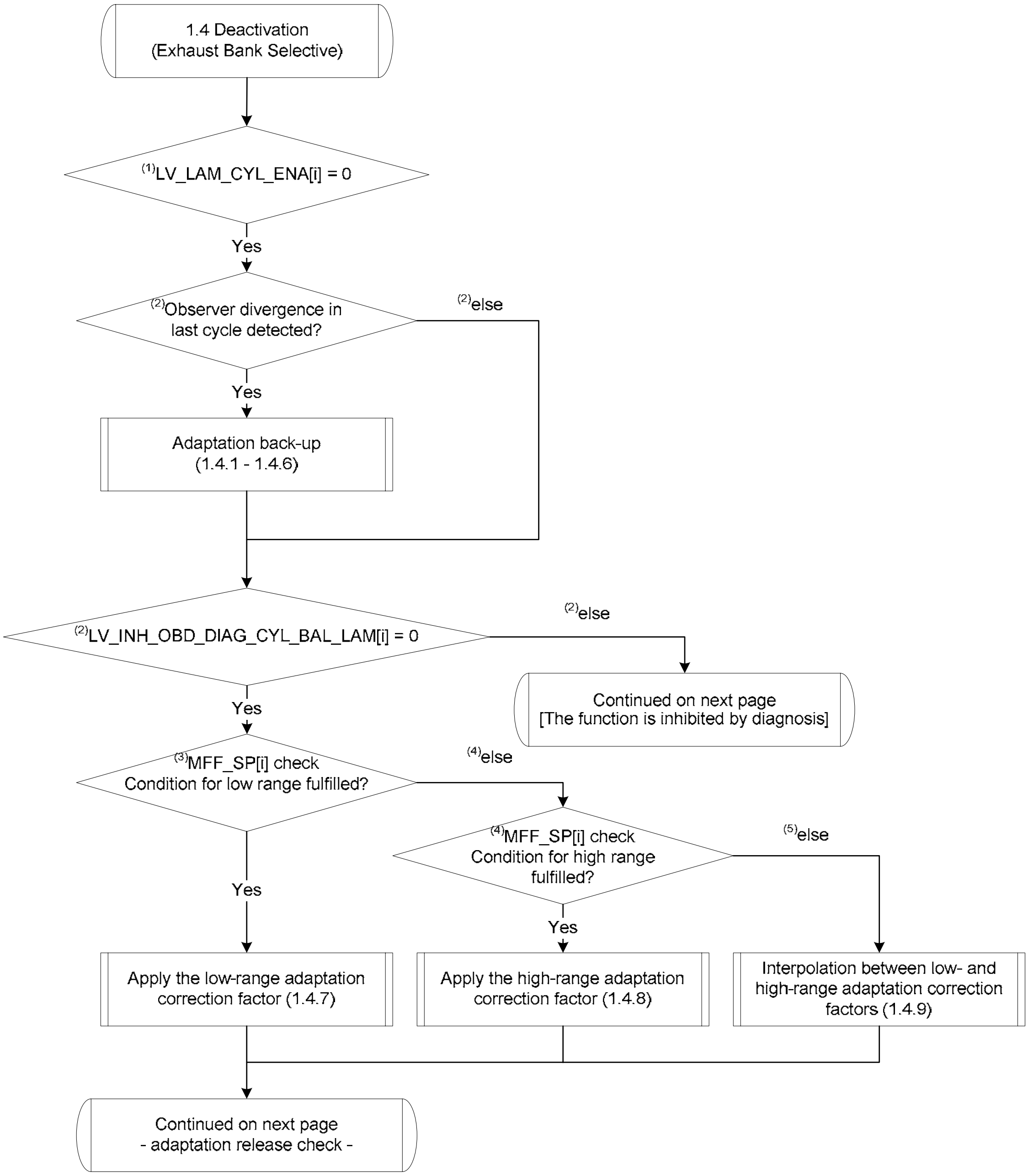
```
                                                              (4.1.4.138)
LV_FAC_CYL_LAM_LIM_MAX[x] = 0
                                                              (4.1.4.139)
LV_FAC_CYL_LAM_VIRT_LIM[i] = 0
                                                              (4.1.4.140)
if$^{(3)}$    LC_LAM_CYL_SEL_ERR_RST = 1
       [adaptation values shall be reset]
then
       if$^{(4)}$    $FAC\_CYL\_LAM\_COR[x]_k > 0$
       then
              $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1}$ -
                               C_FAC_CYL_LAM_ABSV_DEC (4.1.4.141)
              if$^{(5)}$    $FAC\_CYL\_LAM\_COR[x]_k < 0$
              then
                     $FAC\_CYL\_LAM\_COR[x]_k = 0$
                                                              (4.1.4.142)
              endif$^{(5)}$
       else$^{(4)}$
              if$^{(5)}$    $FAC\_CYL\_LAM\_COR[x]_k < 0$
              then
                     $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1}$ +
                               C_FAC_CYL_LAM_ABSV_DEC (4.1.4.143)
                     if$^{(6)}$    $FAC\_CYL\_LAM\_COR[x]_k > 0$
                     then
                            $FAC\_CYL\_LAM\_COR[x]_k = 0$
                                                              (4.1.4.144)
                     endif$^{(6)}$
              endif$^{(5)}$
       endif$^{(4)}$
       $FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_{LDC} = 0$
                                                              (4.1.4.145)
       $FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG[x]_{LDC} = 0$
                                                              (4.1.4.146)
       $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_{LDC} = 0$
                                                              (4.1.4.147)
       $MFF\_ADD\_CYL\_LAM\_COR[x]_{LDC} = 0$
                                                              (4.1.4.148)
       $LV\_LAM\_CYL\_SEL\_ADJ\_H\_RNG\_VLD[i]_{LDC} = 0$
                                                              (4.1.4.149)
       $LV\_LAM\_CYL\_SEL\_ADJ\_L\_RNG\_VLD[i]_{LDC} = 0$
                                                              (4.1.4.150)
       FAC_LAM_CYL_SEL_ADJ[x] = 0
                                                              (4.1.4.151)
       FAC_LAM_CYL_SEL_ADJ_L_RNG[x] = 0
                                                              (4.1.4.152)
       FAC_LAM_CYL_SEL_ADJ_H_RNG[x] = 0
```

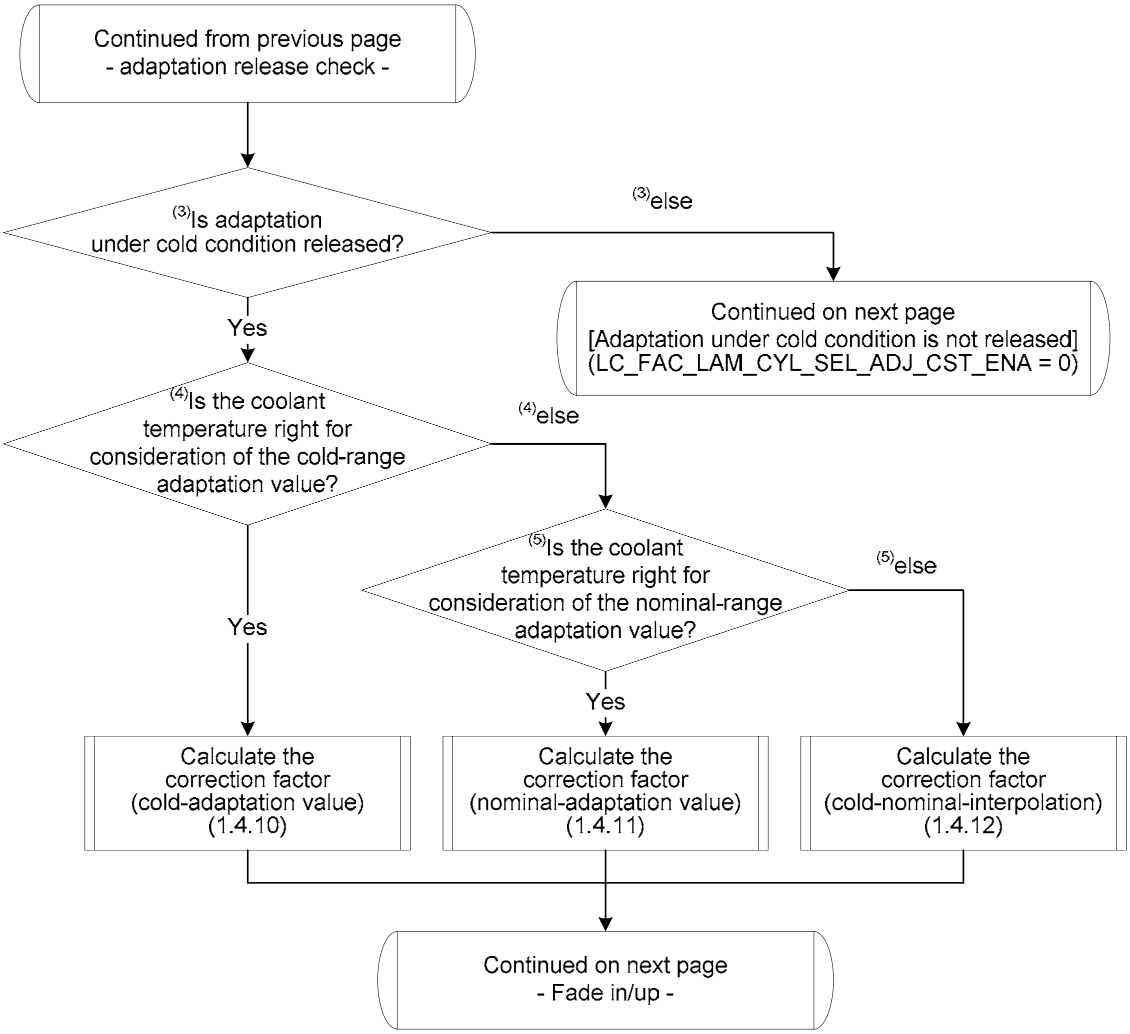
```
                                                          (4.1.4.153)
    FAC_LAM_CYL_SEL_ADJ_CST[x] = 0
                                                          (4.1.4.154)
    MFF_ADD_CYL_LAM_COR[x] = 0
                                                          (4.1.4.155)
    LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] = 0
                                                          (4.1.4.156)
    LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] = 0
                                                          (4.1.4.157)
else$^{(3)}$ [adaptation values shall be kept]
    if$^{(4)}$ $MFF\_SP[i] \leq C\_MFF\_SP\_THD\_CYL\_SEL\_ADJ\_L\_RNG$
    then
        FAC_LAM_CYL_SEL_ADJ[x] =
                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ (4.1.4.158)
    else$^{(4)}$
        if$^{(5)}$ $MFF\_SP[i] \geq C\_MFF\_SP\_THD\_CYL\_SEL\_ADJ\_H\_RNG$
        then
            FAC_LAM_CYL_SEL_ADJ[x] =
                FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ (4.1.4.159)
        else$^{(5)}$
            FAC_LAM_CYL_SEL_ADJ[x] =
                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ +
                (MFF_SP[i] - C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG)*
                (FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ –
                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$)/
                (C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG –
                C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG) (4.1.4.160)
        endif$^{(5)}$
    endif$^{(4)}$
    if$^{(4)}$ LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 1
        [adaptation under cold condition released]
    then
        if$^{(5)}$ $TEMP\_LAM\_CYL\_SEL \leq$
                $C\_TEMP\_THD\_CYL\_SEL\_ADJ\_COLD\_RNG$
        then
            FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$
                                                          (4.1.4.161)
        else$^{(5)}$
            if$^{(6)}$ $TEMP\_LAM\_CYL\_SEL \geq$
                    $C\_TEMP\_THD\_CYL\_SEL\_ADJ\_NOM\_RNG$
            then
                FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                    FAC_LAM_CYL_SEL_ADJ[x]
                                                          (4.1.4.162)
            else$^{(6)}$
                FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                    FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$ +
                    (TEMP_LAM_CYL_SEL –
                    C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG)*
```

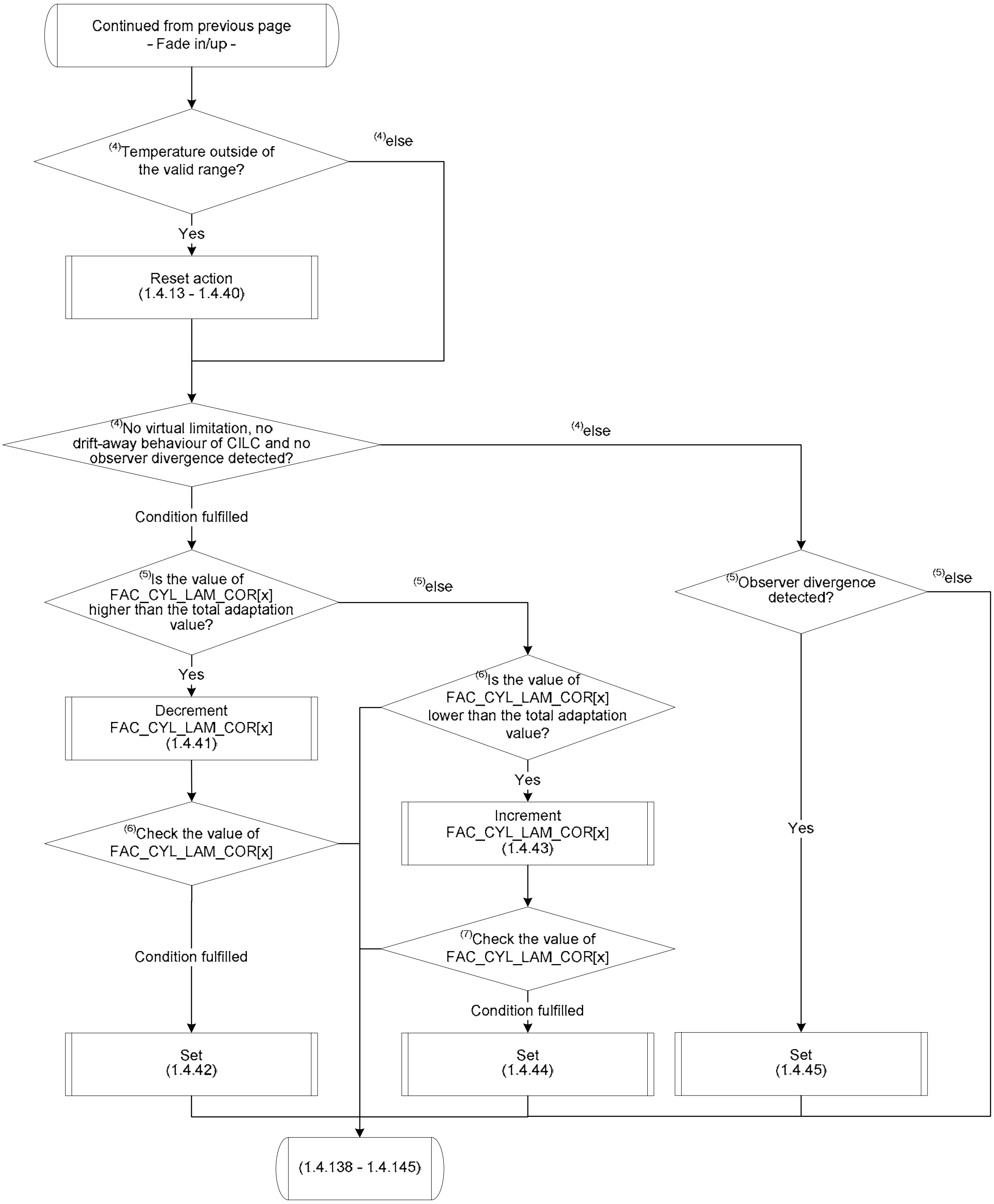
```
            $(FAC\_LAM\_CYL\_SEL\_ADJ[x] - FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_{LDC}) / (C\_TEMP\_THD\_CYL\_SEL\_ADJ\_NOM\_RNG - C\_TEMP\_THD\_CYL\_SEL\_ADJ\_COLD\_RNG)$ (4.1.4.163)
        endif$^{(6)}$
    endif$^{(5)}$
    if$^{(5)}$ $FAC\_CYL\_LAM\_COR[x]_k > FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$
    then
        $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1} - C\_FAC\_CYL\_LAM\_ABSV\_DEC$ (4.1.4.164)
        if$^{(6)}$ $FAC\_CYL\_LAM\_COR[x]_k < FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$
        then
            $FAC\_CYL\_LAM\_COR[x]_k = FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$ (4.1.4.165)
        endif$^{(6)}$
    else$^{(5)}$
        if$^{(6)}$ $FAC\_CYL\_LAM\_COR[x]_k < FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$
        then
            $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1} + C\_FAC\_CYL\_LAM\_ABSV\_DEC$ (4.1.4.166)
            if$^{(7)}$ $FAC\_CYL\_LAM\_COR[x]_k > FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$
            then
                $FAC\_CYL\_LAM\_COR[x]_k = FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$ (4.1.4.167)
            endif$^{(7)}$
        endif$^{(6)}$
    endif$^{(5)}$
else$^{(4)}$ [adaptation under cold condition not released]
    if$^{(5)}$ $FAC\_CYL\_LAM\_COR[x]_k > FAC\_LAM\_CYL\_SEL\_ADJ[x]$
    then
        $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1} - C\_FAC\_CYL\_LAM\_ABSV\_DEC$ (4.1.4.168)
        if$^{(6)}$ $FAC\_CYL\_LAM\_COR[x]_k < FAC\_LAM\_CYL\_SEL\_ADJ[x]$
        then
            $FAC\_CYL\_LAM\_COR[x]_k = FAC\_LAM\_CYL\_SEL\_ADJ[x]$ (4.1.4.169)
        endif$^{(6)}$
    else$^{(5)}$
        if$^{(6)}$ $FAC\_CYL\_LAM\_COR[x]_k < FAC\_LAM\_CYL\_SEL\_ADJ[x]$
        then
            $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1} +$
```

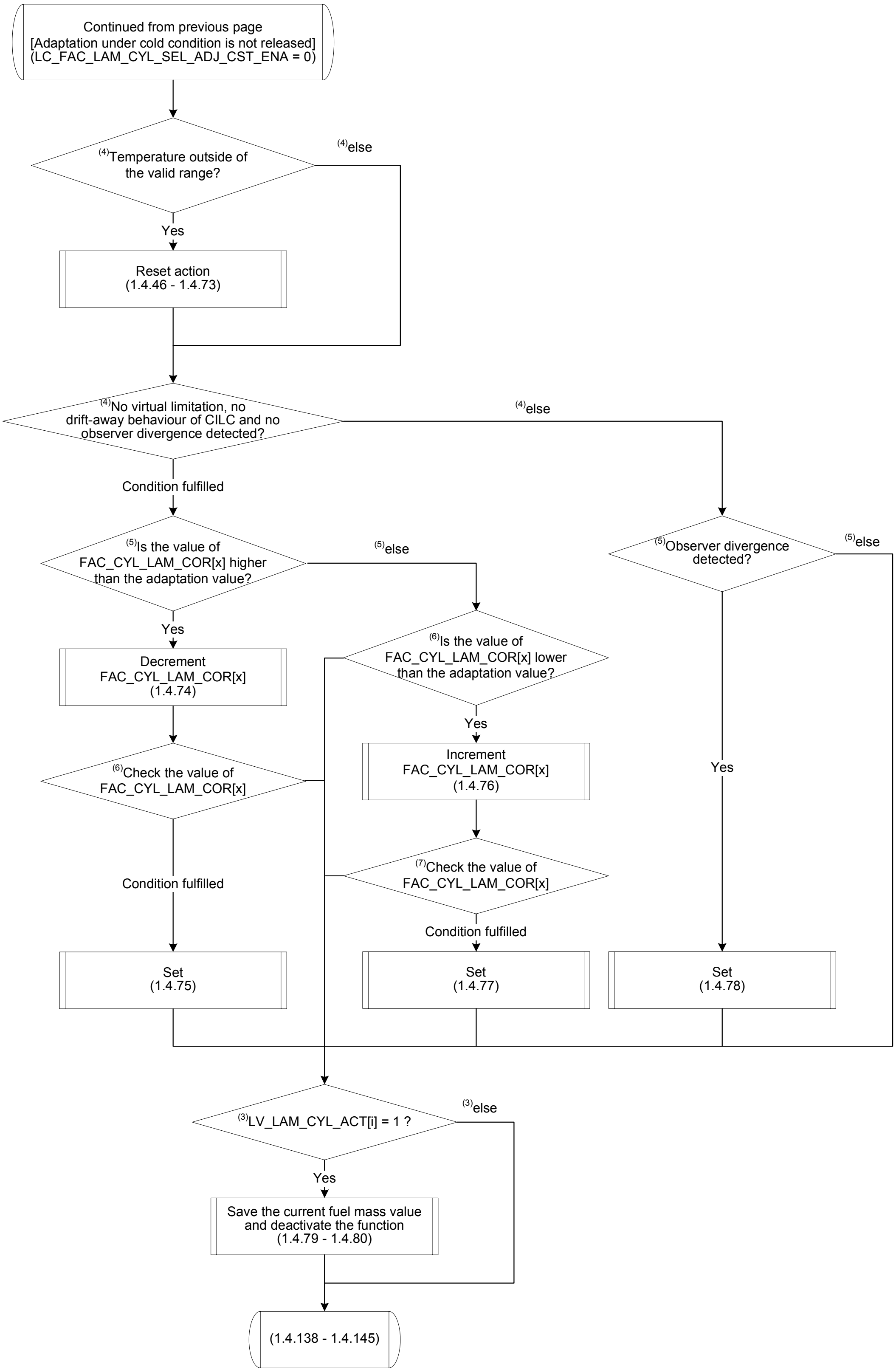
```
                  C_FAC_CYL_LAM_ABSV_DEC (4.1.4.170)
            if(7)  FAC_CYL_LAM_COR[x]_k >
                         FAC_LAM_CYL_SEL_ADJ[x]
            then
                   FAC_CYL_LAM_COR[x]_k =
                     FAC_LAM_CYL_SEL_ADJ[x] (4.1.4.171)
            endif(7)
          endif(6)
        endif(5)
      endif(4)
    endif(3)
  endif(2)
endif(1)
```

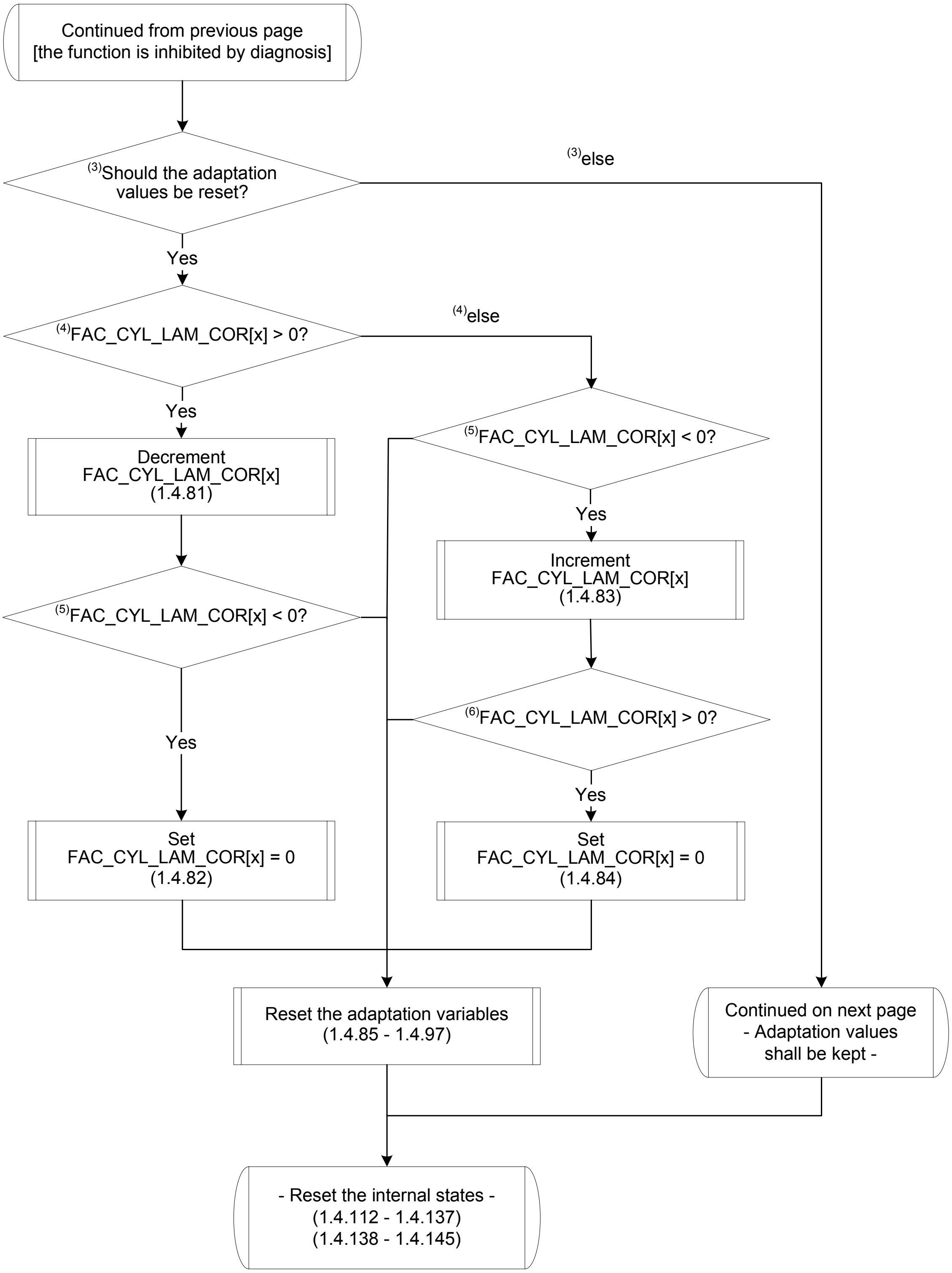
4.1.5 Deactivation (exhaust bank selective):

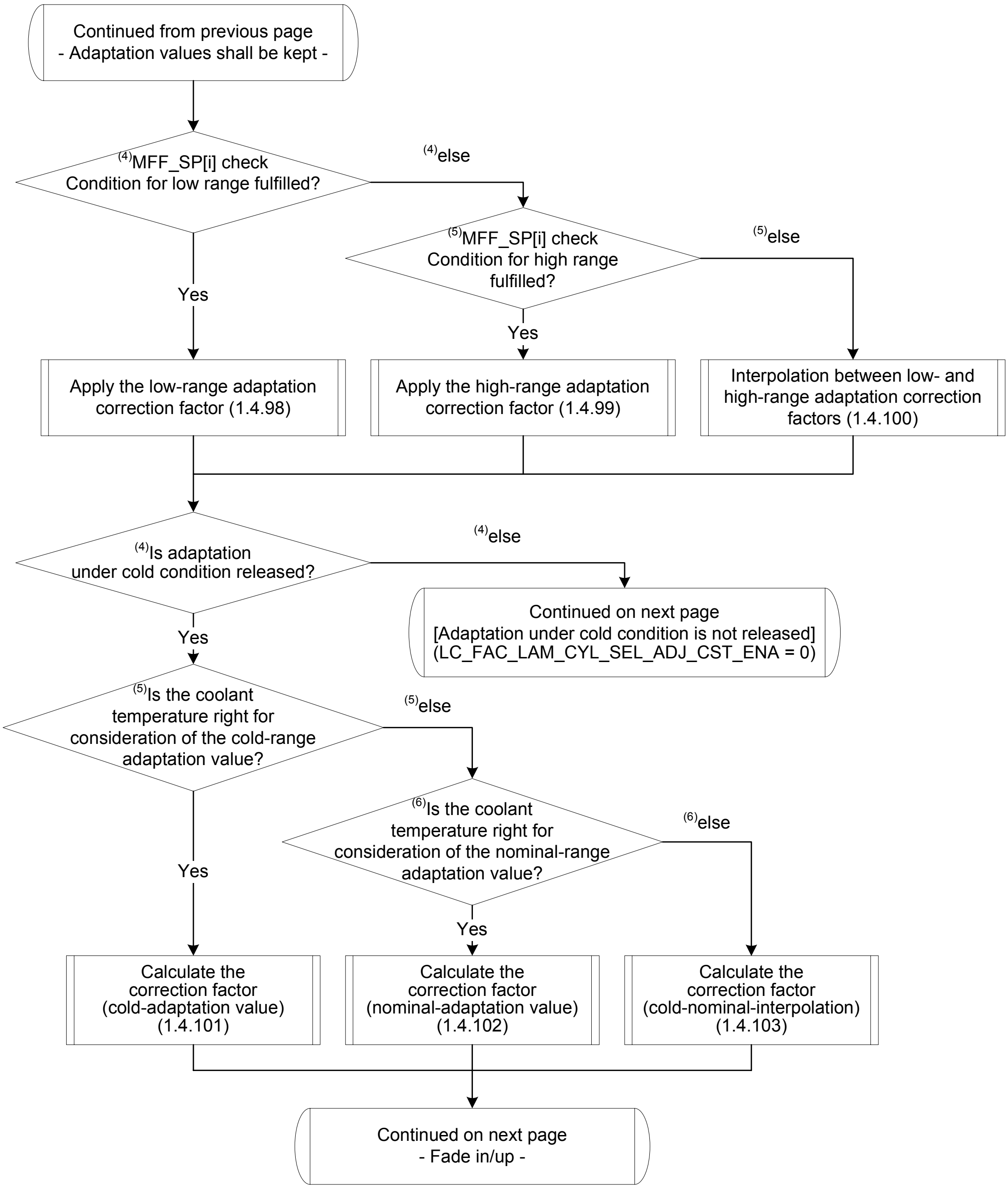


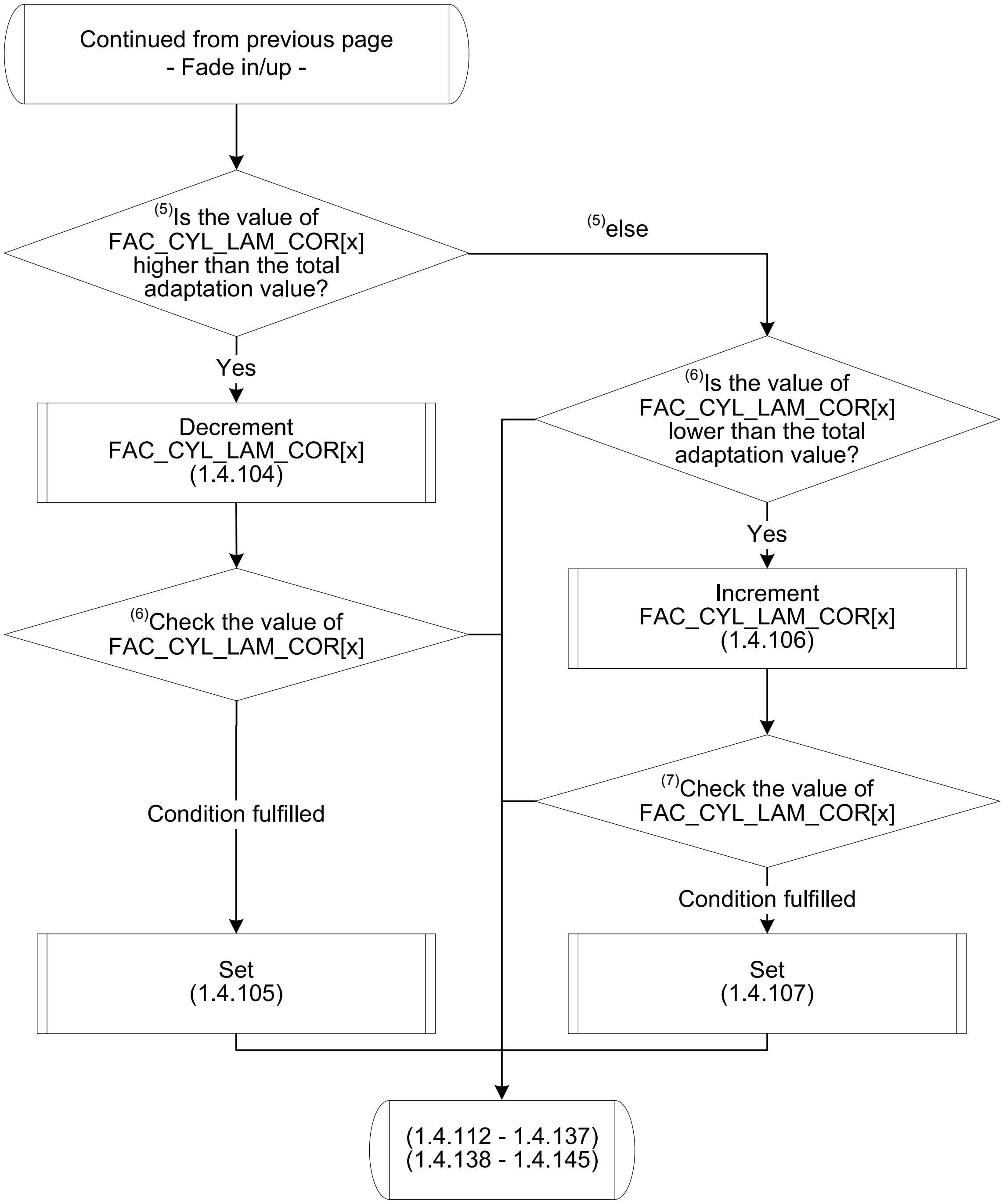

Continued from previous page
- adaptation release check -
(3)Is adaptation
under cold condition released?
(3)else
Continued on next page
[Adaptation under cold condition is not released]
(LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 0)
Yes
(4)Is the coolant
temperature right for
consideration of the cold-range
adaptation value?
(4)else
(5)Is the coolant
temperature right for
consideration of the nominal-range
adaptation value?
(5)else
Yes
Yes
Calculate the
correction factor
(cold-adaptation value)
(1.4.10)
Calculate the
correction factor
(nominal-adaptation value)
(1.4.11)
Calculate the
correction factor
(cold-nominal-interpolation)
(1.4.12)
Continued on next page
- Fade in/up -

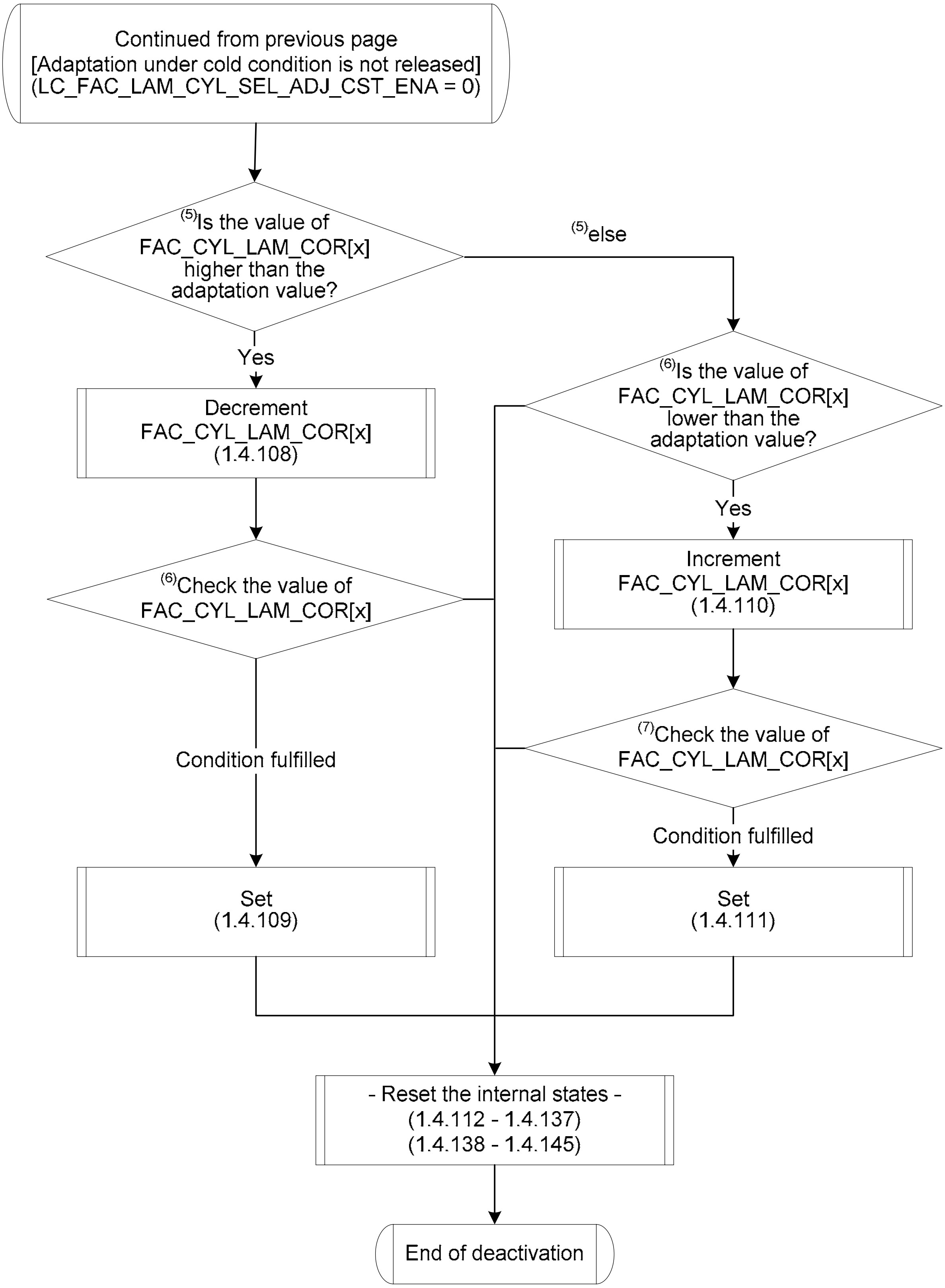

Continued from previous page
- Fade in/up -
(4)Temperature outside of the valid range?
(4)else
Yes
Reset action
(1.4.13 - 1.4.40)
(4)No virtual limitation, no drift-away behaviour of CILC and no observer divergence detected?
(4)else
Condition fulfilled
(5)Is the value of FAC_CYL_LAM_COR[x] higher than the total adaptation value?
(5)else
(5)Observer divergence detected?
(5)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.4.41)
(6)Is the value of FAC_CYL_LAM_COR[x] lower than the total adaptation value?
Yes
Increment FAC_CYL_LAM_COR[x] (1.4.43)
Yes
(6)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
(7)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
Set
(1.4.42)
Set
(1.4.44)
Set
(1.4.45)
(1.4.138 - 1.4.145)


Continued from previous page
[Adaptation under cold condition is not released]
(LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 0)
(4)Temperature outside of the valid range?
(4)else
Yes
Reset action (1.4.46 - 1.4.73)
(4)No virtual limitation, no drift-away behaviour of CILC and no observer divergence detected?
(4)else
Condition fulfilled
(5)Is the value of FAC_CYL_LAM_COR[x] higher than the adaptation value?
(5)else
(5)Observer divergence detected?
(5)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.4.74)
(6)Is the value of FAC_CYL_LAM_COR[x] lower than the adaptation value?
Yes
Increment FAC_CYL_LAM_COR[x] (1.4.76)
Yes
(6)Check the value of FAC_CYL_LAM_COR[x]
(7)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
Condition fulfilled
Set (1.4.75)
Set (1.4.77)
Set (1.4.78)
(3)LV_LAM_CYL_ACT[i] = 1 ?
(3)else
Yes
Save the current fuel mass value and deactivate the function (1.4.79 - 1.4.80)
(1.4.138 - 1.4.145)


Continued from previous page
[the function is inhibited by diagnosis]
(3)Should the adaptation values be reset?
(3)else
Yes
(4)FAC_CYL_LAM_COR[x] > 0?
(4)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.4.81)
(5)FAC_CYL_LAM_COR[x] < 0?
Yes
Increment FAC_CYL_LAM_COR[x] (1.4.83)
(5)FAC_CYL_LAM_COR[x] < 0?
(6)FAC_CYL_LAM_COR[x] > 0?
Yes
Yes
Set FAC_CYL_LAM_COR[x] = 0 (1.4.82)
Set FAC_CYL_LAM_COR[x] = 0 (1.4.84)
Reset the adaptation variables (1.4.85 - 1.4.97)
Continued on next page - Adaptation values shall be kept -
- Reset the internal states -
(1.4.112 - 1.4.137)
(1.4.138 - 1.4.145)


Continued from previous page
- Adaptation values shall be kept -
(4)MFF_SP[i] check
Condition for low range fulfilled?
(4)else
(5)MFF_SP[i] check
Condition for high range
fulfilled?
(5)else
Yes
Yes
Apply the low-range adaptation correction factor (1.4.98)
Apply the high-range adaptation correction factor (1.4.99)
Interpolation between low- and high-range adaptation correction factors (1.4.100)
(4)Is adaptation
under cold condition released?
(4)else
Continued on next page
[Adaptation under cold condition is not released]
(LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 0)
Yes
(5)Is the coolant
temperature right for
consideration of the cold-range
adaptation value?
(5)else
(6)Is the coolant
temperature right for
consideration of the nominal-range
adaptation value?
(6)else
Yes
Yes
Calculate the correction factor (cold-adaptation value) (1.4.101)
Calculate the correction factor (nominal-adaptation value) (1.4.102)
Calculate the correction factor (cold-nominal-interpolation) (1.4.103)
Continued on next page
- Fade in/up -


Continued from previous page
- Fade in/up -
(5)Is the value of FAC_CYL_LAM_COR[x] higher than the total adaptation value?
(5)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.4.104)
(6)Is the value of FAC_CYL_LAM_COR[x] lower than the total adaptation value?
Yes
Increment FAC_CYL_LAM_COR[x] (1.4.106)
(6)Check the value of FAC_CYL_LAM_COR[x]
(7)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
Condition fulfilled
Set (1.4.105)
Set (1.4.107)
(1.4.112 - 1.4.137)
(1.4.138 - 1.4.145)


Continued from previous page
[Adaptation under cold condition is not released]
(LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 0)
(5)Is the value of FAC_CYL_LAM_COR[x] higher than the adaptation value?
(5)else
Yes
Decrement FAC_CYL_LAM_COR[x] (1.4.108)
(6)Is the value of FAC_CYL_LAM_COR[x] lower than the adaptation value?
Yes
Increment FAC_CYL_LAM_COR[x] (1.4.110)
(6)Check the value of FAC_CYL_LAM_COR[x]
(7)Check the value of FAC_CYL_LAM_COR[x]
Condition fulfilled
Condition fulfilled
Set (1.4.109)
Set (1.4.111)
- Reset the internal states -
(1.4.112 - 1.4.137)
(1.4.138 - 1.4.145)
End of deactivation

```
if$^{(1)}$ LV_LAM_CYL_ENA[i] = 0
then
    if$^{(2)}$ STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 2
    then
        for$^{(3)}$ j(x) = 1 : Number of cylinders per exhaust bank
            FAC_LAM_CYL_SEL_ADJ_L_RNG[x] =
                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ (4.1.5.1)
            FAC_LAM_CYL_SEL_ADJ_H_RNG[x] =
                FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ (4.1.5.2)
            FAC_LAM_CYL_SEL_ADJ_CST[x] =
                FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$ (4.1.5.3)
            MFF_ADD_CYL_LAM_COR[x] =
                MFF_ADD_CYL_LAM_COR[x]$_{LDC}$ (4.1.5.4)
            LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] =
                LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i]$_{LDC}$ (4.1.5.5)
            LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] =
                LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i]$_{LDC}$ (4.1.5.6)
        endfor$^{(3)}$
    endif$^{(2)}$
    if$^{(2)}$ LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 0
    then
        if$^{(3)}$ MFF_SP[i] $\le$ C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG
        then
            FAC_LAM_CYL_SEL_ADJ[x] = FAC_LAM_CYL_SEL_ADJ_L_RNG[x]
                (4.1.5.7)
        else$^{(3)}$
            if$^{(4)}$ MFF_SP[i] $\ge$ C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG
            then
                FAC_LAM_CYL_SEL_ADJ[x] =
                    FAC_LAM_CYL_SEL_ADJ_H_RNG[x] (4.1.5.8)
            else$^{(4)}$
                FAC_LAM_CYL_SEL_ADJ[x] =
                    FAC_LAM_CYL_SEL_ADJ_L_RNG[x] +
                    (MFF_SP[i] - C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG)*
                    (FAC_LAM_CYL_SEL_ADJ_H_RNG[x] –
                    FAC_LAM_CYL_SEL_ADJ_L_RNG[x])/
                    (C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG –
                    C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG) (4.1.5.9)
            endif$^{(4)}$
        endif$^{(3)}$
        if$^{(3)}$ LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 1
            [adaptation under cold condition released]
        then
            if$^{(4)}$ TEMP_LAM_CYL_SEL $\le$
                    C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG
            then
                FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                    FAC_LAM_CYL_SEL_ADJ_CST[x] (4.1.5.10)
```

```
else^(4)
        if^(5)  TEMP_LAM_CYL_SEL ≥
                        C_TEMP_THD_CYL_SEL_ADJ_NOM_RNG
        then
                FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                        FAC_LAM_CYL_SEL_ADJ[x] (4.1.5.11)
        else^(5)
                FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                        FAC_LAM_CYL_SEL_ADJ_CST[x] +
                        (TEMP_LAM_CYL_SEL –
                        C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG)*
                        (FAC_LAM_CYL_SEL_ADJ[x] –
                        FAC_LAM_CYL_SEL_ADJ_CST[x])/
                        (C_TEMP_THD_CYL_SEL_ADJ_NOM_RNG –
                        C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG) (4.1.5.12)
        endif^(5)
endif^(4)
if^(4)  [(TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_CST_MIN) or
                (C_TEMP_LAM_CYL_CST_MAX <
                TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_MIN)]
then
        MFF_DELTA_LAM_CYL_SEL_REAC = 0                  (4.1.5.13)
        MFF_SP_SAVE_LAM_CYL_SEL = 0                     (4.1.5.14)
        FAC_VAL_LAM_CYL_SEL_REAC = 0                    (4.1.5.15)
        DELTA_LAMB_CYL_SEL_CQ[i] = 0                    (4.1.5.16)
        DELTA_LAMB_CYL_SEL_CQ_FIL[i] = 0                (4.1.5.17)
        LAMB_CYL_SEL_CQ_REF[i] = 0                      (4.1.5.18)
        LAMB_CYL_SEL_CQ_DIF_REF[i] = 0                  (4.1.5.19)
        CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] = 0            (4.1.5.20)
        CTR_LAMB_CYL_SEL_CQ_STAB[i] = 0                 (4.1.5.21)
        LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 0                 (4.1.5.22)
        LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0            (4.1.5.23)
        DELTA_LAMB_AV[i] = 0                            (4.1.5.24)
        DELTA_LAMB_ERR_RAW[i] = 0                       (4.1.5.25)
```

```
            DELTA_LAMB_ERR[i] = 0
                                                            (4.1.5.26)
            DELTA_LAMB_ERR_CYL_SEL[x] = 0
                                                            (4.1.5.27)
            DELTA_LAMB_CYL[x] = 0
                                                            (4.1.5.28)
            DELTA_LAMB_CYL_SEL[i] = 0
                                                            (4.1.5.29)
            DELTA_LAMB_MDL[i] = 0
                                                            (4.1.5.30)
            DELTA_LAMB_DIF_CYL[x] = 0
                                                            (4.1.5.31)
            FAC_CYL_LAM_INT[x] = 0
                                                            (4.1.5.32)
            FAC_CYL_LAM_I_MV_SHIFT[x] = 0
                                                            (4.1.5.33)
            FAC_CYL_LAM[x] = 0
                                                            (4.1.5.34)
            FAC_CYL_LAM_ABSV_SUM[i] = 0
                                                            (4.1.5.35)
            FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0
                                                            (4.1.5.36)
            FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0
                                                            (4.1.5.37)
            LV_FAC_CYL_LAM_LIM_MIN[x] = 0
                                                            (4.1.5.38)
            LV_FAC_CYL_LAM_LIM_MAX[x] = 0
                                                            (4.1.5.39)
            LV_FAC_CYL_LAM_VIRT_LIM[i] = 0
                                                            (4.1.5.40)
        endif(4)
        if(4)   (LV_FAC_CYL_LAM_VIRT_LIM[i] = 0 and
                        LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0 and
                             STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0)
        then
```

```
        if$^{(5)}$ FAC_CYL_LAM_COR[x]$_k$ >
                    FAC_LAM_CYL_SEL_ADJ_INTER[x]
        then
            FAC_CYL_LAM_COR[x]$_k$ = FAC_CYL_LAM_COR[x]$_{k-1}$ –
                    C_FAC_CYL_LAM_ABSV_DEC (4.1.5.41)
            if$^{(6)}$ FAC_CYL_LAM_COR[x]$_k$ <
                    FAC_LAM_CYL_SEL_ADJ_INTER[x]
            then
                FAC_CYL_LAM_COR[x]$_k$ =
                    FAC_LAM_CYL_SEL_ADJ_INTER[x] (4.1.5.42)
            endif$^{(6)}$
        else$^{(5)}$
            if$^{(6)}$ FAC_CYL_LAM_COR[x]$_k$ <
                    FAC_LAM_CYL_SEL_ADJ_INTER[x]
            then
                FAC_CYL_LAM_COR[x]$_k$ =
                    FAC_CYL_LAM_COR[x]$_{k-1}$
                    + C_FAC_CYL_LAM_ABSV_DEC (4.1.5.43)
                if$^{(7)}$ FAC_CYL_LAM_COR[x]$_k$ >
                    FAC_LAM_CYL_SEL_ADJ_INTER[x]
                then
                    FAC_CYL_LAM_COR[x]$_k$ =
                    FAC_LAM_CYL_SEL_ADJ_INTER[x]
                    (4.1.5.44)
                endif$^{(7)}$
            endif$^{(6)}$
        endif$^{(5)}$
    else$^{(4)}$
        if$^{(5)}$ STATE_LAMB_CYL_SEL_CQ_SLOP[i] $\neq$ 0
        then
            FAC_CYL_LAM_COR[x]$_k$ =
                    FAC_LAM_CYL_SEL_ADJ_INTER[x] (4.1.5.45)
        endif$^{(5)}$
    endif$^{(4)}$
else$^{(3)}$ [adaptation under cold condition not released]
    if$^{(4)}$ TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_MIN
    then

        MFF_DELTA_LAM_CYL_SEL_REAC = 0
                    (4.1.5.46)
        MFF_SP_SAVE_LAM_CYL_SEL = 0
                    (4.1.5.47)
        FAC_VAL_LAM_CYL_SEL_REAC = 0
                    (4.1.5.48)
        DELTA_LAMB_CYL_SEL_CQ[i] = 0
                    (4.1.5.49)
        DELTA_LAMB_CYL_SEL_CQ_FIL[i] = 0
                    (4.1.5.50)
        LAMB_CYL_SEL_CQ_REF[i] = 0
                    (4.1.5.51)
```

LAMB_CYL_SEL_CQ_DIF_REF[i] = 0

(4.1.5.52)

CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] = 0

(4.1.5.53)

CTR_LAMB_CYL_SEL_CQ_STAB[i] = 0

(4.1.5.54)

LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 0

(4.1.5.55)

LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0

(4.1.5.56)

DELTA_LAMB_AV[i] = 0

(4.1.5.57)

DELTA_LAMB_ERR_RAW[i] = 0

(4.1.5.58)

DELTA_LAMB_ERR[i] = 0

(4.1.5.59)

DELTA_LAMB_ERR_CYL_SEL[x] = 0

(4.1.5.60)

DELTA_LAMB_CYL[x] = 0

(4.1.5.61)

DELTA_LAMB_CYL_SEL[i] = 0

(4.1.5.62)

DELTA_LAMB_MDL[i] = 0

(4.1.5.63)

DELTA_LAMB_DIF_CYL[x] = 0

(4.1.5.64)

FAC_CYL_LAM_INT[x] = 0

(4.1.5.65)

FAC_CYL_LAM_I_MV_SHIFT[x] = 0

(4.1.5.66)

FAC_CYL_LAM[x] = 0

(4.1.5.67)

FAC_CYL_LAM_ABSV_SUM[i] = 0

(4.1.5.68)

FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0

(4.1.5.69)

FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0

```
                                                    (4.1.5.70)
        LV_FAC_CYL_LAM_LIM_MIN[x] = 0
                                                    (4.1.5.71)
        LV_FAC_CYL_LAM_LIM_MAX[x] = 0
                                                    (4.1.5.72)
        LV_FAC_CYL_LAM_VIRT_LIM[i] = 0
                                                    (4.1.5.73)
endif$^{(4)}$
if$^{(4)}$   (LV_FAC_CYL_LAM_VIRT_LIM[i] = 0 and
                 LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0 and
                       STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0)
then
    if$^{(5)}$   FAC_CYL_LAM_COR[x]$_k$ > FAC_LAM_CYL_SEL_ADJ[x]
    then
        FAC_CYL_LAM_COR[x]$_k$ = FAC_CYL_LAM_COR[x]$_{k-1}$ –
                   C_FAC_CYL_LAM_ABSV_DEC (4.1.5.74)
        if$^{(6)}$   FAC_CYL_LAM_COR[x]$_k$ <
                          FAC_LAM_CYL_SEL_ADJ[x]
        then
            FAC_CYL_LAM_COR[x]$_k$ =
                          FAC_LAM_CYL_SEL_ADJ[x]
                                                    (4.1.5.75)
        endif$^{(6)}$
    else$^{(5)}$
        if$^{(6)}$   FAC_CYL_LAM_COR[x]$_k$ <
                          FAC_LAM_CYL_SEL_ADJ[x]
        then
            FAC_CYL_LAM_COR[x]$_k$ =
                          FAC_CYL_LAM_COR[x]$_{k-1}$ +
                   C_FAC_CYL_LAM_ABSV_DEC (4.1.5.76)
            if$^{(7)}$   FAC_CYL_LAM_COR[x]$_k$ >
                          FAC_LAM_CYL_SEL_ADJ[x]
            then
                FAC_CYL_LAM_COR[x]$_k$ =
                     FAC_LAM_CYL_SEL_ADJ[x] (4.1.5.77)
            endif$^{(7)}$
        endif$^{(6)}$
    endif$^{(5)}$
else$^{(4)}$
    if$^{(5)}$   STATE_LAMB_CYL_SEL_CQ_SLOP[i] $\neq$ 0
    then
        FAC_CYL_LAM_COR[x]$_k$ =
                     FAC_LAM_CYL_SEL_ADJ[x] (4.1.5.78)
    endif$^{(5)}$
endif$^{(4)}$
```

```
        endif$^{(3)}$
        if$^{(3)}$   LV_LAM_CYL_ACT[i] = 1
        then
                MFF_SP_SAVE_LAM_CYL_SEL = MFF_SP_LAM_CYL_SEL
                                                                  (4.1.5.79)
                LV_LAM_CYL_ACT[i] = 0
                                                                  (4.1.5.80)
        endif$^{(3)}$
    else$^{(2)}$ [the function is inhibited by diagnosis]
        if$^{(3)}$   LC_LAM_CYL_SEL_ERR_RST = 1
                [adaptation values shall be reset]
        then
                if$^{(4)}$   FAC_CYL_LAM_COR[x]$_k$ > 0
                then
                        FAC_CYL_LAM_COR[x]$_k$ = FAC_CYL_LAM_COR[x]$_{k-1}$ -
                                         C_FAC_CYL_LAM_ABSV_DEC (4.1.5.81)
                        if$^{(5)}$   FAC_CYL_LAM_COR[x]$_k$ < 0
                        then
                                FAC_CYL_LAM_COR[x]$_k$ = 0
                                                                  (4.1.5.82)
                        endif$^{(5)}$
                else$^{(4)}$
                        if$^{(5)}$   FAC_CYL_LAM_COR[x]$_k$ < 0
                        then
                                FAC_CYL_LAM_COR[x]$_k$ = FAC_CYL_LAM_COR[x]$_{k-1}$ +
                                         C_FAC_CYL_LAM_ABSV_DEC (4.1.5.83)
                                if$^{(6)}$   FAC_CYL_LAM_COR[x]$_k$ > 0
                                then
                                        FAC_CYL_LAM_COR[x]$_k$ = 0
                                                                  (4.1.5.84)
                                endif$^{(6)}$
                        endif$^{(5)}$
                endif$^{(4)}$
                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ = 0
                                                                  (4.1.5.85)
                FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ = 0
                                                                  (4.1.5.86)
                FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$ = 0
                                                                  (4.1.5.87)
                MFF_ADD_CYL_LAM_COR[x]$_{LDC}$ = 0
                                                                  (4.1.5.88)
                LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i]$_{LDC}$ = 0
                                                                  (4.1.5.89)
                LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i]$_{LDC}$ = 0
                                                                  (4.1.5.90)
                FAC_LAM_CYL_SEL_ADJ[x] = 0
                                                                  (4.1.5.91)
                FAC_LAM_CYL_SEL_ADJ_L_RNG[x] = 0
```

```
                                                        (4.1.5.92)
        FAC_LAM_CYL_SEL_ADJ_H_RNG[x] = 0
                                                        (4.1.5.93)
        FAC_LAM_CYL_SEL_ADJ_CST[x] = 0
                                                        (4.1.5.94)
        MFF_ADD_CYL_LAM_COR[x] = 0
                                                        (4.1.5.95)
        LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] = 0
                                                        (4.1.5.96)
        LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] = 0
                                                        (4.1.5.97)
else(3) [adaptation values shall be kept]
        if(4)   MFF_SP[i] ≤ C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG
        then
                FAC_LAM_CYL_SEL_ADJ[x] =
                        FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ (4.1.5.98)
        else(4)
                if(5)   MFF_SP[i] ≥ C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG
                then
                        FAC_LAM_CYL_SEL_ADJ[x] =
                                FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ (4.1.5.99)
                else(5)
                        FAC_LAM_CYL_SEL_ADJ[x] =
                                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ +
                                (MFF_SP[i] - C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG)*
                                (FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ –
                                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$)/
                                (C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG –
                                C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG) (4.1.5.100)
                endif(5)
        endif(4)
        if(4)   LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA = 1
                [adaptation under cold condition released]
        then
                if(5)   TEMP_LAM_CYL_SEL ≤
                                C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG
                then
                        FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$
                                                        (4.1.5.101)
                else(5)
                        if(6)   TEMP_LAM_CYL_SEL ≥
                                        C_TEMP_THD_CYL_SEL_ADJ_NOM_RNG
                        then
                                FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                        FAC_LAM_CYL_SEL_ADJ[x]
                                                        (4.1.5.102)
                        else(6)
                                FAC_LAM_CYL_SEL_ADJ_INTER[x] =
                                        FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$ +
```

```
                    $(TEMP\_LAM\_CYL\_SEL - C\_TEMP\_THD\_CYL\_SEL\_ADJ\_COLD\_RNG) * (FAC\_LAM\_CYL\_SEL\_ADJ[x] - FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_{LDC}) / (C\_TEMP\_THD\_CYL\_SEL\_ADJ\_NOM\_RNG - C\_TEMP\_THD\_CYL\_SEL\_ADJ\_COLD\_RNG)$ (4.1.5.103)
            endif(6)
        endif(5)
        if(5) $FAC\_CYL\_LAM\_COR[x]_k > FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$
        then
            $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1} - C\_FAC\_CYL\_LAM\_ABSV\_DEC$ (4.1.5.104)
            if(6) $FAC\_CYL\_LAM\_COR[x]_k < FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$
            then
                $FAC\_CYL\_LAM\_COR[x]_k = FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$ (4.1.5.105)
            endif(6)
        else(5)
            if(6) $FAC\_CYL\_LAM\_COR[x]_k < FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$
            then
                $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1} + C\_FAC\_CYL\_LAM\_ABSV\_DEC$ (4.1.5.106)
                if(7) $FAC\_CYL\_LAM\_COR[x]_k > FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$
                then
                    $FAC\_CYL\_LAM\_COR[x]_k = FAC\_LAM\_CYL\_SEL\_ADJ\_INTER[x]$ (4.1.5.107)
                endif(7)
            endif(6)
        endif(5)
    else(4) [adaptation under cold condition not released]
        if(5) $FAC\_CYL\_LAM\_COR[x]_k > FAC\_LAM\_CYL\_SEL\_ADJ[x]$
        then
            $FAC\_CYL\_LAM\_COR[x]_k = FAC\_CYL\_LAM\_COR[x]_{k-1} - C\_FAC\_CYL\_LAM\_ABSV\_DEC$ (4.1.5.108)
            if(6) $FAC\_CYL\_LAM\_COR[x]_k < FAC\_LAM\_CYL\_SEL\_ADJ[x]$
            then
                $FAC\_CYL\_LAM\_COR[x]_k = FAC\_LAM\_CYL\_SEL\_ADJ[x]$ (4.1.5.109)
            endif(6)
        else(5)
            if(6) $FAC\_CYL\_LAM\_COR[x]_k < FAC\_LAM\_CYL\_SEL\_ADJ[x]$
            then
```

```
                FAC_CYL_LAM_COR[x]$_k$ =
                        FAC_CYL_LAM_COR[x]$_{k-1}$ +
                        C_FAC_CYL_LAM_ABSV_DEC (4.1.5.110)
                if(7) FAC_CYL_LAM_COR[x]$_k$ >
                        FAC_LAM_CYL_SEL_ADJ[x]
                then
                    FAC_CYL_LAM_COR[x]$_k$ =
                        FAC_LAM_CYL_SEL_ADJ[x] (4.1.5.111)
                endif(7)
            endif(6)
        endif(5)
    endif(4)
endif(3)
LV_LAM_CYL_ACT[i] = 0
                                        (4.1.5.112)
DELTA_LAMB_CYL_SEL_CQ[i] = 0
                                        (4.1.5.113)
DELTA_LAMB_CYL_SEL_CQ_FIL[i] = 0
                                        (4.1.5.114)
LAMB_CYL_SEL_CQ_REF[i] = 0
                                        (4.1.5.115)
LAMB_CYL_SEL_CQ_DIF_REF[i] = 0
                                        (4.1.5.116)
CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] = 0
                                        (4.1.5.117)
CTR_LAMB_CYL_SEL_CQ_STAB[i] = 0
                                        (4.1.5.118)
LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 0
                                        (4.1.5.119)
LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0
                                        (4.1.5.120)
DELTA_LAMB_AV[i] = 0
                                        (4.1.5.121)
DELTA_LAMB_ERR_RAW[i]          = 0
                                        (4.1.5.122)
DELTA_LAMB_ERR[i] = 0
                                        (4.1.5.123)
DELTA_LAMB_ERR_CYL_SEL[x] =0
                                        (4.1.5.124)
DELTA_LAMB_CYL[x] = 0
                                        (4.1.5.125)
DELTA_LAMB_CYL_SEL[i] = 0
                                        (4.1.5.126)
DELTA_LAMB_MDL[i] = 0
```

```
            DELTA_LAMB_DIF_CYL[x] = 0                          (4.1.5.127)
            FAC_CYL_LAM_INT[x] = 0                             (4.1.5.128)
            FAC_CYL_LAM_I_MV_SHIFT[x] = 0                      (4.1.5.129)
            FAC_CYL_LAM[x] = 0                                 (4.1.5.130)
            FAC_CYL_LAM_ABSV_SUM[i] = 0                        (4.1.5.131)
            FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0                     (4.1.5.132)
            FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0                 (4.1.5.133)
            LV_FAC_CYL_LAM_LIM_MIN[x] = 0                      (4.1.5.134)
            LV_FAC_CYL_LAM_LIM_MAX[x] = 0                      (4.1.5.135)
            LV_FAC_CYL_LAM_VIRT_LIM[i] = 0                     (4.1.5.136)
                                                               (4.1.5.137)
        endif(2)
        STATE_LAM_CYL_SEL[i] = 0 [PAS]                         (4.1.5.138)
        STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]                     (4.1.5.139)
        LV_LAM_CYL_SEL_SEG_REF_CLL[i] = 0                      (4.1.5.140)
        FAC_CYL_LAM_P[x] = 0                                   (4.1.5.141)
        FAC_CYL_LAM_P_MV_SHIFT[x] = 0                          (4.1.5.142)
        CTR_T_LAM_CYL_ADJ[i] = 0                               (4.1.5.143)
        CTR_LAM_CYL_SEL_REF_CLL[i] = 0                         (4.1.5.144)
        LV_LAM_CYL_SEL_ADJ_VLD[i] = 0                          (4.1.5.145)
    endif(1)
```

Formula section:

4.1.6 Signal evaluation (segment synchronous)

The aim of signal evaluation with a low sampling rate is the precise dividing of the lambda sensor output signal into the identified cylinder. Therefore, the sampling moment shall be computed, depending on the engine operating state, precisely. The Sampling moment is influenced by phase displacement predetermined by IP_CRK_SYN[i]. The value of the phase shift is given in °CRK.

4.1.6.1 Calculation of the segment period

Depending on the engine speed the segment period is simply computed by

```
if(1) two separate exhaust banks are concerned, then
        $TA = 2 * \frac{2 * 60}{N * NC\_CYL\_NR}$
                                                        (4.1.6.1.1)
else(1)
        $TA = \frac{2 * 60}{N * NC\_CYL\_NR}$
                                                        (4.1.6.1.2)
endif(1)
```

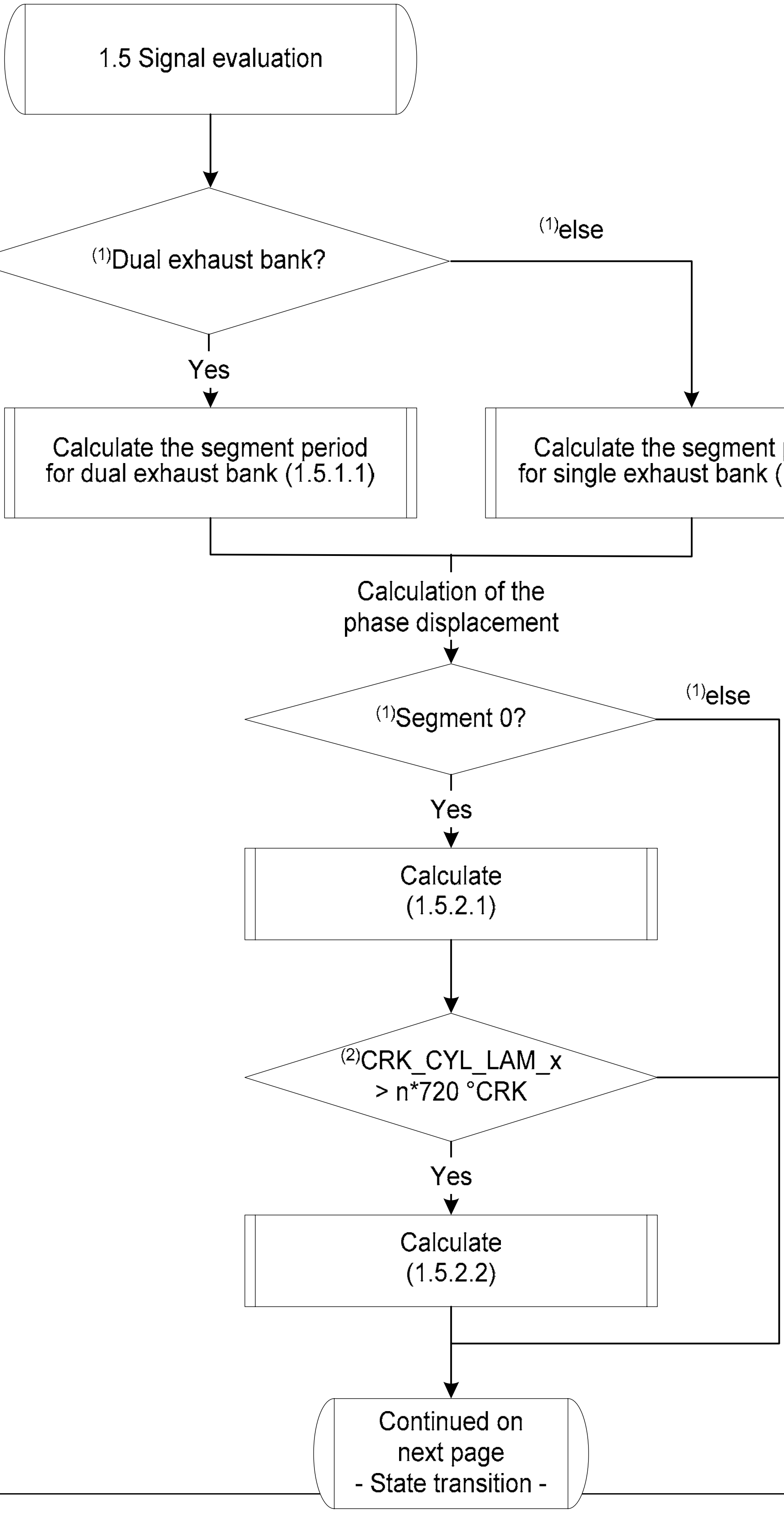
4.1.6.2 Calculation of the phase displacement

The phase shift - caused by system delayed response - is computed once per combustion cycle using the map IP_CRK_SYN[i], adjusted by C_CRK_CYL_LAM[x], PHA_SHIFT_CAM_EX[i] and DELTA_CRK_CYL_LAM[i]. It shall be determined at the beginning of the combustion cycle (SEG_NR = 0). That applies to the first and ensuing segments.

```
if(1) SEG_NR = 0
then
    CRK_CYL_LAM[x] = IP_CRK_SYN[i] + C_CRK_CYL_LAM[x] +
                        PHA_SHIFT_CAM_EX[i] + DELTA_CRK_CYL_LAM[i] +
                        IP_CRK_DELTA_AMP_CYL_LAM   (4.1.6.2.1)

    if(2)   CRK_CYL_LAM[x] > $\alpha$*720 °CRK
    then
            CRK_CYL_LAM[x] = CRK_CYL_LAM[x] – $\alpha$*720 °CRK
                                                        (4.1.6.2.2)
    endif(2)
endif(1)
```

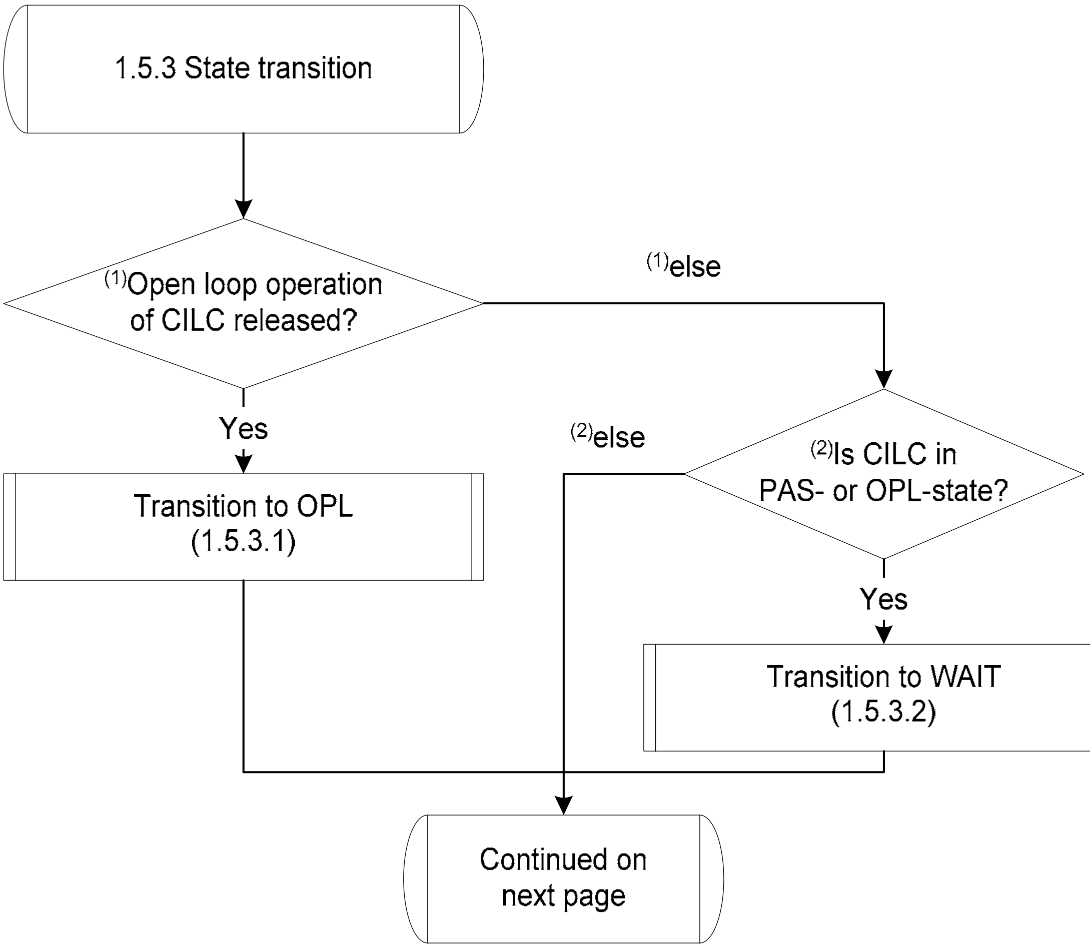
Remark:

$\alpha = 1, 2, ...$

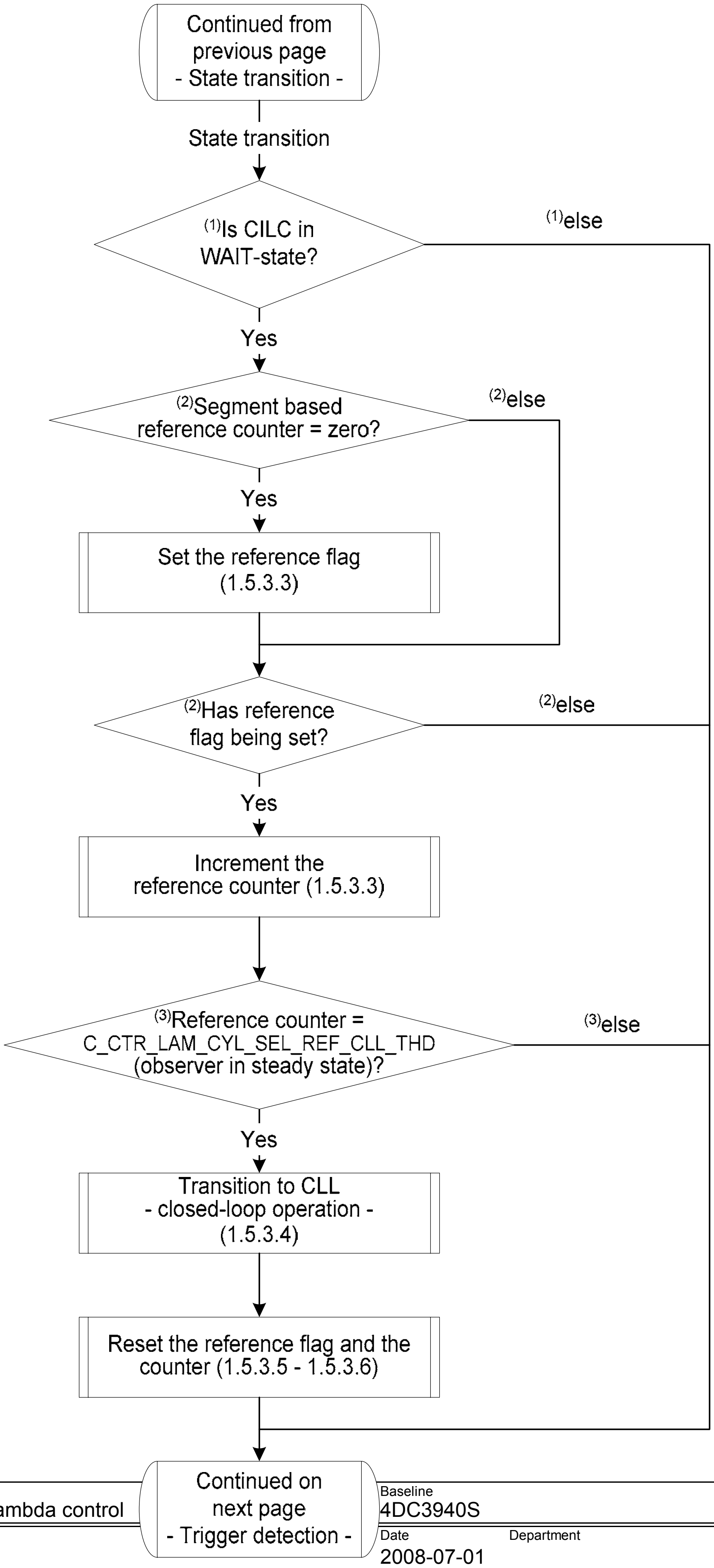

1.5 Signal evaluation
(1)Dual exhaust bank?
(1)else
Yes
Calculate the segment period for dual exhaust bank (1.5.1.1)
Calculate the segment period for single exhaust bank (1.5.1.2)
Calculation of the phase displacement
(1)Segment 0?
(1)else
Yes
Calculate (1.5.2.1)
(2)CRK_CYL_LAM_x > n*720 °CRK
Yes
Calculate (1.5.2.2)
Continued on next page - State transition -

4.1.6.3 State - Transition

In case of first activation or even reactivation the calculations in chapter 1.9 may be carried out on condition that at least one combustion cycle has been completed and the function is not in open loop mode.



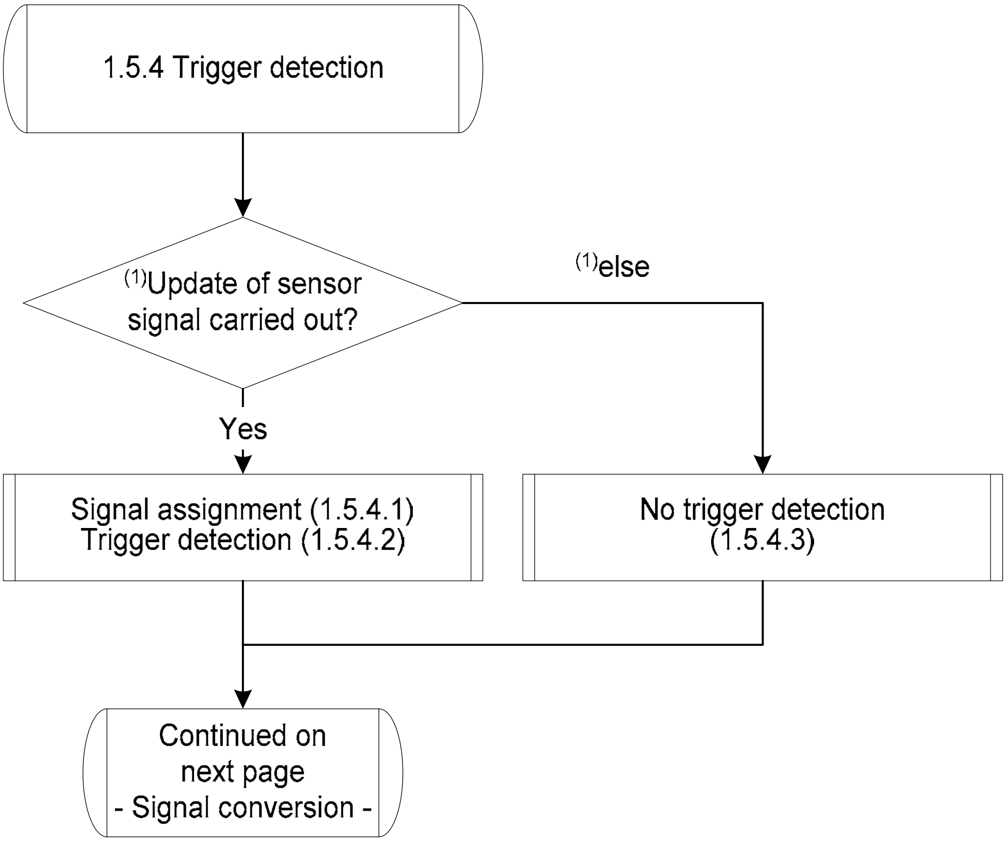
```
if$^{(1)}$ LC_LAM_CYL_SEL_OPL = 1
then
    STATE_LAM_CYL_SEL[i] = 1 [OPL]
                                                        (4.1.6.3.1)
else$^{(1)}$
    if$^{(2)}$   (STATE_LAM_CYL_SEL[i] = 0 [PAS] or STATE_LAM_CYL_SEL[i] = 1 [OPL])
    then
        STATE_LAM_CYL_SEL[i] = 2 [WAIT]
                                                        (4.1.6.3.2)
    endif$^{(2)}$
endif$^{(1)}$
```


Continued from previous page - State transition -
State transition
(1)Is CILC in WAIT-state?
(1)else
Yes
(2)Segment based reference counter = zero?
(2)else
Yes
Set the reference flag (1.5.3.3)
(2)Has reference flag being set?
(2)else
Yes
Increment the reference counter (1.5.3.3)
(3)Reference counter = C_CTR_LAM_CYL_SEL_REF_CLL_THD (observer in steady state)?
(3)else
Yes
Transition to CLL - closed-loop operation - (1.5.3.4)
Reset the reference flag and the counter (1.5.3.5 - 1.5.3.6)
Continued on next page - Trigger detection -

```
if(1) STATE_LAM_CYL_SEL[i] = 2 [WAIT]
then
    if(2)   CTR_LAM_CYL_SEL_REF_CLL[i] = 0
    then
            LV_LAM_CYL_SEL_SEG_REF_CLL[i] = 1
                                                                  (4.1.6.3.3)
    endif(2)
    if(2)   LV_LAM_CYL_SEL_SEG_REF_CLL[i] = 1
    then
            increment (CTR_LAM_CYL_SEL_REF_CLL[i])
            if(3)   CTR_LAM_CYL_SEL_REF_CLL[i] =
                                        C_CTR_LAM_CYL_SEL_REF_CLL_THD
            then
                    STATE_LAM_CYL_SEL[i] = 3 [CLL]
                                                                  (4.1.6.3.4)
                    LV_LAM_CYL_SEL_SEG_REF_CLL[i] = 0
                                                                  (4.1.6.3.5)
                    CTR_LAM_CYL_SEL_REF_CLL[i] = 0
                                                                  (4.1.6.3.6)
            endif(3)
    endif(2)
endif(1)
```

4.1.6.4 Trigger detection

The trigger for the corresponding segment is created if the state of the variable VLS_CYL_TRIG[x] changes. At the same time the value of VLS_CYL[x] shall be transferred to VLS_CYL_SEL[x].



```
if(1) $VLS\_CYL\_TRIG[x] \neq VLS\_CYL\_TRIG[x]_{k-1}$
then
    VLS_CYL_SEL[x] = VLS_CYL[x]                    (4.1.6.4.1)
    LV_TRIG[x] = 1                                 (4.1.6.4.2)
else(1)
    LV_TRIG[x] = 0                                 (4.1.6.4.3)
endif(1)
```

4.1.7 Signal conversion

The voltage value of VLS_CYL_SEL[x] shall be converted to Lambda using the characteristic curve IP_LAMB_CYL$_x$[1] and has to be shifted by its mean value. The resulting value represents the global AFR deviation. Calculation and shifting the mean value shall be carried out in segment step.

if$^{(1)}$ LV_TRIG[x] =1

then

LAMB_CYL_SEL[x] = IP_LAMB_CYL (VLS_CYL_SEL[x])

(4.1.7.1)

$$\text{DELTA\_LAMB\_AV[i]} = \text{LAMB\_CYL\_SEL[x]} - \frac{1}{\alpha}\sum_{k=1}^{\alpha}\text{LAMB\_CYL\_SEL[x]};$$

α :number of cylinders per bank

(4.1.7.2)

4.1.8 Calculation of time lag of the lambda-sensor model

T1_LAMB_CYL_SEL[i] = IP_T1_LAMB_CYL_SEL[i]

(4.1.8.1)

4.1.9 Calculation of the estimation-error

Estimation error represents the difference between the measured value, DELTA_LAMB_AV[i], and the estimated value, DELTA_LAMB_MDL[i].

DELTA_LAMB_ERR_RAW[i] = DELTA_LAMB_AV[i] –
DELTA_LAMB_MDL[i][x(n-1)] **(4.1.9.1)**

Remark: Software shall guarantee a segment synchronous comparison, i.e. the value of DELTA_LAMB_AV[i] at the segment-time x shall be compared with that DELTA_LAMB_MDL[i] which was calculated in segment x during the last combustion cycle [x(n-1)].

4.1.10 Weighting the estimation-error

Depending on T1_LAMB_CYL_SEL[i], the error shall be weighted. This is carried out by means of a characteristic curve. A non-linear dependent weighting function might be considered for late come out.

DELTA_LAMB_ERR[i] = DELTA_LAMB_ERR_RAW[i] * IP_FAC_ERR_LAM_SEL

(4.1.10.1)

---

[1] x...actual cylinder

4.1.11 Signal splitting / Calculation of the cylinder selective AFR / Signal merge

Splitting procedure shall be implemented in order the weighted error signal to be allocated to an internal state, which describes the current estimated value of air -/fuel ratio in each cylinder respectively (Figure 2).

$$\text{DELTA\_LAMB\_ERR\_CYL\_SEL}[x] = \text{DELTA\_LAMB\_ERR}[i]$$

(4.1.11.1)

An adaptation mechanism, including an integral-action element for each cylinder – corrected by exhaust-manifold-port factor C_CRLC_EXV_FAC[x], adjusts the estimated value to the actual one. The adjusted value of the cylinder individual deviation is calculated by the equation

$$\text{DELTA\_LAMB\_CYL}[x]_k = (\text{DELTA\_LAMB\_CYL}[x]_{k-1} + \text{DELTA\_LAMB\_ERR\_CYL\_SEL}[x]_k * \text{TA}) * \text{C\_CRLC\_EXV\_FAC}[x]$$

(4.1.11.2)

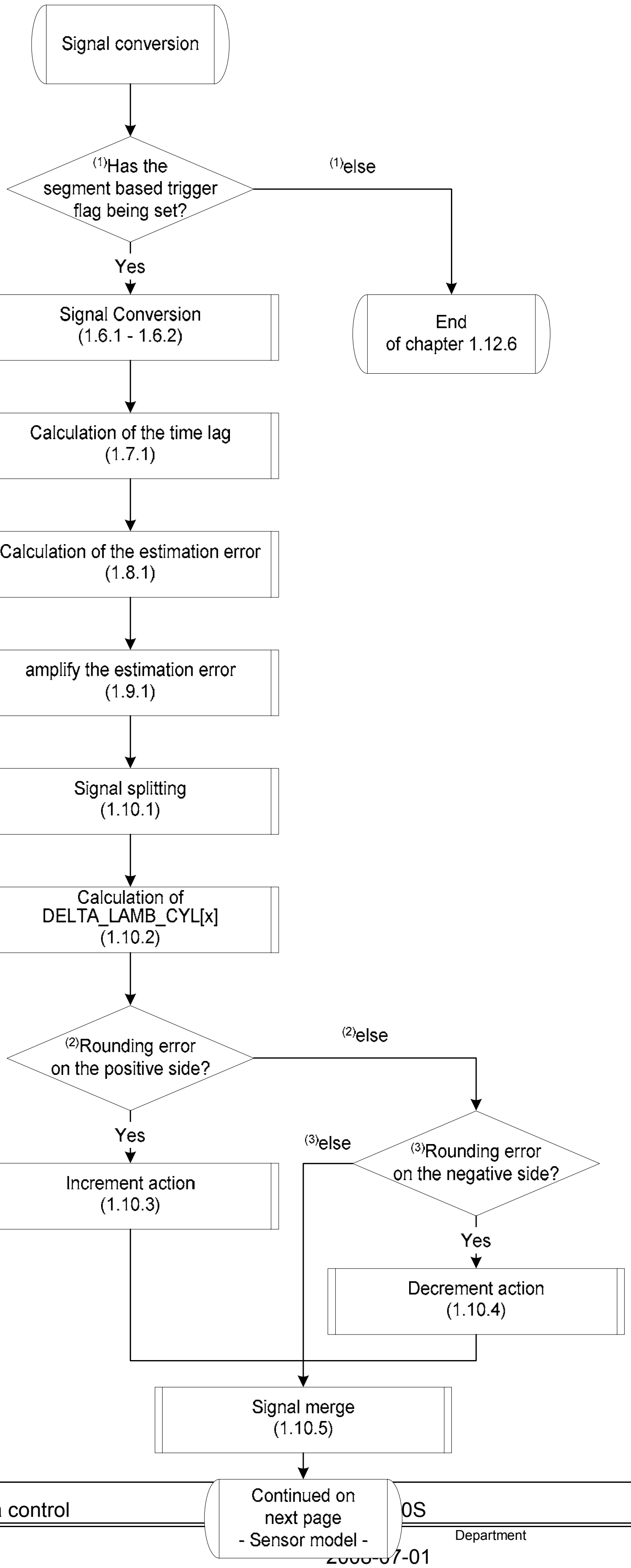
In order to avoid rounding error that leads to zero-readout of the gradient term in aforementioned equation, following scan shall be made:

```
if$^{(2)}$ $(\text{DELTA\_LAMB\_ERR\_CYL\_SEL}[x]_k * \text{TA} = 0$ and $\text{DELTA\_LAMB\_ERR\_CYL\_SEL}[x]_k > 0)$
then
    increment $(\text{DELTA\_LAMB\_ERR\_CYL\_SEL}[x]_k * \text{TA})$
                                                            (4.1.11.3)
else$^{(2)}$
    if$^{(3)}$ $(\text{DELTA\_LAMB\_ERR\_CYL\_SEL}[x]_k * \text{TA} = 0$ and $\text{DELTA\_LAMB\_ERR\_CYL\_SEL}[x]_k < 0)$
    then
        decrement $(\text{DELTA\_LAMB\_ERR\_CYL\_SEL}[x]_k * \text{TA})$
                                                            (4.1.11.4)
    endif$^{(3)}$
endif$^{(2)}$
```

In accordance with the ignition sequence signal-reassembling procedure shall be carried out in order to get the divided signals together. DELTA_LAMB_CYL_SEL[i] is input of the sensor model for the current segment.

$$\text{DELTA\_LAMB\_CYL\_SEL}[i]_k = \text{DELTA\_LAMB\_CYL}[x]_k$$

(4.1.11.5)


Signal conversion
(1)Has the segment based trigger flag being set?
(1)else
Yes
Signal Conversion (1.6.1 - 1.6.2)
End of chapter 1.12.6
Calculation of the time lag (1.7.1)
Calculation of the estimation error (1.8.1)
amplify the estimation error (1.9.1)
Signal splitting (1.10.1)
Calculation of DELTA_LAMB_CYL[x] (1.10.2)
(2)Rounding error on the positive side?
(2)else
Yes
Increment action (1.10.3)
(3)else
(3)Rounding error on the negative side?
Yes
Decrement action (1.10.4)
Signal merge (1.10.5)
Continued on next page - Sensor model -

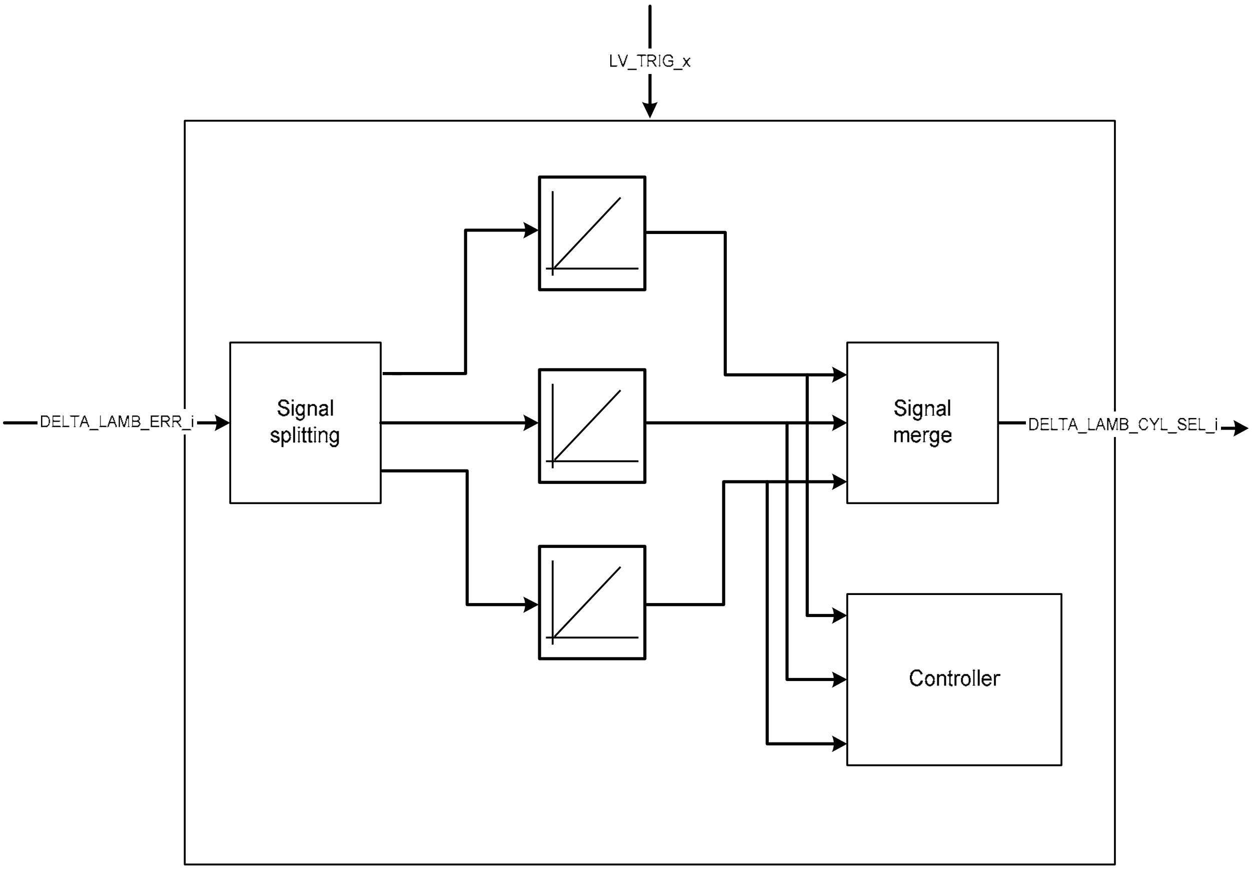


Figure 2: Signal splitting, adjustment and signal merge

4.1.12 Sensor model

The dynamic behaviour of WRAF sensor is described by a sensor model, first order low-pass-filter. This is characterised by a time lag, T1_LAMB_CYL_SEL[i]. Furthermore, the effect of the delay is considered by choice of an adequate sampling time relevant to crankshaft angle. This is depicted in 0 ... 720 ° CRK range.
The approximated model is theoretically defined in time range as

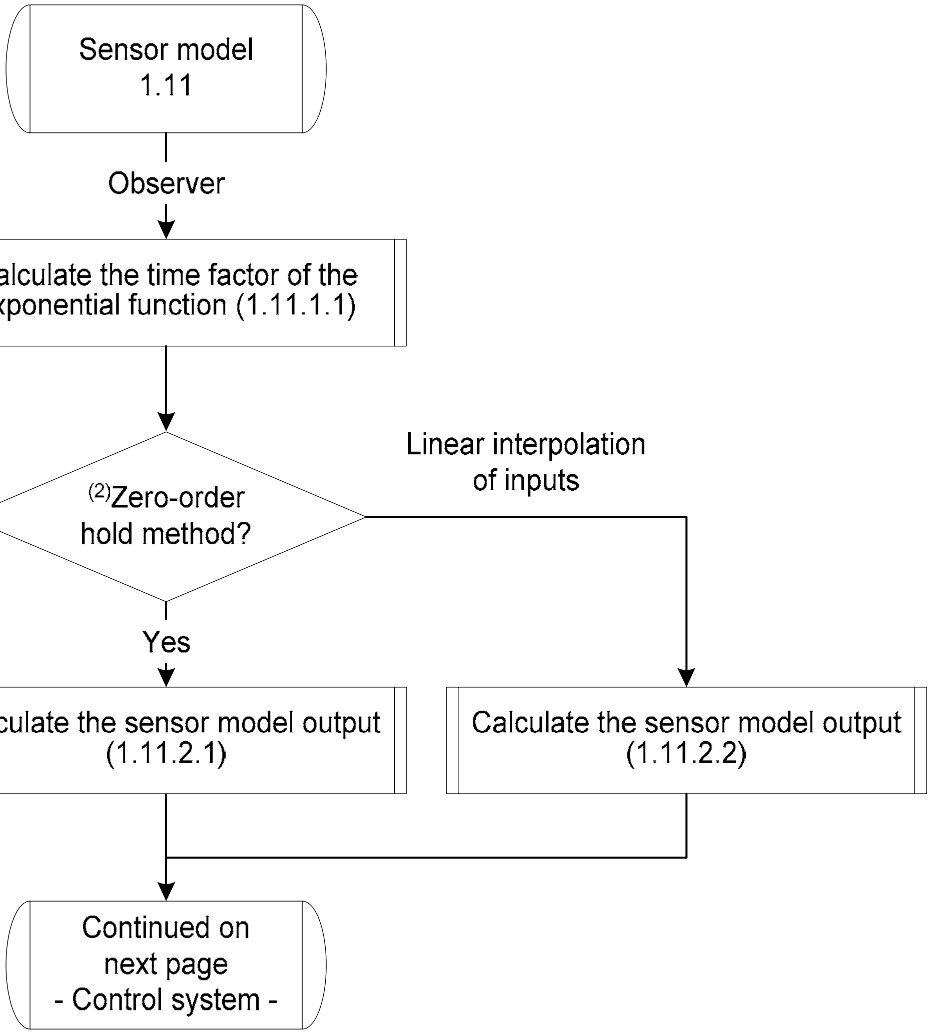
$$y(t) = 1 - e^{-t/T}$$

and described in frequency range as continuous-time model

$$G(s) = \frac{1}{1 + sT}$$

with T = T1_LAMB_CYL_SEL[i]

Conversion of continuous-time model to discrete-time model with sample time TA can be made using either 'zoh'- or 'foh'-method.


Sensor model
1.11
Observer
Calculate the time factor of the exponential function (1.11.1.1)
(2)Zero-order hold method?
Linear interpolation of inputs
Yes
Calculate the sensor model output (1.11.2.1)
Calculate the sensor model output (1.11.2.2)
Continued on next page
- Control system -

4.1.12.1 Calculation of model parameters

$$\mathrm{FAC\_T[i]} = \frac{\mathrm{T1\_LAMB\_CYL\_SEL[i]}}{\mathrm{TA}}$$

(4.1.12.1.1)

- Zero-order hold on the inputs 'zoh' - recommended method (LC_LAMB_DELTA_MDL_HLD = 0)

  calibration maps are

  $\mathrm{IP\_EXP\_LAMB\_MDL[i]} = e^{-\frac{1}{\mathrm{FAC\_T[i]}}}$,

  IP_N1_FAC_LAMB_MDL[i] = 1 - IP_EXP_LAMB_MDL[i] and
  IP_N2_FAC_LAMB_MDL[i] = 0

- Linear interpolation of inputs 'foh' (LC_LAMB_DELTA_MDL_HLD = 1)

  calibration maps are

  $\mathrm{IP\_EXP\_LAMB\_MDL[i]} = e^{-\frac{1}{\mathrm{FAC\_T[i]}}}$,

  IP_N1_FAC_LAMB_MDL[i] = 1 - FAC_T[i] * (1 - IP_EXP_LAMB_MDL[i]) and
  IP_N2_FAC_LAMB_MDL[i] = FAC_T[i] - (1 + FAC_T[i]) * IP_EXP_LAMB_MDL[i]

4.1.12.2 Calculation of the sensor model output

The output of sensor model is governed by DELTA_LAMB_MDL[i].

```
if(2)  LC_LAMB_DELTA_MDL_HLD = 0
then
        $\mathrm{DELTA\_LAMB\_MDL[i]}_k = \mathrm{IP\_EXP\_LAMB\_MDL[i]} * \mathrm{DELTA\_LAMB\_MDL[i]}_{k-1} + \mathrm{IP\_N1\_FAC\_LAMB\_MDL[i]} * \mathrm{DELTA\_LAMB\_CYL\_SEL[i]}_k$
                                                                    (4.1.12.2.1)
else(2)
        $\mathrm{DELTA\_LAMB\_MDL[i]}_k = \mathrm{IP\_EXP\_LAMB\_MDL[i]} * \mathrm{DELTA\_LAMB\_MDL[i]}_{k-1} + \mathrm{IP\_N1\_FAC\_LAMB\_MDL[i]} * \mathrm{DELTA\_LAMB\_CYL\_SEL[i]}_k + \mathrm{IP\_N2\_FAC\_LAMB\_MDL[i]} * \mathrm{DELTA\_LAMB\_CYL\_SEL[i]}_{k-1}$
                                                                    (4.1.12.2.2)
endif(2)
```

Attention:

Precise allocation required. For segment x, $\mathrm{DELTA\_LAMB\_MDL[i]}_{k-1}$ and $\mathrm{DELTA\_LAMB\_CYL\_SEL[i]}_{k-1}$ are buffered variables that were calculated in the segment x -1 for one bank system and x - 2 for two bank system.

4.1.13 Control system

A cylinder individual AFR regulation is carried out (Figure 3).

4.1.13.1 Control deviation

The tracking error for corresponding cylinder is defined as DELTA_LAMB_DIF_CYL[x].

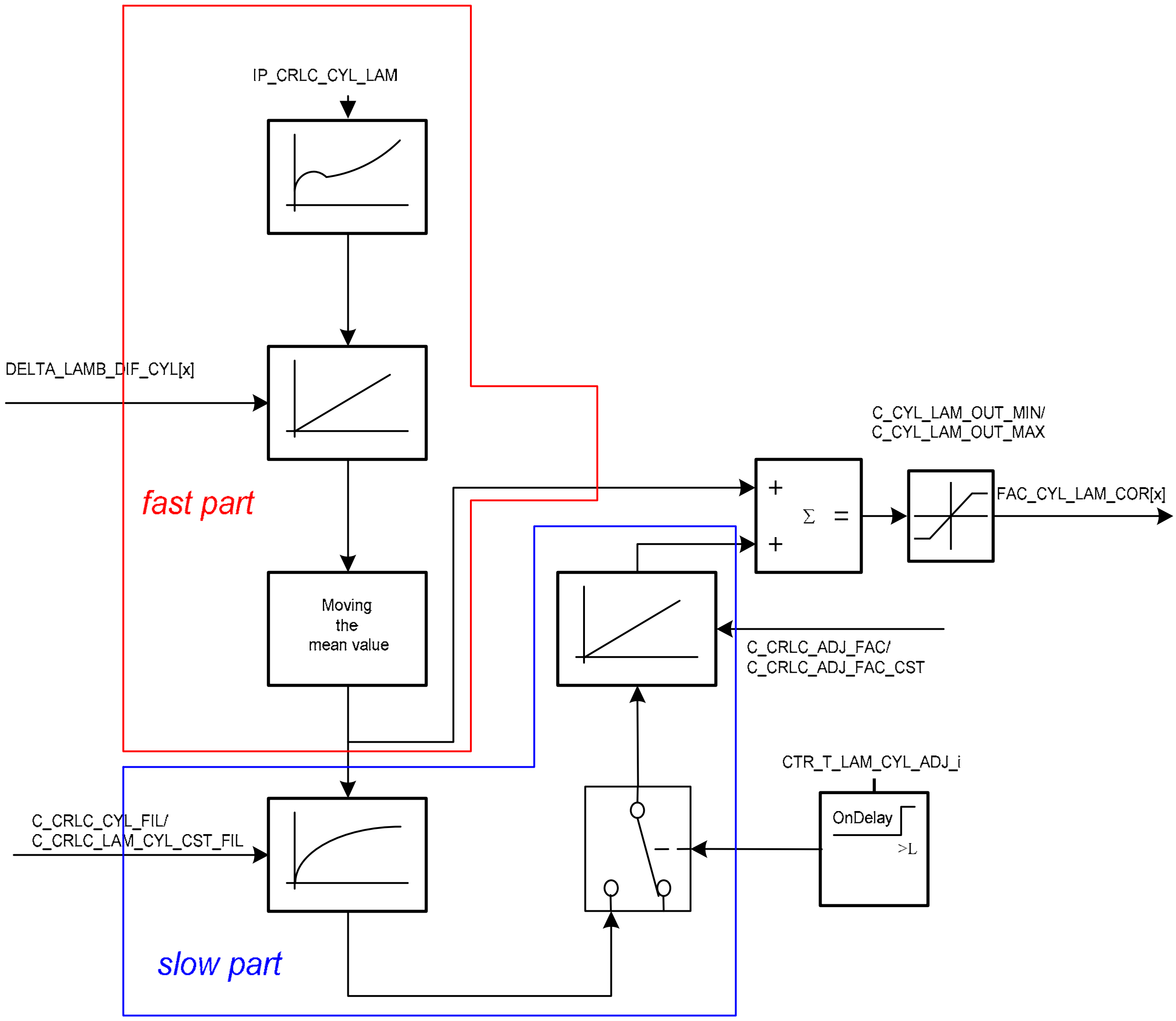
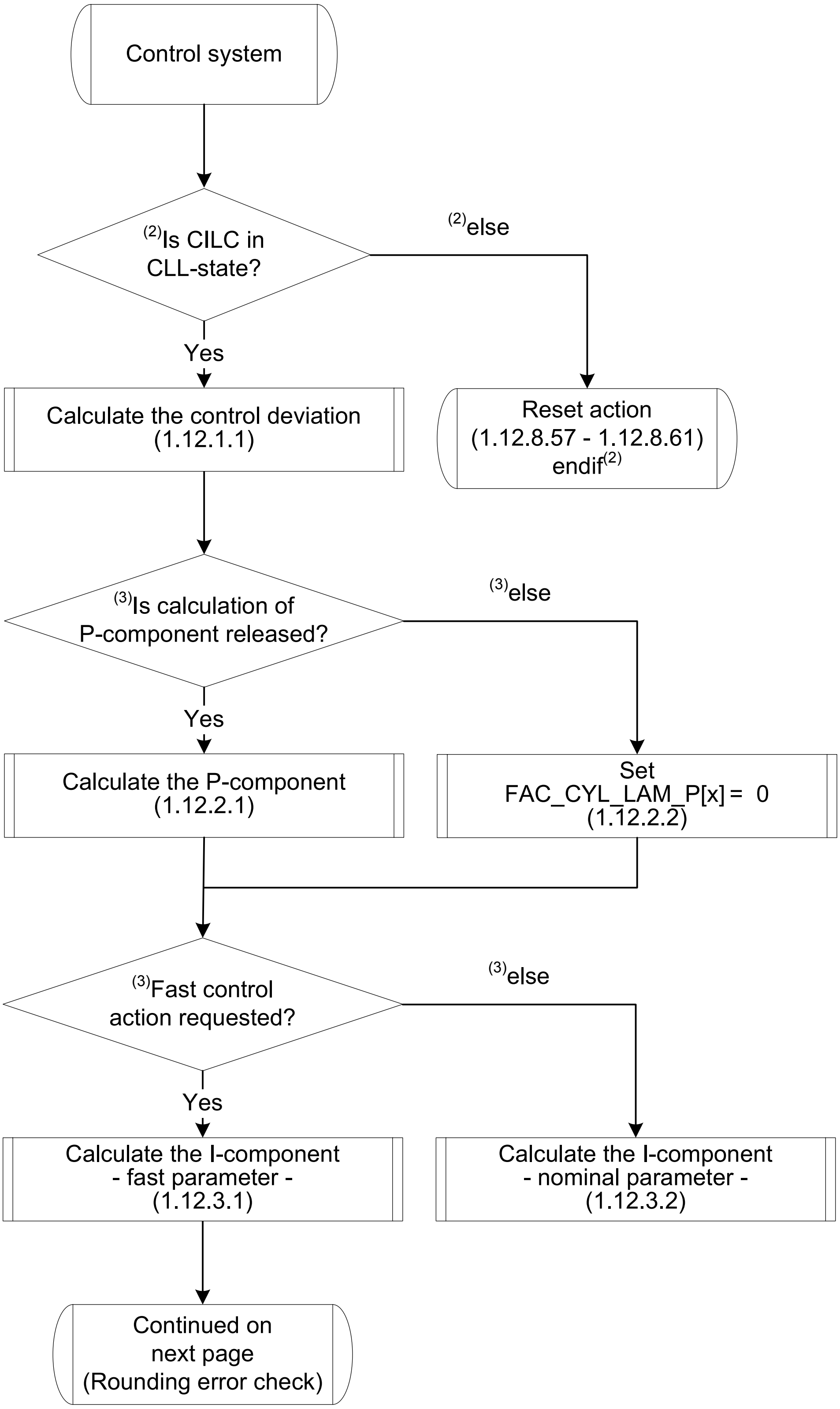


Figure 3: Control system including the adaptation mechanism


Control system
(2)Is CILC in CLL-state?
(2)else
Yes
Calculate the control deviation (1.12.1.1)
Reset action (1.12.8.57 - 1.12.8.61) endif(2)
(3)Is calculation of P-component released?
(3)else
Yes
Calculate the P-component (1.12.2.1)
Set FAC_CYL_LAM_P[x] = 0 (1.12.2.2)
(3)Fast control action requested?
(3)else
Yes
Calculate the I-component - fast parameter - (1.12.3.1)
Calculate the I-component - nominal parameter - (1.12.3.2)
Continued on next page (Rounding error check)

```
if(2)   STATE_LAM_CYL_SEL[i] = 3 [CLL]
then
        DELTA_LAMB_DIF_CYL[x] = DELTA_LAMB_CYL[x]
                                    – C_DELTA_LAMB_CYL_SP (4.1.13.1.1)
```

4.1.13.2 Calculation of proportional-action controller (P-Component)

The P-Component can be switched off using an on-off switch

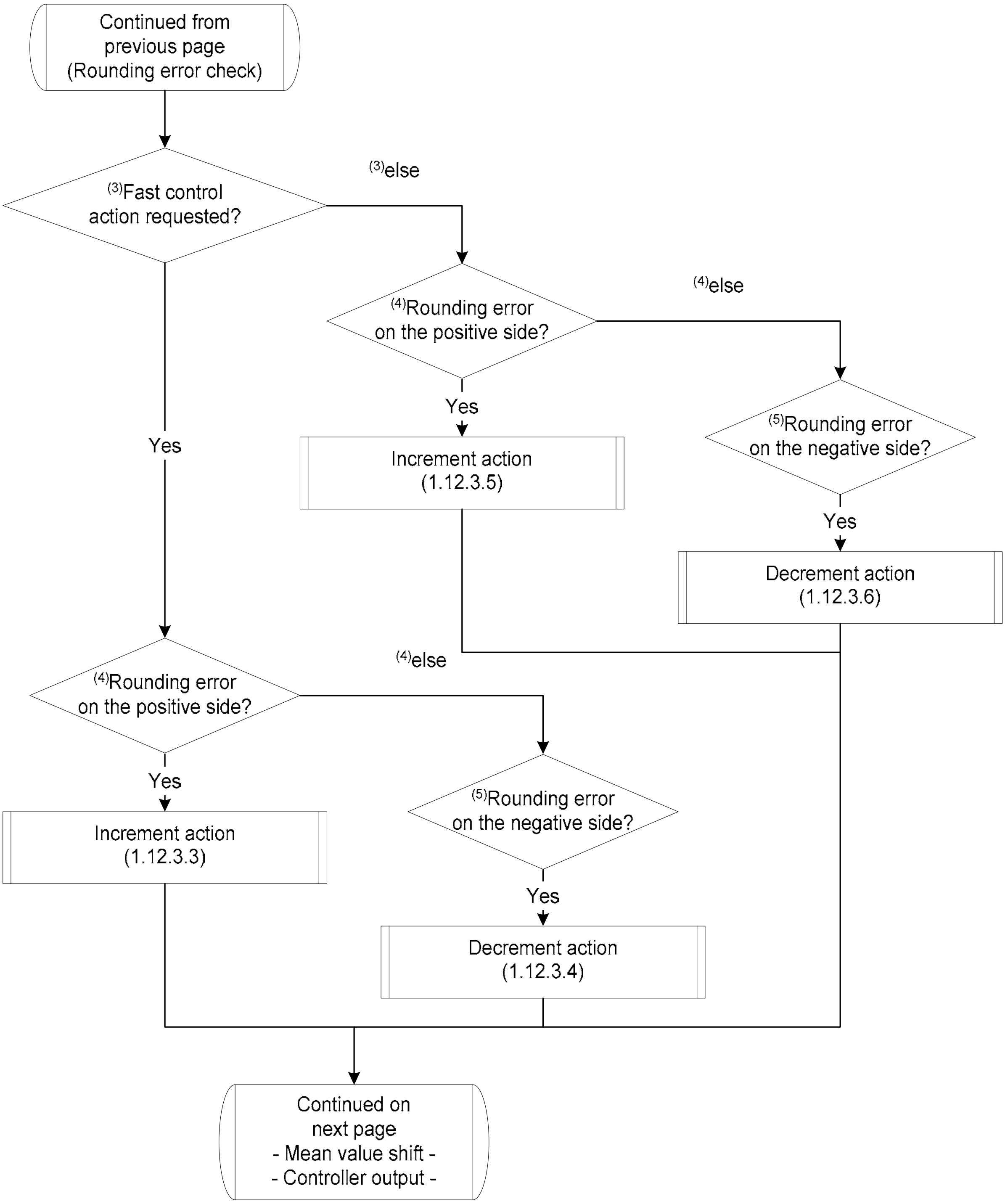
```
if(3)   LC_FAC_CYL_LAM_P_SWI = 1
then
        FAC_CYL_LAM_P[x] =  IP_FAC_GAIN_P_LAM_CYL_SEL*
                DELTA_LAMB_DIF_CYL[x]*IP_CRLC_P_NEUT_RNG (4.1.13.2.1)
else(3)
        FAC_CYL_LAM_P[x] =  0
                                                         (4.1.13.2.2)
endif(3)
```

4.1.13.3 Calculation of integral-action controller (I-Component)

Tracking step disturbances with zero steady state error requires integration of the tracking error. Integral action can be incorporated by defining an internal state FAC_CYL_LAM_INT[x] that represents the integrated tracking error. This state is governed by FAC_CYL_LAM_INT[x]. For the injection systems being not adapted the integral action can be switched to the fast control parameter. The action is then triggered by LV_LAM_CYL_SEL_CTL_FAST_REQ.

```
if(3)   LV_LAM_CYL_SEL_CTL_FAST_REQ = 1
then
        FAC_CYL_LAM_INT[x]$_k$ =  FAC_CYL_LAM_INT[x]$_{k-1}$ +
              IP_CRLC_INT_FAST_CYL_LAM*DELTA_LAMB_DIF_CYL[x]$_k$*
                                      IP_CRLC_I_NEUT_RNG (4.1.13.3.1)
else(3)
        FAC_CYL_LAM_INT[x]$_k$ =  FAC_CYL_LAM_INT[x]$_{k-1}$ +
                   IP_CRLC_CYL_LAM*DELTA_LAMB_DIF_CYL[x]$_k$*
                                      IP_CRLC_I_NEUT_RNG (4.1.13.3.2)
endif(3)
```

In order to avoid rounding error that leads to zero-readout of the gradient term in aforementioned equation, following scan shall be made:

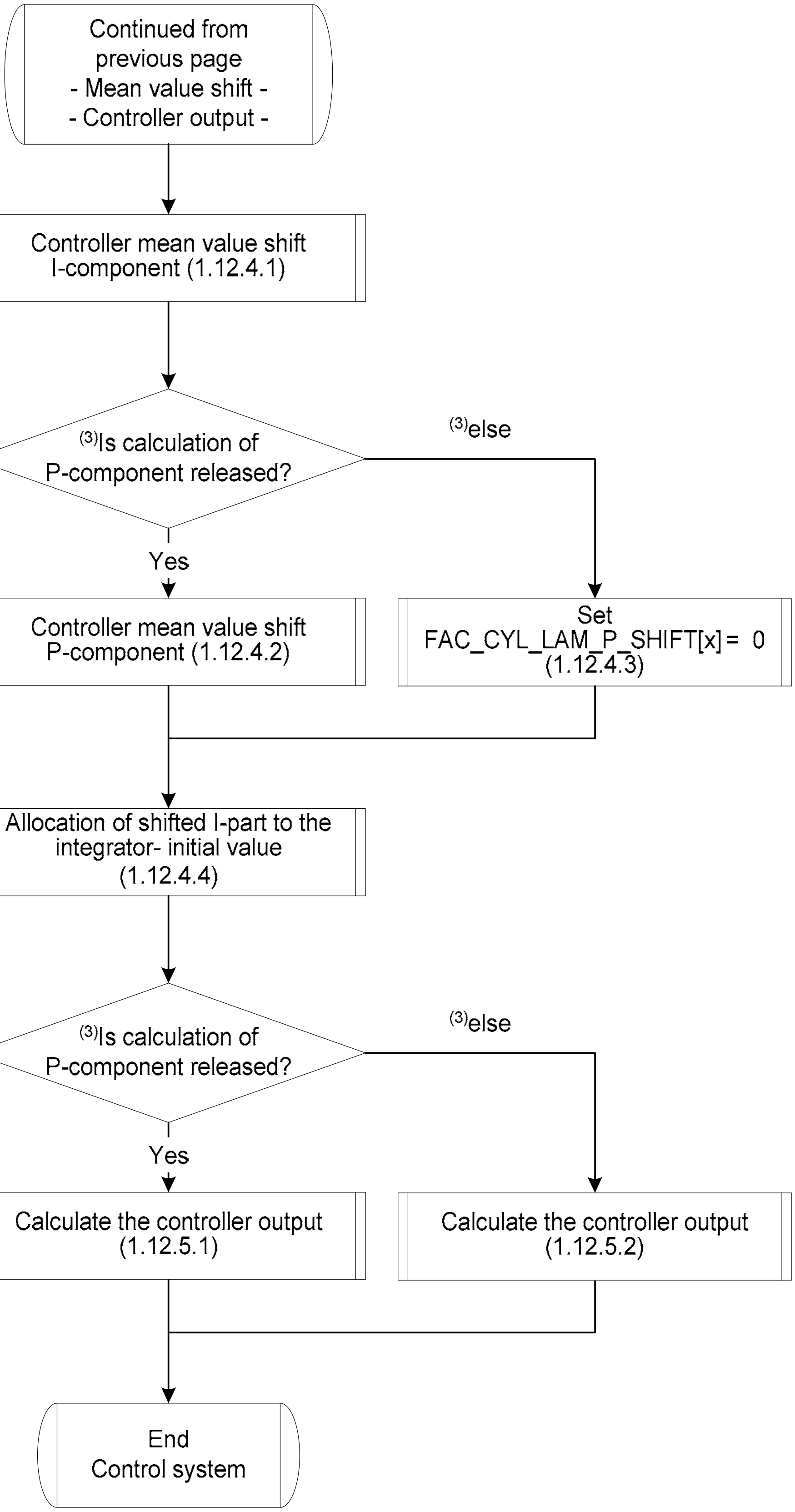


```
if$^{(3)}$ LV_LAM_CYL_SEL_CTL_FAST_REQ = 1
then
    if$^{(4)}$ (IP_CRLC_INT_FAST_CYL_LAM*DELTA_LAMB_DIF_CYL[x]$_k$*
                IP_CRLC_I_NEUT_RNG = 0 and
                IP_CRLC_INT_FAST_CYL_LAM > 0 and
                IP_CRLC_I_NEUT_RNG > 0 and
                DELTA_LAMB_DIF_CYL[x]$_k$ > 0)
    then
        increment (IP_CRLC_INT_FAST_CYL_LAM*
                   DELTA_LAMB_DIF_CYL[x]$_k$*
                   IP_CRLC_I_NEUT_RNG) (4.1.13.3.3)
    else$^{(4)}$
        if$^{(5)}$ (IP_CRLC_INT_FAST_CYL_LAM*
                    DELTA_LAMB_DIF_CYL[x]$_k$*
                    IP_CRLC_I_NEUT_RNG = 0 and
                    IP_CRLC_INT_FAST_CYL_LAM > 0 and
                    IP_CRLC_I_NEUT_RNG > 0 and
                    DELTA_LAMB_DIF_CYL[x]$_k$ < 0)
        then
            decrement (IP_CRLC_INT_FAST_CYL_LAM*
                       DELTA_LAMB_DIF_CYL[x]$_k$*
                       IP_CRLC_I_NEUT_RNG) (4.1.13.3.4)
        endif$^{(5)}$
    endif$^{(4)}$
else$^{(3)}$
    if$^{(4)}$ (IP_CRLC_CYL_LAM*DELTA_LAMB_DIF_CYL[x]$_k$*
                IP_CRLC_I_NEUT_RNG = 0 and IP_CRLC_CYL_LAM > 0 and
                IP_CRLC_I_NEUT_RNG > 0 and
                DELTA_LAMB_DIF_CYL[x]$_k$ > 0)
    then
        increment (IP_CRLC_CYL_LAM*DELTA_LAMB_DIF_CYL[x]$_k$*
                   IP_CRLC_I_NEUT_RNG) (4.1.13.3.5)
    else$^{(4)}$
        if$^{(5)}$ (IP_CRLC_CYL_LAM*DELTA_LAMB_DIF_CYL[x]$_k$*
                    IP_CRLC_I_NEUT_RNG = 0 and
```

```
                                    IP_CRLC_CYL_LAM > 0 and
                                    IP_CRLC_I_NEUT_RNG > 0 and
                                    $DELTA\_LAMB\_DIF\_CYL[x]_k$ < 0)
            then
                decrement (IP_CRLC_CYL_LAM*
                                    $DELTA\_LAMB\_DIF\_CYL[x]_k$*
                                    IP_CRLC_I_NEUT_RNG) (4.1.13.3.6)
            endif(5)
        endif(4)
    endif(3)
```

4.1.13.4 Mean value shift

In order to avoid that the output of the cylinder individual control system affects the main controller for mean λ - control, it shall be shifted by its mean value.


Continued from previous page
- Mean value shift -
- Controller output -
Controller mean value shift I-component (1.12.4.1)
(3)Is calculation of P-component released?
(3)else
Yes
Controller mean value shift P-component (1.12.4.2)
Set FAC_CYL_LAM_P_SHIFT[x] = 0 (1.12.4.3)
Allocation of shifted I-part to the integrator- initial value (1.12.4.4)
(3)Is calculation of P-component released?
(3)else
Yes
Calculate the controller output (1.12.5.1)
Calculate the controller output (1.12.5.2)
End
Control system

$$\mathrm{FAC\_CYL\_LAM\_I\_MV\_SHIFT}[x]_k = \mathrm{FAC\_CYL\_LAM\_INT}[x]_k - \frac{1}{\alpha} * \sum_{x=1}^{\alpha} \mathrm{FAC\_CYL\_LAM\_INT}[x]$$

(4.1.13.4.1)

```
if^(3)  LC_FAC_CYL_LAM_P_SWI = 1
then
        $\mathrm{FAC\_CYL\_LAM\_P\_MV\_SHIFT}[x]_k = \mathrm{FAC\_CYL\_LAM\_P}[x]_k - \frac{1}{\alpha} * \sum_{x=1}^{\alpha} \mathrm{FAC\_CYL\_LAM\_P}[x]$
        α : number of cylinders per bank
                                                                        (4.1.13.4.2)
else^(3)
        FAC_CYL_LAM_P_MV_SHIFT[x] = 0
                                                                        (4.1.13.4.3)
endif^(3)
```

The value of FAC_CYL_LAM_I_MV_SHIFT[x] shall be allocated to the initial value of the integrator:

$$\mathrm{FAC\_CYL\_LAM\_INT}[x]_k = \mathrm{FAC\_CYL\_LAM\_I\_MV\_SHIFT}[x]_k$$

(4.1.13.4.4)

4.1.13.5 Calculation of the controller output

```
if^(3)  LC_FAC_CYL_LAM_P_SWI = 1
then
        $\mathrm{FAC\_CYL\_LAM}[x]_k = \mathrm{FAC\_CYL\_LAM\_P\_MV\_SHIFT}[x]_k + \mathrm{FAC\_CYL\_LAM\_I\_MV\_SHIFT}[x]_k$   (4.1.13.5.1)
else^(3)
        $\mathrm{FAC\_CYL\_LAM}[x]_k = \mathrm{FAC\_CYL\_LAM\_I\_MV\_SHIFT}[x]_k$
                                                                        (4.1.13.5.2)
endif^(3)
```

4.1.13.6 Limitation of the controller output

If the cylinder individual controller output falls into the defined lower/upper limit, the corresponding flag has to be set and the involved controller output shall be limited to the minimum or maximum value.

```
if^(3)  $\mathrm{FAC\_CYL\_LAM}[x]_k \le \mathrm{C\_FAC\_CYL\_LAM\_MIN}$
then
        $\mathrm{FAC\_CYL\_LAM}[x]_k = \mathrm{C\_FAC\_CYL\_LAM\_MIN}$
                                                                        (4.1.13.6.1)
        LV_FAC_CYL_LAM_LIM_MIN[x] = 1
```

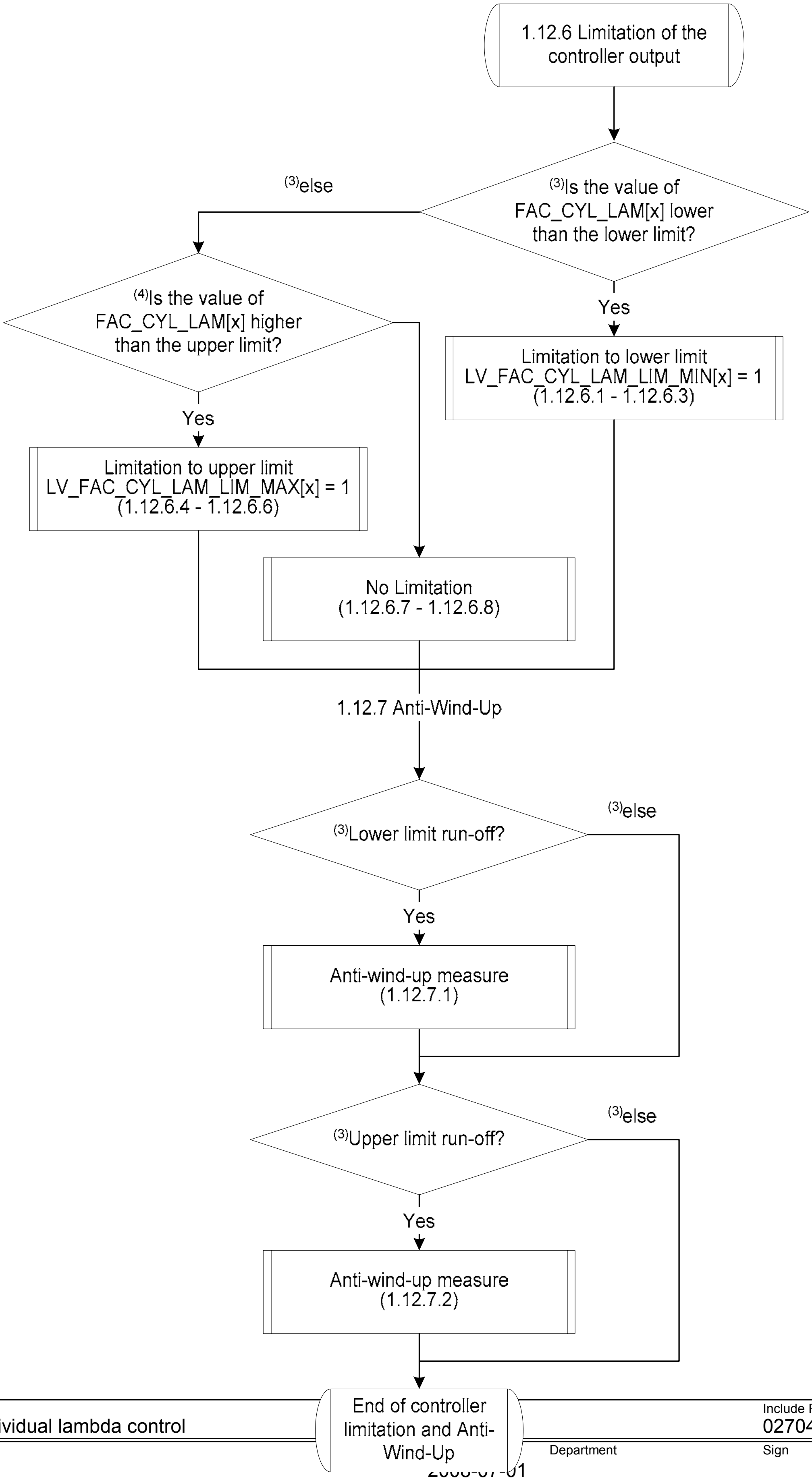
```
                                                              (4.1.13.6.2)
        LV_FAC_CYL_LAM_LIM_MAX[x] = 0
                                                              (4.1.13.6.3)

else$^{(3)}$
        if$^{(4)}$   $FAC\_CYL\_LAM[x]_k \geq C\_FAC\_CYL\_LAM\_MAX$
        then
                $FAC\_CYL\_LAM[x]_k = C\_FAC\_CYL\_LAM\_MAX$
                                                              (4.1.13.6.4)
                LV_FAC_CYL_LAM_LIM_MAX[x] = 1
                                                              (4.1.13.6.5)
                LV_FAC_CYL_LAM_LIM_MIN[x] = 0
                                                              (4.1.13.6.6)
        else$^{(4)}$
                LV_FAC_CYL_LAM_LIM_MIN[x] = 0
                                                              (4.1.13.6.7)
                LV_FAC_CYL_LAM_LIM_MAX[x] = 0
                                                              (4.1.13.6.8)
        endif$^{(4)}$
endif$^{(4)}$
```

4.1.13.7 Anti-Wind-Up

As anti wind up measure for the integral-action element, FAC_CYL_LAM_INT[x], following calculations shall be carried out if the controller output is limited.

```
if$^{(3)}$   LV_FAC_CYL_LAM_LIM_MIN[x] = 1
then
        $FAC\_CYL\_LAM\_INT[x]_k =$
            $\frac{\alpha}{\alpha - 1} * C\_FAC\_CYL\_LAM\_MIN + \frac{1}{\alpha - 1} * \sum_{j=x+1}^{x-1} FAC\_CYL\_LAM\_INT[j]_k$
                                                              (4.1.13.7.1)
endif$^{(3)}$

if$^{(3)}$   LV_FAC_CYL_LAM_LIM_MAX[x] = 1
then
        $FAC\_CYL\_LAM\_INT[x]_k =$
            $\frac{\alpha}{\alpha - 1} * C\_FAC\_CYL\_LAM\_MAX + \frac{1}{\alpha - 1} * \sum_{j=x+1}^{x-1} FAC\_CYL\_LAM\_INT[j]_k$
                                                              (4.1.13.7.2)
endif$^{(3)}$
```

α: number of cylinders per bank


1.12.6 Limitation of the controller output
(3)Is the value of FAC_CYL_LAM[x] lower than the lower limit?
(3)else
Yes
Limitation to lower limit LV_FAC_CYL_LAM_LIM_MIN[x] = 1 (1.12.6.1 - 1.12.6.3)
(4)Is the value of FAC_CYL_LAM[x] higher than the upper limit?
Yes
Limitation to upper limit LV_FAC_CYL_LAM_LIM_MAX[x] = 1 (1.12.6.4 - 1.12.6.6)
No Limitation (1.12.6.7 - 1.12.6.8)
1.12.7 Anti-Wind-Up
(3)Lower limit run-off?
(3)else
Yes
Anti-wind-up measure (1.12.7.1)
(3)Upper limit run-off?
(3)else
Yes
Anti-wind-up measure (1.12.7.2)
End of controller limitation and Anti-Wind-Up

4.1.13.8 Cylinder individual lambda adaptation

The cylinder individual lambda adaptation shall be carried out in an applied MAF- area. The mechanism consists of a first-order lag element. The output of this lowpass-filter is FAC_LAM_CYL_SEL_ADJ_FIL[x] under nominal conditions (optimal operating temperature) and FAC_LAM_CYL_SEL_ADJ_FIL_CST[x] under engine-cold condition.

***Remark*:**
For application purpose the mechanism can be switched off using LC_ADAPT_COR_OUT:
LC_ADAPT_COR_OUT = 1: Adaptation is switched on.
LC_ADAPT_COR_OUT = 0: Adaptation is switched off, and the adaptation output signal shall be set to zero.

```
if(3)   LC_ADAPT_COR_OUT = 1
then
        if(4)   LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0
        then
                if(5)   STATE_LAM_CYL_SEL_ADJ[i] = 1 [WAIT_ENG_COLD]
                then
                        if(6)   ((C_MFF_LAM_CYL_ADJ_CST_MIN <
                                        MFF_SP_LAM_CYL_SEL <
                                        C_MFF_LAM_CYL_ADJ_CST_MAX) and
                                (C_N_LAM_CYL_ADJ_CST_MIN ≤ N_32 ≤
                                        C_N_LAM_CYL_ADJ_CST_MAX))
                        then
                                STATE_LAM_CYL_SEL_ADJ[i] = 2
                                        [ADJ_ENG_COLD] (4.1.13.8.1)
                        endif(6)
                        if(6)   TEMP_LAM_CYL_SEL ≥ C_TEMP_LAM_CYL_MIN
                        then
                                STATE_LAM_CYL_SEL_ADJ[i] = 3
                                        [WAIT_ENG_NOM] (4.1.13.8.2)
                        else(6)
                                if(7)   (TEMP_LAM_CYL_SEL <
                                                C_TEMP_LAM_CYL_CST_MIN or
                                                (TEMP_LAM_CYL_SEL >
                                                C_TEMP_LAM_CYL_CST_MAX and
                                                TEMP_LAM_CYL_SEL <
                                                C_TEMP_LAM_CYL_MIN) or
                                                LV_LAM_LSCL[i] = 0)
                                then
                                        STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]
                                                (4.1.13.8.3)
                                endif(7)
                        endif(6)
                endif(5)

                if(5)   STATE_LAM_CYL_SEL_ADJ_i = 3 [WAIT_ENG_NOM]
```

```
then
    if$^{(6)}$ LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 0
    then
        if$^{(7)}$ ((N_32 ≥ C_N_MIN_FAC_CYL_SEL_ADJ_L_RNG and N_32 ≤ C_N_MAX_FAC_CYL_SEL_ADJ_L_RNG and MFF_SP_LAM_CYL_SEL ≥ C_MFF_MIN_FAC_CYL_SEL_ADJ_L_RNG and MFF_SP_LAM_CYL_SEL ≤ C_MFF_MAX_FAC_CYL_SEL_ADJ_L_RNG))
        then
            STATE_LAM_CYL_SEL_ADJ[i] = 4 [ADJ_NOM_L_RNG] (4.1.13.8.4)
        else$^{(7)}$
            if$^{(8)}$ ((N_32 ≥ C_N_MIN_FAC_CYL_SEL_ADJ_H_RNG and N_32 ≤ C_N_MAX_FAC_CYL_SEL_ADJ_H_RNG and MFF_SP_LAM_CYL_SEL ≥ C_MFF_MIN_FAC_CYL_SEL_ADJ_H_RNG and MFF_SP_LAM_CYL_SEL ≤ C_MFF_MAX_FAC_CYL_SEL_ADJ_H_RNG))
            then
                STATE_LAM_CYL_SEL_ADJ[i] = 5 [ADJ_NOM_H_RNG] (4.1.13.8.5)
            endif$^{(8)}$
        endif$^{(7)}$
    else$^{(6)}$
        STATE_LAM_CYL_SEL_ADJ[i] = 6 [ADJ_NOM_OFS] (4.1.13.8.6)
    endif$^{(6)}$
    if$^{(6)}$ (C_TEMP_LAM_CYL_CST_MIN < TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_CST_MAX and LV_LAM_LSCL[i] = 1)
    then
        STATE_LAM_CYL_SEL_ADJ[i] = 1 [WAIT_ENG_COLD] (4.1.13.8.7)
    else$^{(6)}$
        if$^{(7)}$ (TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_CST_MIN or (TEMP_LAM_CYL_SEL > C_TEMP_LAM_CYL_CST_MAX and TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_MIN))
        then
            STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS] (4.1.13.8.8)
```

```
            endif(7)
        endif(6)
    endif(5)

    if(5)   STATE_LAM_CYL_SEL_ADJ[i] = 2 [ADJ_ENG_COLD]
    then
        if(6)   (MFF_SP_LAM_CYL_SEL <
                        C_MFF_LAM_CYL_ADJ_CST_MIN or
                        MFF_SP_LAM_CYL_SEL >
                        C_MFF_LAM_CYL_ADJ_CST_MAX or
                        N_32 < C_N_LAM_CYL_ADJ_CST_MIN or
                        N_32 > C_N_LAM_CYL_ADJ_CST_MAX)
        then
            if(7)   (C_TEMP_LAM_CYL_CST_MIN <
                            TEMP_LAM_CYL_SEL <
                            C_TEMP_LAM_CYL_CST_MAX and
                            LV_LAM_LSCL[i] = 1)
            then
                STATE_LAM_CYL_SEL_ADJ[i] = 1
                        [WAIT_ENG_COLD] (4.1.13.8.9)
            endif(7)
        endif(6)
        if(6)   TEMP_LAM_CYL_SEL ≥ C_TEMP_LAM_CYL_MIN
        then
            STATE_LAM_CYL_SEL_ADJ[i] = 3
                    [WAIT_ENG_NOM] (4.1.13.8.10)
        else(6)
            if(7)   (TEMP_LAM_CYL_SEL <
                            C_TEMP_LAM_CYL_CST_MIN or
                            (TEMP_LAM_CYL_SEL >
                            C_TEMP_LAM_CYL_CST_MAX and
                            TEMP_LAM_CYL_SEL <
                            C_TEMP_LAM_CYL_MIN) or
                            LV_LAM_LSCL[i] = 0)
            then
                STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]
                        (4.1.13.8.11)
            endif(7)
        endif(6)
    endif(5)

    if(5)   STATE_LAM_CYL_SEL_ADJ[i] = 4 [ADJ_NOM_L_RNG]
    then
        if(6)   (N_32 <
                        C_N_MIN_FAC_CYL_SEL_ADJ_L_RNG or
                        N_32 > C_N_MAX_FAC_CYL_SEL_ADJ_L_RNG or
                        MFF_SP_LAM_CYL_SEL <
                        C_MFF_MIN_FAC_CYL_SEL_ADJ_L_RNG or
                        MFF_SP_LAM_CYL_SEL >
```

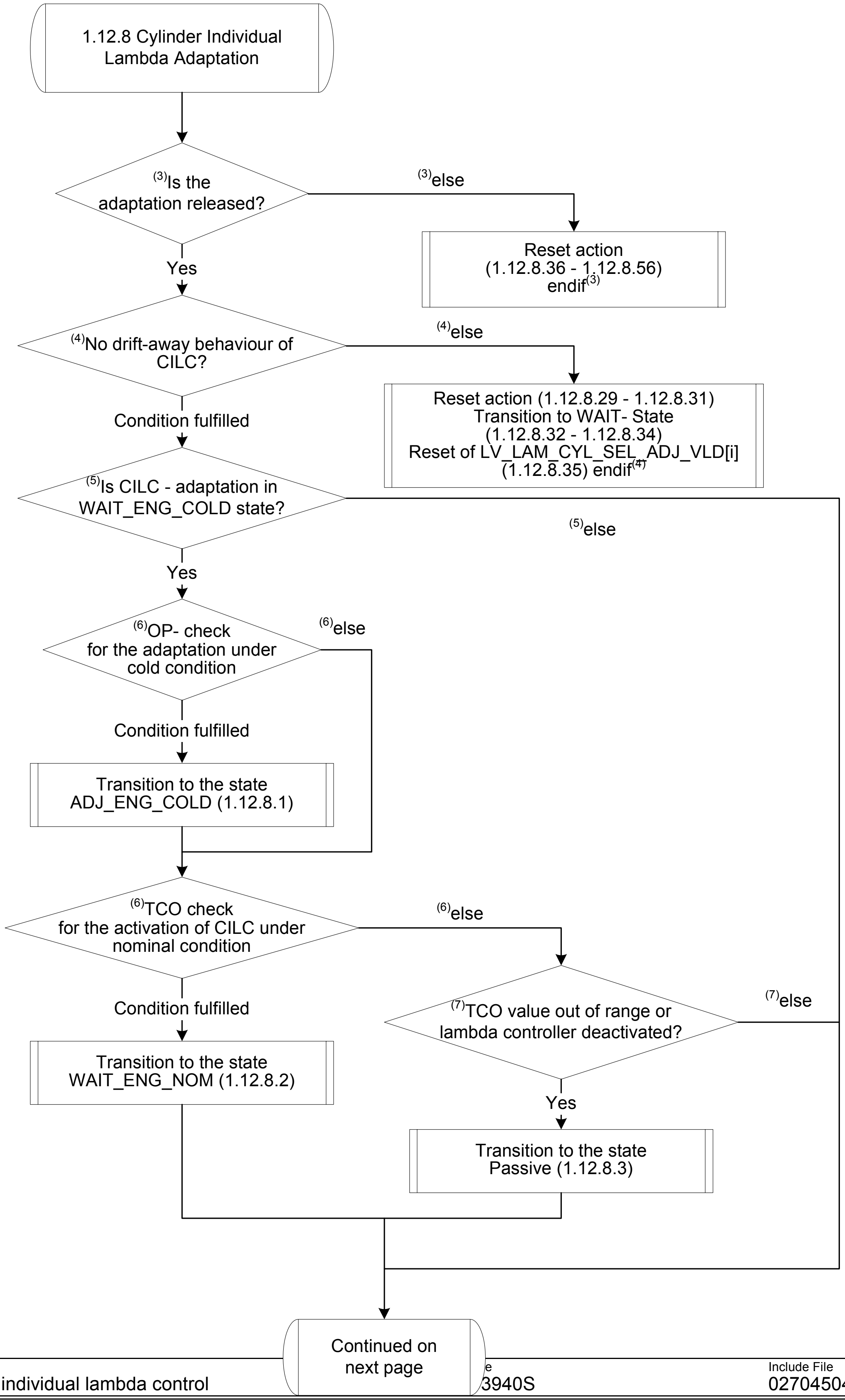
```
                    C_MFF_MAX_FAC_CYL_SEL_ADJ_L_RNG or
                    LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 1)

        then
            if(7)   TEMP_LAM_CYL_SEL ≥
                        C_TEMP_LAM_CYL_MIN
            then
                STATE_LAM_CYL_SEL_ADJ[i] = 3
                    [WAIT_ENG_NOM] (4.1.13.8.12)
            endif(7)
        endif(6)
        if(6)   (C_TEMP_LAM_CYL_CST_MIN <
                    TEMP_LAM_CYL_SEL <
                    C_TEMP_LAM_CYL_CST_MAX and
                    LV_LAM_LSCL[i] = 1)
        then
            STATE_LAM_CYL_SEL_ADJ[i] = 1
                [WAIT_ENG_COLD] (4.1.13.8.13)
        else(6)
            if(7)   (TEMP_LAM_CYL_SEL <
                        C_TEMP_LAM_CYL_CST_MIN or
                        (TEMP_LAM_CYL_SEL >
                        C_TEMP_LAM_CYL_CST_MAX and
                        TEMP_LAM_CYL_SEL <
                        C_TEMP_LAM_CYL_MIN))
            then
                STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]
                    (4.1.13.8.14)
            endif(7)
        endif(6)
    endif(5)

    if(5)   STATE_LAM_CYL_SEL_ADJ[i] = 5 [ADJ_NOM_H_RNG]
    then
        if(6) (N_32 < C_N_MIN_FAC_CYL_SEL_ADJ_H_RNG or
                N_32 > C_N_MAX_FAC_CYL_SEL_ADJ_H_RNG or
                MFF_SP_LAM_CYL_SEL <
                C_MFF_MIN_FAC_CYL_SEL_ADJ_H_RNG or
                MFF_SP_LAM_CYL_SEL >
                C_MFF_MAX_FAC_CYL_SEL_ADJ_H_RNG or
                LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 1)
        then
            if(7)   TEMP_LAM_CYL_SEL ≥
                        C_TEMP_LAM_CYL_MIN
            then
                STATE_LAM_CYL_SEL_ADJ[i] = 3
                    [WAIT_ENG_NOM] (4.1.13.8.15)
            endif(7)
        endif(6)
```

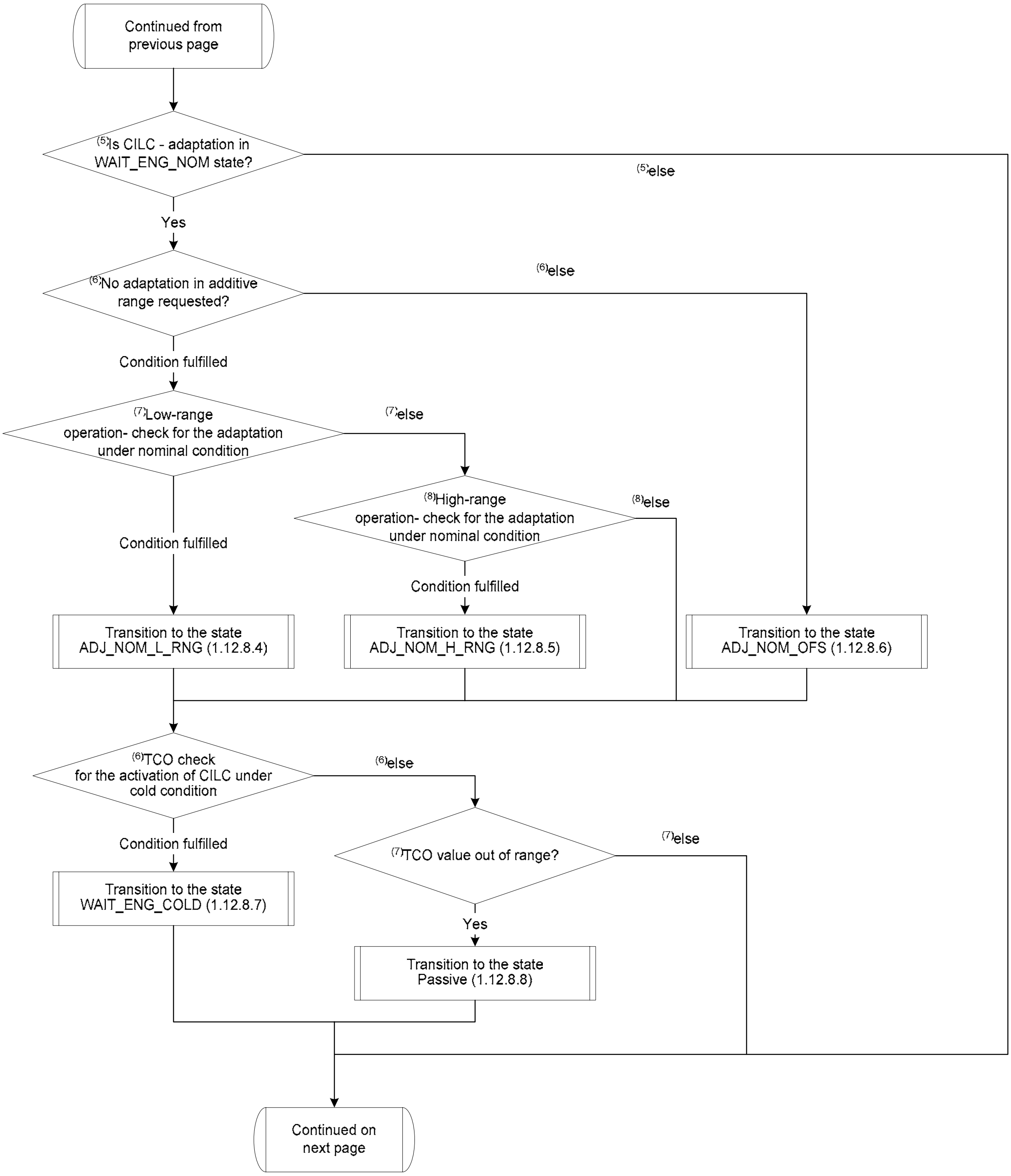
```
        if$^{(6)}$    (C_TEMP_LAM_CYL_CST_MIN <
                                   TEMP_LAM_CYL_SEL <
                         C_TEMP_LAM_CYL_CST_MAX and
                                    LV_LAM_LSCL[i] = 1)
        then
              STATE_LAM_CYL_SEL_ADJ[i] = 1
                           [WAIT_ENG_COLD] (4.1.13.8.16)
        else$^{(6)}$
              if$^{(7)}$    (TEMP_LAM_CYL_SEL <
                             C_TEMP_LAM_CYL_CST_MIN or
                                  (TEMP_LAM_CYL_SEL >
                         C_TEMP_LAM_CYL_CST_MAX and
                                   TEMP_LAM_CYL_SEL <
                                C_TEMP_LAM_CYL_MIN))
              then
                    STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]
                                          (4.1.13.8.17)
              endif$^{(7)}$
        endif$^{(6)}$
  endif$^{(5)}$
  if$^{(5)}$    STATE_LAM_CYL_SEL_ADJ[i] = 6 [ADJ_NOM_OFS]
  then
        if$^{(6)}$    LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 0
        then
              if$^{(7)}$    TEMP_LAM_CYL_SEL ≥
                                  C_TEMP_LAM_CYL_MIN
              then
                    STATE_LAM_CYL_SEL_ADJ[i] = 3
                            [WAIT_ENG_NOM] (4.1.13.8.18)
              endif$^{(7)}$
        endif$^{(6)}$
        if$^{(6)}$    (C_TEMP_LAM_CYL_CST_MIN <
                                   TEMP_LAM_CYL_SEL <
                         C_TEMP_LAM_CYL_CST_MAX and
                                    LV_LAM_LSCL[i] = 1)
        then
              STATE_LAM_CYL_SEL_ADJ[i] = 1
                           [WAIT_ENG_COLD] (4.1.13.8.19)
        else$^{(6)}$
              if$^{(7)}$    (TEMP_LAM_CYL_SEL <
                             C_TEMP_LAM_CYL_CST_MIN or
                                  (TEMP_LAM_CYL_SEL >
                         C_TEMP_LAM_CYL_CST_MAX and
                                   TEMP_LAM_CYL_SEL <
                                C_TEMP_LAM_CYL_MIN))
              then
                    STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]
                                          (4.1.13.8.20)
              endif$^{(7)}$
```

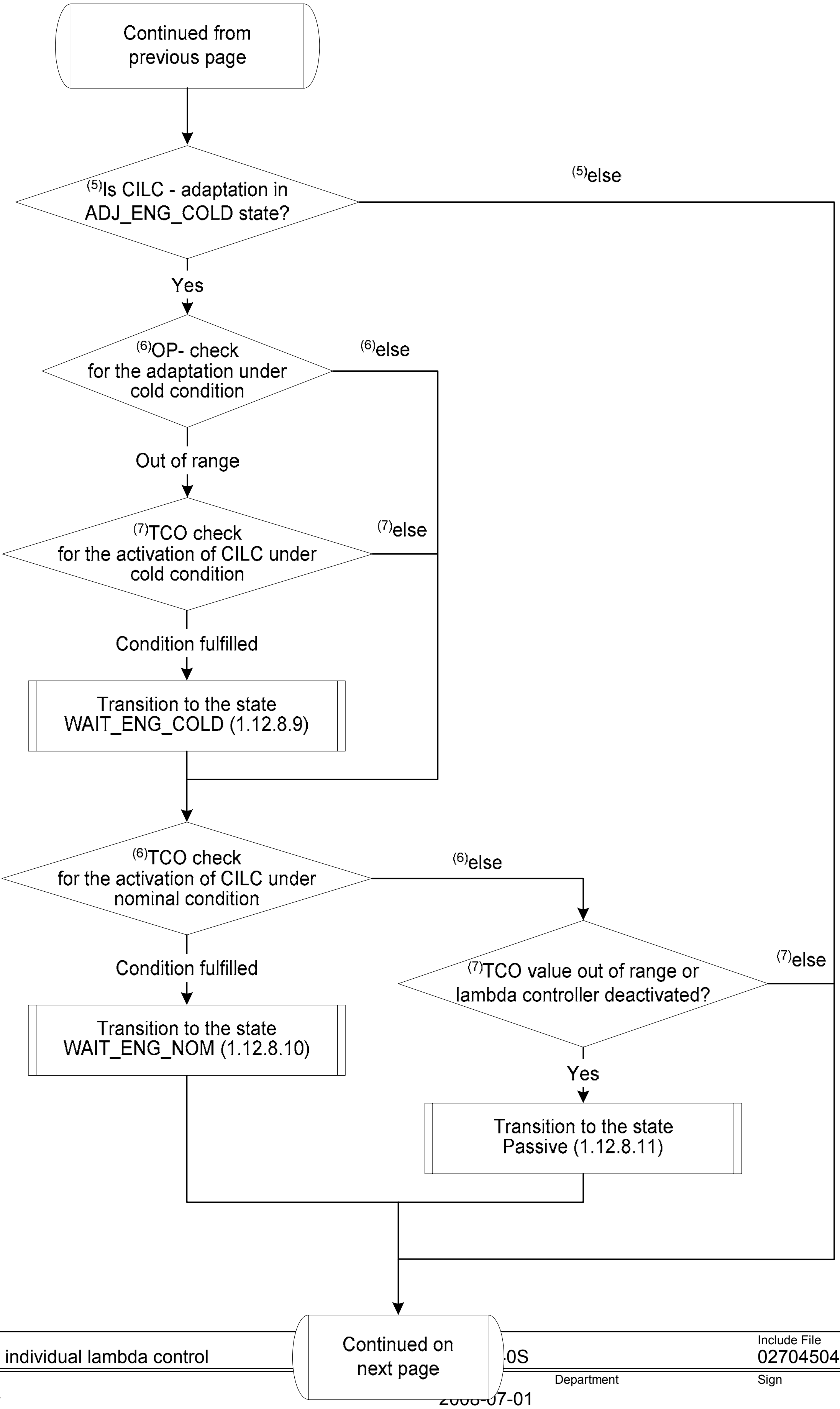
```
        endif$^{(6)}$
endif$^{(5)}$
[Remark: the cylinder based limitation (index x) shall be
assigned to the corresponding exhaust bank (index i)]

if$^{(5)}$ (STATE_LAM_CYL_SEL_ADJ[i] = 1 [WAIT_ENG_COLD] or
        STATE_LAM_CYL_SEL_ADJ[i] = 2 [ADJ_ENG_COLD])
then
        FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$ = 0                                  (4.1.13.8.21)
        if$^{(6)}$ LC_FAC_LAM_CYL_SEL_ADJ_P_I_ENA = 1
        then
                FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]$_k$ =
                        FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]$_{k-1}$ +
                        (FAC_CYL_LAM[x]$_k$ –
                        FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]$_{k-1}$)
                        *C_CRLC_LAM_CYL_CST_FIL                     (4.1.13.8.22)
        else$^{(6)}$
                FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]$_k$ =
                        FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]$_{k-1}$ +
                        (FAC_CYL_LAM_I_MV_SHIFT[x]$_k$ –
                        FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]$_{k-1}$)
                        *C_CRLC_LAM_CYL_CST_FIL                     (4.1.13.8.23)
        endif$^{(6)}$
else$^{(5)}$
        if$^{(6)}$ (STATE_LAM_CYL_SEL_ADJ[i] = 3
                        [WAIT_ENG_NOM] or
                STATE_LAM_CYL_SEL_ADJ[i] = 4
                        [ADJ_NOM_L_RNG] or
                STATE_LAM_CYL_SEL_ADJ[i] = 5
                        [ADJ_NOM_H_RNG] or
                STATE_LAM_CYL_SEL_ADJ[i] = 6
                        [ADJ_NOM_OFS])
        then
                FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]$_k$ = 0                      (4.1.13.8.24)
                if$^{(7)}$ LC_FAC_LAM_CYL_SEL_ADJ_P_I_ENA = 1
                then
                        FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$ =
                                FAC_LAM_CYL_SEL_ADJ_FIL[x]$_{k-1}$ +
                                (FAC_CYL_LAM[x]$_k$ –
                                FAC_LAM_CYL_SEL_ADJ_FIL[x]$_{k-1}$)*
                                C_CRLC_CYL_FIL                      (4.1.13.8.25)
                else$^{(7)}$
                        FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$ =
                                FAC_LAM_CYL_SEL_ADJ_FIL[x]$_{k-1}$ +
                                (FAC_CYL_LAM_I_MV_SHIFT[x]$_k$ –
                                FAC_LAM_CYL_SEL_ADJ_FIL[x]$_{k-1}$)*
                                C_CRLC_CYL_FIL                      (4.1.13.8.26)
```

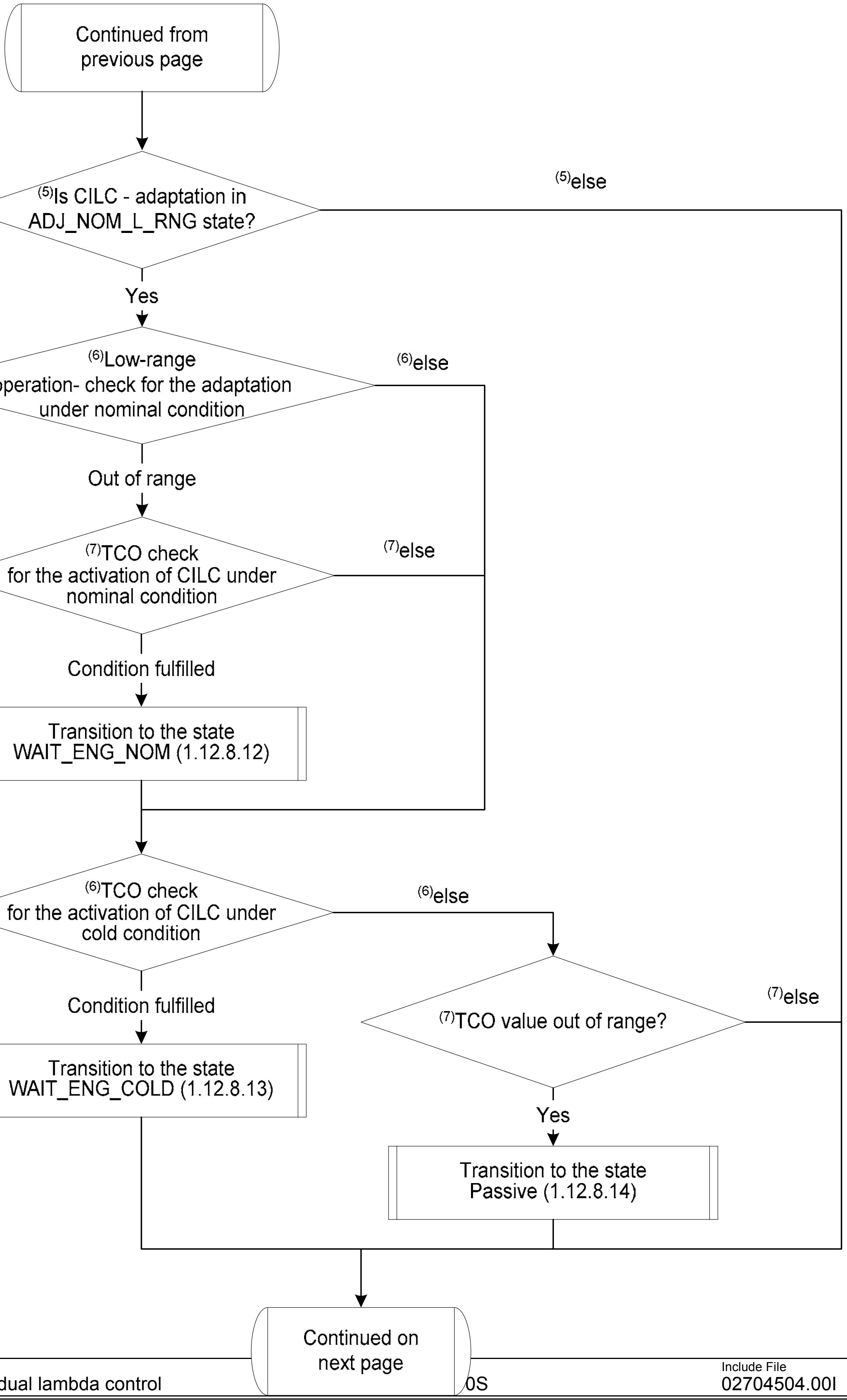
```
                endif$^{(7)}$
            else$^{(6)}$
                FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$ = 0                      (4.1.13.8.27)
                FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]$_k$ = 0                  (4.1.13.8.28)
            endif$^{(6)}$
        endif$^{(5)}$
    else$^{(4)}$
        FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0                                   (4.1.13.8.29)
        FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0                               (4.1.13.8.30)
        CTR_T_LAM_CYL_ADJ[i] = 0                                         (4.1.13.8.31)
        if$^{(5)}$ (C_TEMP_LAM_CYL_CST_MIN <
                        TEMP_LAM_CYL_SEL <
                        C_TEMP_LAM_CYL_CST_MAX and
                        LV_LAM_LSCL[i] = 1)
        then
            STATE_LAM_CYL_SEL_ADJ[i] = 1
                        [WAIT_ENG_COLD]                                  (4.1.13.8.32)
        else$^{(5)}$
            if$^{(6)}$ TEMP_LAM_CYL_SEL $\geq$ C_TEMP_LAM_CYL_MIN
            then
                STATE_LAM_CYL_SEL_ADJ[i] = 3
                        [WAIT_ENG_NOM]                                   (4.1.13.8.33)
            else$^{(6)}$
                STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]                       (4.1.13.8.34)
            endif$^{(6)}$
        endif$^{(6)}$
        LV_LAM_CYL_SEL_ADJ_VLD[i] = 0                                    (4.1.13.8.35)
    endif$^{(4)}$
else$^{(3)}$
    MFF_ADD_CYL_LAM_COR[x] = 0                                           (4.1.13.8.36)
    MFF_ADD_CYL_LAM_COR[x]$_{LDC}$ = 0                                   (4.1.13.8.37)
    FAC_LAM_CYL_SEL_ADJ[x] = 0                                           (4.1.13.8.38)
    FAC_LAM_CYL_SEL_ADJ_L_RNG[x] = 0                                     (4.1.13.8.39)
    FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{LDC}$ = 0                             (4.1.13.8.40)
    FAC_LAM_CYL_SEL_ADJ_H_RNG[x] = 0                                     (4.1.13.8.41)
    FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{LDC}$ = 0
```

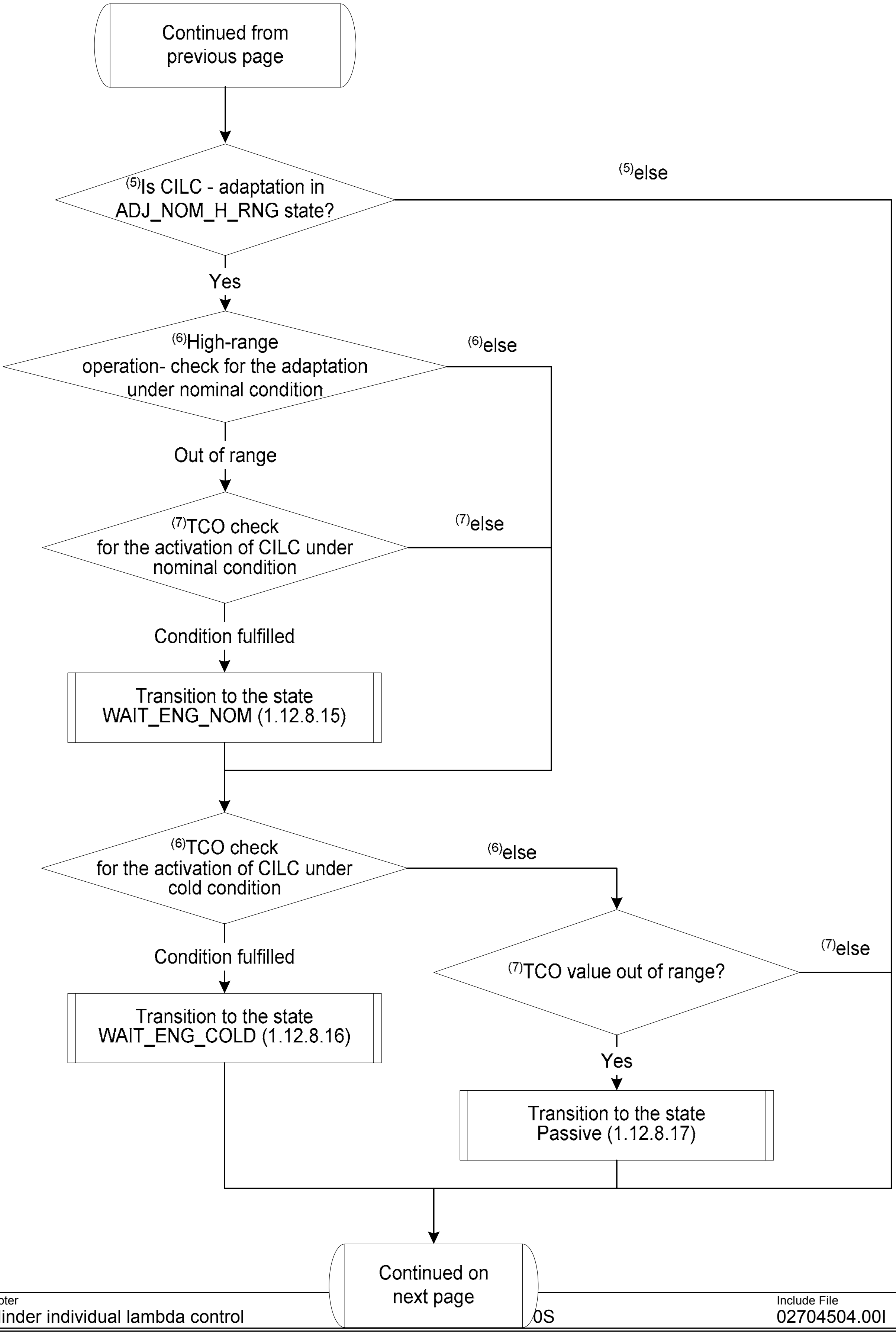
```
                                                                (4.1.13.8.42)
            FAC_LAM_CYL_SEL_ADJ_CST[x] = 0                      (4.1.13.8.43)
            FAC_LAM_CYL_SEL_ADJ_CST[x]$_{LDC}$ = 0              (4.1.13.8.44)
            FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0                      (4.1.13.8.45)
            FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0                  (4.1.13.8.46)
            CTR_T_LAM_CYL_ADJ[i] = 0                            (4.1.13.8.47)
            STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]                  (4.1.13.8.48)
            LV_LAM_CYL_SEL_ADJ_VLD[i] = 0                       (4.1.13.8.49)
            LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] = 0                 (4.1.13.8.50)
            LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] = 0                 (4.1.13.8.51)
            LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i]$_{LDC}$ = 0         (4.1.13.8.52)
            LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i]$_{LDC}$ = 0         (4.1.13.8.53)
            LV_FAC_LAM_CYL_ADJ_CST_LIM[i] = 0                   (4.1.13.8.54)
            LV_FAC_LAM_CYL_SEL_ADJ_LIM[i] = 0                   (4.1.13.8.55)
            LV_MFF_ADD_CYL_LAM_COR_LIM[i] = 0                   (4.1.13.8.56)
        endif(3)
    else(2)
        FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0                          (4.1.13.8.57)
        FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0                      (4.1.13.8.58)
        CTR_T_LAM_CYL_ADJ[i] = 0                                (4.1.13.8.59)
        STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS]                      (4.1.13.8.60)
        LV_LAM_CYL_SEL_ADJ_VLD[i] = 0                           (4.1.13.8.61)
    endif(2)
endif(1)
```

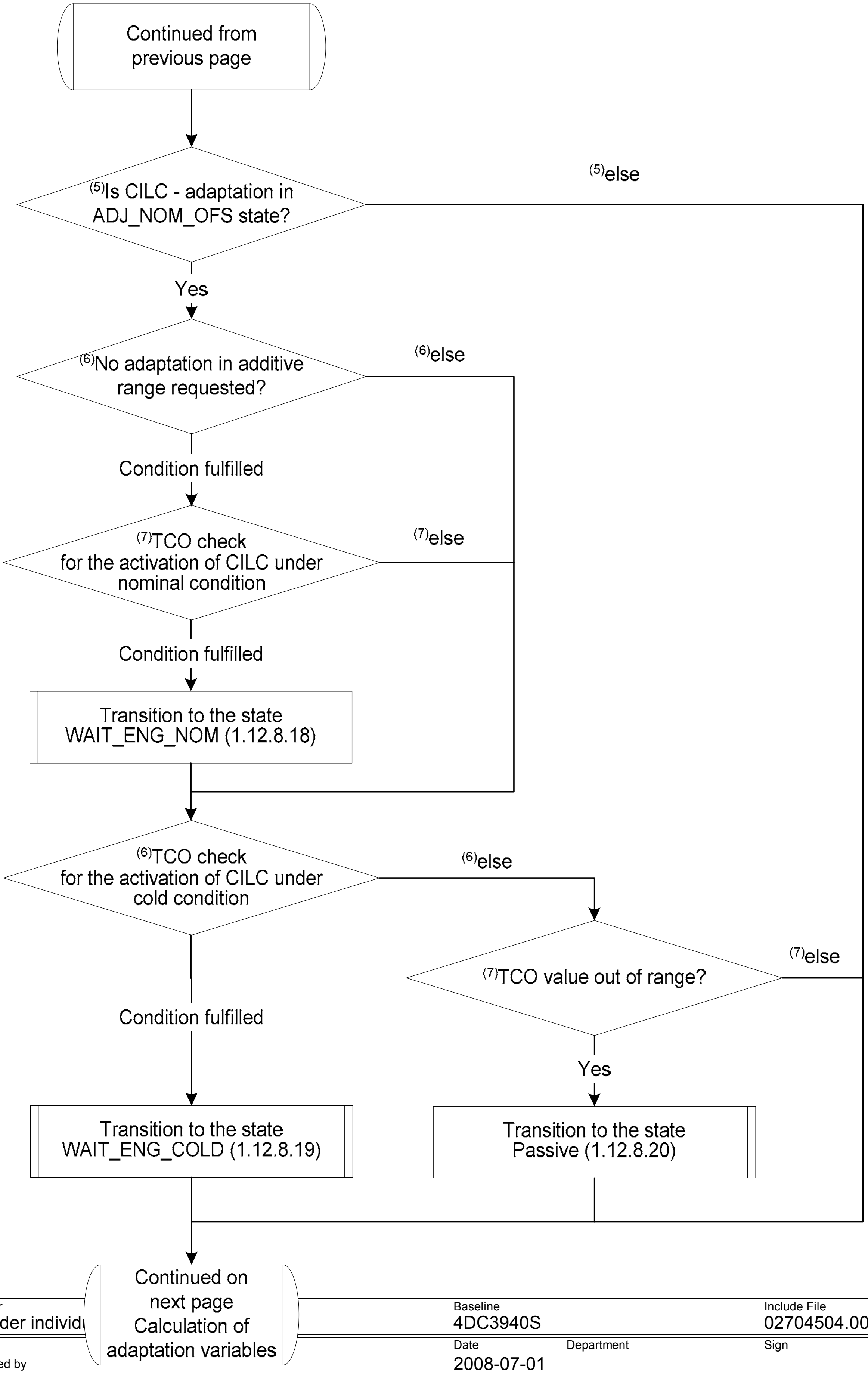

1.12.8 Cylinder Individual Lambda Adaptation
(3)Is the adaptation released?
(3)else
Reset action (1.12.8.36 - 1.12.8.56) endif(3)
Yes
(4)No drift-away behaviour of CILC?
(4)else
Reset action (1.12.8.29 - 1.12.8.31) Transition to WAIT- State (1.12.8.32 - 1.12.8.34) Reset of LV_LAM_CYL_SEL_ADJ_VLD[i] (1.12.8.35) endif(4)
Condition fulfilled
(5)Is CILC - adaptation in WAIT_ENG_COLD state?
(5)else
Yes
(6)OP- check for the adaptation under cold condition
(6)else
Condition fulfilled
Transition to the state ADJ_ENG_COLD (1.12.8.1)
(6)TCO check for the activation of CILC under nominal condition
(6)else
Condition fulfilled
(7)TCO value out of range or lambda controller deactivated?
(7)else
Transition to the state WAIT_ENG_NOM (1.12.8.2)
Yes
Transition to the state Passive (1.12.8.3)
Continued on next page

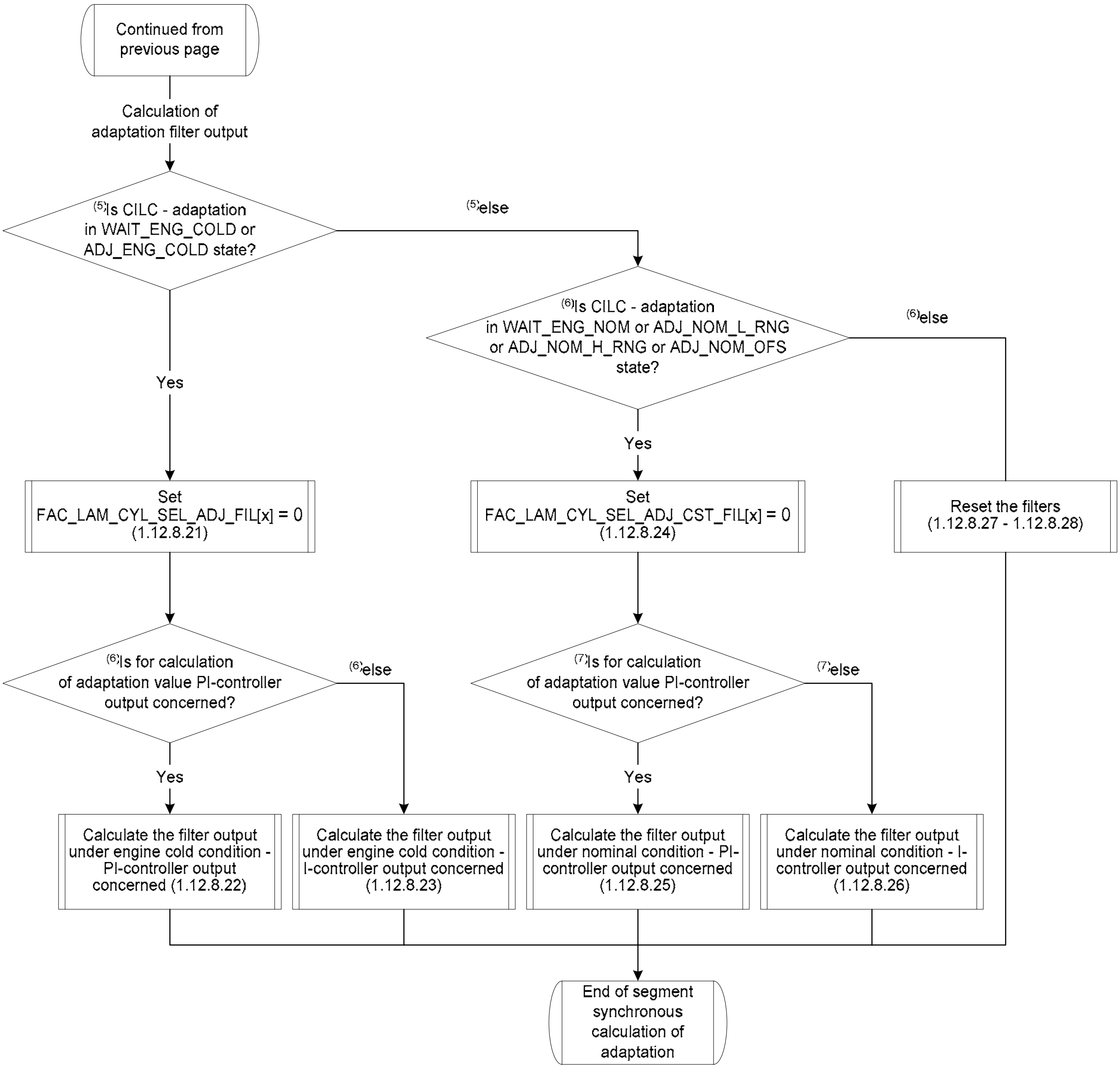

Continued from previous page
(5)Is CILC - adaptation in WAIT_ENG_NOM state?
(5)else
Yes
(6)No adaptation in additive range requested?
(6)else
Condition fulfilled
(7)Low-range operation- check for the adaptation under nominal condition
(7)else
(8)High-range operation- check for the adaptation under nominal condition
(8)else
Condition fulfilled
Condition fulfilled
Transition to the state ADJ_NOM_L_RNG (1.12.8.4)
Transition to the state ADJ_NOM_H_RNG (1.12.8.5)
Transition to the state ADJ_NOM_OFS (1.12.8.6)
(6)TCO check for the activation of CILC under cold condition
(6)else
Condition fulfilled
(7)TCO value out of range?
(7)else
Transition to the state WAIT_ENG_COLD (1.12.8.7)
Yes
Transition to the state Passive (1.12.8.8)
Continued on next page


Continued from previous page
(5)Is CILC - adaptation in ADJ_ENG_COLD state?
(5)else
Yes
(6)OP- check for the adaptation under cold condition
(6)else
Out of range
(7)TCO check for the activation of CILC under cold condition
(7)else
Condition fulfilled
Transition to the state WAIT_ENG_COLD (1.12.8.9)
(6)TCO check for the activation of CILC under nominal condition
(6)else
Condition fulfilled
Transition to the state WAIT_ENG_NOM (1.12.8.10)
(7)TCO value out of range or lambda controller deactivated?
(7)else
Yes
Transition to the state Passive (1.12.8.11)
Continued on next page


Continued from previous page
(5)Is CILC - adaptation in ADJ_NOM_L_RNG state?
(5)else
Yes
(6)Low-range operation- check for the adaptation under nominal condition
(6)else
Out of range
(7)TCO check for the activation of CILC under nominal condition
(7)else
Condition fulfilled
Transition to the state WAIT_ENG_NOM (1.12.8.12)
(6)TCO check for the activation of CILC under cold condition
(6)else
Condition fulfilled
Transition to the state WAIT_ENG_COLD (1.12.8.13)
(7)TCO value out of range?
(7)else
Yes
Transition to the state Passive (1.12.8.14)
Continued on next page

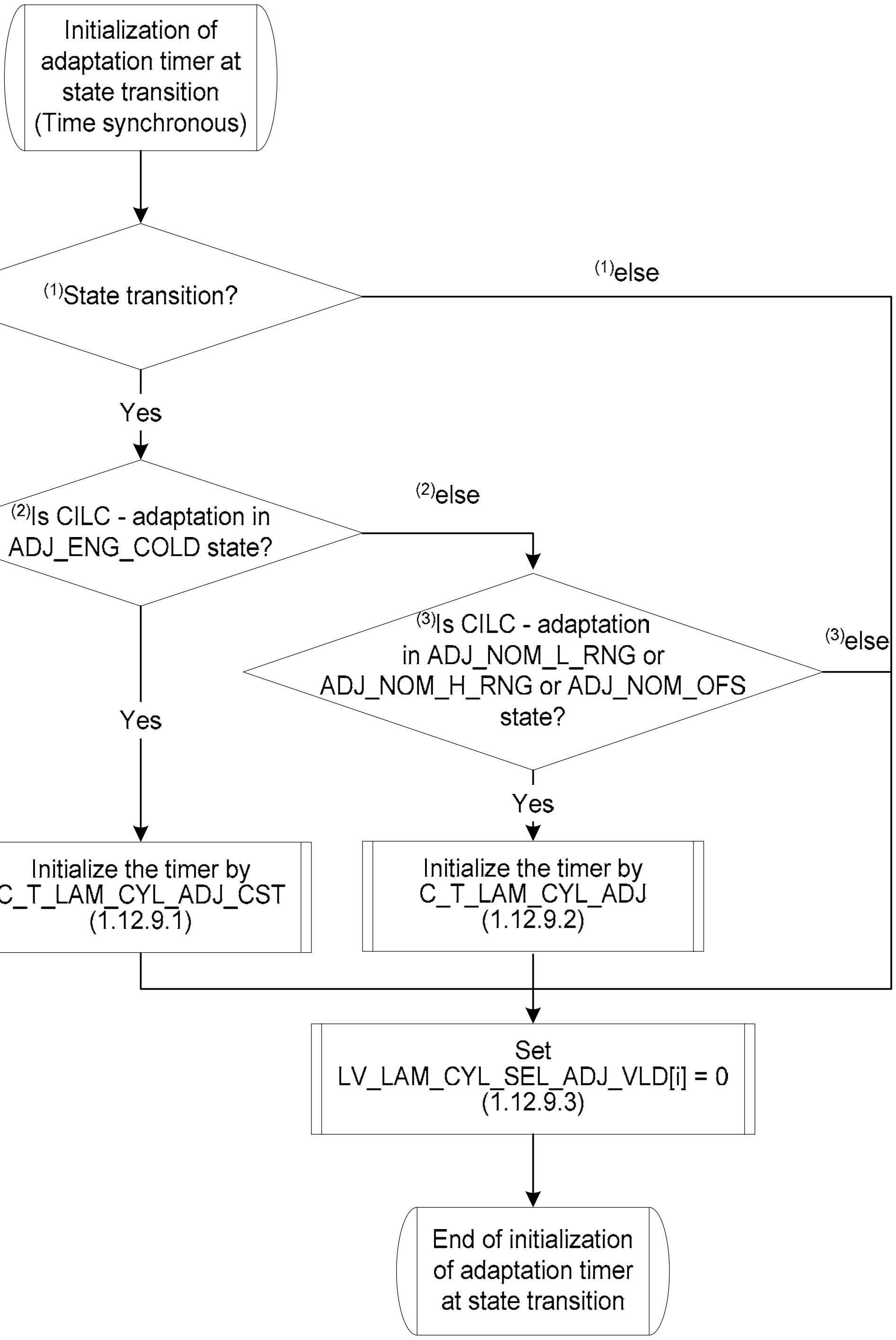

Continued from previous page
(5)Is CILC - adaptation in ADJ_NOM_H_RNG state?
(5)else
Yes
(6)High-range operation- check for the adaptation under nominal condition
(6)else
Out of range
(7)TCO check for the activation of CILC under nominal condition
(7)else
Condition fulfilled
Transition to the state WAIT_ENG_NOM (1.12.8.15)
(6)TCO check for the activation of CILC under cold condition
(6)else
Condition fulfilled
Transition to the state WAIT_ENG_COLD (1.12.8.16)
(7)TCO value out of range?
(7)else
Yes
Transition to the state Passive (1.12.8.17)
Continued on next page

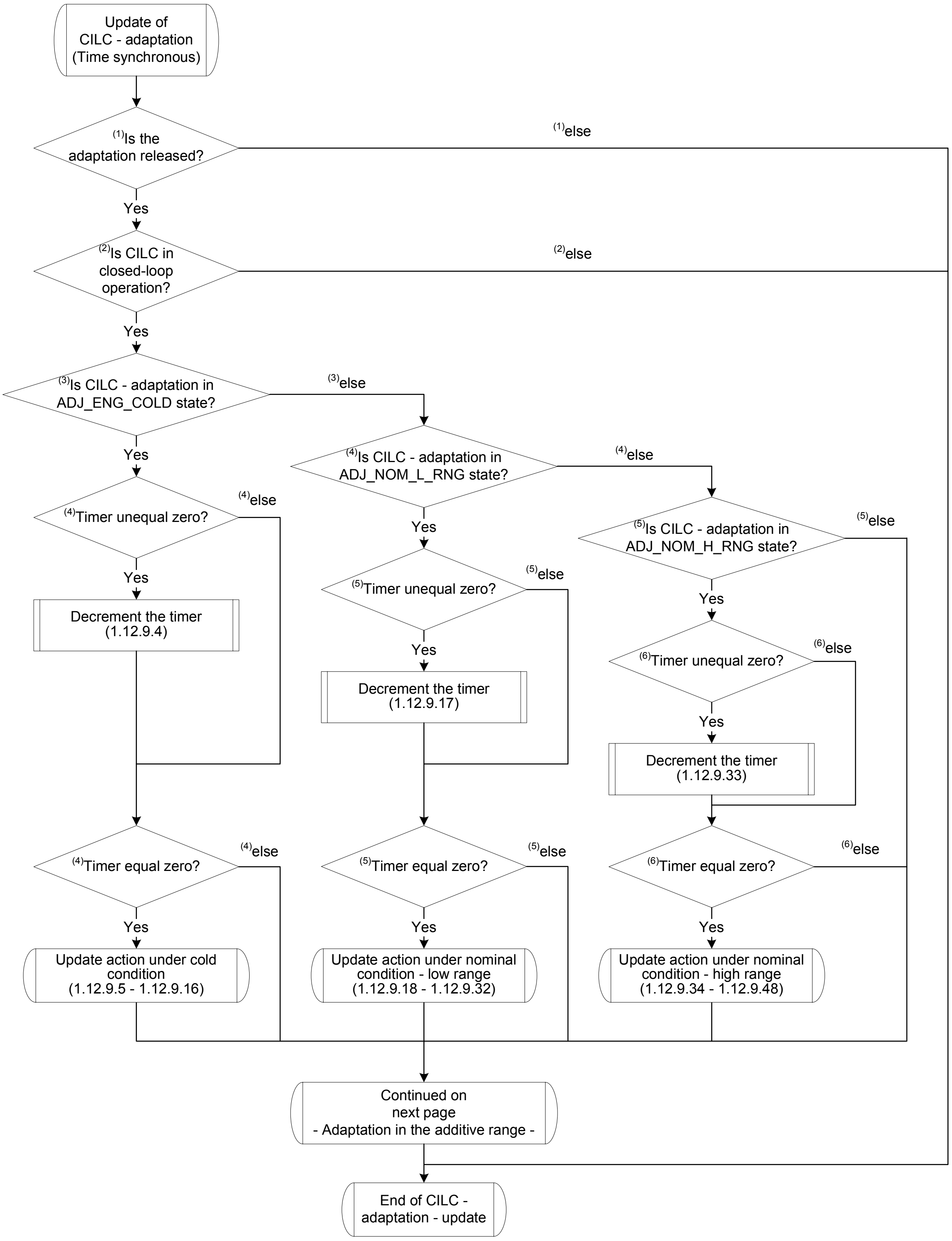

Continued from previous page
(5)Is CILC - adaptation in ADJ_NOM_OFS state?
(5)else
Yes
(6)No adaptation in additive range requested?
(6)else
Condition fulfilled
(7)TCO check for the activation of CILC under nominal condition
(7)else
Condition fulfilled
Transition to the state WAIT_ENG_NOM (1.12.8.18)
(6)TCO check for the activation of CILC under cold condition
(6)else
(7)TCO value out of range?
(7)else
Condition fulfilled
Yes
Transition to the state WAIT_ENG_COLD (1.12.8.19)
Transition to the state Passive (1.12.8.20)
Continued on next page Calculation of adaptation variables

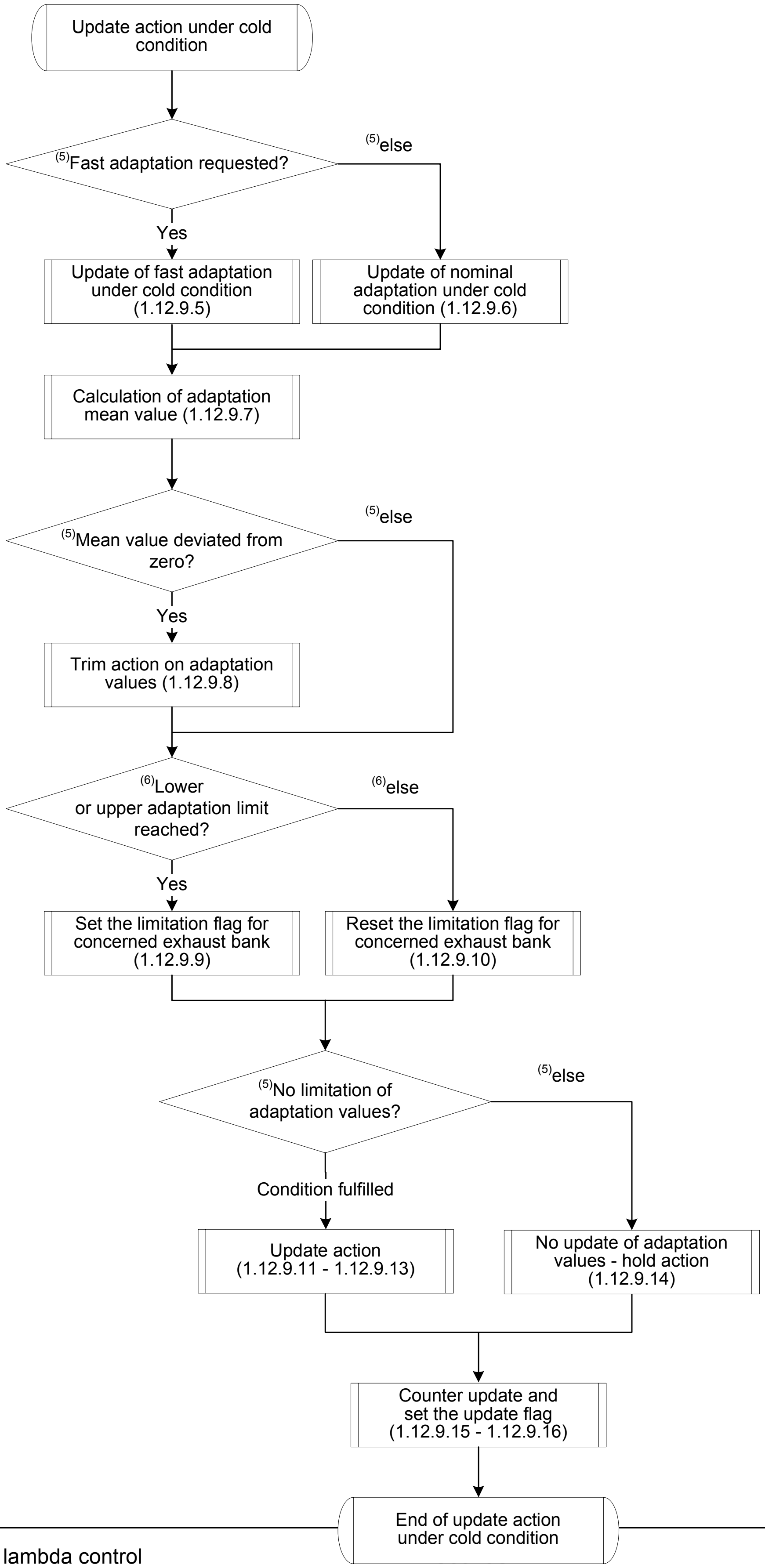

Continued from previous page
Calculation of adaptation filter output
(5)Is CILC - adaptation in WAIT_ENG_COLD or ADJ_ENG_COLD state?
(5)else
Yes
(6)Is CILC - adaptation in WAIT_ENG_NOM or ADJ_NOM_L_RNG or ADJ_NOM_H_RNG or ADJ_NOM_OFS state?
(6)else
Yes
Set FAC_LAM_CYL_SEL_ADJ_FIL[x] = 0 (1.12.8.21)
Set FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] = 0 (1.12.8.24)
Reset the filters (1.12.8.27 - 1.12.8.28)
(6)Is for calculation of adaptation value PI-controller output concerned?
(6)else
(7)Is for calculation of adaptation value PI-controller output concerned?
(7)else
Yes
Yes
Calculate the filter output under engine cold condition - PI-controller output concerned (1.12.8.22)
Calculate the filter output under engine cold condition - I-controller output concerned (1.12.8.23)
Calculate the filter output under nominal condition - PI-controller output concerned (1.12.8.25)
Calculate the filter output under nominal condition - I-controller output concerned (1.12.8.26)
End of segment synchronous calculation of adaptation

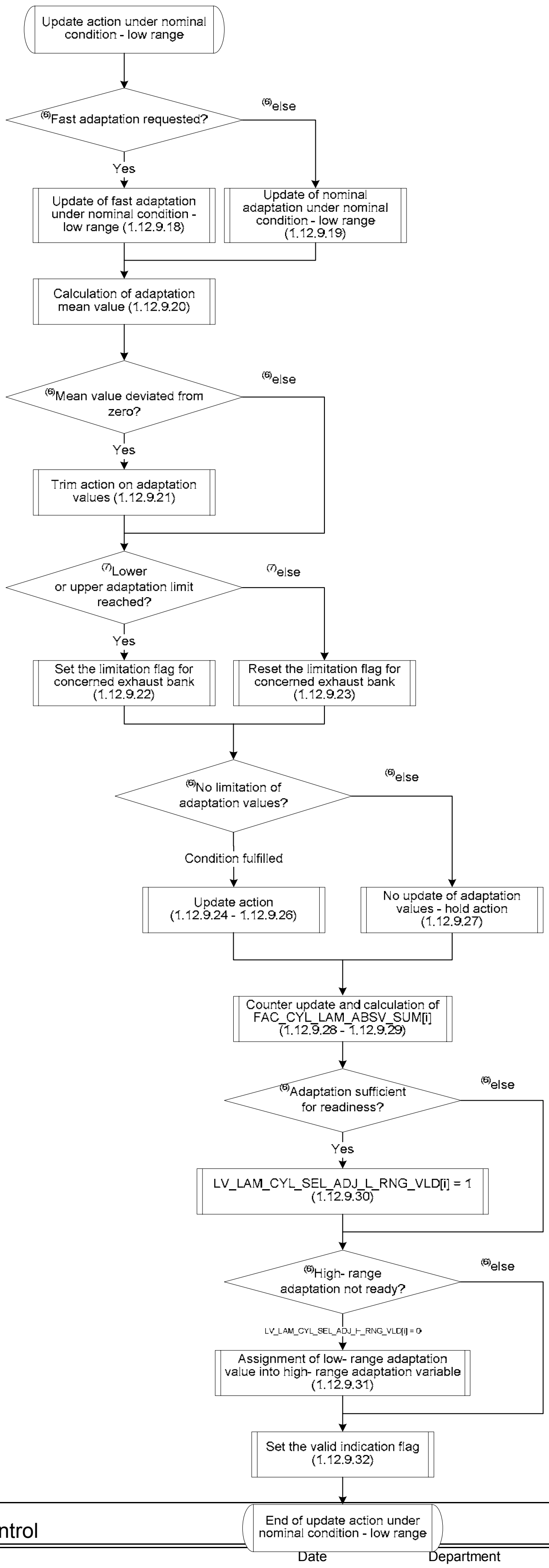
4.1.13.9 Update of cylinder individual lambda adaptation values

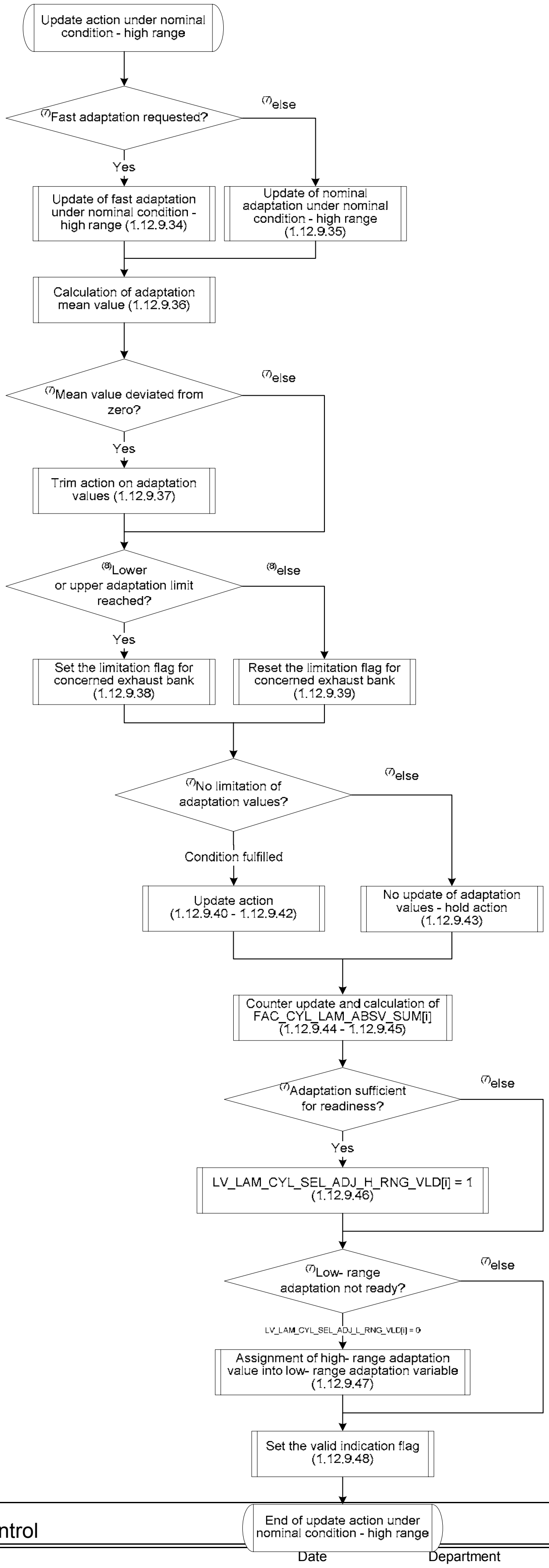
The computation of the cylinder individual lambda adaptation shall be updated in the time-cycle C_T_LAM_CYL_ADJ_CST/C_T_LAM_CYL_ADJ. When the computation was carried out, then the initial value of cylinder individual lambda-controller, FAC_CYL_LAM_INT[x], and low pass filter,
FAC_LAM_CYL_SEL_ADJ_FIL[x]/FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] shall be shifted.
On the deactivation condition all cylinder individual filter outputs,
FAC_LAM_CYL_SEL_ADJ_FIL[x] and the timer CTR_T_LAM_CYL_ADJ[i] shall be set to zero.

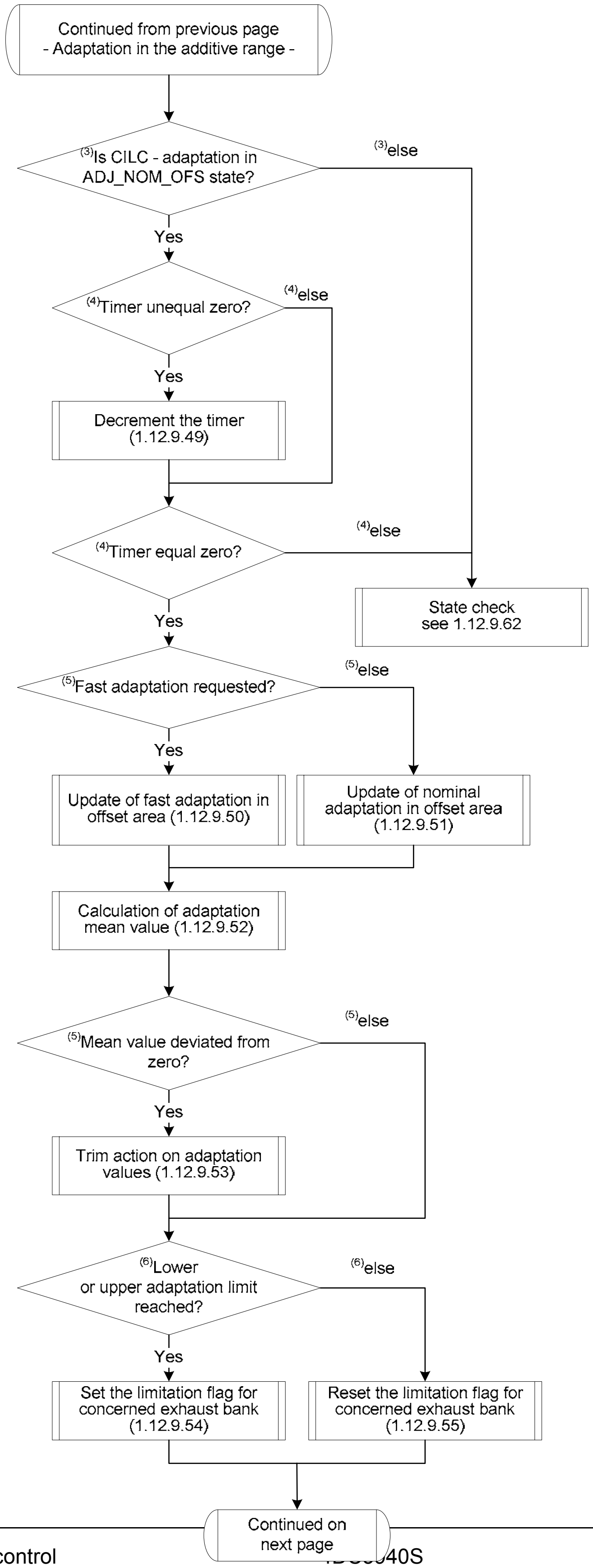

Initialization of adaptation timer at state transition (Time synchronous)
(1)State transition?
(1)else
Yes
(2)Is CILC - adaptation in ADJ_ENG_COLD state?
(2)else
(3)Is CILC - adaptation in ADJ_NOM_L_RNG or ADJ_NOM_H_RNG or ADJ_NOM_OFS state?
(3)else
Yes
Yes
Initialize the timer by C_T_LAM_CYL_ADJ_CST (1.12.9.1)
Initialize the timer by C_T_LAM_CYL_ADJ (1.12.9.2)
Set LV_LAM_CYL_SEL_ADJ_VLD[i] = 0 (1.12.9.3)
End of initialization of adaptation timer at state transition

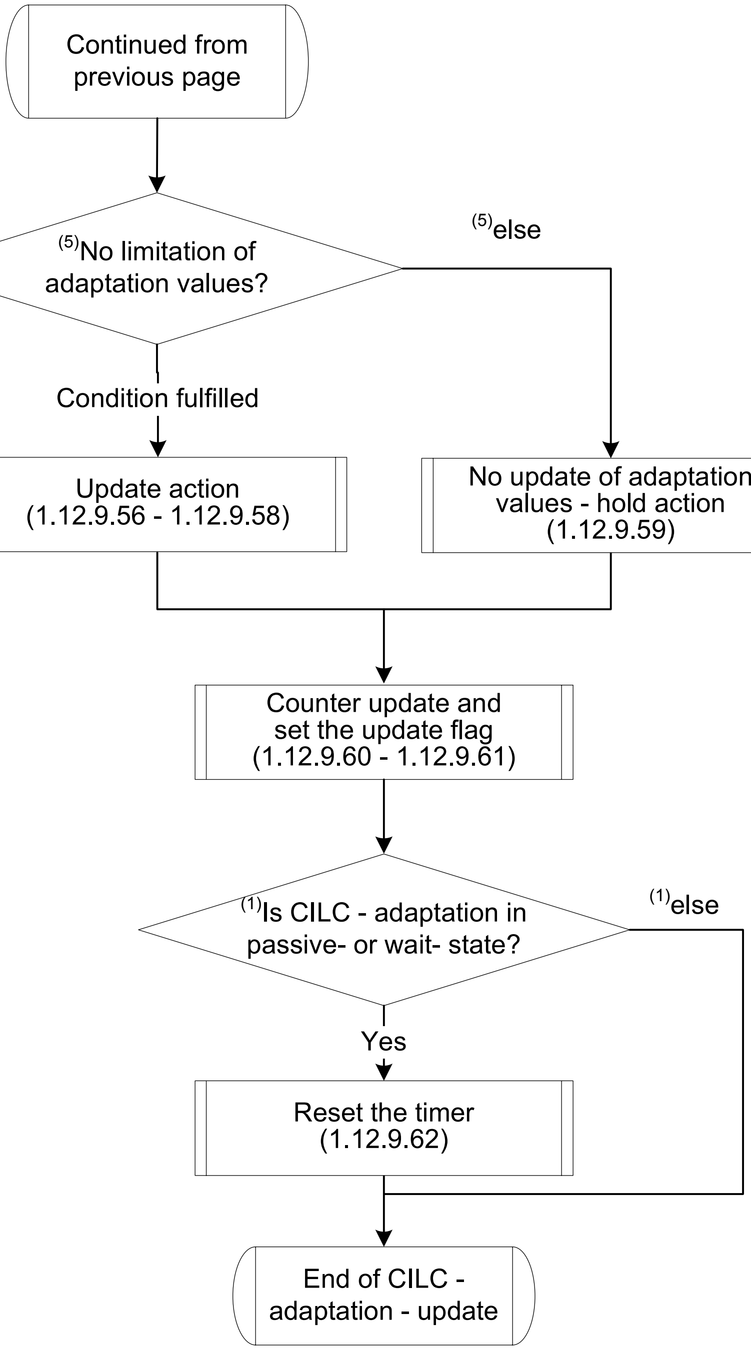

Update of
CILC - adaptation
(Time synchronous)
(1)Is the adaptation released?
(1)else
Yes
(2)Is CILC in closed-loop operation?
(2)else
Yes
(3)Is CILC - adaptation in ADJ_ENG_COLD state?
(3)else
Yes
(4)Timer unequal zero?
(4)else
Yes
Decrement the timer (1.12.9.4)
(4)Timer equal zero?
(4)else
Yes
Update action under cold condition (1.12.9.5 - 1.12.9.16)
(4)Is CILC - adaptation in ADJ_NOM_L_RNG state?
(4)else
Yes
(5)Timer unequal zero?
(5)else
Yes
Decrement the timer (1.12.9.17)
(5)Timer equal zero?
(5)else
Yes
Update action under nominal condition - low range (1.12.9.18 - 1.12.9.32)
(5)Is CILC - adaptation in ADJ_NOM_H_RNG state?
(5)else
Yes
(6)Timer unequal zero?
(6)else
Yes
Decrement the timer (1.12.9.33)
(6)Timer equal zero?
(6)else
Yes
Update action under nominal condition - high range (1.12.9.34 - 1.12.9.48)
Continued on next page - Adaptation in the additive range -
End of CILC - adaptation - update


Update action under cold condition
(5)Fast adaptation requested?
(5)else
Yes
Update of fast adaptation under cold condition (1.12.9.5)
Update of nominal adaptation under cold condition (1.12.9.6)
Calculation of adaptation mean value (1.12.9.7)
(5)Mean value deviated from zero?
(5)else
Yes
Trim action on adaptation values (1.12.9.8)
(6)Lower or upper adaptation limit reached?
(6)else
Yes
Set the limitation flag for concerned exhaust bank (1.12.9.9)
Reset the limitation flag for concerned exhaust bank (1.12.9.10)
(5)No limitation of adaptation values?
(5)else
Condition fulfilled
Update action (1.12.9.11 - 1.12.9.13)
No update of adaptation values - hold action (1.12.9.14)
Counter update and set the update flag (1.12.9.15 - 1.12.9.16)
End of update action under cold condition


Update action under nominal condition - low range
(6)Fast adaptation requested?
(6)else
Yes
Update of fast adaptation under nominal condition - low range (1.12.9.18)
Update of nominal adaptation under nominal condition - low range (1.12.9.19)
Calculation of adaptation mean value (1.12.9.20)
(6)Mean value deviated from zero?
(6)else
Yes
Trim action on adaptation values (1.12.9.21)
(7)Lower or upper adaptation limit reached?
(7)else
Yes
Set the limitation flag for concerned exhaust bank (1.12.9.22)
Reset the limitation flag for concerned exhaust bank (1.12.9.23)
(6)No limitation of adaptation values?
(6)else
Condition fulfilled
Update action (1.12.9.24 - 1.12.9.26)
No update of adaptation values - hold action (1.12.9.27)
Counter update and calculation of FAC_CYL_LAM_ABSV_SUM[i] (1.12.9.28 - 1.12.9.29)
(6)Adaptation sufficient for readiness?
(6)else
Yes
LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] = 1 (1.12.9.30)
(6)High- range adaptation not ready?
(6)else
LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] = 0
Assignment of low- range adaptation value into high- range adaptation variable (1.12.9.31)
Set the valid indication flag (1.12.9.32)
End of update action under nominal condition - low range


Update action under nominal condition - high range
(7)Fast adaptation requested?
(7)else
Yes
Update of fast adaptation under nominal condition - high range (1.12.9.34)
Update of nominal adaptation under nominal condition - high range (1.12.9.35)
Calculation of adaptation mean value (1.12.9.36)
(7)Mean value deviated from zero?
(7)else
Yes
Trim action on adaptation values (1.12.9.37)
(8)Lower or upper adaptation limit reached?
(8)else
Yes
Set the limitation flag for concerned exhaust bank (1.12.9.38)
Reset the limitation flag for concerned exhaust bank (1.12.9.39)
(7)No limitation of adaptation values?
(7)else
Condition fulfilled
Update action (1.12.9.40 - 1.12.9.42)
No update of adaptation values - hold action (1.12.9.43)
Counter update and calculation of FAC_CYL_LAM_ABSV_SUM[i] (1.12.9.44 - 1.12.9.45)
(7)Adaptation sufficient for readiness?
(7)else
Yes
LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] = 1 (1.12.9.46)
(7)Low- range adaptation not ready?
(7)else
LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] = 0
Assignment of high- range adaptation value into low- range adaptation variable (1.12.9.47)
Set the valid indication flag (1.12.9.48)
End of update action under nominal condition - high range


Continued from previous page
- Adaptation in the additive range -
(3)Is CILC - adaptation in ADJ_NOM_OFS state?
(3)else
Yes
(4)Timer unequal zero?
(4)else
Yes
Decrement the timer (1.12.9.49)
(4)Timer equal zero?
(4)else
State check see 1.12.9.62
Yes
(5)Fast adaptation requested?
(5)else
Yes
Update of fast adaptation in offset area (1.12.9.50)
Update of nominal adaptation in offset area (1.12.9.51)
Calculation of adaptation mean value (1.12.9.52)
(5)Mean value deviated from zero?
(5)else
Yes
Trim action on adaptation values (1.12.9.53)
(6)Lower or upper adaptation limit reached?
(6)else
Yes
Set the limitation flag for concerned exhaust bank (1.12.9.54)
Reset the limitation flag for concerned exhaust bank (1.12.9.55)
Continued on next page


Continued from previous page
(5)No limitation of adaptation values?
(5)else
Condition fulfilled
Update action (1.12.9.56 - 1.12.9.58)
No update of adaptation values - hold action (1.12.9.59)
Counter update and set the update flag (1.12.9.60 - 1.12.9.61)
(1)Is CILC - adaptation in passive- or wait- state?
(1)else
Yes
Reset the timer (1.12.9.62)
End of CILC - adaptation - update

```
if$^{(1)}$ STATE_LAM_CYL_SEL_ADJ[i]$_{k}$ ≠ STATE_LAM_CYL_SEL_ADJ[i]$_{k-1}$
then
    if$^{(2)}$ STATE_LAM_CYL_SEL_ADJ[i] = 2 [ADJ_ENG_COLD]
    then
        CTR_T_LAM_CYL_ADJ[i] = C_T_LAM_CYL_ADJ_CST                      (4.1.13.9.1)
    else$^{(2)}$
        if$^{(3)}$ (STATE_LAM_CYL_SEL_ADJ[i] = 4 [ADJ_NOM_L_RNG] or
                   STATE_LAM_CYL_SEL_ADJ[i] = 5 [ADJ_NOM_H_RNG] or
                   STATE_LAM_CYL_SEL_ADJ[i] = 6 [ADJ_NOM_OFS])
        then
            CTR_T_LAM_CYL_ADJ[i] = C_T_LAM_CYL_ADJ                      (4.1.13.9.2)
        endif$^{(3)}$
    endif$^{(2)}$
    LV_LAM_CYL_SEL_ADJ_VLD[i] = 0                                       (4.1.13.9.3)
endif$^{(1)}$

if$^{(1)}$ LC_ADAPT_COR_OUT = 1
then
    if$^{(2)}$ STATE_LAM_CYL_SEL[i] = 3 [CLL]
    then
        if$^{(3)}$ STATE_LAM_CYL_SEL_ADJ[i] = 2 [ADJ_ENG_COLD]
        then
            if$^{(4)}$ CTR_T_LAM_CYL_ADJ[i] ≠ 0
            then
                decrement CTR_T_LAM_CYL_ADJ[i]                          (4.1.13.9.4)
            endif$^{(4)}$
            if$^{(4)}$ CTR_T_LAM_CYL_ADJ[i] = 0
            then
                if$^{(5)}$ LV_LAM_CYL_SEL_CTL_FAST_REQ = 1
                then
                    for$^{(6)}$ j(x) = 1 : Number of cylinders per exhaust bank
                        FAC_LAM_CYL_SEL_ADJ_CST[x]$_{k}$ =
                            FAC_LAM_CYL_SEL_ADJ_CST[x]$_{k-1}$ +
                            IP_FAC_OPP_LAM_CYL_SEL_ADJ_FAST*
                            FAC_LAM_CYL_SEL_ADJ_CST_FIL[x]              (4.1.13.9.5)
                    endfor$^{(6)}$
                else$^{(5)}$
                    for$^{(6)}$ j(x) = 1 : Number of cylinders per exhaust bank
                        FAC_LAM_CYL_SEL_ADJ_CST[x]$_{k}$ =
```

```
                $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_{k-1}$ +
                IP_FAC_OPP_LAM_CYL_SEL_ADJ*
                FAC_LAM_CYL_SEL_ADJ_CST_FIL[x] (4.1.13.9.6)
        endfor$^{(6)}$
endif$^{(5)}$
FAC_LAM_CYL_SEL_ADJ_CST_MV[i] =
        $\frac{1}{\alpha} * \sum_{x=1}^{\alpha} FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]$

        $\alpha$: number of cylinders per bank (4.1.13.9.7)

if$^{(5)}$   FAC_LAM_CYL_SEL_ADJ_CST_MV[i] $\neq 0$
then
        for$^{(6)}$   j(x) = 1 : Number of cylinders per exhaust bank
                $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_k$ =
                        $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_k$ –
                        FAC_LAM_CYL_SEL_ADJ_CST_MV[i] (4.1.13.9.8)
        endfor$^{(6)}$
endif$^{(5)}$

for$^{(5)}$   j(x) = 1 : Number of cylinders per exhaust bank
        if$^{(6)}$   ($FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_k$ >
                        C_FAC_LAM_CYL_SEL_ADJ_CST_MAX or
                        $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_k$ <
                        C_FAC_LAM_CYL_SEL_ADJ_CST_MIN)
        then
                LV_FAC_LAM_CYL_ADJ_CST_LIM[i] = 1
                        (4.1.13.9.9)
                break$^{(5)}$
        else$^{(6)}$
                LV_FAC_LAM_CYL_ADJ_CST_LIM[i] = 0
                        (4.1.13.9.10)
        endif$^{(6)}$
endfor$^{(5)}$

if$^{(5)}$   LV_FAC_LAM_CYL_ADJ_CST_LIM[i] = 0
then
        for$^{(6)}$   j(x) = 1 : Number of cylinders per exhaust bank
                $FAC\_CYL\_LAM[x]_k$ = FAC_CYL_LAM[x] –
                        ($FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_k$ –
                        $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_{k-1}$) (4.1.13.9.11)
```

```
                $FAC\_CYL\_LAM\_INT[x]_k = FAC\_CYL\_LAM[x]_k$
                                                                  (4.1.13.9.12)
                $FAC\_LAM\_CYL\_SEL\_ADJ\_CST\_FIL[x]_k =$
                      $FAC\_LAM\_CYL\_SEL\_ADJ\_CST\_FIL[x] -$
                        $(FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_k -$
                $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_{k-1})$           (4.1.13.9.13)
            endfor$^{(6)}$
        else$^{(5)}$
            for$^{(6)}$  j(x) = 1 : Number of cylinders per exhaust bank
                $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_k =$
                  $FAC\_LAM\_CYL\_SEL\_ADJ\_CST[x]_{k-1}$           (4.1.13.9.14)
            endfor$^{(6)}$
        endif$^{(5)}$
        $CTR\_T\_LAM\_CYL\_ADJ[i] = C\_T\_LAM\_CYL\_ADJ\_CST$
                                                                  (4.1.13.9.15)
        $LV\_LAM\_CYL\_SEL\_ADJ\_VLD[i] = 1$
                                                                  (4.1.13.9.16)
    endif$^{(4)}$
else$^{(3)}$
    if$^{(4)}$  $STATE\_LAM\_CYL\_SEL\_ADJ[i] = 4$ [ADJ_NOM_L_RNG]
    then
        if$^{(5)}$  $CTR\_T\_LAM\_CYL\_ADJ[i] \neq 0$
        then
            decrement    CTR_T_LAM_CYL_ADJ[i]
                                                                  (4.1.13.9.17)
        endif$^{(5)}$
        if$^{(5)}$  $CTR\_T\_LAM\_CYL\_ADJ[i] = 0$
        then
            if$^{(6)}$  $LV\_LAM\_CYL\_SEL\_CTL\_FAST\_REQ = 1$
            then
                for$^{(7)}$  j(x) = 1 : Number of cylinders per exhaust bank
                    $FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_k =$
                      $FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_{k-1} +$
                     $IP\_FAC\_OPP\_LAM\_CYL\_SEL\_ADJ\_FAST*$
                  $FAC\_LAM\_CYL\_SEL\_ADJ\_FIL[x]_k$               (4.1.13.9.18)
                endfor$^{(7)}$
            else$^{(6)}$
                for$^{(7)}$  j(x) = 1 : Number of cylinders per exhaust bank
```

```
                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_k$ =
                    FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_{k-1}$ +
                    IP_FAC_OPP_LAM_CYL_SEL_ADJ*
                    FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$ (4.1.13.9.19)
        endfor(7)
endif(6)
FAC_LAM_CYL_SEL_ADJ_L_RNG_MV[i] =
        $\frac{1}{\alpha} * \sum_{x=1}^{\alpha} FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]$

        $\alpha$: number of cylinders per bank (4.1.13.9.20)

if(6)   FAC_LAM_CYL_SEL_ADJ_L_RNG_MV[i] $\neq 0$
then
        for(7)  j(x) = 1 : Number of cylinders per exhaust bank
                FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_k$ =
                    FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_k$ –
                    FAC_LAM_CYL_SEL_ADJ_L_RNG_MV[i]
                    (4.1.13.9.21)
        endfor(7)
endif(6)

for(6)  j(x) = 1 : Number of cylinders per exhaust bank
        if(7)   (FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_k$ >
                    C_FAC_LAM_CYL_SEL_ADJ_MAX or
                    FAC_LAM_CYL_SEL_ADJ_L_RNG[x]$_k$ <
                    C_FAC_LAM_CYL_SEL_ADJ_MIN)
        then
                LV_FAC_LAM_CYL_SEL_ADJ_LIM[i] = 1
                    (4.1.13.9.22)
                break(6)
        else(7)
                LV_FAC_LAM_CYL_SEL_ADJ_LIM[i] = 0
                    (4.1.13.9.23)
        endif(7)
endfor(6)

if(6)   LV_FAC_LAM_CYL_SEL_ADJ_LIM[i] = 0
then
        for(7)  j(x) = 1 : Number of cylinders per exhaust
```

```
                bank
                $FAC\_CYL\_LAM[x]_k = FAC\_CYL\_LAM[x] - (FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_k - FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_{k-1})$
                (4.1.13.9.24)
                $FAC\_CYL\_LAM\_INT[x]_k = FAC\_CYL\_LAM[x]_k$ (4.1.13.9.25)
                $FAC\_LAM\_CYL\_SEL\_ADJ\_FIL[x]_k = FAC\_LAM\_CYL\_SEL\_ADJ\_FIL[x] - (FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_k - FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_{k-1})$
                (4.1.13.9.26)
            endfor(7)
        else(6)
            for(7) j(x) = 1 : Number of cylinders per exhaust bank
                $FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_k = FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[x]_{k-1}$
                (4.1.13.9.27)
            endfor(7)
        endif(6)
        CTR_T_LAM_CYL_ADJ[i] = C_T_LAM_CYL_ADJ
        (4.1.13.9.28)
        $FAC\_CYL\_LAM\_ABSV\_SUM[i] = \sum_{x=1}^{\alpha} |FAC\_CYL\_LAM[x]|$
        $\alpha$ :nr. of cylinders per bank
        (4.1.13.9.29)
        if(6) FAC_CYL_LAM_ABSV_SUM[i] ≤ C_FAC_LAM_CYL_SEL_ADJ_VLD_THD
        then
            LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] = 1
            (4.1.13.9.30)
        endif(6)
        if(6) LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] = 0
        then
            FAC_LAM_CYL_SEL_ADJ_H_RNG[x] = FAC_LAM_CYL_SEL_ADJ_L_RNG[x] (4.1.13.9.31)
        endif(6)
        LV_LAM_CYL_SEL_ADJ_VLD[i] = 1
        (4.1.13.9.32)
    endif(5)
else(4)
    if(5) STATE_LAM_CYL_SEL_ADJ[i] = 5 [ADJ_NOM_H_RNG]
    then
```

```
if(6)   CTR_T_LAM_CYL_ADJ[i] $\neq$ 0
then
        decrement   CTR_T_LAM_CYL_ADJ[i]
                                                    (4.1.13.9.33)
endif(6)
if(6)   CTR_T_LAM_CYL_ADJ[i] = 0
then
        if(7)   LV_LAM_CYL_SEL_CTL_FAST_REQ = 1
        then
                for(8)   j(x) = 1 : Number of cylinders per
                                                    exhaust bank
                FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_k$ =
                   FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{k-1}$ +
                  IP_FAC_OPP_LAM_CYL_SEL_ADJ_FAST*
                              FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$
                                                    (4.1.13.9.34)
                endfor(8)
        else(7)
                for(8)   j(x) = 1 : Number of cylinders per
                                                    exhaust bank
                FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_k$ =
                   FAC_LAM_CYL_SEL_ADJ_H_RNG[x]$_{k-1}$ +
                           IP_FAC_OPP_LAM_CYL_SEL_ADJ*
                              FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$
                                                    (4.1.13.9.35)
                endfor(8)
        endif(7)
        FAC_LAM_CYL_SEL_ADJ_H_RNG_MV[i] =
                $\frac{1}{\alpha} * \sum_{x=1}^{\alpha}$ FAC_LAM_CYL_SEL_ADJ_H_RNG[x]
                $\alpha$: number of cylinders per bank (4.1.13.9.36)

        if(7)  FAC_LAM_CYL_SEL_ADJ_H_RNG_MV[i] $\neq$0
        then
                for(8)   j(x) = 1 : Number of cylinders per
                                                    exhaust bank
```

```
                $\text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_k = \text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_k - \text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG\_MV}[i]$
                (4.1.13.9.37)
        endfor$^{(8)}$
endif$^{(7)}$
for$^{(7)}$ j(x) = 1 : Number of cylinders per exhaust bank
    if$^{(8)}$ ($\text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_k > \text{C\_FAC\_LAM\_CYL\_SEL\_ADJ\_MAX}$ or $\text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_k < \text{C\_FAC\_LAM\_CYL\_SEL\_ADJ\_MIN}$)
    then
        $\text{LV\_FAC\_LAM\_CYL\_SEL\_ADJ\_LIM}[i] = 1$
        (4.1.13.9.38)
        break$^{(7)}$
    else$^{(8)}$
        $\text{LV\_FAC\_LAM\_CYL\_SEL\_ADJ\_LIM}[i] = 0$
        (4.1.13.9.39)
    endif$^{(8)}$
endfor$^{(7)}$

if$^{(7)}$ $\text{LV\_FAC\_LAM\_CYL\_SEL\_ADJ\_LIM}[i] = 0$
then
        for$^{(8)}$ j(x) = 1 : Number of cylinders per exhaust bank
            $\text{FAC\_CYL\_LAM}[x]_k = \text{FAC\_CYL\_LAM}[x] - (\text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_k - \text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_{k-1})$
            (4.1.13.9.40)
            $\text{FAC\_CYL\_LAM\_INT}[x]_k = \text{FAC\_CYL\_LAM}[x]_k$ (4.1.13.9.41)
            $\text{FAC\_LAM\_CYL\_SEL\_ADJ\_FIL}[x]_k = \text{FAC\_LAM\_CYL\_SEL\_ADJ\_FIL}[x] - (\text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_k - \text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_{k-1})$
            (4.1.13.9.42)
        endfor$^{(8)}$
else$^{(7)}$
        for$^{(8)}$ j(x) = 1 : Number of cylinders per exhaust bank
            $\text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_k = \text{FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG}[x]_{k-1}$
            (4.1.13.9.43)
```

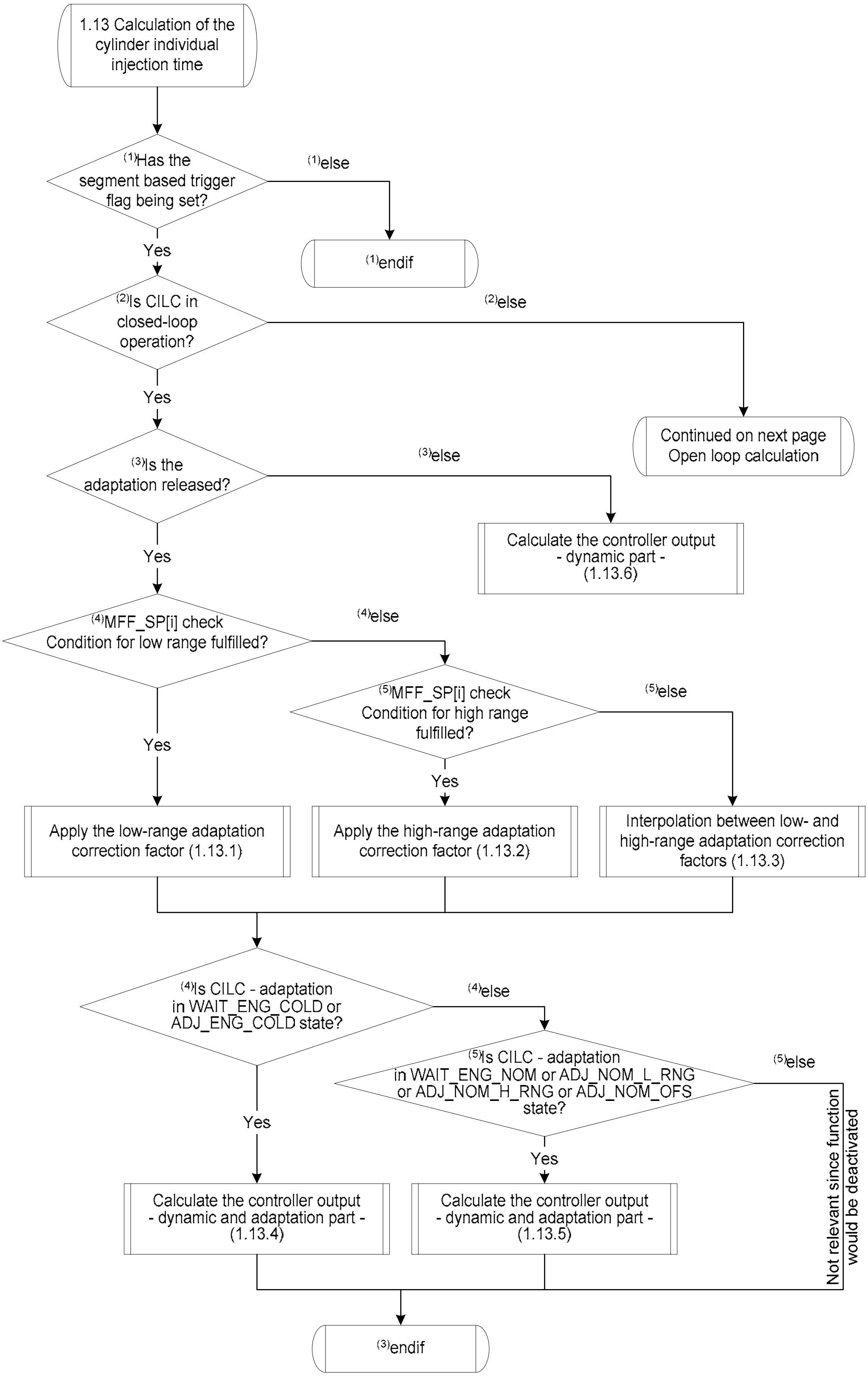
```
                        endfor$^{(8)}$
                    endif$^{(7)}$
                    CTR_T_LAM_CYL_ADJ[i] = C_T_LAM_CYL_ADJ
                                                                  (4.1.13.9.44)
                    $FAC\_CYL\_LAM\_ABSV\_SUM[i] = \sum_{x=1}^{\alpha} |FAC\_CYL\_LAM[x]|$
                    $\alpha$ : nr. of cylinders per bank

                                                                  (4.1.13.9.45)
                    if$^{(7)}$    FAC_CYL_LAM_ABSV_SUM[i] $\leq$
                                 C_FAC_LAM_CYL_SEL_ADJ_VLD_THD
                    then
                        LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[i] = 1
                                                                  (4.1.13.9.46)
                    endif$^{(7)}$
                    if$^{(7)}$ LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[i] = 0
                    then
                        FAC_LAM_CYL_SEL_ADJ_L_RNG[x] =
                    FAC_LAM_CYL_SEL_ADJ_H_RNG[x]                  (4.1.13.9.47)
                    endif$^{(7)}$
                    LV_LAM_CYL_SEL_ADJ_VLD[i] = 1
                                                                  (4.1.13.9.48)
                endif$^{(6)}$
            endif$^{(5)}$
        endif$^{(4)}$
    endif$^{(3)}$
    if$^{(3)}$    STATE_LAM_CYL_SEL_ADJ[i] = 6 [ADJ_NOM_OFS]
    then
        if$^{(4)}$    CTR_T_LAM_CYL_ADJ[i] $\neq$ 0
        then
            decrement    CTR_T_LAM_CYL_ADJ[i]
                                                                  (4.1.13.9.49)
        endif$^{(4)}$
        if$^{(4)}$    CTR_T_LAM_CYL_ADJ[i] = 0
        then
            if$^{(5)}$    LV_LAM_CYL_SEL_CTL_FAST_REQ = 1
            then
                for$^{(6)}$    j(x) = 1 : Number of cylinders per exhaust bank
                    $MFF\_ADD\_CYL\_LAM\_COR[x]_k$ =
```

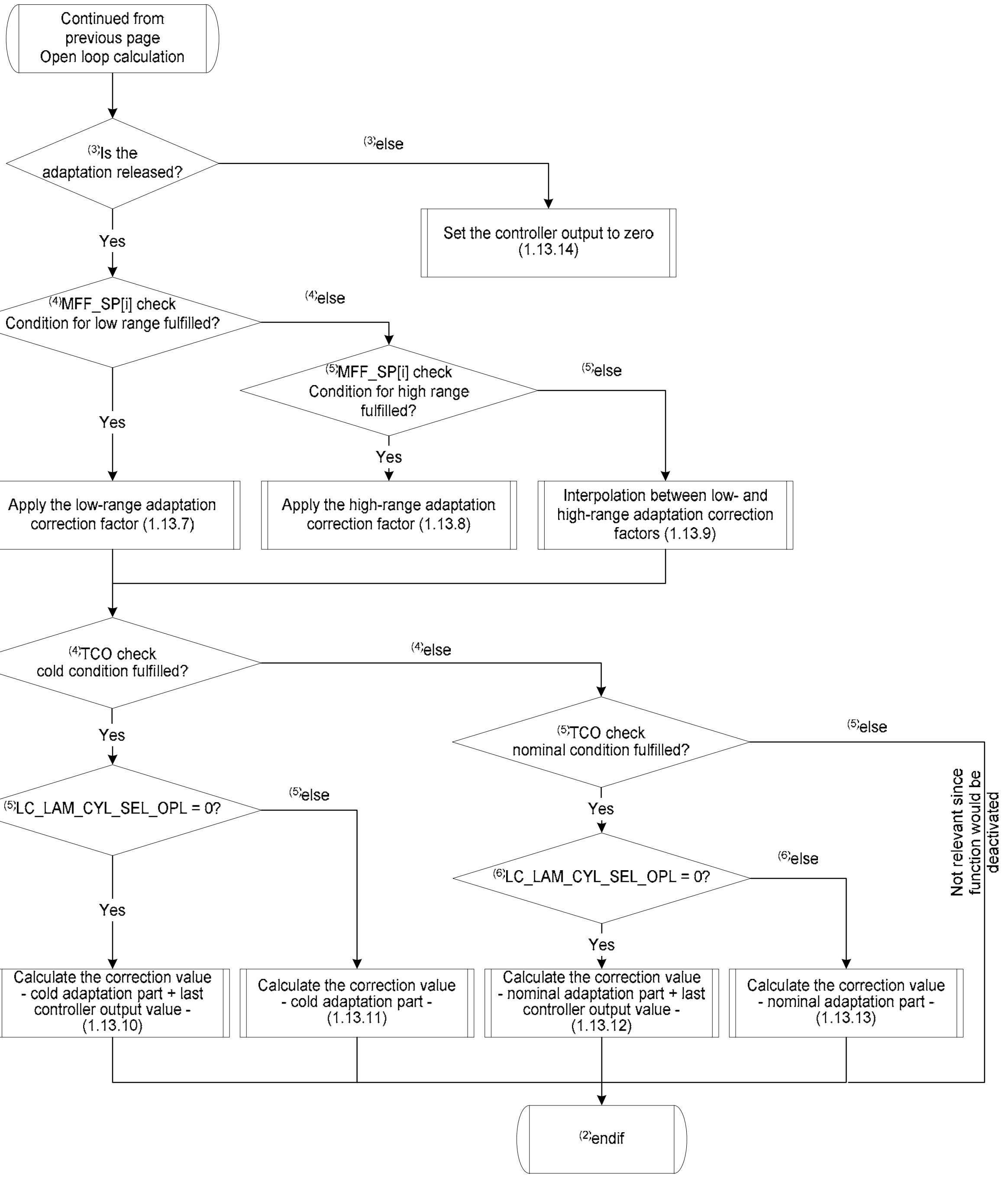
```
                        MFF_ADD_CYL_LAM_COR[x]$_{k-1}$ + MFF_SP[i]*
                        IP_FAC_OPP_LAM_CYL_SEL_ADJ_FAST*
                        FAC_LAM_CYL_SEL_ADJ_FIL[x]$_{k}$ (4.1.13.9.50)
    endfor$^{(6)}$
else$^{(5)}$
    for$^{(6)}$ j(x) = 1 : Number of cylinders per exhaust bank
            MFF_ADD_CYL_LAM_COR[x]$_{k}$ =
                        MFF_ADD_CYL_LAM_COR[x]$_{k-1}$ + MFF_SP[i]*
                        IP_FAC_OPP_LAM_CYL_SEL_ADJ*
                        FAC_LAM_CYL_SEL_ADJ_FIL[x]$_{k}$ (4.1.13.9.51)
    endfor$^{(6)}$
endif$^{(5)}$
MFF_ADD_CYL_LAM_COR_MV[i] =
                        $\frac{1}{\alpha} * \sum_{x=1}^{\alpha}$ MFF_ADD_CYL_LAM_COR[x]
                        $\alpha$: number of cylinders per bank (4.1.13.9.52)

if$^{(5)}$ MFF_ADD_CYL_LAM_COR_MV[i] $\neq 0$
then
    for$^{(6)}$ j(x) = 1 : Number of cylinders per exhaust bank
            MFF_ADD_CYL_LAM_COR[x]$_{k}$ =
                        MFF_ADD_CYL_LAM_COR[x]$_{k}$ -
                        MFF_ADD_CYL_LAM_COR_MV[i] (4.1.13.9.53)
    endfor$^{(6)}$
endif$^{(5)}$

for$^{(5)}$ j(x) = 1 : Number of cylinders per exhaust bank
    if$^{(6)}$ (MFF_ADD_CYL_LAM_COR[x]$_{k}$ >
                        C_MFF_MAX_ADD_LAM_CYL_SEL_ADJ or
                        MFF_ADD_CYL_LAM_COR[x]$_{k}$ <
                        C_MFF_MIN_ADD_LAM_CYL_SEL_ADJ)
    then
            LV_MFF_ADD_CYL_LAM_COR_LIM[i] = 1
                        (4.1.13.9.54)
            break$^{(5)}$
    else$^{(6)}$
            LV_MFF_ADD_CYL_LAM_COR_LIM[i] = 0
                        (4.1.13.9.55)
    endif$^{(6)}$
```

```
                endfor$^{(5)}$

                if$^{(5)}$ LV_MFF_ADD_CYL_LAM_COR_LIM[i] = 0
                then
                    for$^{(6)}$ j(x) = 1 : Number of cylinders per exhaust bank
                        FAC_CYL_LAM[x]$_k$ = FAC_CYL_LAM[x]$_k$ –
                                (MFF_ADD_CYL_LAM_COR[x]$_k$ -
                                MFF_ADD_CYL_LAM_COR[x]$_{k-1}$)/
                                MFF_SP[i]  (4.1.13.9.56)
                        FAC_CYL_LAM_INT[x]$_k$ = FAC_CYL_LAM[x]$_k$
                                (4.1.13.9.57)
                        FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$ =
                                FAC_LAM_CYL_SEL_ADJ_FIL[x]$_k$ -
                                (MFF_ADD_CYL_LAM_COR[x]$_k$ -
                                MFF_ADD_CYL_LAM_COR[x]$_{k-1}$)/
                                MFF_SP[i]  (4.1.13.9.58)
                    endfor$^{(6)}$
                else$^{(5)}$
                    for$^{(6)}$ j(x) = 1 : Number of cylinders per exhaust bank
                        MFF_ADD_CYL_LAM_COR[x]$_k$ =
                                MFF_ADD_CYL_LAM_COR[x]$_{k-1}$  (4.1.13.9.59)
                    endfor$^{(6)}$
                endif$^{(5)}$

                CTR_T_LAM_CYL_ADJ[i] = C_T_LAM_CYL_ADJ
                                (4.1.13.9.60)
                LV_LAM_CYL_SEL_ADJ_VLD[i] = 1
                                (4.1.13.9.61)
            endif$^{(4)}$

        endif$^{(3)}$
    endif$^{(2)}$
endif$^{(1)}$


if$^{(1)}$ (STATE_LAM_CYL_SEL_ADJ[i] = 0 [PAS] or
            STATE_LAM_CYL_SEL_ADJ[i] = 1 [WAIT_ENG_COLD] or
            STATE_LAM_CYL_SEL_ADJ[i] = 3 [WAIT_ENG_NOM])
then
    CTR_T_LAM_CYL_ADJ[i] = 0
                                (4.1.13.9.62)
endif$^{(1)}$
```

4.1.14 Calculation of the cylinder individual injection time correction

The value of the controller output FAC_CYL_LAM_COR[x] shall be calculated depending on adaptation state STATE_LAM_CYL_SEL_ADJ[i].


1.13 Calculation of the cylinder individual injection time
(1)Has the segment based trigger flag being set?
(1)else
(1)endif
Yes
(2)Is CILC in closed-loop operation?
(2)else
Continued on next page Open loop calculation
Yes
(3)Is the adaptation released?
(3)else
Calculate the controller output - dynamic part - (1.13.6)
Yes
(4)MFF_SP[i] check Condition for low range fulfilled?
(4)else
(5)MFF_SP[i] check Condition for high range fulfilled?
(5)else
Yes
Yes
Apply the low-range adaptation correction factor (1.13.1)
Apply the high-range adaptation correction factor (1.13.2)
Interpolation between low- and high-range adaptation correction factors (1.13.3)
(4)Is CILC - adaptation in WAIT_ENG_COLD or ADJ_ENG_COLD state?
(4)else
(5)Is CILC - adaptation in WAIT_ENG_NOM or ADJ_NOM_L_RNG or ADJ_NOM_H_RNG or ADJ_NOM_OFS state?
(5)else
Yes
Yes
Not relevant since function would be deactivated
Calculate the controller output - dynamic and adaptation part - (1.13.4)
Calculate the controller output - dynamic and adaptation part - (1.13.5)
(3)endif


Continued from previous page Open loop calculation
(3)Is the adaptation released?
(3)else
Set the controller output to zero (1.13.14)
Yes
(4)MFF_SP[i] check Condition for low range fulfilled?
(4)else
(5)MFF_SP[i] check Condition for high range fulfilled?
(5)else
Yes
Yes
Apply the low-range adaptation correction factor (1.13.7)
Apply the high-range adaptation correction factor (1.13.8)
Interpolation between low- and high-range adaptation correction factors (1.13.9)
(4)TCO check cold condition fulfilled?
(4)else
Yes
(5)TCO check nominal condition fulfilled?
(5)else
(5)LC_LAM_CYL_SEL_OPL = 0?
(5)else
Yes
(6)LC_LAM_CYL_SEL_OPL = 0?
(6)else
Not relevant since function would be deactivated
Yes
Yes
Calculate the correction value - cold adaptation part + last controller output value - (1.13.10)
Calculate the correction value - cold adaptation part - (1.13.11)
Calculate the correction value - nominal adaptation part + last controller output value - (1.13.12)
Calculate the correction value - nominal adaptation part - (1.13.13)
(2)endif

```
if$^{(1)}$ LV_TRIG[x] =1
then
    if$^{(2)}$ STATE_LAM_CYL_SEL[i] = 3 [CLL]
    then
        if$^{(3)}$ LC_ADAPT_COR_OUT = 1
        then
            if$^{(4)}$ MFF_SP[i] ≤ C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG
            then
                FAC_LAM_CYL_SEL_ADJ[x] =
                        FAC_LAM_CYL_SEL_ADJ_L_RNG[x] (4.1.14.1)
            else$^{(4)}$
                if$^{(5)}$ MFF_SP[i] ≥ C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG
                then
                    FAC_LAM_CYL_SEL_ADJ[x] =
                            FAC_LAM_CYL_SEL_ADJ_H_RNG[x] (4.1.14.2)
                else$^{(5)}$
                    FAC_LAM_CYL_SEL_ADJ[x] =
                            FAC_LAM_CYL_SEL_ADJ_L_RNG[x] +
                            (MFF_SP[i] - C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG)*
                            (FAC_LAM_CYL_SEL_ADJ_H_RNG[x] –
                            FAC_LAM_CYL_SEL_ADJ_L_RNG[x])/
                            (C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG –
                            C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG) (4.1.14.3)
                endif$^{(5)}$
            endif$^{(4)}$
            if$^{(4)}$ (STATE_LAM_CYL_SEL_ADJ[i] = 1 [WAIT_ENG_COLD] or
                    STATE_LAM_CYL_SEL_ADJ[i] = 2 [ADJ_ENG_COLD])
            then
                FAC_CYL_LAM_COR[x]$_k$ = FAC_LAM_CYL_SEL_ADJ_CST[x]
                        + FAC_CYL_LAM[x]$_k$ (4.1.14.4)
            else$^{(4)}$
                if$^{(5)}$ (STATE_LAM_CYL_SEL_ADJ[i] = 3 [WAIT_ENG_NOM]
                    or STATE_LAM_CYL_SEL_ADJ[i] = 4 [ADJ_NOM_L_RNG]
                    or STATE_LAM_CYL_SEL_ADJ[i] = 5 [ADJ_NOM_H_RNG]
                    or STATE_LAM_CYL_SEL_ADJ[i] = 6 [ADJ_NOM_OFS])
                then
                    FAC_CYL_LAM_COR[x]$_k$ = FAC_LAM_CYL_SEL_ADJ[x]
                            + FAC_CYL_LAM[x]$_k$ (4.1.14.5)
                endif$^{(5)}$
            endif$^{(4)}$
        else$^{(3)}$
            FAC_CYL_LAM_COR[x]$_k$ = FAC_CYL_LAM[x]$_k$
                                                        (4.1.14.6)
        endif$^{(3)}$
    else$^{(2)}$
        if$^{(3)}$ LC_ADAPT_COR_OUT = 1
        then
```

```
        if$^{(4)}$  MFF_SP[i] ≤ C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG
        then
                FAC_LAM_CYL_SEL_ADJ[x] =
                        FAC_LAM_CYL_SEL_ADJ_L_RNG[x] (4.1.14.7)
        else$^{(4)}$
                if$^{(5)}$  MFF_SP[i] ≥ C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG
                then
                        FAC_LAM_CYL_SEL_ADJ[x] =
                                FAC_LAM_CYL_SEL_ADJ_H_RNG[x] (4.1.14.8)
                else$^{(5)}$
                        FAC_LAM_CYL_SEL_ADJ[x] =
                                FAC_LAM_CYL_SEL_ADJ_L_RNG[x] +
                                (MFF_SP[i] - C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG)*
                                (FAC_LAM_CYL_SEL_ADJ_H_RNG[x] –
                                FAC_LAM_CYL_SEL_ADJ_L_RNG[x])/
                                (C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG –
                                C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG) (4.1.14.9)
                endif$^{(5)}$
        endif$^{(4)}$
        if$^{(4)}$  (C_TEMP_LAM_CYL_CST_MIN < TEMP_LAM_CYL_SEL <
                        C_TEMP_LAM_CYL_CST_MAX)
        then
                if$^{(5)}$  LC_LAM_CYL_SEL_OPL = 0
                then
                        FAC_CYL_LAM_COR[x]$_{k}$ =
                                FAC_LAM_CYL_SEL_ADJ_CST[x] +
                                FAC_CYL_LAM[x]$_{k}$ (4.1.14.10)
                else$^{(5)}$
                        FAC_CYL_LAM_COR[x]$_{k}$ =
                                FAC_LAM_CYL_SEL_ADJ_CST[x] (4.1.14.11)
                endif$^{(5)}$
        else$^{(4)}$
                if$^{(5)}$  TEMP_LAM_CYL_SEL ≥ C_TEMP_LAM_CYL_MIN
                then
                        if$^{(6)}$  LC_LAM_CYL_SEL_OPL = 0
                        then
                                FAC_CYL_LAM_COR[x]$_{k}$ =
                                        FAC_LAM_CYL_SEL_ADJ[x]
                                        + FAC_CYL_LAM[x]$_{k}$ (4.1.14.12)
                        else$^{(6)}$
                                FAC_CYL_LAM_COR[x]$_{k}$ =
                                        FAC_LAM_CYL_SEL_ADJ[x] (4.1.14.13)
                        endif$^{(6)}$
                endif$^{(5)}$
        endif$^{(4)}$
else$^{(3)}$
        FAC_CYL_LAM_COR[x]$_{k}$ = 0
                                                        (4.1.14.14)
endif$^{(3)}$
```

```
    endif(2)
    [Remark: the cylinder based calculation (index x) shall be
    assigned to the corresponding exhaust bank (index i)]
endif(1)
```

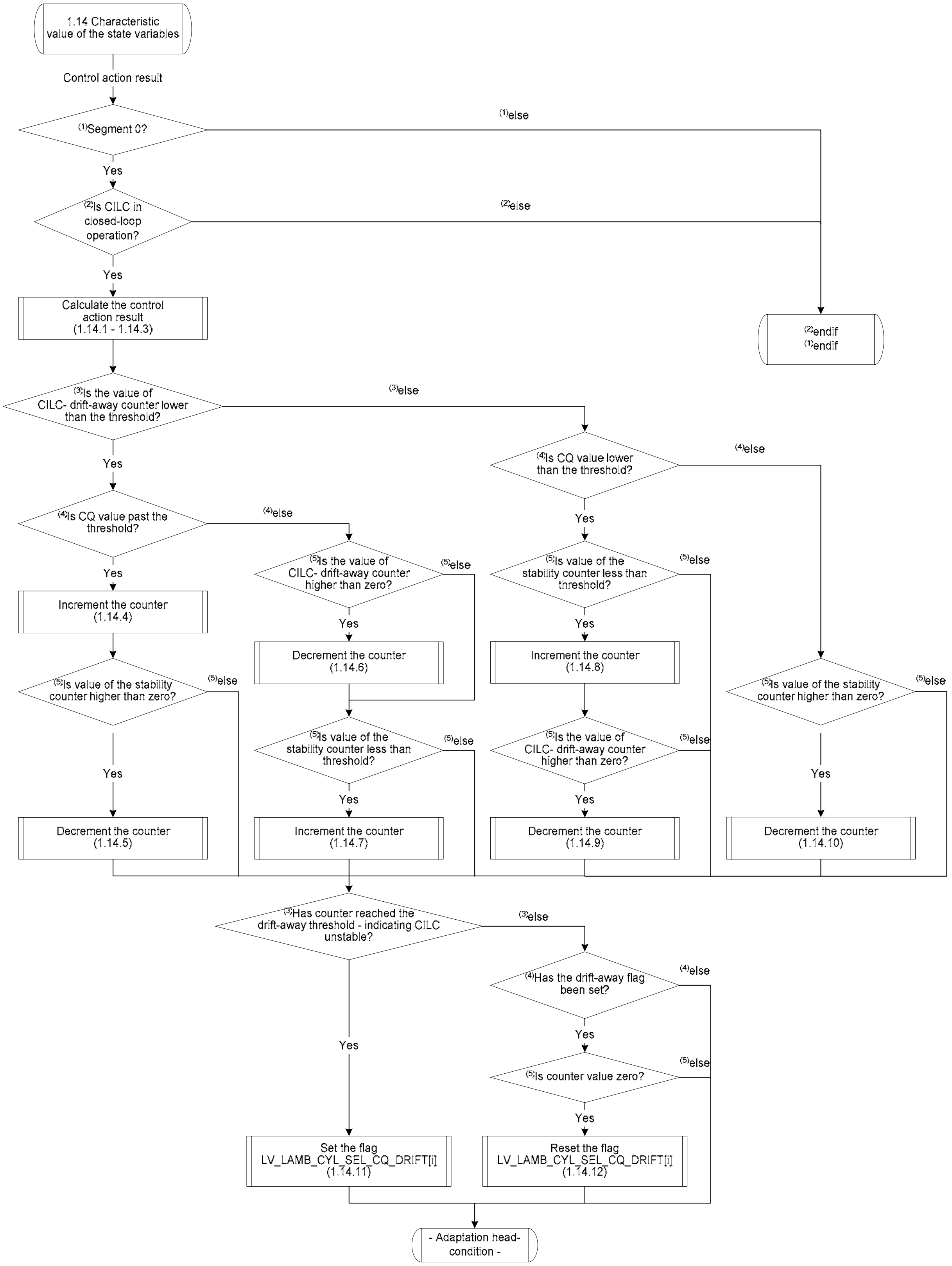
4.1.15 Calculation of the characteristic value of the state variables – control action result

In order to evaluate the control-action result the absolute value of mean value-deviation of the state variables shall be averaged over the whole combustion cycle.
In order to evaluate the adaptation result the sum of the absolute cylinder individual controller outputs is calculated (FAC_CYL_LAM_ABSV_SUM[i]).
Convergence of FAC_CYL_LAM_ABSV_SUM[i] is an indication of a well-running adaptation process.
In case of separate exhaust banks existing, the calculation shall be carried out for both exhaust banks at the beginning of the combustion cycle (SEG_NR = 0).

```
if(1) SEG_NR = 0
then
    if(2) STATE_LAM_CYL_SEL[i] = 3 [calculation is exhaust bank selective]
    then

        $\text{DELTA\_LAMB\_CYL\_SEL\_CQ[i]} = \frac{1}{\alpha}\sum_{x=1}^{\alpha}\left|\left[\frac{1}{\alpha}\sum_{x=1}^{\alpha}\text{DELTA\_LAMB\_CYL[x]}\right] - \text{DELTA\_LAMB\_CYL[x]}\right|$
        $\alpha$: nr. of cylinders per bank
                                                                        (4.1.15.1)

        $\text{FAC\_CYL\_LAM\_ABSV\_SUM[i]} = \sum_{x=1}^{\alpha}\left|\text{FAC\_CYL\_LAM[x]}\right|$
        $\alpha$: nr. of cylinders per bank
                                                                        (4.1.15.2)
        [Remark: the cylinder based variables (index x) shall be assigned to the
        corresponding exhaust bank (index i)]

        $\text{DELTA\_LAMB\_CYL\_SEL\_CQ\_FIL[i]}_k = \text{DELTA\_LAMB\_CYL\_SEL\_CQ\_FIL[i]}_{k-1}$
+
            $(\text{DELTA\_LAMB\_CYL\_SEL\_CQ[i]}_k - \text{DELTA\_LAMB\_CYL\_SEL\_CQ\_FIL[i]}_{k-1})$
                                   $*\text{C\_CRLC\_DELTA\_LAMB\_CYL\_CQ\_FIL}$ (4.1.15.3)

        if(3) CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] <
                                   C_CTR_LAMB_CYL_SEL_CQ_DRIFT_THD
        then
            if(4) DELTA_LAMB_CYL_SEL_CQ_FIL[i] ≥
                                   C_LAMB_CYL_SEL_CQ_DRIFT_THD
            then
                increment (CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i])
                                                                        (4.1.15.4)
                if(5) CTR_LAMB_CYL_SEL_CQ_STAB[i] > 0
                then
                    decrement (CTR_LAMB_CYL_SEL_CQ_STAB[i])
```

```
                                                            (4.1.15.5)
                endif$^{(5)}$
        else$^{(4)}$
                if$^{(5)}$    CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] > 0
                then
                        decrement (CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i])
                                                            (4.1.15.6)
                endif$^{(5)}$
                if$^{(5)}$    CTR_LAMB_CYL_SEL_CQ_STAB[i] <
                                     C_CTR_LAMB_CYL_SEL_CQ_STAB_THD
                then
                        increment (CTR_LAMB_CYL_SEL_CQ_STAB[i])
                                                            (4.1.15.7)
                endif$^{(5)}$
        endif$^{(4)}$
else$^{(3)}$
        if$^{(4)}$    DELTA_LAMB_CYL_SEL_CQ_FIL[i] <
                                     C_LAMB_CYL_SEL_CQ_DRIFT_THD
        then
                if$^{(5)}$    CTR_LAMB_CYL_SEL_CQ_STAB[i] <
                                     C_CTR_LAMB_CYL_SEL_CQ_STAB_THD
                then
                        increment (CTR_LAMB_CYL_SEL_CQ_STAB[i])
                                                            (4.1.15.8)
                endif$^{(5)}$
                if$^{(5)}$    CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] > 0
                then
                        decrement (CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i])
                                                            (4.1.15.9)
                endif$^{(5)}$
        else$^{(4)}$
                if$^{(5)}$    CTR_LAMB_CYL_SEL_CQ_STAB[i] > 0
                then
                        decrement (CTR_LAMB_CYL_SEL_CQ_STAB[i])
                                                            (4.1.15.10)
                endif$^{(5)}$
        endif$^{(4)}$
endif$^{(3)}$

if$^{(3)}$    CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] =
                                     C_CTR_LAMB_CYL_SEL_CQ_DRIFT_THD
then
        LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 1
                                                            (4.1.15.11)
else$^{(3)}$
        if$^{(4)}$    LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 1
```

```
        then
                if(5)    CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] = 0
                then
                         LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 0
                                                                      (4.1.15.12)
                endif(5)
        endif(4)
endif(3)
```

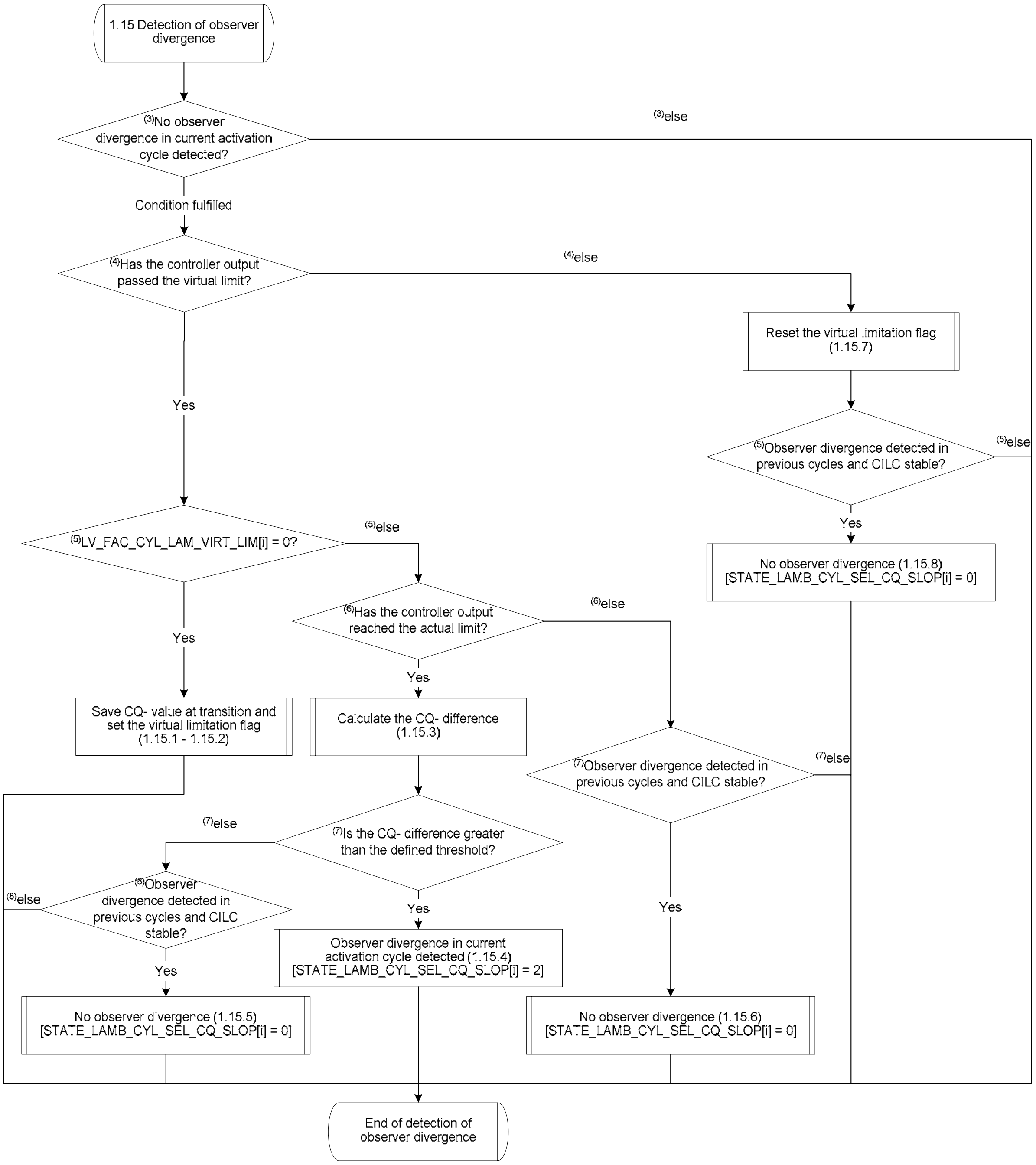

1.14 Characteristic value of the state variables
Control action result
(1)Segment 0?
(1)else
Yes
(2)Is CILC in closed-loop operation?
(2)else
Yes
Calculate the control action result (1.14.1 - 1.14.3)
(2)endif
(1)endif
(3)Is the value of CILC- drift-away counter lower than the threshold?
(3)else
Yes
(4)Is CQ value lower than the threshold?
(4)else
(4)Is CQ value past the threshold?
(4)else
Yes
Yes
Increment the counter (1.14.4)
(5)Is the value of CILC- drift-away counter higher than zero?
(5)else
(5)Is value of the stability counter less than threshold?
(5)else
Yes
Yes
Decrement the counter (1.14.6)
Increment the counter (1.14.8)
(5)Is value of the stability counter higher than zero?
(5)else
(5)Is value of the stability counter higher than zero?
(5)else
(5)Is value of the stability counter less than threshold?
(5)else
(5)Is the value of CILC- drift-away counter higher than zero?
(5)else
Yes
Yes
Yes
Yes
Decrement the counter (1.14.5)
Increment the counter (1.14.7)
Decrement the counter (1.14.9)
Decrement the counter (1.14.10)
(3)Has counter reached the drift-away threshold - indicating CILC unstable?
(3)else
(4)Has the drift-away flag been set?
(4)else
Yes
Yes
(5)Is counter value zero?
(5)else
Yes
Set the flag LV_LAMB_CYL_SEL_CQ_DRIFT[i] (1.14.11)
Reset the flag LV_LAMB_CYL_SEL_CQ_DRIFT[i] (1.14.12)
- Adaptation head-condition -

4.1.16 Detection of observer divergence

In order to detect the observer divergence in closed-loop operation and stop the adaptation process consequently, virtual limitation (C_FAC_CYL_LAM_VIRT_MIN/MAX) is defined. As soon as the single controller output (FAC_CYL_LAM[x]) passes the virtual limit over, the value of CILC- Characteristic Quantity (CQ- value as a measure for the observer behaviour) for the concerned exhaust bank is stored. Should one of the concerned exhaust-bank members (cylinders) reach the actual limit (LV_FAC_CYL_LAM_LIM_MIN/MAX[x] = 1) the current CQ- value has to be compared with the stored value at virtual limit transition. A positive ascent of characteristic value (CQ- value) in current activation cycle (STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 2) is interpreted as for unstable behaviour of CILC and shall involve inhibition of adaptation process.

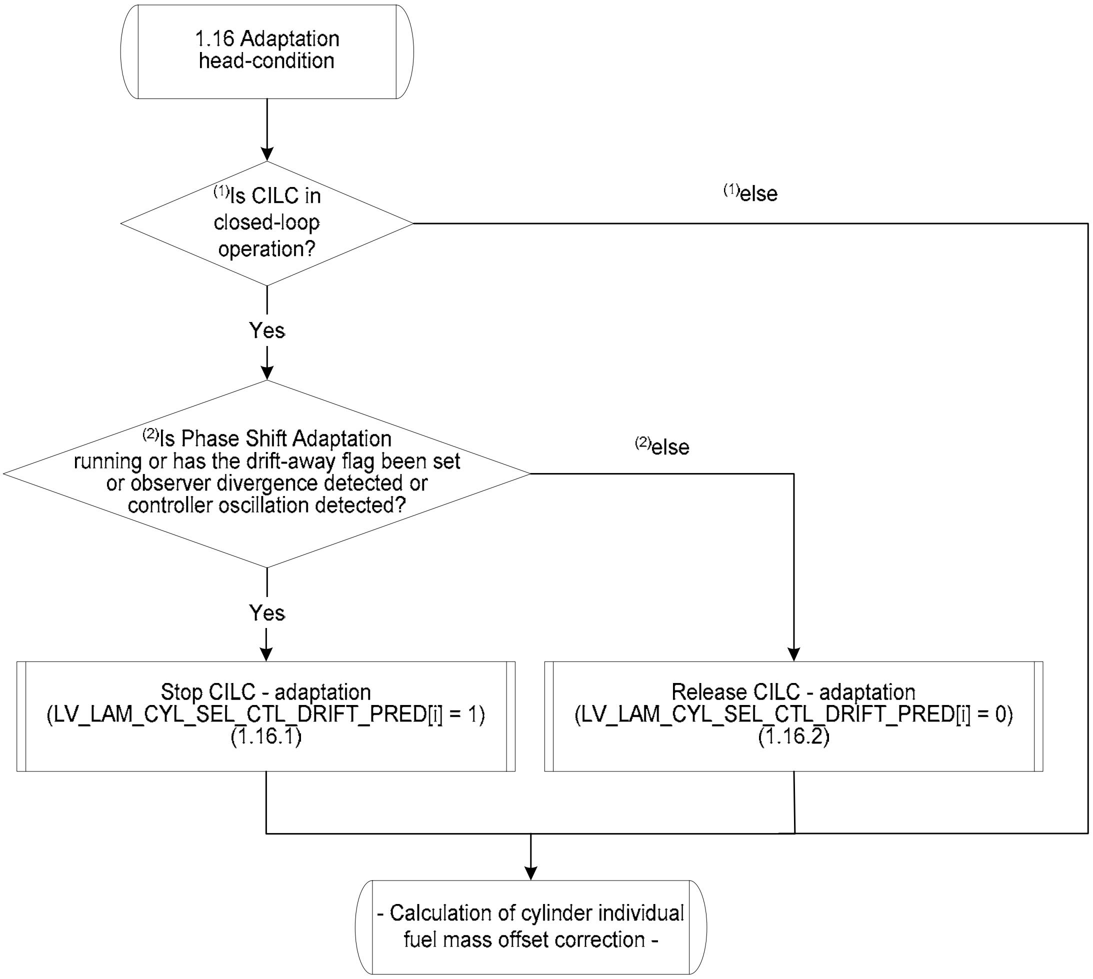
```
if(3) (STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0 or
                                  STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 1)
then
    if(4) (FAC_CYL_LAM[x]$_k$ < C_FAC_CYL_LAM_VIRT_MIN or
                          FAC_CYL_LAM[x]$_k$ > C_FAC_CYL_LAM_VIRT_MAX)
    then
        if(5) LV_FAC_CYL_LAM_VIRT_LIM[i] = 0
        then
            LAMB_CYL_SEL_CQ_REF[i] =
                          DELTA_LAMB_CYL_SEL_CQ_FIL[i]    (4.1.16.1)
            LV_FAC_CYL_LAM_VIRT_LIM[i] = 1
                                                          (4.1.16.2)
        else(5)
            if(6) (LV_FAC_CYL_LAM_LIM_MIN[x] = 1 or
                              LV_FAC_CYL_LAM_LIM_MAX[x] = 1)
            then
                LAMB_CYL_SEL_CQ_DIF_REF[i] =
                              DELTA_LAMB_CYL_SEL_CQ_FIL[i] –
                              LAMB_CYL_SEL_CQ_REF[i]      (4.1.16.3)
                if(7) LAMB_CYL_SEL_CQ_DIF_REF[i] >
                          C_LAMB_CYL_SEL_CQ_DIF_REF_THD
                then
                    STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 2
                                                          (4.1.16.4)
                else(7)
                 if(8) (STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 1
                        and CTR_LAMB_CYL_SEL_CQ_STAB[i] =
                        C_CTR_LAMB_CYL_SEL_CQ_STAB_THD)
```

```
                                then
                                    STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0
                                                                        (4.1.16.5)
                                endif(8)
                            endif(7)
                        else(6)
                            if(7)   (STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 1
                                     and CTR_LAMB_CYL_SEL_CQ_STAB[i] =
                                    C_CTR_LAMB_CYL_SEL_CQ_STAB_THD)
                            then
                                    STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0
                                                                        (4.1.16.6)
                            endif(7)
                        endif(6)
                    endif(5)
                else(4)
                    LV_FAC_CYL_LAM_VIRT_LIM[i] = 0
                                                                        (4.1.16.7)
                    if(5)   (STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 1 and
                                        CTR_LAMB_CYL_SEL_CQ_STAB[i] =
                                    C_CTR_LAMB_CYL_SEL_CQ_STAB_THD)
                    then
                        STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0
                                                                        (4.1.16.8)
                    endif(5)
                endif(4)
            endif(3)
        endif(2)
    endif(1)
```


1.15 Detection of observer divergence
(3)No observer divergence in current activation cycle detected?
(3)else
Condition fulfilled
(4)Has the controller output passed the virtual limit?
(4)else
Reset the virtual limitation flag (1.15.7)
Yes
(5)Observer divergence detected in previous cycles and CILC stable?
(5)else
Yes
No observer divergence (1.15.8) [STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0]
(5)LV_FAC_CYL_LAM_VIRT_LIM[i] = 0?
(5)else
(6)Has the controller output reached the actual limit?
(6)else
Yes
Yes
Save CQ- value at transition and set the virtual limitation flag (1.15.1 - 1.15.2)
Calculate the CQ- difference (1.15.3)
(7)Observer divergence detected in previous cycles and CILC stable?
(7)else
(7)else
(7)Is the CQ- difference greater than the defined threshold?
(8)else
(8)Observer divergence detected in previous cycles and CILC stable?
Yes
Yes
Observer divergence in current activation cycle detected (1.15.4) [STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 2]
Yes
No observer divergence (1.15.5) [STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0]
No observer divergence (1.15.6) [STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0]
End of detection of observer divergence

4.1.17 Adaptation head-condition

In order to stop the long-term adaptation in case of controller outputs drift away, controller behaviour shall be predicted using following auxiliary variables:



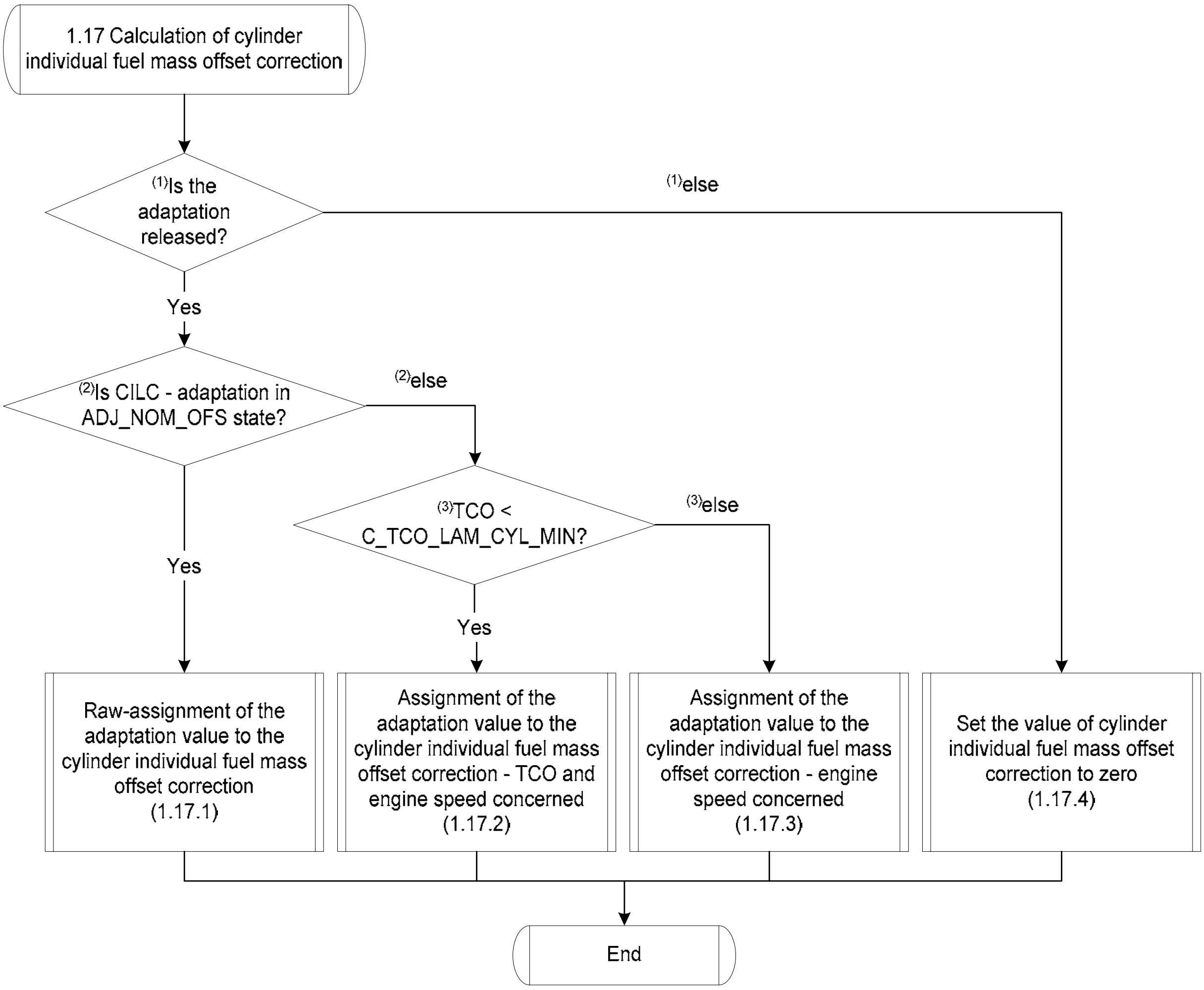
```
if(1) STATE_LAM_CYL_SEL[i] = 3 [calculation is exhaust bank selective]
then
    if(2)   (STATE_DELTA_CRK_CYL_LAM[i] = 3 'AD_INI' or
                    STATE_DELTA_CRK_CYL_LAM[i] = 4 'AD_DEC' or
                    STATE_DELTA_CRK_CYL_LAM[i] = 5 'AD_INC' or
                    LV_LAMB_CYL_SEL_CQ_DRIFT[i] = 1 or
                    STATE_LAMB_CYL_SEL_CQ_SLOP[i] ≠ 0 or
                    LV_FAC_CYL_LAM_COR_OSC[i] = 1)
    then
            LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 1
                                                        (4.1.17.1)
    else(2)
            LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 0
```

(4.1.17.2)

```
        endif(2)
    endif(1)
```

4.1.18 Calculation of cylinder individual fuel mass offset correction – weighted output

The value of cylinder individual fuel mass offset correction should be weighted versus engine speed, and temperature up to engine runs to the nominal temperature for activation of cylinder individual lambda control. The calculation shall be carried out independently of the activation condition for all of the cylinders.



```
if(1) LC_ADAPT_COR_OUT = 1
then
    if(2)   STATE_LAM_CYL_SEL_ADJ[i] = 6 [ADJ_NOM_OFS]
    then
            MFF_ADD_CYL_LAM_COR_OUT[x] = MFF_ADD_CYL_LAM_COR[x]
                                                                    (4.1.18.1)
    else(2)
            if(3)   TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_MIN
            then
                    MFF_ADD_CYL_LAM_COR_OUT[x] = MFF_ADD_CYL_LAM_COR[x]*
                                            IP_FAC_N_MFF_ADD_CYL_LAM_COR*
                                        IP_FAC_WUP_MFF_ADD_CYL_LAM (4.1.18.2)
            else(3)
                    MFF_ADD_CYL_LAM_COR_OUT[x] = MFF_ADD_CYL_LAM_COR[x]*
                                      IP_FAC_N_MFF_ADD_CYL_LAM_COR (4.1.18.3)
            endif(3)
    endif(2)
else(1)
    MFF_ADD_CYL_LAM_COR_OUT[x] = 0
                                                                    (4.1.18.4)
endif(1)
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_FAC_CYL_LAM_MIN	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lower limit of the cylinder individual controller output					
C_FAC_CYL_LAM_MAX	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper limit of the cylinder individual controller output					
C_FAC_CYL_LAM_VIRT_MIN	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lower virtual limit of the cylinder individual controller output					
C_FAC_CYL_LAM_VIRT_MAX	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper virtual limit of the cylinder individual controller output					
C_FAC_LAM_CYL_SEL_ADJ_CST_MIN	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lower limit of the cylinder individual cold-adaptation correction factor					
C_FAC_LAM_CYL_SEL_ADJ_CST_MAX	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper limit of the cylinder individual cold-adaptation correction factor					
C_FAC_LAM_CYL_SEL_ADJ_MIN	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lower limit of the cylinder individual nominal-adaptation correction factor					
C_FAC_LAM_CYL_SEL_ADJ_MAX	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper limit of the cylinder individual nominal-adaptation correction factor					
C_FAC_CYL_LAM_ABSV_DEC	1	0...7FFFH	0...49.99847	1.5259e-3	[%]
Absolute decrement constant to increase or decrease the cylinder individual injection time correction					
C_FAC_LAM_CYL_SEL_ADJ_VLD_THD	1	0...7FFFH	0...49.99847	1.5259e-3	[%]
Absolute-Controller-Sum-Threshold for evaluation of adaptation result					
C_DELTA_LAMB_CYL_SP	1	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Setpoint value					
C_CRLC_CYL_FIL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for the filter output					
C_CRLC_LAM_CYL_CST_FIL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for the filter output under engine cold condition					
C_CRLC_DELTA_LAMB_CYL_CQ_FIL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Filter correlation constant - characteristic quantity of estimated cylinder individual lambda deviation					
C_LAMB_CYL_SEL_CQ_DRIFT_THD	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
CQ- drift threshold for inhibition of cylinder individual lambda adaptation					
C_LAMB_CYL_SEL_CQ_DIF_REF_THD	1	8000...7FFFH	-0.5...0.49998	0.0153e-3	[-]
CQ- slope threshold for inhibition of cylinder individual lambda adaptation					
C_CTR_LAMB_CYL_SEL_CQ_DRIFT_THD	1	0...FFFFH	0...65535	1	[-]
CQ- drift counter threshold for inhibition of cylinder individual lambda adaptation					
C_CTR_LAMB_CYL_SEL_CQ_STAB_THD	1	0...FFFFH	0...65535	1	[-]
CQ- stability counter threshold to release the cylinder individual lambda adaptation					
C_CTR_LAM_CYL_SEL_REF_CLL_THD	1	0...FFFFH	0...65535	1	[-]
Threshold for segment based reference counter to close the control loop					
C_T_LAM_CYL_ADJ	1	0...FFH	0...255	1	[s]
Time recurrence for adaptation update					
C_T_LAM_CYL_ADJ_CST	1	0...FFH	0...255	1	[s]
Time recurrence for adaptation update under engine cold condition					
C_CRLC_EXV_FAC[NC_CYL_NR]	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Exhaust-valve manifold-port correlation factor					
C_CRK_CYL_LAM[NC_CYL_NR]	1	88...78H	-720...720	6	[°CRK]
Factor for different distance between lambda sensor and cylinder individual exhaust-valves					
C_TEMP_LAM_CYL_CST_MIN	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant temperature for the activation of CILC under cold condition					
C_TEMP_LAM_CYL_CST_MAX	1	0...FEH	-48...142.5	0.75	[°C]
Maximum coolant temperature for the activation of CILC under cold condition					
C_TEMP_LAM_CYL_MIN	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant temperature for the activation - nominal condition					

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_TEMP_THD_CYL_SEL_ADJ_COLD_RNG	1	0...FEH	-48...142.5	0.75	[°C]
Maximum coolant temperature threshold for consideration of the cold-range adaptation value					
C_TEMP_THD_CYL_SEL_ADJ_NOM_RNG	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant temperature threshold for consideration of the nominal-range adaptation value					
C_N_LAM_CYL_ADJ_CST_MIN	1	0...FFH	0...8160	32	[rpm]
Lower engine speed threshold for calculation of cylinder individual lambda adaptation (cold condition)					
C_N_LAM_CYL_ADJ_CST_MAX	1	0...FFH	0...8160	32	[rpm]
Upper engine speed threshold for calculation of cylinder individual lambda adaptation (cold condition)					
C_N_MIN_FAC_CYL_SEL_ADJ_L_RNG	1	0...FFH	0...8160	32	[rpm]
Lower engine speed threshold for calculation of cylinder individual lambda adaptation (low-range)					
C_N_MAX_FAC_CYL_SEL_ADJ_L_RNG	1	0...FFH	0...8160	32	[rpm]
Upper engine speed threshold for calculation of cylinder individual lambda adaptation (low-range)					
C_N_MIN_FAC_CYL_SEL_ADJ_H_RNG	1	0...FFH	0...8160	32	[rpm]
Lower engine speed threshold for calculation of cylinder individual lambda adaptation (high-range)					
C_N_MAX_FAC_CYL_SEL_ADJ_H_RNG	1	0...FFH	0...8160	32	[rpm]
Upper engine speed threshold for calculation of cylinder individual lambda adaptation (high-range)					
C_MFF_LAM_CYL_ADJ_CST_MIN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower fuel mass threshold for calculation of cylinder individual lambda adaptation (engine cold)					
C_MFF_LAM_CYL_ADJ_CST_MAX	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Upper fuel mass threshold for calculation of cylinder individual lambda adaptation (engine cold)					
C_MFF_MIN_FAC_CYL_SEL_ADJ_L_RNG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower fuel mass threshold for calculation of cylinder individual lambda adaptation (low-range)					
C_MFF_MAX_FAC_CYL_SEL_ADJ_L_RNG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Upper fuel mass threshold for calculation of cylinder individual lambda adaptation (low-range)					
C_MFF_MIN_FAC_CYL_SEL_ADJ_H_RNG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower fuel mass threshold for calculation of cylinder individual lambda adaptation (high-range)					
C_MFF_MAX_FAC_CYL_SEL_ADJ_H_RNG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Upper fuel mass threshold for calculation of cylinder individual lambda adaptation (high-range)					
C_MFF_SP_THD_CYL_SEL_ADJ_L_RNG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Maximum fuel mass threshold for consideration of the low range adaptation value in the injection correction					
C_MFF_SP_THD_CYL_SEL_ADJ_H_RNG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Minimum fuel mass threshold for consideration of the high range adaptation value in the injection correction					
C_MFF_MIN_ADD_LAM_CYL_SEL_ADJ	1	8000...7FFFH	-694.51059...694.48940	0.0211948	[mg/stk]
Minimum value of cylinder individual fuel mass offset					
C_MFF_MAX_ADD_LAM_CYL_SEL_ADJ	1	8000...7FFFH	-694.51059...694.48940	0.0211948	[mg/stk]
Maximum value of cylinder individual fuel mass offset					
LC_ADAPT_COR_OUT	1	0...1H	0...1	1	[-]
Adaptation switch (=1)					
LC_LAM_CYL_SEL_OPL	1	0...1H	0...1	1	[-]
Switch for open loop operation (=1)					
LC_FAC_LAM_CYL_SEL_ADJ_CST_ENA	1	0...1H	0...1	1	[-]
Switch enabling the adaptation under engine cold condition (=1)					
LC_LAM_CYL_SEL_ERR_RST	1	0...1H	0...1	1	[-]
Switch enabling the reset of adaptation values in case of error (=1)					
LC_LAMB_DELTA_MDL_HLD	1	0...1H	0...1	1	[-]
Calculation switch for the choice of the discretization method of the sensor model (= 0 'zoh'; = 1 'foh')					
LC_FAC_LAM_CYL_SEL_ADJ_P_I_ENA	1	0...1H	0...1	1	[-]
Calculation switch for the choice of the adaptation method (= 0 'I - component concerned'; = 1 'PI concerned')					
LC_FAC_CYL_LAM_P_SWI	1	0...1H	0...1	1	[-]
Calculation switch for the choice of the control structure (= 0 'I - controller'; = 1 'PI controller')					
IP_LAMB_CYL[NC_CBK_EX_NR]	16	0...7FFFH	0...7.99975	0.2441e-3	[-]
LDP_VLS_CYL_SEL_IP_LAMB	16	0...3FFH	0...4.99511	4.8828e-3	[V]
Characteristic curve of the WRAF sensor					

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
IP_EXP_LAMB_MDL[NC_CBK_EX_NR]	16	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_FAC_T_1_CYBL	16	0...FFFFH	0...63.99902	0.9766e-3	[-]
Exponential function for the sensor model					
IP_N1_FAC_LAMB_MDL[NC_CBK_EX_NR]	16	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_FAC_T_1_CYBL	16	0...FFFFH	0...63.99902	0.9766e-3	[-]
N1 coefficient of the sensor model					
IP_N2_FAC_LAMB_MDL[NC_CBK_EX_NR]	16	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_FAC_T_1_CYBL	16	0...FFFFH	0...63.99902	0.9766e-3	[-]
N2 coefficient of the sensor model					
IP_CRK_SYN[NC_CBK_EX_NR]	8*8	0...F0H	0...1440	6	[°CRK]
LDPM_N_1_CYBL	8	0...1FE0H	0...8160	1	[rpm]
LDPM_MAF_1_CYBL	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Phase displacement					
IP_FAC_OPP_LAM_CYL_SEL_ADJ	8*8	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_N_1_CYBL	8	0...1FE0H	0...8160	1	[rpm]
LDPM_MAF_1_CYBL	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual lambda adaptation factor depending on operating point					
IP_FAC_OPP_LAM_CYL_SEL_ADJ_FAST	8*8	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_N_1_CILC	8	0...1FE0H	0...8160	1	[rpm]
LDPM_MAF_1_CILC	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual lambda adaptation factor depending on operating point (fast adaptation)					
IP_CRK_DELTA_AMP_CYL_LAM	6	88...78H	-720...720	6	[°CRK]
LDP_AMP_IP_CRK_DELTA_AMP	6	0...FFFFH	0...5434	0.0829175	[hPa]
Phase shift deviation caused by elevation					
IP_FAC_ERR_LAM_SEL	6	0...FFFFH	0...255.99609	3.9063e-3	[1/s]
LDP_T1_LAMB_CYL_IP_FAC_ERR	6	0...FFFFH	0...1.99996	0.0305e-3	[s]
Weight coefficient					
IP_CRLC_CYL_LAM	8*8	0...FFFFH	0...7.99987	0.1221e-3	[%]
LDPM_N_1_CILC	8	0...1FE0H	0...8160	1	[rpm]
LDPM_MAF_1_CILC	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Correlation characteristic for AFR equalisation - nominal controller parameter					
IP_CRLC_INT_FAST_CYL_LAM	8*8	0...FFFFH	0...7.99987	0.1221e-3	[%]
LDPM_N_1_CILC	8	0...1FE0H	0...8160	1	[rpm]
LDPM_MAF_1_CILC	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Correlation characteristic for AFR equalisation - fast controller parameter					
IP_FAC_GAIN_P_LAM_CYL_SEL	8*8	0...FFFFH	0...63.99902	0.9766e-3	[%]
LDPM_N_1_CILC	8	0...1FE0H	0...8160	1	[rpm]
LDPM_MAF_1_CILC	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Gain of cylinder individual controller (P-component)					
IP_CRLC_P_NEUT_RNG	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_DELTA_LAMB_DIF_CYL_P	8	0...FFFFH	-8...7.99975	0.2441e-3	[-]
Dead zone for calculation of P-component					
IP_CRLC_I_NEUT_RNG	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_DELTA_LAMB_DIF_CYL_I	8	0...FFFFH	-8...7.99975	0.2441e-3	[-]
Dead zone for calculation of I-component					
IP_FAC_VAL_LAM_CYL_SEL_REAC	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_MFF_DELTA_LAM_CYL_IP_FAC	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
MFF-difference- dependent factor for internal states at reactivation moment					
IP_FAC_N_MFF_ADD_CYL_LAM_COR	6	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_N_32_IP_FAC_MFF_ADD_CYL	6	0...FFH	0...8160	32	[rpm]
Engine speed dependent factor for realization of cylinder individual fuel mass offset correction					
IP_FAC_WUP_MFF_ADD_CYL_LAM	6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_TCO_1_CYBL	6	0...FEH	-48...142.5	0.75	[°C]
Temperature dependent factor for realization of cylinder individual fuel mass offset correction during warm-up					
IP_T1_LAMB_CYL_SEL[NC_CBK_EX_NR]	8*8	0...FFFFH	0...1.99996	0.0305e-3	[s]
LDPM_N_1_CYBL	8	0...1FE0H	0...8160	1	[rpm]

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
LDPM_MAF_1_CYBL	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Time lag of the lambda-sensor model specified for cylinder individual lambda control					

4.2 Phase Shift Adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]	O/V/S	88...78H	-720...720	6	[°CRK]
Value of adapted phase displacement					
FAC_CYL_LAM_SAVE[NC_CYL_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Value of the saved cylinder individual lambda controller output					
FAC_CYL_LAM_ABSV_SUM_GRD[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Gradient of the CILC outputs absolute sum					
FAC_CYL_LAM_ABSV_SUM_PRE[NC_CBK_EX_NR]	V	0...7FFFH	0...49.99847	1.5259e-3	[%]
Sum of the absolute cylinder individual controller outputs - previous sample					
CRK_CYL_LAM_INT[NC_CBK_EX_NR]	V	88...78H	-720...720	6	[°CRK]
Output of the integral-action					
CRK_CYL_LAM_DELTA_INI[NC_CBK_EX_NR]	O/V/S	88...78H	-720...720	6	[°CRK]
Initial CRK - adaptation value					
CRK_CHG_FAC_PHA_SHIFT_AD[NC_CBK_EX_NR]	V	1...78H	1...120	1	[-]
Recursive change factor					
STATE_DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]	O/V	0H 1H 2H 3H 4H 5H 6H	AD_PAS AD_ACT AD_CDN AD_INI AD_DEC AD_INC AD_HOLD	1	[-]
Adaptation state					
CTR_SUM_FAC_CYL_LAM_COR_OSC[NC_CYL_NR]	V	0...FFFFH	0...65535	1	[-]
Duration of oscillation monitoring					
CTR_CQ_MON_PHA_SHIFT_AD[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Period of CQ monitoring					
CTR_ER_MON_PHA_SHIFT_AD[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Period of engine roughness monitoring					
CTR_CQ_MAX_OSC[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Duration of exceeded CQ limit					
CTR_RESP_LAM_CYL_SEL[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Response time for adaptation result					
CTR_RESP_CYC_LAM_CYL_SEL[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter for evaluation of cylinder individual lambda control response					
CTR_FAC_CYL_LAM_COR_DIF_ABSV[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter for improved cycles of cylinder individual lambda control response - FAC_CYL_LAM_COR based					
CTR_CQ_FIL_FALL[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter for improved cycles of cylinder individual lambda control response - CQ based					
CTR_PHA_SHIFT_AD_TRIG[NC_CBK_EX_NR]	O/V/S	0...FFH	0...255	1	[-]
Counter for triggering the phase shift adaptation					

LV_DELTA_CRK_CYL_LAM_ERR[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Error flag indicating the adaptation value is not correct					
LV_FAC_CYL_LAM_OSC_MAX_THD[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Flag indicating the upper limit of cylinder individual lambda control was reached - oscillation monitoring					
LV_FAC_CYL_LAM_OSC_MIN_THD[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Flag indicating the lower limit of cylinder individual lambda control was reached - oscillation monitoring					
LV_FAC_CYL_LAM_COR_OSC[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Oscillation detected					
LV_FAC_CYL_LAM_COR_LIM_OSC[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Alternating limitation detected					
LV_ER_ABSV_SUM_GRD_RISE[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Unstable engine roughness detected					
LV_STATE_DELTA_CRK_WAIT[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Waiting for adaptation response					
ER_STND_MMV_DIF_BAL_ABSV_SUM[NC_CBK_EX_NR]	V	0...7FFFH	0...325.77	9.942e-3	[1/s²]
Sum of the absolute difference of filtered normalized and average engine roughness value					
ER_STND_MMV_DIF_ABSV_SUM_PRE[NC_CBK_EX_NR]	V	0...7FFFH	0...325.77	9.942e-3	[1/s²]
Sum of the absolute difference of filtered normalized and average engine roughness value - previous sample					
ER_STND_MMV_DIF_ABSV_SUM_GRD[NC_CBK_EX_NR]	V	8000...7FFFH	-325.78...325.77	9.942e-3	[1/s²]
Gradient of the absolute difference of filtered normalized and average engine roughness value					

Input data:

<table>
<tr><td colspan="3">NC_CYL_NR</td><td colspan="3">NC_CBK_EX_NR</td><td colspan="3">SEG_NR</td><td colspan="3">LV_DC</td></tr>
<tr><td colspan="4">FAC_CYL_LAM[NC_CYL_NR]</td><td colspan="4">LV_ES</td><td colspan="4">LV_ER_STND_ER_BAL_ACT</td></tr>
<tr><td colspan="6">STATE_LAM_CYL_SEL[NC_CBK_EX_NR]</td><td colspan="6">DELTA_LAMB_CYL_SEL_CQ[NC_CBK_EX_NR]</td></tr>
<tr><td colspan="6">LV_FAC_CYL_LAM_LIM_MIN[NC_CYL_NR]</td><td colspan="6">LV_FAC_CYL_LAM_LIM_MAX[NC_CYL_NR]</td></tr>
<tr><td colspan="6">LF_LS_CBK_EX_LAM_CYL_SEL_CONF</td><td colspan="6">LV_LAM_CYL_SEL_CTL_DRIFT_PRED[NC_CBK_EX_NR]</td></tr>
<tr><td colspan="6">ER_STND_MMV_DIF_BAL[NC_CYL_NR]</td><td colspan="6">CTR_LAMB_CYL_SEL_CQ_DRIFT_THD</td></tr>
<tr><td colspan="6">FAC_CYL_LAM_ABSV_SUM[NC_CBK_EX_NR]</td><td colspan="6">DELTA_LAMB_CYL_SEL_CQ_FIL[NC_CBK_EX_NR]</td></tr>
<tr><td colspan="6">LV_CRK_CYL_LAM_DELTA_RST_LS_EXT[NC_CBK_EX_NR]</td><td colspan="6">STATE_LAMB_CYL_SEL_CQ_SLOP[NC_CBK_EX_NR]</td></tr>
</table>

FUNCTION DESCRIPTION:

General information:

This module enables an adaptation of phase shift, which might be influenced by ageing effect of wide range A/F sensor or even calibration-error for instance.

```
if(1) two separate exhaust banks are concerned
then
    i = 1, for cylinder bank 1 ( x = 0, 2, 4, ... )
    i = 2, for cylinder bank 2 ( x = 1, 3, 5, ... )
else(1)
    i = 1 ( x = 0 ... 3 or 4 )
endif(1)
```

Description of index "[i]" and "[x]":

x: logical cylinder
*use **NC_CYL_NR** : number of engine cylinders – index "[x]"*
*use **LF_LS_CBK_EX_LAM_CYL_SEL_CONF**: pattern for allocation of physical cylinders to exhaust bank – index "[i]"*

Time recurrence: every TDC

Application conditions:

4.2.1 Initialization (at reset)

The following initialization shall be carried out after a RESET and upon leaving the engine state Engine Stop (LV_ES).

STATE_DELTA_CRK_CYL_LAM[i] = 0 'AD_PAS'

(4.2.1.1)

FAC_CYL_LAM_SAVE[x] = 0

(4.2.1.2)

LV_FAC_CYL_LAM_COR_OSC[i] = 0

(4.2.1.3)

LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0

(4.2.1.4)

LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 0

(4.2.1.5)

LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 0

(4.2.1.6)

LV_STATE_DELTA_CRK_WAIT[i] = 0

(4.2.1.7)

CTR_SUM_FAC_CYL_LAM_COR_OSC[x] = 0

(4.2.1.8)

CTR_CQ_MON_PHA_SHIFT_AD[i] = 0

(4.2.1.9)

CTR_ER_MON_PHA_SHIFT_AD[i] = 0

(4.2.1.10)

CTR_CQ_MAX_OSC[i] = 0

(4.2.1.11)

CTR_RESP_LAM_CYL_SEL[i] = 0

(4.2.1.12)

CTR_RESP_CYC_LAM_CYL_SEL[i] = 0

(4.2.1.13)

CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] = 0

(4.2.1.14)

CTR_CQ_FIL_FALL[i] = 0

(4.2.1.15)

CRK_CHG_FAC_PHA_SHIFT_AD[i] = 1

(4.2.1.16)

LV_DELTA_CRK_CYL_LAM_ERR[i] = 0

(4.2.1.17)

LV_ER_ABSV_SUM_GRD_RISE[i] = 0

(4.2.1.18)

ER_STND_MMV_DIF_BAL_ABSV_SUM[i] = 0

(4.2.1.19)

ER_STND_MMV_DIF_ABSV_SUM_PRE[i] = 0

(4.2.1.20)

ER_STND_MMV_DIF_ABSV_SUM_GRD[i] = 0

(4.2.1.21)

FAC_CYL_LAM_ABSV_SUM_PRE[i] = 0

(4.2.1.22)

FAC_CYL_LAM_ABSV_SUM_GRD[i] = 0

(4.2.1.23)

CRK_CYL_LAM_INT[i] = DELTA_CRK_CYL_LAM[i]

(4.2.1.24)

```
if(1)  DELTA_CRK_CYL_LAM[i] = 0
then
    CRK_CYL_LAM_DELTA_INI[i] = C_CRK_CYL_LAM_DELTA_INI
                                                        (4.2.1.25)
endif(1)
```

4.2.2 Reset at E²PROM – failure

At E²PROM – failure:

DELTA_CRK_CYL_LAM[i] = 0

(4.2.2.1)

CRK_CYL_LAM_DELTA_INI[i] = 0

(4.2.2.2)

CTR_PHA_SHIFT_AD_TRIG[i] = 0

(4.2.2.3)

4.2.3 Adaptation reset by external tool intervention

Following calculation shall be carried out independently of the activation condition (time recurrence: 100 ms):

```
if(1)  LV_CRK_CYL_LAM_DELTA_RST_LS_EXT[i] = 1
then
    DELTA_CRK_CYL_LAM[i] = 0
                                                        (0.1)
    CRK_CYL_LAM_DELTA_INI[i] = C_CRK_CYL_LAM_DELTA_INI
                                                        (0.2)
    FAC_CYL_LAM_SAVE[x] = 0
                                                        (0.3)
    LV_FAC_CYL_LAM_COR_OSC[i] = 0
                                                        (0.4)
    LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0
                                                        (0.5)
    LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 0
                                                        (0.6)
```

LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 0

(0.7)

LV_STATE_DELTA_CRK_WAIT[i] = 0

(0.8)

CTR_SUM_FAC_CYL_LAM_COR_OSC[x] = 0

(0.9)

CTR_CQ_MON_PHA_SHIFT_AD[i] = 0

(0.10)

CTR_ER_MON_PHA_SHIFT_AD[i] = 0

(0.11)

CTR_CQ_MAX_OSC[i] = 0

(0.12)

CTR_RESP_LAM_CYL_SEL[i] = 0

(0.13)

CTR_RESP_CYC_LAM_CYL_SEL[i] = 0

(0.14)

CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] = 0

(0.15)

CTR_CQ_FIL_FALL[i] = 0

(0.16)

CRK_CHG_FAC_PHA_SHIFT_AD[i] = 1

(0.17)

LV_DELTA_CRK_CYL_LAM_ERR[i] = 0

(0.18)

LV_ER_ABSV_SUM_GRD_RISE[i] = 0

(0.19)

ER_STND_MMV_DIF_BAL_ABSV_SUM[i] = 0

(0.20)

ER_STND_MMV_DIF_ABSV_SUM_PRE[i] = 0

(0.21)

ER_STND_MMV_DIF_ABSV_SUM_GRD[i] = 0

(0.22)

FAC_CYL_LAM_ABSV_SUM_PRE[i] = 0

(0.23)

```
        FAC_CYL_LAM_ABSV_SUM_GRD[i] = 0
                                                                        (0.24)
        CRK_CYL_LAM_INT[i] = 0
                                                                        (0.25)
        CTR_PHA_SHIFT_AD_TRIG[i] = 0
                                                                        (0.26)
    endif(1)
```

4.2.4 Activation

```
    if(1) LV_DELTA_CRK_CYL_LAM_ERR[i] = 0
    then
        if(2)   STATE_DELTA_CRK_CYL_LAM[i] = 0 'AD_PAS'
        then
                STATE_DELTA_CRK_CYL_LAM[i] = 1 'AD_ACT'
                                                                     (4.2.4.1)
        endif(2)
    end if(1)

    if(1) STATE_LAM_CYL_SEL[i] = 0 [calculation is exhaust bank selective]
    then
        if(2)   (LV_FAC_CYL_LAM_COR_OSC[i] = 1 and
                        CTR_PHA_SHIFT_AD_TRIG[i] < C_CTR_PHA_SHIFT_AD_TRIG_THD)
        then
                LV_FAC_CYL_LAM_COR_OSC[i] = 0
                                                                     (4.2.4.2)
                LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 0
                                                                     (4.2.4.3)
                LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 0
                                                                     (4.2.4.4)
                CTR_SUM_FAC_CYL_LAM_COR_OSC[x] = 0
                                                                     (4.2.4.5)
                CTR_CQ_MON_PHA_SHIFT_AD[i] = 0
                                                                     (4.2.4.6)
                CTR_CQ_MAX_OSC[i] = 0
                                                                     (4.2.4.7)
        endif(2)
        if(2)   (LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 1 and
                        CTR_PHA_SHIFT_AD_TRIG[i] < C_CTR_PHA_SHIFT_AD_TRIG_THD)
        then
                LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0
                                                                     (4.2.4.8)
        endif(2)
        if(2)   (LV_ER_ABSV_SUM_GRD_RISE[i] = 1 and
                        CTR_PHA_SHIFT_AD_TRIG[i] < C_CTR_PHA_SHIFT_AD_TRIG_THD)
        then
                LV_ER_ABSV_SUM_GRD_RISE[i] = 0
```

```
                                                              (4.2.4.9)
        endif(2)
    end if(1)
```

4.2.5 Deactivation

```
    if(1) LV_DELTA_CRK_CYL_LAM_ERR[i] = 1
    then
        STATE_DELTA_CRK_CYL_LAM[i] = 0 'AD_PAS'
                                                              (4.2.5.1)
    endif(1)
```

Formula section:

4.2.6 Monitoring the engine roughness

```
    if(1) STATE_LAM_CYL_SEL[i] = 3 [calculation is exhaust bank selective]
    then
        if(2)   (SEG_NR = 0 [calculation for exhaust bank 1] or
                    SEG_NR = C_NR_SEG_REF_CBK_EX_2 [calculation for exhaust bank 2]
        then
            if(3)   LV_ER_STND_ER_BAL_ACT = 1
            then
                if(4)   LV_STATE_DELTA_CRK_WAIT[i] = 0
                then
                    if(5)   CTR_ER_MON_PHA_SHIFT_AD[i] = 0
                    then
                        FAC_CYL_LAM_ABSV_SUM_PRE[i] =
                                    FAC_CYL_LAM_ABSV_SUM[i]   (4.2.6.1)
                    endif(5)
                    increment   CTR_ER_MON_PHA_SHIFT_AD[i]
                                                              (4.2.6.2)
                    if(5)   CTR_ER_MON_PHA_SHIFT_AD[i] =
                                    C_CTR_ER_MON_PHA_SHIFT_AD_MAX
                    then
                        if(6)   ER_STND_MMV_DIF_BAL_ABSV_SUM[i] ≠ 0
                        then
                            ER_STND_MMV_DIF_ABSV_SUM_PRE[i] =
                             ER_STND_MMV_DIF_BAL_ABSV_SUM[i]  (4.2.6.3)
                        endif(6)
                        ER_STND_MMV_DIF_BAL_ABSV_SUM[i] =
                            $\sum_{x=1}^{\alpha} |ER\_STND\_MMV\_DIF\_BAL[x]|$
                        $\alpha$: nr. of cylinders per bank
                                                              (4.2.6.4)
                        [Remark: the cylinder based variables (index x) shall be
                        assigned to the corresponding exhaust bank (index i)]

                        if(6)   ER_STND_MMV_DIF_ABSV_SUM_PRE[i] ≠ 0
```

```
                    then
                        ER_STND_MMV_DIF_ABSV_SUM_GRD[i] =
                            ER_STND_MMV_DIF_BAL_ABSV_SUM[i] –
                        ER_STND_MMV_DIF_ABSV_SUM_PRE[i] (4.2.6.5)
                    endif(6)
                    FAC_CYL_LAM_ABSV_SUM_GRD[i] =
                                      FAC_CYL_LAM_ABSV_SUM[i] –
                          FAC_CYL_LAM_ABSV_SUM_PRE[i] (4.2.6.6)

                    CTR_ER_MON_PHA_SHIFT_AD[i] = 0
                                                          (4.2.6.7)
                endif(5)
            endif(4)
        endif(3)
    endif(2)
else(1)
    CTR_ER_MON_PHA_SHIFT_AD[i] = 0
                                                          (4.2.6.8)
endif(1)
```

4.2.7 Detection of the cylinder individual lambda controller output oscillation:

The controller outputs of cylinder individual lambda control are monitored during the time the function is in closed loop mode (STATE_LAM_CYL_SEL[i] = CLL).

```
if(1) STATE_LAM_CYL_SEL[i] = 3 [calculation is exhaust bank selective]
then
    if(2) (SEG_NR = 0 [calculation for exhaust bank 1] or
               SEG_NR = C_NR_SEG_REF_CBK_EX_2 [calculation for exhaust bank 2]
    then
        if(3) (LC_CRK_DELTA_CQ_SWI = 0 and
                                    LV_ER_ABSV_SUM_GRD_RISE[i] = 0)
        then
            if(4) [(FAC_CYL_LAM[x] >
                           C_FAC_CYL_LAM_MAX_PHA_SHIFT_AD) and
                           LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 0 and
                           LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0]
            then
                LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 1
                                                            (4.2.7.1)
            endif(4)
            if(4) LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 1
            then
                if(5) LV_FAC_CYL_LAM_COR_OSC[i] = 0
                then
                    increment   CTR_SUM_FAC_CYL_LAM_COR_OSC[x]
                                                            (4.2.7.2)
                endif(5)
                if(5) [(FAC_CYL_LAM[x] <
                           C_FAC_CYL_LAM_MIN_PHA_SHIFT_AD) and
                           (CTR_SUM_FAC_CYL_LAM_COR_OSC[x] <
                           C_CTR_SUM_FAC_CYL_LAM_OSC_MAX)]
                then
                    if(6) LV_FAC_CYL_LAM_COR_OSC[i] = 0
                    then
                        LV_FAC_CYL_LAM_COR_OSC[i] = 1
                                                            (4.2.7.3)
                        [Remark: the cylinder based Oscillation (index x)
                        shall be assigned to the corresponding exhaust bank
                                                            (index i)]
                        if(7) CTR_PHA_SHIFT_AD_TRIG[i] <
                                   C_CTR_PHA_SHIFT_AD_TRIG_THD
                        then
                            increment (CTR_PHA_SHIFT_AD_TRIG[i])
                                                            (4.2.7.4)
                        endif(7)
                    endif(6)
                endif(5)
```

```
endif$^{(4)}$

if$^{(4)}$  [(FAC_CYL_LAM[x] <
                        C_FAC_CYL_LAM_MIN_PHA_SHIFT_AD) and
                        LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 0 and
                        LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0]
then
        LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 1
                                                            (4.2.7.5)
endif$^{(4)}$
if$^{(4)}$  LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 1
then
        if$^{(5)}$  LV_FAC_CYL_LAM_COR_OSC[i] = 0
        then
                increment  CTR_SUM_FAC_CYL_LAM_COR_OSC[x]
                                                            (4.2.7.6)
        endif$^{(5)}$
        if$^{(5)}$  [(FAC_CYL_LAM[x] >
                        C_FAC_CYL_LAM_MAX_PHA_SHIFT_AD) and
                        (CTR_SUM_FAC_CYL_LAM_COR_OSC[x] <
                        C_CTR_SUM_FAC_CYL_LAM_OSC_MAX)]
        then
                if$^{(6)}$  LV_FAC_CYL_LAM_COR_OSC[i] = 0
                then
                        LV_FAC_CYL_LAM_COR_OSC[i] = 1
                                                            (4.2.7.7)
                        [Remark: the cylinder based Oscillation (index x)
                        shall be assigned to the corresponding exhaust bank
                        (index i)]
                        if$^{(7)}$  CTR_PHA_SHIFT_AD_TRIG[i] <
                                C_CTR_PHA_SHIFT_AD_TRIG_THD
                        then
                                increment (CTR_PHA_SHIFT_AD_TRIG[i])
                                                            (4.2.7.8)
                        endif$^{(7)}$

                endif$^{(6)}$

        endif$^{(5)}$
endif$^{(4)}$

if$^{(4)}$  [(CTR_SUM_FAC_CYL_LAM_COR_OSC[x] =
                        C_CTR_SUM_FAC_CYL_LAM_OSC_MAX) and
                        LV_FAC_CYL_LAM_COR_OSC[i] = 0)]
then
        CTR_SUM_FAC_CYL_LAM_COR_OSC[x] = 0
                                                            (4.2.7.9)
        LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 0
                                                            (4.2.7.10)
        LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 0
```

```
endif(4)                                   (4.2.7.11)
```

4.2.8 Detection of Cylinder Individual lambda Control Instability – based on CILC-controller output:

For detection of unstable state of CILC the controller output of adjacent cylinders are considered. Should two of the cylinders be at the opposite limit and a drift-away behaviour of cylinder individual lambda control be detected, thus the unstable behaviour of CILC is assumed.

```
if$^{(4)}$    [{(LV_FAC_CYL_LAM_LIM_MIN[x] = 1
                  and LV_FAC_CYL_LAM_LIM_MAX[x+1] = 1) or
                  (LV_FAC_CYL_LAM_LIM_MAX[x] = 1 and
                  LV_FAC_CYL_LAM_LIM_MIN[x+1] = 1)} and
                  LV_LAM_CYL_SEL_CTL_DRIFT_PRED[i] = 1]
              [Remark: the index x and x+1 relate to the in succession
               firing cylinders on the same exhaust bank]
then
    if$^{(5)}$    LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0
    then
        LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 1
                                                            (4.2.8.1)
        [Remark: the cylinder based Oscillation (index x) shall be
        assigned to the corresponding exhaust bank (index i)]
        if$^{(6)}$    CTR_PHA_SHIFT_AD_TRIG[i] <
                          C_CTR_PHA_SHIFT_AD_TRIG_THD
        then
            increment (CTR_PHA_SHIFT_AD_TRIG[i])
                                                            (4.2.8.2)
        endif$^{(6)}$
    endif$^{(5)}$
else$^{(4)}$
    LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0
                                                            (4.2.8.3)
endif$^{(4)}$
```

4.2.9 Detection of Cylinder Individual lambda Control Instability – based on CILC - Characteristic Quantity:

For detection of unstable state of CILC the value of Characteristic Quantity (CQ) is considered (LC_CRK_DELTA_CQ_SWI = 1).

```
elseif$^{(3)}$    (LC_CRK_DELTA_CQ_SWI = 1 and
                      LV_ER_ABSV_SUM_GRD_RISE[i] = 0)
then
    increment    CTR_CQ_MON_PHA_SHIFT_AD[i]
                                                            (4.2.9.1)
    if$^{(4)}$    DELTA_LAMB_CYL_SEL_CQ_FIL[i] $\geq$
                      C_DELTA_LAMB_CYL_SEL_CQ_AD_THD
    then
        if$^{(5)}$    LV_FAC_CYL_LAM_COR_OSC[i] = 0
```

```
            then
                increment     CTR_CQ_MAX_OSC[i]
                                                              (4.2.9.2)
            endif$^{(5)}$
            if$^{(5)}$    [(CTR_CQ_MAX_OSC[i] =
                                  C_CTR_CQ_MIN_PHA_SHIFT_AD) and
                                  (CTR_CQ_MON_PHA_SHIFT_AD[i] <
                                  C_CTR_CQ_MON_PHA_SHIFT_AD_MAX)]

            then
                if$^{(6)}$    LV_FAC_CYL_LAM_COR_OSC[i] = 0
                then
                    LV_FAC_CYL_LAM_COR_OSC[i] = 1
                                                              (4.2.9.3)
                    if$^{(7)}$    CTR_PHA_SHIFT_AD_TRIG[i] <
                                      C_CTR_PHA_SHIFT_AD_TRIG_THD
                    then
                        increment (CTR_PHA_SHIFT_AD_TRIG[i])
                                                              (4.2.9.4)
                    endif$^{(7)}$

                endif$^{(6)}$

            endif$^{(5)}$
        endif$^{(4)}$
        if$^{(4)}$    CTR_CQ_MON_PHA_SHIFT_AD[i] =
                              C_CTR_CQ_MON_PHA_SHIFT_AD_MAX
        then
            CTR_CQ_MON_PHA_SHIFT_AD[i] = 0
                                                              (4.2.9.5)
            if$^{(5)}$    LV_FAC_CYL_LAM_COR_OSC[i] = 0
            then
                CTR_CQ_MAX_OSC[i] = 0
                                                              (4.2.9.6)
            endif$^{(5)}$
        endif$^{(4)}$

    endif$^{(3)}$
```

4.2.10 Detection of Cylinder Individual lambda Control Instability – based on engine roughness:

```
        if(3)   (LV_FAC_CYL_LAM_COR_OSC[i] = 0 and LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0)
        then
            if(4)   LV_ER_ABSV_SUM_GRD_RISE[i] = 0
            then
                if(5)   (ER_STND_MMV_DIF_ABSV_SUM_GRD[i] > C_ER_ABSV_GRD_RISE_THD_CRK_CHG and FAC_CYL_LAM_ABSV_SUM_GRD[i] > C_FAC_CYL_LAM_ABSV_GRD_RISE_THD)
                then
                    LV_ER_ABSV_SUM_GRD_RISE[i] = 1
                                                                    (4.2.10.1)
                    FAC_CYL_LAM_ABSV_SUM_PRE[i] = FAC_CYL_LAM_ABSV_SUM[i]
                                                                    (4.2.10.2)
                    if(6)   CTR_PHA_SHIFT_AD_TRIG[i] < C_CTR_PHA_SHIFT_AD_TRIG_THD
                    then
                        increment (CTR_PHA_SHIFT_AD_TRIG[i])
                                                                    (4.2.10.3)
                    endif(6)
                endif(5)
            endif(4)
        endif(3)
    endif(2)
endif(1)
```

4.2.11 Adaptation start

```
if(1)   (SEG_NR = 0 [calculation for exhaust bank 1] or
            SEG_NR = C_NR_SEG_REF_CBK_EX_2 [calculation for exhaust bank 2]
then

    if(2)   [(STATE_DELTA_CRK_CYL_LAM[i] = 1 'AD_ACT' or
            STATE_DELTA_CRK_CYL_LAM[i] = 6 'AD_HOLD') and
            (LV_FAC_CYL_LAM_COR_OSC[i] = 1 or
            LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 1 or
            LV_ER_ABSV_SUM_GRD_RISE[i] = 1) and
            CTR_PHA_SHIFT_AD_TRIG[i] = C_CTR_PHA_SHIFT_AD_TRIG_THD]
    then
        STATE_DELTA_CRK_CYL_LAM[i] = 2 'AD_CDN'
```

```
                                                                    (4.2.11.1)
    endif$^{(2)}$

    if$^{(2)}$  (STATE_DELTA_CRK_CYL_LAM[i] = 2 'AD_CDN' and
                LV_STATE_DELTA_CRK_WAIT[i] = 0 and STATE_LAM_CYL_SEL[i] = 3)
                                       [calculation is exhaust bank selective]
    then
        STATE_DELTA_CRK_CYL_LAM[i] = 3 'AD_INI'
                                                                    (4.2.11.2)
        CTR_RESP_CYC_LAM_CYL_SEL[i] =
                                  C_CTR_RESP_CYC_MIN_LAM_CYL_SEL (4.2.11.3)
        FAC_CYL_LAM_SAVE[x] = FAC_CYL_LAM[x]
                                                                    (4.2.11.4)
        if$^{(3)}$   LV_ER_ABSV_SUM_GRD_RISE[i] = 1
        then
            CTR_RESP_LAM_CYL_SEL[i] =
                                         C_CTR_ER_RESP_MIN_LAM_CYL_SEL
                                                                    (4.2.11.5)
        else$^{(3)}$
            CTR_RESP_LAM_CYL_SEL[i] = C_CTR_RESP_MIN_LAM_CYL_SEL
                                                                    (4.2.11.6)
        endif$^{(3)}$
        CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] = 0
                                                                    (4.2.11.7)
        CTR_CQ_FIL_FALL[i] = 0
                                                                    (4.2.11.8)
    endif$^{(2)}$
  endif$^{(1)}$
```

4.2.12 Evaluation of adaptation result

```
if(1) STATE_LAM_CYL_SEL[i] = 3 [calculation is exhaust bank selective]
then
    if(2)   (SEG_NR = 0 [calculation for exhaust bank 1] or
                SEG_NR = C_NR_SEG_REF_CBK_EX_2 [calculation for exhaust bank 2]
    then

        if(3)   (LC_CRK_DELTA_CQ_SWI = 0 and
                                        LV_ER_ABSV_SUM_GRD_RISE[i] = 0)
        then
            if(4)   (LV_STATE_DELTA_CRK_WAIT[i] = 1 and
                                        LV_FAC_CYL_LAM_COR_OSC[i] = 1)
            then
                decrement   CTR_RESP_LAM_CYL_SEL[i]
                                                            (4.2.12.1)
                if(5)   CTR_RESP_LAM_CYL_SEL[i] = 0
                then
                    if(6)   [[abs(FAC_CYL_LAM_SAVE[x] –
                                            FAC_CYL_LAM[x]) ≤
                                C_FAC_CYL_LAM_DIF_ABSV_MAX] and
                            (LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0)]
                    then
                        increment
                        CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] (4.2.12.2)
                    endif(6)
                    [Remark: the counter
                    CTR_FAC_CYL_LAM_COR_DIF_ABSV[i]
                    shall be incremented only once provided that the
                    condition is fulfilled for all of the cylinders at the
                    concerned exhaust bank (AND-logical operator)]
                    CTR_RESP_LAM_CYL_SEL[i] =
                            C_CTR_RESP_MIN_LAM_CYL_SEL (4.2.12.3)
                    FAC_CYL_LAM_SAVE[x] = FAC_CYL_LAM[x]
                                                            (4.2.12.4)
                    Decrement   CTR_RESP_CYC_LAM_CYL_SEL[i]
                                                            (4.2.12.5)
                endif(5)
            elseif(4)   (LV_STATE_DELTA_CRK_WAIT[i] = 1 and
                                        LV_FAC_CYL_LAM_COR_OSC[i] = 0)
            then
                decrement   CTR_RESP_LAM_CYL_SEL[i]
                                                            (4.2.12.6)
```

```
        if$^{(5)}$ CTR_RESP_LAM_CYL_SEL[i] = 0
        then
            if$^{(6)}$ LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0
            then
                increment
                CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] (4.2.12.7)
            endif$^{(6)}$
            CTR_RESP_LAM_CYL_SEL[i] =
                    C_CTR_RESP_MIN_LAM_CYL_SEL (4.2.12.8)
            FAC_CYL_LAM_SAVE[x] = FAC_CYL_LAM[x]
                                                  (4.2.12.9)
            decrement   CTR_RESP_CYC_LAM_CYL_SEL[i]
                                                  (4.2.12.10)
        endif$^{(5)}$
    endif$^{(4)}$
    if$^{(4)}$ (LV_STATE_DELTA_CRK_WAIT[i] = 1 and
                        CTR_RESP_CYC_LAM_CYL_SEL[i] = 0)
    then
        CTR_RESP_CYC_LAM_CYL_SEL[i] =
                C_CTR_RESP_CYC_MIN_LAM_CYL_SEL (4.2.12.11)
        if$^{(5)}$ CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] $\geq$
                        C_CTR_FAC_CYL_LAM_DIF_ABSV_OK
        then
            LV_FAC_CYL_LAM_COR_OSC[i] = 0
                                                  (4.2.12.12)
            LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 0
                                                  (4.2.12.13)
            LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 0
                                                  (4.2.12.14)
            LV_STATE_DELTA_CRK_WAIT[i] = 0
                                                  (4.2.12.15)
            CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] = 0
                                                  (4.2.12.16)
            STATE_DELTA_CRK_CYL_LAM[i] = 6 'AD_HOLD'
                                                  (4.2.12.17)
            CRK_CHG_FAC_PHA_SHIFT_AD[i] = 1
                                                  (4.2.12.18)
            if$^{(6)}$ C_CRK_CYL_LAM_DELTA_INI > 0
            then
                CRK_CYL_LAM_DELTA_INI[i] =
                            CRK_CYL_LAM_DELTA_INI[i] -
                        C_CRK_CYL_LAM_INI_RED (4.2.12.19)
                if$^{(7)}$ CRK_CYL_LAM_DELTA_INI[i] < 0
                then
                    CRK_CYL_LAM_DELTA_INI[i] = 0
                                                  (4.2.12.20)
```

```
            endif$^{(7)}$
        else$^{(6)}$
            CRK_CYL_LAM_DELTA_INI[i] =
                        CRK_CYL_LAM_DELTA_INI[i] +
                        C_CRK_CYL_LAM_INI_RED (4.2.12.21)
            if$^{(7)}$ CRK_CYL_LAM_DELTA_INI[i] > 0
            then
                CRK_CYL_LAM_DELTA_INI[i] = 0
                                                (4.2.12.22)
            endif$^{(7)}$
        endif$^{(6)}$
    elseif$^{(5)}$ [(CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] <
                C_CTR_FAC_CYL_LAM_DIF_ABSV_OK) and
                CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] ≥
                C_CTR_FAC_CYL_LAM_DIF_ABSV_MIN]
    then
        LV_STATE_DELTA_CRK_WAIT[i] = 1
                                                (4.2.12.23)
        CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] = 0
                                                (4.2.12.24)

    elseif$^{(5)}$ (CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] <
                C_CTR_FAC_CYL_LAM_DIF_ABSV_MIN)
    then
        LV_STATE_DELTA_CRK_WAIT[i] = 0
                                                (4.2.12.25)
        CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] = 0
                                                (4.2.12.26)
        if$^{(6)}$ STATE_DELTA_CRK_CYL_LAM[i] = 3 'AD_INI'
        then
            STATE_DELTA_CRK_CYL_LAM[i] = 4 'AD_DEC'
                                                (4.2.12.27)
        elseif$^{(6)}$ STATE_DELTA_CRK_CYL_LAM[i] = 4
                                                'AD_DEC'
        then
            STATE_DELTA_CRK_CYL_LAM[i] = 5 'AD_INC'
                                                (4.2.12.28)
        elseif$^{(6)}$ STATE_DELTA_CRK_CYL_LAM[i] = 5
                                                'AD_INC'
        then
            STATE_DELTA_CRK_CYL_LAM[i] = 4 'AD_DEC'
                                                (4.2.12.29)
        endif$^{(6)}$

    endif$^{(5)}$
endif$^{(4)}$
```

```
elseif(3)   (LC_CRK_DELTA_CQ_SWI = 1 and
                                 LV_ER_ABSV_SUM_GRD_RISE[i] = 0)
then
    if(4)   (LV_STATE_DELTA_CRK_WAIT[i] = 1 and
                                 LV_FAC_CYL_LAM_COR_OSC[i] = 1)
    then
        decrement   CTR_RESP_LAM_CYL_SEL[i]
                                                    (4.2.12.30)
        if(5)   CTR_RESP_LAM_CYL_SEL[i] = 0
        then
            if(6)   DELTA_LAMB_CYL_SEL_CQ_FIL[i] <
                             C_DELTA_LAMB_CYL_SEL_CQ_AD_THD
            then
                increment   CTR_CQ_FIL_FALL[i]
                                                    (4.2.12.31)
            endif(6)
            CTR_RESP_LAM_CYL_SEL[i] =

                     C_CTR_RESP_MIN_LAM_CYL_SEL (4.2.12.32)
            FAC_CYL_LAM_SAVE[x] = FAC_CYL_LAM[x]

                                                    (4.2.12.33)
            Decrement   CTR_RESP_CYC_LAM_CYL_SEL[i]

                                                    (4.2.12.34)
        endif(5)
    endif(4)

    if(4)   (LV_STATE_DELTA_CRK_WAIT[i] = 1 and
                              CTR_RESP_CYC_LAM_CYL_SEL[i] = 0)
    then
        CTR_RESP_CYC_LAM_CYL_SEL[i] =
                 C_CTR_RESP_CYC_MIN_LAM_CYL_SEL (4.2.12.35)
        if(5)   CTR_CQ_FIL_FALL[i] ≥ C_CTR_CQ_FIL_FALL_OK
        then
            LV_FAC_CYL_LAM_COR_OSC[i] = 0
                                                    (4.2.12.36)
            LV_STATE_DELTA_CRK_WAIT[i] = 0
                                                    (4.2.12.37)
            CTR_CQ_FIL_FALL[i] = 0
                                                    (4.2.12.38)
            STATE_DELTA_CRK_CYL_LAM[i] = 6 'AD_HOLD'
                                                    (4.2.12.39)
            CRK_CHG_FAC_PHA_SHIFT_AD[i] = 1
                                                    (4.2.12.40)
            if(6)   C_CRK_CYL_LAM_DELTA_INI > 0
            then
                CRK_CYL_LAM_DELTA_INI[i] =

                                  CRK_CYL_LAM_DELTA_INI[i] -
```

```
                        C_CRK_CYL_LAM_INI_RED (4.2.12.41)
            if(7)   CRK_CYL_LAM_DELTA_INI[i] < 0
            then
                    CRK_CYL_LAM_DELTA_INI[i] = 0
                                                  (4.2.12.42)
            endif(7)
        else(6)
            CRK_CYL_LAM_DELTA_INI[i] =
                            CRK_CYL_LAM_DELTA_INI[i] +
                        C_CRK_CYL_LAM_INI_RED (4.2.12.43)
            if(7)   CRK_CYL_LAM_DELTA_INI[i] > 0
            then
                    CRK_CYL_LAM_DELTA_INI[i] = 0
                                                  (4.2.12.44)
            endif(7)
        endif(6)

    elseif(5)       [(CTR_CQ_FIL_FALL[i] <
                            C_CTR_CQ_FIL_FALL_OK) and
            CTR_CQ_FIL_FALL[i] ≥ C_CTR_CQ_FIL_FALL_MIN]
    then
        LV_STATE_DELTA_CRK_WAIT[i] = 1
                                                  (4.2.12.45)
        CTR_CQ_FIL_FALL[i] = 0
                                                  (4.2.12.46)

    elseif(5)       (CTR_CQ_FIL_FALL[i] <
                            C_CTR_CQ_FIL_FALL_MIN)
    then
        LV_STATE_DELTA_CRK_WAIT[i] = 0
                                                  (4.2.12.47)
        CTR_CQ_FIL_FALL[i] = 0
                                                  (4.2.12.48)
        if(6)   STATE_DELTA_CRK_CYL_LAM[i] = 3 'AD_INI'
        then
            STATE_DELTA_CRK_CYL_LAM[i] = 4 'AD_DEC'
                                                  (4.2.12.49)
        elseif(6)   STATE_DELTA_CRK_CYL_LAM[i] = 4
                                                  'AD_DEC'
        then
            STATE_DELTA_CRK_CYL_LAM[i] = 5 'AD_INC'
                                                  (4.2.12.50)
        elseif(6)   STATE_DELTA_CRK_CYL_LAM[i] = 5
                                                  'AD_INC'
        then
            STATE_DELTA_CRK_CYL_LAM[i] = 4 'AD_DEC'
```

```
                                                          (4.2.12.51)
                    endif(6)

                endif(5)
            endif(4)
        elseif(3)   LV_ER_ABSV_SUM_GRD_RISE[i] = 1
        then
            if(4)   LV_STATE_DELTA_CRK_WAIT[i] = 1
            then
                decrement   CTR_RESP_LAM_CYL_SEL[i]
                                                          (4.2.12.52)
                if(5)   CTR_RESP_LAM_CYL_SEL[i] = 0
                then
                    ER_STND_MMV_DIF_ABSV_SUM_PRE[i] =
                        ER_STND_MMV_DIF_BAL_ABSV_SUM[i]   (4.2.12.53)

                    $ER\_STND\_MMV\_DIF\_BAL\_ABSV\_SUM[i] = \sum_{x=1}^{\alpha} |ER\_STND\_MMV\_DIF\_BAL[x]|$
                    $\alpha$ : nr. of cylinders per bank
                                                          (4.2.12.54)
                    [Remark: the cylinder based variables (index x) shall be
                    assigned to the corresponding exhaust bank (index i)]

                    ER_STND_MMV_DIF_ABSV_SUM_GRD[i] =
                                ER_STND_MMV_DIF_BAL_ABSV_SUM[i] –
                        ER_STND_MMV_DIF_ABSV_SUM_PRE[i]   (4.2.12.55)

                    FAC_CYL_LAM_ABSV_SUM_GRD[i] =
                                    FAC_CYL_LAM_ABSV_SUM[i] –
                            FAC_CYL_LAM_ABSV_SUM_PRE[i]   (4.2.12.56)

                    FAC_CYL_LAM_ABSV_SUM_PRE[i] =
                                FAC_CYL_LAM_ABSV_SUM[i]   (4.2.12.57)

                    if(6)   (ER_STND_MMV_DIF_ABSV_SUM_GRD[i] ≤
                                C_ER_ABSV_GRD_FALL_THD_CRK_CHG and
                                    FAC_CYL_LAM_ABSV_SUM_GRD[i] ≤
                                C_FAC_CYL_LAM_ABSV_GRD_FALL_THD)
                    then
                        LV_ER_ABSV_SUM_GRD_RISE[i] = 0
                                                          (4.2.12.58)
                        LV_STATE_DELTA_CRK_WAIT[i] = 0
                                                          (4.2.12.59)
                        STATE_DELTA_CRK_CYL_LAM[i] = 6 'AD_HOLD'
                                                          (4.2.12.60)
                        CRK_CHG_FAC_PHA_SHIFT_AD[i] = 1
```

```
                                                        (4.2.12.61)
if$^{(7)}$   C_CRK_CYL_LAM_DELTA_INI > 0
then
        CRK_CYL_LAM_DELTA_INI[i] =
                        CRK_CYL_LAM_DELTA_INI[i] -
            C_CRK_CYL_LAM_INI_RED (4.2.12.62)
        if$^{(8)}$   CRK_CYL_LAM_DELTA_INI[i] < 0
        then
                CRK_CYL_LAM_DELTA_INI[i] = 0
                                                        (4.2.12.63)
        endif$^{(8)}$
else$^{(7)}$
        CRK_CYL_LAM_DELTA_INI[i] =
                        CRK_CYL_LAM_DELTA_INI[i] +
            C_CRK_CYL_LAM_INI_RED (4.2.12.64)
        if$^{(8)}$   CRK_CYL_LAM_DELTA_INI[i] > 0
        then
                CRK_CYL_LAM_DELTA_INI[i] = 0
                                                        (4.2.12.65)
        endif$^{(8)}$
endif$^{(7)}$

else$^{(6)}$
        LV_STATE_DELTA_CRK_WAIT[i] = 0
                                                        (4.2.12.66)
        if$^{(7)}$   STATE_DELTA_CRK_CYL_LAM[i] = 3
                                                'AD_INI'
        then
                STATE_DELTA_CRK_CYL_LAM[i] = 4
                                    'AD_DEC' (4.2.12.67)
        elseif$^{(7)}$   STATE_DELTA_CRK_CYL_LAM[i] =
                                            4 'AD_DEC'
        then
                STATE_DELTA_CRK_CYL_LAM[i] = 5
                                    'AD_INC' (4.2.12.68)
        elseif$^{(7)}$   STATE_DELTA_CRK_CYL_LAM[i] =
                                            5  'AD_INC'
        then
                STATE_DELTA_CRK_CYL_LAM[i] = 4
                                    'AD_DEC' (4.2.12.69)
        endif$^{(7)}$
        CTR_RESP_LAM_CYL_SEL[i] =
        C_CTR_ER_RESP_MIN_LAM_CYL_SEL (4.2.12.70)
```

```
                    endif(6)
                endif(5)
            endif(4)
        endif(3)
    endif(2)
endif(1)
```

4.2.13 Control action

4.2.13.1 Initial-action

```
if(1) STATE_LAM_CYL_SEL[i] = 3 [calculation is exhaust bank selective]
then
    if(2) (SEG_NR = 0 [calculation for exhaust bank 1] or
              SEG_NR = C_NR_SEG_REF_CBK_EX_2 [calculation for exhaust bank 2]
    then
        if(3) LV_STATE_DELTA_CRK_WAIT[i] = 0
        then
            if(4) STATE_DELTA_CRK_CYL_LAM[i] = 3 'AD_INI'
            then
                $CRK\_CYL\_LAM\_INT[i]_k = CRK\_CYL\_LAM\_INT[i]_{k-1} + CRK\_CYL\_LAM\_DELTA\_INI[i]$ (4.2.13.1.1)
                LV_STATE_DELTA_CRK_WAIT[i] = 1
                                                                 (4.2.13.1.2)
```

4.2.13.2 Recursive decrement-action

```
            elseif(4) STATE_DELTA_CRK_CYL_LAM[i] = 4 ' AD_DEC'
            then
                if(5) CRK_CYL_LAM_DELTA_INI[i] ≥ 0
                then
                    $CRK\_CYL\_LAM\_INT[i]_k = CRK\_CYL\_LAM\_INT[i]_{k-1} - (CRK\_CHG\_FAC\_PHA\_SHIFT\_AD[i] * C\_CRK\_CHG\_STEP + CRK\_CYL\_LAM\_DELTA\_INI[i])$ (4.2.13.2.1)
                else(5)
                    $CRK\_CYL\_LAM\_INT[i]_k = CRK\_CYL\_LAM\_INT[i]_{k-1} - (CRK\_CYL\_LAM\_DELTA\_INI[i] - CRK\_CHG\_FAC\_PHA\_SHIFT\_AD[i] * C\_CRK\_CHG\_STEP)$ (4.2.13.2.2)
                endif(5)
                CRK_CHG_FAC_PHA_SHIFT_AD[i] =
                        CRK_CHG_FAC_PHA_SHIFT_AD[i] + 1 (4.2.13.2.3)
                LV_STATE_DELTA_CRK_WAIT[i] = 1
                                                                 (4.2.13.2.4)
```

4.2.13.3 Recursive increment-action

```
            elseif(4)   STATE_DELTA_CRK_CYL_LAM[i] = 5 'AD_INC'
            then
                if(5)   CRK_CYL_LAM_DELTA_INI[i] ≥ 0
                then
                        CRK_CYL_LAM_INT[i]$_k$ = CRK_CYL_LAM_INT[i]$_{k-1}$ +
                                            (CRK_CHG_FAC_PHA_SHIFT_AD[i]*
                                                          C_CRK_CHG_STEP +
                                        CRK_CYL_LAM_DELTA_INI[i]) (4.2.13.3.1)
                else(5)
                        CRK_CYL_LAM_INT[i]$_k$ = CRK_CYL_LAM_INT[i]$_{k-1}$ +
                                               (CRK_CYL_LAM_DELTA_INI[i] -
                                             CRK_CHG_FAC_PHA_SHIFT_AD[i]*
                                               C_CRK_CHG_STEP) (4.2.13.3.2)
                endif(5)
                CRK_CHG_FAC_PHA_SHIFT_AD[i] =
                        CRK_CHG_FAC_PHA_SHIFT_AD[i] + 1 (4.2.13.3.3)
                LV_STATE_DELTA_CRK_WAIT[i] = 1
                                                             (4.2.13.3.4)
```

4.2.13.4 Hold-action

```
            elseif(4)   STATE_DELTA_CRK_CYL_LAM[i] = 6 'AD_HOLD'
            then
                CRK_CYL_LAM_INT[i]$_k$ = CRK_CYL_LAM_INT[i]$_{k-1}$
                                                             (4.2.13.4.1)
            endif(4)
        endif(3)
    endif(2)
endif(1)
```

4.2.13.5 Calculation of the adaptation value

For application purpose the calculation of DELTA_CRK_CYL_LAM[i] can be switched off using LC_CRK_CYL_LAM_DELTA_OUT

```
if(1) LC_CRK_CYL_LAM_DELTA_OUT = 1
then
    if(2)   (SEG_NR = 0 [calculation for exhaust bank 1] or
                SEG_NR = C_NR_SEG_REF_CBK_EX_2 [calculation for exhaust bank 2]
    then
            DELTA_CRK_CYL_LAM[i] = MINMAX (CRK_CYL_LAM_INT[i],
                                           C_DELTA_CRK_OUT_MIN,
                                           C_DELTA_CRK_OUT_MAX) (4.2.13.5.1)
    endif(2)
    if(2)   CRK_CYL_LAM_DELTA_INI[i] = 0
    then
            CRK_CYL_LAM_DELTA_INI[i] = C_CRK_CYL_LAM_DELTA_INI
                                                                (4.2.13.5.2)
    endif(2)
else(1)
    DELTA_CRK_CYL_LAM[i] = 0
                                                                (4.2.13.5.3)
    CRK_CYL_LAM_INT[i] = 0
                                                                (4.2.13.5.4)
    CRK_CYL_LAM_DELTA_INI[i] = C_CRK_CYL_LAM_DELTA_INI
                                                                (4.2.13.5.5)
    STATE_DELTA_CRK_CYL_LAM[i] = 0 'AD_PAS'
                                                                (4.2.13.5.6)
    FAC_CYL_LAM_SAVE[x] = 0
                                                                (4.2.13.5.7)
    LV_FAC_CYL_LAM_COR_OSC[i] = 0
                                                                (4.2.13.5.8)
    LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0
                                                                (4.2.13.5.9)
    LV_FAC_CYL_LAM_OSC_MAX_THD[x] = 0
                                                                (4.2.13.5.10)
    LV_FAC_CYL_LAM_OSC_MIN_THD[x] = 0
```

(4.2.13.5.11)

LV_STATE_DELTA_CRK_WAIT[i] = 0

(4.2.13.5.12)

CTR_SUM_FAC_CYL_LAM_COR_OSC[x] = 0

(4.2.13.5.13)

CTR_CQ_MON_PHA_SHIFT_AD[i] = 0

(4.2.13.5.14)

CTR_CQ_MAX_OSC[i] = 0

(4.2.13.5.15)

CTR_RESP_LAM_CYL_SEL[i] = 0

(4.2.13.5.16)

CTR_RESP_CYC_LAM_CYL_SEL[i] = 0

(4.2.13.5.17)

CTR_FAC_CYL_LAM_COR_DIF_ABSV[i] = 0

(4.2.13.5.18)

CTR_CQ_FIL_FALL[i] = 0

(4.2.13.5.19)

CTR_ER_MON_PHA_SHIFT_AD[i] = 0

(4.2.13.5.20)

CRK_CHG_FAC_PHA_SHIFT_AD[i] = 1

(4.2.13.5.21)

LV_DELTA_CRK_CYL_LAM_ERR[i] = 0

(4.2.13.5.22)

LV_ER_ABSV_SUM_GRD_RISE[i] = 0

(4.2.13.5.23)

ER_STND_MMV_DIF_BAL_ABSV_SUM[i] = 0

(4.2.13.5.24)

ER_STND_MMV_DIF_ABSV_SUM_PRE[i] = 0

(4.2.13.5.25)

ER_STND_MMV_DIF_ABSV_SUM_GRD[i] = 0

(4.2.13.5.26)

FAC_CYL_LAM_ABSV_SUM_PRE[i] = 0

(4.2.13.5.27)

FAC_CYL_LAM_ABSV_SUM_GRD[i] = 0

(4.2.13.5.28)

```
    CTR_PHA_SHIFT_AD_TRIG[i] = 0
```

(4.2.13.5.29)

endif(1)

4.2.14 Error detection

```
if$^{(1)}$ (SEG_NR = 0 [calculation for exhaust bank 1] or
            SEG_NR = C_NR_SEG_REF_CBK_EX_2 [calculation for exhaust bank 2]
then
    if$^{(2)}$ (DELTA_CRK_CYL_LAM[i] = C_DELTA_CRK_OUT_MIN or
                DELTA_CRK_CYL_LAM[i] = C_DELTA_CRK_OUT_MAX or
                CRK_CHG_FAC_PHA_SHIFT_AD[i] = C_CRK_CHG_FAC_MAX)
    then
        LV_DELTA_CRK_CYL_LAM_ERR[i] = 1
                                                        (4.2.14.1)
        DELTA_CRK_CYL_LAM[i] = DELTA_CRK_CYL_LAM[i]$_{LDC}$
                                                        (4.2.14.2)
    endif$^{(2)}$
endif$^{(1)}$
```

4.2.15 Storing the adaptation value after power latch

The value of DELTA_CRK_CYL_LAM[i] shall be stored in the non-volatile memory on following condition.

```
if$^{(1)}$ LV_DC 1 $\rightarrow$ 0
then
    if$^{(2)}$ LC_CRK_CYL_LAM_DELTA_OUT = 1
    then
        if$^{(3)}$ (LV_DELTA_CRK_CYL_LAM_ERR[i] = 0 and
                    STATE_DELTA_CRK_CYL_LAM[i] = 6 'AD_HOLD')
        then
            DELTA_CRK_CYL_LAM[i] = DELTA_CRK_CYL_LAM[i]$_{CDC}$
                                                        (4.2.15.1)
        else$^{(3)}$
            DELTA_CRK_CYL_LAM[i] = DELTA_CRK_CYL_LAM[i]$_{LDC}$
                                                        (4.2.15.2)
        endif$^{(3)}$
    else$^{(2)}$
        DELTA_CRK_CYL_LAM[i] = 0
                                                        (4.2.15.3)
    endif$^{(2)}$
```

CDC: Current Driving Cycle

LDC: Last Driving Cycle

4.2.16 Counter update in case of CILC stable

```
    if(2)   (LV_FAC_CYL_LAM_COR_OSC[i] = 0 and
                                LV_FAC_CYL_LAM_COR_LIM_OSC[i] = 0 and
                                  LV_ER_ABSV_SUM_GRD_RISE[i] = 0 and
                             CTR_LAMB_CYL_SEL_CQ_DRIFT_THD[i] = 0 and
                               STATE_LAMB_CYL_SEL_CQ_SLOP[i] = 0 and
                                       CTR_PHA_SHIFT_AD_TRIG[i] > 0)
    then
            decrement (CTR_PHA_SHIFT_AD_TRIG[i])
                                                            (4.2.16.1)

    endif(2)
  endif(1)
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_DELTA_CRK_OUT_MIN	1	88...78H	-720...720	6	[°CRK]
Lower limit of the adapted phase displacement					
C_DELTA_CRK_OUT_MAX	1	88...78H	-720...720	6	[°CRK]
Upper limit of the adapted phase displacement					
C_CRK_CYL_LAM_DELTA_INI	1	88...78H	-720...720	6	[°CRK]
CRK - initial constant					
C_CTR_SUM_FAC_CYL_LAM_OSC_MAX	1	0...FFFFH	0...65535	1	[-]
Maximum time of oscillation monitoring - FAC_CYL_LAM_COR - based					
C_CTR_CQ_MIN_PHA_SHIFT_AD	1	0...FFFFH	0...65535	1	[-]
Minimum time of exceeded CQ limit to request the adaptation					
C_CTR_CQ_MON_PHA_SHIFT_AD_MAX	1	0...FFFFH	0...65535	1	[-]
Maximum time of oscillation monitoring - CQ - based					
C_CTR_ER_MON_PHA_SHIFT_AD_MAX	1	0...FFFFH	0...65535	1	[-]
Maximum time for engine roughness monitoring					
C_CTR_RESP_MIN_LAM_CYL_SEL	1	0...FFFFH	0...65535	1	[-]
Minimum response time for adaptation result					
C_CTR_ER_RESP_MIN_LAM_CYL_SEL	1	0...FFFFH	0...65535	1	[-]
Minimum response time of engine roughness for evaluation of adaptation result					
C_FAC_CYL_LAM_MAX_PHA_SHIFT_AD	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper limit of FAC_CYL_LAM_COR value to detect the oscillation					
C_FAC_CYL_LAM_MIN_PHA_SHIFT_AD	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lower limit of FAC_CYL_LAM_COR value to detect the oscillation					
C_CRK_CHG_FAC_MAX	1	1...78H	1...120	1	[-]
Upper limit of the CRK change factor					
C_CRK_CYL_LAM_INI_RED	1	0...78H	0...720	6	[°CRK]
CRK - initial reduction factor					
C_CRK_CHG_STEP	1	88...78H	-720...720	6	[°CRK]
CRK change step					
C_CTR_RESP_CYC_MIN_LAM_CYL_SEL	1	0...FFH	0...255	1	[-]
Minimum number of response cycles to evaluate the adaptation result					
C_CTR_FAC_CYL_LAM_DIF_ABSV_MIN	1	0...FFH	0...255	1	[-]
Least number of improvement cycles to extend the waiting time for evaluation of adaptation result - FAC_CYL_LAM_COR - based					
C_CTR_FAC_CYL_LAM_DIF_ABSV_OK	1	0...FFH	0...255	1	[-]
Minimum number of improved cycles to accept the adaptation value - FAC_CYL_LAM_COR - based					
C_CTR_CQ_FIL_FALL_MIN	1	0...FFH	0...255	1	[-]
Least number of improvement cycles to extend the waiting time for evaluation of adaptation result - CQ - based					
C_CTR_CQ_FIL_FALL_OK	1	0...FFH	0...255	1	[-]
Minimum number of improved cycles to accept the adaptation value - CQ - based					
C_CTR_PHA_SHIFT_AD_TRIG_THD	1	0...FFH	0...255	1	[-]
Trigger threshold for delayed phase shift adaptation					
C_FAC_CYL_LAM_DIF_ABSV_MAX	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Maximum of allowed FAC_CYL_LAM_COR oscillation space					
C_ER_ABSV_GRD_RISE_THD_CRK_CHG	1	0...7FFFH	0...325.77	9.942e-3	[1/s²]
Maximum limit of engine roughness gradient rise					
C_ER_ABSV_GRD_FALL_THD_CRK_CHG	1	8000...7FFFH	-325.78...325.77	9.942e-3	[1/s²]
Threshold of engine roughness gradient drop for a safe adaptation result					
C_FAC_CYL_LAM_ABSV_GRD_RISE_THD	1	0...7FFFH	0...49.99847	1.5259e-3	[%]
Maximum limit of CILC gradient rise					
C_FAC_CYL_LAM_ABSV_GRD_FALL_THD	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Threshold of CILC gradient drop for a safe adaptation result					
C_DELTA_LAMB_CYL_SEL_CQ_AD_THD	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Adaptation-release threshold					

C_NR_SEG_REF_CBK_EX_2	1	0...7H	0...7	1	[-]
Segment number for identification of the first fired cylinder of exhaust bank 2					
LC_CRK_CYL_LAM_DELTA_OUT	1	0...1H	0...1	1	[-]
Adaptation switch					
LC_CRK_DELTA_CQ_SWI	1	0...1H	0...1	1	[-]
Adaptation via characteristic quantity					

4.3 Application Incidences for the Cylinder Individual Lambda Control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAM_CYL_ENA[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Activation of the cylinder individual lambda control is open					
LV_MFF_N_CDN_LAM_CYL_ENA[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Fuel mass- and engine speed condition for the activation of the cylinder individual lambda control fulfilled					
LV_LAM_CYL_ENA_CYL_BAL_DC[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Specific activation conditions for cylinder individual lambda control are fulfilled in DC (CYBL coordination)					
LV_LAM_CYL_SEL_CTL_FAST_REQ	O/V	0...1H	0...1	1	[-]
Fast cylinder individual lambda control requested					
LV_LAM_CYL_SEL_ADJ_OFS_REQ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Cylinder individual fuel mass offset correction requested					
LV_LS_CBK_LAM_CYL_SEL_CONF_VLD	V	0...1H	0...1	1	[-]
Configuration pattern for allocation of physical cylinders to exhaust bank is valid (=1)					
LF_LS_CBK_EX_LAM_CYL_SEL_CONF	O/V	0...FFH	0...255	1	[-]
Logical variables data field for allocation of physical cylinders to exhaust bank					
PHA_SHIFT_CAM_EX[NC_CBK_EX_NR]	O/V	FFFFFF88...78H	-720...720	6	[°CRK]
variable phase shift (variable valve timing)					
IGA_DIF_BAS_CRK_CYL_LAM	V	F4C0...B40H	-180...180	0.0625	[°CRK]
Difference between basic ignition angle stored during calibration of IP_CRK_SYN and actual value					
MFF_SP_LAM_CYL_SEL	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint used by cylinder individual lambda control					
TEMP_LAM_CYL_SEL	O/V	0...FEH	-48...142.5	0.75	[°C]
Temperature relevant for the operation of cylinder individual lambda control					
STATE_CDN_LAM_CYL_SEL_ADJ_RNG[NC_CBK_EX_NR]	O/V	0H 1H 2H 3H	OUT_OF_RNG LOW_RNG HIGH_RNG COLD_RNG	1	[-]
Condition- state of cylinder individual lambda adaptation					

Input data:

N	LV_ERR_IVVT	NC_CBK_EX_NR	LV_CBK_MPL
NC_CYL_NR	LV_ERR_CHG_LS_UP	AMP	LV_ERR_IGC
LV_ERR_CAM	NC_LAMB_REF	LV_IND_FCUT	LV_ERR_CPS
CL_MMV	LV_ERR_TPS	LV_ERR_MAF	C_CAM_OP_EX
LV_ERR_CRK	LV_MIS_STATE_A	LV_MIS_STATE_B	NC_NR_CBK_IVVT
N_32	TCO	LV_ERR_DIAGCPS	IGA_IGC_0_5_H_RNG
LV_VAR_LSH_UP	LV_VAR_TCHA	LV_ERR_WG_DR[NC_NR_TCHA]	LV_ERR_IV[NC_CYL_NR]
MFF_SP_HOM_BAS_MV	NC_NR_CAM_CBK	MFF_SP_S_SWI_HOM	LV_HOM_ACT
TEMP_CAPA_IV_MV	TFU_IV	NC_MAF_NR	LV_INH_BAL_CUS
LV_ERR_FUP_MFP_PLAUS	LV_ERR_VCV[NC_CBK_HPP_NR]	LV_ERR_FUP	LV_ERR_H_PRS_SYS

LV_LAM_LSCL[NC_CBK_EX_NR]	C_MFF_LAM_CYL_ADJ_CST_MAX	LV_IPLSL_VLD[NC_CBK_EX_NR]
LV_TI_CYL_BAL_LAM_ACT	LV_LAMB_COP[NC_CBK_EX_NR]	LV_VPLSL_LIM[NC_CBK_EX_NR]
CAM_EX[NC_NR_CBK_IVVT]	LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_LS_UP[NC_CBK_EX_NR]
C_TEMP_LAM_CYL_MIN	C_TEMP_LAM_CYL_CST_MIN	C_TEMP_LAM_CYL_CST_MAX

C_N_LAM_CYL_ADJ_CST_MIN	C_N_LAM_CYL_ADJ_CST_MAX	C_MFF_LAM_CYL_ADJ_CST_MIN
C_N_LAM_CYL_ADJ_CST_MIN	C_N_LAM_CYL_ADJ_CST_MAX	C_MFF_LAM_CYL_ADJ_CST_MIN
C_MFF_LAM_CYL_ADJ_CST_MAX	LV_ERR_TPS_ST_CHK_2	LV_ERR_TPS_ST_CHK_1
LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_MAP_DIP_SHIFT
LV_ERR_MAP_PLAUS	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_FTL_MIN
LV_ERR_MAP_TPS_PLAUS	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]
LV_CL_MMV_CAL_ACT	LV_ERR_TCHA_LEAK	LV_ERR_TCHA_PRS_CTL
LV_ERR_TCHA_PRS_DIF	LV_ERR_TCHA_PRS_HIGH	LV_ERR_TCHA_PRS_LOW

TEG_CAT_UP_MDL[NC_CBK_EX_NR]	FAC_MV_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]
LV_LAM_CYL_SEL_LDC[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MIN[NC_CBK_EX_NR]
LV_FAC_LAM_LIM_MAX[NC_CBK_EX_NR]	LV_ERR_CAT_DIAG[NC_CBK_EX_NR]
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL
WGPWM[NC_CBK_EX_NR]	LV_LAM_CYL_SEL_ADJ_OFS_EXT_REQ
C_N_MIN_FAC_CYL_SEL_ADJ_L_RNG	C_N_MAX_FAC_CYL_SEL_ADJ_L_RNG
C_MFF_MIN_FAC_CYL_SEL_ADJ_L_RNG	C_MFF_MAX_FAC_CYL_SEL_ADJ_L_RNG
C_N_MIN_FAC_CYL_SEL_ADJ_H_RNG	C_N_MAX_FAC_CYL_SEL_ADJ_H_RNG
C_MFF_MIN_FAC_CYL_SEL_ADJ_H_RNG	C_MFF_MAX_FAC_CYL_SEL_ADJ_H_RNG
LV_ERR_MAF_FRQ_RNG[NC_MAF_NR]	LV_ERR_MAF_FRQ_GRD[NC_MAF_NR]
LV_LAM_CYL_SEL_CTL_DRIFT_PRED[NC_CBK_EX_NR]	LV_INH_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]
LV_ERR_TOOTH_OFF_EX[NC_NR_CAM_CBK]	LV_ERR_TOOTH_OFF_IN[NC_NR_CAM_CBK]
LV_ERR_CYL_BAL_LAM[NC_CYL_NR]	LV_ERR_MAF_FRQ_EL[NC_MAF_NR]
LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]

LV_LAM_AD_INJ_ACT	LAMB_SP_HOM[NC_CBK_EX_NR]		

FUNCTION DESCRIPTION:

General information:

The module serves as a check box for application conditions of the function “cylinder individual lambda control”. If several conditions are fulfilled the flag LV_LAM_CYL_ENA is set to 1. This flag is an input to the cylinder individual lambda control.

Description of index "[i]" and "[x]":

x: logical cylinder
*use **NC_CYL_NR**: number of engine cylinders – index "[x]"*
*use **LF_LS_CBK_EX_LAM_CYL_SEL_CONF**: pattern for allocation of physical cylinders to exhaust bank – index "[i]"*

Application conditions:

Initialisation (at reset):

```
if(1)  (CLF_LS_CBK_EX_LAM_CYL_SEL_CONF = 0 or
                          CLF_LS_CBK_EX_LAM_CYL_SEL_CONF = NC_LAMB_REF)
then
    LF_LS_CBK_EX_LAM_CYL_SEL_CONF = CLF_LS_CBK_EX_LAM_CYL_SEL_CONF
    LV_LS_CBK_LAM_CYL_SEL_CONF_VLD = 1
```

```
else(1)
    LV_LS_CBK_LAM_CYL_SEL_CONF_VLD = 0
endif(1)

if(1) LV_HOM_ACT = 1
then
    MFF_SP_LAM_CYL_SEL = MFF_SP_HOM_BAS_MV
else(1)
    MFF_SP_LAM_CYL_SEL = MFF_SP_S_SWI_HOM
endif(1)

if(1) C_TEMP_IV_SWI_LAM_CYL_SEL = 1
then
    TEMP_LAM_CYL_SEL = TFU_IV
else(1)
    if(2)   C_TEMP_IV_SWI_LAM_CYL_SEL = 2
    then
            TEMP_LAM_CYL_SEL = TEMP_CAPA_IV_MV
    else(2)
            TEMP_LAM_CYL_SEL = TCO
    endif(2)
endif(1)

LV_MFF_N_CDN_LAM_CYL_ENA[i] = 0
LV_LAM_CYL_SEL_CTL_FAST_REQ = 0
LV_LAM_CYL_ENA_CYL_BAL_DC[i] = 0
PHA_SHIFT_CAM_EX[i] = 0
LV_LAM_CYL_ENA[i] = 0
STATE_CDN_LAM_CYL_SEL_ADJ_RNG[i] = 0 [OUT_OF_RNG]
IGA_DIF_BAS_CRK_CYL_LAM = 0
LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 0
```

Recurrence:

every TDC

Activation:

-

Deactivation:

-

Formula section:

```
if(1) LV_HOM_ACT = 1
then
    MFF_SP_LAM_CYL_SEL = MFF_SP_HOM_BAS_MV
else(1)
```

```
        MFF_SP_LAM_CYL_SEL = MFF_SP_S_SWI_HOM
endif(1)

if(1) C_TEMP_IV_SWI_LAM_CYL_SEL = 1
then
        TEMP_LAM_CYL_SEL = TFU_IV
else(1)
        if(2)   C_TEMP_IV_SWI_LAM_CYL_SEL = 2
        then
                TEMP_LAM_CYL_SEL = TEMP_CAPA_IV_MV
        else(2)
                TEMP_LAM_CYL_SEL = TCO
        endif(2)
endif(1)

if(1) LV_LS_CBK_LAM_CYL_SEL_CONF_VLD = 1
then
        if(2)   {(LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 0 or
                        LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 0) and LV_VAR_TCHA = 1}
        then
                LV_LAM_CYL_SEL_CTL_FAST_REQ = 1
                if(3)   ((IP_MFF_LAM_CYL_MIN_EOL < MFF_SP_LAM_CYL_SEL <
                                                        IP_MFF_LAM_CYL_MAX_EOL) and
                                    (C_N_LAM_CYL_MIN_EOL < N < C_N_LAM_CYL_MAX_EOL))
                then
                        LV_MFF_N_CDN_LAM_CYL_ENA[i] = 1
                else(3)
                        LV_MFF_N_CDN_LAM_CYL_ENA[i] = 0
                endif(3)
        else(2)
                if(3)   LC_LAM_CYL_SEL_CTL_FAST_REQ_MAN = 1
                then
                        LV_LAM_CYL_SEL_CTL_FAST_REQ = 1
                else(3)
                        LV_LAM_CYL_SEL_CTL_FAST_REQ = 0
                endif(3)
                if(3)   ((IP_MFF_LAM_CYL_MIN_NOM < MFF_SP_LAM_CYL_SEL <
                                                        IP_MFF_LAM_CYL_MAX_NOM) and
                                    (C_N_LAM_CYL_MIN_NOM < N < C_N_LAM_CYL_MAX_NOM))
                then
                        LV_MFF_N_CDN_LAM_CYL_ENA[i] = 1
                else(3)
                        LV_MFF_N_CDN_LAM_CYL_ENA[i] = 0
                endif(3)
        endif(2)
```

***Remark*:**

1) Variables dependent on NC_CYL_NR shall be assigned to the corresponding exhaust bank (index i).

2) Every variable elements dependent on NC_NR_CAM_CBK or NC_MAF_NR shall be assigned to all exhaust bank.

if(2) **(**AMP > C_AMP_LAM_CYL_MIN
[*the ambient pressure is over the threshold*] **and**
LV_MFF_N_CDN_LAM_CYL_ENA[i] = 1
[*the fuel mass/engine speed is inside of a permitted range*] **and**
LV_ERR_LS_UP[i] = 0 [*no error on upstream O2 sensor signal*] **and**
FAC_MV_DIAG_DYN_LSL_UP[i] < C_FAC_MV_LS_UP_DYN_LAM_CYL_THD
[*The mean value of normalized single sensor signal amplitude is less than the threshold value for cylinder individual lambda control*] **and**
LV_ERR_OFS_LSL_UP[i] = 0
[Offset correction error of the WRAF sensor controller] **and**
LV_ERR_CTL_LSL_UP[i] = 0 [pump current controller diagnosis] **and**
LV_ERR_IVVT = 0 [*The IVVT limp-home-bit not set*] **and**
LV_IND_FCUT = 0 [*no cylinder shut off*] **and**
LV_ERR_CAT_DIAG[i] = 0 [*no failure after filtering diagnosis value - catalyst diagnosis*] **and**
(LV_CL_MMV_CAL_ACT = 0 **or** CL_MMV ≤ C_CL_MMV_LAM_CYL_MAX)
[if *canister load is calculation is active, it did not exceed the inhibit threshold*] **and**
LV_ERR_CHG_LS_UP = 0 [*upstream sensors of banks 1and 2 are not interchanged*] **and**
LV_ERR_MAF = 0 [*no MAF error detected*] **and**
LV_ERR_IV[x] = 0 [*no error on injection valve*]
LV_ERR_IGC = 0 [*no error on ignition output driver*] **and**
LV_ERR_TPS = 0 [*no TPS error detected*] **and**
LV_ERR_CPS = 0 [*no CPS error detected*] **and**
LV_ERR_DIAGCPS = 0 [*no functional error of CPS*] **and**
LV_ERR_FSD[i] = 0 [*no error detected by fuel system monitoring*] **and**
LV_ERR_MAP_DIP_SENS = 0
[no intake manifold pressure difference sensor failure] **and**
LV_ERR_MAP_DIP_PLAUS = 0
[no intake manifold pressure difference plausibility failure] **and**
LV_ERR_MAP_DIP_SHIFT = 0
[no intake manifold pressure difference signal shift failure] **and**
LV_ERR_MAP_TPS_PLAUS = 0 [no MAP ratio plausibility error] **and**
LV_ERR_TPS_ST_CHK_1 = 0 [no TPS error at spring test] **and**
LV_ERR_TPS_ST_CHK_2 = 0 [no TPS error at spring test] **and**
LV_ERR_MAP_PLAUS = 0 [no MAP plausibility error] **and**
LV_ERR_LOAD_TPS_PLAUS = 0 [no error at load pressure compare] **and**
LV_ERR_FTL_MIN = 0 [no fuel tank level min error] **and**
LV_ERR_CYL_BAL_LAM[x] = 0 [no TPS error at spring test] **and**
LV_ERR_WG_DR[i] = 0 [no waste gate error] **and**
LV_ERR_TTIP_MES_LSH_UP[i] = 0
[no plausibility error of LSH tip temperature] **and**
LV_ERR_FSD_LAM_LIM[i] = 0
[no fuel system failure for lambda controller monitoring] **and**
LV_ERR_MAF_FRQ_EL[NC_MAF_NR] = 0
[no electrical error of the MAF sensor] **and**

```
        LV_ERR_MAF_FRQ_GRD[NC_MAF_NR] = 0
        [no error at MAF frequency gradient check] and
        LV_ERR_MAF_FRQ_RNG[NC_MAF_NR] = 0
        [no error at MAF frequency range check] and
        LV_ERR_FUP = 0 [no fuel pressure pump error] and
        LV_ERR_FUP_MFP_PLAUS = 0
        [no error of FUP to mass fuel pump relation] and
        LV_ERR_H_PRS_SYS = 0 [no error of abnormal fuel pressure] and
        LV_ERR_TOOTH_OFF_EX[NC_NR_CAM_CBK] = 0
        [no one-tooth-off camshaft error] and
        LV_ERR_TOOTH_OFF_IN[NC_NR_CAM_CBK] = 0
        [no one-tooth-off camshaft error] and
        LV_ERR_VCV = 0 [no supply voltage error] and
        LV_ERR_CAM = 0 [no error on all camshaft sensor signals] and
        LV_ERR_CRK = 0 [no crankshaft sensor failure] and
        LV_VAR_LSH_UP = 1
        [variant of CAT/exhaust type with upstream lambda sensors recognized] and
        LV_INH_BAL_CUS = 0 [customer request to inhibit the function] and
        LV_IPLSL_VLD[i] = 1 [WRAF sensor pump current is valid] and
        (LV_ERR_TCHA_LEAK = 0 and LV_ERR_TCHA_PRS_CTL = 0 and
        LV_ERR_TCHA_PRS_DIF = 0 and LV_ERR_TCHA_PRS_HIGH = 0 and
        LV_ERR_TCHA_PRS_LOW = 0) [no turbo charger errors]) and
        LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 0)and
        LAMB_SP_HOM[i] > C_LAMB_SP_HOM_LAM_CYL_MIN_NOM
        [Minimum Lambda SP for activation]

then
        LV_LAM_CYL_ENA_CYL_BAL_DC[i] = 1
else(2)
        LV_LAM_CYL_ENA_CYL_BAL_DC[i] = 0
endif(2)

if(2)   (LV_LAM_CYL_ENA_CYL_BAL_DC[i] = 1 and
        LV_LAM_CYL_SEL_LDC[i] = 1 [limited dynamics has been recognized] and
        LV_LAMB_COP[i] = 0 [catalyst overheating prevention not active] and
        TEG_CAT_UP_MDL[i] ≤ C_TEG_CAT_UP_MDL_LAM_CYL_MAX
        [exhaust gas temperature didn't exceed the inhibit threshold] and
        LV_VPLSL_LIM[i] = 0 [pump cell voltage is not at the limit] and
        LV_MIS_STATE_A = 0 [no misfire carb A detected] and
        LV_MIS_STATE_B = 0 [no misfire carb B detected] and
        (LV_FAC_LAM_LIM_MIN[i] = 0 and LV_FAC_LAM_LIM_MAX[i] = 0)
        [output of the lambda control is not limited]) and
        LV_LAM_AD_INJ_ACT = 0 [minimal fuel mass adaptation not active]
then
        if(3)   TEMP_LAM_CYL_SEL ≥ C_TEMP_LAM_CYL_MIN
        then
                if(4)   (N_32 ≥ C_N_MIN_FAC_CYL_SEL_ADJ_L_RNG and
                        N_32 ≤ C_N_MAX_FAC_CYL_SEL_ADJ_L_RNG and
                        MFF_SP_LAM_CYL_SEL ≥
```

```
                                        C_MFF_MIN_FAC_CYL_SEL_ADJ_L_RNG and
                                        MFF_SP_LAM_CYL_SEL ≤
                                        C_MFF_MAX_FAC_CYL_SEL_ADJ_L_RNG)
          then
                STATE_CDN_LAM_CYL_SEL_ADJ_RNG[i] = 1
                                        [LOW_RNG]
          else$^{(4)}$
                if$^{(5)}$   (N_32 ≥ C_N_MIN_FAC_CYL_SEL_ADJ_H_RNG and
                                        N_32 ≤
                                        C_N_MAX_FAC_CYL_SEL_ADJ_H_RNG and
                                        MFF_SP_LAM_CYL_SEL ≥
                                        C_MFF_MIN_FAC_CYL_SEL_ADJ_H_RNG and
                                        MFF_SP_LAM_CYL_SEL ≤
                                        C_MFF_MAX_FAC_CYL_SEL_ADJ_H_RNG)
                then
                      STATE_CDN_LAM_CYL_SEL_ADJ_RNG[i] = 2
                                        [HIGH_RNG]
                else$^{(5)}$
                      STATE_CDN_LAM_CYL_SEL_ADJ_RNG[i] = 0
                                        [OUT_OF_RNG]
                endif$^{(5)}$
          endif$^{(4)}$
    else$^{(3)}$
          if$^{(4)}$   (C_TEMP_LAM_CYL_CST_MIN <
                   TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_CST_MAX and
                                        TEMP_LAM_CYL_SEL < C_TEMP_LAM_CYL_MIN
                                        and LV_LAM_LSCL[i] = 1)
          then
                if$^{(5)}$   ((C_MFF_LAM_CYL_ADJ_CST_MIN <
                                        MFF_SP_LAM_CYL_SEL <
                                        C_MFF_LAM_CYL_ADJ_CST_MAX) and
                                        (C_N_LAM_CYL_ADJ_CST_MIN ≤ N_32 ≤
                                        C_N_LAM_CYL_ADJ_CST_MAX))
                then
                      STATE_CDN_LAM_CYL_SEL_ADJ_RNG[i] = 3
                                        [COLD_RNG]
                else$^{(5)}$
                      STATE_CDN_LAM_CYL_SEL_ADJ_RNG[i] = 0
                                        [OUT_OF_RNG]
                endif$^{(5)}$
          else$^{(4)}$
                STATE_CDN_LAM_CYL_SEL_ADJ_RNG[i] = 0 [OUT_OF_RNG]
          endif$^{(4)}$
    endif$^{(3)}$

    if$^{(3)}$   (LV_TI_CYL_BAL_LAM_ACT = 1
                                        [adaptation via lambda sensor is released] and
                                        WGPWM[i] ≤ C_WGPWM_MAX_LAM_CYL_SEL
                                        [waste gate – PWM signal did not exceed the threshold ])
```

```
        then
            LV_LAM_CYL_ENA[i] = 1
        else(3)
            LV_LAM_CYL_ENA[i] = 0
        endif(3)
    else(2)
        if(3)   LC_LAM_CYL_ENA_CDN_MAN = 1
        then
            LV_LAM_CYL_ENA[i] = 1
        else(3)
            LV_LAM_CYL_ENA[i] = 0
        endif(3)
        STATE_CDN_LAM_CYL_SEL_ADJ_RNG[i] = 0 [OUT_OF_RNG]
    endif(2)
else(1)
    LV_LAM_CYL_ENA[i] = 0
endif(1)

if(1) LF_LS_CBK_EX_LAM_CYL_SEL_CONF = 0
then
    STATE_CDN_LAM_CYL_SEL_ADJ_RNG[2] =
                                    STATE_CDN_LAM_CYL_SEL_ADJ_RNG[1]
    LV_LAM_CYL_ENA_CYL_BAL_DC[2] = LV_LAM_CYL_ENA_CYL_BAL_DC[1]
endif(1)
```

Request for cylinder individual fuel mass offset correction

```
if(1) LV_LAM_CYL_ENA[i] = 1
then
    if(2)   LV_LAM_CYL_SEL_ADJ_OFS_EXT_REQ = 1
    then
        LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 1
    else(2)
        if(3)   LC_LAM_CYL_SEL_ADJ_OFS_RNG_REQ = 1
        then
            if(4)   (N ≤ C_N_MAX_LAM_CYL_SEL_ADJ_ADD and
                                            MFF_SP_LAM_CYL_SEL ≤
                                  C_MFF_MAX_LAM_CYL_SEL_ADJ_ADD)
            then
                LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 1
            else(4)
                LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 0
            endif(4)
        else(3)
            LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 0
```

```
            endif(3)
        endif(2)
else(1)
        LV_LAM_CYL_SEL_ADJ_OFS_REQ[i] = 0
endif(1)
```

Calibration of variable phase shift

Regarding the influence of variable position of exhaust camshaft on phase shift, a variable phase shift, PHA_SHIFT_CAM_EX[i] is defined.
For a six-cylinder engine with a variable valve timing for instance, the variable phase shift is governed by

```
if(1) LV_LAM_CYL_ENA[i] = 1
then
        IGA_DIF_BAS_CRK_CYL_LAM = IP_IGA_BAS_CRK_CYL_LAM –
                                                        IGA_IGC_0_5_H_RNG
        if(2)   LV_CBK_MPL = 0
        then
                PHA_SHIFT_CAM_EX[i] = 360 °CRK + CAM_EX[1] – C_CAM_OP_EX/2
                        + IP_CRK_DELTA_IGA_LAM_CYL_SEL(IGA_DIF_BAS_CRK_CYL_LAM)
        else(2)
                PHA_SHIFT_CAM_EX[i] = 360 °CRK + CAM_EX[i] – C_CAM_OP_EX/2
                        + IP_CRK_DELTA_IGA_LAM_CYL_SEL(IGA_DIF_BAS_CRK_CYL_LAM)
        endif(2)
endif(1)
```

Calibration remark for CLF_LS_CBK_EX_LAM_CYL_SEL_CONF:

Default value:

CLF_LS_CBK_EX_LAM_CYL_SEL_CONF = NC_LAMB_REF

CLF_LS_CBK_EX_LAM_CYL_SEL_CONF	Description	Value - Binary	Value – Decimal
Configuration pattern for allocation of physical cylinders to exhaust bank (bit-wise calibrated)	*single-exhaust-bank*	00000000	0
	4-cylinder-engine 2-exhaust-bank	00001010	10
	6-cylinder-engine 2-exhaust-bank	00101010	42
	8-cylinder-engine 2-exhaust-bank	10101010	170

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_AMP_LAM_CYL_MIN	1	0...FFFFH	0...5434	0.0829175	[hPa]
Minimum ambient pressure for the activation					
IP_MFF_LAM_CYL_MIN_EOL	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_N_3_CYBL	6	0...1FE0H	0...8160	1	[rpm]
Minimum fuel mass/engine speed threshold for the activation of cylinder individual lambda at EOL					
IP_MFF_LAM_CYL_MAX_EOL	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_N_3_CYBL	6	0...1FE0H	0...8160	1	[rpm]
Maximum fuel mass/engine speed threshold for the activation of cylinder individual lambda at EOL					
IP_MFF_LAM_CYL_MIN_NOM	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_N_4_CYBL	6	0...1FE0H	0...8160	1	[rpm]
Minimum fuel mass/engine speed threshold for the activation of cylinder individual lambda – nominal					
IP_MFF_LAM_CYL_MAX_NOM	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_N_4_CYBL	6	0...1FE0H	0...8160	1	[rpm]
Maximum fuel mass/engine speed threshold for the activation of cylinder individual lambda – nominal					
IP_CRK_DELTA_IGA_LAM_CYL_SEL	8	0...F0H	-720...720	6	[°CRK]
LDP_IGA_DIF_BAS_CYL_LAM_IP_CRK	8	0...1680H	-180...180	0.0625	[°CRK]
Phase shift deviation caused by IGA difference					
IP_IGA_BAS_CRK_CYL_LAM	8*8	0...B40H	-90...90	0.0625	[°CRK]
LDPM_N_1_CYBL	8	0...1FE0H	0...8160	1	[rpm]
LDPM_MAF_1_CYBL	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Basic ignition angle stored during calibration of IP_CRK_SYN					
C_WGPWM_MAX_LAM_CYL_SEL	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Maximum threshold of waste gate signal for activation of cylinder individual lambda control					
C_TEG_CAT_UP_MDL_LAM_CYL_MAX	1	0...7FF0H	0...2047	0.0625	[°C]
Maximum exhaust gas temperature; engine out - alternative value for sensor hexagon temperature					
C_FAC_MV_LS_UP_DYN_LAM_CYL_THD	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
CILC-specific threshold of limit-sensor					
C_CL_MMV_LAM_CYL_MAX	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum canister load					
LC_LAM_CYL_ENA_CDN_MAN	1	0...1H	0...1	1	[-]
Manual switch to enable the cylinder individual lambda control ignoring the conditions are fulfilled					
LC_LAM_CYL_SEL_CTL_FAST_REQ_MAN	1	0...1H	0...1	1	[-]
Manual switch for a fast cylinder individual lambda control (=1)					
LC_LAM_CYL_SEL_ADJ_OFS_RNG_REQ	1	0...1H	0...1	1	[-]
Manual request for calculation of cylinder individual lambda adaptation in the additive area (=1)					
C_TEMP_IV_SWI_LAM_CYL_SEL	1	0...2H	0...2	1	[-]
Manual switch for the injection-valve-temperature based cylinder individual lambda control					
CLF_LS_CBK_EX_LAM_CYL_SEL_CONF	1	0...FFH	0...255	1	[-]
Configuration pattern for allocation of physical cylinders to exhaust bank					
C_N_LAM_CYL_MIN_NOM	1	0...1FE0H	0...8160	1	[rpm]
Minimum engine speed for the activation – nominal					
C_N_LAM_CYL_MAX_NOM	1	0...1FE0H	0...8160	1	[rpm]
Maximum engine speed for the activation – nominal					
C_N_LAM_CYL_MIN_EOL	1	0...1FE0H	0...8160	1	[rpm]
Minimum engine speed for the activation at EOL					
C_N_LAM_CYL_MAX_EOL	1	0...1FE0H	0...8160	1	[rpm]
Maximum engine speed for the activation at EOL					
C_N_MAX_LAM_CYL_SEL_ADJ_ADD	1	0...1FE0H	0...8160	1	[rpm]
Upper engine speed threshold for calculation of cylinder individual lambda adaptation in the additive area					
C_MFF_MAX_LAM_CYL_SEL_ADJ_ADD	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Upper fuel mass threshold for calculation of cylinder individual lambda adaptation in the additive area					
C_LAMB_SP_HOM_LAM_CYL_MIN_NOM	1	0...7FFFH	0...1.99993	0.061e-3	[-]
Minimum Lambda SP for activation - nominal					

4.4 Detection of Limited Dynamics for Cylinder Individual Lambda Control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAM_CYL_SEL_LDC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Detection flag of limited dynamics for cylinder individual lambda control					
FAC_LAM_MV_OFS_LDC_LAM_CYL_SEL[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Offset of lambda controller output mean value for detection of limited dynamics (cylinder individual lambda control)					
MAF_INT_LDC_LAM_CYL_SEL[NC_CBK_EX_NR]	V	0...FFFFH	0...1456.33333	0.0222222	[g]
Air mass flow integral during the time after limited dynamics was interrupted					
MAF_OFS_LDC_LAM_CYL_SEL	V	8000...7FFFH	-694.51059...694.48940	0.0211948	[mg/stk]
Offset of air mass flow for detection of limited dynamics for cylinder individual lambda control					
N_OFS_LDC_LAM_CYL_SEL	V	E020...1FE0H	-8160...8160	1	[rpm]
Offset of engine speed for detection of limited dynamics for cylinder individual lambda control					

Input data:

N_DELTA_LDC	MAF_DELTA_LDC	NC_CBK_EX_NR
MAF_CYL	LV_ST_END	LV_ES
FAC_LAM_MV_DELTA_LDC[NC_CBK_EX_NR]	LF_LS_CBK_EX_LAM_CYL_SEL_CONF	

FUNCTION DESCRIPTION:

General information:

if(1) two separate exhaust banks are concerned (NC_CBK_EX_NR = 2)
then
i = 1, for exhaust bank 1
i = 2, for exhaust bank 2

else(1)

i = 1 (NC_CBK_EX_NR = 1)

endif(1)

*use **LF_LS_CBK_EX_LAM_CYL_SEL_CONF** to distinguish between single and 2-exhaust-bank configuration – index "[i]"*

Description:

The condition limited dynamics LV_LAM_CYL_SEL_LDC[i] for cylinder individual lambda control is detected if the engine speed, mass air flow and the mean value of lambda controller output are within the corresponding dynamic window. Once all of the conditions are fulfilled, the integral of the mass air flow is calculated. The limited dynamics is then detected (LV_LAM_CYL_SEL_LDC[i] = 1) if the value of mass air flow integral reaches the calibrated threshold.

Application conditions:

Initialisation:

at reset, LV_IGK = 0 → 1 and clear error memory:
LV_LAM_CYL_SEL_LDC[i] = 0
MAF_INT_LDC_LAM_CYL_SEL[i] = 0
FAC_LAM_MV_OFS_LDC_LAM_CYL_SEL[i] = 0
MAF_OFS_LDC_LAM_CYL_SEL = 0
N_OFS_LDC_LAM_CYL_SEL = 0

Recurrence:

T_SAMPLE = 20 ms

Activation:

LV_ST_END = 1

Deactivation:

LV_ES = 1

Formula section:

4.4.1 Evaluation of the dynamic window for the engine speed

```
if$^{(1)}$ abs(N_DELTA_LDC) > C_N_DYW_LDC_LAM_CYL_SEL
then
    if$^{(2)}$ abs(N_DELTA_LDC - N_OFS_LDC_LAM_CYL_SEL) >
                                                C_N_DYW_LDC_LAM_CYL_SEL
    then
        N_OFS_LDC_LAM_CYL_SEL = N_DELTA_LDC
                                                              (4.4.1.1)
        MAF_INT_LDC_LAM_CYL_SEL[i] = 0
                                                              (4.4.1.2)
    endif$^{(2)}$
else$^{(1)}$
    N_OFS_LDC_LAM_CYL_SEL = 0
                                                              (4.4.1.3)
endif$^{(1)}$
```

4.4.2 Evaluation of the dynamic window for the air mass flow

```
if$^{(1)}$ abs(MAF_DELTA_LDC) > C_MAF_DYW_LDC_LAM_CYL_SEL
then
    if$^{(2)}$ abs(MAF_DELTA_LDC - MAF_OFS_LDC_LAM_CYL_SEL) >
                                              C_MAF_DYW_LDC_LAM_CYL_SEL
    then
        MAF_OFS_LDC_LAM_CYL_SEL = MAF_DELTA_LDC
                                                              (4.4.2.1)
        MAF_INT_LDC_LAM_CYL_SEL[i] = 0
                                                              (4.4.2.2)
    endif$^{(2)}$
else$^{(1)}$
    MAF_OFS_LDC_LAM_CYL_SEL = 0
                                                              (4.4.2.3)
endif$^{(1)}$
```

4.4.3 Evaluation of the dynamic window for the lambda controller output mean value

```
if$^{(1)}$ abs(FAC_LAM_MV_DELTA_LDC[i]) > C_FAC_LAM_DYW_LDC_LAM_CYL_SEL
then
    if$^{(2)}$ abs(FAC_LAM_MV_DELTA_LDC[i] –
                                  FAC_LAM_MV_OFS_LDC_LAM_CYL_SEL[i]) >
                                     C_FAC_LAM_DYW_LDC_LAM_CYL_SEL
    then
        FAC_LAM_MV_OFS_LDC_LAM_CYL_SEL[i] = FAC_LAM_MV_DELTA_LDC[i]
                                                              (4.4.3.1)
        MAF_INT_LDC_LAM_CYL_SEL[i] = 0
                                                              (4.4.3.2)
    endif$^{(2)}$
else$^{(1)}$
    FAC_LAM_MV_OFS_LDC_LAM_CYL_SEL[i] = 0
                                                              (4.4.3.3)
endif$^{(1)}$
```

4.4.4 Calculation of the air mass flow integral – detection of limited dynamics

```
if$^{(1)}$ MAF_INT_LDC_LAM_CYL_SEL[i] < C_MAF_INT_LDC_LAM_CYL_SEL
then
    MAF_INT_LDC_LAM_CYL_SEL[i]$^{[g]}$ = MAF_INT_LDC_LAM_CYL_SEL[i]$^{[g]}$ +
            MAF_CYL$^{[kg/h]}$*T_SAMPLE$^{[ms]}$*1/3600$^{[(g*h)/(kg*ms)]}$ (4.4.4.1)
    LV_LAM_CYL_SEL_LDC[i] = 0
                                                              (4.4.4.2)
else$^{(1)}$
    LV_LAM_CYL_SEL_LDC[i] = 1
                                                              (4.4.4.3)
endif$^{(1)}$
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_DYW_LDC_LAM_CYL_SEL	1	0...1FE0H	0...8160	1	[rpm]
Dynamic window of engine speed for detection of limited dynamics (cylinder individual lambda control)					
C_MAF_DYW_LDC_LAM_CYL_SEL	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Dynamic window of MAF for detection of limited dynamics (cylinder individual lambda control)					
C_FAC_LAM_DYW_LDC_LAM_CYL_SEL	1	0...7FFFH	0...49.99847	1.5259e-3	[%]
Dynamic window of lambda controller output for detection of limited dynamics (cylinder individual lambda control)					
C_MAF_INT_LDC_LAM_CYL_SEL	1	0...FFFFH	0...1456.33333	0.0222222	[g]
Threshold of air mass flow integral for detection of limited dynamics for cylinder individual lambda control					

4.5 Cylinder Individual Lambda Control Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_LAM_CYL_SEL_i	V/O	0...1H	0...1	1	-
CILC error detected over several driving cycles					

Input data:

LV_ES	LV_ST_END	NC_CBK_EX_NR
LV_INH_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]		

FUNCTION DESCRIPTION:

General information:

Variable assignment

Description of index "_i":

*use **NC_LAMB_REF**: pattern for allocation of physical cylinders to exhaust bank - index "_i or [i]"*

***Remark*:**
index "_i" is equivalent to "[i]"

Application conditions:

Initialization:

at reset, LV_IGK = 0 → 1 and clear error memory:

LV_ERR_LAM_CYL_SEL_i = 0

Recurrence:

T_SAMPLE = 100 ms

Activation:

- LV_ST_END = 1

Deactivation:

- LV_ES = 1

Formula section:

LV_ERR_LAM_CYL_SEL_i = LV_INH_OBD_DIAG_CYL_BAL_LAM[i]

4.6 Cylinder balancing diagnosis

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_CDN_DIAG_CYL_BAL_LAM[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Diagnosis condition - output of CILC concerned					
LV_CDN_DIAG_CYL_BAL_ER[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Diagnosis conditions - output of CYBL_ER concerned					
ERR_SYM_CYL_BAL_LAM[NC_CYL_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected failure symptom - output of CILC concerned					
ERR_SYM_CYL_BAL_ER[NC_CYL_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected failure symptom - output of CYBL_ER concerned					
LV_ERR_CYL_BAL_LAM[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Cylinder balancing error detected - output of CILC concerned					
LV_ERR_CYL_BAL_ER[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Cylinder balancing error detected - output of CYBL_ER concerned					
LV_END_DIAG_CYL_BAL_LAM[NC_CYL_NR]	V	0...1H	0...1	1	[-]
End of first diagnosis cycle - output of CILC concerned					
LV_END_DIAG_CYL_BAL_ER[NC_CYL_NR]	V	0...1H	0...1	1	[-]
End of first diagnosis cycle - output of CYBL_ER concerned					
LV_END_DIAG_WIN_CYL_BAL_LAM[NC_CYL_NR]	V	0...1H	0...1	1	[-]
End of diagnosis cycle for similar conditions window - output of CILC concerned					
LV_END_DIAG_WIN_CYL_BAL_ER[NC_CYL_NR]	V	0...1H	0...1	1	[-]
End of diagnosis cycle for similar conditions window - output of CYBL_ER concerned					
T_SUM_CYL_BAL_LAM_LIM_MIN[NC_CYL_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for detected diagnosis limit - lower limit of CILC concerned					
T_SUM_CYL_BAL_LAM_LIM_MAX[NC_CYL_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for detected diagnosis limit - upper limit of CILC concerned					
T_SUM_CYL_BAL_ER_LIM_MIN[NC_CYL_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for detected diagnosis limit - lower limit of CYBL_ER concerned					
T_SUM_CYL_BAL_ER_LIM_MAX[NC_CYL_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for detected diagnosis limit - upper limit of CYBL_ER concerned					
T_SUM_RST_CYL_BAL_LAM[NC_CYL_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer to update the diagnosis window - output of CILC concerned					
T_SUM_RST_CYL_BAL_ER[NC_CYL_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer to update the diagnosis window - output of CYBL_ER concerned					
T_SUM_END_DIAG_WIN_CYL_LAM[NC_CYL_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for the diagnosis window - output of CILC concerned					
T_SUM_END_DIAG_WIN_CYL_BAL_ER[NC_CYL_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for the diagnosis window - output of CYBL_ER concerned					

<table>
<tr><td>CTR_ERR_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]</td><td>O/V/S</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce-counter for detected error up to current driving cycle (output of CILC concerned)</td></tr>
<tr><td>CTR_ERR_OBD_DIAG_CYL_BAL_ER</td><td>O/V/S</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce-counter for detected error up to current driving cycle (output of CYBL_ER concerned)</td></tr>
<tr><td>LV_INH_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Inhibit flag for CILC - cylinder balancing error detected over several driving cycles</td></tr>
<tr><td>LV_INH_OBD_DIAG_CYL_BAL_ER</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Inhibit flag for CYBL_ER - cylinder balancing error detected over several driving cycles</td></tr>
<tr><td>LV_ERR_REF_CYL_BAL_LAM[NC_CBK_EX_NR]</td><td>V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Reference flag indicating the CILC - error was occurred at least once for the current driving cycle</td></tr>
<tr><td>LV_ERR_REF_CYL_BAL_ER</td><td>V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Reference flag indicating the CYBL_ER - error was occurred at least once for the current driving cycle</td></tr>
</table>

Input data:

<table>
<tr><td colspan="3">LV_ST_END</td><td colspan="3">NC_CYL_NR</td><td colspan="3">LV_DC</td><td colspan="3">LV_IGK</td></tr>
<tr><td colspan="4">STATE_DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]</td><td colspan="4">STATE_CTL_TI_ER_BAL</td><td colspan="4">NC_CBK_EX_NR</td></tr>
<tr><td colspan="4">FAC_LAM_CYL_SEL_ADJ_L_RNG[NC_CYL_NR]</td><td colspan="4">FAC_LAM_CYL_SEL_ADJ_H_RNG[NC_CYL_NR]</td><td colspan="4">STATE_LAM_CYL_SEL_ADJ[NC_CBK_EX_NR]</td></tr>
<tr><td colspan="6">LV_CTR_CYL_BAL_RST_EXT</td><td colspan="6">LF_LS_CBK_EX_LAM_CYL_SEL_CONF</td></tr>
<tr><td colspan="6">LV_MFF_ADD_ER_BAL_OBD_MAX_POS[NC_CYL_NR]</td><td colspan="6">LV_MFF_ADD_ER_BAL_OBD_MAX_NEG[NC_CYL_NR]</td></tr>
<tr><td colspan="6">LV_MFF_FAC_ER_BAL_OBD_MAX_POS[NC_CYL_NR]</td><td colspan="6">LV_MFF_FAC_ER_BAL_OBD_MAX_NEG[NC_CYL_NR]</td></tr>
</table>

FUNCTION DESCRIPTION:

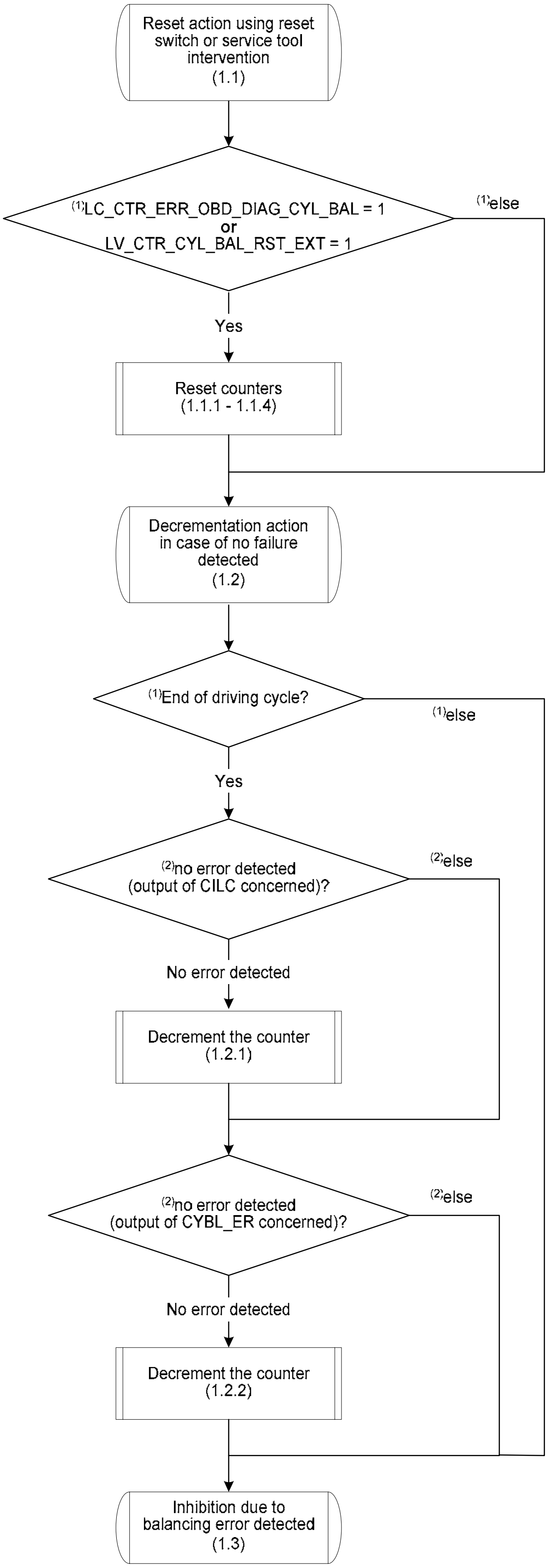
General information:

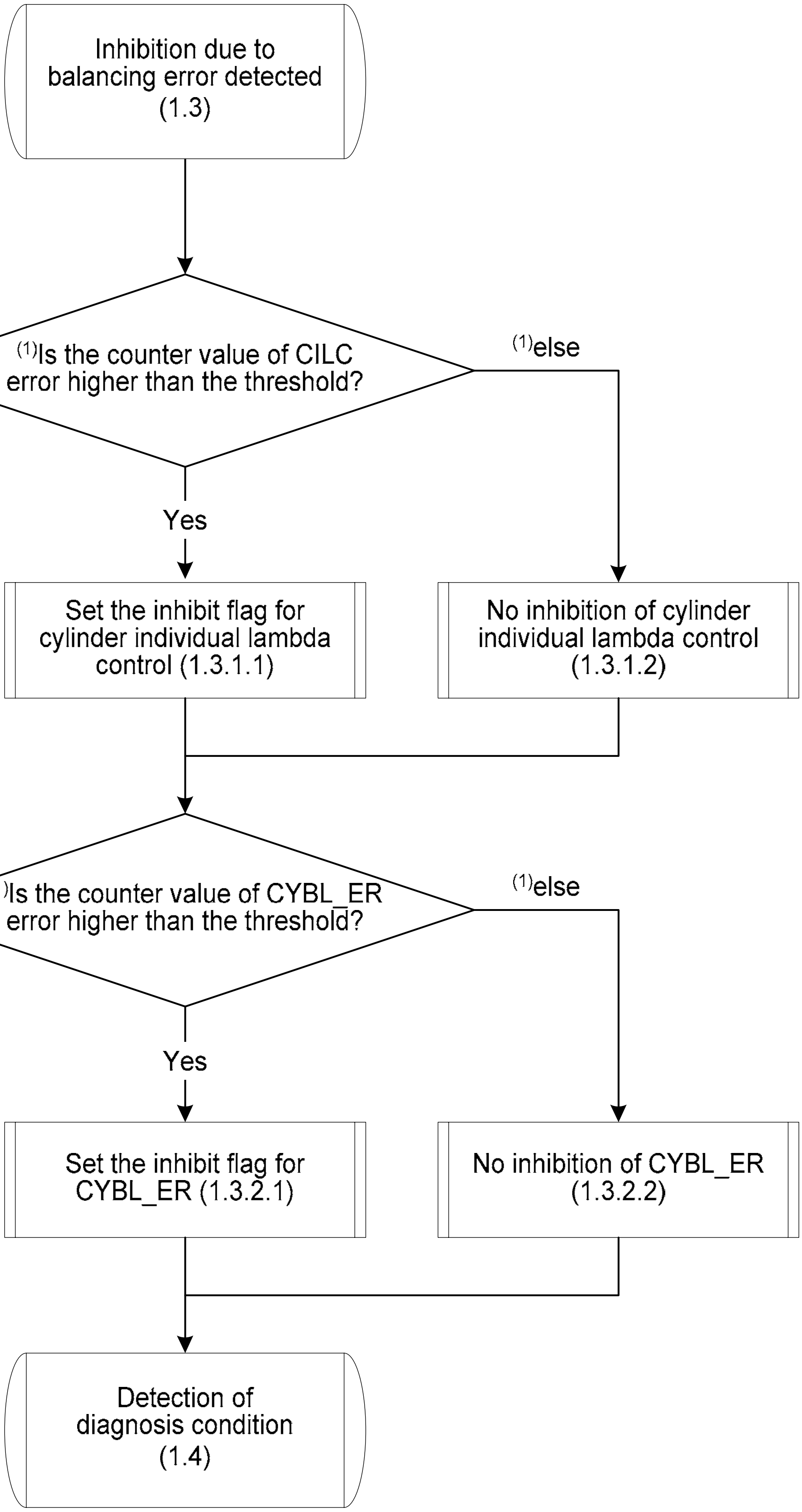
The objective function of cylinder balancing diagnosis is monitoring the output of cylinder individual lambda control in homogenous engine operation and output of cylinder balancing via engine roughness in stratified or homogenous-stratified engine operation mode. Breaking the diagnosis limits for a long time, which may have been caused by injector- failures, shall be diagnosed by cylinder balancing diagnosis. The failure allocation is cylinder selective. In case of failure detection for one of the cylinders the failure symptom of remaining cylinders shall be checked. Should the run-off time of any cylinder be over the minimum threshold, the concerned cylinder has to be detected as defect. Should the output of cylinder individual lambda control be concerned, the diagnosis shall be executed for all of the exhaust-bank-related cylinders (4.6.2, 4.6.4.1, 4.6.5, 4.6.6.1, 4.6.7.1.1).
The diagnosis shall be executed for all of the cylinders if the output of cylinder balancing via engine roughness is concerned (4.6.2, 4.6.4.2, 4.6.5, 4.6.6.2, 4.6.7.2).

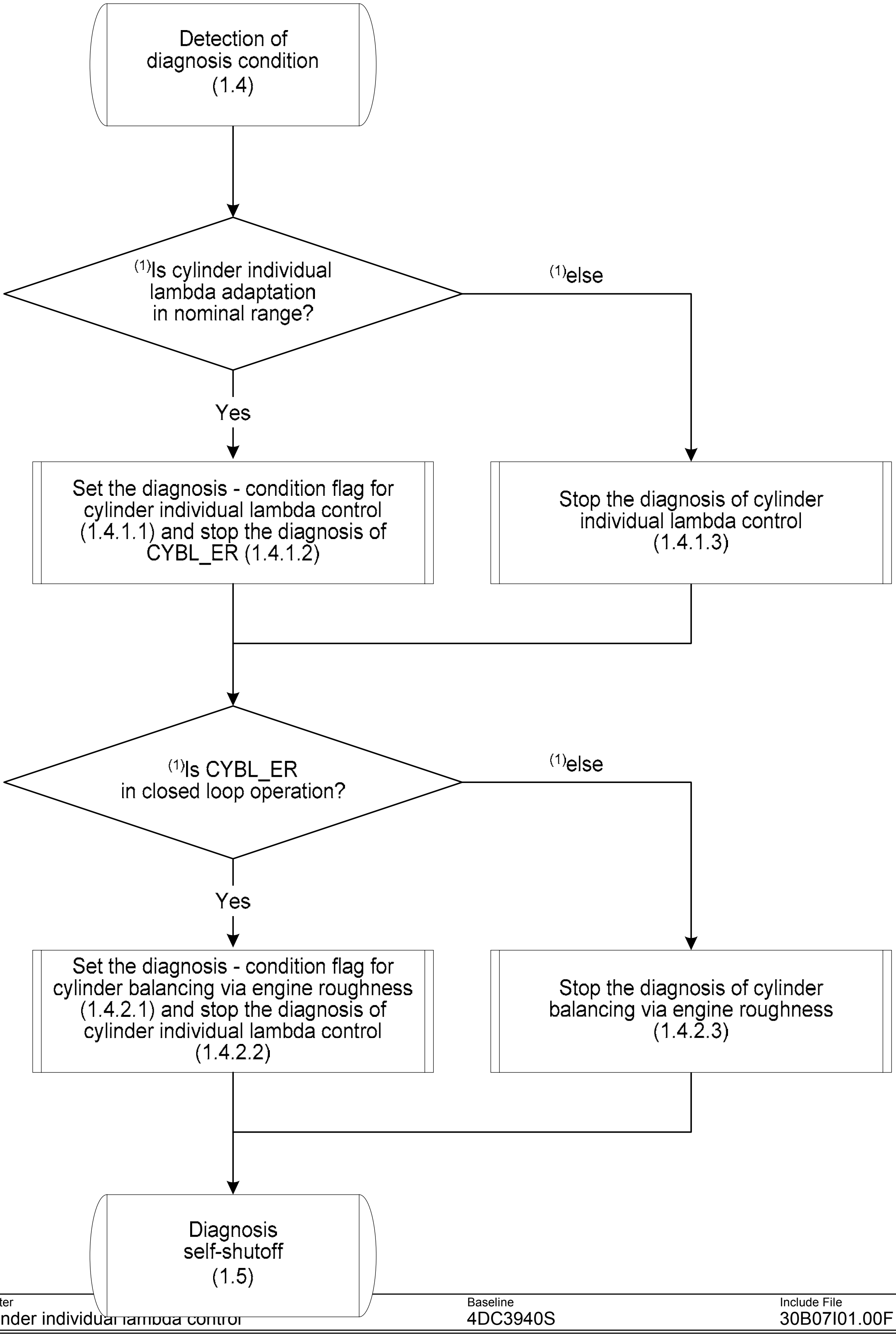
Description of index "[i]" and "[x]":

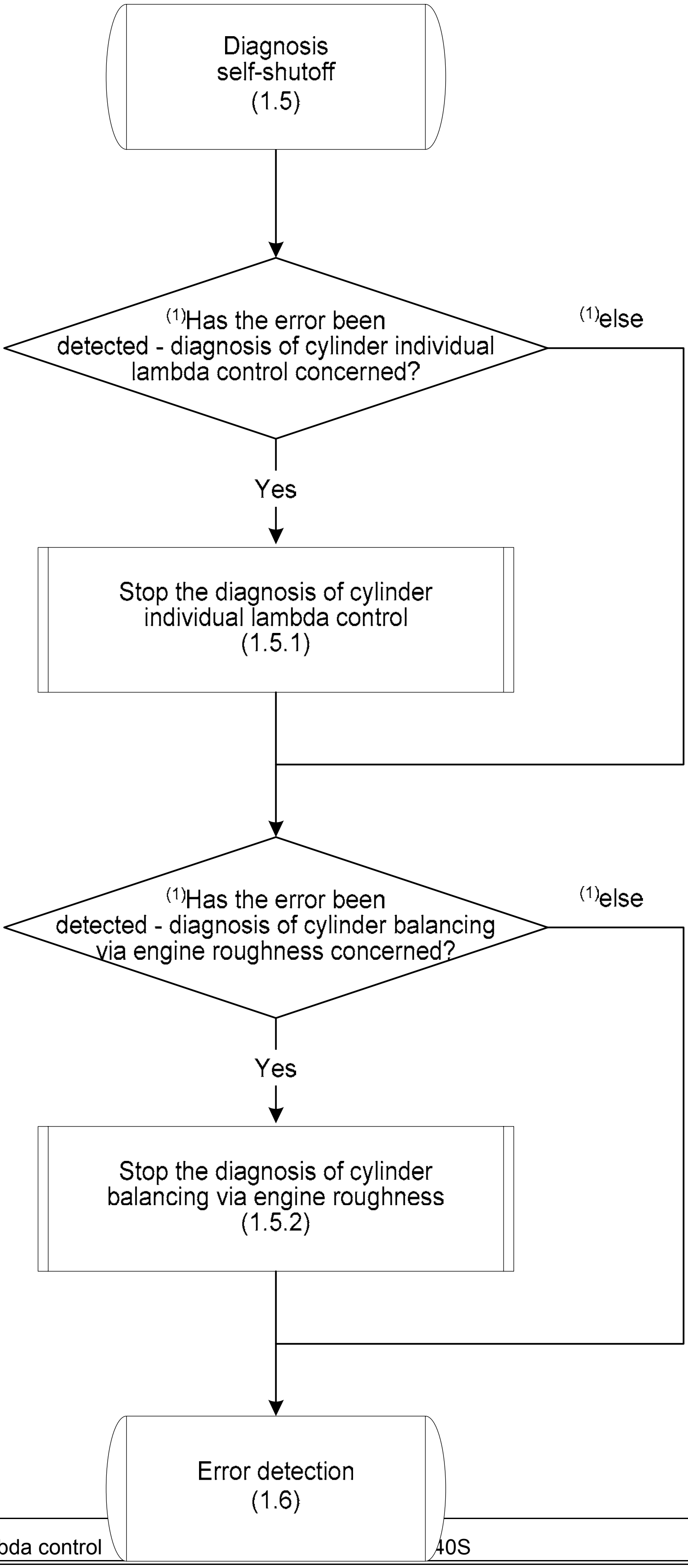
x: logical cylinder
*use **NC_CYL_NR**: number of engine cylinders – index "[x]"*
*use **LF_LS_CBK_EX_LAM_CYL_SEL_CONF**: pattern for allocation of physical cylinders to exhaust bank - index "[i]*

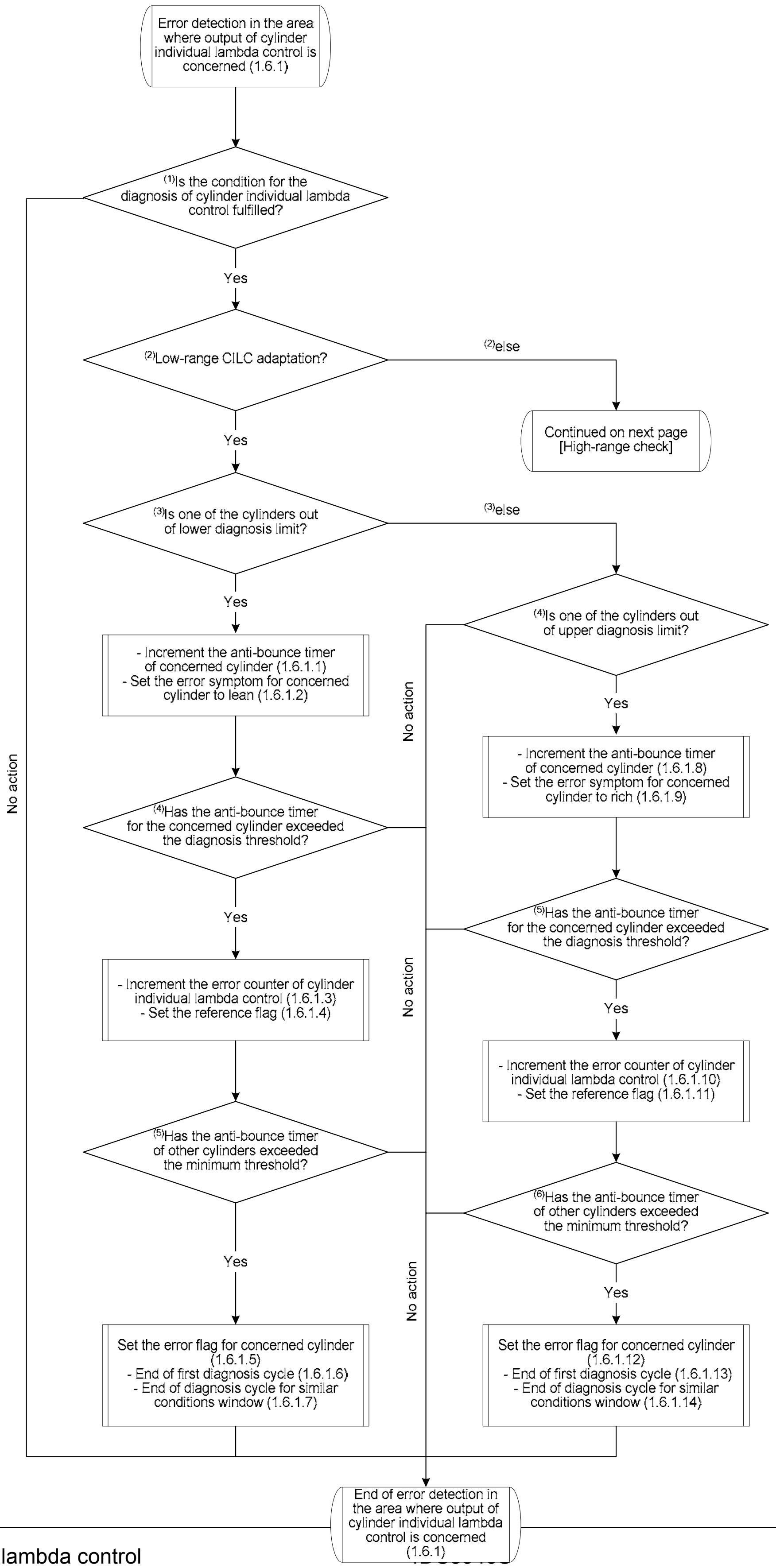
Signal flow diagram:

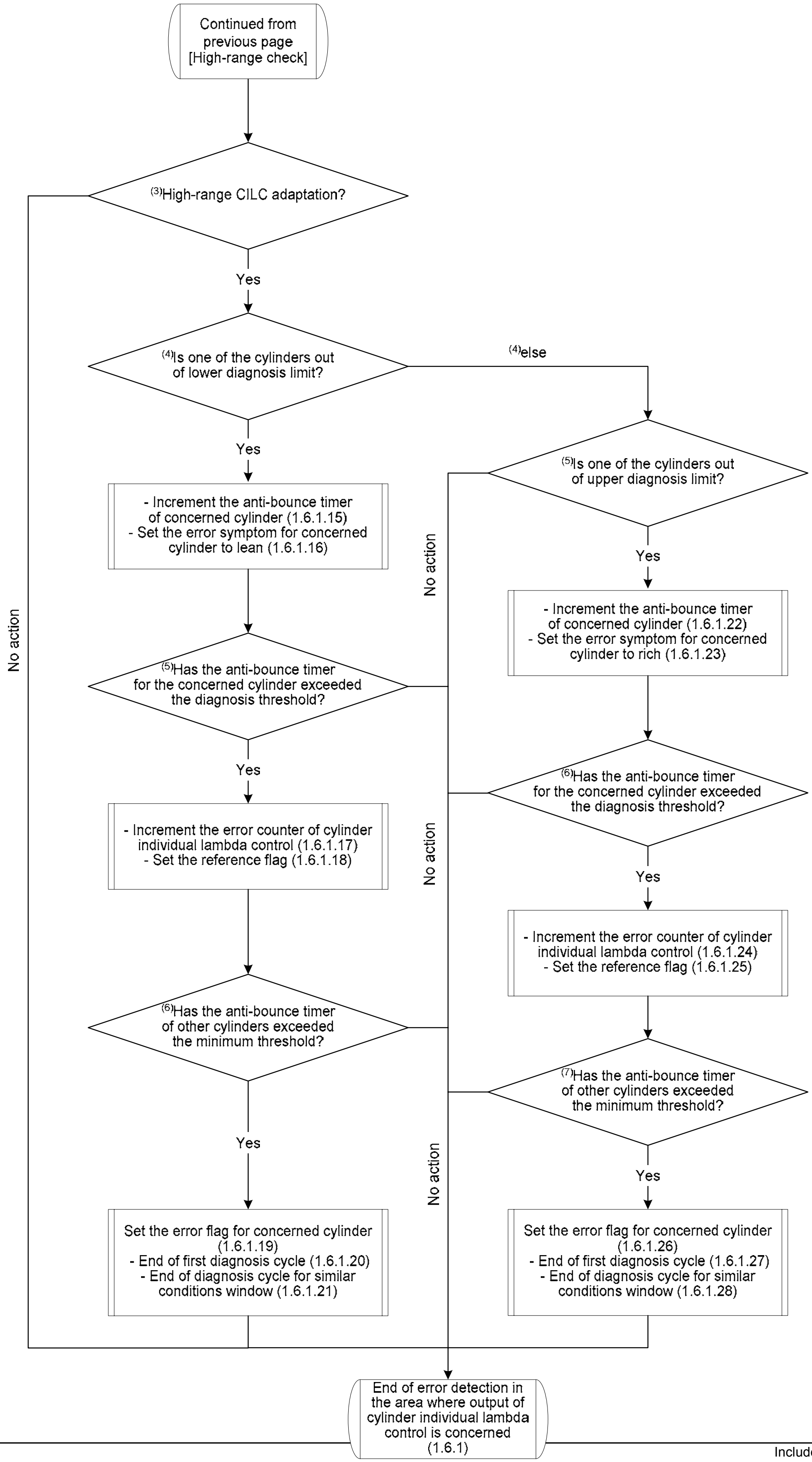


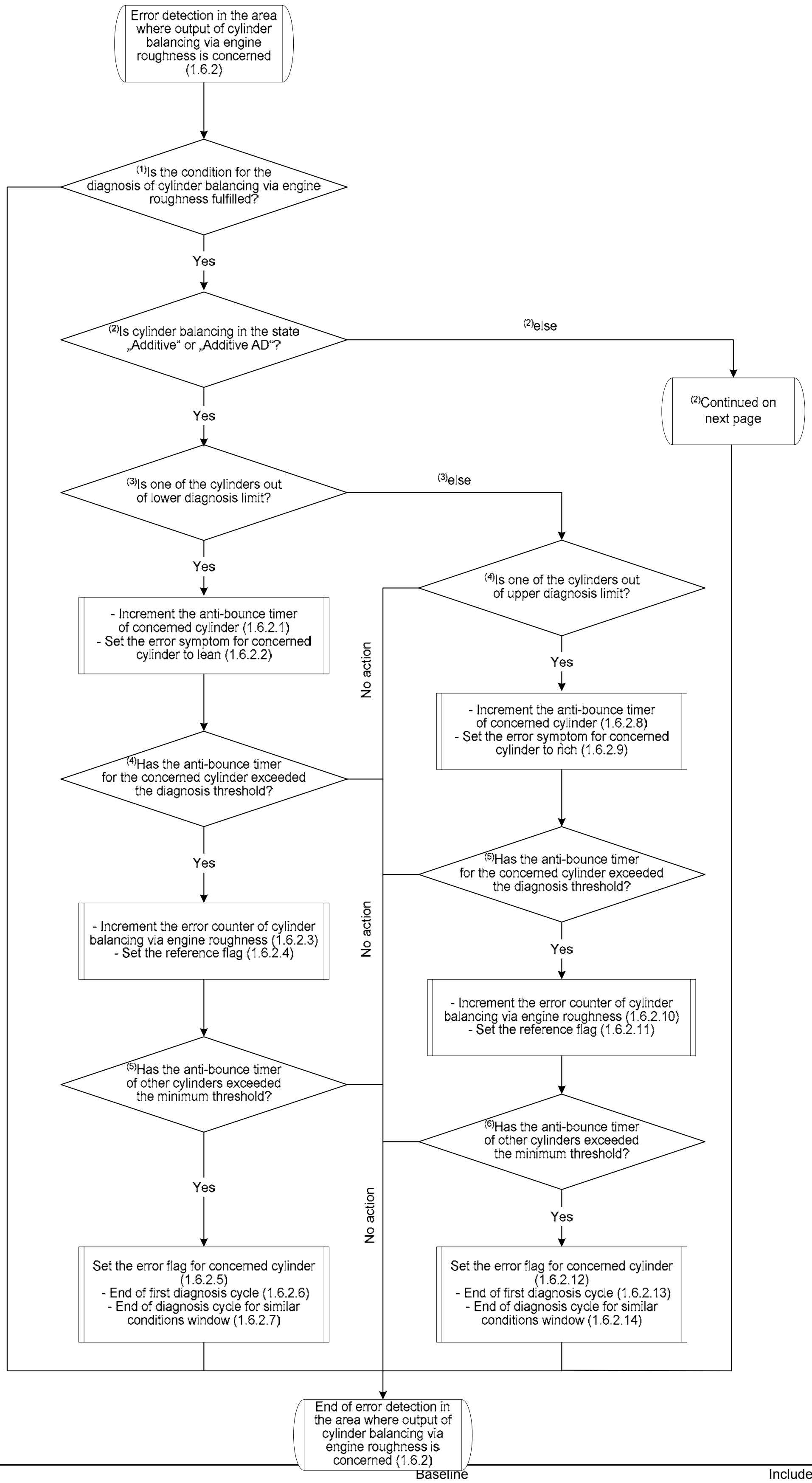

Inhibition due to balancing error detected (1.3)
(1)Is the counter value of CILC error higher than the threshold?
(1)else
Yes
Set the inhibit flag for cylinder individual lambda control (1.3.1.1)
No inhibition of cylinder individual lambda control (1.3.1.2)
(1)Is the counter value of CYBL_ER error higher than the threshold?
(1)else
Yes
Set the inhibit flag for CYBL_ER (1.3.2.1)
No inhibition of CYBL_ER (1.3.2.2)
Detection of diagnosis condition (1.4)

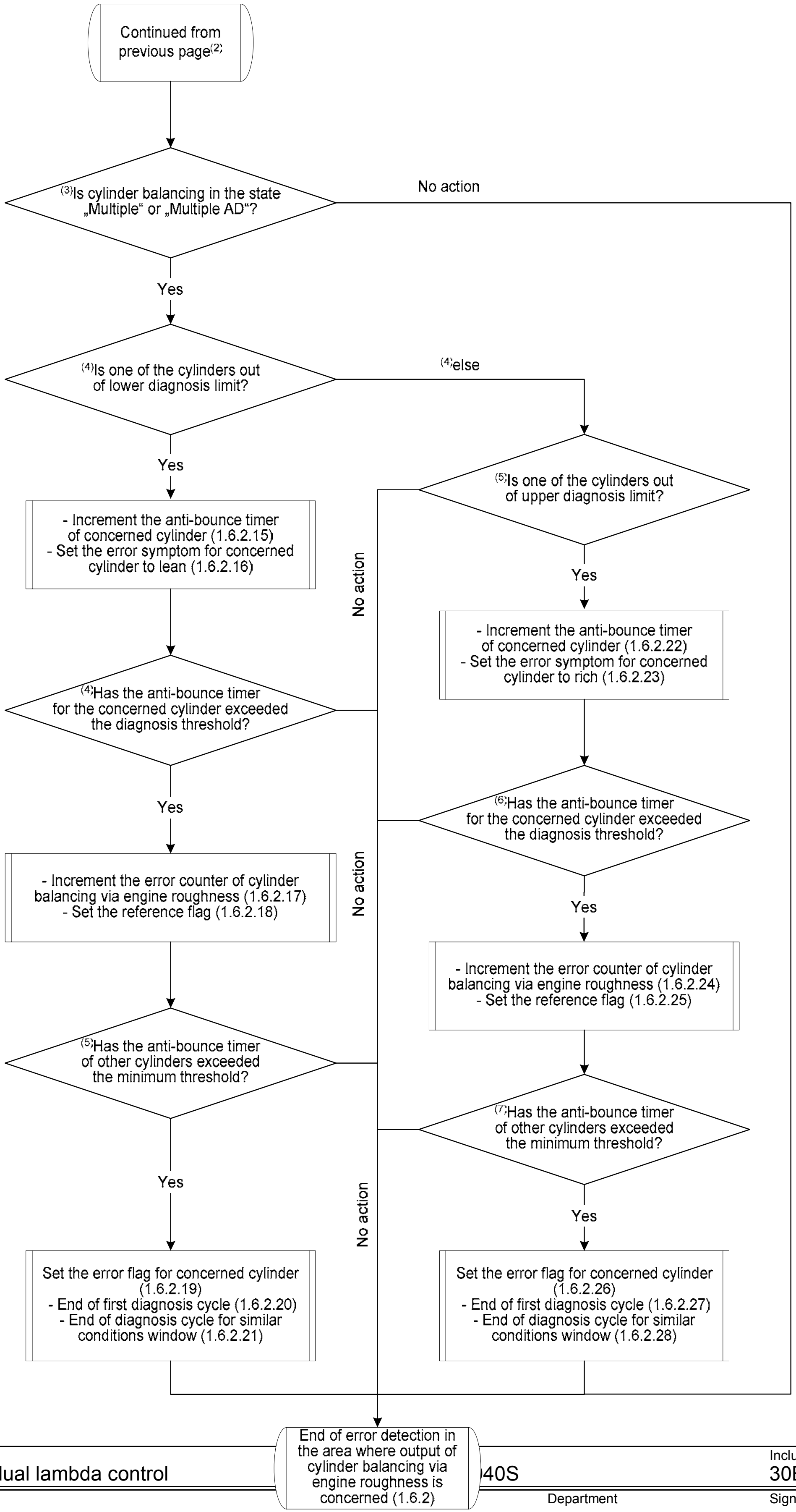

Detection of diagnosis condition (1.4)
(1)Is cylinder individual lambda adaptation in nominal range?
(1)else
Yes
Set the diagnosis - condition flag for cylinder individual lambda control (1.4.1.1) and stop the diagnosis of CYBL_ER (1.4.1.2)
Stop the diagnosis of cylinder individual lambda control (1.4.1.3)
(1)Is CYBL_ER in closed loop operation?
(1)else
Yes
Set the diagnosis - condition flag for cylinder balancing via engine roughness (1.4.2.1) and stop the diagnosis of cylinder individual lambda control (1.4.2.2)
Stop the diagnosis of cylinder balancing via engine roughness (1.4.2.3)
Diagnosis self-shutoff (1.5)

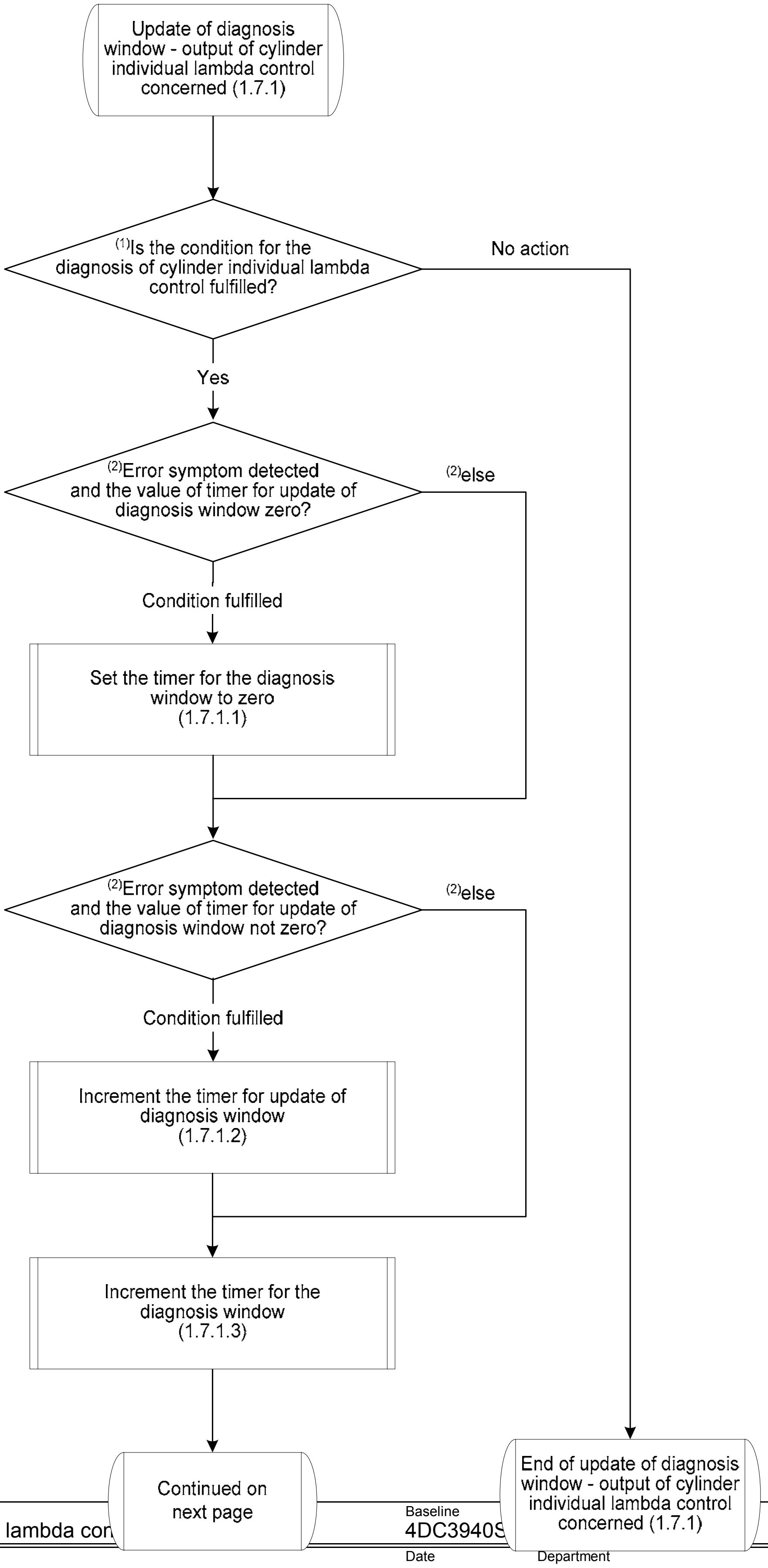

Diagnosis
self-shutoff
(1.5)
(1)Has the error been detected - diagnosis of cylinder individual lambda control concerned?
(1)else
Yes
Stop the diagnosis of cylinder individual lambda control
(1.5.1)
(1)Has the error been detected - diagnosis of cylinder balancing via engine roughness concerned?
(1)else
Yes
Stop the diagnosis of cylinder balancing via engine roughness
(1.5.2)
Error detection
(1.6)

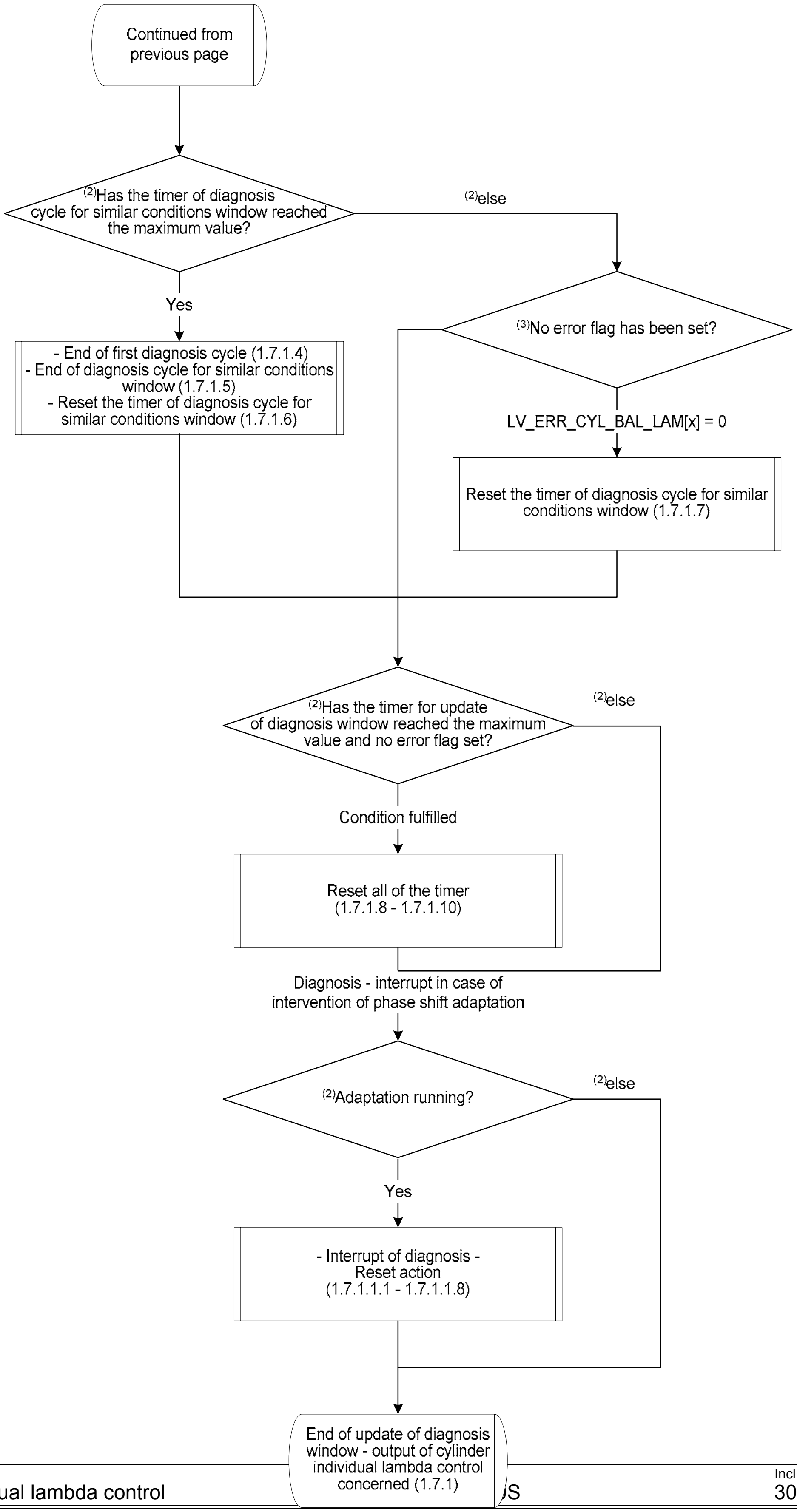

Error detection in the area where output of cylinder individual lambda control is concerned (1.6.1)
(1)Is the condition for the diagnosis of cylinder individual lambda control fulfilled?
No action
Yes
(2)Low-range CILC adaptation?
(2)else
Continued on next page [High-range check]
Yes
(3)Is one of the cylinders out of lower diagnosis limit?
(3)else
Yes
- Increment the anti-bounce timer of concerned cylinder (1.6.1.1)
- Set the error symptom for concerned cylinder to lean (1.6.1.2)
(4)Is one of the cylinders out of upper diagnosis limit?
No action
Yes
- Increment the anti-bounce timer of concerned cylinder (1.6.1.8)
- Set the error symptom for concerned cylinder to rich (1.6.1.9)
(4)Has the anti-bounce timer for the concerned cylinder exceeded the diagnosis threshold?
Yes
(5)Has the anti-bounce timer for the concerned cylinder exceeded the diagnosis threshold?
No action
Yes
- Increment the error counter of cylinder individual lambda control (1.6.1.3)
- Set the reference flag (1.6.1.4)
- Increment the error counter of cylinder individual lambda control (1.6.1.10)
- Set the reference flag (1.6.1.11)
(5)Has the anti-bounce timer of other cylinders exceeded the minimum threshold?
(6)Has the anti-bounce timer of other cylinders exceeded the minimum threshold?
Yes
No action
Yes
Set the error flag for concerned cylinder (1.6.1.5)
- End of first diagnosis cycle (1.6.1.6)
- End of diagnosis cycle for similar conditions window (1.6.1.7)
Set the error flag for concerned cylinder (1.6.1.12)
- End of first diagnosis cycle (1.6.1.13)
- End of diagnosis cycle for similar conditions window (1.6.1.14)
End of error detection in the area where output of cylinder individual lambda control is concerned (1.6.1)

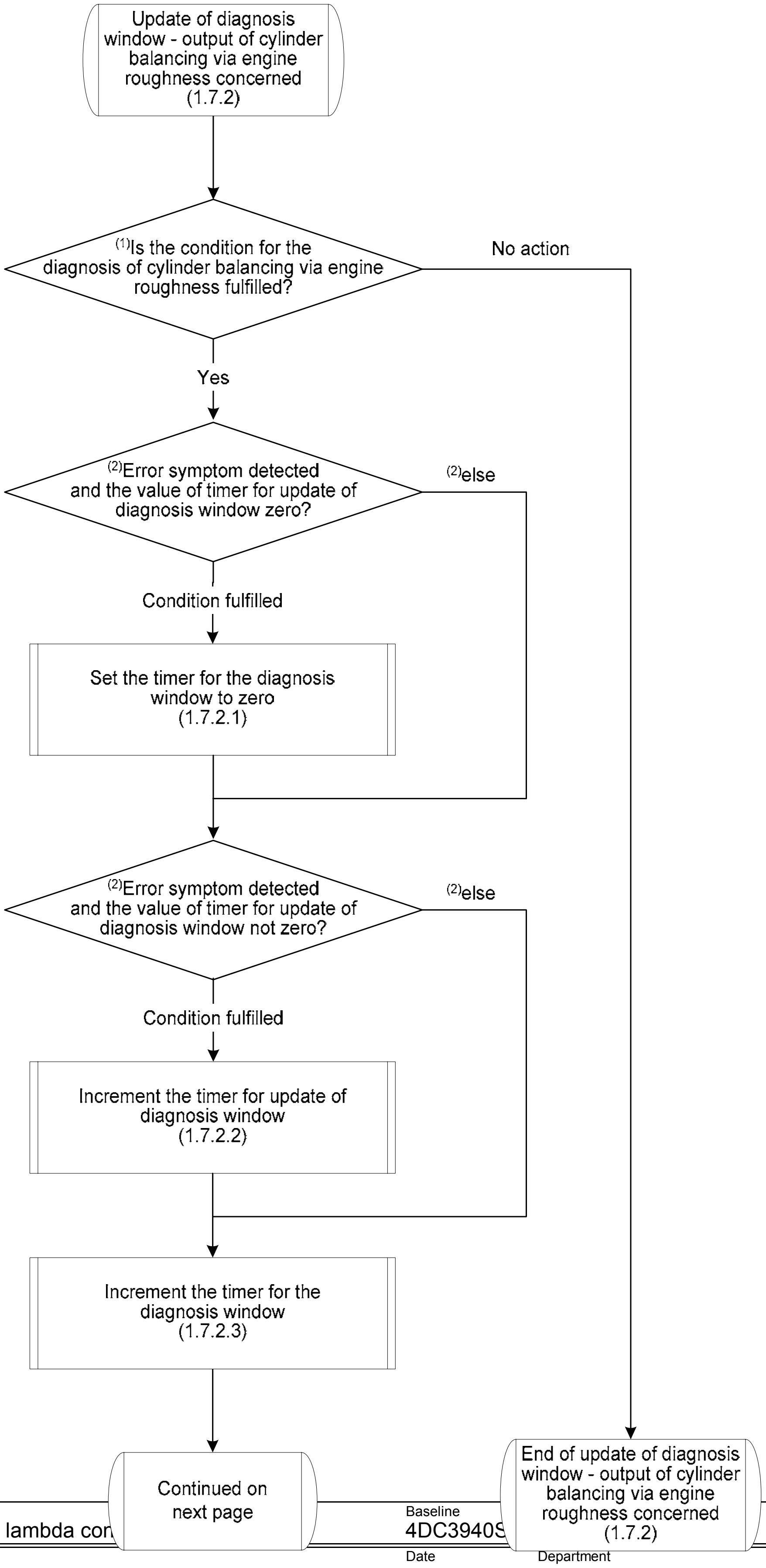

Continued from previous page [High-range check]
(3)High-range CILC adaptation?
Yes
No action
(4)Is one of the cylinders out of lower diagnosis limit?
(4)else
Yes
- Increment the anti-bounce timer of concerned cylinder (1.6.1.15)
- Set the error symptom for concerned cylinder to lean (1.6.1.16)
(5)Has the anti-bounce timer for the concerned cylinder exceeded the diagnosis threshold?
Yes
- Increment the error counter of cylinder individual lambda control (1.6.1.17)
- Set the reference flag (1.6.1.18)
(6)Has the anti-bounce timer of other cylinders exceeded the minimum threshold?
Yes
Set the error flag for concerned cylinder (1.6.1.19)
- End of first diagnosis cycle (1.6.1.20)
- End of diagnosis cycle for similar conditions window (1.6.1.21)
No action
No action
No action
(5)Is one of the cylinders out of upper diagnosis limit?
Yes
- Increment the anti-bounce timer of concerned cylinder (1.6.1.22)
- Set the error symptom for concerned cylinder to rich (1.6.1.23)
(6)Has the anti-bounce timer for the concerned cylinder exceeded the diagnosis threshold?
Yes
- Increment the error counter of cylinder individual lambda control (1.6.1.24)
- Set the reference flag (1.6.1.25)
(7)Has the anti-bounce timer of other cylinders exceeded the minimum threshold?
Yes
Set the error flag for concerned cylinder (1.6.1.26)
- End of first diagnosis cycle (1.6.1.27)
- End of diagnosis cycle for similar conditions window (1.6.1.28)
End of error detection in the area where output of cylinder individual lambda control is concerned (1.6.1)

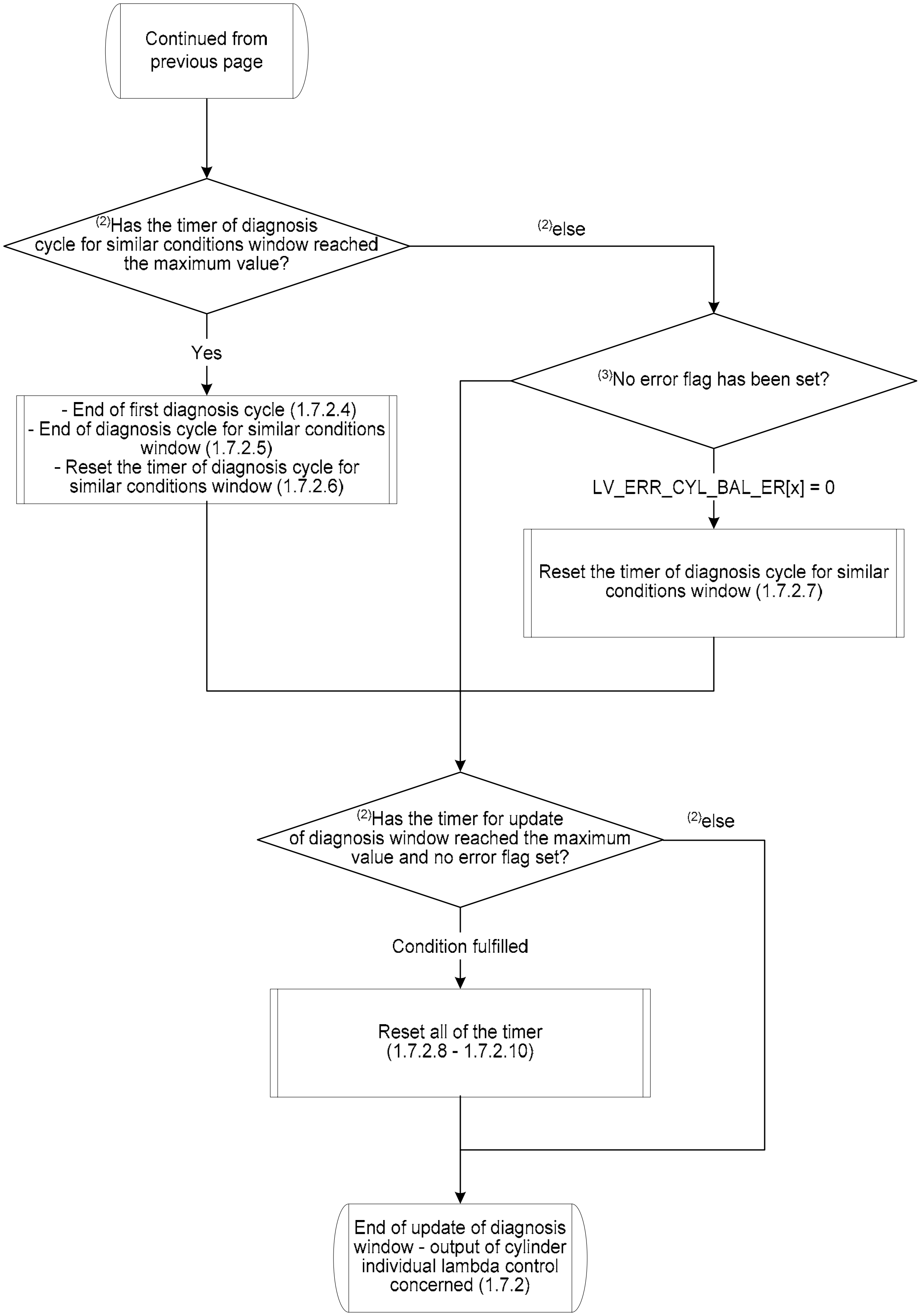

Error detection in the area where output of cylinder balancing via engine roughness is concerned (1.6.2)
(1)Is the condition for the diagnosis of cylinder balancing via engine roughness fulfilled?
No action
Yes
(2)Is cylinder balancing in the state „Additive“ or „Additive AD“?
(2)else
(2)Continued on next page
Yes
(3)Is one of the cylinders out of lower diagnosis limit?
(3)else
Yes
- Increment the anti-bounce timer of concerned cylinder (1.6.2.1)
- Set the error symptom for concerned cylinder to lean (1.6.2.2)
(4)Is one of the cylinders out of upper diagnosis limit?
No action
Yes
- Increment the anti-bounce timer of concerned cylinder (1.6.2.8)
- Set the error symptom for concerned cylinder to rich (1.6.2.9)
(4)Has the anti-bounce timer for the concerned cylinder exceeded the diagnosis threshold?
Yes
(5)Has the anti-bounce timer for the concerned cylinder exceeded the diagnosis threshold?
No action
Yes
- Increment the error counter of cylinder balancing via engine roughness (1.6.2.3)
- Set the reference flag (1.6.2.4)
- Increment the error counter of cylinder balancing via engine roughness (1.6.2.10)
- Set the reference flag (1.6.2.11)
(5)Has the anti-bounce timer of other cylinders exceeded the minimum threshold?
(6)Has the anti-bounce timer of other cylinders exceeded the minimum threshold?
No action
Yes
Yes
Set the error flag for concerned cylinder (1.6.2.5)
- End of first diagnosis cycle (1.6.2.6)
- End of diagnosis cycle for similar conditions window (1.6.2.7)
Set the error flag for concerned cylinder (1.6.2.12)
- End of first diagnosis cycle (1.6.2.13)
- End of diagnosis cycle for similar conditions window (1.6.2.14)
End of error detection in the area where output of cylinder balancing via engine roughness is concerned (1.6.2)


Continued from previous page(2)
(3)Is cylinder balancing in the state „Multiple“ or „Multiple AD“?
No action
Yes
(4)Is one of the cylinders out of lower diagnosis limit?
(4)else
Yes
- Increment the anti-bounce timer of concerned cylinder (1.6.2.15)
- Set the error symptom for concerned cylinder to lean (1.6.2.16)
(5)Is one of the cylinders out of upper diagnosis limit?
No action
Yes
- Increment the anti-bounce timer of concerned cylinder (1.6.2.22)
- Set the error symptom for concerned cylinder to rich (1.6.2.23)
(4)Has the anti-bounce timer for the concerned cylinder exceeded the diagnosis threshold?
(6)Has the anti-bounce timer for the concerned cylinder exceeded the diagnosis threshold?
Yes
No action
- Increment the error counter of cylinder balancing via engine roughness (1.6.2.17)
- Set the reference flag (1.6.2.18)
Yes
- Increment the error counter of cylinder balancing via engine roughness (1.6.2.24)
- Set the reference flag (1.6.2.25)
(5)Has the anti-bounce timer of other cylinders exceeded the minimum threshold?
(7)Has the anti-bounce timer of other cylinders exceeded the minimum threshold?
Yes
No action
Yes
Set the error flag for concerned cylinder (1.6.2.19)
- End of first diagnosis cycle (1.6.2.20)
- End of diagnosis cycle for similar conditions window (1.6.2.21)
Set the error flag for concerned cylinder (1.6.2.26)
- End of first diagnosis cycle (1.6.2.27)
- End of diagnosis cycle for similar conditions window (1.6.2.28)
End of error detection in the area where output of cylinder balancing via engine roughness is concerned (1.6.2)


Update of diagnosis window - output of cylinder individual lambda control concerned (1.7.1)
(1)Is the condition for the diagnosis of cylinder individual lambda control fulfilled?
No action
Yes
(2)Error symptom detected and the value of timer for update of diagnosis window zero?
(2)else
Condition fulfilled
Set the timer for the diagnosis window to zero (1.7.1.1)
(2)Error symptom detected and the value of timer for update of diagnosis window not zero?
(2)else
Condition fulfilled
Increment the timer for update of diagnosis window (1.7.1.2)
Increment the timer for the diagnosis window (1.7.1.3)
Continued on next page
End of update of diagnosis window - output of cylinder individual lambda control concerned (1.7.1)


Continued from previous page
(2)Has the timer of diagnosis cycle for similar conditions window reached the maximum value?
(2)else
Yes
(3)No error flag has been set?
- End of first diagnosis cycle (1.7.1.4)
- End of diagnosis cycle for similar conditions window (1.7.1.5)
- Reset the timer of diagnosis cycle for similar conditions window (1.7.1.6)
LV_ERR_CYL_BAL_LAM[x] = 0
Reset the timer of diagnosis cycle for similar conditions window (1.7.1.7)
(2)Has the timer for update of diagnosis window reached the maximum value and no error flag set?
(2)else
Condition fulfilled
Reset all of the timer (1.7.1.8 - 1.7.1.10)
Diagnosis - interrupt in case of intervention of phase shift adaptation
(2)Adaptation running?
(2)else
Yes
- Interrupt of diagnosis - Reset action (1.7.1.1.1 - 1.7.1.1.8)
End of update of diagnosis window - output of cylinder individual lambda control concerned (1.7.1)


Update of diagnosis window - output of cylinder balancing via engine roughness concerned (1.7.2)
(1)Is the condition for the diagnosis of cylinder balancing via engine roughness fulfilled?
No action
Yes
(2)Error symptom detected and the value of timer for update of diagnosis window zero?
(2)else
Condition fulfilled
Set the timer for the diagnosis window to zero (1.7.2.1)
(2)Error symptom detected and the value of timer for update of diagnosis window not zero?
(2)else
Condition fulfilled
Increment the timer for update of diagnosis window (1.7.2.2)
Increment the timer for the diagnosis window (1.7.2.3)
Continued on next page
End of update of diagnosis window - output of cylinder balancing via engine roughness concerned (1.7.2)


Continued from previous page
(2)Has the timer of diagnosis cycle for similar conditions window reached the maximum value?
(2)else
Yes
(3)No error flag has been set?
- End of first diagnosis cycle (1.7.2.4)
- End of diagnosis cycle for similar conditions window (1.7.2.5)
- Reset the timer of diagnosis cycle for similar conditions window (1.7.2.6)
LV_ERR_CYL_BAL_ER[x] = 0
Reset the timer of diagnosis cycle for similar conditions window (1.7.2.7)
(2)Has the timer for update of diagnosis window reached the maximum value and no error flag set?
(2)else
Condition fulfilled
Reset all of the timer (1.7.2.8 - 1.7.2.10)
End of update of diagnosis window - output of cylinder individual lambda control concerned (1.7.2)

Application conditions:

Initialization:

at reset, LV_IGK = 0 → 1 and clear error memory:
LV_CDN_DIAG_CYL_BAL_LAM[x] = 0
LV_CDN_DIAG_CYL_BAL_ER[x] = 0
ERR_SYM_CYL_BAL_LAM[x] = 0
ERR_SYM_CYL_BAL_ER[x] = 0
LV_ERR_CYL_BAL_LAM[x] = 0
LV_ERR_CYL_BAL_ER[x] = 0
T_SUM_CYL_BAL_LAM_LIM_MIN[x] = 0
T_SUM_CYL_BAL_LAM_LIM_MAX[x] = 0
T_SUM_CYL_BAL_ER_LIM_MIN[x] = 0
T_SUM_CYL_BAL_ER_LIM_MAX[x] = 0
T_SUM_RST_CYL_BAL_LAM[x] = 0
T_SUM_RST_CYL_BAL_ER[x] = 0
T_SUM_END_DIAG_WIN_CYL_LAM[x] = 0
T_SUM_END_DIAG_WIN_CYL_BAL_ER[x] = 0
LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 0
LV_INH_OBD_DIAG_CYL_BAL_ER = 0
LV_ERR_REF_CYL_BAL_LAM[i] = 0
LV_ERR_REF_CYL_BAL_ER = 0

At E²PROM – failure:
CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] = 0
CTR_ERR_OBD_DIAG_CYL_BAL_ER = 0

Recurrence:

100 ms

Activation:

LV_ST_END = 1

Deactivation:

-

Formula section:

4.6.1 Reset action using reset switch or service tool intervention

if[(1)] (LC_CTR_ERR_OBD_DIAG_CYL_BAL = 1 **or** LV_CTR_CYL_BAL_RST_EXT = 1)
then

CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] = 0 **(4.6.1.1)**

CTR_ERR_OBD_DIAG_CYL_BAL_ER = 0 **(4.6.1.2)**

LV_ERR_REF_CYL_BAL_LAM[i] = 0 **(4.6.1.3)**

LV_ERR_REF_CYL_BAL_ER = 0

```
                                                                        (4.6.1.4)
endif(1)
```

4.6.2 Decrement action in case of no failure detected

```
if(1) LV_DC 1 → 0
then
    if(2)  (LV_ERR_CYL_BAL_LAM[x] = 0 and
                                        CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] > 0)
           [Remark: the cylinder selective error flag shall be assigned to the corresponding
                                                  exhaust bank (index x -> index i)]
    then
           CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] =
                                          CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] –
                                          C_CTR_DEC_ERR_OBD_DIAG_CYL_BAL
             [Remark: the counter shall be decremented only once per recurrence] (4.6.2.1)
    endif(2)

    if(2)  (LV_ERR_CYL_BAL_ER[x] = 0 and
                                          CTR_ERR_OBD_DIAG_CYL_BAL_ER > 0)
    then
           CTR_ERR_OBD_DIAG_CYL_BAL_ER =
                                            CTR_ERR_OBD_DIAG_CYL_BAL_ER –
                                          C_CTR_DEC_ERR_OBD_DIAG_CYL_BAL
             [Remark: the counter shall be decremented only once per recurrence] (4.6.2.2)
    endif(2)

endif(1)
```

4.6.3 Inhibition due to balancing error detected

4.6.3.1 Inhibition of cylinder individual lambda control

```
if(1) CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] ≥ C_CTR_MAX_CYL_BAL_LAM_ERR
then
    LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 1
                                                                        (4.6.3.1.1)
else(1)
    LV_INH_OBD_DIAG_CYL_BAL_LAM[i] = 0
                                                                        (4.6.3.1.2)
endif(1)
```

4.6.3.2 Inhibition of CYBL_ER

```
if(1) CTR_ERR_OBD_DIAG_CYL_BAL_ER ≥ C_CTR_MAX_CYL_BAL_ER_ERR
then
    LV_INH_OBD_DIAG_CYL_BAL_ER = 1
```

```
                                                                        (4.6.3.2.1)
else(1)
    LV_INH_OBD_DIAG_CYL_BAL_ER = 0
                                                                        (4.6.3.2.2)
endif(1)
```

4.6.4 Diagnosis condition

4.6.4.1 Diagnosis condition - output of cylinder individual lambda control concerned

```
if(1) (STATE_LAM_CYL_SEL_ADJ[i] = 4 ' ADJ_NOM_L_RNG' or
                         STATE_LAM_CYL_SEL_ADJ[i] = 5 ' ADJ_NOM_H_RNG')
          [Remark: the exhaust bank based calculation shall be executed for all of the
        exhaust-bank-related cylinders (index i -> index x) - (4.6.4.1.1) and (4.6.4.1.3)]
then
    LV_CDN_DIAG_CYL_BAL_LAM[x] = 1
      [Remark: the calculation shall be executed for all of the exhaust-bank-related cylinders
                                                          (index i -> index x)] (4.6.4.1.1)
    LV_CDN_DIAG_CYL_BAL_ER[x] = 0
      [Remark: the calculation shall be executed for all of the cylinders (index x)] (4.6.4.1.2)
else(1)
    LV_CDN_DIAG_CYL_BAL_LAM[x] = 0
          [Remark: the calculation shall be executed for all of the exhaust-bank-related
                                                cylinders (index i -> index x)] (4.6.4.1.3)
endif(1)
```

4.6.4.2 Diagnosis condition - output of CYBL_ER concerned

```
if(1) STATE_CTL_TI_ER_BAL ≠ 0
then
    LV_CDN_DIAG_CYL_BAL_ER[x] = 1
      [Remark: the calculation shall be executed for all of the cylinders (index x)] (4.6.4.2.1)

    LV_CDN_DIAG_CYL_BAL_LAM[x] = 0
      [Remark: the calculation shall be executed for all of the cylinders (index x)] (4.6.4.2.2)

else(1)
    LV_CDN_DIAG_CYL_BAL_ER[x] = 0
      [Remark: the calculation shall be executed for all of the cylinders (index x)] (4.6.4.2.3)
endif(1)
```

4.6.5 Diagnosis self-shutoff

```
if(1) LV_ERR_CYL_BAL_LAM[x] = 1
then
    LV_CDN_DIAG_CYL_BAL_LAM[x] = 0
          [Remark: the calculation shall be executed for all of the exhaust-bank-related
                                                 cylinders (index i -> index x)] (4.6.5.1)
endif(1)
```

```
if(1) LV_ERR_CYL_BAL_ER[x] = 1
then
    LV_CDN_DIAG_CYL_BAL_ER[x] = 0
        [Remark: the calculation shall be executed for all of the cylinders (index x)] (4.6.5.2)
endif(1)
```

4.6.6 Error detection

4.6.6.1 Error detection in the area where the cylinder individual lambda adaptation is concerned

```
if(1) LV_CDN_DIAG_CYL_BAL_LAM[x] = 1
then
    if(2)   STATE_LAM_CYL_SEL_ADJ[i] = 4 ' ADJ_NOM_L_RNG'
                [Remark: the exhaust bank based calculation shall be executed for all of the
                exhaust-bank-related cylinders (index i -> index x) - (4.6.6.1.1) - (4.6.6.1.14)]
    then
        if(3)   FAC_LAM_CYL_SEL_ADJ_L_RNG[x] <
                                        C_FAC_LAM_CYL_SEL_ADJ_DIAG_MIN
        then
            increment (T_SUM_CYL_BAL_LAM_LIM_MIN[x])
                                                            (4.6.6.1.1)
            ERR_SYM_CYL_BAL_LAM[x] = 1 'SYM_0'
                                                            (4.6.6.1.2)
            if(4)   T_SUM_CYL_BAL_LAM_LIM_MIN[x] >
                                        C_T_SUM_THD_CYL_BAL_LIM
            then
                if(5)   LV_ERR_REF_CYL_BAL_LAM[i] = 0
                then
                    CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] =
                                CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] +
                                 C_CTR_INC_ERR_OBD_DIAG_CYL_BAL
                                                            (4.6.6.1.3)
                    LV_ERR_REF_CYL_BAL_LAM[i] = 1
                                                            (4.6.6.1.4)
                endif(5)
                for(5)  j(x) = 1 : Number of cylinders per exhaust bank
                    if(6)   [(T_SUM_CYL_BAL_LAM_LIM_MIN[x] >
                                        C_T_SUM_MIN_CYL_BAL_LIM) or
                                  (T_SUM_CYL_BAL_LAM_LIM_MAX[x] >
                                        C_T_SUM_MIN_CYL_BAL_LIM)]
                    then
                        LV_ERR_CYL_BAL_LAM[x] = 1
                                                            (4.6.6.1.5)
                        LV_END_DIAG_CYL_BAL_LAM[x] = 1
                                                            (4.6.6.1.6)
                        LV_END_DIAG_WIN_CYL_BAL_LAM[x] = 1
                                                            (4.6.6.1.7)
                    endif(6)
```

```
            endfor$^{(5)}$
        endif$^{(4)}$
    else$^{(3)}$
        if$^{(4)}$    FAC_LAM_CYL_SEL_ADJ_L_RNG[x] >
                                  C_FAC_LAM_CYL_SEL_ADJ_DIAG_MAX
        then
            increment (T_SUM_CYL_BAL_LAM_LIM_MAX[x])
                                                      (4.6.6.1.8)
            ERR_SYM_CYL_BAL_LAM[x] = 2 'SYM_1'
                                                      (4.6.6.1.9)
            if$^{(5)}$    T_SUM_CYL_BAL_LAM_LIM_MAX[x] >
                                      C_T_SUM_THD_CYL_BAL_LIM
            then
                if$^{(6)}$    LV_ERR_REF_CYL_BAL_LAM[i] = 0
                then
                    CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] =
                           CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] +
                            C_CTR_INC_ERR_OBD_DIAG_CYL_BAL
                                                      (4.6.6.1.10)
                    LV_ERR_REF_CYL_BAL_LAM[i] = 1
                                                      (4.6.6.1.11)
                endif$^{(6)}$
                for$^{(6)}$   j(x) = 1 : Number of cylinders per exhaust bank
                    if$^{(7)}$    [(T_SUM_CYL_BAL_LAM_LIM_MIN[x] >
                                   C_T_SUM_MIN_CYL_BAL_LIM) or
                                 (T_SUM_CYL_BAL_LAM_LIM_MAX[x] >
                                     C_T_SUM_MIN_CYL_BAL_LIM)]
                    then
                        LV_ERR_CYL_BAL_LAM[x] = 1
                                                      (4.6.6.1.12)
                        LV_END_DIAG_CYL_BAL_LAM[x] = 1
                                                      (4.6.6.1.13)
                        LV_END_DIAG_WIN_CYL_BAL_LAM[x] = 1
                                                      (4.6.6.1.14)
                    endif$^{(7)}$
                endfor$^{(6)}$
            endif$^{(5)}$
        endif$^{(4)}$
    endif$^{(3)}$
else$^{(2)}$
    if$^{(3)}$    STATE_LAM_CYL_SEL_ADJ[i] = 5 ' ADJ_NOM_H_RNG'
          [Remark: the exhaust bank based calculation shall be executed for all of the
        exhaust-bank-related cylinders (index i -> index x) - (4.6.6.1.15) - (4.6.6.1.28)]
    then
        if$^{(4)}$    FAC_LAM_CYL_SEL_ADJ_H_RNG[x] <
                                  C_FAC_LAM_CYL_SEL_ADJ_DIAG_MIN
        then
            increment (T_SUM_CYL_BAL_LAM_LIM_MIN[x])
```

```
                                                            (4.6.6.1.15)
    ERR_SYM_CYL_BAL_LAM[x] = 1 'SYM_0'
                                                            (4.6.6.1.16)
    if$^{(5)}$  T_SUM_CYL_BAL_LAM_LIM_MIN[x] >
                                        C_T_SUM_THD_CYL_BAL_LIM
    then
        if$^{(6)}$  LV_ERR_REF_CYL_BAL_LAM[i] = 0
        then
            CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] =
                    CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] +
                      C_CTR_INC_ERR_OBD_DIAG_CYL_BAL
                                                            (4.6.6.1.17)
            LV_ERR_REF_CYL_BAL_LAM[i] = 1
                                                            (4.6.6.1.18)
        endif$^{(6)}$
        for$^{(6)}$  j(x) = 1 : Number of cylinders per exhaust bank
            if$^{(7)}$  [(T_SUM_CYL_BAL_LAM_LIM_MIN[x] >
                             C_T_SUM_MIN_CYL_BAL_LIM) or
                      (T_SUM_CYL_BAL_LAM_LIM_MAX[x] >
                               C_T_SUM_MIN_CYL_BAL_LIM)]
            then
                LV_ERR_CYL_BAL_LAM[x] = 1
                                                            (4.6.6.1.19)
                LV_END_DIAG_CYL_BAL_LAM[x] = 1
                                                            (4.6.6.1.20)
                LV_END_DIAG_WIN_CYL_BAL_LAM[x] = 1
                                                            (4.6.6.1.21)
            endif$^{(7)}$
        endfor$^{(6)}$
    endif$^{(5)}$
else$^{(4)}$
    if$^{(5)}$  FAC_LAM_CYL_SEL_ADJ_H_RNG[x] >
                                    C_FAC_LAM_CYL_SEL_ADJ_DIAG_MAX
    then
        increment (T_SUM_CYL_BAL_LAM_LIM_MAX[x])
                                                            (4.6.6.1.22)
        ERR_SYM_CYL_BAL_LAM[x] = 2 'SYM_1'
                                                            (4.6.6.1.23)
        if$^{(6)}$  T_SUM_CYL_BAL_LAM_LIM_MAX[x] >
                                        C_T_SUM_THD_CYL_BAL_LIM
        then
            if$^{(7)}$  LV_ERR_REF_CYL_BAL_LAM[i] = 0
            then
                CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] =
                    CTR_ERR_OBD_DIAG_CYL_BAL_LAM[i] +
                      C_CTR_INC_ERR_OBD_DIAG_CYL_BAL
                                                            (4.6.6.1.24)
                LV_ERR_REF_CYL_BAL_LAM[i] = 1
                                                            (4.6.6.1.25)
```

```
            endif(7)
            for(7)    j(x) = 1 : Number of cylinders per exhaust bank
              if(8)[(T_SUM_CYL_BAL_LAM_LIM_MIN[x] >
                      C_T_SUM_MIN_CYL_BAL_LIM) or
                    (T_SUM_CYL_BAL_LAM_LIM_MAX[x] >
                      C_T_SUM_MIN_CYL_BAL_LIM)]
              then
                LV_ERR_CYL_BAL_LAM[x] = 1                 (4.6.6.1.26)
                LV_END_DIAG_CYL_BAL_LAM[x] = 1            (4.6.6.1.27)
                LV_END_DIAG_WIN_CYL_BAL_LAM[x] = 1        (4.6.6.1.28)
              endif(8)
            endfor(7)
          endif(6)
        endif(5)
      endif(4)
    endif(3)
  endif(2)
endif(1)
```

4.6.6.2 Error detection in the area where output of CYBL_ER is concerned

```
if(1) LV_CDN_DIAG_CYL_BAL_ER[x] = 1
              [Remark: the calculation shall be executed for all of the cylinders (index x) –
                                                          (4.6.6.2.1) - (4.6.6.2.28)]
then
    if(2)  (STATE_CTL_TI_ER_BAL = 1 or STATE_CTL_TI_ER_BAL = 2)
    then
        if(3)  LV_MFF_ADD_ER_BAL_OBD_MAX_NEG[x] = 1
        then
            increment (T_SUM_CYL_BAL_ER_LIM_MIN[x])
                                                          (4.6.6.2.1)
            ERR_SYM_CYL_BAL_ER[x] = 1 'SYM_0'
                                                          (4.6.6.2.2)
            if(4)  T_SUM_CYL_BAL_ER_LIM_MIN[x] >
                                        C_T_SUM_THD_CYL_BAL_LIM
            then
                if(5)  LV_ERR_REF_CYL_BAL_ER = 0
                then
                    CTR_ERR_OBD_DIAG_CYL_BAL_ER =
                                    CTR_ERR_OBD_DIAG_CYL_BAL_ER +
                                    C_CTR_INC_ERR_OBD_DIAG_CYL_BAL
                                                          (4.6.6.2.3)
                    LV_ERR_REF_CYL_BAL_ER = 1
                                                          (4.6.6.2.4)
                endif(5)
                for(5)  j(x) = 1 : Number of cylinders
                    if(6)  [(T_SUM_CYL_BAL_ER_LIM_MIN[x] >
                                        C_T_SUM_MIN_CYL_BAL_LIM) or
                            (T_SUM_CYL_BAL_ER_LIM_MAX[x] >
                                        C_T_SUM_MIN_CYL_BAL_LIM)]
                    then
                        LV_ERR_CYL_BAL_ER[x] = 1
                                                          (4.6.6.2.5)
                        LV_END_DIAG_CYL_BAL_ER[x] = 1
                                                          (4.6.6.2.6)
                        LV_END_DIAG_WIN_CYL_BAL_ER[x] = 1
                                                          (4.6.6.2.7)
                    endif(6)
                endfor(5)
            endif(4)
        else(3)
            if(4)  LV_MFF_ADD_ER_BAL_OBD_MAX_POS[x] = 1
            then
                increment (T_SUM_CYL_BAL_ER_LIM_MAX[x])
                                                          (4.6.6.2.8)
                ERR_SYM_CYL_BAL_ER[x] = 2 'SYM_1'
                                                          (4.6.6.2.9)
```

```
                    if(5)  T_SUM_CYL_BAL_ER_LIM_MAX[x] >
                                        C_T_SUM_THD_CYL_BAL_LIM
                    then
                        if(6)  LV_ERR_REF_CYL_BAL_ER = 0
                        then
                            CTR_ERR_OBD_DIAG_CYL_BAL_ER =
                                        CTR_ERR_OBD_DIAG_CYL_BAL_ER +
                                        C_CTR_INC_ERR_OBD_DIAG_CYL_BAL
                                                                (4.6.6.2.10)
                            LV_ERR_REF_CYL_BAL_ER = 1
                                                                (4.6.6.2.11)
                        endif(6)
                        for(6)  j(x) = 1 : Number of cylinders
                            if(7)  [(T_SUM_CYL_BAL_ER_LIM_MIN[x] >
                                            C_T_SUM_MIN_CYL_BAL_LIM) or
                                        (T_SUM_CYL_BAL_ER_LIM_MAX[x] >
                                            C_T_SUM_MIN_CYL_BAL_LIM)]
                            then
                                LV_ERR_CYL_BAL_ER[x] = 1
                                                                (4.6.6.2.12)
                                LV_END_DIAG_CYL_BAL_ER[x] = 1
                                                                (4.6.6.2.13)
                                LV_END_DIAG_WIN_CYL_BAL_ER[x] = 1
                                                                (4.6.6.2.14)
                            endif(7)
                        endfor(6)
                    endif(5)
                endif(4)
            endif(3)
        else(2)
            if(3)  (STATE_CTL_TI_ER_BAL = 3 or STATE_CTL_TI_ER_BAL = 4)
            then
                if(4)  LV_MFF_FAC_ER_BAL_OBD_MAX_NEG[x] = 1
                then
                    increment (T_SUM_CYL_BAL_ER_LIM_MIN[x])
                                                                (4.6.6.2.15)
                    ERR_SYM_CYL_BAL_ER[x] = 1 'SYM_0'
                                                                (4.6.6.2.16)
                    if(5)  T_SUM_CYL_BAL_ER_LIM_MIN[x] >
                                        C_T_SUM_THD_CYL_BAL_LIM
                    then
                        if(6)  LV_ERR_REF_CYL_BAL_ER = 0
                        then
                            CTR_ERR_OBD_DIAG_CYL_BAL_ER =
                                        CTR_ERR_OBD_DIAG_CYL_BAL_ER +
                                        C_CTR_INC_ERR_OBD_DIAG_CYL_BAL
                                                                (4.6.6.2.17)
                            LV_ERR_REF_CYL_BAL_ER = 1
```

```
                                                                  (4.6.6.2.18)
                    endif(6)
                    for(6)  j(x) = 1 : Number of cylinders
                            if(7)   [(T_SUM_CYL_BAL_ER_LIM_MIN[x] >
                                          C_T_SUM_MIN_CYL_BAL_LIM) or
                                        (T_SUM_CYL_BAL_ER_LIM_MAX[x] >
                                           C_T_SUM_MIN_CYL_BAL_LIM)]
                            then
                                    LV_ERR_CYL_BAL_ER[x] = 1
                                                                  (4.6.6.2.19)
                                    LV_END_DIAG_CYL_BAL_ER[x] = 1
                                                                  (4.6.6.2.20)
                                    LV_END_DIAG_WIN_CYL_BAL_ER[x] = 1
                                                                  (4.6.6.2.21)
                            endif(7)
                    endfor(6)
            endif(5)
    else(4)
            if(5)   LV_MFF_FAC_ER_BAL_OBD_MAX_POS[x] = 1
            then
                    increment (T_SUM_CYL_BAL_ER_LIM_MAX[x])
                                                                  (4.6.6.2.22)
                    ERR_SYM_CYL_BAL_ER[x] = 2 'SYM_1'
                                                                  (4.6.6.2.23)
                    if(6)   T_SUM_CYL_BAL_ER_LIM_MAX[x] >
                                              C_T_SUM_THD_CYL_BAL_LIM
                    then
                            if(7)   LV_ERR_REF_CYL_BAL_ER = 0
                            then
                                    CTR_ERR_OBD_DIAG_CYL_BAL_ER =
                                        CTR_ERR_OBD_DIAG_CYL_BAL_ER +
                                      C_CTR_INC_ERR_OBD_DIAG_CYL_BAL
                                                                  (4.6.6.2.24)
                                    LV_ERR_REF_CYL_BAL_ER = 1
                                                                  (4.6.6.2.25)
                            endif(7)
                            for(7)  j(x) = 1 : Number of cylinders
                                    if(8)[(T_SUM_CYL_BAL_ER_LIM_MIN[x] >
                                          C_T_SUM_MIN_CYL_BAL_LIM) or
                                        (T_SUM_CYL_BAL_ER_LIM_MAX[x] >
                                           C_T_SUM_MIN_CYL_BAL_LIM)]
                                    then
                                            LV_ERR_CYL_BAL_ER[x] = 1
                                                                  (4.6.6.2.26)
                                            LV_END_DIAG_CYL_BAL_ER[x] = 1
                                                                  (4.6.6.2.27)
                                      LV_END_DIAG_WIN_CYL_BAL_ER[x] = 1
                                                                  (4.6.6.2.28)
```

```
                            endif(8)
                        endfor(7)
                    endif(6)
                endif(5)
            endif(4)
        endif(3)
    endif(2)
endif(1)
```

4.6.7 Update of diagnosis window

4.6.7.1 Update of diagnosis window - output of cylinder individual lambda control concerned

```
if(1) LV_CDN_DIAG_CYL_BAL_LAM[x] = 1
            [Remark: the calculation shall be executed for all of the cylinders (index x) –
                                                      (4.6.7.1.1) - (4.6.7.1.10)]
then
    if(2)   (ERR_SYM_CYL_BAL_LAM[x] > 0 and T_SUM_RST_CYL_BAL_LAM[x] = 0)
    then
            T_SUM_END_DIAG_WIN_CYL_LAM[x] = 0
                                                                  (4.6.7.1.1)
    endif(2)
    if(2)   (ERR_SYM_CYL_BAL_LAM[x] > 0 or T_SUM_RST_CYL_BAL_LAM[x] > 0)
    then
            increment (T_SUM_RST_CYL_BAL_LAM[x])
                                                                  (4.6.7.1.2)
    endif(2)
    increment (T_SUM_END_DIAG_WIN_CYL_LAM[x])
                                                                  (4.6.7.1.3)
    if(2)   T_SUM_END_DIAG_WIN_CYL_LAM[x] = C_T_SUM_RST_MAX_CYL_BAL
    then
            LV_END_DIAG_CYL_BAL_LAM[x] = 1
                                                                  (4.6.7.1.4)
            LV_END_DIAG_WIN_CYL_BAL_LAM[x] = 1
                                                                  (4.6.7.1.5)
            T_SUM_END_DIAG_WIN_CYL_LAM[x] = 0
                                                                  (4.6.7.1.6)
    else(2)
            if(3)   LV_ERR_CYL_BAL_LAM[x] = 0
            then
                    LV_END_DIAG_WIN_CYL_BAL_LAM[x] = 0
                                                                  (4.6.7.1.7)
            endif(3)
    endif(2)
    if(2)   (T_SUM_RST_CYL_BAL_LAM[x] = C_T_SUM_RST_MAX_CYL_BAL and
                                                  LV_ERR_CYL_BAL_LAM[x] = 0)
    then
            T_SUM_CYL_BAL_LAM_LIM_MIN[x] = 0
                                                                  (4.6.7.1.8)
            T_SUM_CYL_BAL_LAM_LIM_MAX[x] = 0
```

```
                                                        (4.6.7.1.9)
        T_SUM_RST_CYL_BAL_LAM[x] = 0
                                                        (4.6.7.1.10)
    endif(2)
```

4.6.7.1.1 Diagnosis- interrupt in case of intervention of phase shift adaptation

```
    if(2)   (STATE_DELTA_CRK_CYL_LAM[i] = 3 'AD_INI' or
                                      STATE_DELTA_CRK_CYL_LAM[i] = 4 'AD_DEC' or
                                        STATE_DELTA_CRK_CYL_LAM[i] = 5 'AD_INC')
                       [Remark: the exhaust bank based calculation shall be executed for all of the
                 exhaust-bank-related cylinders (index i -> index x) - (4.6.7.1.1.1) - (4.6.7.1.1.8)]
    then
            T_SUM_CYL_BAL_LAM_LIM_MIN[x] = 0
                                                                               (4.6.7.1.1.1)
            T_SUM_CYL_BAL_LAM_LIM_MAX[x] = 0
                                                                               (4.6.7.1.1.2)
            T_SUM_RST_CYL_BAL_LAM[x] = 0
                                                                               (4.6.7.1.1.3)
            T_SUM_END_DIAG_WIN_CYL_LAM[x] = 0
                                                                               (4.6.7.1.1.4)
            LV_END_DIAG_CYL_BAL_LAM[x] = 0
                                                                               (4.6.7.1.1.5)
            LV_END_DIAG_WIN_CYL_BAL_LAM[x] = 0
                                                                               (4.6.7.1.1.6)
            LV_ERR_CYL_BAL_LAM[x] = 0
                                                                               (4.6.7.1.1.7)
            ERR_SYM_CYL_BAL_LAM[x] = 0 'NO_SYM'
                                                                               (4.6.7.1.1.8)
    endif(2)
endif(1)
```

4.6.7.2 Update of diagnosis window - output of CYBL_ER concerned

```
if$^{(1)}$ LV_CDN_DIAG_CYL_BAL_ER[x] = 1
                    [Remark: the calculation shall be executed for all of the cylinders (index x) –
                                                                    (4.6.7.2.1) - (4.6.7.2.10)]
then
    if$^{(2)}$  (ERR_SYM_CYL_BAL_ER[x] > 0 and T_SUM_RST_CYL_BAL_ER[x] = 0)
    then
            T_SUM_END_DIAG_WIN_CYL_BAL_ER[x] = 0
                                                                                  (4.6.7.2.1)
    endif$^{(2)}$
    if$^{(2)}$  (ERR_SYM_CYL_BAL_ER[x] > 0 or T_SUM_RST_CYL_BAL_ER[x] > 0)
    then
            increment (T_SUM_RST_CYL_BAL_ER[x])
                                                                                  (4.6.7.2.2)
    endif$^{(2)}$
    increment (T_SUM_END_DIAG_WIN_CYL_BAL_ER[x])
                                                                                  (4.6.7.2.3)
    if$^{(2)}$  T_SUM_END_DIAG_WIN_CYL_BAL_ER[x] = C_T_SUM_RST_MAX_CYL_BAL
    then
            LV_END_DIAG_CYL_BAL_ER[x] = 1
                                                                                  (4.6.7.2.4)
            LV_END_DIAG_WIN_CYL_BAL_ER[x] = 1
                                                                                  (4.6.7.2.5)
            T_SUM_END_DIAG_WIN_CYL_BAL_ER[x] = 0
                                                                                  (4.6.7.2.6)
    else$^{(2)}$
            if$^{(3)}$  LV_ERR_CYL_BAL_ER[x] = 0
            then
                    LV_END_DIAG_WIN_CYL_BAL_ER[x] = 0
                                                                                  (4.6.7.2.7)
            endif$^{(3)}$
    endif$^{(2)}$
    if$^{(2)}$  (T_SUM_RST_CYL_BAL_ER[x] = C_T_SUM_RST_MAX_CYL_BAL and
                                                            LV_ERR_CYL_BAL_ER[x] = 0)
    then
            T_SUM_CYL_BAL_ER_LIM_MIN[x] = 0
                                                                                  (4.6.7.2.8)
            T_SUM_CYL_BAL_ER_LIM_MAX[x] = 0
                                                                                  (4.6.7.2.9)
            T_SUM_RST_CYL_BAL_ER[x] = 0
                                                                                 (4.6.7.2.10)
    endif$^{(2)}$
endif$^{(1)}$
```

Configuration for diagnostic symptoms:

Diagnostic CYL_BAL_LAM[x]	Symptom description	Symptom	Filter type
Detected failure symptom - output of CILC concerned	*minimum diagnosis limit of cylinder individual lambda control reached*	*SYM_0*	*NO*
	maximum diagnosis limit of cylinder individual lambda control reached	*SYM_1*	
	not used	*SYM_2*	
	not used	*SYM_3*	

Diagnostic CYL_BAL_ER[x]	Symptom description	Symptom	Filter type
Detected failure symptom - output of CYBL_ER concerned	*minimum diagnosis limit of CYBL_ER reached*	*SYM_0*	*NO*
	maximum diagnosis limit of CYBL_ER reached	*SYM_1*	
	not used	*SYM_2*	
	not used	*SYM_3*	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_LAM_CYL_SEL_ADJ_DIAG_MIN	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lower diagnosis limit of the cylinder individual nominal-adaptation correction factor					
C_FAC_LAM_CYL_SEL_ADJ_DIAG_MAX	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper diagnosis limit of the cylinder individual nominal-adaptation correction factor					
C_T_SUM_THD_CYL_BAL_LIM	1	1...FFFFH	0.1...6553.5	0.1	[s]
Time threshold to set the error flag					
C_T_SUM_MIN_CYL_BAL_LIM	1	1...FFFFH	0.1...6553.5	0.1	[s]
Minimum run-off time for error detection					
C_T_SUM_RST_MAX_CYL_BAL	1	1...FFFFH	0.1...6553.5	0.1	[s]
Diagnosis time interval					
C_CTR_INC_ERR_OBD_DIAG_CYL_BAL	1	0...FFH	0...255	1	[-]
Increment constant of anti-bounce-counter					
C_CTR_DEC_ERR_OBD_DIAG_CYL_BAL	1	0...FFH	0...255	1	[-]
Decrement constant of anti-bounce-counter					
C_CTR_MAX_CYL_BAL_LAM_ERR	1	1...FFH	1...255	1	[-]
Threshold of anti-bounce-counter to set the inhibit flag for cylinder individual lambda control					
C_CTR_MAX_CYL_BAL_ER_ERR	1	1...FFH	1...255	1	[-]
Threshold of anti-bounce-counter to set the inhibit flag for CYBL_ER					
LC_CTR_ERR_OBD_DIAG_CYL_BAL	1	0...1H	0...1	1	[-]
Anti-bounce-counter reset					

5 Car body

Table of Contents

Index

5.1 E-box cooling fan

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_EBOX_CFA	V/O	0...1H	0...1	1	[-]
Control bit EBOX cooling fan					

Input data:

TECU	TIA	LV_VAR_EBOX_CFA	LV_ERR_EBOX_CFA
LV_EBOX_CFA_EXT_ADJ	LV_ACT_EBOX_CFA_EXT_ADJ	LV_END_DIAG_EBOX_CFA	N
LV_IGK			

General information:

The function should prevent a overheating of the ECUs in the EBOX. For that, a cooling fan is controlled in dependency of the ECU temperature and the intake air temperature.

Application conditions:

Initialisation: LV_EBOX_CFA = 0 ,

T_MAX_EBOX_CFA_ACT = 0 at transition LV_IGK 0 -->1

LV_EBOX_CFA_ACT_TEST_END = 0 at transition LV_IGK 0 -->1

Recurrence: 1000 ms

Activation: at every engine operating states and

LV_VAR_EBOX_CFA = 1

FUNCTION DESCRIPTION:

Description:

There are two operating states for the activation of the Ebox-cooling-fan:
1) for power stage diagnosis:
the Ebox-cooling-fan has to be briefly activated at every transition LV_IGK 0 --> 1 in order to be able to perform a complete diagnosis cycle. After LV_IGK 0 --> 1 and the exceeding of a calibratable N-threshold the Ebox-CFA is activated either until LV_END_DIAG_EBOX_CFA is set to 1 (diagnosis completed) or the maximum time C_T_MAX_EBOX_CFA_ACT has been reached.
2) normal operation:
activation via external adjustment or due to cooling request because of high TIA or TECU

Formula section:

```
// Activation of Ebox-cooling-fan - for diagnosis (at every transition LV  IGK 0 --> 1):
If (1)      (LV_IGK = 0 --> 1 or  LV_IGK = 1)  and
            N > C_N_THD_EBOX_CFA_ACT    and
            LV_EBOX_CFA_ACT_TEST_END = 0
then (1)    if (2)        (T_MAX_EBOX_CFA_ACT < C_T_MAX_EBOX_CFA_ACT)  and
                          LV_END_DIAG_EBOX_CFA <> 1
            then (2)      T_MAX_EBOX_CFA_ACT = ++
                          LV_EBOX_CFA = 1
            else (2)  if (3)        LV_END_DIAG_EBOX_CFA = 1     or
                              T_MAX_EBOX_CFA_ACT = C_T_MAX_EBOX_CFA_ACT
                      then (3)  LV_EBOX_CFA = 0
                                LV_EBOX_CFA_ACT_TEST_END = 1
                      endif (3)
            endif (2)
else (1)
// Activation of Ebox-cooling-fan – normal operation:
             If (4)        (LV_IGK = 0 --> 1 or  LV_IGK = 1)        and
                           N > C_N_THD_EBOX_CFA_ACT               and
                           LV_EBOX_CFA_ACT_TEST_END = 1           and
                           LV_ERR_EBOX_CFA = 0
             then (4)  if (5)    LV_EBOX_CFA_EXT_ADJ = 1
                       Then (5)  LV_EBOX_CFA = LV_ACT_EBOX_CFA_EXT_ADJ
                       Else (5) If (6)    TECU > C_TECU_THD_EBOX_CFA        and
                                          TIA > C_TIA_THD_EBOX_CFA
                                then (6)  LV_EBOX_CFA = 1
                                else (6)  if (7)        TECU > C_TECU_THD_EBOX_CFA
                                                        – C_TECU_HYS_EBOX_CFA    and
                                                        TIA > C_TIA_THD_EBOX_CFA –
                                                        C_TIA_HYS_EBOX_CFA         and
                                                        LV_EBOX_CFA = 1
                                          then (7)      LV_EBOX_CFA = 1
                                          else (7)      LV_EBOX_CFA = 0
                                          endif (7)
                               endif (6)
                       endif (5)
```

```
            else (4)        LV_EBOX_CFA = 0
            endif (4)
endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TECU_THD_EBOX_CFA	1	0...FEH	-48...142.5	0.75	[°C]
TECU threshold for EBOX cooling fan ON					
C_TIA_THD_EBOX_CFA	1	0...FEH	-48...142.5	0.75	[°C]
TIA threshold for EBOX cooling fan ON					
C_TECU_HYS_EBOX_CFA	1	0...FEH	0...190.5	0.75	[°C]
TECU hysteresis for EBOX cooling fan					
C_TIA_HYS_EBOX_CFA	1	0...FEH	0...190.5	0.75	[°C]
TIA hysteresis for EBOX cooling fan					
C_T_MAX_EBOX_CFA_ACT	1	0...FFH	0...255	1	[s]
Maximum time for activation of EBOX_CFA after Kl.15 on (for diagnosis)					
C_N_THD_EBOX_CFA_ACT	1	0...1FE0H	0...8160	1	[rpm]
N threshold for activation of EBOX_CFA after Kl.15 on (For diagnosis)					

6 Cylinder balancing

Table of Contents

Index

6.1 AGGR CYBL adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TI_ADD[NC_CYL_NR]	V/O	8000...7FFFH	-32.768...32.767	0.001	[ms]
Total additive correction value for cylinder balancing					
TI_FAC[NC_CYL_NR]	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
injection time correction due to external demand (interface for INJR AGGR)					
LV_LAM_CYL_SEL_ADJ_OFS_EXT_REQ	V/O	0...1H	0...1	1	[-]
External request for calculation of cylinder individual lambda adaptation in the additive area					

Input data:

CRK_CYL_LAM_x	LV_LAM_CYL_ENA_i	FAC_TI_BAL[NC_CYL_NR]	NC_CYL_NR
NC_CBK_EX_NR			

Application conditions:

Initialization at Reset:

```
LV_LAM_CYL_SEL_ADJ_OFS_EXT_REQ = 0

FOR m = 0 to NC_CYL_NR - 1
    TI_ADD[m] = 0
    TI_FAC[m] = 1
END FOR
```

Recurrence: every TDC

Activation: every engine state

Deactivation: -

Formula section:

```
FOR m = 0 to NC_CYL_NR - 1
    TI_FAC[m] = FAC_TI_BAL[m]
END FOR
```

6.2 Engine roughness signal preparation for cylinder balancing

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ER_STND_MMV_BAL[NC_CYL_NR]	O/V/S	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Filtered normalized cylinder selective engine roughness values					
ER_STD_MMV_BAL[NC_CYL_NR]	O/V/S	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Filtered standard deviation of the engine roughness					
ER_STND_MMV_DIF_BAL[NC_CYL_NR]	O/V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Difference of filtered normalized and average engine roughness value					
LV_DRV1_STND_BAL_FDOUT	O/V	0...1H	0...1	1	-
Flag for acceleration fade out of cylinder balancing					
ER_STND_MMV_STD_BAL[NC_CYL_NR]	O/V/S	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Filtered normalized engine roughness values for calculation of the standard deviation					
ER_STND_MMV_MV_BAL	V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Mean value of all engine roughness values of the last [NC_CYL_NR] segments					
ER_STD_DIF_BAL	V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Difference of the filtered normalized and normalized engine roughness values for calc. of the standard deviation					
ER_STND_FIL_BAL	V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Normalized cylinder selective engine roughness values after median filter					
CTR_DLY_DRV1_STND_BAL	V	0...FFFFH	0...6.5535E+4	1	-
Segment delay counter for acceleration fade out of cylinder balancing					
ER_STD_BAL[NC_CYL_NR]	V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Standard deviation of the engine roughness					
ER_STND_BAL	V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Normalized cylinder selective engine roughness values					
DRV1_STND_BAL	V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Standardized acceleration component of the ER value					
DLY_DRV1_STND_BAL	V	0...FFFFH	0...6.5535E+4	1	-
Segment delay for acceleration fade out of cylinder balancing					

Input data:

LV_ER_STND_ER_BAL_ACT	ER	SEG_NR_ER	N_32
LV_DRV1_ER_BAL_ACT	DRV1_ER	NC_CYL_NR	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_ER_STD_BAL	1	0...FFH	0...0.99609307	0.00390625	-
Correlation constant for ER_STND_FIL_BAL value filter for standard deviation					
C_CRLC_ER_STD_MMV_BAL	1	0...FFH	0...0.99609307	0.00390625	-
Correlation constant for ER_STD_BAL value filter					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_ER_STND_BAL	1	0...FFH	0...0.99609307	0.00390625	-
Correlation constant for ER_STND_FIL_BAL value filter					
C_CTR_DLY_DRV1_STND_BAL_DEC	1	0...FFH	0...255	1	-
Segment delay counter decrement value					
C_FAC_ER_STD_NEG_BAL	1	0...FFH	0...0.99609307	0.00390625	-
Weighting factor for negative ER_STD_DIF_BAL values					
C_FAC_ER_STD_POS_BAL	1	0...FFH	0...0.99609307	0.00390625	-
Weighting factor for positive ER_STD_DIF_BAL values					
IP_DLY_DRV1_STND_BAL	8	0...FFFFH	0...6.5535E+4	1	-
LDP_DRV1_STND_BAL_IP_DLY_BAL	8	0...FFFFH	-325.78...325.77	0.0099313	1/s²
Segment delay counter start value					

6.2.1 General information:

Attention - This specification version (402U01) is only valid for software variants which are providing the input variables ER and DRV1_ER as 16 bit values.

The engine roughness signal ER as well as the dynamic part of the engine roughness signal DRV1_ER are used as basis for the calculation of the input signal for the cylinder balancing control functions. The calculations are kept cylinder individual corresponding on the actual segment number SEG_NR_ER.

The module "Engine roughness signal preparation for cylinder balancing" is divided in two separate chapters. Within the first part the calculation of the input signals for the cylinder balancing controller is done, while the second part is providing fade out conditions while cylinder balancing has to be interrupted.

Application Condition

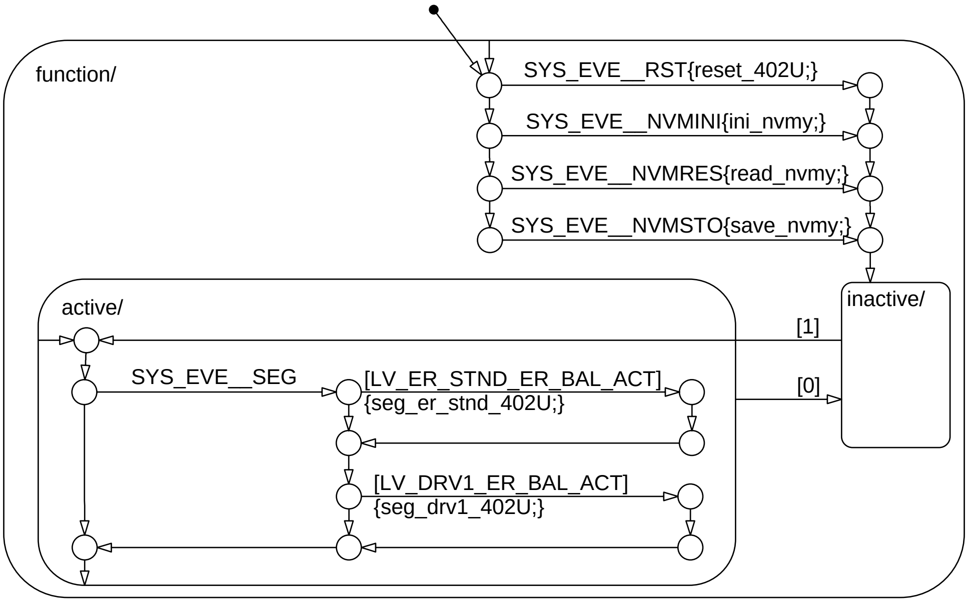


Figure 1 CYBL_SIGNAL/ APP_CDN/ Chart

Function Description

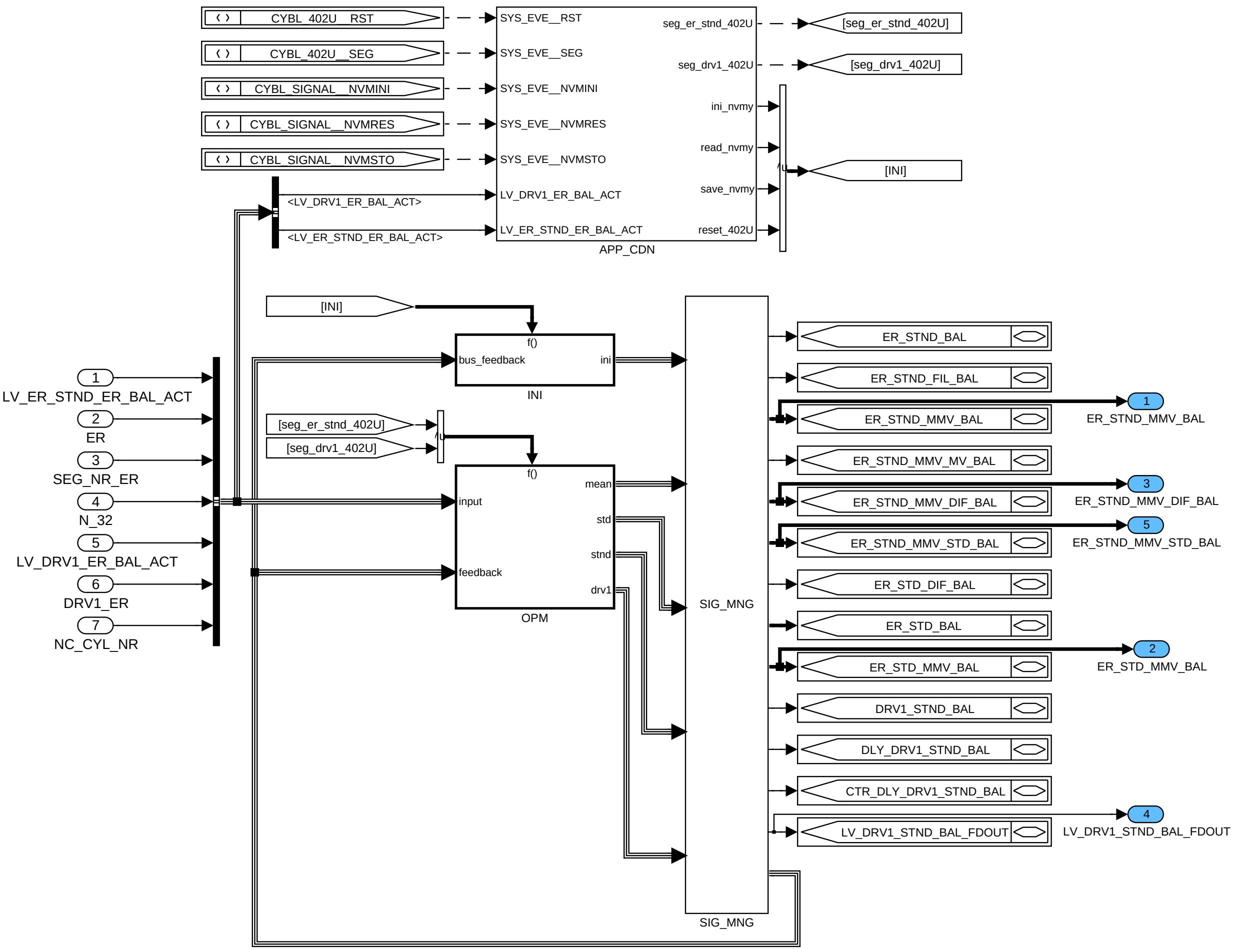


Figure 2 CYBL_SIGNAL

6.2.1.1 SUBFUNCTION: INI

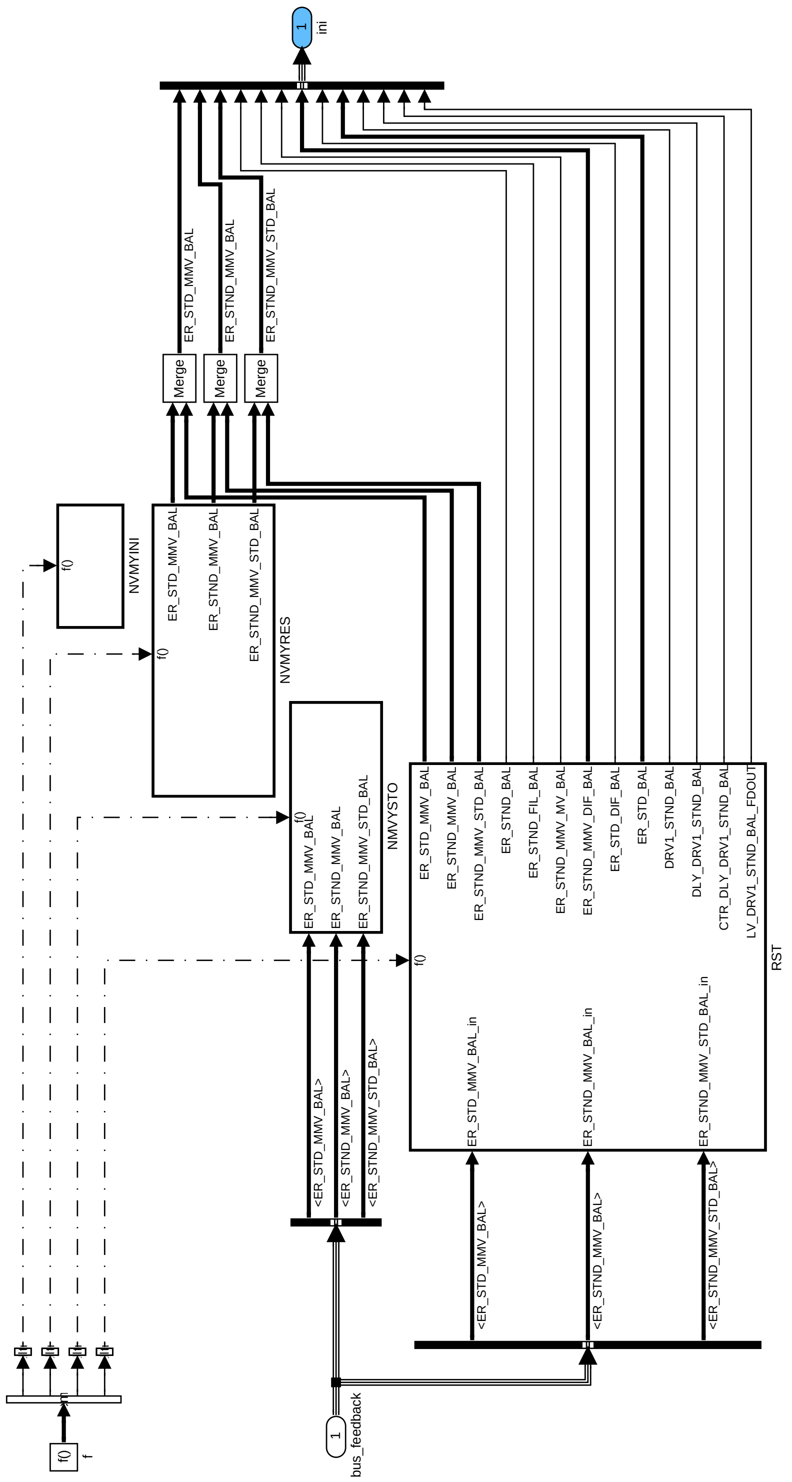


Figure 3 CYBL_SIGNAL/ INI

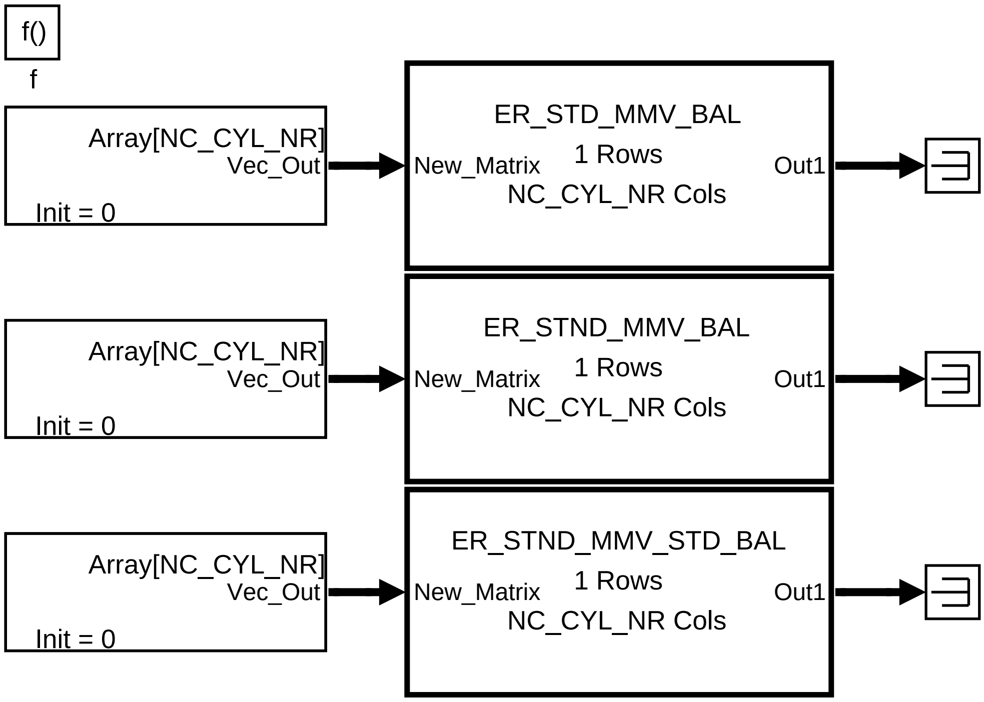


Figure 4 CYBL_SIGNAL/ INI/ NVMYINI

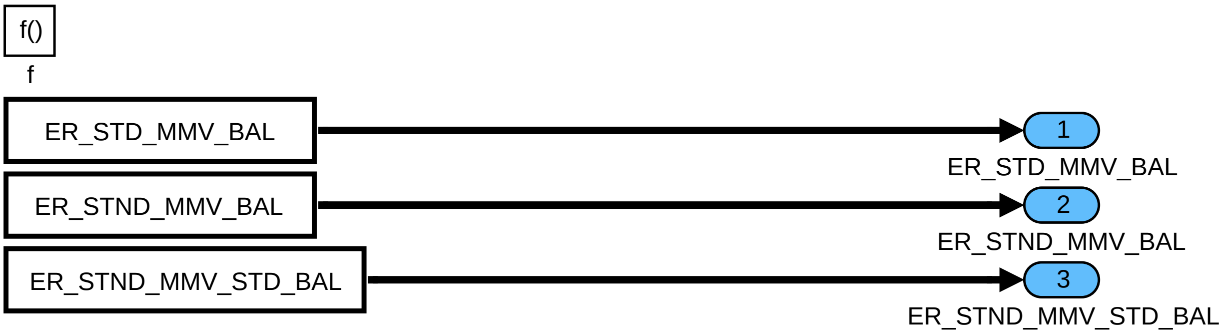


Figure 5 CYBL_SIGNAL/ INI/ NVMYRES

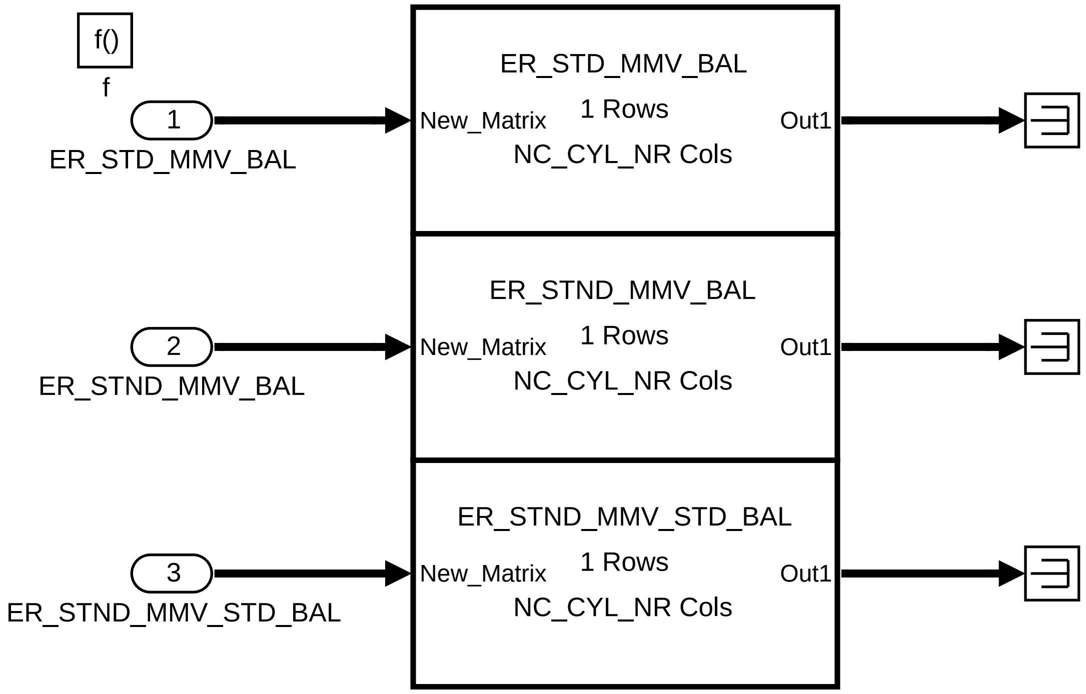


Figure 6 CYBL_SIGNAL/ INI/ NMVYSTO

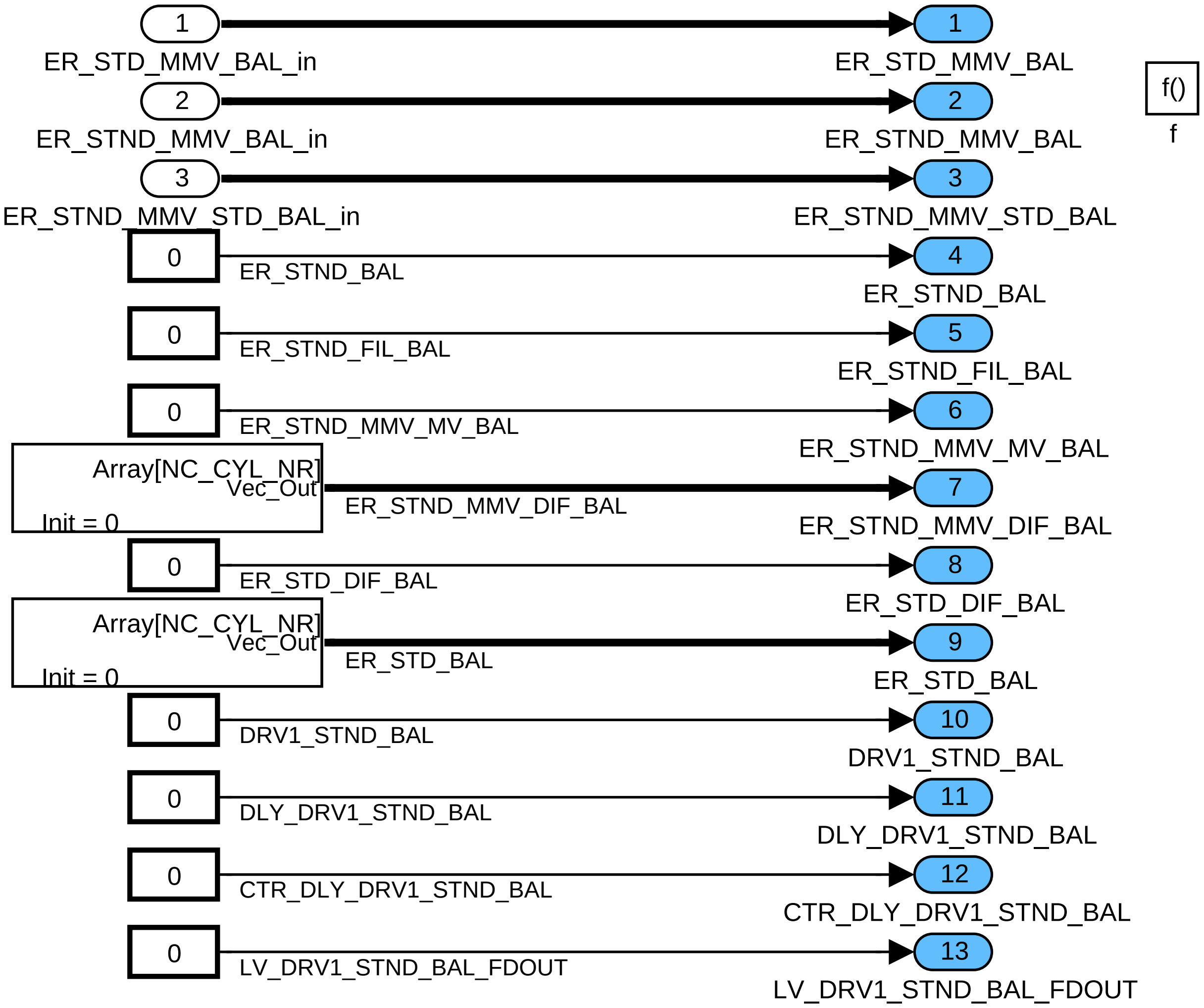


Figure 7 CYBL_SIGNAL/ INI/ RST

6.2.1.2 Calculation of OPR_ER_STND and OPR_DRV1 tasks

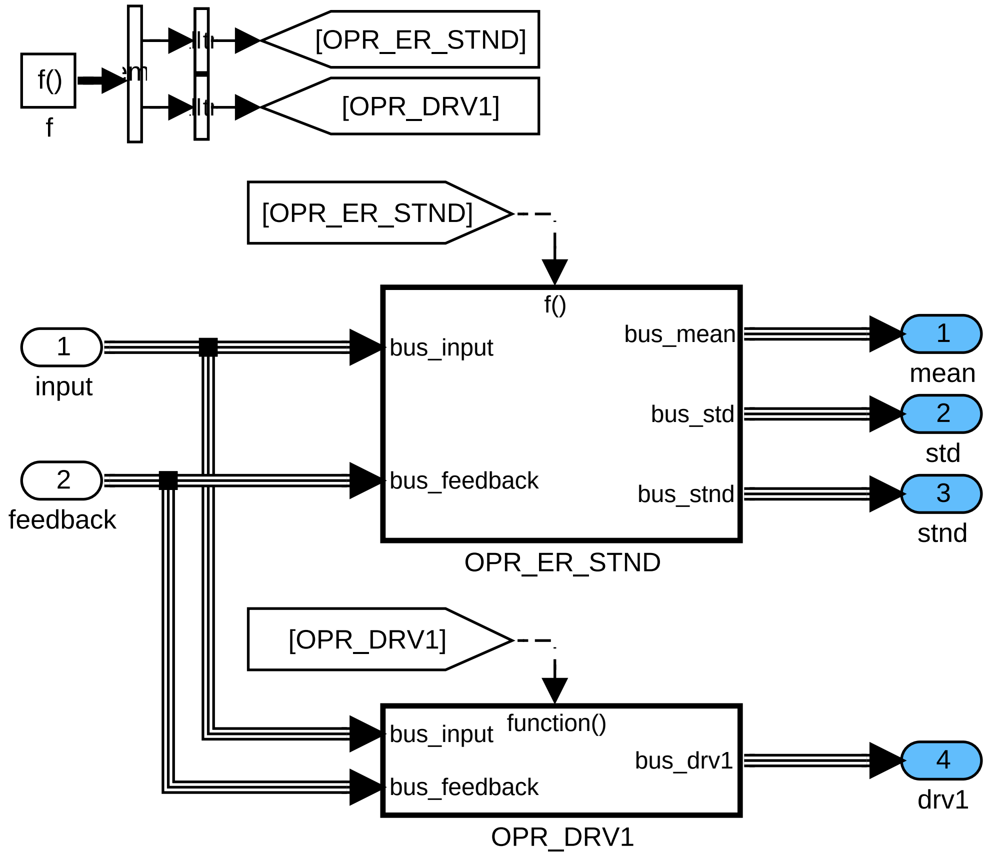


Figure 8 CYBL_SIGNAL/ OPM

<u>Input signals for cylinder balancing adaptation functions</u>

The ER values are used to calculate normalized cylinder selective engine roughness values ER_STND_BAL. After normalization (multiplication by N_32^3), the cylinder selective values are filtered by a Median Filter and a PT1-Filter.

In addition, moving mean values are built (ER_STND_MMV_STD_BAL[SEG_NR_ER]) for the calculation of cylinder specific standard deviations (ER_STD_BAL[SEG_NR_ER]).

The standard deviation ER_STD_DIF_BAL is calculated by subtracting the moving mean values ER_STND_MMV_STD_BAL[SEG_NR_ER] from the ER_STND_BAL values.

The resulting difference ER_STD_DIF_BAL is multiplied with a weighting factor for negative and positive values (C_FAC_ER_STD_NEG/POS_BAL). Afterwards a moving mean value is calculated based on this weighted signal.

To allow a cylinder specific adaptation via the engine roughness, a specific signal has to be created for each cylinder. Therefore, from the variable ER_STND_MMV_BAL[SEG_NR_ER] the value ER_STND_MMV_MV_BAL (average value of all cylinder specific moving mean values) is subtracted.

The values ER_STD_MMV_BAL[SEG_NR_ER], ER_STND_MMV_BAL[SEG_NR_ER] and ER_STND_MMV_STD_BAL[SEG_NR_ER] are stored in the non-volatile memory (NVMY) and used for initialization issues at next engine run.

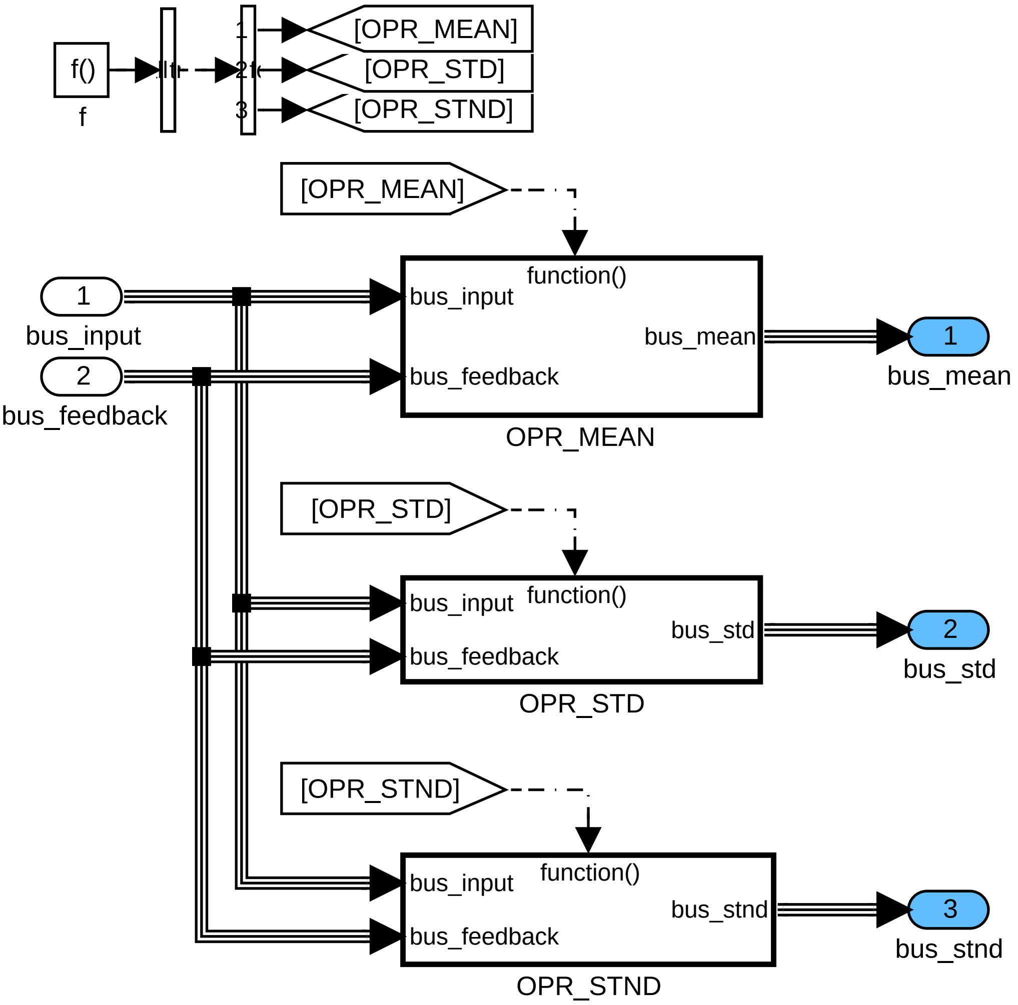


Figure 9 CYBL_SIGNAL/ OPM/ OPR_ER_STND

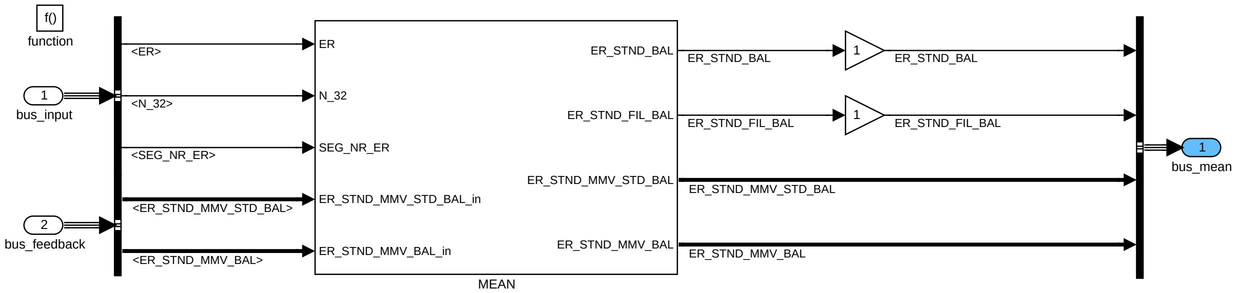


Figure 10 CYBL_SIGNAL/ OPM/ OPR_ER_STND/ OPR_MEAN

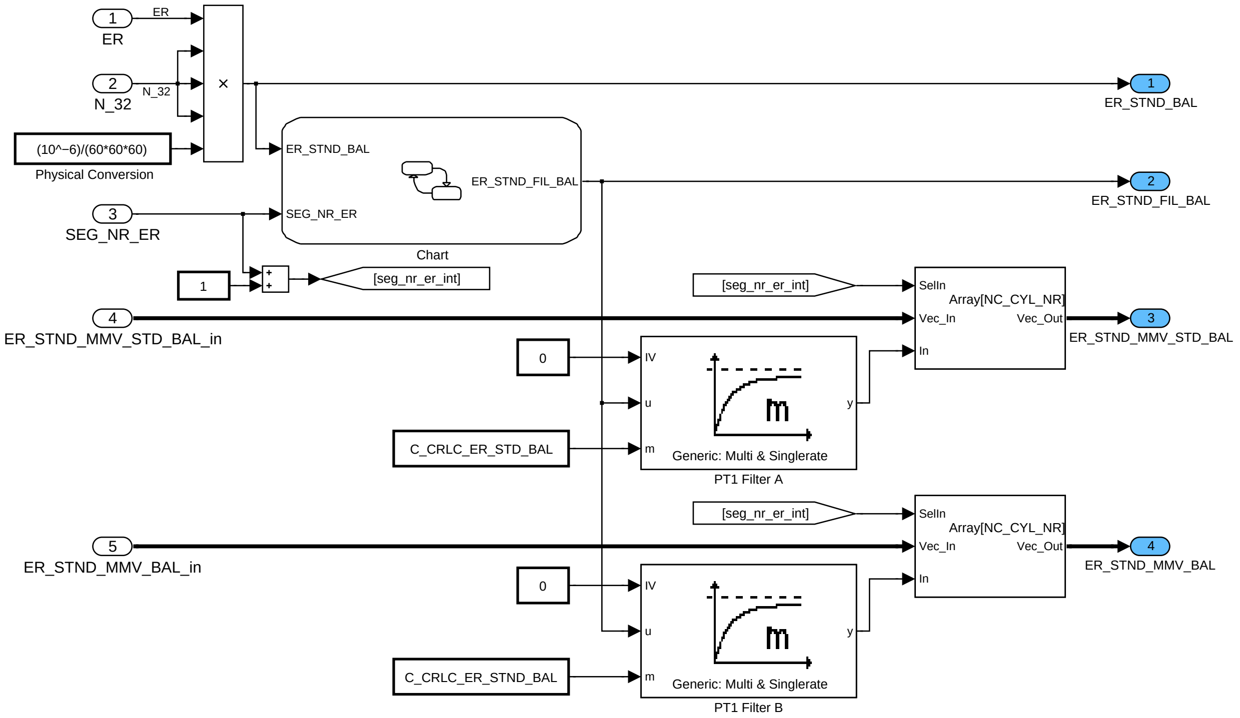


Figure 11 CYBL_SIGNAL/ OPM/ OPR_ER_STND/ OPR_MEAN/ MEAN

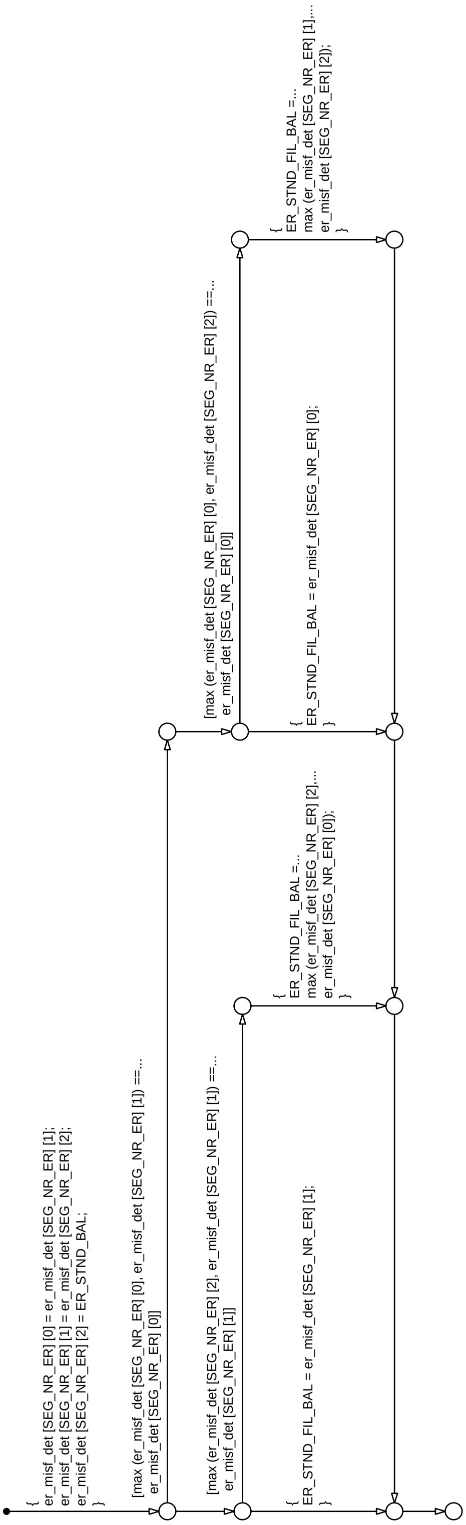


Figure 12 CYBL_SIGNAL/ OPM/ OPR_ER_STND/ OPR_MEAN/ MEAN/ Chart

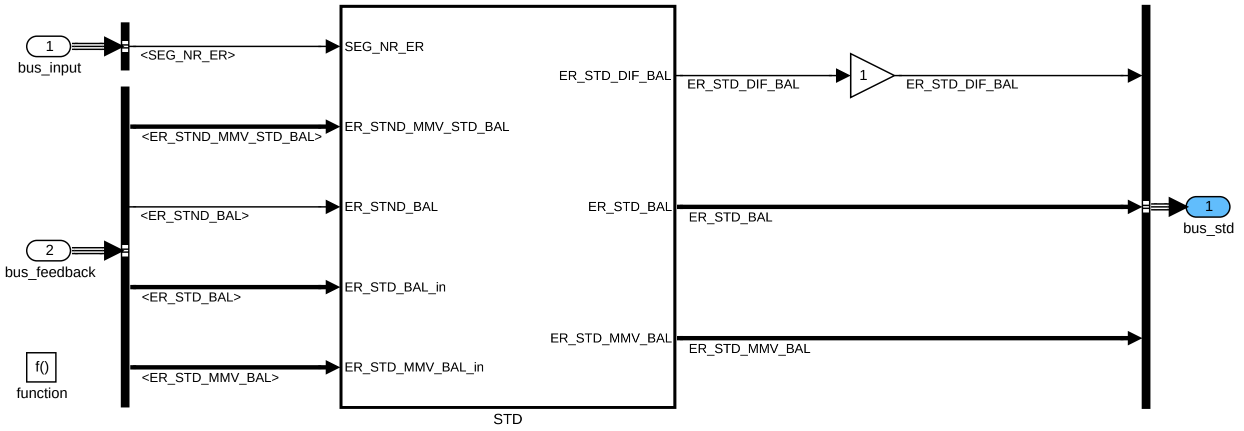


Figure 13 CYBL_SIGNAL/ OPM/ OPR_ER_STND/ OPR_STD

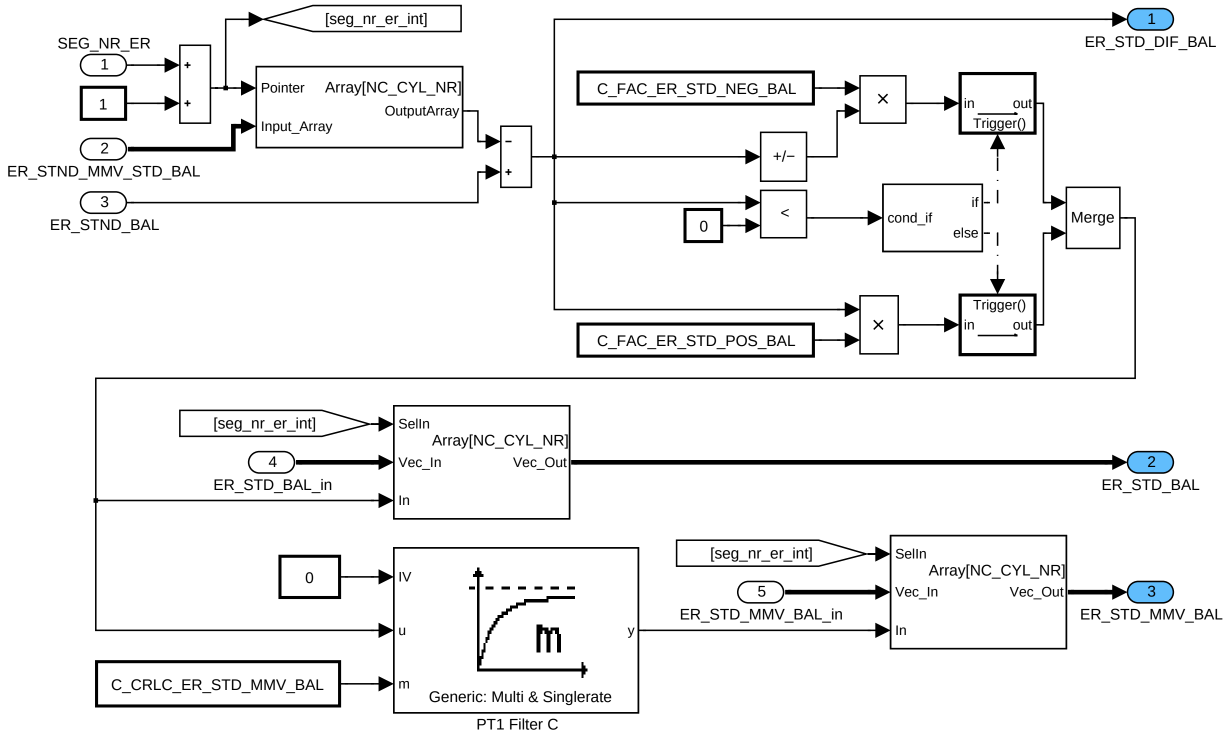


Figure 14 CYBL_SIGNAL/ OPM/ OPR_ER_STND/ OPR_STD/ STD

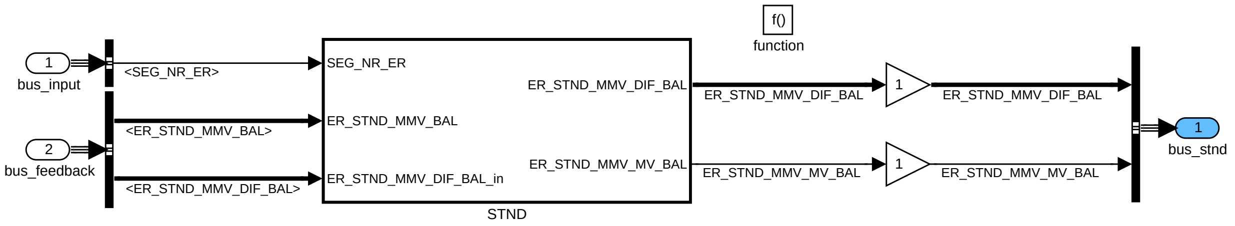


Figure 15 CYBL_SIGNAL/ OPM/ OPR_ER_STND/ OPR_STND

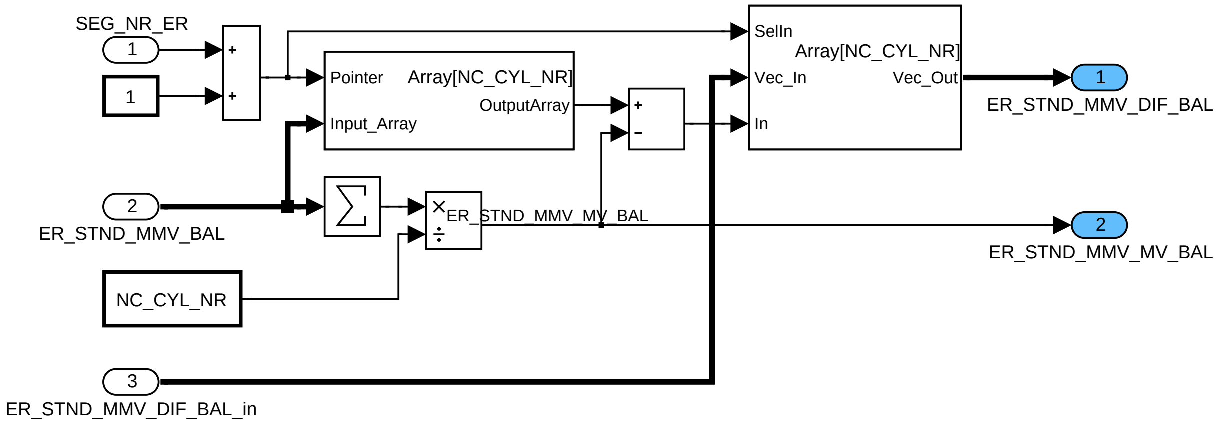


Figure 16 CYBL_SIGNAL/ OPM/ OPR_ER_STND/ OPR_STND/ STND

Calculation of cylinder balancing acceleration fade out conditions

Multiplying the dynamic component (DRV1_ER) part of the engine roughness calculation with the engine speed by the power of three normalizes the signal. The defined signal is reproducing the engine acceleration in numbers. If the acceleration is quite high or low, in case of speeding up or down, the cylinder balancing adaptation is stopped for a certain number of segments. The number of segments are stored in a counter, which is counting back to zero in certain steps as mentioned in the signal flow diagram.

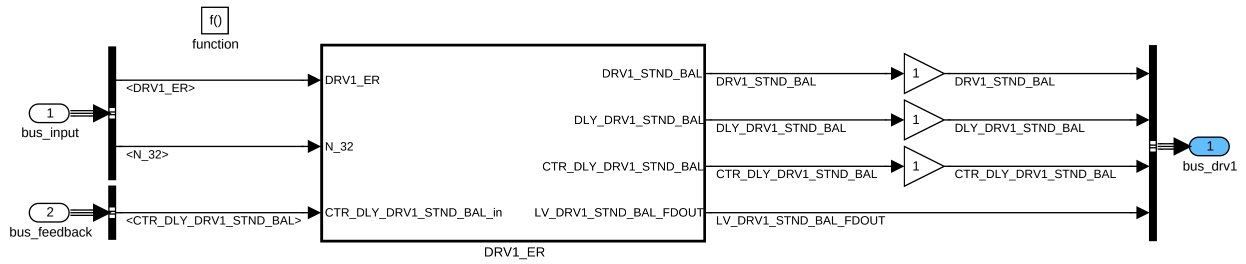


Figure 17 CYBL_SIGNAL/ OPM/ OPR_DRV1

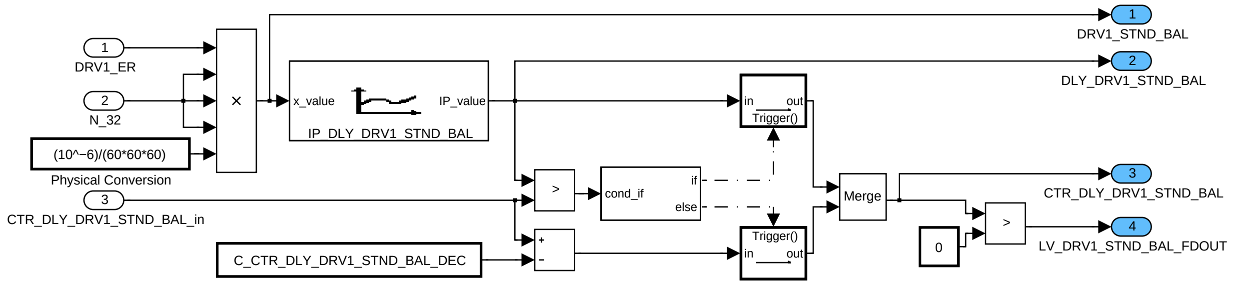


Figure 18 CYBL_SIGNAL/ OPM/ OPR_DRV1/ DRV1_ER

6.2.1.3 SUBFUNCTION: SIG_MNG

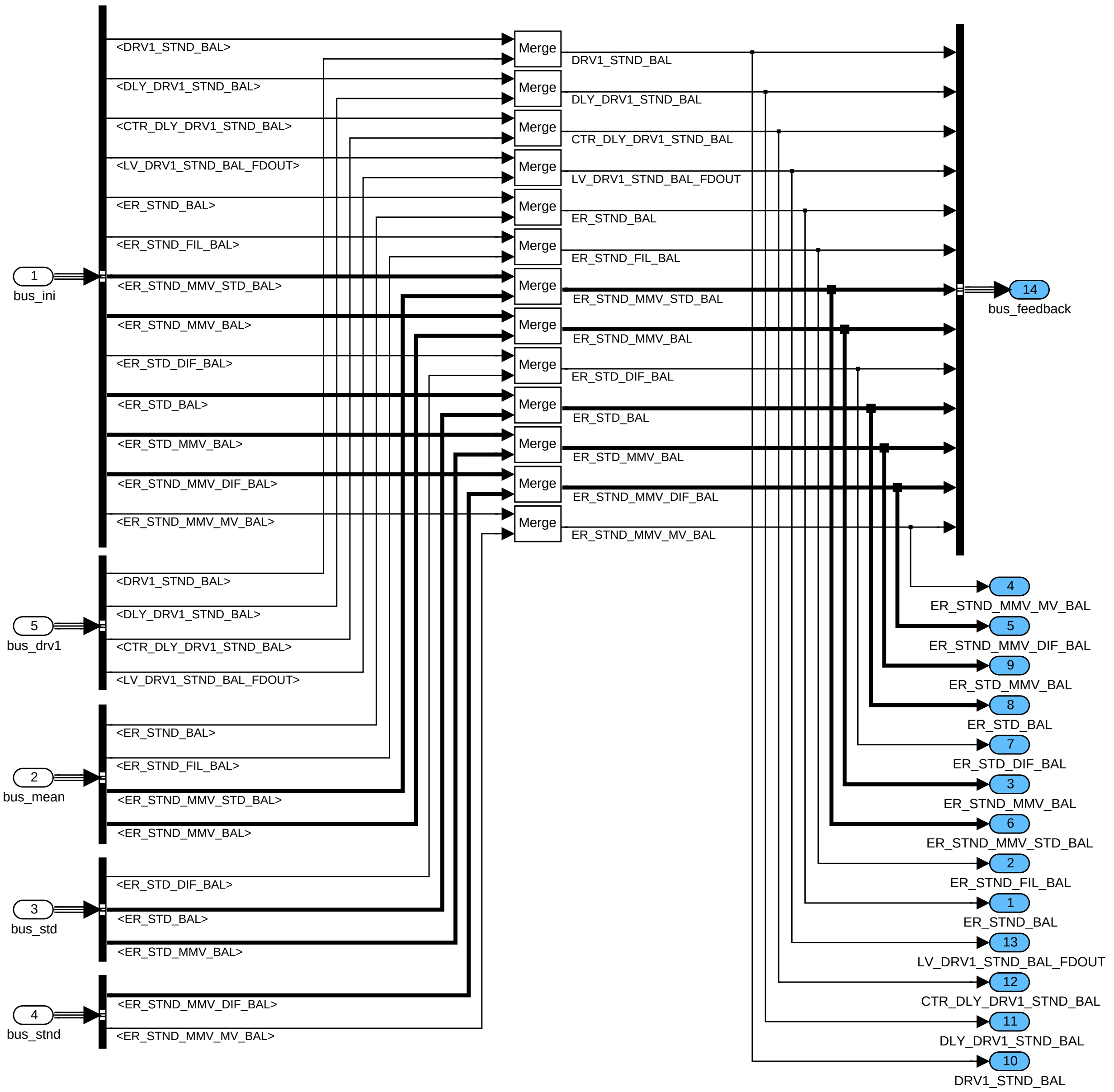


Figure 19 CYBL_SIGNAL/ SIG_MNG

6.3 Coordination of the Injection Time Correction Factors for CYBL

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MFF_ADD_BAL[NC_CYL_NR]	O/V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Total cylinder balancing additive MFF correction value					
FAC_TI_BAL[NC_CYL_NR]	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Total cylinder balancing multiple correction factor					
EGY_LEVEL_IV_BAL_CONV[NC_CYL_NR]	O/V	E000...2000H	-50...50	0.00610352	%
Cylinder balancing injection valve energy level converter output					
EGY_LEVEL_IV_BAL[NC_CYL_NR]	O/V	E000...2000H	-50...50	0.00610352	%
Cylinder balancing injection valve energy level, cylinder individual					
NR_CYC_EGY_LEVEL_IV_BAL_CTL_MAX	V	0...FFH	0...255	1	-
Corresponding segment number of the controller that has reached the maximum limit					
CTR_DLY_TI_BAL_COR_ENA	V	0...FFH	0...255	1	-
Segment counter to delay cylinder balancing TI correction activation					
EGY_LEVEL_IV_BAL_CTL_MAX[NC_CYL_NR]	V	0...FFFFH	0...6.5535E+4	1	-
Cylinder balancing injection valve energy level maximum controller output value reached					
EGY_LEVEL_IV_BAL_CTL[NC_CYL_NR]	V	E000...2000H	-50...50	0.00610352	%
Cylinder balancing injection valve energy level control					
EGY_LEVEL_IV_BAL_LIM	V	E000...2000H	-50...50	0.00610352	%
Cylinder balancing injection valve energy level converter output limited					
EGY_LEVEL_IV_BAL_ADJ[NC_CYL_NR]	V	E000...2000H	-50...50	0.00610352	%
Cylinder balancing injection valve energy level adjustment					
CTR_STEP_FAC_TI_BAL[NC_CYL_NR]	V	80...7FH	-128...127	1	-
Number of cylinder balancing multiple correction steps					
FAC_TI_DIF_BAL[NC_CYL_NR]	V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Cylinder balancing multiple correction factor difference					
FAC_TI_SUM_BAL[NC_CYL_NR]	V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Cylinder balancing multiple correction factor summary					
MFF_ADD_SUM_BAL[NC_CYL_NR]	V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Cylinder balancing additive MFF correction value summary					

Input data:

FAC_TI_ER_BAL[NC_CYL_NR]	MFF_ADD_ER_BAL[NC_CYL_NR]	FAC_CYL_LAM_COR[NC_CYL_NR]	MFF_ADD_CYL_LAM_COR_OUT[NC_CYL_NR]
N	MFF_SP_MV	STATE_TI_ER_BAL	LV_ST_END
LV_S_ACT	SEG_NR_ER	NC_CYL_NR	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_DLY_TI_BAL_COR_INI	1	0...FFH	0...255	1	-
Segment counter initialization value to delay cylinder balancing TI correction activation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EGY_LEVEL_IV_BAL_MAX_NEG	1	E000...2000H	-50...50	0.00610352	%
Maximum negative threshold for injection valve energy level value					
C_EGY_LEVEL_IV_BAL_MAX_POS	1	E000...2000H	-50...50	0.00610352	%
Maximum positive threshold for injection valve energy level value					
C_EGY_LEVEL_IV_BAL_OFS	1	E000...2000H	-50...50	0.00610352	%
Cylinder balancing injection valve energy level offset					
C_FAC_TI_BAL_ACT_MAN	1	0H 1H 2H 3H	DISABLE ADJUSTMENT CONTROL ADJCTL	1	-
Manual adjustment of activation conditions for injection valve energy level calculation					
C_FAC_TI_DIF_BAL	1	0...FFFFH	0...1.99996948	3.05176E-5	-
Threshold for cylinder balancing multiple correction difference					
C_MFF_COR_BAL_MOD_MAN	1	0H 1H 2H 3H 4H 5H	NORM LAM ER TOT ER_ADD LAM_ADD	1	-
Manual switch for cylinder balancing correction mode determination					
LC_FAC_TI_BAL_RST_MAN	1	0...1H	0...1	1	-
Manual switch for cylinder balancing multiple correction reset					
IP_EGY_LEVEL_IV_BAL_CONV	8	0...4000H	-50...50	0.00610352	%
LDP_FAC_TI_SUM_BAL_IP_EGY_BAL	8	0...FFFFH	-1...0.99996948	3.05176E-5	-
Index table for cylinder balancing injection valve energy level conversion					
IP_FAC_EGY_LEVEL_IV_BAL_COR	8	0...FFFFH	0...3.99993896	6.10352E-5	-
LDP_MFF_SP_MV_IP_FAC_EGY_BAL	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Index table for cylinder balancing injection valve energy level correction factor					
IP_FAC_MFF_ADD_ER_BAL	4x4	0...FFFFH	0...1.999969	3.05176E-5	-
LDPM_N_2_CYBL	4	0...1FE0H	0...8.16E+3	1	rpm
LDPM_MFF_SP_MV_1_CYBL	4	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Index table for cylinder balancing additive MFF correction factor					

6.3.1 General information

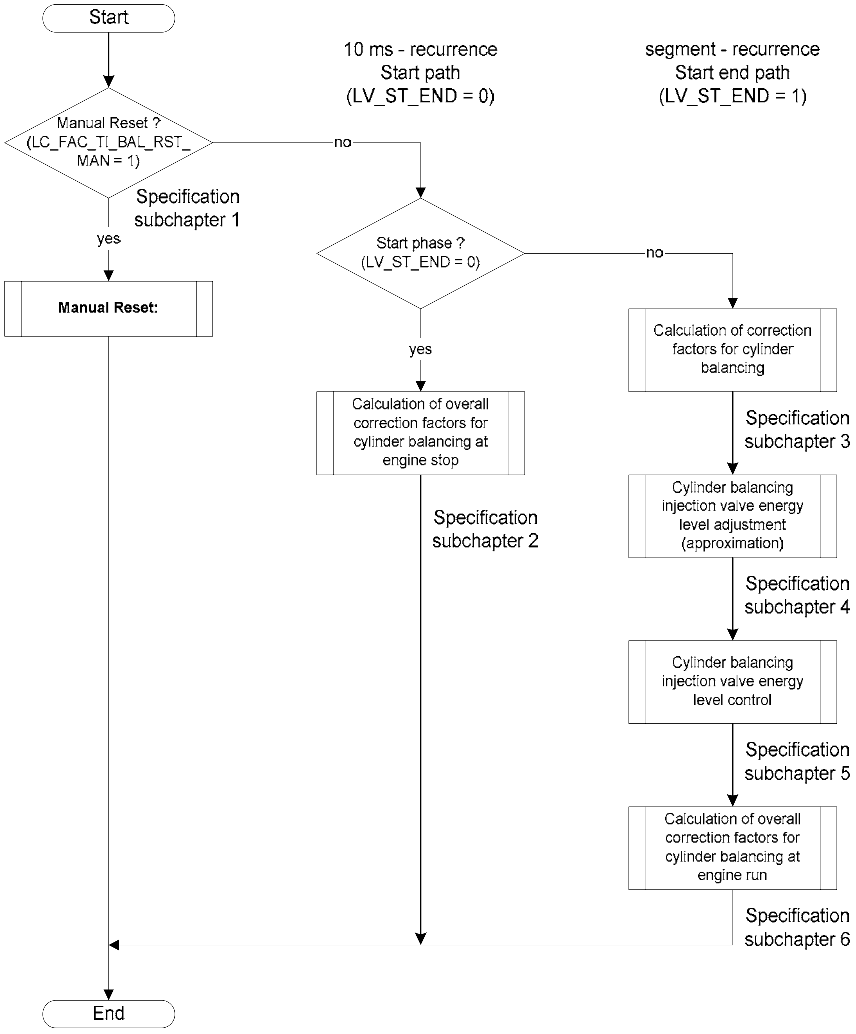
This module coordinates the adaptation and application of the correction factors calculated by the functions "Cylinder Balancing via ER" and "Cylinder Balancing via Lambda Sensor".

The module "Cylinder Balancing via Lambda Sensor" calculates the adaptation values only in homogenous modes (lean or stochiometric), while the module "Cylinder Balancing via ER" calculates the adaptation values only in stratified mode. it is possible to apply manually the calculated adaptation values either in homogeneous or stratified or in both combustion modes with a calibration constant.

With use of a calibration constant a manual adjustment of the used injector energy level calculation is possible. The complete energy level calculation is disabled and the corresponding output values for the injection path are set to zero if the switch is set to "Disable".

Two different algorithms are existing for the injection valve energy level calculation. While the manual switch is set to "Adjustment", the calculation is based on a step by step approximation until a stable energy level is reached. In case the manual switch is set to "Control", the energy level is calculated by a converter.

Both calculation routines can be activated in parallel while the manual switch is set to "AJD/CTL". The common energy level then is a summary of the adjustment value with the controller value.



Application Condition

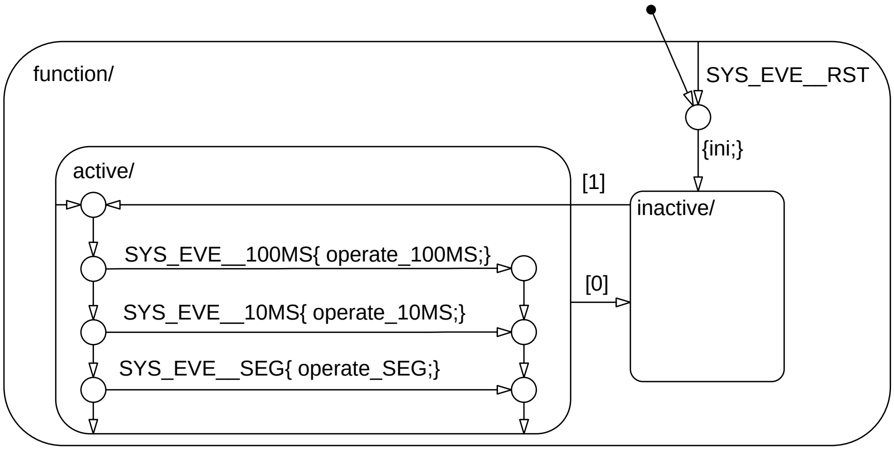


Figure 20 CYBL_M30704P01/ APP_CDN/ Chart

Function Description

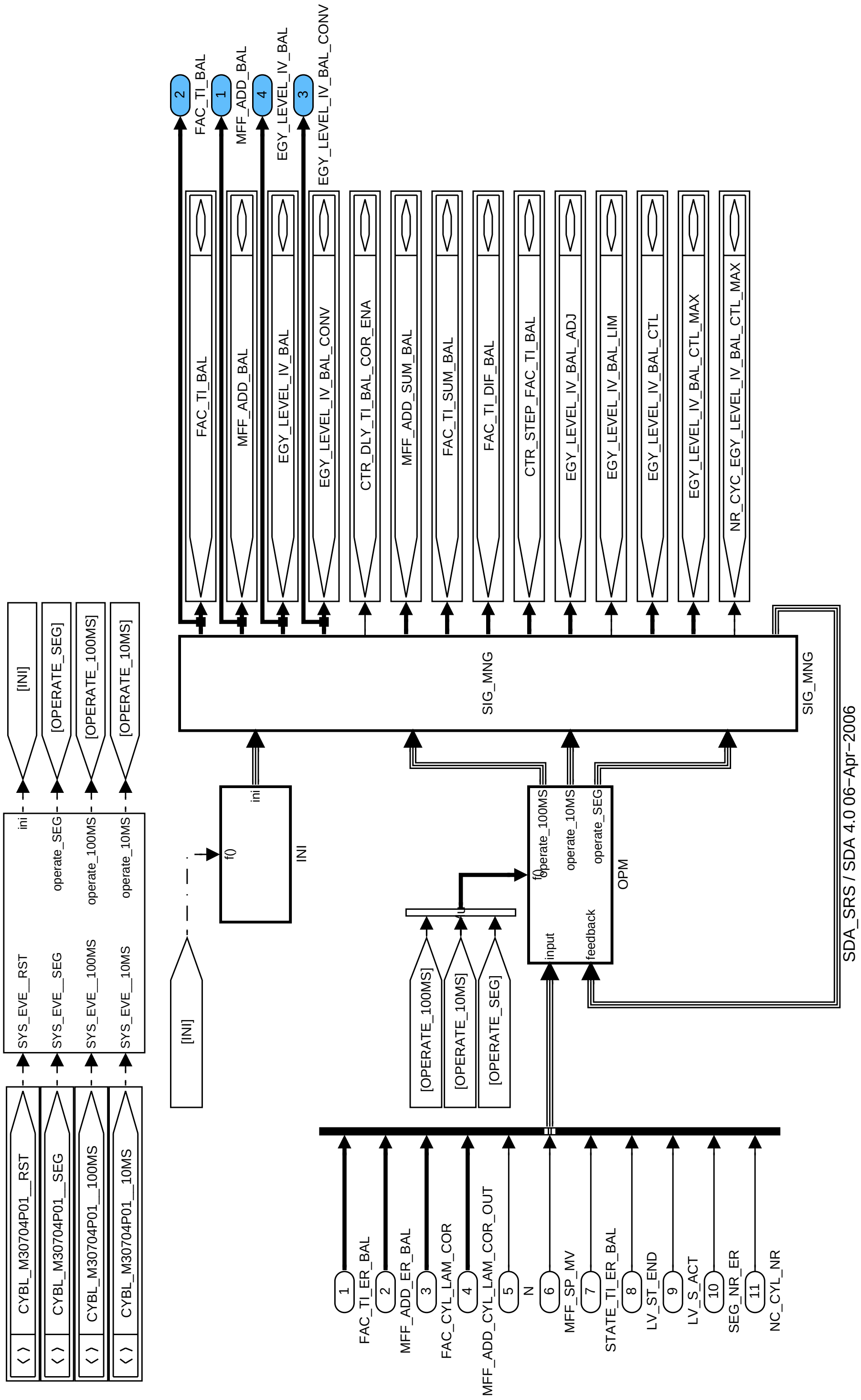


Figure 21 CYBL_M30704P01

6.3.1.1 Calculation of variables at reset

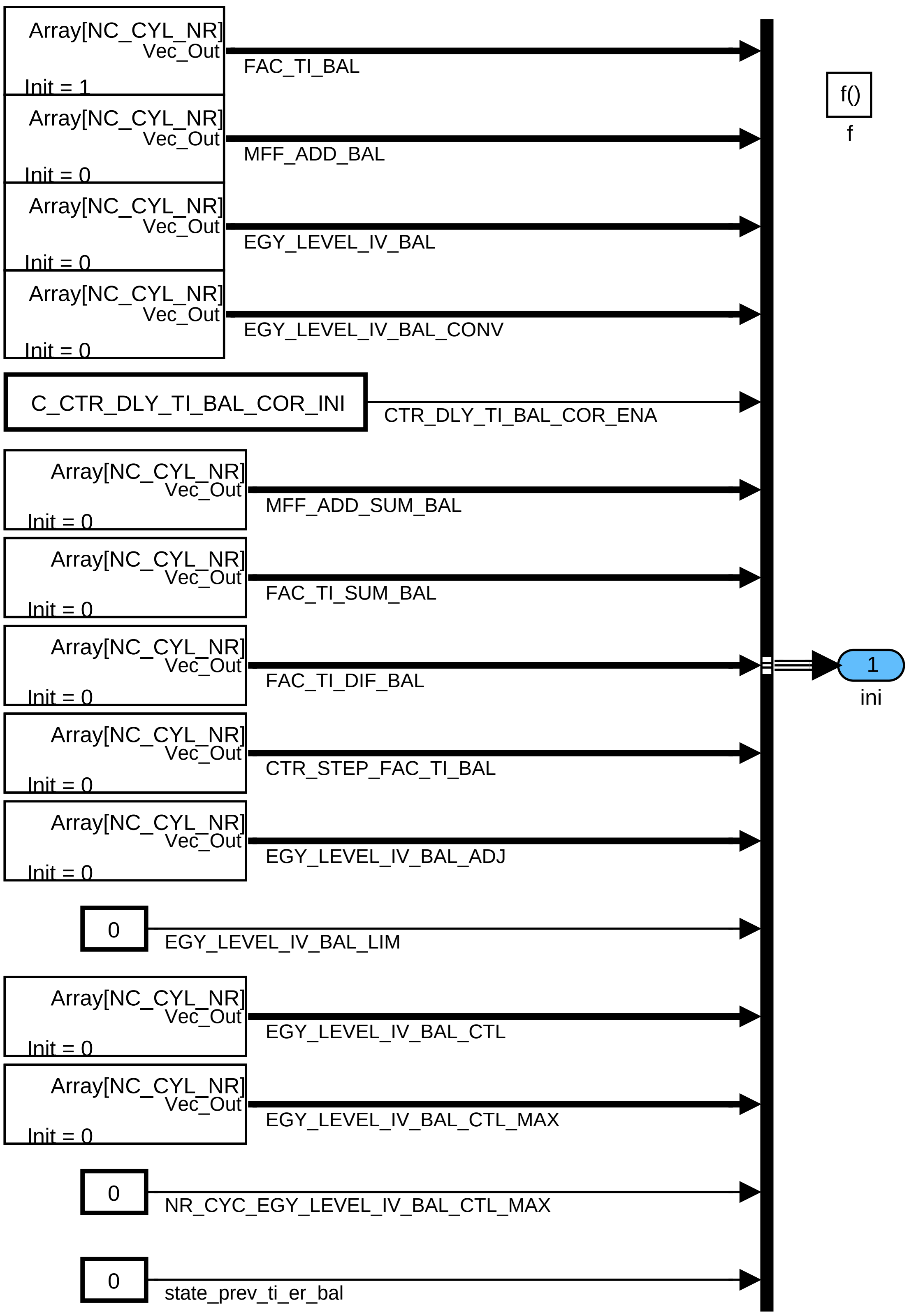


Figure 22 CYBL_M30704P01/ INI

6.3.1.2 Calculation of 100ms, 10ms and segment synchronous tasks

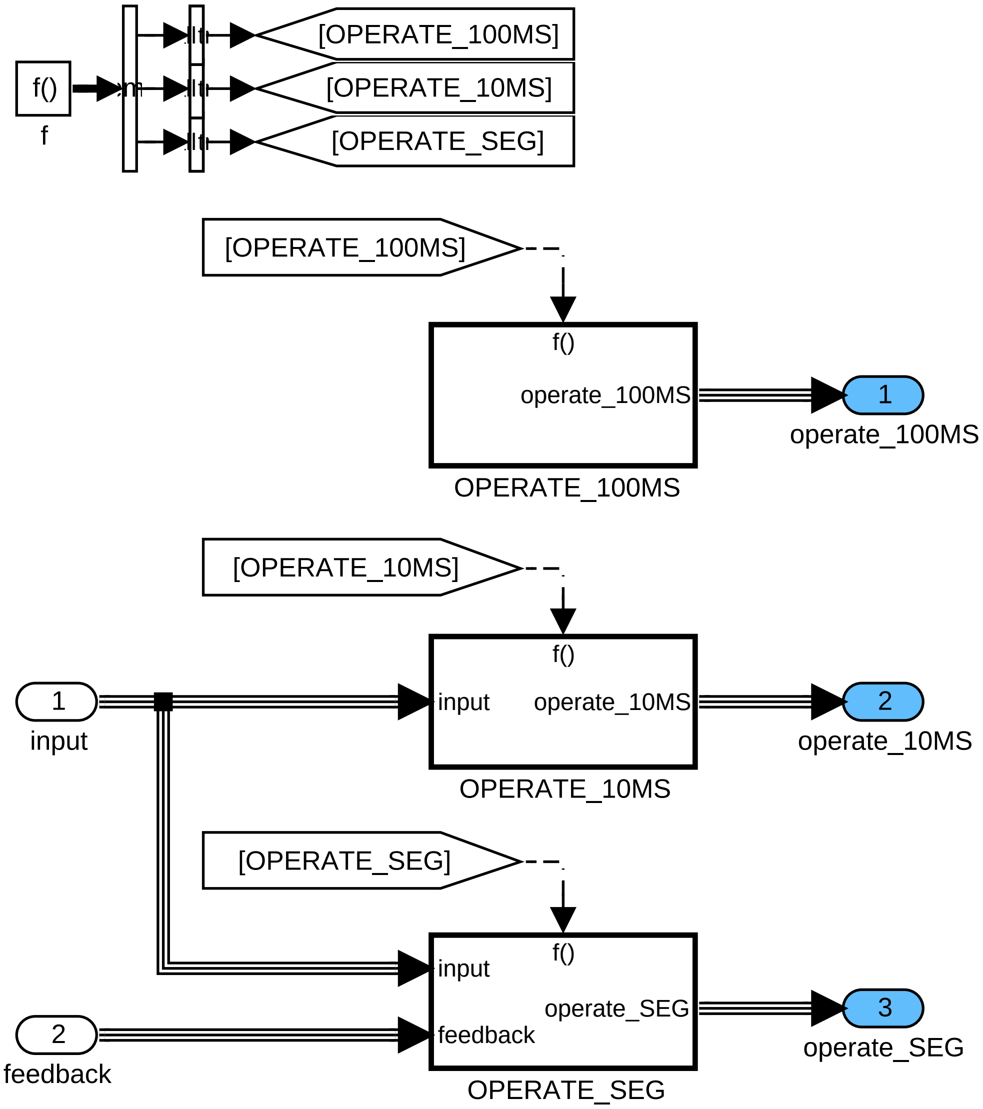


Figure 23 CYBL_M30704P01/ OPM

Manual Reset of cylinder balancing correction values

In case of a manual reset it is possible to adjust the output data to neutral values to make the function test and handling more easier. While the manual function reset is activated, all following subchapters within the module "Coordination of the Injection Time Correction Factors for CYBL" are deactivated during the period.

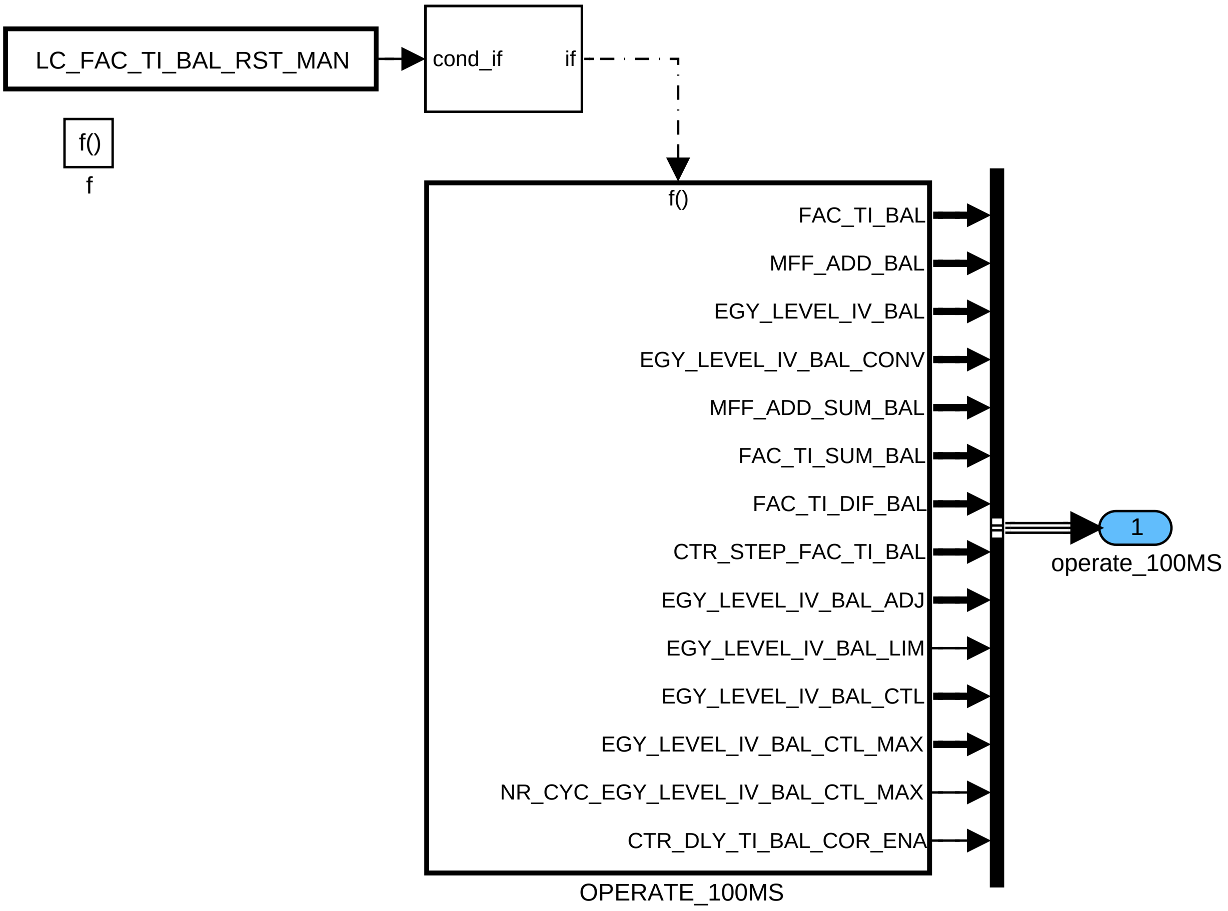


Figure 24 CYBL_M30704P01/ OPM/ OPERATE_100MS

Manual Reset of cylinder balancing correction values OPERATE_100MS

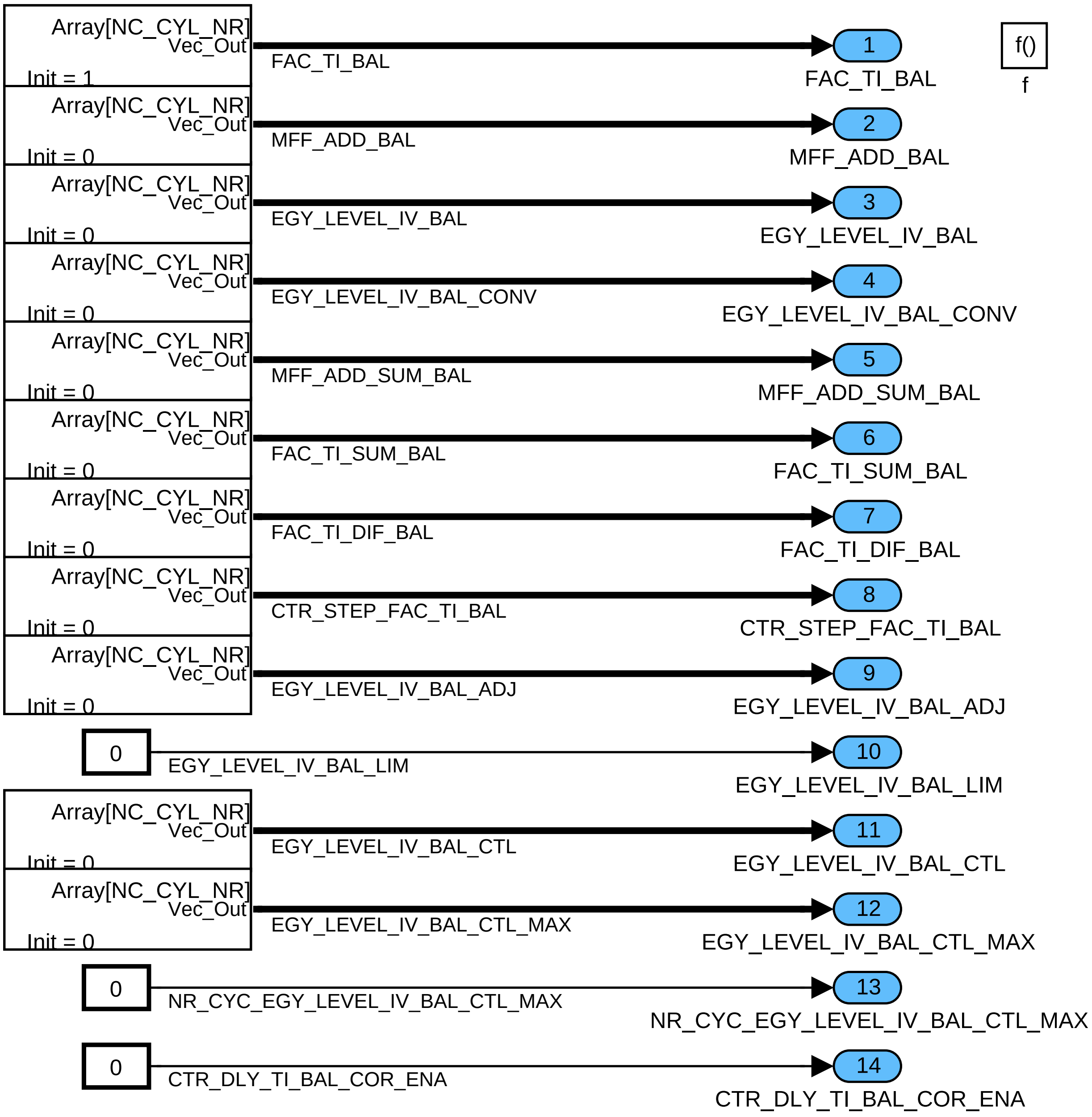


Figure 25 CYBL_M30704P01/ OPM/ OPERATE_100MS/ OPERATE_100MS

Calculation of overall correction factors for cylinder balancing at engine stop

The formula section is only activated if the switch for a manual reset is set to "0". In case of a manual reset (refer:" Manual Reset of cylinder balancing correction values"), the calculations are stopped at once.

The calculation of the total cylinder balancing correction output values takes place at engine stop and engine start in fixed time steps. The output values are used to adjust the wanted cylinder individual corrections. All total cylinder correction values (for every cylinder) are calculated within one time step during this period.

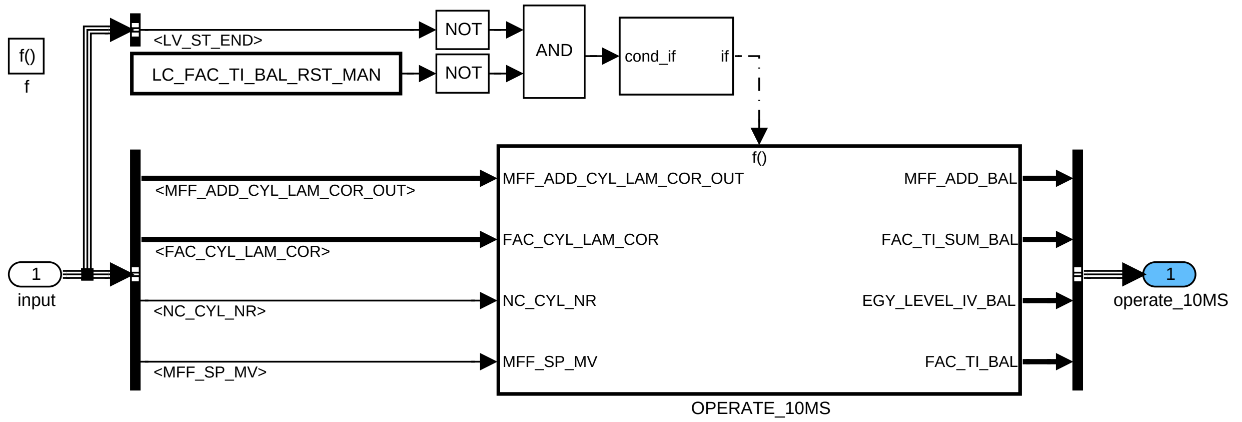


Figure 26 CYBL_M30704P01/ OPM/ OPERATE_10MS

Calculation of overall correction factors for cylinder balancing at engine stop OPERATE_10MS

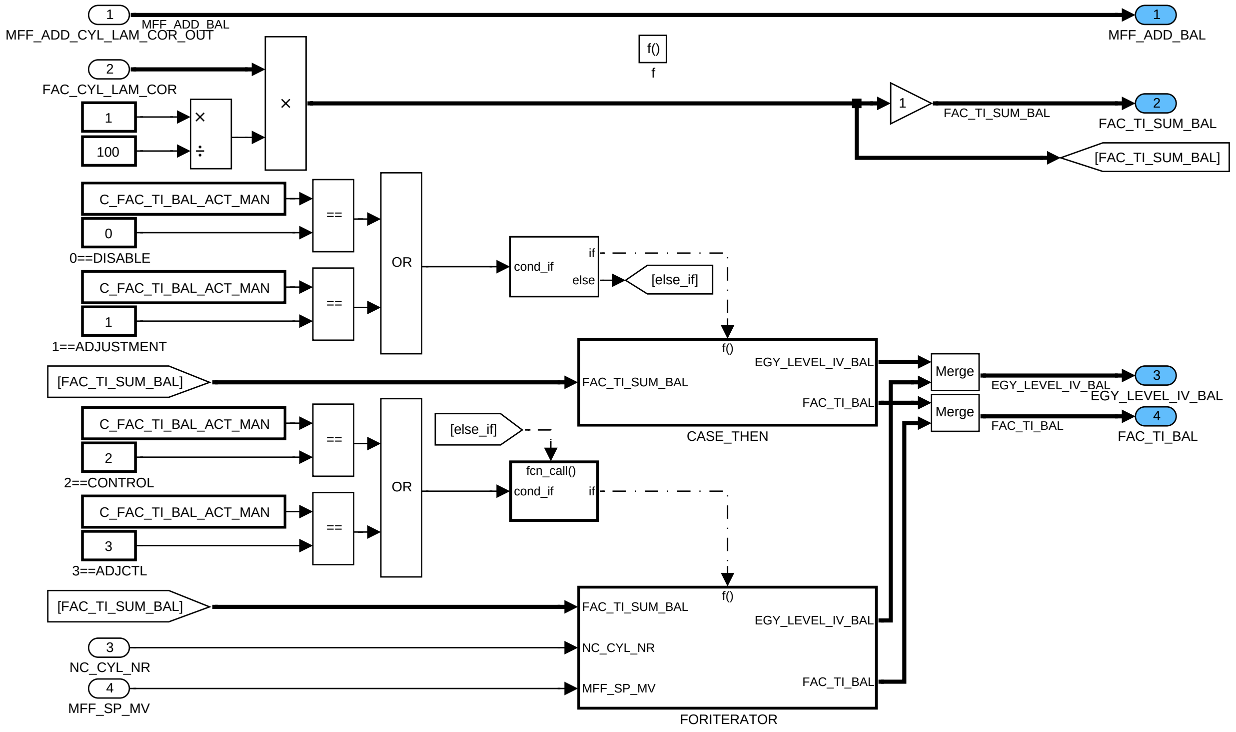


Figure 27 CYBL_M30704P01/ OPM/ OPERATE_10MS/ OPERATE_10MS

Calculation of overall correction factors for cylinder balancing at engine stop OPERATE_10MS CASE_THEN

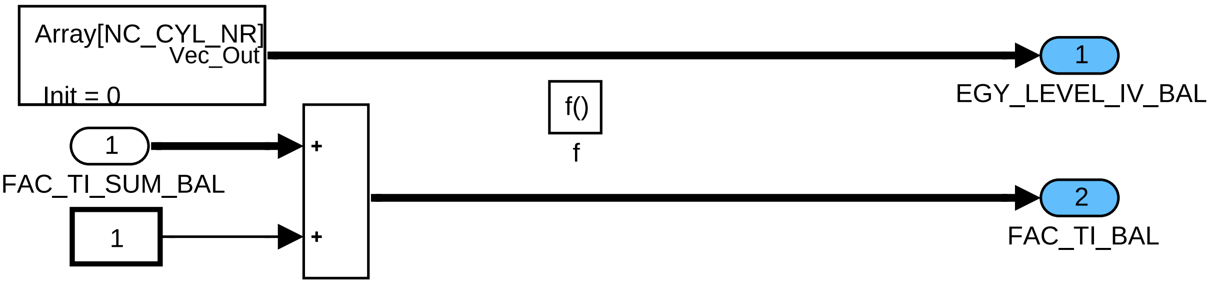


Figure 28 CYBL_M30704P01/ OPM/ OPERATE_10MS/ OPERATE_10MS/ CASE_THEN

Calculation of overall correction factors for cylinder balancing at engine stop OPERATE_10MS FORITERATOR

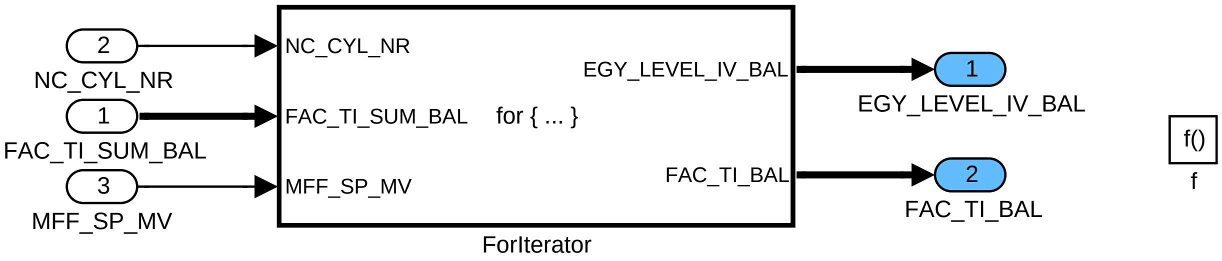


Figure 29 CYBL_M30704P01/ OPM/ OPERATE_10MS/ OPERATE_10MS/ FORITERATOR

Calculation of overall correction factors for cylinder balancing at engine stop OPERATE_10MS FORITERATOR FORITERATOR

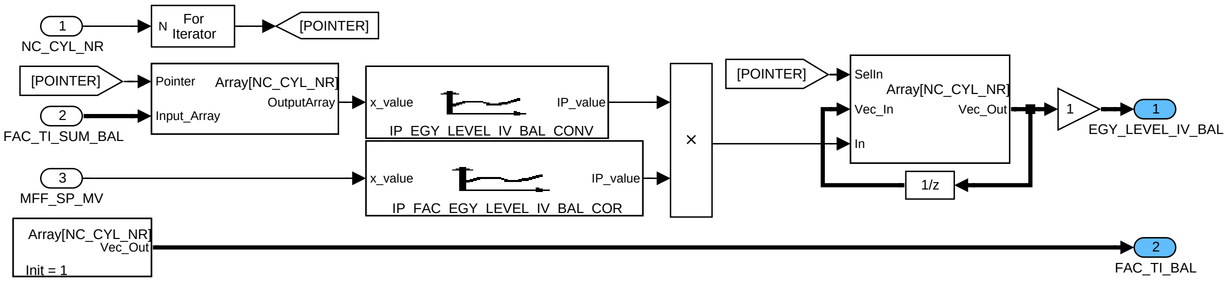


Figure 30 CYBL_M30704P01/ OPM/ OPERATE_10MS/ OPERATE_10MS/ FORITERATOR/ ForIterator

General information for OPERATE_SEG

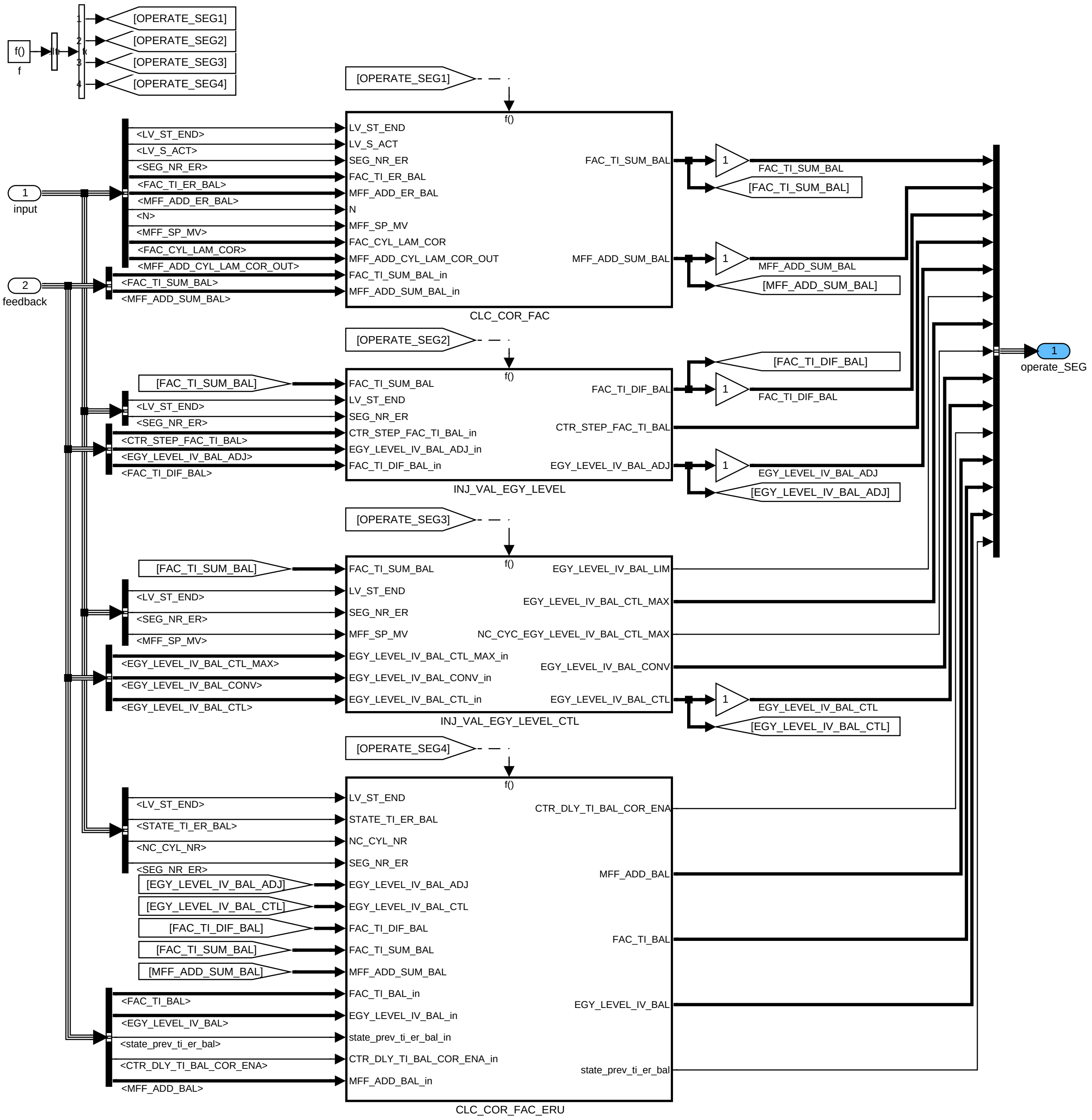


Figure 31 CYBL_M30704P01/ OPM/ OPERATE_SEG

Calculation of correction factors for cylinder balancing

The formula section is only activated if the switch for a manual reset is set to "0". In case of a manual reset (refer: "Manual Reset of cylinder balancing correction values"), the calculations are stopped at once.

The cylinder balancing correction factors for homogenous- and stratified combustion mode are built from the input values of the different cylinder balancing adaptation functions. With a manual calibration constant it is possible to adjust the adaptation values at different combustion modes.

Short description of possible settings:

Constant setting:		**CYBL via LAM sensor applied at combustion mode:**	**CYBL via ER applied at combustion mode:**
"NORM"	(**0**H)	HOM	STR
"LAM"	(**1**H)	HOM & STR	STR
"ER"	(**2**H)	HOM	STR & HOM
"TOT"	(**3**H)	HOM & STR	STR & HOM
"ER_ADD"	(**4**H)	HOM	STR (additive, multiple) HOM (additive)
"LAM_ADD"	(**5**H)	HOM & STR	STR (additive, multiple) HOM (additive)

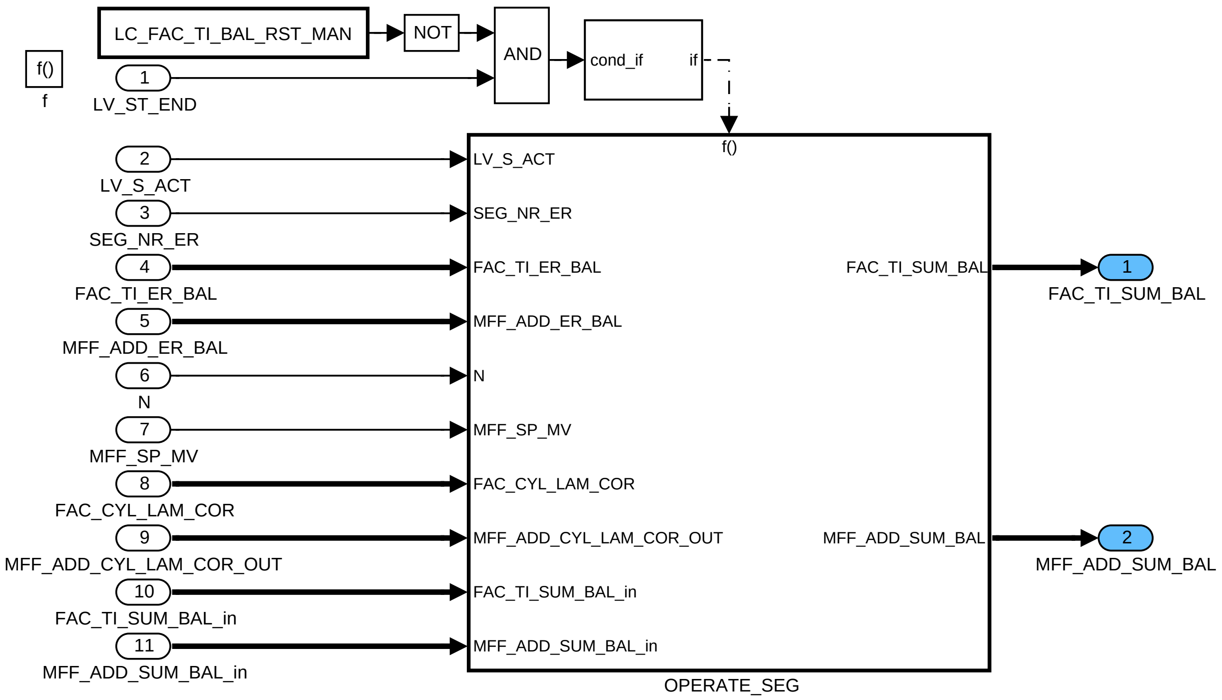


Figure 32 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC

Calculation of correction factors for cylinder balancing OPERATE_SEG

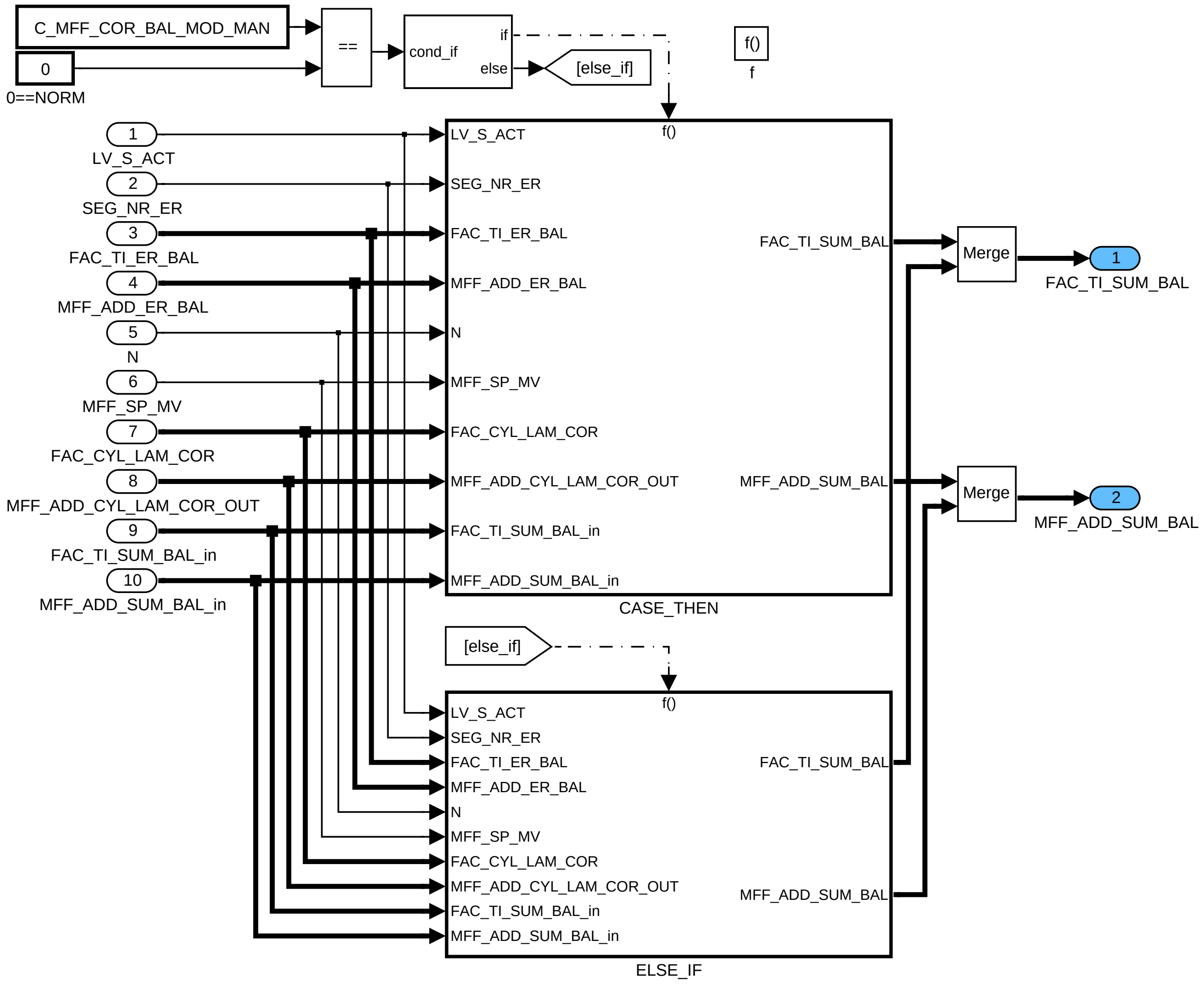


Figure 33 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG

Calculation of correction factors for cylinder balancing OPERATE_SEG CASE_THEN

//CYBL via LAM only applied in HOM, CYBL via ER only applied in STR

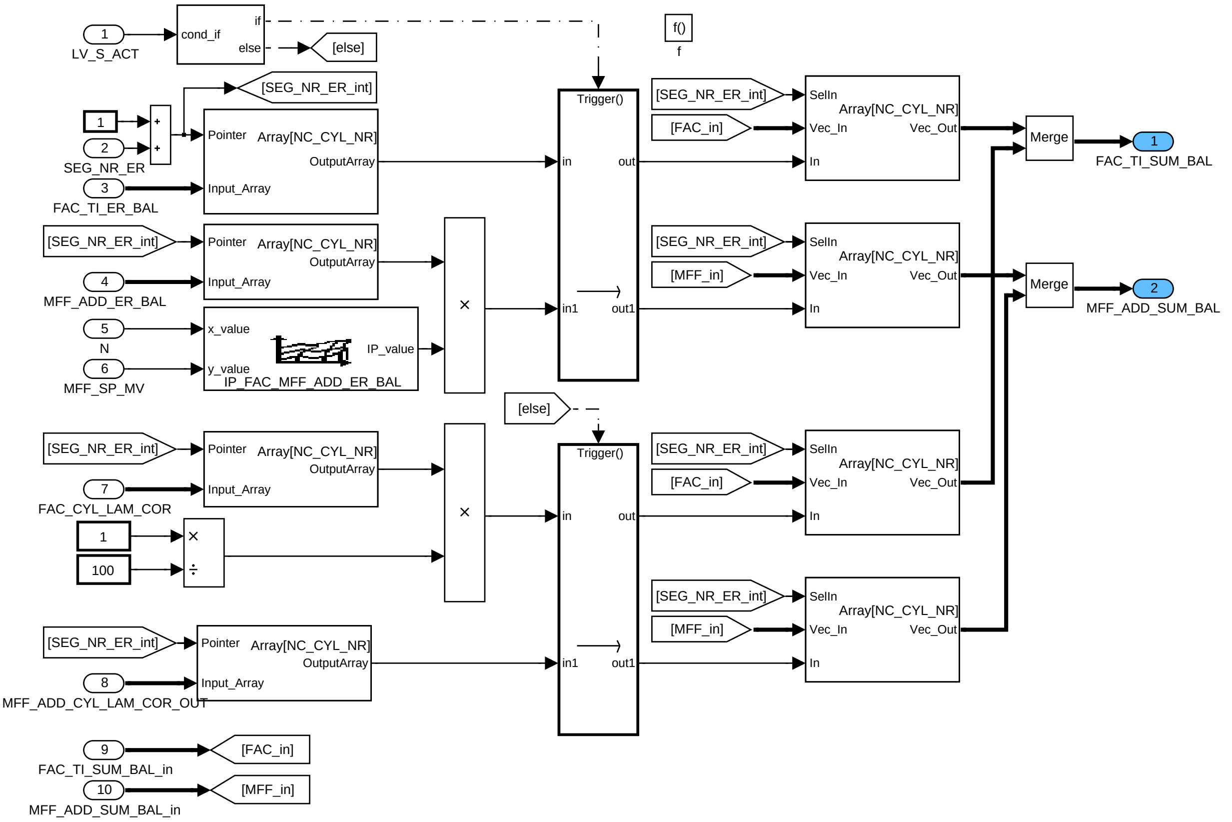


Figure 34 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ CASE_THEN

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF

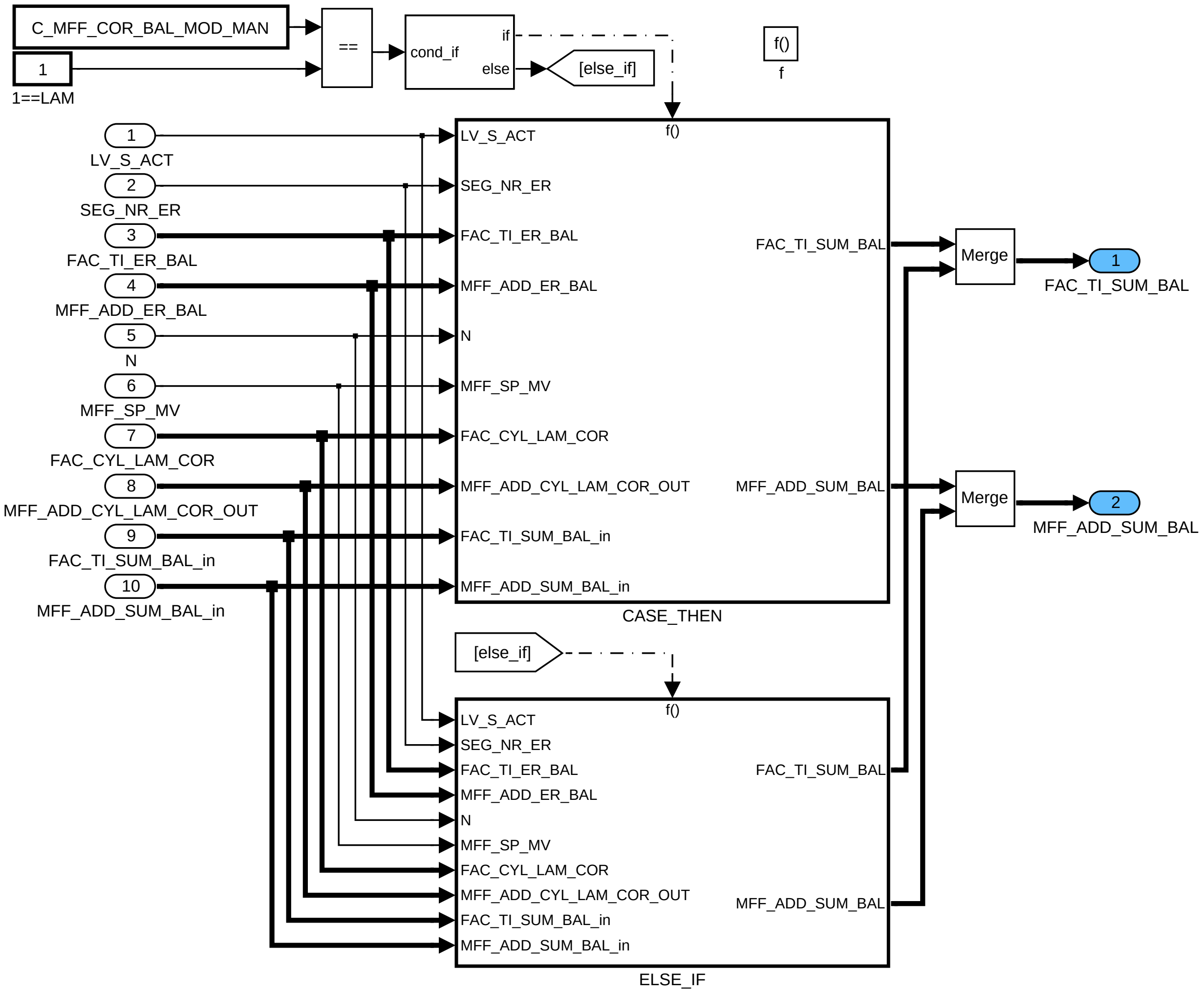


Figure 35 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF CASE_THEN

//CYBL via LAM applied in HOM and STR, CYBL via ER only applied in STR

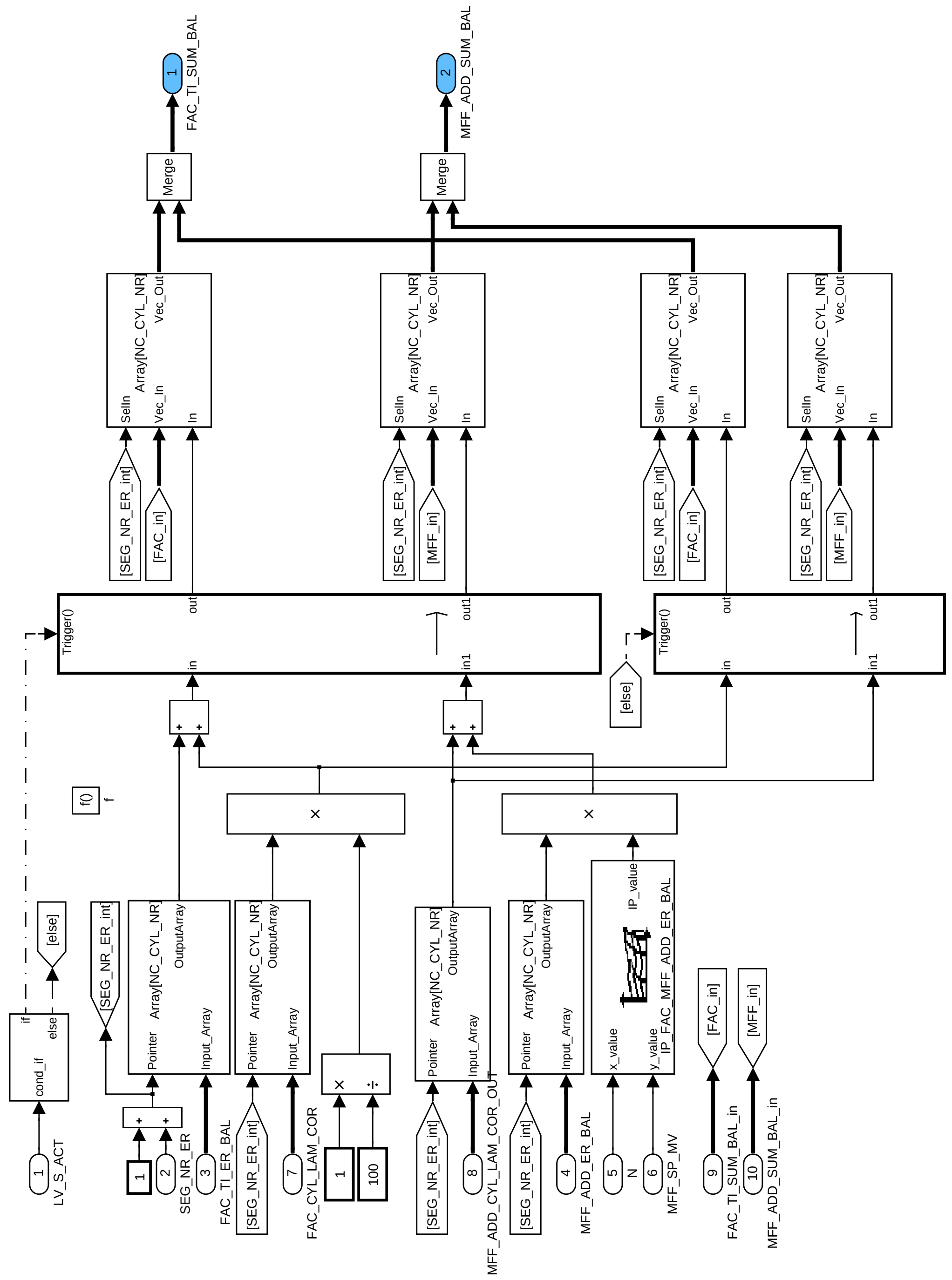


Figure 36 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ CASE_THEN

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF ELSE_IF

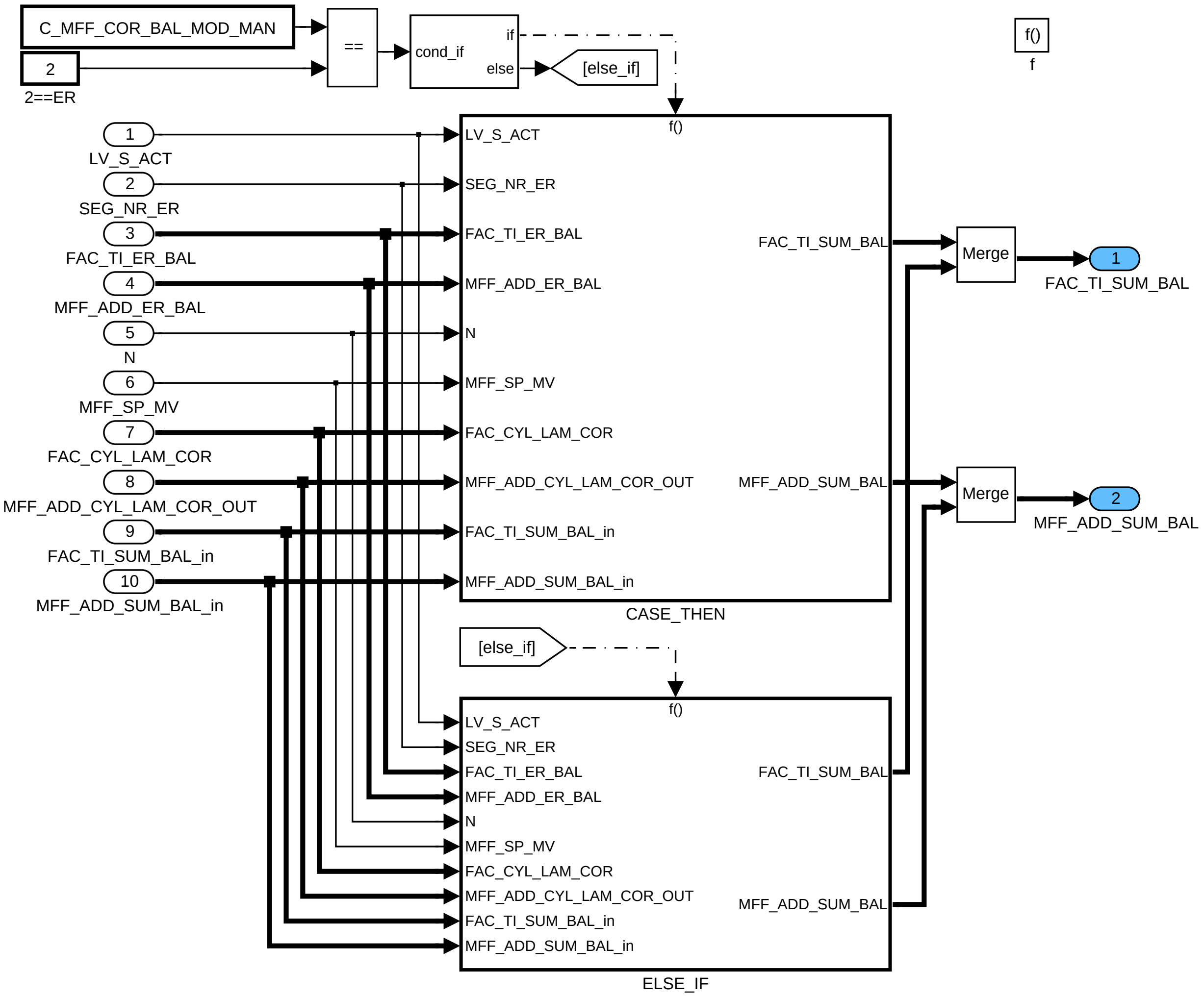


Figure 37 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ ELSE_IF

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF ELSE_IF CASE_THEN

//CYBL via LAM only applied in HOM, CYBL via ER applied in HOM and STR

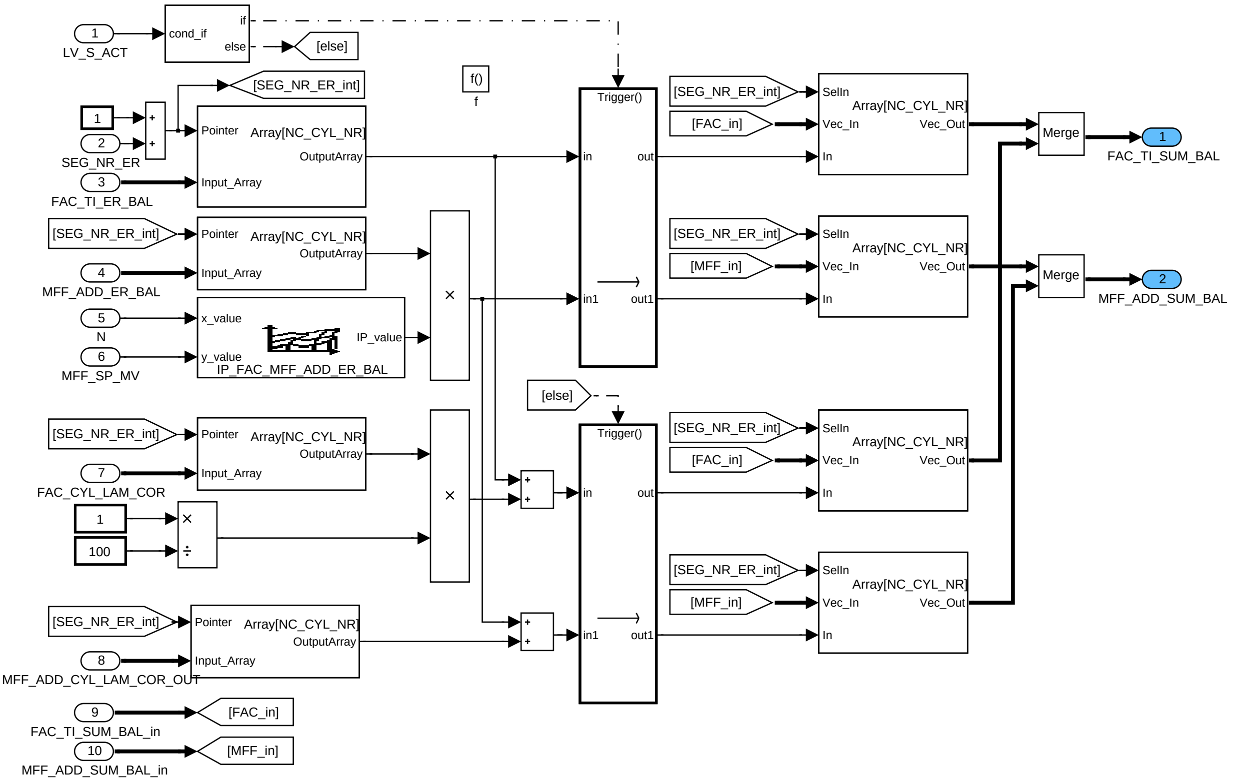


Figure 38 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ ELSE_IF/ CASE_THEN

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF ELSE_IF ELSE_IF

//no combustion mode restriction

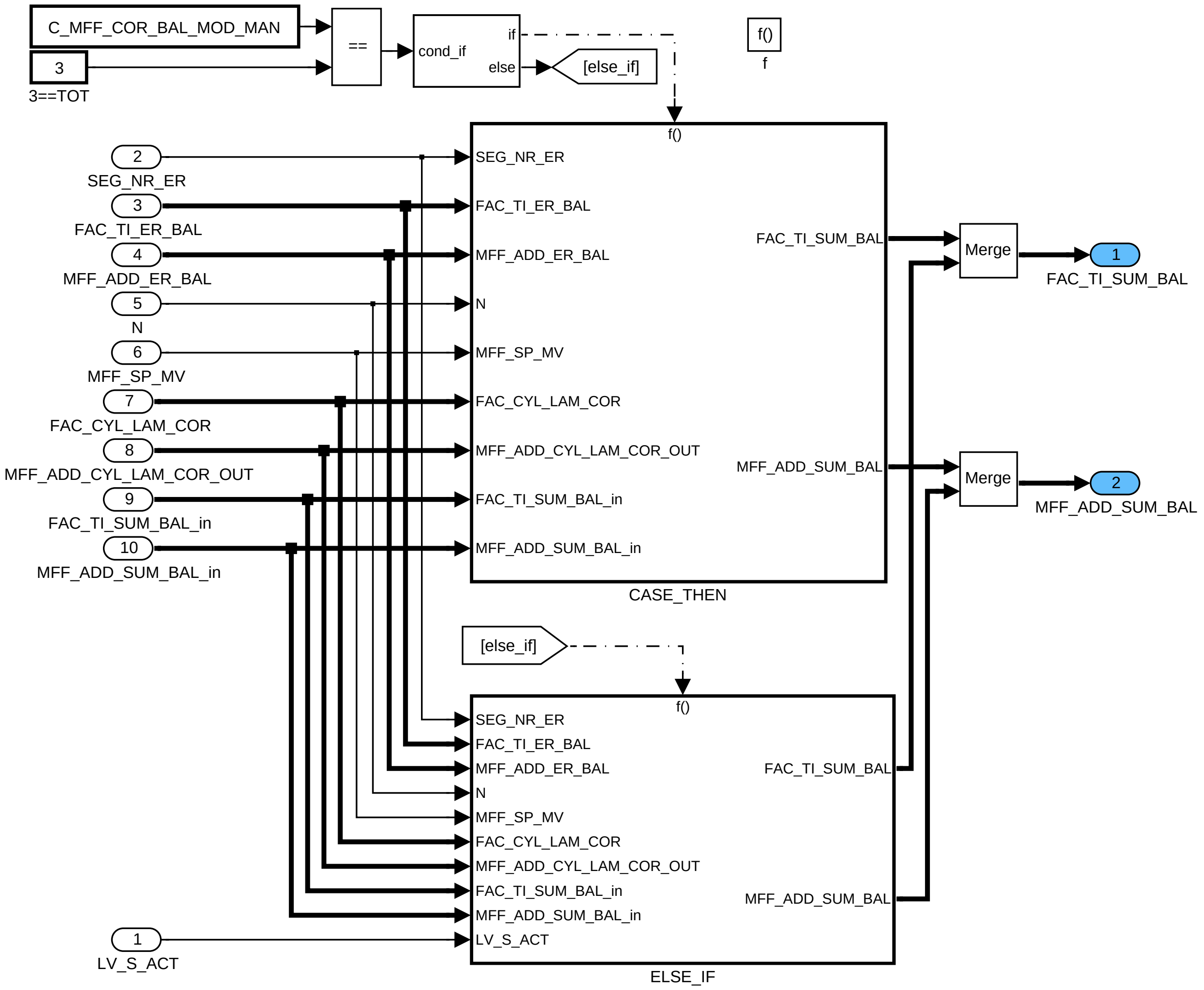


Figure 39 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ ELSE_IF/ ELSE_IF

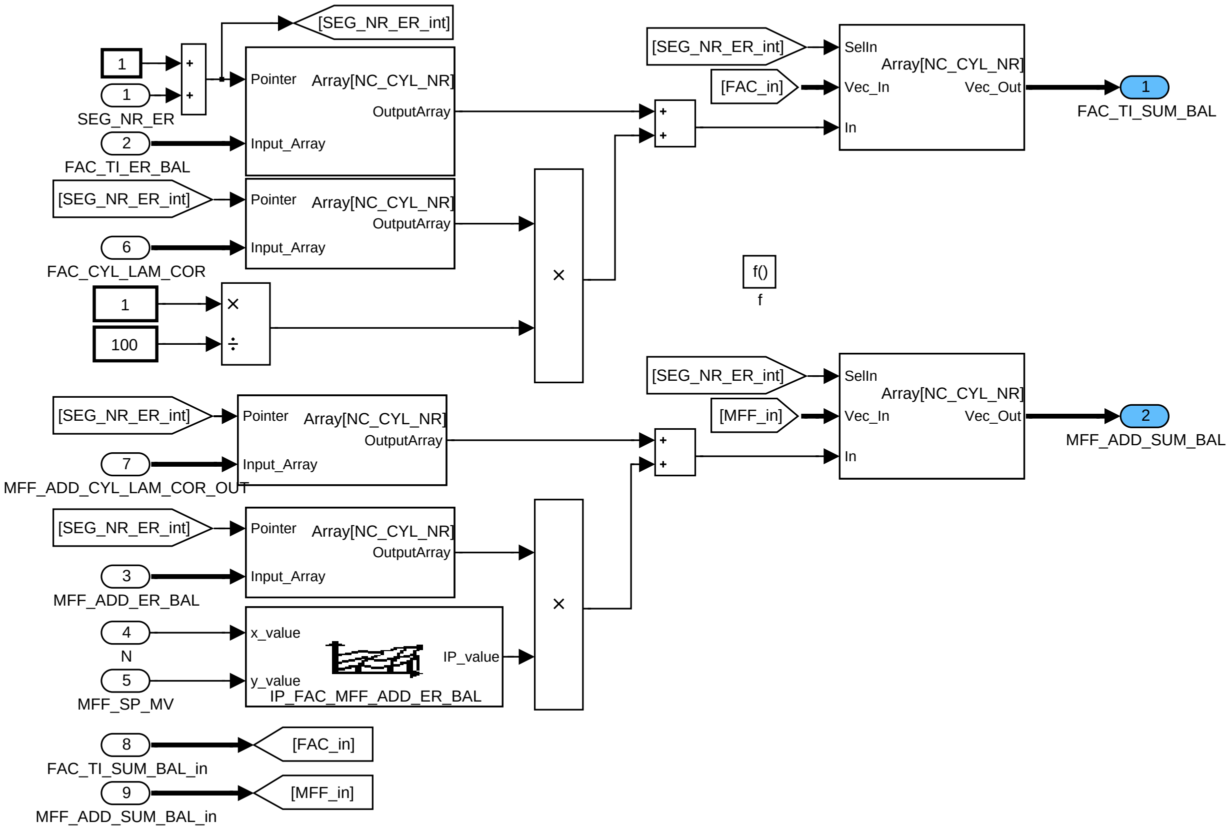


Figure 40 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ ELSE_IF/ ELSE_IF/ CASE_THEN

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF ELSE_IF ELSE_IF ELSE_IF

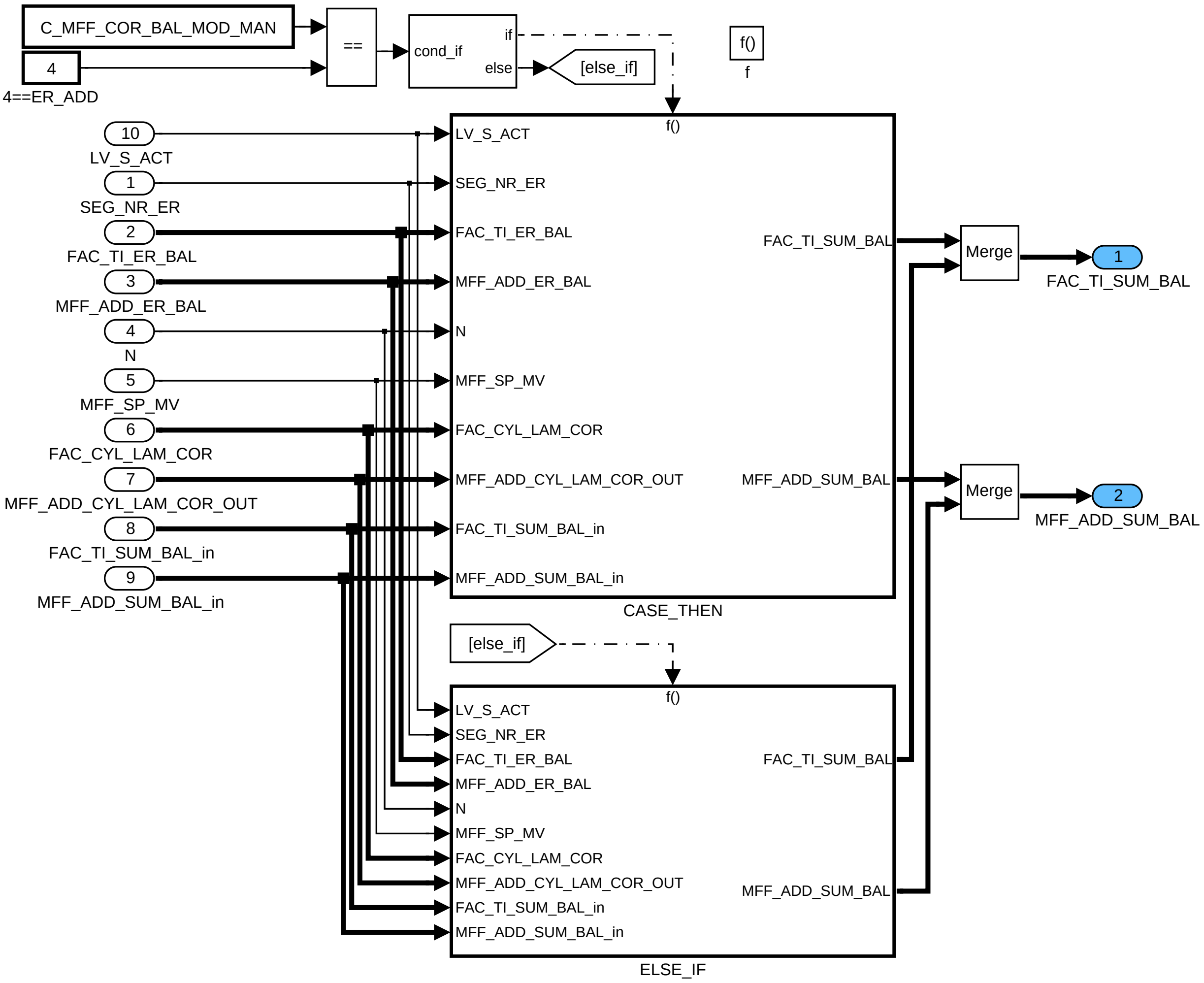


Figure 41 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ ELSE_IF/ ELSE_IF/ ELSE_IF

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF ELSE_IF ELSE_IF ELSE_IF CASE_THEN

//CYBL via LAM only applied in HOM, CYBL via ER applied in STR and partly in HOM

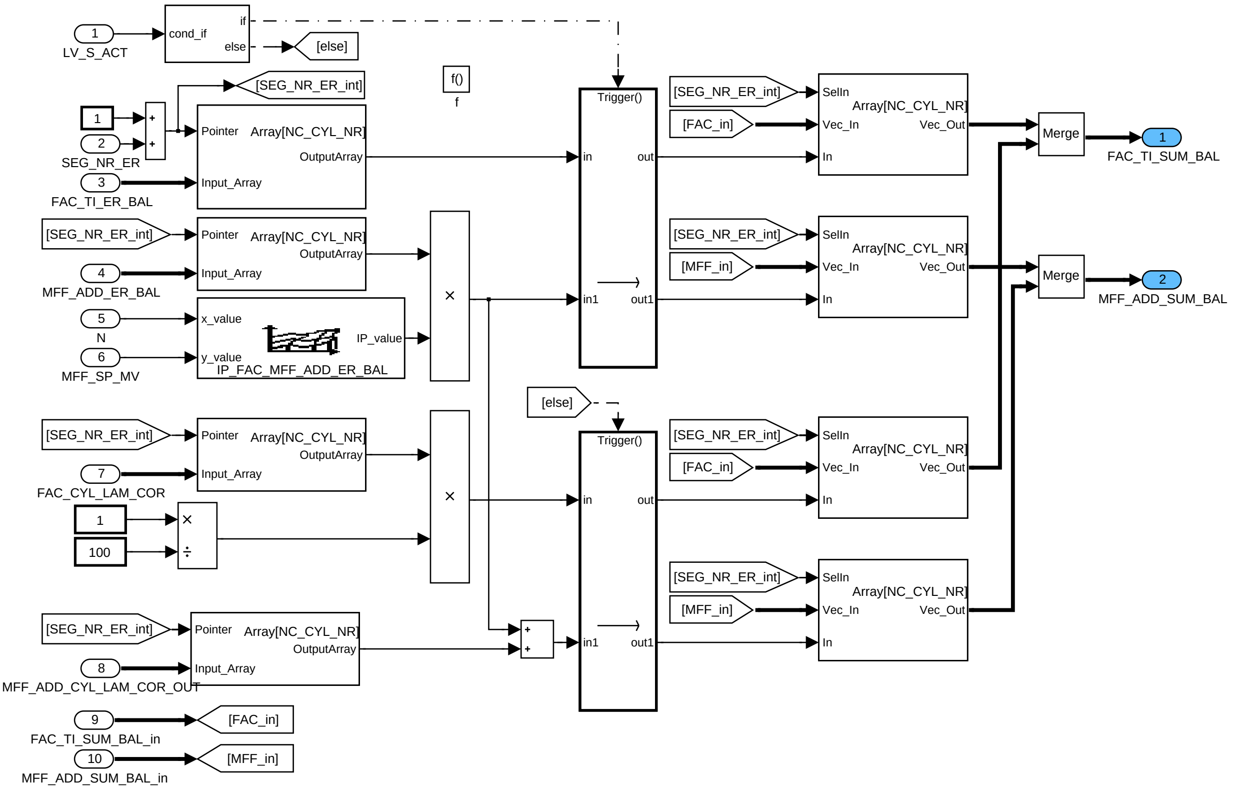


Figure 42 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ ELSE_IF/ ELSE_IF/ ELSE_IF/ CASE_THEN

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF ELSE_IF ELSE_IF ELSE_IF ELSE_IF

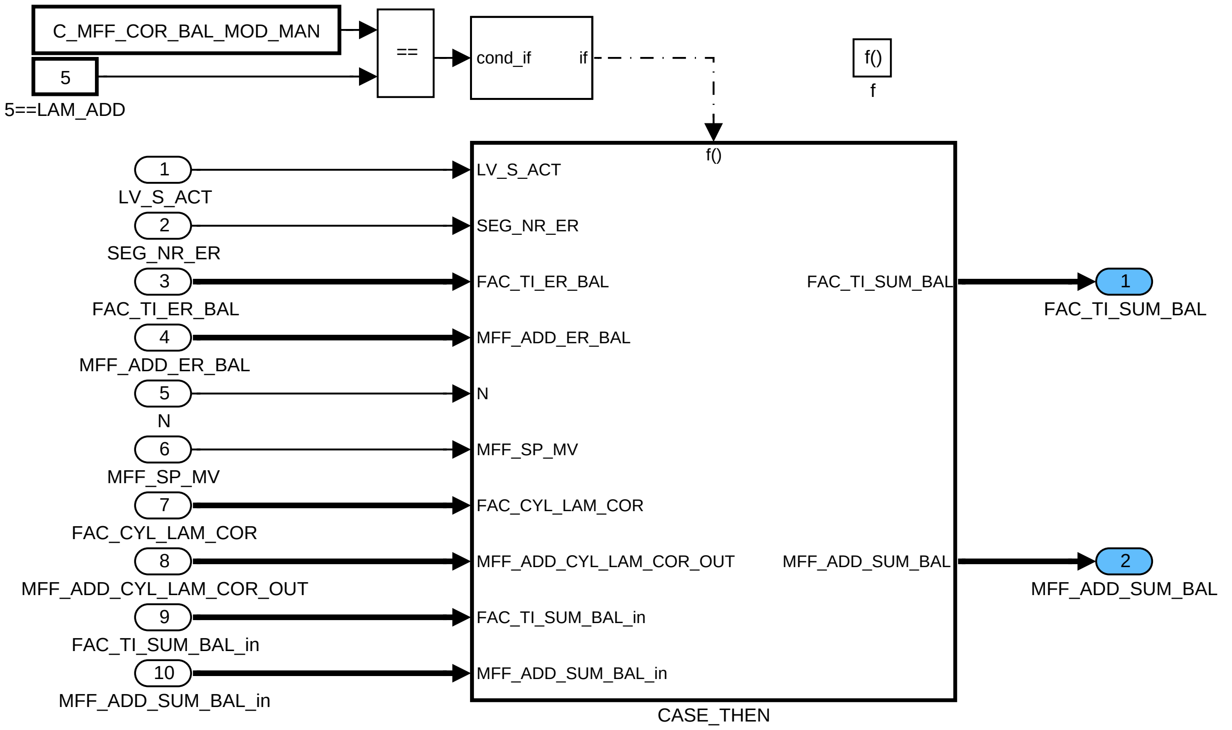


Figure 43 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ ELSE_IF/ ELSE_IF/ ELSE_IF/ ELSE_IF

Calculation of correction factors for cylinder balancing OPERATE_SEG ELSE_IF ELSE_IF ELSE_IF ELSE_IF ELSE_IF CASE_THEN

//CYBL via LAM applied in HOM and STR, CYBL ER applied in STR and partly in HOM

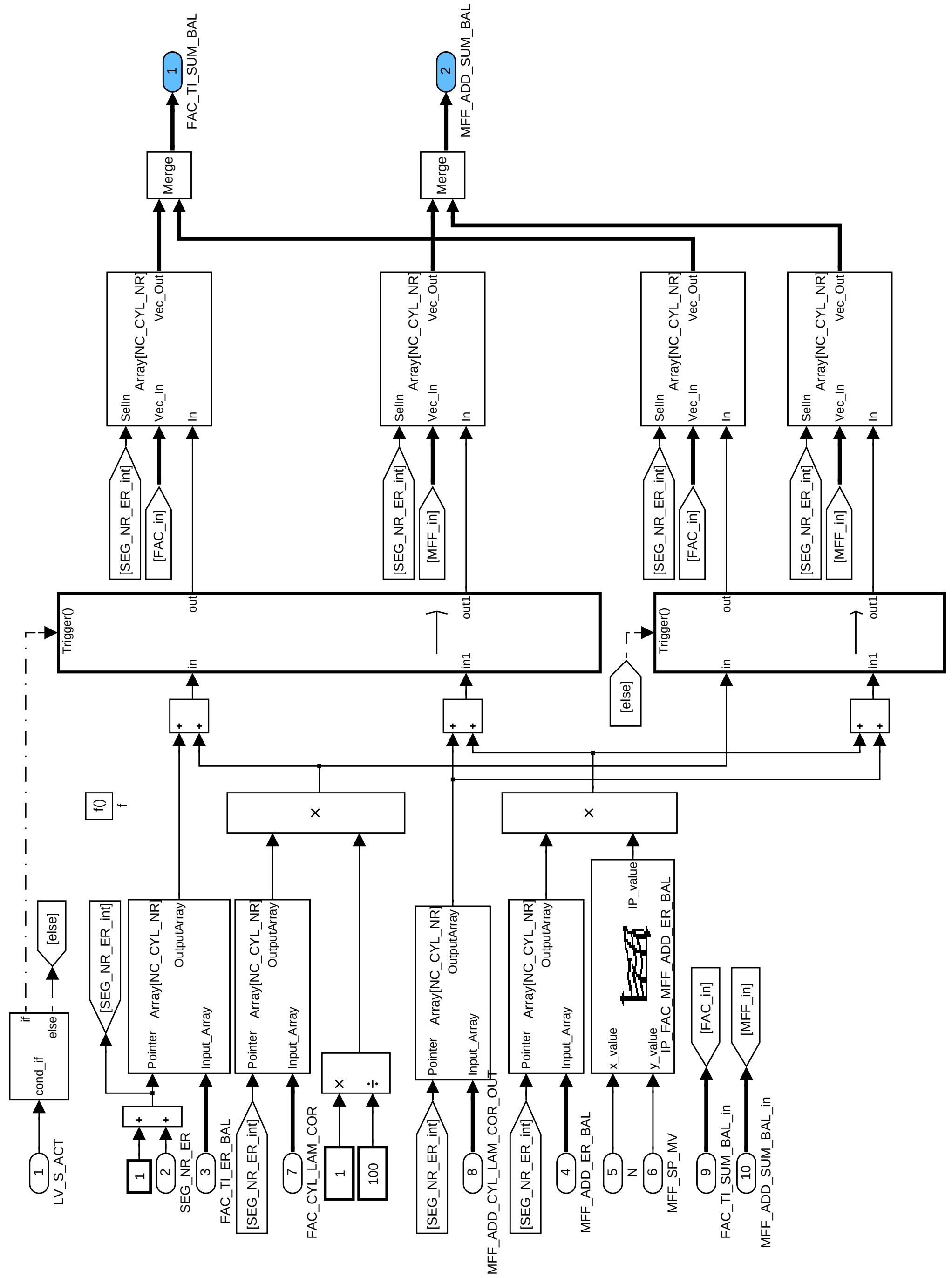


Figure 44 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC/ OPERATE_SEG/ ELSE_IF/ ELSE_IF/ ELSE_IF/ ELSE_IF/ ELSE_IF/ CASE_THEN

Cylinder balancing injection valve energy level adjustment (approximation)

The formula section is only activated if the switch for a manual reset is set to "0". In case of a manual reset (refer: "Manual Reset of cylinder balancing correction values"), the calculations are stopped at once.

The approximation is only enabled while the manual switch for injection valve energy level calculation is set to "Adjustment" or "ADJ&CTL".

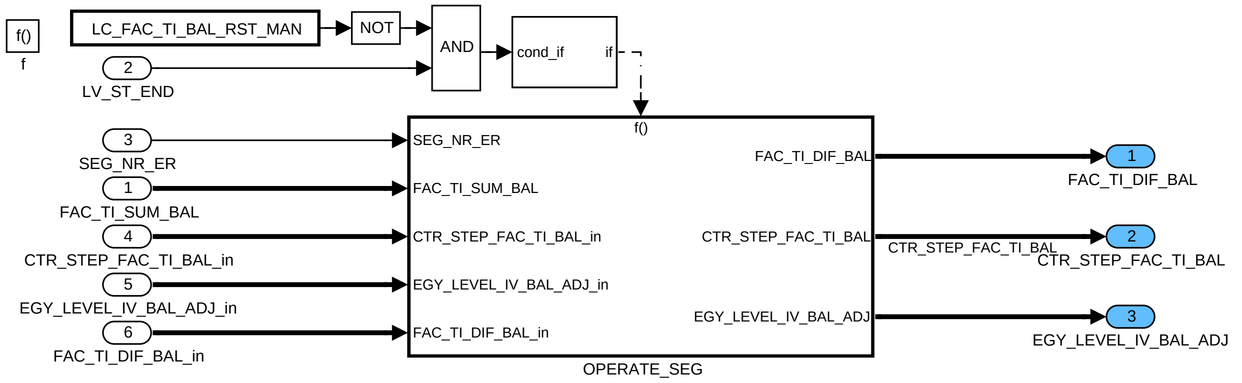


Figure 45 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG

("Calculation of the approximation part disabled") for EGY_LEVEL_IV_BAL_ADJ[m]=0

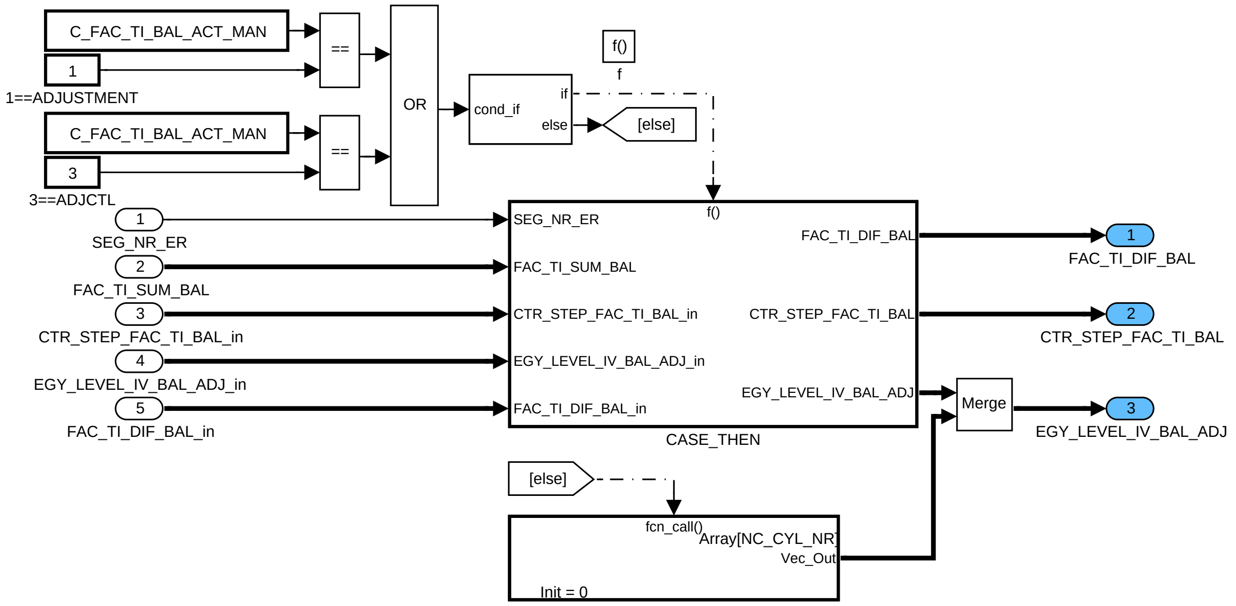


Figure 46 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG CASE_THEN

("Calculation of the approximation part enabled")

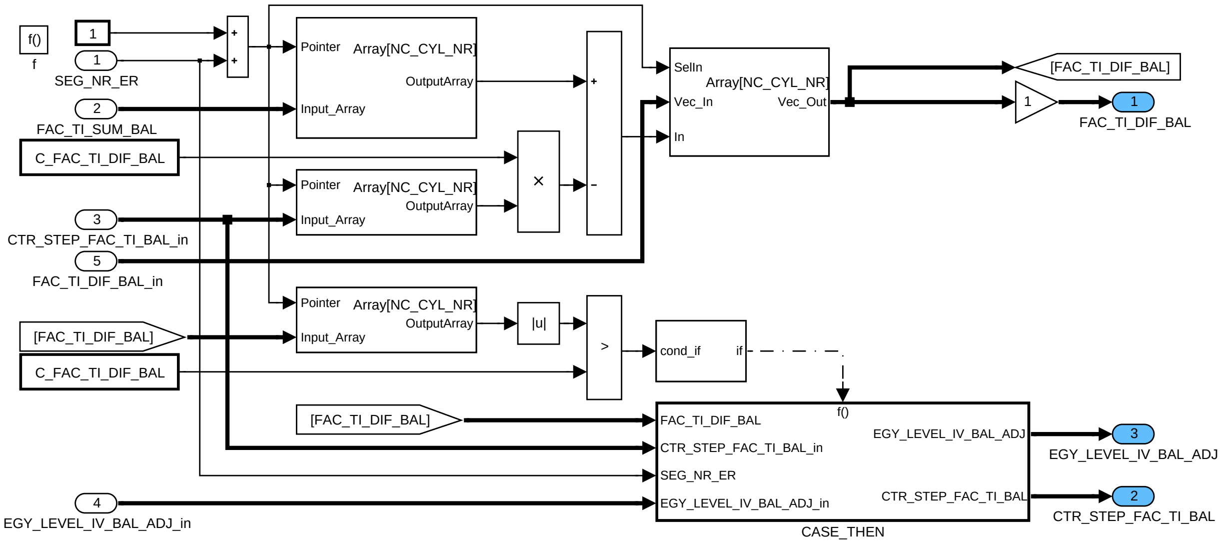


Figure 47 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG/ CASE_THEN

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG CASE_THEN CASE_THEN

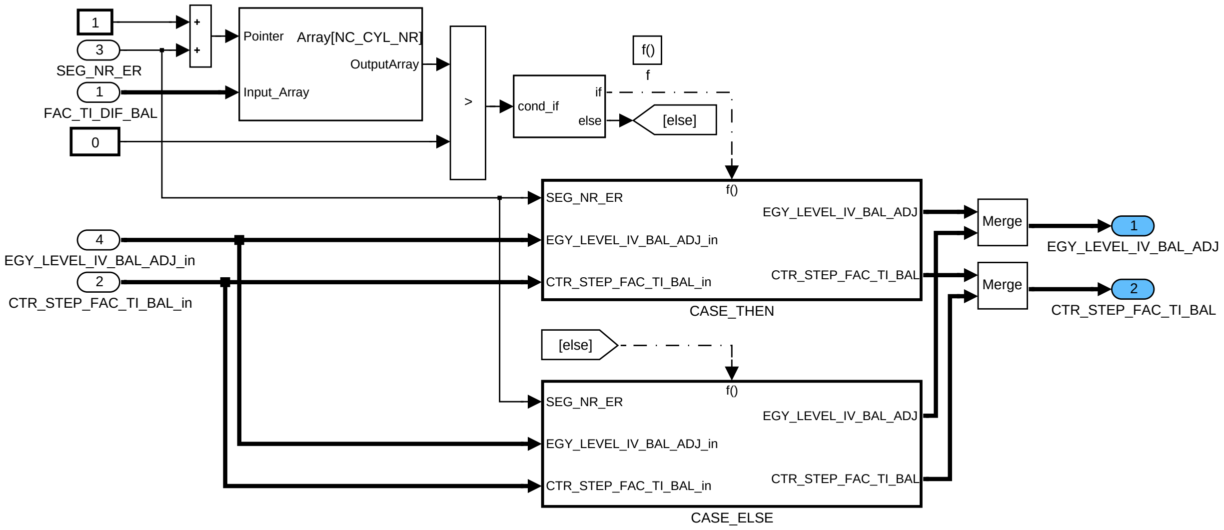


Figure 48 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG/ CASE_THEN/ CASE_THEN

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG CASE_THEN CASE_THEN CASE_THEN

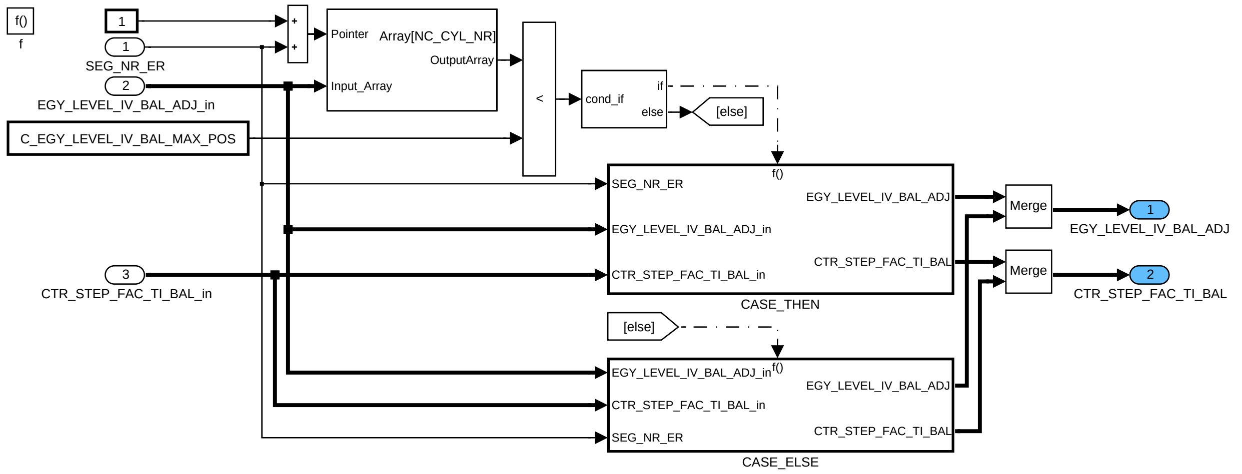


Figure 49 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG/ CASE_THEN/ CASE_THEN/ CASE_THEN

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG CASE_THEN CASE_THEN CASE_THEN CASE_THEN

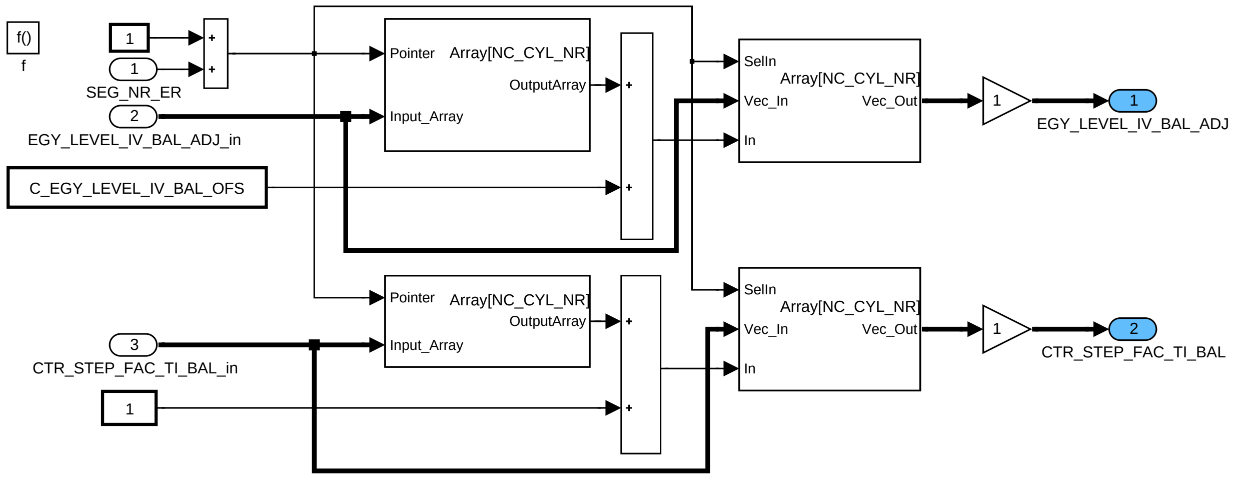


Figure 50 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG/ CASE_THEN/ CASE_THEN/ CASE_THEN/ CASE_THEN

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG CASE_THEN CASE_THEN CASE_THEN CASE_ELSE

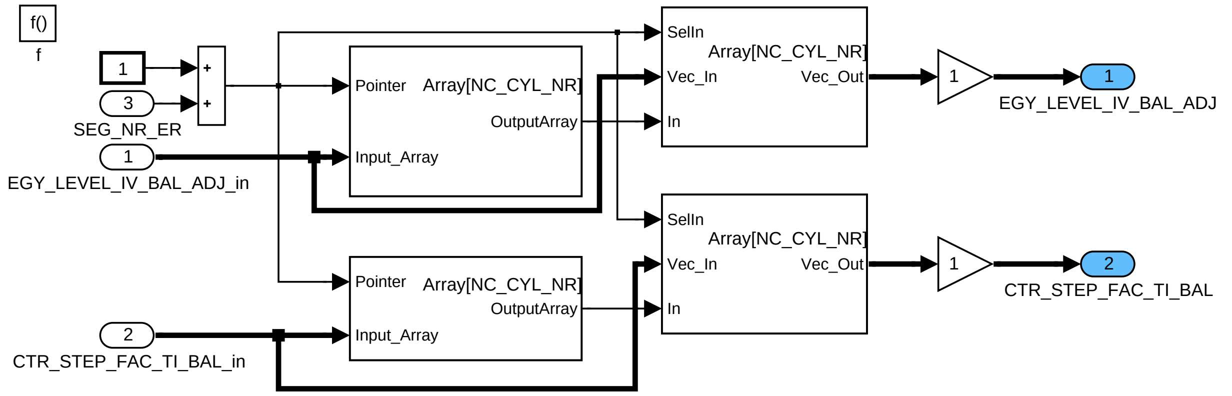


Figure 51 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG/ CASE_THEN/ CASE_THEN/ CASE_THEN/ CASE_ELSE

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG CASE_THEN CASE_THEN CASE_ELSE

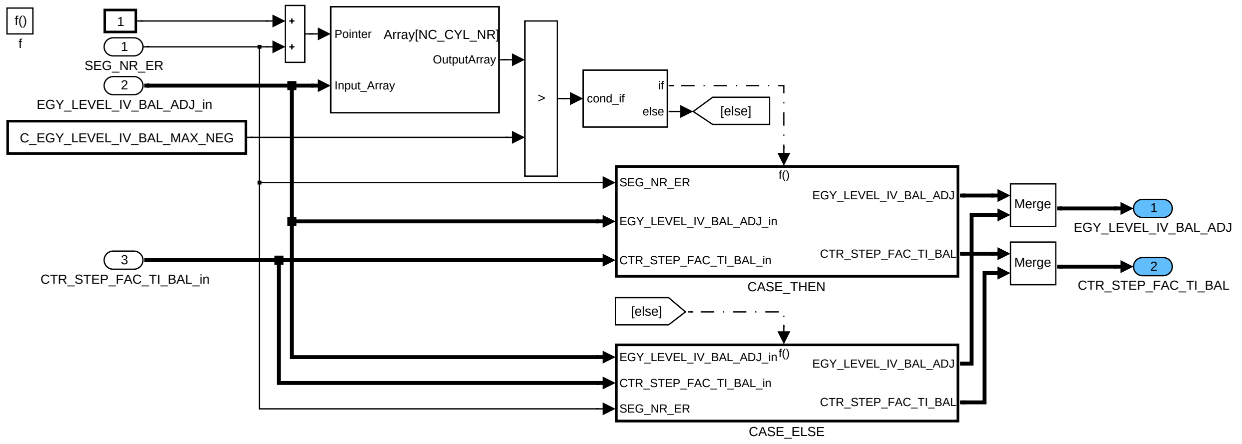


Figure 52 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG/ CASE_THEN/ CASE_THEN/ CASE_ELSE

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG CASE_THEN CASE_THEN CASE_ELSE CASE_THEN

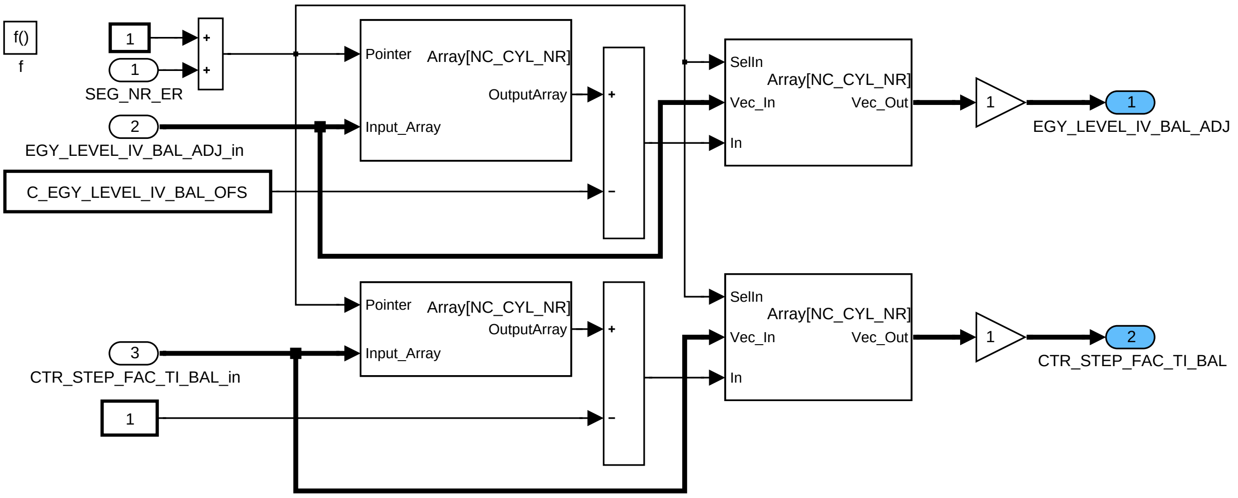


Figure 53 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG/ CASE_THEN/ CASE_THEN/ CASE_ELSE/ CASE_THEN

Cylinder balancing injection valve energy level adjustment (approximation) OPERATE_SEG CASE_THEN CASE_THEN CASE_ELSE CASE_ELSE

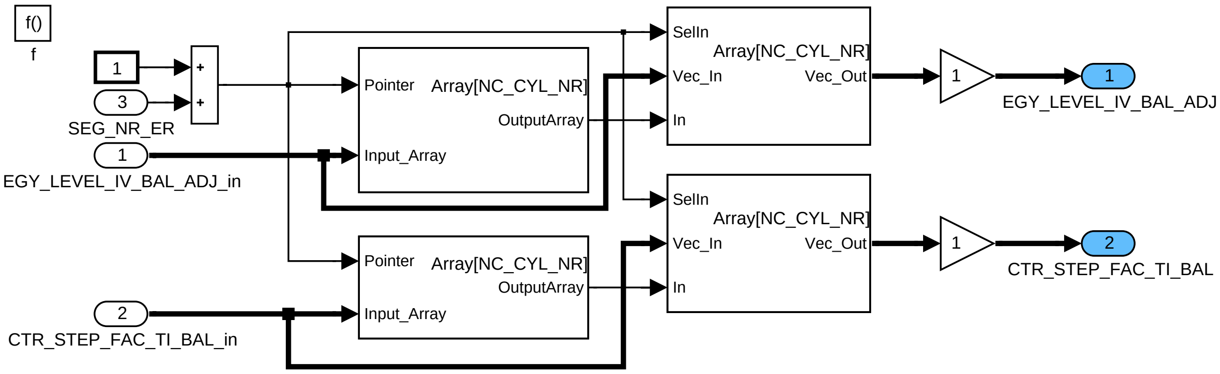


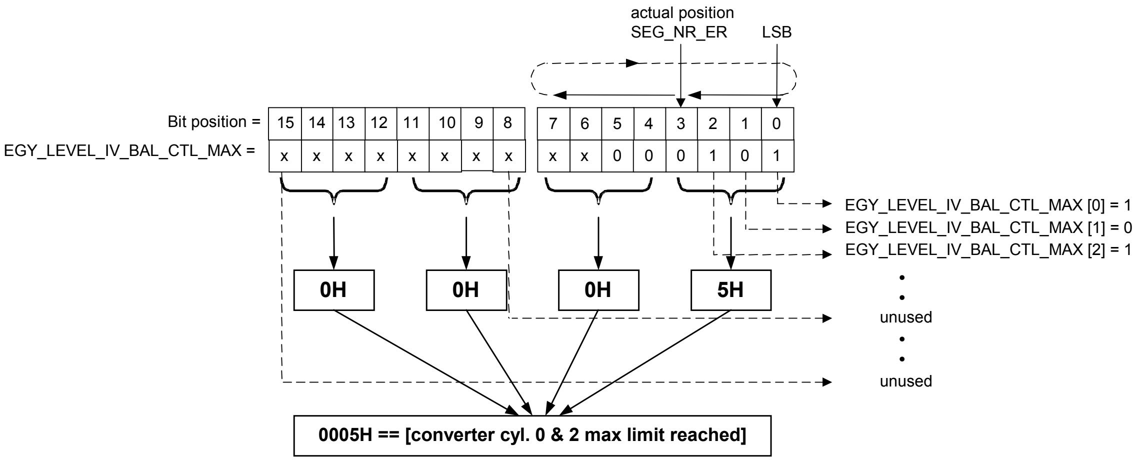
Figure 54 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL/ OPERATE_SEG/ CASE_THEN/ CASE_THEN/ CASE_ELSE/ CASE_ELSE

Cylinder balancing injection valve energy level control

The formula section is only activated if the switch for a manual reset is set to "0". In case of a manual reset (refer: "Manual Reset of cylinder balancing correction values"), the calculations are stopped at once.

The energy level conversion is only enabled while the manual switch for injection valve energy level calculation is set to "Control" or "ADJ&CTL".

Signalization of converter maximum output value reached



Note: always the complete WORD (16BIT) is displayed as visible value

As soon as an energy level conversion of one cylinder is reaching its maximum, a corresponding value is set to "1" to indicate the related cylinder. To save memory resources, the information for all cylinders is located within one output date (WORD). Each BIT information within the WORD is used to indicate one cylinder. It is possible to handle up to 16 cylinders with this method.

For a description more in detail please have a look to the signal flow diagram above. As example, a six-cylinder engine with two converters, which have reached the limits is shown.

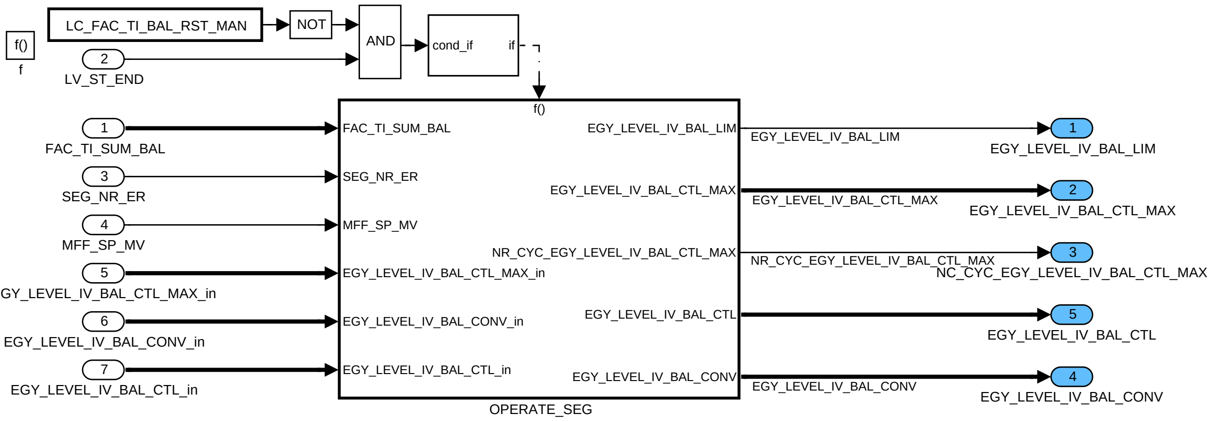


Figure 55 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL_CTL

Cylinder balancing injection valve energy level control OPERATE_SEG

("Calculation of the converter part disabled") (CASE_ELSE)

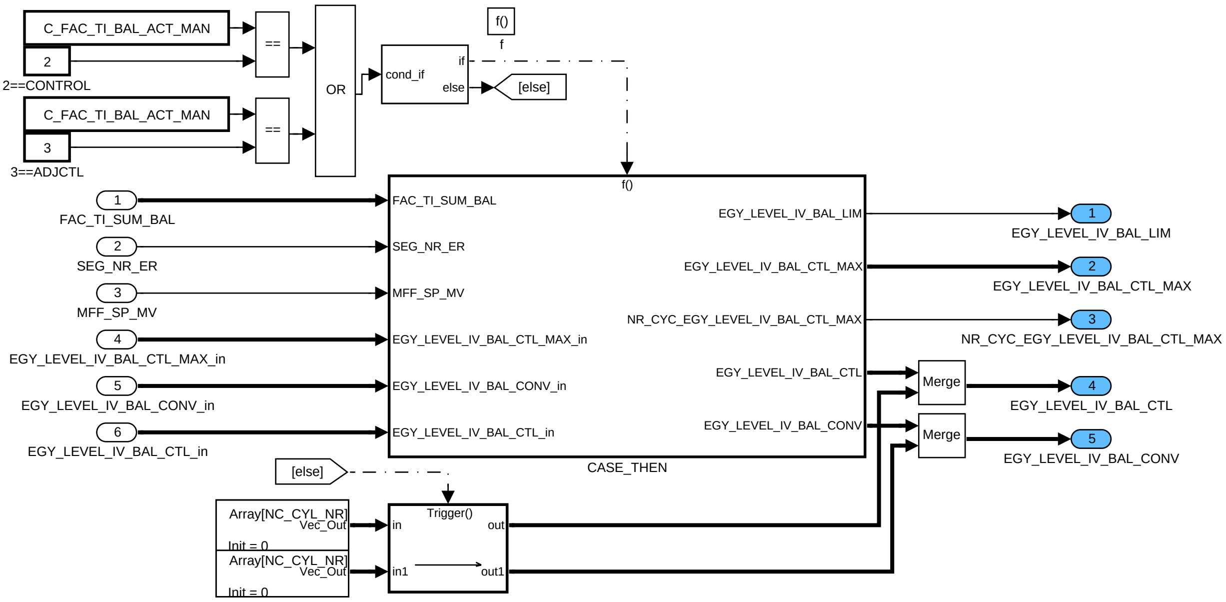


Figure 56 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL_CTL/ OPERATE_SEG

Cylinder balancing injection valve energy level control OPERATE_SEG CASE_THEN

("Calculation of the converter part enabled")

Task of this section is to convert the multiple correction factor provided by the cylinder balancing functions into a corresponding energy level value. The energy level signal is used to adjust the injector needle lift.

The outputs of the converter are limited to a calibrateable threshold. If one converter output has reached the limit a flag is set to one and the corresponding segment number is displayed.

The final converter outputs are corrected with a multiple factor out of an interpolation table to compensate different conversion rates for the energy levels at all engine operating areas.

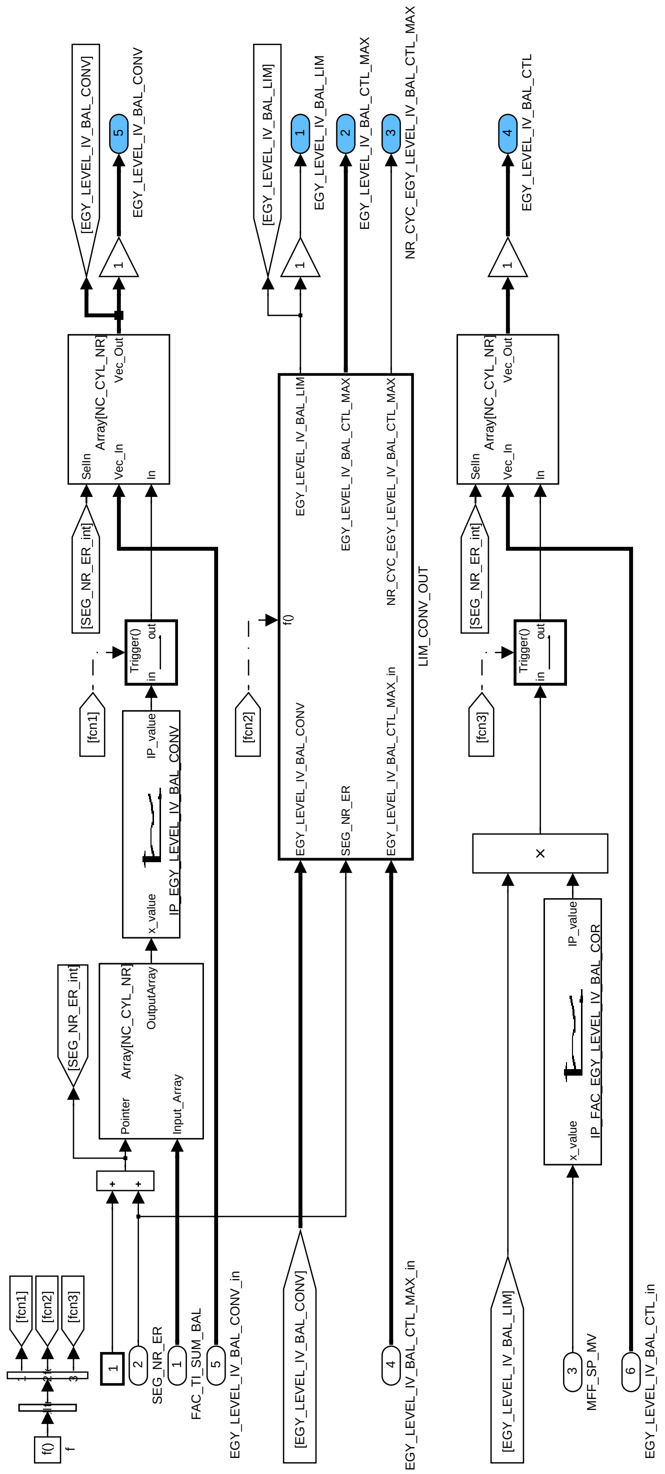


Figure 57 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL_CTL/ OPERATE_SEG/ CASE_THEN

Cylinder balancing injection valve energy level control OPERATE_SEG CASE_THEN LIM_CONV_OUT

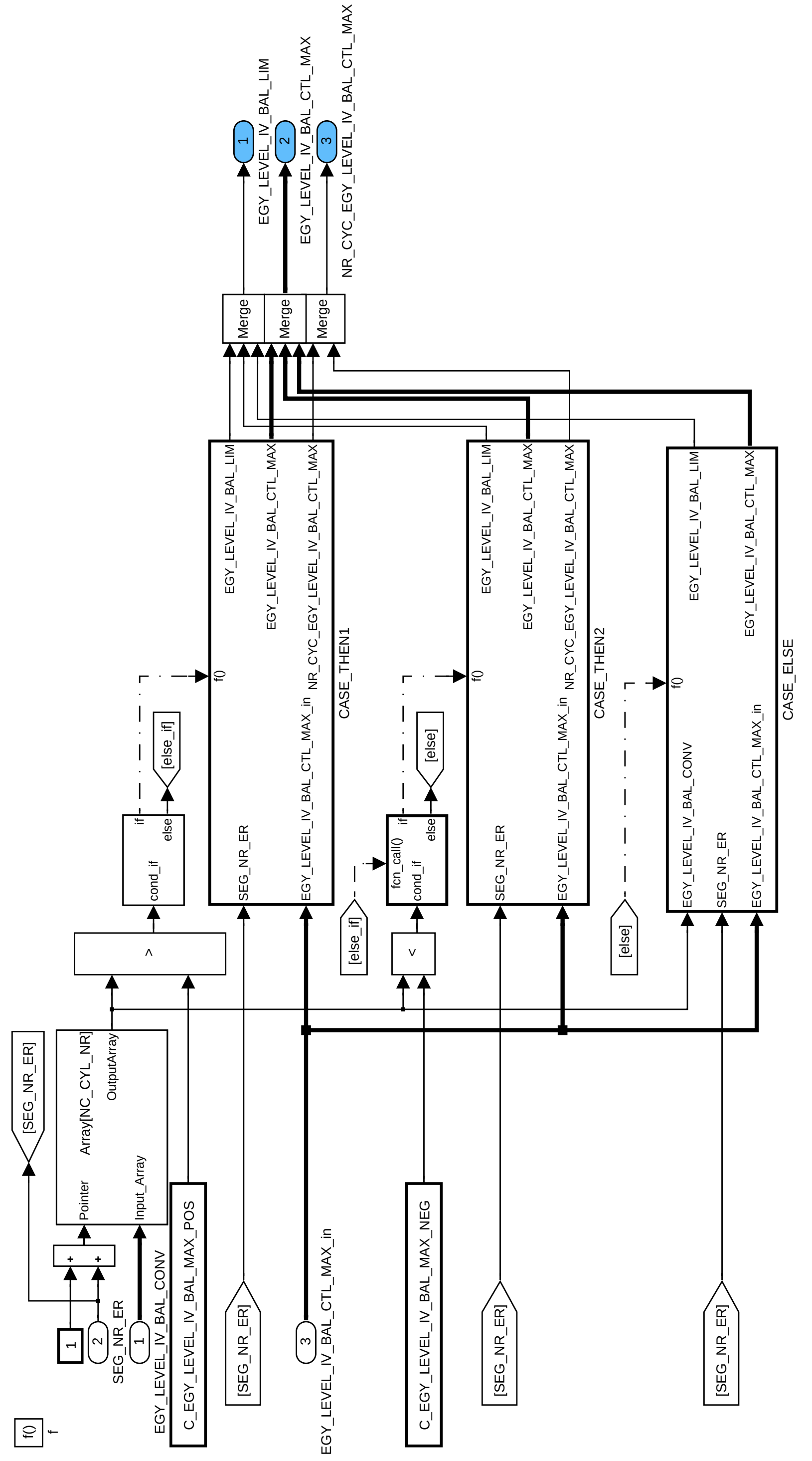


Figure 58 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL_CTL/ OPERATE_SEG/ CASE_THEN/ LIM_CONV_OUT

Cylinder balancing injection valve energy level control OPERATE_SEG CASE_THEN LIM_CONV_OUT CASE_THEN1

//corresponding segment number where the controller has reached the limit

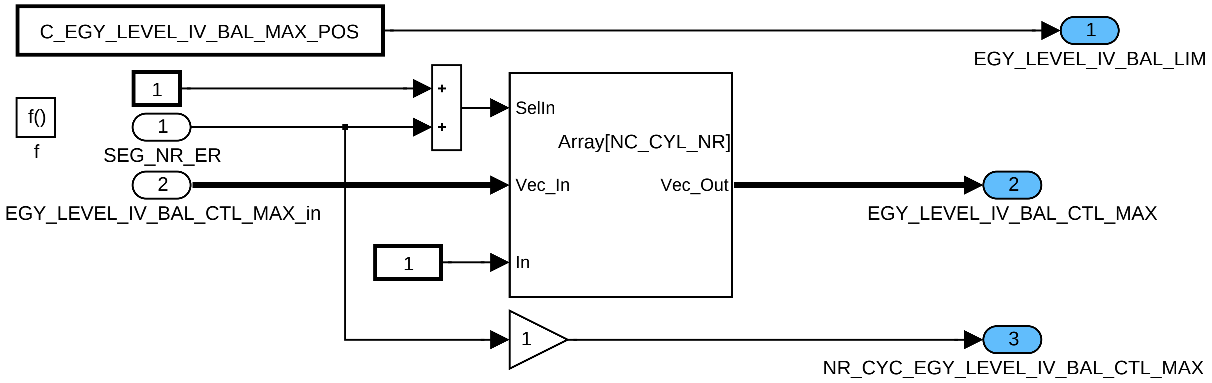


Figure 59 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL_CTL/ OPERATE_SEG/ CASE_THEN/ LIM_CONV_OUT/ CASE_THEN1

Cylinder balancing injection valve energy level control OPERATE_SEG CASE_THEN LIM_CONV_OUT CASE_THEN2

//corresponding segment number where the controller has reached the limit

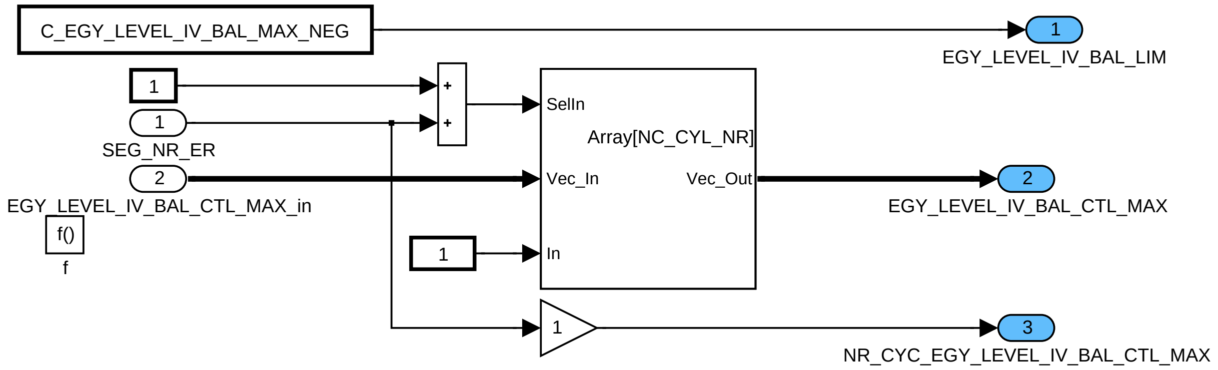


Figure 60 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL_CTL/ OPERATE_SEG/ CASE_THEN/ LIM_CONV_OUT/ CASE_THEN2

Cylinder balancing injection valve energy level control OPERATE_SEG CASE_THEN LIM_CONV_OUT CASE_ELSE

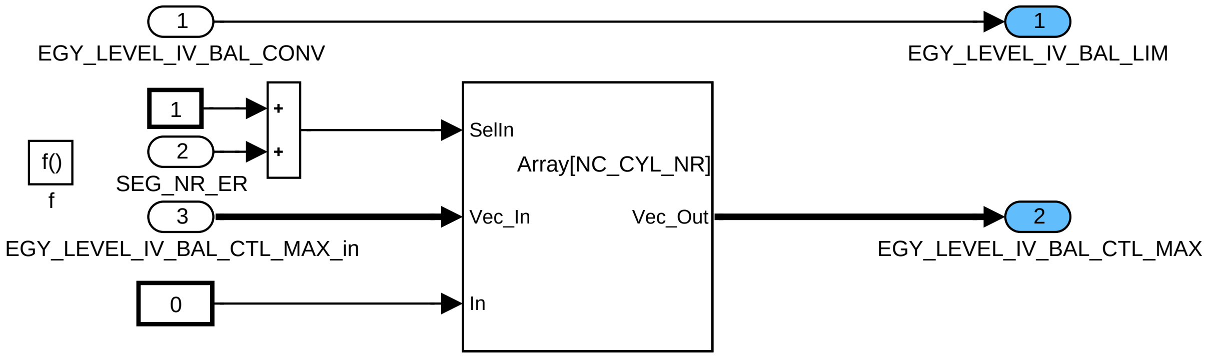


Figure 61 CYBL_M30704P01/ OPM/ OPERATE_SEG/ INJ_VAL_EGY_LEVEL_CTL/ OPERATE_SEG/ CASE_THEN/ LIM_CONV_OUT/ CASE_ELSE

Calculation of overall correction factors for cylinder balancing at engine run

The formula section is only activated if the switch for a manual reset is set to "0". In case of a manual reset (: "Manual Reset of cylinder balancing correction values"), the calculations are stopped at once.

Finally, the calculation of the total cylinder balancing correction output values takes place. The output values are used to adjust the wanted cylinder individual corrections. All injection valve correction values are calculated at any engine operating area depending of the chosen control strategy.

To avoid high cylinder to cylinder torque differences, especially at first activation after engine start and at cylinder balancing via ER activation, all total cylinder correction values are calculated within one segment. Later on, only the correction values of the current valid segment is calculated. The control is realized by a calibrateable for-loop with adjustable "start" and "stop" parameters.

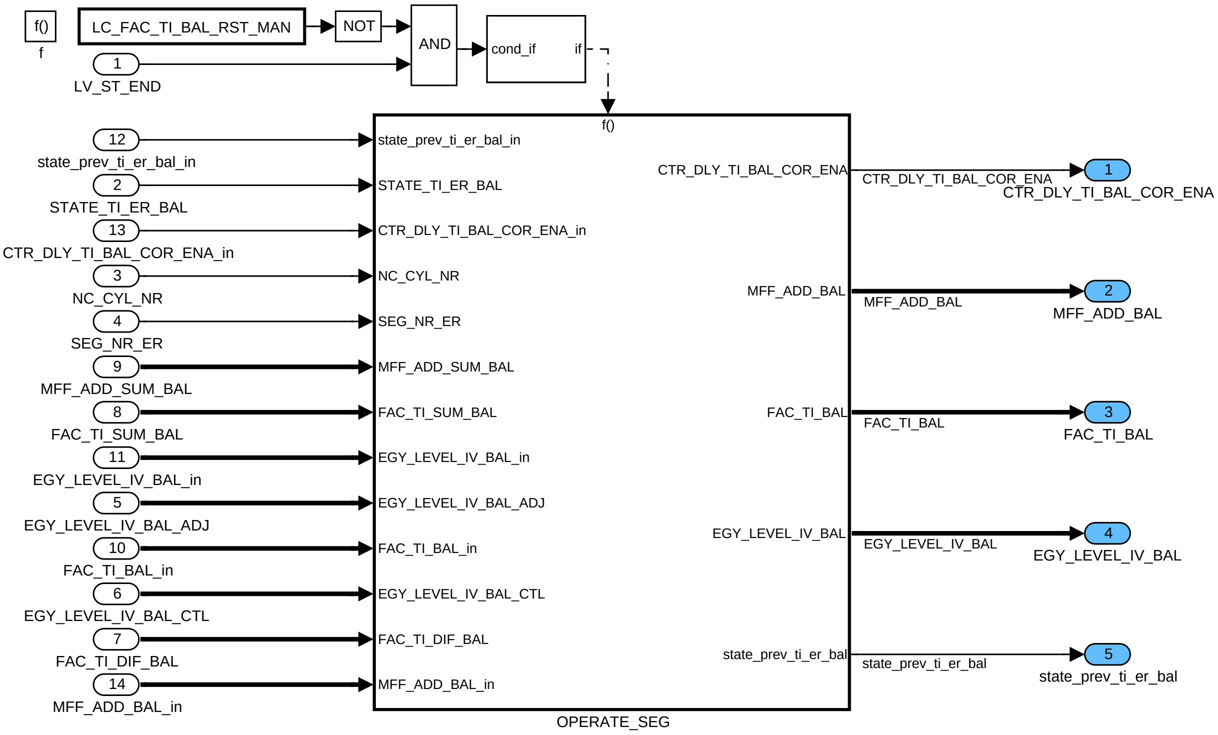


Figure 62 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG

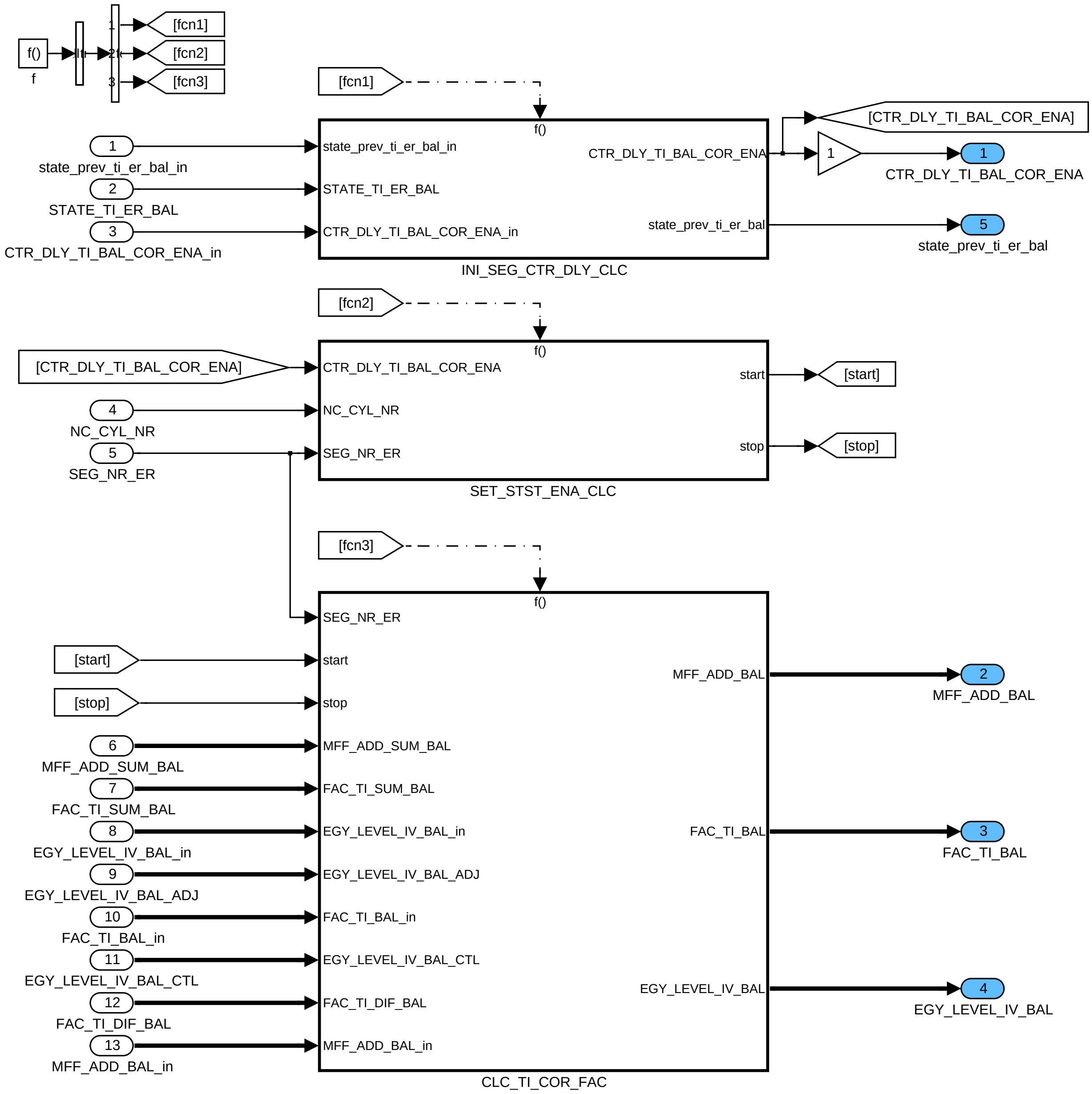


Figure 63 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG INI_SEG_CTR_DLY_CLC

Initialization of segment counter to delay calculation of overall TI correction factors at engine run:

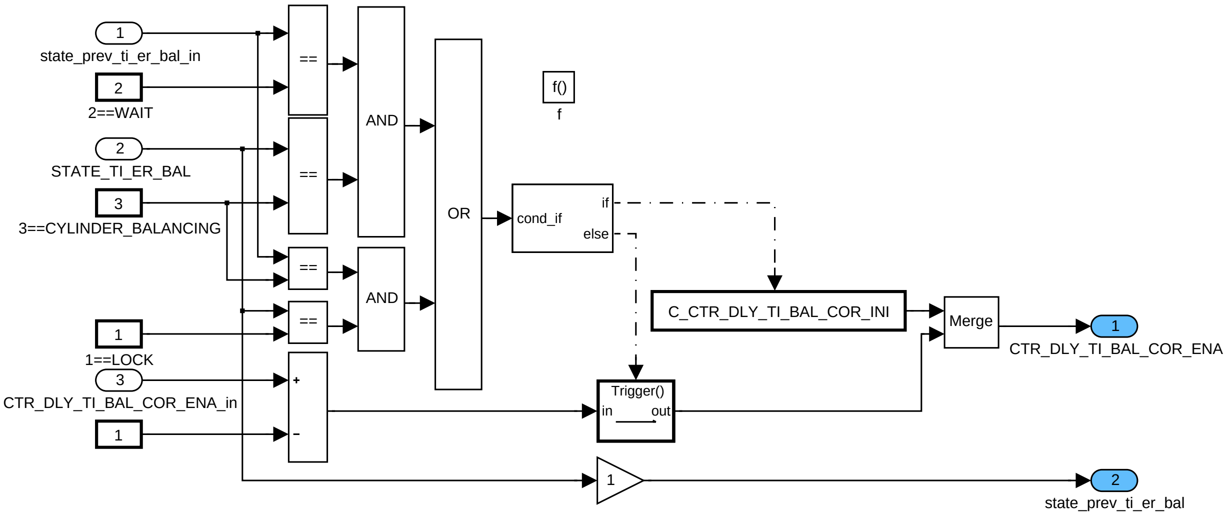


Figure 64 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ INI_SEG_CTR_DLY_CLC

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG SET_STST_ENA_CLC

Setting of "start – "stop" – parameter to enable calculation of overall TI correction factors at engine run:

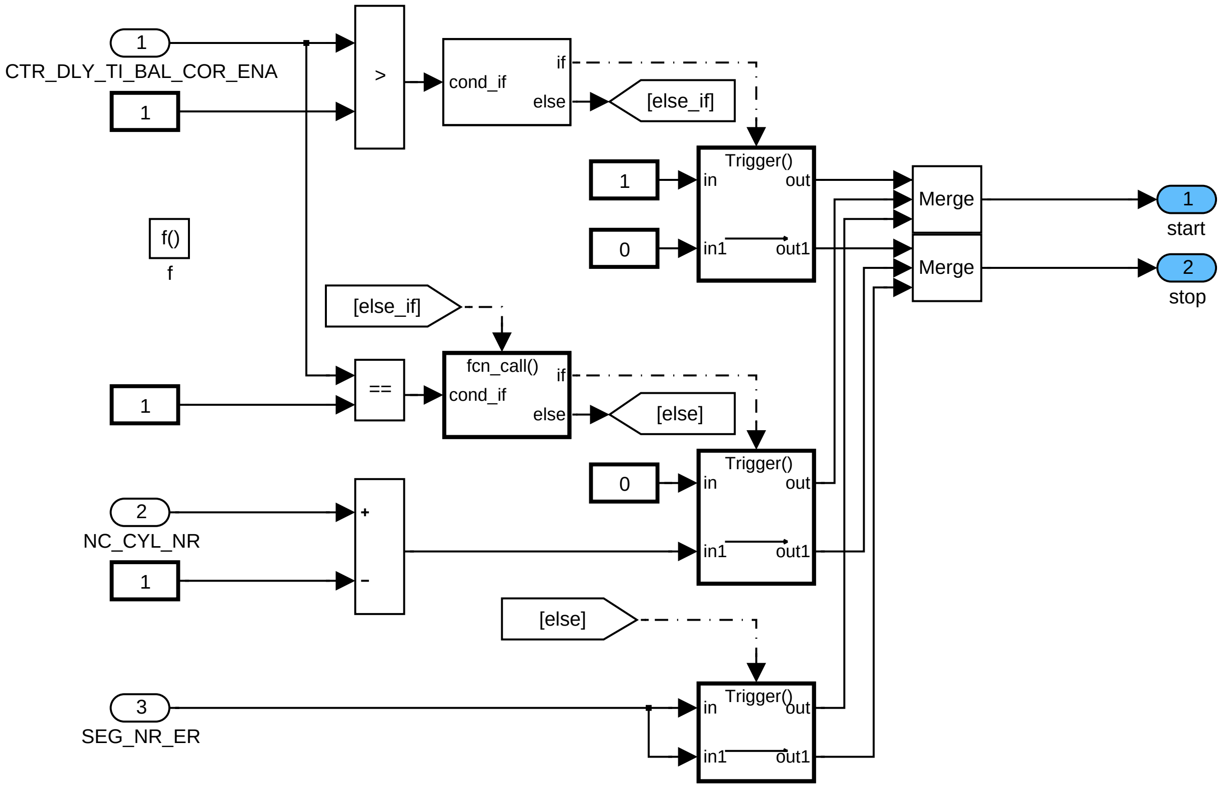


Figure 65 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ SET_STST_ENA_CLC

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC

Calculation of overall TI correction factors (additive, multiple, energy levels) at engine run:

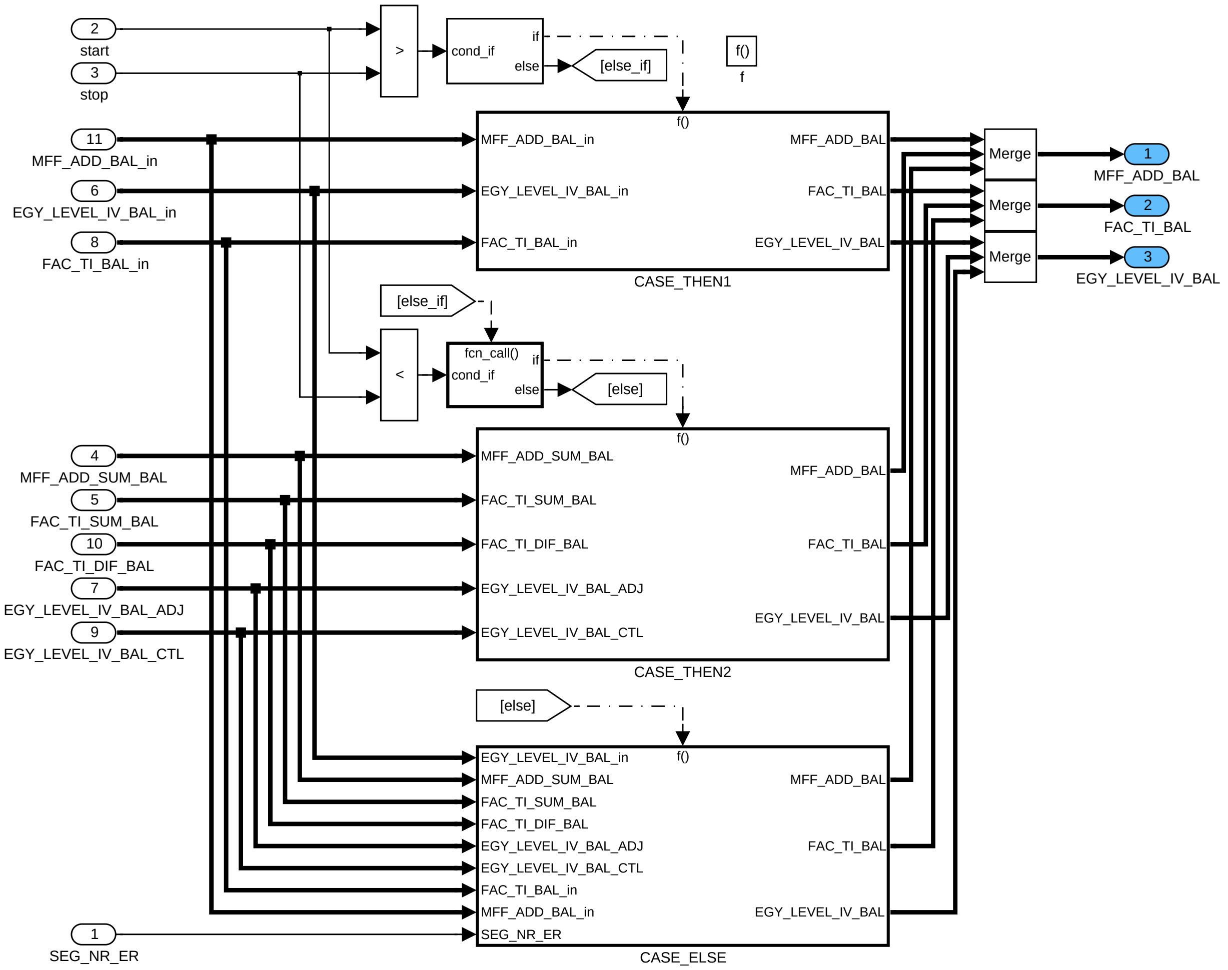


Figure 66 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_THEN1

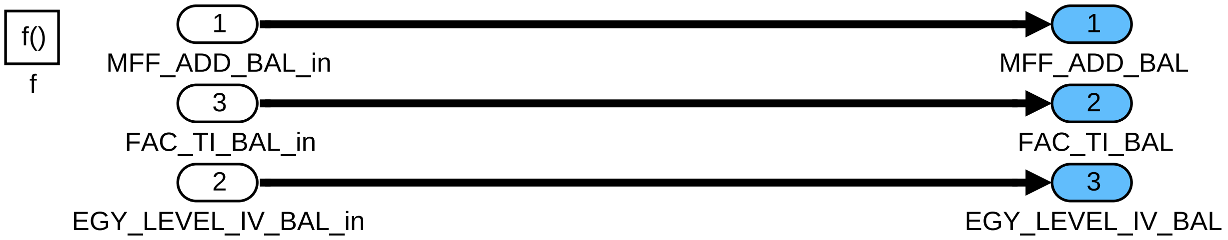


Figure 67 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_THEN1

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_THEN2

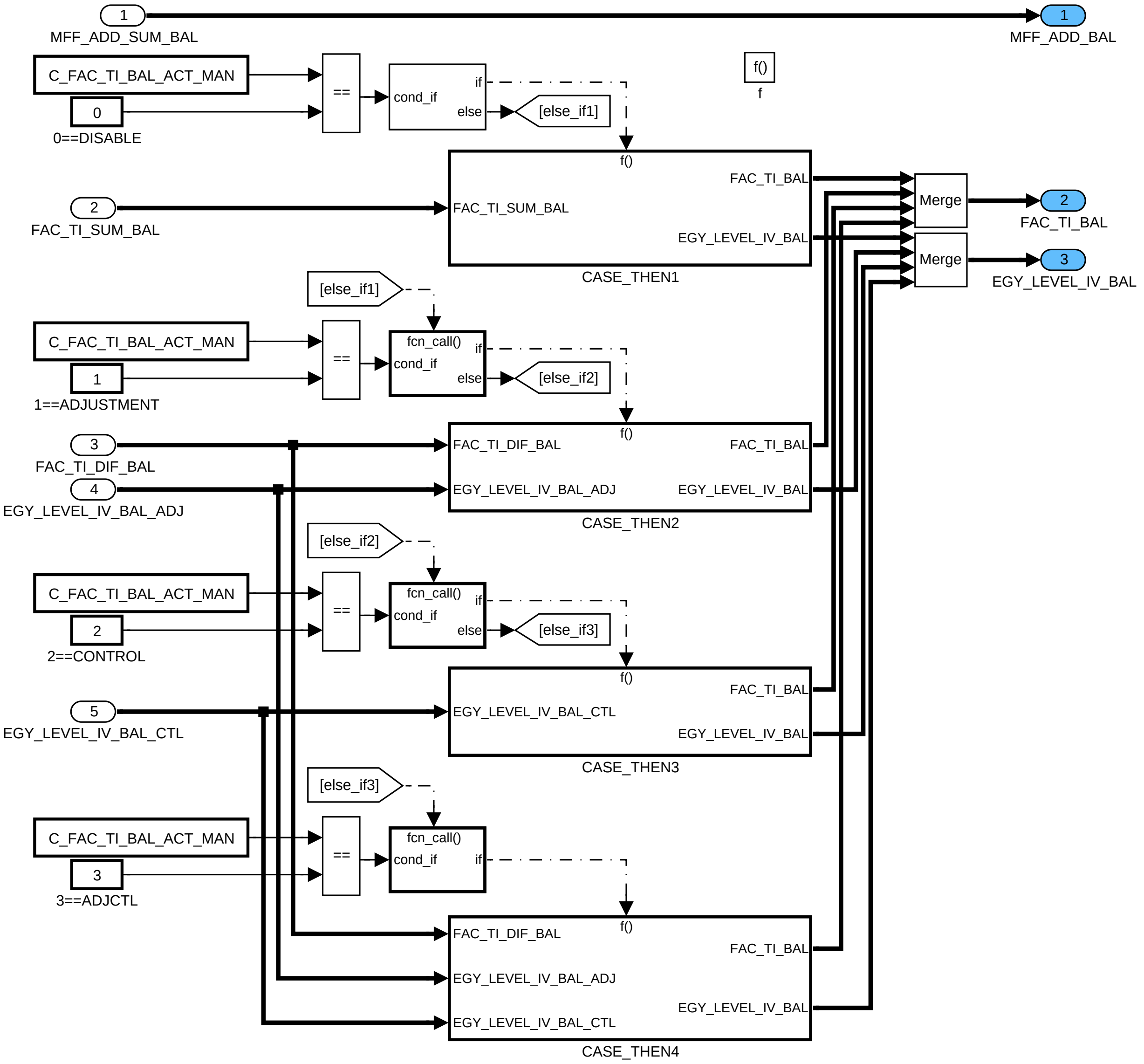


Figure 68 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_THEN2

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_THEN2 CASE_THEN1

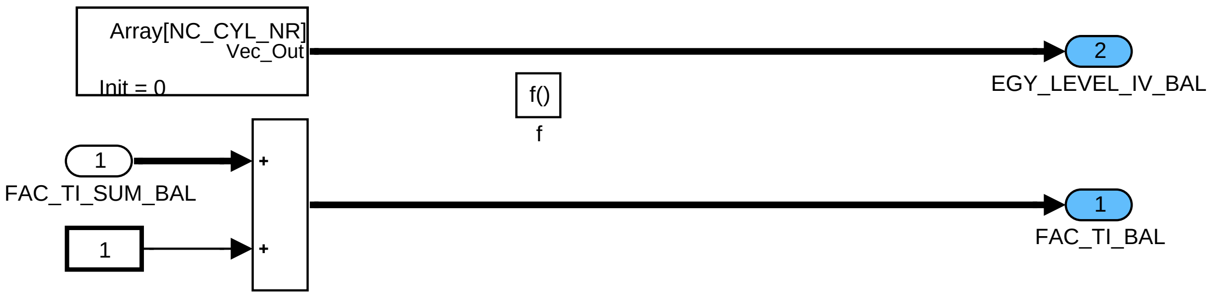


Figure 69 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_THEN2/ CASE_THEN1

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_THEN2 CASE_THEN2

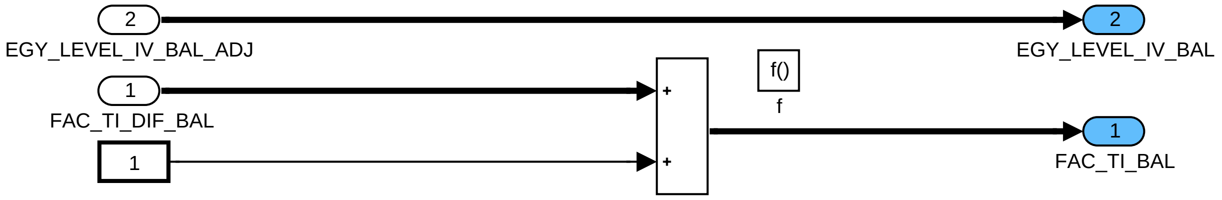


Figure 70 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_THEN2/ CASE_THEN2

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_THEN2 CASE_THEN3

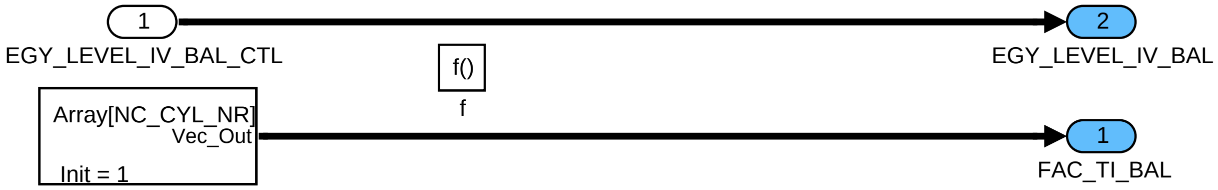


Figure 71 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_THEN2/ CASE_THEN3

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_THEN2 CASE_THEN4

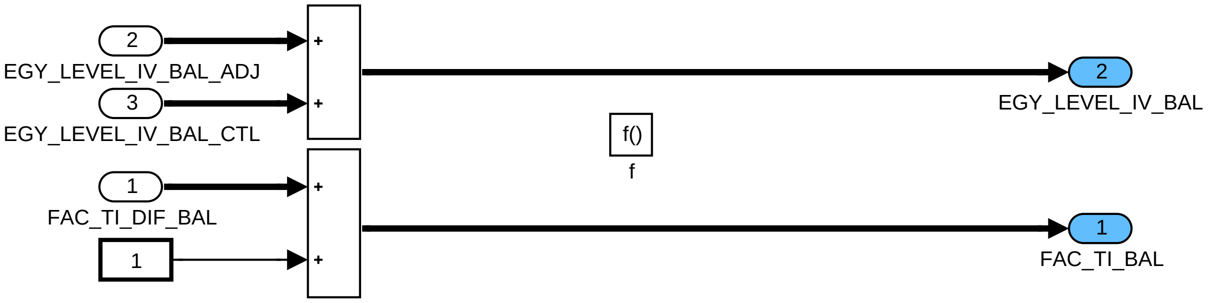


Figure 72 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_THEN2/ CASE_THEN4

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_ELSE

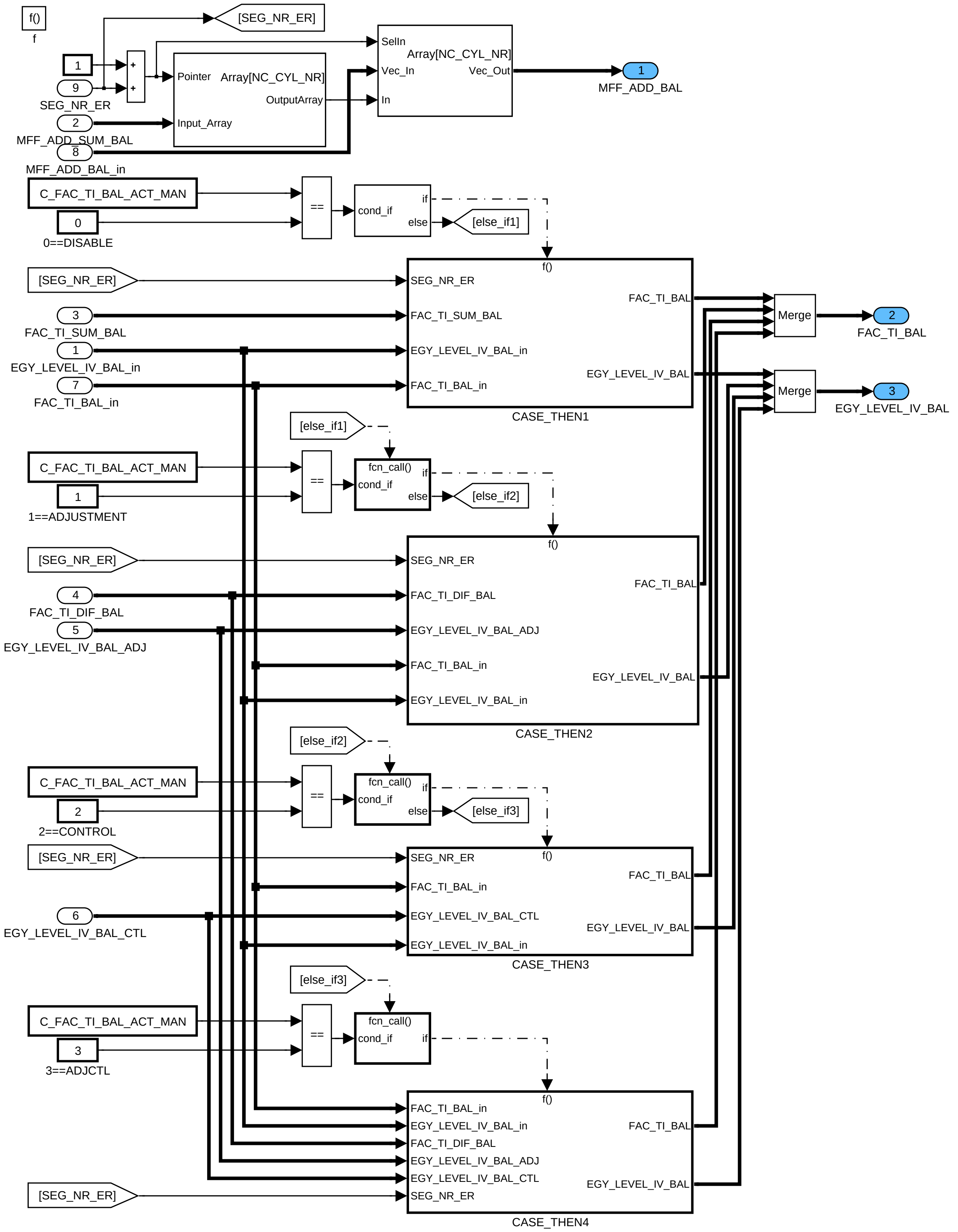


Figure 73 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_ELSE

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_ELSE CASE_THEN1

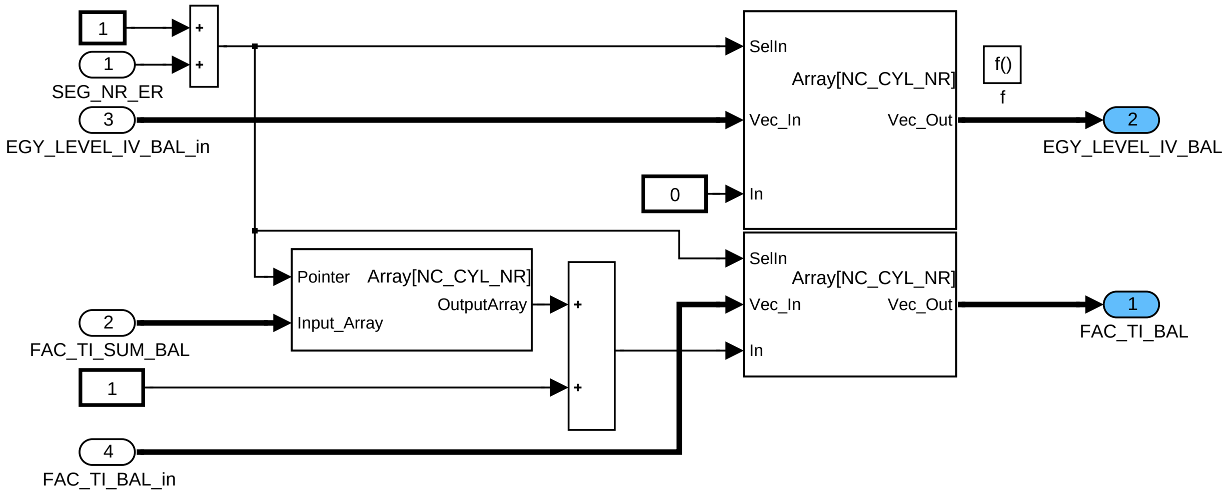


Figure 74 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_ELSE/ CASE_THEN1

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_ELSE CASE_THEN2

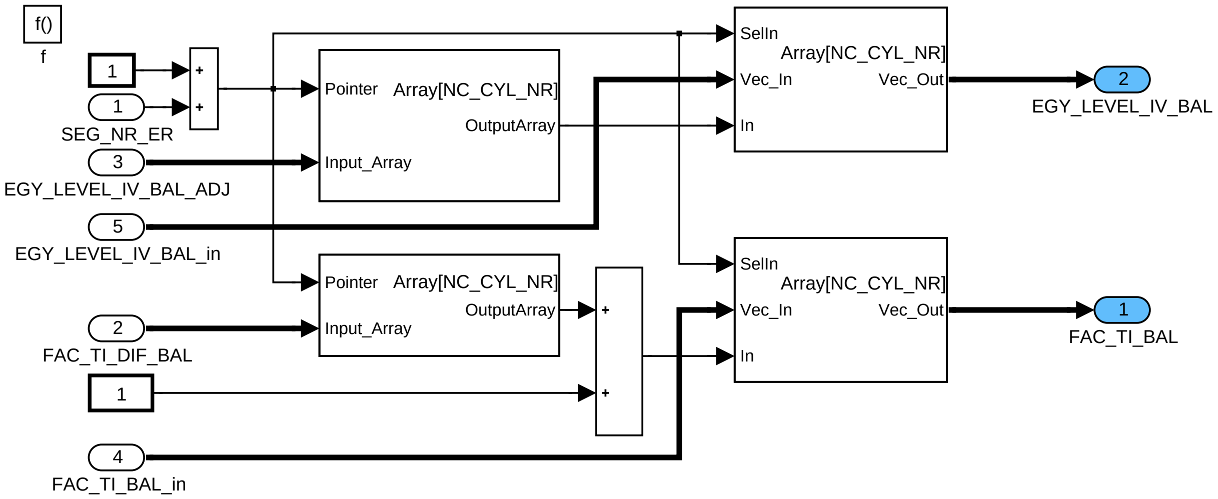


Figure 75 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_ELSE/ CASE_THEN2

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_ELSE CASE_THEN3

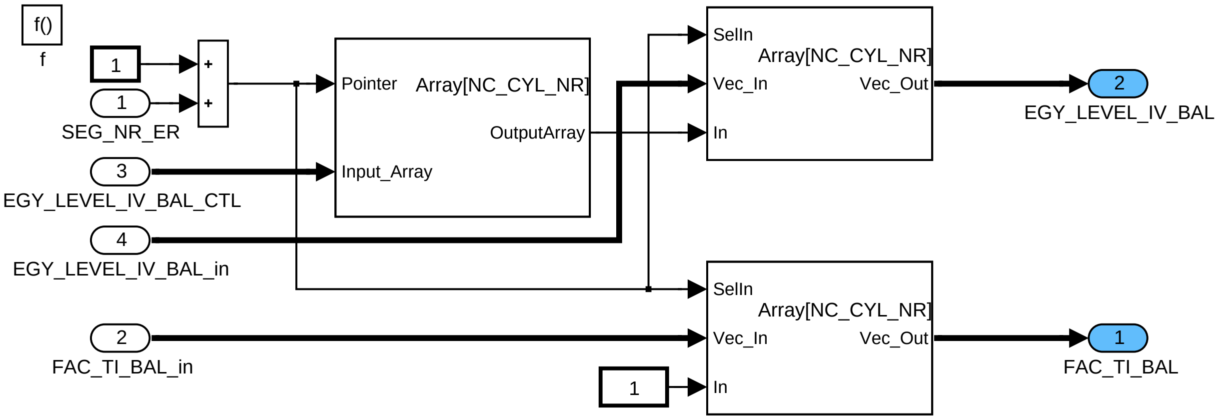


Figure 76 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_ELSE/ CASE_THEN3

Calculation of overall correction factors for cylinder balancing at engine run OPERATE_SEG CLC_TI_COR_FAC CASE_ELSE CASE_THEN4

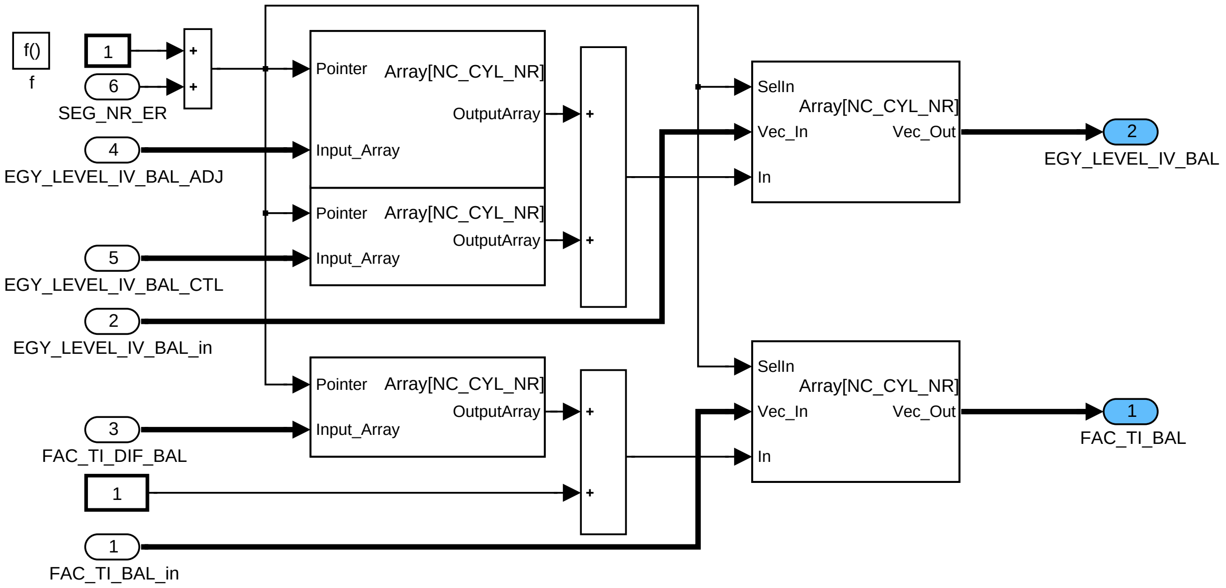


Figure 77 CYBL_M30704P01/ OPM/ OPERATE_SEG/ CLC_COR_FAC_ERU/ OPERATE_SEG/ CLC_TI_COR_FAC/ CASE_ELSE/ CASE_THEN4

6.3.1.3 SUBFUNCTION: SIG_MNG

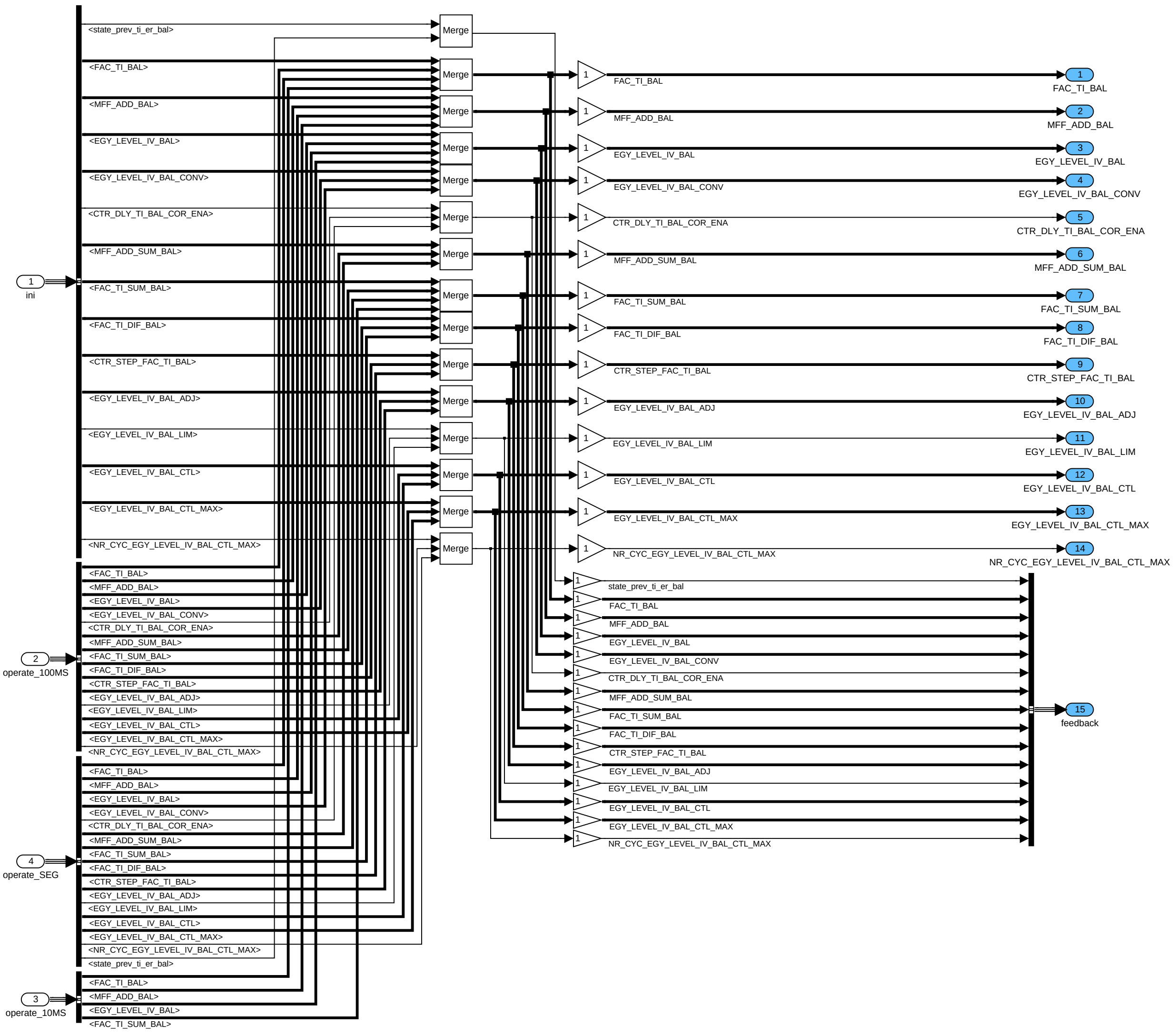


Figure 78 CYBL_M30704P01/ SIG_MNG

Chapter	Baseline		Include File
Cylinder balancing	4DC3940S		30704P01.00M
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 927 of 9643

6.4 Cylinder balancing via ER (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MFF_ADD_BAL_EXT[NC_CYL_NR]	O/V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
External cylinder balancing additive MFF correction value					
TCO_ST_BAL	O/V	0...FEH	-48...142.5	0.75	°C
Coolant temperature at start for cylinder balancing					
T_AST_BAL	O/V	0...FFFFH	0...6.5535E+3	0.1	s
Time after exit start for cylinder balancing					
LV_CYL_BAL_HOM_REQ_EXT	O/V	0...1H	0...1	1	-
External homogenous combustion mode request for cylinder balancing					
DIST_IV_DIF	-	0...FFFFH	0...5.2428E+5	8	km
Difference between actual distance and distance at last injector change					

Input data:

MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR]	LV_ST_END	NC_CYL_NR	DIST_IV_CHG[NC_CYL_NR]
DIST_KWP	TCO_ST	T_AST	LV_CYL_BAL_ER_CDN_BAS
LV_LAM_ORNG_LAM_AD_REQ[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_REQ_CUS	LF_ERR_PLAUS_IV_MFF_CAL	NC_CBK_EX_NR

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR]	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Default value for external cylinder balancing absolute MFF correction value					
C_MFF_NOM_IV_EXT_ADJ	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Default value for external cylinder balancing nominal MFF correction value					
LC_MFF_ADD_IV_EXT_ADJ_ENA	1	0...1H	0...1	1	-
Logical constant to enable external adjustment of cylinder balancing additive MFF correction					
IP_MFF_ADD_IV_AGI_DIST	16	0...FFFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
LDP_DIST_IV_DIF_IP_MFF_ADD_AGI	16	0...FFFFH	0...5.2428E+5	8	km
Additive MFF correction value depending on driving distance for compensation of aging effects					

6.4.1 General information

This application incidence module is used as interface module between the project specific key word protocol (KWP) and the generic cylinder balancing functionality. The KWP provides for every piezo injector and absolute MFF value, which is determined at end of injector production line and imported to the ECU at vehicle production.

To take the aging of each injector into consideration, the distance since the last injector change is calculated. The determined distance is basic for an additive MFF offset value out of an interpolation table.

It is possible to adjust default values provided by a constant array instead of the input values from the key word protocol (KWP). For this purpose a separate switch (logical constant) is existing to manage the absolute MFF setpoint values. In serial production, always the KWP values have to be used.

Application Condition

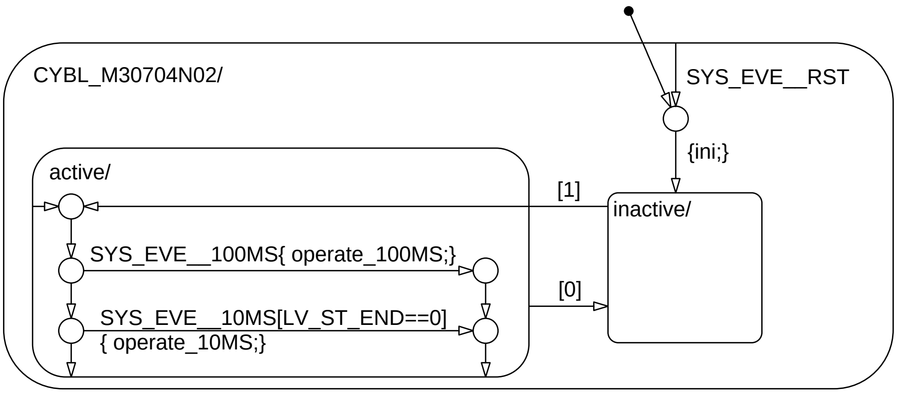


Figure 79 CYBL_M30704N01/ APP_CDN/ Chart

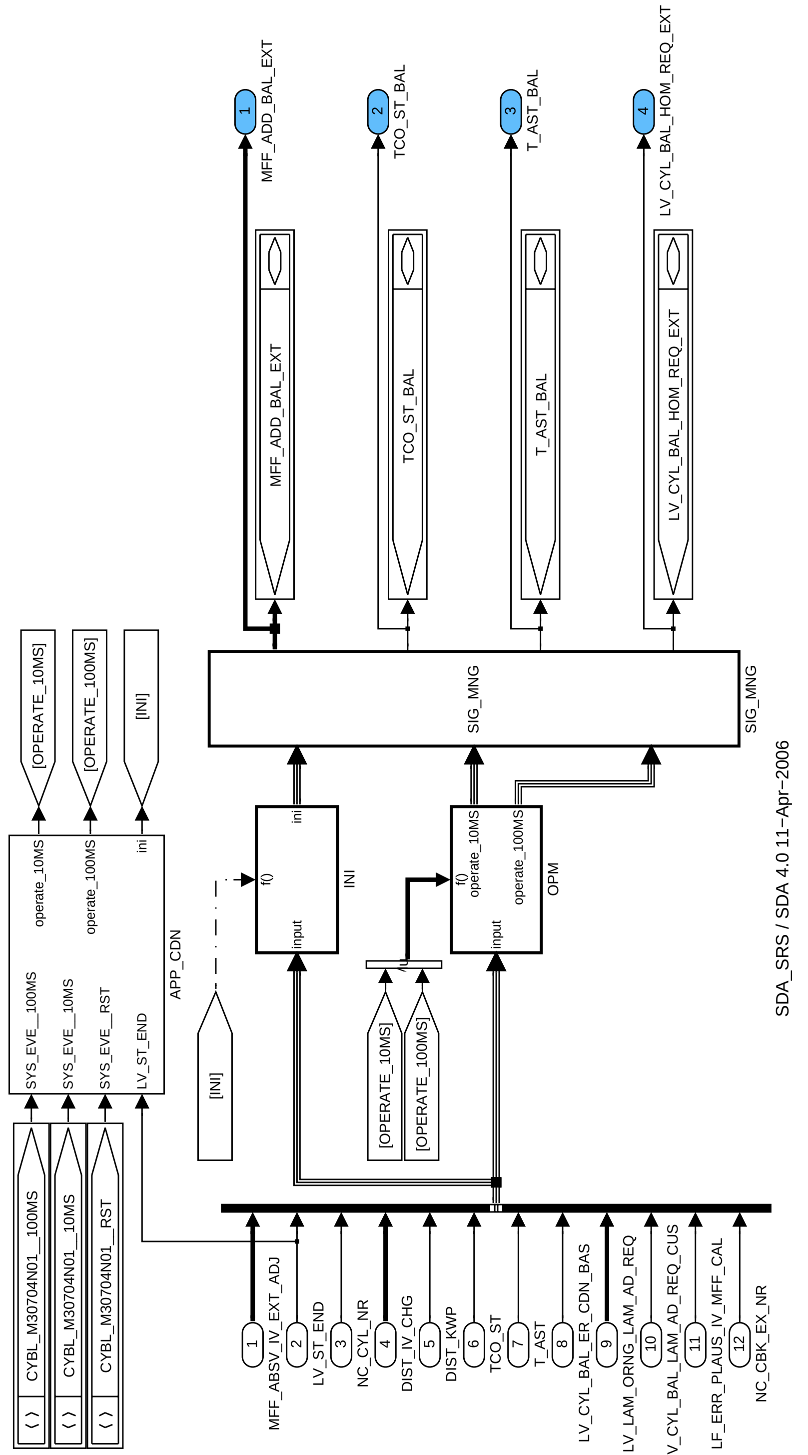


Figure 80 CYBL_M30704N01

6.4.1.1 Calculation of variables at reset

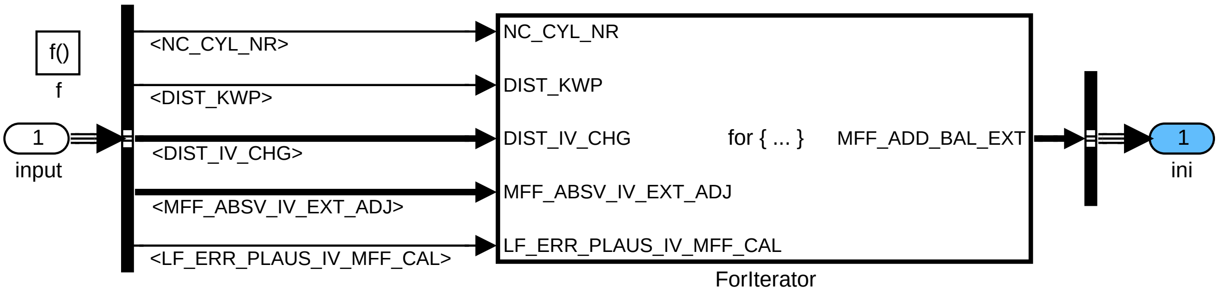


Figure 81 CYBL_M30704N01/ INI

Calculation of variables at reset FORITERATOR

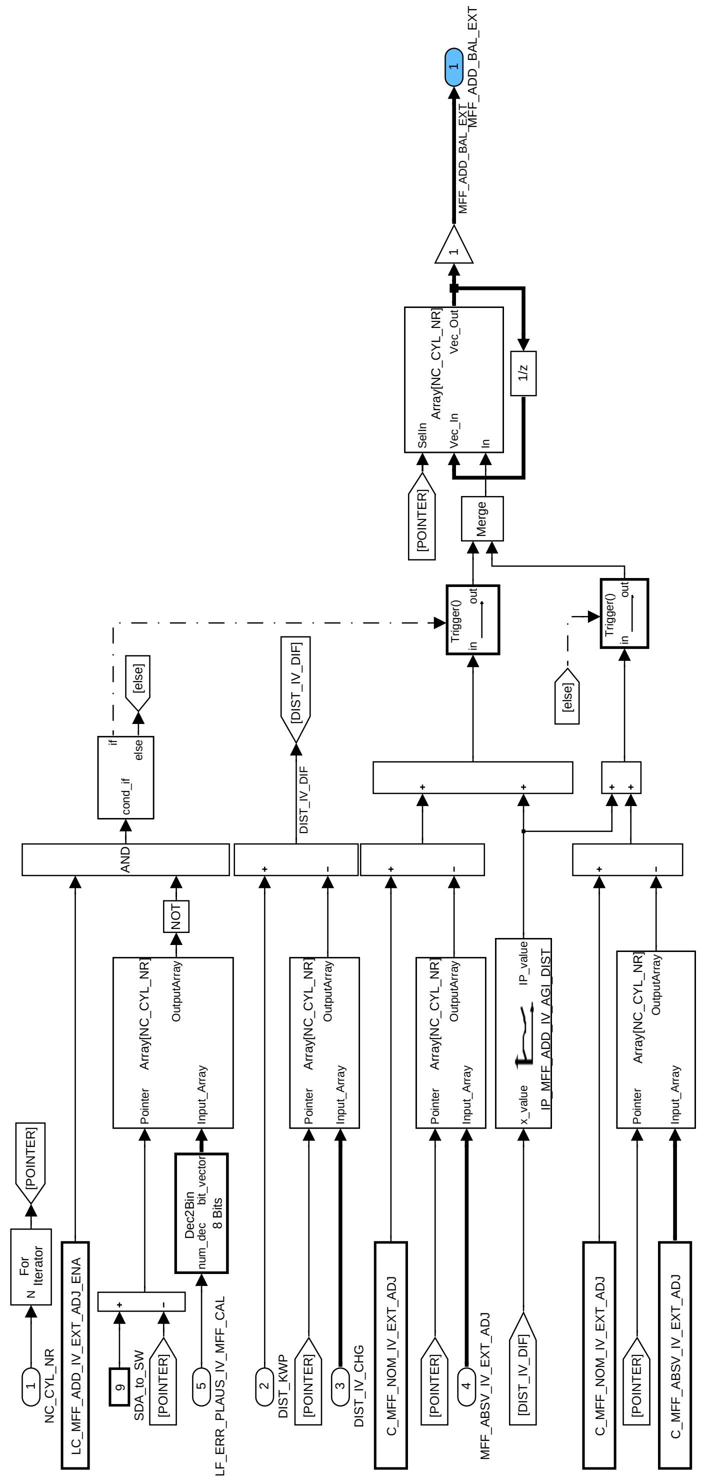


Figure 82 CYBL_M30704N01/ INI/ ForIterator

6.4.1.2 Calculation of 100ms and 10ms tasks

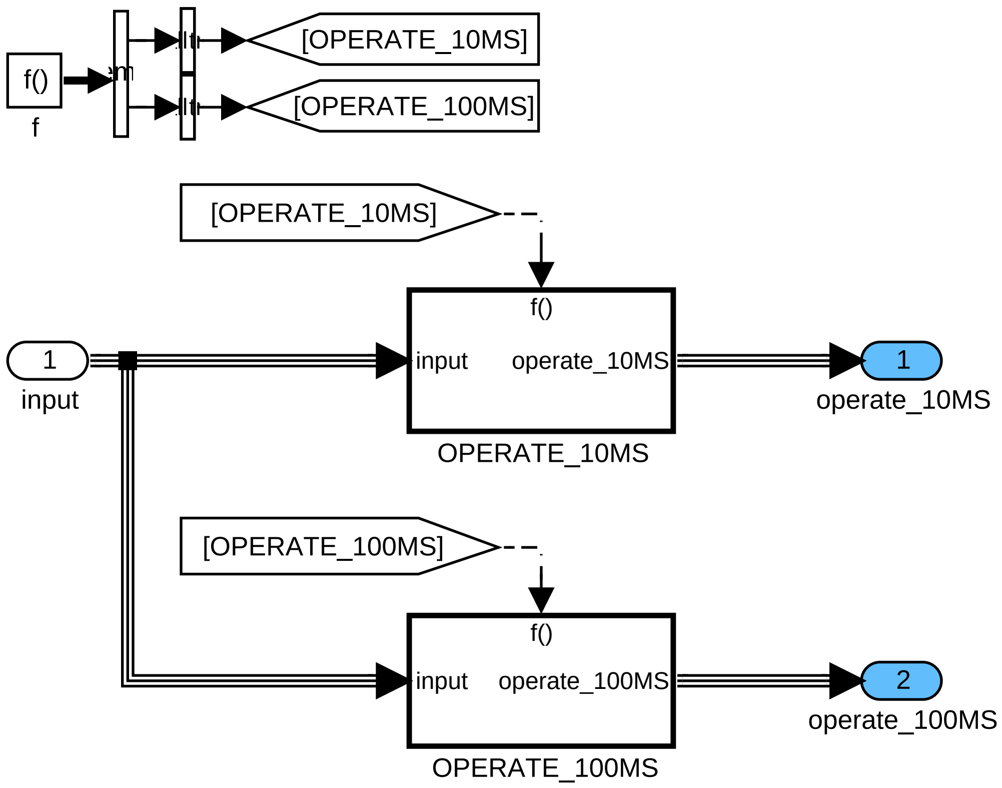


Figure 83 CYBL_M30704N01/ OPM

<u>External aging and coding correction values for cylinder balancing</u>

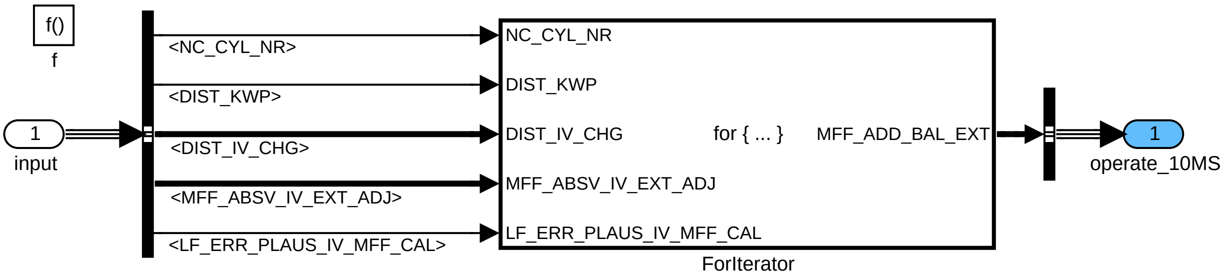


Figure 84 CYBL_M30704N01/ OPM/ OPERATE_10MS

<u>Calculation of external additive MFF offset values</u>

//calculation of additive MFF offset values with KWP inputs (case then)

//adjustment of default additive MFF offset values from constant array (case then)

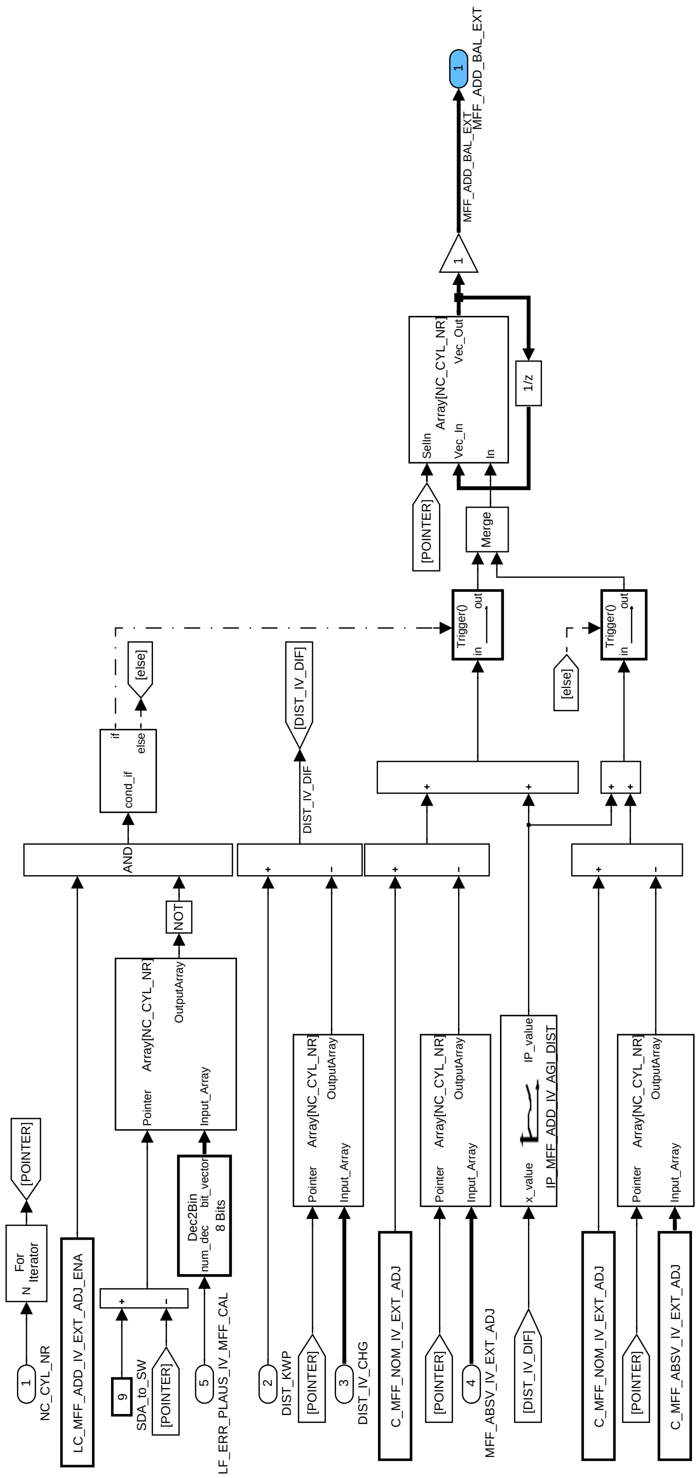


Figure 85 CYBL_M30704N01/ OPM/ OPERATE_10MS/ ForIterator

Adjustment of interface parameter for cylinder balancing

Goal of this chapter is to adjust interface parameter from other aggregates to cylinder balancing issues. Due to this modification it is possible to use one cylinder balancing version for all projects.

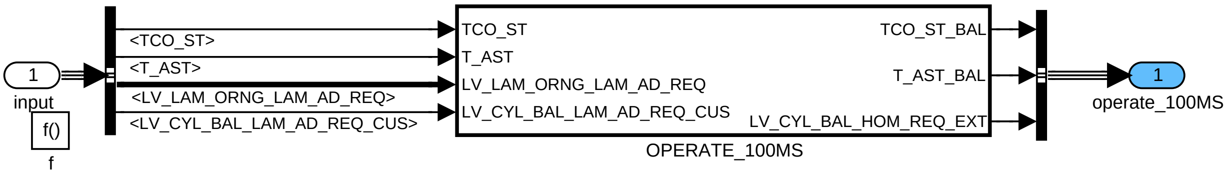


Figure 86 CYBL_M30704N01/ OPM/ OPERATE_100MS

Adjustment of interface parameter for cylinder balancing OPERATE_100MS

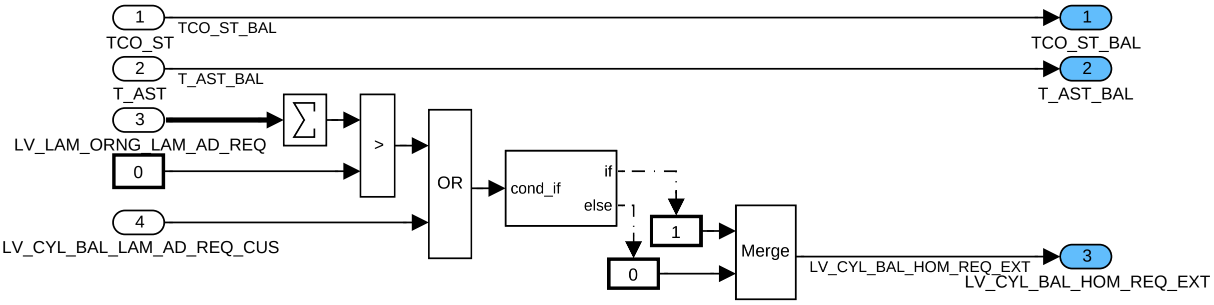


Figure 87 CYBL_M30704N01/ OPM/ OPERATE_100MS/ OPERATE_100MS

6.4.1.3 SUBFUNCTION: SIG_MNG

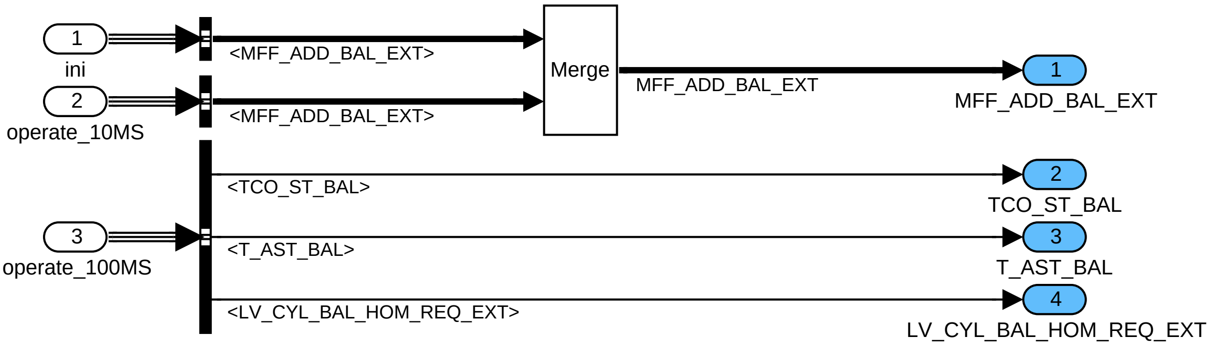


Figure 88 CYBL_M30704N01/ SIG_MNG

6.5 Cylinder balancing via additive MFF intervention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MFF_ADD_AD_ER_BAL[NC_CYL_NR]	O/V/S	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Additive adaptation value for cylinder balancing via MFF intervention					
MFF_ADD_ER_BAL[NC_CYL_NR]	O/V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Additive correction value for cylinder balancing via MFF intervention					
LV_MFF_ADD_ER_BAL_OBD_MAX_POS[NC_CYL_NR]	O/V	0...1H	0...1	1	-
Flag for maximum OBD limit reached at additive part of engine roughness based cylinder balancing					
LV_MFF_ADD_ER_BAL_OBD_MAX_NEG[NC_CYL_NR]	O/V	0...1H	0...1	1	-
Flag for minimum OBD limit reached at additive part of engine roughness based cylinder balancing					
ER_STND_MFF_ADD_ER_BAL	V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Controller input value at additive path of cylinder balancing via MFF intervention					
LV_ER_STND_MFF_ADD_ER_BAL	V	0...1H	0...1	1	-
Flag for controller input at additive path of cylinder balancing via MFF intervention					
MFF_ADD_ER_BAL_CTL	V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Controller output value at additive path of cylinder balancing via MFF intervention					
MFF_ADD_ER_BAL_COR[NC_CYL_NR]	V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Corrected additive correction value for cylinder balancing via MFF intervention					
NR_CYC_MFF_ADD_ER_BAL_CTL_MAX	V	0...FFH	0...255	1	-
Corresponding segment number of the controller that has reached the maximum limit (additive MFF path)					
MFF_ADD_ER_BAL_LIM[NC_CYL_NR]	V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Limited controller output value at additive path of cylinder balancing via MFF intervention					
MFF_ADD_ER_BAL_CTL_I[NC_CYL_NR]	V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
I-part of the controller output value at additive path of cylinder balancing via MFF intervention					
MFF_ADD_ER_BAL_CTL_MAX	V	0...FFFFH	0...6.5535E+4	1	-
Controller output value maximum limit reached at additive path of cylinder balancing via MFF intervention					
MFF_ADD_ER_BAL_CTL_P	V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
P-part of the controller output value at additive path of cylinder balancing via MFF intervention					

Input data:

ER_STND_MMV_DIF_BAL[NC_CYL_NR]	LV_MFF_ADD_ER_BAL_ENA	SEG_NR_ER	NC_CYL_NR
LV_TI_ER_BAL_ACT	LV_MFF_ADD_AD_ER_BAL_ENA	LV_MFF_ADD_AD_ER_BAL_EXT_ADJ	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CYCNR_MFF_ADD_ER_BAL	1	1...FFH	1...255	1	-
Number of cycles for calculation of controller output values for cylinder balancing via MFF intervention					
C_ER_STND_THD_NEG_MFF_ADD_BAL	1	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Negative threshold for ER input value at additive path of cylinder balancing via MFF intervention					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ER_STND_THD_POS_MFF_ADD_BAL	1	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Positive threshold for ER input value at additive path of cylinder balancing via MFF intervention					
C_MFF_ADD_ER_BAL_CTL_MAX_NEG	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Negative maximum limit for controller output at additive path of cylinder balancing via MFF intervention					
C_MFF_ADD_ER_BAL_CTL_MAX_POS	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Positive maximum limit for controller output at additive path of cylinder balancing via MFF intervention					
C_MFF_ADD_ER_BAL_OBD_MAX_NEG	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Negative maximum OBD limit for additive correction value of cylinder balancing via MFF intervention					
C_MFF_ADD_ER_BAL_OBD_MAX_POS	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Positive maximum OBD limit for additive correction value of cylinder balancing via MFF intervention					
C_MFF_ADD_ER_BAL_RST	1	0H 1H 2H	INI_ZERO INI_CTL INI_AD	1	-
Reset of additive correction- and adaptation values for cylinder balancing via MFF intervention					
LC_MFF_ADD_AD_ER_BAL_RST_MAN	1	0...1H	0...1	1	-
Manual reset of additive adaptation values for cylinder balancing via MFF intervention					
LC_MFF_ADD_ER_BAL_COR	1	0...1H	0...1	1	-
Manual switch for overall correction of controller output values for cylinder balancing via MFF intervention					
LC_MFF_ADD_ER_BAL_RST_MAN	1	0...1H	0...1	1	-
Manual reset of additive correction values for cylinder balancing via MFF intervention					
IP_MFF_ADD_I_ER_BAL	8	0...FFFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
LDP_ER_STND_MFF_ADD_IP_I_ER_BAL	8	0...FFFFH	-325.78...325.77	0.0099313	1/s²
Controller I-part value for additive path of cylinder balancing via MFF intervention					
IP_MFF_ADD_P_ER_BAL	8	0...FFFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
LDP_ER_STND_MFF_ADD_IP_P_ER_BAL	8	0...FFFFH	-325.78...325.77	0.0099313	1/s²
Controller P-part value for additive path of cylinder balancing via MFF intervention					

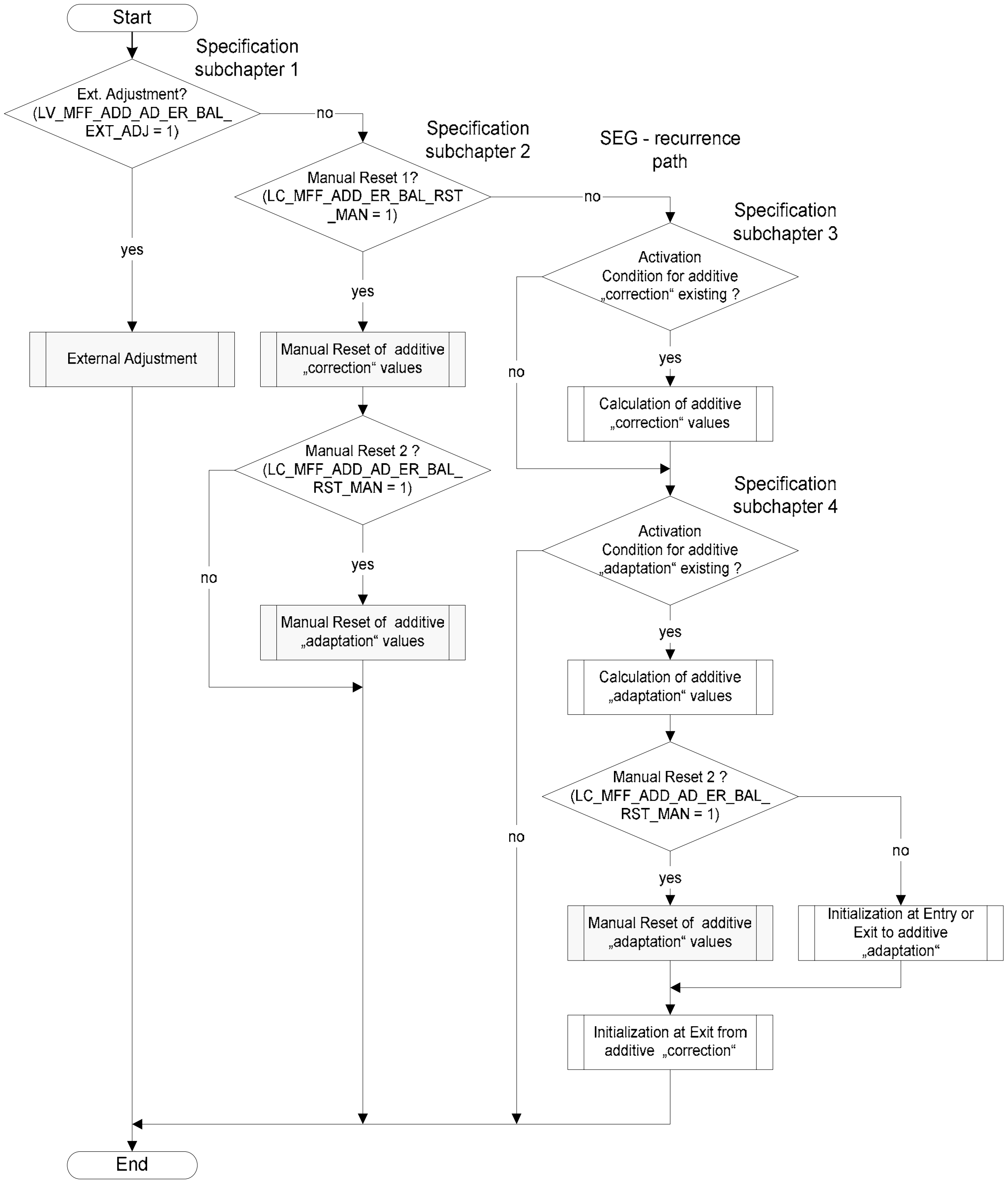
6.5.1 General information

The calculation of the additive correction- or adaptation values is performed, depending on the settings of the corresponding logical activation variable. LV_MFF_ADD_ER_BAL_ENA is used to enable the correction path, while LV_MFF_ADD_AD_ER_BAL_ENA enables the adaptation path. As soon as the corresponding logical value is set to “1” in combination with additional activation conditions, the calculation is started.

The calculated additive adaptation value MFF_ADD_AD_ER_BAL[NC_CYL_NR] for each cylinder is stored in the non-volatile memory to be available for initialization issues at next engine run.

With use of the manual reset it is possible to initialize the additive correction- or adaptation values with zero at any time. In this case, the flags which indicate the exceeding of a controller output limitation are set to zero as well (refer: “Initialization at Manual Reset”)

Signal flow diagram:


Start
Specification subchapter 1
Ext. Adjustment? (LV_MFF_ADD_AD_ER_BAL_EXT_ADJ = 1)
no
yes
External Adjustment
Specification subchapter 2
Manual Reset 1? (LC_MFF_ADD_ER_BAL_RST_MAN = 1)
no
yes
Manual Reset of additive „correction“ values
Manual Reset 2 ? (LC_MFF_ADD_AD_ER_BAL_RST_MAN = 1)
no
yes
Manual Reset of additive „adaptation“ values
SEG - recurrence path
Specification subchapter 3
Activation Condition for additive „correction“ existing ?
no
yes
Calculation of additive „correction“ values
Specification subchapter 4
Activation Condition for additive „adaptation“ existing ?
no
yes
Calculation of additive „adaptation“ values
Manual Reset 2 ? (LC_MFF_ADD_AD_ER_BAL_RST_MAN = 1)
no
yes
Manual Reset of additive „adaptation“ values
Initialization at Entry or Exit to additive „adaptation“
Initialization at Exit from additive „correction“
End

Application Condition

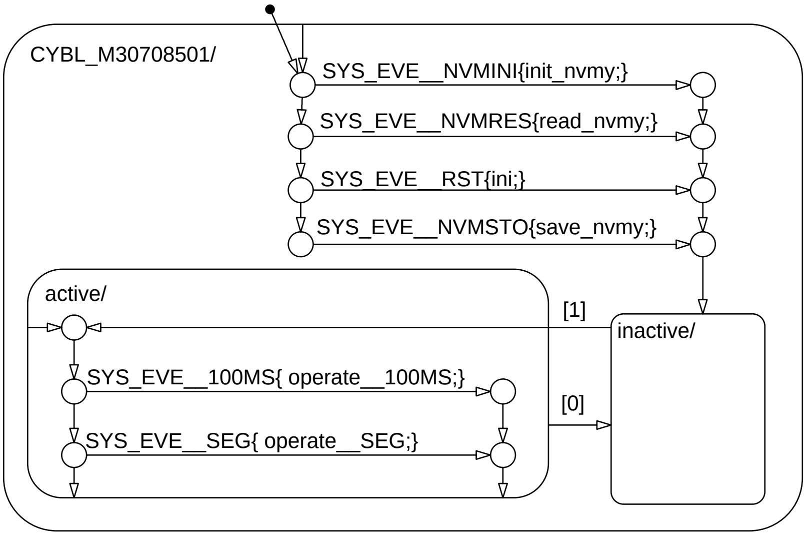


Figure 89 CYBL_M30708501/ APP_CDN/ Chart

Function Description

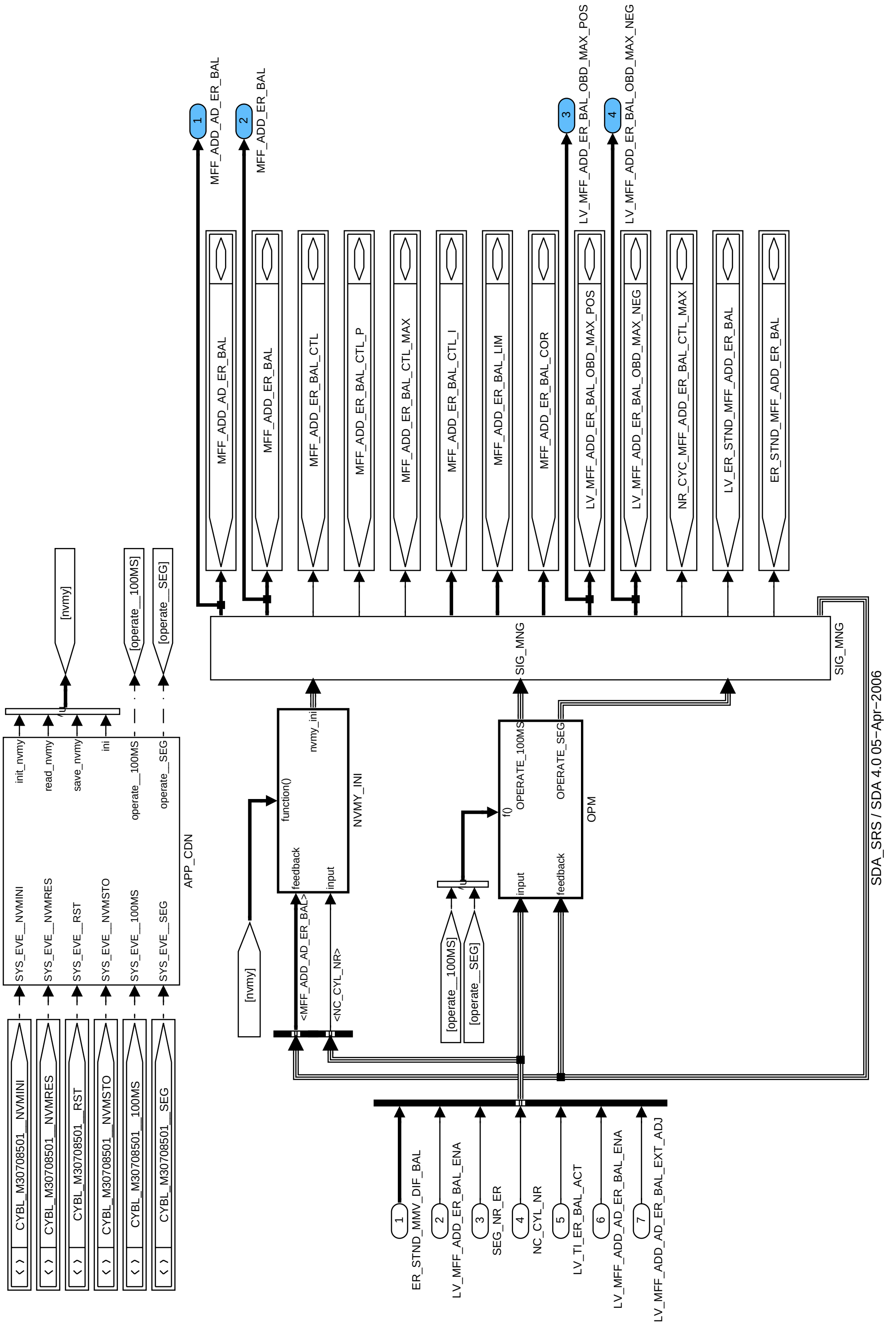


Figure 90 CYBL_M30708501

6.5.1.1 Calculation of non volatile memory tasks

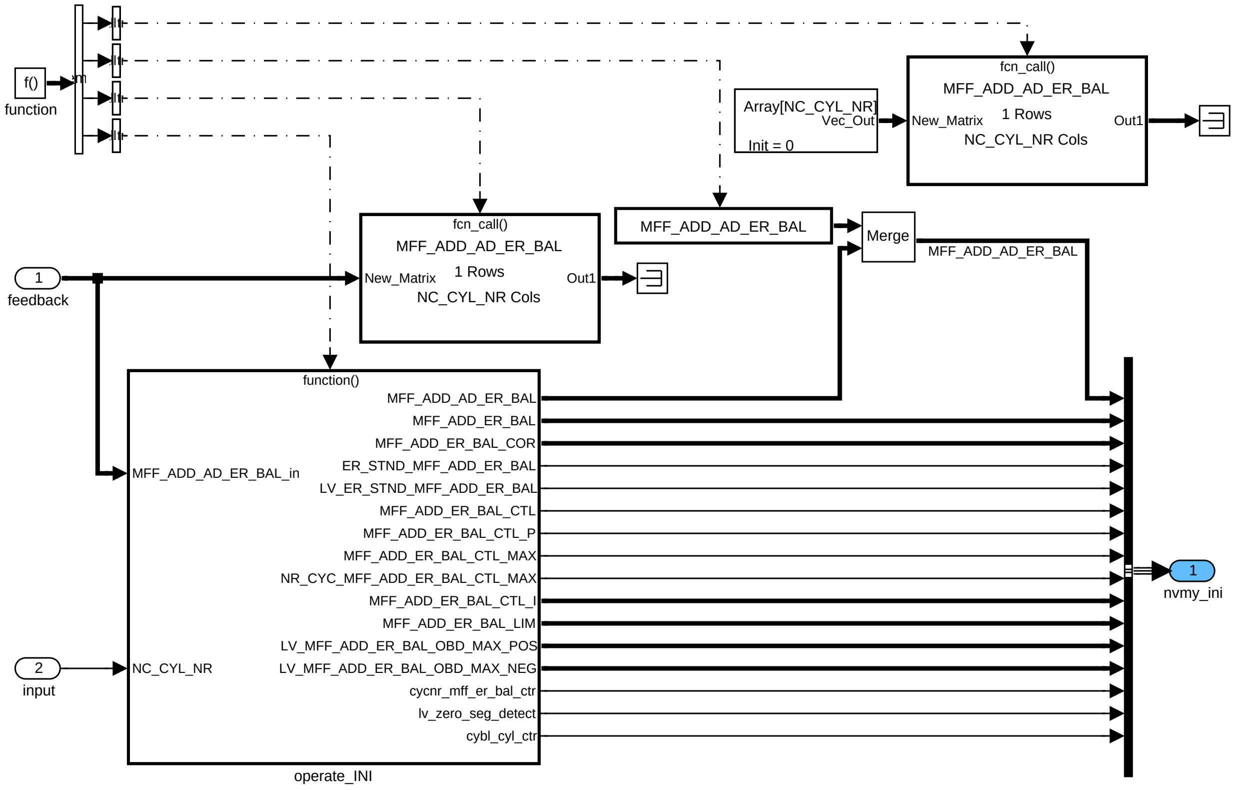


Figure 91 CYBL_M30708501/ NVMY_INI

Calculation of variables at reset task

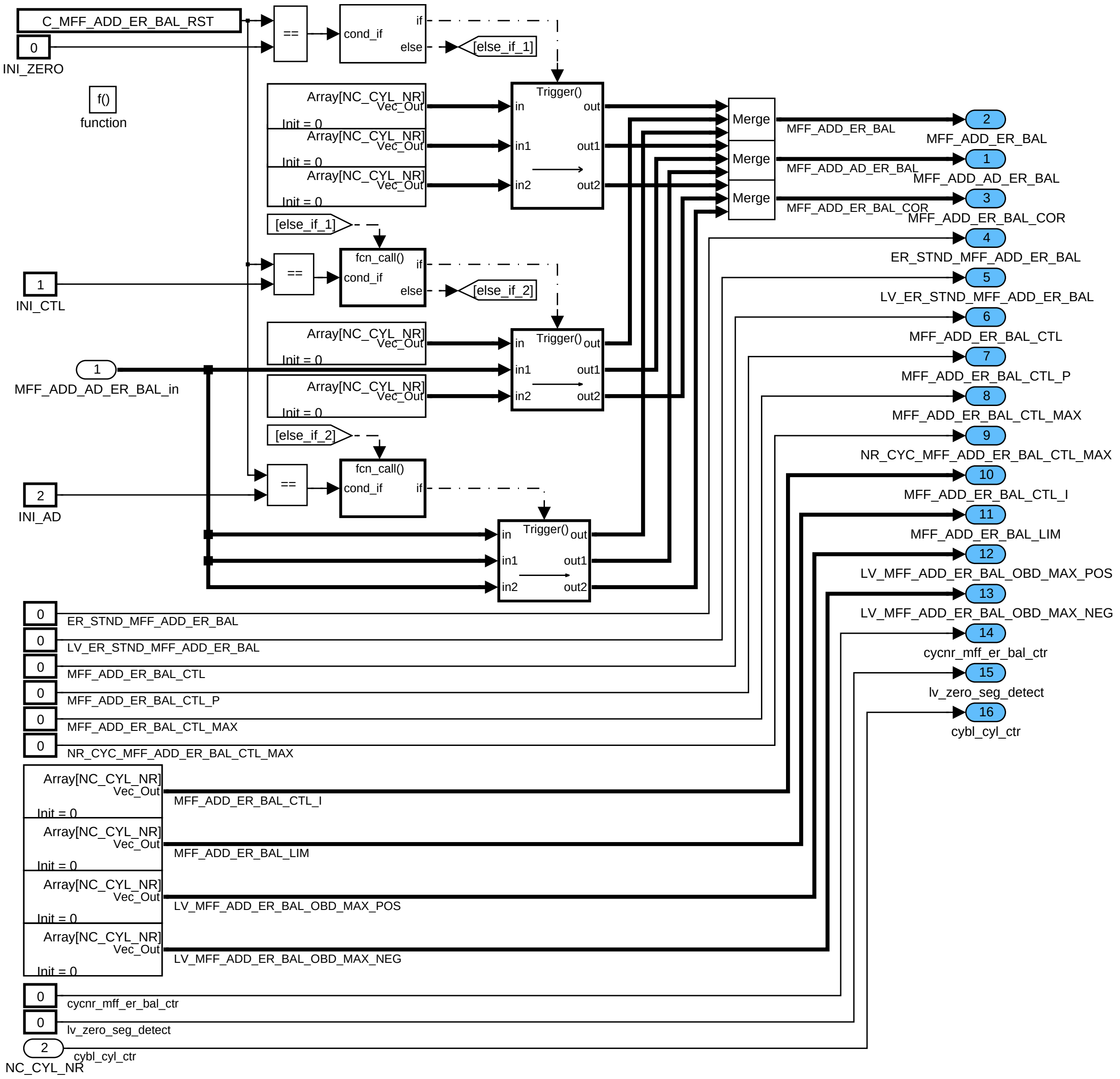


Figure 92 CYBL_M30708501/ NVMY_INI/ operate_INI

6.5.1.2 Calculation of 100ms and segment synchronous tasks

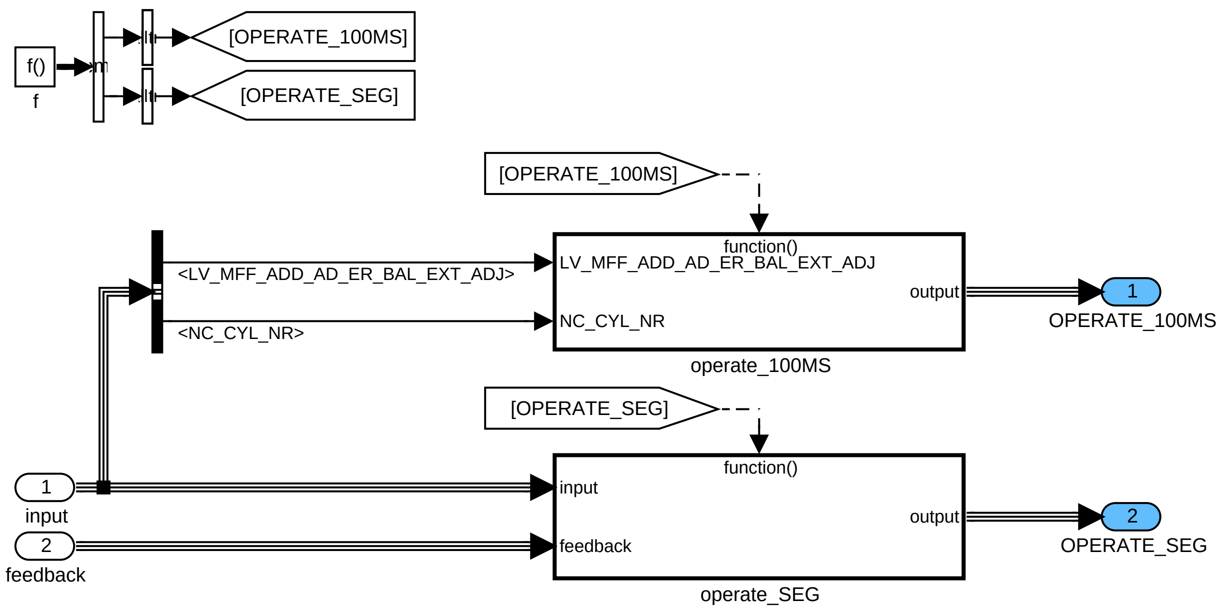


Figure 93 CYBL_M30708501/ OPM

<u>External adjustment of additive correction values</u>

In case of an external adjustment (request from key word protocol) the additive adaptation values for all cylinders are set to zero. While the external adjustment is activated, all following subchapters within the module "Cylinder balancing via additive MFF intervention" are deactivated and the corresponding "Initialization at Deactivation" process for this chapters is blocked during this period.

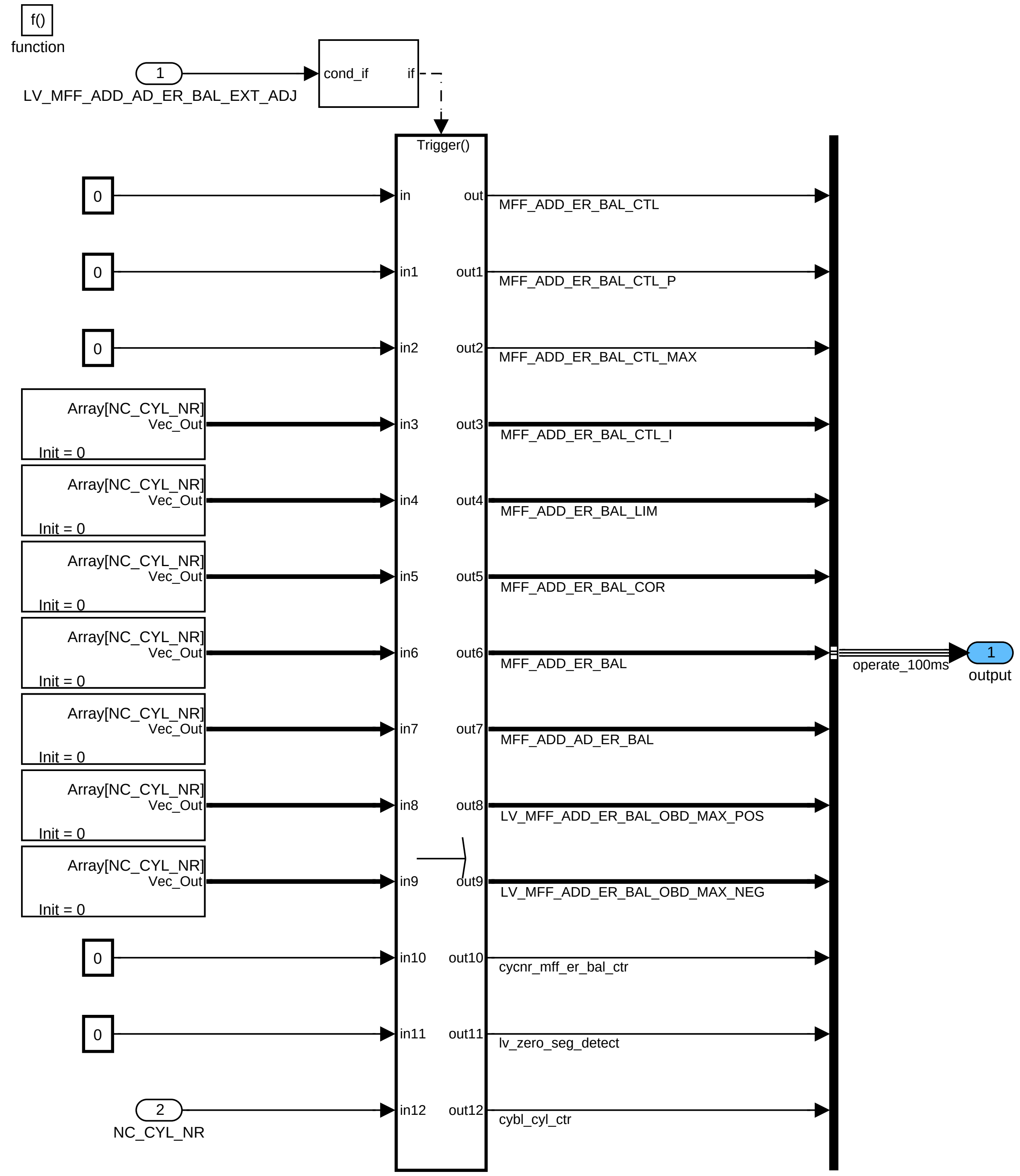


Figure 94 CYBL_M30708501/ OPM/ operate_100MS

Calculation of segment task

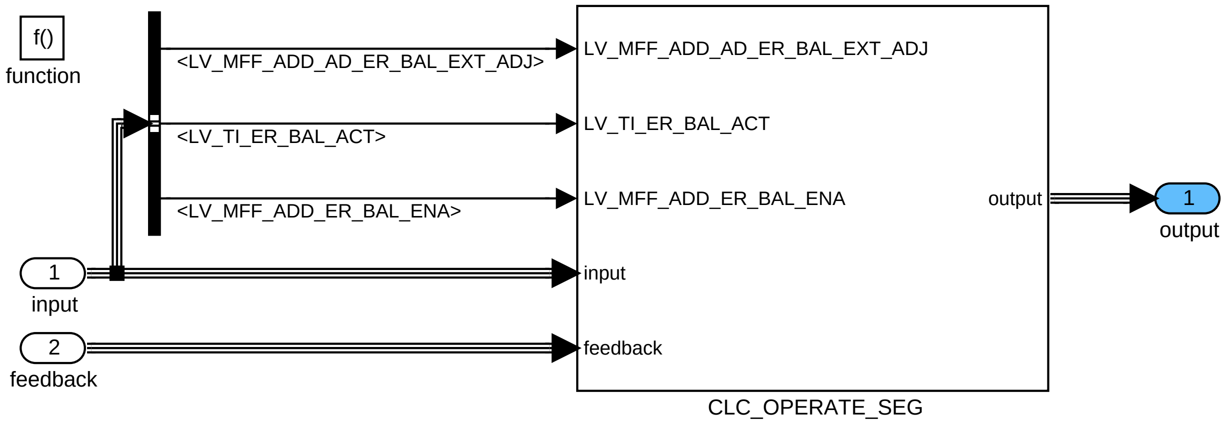


Figure 95 CYBL_M30708501/ OPM/ operate_SEG

Calculation of manual reset, injection time correction and injection time adaptation

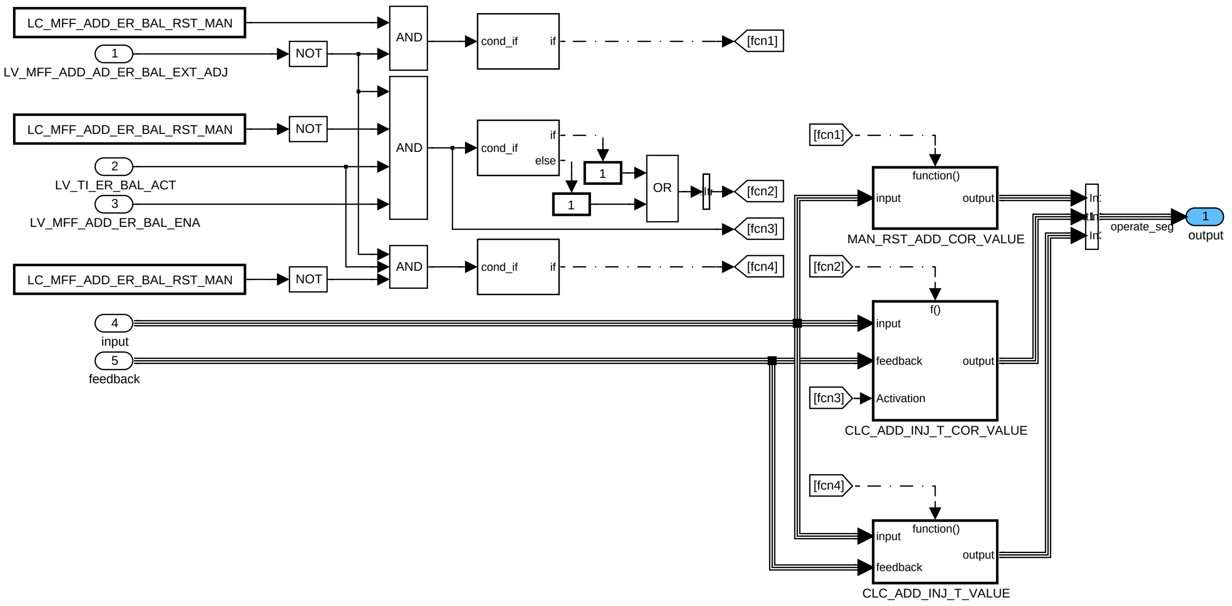


Figure 96 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG

Manual reset of additive correction values

In case of a manual reset it is possible to adjust the exported output data to a wanted value to make the function test and handling more easier. While the manual function reset is activated, all following subchapters within the module "Cylinder balancing via additive MFF intervention" are deactivated and the corresponding "Initialization at Deactivation" process for this chapters is blocked during this period.

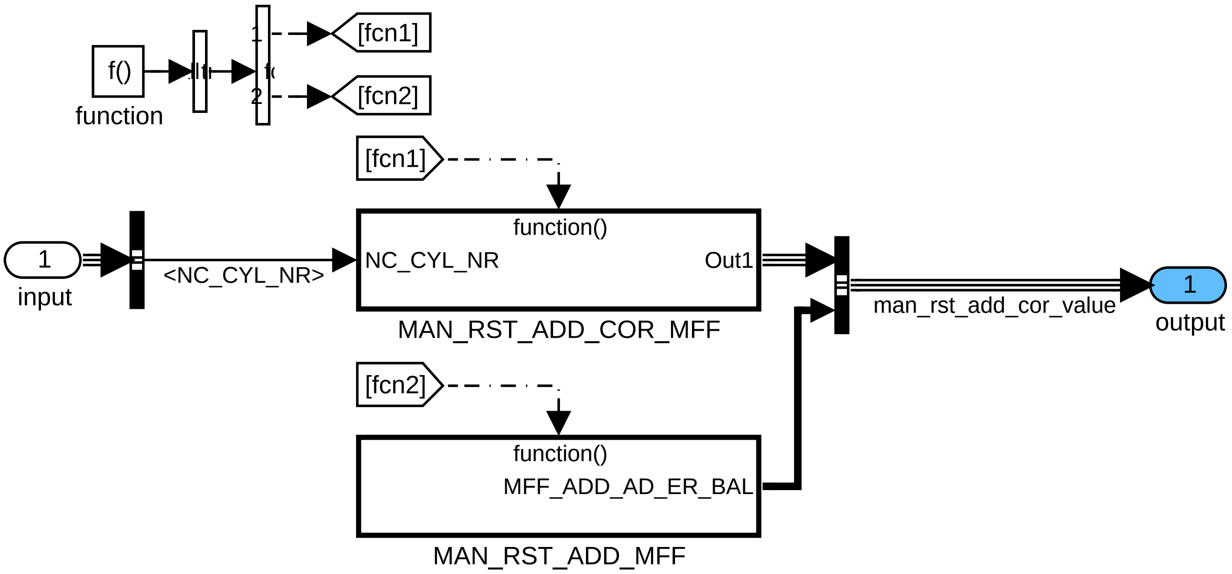


Figure 97 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ MAN_RST_ADD_COR_VALUE

Manual reset of additive correction values for cylinder balancing via MFF intervention

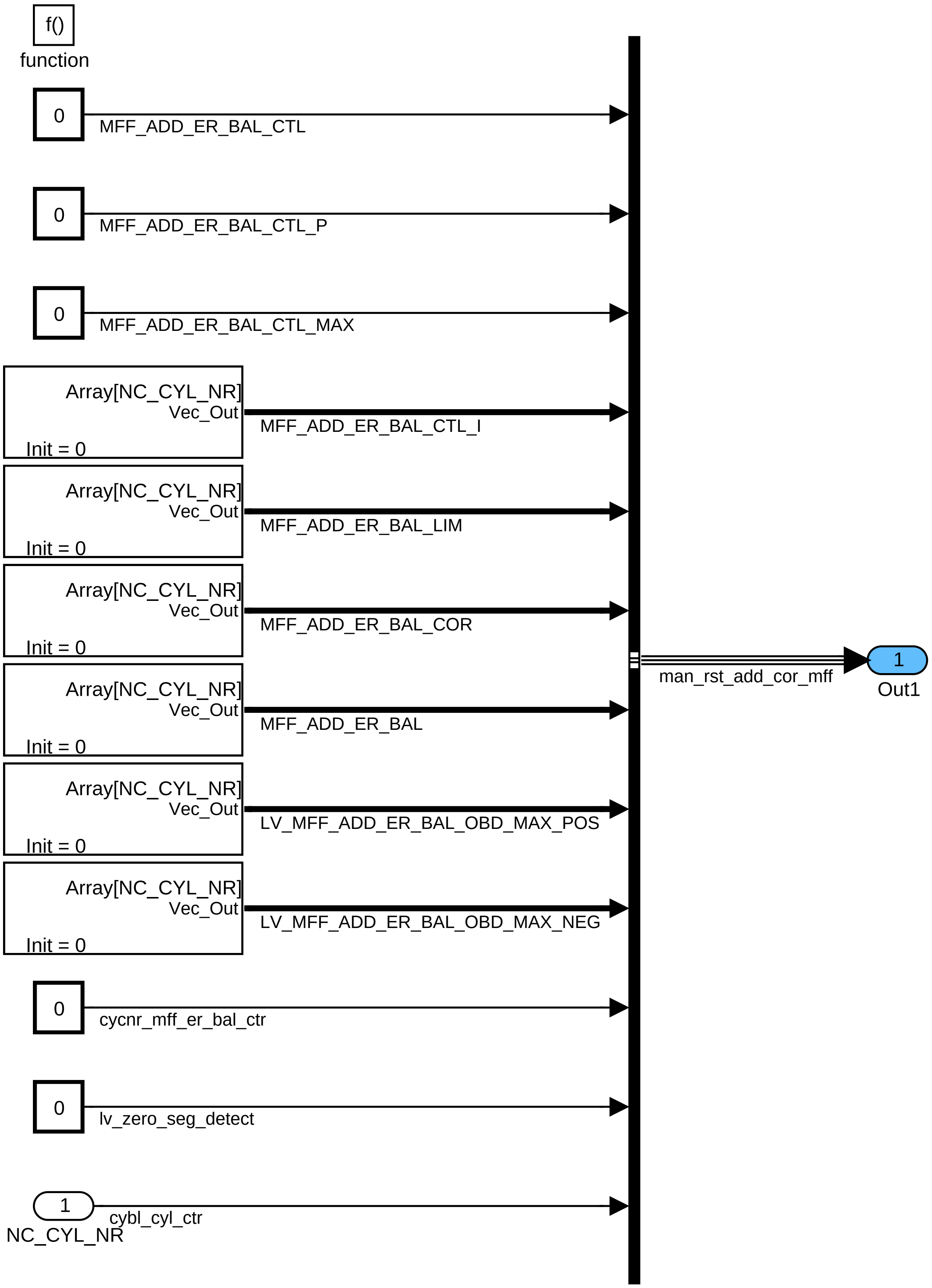


Figure 98 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ MAN_RST_ADD_COR_VALUE/ MAN_RST_ADD_COR_MFF

Manual reset of additive adaptation values for cylinder balancing via MFF intervention

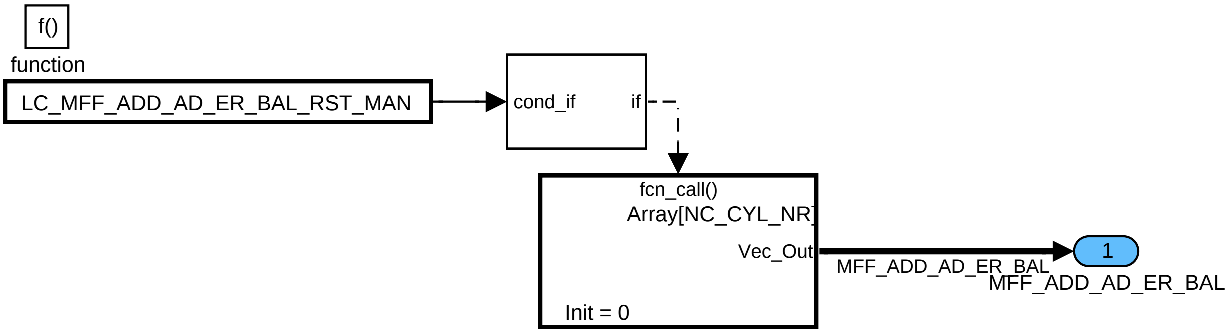


Figure 99 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ MAN_RST_ADD_COR_VALUE/ MAN_RST_ADD_MFF

Calculation of the additive injection time correction values

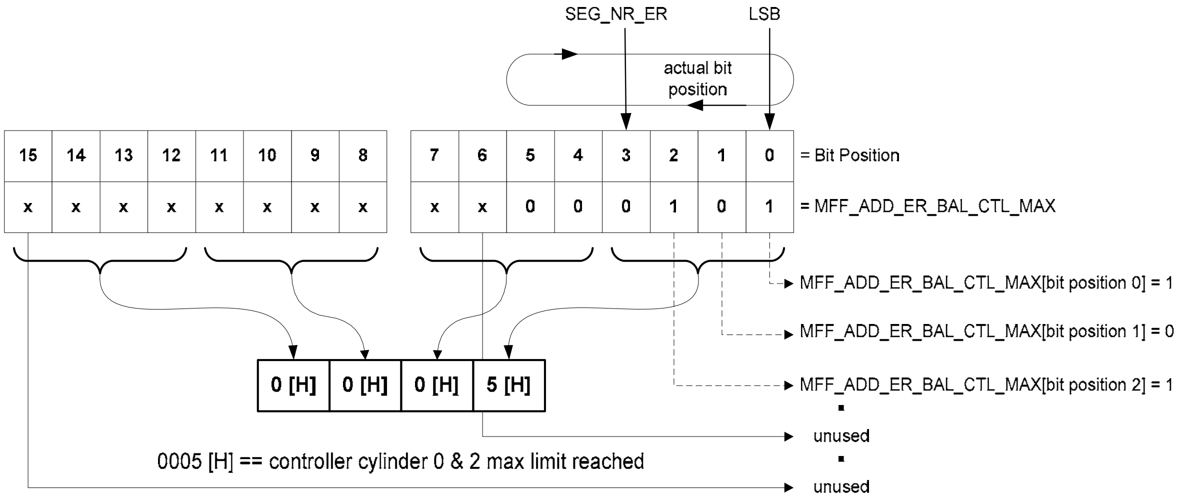
The calculation of the additive injection time correction values are only performed if the corresponding enable bit LV_MFF_ADD_ER_BAL_ENA is set to logical “1” in combination with additional activation conditions. In this case, a separate correction value for each cylinder is determined with use of a PI-controller.

As soon as a PI-controller of one cylinder is reaching its maximum, a corresponding value is set to “1” to indicate the related cylinder. To save memory resources, the information for all cylinders is located within one output data (WORD). Each BIT information within the WORD is used to indicate one cylinder. It is possible to handle up to 16 cylinders with this method.

For a description more in detail please have a look to the signal flow diagram below. As example, a six-cylinder engine with two PI-controllers, which have reached the limits is shown.

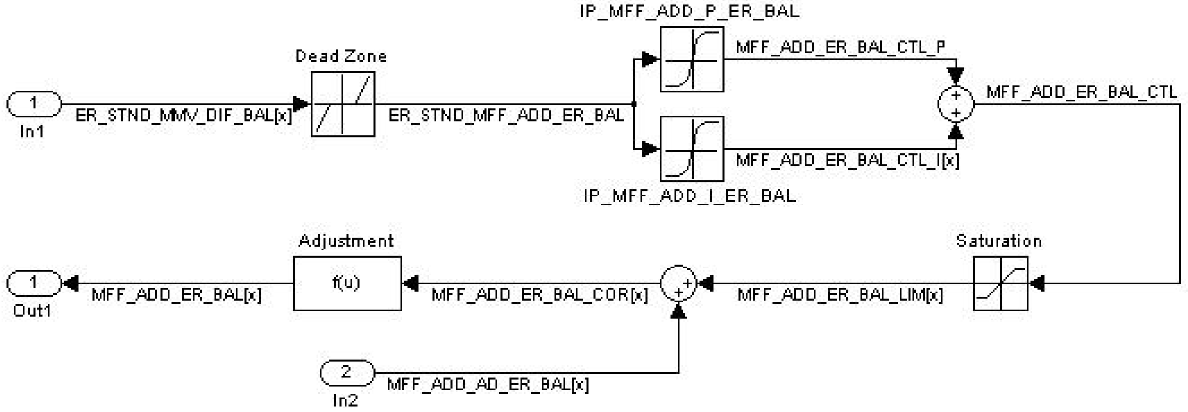
Signal flow diagram:

- Signalization of controller max. output value reached for additive path



Note: always the complete WORD (16 BIT) is displayed as visible value

- Calculation of the additive injection time correction / adaptation values in general



Note: [_x] is representing [SEG_NR_ER] in the signal flow diagram above

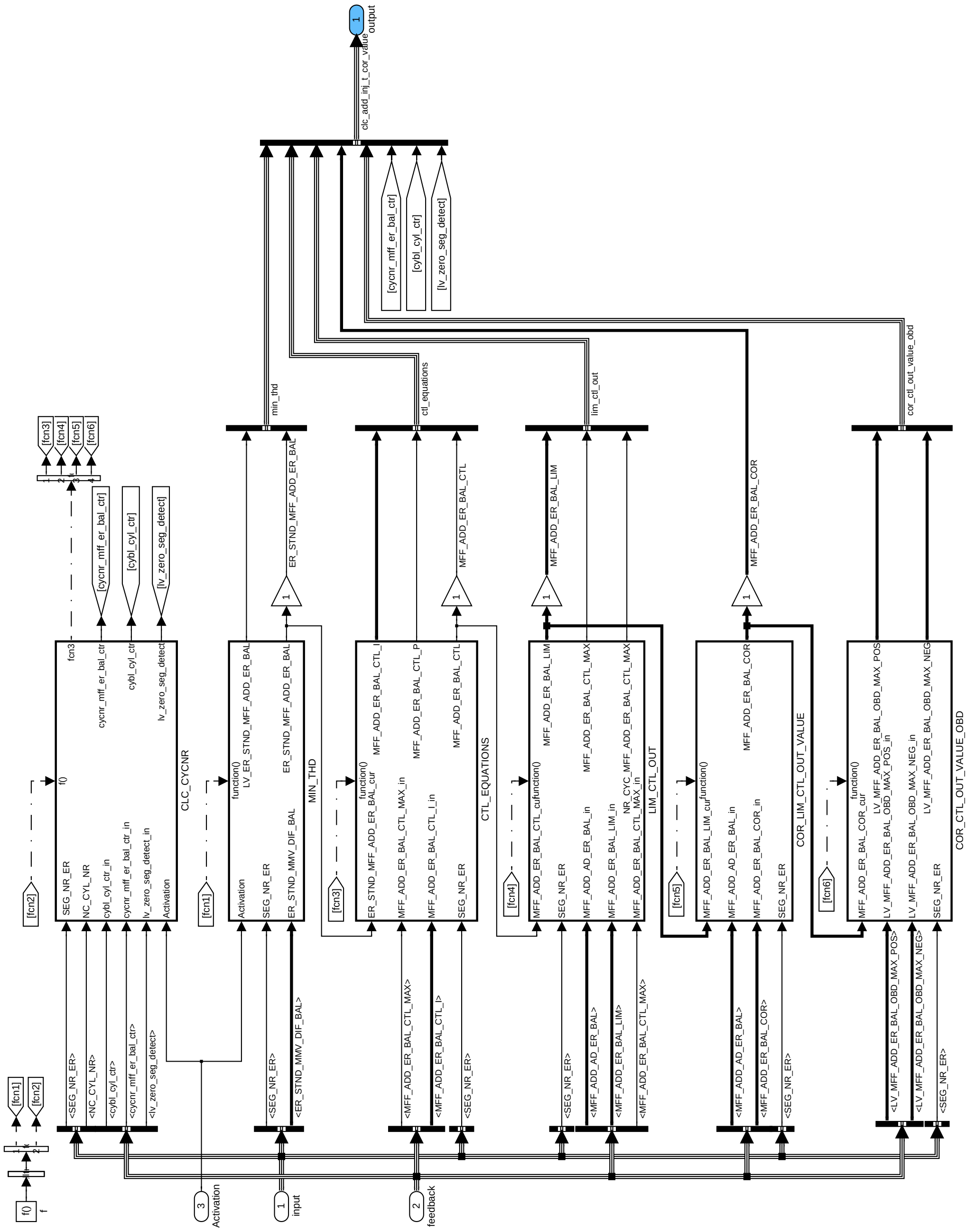


Figure 100 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE

Minimum threshold

If the defined cylinder balancing input values ER_STND_MMV_DIF_BAL[SEG_NR_ER] are outside a calibrateable range, the controller is calculated. Otherwise the controller inputs are set to zero.

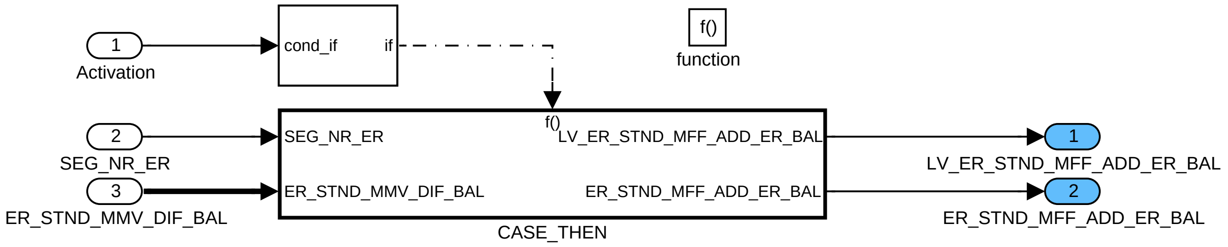


Figure 101 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ MIN_THD

Minimum threshold CASE_THEN

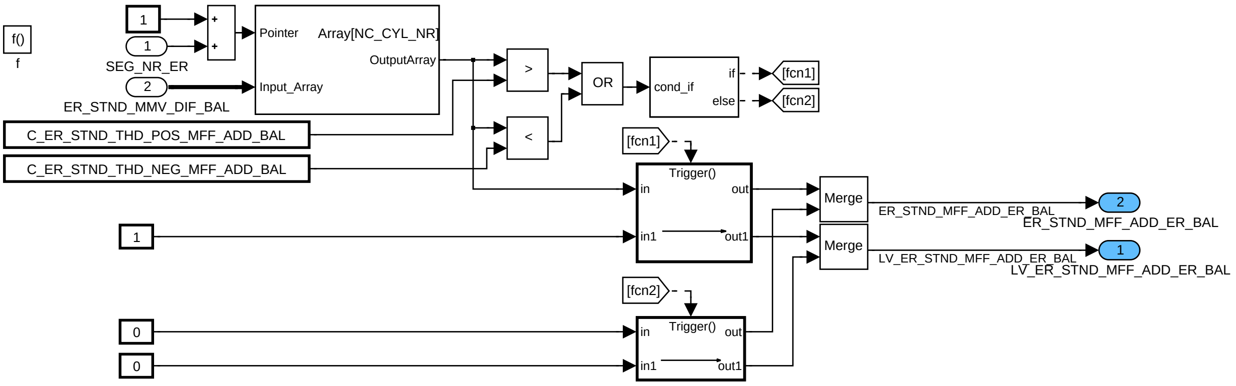


Figure 102 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ MIN_THD/ CASE_THEN

Calculation of segment tasks CLC_OPERATE_SEG CLC_ADD_INJ_T_COR_VALUE CLC_CYCNR

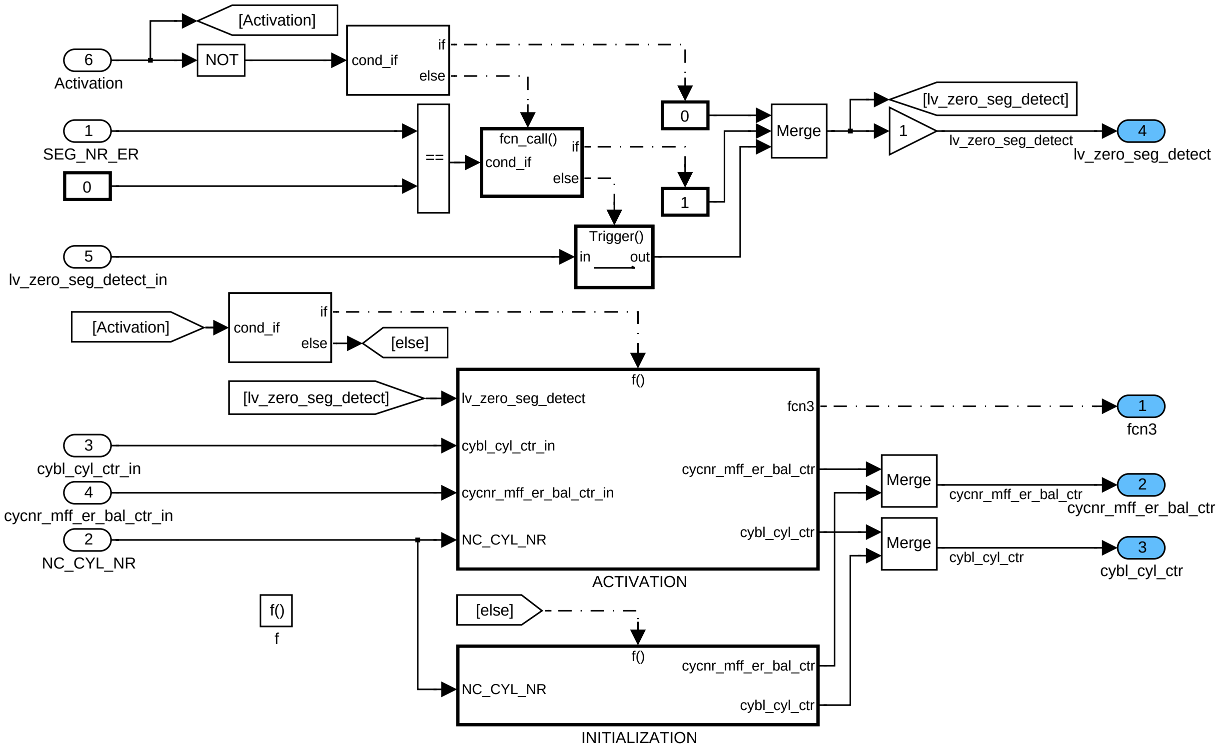


Figure 103 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ CLC_CYCNR

Calculation of segment tasks CLC_OPERATE_SEG CLC_ADD_INJ_T_COR_VALUE CLC_CYCNR ACTIVATION

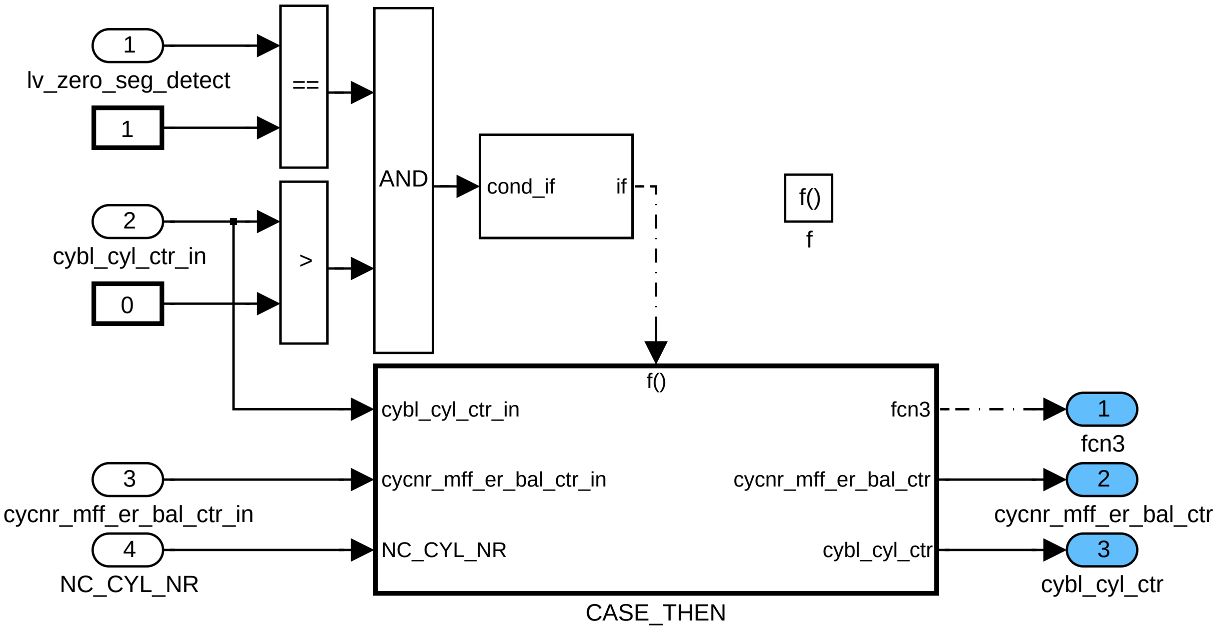


Figure 104 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ CLC_CYCNR/ ACTIVATION

Calculation of segment tasks CLC_OPERATE_SEG CLC_ADD_INJ_T_COR_VALUE CLC_CYCNR ACTIVATION CASE_THEN

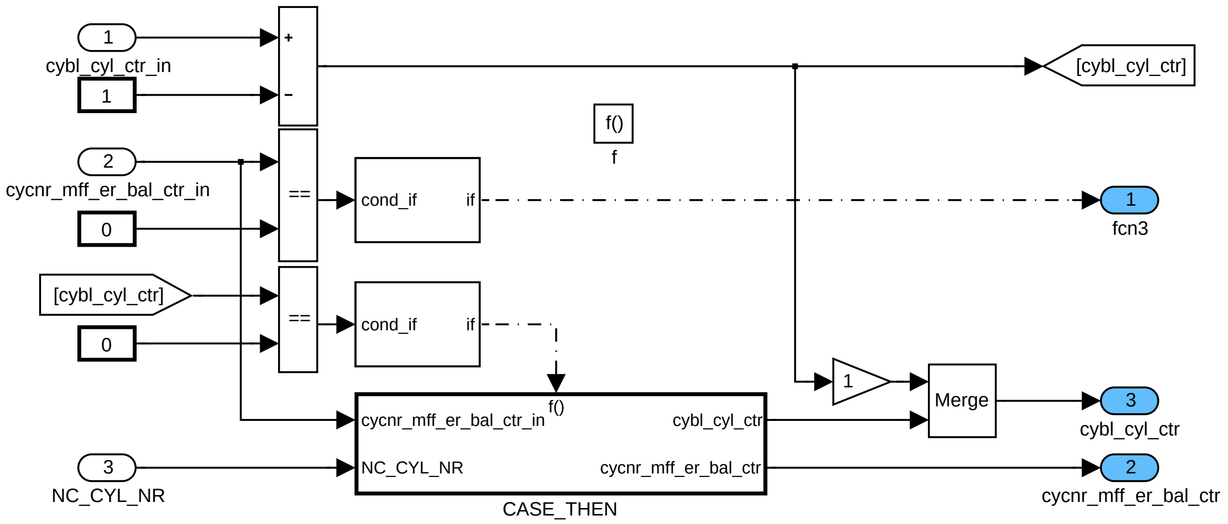


Figure 105 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ CLC_CYCNR/ ACTIVATION/ CASE_THEN

Calculation of segment tasks CLC_OPERATE_SEG CLC_ADD_INJ_T_COR_VALUE CLC_CYCNR ACTIVATION CASE_THEN CASE_THEN

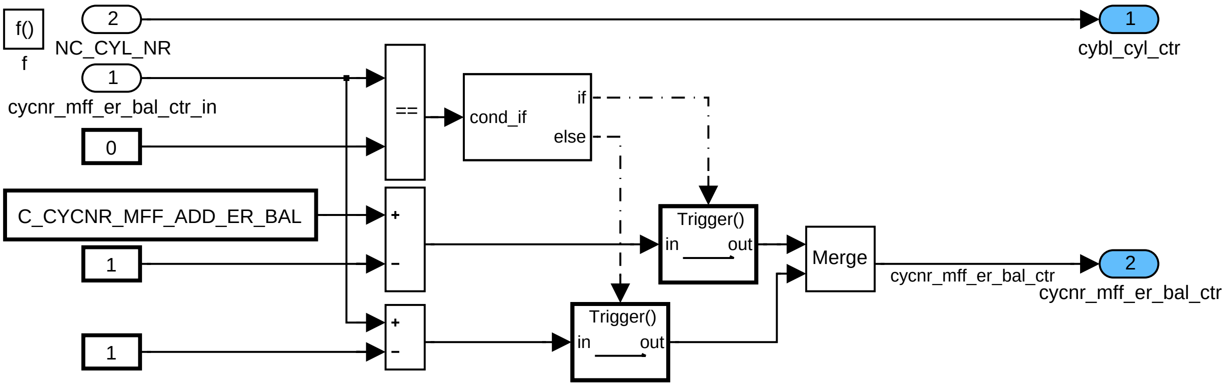


Figure 106 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ CLC_CYCNR/ ACTIVATION/ CASE_THEN/ CASE_THEN

Calculation of segment tasks CLC_OPERATE_SEG CLC_ADD_INJ_T_COR_VALUE CLC_CYCNR INITIALIZATION

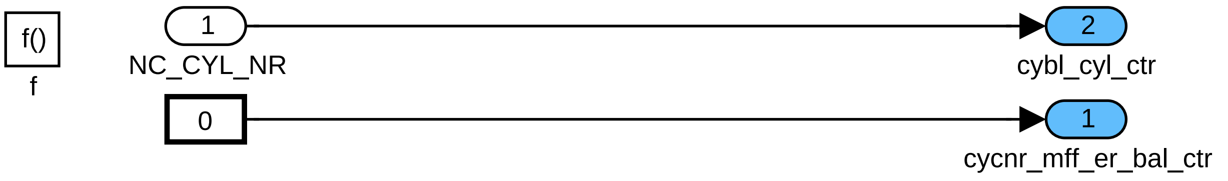


Figure 107 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ CLC_CYCNR/ INITIALIZATION

Controller Equataions

For the additive correction mode, a separate set of NC_CYL_NR controllers exists operating in parallel (one controller for each cylinder).

The computation of the output values is carried out every C_CYCNR_MFF_ADD_ER_BAL engine cycles (= calibrateable number of engine cycles).

Only in case of C_CYCNR_MFF_ADD_ER_BAL = 1, the calculation of the additive path is done without interruption. At every segment the corresponding cylinder specific correction value is built.

If the computation is not done at every engine cycle (C_CYCNR_MFF_ADD_ER_BAL > 1), the trigger of the controller output calculation is always related to SEG_NR_ER = 0.

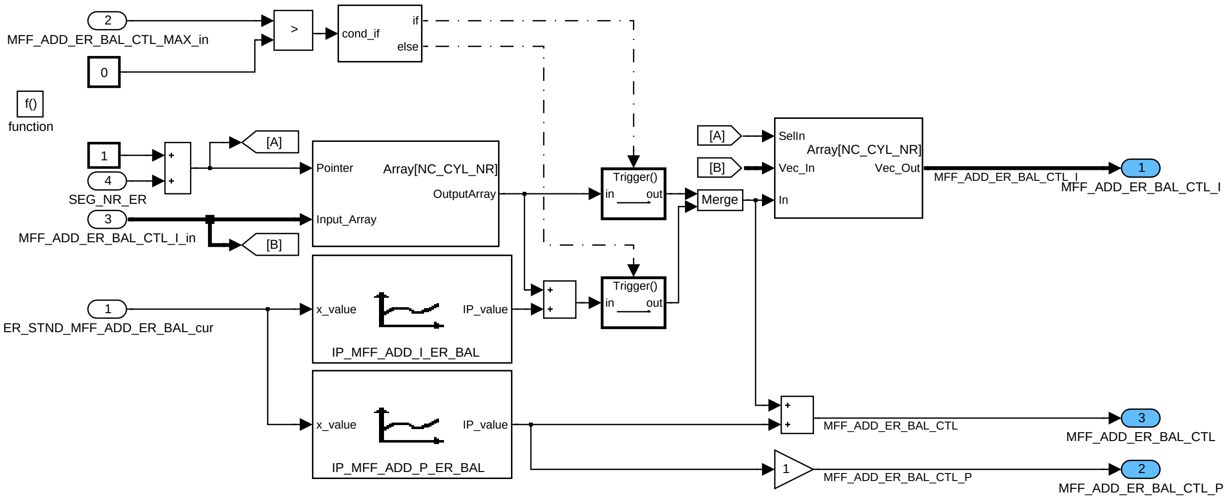


Figure 108 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ CTL_EQUATIONS

Limitation of the Controller output

The outputs of the controllers are limited to a calibrateable threshold. If one controller output has reached the limit, a flag is set at one and the corresponding segment number is displayed.

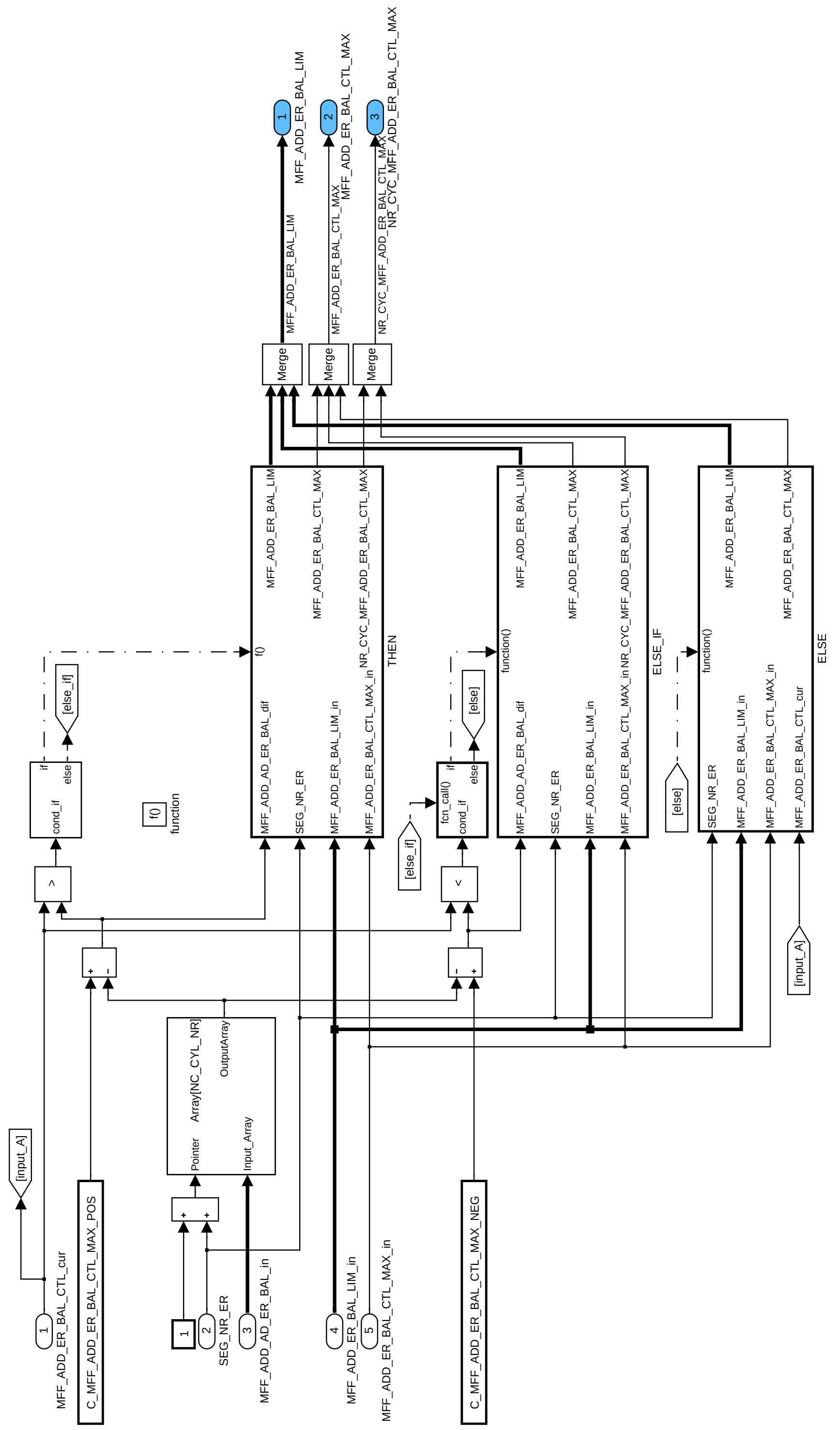


Figure 109 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ LIM_CTL_OUT

Limitation of the Controller output THEN

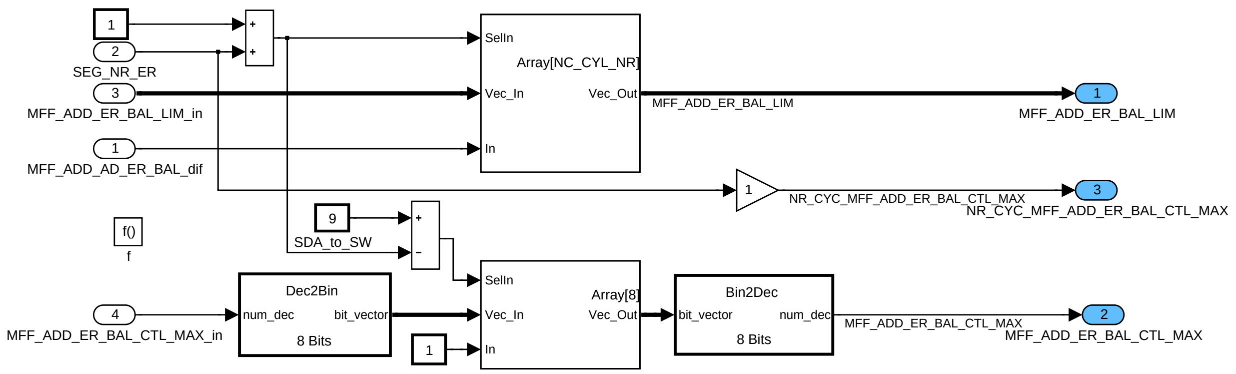


Figure 110 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ LIM_CTL_OUT/ THEN

Limitation of the Controller output ELSE_IF

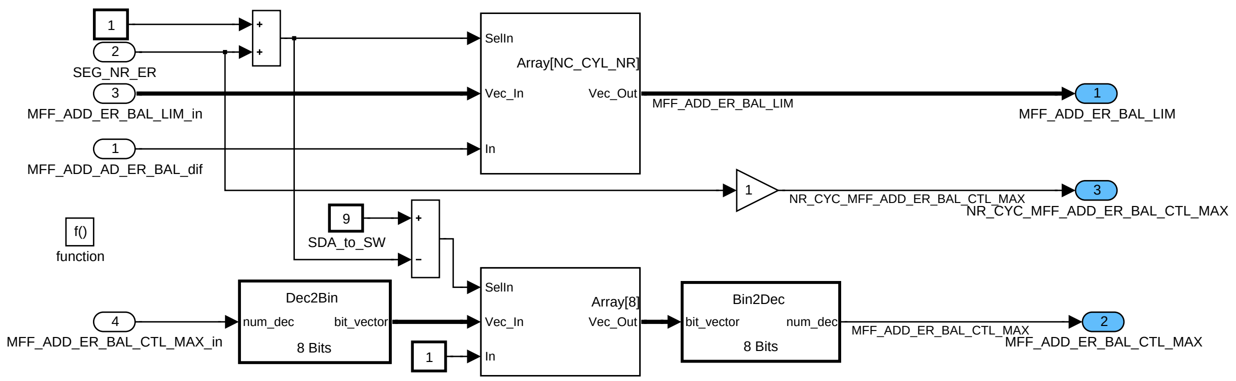


Figure 111 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ LIM_CTL_OUT/ ELSE_IF

Limitation of the Controller output ELSE

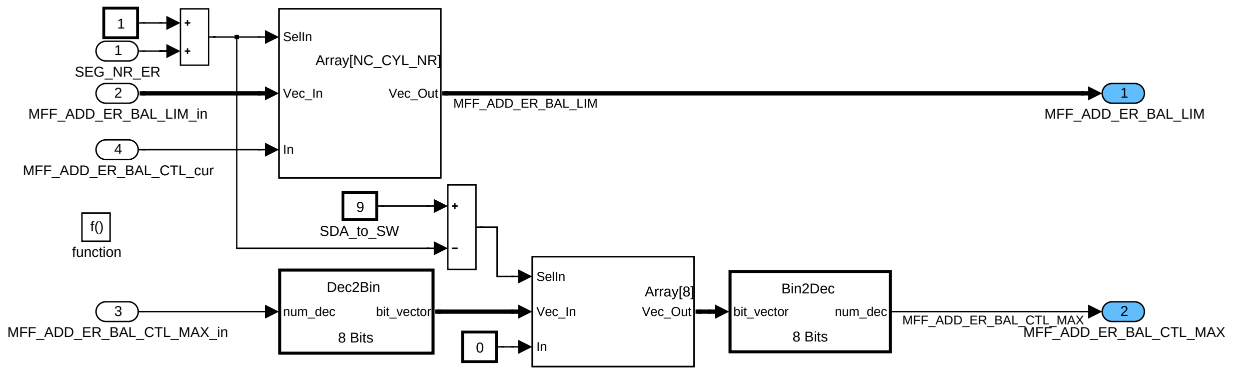


Figure 112 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ LIM_CTL_OUT/ ELSE

Correction of limited controller output values with adaptation values

The limited controller output values are corrected with the currently available adaptive correction values, which are saved in the NVMY and updated at Activation or Deactivation of the additive adaptation path (LV_MFF_ADD_AD_ER_BAL_ENA)

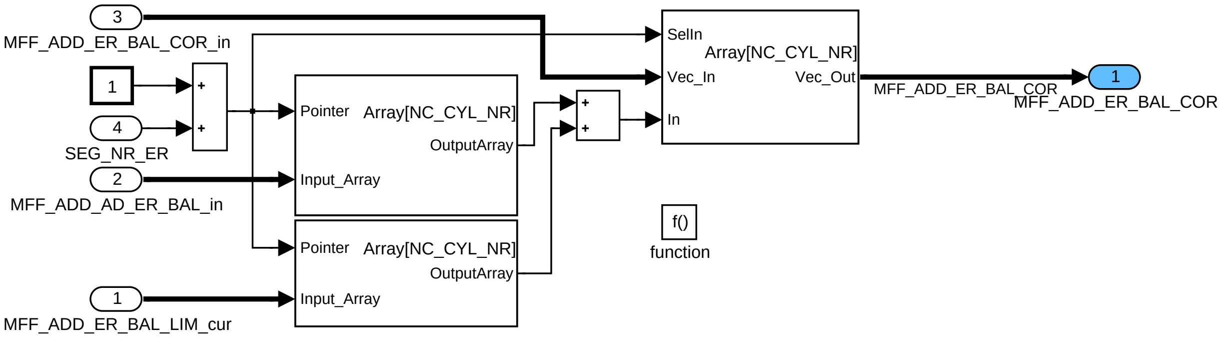


Figure 113 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ COR_LIM_CTL_OUT_VALUE

Observation of corrected controller output values for OBD issues

The additive correction values out of the correction path are observed for OBD purposes. As soon as the minimum or maximum diagnosis threshold is exceeded, a corresponding flag for the related cylinder is set. In this case the OBD system is indicated to handle the further failure treatment. The diagnosis thresholds have always to be less or equal to the absolute minimum or maximum controller output limits.

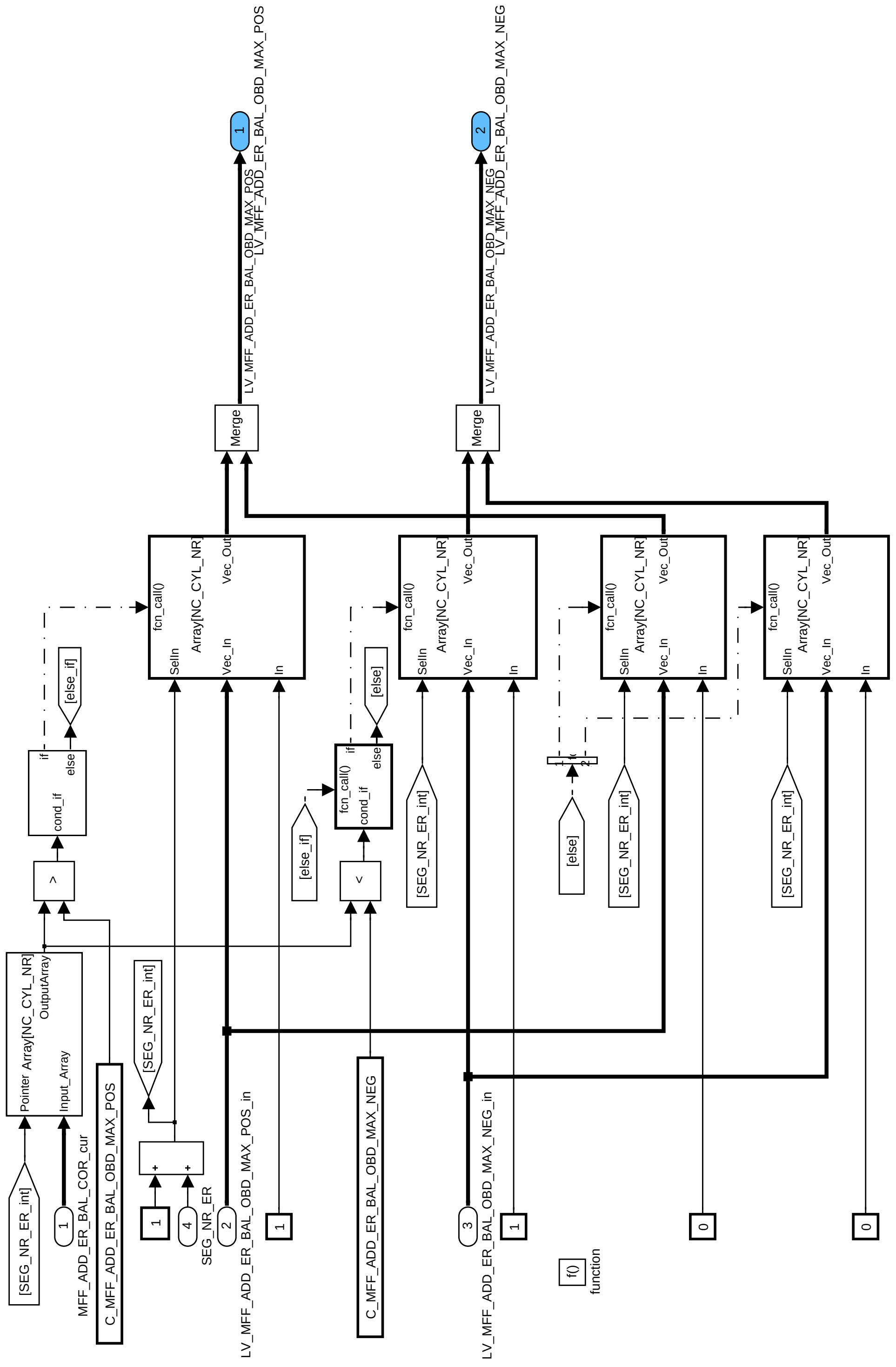


Figure 114 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_COR_VALUE/ COR_CTL_OUT_VALUE_OBD

Calculation of the additive injection time adaptation values

The corrected NC_CYL_NR controller output values are adjusted dependent on the corresponding segment number SEG_NR_ER. During the function is activated, at every new segment the corresponding MFF_ADD_ER_BAL value is updated. Due to this correction the sum of all adaptation output values remains zero to guarantee that the overall injection spreading stays unchanged.

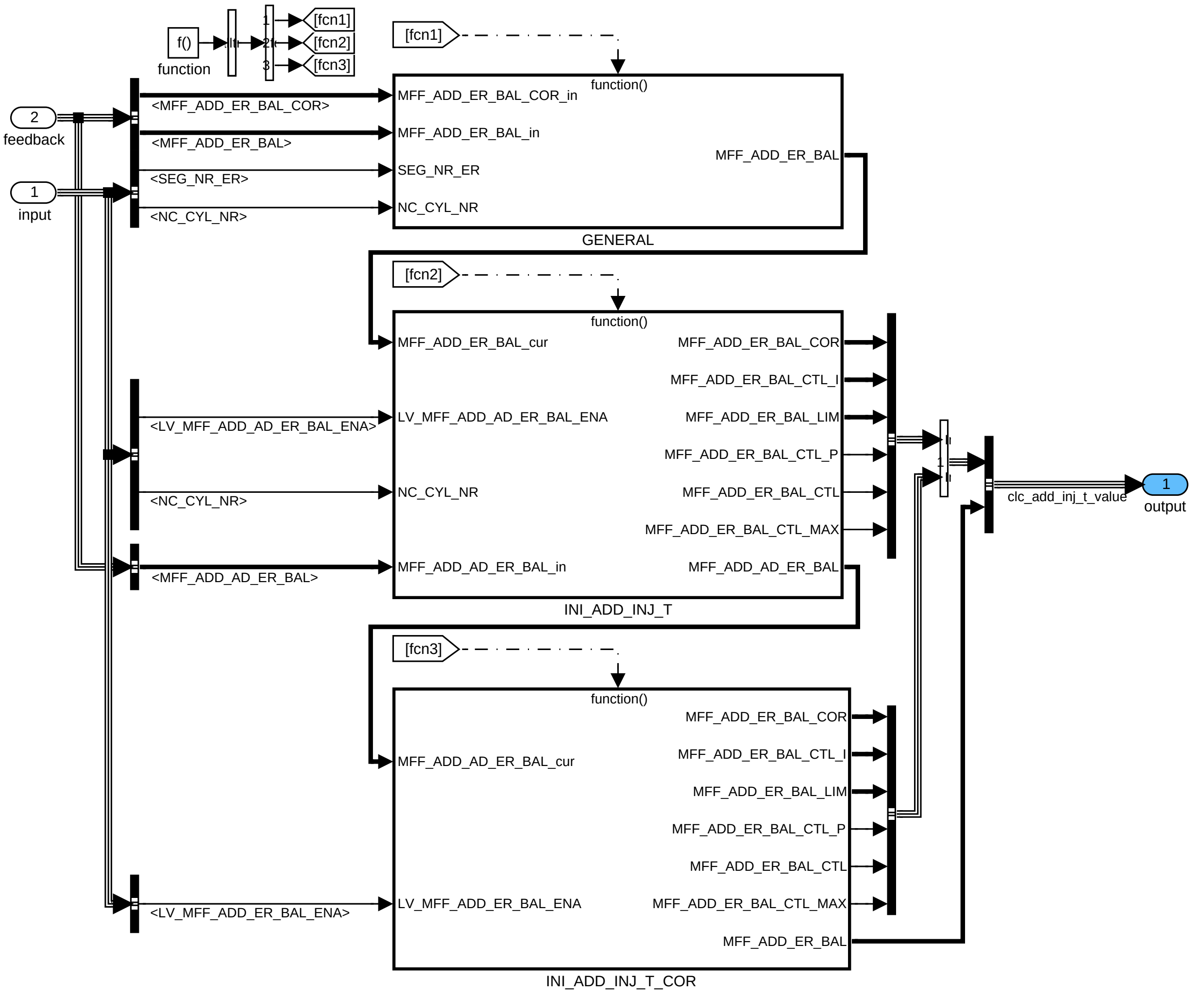


Figure 115 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_VALUE

Calculation of the additive injection time adaptation values - GENERAL

Sum of all NC_CYL_NR controller output values / number of cylinders

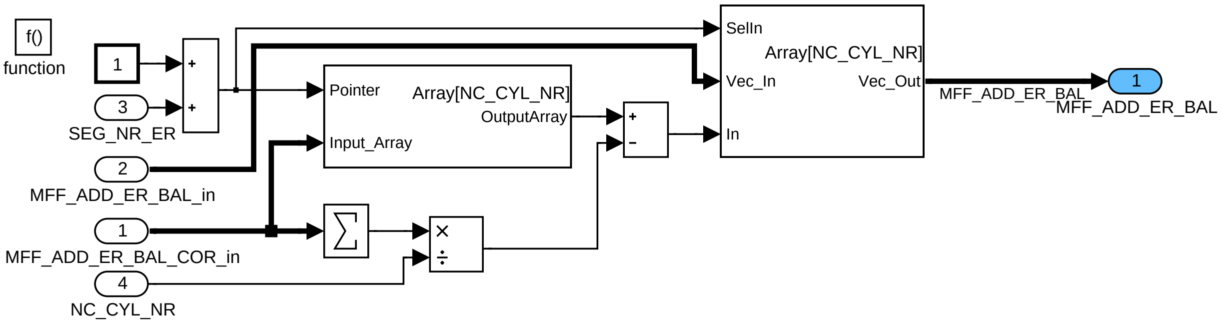


Figure 116 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_VALUE/ GENERAL

Initialization at Entry or Exit to additive injection time adaptation

As soon as the additive injection time adaptation is enabled or disabled, the stored adaptive correction values are overwritten with the currently present global adaptive value and the controller output values are set to zero.

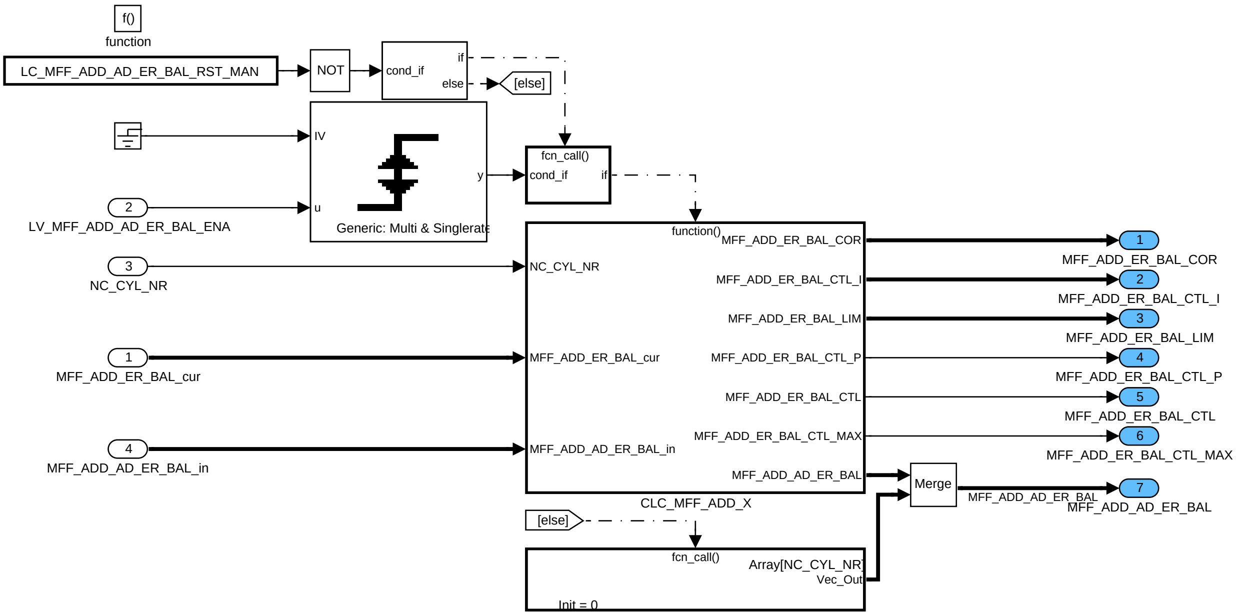


Figure 117 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_VALUE/ INI_ADD_INJ_T

Initialization at Entry or Exit to additive injection time adaptation - CLC_MFF_ADD_X

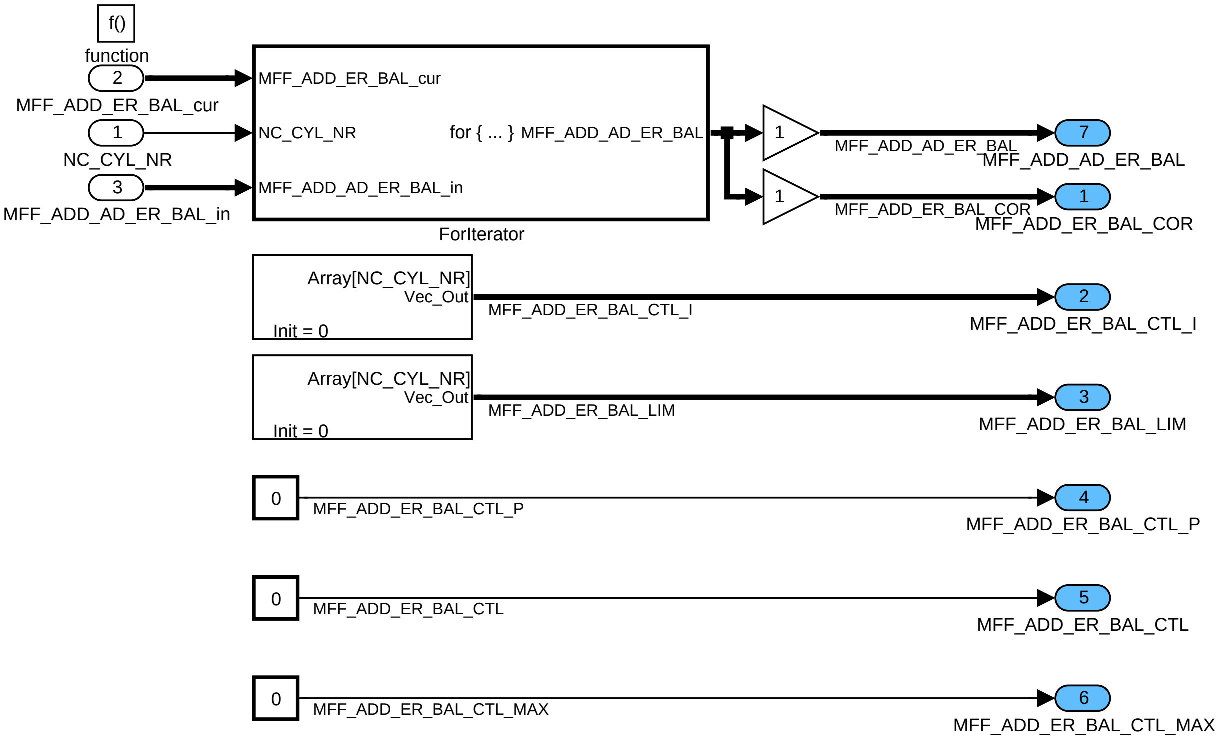


Figure 118 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_VALUE/ INI_ADD_INJ_T/ CLC_MFF_ADD_X

Initialization at Entry or Exit to additive injection time adaptation - CLC_MFF_ADD_X - FORITERATOR

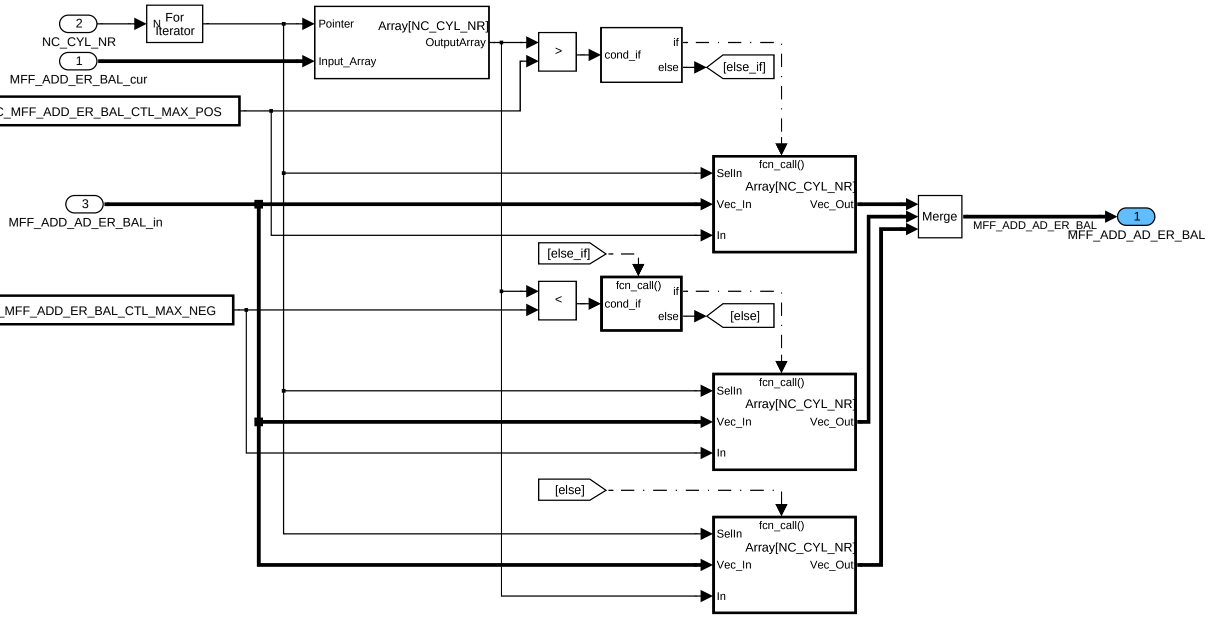


Figure 119 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_VALUE/ INI_ADD_INJ_T/ CLC_MFF_ADD_X/ ForIterator

Initialization at Exit from additive injection time correction

As soon as the additive injection time correction is disabled in combination with the manual switch to force the overall correction (LC_MFF_ADD_ER_BAL_COR), the controller output values are set to zero.

The initialization only takes place, if the logical constant for a manual reset of the additive controller values is set to zero (LC_MFF_ADD_ER_BAL_RST_MAN = 0).

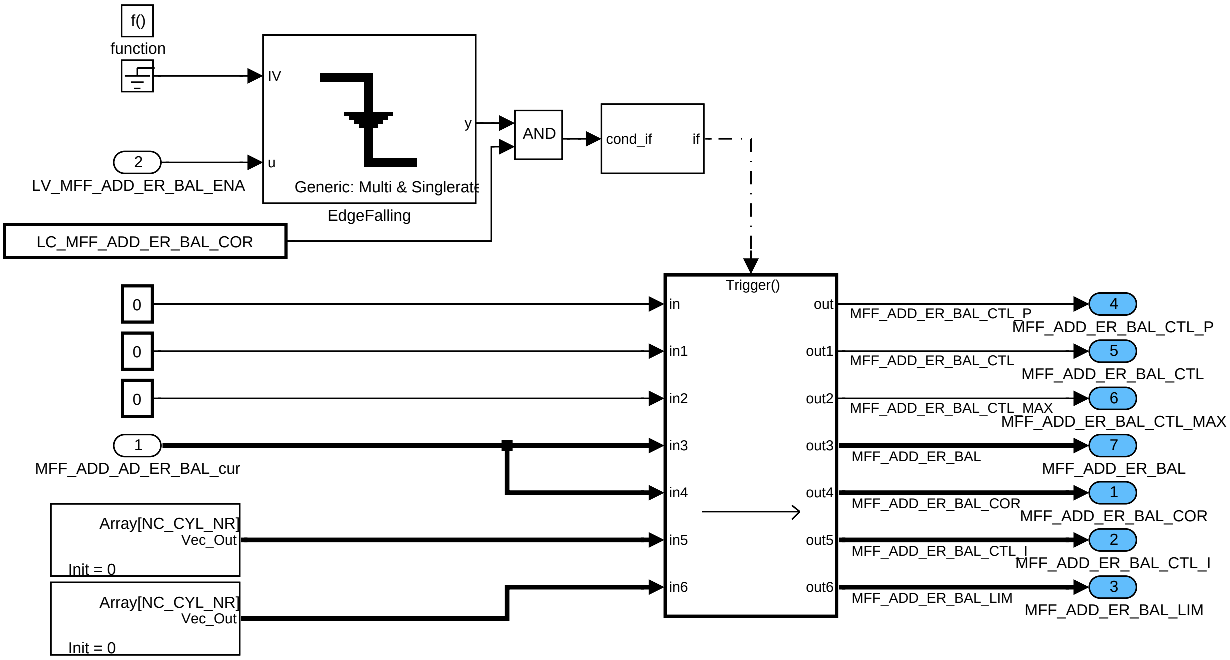


Figure 120 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_VALUE/ INI_ADD_INJ_T_COR

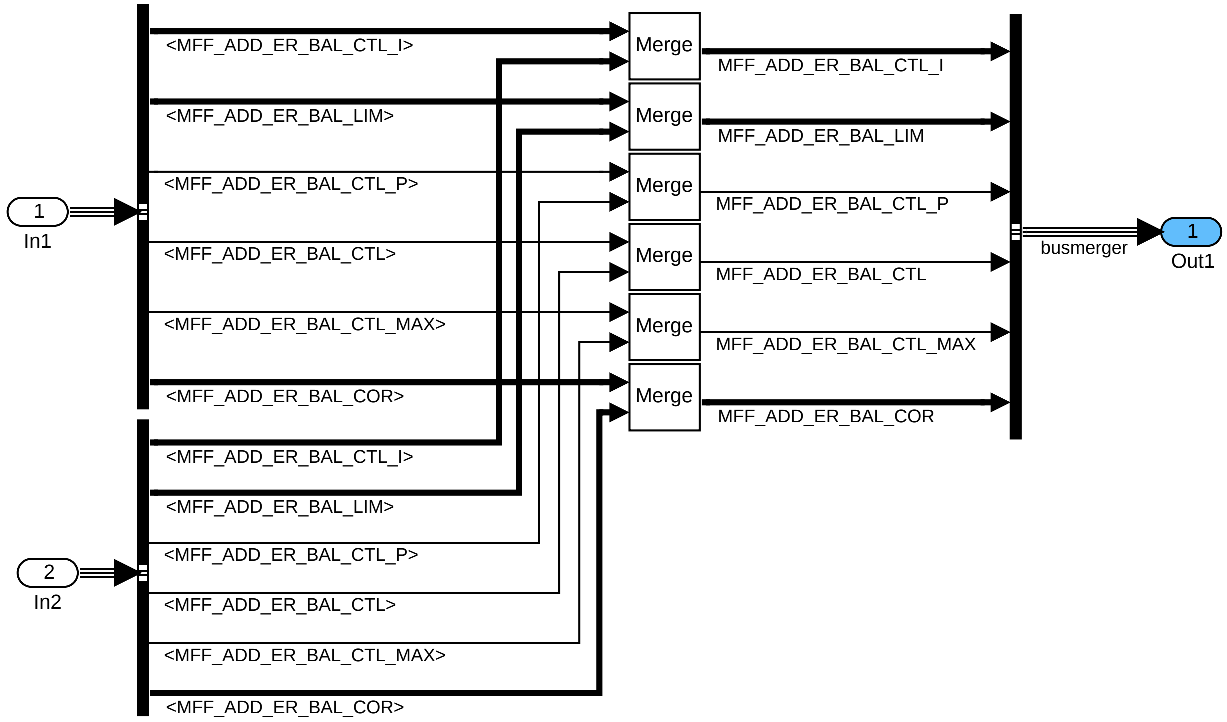


Figure 121 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_ADD_INJ_T_VALUE/ BusMerger

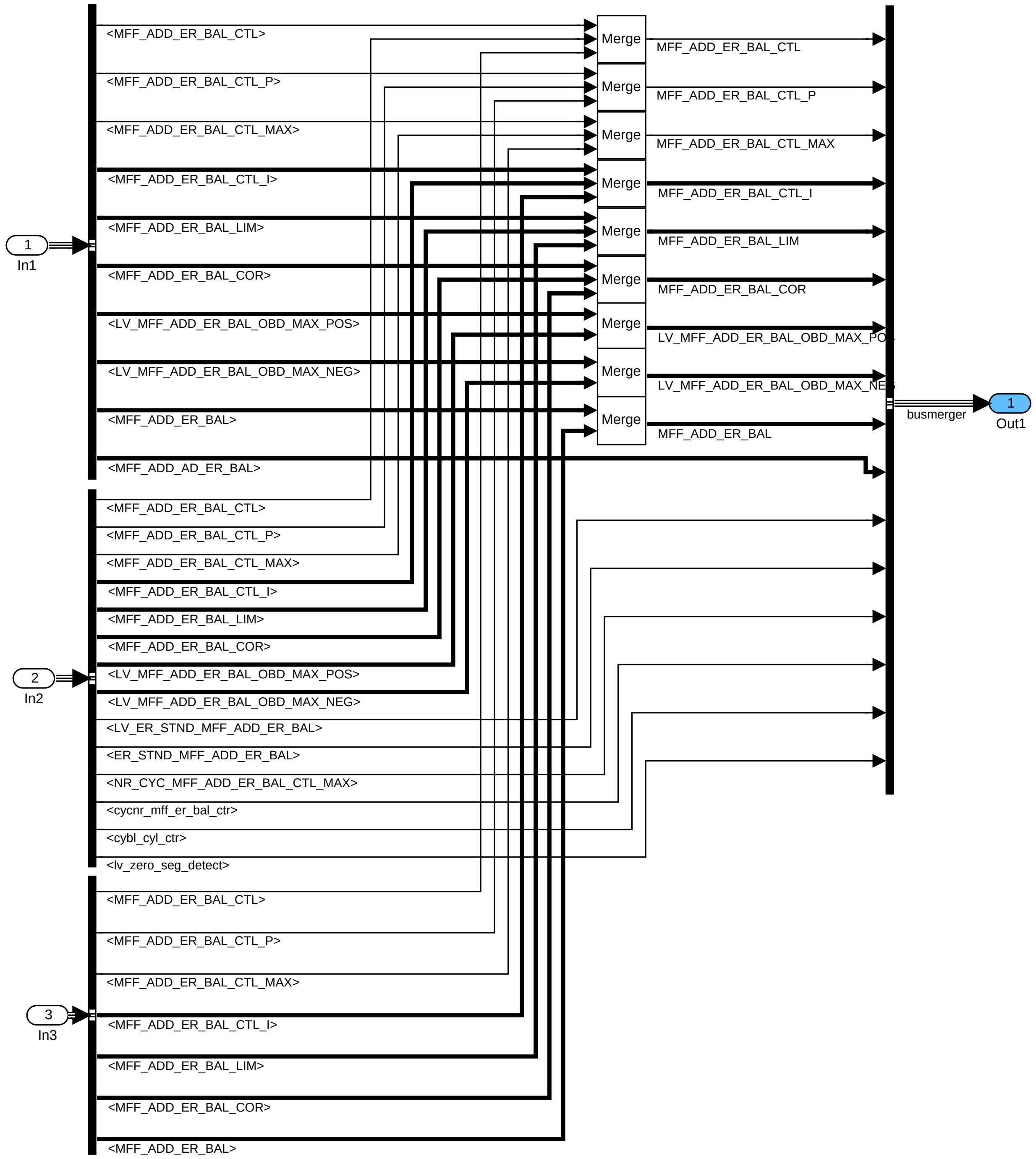


Figure 122 CYBL_M30708501/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ BusMerger

Chapter Cylinder balancing	Baseline 4DC3940S		Include File 30708501.00D
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 963 of 9643

6.5.1.3 SUBFUNCTION: SIG_MNG

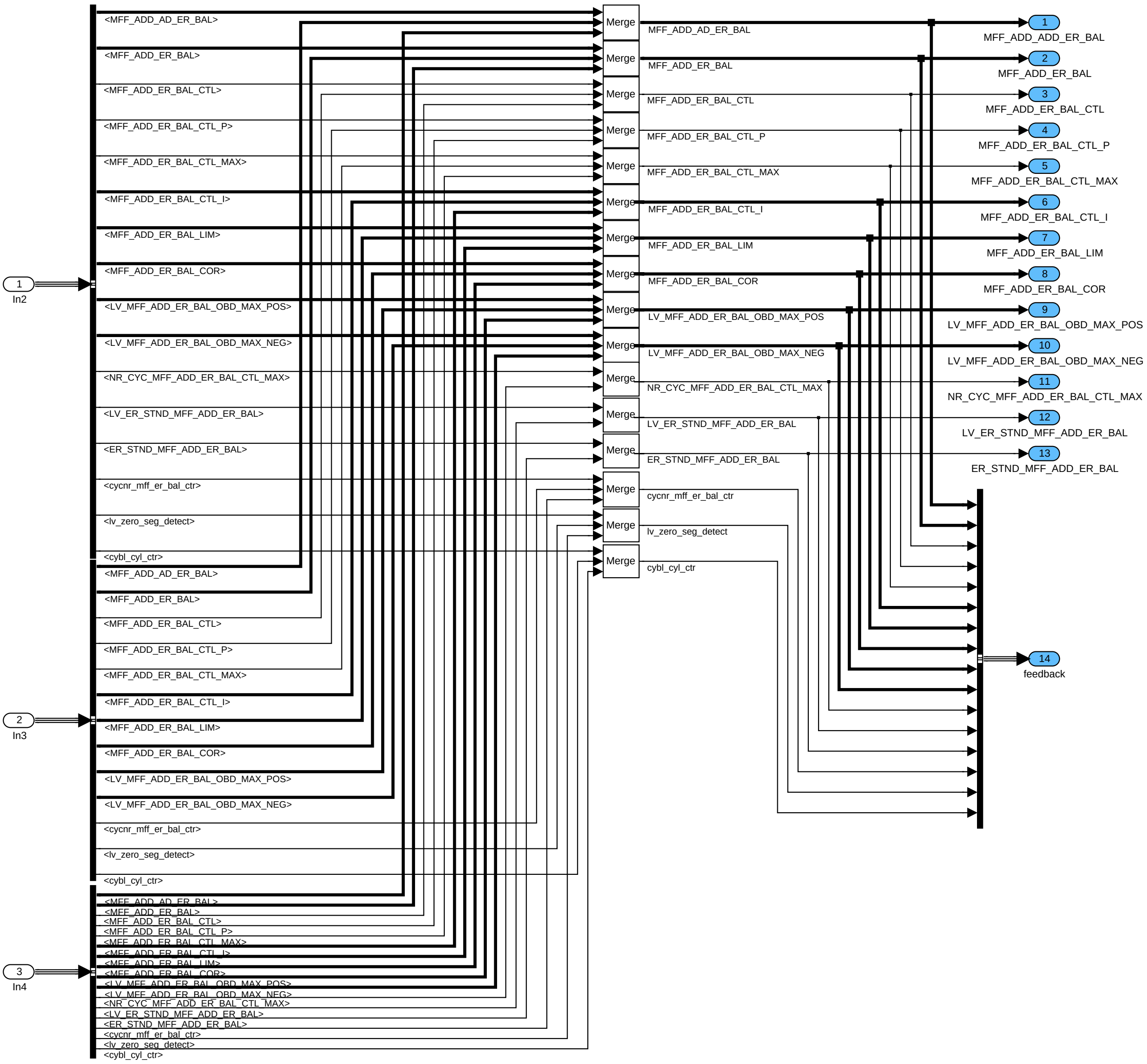


Figure 123 CYBL_M30708501/ SIG_MNG

6.6 Cylinder balancing via multiple TI intervention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TI_AD_ER_BAL[NC_CYL_NR]	O/V/S	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Multiple adaptation value for cylinder balancing via TI intervention					
FAC_TI_ER_BAL[NC_CYL_NR]	O/V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Multiple correction value for cylinder balancing via TI intervention					
LV_MFF_FAC_ER_BAL_OBD_MAX_POS[NC_CYL_NR]	O/V	0...1H	0...1	1	-
Flag for maximum OBD limit reached at multiple part of engine roughness based cylinder balancing					
LV_MFF_FAC_ER_BAL_OBD_MAX_NEG[NC_CYL_NR]	O/V	0...1H	0...1	1	-
Flag for minimum OBD limit reached at multiple part of engine roughness based cylinder balancing					
ER_STND_FAC_TI_ER_BAL	V	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Controller input value at multiple path of cylinder balancing via TI intervention					
LV_ER_STND_FAC_TI_ER_BAL	V	0...1H	0...1	1	-
Flag for controller input at multiple path of cylinder balancing via TI intervention					
FAC_TI_ER_BAL_CTL	V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Controller output value at multiple path of cylinder balancing via TI intervention					
FAC_TI_ER_BAL_COR[NC_CYL_NR]	V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Corrected multiple correction value for cylinder balancing via TI intervention					
NR_CYC_FAC_TI_ER_BAL_CTL_MAX	V	0...FFH	0...255	1	-
Corresponding segment number of the controller that has reached the maximum limit (multiple TI path)					
FAC_TI_ER_BAL_LIM[NC_CYL_NR]	V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Limited controller output value at multiple path of cylinder balancing via TI intervention					
FAC_TI_ER_BAL_CTL_I[NC_CYL_NR]	V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
I-part of the controller output value at multiple path of cylinder balancing via TI intervention					
FAC_TI_ER_BAL_CTL_MAX	V	0...FFFFH	0...6.5535E+4	1	-
Controller output value maximum limit reached at multiple path of cylinder balancing via TI intervention					
FAC_TI_ER_BAL_CTL_P	V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
P-part of the controller output value at multiple path of cylinder balancing via TI intervention					

Input data:

ER_STND_MMV_DIF_BAL[NC_CYL_NR]	LV_FAC_TI_ER_BAL_ENA	SEG_NR_ER	NC_CYL_NR
LV_TI_ER_BAL_ACT	LV_FAC_TI_AD_ER_BAL_ENA	LV_MFF_FAC_AD_ER_BAL_EXT_ADJ	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CYCNR_FAC_TI_ER_BAL	1	1...FFH	1...255	1	-
Number of cycles for calculation of controller output values for cylinder balancing via multiple TI intervention					
C_ER_STND_THD_NEG_FAC_TI_BAL	1	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Negative threshold for ER input value at multiple path of cylinder balancing via TI intervention					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ER_STND_THD_POS_FAC_TI_BAL	1	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Positive threshold for ER input value at multiple path of cylinder balancing via TI intervention					
C_FAC_TI_ER_BAL_CTL_MAX_NEG	1	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Negative maximum limit for controller output at multiple path of cylinder balancing via TI intervention					
C_FAC_TI_ER_BAL_CTL_MAX_POS	1	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Positive maximum limit for controller output at multiple path of cylinder balancing via TI intervention					
C_FAC_TI_ER_BAL_OBD_MAX_NEG	1	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Negative maximum OBD limit for multiple correction value of cylinder balancing via TI intervention					
C_FAC_TI_ER_BAL_OBD_MAX_POS	1	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Positive maximum OBD limit for multiple correction value of cylinder balancing via TI intervention					
C_FAC_TI_ER_BAL_RST	1	0H 1H 2H	INI_ZERO INI_CTL INI_AD	1	-
Reset of multiple correction- and adaptation values for cylinder balancing via TI intervention					
LC_FAC_TI_AD_ER_BAL_RST_MAN	1	0...1H	0...1	1	-
Manual reset of multiple adaptation values for cylinder balancing via TI intervention					
LC_FAC_TI_ER_BAL_COR	1	0...1H	0...1	1	-
Manual switch for overall correction of controller output values for cylinder balancing via multiple TI intervention					
LC_FAC_TI_ER_BAL_RST_MAN	1	0...1H	0...1	1	-
Manual reset of multiple correction values for cylinder balancing via TI intervention					
IP_FAC_TI_I_ER_BAL	8	0...FFFFH	-1...0.99996948	3.05176E-5	-
LDP_ER_STND_FAC_TI_IP_I_ER_BAL	8	0...FFFFH	-325.78...325.77	0.0099313	1/s²
Controller I-part value for multiple path of cylinder balancing via TI intervention					
IP_FAC_TI_P_ER_BAL	8	0...FFFFH	-1...0.99996948	3.05176E-5	-
LDP_ER_STND_FAC_TI_IP_P_ER_BAL	8	0...FFFFH	-325.78...325.77	0.0099313	1/s²
Controller P-part value for multilple path of cylinder balancing via TI intervention					

6.6.1 General information:

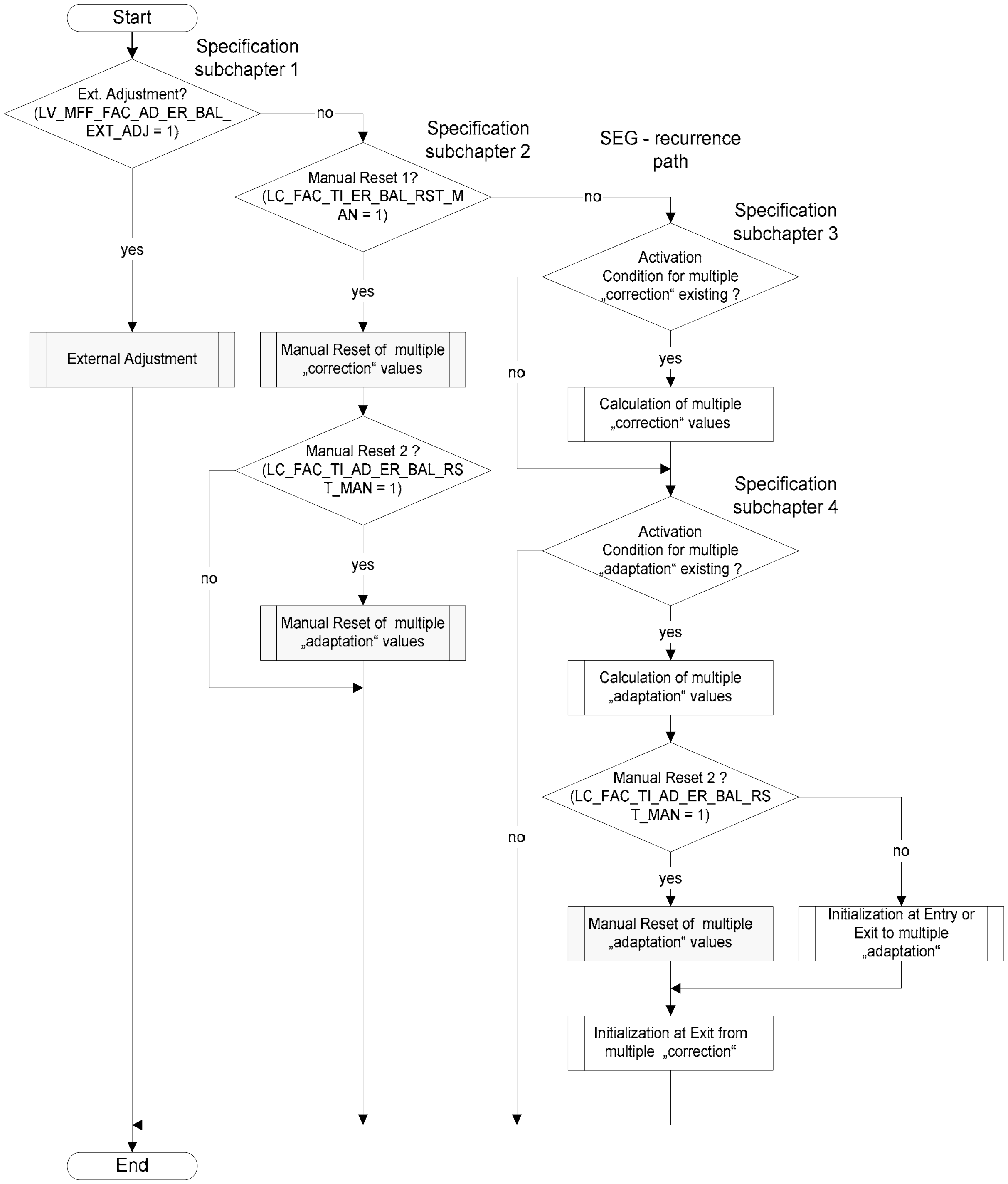
The calculation of the multiple correction- or adaptation values is performed, depending on the settings of the corresponding logical activation variable. LV_FAC_TI_ER_BAL_ENA is used to enable the correction path, while LV_FAC_TI_AD_ER_BAL_ENA enables the adaptation path. As soon as the corresponding logical value is set to "1" in combination with additional activation conditions, the calculation is started.

The calculated additive adaptation value FAC_TI_AD_ER_BAL[NC_CYL_NR] for each cylinder is stored in the non-volatile memory to be available for initialization issues at next engine run.

With use of the manual reset it is possible to initialize the multiple correction- or adaptation values with zero at any time. In this case, the flags which indicate the exceeding of a controller output limitation are set to zero as well (refer: "Initialization at Manual Reset")

Signal flow diagram:

general specification



Application Condition

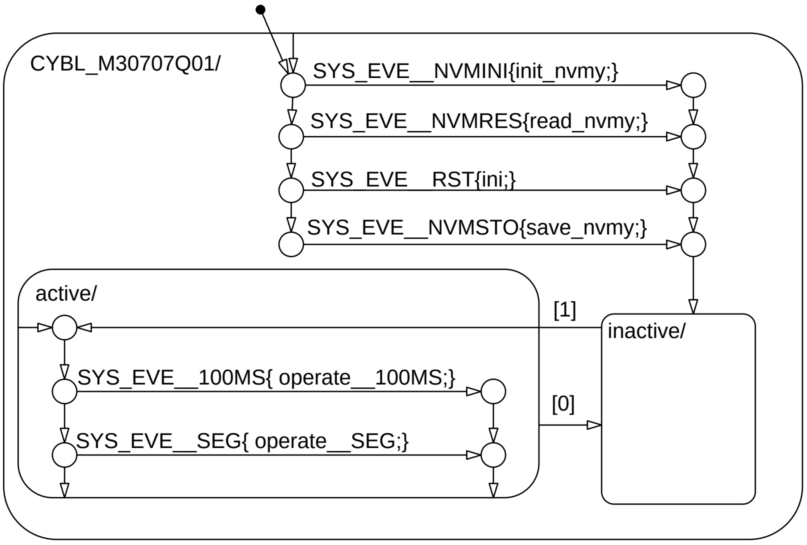


Figure 124 CYBL_M30707Q01/ APP_CDN/ Chart

Function Description

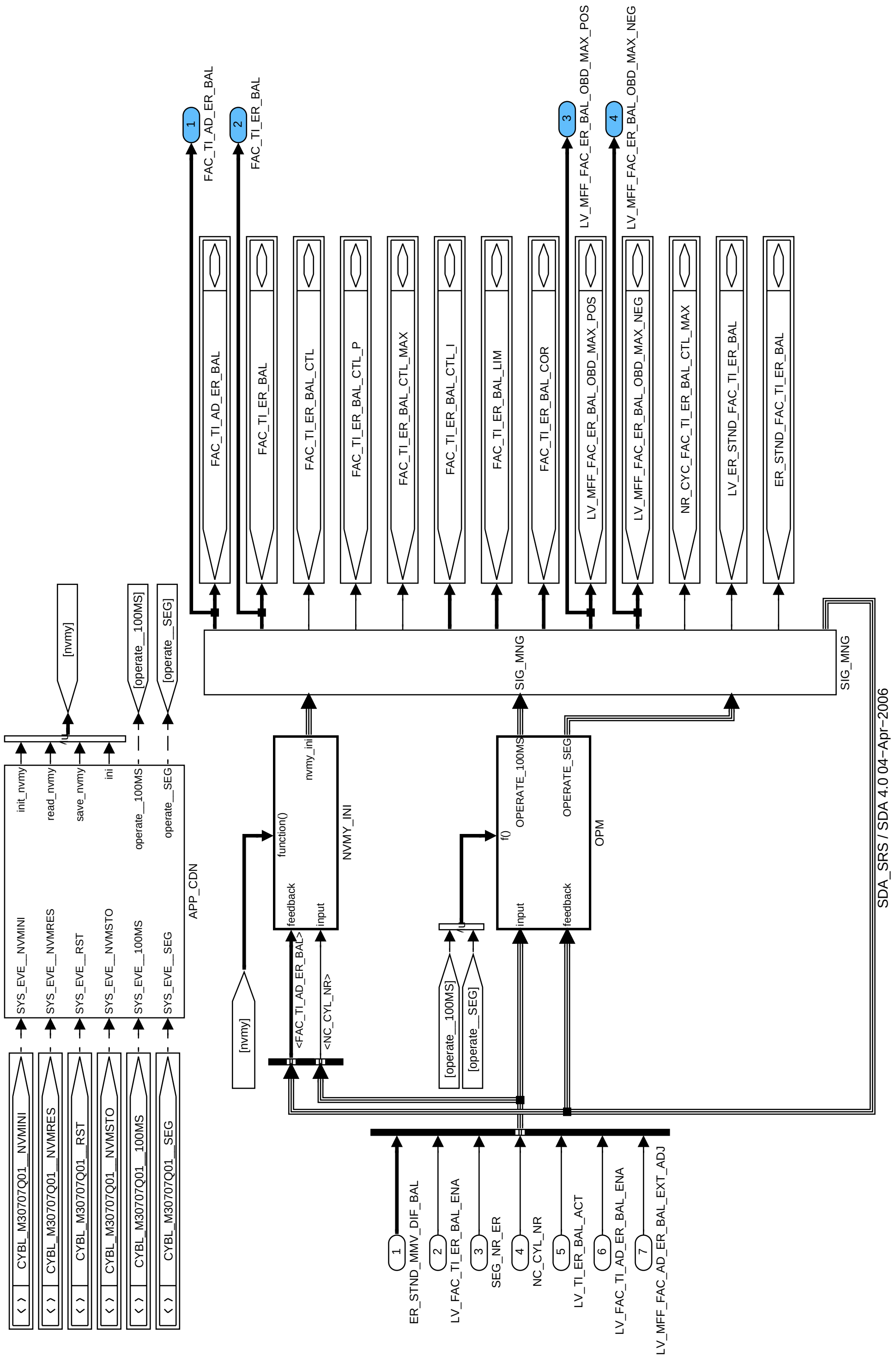


Figure 125 CYBL_M30707Q01

6.6.1.1 Calculation of non volatile memory tasks

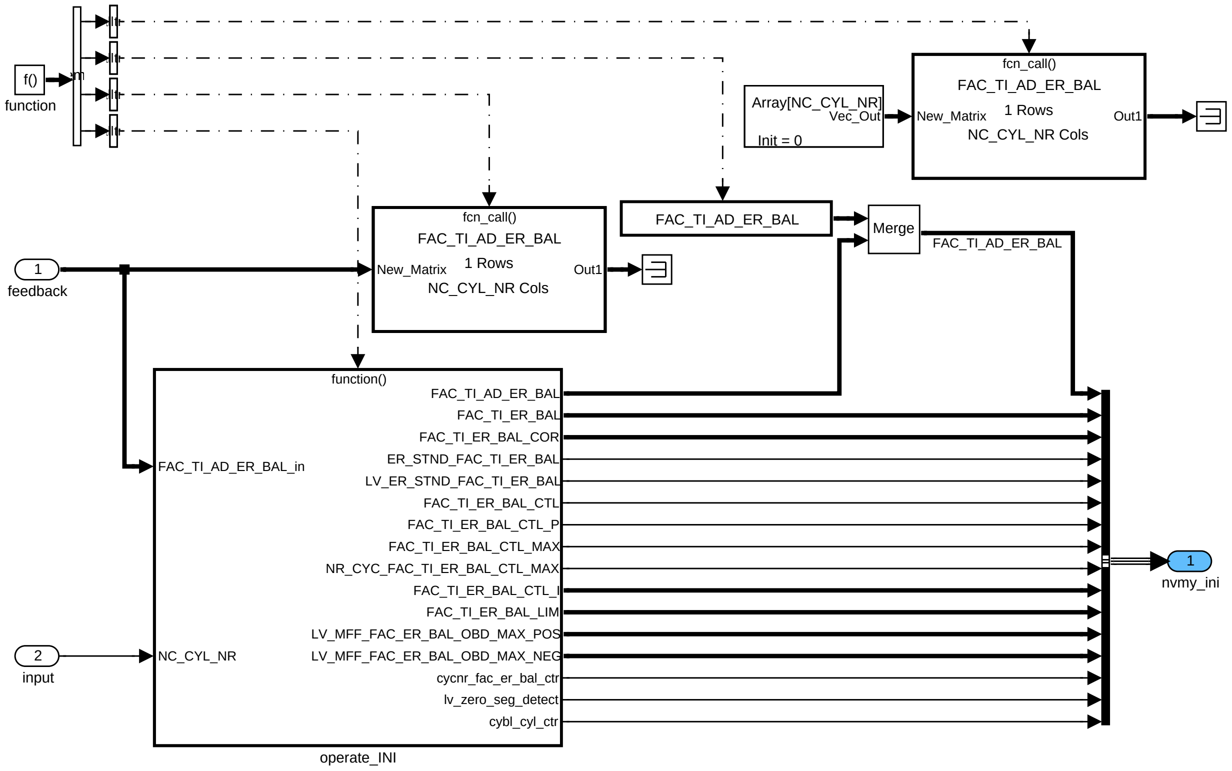


Figure 126 CYBL_M30707Q01/ NVMY_INI

Calculation of variables at reset task

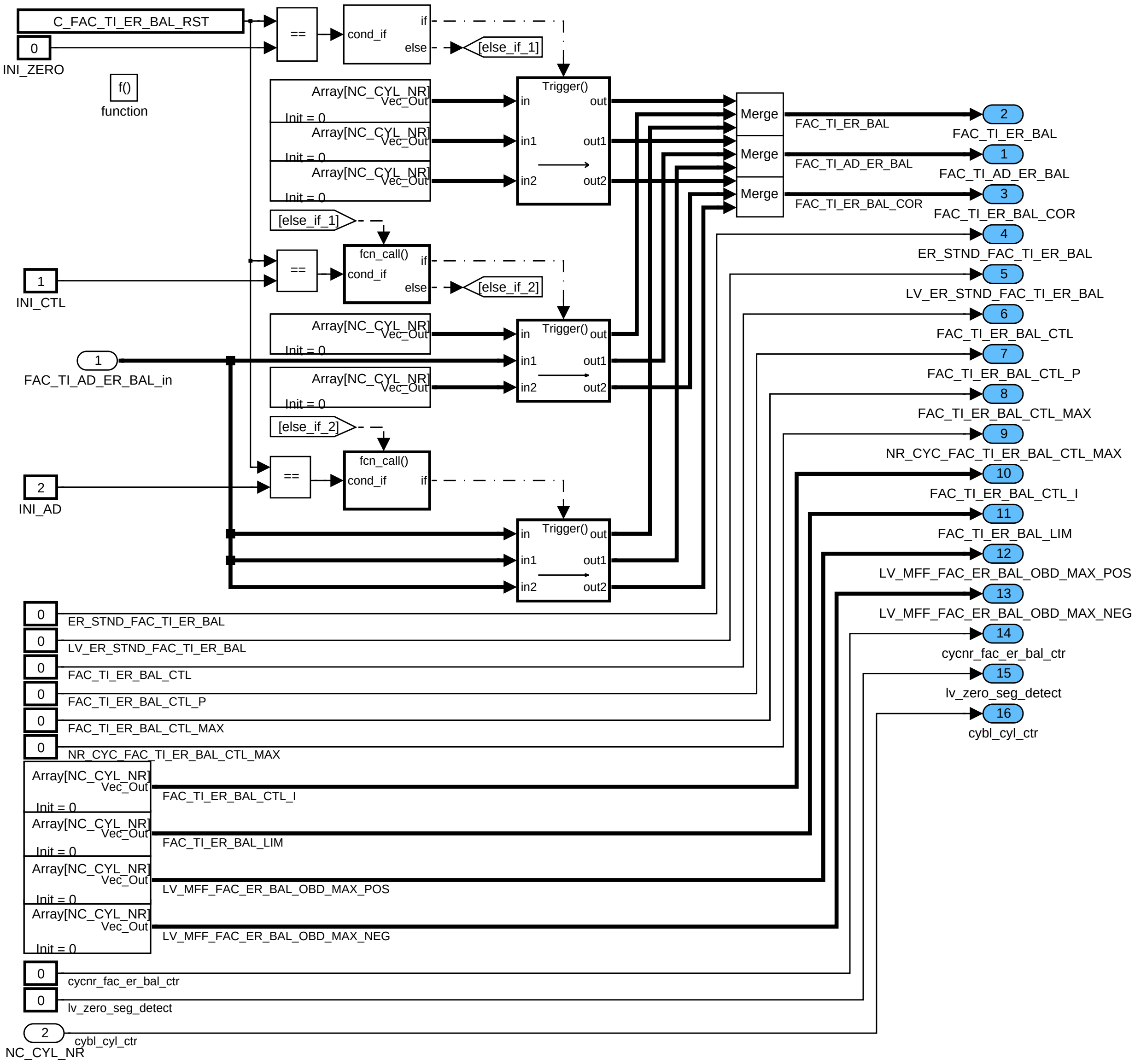


Figure 127 CYBL_M30707Q01/ NVMY_INI/ operate_INI

6.6.1.2 Calculation of 100ms and segment synchronous tasks

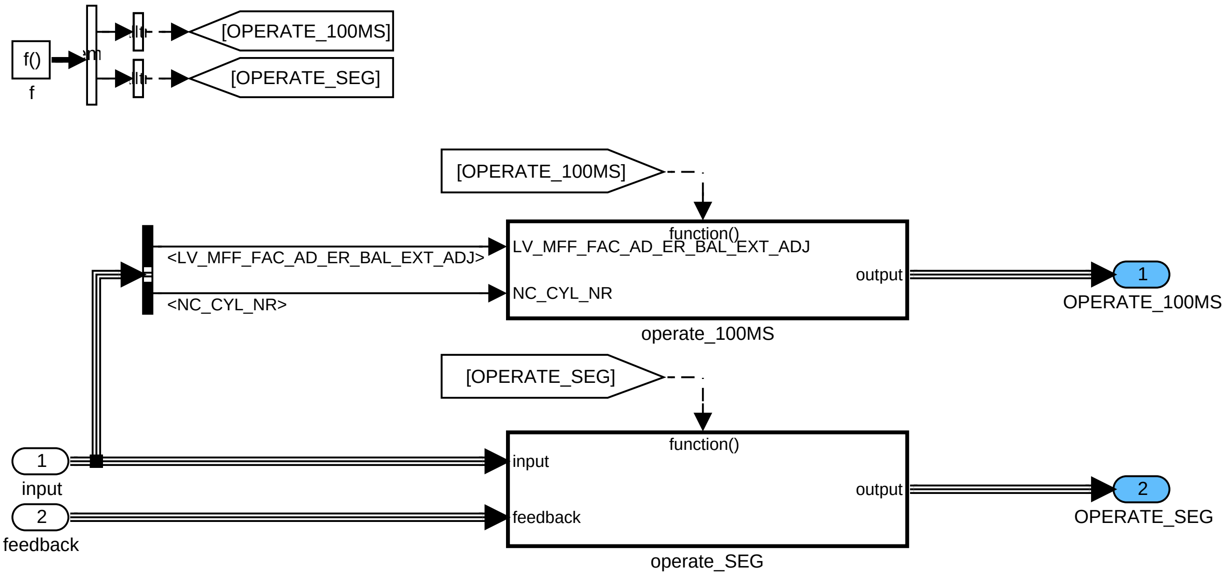


Figure 128 CYBL_M30707Q01/ OPM

External adjustment of multiple correction values

In case of an external adjustment (request from key word protocol) the additive adaptation values for all cylinders are set to zero. While the external adjustment is activated, all following subchapters within the module "Cylinder balancing via multiple TI intervention" are deactivated and the corresponding "Initialization at Deactivation" process for this chapter is blocked during this period.

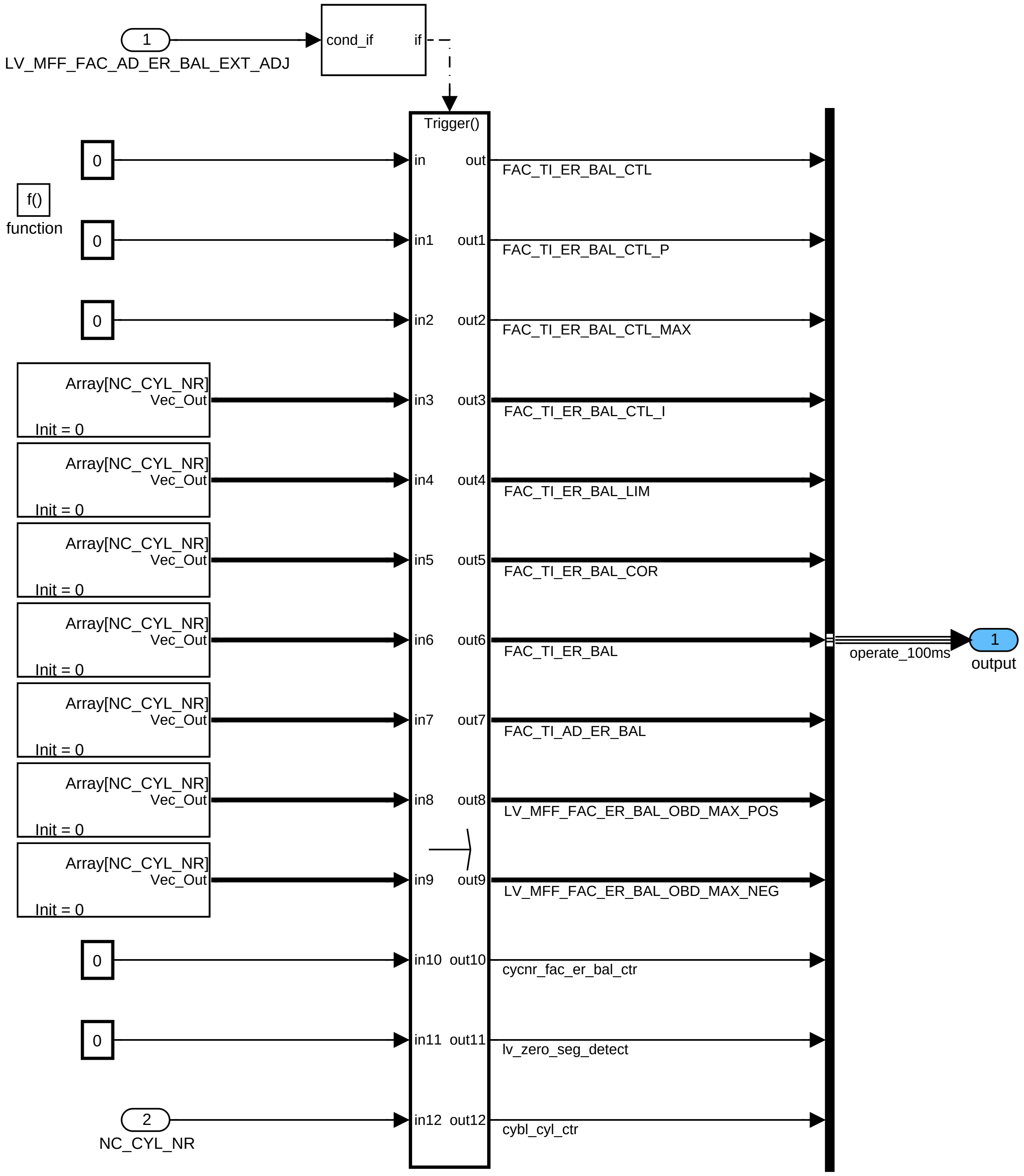


Figure 129 CYBL_M30707Q01/ OPM/ operate_100MS

Calculation of segment task

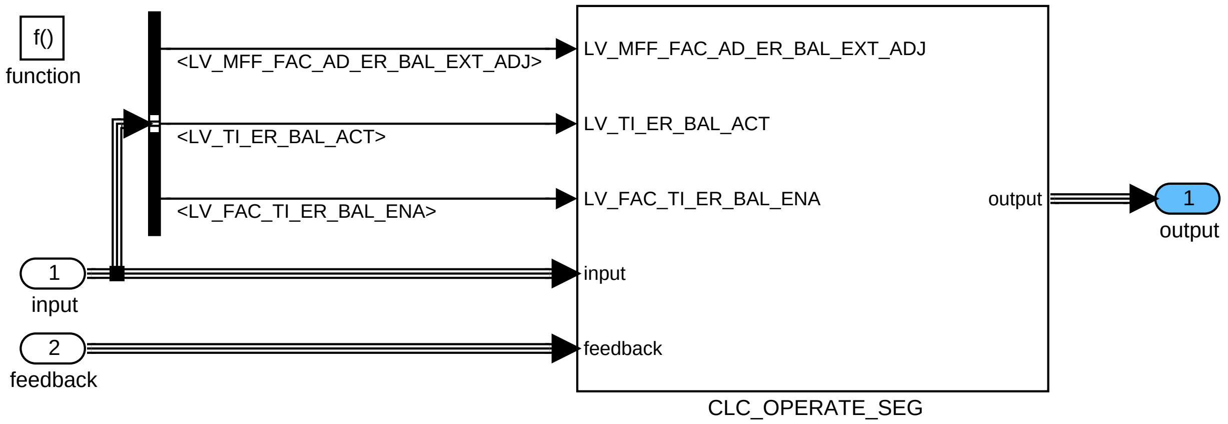


Figure 130 CYBL_M30707Q01/ OPM/ operate_SEG

Calculation of manual reset, injection time correction and injection time adaptation

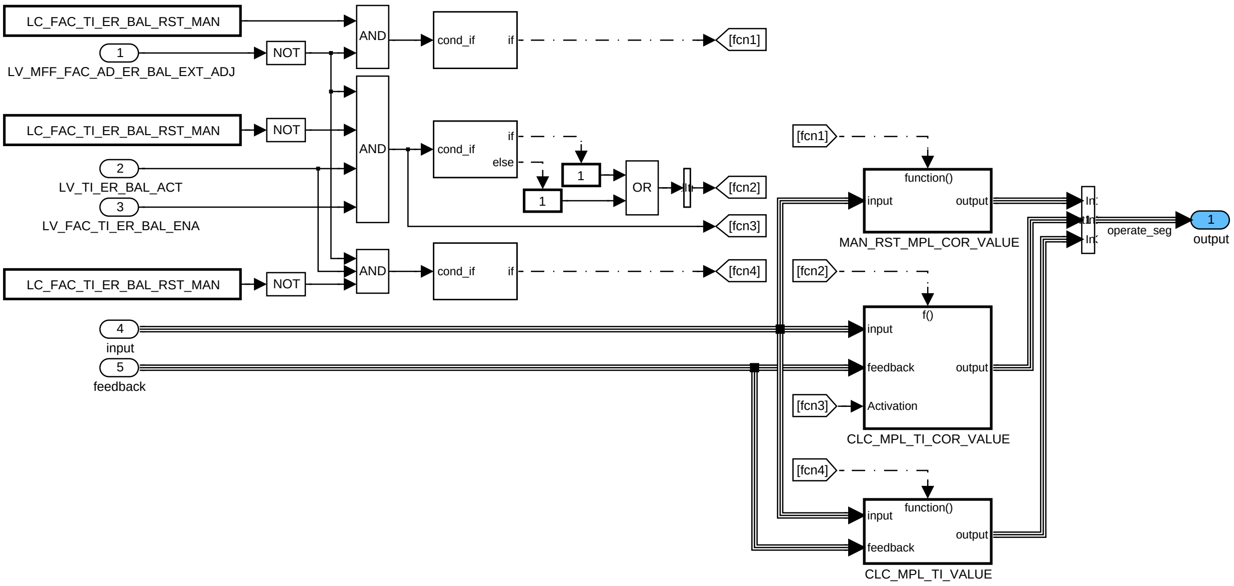


Figure 131 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG

Manual reset of multiple correction values

In case of a manual reset it is possible to adjust the exported output data to a wanted value to make the function test and handling more easier. While the manual function reset is activated, all following subchapters within the module "Cylinder balancing via multiple TI intervention" are deactivated and the corresponding "Initialization at Deactivation" process for this chapters is blocked during this period.

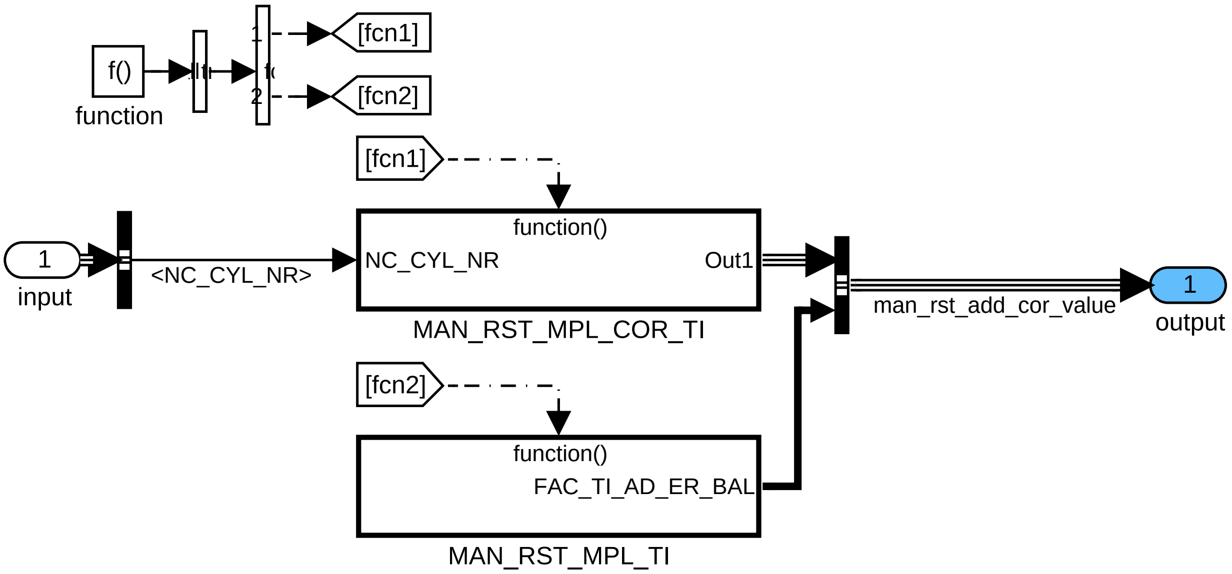


Figure 132 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ MAN_RST_MPL_COR_VALUE

Manual reset of multiple correction values for cylinder balancing via TI intervention

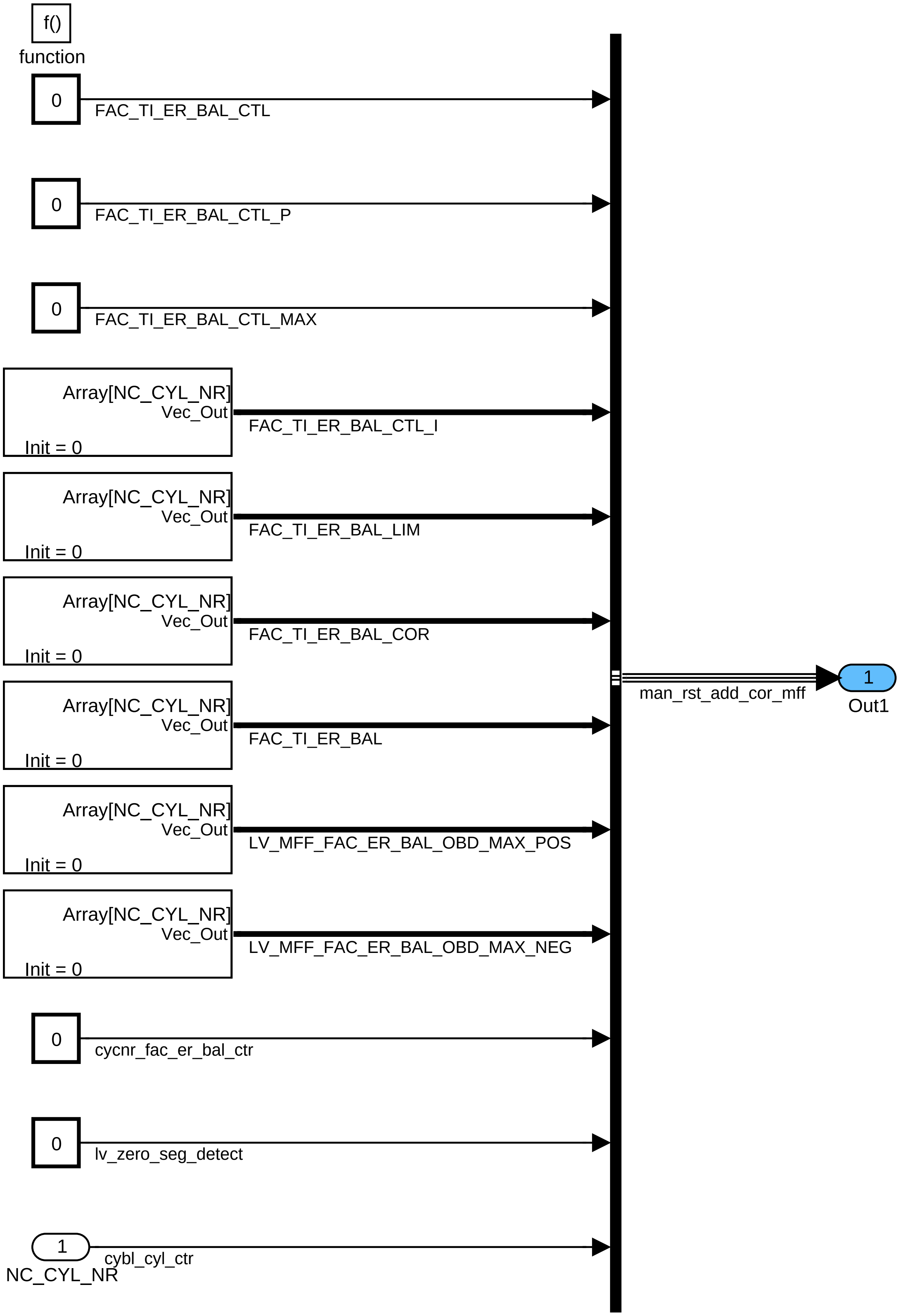

f()
function
0
FAC_TI_ER_BAL_CTL
0
FAC_TI_ER_BAL_CTL_P
0
FAC_TI_ER_BAL_CTL_MAX
Array[NC_CYL_NR]
Vec_Out
Init = 0
FAC_TI_ER_BAL_CTL_I
Array[NC_CYL_NR]
Vec_Out
Init = 0
FAC_TI_ER_BAL_LIM
Array[NC_CYL_NR]
Vec_Out
Init = 0
FAC_TI_ER_BAL_COR
Array[NC_CYL_NR]
Vec_Out
Init = 0
FAC_TI_ER_BAL
Array[NC_CYL_NR]
Vec_Out
Init = 0
LV_MFF_FAC_ER_BAL_OBD_MAX_POS
Array[NC_CYL_NR]
Vec_Out
Init = 0
LV_MFF_FAC_ER_BAL_OBD_MAX_NEG
0
cycnr_fac_er_bal_ctr
0
lv_zero_seg_detect
1
NC_CYL_NR
cybl_cyl_ctr
man_rst_add_cor_mff
1
Out1

Figure 133 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ MAN_RST_MPL_COR_VALUE/ MAN_RST_MPL_COR_TI

Manual reset of multiple adaptation values for cylinder balancing via TI intervention

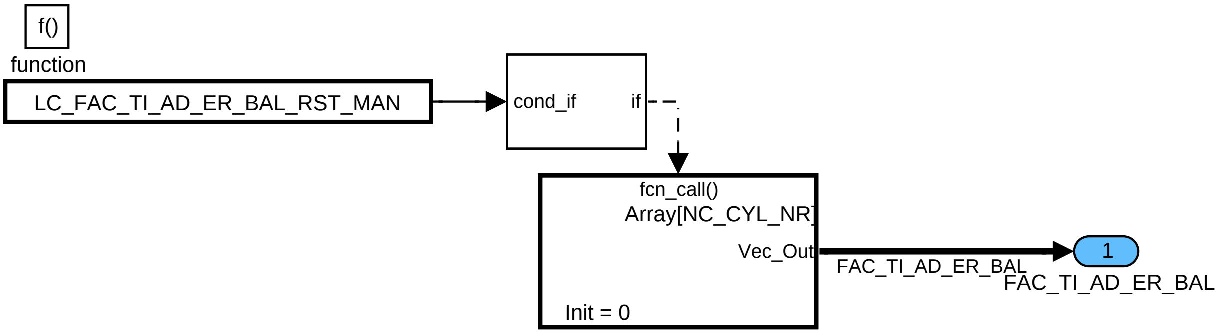


Figure 134 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ MAN_RST_MPL_COR_VALUE/ MAN_RST_MPL_TI

Structure of the multiple injection time correction values

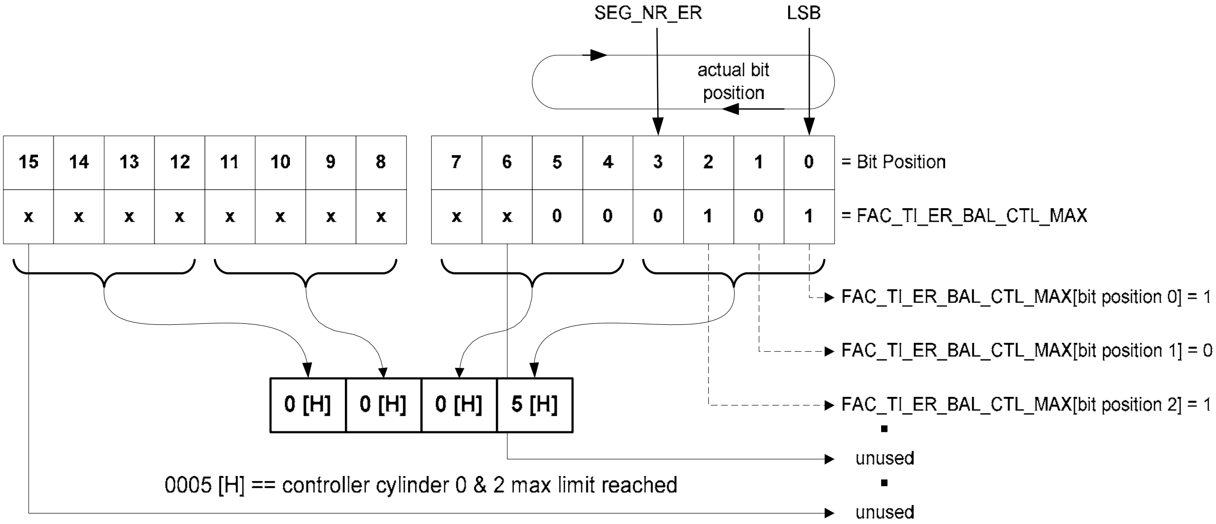
The calculation of the multiple injection time correction values are only performed if the corresponding enable bit LV_FAC_TI_ER_BAL_ENA is set to logical "1" in combination with additional activation conditions. In this case, a separate correction value for each cylinder is determined with use of a PI-controller.

As soon as a PI-controller of one cylinder is reaching its maximum, a corresponding value is set to "1" to indicate the related cylinder. To save memory resources, the information for all cylinders is located within one output data (WORD) is used to indicate one cylinder. It is possible to handle up to 16 cylinders with this method.

For a description more in detail please have a look to the signal flow diagram below. As example, a six-cylinder engine with two PI-controllers, which have reached the limits, is shown.

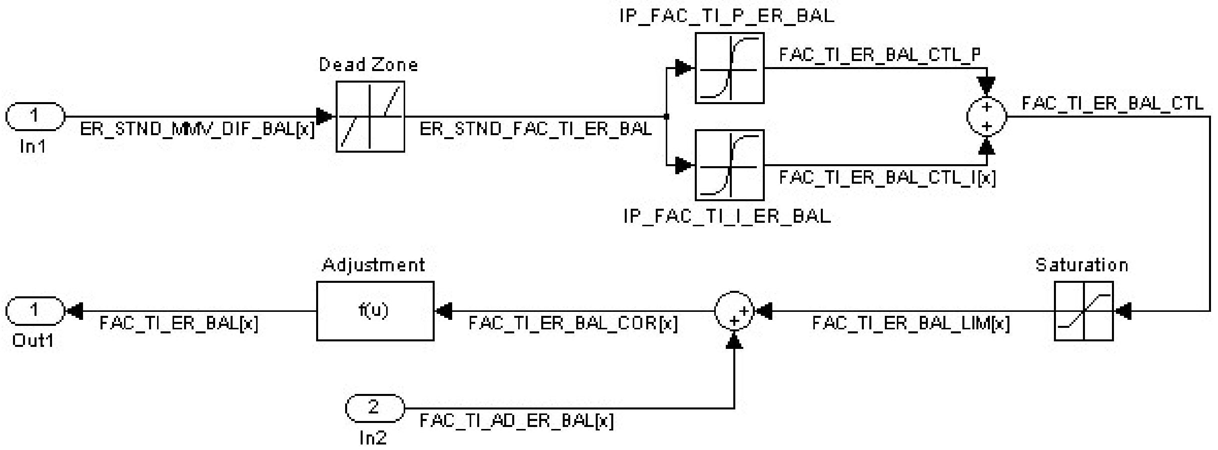
Signal flow diagram:

-Signalization of controller max. output value reached for multiple path



Note: always the complete WORD (16 BIT) is displayed as visible value

- Calculation of the multiple injection time correction / adaptation values in general



Note: [_x] is representing [SEG_NR_ER] in the signal flow diagram above

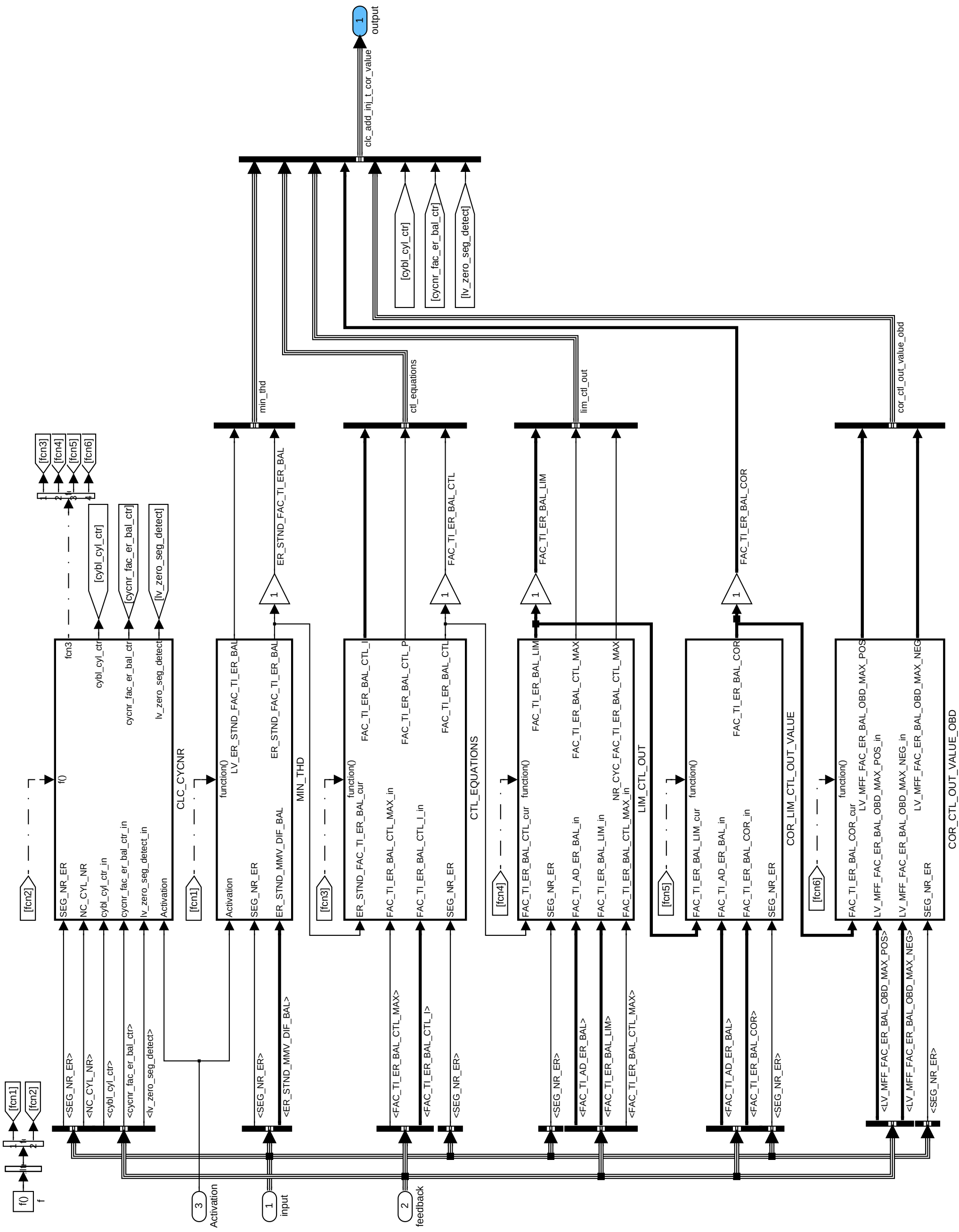


Figure 135 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE

Minimum threshold

If defined cylinder balancing input values ER_STND_MMV_DIF_BAL[SEG_NR_ER] are outside a calibrateable range, the controller is calculated. Otherwise the controller inputs are set to zero.

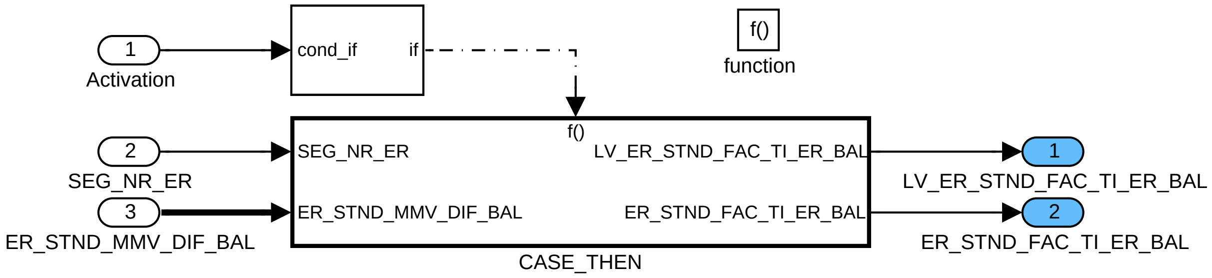


Figure 136 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ MIN_THD

Minimum threshold CASE_THEN

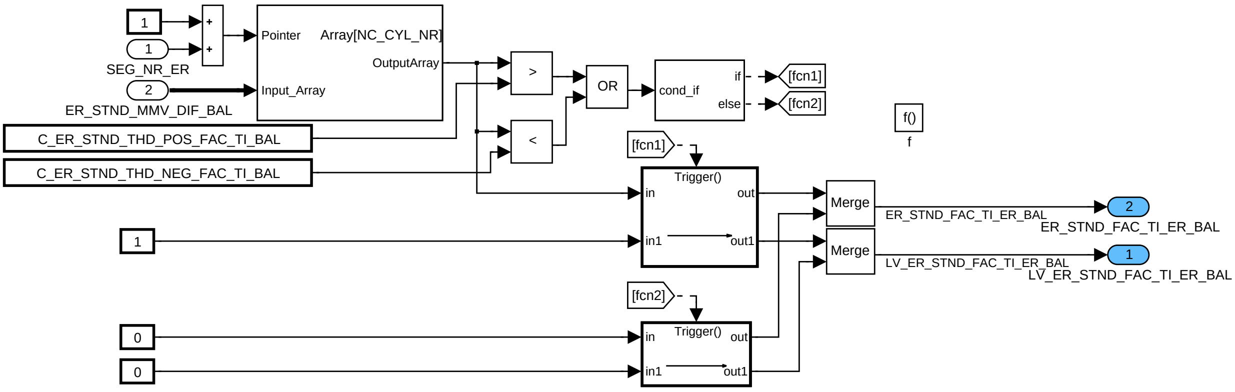


Figure 137 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ MIN_THD/ CASE_THEN

Calculation of segment tasks CLC_OPERATE_SEG CLC_MPL_TI_COR_VALUE CLC_CYCNR

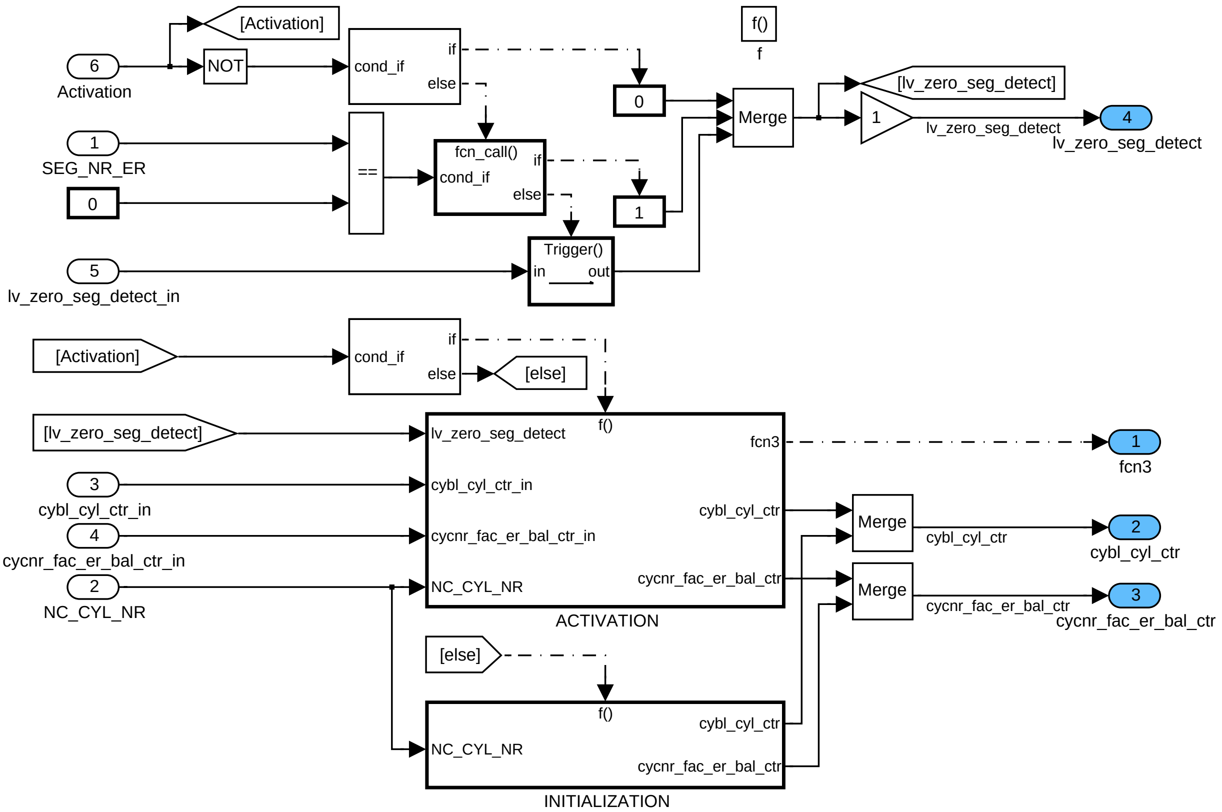


Figure 138 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ CLC_CYCNR

Calculation of segment tasks CLC_OPERATE_SEG CLC_MPL_TI_COR_VALUE CLC_CYCNR ACTIVATION

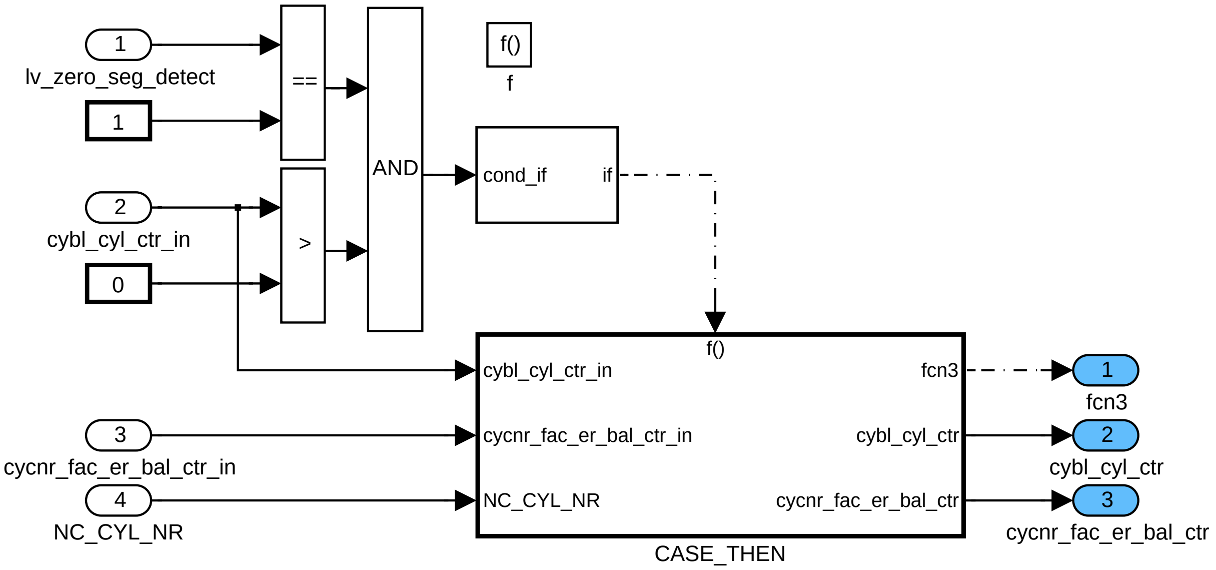


Figure 139 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ CLC_CYCNR/ ACTIVATION

Calculation of segment tasks CLC_OPERATE_SEG CLC_MPL_TI_COR_VALUE CLC_CYCNR ACTIVATION CASE_THEN

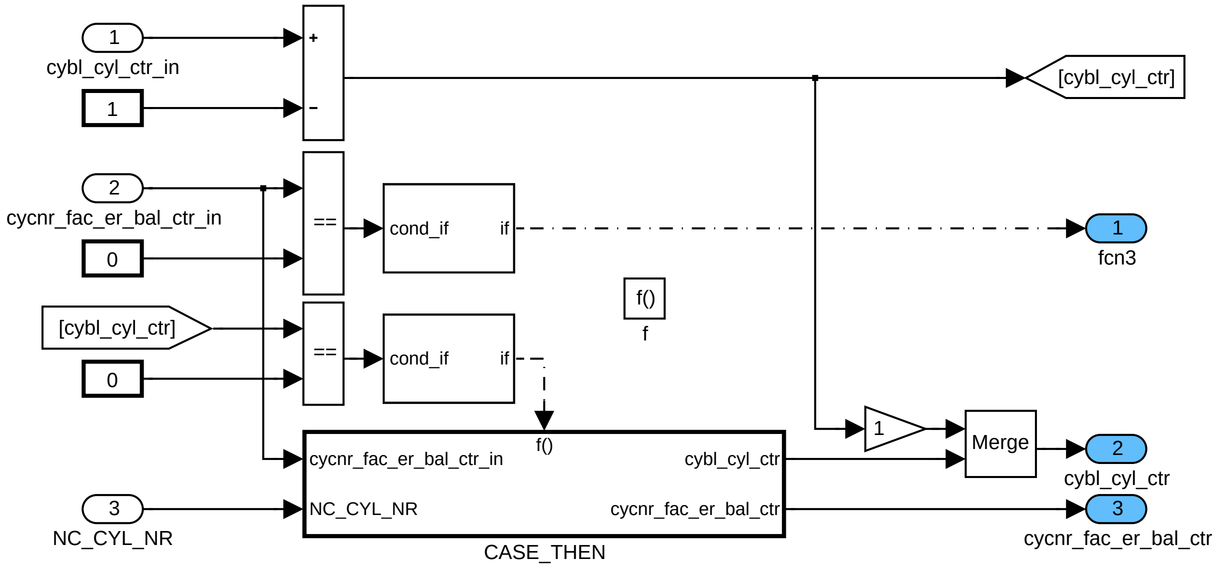


Figure 140 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ CLC_CYCNR/ ACTIVATION/ CASE_THEN

Calculation of segment tasks CLC_OPERATE_SEG CLC_MPL_TI_COR_VALUE CLC_CYCNR ACTIVATION CASE_THEN CASE_THEN

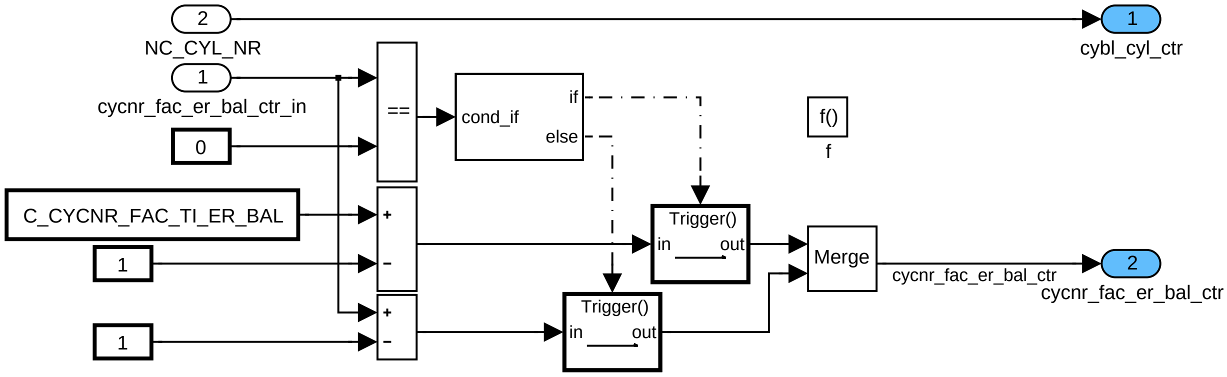


Figure 141 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ CLC_CYCNR/ ACTIVATION/ CASE_THEN/ CASE_THEN

Calculation of segment tasks CLC_OPERATE_SEG CLC_MPL_TI_COR_VALUE CLC_CYCNR INITIALIZATION

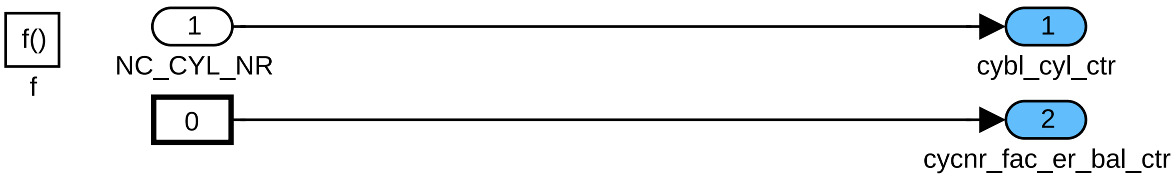


Figure 142 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ CLC_CYCNR/ INITIALIZATION

Controller Equations

For the multiple correction mode, a separate set of NC_CYL_NR controllers exists operating in parallel (one controller for each cylinder).

The computation of the output values is carried out every C_CYCNR_FAC_TI_ER_BAL engine cycles (= calibrateable number of engine cycles).

Only in case of C_CYCNR_FAC_TI_ER_BAL = 1, the calculation of the additive path is done without interruption. At every segment the corresponding cylinder correction value is built.

If the computation is not done at every engine cycle (C_CYCNR_FAC_TI_ER_BAL > 1), the trigger of the controller output calculation is always related to SEG_NR_ER = 0.

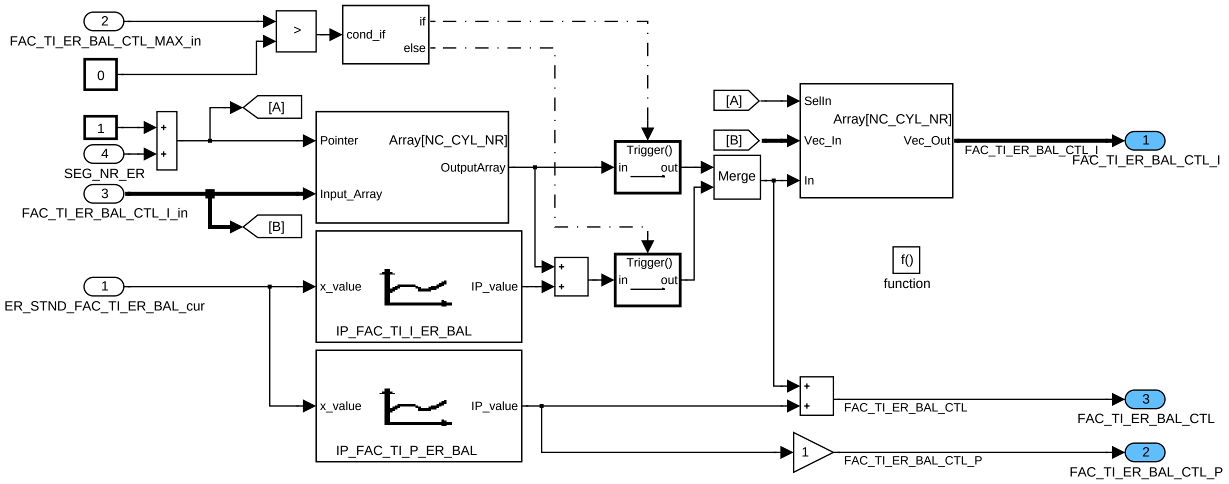


Figure 143 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ CTL_EQUATIONS

Limitation of the Controller Output

The outputs of the controllers are limited to calibrateable threshold. If one controller output has reached the limit, a flag is set at one and the corresponding segment number is displayed.

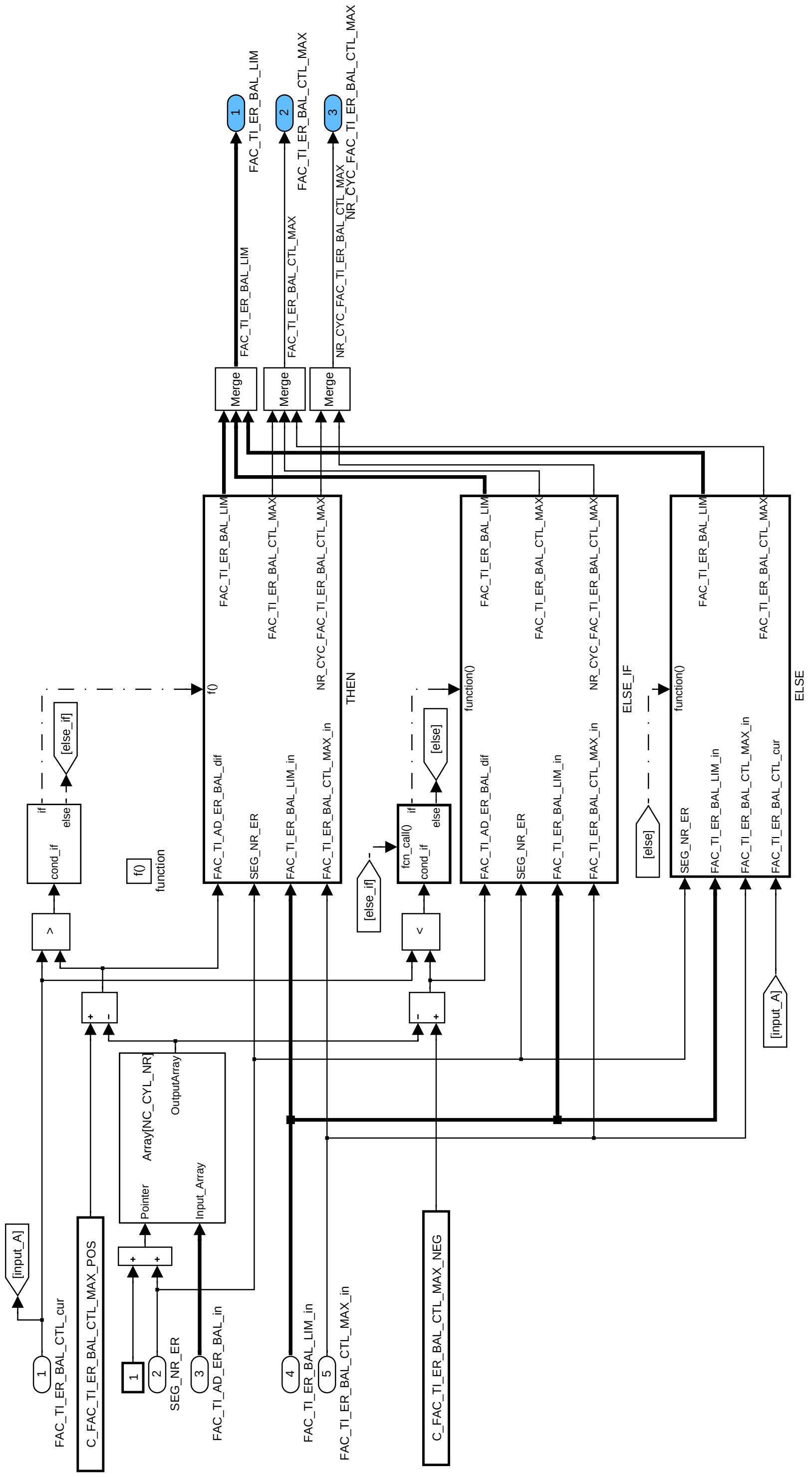


Figure 144 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ LIM_CTL_OUT

Limitation of the Controller Output – then branch

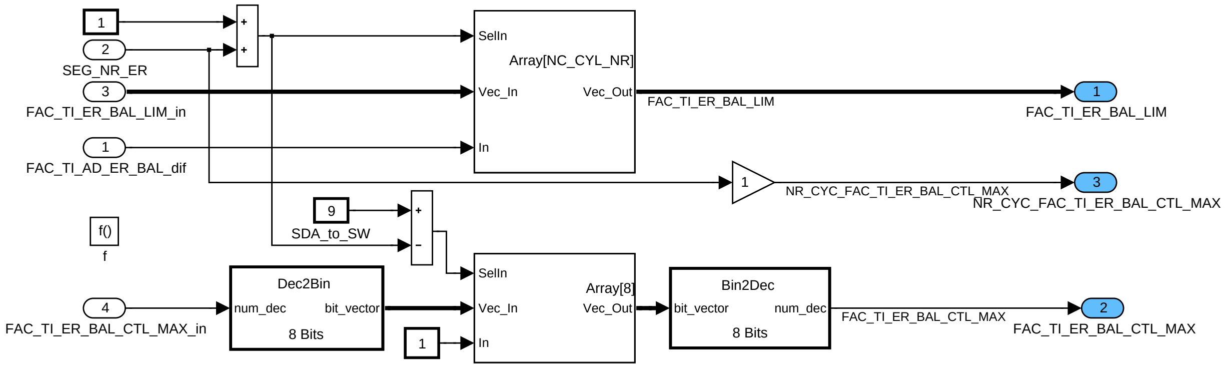


Figure 145 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ LIM_CTL_OUT/ THEN

Limitation of the Controller Output – else if branch

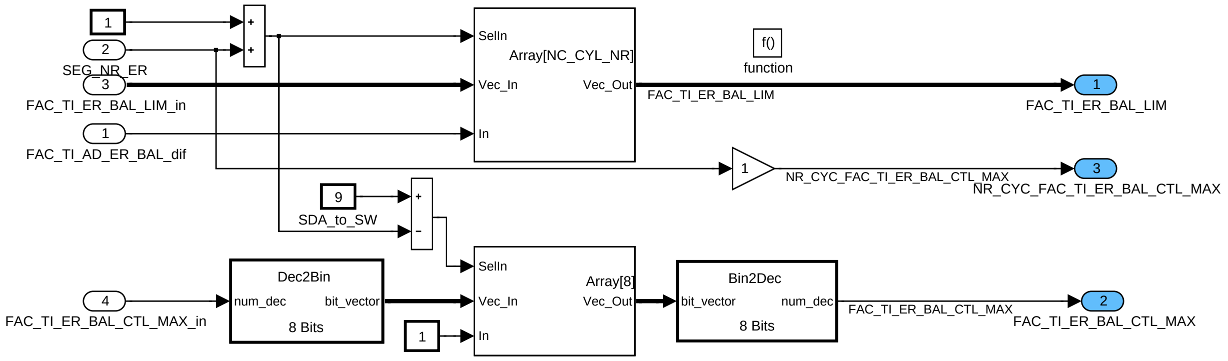


Figure 146 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ LIM_CTL_OUT/ ELSE_IF

Limitation of the Controller Output – else branch

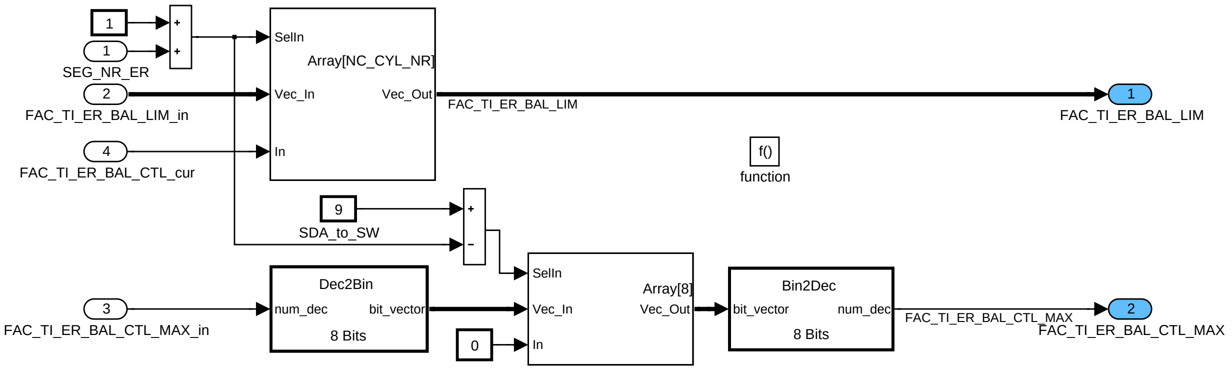


Figure 147 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ LIM_CTL_OUT/ ELSE

Correction of limited controller output values with adaptation values

The limited controller output values are corrected with the currently available adaptive correction values, which is saved in NVMY and updated at Activation or Deactivation of the multiple adaptation path (LV_FAC_TI_AD_ER_BAL_ENA)

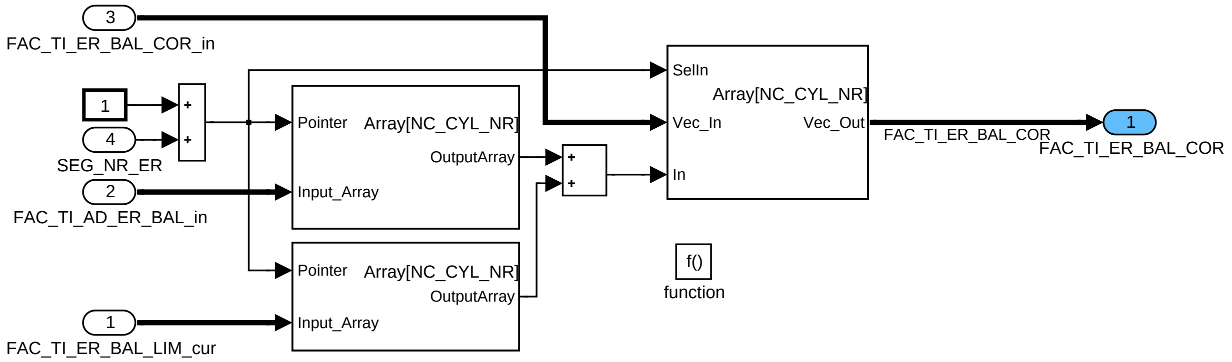


Figure 148 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ COR_LIM_CTL_OUT_VALUE

<u>Observation of corrected controller output values for OBD issues</u>

The multiple correction values out of the correction path are observed for OBD purposes. As soon as the minimum or maximum diagnosis threshold is exceeded, a corresponding flag for the related cylinder is set. In this case the OBD system is indicated to handle the further failure treatment. The diagnosis thresholds have always to be less or equal to the absolute minimum or maximum controller output limits.

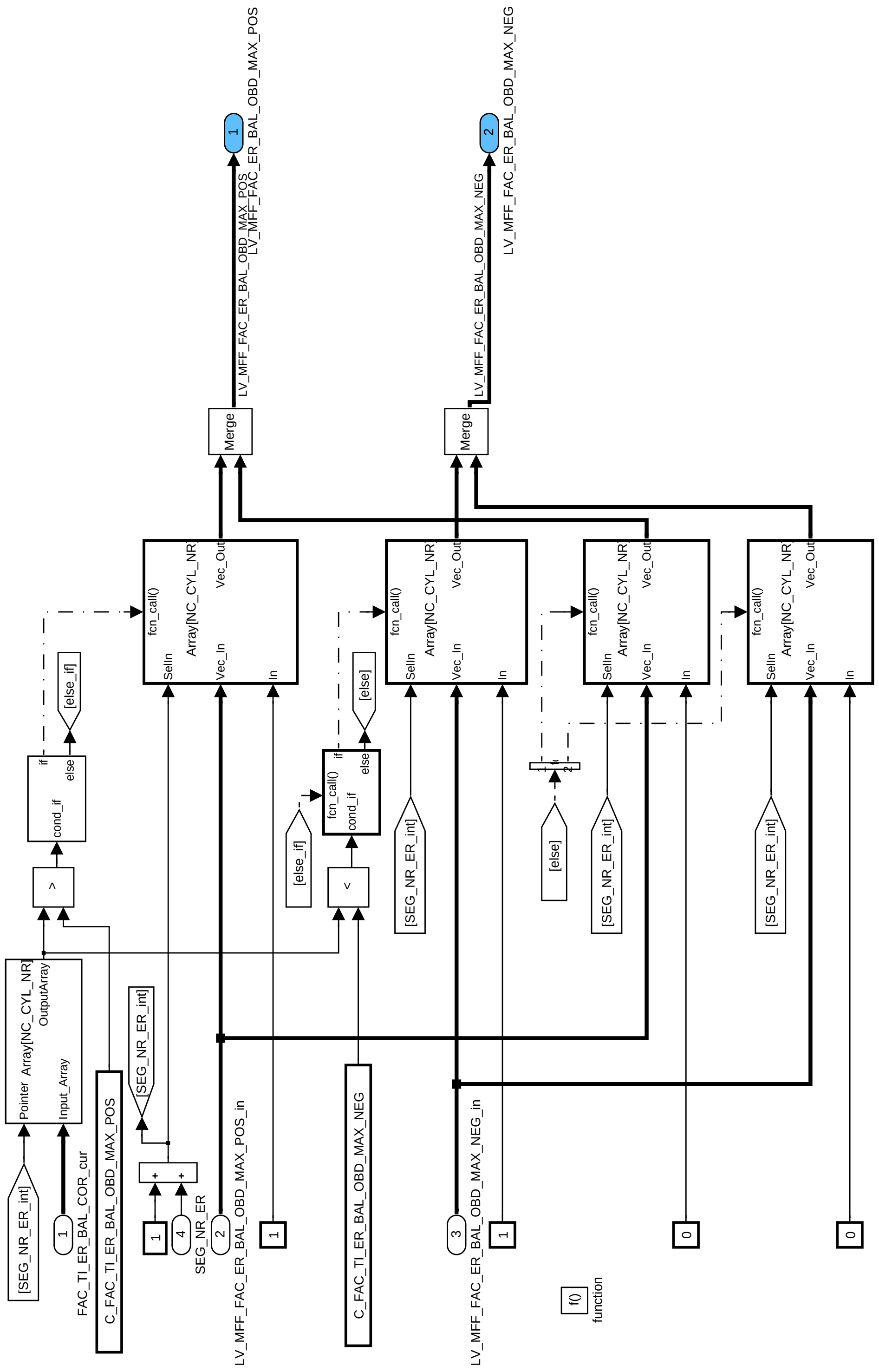


Figure 149 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_COR_VALUE/ COR_CTL_OUT_VALUE_OBD

Calculation of the multiple injection time adaptation values

The corrected NC_CYL_NR controller output values are adjusted dependent on the corresponding segment number SEG_NR_ER. During the function is activated, at every new segment the corresponding FAC_TI_ER_BAL value is updated. Due to this correction the sum of all adaptation output values remains zero to guarantee that the overall injection spreading stays unchanged.

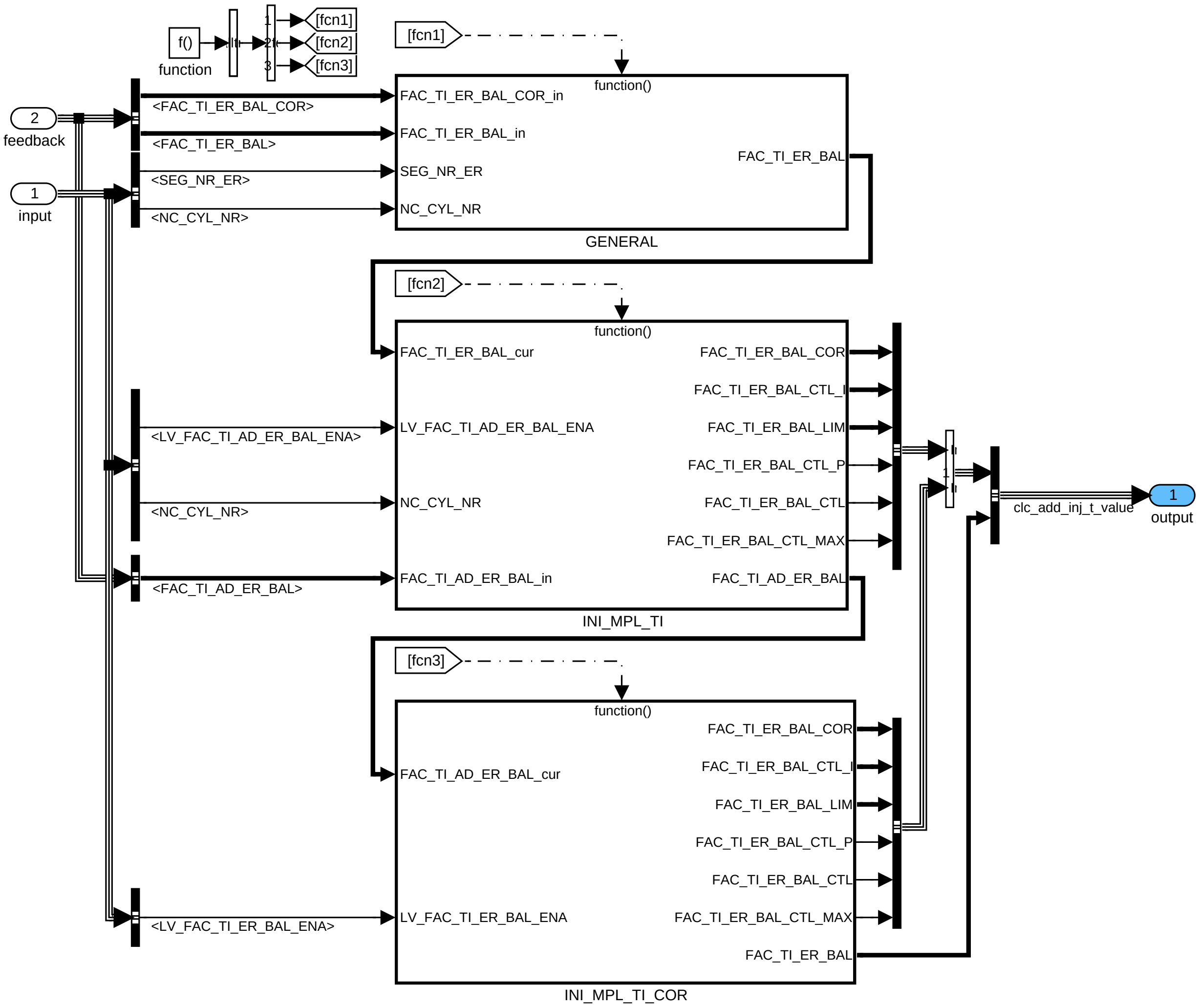


Figure 150 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_VALUE

Calculation of the multiple injection time adaptation values - General

Sum of all NC_CYL_NR controller output values / number of cylinders

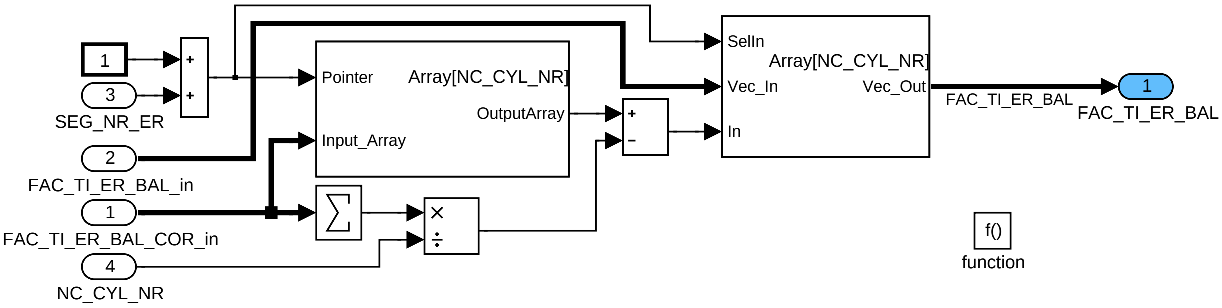


Figure 151 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_VALUE/ GENERAL

Initialization at Entry or Exit to multiple injection time adaptation

As soon as the multiple injection time adaptation is enabled or disabled, the stored adaptive correction values are overwritten with the currently present global adaptive value and the controller output values are set to zero.

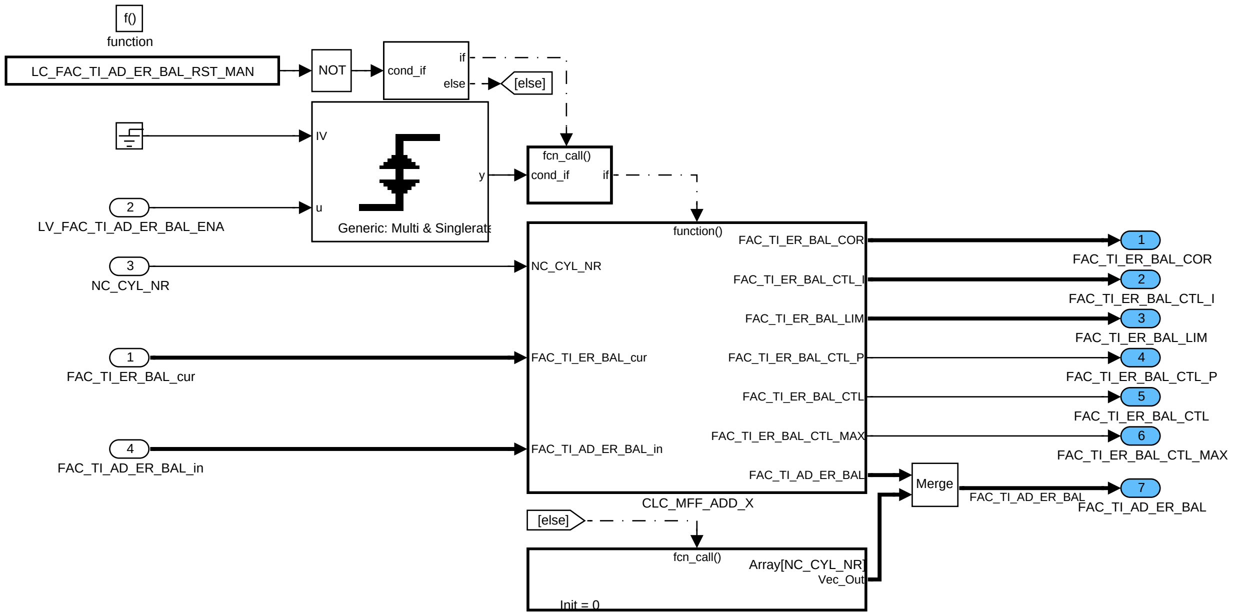


Figure 152 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_VALUE/ INI_MPL_TI

Initialization at Entry or Exit to multiple injection time adaptation – then – then branch

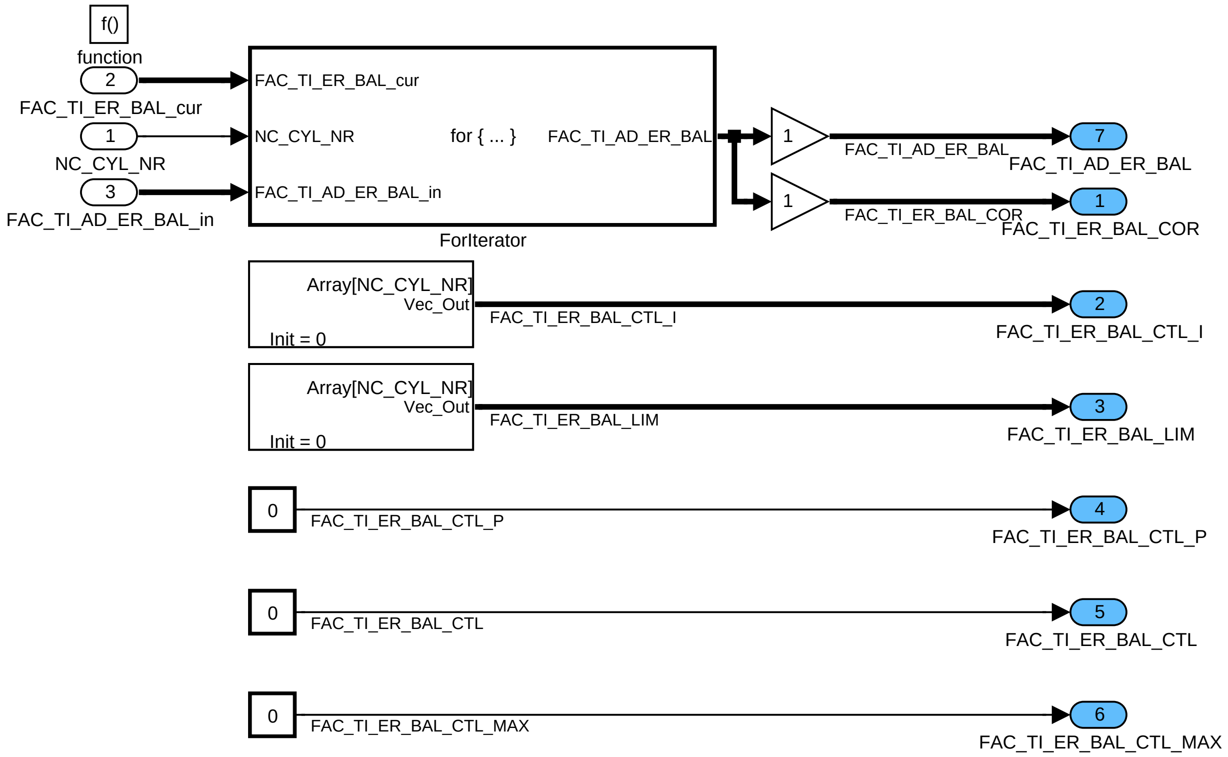


Figure 153 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_VALUE/ INI_MPL_TI/ CLC_MFF_ADD_X

Initialization at Entry or Exit to multiple injection time adaptation – then – then – for iterator branch

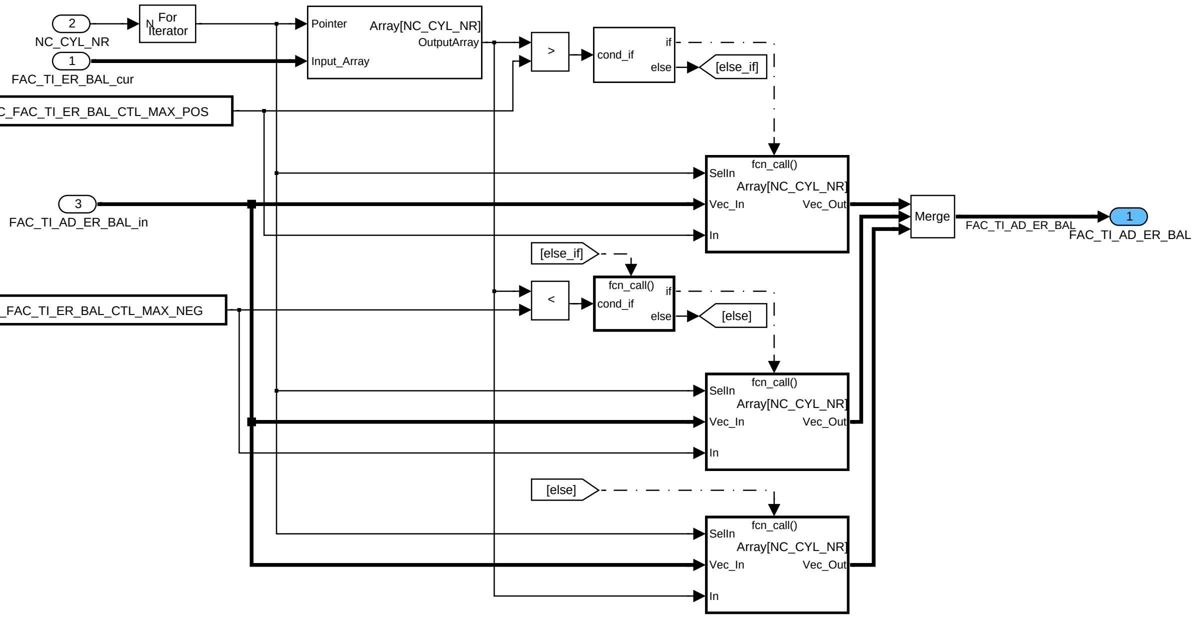


Figure 154 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_VALUE/ INI_MPL_TI/ CLC_MFF_ADD_X/ ForIterator

Initialization at Exit from multiple injection time correction

As soon as the multiple injection time correction is disabled in combination with the manual switch to force the overall correction (LC_FAC_TI_ER_BAL_COR), the controller output values are set to zero.

The initialization only takes place, if the logical constant for a manual reset of the multiple controller values is set to zero (LC_FAC_TI_ER_BAL_RST_MAN =0).

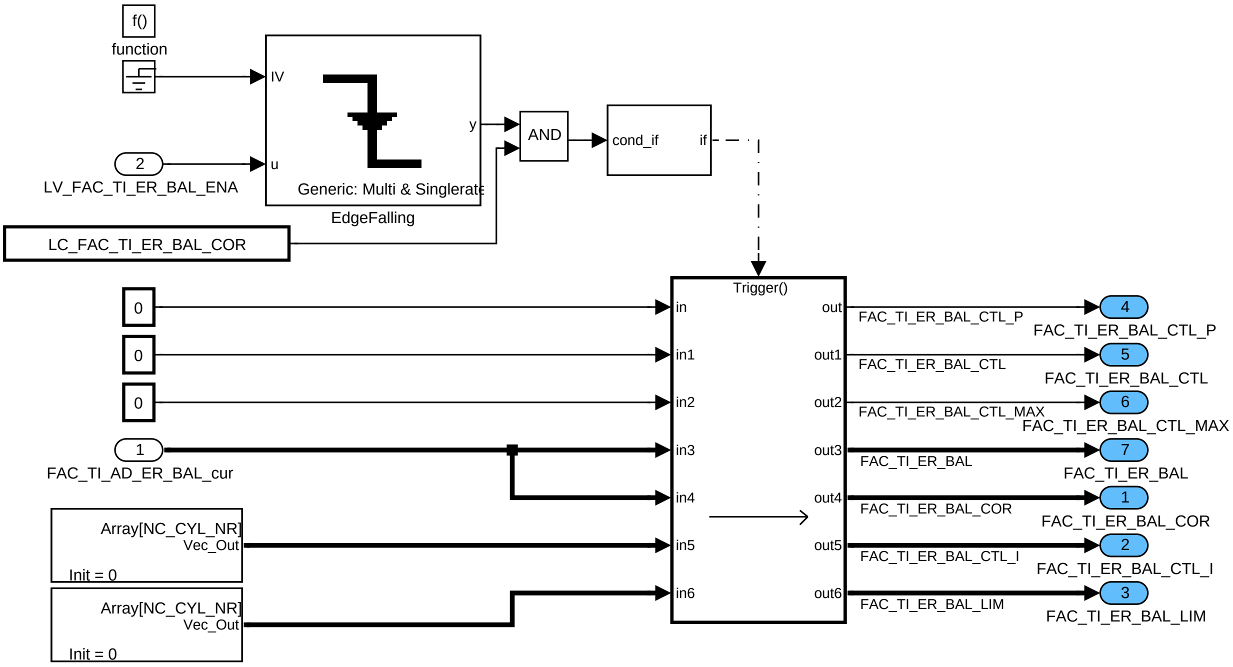


Figure 155 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_VALUE/ INI_MPL_TI_COR

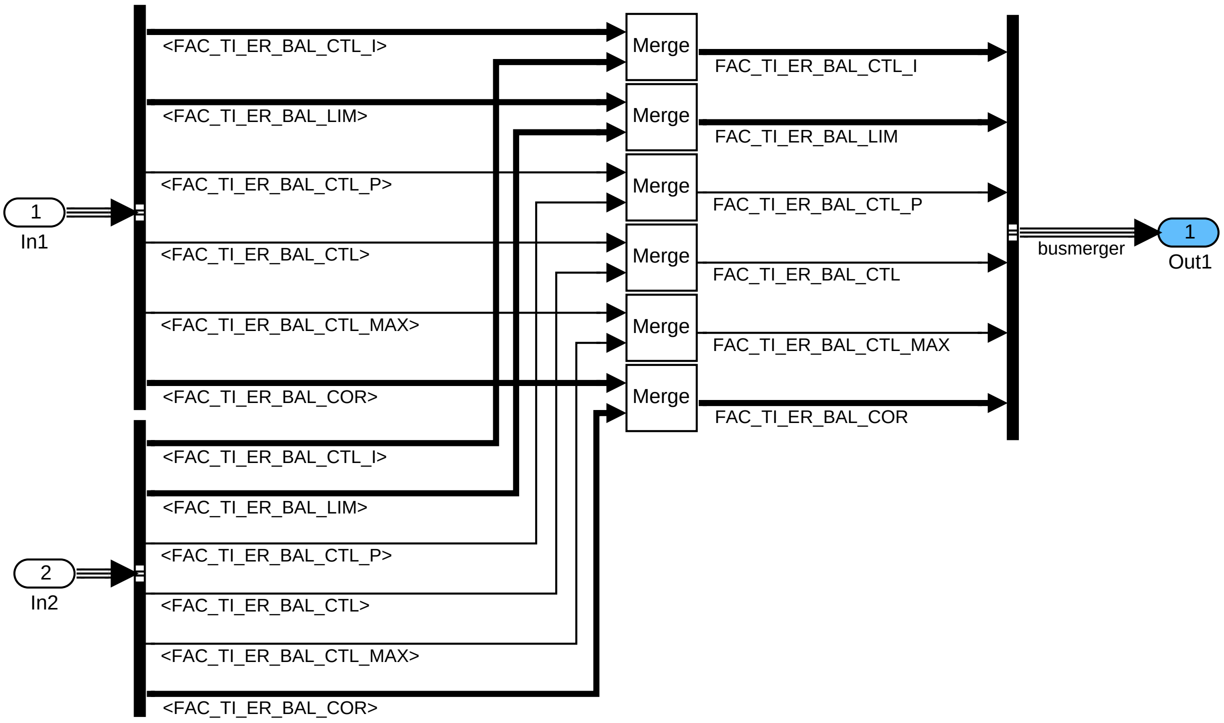


Figure 156 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ CLC_MPL_TI_VALUE/ BusMerger

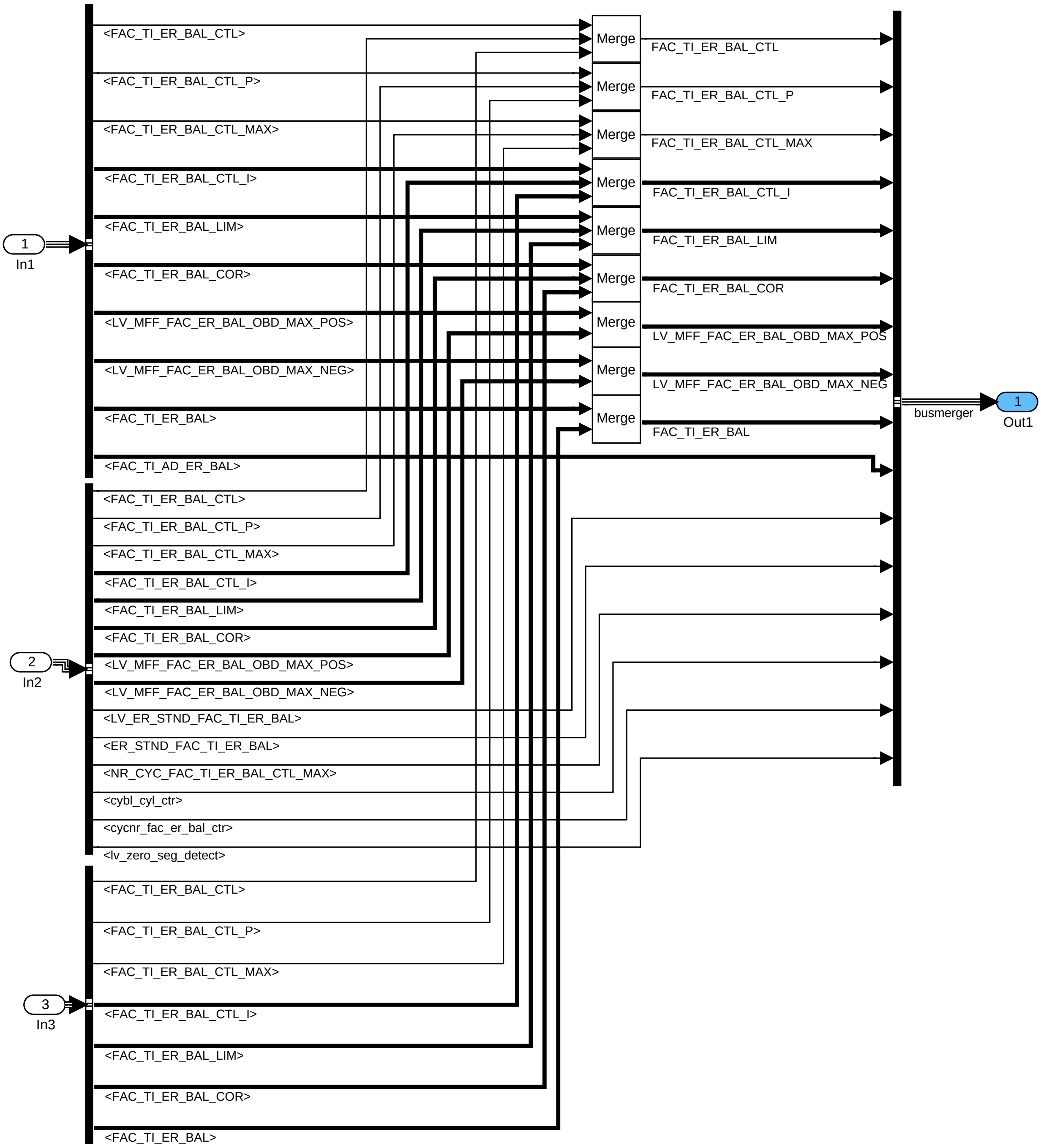


Figure 157 CYBL_M30707Q01/ OPM/ operate_SEG/ CLC_OPERATE_SEG/ BusMerger

6.6.1.3 SUBFUNCTION: SIG_MNG

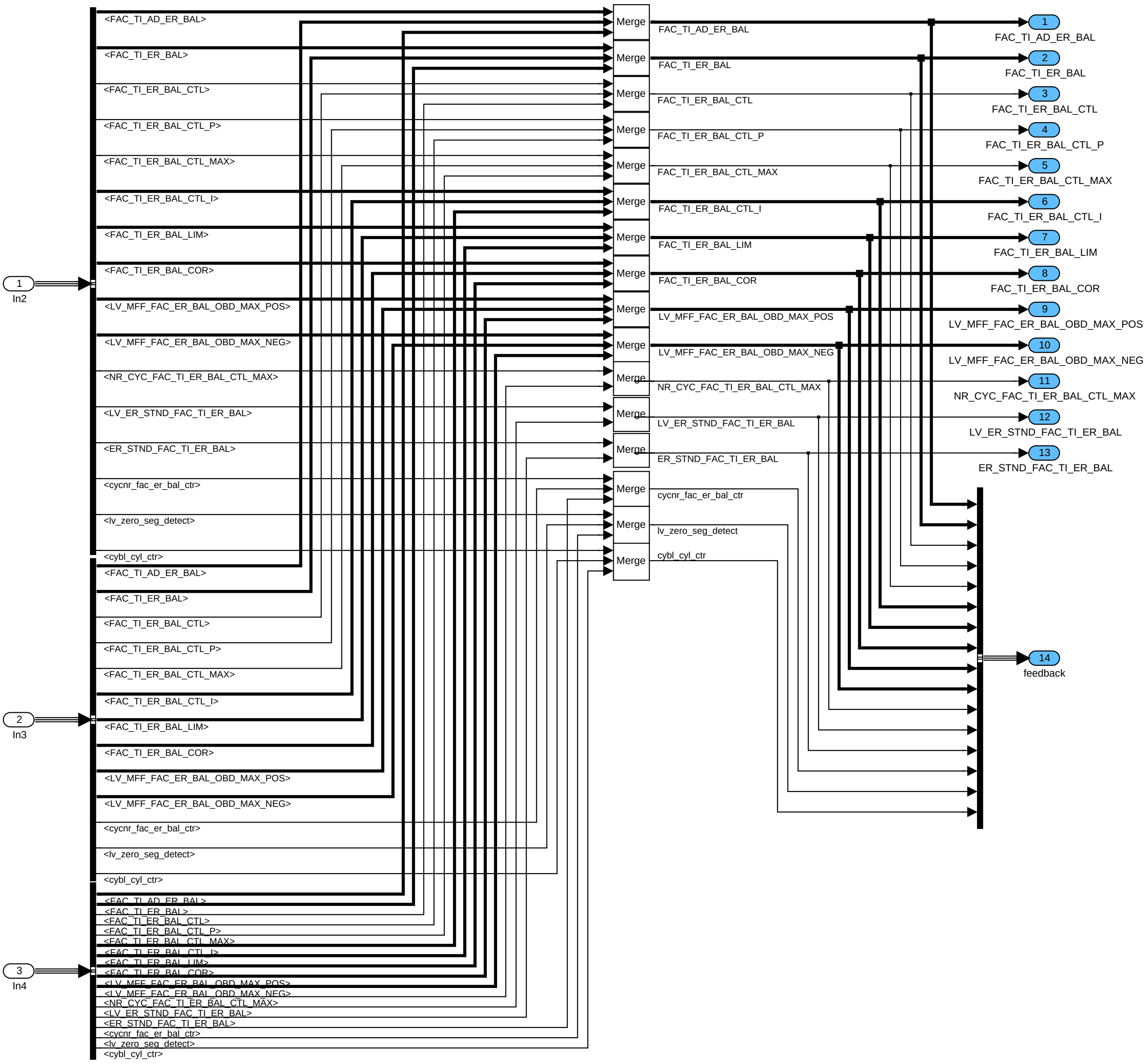


Figure 158 CYBL_M30707Q01/ SIG_MNG

6.7 Cylinder Balancing Manager

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_TI_ER_BAL	O/V	0H 1H 2H 3H	INITIALIZATION LOCK WAIT CYLINDER_BALANCING	1	-
State of cylinder balancing via TI intervention					
STATE_CTL_TI_ER_BAL	O/V	0H 1H 2H 3H 4H	NON ADDITIVE ADDITIVE_AD MULTIPLE MULTIPLE_AD	1	-
State of cylinder balancing via TI intervention					
LV_MFF_ADD_ER_BAL_ENA	O/V	0...1H	0...1	1	-
Flag for additive cylinder balancing MFF correction active					
LV_MFF_ADD_AD_ER_BAL_ENA	O/V	0...1H	0...1	1	-
Flag for additive cylinder balancing MFF adaptation active					
LV_FAC_TI_ER_BAL_ENA	O/V	0...1H	0...1	1	-
Flag for multiple cylinder balancing TI correction active					
LV_FAC_TI_AD_ER_BAL_ENA	O/V	0...1H	0...1	1	-
Flag for multiple cylinder balancing TI adaptation active					
LV_ER_STND_MMV_MAX_TI_ER_BAL	V	0...1H	0...1	1	-
Flag for maximum ER_STND_MMV_BAL value for cylinder balancing via TI intervention					
LV_ER_STD_MMV_MAX_TI_ER_BAL	V	0...1H	0...1	1	-
Flag for maximum ER_STD_MMV_BAL value for cylinder balancing via TI intervention					
LV_TI_ER_BAL_ENA	V	0...1H	0...1	1	-
Flag for cylinder balancing via TI intervention enabled					
CTR_TI_ER_BAL_ENA	V	0...FFFFH	0...6.5535E+4	1	-
Engine cycle counter to delay TI intervention at entry to state cylinder balancing					

Input data:

ER_STND_MMV_BAL[NC_CYL_NR]	ER_STD_MMV_BAL[NC_CYL_NR]	LV_DRV1_STND_BAL_FDOUT	LC_TI_ER_BAL_STOP_MAN
LV_MFF_COR_ER_BAL_ENA_EXT	LV_MFF_ADD_AD_ER_BAL_ENA_EXT	LV_MFF_FAC_AD_ER_BAL_ENA_EXT	TCO
MFF_SP_S	LV_IS	N_32	SEG_NR_ER
NC_CYL_NR			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ER_STD_MMV_MAX_TI_ER_BAL	1	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Maximum ER_STD_MMV_BAL value for cylinder balancing via TI intervention					
C_ER_STND_MMV_MAX_TI_ER_BAL	1	8000...7FFFH	-325.78...325.77	0.00994202	1/s²
Maximum ER_STND_MMV_BAL value for cylinder balancing via TI intervention					
C_MFF_SP_MAX_FAC_TI_AD_ER_BAL	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Maximum mass fuel flow setpoint for cylinder balancing multiple TI adaptation					
C_MFF_SP_MAX_FAC_TI_ER_BAL	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Maximum mass fuel flow setpoint for cylinder balancing multiple TI correction					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MFF_SP_MAX_MFF_ADD_AD_ER_BAL	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Maximum mass fuel flow setpoint for cylinder balancing additive MFF adaptation					
C_MFF_SP_MAX_MFF_ADD_ER_BAL	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Maximum mass fuel flow setpoint for cylinder balancing additive MFF correction					
C_MFF_SP_MIN_FAC_TI_AD_ER_BAL	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Minimum mass fuel flow setpoint for cylinder balancing multiple TI adaptation					
C_MFF_SP_MIN_FAC_TI_ER_BAL	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Minimum mass fuel flow setpoint for cylinder balancing multiple TI correction					
C_MFF_SP_MIN_MFF_ADD_AD_ER_BAL	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Minimum mass fuel flow setpoint for cylinder balancing additive MFF adaptation					
C_MFF_SP_MIN_MFF_ADD_ER_BAL	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Minimum mass fuel flow setpoint for cylinder balancing additive MFF correction					
C_N_32_MAX_FAC_TI_AD_ER_BAL	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for cylinder balancing multiple TI adaptation					
C_N_32_MAX_FAC_TI_ER_BAL	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for cylinder balancing multiple TI correction					
C_N_32_MAX_MFF_ADD_AD_ER_BAL	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for cylinder balancing additive MFF adaptation					
C_N_32_MAX_MFF_ADD_ER_BAL	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for cylinder balancing additive MFF correction					
C_N_32_MIN_FAC_TI_AD_ER_BAL	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for cylinder balancing multiple TI adaptation					
C_N_32_MIN_FAC_TI_ER_BAL	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for cylinder balancing multiple TI correction					
C_N_32_MIN_MFF_ADD_AD_ER_BAL	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for cylinder balancing additive MFF adaptation					
C_N_32_MIN_MFF_ADD_ER_BAL	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for cylinder balancing additive MFF correction					
C_TCO_MAX_TI_ER_BAL	1	0...FEH	-48...142.5	0.75	°C
Maximum coolant temperature value for cylinder balancing via TI intervention					
C_TCO_MIN_FAC_TI_AD_ER_BAL	1	0...FEH	-48...142.5	0.75	°C
Minimum coolant temperature value for cylinder balancing multiple TI adaptation					
C_TCO_MIN_MFF_ADD_AD_ER_BAL	1	0...FEH	-48...142.5	0.75	°C
Minimum coolant temperature value for cylinder balancing additive MFF adaptation					
C_TCO_MIN_TI_ER_BAL	1	0...FEH	-48...142.5	0.75	°C
Minimum coolant temperature value for cylinder balancing via TI intervention					
LC_MFF_ADD_ER_BAL_ENA	1	0...1H	0...1	1	-
Logical constant to enable additive MFF intervention out of idle					
IP_CTR_TI_ER_BAL_ENA_INI	4	0...FFFFH	0...6.5535E+4	1	-
LDP_N_32_IP_CTR_TI_ER_BAL_INI	4	0...FFH	0...8.16E+3	32	rpm
Engine cycle counter start value to delay TI intervention at entry to state cylinder balancing					

6.7.1 General information:

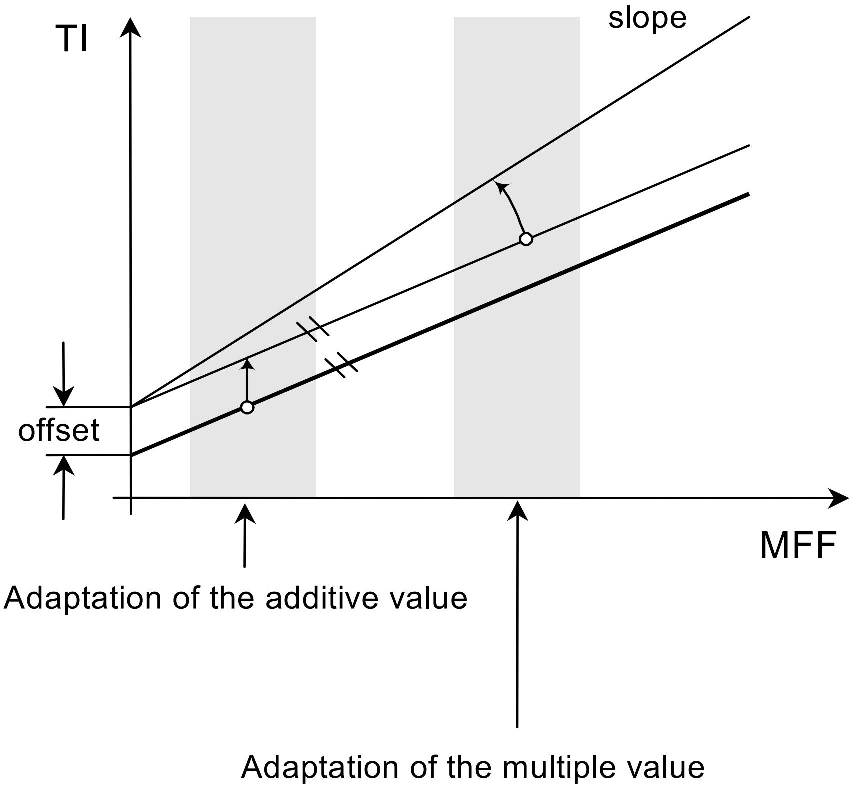
Cylinder balancing via TI intervention means a correction of the cylinder individual injection time values to achieve a balancing of the indicated torque between the single cylinders based on engine roughness calculation. Target is to achieve a smooth engine running.

The "Balancing Manager" is realized with a state machine. Different states with a defined setting of corresponding output variables are existing as well as clear conditions to change from one state to another. The calculation of adaptation values is controlled by the states of the state machine.

By means of the logical constant LC_MFF_ADD_ER_BAL_ENA it is possible to enable (LC_MFF_ADD_ER_BAL_ENA = 1) or disable (LC_MFF_ADD_ER_BAL_ENA = 0) the additive adaptation in part load. At idle speed the additive adaptation is always enabled.

In order to compensate torque differences, e.g. caused by manufacturing tolerances of the injection valves, either an additive or a multiple adaptation correction of the injection times is carried out at state "Cylinder Balancing" depending on the current mass fuel flow setpoint.

The additive adaptation value, which influences the offset of the injection valve characteristic line is learned at low mass fuel flow values. The multiple adaptation factor however, which influences the slop of the injector characteristic line, is learned at higher mass fuel flow values (refer: figure).



Application Condition

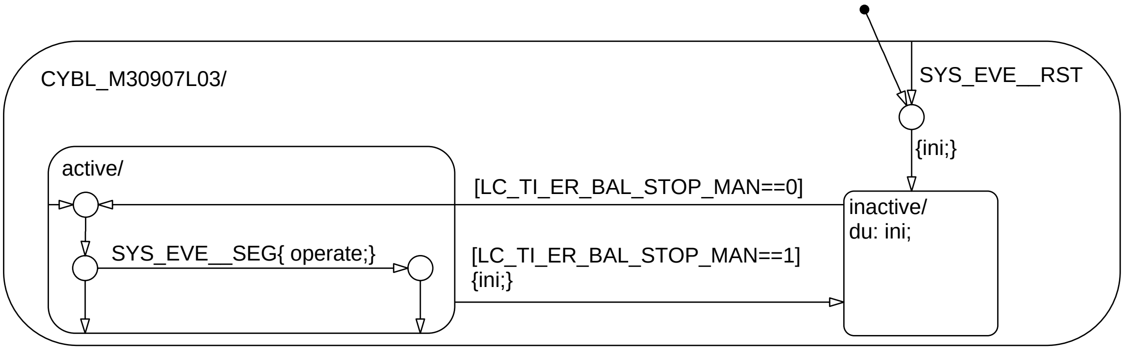


Figure 159 CYBL_M30907L03/ APP_CDN/ Chart

Function Description

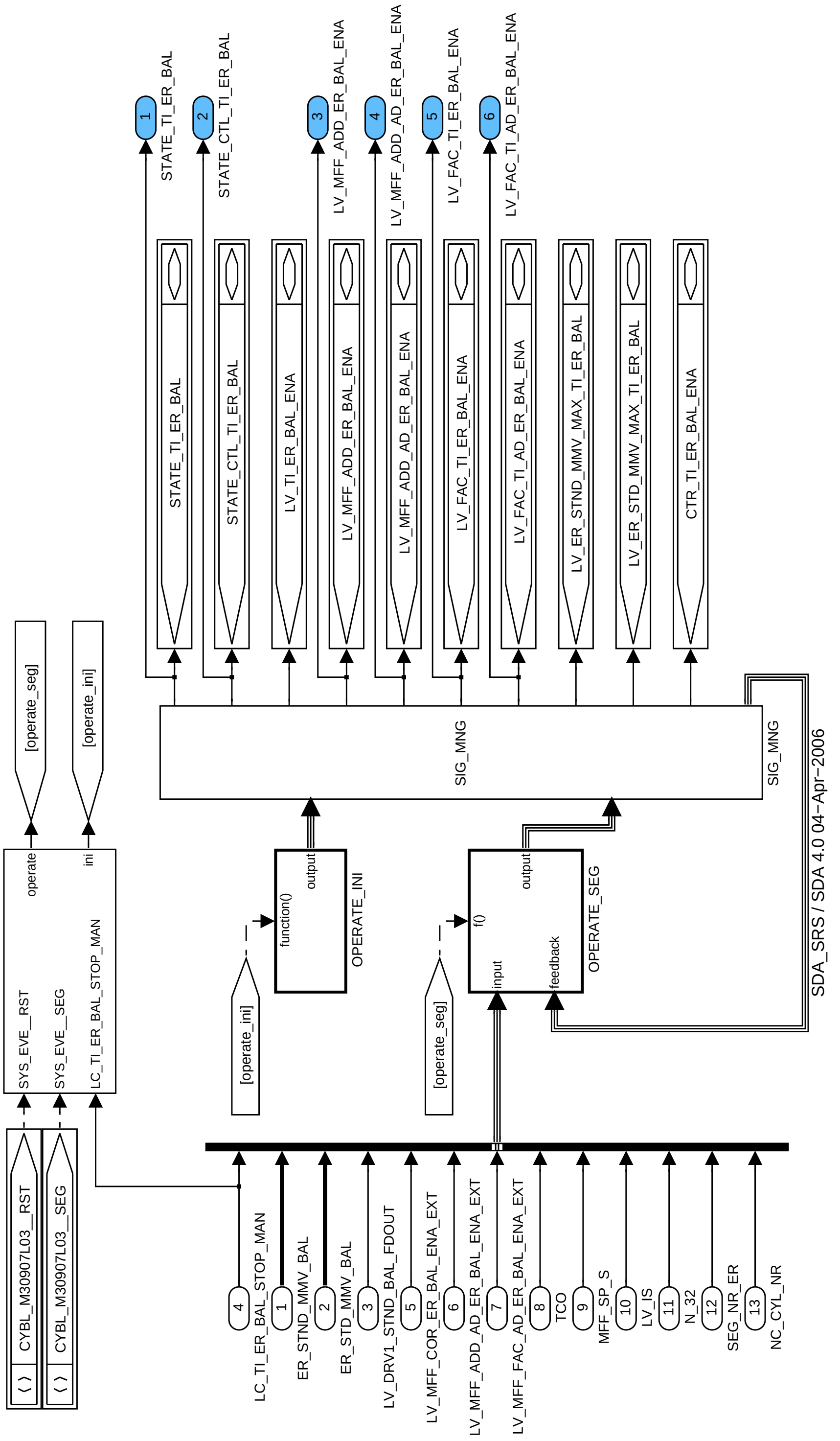


Figure 160 CYBL_M30907L03

6.7.1.1 Initialization of variables

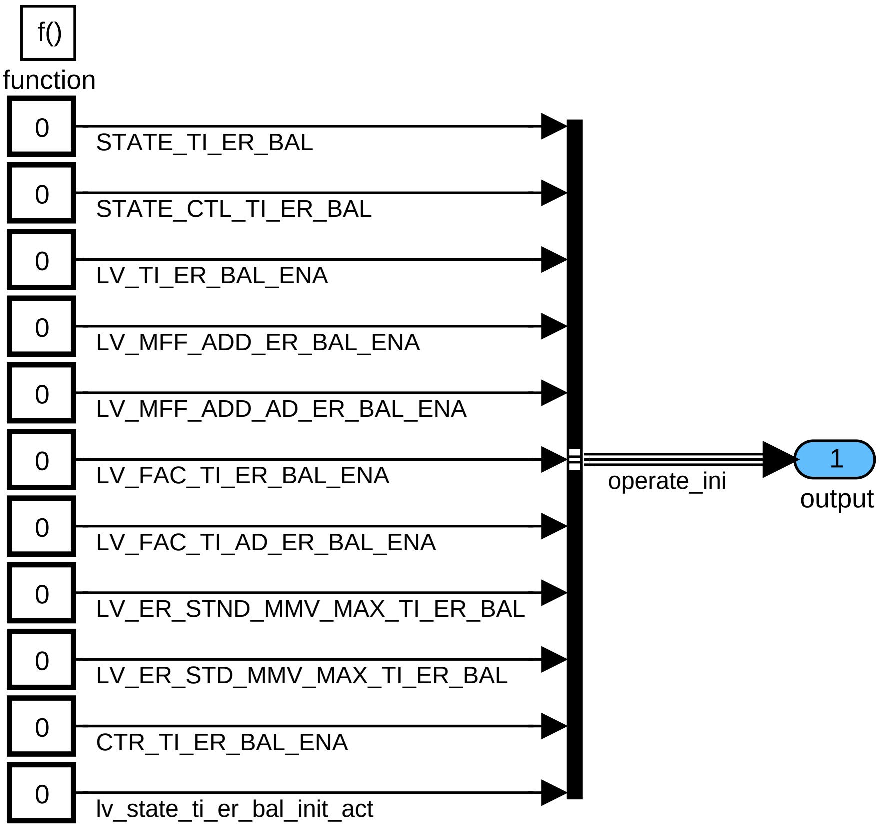


Figure 161 CYBL_M30907L03/ OPERATE_INI

6.7.1.2 Calculation of segment task

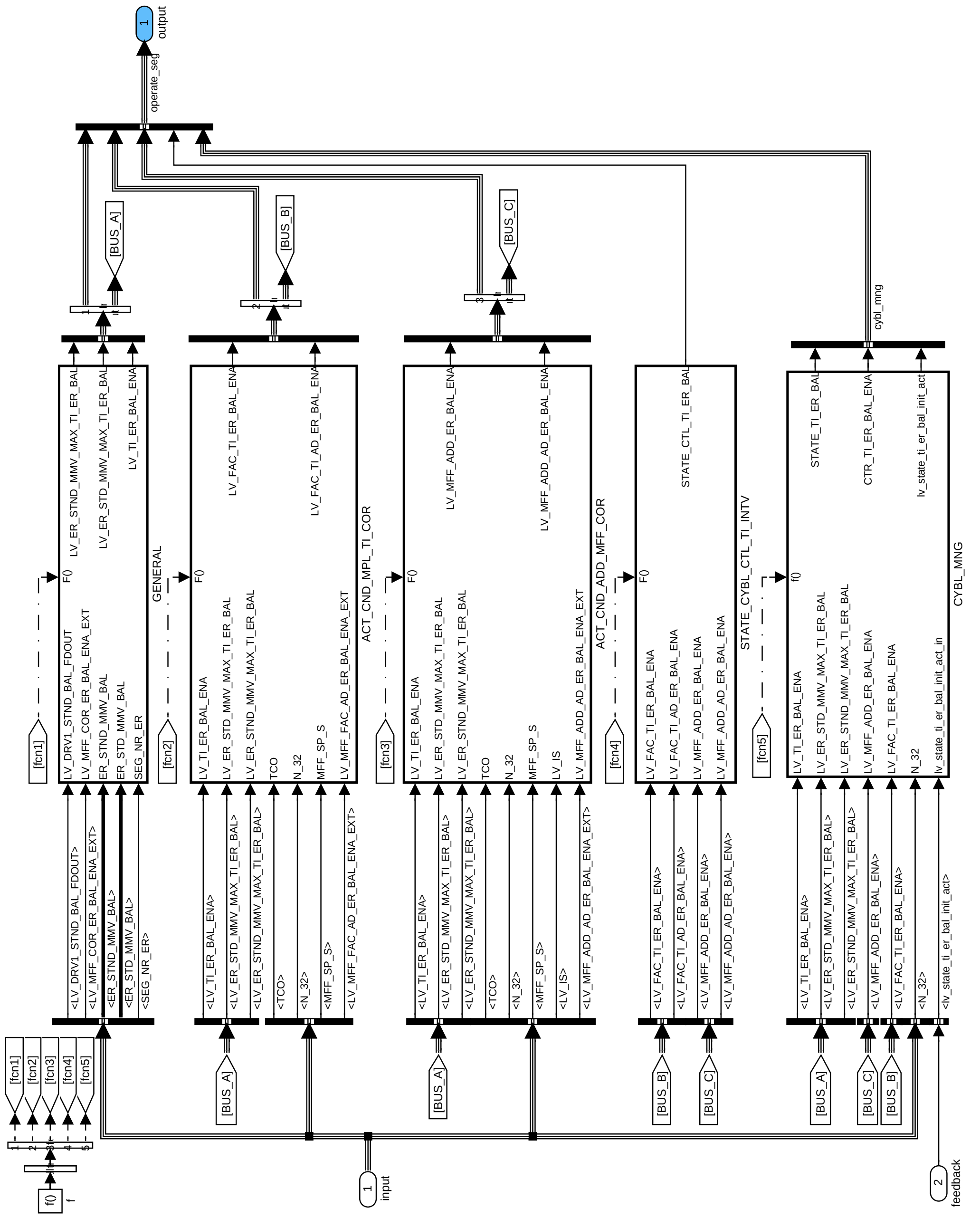


Figure 162 CYBL_M30907L03/ OPERATE_SEG

Calculation of general tasks

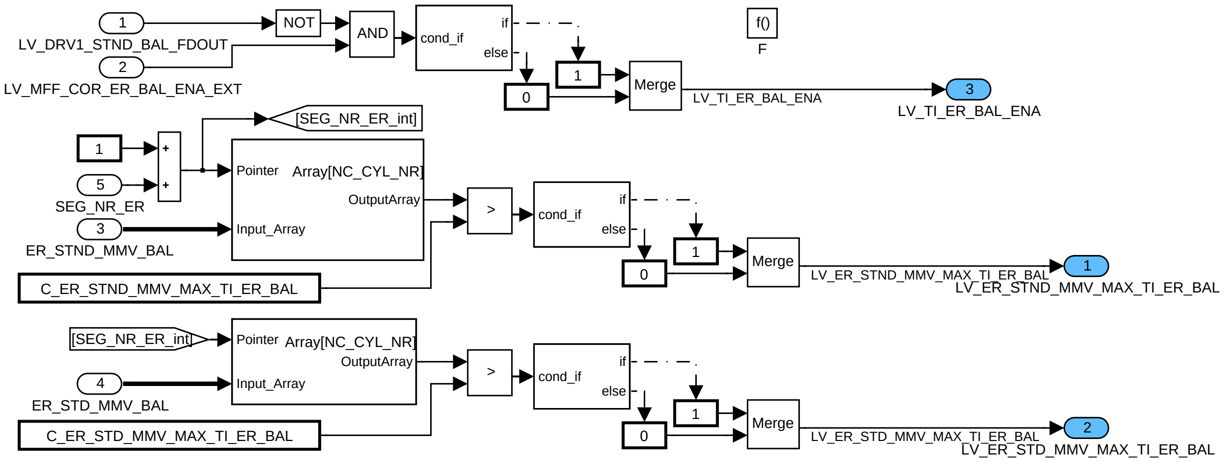


Figure 163 CYBL_M30907L03/ OPERATE_SEG/ GENERAL

Action condition for multiple cylinder balancing TI correction:

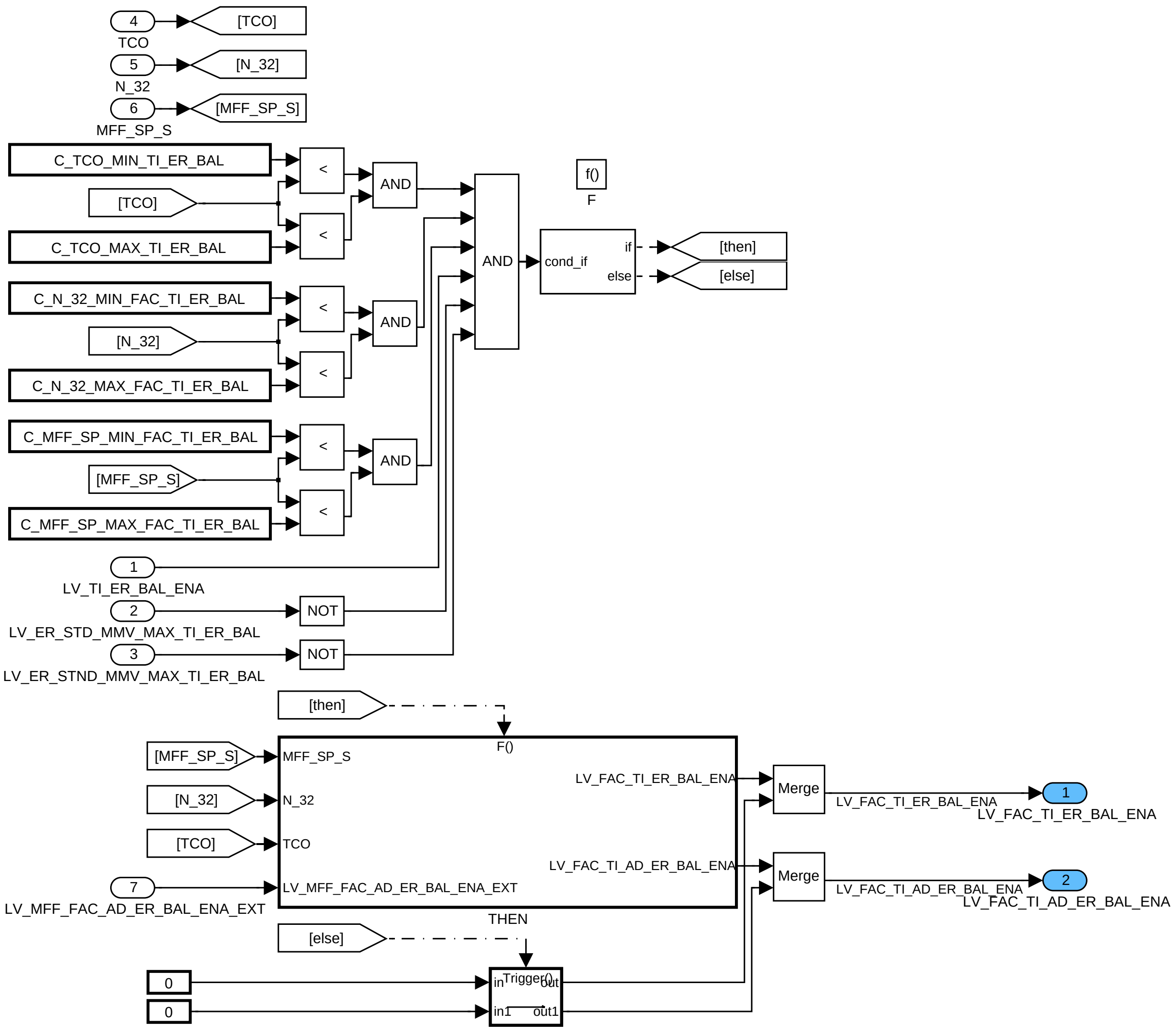


Figure 164 CYBL_M30907L03/ OPERATE_SEG/ ACT_CND_MPL_TI_COR

Action condition for multiple cylinder balancing TI correction – THEN-branch

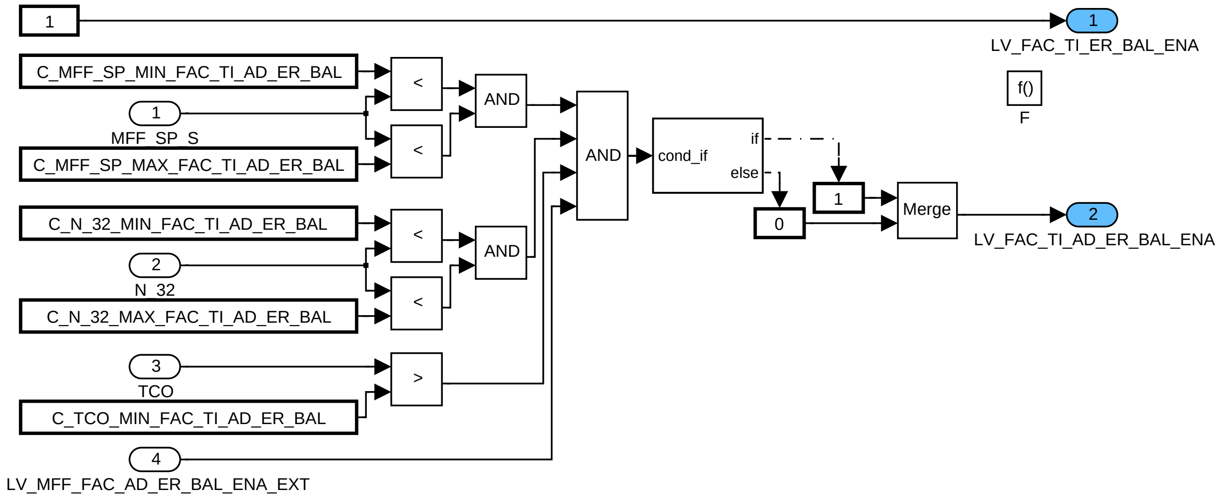


Figure 165 CYBL_M30907L03/ OPERATE_SEG/ ACT_CND_MPL_TI_COR/ THEN

Action condition for additive cylinder balancing MFF correction:

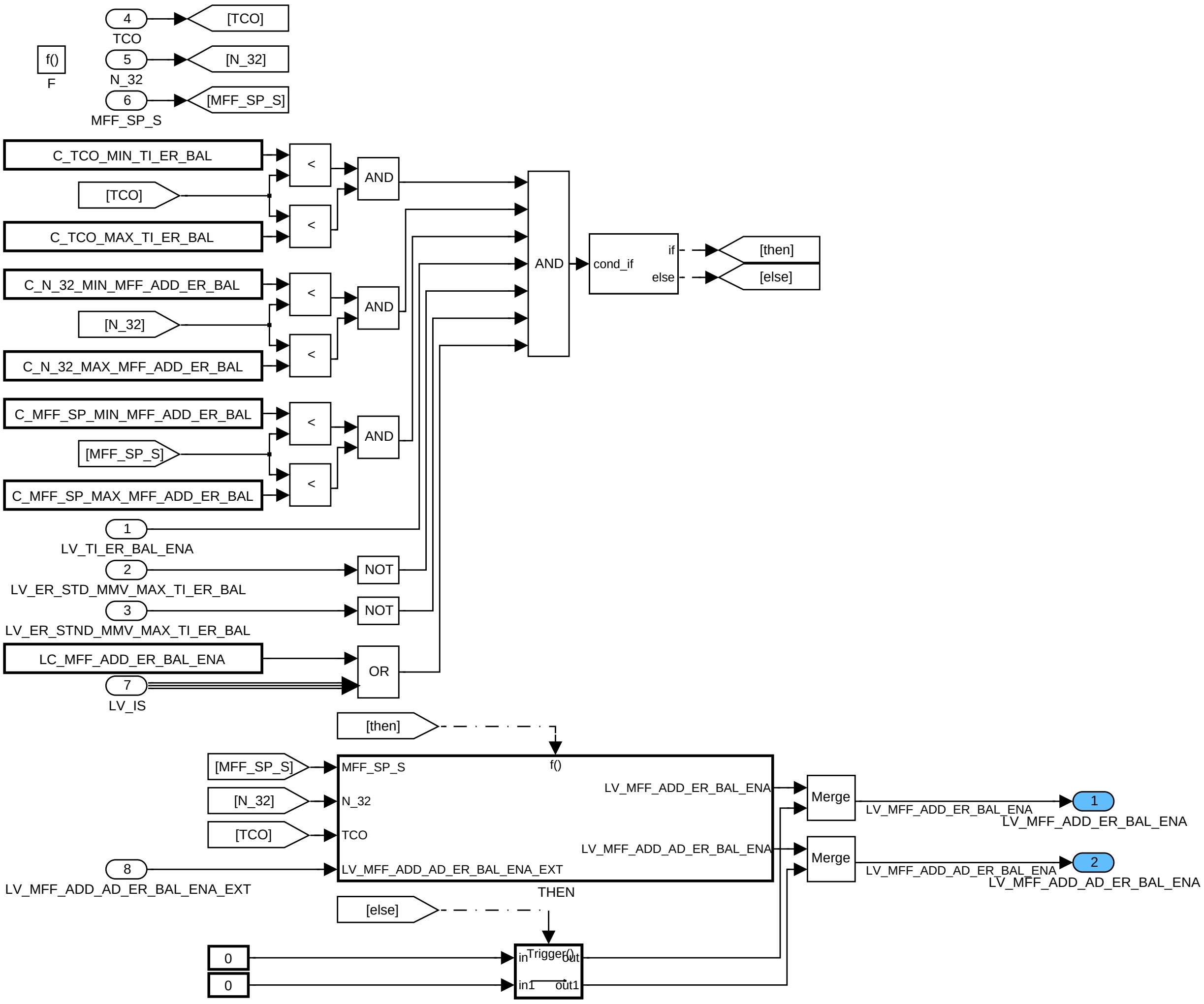


Figure 166 CYBL_M30907L03/ OPERATE_SEG/ ACT_CND_ADD_MFF_COR

Action condition for additive cylinder balancing MFF correction - THEN-branch

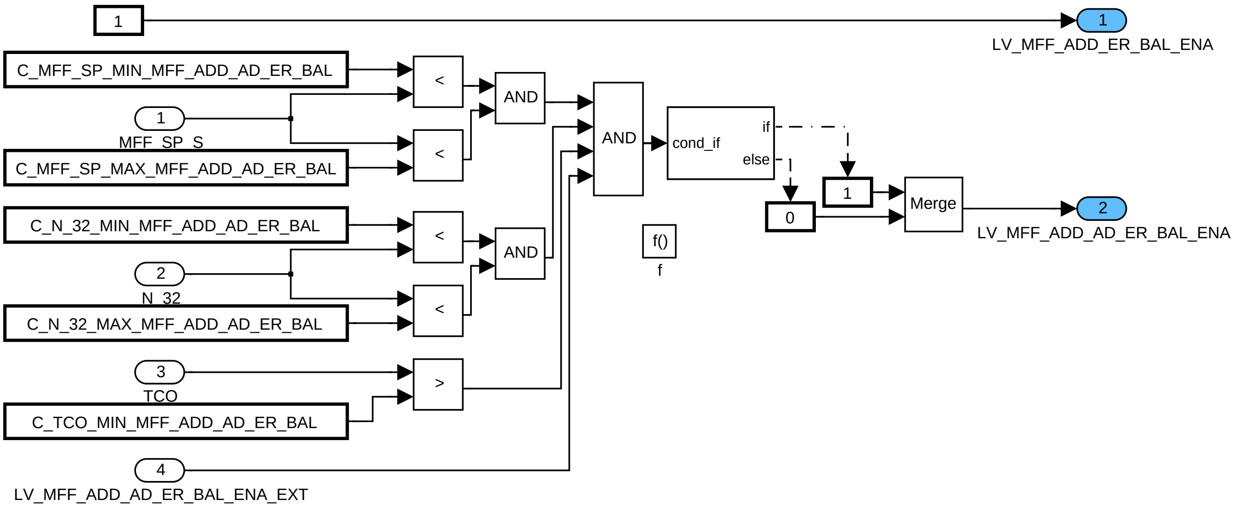


Figure 167 CYBL_M30907L03/ OPERATE_SEG/ ACT_CND_ADD_MFF_COR/ THEN

State of cylinder balancing control via TI intervention:

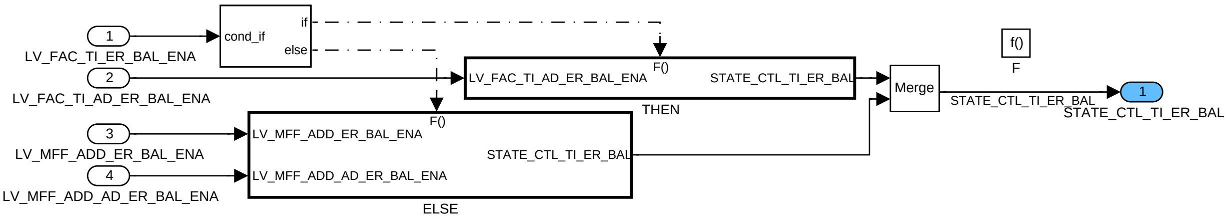


Figure 168 CYBL_M30907L03/ OPERATE_SEG/ STATE_CYBL_CTL_TI_INTV

State of cylinder balancing control via TI intervention – THEN-branch

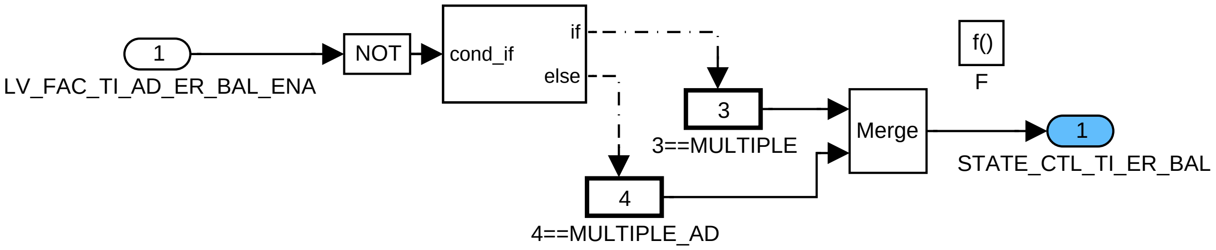


Figure 169 CYBL_M30907L03/ OPERATE_SEG/ STATE_CYBL_CTL_TI_INTV/ THEN

State of cylinder balancing control via TI intervention – ELSE-branch

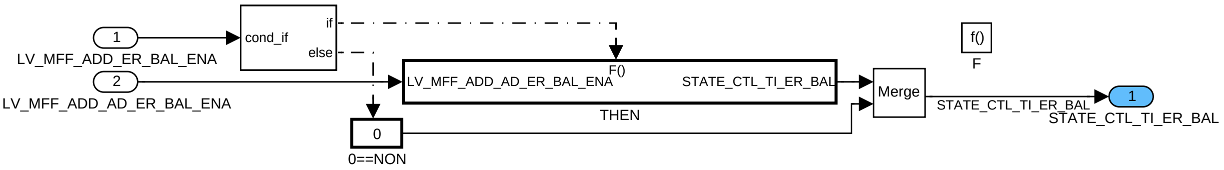


Figure 170 CYBL_M30907L03/ OPERATE_SEG/ STATE_CYBL_CTL_TI_INTV/ ELSE

State of cylinder balancing control via TI intervention – ELSE-THEN-branch

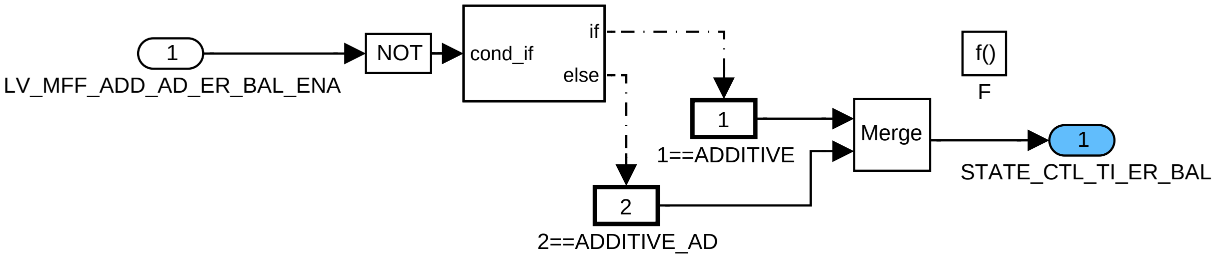
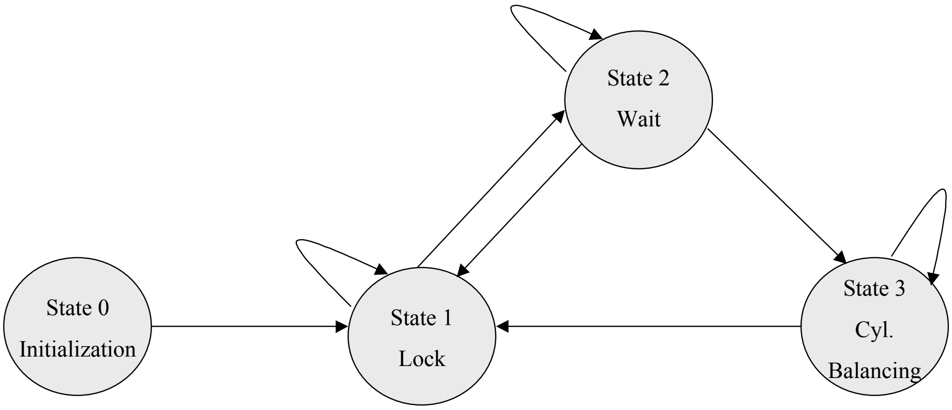


Figure 171 CYBL_M30907L03/ OPERATE_SEG/ STATE_CYBL_CTL_TI_INTV/ ELSE/ THEN

Description of the Cylinder Balancing Manager

State diagram:



Cylinder Balancing Manager: State 0H – "Initialization"

At reset the state of cylinder balancing changes at once to state "Initialization". The state "Initialization" is active for one recurrence (first segment after reset). As soon as the next segment (second segment after reset) is valid, the state changes immediately to state "Lock".

Cylinder Balancing Manager: State 1[H] – "Lock"

The algorithm stays in this state until the flags LV_TI_ER_BAL_ENA is set to "1". This flag is set when certain conditions for controlling and adaptation are fulfilled.

Cylinder Balancing Manager: State 2[H] – "Wait"

The counter CTR_TI_ER_BAL_ENA starts to run. It counts down from its calibrate able initial value IP_CTR_TI_ER_BAL_ENA_INI to zero (decrease of "1" every engine cycle).

Furthermore the input values ER_STND_MMV_BAL[SEG_NR_ER] and ER_STD_MMV_BAL[SEG_NR_ER] have to below a certain limit:

Cylinder Balancing Manager: State [3] – Cylinder Balancing

The input values ER_STND_MMV_BAL[SEG_NR_ER] and ER_STD_MMV_BAL[SEG_NR_ER] have to below a certain limit (analogue to state 2).

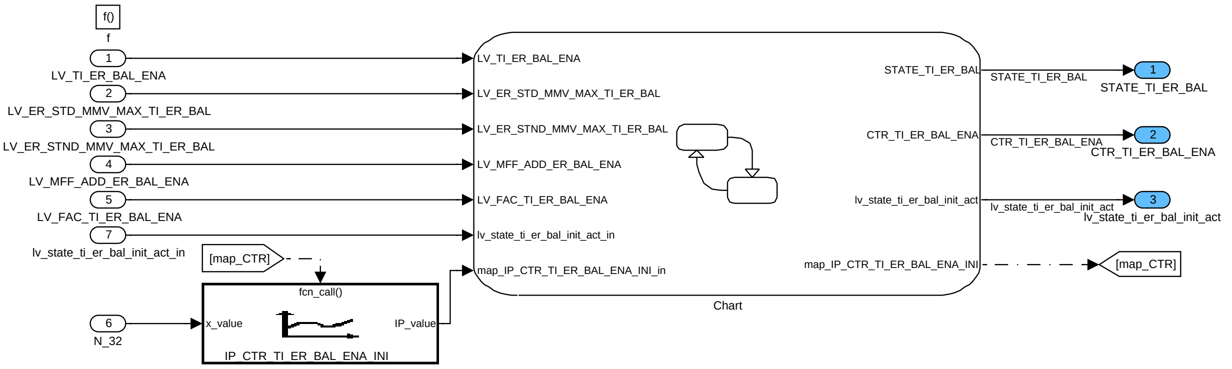


Figure 172 CYBL_M30907L03/ OPERATE_SEG/ CYBL_MNG

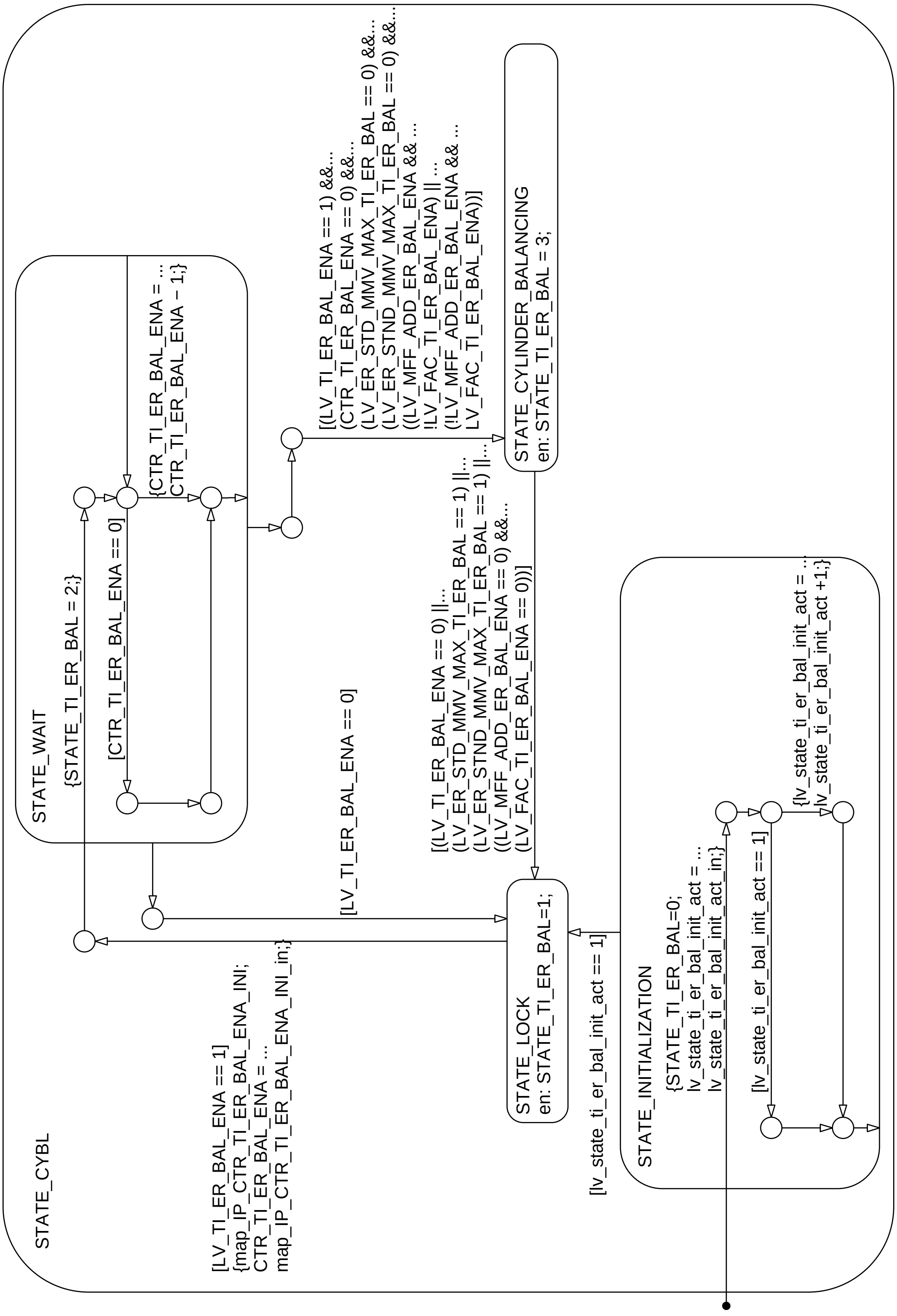


Figure 173 CYBL_M30907L03/ OPERATE_SEG/ CYBL_MNG/ Chart

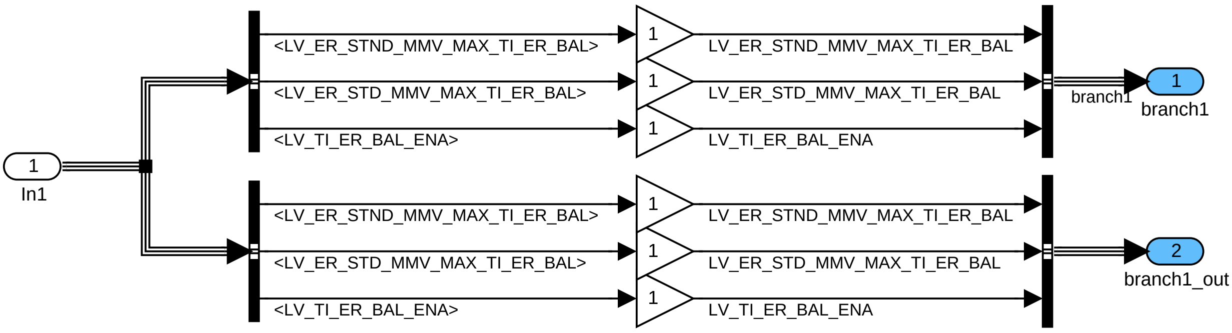


Figure 174 CYBL_M30907L03/ OPERATE_SEG/ BRANCH1

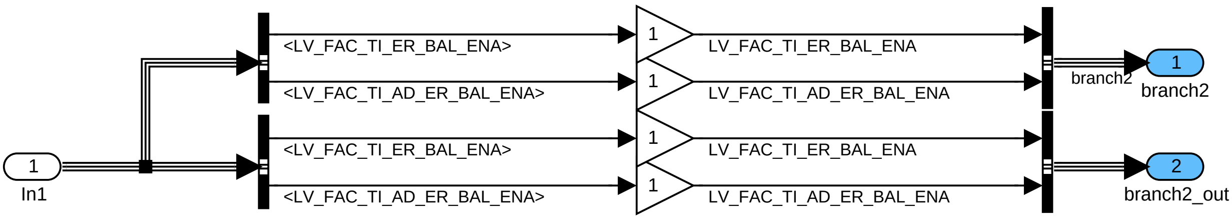


Figure 175 CYBL_M30907L03/ OPERATE_SEG/ BRANCH2

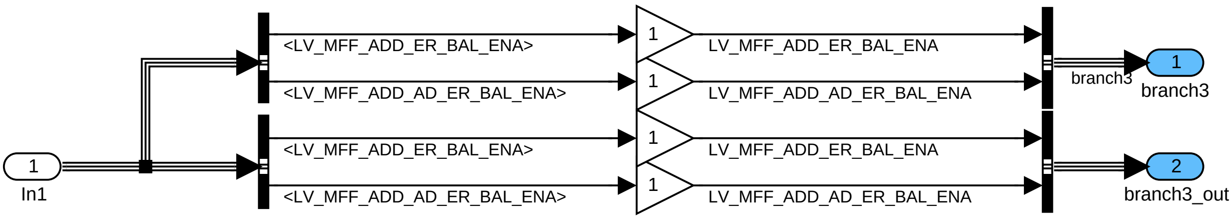


Figure 176 CYBL_M30907L03/ OPERATE_SEG/ BRANCH3

6.7.1.3 SUBFUNCTION: SIG_MNG

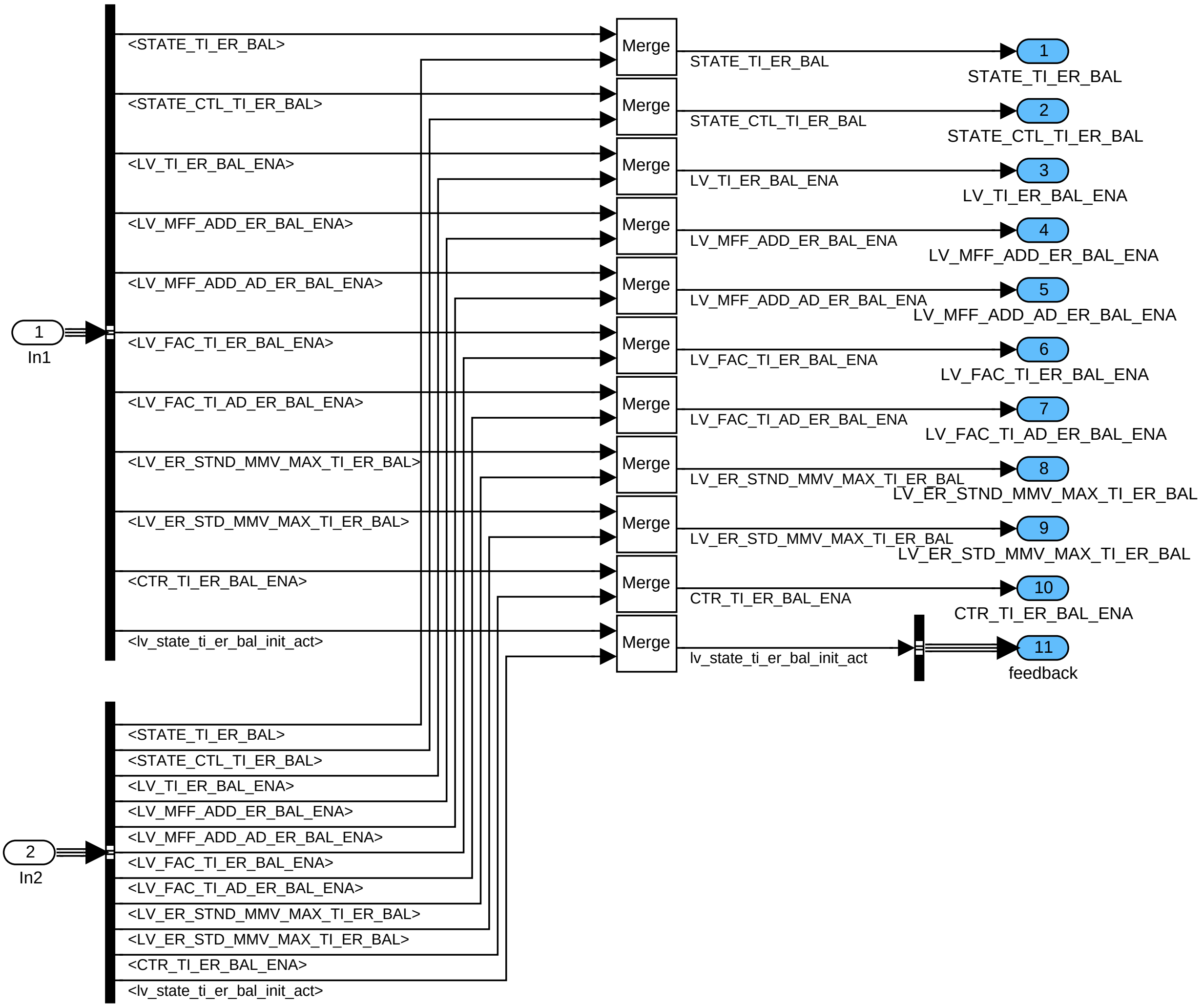


Figure 177 CYBL_M30907L03/ SIG_MNG

6.8 Cylinder balancing manager (Appl Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ER_STND_ER_BAL_ACT	O/V	0...1H	0...1	1	-
Activation condition to enable cylinder balancing signal preparation calculation					
LV_DRV1_ER_BAL_ACT	O/V	0...1H	0...1	1	-
Activation condition to enable cylinder balancing fade out calculation					
LV_TI_ER_BAL_ACT	O/V	0...1H	0...1	1	-
Activation condition to enable cylinder balancing TI intervention calculation					
LV_TI_CYL_BAL_ER_ACT	O/V	0...1H	0...1	1	-
Flag for cylinder balancing (TI intervention) via engine roughness activated					
LV_TI_CYL_BAL_LAM_ACT	O/V	0...1H	0...1	1	-
Flag for cylinder balancing (TI intervention) via lambda sensor activated					
LV_MFF_COR_ER_BAL_ENA_EXT	O/V	0...1H	0...1	1	-
External condition to enable cylinder balancing manager calculations					
LV_MFF_ADD_AD_ER_BAL_ENA_EXT	O/V	0...1H	0...1	1	-
External condition to enable cylinder balancing additive adaptation manager calculations					
LV_MFF_FAC_AD_ER_BAL_ENA_EXT	O/V	0...1H	0...1	1	-
External condition to enable cylinder balancing multiple adaptation manager calculations					
LV_CYL_BAL_ER_CDN_BAS	O/V	0...1H	0...1	1	-
Basic condition to enable cylinder balancing calculations					

Input data:

STATE_TI_ER_BAL	LV_FL	LV_ER_FDOUT	LV_MIS_STATE_A
LV_ERR_CAM	LV_STATE_RR	TQI_AV_S	LV_CH
LV_SEG_AD_LIM_ER	LV_MIS_STATE_B4	LV_ERR_TCO	LV_INH_CYL_BAL_LAM_LIH_CTL
T_AST	LV_S_ACT	LV_SEG_AD_AVL_ER	LV_ERR_CRK
LV_ERR_TPS	LV_INH_CYL_BAL_ER_LIH_CTL	LV_PUC	LV_ENA_ER
LV_DET_MIS	LV_RGN_NT_REQ	LV_ERR_CAT_DIAG[NC_CBK_EX_NR]	LV_ERR_T_SEG_ER
LV_ERR_SEG_AD_ER	LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS
LV_ERR_FUP_ST	LV_ERR_FUP_ORNG	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_VCV
LV_INH_BAL_CUS	LV_ERR_MAP_TPS_PLAUS	NC_CBK_EX_NR	NC_CYL_NR
LV_ERR_CYL_BAL_ER[NC_CYL_NR]	LV_FCUT_IND	STATE_ERR_IV	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DRV1_ER_BAL_ACT_MAN	1	0H 1H 2H	NEUTRAL ENABLE DISABLE	1	-
Manual adjustment of activation conditions for cylinder balancing fade out					
C_ER_STND_ER_BAL_ACT_MAN	1	0H 1H 2H	NEUTRAL ENABLE DISABLE	1	-
Manual adjustment of activation conditions for cylinder balancing signal preparation					
C_TI_ER_BAL_ACT_MAN	1	0H 1H 2H	NEUTRAL ENABLE DISABLE	1	-
Manual adjustment of activation conditions for cylinder balancing via TI intervention					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TQI_MAX_TI_ER_BAL	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Maximum engine torque for cylinder balancing via TI intervention					
C_TQI_MIN_TI_ER_BAL	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Minimum engine torque for cylinder balancing via TI intervention					
C_T_AST_MIN_TI_ER_BAL	1	0...FFFFH	0...6.5535E+3	0.1	s
Minimum time after engine start for cylinder balancing via TO intervention					
LC_TI_ER_BAL_STOP_MAN	1	0...1H	0...1	1	-
Logical constant for manual stop (=1) of all cylinder balancing functions (TI intervention)					

6.8.1 General information

The Appl. Inc. module is used to keep the cylinder balancing functions modular. Therefore the activation conditions for the signal preparation functions, the balancing manager and the cylinder balancing adaptation functions are set within this module. The cylinder balancing functions are activated as soon as the corresponding activation condition is enabled (=1).

With the logical variable LV_MFF_COR_ER_BAL_ENA_EXT it is possible to adapt the main enable condition (to start cylinder balancing via TI intervention) with additional external requirements. The flag is used as input signal for the cylinder balancing manager.

With the logical variable LV_MFF_ADD_AD_ER_BAL_ENA_EXT it is possible to adapt the main adaptation condition (to start cylinder balancing additive adaptation) with additional external requirements. The flag is used as input signal for the cylinder balancing manager.

A logical calibration constant is available to stop all cylinder balancing interventions at once. In case of LC_TI_ER_BAL_STOP_MAN = 1, several activation conditions for the cylinder balancing functions are disabled (=0). If the logical constant is enabled (=1), a re-initialisation ("0"[hex]) of all output values occurs.

Application Condition

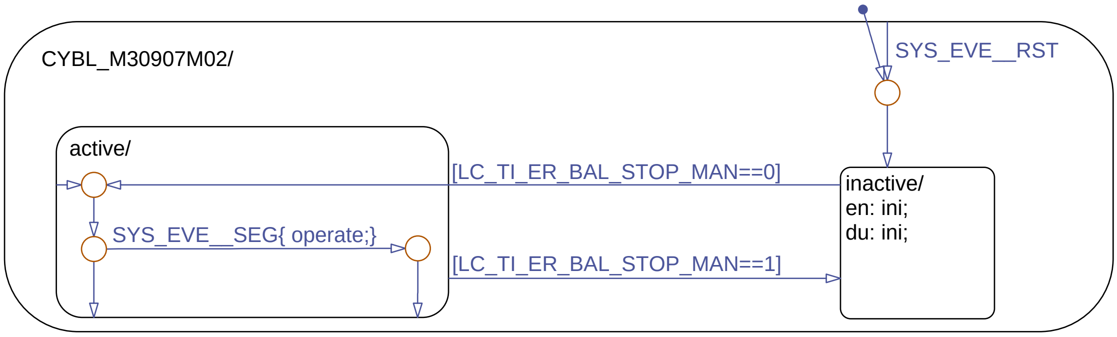
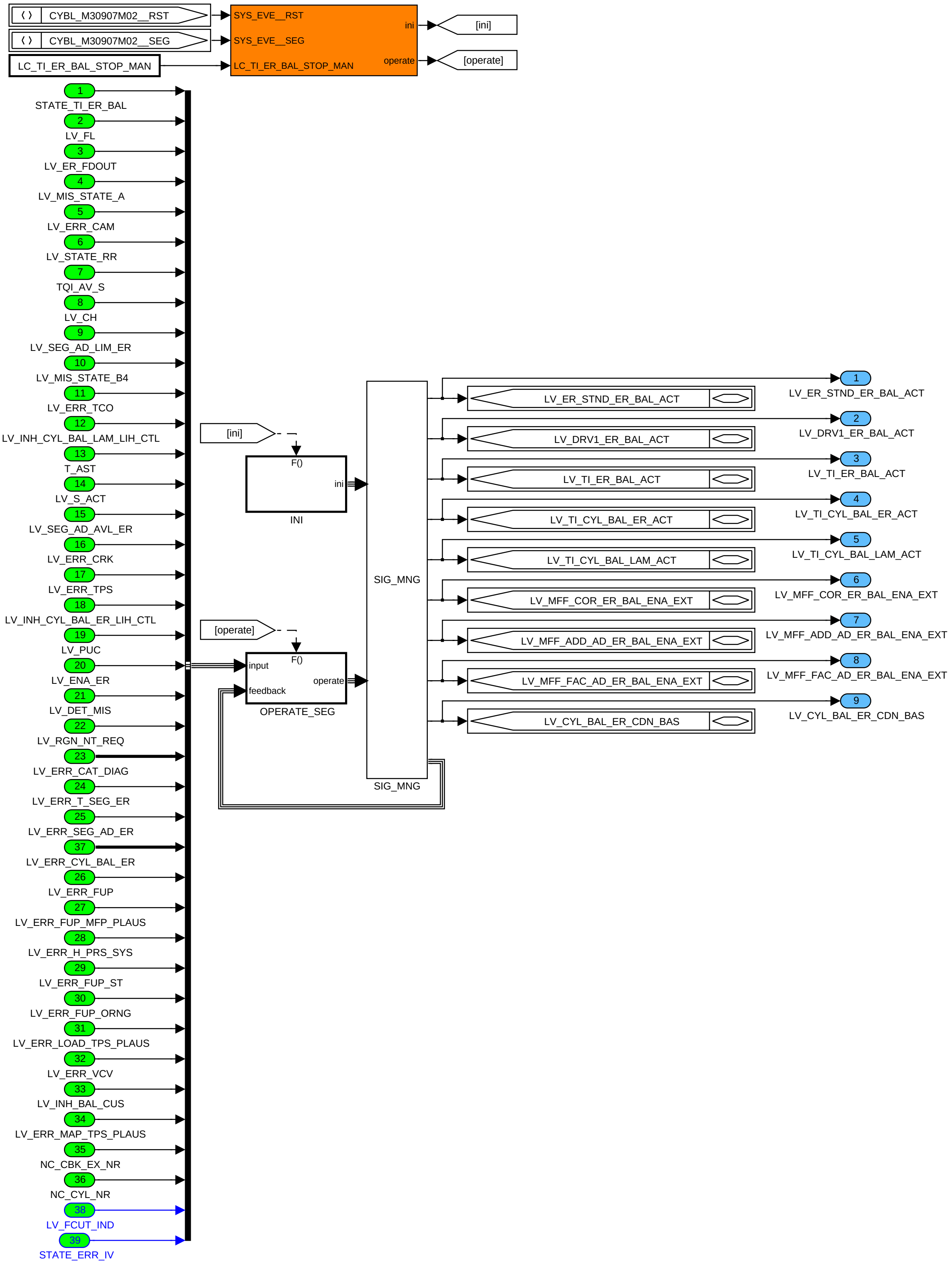


Figure 178 CYBL_M30907M02/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 23−Aug−2006

Figure 179 CYBL_M30907M02

6.8.1.1 Calculation of all variables at reset task or LC_TI_ER_BAL_STOP_MAN = 1

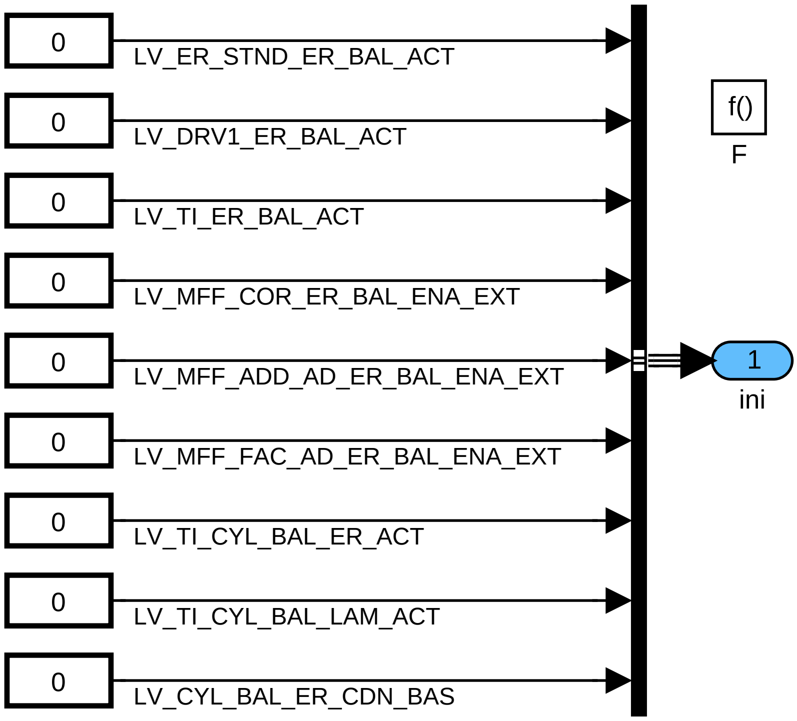


Figure 180 CYBL_M30907M02/ INI

6.8.1.2 Calculation of segment task

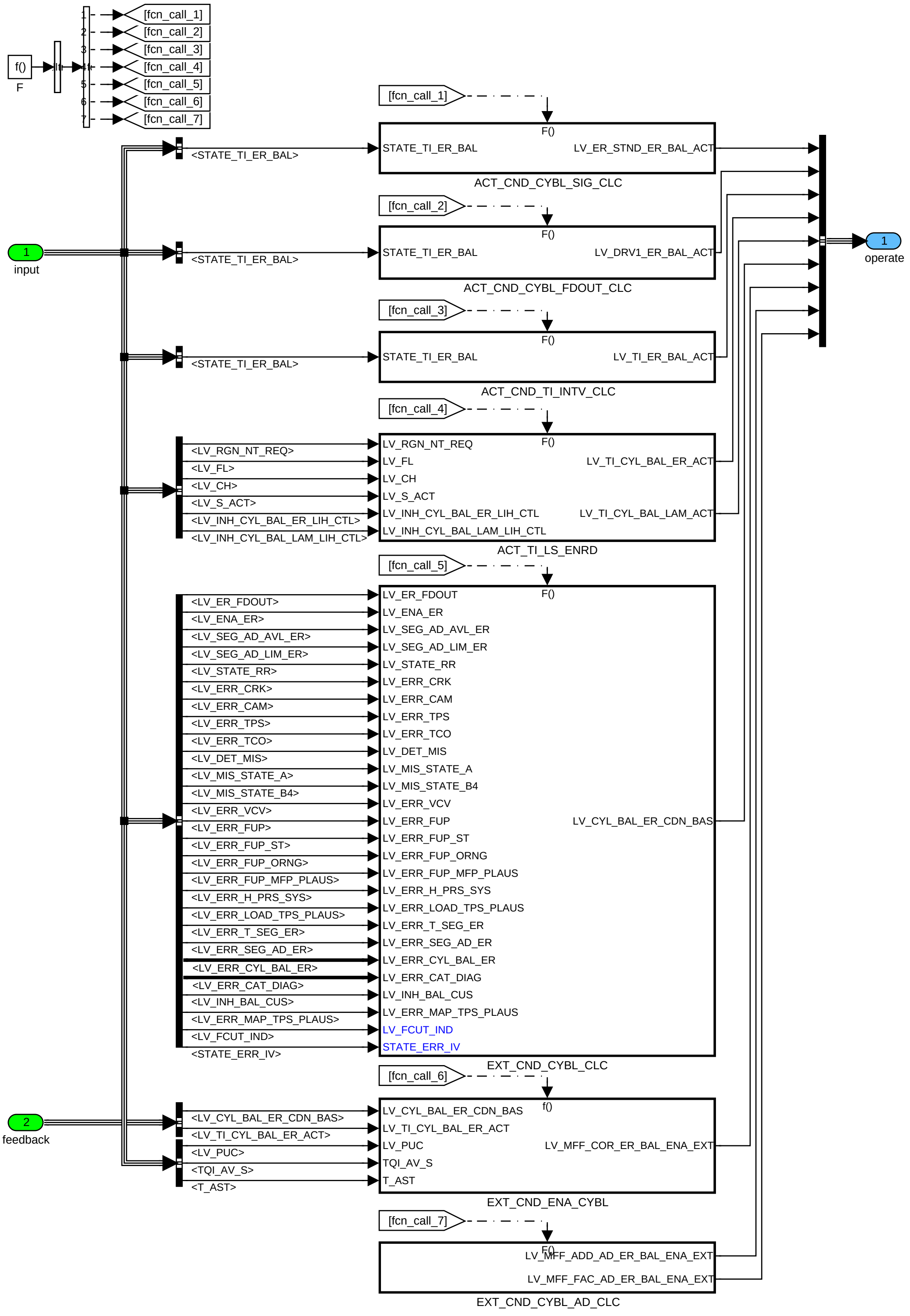


Figure 181 CYBL_M30907M02/ OPERATE_SEG

Activation conditions for cylinder balancing signal preparation calculation

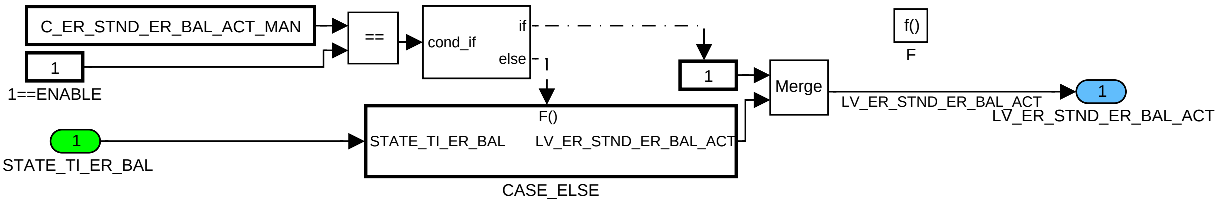


Figure 182 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_CYBL_SIG_CLC

ACT_CND_CYBL_SIG_CLC – elseif-branch

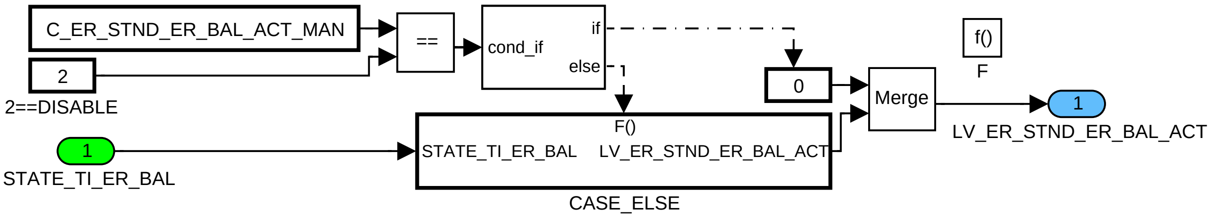


Figure 183 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_CYBL_SIG_CLC/ CASE_ELSE

ACT_CND_CYBL_SIG_CLC – elseif-branch – else-branch

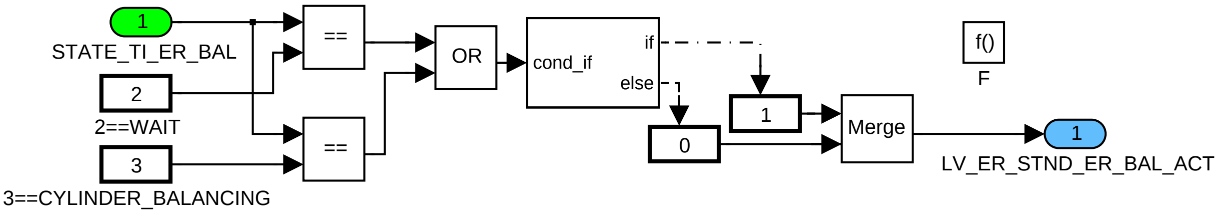


Figure 184 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_CYBL_SIG_CLC/ CASE_ELSE/ CASE_ELSE

Activation conditions for cylinder balancing fade out calculation

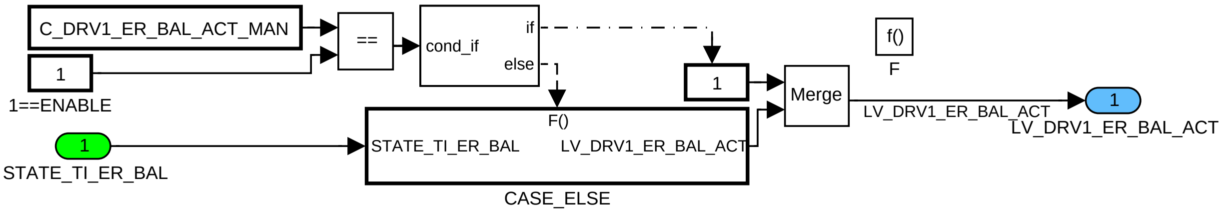


Figure 185 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_CYBL_FDOUT_CLC

ACT_CND_CYBL_FDOUT_CLC – elseif-branch

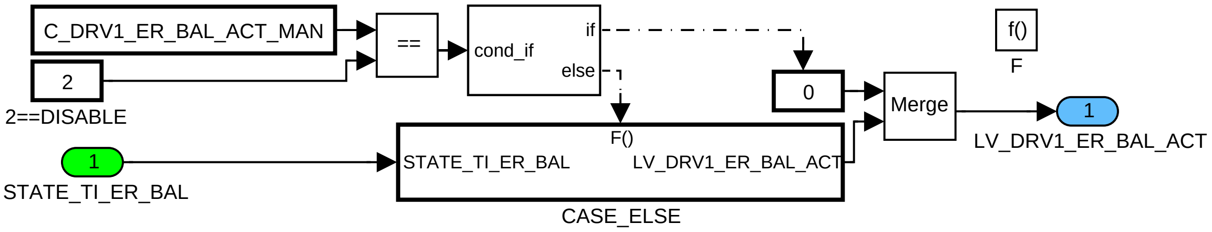


Figure 186 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_CYBL_FDOUT_CLC/ CASE_ELSE

ACT_CND_CYBL_FDOUT_CLC – elseif-branch – else-branch

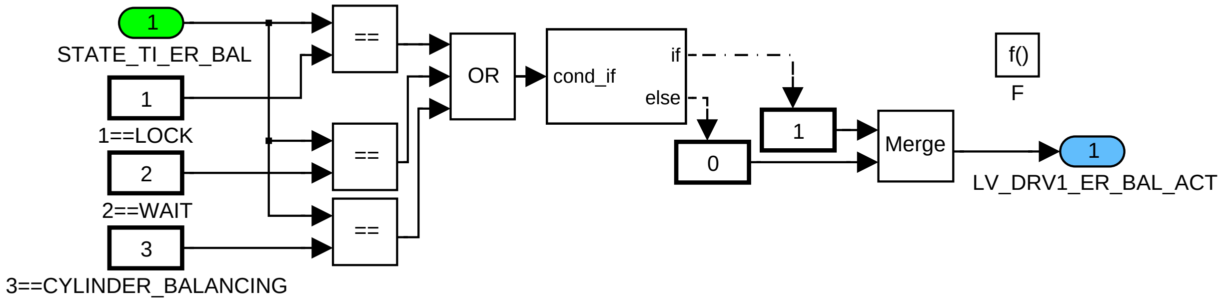


Figure 187 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_CYBL_FDOUT_CLC/ CASE_ELSE/ CASE_ELSE

Activation conditions for cylinder balancing TI intervention calculation

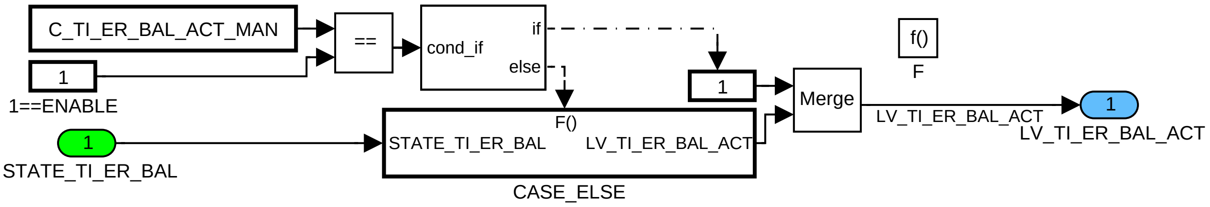


Figure 188 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_TI_INTV_CLC

ACT_CND_TI_INTV_CLC – elseif-branch

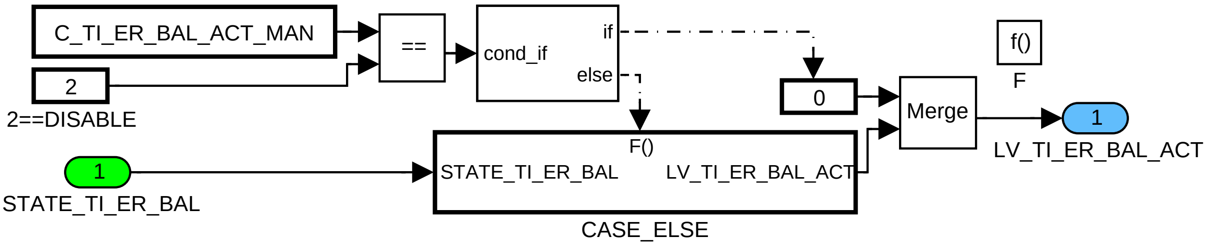


Figure 189 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_TI_INTV_CLC/ CASE_ELSE

ACT_CND_TI_INTV_CLC – elseif-branch – else-branch

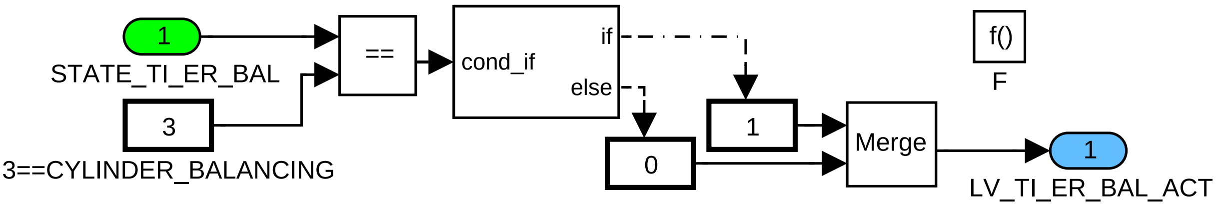


Figure 190 CYBL_M30907M02/ OPERATE_SEG/ ACT_CND_TI_INTV_CLC/ CASE_ELSE/ CASE_ELSE

Activation of cylinder balancing (TI intervention) via lambda sensor or engine roughness

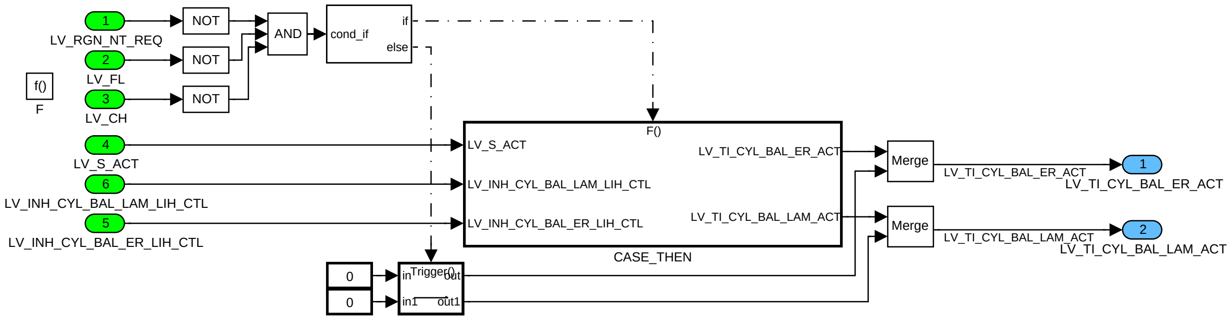


Figure 191 CYBL_M30907M02/ OPERATE_SEG/ ACT_TI_LS_ENRD

ACT_TI_LS_ENRD – thenif-branch

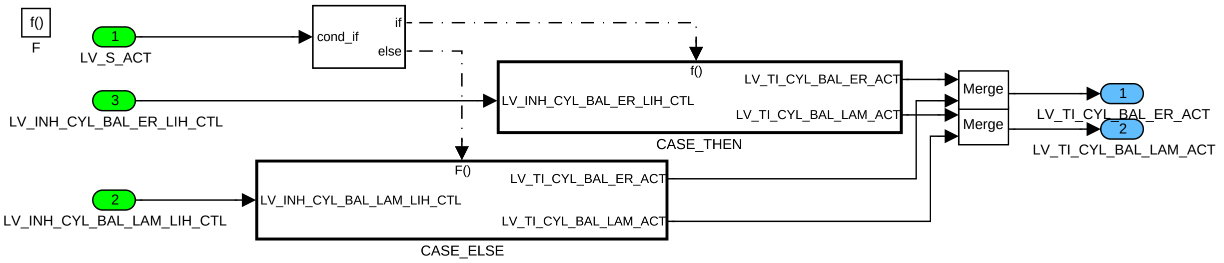


Figure 192 CYBL_M30907M02/ OPERATE_SEG/ ACT_TI_LS_ENRD/ CASE_THEN

ACT_TI_LS_ENRD – thenif-branch – else-branch

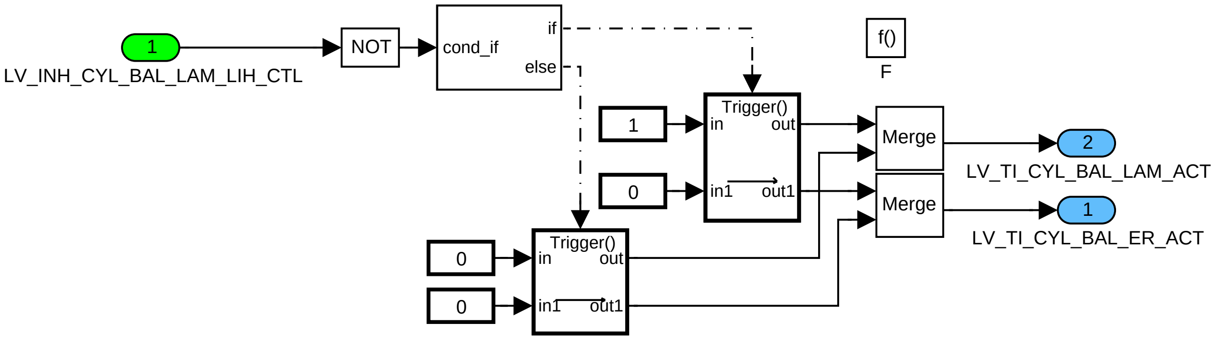


Figure 193 CYBL_M30907M02/ OPERATE_SEG/ ACT_TI_LS_ENRD/ CASE_THEN/ CASE_ELSE

ACT_TI_LS_ENRD – thenif-branch CASE_THEN

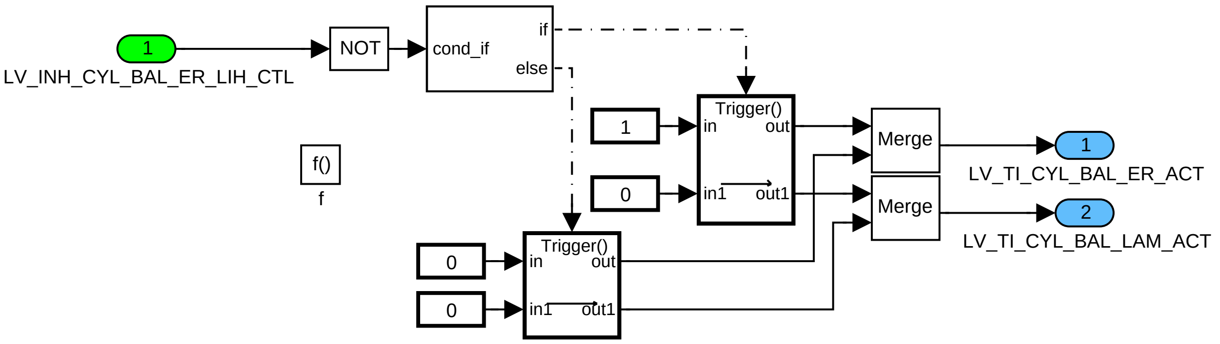


Figure 194 CYBL_M30907M02/ OPERATE_SEG/ ACT_TI_LS_ENRD/ CASE_THEN/ CASE_THEN

Basic condition to enable cylinder balancing manager calculations

LV_ER_FDOUT (flag for engine roughness fade out condition)

LV_ENA_ER (calculation of engine roughness (ER) values valid)

LV_SEG_AD_AVL_ER (segment adaptation process achieved at least one time)

LV_SEG_AD_LIM_ER (engine roughness (ER) adaptation values not out of range)

LV_STATE_RR (no rough road detected)

LV_ERR_CRK (no failure present on crankshaft sensor)

LV_ERR_CAM (no failure present on camshaft sensor)

LV_ERR_TPS (flag for throttle position error)

LV_ERR_TCO (flag for TCO sensor error)

LV_ERR_VCV (flag for voltage supply error)

LV_ERR_FUP (flag for fuel pressure sensor error)

LV_ERR_FUP_ST (flag for "fuel pressure at start to low" error)

LV_ERR_FUP_ORNG (flag for "fuel pressure out of range" error)

LV_ERR_FUP_MFP_PLAUS (flag for "FUP-mass fuel pump relation" error)

LV_ERR_H_PRS_SYS (flag for abnormal fuel pressure error)

LV_ERR_LOAD_TPS_PLAUS (flag for "LOAD-throttle position relation" error)

LV_ERR_T_SEG_ER (flag for "segment time adaptive value" error)

LV_ERR_SEG_AD_ER (flag for "ER segment time adaptation" error)

LV_ERR_CYL_BAL_ER[x] (flag for cylinder balancing error, AND condition over all cylinders

[x] is representing each cylinder separately [1...NC_CYL_NR])

LV_ERR_CAT_DIAG[i] (flag for cat diagnosis error, AND condition over all exhaust banks

[i] is representing each exhaust bank separately [1...NC_CBK_EX_NR])

LV_INH_BAL_CUS (customer request to inhibit the function]

LV_ERR_MAP_TPS_PLAUS (flag for MAP-throttle position plausibility error)

LV_DET_MIS (status for actual detected misfire)

LV_MIIS_STATE_A (no CARB A[200 CRK] misfire failure present)

LV_MIS_STATE_B4 (no CARB B4[1000 CRK] misfire failure present)

LV_FCUT_IND (flag for cylinder shut off)

STATE_ERR_IV (error pattern for injection valve failure)

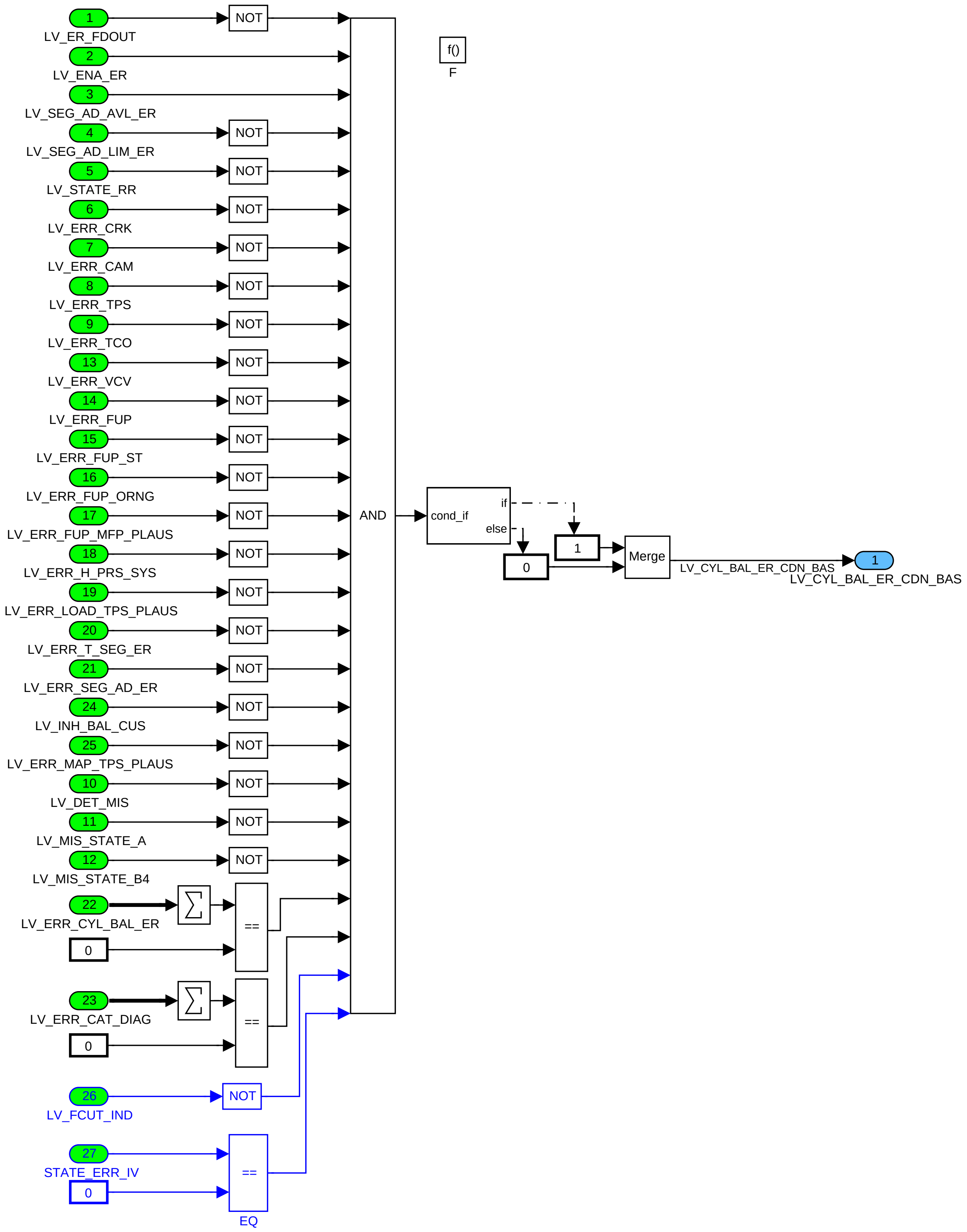


Figure 195 CYBL_M30907M02/ OPERATE_SEG/ EXT_CND_CYBL_CLC

External condition to enable cylinder balancing manager calculations

LV_CYL_BAL_ER_CND_BAS (Flag for basic condition to enable cylinder balancing calculations)

LV_TI_CYL_BAL_ER_ACT (Flag for cylinder balancing (TI intervention) via ER activated)

LV_PUC (flag for trailing throttle fuel cut off)

C_TQI_MIN_TI_ER_BAL (minimum indicated engine torque for cylinder balancing)

C_TQI_MAX_TI_ER_BAL (maximum indicated engine torque for cylinder balancing)

C_T_AST_MIN_TI_ER_BAL (minimum time after start for cylinder balancing)

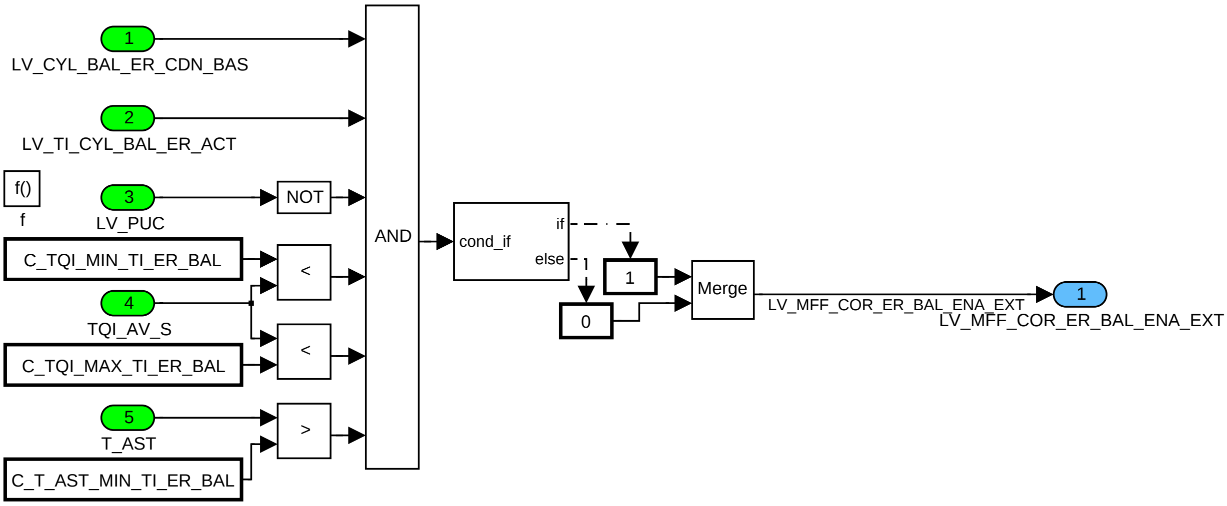


Figure 196 CYBL_M30907M02/ OPERATE_SEG/ EXT_CND_ENA_CYBL

External conditions to enable cylinder balancing adaptation manager calculations

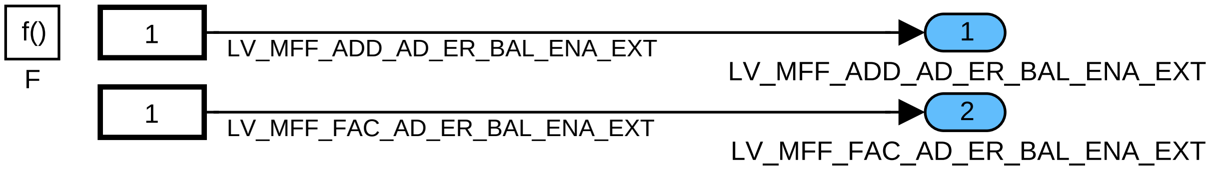


Figure 197 CYBL_M30907M02/ OPERATE_SEG/ EXT_CND_CYBL_AD_CLC

6.8.1.3 SUBFUNCTION: SIG_MNG

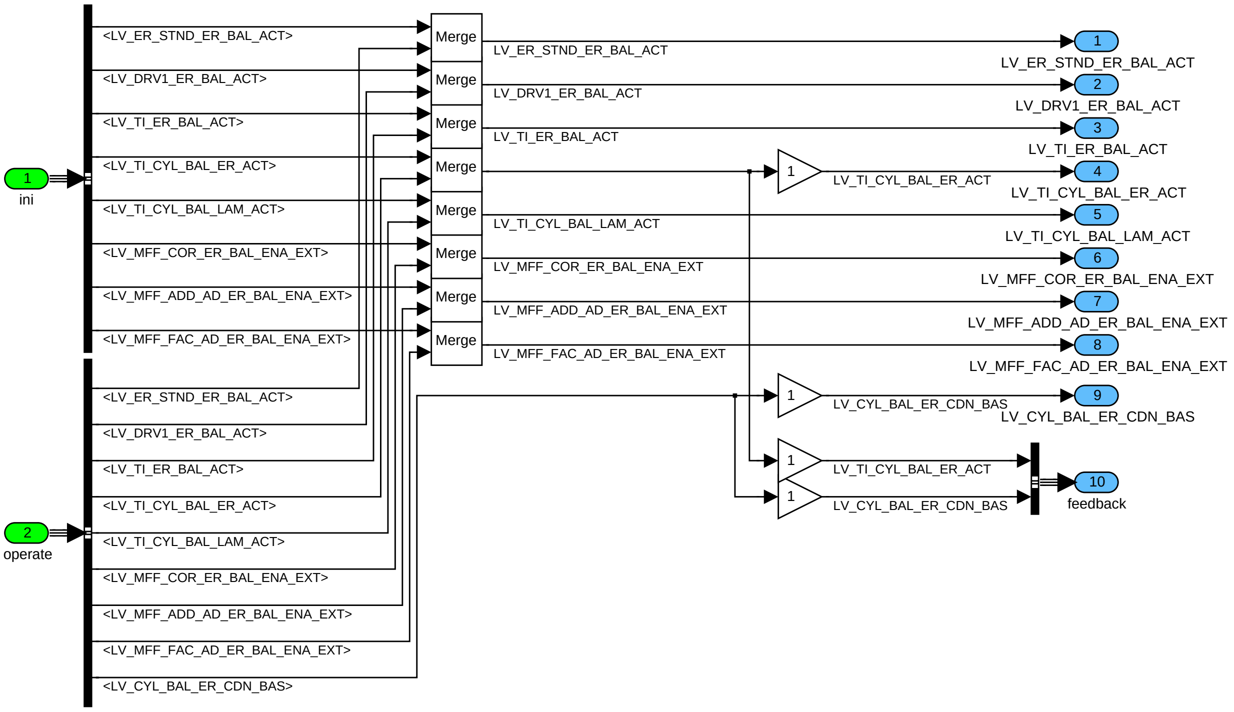


Figure 198 CYBL_M30907M02/ SIG_MNG

6.9 CYBL scheduler

Input data:

LC_TI_ER_BAL_STOP_MAN	LC_FAC_TI_BAL_RST_MAN	LC_MFF_ADD_ER_BAL_RST_MAN	LC_MFF_ADD_AD_ER_BAL_RST_MAN
LC_FAC_TI_ER_BAL_RST_MAN	LC_FAC_TI_AD_ER_BAL_RST_MAN	ER_STND_MMV_BAL[NC_CYL_NR]	ER_STND_MMV_STD_BAL[NC_CYL_NR]
ER_STD_MMV_BAL[NC_CYL_NR]	MFF_ADD_AD_ER_BAL[NC_CYL_NR]	LV_CYL_BAL_LAM_AD_EOL	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	LV_CYL_BAL_ER_AD_FAC_EOL	LV_CYL_BAL_ER_AD_ADD_EOL	LV_CYL_BAL_LAM_SEL_AD_COLD[NC_CBK_EX_NR]
LV_CYL_BAL_LAM_SEL_AD_RNG_H[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_FAC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_ADD[NC_CBK_EX_NR]	LC_CYL_BAL_AD_DC_RST
LC_CYL_BAL_CORD_EOL_STOP_MAN	LC_CYL_BAL_CORD_DC_STOP_MAN	NC_CYL_NR	NC_CBK_EX_NR
LC_CYL_BAL_AD_EOL_RST	FAC_TI_AD_ER_BAL[NC_CYL_NR]	T_CYL_BAL_LAM_SEL_DC	T_CYL_BAL_LAM_AD_DC
LV_CYL_BAL_LAM_AD_DC	LV_CYL_BAL_LAM_SEL_AD_HOT_DC	LV_CYL_BAL_LAM_SEL_AD_COLD_DC	LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[NC_CBK_EX_NR]
LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_ADD_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_FAC_L_DC[NC_CBK_EX_NR]
LV_CYL_BAL_LAM_AD_FAC_H_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_SEL_AD_RNG_L[NC_CBK_EX_NR]		

FUNCTION DESCRIPTION:

General information:

This manager specifies the sequencing of all CYBL tasks.

Description:

Within the cylinder balancing package (CYBL), all functions for Cylinder balancing via engine roughness" (CYBL_ER) are included. This scheduler has to manage the coordination of all involved parts.

Application conditions:

Recurrence: see formula section below

Activation: at every engine operating state

Deactivation: -

Formula section:

6.9.1 Non volatile memory tasks (CYBL_ER):

NVMY_STB	
NVMY_STB	Engine roughness signal preparation for Cylinder balancing (402U) /* ER_STND_MMV_BAL[x] (NVMY) = 0 /* ER_STND_MMV_STD_BAL[x] (NVMY) = 0 /* ER_STD_MMV_BAL[x] (NVMY) = 0
NVMY_STB	Cylinder balancing via additive MFF intervention (7085) /* MFF_ADD_AD_ER_BAL[x] (NVMY) = 0
NVMY_STB	Cylinder balancing via multiple TI intervention (707Q) /* FAC_TI_AD_ER_BAL[x] (NVMY) = 0
NVMY_STB	Cylinder balancing coordination at EOL (908O) /* LV_CYL_BAL_LAM_AD_EOL (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_COLD_EOL (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_HOT_EOL (NVMY) = 0 /* LV_CYL_BAL_ER_AD_FAC_EOL (NVMY) = 0 /* LV_CYL_BAL_ER_AD_ADD_EOL (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_COLD[m] (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] (NVMY) = 0 /* LV_CYL_BAL_LAM_AD_FAC[m] (NVMY) = /* LV_CYL_BAL_LAM_AD_ADD[m] (NVMY) = 0
NVMY_STB	Cylinder balancing coordination at DC (908P) /* T_CYL_BAL_LAM_SEL_DC (NVMY) = 0 /* T_CYL_BAL_LAM_AD_DC (NVMY) = 0 /* LV_CYL_BAL_LAM_AD_DC (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_HOT_DC (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_COLD_DC (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] (NVMY) = 0 /* LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] (NVMY) = 0 /* LV_CYL_BAL_LAM_AD_ADD_DC[m] (NVMY) = 0 /* LV_CYL_BAL_LAM_AD_FAC_L_DC[m] (NVMY) = 0 /* LV_CYL_BAL_LAM_AD_FAC_H_DC[m] (NVMY) = 0
NVMY_RST	
NVMY_RST	Engine roughness signal preparation for Cylinder balancing (402U) /* ER_STND_MMV_BAL[x] (NVMY) --> (RAM) /* ER_STND_MMV_STD_BAL[x] (NVMY) --> (RAM) /* ER_STD_MMV_BAL[x] (NVMY) --> (RAM)
NVMY_RST	Cylinder balancing via additive MFF intervention (7085) /* MFF_ADD_AD_ER_BAL[x] (NVMY) --> (RAM)
NVMY_RST	Cylinder balancing via multiple TI intervention (707Q) /* FAC_TI_AD_ER_BAL[x] (NVMY) --> (RAM)
NVMY_RST	Cylinder balancing coordination at EOL (908O) /* LV_CYL_BAL_LAM_AD_EOL (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_AD_COLD_EOL (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_AD_HOT_EOL (NVMY) --> (RAM) /* LV_CYL_BAL_ER_AD_FAC_EOL (NVMY) --> (RAM) /* LV_CYL_BAL_ER_AD_ADD_EOL (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_AD_COLD[m] (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_AD_FAC[m] (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_AD_ADD[m] (NVMY) --> (RAM)
NVMY_RST	Cylinder balancing coordination at DC (908P) /* T_CYL_BAL_LAM_SEL_DC (NVMY) --> (RAM) /* T_CYL_BAL_LAM_AD_DC (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_AD_DC (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_AD_HOT_DC (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_AD_COLD_DC (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] (NVMY) --> (RAM)

	/* LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_AD_ADD_DC[m] (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_AD_FAC_L_DC[m] (NVMY) --> (RAM) /* LV_CYL_BAL_LAM_AD_FAC_H_DC[m] (NVMY) --> (RAM)
NVMY_UPD	
NVMY_UPD	Engine roughness signal preparation for Cylinder balancing (402U) /* ER_STND_MMV_BAL[x] (RAM) --> (NVMY) /* ER_STND_MMV_STD_BAL[x] (RAM) --> (NVMY) /* ER_STD_MMV_BAL[x] (RAM) --> (NVMY)
NVMY_UPD	Cylinder balancing via additive MFF intervention (7085) /* MFF_ADD_AD_ER_BAL[x] (RAM) --> (NVMY)
NVMY_UPD	Cylinder balancing via multiple TI intervention (707Q) /* FAC_TI_AD_ER_BAL[x] (RAM) --> (NVMY)
NVMY_UPD	Cylinder balancing coordination at EOL (908O) /* LV_CYL_BAL_LAM_AD_EOL (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_COLD_EOL (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_HOT_EOL (RAM) --> (NVMY) /* LV_CYL_BAL_ER_AD_FAC_EOL (RAM) --> (NVMY) /* LV_CYL_BAL_ER_AD_ADD_EOL (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_COLD[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_AD_FAC[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_AD_ADD[m] (RAM) --> (NVMY)
NVMY_UPD	Cylinder balancing coordination at DC (908P) /* T_CYL_BAL_LAM_SEL_DC (RAM) --> (NVMY) /* T_CYL_BAL_LAM_AD_DC (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_AD_DC (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_HOT_DC (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_COLD_DC (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_AD_ADD_DC[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_AD_FAC_L_DC[m] (RAM) --> (NVMY) /* LV_CYL_BAL_LAM_AD_FAC_H_DC[m] (RAM) --> (NVMY)
/* *** Description *** /* NVMY_STB: --> Initialization of NVMY cells at first engine run or NVMY error /* NVMY_RST: --> Initialization of RAM cells with NVMY cells at RESET /* NVMY_UPD: --> Initialization of NVMY cells with RAM cells at ECU power latch /* [x] is representing [NC_CYL_NR] at the task definition above /* [m] is representing [NC_CBK_EX_NR] at the task definition above	

6.9.2 Initialization tasks (CYBL_ER):

RESET (no sequencing behind – only for info)	
RST	/* the RESET of aggregate ENTE has to be performed before CYBL RESET tasks
RST	Cylinder balancing via ER (Appl. Inc.) (704N)
RST	Cylinder balancing manager (Appl. Inc.) (907M)
RST	Engine roughness signal preparation for Cylinder balancing (402U)
RST	Cylinder balancing manager (907L)
RST	Cylinder balancing via multiple TI intervention (707Q)
RST	Cylinder balancing via additive MFF intervention (7085)
RST	Coord. of the injection time correction factors for cylinder balancing (704P)
RST	Cylinder balancing coordination at DC (908P)
RST	Cylinder balancing coordination at EOL (908O)
EXIT_ST (no sequencing behind – only for info)	

EXIT_ST	Cylinder balancing coordination at EOL (908O – chapter 1.3)
STOP_MAN (no system event , section located within SW code- only for info)	
# IF (LC_TI_ER_BAL_STOP_MAN = 1)	
STOP_MAN	Cylinder balancing manager (Appl. Inc.) (907M) /* Initialization at manual stop of function
STOP_MAN	Cylinder balancing manager (907L) /* Initialization at manual stop of function
# ENDIF	
# IF (LC_CYL_BAL_CORD_EOL_STOP_MAN = 1)	
STOP_MAN	Cylinder balancing coordination at EOL (908O - chapter 1.4, 1.5, 1.6) /* Initialization at manual stop of functions
# ENDIF	
# IF (LC_CYL_BAL_CORD_DC_STOP_MAN = 1)	
STOP_MAN	Cylinder balancing coordination at DC (908P - chapter 1.2, 1.3, 1.4, 1.5, 1.6) /* Initialization at manual stop of function
# ENDIF	
RESET_MAN (no system event, section located within SW code – only for info)	
# IF (LC_FAC_TI_BAL_RST_MAN = 1)	
RST_MAN	Coord. of the injection time correction factors for cylinder balancing (704P) /* Manual reset of injector energy correction values
# ENDIF	
# IF (LC_MFF_ADD_ER_BAL_RST_MAN = 1)	
RST_MAN	Cylinder balancing via additive MFF intervention (7085) /* Manual reset of additive correction values
# ENDIF	
# IF (LC_MFF_ADD_AD_ER_BAL_RST_MAN = 1)	
RST_MAN	Cylinder balancing via additive MFF intervention (7085) /* Manual reset of additive adaptation correction values
# ENDIF	
# IF (LC_FAC_TI_ER_BAL_RST_MAN = 1)	
RST_MAN	Cylinder balancing via multiple TI intervention (707Q) /* Manual reset of multiple correction values
# ENDIF	
# IF (LC_FAC_TI_AD_ER_BAL_RST_MAN = 1)	
RST_MAN	Cylinder balancing via multiple TI intervention (707Q) /* Manual reset of multiple adaptation correction values
# ENDIF	
# IF (LC_CYL_BAL_AD_EOL_RST = 1)	
RST_MAN	Cylinder balancing coordination at EOL (908O) /* Manual reset of cylinder balancing adaptation flags at end of line (EOL)
# ENDIF	
# IF (LC_CYL_BAL_AD_DC_RST = 1)	
RST_MAN	Cylinder balancing coordination at DC (908P) /* Manual reset of cylinder balancing adaptation flags at active driving cycle (DC)
# ENDIF	
RESET_DEAC (no system event , section located within SW code – only for info)	
RST_DEAC	Cylinder balancing coordination at EOL (908O - chapter 1.4, 1.5, 1.6) /* Initialization at function deactivation
RST_DEAC	Cylinder balancing coordination at DC (908P - chapter 1.2, 1.3, 1.4, 1.5, 1.6) /* Initialization at function deactivation
/* ** Description ** /* RST: --> ECU RESET	

```
/* RST_MAN:  -->  Initialization with use of a logical constant for setting of the wanted values
/* RST_DEAC:  --> Initialization at function deactivation
/* STOP_MAN:  --> Initialization at manual stop of function
/* EXIT_ST:  -->  Initialization at exit start
```

6.9.3 Recurring tasks(CYBL_ER):

SEG Task	
CYBL_ER	Cylinder balancing manager (Appl. Inc.) (907M)
CYBL_ER	Engine roughness signal preparation for Cylinder balancing (402U)
CYBL_ER	Cylinder balancing manager (907L)
CYBL_ER	Cylinder balancing via multiple TI intervention (707Q)
CYBL_ER	Cylinder balancing via additive MFF intervention (7085)
CYBL_ER	Coord. of the injection time correction factors for cylinder balancing (704P)
10ms - Task	
CYBL_ER	Cylinder balancing via ER (Appl. Inc.) (704N)
CYBL_ER	Coord. of the injection time correction factors for cylinder balancing (704P)
20ms - Task	
CYBL_ER	Cylinder balancing coordination at DC (908P)
CYBL_ER	Cylinder balancing coordination at EOL (908O)
100ms - Task	
CYBL_ER	Cylinder balancing via ER (Appl. Inc.) (704N)
CYBL_ER	Cylinder balancing via multiple TI intervention (707Q)
CYBL_ER	Cylinder balancing via additive MFF intervention (7085)
CYBL_ER	Coord. of the injection time correction factors for cylinder balancing (704P)
CYBL_ER	Cylinder balancing coordination at DC (908P)
CYBL_ER	Cylinder balancing coordination at EOL (908O)

/* *** Description ***

6.9.4 Main interfering AGGR tasks for CYBL_ER:

AGGR Task before CYBL_ER	
AGGR_xx	
ENSD	Engine Position and Speed Determination
ENRD	Engine Roughness Determination
FMSP	Fuel Mass Setpoint
LACO	Lambda Control
CILC	Cylinder Individual Lambda Control
AGGR_xx	
CYBL_ER Task	
CYBL_ER	Cylinder balancing via engine roughness
AGGR – Task after CYBL_ER	
AGGR_xx	
INJR	Injection Realisation
IGRE	Ignition Realisation
AGGR_xx	

/* *** Description ***

6.10 Cylinder balancing coordination at EOL

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CYL_BAL_LAM_AD_EOL	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation at end of line finished					
LV_CYL_BAL_ER_AD_FAC_EOL	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing multiple engine roughness adaptation at end of line finished					
LV_CYL_BAL_ER_AD_ADD_EOL	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing additive engine roughness adaptation at end of line finished					
LV_CYL_BAL_LAM_SEL_AD_COLD_EOL	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (cold) at end of line finished					
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (hot) at end of line finished					
STATE_CYL_BAL_ER_AD_EOL	V	0H 1H 2H	Ready Wait Adaptation	-	-
State of cylinder balancing engine roughness adaptation at end of line					
STATE_CYL_BAL_LAM_SEL_AD_EOL	V	0H 1H 2H	Ready Wait Adaptation	-	-
State of cylinder balancing cylinder selective lambda adaptation at end of line					
STATE_CYL_BAL_LAM_AD_EOL	V	0H 1H 2H	Ready Wait Adaptation	-	-
State of cylinder balancing lambda adaptation at end of line					
T_CYL_BAL_ER_AD	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing engine roughness adaptation at end of line					
T_CYL_BAL_LAM_SEL_AD	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing cylinder selective lambda adaptation at end of line					
T_CYL_BAL_LAM_AD	V	0…FFFFH	0…1310.7	0.02	s
Time for cylinder balancing lambda adaptation at end of line					
T_CYL_BAL_ER_AD_DYW	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing engine roughness adaptation at end of line within limits					
ER_STND_MMV_DIF_BAL_DYW	V	8000H…7FFFH	-325.78...325.77	0.00994	1/s^2
Dynamic window value for cylinder balancing engine roughness adaptation at end of line					
ER_STND_MMV_DIF_BAL_MAX_DYW	V	8000H…7FFFH	-325.78...325.77	0.00994	1/s^2
Maximum ER_STND_MMV_BAL value for dynamic window calculation at end of line					
ER_STND_MMV_DIF_BAL_MIN_DYW	V	8000H…7FFFH	-325.78...325.77	0.00994	1/s^2
Minimum ER_STND_MMV_BAL value for dynamic window calculation at end of line					
T_CYL_BAL_LAM_SEL_CBK_AD[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing cylinder selective lambda adaptation at end of line within limits					
LV_STATE_CYL_BAL_SEL_CHG_AD	V	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation to state adaptation requested					
LV_STATE_CYL_BAL_SEL_CHG_WAIT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation to state wait requested					
LV_CYL_BAL_LAM_SEL_AD_COLD[NC_CBK_EX_NR]	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (cold) exhaust bank specific					
LV_CYL_BAL_LAM_SEL_AD_RNG_L[NC_CBK_EX_NR]	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (low range) exhaust bank specific					
LV_CYL_BAL_LAM_SEL_AD_RNG_H[NC_CBK_EX_NR]	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (high range) exhaust bank specific					
CTR_CYL_BAL_LAM_CBK_AD[NC_CBK_EX_NR]	V	0…FFFFH	0…65535	1	-
Counter for cylinder balancing lambda adaptation at end of line within limits					

LV_STATE_CYL_BAL_LAM_CHG_AD	V	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation to state adaptation requested					
LV_STATE_CYL_BAL_LAM_CHG_WAIT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation to state wait requested					
LV_CYL_BAL_LAM_AD_FAC[NC_CBK_EX_NR]	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation (multiple) exhaust bank specific					
LV_CYL_BAL_LAM_AD_ADD[NC_CBK_EX_NR]	V/S/O	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation (additive) exhaust bank specific					
LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	-
Flag for waste gate open requested for cylinder balancing adaptation at end of line					
LV_CYL_BAL_AD_WG_OPEN_TMP_EOL[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Flag for temporary waste gate open request for cylinder balancing adaptation at end of line					

Input data:

FAC_LAM_CYL_SEL_ADJ_CST[NC_CYL_NR]	FAC_LAM_CYL_SEL_ADJ_L_RNG[NC_CYL_NR]	FAC_LAM_CYL_SEL_ADJ_H_RNG[NC_CYL_NR]	STATE_LAM_CYL_SEL_ADJ[NC_CBK_EX_NR]
FAC_CYL_LAM_ABSV_SUM[NC_CBK_EX_NR]	LV_LAM_CYL_ENA_CYL_BAL_DC[NC_CBK_EX_NR]	MFF_ADD_AD_ER_BAL[NC_CYL_NR]	FAC_TI_AD_ER_BAL[NC_CYL_NR]
STATE_TI_ER_BAL	ER_STND_MMV_DIF_BAL[NC_CYL_NR]	LV_MFF_ADD_AD_ER_BAL_ENA	LV_FAC_TI_AD_ER_BAL_ENA
MFF_ADD_LAM_AD[NC_CBK_EX_NR]	FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	LV_MFF_ADD_RNG_LAM_AD[NC_CBK_EX_NR]	LV_FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]
FAC_LAM_ADJ_LAM_AD[NC_CBK_EX_NR]	LV_LAM_AD_STOP[NC_CBK_EX_NR]	LV_TI_CYL_BAL_ER_ACT	LV_TI_CYL_BAL_LAM_ACT
N_32	T_AST_BAL	TCO	TCO_ST_BAL
LC_TCHA_CONF	NC_CYL_NR	NC_CBK_EX_NR	LV_CYL_BAL_AD_EOL_EXT_ADJ
DIST_KWP	LV_LAM_CYL_SEL_ADJ_VLD[NC_CBK_EX_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_ER_STND_MMV_DIF_BAL_DYW	6	0H…FFFFH	-325.78...325.77	0.00994	1/s^2
LDP_N_32_IP_ER_STND_MMV_DIF_DYW	6	0...FFH	0...8160	32	rpm
LDP_DIST_KWP_IP_ER_STND_DIF_DYW	6	0...FFFFH	0...524280	8	km
Dynamic window threshold for cylinder balancing engine roughness adaptation within limits at end of line					
IP_T_AST_MAX_CYL_BAL_WG_OPEN	4	0…FFFFH	0…6553.5	0.1	s
LDP_TCO_ST_BAL_IP_T_AST_WG_OPEN	4	0...FEH	-48...142.5	0.75	°C
Maximum time after start for cylinder balancing waste gate open request at end of line					
IP_FAC_CYL_LAM_ABSV_SUM_CYL_BAL	4	0...7FFFH	0...49.998474	1.5259E-3	%
LDP_DIST_KWP_IP_FAC_CYL_LAM_BAL	4	0...FFFFH	0...524280	8	km
Threshold for cylinder balancing cylinder selective lambda adaptation within limits at end of line					
IP_FAC_L_LAM_ADJ_DIF_CYL_BAL	4	0...FFFFH	-50...49.998474	1.5259E-3	%
LDP_DIST_KWP_IP_FAC_L_LAM_BAL	4	0...FFFFH	0...524280	8	km
Lambda factor threshold for cylinder balancing lambda adaptation within limits at end of line					
IP_FAC_MFF_LAM_ADJ_DIF_CYL_BAL	4	0...FFFFH	-50...49.998474	1.5259E-3	%
LDP_DIST_KWP_IP_FAC_MFF_LAM_BAL	4	0...FFFFH	0...524280	8	km
Lambda additive threshold for cylinder balancing lambda adaptation within limits at end of line					
C_T_MAX_CYL_BAL_ER_AD	1	0…FFFFH	0…6553.5	0.1	s
Maximum time for cylinder balancing engine roughness adaptation at end of line					
C_T_MIN_CYL_BAL_ER_AD_DYN	1	0…FFFFH	0…6553.5	0.1	s
Minimum time for cylinder balancing engine roughness adaptation within limits at end of line					
C_T_DEC_CYL_BAL_ER_AD_DYN	1	0…FFFFH	0…6553.5	0.1	s

Time decrement value for cylinder balancing engine roughness adaptation within limits at end of line					
C_T_MAX_CYL_BAL_LAM_SEL_AD_COLD	1	0…FFFFH	0…6553.5	0.1	s
Maximum time for cylinder balancing cylinder selective lambda adaptation (cold) at end of line					
C_T_MAX_CYL_BAL_LAM_SEL_AD_HOT	1	0…FFFFH	0…6553.5	0.1	s
Maximum time for cylinder balancing cylinder selective lambda adaptation (hot) at end of line					
C_T_MIN_CYL_BAL_LAM_SEL_AD_COLD	1	0…FFFFH	0…6553.5	0.1	s
Minimum time for cylinder balancing cylinder selective lambda adaptation (cold) within limits at end of line					
C_T_MIN_CYL_BAL_LAM_SEL_AD_HOT	1	0…FFFFH	0…6553.5	0.1	s
Minimum time for cylinder balancing cylinder selective lambda adaptation (hot) within limits at end of line					
C_T_DEC_CYL_BAL_LAM_SEL_CBK_AD	1	0…FFFFH	0…6553.5	0.1	s
Time decrement value for cylinder balancing cylinder selectiv lambda adaptation within limits at end of line					
C_T_MAX_CYL_BAL_LAM_AD	1	0…FFFFH	0…1310.7	0.02	s
Maximum time for cylinder balancing lambda adaptation at end of line					
C_CTR_MIN_CYL_BAL_LAM_AD_FAC	1	0…FFFFH	0…65535	1	-
Minimum counter value for cylinder balancing lambda adaptation (multiple) within limits at end of line					
C_CTR_MIN_CYL_BAL_LAM_AD_ADD	1	0…FFFFH	0…65535	1	-
Minimum counter value for cylinder balancing lambda adaptation (additive) within limits at end of line					
C_CTR_DEC_CYL_BAL_LAM_CBK_AD	1	0…FFFFH	0…65535	1	-
Counter decrement value for cylinder balancing lambda adaptation within limits at end of line					
C_TCO_MAX_CYL_BAL_WG_OPEN	1	0...FEH	-48...142.5	0.75	°C
Maximum coolant temperature for cylinder balancing waste gate open request at end of line					
LC_CYL_BAL_AD_EOL_RST	1	0...1H	0...1	1	-
Manual switch for cylinder balancing adaptation reset at end of line					
C_CYL_BAL_AD_EOL_RST	1	0H 1H 2H 3H 4H 5H 6H 7H	ZERO ON LAM_AD SEL_COLD SEL_HOT ER_FAC ER_ADD WG_OPEN	-	-
Reset of cylinder balancing adaptation readiness flags at end of line					
LC_CYL_BAL_CORD_EOL_STOP_MAN	1	0...1H	0...1	1	-
Logical constant for manual stop (=1) of cylinder balancing coordination at end of line					
C_CYL_BAL_LAM_AD_EOL_CONF	1	0H 1H 2H 3H	ZERO ADD FAC_L ADD_FAC_L	-	-
Configuration of lambda adaptation correction values for cylinder balancing at end of line					
C_CYL_BAL_LAM_SEL_HOT_EOL_CONF	1	0H 1H 2H 3H	ZERO LOW HIGH LOW_HIGH	-	-
Configuration of cylinder selectiv lambda adaptation correction values (hot) for cylinder balancing at end of line					

FUNCTION DESCRIPTION:

General information:

Task of the cylinder balancing coordination is to observe the balancing adaptation functions at end of line test and to set a readiness information as soon as a defined adaptation limit for each adaptation has been reached.

Application conditions:

Initialization at first Engine run or NVMY-error:

LV_CYL_BAL_LAM_AD_EOL = 0 *(first initialization)*

LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 0 *(first initialization)*

LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 0 *(first initialization)*

```
    LV_CYL_BAL_ER_AD_FAC_EOL = 0                                    (first initialization)
    LV_CYL_BAL_ER_AD_ADD_EOL = 0                                    (first initialization)
    FOR m = 0 to NC_CBK_EX_NR – 1
        LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 0                           (first initialization)
        LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 0                          (first initialization)
        LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 0                          (first initialization)
        LV_CYL_BAL_LAM_AD_FAC[m] = 0                                (first initialization)
        LV_CYL_BAL_LAM_AD_ADD[m] = 0                                (first initialization)
    END FOR
```

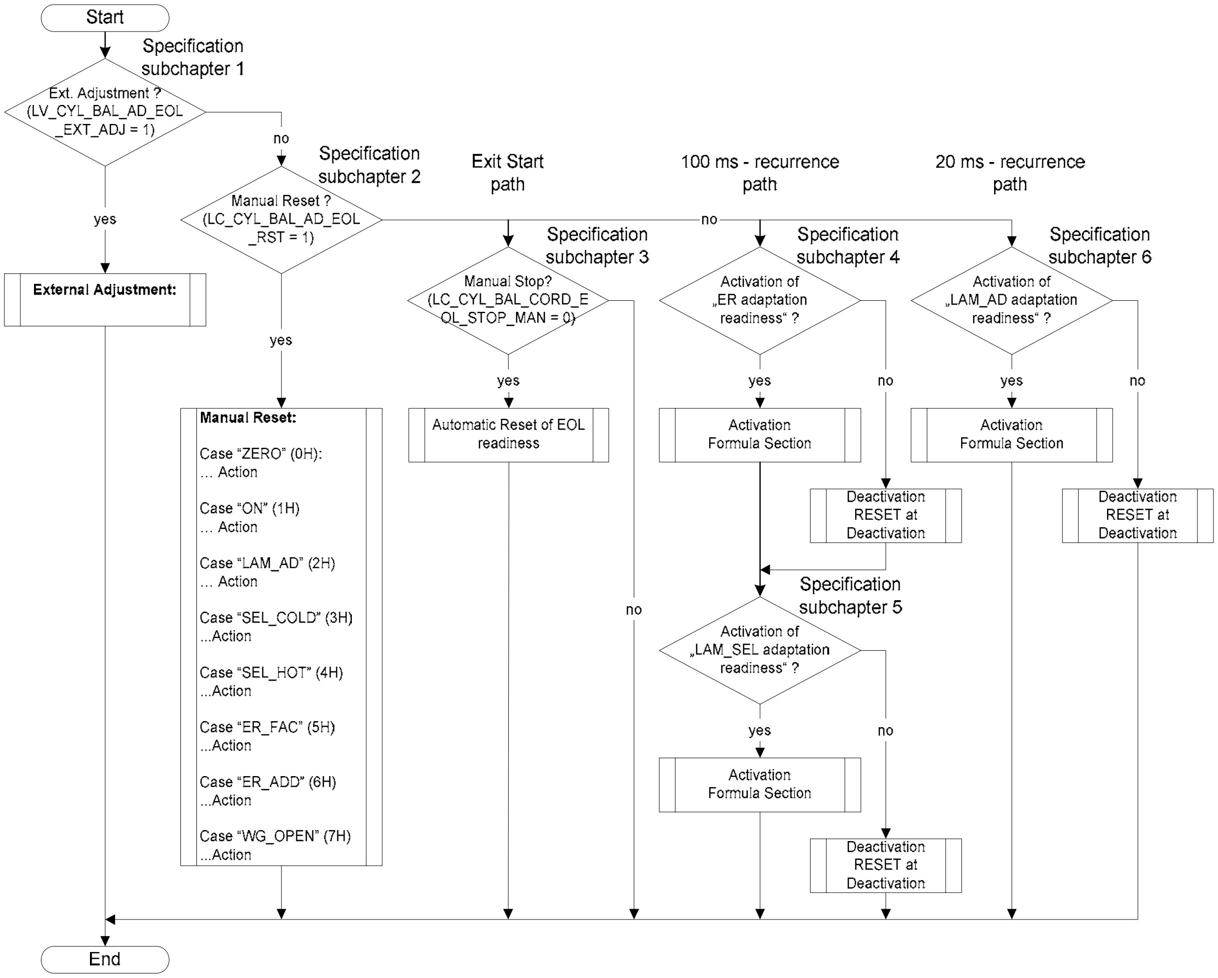
Initialization at Reset:

```
LV_CYL_BAL_LAM_AD_EOL = LV_CYL_BAL_LAM_AD_EOL                                         (NVMY)
LV_CYL_BAL_ER_AD_FAC_EOL = LV_CYL_BAL_ER_AD_FAC_EOL                                   (NVMY)
LV_CYL_BAL_ER_AD_ADD_EOL = LV_CYL_BAL_ER_AD_ADD_EOL                                   (NVMY)
LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = LV_CYL_BAL_LAM_SEL_AD_COLD_EOL                       (NVMY)
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = LV_CYL_BAL_LAM_SEL_AD_HOT_EOL                         (NVMY)
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_SEL_AD_COLD[m] = LV_CYL_BAL_LAM_SEL_AD_COLD[m]                     (NVMY)
    LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = LV_CYL_BAL_LAM_SEL_AD_RNG_L[m]                   (NVMY)
    LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = LV_CYL_BAL_LAM_SEL_AD_RNG_H[m]                   (NVMY)
    LV_CYL_BAL_LAM_AD_FAC[m] = LV_CYL_BAL_LAM_AD_FAC[m]                               (NVMY)
    LV_CYL_BAL_LAM_AD_ADD[m] = LV_CYL_BAL_LAM_AD_ADD[m]                               (NVMY)
END FOR
```

(NVMY: values stored in the non-volatile memory)

Recurrence:	See separate chapters
Activation:	See separate chapters
Deactivation:	See separate chapters

Signal flow diagram:



6.10.1 External adjustment of cylinder balancing readiness at end of line (EOL):

FUNCTION DESCRIPTION:

General information:

In case of an external adjustment (request from key word protocoll) the readiness bits for all cylinder balancing adaptation functions at EOL are set to zero. While the external adjustment is activated, all following subchapters within the module "Cylinder balancing coordination at EOL" are deactivated and the corresponding "Initialization at Deactivation" process for this chapters is blocked during the period.

Application conditions:

Recurrence: 100 ms

Activation: LV_CYL_BAL_AD_EOL_EXT_ADJ = 1

Deactivation: LV_CYL_BAL_AD_EOL_EXT_ADJ = 0

Formula section:

```
LV_CYL_BAL_LAM_AD_EOL = 0
LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 0
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 0
LV_CYL_BAL_ER_AD_FAC_EOL = 0
LV_CYL_BAL_ER_AD_ADD_EOL = 0
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 0
    LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 0
    LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 0
    LV_CYL_BAL_LAM_AD_FAC[m] = 0
    LV_CYL_BAL_LAM_AD_ADD[m] = 0
END FOR
```

6.10.2 Manual reset of cylinder balancing readiness at end of line (EOL):

FUNCTION DESCRIPTION:

General information:

In case of a manual reset it is possible to adjust the exported output data to a wanted value to make the function test and handling more easier. While the manual function reset is activated, all following subchapters within the module "Cylinder balancing coordination at EOL" are deactivated and the corresponding "Initialization at Deactivation" process for this chapters is blocked during the period.

Application conditions:

Recurrence: 100 ms

Activation: LV_CYL_BAL_AD_EOL_EXT_ADJ = 0 **and**
LC_CYL_BAL_AD_EOL_RST = 1

Deactivation: LV_CYL_BAL_AD_EOL_EXT_ADJ = 0 **and**
[LC_CYL_BAL_AD_EOL_RST = 0 **]**

Formula section:

```
IF      C_CYL_BAL_AD_EOL_RST = "ZERO" (0H)
THEN    LV_CYL_BAL_LAM_AD_EOL = 0
        LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 0
        LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 0
        LV_CYL_BAL_ER_AD_FAC_EOL = 0
        LV_CYL_BAL_ER_AD_ADD_EOL = 0
```

```
                FOR m = 0 to NC_CBK_EX_NR – 1
                        LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 0
                        LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 0
                        LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 0
                        LV_CYL_BAL_LAM_AD_FAC[m] = 0
                        LV_CYL_BAL_LAM_AD_ADD[m] = 0
                        LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[m] = 0
                END FOR

ELSEIF          C_CYL_BAL_AD_EOL_RST = “ON” (1H)
THEN            LV_CYL_BAL_LAM_AD_EOL = 1
                LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 1
                LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1
                LV_CYL_BAL_ER_AD_FAC_EOL = 1
                LV_CYL_BAL_ER_AD_ADD_EOL = 1
                FOR m = 0 to NC_CBK_EX_NR – 1
                        LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 1
                        LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 1
                        LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 1
                        LV_CYL_BAL_LAM_AD_FAC[m] = 1
                        LV_CYL_BAL_LAM_AD_ADD[m] = 1
                        LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[m] = 1
                END FOR

ELSEIF          C_CYL_BAL_AD_EOL_RST = “LAM_AD” (2H)
THEN            LV_CYL_BAL_LAM_AD_EOL = 1

ELSEIF          C_CYL_BAL_AD_EOL_RST = “SEL_COLD” (3H)
THEN            LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 1

ELSEIF          C_CYL_BAL_AD_EOL_RST = “SEL_HOT” (4H)
THEN            LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1

ELSEIF          C_CYL_BAL_AD_EOL_RST = “ER_FAC” (5H)
THEN            LV_CYL_BAL_ER_AD_FAC_EOL = 1

ELSEIF          C_CYL_BAL_AD_EOL_RST = “ER_ADD” (6H)
THEN            LV_CYL_BAL_ER_AD_ADD_EOL = 1
```

```
ELSEIF      C_CYL_BAL_AD_EOL_RST = “WG_OPEN” (7H)
THEN
    FOR m = 0 to NC_CBK_EX_NR – 1
            LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[m] = 1
    END FOR
ENDIF
```

6.10.3 Automatic reset of cylinder balancing end of line (EOL) readiness:

FUNCTION DESCRIPTION:

General information:

In case of lost adaptation values due to a manual reset at the garage or faulty non-volatile memory entries, the system detects the changes and the corresponding end of line readiness is reset automatically.

Application conditions:

Initialization at exit start (EXIT_ST):

```
IF          LC_CYL_BAL_CORD_EOL_STOP_MAN = 0          and
            LC_CYL_BAL_AD_EOL_RST = 0                 and
            LV_CYL_BAL_AD_EOL_EXT_ADJ = 0
THEN
            Reset of cylinder balancing global lambda adaptation readiness:
            IF (1)      LV_CYL_BAL_LAM_AD_EOL = 1
            THEN (1)
                    IF (2)    ( C_CYL_BAL_LAM_AD_EOL_CONF = “FAC_L” (2H)   or
                                C_CYL_BAL_LAM_AD_EOL_CONF = “ADD_FAC_L” (3H) )   and
                              $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} |\mathrm{FAC\_L\_RNG\_LAM\_AD[m]}| = 0$
                    THEN  (2)     LV_CYL_BAL_LAM_AD_EOL = 0
                              FOR m = 0 to NC_CBK_EX_NR – 1
                                      LV_CYL_BAL_LAM_AD_FAC[m] = 0
                              END FOR
                    ENDIF (2)
                    IF (2)    ( C_CYL_BAL_LAM_AD_EOL_CONF = “ADD” (1H)       or
                                C_CYL_BAL_LAM_AD_EOL_CONF = “ADD_FAC_L” (3H) )   and
                              $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} |\mathrm{MFF\_ADD\_LAM\_AD[m]}| = 0$
                    THEN  (2)     LV_CYL_BAL_LAM_AD_EOL = 0
                              FOR m = 0 to NC_CBK_EX_NR – 1
```

```
                    LV_CYL_BAL_LAM_AD_ADD[m] = 0
                END FOR
        ENDIF (2)
ENDIF (1)
```

Reset of cylinder balancing lambda selective (cold condition) adaptation readiness:

```
IF (1)      LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 1
THEN (1)
        IF (2)      $\sum_{i=0}^{i=NC\_CYL\_NR-1} |FAC\_LAM\_CYL\_SEL\_ADJ\_CST[i]| = 0$
        THEN  (2)       LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 0
                FOR m = 0 to NC_CBK_EX_NR – 1
                    LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 0
                END FOR
        ENDIF (2)
ENDIF (1)
```

Reset of cylinder balancing lambda selective (hot condition) adaptation readiness:

```
IF (1)      LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1
THEN (1)
        IF (2)  ( C_CYL_BAL_LAM_SEL_HOT_EOL_CONF = “LOW” (1H) or
                  C_CYL_BAL_LAM_SEL_HOT_EOL_CONF = “LOW_HIGH” (3H) )  and
                $\sum_{i=0}^{i=NC\_CYL\_NR-1} |FAC\_LAM\_CYL\_SEL\_ADJ\_L\_RNG[i]| = 0$
        THEN  (2)       LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 0
                FOR m = 0 to NC_CBK_EX_NR – 1
                    LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 0
                END FOR
        ENDIF (2)
        IF (2)  ( C_CYL_BAL_LAM_SEL_HOT_EOL_CONF = “HIGH” (2H) or
                  C_CYL_BAL_LAM_SEL_HOT_EOL_CONF = “LOW_HIGH” (3H) )  and
                $\sum_{i=0}^{i=NC\_CYL\_NR-1} |FAC\_LAM\_CYL\_SEL\_ADJ\_H\_RNG[i]| = 0$
        THEN  (2)       LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 0
                FOR m = 0 to NC_CBK_EX_NR – 1
                    LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 0
                END FOR
        ENDIF (2)
ENDIF (1)
```

Reset of cylinder balancing multiple engine roughness adaptation readiness:

```
IF (1)        LV_CYL_BAL_ER_AD_FAC_EOL = 1
THEN (1)
    IF (2)        $\sum_{i=0}^{i=NC\_CYL\_NR-1} |FAC\_TI\_AD\_ER\_BAL[i]| = 0$
    THEN (2)      LV_CYL_BAL_ER_AD_FAC_EOL = 0
    ENDIF (2)
ENDIF (1)
```

Reset of cylinder balancing additive engine roughness adaptation readiness:

```
IF (1)        LV_CYL_BAL_ER_AD_ADD_EOL = 1
THEN (1)
    IF (2)        $\sum_{i=0}^{i=NC\_CYL\_NR-1} |MFF\_ADD\_AD\_ER\_BAL[i]| = 0$
    THEN (2)      LV_CYL_BAL_ER_AD_ADD_EOL = 0
    ENDIF (2)
ENDIF (1)
ENDIF
```

6.10.4 Calculation of cylinder balancing engine roughness adaptation readiness (EOL):

FUNCTION DESCRIPTION:

General information:

After activation of the corresponding adaptation function (multiple or additive) at stratified combustion mode, a timer is started to observe the filtered engine roughness values used as input for the adaptation controllers. As soon as all controller input values are within a defined range for a specific time interval, readiness is indicated. The function is controlled by a state machine.

Application conditions:

*Initialization at Reset **or** Deactivation*

STATE_CYL_BAL_ER_AD_EOL = "Ready" (0[**H**])
T_CYL_BAL_ER_AD = 0.0[s]
T_CYL_BAL_ER_AD_DYW = 0.0[s]
ER_STND_MMV_DIF_BAL_DYW = 0.0[1/s^2]
ER_STND_MMV_DIF_BAL_MIN_DYW = 0.0[1/s^2]
ER_STND_MMV_DIF_BAL_MAX_DYW = 0.0[1/s^2]

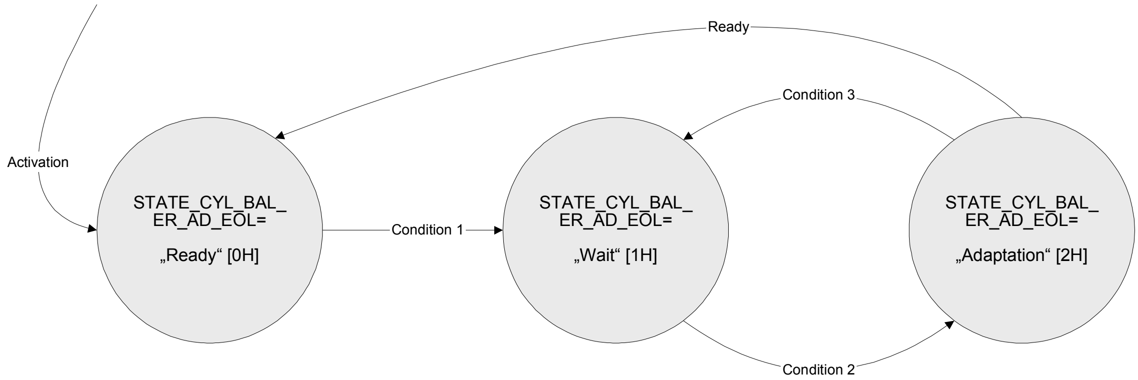
Recurrence: 100 ms

Activation:
LC_CYL_BAL_AD_EOL_RST = 0 **and**
LV_CYL_BAL_AD_EOL_EXT_ADJ = 0 **and**
LC_CYL_BAL_CORD_EOL_STOP_MAN = 0 **and**

LC_TCHA_CONF = 0 **and**
LV_TI_CYL_BAL_ER_ACT = 1 **and**
(LV_CYL_BAL_ER_AD_FAC_EOL = 0 **or**
LV_CYL_BAL_ER_AD_ADD_EOL = 0 **)**

Deactivation: LC_CYL_BAL_AD_EOL_RST = 0 **and**
LV_CYL_BAL_AD_EOL_EXT_ADJ = 0 **and**
[LC_CYL_BAL_CORD_EOL_STOP_MAN = 1 **or**
LC_TCHA_CONF = 1 **or**
LV_TI_CYL_BAL_ER_ACT = 0 **or**
(LV_CYL_BAL_ER_AD_FAC_EOL = 1 **and**
LV_CYL_BAL_ER_AD_ADD_EOL = 1 **)** **]**

Signal flow diagram:



Formula section:

Note: At function activation always State "Ready" is entered

Cylinder balancing engine roughness adaptation at EOL: State 0[H] – “Ready”

Assignments at entry to state “Ready”:

STATE_CYL_BAL_ER_AD_EOL = "Ready" (0[**H**])

Actions during state "Ready":

Transition to State 1[H] “Wait” (Condition 1)

// at least one cylinder balancing engine roughness adaptation at EOL not yet finished

IF **(** LV_CYL_BAL_ER_AD_FAC_EOL = 0 **or** LV_CYL_BAL_ER_AD_ADD_EOL = 0 **)**

Assignments at exit from state “Ready”:

No assignments

Cylinder balancing engine roughness adaptation at EOL: State 1[H] – “Wait”

Assignments at entry to state “Wait”:

STATE_CYL_BAL_ER_AD_EOL = "Wait" (1[**H**])

Actions during state "Wait":

Transition to State 2[H] “Adaptation” (Condition 2)

```
// conditions for cylinder balancing engine roughness adaptation achieved (multiple or additive)
IF          ( STATE_TI_ER_BAL = “Cylinder Balancing” (3[H])      and
              LV_FAC_TI_AD_ER_BAL_ENA = 1                          and
              LV_CYL_BAL_ER_AD_FAC_EOL = 0 )
            OR
            ( STATE_TI_ER_BAL = “Cylinder Balancing” (3[H])      and
              LV_MFF_ADD_AD_ER_BAL_ENA = 1                         and
              LV_CYL_BAL_ER_AD_ADD_EOL = 0 )
```

Assignments at exit from state “Wait”:

T_CYL_BAL_ER_AD = 0.0[s]

T_CYL_BAL_ER_AD_DYW = 0.0[s]

ER_STND_MMV_DIF_BAL_MAX_DYW = 0.0[1/s^2]

ER_STND_MMV_DIF_BAL_MIN_DYW = 0.0[1/s^2]

Cylinder balancing engine roughness adaptation at EOL: State 2[H] – “Adaptation”

Assignments at entry to state “Adaptation”:

STATE_CYL_BAL_ER_AD_EOL = "Adaptation" (2[**H**])

Actions during state "Adaptation":

Calculations:

```
// incrementation of cylinder balancing engine roughness adaptation time with 0.1[s]
T_CYL_BAL_ER_AD(n) = T_CYL_BAL_ER_AD(n-1) + 0.1[s]
// determination of MIN- and MAX-dynamic window values for overall  filtered engine roughness values
FOR i = 0 to NC_CYL_NR - 1
    IF          ER_STND_MMV_DIF_BAL[i] > ER_STND_MMV_DIF_BAL_MAX_DYW
    THEN        ER_STND_MMV_DIF_BAL_MAX_DYW = ER_STND_MMV_DIF_BAL[i]
    ELSEIF      ER_STND_MMV_DIF_BAL[i] < ER_STND_MMV_DIF_BAL_MIN_DYW
    THEN        ER_STND_MMV_DIF_BAL_MIN_DYW = ER_STND_MMV_DIF_BAL[i]
    ENDIF
 END FOR
// calculation of dynamic window based on MIN- and MAX-dynamic window values
ER_STND_MMV_DIF_BAL_DYW = ER_STND_MMV_DIF_BAL_MAX_DYW -
                ER_STND_MMV_DIF_BAL_MIN_DYW
// calculation of cylinder balancing engine roughness adaptation time within dynamic window
IF          ER_STND_MMV_DIF_BAL_DYW < IP_ER_STND_MMV_DIF_BAL_DYW
THEN        T_CYL_BAL_ER_AD_DYW(n) = T_CYL_BAL_ER_AD_DYW(n-1) + 0.1[s]
ELSE        T_CYL_BAL_ER_AD_DYW(n) = T_CYL_BAL_ER_AD_DYW(n-1) –
                C_T_DEC_CYL_BAL_ER_AD_DYN
```

```
            // counter value limited to zero, no timer underrun possible
ENDIF
// calculation of cylinder balancing engine roughness adaptation readiness at end of line
IF (1)          T_CYL_BAL_ER_AD ≥ C_T_MAX_CYL_BAL_ER_AD                    or
                T_CYL_BAL_ER_AD_DYW ≥ C_T_MIN_CYL_BAL_ER_AD_DYN
THEN (1)
    IF (2)          LV_FAC_TI_AD_ER_BAL_ENA = 1
    THEN (2)        LV_CYL_BAL_ER_AD_FAC_EOL = 1
    ELSE (2)        LV_CYL_BAL_ER_AD_ADD_EOL =1
    ENDIF(2)
ENDIF (1)
// initialization of MIN- and MAX-dynamic window values for next dynamic window calculation
ER_STND_MMV_DIF_BAL_MAX_DYW = 0.0[1/s^2]
ER_STND_MMV_DIF_BAL_MIN_DYW = 0.0[1/s^2]
Transition to State 0[H] "Ready" (Ready)
// cylinder balancing engine roughness adaptation at EOL finished
IF              ( LV_CYL_BAL_ER_AD_FAC_EOL = 1 and LV_CYL_BAL_ER_AD_ADD_EOL = 1 )
Transition to State 1[H] "Wait" (Condition 3)
// conditions for cylinder balancing engine roughness adaptation not achieved
IF              STATE_TI_ER_BAL ≠ "Cylinder Balancing" (3[H])
                OR
                ( LV_FAC_TI_AD_ER_BAL_ENA = 0 and LV_MFF_ADD_AD_ER_BAL_ENA = 0 )
                OR
                ( LV_FAC_TI_AD_ER_BAL_ENA = 1 and LV_CYL_BAL_ER_AD_FAC_EOL = 1 )
                OR
                ( LV_MFF_ADD_AD_ER_BAL_ENA = 1  and LV_CYL_BAL_ER_AD_ADD_EOL = 1 )
```

Assignments at exit from state "Adaptation":

No assignments

6.10.5 Calculation of cylinder balancing selective lambda adaptation readiness (EOL):

FUNCTION DESCRIPTION:

General information:

After activation of the corresponding adaptation function (hot or cold) at homogenous combustion mode, a timer is started to observe the characteristic quantity of estimated cylinder individual lambda deviation. As soon as the deviation values are within a defined range for a specific time interval, readiness is indicated. The function is controlled by a state machine.

Application conditions:

*Initialization at Reset **or** Deactivation*

```
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_STATE_CYL_BAL_SEL_CHG_WAIT[m] = 0
    LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[m] = 0
    LV_CYL_BAL_AD_WG_OPEN_TMP_EOL[m] = 0
    T_CYL_BAL_LAM_SEL_CBK_AD[m] = 0.0[s]
END FOR
T_CYL_BAL_LAM_SEL_AD = 0.0[s]
LV_STATE_CYL_BAL_SEL_CHG_AD = 0
STATE_CYL_BAL_LAM_SEL_AD_EOL = "Ready" (0[H])
```

Recurrence: 100 ms

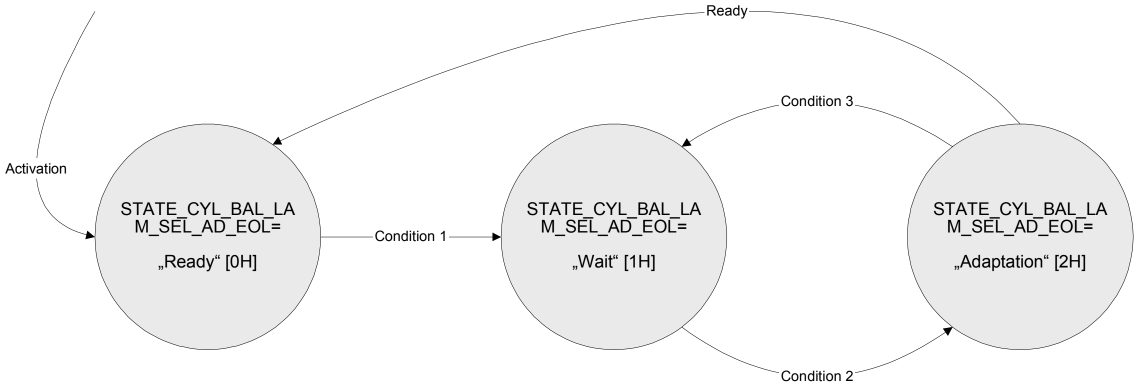
Activation:

LC_CYL_BAL_AD_EOL_RST = 0 **and**
LV_CYL_BAL_AD_EOL_EXT_ADJ = 0 **and**
LC_CYL_BAL_CORD_EOL_STOP_MAN = 0 **and**
LV_TI_CYL_BAL_LAM_ACT = 1 **and**
(LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 0 **or**
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 0 **)**

Deactivation:

LC_CYL_BAL_AD_EOL_RST = 0 **and**
LV_CYL_BAL_AD_EOL_EXT_ADJ = 0 **and**
[LC_CYL_BAL_CORD_EOL_STOP_MAN = 1 **or**
LV_TI_CYL_BAL_LAM_ACT = 0 **or**
(LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 1 **and**
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1 **)** **]**

Signal flow diagram:



Formula section:

// calculation of request for waste gate open for turbo engine (LC_TCHA_CONF=1)

```
IF      LC_TCHA_CONF=1
THEN    // for turbo engine
```

```
        FOR m = 0 to NC_CBK_EX_NR – 1
            IF      LV_LAM_CYL_ENA_CYL_BAL_DC[m] = 1
                    AND
                    T_AST_BAL < IP_T_AST_MAX_CYL_BAL_WG_OPEN
                    AND
                    ( [ TCO < C_TCO_MAX_CYL_BAL_WG_OPEN                  and
                        LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 0                 ]
                      OR
                      [ TCO > C_TCO_MAX_CYL_BAL_WG_OPEN                  and
                        { LV_CYL_BAL_LAM_SEL_AD_RNG_L[m]  = 0      or
                          LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 0       }      ] )
            THEN    LV_CYL_BAL_AD_WG_OPEN_TMP_EOL[m] = 1
            ELSE    LV_CYL_BAL_AD_WG_OPEN_TMP_EOL[m] = 0
            ENDIF
        END FOR
        FOR m = 0 to NC_CBK_EX_NR – 1
            IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_AD\_WG\_OPEN\_TMP\_EOL[m]} > 0$
            THEN    LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[m] = 1
            ELSE    LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[m] = 0
            ENDIF
        END FOR
    ELSE        // for non turbo engine
        FOR m = 0 to NC_CBK_EX_NR – 1
            LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[m]= 0
        END FOR
    ENDIF
```

// calculation of state machine cylinder balancing selective lambda adaptation readiness at EOL

Note: At function activation always State "Ready" is entered

Cylinder balancing selective lambda adaptation at EOL: State 0[H] – “Ready”

Assignments at entry to state “Ready”:

STATE_CYL_BAL_LAM_SEL_AD_EOL = "Ready" (0[**H**])

Actions during state "Ready":

Transition to State 1[H] “Wait” (Condition 1)

// at least one cylinder balancing selective lambda adaptation at EOL not yet finished

```
IF          LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 0              or
            LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 0
```

Assignments at exit from state “Ready”:

No assignments

Cylinder balancing selective lambda adaptation at EOL: State 1[H] – “Wait”

Assignments at entry to state “Wait”:

STATE_CYL_BAL_LAM_SEL_AD_EOL = "Wait" (1[**H**])

Actions during state "Wait":

```
// calculation of logical value to change state to adaptation
FOR m = 0 to NC_CBK_EX_NR – 1
    IF      ( STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_ENG_COLD” (2[H])          and
              LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 0 )
            OR
            ( STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_NOM_L_RNG” (4[H])         and
              LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 0 )
            OR
            ( STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_NOM_H_RNG” (5[H])         and
              LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 0 )
    THEN    LV_STATE_CYL_BAL_SEL_CHG_AD = 1
    ENDIF
END FOR
```

Transition to State 2[H] “Adaptation” (Condition 2)

```
// conditions for cylinder selective lambda adaptation achieved (cold or hot)
IF          LV_STATE_CYL_BAL_SEL_CHG_AD = 1
```

Assignments at exit from state “Wait”:

```
T_CYL_BAL_LAM_SEL_AD = 0.0[s]
LV_STATE_CYL_BAL_SEL_CHG_AD = 0
FOR m = 0 to NC_CBK_EX_NR – 1
    T_CYL_BAL_LAM_SEL_CBK_AD[m] = 0.0[s]
END FOR
```

Cylinder balancing selective lambda adaptation at EOL: State 2[H] – “Adaptation”

Assignments at entry to state “Adaptation”:

STATE_CYL_BAL_LAM_SEL_AD_EOL = "Adaptation" (2[**H**])

Actions during state "Adaptation":

Calculations:

```
// calculation of logical value to change state to wait
FOR m = 0 to NC_CBK_EX_NR – 1
    IF      ( STATE_LAM_CYL_SEL_ADJ[m] <> “ADJ_ENG_COLD” (2[H])and
```

```
                  STATE_LAM_CYL_SEL_ADJ[m] <> “ADJ_NOM_L_RNG” (4[H])          and
                  STATE_LAM_CYL_SEL_ADJ[m] <> “ADJ_NOM_H_RNG” (5[H]) )
                OR
                ( STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_ENG_COLD” (2[H])            and
                  LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 1 )
                OR
                ( STATE_LAM_CYL_SEL_ADJ[m] = "ADJ_NOM_L_RNG” (4[H])           and
                  LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 1 )
                OR
                ( STATE_LAM_CYL_SEL_ADJ[m] = "ADJ_NOM_H_RNG” (5[H])           and
                  LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 1 )
        THEN    LV_STATE_CYL_BAL_SEL_CHG_WAIT[m] = 1
                T_CYL_BAL_LAM_SEL_CBK_AD[m] = 0.0[s]
        ELSE    LV_STATE_CYL_BAL_SEL_CHG_WAIT[m] = 0
        ENDIF
    END FOR

    // incrementation of cylinder selective lambda adaptation time with 0.1[s]
    T_CYL_BAL_LAM_SEL_AD(n) = T_CYL_BAL_LAM_SEL_AD(n-1) + 0.1[s]

    // determination of characteristic quantity of estimated cylinder individual lambda deviation values
    FOR (1) m = 0 to NC_CBK_EX_NR – 1
        IF (2)  LV_LAM_CYL_SEL_ADJ_VLD[m] = 1
        THEN (2)
                IF      FAC_CYL_LAM_ABSV_SUM[m] < IP_FAC_CYL_LAM_ABSV_SUM_CYL_BAL
                THEN    T_CYL_BAL_LAM_SEL_CBK_AD[m](n) =
                                T_CYL_BAL_LAM_SEL_CBK_AD[m](n-1) + 0.1[s]
                                // timer incremantation with 0.1[s]
                ELSE    T_CYL_BAL_LAM_SEL_CBK_AD[m](n) =
                                T_CYL_BAL_LAM_SEL_CBK_AD[m](n-1) –
                                C_T_DEC_CYL_BAL_LAM_SEL_CBK_AD
                                // timer value limited to zero, no timer underrun possible
                ENDIF
        ENDIF (2)
    END FOR (1)

    // calculation of cylinder selective lambda adaptation readiness at end of line (exhaust bank specific)
    FOR m = 0 to NC_CBK_EX_NR – 1
        IF (1a)         STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_ENG_COLD” (2[H])              and
                        LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 0
        THEN (1a)
```

```
            IF      T_CYL_BAL_LAM_SEL_CBK_AD[m] >
                        C_T_MIN_CYL_BAL_LAM_SEL_AD_COLD                 or
                    T_CYL_BAL_LAM_SEL_AD >
                        C_T_MAX_CYL_BAL_LAM_SEL_AD_COLD
            THEN    LV_CYL_BAL_LAM_SEL_AD_COLD[m] = 1
            ENDIF
        ELSEIF (1b) STATE_LAM_CYL_SEL_ADJ[m] = "ADJ_NOM_L_RNG” (4[H])          and
                    LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 0
        THEN (1b)
            IF      T_CYL_BAL_LAM_SEL_CBK_AD[m] >
                        C_T_MIN_CYL_BAL_LAM_SEL_AD_HOT                  or
                    T_CYL_BAL_LAM_SEL_AD >
                        C_T_MAX_CYL_BAL_LAM_SEL_AD_HOT
            THEN    LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 1
            ENDIF
        ELSEIF (1c) STATE_LAM_CYL_SEL_ADJ[m] = "ADJ_NOM_H_RNG” (5[H])          and
                    LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 0
        THEN (1c)
            IF      T_CYL_BAL_LAM_SEL_CBK_AD[m] >
                        C_T_MIN_CYL_BAL_LAM_SEL_AD_HOT                  or
                    T_CYL_BAL_LAM_SEL_AD >
                        C_T_MAX_CYL_BAL_LAM_SEL_AD_HOT
            THEN    LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 1
            ENDIF
        ENDIF (1)
    END FOR
```

```
    // calculation of cylinder selective lambda adaptation readiness (hot condition) at end of line (global)
    IF (1a)             C_CYL_BAL_LAM_SEL_HOT_EOL_CONF = “ZERO” (0H)
    THEN (1a)
        FOR m = 0 to NC_CBK_EX_NR – 1
            LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 1
            LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 1
        END FOR
        LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1
        // adaptation of all exhaust banks finished at EOL
    ELSEIF (1b)         C_CYL_BAL_LAM_SEL_HOT_EOL_CONF = “LOW” (1H)
    THEN (1b)
        FOR m = 0 to NC_CBK_EX_NR – 1
```

```
                LV_CYL_BAL_LAM_SEL_AD_RNG_H[m] = 1
        END FOR
        IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_RNG\_L[m]} = \mathrm{NC\_CBK\_EX\_NR}$
        THEN            LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1
                        // adaptation of all exhaust banks (low range) are finished at EOL
        ENDIF
    ELSEIF (1c)         C_CYL_BAL_LAM_SEL_HOT_EOL_CONF = “HIGH” (2H)
    THEN (1c)
        FOR m = 0 to NC_CBK_EX_NR – 1
                LV_CYL_BAL_LAM_SEL_AD_RNG_L[m] = 1
        END FOR
        IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_RNG\_H[m]} = \mathrm{NC\_CBK\_EX\_NR}$
        THEN            LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1
                        // adaptation of all exhaust banks (high range) are finished at EOL
        ENDIF
    ELSEIF (1d)         C_CYL_BAL_LAM_SEL_HOT_EOL_CONF = “LOW_HIGH” (3H)
    THEN (1d)
        IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_RNG\_L[m]} = \mathrm{NC\_CBK\_EX\_NR}$
                and
                $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_RNG\_H[m]} = \mathrm{NC\_CBK\_EX\_NR}$
        THEN            LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1
                        // ad. of all exhaust banks (low  and high range) are finished at EOL
        ENDIF
    ENDIF (1)
```

// calculation of cylinder selective lambda adaptation readiness (cold condition) at end of line (global)

```
    IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_COLD[m]} = \mathrm{NC\_CBK\_EX\_NR}$
    THEN        LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 1
                // cold adaptation of all exhaust banks are finished at end of line
    ENDIF
```

Transition to State 0[H] "Ready” (Ready)

// cylinder selective lambda adaptation at EOL finished

```
    IF          LV_CYL_BAL_LAM_SEL_AD_COLD_EOL = 1                  and
                LV_CYL_BAL_LAM_SEL_AD_HOT_EOL = 1
```

Transition to State 1[H] “Wait” (Condition 3)

// conditions for cylinder selective lambda adaptation not achieved

IF $$\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_STATE\_CYL\_BAL\_SEL\_CHG\_WAIT[m]} = \mathrm{NC\_CBK\_EX\_NR}$$

// adaptation condition for all exhaust banks are not fulfilled

Assignments at exit from state “Adaptation”:

```
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_STATE_CYL_BAL_SEL_CHG_WAIT[m] = 0
END FOR
```

6.10.6 Calculation of cylinder balancing lambda adaptation readiness (EOL):

FUNCTION DESCRIPTION:

General information:

After activation of the corresponding adaptation function at homogenous combustion mode, a timer is started to observe the deviation between the actual and the last learned lambda deviation value. As soon as the deviation values are within a defined range for a specific time interval, readiness is indicated. The function is controlled by a state machine.

Application conditions:

*Initialization at Reset **or** Deactivation*

```
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_STATE_CYL_BAL_LAM_CHG_WAIT[m] = 0
    CTR_CYL_BAL_LAM_CBK_AD[m] = 0
END FOR
T_CYL_BAL_LAM_AD = 0.00[s]
LV_STATE_CYL_BAL_LAM_CHG_AD = 0
STATE_CYL_BAL_LAM_AD_EOL = "Ready" (0[H])
```

Recurrence: 20 ms

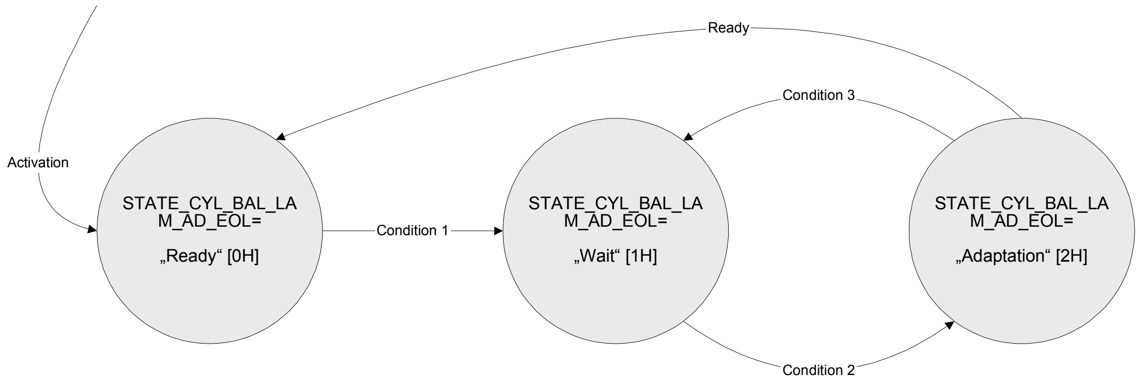
Activation:

```
LC_CYL_BAL_AD_EOL_RST = 0                  and
LV_CYL_BAL_AD_EOL_EXT_ADJ = 0              and
LC_CYL_BAL_CORD_EOL_STOP_MAN = 0           and
LV_CYL_BAL_LAM_AD_EOL = 0
```

Deactivation:

```
LC_CYL_BAL_AD_EOL_RST = 0                  and
LV_CYL_BAL_AD_EOL_EXT_ADJ = 0              and
[   LC_CYL_BAL_CORD_EOL_STOP_MAN = 1   or
    LV_CYL_BAL_LAM_AD_EOL = 1          ]
```

Signal flow diagram:



Formula section:

Note: At function activation always State "Ready" is entered

Cylinder balancing lambda adaptation at EOL: State 0[H] – “Ready”

Assignments at entry to state “Ready”:

STATE_CYL_BAL_LAM_AD_EOL = "Ready" (0[**H**])

Actions during state "Ready":

Transition to State 1[H] “Wait” (Condition 1)

// global lambda adaptation for at least one exhaust bank at EOL not yet finished

```
IF        LV_CYL_BAL_LAM_AD_EOL = 0
```

Assignments at exit from state “Ready”:

No assignments

Cylinder balancing lambda adaptation at EOL: State 1[H] – “Wait”

Assignments at entry to state “Wait”:

STATE_CYL_BAL_LAM_AD_EOL = "Wait" (1[**H**])

Actions during state "Wait":

// calculation of logical value to change state to adaptation

```
FOR m = 0 to NC_CBK_EX_NR – 1
    IF      ( LV_FAC_L_RNG_LAM_AD[m] = 1 and LV_CYL_BAL_LAM_AD_FAC[m] = 0 )
            OR
            ( LV_MFF_ADD_RNG_LAM_AD[m] =1 and LV_CYL_BAL_LAM_AD_ADD[m] = 0 )
    THEN    LV_STATE_CYL_BAL_LAM_CHG_AD = 1
    ENDIF
END FOR
```

Transition to State 2[H] “Adaptation” (Condition 2)

// conditions for global lambda adaptation achieved (multiple or additive)

```
IF        LV_STATE_CYL_BAL_LAM_CHG_AD = 1
```

Assignments at exit from state “Wait”:

```
T_CYL_BAL_LAM_AD = 0.00[s]
LV_STATE_CYL_BAL_LAM_CHG_AD = 0
FOR m = 0 to NC_CBK_EX_NR – 1
    CTR_CYL_BAL_LAM_CBK_AD[m] = 0
END FOR
```

Cylinder balancing lambda adaptation at EOL: State 2[H] – “Adaptation”

Assignments at entry to state “Adaptation”:

STATE_CYL_BAL_LAM_AD_EOL = "Adaptation" (2[**H**])

Actions during state "Adaptation":

Calculations:

```
// calculation of logical value to change state to wait
FOR m = 0 to NC_CBK_EX_NR – 1
    IF      ( LV_FAC_L_RNG_LAM_AD[m] = 0 and LV_MFF_ADD_RNG_LAM_AD[m] = 0 )
            OR
            ( LV_FAC_L_RNG_LAM_AD[m] = 1 and LV_CYL_BAL_LAM_AD_FAC[m] = 1 )
            OR
            ( LV_MFF_ADD_RNG_LAM_AD[m] = 1 and LV_CYL_BAL_LAM_AD_ADD[m] = 1 )
    THEN    LV_STATE_CYL_BAL_LAM_CHG_WAIT[m] = 1
            CTR_CYL_BAL_LAM_CBK_AD[m] = 0
    ELSE    LV_STATE_CYL_BAL_LAM_CHG_WAIT[m] = 0
    ENDIF
END FOR
```

```
// incrementation of lambda adaptation time with 0.02[s]
T_CYL_BAL_LAM_AD(n) = T_CYL_BAL_LAM_AD (n-1) + 0.02[s]
```

```
// determination of deviation between the actual and the last learned lambda adaptation values
FOR m = 0 to NC_CBK_EX_NR – 1
    IF (1)          Transition LV_LAM_AD_STOP[m] = "0" -> "1"
    THEN (1)
        IF (2a)         LV_FAC_L_RNG_LAM_AD[m] = 1
        THEN  (2a)
            IF (3a)         | FAC_LAM_ADJ_LAM_AD[m] | <
                                IP_FAC_L_LAM_ADJ_DIF_CYL_BAL
            THEN (3a)       CTR_CYL_BAL_LAM_CBK_AD[m](n) =
                                CTR_CYL_BAL_LAM_CBK_AD[m](n-1) + 1
                            // counter incremantation with 1
            ELSE (3a)       CTR_CYL_BAL_LAM_CBK_AD[m](n) =
```

```
                                        CTR_CYL_BAL_LAM_CBK_AD[m](n-1) –
                                        C_CTR_DEC_CYL_BAL_LAM_CBK_AD
                                   // counter value limited to zero, no counter underrun possible
                        ENDIF (3a)
                ELSEIF (2b)   LV_MFF_ADD_RNG_LAM_AD[m] = 1
                THEN    (2b)
                        IF (3b)         | FAC_LAM_ADJ_LAM_AD[m] | <
                                        IP_FAC_MFF_LAM_ADJ_DIF_CYL_BAL
                        THEN (3b)       CTR_CYL_BAL_LAM_CBK_AD[m](n) =
                                        CTR_CYL_BAL_LAM_CBK_AD[m](n-1) +1
                                   // counter incremantation with 1
                        ELSE (3b)       CTR_CYL_BAL_LAM_CBK_AD[m](n) =
                                        CTR_CYL_BAL_LAM_CBK_AD[m](n-1) –
                                        C_CTR_DEC_CYL_BAL_LAM_CBK_AD
                                   // counter value limited to zero, no counter underrun possible
                        ENDIF (3b)
                ENDIF (2)
        ENDIF (1)
END FOR
```

```
// calculation lambda adaptation readiness at end of line (exhaust bank specific)
FOR m = 0 to NC_CBK_EX_NR – 1
        IF (1a)         LV_FAC_L_RNG_LAM_AD[m] = 1                         and
                        LV_CYL_BAL_LAM_AD_FAC[m] = 0
        THEN (1a)
                IF      T_CYL_BAL_LAM_AD > C_T_MAX_CYL_BAL_LAM_AD                  or
                        CTR_CYL_BAL_LAM_CBK_AD[m] > C_CTR_MIN_CYL_BAL_LAM_AD_FAC
                THEN    LV_CYL_BAL_LAM_AD_FAC[m] = 1
                ENDIF
        ELSEIF (1b)     LV_MFF_ADD_RNG_LAM_AD[m] = 1                       and
                        LV_CYL_BAL_LAM_AD_ADD[m] = 0
        THEN (1b)
                IF      T_CYL_BAL_LAM_AD(n) > C_T_MAX_CYL_BAL_LAM_AD               or
                        CTR_CYL_BAL_LAM_CBK_AD[m] > C_CTR_MIN_CYL_BAL_LAM_AD_ADD
                THEN    LV_CYL_BAL_LAM_AD_ADD[m] = 1
                ENDIF
        ENDIF (1)
END FOR
```

// calculation of lambda adaptation readiness at end of line (global)

```
IF (1a)          C_CYL_BAL_LAM_AD_EOL_CONF = “ZERO” (0H)
THEN (1a)
    FOR m = 0 to NC_CBK_EX_NR – 1
        LV_CYL_BAL_LAM_AD_FAC[m] = 1
        LV_CYL_BAL_LAM_AD_ADD[m] = 1
    END FOR
    LV_CYL_BAL_LAM_AD_EOL = 1
    // adaptation of all exhaust banks finished at EOL
ELSEIF (1b)      C_CYL_BAL_LAM_AD_EOL_CONF = “ADD” (1H)
THEN (1b)
    FOR m = 0 to NC_CBK_EX_NR – 1
        LV_CYL_BAL_LAM_AD_FAC[m] = 1
    END FOR
    IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_AD\_ADD[m]} = \mathrm{NC\_CBK\_EX\_NR}$
    THEN    LV_CYL_BAL_LAM_AD_EOL = 1
            // adaptation of all exhaust banks (additive) are finished at EOL
    ENDIF
ELSEIF (1c)      C_CYL_BAL_LAM_AD_EOL_CONF = “FAC_L” (2H)
THEN (1c)
    FOR m = 0 to NC_CBK_EX_NR – 1
        LV_CYL_BAL_LAM_AD_ADD[m] = 1
    END FOR
    IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_AD\_FAC[m]} = \mathrm{NC\_CBK\_EX\_NR}$
    THEN    LV_CYL_BAL_LAM_AD_EOL = 1
            // adaptation of all exhaust banks (multiple) are finished at EOL or time out
    ENDIF
ELSEIF (1d)      C_CYL_BAL_LAM_AD_EOL_CONF = “ADD_FAC_L” (3H)
THEN (1d)
    IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_AD\_FAC[m]} = \mathrm{NC\_CBK\_EX\_NR}$      and
            $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_AD\_ADD[m]} = \mathrm{NC\_CBK\_EX\_NR}$
    THEN    LV_CYL_BAL_LAM_AD_EOL = 1
            // ad. of all exhaust banks (multiple and additive) are finished at EOL or time out
    ENDIF
ENDIF (1)
```

Transition to State 0[H] "Ready" (Ready)

// global lambda adaptation at EOL finished (multiple and additive)

IF LV_CYL_BAL_LAM_AD_EOL = 1

Transition to State 1[H] "Wait" (Condition 3)

// conditions for global lambda adaptation (multiple or additive) not achieved

IF

$$\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_STATE\_CYL\_BAL\_LAM\_CHG\_WAIT[m]} = \mathrm{NC\_CBK\_EX\_NR}$$

// adaptation condition for all exhaust banks are not fulfilled

Assignments at exit from state "Adaptation":

```
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_STATE_CYL_BAL_LAM_CHG_WAIT[m] = 0
END FOR
```

6.11 Cylinder balancing coordination at DC

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CYL_BAL_LAM_AD_DC	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation at active driving cycle					
LV_CYL_BAL_LAM_SEL_AD_HOT_DC	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (hot) at active driving cycle					
LV_CYL_BAL_LAM_SEL_AD_COLD_DC	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (cold) at active driving cycle					
LV_CYL_BAL_AD_HOM_REQ_DC	V/O	0...1H	0...1	1	-
Flag for homogenous combustion mode for cylinder balancing adaptation requested at active driving cycle					
LV_CYL_BAL_HOM_REQ_INT_DC	V	0...1H	0...1	1	-
Flag for homogenous combustion mode requested internal at active driving cycle					
LV_CYL_BAL_AD_WG_OPEN_REQ[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	-
Flag for waste gate open requested for cylinder balancing adaptation					
LV_CYL_BAL_AD_WG_OPEN_REQ_DC[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Flag for waste gate open requested for cylinder balancing adaptation at active driving cycle					
STATE_CYL_BAL_LAM_SEL_AD_DC	V	0H 1H 2H	Wait Request Adaptation	-	-
State of cylinder balancing cylinder selective lambda adaptation at active driving cycle					
STATE_CYL_BAL_LAM_AD_DC	V	0H 1H 2H	Wait Request Adaptation	-	-
State of cylinder balancing lambda adaptation at active driving cycle					
LV_STATE_LAM_SEL_DC_CHG_REQ	V	0...1H	0...1	1	-
Flag for stage change to "request" for cylinder balancing selective lambda adaptation at active driving cycle					
LV_STATE_LAM_SEL_DC_CHG_WAIT	V	0...1H	0...1	1	-
Flag for stage change to "wait" for cylinder balancing selective lambda adaptation at active driving cycle					
LV_STATE_LAM_AD_DC_CHG_REQ	V/O	0...1H	0...1	1	-
Flag for stage change to "request" for cylinder balancing lambda adaptation at active driving cycle					
LV_STATE_LAM_AD_DC_CHG_WAIT	V	0...1H	0...1	1	-
Flag for stage change to "wait" for cylinder balancing lambda adaptation at active driving cycle					
T_CYL_BAL_LAM_SEL_AD_REQ_DC[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing selective lambda adaptation request at active driving cycle					
T_CYL_BAL_LAM_AD_REQ_DC[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing lambda adaptation request at active driving cycle					
T_CYL_BAL_LAM_SEL_AD_WAIT_DC	V	0…FFFFH	0…6553.5	0.1	s
Waiting time for cylinder balancing selective lambda adaptation activation at active driving cycle					
T_CYL_BAL_LAM_AD_WAIT_DC	V	0…FFFFH	0…6553.5	0.1	s
Waiting time for cylinder balancing lambda adaptation activation at active driving cycle					
LV_CYL_BAL_LAM_SEL_AD_REQ_DC[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
Flag for cylinder balancing selective lambda adaptation request at active driving cycle					
LV_CYL_BAL_LAM_AD_REQ_DC[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation request at active driving cycle					
LV_CYL_BAL_LAM_SEL_HOM_REQ_DC	V	0...1H	0...1	1	-
Flag for HOM request for cylinder balancing selective lambda adaptation at active driving cycle					
LV_CYL_BAL_LAM_AD_HOM_REQ_DC	V	0...1H	0...1	1	-
Flag for HOM request for cylinder balancing lambda adaptation at active driving cycle					
LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[NC_CBK_EX_NR]	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (low range) bank specific at active driving cycle					

LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[NC_CBK_EX_NR]	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (high range) bank specific at active driving cycle					
T_CYL_BAL_LAM_SEL_AD_RNG_L_DC[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing cylinder selective lambda adaptation (low range) bank specific at active driving cycle					
T_CYL_BAL_LAM_SEL_AD_RNG_H_DC[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing cylinder selective lambda adaptation (high range) bank specific at active driving cycle					
LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[NC_CBK_EX_NR]	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation (cold range) bank specific at active driving cycle					
T_CYL_BAL_LAM_SEL_RNG_COLD_DC[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing cylinder selective lambda adaptation (cold range) bank specific at active driving cycle					
LV_CYL_BAL_LAM_AD_ADD_DC[NC_CBK_EX_NR]	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation (additive) exhaust bank specific at active driving cycle					
LV_CYL_BAL_LAM_AD_FAC_L_DC[NC_CBK_EX_NR]	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation (multiple low) exhaust bank specific at active driving cycle					
LV_CYL_BAL_LAM_AD_FAC_H_DC[NC_CBK_EX_NR]	V/O/S	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation (multiple high) exhaust bank specific at active driving cycle					
T_CYL_BAL_LAM_AD_ADD_DC[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing lambda adaptation (additive) exhaust bank specific at active driving cycle					
T_CYL_BAL_LAM_AD_FAC_L_DC[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing lambda adaptation (multiple low) exhaust bank specific at active driving cycle					
T_CYL_BAL_LAM_AD_FAC_H_DC[NC_CBK_EX_NR]	V	0…FFFFH	0…6553.5	0.1	s
Time for cylinder balancing lambda adaptation (multiple high) exhaust bank specific at active driving cycle					
LV_CYL_BAL_LAM_AD_END[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	-
Flag to indicate end of lambda adaptation cycle for cylinder balancing at active driving cycle					
T_CYL_BAL_LAM_SEL_DC	V/O/S	0…FFFFFFFFH	0…429496729.5	0.1	s
Time gap between cylinder balancing cylinder selective lambda adaptation parts at active driving cycle					
T_CYL_BAL_LAM_AD_DC	V/O/S	0…FFFFFFFFH	0…429496729.5	0.1	s
Time gap between cylinder balancing lambda adaptation parts at active driving cycle					
LV_CYL_BAL_LAM_AD_ENA_DC	V/O	0...1H	0...1	1	-
Flag for EVAC time controller to enable lambda adaptation area for active driving cycle					
LV_T_STEP_CYL_BAL_LAM_SEL_DC	V	0...1H	0...1	1	-
Flag for cylinder balancing cylinder selective lambda adaptation fixed time steps enabled at active driving cycle					
LV_T_STEP_CYL_BAL_LAM_AD_DC	V/O	0...1H	0...1	1	-
Flag for cylinder balancing lambda adaptation fixed time steps enabled at active driving cycle					
STATE_CYL_BAL_LAM_AD[NC_CBK_EX_NR]	V	0H 1H 2H 3H 4H 5H 6H 7H	INIT WAIT CDN_FAC_L CDN_FAC_H CDN_ADD ADAPT_FAC_L ADAPT_FAC_H ADAPT_ADD	1	-
Cylinder balancing lambda adaptation state					

Input data:

T_AST_BAL	LV_ST_END	LV_LAM_AD_ENA	STATE_LAM_AD[NC_CBK_EX_NR]

STATE_CDN_LAM_CYL_SEL_ADJ_RNG[NC_CBK_EX_NR]	STATE_LAM_CYL_SEL_ADJ[NC_CBK_EX_NR]	FAC_CYL_LAM_ABSV_SUM[NC_CBK_EX_NR]	LV_LAM_AD_END_CBK[NC_CBK_EX_NR]
LV_MFF_ADD_RNG_LAM_AD[NC_CBK_EX_NR]	LV_FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	LV_FAC_H_RNG_LAM_AD[NC_CBK_EX_NR]	TCO_ST_BAL
LC_CYL_BAL_CORD_EOL_STOP_MAN	LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[NC_CBK_EX_NR]	LV_LAM_CYL_SEL_ADJ_VLD[NC_CBK_EX_NR]	LV_NT_RGN_REQ
LC_TCHA_CONF	NC_CBK_EX_NR	TCO	LV_LAM_AD_STOP[NC_CBK_EX_NR]
LV_CYL_BAL_HOM_REQ_EXT	NT_AGI		

Calibration data:

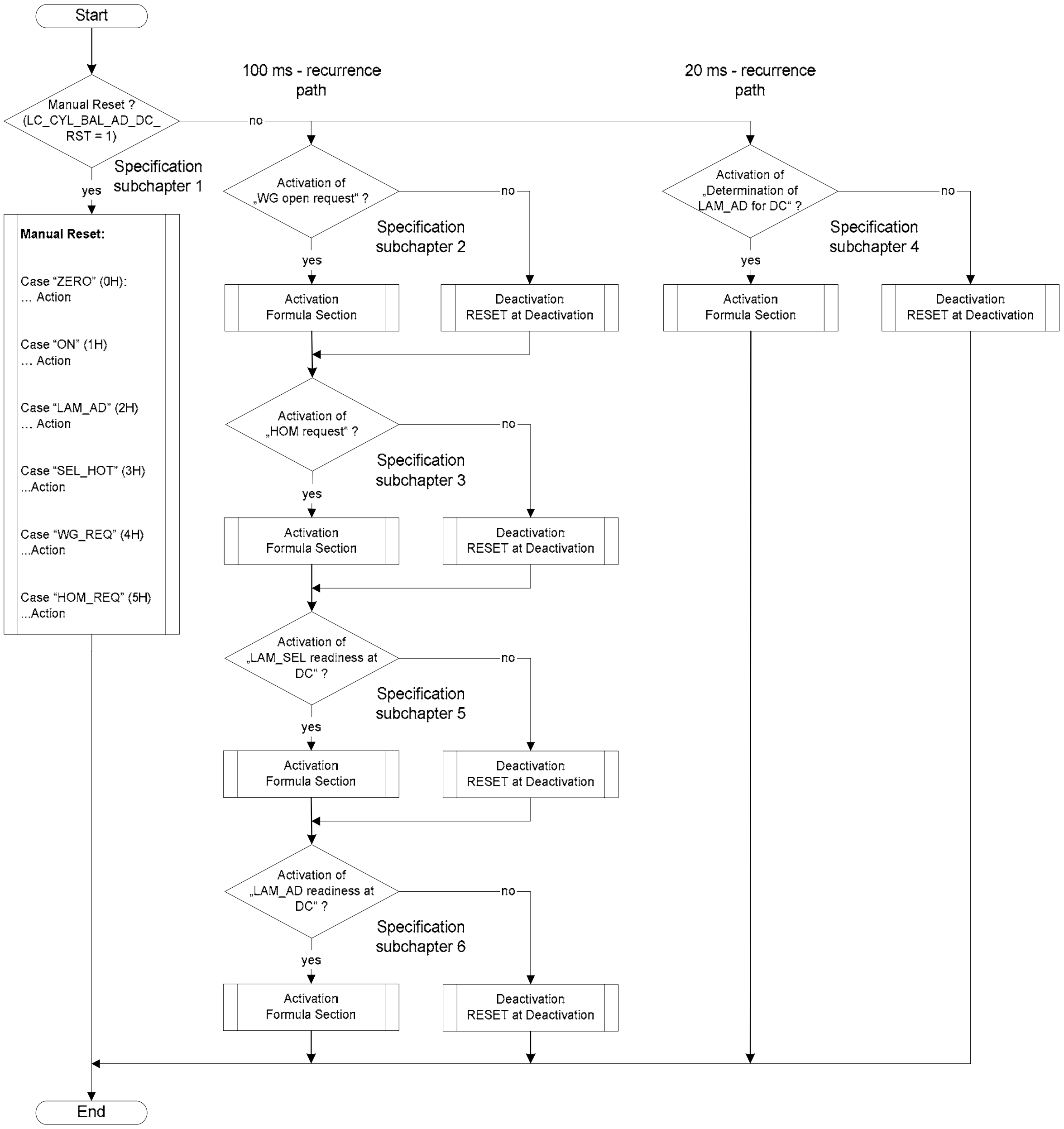
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_T_CYL_BAL_LAM_SEL_AD_DC	8	0…FFFFH	0…6553.5	0.1	s
LDP_T_AST_BAL_ID_T_LAM_SEL_DC	8	0…FFFFH	0…6553.5	0.1	s
Time intervals for control of cylinder balancing selective lambda adaptation at active driving cycle					
ID_T_CYL_BAL_LAM_AD_DC	8	0…FFFFH	0…6553.5	0.1	s
LDP_T_AST_BAL_ID_T_LAM_AD_DC	8	0…FFFFH	0…6553.5	0.1	s
Time intervals for control of cylinder balancing lambda adaptation at active driving cycle					
ID_IDX_CYL_BAL_HOM_REQ_INT_DC	6	0…4H	0…4	1	-
LDP_NT_AGI_ID_IDX_REQ_INT_DC	6	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging index value for internal homogenous combustion mode request at active driving cycle					
ID_T_CYL_BAL_HOM_REQ_INT_DC[5]	6	0…FFFFH	0…6553.5	0.1	s
LDP_T_AST_BAL_ID_T_REQ_INT_DC[5]	6	0…FFFFH	0…6553.5	0.1	s
Time intervals for internal homogenous combustion mode request at active driving cycle					
C_T_AST_MAX_CYL_BAL_AD_DC	1	0…FFFFH	0…6553.5	0.1	s
Maximum time after start for control of cylinder balancing selective lambda adaptation at active driving cycle					
C_TCO_ST_MAX_CYL_BAL_AD_DC	1	0...FEH	-48...142.5	0.75	°C
Maximum TCO_ST start for control of cylinder balancing selective lambda adaptation at active driving cycle					
C_TCO_ST_MIN_CYL_BAL_AD_DC	1	0...FEH	-48...142.5	0.75	°C
Minimum TCO_ST start for control of cylinder balancing selective lambda adaptation at active driving cycle					
C_TCO_THD_CYL_BAL_LAM_SEL_DC	1	0...FEH	-48...142.5	0.75	°C
TCO threshold for control of cylinder balancing selective lambda adaptation at active driving cycle					
C_T_MIN_CYL_BAL_LAM_SEL_DC	1	0…FFFFH	0…6553.5	0.1	s
Minimum time to enable cylinder balancing cylinder selective lambda adaptation part at active driving cycle					
C_T_MAX_CYL_BAL_LAM_SEL_DC	1	0…FFFFFFFFH	0…429496729.5	0.1	s
Maximum time to enable cylinder balancing cylinder selective lambda adaptation part at active driving cycle					
C_T_MIN_CYL_BAL_LAM_AD_DC	1	0…FFFFH	0…6553.5	0.1	s
Minimum time to enable cylinder balancing lambda adaptation part at active driving cycle					
C_T_MAX_CYL_BAL_LAM_AD_DC	1	0…FFFFFFFFH	0…429496729.5	0.1	s
Maximum time to enable cylinder balancing lambda adaptation part at active driving cycle					
C_T_MIN_CYL_BAL_AD_REQ_DC	1	0…FFFFH	0…6553.5	0.1	s
Minimum time for cylinder balancing adaptation conditions within limits at active driving cycle					
C_T_MIN_CYL_BAL_AD_WAIT_DC	1	0…FFFFH	0…6553.5	0.1	s
Minimum waiting time at entry to state adaptation for cylinder balancing adaptation at active driving cycle					
C_FAC_CYL_LAM_ABSV_SUM_BAL_DC	1	0...7FFFH	0...49.998474	1.5259E-3	%
Threshold for cylinder balancing cylinder selective lambda adaptation within limits at active driving cycle					
C_CYL_BAL_LAM_AD_DC_CONF	1	0H 1H 2H 3H	ADD FAC_L ADD_FAC_L ADD_FAC_ALL	-	-
Configuration of lambda adaptation correction values for cylinder balancing at active driving cycle					
C_CYL_BAL_LAM_SEL_HOT_DC_CONF	1	0H 1H 2H	LOW HIGH LOW_HIGH	-	-
Configuration of cylinder selective lambda adaptation correction values for cylinder balancing at driving cycle					
LC_CYL_BAL_AD_DC_RST	1	0...1H	0...1	1	-

Manual switch for cylinder balancing adaptation reset at active driving cycle					
C_CYL_BAL_AD_DC_RST	1	0H 1H 2H 3H 4H 5H 6H	ZERO ON LAM_AD SEL_HOT WG_REQ HOM_REQ SEL_COLD	-	-
Reset of cylinder balancing adaptation readiness flags at active driving cycle					
LC_CYL_BAL_CORD_DC_STOP_MAN	1	0...1H	0...1	1	-
Logical constant for manual stop (=1) of cylinder balancing coordination at driving cycle					

FUNCTION DESCRIPTION:

Task of the cylinder balancing coordination at DC is to control the adaptation functions during the current driving cycle.

Signal flow diagram:



Application conditions:

Initialisation: See separate chapters

Recurrence: See separate chapters

Activation: See separate chapters

Deactivation: See separate chapters

6.11.1 Manual reset of cylinder balancing readiness at active drving cycle (DC):

FUNCTION DESCRIPTION:

General information:

In case of a manual reset it is possible to adjust the exported output data to a wanted value to make the function test and handling more easier. While the manual function reset is activated, all following subchapters within the module "cylinder balancing coordination at DC" are deactivated and the corresponding "Initialization at Deactivation" process for this chapters is blocked during this period.

Application conditions:

Recurrence: 100 ms

Activation: LC_CYL_BAL_AD_DC_RST = 1

Deactivation: LC_CYL_BAL_AD_DC_RST = 0

Formula section:

```
IF        C_CYL_BAL_AD_DC_RST = “ZERO” (0H)
THEN      LV_CYL_BAL_LAM_AD_DC = 0
          LV_CYL_BAL_LAM_SEL_AD_HOT_DC = 0
          LV_CYL_BAL_LAM_SEL_AD_COLD_DC = 0
          LV_CYL_BAL_AD_HOM_REQ_DC = 0
          FOR m = 0 to NC_CBK_EX_NR – 1
                LV_CYL_BAL_AD_WG_OPEN_REQ[m] = 0
          END FOR

ELSEIF    C_CYL_BAL_AD_DC_RST = “ON” (1H)
THEN      LV_CYL_BAL_LAM_AD_DC = 1
          LV_CYL_BAL_LAM_SEL_AD_HOT_DC = 1
          LV_CYL_BAL_LAM_SEL_AD_COLD_DC = 1
          LV_CYL_BAL_AD_HOM_REQ_DC = 1
          FOR m = 0 to NC_CBK_EX_NR – 1
                LV_CYL_BAL_AD_WG_OPEN_REQ[m] = 1
          END FOR
```

```
ELSEIF    C_CYL_BAL_AD_DC_RST = “LAM_AD” (2H)
THEN      LV_CYL_BAL_LAM_AD_DC = 1

ELSEIF    C_CYL_BAL_AD_DC_RST = “SEL_HOT” (3H)
THEN      LV_CYL_BAL_LAM_SEL_AD_HOT_DC = 1

ELSEIF    C_CYL_BAL_AD_DC_RST = “WG_REQ” (4H)
THEN      FOR m = 0 to NC_CBK_EX_NR – 1
                LV_CYL_BAL_AD_WG_OPEN_REQ[m] = 1
          END FOR

ELSEIF    C_CYL_BAL_AD_DC_RST = “HOM_REQ” (5H)
THEN      LV_CYL_BAL_AD_HOM_REQ_DC = 1

ELSEIF    C_CYL_BAL_AD_DC_RST = “SEL_COLD” (6H)
THEN      LV_CYL_BAL_LAM_SEL_AD_COLD_DC = 1
ENDIF
```

6.11.2 Waste gate open request for cylinder balancing adaptation:

FUNCTION DESCRIPTION:

General information:

For a turbo engine, the cylinder individual lambda control is only possible while the waste gate is sufficient opened to identify the exhaust package for each cylinder. In case of a planned adaptation, a request to open the waste gate is carried out for each turbine before.

Application conditions:

*Initialization at Reset **or** Deactivation:*

```
FOR m = 0 to NC_CBK_EX_NR – 1
      LV_CYL_BAL_AD_WG_OPEN_REQ[m] = 0
END FOR
```

Recurrence: 100 ms

Activation:

```
LC_CYL_BAL_AD_DC_RST = 0                      and
LC_TCHA_CONF = 1                              and
( LC_CYL_BAL_CORD_DC_STOP_MAN = 0          or
  LC_CYL_BAL_CORD_EOL_STOP_MAN = 0 )
```

```
Deactivation:   LC_CYL_BAL_AD_DC_RST = 0                                  and
                [     LC_TCHA_CONF = 0                              or
                      ( LC_CYL_BAL_CORD_DC_STOP_MAN = 1 and
                        LC_CYL_BAL_CORD_EOL_STOP_MAN = 1 )          ]
```

Formula section:

```
FOR m = 0 to NC_CBK_EX_NR – 1
    IF       LV_CYL_BAL_AD_WG_OPEN_REQ_EOL[m] = 1     or
             LV_CYL_BAL_AD_WG_OPEN_REQ_DC[m] = 1
    THEN     LV_CYL_BAL_AD_WG_OPEN_REQ[m] = 1
    ELSE     LV_CYL_BAL_AD_WG_OPEN_REQ[m] = 0
    ENDIF
END FOR
```

6.11.3 Homogenous combustion mode request for cylinder balancing adaptation:

FUNCTION DESCRIPTION:

General information:

For a stratified combustion engine, the cylinder individual lambda control as well as the global lambda adaptation are only possible while the engine is running in homogenous combustion mode with activated lambda control. In case of a planned adaptation, a request to switch to homogenous combustion is carried out before.

Application conditions:

Initialization at Reset ***or*** *Deactivation:*

```
    LV_CYL_BAL_AD_HOM_REQ_DC = 0

Recurrence:     100 ms

Activation:     LC_CYL_BAL_AD_DC_RST = 0                           and
                LC_TCHA_CONF = 0                                   and
                LC_CYL_BAL_CORD_DC_STOP_MAN = 0

Deactivation:   LC_CYL_BAL_AD_DC_RST = 0                           and
                [     LC_TCHA_CONF = 1                       or
                      LC_CYL_BAL_CORD_DC_STOP_MAN = 1        ]
```

Formula section:

// determination of NOx aging for homogenous combustion mode request at active driving cycle

```
IF          LV_ST_END = 0
```

Transmittal, reproduction, dissemination and/or editing of this document as well as utilization of its contents and communication there of to others without express authorization are prohibited. Offenders will be held liable for payment of damages. All rights created by patent grant or registration of a utility model or design patent are reserved.

```
THEN    IDX = ID_IDX_CYL_BAL_HOM_REQ_INT_DC
        // NOx aging for homogenous combustion mode request at active driving cycle
ENDIF

// calculation of internal homogenous combustion mode request at active driving cycle
IF      T_AST_BAL < C_T_AST_MAX_CYL_BAL_AD_DC               and
        TCO_ST_BAL < C_TCO_ST_MAX_CYL_BAL_AD_DC             and
        TCO_ST_BAL > C_TCO_ST_MIN_CYL_BAL_AD_DC             and
        T_AST_BAL < ID_T_CYL_BAL_HOM_REQ_INT_DC[IDX]
THEN    LV_CYL_BAL_HOM_REQ_INT_DC = 1
        // internal homogen combustion mode request at active driving cycle
ELSE    LV_CYL_BAL_HOM_REQ_INT_DC = 0
ENDIF

// calculation of overall homogen combustion mode request at active driving cycle
IF      LV_CYL_BAL_LAM_SEL_HOM_REQ_DC = 1        or
        LV_CYL_BAL_LAM_AD_HOM_REQ_DC = 1         or
        LV_CYL_BAL_HOM_REQ_INT_DC = 1            or
        LV_CYL_BAL_HOM_REQ_EXT = 1
THEN    LV_CYL_BAL_AD_HOM_REQ_DC = 1
ELSE    LV_CYL_BAL_AD_HOM_REQ_DC = 0
ENDIF
```

6.11.4 Determination of lambda adaptation cycle for cylinder balancing at DC

FUNCTION DESCRIPTION:

General information:

Because the lambda adaptation function is running with a higher recurrence in comparison to cylinder balancing coordination at DC, the determination of a finished lambda adaptation cycle has to be carried out faster to avoid signal under-sampling.

Application conditions:

*Initialization at Reset **or** Deactivation:*

```
    FOR m = 0 to NC_CBK_EX_NR – 1
        LV_CYL_BAL_LAM_AD_END[m] = 0
        STATE_CYL_BAL_LAM_AD[m] = "INIT" (0[H])
    END FOR
```

Recurrence: 20 ms

```
Activation:     LC_CYL_BAL_AD_DC_RST = 0                        and
                LC_TCHA_CONF = 0                                and
                LC_CYL_BAL_CORD_DC_STOP_MAN = 0

Deactivation:   LC_CYL_BAL_AD_DC_RST = 0                        and
                [     LC_TCHA_CONF = 1                       or
                      LC_CYL_BAL_CORD_DC_STOP_MAN = 1        ]
```

Formula section:

```
FOR m = 0 to NC_CBK_EX_NR – 1
    IF          LV_LAM_AD_STOP[m] = 0 --> 1
    THEN        LV_CYL_BAL_LAM_AD_END[m] = LV_LAM_AD_END_CBK[m]
                STATE_CYL_BAL_LAM_AD[m] = STATE_LAM_AD[m]
    ENDIF
END FOR
```

6.11.5 Calculation of cylinder balancing selective lambda adaptation readiness (DC):

FUNCTION DESCRIPTION:

General information:

A periodic request for homogenous combustion mode is carried out in defined time intervals to allow the start of the cylinder selective adaptation functions, which are only activated in homogenous combustion mode. In case of a turbo engine, a periodic request to open the waste gate is carried out.

Application conditions:

Initialization at first Engine run or NVMY-error:

```
T_CYL_BAL_LAM_SEL_DC = 0.0[s]                         (first initialization)
LV_CYL_BAL_LAM_SEL_AD_HOT_DC = 0                      (first initialization)
LV_CYL_BAL_LAM_SEL_AD_COLD_DC = 0                     (first initialization)
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 0             (first initialization)
    LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 0             (first initialization)
    LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] = 0             (first initialization)
END FOR
```

Initialization at Reset:

T_CYL_BAL_LAM_SEL_DC = T_CYL_BAL_LAM_SEL_DC *(NVMY)*

LV_CYL_BAL_LAM_SEL_AD_HOT_DC = LV_CYL_BAL_LAM_SEL_AD_HOT_DC *(NVMY)*

```
LV_CYL_BAL_LAM_SEL_AD_COLD_DC = LV_CYL_BAL_LAM_SEL_AD_COLD_DC (NVMY)
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] =
                    LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] (NVMY)
    LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] =
                    LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] (NVMY)
    LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] =
                    LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] (NVMY)
END FOR
```

(NVMY: values stored in the non-volatile memory)

```
FOR m = 0 to NC_CBK_EX_NR – 1
    IF          C_CYL_BAL_LAM_SEL_HOT_DC_CONF = “LOW” (0H)
    THEN        LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 1
    ELSEIF      C_CYL_BAL_LAM_SEL_HOT_DC_CONF = “HIGH” (1H)
    THEN        LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 1
    ENDIF
END FOR
```

Initialization at Reset ***or*** *Deactivation:*

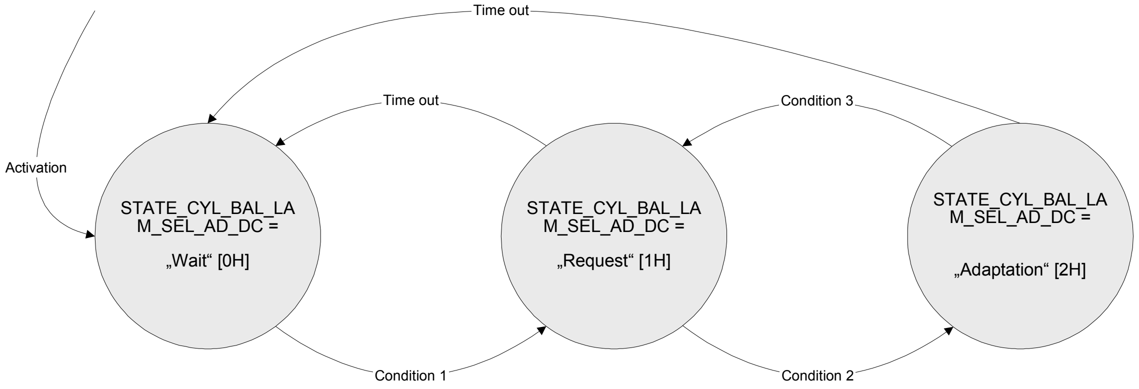
```
LV_CYL_BAL_LAM_SEL_HOM_REQ_DC = 0
STATE_CYL_BAL_LAM_SEL_AD_DC = "Wait" (0[H])
LV_STATE_LAM_SEL_DC_CHG_REQ = 0
LV_STATE_LAM_SEL_DC_CHG_WAIT = 0
T_CYL_BAL_LAM_SEL_AD_WAIT_DC = 0.0[s]
LV_T_STEP_CYL_BAL_LAM_SEL_DC = 0
FOR m = 0 to NC_CBK_EX_NR – 1
    T_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 0.0[s]
    T_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 0.0[s]
    T_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] = 0.0[s]
    LV_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0
    T_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0.0[s]
    LV_CYL_BAL_AD_WG_OPEN_REQ_DC[m] = 0
END FOR
```

Recurrence: 100 ms

Activation:
LC_CYL_BAL_AD_DC_RST = 0 **and**
LV_ST_END = 1 **and**
LC_CYL_BAL_CORD_DC_STOP_MAN = 0

```
Deactivation:   LC_CYL_BAL_AD_DC_RST = 0                        and
                [   LV_ST_END = 0                           or
                    LC_CYL_BAL_CORD_DC_STOP_MAN = 1         ]
```

Signal flow diagram:



Formula section:

```
IF (1)      T_AST_BAL < C_T_AST_MAX_CYL_BAL_AD_DC              and
            TCO_ST_BAL < C_TCO_ST_MAX_CYL_BAL_AD_DC            and
            TCO_ST_BAL > C_TCO_ST_MIN_CYL_BAL_AD_DC
THEN (1)    // DC adaptation requests controlled by fix time steps
    IF (2)          T_AST_BAL < ID_T_CYL_BAL_LAM_SEL_AD_DC
    THEN (2)        LV_STATE_LAM_SEL_DC_CHG_REQ = 1
                    LV_STATE_LAM_SEL_DC_CHG_WAIT = 0
                    LV_T_STEP_CYL_BAL_LAM_SEL_DC = 1
    ELSE (2)        LV_STATE_LAM_SEL_DC_CHG_REQ = 0
                    LV_STATE_LAM_SEL_DC_CHG_WAIT = 1
                    LV_T_STEP_CYL_BAL_LAM_SEL_DC = 0
    ENDIF (2)
ELSE (1)    // DC adaptation requests for controlled by flexible time steps
    LV_T_STEP_CYL_BAL_LAM_SEL_DC = 0
    IF (2)          T_CYL_BAL_LAM_SEL_DC < C_T_MAX_CYL_BAL_LAM_SEL_DC
    THEN (2)        T_CYL_BAL_LAM_SEL_DC(n) = T_CYL_BAL_LAM_SEL_DC(n-1) + 0.1[s]
                    // waiting time incrementation, flexible time steps running
        IF (3)          ( LV_CYL_BAL_LAM_SEL_AD_COLD_DC = 0            and
                          TCO < C_TCO_THD_CYL_BAL_LAM_SEL_DC           )
                        OR
                        ( LV_CYL_BAL_LAM_SEL_AD_HOT_DC = 0             and
                          TCO ≥ C_TCO_THD_CYL_BAL_LAM_SEL_DC           )
```

```
            THEN (3)      // enable conditions to change adaptation state from "Wait" to "Request"
                          LV_STATE_LAM_SEL_DC_CHG_REQ = 1
                          LV_STATE_LAM_SEL_DC_CHG_WAIT = 0
            ELSE (3)      // enable conditions to change adaptation state from "Request" to "Wait"
                          LV_STATE_LAM_SEL_DC_CHG_REQ = 0
                          LV_STATE_LAM_SEL_DC_CHG_WAIT = 1
            ENDIF (3)
        ELSE (2)      // reset of waiting time, new beginning of flexible time steps
            T_CYL_BAL_LAM_SEL_DC = 0.0[s]
            LV_CYL_BAL_LAM_SEL_AD_HOT_DC = 0
            LV_CYL_BAL_LAM_SEL_AD_COLD_DC = 0
            FOR m = 0 to NC_CBK_EX_NR – 1
                LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] = 0
                T_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] = 0.0[s]
                IF            C_CYL_BAL_LAM_SEL_HOT_DC_CONF = “LOW” (0H)
                THEN          LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 0
                              T_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 0.0[s]
                ELSEIF        C_CYL_BAL_LAM_SEL_HOT_DC_CONF = “HIGH” (1H)
                THEN          LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 0
                              T_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 0.0[s]
                ELSEIF        C_CYL_BAL_LAM_SEL_HOT_DC_CONF = “LOW_HIGH” (2H)
                THEN          LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 0
                              LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 0
                              T_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 0.0[s]
                              T_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 0.0[s]
                ENDIF
            END FOR
        ENDIF (2)
    ENDIF (1)
```

Note: At function activation always State "Wait" is entered

Cylinder balancing adaptation at DC: State 0[H] – “Wait”

Assignments at entry to state “Wait”:

```
STATE_CYL_BAL_LAM_SEL_AD_DC = "Wait" (0[H])
LV_CYL_BAL_LAM_SEL_HOM_REQ_DC = 0
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_AD_WG_OPEN_REQ_DC[m] = 0
END FOR
```

Action during state "Wait":

Transition to State 1[H] “Request” (Condition 1)

```
// Lambda Adaptation manager request for change of state
IF          LV_STATE_LAM_SEL_DC_CHG_REQ = 1
```

Assignments at exit from state “Wait”:

No assignments

Cylinder balancing adaptation at DC: State 1[H] – “Request”

Assignments at entry to state “Request”:

STATE_CYL_BAL_LAM_SEL_AD_DC = "Request" (1[**H**])

Action during state "Request":

Transition to State 0[H] “Wait” (Time out)

```
// Lambda Adaptation manager request for change of state
IF          LV_STATE_LAM_SEL_DC_CHG_WAIT = 1
```

Calculations:

```
// conditions for cylinder balancing selective lambda adaptation achieved
FOR m = 0 to NC_CBK_EX_NR – 1
    IF      ( STATE_CDN_LAM_CYL_SEL_ADJ_RNG[m] = “LOW_RNG” (1[H])       and
              [ LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 0      or
                LV_T_STEP_CYL_BAL_LAM_SEL_DC = 1            ]          )
            OR
            ( STATE_CDN_LAM_CYL_SEL_ADJ_RNG[m] = “HIGH_RNG” (2[H])      and
              [ LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 0      or
                LV_T_STEP_CYL_BAL_LAM_SEL_DC = 1            ]          )
            OR
            ( STATE_CDN_LAM_CYL_SEL_ADJ_RNG[m] = “COLD_RNG” (3[H])      and
              [ LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] = 0      or
                LV_T_STEP_CYL_BAL_LAM_SEL_DC = 1            ]          )
    THEN    T_CYL_BAL_LAM_SEL_AD_REQ_DC[m](n) =
                    T_CYL_BAL_LAM_SEL_AD_REQ_DC[m](n-1) + 0.1[s]
    ELSE    T_CYL_BAL_LAM_SEL_AD_REQ_DC[m](n) = 0.0[s]
    ENDIF
END FOR
```

```
// calculation of request for bank specific homogenuous cobustion mode switch
FOR m = 0 to NC_CBK_EX_NR – 1
    IF      T_CYL_BAL_LAM_SEL_AD_REQ_DC[m] ≥ C_T_MIN_CYL_BAL_AD_REQ_DC
```

```
        THEN    LV_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 1
        ELSE    LV_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0
        ENDIF
    END FOR
```

```
// calculation of request for homogenous combustion mode switch for non turbo engine
IF (1)      LC_TCHA_CONF = 0
THEN (1)
    IF (2)      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_REQ\_DC[m]} > 0$
                // homogenous switch after waiting time
                OR
                ( [ LV_NT_RGN_REQ = 1 or LV_T_STEP_CYL_BAL_LAM_SEL_DC = 1 ]        and
                  $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{T\_CYL\_BAL\_LAM\_SEL\_AD\_REQ\_DC[m]} > 0.0[\mathrm{s}]$        )
                // immediate homogenous switch in case of NT request or fixed time steps
    THEN (2)        LV_CYL_BAL_LAM_SEL_HOM_REQ_DC = 1
    ENDIF (2)
ELSE (1)
// calculation of request for waste gate open for turbo engine (LC_TCHA_CONF=1)
    FOR (2) m = 0 to NC_CBK_EX_NR – 1
        IF (3)      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_REQ\_DC[m]} > 0$
                    // waste gate open after waiting time
                    OR
                    ( LV_T_STEP_CYL_BAL_LAM_SEL_DC = 1                and
                      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{T\_CYL\_BAL\_LAM\_SEL\_AD\_REQ\_DC[m]} > 0.0[\mathrm{s}]$     )
                    // immediate waste gate open in case of fixed time steps
        THEN (3)        LV_CYL_BAL_AD_WG_OPEN_REQ_DC[m] = 1
        ENDIF (3)
    END FOR (2)
ENDIF (1)
```

Transition to State 2[H] “Adaptation” (Condition 2)

```
// conditions for combustion mode switch to homogenuous or waste gate open achieved
IF          LV_CYL_BAL_LAM_SEL_HOM_REQ_DC = 1                                  or
            $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_AD\_WG\_OPEN\_REQ\_DC[m]} > 0$
```

Assignments at exit from state “Request”:

```
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0
    T_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0.0[s]
END FOR
```

Cylinder balancing adaptation at DC: State 2[H] – “Adaptation”

Assignments at entry to state “Adaptation”:

```
STATE_CYL_BAL_LAM_SEL_AD_DC = "Adaptation" (2[H])
T_CYL_BAL_LAM_SEL_AD_WAIT_DC = C_T_MIN_CYL_BAL_AD_WAIT_DC
FOR m = 0 to NC_CBK_EX_NR – 1
    T_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = C_T_MIN_CYL_BAL_AD_WAIT_DC
END FOR
```

Actions during state "Adaptation":

Transition to State 0[H] “Wait” (Time out)

```
// Lambda Adaptation manager request for change of state
IF          LV_STATE_LAM_SEL_DC_CHG_WAIT = 1
```

Transition to State 1[H] “Request” (Condition 3)

```
// conditions for combustion mode switch to homogenuous or waste gate open not achieved
IF   ( LC_TCHA_CONF = 0  and  LV_CYL_BAL_LAM_SEL_HOM_REQ_DC = 0 )          or
     ( LC_TCHA_CONF = 1  and  $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_AD\_WG\_OPEN\_REQ\_DC}[m] = 0$ )
```

Calculations:

```
// calculation of waiting time after entry in homogenuous combustion mode or waste gate open
IF (1)          T_CYL_BAL_LAM_SEL_AD_WAIT_DC = 0.0 [s]
                // waiting time after switch to homogenous combustion mode elapsed
THEN (1)
    FOR (2) m = 0 to NC_CBK_EX_NR – 1
        IF (3)          ( STATE_LAM_CYL_SEL_ADJ[m] <> “ADJ_NOM_L_RNG” (4[H])      and
                          STATE_LAM_CYL_SEL_ADJ[m] <> “ADJ_NOM_H_RNG” (5[H])      and
                          STATE_LAM_CYL_SEL_ADJ[m] <> "ADJ_ENG_COLD” (2[H])       )
                        // low range, high range, cold range CILC adaptation area not active
        THEN (3)        T_CYL_BAL_LAM_SEL_AD_REQ_DC[m](n) =
                                T_CYL_BAL_LAM_SEL_AD_REQ_DC[m](n-1) - 0.1[s]
        ELSE (3)        T_CYL_BAL_LAM_SEL_AD_REQ_DC[m](n) =
                                C_T_MIN_CYL_BAL_AD_WAIT_DC
```

```
            ENDIF (3)
            IF (3)        T_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0.0 [s]
                          // time with no CILC adaptation area active elapsed
                          OR
                          ( STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_NOM_L_RNG” (4[H])     and
                          LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 1             )
                          // low range CILC adaptation active – adaptation result achieved
                          OR
                          ( STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_NOM_H_RNG” (5[H])     and
                          LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 1             )
                          // high range CILC adaptation active – adaptation result achieved
                          OR
                          ( STATE_LAM_CYL_SEL_ADJ[m] = "ADJ_ENG_COLD” (2[H])      and
                          LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] = 1                   )
                          // cold range CILC adaptation active – adaptation result achieved
            THEN (3)      LV_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0
                          // conditions for CILC adaptation no longer present
            ELSE (3)      LV_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 1
                          // conditions for CILC adaptation still present
            ENDIF (3)
        END FOR (2)
        IF (2)        LC_TCHA_CONF = 0      //no  turbo engeine
        THEN (2)
            IF (3)        $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_REQ\_DC[m]} = 0$
                          // conditions for CILC no longer achieved for all exhaust banks
            THEN (3)      LV_CYL_BAL_LAM_SEL_HOM_REQ_DC = 0
                          // reset of homogenous combustion mode request
            ENDIF (3)
        ELSE (2)      // turbo engeine
            FOR (3) m = 0 to NC_CBK_EX_NR – 1
                IF (4)        $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_SEL\_AD\_REQ\_DC[m]} = 0$
                THEN (4)      LV_CYL_BAL_AD_WG_OPEN_REQ_DC[m] = 0
                ENDIF (4)
            END FOR (3)
        ENDIF (2)
    ELSE (1)
        IF (2)        LV_T_STEP_CYL_BAL_LAM_SEL_DC = 0
        THEN (2)      T_CYL_BAL_LAM_SEL_AD_WAIT_DC(n) =
                          T_CYL_BAL_LAM_SEL_AD_WAIT_DC(n-1) – 0.1[s]
```

```
                    // decrementation of waiting time after switch to homogenous combustion mode
        ENDIF (2)
    ENDIF (1)
    // determination of cylinder individual lambda adaptation quality at active driving cycle and
    // determination of minimum adaptation time reached for cylinder individual lambda adaptation areas
    FOR m = 0 to NC_CBK_EX_NR – 1
        IF (1)          STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_NOM_L_RNG” (4[H])
        THEN (1)
            IF (2)          LV_LAM_CYL_SEL_ADJ_VLD[m] = 1                          and
                            FAC_CYL_LAM_ABSV_SUM[m] <
                                C_FAC_CYL_LAM_ABSV_SUM_BAL_DC
            THEN (2)        LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 1
            ENDIF (2)
            IF (2)          T_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] ≥
                                C_T_MIN_CYL_BAL_LAM_SEL_DC
                            // minimum adaptation time for CILC (low range) elapsed
            THEN (2)        LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m] = 1
            ELSE (2)        T_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m](n) =
                                T_CYL_BAL_LAM_SEL_AD_RNG_L_DC[m](n-1) + 0.1[s]
                            // incrementation of minimum adaptation time for CILC (low range)
            ENDIF (2)
        ELSEIF (1)      STATE_LAM_CYL_SEL_ADJ[m] = “ADJ_NOM_H_RNG” (5[H])
        THEN (1)
            IF (2)          LV_LAM_CYL_SEL_ADJ_VLD[m] = 1                          and
                            FAC_CYL_LAM_ABSV_SUM[m] <
                                C_FAC_CYL_LAM_ABSV_SUM_BAL_DC
            THEN (2)        LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 1
            ENDIF (2)
            IF (2)          T_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] ≥
                                C_T_MIN_CYL_BAL_LAM_SEL_DC
                            // minimum adaptation time for CILC (high range) elapsed
            THEN (2)        LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m] = 1
            ELSE (2)        T_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m](n) =
                                T_CYL_BAL_LAM_SEL_AD_RNG_H_DC[m](n-1) + 0.1[s]
                            // incrementation of minimum adaptation time for CILC (high range)
            ENDIF (2)
        ELSEIF (1)      STATE_LAM_CYL_SEL_ADJ[m] = "ADJ_ENG_COLD” (2[H])
        THEN (1)
            IF (2)          LV_LAM_CYL_SEL_ADJ_VLD[m] = 1                          and
```

```
                    FAC_CYL_LAM_ABSV_SUM[m] <
                            C_FAC_CYL_LAM_ABSV_SUM_BAL_DC
            THEN (2)    LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] = 1
            ENDIF (2)
            IF (2)      T_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] ≥
                            C_T_MIN_CYL_BAL_LAM_SEL_DC
                        // minimum adaptation time for CILC (cold range) elapsed
            THEN (2)    LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[m] = 1
            ELSE (2)    T_CYL_BAL_LAM_SEL_RNG_COLD_DC[m](n) =
                            T_CYL_BAL_LAM_SEL_RNG_COLD_DC[m](n-1) + 0.1[s]
                        // incrementation of minimum adaptation time for CILC (cold range)
            ENDIF (2)
        ENDIF (1)
    END FOR
```

```
// calculation of cylinder individual lambda adaptation readiness (hot) at active driving cycle (global)
IF      ( $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} LV\_CYL\_BAL\_LAM\_SEL\_AD\_RNG\_L\_DC[m] = NC\_CBK\_EX\_NR$ and
          $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} LV\_CYL\_BAL\_LAM\_SEL\_AD\_RNG\_H\_DC[m] = NC\_CBK\_EX\_NR$ )
THEN    LV_CYL_BAL_LAM_SEL_AD_HOT_DC = 1
        // hot adaptation values of all exhaust banks are within limits at active DC
ENDIF
```

```
// calculation of cylinder individual lambda adaptation readiness (cold) at active driving cycle (global)
IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} LV\_CYL\_BAL\_LAM\_SEL\_RNG\_COLD\_DC[m] = NC\_CBK\_EX\_NR$
THEN    LV_CYL_BAL_LAM_SEL_AD_COLD_DC = 1
        // cold adaptation values of all exhaust banks are within limits at active DC
ENDIF
```

Assignments at exit from state “Adaptation”:

```
LV_CYL_BAL_LAM_SEL_HOM_REQ_DC = 0
T_CYL_BAL_LAM_SEL_AD_WAIT_DC = 0.0[s]
FOR m = 0 to NC_CBK_EX_NR – 1
    T_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0.0 [s]
    LV_CYL_BAL_LAM_SEL_AD_REQ_DC[m] = 0
    LV_CYL_BAL_AD_WG_OPEN_REQ_DC[m] = 0
END FOR
```

6.11.6 Calculation of cylinder balancing lambda adaptation readiness (DC):

FUNCTION DESCRIPTION:

General information:

A periodic request for homogenous combustion mode is carried out in defined time intervals to allow the start of the lambda adaptation functions, which are only activated in homogenous combustion mode.

Application conditions:

Initialization at first Engine run or NVMY-error:

```
T_CYL_BAL_LAM_AD_DC = 0.0[s]                        (first initialization)
LV_CYL_BAL_LAM_AD_DC = 0                            (first initialization)
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_AD_ADD_DC[m] = 0                 (first initialization)
    LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0               (first initialization)
    LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 0               (first initialization)
END FOR
```

Initialization at Reset:

```
T_CYL_BAL_LAM_AD_DC = T_CYL_BAL_LAM_AD_DC (NVMY)
LV_CYL_BAL_LAM_AD_DC = LV_CYL_BAL_LAM_AD_DC (NVMY)
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_AD_ADD_DC[m] = LV_CYL_BAL_LAM_AD_ADD_DC[m] (NVMY)
    LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = LV_CYL_BAL_LAM_AD_FAC_L_DC[m] (NVMY)
    LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = LV_CYL_BAL_LAM_AD_FAC_H_DC[m] (NVMY)
END FOR
```

(NVMY: values stored in the non-volatile memory)

```
FOR m = 0 to NC_CBK_EX_NR – 1
    IF          C_CYL_BAL_LAM_AD_DC_CONF = “ADD” (0H)
    THEN        LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 1
                LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 1
    ELSEIF      C_CYL_BAL_LAM_AD_DC_CONF = “FAC_L” (1H)
    THEN        LV_CYL_BAL_LAM_AD_ADD_DC[m] = 1
                LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 1
    ELSEIF      C_CYL_BAL_LAM_AD_DC_CONF = “ADD_FAC_L” (2H)
    THEN        LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 1
    ENDIF
END FOR
```

*Initialization at Reset **or** Deactivation:*

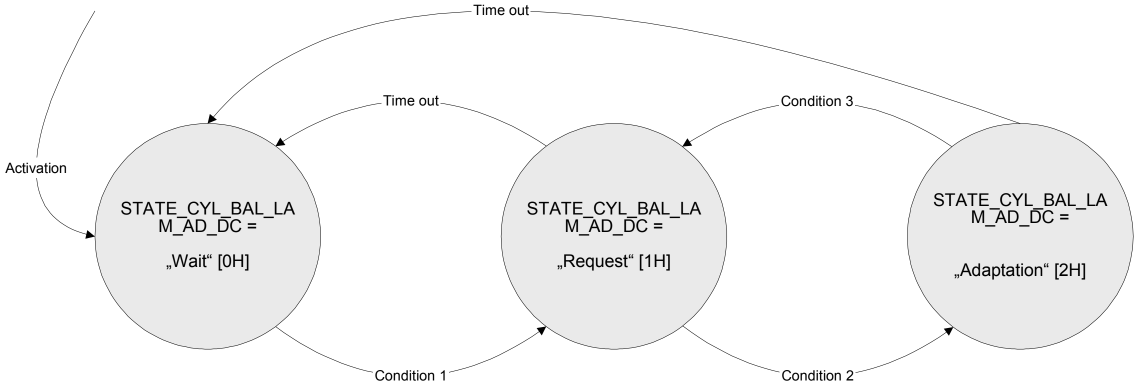
```
LV_CYL_BAL_LAM_AD_HOM_REQ_DC = 0
STATE_CYL_BAL_LAM_AD_DC = "Wait" (0[H])
LV_STATE_LAM_AD_DC_CHG_REQ = 0
LV_STATE_LAM_AD_DC_CHG_WAIT = 0
T_CYL_BAL_LAM_AD_WAIT_DC = 0.0[s]
LV_T_STEP_CYL_BAL_LAM_AD_DC = 0
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_AD_REQ_DC[m] = 0
    T_CYL_BAL_LAM_AD_ADD_DC[m] = 0.0[s]
    T_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0.0[s]
    T_CYL_BAL_LAM_AD_FAC_H_DC[m] = 0.0[s]
    T_CYL_BAL_LAM_AD_REQ_DC[m] = 0.0[s]
END FOR
```

Recurrence: 100 ms

Activation:
LC_CYL_BAL_AD_DC_RST = 0 **and**
LV_ST_END = 1 **and**
LC_TCHA_CONF = 0 **and**
LC_CYL_BAL_CORD_DC_STOP_MAN = 0

Deactivation:
LC_CYL_BAL_AD_DC_RST = 0 **and**
[LV_ST_END = 0 **or**
LC_TCHA_CONF = 1 **or**
LC_CYL_BAL_CORD_DC_STOP_MAN = 1 **]**

Signal flow diagram:



Formula section:

```
IF (1)   T_AST_BAL < C_T_AST_MAX_CYL_BAL_AD_DC          and
         TCO_ST_BAL < C_TCO_ST_MAX_CYL_BAL_AD_DC        and
```

```
                TCO_ST_BAL > C_TCO_ST_MIN_CYL_BAL_AD_DC
  THEN (1)      // DC adaptation requests controlled by fix time steps
      IF (2)            T_AST_BAL < ID_T_CYL_BAL_LAM_AD_DC
      THEN (2)          LV_STATE_LAM_AD_DC_CHG_REQ = 1
                        LV_STATE_LAM_AD_DC_CHG_WAIT = 0
                        LV_T_STEP_CYL_BAL_LAM_AD_DC = 1
      ELSE (2)          LV_STATE_LAM_AD_DC_CHG_REQ = 0
                        LV_STATE_LAM_AD_DC_CHG_WAIT = 1
                        LV_T_STEP_CYL_BAL_LAM_AD_DC = 0
      ENDIF (2)
  ELSE (1)      // DC adaptation requests for controlled by flexible time steps
      LV_T_STEP_CYL_BAL_LAM_AD_DC = 0
      IF (2)            T_CYL_BAL_LAM_AD_DC < C_T_MAX_CYL_BAL_LAM_AD_DC
      THEN (2)          T_CYL_BAL_LAM_AD_DC(n) = T_CYL_BAL_LAM_AD_DC(n-1) + 0.1[s]
                        // waiting time incrementation, flexible time steps running
          IF (3)                LV_CYL_BAL_LAM_AD_DC = 0
          THEN (3)              // enable conditions to change adaptation state from "Wait" to "Request"
                                LV_STATE_LAM_AD_DC_CHG_REQ = 1
                                LV_STATE_LAM_AD_DC_CHG_WAIT = 0
          ELSE (3)              // enable conditions to change adaptation state from "Request" to "Wait"
                                LV_STATE_LAM_AD_DC_CHG_REQ = 0
                                LV_STATE_LAM_AD_DC_CHG_WAIT = 1
          ENDIF (3)
      ELSE (2)          // reset of waiting time, new beginning of flexible time steps
          T_CYL_BAL_LAM_AD_DC = 0.0[s]
          LV_CYL_BAL_LAM_AD_DC = 0
          FOR m = 0 to NC_CBK_EX_NR – 1
              IF                C_CYL_BAL_LAM_AD_DC_CONF = “ADD” (0H)
              THEN              LV_CYL_BAL_LAM_AD_ADD_DC[m] = 0
                                T_CYL_BAL_LAM_AD_ADD_DC[m] = 0.0[s]
              ELSEIF            C_CYL_BAL_LAM_AD_DC_CONF = “FAC_L” (1H)
              THEN              LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0
                                T_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0.0[s]
              ELSEIF            C_CYL_BAL_LAM_AD_DC_CONF = “ADD_FAC_L” (2H)
              THEN              LV_CYL_BAL_LAM_AD_ADD_DC[m] = 0
                                LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0
                                T_CYL_BAL_LAM_AD_ADD_DC[m] = 0.0[s]
                                T_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0.0[s]
              EISEIF            C_CYL_BAL_LAM_AD_DC_CONF = “ADD_FAC_ALL” (3H)
```

```
                THEN    LV_CYL_BAL_LAM_AD_ADD_DC[m] = 0
                        LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0
                        LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 0
                        T_CYL_BAL_LAM_AD_ADD_DC[m] = 0.0[s]
                        T_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0.0[s]
                        T_CYL_BAL_LAM_AD_FAC_H_DC[m] = 0.0[s]
                ENDIF
            END FOR
        ENDIF (2)
    ENDIF (1)
```

Note: At function activation always State "Wait" is entered

Cylinder balancing adaptation at DC: State 0[H] – “Wait”

Assignments at entry to state “Wait”:

STATE_CYL_BAL_LAM_AD_DC = "Wait" (0[**H**])

LV_CYL_BAL_LAM_AD_HOM_REQ_DC = 0

Action during state "Wait":

Transition to State 1[H] “Request” (Condition 1)

```
// Lambda Adaptation manager request for change of state
IF      LV_STATE_LAM_AD_DC_CHG_REQ = 1
```

Assignments at exit from state “Wait”:

No assignments

Cylinder balancing adaptation at DC: State 1[H] – “Request”

Assignments at entry to state “Request”:

STATE_CYL_BAL_LAM_AD_DC = "Request" (1[**H**])

Action during state "Request":

Transition to State 0[H] “Wait” (Time out)

```
// Lambda Adaptation manager request for change of state
IF      LV_STATE_LAM_AD_DC_CHG_WAIT = 1
```

Calculations:

```
// conditions for cylinder balancing global lambda adaptation achieved
FOR m = 0 to NC_CBK_EX_NR – 1
    IF    ( [ STATE_LAM_AD[m] = "CDN_FAC_L" (2[H])          or
```

```
                STATE_LAM_AD[m] = "ADAPT_FAC_L" (5[H])          ]         and
              [ LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 0              or
                LV_T_STEP_CYL_BAL_LAM_AD_DC = 1                ]         )
            OR
            ( [ STATE_LAM_AD[m] = "CDN_FAC_H" (3[H])            or
                STATE_LAM_AD[m] = "ADAPT_FAC_H" (6[H])          ]         and
              [ LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 0              or
                LV_T_STEP_CYL_BAL_LAM_AD_DC = 1                ]         )
            OR
            ( [ STATE_LAM_AD[m] = "CDN_ADD" (4[H]) ])           or
                STATE_LAM_AD[m] = "ADAPT_ADD" (7[H])            ]         and
              [ LV_CYL_BAL_LAM_AD_ADD_DC[m] = 0                or
                LV_T_STEP_CYL_BAL_LAM_AD_DC = 1                ]         )
    THEN    T_CYL_BAL_LAM_AD_REQ_DC[m](n) =
                    T_CYL_BAL_LAM_AD_REQ_DC[m](n-1) + 0.1[s]
    ELSE    T_CYL_BAL_LAM_AD_REQ_DC[m](n) = 0.0[s]
    ENDIF
END FOR
```

```
// calculation of request for bank specific homogenuous cobustion mode switch
FOR m = 0 to NC_CBK_EX_NR – 1
    IF      T_CYL_BAL_LAM_AD_REQ_DC[m] ≥ C_T_MIN_CYL_BAL_AD_REQ_DC
    THEN    LV_CYL_BAL_LAM_AD_REQ_DC[m] = 1
    ELSE    LV_CYL_BAL_LAM_AD_REQ_DC[m] = 0
    ENDIF
END FOR
```

```
// calculation of request flag for EVAC controller to allow activation of lambda adaptation
IF (1)      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{T\_CYL\_BAL\_LAM\_AD\_REQ\_DC}[m] > 0.0[s]$
THEN (1)    LV_CYL_BAL_LAM_AD_ENA_DC = 1
ELSE (1)    LV_CYL_BAL_LAM_AD_ENA_DC = 0
ENDIF (1)
```

```
// calculation of request for homogenous combustion mode switch
IF (1)      ( LV_CYL_BAL_LAM_AD_ENA_DC = 1                 and
              LV_NT_RGN_REQ = 1                            and
              LV_LAM_AD_ENA = 1                   )
            OR
            ( LV_LAM_AD_ENA = 1                            and
```

```
                    $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} LV\_CYL\_BAL\_LAM\_AD\_REQ\_DC[m] > 0$      )
              OR
              ( LV_T_STEP_CYL_BAL_LAM_AD_DC = 1                 and
                LV_CYL_BAL_LAM_AD_ENA_DC = 1                    and
                LV_LAM_AD_ENA = 1                           )
THEN (1)      LV_CYL_BAL_LAM_AD_HOM_REQ_DC = 1
ENDIF (1)
```

<u>Transition to State 2[H] “Adaptation” (Condition 2)</u>

```
// conditions for combustion mode switch to homogenuous
IF            LV_CYL_BAL_LAM_AD_HOM_REQ_DC = 1
```

Assignments at exit from state “Request”:

```
LV_CYL_BAL_LAM_AD_ENA_DC = 0
FOR m = 0 to NC_CBK_EX_NR – 1
    LV_CYL_BAL_LAM_AD_REQ_DC[m] = 0
    T_CYL_BAL_LAM_AD_REQ_DC[m] = 0
END FOR
```

Cylinder balancing adaptation at DC: State 2[H] – “Adaptation”

Assignments at entry to state “Adaptation”:

```
STATE_CYL_BAL_LAM_AD_DC = "Adaptation" (2[H])
T_CYL_BAL_LAM_AD_WAIT_DC = C_T_MIN_CYL_BAL_AD_WAIT_DC
FOR m = 0 to NC_CBK_EX_NR – 1
    T_CYL_BAL_LAM_AD_REQ_DC[m] = C_T_MIN_CYL_BAL_AD_WAIT_DC
END FOR
```

Actions during state "Adaptation":

<u>Transition to State 0[H] “Wait” (Time out)</u>

```
// Lambda Adaptation manager request for change of state
IF            LV_STATE_LAM_AD_DC_CHG_WAIT = 1
```

<u>Transition to State 1[H] “Request” (Condition 3)</u>

```
// conditions for combustion mode switch to homogenuous or waste gate open not achieved
IF            LV_CYL_BAL_LAM_AD_HOM_REQ_DC = 0                  or
              LV_LAM_AD_ENA = 0
```

Calculations:

```
// calculation of waiting time after entry in homogenuous combustion mode for non turbo engine
IF (1)      T_CYL_BAL_LAM_AD_WAIT_DC = 0.0[s]
            // waiting time after switch to homogenous combustion mode elapsed
THEN (1)
   FOR (2)  m = 0 to NC_CBK_EX_NR – 1
            IF (3)      ( LV_MFF_ADD_RNG_LAM_AD[m] = 0          and
                          LV_FAC_L_RNG_LAM_AD[m] = 0            and
                          LV_FAC_H_RNG_LAM_AD[m] = 0            )
                        // neither low range, nor high range,nor add range lambda adaptation area active
            THEN (3)    T_CYL_BAL_LAM_AD_REQ_DC[m](n) =
                               T_CYL_BAL_LAM_AD_REQ_DC[m](n-1)  - 0.1 [s]
            ELSE (3)    T_CYL_BAL_LAM_AD_REQ_DC[m](n) =
                               C_T_MIN_CYL_BAL_AD_WAIT_DC
            ENDIF (3)
            IF (3)      T_CYL_BAL_LAM_AD_REQ_DC[m] = 0.0 [s]
                        // time with no lamba adaptation area active elapsed
                        OR
                        ( LV_MFF_ADD_RNG_LAM_AD[m] = 1          and
                         LV_CYL_BAL_LAM_AD_ADD_DC[m] = 1 )
                        // additive range lambda adaptation active – adaptation result achieved
                        OR
                        ( LV_FAC_L_RNG_LAM_AD[m] = 1            and
                          LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 1 )
                        // low range lambda adaptation active – adaptation result achieved
                        OR
                        ( LV_FAC_H_RNG_LAM_AD[m] = 1            and
                          LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 1 )
                        // high range lambda adaptation active – adaptation result achieved
            THEN (3)    LV_CYL_BAL_LAM_AD_REQ_DC[m] = 0
                        // conditions for lambda adaptation no longer present
            ELSE (3)    LV_CYL_BAL_LAM_AD_REQ_DC[m] = 1
                        // conditions for lambda adaptation still present
            ENDIF (3)
   END FOR (2)
   IF (2)         $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} \mathrm{LV\_CYL\_BAL\_LAM\_AD\_REQ\_DC[m]} = 0$
                  // conditions for lambda adaptation no longer achieved for all exhaust banks
   THEN (2)       LV_CYL_BAL_LAM_AD_HOM_REQ_DC = 0
                  // reset of homogenous combustion mode request
```

```
        ENDIF (2)
    ELSE (1)
        IF (2)          LV_T_STEP_CYL_BAL_LAM_AD_DC = 0
        THEN (2)        T_CYL_BAL_LAM_AD_WAIT_DC(n) = T_CYL_BAL_LAM_AD_WAIT_DC(n-1) – 0.1[s]
                        // decrementation of waiting time after switch to homogenous combustion mode
        ENDIF (2)
    ENDIF (1)
```

```
    // determination of  learned lambda adaptation values
    FOR m = 0 to NC_CBK_EX_NR – 1
        IF (1)          LV_CYL_BAL_LAM_AD_END[m] = 1
        THEN (1)
            LV_CYL_BAL_LAM_AD_END[m] = 0
            IF (2a)         STATE_CYL_BAL_LAM_AD[m] = "ADAPT_ADD" (7[H])
            THEN (2a)       LV_CYL_BAL_LAM_AD_ADD_DC[m] = 1
            ELSEIF (2b)     STATE_CYL_BAL_LAM_AD[m] = "ADAPT_FAC_L" (5[H])
            THEN (2b)       LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 1
            ELSEIF (2c)     STATE_CYL_BAL_LAM_AD[m] = "ADAPT_FAC_H" (6[H])
            THEN (2c)       LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 1
            ENDIF (2)
        ENDIF (1)
    END FOR
```

```
    // determination of minimum adaptation time reached for lambda adaptation areas
    FOR m = 0 to NC_CBK_EX_NR – 1
        IF (1a)         LV_MFF_ADD_RNG_LAM_AD[m] = 1
        THEN (1a)
            IF (2a) T_CYL_BAL_LAM_AD_ADD_DC[m] ≥ C_T_MIN_CYL_BAL_LAM_AD_DC
                    // minimum adaptation time for lambda adaptation (add range) elapsed
            THEN (2a)       LV_CYL_BAL_LAM_AD_ADD_DC[m] = 1
            ELSE (2a)       T_CYL_BAL_LAM_AD_ADD_DC[m](n) =
                                    T_CYL_BAL_LAM_AD_ADD_DC[m](n-1) + 0.1[s]
                            // incrementation of min. adaptation time for LAM_AD (add range)
            ENDIF (2a)
        ELSEIF (1b)     LV_FAC_L_RNG_LAM_AD[m] = 1
        THEN (1b)
            IF (2b) T_CYL_BAL_LAM_AD_FAC_L_DC[m] ≥ C_T_MIN_CYL_BAL_LAM_AD_DC
                    // minimum adaptation time for lambda adaptation (fac low range) elapsed
            THEN (2b)       LV_CYL_BAL_LAM_AD_FAC_L_DC[m] = 1
            ELSE (2b)       T_CYL_BAL_LAM_AD_FAC_L_DC[m](n) =
                                    T_CYL_BAL_LAM_AD_FAC_L_DC[m](n-1) + 0.1[s]
```

```
                        // incrementation of min. adaptation time for LAM_AD (fac low range)
                ENDIF (2b)
        ELSEIF (1c)     LV_FAC_H_RNG_LAM_AD[m] = 1
        THEN (1c)
                IF (2c) T_CYL_BAL_LAM_AD_FAC_H_DC[m] ≥ C_T_MIN_CYL_BAL_LAM_AD_DC
                        // minimum adaptation time for lambda adaptation (fac high range) elapsed
                THEN (2c)       LV_CYL_BAL_LAM_AD_FAC_H_DC[m] = 1
                ELSE (2c)       T_CYL_BAL_LAM_AD_FAC_H_DC[m](n) =
                                        T_CYL_BAL_LAM_AD_FAC_H_DC[m](n-1) + 0.1[s]
                        // incrementation of min. adaptation time for LAM_AD (fac high range)
                ENDIF (2c)
        ENDIF (1)
END FOR
```

```
// calculation of lambda adaptation readiness at active driving cycle (global)
IF      $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} LV\_CYL\_BAL\_LAM\_AD\_ADD\_DC[m] = NC\_CBK\_EX\_NR$    and
        $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} LV\_CYL\_BAL\_LAM\_AD\_FAC\_L\_DC[m] = NC\_CBK\_EX\_NR$    and
        $\sum_{m=0}^{m=NC\_CBK\_EX\_NR-1} LV\_CYL\_BAL\_LAM\_AD\_FAC\_H\_DC[m] = NC\_CBK\_EX\_NR$
THEN    LV_CYL_BAL_LAM_AD_DC = 1
        // adaptation values of all exhaust banks are within limits at active driving cycle
ENDIF
```

Assignments at exit from state “Adaptation”:

```
LV_CYL_BAL_LAM_AD_HOM_REQ_DC = 0
T_CYL_BAL_LAM_AD_WAIT_DC = 0.0[s]
FOR m = 0 to NC_CBK_EX_NR – 1
    T_CYL_BAL_LAM_AD_REQ_DC[m] = 0.0 [s]
    LV_CYL_BAL_LAM_AD_REQ_DC[m] = 0
END FOR
```

6.12 Customer adaptation module: AGGR CYBL

6.12.1 Outputs for BMW which are not defined as CYBL exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_gl_ad	O/V	0...1H	0...1	1	[-]
Lambdaadaption am Bandende hat fertig gelernt					
B_gl_adder	O/V	0...1H	0...1	1	[-]
ER_Balancing am Bandende hat additiv adaptiert					
B_gl_adz	O/V	0...1H	0...1	1	[-]
Lambdaadaption ist nötig, zyklisch während Motorbetrieb zu 1 gesetzt					
B_gl_adz_offset1	O/V	0...1H	0...1	1	[-]
Zylinderselektive Adaption vom Lambda-Offset auf Bank1 ist erfolgt.					
B_gl_adz_offset2	O/V	0...1H	0...1	1	[-]
Zylinderselektive Adaption vom Lambda-Offset auf Bank 2 ist erfolgt.					
B_gl_hz	O/V	0...1H	0...1	1	[-]
Zylindersel. Lambdaregelung fordert homogen an, zyklisch während dem Motorbetrieb zu 1 gesetzt					
B_gl_zsk	O/V	0...1H	0...1	1	[-]
Zylindersel. Lambdaregelung kalt am Bandende hat fertig adaptiert					
B_gl_zsw	O/V	0...1H	0...1	1	[-]
Zylindersel. Lambdaregelung warm am Bandende hat fertig adaptiert					
B_gl_zswz	O/V	0...1H	0...1	1	[-]
Zylindersel. Lambdaregelung warm ist nötig, zyklisch während Motorbetrieb zu 1 gesetzt					
B_gl_wg1	O/V	0...1H	0...1	1	[-]
Zylindersel. Lambdaregelung fordert öffne WG1 an, zyklisch während dem Motorbetrieb zu 1 gesetzt					
B_gl_wg2	O/V	0...1H	0...1	1	[-]
Zylindersel. Lambdaregelung fordert öffne WG2 an, zyklisch während dem Motorbetrieb zu 1 gesetzt					
B_injad_anf	O/V	0...1H	0...1	1	[-]
Anforderung Kleinstmengenadaption					
B_lamzylact_1	O/V	0...1H	0...1	1	[-]
Bedingung CILC aktiv (Bank 1)					
B_lamzylact_2	O/V	0...1H	0...1	1	[-]
Bedingung CILC aktiv (Bank 2)					
B_lurzylact	O/V	0...1H	0...1	1	[-]
Bedingung CYBL aktiv					
F_tizyl[NC_CYL_NR]	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
zylinderindividueller Einspritzfaktor					
FAC_MFF_COR_EXT_LAM_AD_INJ[NC_CYL_NR]	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
External fuel mass correction factor (combustion controller)					
LV_INH_BAL_CUS	O/V	0...1H	0...1	1	[-]
CILC and CYBL inhibt by external request					

Input data:

LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	FAC_TI_BAL[NC_CYL_NR]	LV_CYL_BAL_LAM_AD_ADD_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_EOL
LV_CYL_BAL_ER_AD_ADD_EOL	LV_CYL_BAL_LAM_AD_DC	LV_CYL_BAL_AD_HOM_REQ_DC	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	LV_CYL_BAL_LAM_SEL_AD_HOT_DC	LV_CYL_BAL_AD_WG_OPEN_REQ[NC_CBK_EX_NR]	LV_LAM_CYL_ENA[NC_CBK_EX_NR]
LV_TI_CYL_BAL_ER_ACT	STATE_LAM_AD_INJ_ACT[NC_CBK_EX_NR][1][1]	B_ext_zylstop	F_tikorzsk[NC_CYL_NR]

FUNCTION DESCRIPTION:

Adaption to BMW environment.

Application conditions:

Initialisation at reset or at exit power latch phase:

1: B_gl_ad, B_gl_adder, B_gl_zsk, B_gl_zsw

0: B_gl_adz, B_gl_hz, B_gl_zswz , B_gl_wg1, B_gl_wg2, B_lamzylact_1, B_lamzylact_2, B_lurzylact, B_injad_anf, B_gl_adz_offset1, B_gl_adz_offset2

1.0: F_tizyl[NC_CYL_NR]

Recurrence:
10 ms: B_gl_ad, B_gl_adder, B_gl_adz, B_gl_hz
B_gl_zsk, B_gl_zsw, B_gl_zswz, B_gl_wg1, B_gl_wg2
20ms: B_injad_anf
100ms: F_tizyl[NC_CYL_NR], B_lamzylact_1, B_lamzylact_2,
B_lurzylact, B_gl_adz_offset1, B_gl_adz_offset2

Activation: every engine state

Formula section:

```
If   ECU_STATE = "PWL"
 Then
      B_gl_ad        = 1
      B_gl_adder     = 1
      B_gl_zsk       = 1
      B_gl_zsw       = 1
      B_gl_adz       = 0
      B_gl_hz        = 0
      B_gl_zswz      = 0
      B_gl_wg1       = 0
      B_gl_wg2       = 0
      B_injad_anf    = 0

Else
     B_gl_ad = LV_CYL_BAL_LAM_AD_EOL
     B_gl_adder = LV_CYL_BAL_ER_AD_ADD_EOL
     B_gl_adz = LV_CYL_BAL_LAM_AD_DC
     B_gl_adz_offset1 = LV_CYL_BAL_LAM_AD_ADD_DC[1]
     B_gl_adz_offset2 = LV_CYL_BAL_LAM_AD_ADD_DC[2]
     B_gl_hz = LV_CYL_BAL_AD_HOM_REQ_DC
     B_gl_zsk = LV_CYL_BAL _LAM_SEL_AD_COLD_EOL
     B_gl_zsw = LV_CYL_BAL _LAM_SEL_AD_HOT_EOL
```

```
        B_gl_zswz = LV_CYL_BAL_LAM_SEL_AD_HOT_DC
        B_gl_wg1 = LV_CYL_BAL_AD_WG_OPEN_REQ[1]
        B_gl_wg2 = LV_CYL_BAL_AD_WG_OPEN_REQ[2]
        B_lamzylact_1 = LV_LAM_CYL_ENA[1]
        B_lamzylact_2 = LV_LAM_CYL_ENA[2]
        B_lurzylact    = LV_TI_CYL_BAL_ER_ACT
        F_tizyl[NC_CYL_NR] = FAC_TI_BAL[NC_CYL_NR]
        if      STATE_LAM_AD_INJ_ACT[1] > 0
             or STATE_LAM_AD_INJ_ACT[2] > 0
        then    B_injad_anf = 1
        else    B_injad_anf = 0
        endif
    Endif
```

6.12.2 Outputs for SV aggregates

<u>FUNCTION DESCRIPTION:</u>

Adaptation to BMW environment.

<u>Application conditions:</u>

Initialisation at reset or exit "PWL": 0

Recurrence :	100 ms:	LV_INH_BAL_CUS
	Segment:	FAC_MFF_COR_EXT_LAM_AD_INJ[NC_CYL_NR]
Activation :	every engine state	

<u>Formula section:</u>

```
        LV_INH_BAL_CUS    = B_ext_zylstop
        FAC_MFF_COR_EXT_LAM_AD_INJ[NC_CYL_NR] =  F_tikorzsk[NC_CYL_NR]
```

7 Diagnostic communication

Table of Contents

Index

7.1 Keyword Protocol 2000

7.1.1 Normative reference

This implementation and the description of diagnostic communication protocol KWP 2000 is based on the following standards:

ISO/FDIS 14229, 1996	Road vehicles-Diagnostic systems - Diagnostic Services Specification
ISO/FDIS 14230-1, 1996	Road vehicles-Diagnostic systems Keyword Protocol 2000 part 1 : Physical Layer
ISO/FDIS 14230-2, 1996	Road vehicles-Diagnostic systems Keyword Protocol 2000 part 2 : Data Link Layer
ISO/FDIS 14230-3, 1996	Road vehicles-Diagnostic systems Keyword Protocol 2000 part 3 : Implementation
ISO/DIS 14230-4	Road vehicles-Diagnostic systems Keyword Protocol 2000 part 4 : KWP2000 requirements for emission-related systems
VDA/WD 14230-2, 1997	Keyword Protocol 2000 part 2 : Data Link Layer, VDA-recommended practice
VDA/WD 14230-3, 1997	Keyword Protocol 2000 part 3 : Implementation, VDA-recommended practice

7.1.2 Customers reference

8385772.4	1998	BMW specification: KWP2000: Physical Layer
8385773.4	1998	BMW specification: KWP2000: Data Link Layer
8385774.4	1998	BMW specification: KWP2000: Diagnostic Services
1 430 227	1999	BMW specification: Diame9: KWP* Services

7.2 Serial Communication - Introduction

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ANG_DE_ABSV_PLAUS_CHK_VVL_KWP	O/V	0...FFH	0...179.29687	0.703125	[°]
Absolute value of angle deviation between sensor 1 and 2 for plausibility check converted for KWP					
ANG_EXC_VVL_KWP	O/V	0...FFH	0...199.21875	0.78125	[°]
Selected current value in grad after jitter filtration converted for KWP					
ANG_SP_CTL_VVL_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
Setpoint for the position controller converted for KWP					
CAM_IN_KWP	O/V	0...FFH	60...155.625	0.375	[°CRK]
Actual position inlet CAM VVTI converted for KWP					
CAM_SP_IVVT_IN_KWP	O/V	0...FFH	60...155.625	0.375	[°CRK]
Inlet CAM setpoint for VVTI converted for KWP					
CPU_LOAD_RST_DET_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
CPU_LOAD_RST_DET[0] converted to KWP					
CTR_KM_CAN_KWP	O/V	0...FFH	0...652800	2560	[km]
Vehicle kilometer reading from CAN message INSTR2, converted for KWP					
CUR_DMTL_COR_FIL_KWP	O/V	0...FFH	0...49.80468	0.1953125	[mA]
Corrected and filtered pump current converted for KWP					
CUR_DMTL_REF_LEAK_KWP	O/V	0...FFH	0...49.80468	0.1953125	[mA]
Pump current reference leakage converted for KWP					
CUR_DMTL_ROUGH_LEAK_MIN_KWP	O/V	0...FFH	0...49.80468	0.1953125	[mA]
Min. pump current in case of rough leak measurement converted for KWP					
CUR_MOT_VVL_3_KWP	O/V	80...7FH	-128...127	1	[A]
Actual signed motor current converted for KWP					
DIST_SO2P_END_KWP	O/V	0...FFH	0...10200	40	[km]
Km - counter since last end of desulfation converted for KWP					
EFPPWM_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
Pump speed of the electrical fuel pump as PWM signal converted for KWP					
FAC_MV_DIAG_DYN_LSL_UP_KWP[NC_CBK_EX_NR]	O/V	0...FFH	0...1.02	0.004	[-]
Mean value of normalised single sensor signal amplitude converted for KWP					
FUP_EFP_KWP	O/V	0...FFH	0...10825.7088	42.45376	[hPa]
Fuel pressure EFP converted for KWP					
FUP_KWP[NC_CBK_HPP_NR]	O/V	0...FFH	0...346422	1358.5176	[hPa]
Fuel pressure converted for KWP					
LAMB_DELTA_I_LAM_ADJ_KWP[NC_CBK_EX_NR]	O/V	80...7FH	-0.125...0.12402	0.9766e-3	[-]
LAMB_DELTA_I_LAM_ADJ converted for KWP					
LAMB_DIF_MON_KWP	O/V	80...7FH	-2...1.98437	0.015625	[-]
Deviation between actual and modeled Lambda for KWP					
LAMB_KWP[NC_CBK_EX_NR]	O/V	0...FFH	0...1.99218	0.0078125	[-]
Signal value of lambda sensor					
LAMB_LS_UP_MV_KWP	O/V	0...FFH	0...3.98437	0.015625	[-]
Mean value of lambda in multiple-branched exhaust gas lines converted for KWP					
LAMB_NOX_SENS_KWP	O/V	0...FFH	0...3.98437	0.015625	[-]
Lambda value, measured by NOx sensor converted for KWP					
LAMB_SP_KWP[NC_CBK_EX_NR]	O/V	0...FFH	0...1.99218	0.0078125	[-]
Lambda setpoint low byte value					
LOAD_MIN_MIS_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
Nulllastwert für Aussetzererkennung					
LOAD_MIS_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
Lastwert für Aussetzererkennung					
MFF_SP_MV_KWP	O/V	0...FFH	0...1389	5.4470588	[mg/stk]
Mass fuel flow setpoint after combustion selection converted for KWP					
N_RST_DET_KWP	O/V	0...FFH	0...8160	32	[rpm]
N_RST_DET[0] converted to KWP					

N_SAE_BYTE_KWP	O/V	0...FFH	0...16320	64	[rpm]
Engine speed byte SAE J1979					
NOX_NS_KWP	O/V	FA...5DH	-96...1488	16	[ppm]
NOx concentration value, measured by NOx-Sensor for KWP					
NT_AGI_KWP	O/V	0...FFH	0...1	3.9216e-3	[-]
NOx trap aging factor converted for KWP					
NT_AGI_SUL_KWP	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
NOx trap aging factor due to sulphur load converted for KWP					
NT_SUL_KWP	O/V	0...FFH	0...10444.8	40.96	[mg]
NOx trap sulphur loading					
OBD_AMP	O/V	0...FFH	0...255	1	[kPA]
Barometric pressure SAE J1979					
OBD_FUP	O/V	0...FFH	0...765	3	[kPA]
Fuel pressure SAE J1979					
OBD_FUP_RNG_H	O/V	0...FFFFH	0...655350	10	[kPA]
High range Fuel pressure SAE J1979					
OBD_FUP_RNG_H_H	O/V	0...FFH	0...652800	2560	[kPA]
High byte of OBD_FUP_RNG_H					
OBD_FUP_RNG_H_L	O/V	0...FFH	0...2550	10	[kPA]
Low byte of OBD_FUP_RNG_H					
OBD_IGA_IGC	O/V	0...FFH	-64...63.5	0.5	[°CRK]
Ignition time cylinder 1 SAE J1979					
OBD_LAM_COR[NC_CBK_EX_NR]	O/V	0...FFH	-100...99.21875	0.78125	[%]
Lambda control factor SAE J1979					
OBD_LAMB_SP	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Lambda setpoint for display in mode 01 (PID $44)					
OBD_MAF	O/V	0...FFFFH	0...655.35	0.01	[g/s]
Air mass SAE J1979					
OBD_N	O/V	0...7F80H	0...8160	0.25	[rpm]
Engine speed SAE J1979					
OBD_TCO	O/V	0...FFH	-40...215	1	[°C]
Engine coolant temperature SAE J1979					
OBD_TIA	O/V	0...FFH	-40...215	1	[°C]
Intake air temperature 1 SAE J1979					
OBD_VB	O/V	0...FFFFH	0...65.535	0.001	[V]
Control module voltage SAE J1979					
OPG_ACR_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
Measured opening of the actuator valve converted for KWP					
OPG_SP_ACR_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
Setpoint request after limitation for actuator position control converted for KWP					
P_OEL_IST_KWP	O/V	0...FFH	0...8160	32	[hPa]
Oil pressure IST-value converted to KWP					
P_OEL_SOLL_KWP	O/V	0...FFH	0...8160	32	[hPa]
Oil pressure soll-value converted to KWP					
PBSU_KWP	O/V	0...FFH	0...1353.19335	5.3066406	[hPa]
Absolute pressure in the brake servo, converted for KWP					
PUT_KWP[NC_MAP_SENS_NR]	O/V	0...FFH	0...255	1	[kPA]
Pressure upstream the throttle (Turbo) converted for KWP					
PWM_ACR_KWP	O/V	80...7FH	-100...99.21875	0.78125	[%]
Finally duty cycle of digital actuator control converted to KWP					
PWM_DR_OUT_SET_VVL_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
PWM to be set by PWM generator converted for KWP					
PWM_NEUT_PSN_GB_KWP	O/V	0...FFH	0...99.60937	0.390625	[%]
Tastverhältnis Nullgangsensor converted for KWP					
RATIO_NS_SHIFT_DIAG_KWP[NC_NOX_SENS_CONF]	V	80...7FH	-1...0.99218	0.0078125	[-]
NOx signal shift diagnosis value converted for diagnostic communication					
R_IT_LS_DOWN_KWP_H[NC_CBK_EX_NR]	O/V	0...FFH	0...65280	256	[Ohm]
Internal resistance of downstream oxygen sensor (measured) - high byte					

R_IT_LS_DOWN_KWP_L[NC_CBK_EX_NR]	O/V	0...FFH	0...255	1	[Ohm]
Internal resistance of downstream oxygen sensor (measured) - low byte					
R_IT_LS_UP_KWP_H[NC_CBK_EX_NR]	O/V	0...FFH	0...16320	64	[Ohm]
Internal resistance of upstream oxygen sensor (measured) - high byte					
R_IT_LS_UP_KWP_L[NC_CBK_EX_NR]	O/V	0...FFH	0...63.75	0.25	[Ohm]
Internal resistance of upstream oxygen sensor (measured) - low byte					
STATE_BYTE_SWI_KWP	O/V	0...FFH	0...255	1	[-]
Carrierbyte switch states for KWP					
STATE_ECRAS_SYS_KWP_H	O/V	0...FFH	0...255	1	[-]
Status Luftklappensystem High Byte					
STATE_ECRAS_SYS_KWP_L	O/V	0...FFH	0...255	1	[-]
Status Luftklappensystem Low Byte					
STATE_LRN_ECU_KWP	O/V	only MSD8x: FFH 11H 5AH A5H 4EH AEH BCH A2H 2AH E4H 0H only MSV80: 0H 11H 3DH 5AH 79H 97H A5H AEH BCH CBH D3H EAH FFH	only MSD8x ROM_NOT_PLAUS LEARNING_FAILED C1_LOT1 C1_LOT2 C1_LOT3 C1_LOT4 SERIAL_ECU C2_LOT1 C2_LOT2 C2_LOT3 NOT_LEARNED only MSV80: ROM_PLAUS LEARNING_FAILED LOT8 LOT1 LOT5 LOT9 LOT2 LOT4 SERIAL_ECU LOT6 LOT7 LOT3 NOT_LEARNED	1	-
Variant ECU					

STATE_N_MAX_MON_HIGH_KWP	O/V	0...FFH	0...255	1	[-]
STATE_N_MAX_MON - only high byte					
STATE_N_MAX_MON_LOW_KWP	O/V	0...FFH	0...255	1	[-]
STATE_N_MAX_MON - only low byte					
T_CH_SO2P_ACT_KWP	O/V	0...FFH	0...6528	25.6	[s]
Time with active catalyst heating for desulfation without active desulfation converted for KWP					
T_ES_CUS_KWP	O/V	0...FFH	0...1020	4	[min]
Engine shut-off duration, not deleted at ST_END					
T_ES_KWP	O/V	0...FFH	0...1020	4	[min]
Engine shut-off duration					
T_PER_MAF_FRQ_KWP	O/V	0...FFH	0...8160	32	[µs]
arithmetric average of the period time converted for KWP					
T_PER_MAF_KWP	O/V	0...FFH	0...1.02	0.004	[ms]
period time of air mass converted for KWP					
TCHIP_KWP	O/V	0...FFH	-48...207	1	[°C]
chip temperature of alternator converted for KWP					
TEMP_MOT_VVL_KWP	O/V	E8...7FH	-48...254	2	[°C]
Estimated DC motor temperature converted for KWP					
TEMP_SWI_MES_MAX_VVL_KWP	O/V	80...7FH	-256...254	2	[°C]
Maximal measured temperature of the chopper switch transistors converted for KWP					
TIA_TCHA_KWP	O/V	0...FFH	-48...207	1	[°C]
Air temperature up turbo charger converted for KWP					
TPS_AD_STEP_KWP[NC_ETC_NR]	O/V	0...FFH	0...255	1	[-]
TPS_AD_STEP shifted to BYTE from DWORD					
TQI_SP_KWP	O/V	80...7FH	-256...254	2	[Nm]
Torque setpoint fast path for KWP					
V_ALTER_SP_KWP	O/V	0...FFH	0...25.5	0.1	[V]
setpoint alternator 1 voltage converted for KWP					
V_AMP_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage V_AMP shifted toi FFC0H					
V_DMTL_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage V_DMTL shifted to KWP					
V_FUP_EFP_MV_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
mean value of low fuel pressure EFP sensor converted for KWP					
V_FUP_MV_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Mean value of the acquired sensor voltage converted for KWP					
V_IGK_BAS_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage V_IGK_BAS shifted to FFC0H					
V_PBSU_KWP	O/V	0...FFH	0...4.98046	0.0195312	[V]
Spannung Bremsunterdrucksensor converted to KWP					
V_PUT_KWP[NC_MAP_SENS_NR]	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage of the intake manifold pressure sensor up throttle (for diagnosis) converted for KWP					
V_PVS_1_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Pedal value sensor 1 raw acquisition converted for KWP					
V_PVS_1_MON_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Pedal value sensor 1 raw acquisition monitoring value converted for KWP					
V_PVS_2_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Pedal value sensor 2 raw acquisition converted for KWP					
V_PVS_2_MON_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Pedal value sensor 2 raw acquisition monitoring value converted for KWP					
V_SOF_SWI_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage V_SOF_SWI_KWP - ADC-value					
V_TPS_1_KWP[NC_ETC_NR]	O/V	0...FFH	0...4.98046	0.0195313	[V]
Throttle position sensor 1 raw acquisition converted for KWP					
V_TPS_2_KWP[NC_ETC_NR]	O/V	0...FFH	0...4.98046	0.0195313	[V]
Throttle position sensor 2 raw acquisition converted for KWP					
V_VCC_SENS_VVL_KWP	O/V	0...FFH	0...25.5	0.1	[V]
Sensor supply voltage converted for KWP					
VB_BAS_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage VB_BAS shifted to FFC0H					
VCC_DR_VVL_KWP	O/V	0...FFH	0...25.5	0.1	[V]

Supply voltage filtered with fast filter converted for KWP					
VCC_PVS_1_KWP	O/V	0...FFH	0...9.96093	0.0390625	[V]
Voltage supply for pedal value sensor 1 raw acquisition converted for KWP					
VCC_PVS_2_KWP	O/V	0...FFH	0...9.96093	0.0390625	[V]
Voltage supply for pedal value sensor 2 raw acquisition converted for KWP					
VFF_MFF_SP_FUP_CTL_KWP	O/V	0...FFH	0...255	1	[l/h]
Volume fuel flow through the injectors converted for KWP					
VLS_DIF_LAM_ADJ_KWP[NC_CBK_EX_NR]	O/V	80...7FH	-0.625...0.62011	4.8828e-3	[V]
difference between set point and actual downstream LS signal converted for KWP					
VLS_DOWN_KWP[NC_CBK_EX_NR]	O/V	0...FFH	0...4.98046	0.0195313	[V]
Downstream oxygen sensor voltage converted for KWP					
VLS_NOX_SENS_KWP[NC_NOX_SENS_CONF]	O/V	0...578H	-200...1200	1	[mV]
Binary O2 signal voltage, raw value, measured by NOx-Sensor for KWP					
VLS_OFS_LSL_KWP[NC_CBK_EX_NR]	O/V	80...7FH	-0.625...0.62011	4.8828e-3	[V]
Output signal offset of linear lambda sensor for Ip gain16 converted for KWP					
VLS_UP_KWP[NC_CBK_EX_NR]	O/V	0...FFH	0...4.98046	0.0195313	[V]
Upstream Oxygen sensor voltage converted for KWP					
VP_TCO_2_KWP[NC_NR_TCO_SENS]	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage of coolant temperature sensor (radiator outlet) converted for KWP					
VP_TCO_KWP[NC_NR_TCO_SENS]	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage of coolant temperature sensor shifted to FFC0H					
VP_TECU_KWP	O/V	0...FFH	0...4.98046	0.0195313	[V]
Voltage VP_TECU shifted to FFC0H					
VP_TEG_PCAT_DOWN_KWP	O/V	0...FFH	0...3.3	0.0129412	[V]
Catalyst temperature sensor raw acquisition converted for KWP					
VP_TIA_KWP[NC_SENS_NR_TIA_THR]	O/V	0...FFH	0...4.98046	0.0195313	[V]
VP_TIA converted for KWP					
VP_TIA_TCHA_KWP	O/V	0...FFH	0...3.3	0.0129412	[V]
Voltage for temperatur sensor up turbocharger converted for KWP					
TQI_SP_H_RNG_MON_KWP	O/V	0...FFH	0...1020	4	[°C]
TQI_SP_H_RNG_MON converted for KWP					
TQI_AV_H_RNG_MON_KWP	O/V	0...FFH	0...1020	4	[°C]
TQI_AV_H_RNG_MON converted for KWP					

Input data:

Absch_korr	AMP_MES	ANG_EXC_VVL	ANG_REL_VVL
ANG_SP_CTL_VVL	ANG_SP_EXT_ADJ_VVL	ANG_SP_VVL	ANG_SP_VVL_CUS
ANG_STK_VVL	Atlr	Atlvst	B_bsdprot2
B_dmeini	B_lrfoff	B_vsean_loc	Ba_ist
Ba_wm_ist	BRAKE_PRS	Bsdgencv	Bsdgenregv
C_CPPWM_ADD_AD_MEM	CAM_EX[NC_NR_CBK_IVVT]	CAM_EX[NC_NR_CBK_IVVT]	CAM_IN[NC_NR_CBK_IVVT]
CAM_IN_H[NC_NR_CBK_IVVT]	CAM_IN_H[NC_NR_CBK_IVVT]	CAM_IN_H[NC_NR_CBK_IVVT]	CAM_IN_H[NC_NR_CBK_IVVT]
CAM_SP_IVVT_EX	CAM_SP_IVVT_EX	CAM_SP_IVVT_EX	CAM_SP_IVVT_IN
CAM_SP_IVVT_IN	CAM_SP_IVVT_IN	CAM_SP_REF_EX	CAM_SP_REF_EX
CAM_SP_REF_EX	CAM_SP_REF_IN	CAM_SP_REF_IN	CAM_SP_REF_IN
CAN_ERR_NOX_SENS[NC_NOX_SENS_CONF]	CAN_STATE_NOX_SENS[NC_NOX_SENS_CONF]	CAT_DIAG[NC_CBK_EX_NR]	CHA_IV_1_MES[NC_CYL_NR]
CL_MMV_SAE	CPPWM_CPS	CPU_LOAD_MAX_RST_DET[NC_NR_DBG_NVMY]	CPU_LOAD_RST_DET[NC_NR_DBG_NVMY]
CTR_AD_COLD_LAM_AD_INJ	CTR_AD_HOT_LAM_AD_INJ	CTR_CH_SO2P	CTR_ERR_LSL_IF_SPI_WR[NC_CBK_EX_NR]
CTR_GB_NEUT_NOT_PLAUS_SUM	CTR_KM_BN	CTR_KM_CAN	CTR_NS_AD_CYC[NC_NOX_SENS_CONF]

CTR_NS_SHIFT_CYC[NC_NOX_SENS_CONF]	CTR_STC_TECU_1	CTR_STC_TECU_2	CTR_STC_TECU_3
CTR_STC_TECU_4	CTR_STC_TECU_5	CTR_STC_TECU_6	CTR_STC_TECU_7
CTR_STC_TECU_8	CTR_SWI_AFS_MON	CTR_WRST	
CUR_CNS_CWP	CUR_DMTL_COR_FIL		
CUR_DMTL_DMTLS_TEST	CUR_DMTL_REF_LEAK	CUR_DMTL_ROUGH_LEAK_MIN	CUR_MOT_VVL[NC_FAC_NR_CUR_MESS_VVL]
D_soc	DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]	Dfsiggen	DIST_ACT_MIL
DIST_LAM_AD_INJ_COLD	DIST_LAM_AD_INJ_HOT	DIST_NT_NS_AD[NC_NOX_SENS_CONF]	DIST_NT_NS_SHIFT[NC_NOX_SENS_CONF]
DIST_RST_DET[NC_NR_DBG_NVMY]	DIST_SO2P_END	EAC_COD_SENS	ECFPWM[2]
ECRASPWM	ECTPWM	ECU_STATE	EFF_CAT_DIAG_OBD[NC_CBK_EX_NR]
EFPPWM	EFPPWM_I_AD	EFPPWM_MIN_AD	EGY_STEP_INJ_CHA_GRD[NC_CYL_NR]
Eisyagr_korfak_b	ENVD_0_MON_3	ENVD_1_MON_3	ENVD_2_MON_3
ENVD_3_MON_3	ER_CYL[NC_CYL_NR]	F_atlad	FAC_CYL_LAM_COR[NC_CYL_NR]
FAC_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	FAC_EGY_PWM_AD[NC_CYL_NR]	FAC_FCO_KWP	FAC_H_RNG_LAM_AD[NC_CBK_EX_NR]
FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	FAC_LAM_AD_BAL[NC_CBK_EX_NR]	FAC_LAM_AD_CUS[NC_CBK_EX_NR]	FAC_LAM_AD_OUT[NC_CBK_EX_NR]
FAC_LAM_ADJ_COR_LAM_AD_CUS[NC_CBK_EX_NR]	FAC_LAM_CYL_SEL_ADJ_CST[NC_CYL_NR]	FAC_LAM_CYL_SEL_ADJ_H_RNG[NC_CYL_NR]	FAC_LAM_CYL_SEL_ADJ_L_RNG[NC_CYL_NR]
FAC_LAM_LIM[NC_CBK_EX_NR]	FAC_LAM_MV_MMV[NC_CBK_EX_NR]	FAC_LAM_TCO_A[NC_CBK_EX_NR]	FAC_LAM_TCO_B[NC_CBK_EX_NR]
FAC_LAM_TCO_C[NC_CBK_EX_NR]	FAC_LAM_TCO_D[NC_CBK_EX_NR]	FAC_LAM_TCO_E[NC_CBK_EX_NR]	FAC_LSL_GAIN_AD[NC_CBK_EX_NR]
FAC_MFF_ADD_FAC_LAM_AD[NC_CBK_EX_NR]	FAC_NOX_NS_AD[NC_NOX_SENS_CONF]	FAC_TI_AD_ER_BAL[NC_CYL_NR]	FAC_TI_ER_BAL[NC_CYL_NR]
FLOW_COR_CPS	FTL	FUP[NC_CBK_HPP_NR]	FUP_EFP
		FUP_MPL[NC_CBK_HPP_NR]	FUP_RNG_H_MES
GEAR	Gen_manufak	Gen_typkenn	I_gen
Ierr	Ierrfgrenz	IGA_IGC[NC_CYL_NR]	IVVTPWM_EX[NC_NR_CBK_IVVT]
IVVTPWM_EX[NC_NR_CBK_IVVT]	IVVTPWM_EX[NC_NR_CBK_IVVT]	IVVTPWM_IN[NC_NR_CBK_IVVT]	IVVTPWM_IN[NC_NR_CBK_IVVT]
IVVTPWM_IN[NC_NR_CBK_IVVT]	KNKS[NC_CYL_NR]	KNKS_REL_NL[NC_CYL_NR]	LAMB_BAS[NC_CBK_EX_NR]
LAMB_DELTA_AD_LAM_ADJ[NC_CBK_EX_NR]	LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]	LAMB_DIF_MON	LAMB_LS_UP[NC_CBK_EX_NR]
LAMB_LS_UP_MV	LAMB_NOX_SENS[NC_NOX_SENS_CONF]	LAMB_SP[NC_CBK_EX_NR]	LAMB_SP_HOM[NC_CBK_EX_NR]
LC_SWI_AEB_TYP	LF_ERR_PLAUS_IV_EGY_CAL	LF_ERR_PLAUS_IV_MFF_CAL	LOAD_BAT
LOAD_CLC	LOAD_MIN_MIS	LOAD_MIS	LSHPWM_DOWN[NC_CBK_EX_NR]
LSHPWM_UP[NC_CBK_EX_NR]	LV_ACCOUT_RLY	LV_CAN_TEMP_MIN_THD[NC_NOX_SENS_CONF]	LV_CDN_DIAG_TCHA_LEAK
LV_CRU_MAIN_SWI	LV_CS	LV_CS_CUS	LV_CYL_BAL_AD_HOM_REQ_DC
LV_CYL_BAL_AD_WG_OPEN_REQ[NC_CBK_EX_NR]	LV_CYL_BAL_ER_AD_ADD_EOL	LV_CYL_BAL_ER_AD_FAC_EOL	LV_CYL_BAL_LAM_AD_DC
LV_CYL_BAL_LAM_AD_EOL	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL	LV_CYL_BAL_LAM_SEL_AD_HOT_DC	LV_CYL_BAL_LAM_SEL_AD_HOT_EOL

LV_DBG_INFO_VLD[NC_NR_DBG_NVMY]			
LV_DMTL_PUMP	LV_DMTLS	LV_EBOX_CFA	LV_EF
LV_END_DIAG_TCHA_LEAK	LV_HDMTL_ON	LV_IM_BLS	LV_IM_BTS
LV_IPLSL_VLD[NC_CBK_EX_NR]	LV_IS	LV_KNK	LV_LS_DOWN_READY[NC_CBK_EX_NR]
LV_MIL_CAN	LV_PAS_RAMP_ACT_I_IS	LV_POIL_SWI	LV_RAS_OUT
LV_RLY_CRCV_HEAT			
LV_RLY_ST	LV_RLY_ST_CAN	LV_SAP	LV_SAV
LV_SEG_AD_AVL_ER	LV_SO2P_REQ_2	LV_SOF	LV_SOF_SWI
LV_SOF_SWI_REQ	LV_STATE_RLY_VVL	LV_VAR_MAF_LEARNT	LV_VAR_SAP
LV_WAL_1_CAN	MAF	MAF_KGH_MES[NC_MAF_NR]	MAF_KGH_MES_BAS[NC_MAF_NR]
MAF_MON	MAF_SUB_COR_MMV_MON	MAP	MAP_DIP_MES_BAS[NC_MAP_SENS_NR]
MAP_MES[NC_MAP_SENS_NR]	MAP_SAE	Md_gennm_na	MFF_AD_ADD_MMV_REL[NC_CBK_EX_NR]
MFF_AD_FAC_MMV_REL[NC_CBK_EX_NR]	MFF_ADD_AD_ER_BAL[NC_CYL_NR]		
MFF_ADD_COLD_LAM_AD_INJ[NC_CYL_NR]	MFF_ADD_HOT_LAM_AD_INJ[NC_CYL_NR]	MFF_ADD_LAM_AD[NC_CBK_EX_NR]	MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]
MFF_SP_MV	Msnlgofs_tmp	N	N_32
N_32_MON	N_32_SUB_MON	N_CPU_LOAD_MAX_RST_DET[NC_NR_DBG_NVMY]	N_DIF_SP_IS_MON
N_GRD	N_PERC_ECF	N_REL_CWP	N_REL_CWP_DIF
N_REL_CWP_SP	N_RST_DET[NC_NR_DBG_NVMY]	N_SP_IS	NC_CBK_EX_NR
NC_CBK_HPP_NR	NC_CYL_NR	NC_ECF_NR	NC_NR_CAM_CBK
NC_NR_CBK_IVVT	NC_NR_DBG_NVMY	NC_NR_STACK_ADR_RST	NC_NOX_SENS_CONF
NC_NR_TCO_SENS	NC_RST_DBG_BACKTRACE_SIZE	NC_RST_DBG_LIST_SIZE	NL[NC_CYL_NR]
NOX_NS[NC_NOX_SENS_CONF]	NOX_OFS_LOAD[NC_NOX_SENS_CONF]	NOX_OFS_PUC[NC_NOX_SENS_CONF]	NT_AGI
NT_AGI_SUL	NT_AGI_THERMO	NT_SUL	OBD_LAM_AD[NC_CBK_EX_NR]
OBD_TAM	OPG_ACR	OPG_SP_ACR	Oz_krzcnt
Oz_krzlor2k	Oz_krzmwkor	Oz_krzor	Oz_kvbog
Oz_kvbsm_ul	Oz_langmw	Oz_lf1c	Oz_lf1t
Oz_lf2c	Oz_lf2t	Oz_lgmwcnt	Oz_lp
Oz_lv	Oz_niv	Oz_nivakt	Oz_nivkrzt
Oz_nivlangt	Oz_nivr	Oz_oelkm	Oz_oelzeit
Oz_oricnt	Oz_peil	Oz_permakt	Oz_permbog
Oz_permex	Oz_permlow	Oz_permoff	Oz_permr
Oz_rwanz	Oz_rwkvb	Oz_rwperm	Oz_rwprog
Oz_tempakt	Oz_tempr	Oz_vfanz	Oz_vormw
Oz_vormwcnt	P_oel_ist	P_oel_soll	PBSU
			Pldr_soll
POIL_PWM	PSN_AD_CAM_EX[NC_NR_CAM_CBK]	PSN_AD_CAM_EX_2	PSN_AD_CAM_IN[NC_NR_CAM_CBK]
PSN_AD_CAM_IN_2	PSN_CAM_EX[NC_NR_CAM_CBK]	PSN_CAM_EX_2	PSN_CAM_IN[NC_NR_CAM_CBK]
PSN_CAM_IN_2	PUT[NC_MAP_SENS_NR]	PV_AV	PV_AV_1
PV_AV_2	PV_AV_MON	PV_AV_RAW	Pvdkds
PWM_ACR	PWM_DR_OUT_SET_VVL		RATIO_MMV_NS_SHIFT_DIAG[NC_NOX_SENS_CONF]

RATIO_NS_SHIFT_DIAG[NC_NOX_SENS_CONF]		R_IT_LS_UP[NC_CBK_EX_NR]	REL_CWP_PWR
RST_CLAS_SEC[NC_RST_DBG_LIST_SIZE]	RST_CLAS_TYP[NC_RST_DBG_LIST_SIZE]	RST_DBG_BACK_INFO_VLD[NC_RST_DBG_LIST_SIZE]	RST_INFO_ADD[NC_RST_DBG_LIST_SIZE]
RST_INFO_CTR	RST_SEC	RST_TYP[NC_RST_NR_NVMY]	Rt_bastatg_h
Rt_bastatg_hs	Rt_bastatg_s	Rt_bastatg_sa	SAF_DIAG_MIN
SAF_KGH_MES_BAS	SEG_AD_MMV_ER[NC_CYL_NR]	SEG_T_MES	St_ngang
Stat_sv_reg1	Stat_sv_reg2	STACK_ADR_RST[NC_NR_DBG_NVMY][NC_NR_STACK_ADR_RST]	
STATE_ACIN_CAN	STATE_ACR_AD		
STATE_CRU	STATE_CRU_OFF_IRR	STATE_CRU_OFF_REV	STATE_ECRAS_SYS
STATE_ENG	STATE_EOL_KWP_CPS	STATE_EOL_KWP_DMTL	STATE_EOL_KWP_N_SP_IS
STATE_EOL_KWP_SA	STATE_EOL_KWP_VLS	STATE_ERR_EL_LSL_UP[NC_CBK_EX_NR]	STATE_LRN_ECU
STATE_LS[NC_CBK_EX_NR]	STATE_LSH_DOWN[NC_CBK_EX_NR]	STATE_LSH_UP[NC_CBK_EX_NR]	STATE_MSW_CAN
STATE_N_LIM_ETC_REQ	STATE_N_MAX_MON	STATE_RST_TYP[NC_NR_DBG_NVMY]	STATE_RST_INFO_ADD[NC_NR_DBG_NVMY]
STATE_RST_TYP_ACT		STATE_SYM_DIAG_PUC_LSL_UP[NC_CBK_EX_NR]	STATE_SYM_OBD_LSL_LSH_UP[NC_CBK_EX_NR]
STATE_TQ_CAN_PLAUS	SUM_DIAG_DIAGCPS_SAE	SUM_RR	T_ACT_LEAK_MES
T_AST_SAE	T_CH_SO2P_ACT	T_ES	T_ES_CUS
T_PER_MAF_FRQ[NC_MAF_NR]	T2histshort	T3histshort	T4histshort
TAM	TAM_ST	Tans	Tchip
TCO	TCO_2	TCO_2_MES	TCO_MES
TCO_ST	TECU	TEG_CAT_DOWN_MDL[NC_CBK_EX_NR]	TEG_PCAT_DOWN_i
TEMP_EL_CWP	TEMP_MOT_VVL	TEMP_SWI_MES_MAX_VVL	Tget_b1
Tget_b2	Tget_b3	Tget_b4	Tget_b5
TI_1_MES[NC_CYL_NR]	TIA[NC_IDX_TIA_IM_CYL]	TIA_MES	TIA_ST
TIA_TCHA	Tlrfgen	Tmot_b1	Tmot_b2
Tmot_b3	Tmot_b4	Tmot_b5	Toel
Toel_b1	Toel_b2	Toel_b3	Toel_b4
Toel_b5			
TOIL_MES	TPS_AD_STEP	TPS_AV[NC_ETC_NR]	TPS_AV_1[NC_ETC_NR]
TPS_AV_2[NC_ETC_NR]	TPS_SP[NC_ETC_NR]	TPS_SP_MDL[NC_ETC_NR]	TQ_DIF_I_IS_MON
TQ_DIF_P_D_FAST_IS	TQ_DIF_P_D_SLOW_IS	TQ_LOSS_DIF_MON	TQ_LOSS_MON
TQ_MAX_CLU_DIF_MON	TQ_MIN_CLU_DIF_MON	TQ_REQ_CLU	TQI_AV
TQI_AV_H_RNG_MON	TQI_AV_MON	TQI_GS_FAST_DEC	TQI_MSR_CAN
TQI_SP	TQI_SP_H_RNG_MON	TQI_SP_MON	TRT
TRT_RST_DET[NC_NR_DBG_NVMY]	TTIP_MES_LS_UP[NC_CBK_EX_NR]	Tumg_b1	Tumg_b2
Tumg_b3	Tumg_b4	Tumg_b5	Tvngang
U_batt	U_fgen	Ubt	Uregnom
V_ACR	V_ACR_AD_BOL	V_ACR_AD_BOL_0	V_ACR_AD_TOL
V_ALTER_SP	V_AMP	V_CWP	V_DMTL
V_FUP_EFP	V_FUP_EFP_MV	V_FUP_MV[NC_CBK_HPP_NR]	V_IGK_BAS
V_IGK_MES	V_IV_1_MES[NC_CYL_NR]	V_MAP[NC_MAP_SENS_NR]	V_PBSU
			V_PUT

V_PVS_1	V_PVS_1_MON	V_PVS_2	V_PVS_2_MON
V_SOF_SWI	V_VCC_SENS_VVL	V_VCC_SENS_VVL_RAW	VB
VB_BAS	VCC_DR_VVL	VCC_PVS_1	VCC_PVS_2
VFF_EFP	VFF_MFF_SP_FUP_CTL	VIM_AV	VIMPWM_1
VIMPWM_2	VLS_COR_LSL[NC_CBK_EX_NR]	VLS_DIF_LAM_ADJ[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]
VLS_NOX_SENS[NC_NOX_SENS_CONF]	VLS_OFS_LSL[NC_CBK_EX_NR]	VLS_UP[NC_CBK_EX_NR]	VP_TCO
VP_TECU	VP_TEG_PCAT_DOWN	VP_TIA[NC_SENS_NR_TIA][1][1]	VP_TIA_TCHA
VP_TPS_1[NC_ETC_NR]	VP_TPS_2[NC_ETC_NR]	VS	WGPWM[NC_NR_TCHA]

General information:

This specification is based on the on actual program version of Basic-Software.
All project specific implemented services and parameter of diagnostic communication protocol are listed below.

In the attached chapters, the displayed telegram structure is without the header for e.g. KWP2000, KWP2000* or BMW-FAST Protocol.
The structure from the service ID is the same in all cases.

Below the differences between KWP2000, KWP2000* and BMW-FAST Protocol are described.

KWP2000:

Baudrate: 10 400
Header:

Data Byte	*Byte #*	*Parameter Name*	*CVT*	*Value*	*Mnemonic*
	#1d	**Format byte**	**M**	**#AALLLLLLb**	**FMT**
	#2d	**Target address byte**	**C**	**#12h/#EAh/#EDh/#EFh /#33h**	**TGT**
	#3d	**Source address byte**	**C**	**#F1h**	**SRC**
	#4d	**Additional length byte**	**C**	**#01h-#FFh**	**LEN**
#1d	#5d	Service ID Byte	M	#XXh	SID
•	#6d	Data Byte #02h	C	#XXh	DATA
•	•	•	•	•	
•	•	•	•	•	
•	•	•	•	•	
#255d	#259d	Data Byte #255d	C	#XXh	
	#260d	addition Checksum	M	#00h-#FFh	CS

KWP2000*:

Baudrate: 9 600
Header:

Data Byte	*Byte #*	*Parameter Name*	*CVT*	*Value*	*Mnemonic*
	#1d	**Format byte**	**M**	**#B8h**	**FMT**
	#2d	**Target address byte**	**C**	**#12h**	**TGT**

	#3d	**Source address byte**	**C**	**#F1h**	**SRC**
	#4d	**length byte calculated by the tool**	**C**	**#01h-#FFh**	**LEN**
#1d	#5d	Service ID Byte	M	#XXh	SID
•	#6d	Data Byte #02h	C	#XXh	DATA
•	•	•	•	•	
•	•	•	•	•	
•	•	•	•	•	
#255d	#259d	Data Byte #255d	C	#XXh	
	#260d	exor Checksum	M	#00h-#FFh	CS

BMW-FAST:
Baudrate: 115 200
Header:

Data Byte	*Byte #*	*Parameter Name*	*CVT*	*Value*	*Mnemonic*
	#1d	**Format byte**	**M**	**#AALLLLLLb**	**FMT**
	#2d	**Target address byte**	**C**	**#12h/#EAh/#EDh/#EFh /#33h**	**TGT**
	#3d	**Source address byte**	**C**	**#F1h**	**SRC**
	#4d	**Additional length byte**	**C**	**#01h-#FFh**	**LEN**
#1d	#5d	Service ID Byte	M	#XXh	SID
•	#6d	Data Byte #02h	C	#XXh	DATA
•	•	•	•	•	
•	•	•	•	•	
•	•	•	•	•	
#255d	#259d	Data Byte #255d	C	#XXh	
	#260d	addition Checksum	M	#00h-#FFh	CS

BMW-FAST is identical to KWP2000 except baudrate.

7.2.1 Normative reference

This implementation and the description of diagnostic communication protocol KWP2000 is based on the following standards:

ISO /FDIS 14229: 1996	Road Vehicles-Diagnostic systems - DiagnosticServices Specification
ISO /FDIS 14230-1:1996	Road Vehicles - Diagnostic systems -Keyword Protocol 2000 - Part 1: Physical Layer
ISO/FDIS 14230-2:1996	Road Vehicles - Diagnostic systems-Keyword Protocol 2000 - Part 2: Data Link Layer
ISO /FDIS 14230-3:1996	Road Vehicles - Diagnostic systems -Keyword Protocol 2000 - Part 3: Implementation
ISO/DIS 14230-4	Road Vehicles - Diagnostic Systems Keyword Protocol 2000 - Part 4 KWP 2000 Requirements for emission-related systems

VDA / WD 14230-2: 1997 Keyword Protocol 2000 - Part 2 Data Link Layer
VDA - Recommended Practice

VDA / WD 14230-3: 1997 Keyword Protocol 2000 - Part 3 Implementation
VDA - Recommended Practice

Diagnosekommunikation mit KWP 2000(*) in Motorsteuerungen Revision 1.2
Ident.Nummer **1 430 227 Index d NÄL.-Nr.: E2263.N**

7.2.2 Convention

☐ M:	Mandatory
☐ C:	Conditional; the presence of the parameter depend on other parameters within the service
☐ CVT:	Conventions
☐ #XXh:	hexadecimal number
☐ #XXd:	decimal number
☐ #XXb:	binary number

These values are necessary for the services - read failure memory extra long (21_0A and 17h) and DDLI and for the freeze frame.

When system values (Word) are used for DDLI output as a Byte, then the lowbyte is cut off, so only the high byte is the output. In this case a selection for the environmental value (FMY) is necessary, only the high Byte from the Word is the output.

When system values (Byte) are used for DDLI output as a Word, the Byte is shifted to the high Byte and the low Byte is filled with 00h.

Analog values e.g. 0…3FFh = 0…5V, are shifted 4 Bits to FFC0h.

When these analog values (FFC0h) are used for DDLI output as a Byte, then the lowbyte is cut off, so only the high byte is the output. In this case a selection for the environmental value (FMY) is necessary, only the high Byte from the Word is the output.

List for selection of the environmental datas for each failure also for DDLI service 2C

cross out ist not supportet yet!

7.2.3 Actuator Diagnosis - Timeouts

7.2.3.1 Diagnostic Timeout

The standard timeout of KWP2000 is defined by P3 max.

In KWP2000(*) is no timeout requested.

	P3 max
Default	5000ms infinity for KWP*

Extended	250ms
Programming	16s
Periodic transmission	250ms

After a diagnostic timeout occurs all actual actuator services are stopped.

7.2.3.2 TIMEOUT_TIME_XX_KWP

This timeout is sent by the protocoll from the tester to the ECU. It can be different for each actuator service.

After the timeout TIMEOUT_TIME_XX_KWP occurs the corresponding actuator services is stopped.

If TIMEOUT_TIME_XX_KWP is greater than the diagnostic timeout, the tester has to take special care to sent “tester present” service.

7.2.4 Definitions for EOL Tests

The status STATE_EOL_KWP_xxx should be defined with the single Bits from the table below. Only supported (necessary) states can be described as followed:

Definitions for EOL Tests: see BMW LH 7 507 660	**values of the status if the function supported the singel one**	
SIEMENSVDO_ SW_NAME	**remark**	**hex value**
EOL_TEST_ACT	test currently running -function active	00h **(other)** 05H **(MSD8x)**
EOL_TEST_ST_INH	start of test not allowed – start-/control - conditions not correct	01h
EOL_TEST_PAR_NOT_PLAUS	hand over parameter not plausible- parameter not correct	02h
EOL_TEST_WAIT_REL	function waiting for release	03h
not used	not used	04h
EOL_TEST_NOT ST	test not yet started (default state) – function not started yet	00h (**MSD8x)** 05h **(other)**
EOL_TEST_END_WOUT_RESULT	function finished - without failures otherwise see 08h or 09h	06h
EOL_TEST_ABORTED	test aborted – function stopped	07h
EOL_TEST_END_WOUT_ERR	test finished without errors - function complete, no errors	08h
EOL_TEST_ END_WITH_ERR	test finished with errors – function complete, with errors	09h

Example (this is not a output section!):

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_EOL_KWP_XXX	O/V	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H	**MSD8x**: EOL_TEST_NOT_ST Other : EOL_TEST_ACT EOL_TEST_ST_INH EOL_TEST_PAR_NOT_PLAUS EOL_TEST_WAIT_REL not used **MSD8x**: EOL_TEST_ACT Other : EOL_TEST_NOT_ST EOL_TEST_END_WOUT_RESULT EOL_TEST_ABORTED EOL_TEST_END_WOUT_ERR EOL_TEST_END_WITH_ERR	1	-
State variable for end of line test XXX					

7.2.5 Variable conversion for KWP2000 / KWP2000(*)

Description of Basic Equations for conversion:

The general equation for conversion from Siemens internal Range to Customer Range is

General Equation:

$$Var_{cus\text{-}hex} = m(Var_{int\text{-}hex} - C) \qquad \text{--------------------------}(1)$$

Calculation of m and C:

$$m = \frac{\text{resolution of Customer}}{\text{resolution of Siemens}} \qquad \text{---------------------------}(2)$$

$$C = 0_{int\text{-}hex} - 0_{cus\text{-}hex}/m \qquad \text{----------------------------}(3)$$

$Var_{int\text{-}hex}$ = the hex value of the variable in Siemens internal Range
$Var_{cus\text{-}hex}$ = the hex value of the variable in customer Range

m = the slope of conversion from $Var_{int\text{-}hex}$ to $Var_{cus\text{-}hex.}$
C = the offset of conversion from $Var_{int\text{-}hex}$ to $Var_{cus\text{-}hex.}$

$0_{int\text{-}hex}$ = the hex. value for a phys. value 0 in Siemens Range.
$0_{cus\text{-}hex}$ = the hex. value for a phys. value 0 in Customer Range.

Derivation of Equations

To derive the above equations, consider two instances(a1 and a2) of the equation(1),

$a1_{cus\text{-}hex} = m(a1_{int\text{-}hex} - C)$ ----------------(4)
$a2_{cus\text{-}hex} = m(a2_{int\text{-}hex} - C)$ ----------------(5)

Subtracting (4) from (5) and rearranging, we get

$$m = \frac{(a1_{cus\text{-}hex} - a2_{cus\text{-}hex})}{(a1_{int\text{-}hex} - a2_{int\text{-}hex})}$$

$$= \frac{(a1_{cus\text{-}hex} - a2_{cus\text{-}hex}) / (a1_{phy} - a2_{phy})}{(a1_{int\text{-}hex} - a2_{int\text{-}hex}) / (a1_{phy} - a2_{phy})}$$

$$= \frac{\text{resolution of Customer}}{\text{resolution of Siemens}}, \text{ Hence (2) is proved.}$$

From (4), we get

$C \quad = \quad a1_{int\text{-}hex} - a1_{cus\text{-}hex} / m$

The equation (3) is obtained when a1=0. Hence (3) is proved.

where

$a1_{int\text{-}hex}$ = the hex. value of the instance *'a1'* in Siemens Range.
$a1_{cus\text{-}hex}$ = the hex. value of the instance *'a1'* in Customer Range.
$a2_{int\text{-}hex}$ = the hex. value of the instance *'a2'* in Siemens Range.
$a2_{cus\text{-}hex}$ = the hex. value of the instance *'a2'* in Customer Range.
$a1_{phy}$ = the phys. value of the instance *'a1'*.
$a2_{phy}$ = the phys. value of the instance *'a2'*.

7.3 DDLI- List for environmental data and ERRM

!!! Attention !!!
!!! The output variables above are only calculated at tester request. They are present in the a2l- file and visible in INCA, but no reasonable values will be displayed. !!!

7.3.1 DDLI basic List

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5800	00	Zeit nach Start	T_AST_SAE		X	X	X	X	X	X
5801	01	Umgebungsdruck	OBD_AMP	AMP_MES converted for KWP	X	X	X	X	X	X
5802	02	Zustand Lambdaregelung Bank 1	STATE_LS[1]		X	X	X	X	X	X
5803	03	Zustand Lambdaregelung Bank 2	STATE_LS[2]		X	X	X	X	X	X
5804	04	Berechneter Lastwert	LOAD_CLC		X	X	X	X	X	X
5805	05	Kühlmitteltemperatur	OBD_TCO	TCO_MES converted for KWP	X	X	X	X	X	X
5806	06	Lambda Integrator Gruppe 1	OBD_LAM_COR[1]	FAC_LAM_LIM[1] converted for KWP	X	X	X	X	X	X
5807	07	Lambda Adaption multiplikativ Gruppe 1	OBD_LAM_AD[1]		X	X	X	X	X	-
		Lambda Adaption multiplikativ Gruppe 1	OBD_LAM_AD[1]	FAC_MFF_ADD_FAC_LAM_AD[1] converted for KWP	-	-	-	-	-	X
5808	08	Lambda Integrator Gruppe 2	OBD_LAM_COR[2]	FAC_LAM_LIM [2] converted for KWP	X	X	X	X	X	X
5809	09	Lambda Adaption multiplikativ Gruppe 2	OBD_LAM_AD[2]		X	X	X	X	X	-
		Lambda Adaption multiplikativ Gruppe 2	OBD_LAM_AD[2]	FAC_MFF_ADD_FAC_LAM_AD[2] converted for KWP	-	-	-	-	-	X
580A	0A	Mass fuel flow setpoint after combustion selection	MFF_SP_MV_KWP	MFF_SP_MV converted for KWP	-	X	X	X	X	-
		Kraftstoffdruck	OBD_FUP	FUP converted for KWP	X	-	-	-	-	-
580B	0B	Saugrohrdruck	MAP_SAE		X	X	X	X	X	X
580C	0C	Drehzahl	OBD_N	N converted for KWP	X	X	X	X	X	X
580D	0D	Geschwindigkeit	VS		X	X	X	X	X	X
580E	0E	Zündzeitpunkt Zylinder 1	OBD_IGA_IGC	IGA_IGC[0] converted for KWP	X	X	X	X	X	X
580F	0F	Ansauglufttemperatur	OBD_TIA	TIA_MES converted for KWP	X	X	X	X	X	X
5810	10	Luftdurchsatz OBD	OBD_MAF	MAF_KGH_MES converted for KWP	X	X	X	X	X	-
		Luftdurchsatz OBD	OBD_MAF	MAF_KGH_MES[0] converted for KWP	-	-	-	-	-	X
5811	11	Motordrehzahl	N_32		X	X	X	X	X	X
5812	12	Luftmasse gemessen	MAF_KGH_MES_BAS		X	X	X	X	X	-
5813	13	Relative Last	Rf		X	X	X	X	X	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5814	14	Fahrpedalwert	PV_AV_RAW		X	X	X	X	X	X
5815	15	Batteriespannung	OBD_VB	VB converted for KWP	X	X	X	X	X	X
5816	16	Lambda Setpoint	OBD_LAMB_SP	LAMB_SP_HOM[1] converted for KWP	X	X	X	X	X	X
5817	17	Umgebungstemperatur	OBD_TAM		X	X	X	X	X	X
5818	18	Luftmasse gerechnet	MAF		X	X	X	X	X	-
5819	19	Drehzahl OBD Byte	N_SAE_BYTE_KWP	N converted for KWP	X	X	X	X	X	-
581A	1A	Nockenwelle Einlass	CAM_IN_KWP	CAM_IN[1] converted for KWP	X	X	X	X	X	-
581B	1B	Nockenwelle Einlass Sollwert	CAM_SP_IVVT_IN_KWP	CAM_SP_IVVT_IN converted for KWP	X	X	X	X	X	-
581C	1C	Nockenwelle Auslass	CAM_EX[1]		X	X	X	X	X	-
581D	1D	Nockenwelle Auslass Sollwert	CAM_SP_IVVT_EX		X	X	X	X	X	-
581E	1E	Ansauglufttemperatur	TIA_MES		X	X	X	X	X	-
581F	1F	Motortemperatur	TCO_MES		X	X	X	X	X	X
5820	20	Kühlmitteltemperatur Kühlerausgang	TCO_2_MES		X	X	X	X	X	X
5821	21	Steuergerät Innentemperatur	TECU		X	X	X	X	X	X
5822	22	(Motor) - Öltemperatur	TOIL_MES		X	X	X	X	X	X
5823	23	Zeit Motor steht	T_ES		X	X	X	X	X	X
5824	24	Umgebungstemperatur	TAM		X	X	X	X	X	X
5825	25	Zeit Motor steht	T_ES_CUS_KWP	T_ES_CUS converted for KWP	X	X	X	X	X	X
5826	26	Drosselklappe Sensor 1	TPS_AV_1		X	X	X	X	X	-
5827	27	Lambdasonden Heizung Vorkat 1	LSHPWM_UP[1]		X	X	X	X	X	X
5828	28	Lambdasonden Heizung Vorkat 2	LSHPWM_UP[2]		X	X	X	X	X	X
5829	29	Lambdasonden Heizung Hinterkat 1	LSHPWM_DOWN[1]		X	X	X	X	X	X
582A	2A	Lambdasonden Heizung Hinterkat 2	LSHPWM_DOWN[2]		X	X	X	X	X	X
582B	2B	Drehmomenteingriff über CAN	STATE_TQ_CAN_PLAUS		X	X	X	X	X	X
582C	2C	number of invalid INIT_REG_x write validation cycles	CTR_ERR_LSL_IF_SPI_WR[1]		X	X	X	X	X	X
582D	2D	number of invalid INIT_REG_x write validation cycles	CTR_ERR_LSL_IF_SPI_WR[2]		X	X	X	X	X	X
582E	2E	Adaptionsfaktor Sensor Zeitkonstante vor Kat Bank 1	FAC_DIAG_DYN_LSL_UP[1]		X	X	X	X	X	X
582F	2F	Adaptionsfaktor Sensor Zeitkonstante vor Kat Bank 2	FAC_DIAG_DYN_LSL_UP[2]		X	X	X	X	X	X
5830	30	Mean value of normalised single sensor signal	FAC_MV_DIAG_DYN_LSL_UP_KWP[1]	FAC_MV_DIAG_DYN_LSL_UP[1] converted for KWP	X	X	X	X	X	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
		amplitude								
5831	31	Mean value of normalised single sensor signal amplitude	FAC_MV_DIAG_DYN_LSL_UP_KWP[2]	FAC_MV_DIAG_DYN_LSL_UP[2] converted for KWP	X	X	X	X	X	X
5832	32	Motor Status	STATE_ENG		X	X	X	X	X	X
5833	33	Umgebungstemperatur beim Start	TAM_ST		X	X	X	X	X	X
5834	34	Umgebungsdruck	AMP_MES		X	X	X	X	X	X
5835	35	manufacturer identifier alternator	Gen_manufak		X	X	X	X	X	X
5836	36	Drehzahlgradient	N_GRD		X	X	X	X	X	X
5837	37	Status OBD-I Fehler vor Kat Bank 1	STATE_ERR_EL_LSL_UP[1]		X	X	X	X	X	X
5838	38	Status OBD-I Fehler vor Kat Bank 2	STATE_ERR_EL_LSL_UP[2]		X	X	X	X	X	X
5839	39	Status Drosselklappe Notlauf	STATE_ETC_LIH		X	X	X	X	X	X
583A	3A	Ansauglufttemperatur beim Start	TIA_ST		X	X	X	X	X	X
583B	3B	Kraftstofftank Füllstand	FTL		X	X	X	X	X	X
583C	3C	Spannung Kl. 87	VB		X	X	X	X	X	X
583D	3D	Reset type	RST_CLAS_TYP[0]		X	X	X	X	X	-
		Reset typ / RESET SAFE	STATE_RST_TYPE[0]		-	-	-	-	-	X
583E	3E	Motordrehzahl bei Reset	N_RST_DET_KWP	N_RST_DET[0] converted to KWP	X	X	X	X	X	X
583F	3F	Drosselklappe Sollwert	TPS_SP		X	X	X	X	X	-
		Drosselklappe Sollwert	TPS_SP[0]		-	-	-	-	-	X
5840	40	CPU Last bei Reset	CPU_LOAD_RST_DET_KWP		X	X	X	X	X	X
5841	41	SG-Innentemperatur Rohwert	VP_TECU_KWP	VP_TECU converted for KWP	X	X	X	X	X	X
5842	42	type identifier alternator	Gen_typkenn		X	X	X	X	X	X
5843	43	Versorgung FWG 1	VCC_PVS_1_KWP	VCC_PVS_1 converted for KWP	X	X	X	X	X	X
5844	44	chip temperature alternator	TCHIP_KWP	Tchip converted for KWP	X	X	X	X	X	X
5845	45	Spannung Lambdasonde VorKat 1	VLS_UP_KWP[1]	VLS_UP[1] converted for KWP	X	X	X	X	X	X
5846	46	Spannung Pedalwertgeber 1	V_PVS_1_KWP	V_PVS_1 converted for KWP	X	X	X	X	X	X
5847	47	Spannung Pedalwertgeber 2	V_PVS_2_KWP	V_PVS_2 converted for KWP	X	X	X	X	X	X
5848	48	Spannung Lambdasonde VorKat 2	VLS_UP_KWP[2]	VLS_UP[2] converted for KWP	X	X	X	X	X	X
5849	49	Spannung Lambdasonde HinterKat 1	VLS_DOWN_KWP[1]	VLS_DOWN[1] converted for KWP	X	X	X	X	X	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
584A	4A	Spannung Kl. 15 Rohwert	V_IGK_BAS_KWP	V_IGK_BAS converted for KWP	X	X	X	X	X	X
584B	4B	Spannung Lambdasonde HinterKat 2	VLS_DOWN_KWP[2]	VLS_DOWN[2] converted for KWP	X	X	X	X	X	X
584C	4C	Spannung Drosselklappe Poti 2	V_TPS_2_KWP	V_TPS_2 converted for KWP	X	X	X	X	X	-
584D	4D	korrigierter Sollwert Durchfluss Tankentlüftung	FLOW_COR_CPS		X	X	X	X	X	X
584E	4E	Spannung Drosselklappe Poti 1	V_TPS_1_KWP	V_TPS_1 converted for KWP	X	X	X	X	X	-
584F	4F	Spannung Luftmasse	V_MAF		X	X	X	X	X	_
5850	50	Spannung Motortemperatur	VP_TCO_KWP	VP_TCO[0] converted for KWP	X	X	X	X	X	X
5851	51	Spannung Ansauglufttemperatur	VP_TIA_KWP	VP_TIA convewrted for KWP	X	X	X	X	X	-
5852	52	Kühlmitteltemperatur Kühlerausgang Rohwert	V_TCO_2_KWP	VP_TCO[1] converted for KWP	X	X	X	X	X	X
5853	53	Spannung Kl.87 Rohwert	VB_BAS_KWP	VB_BAS converted for KWP	X	X	X	X	X	X
5854	54	Versorgung FWG 2	VCC_PVS_2_KWP	VCC_PVS_2 converted for KWP	X	X	X	X	X	X
5855	55	Mittelwert Bank 1	FAC_LAM_MV_MMV[1]		X	X	X	X	X	X
5856	56	Mittelwert Bank 2	FAC_LAM_MV_MMV[2]		X	X	X	X	X	X
5857	57	exciting current alternator	Ierr		X	X	X	X	X	X
5858	58	Drosselklappe aktueller Wert	TPS_AV		X	X	X	X	X	-
5859	59	DMTL Strom Referenzleck	CUR_DMTL_REF_LEAK_KWP	CUR_DMTL_REF_LEAK converted for KWP	X	X	X	X	X	X
585A	5A	DMTL Strom Grobleck	CUR_DMTL_ROUGH_LEAK_MIN_KWP	CUR_DMTL_ROUGH_LEAK_MIN converted for KWP	X	X	X	X	X	X
585B	5B	DMTL Strom Diagnoseende	CUR_DMTL_COR_FIL_KWP	CUR_DMTL_COR_FIL converted for KWP	X	X	X	X	X	X
585C	5C	Widerstand Lambdasonde NK 1	R_IT_LS_DOWN_KWP_H[1]	HIGH BYTE (R_IT_LS_DOWN[1])	X	X	X	X	X	X
585D	5D	Widerstand Lambdasonde NK 2	R_IT_LS_DOWN_KWP_H[2]	HIGH BYTE (R_IT_LS_DOWN[2])	X	X	X	X	X	X
585E	5E	unteres Byte Widerstand Lambdasonde NK 1	R_IT_LS_DOWN_KWP_L[1]	LOW BYTE (R_IT_LS_DOWN[1])	X	X	X	X	X	X
585F	5F	unteres Byte Widerstand Lambdasonde NK 2	R_IT_LS_DOWN_KWP_L[2]	LOW BYTE (R_IT_LS_DOWN[2])	X	X	X	X	X	X
5860	60	Widerstand Lambdasonde VK 1	R_IT_LS_UP_KWP_H[1]	HIGH BYTE (R_IT_LS_UP[1])	X	X	X	X	X	X
5861	61	Widerstand Lambdasonde VK 2	R_IT_LS_UP_KWP_H[2]	HIGH BYTE (R_IT_LS_UP[2])	X	X	X	X	X	X
5862	62	Öldruck Sollwert	P_OEL_SOLL_KWP	P_oel_soll converted to KWP	X	X	X	X	X	-
5863	63	untere Byte Widerstand Lambdasonde VK 1	R_IT_LS_UP_KWP_L[1]	LOW BYTE (R_IT_LS_UP[1])	X	X	X	X	X	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5864	64	untere Byte Widerstand Lambdasonde VK 2	R_IT_LS_UP_KWP_L[2]	LOW BYTE (R_IT_LS_UP[2])	X	X	X	X	X	X
5865	65	Oelstand Mittelwert Langzeit	Oz_nivlangt		X	X	X	X	X	X
5866	66	Füllstand Motoröl	Oz_lp		X	X	X	X	X	X
5867	67	Kilometerstand	CTR_KM_CAN_KWP	CTR_KM_CAN (high byte)	X	X	X	X	X	X
5868	68	Status Standverbraucher registriert Teil 1	Stat_sv_reg1		X	X	X	X	X	X
5869	69	Status Standverbraucher registriert Teil 2	Stat_sv_reg2		X	X	X	X	X	X
586A	6A	Batteriespannung von IBS gemessen	U_batt		X	X	X	X	X	X
586B	6B	Zeit mit Ruhestrom 80 - 200 mA	T2histshort		X	X	X	X	X	X
586C	6C	Zeit mit Ruhestrom 200 - 1000 mA	T3histshort		X	X	X	X	X	X
586D	6D	Zähler Erkennung schlechte Strasse	SUM_RR		X	X	X	X	X	X
586E	6E	Zeit mit Ruhestrom >1000 mA	T4histshort		X	X	X	X	X	X
586F	6F	Ist-Öldruck	P_OEL_IST_KWP	P_oel_ist converted to KWP	X	X	X	X	X	X
5870	70	Spannung DME Umgebungsdruck	V_AMP_KWP	V_AMP converted for KWP	X	X	X	X	X	X
5871	71	Lambda-Sollwert Gruppe 1	LAMB_SP_KWP[1]	LAMB_SP[1] converted for KWP	X	X	X	X	X	X
5872	72	controller version alternator	Bsdgenregv		X	X	X	X	X	X
5873	73	Lambda-Sollwert Gruppe 2	LAMB_SP_KWP[2]	LAMB_SP[2] converted for KWP	X	X	X	X	X	X
5874	74	Spannung Strommessung DMTL	V_DMTL_KWP	V_DMTL converted for KWP	X	X	X	X	X	X
5875	75	Sollwert Motormoment	TQI_SP_KWP	TQI_SP converted to TQI_SP_KWP	X	X	X	X	X	X
5876	76	Mittlere Diagnosewert minimale Luftmasse	SAF_DIAG_MIN		X	-	-	-	-	-
		OBD high range fuel pressure (high byte)	OBD_FUP_RNG_H_H (high byte of OBD_FUP_RNG_H))	FUP_RNG_H_MES converted to OBD_FUP_RNG_H	-	X	X	X	X	-
		OBD high range fuel pressure (high byte)	OBD_FUP_RNG_H_H (high byte of OBD_FUP_RNG_H))	FUP_MV	-	-	-	-	-	X
5877	77	Differenz zwischen Maximum und Minimum SAF	SAF_KGH_DIF		X	-	-	-	-	-
		OBD high range fuel pressure (low byte)	OBD_FUP_RNG_H_L (low byte of OBD_FUP_RRNG_H)	FUP_RNG_H_MES converted to OBD_FUP_RNG_H	-	X	X	X	X	-
		OBD high range fuel pressure (low byte)	OBD_FUP_RNG_H_L (low byte of OBD_FUP_RRNG_H)	FUP_MV	-	-	-	-	-	X
5878	78	Lambdaverschiebung	LAMB_DELTA_I_LAM	LAMB_DELTA_I_LAM_ADJ_[1]	X	X	X	X	X	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
		Rückführregler 1	_ADJ_KWP[1]	converted for KWP						
5879	79	Lambdaverschiebung Rückführregler 2	LAMB_DELTA_I_LAM_ADJ_KWP[2]	LAMB_DELTA_I_LAM_ADJ_[2] converted for KWP	X	X	X	X	X	X
587A	7A	Status FGR	STATE_CRU		X	X	X	X	X	X
587B	7B	Maximal squared effectiv current of the chopper	CUR_CUR_EFC_MAX_DR_VVL_KWP	CUR_CUR_EFC_MAX_DR_VVL converted to KWP	X	-	-	-	-	-
		Abgleich AGR-Modell (Faktor)	Eisyagr_korfak_b		-	X	X	X	X	-
587C	7C	Status Motorsteuerung	ECU_STATE		X	X	X	X	X	X
587D	7D	Symptom bei Schubabschaltung Sonde vor Kat Bank1	STATE_SYM_DIAG_PUC_LSL_UP[1]		X	X	X	X	X	X
587E	7E	Symptom bei Schubabschaltung Sonde vor Kat Bank2	STATE_SYM_DIAG_PUC_LSL_UP[2]		X	X	X	X	X	X
587F	7F	Tastverhaeltnis E-Lüfter	ECFPWM[0]		X	X	X	X	X	X
5880	80	Tastverhältnis: Luftklappe	ECRASPWM		X	X	X	X	X	-
5881	81	berechneter Gang	GEAR		X	X	X	X	X	X
5882	82	Motortemperatur beim Start	TCO_ST		X	X	X	X	X	X
5883	83	Spannung Klopfwerte Zylinder 1	NL[0]		X	X	X	X	X	X
5884	84	Actual value exciting current limitation alternator	Ierrfgrenz		X	X	X	X	X	X
5885	85	Spannung Klopfwerte Zylinder 3	NL[2]		X	X	X	X	X	X
5886	86	Spannung Klopfwerte Zylinder 6	NL[3]		X	X	X	X	X	X
5887	87	occupancy of alternator	Dfsiggen		X	X	X	X	X	X
5888	88	Spannung Klopfwerte Zylinder 4	NL[5]		X	X	X	X	X	X
5889	89	Lambda-Istwert Gruppe 1	LAMB_KWP[1]	LAMB_LS_UP[1] converted to KWP	X	X	X	X	X	X
588A	8A	Lambda-Istwert Gruppe 2	LAMB_KWP[2]	LAMB_LS_UP[2] converted to KWP	X	X	X	X	X	X
588B	8B	Zeit seit Startende	T_AST		X	X	X	X	X	X
588C	8C	Keramiktemperatur Lambdasonde VK 1	TTIP_MES_LS_UP[1]		X	X	X	X	X	X
588D	8D	aktuelle Zeit DMTL Leckmessung	T_ACT_LEAK_MES		X	X	X	X	X	X
588E	8E	Pumpenstrom bei DMTL Pumpenprüfung	CUR_DMTL_DMTLS_TEST		X	X	X	X	X	X
588F	8F	Keramiktemperatur Lambdasonde VK 2	TTIP_MES_LS_UP[2]		X	X	X	X	X	X
5890	90	Spannung Bremsunterdrucksensor	V_PBSU_KWP	V_PBSU converted for KWP	-	X	X	X	X	_

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5891	91	Momentanforderung an der Kupplung	TQ_REQ_CLU		X	X	X	X	X	X
5892	92	Bremsunterdruck	PBSU_KWP	For 4cyl:PBSU converted for KWP For 6cyl: stub with 0 hPa	-	X	X	X	X	_
5893	93	Drehmomentabfall schnell bei Gangwechsel	TQI_GS_FAST_DEC		X	X	X	X	X	X
5894	94	Symptom Lambdasondenheizung vor Kat Bank 1	STATE_SYM_OBD_LSL_LSH_UP[1]		X	X	X	X	X	X
5895	95	Symptom Lambdasondenheizung vor Kat Bank 2	STATE_SYM_OBD_LSL_LSH_UP[2]		X	X	X	X	X	X
5896	96	Abgastemperatur nach Kat Bank 1	TEG_CAT_DOWN_MDL[1]		X	X	X	X	X	X
5897	97	Abgastemperatur nach Kat Bank 2	TEG_CAT_DOWN_MDL[2]		X	X	X	X	X	X
5898	98	Generator Sollspannung	V_ALTER_SP_KWP	V_ALTER_SP converted for KWP	X	X	X	X	X	X
5899	99	DISA Position	VIM_AV		X	X	X	X	X	-
589A	9A	Tastverhältnis Nullgangsensor	PWM_NEUT_PSN_GB_KWP	PWM_NEUT_PSN_GB converted for KWP	-	X	X	X	X	-
589B	9B	Spannungsoffset Signalpfad CJ120 1	VLS_OFS_LSL_KWP[1]	VLS_OFS_LSL[1] converted for KWP	X	X	X	X	X	X
589C	9C	Spannungsoffset Signalpfad CJ120 2	VLS_OFS_LSL_KWP[2]	VLS_OFS_LSL[2] converted for KWP	X	X	X	X	X	X
589D	9D	Statusbyte VVT-Endstufe ATIC 61	STATE_DIAG_DR_VVL		X	-	-	-	-	-
		Abweichung Lambdasonde zu Modellwert: Überwachung	LAMB_DIF_MON_KWP	LAMB_DIF_MON converted to LAMB_DIF_MON_KWP		X	X	X	X	X
589E	9E	NOx trap aging factor due to sulphor load	NTA_GI_SUL_KWP	NT_AGI_SUL converted for KWP	-	X	X	X	X	-
		Status VVT-Entlastungsrelais	STATE_RLY_VVL_ADJ_EXT		X	-	-	-	-	-
589F	9F	Time with active catalyst heating for desulfation without active desulfation	T_CH_SO2P_ACT_KWP	T_CH_SO2P_ACT converted for KWP	-	X	X	X	X	-
		VVT Istwinkel	ANG_EXC_VVL_KWP	ANG_EXC_VVL converted	X	-	-	-	-	-
58A0	A0	Km - counter since last end of desulfation	DIST_SO2P_END_KWP	DIST_SO2P_END converted for KWP	-	X	X	X	X	-
		VVT Sollwinkel	ANG_SP_CTL_VVL_KWP	ANG_SP_CTL_VVL converted	X	-	-	-	-	-
58A1	A1	NOx-Konzentration	NOX_NS_KWP	NOX_NS[1] conv erted for KWP	-	X	X	X	X	-
		Ausgegeben Tastverhältnis	PWM_DR_OUT_SET_VVL_KWP	PWM_DR_OUT_SET_VVL converted	X	-	-	-	-	-
58A2	A2	Lineares Lambdasignal NOx-Sensor	LAMB_NOX_SENS_KWP	LAMB_NOX_SENS[1] converted for KWP	-	X	X	X	X	-

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
		VVT Motorstrom	CUR_MOT_VVL_3_KWP	CUR_MOT_VVL_k converted; k = 0...4; use k = 3	X	-	-	-	-	-
58A3	A3	binäres Spannungssignal NOx-Sensor	VLS_NOX_SENS_KWP	VLS_NOX_SENS[1] converted for KWÜ	-	X	X	X	X	-
		VVT Motortemperatur	TEMP_MOT_VVL_KWP	TEMP_MOT_VVL converted	X	-	-	-	-	-
58A4	A4	Status NOx-Sensor	CAN_STATE_NOX_SENS_KWP[1]		-	X	X	X	X	-
		VVT Spannungsversorgung	VCC_DR_VVL_KWP	VCC_DR_VVL converted	X	-	-	-	-	-
58A5	A5	NOx trap sulphur loading	NT_SUL_KWP	NT_SUL converted for KWP	-	X	X	X	X	-
		VVT Sensorversorgungsspannung	V_VCC_SENS_VVL_KWP	V_VCC_SENS_VVL converted	X	-	-	-	-	-
58A6	A6	NOx trap aging factor	NT_AGI_KWP	NT_AGI converted for KWP	-	X	X	X	X	-
		VVT Sensordifferenz	ANG_DE_ABSV_PLAUS_CHK_VVL_KWP	ANG_DE_ABSV_PLAUS_CHK_VVL converted	X	-	-	-	-	-
58A7	A7	Mean value of lambda in multiple-branched exhaust gas lines	LAMB_LS_UP_MV_KWP	LAMB_LS_UP_MV converted for KWP	-	X	X	X	X	-
		VVT Endstufentemperatur	TEMP_SWI_MES_MAX_VVL_KWP	TEMP_SWI_MES_MAX_VVL converted	X	-	-	-	-	-
58A8	A8	Motorabstellzeit	T_ES_KWP	T_ES converted for KWP	X	X	X	X	X	X
58A9	A9	Resetzähler Überwachungsrechner: alter Wert	ENVD_3_MON_3		X	X	X	X	X	X
58AA	AA	Resetzähler Hauptrechner: alter Wert	ENVD_2_MON_3		X	X	X	X	X	X
58AB	AB	Abweichung DK-Ersatzwert und DK-Poti1	TPS_MAF_DIF_INT_1		X	-	-	-	-	-
		Abweichung gemessener DK-Poti1 und modellierter DK-Poti	TPS_DIF_DIAG_COR_1_KWP	TPS_DIF_DIAG_COR_1 converted for KWP	-	X	X	X	X	_
58AC	AC	Abweichung DK-Ersatzwert und DK-Poti2	TPS_MAF_DIF_INT_2		X	-	-	-	-	-
		Abweichung gemessener DK-Poti2 und modellierter DK-Poti	TPS_DIF_DIAG_COR_2_KWP	TPS_DIF_DIAG_COR_2 converted for KWP	-	X	X	X	X	_
58AD	AD	Pedalwertgeber 1	PV_AV_1		X	X	X	X	X	X
58AE	AE	Periodendauer Luftmasse	T_PER_MAF_FRQ_KWP	T_PER_MAF_FRQ_AV[0] converted for KWP	X	X	X	X	X	-
		Periodendauer Luftmasse	T_PER_MAF_FRQ_KWP	T_PER_MAF_FRQ_AV[0]	-	-	-	-	-	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
58AF	AF	Kraftstoff Anforderung an Pumpe	VFF_EFP		X	X	X	X	X	X
58B0	B0	DK-Adaptionsschritt	TPS_AD_STEP_KWP	TPS_AD_STEP converted for KWP	X	X	X	X	X	-
		DK-Adaptionsschritt	TPS_AD_STEP_KWP[0]	TPS_AD_STEP[0] converted for KWP	-	-	-	-	-	X
58B1	B1	Funkenbrenndauer Zylinder 1	V_DUR_IGC[0]		X	X	X	X	X	X
58B2	B2	Funkenbrenndauer Zylinder 5	V_DUR_IGC[1]		X	X	X	X	X	X
58B3	B3	Funkenbrenndauer Zylinder 3	V_DUR_IGC[2]		X	X	X	X	X	X
58B4	B4	Funkenbrenndauer Zylinder 6	V_DUR_IGC[3]		X	X	X	X	X	X
58B5	B5	Funkenbrenndauer Zylinder 2	V_DUR_IGC[4]		X	-	-	X	X	X
58B6	B6	Funkenbrenndauer Zylinder 4	V_DUR_IGC[5]		X	-	-	X	X	X
58B7	B7	Bremsdruck	BRAKE_PRS		X	X	X	X	X	X
58B8	B8	Drehzahl Überwachung	N_32_MON		X	X	X	X	X	X
58B9	B9	Pedalwert Überwachung	PV_AV_MON		X	X	X	X	X	X
58BA	BA	Temperatur PremAir-Sensor	TEMP_EAC_KWP	TEMP_EAC converted for KWP	X	-	-	-	-	-
		Volume fuel flow through the injectors	VFF_MFF_SP_FUP_CTL_KWP	VFF_MFF_SP_FUP_CTL (high byte)	-	X	X	X	X	X
58BB	BB	Status-Byte PremAir-Sensor	EAC_COD_SENS		X	-	-	-	-	-
		Pump speed of the electrical fuel pump as PWM signal	EFPPWM_KWP	EFPPWM converted for KWP	-	X	X	X	X	X
58BC	BC	Luftmasse Überwachung	MAF_MON		X	X	X	X	X	X
		Modellluftmasse Überwachung tiefpassgefiltert	MAF_SUB_COR_MMV_MON		X	-	-	-	-	-
58BD	BD	Zähler für Eisdetection in der Drosselklappe (high byte)	CTR_TPS_JAM_DET_ACT_KWP_H	CTR_TPS_JAM_DET_ACT (high byte)	-	X	X	X	X	_
58BE	BE	Temperaturgradient PremAir-Sensor	TEMP_DIF_EAC_KWP	TEMP_DIF_EAC converted for KWP	X	-	-	-	-	-
		Zähler für Eisdetection in der Drosselklappe (low byte)	CTR_TPS_JAM_DET_ACT_KWP_L	CTR_TPS_JAM_DET_ACT (low byte)	-	X	X	X	X	_
58BF	BF	relative Momentenforderung von MSR über CAN	TQI_MSR_CAN		X	X	X	X	X	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
58C0	C0	Motordrehzahl Eratztwert Überwachung	N_32_SUB_MON		X	X	X	X	X	X
58C1	C1	Laufunruhe Segmentzeit	SEG_T_MES		X	X	X	X	X	X
58C2	C2	Statusbyte MFF-Monitoring	STATE_LV_ERR_MFF_MON_1		-	X	X	X	X	X
58C3	C3	Statusbyte ISC-Monitoring	STATE_LV_ERR_TQ_DIF_ISC_MON_1		-	X	X	X	X	X
58C4	C4	Statusbyte CRU-Monitoring	STATE_LV_ERR_CRU_INH_MON_1		-	X	X	X	X	X
58C5	C5	Drehzahl Überwachung (resetsicher)	N_32_MON_SAVE		-	X	X	X	X	X
58C6	C6	Status Einspritzventile (resetsicher)	PREV_STATE_IV_SAVE		-	X	X	X	X	X
58C7	C7	LL-Solldrehzahlabweichung Überwachung	N_DIF_SP_IS_MON		X	X	X	X	X	X
58C8	C8	I-Anteil Momentdifferenz Überwachung und Modell	TQ_DIF_I_IS_MON		X	X	X	X	X	X
58C9	C9	I-Anteil LL passive Rampe aktiv	LV_PAS_RAMP_ACT_I_IS		X	X	X	X	X	X
58CA	CA	PD-Anteil langsam Leerlaufregelung	TQ_DIF_P_D_SLOW_IS		X	X	X	X	X	X
58CB	CB	PD-Anteil schnell Leerlaufregelung	TQ_DIF_P_D_FAST_IS		X	X	X	X	X	X
58CC	CC	Verlustmoment Überwachung	TQ_LOSS_MON		X	X	X	X	X	X
58CD	CD	Verlustmomentabweichung Überwachung	TQ_LOSS_DIF_MON		X	-	-	-	-	-
58CE	CE	Carrierbyte Schalterstati	STATE_BYTE_SWI_KWP	0x01 LV_IM_BLS 0x02 LV_IM_BTS 0x10 LV_SOF_SWI_REQ 0x80 LV_PAS_RAMP_ACT_I_IS	X	X	X	X	X	X
58CF	CF	Motormoment Sollwert Überwachung	TQI_SP_MON		X	X	X	X	X	-
58CF	CF	Motormoment Sollwert Überwachung	TQI_SP_H_RNG_MON_KWP	TQI_SP_H_RNG_MON converted for KWP	-	-	-	-	-	X
58D0	D0	Motormoment Istwert Überwachung	TQI_AV_MON		X	X	X	X	X	-
58D0	D0	Motormoment Istwert Überwachung	TQI_AV_H_RNG_MON_KWP	TQI_AV_H_RNG_MON converted for KWP	-	-	-	-	-	X
58D1	D1	Moment aktueller Wert	TQI_AV		X	X	X	X	X	X
58D2	D2	Status Luftklappensystem High Byte	STATE_ECRAS_SYS_KWP_H	high byte of STATE_ECRAS_SYS	X	X	X	X	X	-
58D3	D3	Status Luftklappensystem Low Byte	STATE_ECRAS_SYS_KWP_L	low byte of STATE_ECRAS_SYS	X	X	X	X	X	-

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
58D4	D4	Abweichung maximales Moment an Kupplung Überwachung	TQ_MAX_CLU_DIF_MON		X	X	X	X	X	X
58D5	D5	Air temperature up turbo charger	TIA_TCHA_KWP	TIA_TCHA converted for KWP	-	X	X	X	X	-
58D6	D6	Abweichung minimales Moment an Kupplung Überwachung	TQ_MIN_CLU_DIF_MON		X	X	X	X	X	X
58D7	D7	Voltage for temperatur sensor up turbocharger	VP_TIA_TCHA_KWP	VP_TIA_TCHA converted for KWP	-	X	X	X	X	-
58D8	D8	Catalyst temperature sensor raw acquisition	VP_TEG_PCAT_DOWN_KWP	VP_TEG_PCAT_DOWN converted for KWP	-	X	X	X	X	-
58D9	D9	Fehlercode Rechnerüberwachung : aktueller Wert	ENVD_0_MON_3		X	X	X	X	X	X
58DA	DA	Resetzähler Rechnerüberwachung : aktueller Wert	ENVD_1_MON_3		X	X	X	X	X	X
		Fehler Bitfeld high Byte	STATE_N_MAX_MON_HIGH_KWP	STATE_N_MAX_MON (Bits 8-15)	X	-	-	-	-	-
58DB	DB	Inhalt Statusbyte1 Drehzahlüberwachung (resetsicher)	STATE_TQI_N_MAX_MON_1_1_SAVE		-	X	X	X	X	X
		Fehler Bitfeld low Byte	STATE_N_MAX_MON_LOW_KWP	STATE_N_MAX_MON (Bits 0-7)	X	-	-	-	-	-
58DC	DC	Inhalt Statusbyte2 Drehzahlüberwachung (resetsicher)	STATE_TQI_N_MAX_MON_1_2_SAVE		-	X	X	X	X	X
58DD	DD	Pressure upstream the throttle (Turbo)	PUT_KWP	PUT converted for KWP	-	X	X	X	X	-
		Pressure upstream the throttle (Turbo)	PUT_KWP[0]	PUT_MES_BAS[0]	-	-	-	-	-	X
58DE	DE	Voltage of the intake manifold pressure sensor up throttle (for diagnosis)	V_PUT_KWP	V_PUT converted for KWP	-	X	X	X	X	-
		Voltage of the intake manifold pressure sensor up throttle (for diagnosis)	V_PUT_KWP[0]	V_PUT[0]	-	-	-	-	-	X
58DF	DF	Spannung Sportschalter	V_SOF_SWI_KWP	V_SOF_SWI converted for KWP	X	X	X	X	X	-
58E0	E0	Abgleich Drossel-klappenmodell (Faktor)	Eisydk_korfak_b		X	X	X	X	X	X
58E1	E1	Abgleich Drossel-klappenmodell (Offset)	Eisydk_koroff_b		X	-	-	X	X	X
		Status Nullgangerkennung	St_ngang		-	X	X	-	-	-
58E2	E2	Abgleich	Eisyev_korfak_b		X	X	X	X	X	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
		Einlassventilmodell (Faktor)								
58E3	E3	NOx signal shift diagnosis value	RATIO_NS_SHIFT_DIAG_KWP[1]	RATIO_NS_SHIFT_DIAG[1] converted for KWP	-	X	X	X	X	-
		Abgleich Einlassventilmodell (Offset)	Eisyev_koroff_b		X	-	-	-	-	X
58E4	E4	Betriebsart Istwert	Ba_ist		X	X	X	X	X	X
58E5	E5	Lastwert für Aussetzererkennung	LOAD_MIS_KWP	LOAD_MIS converted for KWP	X	X	X	X	X	X
58E6	E6	Nulllastwert für Aussetzererkennung	LOAD_MIN_MIS_KWP	LOAD_MIN_MIS converted for KWP	X	X	X	X	X	X
58E7	E7	Spannung Pedalwertgeber 1: Überwachung	V_PVS_1_MON_KWP	V_PVS_1_MON converted to V_PVS_1_MON_KWP	X	X	X	X	X	X
58E8	E8	Spannung Pedalwertgeber 2: Überwachung	V_PVS_2_MON_KWP	V_PVS_2_MON converted to V_PVS_2_MON_KWP	X	X	X	X	X	X
58E9	E9	Wasserpumpe Spannung	V_CWP		X	X	X	X	X	-
58EA	EA	Wasserpumpe Drehzahl	N_REL_CWP		X	X	X	X	X	-
58EB	EB	Wasserpumpe Drehzahl Soll-Ist-Differenz	N_REL_CWP_DIF		X	X	X	X	X	-
58EC	EC	Wasserpumpe Temperatur Elektronik	TEMP_EL_CWP		X	X	X	X	X	-
58ED	ED	Wasserpumpe Stromaufnahme	CUR_CNS_CWP		X	X	X	X	X	-
		gemittelter Raildruck Bank 1	V_FUP_MV_0_KWP	V_FUP_MV[0] converted for KWP	-	-	-	-	-	X
58EE	EE	Wasserpumpe leistungsreduziert	REL_CWP_PWR		X	X	X	X	X	-
		gemittelter Raildruck Bank 2	V_FUP_MV_1_KWP	V_FUP_MV[1] converted for KWP						X
58EF	EF	Mean value of the acquired sensor voltage	V_FUP_MV_KWP	V_FUP_MV converted for KWP	-	X	X	X	X	-
		gemittelter Raildruck Bank 2	FUP_KWP[1]	FUP_MPL[1] converted for KWP	-	-	-	-	-	X
58F0	F0	Fuel pressure	FUP_KWP	FUP (high byte)	-	X	X	X	X	-
		gemittelter Raildruck Bank 1	FUP_KWP[0]	FUP_MPL[0] converted for KWP	-	-	-	-	-	X
58F1	F1	DME - Losnummer	STATE_LRN_ECU_KWP	STATE_LRN_ECU (high byte)	X	X	X	X	X	X
58F2	F2	PWM signal for the VCV	PWM_VCV_KWP	PWM_VCV (high byte)	-	X	X	X	X	
58F3	F3	Fuel pressure EFP	FUP_EFP_KWP	FUP_EFP converted for KWP	-	X	X	X	X	X
58F4	F4	mean value of low fuel pressure EFP sensor	V_FUP_EFP_MV_KWP	V_FUP_EFP_MV converted for KWP	-	X	X	X	X	X
58F5	F5	Eingangssignal Rückführregler 1	VLS_DIF_LAM_ADJ_KWP[1]	VLS_DIF_LAM_ADJ[1] converted for KWP	X	X	X	X	X	X

ID_ long	ID	Description	Label	Conversion	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
58F6	F6	Eingangssignal Rückführregler 2	VLS_DIF_LAM_ADJ_KWP[2]	VLS_DIF_LAM_ADJ[2] converted for KWP	X	X	X	X	X	X
58F7	F7	Measured opening of the actuator valve	OPG_ACR_KWP	OPG_ACR converted for KWP	-	X	X	X	X	-
58F8	F8	Segmentadaption Laufunruhe Zyl. 5	SEG_AD_MMV_ER[1]		X	X	X	X	X	X
58F9	F9	Segmentadaption Laufunruhe Zyl. 3	SEG_AD_MMV_ER[2]		X	X	X	X	X	X
58FA	FA	canister load fc cps	CL_MMV_SAE		X	X	X	X	X	X
58FB	FB	counter of N-events fc cps (step 2)	SUM_DIAG_DIAGCPS_SAE		X	X	X	X	X	X
58FC	FC	Setpoint request after limitation for actuator position control	OPG_SP_ACR_KWP	OPG_SP_ACR converted for KWP	-	X	X	X	X	-
58FD	FD	Finally duty cycle of digital actuator control	PWM_ACR_KWP	PWM_ACR converted for KWP	-	X	X	X	X	-
58FE	FE	Zähler für Umschaltungen nach HOM durch Monitoring	CTR_SWI_AFS_MON		-	X	X	X	X	-
58FF	FF	-	-	-	-	-	-	-	-	-

7.3.2 Further DDLI- Lists (not for ERRM)

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
200	**Luft / Air**			-	-	-	-	-	
4200	00	TIA	Ansauglufttemperatur 1	X	X	X	X	X	X
4201	01	AMP_MES	Umgebungsdruck	X	X	X	X	X	X
4202	02	MAP_MES	Saugrohrdruck	X	X	X	X	X	X
4203	03	MAF_KGH_MES	Massenstrom vom HFM	X	X	X	X	X	-
4204	04	TAM	Umgebungstemperatur	X	X	X	X	X	X
4205	05	MAP_DIP_MES_BAS	Saugrohrdruck1 / Ladedruck1	X	X	X	X	X	-
4206	06	MAF_KGH_MES_BAS[1]	Saugrohrdruck 1 / Ladedruck 1	-	-	-	-	-	X
4207	07	MAF_KGH_MES_BAS[2]	Saugrohrdruck 2 / Ladedruck 2	-	-	-	-	-	X
4208	08	MAF_KGH_MES[1]	Massenstrom vom HFM Bank 1	-	-	-	-	-	X
4209	09	MAF_KGH_MES[2]	Massenstrom vom HFM Bank 2	-	-	-	-	-	X
				-	-	-	-	-	-
300	**Wasser / Water**			-	-	-	-	-	
4300	00	TCO	Kühlwassertemperatur	X	X	X	X	X	X
4301	01	TCO_2	Kuehlerauslasstemperatur	X	X	X	X	X	X
4302	02	REL_CWP_PWR	Wasserpumpe Leistung ueber BSD	X	X	X	X	X	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
			(Bit Serielle Datenschnittstelle)						
4303	03	TEMP_EL_CWP	Wasserpumpe Elektronik Temperatur	X	X	X	X	X	-
4304	04	CUR_CNS_CWP	Wasserpumpe Strom	X	X	X	X	X	-
4305	05	N_REL_CWP	Wasserpumpe Drehzahl Ist	X	X	X	X	X	-
4306	06	N_REL_CWP_SP	Wasserpumpe Drehzahl Soll	X	X	X	X	X	-
4307	03	Ba_wm_ist	Betriebsart Wasserpumpe	X	X	X	X	X	-
				-	-	-	-	-	
400	**Öl / Oil**			-	-	-	-	-	
4400	00	Oz_nivlangt	Ölstand Mittelwert Langzeit	X	X	X	X	X	X
4401	01	Oz_lp	Füllstand Motoröl	X	X	X	X	X	X
4402	02	Toel	Öltemperatur	X	X	X	X	X	X
4403	03	Oz_kvbsm_ul	Kraftstoff-Verbrauch seit letztem Service	X	X	X	X	X	X
4404	04	Oz_oelkm	km seit letztem Service	X	X	X	X	X	X
4405	05	Oz_nivr	Oelsensor Niveau Rohwert	X	X	X	X	X	X
4406	06	Oz_permr	Oelsensor Qualität Rohwert	X	X	X	X	X	X
4407	07	Oz_tempr	Oelsensor Temperatur Rohwert	X	X	X	X	X	X
4408	08	Oz_tempakt	Oelsensor Temperatur	X	X	X	X	X	X
4409	09	Oz_nivakt	Oelsensor Niveau	X	X	X	X	X	X
440A	0A	Oz_permakt	Oelsensor Qualität	X	X	X	X	X	X
440B	0B	Oz_lf1c	Länderfaktor 1 codiert	X	X	X	X	X	X
440C	0C	Oz_lf2c	Länderfaktor 2 codiert	X	X	X	X	X	X
440D	0D	Oz_lf1t	Länderfaktor 1	X	X	X	X	X	X
440E	0E	Oz_lf2t	Länderfaktor 2	X	X	X	X	X	X
440F	0F	Oz_nivkrzt	Kurzmittelwert-Niveau für den Tester	X	X	X	X	X	X
4410	10	Oz_rwperm	Restweg aus Permittivität abgeleitet	X	X	X	X	X	X
4411	11	Oz_rwkvb	Restweg aus Kraftstoffverbrauch abgeleitet	X	X	X	X	X	X
4412	12	Oz_oelzeit	Oel-Alter in Monate	X	X	X	X	X	X
4413	13	Oz_permlow	aufbereitete Permittivität bei letztem Ölwechsel	X	X	X	X	X	X
4414	14	Oz_permex	Permittivität für Bewertung aufbereitet (extrapoliert)	X	X	X	X	X	X
4415	15	Oz_permoff	Offset für Permittivitätskorrektur	X	X	X	X	X	X
4416	16	Oz_kvbog	zugeteilte Bonuskraftstoffmenge	X	X	X	X	X	X
4417	17	Oz_permbog	zugeteilter Permittivitätsbonus	X	X	X	X	X	X
4418	18	Oz_lv	Status Peilstabanzeige	X	X	X	X	X	X
4419	19	-	-	-	-	-	-	-	-
441A	1A	-	-	-	-	-	-	-	-
441B	1B	-	-	-	-	-	-	-	-
441C	1C	Oz_niv	Ölfüllstand	-	-	-	-	-	X
441D	1D	Oz_peil	Ölfüllstand Peilstab	-	-	-	-	-	X
441E	1E	Oz_krzor	Kurzzeitmittelwert	-	-	-	-	-	X
441F	1F	Oz_vormw	Vormittelwert	-	-	-	-	-	X
4420	20	Oz_langmw	Langzeitmittelwert	-	-	-	-	-	X
4421	21	Oz_oricnt	Orientierungswert Counter	-	-	-	-	-	X
4422	22	Oz_vormwcnt	Vormittelwert Counter	-	-	-	-	-	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
4423	23	Oz_krzcnt	Kurzzeitmittelwert Counter	-	-	-	-	-	X
4424	24	Oz_lgmwcnt	Langzeitmittelwert Counter	-	-	-	-	-	X
4425	25	Oz_rwanz	CBS Restweganzeige	-	-	-	-	-	X
4426	26	Oz_vfanz	CBS Verfügbarkeitsanzeige	-	-	-	-	-	X
4427	27	Oz_rwprog	CBS Restwegprognose	-	-	-	-	-	X
4428	28	Oz_krzlor2k	Kurzmittelwert nach Steigungskorrektur	-	-	-	-	-	X
4429	29	Oz_krzmwkor	korrigierter Kurzmittelwert	-	-	-	-	-	X
442A	2A	B_dmeini	DME initialisiert	-	-	-	-	-	X
500	**MotorBauteile intern / Engine parts intern**								
4500	00	ANG_EXC_VVL	VVT-Excenter Istwert	X	-	-	-	-	-
4501	01	ANG_SP_VVL	VVT-Excenter Sollwert	X	-	-	-	-	-
4502	02	ANG_STK_VVL	Mechanischer Verstellbereich VVT aus Lernroutine	X	-	-	-	-	-
4503	03	ANG_SP_VVL_CUS	Sollwert für Lageregler	X	-	-	-	-	-
4504	04	ANG_SP_EXT_ADJ_VVL	Sollwert für Lageregler vom Tester	X	-	-	-	-	-
4505	05	CAM_SP_IVVT_IN	Sollwert Einlassspreizung	X	X	X	X	X	-
4506	06	PSN_CAM_IN[1]	Nockenwellenposition Einlass	X	X	X	X	X	-
4507	07	PSN_CAM_EX[1]	Nockenwellenposition Auslass	X	X	X	X	X	-
4508	08	CAM_IN[1]	Istwert Einlassspreizung	X	X	X	X	X	-
4509	09	CAM_EX[1]	Istwert Auslassspreizung	X	X	X	X	X	X
450A	0A	CAM_SP_REF_EX	Normspreizung Auslass	X	X	X	X	X	-
450B	0B	CAM_SP_REF_IN	Normspreizung Einlass	X	X	X	X	X	-
450C		IVVTPWM_IN[0]	VANOS PWM Wert Einlass Bank 1	-	-	-	-	-	X
450D		IVVTPWM_IN[1]	VANOS PWM Wert Einlass Bank 2	-	-	-	-	-	X
450E		IVVTPWM_EX[0]	VANOS PWM Wert Auslass Bank 1	-	-	-	-	-	X
450F		IVVTPWM_EX[1]	VANOS PWM Wert Auslass Bank 2	-	-	-	-	-	X
4510		CAM_IN_H[1]	Istwert Einlassspreizung Bank 1	-	-	-	-	-	X
4511		CAM_IN_H[2]	Istwert Einlassspreizung Bank 2	-	-	-	-	-	X
4512		CAM_EX[2]	Istwert Auslassspreizung Bank 2	-	-	-	-	-	X
4513		CAM_SP_IVVT_IN	Sollwert Einlassspreizung Bank 1	-	-	-	-	-	X
4514		CAM_SP_IVVT_IN	Sollwert Einlassspreizung Bank 2	-	-	-	-	-	X
4515		CAM_SP_IVVT_EX	Sollwert Auslassspreizung Bank 1	-	-	-	-	-	X
4516		CAM_SP_IVVT_EX	Sollwert Auslassspreizung Bank 2	-	-	-	-	-	X
4517		CAM_SP_REF_IN	Normspreizung Einlass Bank 1	-	-	-	-	-	X
4518		CAM_SP_REF_IN	Normspreizung Einlass Bank 2	-	-	-	-	-	X
4519		CAM_SP_REF_EX	Normspreizung Auslass Bank 1	-	-	-	-	-	X
451A		CAM_SP_REF_EX	Normspreizung auslass Bank 2	-	-	-	-	-	X
451B		PSN_CAM_IN_2	Nockenwellenposition Einlass Bank 2	-	-	-	-	-	X
451C		PSN_CAM_EX_2	Nockenwellenposition Auslass Bank 2	-	-	-	-	-	X
451D		PSN_AD_CAM_EX_2	Adaptionswert Nockenwelle Auslass Bank 2	-	-	-	-	-	X
451E		PSN_AD_CAM_IN_2	Adaptionswert Nockenwelle Einlass Bank 2	-	-	-	-	-	X
600	**MotorBauteile extern / Engine parts extern**								

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
4600	00	TPS_AV	aktueller Drosselklappenwinkel	X	X	X	X	X	-
4601	01	TPS_SP_MDL	Drosselklappe Sollwert	X	X	X	X	X	-
4602	02	V_ALTER_SP	Generator Sollspannung BSD (Bit Serielle Datenschnittstelle)	X	X	X	X	X	X
4603	03	Tchip	Chiptemperatur Generator1	X	X	X	X	X	X
4604	04	I_gen	Generator Strom	X	X	X	X	X	X
4605	05	Bsdgencv	Chipversion Generator1	X	X	X	X	X	X
4606	06	Bsdgenregv	Reglerversion Generator1	X	X	X	X	X	X
4607	07	Gen_manufak	Herstellercode Generator1	X	X	X	X	X	X
4608	08	Gen_typkenn	Kennung Generatortyp Generator1	X	X	X	X	X	X
4609	09	VB	Kl.87 Spannung / Versorgung DME (Digitale Motor Elektronik)	X	X	X	X	X	X
460A	0A	Ubt	Batteriespannung aktuell	X	X	X	X	X	X
460B	0B	U_batt	Batteriespannung von IBS gemessen	X	X	X	X	X	X
460C	0C	VB_BAS	Batteriespannung vom AD-Wandler DME	X	X	X	X	X	X
460D	0D	Absch_korr	Korrekturwert Abschaltung	X	X	X	X	X	X
460E	0E	D_soc	Abstand zur Startfähigkeitsgrenze	X	X	X	X	X	X
460F	0F	LOAD_BAT	Batterielast	X	X	X	X	X	X
4610	10	VIM_AV	aktuelle Position Disaklappen	X	X	X	X	X	-
4611	11	N_PERC_ECF	Sollwert E-Lüfter als PWM Wert	X	X	X	X	X	X
4612	12	Ierr	Erregerstrom Generator1	X	X	X	X	X	X
4613	13	U_fgen	Kopierter Wert von zum Generator gesendete Sollspannung Generator 1	X	X	X	X	X	X
4614	14	Dfsiggen	Auslastungsgrad Generator 1	X	X	X	X	X	X
4615	15	Ierrfgrenz	Kopie begrenzter Erregerstrom Generator 1	X	X	X	X	X	X
4616	16	Tlrfgen	Kopier Generator1 LR Vorgabe auf Bus gelegt	X	X	X	X	X	X
4617	17	Md_gennm_na	gefiltertes Generatormoment absolut Ausgang	X	X	X	X	X	X
4618	18	B_lrfoff	Kopie Drehzahlschwelle für LR-Funktion Generator 1 aktiv	X	X	X	X	X	X
4619	19	B_bsdprot2	condition for BSD protocol controller type 2	X	X	X	X	X	X
461A	1A	Uregnom	nominal voltage alternator	X	X	X	X	X	X
700	**Lambda**								
4700	00	LV_IPLSL_VLD[1]	WRAF sensor pump current is valid Bank1	X	X	X	X	X	-
		LV_INH_LSCL[1]	Status Lambdasonde betriebsbereit vor Katalysator Bank 1	-	-	-	-	-	X
4701	01	LV_IPLSL_VLD[2]	WRAF sensor pump current is valid Bank2	X	X	X	X	X	-
		LV_INH_LSCL[2]	Status Lambdasonde betriebsbereit vor Katalysator Bank 2	-	-	-	-	-	X
4702	02	VLS_UP_COR[1]	Spannung Lambdasonde Vorkat Bank 1 mit Offsetkorrektur	-	X	X	X	X	X
		VLS_COR_LSL[1]	Spannung Lambdasonde Vorkat Bank 1 mit Offsetkorrektur	X	-	-	-	-	-
4703	03	VLS_UP_COR[2]	Spannung Lambdasonde Vorkat Bank	-	X	X	X	X	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
			2 mit Offsetkorrektur						
		VLS_COR_LSL[2]	Spannung Lambdasonde Vorkat Bank 2 mit Offsetkorrektur	X	-	-	-	-	-
4704	04	LAMB_BAS[1]	Lambda Sollwert Bank1	X	X	X	X	X	X
4705	05	LAMB_BAS[2]	Lambda Sollwert Bank2	X	X	X	X	X	X
4710	10	MFF_ADD_COLD_LAM_AD_INJ[0]		-	X	X	X	X	-
4711	11	MFF_ADD_COLD_LAM_AD_INJ[1]		-	X	X	X	X	-
4712	12	MFF_ADD_COLD_LAM_AD_INJ[2]		-	X	X	X	X	-
4713	13	MFF_ADD_COLD_LAM_AD_INJ[3]		-	X	X	X	X	-
4714	14	MFF_ADD_COLD_LAM_AD_INJ[4]		-	-	-	X	X	-
4715	15	MFF_ADD_COLD_LAM_AD_INJ[5]		-	-	-	X	X	-
4716	16	MFF_ADD_HOT_LAM_AD_INJ[0]		-	X	X	X	X	-
4717	17	MFF_ADD_HOT_LAM_AD_INJ[1]		-	X	X	X	X	-
4718	18	MFF_ADD_HOT_LAM_AD_INJ[2]		-	X	X	X	X	-
4719	19	MFF_ADD_HOT_LAM_AD_INJ[3]		-	X	X	X	X	-
471A	1Â	MFF_ADD_HOT_LAM_AD_INJ[4]		-	-	-	X	X	-
471B	1B	MFF_ADD_HOT_LAM_AD_INJ[5]		-	-	-	X	X	-
471C	1C	DIST_LAM_AD_INJ_COLD		-	X	X	X	X	-
471D	1D	DIST_LAM_AD_INJ_HOT		-	X	X	X	X	-
471E	1E	CTR_AD_COLD_LAM_AD_INJ		-	X	X	X	X	-
471F	1F	CTR_AD_HOT_LAM_AD_INJ		-	X	X	X	X	-
4720	20	RATIO_MMV_NS_SHIFT_DIAG[1]	NOx signal shift diagnosis mean value	-	X	X	X	X	-
4721	21	CTR_NS_SHIFT_CYC[1]	Counter of finished NOx signal shift diagnosis	-	X	X	X	X	-
4721	22	DIST_NT_NS_SHIFT[1]	Distance saved at last successful NOx signal shift diagnosis	-	X	X	X	X	-
800	**Sonstiges / Other**								
4800	00	LV_CS	Kupplungsschalter Status	X	X	X	X	X	X
4801	01	LV_CS_CUS	Kupplungsschalter vorhanden	X	X	X	X	X	X
4802	02	LV_SOF_SWI	Sporttaster aktiv	X	X	X	X	X	X
4803	03	STATE_ACIN_CAN	Status Klima ein	X	X	X	X	X	X
4804	04	LV_SAP	Sekundärluft Pumpe aktiv	X	-	-	-	-	X
4805	05	LV_RLY_ST_CAN	Startrelais über CAN aktiv	X	X	X	X	X	X
4806	06	TECU	Steuergeraete-Innentemperatur	X	X	X	X	X	X
4807	07	N	Motor Drehzahl	X	X	X	X	X	X
4808	08	N_SP_IS	Leerlauf Solldrehzahl	X	X	X	X	X	X
4809	09	LV_IS	Status LL	X	X	X	X	X	X
480A	0A	CTR_KM_BN	Kilometerstand Auflösung 1 km	X	X	X	X	X	X
480B	0B	PV_AV	Pedalwert Fahrerwunsch in %	X	X	X	X	X	X
480C	0C	VP_TIA[1]	Spannung Ansauglufttemperatur 1	-	-	-	-	-	X
480D	0D	VP_TIA[2]	Spannung Ansauglufttemperatur 2	-	-	-	-	-	X
480E	0E	TIA_MES[1]	Rohwert Ansauglufttemperatur 1	-	-	-	-	-	X
480F	0F	TIA_MES[2]	Rohwert Ansauglufttemperatur 2	-	-	-	-	-	X
4810	10	VP_TEG_PCAT_DOWN_1	Sensorspannung Abgastemperatur 1	-	-	-	-	-	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
4811	11	VP_TEG_PCAT_DOWN_2	Sensorspannung Abgastemperatur 2	-	-	-	-	-	X
4812	12	TEG_PCAT_DOWN_1	Abgastemperatur 1	-	-	-	-	-	X
4813	13	TEG_PCAT_DOWN_2	Abgastemperatur 2	-	-	-	-	-	X
4814	14	PVDKDS_I_[0]	Druck vor Drosselklappe Bank 1	-	-	-	-	-	X
4815	15	PS_IST_I_[0]	Druck nach Drosselklappe Bank 1	-	-	-	-	-	X
4816	16	TANS_I_[0]	Temperatur vor Drosselklappe Bank 1	-	-	-	-	-	X
4817	17	PVDKDS_I_[1]	Druck vor Drosselklappe Bank 2	-	-	-	-	-	X
4818	18	PS_IST_I_[1]	Druck nach Drosselklappe Bank 2	-	-	-	-	-	X
4819	19	Tans_i[1]	Temperatur vor Drosselklappe Bank 2	-	-	-	-	-	X
481A	1A	FUP_H_MPL[0]	Raildruck Bank 1	-	-	-	-	-	X
481B	1B	FUP_H_MPL[1]	Raildruck Bank 2	-	-	-	-	-	X
481C	1C	VP_TPS_1[1]	ADC-Wert1 Drosselklappe Bank 1	-	-	-	-	-	X
481D	1D	VP_TPS_2[1]	ADC-Wert2 Drosselklappe Bank 1	-	-	-	-	-	X
481E	1E	VP_TPS_1[2]	ADC-Wert1 Drosselklappe Bank 2	-	-	-	-	-	X
481F	1F	VP_TPS_2[2]	ADC-Wert2 Drosselklappe Bank 2	-	-	-	-	-	X
4820	20	TPS_AV[1]	Drosselklappenwinkel Bank 1	-	-	-	-	-	X
4821	21	TPS_AV[2]	Drosselklappenwinkel Bank 2	-	-	-	-	-	X
4822	22	TPS_SP_MDL[1]	Drosselklappe Sollwert Bank 1	-	-	-	-	-	X
4823	23	TPS_SP_MDL[2]	Drosselklappe Sollwert Bank 2	-	-	-	-	-	X
4824	24	LV_RLY_MTC_HEAT[1]	Status Drosselklappenheizungsrelais Bank 1	-	-	-	-	-	X
4825	25	LV_RLY_MTC_HEAT[2]	Status Drosselklappenheizungsrelais Bank 2	-	-	-	-	-	X
4826	26	STATE_MTC_HEAT	Status Drosselklappenheizung Bank 1	-	-	-	-	-	X
4827	27	STATE_MTC_HEAT	Status Drosselklappenheizung Bank 2	-	-	-	-	-	X
4828	28	LV_EF	Status Abgasklappe 1	-	-	-	-	-	X
4829	29	LV_EF	Status Abgasklappe 2	-	-	-	-	-	X
482A	2A	LV_SAV	Status Sekundärluftventil 1	-	-	-	-	-	X
482B	2B	LV_SAV	Status Sekundärluftventil 1	-	-	-	-	-	X
482C	2C	TPS_SP[1]	Drosselklappe Sollwert Bank 1	-	-	-	-	-	X
482D	2D	TPS_SP[2]	Drosselklappe Sollwert Bank 2	-	-	-	-	-	X
5A	**MSx specific**								
5A00	00	VCC_PVS_1	Versorgung FWG 1	X	X	X	X	X	X
5A01	01	VCC_PVS_2	Versorgung FWG 2	X	X	X	X	X	X
5A02	02	LV_CDN_DIAG_TCHA_LEAK	Boolean for diagnosis condition present on charger leakage diagnosis	-	X	X	X	X	X
		V_VCC_SENS_VVL_RAW	Versorgung VVT Rohwert	X	-	-	-	-	-
5A03	03	LV_END_DIAG_TCHA_LEAK	End of diagnosis charger leakage diagnosis	-	X	X	X	X	X
		V_VCC_SENS_VVL	Versorgung VVT	X	-	-	-	-	-
5A04	04	V_PVS_1	Spannung Pedalwertgeber 1	X	X	X	X	X	X
5A05	05	V_PVS_2	Spannung Pedalwertgeber 2	X	X	X	X	X	X
5A06	06	V_TPS_1	Spannung Drosselklappe Poti 1	X	X	X	X	X	-
5A07	07	V_TPS_2	Spannung Drosselklappe Poti 2	X	X	X	X	X	-
5A08	08	V_TIA	Spannung Ansauglufttemperatur	-	X	X	X	X	_

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
		VP_TIA	Spannung Ansauglufttemperatur	X	-	-	-	-	-
5A09	09	V_TCO	Spannung Motortemperatur	-	X	X	X	X	_
		VP_TCO[1]	Spannung Motortemperatur	X	-	-	-	-	X
5A0A	0A	V_TCO_2	Spannung Kühlmitteltemperatur Kühlerausgang	-	X	X	X	X	_
		VP_TCO[2]	Spannung Kühlmitteltemperatur Kühlerausgang	X	-	-	-	-	X
5A0B	0B	V_AMP	Spannung DME Umgebungsdruck	X	X	X	X	X	X
5A0C	0C	V_MAF	Spannung Luftmasse	X	X	X	X	X	_
5A0D	0D	V_SAF	Spannung Sekundärluft	X	X	X	X	X	X
5A0E	0E	VP_TECU	Spannung SG-Innentemperatur	X	X	X	X	X	X
5A0F	0F	V_IGK_BAS	Spannung Kl.15	X	X	X	X	X	X
5A10	10	V_IGK_MES	Spannung Kl15	X	X	X	X	X	X
5A11	11	VLS_UP[1]	Spannung Lambdasonde VorKat 1	X	X	X	X	X	X
5A12	12	VLS_UP[2]	Spannung Lambdasonde VorKat 2	X	X	X	X	X	X
5A13	13	VLS_DOWN[1]	Spannung Lambdasonde HinterKat 1	X	X	X	X	X	X
5A14	14	VLS_DOWN[2]	Spannung Lambdasonde HinterKat 2	X	X	X	X	X	X
5A15	15	LV_END_DIAG_TCHA_PRS_LOW		-	-	-	X	X	X
5A16	16	V_PBSU	Spannung Bremsunterdrucksensor	-	X	X	-	-	-
		LV_END_DIAG_TCHA_PRS_HIGH		-	-	-	X	X	X
5A17	17	V_DMTL	Spannung Strommessung DMTL	X	X	X	X	X	X
5A18	18	VP_TEG_PCAT_DOWN	Spannung Katalysator-Temperatursensor	-	X	X	X	X	-
5A19	19			-	-	-	-	-	-
5A1A	1A			-	-	-	-	-	-
5A1B	1B			-	-	-	-	-	-
5A1C	1C			-	-	-	-	-	-
5A1D	1D			-	-	-	-	-	-
5A1E	1E	PBSU	Bremsunterdruck	-	X	X	X	X	_
5A1F	1F	TEG_PCAT_DOWN_1	Abgastemperatur	-	X	X	X	X	-
5A20	20	PWM_NEUT_PSN_GB	Tastverhältnis Nullgangssensor	-	X	X	X	X	-
5A21	21	TCO_2_MES	Kühlmitteltemperatur Kühlerausgang	X	X	X	X	X	X
5A22	22	TECU	Steuergerät Innentemperatur	X	X	X	X	X	X
5A23	23	P_oel_soll	Sollwert Öldruck	X	X	X	X	X	-
5A24	24	TPS_SP	Drosselklappe Sollwert	X	X	X	X	X	-
5A25	25	P_oel_ist	Istwert Öldruck	X	X	X	X	X	-
5A26	26	MAP	Umgebungsdruck	X	X	X	X	X	X
5A27	27	PV_AV_1	Pedalwertgeber Poti1	X	X	X	X	X	X
5A28	28	PV_AV_2	Pedalwertgeber Poti2	X	X	X	X	X	X
5A29	29	PV_AV_RAW	Fahrpedalwert	X	X	X	X	X	X
5A2A	2A	SAF_KGH_MES_BAS	Luftmasse Sekundärluft	X	-	-	-	-	-
5A2B	2B	Tans	Temperatur vor Drosselklappe	-	X	X	X	X	-
5A2C	2C	Pvdkds	Druck vor Drosselklappe	-	X	X	X	X	-
5A2D	2D	Ps_ist	Druck nach Drosselklappe	-	X	X	X	X	_
5A2E	2E	FUP_EFP	Kraftstoffniederdrucksensor	-	X	X	X	X	X
5A2F	2F	FUP	Raildruck	-	X	X	X	X	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5A30	30	ER_CYL[0]	Laufunruhe Zylinder 1	X	X	X	X	X	X
5A31	31	ER_CYL[4]	Laufunruhe Zylinder 2 (only 6cyl)	X	X	X	X	X	X
5A32	32	ER_CYL[2]	Laufunruhe Zylinder 3 (6cyl) Laufunruhe Zylinder 4 (4cyl)	X	X	X	X	X	X
5A33	33	ER_CYL[5]	Laufunruhe Zylinder 4 (only 6cyl)	X	X	X	X	X	X
5A34	34	ER_CYL[1]	Laufunruhe Zylinder 5 (6cyl) Laufunruhe Zylinder 3 (4cyl)	X	X	X	X	X	X
5A35	35	ER_CYL[3]	Laufunruhe Zylinder 6 (6cyl) Laufunruhe Zylinder 2 (4cyl)	X	X	X	X	X	X
5A36	36	LV_KNK	Status Klopfen	X	X	X	X	X	X
5A37	37	NL[0]	Spannung Klopfwerte Zylinder 1	X	X	X	X	X	X
5A38	38	NL[4]	Spannung Klopfwerte Zylinder 2 (only 6cyl)	X	X	X	X	X	X
5A39	39	NL[2]	Spannung Klopfwerte Zylinder 3 (6cyl) Spannung Klopfwerte Zylinder 4 (4cyl)	X	X	X	X	X	X
5A3A	3A	NL[5]	Spannung Klopfwerte Zylinder 4 (only 6cyl)	X	X	X	X	X	X
5A3B	3B	NL[1]	Spannung Klopfwerte Zylinder 5 (6cyl) Spannung Klopfwerte Zylinder 3 (4cyl)	X	X	X	X	X	X
5A3C	3C	NL[3]	Spannung Klopfwerte Zylinder 6 (6cyl) Spannung Klopfwerte Zylinder 2(4cyl)	X	X	X	X	X	X
5A3D	3D	KNKS[0]	Klopfsignal Zylinder 1	X	X	X	X	X	X
5A3E	3E	KNKS_REL_NL[0]	Klopfsignal Zylinder 1 relativ	X	X	X	X	X	-
		KNKS[1]	Klopfsignal Zylinder 5	-	-	-	-	-	X
5A3F	3F	KNKS[5]	Klopfsignal Zylinder 3 (for 4cyl. machine: 0ffh –not relevant)	X	X	X	X	X	X
5A40	40	KNKS_REL_NL[5]	Klopfsignal Zylinder 6 relativ (for 4cyl. machine: 0ffh –not relevant)	X	X	X	X	X	-
		KNKS[2]	Klopfsignal Zylinder 4	-	-	-	-	-	X
5A41	41	NT_AGI_SUL	NOx trap aging factor due to sulphur load	-	X	X	X	X	-
		KNKS[3]		-	-	-	-	-	X
5A42	42	TI_1[0]	Einspritzzeit Zylinder 1	X	-	-	-	-	-
		NT_AGI	NOx trap aging factor	-	X	X	X	X	-
		KNKS[4]	Klopfsignal Zylinder	-	-	-	-	-	X
5A43	43	TI_1[4]	Einspritzzeit Zylinder 2	X	-	-	-	-	-
		NT_AGI_THERMO	NOx trap aging factor due to thermal aging	-	X	X	X	X	-
		KNKS[6]	Klopfsignal Zylinder 7	-	-	-	-	-	X
5A44	44	TI_1[2]	Einspritzzeit Zylinder 3	X	-	-	-	-	-
		LV_SO2P_REQ_2	Request of a desulfation (forces catalyst heating)	-	X	X	X	X	-
		KNKS[7]	Klopfsignal Zylinder 2	-	-	-	-	-	X
5A45	45	TI_1[5]	Einspritzzeit Zylinder 4	X	-	-	-	-	-
		CTR_CH_SO2P	Counter of not effectual trials to heat the NOx - Trap for desulfation	-	X	X	X	X	-
		ER_CYL[6]	Laufunruhe Zylinder 7	-	-	-	-	-	X
5A46	46	TI_1[1]	Einspritzzeit Zylinder 5	X	-	-	-	-	-
		DIST_SO2P_END	Km - counter since last end of desulfation	-	X	X	X	X	-
		ER_CYL[7]	Laufunruhe Zylinder 8	-	-	-	-	-	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5A47	47	TI_1[3]	Einspritzzeit Zylinder 6	X	-	-	-	-	-
		T_CH_SO2P_ACT	Time with active catalyst heating for desulfation without active desulfation	-	X	X	X	X	-
		NL[6]	Spannung Klopfwerte Zylinder 7	-	-	-	-	-	X
5A48	48	NL[7]	Spannung Klopfwerte Zylinder 2	-	-	-	-	-	X
		LC_SWI_AEB_TYP	Active engine brackets switch for different engine type	X	X	X	X	X	-
5A49	49	IGA_IGC[0]	Zündwinkel Zylinder1	X	X	X	X	X	X
5A4A	4A			-	-	-	-	-	-
5A4B	4B	LOAD_CLC	Berechneter Lastwert	X	X	X	X	X	X
5A4C	4C	STATE_MTC_HEAT	Status Drosselklappenheizungsrelais	-	X	X	X	X	_
5A4D	4D	LV_RLY_MTC_HEAT	Drosselklappenheizung EIN	-	X	X	X	X	_
5A4E	4E	LV_ACCOUT_RLY	Klimakompressorrelais Ein	X	X	X	X	X	X
5A4F	4F	LV_STATE_RLY_VVL	VVT- Entlastungsrelais Ein	X	-	-	-	-	-
5A50	50	LAMB_LS_UP[1]	Lambdawert vor Kat Bank1	X	X	X	X	X	X
5A51	51	LAMB_LS_UP[2]	Lambdawert vor Kat Bank2	X	X	X	X	X	X
5A52	52	LV_LS_DOWN_READY[1]	Status LS nach Kat Bank1	X	X	X	X	X	X
5A53	53	LV_LS_DOWN_READY[2]	Status LS nach Kat Bank2	X	X	X	X	X	X
5A54	54	STATE_LSH_DOWN[1]	Status LS Heizung nach Kat Bank1	X	X	X	X	X	X
5A55	55	STATE_LSH_DOWN[2]	Status LS Heizung nach Kat Bank2	X	X	X	X	X	X
5A56	56	STATE_LSH_UP[1]	Status LS Heizung vor Kat Bank1	X	X	X	X	X	X
5A57	57	STATE_LSH_UP[2]	Status LS Heizung vor Kat Bank2	X	X	X	X	X	X
5A58	58	LSHPWM_UP[1]	Lambdasondenheizung PWM vor Kat Bank 1	X	X	X	X	X	X
5A59	59	LSHPWM_DOWN[1]	Lambdasondenheizung PWM nach Kat Bank 1	X	X	X	X	X	X
5A5A	5A	LSHPWM_UP[2]	Lambdasondenheizung PWM vor Kat Bank 2	X	X	X	X	X	X
5A5B	5B	LSHPWM_DOWN[2]	Lambdasondenheizung PWM nach Kat Bank 2	X	X	X	X	X	X
5A5C	5C	ERR_VIMPWM_1_FB	Aktive Fehlerrückmeldung DISA-Klappe 1	X	X	X	X	X	-
5A5D	5D	CTR_VIMPWM_1_EDGE	Schalthäufigkeitszähler DISA-Klappe 1	X	X	X	X	X	-
5A5E	5E	ERR_VIMPWM_2_FB	Aktive Fehlerrückmeldung DISA-Klappe 2	X	X	X	X	X	-
5A5F	5F	CTR_VIMPWM_2_EDGE	Schalthäufigkeitszähler DISA-Klappe 2	X	X	X	X	X	-
		LV_RLY_HPDI	Status HPDI-Relais	-	-	-	-	-	X
5A60	60	LV_IM_BLS	Bremslichtschalter Ein	X	X	X	X	X	X
5A61	61	LV_IM_BTS	Bremslichttestschalter Ein	X	X	X	X	X	X
5A62	62	LV_POIL_SWI	Öldruck erreicht, Schalter o. Sensor (4Zyl)	X	X	X	X	X	X
5A63	63	LV_EBOX_CFA	E-Boxlüfter Ein	X	X	X	X	X	X
5A64	64	LV_SWI_AEB	Motorlager weiche Dämpfung	X	X	X	X	X	-
5A65	65	LV_EF	Abgasklappe Ein	X	X	X	X	X	-
5A66	66	LV_DMTL_PUMP	DMTL Pumpe Ein	X	X	X	X	X	X
5A67	67	LV_DMTLS	DMTL Ventil Ein	X	X	X	X	X	X
5A68	68	LV_HDMTL_ON	DMTL Heizung Ein	X	X	X	X	X	X
5A69	69	LV_MIL_CAN	MIL Lampe Ein	X	X	X	X	X	X
5A6A	6A	LV_CRU_MAIN_SWI	Lampe FGR Ein	X	X	X	X	X	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5A6B	6B	LV_WAL_1_CAN	Lampe Check Engine Ein	X	X	X	X	X	X
5A6C	6C	FAC_FCO_KWP	Verbrauchskorrekturfktor	-	X	X	X	X	X
5A6D	6D	STATE_MSW_CAN	Status Taste FGR	X	X	X	X	X	X
5A6E	6E	STATE_CRU_OFF_IRR	Status für irreversible Abschaltbedingung	-	X	X	X	X	X
5A6F	6F	STATE_CRU_OFF_REV	Status für reversible Abschaltbedingung	-	X	X	X	X	X
5A70	70	LV_SOF	Soundklappe Zustand	-	X	X	X	X	X
5A71	71	VIMPWM_1	DISA1 PWM (große/obere Klappe)	X	X	X	X	X	-
5A72	72	VIMPWM_2	DISA2 PWM (kleine/untere Klappe)	X	X	X	X	X	-
5A73	73	LV_RLY_CRCV_HEAT	Kurbelgehaeuseentlueftungsheizung	-	X	X	X	X	X
5A74	74	ECTPWM	Beheizter Thermostat PWM	X	X	X	X	X	X
5A75	75	LV_SAV	Sekundärluft Ventil	X	-	-	-	-	-
5A76	76	CPPWM_ADD_AD_MEM	Adaption Öffnungspunkt Tankentlüftungsventil	X	X	X	X	X	X
5A77	77	CPPWM_CPS	TankEntlüftungsVentil TEV PWM	X	X	X	X	X	X
5A78	78	LV_EF	Abgasklappe Ansteuerung	X	X	X	X	X	-
5A79	79	ECFPWM[0]	E-Lüfter PWM	X	X	X	X	X	X
5A7A	7A	IVVTPWM_IN[0]	VANOS PWM Wert Einlass	X	X	X	X	X	X
5A7B	7B	IVVTPWM_EX[1]	VANOS PWM Wert Auslass	X	X	X	X	X	X
5A7C	7C	FAC_NOX_NS_AD[1]	Adaptation of the NOx sensor characteristic shift	-	X	X	X	X	-
5A7D	7D	CTR_NS_AD_CYC[1]	Counter of NOx signal gain adaptations	-	X	X	X	X	-
5A7E	7E	DIST_NT_NS_AD[1]	Current distance from last NOx signal gain adaptation	-	X	X	X	X	-
5A7F	7F	DELTA_CRK_CYL_LAM[1]	adapted phase displacement, bank1	-	X	X	X	X	X
5A80	80	DELTA_CRK_CYL_LAM[2]	adapted phase displacement, bank 2	-	X	X	X	X	X
5A81	81	FAC_LAM_LIM[1]	Integrator Bank1	X	X	X	X	X	X
5A82	82	FAC_LAM_LIM[2]	Integrator Bank2	X	X	X	X	X	X
5A83	83	MFF_ADD_LAM_AD_OUT[1]	Adaption Offset Lambda Bank1	-	X	X	X	X	X
		MFF_AD_ADD_MMV_REL[1]	Adaption Lambda Bank1, additive	X	-	-	-	-	-
5A84	84	MFF_ADD_LAM_AD_OUT[2]	Adaption Offset Lambda Bank2	-	X	X	X	X	X
		MFF_AD_ADD_MMV_REL[2]	Adaption Lambda Bank, additive2	X	-	-	-	-	-
5A85	85	FAC_LAM_AD_CUS[1]	Adaption Multiplikation Lambda Bank1	-	X	X	X	X	X
		MFF_AD_FAC_MMV_REL[1]	Adaption Lambda Bank1, multiplicative	X	-	-	-	-	-
5A86	86	FAC_LAM_AD_CUS[2]	Adaption Multiplikation Lambda Bank2	-	X	X	X	X	X
		MFF_AD_FAC_MMV_REL[2]	Adaption Lambda Bank2, multiplicative	X	-	-	-	-	-
5A87	87	LAMB_DELTA_AD_LAM_ADJ[1]	Adaptionswert Trimregelung Bank1	X	X	X	X	X	X
5A88	88	LAMB_DELTA_AD_LAM_ADJ[2]	Adaptionswert Trimregelung Bank2	X	X	X	X	X	X
5A89	89	FAC_H_RNG_LAM_AD[1]	multiplikative Gemischadaption hohe Last Bank1	X	X	X	X	X	X
5A8A	8A	FAC_H_RNG_LAM_AD[2]	multiplikative Gemischadaption hohe Last Bank2	X	X	X	X	X	X
5A8B	8B	FAC_L_RNG_LAM_AD[1]	multiplikative Gemischadaption niedrige Last Bank1	X	X	X	X	X	X
5A8C	8C	FAC_L_RNG_LAM_AD[2]	multiplikative Gemischadaption niedrige Last Bank2	X	X	X	X	X	X
5A8D	8D	MFF_ADD_LAM_AD[1]	additive Gemischadaption Leerlauf	X	X	X	X	X	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
			Bank1						
5A8E	8E	MFF_ADD_LAM_AD[2]	additive Gemischadaption Leerlauf Bank2	X	X	X	X	X	X
5A8F	8F	FAC_LSL_GAIN_AD[1]	Adaption Schubabgleich Bank 1	-	X	X	X	X	X
5A90	90	FAC_LSL_GAIN_AD[2]	Adaption Schubabgleich Bank 2	-	X	X	X	X	X
5A91	91	CAT_DIAG[1]	Katalysatordiagnosewert Bank1	X	-	-	-	-	-
		EFF_CAT_DIAG_OBD[1]	Katalysatordiagnosewert Bank1	-	X	X	X	X	X
5A92	92	CAT_DIAG[2]	Katalysatordiagnosewert Bank2	X	-	-	-	-	-
		EFF_CAT_DIAG_OBD[2]	Katalysatordiagnosewert Bank2	-	X	X	X	X	X
5A93	93	CTR_GB_NEUT_NOT_PLAUS_SUM		-	X	X	X	X	-
5A94	94	CAM_SP_IVVT_EX	Nockenwelle Auslass Sollwert	X	X	X	X	X	-
5A95	95	PSN_AD_CAM_EX_1	Adaptionswert Nockenwelle Auslaß	X	X	X	X	X	X
5A96	96	PSN_AD_CAM_IN_1	Adaptionswert Nockenwelle Einlaß	X	X	X	X	X	X
5A97	97	B_vsean_loc	E-VANOS im Anschlag bei letztem Abstellen	X	X	X	X	X	X
5A98	98			-	-	-	-	-	-
5A99	99	LV_SEG_AD_AVL_ER	Kurbelwellen Adaption beendet	X	X	X	X	X	X
5A9A	9A	LV_VAR_MAF_LEARNT	HFM learning function has been carried out	-	X	X	X	X	-
5A9B	9B	FAC_LAM_AD_BAL[1]	Lambda adaption output to be used in the MFF correction, bank1	-	X	X	X	X	-
5A9C	9C	FAC_LAM_AD_BAL[2]	Lambda adaption output to be used in the MFF correction, bank2	-	X	X	X	X	-
5A9D	9D	FAC_LAM_AD_OUT[1]	fuel mass set point factor, output from lambda adaptation, bank 1	-	X	X	X	X	-
5A9E	9E	FAC_LAM_AD_OUT[2]	fuel mass set point factor, output from lambda adaptation, bank 2	-	X	X	X	X	-
5A9F	9F	FAC_LAM_ADJ_COR_LAM_AD_CUS[1]	Corrected value of the Long term customer Lambda Adaption (multiplicative share), bank1	-	X	X	X	X	-
5AA0	A0	FAC_LAM_ADJ_COR_LAM_AD_CUS[2]	Corrected value of the Long term customer Lambda Adaption (multiplicative share), bank2	-	X	X	X	X	-
5AA1	A1	STATE_EOL_KWP_CPS	Status Diagnose TEV	X	X	X	X	X	-
5AA2	A2	STATE_EOL_KWP_DMTL	Status Diagnose DMTL	X	X	X	X	X	-
5AA3	A3	STATE_EOL_KWP_VLS	Status Diagnose Lambdasonden	X	X	X	X	X	-
5AA4	A4	STATE_EOL_KWP_N_SP_IS	Status Diagnose Leerlaufdrehzahlverstellung	X	X	X	X	X	-
5AA5	A5	STATE_EOL_KWP_SA	Status Diagnose Sekundärluft	X	X	X	X	X	-
5AA6	A6	STATE_EOL_KWP_VVL_AD	Status Diagnose VVT Anschläge lernen	X	-	-	-	-	-
5AA7	A7	Msnlgofs_tmp	Leckluftadaptation Istwert	-	X	X	X	X	X
5AA8	A8	STATE_ECRAS_SYS	Status Luftklappensystem	X	X	X	X	X	X
5AA9	A9	ECRASPWM	Tastverhältnis: Luftklappe	X	X	X	X	X	-
5AAA	AA	POIL_PWM	Oilpressure PWM	X	X	X	X	X	_
5AAB	AB	WGPWM[0]	Wastegate 1 PWM	-	X	X	X	X	X
5AAC	AC	WGPWM[1]	Wastegate 2 PWM	-	X	X	X	X	X
5AAD	AD	Atlvst	Vorsteuerung Ladedruckregelung	-	X	X	X	X	-
5AAE	AE	Atlr	Reglerausgang und Vorsteuerung	-	X	X	X	X	-
5AAF	AF	F_atlad	Adaptionswert von der	-	X	X	X	X	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
			Ladedruckregelung						
5AB0	B0	PLDR_SOLL	Solladedruck	-	X	X	X	X	X
		T_PER_MAF_FRQ[1]	Periodendauer Luftmasse 2	-	-	-	-	-	X
5AB1	B1	VS	Geschwindigkeit	X	X	X	X	X	X
5AB2	B2	T_PER_MAF_FRQ[0]	Periodendauer Luftmasse	X	X	X	X	X	X
5AB3	B3	DIST_ACT_MIL	Fahrstreck mit MIL an	X	X	X	X	X	X
5AB4	B4	TRT	Betriebsstundenzähler	X	X	X	X	X	X
5AB5	B5	LV_VAR_SAP	Variante Sekundärluftpumpe	X	-	-	-	-	-
5AB6	B6	TIA_MES	Rohwert Ansauglufttemperatur 1	X	X	X	X	X	-
5AB7	B7	TCO_MES	Rohwert Kühlwassertemperatur	X	X	X	X	X	X
5AB8	B8	V_MAP	Spannung Saugrohrdruck	X	X	X	X	X	-
		V_MAP[0]	Spannung Saugrohrdruck 1	-	-	-	-	-	X
5AB9	B9	V_SOF_SWI	Spannung Sportschalter	X	X	X	X	X	X
5ABA	BA	EFPPWM	PWM Kraftstoffpumpe	X	X	X	X	X	X
5ABB	BB	ANG_REL_VVL	Ausgewählter VVT Istwinkel	X	-	-	-	-	-
5ABC	BC	MAF_KGH_MES_BAS	Luftmasse	X	X	X	X	X	-
5ABD	BD	LV_RLY_ST	Starterrelais aktiv	X	X	X	X	X	X
5ABE	BE			-	-	-	-	-	-
5ABF	BF			-	-	-	-	-	-
5AC0	C0			-	-	-	-	-	-
5AC1	C1			-	-	-	-	-	-
5AC2	C2	RST_DBG_BACKTRACE_ADDRESS[0][0]	Info last available caller address (default 0)	X	X	X	X	X	-
		STACK_ADR_RST[0][0]	Info last available caller addresses (default 0) / RESET SAFE	-	-	-	-	-	X
5AC3	C3	CTR_TPS_JAM_DET_ACT	Zähler für Eisdetection in der Drosselklappe	-	X	X	X	X	_
5AC4	C4	EFPPWM_MIN_AD	Minimum pump speed of the electrical fuel pump as PWM signal	-	X	X	X	X	-
5AC5	C5	EFPPWM_I_AD	Adaptive I-Part of the controller	-	X	X	X	X	-
5AC6	C6	V_ACR	Sensorspannung AGR	-	X	X	X	X	-
5AC7	C7	OPG_ACR	Hub des AGR-Tellerventils	-	X	X	X	X	-
5AC8	C8	V_ACR_AD_TOL	Adaptionswert oberer Anschlag (einmalig gelernt)	-	X	X	X	X	-
5AC9	C9	V_ACR_AD_BOL	Adaptionswert unterer Anschlag (immer wieder neu gelernt)	-	X	X	X	X	-
5ACA	CA	V_ACR_AD_BOL_0	Adaptionswert unterer Anschlag (einmalig am Anfang gelernt, Uradaption)	-	X	X	X	X	-
5ACB	CB	STATE_ACR_AD	Status des Erlernens der AGR-Adaption	-	X	X	X	X	-
5ACC	CC	CTR_STC_TECU_1	DME-Temperaturstatistik, Zähler 1	X	X	X	X	X	-
5ACD	CD	CTR_STC_TECU_2	DME-Temperaturstatistik, Zähler 2	X	X	X	X	X	-
5ACE	CE	CTR_STC_TECU_3	DME-Temperaturstatistik, Zähler 3	X	X	X	X	X	-
5ACF	CF	CTR_STC_TECU_4	DME-Temperaturstatistik, Zähler 4	X	X	X	X	X	-
5AD0	D0	CTR_STC_TECU_5	DME-Temperaturstatistik, Zähler 5	X	X	X	X	X	-
5AD1	D1	CTR_STC_TECU_6	DME-Temperaturstatistik, Zähler 6	X	X	X	X	X	-
5AD2	D2	CTR_STC_TECU_7	DME-Temperaturstatistik, Zähler 7	X	X	X	X	X	-
5AD3	D3	CTR_STC_TECU_8	DME-Temperaturstatistik, Zähler 8	X	X	X	X	X	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5AD4	D4			-	-	-	-	-	-
5AD5	D5	LV_RAS_OUT		-	-	-	X	X	-
5AD6	D6	NOX_OFS_PUC[1]	Fuel Cut Off = Schubabschaltung	-	X	X	X	X	-
5AD7	D7	NOX_OFS_LOAD[1]	Beladungsbetrieb NOx-Kat.	-	X	X	X	X	-
5AD8	D8	NOX_NS[1]	NOx-Konzentration	-	X	X	X	X	-
5AD9	D9	LAMB_NOX_SENS[1]	Lineares Lambdasignal NOx-Sensor	-	X	X	X	X	-
5ADA	DA	VLS_NOX_SENS[1]	binäres Spannungssignal NOx-Sensor	-	X	X	X	X	-
5ADB	DB	CAN_STATE_NOX_SENS[1]	Status NOx-Sensor	-	X	X	X	X	-
5ADC	DC	CAN_ERR_NOX_SENS[1]	NOx-Sensor Error Byte	-	X	X	X	X	-
5ADD	DD			-	-	-	-	-	-
5ADE	DE			-	-	-	-	-	-
5ADF	DF	LV_CAN_TEMP_MIN_THD[1]	Taupunkterkennung für NOx-Sensor	-	X	X	X	X	-
5AE0	E0	RST_SEC	Status-byte: security info for atypical reset (reset-save memory)	X	X	X	X	X	-
5AE1	E1	RST_CTR	For ECU without generic DBG driver: Number of atypical warm-resets since last power-up (reset-save memory)	X	X	X	X	X	X
5AE2	E2	RST_TYP	reset type of last reset	X	X	X	X	X	-
		STATE_RST_TYP_ACT	reset type of last reset	-	-	-	-	-	X
5AE3	E3	RST_DBG_BACK_INFO_VLD[0]	background info for last reset valid	X	X	X	X	X	-
		LV_DBG_INFO_VLD[0]	background info for last reset valid / RESET SAFE	-	-	-	-	-	X
5AE4	E4	RST_INFO_ADD[0]	Additional reset info (cause)	X	X	X	X	X	-
		STATE_RST_INFO_ADD[0]	Additional reset info (cause) / RESET SAFE	-	-	-	-	-	X
5AE5	E5	DIST_RST_DET[0]	Mileage counter at reset	X	X	X	X	X	X
5AE6	E6	TRT_RST_DET[0]	Total runtime at reset	X	X	X	X	X	X
5AE7	E7	CPU_LOAD_MAX_RST_DET[0]	Max. CPU load from reset detection	X	X	X	X	X	X
5AE8	E8	N_CPU_LOAD_MAX_RST_DET[0]	Engine speed at max. cpu load from reset detection	X	X	X	X	X	X
5AE9	E9	RST_CLAS_SEC[0]	Security info	X	X	X	X	X	-
5AEA	EA	RST_INFO_CTR	Number of atypical warm-resets since last power-up (BSW)	X	X	X	X	X	-
		CTR_WRST	Number of atypical warm-resets since last power-up (from BSW) / RESET SAFE	-	-	-	-	-	X
5AEB	EB	Tmot_b1	Kühlmitteltemperatur < 98°C	X	X	X	X	X	-
5AEC	EC	Tmot_b2	98°C =< Kühlmitteltemperatur =< 112°C	X	X	X	X	X	-
5AED	ED	Tmot_b3	113°C =< Kühlmitteltemperatur =< 120°C	X	X	X	X	X	-
5AEE	EE	Tmot_b4	121°C =< Kühlmitteltemperatur =< 125°C	X	X	X	X	X	-
5AEF	EF	Tmot_b5	Kühlmitteltemperatur > 125°C	X	X	X	X	X	-
5AF0	F0	Toel_b1	Motoröltemperatur < 80°C	X	X	X	X	X	-
5AF1	F1	Toel_b2	80°C =< Motoröltemperatur =< 110°C	X	X	X	X	X	-
5AF2	F2	Toel_b3	110°C =< Motoröltemperatur =< 135°C	X	X	X	X	X	-
5AF3	F3	Toel_b4	135°C =< Motoröltemperatur =< 150°C	X	X	X	X	X	-
5AF4	F4	Toel_b5	Motoröltemperatur > 150°C	X	X	X	X	X	-
5AF5	F5	Tget_b1	Getriebeöltemperatur < 80°C	X	X	X	X	X	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5AF6	F6	Tget_b2	80°C =< Getriebeöltemperatur =< 109°C	X	X	X	X	X	-
5AF7	F7	Tget_b3	110°C =< Getriebeöltemperatur =< 124°C	X	X	X	X	X	-
5AF8	F8	Tget_b4	125°C =< Getriebeöltemperatur =< 129°C	X	X	X	X	X	-
5AF9	F9	Tget_b5	Getriebeöltemperatur > 129°C	X	X	X	X	X	-
5AFA	FA	Tumg_b1	Umgebungstemperatur < 3°C	X	X	X	X	X	-
5AFB	FB	Tumg_b2	3°C <= Umgebungstemperatur =< 19°C	X	X	X	X	X	-
5AFC	FC	Tumg_b3	20°C <= Umgebungstemperatur =< 29°C	X	X	X	X	X	-
5AFD	FD	Tumg_b4	30°C <= Umgebungstemperatur =< 39°C	X	X	X	X	X	-
5AFE	FE	Tumg_b5	Umgebungstemperatur > 39°C	X	X	X	X	X	-
5B MSD8x spezifisch									
5B00	00	TI_1_MES[0]	Einspritzzeit Zylinder 1 von der Endstufe rückgemessen	-	X	X	X	X	X
5B01	01	TI_1_MES[4]	Einspritzzeit Zylinder 2 von der Endstufe rückgemessen (for 4cyl and 6cyl) Einspritzzeit Zylinder 6 von der Endstufe rückgemessen (for 8cyl)	-	X	X	X	X	X
5B02	02	TI_1_MES[2]	Einspritzzeit Zylinder 3 von der Endstufe rückgemessen (for 6cyl) Einspritzzeit Zylinder 4 von der Endstufe rückgemessen (for 4cyl and 8cyl)	-	X	X	X	X	X
5B03	03	TI_1_MES[5]	Einspritzzeit Zylinder 4 von der Endstufe rückgemessen (for 4cyl and 6cyl) Einspritzzeit Zylinder 3 von der Endstufe rückgemessen (for 8cyl)	-	X	X	X	X	X
5B04	04	TI_1_MES[1]	Einspritzzeit Zylinder 5 von der Endstufe rückgemessen (for 6cyl and 8cyl) Einspritzzeit Zylinder 3 von der Endstufe rückgemessen (for 4cyl)	-	X	X	X	X	X
5B05	05	TI_1_MES[3]	Einspritzzeit Zylinder 6 von der Endstufe rückgemessen (for 6cyl) Einspritzzeit Zylinder 2 von der Endstufe rückgemessen (for 4cyl) Einspritzzeit Zylinder 8 von der Endstufe rückgemessen (for 8cyl)	-	X	X	X	X	X
5B06	06	TI_1_MES[6]	Einspritzzeit Zylinder 7 von der Endstufe rückgemessen	-	-	-	-	-	X
5B07	07	TI_1_MES[7]	Einspritzzeit Zylinder 2 von der Endstufe rückgemessen (for 8cyl)	-	-	-	-	-	X
5B08	08			-	-	-	-	-	-
5B09	09			-	-	-	-	-	-
5B0A	0A			-	-	-	-	-	-
5B0B	0B			-	-	-	-	-	-
5B0C	0C			-	-	-	-	-	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5B0D	0D			-	-	-	-	-	-
5B0E	0E			-	-	-	-	-	-
5B0F	0F			-	-	-	-	-	-
5B10	10	EGY_STEP_INJ_CHA_GRD[0]	Tastverhältnis Injektor 1 von der Endstufe rückgemessen	-	X	X	X	X	X
5B11	11	EGY_STEP_INJ_CHA_GRD[4]	Tastverhältnis Injektor 2 von der Endstufe rückgemessen (for 6cyl) Tastverhältnis Injektor 6 von der Endstufe rückgemessen (for 8cyl)	-	-	-	X	X	X
5B12	12	EGY_STEP_INJ_CHA_GRD[2]	Tastverhältnis Injektor 3 von der Endstufe rückgemessen (for 6cyl) Tastverhältnis Injektor 4 von der Endstufe rückgemessen (for 4cyl and 8cyl)	-	X	X	X	X	X
5B13	13	EGY_STEP_INJ_CHA_GRD[5]	Tastverhältnis Injektor 4 von der Endstufe rückgemessen (for 6cyl) Tastverhältnis Injektor 3 von der Endstufe rückgemessen (for 8cyl)	-	-	-	X	X	X
5B14	14	EGY_STEP_INJ_CHA_GRD[1]	Tastverhältnis Injektor 5 von der Endstufe rückgemessen (for 4cyl and 8cyl) Tastverhältnis Injektor 3 von der Endstufe rückgemessen (for 4cyl)	-	X	X	X	X	X
5B15	15	EGY_STEP_INJ_CHA_GRD[3]	Tastverhältnis Injektor 6 von der Endstufe rückgemessen (for 6cyl) Tastverhältnis Injektor 2 von der Endstufe rückgemessen (for 4cyl) Tastverhältnis Injektor 8 von der Endstufe rückgemessen (for 8cyl)	-	X	X	X	X	X
5B16	16	EGY_STEP_INJ_CHA_GRD[6]	Tastverhältnis Injektor 7 von der Endstufe rückgemessen	-	-	-	-	-	X
5B17	17	EGY_STEP_INJ_CHA_GRD[7]	Tastverhältnis Injektor 2 von der Endstufe rückgemessen (for 8cyl)	-	-	-	-	-	X
5B18	18			-	-	-	-	-	-
5B19	19			-	-	-	-	-	-
5B1A	1A			-	-	-	-	-	-
5B1B	1B			-	-	-	-	-	-
5B1C	1C			-	-	-	-	-	-
5B1D	1D			-	-	-	-	-	-
5B1E	1E			-	-	-	-	-	-
5B1F	1F			-	-	-	-	-	-
5B20	20	CHA_IV_1_MES[0]	Elektrische Ladung Injektor 1	-	X	X	X	X	X
5B21	21	CHA_IV_1_MES[4]	Elektrische Ladung Injektor 2 (for 6cyl) Elektrische Ladung Injektor 6 (for 8cyl)	-	-	-	X	X	X
5B22	22	CHA_IV_1_MES[2]	Elektrische Ladung Injektor 3 (for 6cyl) Elektrische Ladung Injektor 4 (for 4cyl and 8cyl)	-	X	X	X	X	X
5B23	23	CHA_IV_1_MES[5]	Elektrische Ladung Injektor 4 (for 6cyl) Elektrische Ladung Injektor 3 (for 8cyl)	-	-	-	X	X	X
5B24	24	CHA_IV_1_MES[1]	Elektrische Ladung Injektor 5 (for 6cyl and 8cyl) Elektrische Ladung Injektor 3 (for 4cyl)	-	X	X	X	X	X
5B25	25	CHA_IV_1_MES[3]	Elektrische Ladung Injektor 6 (for 6cyl) Elektrische Ladung Injektor 2 (for 4cyl)	-	X	X	X	X	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
			Elektrische Ladung Injektor 8 (for 8cyl)						
5B26	26	CHA_IV_1_MES[6]	Elektrische Ladung Injektor 7	-	-	-	-	-	X
5B27	27	CHA_IV_1_MES[7]	Elektrische Ladung Injektor 2 (for 8cyl)	-	-	-	-	-	X
5B28	28			-	-	-	-	-	-
5B29	29			-	-	-	-	-	-
5B2A	2A			-	-	-	-	-	-
5B2B	2B			-	-	-	-	-	-
5B2C	2C			-	-	-	-	-	-
5B2D	2D			-	-	-	-	-	-
5B2E	2E			-	-	-	-	-	-
5B2F	2F			-	-	-	-	-	-
5B30	30	V_IV_1_MES[0]	Spannung Injektor 1	-	X	X	X	X	X
5B31	31	V_IV_1_MES[4]	Spannung Injektor 2 (for 6cyl) Spannung Injektor 6 (for 8cyl)	-	-	-	X	X	X
5B32	32	V_IV_1_MES[2]	Spannung Injektor 3 (for 6cyl) Spannung Injektor 4 (for 4cyl and 8cyl)	-	X	X	X	X	X
5B33	33	V_IV_1_MES[5]	Spannung Injektor 4 (for 6cyl) Spannung Injektor 3 (for 8cyl)	-	-	-	X	X	X
5B34	34	V_IV_1_MES[1]	Spannung Injektor 5 (for 6cyl and 8cyl) Spannung Injektor 3 (for 4cyl)	-	X	X	X	X	X
5B35	35	V_IV_1_MES[3]	Spannung Injektor 6 (for 6cyl) Spannung Injektor 2 (for 4cyl) Spannung Injektor 8 (for 8cyl)	-	X	X	X	X	X
5B36	36	V_IV_1_MES[6]	Spannung Injektor 7 (for 8cyl)	-	-	-	-	-	X
5B37	37	V_IV_1_MES[7]	Spannung Injektor 2 (for 8cyl)	-	-	-	-	-	X
5B38	38			-	-	-	-	-	-
5B39	39			-	-	-	-	-	-
5B3A	3A			-	-	-	-	-	-
5B3B	3B			-	-	-	-	-	-
5B3C	3C			-	-	-	-	-	-
5B3D	3D			-	-	-	-	-	-
5B3E	3E			-	-	-	-	-	-
5B3F	3F			-	-	-	-	-	-
5B40	40	FAC_EGY_PWM_AD[0]	Adaptationswert der Endstufe 1	-	X	X	X	X	X
5B41	41	FAC_EGY_PWM_AD[4]	Adaptationswert der Endstufe 2 (for 6cyl) Adaptationswert der Endstufe 6 (for 8cyl)	-	-	-	X	X	X
5B42	42	FAC_EGY_PWM_AD[2]	Adaptationswert der Endstufe 3 (for 6cyl) Adaptationswert der Endstufe 4 (for 4cyl and 8cyl)	-	X	X	X	X	X
5B43	43	FAC_EGY_PWM_AD[5]	Adaptationswert der Endstufe 4 (for 6cyl) Adaptationswert der Endstufe 3 (for 8cyl)	-	-	-	X	X	X
5B44	44	FAC_EGY_PWM_AD[1]	Adaptationswert der Endstufe 5 von der Endstufe rückgemessen (for 4cyl and 8cyl) Adaptationswert der Endstufe 3 (for	-	X	X	X	X	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
			4cyl)						
5B45	45	FAC_EGY_PWM_AD[3]	Adaptationswert der Endstufe 6 (for 6cyl) Adaptationswert der Endstufe 2 (for 4cyl) Adaptationswert der Endstufe 8 (for 8cyl)	-	X	X	X	X	X
5B46	46	FAC_EGY_PWM_AD[6]	Adaptationswert der Endstufe 7 (for 8cyl)	-	-	-	-	-	X
5B47	47	FAC_EGY_PWM_AD[7]	Adaptationswert der Endstufe 2 (for 8cyl)	-	-	-	-	-	X
5B48	48			-	-	-	-	-	-
5B49	49			-	-	-	-	-	-
5B4A	4A			-	-	-	-	-	-
5B4B	4B			-	-	-	-	-	-
5B4C	4C			-	-	-	-	-	-
5B4D	4D			-	-	-	-	-	-
5B4E	4E			-	-	-	-	-	-
5B4F	4F			-	-	-	-	-	-
5B50	50	FAC_CYL_LAM_COR[0]	Momentan eingerechnete CILC-Werte Injektor 1	-	X	X	X	X	X
5B51	51	FAC_CYL_LAM_COR[4]	Momentan eingerechnete CILC-Werte Injektor 2 (for 6cyl) Momentan eingerechnete CILC-Werte Injektor 6 (for 8cyl)	-	-	-	X	X	X
5B52	52	FAC_CYL_LAM_COR[2]	Momentan eingerechnete CILC-Werte Injektor 3 (for 6cyl) Momentan eingerechnete CILC-Werte Injektor 4 (for 4cyl and 8cyl)	-	X	X	X	X	X
5B53	53	FAC_CYL_LAM_COR[5]	Momentan eingerechnete CILC-Werte Injektor 4 (for 6cyl) Momentan eingerechnete CILC-Werte Injektor 3 (for 8cyl)	-	-	-	X	X	X
5B54	54	FAC_CYL_LAM_COR[1]	Momentan eingerechnete CILC-Werte Injektor 5 (for 6cyl and 8cyl) Momentan eingerechnete CILC-Werte Injektor 3 (for 4cyl)	-	X	X	X	X	X
5B55	55	FAC_CYL_LAM_COR[3]	Momentan eingerechnete CILC-Werte Injektor 6 (for 6cyl) Momentan eingerechnete CILC-Werte Injektor 2 (for 4cyl) Momentan eingerechnete CILC-Werte Injektor 8 (for 8cyl)	-	X	X	X	X	X
5B56	56	FAC_CYL_LAM_COR[6]	Momentan eingerechnete CILC-Werte Injektor 7 (for 8cyl)	-	-	-	-	-	X
5B57	57	FAC_CYL_LAM_COR[7]	Momentan eingerechnete CILC-Werte Injektor 2 (for 8cyl)	-	-	-	-	-	X
5B58	58			-	-	-	-	-	-
5B59	59			-	-	-	-	-	-
5B5A	5A			-	-	-	-	-	-
5B5B	5B			-	-	-	-	-	-
5B5C	5C			-	-	-	-	-	-
5B5D	5D			-	-	-	-	-	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5B5E	5E			-	-	-	-	-	-
5B5F	5F			-	-	-	-	-	-
5B60	60	FAC_LAM_CYL_SEL_ADJ_CST[0]	CILC-Adaption kalt Injektor 1	-	X	X	X	X	X
5B61	61	FAC_LAM_CYL_SEL_ADJ_CST[4]	CILC-Adaption kalt Injektor 2 (for 6cyl) CILC-Adaption kalt Injektor 6 (for 8cyl)	-	-	-	X	X	X
5B62	62	FAC_LAM_CYL_SEL_ADJ_CST[2]	CILC-Adaption kalt Injektor 3 (for 6cyl) CILC-Adaption kalt Injektor 4 (for 4cyl and 8cyl)	-	X	X	X	X	X
5B63	63	FAC_LAM_CYL_SEL_ADJ_CST[5]	CILC-Adaption kalt Injektor 4 (for 6cyl) CILC-Adaption kalt Injektor 3 (for 8cyl)	-	-	-	X	X	X
5B64	64	FAC_LAM_CYL_SEL_ADJ_CST[1]	CILC-Adaption kalt Injektor 5 (for 6cyl and 8cyl) CILC-Adaption kalt Injektor 3 (for 4cyl)	-	X	X	X	X	X
5B65	65	FAC_LAM_CYL_SEL_ADJ_CST[3]	CILC-Adaption kalt Injektor 6 (for 6cyl) CILC-Adaption kalt Injektor 2 (for 4cyl) CILC-Adaption kalt Injektor 8 (for 8cyl)	-	X	X	X	X	X
5B66	66	FAC_LAM_CYL_SEL_ADJ_CST[6]	CILC-Adaption kalt Injektor 7 (for 8cyl)	-	-	-	-	-	X
5B67	67	FAC_LAM_CYL_SEL_ADJ_CST[7]	CILC-Adaption kalt Injektor 2 (for 8cyl)	-	-	-	-	-	X
5B68	68			-	-	-	-	-	-
5B69	69			-	-	-	-	-	-
5B6A	6A			-	-	-	-	-	-
5B6B	6B			-	-	-	-	-	-
5B6C	6C			-	-	-	-	-	-
5B6D	6D			-	-	-	-	-	-
5B6E	6E			-	-	-	-	-	-
5B6F	6F			-	-	-	-	-	-
5B70	70	MFF_ADD_AD_ER_BAL[0]	ER-Adaption TI-additiv im LL Schicht für Injektor 1	-	X	X	X	X	-
5B71	71	MFF_ADD_AD_ER_BAL[4]	ER-Adaption TI-additiv im LL Schicht für Injektor 2(for 6cyl)	-	-	-	X	X	-
5B72	72	MFF_ADD_AD_ER_BAL[2]	ER-Adaption TI-additiv im LL Schicht für Injektor 3 ER-Adaption TI-additiv im LL Schicht für Injektor 4(for 4cyl)	-	X	X	X	X	-
5B73	73	MFF_ADD_AD_ER_BAL[5]	ER-Adaption TI-additiv im LL Schicht für Injektor 4(for 6cyl)	-	-	-	X	X	-
5B74	74	MFF_ADD_AD_ER_BAL[1]	ER-Adaption TI-additiv im LL Schicht für Injektor 5 ER-Adaption TI-additiv im LL Schicht für Injektor 3 (for 4cyl N43)	-	X	X	X	X	-
5B75	75	MFF_ADD_AD_ER_BAL[3]	ER-Adaption TI-additiv im LL Schicht für Injektor 6 ER-Adaption TI-additiv im LL Schicht für Injektor 2 (for 4cyl N43)	-	X	X	X	X	-
5B76	76			-	-	-	-	-	-
5B77	77			-	-	-	-	-	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5B78	78			-	-	-	-	-	-
5B79	79			-	-	-	-	-	-
5B7A	7A			-	-	-	-	-	-
5B7B	7B			-	-	-	-	-	-
5B7C	7C			-	-	-	-	-	-
5B7D	7D			-	-	-	-	-	-
5B7E	7E			-	-	-	-	-	-
5B7F	7F			-	-	-	-	-	-
5B80	80	MFF_ADD_ER_BAL[0]	ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 1	-	X	X	X	X	X
5B81	81	MFF_ADD_ER_BAL[4]	ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 2 (for 6cyl)	-	-	-	X	X	-
5B82	82	MFF_ADD_ER_BAL[2]	ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 3(for 6cyl) ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 4 (for 4cyl)	-	X	X	X	X	-
5B83	83	MFF_ADD_ER_BAL[5]	ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 4 (for 6cyl)	-	-	-	X	X	-
5B84	84	MFF_ADD_ER_BAL[1]	ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 5 (for 6cyl) ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 3 (for 4cyl)	-	X	X	X	X	-
5B85	85	MFF_ADD_ER_BAL[3]	ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 6 (for 6cyl) ER_MFF_additiv im LL-Schicht (momentan eingerechnetr Wert) für Injektor 2 (for 4cyl)	-	X	X	X	X	-
5B86	86			-	-	-	-	-	-
5B87	87			-	-	-	-	-	-
5B88	88			-	-	-	-	-	-
5B89	89			-	-	-	-	-	-
5B8A	8A			-	-	-	-	-	-
5B8B	8B			-	-	-	-	-	-
5B8C	8C			-	-	-	-	-	-
5B8D	8D			-	-	-	-	-	-
5B8E	8E			-	-	-	-	-	-
5B8F	8F			-	-	-	-	-	-
5B90	90	FAC_TI_AD_ER_BAL[0]	ER-Adaptionsfaktor in Schicht Teillast für Injektor 1	-	X	X	X	X	-
5B91	91	FAC_TI_AD_ER_BAL[4]	ER-Adaptionsfaktor in Schicht Teillast für Injektor 2 (for 6cyl)	-	-	-	X	X	-
5B92	92	FAC_TI_AD_ER_BAL[2]	ER-Adaptionsfaktor in Schicht Teillast für Injektor 3 (for 6cyl) ER-Adaptionsfaktor in Schicht Teillast	-	X	X	X	X	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
			für Injektor 4 (for 4cyl)						
5B93	93	FAC_TI_AD_ER_BAL[5]	ER-Adaptionsfaktor in Schicht Teillast für Injektor 4(for 6cyl)	-	-	-	X	X	-
5B94	94	FAC_TI_AD_ER_BAL[1]	ER-Adaptionsfaktor in Schicht Teillast für Injektor 5 (for 6cyl) ER-Adaptionsfaktor in Schicht Teillast für Injektor 3 (for 4cyl)	-	X	X	X	X	-
5B95	95	FAC_TI_AD_ER_BAL[3]	ER-Adaptionsfaktor in Schicht Teillast für Injektor 6 (for 6cyl) ER-Adaptionsfaktor in Schicht Teillast für Injektor 2 (for 4cyl)	-	X	X	X	X	-
5B96	96			-	-	-	-	-	-
5B97	97			-	-	-	-	-	-
5B98	98			-	-	-	-	-	-
5B99	99			-	-	-	-	-	-
5B9A	9A			-	-	-	-	-	-
5B9B	9B			-	-	-	-	-	-
5B9C	9C			-	-	-	-	-	-
5B9D	9D			-	-	-	-	-	-
5B9E	9E			-	-	-	-	-	-
5B9F	9F			-	-	-	-	-	-
5BA0	A0	FAC_TI_ER_BAL[0]	ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 1	-	X	X	X	X	-
5BA1	A1	FAC_TI_ER_BAL[4]	ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 2 (for 6cyl)	-	-	-	X	X	-
5BA2	A2	FAC_TI_ER_BAL[2]	ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 3 (for 6cyl) ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 4 (for 4cyl N43)	-	X	X	X	X	-
5BA3	A3	FAC_TI_ER_BAL[5]	ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 4(for 6cyl)	-	-	-	X	X	-
5BA4	A4	FAC_TI_ER_BAL[1]	ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 5(for 6cyl) ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 3 (for 4cyl)	-	X	X	X	X	-
5BA5	A5	FAC_TI_ER_BAL[3]	ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 6 (for 6cyl) ER-Faktor in Schicht Teillast momentan eingerechnet für Injektor 2 (for 4cyl)	-	X	X	X	X	-
5BA6	A6			-	-	-	-	-	-
5BA7	A7			-	-	-	-	-	-
5BA8	A8			-	-	-	-	-	-
5BA9	A9			-	-	-	-	-	-
5BAA	AA			-	-	-	-	-	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5BAB	AB			-	-	-	-	-	-
5BAC	AC			-	-	-	-	-	-
5BAD	AD			-	-	-	-	-	-
5BAE	AE			-	-	-	-	-	-
5BAF	AF			-	-	-	-	-	-
5BB0	B0	LV_CYL_BAL_LAM_AD_EOL	Lambdaadaption am Bandende hat fertig gelernt	-	X	X	X	X	X
5BB1	B1	LV_CYL_BAL_ER_AD_ADD_EOL	ER-Balancing am Bandende hat additiv adaptiert	-	X	X	X	X	-
5BB2	B2	LV_CYL_BAL_LAM_AD_DC	Lambdaadaption ist nötig, zyklisch während Motorbetrieb zu 1 gesetzt	-	X	X	X	X	X
5BB3	B3	LV_CYL_BAL_ER_AD_FAC_EOL	ER-Balancing am Bandende hat den Faktor adaptiert	-	X	X	X	X	-
5BB4	B4	LV_CYL_BAL_AD_HOM_REQ_DC	Zylindersel. Lambdaregelung fordert homogen an, zyklisch während dem Motorbetrieb zu 1 gesetzt	-	X	X	X	X	X
5BB5	B5	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL	Zylindersel. Lambdaregelung kalt am Bandende hat fertig adaptiert	-	X	X	X	X	X
5BB6	B6	LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	Zylindersel. Lambdaregelung warm am Bandende hat fertig adaptiert	-	X	X	X	X	X
5BB7	B7	LV_CYL_BAL_LAM_SEL_AD_HOT_DC	Zylindersel. Lambdaregelung warm ist nötig, zyklisch während Motorbetrieb zu 1 gesetzt	-	X	X	X	X	X
5BB8	B8	LV_CYL_BAL_AD_WG_OPEN_REQ[1]	Zylindersel. Lambdaregelung fordert öffne WG an, zyklisch während dem Motorbetrieb zu 1 gesetzt	-	X	X	X	X	X
5BB9	B9	LV_CYL_BAL_AD_WG_OPEN_REQ[2]	Zylindersel. Lambdaregelung fordert öffnen WG2 an, zyklisch während dem Motorbetrieb zu 1 gesetzt	-	X	X	X	X	X
5BBA	BA	Rt_bastatg_h	Relative Zeit Homogen-Betrieb gesamter Motorlauf	-	X	X	X	X	-
5BBB	BB	Rt_bastatg_hs	Relative Zeit Homogen-Schicht-Betrieb gesamter Motorlauf	-	X	X	X	X	-
5BBC	BC	Rt_bastatg_s	Relative Zeit Schicht-Betrieb gesamter Motorlauf	-	X	X	X	X	-
5BBD	BD	Rt_bastatg_sa	Relative Zeit Homogen-Betrieb gesamter Motorlauf	-	X	X	X	X	-
5BBE	BE	LF_ERR_PLAUS_IV_EGY_CAL	cylinder individuell bit coded not plausible injector valve coding values "Energie" (1 - not plausible, 0 - plausible)	-	X	X	X	X	X
5BBF	BF	LF_ERR_PLAUS_IV_MFF_CAL	cylinder individuell bit coded not plausible injector valve coding values "Kleinmenge" (1 - not plausible, 0 - plausible)	-	X	X	X	X	X
5BC0	C0			-	-	-	-	-	-
5BC1	C1			-	-	-	-	-	-
5BC2	C2			-	-	-	-	-	-
5BC3	C3			-	-	-	-	-	-
5BC4	C4			-	-	-	-	-	-
5BC5	C5			-	-	-	-	-	-
5BC6	C6			-	-	-	-	-	-
5BC7	C7			-	-	-	-	-	-

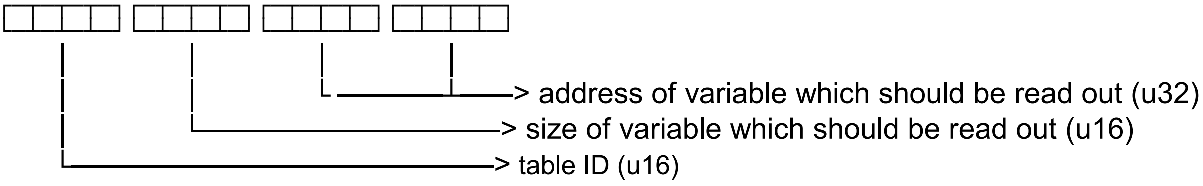
ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5BC8	C8			-	-	-	-	-	-
5BC9	C9			-	-	-	-	-	-
5BCA	CA	FAC_LAM_TCO_A[1]	Lambda-Teillastadaption Bank 1 im Kühlmitteltemperaturmesspunkt A	-	X	X	X	X	X
5BCB	CB	FAC_LAM_TCO_A[2]	Lambda-Teillastadaption Bank 2 im Kühlmitteltemperaturmesspunkt A	-	X	X	X	X	X
5BCC	CC	FAC_LAM_TCO_B[1]	Lambda-Teillastadaption Bank 1 im Kühlmitteltemperaturmesspunkt B	-	X	X	X	X	X
5BCD	CD	FAC_LAM_TCO_B[2]	Lambda-Teillastadaption Bank 2 im Kühlmitteltemperaturmesspunkt B	-	X	X	X	X	X
5BCE	CE	FAC_LAM_TCO_C[1]	Lambda-Teillastadaption Bank 1 im Kühlmitteltemperaturmesspunkt C	-	X	X	X	X	X
5BCF	CF	FAC_LAM_TCO_C[2]	Lambda-Teillastadaption Bank 2 im Kühlmitteltemperaturmesspunkt C	-	X	X	X	X	X
5BD0	D0	FAC_LAM_TCO_D[1]	Lambda-Teillastadaption Bank 1 im Kühlmitteltemperaturmesspunkt D	-	X	X	X	X	X
5BD1	D1	FAC_LAM_TCO_D[2]	Lambda-Teillastadaption Bank 2 im Kühlmitteltemperaturmesspunkt D	-	X	X	X	X	X
5BD2	D2	FAC_LAM_TCO_E[1]	Lambda-Teillastadaption Bank 1 im Kühlmitteltemperaturmesspunkt E	-	X	X	X	X	X
5BD3	D3	FAC_LAM_TCO_E[2]	Lambda-Teillastadaption Bank 2 im Kühlmitteltemperaturmesspunkt E	-	X	X	X	X	X
5BD4	D4			-	-	-	-	-	-
5BD5	D5			-	-	-	-	-	-
5BD6	D6			-	-	-	-	-	-
5BD7	D7			-	-	-	-	-	-
5BD8	D8			-	-	-	-	-	-
5BD9	D9			-	-	-	-	-	-
5BDA	DA			-	-	-	-	-	-
5BDB	DB			-	-	-	-	-	-
5BDC	DC			-	-	-	-	-	-
5BDD	DD			-	-	-	-	-	-
5BDE	DE			-	-	-	-	-	-
5BDF	DF			-	-	-	-	-	-
5BE0	E0	FAC_LAM_CYL_SEL_ADJ_H_RNG[0]	CILC-Adaptionswert warm High-Range Injektor 1	-	X	X	X	X	X
5BE1	E1	FAC_LAM_CYL_SEL_ADJ_H_RNG[4]	CILC-Adaptionswert warm High-Range Injektor 2 (for 6cyl) CILC-Adaptionswert warm High-Range Injektor 6 (for 8cyl)	-	X	X	X	X	X
5BE2	E2	FAC_LAM_CYL_SEL_ADJ_H_RNG[2]	CILC-Adaptionswert warm High-Range Injektor 3 (for 6cyl) CILC-Adaptionswert warm High-Range Injektor 4 (for 4cyl and 8cyl)	-	X	X	X	X	X
5BE3	E3	FAC_LAM_CYL_SEL_ADJ_H_RNG[5]	CILC-Adaptionswert warm High-Range Injektor 4 (for 6cyl) CILC-Adaptionswert warm High-Range Injektor 3 (for 8cyl)	-	X	X	X	X	X
5BE4	E4	FAC_LAM_CYL_SEL_ADJ_H_RNG[1]	CILC-Adaptionswert warm High-Range Injektor 5 (for 6cyl and 8cyl) CILC-Adaptionswert warm High-Range Injektor 3 (for 4cyl)	-	X	X	X	X	X
5BE5	E5	FAC_LAM_CYL_SEL_ADJ_H_RNG[3]	CILC-Adaptionswert warm High-Range	-	X	X	X	X	X

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
			Injektor 6 (for 6cyl) CILC-Adaptionswert warm High-Range Injektor 2 (for 4cyl) CILC-Adaptionswert warm High-Range Injektor 8 (for 8cyl)						
5BE6	E6	FAC_LAM_CYL_SEL_ADJ_H_RNG[6]	CILC-Adaptionswert warm High-Range Injektor 7 (for 8cyl)	-	-	-	-	-	X
5BE7	E7	FAC_LAM_CYL_SEL_ADJ_H_RNG[7]	CILC-Adaptionswert warm High-Range Injektor 2 (for 8cyl)	-	-	-	-	-	X
5BE8	E8			-	-	-	-	-	-
5BE9	E9			-	-	-	-	-	-
5BEA	EA			-	-	-	-	-	-
5BEB	EB			-	-	-	-	-	-
5BEC	EC			-	-	-	-	-	-
5BED	ED			-	-	-	-	-	-
5BEE	EE			-	-	-	-	-	-
5BEF	EF			-	-	-	-	-	-
5BF0	F0	FAC_LAM_CYL_SEL_ADJ_L_RNG[0]	CILC-Adaptionswert warm Low-Range Injektor 1	-	X	X	X	X	X
5BF1	F1	FAC_LAM_CYL_SEL_ADJ_L_RNG[4]	CILC-Adaptionswert warm Low-Range Injektor 2 (for 6cyl) CILC-Adaptionswert warm Low-Range Injektor 6 (for 8cyl)	-	-	-	X	X	X
5BF2	F2	FAC_LAM_CYL_SEL_ADJ_L_RNG[2]	CILC-Adaptionswert warm Low-Range Injektor 3 (for 6cyl) CILC-Adaptionswert warm Low-Range Injektor 4 (for 4cyl and 8cyl)	-	X	X	X	X	X
5BF3	F3	FAC_LAM_CYL_SEL_ADJ_L_RNG[5]	CILC-Adaptionswert warm Low-Range Injektor 4 (for 6cyl) CILC-Adaptionswert warm Low-Range Injektor 3 (for 8cyl)	-	-	-	X	X	X
5BF4	F4	FAC_LAM_CYL_SEL_ADJ_L_RNG[1]	CILC-Adaptionswert warm Low-Range Injektor 5 (for 6cyl and 8cyl) CILC-Adaptionswert warm Low-Range Injektor 3 (for 4cyl)	-	X	X	X	X	X
5BF5	F5	FAC_LAM_CYL_SEL_ADJ_L_RNG[3]	CILC-Adaptionswert warm Low-Range Injektor 6 (for 6cyl) CILC-Adaptionswert warm Low-Range Injektor 2 (for 4cyl) CILC-Adaptionswert warm Low-Range Injektor 8 (for 8cyl)	-	X	X	X	X	X
5BF6	F6	FAC_LAM_CYL_SEL_ADJ_L_RNG[6]	CILC-Adaptionswert warm Low-Range Injektor 7 (for 8cyl)	-	-	-	-	-	X
5BF7	F7	FAC_LAM_CYL_SEL_ADJ_L_RNG[7]	CILC-Adaptionswert warm Low-Range Injektor 2 (for 8cyl)	-	-	-	-	-	X
5BF8	F8			-	-	-	-	-	-
5BF9	F9			-	-	-	-	-	-
5BFA	FA			-	-	-	-	-	-
5BFB	FB			-	-	-	-	-	-
5BFC	FC			-	-	-	-	-	-
5BFD	FD			-	-	-	-	-	-
5BFE	FE			-	-	-	-	-	-

ID long	ID	Label	Description	MSV80	MSD80 (4cyl)	MSD81 (4cyl)	MSD80 (6cyl)	MSD81 (6cyl)	MSD85 (L4)
5BFF	FF			-	-	-	-	-	-
				-	-	-	-	-	-

7.3.3 Further manual definable DDLI- List (not for ERRM)

In the dataset one more DDLI- list is implemented. In this list further variables for DDLI- read can be inserted without a software change. The insertion must be done with help of a hex editior. It is necessary to get the start address of the table DDLI_table_cal from the map- file, open the dataset (*.s19- file) and edit it from the start address on. In the hex- file is always the same character string behind the table identifier. It contains the following information:



On third position (byte 5 - 8) you have to insert the address of the wanted variable. The address can be taken from the map- file of the corresponding software and must be changed at every software change, too. On second position the correct size of the variable has to be inserted (e.g. 8 for a byte- variable).

5Fxx	DDLI- List in dataset for auxiliary variables		
5FE0	E0	nn	nn
5FE1	E1	nn	nn
5FE2	E2	nn	nn
5FE3	E3	nn	nn
5FE4	E4	nn	nn
5FE5	E5	nn	nn
5FE6	E6	nn	nn
5FE7	E7	nn	nn
5FE8	E8	nn	nn
5FE9	E9	nn	nn
5FEA	EA	nn	nn
5FEB	EB	nn	nn
5FEC	EC	nn	nn
5FED	ED	nn	nn
5FEE	EE	nn	nn
5FEF	EF	nn	nn
5FF0	F0	nn	nn
5FF1	F1	nn	nn
5FF2	F2	nn	nn
5FF3	F3	nn	nn
5FF4	F4	nn	nn
5FF5	F5	nn	nn

5FF6	F6	nn	nn
5FF7	F7	nn	nn
5FF8	F8	nn	nn
5FF9	F9	nn	nn
5FFA	FA	nn	nn
5FFB	FB	nn	nn
5FFC	FC	nn	nn
5FFD	FD	nn	nn
5FFE	FE	nn	nn
5FFF	FF	nn	nn

7.4 Message structure

7.4.1 Request Message with Conditional Target, Source Address and length information

The following table shows a general service table structure and its syntax.

Data Byte	*Byte #*	*Parameter Name*	*CVT*	*Value*	*Mnemonic*
	#1d	**Format byte**	**M**	**#AALLLLLLb**	**FMT**
	#2d	**Target address byte**	**C**	**#12h/#EAh/#EDh/ #EFh/ #33h**	**TGT**
	#3d	**Source address byte**	**C**	**#F1h**	**SRC**
	#4d	**Additional length byte**	**C**	**#01h-#FFh**	**LEN**
#1d	#5d	Service ID Byte	M		SID
•	#6d	Data Byte #02h	C	#XXh	DATA
•	•	•	•	•	
•	•	•	•	•	
•	•	•	•	•	
#255d	#259d	Data Byte #255d	C	#XXh	
	#260d	Checksum	M	#00h-#FFh	CS

❑ The input/output buffer of *BSW-KWP* is limited to 255 (#FFH) ***Bytes*** (SID included).

❑ In the following description the table begins with the first data byte (see shaded part of the table above).The headers of the message-flow are always the same construction and therefore in all following diagnostic service descriptions unnecessary.

Header				*Data bytes*					*Checksum*
Fmt	**Tgt**	**Src**	**Len**	**SID**	**Data**	●	●	●	**CS**
max 4 byte				max. 255 byte					1 byte

7.4.2 Method of addressing

❑ The BSW-KWP supports the physical Addressing with the Keybytes #8FEFh (startCommunicationPositive Response; Keybyte1 = #EFh, Keybyte2 = #8Fh), header with conditional target and source address information, with length information in the formatbyte or additional length byte, depending on the response length.

❑ The ECU-address for **physical** adressing is #12h.

❑ The ECU-address for **functional** adressing is #EFh.

❑ The ECU-address for **functional** adressing powertrain CAN is #EAh actual not supported

❑ The ECU-address for **functional** adressing BOS-CAN is #EDh actual not supported

❑ The ECU-address for **OBD services** is #33h (functional).

7.4.3 Length byte

❑ The ECU sends the response with an additional length byte, if the lenght is longer than 64 byte.

❑ The ECU accepts the length information of the request in an additional length byte or in the formatbyte.

7.4.4 Header with or without Length ,Target and Source Address Information

The following two different types of headers are supported the *BSW-KWP* depending of the two high bits in the Format Byte:

				⇦ *Length* ⇨				
Fmt	Tgt	Src	**Len**	SId	Data			CS
⇦ used for *Checksum* calculation ⇨								

Header with address information, with additional length byte

Possible Format Bytes:

Physical Addressing	A1=1	A0=0	L5=X	L4=X	L3=X	L2=X	L1=X	L0=X
Functional Addressing	A1=1	A0=1	L5=X	L4=X	L3=X	L2=X	L1=X	L0=X

		⇦ *Length* ⇨				
Fmt	**Len**	SId	Data			CS
⇦ used for *Checksum* calculation ⇨						

Header without address information, with optional additional length byte

Possible Format Byte:

Physical Addressing without Address Information	A1=0	A0=0	L5=X	L4=X	L3=X	L2=X	L1=X	L0=X

7.4.5 Checksum byte

☐ The checksum is the sum of all bytes in the message (except the checksumbyte itself) modulo 256 (i.e. 1 byte long):

(Checksum = Format + Target + Source + Length + ΣData[SID + DATA1 + … + DATAn])

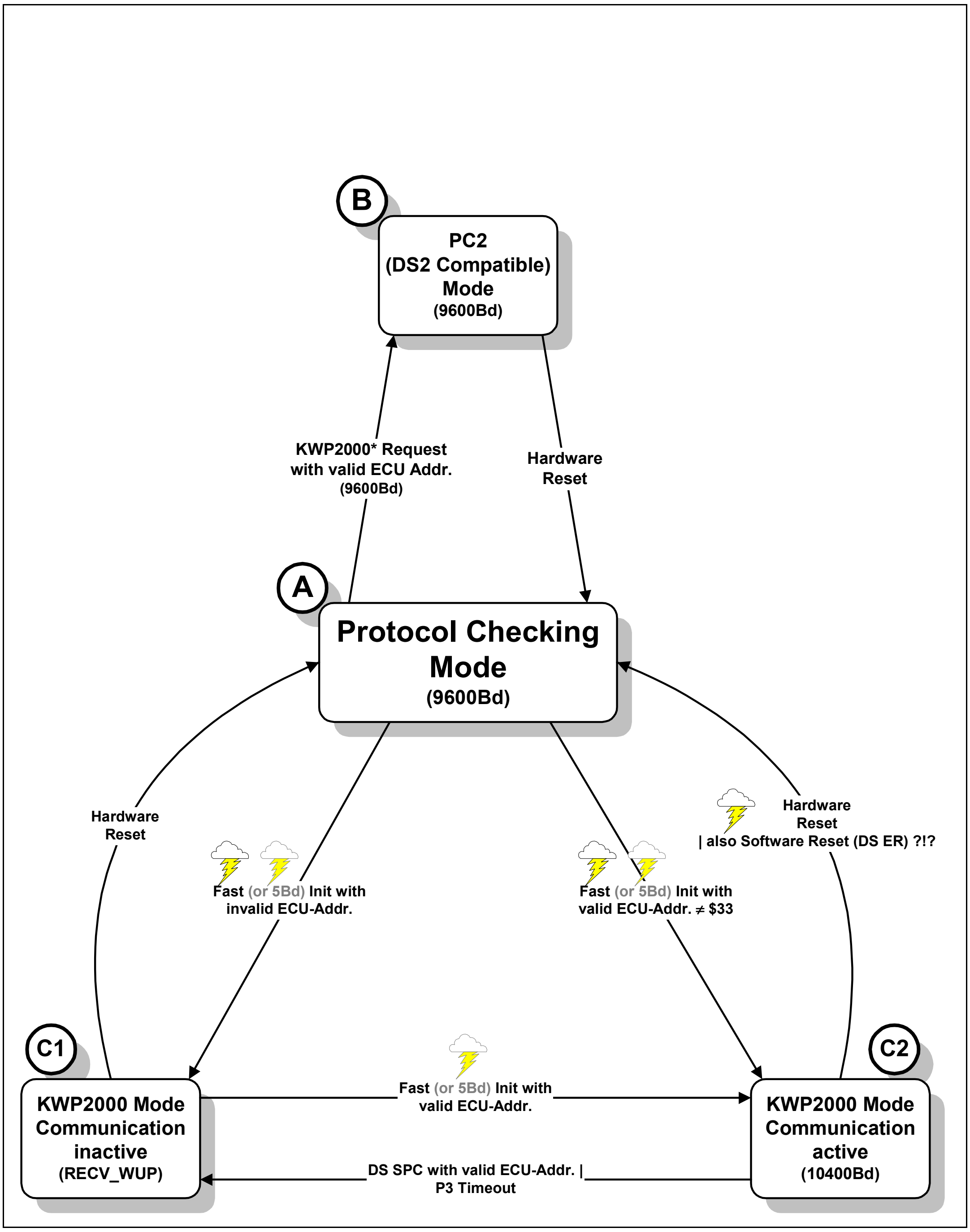
or (Checksum = Format + Target + Source + ΣData[SID + DATA1 + … + DATAn])

or (Checksum = Format + Length + ΣData[SID + DATA1 + … + DATAn])

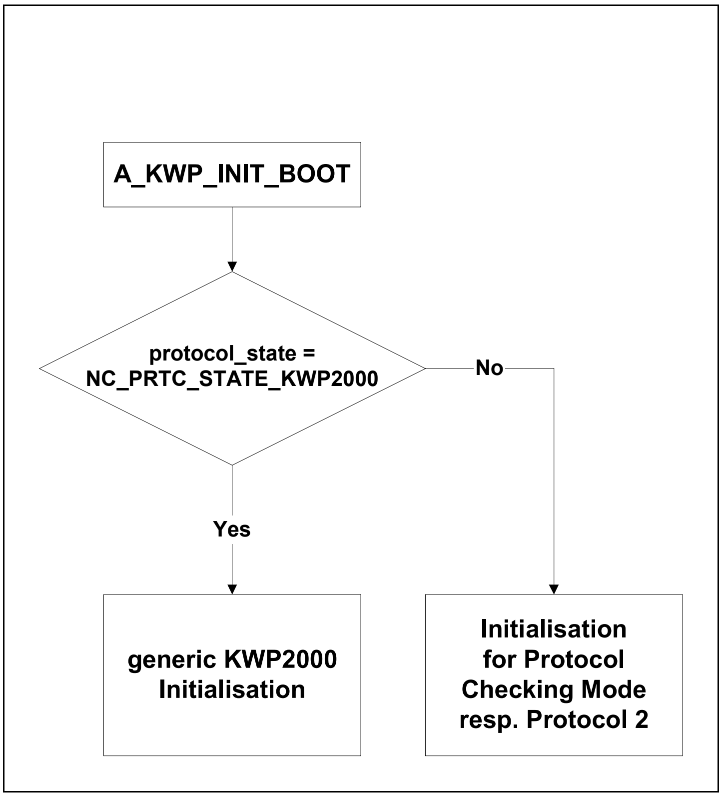
or (Checksum = Format + ΣData[SID + DATA1 + … + DATAn])

7.5 Initialisation and Timing

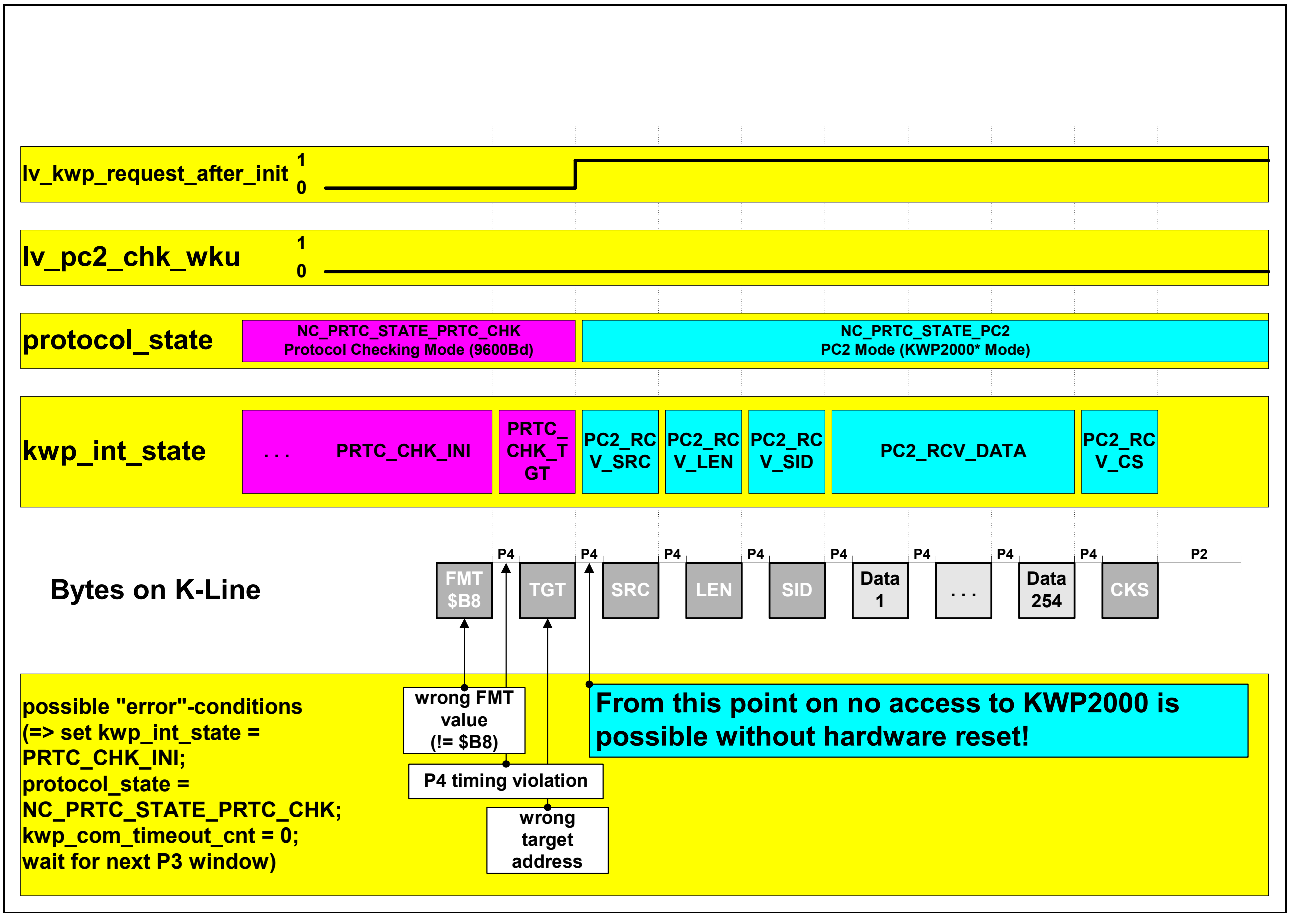
7.5.1 Protocol Switching



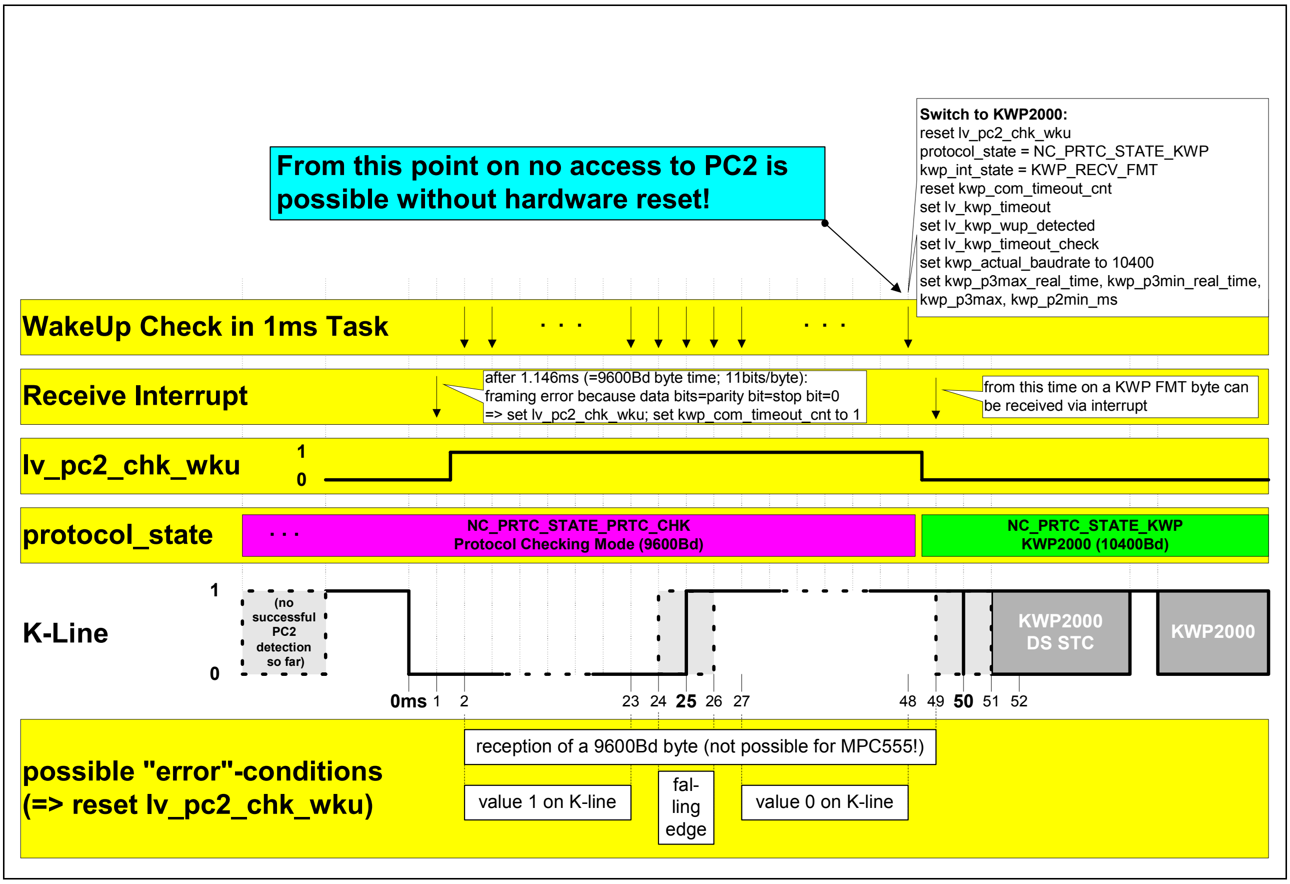
7.5.2 Initialisation of different protocols



7.5.3 KWP* Detection



7.5.4 WUP Detection for KWP



Switch to KWP2000:
reset lv_pc2_chk_wku
protocol_state = NC_PRTC_STATE_KWP
kwp_int_state = KWP_RECV_FMT
reset kwp_com_timeout_cnt
set lv_kwp_timeout
set lv_kwp_wup_detected
set lv_kwp_timeout_check
set kwp_actual_baudrate to 10400
set kwp_p3max_real_time, kwp_p3min_real_time, kwp_p3max, kwp_p2min_ms

after 1.146ms (=9600Bd byte time; 11bits/byte):
framing error because data bits=parity bit=stop bit=0
=> set lv_pc2_chk_wku; set kwp_com_timeout_cnt to 1

from this time on a KWP FMT byte can be received via interrupt

Table Fast Initialization

- The tester may use a baudrate of 10400 baud for initializatian (stc) and communication. For KWP2000* the baudrate is 9600 baud.
- It is possible to change the baudrate with the service **s**tart**D**iagnostic**S**ession.

7.5.5 Timing

Timing	*lower limit values / ms*			*upper limit values / ms*		
Parameter	***min.***	***default***	***Resolution***	***default***	***max.***	***Resolution***
P1	0	0	-	20	20	-
P2	0	25	0,5	50 200ms for KWP*	6375	#25d; #FFh=6375ms
P3	0	55 25ms for KWP*	0.5	5000 infinity for KWP*	63750	#250d; #FFh=63750ms
P4	0	5	0.5	20	20	-
T_{WuP}	49	50	-	-	51	-
T_{inL}	24	25	-	-	26	-
T_{Idle}	-	300	-	-	-	-

Table - Normal Timing Parameter Set (for functional and physical addressing)

Value	*Description*
P1	Inter byte time for ECU response
P2	Time between tester request and ECU response
P3	Time between end of ECU responses and start of new tester request
P4	Inter byte time for tester request
TWuP	High Low sequenz of the Wake-Up-Pattern
TiniL	Low sequenz of the Wake-Up-Pattern
Tidle (W5)	Idle Time

7.6 Error handling

7.6.1 Error handling during physical/functional Fast Initialization

7.6.1.1 Client(tester) Error handling during physical/functional Fast Initialization

Client(tester) detects an...	***Action***
... error in T_{idle} (W5 or P3min)	The client(tester) is responsible to keep to the idle time The server(ECU) is responsible to keep to the time P1min. In case of an error the client(tester) must wait for T_{idle} again.
... error in P1min	No observation necessary. P1min is always 0 ms.
... error in P1max (P1max timeout)	The client(tester) shall ignore the response and shall open a new timing window P2 to receive a directly repeated response from the same server(ECU) or a response from another server(ECU). If the server(ECU) doesn´t repeat the response, the client(tester) shall wait for P3max timeout and afterwards the client(tester) may start a new Initialization beginning with a wake-up-pattern (T_{idle}= 0 ms).
... error in P2min	No observation necessary. P2min is always 0 ms during Initialization.
... error in P2max (no valid response from any server (ECU))	If the client(tester) doesn´t receive any response, the client(tester) shall wait for P3max timeout and afterwards may start a new Initialization beginning with a wake-up-pattern (T_{idle}= 0 ms).
... error in StartCommunication Positive Response (Byte collision) (Response contents) (Response checksum)	The client(tester) shall ignore the response and shall open a new timing window P2 to receive a directly repeated response from the same server(ECU) or a response from another server(ECU). If the server(ECU) doesn´t repeat the response, the client(tester) shall wait for P3max timeout and afterwards the client(tester) may start a new Initialization beginning with a wake-up-pattern (T_{idle}= 0 ms).

Client(tester) Error Handling during physical/functional Fast Initialization

7.6.1.2 Server(ECU) Error Handling during physical/functional Fast Initialization (normal timing only)

Server(ECU) detects an ...	***Action***
... error in T_{idle} (W5 or P3min)	No observation necessary. The client(tester) is responsible to keep to the idle time T_{idle}.
... error in wake-up-pattern	The server(ECU) shall not respond and shall be able to detect immediately a new wake-up-pattern sequence.
... error in P4min	No observation necessary. The client(tester) is responsible to keep to the time P4min
... error in P4max (P4max timeout)	The server(ECU) shall not respond and shall be able to detect immediately a new wake-up-pattern sequence.
... error in StartCommunication Request (checksum) (contents)	The server(ECU) shall not respond and shall be able to detect immediately a new wake-up-pattern sequence.
... error in StartCommunication Positive Response (Byte collision)	The server (ECU) shall repeat the response within a new timing window P2 considering arbitration.
... unknown tester(client) target address or server(ECU) source address	The server(ECU) shall not respond and shall be able to detect immediately a new wake-up-pattern sequence.

Server(ECU) Error Handling during physical/functional Initialization

7.6.2 Error handling after physical/functional Initialization

7.6.2.1 Client(tester) Error handling after physical/functional Initialization

Client(tester) detects an ...	*Action*
... error in P1min	No observation necessary P1min is always 0 ms.
... error in P1max (P1max timeout)	The client(tester) shall ignore the response and shall open a new timing window P2 to receive a directly repeated response from the same server(ECU) or a response from another server(ECU). If the server(ECU) doesn´t repeat the response, the client(tester) may repeat the same request in a new timing window P3.
... error in P2min	No observation necessary. The server(ECU) is responsible to keep to the time P2min.
... error in P2max (no valid response from any server (ECU) or missing responses)	The client(tester) may repeat the last request twice, each within in a new timing window P3. (i.e. three transmission total) any following appropriate action is client(tester) dependent if the client(tester) doesn´t receive a response.
... error in server(ECU) Response (checksum) (contents)	The client(tester) may repeat the last request twice, each within in a new timing window P3. (i.e. three transmission total) any following appropriate action is client(tester) dependent if the client(tester) doesn´t receive a response.

Client(tester) communication Error Handling after physical/functional Initialization

7.6.2.2 Server(ECU) Error Handling after physical/functional Initialization

Server(ECU) detects an ...	*Action*
... error in P3min	No observation necessary. The client(tester) is responsible to keep to the time P3min.
... error in P3max (P3max timeout)	The server(ECU) shall reset communication and shall be able to detect immediately a new wake-up-pattern sequence.
... error in P4min	No observation necessary. The client(tester) is responsible to keep to the time P4min.
... error in P4max (P4max timeout)	The server(ECU) shall ignore the request and shall open a new timing window P3 to receive a new request from the client(tester).
... error in client(tester) request (contents) (checksum)	The server(ECU) shall ignore the request and shall open a new timing window P3 to receive a new request from the client(tester).
... unknown source or target address	The server(ECU) shall ignore the request and shall open a new timing window P3 to receive a new request from the client(tester).
... error in its own Response (byte collision)	If the server(ECU) detects a byte collision within its own response, it must repeat the response within a new timing window P2 considering the arbitration.

Server(ECU) Error Handling after physical/functional Initialization

7.7 Error-Codes

Hex Value	***Response Code***	***Mnemonic***
10	generalReject	GR
11	serviceNotSupported	SNS
12	subFunctionNotSupported-invalidFormat	SFNS_IF
21	busy-repeatRequest	B_RR
22	conditionsNotCorrect /requestSequenceError	CNCORSE
23	routineNotComplete	RNC
31	requestOutOfRange	ROOR
33	securityAccessDenied/Required	SAD
35	invalidKey	IK
36	exceedNumberOfAttempts	ENOA
37	requiredTimeDelayNotExpired	RTDNE
40	downloadNotAccepted	DNA
41	improperDownloadType	IDT
42	can'tDownloadToSpecifiedAddress	CNDTSA
43	can'tDownloadNumberOfBytesRequested	CNDNOBR
50	uploadNotAccepted	UNA
51	improperUploadType	IUT
52	can'tUploadFromSpecifiedAddress	CNUFSA
53	can'tUploadNumberOfBytesRequested	CNUNOBR
71	transferSuspended	TS
72	transferAborted	TA
74	illegalAddressInBlockTransfer	IAIB
75	illegalByteCountInBlockTransfer	IBCIBT
76	illegalBlockTransferType	IBTT
78	requestCorrectlyReceived-ResponsePending	RCRRP
79	incorrectByteCountDuringBlockTransfer	IBCDBT
80	serviceNotSupportedInActiveDiagnosticMode	SNSIADM

7.7.1 General negative Responses

NegativeResponseCode	***Cause of Occurrence***
generalReject	selected diagnosticService is longer than the current receive buffer
serviceNotSupported	selected diagnosticService is not supported by the ECU
serviceNotSupportedInActiveDiagnosticMode	selected diagnosticService is not supported in the current diagnostic mode
securityAccessDenied/Required	selected diagnosticService is not supported in the current securityLevel
incorrectByteCountDuringBlockTransfer	Number of bytes of the telegram does not match to the given length
requestCorrectlyReceived-ResponsePending	the action to be performed by the selected diagnosticService may not be completed yet

- ☐ These negative response codes are possible for all diagnostic services. The occurrence is depending on the system configuration.

7.8 Implemented Diagnostic services

Verwendete Abkürzungen:

CDI	ClearDiagnosticInformation	**RRRBLI**	RequestRoutineResultsByLocalIdentifier
DDLI	DynamicallyDefineLocalIdentifier	**STDS**	StartDiagnosticSession
DNMT	DisableNormalMessageTransmission	**STRBLI**	StartRoutineByLocalIdentifier
IOCBLI	InputOutputControlByLocalIdentifier	**SPRBLI**	StopRoutineByLocalIdentifier
RDBCI	ReadDataByCommonIdentifier	**WDBCI**	WriteDataByCommonIdentifier
RDBLI	ReadDataByLocalIdentifier		
REI	ReadEcuIdentification		

7.8.1 Overview for SID 10h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming	Mode 86 development	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
10_81	Wechsel Diagnosemode Mode 81	STDS	ENA	DI	L5	DI	BOOT	X	X	X	X	X	X
10_82	Wechsel Diagnosemode Mode 82	STDS	ENA	DI	L5	DI	BOOT	X	X	X	X	X	X
10_85	Wechsel Diagnosemode Mode 85	STDS	DI	L3	L5	DI	BOOT	X	X	X	X	X	X
10_86	Wechsel Diagnosemode Mode 86	STDS	DI	DI	L5	DI	BOOT	X	X	X	X	X	X
10_89	Wechsel Diagnosemode Mode 89	STDS	DI	DI	L5	L4	BOOT	X	X	X	X	X	X

7.8.2 Overview for SID 11h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming	Mode 86 development	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)

11_01	EcuReset	EcuReset	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X

7.8.3 Overview for SID 14h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
14_FF_FF	Fehlerspeicher löschen alle Gruppen	CDI	ENA	DI	L5	DI	ECU	X	X	X	X	X	X
14_FF_FB	Fehlerspeicher löschen Powertrain	CDI	ENA	DI	L5	DI	ECU	X	X	X	X	X	X

7.8.4 Overview for SID 17h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
17	Fsp. lesen, lang	readStatusOfDTC	ENA	DI	L5	DI	ECU	X	X	X	X	X	X

7.8.5 Overview for SID 18h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
18	Fehlerspeicher lesen, kurz	read DTCBy Status	ENA	DI	L5	DI	ECU	X	X	X	X	X	X

7.8.6 Overview for SID 1Ah

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
1A_80	Identifikation lesen ECUIDT	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_86	Identifikation lesen CUIFDI	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_87	Identifikation lesen PECUHN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_89	Identifikation lesen SSECUSEN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_8A	Identifikation lesen CRSCOTSN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_8B	Identifikation lesen CPD	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_8C	Identifikation lesen SSBSN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_8D	Identifikation lesen SSESN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_8E	Identifikation lesen SSCN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_8F	Identifikation lesen LDSA	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_90	Identifikation lesen VIN short	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_91	Identifikation lesen VMECUHN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_92	SiemensBootLogisticField lesen	REI	ENA	L3	L5	DI	BOOT	-	-	-	X	X	-
1A_96	Identifikation lesen EROTAN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_97	Identifikation lesen SNOET	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_98	Identifikation lesen RSCOTSN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_99	Identifikation lesen PD	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_9A	Identifikation lesen VMECUHVN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_9B	Identifikation lesen VMCI	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_9C	Identifikation lesen VMDI	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_9D	Identifikation lesen DOECUM	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X
1A_9E	Identifikation lesen SSI	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
1A_9F	Identifikation lesen VMECUSLVN	REI	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X

7.8.7 Overview for SID 20h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
20	Diagnosemode beenden	StopDiagnosticSession	ENA	L3	L5	L4	BOOT	X	X	X	X	X	X

7.8.8 Overview for SID 21h

Hint: overview for EOL-services taken from 17I03901.00O

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
21_01	read system specific adress	RDBLI	ENA	DI	L5	DI	BOOT	X	-	-	-	-	-
21_02	read overtemperature protection	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_03	read engine overspeed detection	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_05	read readiness flags	RDBLI	ENA	DI	L5	DI	ECU	X	X	X	X	X	X
21_06	receiving status of	RDB	EN	D	L	D	ECU	-	-	-	-	-	-

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
	immobilizer	LI	A	I	5	I							
21_07	read cruise control status	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_08	release of multiple start value programming	RDB LI	EN A	D I	L 5	D I	ECU	-	-	-	-	-	-
21_09	read DIT_ACT_MIL	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_0A	read Error Memory with freeze frame data extra long	RDB LI	EN A	D I	L 5	D I	ECU	X	X	X	X	X	X
21_20	Read Status EOL: Secondary air system	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_22	Read Status EOL: CPS	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_25	read INH_IV_KWP	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_26	Read Status EOL: Idle speed setpoint displace	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_27	Read Status EOL: VVL adaption	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_36	read air-conditioner status	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_3F	read idle speed status	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_41	read LSH 4 status	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_42	read LSH 3 status	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_43	read LSH 2 status	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_44	read LSH 1 status	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_49	read Immobilizer status	RDB LI	EN A	D I	L 5	D I	ECU	-	-	-	-	-	-
21_4C	read basic injetion time cylinder 1	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_50	read engine speed	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_54	read intake air temperature	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-
21_55	read coolant temperature	RDB LI	EN A	D I	L 5	D I	ECU	X	-	-	-	-	-

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
21_56	read ignition angle cylinder 1	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_57	read opening angle of the throttle valve (TPS_AV)	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_58	read mass air flow per segment measured (MAF_KGH_MES)	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_5B	read Pedal value sensor 1 raw acquisition (V_PVS_1_KWP)	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_71	read LV_KNK	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_A3	read VB	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_C0	read SAFM status	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_C1	read kva-factor	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_C3	read TRT	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_D5	Read Status EOL: ECRAS system test	RDBLI	ENA	DI	L5	DI	ECU	X	X	X	X	X	X
21_DA	Read Status EOL: DMTL	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_DF	Read Status EOL: Lambda sensors confused / wrong wired	RDBLI	ENA	DI	L5	DI	ECU	X	-	-	-	-	-
21_F0	DDLI handling	RDBLI	ENA	DI	L5	DI	ECU	X	X	X	X	X	X

7.8.9 Overview for SID 22h

Hint: overview taken from 17I00V0L.00R

Overview for SID number	service	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
22_10_00	Read check stamp	ECU	X	X	X	X	X	X
22_10_01	Read value BOS	ECU	X	X	X	X	X	X

Overview for SID number	service	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
22_10_0A	Read Energy saving status	ECU	X	X	X	X	X	X
22_16_00	Read Anzahl Subbusteilnehmer	ECU	X	X	X	X	X	X
22_16_01	Read Identifikation Subbusteilnehmer 1 (normaly: Batteriesensensor)	ECU	X	X	X	X	X	X
22_16_02	Read Identifikation Subbusteilnehmer 2 (normaly:elktr. Wasserpumpe)	ECU	X	X	X	X	X	X
22_16_03	Read Identifikation Subbusteilnehmer 3 (normaly:Generator 1)	ECU	X	X	X	X	X	X
22_20_00	read failure memory short	ECU	X	X	X	X	X	X
22_21_00	Read History Memory	ECU	X	X	X	X	X	X
22_25_00	Programm Reference Backup	BOOT	X	X	X	X	X	X
22_25_01	Erasing time	BOOT	X	X	X	X	X	X
22_25_02	Hardware reference	BOOT	X	X	X	X	X	X
22_25_03	Programm reference	BOOT	X	X	X	X	X	X
22_25_04	Data reference	BOOT	X	X	X	X	X	X
22_25_06	Maximum of block length	Boot-/ECU	X	X	X	X	X	X
22_30_00	MIL Codierung lesen	ECU	X	X	X	X	X	X
22_30_01	Kat Codierung lesen	ECU	X	X	X	X	X	X
22_30_10	VMAX-Codierdaten lesen	ECU	X	X	X	X	X	X
22_30_20	OL/UL Codierung lesen	ECU	X	X	X	X	X	X
22_30_30	Status-Codierung-Protokoll lesen	ECU	X	X	X	X	X	X
22_32_00	Oelwechselintervall- Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_10	IGR-Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_11	Xenonverbau-Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_20	SPA-Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_30	BZE-Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_40	Abgasklappe Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_50	Codierung MSA lesen	ECU	-	X	-	X	-	-
22_32_60	Read CDASMOT	ECU	X	X	X	X	X	X
22_3F_FF	Aenderungsindex Codierdaten lesen	ECU	X	X	X	X	X	X
22_40_00	Read values 4000	ECU	X	X	X	X	X	X
22_40_01	Read values “batterie generator”	ECU	X	X	X	X	X	X
22_40_02	Read Switch Position	ECU	X	X	X	X	X	X
22_40_03	Read values of engine roughness	ECU	X	X	X	X	X	X
22_40_06	Read CAMSHAFT-Sensor Adaption	ECU	X	X	X	X	X	X
22_40_07	Read State of Functions	ECU	X	X	X	X	X	X
22_40_08	Read Throttle and MAF-Adjustment Value	ECU	X	X	X	X	X	X
22_40_0A	Read mixture values	ECU	X	X	X	X	X	X

Overview for SID number	service	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
22_40_0B	VVT Messwerte auslesen	ECU	X	-	-	-	-	-
22_40_15	Read FASTA_Messwertblock 10	ECU	X	X	X	X	X	X
22_40_16	READ IGR-values	ECU	X	X	X	X	X	X
22_40_17	READ LEM-values	ECU	X	X	X	X	X	X
22_40_18	READ MSA-values	ECU	X	X	X	X	X	X
22_40_19	Fsp. Lesen mit Freeze Frame	ECU	X	X	X	X	X	X
22_40_1A	READ BZE-values	ECU	X	X	X	X	X	X
22_40_1B	Read: Results of intelligent alternator diagnosis	ECU	X	X	X	X	X	X
22_40_1C	Ringspeicher MSA lesen	ECU	X	X	X	X	X	X
22_40_1D	Read Verbredinfo	ECU	-	-	-	-	-	X
22_40_1E	Read CBS-Zündkerzen	ECU	-	X	X	X	X	-
22_40_1F	Read BMW-PST (BMW-Programmstandsinformation)	ECU	-	-	-	-	-	X
22_40_21	READ IBS-values	ECU	X	X	X	X	X	X
22_40_22	READ PM1 Status	ECU	X	X	X	X	X	X
22_40_23	READ PM2 Status	ECU	X	X	X	X	X	X
22_40_26	Rate Based Monitoring Mode 9 auslesen	ECU	X	X	X	X	X	X
22_40_27	Rate Based Monitoring Motorsteuerung MSV80 Block 1 auslesen	ECU	X	-	-	-	-	-
22_40_27	Rate Based Monitoring Motorsteuerung MSD8x Block 1 auslesen	ECU	-	X	X	X	X	X
22_40_28	Rate Based Monitoring Motorsteuerung MSV80 Block 2 auslesen	ECU	X	-	-	-	-	-
22_40_28	Rate Based Monitoring Motorsteuerung MSD8x Block 2 auslesen	ECU	-	X	X	X	X	X
22_40_2D	Messwerte Laufruhepruefung auslesen	ECU	X	X	X	X	X	X
22_40_2E	Nullgang-Erkennung lesen	ECU	X	X	X	X	X	X
22_40_2F	MSA lesen	ECU	X	X	X	X	X	X
22_40_30	FASTA_LVS_lesen	ECU	-	X	X	X	X	-
22_40_31	Laufruhe Verbesserungssystem Zylinderstatistik auslesen	ECU	-	X	X	X	X	-
22_40_32	MFMA (Kleinstmengenadaption)	ECU	-	X	X	X	X	X
22_5F_88	Adrecovery lesen	ECU	-	X	X	X	X	-
22_5F_8B	read PM recovery	ECU	X	X	X	X	X	X
22_5F_8E	MSA deactivation lesen	ECU	X	X	X	X	X	X
22_5F_8F	MSA Abschaltverhinderer lesen	ECU	X	X	X	X	X	X
22_5F_90	IMAALLE lesen	ECU	-	X	X	X	X	X
22_5F_DE	VVT- Minhub lesen	ECU	X	-	-	-	-	-
22_5F_F0	Abgleichswert LL lesen	ECU	X	X	X	X	X	X

Overview for SID number	service	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
22_5F_F1	CO Abgleichswert lesen	ECU	X	-	-	-	-	-
22_5F_F2	Varianten lesen	ECU	X	X	X	X	X	X
22_5F_F3	Uebertemperatursicherung lesen	ECU	X	X	X	X	X	X
22_5F_F4	Fusshebelwerk Fehlbedienung lesen	ECU	X	X	X	X	X	-
22_5F_F6	Ueberdrehzahlsicherung lesen	ECU	X	X	X	X	X	X
22_5F_F7	Laufruhepruefung auslesen	ECU	X	X	X	X	X	-
22_5F_F8	Status Homogenbetrieb auslesen	ECU	-	X	X	X	X	-
22_C0_00	Read imob status (STATUS_EWS)	Boot-/ECU	X	X	X	X	X	X
22_C0_02	Read imob k_ews(tmp) (STATUS_EWS_SK)	Boot-/ECU	X	X	X	X	X	X

7.8.10 Overview for SID 23h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
23	Speicher lesen	ReadMemoryByAddress	ENA	L3	L5	DI	BOOT	X	X	X	X	X	X

7.8.11 Overview for SID 28h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
28_01	Enable positive or negative respose	DNMT	ENA	DI	L5	DI	Boot-/ECU	X	X	X	X	X	X
28_02	Disable positive response	DNMT	ENA	DI	L5	DI	Boot-/EC	X	X	X	X	X	X

							U						

7.8.12 Overview for SID 29h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
29	Enable positive respose	DNMT	ENA	DI	L5	DI	Boot-/ECU	X	X	X	X	X	X
29_01	Enable positive respose	DNMT	ENA	DI	L5	DI	Boot-/ECU	X	X	X	X	X	X
29_02	Disable positive response	DNMT	ENA	DI	L5	DI	Boot-/ECU	X	X	X	X	X	X

7.8.13 Overview for SID 2Ch

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)
2C_F0_02	DefineByCommonIdentifier	DDLI	ENA	DI	L5	DI	ECU	X	X	X	X
2C_F0_03	DefineByMemoryAddress	DDLI	ENA	DI	L5	DI	ECU	X	X	X	X
2C_F0_04	ClearDynamicallyDefinedLocalIdentifier	DDLI	ENA	DI	L5	DI	ECU	X	X	X	X

7.8.14 Overview for SID 2Eh

Hint: overview taken from 17I00W03.0AA

SID number	service	MSV_80	MSD_80 (6cyl)	MSD_80 (4cyl)	MSD_81 (6cyl)	MSD_81 (4cyl)	MSD85(L4)
2E_10_00	Prüfstempel schreiben	X	-	-	-	-	-
2E_10_01	BOS löschen via Tester	X	X	X	X	X	X
2E_30_00	MIL-OFF Codierung schreiben	X	X	X	X	X	X
2E_30_01	Kat Codierung schreiben	X	X	X	X	X	X
2E_30_10	VMAX Codierung schreiben	X	X	X	X	X	X
2E_30_20	OL/UL Codierung schreiben	X	X	X	X	X	X
2E_30_30	Codierung-Protokoll schreiben	X	X	X	X	X	X
2E_32_00	Oelwechselintervall- Codierdaten schreiben	X	X	X	X	X	X
2E_32_10	IGR-Codierdaten schreiben	X	X	X	X	X	X
2E_32_11	Xenonverbau - Codierdaten schreiben	X	X	X	X	X	X
2E_32_20	Codierung SPA schreiben	X	X	X	X	X	X
2E_32_30	Codierung BZE schreiben	X	X	X	X	X	X
2E_32_40	Abgasklappe Codierung schreiben	X	X	X	X	X	X
2E_32_50	Codierung MSA schreiben	-	-	X	-	X	-
2E_32_60	Codierung CDASMOTschreiben	X	X	X	X	X	X
2E_3F_FF	Aenderungsindex der Codierdaten schreiben	X	X	X	X	X	X
2E_5F_87	LaufruheVerbesserungssystem Zaehler Reset	-	X	X	X	X	-
2E_5F_88	Schreiben NOx-Adaptationswerte	-	X	X	X	X	-
2E_5F_89	Request Reset MSA-Tabellenspeicher	-	-	X	-	X	-
2E_5F_8A	Schreiben Nullgang-Lernwert	-	-	X	-	X	-
2E_5F_8B	PMB vorgeben	X	X	X	X	X	X
2E_5F_8E	MSA_DEAK schreiben	-	-	X	-	X	-
2E_5F_8F	MSA_DEAK_AV schreiben	-	-	X	-	X	-
2E_5F_90	IMAALLE schreiben	-	X	X	X	X	X
2E_5F_9x	IMA für Zylinder x schreiben	-	X	X	X	X	X
2E_5F_DE_00	VVT- Minhub Ansteuerung beenden	X	-	-	-	-	-
2E_5F_DE_07	VVT- Minhub ansteuern	X	-	-	-	-	-
2E_5F_DE_08	VVT- Minhub programmieren	X	-	-	-	-	-
2E_5F_F0_07	Abgleichswert LL ansteuern	X	X	X	X	X	X
2E_5F_F0_08	Abgleichswert LL programmieren	X	X	X	X	X	X
2E_5F_F1_00	CO- Abgleich beenden	X	-	-	-	-	-

2E_5F_F1_04	CO- Abgleich Reset	X	-	-	-	-	-
2E_5F_F1_07	CO- Abgleich ansteuern	X	-	-	-	-	-
2E_5F_F1_08	CO- Abgleich programmieren	X	-	-	-	-	-
2E_5F_F2_04	Varianten loeschen	X	X	X	X	X	X
2E_5F_F3_04	Uebertemperatursicherung loeschen	X	X	X	X	X	X
2E_5F_F4_04	Fusshebelwerk Fehlbedienung loeschen	X	X	X	X	X	-
2E_5F_F5_04	Powermanagement Histogramm loeschen	X	X	X	X	X	X
2E_5F_F6_04	Ueberdrehzahlsicherung loeschen	X	X	X	X	X	X
2E_5F_F7_00	Laufruheprüfung Vorgeben beenden	X	-	-	-	-	-
2E_5F_F7_07	Laufruheprüfung vorgeben	X	-	-	-	-	-
2E_5F_F7_08	Laufruheprüfung programmieren	X	-	-	-	-	-
2E_5F_F8_00	Homogenbetrieb vorgeben beenden	-	X	X	X	X	-
2E_5F_F8_07	Homogenbetrieb vorgeben	-	X	X	X	X	-
2E_C0_01	Write k_ews4	X	X	X	X	X	X

7.8.15 Overview for SID 30h

Hint: overview taken from 17I00M21.00O

Overview for SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_02_01	Bremslichtschalter auslesen	X	X	X	X	X	X
30_03_01	Bremslichttestschalter auslesen	X	X	X	X	X	X
30_04_01	Kupplungsschalter auslesen	X	X	X	X	X	X
30_05_01	Oeldruckschalter auslesen	X	X	X	X	X	X
30_06_01	Sporttaster auslesen	X	X	X	X	X	-
30_07_01	Nullgangssensor auslesen	-	X	-	X	-	-
30_0A_01	Ansauglufttemperatur 1 auslesen	X	X	X	X	X	X
30_0B_01	Ansauglufttemperatur 2 auslesen	-	-	-	-	-	X
30_0C_01	Motortemperatur auslesen	X	X	X	X	X	X
30_0D_01	Kuehlerauslasstemperatur auslesen	X	X	X	X	X	X
30_0E_01	Oelsensor auslesen	X	X	X	X	X	X
30_10_01	Steuergeraete-Innentemperatur auslesen	X	X	X	X	X	X
30_12_01	Abgastemperatur auslesen	-	X	X	X	X	X
30_13_01	Abgastemperatur 2 auslesen	-	-	-	-	-	X
30_17_01	Umgebungsdruck auslesen	X	X	X	X	X	X
30_18_01	Saugrohrdruck1 / Ladedruck1 auslesen	X	X	X	X	X	X
30_19_01	Saugrohrdruck2 / Ladedruck2 auslesen	-	X	X	X	X	X

Overview for SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_1A_01	Raildrucksenor auslesen	-	X	X	X	X	X
30_1B_01	Kl.15 Spannung auslesen	X	X	X	X	X	X
30_1C_01	Kl.87 Spannung / Versorgung DME auslesen	X	X	X	X	X	X
30_1C_00/07	Kl.87 Spannung / Versorgung DME ansteuern	X	X	X	X	X	X
30_1E_01	Fahrerwunschversorgung 1 auslesen	X	X	X	X	X	X
30_1F_01	Fahrerwunschversorgung 2 auslesen	X	X	X	X	X	X
30_20_01	Versorgungsspannung VVL Sensor auslesen	X	-	-	-	-	-
30_20_01	Versorgung HPD5-Ventil auslesen	-	X	-	X	-	X
30_20_00/07	Versorgung HPD5-Ventil ansteuern	-	X	-	X	-	X
30_21_01	Lambdasonde vor Kat Bank 1 auslesen	X	X	X	X	X	X
30_22_01	Lambdasonde hinter Kat Bank 1 auslesen	X	X	X	X	X	X
30_23_01	Lambdasonde vor Kat Bank 2 auslesen	X	X	X	X	X	X
30_24_01	Lambdasonde hinter Kat Bank 2 auslesen	X	X	X	X	X	X
30_25_01	Luftmassenmesser 1 auslesen	X	X	X	X	X	X
30_27_01	Batteriesensor auslesen	X	X	X	X	X	X
30_28_01	Fahrerwunsch1 auslesen	X	X	X	X	X	X
30_29_01	Fahrerwunsch 2 auslesen	X	X	X	X	X	X
30_2A_01	Drosselklappe auslesen	X	X	X	X	X	X
30_2A_00/07	Drosselklappe ansteuern	X	X	X	X	X	X
30_2E_01	Sekundaerluft HFM auslesen	X	-	-	-	-	X
30_2F_01	Taster Tempomat	X	X	X	X	X	-
30_30_01	Klopfbaustein1 auslesen	X	X	X	X	X	X
30_31_01	Klopfbaustein 2 auslesen	X	X	X	X	X	X
30_32_01	Generator Sollspannung BSD auslesen	X	X	X	X	X	X
30_32_00/07	Generator Sollspannung BSD ansteuern	X	X	X	X	X	X
30_37_01	Oeldrucksensor auslesen	X	X	X	X	X	-
30_3B_01	NOx-Sensor auslesen	X	X	X	X	X	-
30_3F_01	Kraftsoffniederdrucksensor auslesen	-	X	X	X	X	X
30_40_01	Bremsunterdrucksensor auslesen	X	X	-	X	-	-
30_43_01	Raildrucksenor 2 auslesen	-	-	-	-	-	X
30_44_01	Klopfbaustein 3 auslesen	-	-	-	-	-	X
30_45_01	Klopfbaustein 4 auslesen	-	-	-	-	-	X
30_46_01	Luftmassenmesser 2 auslesen	-	-	-	-	-	X
30_80_01	Radiator shutter auslesen	-	-	X	-	X	-
30_80_00/07	Radiator shutter ansteuern	-	-	X	-	X	-
30_82_01	HPD-Relais auslesen	-	X	-	X	-	X
30_82_00/07	HPD-Relais ansteuern	-	X	-	X	-	X

Overview for SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_83_01	Wasserpumpe Turbolader auslesen	-	-	-	-	-	X
30_83_00/07	Wasserpumpe Turbolader ansteuern	-	-	-	-	-	X
30_84_01	Drosselklappenstellergehäuseheizung 2 auslesen	-	-	-	-	-	X
30_84_00/07	Drosselklappenstellergehäuseheizung 2 ansteuern	-	-	-	-	-	X
30_85_01	Drosselklappe 2 auslesen	-	-	-	-	-	X
30_85_00/07	Drosselklappe 2 ansteuern	-	-	-	-	-	X
30_86_01	(second) HPD5-Mengensteuerventil auslesen	-	X	-	X	-	X
30_86_00/07	(second) HPD5-Mengensteuerventil ansteuern	-	X	-	X	-	X
30_87_01	Abgasklappe 2 auslesen	-	-	-	-	-	X
30_87_00/07	Abgasklappe 2 ansteuern	-	-	-	-	-	X
30_88_01	Vanos Einlass Ventil Bank2 auslesen	-	-	-	-	-	X
30_88_00/07	Vanos Einlass Ventil Bank2 ansteuern	-	-	-	-	-	X
30_89_01	Vanos Auslass Ventil Bank2 auslesen	-	-	-	-	-	X
30_89_00/07	Vanos Auslass Ventil Bank2 ansteuern	-	-	-	-	-	X
30_8A_01	Umluftventil Bank 2 auslesen	-	-	-	-	-	X
30_8A_00/07	Umluftventil Bank 2 ansteuern	-	-	-	-	-	X
30_8B_01	Sekundaerluftventil 2 auslesen	-	-	-	-	-	X
30_8B_00/07	Sekundaerluftventil 2 ansteuern	-	-	-	-	-	X
30_9E_01	Drosselklappenstellergehaueseheizung auslesen	-	X	X	X	X	X
30_9E_00/07	Drosselklappenstellergehaueseheizung ansteuern	-	X	X	X	X	X
30_A4_01	Gesteuerte Luftfuehrung (untere Klappe) auslesen	X	X	X	X	X	X
30_A4_00/07	Gesteuerte Luftfuehrung (untere Klappe) ansteuern	X	X	X	X	X	X
30_AB_01	Oel Druck Regelung auslesen	X	X	X	X	X	-
30_AB_00/07	Oel Druck Regelung ansteuern	X	X	X	X	X	-
30_AC_01	Oeldruckventil auslesen	X	X	X	X	X	-
30_AC_00/07	Oeldruckventil ansteuern	X	X	X	X	X	-
30_AD_01	Kurbelgehaeuseentlueftungsheizung auslesen	-	X	X	X	X	X
30_AD_00/07	Kurbelgehaeuseentlueftungsheizung ansteuern	-	X	X	X	X	X
30_AE_01	Variable Sauganlage (DISA) Klappe2 auslesen	X	X	X	X	X	-
30_AE_00/07	Variable Sauganlage (DISA) Klappe2 ansteuern	X	X	X	X	X	-
30_B2_01	Motorlagersteuerung auslesen	-	X	X	X	X	-
30_B2_00/07	Motorlagersteuerung ansteuern	-	X	X	X	X	-
30_B5_01	Umluftventil Bank 1 auslesen	-	-	-	-	-	X
30_B5_00/07	Umluftventil Bank 1 ansteuern	-	-	-	-	-	X

Overview for SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_B6_01	Ladedrucksteller 1 auslesen	-	X	X	X	X	X
30_B6_00/07	Ladedrucksteller 1 ansteuern	-	X	X	X	X	X
30_B7_01	Ladedrucksteller 2 auslesen	-	X	X	X	X	X
30_B7_00/07	Ladedrucksteller 2 ansteuern	-	X	X	X	X	X
30_BD_01	Mengensteuerventil auslesen	-	X	X	X	X	X
30_BD_00/07	Mengensteuerventil ansteuern	-	X	X	X	X	X
30_BE_01	Abgasrückführungsventil auslesen	X	X	X	X	X	-
30_BE_00/07	Abgasrückführungsventil ansteuern	X	X	X	X	X	-
30_BF_01	elektr. Wasserpumpe ueber BSD auslesen	X	X	X	X	X	-
30_BF_00/07	elektr. Wasserpumpe ueber BSD ansteuern	X	X	X	X	X	-
30_C1_01	Abgasklappe auslesen	X	X	X	X	X	X
30_C1_00/07	Abgasklappe ansteuern	X	X	X	X	X	X
30_C2_01	Soundklappeauslesen	X	X	X	X	X	X
30_C2_00/07	Soundklappe ansteuern	X	X	X	X	X	X
30_C3_01	Gesteuerte Luftfuehrung auslesen	X	X	X	X	X	X
30_C3_00/07	Gesteuerte Luftfuehrung ansteuern	X	X	X	X	X	X
30_C4_01	Startrelais auslesen	X	X	X	X	X	X
30_C4_00/07	Startrelais ansteuern	X	X	X	X	X	X
30_C6_01	Variable Sauganlage (DISA) Klappe auslesen	X	X	X	X	X	-
30_C6_00/07	Variable Sauganlage (DISA) Klappe ansteuern	X	X	X	X	X	-
30_C7_01	Klimakompressor-Relais auslesen	X	X	X	X	X	-
30_C7_00/07	Klimakompressor-Relais ansteuern	X	X	X	X	X	-
30_C8_01	EBOX_Controlled_Fan auslesen	X	X	X	X	X	X
30_C8_00/07	EBOX_Controlled_Fan ansteuern	X	X	X	X	X	X
30_C9_01	Electr_Controlled_Thermostat auslesen	X	X	X	X	X	X
30_C9_00/07	Electr_Controlled_Thermostat ansteuern	X	X	X	X	X	X
30_CA_01	Sekundaerluftventil auslesen	X	-	-	-	-	X
30_CA_00/07	Sekundareluftventil ansteuern	X	-	-	-	-	X
30_CB_01	Sekundaerluftpumpe auslesen	X	-	-	-	-	X
30_CB_00/07	Sekundaerluftpumpe ansteuern	X	-	-	-	-	X
30_CC_01	Diagnosemodul-Tank Leckage Pumpe auslesen	X	-	X	-	X	X
30_CC_00/07	Diagnosemodul-Tank Leckage Pumpe ansteuern	X	-	X	-	X	X
30_CD_01	Diagnosemodul-Tank Leckage Ventil auslesen	X	-	X	-	X	X
30_CD_00/07	Diagnosemodul-Tank Leckage Ventil ansteuern	X	-	X	-	X	X
30_CE_01	Diagnosemodul-Tank Leckage Heizung auslesen	X	-	X	-	X	X
30_CE_00/07	Diagnosemodul-Tank Leckage Heizung ansteuern	X	-	X	-	X	X

Overview for SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_CF_01	Canister_Purge_Solenoid auslesen	X	X	X	X	X	X
30_CF_00/07	Canister_Purge_Solenoid ansteuern	X	X	X	X	X	X
30_D0_01	Lambda_Sensor_Heating_Up_1 auslesen	X	X	X	X	X	X
30_D0_00/07	Lambda_Sensor_Heating_Up_1 ansteuern	X	X	X	X	X	X
30_D1_01	Lambda_Sensor_Heating_Down_1 auslesen	X	X	X	X	X	X
30_D1_00/07	Lambda_Sensor_Heating_Down_1 ansteuern	X	X	X	X	X	X
30_D2_01	Lambda_Sensor_Heating_Up_2 auslesen	X	X	X	X	X	X
30_D2_00/07	Lambda_Sensor_Heating_Up_2 ansteuern	X	X	X	X	X	X
30_D3_01	Lambda_Sensor_Heating_Down_2 auslesen	X	X	X	X	X	X
30_D3_00/07	Lambda_Sensor_Heating_Down_2 ansteuern	X	X	X	X	X	X
30_D4_01	Malfunction_Indicator_Lamp auslesen	X	X	X	X	X	X
30_D4_00/07	Malfunction_Indicator_Lamp ansteuern	X	X	X	X	X	X
30_D5_01	Fahrgeschwindigkeitsregler-Lampe auslesen	X	X	X	X	X	X
30_D5_00/07	Fahrgeschwindigkeitsregler-Lampe ansteuern	X	X	X	X	X	X
30_D6_01	EML_Lamp auslesen	X	X	X	X	X	X
30_D6_00/07	EML_Lamp ansteuern	X	X	X	X	X	X
30_D8_01	Electrical_Fuel_Pump auslesen	X	X	X	X	X	X
30_D8_00/07	Electrical_Fuel_Pump ansteuern	X	X	X	X	X	X
30_DA_01	E-Box-Luefter auslesen	X	X	X	X	X	X
30_DA_00/07	E-Box-Luefter ansteuern	X	X	X	X	X	X
30_DC_01	VVT-Entlastungsrelais auslesen	X	-	-	-	-	X
30_DC_00/07	VVT-Entlastungsrelais ansteuern	X	-	-	-	-	X
30_DD_01	ValveTronic Lift auslesen	X	-	-	-	-	X
30_DD_00/07	ValveTronic Lift ansteuern	X	-	-	-	-	X
30_E1_01	Einspritzventil 1 auslesen	X	X	X	X	X	X
30_E1_00/07	Einspritzventil 1 ansteuern	X	-	-	-	-	-
30_E2_01	Einspritzventil 2 auslesen	X	X	X	X	X	X
30_E2_00/07	Einspritzventil 2 ansteuern	X	-	-	-	-	-
30_E3_01	Einspritzventil 3 auslesen	X	X	X	X	X	X
30_E3_00/07	Einspritzventil 3 ansteuern	X	-	-	-	-	-
30_E4_01	Einspritzventil 4 auslesen	X	X	X	X	X	X
30_E4_00/07	Einspritzventil 4 ansteuern	X	-	-	-	-	-
30_E5_01	Einspritzventil 5 auslesen	X	-	X	-	X	X
30_E5_00/07	Einspritzventil 5 ansteuern	X	-	-	-	-	-
30_E6_01	Einspritzventil 6 auslesen	X	-	X	-	X	X
30_E6_00/07	Einspritzventil 6 ansteuern	X	-	-	-	-	-
30_E7_01	Einspritzventil 7 auslesen	-	-	-	-	-	X
30_E8_01	Einspritzventil 8 auslesen	-	-	-	-	-	X
30_ED_01	Vanos Einlass Ventil auslesen	X	X	X	X	X	X

Overview for SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_ED_00/07	Vanos Einlass Ventil ansteuern	X	X	X	X	X	X
30_EE_01	Vanos Auslass Ventil auslesen	X	X	X	X	X	X
30_EE_00/07	Vanos Auslass Ventil ansteuern	X	X	X	X	X	X
30_EF_01	HPD5-Mengensteuerventil 1 auslesen	-	-	-	-	-	X
30_EF_00/07	HPD5-Mengensteuerventil 1 ansteuern	-	-	-	-	-	X

7.8.16 Overview for SID 31h

Hint: overview taken from 17I00N0D.0AB and 17I03901.00O

SID number	service	Service Location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
31_01	ask checksums after reprogramming	BOOT	X	X	X	X	X	X
31_02	erase Memory	BOOT	X	X	X	X	X	X
31_03	erase History Memory	ECU	X	X	X	X	X	X
31_05	power down	ECU	X	X	X	X	X	X
31_07	start authentification	BOOT	X	X	X	X	X	X
31_08	check authentification	BOOT	X	X	X	X	X	X
31_09	check signature	BOOT	X	X	X	X	X	X
31_0A	report reprogramming status	BOOT	X	X	X	X	X	X
31_0C	energy spare mode	ECU	X	X	X	X	X	X
31_1F_EE	EWS4: Start calculation and storage	ECU	X	X	X	X	X	X
31_1F_E3	EWS4: Read out GXMODN data	ECU	X	X	X	X	X	X
31_1F_E4	EWS4: Read out length of GXMODN	ECU	X	X	X	X	X	X
31_1F_F1	EWS4: Transfer data of FSC to ECU	ECU	X	X	X	X	X	X
31_1F_F2	EWS4: Transfer length of ESC (FreiSchaltCode) to ECU	ECU	X	X	X	X	X	X
31_20	EOL Test ON: Secondary air system	ECU	X	X	X	X	X	X
31_22	EOL Test ON: CPS	ECU	X	X	X	X	X	X
31_25	switch off Injection valves	ECU	X	X	X	X	X	X
31_26	EOL Test ON: Idle speed setpoint displace	ECU	X	X	X	X	X	X
31_27	start VVL Adaptation	ECU	X	-	-	-	-	
31_2A	Start intelligent alternator test	ECU	X	X	X	X	X	X
31_2B	Start Ruhestrom test	ECU	X	X	X	X	X	X
31_2C	Start EOL test: Oil Pressure Regulation Diagnosis	ECU	X	X	X	X	X	-

SID number	service	Service Location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
31_2D	Start EOL test: Desulfatisierung	ECU	-	X	X	X	X	-
31_2E	Start Nullgang-Lernen	ECU	-	X	-	X	-	-
31_2F	Start Desulfatisierung Fahrbetrieb	ECU	-	X	X	X	X	-
31_30	delete selective adaption values	ECU	X	X	X	X	X	X
31_31	delete selective adaption values 2	ECU	-	X	X	X	X	X
31_CF	Start : Sperre Tankentlueftung	ECU	-	X	X	X	X	
31_D0	Start: Werkstattdiagnose ATL	ECU	-	X	X	X	X	X
31_D5	EOL Test ON: ECRAS system test	X	X	X	X	X	X	
31_D9	switch off Lambda controller	ECU	X	X	X	X	X	X
31_DA	EOL Test ON: DMTL	ECU	X	X	X	X	X	X
31_DF	EOL Test ON: Lambda sensors confused / wrong wired	ECU	X	X	X	X	X	X
31_E0	Ansteuern Eisy- Adaptionswerte (ungedrosselt)	ECU	X	-	-	-	-	-
31_E1	Ansteuern Eisy- Adaptionswerte (gedrosselt)	ECU	X	X	X	X	X	X
31_E2	Ansteuern Eisy- Adaptionswerte (gedrosselt mit Abgasrueckführung)	ECU	-	X	X	X	X	X
31_E3	Ansteuern Krann- Adaptionswerte	ECU	X	X	X	X	X	X
31_E4	Ansteuern Klann-Adaptionswerte	ECU	X	X	X	X	X	X
31_E5	Ansteuern AGR-Adaptionswerte	ECU	-	X	-	X	-	-
31_F0	Calculate checksum for IncaPC	ECU	X	-	-	-	-	-
31_F1	Programming of calibration data for IncaPC	ECU	X	-	-	-	-	-
31_F4	cruise control delete switch off conditions	ECU	X	X	X	X	X	X
31_F5	switch off Idle speed uplift	ECU	X	X	X	X	X	X
31_F6	set default values PM for BN-SS	ECU	X	X	X	X	X	X
31_F7	Intelligente Generatorregelung	ECU	X	X	X	X	X	
31_FA	Siemens interner Service	ECU / BOOT	X	X	X	X	X	X
31_FB	Siemens interner Service	ECU / BOOT	X	X	X	X	X	X
31_FC	Siemens interner Service	ECU / BOOT	X	X	X	X	X	X
31_FE	Message using CAN message	ECU	X	-	-	-	-	-

7.8.17 Overview for SID 32h

Hint: overview takenfrom 17I00Z02.00F and 17I03901.00O

SID number	service	Service Location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
32_20	EOL Test OFF: Secondary air system	ECU	X	-	-	-	-	X
32_22	EOL Test OFF: CPS	ECU	X	X	X	X	X	X
32_26	EOL Test OFF: Idle speed setpoint displace	ECU	X	X	X	X	X	X
32_27	Stop EOL - VVL Adaptation	ECU	X	-	-	-	-	-
32_2A	Stop intelligent alternator test	ECU	X	X	X	X	X	X
32_2C	Stop EOL test: Oil Pressure Regulation Diagnosis	ECU	X	X	X	X	X	-
32_2D	Stop EOL test: Desulfatisierung	ECU	-	X	X	X	X	-
32_2F	Stop Desulfatisierung Fahrbetrieb	ECU	-	X	X	X	X	-
32_CF	Stop : Sperre Tankentlueftung	ECU	-	X	X	X	X	-
32_D0	Stop: Werkstattdiagnose ATL	ECU	-	X	X	X	X	X
32_D5	Stop EOL: ECRAS system test	ECU	X	X	X	X	X	X
32_D9	stop switch off Lambda controller	ECU	-	X	X	X	X	X
32_DA	EOL Test OFF: DMTL	ECU	X	X	X	X	X	X
32_DF	EOL Test OFF: Lambda sensors confused / wrong wired	ECU	X	X	X	X	X	X
32_F0	stop checksum calculation for Inca PC	ECU	X	-	-	-	-	-
32_F1	stop programming of calibration data for IncaPC	ECU	X	-	-	-	-	-
32_F6	stop default values PM for BN-SS	ECU	-	X	X	X	X	X
32_F7	switch Intelligente Generatorregelung active	ECU	X	X	X	X	X	X

7.8.18 Overview for SID 33h

Hint: overview takenfrom 17I01101.00F and 17I03901.00O

SID number	service	Service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85(8cyl)
33_20	**Report EOL results: Secondary air system**	ECU	**X**	-	-	-	-	**X**
33_22	**Report EOL results: CPS**	ECU	**X**	**X**	**X**	**X**	**X**	**X**
33_25	read INH_IV_KWP - Diagnos Status EV-Ausblendung	ECU	-	X	X	X	X	X
33_26	**Report EOL results: Idle speed setpoint displace**	ECU	**X**	**X**	**X**	**X**	**X**	**X**
33_27	**Read Status EOL: VVL adaptation**	ECU	**X**	-	-	-	-	-
33_2A	**Read Status: inteligent alternator test**	ECU	**X**	**X**	**X**	**X**	**X**	**X**
33_2B	Read result Ruhestrom Test	ECU	X	X	X	X	X	X
33_2F	Auslesen Desulfatisierung Fahrbetrieb	ECU	-	X	X	X	X	-
33_2C	**Read Status EOL test: Oil Pressure Regulation Diagnosis**	ECU	**X**	**X**	**X**	**X**	**X**	-
33_2D	**Read status EOL test: Desulfatisierung**	ECU	-	**X**	**X**	**X**	**X**	-
33_CF	**Read Statrus : Sperre Tankentlueftung**	ECU	-	**X**	**X**	**X**	**X**	-
33_D0	**Read Status: Werkstattdiagnose ATL**	ECU	-	**X**	**X**	**X**	**X**	**X**
33_D5	**Read Status EOL: ECRAS system test**	ECU	**X**	**X**	**X**	**X**	**X**	**X**
33_D9	Read Status: switch off lambda controller	ECU	-	X	X	X	X	X
33_DA	**Report EOL results: DMTL**	ECU	**X**	**X**	**X**	**X**	**X**	**X**
33_DF	**Report EOL results: Lambda sensors confused / wrong wired**	ECU	**X**	**X**	**X**	**X**	**X**	**X**
33_E0	Auslesen Eisy- Adaptionswerte (ungedrosselt)	ECU	X	-	-	-	-	-
33_E1	Auslesen Eisy- Adaptionswerte (gedrosselt)	ECU	X	X	X	X	X	X
33_E2	Auslesen Eisy- Adaptionswerte (gedrosselt mit Abgasrueckfuehrung)	ECU	-	X	X	X	X	X
33_E3	Auslesen Krann- Adaptionswerte	ECU	X	X	X	X	X	X
33_E4	Auslesen Klann-Adaptionswerte	ECU	X	X	X	X	X	X
33_E5	Auslesen AGR-Adaptionswerte	ECU	-	X	X	X	-	-
33_F0	Report results of checksum calculation for INCA PC	ECU	X	-	-	-	-	-
33_F1	Report results of programming the calibration data for INCA PC in the power latch phase	ECU	X	-	-	-	-	-
33_F6	Report Status: Steuern Messsemode stop default value	ECU	-	X	X	X	X	X
33_F7	report results of Intelligente Generatorregelung	ECU	X	X	X	X	X	X

7.8.19 Overview for SID 34h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming	Mode 86 development	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
34	Flash Programmierung	RequestDownload	DI	L3	L5	DI	BOOT	X	X	X	X	X	X

7.8.20 Overview for SID 35h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming	Mode 86 development	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
35	Request upload	UploadRequest	DI	L3	L5	DI	BOOT	X	X	X	X	X	X

7.8.21 Overview for SID 36h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming	Mode 86 development	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
36	Daten austauschen	TransferData	DI	L3	L5	DI	BOOT	X	X	X	X	X	X

7.8.22 Overview for SID 37h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
37	Flash-Programmierung beenden	RequestTransferExit	DI	L3	L5	DI	BOOT	X	X	X	X	X	X

7.8.23 Overview for SID 3Bh

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
3B_90	Kurze Fahrgestellnummer schreiben	WriteDataByLocalIdentifier	ENA	DI	L5	DI	ECU	X	X	X	X	X	X
3B_C1	KVA-Faktor schreiben	WriteDataByLocalIdentifier	ENA	DI	L5	DI	ECU	X	X	X	X	X	X

7.8.24 Overview for SID 3Dh

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
3D	Speicher schreiben	WriteMemoryByAddress	DI	L3	L5	DI	BOOT	X	X	X	X	X	X

7.8.25 Overview for SID 3Eh

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
3E	Enable positive response	TesterPresent	ENA	L3	L5	L4	Boot-/ECU	X	X	X	X	X	X
3E_01	Enable positive response	TesterPresent	ENA	ENA	ENA	ENA	BOOT	X	X	X	X	X	X
3E_02	Disable positive response	TesterPresent	ENA	ENA	ENA	ENA	BOOT	X	X	X	X	X	X

7.8.26 Overview for SID 80h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
80_90_00	K-Linie-Applikation-Tool Reference page	EscapeCode	ENA	DI	L5	DI	BOOT	X	-	-	-	-	-
80_90_01	K-Linie-Applikation-Tool Working page RAM	EscapeCode	ENA	DI	L5	DI	BOOT	X	-	-	-	-	-

7.8.27 Overview for SID 81h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
81	Aufbau Testerverbindung	StartCommunication	ENA	DI	L5	DI	BOOT	X	-	-	-	-	-

7.8.28 Overview for SID 82h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
82	Abbau Testerverbindung	StopCommunication	ENA	L3	L5	L4	BOOT	X	-	-	-	-	-

7.8.29 Overview for SID 83h

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
83_00	Read limits	AccessTimingParameter	ENA	L3	L5	DI	BOOT	X	-	-	-	-	-
83_01	Set parameter to default	AccessTimingParameter	ENA	L3	L5	DI	BOOT	X	-	-	-	-	-
83_02	Read activ parameters	AccessTimingParameter	ENA	L3	L5	DI	BOOT	X	-	-	-	-	-
83_03	Set	AccessTimingParamet	EN	L	L	D	BOO	X	-	-	-	-	-

general specification

SID number	service	type	Mode 81 Standard mode	Mode 85 programming mode	Mode 86 development mode	Mode 89 safety mode	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD_85 (L4)
	parameter	er	A	3	5	I	T						

7.9 SID 10h: startDiagnosticSession Service

Input data:

LV_IGK			

General information:

This function is the first function when a diagnostic session starts.
Changing the diagnostic mode or switch over to another baudrate is possible after the command "stop" or "reset" KWP2000 inactive mode is set.

StartDiagnosticSession Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startDiagnosticSessionRequest Service Id**	**M**	**#10h**	**STDS**
#2d	diagnosticMode= [Stdandard Diag.Mode Periodic Transmission Mode ECU Programming Mode Development Mode safty mode]	M	#XXh= [#81h #82h #85h #86h #89h]	DCM
#3d	baudrateIdentifier (#XXh) = [9600 Baud 19200 Baud 38400 Baud 57600 Baud 115200 Baud (113636 Baud) (*) by tester]	C	#XXh= [#01h #02h #03h #04h #05h #06h]	BI
#4d	Baud rate high byte (if baudrateidentifier = 6)	C	#XXh	#XX
#5d	Baud rate middle byte (if baudrateidentifier = 6)	C	#XXh	#XX
#6d	Baud rate low byte (if baudrateidentifier = 6)	C	#XXh	#XX

Table startDiagnosticSession Request Message

(*) : With the MPC555 controller and a frequency of 40 MHz it is not possible to generate 115200baud with the desired tolerance range of -+1%.
The baudrate, which can be generated exactely by MPC555 and 40MHz, is 113636Baud.

☐ Is the current DiagnosticSession different from StandardDiagnosticSession, it is not possible to get a positive response with the request stopCommunication. To stop the communication the ECU has to be turned back in the StandardDiagnosticSession with stopDiagnosticSession (SPDS) or the startDiagnosticSession (STDS [DCM #81h]) request.

☐ Is the startDiagnosticSession request used without a baudrateMode or is the current request answered by negative response, the last selected baudrate and mode are still active.

StartDiagnosticSession positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startDiagnosticSession Positive response**	**M**	**#50h**	**STDSPR**
#2d	diagnosticMode= [Std.Diagnostic Mode Periodic Transmission Mode ECU Programming Mode Development mode safty mode]	M	#XXh= [#81h #82h #85h #86h #89h]	DCM
#3d	BaudrateIdentifier (#XXh) = [f_{CPU} =16MHz f_{CPU} =20 MHz f_{CPU} =24 MHz 9600 Baud + 0,2% + 0,2% + 0,2% 19200 Baud + 0,2% + 1,7% + 0,2% 38400 Baud + 0,2% + 1,7% + 2,8% 57600 Baud + 8,5% + 8,5% + 0,2% 115200 Baud + 8,5% + 8,5% + 8,5% by tester]	C	#XXh= [#01h #02h #03h #04h #05h #06h]	BI
#4d	Baud rate high byte	C	#XXh	#XX
#5d	Baud rate middle byte	C	#XXh	#XX
#6d	Baud rate low byte	C	#XXh	#XX

Table startDiagnosticSession Positive Response Message

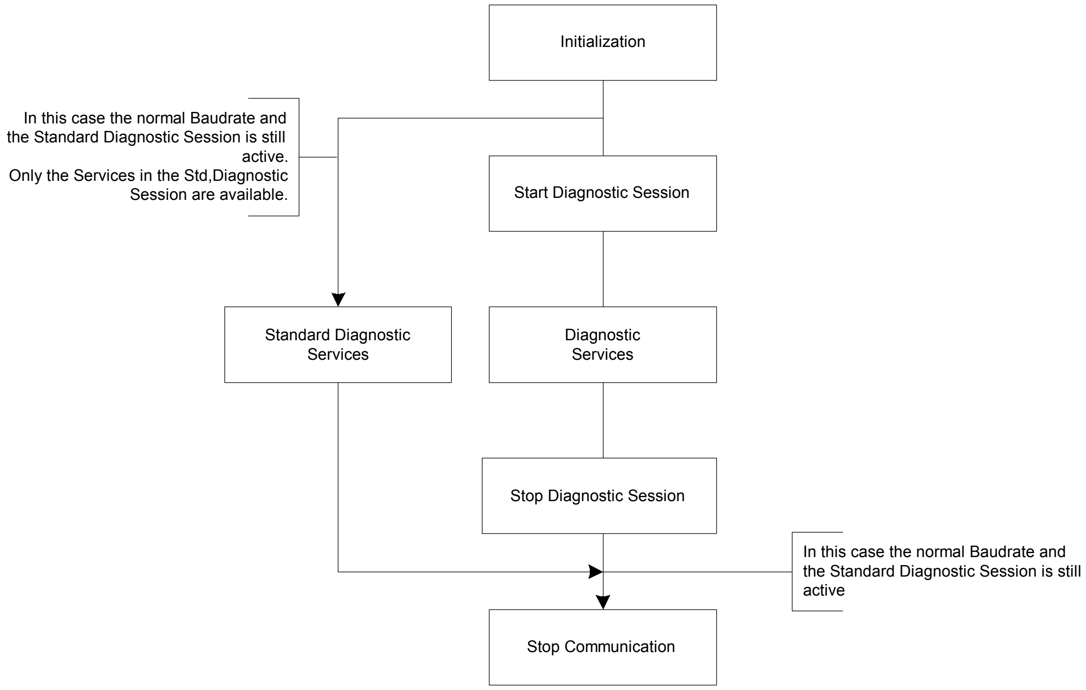
StartDiagnosticSession negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	startDiagnosticSessionRequest Service Id	M	#10h	STDS
#3d	responseCode= [conditionsNotCorrect, requestOutOfRange, wrong baud rate]	M	#XXh= [#22h, #31h,]	RC

Table startDiagnosticSession Negative Response Message

NegativeResponseCode	*Cause of Occurrence*
conditionsNotCorrect	selected diagnosticMode is not supported
requestOutOfRange	selected baudrate is not supported

Message flow Diagnostic Session



StartDiagnosticSession (Periodic Transmission mode #82h)

Is only supported for KWP

KWP2000 Periodic Transmission Mode Message Flow

This section specifies the conditions to enable the ***Periodic Transmission*** mode ($82) in the client (tester) and the server (ECU).
It also specifies the use of the service ***accessTimingParameter*** within this mode to modify the ***PeriodicTransmission mode*** default timing.

Periodic Transmission Mode Message Flow

time	*Client (tester) Request Message*	*Hex*
	startCommunication Request Service ID	**#81h**

time	*Server (ECU) PosResponse Message #2*	*Hex*
P2	**startCommunication PosRspSId** [Key byte 1 Key byte 2]	**#C1h** #XXh #XXh
	goto STEP#1 or STEP#2	

STEP#1 dynamicallyDefineLocalIdentifier Request Service() (optinal)

time	*Client (tester) Request Message*	Hex
P3*	**dynamicallyDefineLocalIdentifier Request Service ID**	**#2Ch**
	[	
	possible parameter see service description	#XXh
	]	
	define the DDLI #F0h	

time	*Server (ECU) PosResponse Message #2*	Hex
P2	**dynamicallyDefineLocalIdentifier PosRspSId=**	**#C3h**
	[	
	...	#XXh
	]	
	goto STEP#2	

Server (ECU) NegResponse Message	Hex
negRespSId	**#7Fh**
dynamicallyDefineLocalIdentifier ReqSId	**#2Ch**
[	
responseCode { see service description]	#XXh
]	

: :

: :

STEP#2 startDiagnosticSession(DM_PeriodicTransmission)

time	*Client (tester) Request Message*	*Hex*
P3	**startDiagnosticSession Request Service ID=**	**#10h**
	[	
	diagnosticMode = PeriodicTransmission SDMWPT	#82h
	]	

time	*Server (ECU) PosResponse Message #1*	*Hex*
P2	**startDiagnosticSession PosRspSId=**	**#50h**
	[	
	diagnosticMode = SDMWPT	#82h
	PeriodicTransmission Mode	
	]	

Server (ECU) NegResponse Message	*Hex*
negResponseSId	**#7Fh**
startDiagnosticSession ReqSId=	**#21h**
[	
responseCode { see service description]	#XXh
]	

*** **PeriodicTransmission mode** with default timing enabled ***

time	*Server (ECU) PosResponse Message #2*	*Hex*
P2	**startDiagnosticSession PosRspSId=**	**#50h**
	[	
	diagnosticMode = SDMWPT	#82h
	]	

: :

time	*Server (ECU) PosResponse Message #n*	*Hex*
P2	**startDiagnosticSession PosRspSId=**	**#50h**
	[	
	diagnosticMode = SDMWPT	#82h
	]	
	goto STEP#2 or STEP #3	

: :

STEP#2 accessTimingParameters(readLimitsOfPossibleValues) (optional)

time	*Client (tester) Request Message*	*Hex*
P3*	**accessTimingParameters Request Service ID =**	**#83h**
	[	
	TPI = readLimitsOfPossibleValues	#00h
	]	

time	*Server (ECU) PosResponse Message #1*	*Hex*
P2	**accessTimingParameters PosRspSId=**	**#C3h**
	[	
	...	#XXh
	]	
	goto STEP#3 or to STEP#4	

Server (ECU) NegResponse Message	*Hex*
negRespSId	**#7Fh**
accessTimingParameters ReqSId=	**#83h**
[	
responseCode { see service description]	#XXh
]	

: :

time	*Server (ECU) PosResponse Message #n*	*Hex*
P2	**accessTimingParameters PosRspSId=**	**#C3h**
	[	
	...	#XXh
	]	
	goto STEP#3 or to STEP#4	

: :

STEP#3 accessTimingParameters(setParameters) (optional)

time	*Client (tester) Request Message*	*Hex*
P3*	**accessTimingParameters Request Service ID =** [TPI = setParameters ...:]	**#83h** #03h #XXh

time	*Server (ECU) PosResponse Message #1*	*Hex*
P2	**accessTimingParameters PosRspSId=** [TPI = setParameters]	**#C3h** #03h
	goto STEP#5	

Server (ECU) NegResponse Message	*Hex*
negRespSId **accessTimingParameters ReqSId** [responseCode { see service description]]	**#7Fh** **#83h** #XXh

*** **modified** standardDiagnosticMode timing parameter active ***

: :

time	*Server (ECU) PosResponse Message #n*	*Hex*
P2	**accessTimingParameters PosRspSId=** [TPI = setParameters]	**#C3h** #03h
	goto STEP#4	

STEP#4 periodic transmission diagnostic service´s(possible services refer to chapter)

time	*Client (tester) Request Message*	*Hex*
P3*	**periodic transmission diagnostic service**	**#XXh**

time	*Server (ECU) PosResponse Message #1*	*Hex*
P2	**periodic transmission diagnostic service PosRspSId=** [record values]	**#XXh** ...
	to STEP#5	

Server (ECU) NegResponse Message	*Hex*
negRespSId **periodic transmission diagnostic service** [responseCode { see service description]]	**#7Fh** **#XXh** #XXh

: :

time	*Server (ECU) PosResponse Message #1*	*Hex*
P2	**periodic transmission diagnostic service PosRspSId=** [record values]	**#XXh** ...
	to STEP#5	

: :

STEP#5 stopDiagnosticSession(DM_PeriodicTransmission)

time	*Client (tester) Request Message*	*Hex*
P3	**stopDiagnosticSession Requset Service Id**	**#20h**

time	*Server (ECU) PosResponse Message #1*	*Hex*
P2	**stopDiagnosticSession PosRspSId**	**#60h**
	to STEP#6	

Server (ECU) NegResponse Message	*Hex*
negRespSId	**#7Fh**
stopDiagnosticSession.ReqSId=	**#20h**
[	
responseCode { see service description]	#XXh
]	

*** ***standardDiagnosticModeWithPeriodicTransmission*** disabled ***
default diagnostic session enabled and normal timing default values active

STEP#6 stopCommunication

time	*Client (tester) Request Message*	*Hex*
P3*	**stopCommunication Requst Service Id**	**#82h**

time	*Server (ECU) PosResponse Message*	*Hex*
P2	**stopCommunication PosRspSId**	**#C2h**

Server (ECU) NegResponse Message	*Hex*
negativeResponse Service Identifier	**#7Fh**
stopCommunication.ReqSId=	**#82h**
[	
responseCode { see service description]	#XXh
]	

Possible Periodic Transmission Services

Possible services in the Periodic Transmission mode	*Mnemonic*	*SID*	*Description*
ReadDataByLocalIdentifier	RDBLI	#21h	Read of previous defined „dynamically defined local id“ (#F0h) or snapshot information.
ReadMemoryByAdress	RMBA	#23h	Read of memory contents
ReadStatusOfDiagnosticTroubleCodes	RSODTC	#17h	Read all Diagnostic Trounble Codes
ReadDiagnosticTroubleCodesByStatus	RDTCBS	#18h	Read one Diagnostic Trounble Codes with System Supplier Data
TesterPresent	TP	#3Eh	Tester Present without contents

NOTE➲: In case the Tester has defined a DynamicallyDefinedLocal ID (#F0h) all other services in the periodic transmission mode are blocked except the ReadDataByLocalID (#F0). With the service DDLI (sub identifier „clear DDLI“) in the Standart Diagnostic Mode the Tester can release all other services for the periodic transmission mode again.

Periodic Transmission Message Flow Examples

NOTE➲: In the column „Activity“ markes with the black row the Tester request messages and with the shaded row the ECU responses.

Periodic Transmission with ATP and 3 Byte Header

Time in ms	*Hex values*	*Mnemonic*	*Activity*
49.2	81.		■
5.1	11		
5.1	F1		
5.1	81	STC	
5.1	04		
28.5	83		▨
0.0	F1		
0.0	11		
0.0	C1	STC_PR	
0.0	EF		
0.0	8F		
0.0	C4		
194.9	82		■
5.1	11		
5.1	F1		
5.1	10	STDS	
5.1	82		
5.1	16		
24.7	82		▨
0.0	F1		
0.0	11		
0.0	50	STDS_PR	
0.0	82		
0.0	56		
14.0	82		■
5.1	11		
5.1	F1		
5.1	83	ATP	
5.1	00		
5.1	07		
29.1	89		▨
0.0	F1		
0.0	11		
0.0	C3	ATP_PR	
0.0	00		
0.0	08		
0.0	01		
0.0	00		
0.0	14		
0.0	00		
0.0	69		
29.5	87		▨
0.0	F1		
0.0	11		
0.0	C3	ATP_PR	
0.0	00		
0.0	08		
0.0	01		
0.0	00		
0.0	14		
0.0	00		
0.0	69		
:			:

Periodic Transmission with DDLI and 1 Byte Header

Time in ms	*Hex values*	*Mnemonic*	*Activity*
49.2	01		■
5.1	81	STC	
5.1	82		
27.0	03		▨
0.0	C1	STC_PR	
0.0	EF		
0.0	8F		
0.0	42		
192.9	08		■
5.1	2C	DDLI	
5.1	F0		
5.1	03		
5.1	01		
5.1	01		
5.1	00		
5.1	00		
5.1	00		
5.1	29		
27.0	02		▨
0.0	6C	DDLI_PR	
0.0	F0		
0.0	5E		
118.4	02		■
5.1	10	STDS	
5.1	82		
5.1	94		
28.8	02		▨
0.0	50	STDS_PR #1	
0.0	82		
0.0	D4		
:			:
26.0	02		▨
0.0	50	STDS_PR #n	
0.0	82		
0.0	D4		
:			:

:   :

Time in ms	Hex value	Mnemonic	Activity
5.1	11		■
5.1	F1		
5.1	83	ATP	
5.1	03		
5.1	08		
5.1	01		
5.1	00		
5.1	14		
5.1	00		
5.1	2C		
32.5	82		░
0.0	F1		
0.0	11		
0.0	C3	ATP_PR #1	
0.0	03		
0.0	4A		
:			:
4.0	82		░
0.0	F1		
0.0	11		
0.0	C3	ATP_PR #n	
0.0	03		
0.0	4A		
1.0	82		■
0.2	11		
0.2	F1		
0.2	83	ATP	
0.2	01		
0.2	08		
6.4	82		░
0.0	F1		
0.0	11		
0.0	C3	ATP_PR #1	
0.0	01		
0.0	48		
34.3	82		░
0.0	F1		
0.0	11		
0.0	C3	ATP_PR #2	
0.0	01		
0.0	48		

:   :

:   :

Time in ms	Hex value	Mnemonic	Activity
14,4	02		■
5,1	21	RDBLI	
5,1	F0		
5.1	13		
32.8	03		░
0.0	61	RDBLI_PR #1	
0.0	F0		
0.0	FA		
0.0	4E		
25.0	03		░
0.0	61	RDBLI_PR #2	
0.0	F0		
0.0	FB		
0.0	4F		
:			:
25.4	03		░
0.0	61	RDBLI_PR #n	
0.0	F0		
0.0	FC		
0.0	50		
6.0	01		■
5.1	20	SPDS	
5.1	21		
26.1	01		░
0.0	60	SPDS_PR	
0.0	61		
88.9	01		■
5.1	82	SPC	
5.1	83		
25.3	01		░
0.0	C2	SPC_PR	
0.0	C3		

7.10 KWP2000 - SID 20h: stopDiagnosticSession

General information:

This function is for ending a diagnostic session.

stopDiagnostic Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**stopDiagnosticSessionRequest Service Id**	**M**	**#20h**	**SPDS**

Table stopDiagnostic Request Message

StopDiagnosticSession positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	stopDiagnosticSession Positive response	M	#60h	SPDSPR

Table ecuReset Positive Response Message

☐ After positive response, the Standard Diagnostic Session, the default timing and baudrate(10400 baud for KWP and 9600baud for KWP*) are active !

StopDiagnosticSession negative Response

☐ A negative response after StopDiagnosticSession is not possible!

7.11 SID 3Eh: testerPresent Service

Input data:

LV_IGK			

General information:

This funktion is for keeping KWP2000 communication alive.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: Boot-/ ECU-SW

TesterPresent Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**testerPresent Request Service Id**	**M**	**#3Eh**	**TP**
#2d	testerPresent Request Service Id [no parameter or #01h: ECU shall send a positive response #02h: ECU shall not send a positive response] **Remark:** A negative response shall be sent in any case if an error occurs	M	#XXh [-- / #01h, #02h]	TP

Table testerPresent Request Message

TesterPresent positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**testerPresent Positive Response Service Id**	**M**	**#7Eh**	**TPPR**

Table testerPresent Positive Response Message

TesterPresent negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	inputOutputControlByLocalIdentifier Request SId	M	#3Eh	IOBLID
#3d	responseCode=only general respons code	M	#XXh	RC

Table InputOutputControlByLocalIdentifier negative response message

7.12 SID 28h: disableNormalMmessageTransmission Services

7.12.1 28h - DisableNormalMessageTransmission service

Input data:

LV_IGK			

General information:

This command is used to inhibit J1850 normal message transmission from PCM (intersystem) while still performing other functions normally. That means, that PCM should still receive all its J1850 messages and process them as required.

Description:

The parameter ***responseRequired*** is used in the DisableNormalMessageTransmission request message. It indicates to the server(s) whether a positive response message is required or not.
A negative response shall even be sent if responseRequired is set to “no”

If the “responseRequired” parameter in the DisableNormalMessageTransmission request message is missing, the server(s) shall sent a response message.

The DisableNormalMessageTransmission service shall be used by a client to stop the normal (non diagnostic) and network management message transmission from the vehicle server(s).

Fault detection and storage is also disabled by this service.

The DisableNormalMessageTransmission request messagecan be used with both physical and functional addressing to stop the message transmission from an individual server or a group of servers. The server(s) being addresed shall respond with a DisableNormalMessageTransmission positive response massage if the response option is set so or, if unable to stop the message transmission, respond with a DisableNormalMessageTransmission negative response message.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see "list of implemented diagnostic services"

DisableNormalMessageTransmission service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	DisableNormalMessageTransmission service	**M**	**#28h**	**DNMT**

#2d	DisableNormalMessageTransmission	M	#XXh	
	[		[	
	yes - server shall send a positive or negative response to the request message		#01h,	
	no - server shall not send a positive response to the request		#02h	
	]		]	

Table ReadDataByCommonIdentifier Request Message

DisableNormalMessageTransmission positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	DisableNormalMessageTransmission service Response Service Id	**M**	**#68h**	

Table ReadDataByCommonIdentifier Positive Response Message

DisableNormalMessageTransmission negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	ReadDataByCommonIdentifier Request Service Id	M	#28h	
#3d	responseCode=only general response code	M	#XXh	

Table: ReadDataByCommonIdentifier Negative Response Message

7.13 SID 29h: enableNormalMmessageTransmission Services

7.13.1 29h - EnableNormalMessageTransmission service

Input data:

LV_IGK			

General information:

This command is used to cancel a mode 28h DisableNormalMessageTransmission request.

The parameter responseRequired is used in the EnableNormalMessageTransmission request message. It Indicates to the server(s) whether a positive response message is required or not. IA negative response shall even be sent if responseRequired is set to no.

Description:

The EnableNormalMessageTransmission request message shall be used to indicate to the server that normal transmission shall be resumed. The service is used in combination with the DisableNormalMessageTransmission service to once again start the normal and network management message transmission from the server.

Fault detection and storage is also Enabled by this service.

The EnableNormalMessageTransmission request message can be used with both physical and functional addressing to resume normal messages transmission friom an individual server or a group of servers.

The server(s) being addresed shall respond with a EnableNormalMessageTransmission negative response message.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see "list of implemented diagnostic services"

EnableNormalMessageTransmission service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	EnableNormalMessageTransmission service	**M**	**#29h**	**DNMT**
#2d	EnableNormalMessageTransmission [no parameter or #01h: ECU shall send a positive response #02h: ECU shall not send a positive response] **Remark:** A negative response shall be sent in any case if an error occurs	M	#XXh [-- / #01h, #02h]	

Table ReadDataByCommonIdentifier Request Message

EnableNormalMessageTransmission positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	EnableNormalMessageTransmission service Response Service Id	**M**	**#69h**	

Table ReadDataByCommonIdentifier Positive Response Message

EnableNormalMessageTransmission negative Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	ReadDataByCommonIdentifier Request Service Id	M	#29h	
#3d	responseCode=only general response code	M	#XXh	

Table: ReadDataByCommonIdentifier Negative Response Message

7.14 SID 11h: ecuReset Service

Input data:

LV_IGK	LV_LOCK_IMOB		

General information:

This function is for reset the ECU during a diagnostic session. Before the ECU is reset, the service writes down the EWS starting value to the E2PROM and once this is finished, the EWS starting value will be copied to flash memory. This happens only if no first code is detected in the flash memory and LV_LOCK_IMOB = 0 after service 31_83_00 has come up with response 00.
This function works only by engine speed=0rpm.

ecuReset Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ecuReset Request Service Id**	**M**	**#11h**	**ER**
#2d	resetMode=[powerOn]	M	#01h	RM

Table ecuReset Request Message

☐ Attention, after the positive response the Communication is lost. A new Initialization is possible after the time W5(300ms).

☐ This service is only possible in the reprogramming session. Only a session change to the ECU programmingMode is not sufficient to execute this service!!!

EcuReset positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ecuReset Positive Response Service Id**	**M**	**#51h**	**ERPR**
#2d	resetMode=reset state	C	#01h	RS

Table ecuReset Positive Response Message

EcuReset negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	ecuReset Request Service Id	M	#11h	ER
#3d	responseCode= [general response codes conditionsNotCorrect,]	 M 	#XXh= [#XX #22h,]	RC

Table ecuReset Negative Response Message

NegativeResponseCode	*Cause of Occurrence*
conditionsNotCorrect	engine is running or ignition is off

7.15 SID 14h: clearDiagnosticInformation Service

7.15.1 14h - ClearDiagnosticInformation service

Input data:

LC_FMY_CLR	LV_IGK		

General information:

This function clears all diagnostic informations of the selected group.

Application conditions:

Initialisation: at reset

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0
Diagnostic timeout

Service location: see "list of implemented diagnostic services"

ClearDiagnosticInformation Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**clearDiagnosticInformation Request Service Id**	**M**	**#14h**	**CDI**
	groupOfDiagnosticInformation (HighByte,LowByte)		#XXh,#XXh=	GODIN
	[	M,M	[	
#2d,#3d	Clear DTC´s in all Groups		#FFh,#FFh	
#2d,#3d	Clear Powertrain DTC		#FFh,#FBh	
	]		]	

Table ClearDiagnosticInformation Request Message

ClearDiagnosticInformation positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**clearDiagnosticInformation Positive Response SId**	**M**	**#54h**	**CDIPR**
	groupOfDiagnosticInformation (HighByte,LowByte)		#XXh,#XXh=	GODIN
	[	M,M	[	
#2d,#3d	Clear DTC´s in all Groups		#FFh,#FFh	
#2d,#3d	Clear Powertrain DTC		#FFh,#FBh	
	]		]	

Table ClearDiagnosticInformation Positive Response Message

ClearDiagnosticInformation negative Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	negativeResponse Service Id	M	#7Fh	NACK
#2d	clearDiagnosticInformation Request Service Id	M	#14h	CLRDTC
#3d	responseCode= [requestOutOfRange incorrectByteCountDuringBlockTransfer]	M	#XXh= [#31h #79h]	RC

Table ClearDiagnosticInformation Negative Response Message

7.16 SID 18h: readDTCByStatus Service

7.16.1 18h - read failure memory short - ReadDiagnosticTroubleCodesByStatus service

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_ERR_DYN_NR_OBD	V	0...FFH	0...255	1	[-]
Number of failures stored in dynamic memory (2 nd layer) with error class >=10h					

Input data:

ERR_TYPE_BYTE[NC_NR_ERR_DYN]	CTR_ERR_DYN_NR	ERR_DTC[NC_NR_ERR_DYN]	LV_IGK
C_ERR_CLAS_XX	NC_NR_ERR_DYN		

General information:

This function is to read out the numbers of faults in the error memory and the status of CDK (see also “Fault-type-byte and fault-type extension byte” (chapter OBDII functions)). Only failures with error class of **>= 10h** are considered.

Application conditions:

Initialisation:	at reset
Activation:	LV_IGK = 1
Deactivation:	LV_IGK = 0 Diagnostic timeout
Service location:	see "list of implemented diagnostic services"

Formula section:

CTR_ERR_DYN_NR_OBD = CTR_ERR_DYN_NR - ∑ errors with C_ERR_CLAS_XX < 10h

Example:

CTR_ERR_DYN_NR = 3

1) DIAG_INST[0] = TCO_EL   C_ERR_CLAS_TCO_EL   = 11H

2) DIAG_INST[1] = VS   C_ERR_CLAS_VS   = 5H

3) DIAG_INST[2] = CAT_DIAG_1   C_ERR_CLAS_CAT_DIAG_1   = 33H

=> CTR_ERR_DYN_NR_OBD = 3 – 1 = 2

If C_ERR_CLAS_XX from error [NC_NR_ERR_DYN] is calibrated with **>= 10h**

Then error is considered for SID18

Else error is not considered for SID18

Endif

ReadDiagnosticTroubleCodesByStatus Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadDiagnosticTroubleCodesByStatus Request SId**	**M**	**#18h**	**RCDKBS**
#2d	statusOfCDK - If the value in the telegram is not 02h, a value “00H” is given out on the reading of memory long;	M	#02h	SOCDK
	groupOfDiagnosticInformation (HighByte,LowByte)		#XXh,#XXh=	GOCDK
	[	M,M	[	
#3d,#4d	Read CDK´s in all Group`s		#FFh,#FFh	
#3d,#4d	Read CDK´s only powertrain faults		#FFh,#FBh	
	]		]	

Table ReadDiagnosticTroubleCodesByStatus Request Message

ReadDiagnosticTroubleCodesByStatus positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadDiagnosticTroubleCodesByStatus Pos. Response SId**	**M**	**#58h**	**RCDKBSPR**
#2d	numberOfCDK - CTR_ERR_DYN_NR_OBD	M	#01h-#XXh	#NROCDK
	ListOfCDKAndStatus=			LOCDKAS
	[		[	
#3d	CDK#0 high byte ERR_DTC[0]	C	#XXh	
#4d	CDK#0 low byte ERR_DTC[0]	C	#XXh	
#5d	statusOfCDK #1 ERR_TYPE_BYTE[0]	C	#XXh	
#6d	CDK#1 high byte ERR_DTC[1]	C	#XXh	
#7d	CDK#1 low byte ERR_DTC[1]	C	#XXh	
#8d	statusOfCDK #2 ERR_TYPE_BYTE[1]	C	#XXh	
.	.	.	.	
.	.	.	.	
.	.	.	.	
#22d	CDK#(NC_NR_ERR_DYN - 1)	C	#XXh	
#23d	CDK#(NC_NR_ERR_DYN - 1)	C	#XXh	
#24d	statusOfCDK #n ERR_TYPE_BYTE[NC_NR_ERR_DYN - 1]	C	#XXh	
	]		]	

Table ReadDiagnosticTroubleCodesByStatus Positive Response Message

ReadDiagnosticTroubleCodesByStatus negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**S**	**#7Fh**	**NR**
#2d	ReadDiagnosticTroubleCodesByStatus Request SId	M	#18h	RCDKBS
#3d	responseCode=	M	#XXh=	RC
	[		[	
	subFunctionNotSupported-invalidFormat,		#12h,	
	requestOutOfRange,		#31h,	
	general response codes		#XXh	
	]		]	

Table ReadDiagnosticTroubleCodesByStatus Negative Response Message

7.17 SID 17h: readStatusOfDTC Service

7.17.1 17h - Read Error Memory long version - ReadStatusOfDiagnosticTroubleCodes service

Input data:

ERR_DTC[NC_NR_ERR_DYN]	CTR_ERR_DYN_NR	ERR_TYPE_BYTE[NC_NR_ERR_DYN]	ERR_TYPE_EXT_BYTE[NC_NR_ERR_DYN]
CTR_FRC[NC_NR_ERR_DYN]	CTR_WUP_CYC[NC_NR_ERR_DYN]	ENVD_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN][NC_NR_FRF_SET][NC_NR_ERR_DYN]	ENVD_CUS_SET_SPC[NC_NR_ENVD_CUS_SET_SPC][NC_NR_FRF_SET][NC_NR_ERR_DYN]
LV_IGK			

General information:

This function allows to read out the numbers of CDK´s , list of CDK´s, the CDK status and the index of the environmental values. The index [x] represents the actual requested CDK (customer diagnostic key).

Application conditions:

Initialisation:	at reset
Activation:	LV_IGK = 1
Deactivation:	LV_IGK = 0 Diagnostic timeout
Service location:	ECU-SW

Formula section:

IF ERR_DTC[x] =0000h **or**

ERR_DTC[x] =FFFFh

THEN RESPONSE=NEGATIVE RESPONSE

ELSE RESPONSE=POSITIVE RESPONSE

readStatusOfDTC Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadStatusOfDiagnosticTroubleCodes Request SId**	**M**	**#17h**	**RSODTC**
	DiagnosticTroubleCode		#XXh,#XXh=	GODTC
	[		[	
#2d	CDK HighByte	M	#00h - #39h	
#3d	CDK LowByte	M	#01h - #99h	
	]		]	

Table ReadStatusOfDiagnosticTroubleCodes Request Message

readStatusOfDTC positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadStatusOfDiagnosticTroubleCodes Pos. Response SId**	**M**	**#57h**	**RSODTCPR**
#2d	numberOfDTC CTR_ERR_DYN_NR	M	#XXh	NRODTC
	listOfDTCAndStatus=			LODTCAS
	[		[	
#3d	Error code CDK (high byte) ERR_DTC[x]	M	#XXh	
#4d	Error code CDK (low byte) ERR_DTC[x]	M	#XXh	
#5d	statusOfDTC ERR_TYPE_BYTE[x]	M	#XXh	
#6d	extended StatusOfDTC ERR_TYPE_EXT_BYTE[x]	M	#XXh	
#7d	P - code DTC (high byte) ACTION_ERRM_ReadDTCLevelByDTCLevel	M	#XXh	
#8d	P - code DTC (low byte) ACTION_ERRM_ReadDTCLevelByDTCLevel	M	#XXh	
#9d	frequency counter CTR_FRC[x]	M	#XXh	
#10d	warmup counter CTR_WUP_CYC[x]	M	#XXh	
#11d	km value by first occur (high byte) ENVD_CUS_SET_CMN[2][1]	M	#XXh	
#12d	km value by first occur (low byte) ENVD_CUS_SET_CMN[3][1]	M	#XXh	
#13d	1 environmental value first occur ENVD_CUS_SET_SPC[1][1]	M	#XXh	
#14d	2 environmental value first occur ENVD_CUS_SET_SPC[2][1]	M	#XXh	
#15d	3 environmental value first occur ENVD_CUS_SET_SPC[3][1]	M	#XXh	
#16d	4 environmental value first occur ENVD_CUS_SET_SPC[4][1]	M	#XXh	
#17d	km value by second occur (high byte) ENVD_CUS_SET_CMN[2][2]	M	#XXh	
#18d	km value by second occur (low byte) ENVD_CUS_SET_CMN[3][2]	M	#XXh	
#19d	1 environmental value second occur ENVD_CUS_SET_SPC[1][2]	M	#XXh	
#20d	2 environmental value second occur ENVD_CUS_SET_SPC[2][2]	M	#XXh	
#21d	3 environmental value second occur ENVD_CUS_SET_SPC[3][2]	M	#XXh	
#22d	4 environmental value second occur ENVD_CUS_SET_SPC[4][2]	M	#XXh	
#23d	km value by third occur (high byte) ENVD_CUS_SET_CMN[2][3]	M	#XXh	
#24d	km value by third occur (low byte) ENVD_CUS_SET_CMN[3][3]	M	#XXh	
#25d	1 environmental value third occur ENVD_CUS_SET_SPC[1][3]	M	#XXh	
#26d	2 environmental value third occur ENVD_CUS_SET_SPC[2][3]	M	#XXh	
#27d	3 environmental value third occur ENVD_CUS_SET_SPC[3][3]	M	#XXh	
#28d	4 environmental value third occur ENVD_CUS_SET_SPC[4][3]	M	#XXh	
	]		]	

Table ReadStatusOfDiagnosticTroubleCodes Positive Response Message if the error is entered in error memory

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadStatusOfDiagnosticTroubleCodes Pos. Response SId**	**M**	**#57h**	**RSODTCPR**
#2d	numberOfDTC CTR_ERR_DYN_NR	M	#01h	NRODTC
	listOfDTCAndStatus=			LODTCAS
	[		[	
#3d	Error code CDK (high byte)	M	#XXh	
#4d	Error code CDK (low byte)	M	#XXh	
#5d	CDK not entered or not defined	M	#00h	
	]		]	

Table ReadStatusOfDiagnosticTroubleCodes Positive Response Message Message if the error is not entered in error memory

readStatusOfDTC negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**S**	**#7Fh**	**NR**
#2d	ReadStatusOfDiagnosticTroubleCodes Request SId	M	#17h	RSODTC
#3d	responseCode= [subFunctionNotSupported-invalidFormat, incorrectByteCountDuringBlockTransfer]	M	#XXh= [#12h, #79h]	RC

Table ReadStatusOfDiagnosticTroubleCodes Negative Response Message

Request and Response (Example)

This table lists the possible responses depending on the requestByFunction or requestByDTC. The CDK values 0xFFFF & 0x0000 are reserved for special purposes(not yet defined now).

Tester Request	*ECU response*
CDK value = 0x0000 or 0xFFFF	• Negative response: Sub-Function Not Supported Or Invalid Format
CDK of a failure which is entered in dynamic layer	• Positive response • number of CDKs entered in fsp_sequ and with matching CDK, i.e. 1 • CDK high • CDK low • statusOfCDK • extended StatusOfDTC • frequency counter • warmup counter • ....
CDK which doesn’t fall under any of the previous cases	• Positive Response • number of CDKs • CDK high • CDK low • CDK low • 0

7.18 SID 1Ah: readECUIdentification Service

7.18.1 1Ah - readECUIdentification Service

Input data:

LV_IGK	VIN_SHO[7]	C_IDX_COD_CONV	

General information:

This function reads out from the ECU specific logistic values.

Application conditions:

Initialisation: at reset

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0
Diagnostic timeout

Service location: see "list of implemented diagnostic services"

Formula section:

ReadEcuIdentification Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readEcuIdentification Request Service Id**	**M**	**#1Ah**	**REI**
#2d	identificationOption	M	#XXh	IO

Table readEcuIdentification Request Message

ReadEcuIdentification positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readEcuIdentification Request Service Id**	**M**	**#5Ah**	**REI**
#2d	identificationOption	M	#XXh	IO
#3d	Data 1 for identification option	M	#XXh	
#3d + 1	Data 2 for identification option			
#3d + n	Data n for identification option	M	#XXh	

Table ReadEcuIdentification Positive Response Message

ReadEcuIdentification negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	readEcuIdentification Request Service Id	M	#1Ah	REI
#3d	responseCode= [general response code ECU ID not supported]	M	#XXh= [#XXh #12]	RC

Table ReadEcuIdentification Negative Response Message

7.18.1.1 Supported identification options

Identification Option nr.	**SW name**	**Size / byte**
ECUIDT 80 h Ecu Identification Data Table Info: 91h: VMECUHN only 1x output	Reading of the Identification options: 91h,9Ah,9Bh,9Ch,97h,9Dh,9Eh,9FH exactly in this order	Compare the single identification options
CUIFDT 86 h actual BMW user information field	bmw_uif	32 blocks a 64 byte
PECUHN 87 h BMW physical HW number	kwp_io_pecuhn	18
SSECUSEN 89 h systemSupplierECUSerialNumber	serial_number	9
CRSCOTSN 8A h Calib.RepairShopCodeOrTesterSerialNr	NC_KWP_IO_CRSCOTSN	10
CPD 8B h Calib.ProgrammingDate	NC_KWP_IO_CPD	4
SSBSN 8C h SubSystemBootSoftwareNumber	sub_sys_boot_sw_nr	12
SSESN 8D h SubSystemECUSoftwareNumber	sub_sys_ecu_sw_nr	12
SSCN 8E h SubSystemCalibrationNumber	cal_customer_if	12
LDSA 8F h reportLogisticDataStartAddress	NC_KWP_IO_LDSA	8
VIN short 90 h short VehicleIdentificationNumber	VIN_SHO[7]	7
VMECUHN 91 h VehicleManufactureECUHWnumber VMECUHN 3 x output	bmw_hw_nr	18
EROTAN 96 h exhaustRegulationOrTypeApprovalnumber	kwp_io_erotan	6
SNOET 97 h systemNameOrEngineType	kwp_io_snoet kwp_io_snoet_cal (if ECU SW is active)	2
RSCOTSN 98 h RepairShopCodeOrTesterSerialNumber	kwp_io_rscatsn	8
PD 99 h	kwp_io_pd	4

ProgrammingDate		
VMECUHVN 9A h vehicleManufacturerECUHWversionNo.	kwp_io_vmecuhvn	1
VMCI 9B h vehicleManufacturerCodingIndex	C_IDX_COD_CONV	1
VMDI 9C h vehicleManufacturerDiagnosticIndex	kwp_io_vmdi	2
DOECUM 9D h dateOfECUManufacturing	date_of_final_test	4
SSI 9E h systemSupplierndex	supplier_index	1
VMECUSLVN 9F h VMECUSoftwareLayerVersionNumber	bmw_sw_nr	12

7.19 SID 21h: readDataByLocalIdentifier Service

General information:

All implemented SID 21h services are described below.

Formula section:

7.19.1 Global negative Responses

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	readDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	responseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

Remark:
The detailed message of the negative response code is described in chapter "Introduction".

7.19.2 21_01 - read system specific adress - ReadDataByLocalIdentifier service

General information:

This function allows to read out the system specific adresses. They are necessary e.g. reprogramming the ECU with the NPS.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read system specific adress	M	#01h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#2d	recordLocalIdentifier= system specific adress	M	#01h	RLI_UEB
#3d	RecordValue #1high byte = Programming voltage 0V	M	#XXh	Not supported
#4d	RecordValue #2 middle byte= Programming voltage 0V	M	#XXh	Not supported
#5d	RecordValue #3 low byte = Programming voltage 0V	M	#XXh	Not supported
#6d	RecordValue #4high byte = Dummy for DP1	M	#FFh	
#7d	RecordValue #5 middle byte = Dummy for DP1	M	#FFh	
#8d	RecordValue #6 low byte = Dummy for DP1	M	#FFh	
#9d	RecordValue #7high byte = Dummy for DP2	M	#FFh	
#10d	RecordValue #8 middle byte = Dummy for DP2	M	#FFh	
#11d	RecordValue #9 low byte = Dummy for DP2	M	#FFh	
#12d	RecordValue #10 high byte = Dummy for DP3	M	#FFh	
#13d	RecordValue #11 middle byte = Dummy for DP3	M	#FFh	
#14d	RecordValue #12 low byte = Dummy for DP3	M	#FFh	
#15d	RecordValue #13 high byte = Dummy for DP4	M	#FFh	
#16d	RecordValue #14 middle byte = Dummy for DP4	M	#FFh	
#17d	RecordValue #15 low byte = Dummy for DP4	M	#FFh	
#18d	RecordValue #16 high byte = Dummy for DP5	M	#FFh	
#19d	RecordValue #17 middle byte = Dummy for DP5	M	#FFh	
#20d	RecordValue #18 low byte = Dummy for DP5	M	#FFh	
#21d	RecordValue #19 high byte = Dummy for DP6	M	#FFh	
#22d	RecordValue #20 middle byte = Dummy for DP6	M	#FFh	
#23d	RecordValue #21 low byte = Dummy for DP6	M	#FFh	
#24d	RecordValue #22 high byte = Dummy for DP7	M	#FFh	
#25d	RecordValue #23 middle byte = Dummy for DP7	M	#FFh	
#26d	RecordValue #24 low byte = Dummy for DP7	M	#FFh	
#27d	RecordValue #25 high byte = Dummy for DP8	M	#FFh	
#28d	RecordValue #26 middle byte = Dummy for DP8	M	#FFh	
#29d	RecordValue #27 low byte = Dummy for DP8	M	#FFh	
#30d	RecordValue #28 high byte = Dummy for DP9	M	#FFh	
#31d	RecordValue #29 middle byte = Dummy for DP9	M	#FFh	
#32d	RecordValue #30 low byte = Dummy for DP9	M	#FFh	
#33d	RecordValue #31 high byte = Dummy for DP10	M	#FFh	
#34d	RecordValue #32 middle byte = Dummy for DP10	M	#FFh	
#35d	RecordValue #33 low byte = Dummy for DP10	M	#FFh	
#36d	RecordValue #34 high byte = Dummy for DP11	M	#FFh	
#37d	RecordValue #35 middle byte = Dummy for DP11	M	#FFh	
#38d	RecordValue #36 low byte = Dummy for DP11	M	#FFh	
#39d	RecordValue #37 high byte = Dummy for DP12	M	#FFh	
#40d	RecordValue #38 middle byte = Dummy for DP12	M	#FFh	
#41d	RecordValue #39 low byte = Dummy for DP12	M	#FFh	
#42d	RecordValue #40 high byte = Dummy for DSK	M	#FFh	
#43d	RecordValue #41 middle byte = Dummy for DSK	M	#FFh	
#44d	RecordValue #42 low byte = Dummy for DSK	M	#FFh	
#45d	RecordValue #43 high byte = Dummy for WCODE	M	#FFh	
#46d	RecordValue #44 middle byte = Dummy for WCODE	M	#FFh	
#47d	RecordValue #45 low byte = Dummy for WCODE	M	#FFh	
#48d	RecordValue #46 high byte = cal_ref_nr data reference	M	#XXh	
#49d	RecordValue #47 middle byte = cal_ref_nr data reference	M	#XXh	
#50d	RecordValue #48 low byte = cal_ref_nr data reference	M	#XXh	
#51d	RecordValue #49 high byte = cif_backup	M	#XXh	
#52d	RecordValue #50 middle byte = cif_backup	M	#XXh	
#53d	RecordValue #51 low byte = cif_backup	M	#XXh	

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#54d	RecordValue #52 high byte = del_time_max erasing time	M	#XXh	
#55d	RecordValue #53 middle byte = del_time_max erasing time	M	#XXh	
#56d	RecordValue #54 low byte = del_time_max erasing time	M	#XXh	
#57d	RecordValue #55 high byte = hw_reference	M	#XXh	
#58d	RecordValue #56 middle byte = hw_reference	M	#XXh	
#59d	RecordValue #57 low byte = hw_reference	M	#XXh	
#60d	RecordValue #58 high byte = ecu_sw_reference programm-reference	M	#XXh	
#61d	RecordValue #59 middle byte = ecu_sw_reference programm-reference	M	#XXh	
#62d	RecordValue #60 low byte = ecu_sw_reference programm-reference	M	#XXh	
#63d	RecordValue #61 high byte = bmw_uif user info field	M	#XXh	
#64d	RecordValue #62 middle byte = bmw_uif user info field	M	#XXh	
#65d	RecordValue #63 low byte = bmw_uif user info field	M	#XXh	
#66d	RecordValue #64 high byte = kwp_buf_len_max max. length of data	M	#XXh	
#67d	RecordValue #65 middle byte = kwp_buf_len_max max. length of data	M	#XXh	
#68d	RecordValue #66 low byte = kwp_buf_len_max max. length of data	M	#XXh	
#69d	RecordValue #67 high byte = date_of_final_test	M	#XXh	
#70d	RecordValue #68 middle byte = date_of_final_test	M	#XXh	
#71d	RecordValue #69 low byte = date_of_final_test	M	#XXh	
#72d	RecordValue #70 high byte = Dummy for the Baud rate table	M	#FFh	
#73d	RecordValue #71 middle byte = Dummy for the Baud rate table	M	#FFh	
#74d	RecordValue #72 low byte = Dummy for the Baud rate table	M	#FFh	

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.3 21_02 - read overtemperature protection - ReadDataByLocalIdentifier service

Input data:

LV_IGK	TOIL_MAX_WARN	VS_TOIL_MAX	N_TOIL_MAX
GEAR_TOIL_MAX	TQI_AV_TOIL_MAX	TAM_TOIL_MAX	TCO_TOIL_MAX
CTR_TOIL_MAX	PV_AV_TOIL_MAX	DIST_TOIL_MAX	LV_CS_TOIL_MAX
TOIL_THD_TOIL_MAX			

General information:

Exact description of the Function see file 14400S02.

Application conditions:

Initialisation: at reset the values from the NVMY

Activation: LV_IGK=1

Deactivation: LV_IGK=0
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read overtemperature protection	M	#02h	RLI_UET

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read overtemperature protection	M	#02h	RLI_UEB
#3d	RecordValue # = TOIL_MAX_WARN	M	#XXh	RV_
#4d	RecordValue # = VS_TOIL_MAX	M	#XXh	RV_
#5d	RecordValue # = N_TOIL_MAX high byte	M	#XXh	RV_
#6d	RecordValue # = N_TOIL_MAX low byte	M	#XXh	RV_
#7d	RecordValue # = GEAR_TOIL_MAX	M	#XXh	RV_
#8d	RecordValue # = TQI_AV_TOIL_MAX high byte	M	#XXh	RV_
#9d	RecordValue # = TQI_AV_TOIL_MAX low byte	M	#XXh	RV_
#10d	RecordValue # = TAM_TOIL_MAX	M	#XXh	RV_
#11d	RecordValue # = CTR_TOIL_MAX	M	#XXh	RV_
#12d	RecordValue # = TCO_TOIL_MAX	M	#XXh	RV_
#13d	RecordValue # = DIST_TOIL_MAX high byte	M	#XXh	RV_
#14d	RecordValue # = DIST_TOIL_MAX low byte	M	#XXh	RV_
#15d	RecordValue # = PV_AV_TOIL_MAX	M	#XXh	RV_
#16d	RecordValue # = LV_CS_TOIL_MAX	M	#XXh	RV_
#17d	RecordValue # = TOIL_THD_TOIL_MAX	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.4 21_03 - read engine overspeed detection - ReadDataByLocalIdentifier service

Input data:

N_MAX	TRT_N_MAX	CTR_N_MAX	CTR_KM_N_MAX
N_GRD_N_MAX	VS_N_MAX	PV_AV_N_MAX	GEAR_EF_N_MAX
LV_CS_N_MAX	T_SUM_N_MAX	T_N_MAX	NR_PAT_SCC_N_MAX

General information:

Exact description of the Function see file 14A06301.

Detection of an overreving event and non-volatile storage of:

An overreving event starts at exceeding the engine speed C_N_LIM_MIN and active engine speed limitation (LV_N_MAX = 1).

At each overreving event, the event frequency counter is incremented by 1. The highest engine speed reached N_MAX and the associated total running time in case of highest engine speed reached TRT_N_MAX is only overwritten if a higher engine speed was reached.

Application conditions:

Initialisation: at reset the values from the NVMY

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read engine overspeed detection	M	#03h	RLI_UEB

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read engine overspeed detection	M	#03h	RLI_UEB
#3d	RecordValue # = N_MAX high byte	M	#XXh	RV_
#4d	RecordValue # = N_MAX low byte	M	#XXh	RV_
#5d	RecordValue # = TRT_N_MAX high word high byte	M	#XXh	RV_
#6d	RecordValue # = TRT_N_MAX high word low byte	M	#XXh	RV_
#7d	RecordValue # = TRT_N_MAX low word high byte	M	#XXh	RV_
#8d	RecordValue # = TRT_N_MAX low word low byte	M	#XXh	RV_
#9d	RecordValue # = CTR_N_MAX	M	#XXh	RV_
#10d	RecordValue # = CTR_KM_N_MAX high byte	M	#XXh	RV_
#11d	RecordValue # = CTR_KM_N_MAX low byte	M	#XXh	RV_
#12d	RecordValue # = N_GRD_N_MAX	M	#XXh	RV_
#13d	RecordValue # = VS_N_MAX	M	#XXh	RV_
#14d	RecordValue # = PV_AV_N_MAX	M	#XXh	RV_
#15d	RecordValue # = GEAR_EF_N_MAX	M	#XXh	RV_
#16d	RecordValue # = LV_CS_N_MAX	M	#XXh	RV_
#17d	RecordValue # = T_SUM_N_MAX high byte	M	#XXh	RV_
#18d	RecordValue # = T_SUM_N_MAX low byte	M	#XXh	RV_
#19d	RecordValue # = T_N_MAX high byte	M	#XXh	RV_
#20d	RecordValue # = T_N_MAX low byte	M	#XXh	RV_
#21d	RecordValue # = NR_PAT_SCC_N_MAX	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.5 21_05 - read readiness flags - ReadDataByLocalIdentifier service

Input data:

LV_IGK	C_STATE_READY_OBD_2		

Import actions:

ACTION_ERRM_ReadReadinessCode

General information:

With this comand the readiness flags described in the chapter “Readiness codes” are shown.

The flags (bit) are default 1 and when the diagnosis is done well, the flag is switched to 0.

Description:

For reading readiness code see chapter “OBD Error management – Communication interface”.

Application conditions:

Initialisation: at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: ECU-SW

Formula section:

```
IF          service 21_05 is received.....
THEN
ACTION_ERRM_ReadReadinessCode(STATE_READY_OBD_1, STATE_READY_OBD_2)
The values
STATE_READY_OBD_1          XXh
STATE_READY_OBD_2          XXh
will be sent from the ECU to the tester
ENDIF
```

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read readiness	M	#05h	RLI_REA

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service with RLI = #05h

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier	M	#05h	RLI_REA
#3d	RecordValue # 1 STATE_READY_OBD_1	M	#XXh	RV_EVS
#4d	RecordValue # 2 STATE_READY_OBD_2	M	#XXh	RV_REA
#5d	RecordValue # 3 C_STATE_READY_OBD_2	M	#XXh	RV_REA

Table readDataByLocalIdentifier Positive Response Message

Byte #	Name	Size	Conversion (hex/bin)	Conversion (physical)	Resol.
#3d	RecordValue # 1 = STATE_READY_OBD_1	1	xxxxxxxX b	Missfire monitoring supported	
			xxxxxxXx b	Fuel system monitoring supported	
			xxxxxXxx b	Comprehensive component monitoring supported	
			xxxxXxxx b	Not used	
			xxxXxxxx b	Missfire monitoring status	
			xxXxxxxx b	Fuel system monitoring status	
			xXxxxxxx b	Comprehensive component monitoring status	
			Xxxxxxxx b	Not used	

Byte #	Name	Size	Conversion (hex/bin)	Conversion (physical)	Resol.
#4d	RecordValue # 2 = STATE_READY_OBD_2	1	xxxxxxxX b	Catalyst monitoring	
			xxxxxxXx b	Heated catalystComprehensive	
			xxxxxXxx b	Evaporative system monitoring	
			xxxxXxxx b	Secundary air system monitoring	
			xxxXxxxx b	A/C system refigerant monitoring	
			xxXxxxxx b	Oxygen sensor monitoring	
			xXxxxxxx b	Oxygen sensor heater monitoring	
			Xxxxxxxx b	EGR system monitorimg	

Table readDataByLocalIdentifier Positive Response Message Bit information

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.6 21_06 - receiving status of immobilizer - ReadDataByLocalIdentifier service

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_IMOB	V/O	0...FFH	0...255	1	[-]
State of the immobilizer - only updated when service ist requested					

General information:

This function gives the actual status of the immobilizer back. The ecu reports to the tester whether the new start value is accepted or if the actual saved start value is identical with the one from the immobilizer. If there is a problem, the error information during synchronisation is also reported to the tester.

The ecu answers the question as long with the respective status until for the first time the answer is 01 or 02. When this answer is received, the ecu can activate this answer again after a Kl.15 change and the ecu power latch is over.

Parameter 04 “process is running” in the positive response switch after timeout occur to 03” Interface failure D_EWS: frame or parity or no signal (timeout)”

In case of an destroyed IMOB CODE STATUS the positive response should be 21h, the same as for -2-out-of-3 start values in the flash memory not OK;

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read status of receiving immobilizer	M	#06h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier	M	#06h	RLI_
#3d	RecordValue # 1 = STATE_IMOB [-start value programming or reset to start value was successful -wrong start value at reset attempt (Immob. and ECU do not match) -telegram contents was no start value (perhaps rolling code) -Interface failure D_EWS: frame or parity or no signal (timeout) -process is running “ busy” -programming of start value or reset to start value doesn´t happen in this Kl.15 cycle -same random number as by the last reset to start value despite of already done code switching -no start value programmed -2-out-of-3 start values in the flash memory not OK (only when UC is programmed e.g. after a power latch phase) or the IMOB_CODE_STATUS is destroyed]	M	#XXh [#00h, #01h, #02h, #03h, #04h, #05h, #06h, #07h, #21h,]	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.7 21_07 - read cruise control status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	STATE_MSW_CAN	STATE_CRU_OFF_IRR	STATE_CRU_OFF_REV

General information:

With this comand the status described in the chapter “Vehicle speed control” are distributed.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on (LV_IGK=1)

Deactivation: Kl.15 off (LV_IGK=0)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**

#2d	recordLocalIdentifier = read cruise control status	M	#07h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= cruise control status	M	#07h	RLI_
#3d	RecordValue # 1 = STATE_MSW_CAN	M	#XXh	RV_
#4d	RecordValue # 2 = STATE_CRU_OFF_IRR high byte	M	#XXh	RV_
#5d	RecordValue # 3 = STATE_CRU_OFF_IRR low byte	M	#XXh	RV_
#6d	RecordValue # 4 = STATE_CRU_OFF_REV high byte	M	#XXh	RV_
#7d	RecordValue # 5 = STATE_CRU_OFF_REV low byte	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.8 21_08 - release of multiple start value programming - ReadDataByLocalIdentifier service

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
RLS_MPL_VALUE_PROG	V/O	0...5H	0...5	1	[-]
release of multiple start value programming - only updated when service ist requested					

Input data:

LV_IGK	TRT		

General information:

This function gives the actual status of release of multiple start value programming.
This service should give always a response as well if the total running time t>90min!

If the response is sent to the ECU by TRT>90 minutes then a negative response “7F_21_22-conditions not correct” occur.

Meaning of RLS_MPL_VALUE_PROG:

RLS_MPL_VALUE_PROG=00h – release of multiple start value programming available
RLS_MPL_VALUE_PROG=01h – release of multiple start value programming not available
RLS_MPL_VALUE_PROG=04h – relase process for start value programming is going on
RLS_MPL_VALUE_PROG=05h – no release process for start value programming done in this cycle

Application conditions:

Initialisation: RLS_MPL_VALUE_PROG=00h at reset

Activation: LV_IGK=1 (Kl.15 on) and

Authenticity check and
TRT<90min

***Deactivation**:* LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = release of multiple start value programming	M	#08h	RLI_FEWS

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	RecordLocalIdentifier= release of multiple start value programming	M	#08h	RLI_FEWS
#3d	RecordValue # 1 = RLS_MPL_VALUE_PROG [- release of multiple start value programming available - release of multiple start value programming not available - release process for start value programming is going on - no release process for start value programming done in this cycle]	M	#XXh [#00h, #01h, #04h, #05h]	RV_FEWS

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.9 21_09 - read DIST_ACT_MIL - ReadDataByLocalIdentifier service

Input data:

LV_IGK	DIST_ACT_MIL		

General information:

This funktion reports the value of the “distance MIL is active”.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier DIST_ACT_MIL	M	#09h	RLI_DIST_ACT_MIL

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier DIST_ACT_MIL	M	#09h	RLI_
#3d	RecordValue = DIST_ACT_MIL high byte high word	M	#XXh	RV_S_
#4d	RecordValue = DIST_ACT_MIL low byte high word	M	#XXh	RV_S_
#5d	RecordValue = DIST_ACT_MIL high byte low word	M	#XXh	RV_S_
#6d	RecordValue = DIST_ACT_MIL low byte low word	M	#XXh	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.10 21_0A - read Error Memory with freeze frame data extra long - ReadDataByLocalIdentifier service

Input data:

ERR_DTC[NC_NR_ERR_DYN]	ERR_TYPE_BYTE[NC_NR_ERR_DYN]	ERR_TYPE_EXT_BYTE[NC_NR_ERR_DYN]	CTR_FRC[NC_NR_ERR_DYN]
CTR_DC[NC_NR_ERR_DYN]	ENVD_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN][NC_NR_FRF_SET][NC_NR_ERR_DYN]	ENVD_CUS_SET_SPC[NC_NR_ENVD_CUS_SET_SPC][NC_NR_FRF_SET][NC_NR_ERR_DYN]	C_ERR_CLAS_XX
LV_IGK	CTR_WUP_CYC[NC_NR_ERR_DYN]	ENVD_OBD[NC_NR_ENVD_OBD][NC_NR_ERR_DYN]	ENVD_CUS_CMN[NC_NR_ENVD_CUS_CMN][NC_NR_ERR_DYN]
CTR_ERR_DYN_NR			

General information:

This function is created to read out the failure memory with all the errors including the freeze frame data. The index of the environment values allows a classification to the list of the environment values from BMW.

Record Common Identifier table usage and reading:

- RCI usage as BYTE Values (Freeze Frame, Umweltbedingung, DDLI etc.) should support the full range 0x00 to 0xFF as VALID. A value that is not supported will be transmitted as 0xFF

- RCI usage as WORD Values (DDLI etc.) should support the range 0x0000 to 0xFFFE as VALID. Invalid is shown with 0xFFFF.

This should enable the full range of internal Byte values to be used with no compromise, but still allow the diagnostic equipment to recognise if a value is available or not.

Application conditions:

Initialisation: at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

IF ERR_DTC[x] = 0000h **or**

ERR_DTC[x] = FFFFh

THEN RESPONSE=NEGATIVE RESPONSE

ELSE RESPONSE=POSITIVE RESPONSE

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = Read Error Memory with freeze frame data	M	#0Ah	RLI_
#3d	Value = Error code ERR_DTC[x] high byte	M	#XXh	RV_
#4d	Value = Error code ERR_DTC[x] low byte	M	#XXh	RV_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier	M	#0Ah	RLI_
#3d	RecordValue # 1 = Error counter total CTR_ERR_DYN_NR	M	#XXh	RV_
#4d	RecordValue # 2 = Error code ERR_DTC[x] high byte	M	#XXh	RV_
#5d	RecordValue # 3 = Error code ERR_DTC[x] low byte	M	#XXh	RV_
#6d	RecordValue # 4 = status DTC#1 ERR_TYPE_BYTE[x]	M	#XXh	RV_
#7d	RecordValue # 5 = Cycle flag ERR_TYPE_EXT_BYTE[x]	M	#XXh	RV_
#8d	RecordValue # 8 = Frequency counter CTR_FRC[x]	M	#XXh	RV_
#9d	RecordValue # 9 = Error recover counter (HLC) CTR_DC[x]	M	#XXh	RV_
#10d	RecordValue # 10 = km-value by first occur high byte ENVD_CUS_SET_CMN[2][1]	M	#XXh	RV_

#11d	RecordValue # 11 = km-value by first occur low byte ENVD_CUS_SET_SPC[3][1]	M	#XXh	RV_
#12d	RecordValue # 12 = 1.environmental value first occur ENVD_CUS_SET_SPC[1][1]	M	#XXh	RV_
#13d	RecordValue # 13 = Index for 1.environmental value	M	#XXh	RV_
#14d	RecordValue # 14 = 2.environmental value first occur ENVD_CUS_SET_SPC[2][1]	M	#XXh	RV_
#15d	RecordValue # 15 = Index for 2.environmental value	M	#XXh	RV_
#16d	RecordValue # 16= 3.environmental value first occur ENVD_CUS_SET_SPC[3][1]	M	#XXh	RV_
#17d	RecordValue # 17 = Index for 3.environmental value	M	#XXh	RV_
#18d	RecordValue # 18 = 4.environmental value first occur ENVD_CUS_SET_SPC[4[1]	M	#XXh	RV_
#19d	RecordValue # 19 = Index for 4.environmental value	M	#XXh	RV_
#20d	RecordValue # 20 = km-value by second occur high byte ENVD_CUS_SET_SPC[2][2]	M	#XXh	RV_
#21d	RecordValue # 21= km-value by second occur low byte ENVD_CUS_SET_SPC[3][2]	M	#XXh	RV_
#22d	RecordValue # 22 = 1.environmental value second occur ENVD_CUS_SET_SPC[1][2]	M	#XXh	RV_
#23d	RecordValue # 23 = 2.environmental value second occur ENVD_CUS_SET_SPC[2][2]	M	#XXh	RV_
#24d	RecordValue # 24 = 3.environmental value second occur ENVD_CUS_SET_SPC[3][2]	M	#XXh	RV_
#25d	RecordValue # 25 = 4.environmental value second occur ENVD_CUS_SET_SPC[4][2]	M	#XXh	RV_
#26d	RecordValue # 26 = km-value by last occur high byte ENVD_CUS_SET_SPC[2][3]	M	#XXh	RV_
#27d	RecordValue # 27 = km-value by last occur low byte ENVD_CUS_SET_SPC[3][3]	M	#XXh	RV_
#28d	RecordValue # 28 = 1.environmental value last occur ENVD_CUS_SET_SPC[1][3]	M	#XXh	RV_
#29d	RecordValue # 29= 2.environmental value last occur ENVD_CUS_SET_SPC[2][3]	M	#XXh	RV_
#30d	RecordValue # 30= 3.environmental value last occur ENVD_CUS_SET_SPC[3][3]	M	#XXh	RV_
#31d	RecordValue # 31 = 4.environmental value last occur ENVD_CUS_SET_SPC[4][3]	M	#XXh	RV_
#32d	RecordValue # 32= Error class C_ERR_CLAS_XX	M	#00h	RV_
#33d	RecordValue # 33 = Importance counter TSF (not def. yet)	M	#00h	RV_
#34d	RecordValue # 34 = Debounce counter (MIL on) FLC FLC = DC_MAX_XX - CTR_DC[x]	M	#XXh	RV_
#35d	RecordValue # 35= Deleting counter DLC CTR_WUP_CYC[x]	M	#XXh	RV_
#36d	RecordValue # 36 = Freeze frame data 0 – lambda control state bench 1 ENVD_OBD[1][NC_NR_ERR_DYN]	M	#XXh	RV_
#37d	RecordValue # 37 = Freeze frame data 1 – lambda control state bench 2 ENVD_OBD[2][NC_NR_ERR_DYN]	M	#XXh	RV_
#38d	RecordValue # 38 = Freeze frame data 2 – relative air mass ENVD_OBD[3][NC_NR_ERR_DYN]	M	#XXh	RV_
#39d	RecordValue # 39 = Freeze frame data 3 – engine coolent temperature ENVD_OBD[4][NC_NR_ERR_DYN]	M	#XXh	RV_
#40d	RecordValue # 40 = Freeze frame data 4 – variable factor bench 1 ENVD_OBD[5][NC_NR_ERR_DYN]	M	#XXh	RV_
#41d	RecordValue # 41 = Freeze frame data 5 – adaptions factor bench 1 ENVD_OBD[6][NC_NR_ERR_DYN]	M	#XXh	RV_

#42d	RecordValue # 40 = Freeze frame data 6 - variable factor bench 2 ENVD_OBD[7][NC_NR_ERR_DYN]	M	#XXh	RV_
#43d	RecordValue # 43 = Freeze frame data 7 - adaptions factor bench 2 ENVD_OBD[8][NC_NR_ERR_DYN]	M	#XXh	RV_
#44d	RecordValue # 44= Freeze frame data 8 – intake air pressure ENVD_OBD[10][NC_NR_ERR_DYN]	M	#XXh	RV_
#45d	RecordValue # 45 = Freeze frame data 9 – engine speed ENVD_CUS_CMN[1][NC_NR_ERR_DYN]	M	#XXh	RV_
#46d	RecordValue # 46 = Freeze frame data 10 – vehicle speed ENVD_CUS_CMN[13][NC_NR_ERR_DYN]	M	#XXh	RV_
#47d	RecordValue # 6 = P code DTC " high byte ACTION_ERRM_ReadDTCLevelByDTCLevel	M	#XXh	RV_
#48d	RecordValue # 7 = P code DTC " low byte ACTION_ERRM_ReadDTCLevelByDTCLevel	M	#XXh	RV_
#49d	RecordValue # 49 = Checksum for each error	M	#00h	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.11 21_25 - read INH_IV_KWP - ReadDataByLocalIdentifier service

Input data:

LV_IGK	INH_IV_KWP		

General information:

This funktion reports the value of the "switch off pattern of the injection valves".

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of "implemented diagnostic services"

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier INH_IV_KWP	M	#25h	RLI_INH_IV_KWP

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier INH_IV_KWP	M	#25h	RLI_

#3d	RecordValue = INH_IV_KWP	M	#XXh	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.12 21_36 - read air-conditioner status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	LV_ACCOUT_RLY		

General information:

With this comand the LV to activate the ACC-relay is distributed.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on (LV_IGK=1)

Deactivation: Kl.15 off (LV_IGK=0) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read air-conditioner status	M	#36h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read air-conditioner status	M	#36h	RLI_
#3d	RecordValue # 1 = LV_ACCOUT_RLY	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.13 21_3F - read idle speed status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	LV_IS		

General information:

With this comand the status of the Idle Speed is hand over.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1

Deactivation: LV_IGK=0
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read Idle Speed “Switch” status - LV_IS	M	#3Fh	RLI_IS

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier	M	#3Fh	RLI_
#3d	RecordValue = LV_IS	M	#XXh	RV_IS

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.14 21_41 - read LSH 4 status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	STATE_LSH_DOWN[NC_CBK_EX_NR]		

General information:

With this comand the status of the lambda sensor heating 4 is distributed.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read LSH 4 status	M	#41h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read LSH 4 status	M	#41h	RLI_
#3d	RecordValue # 1 = STATE_LSH_DOWN[2]	M	#XXh	RV_
#4d	RecordValue # 2 = dummy value fix	M	#FFh	RV_
#5d	RecordValue # 3 = dummy value fix	M	#FFh	RV_
#6d	RecordValue # 4 = dummy value fix	M	#FFh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.15 21_42 - read LSH 3 status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	STATE_LSH_DOWN[NC_CBK_EX_NR]		

General information:

With this comand the status of the lambda sensor heating 3 is distributed.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read LSH 3 status	M	#42h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read LSH 3 status	M	#42h	RLI_
#3d	RecordValue # 1 = STATE_LSH_DOWN[1]	M	#XXh	RV_
#4d	RecordValue # 2 = dummy value fix	M	#FFh	RV_
#5d	RecordValue # 3 = dummy value fix	M	#FFh	RV_
#6d	RecordValue # 4 = dummy value fix	M	#FFh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.16 21_43 - read LSH 2 status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	STATE_LSH_UP[NC_CBK_EX_NR]		

General information:

With this comand the status of the lambda sensor heating 2 is distributed.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read LSH 2 status	M	#43h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read LSH 2 status	M	#43h	RLI_
#3d	RecordValue # 1 = STATE_LSH_UP[2]	M	#XXh	RV_
#4d	RecordValue # 2 = dummy value fix	M	#FFh	RV_
#5d	RecordValue # 3 = dummy value fix	M	#FFh	RV_
#6d	RecordValue # 4 = dummy value fix	M	#FFh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.17 21_44 - read LSH 1 status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	STATE_LSH_UP[NC_CBK_EX_NR]		

General information:

With this comand the status of the lambda sensor heating 1 is distributed.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read LSH 1 status	M	#44h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read LSH 1 status	M	#44h	RLI_
#3d	RecordValue # 1 = STATE_LSH_UP[1]	M	#XXh	RV_
#4d	RecordValue # 2 = dummy value fix	M	#FFh	RV_
#5d	RecordValue # 3 = dummy value fix	M	#FFh	RV_
#6d	RecordValue # 4 = dummy value fix	M	#FFh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.18 21_49 - read Immobilizer status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	LV_LOCK_IMOB		

General information:

With this comand the status of the Immobilizer is distributed.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on (LV_IGK=1)

Deactivation: Kl.15 off (LV_IGK=0)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read Immobilizer status	M	#49h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read Immobilizer status	M	#49h	RLI_
#3d	RecordValue # 1 =LV_LOCK_IMOB	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.19 21_4C - Read basic injection time cylinder 1 - ReadDataByLocalIdentifier service

Input data:

LV_IGK	TI_1_HOM[NC_CYL_NR]		

General information:

This funktion reports the value of the “injection time cylinder logical”.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier TI_1_HOM[0]	M	#4Ch	RLI_BIOS_TI_0

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier TI_1_HOM[0]	M	#4Ch	RLI_
#3d	RecordValue = TI_1_HOM[0] high byte	M	#XXh	RV_S_
#4d	RecordValue = TI_1_HOM[0] low byte	M	#XXh	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.20 21_50 - read engine speed - ReadDataByLocalIdentifier service

Input data:

LV_IGK	N		

General information:

This service reports the actual value of the engine speed.

Application conditions:

Initialisation: at reset with 00h

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read engine speed	M	#50h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= engine speed	M	#50h	RLI_
#3d	RecordValue # = N high byte	M	#XXh	RV_
#4d	RecordValue # = N low byte	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.21 21_54 - read intake air temperature - ReadDataByLocalIdentifier service

Input data:

LV_IGK	TIA_MES		

General information:

This service reports the actual value of the intake air temperature.

Application conditions:

Initialisation: at reset with 00h

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read intake air temperature	M	#54h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= intake air temperature	M	#54h	RLI_
#3d	RecordValue = TIA_MES	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.22 21_55 - read coolant temperature - ReadDataByLocalIdentifier service

Input data:

LV_IGK	TCO_MES		

General information:

This service reports the actual value of the coolant temperature.

Application conditions:

Initialisation: at reset with 00h

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read coolant temperature	M	#55h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= coolant temperature	M	#55h	RLI_
#3d	RecordValue = TCO_MES	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.23 21_56 - Read ignition angle cylinder 1 - ReadDataByLocalIdentifier service

Input data:

LV_IGK	IGA_IGC[NC_CYL_NR]		

General information:

This funktion reports the value of the “Ignition angle applied on cylinder log.0 (phys1)”.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier IGA_IGC[0]	M	#56h	RLI_IGA_IGC_0

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier IGA_IGC[0]	M	#56h	RLI_
#3d	RecordValue = IGA_IGC[0]	M	#XXh	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.24 21_57 - read opening angle of the throttle valve (TPS_AV) - ReadDataByLocalIdentifier service

Input data:

LV_IGK	TPS_AV		

General information:

This service reports the actual value of the opening angle of the throttle valve (TPS_AV).

Application conditions:

Initialisation: at reset with 00h

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read opening angle of the throttle valve (TPS_AV)	M	#57h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= opening angle of the throttle valve (TPS_AV)	M	#57h	RLI_
#3d	RecordValue # = TPS_AV high Byte	M	#XXh	RV_
#4d	RecordValue # = TPS_AV low Byte	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.25 21_58 - read mass air flow per segment measured (MAF_KGH_MES) - ReadDataByLocalIdentifier service

Input data:

LV_IGK	MAF_KGH_MES		

General information:

This service reports the actual value of the mass air flow per segment measured (MAF_KGH_MES)

Application conditions:

Initialisation: at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read mass air flow per segment measured (MAF_KGH_MES)	M	#58h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= mass air flow per segment measured (MAF_KGH_MES)	M	#58h	RLI_
#3d	RecordValue # = MAF_KGH_MES high Byte	M	#XXh	RV_
#4d	RecordValue # = MAF_KGH_MES low Byte	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.26 21_5B - read Pedal value sensor 1 raw acquisition (V_PVS_1_KWP) - ReadDataByLocalIdentifier service

Input data:

LV_IGK	V_PVS_1_KWP		

General information:

This service reports the actual value of the Pedal value sensor 1 raw acquisition of the accelerator pedal (V_PVS_1_KWP).

V_PVS_1_KWP is defined within the DDLI values.

Application conditions:

Initialisation: at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = V_PVS_1_KWP	M	#5Bh	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= V_PVS_1_KWP	M	#5Bh	RLI_
#3d	RecordValue # = V_PVS_1_KWP	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.27 21_71 - read LV_KNK - ReadDataByLocalIdentifier service

Input data:

LV_IGK	LV_KNK		

General information:

This funktion reports the value of the “Ignition angle applied on cylinder log.0 (phys1)”.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier LV_KNK	M	#71h	RLI_LV_KNK

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier LV_KNK	M	#71h	RLI_
#3d	RecordValue = LV_KNK	M	#XXh	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.28 21_A3 - read VB - ReadDataByLocalIdentifier service

Input data:

LV_IGK	VB		

General information:

This funktion reports the value of the “battery voltage”.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier VB	M	#A3h	RLI_VB

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier VB	M	#A3h	RLI_
#3d	RecordValue = VB	M	#XXh	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.29 21_C0 - read SAFM status - ReadDataByLocalIdentifier service

Input data:

LV_IGK	STATE_DIAG_SA_SAFM		

General information:

With this comand the status of the secondary air mass sensor is distributed.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on (LV_IGK=1)

Deactivation: Kl.15 off (LV_IGK=0)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = read SAFM status	M	#C0h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier= read SAFM status	M	#C0h	RLI_
#3d	RecordValue # 1 = STATE_DIAG_SA_SAFM	M	#XXh	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.30 21_C1 - read kva-factor - ReadDataByLocalIdentifier service

Input data:

LV_IGK	FAC_FCO_KWP		

General information:

This funktion shows the additive part of the correctionfactor for the consumption - displayed in the combi - for the customer service.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = KVA-factor	M	#C1h	RLI_KVA

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier	M	#C1h	RLI_KVA
#3d	RecordValue = FAC_FCO_KWP [remain the same negative displace max.12.8% positive displace max.12.7% (1h=0.001/bit)]	M	#XXh [#00h, #80h-FFh #01h-7Fh]	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.31 21_C3 - read TRT - ReadDataByLocalIdentifier service

Input data:

LV_IGK	TRT		

General information:

This funktion reports the value of the “Total running time”.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**

#2d	recordLocalIdentifier TRT	M	#C3h	RLI_TRT

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier TRT	M	#C3h	RLI_
#3d	RecordValue = TRT high byte high word	M	#XXh	RV_S_
#4d	RecordValue = TRT low byte high word	M	#XXh	RV_S_
#5d	RecordValue = TRT high byte low word	M	#XXh	RV_S_
#6d	RecordValue = TRT low byte low word	M	#XXh	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

7.19.32 21_F0 - DDLI handling - ReadDataByLocalIdentifier service

General information:

This function is created to read the DDLI, which has to be defined before with the service 2Ch.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = DDLI	M	#F0h	DDLI

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response

Example 1: DDLIDefinitonMode=DefineByCommonIdentifier and MemorySize=01

Information, which is given by the DDLI service (compare specification “DynamicallyDefineLocalIdentifier service – 2Ch“).

MemorySize: fixed to 01
PIDDDLI_x : PositionInDynamicallyDefinedLocalIdentifier x
RCI_x : RecordCommonID *(according to IO-data list) x*

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier = DDLI	M	#F0h	RLI_
..	..	..	..	..
..	..	..	..	..
#2d+ PIDDDLI_1	Contents of IO-data given by the number in RCI_1 *(compare IO-data list)* in byte format	M	#XXh	
..	..	..	..	..
..	..	..	..	..
..	..	..	..	..
#2d+ PIDDDLI_x	Contents of IO-data given by the number in RCI_x *(compare IO-data list)* in byte format	M	#XXh	

Table readDataByLocalIdentifier Positive Response Message

Example 2: DDLIDefinitonMode=DefineByCommonIdentifier and MemorySize=02

Information, which is given by the DDLI service (compare specification “DynamicallyDefineLocalIdentifier service – 2Ch“).

MemorySize: fixed to 02
PIDDDLI_x : PositionInDynamicallyDefinedLocalIdentifier x
RCI_x : RecordCommonID *(according to IO-data list) x*

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
#1d	**ReadDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier = DDLI	M	#F0h	RLI_
..	..	..	..	..
..	..	..	..	..
#2d+ PIDDDLI_1	Contents of IO-data given by the number in RCI_1 *(compare IO-data list)* high byte	M	#XXh	
#2d+ 1+ PIDDDLI_1	Contents of IO-data given by the number in RCI_1 *(compare IO-data list)* low byte	M	#XXh	
..	..	..	..	..
..	..	..	..	..
#2d+ PIDDDLI_x	Contents of IO-data given by the number in RCI_x *(compare IO-data list)* high byte	M	#XXh	
#2d+ 1+ PIDDDLI_x	Contents of IO-data given by the number in RCI_x *(compare IO-data list)* low byte	M	#XXh	

Table readDataByLocalIdentifier Positive Response Message

Example 3: DDLIDefinitonMode= DefineByMemoryAddress

Information, which is given by the DDLI service (compare specification “DynamicallyDefineLocalIdentifier service – 2Ch“).

MemorySize_x
PIDDDLI_x : PositionInDynamicallyDefinedLocalIdentifier x
MemoryAddress_x
MemoryTypeIdentifier_x

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
#1d	**ReadDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	RecordLocalIdentifier = DDLI	M	#F0h	RLI_
..	..	..	..	..
..	..	..	..	..
#2d+ PIDDDLI_1	Contents of MemoryAddress_1 in MemoryTypeIdentifier_1 section	M	#XXh	
#2d+ 1 + PIDDDLI_1	Contents of MemoryAddress_1+1 in MemoryTypeIdentifier_1 section	M	#XXh	
#2d+ MemorySize_1+ PIDDDLI_1	Contents of MemoryAddress_1+MemorySize_1 in MemoryTypeIdentifier_1 section	M	#XXh	
..	..	..	..	..
..	..	..	..	..
#2d+ PIDDDLI_x	Contents of MemoryAddress_x in MemoryTypeIdentifier_x section	M	#XXh	
#2d+ 1 + PIDDDLI_x	Contents of MemoryAddress_x+1 in MemoryTypeIdentifier_x section	M	#XXh	

#2d+ MemorySize_x+ PIDDDLI_x	Contents of MemoryAddress_x+MemorySize_x in MemoryTypeIdentifier_x section	M	#XXh	

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Global Negative Responses

NegativeResponseCode	***Cause of Occurrence***
SubFunctionNotSupported-invalidFormat	- selected RCI is not supported - selected DDDLI is not supported

7.20 SID 23h: ReadMemoryByAddress service

Input data:

LV_IGK			

General information:

This function ist for reading values from the memory

Application conditions:

Initialisation: at reset

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0
Diagnostic timeout

Service location: see "list of implemented diagnostic services"

7.20.1 ReadMemoryByAddress Request (3-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadMemoryByAddress Request**	**M**	**#23h**	**RMBA**
#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (Middle Byte)	M	#XXh	MA_MB
#4d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
(#5d)	MemoryTypeIdentifier **(optional)** [linear type external FLASH external RAM internal FLASH user information field UIF]	U	#XXh [#00h #02h #05h #06h #07h]	MTI
#5d / (#6d)	MemorySize	M	#1h...#FEh	MS

Table ReadMemoryByAddress Request Message (3-byte addressing)

7.20.2 ReadMemoryByAddress Request (4-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadMemoryByAddress Request**	**M**	**#23h**	**RMBA**
#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (MiddleHigh Byte)	M	#XXh	MA_MHB
#4d	MemoryAddress (MiddleLow Byte)	M	#XXh	MA_MLB
#5d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
#6d	MemoryTypeIdentifier **(mandatory, but not evaluated)** [00h … FFh]	M	#XXh [#XXh]	MTI
#7d	MemorySize	M	#1h...#FEh	MS

Table ReadMemoryByAddress Request Message (4-byte addressing)

Remark:

For 3-byte addressing **independent from delivery of a MemoryTypeIdentifier (byte 5) by the tester´s request, the ECU automatically adds an offset to the requested memory address.**
If no MemoryTypeIdentifier is delivered, the same offset as for MemoryTypeIdentifier 00h (linear addressing) is used.
This base memory offset must be subtracted from any memory address taken from the MAP- / A2L- file if 3-byte addressing is not sufficient for the address range!

For 4-byte addressing **a MemoryTypeIdentifier (byte 6) must be delivered by the tester´s request.**
Though it is not evaluated or used within the ECU for this absolute addressing mode, it is necessary to decide between 3- / 4-byte addressing mode.
In this 4-byte mode the absolute address delivered by the tester is used by the ECU without any memory offset.

For MSx80 (TriCore) the base memory offsets for 3-byte addressing **are:**

MemoryTypeIdentifier:	Memory offset:		
none	**(linear)**		**C000 0000h**
00h	**(linear)**		**C000 0000h**
02h	**(ext. FLASH)**		**8000 0000h (the same as for int. FLASH)**
05h	**(ext. RAM)**	**D000 0000h**	
06h	**(int. FLASH)**	**8000 0000h**	
07h	**(UIF)**		**8000 xxxxh not fixed yet, address might still change !!!**

Boot Software Active			***ECU Software Active***		
Memory ID	***Range (absolute addr.)***	***Security Level***	***Memory ID***	***Range (absolute addr.)***	***Security Level***
00h (linear)	C000 0000h ...C000 FFFFh C03F C000h…C03F FFFFh	None	00h (linear)	C000 0000h ...C000 FFFFh C03F C000h…C03F FFFFh	None
02h (ext. FLASH)	8000 0000h …801F FFFFh	None	02h (ext. FLASH)	8000 0000h …801F FFFFh	None
05h (ext. RAM)	D000 0000h…D000 FFFFh	None	05h (ext. RAM)	D000 0000h…D000 FFFFh	None
06h (int. FLASH)	8000 0000h …801F FFFFh	None	06h (int. FLASH)	8000 0000h …801F FFFFh	None
07h (UIF)	8000 xxxxh … 8000 xxxxh	None	07h (UIF)	8000 xxxxh … 8000 xxxxh	None

Table call conditions readMemoryByAddress (2)

7.20.3 ReadMemoryByAddress positive Response

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
#1d	**ReadMemoryByAddress Positive Response**	**M**	**#63h**	**RMBAPR**
#2d . . . #nd	RecordValue#1 data 1 . . . RecordValue#m	M . . . M	#XXh . . . #XXh	RV

Table ReadMemoryByAddress Positive Response Message

7.20.4 ReadMemoryByAddress negative Response

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
#1d	**NegativeResponse Service Id**	**S**	**#7Fh**	**NR**
#2d	ReadMemoryByAddressRequest Service Id	M	#23h	RMBA
#3d	ResponseCode= [general response code]	M	#XXh= [#XXh]	RC

Table ReadMemoryByAddress Negative Response Message

7.21 SID 3Dh: WriteMemoryByAddress service

Input data:

LV_IGK			

General information:

This function is used for writing values into the ECU´s memory

Application conditions:

Initialisation:	at reset
Activation:	Kl.15 on
Deactivation:	Kl.15 off Diagnostic timeout
Service location:	see "list of implemented diagnostic services"

7.21.1 WriteMemoryByAddress Request (3-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteMemoryByAddress Request**	**M**	**#3Dh**	**WMBA**
#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (Middle Byte)	M	#XXh	MA_MB
#4d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
(#5d)	MemoryTypeIdentifier **(optional)** [external FLASH internal FLASH user information field UIF]	U	#XXh [#02h #06h #07h]	MTI
#5d / (#6d)	MemorySize	M	#1h...#FAh	MS
#6d / (#7d) ... #nd	RecordValue #1 ... RecordValue #m	M ... U	#XXh ... #XXh	RV

Table WriteMemoryByAddress Request Message

7.21.2 WriteMemoryByAddress Request (4-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteMemoryByAddress Request**	**M**	**#3Dh**	**WMBA**
#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (MiddleHigh Byte)	M	#XXh	MA_MHB
#4d	MemoryAddress (MiddleLow Byte)	M	#XXh	MA_MLB
#5d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
#6d	MemoryTypeIdentifier **(mandatory, but not evaluated)** [00h … FFh]	M	#XXh [#XXh]	MTI
#7d	MemorySize	M	#1h...#FAh	MS
#8d ... #nd	RecordValue #1 ... RecordValue #m	M ... U	#XXh ... #XXh	RV

Table WriteMemoryByAddress Request Message

Remark:

For 3-byte addressing **independent from delivery of a MemoryTypeIdentifier (byte 5) by the tester´s request, the ECU automatically adds an offset to the requested memory address.**
If no MemoryTypeIdentifier is delivered, the same offset as for MemoryTypeIdentifier 00h (linear addressing) is used.
This base memory offset must be subtracted from any memory address taken from the MAP- / A2L- file if 3-byte addressing is not sufficient for the address range!

For 4-byte addressing **a MemoryTypeIdentifier (byte 6) must be delivered by the tester´s request.**
Though it is not evaluated or used within the ECU for this absolute addressing mode, it is necessary to decide between 3- / 4-byte addressing mode.
In this 4-byte mode the absolute address delivered by the tester is used by the ECU without any memory offset.

For MSx80 (TriCore) the base memory offsets for 3-byte addressing **are:**

MemoryTypeIdentifier:		Memory offset:
02h	**(ext. FLASH)**	**8000 0000h (the same as for int. FLASH)**
06h	**(int. FLASH)**	**8000 0000h**
07h	**(UIF)**	**8000 xxxxh not fixed yet, address might still change !!!**

Boot Software Active			*ECU Software Active*		
Memory ID	*Range (absolute addr.)*	*Security Level*	*Memory ID*	*Range (absolute addr.)*	*Security Level*
02h (ext. FLASH)	8000 xxxxh … 8000 xxxxh	None	02h (ext. FLASH)	8000 xxxxh … 8000 xxxxh	None
06h (int. FLASH)	8000 xxxxh … 8000 xxxxh	None	06h (int. FLASH)	8000 xxxxh … 8000 xxxxh	None
07h (UIF)	8000 xxxxh … 8000 xxxxh	None	07h (UIF)	8000 xxxxh … 8000 xxxxh	None

Table call conditions writeMemoryByAddress

7.21.3 WriteMemoryByAddress positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteMemoryByAddress Positive Response**	**M**	**#7Dh**	**WMBAPR**

#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (Middle Byte)	M	#XXh	MA_MB
#4d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
(#5d)	MemoryTypeIdentifier **(optional)** [external FLASH internal FLASH user information field UIF]	U	#XXh [#02h #06h #07h]	MTI

Table WriteMemoryByAddress Positive Response Message

7.21.4 WriteMemoryByAddress negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**S**	**#7Fh**	**NR**
#2d	writeMemoryByAddressRequest Service Id	M	#3Dh	WMBA
#3d	responseCode=general response code]	M	#XXh	RC

Table WriteMemoryByAddress Negative Response Message

7.22 SID 30h: inputOutputControlByLocalIdentifier Service

General information:

All implemented SID 30h services are described below.

7.22.1 Overview of implemented subservices

SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_02_01	Bremslichtschalter auslesen	X	X	X	X	X	X
30_03_01	Bremslichttestschalter auslesen	X	X	X	X	X	X
30_04_01	Kupplungsschalter auslesen	X	X	X	X	X	X
30_05_01	Oeldruckschalter auslesen	X	X	X	X	X	X
30_06_01	Sporttaster auslesen	X	X	X	X	X	-
30_07_01	Nullgangssensor auslesen	-	X	-	X	-	-
30_0A_01	Ansauglufttemperatur 1 auslesen	X	X	X	X	X	X
30_0B_01	Ansauglufttemperatur 2 auslesen	-	-	-	-	-	X
30_0C_01	Motortemperatur auslesen	X	X	X	X	X	X
30_0D_01	Kuehlerauslasstemperatur auslesen	X	X	X	X	X	X
30_0E_01	Oelsensor auslesen	X	X	X	X	X	X
30_10_01	Steuergeraete-Innentemperatur auslesen	X	X	X	X	X	X
30_12_01	Abgastemperatur auslesen	-	X	X	X	X	X
30_13_01	Abgastemperatur 2 auslesen	-	-	-	-	-	X
30_17_01	Umgebungsdruck auslesen	X	X	X	X	X	X
30_18_01	Saugrohrdruck1 / Ladedruck1 auslesen	X	X	X	X	X	X
30_19_01	Saugrohrdruck2 / Ladedruck2 auslesen	-	X	X	X	X	X
30_1A_01	Raildrucksenor auslesen	-	X	X	X	X	X
30_1B_01	Kl.15 Spannung auslesen	X	X	X	X	X	X
30_1C_01	Kl.87 Spannung / Versorgung DME auslesen	X	X	X	X	X	X
30_1C_00/07	Kl.87 Spannung / Versorgung DME ansteuern	X	X	X	X	X	X
30_1E_01	Fahrerwunschversorgung 1 auslesen	X	X	X	X	X	X
30_1F_01	Fahrerwunschversorgung 2 auslesen	X	X	X	X	X	X
30_20_01	Versorgungsspannung VVL Sensor auslesen	X	-	-	-	-	-
30_20_01	Versorgung HPD5-Ventil auslesen	-	X	-	X	-	X
30_20_00/07	Versorgung HPD5-Ventil ansteuern	-	X	-	X	-	X
30_21_01	Lambdasonde vor Kat Bank 1 auslesen	X	X	X	X	X	X
30_22_01	Lambdasonde hinter Kat Bank 1 auslesen	X	X	X	X	X	X
30_23_01	Lambdasonde vor Kat Bank 2 auslesen	X	X	X	X	X	X
30_24_01	Lambdasonde hinter Kat Bank 2 auslesen	X	X	X	X	X	X
30_25_01	Luftmassenmesser 1 auslesen	X	X	X	X	X	X
30_27_01	Batteriesensor auslesen	X	X	X	X	X	X

SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_28_01	Fahrerwunsch1 auslesen	X	X	X	X	X	X
30_29_01	Fahrerwunsch 2 auslesen	X	X	X	X	X	X
30_2A_01	Drosselklappe auslesen	X	X	X	X	X	X
30_2A_00/07	Drosselklappe ansteuern	X	X	X	X	X	X
30_2E_01	Sekundaerluft HFM auslesen	X	-	-	-	-	X
30_2F_01	Taster Tempomat	X	X	X	X	X	-
30_30_01	Klopfbaustein1 auslesen	X	X	X	X	X	X
30_31_01	Klopfbaustein 2 auslesen	X	X	X	X	X	X
30_32_01	Generator Sollspannung BSD auslesen	X	X	X	X	X	X
30_32_00/07	Generator Sollspannung BSD ansteuern	X	X	X	X	X	X
30_37_01	Oeldrucksensor auslesen	X	X	X	X	X	-
30_3B_01	NOx-Sensor auslesen	X	X	X	X	X	-
30_3F_01	Kraftsoffniederdrucksensor auslesen	-	X	X	X	X	X
30_40_01	Bremsunterdrucksensor auslesen	X	X	-	X	-	-
30_43_01	Raildrucksenor 2 auslesen	-	-	-	-	-	X
30_44_01	Klopfbaustein 3 auslesen	-	-	-	-	-	X
30_45_01	Klopfbaustein 4 auslesen	-	-	-	-	-	X
30_46_01	Luftmassenmesser 2 auslesen	-	-	-	-	-	X
30_80_01	Radiator shutter auslesen	-	-	X	-	X	-
30_80_00/07	Radiator shutter ansteuern	-	-	X	-	X	-
30_82_01	HPD-Relais auslesen	-	X	-	X	-	X
30_82_00/07	HPD-Relais ansteuern	-	X	-	X	-	X
30_83_01	Wasserpumpe Turbolader auslesen	-	-	-	-	-	X
30_83_00/07	Wasserpumpe Turbolader ansteuern	-	-	-	-	-	X
30_84_01	selklappenstellergehäuseheizung 2 auslesen	-	-	-	-	-	X
30_84_00/07	selklappenstellergehäuseheizung 2 ansteuern	-	-	-	-	-	X
30_85_01	Drosselklappe 2 auslesen	-	-	-	-	-	X
30_85_00/07	Drosselklappe 2 ansteuern	-	-	-	-	-	X
30_86_01	(second) HPD5-Mengensteuerventil auslesen	-	X	-	X	-	X
30_86_00/07	(second) HPD5-Mengensteuerventil ansteuern	-	X	-	X	-	X
30_87_01	Abgasklappe 2 auslesen	-	-	-	-	-	X
30_87_00/07	Abgasklappe 2 ansteuern	-	-	-	-	-	X
30_88_01	Vanos Einlass Ventil Bank2 auslesen	-	-	-	-	-	X
30_88_00/07	Vanos Einlass Ventil Bank2 ansteuern	-	-	-	-	-	X
30_89_01	Vanos Auslass Ventil Bank2 auslesen	-	-	-	-	-	X
30_89_00/07	Vanos Auslass Ventil Bank2 ansteuern	-	-	-	-	-	X
30_8A_01	Umluftventil Bank 2 auslesen	-	-	-	-	-	X
30_8A_00/07	Umluftventil Bank 2 ansteuern	-	-	-	-	-	X

SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_8B_01	Sekundaerluftventil 2 auslesen	-	-	-	-	-	X
30_8B_00/07	Sekundaerluftventil 2 ansteuern	-	-	-	-	-	X
30_9E_01	Drosselklappenstellergehaueseheizung auslesen	-	X	X	X	X	X
30_9E_00/07	Drosselklappenstellergehaueseheizung ansteuern	-	X	X	X	X	X
30_A4_01	Gesteuerte Luftfuehrung (untere Klappe) auslesen	X	X	X	X	X	X
30_A4_00/07	Gesteuerte Luftfuehrung (untere Klappe) ansteuern	X	X	X	X	X	X
30_AB_01	Oel Druck Regelung auslesen	X	X	X	X	X	-
30_AB_00/07	Oel Druck Regelung ansteuern	X	X	X	X	X	-
30_AC_01	Oeldruckventil auslesen	X	X	X	X	X	-
30_AC_00/07	Oeldruckventil ansteuern	X	X	X	X	X	-
30_AD_01	Kurbelgehaeuseentlueftungsheizung auslesen	-	X	X	X	X	X
30_AD_00/07	Kurbelgehaeuseentlueftungsheizung ansteuern	-	X	X	X	X	X
30_AE_01	Variable Sauganlage (DISA) Klappe2 auslesen	X	X	X	X	X	-
30_AE_00/07	Variable Sauganlage (DISA) Klappe2 ansteuern	X	X	X	X	X	-
30_B2_01	Motorlagersteuerung auslesen	-	X	X	X	X	-
30_B2_00/07	Motorlagersteuerung ansteuern	-	X	X	X	X	-
30_B5_01	Umluftventil Bank 1 auslesen	-	-	-	-	-	X
30_B5_00/07	Umluftventil Bank 1 ansteuern	-	-	-	-	-	X
30_B6_01	Ladedrucksteller 1 auslesen	-	X	X	X	X	X
30_B6_00/07	Ladedrucksteller 1 ansteuern	-	X	X	X	X	X
30_B7_01	Ladedrucksteller 2 auslesen	-	X	X	X	X	X
30_B7_00/07	Ladedrucksteller 2 ansteuern	-	X	X	X	X	X
30_BD_01	Mengensteuerventil auslesen	-	X	X	X	X	X
30_BD_00/07	Mengensteuerventil ansteuern	-	X	X	X	X	X
30_BE_01	Abgasrückführungsventil auslesen	X	X	X	X	X	-
30_BE_00/07	Abgasrückführungsventil ansteuern	X	X	X	X	X	-
30_BF_01	elektr. Wasserpumpe ueber BSD auslesen	X	X	X	X	X	-
30_BF_00/07	elektr. Wasserpumpe ueber BSD ansteuern	X	X	X	X	X	-
30_C1_01	Abgasklappe auslesen	X	X	X	X	X	X
30_C1_00/07	Abgasklappe ansteuern	X	X	X	X	X	X
30_C2_01	Soundklappeauslesen	X	X	X	X	X	X
30_C2_00/07	Soundklappe ansteuern	X	X	X	X	X	X
30_C3_01	Gesteuerte Luftfuehrung auslesen	X	X	X	X	X	X
30_C3_00/07	Gesteuerte Luftfuehrung ansteuern	X	X	X	X	X	X
30_C4_01	Startrelais auslesen	X	X	X	X	X	X
30_C4_00/07	Startrelais ansteuern	X	X	X	X	X	X
30_C6_01	Variable Sauganlage (DISA) Klappe auslesen	X	X	X	X	X	-

SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_C6_00/07	Variable Sauganlage (DISA) Klappe ansteuern	X	X	X	X	X	-
30_C7_01	Klimakompressor-Relais auslesen	X	X	X	X	X	-
30_C7_00/07	Klimakompressor-Relais ansteuern	X	X	X	X	X	-
30_C8_01	EBOX_Controlled_Fan auslesen	X	X	X	X	X	X
30_C8_00/07	EBOX_Controlled_Fan ansteuern	X	X	X	X	X	X
30_C9_01	Electr_Controlled_Thermostat auslesen	X	X	X	X	X	X
30_C9_00/07	Electr_Controlled_Thermostat ansteuern	X	X	X	X	X	X
30_CA_01	Sekundaerluftventil auslesen	X	-	-	-	-	X
30_CA_00/07	Sekundareluftventil ansteuern	X	-	-	-	-	X
30_CB_01	Sekundaerluftpumpe auslesen	X	-	-	-	-	X
30_CB_00/07	Sekundaerluftpumpe ansteuern	X	-	-	-	-	X
30_CC_01	Diagnosemodul-Tank Leckage Pumpe auslesen	X	-	X	-	X	X
30_CC_00/07	Diagnosemodul-Tank Leckage Pumpe ansteuern	X	-	X	-	X	X
30_CD_01	Diagnosemodul-Tank Leckage Ventil auslesen	X	-	X	-	X	X
30_CD_00/07	Diagnosemodul-Tank Leckage Ventil ansteuern	X	-	X	-	X	X
30_CE_01	Diagnosemodul-Tank Leckage Heizung auslesen	X	-	X	-	X	X
30_CE_00/07	Diagnosemodul-Tank Leckage Heizung ansteuern	X	-	X	-	X	X
30_CF_01	Canister_Purge_Solenoid auslesen	X	X	X	X	X	X
30_CF_00/07	Canister_Purge_Solenoid ansteuern	X	X	X	X	X	X
30_D0_01	Lambda_Sensor_Heating_Up_1 auslesen	X	X	X	X	X	X
30_D0_00/07	Lambda_Sensor_Heating_Up_1 ansteuern	X	X	X	X	X	X
30_D1_01	Lambda_Sensor_Heating_Down_1 auslesen	X	X	X	X	X	X
30_D1_00/07	Lambda_Sensor_Heating_Down_1 ansteuern	X	X	X	X	X	X
30_D2_01	Lambda_Sensor_Heating_Up_2 auslesen	X	X	X	X	X	X
30_D2_00/07	Lambda_Sensor_Heating_Up_2 ansteuern	X	X	X	X	X	X
30_D3_01	Lambda_Sensor_Heating_Down_2 auslesen	X	X	X	X	X	X
30_D3_00/07	Lambda_Sensor_Heating_Down_2 ansteuern	X	X	X	X	X	X
30_D4_01	Malfunction_Indicator_Lamp auslesen	X	X	X	X	X	X
30_D4_00/07	Malfunction_Indicator_Lamp ansteuern	X	X	X	X	X	X
30_D6_01	EML_Lamp auslesen	X	X	X	X	X	X
30_D6_00/07	EML_Lamp ansteuern	X	X	X	X	X	X
30_D8_01	Electrical_Fuel_Pump auslesen	X	X	X	X	X	X
30_D8_00/07	Electrical_Fuel_Pump ansteuern	X	X	X	X	X	X
30_DA_01	E-Box-Luefter auslesen	X	X	X	X	X	X
30_DA_00/07	E-Box-Luefter ansteuern	X	X	X	X	X	X
30_DC_01	VVT-Entlastungsrelais auslesen	X	-	-	-	-	X
30_DC_00/07	VVT-Entlastungsrelais ansteuern	X	-	-	-	-	X

SID number	service	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
30_DD_01	ValveTronic Lift auslesen	X	-	-	-	-	X
30_DD_00/07	ValveTronic Lift ansteuern	X	-	-	-	-	X
30_E1_01	Einspritzventil 1 auslesen	X	X	X	X	X	X
30_E1_00/07	Einspritzventil 1 ansteuern	X	-	-	-	-	-
30_E2_01	Einspritzventil 2 auslesen	X	X	X	X	X	X
30_E2_00/07	Einspritzventil 2 ansteuern	X	-	-	-	-	-
30_E3_01	Einspritzventil 3 auslesen	X	X	X	X	X	X
30_E3_00/07	Einspritzventil 3 ansteuern	X	-	-	-	-	-
30_E4_01	Einspritzventil 4 auslesen	X	X	X	X	X	X
30_E4_00/07	Einspritzventil 4 ansteuern	X	-	-	-	-	-
30_E5_01	Einspritzventil 5 auslesen	X	-	X	-	X	X
30_E5_00/07	Einspritzventil 5 ansteuern	X	-	-	-	-	-
30_E6_01	Einspritzventil 6 auslesen	X	-	X	-	X	X
30_E6_00/07	Einspritzventil 6 ansteuern	X	-	-	-	-	-
30_E7_01	Einspritzventil 7 auslesen	-	-	-	-	-	X
30_E8_01	Einspritzventil 8 auslesen	-	-	-	-	-	X
30_ED_01	Vanos Einlass Ventil auslesen	X	X	X	X	X	X
30_ED_00/07	Vanos Einlass Ventil ansteuern	X	X	X	X	X	X
30_EE_01	Vanos Auslass Ventil auslesen	X	X	X	X	X	X
30_EE_00/07	Vanos Auslass Ventil ansteuern	X	X	X	X	X	X
30_EF_01	HPD5-Mengensteuerventil 1 auslesen	-	-	-	-	-	X
30_EF_00/07	HPD5-Mengensteuerventil 1 ansteuern	-	-	-	-	-	X

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ANG_SP_EXT_ADJ_VVL	O/V	0...FFH	0...179.29687	0.703125	[°]
Valvtronic lift setpoint from the tester to valveronic controller					
arqtmsv_w_rb	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°CRK]
Control value of VCV via tester					
arqtmsv_w_rb[NC_CBK_HPP_NR]	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°CRK]
Control value of VCV via tester					
B_rqtmsv_rb	O/V	0...1H	0...1	1	[-]
Condition VCV control via tester request					
B_rqtmsv_rb[NC_CBK_HPP_NR]	O/V	0...1H	0...1	1	[-]
Condition VCV control via tester request					
CAM_SP_EX_EXT_ADJ	O/V	0...FFFFH	-128...52300	0.8	[°CRK]
Output signal from the module actuator diagnosis to the function					
CAM_SP_2_EX_EXT_ADJ	O/V	0...FFFFH	-128...52300	0.8	[°CRK]
Output signal from the module actuator diagnosis to the function					
CAM_SP_IN_EXT_ADJ	O/V	0...FFFFH	-128...52300	0.8	[°CRK]
Output signal from the module actuator diagnosis to the function					
CAM_SP_2_IN_EXT_ADJ	O/V	0...FFFFH	-128...52300	0.8	[°CRK]
Output signal from the module actuator diagnosis to the function					
CPPWM_EXT_ADJ	O/V	0...FFH	0...99.60937	0.390625	[%]
CPPWM requested by service tool					
ECFPWM_ECF_EXT_ADJ	O/V	0...FFH	0...99.60937	0.390625	[%]
Electric fan control pulse width modulation-signal for the function InputOutputControl					
ECTPWM_EXT_ADJ	O/V	0...FFH	0...99.60937	0.390625	[%]
PWM output signal for the electronic controlled thermostat in case of external adjustment					
EFPPWM_EXT_ADJ	O/V	0...FFFFH	0...99.99847	1.5259e-3	[%]
EFPPWM while external controling by service tool					
FUP_SP_EXT_ADJ	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
fuel pressure setpoint requested via KWP					
IV_EXT_ADJ[NC_CYL_NR]	O/V	0...FFH	0...20.4	0.08	[ms]
Injection time cylinder individual at external adjustment					
IV_PER_EXT_ADJ[NC_CYL_NR]	O/V	0...FFH	0...2550	10	[ms]
Injection period cylinder individual at external adjustment					
LSHPWM_DOWN_EXT_ADJ[NC_CBK_EX_NR]	O/V	0...FFH	0...99.60937	0.390625	[%]
PWM-signal for the function InputOutputControl					
LSHPWM_UP_EXT_ADJ[NC_CBK_EX_NR]	O/V	0...FFH	0...99.60937	0.390625	[%]
PWM-signal for the function InputOutputControl					
LV_ACCOUT_RLY_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_ACT_ACCOUT_RLY_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for ACC-Relay 0=off, 1=on					
LV_ACT_CRU_EXT_ADJ	O/V	0...1H	0...1	1	[-]
switch for the function InputOutputControl 1 = active 0 = deactive					
LV_ACT_DMTL_PUMP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for DMTL pump 0 = off, 1 = on					
LV_ACT_DMTLH_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for DMTL heater 0 = off, 1 = on					
LV_ACT_DMTLS_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for DMTL solenoid 0 = off, 1 = on					
LV_ACT_EBOX_CFA_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for ebox cooling fan 0 = off, 1 = on					
LV_ACT_ECRAS_DOWN_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Bit used to determine ECRASPWM by external adjustment					
LV_ACT_ECRAS_UP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Bit used to determine ECRASPWM by external adjustment					
LV_ACT_EF_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for Exhaust Flap 0=off, 1=on					
LV_ACT_EF_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
adjustment value for second Exhaust Flap 0=off, 1=on					
LV_ACT_MIL_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for MIL 1 = active 0 = inactive					
LV_ACT_RAS_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for the radiator shutter (0 = off, 1 = on)					
LV_ACT_RLY_CRCV_HEAT_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for crankcase ventilation heater relay 0 = closed, 1 = open					
LV_ACT_RLY_HPDI_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Command from tester for switching Piezo relay (0=off, 1=on)					
LV_ACT_RLY_MAIN_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Command from tester for switching main relay (0 = off, 1 = on)					
LV_ACT_RLY_MTC_HEAT_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for throttle housing heater relay 0 = closed, 1 = open					
LV_ACT_RLY_MTC_2_HEAT_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for second throttle housing heater relay 0 = closed, 1 = open					
LV_ACT_RLY_ST_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for starter relay 0 = off, 1 = on					
LV_ACT_RLY_VCV_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for VCV relay 0 = off, 1 = on					
LV_ACT_RLY_VVL_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for VVL relay 0 = off, 1 = on					
LV_ACT_SAP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Logical variable for SAP actuator test, set if request from service tool is accepted					
LV_ACT_SAV_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Logical variable for SAV actuator test, set if request from service tool is accepted					
LV_ACT_SAV_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Logical variable for second SAV actuator test, set if request from service tool is accepted					
LV_ACT_SOF_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for sound flap 0=off, 1=on					
LV_ACT_VIM_1_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for DISA1 0 = closed, 1 = open					
LV_ACT_VIM_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for DISA2 0 = closed, 1 = open					
LV_ACT_WAL_1_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for SAP 0 = off, 1 = on					
LV_AD_DIS_EXT_ADJ	O/V	0...1H	0...1	1	[-]
boolean for disable adaptions while active external adjust					
LV_ANG_SP_EXT_ADJ_VVL	O/V	0...1H	0...1	1	[-]
Valvtronic lift setpoint from the tester to valveronic controller active					
LV_CAM_SP_EX_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_CAM_SP_2_EX_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_CAM_SP_IN_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_CAM_SP_2_IN_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_CPPWM_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Activation of external adjustment of CPPWM by InputOutputControl					
LV_CRU_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of readiness of the function InputOutputControl					
LV_DMTL_PUMP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_DMTLH_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_DMTLS_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_EBOX_CFA_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_ECFPWM_ECF_EXT_ADJ	O/V	0...1H	0...1	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
activation of external adjustment by InputOutputControl					
LV_ECRAS_DOWN_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Bit indicating that external adjustment of ECRAS_DOWN is active					
LV_ECRAS_UP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Bit indicating that external adjustment of ECRAS_UP is active					
LV_ECTPWM_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of the external adjustment by InputOutputControl					
LV_EF_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Activation of external adjustment by InputOutputControl					
LV_EF_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Activation of external adjustment by InputOutputControl					
LV_EFP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Status of electrical fuel pump requested by service tool					
LV_FUP_SP_REQ_EXT_ADJ	O/V	0...1H	0...1	1	[-]
request of external adjustement of fuel pressure					
LV_IV_EXT_ADJ[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_LSHPWM_DOWN_EXT_ADJ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_N_REL_CWP_SP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_N_REL_CWP_SP_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_OPG_SP_ACR_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Logical variable to activate the actuator setpoint adjustment by service tester					
LV_POIL_PWM_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_POIL_SP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_RAS_EXT_ADJ	O/V	0...1H	0...1	1	[-]
adjustment value for the radiator shutter (0 = off, 1 = on)					
LV_RFPPWM_EXT_ADJ[NC_NR_TCHA]	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_RLY_CRCV_HEAT_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_RLY_HPDI_EXT_ADJ	O/V	0...1H	0...1	1	[-]
HPDI relay control command state (0H = relay shall be switched off, 1H = relay shall be switched on)					
LV_RLY_MAIN_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Request of tester for main relay adjustment					
LV_RLY_MTC_HEAT_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_RLY_MTC_2_HEAT_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_RLY_ST_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_RLY_VCV_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Request of tester for VCV-relay adjustment					
LV_RLY_VVL_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_SAP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Status of SAP requested by service tool					
LV_SAV_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Status of SAV requested by service tool					
LV_SAV_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Status of second SAV requested by service tool					
LV_SOF_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_SWI_AEB_ACT_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Adjustment value for the activation of AEB (0= Off, 1 = On)					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SWI_AEB_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Adjustment value for active engine bracket (0= Off, 1 = On)					
LV_TPS_SP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Logical variable indicates an ETC setpoint request by external device					
LV_TPS_SP_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Logical variable indicates an second ETC setpoint request by external device					
LV_V_ALTER_SP_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_VIM_1_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_VIM_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_WAL_1_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by InputOutputControl					
LV_WGPWM_EXT_ADJ[NC_NR_TCHA]	O/V	0...1H	0...1	1	[-]
activation of external adjustment "Ladedrucksteller ansteuern"					
N_REL_CWP_SP_EXT_ADJ	O/V	0...FFH	0...99.60937	0.390625	[%]
adjustment value for speed of electrical water pump					
N_REL_CWP_SP_2_EXT_ADJ	O/V	0...FFH	0...99.60937	0.390625	[%]
adjustment value for speed of electrical water pump					
OPG_SP_ACR_EXT_ADJ	O/V	0...FFFH	0...99.97558	0.0244141	[%]
Actuator setpoint requested by service tester					
POIL_PWM_EXT_ADJ	O/V	0...FFH	0...99.60937	0.390625	[%]
adjustment value for Oilpressure solenoid setpoint					
POIL_SP_EXT_ADJ	O/V	0...FFH	0...8160	32	[hPa]
adjustment value for Oilpressure setpoint					
RFPPWM_EXT_ADJ[NC_NR_TCHA]	O/V	0...FFH	0...99.60937	0.390625	[%]
adjustment value for RFP setpoint					
TIMEOUT_TIME_1C_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_20_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_2A_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_32_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_80_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_82_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_83_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_84_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_85_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_87_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_88_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_89_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_8A_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_8B_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_9E_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_A4_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TIMEOUT_TIME_AB_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_AC_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_AD_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_AE_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_B2_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_B5_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_B6_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_B7_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_BF_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_C1_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_C2_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_C3_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_C4_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_C6_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_C7_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_C8_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_C9_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_CA_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_CB_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_CC_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_CD_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_CE_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_CF_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_D0_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_D1_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_D2_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_D3_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_D4_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_D5_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_D6_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TIMEOUT_TIME_D8_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_DA_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_DC_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_DD_KWP	O	0...FFH	0...510	2	[s]
Timeout for tester request for valvetronic					
TIMEOUT_TIME_E1_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_E2_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_E3_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_E4_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_E5_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_E6_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_E7_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_E8_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_ED_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_EE_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TIMEOUT_TIME_EF_KWP	O	0...FFH	0...510	2	[s]
timeout time for stopping a started service actuator diagnosis					
TPS_SP_EXT_ADJ	O/V	0...FFH	0...99.60937	0.390625	[%]
Throttle position setpoint from tester					
TPS_SP_2_EXT_ADJ	O/V	0...FFH	0...99.60937	0.390625	[%]
second throttle position setpoint from tester					
V_ALTER_SP_EXT_ADJ	O/V	0...FFFFH	0...6553.5	0.1	[V]
adjustment value for alternator voltage setpoint					
WGPWM_EXT_ADJ[NC_NR_TCHA]	O/V	0...FFFFH	0...99.99847	1.5259e-3	[%]
PWM output wastegate[NC_CBK_EX_NR]					

Input data:

Ba_ist	FUP	GEAR_INFO	LV_AT
LV_BRAKE_DET	LV_CS	LV_ES	LV_IGK
LV_IS	LV_TEMP_MAX_SAP_COIL	LV_VAR_AMT	LV_VAR_TCT
N	NC_CYL_NR	NC_CBK_EX_NR	NC_NR_TCHA
	N_EFP_AV	POIL	STATE_MTC_HEAT
STATE_SA	TCO	VB	VS

Application conditions (expect services 30_xx_00/01, 30_BD_07, 30_CB_07):

Initialisation: 0 at reset

Activation: depending on calibratable conditions

see chapter "Activation conditions for SID 30_xx_07h services"

Deactivation: not activation

Service location: for each project depending on calibration data ID_IDX_ACT_DS_CONF

Global negative Responses

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**negativeResponse Service Id**	**M**	**7Fh**	**NR**
2d	inputOutputControlByLocalIdentifier Request SId	M	30h	IOBLID
3d	responseCode= only general response code	M	XXh	RC

The detailed message of the negative response code is described in chapter "Error-Codes".

7.22.2 Flag for disable adaptions while external adjust with SID 30_xx_07h services

General information:

The flag LV_AD_DIS_EXT_ADJ is set when an external adustment is done. The flag is set back to zero at ECU reset.

Application conditions:

Initialisation: 0 at reset

Activation: if a SID 30,xx,07 job is received

Formula section:

LV_AD_DIS_EXT_ADJ = 1 *// if a SID 30,xx,07 job is received*

7.22.3 Activation conditions for SID 30_xx_07h services

General information:

All SID 30_xx_07 services (external adjust of actuators) are only allowed if several activation conditions are fulfilled. The activation condition(s), which has/have to be fulfilled, can be chosen for every adjustment service via a calibratable bitmask in the table ID_IDX_ACT_DS_CONF. In this table one axis is the identifier of the service. On the other axis on position 0 a word can be calibrated for every service (positions 1...3 in the table are reserved for maximal 3 variables from the DDLI- lists for the read- services, see chapter "actuator diagnosis read value " below). The activation- condition- word must be calibrated in the following way: In the bitmask every activation condition, which has to be checked, must be set to "1". The possible conditions with their positions are listed in the table below. The combination of the conditions gives the hex- value of the word, which has to be written to ID_IDX_ACT_DS_CONF (on position 0 and position with service identifier).

Example: On Position (A1;0) in the table ID_IDX_ACT_DS_CONF 1h (= 0000 0000 0000 0001 bin) is calibrated. So the service 30_A1_07 is only activated if LV_IGK = 1.

If position (A1;0) is calibrated with 81h (0000 0000 1000 0001 bin), service 30_A1_07 is only activated if LV_IGK = 1 and VB > C_VB_MIN_KWP.

Formula section:

Condition name	BitValue	Conditions that have to be fullfilled
Diagnosis (always checked)	00 00	N < C_N_MAX_KWP
Kl15 ON	00 01	LV_IGK = 1
N = 0	00 02	LV_ES = 1
Idle speed	00 04	LV_IS = 1
N > N_MIN	00 08	N > C_N_MIN_KWP
v = 0	00 10	VS < C_VS_MAX_KWP
Engine warm	00 20	TCO > C_TCO_MIN_KWP
Engine temperature normal	00 40	TCO < C_TCO_MAX_KWP
Batt normal	00 80	VB > C_VB_MIN_KWP
NEUTRAL	01 00	

```
If(1)      GEAR_INFO = 0
Then(1)
           If(2) LV_AT = 0 and LV_VAR_AMT = 0 and
                               LV_VAR_TCT = 0
           Then(2)
                      If(3)      LV_CS = 0
                      Then(3)  activation
                      Endif(3)
           Else(2)
                      If(3) LV_VAR_AMT = 1 or LV_VAR_TCT = 1
                      Then(3)
                                 If(4) LV_BRAKE_DET = 0
                                 Then(4)  activation
                                 Endif(4)
                      Else(3)
                                 If(4) LV_AT = 1
                                 Then(4) activation
                                 Endif(4)
                      Endif(3)
           Endif(2)
Endif(1)
```

Fuel pump OFF	02 00	LV_ES = 1 **AND** N_EFP_AV = 0
Minimum oil pressure	04 00	POIL > C_POIL_MIN_KWP and N> C_N_MIN_KWP_POIL
Maxmum oil pressure	08 00	POIL < C_POIL_MAX_KWP and N< C_N_MAX_KWP_POIL
Minimum fuel pressure	10 00	FUP > C_FUP_MIN_KWP
Maxmum fuel pressure	20 00	FUP < C_FUP_MAX_KWP
nn	40 00	0
nn	80 00	0

7.22.4 30_xx_01h - actuator diagnosis read value -

General information:

With this service actual actuator values can be read. The variable(s), which has to be read, can be choosen out of several tables in software (DDLI- lists) via a calibratable identifier. The DDLI- lists are described in the chapter "Introduction." The identifier is a word value which represents the place in the DDLI- list. The calibration can be done within the table ID_IDX_ACT_DS_CONF. In this table one axis is the identifier of the service. On the other axis on positions 1...3 maximal 3 variables from the DDLI- lists can be chosen via the DDLI-identifier. (Position 0 in the table is reserved for activation conditions of adjustment jobs 30_xx_07; see chapter "activation conditions" above.)

The telegram length is variable and depends on the number and length (byte, word) of the chosen variable(s). If an illegal identifier is calibrated on position 1...3 in the table ID_IDX_ACT_DS_CONF (e.g. not defined identifiers) the service does not give back an error message! Instead of this the next valid calibrated variable is given on the next position in telegram.

Example: If the first variable is calibrated valid, it becomes the first position in the telegram (depending on the length of the variable (byte or word) this can be the first one or the first two data bytes of the telegramm.) The second position is not calibrated valid, but the third is. So the variable, calibrated on third position is given back on second position in telegram.

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK =1

Deactivation: not activation

Service location: see list of "implemented diagnostic services"

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2	InputOutputLocalIdentifier -	M	XXh	IOLI_
	[			
	Bremslichtschalter		02h	
	Bremslichttestschalter		03h	
	Kupplungsschalter		04h	
	Oeldruckschalter		05h	
	Sporttaster		06h	
	Nullgangssensor		07h	
	Ansauglufttemperatur 1		0Ah	
	Ansauglufttemperatur 2		0Bh	
	Motortemperatur		0Ch	
	Kuehlerauslasstemperatur		0Dh	
	Oelsensor		0Eh	
	Steuergeraete-Innentemperatur		10h	
	Abgastemperatur		12h	
	Abgastempertur 2		13h	
	Umgebungsdruck		17h	
	Saugrohrdruck1 / Ladedruck1		18h	
	Saugrohrdruck2 / Ladedruck2		19h	
	Raildrucksenor		1Ah	
	Kl.15 Spannung		1Bh	
	Kl.87 Spannung / Versorgung DME (Digitale Motor Elektronik)		1Ch	
	Fahrerwunschversorgung 1		1Eh	
	Fahrerwunschversorgung 2		1Fh	
	Versorgungsspannung VVL Sensor (not MSD85)		20h	
	Versorgung HPDI-Ventil (only MSD85)		20h	
	Lambdasonde vor Kat Bank1		21h	
	Lambdasonde hinter Kat Bank1		22h	
	Lambdasonde vor Kat Bank2		23h	
	Lambdasonde hinter Kat Bank2		24h	
	Luftmassenmesser 1		25h	
	Batteriesensor		27h	
	Fahrerwunsch 1		28h	
	Fahrerwunsch 2		29h	
	Drosselklappe		2Ah	
	Sekundaerluft HFM (Heissfilm Luftmassenmesser)		2Eh	

	Taster Tempomat		2Fh	
	Klopfbaustein 1		30h	
	Klopfbaustein 2		31h	
	Generator Sollspannung BSD (Bit Serielle Datenschnittstelle)		32h	
	Oeldrucksensor		37h	
	NOx-Sensor		3Bh	
	Kraftsoffniederdrucksensor		3Fh	
	Bremsunterdrucksensor		40h	
	Raildrucksensor 2		43h	
	Klopfbaustein 3		44h	
	Klopfbaustein 4		45h	
	Mass air flow 2		46h	
	Radiator shutter		80h	
	HPD-Relais		82h	
	Wasserpumpe Turbolader		83h	
	Drosselklappenstellergehäuseheizung 2		84h	
	Drosselklappe 2		85h	
	(second) HPD5-Mengensteuerventil		86h	
	Abgasklappe 2		87h	
	Vanos Einlass Ventil Bank2		88h	
	Vanos Auslass Ventil Bank2		89h	
	Umluftventil Bank2		8Ah	
	second Sekundaerluftventil		8Bh	
	Drosselklappenstellergehäuseheizung		9Eh	
	Gesteuerte Luftfuehrung (untere Klappe)		A4h	
	Oeldruck-Regelung		ABh	
	Oeldruckventil		ACh	
	Kurbelgehaeuseentlueftungsheizung		ADh	
	Variable Sauganlage (DISA) Klappe2		Aeh	
	Motorlager		B2h	
	Umluftventil Bank1		B5h	
	Mengensteuerventil		BDh	
	Abgasrückführungsventil		BEh	
	elektr. Wasserpumpe ueber BSD (Bit Serielle Datenschnittstelle)		BFh	
	Abgasklappe		C1h	
	Soundklappe		C2h	
	Gesteuerte Luftfuehrung (obere Klappe)		C3h	
	Startrelais		C4h	
	Variable Sauganlage (DISA) Klappe		C6h	
	Klimakompressor-Relais		C7h	
	E-Box-Luefter		C8h	
	Kennfeldthermostat		C9h	
	Sekundaerluftventil		CAh	
	Sekundaerluftpumpe		CBh	
	Diagnosemodul-Tank Leckage Pumpe		CCh	
	Diagnosemodul-Tank Leckage Ventil		CDh	
	Diagnosemodul-Tank Leckage Heizung		CEh	
	Tankentlueftungsventil		CFh	
	Lambdasondenheizung vor Kat Bank1		D0h	
	Lambdasondenheizung hinter Kat Bank1		D1h	
	Lambdasondenheizung vor Kat Bank2		D2h	
	Lambdasondenheizung hinter Kat Bank2		D3h	
	MIL (Malfunction Indicator Lamp)		D4h	
	Fahrgeschwindigkeitsregler-Lampe		D5h	
	EML (Engine Malfunction Lamp)		D6h	
	Elektrische Kraftstoffpumpe 1		D8h	
	E-Box-Luefter		DAh	
	VVT-Entlastungsrelais		DCh	
	VVT		DDh	
	Einspritzventil 1		E1h	
	Einspritzventil 2		E2h	

	Einspritzventil 3		E3h	
	Einspritzventil 4		E4h	
	Einspritzventil 5		E5h	
	Einspritzventil 6		E6h	
	Einspritzventil 7		E7h	
	Einspritzventil 8		E8h	
	Vanos Einlass Ventil		EDh	
	Vanos Auslass Ventil		EEh	
	HPD5-Mengensteuerventil 1		EFh	
	]		]	
3	inputOutputControlParameter	M	01h	IOCP_RCS

InputOutputControlByLocalIdentifier positive Response

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
1	**inputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier -	M	XXh	IOLID
	[			
	Bremslichtschalter		02h	
	Bremslichttestschalter		03h	
	Kupplungsschalter		04h	
	Oeldruckschalter		05h	
	Sporttaster		06h	
	Nullgangssensor		07h	
	Ansauglufttemperatur 1		0Ah	
	Ansauglufttemperatur 2		0Bh	
	Motortemperatur		0Ch	
	Kuehlerauslasstemperatur		0Dh	
	Oelsensor		0Eh	
	Steuergeraete-Innentemperatur		10h	
	Abgastemperatur		12h	
	Abgastempertatur 2		13h	
	Umgebungsdruck		17h	
	Saugrohrdruck1 / Ladedruck1		18h	
	Saugrohrdruck2 / Ladedruck2		19h	
	Raildrucksenor		1Ah	
	Kl.15 Spannung		1Bh	
	Kl.87 Spannung / Versorgung DME (Digitale Motor Elektronik)		1Ch	
	Fahrerwunschversorgung 1		1Eh	
	Fahrerwunschversorgung 2		1Fh	
	Versorgungsspannung VVL Sensor (not MSD85)		20h	
	Versorgung HPDI-Ventil (only MSD85)		20h	
	Lambdasonde vor Kat Bank1		21h	
	Lambdasonde hinter Kat Bank1		22h	
	Lambdasonde vor Kat Bank2		23h	
	Lambdasonde hinter Kat Bank2		24h	
	Luftmassenmesser 1		25h	
	Batteriesensor		27h	
	Fahrerwunsch 1		28h	
	Fahrerwunsch 2		29h	
	Drosselklappe		2Ah	
	Sekundaerluft HFM (Heissfilm Luftmassenmesser)		2Eh	
	Taster Tempomat		2Fh	
	Klopfbaustein 1		30h	
	Klopfbaustein 2		31h	
	Generator Sollspannung BSD (Bit Serielle Datenschnittstelle)		32h	
	Oeldrucksensor		37h	
	NOx-Sensor		3Bh	
	Kraftsoffniederdrucksensor		3Fh	
	Bremsunterdrucksensor		40h	

	Raildrucksensor 2		43h	
	Klopfbaustein 3		44h	
	Klopfbaustein 4		45h	
	Mass air flow 2		46h	
	Radiator shutter		80h	
	HPD-Relais		82h	
	Wasserpumpe Turbolader		83h	
	Drosselklappenstellergehäuseheizung 2		84h	
	Drosselklappe 2 ansteuern		85h	
	(second) HPD5-Mengensteuerventil		86h	
	Abgasklappe 2		87h	
	Vanos Einlass Ventil Bank2		88h	
	Vanos Auslass Ventil Bank2		89h	
	Umluftventil Bank2		8Ah	
	second Sekundaerluftventil		8Bh	
	Drosselklappenstellergehäuseheizung		9Eh	
	Gesteuerte Luftfuehrung (untere Klappe)		A4h	
	Oeldruck-Regelung		ABh	
	Oeldruckventil		ACh	
	Kurbelgehaeuseentlueftungsheizung		ADh	
	Variable Sauganlage (DISA) Klappe2		AEh	
	Motorlager		B2h	
	Umluftventil Bank1		B5h	
	Mengensteuerventil		BDh	
	Abgasrückführungsventil		BEh	
	elektr. Wasserpumpe ueber BSD (Bit Serielle Datenschnittstelle)		BFh	
	Abgasklappe		C1h	
	Soundklappe		C2h	
	Gesteuerte Luftfuehrung (obere Klappe)		C3h	
	Startrelais		C4h	
	Variable Sauganlage (DISA) Klappe		C6h	
	Klimakompressor-Relais		C7h	
	E-Box-Luefter		C8h	
	Kennfeldthermostat		C9h	
	Sekundaerluftventil		CAh	
	Sekundaerluftpumpe		CBh	
	Diagnosemodul-Tank Leckage Pumpe		CCh	
	Diagnosemodul-Tank Leckage Ventil		CDh	
	Diagnosemodul-Tank Leckage Heizung		CEh	
	Tankentlueftungsventil		CFh	
	Lambdasondenheizung vor Kat Bank1		D0h	
	Lambdasondenheizung hinter Kat Bank1		D1h	
	Lambdasondenheizung vor Kat Bank2		D2h	
	Lambdasondenheizung hinter Kat Bank2		D3h	
	MIL (Malfunction Indicator Lamp)		D4h	
	Fahrgeschwindigkeitsregler-Lampe		D5h	
	EML (Engine Malfunction Lamp)		D6h	
	Elektrische Kraftstoffpumpe 1		D8h	
	E-Luefter		DAh	
	VVT-Entlastungsrelais		DCh	
	VVT		DDh	
	Einspritzventil 1		E1h	
	Einspritzventil 2		E2h	
	Einspritzventil 3		E3h	
	Einspritzventil 4		E4h	
	Einspritzventil 5		E5h	
	Einspritzventil 6		E6h	
	Einspritzventil 7		E7h	
	Einspritzventil 8		E8h	
	Vanos Einlass Ventil		EDh	
	Vanos Auslass Ventil		EEh	

	HPD5-Mengensteuerventil 1]		EFh]	
3	inputOutputControlParameter	M	01h	IOCP_RCS
4	Data Byte depending on calibration	M	XXh	xx
...	...	...	...	...
n	Data Byte depending on calibration	M	XXh	xx

Table InputOutputControlByLocalIdentifier positive response message

InputOutputControlByLocalIdentifier negative Response

Global Negative Responses

7.22.5 30_xx_00h - stop actuator diagnosis -

General information:

When this service is received, the actuator diagnosis function stops and control is given back to the ECU control.

xx in the headline represents the identifier (00 ... FFh) of the actuator diagnosis service.

Application conditions:

***Initialisation**:* 0 at reset

***Activation**:* LV_IGK =1

Deactivation: not activation

Service location: see list of “implemented diagnostic services”

InputOutputControlByLocalIdentifier Request

.

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
1	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2	InputOutputLocalIdentifier - [Kl.87 Spannung / Versorgung DME (Digitale Motor Elektronik) Versorgung HPDI-Ventil Drosselklappe Generator Sollspannung BSD (Bit Serielle Datenschnittstelle) Radiator shutter HPD-Relais Wasserpumpe Turbolader Drosselklappenstellergehäuseheizung 2 Drosselklappe 2 (second) HPD5-Mengensteuerventil Abgasklappe 2 Vanos Einlass Ventil Bank2 Vanos Auslass Ventil Bank2	M	XXh [1Ch 20h 2Ah 32h 80h 82h 83h 84h 85h 86h 87h 88h 89h	IOLI_XX

	Umluftventil Bank2		8Ah	
	second Sekundaerluftventil		8Bh	
	Drosselklappenstellergehäuseheizung		9Eh	
	Gesteuerte Luftfuehrung (untere Klappe)		A4h	
	Oeldruck-Regelung		ABh	
	Oeldruckventil		ACh	
	Kurbelgehaeuseentlueftungsheizung		ADh	
	Variable Sauganlage (DISA) Klappe2		AEh	
	Motorlager		B2h	
	Umluftventil Bank1		B5h	
	Ladedrucksteller 1		B6h	
	Ladedrucksteller 2		B7h	
	Mengensteuerventil		BDh	
	Abgasrückführungsventil		BEh	
	elektr. Wasserpumpe ueber BSD (Bit Serielle Datenschnittstelle)		BFh	
	Abgasklappe		C1h	
	Soundklappe		C2h	
	Gesteuerte Luftfuehrung (obere Klappe)		C3h	
	Startrelais		C4h	
	Variable Sauganlage (DISA) Klappe		C6h	
	Klimakompressor-Relais		C7h	
	E-Box-Luefter		C8h	
	Kennfeldthermostat		C9h	
	Sekundaerluftventil		CAh	
	Sekundaerluftpumpe		CBh	
	Diagnosemodul-Tank Leckage Pumpe		CCh	
	Diagnosemodul-Tank Leckage Ventil		CDh	
	Diagnosemodul-Tank Leckage Heizung		CEh	
	Tankentlueftungsventil		CFh	
	Lambdasondenheizung vor Kat Bank1		D0h	
	Lambdasondenheizung hinter Kat Bank1		D1h	
	Lambdasondenheizung vor Kat Bank2		D2h	
	Lambdasondenheizung hinter Kat Bank2		D3h	
	MIL (Malfunction Indicator Lamp)		D4h	
	Fahrgeschwindigkeitsregler-Lampe		D5h	
	EML (Engine Malfunction Lamp)		D6h	
	Elektrische Kraftstoffpumpe 1		D8h	
	E-Box-Luefter		DAh	
	VVT-Entlastungsrelais		DCh	
	VVT		DDh	
	Einspritzventil 1		E1h	
	Einspritzventil 2		E2h	
	Einspritzventil 3		E3h	
	Einspritzventil 4		E4h	
	Einspritzventil 5		E5h	
	Einspritzventil 6		E6h	
	Vanos Einlass Ventil		EDh	
	Vanos Auslass Ventil		EEh	
	HPD5-Mengensteuerventil 1		EFh	
	]		]	
3	inputOutputControlParameter	M	00h	IOCP_RCTECU

InputOutputControlByLocalIdentifier positive Response

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
1	**inputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier - [	M	XXh [	IOLI_

	Kl.87 Spannung / Versorgung DME (Digitale Motor Elektronik)		1Ch	
	Versorgung HPDI-Ventil		20h	
	Drosselklappe		2Ah	
	Generator Sollspannung BSD (Bit Serielle Datenschnittstelle)		32h	
	Radiator shutter		80h	
	HPD-Relais		82h	
	Wasserpumpe Turbolader		83h	
	Drosselklappenstellergehäuseheizung 2		84h	
	Drosselklappe 2		85h	
	(second) HPD5-Mengensteuerventil		86h	
	Abgasklappe 2		87h	
	Vanos Einlass Ventil Bank2		88h	
	Vanos Auslass Ventil Bank2		89h	
	Umluftventil Bank2		8Ah	
	second Sekundaerluftventil		8Bh	
	Drosselklappenstellergehäuseheizung		9Eh	
	Gesteuerte Luftfuehrung (untere Klappe)		A4h	
	Oeldruck-Regelung		ABh	
	Oeldruckventil		ACh	
	Kurbelgehaeuseentlueftungsheizung		ADh	
	Variable Sauganlage (DISA) Klappe2		AEh	
	Motorlager		B2h	
	Umluftventil Bank1		B5h	
	Ladedrucksteller 1		B6h	
	Ladedrucksteller 2		B7h	
	Mengensteuerventil		BDh	
	Abgasrückführungsventil		BEh	
	elektr. Wasserpumpe ueber BSD (Bit Serielle Datenschnittstelle)		BFh	
	Abgasklappe		C1h	
	Soundklappe		C2h	
	Gesteuerte Luftfuehrung (obere Klappe)		C3h	
	Startrelais		C4h	
	Variable Sauganlage (DISA) Klappe		C6h	
	Klimakompressor-Relais		C7h	
	E-Box-Luefter		C8h	
	Kennfeldthermostat		C9h	
	Sekundaerluftventil		CAh	
	Sekundaerluftpumpe		CBh	
	Diagnosemodul-Tank Leckage Pumpe		CCh	
	Diagnosemodul-Tank Leckage Ventil		CDh	
	Diagnosemodul-Tank Leckage Heizung		CEh	
	Tankentlueftungsventil		CFh	
	Lambdasondenheizung vor Kat Bank1		D0h	
	Lambdasondenheizung hinter Kat Bank1		D1h	
	Lambdasondenheizung vor Kat Bank2		D2h	
	Lambdasondenheizung hinter Kat Bank2		D3h	
	MIL (Malfunction Indicator Lamp)		D4h	
	Fahrgeschwindigkeitsregler-Lampe		D5h	
	EML (Engine Malfunction Lamp)		D6h	
	Elektrische Kraftstoffpumpe 1		D8h	
	E-Luefter		DAh	
	VVT-Entlastungsrelais		DCh	
	VVT		DDh	
	Einspritzventil 1		E1h	
	Einspritzventil 2		E2h	
	Einspritzventil 3		E3h	
	Einspritzventil 4		E4h	
	Einspritzventil 5		E5h	
	Einspritzventil 6		E6h	
	Vanos Einlass Ventil		EDh	
	Vanos Auslass Ventil		EEh	

	HPD5-Mengensteuerventil 1]		EFh]	
3	inputOutputControlParameter	M	00h	IOCP_RCTECU

InputOutputControlByLocalIdentifier negative Response

Global Negative Responses

7.22.6 30_1C07 - Kl.87 Spannung / Versorgung DME (Digitale Motor Elektronik) ansteuern

Formula section:

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kl.87 Spannung / Versorgung	M	1Ch	IOLI_UVSG
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_UVSG
5d	InputOutputControlParameter - LV_RLY_MAIN_EXT_ADJ	M	XXh	CS_TV_UVSG
6d	InputOutputControlParameter - TIMEOUT_TIME_1C_KWP	M	XXh	CS_TO_UVSG

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	1Ch	IOLI_UVSG
3d	InputOutputControlParameter	M	07h	IOCP_UVSG

Table InputOutputControlByLocalIdentifier positive response message

InputOutputControlByLocalIdentifier negative Response

Global Negative Responses

7.22.7 30_2007 - Versorgung HPDI-Ventil steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	20h	IOLI_UVLSS
3	inputOutputControlParameter	M	07h	IOCP_STA
4	InputOutputControlParameter –not used	M	FFh	CS_PD_UVLSS
5	InputOutputControlParameter – LV_RLY_HPDI_EXT_ADJ	M	xxh	CS_LV_RLY_HPDI_EXT_ADJ
6	InputOutputControlParameter – TIMEOUT_TIME_20_KWP	M	xxh	CS_TO_UVLSS

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**InputOutputControlByLocalIdentifier Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier	M	20h	IOLI_UVLSS
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.8 30_2A07 - Drosselklappe ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLI**
2d	InputOutputLocalIdentifier - Drosselklappe	M	2Ah	IOLI_DK
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_DK
5d	InputOutputControlParameter - TPS_SP_EXT_ADJ	M	XXh	CS_TV_DK
6d	InputOutputControlParameter - TIMEOUT_TIME_2A_KWP	M	XXh	CS_TO_DK

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	2Ah	IOLI_DK
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.9 30_3207 - Generator Sollspannung BSD (Bit Serielle Datenschnittstelle) ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Generator Sollspannung BSD	M	32h	IOLI_UGEN
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PHY_UGEN
5d	inputOutputControlParameter - V_ALTER_SP_EXT_ADJ high byte	M	XXh	CS_PHY_UGEN
6d	InputOutputControlParameter - V_ALTER_SP_EXT_ADJ low byte	M	XXh	CS_PHY_UGEN

7d	InputOutputControlParameter - TIMEOUT_TIME_32_KWP	M	XXh	CS_TO_UGEN

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	32h	IOLI_UGEN
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.10 30_8007 – Radaitor shutter ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	80h	IOLI_HPDR
3	inputOutputControlParameter	M	07h	IOCP_STA
4	InputOutputControlParameter –not used	M	FFh	CS_PD_RAS
5	InputOutputControlParameter – LV_ACT_RAS_EXT_ADJ	M	xxh	CS_TV_RAS
6	InputOutputControlParameter - TIMEOUT_TIME_80_KWP	M	xxh	CS_TO_RAS

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**InputOutputControlByLocalIdentifier Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier	M	80h	IOLI_RAS
3	InputOutputControlParameter	M	07h	IOCP_RAS

InputOutputControlByLocalIdentifier Negative Response

7.22.11 30_8207 – HPD-Relais steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	82h	IOLI_HPDR
3	inputOutputControlParameter	M	07h	IOCP_STA
4	InputOutputControlParameter –not used	M	FFh	CS_PD_HPDR
5	InputOutputControlParameter – LV_ACT_RLY_VCV_EXT_ADJ	M	xxh	CS_TV_HPDR

6	InputOutputControlParameter - TIMEOUT_TIME_82_KWP	M	xxh	CS_TO_HPDR

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**InputOutputControlByLocalIdentifier Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier	M	82h	IOLI_HPDR
3	InputOutputControlParameter	M	07h	IOCP_HPDR

7.22.12 30_8307 – Wasserpumpe Turbolader steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	83h	IOLI_WAPUT
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter –not used	M	FFh	CS_PD_WAPUT
5	InputOutputControlParameter – N_REL_CWP_SP_2_EXT_ADJ	M	xxh	CS_TV_WAPUT
6	InputOutputControlParameter - TIMEOUT_TIME_83_KWP	M	xxh	CS_TO_WAPUT

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**InputOutputControlByLocalIdentifier Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier	M	83h	IOLI_WAPUT
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.13 30_8407 - Drosselklappenstellergehaeuseheizung 2 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier – Drosselklappenstellergehäuseheizung 2	M	84h	IOLI_DKH2
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_DKH2
5d	InputOutputControlParameter - LV_ACT_RLY_MTC_2_HEAT_EXT_ADJ	M	XXh	CS_TV_DKH2
6d	InputOutputControlParameter - TIMEOUT_TIME_84_KWP	M	XXh	CS_TO_DKH2

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	84h	IOLI_DKH2
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.14 30_8507 – Drosselklappe 2 steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	85h	IOLI_DKP2
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter – not used	M	FFh	CS_PD_DKP2
5	InputOutputControlParameter – TPS_SP_2_EXT_ADJ	M	xxh	CS_TPS_SP_EXT_ADJ[2]
6	InputOutputControlParameter –TIMEOUT_TIME_85_KWP	M	xxh	CS_TO_DKH2

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**InputOutputControlByLocalIdentifier Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier	M	85h	IOLI_DKP2
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.15 30_8607 – (second) HPD5-Mengensteuerventil steuern

Second HPD5-Menegensteuerventil for MSD85 only:

InputOutputControlByLocalIdentifier Request

Data Byte	**Parameter Name**	**Cvt**	**Hex Value**	**Mnemonic**
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	85h	IOLI_MSV2
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter – not used	M	FFh	CS_PD_MSV2
5	InputOutputControlParameter – arqtmsv_w_rb[2] (high byte)	M	xxh	CS_
6	InputOutputControlParameter – arqtmsv_w_rb[2] (low byte)	M	xxh	
7	InputOutputControlParameter –TIMEOUT_TIME_86_KWP	M	xxh	CS_TO_MSV2

Hint: activation with B_rqtmsv_rb[2]

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**InputOutputControlByLocalIdentifier Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier	M	86h	IOLI_MSV2
3	InputOutputControlParameter	M	07h	IOCP_STA

HPD5-Mengensteuerventil for MSD80 4cyl only:

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	85h	IOLI_MSV2
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter – not used	M	FFh	CS_PD_MSV2
5	InputOutputControlParameter – arqtmsv_w_rb (high byte)	M	xxh	CS_
6	InputOutputControlParameter – arqtmsv_w_rb (low byte)	M	xxh	
7	InputOutputControlParameter –TIMEOUT_TIME_86_KWP	M	xxh	CS_TO_MSV2

Hint: activation with B_rqtmsv_rb

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	InputOutputControlByLocalIdentifier Response SId	M	70h	IOCBLIDPR
2	InputOutputLocalIdentifier	M	86h	IOLI_MSV2
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.16 30_8707 – Abgasklappe 2 steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	87	IOLI_AGK2
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter – not used	M	FFh	CS_PD_AGK2
5	InputOutputControlParameter – LV_ACT_EF_2_EXT_ADJ	M	xxh	CS_LV_ACT_EF_EXT_ADJ[2]
6	InputOutputControlParameter – TIMEOUT_TIME_87_KWP	M	xxh	CS_TO_AGK2

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	InputOutputControlByLocalIdentifier Response SId	M	70h	IOCBLIDPR

2	InputOutputLocalIdentifier	M	87h	IOLI_AGK2
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.17 30_8807 - Vanos Einlass Ventil Bank2 steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	88	IOLI_ENWS2
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter -	M	FFh	CS_PD_ENWS2
5	InputOutputControlParameter - CAM_SP_2_IN_EXT_ADJ (high byte)	M	Xxh	CS_CAM_SP_2_IN_EXT_ADJ
6	InputOutputControlParameter – CAM_SP_2_IN_EXT_ADJ (low byte)	M	Xxh	
7	InputOutputControlParameter – TIMEOUT_TIME_88_KWP	M	Xxh	CS_TO_ENWS2

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	InputOutputControlByLocalIdentifier Response SId	M	70h	IOCBLIDPR
2	InputOutputLocalIdentifier	M	88h	IOLI_ENWS2
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.18 30_8907 – Vanos Auslass Ventil Bank 2 steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	89	IOLI_ANSW2
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter - Periodendauer Bank 2	M	FFh	CS_PD_ANSW2
5	InputOutputControlParameter - CAM_SP_2_EX_EXT_ADJ (high byte)	M	xxh	CS_CAM_SP_2_EX_EXT_ADJ
6	InputOutputControlParameter - CAM_SP_2_EX_EXT_ADJ (low byte)	M	xxh	
7	InputOutputControlParameter – TIMEOUT_TIME_89_KWP	M	xxh	CS_TO_ANWS2

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	InputOutputControlByLocalIdentifier Response SId	M	70h	IOCBLIDPR
2	InputOutputLocalIdentifier	M	89h	IOLI_ANSW2
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.19 30_8A07 – Umluftventil Bank 2 steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	8A	IOLI_ULV2
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter – not used	M	FFh	CS_PD_ULV2
5	InputOutputControlParameter – RFPPWM_EXT_ADJ[2]	M	xxh	CS_TS_ULV2
6	InputOutputControlParameter - TIMEOUT_TIME_8A_KWP	M	xxh	CS_TO_ULV2

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	InputOutputControlByLocalIdentifier Response SId	M	70h	IOCBLIDPR
2	InputOutputLocalIdentifier	M	8Ah	IOLI_ULV2
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.20 30_8B07 - Sekundearluftventil 2 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier – Sekundearluftventil 2	M	8Bh	IOLI_SLV2
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_SLV2
5d	InputOutputControlParameter - LV_ACT_SAV_2_EXT_ADJ	M	XXh	CS_TV_SLV2
6d	InputOutputControlParameter - TIMEOUT_TIME_8B_KWP	M	XXh	CS_TO_SLV2

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	CAh	IOLI_SLV2
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.21 30_9E07 - Drosselklappenstellergehäuseheizung ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Drosselklappenstellergehäuseheizung	M	9Eh	IOLI_DKH
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_DKH
5d	InputOutputControlParameter - LV_ACT_RLY_MTC_HEAT_EXT_ADJ	M	XXh	CS_TV_DKH
6d	InputOutputControlParameter - TIMEOUT_TIME_9E_KWP	M	XXh	CS_TO_DKH

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	9Eh	IOLI_DKH
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.22 30_A407 - Gesteuerte Luftfuehrung (untere Klappe) ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLI**
2d	InputOutputLocalIdentifier - Gesteuerte Luftfuehrung ansteuern	M	A4h	IOLI_GLF2
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_GLF2
5d	InputOutputControlParameter - LV_ACT_ECRAS_DOWN_EXT_ADJ	M	XXh	CS_TV_GLF2
6d	InputOutputControlParameter - TIMEOUT_TIME_A4_KWP	M	XXh	CS_TO_GLF2

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIPR**
2d	InputOutputLocalIdentifier	M	A4h	IOLI_GLF2
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.23 30_AB07 – Oeldruck-Regelung (Geregeltes Oeldrucksystem) ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request Sid**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier – Ölsolldruck	M	ABh	IOLI_ODR
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_P_OELSOL_TST
5d	inputOutputControlParameter - POIL_SP_EXT_ADJ	M	XXh	CS_P_OELSOL_TST
7d	InputOutputControlParameter - TIMEOUT_TIME_AB_KWP	M	XXh	CS_TO_ODR

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	ABh	IOLI_ODR
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.24 30_AC07 - Oeldruckventil (Geregeltes Oeldrucksystem) ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Generator Sollspannung BSD	M	ACh	IOLI_ODV
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_ODV
5d	inputOutputControlParameter - POIL_PWM_EXT_ADJ	M	XXh	CS_PD_ODV
6d	InputOutputControlParameter - TIMEOUT_TIME_AC_KWP	M	XXh	CS_TO_ODV

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	ACh	IOLI_ODV
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.25 30_AD07 - Kurbelgehaeuseentlueftungsheizung ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kurbelgehaeuseentlueftungsheizung	M	ADh	IOLI_KGEH
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_KGEH
5d	InputOutputControlParameter - LV_ACT_RLY_CRCV_HEAT_EXT_ADJ	M	XXh	CS_TV_KGEH
6d	InputOutputControlParameter - TIMEOUT_TIME_AD_KWP	M	XXh	CS_TO_KGEH

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	ADh	IOLI_KGEH
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.26 30_AE07 - Variable Sauganlage (DISA) Klappe2 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Variable Sauganlage (DISA) Klappe2	M	AEh	IOLI_DISA2
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_DISA2
5d	InputOutputControlParameter - LV_VIM_2_EXT_ADJ	M	XXh	CS_TV_DISA2
6d	InputOutputControlParameter - TIMEOUT_TIME_AE_KWP	M	XXh	CS_TO_DISA2

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	AEh	IOLI_DISA2
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.27 30_B207 - Motorlager ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kurbelgehaeuseentlueftungsheizung	M	B2h	IOLI_
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_
5d	inputOutputControlParameter - LV_SWI_AEB_EXT_ADJ	M	XXh	CS_
6d	InputOutputControlParameter - TIMEOUT_TIME_B2_KWP	M	XXh	CS_TO_

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	B2h	IOLI_KGEH
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.28 30_B507 - Umluftventil Bank 1 steuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2	inputOutputLocalID	M	B5	IOLI_ULV
3	inputOutputControlParameter	M	07	IOCP_STA
4	InputOutputControlParameter – not used	M	FFh	CS_PD_ULV
5	InputOutputControlParameter - – RFPPWM_EXT_ADJ[1]	M	xxh	CS_TV_ULV
6	InputOutputControlParameter – TIMEOUT_TIME_B5_KWP	M	xxh	CS_TO_ULV

InputOutputControlByLocalIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	InputOutputControlByLocalIdentifier Response SId	M	70h	IOCBLIDPR
2	InputOutputLocalIdentifier	M	xxh	IOLI_ULV
3	InputOutputControlParameter	M	07h	IOCP_STA

7.22.29 30_B607 - Ladedrucksteller 1 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kurbelgehaeuseentlueftungsheizung	M	B6h	IOLI_
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_
5d	inputOutputControlParameter – WGPWM_EXT_ADJ[1] (high byte)	M	XXh	CS_
6d	InputOutputControlParameter - WGPWM_EXT_ADJ[1] (low byte)	M	XXh	CS_
7d	InputOutputControlParameter - TIMEOUT_TIME_B6_KWP	M	XXh	CS_TO_

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	B6h	IOLI_KGEH
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.30 30_B707 - Ladedrucksteller 2 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kurbelgehaeuseentlueftungsheizung	M	B7h	IOLI_
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_
5d	inputOutputControlParameter - WGPWM_EXT_ADJ[2] (high byte)	M	XXh	CS_
6d	InputOutputControlParameter - WGPWM_EXT_ADJ[2] (low byte)	M	XXh	CS_
7d	InputOutputControlParameter - TIMEOUT_TIME_B7_KWP	M	XXh	CS_TO_

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	B7h	IOLI_KGEH
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.31 30_BD07 - Mengensteuerventil ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kurbelgehaeuseentlueftungsheizung	M	BDh	IOLI_
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_
5d	inputOutputControlParameter - FUP_SP_EXT_ADJ (high byte)	M	XXh	CS_
6d	InputOutputControlParameter - FUP_SP_EXT_ADJ (low byte)	M	XXh	CS_
7d	InputOutputControlParameter - TIMEOUT_TIME_AD_KWP	M	XXh	CS_TO_

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	BDh	IOLI_KGEH
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.32 30_BE07 - Abgasrückführungsventil ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kurbelgehaeuseentlueftungsheizung	M	BDh	IOLI_
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_
5d	inputOutputControlParameter - OPG_SP_ACR_EXT_ADJ (high byte)	M	XXh	CS_
6d	InputOutputControlParameter - OPG_SP_ACR_EXT_ADJ (low byte)	M	XXh	CS_
7d	InputOutputControlParameter - TIMEOUT_TIME_AD_KWP	M	XXh	CS_TO_

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	BDh	IOLI_KGEH
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.33 30_BF07 - elektr. Wasserpumpe ueber BSD (Bit Serielle Datenschnittstelle) ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - elektr. Wasserpumpe ueber BSD	M	BFh	IOLI_EWAP
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_EWAP
5d	InputOutputControlParameter - N_REL_CWP_SP_EXT_ADJ	M	XXh	CS_TV_EWAP
6d	InputOutputControlParameter - TIMEOUT_TIME_BF_KWP	M	XXh	CS_TO_EWAP

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	BFh	IOLI_EWAP
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.34 30_C107 - Abgasklappe ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Abgasklappe ansteuern	M	C1h	IOLI_AGK
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_AGK
5d	InputOutputControlParameter - LV_ACT_EF_EXT_ADJ	M	XXh	CS_TV_AGK
6d	InputOutputControlParameter - TIMEOUT_TIME_C1_KWP	M	XXh	CS_TO_AGK

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	C1h	IOLI_AGK
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.35 30_C207 - Soundklappe ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Soundklappe ansteuern	M	C2h	IOLI_SOK
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_SOK
5d	InputOutputControlParameter - LV_ACT_SOF_EXT_ADJ	M	XXh	CS_TV_SOK
6d	InputOutputControlParameter - TIMEOUT_TIME_C2_KWP	M	XXh	CS_TO_SOK

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	C2h	IOLI_SOK
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.36 30_C307 - Gesteuerte Luftfuehrung ansteuern (obere Klappe)

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLI**
2d	InputOutputLocalIdentifier - Gesteuerte Luftfuehrung ansteuern	M	C3h	IOLI_GLF
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_GLF
5d	InputOutputControlParameter - LV_ACT_ECRAS_UP_EXT_ADJ	M	XXh	CS_TV_GLF
6d	InputOutputControlParameter - TIMEOUT_TIME_C3_KWP	M	XXh	CS_TO_GLF

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIPR**
2d	InputOutputLocalIdentifier	M	C3h	IOLI_GLF
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.37 30_C407 - Startrelais ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Startrelais ansteuern	M	C4h	IOLI_SR
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_SR
5d	InputOutputControlParameter - LV_ACT_RLY_ST_EXT_ADJ	M	XXh	CS_TV_SR
6d	InputOutputControlParameter - TIMEOUT_TIME_C4_KWP	M	XXh	CS_TO_SR

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	C4h	IOLI_SR
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.38 30_C607 - Variable Sauganlage (DISA) Klappe ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Variable Sauganlage (DISA) Klappe	M	C6h	IOLI_DISA
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_DISA
5d	InputOutputControlParameter - LV_VIM_1_EXT_ADJ	M	XXh	CS_TV_DISA
6d	InputOutputControlParameter - TIMEOUT_TIME_C6_KWP	M	XXh	CS_TO_DISA

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	C6h	IOLI_DISA
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.39 30_C707 - Klimakompressor-Relais ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Klimakompressor-Relais ansteuern	M	C7h	IOLI_KOREL
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_KOREL
5d	InputOutputControlParameter - LV_ACT_ACCOUT_RLY_EXT_ADJ	M	XXh	CS_TV_KOREL
6d	InputOutputControlParameter - TIMEOUT_TIME_C7_KWP	M	XXh	CS_TO_KOREL

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	C7h	IOLI_KOREL
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.40 30_C807 - E-Box-Luefter ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - E-Box-Luefter	M	C8h	IOLI_EBL
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_EBL
5d	InputOutputControlParameter - LV_ACT_EBOX_CFA_EXT_ADJ	M	XXh	CS_TV_EBL
6d	InputOutputControlParameter - TIMEOUT_TIME_C8_KWP	M	XXh	CS_TO_EBL

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	C8h	IOLI_EBL
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.41 30_C907 - Kennfeldthermostat ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kennfeldthermostat	M	C9h	IOLI_KFT
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_KFT
5d	InputOutputControlParameter - ECTPWM_EXT_ADJ	M	XXh	CS_TV_KFT
6d	InputOutputControlParameter - TIMEOUT_TIME_C9_KWP	M	XXh	CS_TO_KFT

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	C9h	IOLI_KFT
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.42 30_CA07 - Sekundarluftventil ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Sekundarluftventil	M	CAh	IOLI_SLV
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_SLV
5d	InputOutputControlParameter - LV_ACT_SAV_EXT_ADJ	M	XXh	CS_TV_SLV
6d	InputOutputControlParameter - TIMEOUT_TIME_1C_KWP	M	XXh	CS_TO_SLV

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	CAh	IOLI_SLV
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.43 30_CB07 - Sekundaerluftpumpe ansteuern

Additional Application conditions:

Initialisation: 0 at reset

Activation: depending on calibratable conditions

see chapter "activation conditions" above

IF LV_TEMP_MAX_SAP_COIL = 0 **and**
[STATE_SA = INACTIVE **or**
STATE_SA = FINISHED **or**
STATE_SA = CANCELED **or**
STATE_SA = EXT_ADJ **or**
STATE_SA = EXT_ADJ_FIN **or**
STATE_SA = EOL_SA_FIN]

Deactivation: not activation

Service location: see list of “implemented diagnostic services”

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request Sid**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier – Sekundaerluftpumpe	M	CBh	IOLI_SLP
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_SLP
5d	InputOutputControlParameter - LV_ACT_SAP_EXT_ADJ	M	XXh	CS_TV_SLP
6d	InputOutputControlParameter - TIMEOUT_TIME_CB_KWP	M	XXh	CS_TO_SLP

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response Sid**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	CBh	IOLI_SLP
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.44 30_CC07 - Diagnosemodul-Tank Leckage Pumpe ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Diagnosemodul-Tank Leckage Pumpe	M	CCh	IOLI_DMTL_P
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_DMTL_P
5d	InputOutputControlParameter - LV_ACT_DMTL_PUMP_EXT_ADJ	M	XXh	CS_TV_DMTL_P
6d	InputOutputControlParameter - TIMEOUT_TIME_CC_KWP	M	XXh	CS_TO_DMTL_P

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	CCh	IOLI_DMTL_P
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.45 30_CD07 - Diagnosemodul-Tank Leckage Ventil ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Kl.87 Spannung / Versorgung	M	CDh	IOLI_DMTL_V
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_DMTL_V
5d	InputOutputControlParameter - LV_ACT_DMTLS_EXT_ADJ	M	XXh	CS_TV_DMTL_V
6d	InputOutputControlParameter - TIMEOUT_TIME_CD_KWP	M	XXh	CS_TO_DMTL_V

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	CDh	IOLI_DMTL_V
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.46 30_CE07 - Diagnosemodul-Tank Leckage Heizung ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Diagnosemodul-Tank Leckage Heizung	M	CEh	IOLI_DMTLH
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_DMTLH
5d	InputOutputControlParameter - LV_ACT_DMTLH_EXT_ADJ	M	XXh	CS_TV_DMTLH
6d	InputOutputControlParameter - TIMEOUT_TIME_CE_KWP	M	XXh	CS_TO_DMTLH

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	CEh	IOLI_DMTLH
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.47 30_CF07 - Tankentlueftungsventil ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier Tankentlueftungsventil ansteuern	M	CFh	IOLI_TEV
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_TEV
5d	InputOutputControlParameter - CPPWM_EXT_ADJ	M	XXh	CS_TV_TEV
6d	InputOutputControlParameter - TIMEOUT_TIME_CF_KWP	M	XXh	CS_TO_TEV

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	CFh	IOLI_TEV
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.48 30_D007 - Lambdasondenheizung vor Kat Bank1 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Lambdasondenheizung vor Kat Bank1	M	D0h	IOLI_LSH1
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_LSH1
5d	InputOutputControlParameter - LSHPWM_UP_EXT_ADJ[1]	M	XXh	CS_TV_LSH1
6d	InputOutputControlParameter - TIMEOUT_TIME_D0_KWP	M	XXh	CS_TO_LSH1

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	D0h	IOLI_LSH1
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.49 30_D107 - Lambdasondenheizung hinter Kat Bank1 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Lambdasondenheizung hinter Kat Bank1	M	D1h	IOLI_LSH2
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_LSH2
5d	InputOutputControlParameter - LSHPWM_DOWN_EXT_ADJ[1]	M	XXh	CS_TV_LSH2
6d	InputOutputControlParameter - TIMEOUT_TIME_D1_KWP	M	XXh	CS_TO_LSH2

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	D1h	IOLI_LSH2
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.50 30_D207 - Lambdasondenheizung vor Kat Bank2 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Lambdasondenheizung vor Kat Bank2	M	D2h	IOLI_LSH3
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_LSH3
5d	InputOutputControlParameter - LSHPWM_UP_EXT_ADJ[2]	M	XXh	CS_TV_LSH3
6d	InputOutputControlParameter - TIMEOUT_TIME_D2_KWP	M	XXh	CS_TO_LSH3

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	D2h	IOLI_LSH3
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.51 30_D307 - Lambdasondenheizung hinter Kat Bank2 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Lambdasondenheizung hinter Kat Bank2	M	D3h	IOLI_LSH4
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_LSH4
5d	InputOutputControlParameter - LSHPWM_DOWN_EXT_ADJ[2]	M	XXh	CS_TV_LSH4
6d	InputOutputControlParameter - TIMEOUT_TIME_D3_KWP	M	XXh	CS_TO_LSH4

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	D3h	IOLI_LSH4
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.52 30_D407 - MIL (Malfunction Indicator Lamp) ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - MIL (Malfunction Indicator Lamp)	M	D4h	IOLI_MIL
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_MIL
5d	InputOutputControlParameter - LV_ACT_MIL_EXT_ADJ	M	XXh	CS_TV_MIL
6d	InputOutputControlParameter - TIMEOUT_TIME_D4_KWP	M	XXh	CS_TO_MIL

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	D4h	IOLI_MIL
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.53 30_D507 - Fahrgeschwindigkeitsregler-Lampe ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier Fahrgeschwindigkeitsregler-Lampe	M	D5h	IOLI_FGRL
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_FGRL
5d	InputOutputControlParameter - LV_ACT_CRU_EXT_ADJ	M	XXh	CS_TV_FGRL
6d	InputOutputControlParameter - TIMEOUT_TIME_D5_KWP	M	XXh	CS_TO_FGRL

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	D5h	IOLI_FGRL
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.54 30_D607 - EML (Engine Malfunction Lamp) ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - EML (Engine Malfunction Lamp)	M	D6h	IOLI_EML
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_EML
5d	InputOutputControlParameter - LV_ACT_WAL_1_EXT_ADJ	M	XXh	CS_TV_EML
6d	InputOutputControlParameter - TIMEOUT_TIME_D6_KWP	M	XXh	CS_TO_EML

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	D6h	IOLI_EML
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.55 30_D807 - Elektrische Kraftstoffpumpe ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Elektrische Kraftstoffpumpe	M	D8h	IOLI_EKP1
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_EKP1
5d	InputOutputControlParameter - EFPPWM_EXT_ADJ	M	XXh	CS_TV_EKP1
6d	InputOutputControlParameter - TIMEOUT_TIME_D8_KWP	M	XXh	CS_TO_EKP1

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	D8h	IOLI_EKP1
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.56 30_DA07 - E-Luefter ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - E-Luefter	M	DAh	IOLI_ELUE
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_ELUE
5d	InputOutputControlParameter - ECFPWM_ECF_EXT_ADJ	M	XXh	CS_TV_ELUE
6d	InputOutputControlParameter - TIMEOUT_TIME_DA_KWP	M	XXh	CS_TO_ELUE

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	DAh	IOLI_ELUE
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.57 30_DC07 - VVT-Entlastungsrelais ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - VVT-Entlastungsrelais ansteuern	M	DCh	IOLI_VVTR
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_VVTR
5d	InputOutputControlParameter - LV_RLY_VVL_EXT_ADJ	M	XXh	CS_TV_VVTR
6d	InputOutputControlParameter - TIMEOUT_TIME_DC_KWP	M	XXh	CS_TO_VVTR

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	DCh	IOLI_VVTR
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.58 30_DD07 - VVT ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - VVT	M	DDh	IOLI_VVT
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_VVT
5d	inputOutputControlParameter - ANG_SP_EXT_ADJ_VVL	M	XXh	CS_PHY_VVT
6d	InputOutputControlParameter - TIMEOUT_TIME_DD_KWP	M	XXh	CS_TO_VVT

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	DDh	IOLI_VVT
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.59 30_E107 - Einspritzventil 1 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Einspritzventil 1	M	E1h	IOLI_EV1
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - IV_PER_EXT_ADJ[0]	M	XXh	CS_PD_EV1
5d	InputOutputControlParameter - IV_EXT_ADJ[0]	M	XXh	CS_TV_EV1
6d	InputOutputControlParameter - TIMEOUT_TIME_E1_KWP	M	XXh	CS_TO_EV1

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	E1h	IOLI_EV1
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.60 30_E207 - Einspritzventil 2 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Einspritzventil 2	M	E2h	IOLI_EV1
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - IV_PER_EXT_ADJ[4]	M	XXh	CS_PD_EV2
5d	InputOutputControlParameter - IV_EXT_ADJ[4]	M	XXh	CS_TV_EV2
6d	InputOutputControlParameter - TIMEOUT_TIME_E2_KWP	M	XXh	CS_TO_EV2

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	E2h	IOLI_EV2
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.61 30_E307 - Einspritzventil 3 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Einspritzventil 3	M	E3h	IOLI_EV1
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - IV_PER_EXT_ADJ[2]	M	XXh	CS_PD_EV3
5d	InputOutputControlParameter - IV_EXT_ADJ[2]	M	XXh	CS_TV_EV3
6d	InputOutputControlParameter - TIMEOUT_TIME_E3_KWP	M	XXh	CS_TO_EV3

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	E3h	IOLI_EV3
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.62 30_E407 - Einspritzventil 4 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Einspritzventil 4	M	E4h	IOLI_EV1
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - IV_PER_EXT_ADJ[5]	M	XXh	CS_PD_EV4
5d	InputOutputControlParameter - IV_EXT_ADJ[5]	M	XXh	CS_TV_EV4
6d	InputOutputControlParameter - TIMEOUT_TIME_E4_KWP	M	XXh	CS_TO_EV4

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	E4h	IOLI_EV4
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.63 30_E507 - Einspritzventil 5 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Einspritzventil 5	M	E5h	IOLI_EV1
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - IV_PER_EXT_ADJ[1]	M	XXh	CS_PD_EV5
5d	InputOutputControlParameter - IV_EXT_ADJ[1]	M	XXh	CS_TV_EV5
6d	InputOutputControlParameter - TIMEOUT_TIME_E5_KWP	M	XXh	CS_TO_EV5

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	E5h	IOLI_EV5
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.64 30_E607 - Einspritzventil 6 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Einspritzventil 6	M	E6h	IOLI_EV1
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - IV_PER_EXT_ADJ[3]	M	XXh	CS_PD_EV6
5d	InputOutputControlParameter - IV_EXT_ADJ[3]	M	XXh	CS_TV_EV6
6d	InputOutputControlParameter - TIMEOUT_TIME_E6_KWP	M	XXh	CS_TO_EV6

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	E6h	IOLI_EV6
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.65 30_ED07 - Vanos Einlass Ventil ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Vanos Einlass Ventil	M	EDh	IOLI_ENWS
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_ENWS
5d	inputOutputControlParameter - CAM_SP_IN_EXT_ADJ high byte	M	XXh	CS_PD_ENWS
6d	InputOutputControlParameter - CAM_SP_IN_EXT_ADJ low byte	M	XXh	CS_TV_ENWS
7d	InputOutputControlParameter - TIMEOUT_TIME_ED_KWP	M	XXh	CS_TO_ENWS

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**

2d	InputOutputLocalIdentifier	M	EDh	IOLI_ENWS
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.66 30_EE07 - Vanos Auslass Ventil ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**inputOutputControlByLocalIdentifier Request SId**	**M**	**30h**	**IOCBLID**
2d	InputOutputLocalIdentifier - Vanos Auslass Ventil	M	EEh	IOLI_ANWS
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	inputOutputControlParameter - not used	M	FFh	CS_PD_ANWS
5d	inputOutputControlParameter - CAM_SP_EX_EXT_ADJ high byte	M	XXh	CS_PD_ANWS
6d	InputOutputControlParameter - CAM_SP_EX_EXT_ADJ low byte	M	XXh	CS_TV_ANWS
7d	InputOutputControlParameter - TIMEOUT_TIME_EE_KWP	M	XXh	CS_TO_ANWS

Table InputOutputControlByLocalIdentifier request message

InputOutputControlByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**InputOutputControlByLocalIdentifier Positive Response SId**	**M**	**70h**	**IOCBLIDPR**
2d	InputOutputLocalIdentifier	M	EEh	IOLI_ANWS
3d	InputOutputControlParameter	M	07h	IOCP_STA

Table InputOutputControlByLocalIdentifier positive response message

7.22.67 30_EF07 – HPD5 Mengensteuerventil 1 ansteuern

InputOutputControlByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1d	inputOutputControlByLocalIdentifier Request SId	M	30h	IOCBLI
2d	inputOutputLocalID	M	EFh	IOLI_HDP5
3d	inputOutputControlParameter	M	07h	IOCP_STA
4d	InputOutputControlParameter –not used	M	FFh	CS_PD_HDP5
5d	InputOutputControlParameter – arqtmsv_w_rb[1] (high byte)	M	xxh	CS_TV_HDP5
6d	InputOutputControlParameter – arqtmsv_w_rb [1](low byte)	M	xxh	CS_TV_HDP5
7d	InputOutputControlParameter – TIMEOUT_TIME_EF_KWP	M	xxh	CS_TO_HDP5

Hint: activation with B_rqtmsv_rb[1]

InputOutputControlByLocalIdentifier Positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**InputOutputControlByLocalIdentifier Response SId**	**M**	**70h**	**IOCBLIDPR**
2	InputOutputLocalIdentifier	M	EFh	IOLI_HDP5
3	InputOutputControlParameter	M	07h	IOCP_STA

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_IDX_ACT_DS_CONF	70*6	0...FFFFH	0...65535	1	[-]
LDP_IDX_DS_ID_IDX_ACT_DS_CONF	70	0...FFH	0...255	1	[-]
LDP_IDX_ENVD_ID_IDX_ACT_DS_CONF	6	0...FFH	0...255	1	[-]
calibratable index of activation conditions and output variables for SID 30h services					
C_FUP_MAX_KWP	1	0...FFFFH	0...347776	5.3067216	[hPa]
maximum fuel pressure for external adjustment					
C_FUP_MIN_KWP	1	0...FFFFH	0...347776	5.3067216	[hPa]
minimum fuel pressure for external adjustment					
C_N_MAX_KWP	1	0...1FE0H	0...8160	1	[rpm]
max engine speed for accepting a KWP2000(*) service					
C_N_MAX_KWP_POIL	1	0...1FE0H	0...8160	1	[rpm]
maximum engine speed for external adjustment with poil dependence					
C_N_MIN_KWP	1	0...1FE0H	0...8160	1	[rpm]
minimum engine speed for external adjustment					
C_N_MIN_KWP_POIL	1	0...1FE0H	0...8160	1	[rpm]
minimum engine speed for external adjustment with poil dependence					
C_POIL_MAX_KWP	1	0...FFFFH	0...10868	0.165835	[hPa]
Maximum threshold for oil pressure KWP					
C_POIL_MIN_KWP	1	0...FFFFH	0...10868	0.165835	[hPa]
Maximum threshold for oil pressure KWP					
C_VS_MAX_KWP	1	0...FFH	0...255	1	[km/h]
maximum vehicle speed for accepting the service					
C_TCO_MIN_KWP	1	0...FEH	-48...142.5	0.75	[°C]
minimum TCO threshold for activation of external adjustment					
C_TCO_MAX_KWP	1	0...FEH	-48...142.5	0.75	[°C]
maximum TCO threshold for activation of external adjustment					
C_VB_MIN_KWP	1	0...FFH	0...25.89843	0.1015625	[V]
minimum battery voltage for external adjustment					

7.23 KWP2000 - SID 31h: startRoutineByLocalIdentifier Service

General information:

All implemented SID 31h services are described below.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_ecojob1	O/V	0...1H	0...1	1	[-]
Ecos Job1 aktiv					
B_ecojob2	O/V	0...1H	0...1	1	[-]
Ecos Job2 aktiv					
B_lvs_neustart	O/V/S	0...1H	0...1	1	[-]
Rücksetzen der LVS					
B_nglern	O/V	0...1H	0...1	1	[-]
Learning of neutral gear sensor					
Eco_max_i	O/V	0...FFH	0...0.31875	0.00125	[A]
Max. Ruhestromschwelle					
Eco_msb	O/V	0...FFH	0...12.75	0.05	[s]
Ecos Messtartbedingung					
Eco_mz	O/V	0...FFH	0...12.75	0.05	[s]
Dauer Mittelwertmessung					
Eco_timo	O/V	0...FFH	0...255	1	[s]
Ecos Messung Timeout					
INH_IV_KWP	O/V	0...3FH	0...63	1	[-]
Shut off request for cylinder x ------1=enabled / 0=disabledBit0=cylinder 0 (log) Bit1=cylinder 1 (log)Bit2=cylinder 2 (log)Bit3=cylinder 3 (log)Bit4=cylinder 4 (log)Bit5=cylinder 5 (log)					
LV_ACT_CRU_MEM_EXT_ADJ	O	0...1H	0...1	1	[-]
deleting value for STATE_CRU_OFF_IRR and STATE_CRU_OFF_REV =01h					
LV_AD_CLR_LONG_LAM_EXT_ADJ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
tester request for initializing Long Term Lambda Adaptation					
LV_ALTER_CTL_EXT_ADJ	O/V/S	0...1H	0...1	1	[-]
Set alternator control passiv					
LV_CAN_TEST_OPM_ENA_VVL	O/V	0...1H	0...1	1	[-]
Is the CAN operating mode for test purpose enabled ?					
LV_CTR_CYL_BAL_RST_EXT	O/V	0...1H	0...1	1	[-]
Reset request by service tool intervention - used by cylinder balancing OBDII diagnosis					
LV_CTR_TPS_JAM_DET_ACT_EXT_ADJ	O/V	0...1H	0...1	1	[-]
boolean for reset of jam counter when throttle changed in service					
LV_CYL_BAL_AD_DC_EXT_ADJ	O/V	0...1H	0...1	1	[-]
request to clear flags for cylinder balancing lambda adaptation at active driving cycle					
LV_CYL_BAL_AD_EOL_EXT_ADJ	O/V	0...1H	0...1	1	[-]
request to clear flags for cylinder balancing lambda adaptation at end of line					
LV_EFP_CTL_AD_CLR_EXT_REQ	O/V	0...1H	0...1	1	[-]
Logical variable indicating external request for deleting of EFP adaptation values					
LV_HPP_CTL_AD_CLR_EXT_REQ	O/V	0...1H	0...1	1	[-]
External adjustment to reset the adaptation values of high pressure fuel system					
LV_INH_LAM_KWP	O/V	0...1H	0...1	1	[-]
Switch off condition for lambda control via KWP2000 / 01= disabled 00=enabled					
LV_LAM_AD_INJ_CLR_AD_EXT	O/V	0...1H	0...1	1	[-]
Clear non-volatile data of lambda adapation via injection mode with service tester					
LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
external adjustment to reset the injector speciific lambda adaptation values					
LV_MFF_ADD_AD_ER_BAL_EXT_ADJ	O/V	0...1H	0...1	1	[-]
request to clear additive adaptation value for cylinder balancing via TI intervention					
LV_MFF_FAC_AD_ER_BAL_EXT_ADJ	O/V	0...1H	0...1	1	[-]
request to clear multiple adaptation value for cylinder balancing via TI intervention					
LV_POW_MNG_BAT_CHG	O/V	0...1H	0...1	1	[-]
condition for function requested of powermanagement register					
LV_POW_MNG_MES_MOD	O/V/S	0...1H	0...1	1	[-]
set default values PM for BN-SS - set default vaule=01h					
LV_TQ_LOSS_AD_CLR_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Request to clear the adaptation values for torque losses					

STATE_EGY_MIN_KWP	O/V/S	0H 1H 2H 3H	PASSIVE EGY_1 EGY_2 EGY_3	1	[-]
State of FeTraWe / 01= Fertigung 02= Transport 03= Werkstatt					
STATE_KWP_IMOB_STORE_K_EWS_ENA	O	0...FFH	0...255	1	[-]
State calculation and storage of K_EWS4					
LV_PWL_LOCK_CDN_CUS_INH	O/V	0...1H	0...1	1	[-]
PWL_LOCK_CDN for customer functions (IBS) disabled due to powerdown-service					

Input data:

C_FAC_EGY_PWM_MAN[NC_CYL_NR]			
C_N_MAX_KWP	CRK_CYL_LAM_DELTA_INI[NC_CBK_EX_NR]	CTR_ERR_OBD_DIAG_CYL_BAL_ER	CTR_ERR_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]
CTR_PHA_SHIFT_AD_TRIG[NC_CBK_EX_NR]	DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]	FAC_LAM_CYL_SEL_ADJ_CST[NC_CYL_NR]	FAC_LAM_CYL_SEL_ADJ_H_RNG[NC_CYL_NR]
FAC_LAM_CYL_SEL_ADJ_L_RNG[NC_CYL_NR]	FAC_TI_AD_ER_BAL[NC_CYL_NR]	GXMODN	LC_AD_CLR_CPS
LC_AD_CLR_EGCP	LC_AD_CLR_EGR	LC_AD_CLR_ENRD	LC_AD_CLR_ENSD
LC_AD_CLR_IDLE	LC_AD_CLR_IMM	LC_AD_CLR_INSY	LC_AD_CLR_LACO
LC_AD_CLR_LAM	LC_AD_CLR_N_SP_IS	LC_AD_CLR_RON	LC_AD_CLR_SA
LC_AD_CLR_THRO	LC_AD_CLR_TPS	LC_AD_CLR_VAR	LC_AD_CLR_VVL_STATE_NVMY
LV_ACT_N_SP_IS_BAS_EXT_ADJ	LV_CYL_BAL_AD_HOM_REQ_DC	LV_CYL_BAL_AD_WG_OPEN_REQ[NC_CBK_EX_NR]	LV_CYL_BAL_ER_AD_ADD_EOL
LV_CYL_BAL_ER_AD_FAC_EOL	LV_CYL_BAL_LAM_AD_DC	LV_CYL_BAL_LAM_AD_EOL	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL
LV_CYL_BAL_LAM_SEL_AD_HOT_DC	LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	LV_ES	LV_IGK
LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[NC_CBK_EX_NR]	LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[NC_CBK_EX_NR]	MFF_ADD_AD_ER_BAL[NC_CYL_NR]	MFF_ADD_CYL_LAM_COR[NC_CYL_NR]
N	NC_CBK_EX_NR	NC_CYL_NR	NC_IMOB_CALLID
NC_IMOB_GXMODN_LEN	NC_IMOB_MXBL	STATE_ENG	STATE_LAMB_CYL_SEL_CQ_SLOP[NC_CBK_EX_NR]
TRT	Tvneutrin	VS	LV_LAM_AD_INJ_EXT_ENA

Import action

ACTION_MFMA_SetEnableCondition(IN <PRM_FLAG_PAR_IN>)
void ACTION_NOXM_StartCatHeatDesu(void)
ACTION_INJR_SetFacEgyPwmAd (IN <Cyl>, IN <Fac>)
ACTION_NOXD_CleanAdDataNS(OUT <trig_CLR_AD>)
ACTION_NOXD_CleanMMVNSAdapt()
ACTION_NOXD_CleanNSAdapt(void)
ACTION_NOXD_CleanNSShiftAdapt(void)
ACTION_NOXM_CleanNTAdaptAgi(void)
ACTION_NOXM_CleanNTAdaptAgiMdl(void)
ACTION_NOXM_CleanNTAdaptSul(void)
eisy_clrad()

krann_read_ad()
void klann_read_ad(sint16 nkw_loc, uint16 rk_loc, sint16 tmot_loc)
void klann_clrad(void)
void kr_clrad(void)
void ls_clrad(void)
vbr_clrad()
void vns_clrad(void)
void eisyagr_read_ad(uint16 Agrpos)
void eisymr_read_addk(sint16 Nkw, uint16 Vse_spri, uint16 Vsa_spri,sint16 Wdk_ist)
void eisypsr_read_ad(sint16 Nkw, uint16, Vse_spri, uint16 Vsa_spri,sint16 Wdk_ist)
ACTION_EXTD_InitNTEgyStop()

7.23.1 Overview of supported subservices

	service	**Ser** **v** **i**	**MS**	**MS**	**MS**	**MS**	**MS**	**MS**
	ask checksums after reprogramming							
	erase Memory							
	erase History Memory							
	power down							
	start authentification							
	check authentification							
	check signature							

		report reprogramming status						
		energy spare mode						
		EWS4: Start calculation and storage						
		EWS4: Read out GXMODN data						
		EWS4: Read out length of GXMODN						
		EWS4: Transfer data of FSC to ECU						
		EWS4: Transfer length of ESC (FreiSchaltCode) to ECU						
		switch off Injection valves						
		Start Ruhestrom test						
		Start Nullgang-Lernen						

	Start Desulfatisierung Fahrbetrieb							
	delete selective adaption values							
	delete selective adaption values 2							
	switch off Lambda controller							
	Ansteuern Eisy-Adaptionswerte (ungedrosselt)							
	Ansteuern Eisy-Adaptionswerte (gedrosselt)							
	Ansteuern Eisy-Adaptionswerte (gedrosselt mit Abgasrueckführung)							
	Ansteuern Krann-Adaptionswerte							
	Ansteuern Klann-Adaptionswerte							
	Ansteuern AGR-Adaptionswerte							
	Calculate checksum for IncaPC							
	Programming of calibration data for IncaPC							
	cruise control delete switch off conditions							

	switch off Idle speed uplift							
	set default values PM for BN-SS							
	Intelligente Generatorregelung							
	Siemens interner Service							
	Siemens interner Service							
	Siemens interner Service							
	Message using CAN message							

7.23.2 Global negative Responses

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	StartRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	ResponseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

Remark:
The detailed message of the negative response code is described in chapter "Introduction".

7.23.3 31_01 ask checksums after reprogramming

General information:

This function is created to report the checksums after a reprogramming session in order to write the security keys
A negativ response is only possible by a wrong parameter.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier = ask for reprogramming ckecksum	M	#01h	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = ask for reprogramming ckecksum	M	#01h	RELI_
#3d	routineLocalIdentifier = state of checksum] OK Checksum failure within Boot-SW Checksum failure within ECU-SW Checksum failure within caklibration data]	M	#XXh= [#00h, #01h, #02h, #04h]	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.4 31_02 erase Memory (not for Tricore-ECU)

General information:

This function is created to erase the memory for not Tricore-ECUs before a update is programmed.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier = Erase Memory 31_02	M	#02h	RELI
#3d	routineLocalIdentifier = Memory adress high byte	M	#XXh	RELI_
#4d	routineLocalIdentifier = Memory adress middle byte	M	#XXh	RELI_
#5d	routineLocalIdentifier = Memory adress low byte	M	#XXh	RELI_
#6d	routineLocalIdentifier = Memory type 00=linear adress area 02= (external) Flash 06= (internal)Flash	M	#XXh	RELI_
#7d	routineLocalIdentifier = number of bytes - high byte	M	#XXh	RELI_
#8d	routineLocalIdentifier = number of bytes - middle byte	M	#XXh	RELI_
#9d	routineLocalIdentifier = number of bytes - low byte	M	#XXh	RELI_

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = erase memory	M	#02h	RELI_
#3d	routineLocalIdentifier = [deleting o.k. deleting not o.k.]	M	#XXh= [#01h, #02h,]	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.5 31_02 erase memory (for TRICORE ECUs)

General information:

This services is used to erase the FLASH memory of Tricore-ECUs for reprogramming.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request (3-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StartRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier = erase memory	M	#02h	RELI
#3d	RoutineLocalIdentifier = MemoryAddress (High Byte)	M	#XXh	RELI_
#4d	RoutineLocalIdentifier = MemoryAddress (Middle Byte)	M	#XXh	RELI_
#5d	RoutineLocalIdentifier = MemoryAddress (Low Byte)	M	#XXh	RELI_
(#6d)	RoutineLocalIdentifier = MemoryTypeIdentifier [02 = external FLASH 06 = internal FLASH]	U	#XXh= [#02h, #06h]	RELI_
#6d / (#7d)	RoutineLocalIdentifier = MemorySize (High Byte)	U	#XXh	RELI_
#7d / (#8d)	RoutineLocalIdentifier = MemorySize (Middle Byte)	U	#XXh	RELI_
#8d / (#9d)	RoutineLocalIdentifier = MemorySize (Low Byte)	U	#XXh	RELI_

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier Request (4-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StartRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier = erase memory	M	#02h	RELI
#3d	RoutineLocalIdentifier = MemoryAddress (High Byte)	M	#XXh	RELI_
#4d	RoutineLocalIdentifier = MemoryAddress (MiddleHigh Byte)	M	#XXh	RELI_
#5d	RoutineLocalIdentifier = MemoryAddress (MiddleLow Byte)	M	#XXh	RELI_
#6d	RoutineLocalIdentifier = MemoryAddress (Low Byte)	M	#XXh	RELI_
#7d	MemoryTypeIdentifier **(mandatory, but not evaluated)** [00h … FFh]	M	#XXh= [#XXh]	RELI_
#8d	RoutineLocalIdentifier = MemorySize (High Byte)	U	#XXh	RELI_
#9d	RoutineLocalIdentifier = MemorySize (MiddleHigh Byte)	U	#XXh	RELI_
#10d	RoutineLocalIdentifier = MemorySize (MiddleLow Byte)	U	#XXh	RELI_
#11d	RoutineLocalIdentifier = MemorySize (Low Byte)	U	#XXh	RELI_

Table StartRoutineByLocalIdentifier Request Message

Remark:

For 3-byte addressing **independent from delivery of a MemoryTypeIdentifier (byte 6) by the tester´s request, the ECU automatically adds an offset to the requested memory address.**
If no MemoryTypeIdentifier is delivered, the same offset as for MemoryTypeIdentifier 02h (external FLASH) or 06h (internal FLASH) is used. For compatibility reasons both MemoryTypeIdentifiers are supported.
This base memory offset must be subtracted from any memory address taken from the MAP- / A2L- file if 3-byte addressing is used!

For 4-byte addressing **a MemoryTypeIdentifier (byte 7) must be delivered by the tester´s request.**
Though it is not evaluated or used within the ECU for this absolute addressing mode, it is necessary to decide between 3- / 4-byte addressing mode.
In this 4-byte mode the absolute address delivered by the tester is used by the ECU without any memory offset.

For MSx80 (TriCore) the base memory offset for 3-byte addressing **mode of this service is independent from MemoryTypeIdentifier :**

MemoryTypeIdentifier:		Memory offset:
none	**(short)**	**8000 0000h**
02h	**(ext. FLASH)**	**8000 0000h**

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StartRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	RoutineLocalIdentifier = erase memory	M	#02h	RELI_
#3d	RoutineLocalIdentifier = [erasing o.k. erasing not o.k.]	M	#XXh= [#01h, #02h]	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.6 31_03 erase History Memory

General information:

This function is to erase the history memory (max. 10, see also “History memory” (chapter OBDII functions)), then the RAM-Memory is erased during the engine run.

If not, the content of the RAM is transferred to the non volatile memory at the end of the Power latch phase.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= [erase History Memory]	 M	#XXh= [#03h,]	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier erase history memory	M	#03h	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.7 31_05 power down

General information:

This function is created to switches the ECU to the power down mode..

Application conditions:

Initialisation: at reset or LV_IGK 0 --> 1
LV_PWL_LOCK_CDN_CUS_INH = 0

Activation: LV_IGK=0 (Kl.15 off)

Deactivation: LV_IGK=1 (Kl.15 on)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF      LV_IGK = 0          AND
        VS = 0              AND
        LV_ES = 01H (“engine stopped”)
THEN    IF    service 31_05 is received
        THEN
              LV_PWL_LOCK_CDN_CUS_INH = 1
              (reset PWL-lock-condition for customer)
              set Sleep-Indication-Bit and
              wait for  NM-Ackknowledge-Bit
              to shut down operating system
        ENDIF
ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier = power down mode	M	#05h	RELI
#3d	routineLocalIdentifier [$00h=switch all ecu´s into the power down mode $01h= switch all ecu´s except the power module into the power down mode]	M	#XXh [00h, 01h]	RELI_

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = power down mode	M	#05h	RELI_
#3d	routineLocalIdentifier [$00h=switch all ecu´s into the power down mode $01h= switch all ecu´s except the power module into the power down mode]	M	#XXh [00h, 01h]	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

Remark: regarding the BMW LH 8 385 774.4 chapter 10.1.1 table 10.1.1- if within 5s power down is not fullfilled, a negative Response 22h is sent to the tester.

7.23.8 31_07 start authentification

General information:

With this command the authentification procedure is started and the ECU sends the challenge key to the tester.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StartRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier = start authentification	M	#07h	RELI_SA

#3d	RoutineEntryOption (security level) = [security level 3 security level 4 security level 5]	M	#XXh [#03h, #04h, #05h]	REYO_01
#4d	RoutineEntryOption (user-ID byte_1)	M	#XXh	REYO_02
#5d	RoutineEntryOption (user-ID byte_2)	M	#XXh	REYO_03
#6d	RoutineEntryOption (user-ID byte_3)	M	#XXh	REYO_04
#7d	RoutineEntryOption (user-ID byte_4)	M	#XXh	REYO_05

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StartRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	RoutineLocalIdentifier = start authentification	M	#07h	RELI_SA
#3d	RoutineEntryStatus (challenge-key byte_1)	M	#XXh	REYS_01
#4d	RoutineEntryStatus (challenge-key byte_2)	M	#XXh	REYS_02
#5d	RoutineEntryStatus (challenge-key byte_3)	M	#XXh	REYS_03
#6d	RoutineEntryStatus (challenge-key byte_4)	M	#XXh	REYS_04
#7d	RoutineEntryStatus (challenge-key byte_5)	M	#XXh	REYS_05
#8d	RoutineEntryStatus (challenge-key byte_6)	M	#XXh	REYS_06
#9d	RoutineEntryStatus (challenge-key byte_7)	M	#XXh	REYS_07
#10d	RoutineEntryStatus (challenge-key byte_8)	M	#XXh	REYS_08

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.9 31_08 check authentification

General information:

With this command the RSA 512 authentification key delivered by the tester is checked inside the ECU.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StartRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier = check authentification	M	#08h	RELI_CA
#3d	RoutineEntryOption = authentification key length (high byte)	M	#00h	REYO_01
#4d	RoutineEntryOption = authentification key length (mid_high byte)	M	#00h	REYO_02
#5d	RoutineEntryOption = authentification key length (mid_low byte)	M	#00h	REYO_03
#6d	RoutineEntryOption = authentification key length (low byte)	M	#10h	REYO_04
#7d	RoutineEntryOption = authentification key_1 (high byte)	M	#XXh	REYO_05
#8d	RoutineEntryOption = authentification key_1 (mid_high byte)	M	#XXh	REYO_06
#9d	RoutineEntryOption = authentification key_1 (mid_low byte)	M	#XXh	REYO_07
#10d	RoutineEntryOption = authentification key_1 (low byte)	M	#XXh	REYO_08
#11d	RoutineEntryOption = authentification key_2 (high byte)	M	#XXh	REYO_09
#12d	RoutineEntryOption = authentification key_2 (mid_high byte)	M	#XXh	REYO_10
#13d	RoutineEntryOption = authentification key_2 (mid_low byte)	M	#XXh	REYO_11
#14d	RoutineEntryOption = authentification key_2 (low byte)	M	#XXh	REYO_12
#xxd	…	M	#XXh	REYO_xx
#xxd	…	M	#XXh	REYO_xx
#xxd	…	M	#XXh	REYO_xx
#xxd	…	M	#XXh	REYO_xx
#67d	RoutineEntryOption = authentification key_16 (high byte)	M	#XXh	REYO_65
#68d	RoutineEntryOption = authentification key_16 (mid_high byte)	M	#XXh	REYO_66
#69d	RoutineEntryOption = authentification key_16 (mid_low byte)	M	#XXh	REYO_67
#70d	RoutineEntryOption = authentification key_16 (low byte)	M	#XXh	REYO_68

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StartRoutineByLocalIdentifier Positive Response Sid	M	#71h	STRBLIPR
#2d	RoutineLocalIdentifier = check authentification	M	#08h	RELI_CA
#3d	RoutineEntryStatus = [authentification denied authentification accepted]	M	#XXh [#00h, #01h]	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.10 31_09 check signature

General information:

With this command the signature check for ECU-SW or calibration data is started.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on) AND
ECU running in Boot-SW

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StartRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier = check signature	M	#09h	RELI_CS
#3d	RoutineEntryOption = [check signature for ECU-SW check signature for calibration data]	M	#XXh [#02h, #04h]	REYO_01

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StartRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = check signature	M	#09h	RELI_CS
#3d	routineEntryStatus = [signature wrong signature correct]	M	#XXh [#00h, #01h]	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.11 31_0A report reprogramming status

General information:

This funktion check the reprogramming status of the ecu.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= [report reprogramming status]	M	#XXh= [#0Ah,]	RELI_CPSPROGS

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = PSL	M	#0Ah	
#3d	RoutineEntryStatus = reprogramming status: HW reference not ok = 09h Program not complete = 0Ch Program reference not ok = 0Ah Signature pattern PAF not ok = 05h HW reference does not fit to program reference = 0Bh No calibration data available at all = 00h Calibration data not complete = 0Fh Calibration data reference not ok = 0Dh Signature pattern DAF not ok = 06h Program reference does not fit to calibration data ref = 0Eh ECU is correct programed = 01h	M	#XXh	CS_PS

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.12 31_0C energy spare mode

FUNCTION DESCRIPTION:

This module handles the activation of the energy spare mode "FeTraWe". The mode is controlled by external tester and combined with an engine speed limit. The mode is regarded in dynamic error management.

With the received command an engine speed setpoint can be activated. The general function is described in the chapter “engine speed limit coordination”. The set energy spare mode includes either the activated "Fertigungs- / Transport- or Werkstattmode".

The mode is kept in ECU after disconnection of the tester. The same is applied in case of lost power supply.

Application conditions:

Initialisation: at reset and LV_IGK = 1: read non-volatile memory

Activation: LV_IGK=1 (Kl.15 on) **and** LC_EGY_MIN_KWP = 1

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF          activation identifier of service $31_0C_01 is received ***FeTraWe***
THEN        STATE_EGY_MIN_KWP = 1h
            send positive response

ELSEIF      activation identifier of service $31_0C_02 is received ***FeTraWe***
THEN        STATE_EGY_MIN_KWP = 2h
            send positive response

ELSEIF      activation identifier of service $31_0C_04 is received ***FeTraWe***
THEN        STATE_EGY_MIN_KWP = 3h
            send positive response

ELSEIF      deactivation identifier of service $31_0C_00 is received
THEN        STATE_EGY_MIN_KWP = 0h
            send positive response

       Save STATE_EGY_MIN_KWP     in non-volatile memory

ELSE   remains unchanged send negative response

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	**routineLocalIdentifier = energie spare mode**	**M**	**#0Ch**	

#3d	routineLocalIdentifier = set STATE_EGY_MIN_KWP [passive Fertigungsmodus Transportmodus Werkstattmodus]	M	#XXh [#00h #01h #02h #04h]	RELI_

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	**startRoutineByLocalIdentifier = enery spare mode**	**M**	**#0Ch**	
#3d	Routine entry status = FeTraWe [not used Fertigungsmodus Transportmodus Werkstattmodus]	M	#XXh [#00h #01h #02h #04h]	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.13 31_1F_E3 GetKeyFactor

General information:

IMMO: Read out GXMODN data

A negativ response is only possible by a wrong parameter.

Application conditions:

Initialisation: -

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

Plausibility check of CALL_ID

```
If(1) CALL_ID (high byte) << 8 | CALL_ID (low byte) != NC_IMOB_CALLID or
      Telegramm number == 0                    (Data with package 0 not allowed)
Then(1) Global Negative Responses (0x12) (sub-function not supported)
Endif(1)
```

Take GXMODN to pieces for KPW transfer

```
For(1) (i = 0… (NC_IMOB_MXBL-1))
If(1) ((TELEGRAM-NR  -1)* NC_IMOB_MXBL) + i < NC_IMOB_GXMODN_LEN
Then(1) #5d+id = GXMODN[], byte read out, LSB first
Endif(1)
EndFor(1)
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier = immo	M	#1Fh	RELI
#3d	routineLocalIdentifier = ask for GXMODN	M	#E3h	RELI
#4d	routineLocalIdentifier = Telegramm number] Telegramm 1 Telegramm xx]	M	#XXh [#01h, #xxh,]	RELI
#5d	routineLocalIdentifier = CALL_ID (high byte)	M	#XXh	RELI
#6d	routineLocalIdentifier = CALL_ID (low byte)	M	#XXh	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = ask for immob	M	#1Fh	RELI_
#3d	routineLocalIdentifier = ask for data of GXMODN	M	#E3h	RELI_
#4d	routineLocalIdentifier = Telegramm number] Telegramm 1 Telegramm XX]	M	#XXh [#01h, #XXh,]	RELI_
#5d-Xd	routineLocalIdentifier = GXMODN[XX] (see formulasection)	M	#XXh	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.14 31_1F_E4 GetKeyFactorLength

General information:

IMMO: Read out length of GXMODN

A negativ response is only possible by a wrong parameter.

Application conditions:

Initialisation: -

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier = immo	M	#1Fh	RELI
#3d	routineLocalIdentifier = ask for key factor length	M	#E4h	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = ask for immob	M	#1Fh	RELI_
#3d	routineLocalIdentifier = ask for length of GXMODN	M	#E4h	RELI_
#4d	routineLocalIdentifier = result] GXMODN checksum failed Checksum ok]	M	#XXh [#CDh, #00h,]	RELI_
#5d	routineLocalIdentifier = NC_IMOB_GXMODN_LEN high_byte	M	#XXh	RELI_
#6d	routineLocalIdentifier = NC_IMOB_GXMODN_LEN low_byte	M	#XXh	RELI_
#7d	routineLocalIdentifier = 0 (max. Block length high_byte)	M	#00h	RELI_
#8d	routineLocalIdentifier = NC_IMOB_MXBL	M	#XXh	RELI_
#9d	routineLocalIdentifier = NC_IMOB_CALLID high_byte	M	#XXh	RELI_
#10d	routineLocalIdentifier = NC_IMOB_CALLID low_byte	M	#XXh	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.15 31_1F_EE swt_checkFSC

General information:

A negativ response is only possible by a wrong parameter.

Application conditions:

Initialisation: -

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier = immo	M	#1Fh	RELI
#3d	routineLocalIdentifier = ask for check FSC	M	#EEh	RELI
#4d	routineLocalIdentifier = CALLID high_word_high_byte (SWT)	M	#XXh	RELI
#5d	routineLocalIdentifier = CALLID high_word_low_byte (SWT)	M	#XXh	RELI
#6d	routineLocalIdentifier = CALLID low_word_high_byte (SWT)	M	#XXh	RELI
#7d	routineLocalIdentifier = CALLID low_word_low_byte (SWT)	M	#XXh	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = ask for immob	M	#1Fh	RELI_
#3d	routineLocalIdentifier = ask for length of GXMODN	M	#EEh	RELI_
#4d	routineLocalIdentifier = error info	M	#XXh	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

Init-Parameter for swt_storeFSC (Software from customer)

SWT->noOfBytes = 4 (length of data)

SWT ->maxSizeOfReply = NC_IMOB_MXBL-1

SWT ->pData = (adresse kwp_buffer_ptr_rcv + 2)

SWT ->swID.swid_wb.appNr = SWID(highword,lowbyte)

SWT ->swID.swid_wb.upgrIdxDA = SWID(lowword,highbyte)

```
If(1) STATE_KWP_IMOB_STORE_K_EWS_ENA = 1
Then(1)   If(2) (STATE_IMOB_2_ERR & MASK_ERR_SYM_1) = 0 (no nvm read error)
          Then(2) If(3) Return of swt_checkFSC function call = positiv
                  Then(3)   If(4) FSC-Signature calculation and validation = ok
                            Then(4) #4d = 0x0    (data are valid)
                                    STATE_KWP_IMOB_STORE_K_EWS_ENA = 2
                                    NVM Storage of FSC
                            Else(4) #4d = 0xF9   (Signature failed)
                            Endif(4)
                  Else (3) #4d = swt_checkFSC_error_code
                  Endif(3)
          Else(2) #4d = 0xDF (nvm read error)
          Endif(2)
Else(1) Global Negative Responses (0x12) (sub-function not supported)
Endif(1)
```

7.23.16 31_1F_F1 swt_storeFSC

General information:

IMMO: Transfer data of FSC to ECU

A negativ response is only possible by a wrong parameter.

Application conditions:

Initialisation:	-
Activation:	Kl.15 on
Deactivation:	Kl.15 off Diagnostic timeout
Service location:	see list of “implemented diagnostic services”

Init-Parameter for swt_storeFSC (Software from customer)

SWT ->segNo = Telegramm number *(Number of data-block)*
SWT ->noOfBytes = kwp_ds_len – 3 *(length of data)*
SWT ->maxSizeOfReply = NC_IMOB_MXBL-1 *(maximal block length)*
SWT ->pData = (adresse kwp_buffer_ptr_rcv) +2 *(buffer share ASW-SWT)*
SWT ->swID.swid_wb.appNr = SWID(highword,lowbyte) *(SWID)*
SWT ->swID.swid_wb.upgrIdxDA = SWID(lowword,highbyte)

```
If(1)      (routineLocalIdentifier result) = ok
Then(1)    STATE_KWP_IMOB_STORE_K_EWS_ENA = 1
Else(1)    STATE_KWP_IMOB_STORE_K_EWS_ENA = 0
Endif(1)
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier = immo	M	#1Fh	RELI
#3d	routineLocalIdentifier = ask for store FSC to ECU	M	#F1h	RELI
#4d	routineLocalIdentifier = Telegramm number] Telegramm 1 Telegramm xx]	M	#XXh [#01h, #xxh,]	RELI
#5d	routineLocalIdentifier = CALLID high_byte (SWT)	M	#XXh	RELI
#6d	routineLocalIdentifier = CALLID low_byte (SWT)	M	#XXh	RELI
#7d-Xd	routineLocalIdentifier = FSC[X]	M	#XXh	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = ask for immob	M	#1Fh	RELI_
#3d	routineLocalIdentifier = ask for length of GXMODN	M	#F1h	RELI_
#4d	routineLocalIdentifier = result] Failure SWT ok]	M	#XXh [#XXh, #00h,]	RELI_
#5d	routineLocalIdentifier = Telegramm number] Telegramm 1 Telegramm xx]	M	#XXh [#01h, #xxh,]	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.17 31_1F_F2 swt_setLengthOfFSC

General information:

IMMO: Transfer length of FSC (FreiSchaltCode) to ECU

A negativ response is only possible by a wrong parameter.

Application conditions:

Initialisation: -

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of "implemented diagnostic services"

Init-Parameter for swt_setLengthOfFSC (Software from customer)

1)

SWT->noOfBytes = 6 (length of data)

SWT ->maxSizeOfReply = NC_IMOB_MXBL-1

SWT ->pData = (adresse kwp_buffer_ptr_rcv + 2)

SWT ->swID.swid_wb.appNr = SWID(highword,lowbyte)

SWT ->swID.swid_wb.upgrIdxDA = SWID(lowword,highbyte)

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier = immo	M	#1Fh	RELI
#3d	routineLocalIdentifier = ask for FSC length	M	#F2h	RELI
#4d	routineLocalIdentifier = SWID(highword, highbyte) 1)	M	#XXh	RELI
#5d	routineLocalIdentifier = SWID(highword, lowbyte) 1)	M	#XXh	RELI
#6d	routineLocalIdentifier = SWID(lowword, highbyte) 1)	M	#XXh	RELI
#7d	routineLocalIdentifier = SWID(lowword, lowbyte) 1)	M	#XXh	RELI
#8d	routineLocalIdentifier FSC length (high_byte) 1)	M	#XXh	RELI
#8d	routineLocalIdentifier FSC length (low_byte) 1)	M	#XXh	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = ask for immob	M	#1Fh	RELI_
#3d	routineLocalIdentifier = ask for length of FSC	M	#F2h	RELI_
#4d	routineLocalIdentifier = result] Failure SWT ok]	M	#XXh [#XXh, #00h,]	RELI_

#5d	routineLocalIdentifier = 0 (max. Block length high_byte)	M	#00h	RELI_
#6d	routineLocalIdentifier = max. Block length low_byte (SWT)	M	#XXh	RELI_
#7d	routineLocalIdentifier = CALLID high_byte (SWT)	M	#XXh	RELI_
#8d	routineLocalIdentifier = CALLID low_byte (SWT)	M	#XXh	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.18 31_25 switch off Injection valves

General information:

With this command the Injection valves can be switched off. The general function is described in the chapter “Injection - Cylinder shut off”.

Application conditions:

Initialisation: at reset with 00h

Activation:

LV_IGK=1 (Kl.15 on) and
STATE_ENG=02h or 03h (IS or PL) and
N<C_N_MAX_KWP

Deactivation:

LV_IGK=0 (Kl.15 off) or
STATE_ENG<>02h or 03h (IS or PL) or
N>=C_N_MAX_KWP or
Diagnostic timeout
set INH_IV_KWP=00h

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF      activation conditions are fullfilled        and
        service 31_25 is received

THEN    INH_IV_KWP =  sent to the module “Cylinder shut off “
        send positive response

ELSE    INH_IV_KWP=00h
        send negative response

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= shut off request for cylinder x	M	#25h	RELI_
#3d	routineEntryOption = active	M	#01h	REYO_
#4d	routineEntryOption = INH_IV_KWP	M	#XXh	REYO_

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = shut off request for cylinder x	M	#25h	RELI_

#3d	routineEntryStatus = active	M	#01h	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.19 31_2B Ruhestrom steuern

General information:

With this command "Ansteuern Ruhestromprüfung mit IBS" is started.
Hint: Reset of B_ecojob1/2 is normaly done in BMW-module.

Application conditions:

***Initialisation*:** at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or** Diagnostic timeout
set: B_ecojob1=0, B_ecojob2 = 0

Service location: see list of “implemented diagnostic services”

Formula section:

```
If          service 31_2B is received
Then        Eco_.. = received values
            B_ecojob1 = 1
            send positive response
 Else
            send negative response
Endif
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= Ruhestrom_steuern	M	#2Bh	RELI_RUHESTROM
#3d	routineLocalIdentifier = set Eco_max_i	M	#XXh	REYO_I_MAX
#4d	routineLocalIdentifier = set Eco_msb	M	#XXh	REYO_MSB
#5d	routineLocalIdentifier = set Eco_mz	M	#XXh	REYO_MZ
#6d	routineLocalIdentifier = set Eco_timo	M	#XXh	REYO_TO

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = Ruhestrom steuern x	M	#2Bh	RELI_RUHE STROM

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.20 31_2E Starten Nullgang-Lernen

General information:

With this command "Nullgang-Lernen" is started.

Application conditions:

***Initialisation**:* at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or** Diagnostic timeout
set: B_nglern=0

Service location: see list of “implemented diagnostic services”

Formula section:

```
If      service 31_2E is received
Then    B_nglern = 1 for  2 sec
        Tvneutrin = 0.0
        send positive response
Endif
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= Ruhestrom_steuern	M	#2Eh	RELI_NGL

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = Ruhestrom steuern x	M	#2Eh	RELI_NGL

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

7.23.21 31_2F Starten Desulfatisierung Fahrbetrieb

General information:

With this command "Desulfatisierung Fahrbetrieb" is started.

Application conditions:

***Initialisation*:** at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or** Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
If      service 31_2F is received
Then    call ACTION_NOXM_StartCatHeatDesu()
        send positive response
Endif
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= Start Desulfatisierung Fahrbetrieb	M	#2Eh	RELI_DESFAHR

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = Start Desulfatisierung Fahrbetrieb	M	#2Eh	RELI_DESFAHR

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.22 31_30 delete selective adaption values

General information:

With this comand the adaption values described in the chapter “basic sw general operation - initialize non volatile datas” are deleted.

All adaption values are initialized with their default values (compare detailed specifications), single adaption values according to the bit position.

If bit #4 within byte 4d ist masked (set to 1) by the tester, bit LV_POW_MNG_BAT_CHG is set to 1; after 5s LV_POW_MNG_BAT_CHG is automatically reset by the ECU to 0 again.

Within the telegram some macros are called (named with small letter below). This macros call functions over BMW- layer (see also layer- specification).

Application conditions:

Initialisation:	at reset init with 0	
Activation:	LV_IGK = 1 (Kl.15 on)	**and**
	LV_ES = 1	
Deactivation:	LV_IGK= 0 (Kl.15 off)	**and**
	LV_ES = 0	
	Diagnostic timeout	

Service location: see list of “implemented diagnostic services”

Formula section:

setting of LV_POW_MNG_BAT_CHG:

```
IF      bit position #4 in byte #4d set to 1 within tester´s request
THEN    set LV_POW_MNG_BAT_CHG to 1 and start timer
ELSE    do nothing
ENDIF

IF      LV_POW_MNG_BAT_CHG is set to 1 and timer >= 5s
THEN    reset LV_POW_MNG_BAT_CHG to 0
ELSE    do nothing
ENDIF
```

only **MSD80/MSD81:** setting/resetting LV_CYL_BAL_.../ LV_MFF_._/ LV_HPP_CTL_AD_CLR_EXT_REQ:

```
IF      bit position is set to 1 with tester´s request
THEN    set request bit
ELSE    do nothing
```

IF request bit is 1 and init of corresponding values is done

THEN reset request bit 100..200ms later

ELSE do nothing

request bit	corresponding values	init value:
LV_CRK_CYL_LAM_DELTA_RST_LS_EXT[NC_CBK_EX_NR]		
	DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]	0.0
	CRK_CYL_LAM_DELTA_INI[NC_CBK_EX_NR]0.0	
	CTR_PHA_SHIFT_AD_TRIG[NC_CBK_EX_NR]	0
LV_CTR_CYL_BAL_RST_EXT		
	CTR_ERR_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]	0
	CTR_ERR_OBD_DIAG_CYL_BAL_ER	0
LV_CYL_BAL_AD_EOL_EXT_ADJ:	LV_CYL_BAL_LAM_AD_EOL	
	LV_CYL_BAL_ER_AD_ADD_EOL	0
	LV_CYL_BAL_ER_AD_FAC_EOL	0
	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL	**0**
	LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	0
LV_CYL_BAL_AD_DC_EXT_ADJ:	LV_CYL_BAL_LAM_AD_DC,	0
	LV_CYL_BAL_AD_HOM_REQ_DC	0
	LV_CYL_BAL_LAM_SEL_AD_HOT_DC	0
	LV_CYL_BAL_AD_WG_OPEN_REQ[NC_CBK_EX_NR]	0
LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[NC_CBK_EX_NR]		
	FAC_LAM_CYL_SEL_ADJ_L_RNG[NC_CYL_NR]	0.0
	FAC_LAM_CYL_SEL_ADJ_H_RNG[NC_CYL_NR]	0.0
	FAC_LAM_CYL_SEL_ADJ_CST[NC_CYL_NR]	0.0
	MFF_ADD_CYL_LAM_COR[NC_CYL_NR]	0.0
	STATE_LAMB_CYL_SEL_CQ_SLOP[NC_CBK_EX_NR]	0
	LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[NC_CBK_EX_NR]	0
	LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[NC_CBK_EX_NR]	0
LV_MFF_FAC_AD_ER_BAL_EXT_ADJ:	FAC_TI_AD_ER_BAL[NC_CYL_NR]	1.00
LV_MFF_ADD_AD_ER_BAL_EXT_ADJ:	MFF_ADD_AD_ER_BAL [NC_CYL_NR]	0.00
LV_HPP_CTL_AD_CLR_EXT_REQ:	all adaptation values in "High pressure pump control" (02904B01.xxx)	like init for NVRAM

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= erase adaptive values	M	#30h	RELI
#3d	RoutineEntryStatus#1	M	#XXh	REYO_01
#4d	RoutineEntryStatus#2	M	#XXh	REYO_02
#5d	RoutineEntryStatus#3	M	#XXh	REYO_03

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier	M	#30h	RELI_AD
#3d	RoutineEntryStatus#1	M	#XXh	REYO_01

#4d	RoutineEntryStatus#2	M	#XXh	REYO_02
#5d	RoutineEntryStatus#3	M	#XXh	REYO_03

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

Bit Position	***Name RoutineEntryStatus#1***	***Mnemonic***
#0	idle speed LC_AD_CLR_N_SP_IS and LC_AD_CLR_IDLE	REYO_01
#1	knock control kr_clrad()	REYO_01
#2	lambda LC_AD_CLR_EGCP and **(not MSD8x)** LC_AD_CLR_LAM	REYO_01
#3	canister purge LC_AD_CLR_CPS	REYO_01
#4	mass flows to intake manifold eisy_clrad() and LC_AD_CLR_INSY and LC_AD_CLR_IMM	REYO_01
#5	throttle position sensor (not MSD85) LC_AD_CLR_TPS (only MSD85) LC_AD_CLR_THRO	REYO_01
#6	LACO adaptation: klann_clrad() and **(not MSV80)** LC_AD_CLR_LACO:	REYO_01
#7	**MSD8x**: ACRC adaptation LC_AD_CLR_EGR	REYO_01
Bit Position	***Name RoutineEntryStatus#2***	***Mnemonic***
#0	load control ls_clrad()	REYO_01
#1	**MSD80/81:** new Nox-sensor ACTION_NOXD_CleanAdDataNS(OUT <trig_CLR_AD>) ACTION_NOXD_CleanNSAdapt() ACTION_NOXD_CleanNSShiftAdapt() ACTION_NOXD_CleanMMVNSAdapt()	REYO_02
#2	**not MSD8x:** secondary air LC_AD_CLR_SA	REYO_02
#3	**MSD8x**: VCV adaptation LV_HPP_CTL_AD_CLR_EXT_REQ	REYO_02
#4	battery change LV_POW_MNG_BAT_CHG	REYO_02
#5	RON-adaption LC_AD_CLR_RON	REYO_02
#6	adapted variants LC_AD_CLR_VAR	REYO_02
#7	**not MSD8x:** VVL-adaption LC_AD_CLR_VVL_STATE_NVMY	REYO_02
Bit Position	***Name RoutineEntryStatus#3***	***Mnemonic***
#0	IVVT LC_AD_CLR_ENSD and vns_clrad()	REYO_03
#1	missfire LC_AD_CLR_ENRD	REYO_03
#2	**only MSV80:** burn control vbr_clrad()	REYO_03
#3	**only MSD8X:** cyl. balancing for DC and EOL: LV_CYL_BAL_AD_EOL_EXT_ADJ and LV_CYL_BAL_AD_DC_EXT_ADJ	REYO_03
#4	**only MSD8x:** ER-adaptation (faktor / add) and cylinder individuel lambda adaption values LV_MFF_FAC_AD_ER_BAL_EXT_ADJ and LV_MFF_ADD_AD_ER_BAL_EXT_ADJ and LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[NC_CBK_EX_NR] and LV_CTR_CYL_BAL_RST_EXT	REYO_03
#5	**only MSD8x:** phase shift adaptation, bank 1 LV_CRK_CYL_LAM_DELTA_RST_LS_EXT[1]	REYO_03
#6	**only MSD8x:** phase shift adaptation, bank 2 LV_CRK_CYL_LAM_DELTA_RST_LS_EXT[2]	REYO_03
#7	**only MSD8x:** injector needle lift control ACTION_INJR_SetFacEgyPwmAd (IN <Cyl>, IN <Fac>) for all <Cyl> with <Fac> = C_FAC_EGY_PWM_MAN[Cyl]	REYO_03

Table StartRoutineByLocalIdentifier Bit meaning

Refer the spec.on “Non Volatile Data” latest spec. xx000501 for deleting the variants.

7.23.23 31_31 delete selective adaption values 2

General information:

With this comand the adaption values 2 in BMW-software are initialized with their default values.

Application conditions:

Initialisation: at reset or in case of deactivation init with 0

Activation: LV_IGK = 1 (Kl.15 on) **and**
LV_ES = 1

Deactivation: LV_IGK= 0 (Kl.15 off) **or**
LV_ES = 0
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF      bit position#0 in byte#3d is set to 1 with tester´s request
THEN    do nothing
ENDIF

IF      bit position#1 in byte#3d is set to 1 with tester´s request
THEN    LV_AD_CLR_LONG_LAM_EXT_ADJ[1] = 1 for 2sec.
ENDIF

IF      bit position#2 in byte#3d is set to 1 with tester´s request
THEN    LV_AD_CLR_LONG_LAM_EXT_ADJ[2] = 1 for 2sec.
ENDIF

IF      bit position#3 in byte#3d is set to 1 with tester´s request
THEN    LV_EFP_CTL_AD_CLR_EXT_REQ = 1 for 2sec.
ENDIF

IF      bit position#4 in byte#3d is set to 1 with tester´s request
THEN    LV_CTR_TPS_JAM_DET_ACT_EXT_ADJ = 1 for 2sec.
ENDIF

IF      bit position#5 in byte#3d is set to 1 with tester´s request
THEN    LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[1]= 1 for 2sec.
ENDIF

IF      bit position#6 in byte#3d is set to 1 with tester´s request
THEN     LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[2]= 1 for 2sec.
ENDIF


IF      bit position#7 in byte#3d is set to 1 with tester´s request
THEN    call ACTION_MFMA_SetEnableCondition(1),
```

```
                LV_LAM_AD_INJ_CLR_AD_EXT = 1 for 2sec.
ENDIF
Hint: The result of call ACTION_MFMA_SetEnableCondition(1), is visible with
LV_LAM_AD_INJ_EXT_ENA


IF          bit position#6 in byte#4d is set to 1 with tester´s request
THEN        LV_TQ_LOSS_AD_CLR_EXT_ADJ = 1 for 2sec
ENDIF

IF          bit position#7 in byte#4d is set to 1 with tester´s request

THEN        B_lvs_neustart = 1  /*  Hint: reset of B_lvs_neustart to 0 is done in BMW-software */

ENDIF

IF          bit position#5 in byte#4d is set to 1 with tester´s request
THEN        call ACTION_NOXM_CleanNTAdaptSul ()
            call ACTION_NOXM_CleanNTAdaptAgi()
            call ACTION_NOXM_CleanNTAdaptAgiMdl()

ENDIF


IF          bit position#4 in byte#4d is set to 1 with tester´s request
THEN        call ACTION_EXTD_InitNTEgyStop()

Hint:       After call of ACTION_EXTD_InitNTEgyStop() is

                        EGY_DEW_NT_STOP = C_EGY_DEW_NT_STOP

ENDIF

For         other bits with value "1" in request byte#3d..#5d
do          nothing
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= erase adaptive values 2	M	#31h	RELI
#3d	RoutineEntryStatus#1	M	#XXh	REYO_01
#4d	RoutineEntryStatus#2	M	#XXh	REYO_02
#5d	RoutineEntryStatus#3	M	#XXh	REYO_03

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier	M	#31h	RELI_AD
#3d	RoutineEntryStatus#1	M	#XXh	REYO_01
#4d	RoutineEntryStatus#2	M	#XXh	REYO_02
#5d	RoutineEntryStatus#3	M	#XXh	REYO_03

Table StartRoutineByLocalIdentifier Positive Response Message

Bit Position	*Name RoutineEntryStatus#1*	*Mnemonic*
#0	not used	REYO_01

#1	Long Term Lambda Adaptation: LV_AD_CLR_LONG_LAM_EXT_ADJ[1]	REYO_01
#2	Long Term Lambda Adaptation: LV_AD_CLR_LONG_LAM_EXT_ADJ[2]	REYO_01
#3	EKP- AdaptationLV_EFP_CTL_AD_CLR_EXT_REQ	REYO_01
#4	Adaption Jitterfunktion Drosselklappe LV_CTR_TPS_JAM_DET_ACT_EXT_ADJ	REYO_01
#5	**only MSD80/MSD81:** LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[1]	REYO_01
#6	**only MSD80/MSD81:** LV_LAM_CYL_SEL_ADJ_RST_IV_EXT[2]	REYO_01
#7	**only MSD8x:** Kleinstmengenadaption: LV_LAM_AD_INJ_CLR_AD_EXT and call ACTION_MFMA_SetEnableCondition(1)	REYO_01
Bit Position	***Name RoutineEntryStatus#2***	***Mnemonic***
#0	not used	REYO_01
#1	not used	REYO_02
#2	not used	REYO_02
#3	not used	REYO_02
#4	**only MSD80/81 (6cyl):** Adaption Taupunkt NOX-Sensor zurücksetzen ACTION_EXTD_InitNTEgyStop()	REYO_02
#5	**MSD80/MSD81**: new NOx-Catalyst ACTION_NOXM_CleanNTAdaptSul(void) ACTION_NOXM_CleanNTAdaptAgi(void) ACTION_NOXM_CleanNTAdaptAgiMdl(void)	REYO_02
#6	**only MSD80**: Verlustmomentadaption löschen: LV_TQ_LOSS_AD_CLR_EXT_ADJ	REYO_02
#7	**only MSD80**: Restart Laufruheverbesserung: B_lvs_neustart	REYO_02
Bit Position	***Name RoutineEntryStatus#3***	***Mnemonic***
#0	not used	REYO_03
#1	not used	REYO_03
#2	not used	REYO_03
#3	not used	REYO_03
#4	not used	REYO_03
#5	not used	REYO_03
#6	not used	REYO_03
#7	not used	REYO_03

Table StartRoutineByLocalIdentifier Bit meaning

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.24 31_D9 switch off Lambda controller (not MSD80/MSD81)

General information:

With this command the Lambda Controler (bench 1 and 2) can be switched off. The general function is described in the chapter “Injection - Lambda Control”. The command is available for binary and linear lambda sensor systems.

Application conditions:

***Initialisation**:* at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout
set LV_INH_LAM_KWP=00h

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF      activation conditions are fullfilled        and
        service 31_D9 is received

THEN    LV_INH_LAM_KWP =  sent to the module “Lambda Control “
        send positive response

ELSE    LV_INH_LAM_KWP remains unchanged
        send negative response

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= switch off Lambda controller	M	#D9h	RELI_
#3d	routineEntryOption = active	M	#01h	REYO_01
#4d	routineEntryOption = LV_INH_LAM_KWP	M	#XXh	

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = switch off Lambda controller	M	#D9h	RELI_
#3d	routineEntryStatus = active	M	#01h	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.25 31_D9 switch off Lambda controller (only MSD80/MSD81)

General information:

With this command the Lambda Controler (bench 1 and 2) can be switched off. The general function is described in the chapter “Injection - Lambda Control”. The command is available for binary and linear lambda sensor systems.

Application conditions:

***Initialisation*:** at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout
set LV_INH_LAM_KWP=00h

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF      service 31_D9 is received
THEN    LV_INH_LAM_KWP =  1
ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= switch off Lambda controller	M	#D9h	RELI_
#3d	routineEntryOption = active	M	#01h	REYO_01
#4d	routineEntryOption = LV_INH_LAM_KWP	M	#XXh	

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = switch off Lambda controller	M	#D9h	RELI_
#3d	routineEntryStatus = active	M	#01h	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.26 31_E0 Ansteuern Eisy- Adaptionswerte (ungedrosselt)

General information:

With this service a function in the BMW- object is called (over the layer). The parameters, given from the tester via the below listet telegram, were overgiven to the function.

Application conditions:

***Initialisation*:** at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF    service 31_E0 is received

THEN  call BMW- function with the following parameters:
      void eisymr_read_adhub (sint16 Nkw, uint16 Vse_spri, uint16 Vsa_spri,
      uint16 Hubev_ist)

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= switch off Lambda controller	M	#E0h	RELI_EISYUGD
#3d	RecordValue#1 = NKW high byte	M	#XXh	REYO_ NKW
#4d	RecordValue#2 = NKW low byte	M	#XXh	REYO_ NKW
#5d	RecordValue#3 = VSE_SPRI high byte	M	#XXh	REYO_ VSE_SPRI
#6d	RecordValue#4 = VSE_SPRI low byte	M	#XXh	REYO_ VSE_SPRI
#7d	RecordValue#5 = VSA_SPRI high byte	M	#XXh	REYO_ VSA_SPRI
#8d	RecordValue#6 = VSA_SPRI low byte	M	#XXh	REYO_ VSA_SPRI
#9d	RecordValue#7 = HUBEV_IST high byte	M	#XXh	REYO_ HUBEV_IST
#10d	RecordValue#8 = HUBEV_IST low byte	M	#XXh	REYO_ HUBEV_IST

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIDValues	M	#E0h	RELI_EISYUGD

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.27 31_E1 Ansteuern Eisy- Adaptionswerte (gedrosselt)

General information:

With this service a function in the BMW- object is called (over the layer). The parameters, given from the tester via the below listet telegram, were overgiven to the function.

Application conditions:

***Initialisation**:*	at reset with 00h
Activation:	LV_IGK=1 (Kl.15 on)
Deactivation:	LV_IGK=0 (Kl.15 off) Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF      service 31_E1 is received

THEN    call BMW- function with the following parameters:
        void eisymr_read_addk(sint16 Nkw, uint16 Vse_spri, uint16 Vsa_spri,
sint16 Wdk_ist)

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= switch off Lambda controller	M	#E1h	RELI_EISYGD
#3d	RecordValue#1 = NKW high byte	M	#XXh	REYO_ NKW
#4d	RecordValue#2 = NKW low byte	M	#XXh	REYO_ NKW
#5d	RecordValue#3 = VSE_SPRI high byte	M	#XXh	REYO_ VSE_SPRI
#6d	RecordValue#4 = VSE_SPRI low byte	M	#XXh	REYO_ VSE_SPRI
#7d	RecordValue#5 = VSA_SPRI high byte	M	#XXh	REYO_ VSA_SPRI
#8d	RecordValue#6 = VSA_SPRI low byte	M	#XXh	REYO_ VSA_SPRI
#9d	RecordValue#7 = WDK_IST high byte	M	#XXh	REYO_ WDK_IST
#10d	RecordValue#8 = WDK_IST low byte	M	#XXh	REYO_ WDK_IST

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIDValues	M	#E1h	RELI_EISYGD

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.28 31_E2 Ansteuern Eisy- Adaptionswerte (gedrosselt mit Abgasrueckführung)

General information:

With this service a function in the BMW- object is called (over the layer). The parameters, given from the tester via the below listet telegram, were overgiven to the function.

Application conditions:

***Initialisation**:* at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF     service 31_E2 is received

THEN   call BMW- function with the received parameters:
       void eisypsr_read_ad(sint16 Nkw, uint16, Vse_spri, uint16 Vsa_spri,
                            sint16 Wdk_ist)

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= switch off Lambda controller	M	#E2h	RELI_EISYGDAGR
#3d	RecordValue#1 = NKW high byte	M	#XXh	REYO_ NKW
#4d	RecordValue#2 = NKW low byte	M	#XXh	REYO_ NKW
#5d	RecordValue#3 = VSE_SPRI high byte	M	#XXh	REYO_ VSE_SPRI
#6d	RecordValue#4 = VSE_SPRI low byte	M	#XXh	REYO_ VSE_SPRI
#7d	RecordValue#5 = VSA_SPRI high byte	M	#XXh	REYO_ VSA_SPRI
#8d	RecordValue#6 = VSA_SPRI low byte	M	#XXh	REYO_ VSA_SPRI
#9d	RecordValue#7 = WDK_IST high byte	M	#XXh	REYO_ WDK_IST
#10d	RecordValue#7 = WDK_IST low byte	M	#XXh	REYO_ WDK_IST

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**

#2d	routineLocalIDValues	M	#E2h	RELI_EISYG DAGR

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.29 31_E3 Ansteuern Krann- Adaptionswerte

General information:

With this service a function in the BMW- object is called (over the layer). The parameters, given from the tester via the below listet telegram, were overgiven to the function.

Application conditions:

Initialisation: at reset with 00h

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

IF service 31_E3 is received

THEN call BMW- function with the following parameters:
void krann_read_ad (sint16 Nkw, sint16 Rk, sint16 Tans, sint16 Tmot, uint8 Ba)

ENDIF

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= switch off Lambda controller	M	#E3h	RELI_KRANN
#3d	RecordValue#1 = NKW high byte	M	#XXh	REYO_ NKW
#4d	RecordValue#2 = NKW low byte	M	#XXh	REYO_ NKW
#5d	RecordValue#3 = Rk high byte	M	#XXh	REYO_ RK
#6d	RecordValue#4 = Rk low byte	M	#XXh	REYO_ RK
#7d	RecordValue#5 = Tans high byte	M	#XXh	REYO_TANS
#8d	RecordValue#5 = Tans low byte	M	#XXh	REYO_TANS
#9d	RecordValue#5 = Tmot high byte	M	#XXh	REYO_TMOT
#10d	RecordValue#5 = Tmot high byte	M	#XXh	REYO_TMOT
#11d	RecordValue#5 = Ba	M	#XXh	REYO_BA

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIDValues	M	#E3h	RELI_KRANN

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.30 31_E4 Ansteuern Klann- Adaptionswerte

General information:

With this service a function in the BMW- object is called (over the layer). The parameters, given from the tester via the below listet telegram, were overgiven to the function.

Application conditions:

***Initialisation*:** --

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF    service 31_E4 is received

THEN  call BMW- function with the following parameters:
      void klann_read_ad (sint16 nkw_loc, uint16 rk_loc, sint16 tmot_loc )

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= switch off Lambda controller	M	#E4h	RELI_KLANN
#3d	RecordValue#1 = nkw_loc high byte	M	#XXh	REYO_ NKW_LOC
#4d	RecordValue#2 = nkw_loc low byte	M	#XXh	REYO_ NKW_LOC
#5d	RecordValue#3 = rk_loc high byte	M	#XXh	REYO_ RK_LOC
#6d	RecordValue#4 = rk_loc low byte	M	#XXh	REYO_ RK_LOC
#7d	RecordValue#5 = tmot_loc high byte	M	#XXh	REYO_TMOT_LOC
#8d	RecordValue#5 = tmot_loc low byte	M	#XXh	REYO_TMOT_LOC

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIDValues	M	#E4h	RELI_KLANN

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.31 31_E5 Ansteuern AGR- Adaptionswerte

General information:

With this service a function in the BMW- object is called (over the layer). The parameter, given from the tester via the below listet telegram, were overgiven to the function.

Application conditions:

***Initialisation**:*	--
Activation:	LV_IGK=1 (Kl.15 on)
Deactivation:	LV_IGK=0 (Kl.15 off) Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula section:

```
IF      service 31_E5 is received

THEN    call BMW- function with the following parameters:
        void eisyagr_read_ad(uint16 Agrpos)

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= switch off Lambda controller	M	#E5h	RELI_EISYAGR
#3d	RecordValue#1 = Agrpos (high byte)	M	#XXh	REYO_ AGRPOS
#4d	RecordValue#2 = Agrpos (low byte)	M	#XXh	REYO_ AGRPOS

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIDValues	M	#E4h	RELI_EISYAGR

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.32 31_F0 Calculate checksum for IncaPC

General information:

This function starts the checksum calculation of the calibration data especially for IncaPC.

Therefore IncaPC gives the starting and ending addresses for the checksum calculation and also the desired checksum (calculated by IncaPC before).

The starting and ending addresses must be within the area for calibration data. Also startind address must be smaller than the ending address.

The ECU calculates afterwads an “add” checksum of the given area.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier = start checksum calculation for IncaPC	M	#F0h	RELI
#3d	Starting address for checksum calculation - high byte	M	#XXh	RELI
#4d	Starting address for checksum calculation - middle byte	M	#XXh	RELI
#5d	Starting address for checksum calculation - low byte	M	#XXh	RELI
#6d	Ending address for checksum calculation - high byte	M	#XXh	RELI
#7d	Ending address for checksum calculation - middle byte	M	#XXh	RELI
#8d	Ending address for checksum calculation - low byte	M	#XXh	RELI
#9d	Reference checksum - high byte	M	#XXh	RELI
#10d	Reference checksum - low byte	M	#XXh	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	RoutineLocalIdentifier = start checksum calculation for IncaPC	M	#F0h	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.33 31_F1 Programming of calibration data for IncaPC

General information:

This function is requesting for programming the calibration data after the power latch especially for Inca-PC. (Inca-PC function: “copy working-page to reference-page”)

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier = start checksum calculation for IncaPC	M	#F1h	RELI
#3d	Selected RAM page	M	#01h	

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response**	**M**	**#71h**	**STRBLIPR**
#2d	RoutineLocalIdentifier = start checksum calculation for IncaPC	M	#F1h	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.34 31_F4 cruise control - delete switch off conditions

General information:

With this comand the values described in the chapter “cruise control” are deleted.

All values are initialized with their default values (compare detailed specifications).

The deleted values are saved after a powerlatch phase into the non volatile memory.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on (LV_IGK=1)

Deactivation: Kl.15 off (LV_IGK=0)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formular section:

service 31_F4 is received

```
IF          LV_ACT_CRU_MEM_EXT_ADJ =01h     (deleting of CRU-conditions)

    THEN    STATE_CRU_OFF_IRR  =00h         and
            STATE_CRU_OFF_REV =00h

ELSE        LV_ACT_CRU_MEM_EXT_ADJ =00h
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= erase adaptive values	M	#F4h	RELI
#3d	RecordValue # 1 = LV_ACT_CRU_MEM_EXT_ADJ	M	#XXh	RV_

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier	M	#30h	RELI_AD
#3d	RecordValue # 1 = LV_ACT_CRU_MEM_EXT_ADJ	M	#XXh	RV_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.35 31_F5 switch off Idle speed uplift

General information:

This function is for switch off the idle speed uplift via KWP2000(*).

Application conditions:

Initialisation: at reset with 0

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of "implemented diagnostic services"

Formula section:

If service 31_F5 is requested

Then LV_ACT_N_SP_IS_BAS_EXT_ADJ = 1

End If

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= [switch off Idle speed uplift]	 M	#XXh= [#F5h,]	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response	M	#71h	STRBLIPR
#2d	routineLocalIdentifier switch off Idle speed uplift	M	#F5h	RELI_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.36 31_F6 set default values PM for BN-SS

General information:

With this comand the values described in the chapter "Power Management" are set to default values.

All values are initialized with their default values (compare detailed specifications).

The value LV_POW_MNG_MES_MOD is saved after a powerlatch phase into the non volatile memory.

Application conditions:

Initialisation:	at reset with LV_POW_MNG_MES_MOD from NVMY
Activation:	LV_IGK=1
Deactivation:	LV_IGK=0 Diagnostic timeout

Service location: see list of "implemented diagnostic services"

Formular section:

```
IF      service 31_F6    is received
THEN    LV_POW_MNG_MES_MOD = 01h
ELSE    LV_POW_MNG_MES_MOD remains unchanged
ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier	M	#F6h	RELI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier	M	#F6h	RELI_AD

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.37 31_F7 Intelligente Generatorregelung

General information:

With this comand the intelligent alternator control can be switched off.

Application conditions:

Initialisation: from NVMY

Activation: LV_IGK=1

Deactivation: LV_IGK=0
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formular section:

```
IF      service 31_F7        is received
THEN    LV_ALTER_CTL_EXT_ADJ= 01h
ELSE    LV_ALTER_CTL_EXT_ADJ remains unchanged
ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier	M	#F7h	RELI_IGR_OFF

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier	M	#F7h	RELI_IGR_OFF

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.23.38 31_FA Siemens interner Service

undokumentiert

7.23.39 31_FB Siemens interner Service

undokumentiert

7.23.40 31_FC Siemens interner Service

undokumentiert

7.23.41 31_FE Message using CAN message

FUNCTION DESCRIPTION:

General information:

This service is used to enable the CAN communication to a external CAN device. The aim is to activate the CAN channel (for test purpose only) in order to influence internal values (e.g. setpoint). During ECU reset the flag LV_CAN_TEST_OPM_ENA_VVL is checked wether the CAN channel is enabled to be active (LC_STATE_TEST_OPM_CAN_ENA_VVL = 1) or passive (LC_STATE_TEST_OPM_CAN_ENA_VVL = 0). Nevertheless even if the switch is set to 'passive' at ECU reset, the CAN channel can be enabled by the KWP service during engine running.

Description:

This function activates or deactivates the receiving of the CAN setpoint message.

Application conditions:

Initialisation:

LV_CAN_TEST_OPM_ENA_VVL = LC_STATE_TEST_OPM_CAN_ENA_VVL

Recurrence: depending on the user diagnose protocol

Activation: via diagnostic command

Deactivation: -

Service location: see list of “implemented diagnostic services”

Formula section:

// The receiving of the CAN message will be activated, when the VVL_CAN_INI is set to 0x01 //and deactivated, when the value of the VVL_CAN_INI !=0x01

LV_CAN_TEST_OPM_ENA_VVL= (VVL_CAN_INI ==0x01)

Send positive response

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StartRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier= start/stop the receiving of the CAN setpoint message control	M	#FEh	RELI_SSCS
#3d	VVL_CAN_INI = start/stop the receiving of the CAN setpoint message control command	M	#xxh	VVL_CAN_INI

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StartRoutineByLocalIdentifier Positive Response SId**	**M**	**#71h**	**STRBLIPR**
#2d	RoutineLocalIdentifier = start/stop the receiving of the CAN setpoint message control	M	#FEh	RELI_SSCS
#3d	RoutineEntryStatus = LV_CAN_TEST_OPM_ENA_VVL	M	#XXh	LV_CAN_TEST_OPM_ENA_VVL

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
LC_STATE_TEST_OPM_CAN_ENA_VVL	1	0...1H	0...1	1	[-]
Switch to enable or disable the CAN channel (for test purpose) at ECU reset					
LC_EGY_MIN_KWP	1	0...1H	0...1	1	[-]
Switch to enable the configuration of FeTraWe-Mode in ECU / enable = 1					

7.24 SID 34h: RequestDownload service

Input data:

LV_IGK			

General information:

This function is used for reprogramming the ECU.
Flash reprogramming is done with the following procedure:
RequestDownload, TransferData, …, RequestTransferExit, RequestDownload, TansferData, …, RequestTransferExit.
This means at memory address gaps during the programming session the transfer data service is finished with RequestTransferExit service. Resumption is done with a new RequestDownload-Service.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see "list of implemented diagnostic services"

7.24.1 RequestDownload Request (3-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestDownload Request Service Id**	**M**	**#34h**	**RD**
#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (Middle Byte)	M	#XXh	MA_MB
#4d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
(#5d)	MemoryTypeIdentifier **(optional)** [external FLASH internal FLASH]	U	#XXh [#02h #06h]	MTI
#5d / (#6d)	DataFormatIdentifier [uncompresses]	M	#00h	DFI
#6d / (#7d)	UncompressedMemorySize [High Byte]	M	#00h...#FFh	UCMS
#7d / (#8d)	UncompressedMemorySize [Middle Byte]	M	#00h...#FFh	UCMS
#8d / (#9d)	UncompressedMemorySize [Low Byte]	M	#00h...#FFh	UCMS

Table RequestDownload Request Message

7.24.2 RequestDownload Request (4-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestDownload Request Service Id**	**M**	**#34h**	**RD**
#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (MiddleHigh Byte)	M	#XXh	MA_MHB
#4d	MemoryAddress (MiddleLow Byte)	M	#XXh	MA_MLB
#5d	MemoryAddress (Low Byte)	M	#XXh	MA_LB

#6d	MemoryTypeIdentifier **(mandatory, but not evaluated)** [00h … FFh]	M	#XXh [#XXh]	MTI
#7d	DataFormatIdentifier [uncompresses]	M	#00h	DFI
#8d	UncompressedMemorySize [High Byte]	M	#00h...#FFh	UCMS
#9d	UncompressedMemorySize [MiddleHigh Byte]	M	#00h...#FFh	UCMS
#10d	UncompressedMemorySize [MiddleLow Byte]	M	#00h...#FFh	UCMS
#11d	UncompressedMemorySize [Low Byte]	M	#00h...#FFh	UCMS

Table RequestDownload Request Message

Remark:

For 3-byte addressing **independent from delivery of a MemoryTypeIdentifier (byte 5) by the tester´s request, the ECU automatically adds an offset to the requested memory address.**
If no MemoryTypeIdentifier is delivered, the same offset as for MemoryTypeIdentifier 00h (linear addressing) is used. Currently a download to this memory area is not allowed, it will be supported later on for SV-Revival download within a restricted memory area and additional security check.
For compatibility reasons both MemoryTypeIdentifiers (02h ext. FLASH / 06h int. FLASH) are supported and add the same memory offset.
This base memory offset must be subtracted from any memory address taken from the MAP- / A2L- file if 3-byte addressing is used!

For 4-byte addressing **a MemoryTypeIdentifier (byte 6) must be delivered by the tester´s request.**
Though it is not evaluated or used within the ECU for this absolute addressing mode, it is necessary to decide between 3- / 4-byte addressing mode.
In this 4-byte mode the absolute address delivered by the tester is used by the ECU without any memory offset.

For MSx80 (TriCore) the base memory offsets for 3-byte addressing **mode of this service are:**

MemoryTypeIdentifier:		Memory offset:
none	**(linear)**	**C000 0000h supported for SV-Revival download later on**
02h	**(ext. FLASH)**	**8000 0000h**
06h	**(int. FLASH)**	**8000 0000h**

7.24.3 RequestDownload positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestDownload Positive Response Service Id**	**M**	**#74h**	**RDPR**
#2d	TransferResponseParameter=[maxNumberOfBlockLength]	M	#XXh	TREP_ MNROBL

Table RequestDownload Positive Response Message

7.24.4 RequestDownload negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	negativeResponse Service Id	M	#7Fh	NR
#2d	RequestDownload Request Service Id	M	#34h	RD
#3d	responseCode= [general response code, download not allowed, conditins for programming not fulfilled]	M	#XXh= [#XXh, #40h, #22h]	RC

Table RequestDownload Negative Response Message

7.25 SID 35h: RequestUpload service

Input data:

LV_IGK			

General information:

This function allows to upload data from the ECU.

Application conditions:

Initialisation:	at reset
Activation:	Kl.15 on
Deactivation:	Kl.15 off Diagnostic timeout
Service location:	see "list of implemented diagnostic services"

7.25.1 RequestUpload Request (3-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestUpload Request Service Id**	**M**	**#35h**	**RU**
#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (Middle Byte)	M	#XXh	MA_MB
#4d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
(#5d)	MemoryTypeIdentifier **(optional)** [linear type external FLASH external RAM internal FLASH user information field UIF]	U	#XXh [#00h #02h #05h #06h #07h]	MTI
#5d / (#6d)	DataFormatIdentifier (only uncompressed memory supported)	M	#00h	DFI
#6d / (#7d)	UncompressedMemorySize [High Byte]	M	#00h...#FFh	UCMS
#7d / (#8d)	UncompressedMemorySize [Middle Byte]	M	#00h...#FFh	UCMS
#8d / (#9d)	UncompressedMemorySize [Low Byte]	M	#00h...#FFh	UCMS

Table RequestUpload Request Message

7.25.2 RequestUpload Request (4-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestUpload Request Service Id**	**M**	**#35h**	**RU**
#2d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#3d	MemoryAddress (MiddleHigh Byte)	M	#XXh	MA_MHB
#4d	MemoryAddress (MiddleLow Byte)	M	#XXh	MA_MLB
#5d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
#6d	MemoryTypeIdentifier **(mandatory, but not evaluated)** [00h … FFh]	M	#XXh [#XXh]	MTI
#7d	DataFormatIdentifier (only uncompressed memory supported)	M	#00h	DFI

#8d	UncompressedMemorySize [High Byte]	M	#00h...#FFh	UCMS
#9d	UncompressedMemorySize [MiddleHigh Byte]	M	#00h...#FFh	UCMS
#10d	UncompressedMemorySize [MiddleLow Byte]	M	#00h...#FFh	UCMS
#11d	UncompressedMemorySize [Low Byte]	M	#00h...#FFh	UCMS

Table RequestUpload Request Message

Remark:

For 3-byte addressing **independent from delivery of a MemoryTypeIdentifier (byte 5) by the tester´s request, the ECU automatically adds an offset to the requested memory address.**
If no MemoryTypeIdentifier is delivered, the same offset as for MemoryTypeIdentifier 00h (linear addressing) is used.
This base memory offset must be subtracted from any memory address taken from the MAP- / A2L- file if 3-byte addressing is not sufficient for the address range!

For 4-byte addressing **a MemoryTypeIdentifier (byte 6) must be delivered by the tester´s request.**
Though it is not evaluated or used within the ECU for this absolute addressing mode, it is necessary to decide between 3- / 4-byte addressing mode.
In this 4-byte mode the absolute address delivered by the tester is used by the ECU without any memory offset.

For MSx80 (TriCore) the base memory offsets for 3-byte addressing **are:**

MemoryTypeIdentifier:		Memory offset:
none	**(linear)**	**C000 0000h**
00h	**(linear)**	**C000 0000h**
02h	**(ext. FLASH)**	**8000 0000h (the same as for int. FLASH)**
05h	**(ext. RAM)**	**D000 0000h**
06h	**(int. FLASH)**	**8000 0000h**
07h	**(UIF)**	**8000 xxxxh not fixed yet, address might still change !!!**

The supported address ranges are identical to the areas defined for the service ReadMemoryByAddress.

7.25.3 RequestUpload positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**requestUpload Positive Response Service Id**	**M**	**#75h**	**RUPR**
#2d	TransferResponseParameter=[maxNumberOfBlockLength] = 239d	M	#EFh	TREP_ MNROBL

Table RequestUpload Positive Response Message

7.25.4 RequestUpload negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	negativeResponse Service Id	M	#7Fh	NR
#2d	RequestUpload Request Service Id	M	#35h	RU
#3d	responseCode= [CanNotUpload-FromSpecificAdress]	 M 	#XXh= [#52h]	RC_

Table RequestUpload Negative Response Message

7.25.5 SID 36h: transferData Service

General information:

This funktion is for exchanging data after a request upload or download service.

Application conditions:

Initialisation:	at reset
Activation:	Kl.15 on
Deactivation:	Kl.15 off Diagnostic timeout
Service location:	Boot-/ ECU-SW

TransferData Request after a request upload service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	transferData Request Service Id	M	#36h	TD

Table TransferData Request Message after request upload

TransferData positive Response after a request upload service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	transferData Positive Response Service Id	M	#76h	TDPR
#2d	Data 1	M	#XXh	
#3d	Data 2	M	#XXh	
#nd	Data n	M	#XXh	

Table TransferData Positive Response Message after request upload

TransferData negative Response after a request upload service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	TransferData Request Service Id	M	#36h	TD
#3d	responseCode= [general response code all bytes are uploaded]	M	#XXh= [#XXh, #53h,]	RC

Table TransferData Negative Response Message after request upload

TransferData Request after a request download service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	transferData Request Service Id	M	#36h	TD
#2d	Data 1	M	#XXh	
#3d	Data 2	M	#XXh	
#nd	Data n	M	#XXh	

Table TransferData Request Message after request download

TransferData positive Response after a request download service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	transferData Positive Response Service Id	M	#76h	TDPR
#2d	number of transfer commands since 1. request high_Byte	M	#XXh	TDPR
#3d	number of transfer commands since 1. request low_Byte	M	#XXh	TDPR
#4d	status byte [reprogramming ok reprogramming not ok.]	M	#XXh [#01h, #02h]	TDPR

Table TransferData Positive Response Message after request download

TransferData negative Response after a request download service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	TransferData Request Service Id	M	#36h	TD
#3d	ResponseCode= [general response code conditions not correct]	M	#XXh= [#XXh, #22h,]	RC

Table TransferData Negative Response Message after request download

Time	*client (Tester)*	*server (ECU)*
P3	transferData.Request#1[...]	
P2		transferData.PositiveResponse#1[...]
P3	transferData.Request#2[...]	
P2		transferData.PositiveResponse#2[...]
.	.	.
.	.	.
.	.	.
P3	transferData.Request#n[...]	
P2		transferData.PositiveResponse#n[...]

Table transferData Message Flow

7.26 SID 37h: RequestTransferExit service

Input data:

LV_IGK			

General information:

This function is necessary to stop the RequestDownload / RequestUpload process.
Remark: Ending a reprogramming session. There are no new informations since the last transfer data available, therefore it is not necessary to transmit a verify byte.

Application conditions:

Initialisation:	at reset
Activation:	Kl.15 on
Deactivation:	Kl.15 off Diagnostic timeout
Service location:	see "list of implemented diagnostic services"

7.26.1 RequestTransferExit Request (short form)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestTransferExit**	**M**	**#37h**	**RTE**

Table RequestTransferExit Request Message

7.26.2 RequestTransferExit Request (3-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestTransferExit**	**M**	**#37h**	**RTE**
#2d	Download / Upload MemoryAddress (High Byte) **(not evaluated)**	U	#XXh	MA_HB
#3d	Download / Upload MemoryAddress (Middle Byte) **(not evaluated)**	U	#XXh	MA_MB
#4d	Download / Upload MemoryAddress (Low Byte) **(not evaluated)**	U	#XXh	MA_LB
(#5d)	Download / Upload MemoryAddress (Low Byte) **(not evaluated)**	U	#XXh	MTI
#5d / (#6d)	MemoryTypeIdentifier **(not evaluated)**	U	#00h	DFI
#6d / (#7d)	UncompressedMemorySize [High Byte] **(not evaluated)**	U	#00h...#FFh	UCMS
#7d / (#8d)	UncompressedMemorySize [Middle Byte] **(not evaluated)**	U	#00h...#FFh	UCMS
#8d / (#9d)	UncompressedMemorySize [Low Byte] **(not evaluated)**	U	#00h...#FFh	UCMS

Table RequestTransferExit Request Message

7.26.3 RequestTransferExit Request (4-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestTransferExit**	**M**	**#37h**	**RTE**
#2d	Download / Upload MemoryAddress (High Byte) **(not evaluated)**	U	#XXh	MA_HB
#3d	Download / Upload MemoryAddress (MiddleHigh Byte) **(not evaluated)**	U	#XXh	MA_MHB
#4d	Download / Upload MemoryAddress (Middlelow Byte) **(not evaluated)**	U	#XXh	MA_MLB
#5d	Download / Upload MemoryAddress (Low Byte) **(not evaluated)**	U	#XXh	MA_LB
#6d	MemoryTypeIdentifier **(mandatory, but not evaluated)**	U	#XXh	MTI
#7d	DataFormatIdentifier [uncompresses] **(not evaluated)**	U	#00h	DFI

#8d	UncompressedMemorySize [High Byte] **(not evaluated)**	U	#00h...#FFh	UCMS
#9d	UncompressedMemorySize [MiddleHigh Byte] **(not evaluated)**	U	#00h...#FFh	UCMS
#10d	UncompressedMemorySize [MiddleLow Byte] **(not evaluated)**	U	#00h...#FFh	UCMS
#11d	UncompressedMemorySize [Low Byte] **(not evaluated)**	U	#00h...#FFh	UCMS

Table RequestTransferExit Request Message

Remark:

No additional formal parameters are evaluated. The only criteria checked is the request length.
Accepted request lengths and telegram formats are:

1 byte:	**SID**
8 bytes:	**SID, Addr3, Addr2, Addr1, DFI, Size3, Size2, Size1**
9 bytes:	**SID, Addr3, Addr2, Addr1, MTI, DFI, Size3, Size2, Size1**
11 bytes	**SID, Addr4, Addr3, Addr2, Addr1, MTI, DFI, Size4, Size3, Size2, Size1**

This service is only accepted after all previously defined data transfer (by RequestDownload / RequestUpload) is done, i.e. no more data is left from TransferData service.

7.26.4 RequestTransferExit positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestTransferExit PositiveResponse**	**M**	**#77h**	**PR**

Table RequestTransferExit Positive Response Message

7.26.5 RequestTransferExit negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestTransferExit Service Id	M	#37h	RTE
#3d	ResponseCode=General response code Upload / Download not complete	M	#XXh #22h	RC

Table RequestTransferExit Negative Response Message

7.27 SID 3Bh: writeDataByLocalIdentifier Service

General information:

All implemented SID 30h services are described below.

Formula section:

7.27.1 Global negative Responses

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	writeDataByLocalIdentifier Request Service ID	M	#3Bh	WDBLI
#3d	responseCode= only general response code	M	#XXh	RC

Table writeDataByLocalIdentifier Negative Response Message

Remark:
The detailed message of the negative response code is described in chapter "Error-Codes".

7.27.2 3B_90 - Kurze Fahrgestellnummer schreiben - writeDataByLocalIdentifier Service

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VIN_SHO[7]	V/O/S	0...FFH	0...255	1	[-]
short vehicle identification number					

General information:

With this service the short vehicle identification code is written to E2PROM. Furthermore the identification number of the dataset (Cal-ID) is written to E2PROM. The receiving of this command is interpreted as successful coding of BMW variants.

Application conditions:

Initialisation:	at reset
Activation:	LV_IGK = 1 **and** LV_ES = 1
Deactivation:	LV_IGK = 0 **or** LV_ES = 0 Diagnostic timeout
Service location:	see list of “implemented diagnostic services”

writeDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByLocalID Request Service Id**	**M**	**#3Bh**	**WDBLI**

#2d	recordLocalIdentifier = Kurze Fahrgestellnummer	M	#90h	RLI_
#3d	RecordValue = VIN_SHO[0]	M	#XXh	RV_
#4d	RecordValue = VIN_SHO[1]	M	#XXh	RV_
#5d	RecordValue = VIN_SHO[2]	M	#XXh	RV_
#6d	RecordValue = VIN_SHO[3]	M	#XXh	RV_
#7d	RecordValue = VIN_SHO[4]	M	#XXh	RV_
#8d	RecordValue = VIN_SHO[5]	M	#XXh	RV_
#9d	RecordValue = VIN_SHO[6]	M	#XXh	RV_

Table writeDataByLocalIdentifier Request Message

writeDataByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByLocal ID Positive Response SId**	**M**	**#7Bh**	**WDBLIPR**
#2d	routineLocalIdentifier= Kurze Fahrgestellnummer	M	#90h	RLI_

Table writeDataByLocalIdentifier Positive Response Message

writeDataByLocalIdentifier negative Response

Global Negative Responses

7.27.3 3B_C1 - write kva-factor - writeDataByLocalIdentifier Service

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_FCO_KWP	V/O/S	80...7FH	-0.128...0.127	0.001	[-]
Correction factor for consumption					

General information:

With this service the additive part of the correction-factor is updated.
The ECU works after that with this new value.
The additive part of this correction-factor is between –0,128 and +0,127.

Application conditions:

Initialisation:	at reset
Activation:	Kl.15 on
Deactivation:	Kl.15 off Diagnostic timeout
Service location:	see list of “implemented diagnostic services”

writeDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByLocalID Request Service Id**	**M**	**#3Bh**	**WDBLI**
#2d	recordLocalIdentifier = KVA-factor	M	#C1h	RLI_KVA
#3d	RecordValue = FAC_FCO_KWP [remain the same negative displace max.12.8% positive displace max.12.7% (1h=0.001/bit)]	M	#XXh [#00h, #80h-FFh #01h-7Fh]	RV_KVA_KORR

Table writeDataByLocalIdentifier Request Message

writeDataByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByLocal ID Positive Response SId**	**M**	**#7Bh**	**WDBLIPR**
#2d	routineLocalIdentifier=KVA	M	#C1h	RLI_KVA

Table writeDataByLocalIdentifier Positive Response Message

writeDataByLocalIdentifier negative Response

Global Negative Responses

7.28 SID 2Ch: DynamicallyDefineLocalIdentifier service

Output data:

see introduction

Input data:

see introduction

General information:

This function is used to define / clear single or multiple data that can be read as one complete block by ReadDataByLocalId service 21_F0 after block definition.
Data definition can be made either directly by address or by common identifier RCI from DDLI table (see specification 17I00101, "Introduction") defined inside ECU-SW.

Application conditions:

Initialisation: at reset

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0
Diagnostic timeout

Service location: see "list of implemented diagnostic services"

7.28.1 DynamicallyDefineLocalIdentifier Request - DefineByCommonIdentifier - 2C_02

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**DynamicallyDefineLocalIdentifier Request Service Id**	**M**	**#2Ch**	**DDLI**
#2d	DynamicallyDefinedLocalIdentifier	M	#F0h	DDDLI
#3d	DefinitionMode=XX [DefineByCommonID]	M	#XXh= [#02h]	DNM DBLI
#4d	PositionInDynamicallyDefinedLocalIdentifier	M	#XXh	PIDDDLI
#5d	MemorySize	M	#XXh	MS
#6d	RecordCommonID (high byte) *(Index according to DDLI data list)*	M	#XXh	RCI_HB
#7d	RecordCommonID (low byte) *(Index according to DDLI data list)*	M	#XXh	RCI_LB
#8d	PositionInRecordCommonIdentifier	M	#01h	PIRCI
...	...	C	...	
#n-5d	DefinitionMode=XX [DefineByCommonID]	C	#XXh= [#02h]	DNM
#n-4d	PositionInDynamicallyDefinedLocalIdentifier	C	#XXh	PIDDDLI
#n-3d	MemorySize	C	#XXh	MS
#n-1d	RecordCommonID (high byte) *(Index according to DDLI data list)*	C	#XXh	RCI_HB
#n-1d	RecordCommonID (low byte) *(Index according to DDLI data list)*	C	#XXh	RCI_LB
#nd	PositionInRecordCommonIdentifier	C	#01h	PICLI

Table DynamicallyDefineLocalIdentifier Request Message - 2C_02

7.28.2 DynamicallyDefineLocalIdentifier Request - DefineByMemoryAddress - 2C_03

(3-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**DynamicallyDefineLocalIdentifier Request Service Id**	**M**	**#2Ch**	**DDLI**
#2d	DynamicallyDefinedLocalIdentifier	M	#F0h	DDDLI
#3d	DefinitionMode=XX [DefineByMemoryAddress]	M	#XXh= [#03h]	DNM DBMA
#4d	PositionInDynamicallyDefinedLocalIdentifier	M	#XXh	PIDDDLI
#5d	MemorySize	M	#XXh	MS
#6d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#7d	MemoryAddress (Middle Byte)	M	#XXh	MA_MB
#8d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
(#9d)	MemoryTypeIdentifier **(optional)** [linear type external FLASH external RAM internal FLASH user information field UIF]	U	#XXh [#00h #02h #05h #06h #07h]	MTI
...	...	C	...	
#n-6d	DefinitionMode=XX [DefineByMemoryAddress]	C	#XXh= [#03h]	DNM DBMA
#n-5d	PositionInDynamicallyDefinedLocalIdentifier	C	#XXh	PIDDDLI
#n-4d	MemorySize	C	#XXh	MS
#n-3d	MemoryAddress (High Byte)	C	#XXh	MA_HB
#n-2d	MemoryAddress (Middle Byte)	C	#XXh	MA_MB
#n-1d	MemoryAddress (Low Byte)	C	#XXh	MA_LB
(#nd)	MemoryTypeIdentifier **(optional)** [linear type external FLASH external RAM internal FLASH user information field UIF]	U	#XXh [#00h #02h #05h #06h #07h]	MTI

Table DynamicallyDefineLocalIdentifier Request Message - 2C_03 (3-byte addressing)

7.28.3 DynamicallyDefineLocalIdentifier Request - DefineByMemoryAddress - 2C_03

(4-byte addressing)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**DynamicallyDefineLocalIdentifier Request Service Id**	**M**	**#2Ch**	**DDLI**
#2d	DynamicallyDefinedLocalIdentifier	M	#F0h	DDDLI
#3d	DefinitionMode=XX [DefineByMemoryAddress]	M	#XXh= [#03h]	DNM DBMA
#4d	PositionInDynamicallyDefinedLocalIdentifier	M	#XXh	PIDDDLI

#5d	MemorySize	M	#XXh	MS
#6d	MemoryAddress (High Byte)	M	#XXh	MA_HB
#7d	MemoryAddress (MiddleHigh Byte)	M	#XXh	MA_MHB
#8d	MemoryAddress (MiddleLow Byte)	M	#XXh	MA_MLB
#9d	MemoryAddress (Low Byte)	M	#XXh	MA_LB
#10d	MemoryTypeIdentifier **(mandatory, but not evaluated)** [00h … FFh]	M	#XXh [#XXh]	MTI
...	...	C	...	
#n-7d	DefinitionMode=XX [DefineByMemoryAddress]	C	#XXh= [#03h]	DNM DBMA
#n-6d	PositionInDynamicallyDefinedLocalIdentifier	C	#XXh	PIDDDLI
#n-5d	MemorySize	C	#XXh	MS
#n-4d	MemoryAddress (High Byte)	C	#XXh	MA_HB
#n-3d	MemoryAddress (MiddleHigh Byte)	M	#XXh	MA_MHB
#n-2d	MemoryAddress (MiddleLow Byte)	M	#XXh	MA_MLB
#n-1d	MemoryAddress (Low Byte)	C	#XXh	MA_LB
#nd	MemoryTypeIdentifier **(mandatory, but not evaluated)** [00h … FFh]	M	#XXh [#XXh]	MTI

Table DynamicallyDefineLocalIdentifier Request Message - 2C_03 (4-byte addressing)

Remark:

For 3-byte addressing **independent from delivery of a MemoryTypeIdentifier (byte 9) by the tester´s request, the ECU automatically adds an offset to the requested memory address.**
If no MemoryTypeIdentifier is delivered, the same offset as for MemoryTypeIdentifier 00h (linear addressing) is used.
This base memory offset must be subtracted from any memory address taken from the MAP- / A2L- file if 3-byte addressing is not sufficient for the address range!

For 4-byte addressing **a MemoryTypeIdentifier (byte 10) must be delivered by the tester´s request. Though it is not evaluated or used within the ECU for this absolute addressing mode, it is necessary to decide between 3- / 4-byte addressing mode.**
In this 4-byte mode the absolute address delivered by the tester is used by the ECU without any memory offset.

For MSx80 (TriCore) the base memory offsets for 3-byte addressing **are:**

MemoryTypeIdentifier:		Memory offset:	
none	**(linear)**		**C000 0000h**
00h	**(linear)**		**C000 0000h**
02h	**(ext. FLASH)**		**8000 0000h (the same as for int. FLASH)**
05h	**(ext. RAM)**	**D000 0000h**	
06h	**(int. FLASH)**	**8000 0000h**	
07h	**(UIF)**		**8000 xxxxh not fixed yet, address might still change !!!**

The DDLI Service with the definitionMode "defineByMemoryAddress“ can only read memory areas, which are also readable by RMBA-service (for details about allowed memory areas see specification 17I00J02, "SID23h: ReadMemoryByAddress").

7.28.4 DynamicallyDefineLocalIdentifier Request - ClearDDLI - 2C_04

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
#1d	**DynamicallyDefineLocalIdentifier Request Service Id**	**M**	**#2Ch**	**DDLI**
#2d	DynamicallyDefinedLocalIdentifier	M	#F0h	DDDLI
#3d	DefinitionMode=XX [ClearDynamicallyDefinedLocalIdentifier]	M	#XXh= [#04h]	DNM CDDDLI

Table DynamicallyDefineLocalIdentifier Request Message – 2C_04

7.28.5 DynamicallyDefineLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**DynamicallyDefineLocalIdentifier Positive Response Id**	**M**	**#6Ch**	**DDLIPR**
#2d	DynamicallyDefinedLocalIdentifier	M	#F0h	DDDLI

Table DynamicallyDefineLocalIdentifier Positive Response Message

7.28.6 DynamicallyDefineLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	DynamicallyDefineLocalIdentifier Request Service Id	M	#2Ch	DDLI
#3d	responseCode= [subFunctionNotSupported-invalidFormat, conditionNotCorrect, requestOutOfRange, uploadNotAccepted, can'tUploadNumberOfBytesRequested, general response codes]	M	#XXh= [#12h, #22h, #31h, #50h, #53h, #XXh]	RC

Table DynamicallyDefineLocalIdentifier Negative Response Message

NegativeResponseCode	*Cause of Occurrence*
SubFunctionNotSupported-invalidFormat	- selected RCI is not supported - selected DDDLI is not supported
ConditionNotCorrect	selected DNM is not supported
RequestOutOfRange	selected PIRCI does not exist
UploadNotAccepted	error in PIDDDLI information (outside DDLI area)
can'tUploadNumberOfBytesRequested	the requested information is bigger than the reserved buffer

7.29 SID 22h: KWP readDataByCommonIdentifier Service

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Bosanfsgbd	O	0...FFH	0...255	1	[-]
BOS-Lesen-Anforderung vom Tester					
CTR_BOS_READ	V	0...FFFFH	0...65535	1	[-]
timeout counter for BOS read					
Ecu_sw_ref_bmw[9]	O	0...FFH	ASCII	1	[-]
BMW-ECU software reference					
LV_BOS_READ_REQ	V	0...1H	0...1	1	[-]
flag indicating request BOS data for tester					

Input data:

ALTER_COD_0	ALTER_COD_1	Amo_05	Amo_10
Amo_15	Amo_20	Amo_30	Amo_40
ANG_1_RAW_VVL	ANG_DE_ABSV_PLAUS_CHK_VVL	B_cdxenonr	B_dev0detec
B_dev3detec	B_dev6detec	B_gangnull	B_kupp_ext
B_kwdreht	B_msasw	B_nggelernt	B_ngimlf
B_qvch2o	B_spa_cist	B_spa_csoll	Ba_ist
Ba_soll	Bosbtvfbk	Bosconf	Bosmanip
Bosmziel	Bosprog2	Bosres	Bosrlsm
Bosrw2	Bosstate	Bostoken	Bszsi
C_ERR_CLAS_XX	CAM_EX[NC_NR_CBK_IVVT]	CAM_IN[NC_NR_CBK_IVVT]	CHK_STAMP
CONF_EF	CONF_SOF_SWI	CPPWM_CPS	CTR_AD_COLD_LAM_AD_INJ
CTR_AD_HOT_LAM_AD_INJ	CTR_CDN_RBM[NC_NR_DIAG_RBM]	CTR_COMP_RBM[NC_NR_DIAG_RBM]	CTR_DC[NC_NR_ERR_DYN]
CTR_ERR_DYN_NR	CTR_ERR_HIS_NR	CTR_FRC[NC_NR_ERR_DYN]	CTR_KM_N_MAX
CTR_N_MAX	CTR_NS_AD_CYC[NC_NOX_SENS_CONF]	CTR_NS_SHIFT_CYC[NC_NOX_SENS_CONF]	CTR_NT_AGI_AD_CMPL_SUM
CTR_NT_AGI_SO2P_FQ	CTR_TOIL_MAX	CTR_WUP_CYC[NC_NR_ERR_DYN]	DC_MAX_XX
Dfds[16]	Dfmonitor	DIST_LAM_AD_INJ_COLD	DIST_LAM_AD_INJ_HOT
DIST_NS_NEW	DIST_NT_NS_AD[NC_NOX_SENS_CONF]	DIST_NT_NS_SHIFT[NC_NOX_SENS_CONF]	DIST_TOIL_MAX
EGY_SP_IV_EXT_ADJ[NC_CYL_NR]	Eisydk_korfak_b	Eisydk_koroff_b	Eisyev_korfak_b
Eisyev_koroff_b	ENVD_CUS_SET_CMN[NC_NR_ERR_DYN][NC_NR_FRF_SET][NC_NR_ENVD_CUS_SET_CMN]	ENVD_CUS_SET_SPC[NC_NR_ERR_DYN][NC_NR_FRF_SET][NC_NR_ENVD_CUS_SET_SPC]	ENVD_OBD[NC_NR_ERR_DYN][NC_NR_ENVD_OBD]
ER_MMV_IS_DIAG[NC_CYL_NR]	ERR_DTC[NC_NR_ERR_DYN]	ERR_TYPE_BYTE[NC_NR_ERR_DYN]	ERR_TYPE_EXT_BYTE[NC_NR_ERR_DYN]
Exwink_ist	Exwinkkor	F_minhub	F_tikorrvr[NC_CYL_NR]
FAC_LAM_AD_CUS[NC_CBK_EX_NR]	FAC_LAM_ADJ_COR_LAM_AD_CUS[NC_CBK_EX_NR]	FAC_LAM_ADJ_LAM_AD[NC_CBK_EX_NR]	FAC_LAM_LIM[NC_CBK_EX_NR]
FAC_MFF_ADD_EXT_ADJ	FAC_NOX_NS_AD[NC_NOX_SENS_CONF]	FAC_NT_AGI_LIM	FAC_NT_AGI_MDL
FAC_OIL_EXT_REQ_1	FAC_OIL_EXT_REQ_2	GEAR_EF_N_MAX	GEAR_TOIL_MAX
HIS[NC_NR_HIS][NC_NR_ERR_HIS]	HIS_COD[NC_NR_ERR_HIS]	Ibhwversi	Ibswbase
Ibswchang	Id_bosmgt	IGA_IGC[NC_CYL_NR]	Igenk

Igrinfo[30]	IMOB_KWP_K_EWS4[16]	IMOB_KWP_K_EWS4_TMP0[16]	IMOB_KWP_STATE_0
IMOB_KWP_STATE_15	IMOB_KWP_STATE_8	IMOB_NR_VLD_K_EWS4	Ivvtmot[NC_CYL_NR]
K_EWS4[16]		Leminfo[40]	LF_BSD_CPT_AVL
LF_CWP_INFO_COD	LF_POW_CWP_COD	LSHPWM_DOWN[NC_CBK_EX_NR]	LSHPWM_UP[NC_CBK_EX_NR]
Lurabs_f[NC_CYL_NR]	Lurdif_f[NC_CYL_NR]	LV_ACCOUT_RLY	LV_ALTER_CTL_ENA
LV_AT	LV_CAT_CONF_DIS_EXT_REQ	LV_CS	LV_CS_N_MAX
LV_CS_TOIL_MAX	LV_DRI	LV_ES	LV_FAC_MFF_ADD_EXT_ADJ_NVMY
LV_FL	LV_IGK	LV_IGK_OFF_ACK_ENA	LV_IM_BLS
LV_IM_BTS	LV_IS	LV_KD	LV_LAM_LSCL[NC_CBK_EX_NR]
LV_LS_DOWN_READY[NC_CBK_EX_NR]	LV_LS_UP_READY[NC_CBK_EX_NR]	LV_NT_AFS_REQ_AGI	LV_NT_AFS_REQ_AGI_TMP_3
LV_PUC	LV_RST_MFF_ADD_EXT_ADJ_NVMY	LV_SEG_AD_AVL_ER	LV_SENS_BAT_SMT_DET
LV_SO2P_REQ_1	LV_SO2P_REQ_2	LV_SO2P_REQ_FQ	LV_STST_VAR_COD
LV_TPS_AD_REQ	LV_VAR_ACIN	LV_VAR_AEB	LV_VAR_AMT
LV_VAR_ARS	LV_VAR_ASR_3	LV_VAR_ASR_4	LV_VAR_BN
LV_VAR_BN_EFP	LV_VAR_BN_GEAR_REV	LV_VAR_BN_LDM	LV_VAR_BN_LTG_HDLP_L
LV_VAR_BN_MSW	LV_VAR_BN_TRL	LV_VAR_DCC	LV_VAR_EAC
LV_VAR_EBOX_CFA	LV_VAR_ECRAS_DOWN	LV_VAR_ECRAS_UP	LV_VAR_EF
LV_VAR_ETCU	LV_VAR_ETCU_SPT	LV_VAR_ICL	LV_VAR_LSH_DOWN
LV_VAR_LSH_UP	LV_VAR_MAF	LV_VAR_MSW	LV_VAR_NOX
LV_VAR_PSTE	LV_VAR_PSTE_2	LV_VAR_PSTE_3	
		LV_VAR_RLY_ACCOUT	LV_VAR_RLY_ST
LV_VAR_SAP	LV_VAR_SAV	LV_VAR_SOF	LV_VAR_STST
LV_VAR_TCT	LV_VAR_TQ_PBR	LV_VAR_VEH	MAF_KGH_MES_BAS
MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR]	MFF_ADD_COLD_LAM_AD_INJ[NC_CYL_NR]	MFF_ADD_EXT_ADJ[NC_CBK_EX_NR]	MFF_ADD_EXT_ADJ_NVMY
MFF_ADD_HOT_LAM_AD_INJ[NC_CYL_NR]	MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	Minhub	Minhub_roh
Minhubvs	Minhubvs_ist	Minhubvsnv	Msa_arravrs[60]
Msa_indexrs	Msainfo[50]	Msastz	Msastzmsa
N	N_GRD_N_MAX	N_KWP_OFS	N_KWP_OFS_ACC
N_KWP_OFS_ACC_DRI	N_KWP_OFS_DRI	N_KWP_OFS_VB	N_MAX
N_SP_IS	N_TOIL_MAX	NC_CBK_EX_NR	NC_CYL_NR
NC_K_EWS_MAX	NC_K_EWS_UNLOCKED	NC_NOX_SENS_CONF	NC_NR_CAM_CBK
NC_NR_DIAG_RBM	NC_NR_EDGE_CAM_EX	NC_NR_EDGE_CAM_IN	NC_NR_ERR_DYN
NC_NR_ERR_HIS	NC_NT_NR	NC_PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	NC_PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]
NL[NC_CYL_NR]	NR_PAT_SCC_N_MAX	NT_AGI	NT_AGI_SO2P_FQ_SUM
NT_AGI_SUL	NT_AGI_SUL_SNG[NC_NT_NR]	NT_AGI_THERMO	NT_AGI_THERMO_SNG[NC_NT_NR]
NT_SUL_32[NC_NT_NR]	NT_SUL_H_32[NC_NT_NR]	OBD_FUP_RNG_H	Oz_manip
Pmbackup[7]	Pminfo1[37]	Pminfo2[29]	POW_CONF_IDX_EXT_REQ
PROT_CONF_IDX_EXT_REQ	PSN_EDGE_AD_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	PSN_EDGE_AD_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	PV_AV_N_MAX
PV_AV_TOIL_MAX	Qv_cdherst_1	Qv_cdherst_2	Qv_cdherst_3
Qv_cdherst_4	Qv_cdherst_5	Qv_cdherst_6	Qv_cdherst_7
Qv_cdherst_8	Qv_h2o	Qv_h2oquali	Qv_h2ostatus
Qv_nv_ezm	Qv_nv_zh	Qv_out_1	Qv_out_2
Qv_out_3	Qv_out_4	Qv_out_5	Qv_out_m

Qv_quali_1	Qv_quali_2	Qv_quali_3	Qv_quali_4
Qv_quali_5	Qv_quali_m	Qv_status	Qv_td1
Qv_td2	Qv_td3	Qv_td4	Qv_td5
Qvc_status_1	Qvc_status_2	Qvc_status_3	Qvc_status_4
RATIO_MMV_NS_SHIFT_DIAG[NC_NOX_SENS_CONF]	S_vsmnhb	S_vsmnhbnv	Snibs
St_dgenerrst_md1	St_dgenerrst_md2	St_dgengrenz1	St_dgengrenz1_md1
St_dgengrenz1_md2	St_dgengrenz2	St_dgengrenz2_md1	St_dgengrenz2_md2
St_dgengrenzerr	St_dgengrenzerr_md1	St_dgengrenzerr_md2	St_dgengrenznz
St_dgengrenznz_md1	St_dgengrenznz_md2	St_dgenub1	St_dgenub1_md1
St_dgenub1_md2	St_dgenub2	St_dgenub2_md1	St_dgenub2_md2
St_dgenuberr	St_dgenuberr_md1	St_dgenuberr_md2	St_dgenubnz
St_dgenubnz_md1	St_dgenubnz_md2	St_dgenugen1	St_dgenugen1_md1
St_dgenugen1_md2	St_dgenugen2	St_dgenugen2_md1	St_dgenugen2_md2
St_dgenugenerr	St_dgenugenerr_md1	St_dgenugenerr_md2	St_dgenugennz
St_dgenugennz_md1	St_dgenugennz_md2	St_dvovrld	St_ngang0
St_vbrvs_aus	St_vbrvs_ein	St_vvt_err	STATE_CRU_OFF_IRR
STATE_CRU_OFF_REV	STATE_ECRAS_UP_VAR	STATE_EGY_MIN_KWP	STATE_IMOB_K_EWS
STATE_MIL_ON_DIS_EXT_REQ	STATE_PSTE_3_SRC	STATE_TBL_DRIV[8][6]	Stmsa
Stmsaaa	Stmsaav	Stmsaea	Stmsaev
Swmsaav	T_N_MAX	T_SUM_N_MAX	TAM_TOIL_MAX
TCO_2_MES	TCO_MES	TCO_TOIL_MAX	Tget_b1
Tget_b2	Tget_b3	Tget_b4	Tget_b5
TI_1_HOM[NC_CYL_NR]	TIA	Tmot_b1	Tmot_b2
Tmot_b3	Tmot_b4	Tmot_b5	Toel_b1
Toel_b2	Toel_b3	Toel_b4	Toel_b5
TOIL_MAX_WARN	TOIL_THD_TOIL_MAX	TPS_AV_1	TQI_AV
TQI_AV_TOIL_MAX	TRT_N_MAX	Tumg_b1	Tumg_b2
Tumg_b3	Tumg_b4	Tumg_b5	Tvneutral
Tvngang	V_PVS_1	VB	VCC_DR_VVL
Verbredinfo[32]	VLS_UP[NC_CBK_EX_NR]	VS	VS_H_N_MAX
VS_MAX_SEL_EXT_REQ	VS_N_MAX	VS_TOIL_MAX	Vvt_soll
Zbibs	Zr_lvs_0	Zr_lvs_1	Zr_lvs_2
Zr_lvs_3	Zr_lvs_ll_reakt	Zr_lvssekt_0	Zr_lvssekt_1
Zr_lvssekt_2	Zr_lvssekt_3	Zr_lvssekt_4	Zr_lvssekt_5
Zr_lvssekt_6	Zr_lvssekt_7	Zr_lvssekt_8	Zr_lvszyl_0
Zr_lvszyl_1	Zr_lvszyl_2	Zr_lvszyl_3	Zr_lvszyl_4
Zr_lvszyl_5	Zrbosmld	Zw_offkorrvr[NC_CYL_NR]	Zylhubkor

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_SENS_ZK	1	0...1H	0...1	1	[-]
type of spark (1- sensierte Zündkerzen, 0- nicht sensierte Zündkerzen)					

General information:

All implemented SID 22h services are described below.

7.29.1 Overview of implemented subservices

SID number	service	service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85 (L4)
22_10_00	Read check stamp	ECU	X	X	X	X	X	X
22_10_01	Read value BOS	ECU	X	X	X	X	X	X
22_10_0A	Read Energy saving status	ECU	X	X	X	X	X	X
22_16_00	Read Anzahl Subbusteilnehmer	ECU	X	X	X	X	X	X
22_16_01	Read Identifikation Subbusteilnehmer 1 (normaly: Batteriesensensor)	ECU	X	X	X	X	X	X
22_16_02	Read Identifikation Subbusteilnehmer 2 (normaly:elektr. Wasserpumpe)	ECU	X	X	X	X	X	X
22_16_03	Read Identifikation Subbusteilnehmer 3 (normaly:Generator 1)	ECU	X	X	X	X	X	X
22_16_04	Read Identifikation Subbusteilnehmer 34(normaly:ECRAS_UP)	ECU	-	-	-	-	-	X
22_20_00	read failure memory short	ECU	X	X	X	X	X	X
22_21_00	Read History Memory	ECU	X	X	X	X	X	X
22_25_00	Programm Reference Backup	BOOT	X	X	X	X	X	X
22_25_01	Erasing time	BOOT	X	X	X	X	X	X
22_25_02	Hardware reference	BOOT	X	X	X	X	X	X
22_25_03	Programm reference	BOOT	X	X	X	X	X	X
22_25_04	Data reference	BOOT	X	X	X	X	X	X
22_25_06	Maximum of block length	Boot-/ECU	X	X	X	X	X	X
22_30_00	MIL Codierung lesen	ECU	X	X	X	X	X	X
22_30_01	Kat Codierung lesen	ECU	X	X	X	X	X	X
22_30_10	VMAX-Codierdaten lesen	ECU	X	X	X	X	X	X
22_30_20	OL/UL Codierung lesen	ECU	X	X	X	X	X	X
22_30_30	Status-Codierung-Protokoll lesen	ECU	X	X	X	X	X	-
22_32_00	Oelwechselintervall- Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_10	IGR-Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_11	Xenonverbau-Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_20	SPA-Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_30	BZE-Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_40	Abgasklappe Codierdaten lesen	ECU	X	X	X	X	X	X
22_32_50	Codierung MSA lesen	ECU	-	X	-	X	-	-
22_32_60	Read CDASMOT	ECU	X	X	X	X	X	X
22_3F_FF	Aenderungsindex Codierdaten lesen	ECU	X	X	X	X	X	X
22_40_00	Read values 4000	ECU	X	X	X	X	X	X
22_40_01	Read values “batterie generator”	ECU	X	X	X	X	X	X
22_40_02	Read Switch Position	ECU	X	X	X	X	X	X
22_40_03	Read values of engine roughness	ECU	X	X	X	X	X	X

22_40_06	Read CAMSHAFT-Sensor Adaption	ECU	X	X	X	X	X	X
22_40_07	Read State of Functions	ECU	X	X	X	X	X	X
22_40_08	Read Throttle and MAF-Adjustment Value	ECU	X	X	X	X	X	X
22_40_0A	Read mixture values	ECU	X	X	X	X	X	X
22_40_0B	VVT Messwerte auslesen	ECU	X	-	-	-	-	-
22_40_15	Read FASTA_Messwertblock 10	ECU	X	X	X	X	X	X
22_40_16	READ IGR-values	ECU	X	X	X	X	X	X
22_40_17	READ LEM-values	ECU	X	X	X	X	X	X
22_40_18	READ MSA-values	ECU	X	X	X	X	X	-
22_40_19	Fsp. Lesen mit Freeze Frame	ECU	X	X	X	X	X	X
22_40_1A	READ BZE-values	ECU	X	X	X	X	X	X
22_40_1B	Read: Results of intelligent alternator diagnosis	ECU	X	X	X	X	X	X
22_40_1C	Ringspeicher MSA lesen	ECU	X	X	X	X	X	-
22_40_1D	Read Verbredinfo	ECU	-	-	-	-	-	X
22_40_1E	Read CBS-Zündkerzen	ECU	-	X	X	X	X	-
22_40_1F	Read BMW-PST (BMW-Programmstandsinformation)	ECU	-	-	-	-	-	X
22_40_21	READ IBS-values	ECU	X	X	X	X	X	X
22_40_22	READ PM1 Status	ECU	X	X	X	X	X	X
22_40_23	READ PM2 Status	ECU	X	X	X	X	X	X
22_40_26	Rate Based Monitoring Mode 9 auslesen	ECU	X	X	X	X	X	X
22_40_27	Rate Based Monitoring Motorsteuerung MSV80 Block 1 auslesen	ECU	X	-	-	-	-	-
22_40_27	Rate Based Monitoring Motorsteuerung MSD8x Block 1 auslesen	ECU	-	X	X	X	X	X
22_40_28	Rate Based Monitoring Motorsteuerung MSV80 Block 2 auslesen	ECU	X	-	-	-	-	-
22_40_28	Rate Based Monitoring Motorsteuerung MSD8x Block 2 auslesen	ECU	-	X	X	X	X	X
22_40_2D	Messwerte Laufruhepruefung auslesen	ECU	X	X	X	X	X	X
22_40_2E	Nullgang-Erkennung lesen	ECU	X	X	X	X	X	-
22_40_2F	MSA lesen	ECU	X	X	X	X	X	-
22_40_30	FASTA_LVS_lesen	ECU	-	X	X	X	X	-
22_40_31	Laufruhe Verbesserungssystem Zylinderstatistik auslesen	ECU	-	X	X	X	X	-
22_40_32	MFMA (Kleinstmengenadaption)	ECU	-	X	X	X	X	X
22_5F_88	Adrecovery lesen	ECU	-	X	X	X	X	-
22_5F_8B	read PM recovery	ECU	X	X	X	X	X	X
22_5F_8E	MSA deactivation lesen	ECU	X	X	X	X	X	-
22_5F_8F	MSA Abschaltverhinderer lesen	ECU	X	X	X	X	X	-
22_5F_90	IMAALLE lesen	ECU	-	X	X	X	X	X
22_5F_DE	VVT- Minhub lesen	ECU	X	-	-	-	-	-
22_5F_F0	Abgleichswert LL lesen	ECU	X	X	X	X	X	X
22_5F_F1	CO Abgleichswert lesen	ECU	X	-	-	-	-	-
22_5F_F2	Varianten lesen	ECU	X	X	X	X	X	X

22_5F_F3	Uebertemperatursicherung lesen	ECU	**X**	**X**	**X**	**X**	**X**	**X**
22_5F_F4	Fusshebelwerk Fehlbedienung lesen	ECU	**X**	**X**	**X**	**X**	**X**	**-**
22_5F_F6	Ueberdrehzahlsicherung lesen	ECU	**X**	**X**	**X**	**X**	**X**	**X**
22_5F_F7	Laufruhepruefung auslesen	ECU	**X**	**X**	**X**	**X**	**X**	**-**
22_5F_F8	Status Homogenbetrieb auslesen	ECU	**-**	**X**	**X**	**X**	**X**	**-**
22_C0_00	Read imob status (STATUS_EWS)	Boot-/ECU	**X**	**X**	**X**	**X**	**X**	**X**
22_C0_02	Read imob k_ews(tmp) (STATUS_EWS_SK)	Boot-/ECU	**X**	**X**	**X**	**X**	**X**	**X**

7.29.2 Global Application conditions (default)

If no Application conditions block is specified, use default

Application conditions:

Initialisation:	at reset
Activation:	Kl.15 on
Deactivation:	Kl.15 off Diagnostic timeout
Service location:	see list of “implemented diagnostic services”

7.29.3 Global negative Responses (default)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**negativeResponse Service ID**	**M**	**7Fh**	**NR**
2	Request Service ID	M	22h	requestServiceID
3	responseCode (12h - subfunction not suppoerted, 22h – conditionsNotCorrectOr RequestSequenceError)	M	XXh [#12h, #22h]	responseCode

Remark:
If no negative response is specified, send Global negative Responses
The detailed message of the negative response code is described in chapter "Error Codes".

7.29.4 22_1000 - Read check stamp

General information:

With this comand the 3 bytes of the check stamp was read out to the ecu. Those are actually in the volatile memory. In the latch off phase (n = 0 rpm and Kl.15 off) this values are written down into the non volatile memory. If the battery voltage is removed from the ECU, bevor the power latch phase ends, so the storage of the check stamp was cancelled and the value up to now was keept.
The writing of the check stamp was reserved for the customers production, that means that the check stamp can´t be read by the ECU production factory.

The service only accept ram-reading.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	10h	RCI_TSP
3	RecordCommonIdentifier	M	00h	RCI_TSP

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	10h	RCI_TSP
3	RecordCommonIdentifier	M	00h	RCI_TSP
4	recordValue1 CHK_STAMP Byte 2	M	XXh	RV_PST
5	recordValue2 CHK_STAMP Byte 1	M	XXh	RV_PST
6	recordValue3 CHK_STAMP Byte 0 (LSB)	M	XXh	RV_PST

7.29.5 22_1001 - Read values BOS

General information:

This function is created to read out the described values from the ecu.

Formula section:

```
If  service 22_1001 is recieved
then
      send 78 H
endif

If(1) 78H handling is activated for this function
Then(1)
      increment CTR_BOS_READ
      if(2)     CTR_BOS_READ > 1C2H            // timeout condition
      then(2)send negative response
                      LV_BOS_READ_REQ          = 0
                      CTR_BOS_READ             = 0
                      Bosanfsgbd               = 0
                      End data handling 78H
                                               // End data transmission to tester
      else(2)
            if(3)      LV_BOS_READ_REQ =0
            then(3)
                      If(4)      Bosconf = 1. bit
                      then(4)          Bosanfsgbd = 1
                                       LV_BOS_READ_REQ = 1
                      else(4)          send negative response
                                       LV_BOS_READ_REQ          = 0
                                       CTR_BOS_READ             = 0
                                       Bosanfsgbd               = 0
                                             End data handling 78H
                                                   // End data transmission to tester
                      endif(4)
            else(3)
                      if(4)     Bosanfsgbd = Bostoken       // data for tester are ready
                      then(4) LV_BOS_READ_REQ = 0
                            send positive response
                            LV_BOS_READ_REQ           = 0
                            CTR_BOS_READ              = 0
```

```
                    Bosanfsgbd                      = 0
                    End data handling 78H
                                                    // End data transmission to tester
                endif(4)
            endif(3)
        endif(2)
    endif(1)
```

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier read values	M	10h	RCI_
3	RecordCommonIdentifier	M	01h	RCI_

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	10h	RCI_BOS
3	RecordCommonIdentifier	M	01h	RCI_BOS
4	recordValue1 = Id_bosmgt for first CBS-member	M	XXh	RV_Id_bosmgt
5	recordValue2 = Bosrlsm high byte for first CBS-member	M	XXh	RV_Bosrlsm
6	recordValue3 = Bosrlsm low byte for first CBS-member	M	XXh	RV_Bosrlsm
7	recordValue4 = Bosstate for first CBS-member	M	XXh	RV_Bosstate
8	recordValue5 = Zrbosmld for first CBS-member	M	XXh	RV_Zrbosmld
9	recordValue6 = Bosbtvfbk for first CBS-member	M	XXh	RV_Bosbtvfbk
10	recordValue7 = Bosrw2 for first CBS-member	M	XXh	RV_Bosrw2
11	recordValue8 = Bosmziel for first CBS-member	M	XXh	RV_Bosmziel
12	recordValue9 = Bosjziel for first CBS-member	M	XXh	RV_Bosjziel
13	recordValue10 = Bosprog2 for first CBS-member	M	XXh	RV_Bosprog2
14	recordValue11 = Bosmanip for first CBS-member	M	XXh	RV_Bosmanip
15	recordValue12 = Bosres for first CBS-member	M	XXh	RV_Bosres
16	recordValue13 = Bostoken for first CBS-member	M	XXh	RV_Bostoken

7.29.6 22_100A - RDBCI_EGY_lesen

General information:

Read energy saving status
This function is created to read out the described values from the ecu.
Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	10h	RCI_ESS
3	RecordCommonIdentifier Low Byte	M	0Ah	RCI_ESS

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	10h	RDBCI_EGY_lesen
3	RecordCommonIdentifier Low Byte	M	0Ah	RDBCI_EGY_lesen
4	recordValue1 = STATE_EGY_MIN_KWP	M	XXh	RV_STATE_EGY_MIN_KWP

7.29.7 22_1600 - Read number of intelligente subbus members

General information:

This function is created to read out the number of intelligente sensors on Subbus.

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier read values	M	16h	RCI_
3	RecordCommonIdentifier	M	00h	RCI_

Positive Response on service (not MSD85)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	00h	RCI_BOS
4	RecordValue1= number of 1-bits from LF_BSD_CPT_AVL Bit0, Bit3 and Bit6 (LF_BSD_CPT_AVL & 49h) = number of sucessfully learnt intelligent sensors (posible components: Intelligent battery sensor,electrical coolant water pump and alternator) on subbus	M	XXh [00h, 01h, 02h, 03h]	RV_

Positive Response on service (only MSD85)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	00h	RCI_BOS
4	RecordValue1= number of 1-bits from LF_BSD_CPT_AVL Bit0, Bit3 and Bit6 (LF_BSD_CPT_AVL & 49h) + LV_VAR_ECRAS_UP = number of sucessfully learnt intelligent sensors on BSD-BUS (possible components: Intelligent battery sensor,electrical coolant water pump and alternator) plus number of sucessfully learnt intelligent sensors on LIN-BUS (possible component: ECRAS UP)	M	XXh [00h, 01h, 02h, 03h, 04h]	RV_

7.29.8 22_1601 - Read detailed informations of the first intelligent subbus member

General information:

This function is created to read out the detailed informations of the first intelligente subbus member.

Request Service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier read values	M	16h	RCI_
3	RecordCommonIdentifier	M	01h	RCI_

Formula section (not for MSD85):

If(1) service 22_16_01 is received
Then(1)
If(2) B_dev0detec = 1
Then(2) send Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	01h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	01h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = always zero	M	00h	RCI_
7	RecordValue 4 = always zero	M	00h	RCI_
8	RecordValue 5 = Zbibs (MSB)	M	XXh	RCI_
9	RecordValue 6 = Zbibs	M	XXh	RCI_
10	RecordValue 7 = Zbibs	M	XXh	RCI_
11	RecordValue 8 = Zbibs (LSB)	M	XXh	RCI_

If (3) B_dev0detec = 0 **and** B_dev3detec = 1
Then(3) send Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	01h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	02h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = LF_CWP_INFO_COD	M	XXh	RCI_
13	RecordValue 10 = LF_POW_CWP_COD	M	XXh	RCI_

If (4) B_dev0detec = 0 **and** B_dev3detec **= 0 and** B_dev6detec = 1
Then(4) send Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	01h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	03h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_

6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = ALTER_COD_0	M	XXh	RCI_
13	RecordValue 10 = ALTER_COD_1	M	XXh	RCI_

```
                Endif(4)
            Endif(3)
        Else(2)         send negative Response "Request out of range" ( 31h)
        Endif(2)
    Endif(1)
```

Formula section (only MSD85):

```
If(1)     service 22_16_01 is received
Then(1)
        If(2)           B_dev0detec = 1
        Then(2)         send Positive Response
```

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	01h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	01h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = always zero	M	00h	RCI_
7	RecordValue 4 = always zero	M	00h	RCI_
8	RecordValue 5 = Zbibs (MSB)	M	XXh	RCI_
9	RecordValue 6 = Zbibs	M	XXh	RCI_
10	RecordValue 7 = Zbibs	M	XXh	RCI_
11	RecordValue 8 = Zbibs (LSB)	M	XXh	RCI_

```
        Else(2)

            If (3)          B_dev0detec = 0 and B_dev3detec = 1
            Then(3)         send Positive Response
```

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	01h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	02h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = LF_CWP_INFO_COD	M	XXh	RCI_
13	RecordValue 10 = LF_POW_CWP_COD	M	XXh	RCI_

```
Else(3)
    If (4) B_dev0detec = 0 and B_dev3detec = 0 and B_dev6detec = 1
    Then(4)        send Positive Response
```

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	01h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	03h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = ALTER_COD_0	M	XXh	RCI_
13	RecordValue 10 = ALTER_COD_1	M	XXh	RCI_

```
    Else(4)
        If (5)      LV_VAR_ECRAS_UP = 1
              and   B_dev0detec = 0
              and   B_dev3detec = 0
              And   B_dev6detec = 0
        Then(5)     send Positive Response
```

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	01h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	3Dh	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = STATE_ECRAS_UP_VAR	M	XXh	RCI_

```
        Else(5)
            send negative Response "Request out of range" ( 31h)
        Endif(5)
    Endif(4)
Endif(3)

  Endif(2)
Endif(1)
```

7.29.9 22_1602 - Read detailed informations of second intelligent subbus member

General information:

This function is created to read out the detailed informations of second intelligent subbus member

Request Service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier read values	M	16h	RCI_
3	RecordCommonIdentifier	M	02h	RCI_

Formula section (not for MSD85):

If(1) service 22_16_02 is received
Then(1)
If(2) B_dev0detec = 1 **and** B_dev3detec = 1
Then(2) send Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	02h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	02h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = LF_CWP_INFO_COD	M	XXh	RCI_
13	RecordValue 10 = LF_POW_CWP_COD	M	XXh	RCI_

Elseif(2)
If(3) (B_dev0detec = 1 **and** B_dev3detec= 0 **and** B_dev6detec = 1)
or
(B_dev0detec = 0 **and** B_dev3detec= **1 and** B_dev6detec = 1)
Then(3) send Positive Response

1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	02h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	03h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_

12	RecordValue 9 = ALTER_COD_0	M	XXh	RCI_
13	RecordValue 10 = ALTER_COD_1	M	XXh	RCI_

```
        Else(3)       send negative Response "Request out of range" ( 31h)
        Endif(3)
    Endif(2)
  Endif(1)
```

<u>Formula section (only for MSD85):</u>

```
If(1)   service 22_16_02 is received
Then(1)
    If(2)   B_dev0detec = 1 and B_dev3detec = 1
    Then(2)       send Positive Response
```

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	02h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	02h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = LF_CWP_INFO_COD	M	XXh	RCI_
13	RecordValue 10 = LF_POW_CWP_COD	M	XXh	RCI_

```
    Elseif(2)
        If(3)         (B_dev0detec = 1 and B_dev3detec= 0 and B_dev6detec = 1)
              or
                      (B_dev0detec = 0 and B_dev3detec= 1 and B_dev6detec = 1)
        Then(3)       send Positive Response
```

1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	02h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	03h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = ALTER_COD_0	M	XXh	RCI_
13	RecordValue 10 = ALTER_COD_1	M	XXh	RCI_

```
        Else(3)
```

```
            If(4)       LV_VAR_ECRAS_UP = 1
                and
                (       (B_dev0detec = 1 and B_dev3detec= 0 and B_dev6detec = 0)
                        or
                        (B_dev0detec = 0 and B_dev3detec= 1 and B_dev6detec = 0)
                        or
                        (B_dev0detec = 0 and B_dev3detec= 0 and B_dev6detec = 1)
                )
            Then(4)     send Positive Response
```

1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	02h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	3Dh	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = STATE_ECRAS_UP_VAR	M	XXh	RCI_

```
            Else(4)     send negative Response "Request out of range" ( 31h)
            Endif(4)
        Endif(3)
    Endif(2)
Endif(1)
```

7.29.10 22_1603 - Read detailed informations of third intelligent subbus member

General information:

This function is created to read out the detailed informations of third intelligente subbus member.

Request Service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier read values	M	16h	RCI_
3	RecordCommonIdentifier	M	03h	RCI_

Formula section (not for MSD85):

```
If(1)       service 22_16_03 is received
```

```
Then(1)
    If(2)        B_dev0detec = 1 and B_dev3detec= 1 and B_dev6detec = 1
    Then(2)      send Positive Response
```

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	03h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	03h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = ALTER_COD_0	M	XXh	RCI_
13	RecordValue 10 = ALTER_COD_1	M	XXh	RCI_

```
    Else(2)      send negative Response "Request out of range" ( 31h)
    Endif(2)
Endif(1)
```

Formula section (onyl for MSD85):

```
If(1)        service 22_16_03 is received
Then(1)
    If(2)        B_dev0detec = 1 and B_dev3detec= 1 and B_dev6detec = 1
    Then(2)      send Positive Response
```

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	03h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	03h	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = ALTER_COD_0	M	XXh	RCI_
13	RecordValue 10 = ALTER_COD_1	M	XXh	RCI_

```
    Elseif(2)
        If(3)        LV_VAR_ECRAS_UP = 1
                 and
                 (
                         (B_dev0detec = 1 and B_dev3detec= 1 and B_dev6detec = 0)
                     or  (B_dev0detec = 1 and B_dev3detec= 0 and B_dev6detec = 1)
```

```
            or  (B_dev0detec = 0 and B_dev3detec= 1 and B_dev6detec = 1)
         )
      Then(3)       send Positive Response
```

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	03h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	3Dh	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = STATE_ECRAS_UP_VAR	M	XXh	RCI_

```
      Else(3)       send negative Response "Request out of range" ( 31h)
      Endif(3)
   Endif(2)
Endif(1)
```

7.29.11 22_1604 - Read detailed informations of fourth intelligent subbus member

General information:

This function is created to read out the detailed informations of fourth intelligente subbus member.

Request Service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier read values	M	16h	RCI_
3	RecordCommonIdentifier	M	04h	RCI_

Formula section (not for MSD85):

```
If(1)     service 22_16_04 is received
Then(1)   send negative Response "Request out of range" (31h)
Endif(1)
```

Formula section (only for MSD85):

```
If(1)     service 22_16_04 is received
Then(1)
```

```
If(2)           B_dev0detec = 1
            and B_dev3detec = 1
            and B_dev6detec = 1
            and LV_VAR_ECRAS_UP = 1
Then(2)         send Positive Response
```

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	16h	RCI_BOS
3	RecordCommonIdentifier	M	04h	RCI_BOS
4	RecordValue 1 = Verbauort (high byte)	M	3Dh	RCI_
5	RecordValue 2 = Verbauort (low byte)	M	00h	RCI_
6	RecordValue 3 = BMW-part number (always 0ffh)	M	FFh	RCI_
7	RecordValue 4 = BMW-part number (always 0ffh)	M	FFh	RCI_
8	RecordValue 5 = BMW-part number (always 0ffh)	M	FFh	RCI_
9	RecordValue 6 = BMW-part number (always 0ffh)	M	FFh	RCI_
10	RecordValue 7 = BMW-part number (always 0ffh)	M	FFh	RCI_
11	RecordValue 8 = BMW-part number (always 0ffh)	M	FFh	RCI_
12	RecordValue 9 = STATE_ECRAS_UP_VAR	M	XXh	RCI_

```
    Else(2)     send negative Response "Request out of range" ( 31h)
    Endif(2)
Endif(1)
```

7.29.12 22_1605 ..22_16FF Read detailed informations of other intelligent subbus member

General information:

This function is created to read out the detailed informations of other intelligente subbus members

Request Service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier read values	M	16h	RCI_
3	RecordCommonIdentifier	M	XXh [05h, .. FFh]	RCI_

Formula section:

```
If(1)     service 22_16_05..FF is received
Then(1)   send negative Response "Request out of range" ( 31h)
Endif(1)
```

7.29.13 22_2000 - Read failure memory short

General information:

This function is to read out the numbers of faults in the error memory and the status of CDK (see also “Fault-type-byte and fault-type extension byte” (chapter OBDII functions)).

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	20h	RCI_RDTCBS
3	RecordCommonIdentifier Low Byte	M	00h	RCI_RDTCBS

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Response Service ID	M	62h	RDBCIPR
2	RecordCommonIdentifier High Byte	M	20h	RCI_RDTCBS
3	RecordCommonIdentifier Low Byte	M	00h	RCI_RDTCBS
4	numberOfCDK - CTR_ERR_DYN_NR	M	01h-XXh	NROCDK
5..24	ListOfCDKAndStatus= [CDK0 high byte ERR_DTC[0] CDK0 low byte ERR_DTC[0] statusOfCDK 1 ERR_TYPE_BYTE[0] CDK1 high byte ERR_DTC[1] CDK1 low byte ERR_DTC[1] statusOfCDK 2 ERR_TYPE_BYTE[1] . . . CDK(NC_NR_ERR_DYN - 1) CDK(NC_NR_ERR_DYN - 1) statusOfCDK n ERR_TYPE_BYTE[NC_NR_ERR_DYN - 1]]	C C C C C C . . . C C C	[XXh XXh XXh XXh XXh XXh . . . XXh XXh XXh]	LOCDKAS

7.29.14 22_2100 - Read History Memory

General information:

This function is created to read out the history memory with all the errors (max. 10, see also “History memory” (chapter OBDII functions)).. The checksum calculation for each error is a simple XOR of all the bytes belonging to that error.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	21h	RCI_
3	RecordCommonIdentifier Low Byte	M	00h	RCI_

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier= Read History Memory	M	21h	RCI_TSP
3	RecordCommonIdentifier	M	00h	RCI_TSP
4	RecordValue 1 = Error counter total (Siemens name: CTR_ERR_HIS_NR)	M	XXh	RV_
5	RecordValue 2 = Error code “CDK” high byte (Siemens name: HIS_COD[x])	M	XXh	RV_
6	RecordValue 3 = Error code “CDK” low byte (Siemens name: HIS_COD[x])	M	XXh	RV_
7	RecordValue 4 = status DTC1 (Siemens name: HIS[1][x])	M	XXh	RV_
8	RecordValue 5 = Frequency counter (Siemens name: HIS[2][x])	M	XXh	RV_
9	RecordValue 6 = km-value by first occur high byte (Siemens name: HIS[3][x])	M	XXh	RV_
10	RecordValue 7 = km-value by first occur low byte (Siemens name: HIS[4][x])	M	XXh	RV_
11	RecordValue 8 = km-value by second occur high byte (Siemens name: HIS[5][x])	M	XXh	RV_
12	RecordValue 9 = km-value by second occur low byte (Siemens name: HIS[6][x])	M	XXh	RV_
13	RecordValue 8 = km-value by last occur high byte (Siemens name: HIS[7][x])	M	XXh	RV_
14	RecordValue 9 = km-value by last occur low byte (Siemens name: HIS[8][x])	M	XXh	RV_
15	RecordValue 10 = Error class “CLA” (Siemens name: HIS[9][x])	M	FFh	RV_
16	RecordValue 11= checksum for this DTC	M	FFh	RV_
17	next failure - convention see 5d as follows	M	XXh	RV_
18	:			RV_
19	:			RV_
5 +(n*9)	RecordValue 2 +(n*9)d 11= checksum for the last DTC (maximum is 10 DTC´s)	M	FFh	RV_

7.29.15 22_2500 - Programm Reference Backup

General information:

This function is for reading the Programmreference backup from the logistic data of the ecu.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on
engine speed=0rpm

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	25h	RCI_

3	RecordCommonIdentifier Low Byte	M	00h	RCI_

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	25h	RCI_TSP
3	RecordCommonIdentifier Low Byte	M	00h	RCI_TSP
4+ 3*(12+6)	recordValue1 backup of programm reference system - cif_backup	M	XXh	RV_

7.29.16 22_2501 - Erasing time

General information:

This function ist for reading the erasing time of the flash memory of the ecu.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on
engine speed=0rpm

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	25h	RCI_
3	RecordCommonIdentifier Low Byte	M	01h	RCI_

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	25h	RCI_TSP
3	RecordCommonIdentifier Low Byte	M	01h	RCI_TSP
4	recordValue1 max. erasing time del_time_max	M	XXh	RV_
5	recordValue1 max. time for signature check kwp_sigt	M	XXh	RV_
6	recordValue1 max. time for ECU reset kwp_rst	M	XXh	RV_
7	recordValue1 max. time for authentification kwp_autht	M	XXh	RV_

7.29.17 22_2502 - Hardware reference

General information:

This function ist for reading the hardware reference of the ecu.

Application conditions:

Initialisation: at reset

Activation: Kl.15 on
engine speed=0rpm

Deactivation: Kl.15 off
Diagnostic timeout
Service location: see list of “implemented diagnostic services”

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	25h	RCI_
3	RecordCommonIdentifier Low Byte	M	02h	RCI_

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	25h	RCI_TSP
3	RecordCommonIdentifier Low Byte	M	02h	RCI_TSP
4	recordValue1 hardware reference hw_reference	M	XXh	RV_

7.29.18 22_2503 - Programm reference

General information:

This function ist for reading the programm reference of the ecu.

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	25h	RCI_
3	RecordCommonIdentifier Low Byte	M	03h	RCI_

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	25h	RCI_TSP
3	RecordCommonIdentifier Low Byte	M	03h	RCI_TSP
4 6+3*12	recordValue1 ecu_sw_reference	M	XXh	RV_

7.29.19 22_2504 - Data reference

General information:

This function ist for reading the data reference of the ecu.

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	25h	RCI_
3	RecordCommonIdentifier Low Byte	M	04h	RCI_

Table Request Message

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**

2	RecordCommonIdentifier High Byte	M	25h	RCI_TSP
3	RecordCommonIdentifier Low Byte	M	04h	RCI_TSP
4 6+3*17	RecordValue1 cal_ref_nr	M	XXh	RV_

7.29.20 22_2506 - Maximum of block length

General information:

This function is for reading the maximum of block length, witch is possible for the ecu.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	25h	RCI_
3	RecordCommonIdentifier Low Byte	M	06h	RCI_

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	25h	RCI_TSP
3	RecordCommonIdentifier Low Byte	M	06h	RCI_TSP
4	RecordValue1 high byte = kwp_max_tel_len maximun block length	M	XXh	RV_
5	RecordValue1 low byte = kwp_max_tel_len maximun block length	M	XXh	RV_

7.29.21 22_3000 - RCI_MIL_lesen

General information:

MIL Codierung lesen

This function is created to read out the selected value of MIL (malfunction indication lamp) variant code.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	30h	RCI_MIL_lesen
3	RecordCommonIdentifier Low Byte	M	00h	RCI_MIL_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	30h	RCI_MIL_lesen
3	RecordCommonIdentifier Low Byte	M	00h	RCI_MIL_lesen
4	recordValue = STATE_MIL_ON_DIS_EXT_REQ	M	XXh	RV_STATE_MIL_ON_DIS_EXT_REQ

7.29.22 22_3001 - RCI_KAT_lesen

General information:

Kat Codierung lesen
This function is created to read out the selected value of catalyst variant code.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	30h	RCI_KAT_lesen
3	RecordCommonIdentifier Low Byte	M	01h	RCI_KAT_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	30h	RCI_KAT_lesen
3	RecordCommonIdentifier Low Byte	M	01h	RCI_KAT_lesen
4	recordValue 1= LV_CAT_CONF_DIS_EXT_REQ	M	XXh	RV_LV_CAT_CONF_DIS_EXT_REQ

7.29.23 22_3010 - RCI_VMAX_lesen

General information:

VMAX-Codierdaten lesen
This function is created to read out the selected value for maximum vehicle speed. The values 0 ... 3 represent one of four possible values for maximum vehicle speed.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	30h	RCI_VMAX_lesen
3	RecordCommonIdentifier Low Byte	M	10h	RCI_VMAX_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	30h	RCI_VMAX_lesen
3	RecordCommonIdentifier Low Byte	M	10h	RCI_VMAX_lesen
4	recordValue 1 = VS_MAX_SEL_EXT_REQ	M	XXh	RV_VS_MAX_SEL_EXT_REQ

7.29.24 22_3020 - RCI_CLS_lesen

General information:

STATUS_CODIERUNG_LEISTUNGSSTUFE
This function is created to read out the coding of power stage (Leistungsstufe)

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	30h	RCI_CLS_lesen
3	RecordCommonIdentifier Low Byte	M	20h	RCI_CLS_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	30h	RCI_CLS_lesen
3	RecordCommonIdentifier Low Byte	M	20h	RCI_CLS_lesen
4	recordValue 1 = POW_CONF_IDX_EXT_REQ	M	XXh	RV_POW_CONF_IDX_EXT_REQ

7.29.25 22_3030 - Status Codierung Protokoll lesen

General information:

This function is created to read out the active communication protocol (14230 <-> 15765-4).

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	30h	RCI_CDSP
3	RecordCommonIdentifier Low Byte	M	30h	RCI_CDSP

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	30h	RCI_CDSP
3	RecordCommonIdentifier Low Byte	M	30h	RCI_CDSP
4	recordValue = PROT_CONF_IDX_EXT_REQ	M	XXh	RV_CDSP

7.29.26 22_3200 – RCI_OEL_lesen

General information:

Oelwechsel intervall- Codierdaten lesen

This function is created to read out the country dependant factors for oil change intervall.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	32h	RCI_OEL_lesen
3	RecordCommonIdentifier Low Byte	M	00h	RCI_OEL_lesen

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	32h	RCI_OEL_lesen
3	RecordCommonIdentifier Low Byte	M	00h	RCI_OEL_lesen
4	recordValue 1 = FAC_OIL_EXT_REQ_1	M	XXh	RV_FAC_OIL_EXT_REQ_1
5	recordValue 2 = FAC_OIL_EXT_REQ_2	M	XXh	RV_FAC_OIL_EXT_REQ_2

7.29.27 22_3210 – RCI_IGR_lesen

General information:

Read coding of IGR (Intelligente Generator-Regelung)
This function is created to read out the selected value intelligent alternator control variant code.

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	32h	RCI_IGR_lesen
3	RecordCommonIdentifier Low Byte	M	10h	RCI_IGR_lesen

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	32h	RCI_IGR_lesen
3	RecordCommonIdentifier Low Byte	M	10h	RCI_IGR_lesen
4	recordValue = LV_ALTER_CTL_ENA	M	XXh	RV_ LV_ALTER_CTL_ENA

7.29.28 22_3211 – Xenonverbau-Codierdaten lesen

General information:

This function is created to read out the selected value of "xenon-filled headlight mounted" variant code.

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	32h	RCI_CDXEN
3	RecordCommonIdentifier Low Byte	M	11h	RCI_CDXEN

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	32h	RCI_CDXEN
3	RecordCommonIdentifier Low Byte	M	11h	RCI_CDXEN
4	recordValue = B_cdxenonr	M	XXh	RV_B_CDXENON

7.29.29 22_3220 – RCI_SPA_lesen

General information:

SPA-Codierdaten lesen
This function is created to read out the "Schaltpunktanzeige"

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Request Service ID	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	32h	RCI_SPA_lesen
3	RecordCommonIdentifier Low Byte	M	20h	RCI_SPA_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	32h	RCI_SPA_lesen
3	RecordCommonIdentifier Low Byte	M	20h	RCI_SPA_lesen
4	recordValue1 = B_spa_csoll	M	XXh	RV_SPA_CSOLL
5	recordValue1 = B_spa_cist	M	XXh	RV_SPA_CIST

7.29.30 22_3230 - RCI_BZE_lesen

General information:

BZE-Codierdaten lesen
This function is created to read out the "Batteriezustands Erkennung"

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Request Service ID	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	32h	RCI_BZE_lesen
3	RecordCommonIdentifier Low Byte	M	30h	RCI_BZE_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	32h	RCI_BZE_lesen
3	RecordCommonIdentifier Low Byte	M	30h	RCI_BZE_lesen
4	recordValue1 = Qv_cdherst_1	M	XXh	RV_CD_HERST1
5	recordValue2 = Qv_cdherst_2	M	XXh	RV_CD_HERST2
6	recordValue3 = Qv_cdherst_3	M	XXh	RV_CD_HERST3
7	recordValue4 = Qv_cdherst_4	M	XXh	RV_CD_HERST4
8	recordValue5 = Qv_cdherst_5	M	XXh	RV_CD_HERST5
9	recordValue6 = Qv_cdherst_6	M	XXh	RV_CD_HERST6
10	recordValue7 = Qv_cdherst_7	M	XXh	RV_CD_HERsT7
11	recordValue8 = Qv_cdherst_8	M	XXh	RV_CD_HESRT8

7.29.31 22_3240 - RCI_AGK_lesen

General information:

Read Coding Exhaust gas flap
This function is created to read out "Codierung fuer Abgasklappe auslesen"

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	Request Service ID	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	32h	RCI_AGK_lesen
3	RecordCommonIdentifier Low Byte	M	40h	RCI_AGK_lesen

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	32h	RCI_AGK_lesen
3	RecordCommonIdentifier Low Byte	M	40h	RCI_AGK_lesen
4	recordValue1 = CONF_EF	M	XXh	RV_CODIERUNG_AGK

7.29.32 22_3250 - Read Coding of MSA

General information:

With this comand the coding value of MSA is read out.

Application conditions:

Initialisation: ---

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formula Section:

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	32h	RCI_CDMSA
3	RecordCommonIdentifier	M	50h	RCI_CDMSA

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**

2	RecordCommonIdentifier	M	32h	RCI_CDMSA
3	RecordCommonIdentifier	M	50h	RCI_CDMSA
4	recordValue1 = LV_STST_VAR_COD	M	XXh	RV_CODIERUNG_MSA

7.29.33 22_3260 - Read CDASMOT

General information:

This function is created to read out the "Codierung elektrische Diagnose der Funktion Abschaltung Klemme 15"

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Request Service ID	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	32h	RCI_CDASMOT
3	RecordCommonIdentifier Low Byte	M	60h	RCI_CDASMOT

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	32h	RCI_CDASMOT
3	RecordCommonIdentifier Low Byte	M	60h	RCI_CDASMOT
5	recordValue1 = LV_IGK_OFF_ACK_ENA (0 - diagnosis off, 1- diagnosis on)	M	XXh [0h, 1h]	RV_CODIERUNG_ASMOT

7.29.34 22_4000 - RCI_MW_lesen

General information:

STATUS_MESSWERTE

This function is created to read out the described values from the ecu.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MW
3	RecordCommonIdentifier Low Byte	M	00h	RCI_MW

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MW
3	RecordCommonIdentifier Low Byte	M	00h	RCI_MW
4	recordValue1 = TI_1_HOM[0] high byte	M	XXh	RV_TI_1_HOM[0]
5	recordValue2 = TI_1_HOM[0] low byte	M	XXh	RV_TI_1_HOM[0]

6	recordValue3 = FAC_LAM_LIM[[1] high byte	M	XXh	RV_FAC_LAM_LIM[[1]
7	recordValue4 = FAC_LAM_LIM[[1] low byte	M	XXh	RV_FAC_LAM_LIM[[1]
8	recordValue5 = FAC_LAM_LIM[[2] high byte	M	XXh	RV_FAC_LAM_LIM[[2]
9	recordValue6 = FAC_LAM_LIM[[2] low byte	M	XXh	RV_FAC_LAM_LIM[[2]
10	recordValue7 = VS	M	XXh	RV_VS
11	recordValue8 = N high byte	M	XXh	RV_N
12	recordValue9 = N low byte	M	XXh	RV_N
13	recordValue10 = N_SP_IS high byte	M	XXh	RV_N_SP_IS
14	recordValue11 = N_SP_IS low byte	M	XXh	RV_N_SP_IS
15	recordValue12 **=** CAM_IN[1] high byte -if word **(not MSD85)** 0 - if byte **(not MSDS85)** CAM_IN_H[1] high byte **(only MSD85)**	M	XXh	RV_CAM_IN_H[1]
16	recordValue13 = CAM_IN[1] low byte- if word **(not MSD85)** CAM_IN[1] – if byte **(not MSD85)** CAM_IN_H[1] low byte (only MSD85)	M	XXh	RV_CAM_IN_H[1]
17	recordValue14 = CAM_EX[1]	M	XXh	RV_CAM_EX[1]
18	recordValue15 = Dummy	M	FFh	RV_DUMMY_1
19	recordValue16 = TIA	M	XXh	RV_TIA
20	recordValue17 = TCO_MES	M	XXh	RV_TCO_MES
21	recordValue18 = IGA_IGC[0]	M	XXh	RV_IGA_IGC[0]
22	recordValue19 = TPS_AV_1 high byte (**not MS85)** TPS_AV_1[0] high byte **(only MSD85)**	M	XXh	RV_TPS_AV_1[1]
23	recordValue20 = TPS_AV_1 low byte (**not MSD85)** TPS_AV_1[0] low byte **(only MSD85)**	M	XXh	RV_TPS_AV_1[1]
24	recordValue21 = MAF_KGH_MES_BAS high byte (**not MSD85)** MAF_KGH_MES_BAS[0] high byte **(only MSD85)**	M	XXh	RV_MAF_KGH_MES_BAS[1]
25	recordValue22 = MAF_KGH_MES_BAS low byte (**not MSD85)** MAF_KGH_MES_BAS[0] high byte **(only MSD85)**	M	XXh	RV_MAF_KGH_MES_BAS[1]
26	recordValue23 = TQI_AV high byte	M	XXh	RV_TQI_AV
27	recordValue24 = TQI_AV low byte	M	XXh	RV_TQI_AV
28	recordValue25 = VB	M	XXh	RV_VB
29	recordValue26 = V_PVS_1 high byte	M	XXh	V_PVS_1
30	recordValue27 = V_PVS_1 low byte	M	XXh	RV_V_PVS_1
31	recordValue28 = TCO_2_MES	M	XXh	RV_TCO_2_MES
32	recordValue29 = NL[0] high byte	M	XXh	RV_NL[0]
33	recordValue30 = NL[0] low byte	M	XXh	RV_NL[0]
34	recordValue31 = NL[1] high byte	M	XXh	RV_NL[1]
35	recordValue32 = NL[1] low byte	M	XXh	RV_NL[1]
36	recordValue33 = NL[2] high byte	M	XXh	RV_NL[2]
37	recordValue34 = NL[2] low byte	M	XXh	RV_NL[2]
38	recordValue35 = NL[3] high byte	M	XXh	RV_NL[3]
39	recordValue36 = NL[3] high byte	M	XXh	RV_NL[3]
40	recordValue37 = NL[4] low byte	M	XXh	RV_NL[4]
41	recordValue38 = NL[4] low byte	M	XXh	RV_NL[4]
42	recordValue39 = NL[5] high byte	M	XXh	RV_NL[5]
43	recordValue40 = NL[5] low byte	M	XXh	RV_NL[5]
44	recordValue41 = NL[6] low byte	M	XXh	RV_NL[6]
45	recordValue42 = NL[6] low byte	M	XXh	RV_NL[6]
46	recordValue43 = NL[7] high byte	M	XXh	RV_NL[7]
47	recordValue44 = NL[7] low byte	M	XXh	RV_NL[7]

Note: RecordValue data for not available components are FFh

7.29.35 22_4001 - RCI_BLZ_lesen

General information:

Read values “batterie generator”

This function is created to read out the described values from the ecu.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_BLZ_lesen
3	RecordCommonIdentifier Low Byte	M	01h	RCI_BLZ_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_BLZ_lesen
3	RecordCommonIdentifier Low Byte	M	01h	RCI_BLZ_lesen
4	recordValue1 = Dfmonitor	M	XXh	Dfmonitor

7.29.36 22_4002 - RCI_SSZ1_lesen

General information:

Read Switch Position

This function is created to read the actual position of the car switches.
All switches are active “1” in the Bit position, as well as ground was switched.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_SSZ1_lesen
3	RecordCommonIdentifier Low Byte	M	02h	RCI_SSZ1_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_SSZ1_lesen
3	RecordCommonIdentifier Low Byte	M	02h	RCI_SSZ1_lesen
4	recordValue1 = Bitfield1	M	XXh	RV_SWITCH_POSITION

Note: For data, longer than one byte, the lowest byte is sent first

Byte #	*Name*	*Size*	*Conversion (hex/bin)*	*Conversion (physical)*	*Mnemonic*	*Resol.*
#4d	Bitfield 1	1	xxxxxxx1 b xxxxxx1x b xxxxx1xx b xxxx1xxx b xxx1xxxx b xx1xxxxx b x1xxxxxx b 1xxxxxxx b	LV_IGK LV_ES LV_CS LV_IM_BLS LV_IM_BTS not used not used LV_ACCOUT_RLY	RV_LV_IGK RV_LV_ES RV_LV_CS RV_LV_IM_BLS RV_LV_IM_BTS RV_LV_ACCOUT_RLY	

7.29.37 22_4003 - RCI_MLR_lesen

General information:

Read values of engine roughness
This function is created to read out the values from the function-test engine roughness. The calculation of the function is described in the chapter OBDII functions.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MLR
3	RecordCommonIdentifier Low Byte	M	03h	RCI_MLR

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MLR
3	RecordCommonIdentifier Low Byte	M	03h	RCI_MLR
4	recordValue1 high byte = ER_MMV_IS_DIAG[0]	M	XXh	RV_ER_MMV_IS_DIAG[0]
5	recordValue2 low byte = ER_MMV_IS_DIAG[0]	M	XXh	RV_ER_MMV_IS_DIAG[0]
6	recordValue3 high byte = ER_MMV_IS_DIAG[1]	M	XXh	RV_ER_MMV_IS_DIAG[1]
7	recordValue4 low byte = ER_MMV_IS_DIAG[1]	M	XXh	RV_ER_MMV_IS_DIAG[1]
8	recordValue5 high byte = ER_MMV_IS_DIAG[2]	M	XXh	RV_ER_MMV_IS_DIAG[2]
9	recordValue6 low byte = ER_MMV_IS_DIAG[2]	M	XXh	RV_ER_MMV_IS_DIAG[2]
10	recordValue7 high byte = ER_MMV_IS_DIAG[3]	M	XXh	RV_ER_MMV_IS_DIAG[3]
11	recordValue8 low byte = ER_MMV_IS_DIAG[3]	M	XXh	RV_ER_MMV_IS_DIAG[3]
12	recordValue9 high byte = ER_MMV_IS_DIAG[4]	M	XXh	RV_ER_MMV_IS_DIAG[4]
13	recordValue10 low byte = ER_MMV_IS_DIAG[4]	M	XXh	RV_ER_MMV_IS_DIAG[4]
14	recordValue11 high byte = ER_MMV_IS_DIAG[5]	M	XXh	RV_ER_MMV_IS_DIAG[5]
15	recordValue12 low byte = ER_MMV_IS_DIAG[5]	M	XXh	RV_ER_MMV_IS_DIAG[5]
16	recordValue13 Byte = LV_SEG_AD_AVL_ER	M	XXh	RV_LV_SEG_AD_AVL_ER
17	recordValue14 high byte = VLS_UP[1]	M	XXh	RV_VLS_UP[1]
18	recordValue15 low byte = VLS_UP[1]	M	XXh	RV_VLS_UP[1]
19	recordValue16 high byte = VLS_UP[2]	M	XXh	RV_VLS_UP[2]
20	recordValue17 low byte = VLS_UP[2]	M	XXh	RV_VLS_UP[2]
21	recordValue18 high byte = ER_MMV_IS_DIAG[6]	M	XXh	RV_ER_MMV_IS_DIAG[6]
22	recordValue19 low byte = ER_MMV_IS_DIAG[6]	M	XXh	RV_ER_MMV_IS_DIAG[6]
23	recordValue 20 high byte = ER_MMV_IS_DIAG[7]	M	XXh	RV_ER_MMV_IS_DIAG[7]
24	recordValue 21 low byte = ER_MMV_IS_DIAG[7]	M	XXh	RV_ER_MMV_IS_DIAG[7]

Note: RecordValue data for not available components are FFh

7.29.38 22_4006 - RCI_NWA_lesen

General information:

Read CAMSHAFT-Sensor Adaption

This function is created to read the actual adaption data form the camshaft sensor of the engine. All switches are active “1” in the Bit position, as well as ground was switched.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_NWA_lesen
3	RecordCommonIdentifier Low Byte	M	06h	RCI_NWA_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_NWA_lesen
3	RecordCommonIdentifier Low Byte	M	06h	RCI_NWA_lesen
4	recordValue1 = NC_PSN_EDGE_CAM_EX[1][1] high Byte	M	XXh	RV_NC_PSN_EDGE_CAM_EX[1][1]
5	RecordValue 2 = NC_PSN_EDGE_CAM_EX[1][1] low Byte	M	XXh	RV_NC_PSN_EDGE_CAM_EX[1][1]
6	RecordValue 3= NC_PSN_EDGE_CAM_EX[6][1] high Byte	M	XXh	RV_NC_PSN_EDGE_CAM_EX[6][1]
7	RecordValue 4 = NC_PSN_EDGE_CAM_EX[6][1] low Byte	M	XXh	RV_NC_PSN_EDGE_CAM_EX[6][1]
8	RecordValue 5 = NC_PSN_EDGE_CAM_IN[1][1] high Byte	M	XXh	RV_NC_PSN_EDGE_CAM_IN[1][1]
9	RecordValue 6 = NC_PSN_EDGE_CAM_IN[1][1] low Byte	M	XXh	RV_NC_PSN_EDGE_CAM_IN[1][1]
10	RecordValue 7 = NC_PSN_EDGE_CAM_IN[6][1] high Byte	M	XXh	RV_NC_PSN_EDGE_CAM_IN[6][1]
11	RecordValue 8 = NC_PSN_EDGE_CAM_IN[6][1] low Byte	M	XXh	RV_NC_PSN_EDGE_CAM_IN[6][1]
12	RecordValue 9 = PSN_EDGE_AD_CAM_IN[1] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[1] [1]
13	RecordValue 10 = PSN_EDGE_AD_CAM_IN[1] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[1] [1]
14	RecordValue 11 = PSN_EDGE_AD_CAM_IN[2] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[2] [1]
15	RecordValue 12 = PSN_EDGE_AD_CAM_IN[2] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[2] [1]
16	RecordValue 13 = PSN_EDGE_AD_CAM_IN[3] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[3] [1]
17	RecordValue 14 = PSN_EDGE_AD_CAM_IN[3] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[3] [1]
18	RecordValue 15 = PSN_EDGE_AD_CAM_IN[4] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[4] [1]
19	RecordValue 16 = PSN_EDGE_AD_CAM_IN[4] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[4] [1]
20	RecordValue 17 = PSN_EDGE_AD_CAM_IN[5] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[5] [1]
21	RecordValue 18 = PSN_EDGE_AD_CAM_IN[5] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[5] [1]
22	RecordValue 19 = PSN_EDGE_AD_CAM_IN[6] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[6] [1]
23	RecordValue 20 = PSN_EDGE_AD_CAM_IN[6] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[6] [1]
24	RecordValue 21 = PSN_EDGE_AD_CAM_EX[1] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[1] [1]
25	RecordValue 22 = PSN_EDGE_AD_CAM_EX[1] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[1] [1]
26	RecordValue 23 = PSN_EDGE_AD_CAM_EX[2] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[2] [1]
27	RecordValue 23 = PSN_EDGE_AD_CAM_EX[2] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[2] [1]
28	RecordValue 25 = PSN_EDGE_AD_CAM_EX[3] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[3] [1]
29	RecordValue 26 = PSN_EDGE_AD_CAM_EX[3] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[3] [1]

30	RecordValue 27 = PSN_EDGE_AD_CAM_EX[4] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[4] [1]
31	RecordValue 28 = PSN_EDGE_AD_CAM_EX[4] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[4] [1]
32	RecordValue 29 = PSN_EDGE_AD_CAM_EX[5] [1] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[5] [1]
33	RecordValue 30 = PSN_EDGE_AD_CAM_EX[5] [1] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[5] [1]
34	RecordValue 31 = PSN_EDGE_AD_CAM_EX[6] [1] high Byte	M	XXh	RV_SN_EDGE_AD_CAM_EX[6] [1]
35	RecordValue 32 = PSN_EDGE_AD_CAM_EX[6] [1] low Byte	M	XXh	RV_SN_EDGE_AD_CAM_EX[6] [1]
36	RecordValue 33 = NC_PSN_EDGE_CAM_EX[1] [2] high Byte	M	XXh	RV_NC_PSN_EDGE_CAM_EX[1] [2]
37	RecordValue 34 = NC_PSN_EDGE_CAM_EX[1] [2] low Byte	M	XXh	RV_NC_PSN_EDGE_CAM_EX[1] [2]
38	RecordValue 35 = NC_PSN_EDGE_CAM_EX[6] [2] high Byte	M	XXh	RV_NC_PSN_EDGE_CAM_EX[6] [2]
39	RecordValue 36 = NC_PSN_EDGE_CAM_EX[6] [2] low Byte	M	XXh	RV_NC_PSN_EDGE_CAM_EX[6] [2]
40	RecordValue 37 = NC_PSN_EDGE_CAM_IN[1] [2] high Byte	M	XXh	RV_NC_PSN_EDGE_CAM_IN[1] [2]
41	RecordValue 38 = NC_PSN_EDGE_CAM_IN[1] [2] low Byte	M	XXh	RV_NC_PSN_EDGE_CAM_IN[1] [2]
42	RecordValue 38 = NC_PSN_EDGE_CAM_IN[6] [2] high Byte	M	XXh	RV_NC_PSN_EDGE_CAM_IN[6] [2]
43	RecordValue 39 = NC_PSN_EDGE_CAM_IN[6] [2] low Byte	M	XXh	RV_NC_PSN_EDGE_CAM_IN[6] [2]
44	RecordValue 40 = PSN_EDGE_AD_CAM_IN[1][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[1][2]
45	RecordValue 41 = PSN_EDGE_AD_CAM_IN[1][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[1][2]
46	RecordValue 42 = PSN_EDGE_AD_CAM_IN[2][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[2][2]
47	RecordValue 43 = PSN_EDGE_AD_CAM_IN[2][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[2][2]
48	RecordValue 44 = PSN_EDGE_AD_CAM_IN[3][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[3][2]
49	RecordValue 45 = PSN_EDGE_AD_CAM_IN[3][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[3][2]
50	RecordValue 46 = PSN_EDGE_AD_CAM_IN[4][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[4][2]
51	RecordValue 47 = PSN_EDGE_AD_CAM_IN[4][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[4][2]
52	RecordValue 48 = PSN_EDGE_AD_CAM_IN[5][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[5][2]
53	RecordValue 49 = PSN_EDGE_AD_CAM_IN[5][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[5][2]
54	RecordValue 50 = PSN_EDGE_AD_CAM_IN[6][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[6][2]
55	RecordValue 50 = PSN_EDGE_AD_CAM_IN[6][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_IN[6][2]
56	RecordValue 40 = PSN_EDGE_AD_CAM_EX[1][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[1][2]
57	RecordValue 41 = PSN_EDGE_AD_CAM_EX[1][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[1][2]
58	RecordValue 42 = PSN_EDGE_AD_CAM_EX[2][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[2][2]

59	RecordValue 43 = PSN_EDGE_AD_CAM_EX[2][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[2][2]
60	RecordValue 44 = PSN_EDGE_AD_CAM_EX[3][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[3][2]
61	RecordValue 45 = PSN_EDGE_AD_CAM_EX[3][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[3][2]
62	RecordValue 46 = PSN_EDGE_AD_CAM_EX[4][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[4][2]
63	RecordValue 47 = PSN_EDGE_AD_CAM_EX[4][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[4][2]
64	RecordValue 48 = PSN_EDGE_AD_CAM_EX[5][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[5][2]
65	RecordValue 49 = PSN_EDGE_AD_CAM_EX[5][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[5][2]
66	RecordValue 50 = PSN_EDGE_AD_CAM_EX[6][2] high Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[6][2]
67	RecordValue 50 = PSN_EDGE_AD_CAM_EX[6][2] low Byte	M	XXh	RV_PSN_EDGE_AD_CAM_EX[6][2]

Note: RecordValue data for not available components are FFh

7.29.39 22_4007 - RCI_SSZ0_lesen

General information:

Read State of Functions

This function is created to read the actual condition of the engine.
All switches are active “1” in the Bit position, as well as ground was switched.

7.29.39.1 MSD 85 only

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_SSZ0_lesen
3	RecordCommonIdentifier Low Byte	M	07h	RCI_SSZ0_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_SSZ0_lesen
3	RecordCommonIdentifier Low Byte	M	07h	RCI_SSZ0_lesen
4	RecordValue 1 Bitfield1	M	XXh	2_BYTE_SWITCH_POSITION
5	RecordValue 2 Bitfield2	M	XXh	2_BYTE_SWITCH_POSITION
6	Not used	M	FFh	RV_DUMMY_1
7	Not used	M	FFh	RV_DUMMY_2
8	Not used	M	FFh	RV_DUMMY_3
9	Not used	M	FFh	RV_DUMMY_4

➲ Note: For data longer than one byte, the lowest byte is sent first

Byte	Name	Size	Conversion (hex/bin)	Conversion (physical)		Resol.
4	Bitfield 1	1	xxxxxxx1 b	LV_IS	RV_LV_IS	
			xxxxxx1x b	LV_FL	RV_LV_FL	
			xxxxx1xx b	LV_LS_DOWN_READY[1]	RV_LV_LS_DOWN_READY[1]	
			xxxx1xxx b	LV_LS_DOWN_READY[2]	RV_LV_LS_DOWN_READY[2]	
			xxx1xxxx b	LV_LS_UP_READY[2]	RV_LV_LS_UP_READY[2]	
			xx1xxxxx b	LV_LS_UP_READY[1]	RV_LV_LS_UP_READY[1]	
			x1xxxxxx b	LV_LAM_LSCL[2]	RV_LV_LAM_LSCL[2]	
			1xxxxxxx b	LV_LAM_LSCL[1]	RV_LV_LAM_LSCL[1]	
Byte	**Name**	**Size**	**Conversion (hex/bin)**	**Conversion (physical)**		**Resol.**
5	Bitfield 2	1	xxxxxxx1 b	not used	not used	
			xxxxxx1x b	not used	not used	
			xxxxx1xx b	LV_KD	RV_LV_KD	
			xxxx1xxx b	LV_DRI	RV_LV_DRI	
			xxx1xxxx b	not used	not used	
			xx1xxxxx b	not used	not used	
			x1xxxxxx b	LV_PUC	RV_LV_PUC	
			1xxxxxxx b	LV_TPS_AD_REQ	RV_LV_TPS_AD_REQ	
			1xxxxxxx b			

7.29.39.2 All other projects

General information:

This function is created to read the actual condition of the engine.
All switches are active “1” in the Bit position, as well as ground was switched.

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier = Read state of the functions	M	40h	RCI_FUN
3	RecordCommonIdentifier	M	07h	RCI_FUN

Table ReadDataByCommonIdentifier Request Message

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier= Read state of the functions	M	40h	RCI_SCH
3	RecordCommonIdentifier	M	07h	RCI_SCH
4	RecordValue Bitfield1	M	XXh	RV_B1
5	RecordValue Bitfield2	M	XXh	RV_B2
6	RecordValue Bitfield3	M	XXh	RV_B3
7	RecordValue Bitfield4	M	XXh	RV_B4
8	RecordValue Bitfield5	M	XXh	RV_B5
9	RecordValue Bitfield6	M	XXh	RV_B6

Table ReadDataByCommonIdentifier Positive Response Message

➲ Note: For data longer than one byte, the lowest byte is sent first

Byte	Name	Size	Conversion (hex/bin)	Conversion (physical)	Resol.
4	Bitfield 1	1	xxxxxxx1 b	LV_IS	
			xxxxxx1x b	LV_FL	
			xxxxx1xx b	LV_LS_DOWN_READY[1]	
			xxxx1xxx b	LV_LS_DOWN_READY[2]	
			xxx1xxxx b	LV_LS_UP_READY[2]	
			xx1xxxxx b	LV_LS_UP_READY[1]	
			x1xxxxxx b	LV_LAM_LSCL[2]	
			1xxxxxxx b	LV_LAM_LSCL[1]	
Byte	**Name**	**Size**	**Conversion (hex/bin)**	**Conversion (physical)**	**Resol.**
5	Bitfield 2	1	xxxxxxx1 b	not used	
			xxxxxx1x b	not used	
			xxxxx1xx b	LV_KD	
			xxxx1xxx b	LV_DRI	
			xxx1xxxx b	not used	
			xx1xxxxx b	not used	
			x1xxxxxx b	LV_PUC	
			1xxxxxxx b	LV_TPS_AD_REQ	
			1xxxxxxx b		
Byte	**Name**	**Size**	**Conversion (hex/bin)**	**Conversion (physical)**	**Resol.**
6	Bitfield 3 STATE_CRU_OFF_IRR (low_byte)	1	xxxxxxx1 b	Limit dynamic 1	
			xxxxxx1x b	Limit dynamic 2	
			xxxxx1xx b	Error clutch switch	
			xxxx1xxx b	MSW error	
			xxx1xxxx b	Timeout ETCU 1	
			xx1xxxxx b	Brake error	
			x1xxxxxx b	not used	
			1xxxxxxx b	not used	
Byte	**Name**	**Size**	**Conversion (hex/bin)**	**Conversion (physical)**	**Resol.**
7	Bitfield 4 STATE_CRU_OFF_IRR (high_byte)	1	xxxxxxx1 b	Monitoring level 2	
			xxxxxx1x b	VS plausibility	
			xxxxx1xx b	ISA limp home aktive	
			xxxx1xxx b	TPS limp home	
			xxx1xxxx b	Error VS	
			xx1xxxxx b	Time out ASR	
			x1xxxxxx b	ETC limp home	
			1xxxxxxx b	not used	
Byte	**Name**	**Size**	**Conversion (hex/bin)**	**Conversion (physical)**	**Resol.**
8	Bitfield 5 STATE_CRU_OFF_REV (low_byte)	1	xxxxxxx1 b	To high VS deviation	
			xxxxxx1x b	Overtaking funct. To long or to high VS_DIF	
			xxxxx1xx b	VS_SP_MAX to long	
			xxxx1xxx b	External TQ intervention	
			xxx1xxxx b	MSW request “OFF”	
			xx1xxxxx b	not used	
			x1xxxxxx b	not used	
			1xxxxxxx b	not used	
9	Bitfield 6 STATE_CRU_OFF_REV (high_byte)	1	xxxxxxx1 b	Error VS_CAN to long	
			xxxxxx1x b	Acceleration monitoring	
			xxxxx1xx b	Run up lock	
			xxxx1xxx b	Overtaking and VS_MAX active	
			xxx1xxxx b	VS_FIL to low	
			xx1xxxxx b	Engine speed limitation	
			x1xxxxxx b	Brake detection	
			1xxxxxxx b	MSW request HARD_OFF	

7.29.40 22_4008 - RDBCI_DK_lesen

General information:

Read Throttle and MAF-Adjustment Value

7.29.40.1 MSD 85 only

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_DKHF
3	RecordCommonIdentifier Low Byte	M	08h	RCI_DKHF

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_DKHF
3	RecordCommonIdentifier Low Byte	M	08h	RCI_DKHF
4	RecordValue 1 = EISYEV_KOROFF_B_I_[0]	M	XXh	RV_EISYEV_KOROFF_B_I_[0]
5	RecordValue 2 = EISYEV_KOROFF_B_I_[1]	M	XXh	RV_EISYEV_KOROFF_B_I_[1]
6	RecordValue 3 = EISYEV_KORFAC_B_I_[0]	M	XXh	RV_EISYEV_KORFAC_B_I_[0]
7	RecordValue 4 = EISYEV_KORFAC_B_I_[1]	M	XXh	RV_EISYEV_KORFAC_B_I_[1]
8	RecordValue 5 = EISYDK_KOROFF_B_I_[0]	M	XXh	RV_EISYDK_KOROFF_B_I_[0]
9	RecordValue 6 = EISYDK_KOROFF_B_I_[1]	M	XXh	RV_EISYDK_KOROFF_B_I_[1]
10	RecordValue 7 = EISYDK_KORFAC_B_I_[0]	M	XXh	RV_EISYDK_KORFAC_B_I_[0]
11	RecordValue 8 = EISYDK_KORFAC_B_I_[1]	M	XXh	RV_EISYDK_KORFAC_B_I_[1]

7.29.40.2 All other projects

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	40h	RCI_DKHF
3	RecordCommonIdentifier= Read Throttle and MAF- Adjustment Value	M	08h	RCI_DKHF

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	40h	RCI_
3	RecordCommonIdentifier= Read Throttle and MAF- Adjustment Value	M	08h	RCI_
4	RecordValue 1 = dummy value fix	M	00h	RV_
5	RecordValue 2 = Eisyev_koroff_b	M	XXh	RV_
6	RecordValue 3 = dummy value fix	M	00h	RV_
7	RecordValue 4 = Eisyev_korfak_b	M	XXh	RV_
8	RecordValue 5 = dummy value fix	M	00h	RV_
9	RecordValue 6 = Eisydk_koroff_b	M	XXh	RV_
10	RecordValue 5 = dummy value fix	M	00h	RV_
11	RecordValue 6 = Eisydk_korfak_b	M	XXh	RV_

7.29.41 22_400A - RCI_GAW_lesen

General information:

Read mixture values 'STATUS_ADAPTION_GEMISCH'
This function is created to read the actual mixture values of the engine.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_GAW_lesen
3	RecordCommonIdentifier Low Byte	M	0Ah	RCI_GAW_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_GAW_lesen
3	RecordCommonIdentifier Low Byte	M	0Ah	RCI_GAW_lesen
4	RecordValue 1 = MFF_ADD_LAM_AD_OUT[1] high Byte	M	XXh	RV_MFF_ADD_LAM_AD_OUT[1]
5	RecordValue 2 = MFF_ADD_LAM_AD_OUT[1] low Byte	M	XXh	RV_MFF_ADD_LAM_AD_OUT[1]
6	RecordValue 3 = MFF_ADD_LAM_AD_OUT[2] high Byte	M	XXh	RV_MFF_ADD_LAM_AD_OUT[2]
7	RecordValue 4 = MFF_ADD_LAM_AD_OUT[2] low Byte	M	XXh	RV_MFF_ADD_LAM_AD_OUT[2]
8	RecordValue 5 = FAC_LAM_AD_CUS[1] high Byte	M	XXh	RV_ FAC_LAM_AD_CUS[1]
9	RecordValue 6 = FAC_LAM_AD_CUS[1] low Byte	M	XXh	RV_ FAC_LAM_AD_CUS[1]
10	RecordValue 7 = FAC_LAM_AD_CUS[2] high Byte	M	XXh	RV_ FAC_LAM_AD_CUS[2]
11	RecordValue 8 = FAC_LAM_AD_CUS[2] low Byte	M	XXh	RV_ FAC_LAM_AD_CUS[2]
12	RecordValue 9 = LSHPWM_UP[1]	M	XXh	RV_LSHPWM_UP[1]
13	RecordValue 10 = LSHPWM_UP[2]	M	XXh	RV_LSHPWM_UP[2]
14	RecordValue 11 = LSHPWM_DOWN[1]	M	XXh	RV_LSHPWM_DOWN[1]
15	RecordValue 12 = LSHPWM_DOWN[2]	M	XXh	RV_LSHPWM_DOWN[2]

7.29.42 22_400B - VVT Messwerte auslesen

General information:

This function is created to read out several values of the VVL- system

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_VVT
3	RecordCommonIdentifier Low Byte	M	0Bh	RCI_VVT

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_VVT
3	RecordCommonIdentifier Low Byte	M	0Bh	RCI_VVT
4	recordValue = Vvt_soll high byte	M	XXh	RV_SOLLW_VVT
5	recordValue = Vvt_soll low byte	M	XXh	RV_SOLLW_VVT
6	recordValue = Exwink_ist high byte	M	XXh	RV_ISTW_VVT

7	recordValue = Exwink_ist low byte	M	XXh	RV_ISTW_VVT
8	recordValue = Ivvtmot[2] high byte	M	XXh	RV_I_VVT
9	recordValue = Ivvtmot[2] low byte	M	XXh	RV_I_VVT
10	recordValue = VCC_DR_VVL high byte	M	XXh	RV_U_VVT
11	recordValue = VCC_DR_VVL low byte	M	XXh	RV_U_VVT
12	recordValue = ANG_1_RAW_VVL[2] high byte	M	XXh	RV_SRF_VVT
13	recordValue = ANG_1_RAW_VVL[2] low byte	M	XXh	RV_SRF_VVT
14	recordValue = St_vvt_err	M	XXh	RV_NOTL_VVT
15	recordValue = St_dvovrld	M	XXh	RV_SUEL_VVT
16	recordValue = ANG_DE_ABSV_PLAUS_CHK_VVL high byte	M	XXh	RV_DIFF_VVT
17	recordValue = ANG_DE_ABSV_PLAUS_CHK_VVL low byte	M	XXh	RV_DIFF_VVT

7.29.43 22_4015 - RCI_FASTA10_lesen

General information:

FASTA_Messwerteblock_10_lesen

This function ist for reading the data of the FASTA (Fahrzeugnutzungsprofil).

7.29.43.1 MSD 85 only

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_FAS10
3	RecordCommonIdentifier Low Byte	M	15h	RCI_FAS10

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_FAS10
3	RecordCommonIdentifier Low Byte	M	15h	RCI_FAS10
4	RecordValue1 = Bszsi (high byte, high word)	M	XXh	RV_BSZS1
5	RecordValue2 = Bszsi (low byte, high word)	M	XXh	RV_BSZS1
6	RecordValue3 = Bszsi (high byte, low word)	M	XXh	RV_BSZS1
7	RecordValue4 = Bszsi (low byte, low word)	M	XXh	RV_BSZS1
	n/Md-Kennfeld			
8	RecordValue5 = STATE_TBL_DRIV[0][0]	M	XXh	nmdsfnp
9	RecordValue6 = STATE_TBL_DRIV[1][0]	M	XXh	nmdsfnp
10	RecordValue7 = STATE_TBL_DRIV[2][0]	M	XXh	nmdsfnp
11	RecordValue8 = STATE_TBL_DRIV[3][0]	M	XXh	nmdsfnp
12	RecordValue9 = STATE_TBL_DRIV[4][0]	M	XXh	nmdsfnp
13	RecordValue10= STATE_TBL_DRIV[5][0]	M	XXh	nmdsfnp
14	RecordValue11 = STATE_TBL_DRIV[6][0]	M	XXh	nmdsfnp
15	RecordValue12 = STATE_TBL_DRIV[7][0]	M	XXh	nmdsfnp
16	RecordValue13 = STATE_TBL_DRIV[0][1]	M	XXh	nmdsfnp
17	RecordValue14 = STATE_TBL_DRIV[1][1]	M	XXh	nmdsfnp
18	RecordValue15 = STATE_TBL_DRIV[2][1]	M	XXh	nmdsfnp
19	RecordValue16 = STATE_TBL_DRIV[3][1]	M	XXh	nmdsfnp
20	RecordValue17 = STATE_TBL_DRIV[4][1]	M	XXh	nmdsfnp
21	RecordValue18 = STATE_TBL_DRIV[5][1]	M	XXh	nmdsfnp
22	RecordValue19 = STATE_TBL_DRIV[6][1]	M	XXh	nmdsfnp

23	RecordValue20 = STATE_TBL_DRIV[7][1]	M	XXh	nmdsfnp
24	RecordValue21 = STATE_TBL_DRIV[0][2]	M	XXh	nmdsfnp
25	RecordValue22 = STATE_TBL_DRIV[1][2]	M	XXh	nmdsfnp
26	RecordValue23 = STATE_TBL_DRIV[2][2]	M	XXh	nmdsfnp
27	RecordValue24 = STATE_TBL_DRIV[3][2]	M	XXh	nmdsfnp
28	RecordValue25 = STATE_TBL_DRIV[4][2]	M	XXh	nmdsfnp
29	RecordValue26 = STATE_TBL_DRIV[5][2]	M	XXh	nmdsfnp
30	RecordValue27 = STATE_TBL_DRIV[6][2]	M	XXh	nmdsfnp
31	RecordValue28 = STATE_TBL_DRIV[7][2]	M	XXh	nmdsfnp
32	RecordValue29 = STATE_TBL_DRIV[0][3]	M	XXh	nmdsfnp
33	RecordValue30 = STATE_TBL_DRIV[1][3]	M	XXh	nmdsfnp
34	RecordValue31 = STATE_TBL_DRIV[2][3]	M	XXh	nmdsfnp
35	RecordValue32 = STATE_TBL_DRIV[3][3]	M	XXh	nmdsfnp
36	RecordValue33 = STATE_TBL_DRIV[4][3]	M	XXh	nmdsfnp
37	RecordValue34 = STATE_TBL_DRIV[5][3]	M	XXh	nmdsfnp
38	RecordValue35 = STATE_TBL_DRIV[6][3]	M	XXh	nmdsfnp
39	RecordValue36 = STATE_TBL_DRIV[7][3]	M	XXh	nmdsfnp
40	RecordValue37 = STATE_TBL_DRIV[0][4]	M	XXh	nmdsfnp
41	RecordValue38 = STATE_TBL_DRIV[1][4]	M	XXh	nmdsfnp
42	RecordValue39 = STATE_TBL_DRIV[2][4]	M	XXh	nmdsfnp
43	RecordValue40 = STATE_TBL_DRIV[3][4]	M	XXh	nmdsfnp
44	RecordValue41 = STATE_TBL_DRIV[4][4]	M	XXh	nmdsfnp
45	RecordValue42 = STATE_TBL_DRIV[5][4]	M	XXh	nmdsfnp
46	RecordValue43 = STATE_TBL_DRIV[6][4]	M	XXh	nmdsfnp
47	RecordValue44 = STATE_TBL_DRIV[7][4]	M	XXh	nmdsfnp
48	RecordValue45 = STATE_TBL_DRIV[0][5]	M	XXh	nmdsfnp
49	RecordValue46 = STATE_TBL_DRIV[1][5]	M	XXh	nmdsfnp
50	RecordValue47 = STATE_TBL_DRIV[2][5]	M	XXh	nmdsfnp
51	RecordValue48 = STATE_TBL_DRIV[3][5]	M	XXh	nmdsfnp
52	RecordValue49 = STATE_TBL_DRIV[4][5]	M	XXh	nmdsfnp
53	RecordValue50 = STATE_TBL_DRIV[5][5]	M	XXh	nmdsfnp
54	RecordValue51 = STATE_TBL_DRIV[6][5]	M	XXh	nmdsfnp
55	RecordValue52 = STATE_TBL_DRIV[7][5]	M	XXh	nmdsfnp
	Generatorauslastungsprofil			
56	RecordValue53 = Dfds[0]	M	XXh	dfdsprofle
57	RecordValue54 = Dfds[1]	M	XXh	dfdsprofle
58	RecordValue55 = Dfds[2]	M	XXh	dfdsprofle
59	RecordValue56 = Dfds[3]	M	XXh	dfdsprofle
60	RecordValue57 = Dfds[4]	M	XXh	dfdsprofle
61	RecordValue58= Dfds[5]	M	XXh	dfdsprofle
62	RecordValue59 = Dfds[6]	M	XXh	dfdsprofle
63	RecordValue60 = Dfds[7]	M	XXh	dfdsprofle
64	RecordValue61 = Dfds[8]	M	XXh	dfdsprofle
65	RecordValue62 = Dfds[9]	M	XXh	dfdsprofle
66	RecordValue63 = Dfds[10]	M	XXh	dfdsprofle
67	RecordValue64 = Dfds[11]	M	XXh	dfdsprofle
68	RecordValue65 = Dfds[12]	M	XXh	dfdsprofle
69	RecordValue66 = Dfds[13]	M	XXh	dfdsprofle
70	RecordValue67 = Dfds[14]	M	XXh	dfdsprofle
71	RecordValue68 = Dfds[15]	M	XXh	dfdsprofle
	Energieaufnahme Bordnetz			
72	RecordValue69 = Igenk (high byte, high word)	M	XXh	RV_IGENK
73	RecordValue70 = Igenk (low byte, high word)	M	XXh	RV_IGENK

74	RecordValue71 = Igenk (high byte, low word)	M	XXh	RV_IGENK
75	RecordValue72 = Igenk (low byte, low word)	M	XXh	RV_IGENK
	Statistik Kühlsystemtemperaturen			
76	RecordValue73 = Tmot_b1 (high byte)	M	XXh	TSTAT
77	RecordValue73 = Tmot_b1 (low byte)	M	XXh	TSTAT
78	RecordValue74 = Tmot_b2 (high byte)	M	XXh	TSTAT
79	RecordValue73 = Tmot_b2 (low byte)	M	XXh	TSTAT
80	RecordValue75 = Tmot_b3 (high byte)	M	XXh	TSTAT
81	RecordValue73 = Tmot_b3 (low byte)	M	XXh	TSTAT
82	RecordValue76 = Tmot_b4 (high byte)	M	XXh	TSTAT
83	RecordValue73 = Tmot_b4 (low byte)	M	XXh	TSTAT
84	RecordValue77 = Tmot_b5 (high byte)	M	XXh	TSTAT
85	RecordValue73 = Tmot_b5 (low byte)	M	XXh	TSTAT
86	RecordValue78 = Toel_b1 (high byte)	M	XXh	TSTAT
87	RecordValue78 = Toel_b1 (low byte)	M	XXh	TSTAT
88	RecordValue79 = Toel_b2 (high byte)	M	XXh	TSTAT
89	RecordValue78 = Toel_b2 (low byte)	M	XXh	TSTAT
90	RecordValue80 = Toel_b3 (high byte)	M	XXh	TSTAT
91	RecordValue78 = Toel_b3 (low byte)	M	XXh	TSTAT
92	RecordValue81 = Toel_b4 (high byte)	M	XXh	TSTAT
93	RecordValue78 = Toel_b4 (low byte)	M	XXh	TSTAT
94	RecordValue82 = Toel_b5 (high byte)	M	XXh	TSTAT
95	RecordValue78 = Toel_b5 (low byte)	M	XXh	TSTAT

7.29.43.2 All other projects

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReaataByCommonIentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	40h	RCI_FAS10
3	RecordCommonIdentifier	M	15h	RCI_FAS10

Table ReadDataByCommonIdentifier Request Message

Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	40h	RCI_FAS10
3	RecordCommonIdentifier	M	15h	RCI_FAS10
4	RecordValue1 = Bszsi (high byte, high word)	M	XXh	RV_BSZSI
5	RecordValue2 = Bszsi (low byte, high word)	M	XXh	RV_BSZSI
6	RecordValue3 = Bszsi (high byte, low word)	M	XXh	RV_BSZSI
7	RecordValue4 = Bszsi (low byte, low word)	M	XXh	RV_BSZSI
	n/Md-Kennfeld			
8	RecordValue5 = STATE_TBL_DRIV[0][0]	M	XXh	RV_NMDS
9	RecordValue6 = STATE_TBL_DRIV[1][0]	M	XXh	RV_NMDS
10	RecordValue7 = STATE_TBL_DRIV[2][0]	M	XXh	RV_NMDS
11	RecordValue8 = STATE_TBL_DRIV[3][0]	M	XXh	RV_NMDS
12	RecordValue9 = STATE_TBL_DRIV[4][0]	M	XXh	RV_NMDS
13	RecordValue10= STATE_TBL_DRIV[5][0]	M	XXh	RV_NMDS
14	RecordValue11 = STATE_TBL_DRIV[6][0]	M	XXh	RV_NMDS

15	RecordValue12 = STATE_TBL_DRIV[7][0]	M	XXh	RV_NMDS
16	RecordValue13 = STATE_TBL_DRIV[0][1]	M	XXh	RV_NMDS
17	RecordValue14 = STATE_TBL_DRIV[1][1]	M	XXh	RV_NMDS
18	RecordValue15 = STATE_TBL_DRIV[2][1]	M	XXh	RV_NMDS
19	RecordValue16 = STATE_TBL_DRIV[3][1]	M	XXh	RV_NMDS
20	RecordValue17 = STATE_TBL_DRIV[4][1]	M	XXh	RV_NMDS
21	RecordValue18 = STATE_TBL_DRIV[5][1]	M	XXh	RV_NMDS
22	RecordValue19 = STATE_TBL_DRIV[6][1]	M	XXh	RV_NMDS
23	RecordValue20 = STATE_TBL_DRIV[7][1]	M	XXh	RV_NMDS
24	RecordValue21 = STATE_TBL_DRIV[0][2]	M	XXh	RV_NMDS
25	RecordValue22 = STATE_TBL_DRIV[1][2]	M	XXh	RV_NMDS
26	RecordValue23 = STATE_TBL_DRIV[2][2]	M	XXh	RV_NMDS
27	RecordValue24 = STATE_TBL_DRIV[3][2]	M	XXh	RV_NMDS
28	RecordValue25 = STATE_TBL_DRIV[4][2]	M	XXh	RV_NMDS
29	RecordValue26 = STATE_TBL_DRIV[5][2]	M	XXh	RV_NMDS
30	RecordValue27 = STATE_TBL_DRIV[6][2]	M	XXh	RV_NMDS
31	RecordValue28 = STATE_TBL_DRIV[7][2]	M	XXh	RV_NMDS
32	RecordValue29 = STATE_TBL_DRIV[0][3]	M	XXh	RV_NMDS
33	RecordValue30 = STATE_TBL_DRIV[1][3]	M	XXh	RV_NMDS
34	RecordValue31 = STATE_TBL_DRIV[2][3]	M	XXh	RV_NMDS
35	RecordValue32 = STATE_TBL_DRIV[3][3]	M	XXh	RV_NMDS
36	RecordValue33 = STATE_TBL_DRIV[4][3]	M	XXh	RV_NMDS
37	RecordValue34 = STATE_TBL_DRIV[5][3]	M	XXh	RV_NMDS
38	RecordValue35 = STATE_TBL_DRIV[6][3]	M	XXh	RV_NMDS
39	RecordValue36 = STATE_TBL_DRIV[7][3]	M	XXh	RV_NMDS
40	RecordValue37 = STATE_TBL_DRIV[0][4]	M	XXh	RV_NMDS
41	RecordValue38 = STATE_TBL_DRIV[1][4]	M	XXh	RV_NMDS
42	RecordValue39 = STATE_TBL_DRIV[2][4]	M	XXh	RV_NMDS
43	RecordValue40 = STATE_TBL_DRIV[3][4]	M	XXh	RV_NMDS
44	RecordValue41 = STATE_TBL_DRIV[4][4]	M	XXh	RV_NMDS
45	RecordValue42 = STATE_TBL_DRIV[5][4]	M	XXh	RV_NMDS
46	RecordValue43 = STATE_TBL_DRIV[6][4]	M	XXh	RV_NMDS
47	RecordValue44 = STATE_TBL_DRIV[7][4]	M	XXh	RV_NMDS
48	RecordValue45 = STATE_TBL_DRIV[0][5]	M	XXh	RV_NMDS
49	RecordValue46 = STATE_TBL_DRIV[1][5]	M	XXh	RV_NMDS
50	RecordValue47 = STATE_TBL_DRIV[2][5]	M	XXh	RV_NMDS
51	RecordValue48 = STATE_TBL_DRIV[3][5]	M	XXh	RV_NMDS
52	RecordValue49 = STATE_TBL_DRIV[4][5]	M	XXh	RV_NMDS
53	RecordValue50 = STATE_TBL_DRIV[5][5]	M	XXh	RV_NMDS
54	RecordValue51 = STATE_TBL_DRIV[6][5]	M	XXh	RV_NMDS
55	RecordValue52 = STATE_TBL_DRIV[7][5]	M	XXh	RV_NMDS
	Generatorauslastungsprofil			
56	RecordValue53 = Dfds[0]	M	XXh	RV_DFDS
57	RecordValue54 = Dfds[1]	M	XXh	RV_DFDS
58	RecordValue55 = Dfds[2]	M	XXh	RV_DFDS
59	RecordValue56 = Dfds[3]	M	XXh	RV_DFDS
60	RecordValue57 = Dfds[4]	M	XXh	RV_DFDS
61	RecordValue58= Dfds[5]	M	XXh	RV_DFDS
62	RecordValue59 = Dfds[6]	M	XXh	RV_DFDS
63	RecordValue60 = Dfds[7]	M	XXh	RV_DFDS
64	RecordValue61 = Dfds[8]	M	XXh	RV_DFDS
65	RecordValue62 = Dfds[9]	M	XXh	RV_DFDS
66	RecordValue63 = Dfds[10]	M	XXh	RV_DFDS

67	RecordValue64 = Dfds[11]	M	XXh	RV_DFDS
68	RecordValue65 = Dfds[12]	M	XXh	RV_DFDS
69	RecordValue66 = Dfds[13]	M	XXh	RV_DFDS
70	RecordValue67 = Dfds[14]	M	XXh	RV_DFDS
71	RecordValue68 = Dfds[15]	M	XXh	RV_DFDS
	Energieaufnahme Bordnetz			
72	RecordValue69 = Igenk (high byte, high word)	M	XXh	RV_IGENK
73	RecordValue70 = Igenk (low byte, high word)	M	XXh	RV_IGENK
74	RecordValue71 = Igenk (high byte, low word)	M	XXh	RV_IGENK
75	RecordValue72 = Igenk (low byte, low word)	M	XXh	RV_IGENK
	Statistik Kühlsystemtemperaturen			
76	RecordValue73 = Tmot_b1 (high byte)	M	XXh	RV_TMOTB1
77	RecordValue73 = Tmot_b1 (low byte)	M	XXh	RV_TMOTB1
78	RecordValue74 = Tmot_b2 (high byte)	M	XXh	RV_TMOTB2
79	RecordValue73 = Tmot_b2 (low byte)	M	XXh	RV_TMOTB2
80	RecordValue75 = Tmot_b3 (high byte)	M	XXh	RV_TMOTB3
81	RecordValue73 = Tmot_b3 (low byte)	M	XXh	RV_TMOTB3
82	RecordValue76 = Tmot_b4 (high byte)	M	XXh	RV_TMOTB4
83	RecordValue73 = Tmot_b4 (low byte)	M	XXh	RV_TMOTB4
84	RecordValue77 = Tmot_b5 (high byte)	M	XXh	RV_TMOTB5
85	RecordValue73 = Tmot_b5 (low byte)	M	XXh	RV_TMOTB5
86	RecordValue78 = Toel_b1 (high byte)	M	XXh	RV_TOELB1
87	RecordValue78 = Toel_b1 (low byte)	M	XXh	RV_TOELB1
88	RecordValue79 = Toel_b2 (high byte)	M	XXh	RV_TOELB2
89	RecordValue78 = Toel_b2 (low byte)	M	XXh	RV_TOELB2
90	RecordValue80 = Toel_b3 (high byte)	M	XXh	RV_TOELB3
91	RecordValue78 = Toel_b3 (low byte)	M	XXh	RV_TOELB3
92	RecordValue81 = Toel_b4 (high byte)	M	XXh	RV_TOELB4
93	RecordValue78 = Toel_b4 (low byte)	M	XXh	RV_TOELB4
94	RecordValue82 = Toel_b5 (high byte)	M	XXh	RV_TOELB5
95	RecordValue78 = Toel_b5 (low byte)	M	XXh	RV_TOELB5
96	RecordValue83 = Tget_b1 (high byte)	M	XXh	RV_TGETB1
97	RecordValue83 = Tget_b1 (low byte)	M	XXh	RV_TGETB1
98	RecordValue84 = Tget_b2 (high byte)	M	XXh	RV_TGETB2
99	RecordValue83 = Tget_b2 (low byte)	M	XXh	RV_TGETB2
100	RecordValue85 = Tget_b3 (high byte)	M	XXh	RV_TGETB3
101	RecordValue83 = Tget_b3 (low byte)	M	XXh	RV_TGETB3
102	RecordValue86 = Tget_b4 (high byte)	M	XXh	RV_TGETB4
103	RecordValue83 = Tget_b4 (low byte)	M	XXh	RV_TGETB4
104	RecordValue87 = Tget_b5 (high byte)	M	XXh	RV_TGETB5
105	RecordValue83 = Tget_b5 (low byte)	M	XXh	RV_TGETB5
106	RecordValue88 = Tumg_b1 (high byte)	M	XXh	RV_TUMGB1
107	RecordValue88 = Tumg_b1 (low byte)	M	XXh	RV_TUMGB1
108	RecordValue89 = Tumg_b2 (high byte)	M	XXh	RV_TUMGB2
109	RecordValue88 = Tumg_b2 (low byte)	M	XXh	RV_TUMGB2
110	RecordValue90 = Tumg_b3 (high byte)	M	XXh	RV_TUMGB3
111	RecordValue88 = Tumg_b3 (low byte)	M	XXh	RV_TUMGB3
112	RecordValue91 = Tumg_b4 (high byte)	M	XXh	RV_TUMGB4
113	RecordValue88 = Tumg_b4 (low byte)	M	XXh	RV_TUMGB4
114	RecordValue92 = Tumg_b5 (high byte)	M	XXh	RV_TUMGB5
115	RecordValue88 = Tumg_b5 (low byte)	M	XXh	RV_TUMGB5

7.29.44 22_4016 - RCI_IGRINFO_lesen

General information:

READ IGR-values

This function ist for reading the data of the IGR (Infospeicher Intelligente Genrator Regelung).

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_IGRINFO
3	RecordCommonIdentifier Low Byte	M	16h	RCI_IGRINFO

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_IGRINFO
3	RecordCommonIdentifier Low Byte	M	16h	RCI_IGRINFO
4	RecordValue1 = Igrinfo[1]	M	XXh	RV_IGRINFO
5	RecordValue2 = Igrinfo[2]	M	XXh	RV_IGRINFO
6	RecordValue3 = Igrinfo[3]	M	XXh	RV_IGRINFO
7	RecordValue4 = Igrinfo[4]	M	XXh	RV_IGRINFO
8	RecordValue5 = Igrinfo[5]	M	XXh	RV_IGRINFO
9	RecordValue6 = Igrinfo[6]	M	XXh	RV_IGRINFO
10	RecordValue7 = Igrinfo[7]	M	XXh	RV_IGRINFO
11	RecordValue8 = Igrinfo[8]	M	XXh	RV_IGRINFO
12	RecordValue9 = Igrinfo[9]	M	XXh	RV_IGRINFO
13	RecordValue10 = Igrinfo[10]	M	XXh	RV_IGRINFO
14	RecordValue11 = Igrinfo[11]	M	XXh	RV_IGRINFO
15	RecordValue12 = Igrinfo[12]	M	XXh	RV_IGRINFO
16	RecordValue13 = Igrinfo[13]	M	XXh	RV_IGRINFO
17	RecordValue14 = Igrinfo[14]	M	XXh	RV_IGRINFO
18	RecordValue15 = Igrinfo[15]	M	XXh	RV_IGRINFO
19	RecordValue16 = Igrinfo[16]	M	XXh	RV_IGRINFO
20	RecordValue17 = Igrinfo[17]	M	XXh	RV_IGRINFO
21	RecordValue18 = Igrinfo[18]	M	XXh	RV_IGRINFO
22	RecordValue19 = Igrinfo[19]	M	XXh	RV_IGRINFO
23	RecordValue20 = Igrinfo[20]	M	XXh	RV_IGRINFO
24	RecordValue21 = Igrinfo[21]	M	XXh	RV_IGRINFO
25	RecordValue22 = Igrinfo[22]	M	XXh	RV_IGRINFO
26	RecordValue23 = Igrinfo[23]	M	XXh	RV_IGRINFO
27	RecordValue24 = Igrinfo[24]	M	XXh	RV_IGRINFO
28	RecordValue25 = Igrinfo[25]	M	XXh	RV_IGRINFO
29	RecordValue26 = Igrinfo[26]	M	XXh	RV_IGRINFO
30	RecordValue27 = Igrinfo[27]	M	XXh	RV_IGRINFO
31	RecordValue28 = Igrinfo[28]	M	XXh	RV_IGRINFO
32	RecordValue29 = Igrinfo[29]	M	XXh	RV_IGRINFO
33	RecordValue30 = Igrinfo[30]	M	XXh	RV_IGRINFO

7.29.45 22_4017 - RCI_LEMINFO_lesen

General information:

READ LEM-values

This function ist for reading the data of the LEM (Infospeicher Leistungskoordination Elektrisch Mechanisch).

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_LEMINFO
3	RecordCommonIdentifier Low Byte	M	17h	RCI_LEMINFO

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_LEMINFO
3	RecordCommonIdentifier Low Byte	M	17h	RCI_LEMINFO
4	RecordValue1 = Leminfo[1]	M	XXh	RV_LEMINFO
5	RecordValue2 = Leminfo[2]	M	XXh	RV_LEMINFO
6	RecordValue3 = Leminfo[3]	M	XXh	RV_LEMINFO
7	RecordValue4 = Leminfo[4]	M	XXh	RV_LEMINFO
8	RecordValue5 = Leminfo[5]	M	XXh	RV_LEMINFO
9	RecordValue6 = Leminfo[6]	M	XXh	RV_LEMINFO
10	RecordValue7 = Leminfo[7]	M	XXh	RV_LEMINFO
11	RecordValue8 = Leminfo[8]	M	XXh	RV_LEMINFO
12	RecordValue9 = Leminfo[9]	M	XXh	RV_LEMINFO
13	RecordValue10 = Leminfo[10]	M	XXh	RV_LEMINFO
14	RecordValue11 = Leminfo[11]	M	XXh	RV_LEMINFO
15	RecordValue12 = Leminfo[12]	M	XXh	RV_LEMINFO
16	RecordValue13 = Leminfo[13]	M	XXh	RV_LEMINFO
17	RecordValue14 = Leminfo[14]	M	XXh	RV_LEMINFO
18	RecordValue15 = Leminfo[15]	M	XXh	RV_LEMINFO
19	RecordValue16 = Leminfo[16]	M	XXh	RV_LEMINFO
20	RecordValue17 = Leminfo[17]	M	XXh	RV_LEMINFO
21	RecordValue18 = Leminfo[18]	M	XXh	RV_LEMINFO
22	RecordValue19 = Leminfo[19]	M	XXh	RV_LEMINFO
23	RecordValue20 = Leminfo[20]	M	XXh	RV_LEMINFO
24	RecordValue21 = Leminfo[21]	M	XXh	RV_LEMINFO
25	RecordValue22 = Leminfo[22]	M	XXh	RV_LEMINFO
26	RecordValue23 = Leminfo[23]	M	XXh	RV_LEMINFO
27	RecordValue24 = Leminfo[24]	M	XXh	RV_LEMINFO
28	RecordValue25 = Leminfo[25]	M	XXh	RV_LEMINFO
29	RecordValue26 = Leminfo[26]	M	XXh	RV_LEMINFO
30	RecordValue27 = Leminfo[27]	M	XXh	RV_LEMINFO
31	RecordValue28 = Leminfo[28]	M	XXh	RV_LEMINFO
32	RecordValue29 = Leminfo[29]	M	XXh	RV_LEMINFO
33	RecordValue30 = Leminfo[30]	M	XXh	RV_LEMINFO
34	RecordValue31 = Leminfo[31]	M	XXh	RV_LEMINFO
35	RecordValue32 = Leminfo[32]	M	XXh	RV_LEMINFO
36	RecordValue33 = Leminfo[33]	M	XXh	RV_LEMINFO

37	RecordValue34 = Leminfo[34]	M	XXh	RV_LEMINFO
38	RecordValue35 = Leminfo[35]	M	XXh	RV_LEMINFO
39	RecordValue36 = Leminfo[36]	M	XXh	RV_LEMINFO
40	RecordValue37 = Leminfo[37]	M	XXh	RV_LEMINFO

7.29.46 22_4018 - READ MSA-values

General information:

This function ist for reading the data of the MSA (Infospeicher Motor-Start/Stop Automatik).

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MSA
3	RecordCommonIdentifier Low Byte	M	18h	RCI_MSA

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MSA
3	RecordCommonIdentifier Low Byte	M	18h	RCI_MSA
4	RecordValue1 = Msainfo[1]	M	XXh	RV_
5	RecordValue2 = Msainfo[2]	M	XXh	RV_
6	RecordValue3 = Msainfo[3]	M	XXh	RV_
7	RecordValue4 = Msainfo[4]	M	XXh	RV_
8	RecordValue5 = Msainfo[5]	M	XXh	RV_
9	RecordValue6 = Msainfo[6]	M	XXh	RV_
10	RecordValue7 = Msainfo[7]	M	XXh	RV_
11	RecordValue8 = Msainfo[8]	M	XXh	RV_
12	RecordValue9 = Msainfo[9]	M	XXh	RV_
13	RecordValue10 = Msainfo[10]	M	XXh	RV_
14	RecordValue11 = Msainfo[11]	M	XXh	RV_
15	RecordValue12 = Msainfo[12]	M	XXh	RV_
16	RecordValue13 = Msainfo[13]	M	XXh	RV_
17	RecordValue14 = Msainfo[14]	M	XXh	RV_
18	RecordValue15 = Msainfo[15]	M	XXh	RV_
19	RecordValue16 = Msainfo[16]	M	XXh	RV_
20	RecordValue17 = Msainfo[17]	M	XXh	RV_
21	RecordValue18 = Msainfo[18]	M	XXh	RV_
22	RecordValue19 = Msainfo[19]	M	XXh	RV_
23	RecordValue20 = Msainfo[20]	M	XXh	RV_
24	RecordValue21 = Msainfo[21]	M	XXh	RV_
25	RecordValue22 = Msainfo[22]	M	XXh	RV_
26	RecordValue23 = Msainfo[23]	M	XXh	RV_
27	RecordValue24 = Msainfo[24]	M	XXh	RV_
28	RecordValue25 = Msainfo[25]	M	XXh	RV_
29	RecordValue26 = Msainfo[26]	M	XXh	RV_
30	RecordValue27 = Msainfo[27]	M	XXh	RV_
31	RecordValue28 = Msainfo[28]	M	XXh	RV_
32	RecordValue29 = Msainfo[29]	M	XXh	RV_
33	RecordValue30 = Msainfo[30]	M	XXh	RV_

34	RecordValue31 = Msainfo[31]	M	XXh	RV_
35	RecordValue32 = Msainfo[32]	M	XXh	RV_
36	RecordValue33 = Msainfo[33]	M	XXh	RV_
37	RecordValue34 = Msainfo[34]	M	XXh	RV_
38	RecordValue35 = Msainfo[35]	M	XXh	RV_
39	RecordValue36 = Msainfo[36]	M	XXh	RV_
40	RecordValue37 = Msainfo[37]	M	XXh	RV_
41	RecordValue38 = Msainfo[38]	M	XXh	RV_
42	RecordValue39 = Msainfo[39]	M	XXh	RV_
43	RecordValue40 = Msainfo[40]	M	XXh	RV_
44	RecordValue41 = Msainfo[41]	M	XXh	RV_
45	RecordValue42 = Msainfo[42]	M	XXh	RV_
46	RecordValue43 = Msainfo[43]	M	XXh	RV_
47	RecordValue44 = Msainfo[44]	M	XXh	RV_
48	RecordValue45 = Msainfo[45]	M	XXh	RV_
49	RecordValue46 = Msainfo[46]	M	XXh	RV_
50	RecordValue47 = Msainfo[47]	M	XXh	RV_
51	RecordValue48 = Msainfo[48]	M	XXh	RV_
52	RecordValue49 = Msainfo[49]	M	XXh	RV_
53	RecordValue50 = Msainfo[50]	M	XXh	RV_

7.29.47 22_4019_000 - RDBCI_FSPFR

General information:

read Error Memory with freeze frame data extra long, extended

This function is created to read out the failure memory with all the errors including the freeze frame data. The index of the environment values allows a classification to the list of the environment values from BMW.

Record Common Identifier table usage and reading:

- record values usage as BYTE Values (Freeze Frame, Umweltbedingung, DDLI etc.) should support the full range 0x00 to 0xFF as VALID. A value that is not supported will be transmitted as 0xFF

- record values usage as WORD Values (DDLI etc.) should support the range 0x0000 to 0xFFFE as VALID. Invalid is shown with 0xFFFF.

This should enable the full range of internal Byte values to be used with no compromise, but still allow the diagnostic equipment to recognise if a value is available or not.

Formula section:

IF ERR_DTC[x] = 0000h **or**

ERR_DTC[x] = FFFFh

THEN send **negative respones**

ELSE send **positive response**

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_FSPFR
3	RecordCommonIdentifier Low Byte	M	19h	RCI_FSPFR
4	Value = Error code ERR_DTC[x] high byte	M	XXh	RV_FSFF_F
5	Value = Error code ERR_DTC[x] low byte	M	XXh	RV_FSFF_F

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	**Positive Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_FSPFR
3	RecordCommonIdentifier Low Byte	M	19h	RCI_FSPFR
4	RecordValue 1 = Error counter total CTR_ERR_DYN_NR	M	XXh	RV_FSFF_FZG
5	RecordValue 2 = Error code ERR_DTC[x] high byte	M	XXh	RV_FSFF_FC
6	RecordValue 3 = Error code ERR_DTC[x] low byte	M	XXh	RV_FSFF_FC
7	RecordValue 4 = status DTC1 ERR_TYPE_BYTE[x]	M	XXh	RV_FSFF_FA
8	RecordValue 5 = Cycle flag ERR_TYPE_EXT_BYTE[x]	M	XXh	RV_FSFF_ERW
9	RecordValue 6 = Frequency counter CTR_FRC[x]	M	XXh	RV_FSFF_HZ
10	RecordValue 7 = Error recover counter (HLC) CTR_DC[x]	M	XXh	RV_FSFF_HLC
11	RecordValue 8 = km-value by first occur high byte ENVD_CUS_SET_CMN[2][1]	M	XXh	RV_FSFF_KM1
12	RecordValue 9= km-value by first occur low byte ENVD_CUS_SET_CMN[3][1]	M	XXh	RV_FSFF_KM1
13	RecordValue 10 = 1.environmental value first occur ENVD_CUS_SET_SPC[1][1]	M	XXh	RV_FSFF_UW11
14	RecordValue 11 = Index for 1.environmental value	M	XXh	RV_FSFF_UW11
15	RecordValue 12 = 2.environmental value first occur ENVD_CUS_SET_SPC[2][1]	M	XXh	RV_FSFF_UW21
16	RecordValue 13 = Index for 2.environmental value	M	XXh	RV_FSFF_UW21A
17	RecordValue 14= 3.environmental value first occur ENVD_CUS_SET_SPC[3][1]	M	XXh	RV_FSFF_UW31
18	RecordValue 15 = Index for 3.environmental value	M	XXh	RV_FSFF_UW31A
19	RecordValue 16 = 4.environmental value first occur ENVD_CUS_SET_SPC[4[1]	M	XXh	RV_FSFF_UW41
20	RecordValue 17 = Index for 4.environmental value	M	XXh	RV_FSFF_UW41A
21	RecordValue 18 = km-value by second occur high byte ENVD_CUS_SET_CMN[2][2]	M	XXh	RV_FSFF_KM2
22	RecordValue 19= km-value by second occur low byte ENVD_CUS_SET_CMN[3][2]	M	XXh	RV_FSFF_KM2
23	RecordValue 20 = 1.environmental value second occur ENVD_CUS_SET_SPC[1][2]	M	XXh	RV_FSFF_UW12
24	RecordValue 21 = 2.environmental value second occur ENVD_CUS_SET_SPC[2][2]	M	XXh	RV_FSFF_UW22
25	RecordValue 22 = 3.environmental value second occur ENVD_CUS_SET_SPC[3][2]	M	XXh	RV_FSFF_UW23
26	RecordValue 23 = 4.environmental value second occur ENVD_CUS_SET_SPC[4][2]	M	XXh	RV_FSFF_UW24
27	RecordValue 24 = km-value by last occur high byte ENVD_CUS_SET_CMN[2][3]	M	XXh	RV_FSFF_KMX
28	RecordValue 25 = km-value by last occur low byte ENVD_CUS_SET_CMN[3][3]	M	XXh	RV_FSFF_KMX
29	RecordValue 26 = 1.environmental value last occur ENVD_CUS_SET_SPC[1][3]	M	XXh	RV_FSFF_UW1X

30	RecordValue 27= 2.environmental value last occur ENVD_CUS_SET_SPC[2][3]	M	XXh	RV_FSFF_UW2X
31	RecordValue 28= 3.environmental value last occur ENVD_CUS_SET_SPC[3][3]	M	XXh	RV_FSFF_UW3X
32	RecordValue 29 = 4.environmental value last occur ENVD_CUS_SET_SPC[4][3]	M	XXh	RV_FSFF_UW4X
33	RecordValue 30= Error class C_ERR_CLAS_XX	M	00h	RV_FSFF_CLA
34	RecordValue 31 = Importance counter TSF (not def. yet)	M	FFh	RV_FSFF_TSF
35	RecordValue 32 = Debounce counter (MIL on) FLC FLC = DC_MAX_XX - CTR_DC[x]	M	XXh	RV_FSFF_FLC
36	RecordValue 33= Deleting counter DLC CTR_WUP_CYC[x]	M	XXh	RV_FSFF_DLC
37	RecordValue 34 = P code DTC ” high byte ACTION_ERRM_ReadDTCLevelByDTCLevel	M	XXh	RV_FSFF_PID02
38	RecordValue 35 = P code DTC ” low byte ACTION_ERRM_ReadDTCLevelByDTCLevel	M	XXh	RV_FSFF_PID02
39	RecordValue 36 = Freeze frame data 0 – STATE_LS_1 (fuel system state bench 1) ENVD_OBD[1][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID03
40	RecordValue 37 = Freeze frame data 1 – STATE_LS_2 (fuel system state bench 2) ENVD_OBD[2][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID03
41	RecordValue 38 = Freeze frame data 2 – LOAD_CLC (calculated load value) ENVD_OBD[3][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID04
42	RecordValue 39 = Freeze frame data 3 – OBD_TCO (engine coolant temperature) ENVD_OBD[4][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID05
43	RecordValue 40 = Freeze frame data 4 – OBD_LAM_COR_1 (short term fuel trim bank 1) ENVD_OBD[5][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID06
44	RecordValue 41 = Freeze frame data 5 – OBD_LAM_AD_1 (long term fuel trim bank 1) ENVD_OBD[6][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID07
45	RecordValue 42 = Freeze frame data 6 – OBD_LAM_COR_2 (short term fuel trim bank 2) ENVD_OBD[7][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID08
46	RecordValue 43 = Freeze frame data 7 – OBD_LAM_AD_2 (long term fuel trim bank 2) ENVD_OBD[8][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID09
47	RecordValue 44 = Freeze frame data 8 – OBD_FUP (fuel pressure ENVD_OBD[9][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID0A
48	RecordValue 45 = Freeze frame data 9 – OBD_MAP (intake manifold absolutr pressure) ENVD_OBD[10][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID0B
49	RecordValue 46 = Freeze frame data 10 – OBD_N (engine rpm, high byte) ENVD_OBD[11][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID0C
50	RecordValue 47 = Freeze frame data 11 – OBD_N (engine rpm, low byte) ENVD_OBD[12][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID0C
51	RecordValue 48 = Freeze frame data 12 – VS (vehicle speed) ENVD_OBD[13][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID0D
52	RecordValue 49 = Freeze frame data 13 – OBD_IGA_IGC (ignition timing advance cylinder 1) ENVD_OBD[14][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID0E
53	RecordValue 50 = Freeze frame data 14 – OBD_TIA (intake air temperature) ENVD_OBD[15][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID0F

54	RecordValue 51 = Freeze frame data 15 – OBD_MAF (air flow rate mass air flow sensor, high byte) ENVD_OBD[16][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID10
55	RecordValue 52 = Freeze frame data 16 – OBD_MAF (air flow rate mass air flow sensor, high byte) ENVD_OBD[17][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID10
56	RecordValue 53 = Freeze frame data 17 – OBD_TPS_1 (absolute throttle position) ENVD_OBD[18][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID11
57	RecordValue 54 = Freeze frame data 18 – STATE_OBD_SA (commanded secondary air status) ENVD_OBD[19][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID12
58	RecordValue 55 = Freeze frame data 19 – not supported (auxilary input)	M	FFh	RV_FSFF_PID1E
59	RecordValue 56 = Freeze frame data 20 – OBD_T_AST (time since engine start, high byte) ENVD_OBD[20][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID1F
60	RecordValue 57 = Freeze frame data 21 – OBD_T_AST (time since engine start, low byte) ENVD_OBD[21][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID1F
61	RecordValue 58 = Freeze frame data 22 – not supported (fuel rail pressure relative to manifold vacuum , high byte)	M	FFh	RV_FSFF_PID22
62	RecordValue 59 = Freeze frame data 23 – not supported (fuel rail pressure relative to manifold vacuum , low byte)	M	FFh	RV_FSFF_PID22
63	RecordValue 60 = Freeze frame data 24 – OBD_FUP_RNG_H (fuel rail pressure , high byte) ENVD_OBD[22][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID23
64	RecordValue 61 = Freeze frame data 25 – OBD_FUP_RNG_H (fuel rail pressure , low byte) ENVD_OBD[23][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID23
65	RecordValue 62 = Freeze frame data 26 – not supported (commanded EGR)	M	FFh	RV_FSFF_PID2C
66	RecordValue 63 = Freeze frame data 27 – not supported (EGR error)	M	FFh	RV_FSFF_PID2D
67	RecordValue 64 = Freeze frame data 28 –CPPWM_CPS (commanded evaporative purge) ENVD_OBD[24][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID2E
68	RecordValue 65 = Freeze frame data 29 – OBD_FTL (fuel level input) ENVD_OBD[25][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID2F
69	RecordValue 66 = Freeze frame data 30 – OBD_AMP (barometric pressure) ENVD_OBD[26][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID33
70	RecordValue 67 = Freeze frame data 31 – OBD_VB (control module voltage, high byte) ENVD_OBD[27][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID42
71	RecordValue 68 = Freeze frame data 32 – OBD_VB (control module voltage, low byte) ENVD_OBD[28][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID42
72	RecordValue 69 = Freeze frame data 33 – LOAD_ABSV (absolute load value, high byte) ENVD_OBD[29][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID43
73	RecordValue 70 = Freeze frame data 34 – LOAD_ABSV (absolute load value, low byte) ENVD_OBD[30][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID43
74	RecordValue 71 = Freeze frame data 35 – OBD_LAMB_SP (commanded equivalence ratio, high byte) ENVD_OBD[31][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID44

75	RecordValue 72 = Freeze frame data 36 – OBD_LAMB_SP (commanded equivalence ratio, low byte) ENVD_OBD[32][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID44
76	RecordValue 73 = Freeze frame data 37 – OBD_TPS_REL (relative throttle position) ENVD_OBD[33][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID45
77	RecordValue 74 = Freeze frame data 38 – OBD_TAM (ambiente air temperature) ENVD_OBD[34][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID46
78	RecordValue 75 = Freeze frame data 39 – OBD_TPS_2 (absolute throttle position, sensor 2) ENVD_OBD[35][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID47
79	RecordValue 76 = Freeze frame data 40 – not supported (absolue throttle position C)	M	FFh	RV_FSFF_PID48
80	RecordValue 77 = Freeze frame data 41 – OBD_PV_1 (accelaration pedal position, sensor 1) ENVD_OBD[36][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID49
81	RecordValue 78 = Freeze frame data 42 – OBD_PV_2 (accelaration pedal position, sensor 2) ENVD_OBD[37][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID4A
82	RecordValue 79 = Freeze frame data 43 – not supported (accelaration pedal position F)	M	FFh	RV_FSFF_PID4B
83	RecordValue 80 = Freeze frame data 44 – OBD_TPS_SP (commanded throttle position) ENVD_OBD[38][NC_NR_ERR_DYN]	M	XXh	RV_FSFF_PID4C

7.29.48 22_401A - RCI_BZEINFO_lesen

General information:

READ BZE-values

This function ist for reading the data of the BZE (Batterie Zustands Erkennung)

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_BZEINFO
3	RecordCommonIdentifier Low Byte	M	1A	RCI_BZEINFO

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_BZEINFO
3	RecordCommonIdentifier Low Byte	M	1A	RCI_BZEINFO
4	RecordValue1 = Qv_out_m	M	XXh	RV_QV_OUT_M
5	RecordValue2 = Qv_quali_m	M	XXh	RV_QV_QUALI_M
6	RecordValue3 = Qv_status	M	XXh	RV_QV_STATUS
7	RecordValue4 = Qv_out_1	M	XXh	RV_QV_OUT_1
8	RecordValue5 = Qv_out_2	M	XXh	RV_QV_OUT_2
9	RecordValue6 = Qv_out_3	M	XXh	RV_QV_OUT_3
10	RecordValue7 = Qv_out_4	M	XXh	RV_QV_OUT_4
11	RecordValue8 = Qv_out_5	M	XXh	RV_QV_OUT_5
12	RecordValue9 = Qv_quali_1	M	XXh	RV_QV_QUALI_1
13	RecordValue10 = Qv_quali_2	M	XXh	RV_QV_QUALI_2

14	RecordValue11 = Qv_quali_3	M	XXh	RV_QV_QUALI_3
15	RecordValue12 = Qv_quali_4	M	XXh	RV_QV_QUALI_4
16	RecordValue13 = Qv_quali_5	M	XXh	RV_QV_QUALI_5
17	RecordValue14 = Qv_td1 (high byte)	M	XXh	RV_QV_TD1
18	RecordValue15 = Qv_td1 (low byte)	M	XXh	RV_QV_TD1
19	RecordValue16 = Qv_td2 (high byte)	M	XXh	RV_QV_TD2
20	RecordValue17 = Qv_td2 (low byte)	M	XXh	RV_QV_TD2
21	RecordValue18 = Qv_td3 (high byte)	M	XXh	RV_QV_TD3
22	RecordValue19 = Qv_td3 (low byte)	M	XXh	RV_QV_TD3
23	RecordValue20 = Qv_td4 (high byte)	M	XXh	RV_QV_TD4
24	RecordValue20 = Qv_td4 (low byte)	M	XXh	RV_QV_TD4
25	RecordValue18 = Qv_td5 (high byte)	M	XXh	RV_QV_TD5
26	RecordValue18 = Qv_td5 (low byte)	M	XXh	RV_QV_TD5
27	RecordValue24 = Qvc_status_1	M	XXh	RV_QVC_STATUS_1
28	RecordValue25 = Qvc_status_2	M	XXh	RV_QVC_STATUS_2
29	RecordValue26 = Qvc_status_3	M	XXh	RV_QVC_STATUS_3
30	RecordValue27 = Qvc_status_4	M	XXh	RV_QVC_STATUS_4
31	RecordValue28 = Qv_nv_zh (high byte, high word)	M	XXh	RV_QV_NV_ZH
32	RecordValue29 = Qv_nv_zh (low byte, high word)	M	XXh	RV_QV_NV_ZH
33	RecordValue30 = Qv_nv_zh (ligh byte, low word)	M	XXh	RV_QV_NV_ZH
34	RecordValue31 = Qv_nv_zh (low byte, low word)	M	XXh	RV_QV_NV_ZH
35	RecordValue32 = Qv_nv_ezm (high byte)	M	XXh	RV_QV_NV_EZM
36	RecordValue33 = Qv_nv_ezm (low byte)	M	XXh	RV_QV_NV_EZM
37	RecordValue34 = Qv_h2o (high byte)	M	XXh	RV_QV_H2O
38	RecordValue35 = Qv_h2o (low byte)	M	XXh	RV_QV_H2O
39	RecordValue36 = Qv_h2oquali	M	XXh	RV_QV_H2OQUALI
40	RecordValue37 = B_qvch2o	M	XXh [0, 1]	RV_ST_QVC1
41	RecordValue38 = Qv_h2ostatus	M	XXh	RV_QV_H2OSTATUS

7.29.49 22_401B - RCI_GENINFO_lesen

General information:

Generator Diagnoseerweiterung

This function is created to read out the values from the function-test of intelligent alternator.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_GENINFO
3	RecordCommonIdentifier Low Byte	M	1Bh	RCI_GENINFO

Table Request Message

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_GENINFO
3	RecordCommonIdentifier Low Byte	M	1Bh	RCI_GENINFO

4	recordValue1 = St_dgenub1 (high byte)	M	XXh	RV_ST_DGENUB1
5	recordValue2 = St_dgenub1 (low byte)	M	XXh	RV_ST_DGENUB1
6	recordValue3 = St_dgenub2 (hiigh byte)	M	XXh	RV_ST_DGENUB2
7	recordValue4 = St_dgenub2 (low byte)	M	XXh	RV_ST_DGENUB2
8	recordValue5 = St_dgenubnz (high byte)	M	XXh	RV_ST_DGENUBNZ
9	recordValue6 = St_dgenubnz (low byte)	M	XXh	RV_ST_DGENUBNZ
10	recordValue7 = St_dgenuberr	M	XXh	RV_ST_DGENUBERR
11	recordValue8 = St_dgenugen1 (high byte)	M	XXh	RV_ST_DGENUGEN1
12	recordValue9 = St_dgenugen1 (low byte)	M	XXh	RV_ST_DGENUGEN1
13	recordValue10 = St_dgenugen2 (high byte)	M	XXh	RV_ST_DGENUGEN2
14	recordValue11 = St_dgenugen2 (low byte)	M	XXh	RV_ST_DGENUGEN2
15	recordValue12 = St_dgenugennz (high byte)	M	XXh	RV_ST_DGENUGENNZ
16	recordValue13 = St_dgenugennz (low byte)	M	XXh	RV_ST_DGENUGENNZ
17	recordValue14 = St_dgenugenerr	M	XXh	RV_ST_DGENUGENERR
18	recordValue15 = St_dgengrenz1	M	XXh	RV_ST_DGENGRENZ1
19	recordValue16 = St_dgengrenz2	M	XXh	RV_ST_DGENGRENZ2
20	recordValue17 = St_dgengrenznz	M	XXh	RV_ST_DGENGRENZNZ
21	recordValue18 = St_dgengrenzerr	M	XXh	RV_ST_DGENGRENZERR
22	recordValue19 = St_dgenub1_md1 (high byte)	M	XXh	RV_ST_DGENUB1_MD1
23	recordValue20 = St_dgenub1_md1 (low byte)	M	XXh	RV_ST_DGENUB1_MD1
24	recordValue21 = St_dgenub2_md1 (high byte)	M	XXh	RV_ST_DGENUB2_MD1
25	recordValue22 = St_dgenub2_md1 (low byte)	M	XXh	RV_ST_DGENUB2_MD1
26	recordValue23 = St_dgenubnz_md1 (high byte)	M	XXh	RV_ST_DGENUBNZ_MD1
27	recordValue24 = St_dgenubnz_md1 (low byte)	M	XXh	RV_ST_DGENUBNZ_MD1
28	recordValue25 = St_dgenuberr_md1	M	XXh	RV_ST_DGENUBERR_MD1
29	recordValue26 = St_dgenugen1_md1 (high byte)	M	XXh	RV_ST_DGENUGEN1_MD1
30	recordValue27 = St_dgenugen1_md1 (low byte)	M	XXh	RV_ST_DGENUGEN1_MD1
31	recordValue28 = St_dgenugen2_md1 (high byte)	M	XXh	RV_ST_DGENUGEN2_MD1
32	recordValue29 = St_dgenugen2_md1 (low byte)	M	XXh	RV_ST_DGENUGEN2_MD1
33	recordValue30 = St_dgenugennz_md1 (high byte)	M	XXh	RV_ST_DGENUGENNZ_MD1
34	recordValue31 = St_dgenugennz_md1 (low byte)	M	XXh	RV_ST_DGENUGENNZ_MD1
35	recordValue32 = St_dgenugenerr_md1	M	XXh	RV_ST_DGENUGENERR_MD1
36	recordValue33 = St_dgengrenz1_md1	M	XXh	RV_ST_DGENGRENZ1_MD1
37	recordValue34 = St_dgengrenz2_md1	M	XXh	RV_ST_DGENGRENZ2_MD1
38	recordValue35 = St_dgengrenznz_md1	M	XXh	RV_ST_DGENGRENZNZ_MD1
39	recordValue36 = St_dgengrenzerr_md1	M	XXh	RV_ST_DGENGRENZERR_MD1
40	recordValue37 = St_dgenub1_md2 (high byte)	M	XXh	RV_ST_DGENUB1_MD2
41	recordValue38 = St_dgenub1_md2 (low byte)	M	XXh	RV_ST_DGENUB1_MD2
42	recordValue39 = St_dgenub2_md2 (high byte)	M	XXh	RV_ST_DGENUB2_MD2
43	recordValue40 = St_dgenub2_md2 (low byte)	M	XXh	RV_ST_DGENUB2_MD2
44	recordValue41 = St_dgenubnz_md2 (high hbyte)	M	XXh	RV_ST_DGENUBNZ_MD2
45	recordValue42 = St_dgenubnz_md2 (low byte)	M	XXh	RV_ST_DGENUBNZ_MD2
46	recordValue43 = St_dgenuberr_md2	M	XXh	RV_ST_DGENUBERR_MD2
47	recordValue44 = St_dgenugen1_md2 (high byte)	M	XXh	RV_ST_DGENUGEN1_MD2
48	recordValue45 = St_dgenugen1_md2 (low byte)	M	XXh	RV_ST_DGENUGEN1_MD2
49	recordValue46 = St_dgenugen2_md2	M	XXh	RV_ST_DGENUGEN2_MD2
50	recordValue47 = St_dgenugennz_md2 (high byte)	M	XXh	RV_ST_DGENUGENNZ_MD2
51	recordValue48 = St_dgenugennz_md2 (low byte)	M	XXh	RV_ST_DGENUGENNZ_MD2
52	recordValue49 = St_dgenugenerr_md2	M	XXh	RV_ST_DGENUGENERR_MD2
53	recordValue50 = St_dgengrenz1_md2	M	XXh	RV_ST_DGENGRENZ1_MD2
54	recordValue51 = St_dgengrenz2_md2	M	XXh	RV_ST_DGENGRENZ2_MD2
55	recordValue52 = St_dgengrenznz_md2	M	XXh	RV_ST_DGENGRENZNZ_MD2
56	recordValue53 = St_dgengrenzerr_md2	M	XXh	RV_ST_DGENGRENZERR_MD2
57	recordValue54 = St_dgenerrst_md1 (high byte)	M	XXh	RV_ST_DGENERRST_MD1

58	recordValue55 = St_dgenerrst_md1 (low byte)	M	XXh	RV_ST_DGENERRST_MD1
59	recordValue56 = St_dgenerrst_md2 (high byte)	M	XXh	RV_ST_DGENERRST_MD2
60	recordValue57 = St_dgenerrst_md2 (low byte)	M	XXh	RV_ST_DGENERRST_MD2

7.29.50 22_401C - Ringspeicher MSA lesen

General information:

This function is created to read out the "Ringspeicher MSA"

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	40h	RCI_MSARING
3	RecordCommonIdentifier Low Byte	M	1Ch	RCI_MSARING

Table Request Message

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MSARING
3	RecordCommonIdentifier Low Byte	M	1Ch	RCI_MSARING
4	recordValue1 = Msastz (high byte)	M	XXh	RV_MSASTZ
5	recordValue2 = Msastz (low byte)	M	XXh	RV_MSASTZ
6	recordValue3 = Msastzmsa (high byte)	M	XXh	RV_MSASTZMSA
7	recordValue4 = Msastzmsa (low byte)	M	XXh	RV_MSASTZMSA
8	recordValue5 = Msa_indexrs	M	XXh	RV_MSA_INDEXRS
9	recordValue6 = Msa_arravrs[1]	M	XXh	RV_MSA_ARRAVRS
10	recordValue7 = Msa_arravrs[2]	M	XXh	RV_MSA_ARRAVRS
11	recordValue8 = Msa_arravrs[3]	M	XXh	RV_MSA_ARRAVRS
12	recordValue9 = Msa_arravrs[4]	M	XXh	RV_MSA_ARRAVRS
13	recordValue10 = Msa_arravrs[5]	M	XXh	RV_MSA_ARRAVRS
14	recordValue11 = Msa_arravrs[6]	M	XXh	RV_MSA_ARRAVRS
15	recordValue12 = Msa_arravrs[7]	M	XXh	RV_MSA_ARRAVRS
16	recordValue13 = Msa_arravrs[8]	M	XXh	RV_MSA_ARRAVRS
17	recordValue14= Msa_arravrs[9]	M	XXh	RV_MSA_ARRAVRS
18	recordValue15 = Msa_arravrs[10]	M	XXh	RV_MSA_ARRAVRS
19	recordValue16 = Msa_arravrs[11]	M	XXh	RV_MSA_ARRAVRS
10	recordValue17 = Msa_arravrs[12]	M	XXh	RV_MSA_ARRAVRS
21	recordValue18 = Msa_arravrs[13]	M	XXh	RV_MSA_ARRAVRS
22	recordValue19= Msa_arravrs[14]	M	XXh	RV_MSA_ARRAVRS
23	recordValue20 = Msa_arravrs[15]	M	XXh	RV_MSA_ARRAVRS
24	recordValue21 = Msa_arravrs[16]	M	XXh	RV_MSA_ARRAVRS
25	recordValue22 = Msa_arravrs[17]	M	XXh	RV_MSA_ARRAVRS
26	recordValue23 = Msa_arravrs[18]	M	XXh	RV_MSA_ARRAVRS
27	recordValue24 = Msa_arravrs[19]	M	XXh	RV_MSA_ARRAVRS
28	recordValue25 = Msa_arravrs[20]	M	XXh	RV_MSA_ARRAVRS
29	recordValue26 = Msa_arravrs[21]	M	XXh	RV_MSA_ARRAVRS
30	recordValue27 = Msa_arravrs[22]	M	XXh	RV_MSA_ARRAVRS
31	recordValue28 = Msa_arravrs[23]	M	XXh	RV_MSA_ARRAVRS
32	recordValue29 = Msa_arravrs[24]	M	XXh	RV_MSA_ARRAVRS
33	recordValue30 = Msa_arravrs[25]	M	XXh	RV_MSA_ARRAVRS

34	recordValue31 = Msa_arravrs[26]	M	XXh	RV_MSA_ARRAVRS
35	recordValue32 = Msa_arravrs[27]	M	XXh	RV_MSA_ARRAVRS
36	recordValue33 = Msa_arravrs[28]	M	XXh	RV_MSA_ARRAVRS
37	recordValue34 = Msa_arravrs[29]	M	XXh	RV_MSA_ARRAVRS
38	recordValue35 = Msa_arravrs[30]	M	XXh	RV_MSA_ARRAVRS
39	recordValue36 = Msa_arravrs[31]	M	XXh	RV_MSA_ARRAVRS
40	recordValue37 = Msa_arravrs[32]	M	XXh	RV_MSA_ARRAVRS
41	recordValue38 = Msa_arravrs[33]	M	XXh	RV_MSA_ARRAVRS
42	recordValue39 = Msa_arravrs[34]	M	XXh	RV_MSA_ARRAVRS
43	recordValue40 = Msa_arravrs[35]	M	XXh	RV_MSA_ARRAVRS
44	recordValue41 = Msa_arravrs[36]	M	XXh	RV_MSA_ARRAVRS
45	recordValue42 = Msa_arravrs[37]	M	XXh	RV_MSA_ARRAVRS
46	recordValue43 = Msa_arravrs[38]	M	XXh	RV_MSA_ARRAVRS
47	recordValue44 = Msa_arravrs[39]	M	XXh	RV_MSA_ARRAVRS
48	recordValue45 = Msa_arravrs[40]	M	XXh	RV_MSA_ARRAVRS
49	recordValue46 = Msa_arravrs[41]	M	XXh	RV_MSA_ARRAVRS
50	recordValue47 = Msa_arravrs[42]	M	XXh	RV_MSA_ARRAVRS
51	recordValue48 = Msa_arravrs[43]	M	XXh	RV_MSA_ARRAVRS
52	recordValue49 = Msa_arravrs[44]	M	XXh	RV_MSA_ARRAVRS
53	recordValue50 = Msa_arravrs[45]	M	XXh	RV_MSA_ARRAVRS
54	recordValue51 = Msa_arravrs[46]	M	XXh	RV_MSA_ARRAVRS
55	recordValue52 = Msa_arravrs[47]	M	XXh	RV_MSA_ARRAVRS
56	recordValue53 = Msa_arravrs[48]	M	XXh	RV_MSA_ARRAVRS
57	recordValue54 = Msa_arravrs[49]	M	XXh	RV_MSA_ARRAVRS
58	recordValue55 = Msa_arravrs[50]	M	XXh	RV_MSA_ARRAVRS
59	recordValue56 = Msa_arravrs[51]	M	XXh	RV_MSA_ARRAVRS
60	recordValue57 = Msa_arravrs[52]	M	XXh	RV_MSA_ARRAVRS
61	recordValue58 = Msa_arravrs[53]	M	XXh	RV_MSA_ARRAVRS
62	recordValue59 = Msa_arravrs[54]	M	XXh	RV_MSA_ARRAVRS
63	recordValue60 = Msa_arravrs[55]	M	XXh	RV_MSA_ARRAVRS
64	recordValue61 = Msa_arravrs[56]	M	XXh	RV_MSA_ARRAVRS
65	recordValue62 = Msa_arravrs[57]	M	XXh	RV_MSA_ARRAVRS
66	recordValue63 = Msa_arravrs[58]	M	XXh	RV_MSA_ARRAVRS
67	recordValue64 = Msa_arravrs[59]	M	XXh	RV_MSA_ARRAVRS
68	recordValue65 = Msa_arravrs[60]	M	XXh	RV_MSA_ARRAVRS

7.29.51 22_401D - RCI_VERBREDINFO_lesen

General information:

This function is created to read out the "Verbredinfo" - Array

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	40h	RCI_VERBREDINFO
3	RecordCommonIdentifier Low Byte	M	1Dh	RCI_VERBREDINFO

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_VERBREDINFO
3	RecordCommonIdentifier Low Byte	M	1Dh	RCI_VERBREDINFO

4	recordValue1 = Verbredinfo[1]	M	XXh	RV_VERBREDINFO
5	recordValue2 = Verbredinfo[2]	M	XXh	RV_VERBREDINFO
6	recordValue3 = Verbredinfo[3]	M	XXh	RV_VERBREDINFO
7	recordValue4 = Verbredinfo[4]	M	XXh	RV_VERBREDINFO
8	recordValue5 = Verbredinfo[5]	M	XXh	RV_VERBREDINFO
9	recordValue6 = Verbredinfo[6]	M	XXh	RV_VERBREDINFO
10	recordValue7 = Verbredinfo[7]	M	XXh	RV_VERBREDINFO
11	recordValue8 = Verbredinfo[8]	M	XXh	RV_VERBREDINFO
12	recordValue9= Verbredinfo[9]	M	XXh	RV_VERBREDINFO
13	recordValue10 = Verbredinfo[10]	M	XXh	RV_VERBREDINFO
14	recordValue11 = Verbredinfo[11]	M	XXh	RV_VERBREDINFO
15	recordValue12 = Verbredinfo[12]	M	XXh	RV_VERBREDINFO
16	recordValue13 = Verbredinfo[13]	M	XXh	RV_VERBREDINFO
17	recordValue14= Verbredinfo[14]	M	XXh	RV_VERBREDINFO
18	recordValue15 = Verbredinfo[15]	M	XXh	RV_VERBREDINFO
19	recordValue16 = Verbredinfo[16]	M	XXh	RV_VERBREDINFO
20	recordValue17 = Verbredinfo[17]	M	XXh	RV_VERBREDINFO
21	recordValue18 = Verbredinfo[18]	M	XXh	RV_VERBREDINFO
22	recordValue19 = Verbredinfo[19]	M	XXh	RV_VERBREDINFO
23	recordValue20 = Verbredinfo[20]	M	XXh	RV_VERBREDINFO
24	recordValue21 = Verbredinfo[21]	M	XXh	RV_VERBREDINFO
25	recordValue22 = Verbredinfo[22]	M	XXh	RV_VERBREDINFO
26	recordValue23 = Verbredinfo[23]	M	XXh	RV_VERBREDINFO
27	recordValue24 = Verbredinfo[44]	M	XXh	RV_VERBREDINFO
28	recordValue25 = Verbredinfo[25]	M	XXh	RV_VERBREDINFO
29	recordValue26 = Verbredinfo[26]	M	XXh	RV_VERBREDINFO
30	recordValue27 = Verbredinfo[27]	M	XXh	RV_VERBREDINFO
31	recordValue28 = Verbredinfo[28]	M	XXh	RV_VERBREDINFO
32	recordValue29= Verbredinfo[29]	M	XXh	RV_VERBREDINFO
33	recordValue30 = Verbredinfo[30]	M	XXh	RV_VERBREDINFO
34	recordValue31 = Verbredinfo[31]	M	XXh	RV_VERBREDINFO
35	recordValue32 = Verbredinfo[32]	M	XXh	RV_VERBREDINFO

7.29.52 22_401E - Read CBS-Daten Zündkerzen

General information:

This function is created to read out the described values from the ecu.
Input and Output data see at service 22_10_01 for MSx80.

Formula section:

```
If  service 22_40_1E is recieved
then
      send 78 H
endif

If(1) 78H handling is activated for this function
Then(1)
      increment CTR_BOS_READ
      if(2)     CTR_BOS_READ > 1C2H             // timeout condition
      then(2)send negative response
                      LV_BOS_READ_REQ           = 0
```

```
                    CTR_BOS_READ              = 0
                    Bosanfsgbd                = 0
                    End data handling 78H
                                              // End data transmission to tester
    else(2)
            if(3)    LV_BOS_READ_REQ =0
            then(3)
                    If(4)    LC_SENS_ZK = 1
                    then(4)         Bosanfsgbd = 0AH
                                    LV_BOS_READ_REQ = 1
                    else(4)         send negative response
                                    LV_BOS_READ_REQ         = 0
                                    CTR_BOS_READ            = 0
                                    Bosanfsgbd              = 0
                                            End data handling 78H
                                               // End data transmission to tester
                    endif(4)
            else(3)
                    if(4)   Bosanfsgbd = Bostoken        // data for tester are ready
                    then(4) LV_BOS_READ_REQ = 0
                            send positive response
                            LV_BOS_READ_REQ         = 0
                            CTR_BOS_READ            = 0
                            Bosanfsgbd              = 0
                            End data handling 78H
                                              // End data transmission to tester
                    endif(4)
            endif(3)
    endif(2)
endif(1)
```

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_
3	RecordCommonIdentifier Low Byte	M	1Eh	RCI_

Table Request Message

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	40h	RCI_CBSZK_lesen
3	RecordCommonIdentifier	M	1Eh	RCI_CBSZK_lesen
4	recordValue1 = Id_bosmgt	M	XXh	RV_Id_bosmgt
5	recordValue2 = Bosrlsm (high byte)	M	XXh	RV_Bosrlsm
6	recordValue3 = Bosrlsm (low byte)	M	XXh	RV_Bosrlsm

7	recordValue4 = Bosstate	M	XXh	RV_Bosstate
8	recordValue5 = Zrbosmld	M	XXh	RV_Zrbosmld
9	recordValue6 = Bosbtvfbk	M	XXh	RV_Bosbtvfbk
10	recordValue7 = Bosrw2	M	XXh	RV_Bosrw2
11	recordValue8 = Bosmziel	M	XXh	RV_Bosmziel
12	recordValue9 = Bosjziel	M	XXh	RV_Bosjziel
13	recordValue10 = Bosprog2	M	XXh	RV_Bosprog2
14	recordValue11 = Bosmanip	M	XXh	RV_Bosmanip
15	recordValue12 = Bosres	M	XXh	RV_Bosres
16	recordValue13 = Bostoken	M	XXh	RV_Bostoken

7.29.53 22_401F - Read BMW-PST

General information:

This function is created to read out the "BMW-Programmstandsinformation"

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	40h	RCI_PST
3	RecordCommonIdentifier Low Byte	M	1Fh	RCI_PST

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_PST
3	RecordCommonIdentifier Low Byte	M	1Fh	RCI_PST
4	recordValue1 = Ecu_sw_ref_bmw[1]	M	XXh	RV_PST
5	recordValue2 = Ecu_sw_ref_bmw[2]	M	XXh	RV_PST
6	recordValue3 = Ecu_sw_ref_bmw[3]	M	XXh	RV_PST
7	recordValue4 = Ecu_sw_ref_bmw[4]	M	XXh	RV_PST
8	recordValue5 = Ecu_sw_ref_bmw[5]	M	XXh	RV_PST
9	recordValue6 = Ecu_sw_ref_bmw[6]	M	XXh	RV_PST
10	recordValue7 = Ecu_sw_ref_bmw[7]	M	XXh	RV_PST
11	recordValue8 = Ecu_sw_ref_bmw[8]	M	XXh	RV_PST
12	recordValue9 = Ecu_sw_ref_bmw[9]	M	XXh	RV_PST

7.29.54 22_4021 - RCI_IBS_lesen

Formula section:

BMW part numbers ASCII byte 7-1 conversion from Hex to ASCII:

The value of Zbibs is limited to a valid range between 0d and 9 999 999d for converting and reporting the data as ASCII string.
Note: zbibs_rest and temp_byte are only help values for conversion, they are not accessable as online-values or RAM-cells.

```
IF      Zbibs < 9 999 999d
THEN    zbibs_rest = zbibs
ELSE    zbibs_rest = 9 999 999d
ENDIF

temp_byte = (zbibs_rest / 1 000 000d)
BMW part numbers ASCII byte 7 = temp_byte + 30h
zbibs_rest = Zbibs - (temp_byte * 1 000 000d)

temp_byte = (zbibs_rest / 100 000d)
BMW part numbers ASCII byte 6 = temp_byte + 30h
zbibs_rest = zbibs_rest - (temp_byte * 100 000d)

temp_byte = (zbibs_rest / 10 000d)
BMW part numbers ASCII byte 5 = temp_byte + 30h
zbibs_rest = zbibs_rest - (temp_byte * 10 000d)

temp_byte = (zbibs_rest / 1 000d)
BMW part numbers ASCII byte 4 = temp_byte + 30h
zbibs_rest = zbibs_rest - (temp_byte * 1 000d)

temp_byte = (zbibs_rest / 100d)
BMW part numbers ASCII byte 3 = temp_byte + 30h
zbibs_rest = zbibs_rest - (temp_byte * 100d)

temp_byte = (zbibs_rest / 10d)
BMW part numbers ASCII byte 2 = temp_byte + 30h
zbibs_rest = zbibs_rest - (temp_byte * 10d)

BMW part numbers ASCII byte 1 = zbibs_rest + 30h
```

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Request Service Id**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	40h	RCI_IBS
3	RecordCommonIdentifier	M	21h	RCI_IBS

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**ReadDataByCommonIdentifier Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	40h	RCI_IBS
3	RecordCommonIdentifier	M	21h	RCI_IBS
4	RecordValue1 BMW part numbers ASCII byte 7	M	XXh	RV_ZB_IBS
5	RecordValue2 BMW part numbers ASCII byte 6	M	XXh	RV_ZB_IBS
6	RecordValue3 BMW part numbers ASCII byte 5	M	XXh	RV_ZB_IBS
7	RecordValue4 BMW part numbers ASCII byte 4	M	XXh	RV_ZB_IBS
8	RecordValue5 BMW part numbers ASCII byte 3	M	XXh	RV_ZB_IBS
9	RecordValue6 BMW part numbers ASCII byte 2	M	XXh	RV_ZB_IBS
10	RecordValue7 BMW part numbers ASCII byte 1	M	XXh	RV_ZB_IBS
11	RecordValue8 Snibs byte 3	M	XXh	RV_SN_IBS

12	RecordValue9 Snibs byte 2	M	XXh	RV_SN_IBS
13	RecordValue10 Snibs byte 1	M	XXh	RV_SN_IBS
14	RecordValue11 Snibs byte 0	M	XXh	RV_SN_IBS
15	RecordValue12 SW-baseline IBS_DEZ byte = Ibswbase	M	XXh	RV_SW_BL_IBS
16	RecordValue13 SW-index IBS_DEZ byte = Ibswchang	M	XXh	RV_SWIDX_IBS
17	RecordValue14 HW-index IBS_DEZ byte = Ibhwversi	M	XXh	RV_HWIDX_IBS
18	RecordValue15 reserved dummy byte not used	M	FFh	RV_
19	RecordValue16 reserved dummy byte not used	M	FFh	RV_
20	RecordValue17 reserved dummy byte not used	M	FFh	RV_
21	RecordValue18 reserved dummy byte not used	M	FFh	RV_
22	RecordValue19 reserved dummy byte not used	M	FFh	RV_
23	RecordValue20 reserved dummy byte not used	M	FFh	RV_

7.29.55 22_4022 - RCI_PM1_lesen

General information:

READ PM1 Status

This function ist for reading the data (array Pminfo1)of the Powermanagement.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_PM1_lesen
3	RecordCommonIdentifier Low Byte	M	22h	RCI_PM1_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_PM1_lesen
3	RecordCommonIdentifier Low Byte	M	22h	RCI_PM1_lesen
4	RecordValue1 = Pminfo1 Word 0 high byte	M	XXh	RV_PMINFO1
5	RecordValue2 = Pminfo1 Word 0 low byte	M	XXh	RV_PMINFO1
6	RecordValue3 = Pminfo1 Word 1 high byte	M	XXh	RV_PMINFO1
7	RecordValue4 = Pminfo1 Word 1 low byte	M	XXh	RV_PMINFO1
8	RecordValue5 = Pminfo1 Word 2 high byte	M	XXh	RV_PMINFO1
9	RecordValue6 = Pminfo1 Word 2 low byte	M	XXh	RV_PMINFO1
10	RecordValue7 = Pminfo1 Word 3 high byte	M	XXh	RV_PMINFO1
11	RecordValue8 = Pminfo1 Word 3 low byte	M	XXh	RV_PMINFO1
12	RecordValue9 = Pminfo1 Word 4 high byte	M	XXh	RV_PMINFO1
13	RecordValue10 = Pminfo1 Word 4 low byte	M	XXh	RV_PMINFO1
14	RecordValue11 = Pminfo1 Word 5 high byte	M	XXh	RV_PMINFO1
15	RecordValue12 = Pminfo1 Word 5 low byte	M	XXh	RV_PMINFO1
16	RecordValue13 = Pminfo1 Word 6 high byte	M	XXh	RV_PMINFO1
17	RecordValue14 = Pminfo1 Word 6 low byte	M	XXh	RV_PMINFO1
18	RecordValue15 = Pminfo1 Word 7 high byte	M	XXh	RV_PMINFO1
19	RecordValue16 = Pminfo1 Word 7 low byte	M	XXh	RV_PMINFO1
20	RecordValue17 = Pminfo1 Word 8 high byte	M	XXh	RV_PMINFO1
21	RecordValue18 = Pminfo1 Word 8 low byte	M	XXh	RV_PMINFO1
22	RecordValue19 = Pminfo1 Word 9 high byte	M	XXh	RV_PMINFO1

23	RecordValue20 = Pminfo1 Word 9 low byte	M	XXh	RV_PMINFO1
24	RecordValue21 = Pminfo1 Word 10 high byte	M	XXh	RV_PMINFO1
25	RecordValue22 = Pminfo1 Word 10 low byte	M	XXh	RV_PMINFO1
26	RecordValue23 = Pminfo1 Word 11 high byte	M	XXh	RV_PMINFO1
27	RecordValue24 = Pminfo1 Word 11 low byte	M	XXh	RV_PMINFO1
28	RecordValue25 = Pminfo1 Word 12 high byte	M	XXh	RV_PMINFO1
29	RecordValue26 = Pminfo1 Word 12 low byte	M	XXh	RV_PMINFO1
30	RecordValue27 = Pminfo1 Word 13 high byte	M	XXh	RV_PMINFO1
31	RecordValue28 = Pminfo1 Word 13 low byte	M	XXh	RV_PMINFO1
32	RecordValue29 = Pminfo1 Word 14 high byte	M	XXh	RV_PMINFO1
33	RecordValue30 = Pminfo1 Word 14 low byte	M	XXh	RV_PMINFO1
34	RecordValue31 = Pminfo1 Word 15 high byte	M	XXh	RV_PMINFO1
35	RecordValue32 = Pminfo1 Word 15 low byte	M	XXh	RV_PMINFO1
36	RecordValue33 = Pminfo1 Word 16 high byte	M	XXh	RV_PMINFO1
37	RecordValue34 = Pminfo1 Word 16 low byte	M	XXh	RV_PMINFO1
38	RecordValue35 = Pminfo1 Word 17 high byte	M	XXh	RV_PMINFO1
39	RecordValue36 = Pminfo1 Word 17 low byte	M	XXh	RV_PMINFO1
40	RecordValue37 = Pminfo1 Word 18 high byte	M	XXh	RV_PMINFO1
41	RecordValue38 = Pminfo1 Word 18 low byte	M	XXh	RV_PMINFO1
42	RecordValue39 = Pminfo1 Word 19 high byte	M	XXh	RV_PMINFO1
43	RecordValue40 = Pminfo1 Word 19 low byte	M	XXh	RV_PMINFO1
44	RecordValue41 = Pminfo1 Word 20 high byte	M	XXh	RV_PMINFO1
45	RecordValue42 = Pminfo1 Word 20 low byte	M	XXh	RV_PMINFO1
46	RecordValue43 = Pminfo1 Word 21 high byte	M	XXh	RV_PMINFO1
47	RecordValue44 = Pminfo1 Word 21 low byte	M	XXh	RV_PMINFO1
48	RecordValue45 = Pminfo1 Word 22 high byte	M	XXh	RV_PMINFO1
49	RecordValue46 = Pminfo1 Word 22 low byte	M	XXh	RV_PMINFO1
50	RecordValue47 = Pminfo1 Word 23 high byte	M	XXh	RV_PMINFO1
51	RecordValue48 = Pminfo1 Word 23 low byte	M	XXh	RV_PMINFO1
52	RecordValue49 = Pminfo1 Word 24 high byte	M	XXh	RV_PMINFO1
53	RecordValue50 = Pminfo1 Word 24 low byte	M	XXh	RV_PMINFO1
54	RecordValue51 = Pminfo1 Word 25 high byte	M	XXh	RV_PMINFO1
55	RecordValue52 = Pminfo1 Word 25 low byte	M	XXh	RV_PMINFO1
56	RecordValue53 = Pminfo1 Word 26 high byte	M	XXh	RV_PMINFO1
57	RecordValue54 = Pminfo1 Word 26 low byte	M	XXh	RV_PMINFO1
58	RecordValue55 = Pminfo1 Word 27 high byte	M	XXh	RV_PMINFO1
59	RecordValue56 = Pminfo1 Word 27 low byte	M	XXh	RV_PMINFO1
60	RecordValue57 = Pminfo1 Word 28 high byte	M	XXh	RV_PMINFO1
61	RecordValue58 = Pminfo1 Word 28 low byte	M	XXh	RV_PMINFO1
62	RecordValue59 = Pminfo1 Word 29 high byte	M	XXh	RV_PMINFO1
63	RecordValue60 = Pminfo1 Word 29 low byte	M	XXh	RV_PMINFO1
64	RecordValue61 = Pminfo1 Word 30 high byte	M	XXh	RV_PMINFO1
65	RecordValue62 = Pminfo1 Word 30 low byte	M	XXh	RV_PMINFO1
66	RecordValue63 = Pminfo1 Word 31 high byte	M	XXh	RV_PMINFO1
67	RecordValue64 = Pminfo1 Word 31 low byte	M	XXh	RV_PMINFO1
68	RecordValue65 = Pminfo1 Word 32 high byte	M	XXh	RV_PMINFO1
69	RecordValue66 = Pminfo1 Word 32 low byte	M	XXh	RV_PMINFO1
70	RecordValue67 = Pminfo1 Word 33 high byte	M	XXh	RV_PMINFO1
71	RecordValue68 = Pminfo1 Word 33 low byte	M	XXh	RV_PMINFO1
72	RecordValue69 = Pminfo1 Word 34 high byte	M	XXh	RV_PMINFO1
73	RecordValue70 = Pminfo1 Word 34 low byte	M	XXh	RV_PMINFO1
74	RecordValue71 = Pminfo1 Word 35 high byte	M	XXh	RV_PMINFO1
75	RecordValue72 = Pminfo1 Word 35 low byte	M	XXh	RV_PMINFO1

76	RecordValue73 = Pminfo1 Word 36 high byte	M	XXh	RV_PMINFO1
77	RecordValue74 = Pminfo1 Word 36 low byte	M	XXh	RV_PMINFO1

7.29.56 22_4023 - RCI_PM2_lesen

General information:

READ PM2 Status

This function ist for reading the data (array Pminfo2)of the Powermanagement.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_PM2_lesen
3	RecordCommonIdentifier Low Byte	M	23h	RCI_PM2_lesen

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_PM2_lesen
3	RecordCommonIdentifier Low Byte	M	23h	RCI_PM2_lesen
4	RecordValue1 = Pminfo2 byte 0	M	XXh	RV_PMINFO2
5	RecordValue2 = Pminfo2 byte 1	M	XXh	RV_PMINFO2
6	RecordValue3 = Pminfo2 byte 2	M	XXh	RV_PMINFO2
7	RecordValue4 = Pminfo2 byte 3	M	XXh	RV_PMINFO2
8	RecordValue5 = Pminfo2 byte 4	M	XXh	RV_PMINFO2
9	RecordValue6 = Pminfo2 byte 5	M	XXh	RV_PMINFO2
10	RecordValue7 = Pminfo2 byte 6	M	XXh	RV_PMINFO2
11	RecordValue8 = Pminfo2 byte 7	M	XXh	RV_PMINFO2
12	RecordValue9 = Pminfo2 byte 8	M	XXh	RV_PMINFO2
13	RecordValue10 = Pminfo2 byte 9	M	XXh	RV_PMINFO2
14	RecordValue11 = Pminfo2 byte 10	M	XXh	RV_PMINFO2
15	RecordValue12 = Pminfo2 byte 11	M	XXh	RV_PMINFO2
16	RecordValue13 = Pminfo2 byte 12	M	XXh	RV_PMINFO2
17	RecordValue14 = Pminfo2 byte 13	M	XXh	RV_PMINFO2
18	RecordValue15 = Pminfo2 byte 14	M	XXh	RV_PMINFO2
19	RecordValue16 = Pminfo2 byte 15	M	XXh	RV_PMINFO2
20	RecordValue17 = Pminfo2 byte 16	M	XXh	RV_PMINFO2
21	RecordValue18 = Pminfo2 byte 17	M	XXh	RV_PMINFO2
22	RecordValue19 = Pminfo2 byte 18	M	XXh	RV_PMINFO2
23	RecordValue20 = Pminfo2 byte 19	M	XXh	RV_PMINFO2
24	RecordValue21 = Pminfo2 byte 20	M	XXh	RV_PMINFO2
25	RecordValue22 = Pminfo2 byte 21	M	XXh	RV_PMINFO2
26	RecordValue23 = Pminfo2 byte 22	M	XXh	RV_PMINFO2
27	RecordValue24 = Pminfo2 byte 23	M	XXh	RV_PMINFO2
28	RecordValue25 = Pminfo2 byte 24	M	XXh	RV_PMINFO2
29	RecordValue26 = Pminfo2 byte 25	M	XXh	RV_PMINFO2
30	RecordValue27 = Pminfo2 byte 26	M	XXh	RV_PMINFO2
31	RecordValue28 = Pminfo2 byte 27	M	XXh	RV_PMINFO2
32	RecordValue29 = Pminfo2 byte 28	M	XXh	RV_PMINFO2

7.29.57 22_4026 - RCI_RBMMODE9_lesen

General information:

Rate Based Monitoring Mode 9 auslesen

This function ist for reading the rate based monitoring data. It gives back the denmominators and numerators of the RBM relevant diagnoses.

Description:

API to select Rate-Based Monitoring data to be transmitted via Mode $09

based on spec 30B02601.00E

This is only for information! This chapter will not be updated when the spec 30B02601 changes!

The following paragraph descibes the data selection algorithm to be executed only once (for CPU load saving reason), upon external tool request via Mode $09, by calling an API.

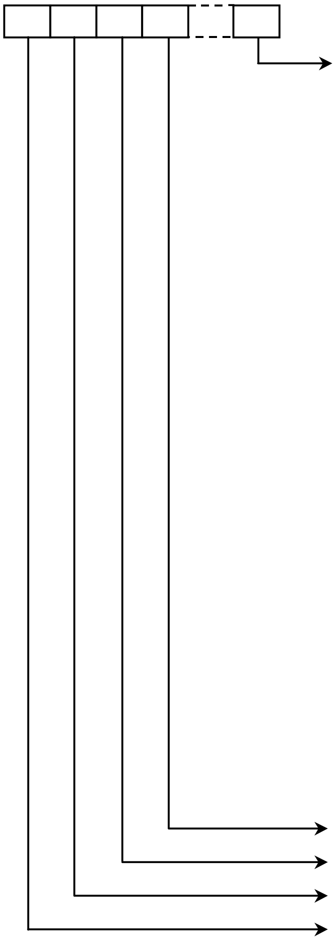
Syntax :	ACTION_ERRM_SelectRbmData (INOUT < ListRbmData >, OUT < ResultRbmData >)
Parameter (in) :	No parameter
Parameter (out) :	ResultRbmData
Parameter(inout) :	ListRbmData
Short description :	This API calculates and returns the Rate-Based Monitoring data to be transmitted to the Scan-Tool, when requested by the Mode 09$ (InfoType $08).

ResultRbmData : Indicates if the API has been executed or not.

Authorised values for ResultRbmDataSelection are :

- POSITIVE_RESPONSE : the call of the API is successful
- NEGATIVE_RESPONSE : the call of the API is unsuccessful

ListRbmData : Software structure filled-up with 16 counters of 2 bytes coming from Rate-Based Monitoring



EVAP Monitor Conditions Encountered Counts (denominator - LSB byte)
EVAP Monitor Conditions Encountered Counts (denominator - MSB byte)
EVAP Monitor Completion Condition Counts (numerator - LSB byte)
EVAP Monitor Completion Condition Counts (numerator - MSB byte)
AIR Monitor Conditions Encountered Counts (denominator - LSB byte)
AIR Monitor Conditions Encountered Counts (denominator - MSB byte)
AIR Monitor Completion Condition Counts (numerator - LSB byte)
AIR Monitor Completion Condition Counts (numerator - MSB byte)
EGR Monitor Conditions Encountered Counts (denominator - LSB byte)
EGR Monitor Conditions Encountered Counts (denominator – MSB byte)
EGR Monitor Completion Condition Counts (numerator - LSB byte)
EGR Monitor Completion Condition Counts (numerator - MSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 2 (denominator – LSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 2 (denominator - MSB byte)
O2 Sensor Monitor Completion Counts Bank 2 (numerator – LSB byte)
O2 Sensor Monitor Completion Counts Bank 2 (numerator - MSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 1 (denominator - LSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 1 (denominator - MSB byte)
O2 Sensor Monitor Completion Counts Bank 1 (numerator – LSB byte)
O2 Sensor Monitor Completion Counts Bank 1 (numerator – MSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 2 (denominator - LSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 2 (denominator - MSB byte)
Catalyst Monitor Completion Counts Bank 2 (numerator - LSB byte)
Catalyst Monitor Completion Counts Bank 2 (numerator - MSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 1 (denominator - LSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 1 (denominator - MSB byte)
Catalyst Monitor Completion Counts Bank 1 (numerator - LSB byte)
Catalyst Monitor Completion Counts Bank 1 (numerator - MSB byte)
Ignition Counter (LSB byte)
Ignition Counter (MSB byte)
OBD Monitoring Conditions Encountered Counts (general denominator - LSB byte)
OBD Monitoring Conditions Encountered Counts (general denominator - MSB byte)

The function described above is used to bring the RBM- values into KWP- buffer. The content of this KWP- buffer is given to tester with the telegram- layout described below.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMODE9
3	RecordCommonIdentifier Low Byte	M	26h	RCI_RBMMODE9

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMODE9
3	RecordCommonIdentifier Low Byte	M	26h	RCI_RBMMODE9
4	RecordValue1 = OBD Monitoring Conditions Encountered Counts (general denominator - MSB byte)	M	XXh	RV_OBDCOND
5	RecordValue2 = OBD Monitoring Conditions Encountered Counts (general denominator - LSB byte)	M	XXh	RV_OBDCOND
6	RecordValue3 = Ignition Counter (MSB byte)	M	XXh	RV_IGNCNTR

7	RecordValue4 = Ignition Counter (LSB byte)	M	XXh	RV_IGNCNTR
8	RecordValue5 = Catalyst Monitor Completion Counts Bank 1 (numerator - MSB byte)	M	XXh	RV_CATCOMP1
9	RecordValue6 = Catalyst Monitor Completion Counts Bank 1 (numerator - LSB byte)	M	XXh	RV_CATCOMP1
10	RecordValue7 = Catalyst Monitor Conditions Encountered Counts Bank 1 (denominator - MSB byte)	M	XXh	RV_CATCOND1
11	RecordValue8 = Catalyst Monitor Conditions Encountered Counts Bank 1 (denominator - LSB byte)	M	XXh	RV_CATCOND1
12	RecordValue9 = Catalyst Monitor Completion Counts Bank 2 (numerator - MSB byte)	M	XXh	RV_CATCOMP2
13	RecordValue10 = Catalyst Monitor Completion Counts Bank 2 (numerator - LSB byte)	M	XXh	RV_CATCOMP2
14	RecordValue11 = Catalyst Monitor Conditions Encountered Counts Bank 2 (denominator - MSB byte)	M	XXh	RV_CATCOND2
15	RecordValue12 = Catalyst Monitor Conditions Encountered Counts Bank 2 (denominator - LSB byte)	M	XXh	RV_CATCOND2
16	RecordValue13 = O2 Sensor Monitor Completion Counts Bank 1 (numerator – MSB byte)	M	XXh	RV_O2SCOMP1
17	RecordValue14 = O2 Sensor Monitor Completion Counts Bank 1 (numerator – LSB byte)	M	XXh	RV_O2SCOMP1
18	RecordValue15 = O2 Sensor Monitor Conditions Encountered Counts Bank 1 (denominator - MSB byte)	M	XXh	RV_O2SCOND1
19	RecordValue16 = O2 Sensor Monitor Conditions Encountered Counts Bank 1 (denominator - LSB byte)	M	XXh	RV_O2SCOND1
20	RecordValue17 = O2 Sensor Monitor Completion Counts Bank 2 (numerator - MSB byte)	M	XXh	RV_O2SCOMP2
21	RecordValue18 = O2 Sensor Monitor Completion Counts Bank 2 (numerator – LSB byte)	M	XXh	RV_O2SCOMP2
22	RecordValue19 = O2 Sensor Monitor Conditions Encountered Counts Bank 2 (denominator - MSB byte)	M	XXh	RV_O2SCOND2
23	RecordValue20 = O2 Sensor Monitor Conditions Encountered Counts Bank 2 (denominator – LSB byte)	M	XXh	RV_O2SCOND2
24	RecordValue21 = EGR Monitor Completion Condition Counts (numerator - MSB byte)	M	XXh	RV_EGRCOMP

25	RecordValue22 = EGR Monitor Completion Condition Counts (numerator - LSB byte)	M	XXh	RV_EGRCOMP
26	RecordValue23 = EGR Monitor Conditions Encountered Counts (denominator – MSB byte)	M	XXh	RV_EGRCOND
27	RecordValue24 = EGR Monitor Conditions Encountered Counts (denominator - LSB byte)	M	XXh	RV_EGRCOND
28	RecordValue25 = AIR Monitor Completion Condition Counts (numerator - MSB byte)	M	XXh	RV_AIRCOMP1
29	RecordValue26 = AIR Monitor Completion Condition Counts (numerator - LSB byte)	M	XXh	RV_AIRCOMP1
30	RecordValue27 = AIR Monitor Conditions Encountered Counts (denominator - MSB byte)	M	XXh	RV_AIRCOND1
31	RecordValue28 = AIR Monitor Conditions Encountered Counts (denominator - LSB byte)	M	XXh	RV_AIRCOND1
32	RecordValue29 = 00H	M	XXh	RV_AIRCOMP2
33	RecordValue30 = 00H	M	XXh	RV_AIRCOMP2
34	RecordValue31 = 00H	M	XXh	RV_AIRCOND2
35	RecordValue32 = 00H	M	XXh	RV_AIRCOND2
36	RecordValue33 = EVAP Monitor Completion Condition Counts (numerator - MSB byte)	M	XXh	RV_EVAPCOMP
37	RecordValue34 = EVAP Monitor Completion Condition Counts (numerator - LSB byte)	M	XXh	RV_EVAPCOMP
38	RecordValue35 = EVAP Monitor Conditions Encountered Counts (denominator - MSB byte)	M	XXh	RV_EVAPCOND
39	RecordValue36 = EVAP Monitor Conditions Encountered Counts (denominator - LSB byte)	M	XXh	RV_EVAPCOND
40	RecordValue37 = 00H	M	XXh	RV_VVTCOMP1
41	RecordValue38 = 00H	M	XXh	RV_VVTCOMP1
42	RecordValue39 = 00H	M	XXh	RV_VVTCOND1
43	RecordValue40 = 00H	M	XXh	RV_VVTCOND1
44	RecordValue41 = 00H	M	XXh	RV_VVTCOMP2
45	RecordValue42 = 00H	M	XXh	RV_VVTCOMP2
46	RecordValue43 = 00H	M	XXh	RV_VVTCOND2
47	RecordValue44 = 00H	M	XXh	RV_VVTCOND2

7.29.58 22_4027 - RCI_RBMMS1_lesen

General information:

Rate Based Monitoring Motorsteuerung Block 1 auslesen

This function reads out RBM values block 1.

7.29.58.1 MSD8x

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMS1
3	RecordCommonIdentifier Low Byte	M	27h	RCI_RBMMS1

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	40h	RCI_RBMMS1
3	RecordCommonIdentifier = Rate Based Monitoring Motorsteuerung MSV70 Block 1	M	27h	RCI_RBMMS1
4	RecordValue1 = CTR_COMP_RBM_CAT_DIAG_1 high byte	M	XXh	RV_CTR_COMP_RBM_CAT_DIAG_1
5	RecordValue2 = CTR_COMP_RBM_CAT_DIAG_1 low byte	M	XXh	RV_CTR_COMP_RBM_CAT_DIAG_1
6	RecordValue3 = CTR_CDN_RBM_CAT_DIAG_1 high byte	M	XXh	RV_CTR_CDN_RBM_CAT_DIAG_1
7	RecordValue4 = CTR_CDN_RBM_CAT_DIAG_1 low byte	M	XXh	RV_CTR_CDN_RBM_CAT_DIAG_1
8	RecordValue5 = CTR_COMP_RBM_CAT_DIAG_2 high byte	M	XXh	RV_CTR_COMP_RBM_CAT_DIAG_2
9	RecordValue6 = CTR_COMP_RBM_CAT_DIAG_2 low byte	M	XXh	RV_CTR_COMP_RBM_CAT_DIAG_2
10	RecordValue7 = CTR_CDN_RBM_CAT_DIAG_2 high byte	M	XXh	RV_CTR_CDN_RBM_CAT_DIAG_2
11	RecordValue8 = CTR_CDN_RBM_CAT_DIAG_2 low byte	M	XXh	RV_CTR_CDN_RBM_CAT_DIAG_2
12	RecordValue9 = CTR_COMP_RBM_DYN_VLD_LS_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_DYN_VLD_LS_UP_1
13	RecordValue10 = CTR_COMP_RBM_DYN_VLD_LS_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_DYN_VLD_LS_UP_1
14	RecordValue11 = CTR_CDN_RBM_DYN_VLD_LS_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_DYN_VLD_LS_UP_1
15	RecordValue12 = CTR_CDN_RBM_DYN_VLD_LS_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_DYN_VLD_LS_UP_1
16	RecordValue13 = CTR_COMP_RBM_DYN_VLD_LS_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_DYN_VLD_LS_UP_2
17	RecordValue14 = CTR_COMP_RBM_DYN_VLD_LS_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_DYN_VLD_LS_UP_2
18	RecordValue15 = CTR_CDN_RBM_DYN_VLD_LS_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_DYN_VLD_LS_UP_2
19	RecordValue16 = CTR_CDN_RBM_DYN_VLD_LS_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_DYN_VLD_LS_UP_2
20	RecordValue17 = CTR_COMP_RBM_SHIFT_AFL_LSL_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFL_LSL_UP_1
21	RecordValue18 = CTR_COMP_RBM_SHIFT_AFL_LSL_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFL_LSL_UP_1
22	RecordValue19 = CTR_CDN_RBM_SHIFT_AFL_LSL_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFL_LSL_UP_1
23	RecordValue20 = CTR_CDN_RBM_SHIFT_AFL_LSL_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFL_LSL_UP_1
24	RecordValue21 = CTR_COMP_RBM_SHIFT_AFL_LSL_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFL_LSL_UP_2
25	RecordValue22 = CTR_COMP_RBM_SHIFT_AFL_LSL_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFL_LSL_UP_2
26	RecordValue23 = CTR_CDN_RBM_SHIFT_AFL_LSL_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFL_LSL_UP_2

27	RecordValue24 = CTR_CDN_RBM_SHIFT_AFL_LSL_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFL_LSL_UP_2
28	RecordValue25 = CTR_COMP_RBM_SHIFT_AFR_LSL_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFR_LSL_UP_1
29	RecordValue26 = CTR_COMP_RBM_SHIFT_AFR_LSL_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFR_LSL_UP_1
30	RecordValue27 = CTR_CDN_RBM_SHIFT_AFR_LSL_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFR_LSL_UP_1
31	RecordValue28 = CTR_CDN_RBM_SHIFT_AFR_LSL_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFR_LSL_UP_1
32	RecordValue29 = CTR_COMP_RBM_SHIFT_AFR_LSL_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFR_LSL_UP_2
33	RecordValue30 = CTR_COMP_RBM_SHIFT_AFR_LSL_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFR_LSL_UP_2
34	RecordValue31 = CTR_CDN_RBM_SHIFT_AFR_LSL_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFR_LSL_UP_2
35	RecordValue32 = CTR_CDN_RBM_SHIFT_AFR_LSL_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFR_LSL_UP_2
36	RecordValue33 = CTR_COMP_RBM_AIR_LSL_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_AIR_LSL_UP_1
37	RecordValue34 = CTR_COMP_RBM_AIR_LSL_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_AIR_LSL_UP_1
38	RecordValue35 = CTR_CDN_RBM_AIR_LSL_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_AIR_LSL_UP_1
39	RecordValue36 = CTR_CDN_RBM_AIR_LSL_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_AIR_LSL_UP_1
40	RecordValue37 = CTR_COMP_RBM_AIR_LSL_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_AIR_LSL_UP_2
41	RecordValue38 = CTR_COMP_RBM_AIR_LSL_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_AIR_LSL_UP_2
42	RecordValue39 = CTR_CDN_RBM_AIR_LSL_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_AIR_LSL_UP_2
43	RecordValue40 = CTR_CDN_RBM_AIR_LSL_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_AIR_LSL_UP_2
44	RecordValue41 = CTR_COMP_RBM_SMALL_LEAK high byte	M	XXh	RV_CTR_COMP_RBM_SMALL_LEAK
45	RecordValue42 = CTR_COMP_RBM_SMALL_LEAK low byte	M	XXh	RV_CTR_COMP_RBM_SMALL_LEAK
46	RecordValue43 = CTR_CDN_RBM_SMALL_LEAK high byte	M	XXh	RV_CTR_CDN_RBM_SMALL_LEAK
47	RecordValue44 = CTR_CDN_RBM_SMALL_LEAK low byte	M	XXh	RV_CTR_CDN_RBM_SMALL_LEAK
48	RecordValue45 = not used	M	00h	RV_DUMMY_1
49	RecordValue46 = not used	M	00h	RV_DUMMY_2
50	RecordValue47 = not used	M	00h	RV_DUMMY_3
51	RecordValue48 = not used	M	00h	RV_DUMMY_4
52	RecordValue49 = not used	M	00h	RV_DUMMY_5
53	RecordValue50 = not used	M	00h	RV_DUMMY_6
54	RecordValue51 = not used	M	00h	RV_DUMMY_7
55	RecordValue52 = not used	M	00h	RV_DUMMY_8
56	RecordValue53 = CTR_COMP_RBM_MEC_IVVT_IN high byte	M	XXh	RV_CTR_COMP_RBM_MEC_IVVT_IN
57	RecordValue54 = CTR_COMP_RBM_MEC_IVVT_IN low byte	M	XXh	RV_CTR_COMP_RBM_MEC_IVVT_IN
58	RecordValue55 = CTR_CDN_RBM_MEC_IVVT_IN high byte	M	XXh	RV_CTR_CDN_RBM_MEC_IVVT_IN
59	RecordValue56 = CTR_CDN_RBM_MEC_IVVT_IN low byte	M	XXh	RV_CTR_CDN_RBM_MEC_IVVT_IN
60	RecordValue57 = CTR_COMP_RBM_MEC_IVVT_EX high byte	M	XXh	RV_CTR_COMP_RBM_MEC_IVVT_EX

61	RecordValue58 = CTR_COMP_RBM_MEC_IVVT_EX low byte	M	XXh	RV_CTR_COMP_RBM_MEC_IVVT_EX
62	RecordValue59 = CTR_CDN_RBM_MEC_IVVT_EX high byte	M	XXh	RV_CTR_CDN_RBM_MEC_IVVT_EX
63	RecordValue60 = CTR_CDN_RBM_MEC_IVVT_EX low byte	M	XXh	RV_CTR_CDN_RBM_MEC_IVVT_EX
64	RecordValue61 = CTR_COMP_RBM_TOOTH_OFF_IN_1 high byte	M	XXh	RV_CTR_COMP_RBM_TOOTH_OFF_IN_1
65	RecordValue62 = CTR_COMP_RBM_TOOTH_OFF_IN_1 low byte	M	XXh	RV_CTR_COMP_RBM_TOOTH_OFF_IN_1
66	RecordValue63 = CTR_CDN_RBM_TOOTH_OFF_IN_1 high byte	M	XXh	RV_CTR_CDN_RBM_TOOTH_OFF_IN_1
67	RecordValue64 = CTR_CDN_RBM_TOOTH_OFF_IN_1 low byte	M	XXh	RV_CTR_CDN_RBM_TOOTH_OFF_IN_1
68	RecordValue65 = CTR_COMP_RBM_TOOTH_OFF_EX_1 high byte	M	XXh	RV_CTR_COMP_RBM_TOOTH_OFF_EX_1
69	RecordValue66 = CTR_COMP_RBM_TOOTH_OFF_EX_1 low byte	M	XXh	RV_CTR_COMP_RBM_TOOTH_OFF_EX_1
70	RecordValue67 = CTR_CDN_RBM_TOOTH_OFF_EX_1 high byte	M	XXh	RV_CTR_CDN_RBM_TOOTH_OFF_EX_1
71	RecordValue68 = CTR_CDN_RBM_TOOTH_OFF_EX_1 low byte	M	XXh	RV_CTR_CDN_RBM_TOOTH_OFF_EX_1
72	RecordValue69 = (dummy) CTR_COMP_RBM_SA high byte	M	00h	RV_CTR_COMP_RBM_SA
73	RecordValue70 = (dummy) CTR_COMP_RBM_SA low byte	M	00h	RV_CTR_COMP_RBM_SA
74	RecordValue71 = (dummy) CTR_CDN_RBM_SA high byte	M	00h	RV_CTR_CDN_RBM_SA
75	RecordValue72 = (dummy) CTR_CDN_RBM_SA low byte	M	00h	RV_CTR_CDN_RBM_SA

Note: not available Values are 00h

7.29.58.2 MSV80

General information:

Rate Based Monitoring Motorsteuerung Block 1 auslesen

This function reads out RBM values block 1.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMS1
3	RecordCommonIdentifier Low Byte	M	27h	RCI_RBMMS1

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMS1
3	RecordCommonIdentifier Low Byte	M	27h	RCI_RBMMS1
4	RecordValue1 = CTR_COMP_RBM_CAT_DIAG_1 high byte	M	XXh	RV_CTR_COMP_RBM_CAT_DIAG_1
5	RecordValue2 = CTR_COMP_RBM_CAT_DIAG_1 low byte	M	XXh	RV_CTR_COMP_RBM_CAT_DIAG_1
6	RecordValue3 = CTR_CDN_RBM_CAT_DIAG_1 high byte	M	XXh	RV_CTR_CDN_RBM_CAT_DIAG_1
7	RecordValue4 = CTR_CDN_RBM_CAT_DIAG_1 low byte	M	XXh	RV_CTR_CDN_RBM_CAT_DIAG_1

8	RecordValue5 = CTR_COMP_RBM_CAT_DIAG_2 high byte	M	XXh	RV_CTR_COMP_RBM_CAT_DIAG_2
9	RecordValue6 = CTR_COMP_RBM_CAT_DIAG_2 low byte	M	XXh	RV_CTR_COMP_RBM_CAT_DIAG_2
10	RecordValue7 = CTR_CDN_RBM_CAT_DIAG_2 high byte	M	XXh	RV_CTR_CDN_RBM_CAT_DIAG_2
11	RecordValue8 = CTR_CDN_RBM_CAT_DIAG_2 low byte	M	XXh	RV_CTR_CDN_RBM_CAT_DIAG_2
12	RecordValue9 = CTR_COMP_RBM_DYN_VLD_LS_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_DYN_VLD_LS_UP_1
13	RecordValue10 = CTR_COMP_RBM_DYN_VLD_LS_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_DYN_VLD_LS_UP_1
14	RecordValue11 = CTR_CDN_RBM_DYN_VLD_LS_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_DYN_VLD_LS_UP_1
15	RecordValue12 = CTR_CDN_RBM_DYN_VLD_LS_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_DYN_VLD_LS_UP_1
16	RecordValue13 = CTR_COMP_RBM_DYN_VLD_LS_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_DYN_VLD_LS_UP_2
17	RecordValue14 = CTR_COMP_RBM_DYN_VLD_LS_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_DYN_VLD_LS_UP_2
18	RecordValue15 = CTR_CDN_RBM_DYN_VLD_LS_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_DYN_VLD_LS_UP_2
19	RecordValue16 = CTR_CDN_RBM_DYN_VLD_LS_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_DYN_VLD_LS_UP_2
20	RecordValue17 = CTR_COMP_RBM_SHIFT_AFL_LSL_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFL_LSL_UP_1
21	RecordValue18 = CTR_COMP_RBM_SHIFT_AFL_LSL_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFL_LSL_UP_1
22	RecordValue19 = CTR_CDN_RBM_SHIFT_AFL_LSL_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFL_LSL_UP_1
23	RecordValue20 = CTR_CDN_RBM_SHIFT_AFL_LSL_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFL_LSL_UP_1
24	RecordValue21 = CTR_COMP_RBM_SHIFT_AFL_LSL_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFL_LSL_UP_2
25	RecordValue22 = CTR_COMP_RBM_SHIFT_AFL_LSL_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFL_LSL_UP_2
26	RecordValue23 = CTR_CDN_RBM_SHIFT_AFL_LSL_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFL_LSL_UP_2
27	RecordValue24 = CTR_CDN_RBM_SHIFT_AFL_LSL_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFL_LSL_UP_2
28	RecordValue25 = CTR_COMP_RBM_SHIFT_AFR_LSL_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFR_LSL_UP_1
29	RecordValue26 = CTR_COMP_RBM_SHIFT_AFR_LSL_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFR_LSL_UP_1
30	RecordValue27 = CTR_CDN_RBM_SHIFT_AFR_LSL_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFR_LSL_UP_1
31	RecordValue28 = CTR_CDN_RBM_SHIFT_AFR_LSL_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFR_LSL_UP_1
32	RecordValue29 = CTR_COMP_RBM_SHIFT_AFR_LSL_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFR_LSL_UP_2
33	RecordValue30 = CTR_COMP_RBM_SHIFT_AFR_LSL_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_SHIFT_AFR_LSL_UP_2
34	RecordValue31 = CTR_CDN_RBM_SHIFT_AFR_LSL_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFR_LSL_UP_2
35	RecordValue32 = CTR_CDN_RBM_SHIFT_AFR_LSL_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_SHIFT_AFR_LSL_UP_2
36	RecordValue33 = CTR_COMP_RBM_AIR_LSL_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_AIR_LSL_UP_1
37	RecordValue34 = CTR_COMP_RBM_AIR_LSL_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_AIR_LSL_UP_1
38	RecordValue35 = CTR_CDN_RBM_AIR_LSL_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_AIR_LSL_UP_1

39	RecordValue36 = CTR_CDN_RBM_AIR_LSL_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_AIR_LSL_UP_1
40	RecordValue37 = CTR_COMP_RBM_AIR_LSL_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_AIR_LSL_UP_2
41	RecordValue38 = CTR_COMP_RBM_AIR_LSL_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_AIR_LSL_UP_2
42	RecordValue39 = CTR_CDN_RBM_AIR_LSL_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_AIR_LSL_UP_2
43	RecordValue40 = CTR_CDN_RBM_AIR_LSL_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_AIR_LSL_UP_2
44	RecordValue41 = CTR_COMP_RBM_SMALL_LEAK high byte	M	XXh	RV_CTR_COMP_RBM_SMALL_LEAK
45	RecordValue42 = CTR_COMP_RBM_SMALL_LEAK low byte	M	XXh	RV_CTR_COMP_RBM_SMALL_LEAK
46	RecordValue43 = CTR_CDN_RBM_SMALL_LEAK high byte	M	XXh	RV_CTR_CDN_RBM_SMALL_LEAK
47	RecordValue44 = CTR_CDN_RBM_SMALL_LEAK low byte	M	XXh	RV_CTR_CDN_RBM_SMALL_LEAK
48	RecordValue45 = not used	M	00h	RV_DUMMY
49	RecordValue46 = not used	M	00h	RV_DUMMY
50	RecordValue47 = not used	M	00h	RV_DUMMY
51	RecordValue48 = not used	M	00h	RV_DUMMY
52	RecordValue49 = not used	M	00h	RV_DUMMY
53	RecordValue50 = not used	M	00h	RV_DUMMY
54	RecordValue51 = not used	M	00h	RV_DUMMY
55	RecordValue52 = not used	M	XXh	RV_DUMMY
56	RecordValue53 = CTR_COMP_RBM_MEC_IVVT_IN high byte	M	XXh	RV_CTR_COMP_RBM_MEC_IVVT_IN
57	RecordValue54 = CTR_COMP_RBM_MEC_IVVT_IN low byte	M	XXh	RV_CTR_COMP_RBM_MEC_IVVT_IN
58	RecordValue55 = CTR_CDN_RBM_MEC_IVVT_IN high byte	M	XXh	RV_CTR_CDN_RBM_MEC_IVVT_IN
59	RecordValue56 = CTR_CDN_RBM_MEC_IVVT_IN low byte	M	XXh	RV_CTR_CDN_RBM_MEC_IVVT_IN
60	RecordValue57 = CTR_COMP_RBM_MEC_IVVT_EX high byte	M	XXh	RV_CTR_COMP_RBM_MEC_IVVT_EX
61	RecordValue58 = CTR_COMP_RBM_MEC_IVVT_EX low byte	M	XXh	RV_CTR_COMP_RBM_MEC_IVVT_EX
62	RecordValue59 = CTR_CDN_RBM_MEC_IVVT_EX high byte	M	XXh	RV_CTR_CDN_RBM_MEC_IVVT_EX
63	RecordValue60 = CTR_CDN_RBM_MEC_IVVT_EX low byte	M	XXh	RV_CTR_CDN_RBM_MEC_IVVT_EX
64	RecordValue61 = CTR_COMP_RBM_TOOTH_OFF_IN_1 high byte	M	XXh	RV_CTR_COMP_RBM_ TOOTH_OFF_IN_1
65	RecordValue62 = CTR_COMP_RBM_ TOOTH_OFF_IN_1 low byte	M	XXh	RV_CTR_COMP_RBM_ TOOTH_OFF_IN_1
66	RecordValue63 = CTR_CDN_RBM_ TOOTH_OFF_IN_1 byte	M	XXh	RV_CTR_CDN_RBM_ TOOTH_OFF_IN_1
67	RecordValue64 = CTR_CDN_RBM_ TOOTH_OFF_IN_1 low byte	M	XXh	RV_CTR_CDN_RBM_ TOOTH_OFF_IN_1
68	RecordValue65 = CTR_COMP_RBM_TOOTH_OFF_EX_1 high byte	M	XXh	RV_CTR_COMP_RBM_ TOOTH_OFF_EX_1
69	RecordValue66 = CTR_COMP_RBM_ TOOTH_OFF_EX_1 low byte	M	XXh	RV_CTR_COMP_RBM_ TOOTH_OFF_EX_1
70	RecordValue67 = CTR_CDN_RBM_ TOOTH_OFF_EX_1 byte	M	XXh	RV_CTR_CDN_RBM_ TOOTH_OFF_EX_1
71	RecordValue68 = CTR_CDN_RBM_ TOOTH_OFF_EX low byte	M	XXh	RV_CTR_CDN_RBM_ TOOTH_OFF_EX_1
72	RecordValue69 = CTR_COMP_RBM_SA_SYS high byte	M	XXh	RV_CTR_COMP_RBM_SA_SYS
73	RecordValue70 = CTR_COMP_RBM_SA_SYS low byte	M	XXh	RV_CTR_COMP_RBM_SA_SYS
74	RecordValue71 = CTR_CDN_RBM_SA_SYS high byte	M	XXh	RV_CTR_CDN_RBM_SA_SYS

75	RecordValue72 = CTR_CDN_RBM_SA_SYS low byte	M	XXh	RV_CTR_CDN_RBM_SA_SYS
76	RecordValue73 = CTR_COMP_RBM_SA_SAFM high byte	M	XXh	RV_CTR_COMP_RBM_SA_SAFM
77	RecordValue74 = CTR_COMP_RBM_SA_SAFM low byte	M	XXh	RV_CTR_COMP_RBM_SA_SAFM
78	RecordValue75 = CTR_CDN_RBM_SA_SAFM high byte	M	XXh	RV_CTR_CDN_RBM_SA_SAFM
79	RecordValue76 = CTR_CDN_RBM_SA_SAFM low byte	M	XXh	RV_CTR_CDN_RBM_SA_SAFM
80	RecordValue77 = CTR_COMP_RBM_SA_SAV high byte	M	XXh	RV_CTR_COMP_RBM_SA_SAV
81	RecordValue78 = CTR_COMP_RBM_SA_SAV low byte	M	XXh	RV_CTR_COMP_RBM_SA_SAV
82	RecordValue79 = CTR_CDN_RBM_SA_SAV high byte	M	XXh	RV_CTR_CDN_RBM_SA_SAV
83	RecordValue80 = CTR_CDN_RBM_SA_SAV low byte	M	XXh	RV_CTR_CDN_RBM_SA_SAV
84	RecordValue81 = CTR_COMP_RBM_SA_SAP high byte	M	XXh	RV_CTR_COMP_RBM_SA_SAP
85	RecordValue82 = CTR_COMP_RBM_SA_SAP low byte	M	XXh	RV_CTR_COMP_RBM_SA_SAP
86	RecordValue83 = CTR_CDN_RBM_SA_SAP high byte	M	XXh	RV_CTR_CDN_RBM_SA_SAP
87	RecordValue84 = CTR_CDN_RBM_SA_SAP low byte	M	XXh	RV_CTR_CDN_RBM_SA_SAP
88	RecordValue85 = CTR_COMP_RBM_SA_SAV_LSL high byte	M	XXh	RV_CTR_COMP_RBM_SA_SAV_LSL
89	RecordValue86 = CTR_COMP_RBM_SA_SAV_LSL low byte	M	XXh	RV_CTR_COMP_RBM_SA_SAV_LSL
90	RecordValue87 = CTR_CDN_RBM_SA_SAV_LSL high byte	M	XXh	RV_CTR_CDN_RBM_SA_SAV_LSL
91	RecordValue88 = CTR_CDN_RBM_SA_SAV_LSL low byte	M	XXh	RV_CTR_CDN_RBM_SA_SAV_LSL

7.29.59 22_4028 - RCI_RBMMS2_lesen

General information:

Rate Based Monitoring Motorsteuerung Block 2 auslesen

This function reads out RBM values block 2.

7.29.59.1 MSD8x

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMS2
3	RecordCommonIdentifier Low Byte	M	28h	RCI_RBMMS2

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMS2
3	RecordCommonIdentifier Low Byte	M	28h	RCI_RBMMS2
4	RecordValue1 = CTR_COMP_RBM_DIAGCPS high byte	M	XXh	RV_CTR_COMP_RBM_DIAGCPS
5	RecordValue2 = CTR_COMP_RBM_DIAGCPS low byte	M	XXh	RV_CTR_COMP_RBM_DIAGCPS
6	RecordValue3 = CTR_CDN_RBM_DIAGCPS high byte	M	XXh	RV_CTR_CDN_RBM_DIAGCPS
7	RecordValue4 = CTR_CDN_RBM_DIAGCPS low byte	M	XXh	RV_CTR_CDN_RBM_DIAGCPS
8	RecordValue5 = CTR_COMP_RBM_ROUGH_LEAK high byte	M	XXh	RV_CTR_COMP_RBM_ROUGH_LEAK
9	RecordValue6 = CTR_COMP_RBM_ROUGH_LEAK low byte	M	XXh	RV_CTR_COMP_RBM_ROUGH_LEAK
10	RecordValue7 = CTR_CDN_RBM_ROUGH_LEAK high byte	M	XXh	RV_CTR_CDN_RBM_ROUGH_LEAK
11	RecordValue8 = CTR_CDN_RBM_ROUGH_LEAK low byte	M	XXh	RV_CTR_CDN_RBM_ROUGH_LEAK

12	RecordValue9 = CTR_COMP_RBM_DMTLM high byte	M	XXh	RV_CTR_COMP_RBM_DMTLM
13	RecordValue10 = CTR_COMP_RBM_DMTLM low byte	M	XXh	RV_CTR_COMP_RBM_DMTLM
14	RecordValue11 = CTR_CDN_RBM_DMTLM high byte	M	XXh	RV_CTR_CDN_RBM_DMTLM
15	RecordValue12 = CTR_CDN_RBM_DMTLM low byte	M	XXh	RV_CTR_CDN_RBM_DMTLM
16	RecordValue13 = CTR_COMP_RBM_TH high byte	M	XXh	RV_CTR_COMP_RBM_TH
17	RecordValue14 = CTR_COMP_RBM_TH low byte	M	XXh	RV_CTR_COMP_RBM_TH
18	RecordValue15 = CTR_CDN_RBM_TH high byte	M	XXh	RV_CTR_CDN_RBM_TH
19	RecordValue16 = CTR_CDN_RBM_TH low byte	M	XXh	RV_CTR_CDN_RBM_TH
20	RecordValue17 = CTR_COMP_RBM_TCO_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_TCO_PLAUS
21	RecordValue18 = CTR_COMP_RBM_TCO_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_TCO_PLAUS
22	RecordValue19 = CTR_CDN_RBM_TCO_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_TCO_PLAUS
23	RecordValue20 = CTR_CDN_RBM_TCO_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_TCO_PLAUS
24	RecordValue21 = CTR_COMP_RBM_TCO_STUCK high byte	M	XXh	RV_CTR_COMP_RBM_TCO_STUCK
25	RecordValue22 = CTR_COMP_RBM_TCO_STUCK low byte	M	XXh	RV_CTR_COMP_RBM_TCO_STUCK
26	RecordValue23 = CTR_CDN_RBM_TCO_STUCK high byte	M	XXh	RV_CTR_CDN_RBM_TCO_STUCK
27	RecordValue24 = CTR_CDN_RBM_TCO_STUCK low byte	M	XXh	RV_CTR_CDN_RBM_TCO_STUCK
28	RecordValue25 = CTR_COMP_RBM_TCO_2_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_TCO_2_PLAUS
29	RecordValue26 = CTR_COMP_RBM_TCO_2_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_TCO_2_PLAUS
30	RecordValue27 = CTR_CDN_RBM_TCO_2_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_TCO_2_PLAUS
31	RecordValue28 = CTR_CDN_RBM_TCO_2_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_TCO_2_PLAUS
32	RecordValue29 = CTR_COMP_RBM_TAM_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_TAM_PLAUS
33	RecordValue30 = CTR_COMP_RBM_TAM_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_TAM_PLAUS
34	RecordValue31 = CTR_CDN_RBM_TAM_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_TAM_PLAUS
35	RecordValue32 = CTR_CDN_RBM_TAM_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_TAM_PLAUS
36	RecordValue33 = CTR_COMP_RBM_VS_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_VS_PLAUS
37	RecordValue34 = CTR_COMP_RBM_VS_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_VS_PLAUS
38	RecordValue35 = CTR_CDN_RBM_VS_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_VS_PLAUS
39	RecordValue36 = CTR_CDN_RBM_VS_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_VS_PLAUS
40	RecordValue37 = CTR_COMP_RBM_FTL_OBD high byte	M	XXh	RV_CTR_COMP_RBM_FTL_OBD
41	RecordValue38 = CTR_COMP_RBM_FTL_OBD low byte	M	XXh	RV_CTR_COMP_RBM_FTL_OBD
42	RecordValue39 = CTR_CDN_RBM_FTL_OBD high byte	M	XXh	RV_CTR_CDN_RBM_FTL_OBD
43	RecordValue40 = CTR_CDN_RBM_FTL_OBD low byte	M	XXh	RV_CTR_CDN_RBM_FTL_OBD
44	RecordValue41 = CTR_COMP_RBM_OBD_LSH_DOWN_1 high byte	M	XXh	RV_CTR_COMP_RBM_OBD_LSH_DOWN_1
45	RecordValue42 = CTR_COMP_RBM_OBD_LSH_DOWN_1 low byte	M	XXh	RV_CTR_COMP_RBM_OBD_LSH_DOWN_1
46	RecordValue43 = CTR_CDN_RBM_OBD_LSH_DOWN_1 high byte	M	XXh	RV_CTR_CDN_RBM_OBD_LSH_DOWN_1
47	RecordValue44 = CTR_CDN_RBM_OBD_LSH_DOWN_1 low byte	M	XXh	RV_CTR_CDN_RBM_OBD_LSH_DOWN_1
48	RecordValue45 = CTR_COMP_RBM_OBD_LSH_DOWN_2 high byte	M	XXh	RV_CTR_COMP_RBM_OBD_LSH_DOWN_2
49	RecordValue46 = CTR_COMP_RBM_OBD_LSH_DOWN_2 low byte	M	XXh	RV_CTR_COMP_RBM_OBD_LSH_DOWN_2
50	RecordValue47 = CTR_CDN_RBM_OBD_LSH_DOWN_2 high byte	M	XXh	RV_CTR_CDN_RBM_OBD_LSH_DOWN_2

51	RecordValue48 = CTR_CDN_RBM_OBD_LSH_DOWN_2 low byte	M	XXh	RV_CTR_CDN_RBM_OBD_LSH_DOWN_2
52	RecordValue49 = CTR_COMP_RBM_OBD_VLD_LSH_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_OBD_VLD_LSH_UP_1
53	RecordValue50 = CTR_COMP_RBM_OBD_VLD_LSH_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_OBD_VLD_LSH_UP_1
54	RecordValue51 = CTR_CDN_RBM_OBD_VLD_LSH_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_OBD_VLD_LSH_UP_1
55	RecordValue52 = CTR_CDN_RBM_OBD_VLD_LSH_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_OBD_VLD_LSH_UP_1
56	RecordValue53 = CTR_COMP_RBM_OBD_VLD_LSH_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_OBD_VLD_LSH_UP_2
57	RecordValue54 = CTR_COMP_RBM_OBD_VLD_LSH_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_OBD_VLD_LSH_UP_2
58	RecordValue55 = CTR_CDN_RBM_OBD_VLD_LSH_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_OBD_VLD_LSH_UP_2
59	RecordValue56 = CTR_CDN_RBM_OBD_VLD_LSH_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_OBD_VLD_LSH_UP_2
60	RecordValue57 = CTR_COMP_RBM_CS high byte	M	XXh	RV_CTR_COMP_RBM_CS
61	RecordValue58 = CTR_COMP_RBM_CS low byte	M	XXh	RV_CTR_COMP_RBM_CS
62	RecordValue59 = CTR_CDN_RBM_CS high byte	M	XXh	RV_CTR_CDN_RBM_CS
63	RecordValue60 = CTR_CDN_RBM_CS low byte	M	XXh	RV_CTR_CDN_RBM_CS
64	RecordValue61 = CTR_COMP_RBM_ISC high byte	M	XXh	RV_CTR_COMP_RBM_ISC
65	RecordValue62 = CTR_COMP_RBM_ISC low byte	M	XXh	RV_CTR_COMP_RBM_ISC
66	RecordValue63 = CTR_CDN_RBM_ISC high byte	M	XXh	RV_CTR_CDN_RBM_ISC
67	RecordValue64 = CTR_CDN_RBM_ISC low byte	M	XXh	RV_CTR_CDN_RBM_ISC
68	RecordValue65 = CTR_COMP_RBM_MAF high byte	M	XXh	RV_CTR_COMP_RBM_MAF
69	RecordValue66 = CTR_COMP_RBM_MAF low byte	M	XXh	RV_CTR_COMP_RBM_MAF
70	RecordValue67 = CTR_CDN_RBM_MAF high byte	M	XXh	RV_CTR_CDN_RBM_MAF
71	RecordValue68 = CTR_CDN_RBM_MAF low byte	M	XXh	RV_CTR_CDN_RBM_MAF
77	RecordValue69 = CTR_COMP_RBM_TIA_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_TIA_PLAUS
73	RecordValue70 = CTR_COMP_RBM_TIA_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_TIA_PLAUS
74	RecordValue71 = CTR_CDN_RBM_TIA_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_TIA_PLAUS
75	RecordValue72 = CTR_CDN_RBM_TIA_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_TIA_PLAUS
76	RecordValue73 = CTR_COMP_RBM_AMP_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_AMP_PLAUS
77	RecordValue74 = CTR_COMP_RBM_AMP_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_AMP_PLAUS
78	RecordValue75 = CTR_CDN_RBM_AMP_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_AMP_PLAUS
79	RecordValue76 = CTR_CDN_RBM_AMP_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_AMP_PLAUS
80	RecordValue77 = CTR_COMP_RBM_LOAD_TPS_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_LOAD_TPS_PLAUS
81	RecordValue78 = CTR_COMP_RBM_LOAD_TPS_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_LOAD_TPS_PLAUS
82	RecordValue79 = CTR_CDN_RBM_LOAD_TPS_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_LOAD_TPS_PLAUS
83	RecordValue80 = CTR_CDN_RBM_LOAD_TPS_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_LOAD_TPS_PLAUS
84	RecordValue81 = CTR_COMP_RBM_MAP_DIP_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_MAP_DIP_PLAUS
85	RecordValue82 = CTR_COMP_RBM_MAP_DIP_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_MAP_DIP_PLAUS
86	RecordValue83 = CTR_CDN_RBM_MAP_DIP_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_MAP_DIP_PLAUS
87	RecordValue84 = CTR_CDN_RBM_MAP_DIP_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_MAP_DIP_PLAUS

88	RecordValue85 = CTR_COMP_RBM_VLS_DOWN_DIF_1 high byte	M	XXh	RV_CTR_COMP_RBM_VLS_DOWN_DIF_1
89	RecordValue86 = CTR_COMP_RBM_VLS_DOWN_DIF_1 low byte	M	XXh	RV_CTR_COMP_RBM_VLS_DOWN_DIF_1
90	RecordValue87 = CTR_CDN_RBM_VLS_DOWN_DIF_1 high byte	M	XXh	RV_CTR_CDN_RBM_VLS_DOWN_DIF_1
91	RecordValue88 = CTR_CDN_RBM_VLS_DOWN_DIF_1 low byte	M	XXh	RV_CTR_CDN_RBM_VLS_DOWN_DIF_1
92	RecordValue89 = CTR_COMP_RBM_VLS_DOWN_DIF_2 high byte	M	XXh	RV_CTR_COMP_RBM_VLS_DOWN_DIF_2
93	RecordValue90 = CTR_COMP_RBM_VLS_DOWN_DIF_2 low byte	M	XXh	RV_CTR_COMP_RBM_VLS_DOWN_DIF_2
94	RecordValue91 = CTR_CDN_RBM_VLS_DOWN_DIF_2 high byte	M	XXh	RV_CTR_CDN_RBM_VLS_DOWN_DIF_2
95	RecordValue92 = CTR_CDN_RBM_VLS_DOWN_DIF_2 low byte	M	XXh	RV_CTR_CDN_RBM_VLS_DOWN_DIF_2
96	RecordValue93 = CTR_COMP_RBM_CHK_LS_DOWN_1 high byte	M	XXh	RV_CTR_COMP_RBM_CHK_LS_DOWN_1
97	RecordValue94 = CTR_COMP_RBM_CHK_LS_DOWN_1 low byte	M	XXh	RV_CTR_COMP_RBM_CHK_LS_DOWN_1
98	RecordValue95 = CTR_CDN_RBM_CHK_LS_DOWN_1 high byte	M	XXh	RV_CTR_CDN_RBM_CHK_LS_DOWN_1
99	RecordValue96 = CTR_CDN_RBM_CHK_LS_DOWN_1 low byte	M	XXh	RV_CTR_CDN_RBM_CHK_LS_DOWN_1
100	RecordValue97 = CTR_COMP_RBM_CHK_LS_DOWN_2 high byte	M	XXh	RV_CTR_COMP_RBM_CHK_LS_DOWN_2
101	RecordValue98 = CTR_COMP_RBM_CHK_LS_DOWN_2 low byte	M	XXh	RV_CTR_COMP_RBM_CHK_LS_DOWN_2
102	RecordValue99 = CTR_CDN_RBM_CHK_LS_DOWN_2 high byte	M	XXh	RV_CTR_CDN_RBM_CHK_LS_DOWN_2
103	RecordValue100= CTR_CDN_RBM_CHK_LS_DOWN_2 low byte	M	XXh	RV_CTR_CDN_RBM_CHK_LS_DOWN_2
104	RecordValue101= CTR_COMP_RBM_SWT_LS_DOWN_1 high byte	M	XXh	RV_CTR_COMP_RBM_SWT_LS_DOWN_1
105	RecordValue102 = CTR_COMP_RBM_SWT_LS_DOWN_1 low byte	M	XXh	RV_CTR_COMP_RBM_SWT_LS_DOWN_1
106	RecordValue103 = CTR_CDN_RBM_SWT_LS_DOWN_1 high byte	M	XXh	RV_CTR_CDN_RBM_SWT_LS_DOWN_1
107	RecordValue104 = CTR_CDN_RBM_SWT_LS_DOWN_1 low byte	M	XXh	RV_CTR_CDN_RBM_SWT_LS_DOWN_1
108	RecordValue105 = CTR_COMP_RBM_SWT_LS_DOWN_2 high byte	M	XXh	RV_CTR_COMP_RBM_SWT_LS_DOWN_2
109	RecordValue106 = CTR_COMP_RBM_SWT_LS_DOWN_2 low byte	M	XXh	RV_CTR_COMP_RBM_SWT_LS_DOWN_2
110	RecordValue107 = CTR_CDN_RBM_SWT_LS_DOWN_2 high byte	M	XXh	RV_CTR_CDN_RBM_SWT_LS_DOWN_2
111	RecordValue108 = CTR_CDN_RBM_SWT_LS_DOWN_2 low byte	M	XXh	RV_CTR_CDN_RBM_SWT_LS_DOWN_2
112	RecordValue109 = CTR_COMP_RBM_T_ES high byte	M	XXh	RV_CTR_COMP_RBM_T_ES
113	RecordValue110 = CTR_COMP_RBM_T_ES low byte	M	XXh	RV_CTR_COMP_RBM_T_ES
114	RecordValue111 = CTR_CDN_RBM_T_ES high byte	M	XXh	RV_CTR_CDN_RBM_T_ES
115	RecordValue112 = CTR_CDN_RBM_T_ES low byte	M	XXh	RV_CTR_CDN_RBM_T_ES
116	RecordValue113 = CTR_COMP_RBM_TPS high byte	M	XXh	RV_CTR_COMP_RBM_TPS
117	RecordValue114 = CTR_COMP_RBM_TPS low byte	M	XXh	RV_CTR_COMP_RBM_TPS
118	RecordValue115 = CTR_CDN_RBM_TPS high byte	M	XXh	RV_CTR_CDN_RBM_TPS
119	RecordValue116 = CTR_CDN_RBM_TPS low byte	M	XXh	RV_CTR_CDN_RBM_TPS
120	RecordValue117 = CTR_COMP_RBM_ISC_CST high byte	M	XXh	RV_CTR_COMP_RBM_ISC_CST
121	RecordValue118 = CTR_COMP_RBM_ISC_CST low byte	M	XXh	RV_CTR_COMP_RBM_ISC_CST

122	RecordValue119 = CTR_CDN_RBM_ISC_CST high byte	M	XXh	RV_CTR_CDN_RBM_ISC_CST
123	RecordValue120 = CTR_CDN_RBM_ISC_CST low byte	M	XXh	RV_CTR_CDN_RBM_ISC_CST
124	RecordValue121 = CTR_COMP_RBM_TQ_CSTL high byte	M	XXh	RV_CTR_COMP_RBM_TQ_CSTL
125	RecordValue122 = CTR_COMP_RBM_TQ_CSTL low byte	M	XXh	RV_CTR_COMP_RBM_TQ_CSTL
126	RecordValue123 = CTR_CDN_RBM_TQ_CSTL high byte	M	XXh	RV_CTR_CDN_RBM_TQ_CSTL
127	RecordValue124 = CTR_CDN_RBM_TQ_CSTL low byte	M	XXh	RV_CTR_CDN_RBM_TQ_CSTL

7.29.59.2 MSV80

General information:

Rate Based Monitoring Motorsteuerung Block 2 auslesen

This function reads out RBM values block 2.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMS2
3	RecordCommonIdentifier Low Byte	M	28h	RCI_RBMMS2

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_RBMMS2
3	RecordCommonIdentifier Low Byte	M	28h	RCI_RBMMS2
4	RecordValue1 = CTR_COMP_RBM_DIAGCPS high byte	M	XXh	RV_CTR_COMP_RBM_DIAGCPS
5	RecordValue2 = CTR_COMP_RBM_DIAGCPS low byte	M	XXh	RV_CTR_COMP_RBM_DIAGCPS
6	RecordValue3 = CTR_CDN_RBM_DIAGCPS high byte	M	XXh	RV_CTR_CDN_RBM_DIAGCPS
7	RecordValue4 = CTR_CDN_RBM_DIAGCPS low byte	M	XXh	RV_CTR_CDN_RBM_DIAGCPS
8	RecordValue5 = CTR_COMP_RBM_ROUGH_LEAK high byte	M	XXh	RV_CTR_COMP_RBM_ROUGH_LEAK
9	RecordValue6 = CTR_COMP_RBM_ROUGH_LEAK low byte	M	XXh	RV_CTR_COMP_RBM_ROUGH_LEAK
10	RecordValue7 = CTR_CDN_RBM_ROUGH_LEAK high byte	M	XXh	RV_CTR_CDN_RBM_ROUGH_LEAK
11	RecordValue8 = CTR_CDN_RBM_ROUGH_LEAK low byte	M	XXh	RV_CTR_CDN_RBM_ROUGH_LEAK
12	RecordValue9 = CTR_COMP_RBM_DMTLM high byte	M	XXh	RV_CTR_COMP_RBM_DMTLM
13	RecordValue10 = CTR_COMP_RBM_DMTLM low byte	M	XXh	RV_CTR_COMP_RBM_DMTLM
14	RecordValue11 = CTR_CDN_RBM_DMTLM high byte	M	XXh	RV_CTR_CDN_RBM_DMTLM
15	RecordValue12 = CTR_CDN_RBM_DMTLM low byte	M	XXh	RV_CTR_CDN_RBM_DMTLM
16	RecordValue13 = CTR_COMP_RBM_TH high byte	M	XXh	RV_CTR_COMP_RBM_TH
17	RecordValue14 = CTR_COMP_RBM_TH low byte	M	XXh	RV_CTR_COMP_RBM_TH
18	RecordValue15 = CTR_CDN_RBM_TH high byte	M	XXh	RV_CTR_CDN_RBM_TH
19	RecordValue16= CTR_CDN_RBM_TH low byte	M	XXh	RV_CTR_CDN_RBM_TH
20	RecordValue17= CTR_COMP_RBM_TCO_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_TCO_PLAUS
21	RecordValue18= CTR_COMP_RBM_TCO_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_TCO_PLAUS
22	RecordValue19 = CTR_CDN_RBM_TCO_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_TCO_PLAUS
23	RecordValue20 = CTR_CDN_RBM_TCO_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_TCO_PLAUS
24	RecordValue21 = CTR_COMP_RBM_TCO_STUCK high	M	XXh	RV_CTR_COMP_RBM_TCO_STUCK

	byte			
25	RecordValue22 = CTR_COMP_RBM_TCO_STUCK low byte	M	XXh	RV_CTR_COMP_RBM_TCO_STUCK
26	RecordValue23 = CTR_CDN_RBM_TCO_STUCK high byte	M	XXh	RV_CTR_CDN_RBM_TCO_STUCK
27	RecordValue24 = CTR_CDN_RBM_TCO_STUCK low byte	M	XXh	RV_CTR_CDN_RBM_TCO_STUCK
28	RecordValue25 = CTR_COMP_RBM_TCO_2_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_TCO_2_PLAUS
29	RecordValue26 = CTR_COMP_RBM_TCO_2_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_TCO_2_PLAUS
30	RecordValue27 = CTR_CDN_RBM_TCO_2_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_TCO_2_PLAUS
31	RecordValue28 = CTR_CDN_RBM_TCO_2_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_TCO_2_PLAUS
32	RecordValue29 = CTR_COMP_RBM_TAM_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_TAM_PLAUS
33	RecordValue30 = CTR_COMP_RBM_TAM_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_TAM_PLAUS
34	RecordValue31 = CTR_CDN_RBM_TAM_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_TAM_PLAUS
35	RecordValue32 = CTR_CDN_RBM_TAM_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_TAM_PLAUS
36	RecordValue33 = CTR_COMP_RBM_VS_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_VS_PLAUS
37	RecordValue34 = CTR_COMP_RBM_VS_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_VS_PLAUS
38	RecordValue35 = CTR_CDN_RBM_VS_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_VS_PLAUS
39	RecordValue36 = CTR_CDN_RBM_VS_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_VS_PLAUS
40	RecordValue37 = CTR_COMP_RBM_FTL_OBD high byte	M	XXh	RV_CTR_COMP_RBM_FTL_OBD
41	RecordValue38 = CTR_COMP_RBM_FTL_OBD low byte	M	XXh	RV_CTR_COMP_RBM_FTL_OBD
42	RecordValue39 = CTR_CDN_RBM_FTL_OBD high byte	M	XXh	RV_CTR_CDN_RBM_FTL_OBD
43	RecordValue40 = CTR_CDN_RBM_FTL_OBD low byte	M	XXh	RV_CTR_CDN_RBM_FTL_OBD
44	RecordValue41 = CTR_COMP_RBM_OBD_LSH_DOWN_1 high byte	M	XXh	RV_CTR_COMP_RBM_OBD_LSH_DOWN_1
45	RecordValue42 = CTR_COMP_RBM_OBD_LSH_DOWN_1 low byte	M	XXh	RV_CTR_COMP_RBM_OBD_LSH_DOWN_1
46	RecordValue43 = CTR_CDN_RBM_OBD_LSH_DOWN_1 high byte	M	XXh	RV_CTR_CDN_RBM_OBD_LSH_DOWN_1
47	RecordValue44 = CTR_CDN_RBM_OBD_LSH_DOWN_1 low byte	M	XXh	RV_CTR_CDN_RBM_OBD_LSH_DOWN_1
48	RecordValue45 = CTR_COMP_RBM_OBD_LSH_DOWN_2 high byte	M	XXh	RV_CTR_COMP_RBM_OBD_LSH_DOWN_2
49	RecordValue46 = CTR_COMP_RBM_OBD_LSH_DOWN_2 low byte	M	XXh	RV_CTR_COMP_RBM_OBD_LSH_DOWN_2
50	RecordValue47 = CTR_CDN_RBM_OBD_LSH_DOWN_2 high byte	M	XXh	RV_CTR_CDN_RBM_OBD_LSH_DOWN_2
51	RecordValue48 = CTR_CDN_RBM_OBD_LSH_DOWN_2 low byte	M	XXh	RV_CTR_CDN_RBM_OBD_LSH_DOWN_2
52	RecordValue49 = CTR_COMP_RBM_OBD_VLD_LSH_UP_1 high byte	M	XXh	RV_CTR_COMP_RBM_OBD_VLD_LSH_UP_1
53	RecordValue50 = CTR_COMP_RBM_OBD_VLD_LSH_UP_1 low byte	M	XXh	RV_CTR_COMP_RBM_OBD_VLD_LSH_UP_1
54	RecordValue51 = CTR_CDN_RBM_OBD_VLD_LSH_UP_1 high byte	M	XXh	RV_CTR_CDN_RBM_OBD_VLD_LSH_UP_1
55	RecordValue52 = CTR_CDN_RBM_OBD_VLD_LSH_UP_1 low byte	M	XXh	RV_CTR_CDN_RBM_OBD_VLD_LSH_UP_1
56	RecordValue53 = CTR_COMP_RBM_OBD_VLD_LSH_UP_2 high byte	M	XXh	RV_CTR_COMP_RBM_OBD_VLD_LSH_UP_2
57	RecordValue54 = CTR_COMP_RBM_OBD_VLD_LSH_UP_2 low byte	M	XXh	RV_CTR_COMP_RBM_OBD_VLD_LSH_UP_2
58	RecordValue55 = CTR_CDN_RBM_OBD_VLD_LSH_UP_2 high byte	M	XXh	RV_CTR_CDN_RBM_OBD_VLD_LSH_UP_2

59	RecordValue56 = CTR_CDN_RBM_OBD_VLD_LSH_UP_2 low byte	M	XXh	RV_CTR_CDN_RBM_OBD_VLD_LSH_UP_2
60	RecordValue57 = CTR_COMP_RBM_CS high byte	M	XXh	RV_CTR_COMP_RBM_CS
61	RecordValue58 = CTR_COMP_RBM_CS low byte	M	XXh	RV_CTR_COMP_RBM_CS
62	RecordValue59 = CTR_CDN_RBM_CS high byte	M	XXh	RV_CTR_CDN_RBM_CS
63	RecordValue60 = CTR_CDN_RBM_CS low byte	M	XXh	RV_CTR_CDN_RBM_CS
64	RecordValue61 = CTR_COMP_RBM_ISC high byte	M	XXh	RV_CTR_COMP_RBM_ISC
65	RecordValue62 = CTR_COMP_RBM_ISC low byte	M	XXh	RV_CTR_COMP_RBM_ISC
66	RecordValue63 = CTR_CDN_RBM_ISC high byte	M	XXh	RV_CTR_CDN_RBM_ISC
67	RecordValue64 = CTR_CDN_RBM_ISC low byte	M	XXh	RV_CTR_CDN_RBM_ISC
68	RecordValue65 = CTR_COMP_RBM_TIA_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_TIA_PLAUS
69	RecordValue66 = CTR_COMP_RBM_TIA_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_TIA_PLAUS
70	RecordValue67 = CTR_CDN_RBM_TIA_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_TIA_PLAUS
71	RecordValue68 = CTR_CDN_RBM_TIA_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_TIA_PLAUS
72	RecordValue69 = CTR_COMP_RBM_AMP_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_AMP_PLAUS
73	RecordValue70 = CTR_COMP_RBM_AMP_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_AMP_PLAUS
74	RecordValue71 = CTR_CDN_RBM_AMP_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_AMP_PLAUS
75	RecordValue72 = CTR_CDN_RBM_AMP_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_AMP_PLAUS
76	RecordValue73 = CTR_COMP_RBM_LOAD_TPS_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_LOAD_TPS_PLAUS
77	RecordValue74 = CTR_COMP_RBM_LOAD_TPS_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_LOAD_TPS_PLAUS
78	RecordValue75 = CTR_CDN_RBM_LOAD_TPS_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_LOAD_TPS_PLAUS
79	RecordValue76 = CTR_CDN_RBM_LOAD_TPS_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_LOAD_TPS_PLAUS
80	RecordValue77 = CTR_COMP_RBM_MAP_DIP_PLAUS high byte	M	XXh	RV_CTR_COMP_RBM_MAP_DIP_PLAUS
81	RecordValue78 = CTR_COMP_RBM_MAP_DIP_PLAUS low byte	M	XXh	RV_CTR_COMP_RBM_MAP_DIP_PLAUS
82	RecordValue79 = CTR_CDN_RBM_MAP_DIP_PLAUS high byte	M	XXh	RV_CTR_CDN_RBM_MAP_DIP_PLAUS
83	RecordValue80 = CTR_CDN_RBM_MAP_DIP_PLAUS low byte	M	XXh	RV_CTR_CDN_RBM_MAP_DIP_PLAUS
84	RecordValue81 = CTR_COMP_RBM_VLS_DOWN_DIF_1 high byte	M	XXh	RV_CTR_COMP_RBM_VLS_DOWN_DIF_1
85	RecordValue82 = CTR_COMP_RBM_VLS_DOWN_DIF_1 low byte	M	XXh	RV_CTR_COMP_RBM_VLS_DOWN_DIF_1
86	RecordValue83 = CTR_CDN_RBM_VLS_DOWN_DIF_1 high byte	M	XXh	RV_CTR_CDN_RBM_VLS_DOWN_DIF_1
87	RecordValue84 = CTR_CDN_RBM_VLS_DOWN_DIF_1 low byte	M	XXh	RV_CTR_CDN_RBM_VLS_DOWN_DIF_1
88	RecordValue85 = CTR_COMP_RBM_VLS_DOWN_DIF_2 high byte	M	XXh	RV_CTR_COMP_RBM_VLS_DOWN_DIF_2
89	RecordValue86 = CTR_COMP_RBM_VLS_DOWN_DIF_2 low byte	M	XXh	RV_CTR_COMP_RBM_VLS_DOWN_DIF_2
90	RecordValue87 = CTR_CDN_RBM_VLS_DOWN_DIF_2 high byte	M	XXh	RV_CTR_CDN_RBM_VLS_DOWN_DIF_2
91	RecordValue88 = CTR_CDN_RBM_VLS_DOWN_DIF_2 low byte	M	XXh	RV_CTR_CDN_RBM_VLS_DOWN_DIF_2
92	RecordValue89 = CTR_COMP_RBM_CHK_LS_DOWN_1 high byte	M	XXh	RV_CTR_COMP_RBM_CHK_LS_DOWN_1
93	RecordValue90 = CTR_COMP_RBM_CHK_LS_DOWN_1 low byte	M	XXh	RV_CTR_COMP_RBM_CHK_LS_DOWN_1

94	RecordValue91 = CTR_CDN_RBM_CHK_LS_DOWN_1 high byte	M	XXh	RV_CTR_CDN_RBM_CHK_LS_DOWN_1
95	RecordValue92 = CTR_CDN_RBM_CHK_LS_DOWN_1 low byte	M	XXh	RV_CTR_CDN_RBM_CHK_LS_DOWN_1
96	RecordValue93 = CTR_COMP_RBM_CHK_LS_DOWN_2 high byte	M	XXh	RV_CTR_COMP_RBM_CHK_LS_DOWN_2
97	RecordValue94 = CTR_COMP_RBM_CHK_LS_DOWN_2 low byte	M	XXh	RV_CTR_COMP_RBM_CHK_LS_DOWN_2
98	RecordValue95 = CTR_CDN_RBM_CHK_LS_DOWN_2 high byte	M	XXh	RV_CTR_CDN_RBM_CHK_LS_DOWN_2
99	RecordValue96 = CTR_CDN_RBM_CHK_LS_DOWN_2 low byte	M	XXh	RV_CTR_CDN_RBM_CHK_LS_DOWN_2
100	RecordValue97 = CTR_COMP_RBM_SWT_LS_DOWN_1 high byte	M	XXh	RV_CTR_COMP_RBM_SWT_LS_DOWN_1
101	RecordValue98 = CTR_COMP_RBM_SWT_LS_DOWN_1 low byte	M	XXh	RV_CTR_COMP_RBM_SWT_LS_DOWN_1
102	RecordValue99 = CTR_CDN_RBM_SWT_LS_DOWN_1 high byte	M	XXh	RV_CTR_CDN_RBM_SWT_LS_DOWN_1
103	RecordValue100 = CTR_CDN_RBM_SWT_LS_DOWN_1 low byte	M	XXh	RV_CTR_CDN_RBM_SWT_LS_DOWN_1
104	RecordValue101 = CTR_COMP_RBM_SWT_LS_DOWN_2 high byte	M	XXh	RV_CTR_COMP_RBM_SWT_LS_DOWN_2
105	RecordValue102 = CTR_COMP_RBM_SWT_LS_DOWN_2 low byte	M	XXh	RV_CTR_COMP_RBM_SWT_LS_DOWN_2
106	RecordValue103 = CTR_CDN_RBM_SWT_LS_DOWN_2 high byte	M	XXh	RV_CTR_CDN_RBM_SWT_LS_DOWN_2
107	RecordValue104 = CTR_CDN_RBM_SWT_LS_DOWN_2 low byte	M	XXh	RV_CTR_CDN_RBM_SWT_LS_DOWN_2
108	RecordValue105 = CTR_COMP_RBM_T_ES high byte	M	XXh	RV_CTR_COMP_RBM_T_ES
109	RecordValue106 = CTR_COMP_RBM_T_ES low byte	M	XXh	RV_CTR_COMP_RBM_T_ES
110	RecordValue107 = CTR_CDN_RBM_T_ES high byte	M	XXh	RV_CTR_CDN_RBM_T_ES
111	RecordValue108 = CTR_CDN_RBM_T_ES low byte	M	XXh	RV_CTR_CDN_RBM_T_ES
112	RecordValue109 = CTR_COMP_RBM_TPS high byte	M	XXh	RV_CTR_COMP_RBM_TPS
113	RecordValue110 = CTR_COMP_RBM_TPS low byte	M	XXh	RV_CTR_COMP_RBM_TPS
114	RecordValue111 = CTR_CDN_RBM_TPS high byte	M	XXh	RV_CTR_CDN_RBM_TPS
115	RecordValue112 = CTR_CDN_RBM_TPS low byte	M	XXh	RV_CTR_CDN_RBM_TPS
116	RecordValue113 = CTR_COMP_RBM_ANG_INST_AD_VVL high byte	M	XXh	RV_CTR_COMP_RBM_ANG_INST_AD_VVL
117	RecordValue114 = CTR_COMP_RBM_ANG_INST_AD_VVL low byte	M	XXh	RV_CTR_COMP_RBM_ANG_INST_AD_VVL
118	RecordValue115 = CTR_CDN_RBM_ANG_INST_AD_VVL high byte	M	XXh	RV_CTR_CDN_RBM_ANG_INST_AD_VVL
119	RecordValue116 = CTR_CDN_RBM_ANG_INST_AD_VVL low byte	M	XXh	RV_CTR_CDN_RBM_ANG_INST_AD_VVL
120	RecordValue117 = CTR_COMP_RBM_ANG_CHK_MAX_VVL high byte	M	XXh	RV_CTR_COMP_RBM_ANG_CHK_MAX_VVL
121	RecordValue118 = CTR_COMP_RBM_ANG_CHK_MAX_VVL low byte	M	XXh	RV_CTR_COMP_RBM_ANG_CHK_MAX_VVL
122	RecordValue119 = CTR_CDN_RBM_ANG_CHK_MAX_VVL high byte	M	XXh	RV_CTR_CDN_RBM_ANG_CHK_MAX_VVL
123	RecordValue120 = CTR_CDN_RBM_ANG_CHK_MAX_VVL low byte	M	XXh	RV_CTR_CDN_RBM_ANG_CHK_MAX_VVL
124	RecordValue121 = CTR_COMP_RBM_ISC_CST high byte	M	XXh	RV_CTR_COMP_RBM_ISC_CST
125	RecordValue122 = CTR_COMP_RBM_ISC_CST low byte	M	XXh	RV_CTR_COMP_RBM_ISC_CST
126	RecordValue123 = CTR_CDN_RBM_ISC_CST high byte	M	XXh	RV_CTR_CDN_RBM_ISC_CST
127	RecordValue124 = CTR_CDN_RBM_ISC_CST low byte	M	XXh	RV_CTR_CDN_RBM_ISC_CST
128	RecordValue125 = CTR_COMP_RBM_TQ_CST high byte	M	XXh	RV_CTR_COMP_RBM_TQ_CST

129	RecordValue126 = CTR_COMP_RBM_TQ_CST low byte	M	XXh	RV_CTR_COMP_RBM_TQ_CST
130	RecordValue127 = CTR_CDN_RBM_TQ_CST high byte	M	XXh	RV_CTR_CDN_RBM_TQ_CST
131	RecordValue128 = CTR_CDN_RBM_TQ_CST low byte	M	XXh	RV_CTR_CDN_RBM_TQ_CST
132	RecordValue129 = CTR_COMP_RBM_EAC high byte	M	XXh	RV_CTR_COMP_RBM_EAC
133	RecordValue130 = CTR_COMP_RBM_EAC low byte	M	XXh	RV_CTR_COMP_RBM_EAC
134	RecordValue131 = CTR_CDN_RBM_EAC high byte	M	XXh	RV_CTR_CDN_RBM_EAC
135	RecordValue132 = CTR_CDN_RBM_EAC low byte	M	XXh	RV_CTR_CDN_RBM_EAC

7.29.60 22_402D - RCI_MWLRP_lesen

General information:

Messwerte Laufruhepruefung auslesen

This function reads out values from the engine roughness check.

7.29.60.1 MSD 85 only

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MWLRP
3	RecordCommonIdentifier Low Byte	M	2Dh	RCI_MWLRP

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	RDBCIR
2	RecordCommonIdentifier High Byte	M	40h	RCI_MWLRP
3	RecordCommonIdentifier Low Byte	M	2Dh	RCI_MWLRP
4	Not used	M	FFh	RV_DUMMY_1
5	Not used	M	FFh	RV_DUMMY_2
6	Not used	M	FFh	RV_DUMMY_3
7	Not used	M	FFh	RV_DUMMY_4
8	RecordValue5 = Amo_05 high byte	M	XXh	RV_AMO_05
9	RecordValue6 = Amo_05 low byte	M	XXh	RV_AMO_05
10	RecordValue7 = Amo_10high byte	M	XXh	RV_AMO_10
11	RecordValue8 = Amo_10low byte	M	XXh	RV_AMO_10
12	RecordValue9 = Amo_15 high byte	M	XXh	RV_AMO_15
13	RecordValue10 = Amo_15 low byte	M	XXh	RV_AMO_15
14	RecordValue11 = Amo_20 high byte	M	XXh	RV_AMO_20
15	RecordValue12 = Amo_20 low byte	M	XXh	RV_AMO_20
16	Not used	M	FFh	RV_DUMMY_5
17	Not used	M	FFh	RV_DUMMY_6
18	Not used	M	FFh	RV_DUMMY_7
19	Not used	M	FFh	RV_DUMMY_8
20	Not used	M	FFh	RV_DUMMY_9
21	Not used	M	FFh	RV_DUMMY_10
22	Not used	M	FFh	RV_DUMMY_11
23	Not used	M	FFh	RV_DUMMY_12
24	Not used	M	FFh	RV_DUMMY_13
25	Not used	M	FFh	RV_DUMMY_14
26	Not used	M	FFh	RV_DUMMY_15

M

27	Not used	M	FFh	RV_DUMMY_16
28	Not used	M	FFh	RV_DUMMY_17
29	Not used	M	FFh	RV_DUMMY_18
30	Not used	M	FFh	RV_DUMMY_19
31	Not used	M	FFh	RV_DUMMY_20
32	Not used	M	FFh	RV_DUMMY_21
33	Not used	M	FFh	RV_DUMMY_22
34	Not used	M	FFh	RV_DUMMY_23
35	RecordValue32 = Lurabs_f[0] high byte	M	XXh	RV_LURABS_F[0]
36	RecordValue33 = Lurabs_f[0] low byte	M	XXh	RV_LURABS_F[0]
37	RecordValue34 = Lurdif_f[0] high byte	M	XXh	RV_LURDIF_F[0]
38	RecordValue35 = Lurdif_f[0] low byte	M	XXh	RV_LURDIF_F[0]
39	RecordValue36 = Zw_offkorrvr[0] high byte	M	XXh	RV_ZW_OFFKORRVR[0]
40	RecordValue37 = Zw_offkorrvr[0] low byte	M	XXh	RV_ZW_OFFKORRVR[0]
41	Not used	M	FFh	RV_DUMMY_24
42	Not used	M	FFh	RV_DUMMY_25
43	RecordValue40 = Lurabs_f[1] high byte	M	XXh	RV_LURABS_F[1]
44	RecordValue41 = Lurabs_f[1] low byte	M	XXh	RV_LURABS_F[1]
45	RecordValue42 = Lurdif_f[1] high byte	M	XXh	RV_LURDIF_F[1]
46	RecordValue43 = Lurdif_f[1] low byte	M	XXh	RV_LURDIF_F[1]
47	RecordValue44 = Zw_offkorrvr[1] high byte	M	XXh	RV_ZW_OFFKORRVR[1]
48	RecordValue45 = Zw_offkorrvr[1] low byte	M	XXh	RV_ZW_OFFKORRVR[1]
49	Not used	M	FFh	RV_DUMMY_26
50	Not used	M	FFh	RV_DUMMY_27
51	RecordValue48 = Lurabs_f[2] high byte	M	XXh	RV_LURABS_F[2]
52	RecordValue49 = Lurabs_f[2] low byte	M	XXh	RV_LURABS_F[2]
53	RecordValue50 = Lurdif_f[2] high byte	M	XXh	RV_LURDIF_F[2]
54	RecordValue51 = Lurdif_f[2] low byte	M	XXh	RV_LURDIF_F[2]
55	RecordValue52 = Zw_offkorrvr[2] high byte	M	XXh	RV_ZW_OFFKORRVR[2]
56	RecordValue53 = Zw_offkorrvr[2] low byte	M	XXh	RV_ZW_OFFKORRVR[2]
57	Not used	M	FFh	RV_DUMMY_28
58	Not used	M	FFh	RV_DUMMY_29
59	RecordValue56 = Lurabs_f[3] high byte	M	XXh	RV_LURABS_F[3]
60	RecordValue57 = Lurabs_f[3] low byte	M	XXh	RV_LURABS_F[3]
61	RecordValue58 = Lurdif_f[3] high byte	M	XXh	RV_LURDIF_F[3]
62	RecordValue59 = Lurdif_f[3] low byte	M	XXh	RV_LURDIF_F[3]
63	RecordValue60 = Zw_offkorrvr[3] high byte	M	XXh	RV_ZW_OFFKORRVR[3]
64	RecordValue61 = Zw_offkorrvr[3] low byte	M	XXh	RV_ZW_OFFKORRVR[3]
65	Not used	M	FFh	RV_DUMMY_30
66	Not used	M	FFh	RV_DUMMY_31
67	RecordValue64 = Lurabs_f[4] high byte	M	XXh	RV_LURABS_F[4]
68	RecordValue65 = Lurabs_f[4] low byte	M	XXh	RV_LURABS_F[4]
69	RecordValue66 = Lurdif_f[4] high byte	M	XXh	RV_LURDIF_F[4]
70	RecordValue67 = Lurdif_f[4] low byte	M	XXh	RV_LURDIF_F[4]
71	RecordValue68 = Zw_offkorrvr[4] high byte	M	XXh	RV_ZW_OFFKORRVR[4]
72	RecordValue69 = Zw_offkorrvr[4] low byte	M	XXh	RV_ZW_OFFKORRVR[4

				]
73	Not used	M	FFh	RV_DUMMY_32
74	Not used	M	FFh	RV_DUMMY_33
75	RecordValue72 = Lurabs_f[5] high byte	M	XXh	RV_LURABS_F[5]
76	RecordValue73 = Lurabs_f[5] low byte	M	XXh	RV_LURABS_F[5]
77	RecordValue74 = Lurdif_f[5] high byte	M	XXh	RV_LURDIF_F[5]
78	RecordValue75 = Lurdif_f[5] low byte	M	XXh	RV_LURDIF_F[5]
79	RecordValue76 = Zw_offkorrvr[5] high byte	M	XXh	RV_ZW_OFFKORRVR[5]
80	RecordValue77 = Zw_offkorrvr[5] low byte	M	XXh	RV_ZW_OFFKORRVR[5]
81	Not used	M	FFh	RV_DUMMY_34
82	Not used	M	FFh	RV_DUMMY_35
83	RecordValue80 = Lurabs_f[6] high byte	M	XXh	RV_LURABS_F[6]
84	RecordValue81 = Lurabs_f[6] low byte	M	XXh	RV_LURABS_F[6]
85	RecordValue82 = Lurdif_f[6] high byte	M	XXh	RV_LURDIF_F[6]
86	RecordValue83 = Lurdif_f[6] low byte	M	XXh	RV_LURDIF_F[6]
87	RecordValue84 = Zw_offkorrvr[6] high byte	M	XXh	RV_ZW_OFFKORRVR[6]
88	RecordValue85 = Zw_offkorrvr[6] low byte	M	XXh	RV_ZW_OFFKORRVR[6]
89	Not used	M	FFh	RV_DUMMY_36
90	Not used	M	FFh	RV_DUMMY_37
91	RecordValue88 = Lurabs_f[7] high byte	M	XXh	RV_LURABS_F[7]
92	RecordValue89 = Lurabs_f[7] low byte	M	XXh	RV_LURABS_F[7]
93	RecordValue90 = Lurdif_f[7] high byte	M	XXh	RV_LURDIF_F[7]
94	RecordValue91 = Lurdif_f[7] low byte	M	XXh	RV_LURDIF_F[7]
95	RecordValue92 = Zw_offkorrvr[7] high byte	M	XXh	RV_ZW_OFFKORRVR[7]
96	RecordValue93 = Zw_offkorrvr[7] low byte	M	XXh	RV_ZW_OFFKORRVR[7]
97	RecordValue94 = Amo_30 high byte	M	XXh	RV_AMO_30
98	RecordValue95 = Amo_30 low byte	M	XXh	RV_AMO_30
99	RecordValue96 = Amo_40 high byte	M	XXh	RV_AMO_40
100	RecordValue97 = Amo_40 low byte	M	XXh	RV_AMO_40

Hint: all record values which are not defined in BMW-layer.h are FFh,
also RecordValue 65d ... 93d for 4cyl-machine,
RecordValue 78d ... 93d for 6cyl-machine

7.29.60.2 All other projects

General information:

Messwerte Laufruhepruefung auslesen

This function reads out values from the engine roughness check.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MWLRP
3	RecordCommonIdentifier Low Byte	M	2Dh	RCI_MWLRP

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	RDBCIR
2	RecordCommonIdentifier High Byte	M	40h	RCI_MWLRP
3	RecordCommonIdentifier Low Byte	M	2Dh	RCI_MWLRP
4	RecordValue1 = St_vbrvs_aus high byte	M	XXh	RV_ST_VBRVS_AUS
5	RecordValue2 = St_vbrvs_aus low byte	M	XXh	RV_ST_VBRVS_AUS
6	RecordValue3 = St_vbrvs_ein	M	XXh	RV_ST_VBRVS_EIN
7	RecordValue4 = dummy value fix	M	FFh	RV_ST_VBRVS_EINNV
8	RecordValue5 = Amo_05 high byte	M	XXh	RV_AMO_05
9	RecordValue6 = Amo_05 low byte	M	XXh	RV_AMO_05
10	RecordValue7 = Amo_10high byte	M	XXh	RV_AMO_10
11	RecordValue8 = Amo_10low byte	M	XXh	RV_AMO_10
12	RecordValue9 = Amo_15 high byte	M	XXh	RV_AMO_15
13	RecordValue10 = Amo_15 low byte	M	XXh	RV_AMO_15
14	RecordValue11 = Amo_20 high byte	M	XXh	RV_AMO_20
15	RecordValue12 = Amo_20 low byte	M	XXh	RV_AMO_20
16	RecordValue13 = [Exwinkkor high byte **(only MSV80),** FFh **(other)]**	M	XXh	RV_EXWINKKOR
17	RecordValue14 = [Exwinkkor low byte **(only MSV80),** FFh **(other)]**	M	XXh	RV_EXWINKKOR
18	RecordValue15 = Zylhubkor	M	XXh	RV_ZYLHUBKOR
19	RecordValue16 =[Minhub high byte **(only MSV80),** FFh **(other)]**	M	XXh	RV_MINHUB
20	RecordValue17 = [Minhub low byte **(only MSV80),** FFh **(other**)]	M	XXh	RV_MINHUB
21	RecordValue18 = F_minhub high byte	M	XXh	RV_F_MINHUB
22	RecordValue19 = F_minhub low byte	M	XXh	RV_F_MINHUB
23	RecordValue20 = Minhub_roh high byte	M	XXh	RV_MINHUB_ROH
24	RecordValue21 = Minhub_roh low byte	M	XXh	RV_MINHUB_ROH
25	RecordValue22 = Minhubvs high byte	M	XXh	RV_MINHUBVS
26	RecordValue23 = Minhubvs low byte	M	XXh	RV_MINHUBVS
27	RecordValue24 = Minhubvs_ist high byte	M	XXh	RV_MINHUBVS_IST
28	RecordValue25 = Minhubvs_ist low byte	M	XXh	RV_MINHUBVS_IST
29	RecordValue26 = Minhubvsnv high byte	M	XXh	RV_MINHUBVSNV
30	RecordValue27 = Minhubvsnv low byte	M	XXh	RV_MINHUBVSNV
31	RecordValue28 = S_vsmnhb	M	XXh	RV_S_VSMNHB
32	RecordValue29 = S_vsmnhbnv	M	XXh	RV_S_VSMNHBNV
33	RecordValue30 = F_tikorrvr[0] high byte	M	XXh	RV_F_TIKORRVR[0]
34	RecordValue31 = F_tikorrvr[0] low byte	M	XXh	RV_F_TIKORRVR[0]
35	RecordValue32 = Lurabs_f[0] high byte	M	XXh	RV_LURABS_F[0]
36	RecordValue33 = Lurabs_f[0] low byte	M	XXh	RV_LURABS_F[0]
37	RecordValue34 = Lurdif_f[0] high byte	M	XXh	RV_LURDIF_F[0]
38	RecordValue35 = Lurdif_f[0] low byte	M	XXh	RV_LURDIF_F[0]
39	RecordValue36 = Zw_offkorrvr[0] high byte	M	XXh	RV_ZW_OFFKORRVR[0]
40	RecordValue37 = Zw_offkorrvr[0] low byte	M	XXh	RV_ZW_OFFKORRVR[0]
41	RecordValue38 = F_tikorrvr[1] high byte	M	XXh	RV_F_TIKORRVR[1]
42	RecordValue39 = F_tikorrvr[1] low byte	M	XXh	RV_F_TIKORRVR[1]
43	RecordValue40 = Lurabs_f[1] high byte	M	XXh	RV_LURABS_F[1]
44	RecordValue41 = Lurabs_f[1] low byte	M	XXh	RV_LURABS_F[1]
45	RecordValue42 = Lurdif_f[1] high byte	M	XXh	RV_LURDIF_F[1]
46	RecordValue43 = Lurdif_f[1] low byte	M	XXh	RV_LURDIF_F[1]

47	RecordValue44 = Zw_offkorrvr[1] high byte	M	XXh	RV_ZW_OFFKORRVR[1]
48	RecordValue45 = Zw_offkorrvr[1] low byte	M	XXh	RV_ZW_OFFKORRVR[1]
49	RecordValue46 = F_tikorrvr[2] high byte	M	XXh	RV_F_TIKORRVR[2]
50	RecordValue47 = F_tikorrvr[2] low byte	M	XXh	RV_F_TIKORRVR[2]
51	RecordValue48 = Lurabs_f[2] high byte	M	XXh	RV_LURABS_F[2]
52	RecordValue49 = Lurabs_f[2] low byte	M	XXh	RV_LURABS_F[2]
53	RecordValue50 = Lurdif_f[2] high byte	M	XXh	RV_LURDIF_F[2]
54	RecordValue51 = Lurdif_f[2] low byte	M	XXh	RV_LURDIF_F[2]
55	RecordValue52 = Zw_offkorrvr[2] high byte	M	XXh	RV_ZW_OFFKORRVR[2]
56	RecordValue53 = Zw_offkorrvr[2] low byte	M	XXh	RV_ZW_OFFKORRVR[2]
57	RecordValue54 = F_tikorrvr[3] high byte	M	XXh	RV_F_TIKORRVR[3]
58	RecordValue55 = F_tikorrvr[3] low byte	M	XXh	RV_F_TIKORRVR[3]
59	RecordValue56 = Lurabs_f[3] high byte	M	XXh	RV_LURABS_F[3]
60	RecordValue57 = Lurabs_f[3] low byte	M	XXh	RV_LURABS_F[3]
61	RecordValue58 = Lurdif_f[3] high byte	M	XXh	RV_LURDIF_F[3]
62	RecordValue59 = Lurdif_f[3] low byte	M	XXh	RV_LURDIF_F[3]
63	RecordValue60 = Zw_offkorrvr[3] high byte	M	XXh	RV_ZW_OFFKORRVR[3]
64	RecordValue61 = Zw_offkorrvr[3] low byte	M	XXh	RV_ZW_OFFKORRVR[3]
65	RecordValue62 = F_tikorrvr[4] high byte	M	XXh	RV_F_TIKORRVR[4]
66	RecordValue63 = F_tikorrvr[4] low byte	M	XXh	RV_F_TIKORRVR[4]
67	RecordValue64 = Lurabs_f[4] high byte	M	XXh	RV_LURABS_F[4]
68	RecordValue65 = Lurabs_f[4] low byte	M	XXh	RV_LURABS_F[4]
69	RecordValue66 = Lurdif_f[4] high byte	M	XXh	RV_LURDIF_F[4]
70	RecordValue67 = Lurdif_f[4] low byte	M	XXh	RV_LURDIF_F[4]
71	RecordValue68 = Zw_offkorrvr[4] high byte	M	XXh	RV_ZW_OFFKORRVR[4]
72	RecordValue69 = Zw_offkorrvr[4] low byte	M	XXh	RV_ZW_OFFKORRVR[4]
73	RecordValue70 = F_tikorrvr[5] high byte	M	XXh	RV_F_TIKORRVR[5]
74	RecordValue71 = F_tikorrvr[5] low byte	M	XXh	RV_F_TIKORRVR[5]
75	RecordValue72 = Lurabs_f[5] high byte	M	XXh	RV_LURABS_F[5]
76	RecordValue73 = Lurabs_f[5] low byte	M	XXh	RV_LURABS_F[5]
77	RecordValue74 = Lurdif_f[5] high byte	M	XXh	RV_LURDIF_F[5]
78	RecordValue75 = Lurdif_f[5] low byte	M	XXh	RV_LURDIF_F[5]
79	RecordValue76 = Zw_offkorrvr[5] high byte	M	XXh	RV_ZW_OFFKORRVR[5]
80	RecordValue77 = Zw_offkorrvr[5] low byte	M	XXh	RV_ZW_OFFKORRVR[5]
81	RecordValue78 = F_tikorrvr[6] high byte	M	XXh	RV_F_TIKORRVR[6]
82	RecordValue79 = F_tikorrvr[6] low byte	M	XXh	RV_F_TIKORRVR[6]
83	RecordValue80 = Lurabs_f[6] high byte	M	XXh	RV_LURABS_F[6]
84	RecordValue81 = Lurabs_f[6] low byte	M	XXh	RV_LURABS_F[6]
85	RecordValue82 = Lurdif_f[6] high byte	M	XXh	RV_LURDIF_F[6]
86	RecordValue83 = Lurdif_f[6] low byte	M	XXh	RV_LURDIF_F[6]
87	RecordValue84 = Zw_offkorrvr[6] high byte	M	XXh	RV_ZW_OFFKORRVR[6]
88	RecordValue85 = Zw_offkorrvr[6] low byte	M	XXh	RV_ZW_OFFKORRVR[6]
89	RecordValue86 = F_tikorrvr[7] high byte	M	XXh	RV_F_TIKORRVR[7]

90	RecordValue87 = F_tikorrvr[7] low byte	M	XXh	RV_F_TIKORRVR[7]
91	RecordValue88 = Lurabs_f[7] high byte	M	XXh	RV_LURABS_F[7]
92	RecordValue89 = Lurabs_f[7] low byte	M	XXh	RV_LURABS_F[7]
93	RecordValue90 = Lurdif_f[7] high byte	M	XXh	RV_LURDIF_F[7]
94	RecordValue91 = Lurdif_f[7] low byte	M	XXh	RV_LURDIF_F[7]
95	RecordValue92 = Zw_offkorrvr[7] high byte	M	XXh	RV_ZW_OFFKORRVR[7]
96	RecordValue93 = Zw_offkorrvr[7] low byte	M	XXh	RV_ZW_OFFKORRVR[7]
97	RecordValue94 = Amo_30 high byte	M	XXh	RV_AMO_30
98	RecordValue95 = Amo_30 low byte	M	XXh	RV_AMO_30
99	RecordValue96 = Amo_40 high byte	M	XXh	RV_AMO_40
100	RecordValue97 = Amo_40 low byte	M	XXh	RV_AMO_40

Hint: all record values which are not defined in BMW-layer.h are FFh,
also RecordValue 65d ... 93d for 4cyl-machine,
RecordValue 78d ... 93d for 6cyl-machine

7.29.61 22_402E - Nullgang-Erkennung lesen

General information:

This function is created to read out the info about "Nullgang Erkennung"

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Request Service ID	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	40h	RCI_NGE
3	RecordCommonIdentifier Low Byte	M	2Eh	RCI_NGE

Table Request Message

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_NGE
3	RecordCommonIdentifier Low Byte	M	2Eh	RCI_NGE
4	recordValue1 = Tvngang (high byte)	M	XXh	RV_PHY_NG_POSITION
5	recordValue2 = Tvngang (low byte)	M	XXh	RV_PHY_NG_POSITION
6	recordValue3 = Tvneutral (high byte)	M	XXh	RV_PHY_NG_LERN
7	recordValue4 = Tvneutral (low byte)	M	XXh	RV_PHY_NG_LERN
8	recordValue5 = 0 - if B_gangnull = FALSE 1 - if B_gangnull = TRUE	M	XXh	RV_STAT_NG_ERKANNT
9	recordValue6 = 0 - if B_nggelernt = 0 1 - if B_nggelernt = 1	M	XXh	RV_STAT_NG_GELERNT
10	recordValue7 = 0 - if B_kupp_ext = FALSE 1 - if B_kupp_ext = TRUE	M	XXh	RV_STATKUP_BETAETIGT
11	recordValue8 = 0 - if B_kwdreht = FALSE 1 - if B_kwdreht = TRUE	M	XXh	RV_STAT_MOT_DREHT
12	recordValue9 = 0 - if B_ngimlf = FALSE 1 - if B_ngimlf = TRUE	M	XXh	RV_STAT_NG_IM_LF

7.29.62 22_402F - MSA lesen

General information:

This function is created to read out the state of "MotorStopAutomatik"

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Request Service ID	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	40h	RCI_MSA
3	RecordCommonIdentifier Low Byte	M	2Fh	RCI_MSA

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	40h	RCI_MSA
3	RecordCommonIdentifier Low Byte	M	2Fh	RCI_MSA
4	recordValue1 = Stmsa (high byte, high word)	M	XXh	RV_STAT_MSA
5	recordValue2 = Stmsa (low byte, high word)	M	XXh	RV_STAT_MSA
6	recordValue3 = Stmsa (high byte, low word)	M	XXh	RV_STAT_MSA
7	recordValue4 = Stmsa (low byte, low word)	M	XXh	RV_STAT_MSA
8	recordValue5 = Stmsaav (high byte, high word)	M	XXh	RV_STAT_MSAAV
9	recordValue6 = Stmsaav (low byte, high word)	M	XXh	RV_STAT_MSAAV
10	recordValue7 = Stmsaav (high byte, low word)	M	XXh	RV_STAT_MSAAV
11	recordValue8 = Stmsaav (low byte, low word)	M	XXh	RV_STAT_MSAAV
12	recordValue9 = Stmsaev (high byte, high word)	M	XXh	RV_STAT_MSAEVV
13	recordValue10 = Stmsaev (low byte, high word)	M	XXh	RV_STAT_MSAEVV
14	recordValue11 = Stmsaev (high byte, low word)	M	XXh	RV_STAT_MSAEVV
15	recordValue12 = Stmsaev (low byte, low word)	M	XXh	RV_STAT_MSAEVV
16	recordValue13 = Stmsaaa (high byte, high word)	M	XXh	RV_STAT_MSAAA
17	recordValue14 = Stmsaaa (low byte, high word)	M	XXh	RV_STAT_MSAAA
18	recordValue15 = Stmsaaa (high byte, low word)	M	XXh	RV_STAT_MSAAA
19	recordValue16 = Stmsaaa (low byte, low word)	M	XXh	RV_STAT_MSAAA
20	recordValue17 = Stmsaea (high byte, high word)	M	XXh	RV_STAT_MSAEA
21	recordValue18 = Stmsaea (low byte, high word)	M	XXh	RV_STAT_MSAEA
22	recordValue19 = Stmsaea (high byte, low word)	M	XXh	RV_STAT_MSAEA
23	recordValue20 = Stmsaea (low byte, low word)	M	XXh	RV_STAT_MSAEA

7.29.63 22_4030 - FASTA_LVS_lesen

General information:

This function ist for reading the data of the FASTA_LVS (Fahrzeugnutzungsprofil)

Formula section:

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	40h	RCI_LVS
3	RecordCommonIdentifier	M	30h	RCI_LVS

Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	40h	RCI_LVS
3	RecordCommonIdentifier	M	30h	RCI_LVS
4	RecordValue1 = Zr_lvs_0 (high byte)	M	XXh	RV_ZR_LVS_0
5	RecordValue2 = Zr_lvs_0 (low byte)	M	XXh	RV_ ZR_LVS_0
6	RecordValue3 = Zr_lvs_1 (high byte)	M	XXh	RV_ ZR_LVS_1
7	RecordValue4 = Zr_lvs_1 (low byte)	M	XXh	RV_ ZR_LVS_1
8	RecordValue5 = Zr_lvs_2(high byte)	M	XXh	RV_ ZR_LVS_2
9	RecordValue6 = Zr_lvs_2(low byte)	M	XXh	RV_ ZR_LVS_2
10	RecordValue7 = Zr_lvs_3 (high byte)	M	XXh	RV_ ZR_LVS_3
11	RecordValue8 = Zr_lvs_3 (low byte)	M	XXh	RV_ ZR_LVS_3
12	RecordValue9 = Zr_lvssekt_0 (high byte)	M	XXh	RV_ZR_LVSSEKT_0
13	RecordValue10= Zr_lvssekt_0 (low byte)	M	XXh	RV_ZR_LVSSEKT_0
14	RecordValue11 = Zr_lvssekt_1 (high byte)	M	XXh	RV_ZR_LVSSEKT_1
15	RecordValue12 = Zr_lvssekt_1 (low byte)	M	XXh	RV_ZR_LVSSEKT_1
16	RecordValue13 = Zr_lvssekt_2 (high byte)	M	XXh	RV_ZR_LVSSEKT_2
17	RecordValue14 = Zr_lvssekt_2 (low byte)	M	XXh	RV_ZR_LVSSEKT_2
18	RecordValue15 = Zr_lvssekt_3 (high byte)	M	XXh	RV_ZR_LVSSEKT_3
19	RecordValue16 = Zr_lvssekt_3 (low byte)	M	XXh	RV_ZR_LVSSEKT_3
20	RecordValue17 = Zr_lvssekt_4 (high byte)	M	XXh	RV_ZR_LVSSEKT_4
21	RecordValue18 = Zr_lvssekt_4 (low byte)	M	XXh	RV_ZR_LVSSEKT_4
22	RecordValue19 = Zr_lvssekt_5 (high byte)	M	XXh	RV_ZR_LVSSEKT_5
23	RecordValue20 = Zr_lvssekt_5 (low byte)	M	XXh	RV_ZR_LVSSEKT_5
24	RecordValue21 = Zr_lvssekt_6 (high byte)	M	XXh	RV_ZR_LVSSEKT_6
25	RecordValue22 = Zr_lvssekt_6 (low byte)	M	XXh	RV_ZR_LVSSEKT_6
26	RecordValue23 = Zr_lvssekt_7 (high byte)	M	XXh	RV_ZR_LVSSEKT_7
27	RecordValue24 = Zr_lvssekt_7 (low byte)	M	XXh	RV_ZR_LVSSEKT_7
28	RecordValue25 = Zr_lvssekt_8 (high byte)	M	XXh	RV_ZR_LVSSEKT_8
29	RecordValue26 = Zr_lvssekt_8 (low byte)	M	XXh	RV_ZR_LVSSEKT_8
30	RecordValue27 = Zr_lvs_ll_reakt	M	XXh	RV_ZR_LVS_LL_REAKT

7.29.64 22_4031 - LVSZYL_lesen

General information:

This function ist for reading the data of the LVSZYL (LaufruheVerbesserungssystem Zylinderstatistik)

Formula section:

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	40h	RCI_LVSZYL
3	RecordCommonIdentifier	M	31h	RCI_LVSZYL

Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	40h	RCI_LVSZYL
3	RecordCommonIdentifier	M	31h	RCI_LVSZYL
4	RecordValue1 = Zr_lvszyl_0 (high byte)	M	XXh	RV_STAT_ZR_LVSZYL_0
5	RecordValue2 = Zr_lvszyl_0 (low byte)	M	XXh	RV_STAT_ZR_LVSZYL_0
6	RecordValue3 = Zr_lvszyl_1 (high byte)	M	XXh	RV_STAT_ZR_LVSZYL_1
7	RecordValue4 = Zr_lvszyl_1 (low byte)	M	XXh	RV_STAT_ZR_LVSZYL_1
8	RecordValue5 = Zr_lvszyl_2(high byte)	M	XXh	RV_STAT_ZR_LVSZYL_2
9	RecordValue6 = Zr_lvszyl_2(low byte)	M	XXh	RV_STAT_ZR_LVSZYL_2
10	RecordValue7 = Zr_lvszyl_3 (high byte)	M	XXh	RV_STAT_ZR_LVSZYL_3
11	RecordValue8 = Zr_lvszyl_3 (low byte)	M	XXh	RV_STAT_ZR_LVSZYL_3
12	RecordValue9 = Zr_lvszyl_4(high byte)	M	XXh	RV_STAT_ZR_LVSZYL_4
13	RecordValue10 = Zr_lvszyl_4(low byte)	M	XXh	RV_STAT_ZR_LVSZYL_4
14	RecordValue11= Zr_lvszyl_5 (high byte)	M	XXh	RV_STAT_ZR_LVSZYL_5
15	RecordValue12 = Zr_lvszyl_5 (low byte)	M	XXh	RV_STAT_ZR_LVSZYL_5

7.29.65 22_4032 - MFMA_lesen

General information:

This function ist for reading the data of MFMA (Kleinstmengenadaption)

Formula section:

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	40h	RCI_MFMA
3	RecordCommonIdentifier	M	32h	RCI_MFMA

Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service Id**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	40h	RCI_MFMA
3	RecordCommonIdentifier	M	32h	RCI_MFMA
4	RecordValue1 = CTR_AD_COLD_LAM_AD_INJ	M	XXh	RV_ CTR_AD_COLD_LAM_AD_INJ
5	RecordValue2 = CTR_AD_HOT_LAM_AD_INJ	M	XXh	RV_ CTR_AD_HOT_LAM_AD_INJ

6	RecordValue3 = DIST_LAM_AD_INJ_COLD (high byte)	M	XXh	RV_ DIST_LAM_AD_INJ_COLD
7	RecordValue4 = DIST_LAM_AD_INJ_COLD (low byte)	M	XXh	RV_ DIST_LAM_AD_INJ_COLD
8	RecordValue5 = DIST_LAM_AD_INJ_HOT (high byte)	M	XXh	RV_ DIST_LAM_AD_INJ_HOT
9	RecordValue6 = DIST_LAM_AD_INJ_HOT (low byte)	M	XXh	RV_ DIST_LAM_AD_INJ_HOT
10	RecordValue7 = FAC_LAM_ADJ_COR_LAM_AD_CUS[1] (high byte)	M	XXh	RV_ FAC_LAM_ADJ_COR_LAM_AD_CUS[1]
11	RecordValue8 = FAC_LAM_ADJ_COR_LAM_AD_CUS[1] (low byte)	M	XXh	RV_ FAC_LAM_ADJ_COR_LAM_AD_CUS[1]
12	RecordValue9 = FAC_LAM_ADJ_COR_LAM_AD_CUS[2] (high byte)	M	XXh	RV_ FAC_LAM_ADJ_COR_LAM_AD_CUS[2]
13	RecordValue10 =FAC_LAM_ADJ_COR_LAM_AD_CUS[2] (high byte)	M	XXh	RV_ FAC_LAM_ADJ_COR_LAM_AD_CUS[2]
14	RecordValue11 = FAC_LAM_ADJ_LAM_AD[1] (high byte)	M	XXh	RV_ FAC_LAM_ADJ_LAM_AD[1]
15	RecordValue12 = FAC_LAM_ADJ_LAM_AD[1] (low byte)	M	XXh	RV_ FAC_LAM_ADJ_LAM_AD[1]
16	RecordValue13 = FAC_LAM_ADJ_LAM_AD[2] (high byte)	M	XXh	RV_ FAC_LAM_ADJ_LAM_AD[2]
17	RecordValue14 = FAC_LAM_ADJ_LAM_AD[2] (low byte)	M	XXh	RV_ FAC_LAM_ADJ_LAM_AD[2]
18	RecordValue15 = MFF_ADD_COLD_LAM_AD_INJ[0] (high byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[0]
19	RecordValue16 = MFF_ADD_COLD_LAM_AD_INJ[0] (low byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[0]
20	RecordValue17 = MFF_ADD_HOT_LAM_AD_INJ[0] (high byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[0]
21	RecordValue18 = MFF_ADD_HOT_LAM_AD_INJ[0] (low byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[0]
22	RecordValue19 = MFF_ADD_COLD_LAM_AD_INJ[1] (high byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[1]
23	RecordValue20 = MFF_ADD_COLD_LAM_AD_INJ1] (low byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[1]
24	RecordValue 21 = MFF_ADD_HOT_LAM_AD_INJ[1] (high byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[1]
25	RecordValue22 = MFF_ADD_HOT_LAM_AD_INJ[1] (low byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[1]
26	RecordValue23 = MFF_ADD_COLD_LAM_AD_INJ[2] (high byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[2]
27	RecordValue24 = MFF_ADD_COLD_LAM_AD_INJ[2] (low byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[2]
28	RecordValue25 = MFF_ADD_HOT_LAM_AD_INJ[2] (high byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[2]
29	RecordValue26 = MFF_ADD_HOT_LAM_AD_INJ[2] (low byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[2]
30	RecordValue27 = MFF_ADD_COLD_LAM_AD_INJ[3] (high byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[3]
31	RecordValue28 = MFF_ADD_COLD_LAM_AD_INJ[3] (low byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[3]
32	RecordValue29 = MFF_ADD_HOT_LAM_AD_INJ[3] (high byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[3]
33	RecordValue30 = MFF_ADD_HOT_LAM_AD_INJ[3] (low byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[3]
34	RecordValue31 = MFF_ADD_COLD_LAM_AD_INJ[4] (high byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[4]
35	RecordValue32 = MFF_ADD_COLD_LAM_AD_INJ[4] (low byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[4]
36	RecordValue33 = MFF_ADD_HOT_LAM_AD_INJ[4] (high byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[4]
37	RecordValue34 = MFF_ADD_HOT_LAM_AD_INJ[4] (low byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[4]
38	RecordValue35 = MFF_ADD_COLD_LAM_AD_INJ[5] (high byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[5]

39	RecordValue36 = MFF_ADD_COLD_LAM_AD_INJ[5] (low byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[5]
40	RecordValue37 = MFF_ADD_HOT_LAM_AD_INJ[5] (high byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[5]
41	RecordValue38 = MFF_ADD_HOT_LAM_AD_INJ[5] (low byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[5]
42	RecordValue39 = MFF_ADD_COLD_LAM_AD_INJ[6] (high byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[6]
43	RecordValue40 = MFF_ADD_COLD_LAM_AD_INJ[6] (low byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[6]
44	RecordValue41 = MFF_ADD_HOT_LAM_AD_INJ[6] (high byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[6]
45	RecordValue42 = MFF_ADD_HOT_LAM_AD_INJ[6] (low byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[6]
46	RecordValue43 = MFF_ADD_COLD_LAM_AD_INJ[7] (high byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[7]
47	RecordValue44 = MFF_ADD_COLD_LAM_AD_INJ[7] (low byte)	M	XXh	RV_ MFF_ADD_COLD_LAM_AD_INJ[7]
48	RecordValue45 = MFF_ADD_HOT_LAM_AD_INJ[7] (high byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[7]
49	RecordValue46 = MFF_ADD_HOT_LAM_AD_INJ[7] (low byte)	M	XXh	RV_ MFF_ADD_HOT_LAM_AD_INJ[7]

Hint: transmitte 0ffh for variables which are not exist in project

7.29.66 22_5F88 - ADRECOVERY_lesen

General information:

This command read out the informations about the Nox-trap from ECU. This is necessary before reprogramming or change of ECU.

Formula section:

```
If      service 22_5F_88 is received
then
        send positive response
endif
```

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier	M	5Fh	RCI_ADRECOVERY
3	RecordCommonIdentifier	M	88h	RCI_ADRECOVERY

Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier	M	5Fh	RCI_ADRECOVERY
3	recordCommonIdentifier	M	88h	RCI_ADRECOVERY
4	recordValue1 = NT_SUL_32[1] (high word, high byte)	M	XXh	RV_NT_SUL_32[1]

5	recordValue2 = NT_SUL_32[1] (high word, low byte)	M	XXh	RV_NT_SUL_32[1]
6	recordValue3 = NT_SUL_32[1] (low word, high byte)	M	XXh	RV_NT_SUL_32[1]
7	recordValue4 = NT_SUL_32[1] (low word, low byte)	M	XXh	RV_NT_SUL_32[1]
8	recordValue5 = NT_SUL_32[2] (high word, high byte)	M	XXh	RV_NT_SUL_32[2]
9	recordValue6 = NT_SUL_32[2] (high word, low byte)	M	XXh	RV_NT_SUL_32[2]
10	recordValue7 = NT_SUL_32[2] (low word, high byte)	M	XXh	RV_NT_SUL_32[2]
11	recordValue8 = NT_SUL_32[2] (low word, low byte)	M	XXh	RV_NT_SUL_32[2]
12	recordValue9 = NT_SUL_H_32[1] (high word, high byte)	M	XXh	RV_NT_SUL_H_32[1]
13	recordValue10 = NT_SUL_H_32[1] (high word, low byte)	M	XXh	RV_NT_SUL_H_32[1]
14	recordValue11 = NT_SUL_H_32[1] (low word, high byte)	M	XXh	RV_NT_SUL_H_32[1]
15	recordValue12 = NT_SUL_H_32[1] (low word, low byte)	M	XXh	RV_NT_SUL_H_32[1]
16	recordValue13 = NT_SUL_H_32[2] (high word, high byte)	M	XXh	RV_NT_SUL_H_32[2]
17	recordValue14 = NT_SUL_H_32[2] (high word, low byte)	M	XXh	RV_NT_SUL_H_32[2]
18	recordValue15 = NT_SUL_H_32[2] (low word, high byte)	M	XXh	RV_NT_SUL_H_32[2]
19	recordValue16 = NT_SUL_H_32[2] (low word, low byte)	M	XXh	RV_NT_SUL_H_32[2]
10	recordValue17 = NT_AGI (high byte)	M	XXh	RV_NT_AGI
21	recordValue18 = NT_AGI (low byte)	M	XXh	RV_NT_AGI
22	recordValue19 = NT_AGI_SO2P_FQ_SUM (high word, high byte)	M	XXh	RV_NT_AGI_SO2P_FQ_SUM
23	recordValue20 = NT_AGI_SO2P_FQ_SUM (high word, low byte)	M	XXh	RV_NT_AGI_SO2P_FQ_SUM
24	recordValue21 = NT_AGI_SO2P_FQ_SUM (low word, high byte)	M	XXh	RV_NT_AGI_SO2P_FQ_SUM
25	recordValue22 = NT_AGI_SO2P_FQ_SUM (low word, low byte)	M	XXh	RV_NT_AGI_SO2P_FQ_SUM
26	recordValue23 = NT_AGI_SUL (high byte)	M	XXh	RV_NT_AGI_SUL
27	recordValue24 = NT_AGI_SUL (low byte)	M	XXh	RV_NT_AGI_SUL
28	recordValue25 = NT_AGI_SUL_SNG[1] high word, high byte)	M	XXh	RV_NT_AGI_SUL_SNG[1]
29	recordValue26 = NT_AGI_SUL_SNG[1] (high word, low byte)	M	XXh	RV_NT_AGI_SUL_SNG[1]
30	recordValue27 = NT_AGI_SUL_SNG[1] (low word, high byte)	M	XXh	RV_NT_AGI_SUL_SNG[1]
31	recordValue28 = NT_AGI_SUL_SNG[1] (low word, low byte)	M	XXh	RV_NT_AGI_SUL_SNG[1]
32	recordValue29 = NT_AGI_SUL_SNG[2](high word, high byte)	M	XXh	RV_NT_AGI_SUL_SNG[2]
33	recordValue30 = NT_AGI_SUL_SNG[2] (high word, low byte)	M	XXh	RV_NT_AGI_SUL_SNG[2]
34	recordValue31 = NT_AGI_SUL_SNG[2] (low word, high byte)	M	XXh	RV_NT_AGI_SUL_SNG[2]
35	recordValue32 = NT_AGI_SUL_SNG[2] (low word, low byte)	M	XXh	RV_NT_AGI_SUL_SNG[2]
36	recordValue33 = NT_AGI_THERMO (high byte)	M	XXh	RV_NT_AGI_THERMO
37	recordValue34 = NT_AGI_THERMO (low byte)	M	XXh	RV_NT_AGI_THERMO
38	recordValue35 = NT_AGI_THERMO_SNG[1] (high byte)	M	XXh	RV_NT_AGI_THERMO_SNG[1]
39	recordValue36 = NT_AGI_THERMO_SNG[1] (high byte)	M	XXh	RV_NT_AGI_THERMO_SNG[1]
40	recordValue37 = NT_AGI_THERMO_SNG[2] (high byte)	M	XXh	RV_NT_AGI_THERMO_SNG[2]
41	recordValue38 = NT_AGI_THERMO_SNG[2] (low byte)	M	XXh	RV_NT_AGI_THERMO_SNG[2]
42	recordValue39 = CTR_NT_AGI_AD_CMPL_SUM (high byte)	M	XXh	RV_CTR_NT_AGI_AD_CMPL_SUM
43	recordValue40 = CTR_NT_AGI_AD_CMPL_SUM (low byte)	M	XXh	RV_CTR_NT_AGI_AD_CMPL_SUM
44	recordValue41 = CTR_NT_AGI_SO2P_FQ (high byte)	M	XXh	RV_CTR_NT_AGI_SO2P_FQ
45	recordValue42 = CTR_NT_AGI_SO2P_FQ (low byte)	M	XXh	RV_CTR_NT_AGI_SO2P_FQ
46	recordValue43 = LV_NT_AFS_REQ_AGI	M	XXh	RV_LV_NT_AFS_REQ_AGI
47	recordValue44 = LV_NT_AFS_REQ_AGI_TMP_3	M	XXh	RV_LV_NT_AFS_REQ_AGI_TMP_3
48	recordValue45 = LV_SO2P_REQ_1	M	XXh	RV_LV_SO2P_REQ_1
49	recordValue46 = LV_SO2P_REQ_2	M	XXh	RV_LV_SO2P_REQ_2
50	recordValue47 = LV_SO2P_REQ_FQ	M	XXh	RV_LV_SO2P_REQ_FQ
51	recordValue48 =FAC_NT_AGI_LIM (high byte)	M	XXh	RV_ FAC_NT_AGI_LIM
52	recordValue49 =FAC_NT_AGI_LIM (low byte)	M	XXh	RV_ FAC_NT_AGI_LIM
53	recordValue50 = FAC_NT_AGI_MDL (high word, high byte)	M	XXh	RV_FAC_NT_AGI_MDL
54	recordValue51 = FAC_NT_AGI_MDL (high word, low byte)	M	XXh	RV_FAC_NT_AGI_MDL
55	recordValue52 = FAC_NT_AGI_MDL (low word, high byte)	M	XXh	RV_FAC_NT_AGI_MDL
56	recordValue53 = FAC_NT_AGI_MDL (low word, low byte)	M	XXh	RV_FAC_NT_AGI_MDL
57	recordValue54 = DIST_NS_NEW (high byte)	M	XXh	RV_DIST_NS_NEW

58	recordValue55 = DIST_NS_NEW (low byte)	M	XXh	RV_DIST_NS_NEW
59	recordValue56 = DIST_NT_NS_SHIFT[1] (high byte)	M	XXh	RV_DIST_NT_NS_SHIFT[1]
60	recordValue57 = DIST_NT_NS_SHIFT[1] (low byte)	M	XXh	RV_DIST_NT_NS_SHIFT[1]
61	recordValue58 = CTR_NS_AD_CYC[1] (high byte)	M	XXh	RV_ CTR_NS_AD_CYC[1]
62	recordValue59 = CTR_NS_AD_CYC[1] (low byte)	M	XXh	RV_ CTR_NS_AD_CYC[1]
63	recordValue60 = DIST_NT_NS_AD[1] (high byte)	M	XXh	RV_DIST_NT_NS_AD[1]
64	recordValue61 = DIST_NT_NS_AD[1] (low byte)	M	XXh	RV_DIST_NT_NS_AD[1]
65	recordValue62 = FAC_NOX_NS_AD[1] (high byte)	M	XXh	RV_FAC_NOX_NS_AD[1]
66	recordValue63 = FAC_NOX_NS_AD[1] (low byte)	M	XXh	RV_FAC_NOX_NS_AD[1]
67	recordValue64 = CTR_NS_SHIFT_CYC[1] (high byte)	M	XXh	RV_CTR_NS_SHIFT_CYC[1]
68	recordValue65 = CTR_NS_SHIFT_CYC[1] (low byte)	M	XXh	RV_CTR_NS_SHIFT_CYC[1]
69	recordValue66 = RATIO_MMV_NS_SHIFT_DIAG[1] (high byte)	M	XXh	RV_RATIO_MMV_NS_SHIFT_DIAG [1]
70	recordValue67 = RATIO_MMV_NS_SHIFT_DIAG[1] (low byte)	M	XXh	RV_ RATIO_MMV_NS_SHIFT_DIAG [1]

Note: In case of NC_NT_NR =1 must 0ffh transmitte for all "record Values " with array element[2])

7.29.67 22_5F8B - RCI_PMB_lesen

General information:

STATUS_PM_BACKUP
This service reads out the battery status.

Formula section:

```
If      service 22_5F8B received
then    send positive response
endif
```

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_PMB
3	RecordCommonIdentifier low byte	M	8Bh	RCI_PMB

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_PMB
3	RecordCommonIdentifier low byte	M	8Bh	RCI_PMB
4	RecordValue1 = Pmbackup[0]	M	XXh	RV_PMBACKUP[0]
5	RecordValue2 = Pmbackup[1]	M	XXh	RV_PMBACKUP[1]
6	RecordValue3 = Pmbackup[2]	M	XXh	RV_PMBACKUP[2]
7	RecordValue4 = Pmbackup[3]	M	XXh	RV_PMBACKUP[3]
8	RecordValue5 = Pmbackup[4]	M	XXh	RV_PMBACKUP[4]
9	RecordValue6 = Pmbackup[5]	M	XXh	RV_PMBACKUP[5]

10	RecordValue7 = Pmbackup[6]	M	XXh	RV_PMBACKUP[6]

7.29.68 22_5F8E - MSA deactivation lesen

General information:

This function is created to read out the state of "MotorStopAutomatik deaktivieren"

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	Request Service ID	M	22h	RDBCI
2d	RecordCommonIdentifier High Byte	M	5Fh	RCI_MSA_DEAK
3d	RecordCommonIdentifier Low Byte	M	8Eh	RCI_MSA_DEAK

Table Request Message

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2d	RecordCommonIdentifier High Byte	M	5Fh	RCI_MSA_DEAK
3d	RecordCommonIdentifier Low Byte	M	8Eh	RCI_MSA_DEAK
4d	recordValue1 = 0 -if B_msasw = FALSE 1 - if B_msasw = TRUE	M	XXh	RV_STAT_MSA_DEAK

7.29.69 22_5F8F - MSA Abschaltverhinderer lesen

General information:

This function is created to read out the state of "Selektive Deaktivierung Abschaltverhinderer MotorStopAutomatik "

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	Request Service ID	M	22h	RDBCI
2d	RecordCommonIdentifier High Byte	M	5Fh	RCI_MSA_DEAK_AV
3d	RecordCommonIdentifier Low Byte	M	8Fh	RCI_MSA_DEAK_AV

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2d	RecordCommonIdentifier High Byte	M	5Fh	RCI_MSA_DEAK_AV
3d	RecordCommonIdentifier Low Byte	M	8Fh	RCI_MSA_DEAK_AV
4d	recordValue1 = Swmsaav (high byte, high word)	M	XXh	RV_STAT_MSA_DEAK_AV
5d	recordValue2 = Swmsaav (low byte, high word)	M	XXh	RV_STAT_MSA_DEAK_AV
6d	recordValue3 = Swmsaav (high byte, low word)	M	XXh	RV_STAT_MSA_DEAK_AV

7d	recordValue4 = Swmsaav (low byte, low word)	M	XXh	RV_STAT_MSA_DEAK_AV

7.29.70 22_5F90 - IMAALLE lesen

General Information: This function is created to read out the described values from the ECU.

7.29.70.1 MSD 85 only

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	Response Service ID	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	5Fh	RV_IMAALLE
3	RecordCommonIdentifier Low Byte	M	90h	RV_IMAALLE

Positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1	Request Service ID	M	62h	RDBCIPR
2	RecordCommonIdentifier High Byte	M	5Fh	RV_IMAALLE
3	RecordCommonIdentifier Low Byte	M	90h	RV_IMAALLE
4	recordValue1 = EGY_SP_IV_EXT_ADJ[0] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[0]
5	rcord value 2 = EGY_SP_IV_EXT_ADJ[0] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[0]
6	recordValue3 = MFF_ABSV_IV_EXT_ADJ[0] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[0]
7	recordValue4 = MFF_ABSV_IV_EXT_ADJ[0] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[0]
8	recordValue5 = EGY_SP_IV_EXT_ADJ[1] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[1]
9	rcord value6 = EGY_SP_IV_EXT_ADJ[1] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[1]
10	recordValue7 = MFF_ABSV_IV_EXT_ADJ[1] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[1]
11	recordValue8 = MFF_ABSV_IV_EXT_ADJ[1] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[1]
12	recordValue9 = EGY_SP_IV_EXT_ADJ[2] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[2]
13	rcord value10 = EGY_SP_IV_EXT_ADJ[2] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[2]
14	recordValue11 = MFF_ABSV_IV_EXT_ADJ[2] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[2]
15	recordValue12 = MFF_ABSV_IV_EXT_ADJ[2] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[2]
16	recordValue139 = EGY_SP_IV_EXT_ADJ[3] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[3]
17	rcord value14 = EGY_SP_IV_EXT_ADJ[3] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[3]
18	recordValue15 = MFF_ABSV_IV_EXT_ADJ[3] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[3]
19	recordValue16 = MFF_ABSV_IV_EXT_ADJ[3] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[3]
20	recordValue17 = EGY_SP_IV_EXT_ADJ[4] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[4]
21	rcord value18 = EGY_SP_IV_EXT_ADJ[4] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[4]

22	recordValue19 = MFF_ABSV_IV_EXT_ADJ[4] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[4]
23	recordValue20 = MFF_ABSV_IV_EXT_ADJ[4] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[4]
24	recordValue21 = EGY_SP_IV_EXT_ADJ[5] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[5]
25	rcord value22 = EGY_SP_IV_EXT_ADJ[5] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[5]
26	recordValue23 = MFF_ABSV_IV_EXT_ADJ[5] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[5]
27	recordValue24 = MFF_ABSV_IV_EXT_ADJ[5] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[5]
28	recordValue25 = EGY_SP_IV_EXT_ADJ[6] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[6]
29	rcord value26 = EGY_SP_IV_EXT_ADJ[6] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[6]
30	recordValue27 = MFF_ABSV_IV_EXT_ADJ[6] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[6]
31	recordValue28 = MFF_ABSV_IV_EXT_ADJ[6] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[6]
32	recordValue29 = EGY_SP_IV_EXT_ADJ[7] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[7]
33	rcord value30 = EGY_SP_IV_EXT_ADJ[7] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[7]
34	recordValue31 = MFF_ABSV_IV_EXT_ADJ[7] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[7]
35	recordValue32 = MFF_ABSV_IV_EXT_ADJ[7] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[7]

7.29.70.2 All other projects

Description: This function is created to read out the described values from the ECU.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Response Service ID	M	22h	RDBCI
2	RecordCommonIdentifier High Byte	M	5Fh	RV_IMAALLE
3	RecordCommonIdentifier Low Byte	M	90h	RV_IMAALLE

Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Request Service ID	M	62h	RDBCIPR
2	RecordCommonIdentifier High Byte	M	5Fh	RV_IMAALLE
3	RecordCommonIdentifier Low Byte	M	90h	RV_IMAALLE
4	recordValue1 = EGY_SP_IV_EXT_ADJ[0] high byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[0]
5	rcord value 2 = EGY_SP_IV_EXT_ADJ[0] low byte	M	XXh	RV_EGY_SP_IV_EXT_ADJ[0]
6	recordValue3 = MFF_ABSV_IV_EXT_ADJ[0] high byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[0]
7	recordValue4 = MFF_ABSV_IV_EXT_ADJ[0] low byte	M	XXh	RV_MFF_ABSV_IV_EXT_ADJ[0]
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
4*NC_CYL_NR	recordValue4*NC_CYL_NR-3 = EGY_SP_IV_EXT_ADJ[NC_CYL_NR-1] high byte	M	XXh	RV_
4*NC_CYL_NR +1	recordValue4*NC_CYL_NR-2= EGY_SP_IV_EXT_ADJ[NC_CYL_NR-1] low byte	M	XXh	RV_

4*NC_CYL_NR +2	recordValue4*NC_CYL_NR-1= MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR-1] high byte	M	XXh	RV_
4*NC_CYL_NR +3	recordValue4*NC_CYL_NR = MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR-1] low byte	M	XXh	RV_

7.29.71 22_5FDE - VVT- Minhub lesen

General information:

This service reads out the specified value(s).

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_
3	RecordCommonIdentifier low byte	M	DEh	RCI_

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_
3	RecordCommonIdentifier low byte	M	DEh	RCI_
4	RecordValue1 = Minhub high byte	M	XXh	RV_
5	RecordValue2 = Minhub low byte	M	XXh	RV_

7.29.72 22_5FF0 - RCI_LL_ABGLEICH_lesen

General information:

Abgleichswert LL lesen – ReadDataByCommonIdentifier

With the tester via KWP telegram an idle speed offset for five different conditions can be read.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_RCI_LL_ABGLEICH
3	RecordCommonIdentifier low byte	M	F0h	RCI_RCI_LL_ABGLEICH

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	***Response Service ID***	***M***	***62h***	***RDBCIPR***
2	RecordCommonIdentifier high byte	M	5Fh	RCI_RCI_LL_ABGLEICH
3	RecordCommonIdentifier low byte	M	F0h	RCI_RCI_LL_ABGLEICH
4	RecordValue1 = N_KWP_OFS_ACC_DRI	M	XXh	RV_ N_KWP_OFS_ACC_DRI
5	RecordValue2 = N_KWP_OFS_DRI	M	XXh	RV_ N_KWP_OFS_DRI
6	RecordValue3 = N_KWP_OFS	M	XXh	RV_N_KWP_OFS

7	RecordValue4 = N_KWP_OFS_ACC	M	XXh	RV_N_KWP_OFS_ACC
8	RecordValue5 = N_KWP_OFS_VB	M	XXh	RV_N_KWP_OFS_VB

7.29.73 22_5FF1 - CO Abgleichswert lesen

General information:

With the tester via KWP telegram the values for CO adjustment can be read.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_
3	RecordCommonIdentifier low byte	M	F1h	RCI_

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	***Response Service ID***	***M***	***62h***	***RDBCIPR***
2	RecordCommonIdentifier high byte	M	5Fh	RCI_
3	RecordCommonIdentifier low byte	M	F1h	RCI_
4	RecordValue1 = LV_FAC_MFF_ADD_EXT_ADJ_NVMY	M	XXh	RV_
5	RecordValue2 = LV_RST_MFF_ADD_EXT_ADJ_NVMY	M	XXh	RV_
6	RecordValue3 = FAC_MFF_ADD_EXT_ADJ	M	XXh	RV_
7	RecordValue4 = MFF_ADD_EXT_ADJ[1] high byte	M	XXh	RV_
8	RecordValue5 = MFF_ADD_EXT_ADJ[1] low byte	M	XXh	RV_
9	RecordValue6 = MFF_ADD_EXT_ADJ[2] high byte	M	XXh	RV_
10	RecordValue7 = MFF_ADD_EXT_ADJ[2] low byte	M	XXh	RV_
11	RecordValue8 = MFF_ADD_EXT_ADJ_NVMY high byte	M	XXh	RV_
12	RecordValue9 = MFF_ADD_EXT_ADJ_NVMY low byte	M	XXh	RV_

7.29.74 22_5FF2 - RCI_ECU_CONFIG_lesen

General information:

This service shows the actual recognized variants.

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_ECU_CONFIG
3	RecordCommonIdentifier low byte	M	F2h	RCI_ECU_CONFIG

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_ECU_CONFIG
3	RecordCommonIdentifier low byte	M	F2h	RCI_ECU_CONFIG

4	inputOutputControlParameter - LV_AT	M	XXh	RV_ LV_AT
5	inputOutputControlParameter - LV_VAR_ACIN	M	XXh	RV_LV_VAR_ACIN
6	inputOutputControlParameter - LV_VAR_AMT	M	XXh	RV_LV_VAR_AMT
7	inputOutputControlParameter - LV_VAR_ARS	M	XXh	RV_ LV_VAR_ARS
8	inputOutputControlParameter - LV_VAR_ASR	M	XXh	RV_ LV_VAR_ASR
9	inputOutputControlParameter - LV_VAR_BN	M	XXh	RV_LV_VAR_BN
10	inputOutputControlParameter - LV_VAR_BN_MSW	M	XXh	RV_LV_VAR_BN_MSW
11	inputOutputControlParameter - LV_VAR_DCC	M	XXh	RV_LV_VAR_DCC
12	inputOutputControlParameter - LV_VAR_EBOX_CFA	M	XXh	RV_LV_VAR_EBOX_CFA
13	inputOutputControlParameter - LV_VAR_ETCU	M	XXh	RV_ LV_VAR_ETCU
14	inputOutputControlParameter - LV_VAR_ICL	M	XXh	RV_LV_VAR_ETCU
15	inputOutputControlParameter - LV_VAR_MSW	M	XXh	RV_LV_VAR_MSW
16	inputOutputControlParameter - LV_VAR_PSTE	M	XXh	RV_LV_VAR_PSTE
17	inputOutputControlParameter - LV_VAR_SOF	M	XXh	RV_LV_VAR_SOF
18	inputOutputControlParameter - CONF_SOF_SWI (00h= no SOF_SWI, 01h=SOF_SWI, 02h=SOF_SWI_AMT)	M	XXh	RV_LV_VAR_SOF_SWI
19	inputOutputControlParameter - LV_VAR_BN_GEAR_REV	M	XXh	RV_LV_VAR_BN_GEAR_REV
20	inputOutputControlParameter – LV_VAR_EAC **(only MSV80)** not supported **(other)**	M	[XXh, FFh]	
21	inputOutputControlParameter - LV_VAR_VEH	M	XXh	RV_LV_VAR_VEH
22	inputOutputControlParameter - LV_VAR_SAP	M	XXh	
23	inputOutputControlParameter - LV_VAR_EF	M	XXh	RV_ LV_VAR_EF
24	inputOutputControlParameter - LV_VAR_ECRAS_UP	M	XXh	RV_ LV_VAR_ECRAS_UP
25	inputOutputControlParameter - LV_VAR_RLY_ACCOUT (not MSD85) not supported (MSD85)	M	[XXh, FFh]	RV_LV_VAR_RLY_ACCOUT
26	inputOutputControlParameter - LV_VAR_SAV	M	XXh	
27	inputOutputControlParameter - LV_VAR_RLY_ST	M	XXh	RV_LV_VAR_RLY_ST
28	inputOutputControlParameter - LV_VAR_ASR_3	M	XXh	RV_LV_VAR_ASR_3
29	inputOutputControlParameter - LV_VAR_BN_LDM	M	XXh	RV_LV_VAR_BN_LDM
30	inputOutputControlParameter - LV_VAR_BN_LTG_HDLP_L	M	XXh	RV_LV_VAR_BN_LTG_HDLP_L
31	inputOutputControlParameter - LV_VAR_LSH_DOWN	M	XXh	RV_LV_VAR_LSH_DOWN
32	inputOutputControlParameter - LV_VAR_LSH_UP	M	XXh	RV_LV_VAR_LSH_UP
33	inputOutputControlParameter - LV_VAR_ASR_4	M	XXh	RV_ LV_VAR_ASR_4
34	inputOutputControlParameter - LV_VAR_MAF	M	XXh	RV_LV_VAR_MAF
35	inputOutputControlParameter - LV_VAR_PSTE_2 **or** (LV_VAR_PSTE_3 **and** STATE_PSTE_3_SRC = "AFS")	M	XXh	RV_LV_VAR_PSTE_2
36	inputOutputControlParameter - LV_VAR_BN_EFP	M	XXh	RV_LV_VAR_BN_EFP
37	inputOutputControlParameter - LV_SENS_BAT_SMT_DET	M	XXh	RV_LV_VAR_BN_EFP
38	inputOutputControlParameter - LV_VAR_BN_TRL	M	XXh	RV_LV_VAR_BN_TRL
39	inputOutputControlParameter - not supported	M	FFh	
40	inputOutputControlParameter - LV_VAR_ECRAS_DOWN	M	XXh	RV_LV_VAR_ECRAS_DOWN
41	inputOutputControlParameter - LV_VAR_NOX **(only MSD80/MSD81)** not supported **(other)**	M	[XXh, FFh]	
42	inputOutputControlParameter - not supported	M	FFh	
43	inputOutputControlParameter - LV_VAR_ETCU_SPT **(only MSV80)** not supported **(other)**	M	[XXh, FFh]	RV_ LV_VAR_ETCU_SPT
44	inputOutputControlParameter - LV_VAR_TCT	M	XXh	RV_LV_VAR_TCT
45	inputOutputControlParameter - LV_VAR_AEB	M	XXh	RV_LV_VAR_AEB
46	inputOutputControlParameter - LV_VAR_TQ_PBR	M	XXh	RV_LV_VAR_TQ_PBR

7.29.75 22_5FF3 - RCI_UET_lesen

General information:

This service reads out the specified value(s).

Application conditions:

Initialisation: at reset the values from the NVMY

Activation: LV_IGK=1

Deactivation: LV_IGK=0
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_UET
3	RecordCommonIdentifier low byte	M	F3h	RCI_UET

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_UET
3	RecordCommonIdentifier low byte	M	F3h	RCI_UET
4	RV_TOIL_MAX_WARN	M	XXh	RV_TOIL_MAX_WARN
5	RecordValue = VS_TOIL_MAX	M	XXh	RV_VS_TOIL_MAX
6	RecordValue = N_TOIL_MAX high byte	M	XXh	RV_N_TOIL_MAX
7	RecordValue = N_TOIL_MAX low byte	M	XXh	RV_N_TOIL_MAX
8	RecordValue = GEAR_TOIL_MAX	M	XXh	RV_GEAR_TOIL_MAX
9	RecordValue = TQI_AV_TOIL_MAX high byte	M	XXh	RV_TQI_AV_TOIL_MAX
10	RecordValue = TQI_AV_TOIL_MAX low byte	M	XXh	RV_TQI_AV_TOIL_MAX
11	RecordValue = TAM_TOIL_MAX	M	XXh	RV_TAM_TOIL_MAX
12	RecordValue = CTR_TOIL_MAX	M	XXh	RV_CTR_TOIL_MAX
13	RecordValue = TCO_TOIL_MAX	M	XXh	RV_TCO_TOIL_MAX
14	RecordValue = DIST_TOIL_MAX high byte	M	XXh	RV_DIST_TOIL_MAX
15	RecordValue = DIST_TOIL_MAX low byte	M	XXh	RV_DIST_TOIL_MAX
16	RecordValue = PV_AV_TOIL_MAX	M	XXh	RV_PV_AV_TOIL_MAX
17	RecordValue = LV_CS_TOIL_MAX	M	XXh	RV_LV_CS_TOIL_MAX
18	RecordValue = TOIL_THD_TOIL_MAX	M	XXh	RV_TOIL_THD_TOIL_MAX

7.29.76 22_5FF4 - Fusshebelwerk Fehlbedienung lesen

General information:

This service reads out the specified value(s).

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_
3	RecordCommonIdentifier low byte	M	F4h	RCI_

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_
3	RecordCommonIdentifier low byte	M	F4h	RCI_
4	RecordValue1 = dummy fix	M	FFh	RV_

7.29.77 22_5FF6 - RCI_UEN_lesen

General information:

Exact description of the Function see file 14A06301.

Detection of an overreving event and non-volatile storage of:

An overreving event starts at exceeding the engine speed C_N_LIM_MIN and active engine speed limitation (LV_N_MAX = 1).

At each overreving event, the event frequency counter is incremented by 1. The highest engine speed reached N_MAX and the associated total running time in case of highest engine speed reached TRT_N_MAX is only overwritten if a higher engine speed was reached.

Application conditions:

Initialisation:	at reset the values from the NVMY
Activation:	Kl.15 on
Deactivation:	Kl.15 off Diagnostic timeout
Service location:	see list of “implemented diagnostic services”

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Request Service ID**	**M**	**22h**	**RDBCI**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_UEN
3	RecordCommonIdentifier low byte	M	F6h	RCI_UEN

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2	RecordCommonIdentifier high byte	M	5Fh	RCI_UEN
3	RecordCommonIdentifier low byte	M	F6h	RCI_UEN
4	RecordValue = N_MAX high byte	M	XXh	RV_N_MAX
5	RecordValue = N_MAX low byte	M	XXh	RV_N_MAX
6	RecordValue = TRT_N_MAX high word high byte	M	XXh	RV_TRT_N_MAX
7	RecordValue = TRT_N_MAX high word low byte	M	XXh	RV_TRT_N_MAX
8	RecordValue = TRT_N_MAX low word high byte	M	XXh	RV_TRT_N_MAX
9	RecordValue = TRT_N_MAX low word low byte	M	XXh	RV_TRT_N_MAX
10	RecordValue = CTR_N_MAX	M	XXh	RV_CTR_N_MAX
11	RecordValue = CTR_KM_N_MAX high byte	M	XXh	RV_CTR_KM_N_MAX
12	RecordValue = CTR_KM_N_MAX low byte	M	XXh	RV_CTR_KM_N_MAX
13	RecordValue = N_GRD_N_MAX	M	XXh	RV_N_GRD_N_MAX
14	RecordValue = VS_N_MAX (**not for MSS70**) = VS_H_N_MAX (high byte, **only MSS70**)	M	XXh	RV_VS_N_MAX
15	RecordValue = PV_AV_N_MAX	M	XXh	RV_PV_AV_N_MAX
16	RecordValue = GEAR_EF_N_MAX	M	XXh	RV_GEAR_EF_N_MAX
17	RecordValue = LV_CS_N_MAX	M	XXh	RV_LV_CS_N_MAX
18	RecordValue = T_SUM_N_MAX high byte	M	XXh	RV_T_SUM_N_MAX
19	RecordValue = T_SUM_N_MAX low byte	M	XXh	RV_T_SUM_N_MAX
20	RecordValue = T_N_MAX high byte	M	XXh	RV_T_N_MAX
21	RecordValue = T_N_MAX low byte	M	XXh	RV_T_N_MAX
22	RecordValue = NR_PAT_SCC_N_MAX	M	XXh	RV_NR_PAT_SCC_N_MAX

Hint to Data Byte 14d: physical range is 0..510 km/h for MSS70, 0..255 km/h for all other

7.29.78 22_5FF7 - Laufruhepruefung auslesen

General information:

This service reads out the specified value(s).

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**Request Service ID**	**M**	**22h**	**RDBCI**
2d	RecordCommonIdentifier high byte	M	5Fh	RCI_LRP
3d	RecordCommonIdentifier low byte	M	F7h	RCI_LRP

Table Request Message

Positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2d	RecordCommonIdentifier high byte	M	5Fh	RCI_LRP
3d	RecordCommonIdentifier low byte	M	F7h	RCI_LRP

4d	RecordValue1 = St_vbrvs_aus Bit 7	M	XXh	RV_STAT_LRP_BIT7
5d	RecordValue2 = St_vbrvs_aus Bit 6	M	XXh	RV_STAT_LRP_BIT6
6d	RecordValue3 = St_vbrvs_aus Bit 5	M	XXh	RV_STAT_LRP_BIT5
7d	RecordValue4 = St_vbrvs_aus Bit 4	M	XXh	RV_STAT_LRP_BIT4
8d	RecordValue5 = St_vbrvs_aus Bit 3	M	XXh	RV_STAT_HUBEINGRIFF_INAKTIV
9d	RecordValue6 = St_vbrvs_aus Bit 2	M	XXh	RV_STAT_MINHUBEINGRIFF_INAKTIV
10d	RecordValue7 = St_vbrvs_aus Bit 1	M	XXh	RV_STAT_ZUENDWINKELEINGRIFF_INAKTIV
11d	RecordValue8 = St_vbrvs_aus Bit 0	M	XXh	RV_STAT_GEMISCHEINGRIFF_INAKTIV

Table Positive Response Message

7.29.79 22_5FF8 - Status Homogenbetrieb auslesen

General information:

This service reads out the engine operation mode.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**Request Service ID**	**M**	**22h**	**RDBCI**
2d	RecordCommonIdentifier high byte	M	5Fh	RCI_LRP
3d	RecordCommonIdentifier low byte	M	F8h	RCI_LRP

Table Request Message

Positive Response on service

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1d	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2d	RecordCommonIdentifier high byte	M	5Fh	RCI_LRP
3d	RecordCommonIdentifier low byte	M	F8h	RCI_LRP
4d	RecordValue1 = Ba_soll	M	XXh	RV_BA_
5d	RecordValue2 = Ba_ist	M	XXh	RV_BA_

Table Positive Response Message

7.29.80 22_C000 – Read imob status (STATUS_EWS)

General information:

IMMO: RD_IMOB_STATUS

The service is different for boot-sw and ecu-sw.

A negativ response is possible by a wrong parameter.

Formula section:

Only for boot sw-version:

```
tmp_k_ews4_locked = 0

If(1)    (K_EWS4[0..15] > 0x0) and (K_EWS4[0..15] < 0xFF)      (Blank Check Page 0)
 Then(1)        tmp_k_ews4_locked = 1
 Else(1)        tmp_k_ews4_locked  unchanged
 Endif(1)
```

Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1d	**Request Service ID**	**M**	**22h**	**RDBCI**
2d	RecordCommonIdentifier	M	C0h	RCI_TSP
3d	RecordCommonIdentifier	M	00h	RCI_TSP

Table Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
1d	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2d	RecordCommonIdentifier	M	C0h	RCI_TSP
3d	RecordCommonIdentifier	M	00h	RCI_TSP
4d	RecordValue 1 = tmp_k_ews4_locked <<7 | 1 **(boot sw version)** = IMOB_KWP_STATE_0 **(ecu sw version)**	M	XXh	RV_PST

5d	RecordValue 2 = xx	M	FFh	RV_PST
6d	RecordValue 3 = xx	M	FFh	RV_PST
7d	RecordValue 4 = xx	M	FFh	RV_PST
8d	RecordValue 5 = xx	M	FFh	RV_PST
9d	RecordValue 6 = xx	M	FFh	RV_PST
10d	RecordValue 7 = xx	M	FFh	RV_PST
11d	RecordValue 8 = xx	M	FFh	RV_PST
12d	RecordValue 9 = 8- tmp_k_ews4_locked **(boot sw version)** = IMOB_KWP_STATE_8 **(ecu sw version)**	M	XXh	RV_PST
13d	RecordValue 10 = xx	M	FFh	RV_PST
14d	RecordValue 11 = xx	M	FFh	RV_PST
15d	RecordValue 12 = xx	M	FFh	RV_PST
16d	RecordValue 13 = xx	M	FFh	RV_PST
17d	RecordValue 14 = xx	M	FFh	RV_PST
18d	RecordValue 15 = xx	M	FFh	RV_PST
19d	RecordValue 16 = 1 **(boot sw version)** = IMOB_KWP_STATE_15 **(ecu sw version)**	M	XXh	RV_PST

Table Positive Response Message

7.29.81 22_C002 - Read imob k_ews(tmp) (STATUS_EWS_SK)

General information:

IMMO: Read RD_IMOB_K_EWS (Read "temporary secret key" from ECU-Ram)

The service is different for boot-sw and ecu-sw.

A negative response is possible by a wrong parameter

Formula section:

Only for boot sw-version:

```
    If(1)    (K_EWS4[0..15] > 0x0) && (K_EWS4[0..15] < 0xFF)        (Blank Check Page 0)
    Then(1)        IMOB_KWP_RESP[0..15] = 0 (whole array)
    Else(1)
            If(2)        ( IMOB_KWP_K_EWS4[0..15] !=
                         IMOB_KWP_K_EWS4_TMP0[0..15] xor 0xFF )
                         (short data consistence check)
            Then(2)      IMOB_KWP_RESP[0..15] = 0xFF (whole array)
            Else(2)      IMOB_KWP_RESP[0..15] = IMOB_KWP_K_EWS4[0..15]
            Endif(2)
Endif(1)
```

Only for ecu sw-version:

```
If(1)      STATE_IMOB_K_EWS != NC_K_EWS_UNLOCKED
           and NC_K_EWS_MAX = IMOB_NR_VLD_K_EWS4
Then(1)    IMOB_KWP_RESP[0-15] = 0 (whole array)
Else(1)
           If(2)      ( IMOB_KWP_K_EWS4[0..15] !=
                      IMOB_KWP_K_EWS4_TMP0[0..15] xor 0xFF )
                      (short data consistence check)
           Then(2)    IMOB_KWP_RESP[0..15] = 0xFF (whole array)
           Else(2)    IMOB_KWP_RESP[0..15] = IMOB_KWP_K_EWS4[0..15]
           Endif(2)
Endif(1)
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**Request Service ID**	**M**	**22h**	**RDBCI**
2d	RecordCommonIdentifier	M	C0h	RCI_TSP
3d	RecordCommonIdentifier	M	02h	RCI_TSP

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1d	**Response Service ID**	**M**	**62h**	**RDBCIPR**
2d	RecordCommonIdentifier	M	C0h	RCI_TSP
3d	RecordCommonIdentifier	M	02h	RCI_TSP
4d	RecordValue 1	M	FFh	RV_PST
5d	RecordValue 2	M	FFh	RV_PST
6d	RecordValue 3	M	FFh	RV_PST
7d	RecordValue 4	M	FFh	RV_PST
8d	RecordValue 5	M	FFh	RV_PST
9d	RecordValue 6	M	FFh	RV_PST
10d	RecordValue 7	M	FFh	RV_PST
11d	RecordValue 8	M	FFh	RV_PST
12d	RecordValue 9	M	FFh	RV_PST
13d	RecordValue 10	M	FFh	RV_PST
14d	RecordValue 11	M	FFh	RV_PST
15d	RecordValue 12	M	FFh	RV_PST
16d	RecordValue 13	M	FFh	RV_PST
17d	RecordValue 14	M	FFh	RV_PST
18d	RecordValue 15	M	FFh	RV_PST
19d	RecordValue 16	M	FFh	RV_PST
20d	RecordValue 17 = IMOB_KWP_RESP[0]	M	XXh	RV_PST
21d	RecordValue 18 = IMOB_KWP_RESP[1]	M	XXh	RV_PST
22d	RecordValue 19 = IMOB_KWP_ RESP[2]	M	XXh	RV_PST

23d	RecordValue 20 = IMOB_KWP_ RESP[3]	M	XXh	RV_PST
24d	RecordValue 21 = IMOB_KWP_ RESP[4]	M	XXh	RV_PST
25d	RecordValue 22 = IMOB_KWP_ RESP[5]	M	XXh	RV_PST
26d	RecordValue 23 = IMOB_KWP_ RESP[6]	M	XXh	RV_PST
27d	RecordValue 24 = IMOB_KWP_ RESP[7]	M	XXh	RV_PST
28d	RecordValue 25 = IMOB_KWP_ RESP[8]	M	XXh	RV_PST
29d	RecordValue 26 = IMOB_KWP_ RESP[9]	M	XXh	RV_PST
30d	RecordValue 27 = IMOB_KWP_ RESP[10]	M	XXh	RV_PST
31d	RecordValue 28 = IMOB_KWP_ RESP[11]	M	XXh	RV_PST
32d	RecordValue 29 = IMOB_KWP_ RESP[12]	M	XXh	RV_PST
33d	RecordValue 30 = IMOB_KWP_ RESP[13]	M	XXh	RV_PST
34d	RecordValue 31 = IMOB_KWP_ RESP[14]	M	XXh	RV_PST
35d	RecordValue 32 = IMOB_KWP_ RESP[15]	M	XXh	RV_PST

7.30 SID 2Eh: writeDataByCommonIdentifier Service

General information:

All implemented SID 2Eh services are described below.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_msahfkreset	O/V	0...1H	0...1	1	[-]
Flag indicating request "Reset MSA-Tabellenspeicher"					
B_msasw	O/V	0...1H	0...1	1	[-]
Bedingung MSA bis zum nächsten Zündungswechsel deaktivieren					
B_pmrestore	O/V	0...1H	0...1	1	[-]
Bedingung Daten zurückgeschrieben					
B_spa_csoll	O/V/S	0...1H	0...1	1	[-]
Codierung SPA aus Tester					
B_zrlvs_clr	O/V	0...1H	0...1	1	[-]
Zurücksetzen der internen Zähler des LVS					
CHK_STAMP	O/S	0...FFFFFFFFH	0...4294967295	1	[-]
Check stamp written by the customers production					
CONF_EF	O/V/S	0...FFH	0...255	1	[-]
Configuration of exhaust gas flap					
CTR_NS_AD_CYC_EXT_ADJ[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...65535	1	[-]
External Adjustment value for Counter of NOx signal gain adaptations					
CTR_NT_AGI_AD_CMPL_SUM_EXT_ADJ	O/V	0...FFFFH	0...65535	1	[-]
External adjustment value for counter of completed aging adaptation					
CTR_NT_AGI_SO2P_FQ_EXT_ADJ	O/V	0...FFFFH	0...65535	1	[-]
External adjustment value for counter of completed aging adaptation during FQ adaptation					
CTR_NS_SHIFT_CYC_EXT_ADJ[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...65535	1	[-]
External values of CTR_NS_SHIFT_CYC					
DIST_NS_NEW_EXT_ADJ	O/V	0...FFFFH	0...524280	8	[km]
External adjust value for mileage counter value of last exchange of NOx sensor					
DIST_NT_NS_AD_EXT_ADJ[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...524280	8	[km]
External Adjustment for Current distance from last NOx signal gain adaptation					
DIST_NT_NS_SHIFT_EXT_ADJ[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...524280	8	[km]
External adjustment value for DIST_NT_NS_SHIFT					
EGY_SP_IV_EXT_ADJ[NC_CYL_NR]	O/V/S	0...FFFFH	0...255	3.8911e-3	[mJ]
external adjusted injection valve EGY setpoint value (high flow)					
FAC_MFF_ADD_EXT_ADJ	O/V	0...FFH	-50...49.60937	0.390625	[%]
Factor for CO correction in idling					
FAC_NOX_NS_AD_EXT_ADJ[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...127.99804	1.9531e-3	[-]
External Adjustment for Adaptation of the NOx sensor characteristic shift					
FAC_NT_AGI_LIM_EXT_ADJ	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
External adjustment value for limited NOx trap aging factor					
FAC_NT_AGI_MDL_EXT_ADJ	O/V	0...FFFFFFFFH	0...0.99999	0.2328e-9	[-]
External value for modeled aging of NT					
FAC_OIL_EXT_REQ_1	O/V/S	0...FFH	0...2.55	0.01	[-]
Country dependant factor for oil change intervall via tester					
FAC_OIL_EXT_REQ_2	O/V/S	0...FFH	0...2.55	0.01	[-]
Country dependant factor for oil change intervall via tester					
IDX_VAR_COD[2]	O/V/S	0...FFH	0...255	1	[-]
changing index of vehicle configuration code via tester					
IMOB_KWP_K_EWS4[16]	O	0...FFH	0...255	1	[-]
Received secret key					
IMOB_KWP_K_EWS4_1[16]	O	0...FFH	0...255	1	[-]
Received secret key, 1st backup					
IMOB_KWP_K_EWS4_2[16]	O	0...FFH	0...255	1	[-]
Received secret key, 2nd back up					
IMOB_KWP_K_EWS4_TMP0[16]	O	0...FFH	0...255	1	[-]
Received secret key, temporaery					
LV_ACT_N_SP_IS_BAS_EXT_ADJ	O	0...1H	0...1	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
logical variable for switching off the max-selection					
LV_ALTER_CTL_ENA	O/V/S	0...1H	0...1	1	[-]
Intelligent alternator control activated					
LV_CAT_CONF_DIS_EXT_REQ	O/V/S	0...1H	0...1	1	[-]
disable of learnig cat variants and setting variant to "no cat" via tester					
LV_FAC_MFF_ADD_EXT_ADJ_NVMY	O/V	0...1H	0...1	1	[-]
Logical value for storing CO correction in NVMY					
LV_FHW_RST	O/V	0...1H	0...1	1	[-]
boolean for reset of wrong used pedal unit					
LV_IGK_OFF_ACK_ENA	O/V/S	0...1H	0...1	1	[-]
Diagnosis acknowledge IGK off enabled via service tester job (1= enabled)					
LV_KWP_ENA	O/V	0...1H	0...1	1	[-]
logical variable for idle speed setpoint offset by tester enabled					
LV_LTG_GAS_ENA	O/V/S	0...1H	0...1	1	[-]
xenon-filled headlights mounted					
LV_NT_AFS_REQ_AGI_TMP_3_EXT_ADJ	O/V	0...1H	0...1	1	[-]
External adjustment value for logical value for the request of lambda =1 operation					
LV_POW_MNG_HIS_RST	O/V	0...1H	0...1	1	[-]
conditions for function request parking power					
LV_RST_MFF_ADD_EXT_ADJ_NVMY	O/V	0...1H	0...1	1	[-]
Flag to reset external MFF correction to 0					
LV_SO2P_REQ_2_EXT_ADJ	O/V	0...1H	0...1	1	[-]
External adjustment value for request of a desulfation (forces catalyst heating)					
LV_SO2P_REQ_FQ_EXT_ADJ	O/V	0...1H	0...1	1	[-]
External adjustment value for logical value for active FQ adaptation					
LV_STST_DEAC	O/V/S	0...1H	0...1	1	[-]
Logical value for switch off the MSA permanently					
LV_VAR_OBDC_CAN	O/V	0...1H	0...1	1	[-]
Variable to activate OBD communication on CAN (=1)					
LV_STST_VAR_COD	O/V/S	0...1H	0...1	1	[-]
logical value for MSA-coding (0 - without MSA. 1 - with MSA)					
LV_VVL_MIN_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by tester					
LV_VVL_MIN_NVMY_EXT_ADJ	O/V	0...1H	0...1	1	[-]
activation of external adjustment by tester					
MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR]	O/V/S	0...FFFFH	0...1389	0.0211948	[mg/stk]
External adjusted injection valve absolute MFF correction value (low flow)					
N_KWP_OFS	O/V	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 0 and LV_DRI = 0					
N_KWP_OFS_ACC	O/V	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 1 and LV_DRI = 0					
N_KWP_OFS_ACC_DRI	O/V	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 1 and LV_DRI = 1					
N_KWP_OFS_ACC_DRI_KWP	O	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 1 and LV_DRI = 1 / Byte 4 in the KWP telegramm					
N_KWP_OFS_ACC_DRI_NVMY	O/V/S	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 1 and LV_DRI = 1 stored in non volatile memory					
N_KWP_OFS_ACC_KWP	O	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 1 and LV_DRI = 0 / Byte 7 in the KWP telegramm					
N_KWP_OFS_ACC_NVMY	O/V/S	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 1 and LV_DRI = 0 stored in non volatile memory					
N_KWP_OFS_DRI	O/V	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 0 and LV_DRI = 1					
N_KWP_OFS_DRI_KWP	O	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 0 and LV_DRI = 1 / Byte 5 in the KWP telegramm					
N_KWP_OFS_DRI_NVMY	O/V/S	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 0 and LV_DRI = 1 stored in non volatile memory					
N_KWP_OFS_KWP	O	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at LV_ACIN = 0 and LV_DRI = 0 / Byte 6 in the KWP telegramm					
N_KWP_OFS_NVMY	O/V/S	80...7FH	-256...254	2	[rpm]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
idle speed setpoint offset at LV_ACIN = 0 and LV_DRI = 0 stored in non volatile memory					
N_KWP_OFS_VB	O/V	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at battery charging is active					
N_KWP_OFS_VB_KWP	O	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at battery charging is active / Byte 8 in the KWP telegramm					
N_KWP_OFS_VB_NVMY	O/V/S	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset at battery charging is active stored in non volatile memory					
N_SP_OFS_KWP	O/V	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset from the tester to idle speed module depend of the 5 conditions					
N_SP_OFS_KWP_1	O	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset from the tester to idle speed module for calculation					
NT_AGI_SO2P_FQ_SUM_EXT_ADJ	O/V	0...FFFFFFH	0...255.99998	0.0153e-3	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
External adjustment value for sum of NOx trap aging factor during FQ adaptation					
NT_AGI_SUL_SNG_EXT_ADJ[NC_NT_NR]	O/V	0...FFFFFFFFH	0...0.99999	0.2328e-9	[-]
External adjustment value for NOx trap aging factor due to sulphur load (bench selective)					
NT_AGI_THERMO_SNG_EXT_ADJ[NC_NT_NR]	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
External adjustment value for NOx trap aging factor due to thermal aging (bench selective)					
NT_SUL_32_EXT_ADJ[NC_NT_NR]	O/V	0...FFFFFFFFH	0...10485.6	2.4414e-6	[mg]
External adjustment value for NOx trap sulphur loading					
NT_SUL_H_32_EXT_ADJ[NC_NT_NR]	O/V	0...FFFFFFFFH	0...10485.6	2.4414e-6	[mg]
External adjustment value for NOx trap sulphur loading for high sulphured fuel					
Pmrestore[7]	O/V	0...FFH	0...255	1	[-]
Restorespeicher					
POW_CONF_IDX_EXT_REQ	O/V/S	0...3H	0...3	1	[-]
Selected power configuration code via tester					
PROT_CONF_IDX_EXT_REQ	O/V/S	0...2H	0...2	1	[-]
Selected data transmission protocol via tester(0/1 - 15765-4, 2 -14230)					
QOIL_DS_RST_KWP_1	O/V	0...FFH	0...255	1	[-]
Reset oil quality via tester					
QOIL_DS_RST_KWP_2	O/V	0...FFH	0...255	1	[-]
Reset availability via tester					
QOIL_DS_RST_KWP_3	O/V	0...FFH	0...255	1	[-]
Reset number of services via tester					
QOIL_DS_RST_KWP_4	O/V	8000...7FFFH	-327680...327670	10	[km]
Reset rest distance via tester					
QOIL_DS_RST_KWP_5	O/V	0...FFH	0...255000	1000	[km]
Reset forecast way clearance via tester					
QOIL_DS_RST_KWP_6	O/V	0...FFH	0...255	1	[-]
Reset target month via tester					
QOIL_DS_RST_KWP_7	O/V	0...FFH	0...255	1	[-]
Reset target year via tester					
QOIL_DS_RST_KWP_8	O/V	0...FFH	0...255	1	[-]
Reset forecast time clearance via tester					
QOIL_DS_RST_KWP_9	O/V	0...FFH	0...255	1	[-]
Unit base for service clear via tester					
Qv_cdherst_1	O/V/S	0...FFH	0...255	1	[-]
Codierung Hersteller 1					
Qv_cdherst_2	O/V/S	0...FFH	0...255	1	[-]
Codierung Hersteller 2					
Qv_cdherst_3	O/V/S	0...FFH	0...255	1	[-]
Codierung Hersteller 3					
Qv_cdherst_4	O/V/S	0...FFH	0...255	1	[-]
Codierung Hersteller 4					
Qv_cdherst_5	O/V/S	0...FFH	0...255	1	[-]
Codierung Hersteller 5					
Qv_cdherst_6	O/V/S	0...FFH	0...255	1	[-]
Codierung Hersteller 6					
Qv_cdherst_7	O/V/S	0...FFH	0...255	1	[-]
Codierung Hersteller 7					
Qv_cdherst_8	O/V/S	0...FFH	0...255	1	[-]
Codierung Hersteller 8					
RATIO_MMV_NS_SHIFT_DIAG_EXT_ADJ[NC_NOX_SENS_CONF]	O/V	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
External value of RATIO_MMV_NS_SHIFT_DIAG					
STATE_CMB_CTL_KWP	O/V	0...FFH	0...255	1	[-]
carrier byte state combustion control					
STATE_CMB_CTL_NVMY_KWP	O/V/S	0...FFH	0...255	1	[-]
carrier byte state combustion control non volatile saved					
STATE_HOM_AFS_REQ_EXT_ADJ	O/V	0...FFFFH	0...65535	1	[-]
raw operation mode request from tester					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_IV_CHG	O/V/S	0...FFFFH	0...65535	1	[-]
Bit coded state indicating wether injection valve was changed - 1 on bit position x indicates injector change for logical cylinder x					
STATE_MIL_ON_DIS_EXT_REQ	O/V/S	0...2H	0...2	1	[-]
request for variant "MIL_OFF" via tester					
Swmsaav	O/V	0...FFFFFFFFH	0...4294967295	1	[-]
Selektive temporäre Deaktivierung AVs bis zum nächsten Zündungswechsel					
Tvneutrin	O/V	0...FFFFH	0...655.35	0.01	[%]
Tastverhältnis Nullgangsensor vorgeben					
VS_MAX_SEL_EXT_REQ	O/V/S	0...3H	0...3	1	[-]
Selected maximum vehicle speed code via tester					
VVL_MIN_EXT_ADJ	O/V	0...FFFFH	0...65.535	0.001	[mm]
adjustment value for minimum valve lift					
VVL_MIN_NVMY_EXT_ADJ	O/V	0...FFFFH	0...65.535	0.001	[mm]
adjustment value for minimum valve lift					

Input data:

CHA_CDN_BAT	CONF_SOF_SWI	CTR_ERR_DYN_NR	CTR_KM_CAN
CTR_N_MAX	CTR_TOIL_MAX	DIST_TOIL_MAX	GEAR_TOIL_MAX
Id_bosrtak	IMOB_CONFIG_EWS4[4]	IMOB_NR_VLD_K_EWS4	K_EWS4[256]
LC_N_OFS_KWP_DISABLE	LV_ACIN	LV_AT	LV_CS_TOIL_MAX
LV_DRI	LV_ERR_TCO	LV_ES	LV_ES
LV_IGK	LV_IS	LV_REQ_ISC	LV_SENS_BAT_SMT_DET
	LV_VAR_ACIN	LV_VAR_AEB	LV_VAR_AMT
LV_VAR_ARS	LV_VAR_ASR	LV_VAR_ASR_3	LV_VAR_ASR_4
LV_VAR_BN_EFP	LV_VAR_BN_GEAR_REV	LV_VAR_BN_LDM	LV_VAR_BN_LTG_HDLP_L
LV_VAR_BN_MSW	LV_VAR_BN_TRL	LV_VAR_DCC	LV_VAR_EBOX_CFA
LV_VAR_ECRAS_DOWN	LV_VAR_ECRAS_UP	LV_VAR_EF	LV_VAR_ETCU
LV_VAR_ETCU_SPT	LV_VAR_ICL	LV_VAR_LSH_DOWN	LV_VAR_LSH_UP
LV_VAR_MAF	LV_VAR_MAF_LEARNT	LV_VAR_MSW	LV_VAR_NOX
LV_VAR_PSTE	LV_VAR_PSTE_2	LV_VAR_PSTE_3	
		LV_VAR_RLY_ACCOUT	LV_VAR_RLY_ST
LV_VAR_SAP	LV_VAR_SAV	LV_VAR_SOF	LV_VAR_TCT
LV_VAR_TQ_PBR	N_MAX	N_TOIL_MAX	NC_K_EWS_MAX
NC_NOX_SENS_CONF	NC_CYL_NR	NC_NT_NR	NT_AGI
PV_AV_TOIL_MAX	STATE_IMOB_K_EWS	STATE_PSTE_3_SRC	TAM_TOIL_MAX
TCO	TCO_TOIL_MAX	TOIL_MAX_WARN	TOIL_THD_TOIL_MAX
TQI_AV_TOIL_MAX	TRT	TRT_N_MAX	VS_TOIL_MAX

Import Actions:

ACTION_ECM3_DisableReq	ACTION_NOXM_WriteSulfurExtAdj(void)
ACTION_NOXM_WriteAgingExtAdj(void)	ACTION_NOXM_WriteAgiMExtAdj
ACTION_NOXD_WriteNSAdExtAdj	ACTION_NOXD_WriteNSGainDiagExtAdj
ACTION_NOXD_WriteNSShiftDiagExtAdj	ACTION_NOXD_WriteMMVNSExtAdj(void)

7.30.1 Overview of supported subservices

SID number	service	MSV_80	MSD_80 (6cyl)	MSD_80 (4cyl)	MSD_81 (6cyl)	MSD_81 (4cyl)	MSD85(L4)
2E_10_00	Prüfstempel schreiben	X	X	X	X	X	X
2E_10_01	BOS löschen via Tester	X	X	X	X	X	X
2E_30_00	MIL-OFF Codierung schreiben	X	X	X	X	X	X
2E_30_01	Kat Codierung schreiben	X	X	X	X	X	X
2E_30_10	VMAX Codierung schreiben	X	X	X	X	X	X
2E_30_20	OL/UL Codierung schreiben	X	X	X	X	X	X
2E_30_30	Codierung-Protokoll schreiben	X	X	X	X	X	X
2E_32_00	Oelwechselintervall-Codierdaten schreiben	X	X	X	X	X	X
2E_32_10	IGR-Codierdaten schreiben	X	X	X	X	X	X
2E_32_11	Xenonverbau - Codierdaten schreiben	X	X	X	X	X	X
2E_32_20	Codierung SPA schreiben	X	X	X	X	X	X
2E_32_30	Codierung BZE schreiben	X	X	X	X	X	X
2E_32_40	Abgasklappe Codierung schreiben	X	X	X	X	X	X
2E_32_50	Codierung MSA schreiben	-	-	X	-	X	-
2E_32_60	Codierung CDASMOTschreiben	X	X	X	X	X	X
2E_3F_FF	Aenderungsindex der Codierdaten schreiben	X	X	X	X	X	X
2E_5F_87	LaufruheVerbesserungssystem Zaehler Reset	-	X	X	X	X	-
2E_5F_88	Schreiben NOx-Adaptationswerte	-	X	X	X	X	-
2E_5F_89	Request Reset MSA-Tabellenspeicher	-	-	X	-	X	-
2E_5F_8A	Schreiben Nullgang-Lernwert	-	-	X	-	X	-
2E_5F_8B	PMB vorgeben	X	X	X	X	X	X
2E_5F_8E	MSA_DEAK schreiben	-	-	X	-	X	-
2E_5F_8F	MSA_DEAK_AV schreiben	-	-	X	-	X	-
2E_5F_90	IMAALLE schreiben	-	X	X	X	X	X
2E_5F_9x	IMA für Zylinder x schreiben	-	X	X	X	X	X
2E_5F_DE_00	VVT- Minhub Ansteuerung beenden	X	-	-	-	-	-
2E_5F_DE_07	VVT- Minhub ansteuern	X	-	-	-	-	-

2E_5F_DE_08	VVT- Minhub programmieren	X	-	-	-	-	-
2E_5F_F0_07	Abgleichswert LL ansteuern	X	X	X	X	X	X
2E_5F_F0_08	Abgleichswert LL programmieren	X	X	X	X	X	X
2E_5F_F1_00	CO- Abgleich beenden	X	-	-	-	-	-
2E_5F_F1_04	CO- Abgleich Reset	X	-	-	-	-	-
2E_5F_F1_07	CO- Abgleich ansteuern	X	-	-	-	-	-
2E_5F_F1_08	CO- Abgleich programmieren	X	-	-	-	-	-
2E_5F_F2_04	Varianten loeschen	X	X	X	X	X	X
2E_5F_F3_04	Uebertemperatursicherung loeschen	X	X	X	X	X	X
2E_5F_F4_04	Fusshebelwerk Fehlbedienung loeschen	X	X	X	X	X	-
2E_5F_F5_04	Powermanagement Histogramm loeschen	X	X	X	X	X	X
2E_5F_F6_04	Ueberdrehzahlsicherung loeschen	X	X	X	X	X	X
2E_5F_F7_00	Laufruheprüfung Vorgeben beenden	X	-	-	-	-	-
2E_5F_F7_07	Laufruheprüfung vorgeben	X	-	-	-	-	-
2E_5F_F7_08	Laufruheprüfung programmieren	X	-	-	-	-	-
2E_5F_F8_00	Homogenbetrieb vorgeben beenden	-	X	X	X	X	-
2E_5F_F8_07	Homogenbetrieb vorgeben	-	X	X	X	X	-
2E_C0_01	Write k_ews4	X	X	X	X	X	X

7.30.2 Standard application conditions

Initialisation at reset: 0, **except:** nonvolatile variables from NVRAM

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0 **or** Diagnostic timeout

Service location: see "list of implemented diagnostic services"

Standard Global negative Responses:

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	WriteDataByCommonIdentifier Request Service Id	M	#2Eh	WDBCID
#3d	General response code	M	#XX	RC

Table: WriteDataByCommonIdentifier Negative Response Message

Remark:
The detailed message of the negative response code is described in chapter "Error codes".

7.30.3 2E_10_00 Write check stamp

General information:

With this comand the 3 bytes of the check stamp are written to the ECU (only lowest 3 byte of CHK_STAMP are used in software). Those are actually in the volatile memory. In the latch off phase (n = 0 rpm and Kl.15 off) this values are written down into the non volatile memory. If the battery voltage is removed from the ECU, bevor the power latch phase ends, the storage of the check stamp is cancelled and the value up to now was kept.
The writing of the check stamp was reserved for the customers production, that means that the check stamp can´t be writen by the ECU production factory.

Because for writing CHK_STAMP NVMY programing is necessary it is written to the EEPROM during the ECU power latch. During this time the value of CHK_STAMP is buffered in RAM.

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier	M	#10h	RCI_TSP
#3d	RecordCommonIdentifier	M	#00h	RCI_TSP
#4d	recordValue#1 = CHK_STAMP Byte 2	M	#XXh	RV_PST
#5d	recordValue#2 = CHK_STAMP Byte 1	M	#XXh	RV_PST
#6d	recordValue#3 = CHK_STAMP Byte 0 (LSB)	M	#XXh	RV_PST

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response on service with RLI = #F0h

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	**WDBCIPR**
#2d	RecordCommonIdentifier	M	#10h	RCI_TSP
#3d	recordCommonIdentifier	M	#00h	RCI_TSP

Table WriteDataByCommonIdentifier Positive Response Message

7.30.4 2E_10_01 Reset BOS data

General information:

With this comand the required service data (BOS-bedarfsorientierte Servicedaten löschen) can be reseted.
The BMW module OZ/PM/.. makes a reset if the layervalue Id_bosrtt (via QOIL_DS_RST_KWP_1) is not equal 00h.

Additional Application conditions:

Recurrence: 100 ms

Formula section:

```
If(1)     service 2E_10_01 is received
then(1)
          QOIL_DS_RST_KWP_i = corresponding record value
elseif(1)

//this part has to be calculated with a recurrence of 100ms:
if        Id_bosrtak = QOIL_DS_RST_KWP_1
then      QOIL_DS_RST_KWP_1 = 0
endif
```

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier high byte	M	#10h	RCI_BOS
#3d	RecordCommonIdentifier low byte	M	#01h	RCI_BOS
#4d	recordValue#1 = reset mode (fix)	M	#01h	RV_RES
#5d	recordValue#2 = QOIL_DS_RST_KWP_1	M	#XXh	RV_Id_bosrtt
#6d	recordValue#3 = QOIL_DS_RST_KWP_2	M	#XXh	RV_Bosbtvfbkt
#7d	recordValue#4 = QOIL_DS_RST_KWP_3	M	#XXh	RV_Zrbosrt
#8d	recordValue#5 = QOIL_DS_RST_KWP_4 high byte	M	#XXh	RV_Bosrlsmt
#9d	recordValue#6 = QOIL_DS_RST_KWP_4 low byte	M	#XXh	RV_Bosrlsmt
#10d	recordValue#7 = QOIL_DS_RST_KWP_9	M	#xxh	RV_Einheit
#11d	recordValue#8 = QOIL_DS_RST_KWP_5	M	#XXh	RV_Bosrw2t
#12d	recordValue#9 = QOIL_DS_RST_KWP_6	M	#XXh	RV_Bosmzielt
#13d	recordValue#10 = QOIL_DS_RST_KWP_7	M	#XXh	RV_Bosjzielt
#14d	recordValue#11 = QOIL_DS_RST_KWP_8	M	#XXh	RV_Bosprog2t
#15d	recordValue#12 = dummy value (not used)	M	#xxh	RV_RESERVE

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	**WDBCIPR**
#2d	RecordCommonIdentifier	M	#10h	RCI_BOS
#3d	RecordCommonIdentifier	M	#01h	RCI_BOS

Table WriteDataByCommonIdentifier Positive Response Message

7.30.5 2E_30_00 MIL-OFF Codierung schreiben

General information:

With this command the MIL (malfunction indication lamp) can be switched off permanently. In case of a production ECU (active authentification/signature) the selection is only possible within the first 10 hours of ECU- runtime (TRT < 10 hours). The selection is stored in non volatile memory immediately during runnig time (because after coding the ECU is reset) and can not be changed after 10 hours. After 10 hours the ECU answers to the write- command with "subFunctionNotSupported."

Additional Application conditions:

Activation: LV_ES = 1 **and**
(TRT <= 10 hours - only if authentification/signature is active)

Deactivation: Diagnostic timeout **or**
(TRT > 10 hours - only if authentification/signature is active)
//send negative response code 12: subFunctionNotSupported

Formula section:

```
IF      C_VAR_STATE = 1
THEN    STATE_MIL_ON_DIS_EXT_REQ = 0
ELSE    IF    recordValue#1 > 2H
        THEN  send negative response code 31: requestOutOfRange
        ELSE  STATE_MIL_ON_DIS_EXT_REQ = recordValue#1
```

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSM
#3d	RecordCommonIdentifier Low Byte	M	#00h	RCI_CDSM
#4d	recordValue#1 = STATE_MIL_ON_DIS_EXT_REQ	M	#XXh	RV_MIL

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSM
#3d	RecordCommonIdentifier Low Byte	M	#00h	RCI_CDSM

Table WriteDataByCommonIdentifier Positive Response Message

Meaning of C_VAR_STATE:
0 = ECE
1 = US

7.30.6 2E_30_01 Kat Codierung schreiben

General information:

With this command automatically lerning of catalyst variants can be disabled. Catalyst variant is set to "no cat." In case of a production ECU (active authentification/signature) the selection is only possible within the first 10 hours of ECU- runtime (TRT < 10 hours). The selection is stored in non volatile memory immediately during runnig time (because after coding the ECU is reset) and can not be changed after 10 hours. After 10 hours the ECU answers to the write- command with "subFunctionNotSupported."

Additional Application conditions:

Activation: LV_ES = 1 (Engine Stop) **and**
(TRT <= 10 hours - only if authentification/signature is active)

Deactivation: Diagnostic timeout **or**
(TRT > 10 hours - only if authentification/signature is active)
//send negative response code 12: subFunctionNotSupported

Formula section:

IF recordValue#1 > 1H

THEN send negative response code 31: requestOutOfRange

ELSE LV_CAT_CONF_DIS_EXT_REQ = recordValue#1

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSO
#3d	RecordCommonIdentifier Low Byte	M	#01h	RCI_CDSO
#4d	recordValue#1 = LV_CAT_CONF_DIS_EXT_REQ	M	#XXh	RV_KATV

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSO
#3d	RecordCommonIdentifier Low Byte	M	#01h	RCI_CDSO

Table WriteDataByCommonIdentifier Positive Response Message

7.30.7 2E_30_10 VMAX Codierung schreiben

General information:

With this command one of four possible values for maximum vehicle speed is selected. In case of a production ECU (active authentification/signature) the selection is only possible within the first 10 hours of ECU- runtime (TRT < 10 hours). The selection is stored in non volatile memory immediately during runnig time (because after coding the ECU is reset) and can not be changed after 10 hours. After 10 hours the ECU answers to the write- command with "subFunctionNotSupported."

Additional Application conditions:

Activation: LV_ES = 1 (Engine Stop) **and**
(TRT <= 10 hours - only if authentification/signature is active)

Deactivation: Diagnostic timeout **or**
(TRT > 10 hours - only if authentification/signature is active)
//send negative response code 12: subFunctionNotSupported

Formula section:

IF recordValue#1 > 3H

THEN send negative response code 31: requestOutOfRange

ELSE VS_MAX_SEL_EXT_REQ = recordValue#1

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSV MAX
#3d	RecordCommonIdentifier Low Byte	M	#10h	RCI_CDSV MAX
#4d	recordValue#1 = VS_MAX_SEL_EXT_REQ	M	#XXh	RV_VMAX

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSV MAX
#3d	RecordCommonIdentifier Low Byte	M	#10h	RCI_CDSV MAX

Table WriteDataByCommonIdentifier Positive Response Message

7.30.8 2E_30_20 OL/UL Codierung schreiben

General information:

With this command one of three power configuration codes is selected. This code is taken for the plausibility check of the two engine power variants OL ("obere Leistung") and UL ("untere Leistung"). In case of a production ECU (active authentification/signature) the selection is only possible within the first 10 hours of ECU- runtime (TRT < 10 hours). The selection is stored in non volatile memory immediately during runnig time (because after coding the ECU is reset) and can not be changed after 10 hours. After 10 hours the ECU answers to the write- command with "subFunctionNotSupported."

Additional Application conditions:

Activation:	LV_ES = 1 (Engine Stop) **and** (TRT <= 10 hours - only if authentification/signature is active)
Deactivation:	Diagnostic timeout **or** (TRT > 10 hours - only if authentification/signature is active) *//send negative response code 12: subFunctionNotSupported*

Formula section:

```
IF      recordValue#1 > 3H
THEN    send negative response code 31: requestOutOfRange
ELSE    POW_CONF_IDX_EXT_REQ = recordValue#1
```

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSL
#3d	RecordCommonIdentifier Low Byte	M	#20h	RCI_CDSL
#4d	recordValue#1 = POW_CONF_IDX_EXT_REQ	M	#XXh	RV_OLUL

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSL
#3d	RecordCommonIdentifier Low Byte	M	#20h	RCI_CDSL

Table WriteDataByCommonIdentifier Positive Response Message

7.30.9 2E_30_30 Codierung Protokoll schreiben

General information:

With this command the ISO-protocol variant is selected. In case of a production ECU (active authentification/signature) the selection is only possible within the first 10 hours of ECU-runtime (TRT < 10 hours). The selection is stored in non volatile memory immediately during runnig time (because after coding the ECU is reset) and can not be changed after 10 hours. After 10 hours the ECU answers to the write- command with "subFunctionNotSupported."

Additional Application conditions:

Initialisation at first run of ECU:

PROT_CONF_IDX_EXT_REQ = C_PROT_CONF_IDX_EXT_REQ

Initialisation at reset or at first run of ECU:

```
If PROT_CONF_IDX_EXT_REQ = 2  // 14230
    Then
        LV_VAR_OBDC_CAN = 0
    Else
        LV_VAR_OBDC_CAN = 1
    Endif
```

Activation: LV_ES = 1 **and**
(TRT <= 10 hours - only if authentification/signature is active)

Deactivation: (TRT > 10 hours - only if authentification/signature is active)
//send negative response code 12: subFunctionNotSupported

Formula section:

```
If(1)       recordValue#1 > 2H
Then(1)     send negative response code 31: requestOutOfRange
Else(1)
    If(2)   C_ PROT_CONF_IDX_EXT_REQ = 0
    Then (2) PROT_CONF_IDX_EXT_REQ = recordValue#1
            If(3) PROT_CONF_IDX_EXT_REQ = 2  // 14230
              Then(3)
                LV_VAR_OBDC_CAN = 0
              Else(3)
                LV_VAR_OBDC_CAN = 1
              Endif(3)
    Else(2) send negative response subFunctionNotSupported
    End(2)
```

End(1)

WriteDataByCommonIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSL
#3d	RecordCommonIdentifier Low Byte	M	#30h	RCI_CDSL
#4d	recordValue#1 = PROT_CONF_IDX_EXT_REQ	M	#XXh	RV_OLUL

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#30h	RCI_CDSL
#3d	RecordCommonIdentifier Low Byte	M	#30h	RCI_CDSL

Table WriteDataByCommonIdentifier Positive Response Message

7.30.10 2E_32_00 Oelwechselintervall- Codierdaten schreiben

General information:

With this command two country dependant factors for oil change intervall can be coded. The selection is stored in non volatile memory immediately during runnig time (because after coding the ECU is reset).

Additional Application conditions:

Activation: LV_ES = 1

Deactivation: LV_ES =0

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#32h	RCI_CDSO
#3d	RecordCommonIdentifier Low Byte	M	#00h	RCI_CDSO
#4d	recordValue#1 = FAC_OIL_EXT_REQ_1	M	#XXh	RV_OZLF1C
#5d	recordValue#2 = FAC_OIL_EXT_REQ_2	M	#XXh	RV_OZLF2C

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#32h	RCI_CDSO
#3d	RecordCommonIdentifier Low Byte	M	#00h	RCI_CDSO

Table WriteDataByCommonIdentifier Positive Response Message

7.30.11 2E_32_10 IGR-Codierdaten schreiben

General information:

With this command the Intelligent alternator control can be coded.

Additional Application conditions:

Activation: LV_ES = 1

Deactivation: LV_ES = 0

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#32h	RCI_CDIGR
#3d	RecordCommonIdentifier Low Byte	M	#10h	RCI_CDIGR
#4d	recordValue#1 = LV_ALTER_CTL_ENA	M	#XXh	RV_B_CDIGRON

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#32h	RCI_CDIGR
#3d	RecordCommonIdentifier Low Byte	M	#10h	RCI_CDIGR

Table WriteDataByCommonIdentifier Positive Response Message

7.30.12 2E_32_11 Xenonverbau Codierdaten schreiben

General information:

With this command the mounting of xenon-filled headlights can be coded.

Additional Application conditions:

Activation: LV_ES = 1

Deactivation: LV_ES = 0

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#32h	RCI_CDXEN
#3d	RecordCommonIdentifier Low Byte	M	#11h	RCI_CDXEN
#4d	recordValue#1 = LV_LTG_GAS_ENA	M	#XXh	RV_B_CDXENON

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#32h	RCI_CDXEN
#3d	RecordCommonIdentifier Low Byte	M	#11h	RCI_CDXEN

Table WriteDataByCommonIdentifier Positive Response Message

7.30.13 2E_32_20 Codierung SPA

General information:

With this command the "Schaltpunktanzeige" can be coded.

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#32h	WDBCI_SPA_vorgeben
#3d	RecordCommonIdentifier Low Byte	M	#20h	WDBCI_SPA_vorgeben
#4d	Record value #1 = B_spa_csoll	M	#XXh	RV_SPA_CSOLL
#5d	Record value #2 = Dummy	M	#XXh	RV_DUMMY

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	WriteDataByCommonIdentifier Response Service Id	M	#6Eh	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#32h	WDBCI_SPA_vorgeben
#3d	RecordCommonIdentifier Low Byte	M	#20h	WDBCI_SPA_vorgeben

Table WriteDataByCommonIdentifier Positive Response Message

7.30.14 2E_32_30 Codierung BZE schreiben

General information:

With this command the Intelligent alternator control can be coded.

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#32h	WDBCI_BZE_vorgeben
#3d	RecordCommonIdentifier Low Byte	M	#30h	WDBCI_BZE_vorgeben
#4d	Record value #1 = Qv_cdherst_1	M	#XXh	RV_CD_HERST1
#5d	Record value #2 = Qv_cdherst_2	M	#XXh	RV_CD_HERST2
#6d	Record value #3 = Qv_cdherst_3	M	#XXh	RV_CD_HERST3
#7d	Record value #4 = Qv_cdherst_4	M	#XXh	RV_CD_HERST4
#8d	Record value #5 = Qv_cdherst_5	M	#XXh	RV_CD_HERST5
#9d	Record value #6 = Qv_cdherst_6	M	#XXh	RV_CD_HERST6
#10d	Record value #7 = Qv_cdherst_7	M	#XXh	RV_CD_HERST7
#11d	Record value #8 = Qv_cdherst_8	M	#XXh	RV_CD_HERST8

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	WriteDataByCommonIdentifier Response Service Id	M	#6Eh	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#32h	WDBCI_BZE_vorgeben
#3d	RecordCommonIdentifier Low Byte	M	#30h	WDBCI_BZE_vorgeben

Table WriteDataByCommonIdentifier Positive Response Message

7.30.15 2E_32_40 Abgasklappe Codierung schreiben

General information:

With this command the used exhaust flap will be defined.

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#32h	WDBCI_AGK_vorgeben
#3d	RecordCommonIdentifier Low Byte	M	#40h	WDBCI_AGK_vorgeben
#4d	recordValue#1 = CONF_EF	M	#XXh	RV_CODIERUNG_AGK

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#32h	WDBCI_AGK_vorgeben
#3d	RecordCommonIdentifier Low Byte	M	#40h	WDBCI_AGK_vorgeben

Table WriteDataByCommonIdentifier Positive Response Message

7.30.16 2E_32_50 MSA Codierung schreiben

General information:

With this command the use of MSA will be defined.
The selection is only possible within the first 10 hours of ECU- runtime (TRT < 10 hours and LC_STST_VAR_COD = 0) or LC_STST_VAR_COD_ENA = 1. The selection is stored in non volatile memory .
In case of LC_STST_VAR_CO_ENA = 0.no change after 10 hours is possible and the ECU answers to the write- command with "subFunctionNotSupported."

Additional Application conditions:

Activation: LV_ES = 1 **and**
(LC_STST_VAR_COD_ENA = 1
or
TRT <= 10 hours – if authentification/signature is active)

Deactivation: LV_ES = 0 **or**
(TRT > 10 hours – if authentification/signature is active
and LC_STST_VAR_COD_ENA = 0,
//send negative response code 12: subFunctionNotSupported)

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#32h	WDBCI_CDMSA
#3d	RecordCommonIdentifier Low Byte	M	#50h	WDBCI_CDMSA
#4d	recordValue#1 = LV_STST_VAR_COD (0 – without MSA, 1 – with MSA)	M	#XXh	RV_CODIERUNG_MSA

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#32h	WDBCI_CDMSA
#3d	RecordCommonIdentifier Low Byte	M	#50h	WDBCI_CDMSA

Table WriteDataByCommonIdentifier Positive Response Message

7.30.17 2E_32_60 Codierung CDASMOT schreiben

General information:

With this function the "Codierung elektrische Diagnose der Funktion Abschaltung Klemme 15" is written in NVRAM

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	Request Service ID	M	2Eh	WDBCI
2	RecordCommonIdentifier High Byte	M	32h	WDBCI_CDASMOT
3	RecordCommonIdentifier Low Byte	M	60h	WDBCI_CDASMOT
4	recordValue#1 = LV_IGK_OFF_ACK_ENA (0 - diagnosis off, 1 - diagnosis on)	M	XXh [0h, 1h]	RV_CODIERUNG_ASMOT

WriteDataByCommonIdentifier Positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
1	**Response Service ID**	**M**	**6Eh**	**RDBCIPR**
2	RecordCommonIdentifier High Byte	M	32h	WDBCI_CDASMOT
3	RecordCommonIdentifier Low Byte	M	60h	WDBCI_ CDASMOT

7.30.18 2E_3F_FF Aenderungsindex der Codierdaten schreiben

General information:

With this command the changing index of the coding data is written. The value is stored in non volatile memory immediately during runnig time (because after coding the ECU is reset).

Additional Application conditions:

Activation: LV_ES = 1
Deactivation: LV_ES = 0

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#3Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#FFh	RCI_
#4d	recordValue#1 = IDX_VAR_COD[2]	M	#XXh	RV_
#5d	recordValue#2 = IDX_VAR_COD[1]	M	#XXh	RV_

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#3Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#FFh	RCI_

Table WriteDataByCommonIdentifier Positive Response Message

7.30.19 2E_5F_87 LaufruheVerbesserungssystem Zaehler Reset

General information:

This command generate a request to reset the counter of "Laufruheverbesserungssystem" in BMW-software.

Formula section:

```
If      service 2E_5F_87 is received
then
        set B_zrlvs_clr = 1 for 2 sec.
        send positive response
endif
```

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_LVSZYL_RESET
#3d	RecordCommonIdentifier	M	#87h	WDCI_LVSZYL_RESET

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	**WDBCIPR**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_LVSZYL_RESET
#3d	recordCommonIdentifier	M	#87h	WDCI_LVSZYL_RESET

Table WriteDataByCommonIdentifier Positive Response Message

7.30.20 2E_5F_88 ADRECOVERY vorgeben

General information:

This command write informations about the Nox-trap into ECU. This is necessary after reprogramming or change of ECU.

Formula section:

```
If  service 2E_5F_88 is received
Then
        store received values into corresponding variables
        call ACTION_NOXM_WriteSulfurExtAdj(void)
        call ACTION_NOXM_WriteAgingExtAdj(void)
```

```
            call ACTION_NOXM_WriteAgiMExtAdj
            call ACTION_NOXD_WriteNSAdExtAdj
            call ACTION_NOXD_WriteNSShiftDiagExtAdj
            call ACTION_NOXD_WriteNSGainDiagExtAdj
            call ACTION_NOXD_WriteMMVNSExtAdj(void)
            send positive response
    endif
```

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*C vt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_ADRESTORE
#3d	RecordCommonIdentifier	M	#88h	WDCI_ADRESTORE
#4d	recordValue#1 = NT_SUL_32_EXT_ADJ[1] (high word, high byte)	M	#XXh	RV_NT_SUL_32[1]
#5d	recordValue#2 = NT_SUL_32_EXT_ADJ [1] (high word, low byte)	M	#XXh	RV_NT_SUL_32[1]
#6d	recordValue#3 = NT_SUL_32_EXT_ADJ [1] (low word, high byte)	M	#XXh	RV_NT_SUL_32[1]
#7d	recordValue#4 = NT_SUL_32_EXT_ADJ [1] (low word, low byte)	M	#XXh	RV_NT_SUL_32[1]
#8d	recordValue#5 = NT_SUL_32_EXT_ADJ [2] (high word, high byte)	M	#XXh	RV_NT_SUL_32[2]
#9d	recordValue#6 = NT_SUL_32_EXT_ADJ [2] (high word, low byte)	M	#XXh	RV_NT_SUL_32[2]
#10d	recordValue#7 = NT_SUL_32_EXT_ADJ [2] (low word, high byte)	M	#XXh	RV_NT_SUL_32[2]
#11d	recordValue#8 = NT_SUL_32_EXT_ADJ [2] (low word, low byte)	M	#XXh	RV_NT_SUL_32[2]
#12d	recordValue#9 = NT_SUL_H_32_EXT_ADJ [1] (high word, high byte)	M	#XXh	RV_NT_SUL_H_32[1]
#13d	recordValue#10 = NT_SUL_H_32_EXT_ADJ [1] (high word, low byte)	M	#XXh	RV_NT_SUL_H_32[1]
#14d	recordValue#11 = NT_SUL_H_32_EXT_ADJ [1] (low word, high byte)	M	#XXh	RV_NT_SUL_H_32[1]
#15d	recordValue#12 = NT_SUL_H_32_EXT_ADJ [1] (low word, low byte)	M	#XXh	RV_NT_SUL_H_32[1]
#16d	recordValue#13 = NT_SUL_H_32_EXT_ADJ [2] (high word, high byte)	M	#XXh	RV_NT_SUL_H_32[2]
#17d	recordValue#14 = NT_SUL_H_32_EXT_ADJ 2] (high word, low byte)	M	#XXh	RV_NT_SUL_H_32[2]
#18d	recordValue#15 = NT_SUL_H_32_EXT_ADJ [2] (low word, high byte)	M	#XXh	RV_NT_SUL_H_32[2]
#19d	recordValue#16 = NT_SUL_H_32_EXT_ADJ [2] (low word, low byte)	M	#XXh	RV_NT_SUL_H_32[2]
#20d	recordValue#17 = not used	M	#XXh	RV_
#21d	recordValue#18 = not used	M	#XXh	RV_
#22d	recordValue#19 = NT_AGI_SO2P_FQ_SUM_EXT_ADJ (high word, high byte)	M	#XXh	RV_NT_AGI_SO2P_FQ_SUM
#23d	recordValue#20 = NT_AGI_SO2P_FQ_SUM_EXT_ADJ (high word, low byte)	M	#XXh	RV_NT_AGI_SO2P_FQ_SUM
#24d	recordValue#21 = NT_AGI_SO2P_FQ_SUM_EXT_ADJ (low word, high byte)	M	#XXh	RV_NT_AGI_SO2P_FQ_SUM

#25d	recordValue#22 = NT_AGI_SO2P_FQ_SUM_EXT_ADJ (low word, low byte)	M	#XXh	RV_NT_AGI_SO2P_FQ_SUM
#26d	recordValue#23 = not used	M	#XXh	RV_NT_AGI_SUL
#27d	recordValue#24 = not used	M	#XXh	RV_NT_AGI_SUL
#28d	recordValue#25 = NT_AGI_SUL_SNG_EXT_ADJ [1] high word, high byte)	M	#XXh	RV_NT_AGI_SUL_SNG[1]
#29d	recordValue#26 = NT_AGI_SUL_SNG_EXT_ADJ [1] (high word, low byte)	M	#XXh	RV_NT_AGI_SUL_SNG[1]
#30d	recordValue#27 = NT_AGI_SUL_SNG_EXT_ADJ [1] (low word, high byte)	M	#XXh	RV_NT_AGI_SUL_SNG[1]
#31d	recordValue#28 = NT_AGI_SUL_SNG_EXT_ADJ [1] (low word, low byte)	M	#XXh	RV_NT_AGI_SUL_SNG[1]
#32d	recordValue#29 = NT_AGI_SUL_SNG_EXT_ADJ [2](high word, high byte)	M	#XXh	RV_NT_AGI_SUL_SNG[2]
#33d	recordValue#30 = NT_AGI_SUL_SNG_EXT_ADJ [2] (high word, low byte)	M	#XXh	RV_NT_AGI_SUL_SNG[2]
#34d	recordValue#31 = NT_AGI_SUL_SNG_EXT_ADJ [2] (low word, high byte)	M	#XXh	RV_NT_AGI_SUL_SNG[2]
#35d	recordValue#32 = NT_AGI_SUL_SNG_EXT_ADJ [2] (low word, low byte)	M	#XXh	RV_NT_AGI_SUL_SNG[2]
#36d	recordValue#33 = not used	M	#XXh	RV_
#37d	recordValue#34 = not used	M	#XXh	RV_
#38d	recordValue#35 = NT_AGI_THERMO_SNG_EXT_ADJ [1] (high byte)	M	#XXh	RV_NT_AGI_THERMO_SNG[1]
#39d	recordValue#36 = NT_AGI_THERMO_SNG_EXT_ADJ [1] (high byte)	M	#XXh	RV_NT_AGI_THERMO_SNG[1]
#40d	recordValue#37 = NT_AGI_THERMO_SNG_EXT_ADJ [2] (high byte)	M	#XXh	RV_NT_AGI_THERMO_SNG[2]
#41d	recordValue#38 = NT_AGI_THERMO_SNG_EXT_ADJ [2] (low byte)	M	#XXh	RV_NT_AGI_THERMO_SNG[2]
#42d	recordValue#39 = CTR_NT_AGI_AD_CMPL_SUM_EXT_ADJ (high byte)	M	#XXh	RV_CTR_NT_AGI_AD_CMPL_SUM
#43d	recordValue#40 = CTR_NT_AGI_AD_CMPL_SUM_EXT_ADJ (low byte)	M	#XXh	RV_CTR_NT_AGI_AD_CMPL_SUM
#44d	recordValue#41 = CTR_NT_AGI_SO2P_FQ_EXT_ADJ (high byte)	M	#XXh	RV_CTR_NT_AGI_SO2P_FQ
#45d	recordValue#42 = CTR_NT_AGI_SO2P_FQ_EXT_ADJ (low byte)	M	#XXh	RV_CTR_NT_AGI_SO2P_FQ
#46d	recordValue#43 = not used	M	#XXh	RV_
#47d	recordValue#44 = LV_NT_AFS_REQ_AGI_TMP_3_EXT_ADJ	M	#XXh	RV_LV_NT_AFS_REQ_AGI_TMP_3
#48d	recordValue#45 = not used	M	#XXh	RV_
#49d	recordValue#46 = LV_SO2P_REQ_2_EXT_ADJ	M	#XXh	RV_LV_SO2P_REQ_2
#50d	recordValue#47 = LV_SO2P_REQ_FQ_EXT_ADJ	M	#XXh	RV_LV_SO2P_REQ_FQ
#51d	ecordValue#48 = FAC_NT_AGI_LIM_EXT_ADJ (high byte)	M	#XXh	RV_FAC_NT_AGI_LIM
#52d	ecordValue#49 = FAC_NT_AGI_LIM_EXT_ADJ (low byte)	M	#XXh	RV_FAC_NT_AGI_LIM
#53	recordValue50 = FAC_NT_AGI_MDL_EXT_ADJ (high word, high byte)	M	#XXh	RV_FAC_NT_AGI_MDL_EXT_ADJ
#54	recordValue51 = FAC_NT_AGI_MDL_EXT_ADJ (high word, low byte)	M	#XXh	RV_FAC_NT_AGI_MDL_EXT_ADJ
#55	recordValue52 = FAC_NT_AGI_MDL _EXT_ADJ (low word, high byte)	M	#XXh	RV_FAC_NT_AGI_MDL_EXT_ADJ
#56	recordValue53 = FAC_NT_AGI_MDL _EXT_ADJ (low word, low byte)	M	#XXh	RV_FAC_NT_AGI_MDL_EXT_ADJ
#57	recordValue54 = DIST_NS_NEW_EXT_ADJ (high byte)	M	#XXh	RV_DIST_NS_NEW_EXT_ADJ
#58	recordValue55 = DIST_NS_NEW_EXT_ADJ (low byte)	M	#XXh	RV_DIST_NS_NEW_EXT_ADJ
#59	recordValue56 = DIST_NT_NS_SHIFT_EXT_ADJ [1] (high byte)	M	#XXh	RV_DIST_NT_NS_SHIFT_EXT_ADJ[1]

#60	recordValue57 = DIST_NT_NS_SHIFT_EXT_ADJ [1] (low byte)	M	#XXh	RV_DIST_NT_NS_SHIFT_EXT_ADJ[1]
#61	recordValue58 = CTR_NS_AD_CYL_EXT_ADJ [1] (high byte)	M	#XXh	RV_ CTR_NS_AD_CYL_EXT_ADJ[1]
#62	recordValue59 = CTR_NS_AD_CYL_EXT_ADJ [1] (low byte)	M	#XXh	RV_ CTR_NS_AD_CYL_EXT_ADJ[1]
#63	recordValue60 = DIST_NT_NS_AD_EXT_ADJ [1] (high byte)	M	#XXh	RV_DIST_NT_NS_AD_EXT_ADJ[1]
#64	recordValue61 = DIST_NT_NS_AD_EXT_ADJ [1] (low byte)	M	#XXh	RV_DIST_NT_NS_AD_EXT_ADJ[1]
#65	recordValue62 = FAC_NOX_NS_AD_EXT_ADJ [1] (high byte)	M	#XXh	RV_FAC_NOX_NS_AD_EXT_ADJ[1]
#66	recordValue63 = FAC_NOX_NS_AD_EXT_ADJ [1] (low byte)	M	#XXh	RV_FAC_NOX_NS_AD_EXT_ADJ[1]
#67	recordValue64 = CTR_NS_SHIFT_CYC_EXT_ADJ[1] (high byte)	M	XXh	RV_CTR_NS_SHIFT_CYC_EXT_ADJ[1]
#68	recordValue65 = CTR_NS_SHIFT_CYC_EXT_ADJ[1] (low byte)	M	XXh	RV_CTR_NS_SHIFT_CYC_EXT_ADJ[1]
#69	recordValue66 = RATIO_MMV_NS_SHIFT_DIAG_EXT_ADJ[1] (high byte)	M	XXh	RV_RATIO_MMV_NS_SHIFT_DIAG_EXT_ADJ[1]
#70	recordValue67 = RATIO_MMV_NS_SHIFT_DIAG_EXT_ADJ[1] (low byte)	M	XXh	RV_RATIO_MMV_NS_SHIFT_DIAG_EXT_ADJ [1]

Table WriteDataByCommonIdentifier Request Message
(Hint: In case of NC_NT_NR =1 must 0ffh transmitte for all "recorde values " with array element[2])

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	**WDBCIPR**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_ADRESTORE
#3d	recordCommonIdentifier	M	#88h	WDCI_ADRESTORE

Table WriteDataByCommonIdentifier Positive Response Message

7.30.21 2E_5F_89 Request Reset MSA-Tabellenspeicher

General information:

This command the request Reset MSA-Tabellenspeicher is send to BMW-software.

Additionnal Application conditions:

Initalisation at reset and in case of LV_IGK 1-> 0:
B_msahfkreset = 0

Formula section:

```
If      service 2E_5F_89 is received
then
        B_msahfkreset = 1
        send positive response
endif
```

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_MSA RING_HFKR ESET
#3d	RecordCommonIdentifier	M	#89h	WDCI_MSA RING_HFKR ESET WDCI_NGS

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	**WDBCIPR**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_MSA RING_HFKR ESET WDCI_NGS
#3d	recordCommonIdentifier	M	#89h	WDCI_MSA RING_HFKR ESET WDCI_NGS

Table WriteDataByCommonIdentifier Positive Response Message

7.30.22 2E_5F_8A Nullgang_Lernwert_vorgeben

General information:

This command write the Nullgang_Lernwert_vorgeben to BMW-software.

Additional Application conditions:

Initalisation at reset and in case of LV_IGK 1-> 0:
Tvneutrin = 0.0

Formula section:

```
If      service 2E_5F_8A is received
then
        Tvneutrin = corresponding received values
        send positive response
endif
```

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_NGS
#3d	RecordCommonIdentifier	M	#8Ah	WDCI_NGS
#4d	recordValue#1 = Tvneutrin (high byte)	M	#XXh	WDCI_NGS
#5d	recordValue#2 = Tvneutrin (low byte)	M	#XXh	WDCI_NGS

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	**WDBCIPR**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_NGS
#3d	recordCommonIdentifier	M	#8Ah	WDCI_NGS

Table WriteDataByCommonIdentifier Positive Response Message

7.30.23 2E_5F_8B PMB_vorgeben

General information:

This command write informations about the state of battery into ECU. This is necessary after reprogramming or change of ECU.

Formula section:

```
If      service 2E_5F_8B is received
then
        Pmrestore[7] = corresponding received values
        set B_pmrestore = 1 for 5 sec.
        send positive response
endif
```

WriteDataByCommonIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_PMB
#3d	RecordCommonIdentifier	M	#8Bh	WDCI_PMB
#4d	recordValue#1 = Pmrestore[0]	M	#XXh	RV_PMRESTORE[0]
#5d	recordValue#2 = Pmrestore[1]	M	#XXh	RV_PMRESTORE[1]

#6d	recordValue#3 = Pmrestore[2]	M	#XXh	RV_PMRESTORE[2]
#7d	recordValue#4 = Pmrestore[3]	M	#XXh	RV_PMRESTORE[3]
#8d	recordValue#5 = Pmrestore[4]	M	#XXh	RV_PMRESTORE[4]
#9d	recordValue#6 = Pmrestore[5]	M	#XXh	RV_PMRESTORE[5]
#10d	recordValue#7 = Pmrestore[6]	M	#XXh	RV_PMRESTORE[6]

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	**WDBCIPR**
#2d	RecordCommonIdentifier	M	#5Fh	WDCI_PMB
#3d	recordCommonIdentifier	M	#8Bh	WDCI_PMB

Table WriteDataByCommonIdentifier Positive Response Message

7.30.24 2E_5F_8E MSA_DEAK

General information:

With this command the deactivation of MSA can be switched on / off.
In the first 250 km the MSA can switched off permanently with service request 2E_5F_8E_08. After more then 250km the ECU answers on this request with "subFunctionNotSupported" and the deactivation of MSA with 2E_5F_8E_07 is only possible for the actual driving cylce.

Additional Application conditions:

Initialisation at reset or LV_IGK 1 > 0:

```
If      CTR_KM_CAN >249 km and LV_STST_DEAC = 1
Then    store LV_STST_DEAC = 0 in non volatile memory
Endif
B_msasw =  LV_STST_DEAC
```

Formula section:

```
if(1)     record value#1 = 0
then(1)
          LV_STST_DEAC      = 0
          B_msasw      = 0
          store LV_STST_DEAC = 0 in non volatile memory
          send positive respones
endif(1)
```

```
if(1)   record value#1 = 7
then(1)
        B_msasw = 1
        send positve response
endif(1)
if(1)   record value#1 = 8
then(1)
        if(2)       CTR_KM_CAN  > 249 km
        then(2)     LV_STST_DEAC = 0
                    B_msasw = 0
                    send negative response
        else(2)     LV_STST_DEAC =1
                    B_msasw = 1
                    send positive response
        endif(2)
        store LV_STST_DEAC  in non volatile memory
else(1)
        send negative response
endif(1)
```

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MSA_DEAK
#3d	RecordCommonIdentifier Low Byte	M	#8Eh	RCI_MSA_DEAK
#4d	Record value#1	M	#XXh	RV_RCTECU

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	WriteDataByCommonIdentifier Response Service Id	M	#6Eh	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MSA_DEAK
#3d	RecordCommonIdentifier Low Byte		#8Eh	RCI_MSA_DEAK

Table WriteDataByCommonIdentifier Positive Response Message

7.30.25 2E_5F_8F MSA_DEAK_AV

General information:

With this command the selective deactivation of "Abschaltverhinderer MSA" can be coded.

Additional Application conditions:

Initialisation at reset or LV_IGK 1 -> 0: Swmsaav = 0

Formula section:

```
if      record value#1 = 0
then
        Swmsaav = 0
        send positive response
endif

if      record value#1 = 7
then
        Swmsaav = received value
        send positve response
elseif
        send negative response

endif
```

WriteDataByCommonIdentifier Request (only service 2E_5F_8F_00)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MSA_DEAK_AV
#3d	RecordCommonIdentifier Low Byte	M	#8Fh	RCI_MSA_DEAK_AV
#4d	Record value#1	M	#00h	RV_RCTECU

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier Request (only service 2E_5F_8F_07)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MSA_DEAK_AV
#3d	RecordCommonIdentifier Low Byte	M	#8Fh	RCI_MSA_DEAK_AV
#4d	Record value#1	M	#07h	RV_STA
#5d	Record value#2 = Swmsaav (high byte , high word)	M	#XXh	RV_MSA_DEAK_AV
#6d	Record value#3 = Swmsaav (low byte , high word)	M	#XXh	RV_MSA_DEAK_AV
#7d	Record value#4 = Swmsaav (high byte , low word)	M	#XXh	RV_MSA_DEAK_AV
#8d	Record value#5 = Swmsaav (low byte , low word)	M	#XXh	RV_MSA_DEAK_AV

Table WriteDataByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	WriteDataByCommonIdentifier Response Service Id	M	#6Eh	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MSA_DEAK_AV
#3d	RecordCommonIdentifier Low Byte		#8Fh	RCI_MSA_DEAK_AV

Table WriteDataByCommonIdentifier Positive Response Message

7.30.26 2E_5F_90 IMAALLE programmieren

Description:

With this service all injection valve corrections are written indivudal for each cylinder. The values EGY_SP_IV_EXT_ADJ[NC_CYL_NR] and MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR] must be stored in **reprogramming resistant** non volatile memory. At first run of new ECU or in case of NVRAM error the output data must initialized with zero.
STATE_IV_CHG is used at engine stop to engine run.

Additional Application conditions:

```
Initialisation:  at reset: from NVRAM and from reprogramming resistant NVRAM ,
                 at exit start:   if STATE_IV_CHG <>  0
                                  then   STATE_IV_CHG = 0
                                         write NVRAM "BMW coding"
                                  endif
Activation:      LV_ES = 1
Deactivation:    LV_ES = 0
```

Formula Section:

```
If received service 2E_5F_90
then
    EGY_SP_IV_EXT_ADJ[NC_CYL_NR] = corresponding received values
    MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR] = corresponding received values
    STATE_IV_CHG( bit 0 ).. STATE_IV_CHG(bit NC_CYL_NR-1) = 1
    send positive response
endif
```

WriteDataByCommonIdentifier Request

.

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	WriteDataByCommonIdentifier Request Service Id	M	#2Eh	WDBCI
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_IMAALLE
#3d	RecordCommonIdentifier Low Byte	M	#90h	RCI_IMAALLE
#4d	RecordValue#1 = - EGY_SP_IV_EXT_ADJ[0] high byte	M	#XXh	RCI_
#5d	RecordValue#2 = - EGY_SP_IV_EXT_ADJ[0] low byte	M	#XXh	RCI_
#6d	RecordValue#3 = - MFF_ABSV_IV_EXT_ADJ[0] high byte	M	#XXh	RCI_
#7d	RecordValue#4 = - MFF_ABSV_IV_EXT_ADJ[0] low byte	M	#XXh	RCI_
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
#4*NC_CYL_NR	RecordValue#(4*NC_CYL_NR-3) = EGY_SP_IV_EXT_ADJ[NC_CYL_NR-1] high byte	M	#XXh	RCI_
#4*NC_CYL_NR +1	RecordValue#(4*NC_CYL_NR -2) = EGY_SP_IV_EXT_ADJ[NC_CYL_NR-1] low byte	M	#XXh	RCI_
#4*NC_CYL_NR +2	RecordValue#(4*NC_CYL_NR -1) = MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR-1] high byte	M	#XXh	RCI_
#4*NC_CYL_NR +3	RecordValue#(4*NC_CYL_NR) = MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR-1] low byte	M	#XXh	RCI_

Table WriteDataByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	WriteDataByCommonIdentifier Response Service Id	M	#6Eh	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_IMAALLE
#3d	RecordCommonIdentifier Low Byte	M	#90h	RCI_IMAALLE

Table WriteDataByLocalIdentifier Positive Response Message

7.30.27 2E_5F_9x IMA0x programmieren

Description:

With this service cylinder balancing injection valve corrections is written only for cylinder x.

Additional Application conditions:

Initialisation at exit start: STATE_IV_CHG = 0

Activation: LV_ES = 1

Deactivation: LV_ES = 0

Formula Section:

```
if      received service 2E_5F_9x and x<= NC_CYL_NR    // x from 2E_5F_9x
then
        EGY_SP_IV_EXT_ADJ[x-1] = received value
        MFF_ABSV_IV_EXT_ADJ[x-1] = received value
        STATE_IV_CHG(bit x-1) = 1
```

```
            send positive response
        endif
```

WriteDataByCommonIdentifier Request

.

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_IMA0x
#3d	RecordCommonIdentifier Low Byte	M	#9xh	RCI_IMA0x
#4d	RecordValue#1 = - EGY_SP_IV_EXT_ADJ[x-1] high byte	M	#XXh	RCI_
#5d	RecordValue#2= - EGY_SP_IV_EXT_ADJ[x-1] low byte	M	#XXh	RCI_
#6d	RecordValue#3= - MFF_ABSV_IV_EXT_ADJ[x-1] high byte	M	#XXh	RCI_
#7d	RecordValue#4= - MFF_ABSV_IV_EXT_ADJ[x-1] low byte	M	#XXh	RCI_

Table WriteDataByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_IMA0x
#3d	RecordCommonIdentifier Low Byte	M	#9xh	RCI_IMA0x

Table WriteDataByLocalIdentifier Positive Response Message

7.30.28 2E_5F_DE_00 VVT- Minhub Ansteuerung beenden

General information:

When this service is received, the actuator diagnosis function stops and control is given back to the ECU control.

Formula section:

WriteDataByCommonIdentifier Request

.

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_VH
#3d	RecordCommonIdentifier Low Byte	M	#DEh	RCI_VH
#4d	Inputoutputcontrolparameter	M	#00h	IOCP_RCTECU

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_VH

#3d	RecordCommonIdentifier Low Byte	M	#DEh	RCI_VH
#4d	Inputoutputcontrolparameter	M	#00h	IOCP_RCTECU

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.29 2E_5F_DE_07 VVT- Minhub ansteuern

Description:

With this service the minimum valve lift can be adjusted.

Formula section:

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MINH
#3d	RecordCommonIdentifier Low Byte	M	#DEh	RCI_MINH
#4d	Inputoutputcontrolparameter	M	#07h	IOCP_STA
#5d	inputOutputControlParameter - VVL_MIN_EXT_ADJ high byte	M	#XXh	CS_MINH
#6d	InputOutputControlParameter - VVL_MIN_EXT_ADJ low byte	M	#XXh	CS_MINH

Table inputoutputcontrolByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MINH
#3d	RecordCommonIdentifier Low Byte	M	#DEh	RCI_MINH
#4d	Inputoutputcontrolparameter	M	#07h	IOCP_STA

Table inputoutputcontrolByLocalIdentifier Positive Response Message

7.30.30 2E_5F_DE_08 VVT- Minhub programmieren

Description:

With this service the minimum valve lift can be programmed into non volatile memory after power latch phase.

Additional Application conditions:

Activation: LV_ES = 1

Deactivation: LV_ES = 0

Formula section:

WriteDataByCommonIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MINH
#3d	RecordCommonIdentifier Low Byte	M	#DEh	RCI_MINH
#4d	Inputoutputcontrolparameter	M	#08h	IOCP_LTA
#5d	inputOutputControlParameter - VVL_MIN_NVMY_EXT_ADJ high byte	M	#XXh	CS_MINH
#6d	InputOutputControlParameter - VVL_MIN_NVMY_EXT_ADJ low byte	M	#XXh	CS_MINH

Table inputoutputcontrolByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_MINH
#3d	RecordCommonIdentifier Low Byte	M	#DEh	RCI_MINH
#4d	Inputoutputcontrolparameter	M	#08h	IOCP_LTA

Table inputoutputcontrolByLocalIdentifier Positive Response Message

7.30.31 2E_5F_F0_07 Abgleichswert LL ansteuern

General information:

With the tester via KWP telegram an idle speed offset for five different conditions can be calibrated. The idle speed setpoint offsets stored into the non volatile memory are initialized with 0 as default value. In case of a tester is connected via KWP N_KWP_OFS_ACC_DRI, N_KWP_OFS_DRI, N_KWP_OFS_ACC, N_KWP_OFS and N_KWP_OFS_VB are sent and the bit “store” was sent the adjusted values are stored into non volatile memory.
In case of LV_KWP_ENA = 1 and a tester is connected via KWP the adjustment values N_KWP_OFS_ACC_DRI_KWP, N_KWP_OFS_DRI_KWP, N_KWP_OFS_KWP, N_KWP_OFS_ACC_KWP and N_KWP_OFS_VB_KWP from tester are used.

N_SP_OFS_KWP is permanently used in the idle speed module, so the calculation within the KWP-module have to do permanantly.

For the time, when the tester works on the car, the max. selection should be off for this Kl.15 cycle. Now the new setting of the offsets can be made.

Additional Application conditions:

Initialisation at reset:

N_KWP_OFS_ACC_DRI = N_KWP_OFS_ACC_DRI_NVMY

N_KWP_OFS_DRI = N_KWP_OFS_DRI_NVMY

N_KWP_OFS = N_KWP_OFS_NVMY

N_KWP_OFS_ACC = N_KWP_OFS_ACC_NVMY

N_KWP_OFS_VB = N_KWP_OFS_VB_NVMY

N_SP_OFS_KWP = 0

Activation: LV_KWP_ENA = 1

Deactivation: LV_KWP_ENA = 0

Recurrence: 10 ms

Formula section:

switch off the max-selection in this Kl.15 cycle:

```
IF          the service 2E_5F_F0_07 is received
    THEN    LV_ACT_N_SP_IS_BAS_EXT_ADJ=01h
    ELSE    LV_ACT_N_SP_IS_BAS_EXT_ADJ=00h
ENDIF
```

idle speed setpoint offset at LV_ACIN = 1 and LV_DRI = 1 (N_KWP_OFS_ACC_DRI_KWP) :

```
IF          N_KWP_OFS_ACC_DRI_KWP    ≥  C_N_OFS_ACC_DRI_MAX
THEN        N_KWP_OFS_ACC_DRI        =  C_N_OFS_ACC_DRI_MAX
    ELSEIF  N_KWP_OFS_ACC_DRI_KWP    ≤  C_N_OFS_ACC_DRI_MIN
    THEN    N_KWP_OFS_ACC_DRI        =  C_N_OFS_ACC_DRI_MIN
ELSE        N_KWP_OFS_ACC_DRI        =  N_KWP_OFS_ACC_DRI_KWP
ENDIF
```

idle speed setpoint offset at LV_ACIN = 0 and LV_DRI = 1 (N_KWP_OFS_DRI_KWP):

```
IF          N_KWP_OFS_DRI_KWP    ≥    C_N_OFS_DRI_MAX
THEN        N_KWP_OFS_DRI        =    C_N_OFS_DRI_MAX
    ELSEIF  N_KWP_OFS_DRI_KWP    ≤    C_N_OFS_DRI_MIN
    THEN    N_KWP_OFS_DRI        =    C_N_OFS_DRI_MIN
ELSE        N_KWP_OFS_DRI        =    N_KWP_OFS_DRI_KWP
ENDIF
```

idle speed setpoint offset at LV_ACIN = 0 and LV_DRI = 0 (N_KWP_OFS_KWP):

```
IF          N_KWP_OFS_KWP    ≥    C_N_OFS_MAX
THEN        N_KWP_OFS        =    C_N_OFS_MAX
    ELSEIF  N_KWP_OFS_KWP    ≤    C_N_OFS_MIN
    THEN    N_KWP_OFS        =    C_N_OFS_MIN
```

```
ELSE            N_KWP_OFS               =    N_KWP_OFS_KWP
ENDIF
```

idle speed setpoint offset at LV_ACIN = 1 and LV_DRI = 0 (N_KWP_OFS_ACC_KWP):

```
IF              N_KWP_OFS_ACC_KWP       ≥    C_N_OFS_ACC_MAX
THEN            N_KWP_OFS_ACC           =    C_N_OFS_ACC_MAX
    ELSEIF      N_KWP_OFS_ACC_KWP       ≤    C_N_OFS_ACC_MIN
    THEN        N_KWP_OFS_ACC           =    C_N_OFS_ACC_MIN
ELSE            N_KWP_OFS_ACC           =    N_KWP_OFS_ACC_KWP
ENDIF
```

idle speed setpoint offset at battery charging is active (N_KWP_OFS_VB_KWP) :

```
IF              N_KWP_OFS_VB_KWP        ≥    C_N_OFS_VB_MAX
THEN            N_KWP_OFS_VB            =    C_N_OFS_VB_MAX
    ELSEIF      N_KWP_OFS_VB_KWP        ≤    C_N_OFS_VB_MIN
    THEN        N_KWP_OFS_VB            =    C_N_OFS_VB_MIN
ELSE            N_KWP_OFS_VB            =    N_KWP_OFS_VB_KWP
ENDIF
```

idle speed setpoint - LV_KWP_ENA :

```
IF          TCO                     > C_TCO_MIN_N_OFS_KWP
    and     LV_ERR_TCO              = 0
    and     LC_N_OFS_KWP_DISABLE    = 0
    and     LV_REQ_ISC              = 1
THEN        LV_KWP_ENA              = 1
ELSE        LV_KWP_ENA              = 0
ENDIF
IF          LV_KWP_ENA = 0
THEN        no adjustment and no programming by tester is allowed
    and     the ECU send `invalid parameter` response to the tester request
ELSE        adjustment and programming by tester is allowed
ENDIF
```

Idle Speed Setpoint Offset by Tester (KWP):

```
IF              LV_DRI = 1    and    LV_ACIN = 1          (AC is ready and drive engaged)
THEN            N_SP_OFS_KWP_1 = N_KWP_OFS_ACC_DRI
    ELSEIF (1)  LV_DRI = 1    and    LV_ACIN = 0      (AC is not ready and drive engaged)
```

```
        THEN (1)      N_SP_OFS_KWP_1 = N_KWP_OFS_DRI
        ELSEIF (2)    LV_DRI = 0   and   LV_ACIN = 0   (AC is not ready and drive not engaged)
        THEN (2)      N_SP_OFS_KWP_1 = N_KWP_OFS
        ELSEIF (3)    LV_DRI = 0   and   LV_ACIN = 1   (AC is ready and drive not engaged)
        THEN (3)      N_SP_OFS_KWP_1 = N_KWP_OFS_ACC
    ELSE              N_SP_OFS_KWP_1 = 0
    ENDIF

    IF        CHA_CDN_BAT > C_ CHA_CDN_BAT_KWP_MIN          (battery charging act.)
    THEN      N_SP_OFS_KWP = N_SP_OFS_KWP_1 + N_KWP_OFS_VB
    ELSE      N_SP_OFS_KWP = N_SP_OFS_KWP_1
    ENDIF
```

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_ABLL
#3d	RecordCommonIdentifier Low Byte	M	#F0h	RCI_ABLL
#4d	Inputoutputcontrolparameter	M	#07h	IOCP_STA
#5d	InputoutputLocalValue = N_KWP_OFS_ACC_DRI_KWP Value offset drive position and air condition	M	#XXh	IOLI_
#6d	InputoutputLocalValue = N_KWP_OFS_DRI_KWP Value offset drive position	M	#XXh	IOLI_
#7d	InputoutputLocalValue = N_KWP_OFS_KWP Value offset for idle speed	M	#XXh	IOLI_
#8d	InputoutputLocalValue N_KWP_OFS_ACC_KWP Value offset air condition	M	#XXh	IOLI_
#9d	InputoutputLocalValue = N_KWP_OFS_VB_KWP Value offset voltage battery	M	#XXh	IOLV

Table inputoutputcontrolByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_ABLL
#3d	RecordCommonIdentifier Low Byte	M	#F0h	RCI_ABLL
#4d	Inputoutputcontrolparameter	M	#07h	IOCP_STA
#5d	InputoutputLocalValue = N_KWP_OFS_ACC_DRI_KWP Value offset drive position and air condition	M	#XXh	IOLI_
#6d	InputoutputLocalValue = N_KWP_OFS_DRI_KWP Value offset drive position	M	#XXh	IOLI_
#7d	InputoutputLocalValue = N_KWP_OFS_KWP Value offset for idle speed	M	#XXh	IOLI_
#8d	InputoutputLocalValue N_KWP_OFS_ACC_KWP Value offset air condition	M	#XXh	IOLI_

#9d	InputoutputLocalValue = N_KWP_OFS_VB_KWP Value offset voltage battery	M	#XXh	IOLV

Table inputoutputcontrolByLocalIdentifier Positive Response Message

7.30.32 2E_5F_F0_08 Abgleichswert LL programmieren

General information:

In a tester is connected via KWP and the bit “store” is received, the adjustment values from tester N_KWP_OFS_ACC_DRI_KWP, N_KWP_OFS, N_KWP_OFS_DRI_KWP, N_KWP_OFS_ACC_KWP and N_KWP_OFS_VB_KWP are stored into the non volatile memory. The non volatile stored values are used after reset as initialization values for idle speed setpoint offset.

Additional Application conditions:

Activation: LV_KWP_ENA = 1

Deactivation: LV_KWP_ENA = 0

Formula Section:

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_ABLL
#3d	RecordCommonIdentifier Low Byte	M	#F0h	RCI_ABLL
#4d	Inputoutputcontrolparameter	M	#08h	IOCP_LTA
#5d	InputoutputLocalValue = N_KWP_OFS_ACC_DRI_KWP Value offset drive position and air condition	M	#XXh	IOLI_
#6d	InputoutputLocalValue = N_KWP_OFS_DRI_KWP Value offset drive position	M	#XXh	IOLI_
#7d	InputoutputLocalValue = N_KWP_OFS_KWP Value offset for idle speed	M	#XXh	IOLI_
#8d	InputoutputLocalValue N_KWP_OFS_ACC_KWP Value offset air condition	M	#XXh	IOLI_
#9d	InputoutputLocalValue = N_KWP_OFS_VB_KWP Value offset voltage battery	M	#XXh	IOLV

Table inputoutputcontrolByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_ABLL
#3d	RecordCommonIdentifier Low Byte	M	#F0h	RCI_ABLL
#4d	Inputoutputcontrolparameter	M	#08h	IOCP_LTA
#5d	InputoutputLocalValue = N_KWP_OFS_ACC_DRI_KWP Value offset drive position and air condition	M	#XXh	IOLI_
#6d	InputoutputLocalValue = N_KWP_OFS_DRI_KWP Value offset drive position	M	#XXh	IOLI_

#7d	InputoutputLocalValue = N_KWP_OFS_KWP Value offset for idle speed	M	#XXh	IOLI_
#8d	InputoutputLocalValue N_KWP_OFS_ACC_KWP Value offset air condition	M	#XXh	IOLI_
#9d	InputoutputLocalValue = N_KWP_OFS_VB_KWP Value offset voltage battery	M	#XXh	IOLV

Table inputoutputcontrolByLocalIdentifier Positive Response Message

7.30.33 2E_5F_F1_00 CO-Abgleich beenden

General information:

When this service is received, the actuator diagnosis function stops and control is given back to the ECU control.

Formula Section:

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_VH
#3d	RecordCommonIdentifier Low Byte	M	#F1h	RCI_VH
#4d	Inputoutputcontrolparameter	M	#00h	IOCP_RCTECU

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F1h	RCI_
#4d	Inputoutputcontrolparameter	M	#00h	IOCP_RCTECU

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.34 2E_5F_F1_04 CO- Abgleich Reset

Description:

With this service the MFF trim value for CO- Adjust can be reset.

Additional Application conditions:

Activation: LV_VAR_LSH_UP = 0

Deactivation: LV_VAR_LSH_UP = 1

Formula section:

```
IF      service 2E_5F_F1_04 is received
THEN    LV_RST_MFF_ADD_EXT_ADJ_NVMY = 1 for 2s
ENDIF
```

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F1h	RCI_
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_

Table inputoutputcontrolByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F1h	RCI_
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_STA

Table inputoutputcontrolByLocalIdentifier Positive Response Message

7.30.35 2E_5F_F1_07 CO- Abgleich ansteuern

Description:

With this service the MFF trim value for CO- Adjust can be adjusted. This is necessary for cars without catalyst.

The value from tester is written in the variable FAC_MFF_ADD_EXT_ADJ for 1 second. Then the variable FAC_MFF_ADD_EXT_ADJ is set to 0 for 1s. After the 2 seconds, the next value can be sent. During this 2 seconds no new value from tester is accepted to avoid too fast trimming (= negative response).

Additional Application conditions:

Initialisation: FAC_MFF_ADD_EXT_ADJ = 0 %
TIMER_1 = 0 s

Activation: LV_VAR_LSH_UP = 0 **and**
LV_IS = 1 **and**
CTR_ERR_DYN_NR = 0 **and**
TCO > C_TCO_MIN_MFF_ADD_EXT_ADJ

Deactivation: not activation or timeout

Formula section:

```
IF       Service 2E_5F_F1_07 is received  and
         TIMER_1 = 0 s
THEN     initialise TIMER_1 with 2 s                 and
         FAC_MFF_ADD_EXT_ADJ = value from tester     and
         send positive response

ELSEIF   TIMER_1 = 2 s or 1s
THEN     FAC_MFF_ADD_EXT_ADJ = 0
         TIMER_1 = TIMER_1 - 1s
         send negative response "conditions not correct" when Service 2E_5F_F1_07 is
         received  again
ENDIF
```

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F1h	RCI_
#4d	Inputoutputcontrolparameter	M	#07h	IOCP_STA
#5d	inputOutputControlParameter - FAC_MFF_ADD_EXT_ADJ	M	#XXh	CS_

Table inputoutputcontrolByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F1h	RCI_
#4d	Inputoutputcontrolparameter	M	#07h	IOCP_STA

Table inputoutputcontrolByLocalIdentifier Positive Response Message

7.30.36 2E_5F_F1_08 CO- Abgleich programmieren

Description:

With this service the MFF trim value for CO- Adjust can be programmed into non volatile memory after power latch phase.

After receiving the service 2E_5F_F1_08, the flag LV_FAC_MFF_ADD_EXT_ADJ_NVMY is set to 1 for 2 seconds. The flag LV_FAC_MFF_ADD_EXT_ADJ_NVMY triggers the CO-Adjust function to save the trim value into non volatile memory after power latch phase.

Additional Application conditions:

Activation: LV_IS = 1 **and**
LV_VAR_LSH_UP = 0 **and**
CTR_ERR_DYN_NR = 0 **and**
TCO > C_TCO_MIN_MFF_ADD_EXT_ADJ

Deactivation: not activation or timeout

Formula section:

```
IF      service 2E_5F_F1_08 is received
THEN    LV_FAC_MFF_ADD_EXT_ADJ_NVMY = 1 for 2s
ENDIF
```

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F1h	RCI_
#4d	Inputoutputcontrolparameter	M	#08h	IOCP_LTA

Table inputoutputcontrolByLocalIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F1h	RCI_
#4d	Inputoutputcontrolparameter	M	#08h	IOCP_LTA

Table inputoutputcontrolByLocalIdentifier Positive Response Message

7.30.37 2E_5F_F2_04 Varianten loeschen

General information:

This service resets the bits of learned variants. Re- learning is possible depending on the learning conditions of every device (see chapter "variant coding"). If the byte on position x in the tester telegram is 01, the variant flag/variable which is dedicated to position x is set to 0 and so the variant is erased.
The variants LV_VAR_BN and LV_VAR_VEH are vehicle specific can not be reset via tester (safety reasons; see also chapter "variant coding").

Additional Application conditions:

Activation: LV_ES=1

Deactivation: LV_ES=0

Formular section:

InputOutputControlByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F2h	RCI_
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD
#5d	inputOutputControlParameter - LV_AT	M	#XXh	
#6d	inputOutputControlParameter - LV_VAR_ACIN	M	#XXh	
#7d	inputOutputControlParameter - LV_VAR_AMT	M	#XXh	
#8d	inputOutputControlParameter - LV_VAR_ARS	M	#XXh	
#9d	inputOutputControlParameter - LV_VAR_ASR	M	#XXh	
#10d	inputOutputControlParameter - not used	M	#FFh	
#11d	inputOutputControlParameter - LV_VAR_BN_MSW	M	#XXh	
#12d	inputOutputControlParameter - LV_VAR_DCC	M	#XXh	
#13d	inputOutputControlParameter - LV_VAR_EBOX_CFA	M	#XXh	
#14d	inputOutputControlParameter - LV_VAR_ETCU	M	#XXh	
#15d	inputOutputControlParameter - LV_VAR_ICL	M	#XXh	
#16d	inputOutputControlParameter - LV_VAR_MSW	M	#XXh	
#17d	inputOutputControlParameter - LV_VAR_PSTE	M	#XXh	
#18d	inputOutputControlParameter - LV_VAR_SOF	M	#XXh	
#19d	inputOutputControlParameter - CONF_SOF_SWI	M	#XXh	
#20d	inputOutputControlParameter - LV_VAR_BN_GEAR_REV	M	#XXh	
#21d	inputOutputControlParameter - not used	M	#FFh	
#22d	inputOutputControlParameter – not used	M	#FFh	
#23d	inputOutputControlParameter - LV_VAR_SAP (not MSD80/81), not used (MSD80/81)	M	[#XXh, #FFh]	
#24d	inputOutputControlParameter - LV_VAR_EF	M	#XXh	
#25d	inputOutputControlParameter LV_VAR_ECRAS_UP	M	#XXh	
#26d	inputOutputControlParameter - LV_VAR_RLY_ACCOUT (not MSD85) not supported (MSD85)	M	[#XXh, #FFh]	
#27d	inputOutputControlParameter - LV_VAR_SAV (not MSDS80/81), not used (MSD80/81)	M	[#XXh, #FFh]	
#28d	inputOutputControlParameter - LV_VAR_RLY_ST	M	#XXh	
#29d	inputOutputControlParameter - LV_VAR_ASR_3	M	#XXh	
#30d	inputOutputControlParameter - LV_VAR_BN_LDM	M	#XXh	
#31d	inputOutputControlParameter - LV_VAR_BN_LTG_HDLP_L	M	#XXh	
#32d	inputOutputControlParameter - LV_VAR_LSH_DOWN	M	#XXh	
#33d	inputOutputControlParameter - LV_VAR_LSH_UP	M	#XXh	
#34d	inputOutputControlParameter - LV_VAR_ASR_4	M	#XXh	
#35d	inputOutputControlParameter - LV_VAR_MAF and LV_VAR_MAF_LEARNT	M	#XXh	
#36d	inputOutputControlParameter - LV_VAR_PSTE_2 and LV_VAR_PSTE_3 and STATE_PSTE_3_SRC	M	#XXh	
#37d	inputOutputControlParameter - LV_VAR_BN_EFP	M	#XXh	
#38d	inputOutputControlParameter - LV_SENS_BAT_SMT_DET	M	#XXh	
#39d	inputOutputControlParameter - LV_VAR_BN_TRL	M	#XXh	
#40d	inputOutputControlParameter - not supported	M	#FFh	

#41d	inputOutputControlParameter – not supported	M	#FFh	
#42d	inputOutputControlParameter - LV_VAR_NOX (only MSD80/81) not supported (other)	M	[#XXh, #FFh]	
#43d	inputOutputControlParameter - not supported	M	#FFh	
#44d	inputOutputControlParameter – LV_VAR_ETCU_SPT	M	#XXh	
#45d	inputOutputControlParameter – LV_VAR_TCT	M	#XXh	
#46d	inputOutputControlParameter – LV_VAR_AEB	M	#XXh	
#47d	inputOutputControlParameter – LV_VAR_TQ_PBR	M	#XXh	

Table InputOutputControlByCommonIdentifier request message

InputOutputControlByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F2h	RCI_
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.38 2E_5F_F3_04 Uebertemperatursicherung loeschen

General information:

The service sets the values for over temperature protection to default values.

Formula section:

IF service 2E_5F_F3_04 is received

THEN
CTR_TOIL_MAX = 00h
TOIL_MAX_WARN = 00h
VS_TOIL_MAX = 00h
N_TOIL_MAX = 00h
GEAR_TOIL_MAX = 00h
TQI_AV_TOIL_MAX = 00h
TAM_TOIL_MAX = 00h
TCO_TOIL_MAX = 00h
DIST_TOIL_MAX = 00h
PV_AV_TOIL_MAX = 00h
LV_CS_TOIL_MAX = 00h
TOIL_THD_TOIL_MAX = 00h

ELSE the values keep unchanged

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_UET
#3d	RecordCommonIdentifier Low Byte	M	#F3h	RCI_UET
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_UET
#3d	RecordCommonIdentifier Low Byte	M	#F3h	RCI_UET
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.39 2E_5F_F4_04 Fusshebelwerk Fehlbedienung loeschen

General information:

no description

Additional Application conditions:

Activation: timeout time < 5s

Deactivation: timeout time >= 5s

Formula section:

```
IF          service 2E_5F_F4_04 is received
THEN        timeout time = 0s
                    LV_FHW_RST = 01h
ELSE        increment timeout time
ENDIF

IF          timeout time >= 5s
THEN        LV_FHW_RST = 00h
ELSE        LV_FHW_RST reains unchanged
ENDIF
```

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_FHW
#3d	RecordCommonIdentifier Low Byte	M	#F4h	RCI_FHW
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_FHW
#3d	RecordCommonIdentifier Low Byte	M	#F4h	RCI_FHW
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.40 2E_5F_F5_04 Powermanagement Histogramm loeschen

General information:

With this comand the values described in the chapter “Power Management” are set to default values.

All values are initialized with their default values (compare detailed specifications).

After the timeout time of 5s, the function ist stopped, and the value for LV_POW_MNG_HIS_RST is set to 00h;

Additional Application conditions:

Activation: timeout ime < 5s

Deactivation: timeout time >= 5s

Formula section:

```
IF        service 2E_5F_F5_04 is received
THEN      timeout time = 0s
                LV_POW_MNG_HIS_RST = 01h
ELSE      increment timeout time
ENDIF

IF        timeout time >= 5s
THEN      LV_POW_MNG_HIS_RST = 00h
ELSE      LV_POW_MNG_HIS_RST    reains unchanged
ENDIF
```

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_HISR
#3d	RecordCommonIdentifier Low Byte	M	#F5h	RCI_HISR
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_HISR
#3d	RecordCommonIdentifier Low Byte	M	#F5h	RCI_HISR
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.41 2E_5F_F6_04 Ueberdrehzahlsicherung loeschen

General information:

Detection of an overreving event and non-volatile storage of:

- highest engine speed reached N_MAX
- associated total running time in case of highest engine speed reached TRT_N_MAX
- event frequency counter CTR_N_MAX

An overreving event starts at exceeding the engine speed C_N_LIM_MIN and active engine speed limitation (LV_N_MAX = 1).

At each overreving event, the event frequency counter is incremented by 1. The highest engine speed reached N_MAX and the associated total running time in case of highest engine speed reached TRT_N_MAX is only overwritten if a higher engine speed was reached.

For this service authenticity is necessary.

Additional Application conditions:

Activation: LV_ES = 1 **and**
authentification is necessary

Deactivation: LV_ES= 0

Formula section:

IF service 2E_5F_F6_04 is received

THEN N_MAX = 00h
TRT_N_MAX = 00h
CTR_N_MAX = 00h

ELSE the values keep unchanged

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**

#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_OVR
#3d	RecordCommonIdentifier Low Byte	M	#F6h	RCI_OVR
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_OVR
#3d	RecordCommonIdentifier Low Byte	M	#F6h	RCI_OVR
#4d	Inputoutputcontrolparameter	M	#04h	IOCP_RTD
#5d	Verify byte = 01h = ok Verify byte = 02h = not ok	M	#XXh	IOCP

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.42 2E_5F_F7_00 Laufruheprüfung Vorgeben beenden

General information:

no description

Formula section:

IF service 2E_5F_F7_00 is received

THEN set STATE_CMB_CTL_KWP = 00h

ELSE STATE_CMB_CTL_KWP remains unchanged

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_LRP
#3d	RecordCommonIdentifier Low Byte	M	#F7h	RCI_LRP
#4d	Inputoutputcontrolparameter	M	#00h	RV_RCTECU

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_LRP
#3d	RecordCommonIdentifier Low Byte	M	#F7h	RCI_LRP

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.43 2E_5F_F7_07 Laufruheprüfung vorgeben

General information:

no description

Formula section:

IF one of the bytes in the telegram is > 1h

THEN send negative response code 31: requestOutOfRange

ELSE built STATE_CMB_CTL_NVMY_KWP from telegram

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_LRP
#3d	RecordCommonIdentifier Low Byte	M	#F7h	RCI_LRP
#4d	Inputoutputcontrolparameter	M	#07h	RV_STA
#5d	recordValue#1 = STATE_CMB_CTL_KWP Bit 0	M	#XXh	STAT_GEMISCHEINGRIFF_INAKTIV
#6d	recordValue#2 = STATE_CMB_CTL_KWP Bit 1	M	#XXh	STAT_ZUENDWINKELEINGRIFF_INAKTIV
#7d	recordValue#3 = STATE_CMB_CTL_KWP Bit 2	M	#XXh	STAT_MINHUBEINGRIFF_INAKTIV
#8d	recordValue#4 = STATE_CMB_CTL_KWP Bit 3	M	#XXh	STAT_HUBEINGRIFF_INAKTIV
#9d	recordValue#5 = dummy value fix	M	#00h	RV_LRP_BIT4
#10d	recordValue#6 = dummy value fix	M	#00h	RV_LRP_BIT5
#11d	recordValue#7 = dummy value fix	M	#00h	RV_LRP_BIT6
#12d	recordValue#8 = dummy value fix	M	#00h	RV_LRP_BIT7

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_LRP
#3d	RecordCommonIdentifier Low Byte	M	#F7h	RCI_LRP

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.44 2E_5F_F7_08 Laufruheprüfung programmieren

General information:

no description

Additional Application conditions:

Initialisation: STATE_CMB_CTL_NVMY_KWP at clearing NVMY with 00h else from NVMY

Formula section:

IF one of the bytes in the telegram is > 1h

THEN send negative response code 31: requestOutOfRange

ELSE built STATE_CMB_CTL_NVMY_KWP from telegram

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_LRP
#3d	RecordCommonIdentifier Low Byte	M	#F7h	RCI_LRP
#4d	Inputoutputcontrolparameter	M	#08h	RV_LTA
#5d	recordValue#1 = STATE_CMB_CTL_NVMY_KWP Bit 0	M	#XXh	STAT_GEMISCHEINGRIFF_INAKTIV
#6d	recordValue#2 = STATE_CMB_CTL_NVMY_KWP Bit 1	M	#XXh	STAT_ZUENDWINKELEINGRIFF_INAKTIV
#7d	recordValue#3 = STATE_CMB_CTL_NVMY_KWP Bit 2	M	#XXh	STAT_MINHUBEINGRIFF_INAKTIV
#8d	recordValue#4 = STATE_CMB_CTL_NVMY_KWP Bit 3	M	#XXh	STAT_HUBEINGRIFF_INAKTIV
#9d	recordValue#5 = dummy value fix	M	#00h	RV_LRP_BIT4
#10d	recordValue#6 = dummy value fix	M	#00h	RV_LRP_BIT5
#11d	recordValue#7 = dummy value fix	M	#00h	RV_LRP_BIT6
#12d	recordValue#8 = dummy value fix	M	#00h	RV_LRP_BIT7

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_LRP
#3d	RecordCommonIdentifier Low Byte	M	#F7h	RCI_LRP

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.45 2E_5F_F8_00 Homogenbetrieb vorgeben beenden

General information:

When this service is received, the actuator diagnosis function stops and control is given back to the ECU control.

Formula section:

IF service 2E_5F_F8_00 is received

THEN STATE_HOM_AFS_REQ_EXT_ADJ = 00h

ELSE STATE_HOM_AFS_REQ_EXT_ADJ remains unchanged

ENDIF

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_VH
#3d	RecordCommonIdentifier Low Byte	M	#F8h	RCI_VH
#4d	Inputoutputcontrolparameter	M	#00h	IOCP_RCTECU

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F8h	RCI_
#4d	Inputoutputcontrolparameter	M	#00h	IOCP_RCTECU

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.46 2E_5F_F8_07 Homogen vorgeben

General information:

When this service is received, the request for operation mode is sended to the module "operation mode request from tester".

Additional Application conditions:

Initialisation at reset or at LV_IGK 1 -> 0: STATE_HOM_AFS_REQ_EXT_ADJ = 0

***Activation**:* LV_IS = 1

Deactivation: LV_IS = 0

Formula section:

WriteDataByCommonIdentifier Request

.

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F8h	RCI_
#4d	Inputoutputcontrolparameter	M	#07h	IOCP_
#5d	recordValue #1 = STATE_HOM_AFS_REQ_EXT_ADJ (high byte)	M	#XXh	IOCP_
#6d	recordValue #2 = STATE_HOM_AFS_REQ_EXT_ADJ(low byte)	M	#XXh	IOCP_

Table inputoutputcontrolByCommonIdentifier Request Message

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#5Fh	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#F8h	RCI_
#4d	Inputoutputcontrolparameter	M	#07h	IOCP_

Table inputoutputcontrolByCommonIdentifier Positive Response Message

7.30.47 2E_C0_01 Write k_ews4 (STEUERN_EWS)

General information:

IMMO: Write STEUERN_EWS4 (Save secret key to ECU)

A negativ response is possible by a wrong parameter and formula section.

Formula section:

Only for boot sw-version:

```
If(1)    K_EWS4[0..255] > 0x0 and K_EWS4[0..255] < 0xFF (Blank Check Page 0)
Then(1)        "Negative Response: KWP_RC_CNCORSE"
               break
Else(1)
    If(2)      "record value#1"  = 0x3
    Then(2)     IMOB_KWP_K_EWS4 [0..15] = "record value#2..#17"
                If(2a) IMOB_KWP_K_EWS4 [0-15] != 0xFF or 0x0 (all data not 0 or max.)
               Then(2a)     IMOB_KWP_K_EWS4_TMP0[0-15] =
                            IMOB_KWP_K_EWS4[0-15] xor 0xFF
                            "Positive Response"
```

```
                Else(2a)      "Negative Response
                Endif(2a)
          Else(2)
            If(3)    "record value#1" = 0x1
            Then(3)
              If(4)    IMOB_KWP_K_EWS4[0..15] !=
                            IMOB_KWP_K_EWS4_TMP0[0..15] xor 0xFF
              Then(4)    "Negative Response: KWP_RC_CNCORSE"
                           break
               Else(4)   "Positive Response"
                         Flash:
                                 IMOB_KWP_K_EWS4[0..15]
                                 IMOB_KWP_K_EWS4_1[0..15] xor 0xFF
                                 IMOB_KWP_K_EWS4_2[0..15 ]xor 0xAA
                Endif(4)
             Else(3)     "Negative Response: KWP_RC_SFNS_IF"
             Endif(3)
            Endif(2)
        Endif(1)
```

Only for ECU sw-version:

```
If(1)   STATE_IMOB_K_EWS = NC_K_EWS_UNLOCKED
        and IMOB_NR_VLD_K_EWS4 < NC_K_EWS_MAX
Then(1)
        If(2)    "record value#1" = 0x3
        Then(2)
                IMOB_KWP_K_EWS4 [0..15] =  "record value#2..#17"
                If(2a) IMOB_KWP_K_EWS4 [0-15] != 0xFF or 0x0 (all data not 0 or max.)
                Then(2a)        IMOB_KWP_K_EWS4_TMP0[0-15] =
                                IMOB_KWP_K_EWS4[0-15] xor 0xFF
                                send  "Positive Response"
                Else(2a)        send  "Negative Response
                Endif(2a)
                ACTION_ECM3_DisableReq (disable MU for storage)

        Else(2)  If(3)   "record value#1" = 0x1
```

```
            Then(3)
                If(4)  IMOB_KWP_K_EWS4[0..15] !=
                           IMOB_KWP_K_EWS4_TMP0[0..15] xor 0xAA
                Then(4)      "Negative Response: KWP_RC_CNCORSE"
                             break
                Else(4)      send "Positive Response"
                             Flash:  k_ews4[at next free block]
                                       (storage place depending on
                                       imob_nr_vld_k_ews4)
                                     IMOB_KWP_K_EWS4[0-15]
                                     IMOB_KWP_K_EWS4_1[0-15] xor 0xFF
                                     IMOB_KWP_K_EWS4_2[0-15] xor 0xAA
                             Flash:  imob_config_ews4[0-3]
                                     imob_config_ews4_1[0-3] xor 0xFF
                                     imob_config_ews4_2[0-3] xor 0xAA

                Endif(4)
            Else(3)         "Negative Response: KWP_RC_SFNS_IF"
            Endif(3)
        Endif(2)
    Else(1)         "Negative Response: KWP_RC_CNCORSE"
    Endif(1)
```

WriteDataByCommonIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Request Service Id**	**M**	**#2Eh**	**WDBCI**
#2d	RecordCommonIdentifier	M	#C0h	RCI_TSP
#3d	RecordCommonIdentifier	M	#01h	RCI_TSP
#4d	recordValue#1 = IMOB_KWP_MODE_BYTE	M	#xxh	RV_PST
#5d	recordValue#2 = IMOB_KWP_K_EWS4[0]	M	#xxh	RV_PST
#6d	recordValue#3 = IMOB_KWP_K_EWS4[1]	M	#xxh	RV_PST
#7d	recordValue#4 = IMOB_KWP_K_EWS4[2]	M	#xxh	RV_PST
#8d	recordValue#5 = IMOB_KWP_K_EWS4[3]	M	#xxh	RV_PST
#9d	recordValue#6 = IMOB_KWP_K_EWS4[4]	M	#xxh	RV_PST
#10d	recordValue#7 = IMOB_KWP_K_EWS4[5]	M	#xxh	RV_PST
#11d	recordValue#8 = IMOB_KWP_K_EWS4[6]	M	#xxh	RV_PST
#12d	recordValue#9 = IMOB_KWP_K_EWS4[7]	M	#xxh	RV_PST
#13d	recordValue#10 = IMOB_KWP_K_EWS4[8]	M	#xxh	RV_PST
#14d	recordValue#11 = IMOB_KWP_K_EWS4[9]	M	#xxh	RV_PST
#15d	recordValue#12 = IMOB_KWP_K_EWS4[10]	M	#xxh	RV_PST
#16d	recordValue#13 = IMOB_KWP_K_EWS4[11]	M	#xxh	RV_PST

#17d	recordValue#14 = IMOB_KWP_K_EWS4[12]	M	#xxh	RV_PST
#18d	recordValue#15 = IMOB_KWP_K_EWS4[13]	M	#xxh	RV_PST
#19d	recordValue#16 = IMOB_KWP_K_EWS4[14]	M	#xxh	RV_PST
#20d	recordValue#17 = IMOB_KWP_K_EWS4[15]	M	#xxh	RV_PST

WriteDataByCommonIdentifier positive Response

WriteDataByCommonIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**WriteDataByCommonIdentifier Response Service Id**	**M**	**#6Eh**	WDBCIPR
#2d	RecordCommonIdentifier High Byte	M	#C0h	RCI_
#3d	RecordCommonIdentifier Low Byte	M	#01h	RCI_

Table inputoutputcontrolByCommonIdentifier Positive Response Message

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VAR_STATE	1	0...FFH	0...255	1	[-]
Country variant of the dataset					
C_PROT_CONF_IDX_EXT_REQ	1	0...2H	0...2	1	[-]
init value for selected data transmission protocol via tester(0-- 15765, change with tester allowed 1 -15765, change with tester not allowed2 -14230, change with tester not allowed)					
LC_N_OFS_KWP_DISABLE	1	0...1H	0...1	1	[-]
disable bit for idle speed setpoint offset by tester					
LC_STST_VAR_COD_ENA	1	0...1H	0...1	1	[-]
enable STST-coding after TRT >= 10H (0- disable, 1 - enable)					
C_CHA_CDN_BAT_KWP_MIN	1	0...FFH	0...127	0.4980392	[Ah]
minimum threshold for active battery charging due to enable conditions of idle speed offset by tester					
C_N_OFS_ACC_DRI_MAX	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset maximum threshold at LV_ACIN = 1 and LV_DRI = 1					
C_N_OFS_ACC_DRI_MIN	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset minimum threshold at LV_ACIN = 1 and LV_DRI = 1					
C_N_OFS_DRI_MAX	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset maximum threshold at LV_ACIN = 0 and LV_DRI = 1					
C_N_OFS_DRI_MIN	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset minimum threshold at LV_ACIN = 0 and LV_DRI = 1					
C_N_OFS_MAX	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset maximum threshold at LV_ACIN = 0 and LV_DRI = 0					
C_N_OFS_MIN	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset minimum threshold at LV_ACIN = 0 and LV_DRI = 0					
C_N_OFS_ACC_MAX	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset maximum threshold at LV_ACIN = 1 and LV_DRI = 0					
C_N_OFS_ACC_MIN	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset minimum threshold at LV_ACIN = 1 and LV_DRI = 0					
C_N_OFS_VB_MAX	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset maximum threshold at battery charging is active					
C_N_OFS_VB_MIN	1	80...7FH	-256...254	2	[rpm]
idle speed setpoint offset minimum threshold at battery charging is active					
C_TCO_MIN_N_OFS_KWP	1	0...FEH	-48...142.5	0.75	[°C]
Minimum TCO threshold for activate idle speed offset					
C_TCO_MIN_MFF_ADD_EXT_ADJ	0	0...FEH	-48...142.5	0.75	[°C]
Minimum TCO for CO- adjustment via tester					

7.31 SID 32h: stopRoutineByLocalIdentifier Service

General information:

All implemented SID 32h services are described below.

Input data:

LV_IGK	LV_POW_MNG_MES_MOD	LV_INH_LAM_KWP	

Import action

void ACTION_NOXM_StopCatHeatDesu(void)

7.31.1 Overview of supported subservices

SID number	service	Service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85(8cyl)
32_2F	Stop Desulfatisierung Fahrbetrieb	ECU	-	X	X	X	X	-
32_D9	stop switch off Lambda controller	ECU	-	X	X	X	X	X
32_F0	stop checksum calculation for Inca PC	ECU	X	-	-	-	-	-
32_F1	stop programming of calibration data for IncaPC	ECU	X	-	-	-	-	-
32_F6	stop default values PM for BN-SS	ECU	-	X	X	X	X	X
32_F7	switch Intelligente Generatorregelung active	ECU	X	X	X	X	X	X

7.31.2 Global negative Responses

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	stopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table StopRoutineByLocalIdentifier Negative Response Message

Remark:
The detailed message of the negative response code is described in chapter "Introduction".

7.31.3 32_2F stop Desulfatisierung Fahrbetrieb

General information:

With this command the "Desulfatisierung Fahrbetrieb" is switch off.

Application conditions:

Initialisation: ---

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formular section:

```
If      service 32_2F       is received

Then    call ACTION_NOXM_StopCatHeatDesu()
        Send positive response
Endif
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier	M	#2Fh	RELI_DESFAHR

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier	M	#2Fh	RELI_DESFAHR

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.31.4 32_D9 stop switch off Lambda controller

General information:

With this command the switch off of lambda controller by KWP2000 is ended.

Application conditions:

Initialisation: ---

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formular section:

IF service 32_D9 is received

THEN LV_INH_LAM_KWP = 00h

ELSE LV_INH_LAM_KWP remains unchanged

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier	M	#D9h	RELI_

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier	M	#D9h	RELI_

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.31.5 32_F0 stop checksum calculation for Inca PC

General information:

With this service the tester stops the checksum calculation, which is started with the service 31_F0.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: Kl.15 off
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = checksum calculation for IncaPC	M	#F0h	RELI_

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Positive Response**	**M**	**#72h**	**SPRBLIPR**
#2d	RoutineLocalIdentifier = checksum calculation for IncaPC	M	#F0h	RELI_

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.31.6 32_F1 stop programming of calibration data for IncaPC

General information:

With this service the tester deletes the request for programming of calibration data after the power latch, which is started with the service 31_F1.

Once the programming of calibration data is started after the power latch it can not be interrupted.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = checksum calculation for IncaPC	M	#F1h	RELI_

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = checksum calculation for IncaPC	M	#F1h	RELI_

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.31.7 32_F6 stop default values PM for BN-SS

General information:

With this comand the values described in the chapter “Power Management” are set to default values.

The value LV_POW_MNG_MES_MOD is saved after a powerlatch phase into the non volatile memory.

Application conditions:

Initialisation: at reset with B_fapmmess from NVM

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formular section:

IF service 32_F6 is received

THEN LV_POW_MNG_MES_MOD = 00h

ELSE LV_POW_MNG_MES_MOD remains unchanged

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier	M	#F6h	RELI_

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier	M	#F6h	RELI_

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.31.8 32_F7 switch Intelligente Generatorregelung active

General information:

With this comand the intelligent altenator control is switched active again.

Application conditions:

Initialisation: from NVMY

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

Formular section:

IF service 32_F7 is received

THEN LV_ALTER_CTL_EXT_ADJ = 00h

ELSE LV_ALTER_CTL_EXT_ADJ remains unchanged

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier	M	#F7h	RELI_IGR_OFF

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier	M	#F7h	RELI_IGR_OFF

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.32 SID 33h: requestRoutineResultsByLocalIdentifier Service

General information:

All implemented SID 33h services are described below.

Input data:

B_diagigr	B_maregdk_ad	B_maregdk_ad	B_mareghub_ad
B_psragr_ad	Eco_jobstat1	Eco_result1	INH_IV_KWP
Klann_test1	Klann_test2	Krnn_test	LV_CH_SO2P_WOUT_LIM
LV_IGK	LV_INH_LAM_KWP	LV_KWP_PROG_DATA	LV_POW_MNG_MES_MOD
LV_SO2P_REQ_2	Mrnn_test_dk	Mrnn_test_pr	Mrnn_test_vvt
NT_SUL	Prnn_test_agr		

7.32.1 Overview of supported subservices:

SID number	service	Service location	MSV_80	MSD_80 (4cyl)	MSD_80 (6cyl)	MSD_81 (4cyl)	MSD_81 (6cyl)	MSD85(8cyl)
33_25	**read INH_IV_KWP - Diagnos Status EV-Ausblendung**	**ECU**	**-**	**X**	**X**	**X**	**X**	**X**
33_2B	**Read result Ruhestrom Test**	**ECU**	**X**	**X**	**X**	**X**	**X**	**X**
33_2F	**Auslesen Desulfatisierung Fahrbetrieb**	**ECU**	**-**	**X**	**X**	**X**	**X**	**-**
33_D9	**Read Status: switch off lambda controller**	**ECU**	**-**	**X**	**X**	**X**	**X**	**X**
33_E0	**Auslesen Eisy- Adaptionswerte (ungedrosselt)**	**ECU**	**X**	**-**	**-**	**-**	**-**	**-**
33_E1	**Auslesen Eisy- Adaptionswerte (gedrosselt)**	**ECU**	**X**	**X**	**X**	**X**	**X**	**X**
33_E2	**Auslesen Eisy- Adaptionswerte (gedrosselt mit Abgasrueckfuehrung)**	**ECU**	**-**	**X**	**X**	**X**	**X**	**X**
33_E3	**Auslesen Krann- Adaptionswerte**	**ECU**	**X**	**X**	**X**	**X**	**X**	**X**
33_E4	**Auslesen Klann-Adaptionswerte**	**ECU**	**X**	**X**	**X**	**X**	**X**	**X**
33_E5	**Auslesen AGR-Adaptionswerte**	**ECU**	**-**	**X**	**X**	**X**	**-**	**-**
33_F0	**Report results of checksum calculation for INCA PC**	**ECU**	**X**	**-**	**-**	**-**	**-**	**-**
33_F1	**Report results of programming the calibration data for INCA PC in the power latch phase**	**ECU**	**X**	**-**	**-**	**-**	**-**	**-**
33_F6	**Report Status: Steuern Messsemode stop default value**	**ECU**	**-**	**X**	**X**	**X**	**X**	**X**
33_F7	**report results of Intelligente Generatorregelung**	**ECU**	**X**	**X**	**X**	**X**	**X**	**X**

7.32.2 Global negative Responses

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

Remark:
The detailed message of the negative response code is described in chapter "Introduction".

7.32.3 33_25 Read INH_IV_KWP -diagnosis status EV-Ausblendung - requestRoutineResultsByLocalIdentifier

General information:

With this service the tester can ask for the results of the -diagnosis status EV-Ausblendung

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#25h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier	M	#25h	RELI_
#3d	RecordValue#3 = INH_IV_KWP	M	#XXh	RRS_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.4 33_2B Read result Ruhestrom Test

General information:

With this service the tester can ask for the results of "Ruhestromprüfung mit IBS"

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#2Bh	RELI_RUHESTROM

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier	M	#2Bh	RELI_
#3d	RecordValue#1 = Eco_jobstat1	M	#XXh	RRS_FS_RUHESTROM
#4d	RecordValue#2 = Eco_result1 (high byte)	M	#XXh	RRS_RUHESTROM
#5d	RecordValue#3 = Eco_result1 (low byte)	M	#XXh	RRS_RUHESTROM

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.5 33_2F Read result Desulfatisierung Fahrbetrieb

General information:

With this service the tester can ask for the results of "Desulfatisierung Fahrbetrieb"

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#2Fh	RELI_DESFAHR

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier	M	#2Fh	RELI_DESFAHR
#3d	RecordValue#1 = LV_CH_SO2P_WOUT_LIM	M	#XXh	RRS_ LV_CH_SO2P_WOUT_LIM
#4d	RecordValue#2 = NT_SUL (high byte)	M	#XXh	RRS_NT_SUL
#5d	RecordValue#3 = NT_SUL (low byte)	M	#XXh	RRS_NT_SUL
#6d	RecordValue#4 = LV_SO2P_REQ_2			

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.6 33_D9 Read status: switch off lambda controller-requestRoutineResultsByLocalIdentifier

General information:

With this service the tester can ask for switch off lambda controller by KWP-service.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#D9h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier	M	#D9h	RELI_
#3d	RecordValue#1 = LV_INH_LAM_KWP	M	#XXh	RRS_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.7 33_E0 Auslesen Eisy- Adaptionswerte (ungedrosselt)

General information:

With this service the tester can ask for the results of the Eisy- adaption.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#E0h	RELI_EISYUGD

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier	M	#E0h	RELI_EISYUGD
#3d	RecordValue#1 = B_mareghub_ad	M	#XXh	RRS_FS_ EISYUGD
#4d	RecordValue#2 = Mrnn_test_vvt high byte	M	#XXh	RRS_ MRNN_TEST_VVT
#5d	RecordValue#3 = Mrnn_test_vvt low byte	M	#XXh	RRS_ MRNN_TEST_VVT

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.8 33_E1 Auslesen Eisy- Adaptionswerte (gedrosselt)

General information:

With this service the tester can ask for the results of the Eisy- adaption.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#E1h	RELI_EISYGD

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifierValue	M	#E1h	RELI_EISYGD
#3d	RecordValue#1 = B_maregdk_ad	M	#XXh	RRS_FS_ EISYGD
#4d	RecordValue#2 = Mrnn_test_dk high byte	M	#XXh	RRS_ MRNN_TEST _DK
#5d	RecordValue#3 = Mrnn_test_dk low byte	M	#XXh	RRS_ MRNN_TEST _DK

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.9 33_E2 Auslesen Eisy- Adaptionswerte (gedrosselt mit Abgasrueckfuehrung)

General information:

With this service the tester can ask for the results of the Eisy- adaption.

__Application conditions:__

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#E2h	RELI_EISYGDAGR

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifierValue	M	#E2h	RELI_EISYGDAGR
#3d	RecordValue#1 = B_maregdk_ad	M	#XXh	RRS_FS_EISYGD
#4d	RecordValue#2 = Mrnn_test_pr high byte	M	#XXh	RRS_MRNN_TEST_PR
#5d	RecordValue#3 = Mrnn_test_pr low byte	M	#XXh	RRS_MRNN_TEST_PR

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.10 33_E3 Auslesen Krann- Adaptionswerte

__General information:__

With this service the tester can ask for the results of the KRANN- adaption.

__Application conditions:__

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#E3h	RELI_KRANN

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifierValue	M	#E3h	RELI_KRANN
#3d	RecordValue#1 = dummy value fix	M	#FFh	RRS_FS_KRANN
#4d	RecordValue#2 = Krnn_test high byte	M	#XXh	RRS_KRNN_TEST
#5d	RecordValue#3 = Krnn_test low byte	M	#XXh	RRS_ KRNN_TEST

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.11 33_E4 Auslesen Klann- Adaptionswerte

General information:

With this service the tester can ask for the results of the KLANN- adaption.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#E4h	RELI_KLANN

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifierValue	M	#E4h	RELI_KLANN
#3d	RecordValue#1 = dummy value fix	M	#FFh	RRS_FS_KLANN
#4d	RecordValue#2 = Klann_test1 high byte	M	#XXh	RRS_KLNN_READ_AD_1
#5d	RecordValue#3 = Klann_test1 low byte	M	#XXh	RRS_KLANN_READ_AD_1
#6d	RecordValue#2 = Klann_test2 high byte	M	#XXh	RRS_KLNN_READ_AD_2
#7d	RecordValue#3 = Klann_test2 low byte	M	#XXh	RRS_KLANN_READ_AD_2

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.12 33_E5 Auslesen AGR - Adaptionswerte

General information:

With this service the tester can ask for the results of the AGR- adaption.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#E5h	RELI_EISYAGR

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifierValue	M	#E5h	RELI_EISYAGR
#3d	RecordValue#1 = B_psragr_ad	M	#FFh	RRS_FS_EISYAGR
#4d	RecordValue#2 = Prnn_test_agr high byte	M	#XXh	RRS_PRNN_TEST_AGR
#5d	RecordValue#3 = Prnn_test_agr low byte	M	#XXh	RRS_PRNN_TEST_AGR

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.13 33_F0 Report results of checksum calculation for IncaPC

General information:

With this service the tester can ask for the results of the checksum calculation, which is started with the service 31_F0.

If the checksum is equal to the checksum, which is given by 31_F0, the ECU reports a positive response. Otherwise a negative response is sent.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = checksum calculation for IncaPC	M	#F0h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response (= equal checksums)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier = checksum calculation for IncaPC	M	#F0h	RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.14 33_F1 Report results of “programming the calibration data in the power latch” of IncaPC

General information:

With this service the tester can ask for the results of the programming process of the calibration data in the power latch, which is started with the service 31_F1

Please consider that the calibration data are programmed at the end of the power latch phase. That means the ECU is not able to send the positive response for finishing the programming of the calibration data.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = programming of calibration data in the power latch phase for IncaPC	M	#F1h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response (= equal checksums)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier = programming of calibration data in the power latch phase for IncaPC	M	#F1h	RELI_

#3d	Programming of calibration data not yet requested:
	If (lv_kwp_prog_data = 0) => status = #84h
	IGK is still on:
	If (lv_kwp_prog_data=1 AND lv_igk=1) => status = #80h
	ECU will start the programming in the power latch phase:
	If (lv_kwp_prog_data=1 AND lv_igk=0) => status = #82h
		M	#XXh	

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.15 33_F6 Read status: PM-Messmode- requestRoutineResultsByLocalIdentifier

General information:

With this service the tester can ask the state of PM-Messmode

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier	M	#F6h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier	M	#F6h	RELI_
#3d	RecordValue#1 = LV_POW_MNG_MES_MOD	M	#XXh	RRS_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.32.16 33_F7 Report results of Intelligente Generatorregelung

General information:

With this service the tester can ask for the status “Intelligente Generatorregelung”, which is started with the service 31_F7

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off)
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = Intelligente Generatorregelung	M	#F7h	RELI_IGR_OFF

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response (= equal checksums)

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier = Intelligente Generatorregelung	M	#F7h	RELI_IGR_OFF
#3d	RecordValue#1 = #05h **if** B_diagigr = 1 or #00h **if** B_diagigr = 0	M	#XXh	PRS_FS_IGR_OFF

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.33 Services for EOL (end of line) tests

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_gentestanf	O/V	0...1H	0...1	1	[-]
Bedingung Testeransteuerung Generator					
B_testpoelsys	O/V	0...1H	0...1	1	[-]
Flag indicating "Diagnosefunktion Oeldruckregelung"					
LAMB_SP_EXT_ADJ[NC_CBK_EX_NR]	O/V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Bank selective external lambda-setpoint (EOL test)					
LV_ACT_ECRAS_EOL_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Bit indicating activated ECRAS System test					
LV_ACT_N_SP_IS_EXT_ADJ	O/V	0...1H	0...1	1	[-]
LV for displace N_SP_IS					
LV_ACT_SA_EOL	O/V	0...1H	0...1	1	[-]
Activation bit for secondary air EOL test					
LV_ACT_VLS_EOL_EXT_ADJ	O/V	0...1H	0...1	1	[-]
LV for start EOL test: Lambda sensors confused / not wired					
LV_EOL_CPS	O/V	0...1H	0...1	1	[-]
LV for the end of line test CPS					
LV_LAM_AD_EXT_ADJ	O/V	0...1H	0...1	1	[-]
Request of "Sperre Tankentlueftung"					
LV_LAMB_SP_EXT_ADJ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag to activate bank selective external lambda-setpoint (EOL test)					
LV_POIL_EXT_ADJ_ACT	O/V	0...1H	0...1	1	[-]
Flag indicating active oil pressure diagnosis function					
LV_NT_SO2P_EXT_ADJ_REQ_NOT_STOP	O/V	0...1H	0...1	1	[-]
request STOP desulfation by tester (0 -stop)					
LV_NT_SO2P_EXT_ADJ_REQ_ST	O/V	0...1H	0...1	1	[-]
request START desulfation by tester (1-start)					
LV_TCHA_DIAG_EXT_REQ	O/V	0...1H	0...1	1	[-]
external request by tester for diagnosis of ATL					
N_SP_IS_EXT_ADJ	O/V	0...1FE0H	0...8160	1	[rpm]
idle speed setpoint from the tester					
STATE_EOL_KWP_CPS	O/V	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H	NOT_START ST_INH PAR_NOT_PLAUS WAIT_REL UNDEF ACT END_WOUT_RESULT ABORTED END_WOUT_ERR END_WITH_ERR	1	[-]
State variable for EOL test: CPS					

STATE_EOL_KWP_DMTL	O/V	0H	NOT_START	1	[-]
		1H	ST_INH		
		2H	PAR_NOT_PLAUS		
		3H	WAIT_REL		
		4H	UNDEF		
		5H	ACT		
		6H	END_WOUT_RESULT		
		7H	ABORTED		
		8H	END_WOUT_ERR		
		9H	END_WITH_ERR		
State variable for end of line test DMTL					
STATE_EOL_KWP_ECRAS	O/V	0H	NOT_START	1	[-]
		1H	ST_INH		
		2H	PAR_NOT_PLAUS		
		3H	WAIT_REL		
		4H	UNDEF		
		5H	ACT		
		6H	END_WOUT_RESULT		
		7H	ABORTED		
		8H	END_WOUT_ERR		
		9H	END_WITH_ERR		
State variable for EOL test: Ecras system test					
STATE_EOL_KWP_N_SP_IS	O/V	0H	NOT_START	1	[-]
		1H	ST_INH		
		2H	PAR_NOT_PLAUS		
		3H	WAIT_REL		
		4H	UNDEF		
		5H	ACT		
		6H	END_WOUT_RESULT		
		7H	ABORTED		
		8H	END_WOUT_ERR		
		9H	END_WITH_ERR		
State variable for displace N_SP_IS					
STATE_EOL_KWP_SA	O/V	0H	NOT_START	1	[-]
		1H	ST_INH		
		2H	PAR_NOT_PLAUS		
		3H	WAIT_REL		
		4H	UNDEF		
		5H	ACT		
		6H	END_WOUT_RESULT		
		7H	ABORTED		
		8H	END_WOUT_ERR		
		9H	END_WITH_ERR		
Status of SA EOL diagnosis					

STATE_EOL_KWP_VLS	O/V	0H	NOT_START	1	[-]
		1H	ST_INH		
		2H	PAR_NOT_PLAUS		
		3H	WAIT_REL		
		4H	UNDEF		
		5H	ACT		
		6H	END_WOUT_RESULT		
		7H	ABORTED		
		8H	END_WOUT_ERR		
		9H	END_WITH_ERR		
State variable for EOL test: Lambda sensors confused / not wired					
STATE_EOL_KWP_VVL_AD	O/V	0H	NOT_START	1	[-]
		1H	ST_INH		
		2H	PAR_NOT_PLAUS		
		3H	WAIT_REL		
		4H	UNDEF		
		5H	ACT		
		6H	END_WOUT_RESULT		
		7H	ABORTED		
		8H	END_WOUT_ERR		
		9H	END_WITH_ERR		
State variable for EOL - VVL adaptation					
STATE_NT_SO2P_EXT_ADJ	O/V	0H	FAST	1	[-]
		1H	ALTERNATING		
Additional information for desulfation mode by external request from tester					

Hint for MSV80: In STATE_EOL_KWP_... is 0H = ACT and 5H = NOT_START

Input data:

AMP	Atlsvc_dpvdk1	Atlsvc_dpvdk2	Atlsvc_dpvdk3
B_kupp	C_N_MAX_KWP	C_TCO_MIN_VLS_EOL	C_TEG_CAT_DOWN_EOL
C_VS_MAX_KWP	CTL_SHIFT_LOCK_CAN	CTR_NS_SHIFT_CYC_NOT_VLD[NC_NOX_SENS_CONF]	CTR_NS_SHIFT_CYC_VLD[NC_NOX_SENS_CONF]
CUR_DMTL_COR_FIL_EOL	CUR_DMTL_REF_LEAK_EOL	CUR_DMTL_ROUGH_LEAK_MIN_EOL	EGY_DEW_END_NT
EGY_DEW_END_NT_INT	Gangi	GEAR	GEAR_INFO
Genitest_tol	Geniutest_ab	I_gentest	LAMB_LS_UP_AFL_EOL[NC_CBK_EX_NR]
LAMB_LS_UP_AFR_EOL[NC_CBK_EX_NR]	LAMB_SP[NC_CBK_EX_NR]	LAMB_SP_SA_EOL	LAMB_SP_VLS_EOL[NC_CBK_EX_NR]
LC_SA_SWI_ACQ	LV_ACT_VLS_EOL	LV_AT	LV_BRAKE_DET
LV_CAN_TEMP_MIN_THD[NC_NOX_SENS_CONF]			
LV_CDN_VB_MIN_DIAG	LV_CLU_SWI	LV_CONF_DMTL	LV_CS
LV_CUR_DMTL_REF_DIF_MAX	LV_CUR_DMTL_THD_DIF_MES	LV_DET_FUC_OPEN	LV_DET_REFU
LV_DIAG_ACT_INH_LS_UP_DOWN[NC_CBK_EX_NR]	LV_ECRAS_DOWN_EXT_ADJ	LV_ECRAS_EOL_INH	LV_ECRAS_UP_EXT_ADJ
LV_ENA_LEAK_DMTL	LV_END_DIAG_DIAGCPS	LV_END_DIAG_SA_SAFM	LV_END_DIAG_SA_SAP
LV_END_DIAG_SA_SAV	LV_END_DIAG_SA_SYS	LV_EOL_CPS_ERR	LV_ERR_AMP
LV_ERR_AMP_PLAUS	LV_ERR_CHG_LS_UP	LV_ERR_CRK	LV_ERR_DIAGCPS
LV_ERR_ECRAS_EOL	LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_FTL_MIN	LV_ERR_LOAD_TPS_PLAUS

LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_ERR_MAF	LV_ERR_PUT_EL	LV_ERR_SA_SAFM
LV_ERR_SA_SAP	LV_ERR_SA_SAV	LV_ERR_SA_SAV_LSL	LV_ERR_SA_SYS
LV_ERR_SAP	LV_ERR_SAV	LV_ERR_TCO	LV_FCT_LIH_SA
LV_IGK	LV_INH_DIAGCPS	LV_IS	LV_LAM_AD_ENA
LV_LAM_AD_END	LV_LAMB_LS_UP_VLD[NC_CBK_EX_NR]	LV_LS_DOWN_READY[NC_CBK_EX_NR]	LV_MAF_SP_TQI_DYW_DIAGCPS
LV_MIS_STATE_A	LV_MIS_STATE_B	LV_PL	LV_ST_END
LV_T_DMTL_MAX	LV_TEMP_MAX_SAP_COIL	LV_VAR_AMT	LV_VAR_TCT
LV_VB_DIF_MAX	LV_VB_JUMP	LV_VB_RANGE_DMTL	MAF_INT_MIN_VLS_EOL
N	N_32	NC_NOX_SENS_CONF	Nkw
Nkw_poel_soll	NT_SUL	OPM_AV	OPM_AV_DIAGCPS
P_oel_ist	P_oel_soll	Pu	PV
Pwg_ist	SAF_DIAG_MAX	SAF_DIAG_MIN	St_alsvc
St_atlsvc_pvdk	St_gentest	St_testpoelsys	St_testpoelsys2
STATE_DIAG_SA_LS	STATE_DIAG_SA_SAFM	STATE_DIAGCPS	STATE_DMTL
STATE_DMTL_EOL	STATE_ECRAS_SYS	STATE_KWP_SO2P	STATE_KWP_SRV_AD_REQ_VVL
STATE_LSH_DOWN[NC_CBK_EX_NR]	STATE_LSH_UP[NC_CBK_EX_NR]	STATE_EOL_KWP_NS_SHIFT_DIAG[NC_NOX_SENS_CONF]	STATE_NT_SO2P_EXT_ADJ_ACT
STATE_NS_SHIFT_DIAG[NC_NOX_SENS_CONF]	STATE_VLS_EOL	SUM_AFL_VLS_DIAG_SA_1	SUM_AFL_VLS_DIAG_SA_2
SUM_DIAG_DIAGCPS_EOL	SUM_FLOW_SP_DIAGCPS_EOL	T_NS_SHIFT_DEAC_TEMP[NC_NOX_SENS_CONF]	T_NS_SHIFT_WAIT_REP[NC_NOX_SENS_CONF]
T_NT_SO2P_EXT_ADJ_ACT	TAM	TCO	TEG_CAT_DOWN_MDL
TEG_WALL_NT_DOWN_MDL	TIA	Tmot	Tn_abstell
TNT_MDL_L	TNT_MDL_MV	Toel	U_gentest
V	VLS_DOWN_AFL_EOL[NC_CBK_EX_NR]	VLS_DOWN_AFR_EOL[NC_CBK_EX_NR]	VS

Imported Actions:

ACTION_NOXD_StartNNShiftDiag()
ACTION_NOXD_EndNNShiftDiag()

7.33.1 Overview of implemented subservices

SID number	service	MSV_80	MSD_80 (4-cyl)	MSD_80 (6-cyl)	MSD_81 (4-cyl)	MSD_81 (6-cyl)	MSD85 (L4)
21_20	**Read Status EOL: Secondary air system**	**X**	**-**	**-**	**-**	**-**	**-**
21_22	**Read Status EOL: CPS**	**X**	**-**	**-**	**-**	**-**	**-**

21_26	Read Status EOL: Idle speed setpoint displace	X	-	-	-	-	-
21_27	Read Status EOL: VVL adaption	X	-	-	-	-	-
21_D5	Read Status EOL: ECRAS system test	X	X	X	X	X	X
21_DA	Read Status EOL: DMTL	X	-	-	-	-	-
21_DF	Read Status EOL: Lambda sensors confused / wrong wired	X	-	-	-	-	-
31_20	EOL Test ON: Secondary air system						
31_22	EOL Test ON: CPS	X	X	X	X	X	X
31_26	EOL Test ON: Idle speed setpoint displace	X	X	X	X	X	X
31_27	start VVL Adaptation	X	-	-	-	-	-
31_2A	Start intelligent alternator test	X	X	X	X	X	X
31_2C	Start EOL test: Oil Pressure Regulation Diagnosis	X	X	X	X	X	-
31_2D	Start EOL test: Desulfatisierung	-	X	X	X	X	-
31_33	Start EOLtest: NOX-SensorShiftDiagnosis	-	X	X	X	X	-
31_CF	Start : Sperre Tankentlueftung	-	X	X	X	X	-
31_D0	Start: Werkstattdiagnose ATL	-	X	X	X	X	X
31_D5	EOL Test ON: ECRAS system test	X	X	X	X	X	X
31_DA	EOL Test ON: DMTL	X	X	X	X	X	X
31_DF	EOL Test ON: Lambda sensors confused / wrong wired	X	X	X	X	X	X
32_20	EOL Test OFF: Secondary air system	X	-	-	-	-	X
32_22	EOL Test OFF: CPS	X	X	X	X	X	X
32_26	EOL Test OFF: Idle speed setpoint displace	X	X	X	X	X	X
32_27	Stop EOL - VVL Adaptation	X	-	-	-	-	-
32_2A	Stop intelligent alternator test	X	X	X	X	X	X
32_2C	Stop EOL test: Oil Pressure Regulation Diagnosis	X	X	X	X	X	-
32_2D	Stop EOL test: Desulfatisierung	-	X	X	X	X	-
32_33	Stop EOLtest: NOX-SensorShiftDiagnosis	-	X	X	X	X	-
32_CF	Stop : Sperre Tankentlueftung	-	X	X	X	X	-
32_D0	Stop: Werkstattdiagnose ATL	-	X	X	X	X	X
32_D5	Stop EOL: ECRAS system test	X	X	X	X	X	X
32_DA	EOL Test OFF: DMTL	X	X	X	X	X	X
32_DF	EOL Test OFF: Lambda sensors confused / wrong wired	X	X	X	X	X	X
33_20	Report EOL results: Secondary air system	X	-	-	-	-	X
33_22	Report EOL results: CPS	X	X	X	X	X	X
33_26	Report EOL results: Idle speed setpoint displace	X	X	X	X	X	X
33_27	Read Status EOL: VVL adaptation	X	-	-	-	-	-
33_2A	Read Status: inteligent alternator test	X	X	X	X	X	X
33_2C	Read Status EOL test: Oil Pressure Regulation Diagnosis	X	X	X	X	X	-
33_2D	Read status EOL test: Desulfatisierung	-	X	X	X	X	-
33_33	Read Status EOL-test: NOX-SensorShiftDiagnosis	-	X	X	X	X	-

33_CF	Read Statrus : Sperre Tankentlueftung	-	X	X	X	X	-
33_D0	Read Status: Werkstattdiagnose ATL	-	X	X	X	X	X
33_D5	Read Status EOL: ECRAS system test	X	X	X	X	X	X
33_DA	Report EOL results: DMTL	X	X	X	X	X	X
33_DF	Report EOL results: Lambda sensors confused / wrong wired	X	X	X	X	X	X

7.33.2 Lambda- setpoint for external adjustment

FUNCTION DESCRIPTION:

General information:

This function is for shifting the lambda-setpoint in the lambda coordination to fulfill requests of external adjustment / EOL diagnosis.

i=1 for cylinder bank 1 and i = 2 for cylinder bank 2

Description:

The lambda-setpoint LAMB_SP_EXT_ADJ[i] is calculated as long as the activation bits:

- LV_ACT_VLS_EOL *//Lambda setpoint shifting for lambda sensor EOL test*
- LV_ACT_SA_EOL *//Lambda setpoint shifting for secondary air EOL test*

are active. The bit are set/reset by customer tool.

Application conditions:

Initialisation: LV_LAMB_SP_EXT_ADJ[i] = 0
LAMB_SP_EXT_ADJ[i] = 1
at LV_IGK =1 **or** ECU reset

Recurrence: 20ms

Activation: LV_IGK = 1

Formula section:

Lambda setpoint shifting for lambda sensor EOL test

If LV_ACT_VLS_EOL = 1
Then LV_LAMB_SP_EXT_ADJ[i]= 1 *//bank 1 and bank 2*
LAMB_SP_EXT_ADJ[i] = LAMB_SP_VLS_EOL[i]

Lambda setpoint shifting for secondary air EOL test

```
ElseIf    LV_ACT_SA_EOL = 1
Then      LV_LAMB_SP_EXT_ADJ_[i] = 1          //bank 1 and bank 2
          LAMB_SP_EXT_ADJ[i] =  LAMB_SP_SA_EOL

Lambda setpoint shifting passive
Else      LV_LAMB_SP_EXT_ADJ_[i] = 0          //bank 1 and bank 2
          LAMB_SP_EXT_ADJ[i] =  1
Endif
```

7.33.3 Services 21_20 / 31_20 / 32_20 / 33_20 for end of line test: SA-system test

FUNCTION DESCRIPTION:

General information:

The EOL test is performed in order to check the SA system in the factory or in the workshop. If EOL test is active (LV_ACT_EOL_SA = 1) then secondary air function (Pump and Valve) and the configurated diagnosis (depending of LC_SA_SWI_ACQ) is activated as long as EOL is requested or diagnosis is finished.

Description:

The EOL test contain the services:
- **21_20 - Read Status EOL- test**
- **31_20 - EOL Test start**
- **32_20 - EOL Test stop**
- **33_20 - Report EOL test results**

Application conditions:

Initialisation:	at LV_IGK 0->1 or reset STATE_EOL_KWP_SA = NOT_START LV_ACT_SA_EOL = 0
Recurrence:	same as SA function
Activation:	see following chapter
Service location:	ECU-SW

7.33.3.1 Secondary air system diagnosis through SAF-meter (SAFM)

Description:

This service is only performed if **LC_SA_SWI_ACQ = 1**

7.33.3.1.1 21_20 Read Status EOL- test – SA-system - ReadDataByLocalIdentifier service

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = SA-system test	M	#20h	RELI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier = SA-system test	M	#20h	RELI_
#3d	RecordValue = STATE_EOL_KWP_SA [**MSV70**:function active , **MSx80**:function not started yet conditions not correct parameter not correct funciton waiting not used **MSV70:** function not started yet, **MSx80**: function active function finished, no result function stopped function finished, no error function finished, error present]	M	#XXh [#00h, #01h, #02h #03h #04h #05h, #06h #07h #08h #09h]	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	readDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	responseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

Formula section:

The determination of STATE_EOL_KWP_SA is stopped if EOL-test is stopped by tester request (service 32_20). In this case STATE_EOL_KWP_SA = NOT_START).

```
If      LV_ACT_SA_EOL = 1
Then    If      LV_IS = 0                       or
                LV_FCT_LIH_SA = 1               or
                LV_VB_JUMP = 1                  or
                LV_TEMP_MAX_SAP_COIL = 1        or
                LV_ERR_MAF = 1                  or
                LV_ERR_AMP = 1                  or
                LV_ERR_AMP_PLAUS = 1            or
                LV_ERR_TCO = 1                  or
                LV_ERR_CRK = 1                  or
                LV_ERR_FTL_MIN = 1              or
                LV_MIS_STATE_A = 1              or
                LV_MIS_STATE_B = 1              or
                LV_ERR_SAV = 1                  or
                LV_ERR_SAP = 1                  or
                LV_ERR_LOAD_TPS_PLAUS = 1
        Then    LV_ACT_SA_EOL = 0
                STATE_EOL_KWP_SA = 1H           //conditions not correct
        Elseif  LV_ERR_SA_SAFM = 1
        Then    LV_ACT_SA_EOL = 0
                STATE_EOL_KWP_SA = 9H           //function finished, error present
        Elseif  LV_ERR_SA_SAV = 1               or
                LV_ERR_SA_SYS = 1
        Then    LV_ACT_SA_EOL = 0
                STATE_EOL_KWP_SA = 9H           //function finished, error present
        Elseif  LV_END_DIAG_SA_SAV =1           and
                LV_END_DIAG_SA_SYS =1           and
                LV_END_DIAG_SA_SAFM =1          and
```

```
                  LV_ERR_SA_SAV = 0             and
                  LV_ERR_SA_SYS = 0             and
                  LV_ERR_SA_SAFM = 0
            Then  LV_ACT_SA_EOL = 0
                  STATE_EOL_KWP_SA = 8H         //function finished, no error
            Endif
Else        no change
Endif
```

7.33.3.1.2 31_20 EOL Test start - SA-system test - StartRoutineByLocalIdentifier service

Formula section:

```
If          request 31_20 is received       and
            LV_IS = 1                       and
            TCO > C_TCO_MIN_SA_EOL          and
            LV_VB_JUMP = 0                  and
            LV_TEMP_MAX_SAP_COIL = 0        and
            LV_ERR_SA_SAFM = 0              and
            LV_ERR_SA_SYS = 0               and
            LV_ERR_SA_SAV = 0               and
            LV_ERR_MAF = 0                  and
            LV_ERR_AMP = 0                  and
            LV_ERR_AMP_PLAUS = 0            and
            LV_ERR_TCO = 0                  and
            LV_ERR_CRK = 0                  and
            LV_ERR_FTL_MIN = 0              and
            LV_MIS_STATE_A = 0              and
            LV_MIS_STATE_B = 0              and
            LV_ERR_SAV = 0                  and
            LV_ERR_SAP = 0                  and
            LV_ERR_LOAD_TPS_PLAUS = 0       and
            (STATE_DIAG_SA_SAFM = SAFM_DIAG_END or SAFM_DIAG_CNL)
Then        LV_ACT_SA_EOL = 1
            STATE_EOL_KWP_SA = ACT          //active
Else        STATE_EOL_KWP_SA = 1H           //conditions not correct
Endif
```

StartRoutineByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= start EOL Test – SA-system test	M	#20h	RELI_

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = EOL Test SA-system test	M	#20h	RELI_
#3d	routineEntryStatus = EOL Test active	M	ACT	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	startRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	responseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

7.33.3.1.3 32_20 EOL Test OFF - SA-system test - StopRoutineByLocalIdentifier service

Formula section:

```
If      request 32_20 is received
Then    LV_ACT_SA_EOL = 0
        STATE_EOL_KWP_SA = NOT_START     //function not started
Endif
```

StopRoutineByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = stop EOL Test SA-system test	M	#20h	RELI_

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = stop EOL Test SA-system test	M	#20h	RELI_
#3d	routineEntryStatus = EOL Test inactive	M	NOT_START	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	StopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table StopRoutineByLocalIdentifier Negative Response Message

7.33.3.1.4 33_20 - Report EOL results - SA-system test - RequestRoutineResultsByLocalIdentifier service

RequestRoutineResultsByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = ask for results SA-system test	M	#20h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

Not for MSD80:RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = SA-system test	M	#20h	RELI_
#3d	RecordValue = STATE_DIAG_SA_SAFM	M	#XXh	RELI_
#4d	RecordValue = SAF_DIAG_MIN high byte	M	#XXh	RELI_
#5d	RecordValue = SAF_DIAG_MIN low byte	M	#XXh	RELI_
#4d	RecordValue = SAF_DIAG_MAX high byte	M	#XXh	RELI_
#5d	RecordValue = SAF_DIAG_MAX low byte	M	#XXh	RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

Only for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = SA-system test	M	#20h	RELI_
#3d	RecordValue = STATE_EOL_KWP_SA	M	#XXh	RELI_
#4d	RecordValue = STATE_DIAG_SA_SAFM	M	#XXh	RELI_
#5d	RecordValue = SAF_DIAG_MIN high byte	M	#XXh	RELI_
#6d	RecordValue = SAF_DIAG_MIN low byte	M	#XXh	RELI_
#7d	RecordValue = SAF_DIAG_MAX high byte	M	#XXh	RELI_
#8d	RecordValue = SAF_DIAG_MAX low byte	M	#XXh	RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

7.33.3.2 Secondary air system diagnosis through lambda sensor (LISL)

Description:

This service is only performed if **LC_SA_SWI_ACQ = 0**

7.33.3.2.1 21_20 Read Status EOL- test – SA-system - ReadDataByLocalIdentifier service

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = SA-system test	M	#20h	RELI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier = SA-system test	M	#20h	RELI_
#3d	RecordValue = STATE_EOL_KWP_SA [**MSV70**:function active , **MSx80**:function not started yet conditions not correct parameter not correct funciton waiting not used **MSV70:** function not started yet, **MSx80**: function active function finished, no result function stopped function finished, no error function finished, error present]	M	#XXh [#00h, #01h, #02h #03h #04h #05h, #06h #07h #08h #09h]	RV_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	readDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	responseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

Formula section:

The determination of STATE_EOL_KWP_SA is stopped if EOL-test is stopped by tester request (service 32_20). In this case STATE_EOL_KWP_SA = NOT_START.

If LV_ACT_SA_EOL = 1

Then **If** LV_IS = 0 **or**

LV_FCT_LIH_SA = 1 **or**

```
                LV_VB_JUMP = 1                  or
                LV_TEMP_MAX_SAP_COIL = 1        or
                LV_ERR_AMP = 1                  or
                LV_ERR_AMP_PLAUS = 1            or
                LV_ERR_TCO = 1                  or
                LV_ERR_CRK = 1                  or
                LV_ERR_FTL_MIN = 1              or
                LV_MIS_STATE_A = 1              or
                LV_MIS_STATE_B = 1              or
                LV_ERR_SAV = 1                  or
                LV_ERR_SAP = 1                  or
                LV_ERR_FSD[i] = 1               or
                LV_ERR_LS_UP[i] = 1             or
                LV_ERR_CHG_LS_UP = 1            or
                LV_ERR_MAF = 1
        Then    LV_ACT_SA_EOL = 0
                STATE_EOL_KWP_SA = 1H           //conditions not correct
        Elself  LV_ERR_SA_SAV = 1               or
                LV_ERR_SA_SAP = 1
        Then    LV_ACT_SA_EOL = 0
                STATE_EOL_KWP_SA = 9H           //function finished, error present
        Elseif  LV_END_DIAG_SA_SAP =1           and
                LV_ERR_SA_SAV = 0               and
                LV_ERR_SA_SAP = 0
        Then    LV_ACT_SA_EOL = 0
                STATE_EOL_KWP_SA = 8H       //function finished, no error
        Endif
Else    no change
Endif
```

7.33.3.2.2 31_20 EOL Test start - SA-system test - StartRoutineByLocalIdentifier service

Formula section:

```
If      request 31_20 is received       and
        LV_IS = 1                       and
        TCO > C_TCO_MIN_SA_EOL          and
```

```
            LV_VB_JUMP = 0                     and
            LV_TEMP_MAX_SAP_COIL = 0           and
            LV_ERR_AMP = 0                     and
            LV_ERR_AMP_PLAUS = 0               and
            LV_ERR_TCO = 0                     and
            LV_ERR_CRK = 0                     and
            LV_ERR_FTL_MIN = 0                 and
            LV_MIS_STATE_A = 0                 and
            LV_MIS_STATE_B = 0                 and
            LV_ERR_SAV = 0                     and
            LV_ERR_SAP = 0                     and
            LV_ERR_FSD[i] = 0                  and
            LV_ERR_LS_UP[i] = 0                and
            LV_ERR_CHG_LS_UP = 0               and
            LV_ERR_MAF = 0                     and
            (STATE_DIAG_SA_LS = SA_DIAG_END or SA_DIAG_CNL)
Then        LV_ACT_SA_EOL = 1
            STATE_EOL_KWP_SA = ACT             //active
Else        STATE_EOL_KWP_SA = 1H              //conditions not correct
Endif
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= start EOL Test – SA-system test	M	#20h	RELI_
#3d	routineEntryOption = EOL Test active	M	#00h	REYO_01

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = EOL Test SA-system test	M	#20h	RELI_
#3d	routineEntryStatus = EOL Test active	M	#00h	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	startRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	responseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

7.33.3.2.3 32_20 EOL Test OFF - SA-system test - StopRoutineByLocalIdentifier service

Formula section:

```
If      request 32_20 is received
Then    LV_ACT_SA_EOL = 0
        STATE_EOL_KWP_SA = NOT_START     //function not started
Endif
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = stop EOL Test SA-system test	M	#20h	RELI_
#3d	routineEntryOption = EOL Test inactive	M	NOT_START	REYO_01

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = stop EOL Test SA-system test	M	#20h	RELI_
#3d	routineEntryStatus = EOL Test inactive	M	NOT_START	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	StopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table StopRoutineByLocalIdentifier Negative Response Message

7.33.3.2.4 33_20 - Report EOL results - SA-system test - RequestRoutineResultsByLocalIdentifier service

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = ask for results SA-system test	M	#20h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

Not for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = SA-system test	M	#20h	RELI_
#3d	RecordValue = STATE_DIAG_SA_LS	M	#XXh	RELI_
#4d	RecordValue = SUM_AFL_VLS_DIAG_SA_1 high byte	M	#XXh	RELI_
#5d	RecordValue = SUM_AFL_VLS_DIAG_SA_1 low byte	M	#XXh	RELI_
#6d	RecordValue = SUM_AFL_VLS_DIAG_SA_2 high byte	M	#XXh	RELI_
#7d	RecordValue = SUM_AFL_VLS_DIAG_SA_2 low byte	M	#XXh	RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

Only for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = SA-system test	M	#20h	RELI_
#3d	RecordValue = STATE_EOL_KWP_SA	M	#XXh	RELI_
#4d	RecordValue = STATE_DIAG_SA_LS	M	#XXh	RELI_
#5d	RecordValue = SUM_AFL_VLS_DIAG_SA_1 high byte	M	#XXh	RELI_
#6d	RecordValue = SUM_AFL_VLS_DIAG_SA_1 low byte	M	#XXh	RELI_
#7d	RecordValue = SUM_AFL_VLS_DIAG_SA_2 high byte	M	#XXh	RELI_
#8d	RecordValue = SUM_AFL_VLS_DIAG_SA_2 low byte	M	#XXh	RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

7.33.4 Services 21_22 / 31_22 / 32_22 / 33_22 for end of line test: CPS Check

7.33.4.1 31_22 – Start EOL-Test CPS-Check - StartRoutineByLocalIdentifier service

General information:

This funktion handles the functional check CPS also “end of line test CPS” called.

Application conditions:

Initialisation: LV_EOL_CPS =00h at reset
STATE_EOL_KWP_CPS= NOT_START at reset

Activation: LV_IGK=1 (Kl.15 on) **and**
LV_EOL_CPS = 1

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

Formula section:

calculation of LV_EOL_CPS
IF LV_IGK=1 (Kl.15 on) **and**
LV_ST_END = 1 **and**

```
                LV_INH_DIAGCPS = 0                      and
                VS = 0                                  and
                LV_MAF_SP_TQI_DYW_DIAGCPS = 1           and
                (OPM_AV_DIAGCPS = 1     or     OPM_AV_DIAGCPs = 2)          and
                LV_ERR_DIAGCPS  = 0
    THEN        LV_EOL_CPS = 1
    ELSE        LV_EOL_CPS = 0
    ENDIF
```

calculation of STATE_EOL_KWP_CPS

```
IF                          activation conditions are not fulfilled
    THEN                    STATE_EOL_KWP_CPS     =01H  // EOL_TEST_ST_INH

ELSE        IF              STATE_DIAGCPS         >= 01H        AND
                            LV_END_DIAG_DIAGCPS = 00H
            THEN            STATE_EOL_KWP_CPS     = ACT // EOL_TEST_ACT

            ELSEIF          STATE_DIAGCPS         >= 1H         AND
                            LV_END_DIAG_DIAGCPS = 1             AND
                            LV_EOL_CPS_ERR = 1
            THEN            STATE_EOL_KWP_CPS     =09H  // EOL_TEST_END_WITH_ERR

            ELSEIF          STATE_DIAGCPS         >= 1H         AND
                            LV_END_DIAG_DIAGCPS = 1             AND
                            LV_EOL_CPS_ERR = 0
            THEN            STATE_EOL_KWP_CPS     =08H  //EOL_TEST_END_WOUT_ERR

            ELSEIF          STATE_DIAGCPS         = 0H

            THEN            STATE_EOL_KWP_CPS     = NOT_START
            ENDIF
ENDIF
```

activation of EOL test

```
    IF              service 31_22 received and activation conditions for EOL test fullfilled
            THEN            STATE_EOL_KWP_CPS     =  ACT          // EOL_TEST_ACT
                            LV_EOL_CPS=01H                     //request to start EOL test

    ELSE                    STATE_EOL_KWP_DMTL remains unchanged

    ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= start EOL Test CPS	M	#22h	RELI_CPS

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = start EOL Test CPS	M	#22h	RELI_CPS
#3d	routineEntryStatus = STATE_EOL_KWP_CPS	M	ACT	REYS_

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	startRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	responseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

7.33.4.2 21_22 - Read STATUS EOL - CPS-Check - ReadDataByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier - EOL read state CPS Check	M	#22h	RELI_CPS

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier - EOL read state CPS Check	M	#22h	RELI_CPS
#3d	RecordValue = STATE_EOL_KWP_CPS	M	#XXh	RV_S_CPS

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	readDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	responseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

7.33.4.3 32_22 - Stop EOL-Test CPS-Check - StopRoutineByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

Formula section:

```
stop of EOL test
IF        service 32_22 received
THEN      STATE_EOL_KWP_CPS      = NOT_START // EOL_TEST_NOT_ST
                LV_EOL_CPS=00H                  //request to stop EOL test
ELSE      STATE_EOL_KWP_CPS remains unchanged
ENDIF
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**stopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	routineLocalIdentifier = stop EOL Test CPS	M	#22h	RELI_CPS

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	stopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	routineLocalIdentifier = stop EOL Test CPS	M	#22h	RELI_CPS
#3d	routineEntryStatus = STATE_EOL_KWP_CPS	M	NOT_START	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	stopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table StopRoutineByLocalIdentifier Negative Response Message

7.33.4.4 33_22 - Report results of - CPS EOL - RequestRoutineResultsByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = CPS EOL test values	M	#22h	RELI_CPS

Table RequestRoutineResultsByLocalIdentifier Request Message

Not for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = CPS EOL test values	M	#22h	RELI_CPS
#3d	RecordValue = STATE_DIAGCPS [Step_INIT Step_1 Step_2 Step_3 CPS_RAMP LOCK_STEP]	M	#XXh [#00h, #01h, #02h, #03h, #04h, #05h,]	RELI_
#4d	RecordValue = SUM_DIAG_DIAGCPS_EOL	M	#XXh	RELI_
#5d	RecordValue = SUM_FLOW_SP_DIAGCPS_EOL	M	#XXh	RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

Only for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = CPS EOL test values	M	#22h	RELI_CPS
#3d	RecordValue = STATE_EOL_KWP_CPS	M	#XXh	RELI_
#4d	RecordValue = STATE_DIAGCPS [Step_INIT Step_1 Step_2 Step_3 CPS_RAMP LOCK_STEP]	M	#XXh [#00h, #01h, #02h, #03h, #04h, #05h,]	RELI_
#5d	RecordValue = SUM_DIAG_DIAGCPS_EOL	M	#XXh	RELI_

#6d	RecordValue = SUM_FLOW_SP_DIAGCPS_EOL	M	#XXh	RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	responseCode= general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

7.33.5 Services 21_26 / 31_26 / 32_26 / 33_26 for end of line test: EOL- N_SP_IS displace

General information:

This EOL test is to calculate the N_SP_IS displace. The general function is described in the chapter “System variables - Idle speed setpoint calculation”.

Application conditions:

Initialisation at reset :

STATE_EOL_KWP_N_SP_IS = NOT_START

LV_ACT_N_SP_IS_EXT_ADJ = 0

N_SP_IS_EXT_ADJ = 0

Activation:	LV_IGK=1 (Kl.15 on)	and
	LV_IS=1 or LV_PL=1	and
	N<C_N_MAX_KWP	and
	GEAR_INFO=0h	and
	VS<= C_VS_MAX_KWP	and
	LV_CS=0	(for LV_AT=0 and LV_VAR_AMT=0 and LV_VAR_TCT = 0) and
	LV_BRAKE_DET=0	for LV_VAR_AMT=1
Deactivation:	LV_IGK=0 (Kl.15 off)	or
	Diagnostic timeout	
Service location:	ECU-SW	

7.33.5.1 31_26 – ON EOL- N_SP_IS displace - StartRoutineByLocalIdentifier service

Formula section:

IF activation conditions are fullfilled and
service 31_26 is received

THEN STATE_EOL_KWP_N_SP_IS = ACT (EOL test active)
LV_ACT_N_SP_IS_EXT_ADJ=1
CTL_SHIFT_LOCK_CAN=03h
send positive response

```
ELSE        STATE_EOL_KWP_N_SP_IS =01h (EOL test inhibited)
            LV_ACT_N_SP_IS_EXT_ADJ remains unchanged
            CTL_SHIFT_LOCK_CAN remains unchanged
            send negative response

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StartRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	RoutineLocalIdentifier= start EOL Test N_SP_IS displace	M	#26h	RELI_
#3d	RoutineEntryOption = N_SP_IS_EXT_ADJ high byte	M	#XXh	
#4d	RoutineEntryOption = N_SP_IS_EXT_ADJ low byte	M	#XXh	

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StartRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	RoutineLocalIdentifier = EOL Test N_SP_IS displace	M	#26h	RELI_
#3d	routineEntryStatus = EOL Test active	M	#00h	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	StartRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	ResponseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

7.33.5.2 32_26 - OFF EOL-Test N_SP_IS displace - StopRoutineByLocalIdentifier service

Formula section:

```
IF          service 32_26 received

THEN        STATE_EOL_KWP_N_SP_IS = NOT_START (EOL test not started)
            LV_ACT_N_SP_IS_EXT_ADJ =0

ELSE        STATE_EOL_KWP_N_SP_IS remains unchanged
            LV_ACT_N_SP_IS_EXT_ADJ remains unchanged

ENDIF
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = stop EOL Test N_SP_IS displace	M	#26h	RELI_

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = stop EOL Test N_SP_IS displace	M	#26h	RELI_
#3d	routineEntryStatus = EOL Test inactive	M	NOT_START	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	StopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	ResponseCode=general response codes	M	#XXh	RC

Table StopRoutineByLocalIdentifier Negative Response Message

7.33.5.3 21_26 - Read STATUS EOL– N_SP_IS displace - ReadDataByLocalIdentifier service

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	RecordLocalIdentifier = N_SP_IS displace	M	#26h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**ReadDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	RecordLocalIdentifier = N_SP_IS displace	M	#26h	RLI_
#3d	RecordValue = STATE_EOL_KWP_N_SP_IS [**MSV70**: function active, **MSX80**: function not started yet function start inhibited function start parameter not plausible **MSV70**: function not started yet, **MSx80**: function active]	M	#XXh [#00h, #01h, #02h, #05h]	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	ReadDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	ResponseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

7.33.5.4 33_26 - Report results of - N_SP_IS displace - RequestRoutineResultsByLocalIdentifier service

RequestRoutineResultsByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = N_SP_IS displace EOL test values	M	#26h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

Not for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = N_SP_IS displace EOL test values	M	#26h	RELI_
#3d	RecordValue = N high byte	M	#XXh	RELI_
#4d	RecordValue = N low byte	M	#XXh	RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

Only for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = N_SP_IS displace EOL test values	M	#26h	RELI_
#3d	RecordValue = STATE_EOL_KWP_N_SP_IS	M	#XXh	RELI_
#4d	RecordValue = N high byte	M	#XXh	RELI_
#5d	RecordValue = N low byte	M	#XXh	RELI_ RELI_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	ResponseCode= general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

7.33.6 Services 21_27 / 31_27 / 32_27 / 33_27 for end of line test: VVL adaption

7.33.6.1 31_27 - start VVL Adaptation - StartRoutineByLocalIdentifier service

General information:

This function handles the tester adaption request. After receiving the Tester request the function sets the variables KWP_B_FA, KWP_B_FAVVTAL and VVT_KWP_EXT_AD_ERV_REQ. Then it waits till the adaption process starts which can be detected testing the STATE_AD_RUN_VVL variable. After finishing the adaption process the function returns the result of the adaption routine
The general function is described in the chapter “VVL Position Sensor Adaptation”.

Application conditions:

Initialisation at reset state / at restart:
STATE_EOL_KWP_VVL_AD = NOT_START

Recurrence: 10ms - see Activation

Activation: started from the KW2000 request handler when the corresponding diagnose message is received

Deactivation: response sent to the tester

Service location: see list of “implemented diagnostic services”

Formula section:

```
if (STATE_KWP_SRV_AD_REQ_VVL!=“TERMINATE”)
then
    Send negative response with ResponseCode=22h;
    //ConditionNotCorrectOrRequestSequenceError
else
    if(2) ((REYS_01==01h) || ( REYS_01==06h))
    then(2)
        Send positive response
        call function “Set KWP adaption request state variable “ with parameter (REYS_01)
    else(2)
        Send negative response with ResponseCode=22h;
        //ConditionNotCorrectOrRequestSequenceError
    endif(2)
endif
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= start EOL Test VVL adaptation	M	#27h	RELI_VVL
#3d	REYS_01= Adaption request command	M	#XXh	REYS_01

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = EOL Test VVL adaptation	M	#27h	RELI_VVL
#3d	routineEntryStatus = EOL Test active	M	ACT	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**

#2d	startRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	responseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

7.33.6.2 21_27 - Read Status EOL: VVL adaption - ReadDataByLocalIdentifier service

General information:

This funktion reports the state of the EOL VVL adaption.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier STATE_EOL_KWP_VVL_AD	M	#27h	RLI_

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier STATE_EOL_KWP_VVL_AD	M	#27h	RLI_
#3d	RecordValue = STATE_EOL_KWP_VVL_AD	M	#XXh	RV_S_

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	readDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	responseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

7.33.6.3 32_27 - Stop EOL - VVL Adaptation - stopRoutineByLocalIdentifier

General information:

With this comand the EOL-Test VVL adaptation is stopped. The general function is described in the chapter “VVL Position Sensor Adaptation”.

Application conditions:

Initialisation: at reset

Activation: at receiving the corresponding KWP request

Deactivation:

Service location: see list of “implemented diagnostic services”

Formula section:

```
if      (request 32_27 received)
then
        Send positive response
        call function “Set KWP adaption request state variable “ with parameter ( “TERMINATE”)
endif
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**stopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	routineLocalIdentifier = stop EOL Test VVL adaptation	M	#27h	RELI_VVL

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	stopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	routineLocalIdentifier = stop EOL Test VVL adaptation	M	#27h	RELI_VVL
#3d	routineEntryStatus = EOL Test inactive	M	NOT_START	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	stopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table StopRoutineByLocalIdentifier Negative Response Message

7.33.6.4 33_27 - Read Status EOL: VVL adaptation - RequestRoutineResultsByLocalIdentifier service

General information:

This function reports the state of the EOL VVL adaption.

Application conditions:

Initialisation: at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see list of “implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = VVL adaptation EOL test values	M	#27h	RELI_

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = STATE_EOL_KWP_VVL_AD	M	#27h	RELI_
#4d	RecordValue = STATE_EOL_KWP_VVL_AD	M	#XXh	

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	ResponseCode= general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

7.33.7 Services 31_2A / 32_2A / 33_2A for EOL test: Generator

Application conditions:

Initialisation at reset: B_gentestanf = 0

Recurrence: **if** B_gentestanf =1 then 100ms
else event triggered

Activation: LV_IGK=1

Deactivation: LV_IGK=0

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

7.33.7.1 31_2A – start EOL Generatortest - StartRoutineByLocalIdentifier service

General information:

With this command the test of generatortest is started.

Formula section:

```
IF        service 31_2A is received

THEN      B_gentestanf = 1
          send positive response

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= Generatortest	M	#2Ah	RELI_GENUITEST

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response Id**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = Generatortest	M	#2Ah	RELI_GENUITEST
#3d	routineEntryStatus = EOL Generatortest activr	M	#05h	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.7.2 32_2A – stop EOL Generatortest - stopRoutineByLocalIdentifier

General information:

With this comand the test of generatortest is switched off.

Formular section:

```
/* event triggered:*/
If        service 32_2A is received

Then      B_gentestanf = 0
```

```
                    send positive response

Endif

/* with 100ms recurrence: */
If                  St_gentest = 6  /* test finished */
            or      St_gentest = 7  /* test interrupted */
            or      St_gentest = 8  /* test conditions not fullfilled */

Then                B_gentestanf = 0
Endif
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = Generatortest	M	#2Ah	RELI_

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = Generatortest	M	#2Ah	RELI_
#3d	routineEntryStatus = EOL Generatortest inactiv	M	#00h	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.7.3 33_2A – read EOL Generatortest results - requestRoutineResultsByLocalIdentifier

General information:

With this service the tester can ask for the results of generatortest.

Formular section:

```
If                  service 33_2A is received

Then                send positive response

Endif
```

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**

#2d	RoutineLocalIdentifier	M	#2Ah	RELI_GENUITEST

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier	M	#2Ah	RELI_GENUITEST
#3d	RecordValue#1 = St_gentest	M	#XXh	RRS_FS_GENUITEST
#4d	RecordValue#2 = Geniutest_err	M	#XXh	RRS_GENUITEST_ERR
#5d	RecordValue#3 = Geniutest_ab	M	#XXh	RRS_GENUITEST_AB
#6d	RecordValue#4 = Genitest_tol	M	#XXh	RRS_GENUITEST_TOL
#7d	RecordValue#5 = Genutest_tol	M	#XXh	RRS_GENUITEST_BOL
#8d	RecordValue#6 = I_gentest	M	#XXh	RRS_I_GENTEST
#9d	RecordValue#7 = U_gentest (high byte)	M	#XXh	RRS_U_GENTEST
#10d	RecordValue#8 = U_gentest (low byte)	M	#XXh	RRS_U_GENTEST

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.33.8 Services 31_2C / 32_2C / 33_2C for EOL test: ODR (Oil Pressure Regulation Diagnosis)

Application conditions:

Initialisation at reset: B_testpoelsys = 0
LV_POIL_EXT_ADJ_ACT = 0

Activation: LV_IGK=1

Deactivation: LV_IGK=0

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

7.33.8.1 31_2C – start EOL ODR test - StartRoutineByLocalIdentifier service

General information:

With this command the test of ODR (Oil Pressure Regulation) is started.

Formula section:

```
If        service 31_2C is received

Then      B_testpoelsys = 1
          LV_POIL_EXT_ADJ_ACT = 1
          send positive response

Endif
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= ODR	M	#2Ch	RELI_ODR

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response Id**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = Diagnosefunktion Oeldruckregelung	M	#2Ch	STRBLI_ODR_steuern
#3d	routineEntryStatus = EOL ODR test activ	M	#05h	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.8.2 32_2C – stop EOL ODR test – stopRoutineByLocalIdentifier service

General information:

With this command the test of ODR (Oil Pressure Regulation) is switched off.

Formula section:

```
If        service 32_2C is received

Then      B_testpoelsys = 0
          LV_POIL_EXT_ADJ_ACT = 0
          send positive response
```

Endif

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = ODR	M	#2Ch	RELI_ODR

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = Diagnosefunktion Oeldruckregelung beenden	M	#2Ch	SPRBLI_ODR_ende
#3d	routineEntryStatus = EOL ODR test inactiv	M	#00h	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.8.3 33_2C – Report EOL ODR results – requestRoutineResultsByLocalIdentifier service

General information:

With this service the tester can ask for the results of the Oil Pressure Regulation diagnosis.

Formula section:

If service 33_2C is received

Then send positive response

Endif

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = ODR	M	#2Ch	RELI_ODR

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier = RRRBLI_ODR_lesen	M	#2Ch	RRRBLI_ODR_lesen
#3d	RecordValue#1 = St_testpoelsys	M	#XXh	RRS_ST_TESTPOELSYS

#4d	RecordValue#2 = St_testpoelsys2	M	#XXh	RRS_ST_TESTPOELSYS2
#5d	RecordValue#3 = Toel high byte	M	#XXh	RRS_TOEL
#6d	RecordValue#4 = Toel low byte	M	#XXh	RRS_TOEL
#7d	RecordValue#5 = P_oel_soll high byte	M	#XXh	RRS_P_OEL_SOLL
#8d	RecordValue#6 = P_oel_soll low byte	M	#XXh	RRS_P_OEL_SOLL
#9d	RecordValue#7 = P_oel_ist high byte	M	#XXh	RRS_P_OEL_IST
#10d	RecordValue#8 = P_oel_ist low byte	M	#XXh	RRS_P_OEL_IST
#11d	RecordValue#9 = Nkw_soll high byte	M	#XXh	RRS_NKW_SOLL
#12d	RecordValue#10 = Nkw_soll low byte	M	#XXh	RRS_NKW_SOLL
#13d	RecordValue#11 = Nkw high byte	M	#XXh	RRS_NKW
#14d	RecordValue#12 = Nkw low byte	M	#XXh	RRS_NKW

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.33.9 Services 31_2D / 32_2D / 33_2D for EOL test: Steuern Desulfatisierung

<u>Application conditions:</u>

Initialisation at reset and at LV_IGK 1-> 0:

LV_NT_SO2P_EXT_ADJ_REQ_ST = 0
STATE_NT_SO2P_EXT_ADJ = NOT_START
LV_NT_SO2P_EXT_ADJ_REQ_NOT_STOP = 1

Activation: LV_IGK=1

Deactivation: LV_IGK=0

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

7.33.9.1 31_2D – start EOL Steuern Desulfatisierung- StartRoutineByLocalIdentifier service

<u>General information:</u>

With this command the desulfation is started.

<u>Formula section:</u>

If service 31_2D is received

Then LV_NT_SO2P_EXT_ADJ_REQ_NOT_STOP = 1
STATE_NT_SO2P_EXT_ADJ = received value
LV_NT_SO2P_EXT_ADJ_REQ_ST = 1 (only for 2 sec)
send positive response

Endif

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= Desulfation	M	#2Dh	RELI_DES
#3d	record value #1 = STATE_NT_SO2P_EXT_ADJ	M	#XXh	RV_STATE_NT_SO2P_EXT_ADJ

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response Id**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = Desulfation	M	#2Dh	STRBLI_DES
#3d	routineEntryStatus = Desulfation is activ	M	#05h	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.9.2 32_2D – stop EOL Steuern Desulfatisierung – stopRoutineByLocalIdentifier service

General information:

With this command the desulfation is switched off.

Formula section:

If service 32_2D is received

Then LV_NT_SO2P_EXT_ADJ_REQ_ST = 0
LV_NT_SO2P_EXT_ADJ_REQ_NOT_STOP = 0 (only for 2 sec)
send positive response

Endif

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = Desulfation	M	#2Dh	RELI_DES

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = Diagnosefunktion Oeldruckregelung beenden	M	#2Dh	SPRBLI_DES_ende
#3d	routineEntryStatus = Desulfation is inactiv	M	#00h	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.9.3 33_2D – Report EOL desulfation result – requestRoutineResultsByLocalIdentifier service

General information:

With this service the tester can ask for the results of the desulfation

Formula section:

If service 33_2D is received

Then send positive response

Endif

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = DES	M	#2Dh	RELI_DES

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier = RRRBLI_DES_lesen	M	#2Dh	RRRBLI_DES_lesen
#3d	RecordValue#1 = STATE_KWP_SO2P	M	#XXh	RRS_STATE_KWP_SO2P
#4d	RecordValue#2 = STATE_NT_SO2P_EXT_ADJ_ACT	M	#XXh	RRS_STATE_NT_SO2P_EXT_ADJ_ACT

#5d	RecordValue#3 = TNT_MDL_L (high byte)	M	#XXh	RRS_TNT_MDL_L
#6d	RecordValue#4 = TNT_MDL_L (low byte)	M	#XXh	RRS_TNT_MDL_L
#7d	RecordValue#5 = TNT_MDL_MV (high byte)	M	#XXh	RRS_TNT_MDL_MV
#8d	RecordValue#6 = TNT_MDL_MV (low byte)	M	#XXh	RRS_TNT_MDL_MV
#9d	RecordValue#7 = T_NT_SO2P_EXT_ADJ_ACT (high byte)	M	#XXh	RRS_ T_NT_SO2P_EXT_ADJ_ACT
#10d	RecordValue#8 = T_NT_SO2P_EXT_ADJ_ACT (low byte)	M	#XXh	RRS_ T_NT_SO2P_EXT_ADJ_ACT
#11d	RecordValue#9 = NT_SUL (high byte)	M	#XXh	RRS_NT_SUL
#12d	RecordValue#10 = NT_SUL (low byte)	M	#XXh	RRS_NT_SUL
#13d	RecordValue#11 = LAMB_SP (high byte)	M	#XXh	RRS_LAMB_SP
#14d	RecordValue#12 = LAMB_SP (low byte)	M	#XXh	RRS_LAMB_SP
#15	RecordValue#13 = LAMB_SP[1] (high byte)	M	#XXh	RRS_LAMB_SP[1]
#16d	RecordValue#14 = LAMB_SP[1] (low byte)	M	#XXh	RRS_LAMB_SP[1]
#17d	RecordValue#15 = LAMB_SP[2] (high byte)	M	#XXh	RRS_LAMB_SP[2]
#18d	RecordValue#16 = LAMB_SP[2] (low byte)	M	#XXh	RRS_LAMB_SP[2]
#19d	RecordValue#17 =N_32	M	#XXh	RRS_N_32
#20d	RecordValue#18 =VS	M	#XXh	RRS_VS
#21d	RecordValue#19 =PV (high byte)	M	#XXh	RRS_PV
#22d	RecordValue#20 =PV (low byte)	M	#XXh	RRS_PV
#23d	RecordValue#21 =GEAR	M	#XXh	RRS_GEAR
#24d	RecordValue#22 =LV_CLU_SWI	M	#XXh	RRS_LV_CLU_SWI

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.33.10 Services 31_33 / 32_33 / 33_33 for NOx-SensorShift Diagnose

Application conditions:

Initialisation at reset and at LV_IGK 0->1: Call ACTION_NOXD_EndNNShiftDiag()

Activation: LV_IGK=1

Deactivation: LV_IGK=0

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

7.33.10.1 31_33 – Start Werkstattdiagnose NOx-SensorShift - StartRoutineByLocalIdentifier service

General information:

With this command the "Werkstattdiagnose NOx-SensorShift" is requested

Formula section:

```
If(1)       service 31_33 is received
            Call ACTION_NOXD_StartNNShiftDiag()
            send positive response
Endif(1)
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= Auftrag NOx-SensorShiftDiagnose	M	#33h	STRBLI_NOXS

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response Id**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = Auftrag NOx-SensorShiftDiagnose	M	#33h	STRBLI_NOXS
#3d	routineEntryStatus = Status Auftrag NOx-SensorShiftDiagnose	M	#05h	STRBLI_ATL_steuern

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.10.2 32_33 – Stop Werkstatdiagnose NOx-SensorShift - stopRoutineByLocalIdentifier

General information:

With this comand the "Werkstattdiagnose NOx-SensorShift" is deactivated

Formular section:

```
If      service 32_33 is received
Then    Call ACTION_NOXD_EndNNShiftDiag()
        send positive response
Endif
```

StopRoutineByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = Auftrag Werkstatdiagnose NOx-SensorShift beenden	M	#33h	STRBLI_NOXS _beenden

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = Auftrag Werkstatdiagnose NOx-SensorShift beenden	M	#33h	STRBLI_NOXS_beenden
#3d	routineEntryStatus = Status Auftrag Werkstatdiagnose NOx-SensorShift is inactiv	M	#00h	STRBLI_NOXS_beenden

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.10.3 33_33 – Read Status Auftrag Diagnosefunktion NOx-SensorShift - requestRoutineResultsByLocalIdentifier service

General information:

With this service the tester can read out the state of "Werkstattdiagnose NOx-SensorShift".

Formular section:

If service 33_33 is received
Then send positive response
Endif

RequestRoutineResultsByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier – Status Werkstatdiagnose NOX-SensorShift lesen	M	#33h	RRRBLI_NOXS_lesen

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**

#2d	RoutineLocalIdentifier – Status Werkstatdiagnose NOX-SensorShift lesen	M	#33h	RRRBLI_NOXS _lesen
#3d	RecordValue#1 = STATE_EOL_KWP_NS_SHIFT_DIAG[1]	M	#XXh	RRS_
#4d	RecordValue#2 = STATE_NS_SHIFT_DIAG[1] (high byte	M	#XXh	RRS_
#5d	RecordValue#3 = STATE_NS_SHIFT_DIAG[1] (low byte)	M	#XXh	RRS_
#6d	RecordValue#4 = EGY_DEW_END_NT (high word, high byte)	M	#XXh	RRS_
#7d	RecordValue#5 = EGY_DEW_END_NT (high word, low byte)	M	#XXh	RRS_
#8d	RecordValue#6 = EGY_DEW_END_NT (low word, high byte)	M	#XXh	RRS_
#9d	RecordValue#7 = EGY_DEW_END_NT (low word, low byte)	M	#XXh	RRS_
#10d	RecordValue#8= EGY_DEW_END_NT_INT (high word, high byte)	M	#XXh	RRS_
#11d	RecordValue#9= EGY_DEW_END_NT_INT (high word, low byte)	M	#XXh	RRS_
#12d	RecordValue#10= EGY_DEW_END_NT_INT (low word, high byte)	M	#XXh	RRS_
#13d	RecordValue#11= EGY_DEW_END_NT_INT (low word, low byte)	M	#XXh	RRS_
#14d	RecordValue#12= TEG_WALL_NT_DOWN_MDL (high byte)	M	#XXh	RRS_
#15d	RecordValue#13 = TEG_WALL_NT_DOWN_MDL (low byte)	M	#XXh	RRS_
#16d	RecordValue#14 = T_NS_SHIFT_DEAC_TEMP[1] (high byte)	M	#XXh	RRS_
#17d	RecordValue#15 = T_NS_SHIFT_DEAC_TEMP[1] (low byte)	M	#XXh	RRS_
#18d	RecordValue#16 = T_NS_SHIFT_WAIT_REP[1] (high byte)	M	#XXh	RRS_
#19d	RecordValue#17 = T_NS_SHIFT_WAIT_REP[1] (low byte)	M	#XXh	RRS_
#20d	RecordValue#18 = LV_CAN_TEMP_MIN_THD[1]	M	#XXh	RRS_
#21d	RecordValue#19 = CTR_NS_SHIFT_CYC_NOT_VLD[1]	M	#XXh	RRS_
#22d	RecordValue#20 = CTR_NS_SHIFT_CYC_VLD[1]	M	#XXh	RRS_
#23d	RecordValue#21 = OPM_AV	M	#XXh	RRS_

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.33.11 Services 31_CF / 32_CF / 33_CF for : Sperre Tankentlueftung

Application conditions:

Initialisation at reset and at LV_IGK 0->1: LV_LAM_AD_ENA = 0

Activation: LV_IGK=1

Deactivation: LV_IGK=0

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

7.33.11.1 31_CF – Start Sperre Tankentlueftung - StartRoutineByLocalIdentifier service

General information:

With this command the "Sperre Tankentlueftung" is activated

Formula section:

```
IF        service 31_CF is received

THEN      LV_LAM_AD_EXT_ADJ = 1
          send positive response

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= Auftrag Deaktivierung TEV-Regelung starten	M	#CFh	STRBLI_TEVR-steuern

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response Id**	**M**	**#71h**	**STRBLIPR**
#2d	routineLocalIdentifier = Auftrag Deaktivierung TEV-Regelung starten	M	#CFh	STRBLI_TEVR-steuern
#3d	routineEntryStatus = Status Deaktivierung TEV-Regelung aktiv	M	#05h	STRBLI_TEVR-steuern

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.11.2 32_CF – Stop Sperre Tankentlueftung –StopRoutineByLocalIdentifier service

General information:

With this comand the "Sperre Tankentlueftung" is deactivated

Formular section:

```
If        service 32_CF is received

Then      LV_LAM_AD_EXT_ADJ = 0
          send positive response

Endif
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**

#2d	RoutineLocalIdentifier = Auftrag Deaktivierung TEV-Regelung beenden	M	#CFh	STRBLI_TEVR-steuern

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = Auftrag Deaktivierung TEV-Regelung beenden	M	#CFh	STRBLI_TEVR-steuern
#3d	routineEntryStatus = Status Deaktivierung TEV-Regelung inaktiv	M	#00h	STRBLI_TEVR-steuern

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.11.3 33_CF – read Status Sperre Tankentlueftung - RequestRoutineResultsByLocalIdentifier service

General information:

With this service the tester can ask for the status of "Sperre Tankentlueftung".

Formular section:

If service 33_CF is received

Then send positive response

Endif

RequestRoutineResultsByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = Auftrag Deaktivierung TEV-Regelung Status auslesen	M	#CFh	RRRBLI_TEVR_lesen

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier = Auftrag Deaktivierung TEV-Regelung Status auslesen	M	#CFh	RRRBLI_TEVR_lesen
#3d	RecordValue#1 = LV_LAM_AD_ENA	M	#XXh	RRS_LV_LAM_AD_ENA

#4d	RecordValue#2 = LV_LAM_AD_END	M	#XXh	RRS_LV_LAM_AD_END

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.33.12 Services 31_D0 / 32_D0 / 33_D0 for Werkstattdiagnose Abgas-Turbolader

Application conditions:

Initialisation at reset and at LV_IGK 0->1: LV_TCHA_DIAG_EXT_REQ = 0

Activation: LV_IGK=1

Deactivation: LV_IGK=0

Diagnostic timeout

Service location: see list of “implemented diagnostic services”

7.33.12.1 31_D0 – Start Werkstatdiagnose Abgas-Turbolader - StartRoutineByLocalIdentifier service

General information:

With this command the "Werkstatdiagnose Abgas-Turbolader" is requested

Formula section:

```
If      service 31_D0 is received

Then    LV_TCHA_DIAG_EXT_REQ = 1
        send positive response

Endif
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= Auftrag Werkstatdiagnose Abgas-Turbolader starten	M	#D0h	STRBLI_ATL_steuern

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Positive Response Id**	**M**	**#71h**	**STRBLIPR**

#2d	routineLocalIdentifier = Auftrag Werkstatdiagnose Abgas-Turbolader starten	M	#D0h	STRBLI_ATL_steuern
#3d	routineEntryStatus = Status Auftrag Werkstatdiagnose Abgas-Turbolader starten active	M	#05h	STRBLI_ATL_steuern

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.12.2 32_D0 – Stop Werkstatdiagnose Abgas-Turbolader - stopRoutineByLocalIdentifier

General information:

With this comand the "Sperre Tankentlueftung" is deactivated

Formular section:

```
If          service 32_D0 is received

Then        LV_TCHA_DIAG_EXT_REQ = 0
            send positive response

Endif
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier = Auftrag Werkstatdiagnose Abgas-Turbolader beenden	M	#D0h	STRBLI_ATL_beenden

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = Auftrag Werkstatdiagnose Abgas-Turbolader beenden	M	#D0h	STRBLI_ATL_beenden
#3d	routineEntryStatus = Status Auftrag Werkstatdiagnose Abgas-Turbolader inactive	M	#00h	STRBLI_ATL_beenden

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Global Negative Responses

7.33.12.3 33_D0 – read Status Auftrag Diagnosefunktion Abgasturbolader - requestRoutineResultsByLocalIdentifier service

General information:

With this service the tester can read out the state of "Werkstattdiagnose Abgasturbolader".

Formular section:

If service 33_D0 is received

Then send positive response

Endif

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier – Status Werkstatdiagnose Abgasturbolader lesen	M	#D0h	RRRBLI_ATL_lesen

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Positive Response**	**M**	**#73h**	**RRRBLIPR**
#2d	RoutineLocalIdentifier – Status Werkstatdiagnose Abgasturbolader lesen	M	#D0h	RRRBLI_ATL_lesen
#3d	RecordValue#1 = St_atlsvc	M	#XXh	RRS_ST_ATLSVC
#4d	RecordValue#2 = St_atlsvc_pvdk	M	#XXh	RRS_ST_ATLSVC_PVDK
#5d	RecordValue#3 = AMP (high byte)	M	#XXh	RRS_AMP
#6d	RecordValue#4 = AMP (low byte)	M	#XXh	RRS_AMP
#7d	RecordValue#5 = TAM	M	#XXh	RRS_TAM
#8d	RecordValue#6 = TIA	M	#XXh	RRS_TIA
#9d	RecordValue#7 = Atlsvc_dpvdk1 (high byte)	M	#XXh	RRS_ATLSVC_DPVDK1
#10d	RecordValue#8= Atlsvc_dpvdk1 (low byte)	M	#XXh	RRS_ATLSVC_DPVDK1
#11d	RecordValue#9= Atlsvc_dpvdk2 (high byte)	M	#XXh	RRS_ATLSVC_DPVDK2
#12d	RecordValue#10= Atlsvc_dpvdk2 (low byte)	M	#XXh	RRS_ATLSVC_DPVDK2
#13d	RecordValue#11= Atlsvc_dpvdk3 (high byte)	M	#XXh	RRS_ATLSVC_DPVDK3
#14d	RecordValue#12= Atlsvc_dpvdk3 (low byte)	M	#XXh	RRS_ATLSVC_DPVDK3
#15d	RecordValue#13 = Pwg_ist (high byte)	M	#XXh	RRS_PWG_IST
#16d	RecordValue#14 = Pwg_ist (low byte)	M	#XXh	RRS_PWG_IST
#17d	RecordValue#15 = Tn_abstell (high byte)	M	#XXh	RRS_TN_ABSTELL

#18d	RecordValue#16 = Tn_abstell (loe byte)	M	#XXh	RRS_TN_ABSTELL
#19d	RecordValue#17 = B_kupp	M	#XXh	RRS_B_KUPP
#20d	RecordValue#18 = Gangi	M	#XXh	RRS_GANGI
#21d	RecordValue#19 = V (high byte)	M	#XXh	RRS_V
#22d	RecordValue#20 = V (low byte)	M	#XXh	RRS_V
#23d	RecordValue#21 = Tmot (high byte)	M	#XXh	RRS_TMOT
#24d	RecordValue#22 = Tmot (low byte)	M	#XXh	RRS_TMOT
#25d	RecordValue#23 = Pu (high byte)	M	#XXh	RRS_PU
#26d	RecordValue#24 = Pu (low byte)	M	#XXh	RRS_PU
#27d	RecordValue#25 = LV_ERR_PUT_EL	M	#XXH	RRS_LV_ERR_PUT_EL

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Global Negative Responses

7.33.13 Services 21_DA / 31_DA / 32_DA / 33_DA for end of line test: Leakage detection

7.33.13.1 31_DA – ON EOL-Test DMTL - StartRoutineByLocalIdentifier service

General information:

This function shows the diagnosis status of the “end of line test DMTL”.

Application conditions:

Initialisation: STATE_EOL_KWP_DMTL = NOT_START (EOL_TEST_NOT_ST)) at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

Formula section:

calculation of STATE_EOL_KWP_DMTL

```
IF              LV_CONF_DMTL = 0

THEN            STATE_EOL_KWP_DMTL remains unchanged
                send negative response
ELSE
    IF(1)       conditions for EOL test are fulfilled

    THEN(1)     STATE_EOL_KWP_DMTL = ACT (EOL test active)
                send positive response

    ELSE(1)     STATE_EOL_KWP_DMTL = EOL_TEST_INH (EOL test inhibited=01H)
                send negative response
```

```
        ENDIF(1)
IF              service 32_DA is received
        THEN            STATE_EOL_KWP_DMTL = NOT_START
IF                      STATE_DMTL_EOL<=11h <=26H (DMTL cancelled)
        OR              DMTL cancelled due to other conditions(in this case
                        STATE_DMTL_EOL reports last active state before cancellation)
        THEN            STATE_EOL_KWP_DMTL = EOL_TEST_ABORTED (=07H)

IF                      STATE_DMTL=08h (DMTL function is finished, tank tight)
        THEN            STATE_EOL_KWP_DMTL = EOL_TEST_END_WOUT_ERR (=08H)

IF                      STATE_DMTL=09h (DMTL function is finished, small leak)
        OR              STATE_DMTL=0Ah (DMTL function is finished, rough leak)
        OR              STATE_DMTL=0Bh (DMTL function is finished, module error)
        THEN            STATE_EOL_KWP_DMTL = EOL_TEST_END_WITH_ERR (=09H)
ENDIF

activation of EOL test
IF              service 31_DA received and activation conditions for EOL test fullfilled
        THEN            STATE_EOL_KWP_DMTL   = ACT // EOL_TEST_ACT
        ELSE                    STATE_EOL_KWP_DMTL remains unchanged

ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier - start EOL Test DMTL	M	#DAh	RELI_DMTL

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = start EOL Test DMTL	M	#DAh	RELI_DMTL
#3d	routineEntryStatus = STATE_EOL_KWP_DMTL	M	ACT	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	startRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	responseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

7.33.13.2 21_DA - read STATUS DMTL EOL - ReadDataByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = DMTL EOL	M	#DAh	RLI_DMTL

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier - DMTL EOL	M	#DAh	RLI_KVA
#3d	RecordValue = STATE_EOL_KWP_DMTL	M	#XXh	RV_S_DMTL

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	readDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	responseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

7.33.13.3 32_DA - OFF EOL-Test DMTL - StopRoutineByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

Formula section:

stop of EOL test

```
    IF      service 32_DA received
    THEN    STATE_EOL_KWP_DMTL   = NOT_START// EOL_TEST_NOT_ST
ELSE        STATE_EOL_KWP_DMTL remains unchanged
ENDIF
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**stopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	routineLocalIdentifier = stop EOL Test DMTL	M	#Dah	RELI_DMTL

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	stopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	routineLocalIdentifier - stop EOL Test DMTL	M	#Dah	RELI_DMTL
#2d	routineEntryStatus = STATE_EOL_KWP_DMTL	M	NOT_START	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	stopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table StopRoutineByLocalIdentifier Negative Response Message

7.33.13.4 33_DA - report results of - DMTL EOL - RequestRoutineResultsByLocalIdentifier service

General information:

With this service the tester can ask for the results of DMTL end of line test which is started with the service 31_DA.

Remark: This service resets the EOL-state variable STATE_EOL_KWP_DMTL to NOT_START ("function not started yet") if EOL test has finished/cancelled before
(STATE_EOL_KWP_DMTL = 07h/08h/09h, "function finished or cancelled").
This is necessary because of the NVMY data storage of the DMTL-EOL test.

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier - DMTL EOL test values	M	#DAh	RELI_DMTL

Table RequestRoutineResultsByLocalIdentifier Request Message

Not for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = DMTL EOL test values	M	#DAh	RELI_DMTL
#3d	RecordValue = STATE_DMTL_EOL [START REF_LEAK_MES ROUGH_LEAK_MES ROUGH_LEAK_MES_LEN ROUGH_LEAK_MES_END SMALL_LEAK_MES SMALL_LEAK_MES_LEN 2.REF_LEAK_MES TANK_PROOFED SMALL_LEAK ROUGH_LEAK MODULE_ERROR ! LV_VB_RANGE_DMTL (VB out of range) ! LV_ENA_LEAK_DMTL (electrical error) LV_DET_REFU (fuel refill detected) LV_DET_FUC_OPEN (fuel cap open) LV_VB_DIF_MAX (VB fluctuation too high) LV_T_DMTL_MAX (maximum duration of diagnosis reached) LV_CUR_DMTL_REF_DIF_MAX (fluctuation of ref. current too high) LV_CUR_DMTL_THD_DIF_MES (decr. pump current during measure) END]	M	#XXh [#00h, #01h, #02h, #03h, #04h, #05h, #06h, #07h, #08h, #09h, #0Ah, #0Bh, #11h, #12h, #21h, #22h, #23h, #24h, #25h, #26h, #0Ch]	
#4d	RecordValue = CUR_DMTL_REF_LEAK_EOL high Byte	M	#XXh	
#5d	RecordValue = CUR_DMTL_REF_LEAK_EOL low Byte	M	#XXh	
#6d	RecordValue = CUR_DMTL_ROUGH_LEAK_MIN_EOL high Byte	M	#XXh	
#7d	RecordValue = CUR_DMTL_ROUGH_LEAK_MIN_EOL low Byte	M	#XXh	
#8d	RecordValue = CUR_DMTL_COR_FIL_EOL high Byte	M	#XXh	
#9d	RecordValue = CUR_DMTL_COR_FIL_EOL low Byte	M	#XXh	

Table RequestRoutineResultsLocalIdentifier Positive Response Message

Only for MSD80: RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = DMTL EOL test values	M	#DAh	RELI_DMTL
#3d	RecordValue = STATE_EOL_KWP_DMTL	M	#XXh	RELI_DMTL

#4d	RecordValue = STATE_DMTL_EOL [START REF_LEAK_MES ROUGH_LEAK_MES ROUGH_LEAK_MES_LEN ROUGH_LEAK_MES_END SMALL_LEAK_MES SMALL_LEAK_MES_LEN 2.REF_LEAK_MES TANK_PROOFED SMALL_LEAK ROUGH_LEAK MODULE_ERROR ! LV_VB_RANGE_DMTL (VB out of range) ! LV_ENA_LEAK_DMTL (electrical error) LV_DET_REFU (fuel refill detected) LV_DET_FUC_OPEN (fuel cap open) LV_VB_DIF_MAX (VB fluctuation too high) LV_T_DMTL_MAX (maximum duration of diagnosis reached) LV_CUR_DMTL_REF_DIF_MAX (fluctuation of ref. current too high) LV_CUR_DMTL_THD_DIF_MES (decr. pump current during measure) END]	M	#XXh [#00h, #01h, #02h, #03h, #04h, #05h, #06h, #07h, #08h, #09h, #0Ah, #0Bh, #11h, #12h, #21h, #22h, #23h, #24h, #25h, #26h, #0Ch]	
#5d	RecordValue = CUR_DMTL_REF_LEAK_EOL high Byte	M	#XXh	RELI_DMTL
#6d	RecordValue = CUR_DMTL_REF_LEAK_EOL low Byte	M	#XXh	RELI_DMTL
#7d	RecordValue = CUR_DMTL_ROUGH_LEAK_MIN_EOL high Byte	M	#XXh	RELI_DMTL
#8d	RecordValue = CUR_DMTL_ROUGH_LEAK_MIN_EOL low Byte	M	#XXh	RELI_DMTL
#9d	RecordValue = CUR_DMTL_COR_FIL_EOL high Byte	M	#XXh	RELI_DMTL
#10d	RecordValue = CUR_DMTL_COR_FIL_EOL low Byte	M	#XXh	RELI_DMTL

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

7.33.14 Services 21_DF / 31_DF / 32_DF / 33_DF for end of line test: Lambda sensors confused / wrong wired - linear sensors

7.33.14.1 31_DF EOL Test ON - Lambda sensors confused / wrong wired - StartRoutineByLocalIdentifier service

FUNCTION DESCRIPTION:

General information:

This EOL test is to detect confused or wrong wired linear lambda sensors in order to prevent bad emmissions.

i = 1 for cylinder bank1 and i = 2 for cylinder bank 2

Application conditions:

Initialisation: STATE_EOL_KWP_VLS = NOT_START at reset
LV_ACT_VLS_EOL_EXT_ADJ = 00h at reset

Activation: LV_IGK=1 (Kl.15 on) **and**
LV_IS = 1 **and**
TCO > C_TCO_MIN_VLS_EOL **and**
TEG_CAT_DOWN_MDL[i]
> C_TEG_CAT_DOWN_EOL **and** *//for both exhaust banks*
LV_LAMB_LS_UP_VLD[i] = 1 **and** *//for both exhaust banks*
LV_LS_DOWN_READY[i] = 1 **and** *//for both exhaust banks*
STATE_LSH_UP[i] = LSH_POW_CTL **and** *//for both exhaust banks*
STATE_LSH_DOWN[i] = LSH_POW_CTL **and** *//for both exhaust banks*
LV_DIAG_ACT_INH_LS_UP_DOWN[i] = 0 *//for both exhaust banks*

Deactivation: LV_IGK=0 (Kl.15 off) **or**
conditions from activation not fullfilled **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

Formula section:

activation of EOL test

```
IF      service 31_DF   is received                      and
        activation conditions for EOL test fullfilled (see above)
THEN    STATE_EOL_KWP_VLS      = ACT           // ACT
        LV_ACT_VLS_EOL_EXT_ADJ = 1             //request to start EOL test
ELSE    STATE_EOL_KWP_VLS remains unchanged
        LV_ACT_VLS_EOL_EXT_ADJ = 0
ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= start EOL Test shifted lambdasensor check	M	#DFh	RELI_LSEOL

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = EOL Test shifted lambdasensor check	M	#DFh	RELI_LSEOL
#3d	routineEntryStatus = STATE_EOL_KWP_VLS	M	ACT	REYS_01

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	startRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	responseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

7.33.14.2 21_DF Read Status EOL – Lambda sensors confused / wrong wired - ReadDataByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = shifted lambdasensor check	M	#DFh	RELI_LSEOL

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier = shifted lambdasensor check	M	#DFh	RELI_LSEOL
#3d	RecordValue = STATE_EOL_KWP_VLS	M	#XXh	RV_S_LSEOL

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	readDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	responseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

7.33.14.3 32_DF EOL Test OFF - Lambda sensors confused / wrong wired - StopRoutineByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

Formula section:

stop of EOL test

```
IF      service 32_DF received
THEN    STATE_EOL_KWP_VLS = NOT_START          // NOT_START
        LV_ACT_VLS_EOL_EXT_ADJ = 0             //request to stop EOL test
ELSE    STATE_EOL_KWP_VLS remains unchanged
ENDIF
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier - stop EOL Test shifted lambdasensor check	M	#DFh	RELI_LSEOL

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = stop EOL Test shifted lambdasensor check	M	#DFh	RELI_LSEOL
#3d	routineEntryStatus = STATE_EOL_KWP_VLS	M	NOT_START	REYS_01

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	StopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	responseCode=general response codes	M	#XX	RC

Table StopRoutineByLocalIdentifier Negative Response Message

7.33.14.4 33_DF - Report EOL results - Lambda sensors confused / wrong wired - RequestRoutineResultsByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

RequestRoutineResultsByLocalIdentifier Request

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = ask for shifted lambdasensor check EOL test values	M	#DFh	RELI_LSEOL

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	***Parameter Name***	***Cvt***	***Hex Value***	***Mnemonic***
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = shifted lambdasensor check EOL test values	M	#DFh	RELI_LSEOL
#3d	RecordValue#1 = STATE_EOL_KWP_VLS	M	#XXh	
#4d	RecordValue#2 = STATE_VLS_EOL [diagnosis inactive diagnosis 1.step diagnosis 2.step diagnosis finished, sensor OK diagnosis finished, upstream oxygen sensor shifted diagnosis finished, downstream oxygen sensor shifted diagnosis finished, upstream and downstream oxygen sensor shifted diagnosis finished, upstream oxygen sensor bank 1 not plausible diagnosis finished, upstream oxygen sensor bank 2 not plausible diagnosis finished, downstream oxygen sensor bank 1 not plausible diagnosis finished, downstream oxygen sensor bank 2 not plausible diagnosis finished, no interpretable result]	M	#XXh [#00h, #01h, #02h, #10h, #11h, #12h, #13h, #14h, #15h, #16h, #17h, #18h]	
#5d	RecordValue#3 = LAMB_LS_UP_AFR_EOL[1] high byte	M	#XXh	
#6d	RecordValue#4 = LAMB_LS_UP_AFR_EOL[1] low byte	M	#XXh	
#7d	RecordValue#5 = LAMB_LS_UP_AFR_EOL[2] high byte	M	#XXh	
#8d	RecordValue#6 = LAMB_LS_UP_AFR_EOL[2] low byte	M	#XXh	
#9d	RecordValue#7 = VLS_DOWN_AFR_EOL[1] high byte	M	#XXh	
#10d	RecordValue#8 = VLS_DOWN_AFR_EOL[1] low byte	M	#XXh	
#11d	RecordValue#9 = VLS_DOWN_AFR_EOL[2] high byte	M	#XXh	
#12d	RecordValue#10 = VLS_DOWN_AFR_EOL[2] low byte	M	#XXh	
#13d	RecordValue#11 = LAMB_LS_UP_AFL_EOL[1] high byte	M	#XXh	
#14d	RecordValue#12 = LAMB_LS_UP_AFL_EOL[1] low byte	M	#XXh	
#15d	RecordValue#13 = LAMB_LS_UP_AFL_EOL[2] high byte	M	#XXh	
#16d	RecordValue#14 = LAMB_LS_UP_AFL_EOL[2] low byte	M	#XXh	

#17d	RecordValue#15 = VLS_DOWN_AFL_EOL[1] high byte	M	#XXh	
#18d	RecordValue#16 = VLS_DOWN_AFL_EOL[1] low byte	M	#XXh	
#19d	RecordValue#17 = VLS_DOWN_AFL_EOL[2] high byte	M	#XXh	
#20d	RecordValue#18 = VLS_DOWN_AFL_EOL[2] low byte	M	#XXh	
#21d	RecordValue#19 = MAF_INT_MIN_VLS_EOL high Byte	M	#XXh	
#22d	RecordValue#20 = MAF_INT_MIN_VLS_EOL low Byte	M	#XXh	

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

7.33.15 Services 21_D5 / 31_D5 / 32_D5 / 33_D5 for end of line test: ECRAS system test

FUNCTION DESCRIPTION:

General information:

This EOL test is to test the ECRAS System. The ECRAS diagnosis will be enabled.

Application conditions:

Initialisation: at reset or LV_IGK 0->1: STATE_EOL_KWP_ECRAS = NOT_START,
LV_ACT_ECRAS_EOL_EXT_ADJ = 00h

Recurrence: same as ECRAS diagnosis function (100ms)

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) **or**
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

Formula section:

Calculation of STATE_EOL_KWP_ECRAS:

The calculation of the state STATE_EOL_KWP_ECRAS is started with activation of LV_ACT_ECRAS_EOL_EXT_ADJ (0->1). Depending on several conditions the state and LV_ACT_ECRAS_EOL_EXT_ADJ will be set. The Determination of STATE_EOL_KWP_ECRAS is stopped after deactivation of the EOL by tester. In this case STATE_EOL_KWP_ECRAS = NOT_START (function not started). Once the calculation of the STATE_EOL_KWP_ECRAS is started, it has to be calculated with a recurrence of 100ms until deactivation.

If LV_ACT_ECRAS_EOL_EXT_ADJ = 1

Then **If** LV_ECRAS_UP_EXT_ADJ = 1 **or**

```
                LV_ECRAS_DOWN_EXT_ADJ = 1  or
                LV_CDN_VB_MIN_DIAG = 0
         Then   LV_ACT_ECRAS_EOL_EXT_ADJ = 0
                STATE_EOL_KWP_ECRAS = 01H  //conditions not correct
         Elself LV_ECRAS_EOL_INH = 1       and
                LV_ERR_ECRAS_EOL = 1
         Then   LV_ACT_ECRAS_EOL_EXT_ADJ = 0
                STATE_EOL_KWP_ECRAS = 9H   //function finished, error present
         Elseif LV_ECRAS_EOL_INH = 1
         Then   LV_ACT_ECRAS_EOL_EXT_ADJ = 0
                STATE_EOL_KWP_ECRAS = 8H   //function finished, without error
         Endif
Else     no change
Endif
```

7.33.15.1 31_D5 EOL Test ON - ECRAS system test - StartRoutineByLocalIdentifier service

Description:

If the activation service is received and the activation conditions are fulfilled the ECRAS system test is started (LV_ACT_ECRAS_EOL_EXT_ADJ 0-> 1)

Formula section:

activation of EOL test

```
IF       service 31_D5   is received         and
         LV_ECRAS_UP_EXT_ADJ = 0             and
         LV_ECRAS_DOWN_EXT_ADJ = 0           and
         LV_CDN_VB_MIN_DIAG = 1              and
         LV_ECRAS_EOL_INH = 0
THEN     STATE_EOL_KWP_ECRAS = ACT           // ACT
         LV_ACT_ECRAS_EOL_EXT_ADJ = 1        //request to start EOL test
ELSE     STATE_EOL_KWP_ECRAS = 1H            //conditions not correct
         LV_ACT_ECRAS_EOL_EXT_ADJ remains unchanged
ENDIF
```

StartRoutineByLocalIdentifier Request

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1d	**startRoutineByLocalIdentifier Request Service Id**	**M**	**#31h**	**STRBLI**
#2d	routineLocalIdentifier= start EOL ECRAS system test	M	#D5h	RELI_GLFS

Table StartRoutineByLocalIdentifier Request Message

StartRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	startRoutineByLocalIdentifier Positive Response SId	M	#71h	STRBLIPR
#2d	routineLocalIdentifier = EOL ECRAS system test	M	#D5h	RELI_GLFS
#3d	routineEntryStatus = STATE_EOL_KWP_ECRAS	M	ACT	RRR_FS_GLFS

Table StartRoutineByLocalIdentifier Positive Response Message

StartRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	startRoutineByLocalIdentifier Request Service Id	M	#31h	STRBLI
#3d	responseCode=only general response code	M	#XXh	RC

Table StartRoutineByLocalIdentifier Negative Response Message

7.33.15.2 21_D5 Read Status EOL – ECRAS system test - ReadDataByLocalIdentifier service

Application conditions:

Initialisation: 0 at reset

Activation: LV_IGK=1 (Kl.15 on)

Deactivation: LV_IGK=0 (Kl.15 off) or
Diagnostic timeout

Service location: see “list of implemented diagnostic services”

ReadDataByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Request Service Id**	**M**	**#21h**	**RDBLI**
#2d	recordLocalIdentifier = ECRAS system test	M	#D5h	RELI_LSEOL

Table readDataByLocalIdentifier Request Message

ReadDataByLocalIdentifier positive Response on service

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**readDataByLocalIdentifier Positive Response Service Id**	**M**	**#61h**	**RDBLIDPR**
#2d	recordLocalIdentifier = ECRAS system test	M	#D5h	RELI_LSEOL
#3d	RecordValue = STATE_EOL_KWP_ECRAS	M	#XXh	RV_S_LSEOL

Table readDataByLocalIdentifier Positive Response Message

ReadDataByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**

#2d	readDataByLocalIdentifier Request Service Id	M	#21h	RDBLID
#3d	responseCode=only general response code	M	#XXh	RC

Table: readDataByLocalIdentifier Negative Response Message

7.33.15.3 32_D5 EOL Test OFF – ECRAS system test - StopRoutineByLocalIdentifier service

Description:

If this service is received the EOL test will be stopped.

Formula section:

stop of EOL test

```
IF      service 32_D5 received
THEN    STATE_EOL_KWP_ECRAS = NOT_START       // NOT_START
        LV_ACT_ECRAS_EOL_EXT_ADJ = 0          //request to stop EOL test
ELSE    STATE_EOL_KWP_ECRAS remains unchanged
ENDIF
```

StopRoutineByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**StopRoutineByLocalIdentifier Request Service Id**	**M**	**#32h**	**SPRBLI**
#2d	RoutineLocalIdentifier - stop EOL ECRAS system test	M	#D5h	RELI_GLFS

Table StopRoutineByLocalIdentifier Request Message

StopRoutineByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	StopRoutineByLocalIdentifier Positive Response	M	#72h	SPRBLIPR
#2d	RoutineLocalIdentifier = stop EOL ECRAS system test	M	#D5h	RELI_GLFS
#3d	routineEntryStatus = STATE_EOL_KWP_ECRAS	M	NOT_START	RRR_FS_ GLFS

Table StopRoutineByLocalIdentifier Positive Response Message

StopRoutineByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**negativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	StopRoutineByLocalIdentifier Request Service Id	M	#32h	SPRBLI
#3d	responseCode=general response codes	M	#XX	RC

Table StopRoutineByLocalIdentifier Negative Response Message

7.33.15.4 33_D5 - Report EOL results - ECRAS system test - RequestRoutineResultsByLocalIdentifier service

Description:

The EOL state (STATE_EOL_KWP_ECRAS) and the status of the ECRAS system (STATE_ECRAS_SYS) will send if this service is received.

RequestRoutineResultsByLocalIdentifier Request

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**RequestRoutineResultsByLocalIdentifier Request Service Id**	**M**	**#33h**	**RRRBLI**
#2d	RoutineLocalIdentifier = ask EOL test state and values	M	#D5h	RELI_GLFS

Table RequestRoutineResultsByLocalIdentifier Request Message

RequestRoutineResultsByLocalIdentifier positive Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	RequestRoutineResultsByLocalIdentifier Positive Response	M	#73h	RRRBLIPR
#2d	RoutineLocalIdentifier = ECRAS system test EOL test values	M	#D5h	RELI_GLFS
#3d	RecordValue = STATE_EOL_KWP_ECRAS	M	#XXh	RRR_FS_GLFS
#4d	RecordValue = STATE_ECRAS_SYS high byte	M	#XXh	
#5d	RecordValue = STATE_ECRAS_SYS low byte	M	#XXh	

Table RequestRoutineResultsLocalIdentifier Positive Response Message

RequestRoutineResultsByLocalIdentifier negative Response

Data Byte	*Parameter Name*	*Cvt*	*Hex Value*	*Mnemonic*
#1d	**NegativeResponse Service Id**	**M**	**#7Fh**	**NR**
#2d	RequestRoutineResultsByLocalIdentifier Request Service Id	M	#33h	RRRBLI
#3d	responseCode=general response codes	M	#XXh	RC

Table RequestRoutineResultsByLocalIdentifier Negative Response Message

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_MIN_SA_EOL	1	0...FEH	-48...142.5	0.75	[°C]
TCO Threshold for EOL activation					

8 Driver request pedals determination

Table of Contents

Index

8.1 AGGR DRRQ adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PV	-	0...3FFH	0...99.90234	9.77E-02	[%]
Global degree of activation of the accelerator pedal (high resolution)					

Input data:

PV_AV			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: 0 % at reset

Recurrence : 10 ms, as input data (sequencing: input before output)

Activation: every engine state

Formula section:

PV = PV_AV

8.2 Determination of accelerator pedal value

General information:

The accelerator pedal value is determined by means of a two-channeled system (PVS: pedal value sensor). Upon operating the accelerator pedal, both independently supplied sensor channels supply a synchronized Voltage signal, which shows a definite Voltage-ratio (2:1). Both Voltage-signals are adapted to the supply Voltage of the potis in order to minimize measurement inaccuracy.

Plausibility unit monitors both output signals of the pedal sensor system as well as the supply Voltages and determine from this the state of the PVS's system (see chapter PVS diagnosis).

From the Voltages of both PVS-channels the Kick-Down operation is determined.

Dependent on the System State, the driver's demand is determined from the pedal activation values. In the case of a disturbed PVS-signal (ERR_PVS ≠ 0), a selection of pedal values occurs from one of both PVS-channels and/or an output of zero.

Figure 1 shows a function overview of determination of the driver's demand.

Signal flow diagram:

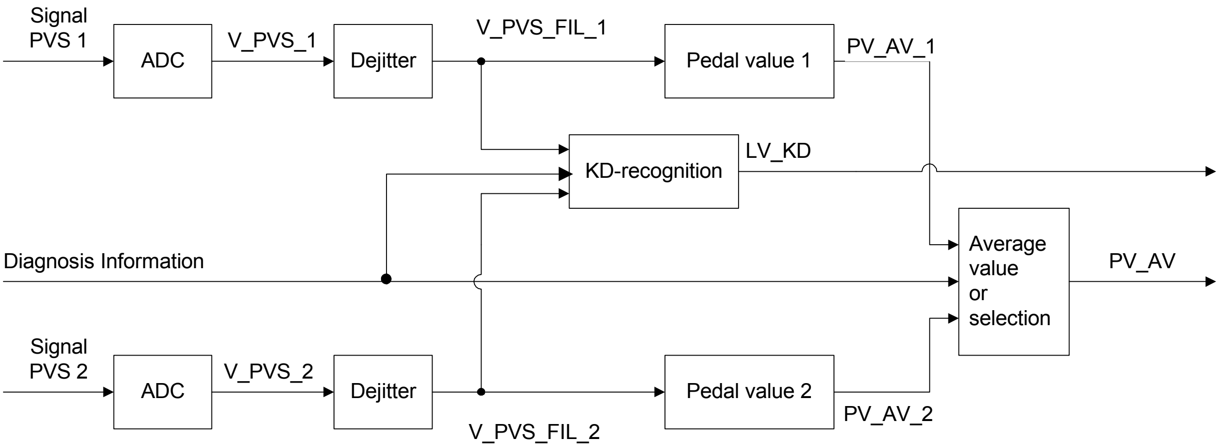


Fig. 1: Function overview of determination of the driver's demand

Comment:
Since we are dealing with a two-channel system, the letter 'X' is used for either channel 1 or 2, in order to simplify the description. The letter 'Y' indicates the other respective channel.

Application conditions:

Recurrence: 10 ms

8.2.1 Filtration of the PVS-values

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
V_PVS_FIL_1	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Filtered voltage of Pedal Value Sensor 1					
V_PVS_FIL_2	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Filtered voltage of Pedal Value Sensor 2					

Input data:

V_PVS_1	V_PVS_2		

FUNCTION DESCRIPTION:

In order to maintain a stabile driver’s demand, the PVS-Voltage values need to be filtered.

During the filtration, a hysteresis band is laid around the determined PVS-value (Width: 2 * C_V_PVS_HYS_FIL). The PVS-value is only changed when the current measurement value exceeds the limits of the hysteresis band. Thereby a jitter-free signal is produced, which actually hurries after the ‘true’ measurement value around the width of the hysteresis band. However, this signal reacts without hesitation to dynamic alterations of the driver’s demand.

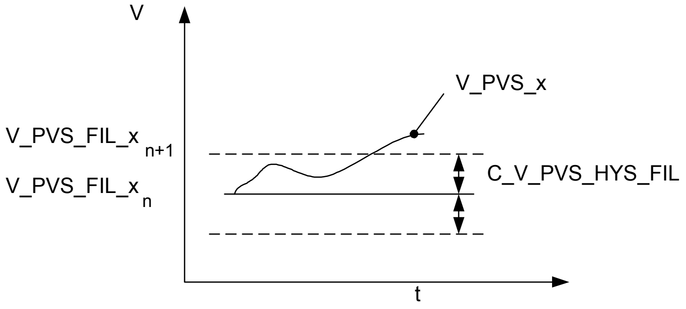


Fig. 2 Filtration of PVS Voltage

Formula section:

If $V\_PVS\_x_{(n)} > V\_PVS\_FIL\_x_{(n-1)} + C\_V\_PVS\_HYS\_FIL$ **or**
$V\_PVS\_x_{(n)} < V\_PVS\_FIL\_x_{(n-1)} - C\_V\_PVS\_HYS\_FIL$

Then $V\_PVS\_FIL\_X_{(n)} = V\_PVS\_x_{(n)}$

Else $V\_PVS\_FIL\_X_{(n)} = V\_PVS\_FIL\_x_{(n-1)}$

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_PVS_HYS_FIL	1	0...FFH	0...1.24511	4.8828e-3	[V]
Area of the dejitter-filters					

8.2.2 Standardisation of the PVS-voltage value

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PV_AV_1	O/V	0...FFH	0...99.60937	0.390625	[%]
degree of activation of the accelerator pedal from PVS-channel 1					
PV_AV_2	O/V	0...FFH	0...99.60937	0.390625	[%]
degree of activation of the accelerator pedal from PVS-channel 2					
PV_CUS_1	O/V	0...FFFFH	0...99.99847	1.5259e-3	[%]
degree of activation of the accelerator pedal from PVS-channel 1 with higher resolution for customer purposes					
PV_CUS_2	O/V	0...FFFFH	0...99.99847	1.5259e-3	[%]
degree of activation of the accelerator pedal from PVS-channel 2 with higher resolution for customer purposes					

FUNCTION DESCRIPTION:

The pedal values are normalized by the determined producer values for the potentiometer slope C_PVS_SLOP and offset C_PVS_OFS in a range between 0% and 99,6%.

Formula section:

PV_CUS_x = PV_AV_x = (V_PVS_FIL_x - C_V_PVS_OFS_x) * C_PVS_SLOP_x

Calibration hint: in order to prevent the signal PVS 2 with it's worse resolution is selected for PV_AV_RAW (see following Value selection), the offset and slope for that sensor should be calibrated accordingly. PV_AV_2 always should be a little higher than PV_AV_1. The difference should correspond to the usual occurring signal jitter which depends from the used sensor

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_PVS_SLOP_1	1	0...FFH	0...79.6875	0.3125	[%/V]
Slope value of the pedal value 1					
C_PVS_SLOP_2	1	0...FFH	0...79.6875	0.3125	[%/V]
Slope value of the pedal value 2					
C_V_PVS_OFS_1	1	0...FFH	0...1.24511	4.8828e-3	[V]
Offset of the pedal value 1					
C_V_PVS_OFS_2	1	0...FFH	0...1.24511	4.8828e-3	[V]
Offset of the pedal value 2					

8.2.3 Value selection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PV_AV_RAW	O/V	0...FFH	0...99.60937	0.390625	[%]
Global degree of activation of the gas pedal					
PV_CUS_RAW	O/V	0...FFFFH	0...99.99847	1.5259e-3	[%]
Global degree of activation of the gas pedal with higher resolution for customer purpose					

Input data:

LV_DET_NOT_PLAUS_ACT	STATE_PVS_DIAG	N_32	LV_IGK
LV_ERR_PVS_BLS_NOT_PLAUS	LV_PVS_BLS_NOT_PLAUS_ACT		

FUNCTION DESCRIPTION:

For an error-free system, the pedal value PV_AV_RAW (PV_CUS_RAW) is determined from minimum of both pedal values PV_AV_x (PV_CUS_x). In case of an error the selection depends on the PVS Diagnostic status. The restriction of the accelerator pedal value takes place at the limp home management.
In case of a not plausible accelerator-brake plausibility the driver demand is limited to C_PVS_BLS_NOT_PLAUS to defuse a clamp driver pedal.

Application conditions:

Initialisation: 0

Recurrence: 10ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0, set PV_AV_RAW=0, PV_CUS_RAW=0

Formula section:

```
If(1)        STATE_PVS_DIAG = PVS_NO_ERROR
Then(1)      PV_AV_MIN  =  MIN  ( PV_AV_1; PV_AV_2 )
             PV_CUS_MIN  =  MIN  ( PV_CUS_1; PV_CUS_2 )
    If(2)           LV_PVS_BLS_NOT_PLAUS_ACT = 0       and
                    LV_ERR_PVS_BLS_NOT_PLAUS = 0
    Then(2)         PV_AV_RAW = PV_AV_MIN
                    PV_CUS_RAW = PV_CUS_MIN
    Else(2)If(3)    LV_DET_NOT_PLAUS_ACT =1
             Then(3) If(4)      PV_AV_RAW >  C_PV_AV_BLS_NOT_PLAUS
                 Then(4)        $PV\_AV\_RAW_n$  = MIN (PV_AV_MIN; $PV\_AV\_RAW_{n-1}$ -
                                        C_PV_AV_BLS_NOT_PLAUS_LGRD)
                                $PV\_CUS\_RAW_n$  = MIN (PV_CUS_MIN; $PV\_CUS\_RAW_{n-1}$
-
                                        C_PV_AV_BLS_NOT_PLAUS_LGRD)
                 Else(4)  If(5)     PV_AV_RAW $\geq$ PV_AV_MIN
```

```
                    Then(5)   $PV\_AV\_RAW = PV\_AV\_MIN$
                              $PV\_CUS\_RAW = PV\_CUS\_MIN$
                    Else(5)   $PV\_AV\_RAW_n = PV\_AV\_RAW_{n-1}$
                              $PV\_CUS\_RAW_n = PV\_CUS\_RAW_{n-1}$
        Else(3)If(6)  $|PV\_AV\_RAW_{n-1} - PV\_AV\_MIN| > C\_PV\_AV\_BLS\_NOT\_PLAUS\_HYS$
            Then(6) If(7)     $PV\_AV\_RAW_{n-1} < PV\_AV\_MIN$
                    Then(7)   $PV\_AV\_RAW_n = PV\_AV\_RAW_{n-1} + C\_PV\_AV\_BLS\_NOT\_PLAUS\_LGRD$
                              $PV\_CUS\_RAW_n = PV\_CUS\_RAW_{n-1} + C\_PV\_AV\_BLS\_NOT\_PLAUS\_LGRD$
                    Else(7)   $PV\_AV\_RAW_n = PV\_AV\_MIN$
                              $PV\_CUS\_RAW_n = PV\_CUS\_MIN$
            Else(6) $PV\_AV\_RAW_n = PV\_AV\_MIN$
                    $PV\_CUS\_RAW_n = PV\_CUS\_MIN$
Endif
```

In case of a present PVS error witch is not debounced yet, the pedal value is frozen in risen direction:

```
If      $STATE\_PVS\_DIAG = PVS\_LIH\_PRED$ and
        $PV\_AV\_1 \geq PV\_AV\_RAW$
Then    $PV\_AV\_RAW_n = PV\_AV\_RAW_{n-1}$
        $PV\_CUS\_RAW_n = PV\_CUS\_RAW_{n-1}$
Else    $PV\_AV\_RAW_n = PV\_AV\_1$
        $PV\_CUS\_RAW_n = PV\_CUS\_1$
Endif

If      $STATE\_PVS\_DIAG = PVS\_LIH\_MIN$ or $PVS\_LIH\_1$ or $PVS\_LIH\_2$
Then    PV_AV_RAW and PV_CUS_RAW are limited upwards to IP_PV_LIH_MAX and
        the positive gradient is restricted to
        $PV\_AV\_RAW_n \leq PV\_AV\_RAW_{n-1} + C\_PV\_GRD\_LIH\_MAX$
        $PV\_CUS\_RAW_n \leq PV\_CUS\_RAW_{n-1} + C\_PV\_GRD\_LIH\_MAX$
Endif

If      $STATE\_PVS\_DIAG = PVS\_LIH\_MIN$
Then    $PV\_AV\_RAW = Min(PV\_AV\_1, PV\_AV\_2)$
        $PV\_CUS\_RAW = Min(PV\_CUS\_1, PV\_CUS\_2)$
Endif
If      $STATE\_PVS\_DIAG = PVS\_LIH\_1$
Then    $PV\_AV\_RAW = PV\_AV\_1$
        $PV\_CUS\_RAW = PV\_CUS\_1$
Endif
If      $STATE\_PVS\_DIAG = PVS\_LIH\_2$
```

```
Then    PV_AV_RAW = PV_AV_2
        PV_CUS_RAW = PV_CUS_2
Endif

If      STATE_PVS_DIAG = PVS_DBL_ERROR  or
        STATE_PVS_DIAG = PVS_LIH_1_PRE  or
        STATE_PVS_DIAG = PVS_LIH_2_PRE  or
        STATE_PVS_DIAG = PVS_LIH_BLS
Then    PV_AV_RAW = 0
        PV_CUS_RAW = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_PV_GRD_LIH_MAX	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Max. (positive) pedal gradient in case of an error					
C_PV_AV_BLS_NOT_PLAUS	1	0...FFH	0...99.60937	0.390625	[%]
pedal position in case of an PV_AV/Brake implausibility					
C_PV_AV_BLS_NOT_PLAUS_LGRD	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Change limitation of pedal position in case of an PV_AV/Brake implausibility					
C_PV_AV_BLS_NOT_PLAUS_HYS	1	0...FFH	0...99.60937	0.390625	[%]
Hysteresis in case of an PV_AV/Brake implausibility					
IP_PV_LIH_MAX	6	0...FFH	0...99.60937	0.390625	[%]
LDP_N_32__PV_LIH_MAX	6	0...FFH	0...8160	32	[rpm]
Max. pedal position in case of an error					

8.2.4 Driver progression correction

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PV_AV	V/O	0...FFH	0...99.60937	0.390625	[%]
Corrected pedal value interpretation					
FAC_PV_CS_COR	V	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
Starting correction					
FAC_PV_GS_COR	V	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
Gear shift correction					
FAC_PV_BAS_COR	V	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
basic correction					
FAC_PV_COR	V	0...7FFFH	0...0.99996	0.0305e-3	[-]
Summation of corrections					
PV_CUS	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Corrected pedal value interpretation with higher resolution for customer purpose					

Input data:

GEAR_INFO	LV_AT		STATE_GEAR_REV_AT_AMT
VS_FIL	LV_IM_CS_PN		STATE_GEAR_REV_CAN
N_DIF	N	PV_AV_MAX_BRAKE_DET	
LV_RNG_L_REQ	LV_VAR_TCT	STATE_ETCU_CLU	

FUNCTION DESCRIPTION:

PV_AV and PV_CUS are the corrected driver interpretation and are used in the torque structure, CAN outputs and other functions.
The correction is a summation out of three parts, the basic, starting and gear shift correction.

Application conditions:

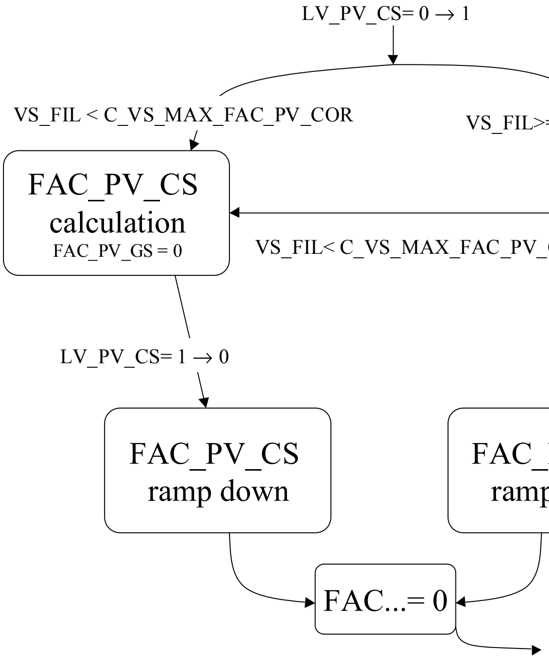
Initialisation: 0

Recurrence: 10ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0, set PV_AV=0, PV_CUS=0

Signal flow diagram:



Formula section:

```
If(1) LV_AT = 1
        Then(1)
            If(2) GEAR_INFO = 1 and VS_FIL < C_VS_MAX_FAC_PV_COR_AT
            Then(2)         LV_PV_CS = 1
            Else(2)         LV_PV_CS = 0
            Endif(2)
        Endif(1)
Elseif(1) LV_VAR_TCT = 1
        Then(3)
            If(4) STATE_ETCU_CLU = 03H (CLU_CLOSED)
                OR 04H (CLU_SLIP_CTRL) OR 05H (CLU_DYN_SLIP)
            Then(4)         LV_PV_CS = 0
            Else(4)         LV_PV_CS = 1
            Endif(4)
        Endif(3)
Endif(1)
```

```
If      LV_AT = 0   and   LV_VAR_TCT = 0
```

```
then    LV_PV_CS = LV_IM_CS_PN
Endif
```

Basic correction:

PV_AV = MIN ((PV_AV_RAW * (1 - FAC_PV_COR)); PV_AV_MAX_BRAKE_DET)

PV_CUS = MIN ((PV_CUS_RAW * (1 - FAC_PV_COR)); PV_AV_MAX_BRAKE_DET)

FAC_PV_COR = min(1 ; max(0 ; (FAC_PV_CS_COR + FAC_PV_GS_COR + FAC_PV_BAS_COR)))

FAC_PV_COR has to be limited in the range between zero and one, so that no amplification of the raw value exists!

Progression correction at drive off FAC_PV_CS_COR:

Calculation by getting active Function LV_PV_CS = 0 to 1:

```
If(1)      VS_FIL < C_VS_MAX_FAC_PV_COR
Then(1)    If(2)      LV_VAR_TCT = 1
           Then(2)    FAC_PV_CS_COR = IP_FAC_PV_CS_COR_TCT
           Else(2)    If(3)      LV_AT = 0
                      Then(3)    FAC_PV_CS_COR = IP_FAC_PV_CS_COR_MT
                      Else(3)    If(4)      LV_RNG_L_REQ = 1
                                 Then(4)    FAC_PV_CS_COR =
                                            IP_FAC_PV_CS_COR_AT_RNG_L
                                 Else(4)    FAC_PV_CS_COR =
                                            IP_FAC_PV_CS_COR_AT
                                            Endif(4)
                      Endif(3)
           Endif(2)
Else(1)    "see further definitions"
Endif(1)
```

Calculation by active Function LV_PV_CS = 1:

The calculation conditions remains from "Calculation by getting active Function:"
If there is a transition from VS_FIL < C_VS_MAX_FAC_PV_COR to VS_FIL ≥ C_VS_MAX_FAC_PV_COR during active clutch, then the calculation remains as well as VS_FIL < C_VS_MAX_FAC_PV_COR fullfilled condition . FAC_PV_CS_COR and FAC_PV_GS_COR doesn't exists at the same time.

Calculation triggered by leaving active Function LV_PV_CS = 1 to 0:

At transition LV_PV_CS 1 to 0 then FAC_PV_CS_COR is ramped down as followed:

- FAC_PV_CS_COR is basically calculated like the active function.

```
If(1)       $LV\_PV\_CS_{(n)} = 0$ and $LV\_PV\_CS_{(n-1)} = 1$
Then(1)     If(2)       $LV\_AT = 1$
            Then(2)     start decremented timer $T\_PV\_CS\_COR = C\_T\_FAC\_PV\_CS\_AT$
                        If(3)       $LV\_RNG\_L\_REQ = 1$
                        Then(3)     $FAC\_PV\_CS\_COR = IP\_FAC\_PV\_CS\_COR\_AT\_RNG\_L * T\_PV\_CS\_COR / C\_T\_FAC\_PV\_CS\_AT$
                        Else(3)     $FAC\_PV\_CS\_COR = IP\_FAC\_PV\_CS\_COR\_AT * T\_PV\_CS\_COR / C\_T\_FAC\_PV\_CS\_AT$
                        Endif(3)
            Else(2)     start decremented timer $T\_PV\_CS\_COR = C\_T\_FAC\_PV\_CS\_MT$
                        $FAC\_PV\_CS\_COR = IP\_FAC\_PV\_CS\_COR\_MT * \frac{T\_PV\_CS\_COR}{C\_T\_FAC\_PV\_CS\_MT}$
            Endif(2)
            Function is finished at $T\_PV\_CS\_COR = 0$.
Endif(1)
```

The progression correction is retriggered at detection of LV_PV_CS 0 to 1 during ramp.

Progression correction at starting FAC_PV_GS_COR *(only MT or TCT)*:

Calculation by transition Function LV_PV_CS = 0 to 1:

```
IF(1)       $VS\_FIL \geq C\_VS\_MAX\_FAC\_PV\_COR$
Then (1)    FAC_PV_GS_COR  calculation see bellow
Else (1)    FAC_PV_GS_COR is not calulated FAC_PV_CS_COR is calculated furthermore
endif
```

Calculation by active Function LV_PV_CS = 1:

```
IF(1)       $VS\_FIL \geq C\_VS\_MAX\_FAC\_PV\_COR$
Then(1) IF(2) $LV\_VAR\_TCT = 1$
    Then(2)     $FAC\_PV\_GS\_COR = IP\_FAC\_PV\_GS\_COR\_TCT * IP\_FAC\_VS\_PV\_GS\_COR\_TCT$
    Else (2)IF(3) $LV\_AT = 0$
```

```
                Then(3)    FAC_PV_GS_COR = IP_FAC_PV_GS_COR_MT *
                                           IP_FAC_VS_PV_GS_COR_MT
                Else(3)    FAC_PV_GS_COR = 0
Else(1)    FAC_PV_GS_COR = 0
Endif
```

Calculation leaving of the active function LV_PV_CS 1 to 0:

At transition LV_PV_CS 1 to 0 then FAC_PV_GS_COR is ramped down as followed:

- FAC_PV_GS_COR is basically calculated like the active function.

```
IF    $LV\_PV\_CS_{(n)} = 0$ and $LV\_PV\_CS_{(n-1)} = 1$
Then      start decremented timer T_PV_GS_COR = C_T_FAC_PV_GS
    FAC_PV_GS_COR = (IP_FAC_PV_GS_COR_x * IP_FAC_VS_PV_GS_COR_x) *
                                   T_PV_GS_COR / C_T_FAC_PV_GS
          Function is finished at T_PV_GS_COR = 0.
Endif
```

The progression correction is retriggered at detection of LV_PV_CS 0 to 1 during ramp.

Progression correction general FAC_PV_BAS_COR:

Dependig on the type of gearbox and state of reverse gear the general correction is calculated. This correction depends from N, VS_FIL and PV_AV_RAW.

```
IF        STATE_GEAR_REV_AT_AMT = 1 or STATE_GEAR_REV_CAN = 1
Then      FAC_PV_BAS_COR = IP_FAC_PV_BAS_COR_x * IP_FAC_VS_PV_BAS_COR_x
                           + IP_FAC_PV_BAS_COR_GEAR_REV_y
Else      FAC_PV_BAS_COR = IP_FAC_PV_BAS_COR_x * IP_FAC_VS_PV_BAS_COR_x
Endif


IF    LV_VAR_TCT = 1
then x = TCT
          y = TCT
else      if      LV_AT = 1
          then    y = AT
                  (in first recurrence x is initialised with AT)
                  Progression changed if mode of transmission is changed (AT_SPT...
                  sport mode, AT_MAN...manual mode), changes become only valid if
                  PV_AV is under or above a certain threshold to avoid jumps in PV_AV
                  signal.
                  if      [(STATE_ETCU_PROG_INFO = 2 and LV_VAR_BN = 1) or
                           (STATE_ETCU_PROG_INFO = 1 and LV_VAR_BN = 0) ]   and
```

```
                [PV_AV < C_THD_MIN_PV_AV or
                 PV_AV > C_THD_MAX_PV_AV]
            Then   x = AT_MAN
            Elseif [(STATE_ETCU_PROG_INFO = 1 and LV_VAR_BN = 1) or
                    (STATE_ETCU_PROG_INFO = 2 and LV_VAR_BN = 0) ]  and
                   [PV_AV < C_THD_MIN_PV_AV or
                    PV_AV > C_THD_MAX_PV_AV]
            Then   x = AT_SPT
            Elseif  [PV_AV < C_THD_MIN_PV_AV or
                    PV_AV > C_THD_MAX_PV_AV]
            Then   x = AT
            Else   x = keep last
            endif
        else    x = MT
                y = MT
        endif
    endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_PV_BAS_COR_TCT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N__PV_FAC_COR	6	0...1FE0H	0...8160	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Basic correction					
IP_FAC_PV_BAS_COR_MT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N__PV_FAC_COR	6	0...1FE0H	0...8160	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Basic correction					
IP_FAC_PV_BAS_COR_AT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N__PV_FAC_COR	6	0...1FE0H	0...8160	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Basic correction					
IP_FAC_PV_CS_COR_TCT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N_DIF__PV_FAC_COR	6	0...FFFFH	-32768...32767	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Clutch intervention correction					
IP_FAC_PV_CS_COR_MT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N_DIF__PV_FAC_COR	6	0...FFFFH	-32768...32767	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Clutch intervention correction					
IP_FAC_PV_GS_COR_TCT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N__PV_FAC_COR	6	0...1FE0H	0...8160	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Gear shift intervention correction					
IP_FAC_PV_GS_COR_MT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N__PV_FAC_COR	6	0...1FE0H	0...8160	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Gear shift intervention correction					
IP_FAC_VS_PV_GS_COR_TCT	8	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL__PV_FAC_COR	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction					
IP_FAC_VS_PV_GS_COR_MT	8	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL__PV_FAC_COR	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction					
IP_FAC_VS_PV_BAS_COR_MT	8	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL__PV_FAC_COR	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction					
IP_FAC_VS_PV_BAS_COR_AT	8	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL__PV_FAC_COR	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction					
IP_FAC_VS_PV_BAS_COR_TCT	8	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL__PV_FAC_COR	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction					
C_T_FAC_PV_GS	1	1...FFFFH	0.01...655.35	0.01	[s]
Timer switch off GS correction					
C_VS_MAX_FAC_PV_COR	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Starting or gear shift detection threshold					
IP_FAC_PV_CS_COR_AT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N_DIF__PV_FAC_COR	6	0...FFFFH	-32768...32767	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Clutch intervention correction					
C_T_FAC_PV_CS_AT	1	1...FFFFH	0.01...655.35	0.01	[s]
Timer switch off CS correction for AT					
C_T_FAC_PV_CS_MT	1	1...FFFFH	0.01...655.35	0.01	[s]
Timer switch off CS correction for MT and TCT					
C_VS_MAX_FAC_PV_COR_AT	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Starting or gear shift detection threshold for AT					
IP_FAC_PV_CS_COR_AT_RNG_L	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N_DIF__PV_FAC_COR	6	0...FFFFH	-32768...32767	1	[rpm]

LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Clutch intervention correction in low range mode					
IP_FAC_PV_BAS_COR_AT_MAN	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N__PV_FAC_COR	6	0...1FE0H	0...8160	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Basic correction for AT in 'M' mode (manual)					
IP_FAC_PV_BAS_COR_AT_SPT	6*6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_N__PV_FAC_COR	6	0...1FE0H	0...8160	1	[rpm]
LDPM_PV_AV_RAW__PV_FAC_COR	6	0...FFH	0...99.60937	0.390625	[%]
Basic correction for AT in 'S' mode (sport)					
IP_FAC_VS_PV_BAS_COR_AT_MAN	8	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL__PV_FAC_COR	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction for AT in 'M' mode (manual)					
IP_FAC_VS_PV_BAS_COR_AT_SPT	8	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL__PV_FAC_COR	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction for AT in 'S' mode (sport)					
C_THD_MIN_PV_AV	1	0...FFH	0...99.60937	0.390625	[%]
Threshold for switching progression					
C_THD_MAX_PV_AV	1	0...FFH	0...99.60937	0.390625	[%]
Threshold for switching progression					
IP_FAC_PV_BAS_COR_GEAR_REV_TCT	6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL_PV_FAC_COR_GEAR_REV	6	0...FFFFH	0...511.99218	0.0078125	[km/h]
Additve correction at reverse gear for TCT					
IP_FAC_PV_BAS_COR_GEAR_REV_MT	6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL_PV_FAC_COR_GEAR_REV	6	0...FFFFH	0...511.99218	0.0078125	[km/h]
Additve correction at reverse gear for MT					
IP_FAC_PV_BAS_COR_GEAR_REV_AT	6	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_VS_FIL_PV_FAC_COR_GEAR_REV	6	0...FFFFH	0...511.99218	0.0078125	[km/h]
Additve correction at reverse gear for AT					

8.2.5 Kick-Down-Recognition

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_KD	V/O	0...1H	0...1	1	[-]
Flag for Kick-Down-Recognition (internally recognized from both PVS-voltages)					

Input data:

	STATE_PVS_DIAG		

FUNCTION DESCRIPTION:

The recognition of kick-down (LV_KD) occurs, if the filtered pedal values (V_PVS_FIL_1) of channel one is exceed the respective threshold values C_V_PVS_THD_KD_x.
Because of more jitters at Channel two only Channel one is used.

Application conditions:

Initialisation: 0

Recurrence: 10ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0, set LV_KD=0

Formula section:

```
If(1)      STATE_PVS_DIAG = PVS_NO_ERROR   or
           STATE_PVS_DIAG = PVS_LIH_PRED
Then(1)
    If(2)  V_PVS_FIL_1 > C_V_PVS_THD_KD        and    LV_KD = 0
    Then(2)        LV_KD = 1
    Else(2)
       If (3) V_PVS_FIL_1 < C_V_PVS_THD_KD - C_V_PVS_THD_KD_HYS        and
              LV_KD = 1

       Then(3)     LV_KD = 0
       Else(3)     LV_KD = 1
Else(1)    LV_KD = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_PVS_THD_KD	1	0...FFH	0...4.998	0.0195313	[V]
Pedal value to kick-down-recognition (Channel 1)					
C_V_PVS_THD_KD_HYS	1	0...FFH	0...4.998	0.0195313	[V]
Hysteresis of kick down recognition Channel 1					

8.2.6 Selection of maximum pedal value

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PV_MAX	V/O	0...FFH	0...99.60937	0.390625	[%]
Maximum selection of Pedal value interpretation					
PV_CRU	V/O	0...FFH	0...99.60937	0.390625	[%]
Pedal value interpretation of Cruise control					
PV_CRU_1	V	0...FFH	0...99.60937	0.390625	[%]
Pedal value interpretation of Cruise control temporary					
PV_CRU_RAW	V	0...FFH	0...99.60937	0.390625	[%]
Pedal value interpretation of Cruise control rawsignal					

Input data:

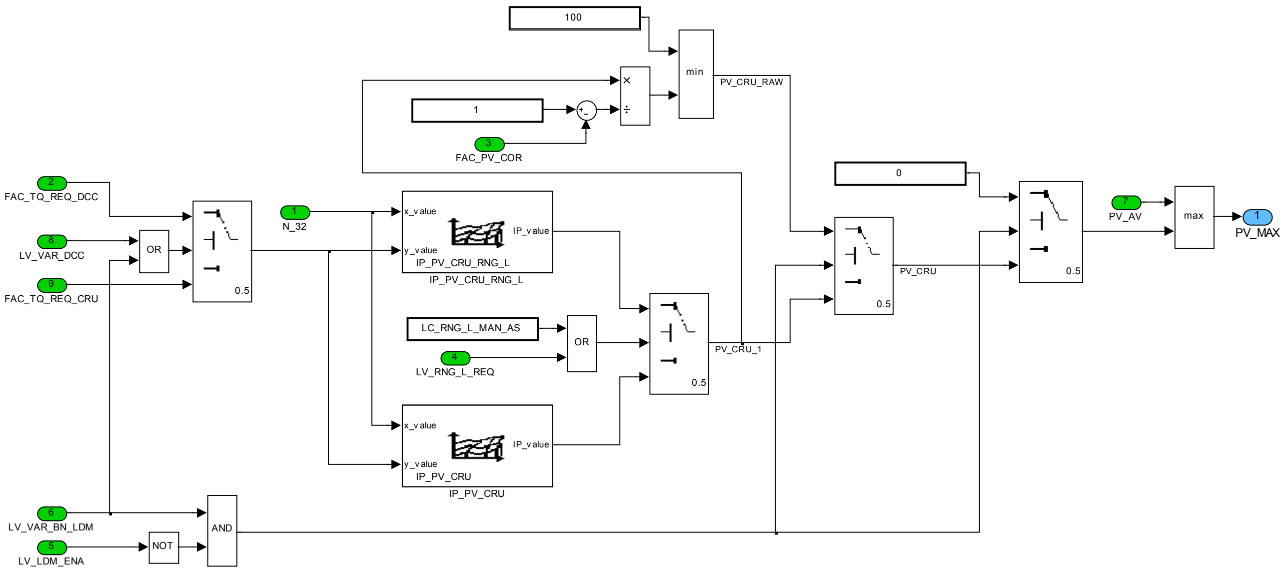
N_32	FAC_TQ_REQ_CRU	LV_LDM_ENA	LC_RNG_L_MAN_AS
LV_RNG_L_REQ	FAC_TQ_REQ_DCC	LV_VAR_DCC	LV_VAR_BN_LDM

FUNCTION DESCRIPTION:

For CAN messages during cruise control and other functions the maximum selection between Pedal value and interpretation of cruise control charge request is needed.

The maps IP_PV_CRU__RNG_L and IP_PV_CRU have to be inverse maps of IP_FAC_TQ_REQ_RNG_L and IP_FAC_TQ_REQ_DRIV.

Signal flow diagram:



Calibration data:

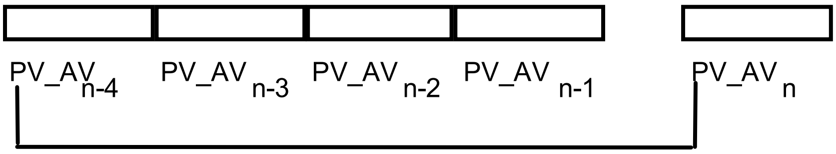
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_PV_CRU	8*8	0...FFH	0...99.60937	0.390625	[%]
LDPM_N__FAC_TQ_REQ_DRIV	8	0...FFH	0...8160	32	[rpm]
LDPM_FAC_TQ_CRU_DCC	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
Scaling factor for requested torque at clutch from driver					
IP_PV_CRU_RNG_L	8*8	0...FFH	0...99.60937	0.390625	[%]
LDPM_N__FAC_TQ_REQ_DRIV	8	0...FFH	0...8160	32	[rpm]
LDPM_FAC_TQ_CRU_DCC	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
Scaling factor for requested torque at clutch from driver in low range mode					

8.2.7 Calculation of PV_AV gradient:

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PV_AV_GRD	V/O	0...FFH	-1250... 1240.23437	9.765625	[%/s]
Pedal value gradient					

The PV_AV gradient is calculated at intervals of 10 ms.
To this effect, the PV_AV values are written into the ring buffer. Thus, the last four PV_AV values are available in addition to the current PV_AV value.



The PV_AV gradient is based on a PV_AV change over 40 ms.

The PV_AV_GRD reflects the gradient of the actual driver's demand prior to the corrections.

$$PV\_AV\_GRD = (PV\_AV_{n} - PV\_AV_{n-4}) * 25$$

8.3 Pedal value sensor diagnosis (PVS_x)

FUNCTION DESCRIPTION:

General information:

The following diagnosis are performed:

- Diagnosis of sensor–supply voltage VCC_PVS_i
- PVS-synchronization control (RATIO-Check)
- Signal range check low and high (SRCL, SRCH)
- PVS double error
- Plausibility between brake / PVS

Input data:

VCC_PVS_1	V_PVS_1	LV_IGK	N
VCC_PVS_2	V_PVS_2	VS	PV_AV
LV_IM_BLS	LV_IM_BTS	BRAKE_PRS	LV_BRAKE_DET
LV_ERR_BLS_PLAUS	STATE_BRAKE_PRS		

8.3.1 Global PVS system error LV_ERR_PVS

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_PVS	O/V	0...1H	0...1	1	[-]
Global PVS system error					

Description:

The global error LV_ERR_PVS is set if at least one PVS error is set.

Application conditions:

Initialisation: 0 at LV_IGK 0->1 **or** reset

Recurrence: 10ms

Activation: LV_IGK = 1

Formula section:

If LV_ERR_V_REF_x = 1 *// 1 or 2*

or LV_ERR_PVS_x = 1

or LV_ERR_PVS_RATIO = 1

Then LV_ERR_PVS = 1

Else LV_ERR_PVS = 0

Endif

8.3.2 Diagnosis of sensor-supply voltage VCC_PVS_i

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_V_REF_1	O/V	0...1H	0...1	1	[-]
Present error PVS V_REF					
LV_ERR_V_REF_2	O/V	0...1H	0...1	1	[-]
Present error PVS V_REF					
LV_CDN_DIAG_V_REF_1	O/V	0...1H	0...1	1	[-]
Diagnosis condition PVS V_REF					
LV_CDN_DIAG_V_REF_2	O/V	0...1H	0...1	1	[-]
Diagnosis condition PVS V_REF					
ERR_SYM_V_REF_1	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom PVS V_REF					
ERR_SYM_V_REF_2	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom PVS V_REF					
LV_END_DIAG_V_REF_1	O/V	0...1H	0...1	1	[-]
End of diagnosis PVS V_REF					
LV_END_DIAG_V_REF_2	O/V	0...1H	0...1	1	[-]
End of diagnosis PVS V_REF					

General information:

The PVS sensor supply voltage 5V [Range 0...9.9902 V] is compared with the measured actual supply voltage every 10 ms (forming differences).

x = 1 and 2

Description:

If diagnosis condition is reached and error symptom is detected ERR_SYM_V_REF_i is set. After the debounce LV_ERR_V_REF_1 (at error channel 1), LV_ERR_V_REF_2 (at error channel 2) are set. The faulty channel is not evaluated during this driving cycle.

Remark:

During the debounce time (ERR_SYM_V_REF_x = SYM_3) the PV_AV value is limited upwards to the last valid PVS-value (see chapter 4 "Determination of accelerator pedal value").

Error-symptoms are defined to this diagnosis function as following :

```
 │  │  │  └─> -                                   (= SYM_0)
 │  │  └───> -                                    (= SYM_1)
 │  └─────> -                                     (= SYM_2)
 └───────> PVS - Voltage controller faulty        (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset

Recurrence: 10ms

Activation:

```
If      LV_IGK = 1                  and
        LV_ERR_V_REF_x = 0
Then    LV_CDN_DIAG_V_REF_x = 1
Else    LV_CDN_DIAG_V_REF_x = 0
Endif
```

Formula section:

Error detection PVS-Channel:

```
If(1)       | 5V – VCC_PVS_x | > C_VCC_DIF_PVS
Then(1)     ERR_SYM_V_REF_x = SYM_3
Set of LV_ERR_V_REF_1 (or 2) and LV_ERR_PVS:
    If(2)       ABC counter reaches max
                    // CTR_ABC_V_REF_x >=  C_ABC_MAX_V_REF_x
    Then(2)     LV_ERR_V_REF_x =  1       // set for this DC
    Endif(2)
Else(1)     ERR_SYM_V_REF_x = NO_SYM
Endif(2)
```

Calculation end of diagnosis

LV_END_DIAG_V_REF_x is calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VCC_DIF_PVS	1	0...3FFH	0...9.9902	9.7656e-3	[V]
max. permissible variance of the supply voltage of 10V					
C_ABC_INC_V_REF_1	1	0...FFH	0...255	1	[-]
Anti-bounce increment value (x=1,2)					
C_ABC_INC_V_REF_2	1	0...FFH	0...255	1	[-]
Anti-bounce increment value (x=1,2)					
C_ABC_MAX_V_REF_1	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value (x=1,2)					
C_ABC_MAX_V_REF_2	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value (x=1,2)					

8.3.3 PVS-synchronization control (RATIO-Check)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_PVS_RATIO	O/V	0...1H	0...1	1	[-]
Present error PVS_RATIO					
LV_CDN_DIAG_PVS_RATIO	O/V	0...1H	0...1	1	[-]
Diagnosis condition PVS_RATIO					
ERR_SYM_PVS_RATIO	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom PVS_RATIO					
LV_END_DIAG_PVS_RATIO	O/V	0...1H	0...1	1	[-]
End of diagnosis PVS_RATIO					

Description:

In order to have an error free system the pedal value sensor voltages V_PVS_x may not deviate more than IP_PVS_RATIO_HYS (V_PVS_2 must be doubled!) from each other. The smaller PVS-voltage serves as a break point for the characteristic curve.

The RATIO-Check is then only carried out if both channels are error - free.

ERR_SYM_PVS_RATIO is set when the error conditions are present; after debounce the error bit LV_ERR_PVS_RATIO is set.

During the debounce time (ERR_SYM_PVS_RATIO = SYM_3) the PV_AV value is limited upwards to the last valid PVS-value (see chapter 4 "Determination of accelerator pedal value").

Error-symptoms are defined to this diagnosis function as following :

```
|_|_|_|_|
 | | | └─> -                            (= SYM_0)
 | | └───> -                            (= SYM_1)
 | └─────> -                            (= SYM_2)
 └───────> PVS voltages not plausible   (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset

Recurrence: 10ms

Activation:

```
If      LV_IGK = 1                  and
        LV_ERR_PVS = 0              and
Then    LV_CDN_DIAG_PVS_RATIO = 1
Else    LV_CDN_DIAG_PVS_RATIO = 0
Endif
```

Formula section:

Error detection:

```
V_PVS_MIN  =  Min(V_PVS_1, 2 * V_PVS_2)
If(1)       | V_PVS_1 - (2 * V_PVS_2) |  >  IP_PVS_RATIO_HYS__V_PVS_MIN
Then(1)     ERR_SYM_PVS_RATIO = SYM_3
Set of LV_ERR_PVS_RATIO and LV_ERR_PVS:
    If(2)           ABC counter reaches max
                        // CTR_ABC_PVS_RATIO  >=  C_ABC_MAX_PVS_RATIO
    Then(2)         LV_ERR_PVS_RATIO =       1        // set for this DC
    Endif(2)
Else(1)     ERR_SYM_PVS_RATIO = NO_SYM
Endif(1)
```

Calculation end of diagnosis

LV_END_DIAG_PVS_RATIO is calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_PVS_RATIO_HYS	2	0...3FFH	0...4.99511	4.8828e-3	[V]
LDP_V_PVS_MIN	2	0...3FFH	0...4.99511	4.8828e-3	[V]
permissible variance of the PVS-values					
C_ABC_INC_PVS_RATIO	1	0...FFH	0...255	1	[-]
Anti-bounce increment value					
C_ABC_MAX_PVS_RATIO	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value					

8.3.4 Signal-Range-Check (High: SRCH / Low: SRCL)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_PVS_1	O/V	0...1H	0...1	1	[-]
Global PVS system error					
LV_ERR_PVS_2	O/V	0...1H	0...1	1	[-]
Global PVS system error					
LV_CDN_DIAG_PVS_1	O/V	0...1H	0...1	1	[-]
Diagnosis condition PVS					
LV_CDN_DIAG_PVS_2	O/V	0...1H	0...1	1	[-]
Diagnosis condition PVS					
ERR_SYM_PVS_1	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom PVS					
ERR_SYM_PVS_2	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom PVS					
LV_END_DIAG_PVS_1	O/V	0...1H	0...1	1	[-]
End of diagnosis PVS					
LV_END_DIAG_PVS_2	O/V	0...1H	0...1	1	[-]
End of diagnosis PVS					

General information:

Both diagnosis are only performed if the channel is not debounced as faulty (x = 1 and 2).

- The **SRCH** diagnosis is to detect a short to Vbatt error.
- The **SRCL** diagnosis is to detect a short to Ground /open load error.

Description:

SRCH diagnosis:

The sensor voltage must lie below the permissible voltage limits (C_V_PVS_MAX_DIAG_x) for an error-free system. ERR_SYM_PVS_x is set while the error symptom is present, the bits LV_ERR_PVS_X are set after debounce. The channel x is no longer evaluated during this driving cycle.

SRCL diagnosis:

The sensor voltage must lie upper the permissible voltage limits (C_V_PVS_MIN_DIAG_x) for an error-free system. ERR_SYM_PVS_x is set while the error symptom is present, the bits LV_ERR_PVS_X are set after debounce. The channel x is no longer evaluated during this driving cycle.

Remark:

During the debounce time (ERR_SYM_PVS_x = SYM_0 or SYM_1) and while error conditions are true the PV_AV value is limited upwards to the last valid PVS-value (see chapter 4 "Determination of accelerator pedal value").

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Vbatt                (= SYM_0)
 │ │ └───> Short circuit to GND or open load     (= SYM_1)
 │ └─────> -                                     (= SYM_2)
 └───────> -                                     (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset

Recurrence: 10ms

Activation:

```
If      LV_IGK = 1              and
        LV_ERR_PVS_x = 0        and
        LV_ERR_V_REF_x = 0
Then    LV_CDN_DIAG_PVS_x = 1
Else    LV_CDN_DIAG_PVS_x = 0
Endif
```

Formula section:

Short circuit to Vbatt:

```
If(1)        V_PVS_x > C_V_PVS_MAX_DIAG_x          (X = 1,2)
Then(1)      ERR_SYM_PVS_x = SYM_0
Short circuit to GND or open load
Else(1)If(2)     V_PVS_x < C_V_PVS_MIN_DIAG_x
    Then(2)      ERR_SYM_PVS_x = SYM_1
    PVS voltage is in valid range
    Else(2)      ERR_SYM_PVS_x = NO_SYM
    Endif(2)
Endif(1)
```

Calculation end of diagnosis and present failure

LV_ERR_PVS_x and LV_END_DIAG_PVS_x are calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_PVS_MAX_DIAG_1	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Upper diagnosis threshold					
C_V_PVS_MAX_DIAG_2	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Upper diagnosis threshold					
C_V_PVS_MIN_DIAG_1	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lower diagnosis threshold					
C_V_PVS_MIN_DIAG_2	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lower diagnosis threshold					
C_ABC_INC_PVS_1	1	0...FFH	0...255	1	[-]
Anti-bounce increment value					
C_ABC_INC_PVS_2	1	0...FFH	0...255	1	[-]
Anti-bounce increment value					
C_ABC_MAX_PVS_1	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value					
C_ABC_MAX_PVS_2	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value					

8.3.5 PVS – Double error

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_PVS_DOUBLE	O/V	0...1H	0...1	1	[-]
Present error PVS DOUBLE					
LV_CDN_DIAG_PVS_DOUBLE	O/V	0...1H	0...1	1	[-]
Diagnosis condition PVS DOUBLE					
ERR_SYM_PVS_DOUBLE	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom PVS DOUBLE					
LV_END_DIAG_PVS_DOUBLE	O/V	0...1H	0...1	1	[-]
End of diagnosis PVS DOUBLE					

Description:

In order to illuminate the MIL if both PVS channels are detected as faulty there is a seperate error flag LV_ERR_PVS_DOUBLE (all other PVS errors shall be calibrated as non – MIL relevant).

If a channel is already detected as faulty (through sensor supply, SRCH, SRCL) and afterwards another sensor error appears, then the system goes into the PVS limp home for this driving cycle (see limited dynamic 2).

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                          (= SYM_0)
 │ │ └───> -                          (= SYM_1)
 │ └─────> -                          (= SYM_2)
 └───────> PVS - Double error         (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset

Recurrence: 10ms

Activation:

```
If      LV_IGK = 1                    and
        LV_ERR_PVS_DOUBLE = 0
Then    LV_CDN_DIAG_PVS_DOUBLE = 1
        LV_END_DIAG_PVS_DOUBLE = 1
Else    LV_CDN_DIAG_PVS_DOUBLE = 0
Endif
```

Formula section:

If [LV_ERR_V_REF_1 = 1 **or**

```
        LV_ERR_PVS_1 = 1 ]        and
        [ LV_ERR_V_REF_2 = 1      or
        LV_ERR_PVS_2 = 1 ]
Then    ERR_SYM_PVS_DOUBLE = SYM_3
        LV_ERR_PVS_DOUBLE = 1          //set direct
Endif
```

8.3.6 Plausibility between PVS signal and brake detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_PVS_BLS_NOT_PLAUS	O/V	0...1H	0...1	1	[-]
Plausibility error PVS / Brake recognized with error entry					
LV_CDN_DIAG_PVS_BLS_NOT_PLAUS	O/V	0...1H	0...1	1	[-]
Diagnosis condition Plausibility error PVS / Brake					
ERR_SYM_PVS_BLS_NOT_PLAUS	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom Plausibility error PVS / Brake					
LV_END_DIAG_PVS_BLS_NOT_PLAUS	O/V	0...1H	0...1	1	[-]
End of diagnosis Plausibility error PVS / Brake					
LV_DET_NOT_PLAUS_ACT	O/V	0...1H	0...1	1	[-]
PV_AV / Brake signal current not plausible					
CTR_PVS_BLS_NOT_PLAUS_ACT	V	0...FFH	0...255	1	[-]
Counter plausibility error PVS / Brake recognized without error entry					
LV_PVS_BLS_NOT_PLAUS_ACT	O/V	0...1H	0...1	1	[-]
Plausibility error PVS / Brake recognized without error entry					
LV_PV_AV_LIM_ACT	O/V	0...1H	0...1	1	[-]
Logical value for PV_AV over threshold for PVS_BLS_BTS_PLAUS					
CTR_C_T_VS_BRAKE_MAX	V	0...FFH	0...2.55	0.01	[s]
Delay timer for vehicle speed in PVS_BLS_BTS_PLAUS					
LV_CDN_VS_VLD_1	V	0...1H	0...1	1	[-]
Logical value for VS condition for PVS_BLS_BTS_PLAUS					
LV_CDN_VS_VLD_2	V	0...1H	0...1	1	[-]
Logical value for brake pressure condition for PVS_BLS_BTS_PLAUS					
PV_AV_MAX_BRAKE_DET	O/V	0...FFH	0...99.60937	0.390625	[%]
PV_AV_MAX_BRAKE_DET					

General information:

This function is to detect a mechanical stucking PVS sensor or a break actuation while the gas pedal is already activated.
The diagnosis algorithm can be configured with C_PVS_BLS_BTS_PLAUS_CONF:

0 Function is disabled
1 Function is enabled but no error entry
2 Function is enabled with error entry

Description:

Error detection is based on a plausibility check brake/pedal value.

LV_PVS_BLS_NOT_PLAUS_ACT is using the same debounce calibration data like LV_ERR_PVS_BLS_NOT_PLAUS but without error entry.
If the configuration is switched within the active engine run after an active error was present then the error is not reseted until the error memory is cleared.

Error-symptoms are defined to this diagnosis function as following :

```
☐☐☐☐
 |  |  |  └─> -                                  (= SYM_0)
 |  |  └───> -                                   (= SYM_1)
 |  └─────> -                                    (= SYM_2)
 └───────> PVS to Brake not plausible            (= SYM_3)
```

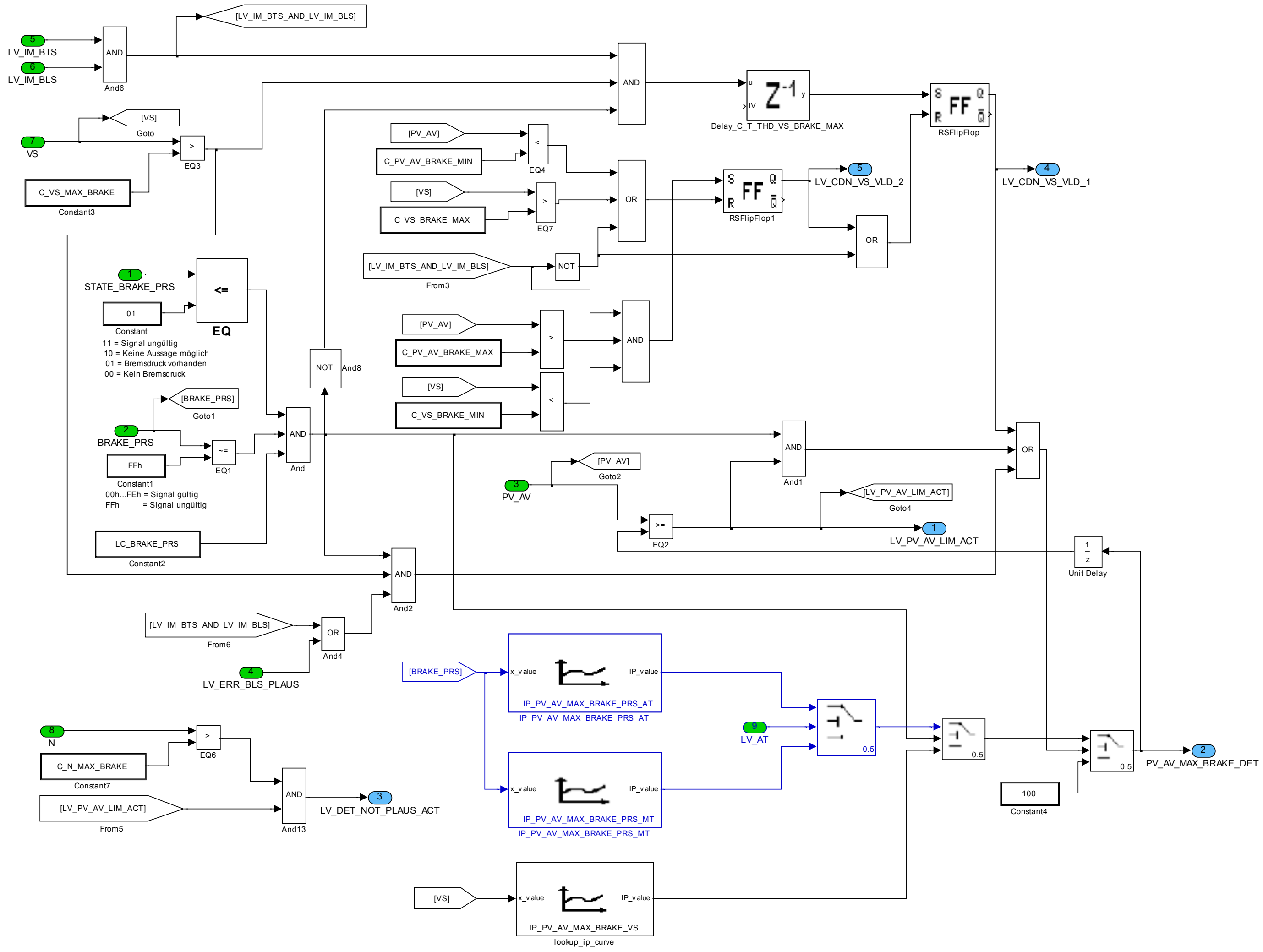
Application conditions:

Initialisation: at LV_IGK 0->1 **or** reset **or** at clearing error memory:
PV_AV_MAX_BRAKE_DET = 100%, all other output 0

Recurrence: 10 ms

```
Activation:  If    C_PVS_BLS_BTS_PLAUS_CONF = 0   or
                   LV_IGK = 0
             Then       Function is disabled  and
                        PV_AV_MAX_BRAKE_DET = 100%  and
                        LV_CDN_DIAG_PVS_BLS_NOT_PLAUS = 0
             Else       function is active
             Endif
```

Formula section:



Not Plaus Activation

```
IF(1)  LV_DET_NOT_PLAUS_ACT = 1
    IF(2)  C_PVS_BLS_BTS_PLAUS_CONF = 2
    THEN(2)  LV_CDN_DIAG_PVS_BLS_NOT_PLAUS = 1
             ERR_SYM_PVS_BLS_NOT_PLAUS = SYM_3
             Debounce of LV_ERR_PVS_BLS_NOT_PLAUS
    ENDIF(2)

    IF(3)  C_PVS_BLS_BTS_PLAUS_CONF = 1
```

```
    THEN(3)  LV_CDN_DIAG_PVS_BLS_NOT_PLAUS = 0
             $\mathrm{CTR\_PVS\_BLS\_NOT\_PLAUS\_ACT}_{n}$ =
             $\mathrm{CTR\_PVS\_BLS\_NOT\_PLAUS\_ACT}_{n-1}$ +
             C_ABC_INC_PVS_BLS_NOT_PLAUS
        IF(4)  CTR_PVS_BLS_NOT_PLAUS_ACT >=
               C_ABC_MAX_PVS_BLS_NOT_PLAUS
        THEN(4)  LV_PVS_BLS_NOT_PLAUS_ACT = 1
        ENDIF(4)
    ENDIF(3)
ELSE(1)
    IF(5)  [( LV_ERR_PVS_BLS_NOT_PLAUS = 0             and
             LV_PVS_BLS_NOT_PLAUS_ACT = 0 )            or
             (( LV_ERR_PVS_BLS_NOT_PLAUS = 1           or
             LV_PVS_BLS_NOT_PLAUS_ACT = 1 )            and
             LV_PV_AV_LIM_ACT = 0 )]
    THEN(5)
      IF(6)  C_PVS_BLS_BTS_PLAUS_CONF = 2
      THEN(6)  LV_CDN_DIAG_PVS_BLS_NOT_PLAUS = 1
               ERR_SYM_PVS_BLS_NOT_PLAUS = NO_SYM
               LV_ERR_PVS_BLS_NOT_PLAUS = 0
               The debounce counters are reset to 0.
      ENDIF(6)
      IF(7)  C_PVS_BLS_BTS_PLAUS_CONF = 1
      THEN(7)  LV_CDN_DIAG_PVS_BLS_NOT_PLAUS = 0
               CTR_PVS_BLS_NOT_PLAUS_ACT = 0
               LV_PVS_BLS_NOT_PLAUS_ACT = 0
      ENDIF(7)
    ELSE(5)
      IF(8)  C_PVS_BLS_BTS_PLAUS_CONF = 2
      THEN(8)  LV_CDN_DIAG_PVS_BLS_NOT_PLAUS = 1
               ERR_SYM_PVS_BLS_NOT_PLAUS = NO_SYM
      ENDIF(8)
      IF(9)  C_PVS_BLS_BTS_PLAUS_CONF = 1
      THEN(9)  LV_CDN_DIAG_PVS_BLS_NOT_PLAUS = 0
        IF(10)  CTR_PVS_BLS_NOT_PLAUS_ACT = 0
        THEN(10)   LV_PVS_BLS_NOT_PLAUS_ACT = 0
```

ENDIF(10)

ENDIF(9)

ENDIF(1)

Calculation end of diagnosis

LV_END_DIAG_PVS_BLS_NOT_PLAUS is calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_MAX_PVS_BLS_NOT_PLAUS	1	1...FFH	1...255	1	[-]
PVS_BLS_BTS_PLAUS counter maximum value					
C_ABC_INC_PVS_BLS_NOT_PLAUS	1	0...FFH	0...255	1	[-]
PVS_BLS_BTS_PLAUS counter increment value					
C_PVS_BLS_BTS_PLAUS_CONF	1	0...2H	0...2	1	[-]
Enable bit for plausibility check;					
C_PV_AV_BRAKE_MIN	1	0...FFH	0...99.60937	0.390625	[%]
Minimum PV_AV threshold for PVS_BLS_BTS_PLAUS					
C_PV_AV_BRAKE_MAX	1	0...FFH	0...99.60937	0.390625	[%]
Maximum PV_AV threshold for PVS_BLS_BTS_PLAUS					
C_VS_BRAKE_MIN	1	0...FFH	0...255	1	[km/h]
Minimum VS threshold for PVS_BLS_BTS_PLAUS					
C_VS_BRAKE_MAX	1	0...FFH	0...255	1	[km/h]
Maximum VS threshold for PVS_BLS_BTS_PLAUS					
C_VS_MAX_BRAKE	1	0...FFH	0...255	1	[km/h]
Maximum VS threshold for PVS_BLS_BTS_PLAUS					
LC_BRAKE_PRS	1	0...1H	0...1	1	[-]
Switch to enable brake pressure monitoring for plausibility check					
C_T_THD_VS_BRAKE_MAX	1	0...FFH	0...2.55	0.01	[s]
Delay timer threshold for vehicle speed in PVS_BLS_BTS_PLAUS					
C_N_MAX_BRAKE	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for ABC in PVS_BLS_BTS_PLAUS					
IP_PV_AV_MAX_BRAKE_VS	5	0...FFH	0...99.60937	0.390625	[%]
LDP_VS_IP_PV_AV_MAX_BRAKE	5	0...FFH	0...255	1	[km/h]
PV_AV limit for VS					
IP_PV_AV_MAX_BRAKE_PRS_MT	5	0...FFH	0...99.60937	0.390625	[%]
LDPM_BRAKE_PRS_IP_PV_AV_MAX	5	0...FFH	0...255	1	[bar]
PV_AV limit for brake pressure for manual gearbox					
IP_PV_AV_MAX_BRAKE_PRS_AT	5	0...FFH	0...99.60937	0.390625	[%]
LDPM_BRAKE_PRS_IP_PV_AV_MAX	5	0...FFH	0...255	1	[bar]
PV_AV limit for brake pressure for automatic gearbox					

8.4 Pedal value sensor limp home

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DYN_LIM_1_ACT	O/V	0...1H	0...1	1	[-]
Limited dynamic 1 is active / passive					
LV_DYN_LIM_2_ACT	O/V	0...1H	0...1	1	[-]
Limited dynamic 2 is active / passive					
LV_MTC_LIH_ACT	O/V	0...1H	0...1	1	[-]
Limited dynamic speed limit demand					
LV_N_SP_IS_LIH_ACT	O/V	0...1H	0...1	1	[-]
Locical value for LIH idle speed setpoint					
STATE_PVS_DIAG	O/V	0H 1H 2H 4H 8H 10H 20H 40H 80H	PVS_NO_ERROR PVS_LIH_PRED PVS_LIH_1 PVS_LIH_2 PVS_LIH_MIN PVS_DBL_ERROR PVS_LIH_BLS PVS_LIH_1_PRED PVS_LIH_2_PRED	1	[-]
Status of PVS-diagnosis					
LV_DET_ERR_PVS	V	0...1H	0...1	1	[-]
Logical value (set, if a PVS-error is present)					

General information:

Depending if there is a ratio or single or double PVS error, different limp homes are applied.

PVS_LIH_PRED: Is active during debounce – time is active. The pedal value is frozen in rising direction.

PVS_LIH_x_PRED: Is active if only one channel is detected as faulty and the remaining channel was not in idle detection range for a calibratable time. Pedal value is 0.

PV_LIH_x: Is active if only one channel is detected as faulty and the remaining channel was in idle detection range for a calibratable time. Limited dynamic 1 is applied (= Lower engine speed limitation and if brake or brake error is active then pedal value is 0).

PVS_LIH_MIN: Is active if ratio error is present. Pedal value is selected as a MIN-selection.

PVS_DBL_ERROR: Is active if both channels are detected as faulty or with a present ratio error and a high range error of the observed (=MIN) channel. Limited dynamic 2 is applied (= pedal value is 0, Idle speed with a higher idle speed setpoint. If brake or brake error is active then the normal idle speed setpoint is applied).

PVS_LIH_BLS: Is active if limited dynamic 1 or 2 is active and brake is detected or brake failure is present. Then pedal value is 0

Signal flow diagram:

PVS - Limp

8.4.1 PVS limp-home limited dynamic 1

Description:

The dynamic of the vehicle is limited in order secure controlled vehicle reactions and adequately longer reaction times for the driver when **one** of the both PVS-channels are detected as faulty.
When brake activity or an brake error is detected a lower engine speed limitation is active.
If the error memory is cleared the limp home LV_DYN_LIM_1_ACT is set to 0 as well.

Formula section:

```
If(1)     STATE_PVS_DIAG =     PVS_LIH_1         or
                               PVS_LIH_2         or
                               PVS_LIH_MIN       or
                               PVS_LIH_1_PRED    or
                               PVS_LIH_2_PRED    or
                               PVS_LIH_BLS
Then(1)   LV_DYN_LIM_1_ACT = 1
          If (2)      STATE_PVS_DIAG = PVS_LIH_BLS
          Then (2)    LV_MTC_LIH_ACT = 1          //Limp home engine speed limit
          Else (2)    LV_MTC_LIH_ACT = 0
          Endif (2)
Else(1)   LV_DYN_LIM_1_ACT = 0
Endif (1)
```

8.4.2 PVS limp-home limited dynamic 2

Description:

The dynamic of the vehicle is limited in order secure controlled vehicle reactions and adequately longer reaction times for the driver when **both** channels are detected as faulty.
If an error free pedal value sensor signal is no longer present “Limited dynamic 2” is set.
With the Limited dynamic 2 the vehicle is more handicaped than with Limited dynamic 1, the vehicle is driven by two idle speed setpoints.
If the error memory is cleared the limp home LV_DYN_LIM_2_ACT is set to 0 as well.

Formula section:

```
If(1)     STATE_PVS_DIAG = PVS_DOUBLE_ERROR
Then(1)   LV_DYN_LIM_2_ACT = 1
          LV_MTC_LIH_ACT = 1      (see “Engine speed limit coordination)
          If(2)       LV_BRAKE_DET = 1          or
                      LV_ERR_BLS_PLAUS = 1
          Then(2)     LV_N_SP_IS_LIH_ACT = 0              (PV_AV = 0)
          Else(2)     LV_N_SP_IS_LIH_ACT = 1
```

```
        Endif(2)
Else(1) LV_DYN_LIM_2_ACT = 1
Endif (1)
```

8.4.3 Determination of STATE_PVS_DIAG

Description:

The PVS Value selection (See also: ”Determination of accelerator pedal value, Value selection”) and the PVS-Limp home determination is based on the status value: STATE_PVS_DIAG.

Formula section:

Status of non-debounced PVS_error

```
If      ERR_SYM_V_REF_1 (and 2) = NO_SYM   and
        ERR_SYM_PVS_1 (and 2) = NO_SYM     and
        ERR_SYM_PVS_RATIO = NO_SYM
Then    LV_DET_ERR_PVS = 0
Else    LV_DET_ERR_PVS = 1
Endif
```

STATE_PVS_DIAG = PVS_NO_ERROR

```
If  LV_ERR_V_REF_1/2 = 0           and        // no detected and present failures
    LV_ERR_PVS_1/2 = 0             and
    LV_ERR_PVS_RATIO = 0           and
    LV_DET_ERR_PVS = 0
Then
    STATE_PVS_DIAG = PVS_NO_ERROR
Endif
```

STATE_PVS_DIAG = PVS_LIH_PRED

```
If  LV_ERR_PVS = 0                         and
    LV_DET_ERR_PVS = 1
Then
    STATE_PVS_DIAG = PVS_LIH_PRED
Endif
```

STATE_PVS_DIAG = PVS_LIH_1_PRED / PVS_LIH_1

```
If    LV_ERR_V_REF_2 = 1        or
      LV_ERR_PVS_2 = 1
Then
      STATE_PVS_DIAG = PVS_LIH_1_PRED
      If      V_PVS_1 < C_V_PVS_MIN_DIAG_MIN_1
      Then    decrement T_IS_PVS_ERR timer
              If      T_IS_PVS_ERR = 0
              Then    STATE_PVS_DIAG = PVS_LIH_1
              Endif
      Else    T_IS_PVS_ERR = C_T_IS_PVS_ERR
      Endif
Endif
```

STATE_PVS_DIAG = PVS_LIH_2_PRED / PVS_LIH_2

```
If    LV_ERR_V_REF_1 = 1        or
      LV_ERR_PVS_1 = 1
Then
      STATE_PVS_DIAG = PVS_LIH_2_PRED
      If      V_PVS_2 < C_V_PVS_MIN_DIAG_MIN_2
      Then    decrement T_IS_PVS_ERR timer
              If      T_IS_PVS_ERR = 0
              Then    STATE_PVS_DIAG = PVS_LIH_2
              Endif
      Else    T_IS_PVS_ERR = C_T_IS_PVS_ERR
      Endif
Endif
```

STATE_PVS_DIAG = PVS_LIH_MIN

```
If    LV_ERR_PVS_RATIO = 1
Then
      STATE_PVS_DIAG = PVS_LIH_MIN
Endif
```

STATE_PVS_DIAG = PVS_DBL_ERROR

```
If    LV_ERR_PVS_DOUBLE = 1
```

```
Then
    STATE_PVS_DIAG = PVS_DBL_ERROR
Endif
```

STATE_PVS_DIAG = PVS_LIH_BLS

```
If(1)         STATE_PVS_DIAG = PVS_LIH_1 or PVS_LIH_2 or PVS_LIH_MIN
Then (1)
              If (2)        LV_BRAKE_DET = 1          or
                            LV_ERR_BLS_PLAUS = 1
              Then (2)      STATE_PVS_DIAG = PVS_LIH_BLS
              Endif (2)
Endif (1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_IS_PVS_ERR	1	0...FFH	0...2.55	0.01	[s]
Minimum idle dwell time in error cases					
C_V_PVS_MIN_DIAG_MIN_1	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Upper diagnosis threshold for switching from faulty channel to error-free					
C_V_PVS_MIN_DIAG_MIN_2	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Upper diagnosis threshold for switching from faulty channel to error-free					

8.5 Brake light switch diagnosis (BLS/BTS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BLS_PLAUS	V/O	0...1H	0...1	1	[-]
Brake plausibility information flag					
LV_ERR_IM_BTS_PLAUS	V/O	0...1H	0...1	1	[-]
BTS plausibility error flag					
LV_CDN_DIAG_IM_BTS_PLAUS	V/O	0...1H	0...1	1	[-]
Diagnosis condition BTS					
ERR_SYM_IM_BTS_PLAUS	V/O	0H	NO_SYM	1	[-]
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Detected symptom BTS					
LV_END_DIAG_IM_BTS_PLAUS	V/O	0...1H	0...1	1	[-]
End of Diagnosis BTS					
LV_ERR_IM_BLS_PLAUS	V/O	0...1H	0...1	1	[-]
BLS plausibility error BLS					
LV_CDN_DIAG_IM_BLS_PLAUS	V/O	0...1H	0...1	1	[-]
Diagnosis condition BLS					
ERR_SYM_IM_BLS_PLAUS	V/O	0H	NO_SYM	1	[-]
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Detected symptom BLS					
LV_END_DIAG_IM_BLS_PLAUS	V/O	0...1H	0...1	1	[-]
End of Diagnosis BLS					
LV_BLS_EDGE	V	0...1H	0...1	1	[-]
LV_BLS_EDGE					
LV_BTS_EDGE	V	0...1H	0...1	1	[-]
LV_BTS_EDGE					
T_BTS_PLAUS	V	0...FFFFH	0...6553.5	0.1	[s]
T_BTS_PLAUS					

Input data:

LV_IGK	LV_IM_BLS	LV_IM_BTS	LV_CDN_VB_MIN_DIAG

FUNCTION DESCRIPTION:

General information:

In principle both brake light switch and brake test switch must be active at each braking. An edge mismatch between the brake light switch and brake test switch signal of +/- 20 µs can occur. A defective BTS is therefore detected by means of the following function.

Description:

The following errors can be detected:

- BLS implausible

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │ │ │ └─> -                              (= SYM_0)
  │ │ └───> -                              (= SYM_1)
  │ └─────> -                              (= SYM_2)
  └───────> LV_IM_BLS not plausible        (= SYM_3)
```

- BTS implausible

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │ │ │ └─> -                              (= SYM_0)
  │ │ └───> -                              (= SYM_1)
  │ └─────> -                              (= SYM_2)
  └───────> - LV_IM_BTS not plausible      (= SYM_3)
```

Application conditions:

Initialisation:	according filter type: ***"STD-INI"***
Recurrence:	100 ms (note: the formula section is not calculated during the first recurrence, when the activation conditions are fulfilled for the first time)
Activation:	LV_CDN_VB_MIN_DIAG = 1 **and** LV_IGK = 1

Formula section:

```
LV_BLS_EDGE = 0
If LV_IM_BLS = 0→1  or LV_IM_BLS = 1→0
Then LV_BLS_EDGE = 1
Endif

LV_BTS_EDGE = 0
If LV_IM_BTS = 0→1  or LV_IM_BTS = 1→0
Then LV_BTS_EDGE = 1
Endif

If(1) LV_BTS_EDGE = LV_BLS_EDGE
Then(1)
    If(2) LV_IM_BLS = LV_IM_BTS
```

```
Then(2)
    If(2.1) LV_BLS_EDGE = 1 and LV_BTS_EDGE = 1
    Then(2.1)
        Reset timer T_BTS_PLAUS
        LV_CDN_DIAG_IM_BLS_PLAUS = 1
        ERR_SYM_IM_BLS_PLAUS = "NO_SYM"
        (error flag reset by the error manager)

        LV_CDN_DIAG_IM_BTS_PLAUS = 1
        ERR_SYM_IM_BTS_PLAUS = "NO_SYM"
        (error flag reset by the error manager)
    Endif(2.1)
Else(2)
    If(3) CTR_ABC_IM_BLS_PLAUS > 0
    Then(3)
        Reset timer T_BTS_PLAUS
        LV_CDN_DIAG_IM_BLS_PLAUS = 1
        ERR_SYM_IM_BLS_PLAUS = "SYM_3"
        (error flag set by the error manager)
    Else(3)
        LV_CDN_DIAG_IM_BLS_PLAUS = 0
        ERR_SYM_IM_BLS_PLAUS = "NO_SYM"
    Endif(3)
    If(4) CTR_ABC_IM_BTS_PLAUS > 0
    Then(4)
        Reset timer T_BTS_PLAUS
        LV_CDN_DIAG_IM_BTS_PLAUS = 1
        ERR_SYM_IM_BTS_PLAUS = "SYM_3"
        (error flag set by the error manager)
    Else(4)
        LV_CDN_DIAG_IM_BTS_PLAUS = 0
        ERR_SYM_IM_BTS_PLAUS = "NO_SYM"
        If(5) ERR_SYM_IM_BLS_PLAUS = "NO_SYM"
        Then(5)
            If(6) T_BTS_PLAUS > C_T_BTS_PLAUS_MAX
            Then(6)
```

```
                        If(7) LV_IM_BLS = 1
                        Then(7)
                            Reset timer T_BTS_PLAUS
                            LV_CDN_DIAG_IM_BLS_PLAUS = 1
                            ERR_SYM_IM_BLS_PLAUS = "SYM_3"
                            (error flag set by the error manager)

                            LV_CDN_DIAG_IM_BTS_PLAUS = 0
                            ERR_SYM_IM_BTS_PLAUS = "NO_SYM"
                        Else(7)
                            Reset timer T_BTS_PLAUS
                            LV_CDN_DIAG_IM_BTS_PLAUS = 1
                            ERR_SYM_IM_BTS_PLAUS = "SYM_3"
                            (error flag set by the error manager)

                            LV_CDN_DIAG_IM_BLS_PLAUS = 0
                            ERR_SYM_IM_BLS_PLAUS = "NO_SYM"
                        Endif(7)
                    Else(6)
                        Increment timer T_BTS_PLAUS
                    Endif(6)
                Endif(5)
            Endif(4)
        Endif(2)
  Else(1)
        If(10) LV_BLS_EDGE = 0
        Then(10)
            LV_CDN_DIAG_IM_BLS_PLAUS = 1
            ERR_SYM_IM_BLS_PLAUS = "SYM_3"
            (error flag set by the error manager)

            LV_CDN_DIAG_IM_BTS_PLAUS = 0
            ERR_SYM_IM_BTS_PLAUS = "NO_SYM"
        Else(10)
            LV_CDN_DIAG_IM_BTS_PLAUS = 1
            ERR_SYM_IM_BTS_PLAUS = "SYM_3"
```

```
                (error flag set by the error manager)

                LV_CDN_DIAG_IM_BLS_PLAUS = 0
                ERR_SYM_IM_BLS_PLAUS = "NO_SYM"
        Endif(10)
    Endif(1)
```

Brake switch error detection:

```
IF      LV_ERR_IM_BLS_PLAUS = 1     or
        LV_ERR_IM_BTS_PLAUS = 1
Then    LV_ERR_BLS_PLAUS = 1
Else    LV_ERR_BLS_PLAUS = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_IM_BLS_PLAUS	1	0...FFH	0...255	1	[-]
BLS debounce counter increment					
C_ABC_MAX_IM_BLS_PLAUS	1	1...FFH	1...255	1	[-]
BLS debounce counter maximum value					
C_ABC_MAX_IM_BTS_PLAUS	1	1...FFH	1...255	1	[-]
BTS debounce counter maximum value, error debounce					
C_ABC_INC_IM_BTS_PLAUS	1	0...FFH	0...255	1	[-]
BTS debounce counter increment					
C_T_BTS_PLAUS_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum value of the time counter for debounce					

8.5.1 Brake detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_BRAKE_DET	V/O	0...1H	0...1	1	[-]
LV_IM_BLS or _BTS detected					

Input data:

LV_IM_BLS	LV_IM_BTS		

Description:

A brake operation is detected if brake light switch or brake test switch is engaged.

Application conditions:

Initialisation: 0

Recurrence: 10 ms

Activation: in all operating states

Formula section:

```
If      LV_IM_BLS = 1    or    LV_IM_BTS = 1
Then    LV_BRAKE_DET = 1
Else    LV_BRAKE_DET = 0
Endif
```

9 Driver request

Table of Contents

Index

9.1 Pedal Value Interpretation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TQ_REQ_DRIV	O/V	0...FFFFH	0...1.999969	3.05176E-5	-
scaling factor for requested torque at clutch from driver					
LV_RNG_L_REQ	O/V	0...1H	0...1	1	-
Activation condition for low-range-mode					

Input data:

LV_RNG_L	LV_RNG_L_AT	LV_SOF_SWI_REQ	N
PV_AV_GRD	PV_CUS	STATE_GEAR_REV_AT_AMT	STATE_GEAR_REV_CAN

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_TQ_REQ_DRIV_SOF_MAX	1	0...FFFFH	0...1.999969	3.05176E-5	-
scaling factor treshold for activation IP_FAC_TQ_REQ_DRIV_RNG_L					
C_PV_AV_GRD_SOF_MAX	1	0...FFH	-1.25E+3 ... 1.24023E+3	9.765625	%/s
Pedal value gradient threshold for recognition of LV_RNG_L					
C_PV_CUS_SOF_MIN	1	0...FFFFH	0...99.9984741	0.00152588	%
PV_CUS threshold for activation IP_FAC_TQ_REQ_DRIV_RNG_L					
C_T_PV_AV_GRD_SOF_MAX	1	0...FFFFH	0...655.35	0.01	s
Minimum time for pedal value gradient for recognition of LV_RNG_L					
LC_RNG_L_MAN_AS	1	0...1H	0...1	1	-
Calibration constant for to activate IP_FAC_TQ_REQ_DRIV_RNG_L (low range mode) in case if LV_RNG_L = 0 (default value = 0)					
IP_FAC_TQ_REQ_DRIV	12x12	0...FFFFH	0...1.999969	3.05176E-5	-
LDPM_N_FAC_TQ_REQ_DRIV	12	0...1FE0H	0...8.16E+3	1	rpm
LDPM_PV_CUS_FAC_TQ_REQ_DRIV	12	0...FFFFH	0...99.9984741	0.00152588	%
Scaling factor for requested torque at clutch from driver					
IP_FAC_TQ_REQ_DRIV_RNG_L	12x12	0...FFFFH	0...1.999969	3.05176E-5	-
LDPM_N_FAC_TQ_REQ_DRIV	12	0...1FE0H	0...8.16E+3	1	rpm
LDPM_PV_CUS_FAC_TQ_REQ_DRIV	12	0...FFFFH	0...99.9984741	0.00152588	%
Scaling factor for requested torque at clutch from driver in low range mode					

9.1.1 FUNCTION DESCRIPTION:

General information:

The scaling factor FAC_TQ_REQ_DRIV is the interpretation of the driver torque demand depending on engine speed and pedal value. Depending on gear mode LV_RNG_L (LV_RNG_L =1 if low range is engaged) two maps can be calibrated with different pedal value interpretation for normal and low range mode.

Application Condition

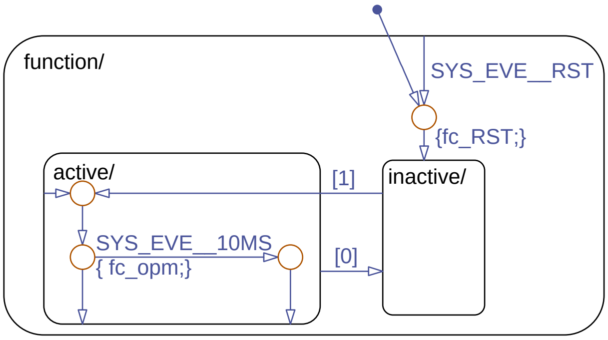
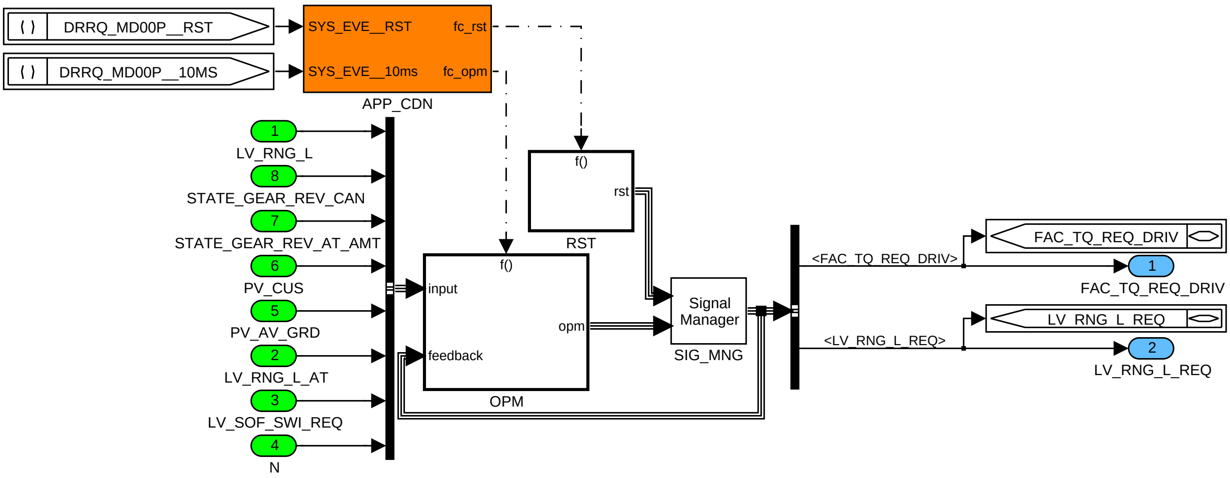


Figure 1 DRRQ_MD00P/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 28−Mar−2006

Figure 2 DRRQ_MD00P

9.1.1.1 Initialisation

Scaling factor FAC_TQ_REQ_DRIV is initialised to 0 at reset

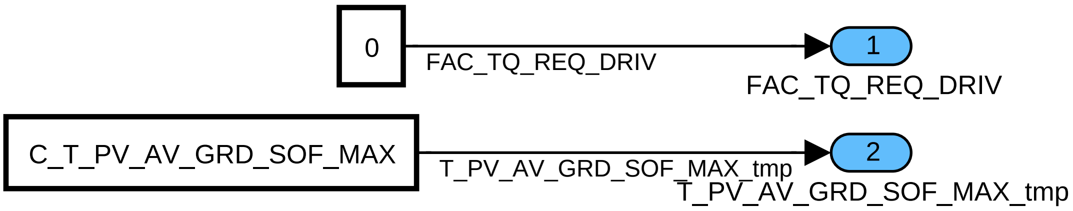


Figure 3 DRRQ_MD00P/ RST/ INI

9.1.1.2 Formula section

Gear mode detection

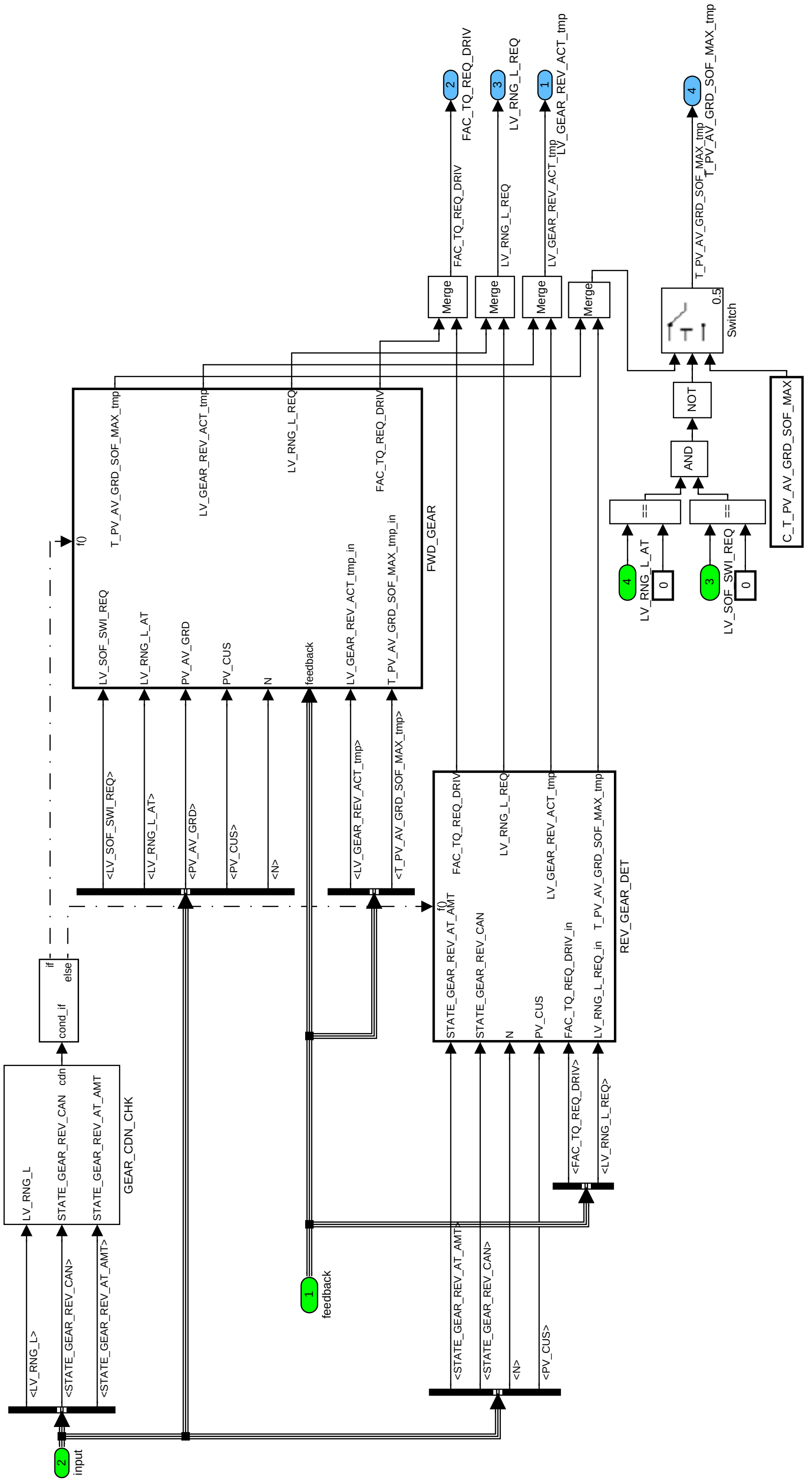


Figure 4 DRRQ_MD00P/ OPM/ OPM_10MS

Detection of gear condition

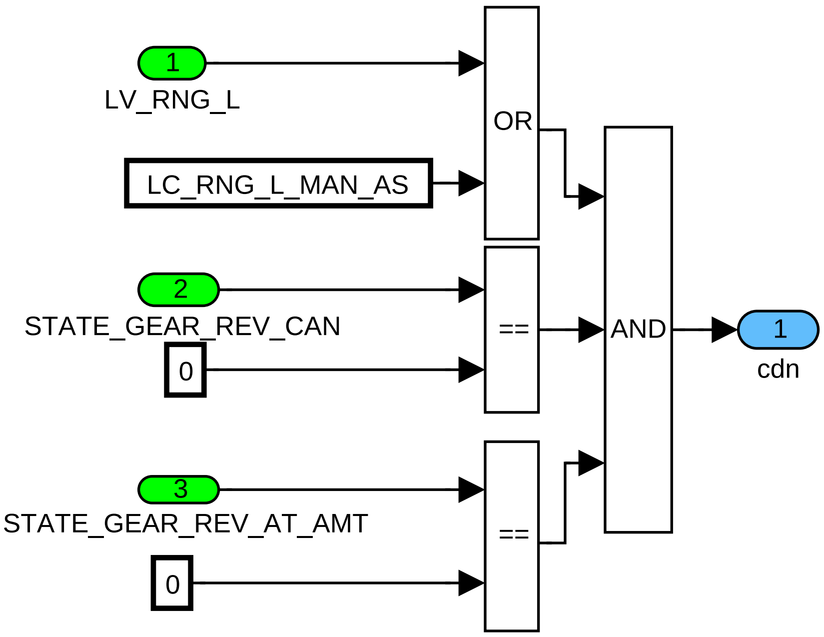


Figure 5 DRRQ_MD00P/ OPM/ OPM_10MS/ GEAR_CDN_CHK

Forward gear detection

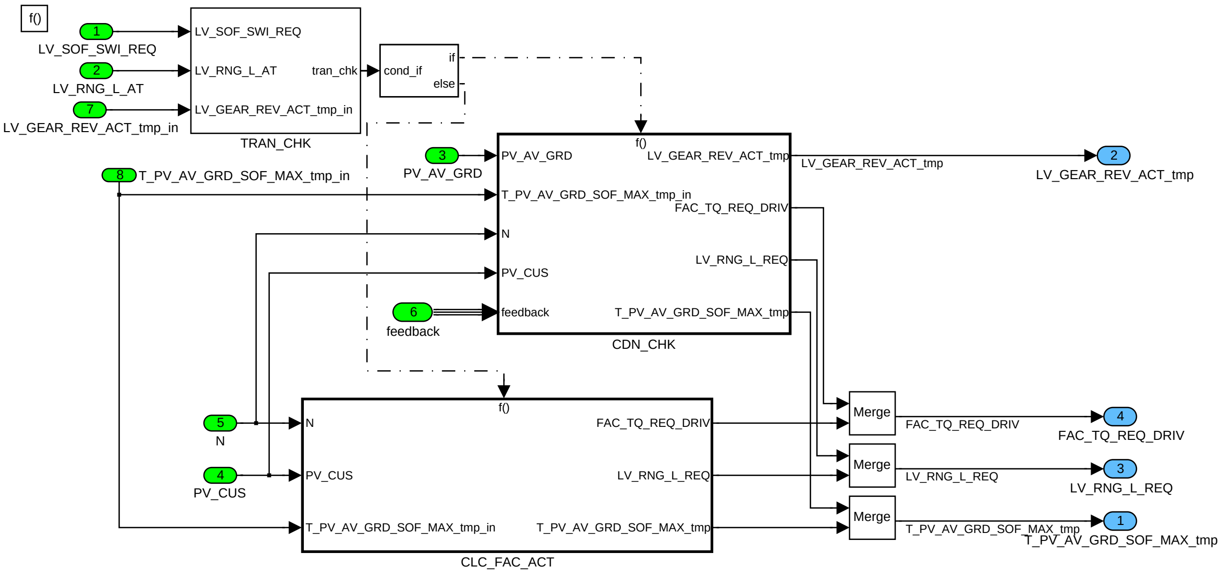


Figure 6 DRRQ_MD00P/ OPM/ OPM_10MS/ FWD_GEAR

Transition check

Transition of LV_SOF_SWI_REQ, LV_RNG_L_AT & Reverse gear to forward gear are checked.

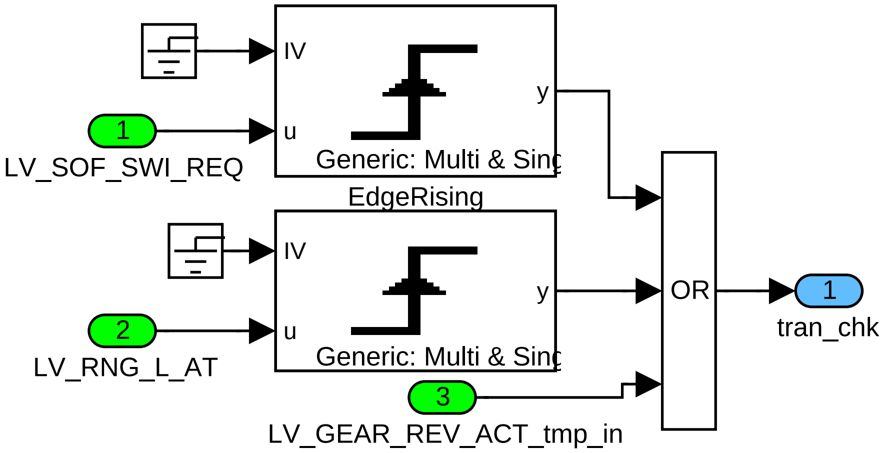


Figure 7 DRRQ_MD00P/ OPM/ OPM_10MS/ FWD_GEAR/ TRAN_CHK

Formula section

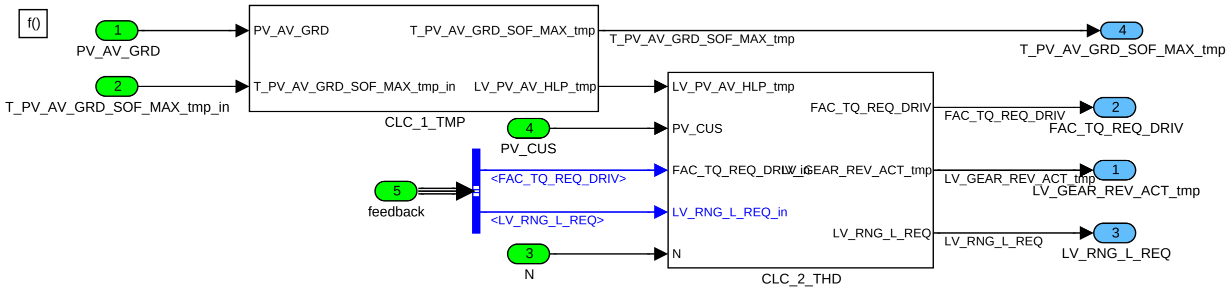


Figure 8 DRRQ_MD00P/ OPM/ OPM_10MS/ FWD_GEAR/ CDN_CHK

Timer calculation

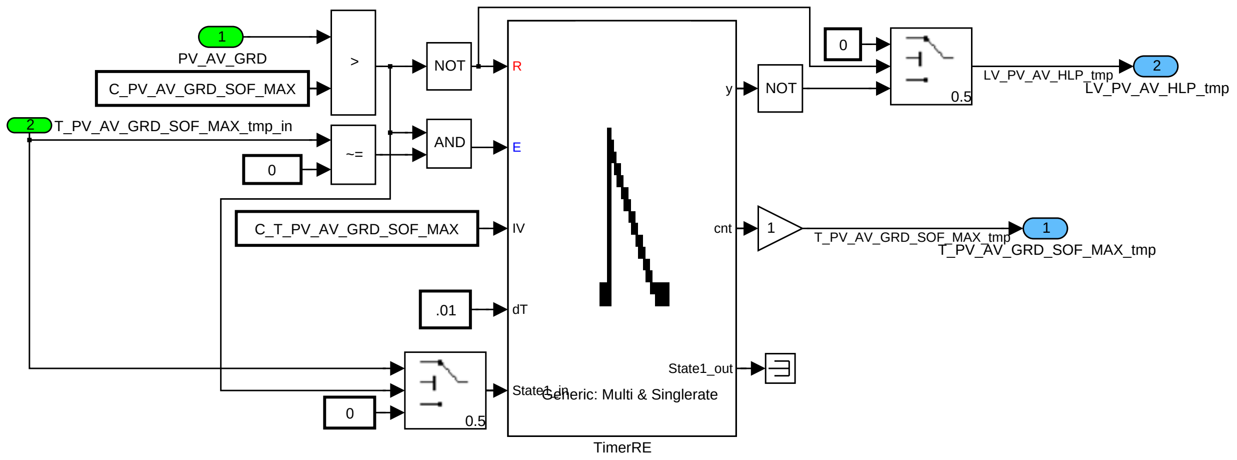


Figure 9 DRRQ_MD00P/ OPM/ OPM_10MS/ FWD_GEAR/ CDN_CHK/ CLC_1_TMP

PV_CUS threshold calculation

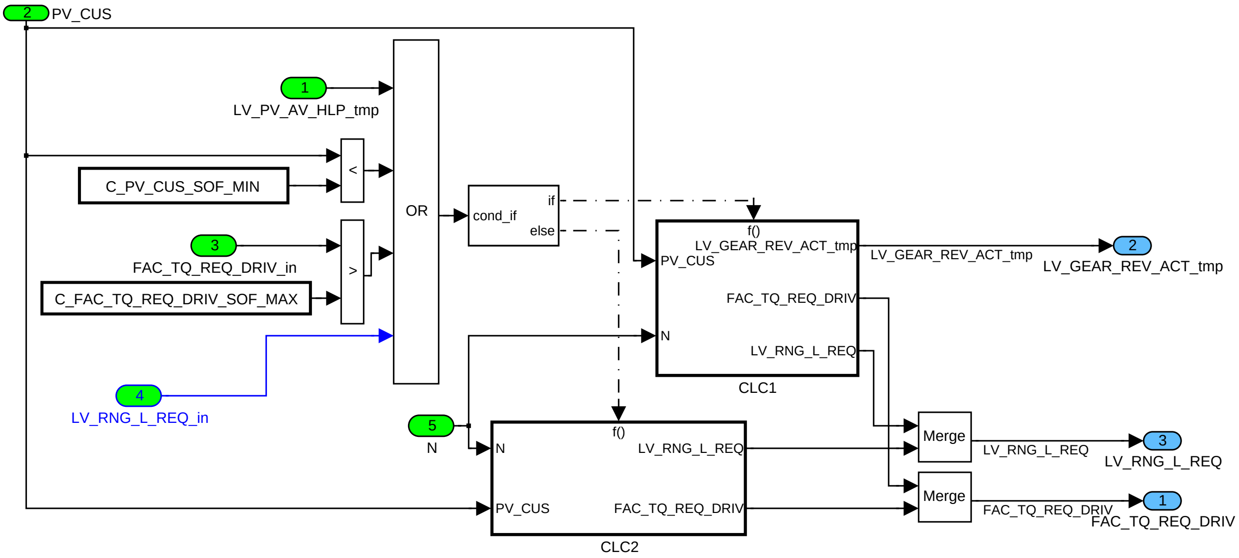


Figure 10 DRRQ_MD00P/ OPM/ OPM_10MS/ FWD_GEAR/ CDN_CHK/ CLC_2_THD

Calculation of scaling factor and activation condition

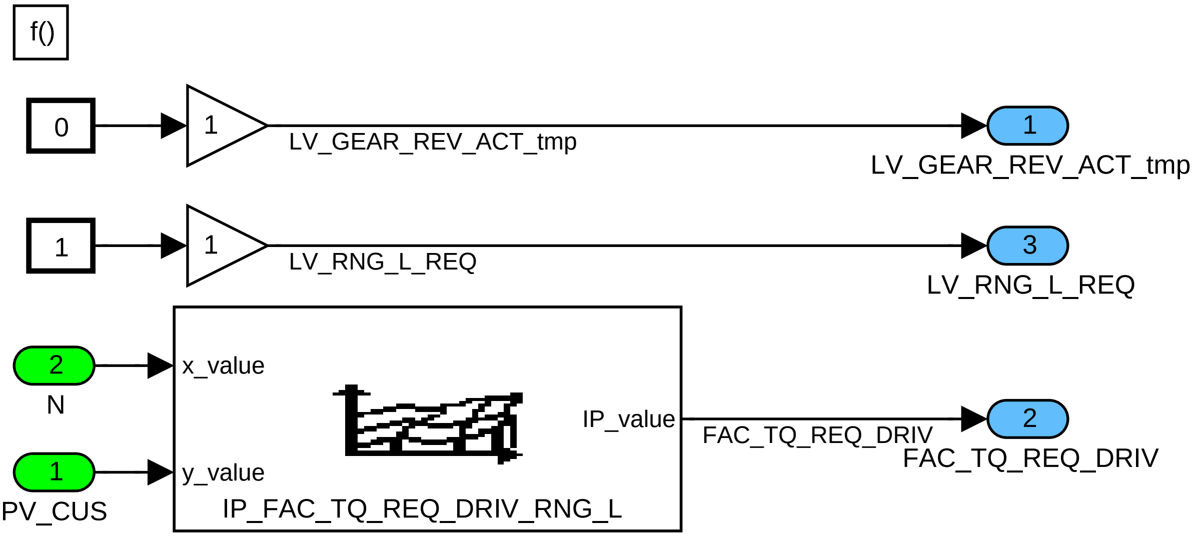


Figure 11 DRRQ_MD00P/ OPM/ OPM_10MS/ FWD_GEAR/ CDN_CHK/ CLC_2_THD/ CLC1

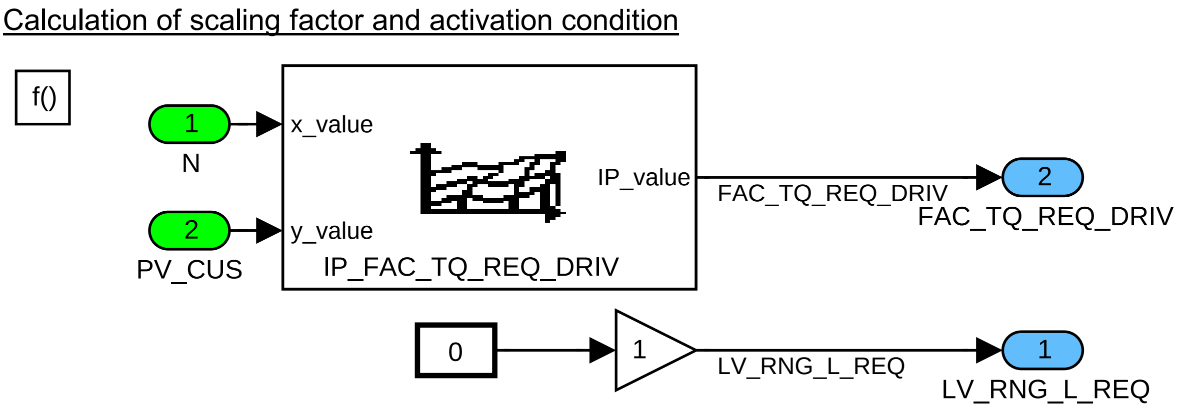


Figure 12 DRRQ_MD00P/ OPM/ OPM_10MS/ FWD_GEAR/ CDN_CHK/ CLC_2_THD/ CLC2

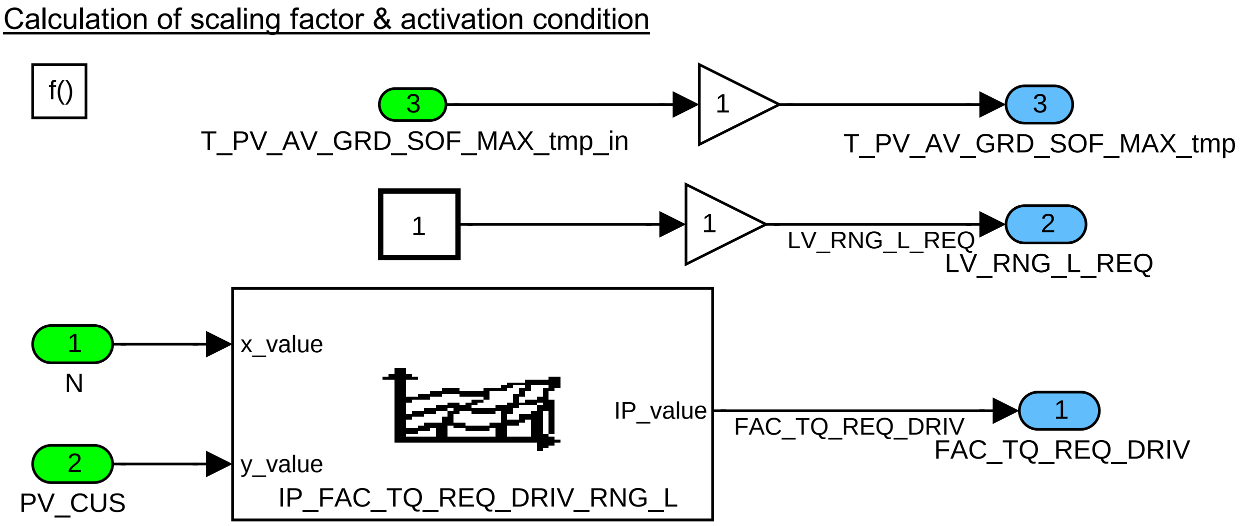


Figure 13 DRRQ_MD00P/ OPM/ OPM_10MS/ FWD_GEAR/ CLC_FAC_ACT

Reverse gear detection

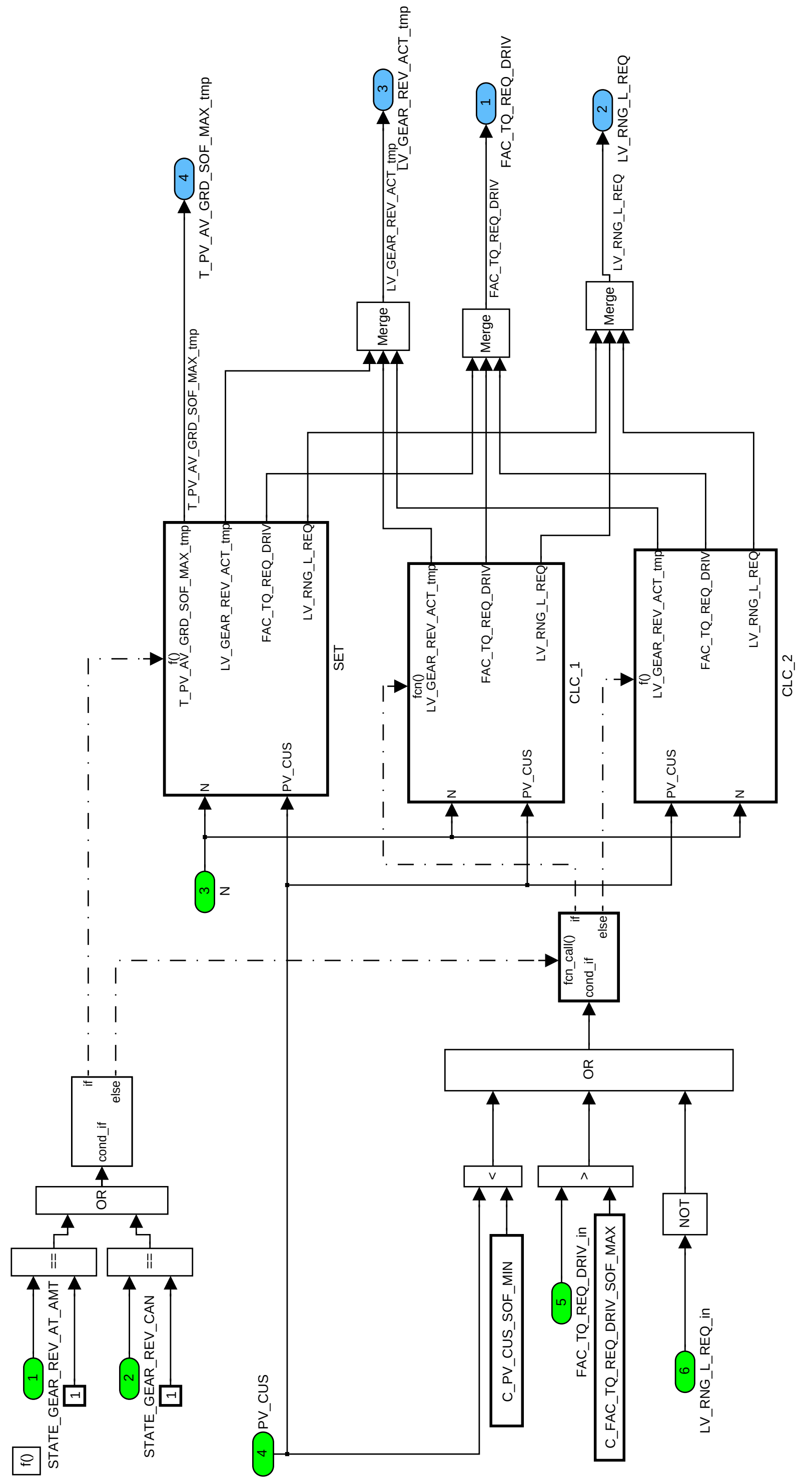


Figure 14 DRRQ_MD00P/ OPM/ OPM_10MS/ REV_GEAR_DET

Calculation of LV_GEAR_REV_ACT_tmp & T_PV_AV_GRD_SOF_MAX_tmp

Both these variables are temporary variables & used only in this module

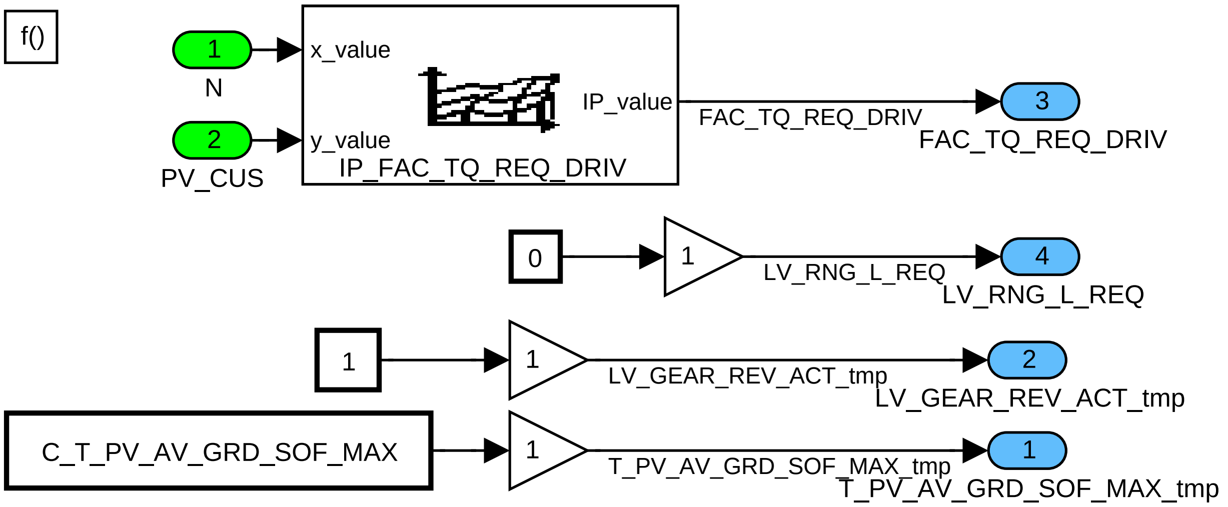


Figure 15 DRRQ_MD00P/ OPM/ OPM_10MS/ REV_GEAR_DET/ SET

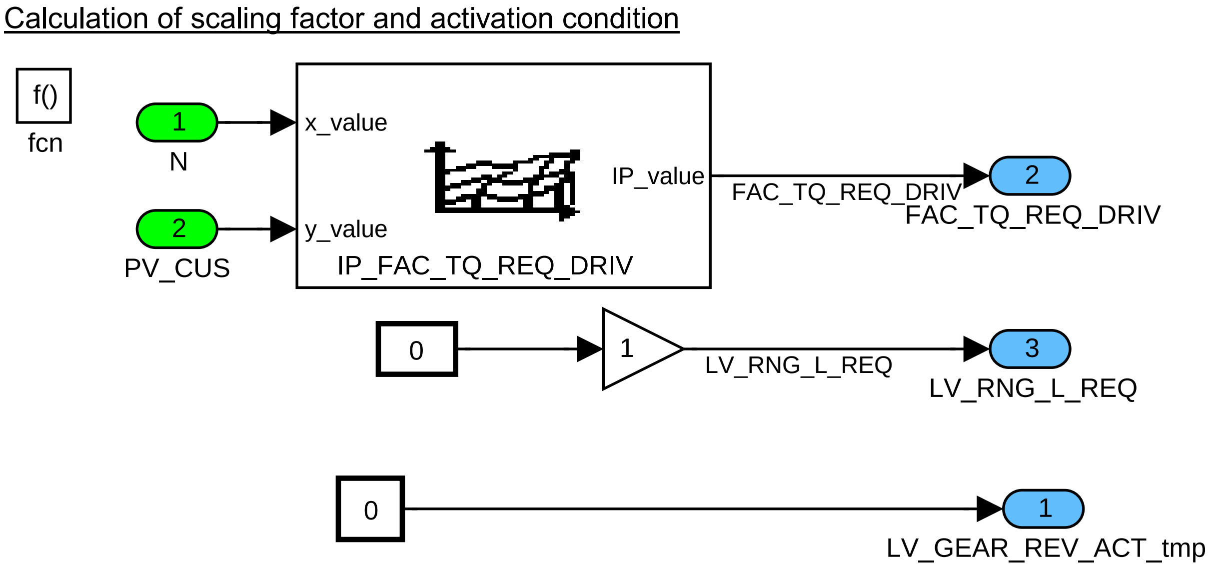


Figure 16 DRRQ_MD00P/ OPM/ OPM_10MS/ REV_GEAR_DET/ CLC_1

Calculation of scaling factor and activation condition

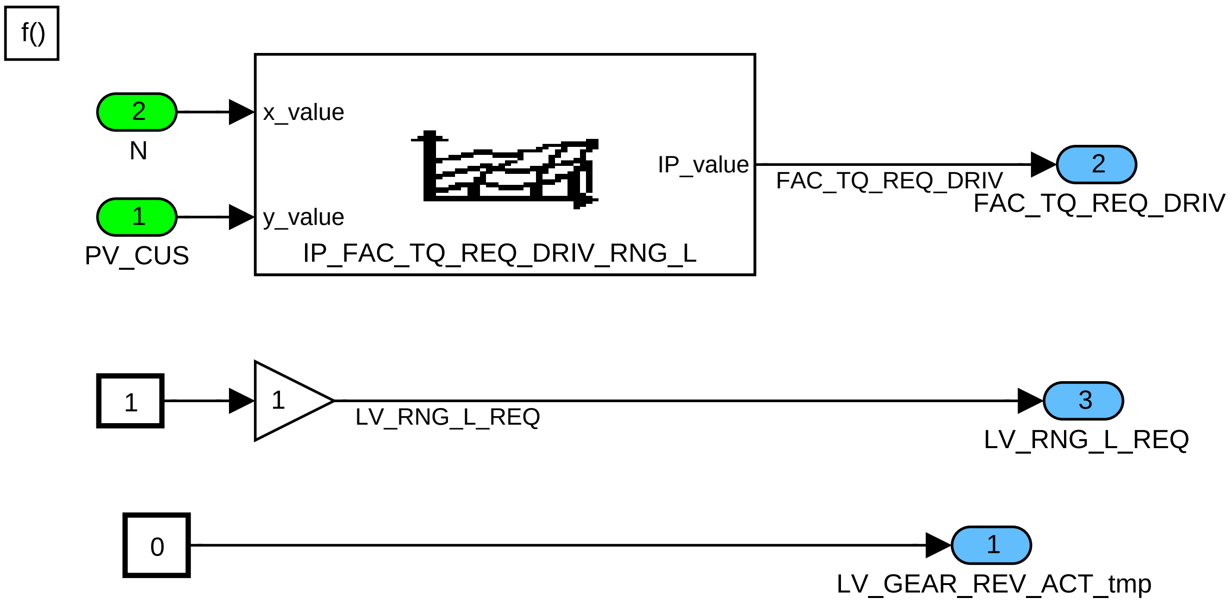


Figure 17 DRRQ_MD00P/ OPM/ OPM_10MS/ REV_GEAR_DET/ CLC_2

9.2 Customer adaptation module DRRQ: Driver Request

9.2.1 Outputs for BMW which are defined as DRRQ exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Pwg_ist	O/V	8000...7FFFH	-800...799.97558	0.0244141	[%]
Pedalwert Fahrerwunsch in %					
Pwg_ist_roh	O/V	8000...7FFFH	-800...799.97558	0.0244141	[%]
Pedalwert Fahrerwunsch in %, Rohwert					
Pwg_max	O/V	8000...7FFFH	-800...799.97558	0.0244141	[%]
Maximum selection of Pwg_ist and virtual pedal value					

Input data:

PV_CUS	LV_IGK	PV_MAX	PV_CUS_RAW

FUNCTION DESCRIPTION:

Adaption to BMW environment.

Remark: The really possible resolution and limits of the output are different to the specified values due to the input data attributes.

Application conditions:

Initialisation: 0% at reset

Recurrence: 10ms

Note:

Pwg_ist(_roh) has to be updated directly after PV_CUS(_RAW) has been calculated.

Pwg_max has to be updated directly after PV_MAX has been calculated.

Activation: *after reset*

Deactivation: *after power latch phase*

Formula section:

Remark: all formulas are valid in a **physical** meaning

If LV_IGK = 1

Then Pwg_ist = PV_CUS

Pwg_ist_roh = PV_CUS_RAW

Pwg_max = PV_MAX

Else Pwg_ist = 0%

Pwg_ist_roh = 0%

Pwg_max = 0%

9.2.2 Outputs for BMW which are not defined as DRRQ exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_bls	V/O	0...1H	0...1	1	[-]
Driver request; pedal value					

Input data:

LV_BRAKE_DET			

FUNCTION DESCRIPTION:

Adaption to BMW environment.

Application conditions:

Initialisation: -

Recurrence: 100ms

Activation: after reset (always)

Deactivation: -

Formula section:

B_bls = LV_BRAKE_DET

10 Driveability

Table of Contents

Index

10.1 Sequential fuel cut off and restart fuel feed

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SEL_CYL	V/O	0...1H	0...1	1	-
Boolean for Cylinder specific transition PU-PUC, PUC-IS or PUC-PL ongoing					
NR_PAT_SEL_CYL	V/O	0...FFH	0...255	1	-
Selected index of Fuel cut off pattern for transition PU-PUC, PUC-IS, PUC-PL					
T_FCUT_AJ	V	0...FF H	0...2.55	0.01	s
Timer value during jerk minimal sequential fuel cut-off					
T_OSC_DT	V	0...FF H	0...2.55	0.01	s
Period of time of the oscillation of the drive-train					
NR_SEL_CYL	V	1...8 H	1...8	1	-
Index number of the selective cylinder pattern					
CASE_SEL_CYL	V	0...3 H	0...3	1	-
Case selector					
TQI_SP_SLOW_DIF	V	8000...7FFFH	-1024...1023.97	0.03125	Nm
Difference in requested torque for slow path from actual to last segment					
LV_FCUT_CDN_PUC_PU_IS	V	0...1H	0...1	1	[-]
External authorization of sequential or 2-step-method at transition from PUC to PU or IS					
LV_FCUT_CDN_PUC_PL	V	0...1H	0...1	1	[-]
External authorization of sequential or 2-step-method at transition from PUC to PL					

Input data:

LV_IGK	N_DIF_COR	N_GRD	TQI_SP_SLOW
LV_PU	LV_PUC	LV_PL	LV_IS
LV_DT	LV_AT	GEAR	LV_ES
LV_ST	T_SEG_AV	STATE_ENG	

FUNCTION DESCRIPTION:

General information:

At transition from trailing throttle to trailing throttle fuel cut off, torque jumps generally occur at trailing throttle fuel cut off which are noticeable as jerks in the vehicle. To reduce the torque at this transition, the ignition timing is retarded and the combustion air is adjusted to small values.

These procedures however do not always achieve the required torque reduction, because ignition timing and combustion air can only be reduced that far, that jerk free engine operation is still possible.

To support the above mentioned torque reduction measurement, it is possible that the injection is switched on and/or switched off for individual cylinders within a defined number of engine cycles (on and/or deactivation) incase of PUC and reentry into Idle.

The engine speed gradient is monitored in order to cancel the selective reentry idle at rapid engine speed drops. If the engine speed gradient is below the threshold C_N_GRD_MIN_SEL_RIS or N_DIF_COR is greater than or equal to C_N_DIF_MAX_SEL_RIS then all cylinders are immediately activated (-> CASE 2). The same happens if during selective reentry there is a significant increase of torque request, e.g. TQI_SP_SLOW_DIF is higher than C_TQI_SP_SLOW_DIF_MAX (-> CASE 2 and 3).

10.1.1 Calculation of TQI_SP_SLOW dynamic

Description:

This value is needed for fast restart of fuel feed.

Application conditions:

Initialisation: TQI_SP_SLOW_DIF = 0

Recurrence: 20ms

Activation: at every engine state

Deactivation: -

Formula section:

TQI_SP_SLOW_DIF = TQI_SP_SLOW(n) - TQI_SP_SLOW(n-1)

10.1.2 NR_PAT_SEL_CYL calculation

Recurrence: Segment-synchronous

10.1.2.1 Calculation of T_OSC_DT:

```
if LV_AT = 0
then        T_OSC_DT = ID_T_OSC_DT(GEAR)
else        T_OSC_DT = 0
endif
```

Description (Example for 4-cylinder engines)

At a transition LV_PU → LV_PUC,. LV_PUC → LV_IS and LV_PUC → LV_PL , 32 cycles (8 x 720°CRK) in case of a 4-cylinder can be influenced. To realize this, a table ID_PAT_SEL_CYL contains the sequences of injection patterns (0...7), predefined in the table ID_PAT_IND. “ NR_SEL_CYL ” means the logical number in the shut off sequence (1...8).

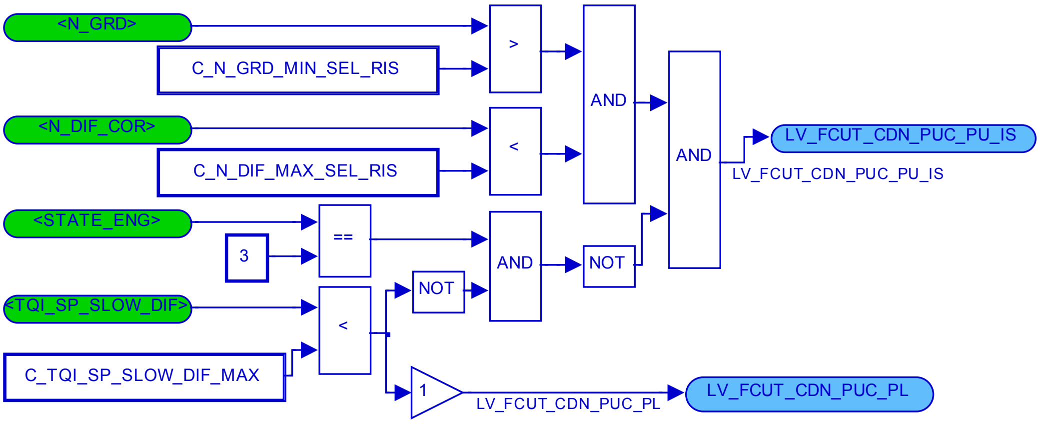
Application conditions:

Activation:

At active cylinder specific transitions LV_SEL_CYL =1

10.1.2.2 Calculation of exit conditions for running sequences (CASE 2 and CASE 3)

These bits are needed to terminate running sequences.



10.1.2.3 Transition from PU → PUC (CASE 1)

The entrance in the transition pattern for PUC follows, when all conditions for PUC are fulfilled.

```
if LC_ENA_FCUT_AJ = 0 or LV_AT = 1
        then if (LV_DT = 0 and LC_INH_DT_PU = 0)
                then    NR_SEL_CYL = 8
                        NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                else    NR_SEL_CYL starts from 1 and is incremented every 720 °KW
                        until its maximum value. In case of deactivation NR_SEL_CYL
                        is reset to 1.
        else
        if (LV_DT = 0 and LC_INH_DT_PU = 0)
        then    NR_SEL_CYL = 8
                NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                T_FCUT_AJ is initialized with T_OSC_DT / 2
        else    NR_SEL_CYL = C_NR_SEL_CYL_1
                NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                LV_SEL_CYL = 1
                if T_FCUT_AJ - T_SEG_AV = 0
                then    NR_SEL_CYL = 8
                        NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                        T_FCUT_AJ is initialized with T_OSC_DT / 2
                        LV_SEL_CYL = 0
                else    $NR\_SEL\_CYL_n = NR\_SEL\_CYL_{n-1}$
                T_FCUT_AJ is decremented
                end
        end
end
```

The transition is completed, when the last of the 32 transition cycles has run out. The pattern index applied as last one has to be the highest used for the cut-off strategy , that means all cylinders are cut off. NR_SEL_CYL is reset to 1.

If in the course of the transition pattern or while the time sequence an engine operating state change to PU, PL or IS happens, all cylinders are switched on immediatedly (pattern index : “0“), NR_SEL_CYL is reset to 1.

10.1.2.4 Transition from PUC → IS or PUC → PU (CASE 2)

The entrance in the transition pattern for RIS (reactivation idle speed) follows, when all conditions for IS resp. PU are fulfilled.

```
if LC_ENA_FCUT_AJ = 0 or LV_AT = 1
then    if (LV_DT = 0 and LC_INH_DT_PU = 0) or LV_FCUT_CDN_PUC_PU_IS = 0
        then    NR_SEL_CYL = 8
                NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
        else    NR_SEL_CYL starts from 1 and is incremented every 720 °KW
                until its maximum value.

else
        if (LV_DT = 0 and LC_INH_DT_PU = 0) or LV_FCUT_CDN_PUC_PU_IS = 0
        then    NR_SEL_CYL = 8
                NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                T_FCUT_AJ is initialized with T_OSC_DT / 2
        else    NR_SEL_CYL = C_NR_SEL_CYL_2
                NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                LV_SEL_CYL = 1
                if T_FCUT_AJ - T_SEG_AV = 0
                then    NR_SEL_CYL = 8
                        NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                        T_FCUT_AJ is initialized with T_OSC_DT / 2
                        LV_SEL_CYL = 0
                else    NR_SEL_CYL$_n$ = NR_SEL_CYL$_{n-1}$
                        T_FCUT_AJ is decremented
                end
        end
end
```

The transition pattern is completed, when the last cycle of the 32 transition cycles has run out. The pattern number applied as last one has to be „0“ , that means all cylinders are switched on. NR_SEL_CYL is reset to 1.

If in the course of the transition pattern or while time sequence a engine operating state change to PUC happens, the transition pattern or time sequence for RIS runs out to its end and after that the transition pattern for PUC starts, if required.

If the function is deactivated (see deactivation conditions), all cylinders are switched on immediatedly. In case of deactivation NR_SEL_CYL is reset to 1.

10.1.2.5 Transition from PUC → PL (CASE 3)

The entrance in the transition pattern for RPL (reactivation part load) follows, when all conditions for PL are fulfilled.

```
if LC_ENA_FCUT_AJ = 0 or LV_AT = 1
then    if (LV_DT = 0 and LC_INH_DT_PU = 0) or LV_FCUT_CDN_PUC_PL = 0
        then    NR_SEL_CYL = 8
                NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
        else    NR_SEL_CYL starts from 1 and is incremented every 720 °KW
                until its maximum value. In case of deactivation NR_SEL_CYL
                is reset to 1.


else
        if (LV_DT = 0 and LC_INH_DT_PU = 0) or LV_FCUT_CDN_PUC_PL = 0
        then    NR_SEL_CYL = 8
                NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                T_FCUT_AJ is initialized with T_OSC_DT / 2
        else    NR_SEL_CYL = C_NR_SEL_CYL_3
                NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                LV_SEL_CYL = 1
                if T_FCUT_AJ - T_SEG_AV = 0
                then    NR_SEL_CYL = 8
                        NR_PAT_SEL_CYL = ID_PAT_SEL_CYL(NR_SEL_CYL)
                        T_FCUT_AJ is initialized with T_OSC_DT / 2
                        LV_SEL_CYL = 0
                else    $NR\_SEL\_CYL_n = NR\_SEL\_CYL_{n-1}$
                        T_FCUT_AJ is decremented
                end
        end
end
```

The transition pattern is completed, when the last cycle of the 32 transition cycles has run out. The pattern number applied as last one has to be „0“ , that means all cylinders are switched on. NR_SEL_CYL is reset to 1.

If in the course of the transition pattern a engine operating state change to PUC happens, the transition pattern for reentry PL runs out to its end and after that the transition pattern for PUC starts, if required. If the function is deactivated, all cylinders are switched on immediatedly.

Deactivation:

- LV_ES = 1 **or**
- LV_ST = 1 **or**
- LV_IGK = 0 **or**
- Pattern completed

At passive cylinder specific transitions:

LV_SEL_CYL = 0 and NR_SEL_CYL = 1

T_FCUT_AJ is initialized with T_OSC_DT / 2

Transition tables

The transition pattern indexes are adjustable in the table ID_PAT_SEL_CYL. There are 8 fade out patterns which are applicable per transition table (definition of patterns see chapter 7, “Cylinder shut off with pattern“).

The following cases are distinguished in ID_PAT_SEL_CYL:

CASE 1: Trailing throttle fuel cut off (PUC)
CASE 2: Restart IS (RIS)
CASE 3: Restart PL (RPL)

Example for shut-off sequences with 8 levels and a shut off pattern for the 4 next injections (4-cylinder)

CASE_SEL_CYL↓ NR_SEL_CYL →

	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***
1	1	1	2	2	3	3	4	4
2	5	5	5	6	6	7	7	0
3	5	6	6	7	0	0	0	0

According shut off sequences: (0 : Injection, 1: no injection)

Case 1 1 0 0 0 1 0 0 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1

Case 2 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0

Case 3 0 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Formula section:

NR_PAT_SEL_CYL = ID_PAT_SEL_CYL

Application assistances

Handling of reactivation functions (out of PUC):

Injection (Reactivation fuel feed) : To ensure, that all cylinders get at least once the same amount of TI_ADD_REAC, TI_ADD_REAC is not decremented as long as sequential transition PUC → IS resp. PUC → PL is ongoing.

Ignition: The according reactivation functions are not concerned by sequential transition functions.

Handling of engine states:

The detection of PU, PUC, PL and IS is not concerned by sequential transition functions.

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
ID_PAT_SEL_CYL	3*8	0...FFH	0...255	1	-
LDP_CASE_SEL_CYL_ID_PAT_SEL_CYL	3	1...3H	1...3	1	-
LDP_NR_SEL_CYL_ ID_PAT_SEL_CYL	8	1...8H	1...8	1	-
Table for cylinder specific restart and trailing throttle fuel cut off					
C_N_GRD_MIN_SEL_RIS	1	80H...7FH	-4096...4064	32	1/min/s
N_GRD threshold for cancelling cylinder individual cut-off for restart IS					
C_N_DIF_MAX_SEL_RIS	1	8000H..7FFFH	-32768..32767	1	rpm
N_DIF_COR threshold for deactivation of cylinder individual cut-off for restart IS					
ID_T_OSC_DT	8	0...FF H	0 ... 2.55	0.01	s
LDP_GEAR_1_DRVB	8	0...FF H	0 ... 255	1	-
Characteristic period of time of the eigenfrequency of the powertrain					
LC_INH_DT_PU	1	0 , 1 H	0 , 1	1	-
Logical variable to inhibit LV_DT impact on jerk minimal sequential fuel cut-off activation					
LC_ENA_FCUT_AJ	1	0 , 1 H	0 , 1	1	-
Logical variable to enable jerk minimal sequential fuel cut-off					
C_NR_SEL_CYL_1	1	1...8 H	1...8	1	-
Start pattern index for transition from PU -> PUC (case 1)					
C_NR_SEL_CYL_2	1	1...8 H	1...8	1	-
Start pattern index for transition from PUC -> IS, PU (case 2)					
C_NR_SEL_CYL_3	1	1...8 H	1...8	1	-
Start pattern index for transition from PUC -> PL (case 3)					
C_TQI_SP_SLOW_DIF_MAX	1	8000...7FFFH	-1024...1023.97	0.03125	Nm
TQI_SP_SLOW_DIF threshold for cancelling cylinder individual cut-off for restart in PL					

10.2 Customer adaptation module DRVB: Driveability

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQI_REQ_TRA	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Indicated torque request after transient torque function					
LV_ACT_AJ	V/O	0...1H	0...1	1	[-]
LV anti-jerk function active					

Input data:

B_ar_akt	Mdk_wunsch_filt	Mdi_reib	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: at reset and at each activation and deactivation:
all outputs are initialised with 0

Recurrence : 10 ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

Remark: all formulas are valid in a **physical** meaning

LV_ACT_AJ = B_ar_akt

TQI_REQ_TRA = Mdk_wunsch_filt – Mdi_reib

11 ECU Process Monitoring

Table of Contents

Index

11.1 ECM2 - Requirements to Infrastructure

Export actions:

ACTION_INFR_GetVPVSMON(OUT <V_pvs_1_mon>, OUT<V_pvs_2_mon>)
Readout both PVS channels
ACTION_INFR_GetVTPSMON(OUT <V_tps_1_mon>, OUT<V_tps_2_mon>)
Readout both TPS channels
ACTION_INFR_GetTSEGSWMON(OUT <T_seg_sw_mon>)
Readout segment time from Infrastructure timer
ACTION_INFR_ResetTSEGSWMON()
Reset segment time from Infrastructure timer

Description for actions:

ACTION_INFR_GetVPVSMON(V_pvs_1_mon, V_pvs_2_mon)					
Readout both PVS channels with a defined coherency.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
V_pvs_1_mon	OUT	0 ... 3FFH	0 ... 4.9952	5/1024	V
Channel 1 of PVS for monitoring					
V_pvs_2_mon	OUT	0 ... 3FFH	0 ... 4.9952	5/1024	V
Channel 2 of PVS for monitoring					

ACTION_INFR_GetVTPSMON(V_tps_1_mon, V_tps_2_mon)					
Readout both TPS channels with a defined coherency.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
V_tps_1_mon	OUT	0 ... 3FFH	0 ... 4.9952	5/1024	V
Channel 1 of TPS for monitoring					
V_tps_2_mon	OUT	0 ... 3FFH	0 ... 4.9952	5/1024	V
Channel 2 of TPS for monitoring					

ACTION_INFR_GetTSEGSWMON(T_seg_sw_mon)					
Readout segment time from Infrastructure timer					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
T_seg_sw_mon	OUT	0 ... FFFFFFFH	0 ... 268435455	1	µs
segment time determined by means of an infrastructure timer					

ACTION_INFR_ResetTSEGSWMON()
Reset segment time from Infrastructure timer

FUNCTION DESCRIPTION:

General information:

ACTION_INFR_GetVPVSMON reads out the two PVS channels of an analogue pedal value sensor with a defined coherency/coincidence and diversity level.
To define the scope of the *Process Monitoring* as wide as possible, the voltages shall be a direct output of the ADCs. As far as possible, the used program syntax must prevent the action from returning consistent wrong information for both channels. This means for wrong PVS**:** Signals are too high and V_pvs_1_mon would be double of V_pvs_2_mon.

ACTION_INFR_GetVTPSMON reads out the two TPS channels of an analogue throttle position sensor with a defined coherency/coincidence diversity level.
To define the scope of the *Process Monitoring* as wide as possible, the voltages shall be a direct output of the ADCs. As far as possible, the used program syntax must prevent the action from returning consistent wrong information for both channels. This means for wrong TPS: Signal V_tps_2_mon would be Vref - V_tps_1_mon.

ACTION_INFR_GetTSEGSWMON reads out the segment time. The segment time is generated by a timer solution in the infrastructure and shall be independent of the level-1-timer- solution in ENSD part (Engine position and speed determination – aggregate). Reason is that the segment time T_SEG_SW_MON is necessary to generate an engine speed signal that is independent from N_32 (generated from T_SEG). It is used in the module ‚Monitoring of engine speed‘ of the aggregate ECM2.

The timer is triggered by crankshaft segments and produces time stamps for the trigger events. The maximum delay between event and time stamp, i.e. the temporal difference of determination/calculation of T_SEG and T_SEG_SW_MON, must be less than 300 µs (split into an invariable part of 50 µs and a maximum variable part of 250µs). From the time stamps the signal T_SEG_SW_MON is determined; its resolution is 1 µs.
The precision of the signal, i.e. the maximum allowed difference in value of the segment time level 2 to level 1, has to be 5 % of the actual segment time T_SEG.

Furthermore, after first initialisation of the segment time T_SEG_SW_MON while power on of ECU, its value has to be re-initialized with the calculated/extrapolated value of T_SEG of level 1 once more for only one time. Both signals shall be equal at first calculation of T_SEG. After that, they shall always be separately determined.

The segment trigger used of I/O-level is called *SIG_SegTrig*.

ACTION_INFR_ResetTSEGSWMON reset the segment time to the maximum value.

Since for the event <Engine Stop> the engine speed N_32 is set to zero in the function level (and T_SEG is set to its maximum value, respectively), the segment time T_SEG_SW_MON has to be set to its maximum value for <Engine Stop> as well. This action therefore resets T_SEG_SW_MON.

However, this action must not be triggered by using the variable LV_ES=0, but must be initialized within the application layer when recognizing the transition <Engine Run to Engine Stop> by the engine operation states manager.

Requirements for ACTION_INFR_GetVPVSMON:

Data acquisition

Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment
V_pvs_1_mon	Not relevant	1% of reference supply voltage for sensor	10bit	The time duration between the digitalization of both analog voltages must be less than 1.0 ms	The precisions and resolutions of V_pvs_2_mon must match the one of V_pvs_1_mon
V_pvs_2_mon	- see V_pvs_1_mon -				

Diagnosis: no electrical diagnosis done here

Coincidence requirements: -

Requirements for ACTION_INFR_GetVTPSMON:

Data acquisition

Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment
V_tps_1_mon	Not relevant	1% of reference supply voltage for sensor	10bit	The time duration between the digitalization of both analog voltages must be less than 1.0 ms	The precisions and resolutions of V_tps_2_mon must match the one of V_tps_1_mon
V_tps_2_mon	- see V_tps_1_mon -				

Diagnosis: no electrical diagnosis done here

Coincidence requirements: -

Requirements for ACTION_INFR_GetTSEGSWMON:

Data acquisition

Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment
T_seg_sw_mon	--	5 % of actual segment time **the relative precision must be guaranteed**	1 µs	The maximum delay between event and time stamp must be less than 300 µs	T_seg_sw_mon has to be set to its maximum by the BSW while having lack of synchronisation **OR** at first initialization after power on. Re-Initialisation of T_seg_sw_mon for <u>only one time</u> simultaneously with first computation/calculation of T_seg after power on.

Diagnosis: no electrical diagnosis done here

Coincidence requirements: -

Requirements for ACTION_INFR_ResetTSEGSWMON:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
--	--	--	--	--	T_seg_sw_mon has to be set to its maximum value (FFFFFFFFH) by the BSW. This action is called within the application layer if recognizing the event <Engine Run to Engine Stop> or the status "RESET".

Diagnosis: no electrical diagnosis done here

Coincidence requirements: -

11.2 Reset independant ETC- Limp Home

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_N_LIM_REQ_RST_CHK	V/O	0...1H	0...1	1	[-]
Logical variable to set engine speed limitation after ECU reset					
LIH_ACT	V	0...FFFFH	0...65535	1	[-]
0H = kein E-GAS-Notlauf / AF50H = E-Gas-Notlauf aktiv (gespeichert im resetfesten Bereich)					

Input data:

LV_IGK	LV_TPS_MTC_N_LIM	LV_N_LIM_REQ_MON	
LV_MTC_LIH_ACT	LV_ERR_TQI_N_MAX_MON		

FUNCTION DESCRIPTION:

General information:

If ETC-Limp home is active, performance is limited for the driver. After a reset of the ECU this limitation of performance is lost from one second to another. To prevent this dangerous situation the word LIH_ACT is saved to reset independant memory every 100 ms. This special part of memory is not deleted in case of warm reset (LV_WARM_RST = 1). In case of cold reset (LV_WARM_RST = 0) even information in this reset resistant memory is lost.

While initializing the ECU the word LIH_ACT is been read out of the reset resistant memory. If it is set, LV_N_LIM_REQ_RST_CHK is set to 1 for the current driving cycle. At transition LV_IGK = 1 → 0 or if the failure memory is been deleted, LV_N_LIM_REQ_RST_CHK and LIH_ACT are reset to 0.

Application Conditions:

Recurrence: 100ms

Initialisation:

at reset (but not in case of warm-reset LV_WARM_RST = 1) **OR**
at transition LV_IGK = 1 → 0 **OR**
after clearing failure memory

LV_N_LIM_REQ_RST_CHK = 0
LIH_ACT = 0

The following initialization is only used after warm-reset (LV_WARM_RST = 1):

```
IF        after warm-reset   LIH_ACT = AF50 hex (44880 dec)
          (read after warm-reset from reset resistant memory)

THEN      LV_N_LIM_REQ_RST_CHK = 1
          (to be written only once while initializing ECU)

ELSE      LV_N_LIM_REQ_RST_CHK = 0
          (to be written only once while initializing ECU)

ENDIF.
```

Activation:

LV_IGK = 1

Deactivation:

LV_IGK = 0

Formula section:

```
IF        LV_TPS_MTC_N_LIM = 1          OR    // ETC-LIH request
          LV_MTC_LIH_ACT = 1            OR    // PVS-LIH request
          LV_N_LIM_REQ_MON = 1          OR    // MON-LIH request
          LV_ERR_TQI_N_MAX_MON = 1      OR
          LV_N_LIM_REQ_RST_CHK = 1

THEN      LIH_ACT = AF50 hex (44880 dec)
          (to be written to reset resistant memory every 100ms)

ELSE      LIH_ACT = 0 (to be written to reset resistant memory every 100ms)

ENDIF.
```

11.3 General

11.3.1 Basic Idea

The Monitoring Concept has to prevent an ETC system from dangerous behaviour like sudden acceleration against the driver's intent. At first faults, this must be without intervention of the driver. To avoid immobile cars at first faults, a limited limp home function is activated. This means the dynamic behaviour of the car is limited, and so in case of another (second) fault, the driver must possibly react (activation of brake pedal). The accepted fault reaction time is 500 ms. Minor deviations in the order of about 5% are possible due to run time fluctuations, aging effects and temperature influences. No or low engine power is seen as a safe state.

The Siemens ETC-Monitoring-Concept is based on the ETC-Safety-Concept of VDA Arbeitskreis E-Gas using the following main modification:

Additional, diverse AD-converter in monitoring unit (MU) for monitoring af A/D conversion. The used analog voltage must be non-constant, the frequency of the voltage changes must match to the synchronisation constraints given in the Processor Monitoring.

The monitoring concept consists of three levels, as shown in the following figure:

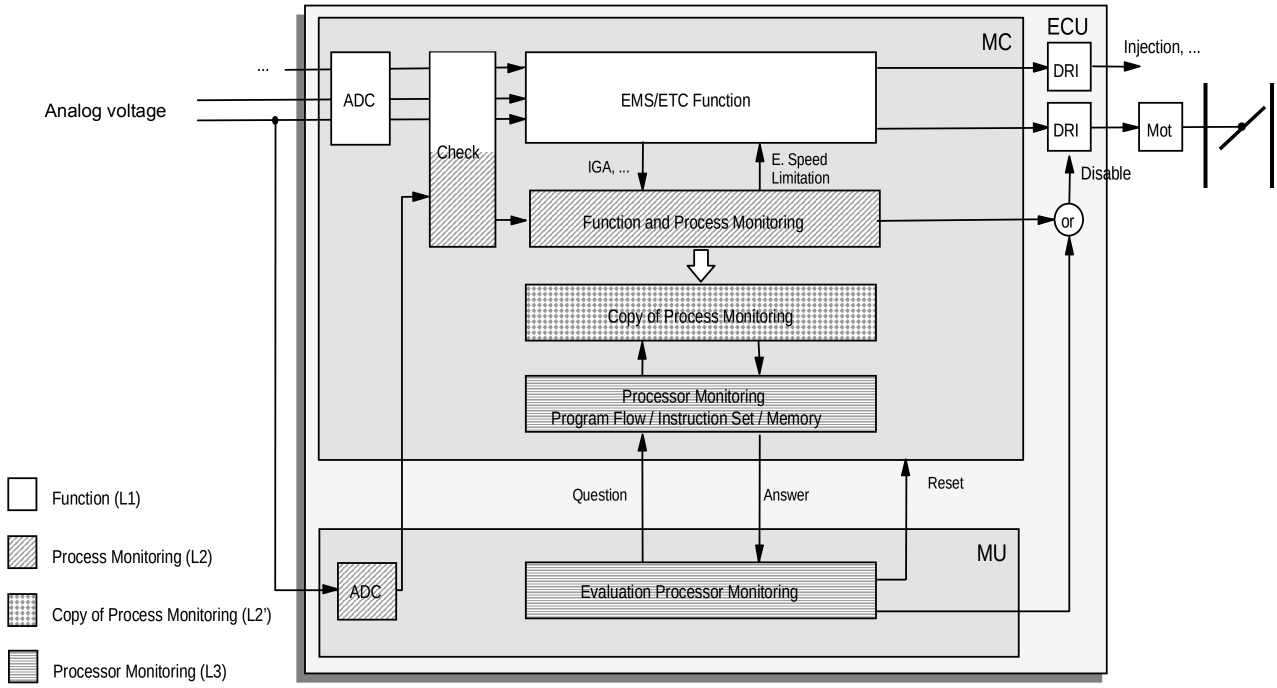


Figure 1: The three levels of the monitoring concept.

- The function level is located on the main processor (MC).
- The function/process monitoring level is also located on the main processor.
- The processor monitoring level is located on the main processor and on the monitoring processor (MU).

11.3.1.1 Function Level (VDA-name: Level 1)

The function level performs standard EMS functions including ETC functionality (description *not* to be found in this chapter):

- Determination of the drivers demand (via PVS) including standard diagnosis (signal range check, ...)
- Determination of throttle position (via TPS) including standard diagnosis (signal range check, ...)
- Receiving external torque requests (e.g. ASR, MSR, ...) including standard diagnosis (e.g. monitoring CAN by parity bit, ...). External modules must generate correct torque requests (safety requirement!) the ECU can only check, whether the transmission of this request was without faults.
- Internal calculation of torque requests (ISC, Cruise Control)
- Co-ordination and realisation of requested torque

\- output: throttle position, ignition angle, injection time and cylinder cut off

11.3.1.2 Process Monitoring Level (VDA-name: Level 2)

The process monitoring performs monitoring of sensors, actuators, torque effective functions, torque co-ordination, and torque realization as a check of the function level.

The description can be found in the modules:

- Process Monitoring
- Monitoring Sensor Signals (e.g. PVS, TCO, ...)
- Monitoring Internal Signals (e.g. MAF, N_32, ...)
- Monitoring CAN Signals
- Torque Monitoring
- Fault Reaction

11.3.1.3 Processor Monitoring Level (VDA-name: Level 3)

The processor monitoring monitors proper processor function: memory, operations of the processor, program flow, analog digital conversion

The description can be found in the modules:

- Processor Monitoring
- Communication between MC and MU
- Fault Reaction

There are additional port extensions located on the monitoring unit. Note that this functionality is not a core functionality of level 3:

- Input / Output (Description of port extension facility of the monitoring unit)

11.3.2 Hardware Safety Module

The main-processor (MC) and the monitoring processor (MU, incl. port extensions) are connected via an internal communication. MU has its own oscillator (ceramic resonator).

When resetting, MC gives a reset signal to MU so it will also reset. This enables synchronisation between them.

Via the internal communication informations are excanged. This informations belongs to: test calculations (level 2'), program flow of MC, monitoring of ADC, port extensions.

For the purpose of test calculations answers are compared to known results on MU. In case of faults, the power stages of the throttle and the injectors are disabled.

One analog voltage is connected with the first AD-channel of MU and the dedicated channel on MC. The MU value is transmitted to MC only to check the ADCs. Remaining AD-channels and digital outputs can be used as port extension (no monitoring function, but it has to be assured, that no damaging voltage is fed to the ports in case of any fault).

11.4 PROCESS MONITORING

Import actions:

ACTION_ECM3_ServicePfm(IN <>)

Note: This action is defined in chapter "Processor Monitoring", subsection "Program Flow Monitoring services for Process Monitoring". This action has to be executed as last instruction after all tasks have been executed. After the first block of tasks, the action has to be called with argument 1, after the second with 2, after the third with 3 and after the fourth with 4. At the very end of the scheduler finally the action has to be called with argument 0, 6, and 7.

ACTION_ECM2_LockPws()
ACTION_ECM2_UnlockPws()

Note: These two actions are defined in module "Fault Reaction of process monitoring".

Input data:

LV_RLY_MAIN_DLY_ERR	LV_DIAG_END_RLY_MAIN_DLY		

General information:

The task of process monitoring is to check whether ETC / torque functions (located in function level) work properly and to take a fault reaction, if not. This is described in the following modules:

- Limitation of Negative Temperature Gradient

Engine Losses depend on engine temperature (indicated by TOIL or TCO). Cold temperature means big losses. These losses are compensated by the torque structure. This means faults like a temperature step from hot to cold within permitted signal range lead to miscompensation of losses (undesired torque increase). To avoid this, the max. negative temperature gradient is limited to (small) values, given by physics.

- Monitoring Sensor Signals

Sensor Signals (PVS, TPS) are checked a second time in process monitoring. This is done by using bigger tolerances and longer debouncing times. Having detected a fault in process monitoring, but not in function level, means a faulty function level. Fault reaction is done as described below. A fault, detected in function level earlier than in process monitoring or at the same time, means normal working of function level, therefore standard reaction of function level is done.

- Monitoring CAN Signals

CAN signals are only checked against protocol (parity bit, message counter, ...). No plausibilisation of values is possible. CAN monitoring performed in process monitoring is a subset of CAN monitoring performed in function level.

- Monitoring Internal Signals

Internal signals (engine speed, torque demand of idle speed controller, torque demand of Cruise Control, cylinder-cut-off efficiency, ...) are partly moved from function level to process monitoring as value and its complement (engine speed); partly a plausibility check of the values is done (Cruise Control request) or the values are computed a second time (idle speed controller). Fault reaction is done as described below.

- Torque Monitoring

The desired maximum indicated engine torque is calculated from individual torque requests and compared to the actual indicated engine torque, calculated from MFF, engine speed, ignition angle- and other efficiencies. If the actual torque exceeds the desired torque by more than a certain threshold, a fault reaction is done as described below.

- Fault Reaction

Standard fault reaction of process monitoring is disabling of throttle actuator and engine speed limitation, commanded to function level. As function level is detected to be faulty, the commanded limitation is monitored and if it is not realized properly, a command is given to send wrong answers to the monitoring processor. This means, the monitoring processor will cut off throttle actuator and injector valves directly.

In case of faults detected when transmitting values and their complement (fault in memory or in instruction set), powerstages are disabled and a reset is performed.

Formula Section:

```
IF        LV_DIAG_END_RLY_MAIN_DLY == 1  AND   LV_RLY_MAIN_DLY_ERR == 0
THEN
      ACTION_ECM2_UnlockPws();
      /* Has to be executed exactly once!  */
      Tasks/ECM2-functions activated according to table 1 starting with task "1";
ELSE      ACTION_ECM2_LockPws();
ENDIF
```

11.4.1 Task scheduler

The process monitoring functionality is executed with an update rate of 40 ms. In order to achieve a balanced computational load of the controller, the functionality of the process monitoring is divided into 4 tasks (see following Table 1). For each task the first called module is the related part of Application Incidences of process monitoring.

Table 1: Distribution of the process monitoring functionality over four tasks. The task scheduler is executed every 10 ms. Within one task, the different functions are executed in the order according to the rows of the table.

Task 1 (k)	Task 2 (k+10 ms)	Task 3 (k+20 ms)	Task 4 (k+30 ms)
Application incidences of process monitoring(k)	Application incidences of process monitoring(k+10ms)	Application incidences of process monitoring(k+20ms)	Application incidences of process monitoring(k+30ms)
Monitoring of A/D-conversion	Monitoring of pedal value signals	Monitoring of engine speed limitation	Actual indicated engine torque
Monitoring of TCO		Monitoring of CAN signals	Desired indicated engine torque
Monitoring of engine speed	Monitoring of torque losses	Monitoring of cruise control conditions	Monitoring of actual indicated engine torque
Monitoring of idle speed controller	Monitoring of maximum torque at clutch	Monitoring of minimum torque at clutch	Fault reaction of process monitoring
Actual Fuel Mass Flow	Actual lambda deviation	Actual efficiencies	Combustion mode switch request
ACTION_ECM3_ ServicePfm(1)	Monitoring of Mass Fuel Flow		ACTION_ECM3_ ServicePfm(4)
ACTION_ECM3_ ServicePfm(0)	ACTION_ECM3_ ServicePfm(2)	ACTION_ECM3_ ServicePfm(3)	Error memory management of process monitoring
	ACTION_ECM3_ ServicePfm(0)	ACTION_ECM3_ ServicePfm(0)	ACTION_ECM3_ ServicePfm(0)

11.5 Debounce Mechanism

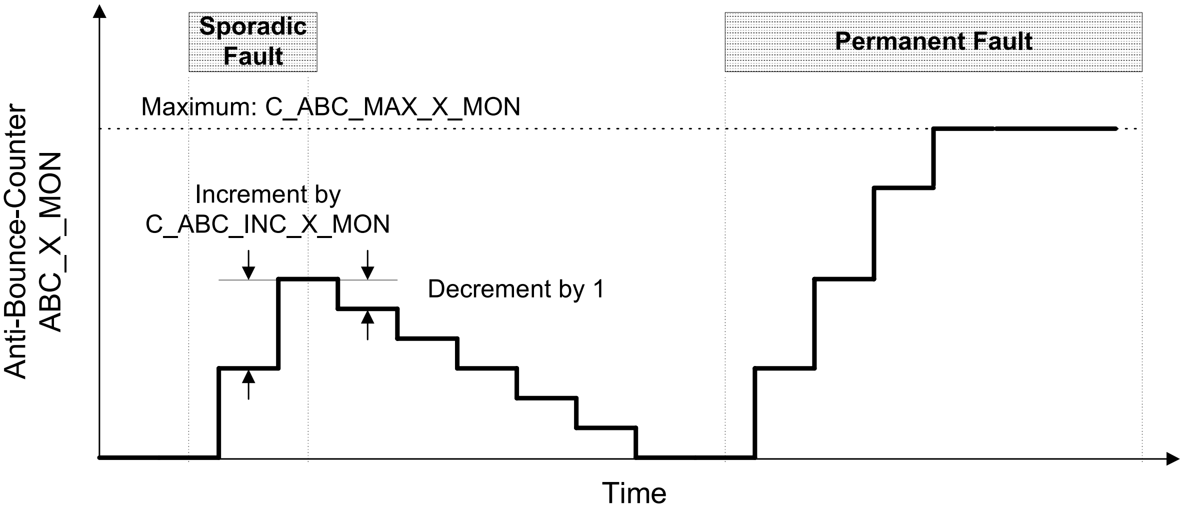
FUNCTION DESCRIPTION:

General information:

Many diagnosis algorithms need a debounce mechanism in order to avoid a faulty diagnosis caused by normal disturbances. Since the process monitoring is intended to represent redundancy to the functions of level 1, there is also a redundant debounce mechanism required. That means an algorithm with separate code, ROM- and RAM data. The debounce algorithm is used for the functions of level 2 and 2' (copy of process monitoring).

Description:

The anti-bounce-counter ABC_X_MON is increased by C_ABC_INC_X_MON, if the conditions for fault detection are fulfilled. Otherwise, the anti-bounce counter is decremented by 1. If the anti-bounce-counter ABC_X_MON exceeds its threshold C_ABC_MAX_X_MON the first time during this engine running, the relating error-flag is set and a fault can be stored in the customer error memory.



Application conditions:

The activation, deactivation and recurrency of this module depends on the data of the related diagnosis function.

Formula section:

```
IF   (Fault detection condition fulfilled)
THEN
     Increment ABC_X_MON by C_ABC_INC_X_MON
     (limited to C_ABC_MAX_X_MON)
ELSE
     Decrement ABC_X_MON by 1          (limited to 0)
```

11.6 Application Incidences and Configuration of Process Monitoring

Import actions:

ACTION_ECM3_Service0TaskPfm(IN <>)
ACTION_ECM3_Service1TaskPfm(IN <>)
ACTION_ECM3_Service2TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The input arguments have the values 1, 2, 3 and 4.

For further information concerning the order of action calls compared to the specified functions, please have a look on the following page.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service0TaskPfm() and ACTION_ECM3_Service1TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

ACTION_INFR_GetVTPSMON(OUT <V_tps_1_mon>, OUT<V_tps_2_mon>)

Note: This action is defined in chapter "ECM2 – Requirements to infrastructure interface".

Order of action calls:

1 Application Incidences and Configuration or Process Monitoring

Initialization conditions

ACTION_ECM3_Service0TaskPfm(1)

ACTION_ECM3_McChkStack()

ACTION_ECM3_Service1TaskPfm(1)

Application incidences for the activation conditions of process monitoring

Application incidences for monitoring of torque demand from idle speed controller

1.16.1 Acquisition of sport-mode-switch for monitoring

1.20 Application incidences for actual fuel mass flow

ACTION_ECM3_Service2TaskPfm(1)

ACTION_ECM3_Service0TaskPfm(2)

ACTION_ECM3_McChkStack()

ACTION_ECM3_Service1TaskPfm(2)

Application incidences for monitoring of pedal value signals

1.5 Adaptation of OPM_AV to OPM_AV_MON

1.6 Application incidences for the monitoring of torque losses

1.7 Appl. Inc. for actual lambda deviation

1.16.1 Acquisition of sport-mode-switch for monitoring

ACTION_ECM3_Service2TaskPfm(2)

ACTION_ECM3_Service0TaskPfm(3)

ACTION_ECM3_McChkStack()

ACTION_ECM3_Service1TaskPfm(3)

1.8 Forced injection shut off for Level 2 – engine speed limitation

1.9 State of torque intervention when level 2/3 error is detected

1.10 Application incidences for cruise control input

1.11 Application incidences for IPM monitoring

1.12 Application incidences for monitoring of CAN signals

1.13 Application incidences for monitoring of minimum torque at clutch

1.14 Application incidences for the actual efficiencies

1.15 Application incidences for the actual indicated engine torque

1.16.1 Acquisition of sport-mode-switch for monitoring

ACTION_ECM3_Service2TaskPfm(3)

ACTION_ECM3_Service0TaskPfm(4)

ACTION_ECM3_McChkStack()

ACTION_ECM3_Service1TaskPfm(4)

1.16 Sport switch monitoring

1.17 Driving off assistance

1.18 Application incidences for desired indicated engine torque

1.19 Application incidences for the combustion mode switch request

ACTION_ECM3_Service2TaskPfm(4)

11.6.1 Initialization conditions

FUNCTION DESCRIPTION:

General information:

The application condition for the initialization of e.g. the error bits, the anti-bounce-counters and other variables of the process monitoring is commonly defined here for all modules of the process monitoring. There is only a reference to the following definition of the application conditions in the different modules.

The error bits, anti-bounce-counters and other variables are initialized
- at reset
- at transition from LV_IGK = 0 to 1
- at clearing of the failure memory
- with physical "0" if not defined differently!

There is one exception from the rule above. If the reset counter at the monitoring unit increased up to 7 resets, the fault reaction of the monitoring unit will be continued until the power supply of the monitoring unit is switched off.

11.6.2 Application incidences for the activation conditions of process monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_N_LIM_ETC_MON	O/V	0...1H	0...1	1	[-]
Logical variable for activation of engine speed limitation monitoring					
LV_TQI_MON_ACT_MON	O/V	0...1H	0...1	1	[-]
Activation condition for fuel cut off monitoring fulfilled - generic interface					
V_TPS_1_MON	O/V	0...3FFH	0...4.99511	4.8828e-3	[V]
Adapted TPS signal channel 1					
V_TPS_2_MON	O/V	0...3FFH	0...4.99511	4.8828e-3	[V]
Adapted TPS signal channel 2					
LV_IGK_MON	O/V	0...1H	0...1	1	[-]
Logical variable for ignition key on					

Input data:

LV_N_LIM_REQ_MON	LV_N_LIM_ETC_LIH	LV_N_LIM_ETC_LIH_REV	PV_AV_MON
LV_MTC_CUR_OFF	LV_IGK	LV_N_LIM_REQ_RST_CHK	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_TPS_1_LIH_MON	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold for detection of throttle position 1 outside limp home position					
C_V_TPS_2_LIH_MON	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold for detection of throttle position 2 outside limp home position					

FUNCTION DESCRIPTION:

General information:

The logical variable LV_N_LIM_ETC_MON indicates, that any etc-safety-function of level 1, 2 or 3 has requested engine speed limitation or ignition key is switched off. The monitoring of the engine speed limitation is also active, when

- any fault of the ETC-system (pedal value sensor, throttle position sensor, throttle actuator, power stage, idle speed actuator) is present and the driver demand is zero
- throttle is disabled and at least one throttle position signal is outside the specified limp home position
- after a warm reset in case any etc-safety-function of level 1, 2 or 3 had requested engine speed limitation before the warm reset

The torque monitoring can only be active, when the ignition key is on and the throttle is not disabled and no mechanical/electrical error of the idle speed actuator is detected. Otherwise the control of the torque is not possible and torque monitoring is forbidden. This is not a risk, because engine is switched off or in limp home mode then, and the monitoring of the engine speed limitation is active in most cases (at least when the driver releases the pedal).

The logical variable LV_IGK is copied to level 2 in the variable LV_IGK_MON, because it is used as a deactivation condition for the monitoring of PVS in level 2. In order to ensure the safety, the variable LV_IGK_MON is also used for the calculation of the global activation conditions LV_N_LIM_ETC_MON and LV_TQI_MON_ACT_MON.

Application conditions:

Activation: at every engine state

Update Rate: 40ms

Formula section:

```
LV_IGK_MON = LV_IGK

If  LV_IGK_MON                      =     0     Or
    LV_N_LIM_REQ_MON                =     1     Or    (request from level 2, 3)
    LV_N_LIM_ETC_LIH                =     1     Or    (requests from level 1)
    LV_N_LIM_ETC_LIH_REV            =     1     Or    (rev. SAS active, level 1)
    LV_N_LIM_REQ_RST_CHK            =     1     Or    (reset resisting LIH)

    (LV_MTC_CUR_OFF         =     1     And
    PV_AV_MON               =     0)            Or

    (LV_MTC_CUR_OFF         =     1     And
    (V_TPS_1_MON >  C_V_TPS_1_LIH_MON           Or
    V_TPS_2_MON <  C_V_TPS_2_LIH_MON))

Then
    LV_N_LIM_ETC_MON        =     1
Else
    LV_N_LIM_ETC_MON        =     0
Endif


If  LV_IGK_MON              =     1     And   (ignition key on)
    LV_MTC_CUR_OFF          =     0     And   (MTC NOT switched off)
    LV_N_LIM_ETC_LIH_REV    =     0
Then
    LV_TQI_MON_ACT_MON      =     1
Else
    LV_TQI_MON_ACT_MON      =     0
Endif
```

11.6.3 Application incidences for monitoring of torque demand from idle speed controller

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CT_MON	O/V	0...1H	0...1	1	[-]
Logical bit for detecion of driver request passive					
FAC_RAMP_NEG_P_D_IS_MON	O/V	0...FA0H	0...0.5	0.000125	[-]
Factor for monitoring negative PD deactivation-ramp operations					
N_DIF_SP_IS_MON	O/V	80...7FH	-4096...4064	32	[rpm]
Deviation of actual engine speed from idle speed setpoint					
N_DIF_MON	O/V	80...7FH	-4096...4064	32	[rpm]
Engine speed deviation N_SP_IS - N					
N_SP_IS_RATIO_MON	O/V	0...FFFFH	0...7.99987	0.1221e-3	[-]
Ratio between actual engine speed and idle speed setpoint					
N_SP_IS_BAS_MON	V	0...FFH	0...8160	32	[rpm]
Monitored basis idle-speed setpoint					
N_SP_IS_MON	O/V	0...FFH	0...8160	32	[rpm]
Monitored value of idle speed setpoint					
N_DIF_MMV_MON	V	0...FFH	-4096...4064	32	[rpm]
N_DIF moving mean value					
N_DIF_COR_MON	V	0...FFH	-4096...4064	32	[rpm]
Idle speed control variable					
N_SP_IS_EXT_REQ_MON	V	0...FFH	0...8160	32	[rpm]
Plausible basis idle-speed setpoint while DROF or external tester demand active					
N_SP_IS_CS_MON	V	0...FFH	0...8160	32	[rpm]
Idle-speed setpoint given by DROF					
N_SP_IS_EXT_ADJ_MON	V	0...FFH	0...8160	32	[rpm]
Idle speed setpoint from external influences					
LV_AT_MON	O/V	0...1H	0...1	1	[-]
Monitoring bit for AT coded					
LV_VAR_TCT_MON	O/V	0...1H	0...1	1	[-]
TCT-transmission recognized					
LV_ACT_N_SP_IS_EXT_ADJ_MON	V	0...1H	0...1	1	[-]
LV for the displace of N_SP_IS_MON by tester demand					
N_VS_RATIO_MON	V	0...FFH	0...255	1	[rpm/(km/h)]
Monitored engine speed / vehicle speed ratio					
GR_MT_MON	V	0...FFH	0...255	1	[-]
Monitored value of manual transmission gear					
LV_NT_SO2P_EXT_ADJ_ENA_MON	V	0...1H	0...1	1	[-]
Condition for enabling desulfation by external request (monitoring)					
LV_TCHA_DIAG_REQ_MON	V	0...1H	0...1	1	[-]
Condition for enabling charger diagnosis by external request (monitoring)					
CTR_N_SP_IS_EXT_REQ_DEC_MON	V	0...FFH	0...255	1	[-]
Counter of iteration to allow N_SP_IS_EXT_REQ_MON decrementation					
LV_ACT_N_SP_IS_POIL_EXT_MON	V	0...1H	0...1	1	[-]
Condition for enabling oil pressure diagnosis by external request (monitoring)					
TQ_DIF_P_D_IS_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Monitored PD-part of the idle speed controller					
TQ_DIF_P_D_IS_MAX_MON	O/V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Maximum value of PD-part					
TQ_DIF_P_D_IS_MAX_2_MON	V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Temporary variable					
STATE_PAS_RAMP_ACT_P_D_IS_MON	O/V	0...FFH	0...255	1	[-]

Circular buffer indicating statuts of PD-part deactivation ramp					
STATE_PAS_RAMP_ACT_I_IS_MON	O/V	0...FFH	0...255	1	[-]
Circular buffer indicating statuts of I-part deactivation ramp					
LV_PAS_RAMP_ACT_P_D_CHG_MON	O/V	0...1H	0...1	1	[-]
Activation bit for initialisation of PD-ramp					
LV_PAS_RAMP_ACT_I_CHG_MON	O/V	0...1H	0...1	1	[-]
Activation bit for initialisation of I-ramp					
STATE_REQ_ISC_MON	O/V	0H 1H 2H 3H	NOT_ACTIVE IDLE TRAILING_THR PART_LOAD	1	[-]
State of idle speed controller monitoring					

Input data:

LV_ERR_PVS_RATIO_MON	LV_ERR_CAN_BOFF	TCO_MON	N_SP_IS
LV_N_SP_IS_LIH_ACT	N_SP_IS_CS	GEAR_INFO	LV_CT
STATE_GEAR_REV_AT_AMT	N_SP_IS_POIL_CTL	N_SP_IS_EXT_ADJ	LV_AT
LV_POIL_EXT_ADJ_ACT	LV_IM_BTS_MON	LV_ERR_BN_ETCU	VS_MON
LV_ACT_N_SP_IS_EXT_ADJ	N_SP_IS_BRAKE	LV_IM_BLS_MON	GEAR
STATE_EOL_KWP_N_SP_IS	T_CTR_PU_ISC_ACT	N_32_MON	PV_AV_MON
LV_NT_SO2P_EXT_ADJ_ACT	TQ_DIF_P_D_SLOW_IS	LV_TCHA_DIAG_REQ	LV_CS_MON
STATE_PAS_RAMP_ACT_I_IS	TQ_DIF_P_D_FAST_IS	LV_REQ_ISC	LV_CLU_SWI
STATE_PAS_RAMP_ACT_P_D_IS	LV_TQ_DROF_ACT_MON	LV_VAR_TCT	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_RAMP_P_D_MON	1	0...FA0H	0...0.5	0.000125	[-]
Factor for monitoring negative DROF deactivation-ramp operation					
C_N_SP_IS_BAS_DIF_MAX_MON	1	0...7FH	0...4064	32	[rpm]
Idle speed gradient limitation					
C_N_SP_IS_CS_MAX_MON	1	0...FFH	0...8160	32	[rpm]
Max. selection of Idle speed setpoint for drive-off-support					
IP_N_SP_IS_BAS_MAX_MON	5	0...FFH	0...8160	32	[rpm]
LDP_TCO_MON_IP_N_SP_IS_MON	5	0...FEH	-48...142.5	0.75	[°C]
Maximum basis idle speed setpoint depending on coolant temperature					
LC_N_SP_IS_CLC_INH_MON	1	0...1H	0...1	1	[-]
Logical bit for switching between two engine-speed-deviation calculation paths(N_DIF_COR_x_MON)					
C_N_DIF_CRLC_HIGH_MON	1	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation factor for N_DIF_MMV calculation					
C_N_DIF_FAC_MON	1	0...FFH	0...0.99609	3.9063e-3	[-]
Multiplicative factor for N_DIF_COR calculation					
C_N_DIF_CRLC_LOW_MON	1	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation factor for N_DIF_MMV calculation					
C_N_SP_IS_LIH_MAX_MON	1	0...FFH	0...8160	32	[rpm]
Maximum idle-speed setpoint when error with acclerator pedal module detected					
C_N_SP_IS_MAX_EXT_ADJ_MON	1	0...FFH	0...8160	32	[rpm]
Maximum idle-speed setpoint when external Tester demand is active					
C_N_MAX_KWP_MON	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed for accepting the EOL- N_SP_IS_MON displace service					
C_VS_MAX_KWP_MON	1	0...FFH	0...255	1	[km/h]
Maximum vehicle speed for accepting the EOL- N_SP_IS_MON displace service					
C_MIN_VS_RATIO_MON	1	0...FFH	0...255	1	[km/h]
Minimum engine speed for the calculation of the monitored engine speed / vehicle speed ratio					
ID_GR_MT_MON	13	0...FFH	0...255	1	[-]
LDP_N_VS_RATIO_MON_GR_MT_MON	13	0...FFH	0...255	1	[rpm/(km/h)]
Index table for Gear Ratio detection in the monitoring level					
C_N_NT_SO2P_EXT_ADJ_ACT_MON	1	0...FFH	0...8160	32	[rpm]
Idle speed setpoint for SO2P in service (monitoring)					
C_VS_SO2P_EXT_MAX_MON	1	0...FFH	0...255	1	[km/h]
Vehicle speed threshold for disabling external request for desulfation (Monitoring)					
C_PV_SO2P_EXT_MAX_MON	1	0...FFH	0...99.60937	0.390625	[%]
Accelerator pedal threshold for disabling external request for desulfation (Monitoring)					
C_N_IS_SO2P_MIN_MON	1	0...FFH	0...8160	32	[rpm]
Threshold for detecting running engine (Monitoring)					
C_VS_TCHA_DIAG_EXT_MAX_MON	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed threshold for disabling external request for charger diagnosis (Monitoring)					
C_PV_TCHA_DIAG_EXT_MAX_MON	1	0...FFH	0...99.60937	0.390625	[%]
Accelerator pedal threshold for disabling external request for charger diagnosis (Monitoring)					
C_N_TCHA_DIAG_REQ_MAX_MON	1	0...FFH	0...8160	32	[rpm]
Maximum idle speed setpoint for charger diagnosis in service (monitoring)					
C_N_SP_LGRD_IS_EXT_REQ_DEC_MON	1	0...FFH	0...8160	32	[rpm]
Decrement for idle speed SP for external request (Monitoring)					
C_N_SP_LGRD_IS_EXT_REQ_INC_MON	1	0...FFH	0...8160	32	[rpm]
Increment for idle speed SP for external request (Monitoring)					
C_CTR_N_SP_IS_REQ_DEC_MAX_MON	1	0...FFH	0...255	1	[-]
Maximum for counter of iteration to allow N_SP_IS_EXT_REQ_MON decrementation					
C_VS_POIL_EXT_ADJ_MAX_MON	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed threshold for disabling external request for oil pressure diagnosis (Monitoring)					
C_PV_POIL_EXT_ADJ_MAX_MON	1	0...FFH	0...99.60937	0.390625	[%]

Accelerator pedal threshold for disabling external request for oil pressure diagnosis (Monitoring)					
C_N_POIL_EXT_ADJ_MAX_MON	1	0...FFH	0...8160	32	[rpm]
Maximum idle speed setpoint for oil pressure diagnosis in service (monitoring)					
C_N_DIF_MAX_IS_MON	1	80...7FH	-4096...4064	32	[rpm]
Maximum idle speed setpoint deviation for active PD-part					
C_N_DIF_HYS_IS_PU_MON	1	80...7FH	-4096...4064	32	[rpm]
N_DIF_MON hysteresis before state PU					
C_FIL_TQ_P_D_IS_MAX_MON	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Filter correlation constant for PD-part torque threshold					
IP_TQ_DIF_P_D_IS_MAX_MON	6	0...FFH	0...510	2	[Nm]
LDP_N_DIF_SP_IS_MON_IP_TQ_MON	6	0...FFH	-4096...4064	32	[rpm]
Maximum PD-part depending on speed deviation to idle speed setpoint					
IP_FAC_TQ_DIF_P_D_IS_MON	5	0...FFH	0...1.99218	0.0078125	[-]
LDP_TCO_MON_IP_FAC_TQ_P_D_MON	5	0...FEH	-48...142.5	0.75	[°C]
Maximum scaling factor due to separation of PD-part					
LC_TQ_DROF_ISC_OFF_MON	1	0...1H	0...1	1	[-]
Boolean to enable/disable idle speed controller activation during drive off; 1=disabled (monitoring)					
C_N_SP_IS_MAX_PL_ACT_MON	1	0...FFH	0...8160	32	[rpm]
Maximum allowed idle-speed for active ISC in PL					
C_N_DIF_MAX_PL_ACT_MON	1	80...7FH	-4096...4064	32	[rpm]
Maximum idle-speed deviation allowed for active ISC in PL					

11.6.3.1 Application incidences for monitoring driver request passive

FUNCTION DESCRIPTION:

The aim of this chapter is the monitoring of the variable LV_CT which indicates whether a ‘driver request’ is present or not where the request can be not only from the driver but also from other sources like cruise control or vehicle speed limitation functionality.

As long as the impact between LV_CT = 0 and LV_CT = 1 can be tolerated in terms of torque thresholds, the value can be copied to L2 without plausibility check.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialisation: for condition see ‘Application Incidences of Process Monitoring’

LV_CT_MON = 1

Update Rate: 40 ms

Formula section:

LV_CT_MON = LV_CT

11.6.3.2 Application incidences for monitoring of idle speed setpoint

General information:

The objective of this function is the monitoring of the idle speed setpoint (N_SP_IS) and the corrected idle speed setpoint. Both of these values have to be monitored because they are important inputs for the torque monitoring functionalities. This function consists of two sections.

The first section involves the monitoring of the basis idle-speed setpoint and external offsets (driving-off assistance, CVT etc.) that might be added depending on certain activation conditions. In the second part of this functionality, the Level 2 model value for the idle-speed setpoint and other torque monitoring relevant variables (engine speed deviation, engine speed ratio etc.) are calculated.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialisation: (for condition see ‘Application Incidences of Process Monitoring’
OR LV_TQI_MON_ACT_MON = 0 → 1)
N_DIF_SP_IS_MON = 0 rpm
N_SP_IS_RATIO_MON = 0 rpm
N_SP_IS_MON = 0 rpm
N_DIF_MON = 0 rpm
N_DIF_MMV_MON = 0 rpm
N_SP_IS_BAS_MON = **min(**N_SP_IS, IP_N_SP_IS_BAS_MAX_MON**)**
FAC_RAMP_NEG_P_D_IS_MON = 0
LV_AT_MON = LV_AT
LV_VAR_TCT_MON = LV_VAR_TCT
N_SP_IS_EXT_ADJ_MON = **min**(N_SP_IS, C_N_SP_IS_MAX_EXT_ADJ_MON)
LV_ACT_N_SP_IS_EXT_ADJ_MON = 0
GR_MT_MON = 0 **at** [LV_ES = 1 **or** LV_ST = 1 **or** LV_CS = 1]
N_VS_RATIO_MON = 0
N_SP_IS_CS_MON = **min**(N_SP_IS, C_N_SP_IS_CS_MAX_MON)
N_SP_IS_EXT_REQ_MON $_{K}$ = **min(**N_SP_IS, IP_N_SP_IS_BAS_MAX_MON**)**
N_SP_IS_EXT_REQ_MON $_{K-1}$= **min(**N_SP_IS, IP_N_SP_IS_BAS_MAX_MON**)**
N_DIF_COR_MON = 0
LV_NT_SO2P_EXT_ADJ_ENA_MON= 0
CTR_N_SP_IS_EXT_REQ_DEC_ MON $_{K}$ = 0
CTR_N_SP_IS_EXT_REQ_DEC_ MON $_{K-1}$ = 0
LV_TCHA_DIAG_REQ_MON = 0
LV_ACT_N_SP_IS_POIL_EXT_MON = 0

Update rate: 40 ms

11.6.3.2.1 Plausibilisation of basis idle-speed setpoint and external-influences

General information:

The basis idle speed setpoint is limited by a coolant temperature dependent maximum value, IP_N_SP_IS_BAS_MAX_MON. Under certain circumstances (for e.g. driving off situation i.e. the driver is slightly on the gas pedal and releases the clutch abruptly --> can lead to engine stalling), to assist the driver in this situation, N_SP_IS_MON is raised to achieve a torque increasing demand from ISC. In case of an error, such external influences can lead to relatively large ‘unwanted’ acceleration which makes the monitoring of the relevant activation conditions imperative.

Formula section:

{N_SP_IS_LIM_MON represents the basis idle-speed setpoint plus any offsets (for e.g. catalyst heating or low battery voltage) that might be added in L1 based on certain activation conditions.}

N_SP_IS_LIM_MON = N_SP_IS

{Model value of maximum basis setpoint}
N_SP_IS_BAS_MAX_MON = IP_N_SP_IS_BAS_MAX_MON(TCO_MON)

{Idle speed setpoint plausibilisation through gradient limitation}
IF |N_SP_IS_LIM_MON$_{k}$ - N_SP_IS_BAS_MON$_{k-1}$**|** > C_N_SP_IS_BAS_DIF_MAX_MON
THEN N_SP_IS_BAS_MON$_{k}$ = **min(**N_SP_IS_BAS_MON$_{k-1}$, N_SP_IS_BAS_MAX_MON**)**
ELSE N_SP_IS_BAS_MON$_{k}$ = **min(**N_SP_IS_LIM_MON$_{k}$, N_SP_IS_BAS_MAX_MON**)**
ENDIF

FAC_RAMP_NEG_P_D_IS_MON = C_FAC_RAMP_P_D_MON

11.6.3.2.2 Calculation of idle-speed setpoint model value

The outputs of the function are:

- ratio between idle speed setpoint and actual engine speed - **N_SP_IS_RATIO_MON**
- deviation of actual engine speed from idle speed setpoint - **N_DIF_SP_IS_MON**
- deviation of actual engine speed from N_SP_IS - **N_DIF_MON**

Formula section:

Plausibility check for IS_SP request from external tester (KWP-Service)

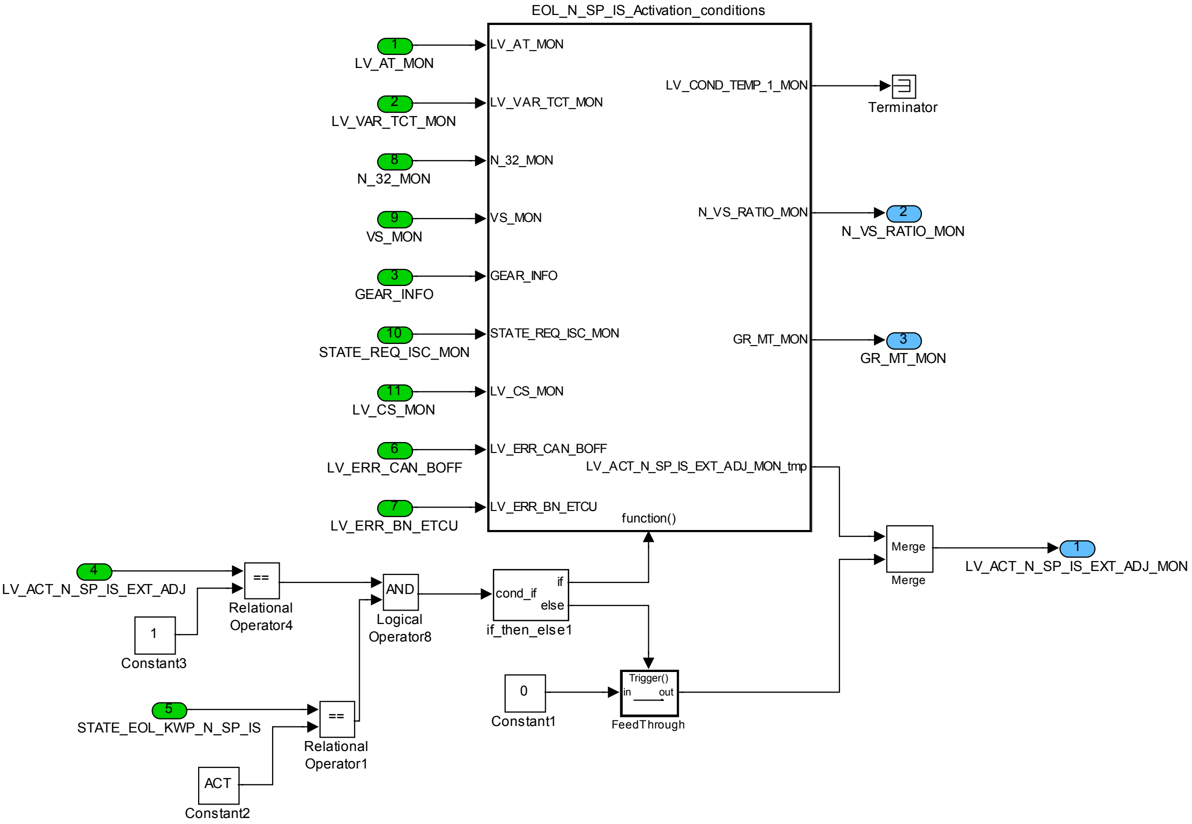


Figure 1.: ECM2_CONFIACPR/EOLService_N_SP_IS_Displace

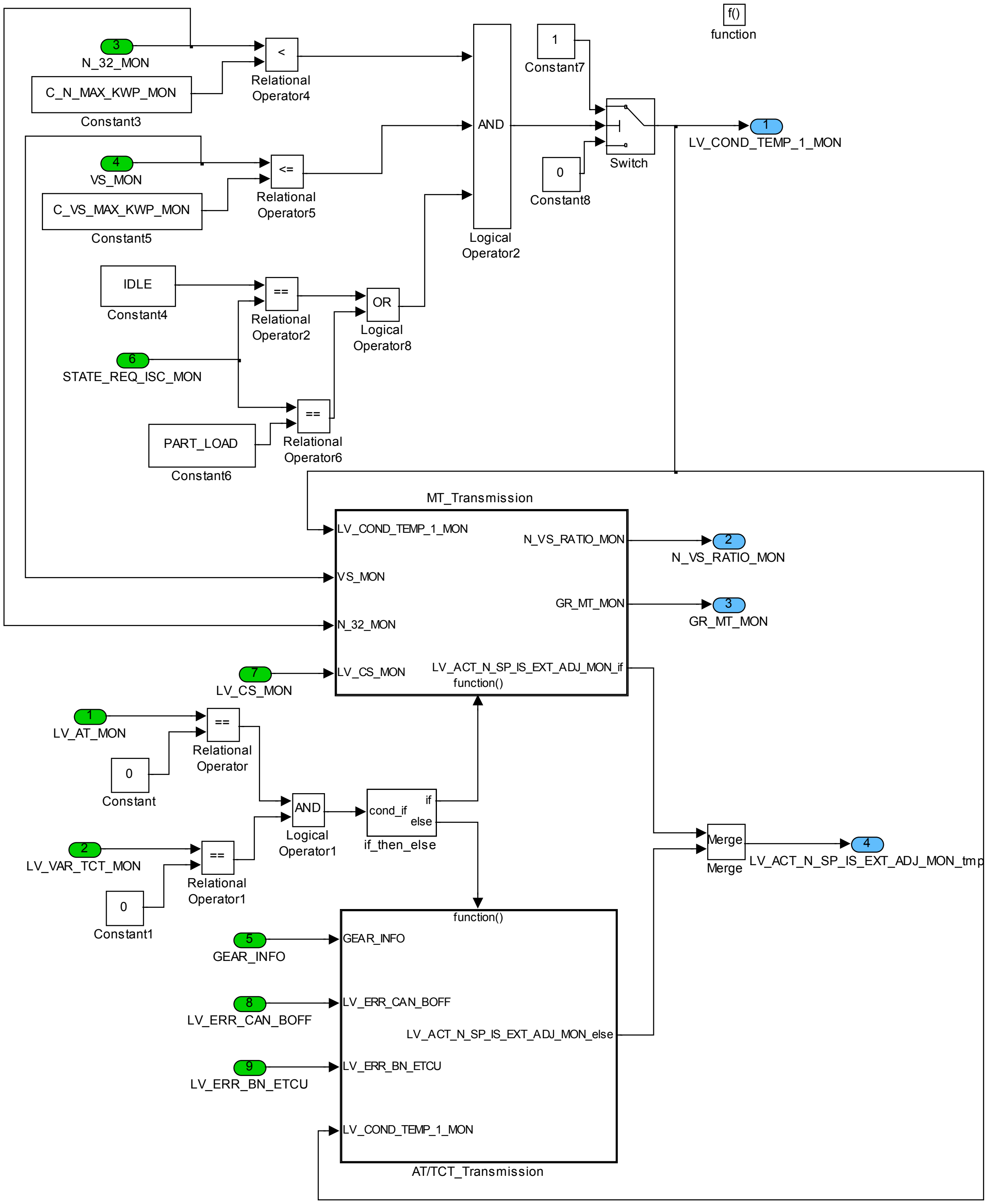


Figure 2.: ECM2_CONFIACPR/EOLService_N_SP_IS_Displace/ EOL_N_SP_IS_Activation_conditions

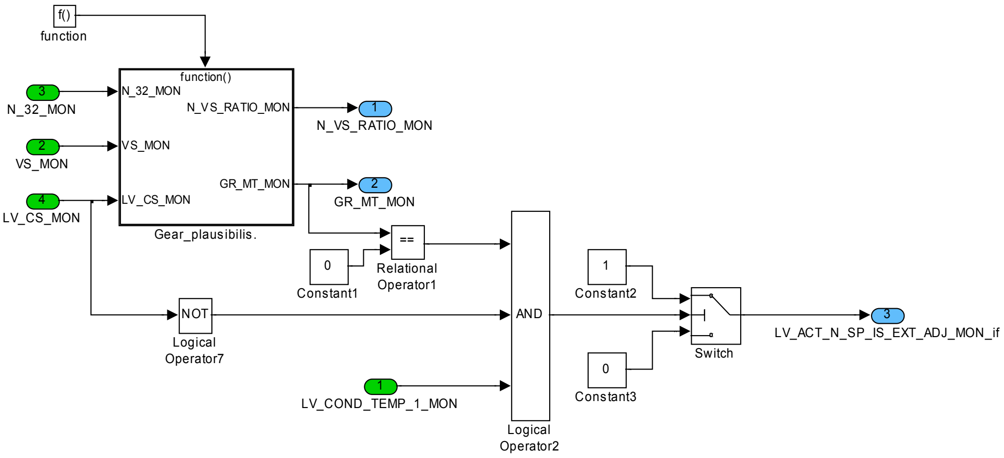


Figure 3.: ECM2_CONFIACPR/EOLService_N_SP_IS_Displace/ EOL_N_SP_IS_Activation_conditions/MT_Transmission

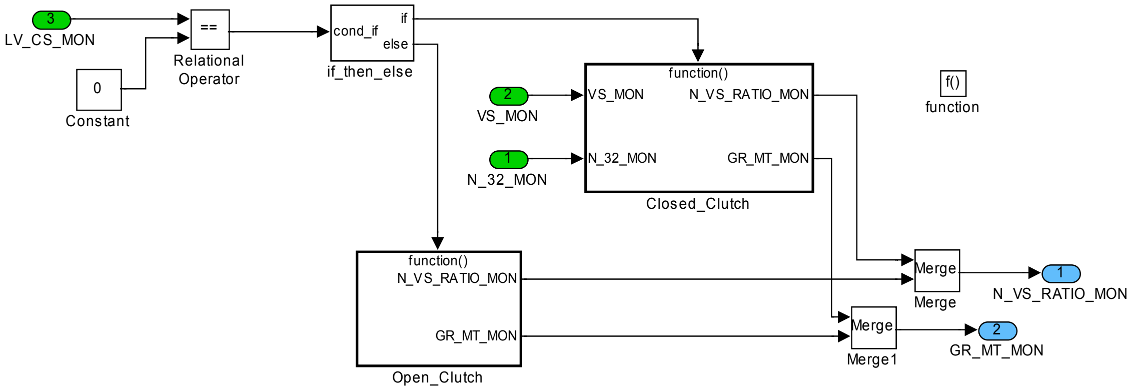


Figure 4.: ECM2_CONFIACPR/EOLService_N_SP_IS_Displace/ EOL_N_SP_IS_Activation_conditions/MT_Transmission/ Gear_plausibilis.

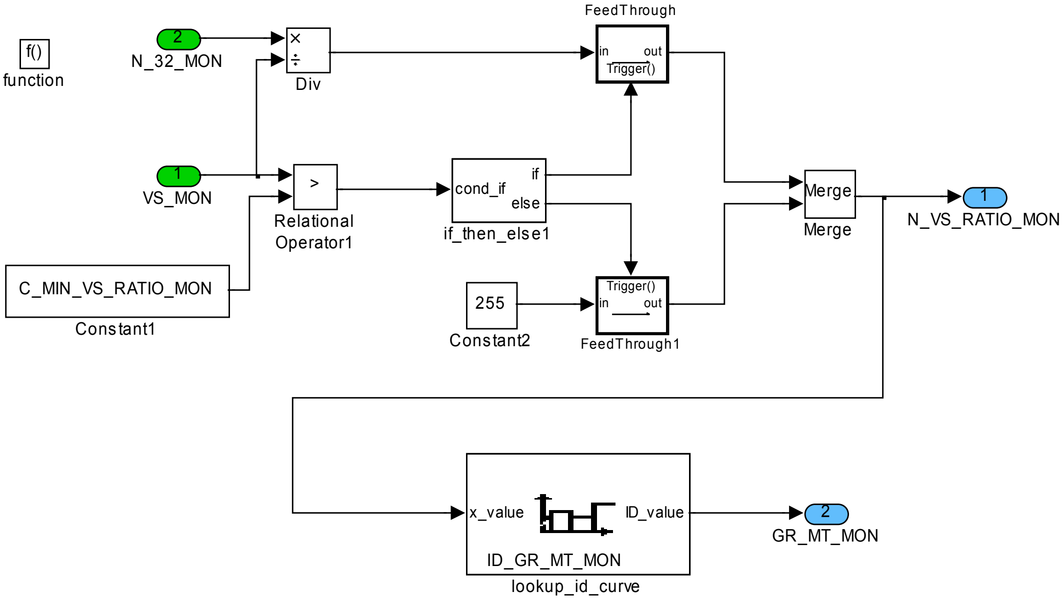


Figure 5.: ECM2_CONFIACPR/EOLService_N_SP_IS_Displace/
EOL_N_SP_IS_Activation_conditions/MT_Transmission/ Gear_plausibilis./ Closed_Clutch

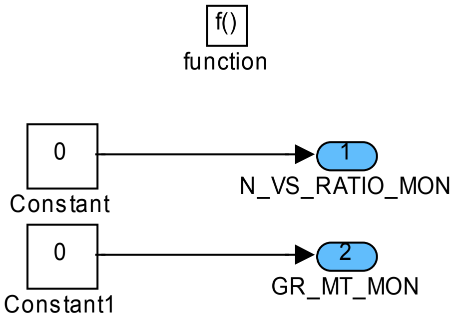


Figure 6.: ECM2_CONFIACPR/EOLService_N_SP_IS_Displace/
EOL_N_SP_IS_Activation_conditions/MT_Transmission/ Gear_plausibilis./ Open_Clutch

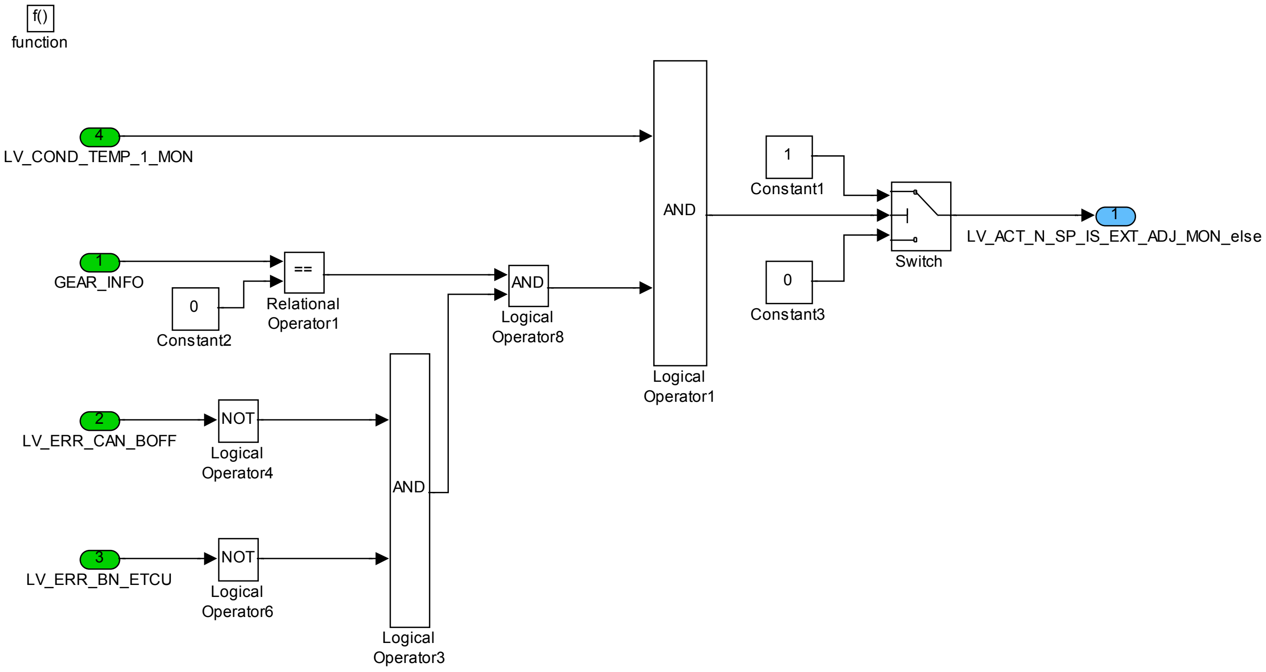


Figure 7.: ECM2_CONFIACPR/EOLService_N_SP_IS_Displace/
EOL_N_SP_IS_Activation_conditions/AT/TCT_Transmission

Calculation of LV_NT_SO2P_EXT_ADJ_ENA_MON
(raised idle speed SP due to external demand for desulfation)

If LV_NT_SO2P_EXT_ADJ_ACT = 1 AND
N_32_MON >= C_N_IS_SO2P_MIN_MON AND
GEAR = 0 AND
STATE_GEAR_REV_AT_AMT = 0 AND
LV_CLU_SWI = 0 AND
VS_MON <= C_VS_SO2P_EXT_MAX_MON AND
PV_AV_MON <= C_PV_SO2P_EXT_MAX_MON

Then

```
        LV_NT_SO2P_EXT_ADJ_ENA_MON = 1
Else
        LV_NT_SO2P_EXT_ADJ_ENA_MON = 0
Endif
```

Calculation of LV_TCHA_DIAG_REQ_MON
(raised idle speed SP due to external demand for charger diagnosis)

```
If      LV_TCHA_DIAG_REQ = 1                            AND
        LV_CLU_SWI = 0                                  AND
        VS_MON < C_VS_TCHA_DIAG_EXT_MAX_MON             AND
        PV_AV_MON < C_PV_TCHA_DIAG_EXT_MAX_MON
Then
        LV_TCHA_DIAG_REQ_MON = 1
Else
        LV_TCHA_DIAG_REQ_MON = 0
Endif
```

Calculation of LV_ACT_N_SP_IS_POIL_EXT_MON
(raised idle speed SP due to external demand for oil pressure diagnosis)

```
If      LV_POIL_EXT_ADJ_ACT = 1                         AND
        GEAR = 0                                        AND
        STATE_GEAR_REV_AT_AMT = 0                       AND
        LV_CLU_SWI = 0                                  AND
        VS_MON < C_VS_POIL_EXT_ADJ_MAX_MON              AND
        PV_AV_MON < C_PV_POIL_EXT_ADJ_MAX_MON
Then
        LV_ACT_N_SP_IS_POIL_EXT_MON = 1
Else
        LV_ACT_N_SP_IS_POIL_EXT_MON = 0
Endif
```

Calculation of N_SP_IS_CS_MON (only if LV_TQ_DROF_ACT_MON = 1)
(raised idle speed SP due to DROF active)

```
{MAX-limitation}
N_SP_IS_CS_MON = min(N_SP_IS_CS, C_N_SP_IS_CS_MAX_MON)
```

Calculation of N_SP_IS_EXT_ADJ_MON
(only if LV_ACT_N_SP_IS_EXT_ADJ_MON = 1)

(raised idle speed SP due to external tester demand via K-line active)

```
{MAX-limitation}
N_SP_IS_EXT_ADJ_MON = min(N_SP_IS_EXT_ADJ,
                          C_N_SP_IS_MAX_EXT_ADJ_MON)
```

Calculation of N_SP_IS_EXT_REQ_MON
(idle speed SP due to active DROF or external tester demand)

```
IF          LV_ACT_N_SP_IS_EXT_ADJ_MON = 1

THEN        N_SP_IS_EXT_REQ_MON = N_SP_IS_EXT_ADJ_MON

ELSE IF     LV_TQ_DROF_ACT_MON = 1

THEN        N_SP_IS_EXT_REQ_MON = N_SP_IS_CS_MON

ELSE IF     LV_NT_SO2P_EXT_ADJ_ENA_MON = 1

THEN        N_SP_IS_EXT_REQ_MON = C_N_NT_SO2P_EXT_ADJ_ACT_MON

ELSE IF     LV_TCHA_DIAG_REQ_MON = 1

THEN        N_SP_IS_EXT_REQ_MON  = min(N_SP_IS_BRAKE,
                                       C_N_TCHA_DIAG_REQ_MAX_MON)

ELSE IF     LV_ACT_N_SP_IS_POIL_EXT_MON = 1

THEN        N_SP_IS_EXT_REQ_MON  = min(N_SP_IS_POIL_CTL,
                                       C_N_POIL_EXT_ADJ_MAX_MON)

ELSE        N_SP_IS_EXT_REQ_MON = N_SP_IS_BAS_MON

END IF

{gradient limitation}
IF(1)       N_SP_IS_EXT_REQ_MON $_{K}$ < N_SP_IS_EXT_REQ_MON $_{K-1}$

THEN(1)

    IF(2)       CTR_N_SP_IS_EXT_REQ_DEC_MON $_{K-1}$ >=
                                    C_CTR_N_SP_IS_REQ_DEC_MAX_MON

    THEN(2)

                CTR_N_SP_IS_EXT_REQ_DEC_MON $_{K}$ = 0
                N_SP_IS_EXT_REQ_MON $_{K}$ =
                        min(N_SP_IS, N_SP_IS_EXT_ REQ _MON $_{K-1}$ -
                                     C_N_SP_LGRD_IS_EXT_REQ_DEC_MON)
    ELSE(2)

                CTR_N_SP_IS_EXT_REQ_DEC_MON $_{K}$ =
                                    CTR_N_SP_IS_EXT_REQ_DEC_MON $_{K-1}$ + 1
                N_SP_IS_EXT_REQ_MON $_{K}$ = N_SP_IS_EXT_REQ_MON $_{K-1}$

    END IF(2)
```

```
ELSE IF(1)    N_SP_IS_EXT_REQ_MON $_K$ > N_SP_IS_EXT_REQ_MON $_{K-1}$

THEN(1)
              CTR_N_SP_IS_EXT_REQ_DEC_MON $_K$ = 0
              N_SP_IS_EXT_REQ_MON$_K$ = min(N_SP_IS, N_SP_IS_EXT_REQ_MON$_{K-1}$ +
                                    C_N_SP_LGRD_IS_EXT_REQ_INC_MON)

ELSE(1)
              CTR_N_SP_IS_EXT_REQ_DEC_MON $_K$ = 0
END IF(1)


{Calculation of final L2 idle-speed setpoint model value}

IF (LV_N_SP_IS_LIH_ACT==1) AND (LV_ERR_PVS_RATIO_MON==1)

THEN N_SP_IS_MON = min(N_SP_IS, C_N_SP_IS_LIH_MAX_MON)

ELSE N_SP_IS_MON     =     max(N_SP_IS_BAS_MON,
                               N_SP_IS_EXT_REQ_MON)
ENDIF


   N_SP_IS_RATIO_MON = N_32_MON / N_SP_IS_MON     {Idle-speed setpoint ratio}

   {Deviation between engine speed idle-speed setpoint}

   N_DIF_MON = N_SP_IS_MON – N_32_MON

   {Calculation of engine speed deviation – N_DIF_COR_MON}
   IF     LC_N_SP_IS_CLC_INH_MON == 1
   THEN   {Calculation of engine speed deviation from idle setpoint}
          N_DIF_COR_MON  = N_SP_IS_MON - N_32_MON
   ELSE   {Limitation of corrected idle speed setpoint between min&max thresholds}
              IF      N_32_MON > N_SP_IS_MON
              THEN N_DIF_CRLC_MON = C_N_DIF_CRLC_HIGH_MON
              ELSE  N_DIF_CRLC_MON = C_N_DIF_CRLC_LOW_MON
              ENDIF
              N_DIF_MMV_MON (n) = N_DIF_MMV_MON (n-1) + (N_DIF_MON -
              N_DIF_MMV_MON(n-1)) * N_DIF_CRLC_MON
          {Calculation of engine speed deviation from corrected idle setpoint}

              N_DIF_COR_MON = N_DIF_MON - N_DIF_MMV_MON * C_N_DIF_FAC_MON
   ENDIF

   {Deviation between engine speed and corrected idle-speed setpoint – input for ISC}
```

N_DIF_SP_IS_MON = N_DIF_COR_MON

11.6.3.3 Monitoring of Idle-speed controller activation conditions

Application conditions:

Activation:	LV_TQI_MON_ACT_MON = 1
Deactivation:	otherwise
Initialisation:	(for condition see 'Application Incidences of Process Monitoring' ***OR*** LV_TQI_MON_ACT_MON = 0 → 1) **STATE_REQ_ISC_MON = 'NOT_ACTIVE'** TQ_DIF_P_D_IS_MAX_2_MON = **max**(TQ_DIF_P_D_SLOW_IS$_k$, TQ_DIF_P_D_FAST_IS$_k$) TQ_DIF_P_D_IS_MON = 0 Nm TQ_DIF_P_D_IS_MAX_MON = **max**(TQ_DIF_P_D_SLOW_IS$_k$, TQ_DIF_P_D_FAST_IS$_k$) STATE_PAS_RAMP_ACT_P_D_IS_MON = 0 STATE_PAS_RAMP_ACT_I_IS_MON = 0 LV_PAS_RAMP_ACT_P_D_CHG_MON = 0 LV_PAS_RAMP_ACT_I_CHG_MON = 0
Update Rate:	40 ms

General information:

The ISC can be active either when the engine state idle is requested(LV_REQ_ISC =1) or when the DROF functionality is active or under certain conditions in trailing-throttle state (LV_PU=1) and part-load state (LV_PL=1) and. So all of these possible "ISC-active" phases have to be monitored. The activation conditions are based on the L1 ISC activation conditions as described in the module 'Application incidences for the idle speed controller'. In addition to the normal activation states, a 'NOT_ACTIVE' state is generated in order to be able to monitor the ISC if the activation information from the function level is faulty.

Formula section:

{The L1 ISC-Torque demand signal is copied to the process monitoring (L2) RAM}
TQ_DIF_P_D_IS_MON$_k$ = **max**(TQ_DIF_P_D_SLOW_IS$_k$, TQ_DIF_P_D_FAST_IS$_k$)

{L2 threshold of 'normal' PD-part torque request}
TQ_DIF_P_D_IS_MAX_1_MON = IP_TQ_DIF_P_D_IS_MAX_MON(N_DIF_SP_IS_MON$_k$)

{Calculation of monitoring threshold for PD-part and corresponding filter parameters. Filtering only in falling direction.}

```
IF TQ_DIF_P_D_IS_MAX_1_MON$_k$ < TQ_DIF_P_D_IS_MAX_2_MON$_{k-1}$
THEN TQ_DIF_P_D_IS_MAX_2_MON$_k$ =
        [C_FIL_TQ_P_D_IS_MAX_MON * TQ_DIF_P_D_IS_MAX_1_MON$_k$
       + (1- C_FIL_TQ_P_D_IS_MAX_MON) * TQ_DIF_P_D_IS_MAX_2_MON$_{k-1}$]
ELSE TQ_DIF_P_D_IS_MAX_2_MON$_k$ = TQ_DIF_P_D_IS_MAX_1_MON$_k$
```

```
ENDIF
        $\mathrm{TQ\_DIF\_P\_D\_IS\_MAX\_MON}_k = \mathrm{TQ\_DIF\_P\_D\_IS\_MAX\_2\_MON}_k$
                                        $* \mathrm{IP\_FAC\_TQ\_DIF\_P\_D\_IS\_MON(TCO\_MON)}$
{Evaluation of ISC-deactivation-ramp status in L1 to check for activation during current
recurrence or between two L2 recurrences. To simplify the usage of individual Bit
positions in the formula section, the following representation will be used:
STATE_PAS_RAMP_ACT_P_D_IS_MON(i) = STATE_PAS_RAMP_ACT_P_D_IS_MON : Bit i where i=0 ... 4
STATE_PAS_RAMP_ACT_I_IS_MON (j) = STATE_PAS_RAMP_ACT_I_IS_MON : Bit j → where j=0 ... 2 }

STATE_PAS_RAMP_ACT_P_D_IS_MON = STATE_PAS_RAMP_ACT_P_D_IS
IF {(STATE_PAS_RAMP_ACT_P_D_IS_MON (0) == 1
                AND STATE_PAS_RAMP_ACT_P_D_IS_MON (4) == 0)
        OR [ ( STATE_PAS_RAMP_ACT_P_D_IS_MON (0) == 1
                AND  STATE_PAS_RAMP_ACT_P_D_IS_MON (4) == 1 )
            AND (STATE_PAS_RAMP_ACT_P_D_IS_MON (1) == 0
                    OR STATE_PAS_RAMP_ACT_P_D_IS_MON (2) == 0
                        OR STATE_PAS_RAMP_ACT_P_D_IS_MON (3) == 0 ) ]
    OR (LV_TQI_MON_ACT_MON = 0 → 1 AND STATE_PAS_RAMP_ACT_P_D_IS_MON (0) == 1)}
THEN    LV_PAS_RAMP_ACT_P_D_CHG_MON = 1
ELSE    LV_PAS_RAMP_ACT_P_D_CHG_MON = 0
ENDIF

STATE_PAS_RAMP_ACT_I_IS_MON = STATE_PAS_RAMP_ACT_I_IS
IF (STATE_PAS_RAMP_ACT_I_IS_MON (0)==1
                AND STATE_PAS_RAMP_ACT_I_IS_MON(2)==0)
        OR [STATE_PAS_RAMP_ACT_I _IS_MON (0) ==1
                AND STATE_PAS_RAMP_ACT_I _IS_MON (2) == 1
                    AND STATE_PAS_RAMP_ACT_I _IS_MON (1) == 0 ]
OR (LV_TQI_MON_ACT_MON = 0 → 1 AND STATE_PAS_RAMP_ACT_I_IS_MON (0)==1)
THEN    LV_PAS_RAMP_ACT_I_CHG_MON = 1
ELSE    LV_PAS_RAMP_ACT_I_CHG_MON = 0
ENDIF

{Plausibilisation of ISC activation conditions}
IF [(N_DIF_MON ≥ C_N_DIF_MAX_IS_MON)    AND (LV_CT_MON == 1)
                                        AND  (LV_REQ_ISC == 1)]
THEN  STATE_REQ_ISC_MON = ‘IDLE’
ELSE  IF[ (N_DIF_MON < C_N_DIF_MAX_IS_MON - C_N_DIF_HYS_IS_PU_MON)
            AND (LV_CT_MON = 1) AND (T_CTR_PU_ISC_ACT == 0)]
        THEN    STATE_REQ_ISC_MON = ‘TRAILING_THR’
        ELSE  IF (LV_TQ_DROF_ACT_MON == 1 AND LC_TQ_DROF_ISC_OFF_MON == 0)
                OR  ( (N_SP_IS_MON < C_N_SP_IS_MAX_PL_ACT_MON) AND
            (LV_CT_MON == 0) AND (N_DIF_MON > C_N_DIF_MAX_PL_ACT_MON))
```

```
            OR LV_MSR_ACT_MON = 1
        THEN  STATE_REQ_ISC_MON = ‘PART_LOAD’
        ELSE STATE_REQ_ISC_MON = ‘NOT_ACTIVE’
        ENDIF
    ENDIF
ENDIF
```

11.6.4 Application incidences for monitoring of pedal value signals

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_BRAKE_MON	O/V	0...1H	0...1	1	[-]
Brake active or brake fault present					
LV_ERR_VCC_PVS_1_MON	O/V	0...1H	0...1	1	[-]
Fault in power supply for PVS_1 present					
LV_ERR_VCC_PVS_2_MON	O/V	0...1H	0...1	1	[-]
Fault in power supply for PVS_2 present					
LV_ERR_VCC_PVS_MON	O/V	0...1H	0...1	1	[-]
Fault in power supply for PVS present					

Input data:

LV_ERR_BLS_PLAUS	LV_IM_BLS_MON	LV_IM_BTS_MON	LV_ERR_V_REF_1
LV_ERR_V_REF_2			

FUNCTION DESCRIPTION:

General information:

In order to be able to use the generic module for the monitoring of the pedal value signals in level 2, different information about the state of the brake are combined to one resulting information LV_BRAKE_MON. It is set, when at least one sensor (brake light switch or brake test switch) indicates an activated brake or a fault of a brake switch has been detected. The switch signals are read directly from port.

Application conditions:

Activation:	at every engine state
Deactivation:	otherwise
Initialisation:	for condition see ‘Application Incidences of Process Monitoring’
Update Rate:	40 ms

Formula section:

```
If      (LV_IM_BLS_MON = 1)       Or
        (LV_IM_BTS_MON = 1)       Or
        (LV_ERR_BLS_PLAUS = 1)
Then    LV_BRAKE_MON = 1
Else    LV_BRAKE_MON = 0
```

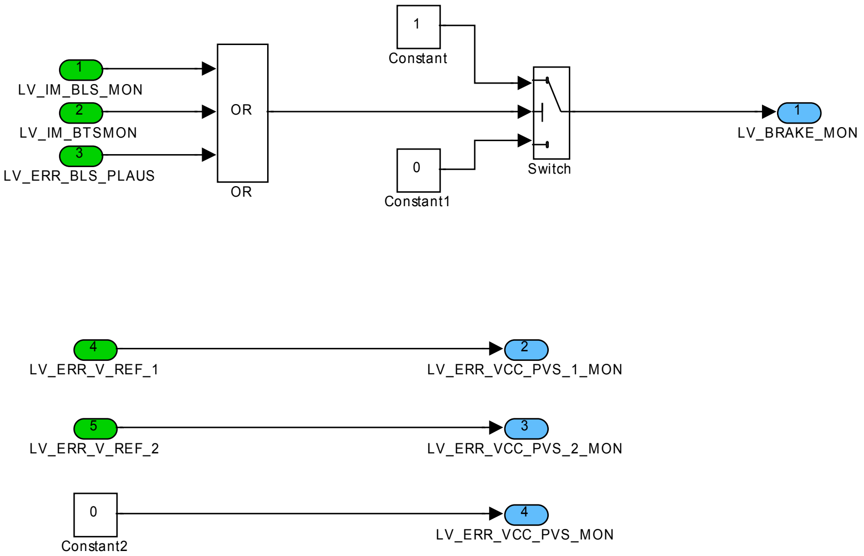
Endif

Supply voltage error bits:

LV_ERR_VCC_PVS_1_MON = LV_ERR_V_REF_1

LV_ERR_VCC_PVS_2_MON = LV_ERR_V_REF_2

LV_ERR_VCC_PVS_MON = 0



11.6.5 Adaptation of OPM_AV to OPM_AV_MON

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
OPM_AV_MON	O/V	0H 1H 2H 3H 8H	- S AFS AFL LIH	1	[-]
actual engine operation mode (process monitoring)					
LV_ERR_OPM_AV_MON	O/V	0...1H	0...1	1	[-]
Error in engine operation mode					
CTR_OPM_AV_PLAUS_MON	V	0...FFH	0...255	1	[-]
Debounce counter for error detection of OPM_AV					

Input data:

LAMB_MON	OPM_AV	FAC_TQ_REQ_MON	C_THD_MAX_LAMB_MON
EFF_SCC_MON			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_SWI_OPM_AV_MON	1	0...1H	0...1	1	[-]
switch between OPM_AV (level 1) and predetermined state					
C_SWI_OPM_AV_MON	1	0...3H	0...3	1	[-]
manual demand OPM_AV_MON					
C_L_RNG_THD_OPM_AV_PLAUS_MON	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Low threshold to detect unplausible OPM_AV					
C_H_RNG_THD_OPM_AV_PLAUS_MON	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
High threshold to detect unplausible OPM_AV					
C_THD_MIN_FAC_TQ_REQ_MON	1	0...FFH	0...1.99218	0.0078125	[-]
Minimum threshold for driver request in order to detect PL					
LC_OPM_AV_PLAUS_ENA_MON	1	0...1H	0...1	1	[-]
Enable plausibilisation of OPM_AV					
C_INC_OPM_AV_PLAUS_MON	1	0...FFH	0...255	1	[-]
Debounce increment for plausibilisation of OPM_AV error					
C_DEC_OPM_AV_PLAUS_MON	1	0...FFH	0...255	1	[-]
Debounce decrement for plausibilisation of OPM_AV error					
C_OPM_AV_PLAUS_MAX_MON	1	1...FFH	1...255	1	[-]
Threshold for detection of OPM_AV error					

General information:

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialisation: for condition see ‘Application Incidences of Process Monitoring’

And at LV_TQI_MON_ACT_MON 1 -> 0 OPM_AV_MON is set to AFS

Update Rate: 40 ms

FUNCTION DESCRIPTION:

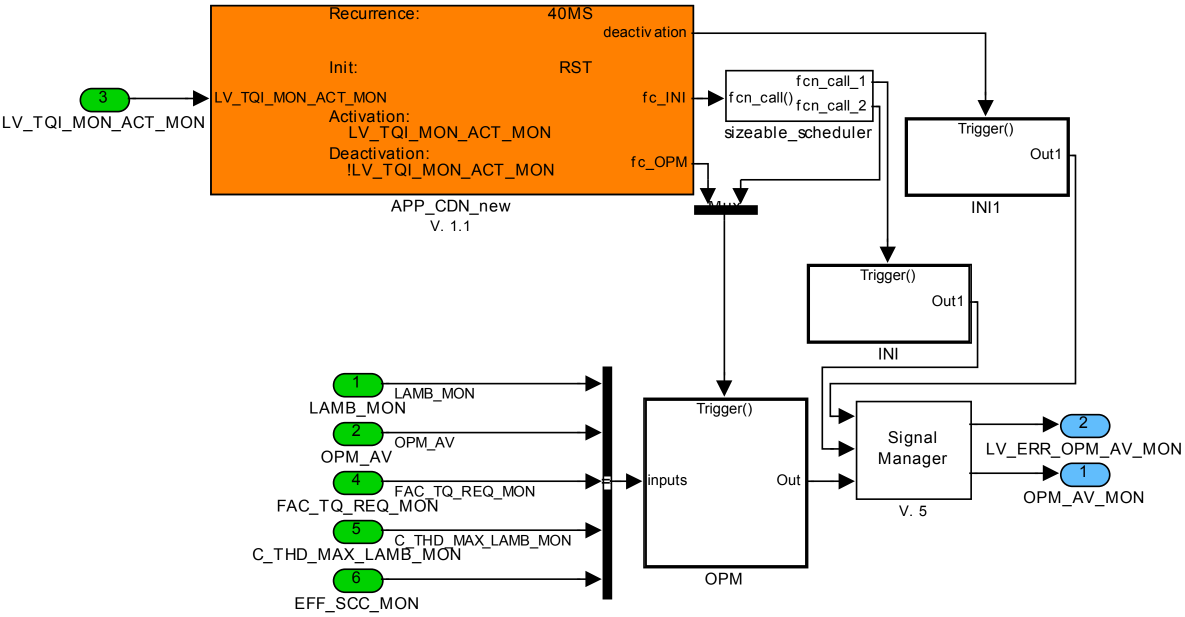


Figure 1: ECM2_DTSYSMODULE

11.6.5.1.1 SUBFUNCTION: INIT

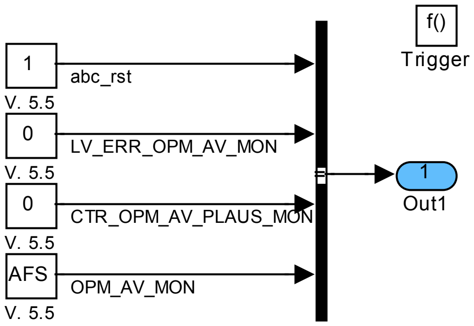


Figure 2: ECM2_DTSYSMODULE/INIT

11.6.5.1.2 SUBFUNCTION: INIT1

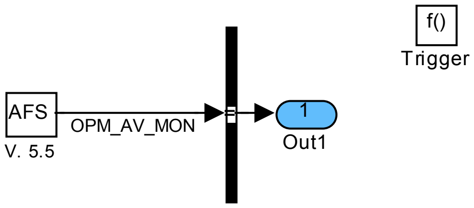


Figure 2: ECM2_DTSYSMODULE/INIT1

11.6.5.1.3 SUBFUNCTION: OPM

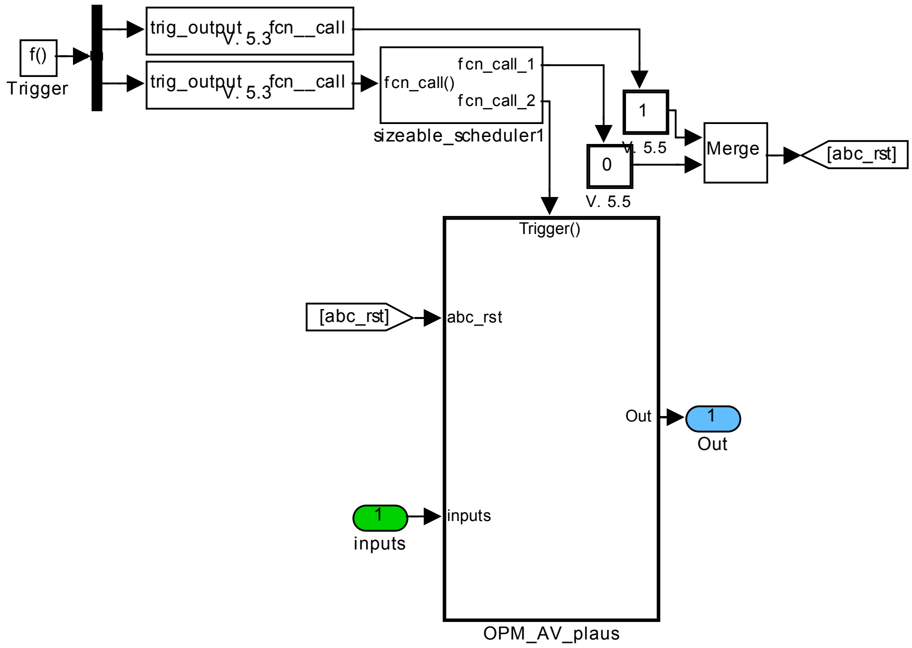


Figure 4: ECM2_DTSYSMODULE /OPM

11.6.5.1.4 SUBFUNCTION: OPM_AV_plaus

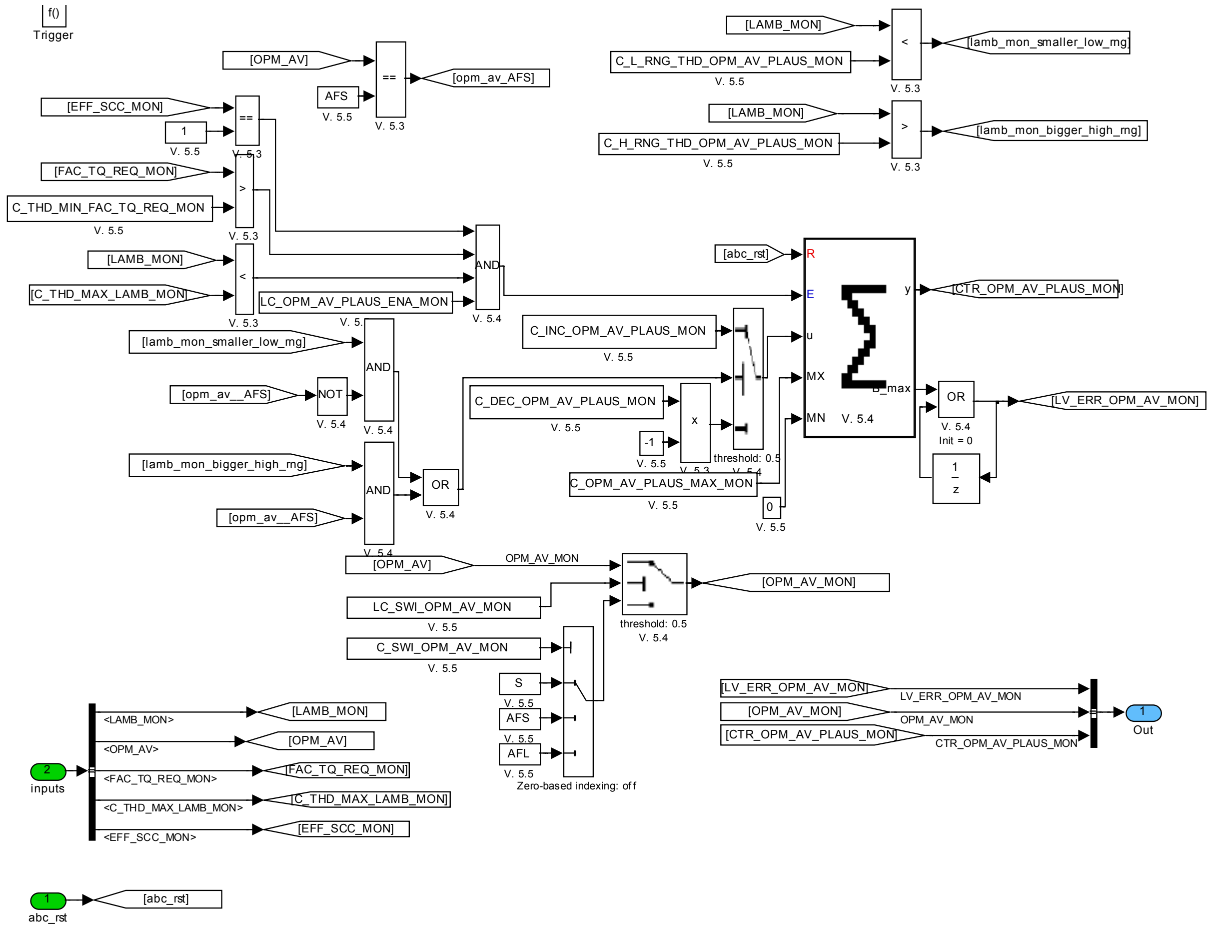


Figure 5: ECM2_DTSYSMODULE /OPM/ OPM_AV_PLAUS

11.6.6 Application incidences for the monitoring of torque losses

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_LOSS_ADD_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Additional torque losses					
TQ_LOSS_ACC_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Air conditioning compressor torque losses					
TQ_LOSS_ALTER_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Alternator torque losses					
TQ_LOSS_ALTER_MDL_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Modelled alternator torque losses					
TQ_LOSS_ARS_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Anti roll stabilisation torque losses					
TQ_LOSS_PSTE_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
steering wheel compressor torque losses					
TQ_LOSS_PSTE_1_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
steering wheel compressor torque losses					
TQ_LOSS_PSTE_2_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
steering wheel compressor torque losses					
CUR_ALTER_MON	V	0...FFH	0...255	1	[A]
currency torque losses monitoring					
N_ALTER_MON	V	0...FFFFH	0...65535	1	[rpm]
currency torque losses monitoring					
DFFGEN_FIL_MON	V	0...FFH	0...99.60937	0.390625	[%]
currency torque losses monitoring					
TQ_LOSS_SA_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
secundary air torque losses					
VS_MON	O/V	0...FFFFH	0...511.99218	0.0078125	[km/h]
vehicle speed for monitoring PSTE_2					
TQ_LOSS_PSTE_3_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
power steering torque losses pste_3					
TQ_LOSS_ADD_FIL_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Filtered additional torque losses					
TQ_REQ_CLU_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Monitored driver requested torque at clutch*					
TQ_REQ_CLU_GRD_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Gradient of driver requested torque at clutch (monitoring)					
TQ_LOSS_FIL_MMV_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Filtered additional torque losses (monitoring)					

Input data:

LV_VAR_PSTE_2	TQ_LOSS_ALTER	POW_REL_ALTER_CLC	LV_ARS_ENA

TQ_LOSS_ARS	LV_PSTE_ENA	TQ_LOSS_PSTE_1	LV_PSTE_2_ENA
N_32_MON	TQ_LOSS_PSTE_2	LV_ERR_GEN	LV_ERR_ALTER_COM
LV_ERR_GEN_DIAG	LV_ERR_BSD	LV_SAP	ANG_PSTE_STND
TQ_LOSS_ACC	TQ_LOSS_PSTE_3	LV_PSTE_3_ENA	LV_VAR_PSTE_3
TQ_LOSS_ADD_FIL	TQ_REQ_CLU		VS_FIL
VS_SENS	LV_ERR_VS	LV_ERR_VS_CAN	LV_ERR_BN_VS_TCS

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQ_LOSS_PSTE_2_MIN_MON	6*6	0...8000H	-1024...0	0.03125	[Nm]
LDP_VS_MON__TQ_LOSS_MON	6	0...FFFFH	0...511.99218	0.0078125	[km/h]
LDP_ANG_PSTE_STND__TQ_LOSS_MON	6	0...FFFFH	-100...99.99694	3.0518e-3	[%]
minimum additional torque loss for PSTE_2 active					
C_TQ_LOSS_PSTE_1_MIN_MON	1	8000...0H	-1024...0	0.03125	[Nm]
Minimum additional torque loss for PSTE_1 active					
C_TQ_LOSS_ALTER_MIN_MON	1	8000...0H	-1024...0	0.03125	[Nm]
Minimum additional torque loss for alternator					
C_TQ_LOSS_ADD_ACC_MIN_MON	1	8000...0H	-1024...0	0.03125	[Nm]
Minimum additional torque loss for ACC active					
IP_CUR_ALTER_MON	6*6	0...FFH	0...255	1	[A]
LDP_DFFGEN_FIL_MON__LOSS_MON	6	0...FFH	0...99.60937	0.390625	[%]
LDPM_N_ALTER_MON__TQ_LOSS_MON	6	0...FFFFH	0...65535	1	[rpm]
Alternator torque losses					
IP_EFF_ALTER_MON	6*6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_N_ALTER_MON__TQ_LOSS_MON	6	0...FFFFH	0...65535	1	[rpm]
LDP_CUR_ALTER_MON__TQ_LOSS_MON	6	0...FFH	0...255	1	[A]
Alternator torque losses					
C_CRLC_ALTER_MON	1	0...FFH	0...0.99609	3.9063e-3	[-]
Alternator torque losses low pass					
C_FAC_EFF_ALTER_MON	1	0...FFH	0...7.96875	0.03125	[-]
Alternator torque losses low pass					
C_V_TQ_ALTER_MON	1	0...FFH	0...25.89843	0.1015625	[V]
Voltage range of monitoring					
C_FAC_N_ALTER_MON	1	0...FFH	0...7.96875	0.03125	[-]
-					
C_TQ_LOSS_ARS_MIN_MON	1	8000...0H	-1024...0	0.03125	[Nm]
Minimum additional torque loss for ARS active					
C_TQ_LOSS_SA_MIN_MON	1	8000...0H	-1024...0	0.03125	[Nm]
Minimum additional torque loss for SA active					
C_VS_DIF_MAX_MON	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
minimum additional torque loss for SA active					
C_VS_LIH_MON	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
minimum additional torque loss for SA active					
IP_TQ_LOSS_PSTE_3_MIN_MON	6*6	0...8000H	-1024...0	0.03125	[Nm]
LDP_VS_MON__TQ_LOSS_PSTE3_MON	6	0...FFFFH	0...511.99218	0.0078125	[km/h]
LDP_ANG_PSTE_STND__TQ_PSTE3_MON	6	0...FFFFH	-100...99.99694	3.0518e-3	[%]
minimum additional torque loss for PSTE_3 active					
C_TQ_LOSS_PSTE_LIH_MON	1	8000...0H	-1024...0	0.03125	[Nm]
Default value torque losses if no PSTE enabled					
C_TQ_LOSS_ADD_FIL_DIF_MAX_MON	1	0...7FFFH	0...1023.96875	0.03125	[Nm]
Maximum allowed difference between L2 and L1 filtered additional torque losses					
LC_TQ_LOSS_ADD_FIL_MON	1	0...1H	0...1	1	[-]
Switch for TQ_LOSS_ADD_FIL_MON calculation					
IP_FAC_TQ_LOSS_ADD_FIL_MON	8*8	0...FFH	0...0.99609	3.9062e-3	[-]
LDP_TQ_REQ_CLU__ADD_FIL_MON	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_TQ_REQ_CLU_GRD__ADD_FIL_MON	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
Filter constant determination					
C_TQ_LOSS_FIL_MMV_INI_MIN_MON	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Minimum filtered additional torque losses used for initalisation of filtering function					

FUNCTION DESCRIPTION:

General information:

In order to be able to use module for the monitoring of torque losses in level 2 the not engine related additional torque losses are calculated here.

The level 1 value of each additional torque loss will be compared with a modelled one. The larger value (minimum absolute Torque value) of the two is added to the sum TQ_LOSS_ADD_MON.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

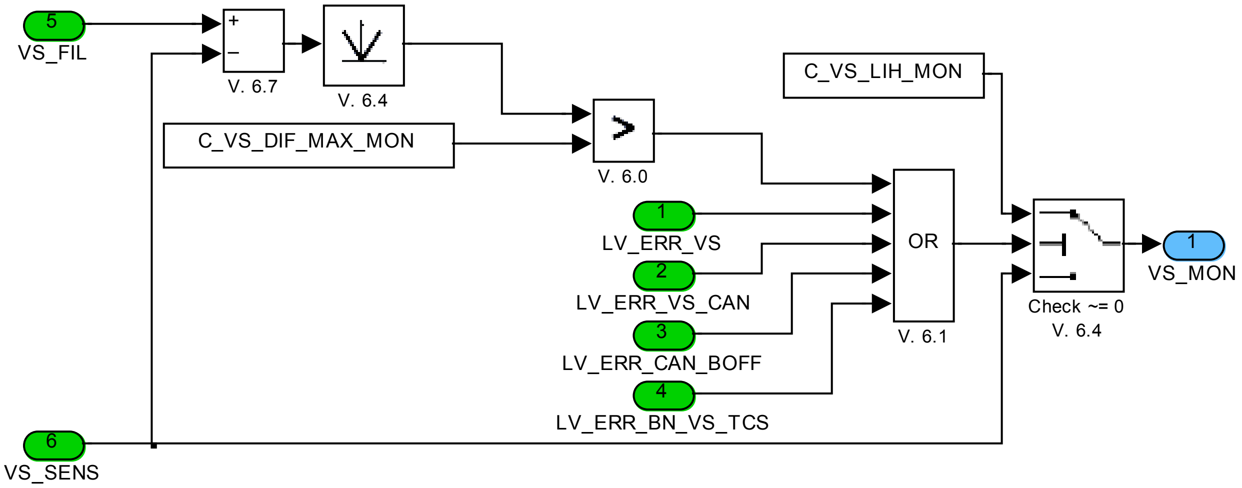
Deactivation: otherwise

Initialisation: for condition see 'Application Incidences of Process Monitoring'
TQ_LOSS_ADD_MON = 0
VS_MON = 0 km/h
TQ_LOSS_ADD_FIL_MON = 0
TQ_REQ_CLU_MON = 0
TQ_REQ_CLU_GRD_MON = 0
TQ_LOSS_FIL_MMV_MON =
max(TQ_LOSS_ADD_FIL , C_TQ_LOSS_FIL_MMV_INI_MIN_MON)

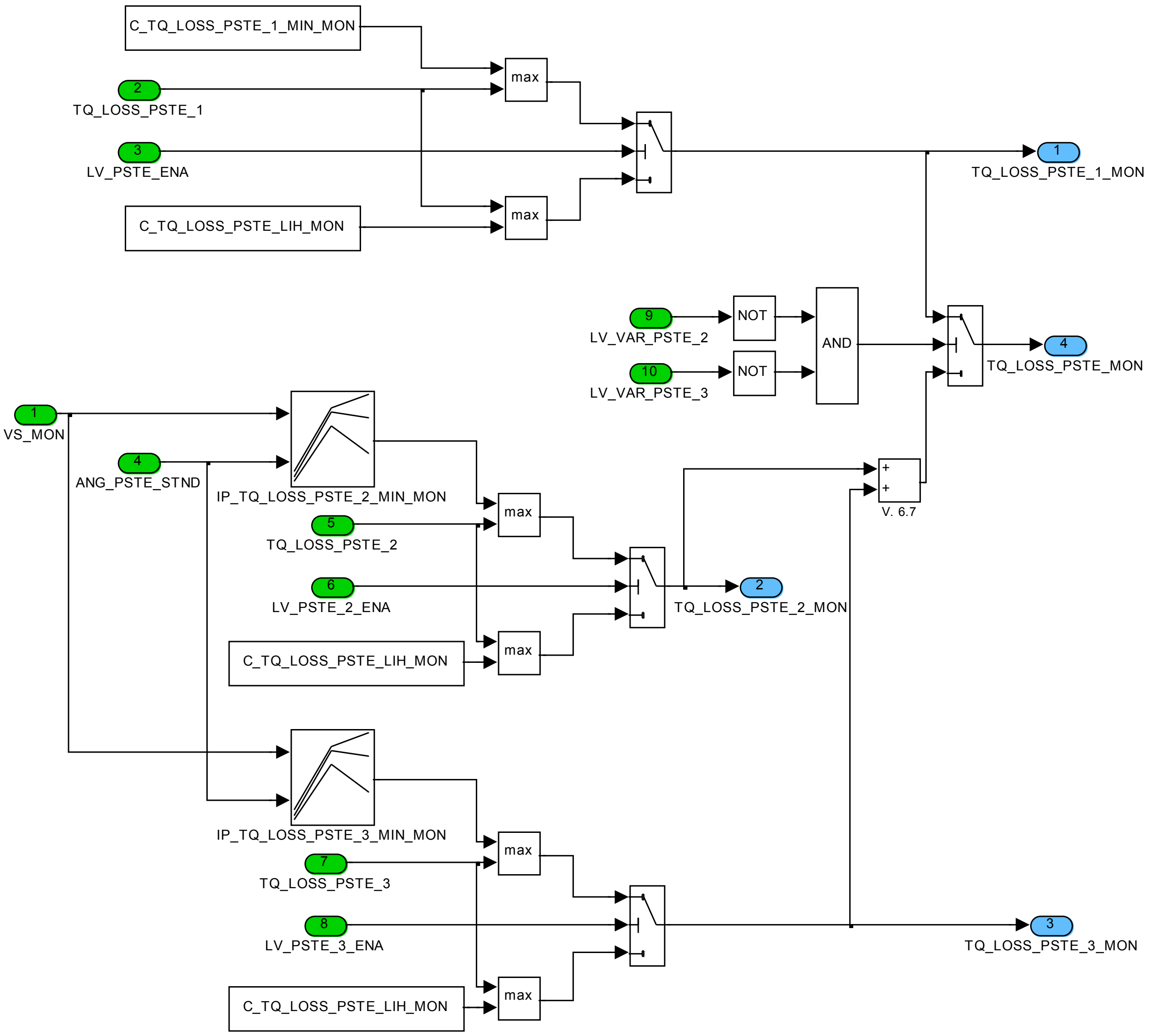
Recurrence: 40 ms

Signal flow diagram:

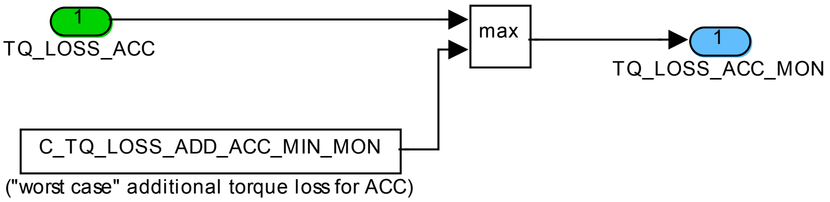
11.6.6.1 Monitoring of vehicule speed



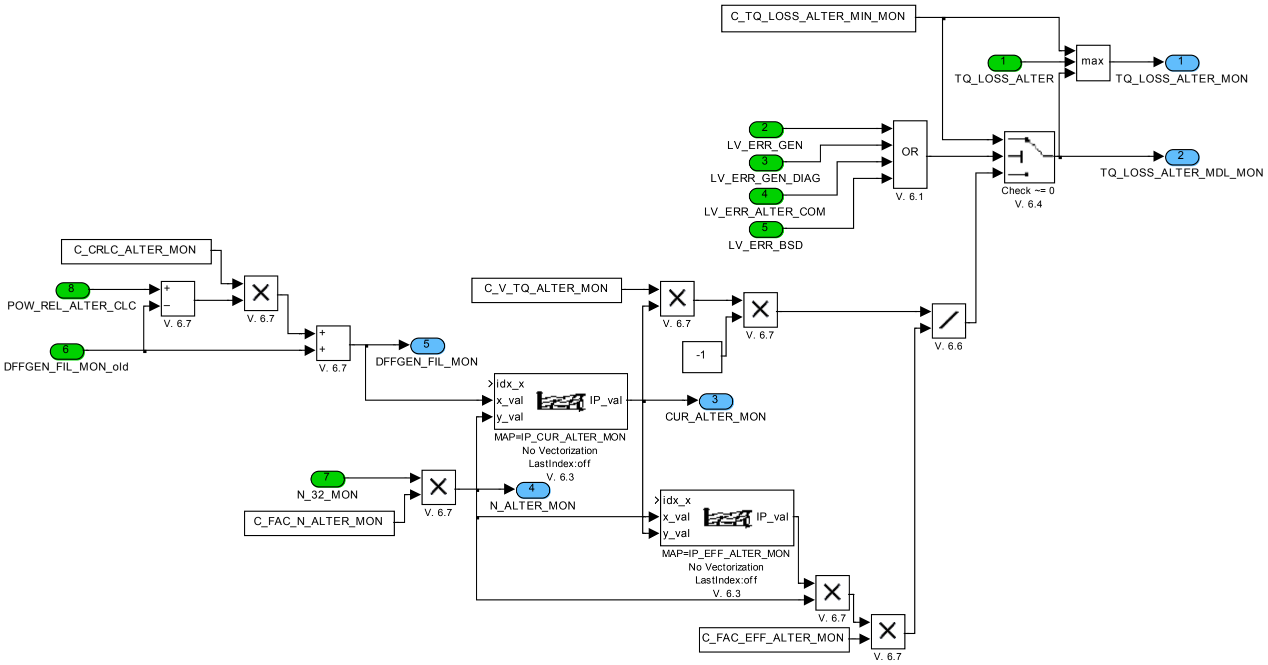
11.6.6.2 Monitoring of torque losses for powersteering



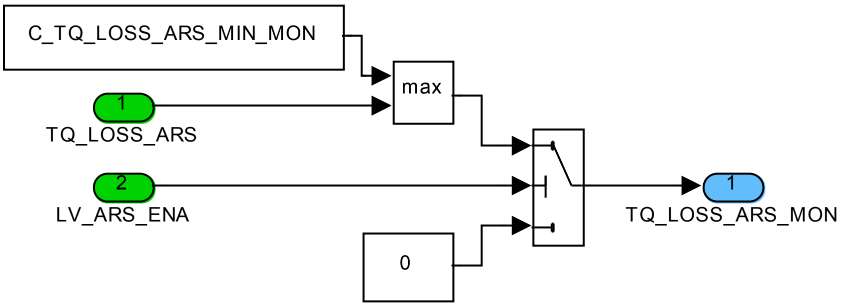
11.6.6.3 Monitoring of torque losses for ACC



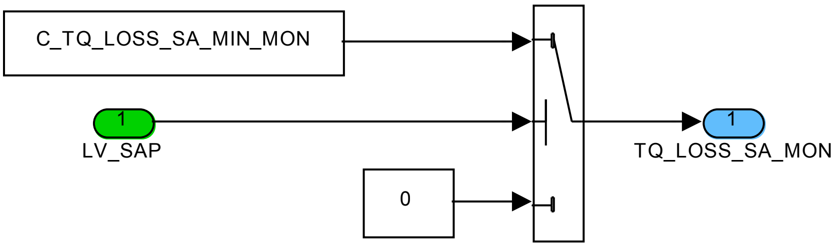
11.6.6.4 Monitoring of torque losses for alternator



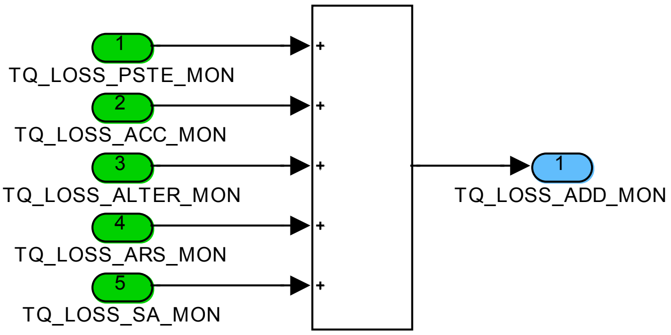
11.6.6.5 Monitoring of torque losses for anti roll stabilisation



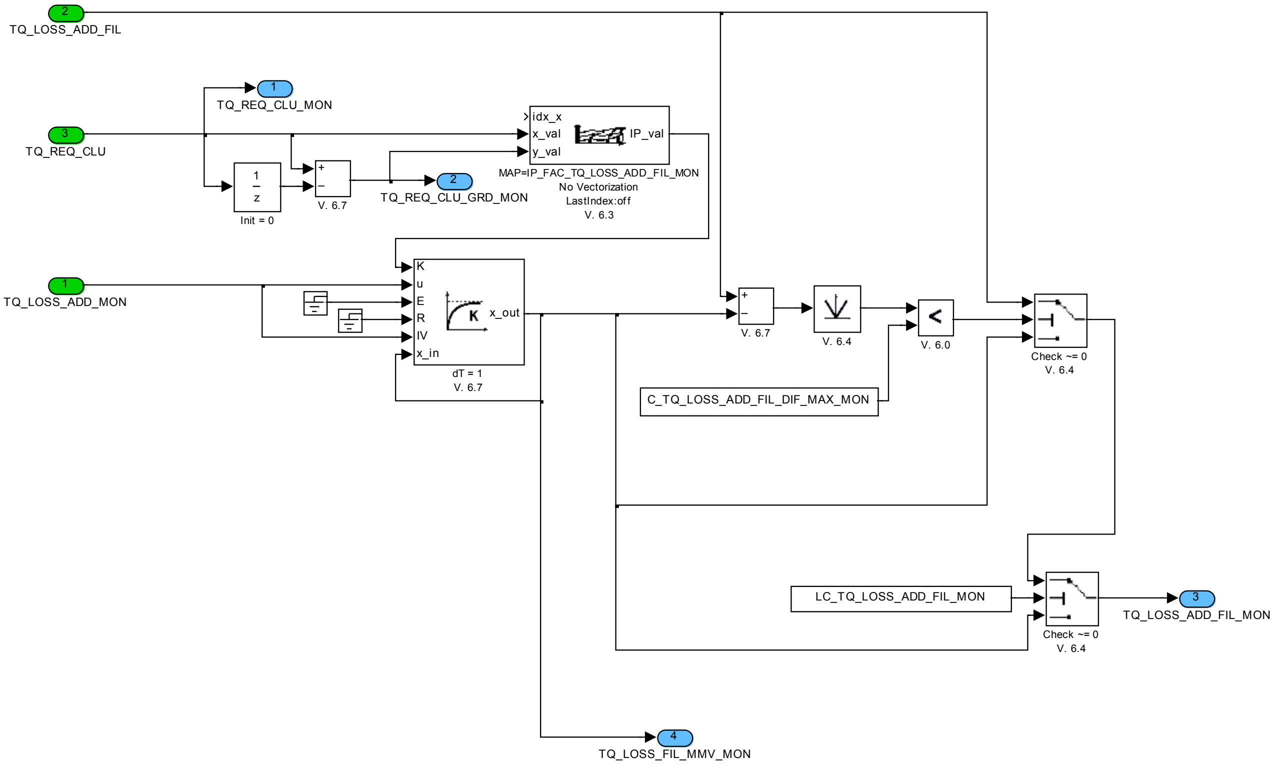
11.6.6.6 Monitoring of torque losses for secundary air



11.6.6.7 Calculation of the additionnal torque losses



11.6.6.8 Calculation of the filtered additionnal torque losses



11.6.7 Application incidences for the actual lambda deviation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAMB_LS_UP_VLD_MON[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Boolean flag indicating that lambda signal from WRAF system is reliable and up-to-date (process monitoring)					

Input data:

LV_LAMB_LS_UP_VLD[NC_CBK_EX_NR]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DLY_LAMB_LS_UP_NOT_VLD_MON	1	0...FFH	0...255	1	[-]
Delay (in samples) for switching LV_LS_UP_VLD_MON from1 to 0					

11.6.7.1 ECM2_DTSYSLADIF

General information:

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialisation: for condition see ‘Application Incidences of Process Monitoring’

Update Rate: 40 ms

FUNCTION DESCRIPTION:

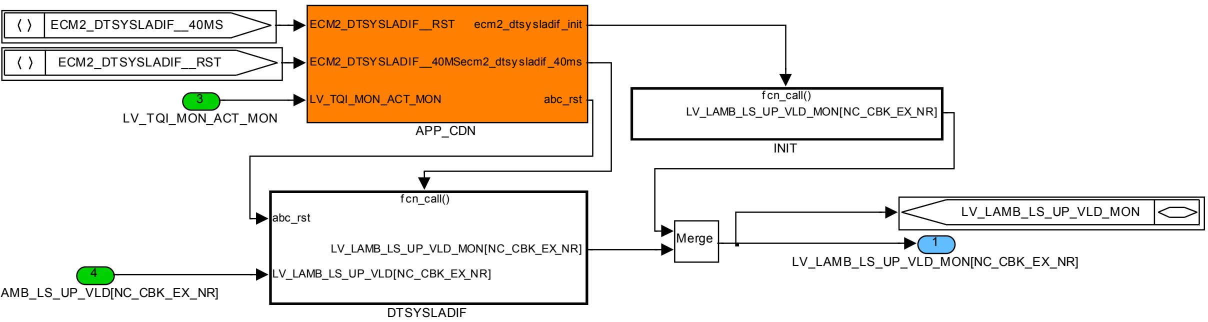


Figure 1: ECM2_DTSYSLADIF

11.6.7.1.1 SUBFUNCTION: INIT

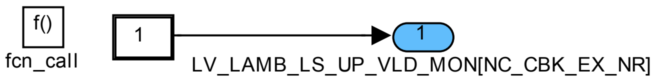


Figure 2: ECM2_DTSYSLADIF/INIT

11.6.7.1.2 SUBFUNCTION: determination_of_act_lamb_dev_value

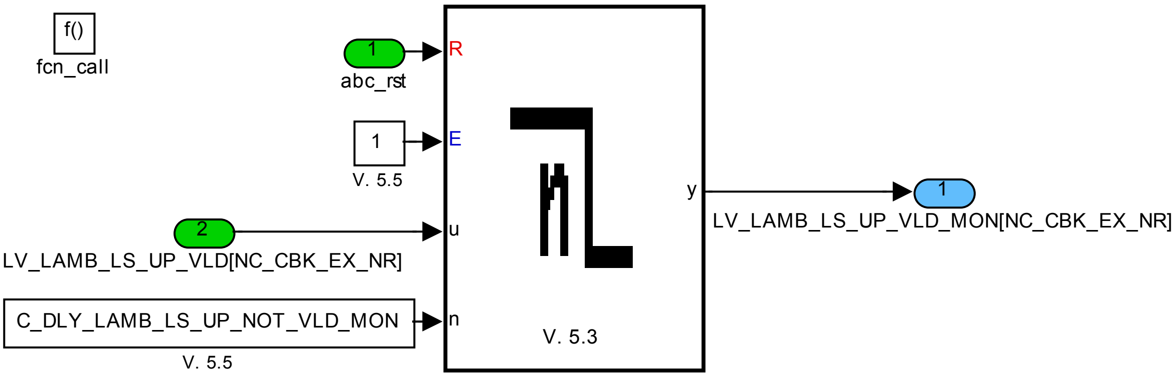


Figure 3: ECM2_DTSYSLADIF/determination_of_act_lamb_dev_value

11.6.8 Forced injection shut-off for Level 2 - engine speed limitation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_OFF_IV_N_LIM_ETC_MON	O/V	0...1H	0...1	1	[-]
Reversible fuel Cut Off Switch for all cylinders					

Input data:

N_32_MON			

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_N_THD_FCUT_N_LIM_ETC_MON	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for forced injection cut off in SAS mode					

FUNCTION DESCRIPTION:

General information:

In order to ensure a safe injection cut off while engine speed limitation is active (LV_N_LIM_ETC_MON == 1) and engine speed is above the threshold C_N_THD_FCUT_N_LIM_ETC_MON, the resulting logical variable LV_OFF_IV_N_LIM_ETC_MON is used at individual cylinder shut off functionality (same use and priority as LV_OFF_IV_MON).

Application conditions:

Activation: at every engine state

Deactivation: otherwise

Initialisation: for condition see ‘Application Incidences of Process Monitoring’

LV_OFF_IV_N_LIM_ETC_MON = 0

Update Rate: 40 ms

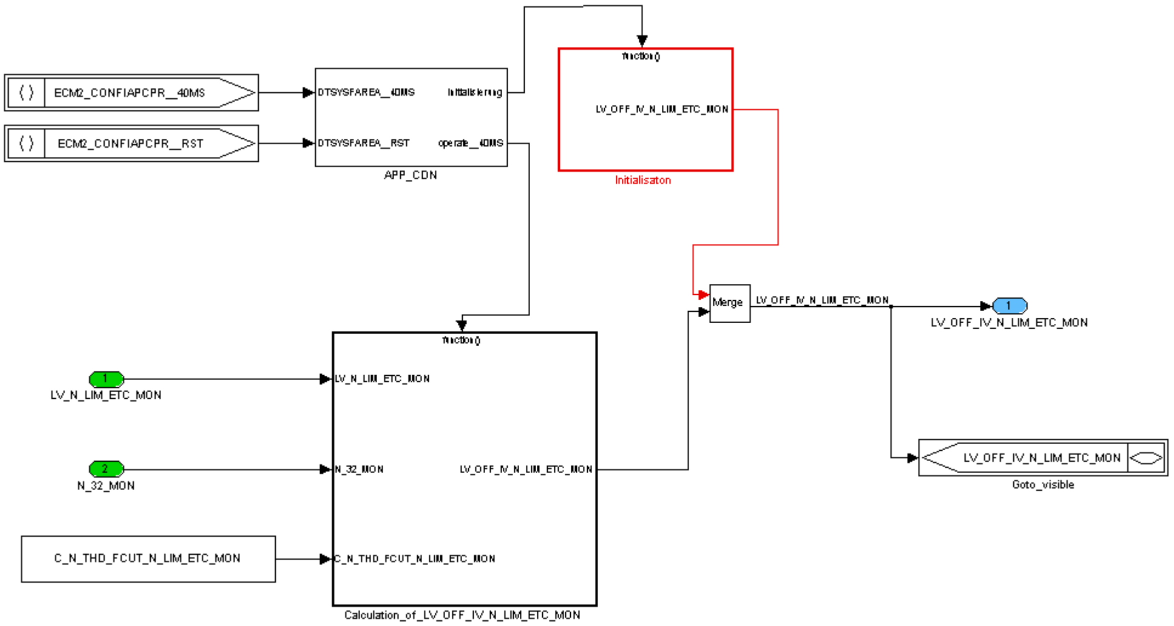


Figure 1: CONFIAPCPR

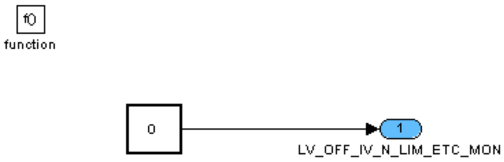


Figure 2: CONFIAPCPR /Initialisation

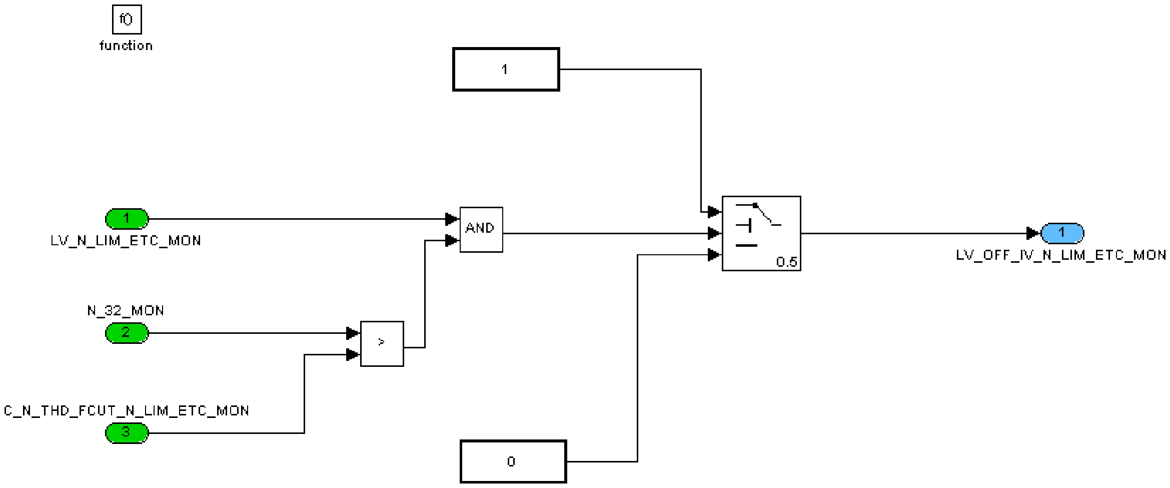


Figure 3: CONFIAPCPR/Calculation_of_LV_OFF_IV_N_LIM_ETC_MON

11.6.9 State of torque intervention when level 2/3 error is detected

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SF_TQD_MON	O/V	0...1H	0...1	1	[-]
Flag indicating level 2/3 error					

Input data:

LV_ERR_CONV_MON	LV_ERR_MSR_INH_MON	LV_ERR_N_32_MON	LV_ERR_PVS_MON
LV_ERR_TQI_N_MAX_MON	LV_ERR_MFF_MON	LV_ERR_CRU_INH_MON	LV_ERR_TQI_AV_MON
LV_ERR_AMT_INH_MON	LV_ERR_MU_MC	LV_ERR_DCC_INH_MON	LV_ERR_LDM_INH_MON
LV_ERR_GS_INH_MON			

General information:

When a level 2/3 error is present except “monitoring of cruise control” the CAN status flag SF_TQD = 3 is set via LV_SF_TQD_MON = 1:

Formula section:

Error bit	SF_TQD
LV_ERR_CONV_MON	LV_SF_TQD_MON = 1
LV_ERR_MSR_INH_MON	LV_SF_TQD_MON = 1
LV_ERR_N_32_MON	LV_SF_TQD_MON = 1
LV_ERR_PVS_MON	LV_SF_TQD_MON = 1
LV_ERR_MFF_MON	LV_SF_TQD_MON = 1
LV_ERR_CRU_INH_MON	LV_SF_TQD_MON = 0
LV_ERR_TQI_AV_MON	LV_SF_TQD_MON = 1
LV_ERR_TQI_N_MAX_MON	LV_SF_TQD_MON = 1
LV_ERR_MU_MC	LV_SF_TQD_MON = 1
LV_ERR_DCC_INH_MON	LV_SF_TQD_MON = 1
LV_ERR_AMT_INH_MON	LV_SF_TQD_MON = 1
LV_ERR_GS_INH_MON	LV_SF_TQD_MON = 1
LV_ERR_LDM_INH_MON	LV_SF_TQD_MON = 1

11.6.10 Application incidences for the cruise control input

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CRU_INH_MON	O/V	0...1H	0...1	1	[-]
Logival variable for inhibition of cruise control					

Input data:

LV_ERR_CRU_MON			

Formula section:

LV_ERR_CRU_INH_MON = LV_ERR_CRU_MON

11.6.11 Application incidences for the IPM-monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
GR_DT_MON	O/V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Gear ratio factor for monitoring					
GR_DT_MDL_MON	V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Modelled gear ratio for monitoring					

Input data:

N_32_MON	VS	VS_CAN	GR_DT

Calibration data:

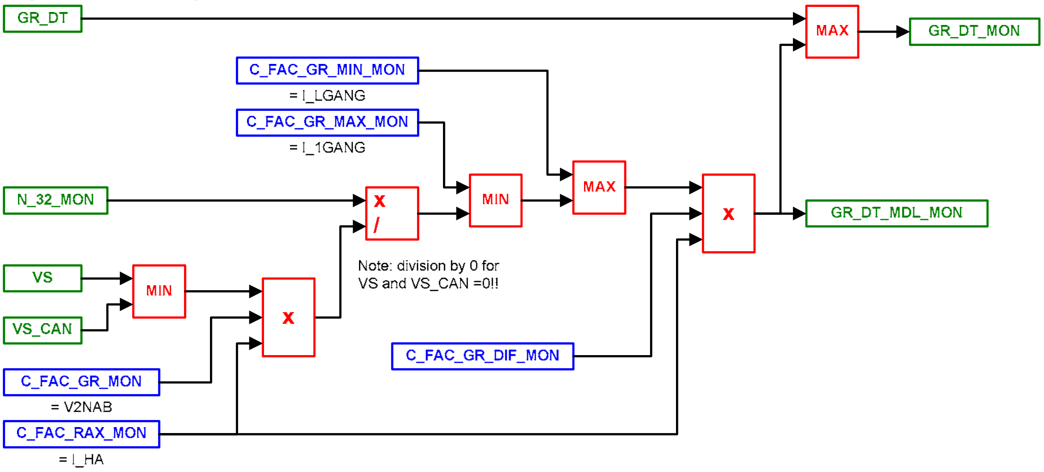
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_RAX_MON	1	0...FFH	0...7.96875	0.03125	[-]
Gear ratio rear axle					
C_FAC_GR_MON	1	0...FFH	0...2.55	0.01	[-]
1 / Wheel circumference					
C_FAC_GR_DIF_MON	1	0...FFH	0...1.99218	0.0078125	[-]
Max. permitted difference in gear ratio					
C_FAC_GR_MIN_MON	1	0...FFH	0...31.875	0.125	[-]
Min. value for gear ratio (longest gear step)					
C_FAC_GR_MAX_MON	1	0...FFH	0...31.875	0.125	[-]
Max. value for gear ratio (shortest gear step)					

FUNCTION DESCRIPTION:

General information:

For the monitoring of the torque request from IPM (intelligent powertrain management) it is necessary to determine the gear ratio factor GR_DT_MON.

Signal flow diagram:



Application conditions:

Activation:	LV_TQI_MON_ACT_MON = 1
Deactivation:	otherwise
Initialisation:	for condition see ‘Application Incidences of Process Monitoring’
	GR_DT_MON = 1
	GR_DT_MDL_MON = 1
Update Rate:	40 ms

Formula section:

Gear ratio of gear box:

GR_DT_MDL_MON_1 = (N_32_MON / 60) / *N_32 in [1/sec]*
((MIN (VS ; VS_CAN) / 3.6) * *VS in [m/sec]*
C_FAC_GR_MON * *1/wheelcircumference*
C_FAC_RAX_MON) *gear ratio rear axle*

Note: for VS and VS_CAN = 0 there is a devision by zero!

Limitation of gear ratio of gear box:

GR_DT_MDL_MON_2 = MAX (C_FAC_GR_MIN_MON ; *limit longest gear*
MIN (C_FAC_GR_MAX_MON; *limit shortest gear*
GR_DT_MDL_MON_1))

Modelled total gear ratio of powertrain

GR_DT_MDL_MON = GR_DT_MDL_MON_2 *
C_FAC_GR_DIF_MON *
C_FAC_RAX_MON

Powertrain gear ratio used for monitoring:

GR_DT_MON = MAX (GR_DT ; GR_DT_MDL_MON)

11.6.12 Application incidences for the monitoring of CAN signals

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQI_INC_CAN_MON	O/V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Torque demand from external CAN signals					
LV_MSR_ACT_MON	O/V	0...1H	0...1	1	[-]
Logical variable for torque intervention due to MSR for monitoring use					

Input data:

LV_VAR_BN	LV_MSR_ACT	TQI_MSR_MON	TQI_AMT_MON
TQI_GS_MON	TQ_MSR_MON	TQ_DCC_MON	TQ_WHEEL_LDM_MON
GR_DT_MON	TQ_AMT_MON	TQ_GS_MON	TQ_LOSS_MON

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialisation: for condition see ‘Application Incidences of Process Monitoring’

Recurrence: 40ms

Formula section:

Plausibilisation of active MSR

```
IF        LV_MSR_ACT = 1 AND (TQI_MSR_MON > 0 OR TQ_MSR_MON > 0)
THEN      LV_MSR_ACT_MON = 1
ELSE      LV_MSR_ACT_MON = 0
ENDIF
```

Torque demand from external CAN signals

```
IF(1)     LV_VAR_BN = 0
THEN(1)   TQI_INC_CAN_MON = max (TQI_MSR_MON, TQI_AMT_MON, TQI_GS_MON)
ELSEIF(1) (LV_VAR_BN = 1)
          IF(2)  TQ_MSR_MON = 0              AND
                 TQ_DCC_MON = 0              AND
                 TQ_WHEEL_LDM_MON = 0        AND
                 TQ_AMT_MON = 0              AND
                 TQ_GS_MON = 0
          THEN(2)     TQI_INC_CAN_MON = 0
          ELSE(2)     TQI_INC_CAN_MON =
                      [max   (TQ_MSR_MON, TQ_DCC_MON,
```

TQ_WHEEL_LDM_MON / GR_DT_MON, TQ_AMT_MON,

TQ_GS_MON)] – TQ_LOSS_MON

ENDIF(2)

ENDIF(1)

11.6.13 Application incidences for the monitoring of minimum torque at clutch

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
STATE_ERR_DET_TQ_MIN_MON	O/V	0H 1H 2H	PU NON_PU PASSIVE	1	[-]
Error-detection-state for monitoring of minimum torque at clutch					
TQ_ADD_MIN_CLU_MON	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Additional torque request from external sources					
TQ_CONV_MAX_MDL_MON	V	0...FFH	0...510	2	[Nm]
Maximum converter torque losses					
TQ_CONV_MAX_MON	O/V	0...FFH	0...510	2	[Nm]
Monitored value of converter torque losses					
LV_SWI_TQ_MIN_CLU_MON	O/V	0...1H	0...1	1	[-]
Idle speed setpoint from external influences					

Input data:

LV_AT	LV_CT_MON	TQ_CONV	C_N_DIF_HYS_IS_PU_MON
EFF_SCC_MON	C_N_DIF_MAX_IS_MON	TCO_MON	LV_PUC
LV_VAR_TCT	TQ_REQ_CLU_GB		

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
IP_FAC_CONV_MON	6	0...FFFFH	0...0.00048	7.4464e-9	[Nm/rpm²]
LDP_TCO_MON_IP_FAC_CONV_MON	6	0...FEH	-48...142.5	0.75	[°C]
Converter torque characteristics					
LC_INH_PU_REQ_MON	1	0...1H	0...1	1	[-]
Logical bit to inhibit LV_PU in calculation of TQ_MIN_CLU_MDL_MON					
C_TQ_DIF_IS_AD_CONV_MAX_MON	1	0...FFH	0...510	2	[Nm]
Max. limitation for converter torque adaptation					
C_TQ_REQ_CLU_GB_MAX_MON	1	0...FFH	0...510	2	[Nm]
Max. limitation for torque request from gearbox					

FUNCTION DESCRIPTION:

General information:

The objective of this function is the determination of the error-detection paths of the minimum torque at clutch. The error-detection is divided into PU state and non-PU state. This separation is done because the biggest influence on the value of the minimum torque that is available at the clutch is the self-stabilising factor FAC_N_SP_IS_RATIO_MON and this has a parabolic trajectory where the transition point from the positive values to TQ_LOSS_MON is around the idle-speed setpoint (N_SP_IS_MON). STATE_ERR_DET_TQ_MIN_MON is an input into the module 'Monitoring of minimum torque at clutch'.

Application conditions:

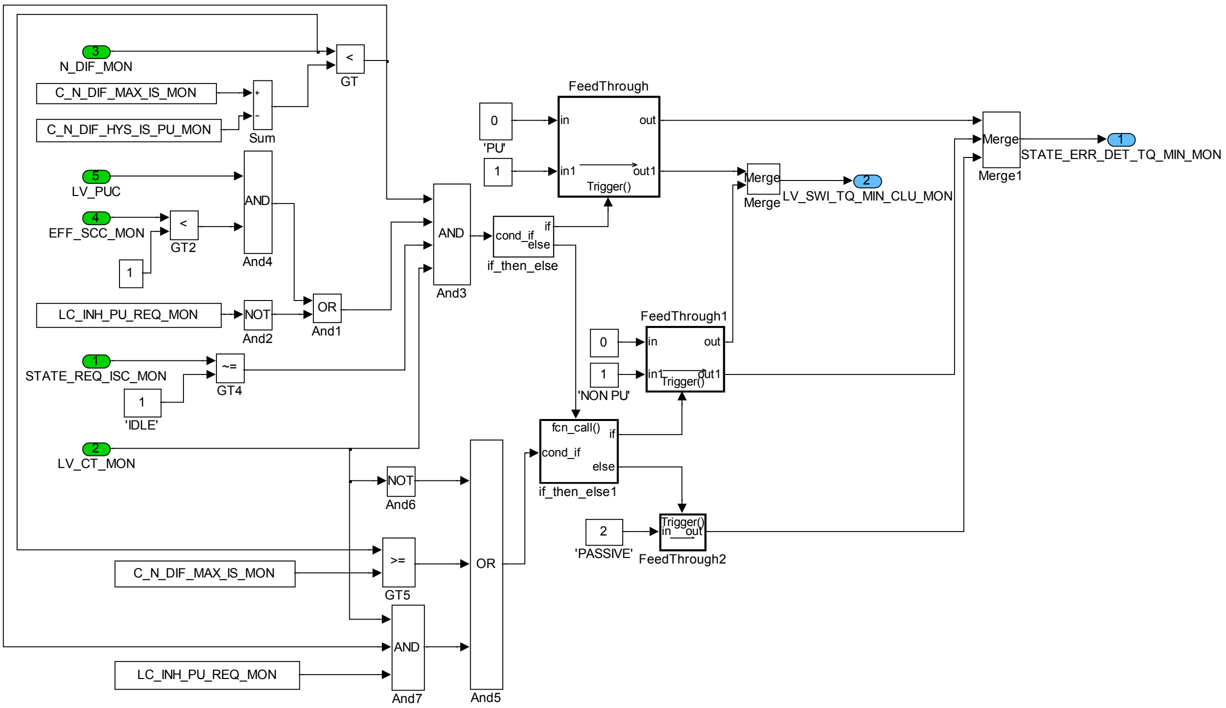
Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

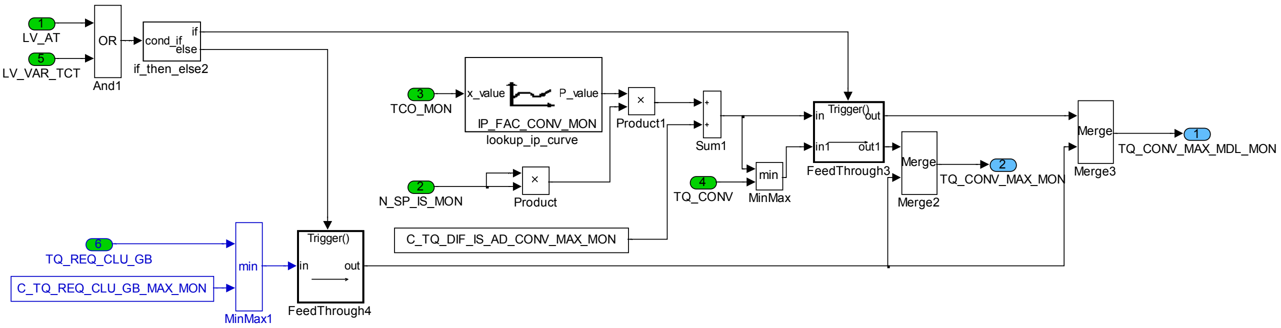
Initialisation: for condition see 'Application Incidences of Process Monitoring'
STATE_ERR_DET_TQ_MIN_MON = 'PASSIVE'
TQ_ADD_MIN_CLU_MON = 0 Nm
TQ_CONV_MAX_MON = 0 Nm
TQ_CONV_MAX_MDL_MON = 0 Nm
LV_SWI_TQ_MIN_CLU_MON = 0

Update Rate: 40 ms

Formula section:



{Calculation of Maximum Converter Torque
If a CVT system is used, then the torque demand/engine load resulting from the transmission has to be considered here. If this value is transmitted from the TCU via CAN, then it has to be checked first in the module 'Monitoring of CAN-signals' and only then used here. If a normal automatic transmission is used, then the torque losses resulting from the converter can be calculated as given below. The torque request of the manual gearbox is considered here. }



11.6.14 Application incidences for the actual efficiencies

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_REF_HOM_COR_EXT_MON	O/V	FA60...5A0H	-90...90	0.0625	[°CRK]
Correction of reference ignition angle in homogeneous (for external functions e.g. VVT, EGR...)					
IGA_REF_HOMS_COR_EXT_MON	O/V	FA60...5A0H	-90...90	0.0625	[°CRK]
Correction of reference ignition angle in homogeneous-stratified (for external functions e.g. VVT, EGR...)					
MFF_LAMB_REF_IGA_MON	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lambda relevant Mass Fuel Flow for stoichometric conditions (corrected by scavaging effect)					

Input data:

IGA_REF_LAMB_COR	IGA_REF_TEMP_COR	IGA_REF_EGR_HOM_COR	IGA_REF_EGR_HOMS_COR
MFF_LAMB_REF_MON	MAF_SCAV_EXT	C_FAC_MFF_MAF_MON	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_IGA_REF_TEMP_COR_LIM_MON	1	0...B40H	-90...90	0.0625	[°CRK]
Correction of reference ignition angle limitation TEMP					
C_IGA_REF_LAMB_COR_LIM_MON	1	0...B40H	-90...90	0.0625	[°CRK]
Correction of reference ignition angle limitation LAMB_SP					
C_IGA_REF_EGR_HOM_COR_LIM_MON	1	0...B40H	-90...90	0.0625	[°CRK]
Correction of reference ignition angle limitation EGR in HOM					
C_IGA_REF_EGR_HOMS_COR_LIM_MON	1	0...B40H	-90...90	0.0625	[°CRK]
Correction of reference ignition angle limitation EGR in HOMS					

11.6.14.1 CONFIAPCPR

General information:

The calculation of IGA_REF_HOM_COR_EXT_MON and IGA_REF_HOMS_COR_EXT_MON is an additive combination of limited values of level 1. The limitation of each single value allows having a close tolerance range for monitoring. Furthermore, MFF_LAMB_REF_IGA_MON has to be provided for the "Actual Efficiencies" as scavaging corrected MFF-value.

Application conditions:

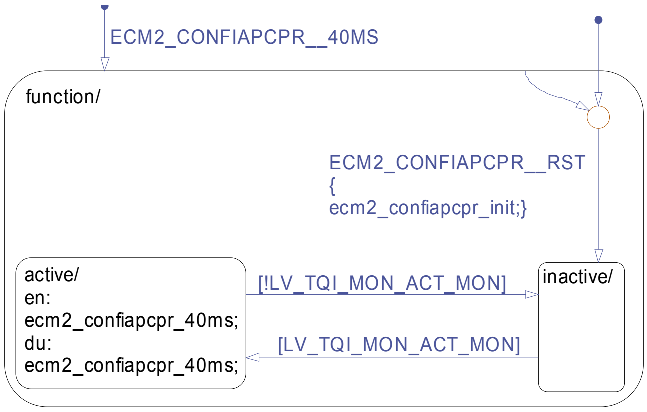


Figure 1: ECM2_CONFIAPCPR/APP_CDN_Chart

FUNCTION DESCRIPTION:

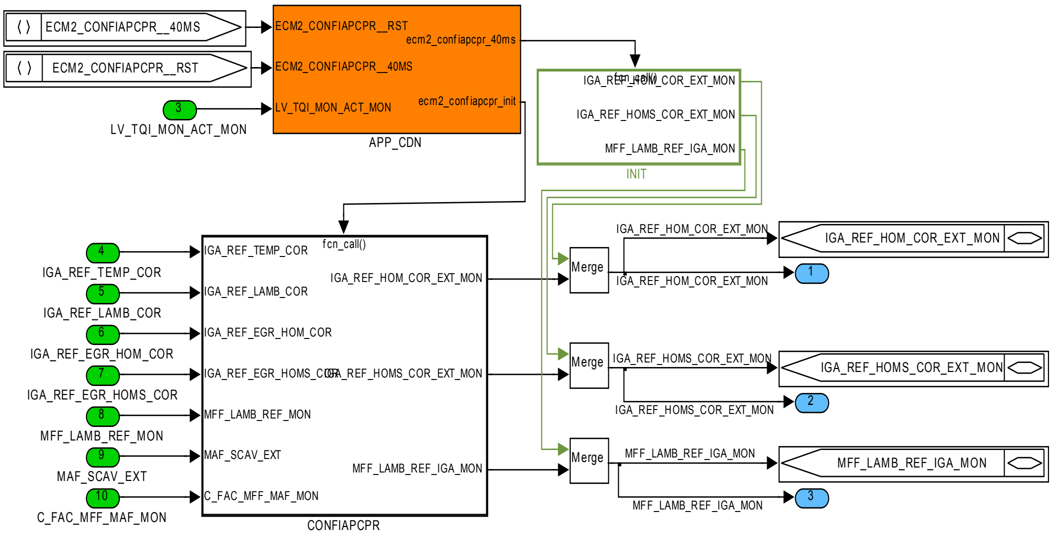


Figure 2: ECM2_CONFIAPCPR

11.6.14.1.1 SUBFUNCTION: INIT

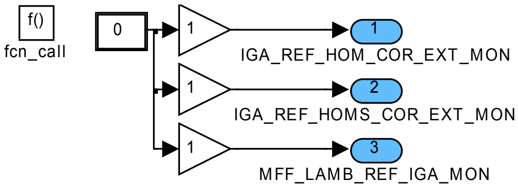


Figure 3: ECM2_CONFIAPCPR/INIT

11.6.14.1.2 SUBFUNCTION: CONFIAPCPR

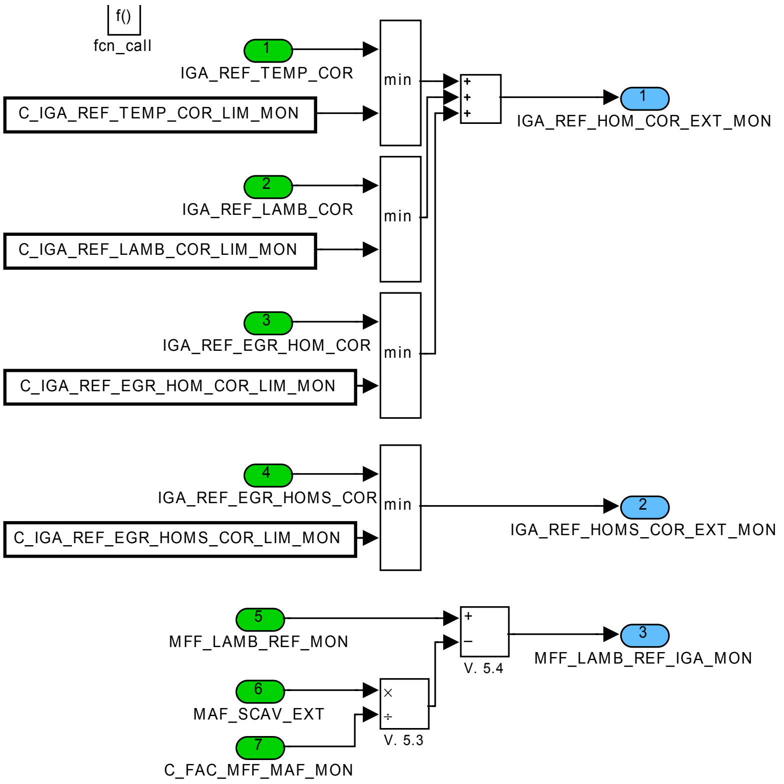


Figure 3: ECM2_CONFIAPCPR/CONFIAPCPR

11.6.15 Application incidences for the actual indicated engine torque

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EFF_MFF_TQ_COR_MON	O/V	0...FFH	0...1.99218	0.0078125	[-]
efficiency on torque relevant mass fuel flow					

Input data:

MFF_ADD_CYL_CP	MFF_ADD_TQ_CP	MFF_TQ_MON	FAC_MFF_TQ_COR_SCAV

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_FAC_MFF_ADD_CP_MIN_MON	1	0...FFH	0...1.99218	0.0078125	[-]
weighting factor to limit torque relevant MFF_CP in stratified					
C_FAC_MFF_TQ_COR_SCAV_MIN_MON	1	0...FFH	0...199.21875	0.78125	[%]
Minimal efficiency correction factor for scavenging					

11.6.15.1 CONFIAPCPR

General information:

Application conditions:

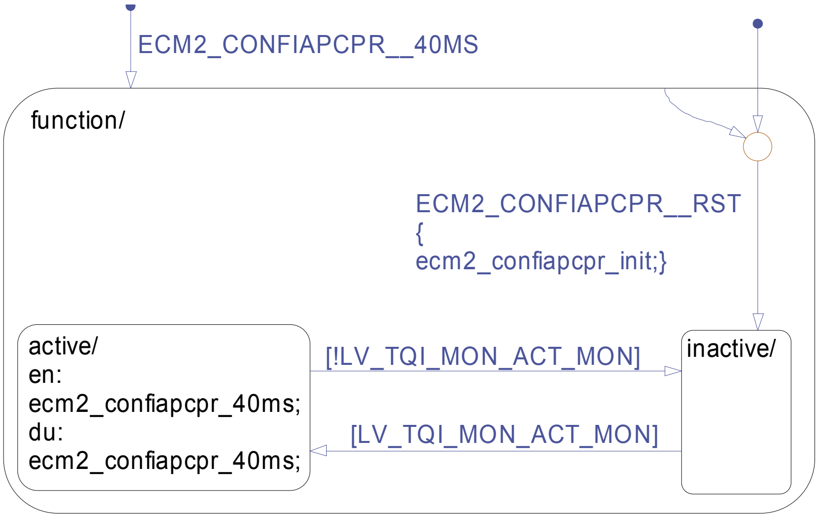


Figure 1: ECM2_CONFIAPCPR/APP_CDN_Chart

FUNCTION DESCRIPTION:

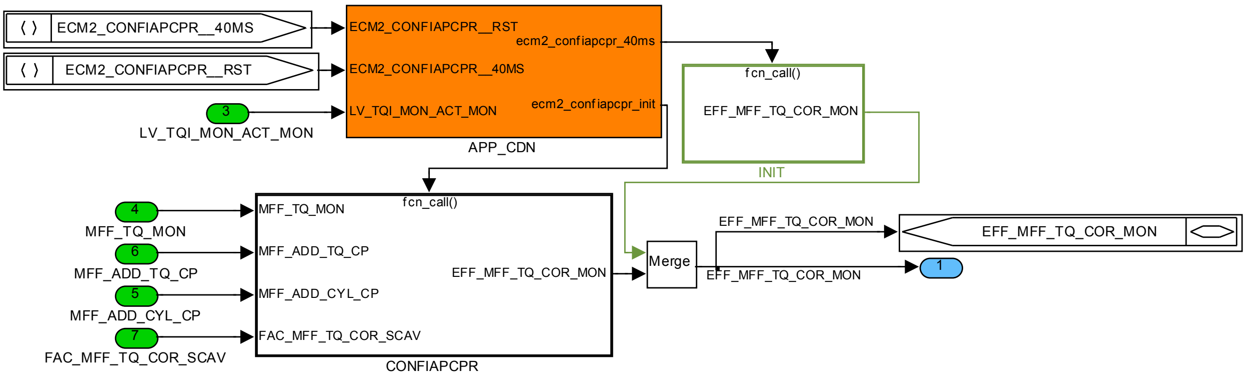


Figure 2: ECM2_CONFIAPCPR

11.6.15.1.1 SUBFUNCTION: INIT

Figure 3: ECM2_CONFIAPCPR/INIT

11.6.15.1.2 SUBFUNCTION: CONFIAPCPR

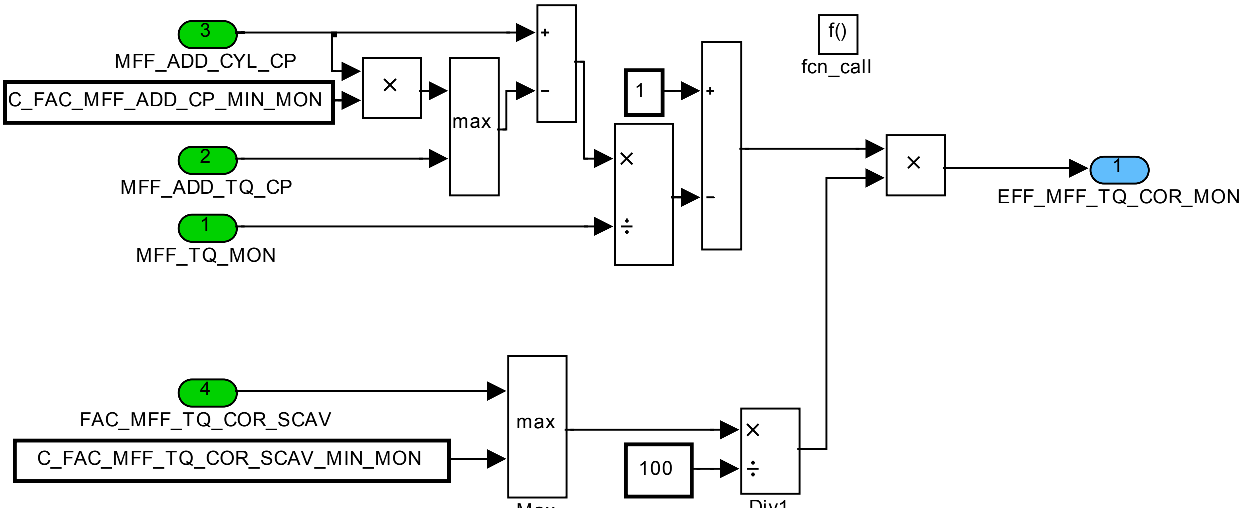


Figure 4: ECM2_CONFIAPCPR/CONFIAPCPR

11.6.16 Sport switch monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SOF_SWI_REQ_MON	O/V	0...1H	0...1	1	[-]
Sport-mode required by switch					
LV_SOF_SWI_MON	V	0...1H	0...1	1	[-]
Sport-mode switch acting					
LV_RNG_L_MON	O/V	0...1H	0...1	1	[-]
logical variable for engaged low range monitroring					
LV_RNG_L_AT_MON	O/V	0...1H	0...1	1	[-]
logical variable for engaged low range AT monitoring					
LV_SWI_FAC_TQ_REQ_MON	O/V	0...1H	0...1	1	[-]
logical variable for switching to other factors for requested torque at clutch					
LV_CDN_RNG_L_REQ_MON	V	0...1H	0...1	1	[-]
Condition for transition to low-range-mode in monitoring level					
LV_ERR_SOF_INH_MON	O/V	0...1H	0...1	1	[-]
error flag sport-mode switch request monitoring					
ABC_SOF_INH_MON	V	0...FFH	0...255	1	[-]
error flag sport-mode switch request monitoring					

Input data:

V_SOF_SWI_MON	LV_VAR_ETCU_SPT	LV_ETCU_SPT_SWI	LV_VAR_BN
STATE_GEAR_REV_CAN	LV_ERR_SOF_REQ	LV_ERR_BN_VEH_MOD	LV_ERR_CAN_BOFF
STATE_ETCU_PROG_INFO	LV_CITY	CONF_SOF_SWI	LV_VAR_AMT
FAC_TQ_REQ_DRIV_MON	LV_VAR_TCT	LV_AT	PV_AV_MON
STATE_GEAR_REV_AT_AMT	PV_AV_GRD	STATE_VEH_MOD	LV_RNG_L_REQ
LV_ERR_BN_GEAR_REV			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_SOF_SWI_ON_TOL_MON	1	0...3FFH	0...4.99511	4.88E-03	[V]
Voltage-Threshold top limit for sport mode ON					
C_V_SOF_SWI_ON_BOL_MON	1	0...3FFH	0...4.99511	4.88E-03	[V]
Voltage-Threshold bottom limit for sport mode ON					
LC_VAR_SOF_SWI_MON	1	0...1H	0...1	1	[-]
Calibration sport Switch (1 on / 0 off, 2 AMT on)					
LC_RNG_L_MON	1	0...1H	0...1	1	[-]
Manual switch for low range detection monitoring					
LC_CONF_CITY_ENA_MON	1	0...1H	0...1	1	[-]
Enable City mode off condition (1 on / 0 off, 2 AMT on)					
C_STATE_ETCU_PROG_INFO_MON	1	0...FFH	0...255	1	[-]
Manual activation of low range mode AT					
C_PV_AV_SOF_MIN_MON	1	0...FFH	0...99.60937	0.390625	[%]
PV_AV threshold for activation IP_FAC_TQ_REQ_DRIV_RNG_L_MON					
C_FAC_TQ_REQ_DRIV_SOF_MAX_MON	1	0...FFH	0...1.99218	0.0078125	[-]
scaling factor treshold for activation IP_FAC_TQ_REQ_DRIV_RNG_L_MON					
C_PV_AV_GRD_SOF_MAX_MON	1	0...FFH	-1250... 1240.23437	9.765625	[%/s]
Pedal value gradient threshold for recognition of LV_RNG_L_MON					
C_T_PV_AV_GRD_SOF_MAX_MON	1	0...FFH	0...10.2	0.04	[s]
Minimum time for pedal value gradient for recognition of LV_RNG_L_MON					
C_V_SOF_SWI_MIN_MON	1	0...3FFH	0...4.99511	4.88E-03	[V]

input voltage threshold min					
C_V_SOF_SWI_MAX_MON	1	0...3FFH	0...4.99511	4.88E-03	[V]
input voltage threshold max					
C_ABC_INC_SOF_INH_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_SOF_INH_MON	1	1...FFH	1...255	1	[-]
Anti bounce counter max value					
LC_RNG_L_MAN_AS_MON	1	0...1H	0...1	1	[-]
Calibration constant for to activate IP_FAC_TQ_REQ_DRIV_RNG_L (low range mode) in case if LV_RNG_L = 0 (default value = 0)					

11.6.16.1 Acquisition of sport-mode-switch for monitoring

General information:

Calculation of the sport-mode request by taking the voltage of the sport-mode-switch and at BN 2000 variants on CAN message.
The request acts according to the rising edge of the voltage signal. This monitoring function connects to the level 1 function Chapter 9.

Application conditions:

Activation: LV_TQI_MON_ACT_MON ==1

Deactivation: otherwise

Initialisation: for condition see ‘Application Incidences of Process Monitoring’

In case of switch off the engine LV_ES = 0 to 1 or reset:

LV...MON = 0

Update Rate: 40 ms in general, exception voltage switch detection in 10 ms (this is realised by calling chapter 1.16.1 Acquisition of sport-mode-switch for monitoring" within each service task (see "Order of actions calls" at the beginning of this document)

Formula section:

Detection of tip switch position:

Application conditions:

Update rate: 10ms this in necessary to connect to level 1 because of monitoring 3 evaluation Cycles.

```
if (1)    LV_VAR_BN = 0
then (1)
    if (2)    LV_ERR_SOF_REQ = 0
    then (2)
        if (3)    (V_SOF_SWI_MON$_{n-1}$ > C_V_SOF_SWI_ON_TOL_MON) changes to
                  (C_V_SOF_SWI_ON_BOL_MON ≤  V_SOF_SWI_MON$_{n}$ ≤
                                                C_V_SOF_SWI_ON_TOL_MON)
                  (equal to 0 ->1)
                  and this result is valid for 3 evaluations
        then (3) LV_SOF_SWI_MON = ! LV_SOF_SWI_MON
```

```
            endif (3)
        else (2)      LV_SOF_SWI_MON = 0

        endif (2)
  else (1)
        if (4)     LV_ERR_BN_VEH_MOD = 0      and
                   LV_ERR_CAN_BOFF = 0        and
                   LV_ERR_BN_GEAR_REV = 0     and
                   STATE_VEH_MOD = 2H
        then (4)   LV_SOF_SWI_MON = 1
        else (4)   LV_SOF_SWI_MON = 0
        endif (4)
  endif (1)
```

Calculation of LV_SOF_SWI_REQ_MON:

```
If     (LV_CITY = 0  or  LC_CONF_CITY_ENA_MON =1)       and
       CONF_SOF_SWI     ≠ 2              and
       LC_VAR_SOF_SWI_MON = 1            and
LV_ERR_SOF_INH_MON = 0       and
       LV_SOF_SWI_MON  = 1
then   LV_SOF_SWI_REQ_MON   = 1
else   LV_SOF_SWI_REQ_MON   = 0
endif
```

11.6.16.2 Low range detection (LV_RNG_L)

Description:

Depending on LC_RNG_L_MON, the sport-switch and the LV_RNG_L_AT_MON -request from EGS the bit LV_RNG_L_MON will be activated.

Meaning of C_STATE_ETCU_PROG_INFO_MON:

Bit	LV_RNG_L_AT_MON will be set at condition:
0	*AT / TCT in "S" / "DS" ("comfort transmission"; LV_VAR_ETCU_SPT=0)*
1	*AT / TCT in manual mode "M1 ... 6" ("comfort transmission"; LV_VAR_ETCU_SPT=0)*
2	*AT- / TCT-sport-switch in "S" / "DS" ("sport transmission"; LV_VAR_ETCU_SPT=1)*
3	*AT- / TCT-sport-switch in manual mode "M1 ... 6" ("sport transmission"; LV_VAR_ETCU_SPT=1)*
4	*Gearbox-sportswitch active and AT / TCT ("comfort transmission"; LV_VAR_ETCU_SPT=0)*
5	*Gearbox-sportswitch active and AT / TCT ("sport transmission"; LV_VAR_ETCU_SPT=1)*
6	*always with MT*
7	*not used*

Formula section:

If

AT / TCT "comfort" in "S-mode"

[STATE_ETCU_PROG_INFO = 1H **and**

LV_VAR_ETCU_SPT = 0 **and**

Bit 0 of C_STATE_ETCU_PROG_INFO_MON is set**]** **or**

AT / TCT "comfort" in "M-mode"

[STATE_ETCU_PROG_INFO = 2H **and**

LV_VAR_ETCU_SPT = 0 **and**

Bit 1 of C_STATE_ETCU_PROG_INFO_MON is set**]** **or**

AT/TCT "sport" in "S-mode"

[STATE_ETCU_PROG_INFO = 1H **and**

LV_VAR_ETCU_SPT = 1 **and**

Bit 2 of C_STATE_ETCU_PROG_INFO_MON is set**]** **or**

AT/TCT "sport" in "M-mode"

[STATE_ETCU_PROG_INFO = 2H **and**

LV_VAR_ETCU_SPT = 1 **and**

Bit 3 of C_STATE_ETCU_PROG_INFO_MON is set**]** **or**

AT / TCT "comfort" in "sport-mode-gearbox"

[LV_VAR_ETCU_SPT = 0 **and**

LV_ETCU_SPT_SWI = 1 **and**

Bit 4 of C_STATE_ETCU_PROG_INFO_MON is set**]** **or**

AT / TCT "sport" in "sport-mode-gearbox"

[LV_VAR_ETCU_SPT = 1 **and**

LV_ETCU_SPT_SWI = 1 **and**

Bit 5 of C_STATE_ETCU_PROG_INFO_MON is set**]** **or**

MT continious in "sport"-mode

[LV_AT = 0 **and** LV_VAR_AMT = 0 **and** LV_VAR_TCT = 0 **and**

Bit 6 of C_STATE_ETCU_PROG_INFO_MON is set**]**

```
Then    LV_RNG_L_AT_MON = 1
Else    LV_RNG_L_AT_MON = 0
Endif
IF      LC_RNG_L_MON = 1            or    // request from application system
        LV_SOF_SWI_REQ_MON = 1      or    // request from sport switch
        LV_RNG_L_AT_MON = 1               // request from gearbox
Then    LV_RNG_L_MON = 1
```

```
Else      LV_RNG_L_MON= 0
Endif
```

11.6.16.3 Pedal value interpretation for monitoring

FUNCTION DESCRIPTION:

General information:

Depending on gear mode LV_RNG_L_MON (LV_RNG_L_MON =1) two maps can be calibrated with different pedal value interpretation for normal and low range mode. For this the switch condition LV_SWI_FAC_TQ_REQ_MON is triggered.
Because of the different recurrences of level 1 (10ms) and level 2 (40 ms) and the not regarded driver progression correction of level 1, the switch to the RNG_L_MON progression of monitoring follows by fulfilled conditions after switch of level 1 LV_RNG_L_REQ.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialisation: for condition see ‘Application Incidences of Process Monitoring’

LV_SWI_FAC_TQ_REQ_MON= 0

Update Rate: 40 ms

Formula section:

```
IF      PV_AV_MON < C_PV_AV_SOF_MIN_MON                              or
        FAC_TQ_REQ_DRIV_MON > C_FAC_TQ_REQ_DRIV_SOF_MAX_MON or
        PV_AV_GRD > C_PV_AV_GRD_SOF_MAX_MON
        for at least C_T_PV_AV_GRD_SOF_MAX_MON
THEN    LV_CDN_RNG_L_REQ_MON = 1
ELSE    LV_CDN_RNG_L_REQ_MON = 0
ENDIF

IF      (LV_RNG_L_MON = 1 or  LC_RNG_L_MAN_AS_MON = 1)
        and
        (STATE_GEAR_REV_CAN = 0
        and
        STATE_GEAR_REV_AT_AMT  = 0  (no reverse gear active))
        and
        LV_RNG_L_REQ = 1  (Level 1 has switched))
        and
        {       To activate IP_FAC_TQ_REQ_DRIV_1_MON following conditions
                has to be checked. After activating this conditions are not used any more.
                LV_SWI_FAC_TQ_REQ_MON = 0
                and
                [ LV_SOF_SWI_REQ_MON changes from 0 -> 1  or
                LV_RNG_L_AT_MON changes from 0 -> 1         or
                STATE_GEAR_REV_CAN changes from 1 -> 0     or
                STATE_GEAR_REV_AT_AMT changes from 1 -> 0]
```

```
                and
                [ LV_CDN_RNG_L_REQ_MON = 1          or
                LV_CDN_RNG_L_REQ_MON $_{n-1}$ = 1 ]}
THEN        LV_SWI_FAC_TQ_REQ_MON = 1
ELSE
            IF(2)   (STATE_GEAR_REV_CAN = 1                                  OR
                    STATE_GEAR_REV_AT_AMT  = 1  ( reverse gear active)       OR
                    LV_RNG_L_REQ = 0)
            THEN(2)
                    LV_SWI_FAC_TQ_REQ_MON = 0
            ENDIF(2)
ENDIF
```

11.6.16.4 Sport switch voltage range check monitoring diagnosis

FUNCTION DESCRIPTION:

General information:

This diagnosis is for checking the range of input signal voltage from the sport-mode-switch in compare to level one diagnosis.

Application conditions:

Activation: LV_VAR_BN = 0 **and**
LV_IGK_MON = 1 **and**
CONF_SOF_SWI = 1 **and**
LV_ERR_BN_GEAR_REV = 0 **and**
LV_ERR_SOF_REQ = 0

Deactivation: otherwise

Initialisation: for condition see ‘Application Incidences of Process Monitoring’
at reset or LV_IGK_MON = 0 → 1 all 0

Update Rate: 40 ms

Formula section:

```
If          [(V_SOF_SWI_MON < C_V_SOF_SWI_MIN_MON (Short circuit to ground)
                              AND
                              CONF_SOF_SWI = 1)
                             or
            V_SOF_SWI_MON > C_V_SOF_SWI_MAX_MON (Short circuit to Ubatt or open line)]
                             and
            LV_ERR_SOF_REQ = 0
Then        LV_ERR_SOF_INH_MON = 1 after debounce
            The ETC Monitoring "Debounce mechanism" is used
Endif
```

11.6.17 Driving off assistance monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_DROF_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
TQ request for drive off assistance					
TQ_AV_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
effective torque					
TQ_DROF_MDL_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
drive off model value					
LV_TQ_DROF_ACT_MON	V	0...1H	0...1	1	[-]
function active					
LV_CS_MON	V	0...1H	0...1	1	[-]
clutch switch level 2					
N_DIF_DROF_MON	V	0...FFH	-4096...4064	32	[rpm]
Engine speed difference effective for TQ_DROF					

Input data:

N_32_MON	TQI_AV_MON	LV_ISC_OFF_DROF	TQ_LOSS_MON
TQ_DROF_SLOW	LV_VAR_TCT	FAC_TQ_REQ_MON	TQ_DROF_FAST
LV_ERR_CRK	LV_ERR_PVS	LV_ERR_VS	LV_AT
LV_VAR_AMT			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_N_GRD_TQ_AV_MON	4	0...FFH	0...1.99218	0.0078125	[-]
LDP_TQ_AV_MON__TQ_DROF	4	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
N_DIF correction					
IP_TQ_DROF_MON	6	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_DIF_DROF_MON__TQ_DROF	6	0...FFH	-4096...4064	32	[rpm]
drive off torque request					
IP_FAC_TQ_DROF_VS_MON	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_VS_MON__TQ_DROF	6	0...FFFFH	0...511.99218	0.0078125	[km/h]
LDP_N_32_MON__TQ_DROF	6	0...FFH	0...8160	32	[rpm]
correction depending on VS and N_32					
IP_FAC_TQ_DROF_FAC_TQ_REQ_MON	6	0...FFH	0...1.99218	0.0078125	[-]
LDP_FAC_TQ_REQ_MON__TQ_DROF	6	0...FFH	0...1.99218	0.0078125	[-]
correction FAC_TQ_REQ_MON					
C_N_DIF_DROF_MAX_MON	1	0...FFH	-4096...4064	32	[rpm]
max threshold N_DIF_DROF_MON					
C_N_DIF_DROF_MIN_MON	1	0...FFH	-4096...4064	32	[rpm]
min threshold N_DIF_DROF_MON					
C_N_DIF_ADD_DROF_MON	1	0...FFH	-4096...4064	32	[rpm]
offset threshold N_DIF_DROF_MON					
C_N_32_TQ_DROF_MAX_MON	1	0...FFH	0...8160	32	[rpm]
min threshold N_DIF_DROF_MON					
ID_TQ_DROF_ENA_MON	3	0...1H	0...1	1	[-]
LDP_N_DIF_DROF_MON__ID_TQ_DROF	3	0...FFH	-4096...4064	32	[rpm]
table for activation/deactivation of DROF					
IP_TQ_DROF_GRD_DEAC_MON	3	0...7FFFH	0...1023.96875	0.03125	[Nm]
LDP_TQ_DROF_MDL_MON_1	3	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
Decrement for the termination ramp of TQ_DROF_MDL_MON					

FUNCTION DESCRIPTION:

The function “Driving off assistance” shall prevent the engine from stalling at driving off. This is shown by LV_N_SP_IS_CS = 1.
An extra torque request is calculated under individual conditions. While this extra torque is effective. Using the handshake to level 1 Please verify that LV_ISC_OFF_DROF in Level one is calibrated to 1.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1 **and** LV_ERR_CRK = 0 **and** LV_AT = 0 **and** LV_ERR_VS = 0 **and** LV_ERR_PVS = 0 **and** LV_VAR_AMT = 0 **and** LV_VAR_TCT = 0 **and** N_32_MON < C_N_32_TQ_DROF_MAX_MON

Deactivation: LV_TQI_MON_ACT_MON = 0 **or** LV_ERR_CRK = 1 **or** LV_AT = 1 ***or*** LV_ERR_VS = 1 **or** LV_ERR_PVS = 1 **or** LV_VAR_AMT = 1 **or** LV_VAR_TCT = 1 **or** N_32_MON ≥ C_N_32_TQ_DROF_MAX_MON

Initialisation: at ECU-reset **or** at function deactivation:
all = 0 (phy)

Recurrence: 40 ms

Formula section:

LV_CS_MON is read direct on the clutch switch pinning input.

TQ_AV_MON = TQI_AV_MON + TQ_LOSS_MON

N_DIF_DROF_MON = IP_FAC_N_GRD_TQ_AV_MON * C_N_DIF_ADD_DROF_MON + N_DIF_SP_IS_MON

```
If      N_DIF_DROF_MON <= C_N_DIF_DROF_MAX_MON         and
        N_DIF_DROF_MON >= C_N_DIF_DROF_MIN_MON         and
        LV_CS_MON ==1                                   and
        ID_TQ_DROF_ENA_MON == 1                         and
        LV_ISC_OFF_DROF == 1        /*check if level 1 is active*/
Then    LV_TQ_DROF_ACT_MON == 1
Else    LV_TQ_DROF_ACT_MON == 0
Endif
If      LV_TQ_DROF_ACT_MON == 1
Then    TQ_DROF_MDL_MON =     IP_TQ_DROF_MON *
                              IP_FAC_TQ_DROF_FAC_TQ_REQ_MON *
                              IP_FAC_TQ_DROF_VS_MON
Else    $TQ\_DROF\_MDL\_MON_{n}$ = $TQ\_DROF\_MDL\_MON_{n-1}$ -
                              IP_TQ_DROF_GRD_DEAC_MON
Endif
```

TQ_DROF_MON =

MIN(MAX(TQ_DROF_FAST;TQ_DROF_SLOW);TQ_DROF_MDL_MON)

11.6.18 Application incidences for the desired indicated engine torque

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PV_AV_AD_MON	O/V	0...FFH	0...99.60937	0.390625	[%]
Modified pedal value signal					
TQI_INC_EXT_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Monitored external torque demands					
TQI_MIN_PU_MON	O/V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Plausiblised L2 value of Minimum indicated torque in PU phase					
TQI_MIN_PU_MAX_MON	V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Monitoring threshold for minimum indicated torque in PU phase					
T_TQI_OFS_ST_MON	V	0...FFH	0...10.2	0.04	[s]
Timer for start offset					
TQI_OFS_ST_MON	V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Start offset for TQI_MIN_PU					
FAC_PV_BAS_COR_MON	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
basic correction - monitoring level					
FAC_PV_COR_MON	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Summation of corrections - monitoring level					
TQI_N_CTL_TCT_MON	O/V	0...FFH	0...510	2	[Nm]
Monitored external torque request from TCT					
INT_TQI_N_CTL_TCT_MON	O/V	0...FFFFH	0...204.79687	0.003125	[Nm*s]
Integrated torque request from TCT					
LV_INT_TQI_N_CTL_TCT_MON	V	0...1H	0...1	1	[-]
Inhibition of external torque request from TCT due to integral check					
T_TQI_N_CTL_TCT_MON	V	0...FFH	0...10.2	0.04	[s]
Duration of external torque request from TCT					
LV_T_MAX_TQI_N_CTL_TCT_MON	V	0...1H	0...1	1	[-]
Inhibition of external torque request from TCT due to maximal time check					
T_TQI_N_CTL_TCT_MAX_MON	V	0...FFH	0...10.2	0.04	[s]
Duration of external torque request from TCT over the maximum allowed					
LV_TQ_MAX_TQI_N_CTL_TCT_MON	V	0...1H	0...1	1	[-]
Inhibition of external torque request from TCT due to maximal torque check					
LV_INH_TQI_N_CTL_TCT_MON	V	0...1H	0...1	1	[-]
General Inhibition bit of external torque request from TCT					

Input data:

PV_AV_MON	TQI_INC_CAN_MON	LV_TQI_MON_ACT_MON	TQI_N_CTL_TCT_FAST
N_32_MON	TCO_MON	VS_MON	TQI_N_CTL_TCT_SLOW

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQI_MIN_PU_MAX_MON	6*3	0...7FFFH	0...1023.96875	0.03125	[Nm]
LDP_N_32_MON_IP_TQI_PU_MAX_MON	6	0...FFH	0...8160	32	[rpm]
LDP_TCO_MON_IP_TQI_PU_MAX_MON	3	0...FEH	-48...142.5	0.75	[°C]
Monitoring threshold for minimum indicated torque in PU phase					
C_T_TQI_OFS_ST_MON	1	0...FFH	0...10.2	0.04	[s]
Timer for TQI_MIN_PU startoffset					
C_N_ST_MIN_MON	1	0...FFH	0...8160	32	[rpm]
Enginespeed to activate startoffset for TQI_MIN_PU					
C_TQI_OFS_ST_MON	1	0...7FFFH	0...1023.96875	0.03125	[Nm]
TQI_MIN_PU startoffset					
IP_FAC_PV_BAS_COR_MON	6*6	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_32_MON__FAC_PV_COR_MON	6	0...FFH	0...8160	32	[rpm]
LDPM_PV_AV_MON__FAC_PV_COR_MON	6	0...FFH	0...99.60937	0.390625	[%]
Basic correction for PV_AV_MON					
IP_FAC_VS_PV_BAS_COR_MON	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_VS_MON__FAC_PV_COR_MON	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction for PV_AV_MON					
C_CRLC_TQI_N_CTL_TCT_PLAUS_MON	1	0...FFH	0...1	3.9216e-3	[-]
Correlation factor for calculation of integrated torque request from TCT					
C_INT_MAX_TQI_N_CTL_TCT_MON	1	0...FFFFH	0...204.79687	0.003125	[Nm*s]
Maximum value of integrated torque request from TCT					
C_INT_MIN_TQI_N_CTL_TCT_MON	1	0...FFFFH	0...204.79687	0.003125	[Nm*s]
Minimum value of integrated torque request from TCT					
C_T_MAX_TQI_N_CTL_TCT_MON	1	0...FFH	0...10.2	0.04	[s]
Maximum time for torque request from TCT					
IP_TQ_MAX_TQI_N_CTL_TCT_MON	6	0...FFH	0...510	2	[Nm]
LDP_N_32__TQI_N_CTL_TCT_MON	6	0...FFH	0...8160	32	[rpm]
Threshold for activation of maximal torque check for torque request from TCT					
C_T_MAX_TQI_N_CTL_TCT_MAX_MON	1	0...FFH	0...10.2	0.04	[s]
Maximum time for torque request from TCT over the maximum allowed					
C_DEC_T_TQI_N_CTL_TCT_MAX_MON	1	0...FFH	0...10.2	0.04	[s]
Decrement of time for torque request from TCT over the maximum allowed					
C_DEC_T_TQI_N_CTL_TCT_MON	1	0...FFH	0...10.2	0.04	[s]
Decrement of time for torque request from TCT					
C_DEC_INT_TQI_N_CTL_TCT_MON	1	0...FFFFH	0...204.79687	0.003125	[Nm*s]
Decrement of integrated torque request from TCT					
C_MIN_TQI_N_CTL_TCT_MON	1	0...FFH	0...510	2	[Nm]
Threshold for monitoring of external torque request from TCT					

FUNCTION DESCRIPTION:

General information:

The torque demands from cruise control, electronic gear shift control and the electronic stability program are set to zero, if these functions are not available. Otherwise, the torque demands have to be copied to the corresponding variable of level 2 and these variables have to be checked for plausibility (e.g. CAN monitoring, CRU monitoring, control units [ESP, EGS] with internal plausibility check).

For torque request at trailing throttle (TQI_MIN_PU_MON) the map output TQI_MIN_PU_MAX_MON (monitoring threshold) will be taken into account. For consideration of the torque setpoint offset at engine start the corresponding torque value is added to TQI_MIN_PU_MAX_MON. After engine start this value is set to "0".

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialisation: All 0 for condition see ‘Application Incidences of Process Monitoring’

Update Rate: 40 ms

Formula section:

11.6.18.1 Calculation of PV_AV_AD_MON

FAC_PV_BAS_COR_MON = IP_FAC_PV_BAS_COR_MON (N32_MON; PV_AV_MON)
*IP_FAC_VS_PV_BAS_COR_MON (VS_MON)

FAC_PV_COR_MON = min(1;max(0;FAC_PV_BAS_COR_MON))

PV_AV_AD_MON = PV_AV_MON*(1 – FAC_PV_COR_MON)

11.6.18.2 Calculation of external torque demand from TCT

TQI_N_CTL_TCT_MON = max (TQI_N_CTL_TCT_FAST ; TQI_N_CTL_TCT_SLOW)

11.6.18.2.1 Integrated torque request monitoring

```
If      TQI_N_CTL_TCT_MON >= C_MIN_TQI_N_CTL_TCT_MON
Then    INT_TQI_N_CTL_TCT_MON = MIN{C_INT_MAX_TQI_N_CTL_TCT_MON;
                MAX(C_INT_MIN_TQI_N_CTL_TCT_MON ;
                INT_TQI_N_CTL_TCT_MON_{k-1} + (TQI_N_CTL_TCT_MON *
                (C_CRLC_TQI_N_CTL_TCT_PLAUS_MON * 0,04s)))}
Else    INT_TQI_N_CTL_TCT_MON = MIN{ C_INT_MAX_TQI_N_CTL_TCT_MON;
                MAX(C_INT_MIN_TQI_N_CTL_TCT_MON ;
                INT_TQI_N_CTL_TCT_MON_{k-1} –
                C_DEC_INT_TQI_N_CTL_TCT_MON)}
Endif
If      INT_TQI_N_CTL_TCT_MON = C_INT_MAX_TQI_N_CTL_TCT_MON
Then    LV_INT_TQI_N_CTL_TCT_MON = 1
Else    LV_INT_TQI_N_CTL_TCT_MON = 0
Endif
```

11.6.18.2.2 Maximum request time monitoring

```
If      TQI_N_CTL_TCT_MON >= C_MIN_TQI_N_CTL_TCT_MON
Then    T_TQI_N_CTL_TCT_MON_{k} =
            MIN(T_TQI_N_CTL_TCT_MON_{k-1} + 40 ms ;
                              C_T_MAX_TQI_N_CTL_TCT_MON)
Else    T_TQI_N_CTL_TCT_MON_{k} = T_TQI_N_CTL_TCT_MON_{k-1} –
                              C_DEC_T_TQI_N_CTL_TCT_MON
```

```
Endif

If      T_TQI_N_CTL_TCT_MON = C_T_MAX_TQI_N_CTL_TCT_MON
Then    LV_T_MAX_TQI_N_CTL_TCT_MON = 1
Else    LV_T_MAX_TQI_N_CTL_TCT_MON = 0
Endif
```

11.6.18.2.3 Maximum torque request monitoring

```
IF      TQI_N_CTL_TCT_MON > IP_TQ_MAX_TQI_N_CTL_TCT_MON
Then    T_TQI_N_CTL_TCT_MAX_MON $_{k}$ =
                MIN(T_TQI_N_CTL_TCT_MAX_MON $_{k-1}$ + 40 ms ;
                                        C_T_MAX_TQI_N_CTL_TCT_MAX_MON)
Else    T_TQI_N_CTL_TCT_MAX_MON $_{k}$ = T_TQI_N_CTL_TCT_MAX_MON $_{k-1}$ -
                                        C_DEC_T_TQI_N_CTL_TCT_MAX_MON
Endif

IF      T_TQI_N_CTL_TCT_MAX_MON = C_T_MAX_TQI_N_CTL_TCT_MAX_MON
Then    LV_TQ_MAX_TQI_N_CTL_TCT_MON = 1
Else    LV_TQ_MAX_TQI_N_CTL_TCT_MON = 0
Endif
```

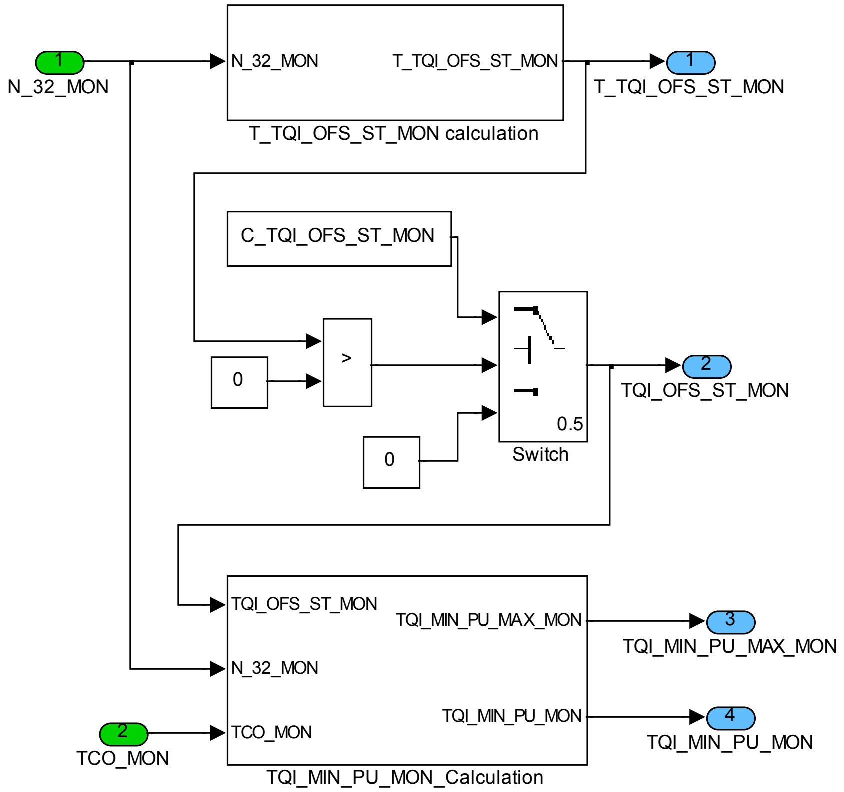
11.6.18.2.4 Inbhibition of external torque demand from TCT

```
IF      LV_INT_TQI_N_CTL_TCT_MON = 1        or
        LV_T_MAX_TQI_N_CTL_TCT_MON = 1      or
        LV_TQ_MAX_TQI_N_CTL_TCT_MON = 1
Then    LV_INH_TQI_N_CTL_TCT_MON = 1
        TQI_N_CTL_TCT_MON = 0 Nm
Endif
```

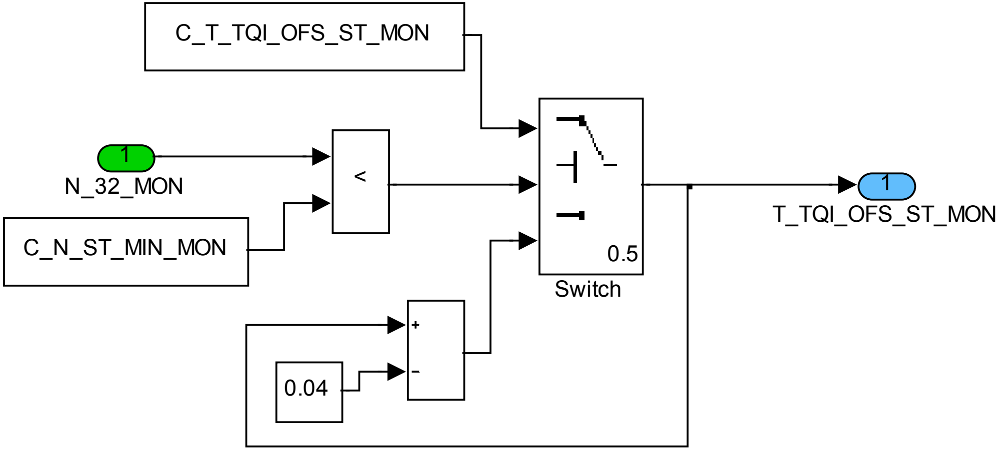
11.6.18.3 Calculation of external torque demands

TQI_INC_EXT_MON = MAX(TQI_INC_CAN_MON ; TQI_N_CTL_TCT_MON)

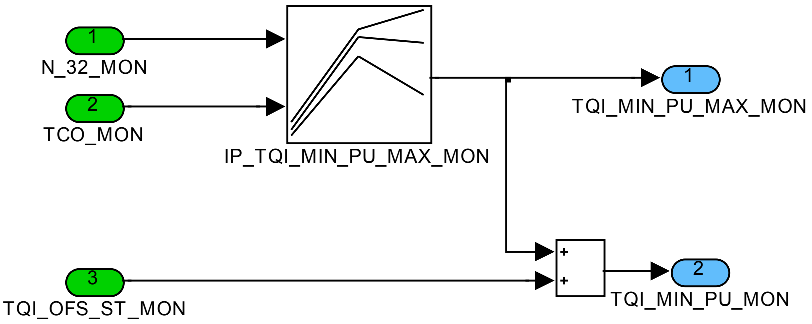
11.6.18.4 Minimum indicated engine torque in trailing throttle with consideration of torque setpoint offset at engine start



Appl.INC_Desired_indicated_torque/Plausibilis. of minimum indicated torque in PU



Appl.INC_Desired_indicated_torque/Plausibilis. of minimum indicated torque in PU/ T_TQI_OFS_ST_MON calculation



Appl.INC_Desired_indicated_torque/Plausibilis. of minimum indicated torque in PU/ TQI_MIN_PU_MON_Calculation

11.6.19 Application incidences for the combustion mode switch request

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SWI_AFS_LAMB_MON	O/V	0...1H	0...1	1	[-]
Flag for switching to AFS mode					
LV_SWI_AFS_POST_INJ_MON	O/V	0...1H	0...1	1	[-]
Flag for switching to AFS mode					

General information:

FUNCTION DESCRIPTION:

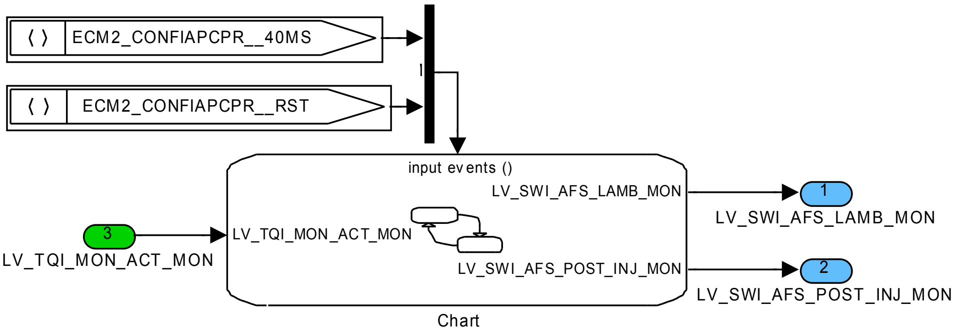


Figure 1: ECM2_CONFIAPCPR

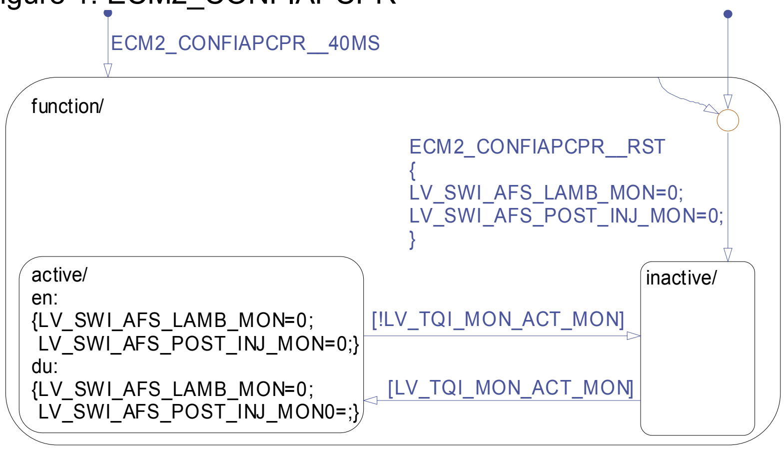


Figure 2: ECM2_CONFIAPCPR/Chart

11.6.20 Application incidences for the actual fuel mass flow

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_MFF_WUP_CUS_MON	O/V	0...FFH	0...1.99218	0.0078125	[-]
OPM dependent mass fuel flow correction in warm-up phase (Monitoring level)					
IDX_TI_1_MON[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for first pulse (monitoring)					
IDX_TI_2_MON[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for second pulse (monitoring)					
IDX_TI_POST_MON[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for post pulse (monitoring)					

Input data:

LV_MPL_INJ_ACT[NC_CBK_EX_NR]	FAC_MFF_WUP_HOMS	IDX_TI_1_HOM_CLC[NC_CYL_NR]	IDX_TI_2_HOM_CLC[NC_CYL_NR]
IDX_TI_3_HOM_CLC[NC_CYL_NR]	FAC_MFF_WUP_S	IDX_TI_1_S_CLC[NC_CYL_NR]	IDX_TI_2_S_CLC[NC_CYL_NR]
IDX_TI_3_S_CLC[NC_CYL_NR]	TI_POST_INJ[NC_CYL_NR]	LV_POST_INJ_ACT	

Application conditions:

Activation:	LV_TQI_MON_ACT_MON = 1
Deactivation:	LV_TQI_MON_ACT_MON = 0
Initialisation:	for condition see ‘Application Incidences of Process Monitoring’ FAC_MFF_WUP_CUS_MON = 1 IDX_TI_1_MON = 0 IDX_TI_2_MON = 0 IDX_TI_POST_MON = 0
Recurrence:	40ms

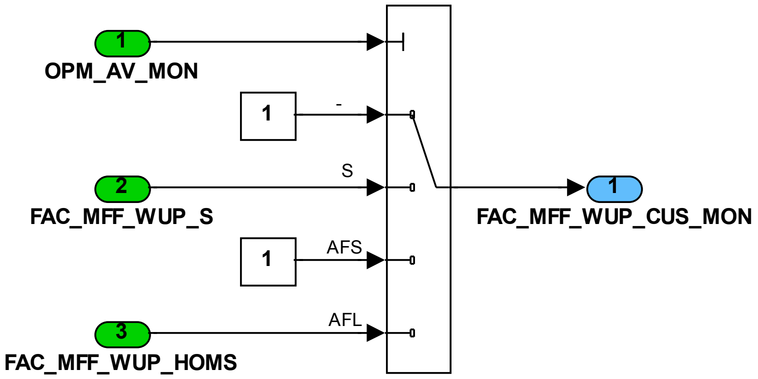
FUNCTION DESCRIPTION:

General information:

Dependent from OPM_AV_MON different fuel mass flow corrections for the warm-up phase are considered in FAC_MFF_WUP_CUS_MON.

When MFMA is activated IDX_TI_1_MON is the total monitored injection index.

Signal flow diagram:



IDX_TI_x_MON calculation:

```
[i] = [NC_CYL_NR]
IF OPM_AV_MON == AFS
    IF      (LV_MPL_INJ_ACT[1] == 0  AND LV_MPL_INJ_ACT[2]==0)
    THEN
            IDX_TI_1_MON[i] = IDX_TI_1_HOM_CLC[i];
            IDX_TI_2_MON[i] = IDX_TI_2_HOM_CLC[i];
            IDX_TI_POST_MON[i] = IDX_TI_3_HOM_CLC[i];
    ELSE  active MFMA
            IDX_TI_1_MON[i] =
               IDX_TI_1_HOM_CLC[i] + IDX_TI_2_HOM_CLC[i] + IDX_TI_3_HOM_CLC[i];
            IDX_TI_2_MON[i] = 0;
            IDX_TI_POST_MON[i] = 0;
    ENDIF
ELSE
    IF      (LV_MPL_INJ_ACT[1] == 0  AND LV_MPL_INJ_ACT[2]==0)
    THEN
            IDX_TI_1_MON[i] = IDX_TI_1_S_CLC[i];
            IDX_TI_2_MON[i] = IDX_TI_2_S_CLC[i];
            IF              LV_POST_INJ_ACT == 0
            THEN
                            IDX_TI_POST_MON[i] = IDX_TI_3_S_CLC[i];
            ELSE
                            IDX_TI_POST_MON[i] = TI_POST_INJ[i] / 0.001ms
```

```
                ENDIF
        ELSE  active MFMA
                IDX_TI_1_MON[i] =
                IDX_TI_1_S_CLC[i] + IDX_TI_2_S_CLC[i] + IDX_TI_3_S_CLC[i];
                IDX_TI_2_MON[i] = 0;
                IDX_TI_POST_MON[i] = 0;
        ENDIF
    ENDIF
```

11.7 Monitoring of analogue to digital conversion

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CONV_MON	O/V	0...1H	0...1	1	-
Logical variable for ADC error					
ABC_CONV_MON	O/V	0...FFH	0...255	1	-
Anti-bounce counter for ADC error					

Input data:

VP_MC_AN_DIG_MON	VP_MU_AN_DIG_MON		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_CONV_MON	1	0...FFH	0...255	1	-
Anti bounce counter increment (additive value in case of ADC error)					
C_ABC_MAX_CONV_MON	1	1...FFH	1...255	1	-
Value at which ADC error is recognised when reached					
C_STATE_AN_DIG_ERR_ACT_MON	1	0...FFH	0...255	1	-
Calibratable value used for provoking an error of the A/D conversion: if C_STATE_AN_DIG_ERR_ACT_MON has value F8H (=248) then the constant value 0, instead of ACQ_MU_0_MON is used as input for the comparison, resulting in an error.					
C_VP_ACQ_CONV_HYS_MON	1	0...7FFFH	0...4.999847	1.52588E-4	V
Maximal allowed difference between the voltages on main controller and monitoring unit					

Import actions:

ACTION_ECM3_Service3TaskPfm(IN <>)

ACTION_ECM3_Service4TaskPfm(IN <>)

ACTION_ECM3_Service5TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 1.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service3TaskPfm() and ACTION_ECM3_Service4TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)

ACTION_ECM3_ChkCpl(IN <>, IN <>)

ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

General information:

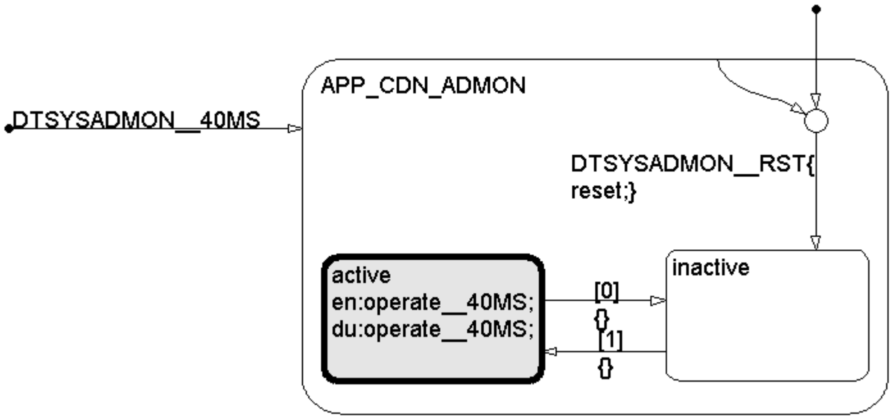
The analogue-digital-converter (ADC) diagnosis is performed by comparing two digital signals. The according analog signal is the same, it is converted once on the main controller (VP_MC_AN_DIG_MON) and also on the monitoring unit (VP_MU_ AN_DIG_MON).

On an error free system, both values are not allowed to differ more than the hysteresis C_VP_ACQ_CONV_HYS_MON.

This hysteresis also covers differences caused by the different sampling times.

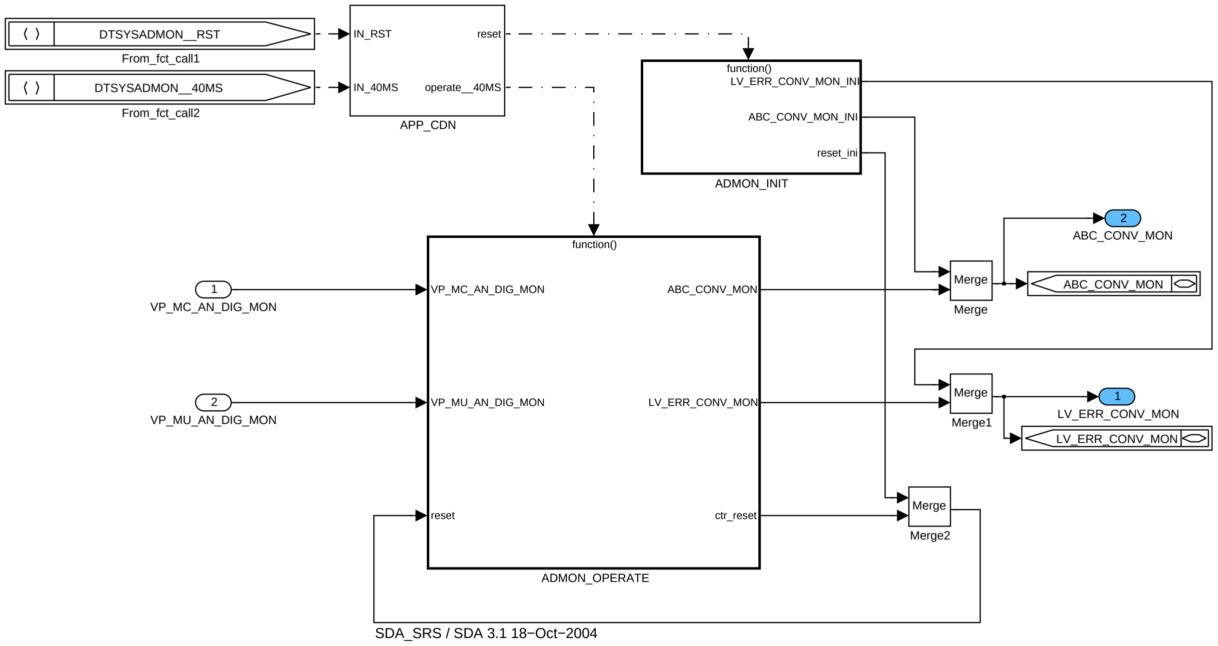
There is the possibility to test whether the A/D monitoring works; this can be achieved by setting C_STATE_AN_DIG_ERR_ACT_MON to the value 248 (=F8H), resulting in the fact that the constant 0 (instead of VP_MU_AN_DIG_MON) is used as one of the inputs for the comparison, provoking a fault reaction of the ADC monitoring.

Application Condition

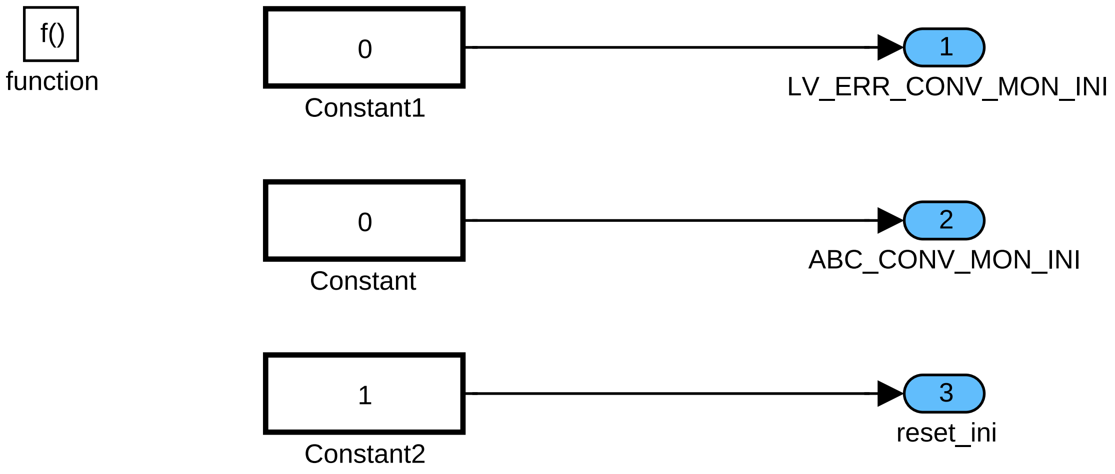


Module_ECM2_DTSYSADMON/APP_CDN/Chart

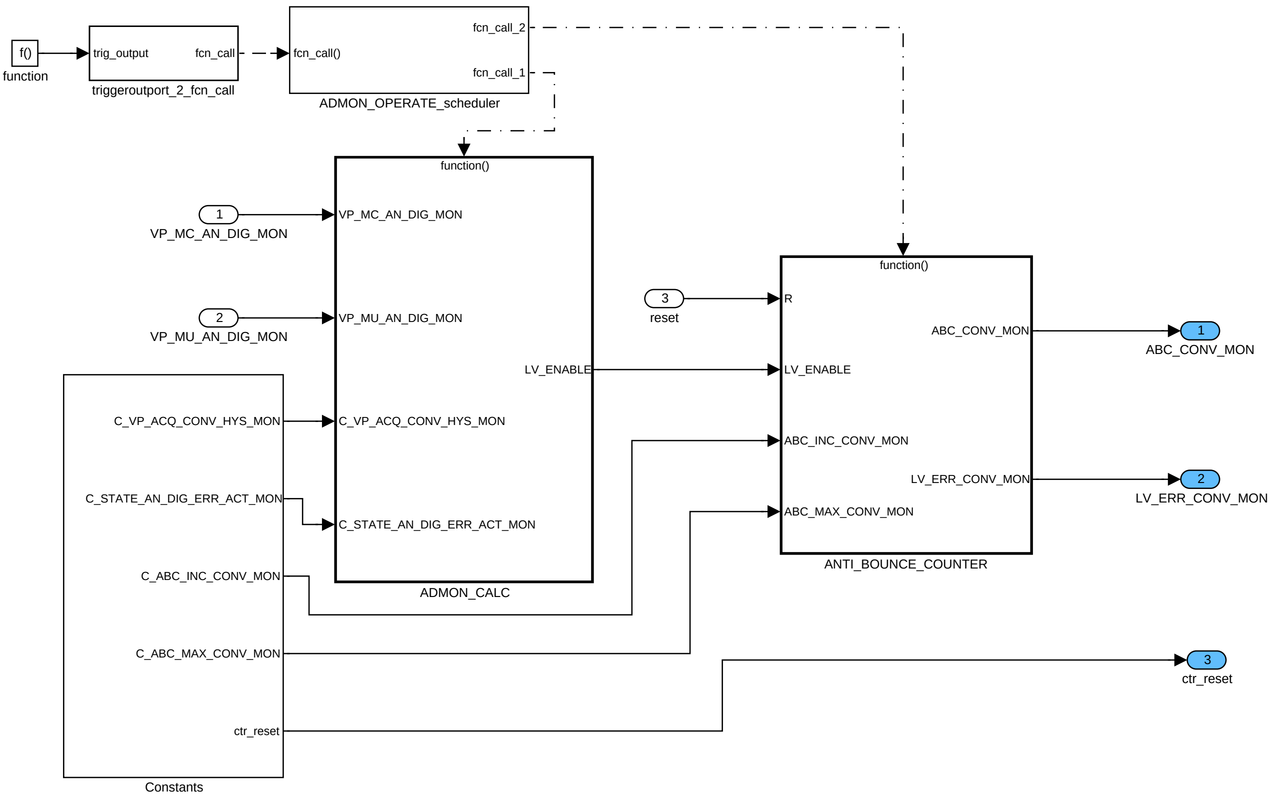
Function Description



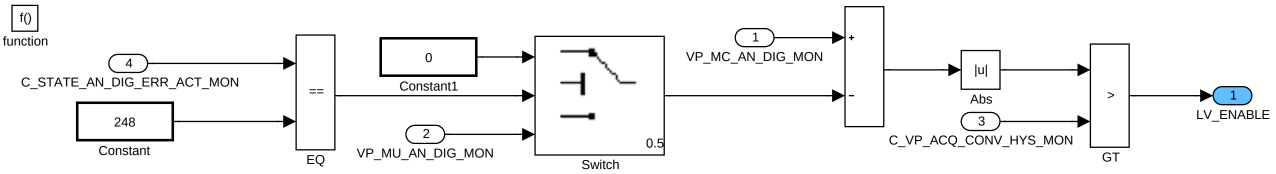
ECM2_DTSYSADMON



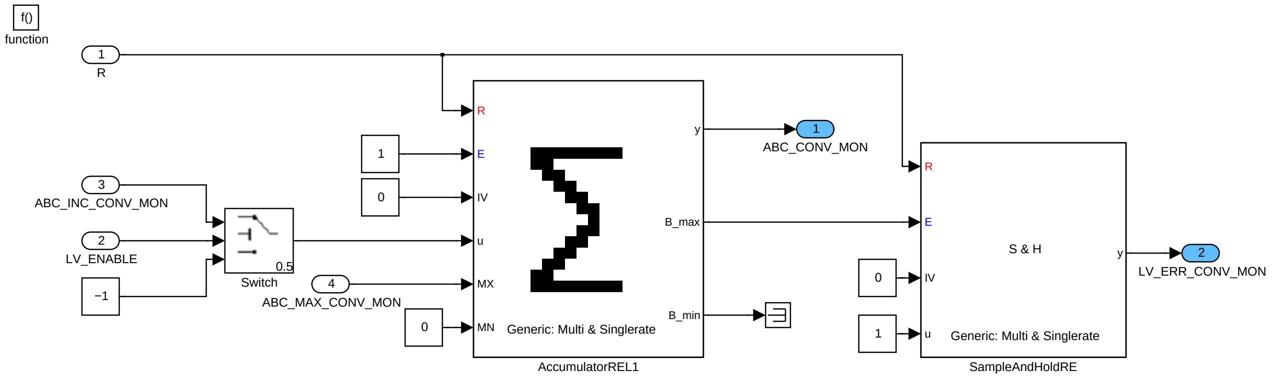
ECM2_DTSYSADMON/ADMON_INIT



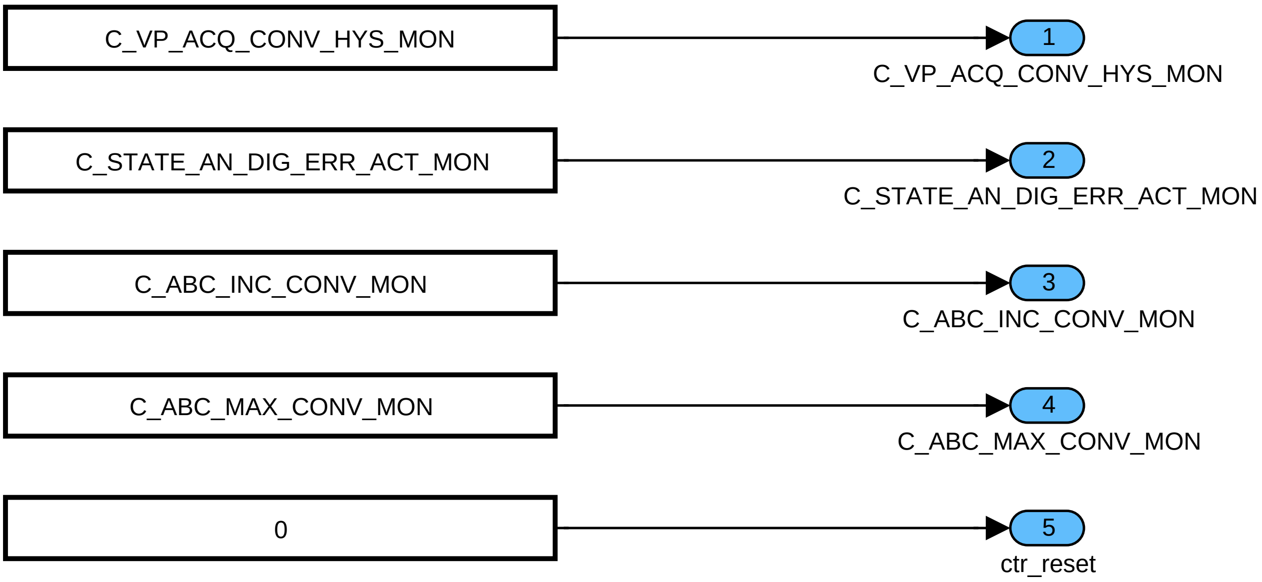
ECM2_DTSYSADMON/ADMON_OPERATE



ECM2_DTSYSADMON/ADMON_OPERATE/ADMON_CALC



ECM2_DTSYSADMON/ADMON_OPERATE/ANTI_BOUNCE_COUNTER



ECM2_DTSYSADMON/ADMON_OPERATE/Constants

11.8 Monitoring of coolant temperature

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TCO_MON	V/O	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature (monitoring level)					
T_TCO_GRD_MON	V	0...FFH	0...10.2	0.04	[s]
Timer for negative TCO gradient limitation					

Input data:

LV_TQI_MON_ACT_MON	TCO	TCO_SUB	

Import actions:

ACTION_ECM3_Service6TaskPfm(IN <>)

ACTION_ECM3_Service7TaskPfm(IN < >)

ACTION_ECM3_Service8TaskPfm(IN < >)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 1.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service6TaskPfm() and ACTION_ECM3_Service7TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)

ACTION_ECM3_ChkCpl(IN <>, IN <>)

ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_GRD_MAX_MON	1	0...FEH	0...190.5	0.75	[°C]
Maximum negative gradient for the coolant temperature					
C_T_TCO_GRD_MAX_MON	1	0...FFH	0...10.2	0.04	[s]
Minimum time before a new change of TCO_MON is allowed					
C_TCO_THD_MIN_MON	1	0...FEH	-48...142.5	0.75	[°C]
Threshold for the min. TCO temperature while TCO_SUB is higher than C_TCO_SUB_THD_MAX_MON					
C_TCO_SUB_THD_MAX_MON	1	0...FE00H	-48...142.5	2.9297e-3	[°C]
Threshold for the max. TCO_SUB temperature while TCO is lower than C_TCO_THD_MIN_MON					

11.8.1 FUNCTION PART: ECM2_DTSYSTCO

General information:

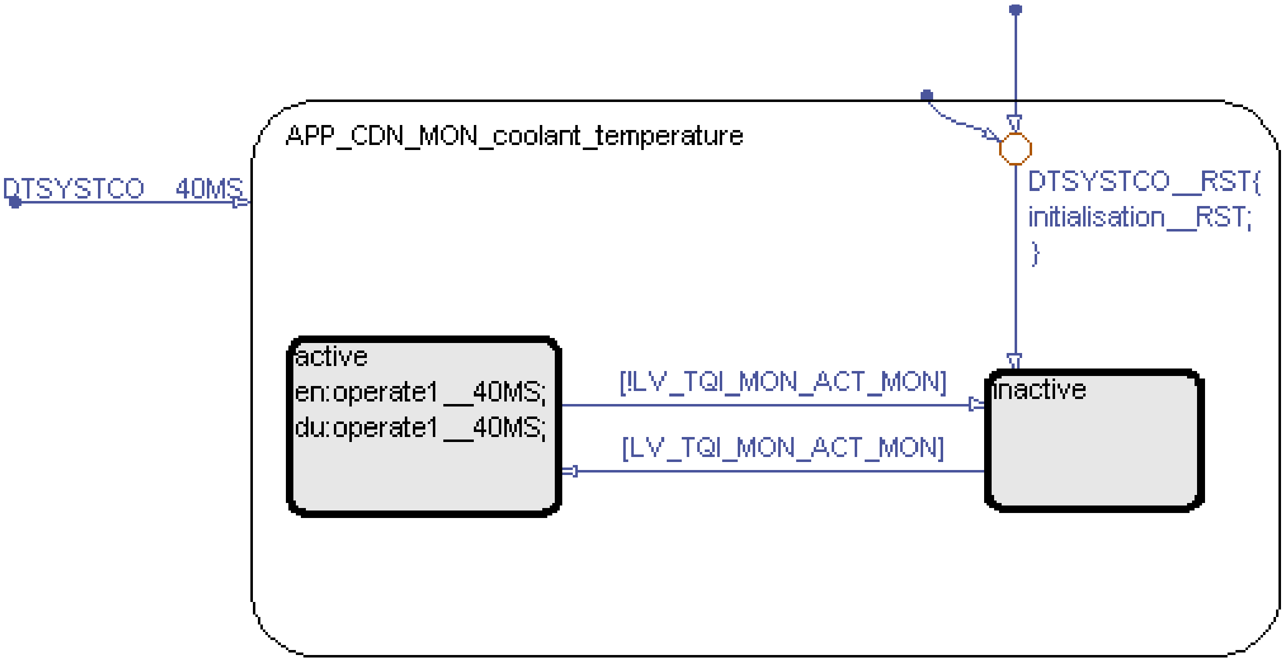
A sudden decrease in the coolant temperature TCO caused by a fault can lead to an undesired increase of the torque demand and thus to an undesired acceleration of the car because of the temperature dependent friction compensation. Therefore the negative temperature gradient of TCO shall be limited in the function level. In order to detect process faults, the gradient of the coolant temperature which is copied to the process monitoring level is also limited, i.e. to C_TCO_GRD_MAX_MON. Also the diagnosis for a too low TCO-sensor temperature is monitored. The resulting value TCO_MON is then used within the monitoring level. If a fault in TCO leads to an increased engine torque in the idle speed controller the module ‘Monitoring of idle speed controller’ detects this fault as an increased torque demand from the idle speed controller.

In case of an error free system the temperature TCO_MON used in the monitoring level should be the same as the temperature TCO used in the function level.

An additional timer is needed to compensate the different recurrencies of level 1 and level 2. The maximum counter threshold C_T_TCO_GRD_MAX_MON and the maximum negative temperature gradient C_TCO_GRD_MAX_MON need to be adapted in regard to the function recurrency and to C_TCO_GRD_MAX of level 1.

The resolution of T_TCO_GRD_MON and C_T_TCO_GRD_MAX_MON depend on the update rate of the module.

Application Condition



Note: DTSYSTCO__RST includes the function calls as defined in application incidences.

Function Description

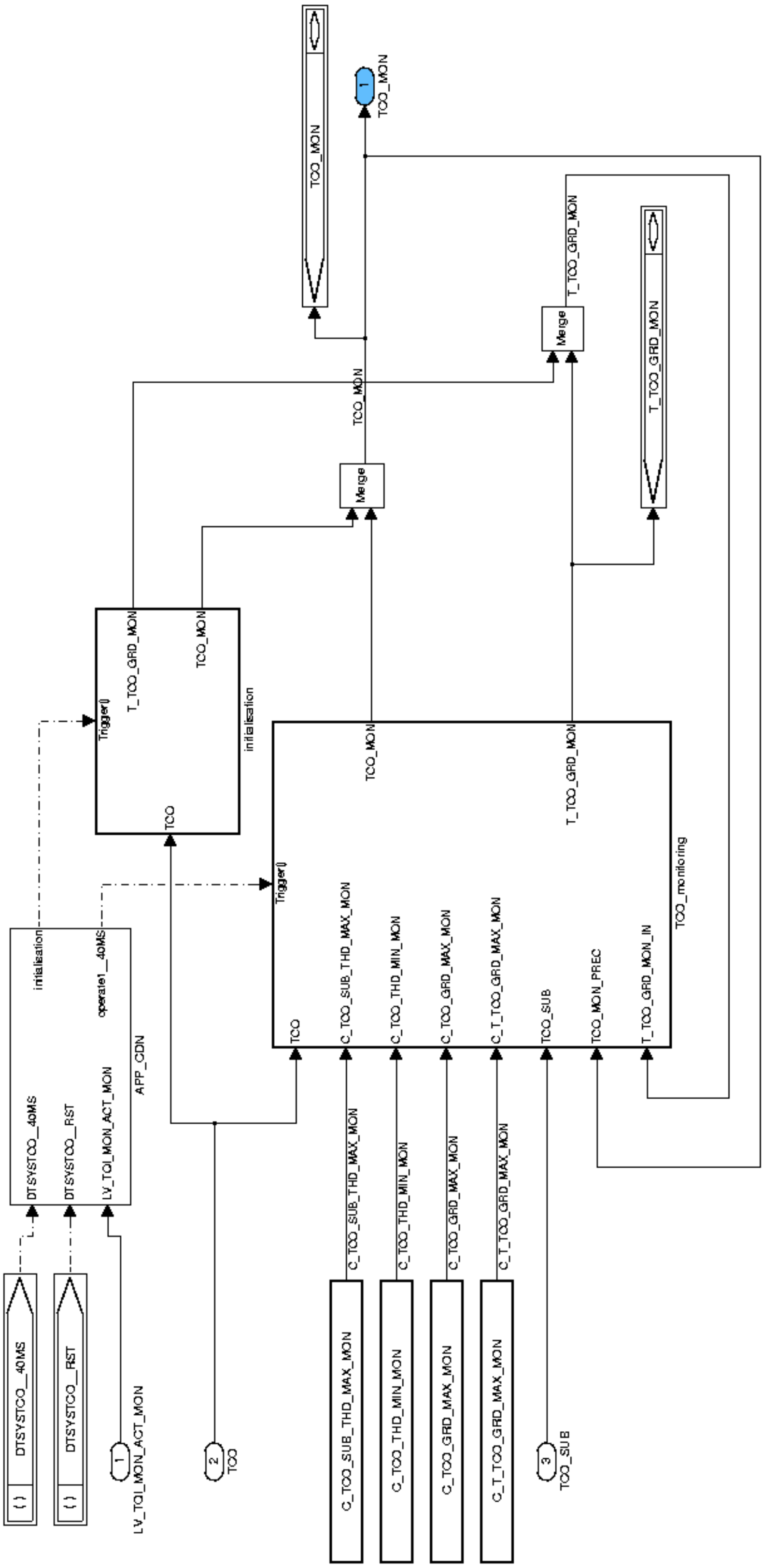


Figure 1 ECM2_DTSYSTCO

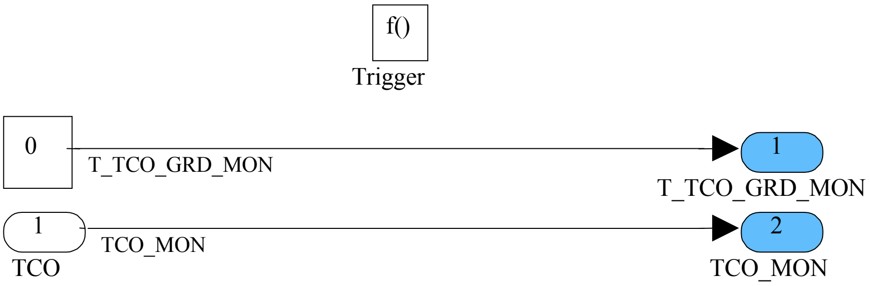


Figure 2 ECM2_DTSYSTCO/ INITIALISATION

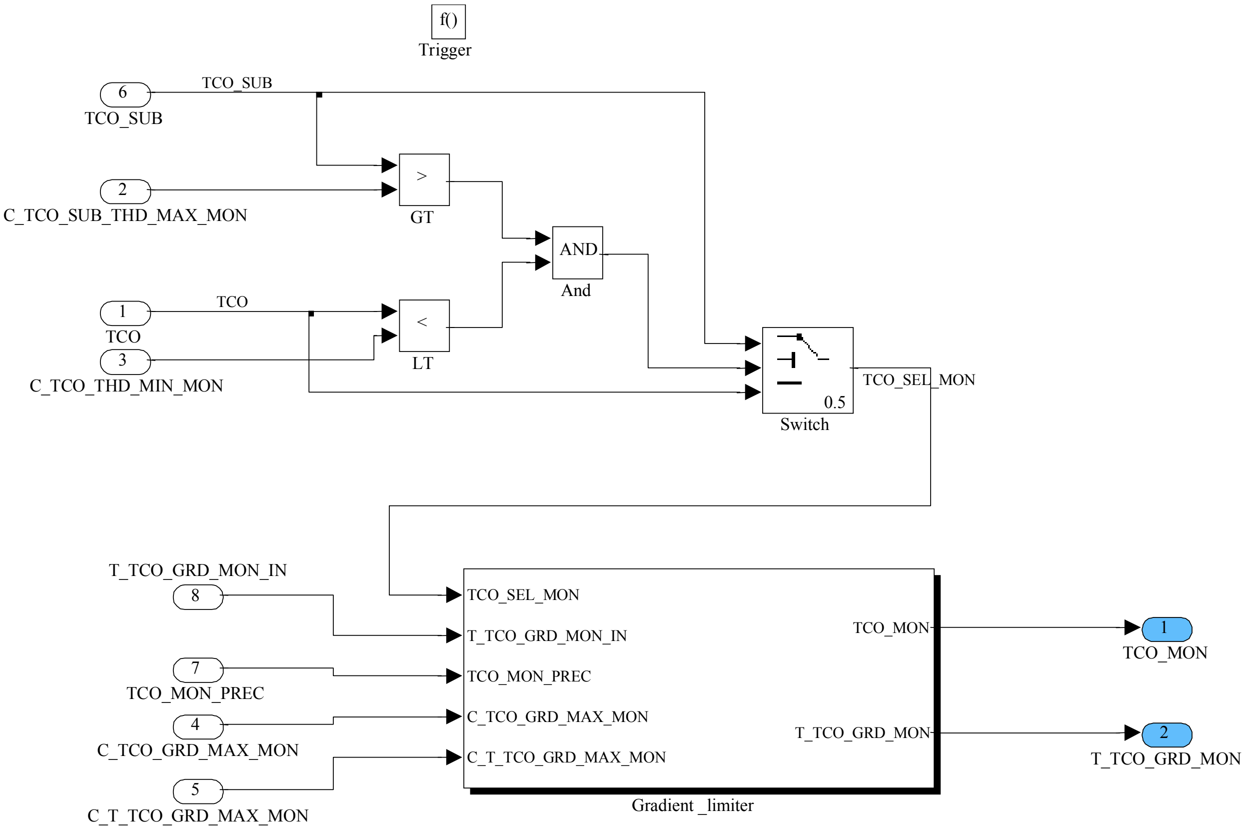


Figure 3 ECM2_DTSYSTCO/ TCO_MONITORING

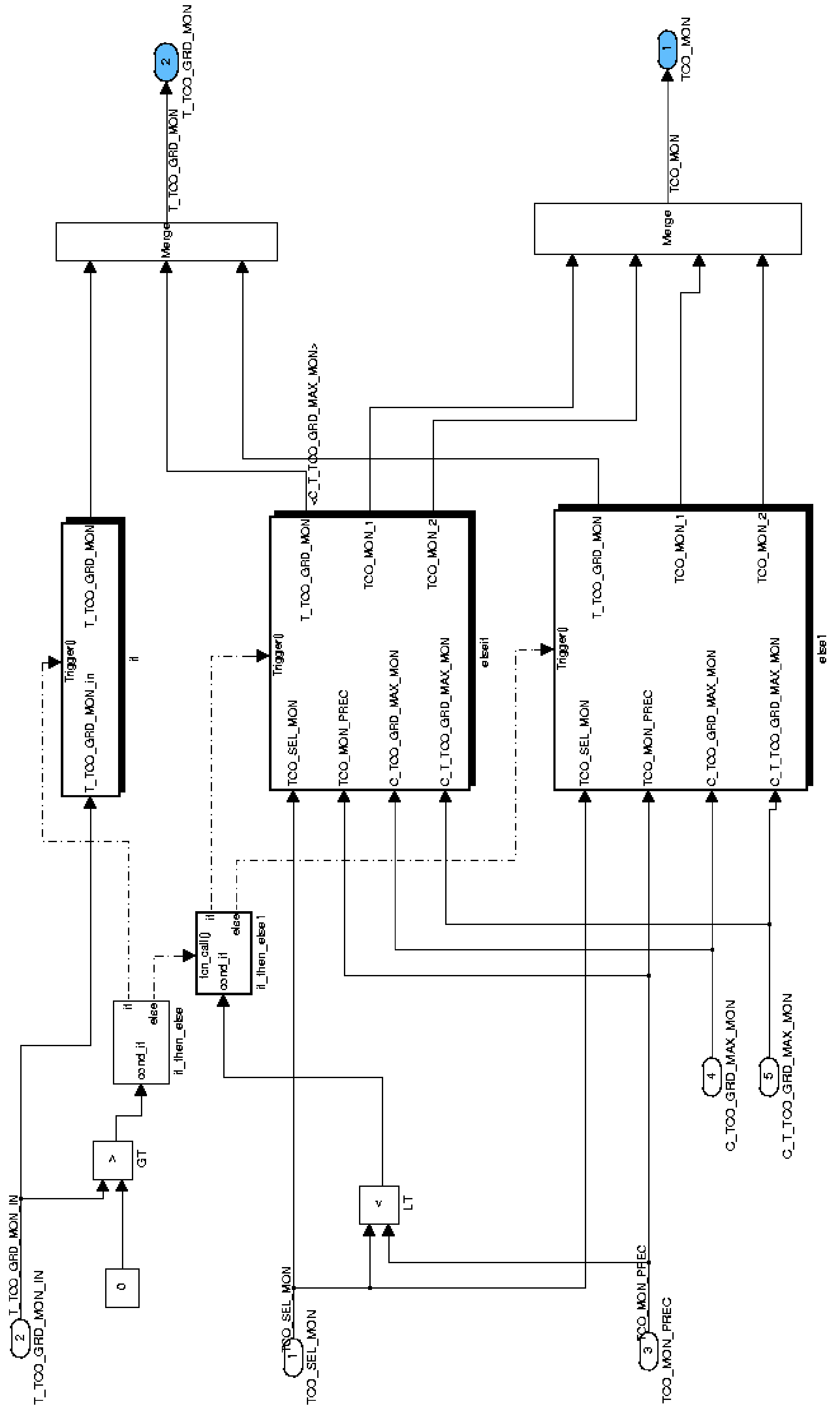


Figure 4 ECM2_DTSYSTCO/ TCO_MONITORING/ GRADIENT _LIMITER

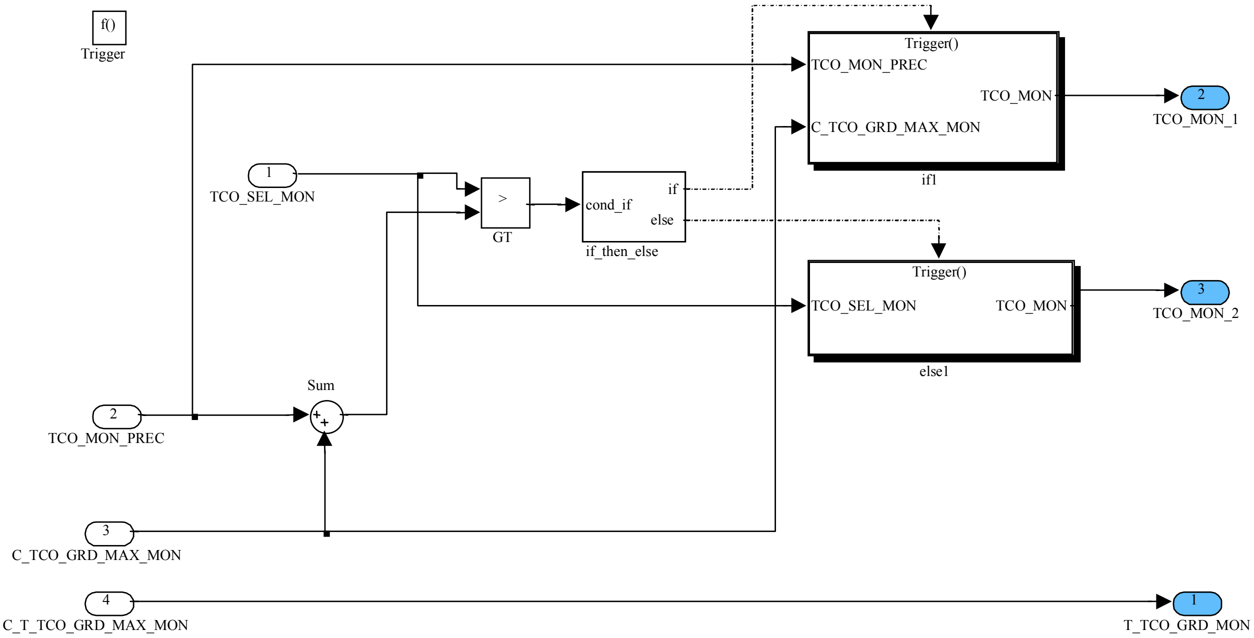


Figure 5 ECM2_DTSYSTCO/ TCO_monitoring/ Gradient _limiter/ else1

Figure 6 ECM2_DTSYSTCO/ TCO_monitoring/ Gradient _limiter/ else1/ else1

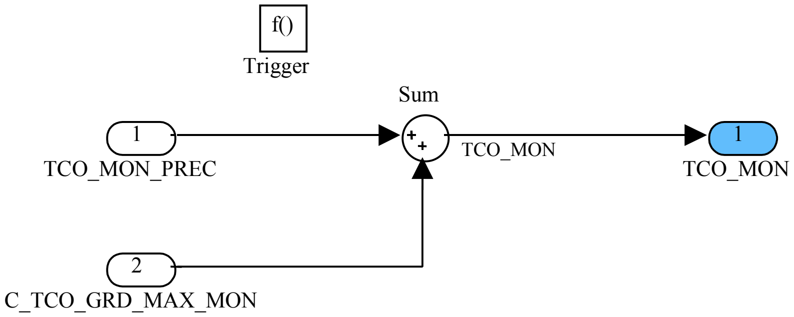


Figure 7 ECM2_DTSYSTCO/ TCO_monitoring/ Gradient _limiter/ else1/ if1

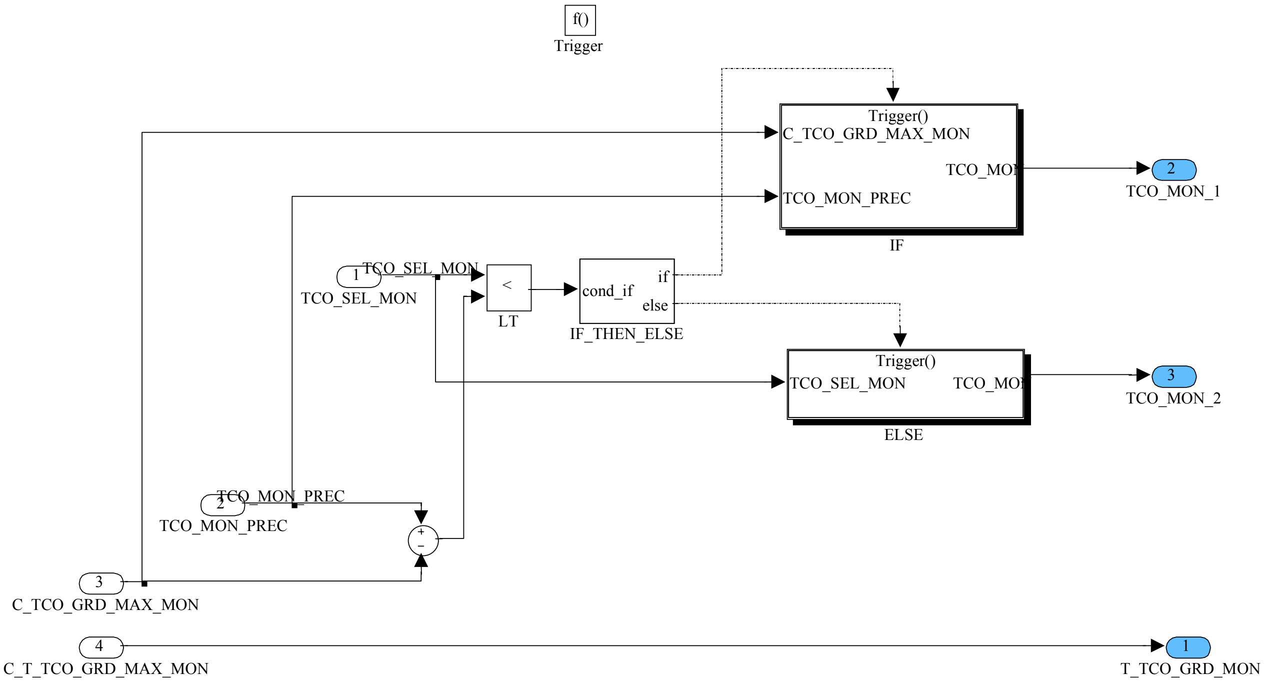


Figure 8 ECM2_DTSYSTCO/ TCO_monitoring/ Gradient _limiter/ elseif

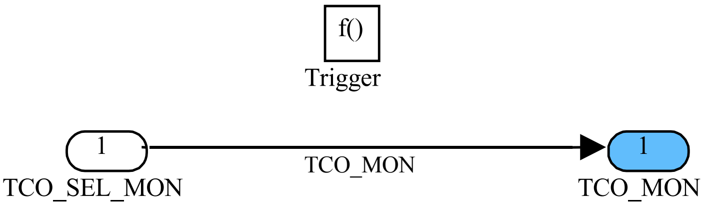


Figure 9 ECM2_DTSYSTCO/ TCO_monitoring/ Gradient _limiter/ elseif/ ELSE

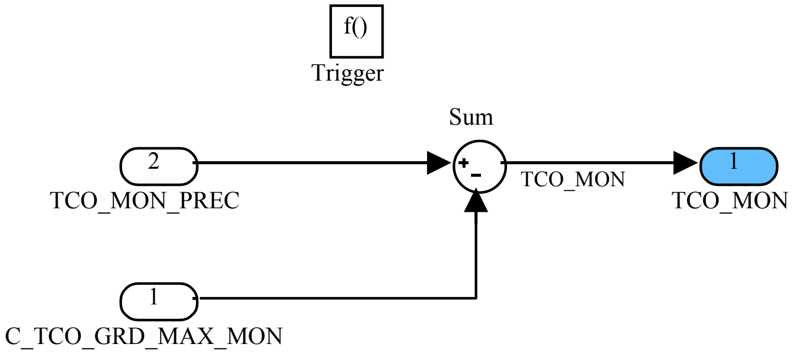


Figure 10 ECM2_DTSYSTCO/ TCO_monitoring/ Gradient _limiter/ elseif/ IF

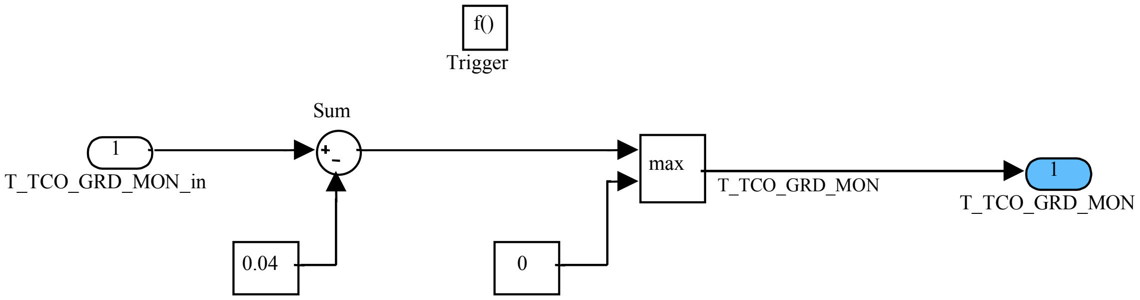


Figure 11 ECM2_DTSYSTCO/ TCO_monitoring/ Gradient _limiter/ if

11.9 Monitoring of torque demand from idle speed controller

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TQ_DIF_I_IS_MON	V/O	0...1H	0...1	1	[-]
Fault currently present in torque generation, symptom "idle speed controller - I-part"					
LV_ERR_TQ_DIF_P_D_IS_MON	V/O	0...1H	0...1	1	[-]
Fault currently present in torque generation, symptom "idle speed controller - PD-part"					
TQ_DIF_I_IS_MON	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Monitored I-part of the idle speed controller					
ABC_TQ_DIF_I_IS_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for monitoring of I_PART of idle speed controller					
TQ_ADD_P_D_IS_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Decrement for modelled PD-part during ramp function					
TQ_DIF_I_IS_DIF_MON	V	80...7FH	-256...254	2	[Nm]
Difference between monitored and modelled I-part					
TQ_DIF_I_IS_RAMP_MON	V	8000...7FFFH	-256...255.99218	0.0078125	[Nm]
Modelled I-part during ramp function					
TQ_DIF_P_D_IS_DIF_MON	V	80...7FH	-256...254	2	[Nm]
Difference between monitored and modelled PD-part					
TQ_DIF_P_D_IS_RAMP_MON	V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Modelled PD-part during ramp function					
ABC_TQ_DIF_P_D_IS_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for monitoring of idle speed controller - PD-part					
LV_RAMP_P_D_BEG_VALUE_WRG_MON	V	0...1H	0...1	1	[-]
Logical bit indicating wrong start value of PD-ramp					
STATE_ERR_TQ_DIF_I_IS_MON	V/O	0H 1H 2H 3H 4H	NO_ERROR ERROR_PATH1 ERROR_PATH2 ERROR_PATH3 ERROR_PATH4	1	[-]
State variable for fault detection in torque generation, symptom ‘idle speed controller – I-part’					
STATE_ERR_TQ_DIF_P_D_IS_MON	V/O	0H 1H 2H 3H 4H	NO_ERROR ERROR_PATH1 ERROR_PATH2 ERROR_PATH3 ERROR_PATH4	1	[-]
State variable for fault detection in torque generation, symptom ‘idle speed controller – PD-part’					
TQ_ADD_I_IS_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Decrement for modelled I-part during ramp function					

Input data:

TQ_DIF_I_IS	TQ_DIF_P_D_IS_MON	LV_REQ_ISC	LV_TQI_MON_ACT_MON
LV_PAS_RAMP_ACT_I_CHG_MON	LV_PAS_RAMP_ACT_P_D_CHG_MON	STATE_PAS_RAMP_ACT_I_IS_MON	STATE_PAS_RAMP_ACT_P_D_IS_MON
STATE_REQ_ISC_MON	TQ_DIF_P_D_IS_MAX_MON	FAC_RAMP_NEG_P_D_IS_MON	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TQ_DIF_I_IS_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_TQ_DIF_I_IS_MON	1	1...FFH	1...255	1	[-]
Maximum of anti bounce counter					
C_TQ_DIF_I_IS_MAX_MON	1	0...FFH	0...255	1	[Nm]
Maximum I-part					
C_TQ_I_ISC_NOT_ACT_MAX_MON	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Maximum I-Part allowed when ISC not active					
C_TQ_PAS_I_IS_MON	1	0...FFH	0...255	1	[Nm]
Target value for monitoring ISC deactivation-ramp operation					
C_TQ_DIF_I_IS_ADD_MON	1	0...FFH	0...255	1	[Nm]
Additive constant for I-part model value					
C_FAC_RAMP_NEG_I_IS_MON	1	0...FA0H	0...0.5	0.000125	[-]
Factor for monitoring negative ISC deactivation-ramp of I-part					
C_ABC_INC_TQ_DIF_P_D_IS_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_TQ_DIF_P_D_IS_MON	1	1...FFH	1...255	1	[-]
Maximum of anti bounce counter					
C_TQ_P_D_ISC_NOT_ACT_MAX_MON	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Maximum PD-part allowed when ISC not active					
C_TQ_DIF_P_D_IS_ADD_MON	1	0...FFH	0...255	1	[Nm]
Additive constant for PD-part model value					
C_TQ_DIF_P_D_RAMP_CHK_OFS_MON	1	0...FFH	0...255	1	[Nm]
Offset for monitoring ramp start value					

Import actions:

ACTION_ECM3_Service12TaskPfm(IN <>)
ACTION_ECM3_Service13TaskPfm(IN <>)
ACTION_ECM3_Service14TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 1.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service12TaskPfm() and ACTION_ECM3_Service13TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: These actions are defined in chapter "Processor Monitroing", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

Function Description

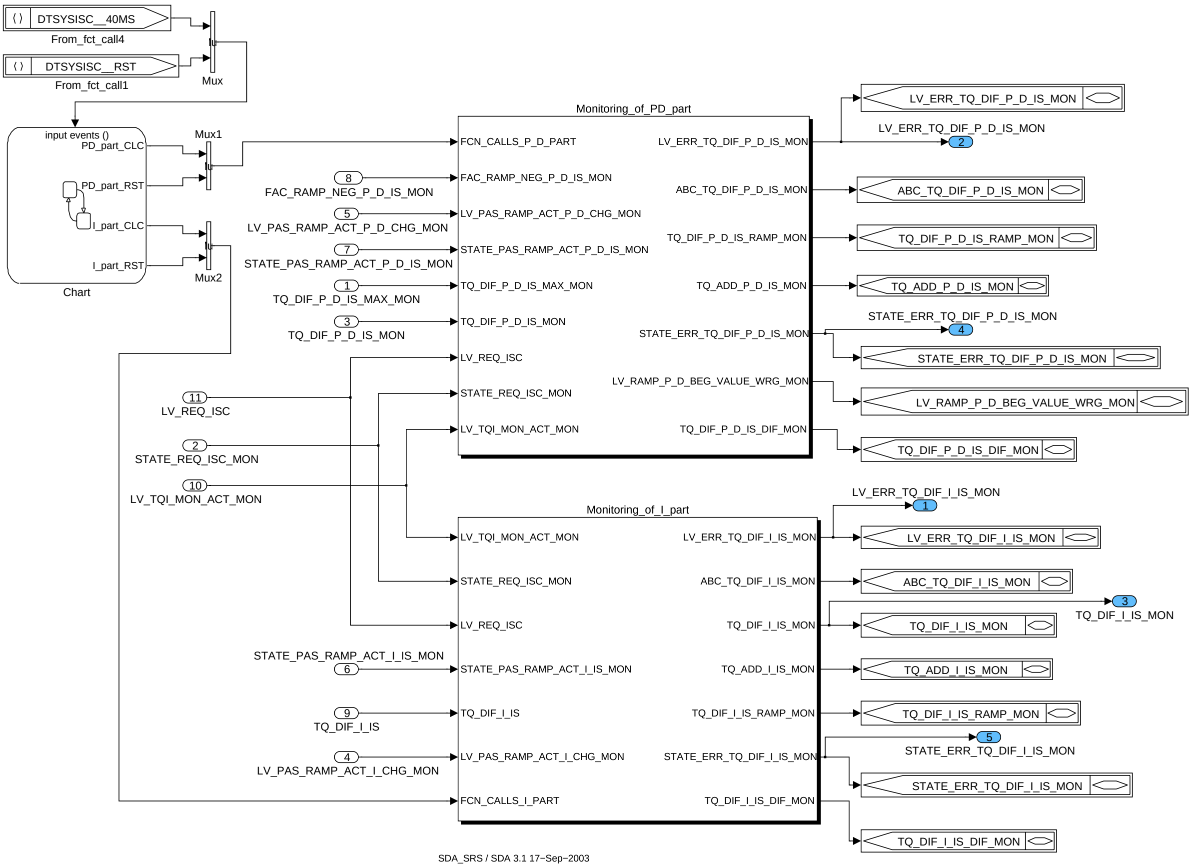


Figure 2 ECM2_DTSYSISC

11.9.1 Monitoring of PD controller

General information:

The objective of this function is the monitoring of the torque demand from the PD-part of the idle speed controller function. Depending on the speed deviation to the idle speed setpoint and the coolant temperature a maximum PD-part is generated (the worst case values for the PD-part in the function level, i.e. great PD-part values are allowed for the most negative speed gradients => for N < N_SP_IS_MON and for N > N_SP_IS_MON small PD-part values are allowed). To additionally monitor the activation of the idle-speed controller in L1, a model value STATE_REQ_ISC_MON) of the activation status is calculated in the module ‘Application incidences of process monitoring’. To effectively plausiblise the values of TQ_DIF_P_D_IS and TQ_DIF_I_IS from the function level, both the monitoring thresholds as well as the activation condition model value are used.

The functionality that is depicted in the following pages consists of two fault scenarios:

Error Path – ISC not active:

Case 1: If the ISC is in the ramp limit operation due to deactivation of ISC, this ramp function is monitored. If LV_PAS_RAMP_ACT_P_D_CHG_MON is 1(i.e. indicates a change of STATE_PAS_RAMP_ACT_P_D_IS(0) from 0 to 1 or a flank change between two L2 sample times), the ramp calculation is triggered. It is first checked if the L1 ramp start-value is below the maximum PD-part. If this is the case, a ramp decrement TQ_ADD_P_D_ IS_MON is calculated and the L1 start-value,TQ_DIF_P_D_IS_MON, plus a calibratable additive constant is copied to TQ_DIF_P_D_IS_RAMP_MON. During the ramp operation, the ramp model value is calculated by adding the ramp decrement to the model value ($TQ\_DIF\_P\_D\_IS\_RAMP\_MON_{k-1}$) from the previous recurrency. If however the ramp start-value is greater than the threshold, TQ_DIF_P_D_IS_MAX_MON, then LV_RAMP_P_D_BEG_VALUE_WRG_MON is set to one. The difference between monitored and modelled PD-part is set as a visible variable for application work.

In the error detection part of this functionality, either $TQ\_DIF\_P\_D\_IS\_MON_k >$ TQ_DIF_P_D_IS_RAMP_MON or LV_RAMP_P_D_BEG_VALUE_WRG_MON==1 will lead to the error flag STATE_ERR_TQ_DIF_P_D_IS_MON being set.

Function Description

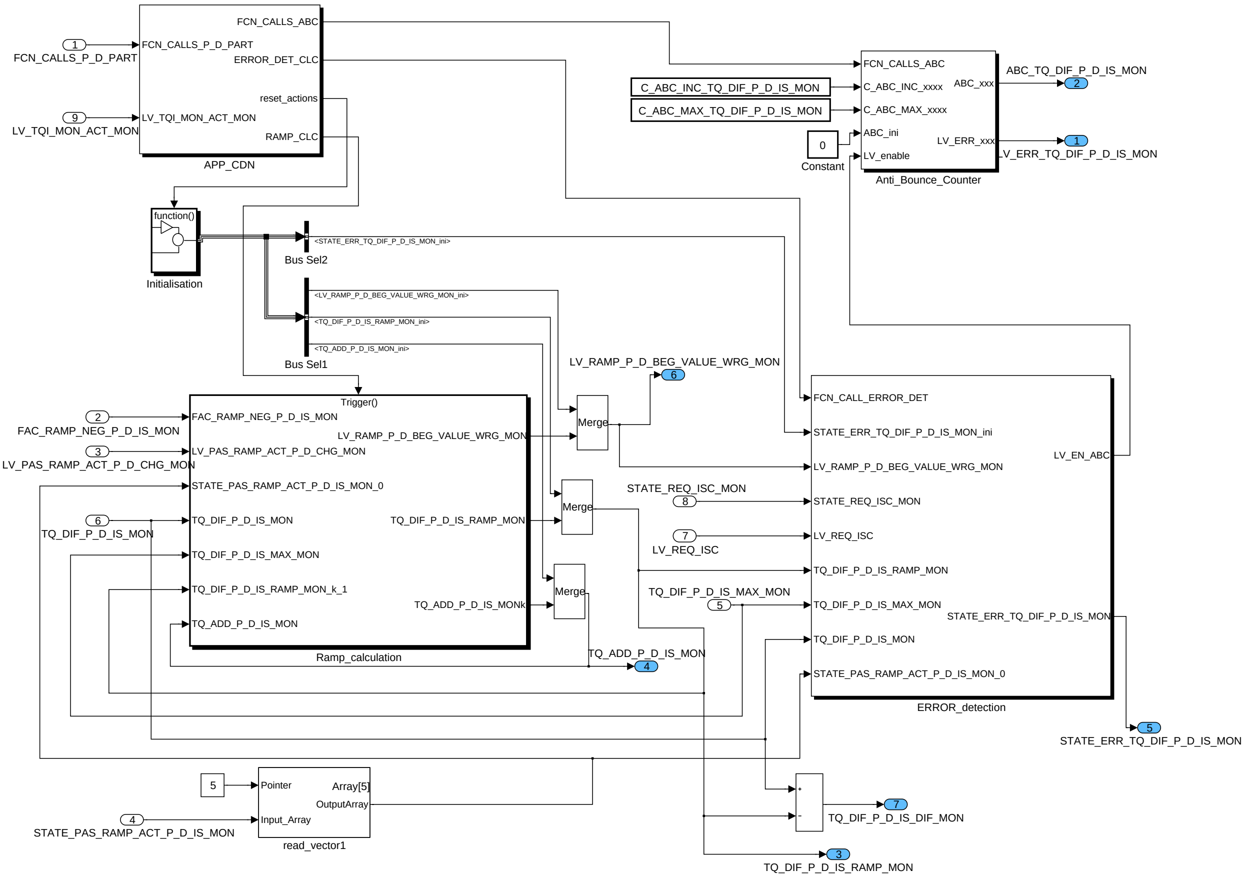


Figure 3 ECM2_DTSYSISC/ Monitoring_of_PD_part

Case 2: If the deactivation ramp is over, a torque demand from the PD-part of the controller greater than the passive value of zero indicates an error in the function level.

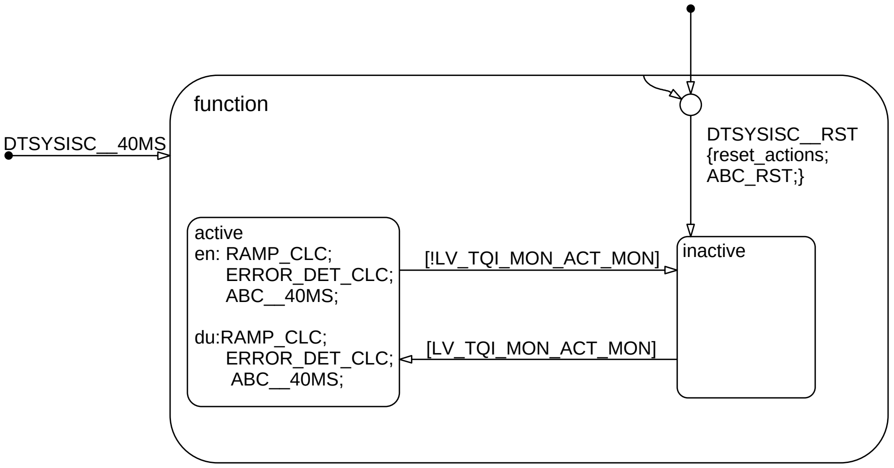
Error Path – ISC active:

Case 1: The ISC is when the engine is in idle state(LV_IS=1) or under certain conditions also in trailing-throttle(LV_PU=1) or part-load(LV_PL=1). So all of these possible "ISC-active" phases have to be monitored. A fault is set, if L1 declares the ISC be active and if L2 also detects the same based on plausiblised activation conditions (STATE_ REQ_ISC_MON = IDLE or TRAILING_THR or PART_LOAD) and if the monitored PD-part TQ_DIF_P_D_IS_MON exceeds the maximum PD-part TQ_DIF_P_D_IS_MAX_ MON.

Case 2: If in L1 the ISC is active(LV_REQ_ISC=1) but the activation conditions in L2 are not fulfilled i.e. STATE_ REQ_ISC_MON ≠ IDLE or TRAILING_THR or PART_LOAD, then if TQ_DIF_P_D_IS_MON is greater than a threshold value (C_TQ_P_D_ISC_NOT_ ACT_MAX_MON), the error-flag STATE_ERR_TQ_DIF_P_D_IS_MON is set .

If one of the errors above is detected, a fault is declared after debouncing by the idle speed controller montoring.

Application Conditions



Initialisation (DTSYSISC__RST)**:** for condition see ‘Application Incidences of Process Monitoring’.

Figure 4 ECM2_DTSYSISC/ Monitoring_of_PD_part/ APP_CDN/ Chart

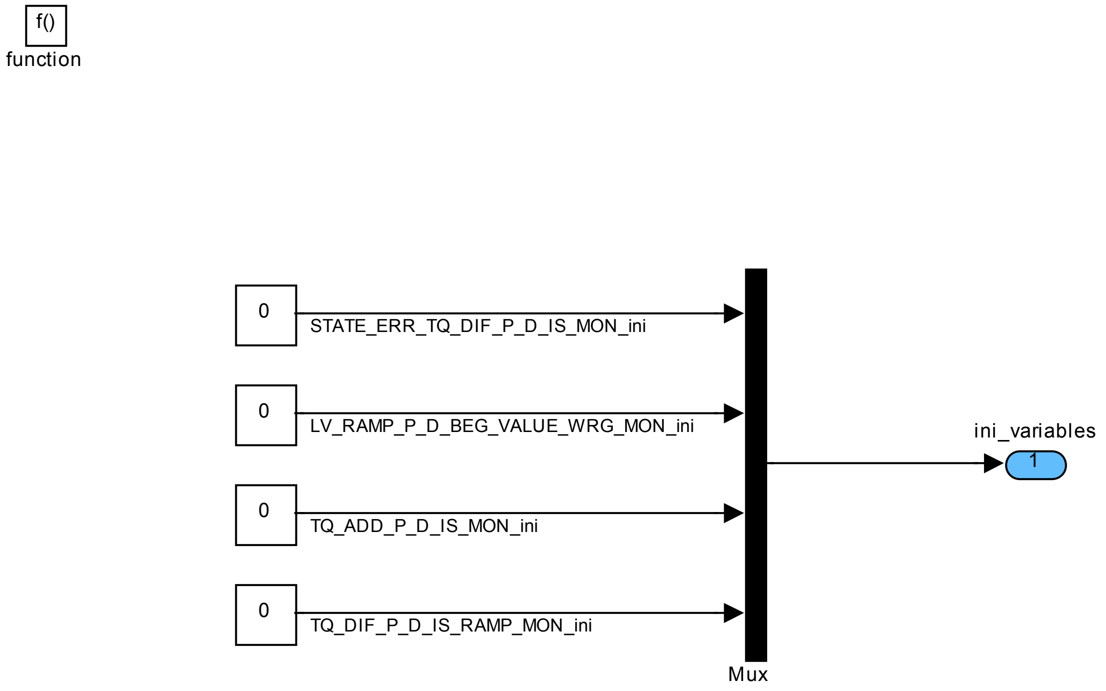


Figure 5 ECM2_DTSYSISC/ Monitoring_of_PD_part/ Initialisation

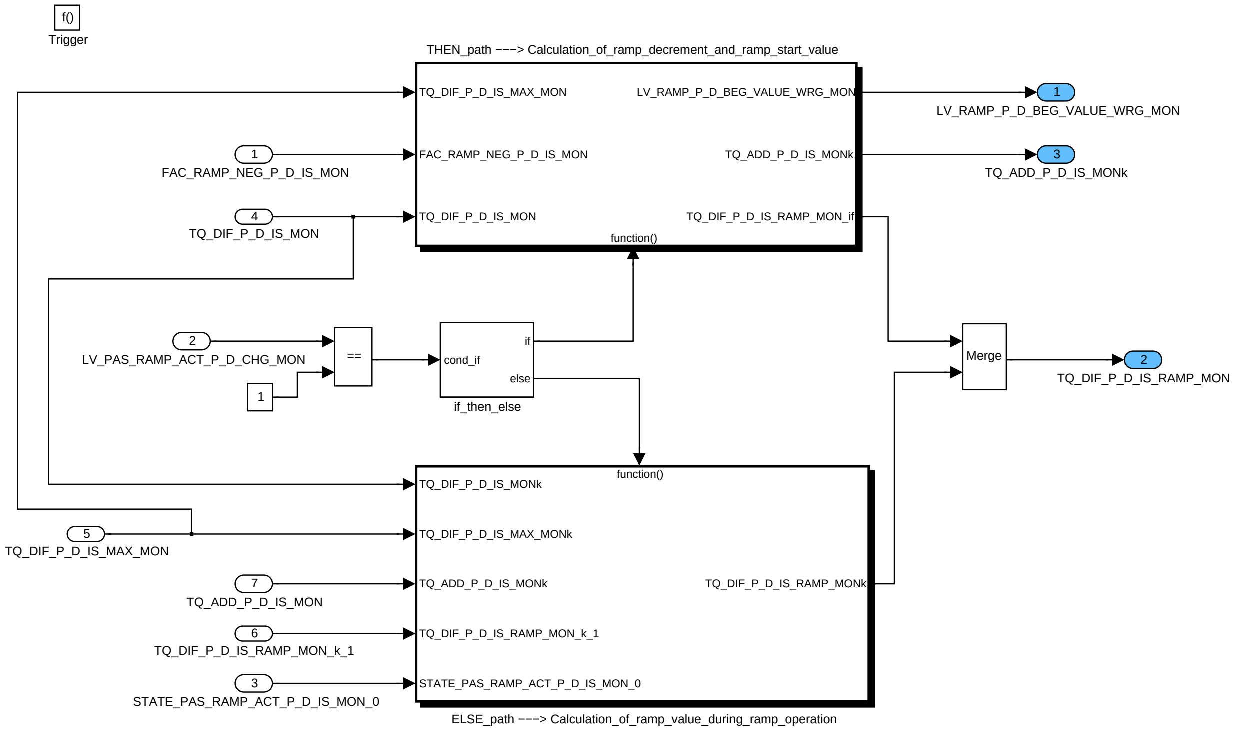


Figure 6 ECM2_DTSYSISC/ Monitoring_of_PD_part/ Ramp_calculation

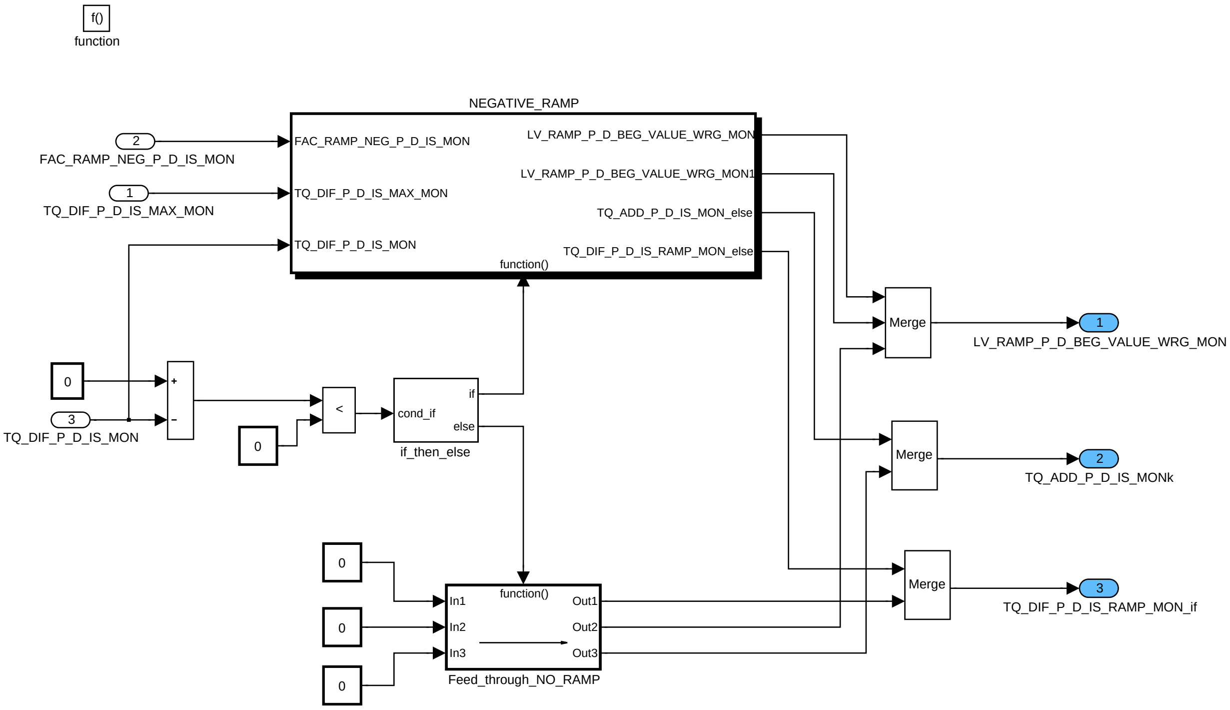
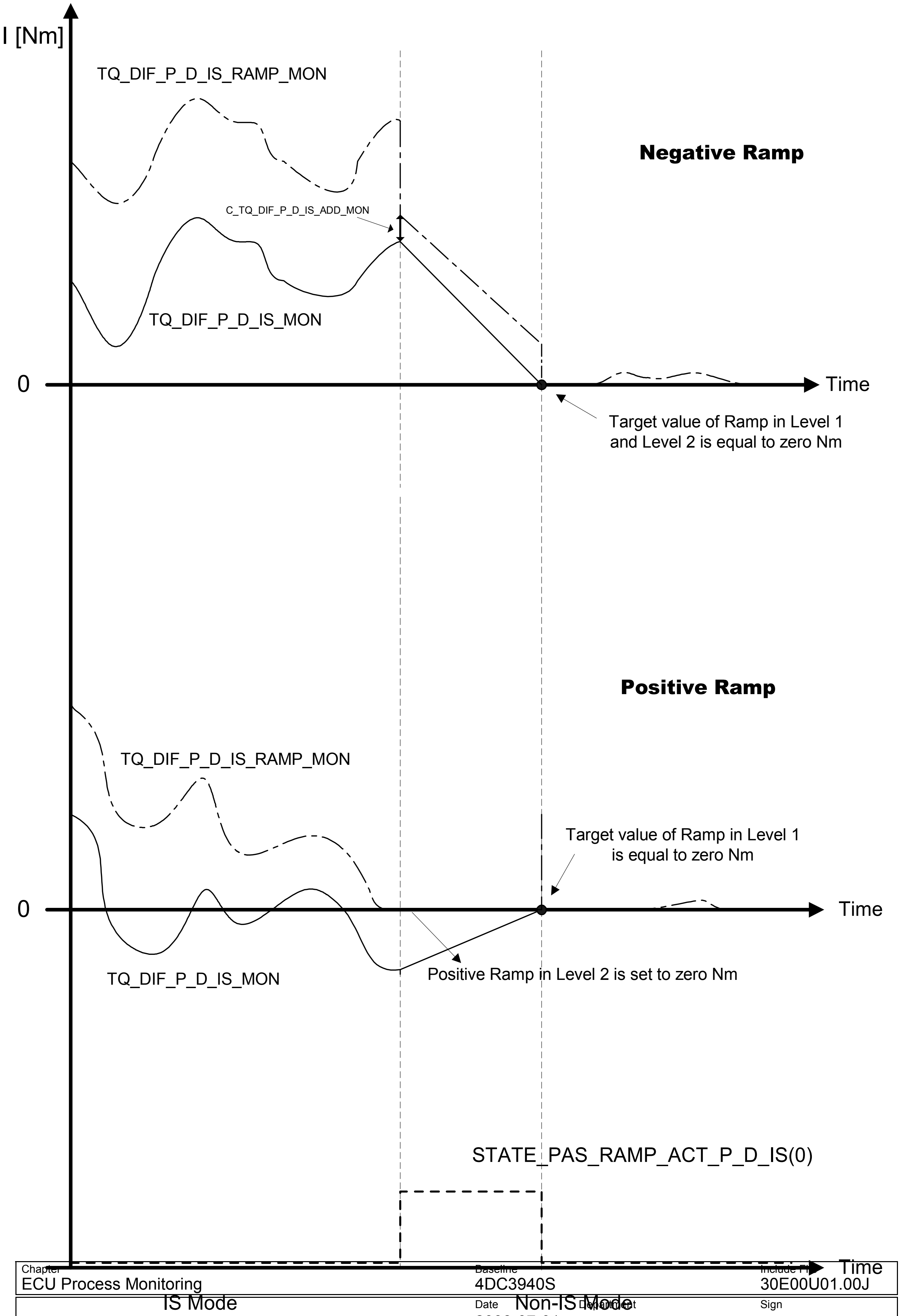


Figure 7 ECM2_DTSYSISC/ Monitoring_of_PD_part/ Ramp_calculation/ THEN_path

NOTE: The minimum PD-ramp increment/decrement is equal to the resolution of TQ_ADD_P_D_ IS_MON. If the theoretically calculated *absolute value* of TQ_ADD_P_D_IS_ MON **is *smaller* than the resolution and not equal to zero**, then is to be limited to the resolution. This would lead to a slower/faster descent of the deactivation ramp i.e. a shorter ramp time. If the theoretical value of TQ_ADD_P_D_IS_MON is equal to zero, this value is then maintained.


I [Nm]
TQ_DIF_P_D_IS_RAMP_MON
Negative Ramp
C_TQ_DIF_P_D_IS_ADD_MON
TQ_DIF_P_D_IS_MON
0
Time
Target value of Ramp in Level 1 and Level 2 is equal to zero Nm
Positive Ramp
TQ_DIF_P_D_IS_RAMP_MON
Target value of Ramp in Level 1 is equal to zero Nm
0
Time
TQ_DIF_P_D_IS_MON
Positive Ramp in Level 2 is set to zero Nm
STATE_PAS_RAMP_ACT_P_D_IS(0)
Time
IS Mode
Non-IS Mode

For e.g: theoretical value = -0.04 Nm ---> Limited value = -0.03125 Nm
theoretical value = -0.01 Nm ---> Limited value = -0.03125 Nm

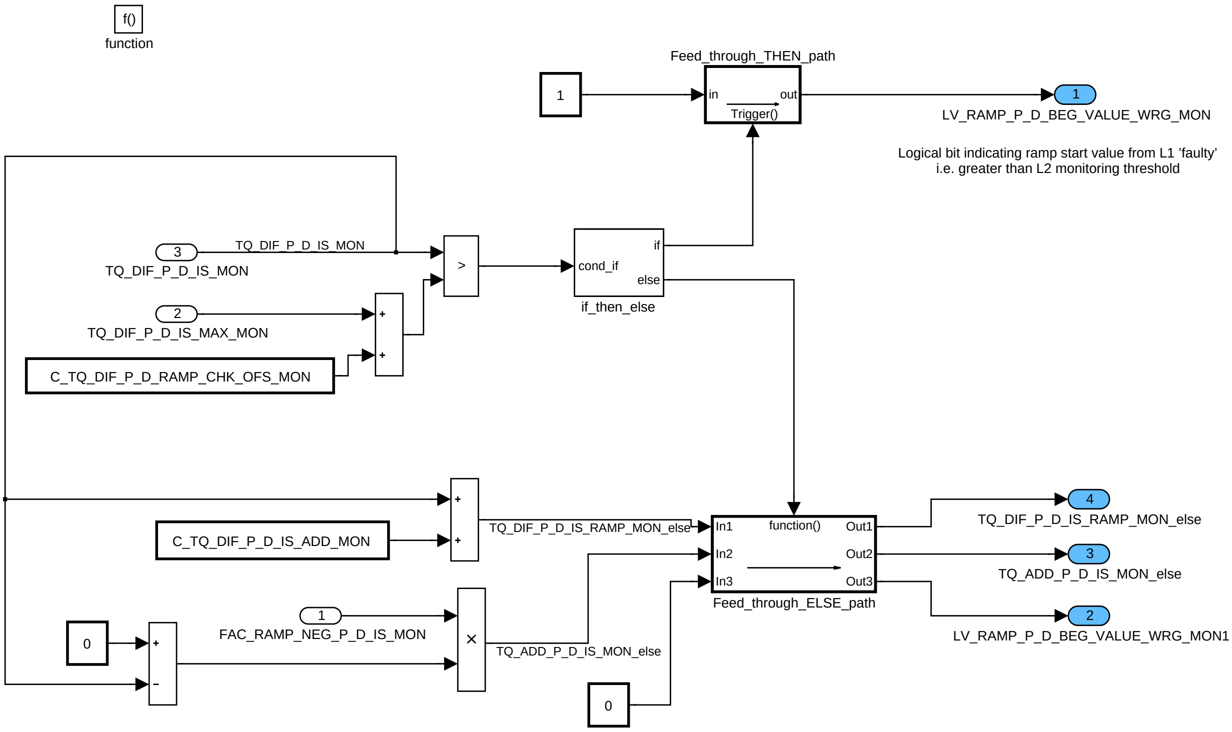


Figure 8 ECM2_DTSYSISC/ Monitoring_of_PD_part/ Ramp_calculation/ THEN_path/ NEGATIVE_RAMP

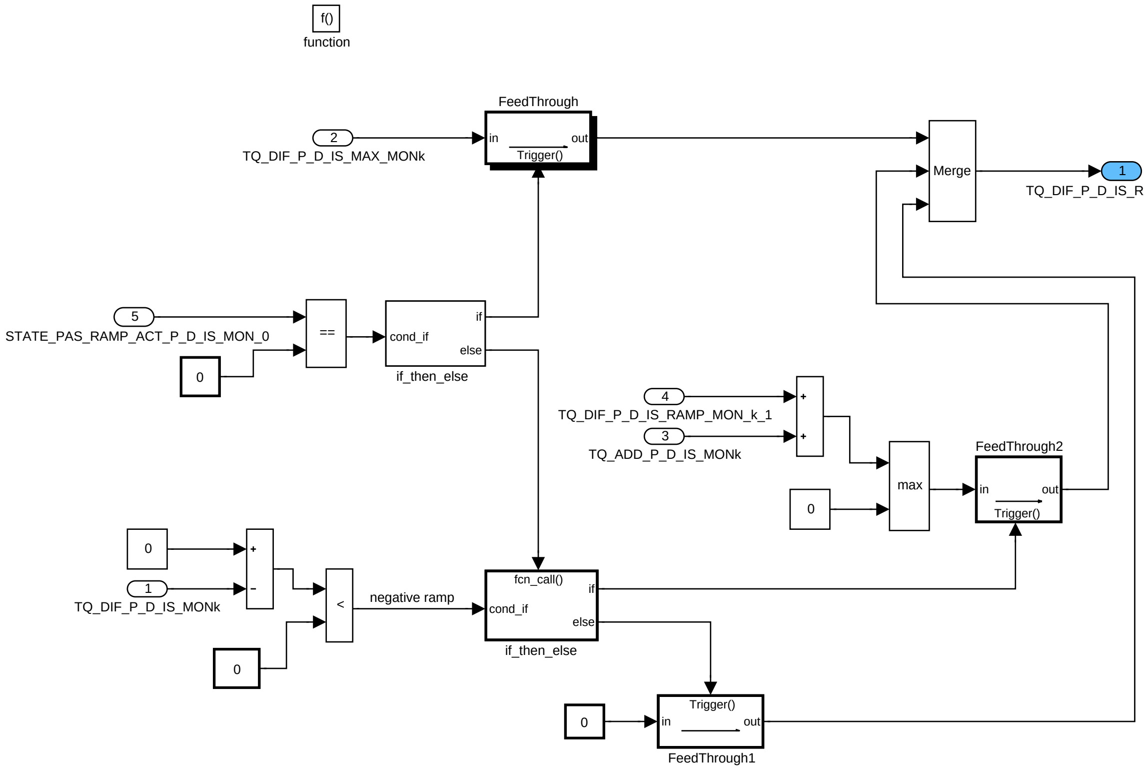


Figure 9 ECM2_DTSYSISC/ Monitoring_of_PD_part/ Ramp_calculation/ ELSE_path

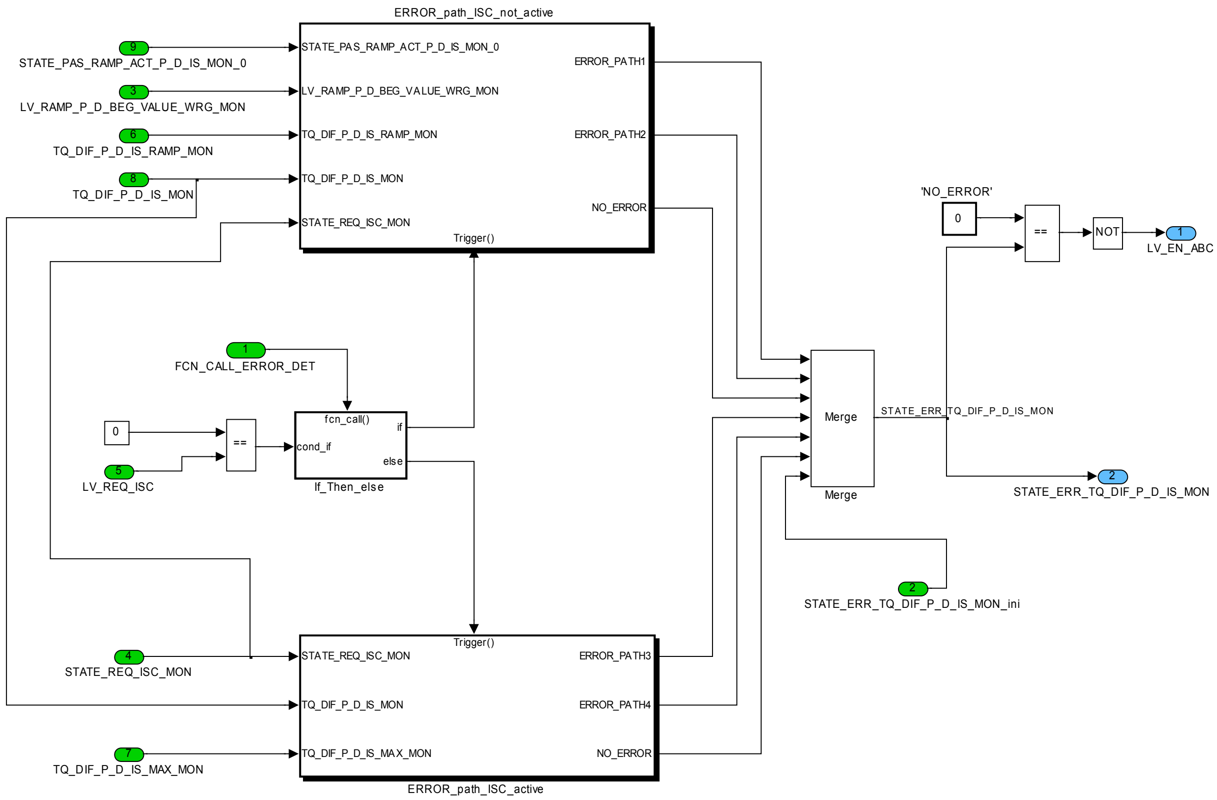


Figure 10 ECM2_DTSYSISC/ Monitoring_of_PD_part/ ERROR_detection

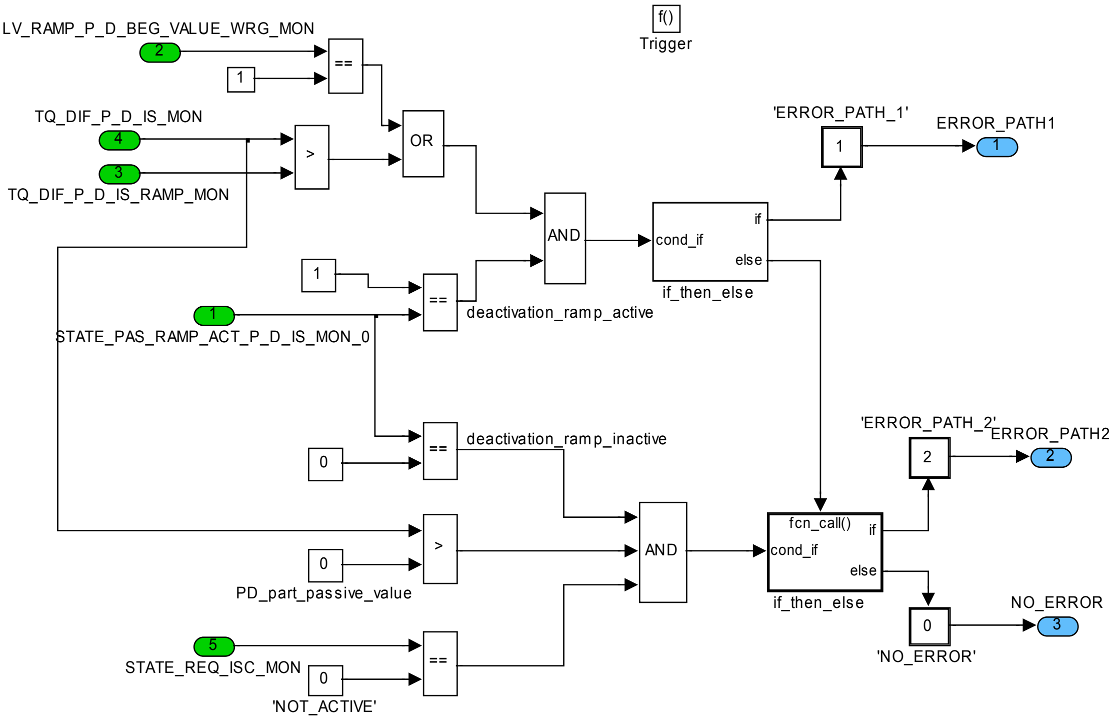


Figure 11 ECM2_DTSYSISC/ Monitoring_of_PD_part/ ERROR_detection/ ERROR_path_ISC_not_active

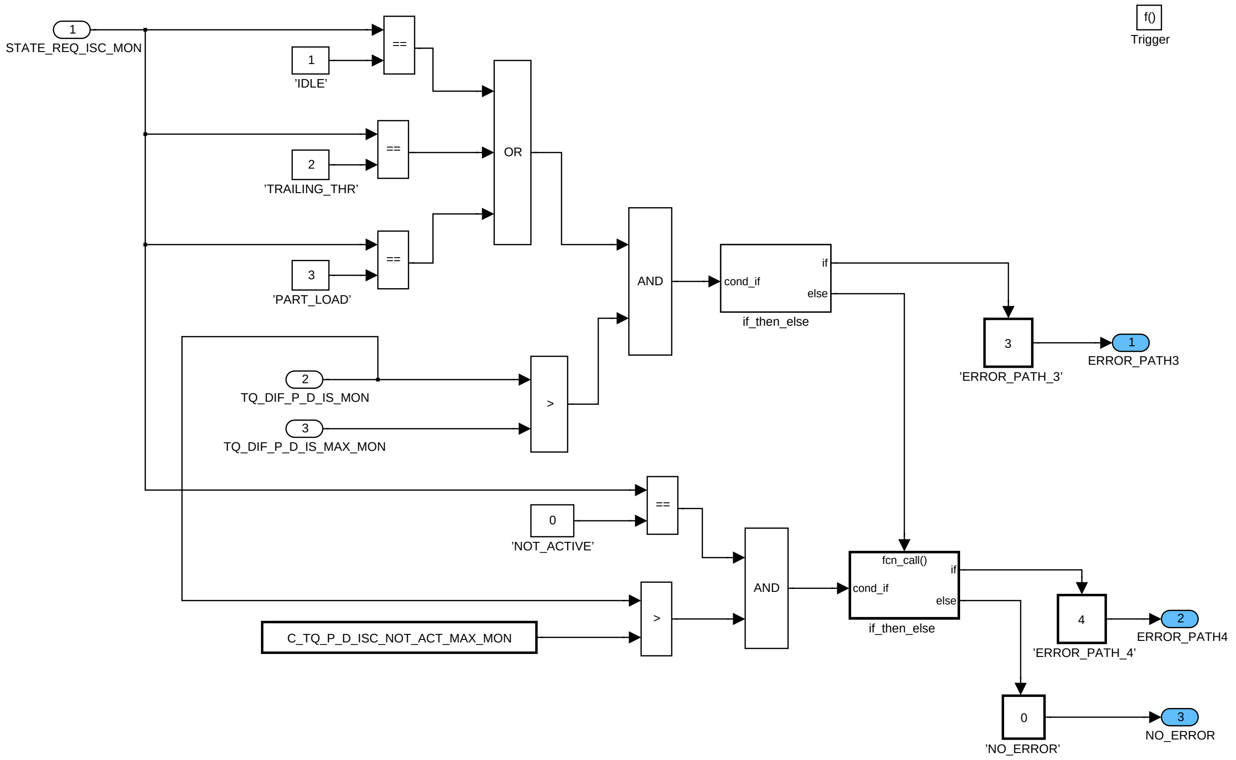


Figure 12 ECM2_DTSYSISC/ Monitoring_of_PD_part/ ERROR_detection/ ERROR_path_ISC_active

11.9.2 Monitoring of I controller

General information:

The objective of this function is the monitoring of the torque demand from the I-part of the idle speed controller function. Two different faults scenarios are considered:

Error Path – ISC not active:

Case 1: If the ISC is actually in the ramp limit operation due to deactivation of ISC, this ramp function is monitored. If LV_PAS_RAMP_ACT_I_CHG_MON is 1(i.e. indicates a change of STATE_PAS_RAMP_ACT_I_IS(0) from 0 to 1 or a flank change between two L2 sample times), a ramp decrement (TQ_ADD_I_IS_MON) is calculated and the actual I-part (TQ_DIF_I_IS_MON) plus a calibratable additive constant is copied to TQ_DIF_I_IS_RAMP_MON. Further on, TQ_DIF_I_IS_RAMP_MON is calculated by adding the calculated ramp decrement to the last delayed value. A fault is set, if TQ_DIF_I_IS_MON > TQ_DIF_I_IS_RAMP_MON or greater than the monitoring threshold C_TQ_DIF_I_IS_MAX_MON. If one of the errors is detected, a fault is declared after debouncing. The difference between monitored and modelled value is declared as a visible variable for application purposes.

Case 2: If in L2 idle speed controller is not active i.e. STATE_ REQ_ISC_MON = NOT_ ACTIVE but the I-part in the function level has not reached the passive value C_TQ_PAS_ I_IS_MON at the end the deactivation ramp, then an error is declared.

Error Path – ISC active:

Case 1: The ISC can be active either when the engine is in idle state (LV_IS=1) or under certain conditions in trailing-throttle state (LV_PU=1) and part-load state (LV_PL=1).So all of these possible “ISC-active” phases have to be monitored. A fault is set, if L1 declares the ISC be active and if L2 also detects the same based on plausiblised activation conditions (STATE_

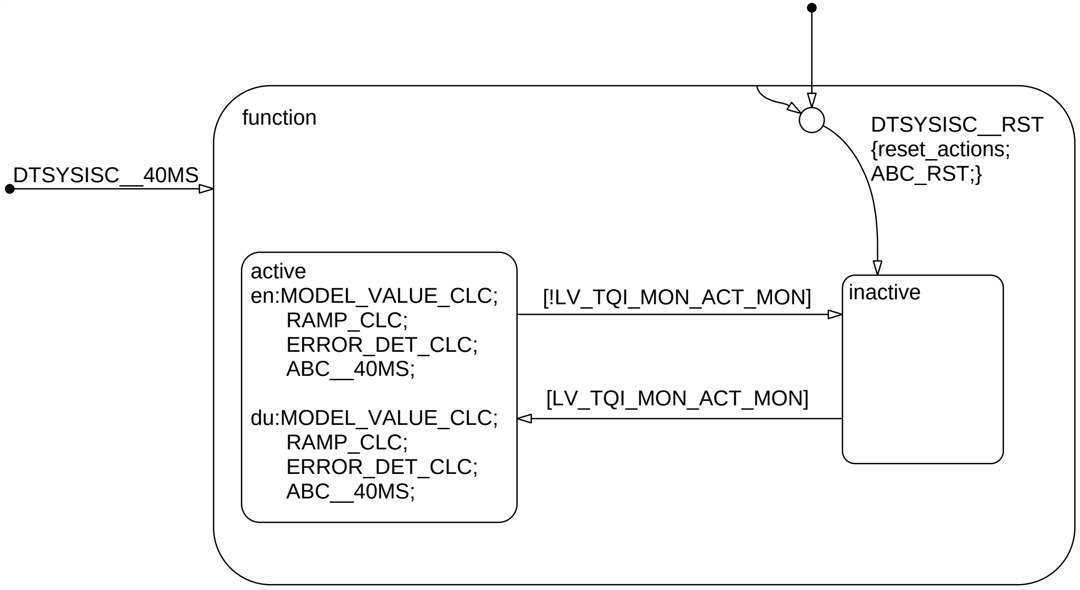
REQ_ISC_MON = IDLE or TRAILING_THR or PART_LOAD) and if torque demand fromL1, TQ_DIF_I_IS_MON, exceeds a threshold value C_TQ_DIF_I_IS_MAX _MON.

Case 2: If in L1 the ISC is active (LV_REQ_ISC=1) but the monitoring functionality detects this activation to be 'faulty' i.e. (STATE_ REQ_ISC_MON ≠ IDLE or TRAILING_THR or PART_LOAD), then an I-part torque demand from L1 greater than a threshold value (C_TQ_I_ISC_NOT _ACT_MAX_MON) will lead to an error detection.

If one of the errors above is detected, a fault is declared after debouncing by the idle speed controller montoring.

Note: The value of STATE_PAS_RAMP_ACT_I_IS (bit coded byte) and STATE_PAS_RAMP_ ACT_P_D_IS(bit coded byte) for the current recurrency (represented by STATE_PAS_RAMP_ACT_I_IS_0 and STATE_PAS_ RAMP_ACT_P_D_IS_0 respectively in the signal flow diagrams) are stored in the bit position 1.

Application Conditions



***Initialisation (DTSYSISC__RST)*:** for condition see 'Application Incidences of Process Monitoring'.

Figure 13 ECM2_DTSYSISC/ Monitoring_of_I_part/ APP_CDN/ Chart

Function Description

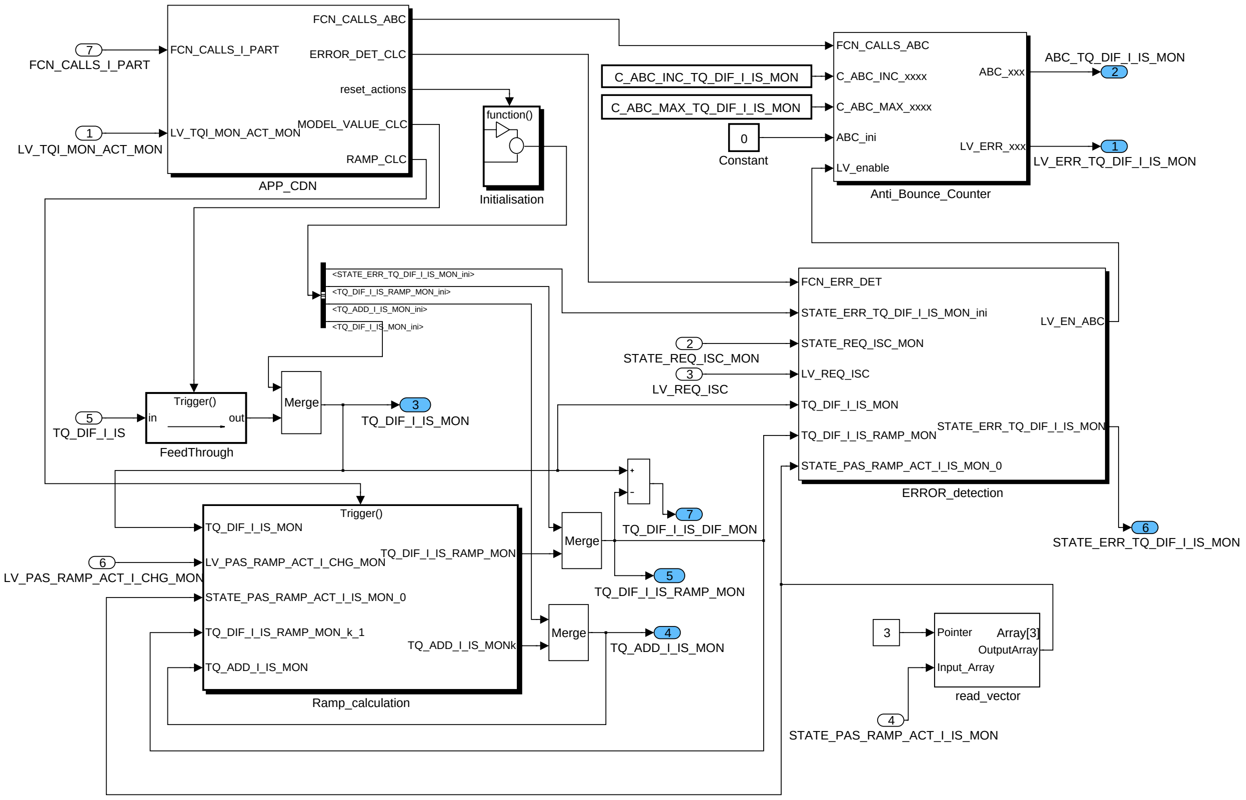


Figure 14 ECM2_DTSYSISC/ Monitoring_of_I_part

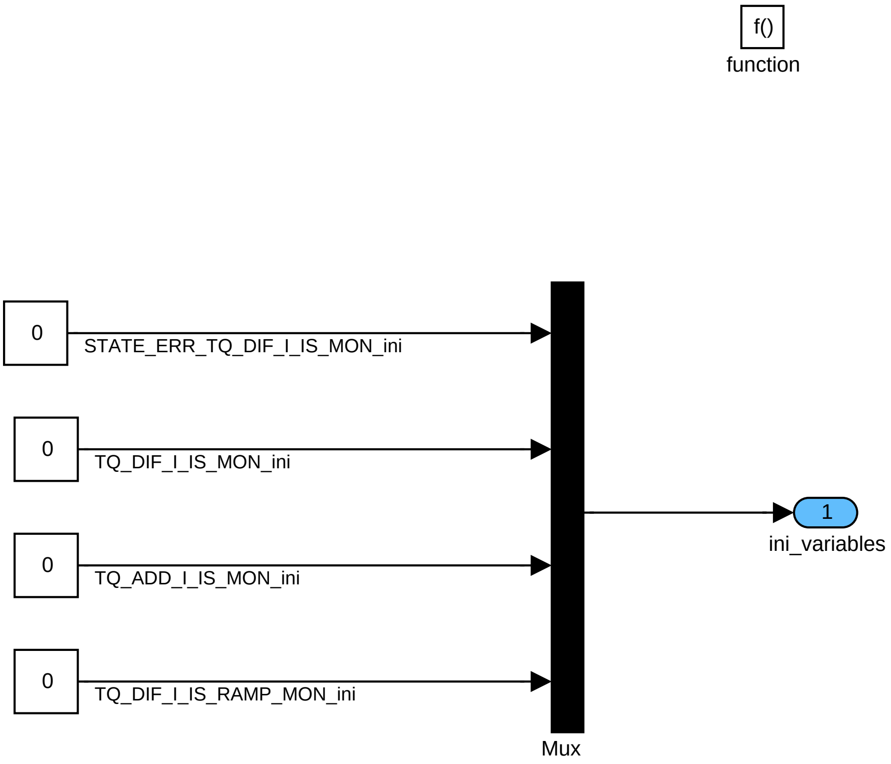


Figure 15 ECM2_DTSYSISC/ Monitoring_of_I_part/ Initialisation

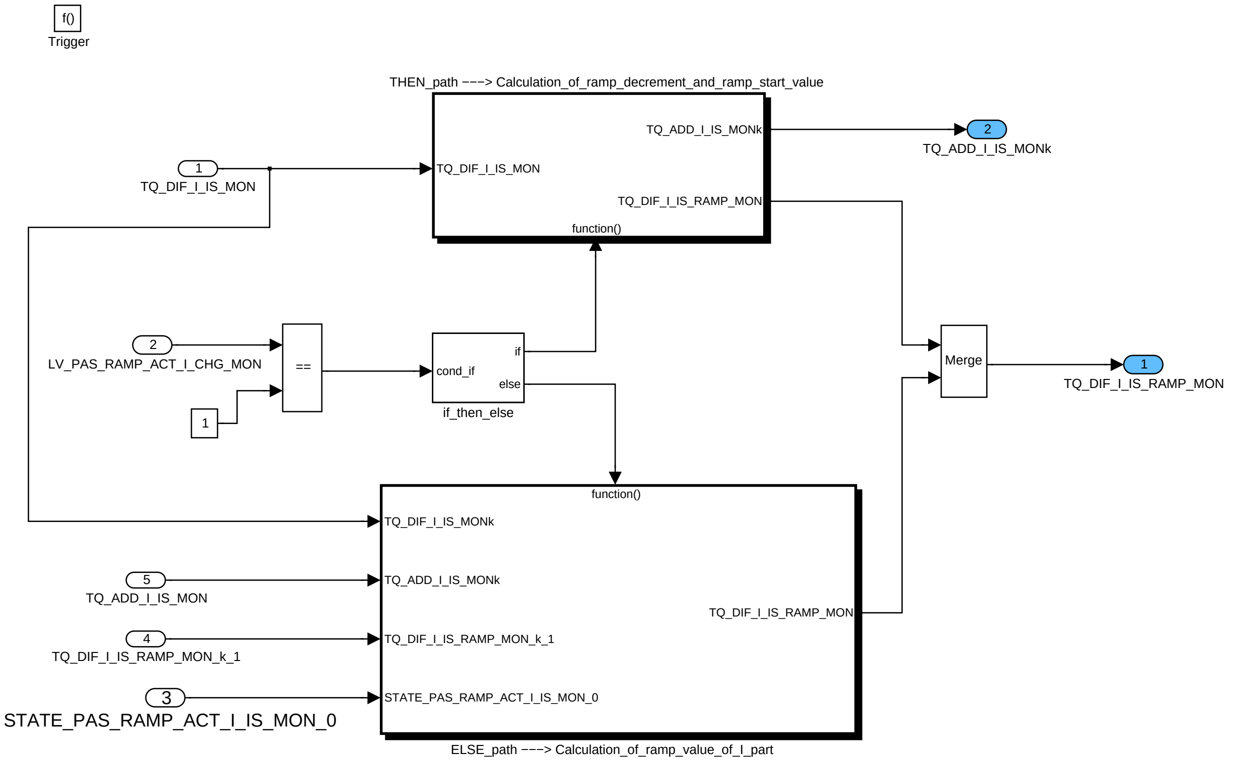


Figure 16 ECM2_DTSYSISC/ Monitoring_of_I_part/ Ramp_calculation

Note: The minimum I-ramp increment/decrement is equal to the resolution of TQ_ADD_I_IS_MON (0.03125Nm). If the theoretically calculated **absolute value** of TQ_ADD_I_IS_MON is ***smaller* than the resolution and not equal to zero**, then is to be limited to the resolution.This would lead to a faster ascent/descent of the deactivation ramp i.e. shorter ramp time. If the theoretical value of TQ_ADD_I_IS_MON is equal to zero, this value is then maintained.

For e.g: theoretical value = -0.04 Nm ---> Limited value = -0.03125 Nm
theoretical value = -0.01 Nm ---> Limited value = -0.03125 Nm
theoretical value = 0.04 Nm ---> Limited value = 0.0625 Nm
theoretical value = 0.01 Nm ---> Limited value = 0.03125 Nm

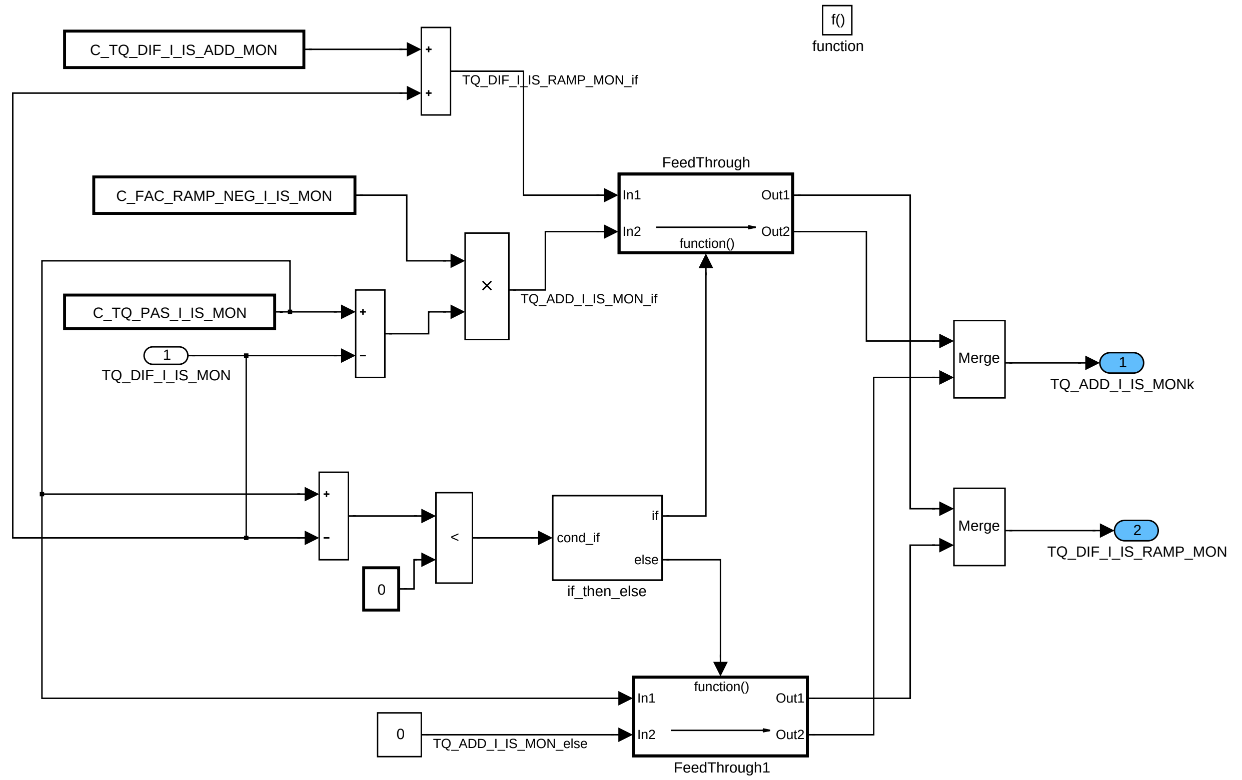


Figure 17 ECM2_DTSYSISC/ Monitoring_of_I_part/ Ramp_calculation/ THEN_path

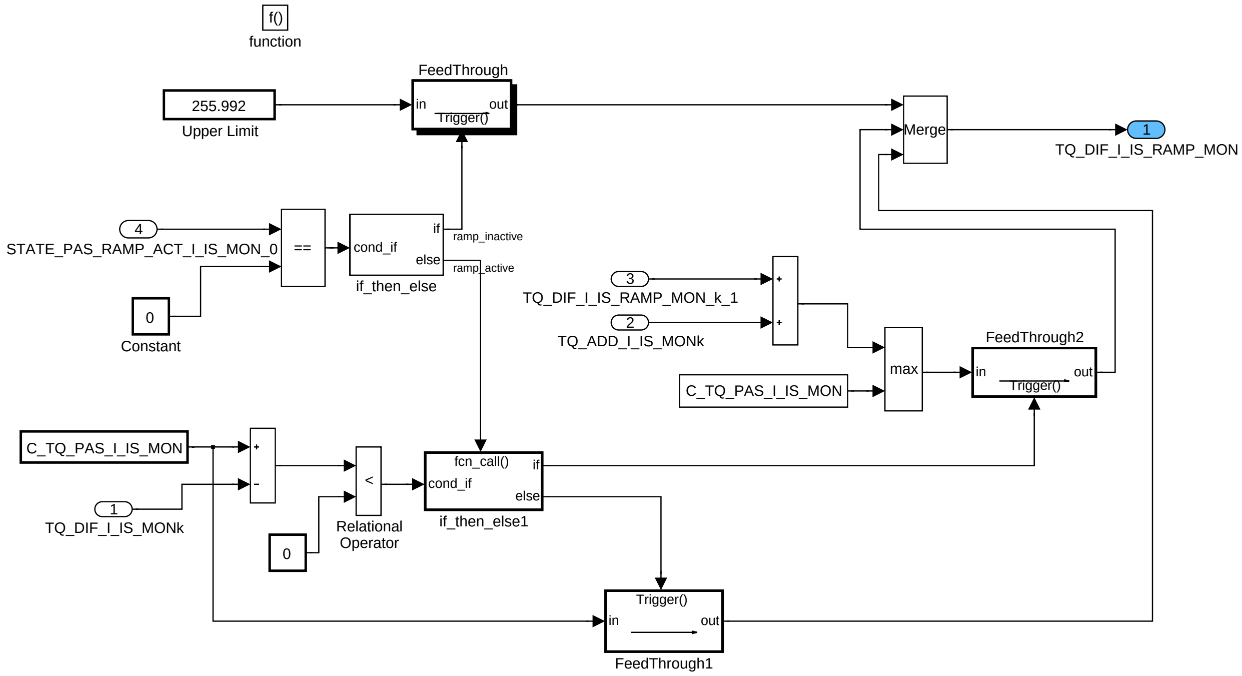
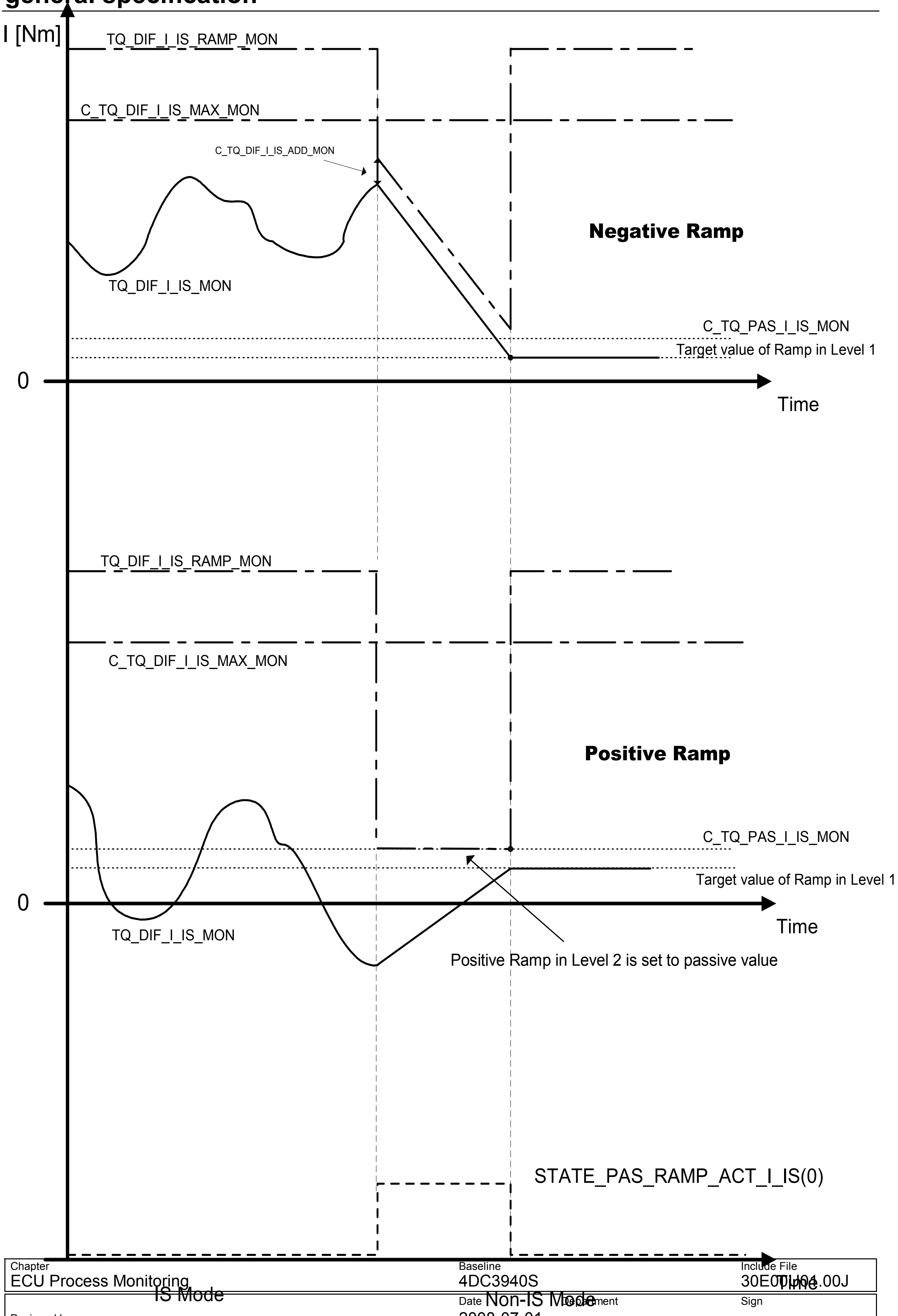


Figure 18 ECM2_DTSYSISC/ Monitoring_of_I_part/ Ramp_calculation/ ELSE_path


I [Nm]
TQ_DIF_I_IS_RAMP_MON
C_TQ_DIF_I_IS_MAX_MON
C_TQ_DIF_I_IS_ADD_MON
Negative Ramp
TQ_DIF_I_IS_MON
C_TQ_PAS_I_IS_MON
Target value of Ramp in Level 1
0
Time
TQ_DIF_I_IS_RAMP_MON
C_TQ_DIF_I_IS_MAX_MON
Positive Ramp
C_TQ_PAS_I_IS_MON
Target value of Ramp in Level 1
0
Time
TQ_DIF_I_IS_MON
Positive Ramp in Level 2 is set to passive value
STATE_PAS_RAMP_ACT_I_IS(0)
IS Mode
Non-IS Mode
Time

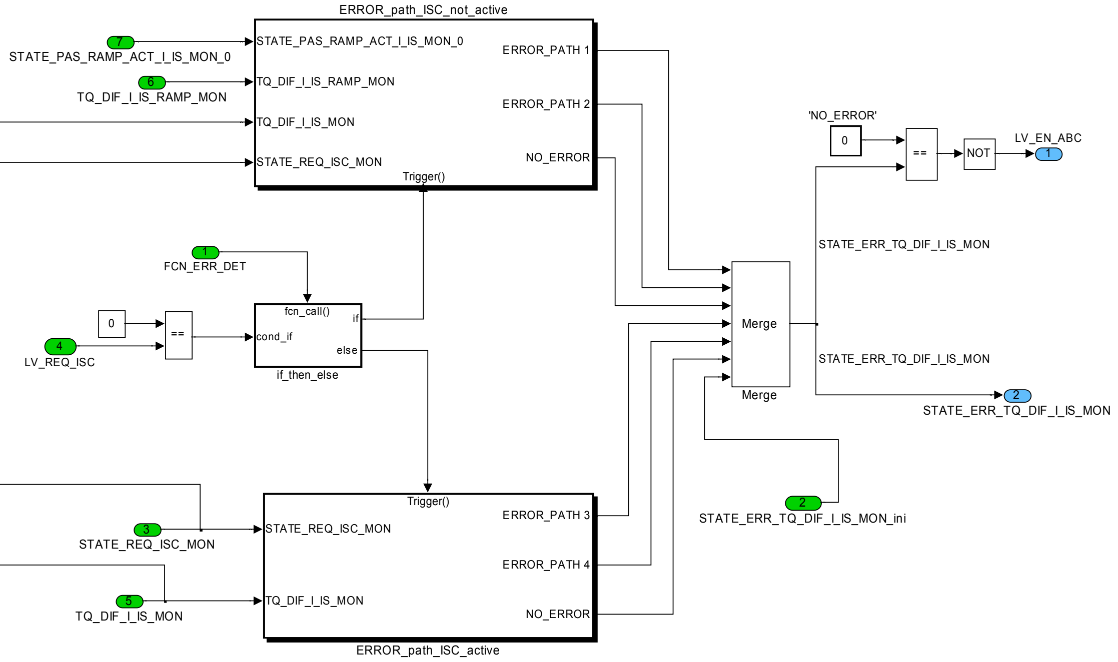


Figure 19 ECM2_DTSYSISC/ Monitoring_of_I_part/ ERROR_detection

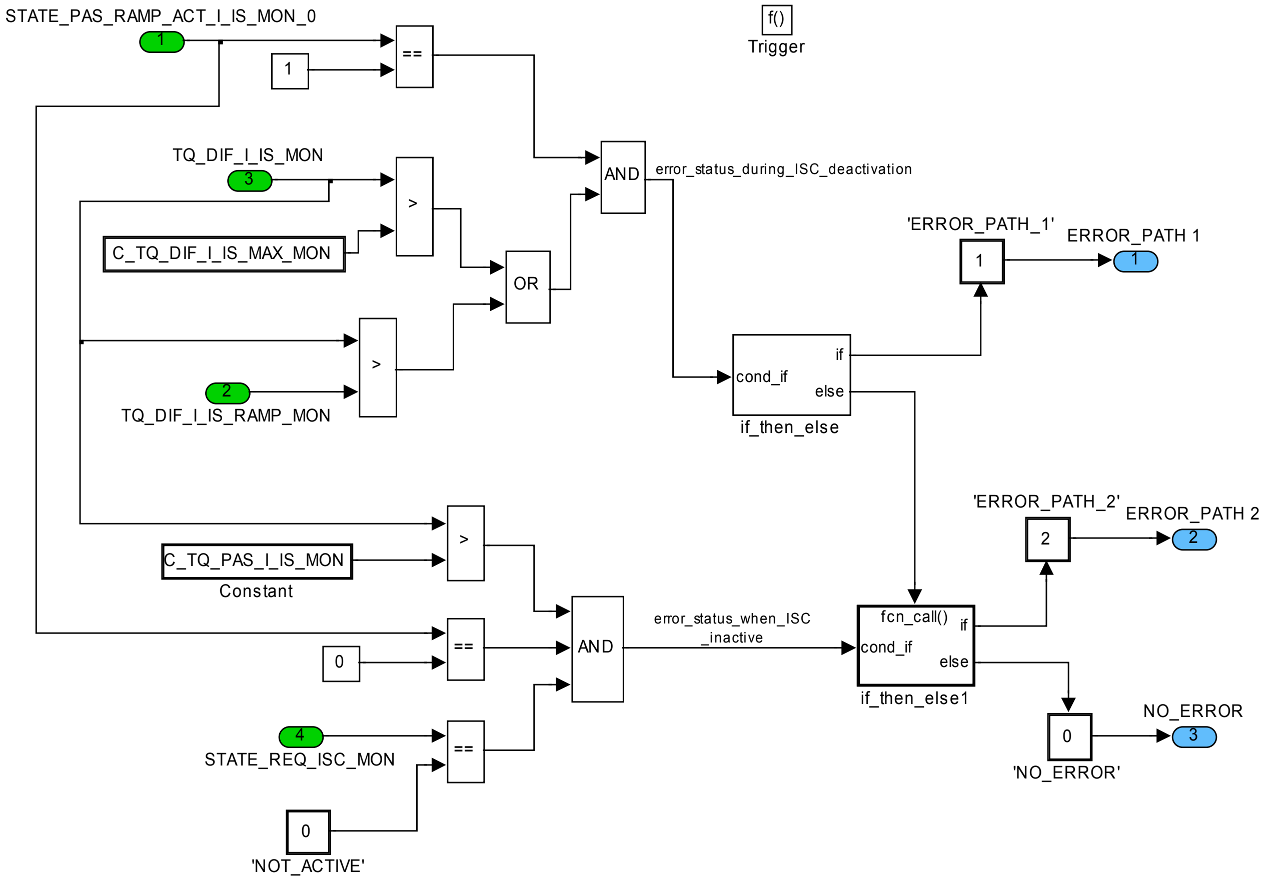


Figure 20 ECM2_DTSYSISC/ Monitoring_of_I_part/ ERROR_detection/ ERROR_path_ISC_not_active

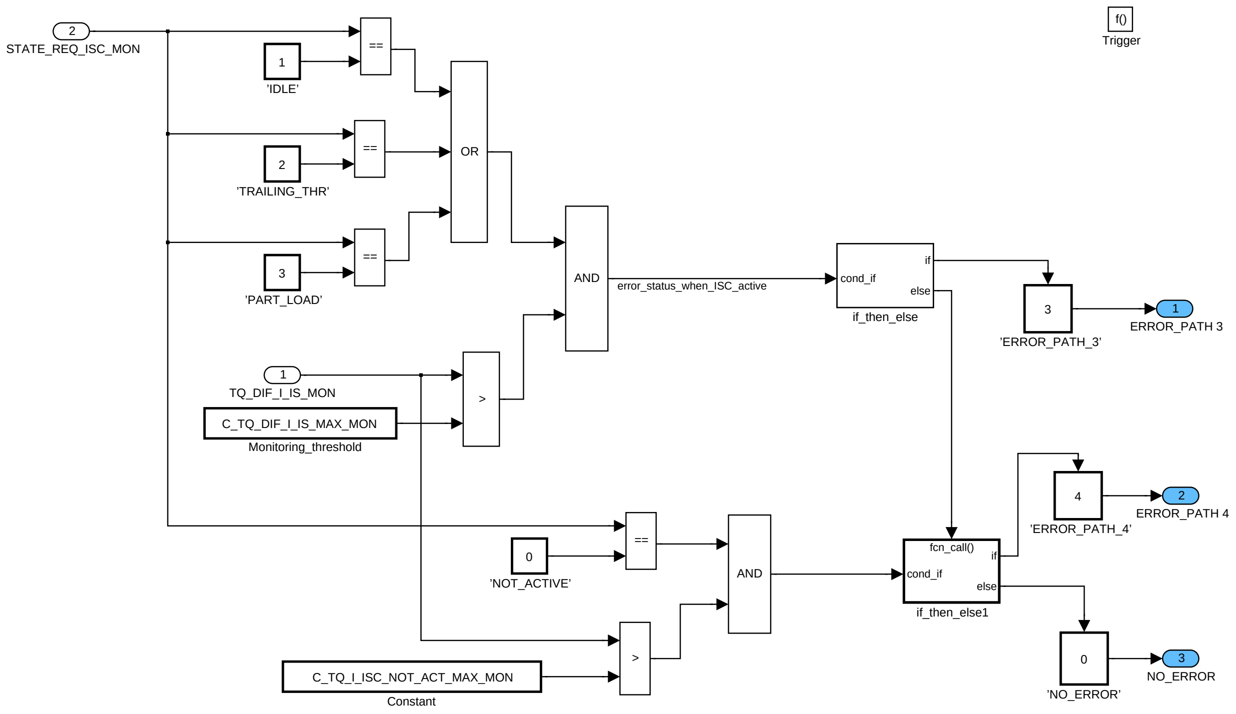


Figure 21 ECM2_DTSYSISC/ Monitoring_of_I_part/ ERROR_detection/ ERROR_path_ISC_active

11.10 Monitoring of torque losses

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
ABC_TQ_LOSS_MON	O/V	0... FFH	0... 255	1	[-]
Anti bounce counter for monitoring of torque losses					
LV_ERR_TQ_LOSS_MON	O/V	0... 1H	0... 1	1	[-]
Fault currently present in torque generation, symptom "torque losses"					
TQ_LOSS_MDL_MON	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Modelled torque losses					
TQ_LOSS_MON	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque losses					
TQ_LOSS_REQ_CLU_MDL_MON	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Modelled filtered torque losses					
TQ_LOSS_REQ_CLU_MON	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque losses without pumping losses					

Input Data:

LV_TQI_MON_ACT_MON	MAF_MON	N_32_MON	TCO_MON
TQ_DIF_IS_AD	TQ_LOSS	TQ_LOSS_ADD_MON	TQ_LOSS_ADD_FIL_MON
TQ_LOSS_REQ_CLU			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ABC_INC_TQ_LOSS_MON	1	0... FFH	0... 255	1	[-]
anti bounce counter increment					
C_ABC_MAX_TQ_LOSS_MON	1	1... FFH	1... 255	1	[-]
maximum of anti bounce counter					
C_MAF_STK_CYL_REF_MON	1	0... FFH	0... 1389	5.4470588	[mg/stk]
Constant for the determination of the monitored engine friction losses					
C_TQ_IS_AD_MIN_MON	1	80... 7FH	-256 ...254	2	[Nm]
Min. limitation of engine torque losses adaptation					
IP_TQFR_ADD_MON	5	0... FFH	0... 510	2	[Nm]
LDP_TCO_MON_IP_TQFR_ADD_MON	5	0... FEH	-48... 142.5	0.75	[°C]
Coolant temperature correction of friction torque					
IP_TQFR_MON	5*5	0... FFH	0... 510	2	[Nm]
LDP_N_32_MON_IP_TQFR_MON	5	0... FFH	0... 8160	32	[rpm]
LDP_MAF_MON_IP_TQFR_MON	5	0... FFH	0... 1389	5.4470588	[mg/stk]
Base friction torque and pumping losses					

Import Actions:

ACTION_ECM3_ChkCpl(**IN** <>, **IN** <>)
ACTION_ECM3_McChkStack()
ACTION_ECM3_ReadChkCpl(**IN** <>, **IN** <>, **OUT** <>)
ACTION_ECM3_Service10TaskPfm(**IN** <>)
ACTION_ECM3_Service11TaskPfm(**IN** <>)

General Information

Import actions:

ACTION_ECM3_Service9TaskPfm(IN< >)

ACTION_ECM3_Service10TaskPfm(IN< >)

ACTION_ECM3_Service11TaskPfm(IN< >)

Note: This actions are defined in chapter "Processor Monitoring", subsection "Program Flow Monitoring services for Process Monitoring". The first action has to be executed as first instruction in this module, the second action has to be executed as last instruction of this module. The argument n has the value 2.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service9TaskPfm() and ACTION_ECM3_Service10TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT < >, OUT < >, IN < >)

ACTION_ECM3_ChkCpl(IN < >, IN < >)

ACTION_ECM3_ReadChkCplOUT < >, IN < >, IN < >)

Note: This actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode O and history variables in modules of the Process Monitoring have to be checked by the cyclical RAM test.

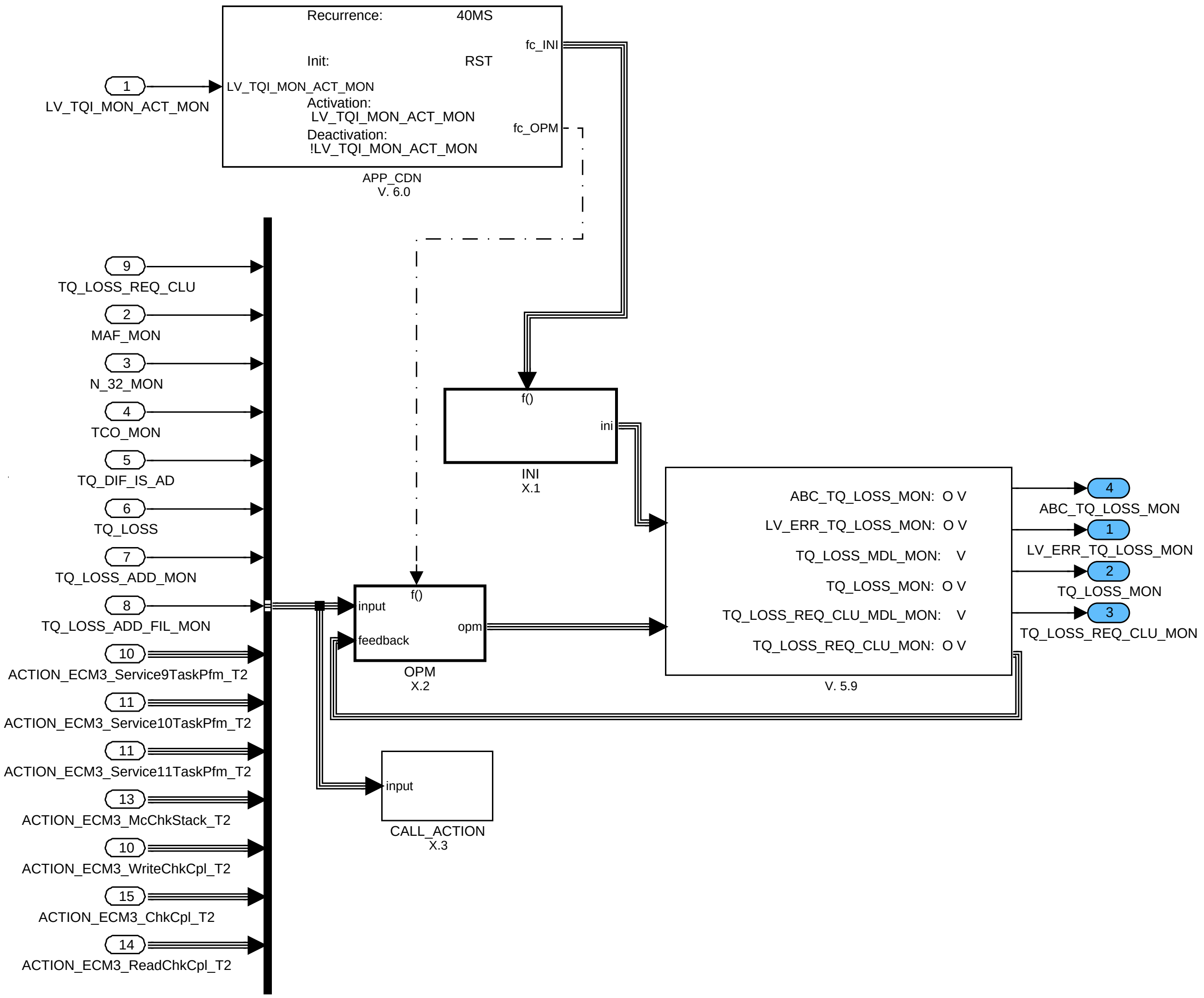
General information:
The objective of this function is the monitoring of the torque losses. It observes both level 1 values, TQ_LOSS and TQ_LOSS_REQ_CLU (filtered TQ_LOSS without pumping losses), that is exclusively used for the determination of TQ_MIN_CLU and TQ_MAX_CLU.
It has to be distinguished between engine immanent torque losses (e.g. TQFR) and not engine immanent torque losses (e.g. TQ_LOSS_ACC), that are determined in the module "Appl. Inc. and configuration of process monitoring".
A fault is detected after debouncing, if the monitored TQ_LOSS is smaller than the modelled one (TQ_LOSS_MDL_MON) or if the monitored TQ_LOSS_REQ_CLU is smaller than the modelled one (TQ_LOSS_REQ_CLU_MDL_MON).

Application Conditions

Initialization: RST
Recurrence: 40MS
Activation: LV_TQI_MON_ACT_MON

Deactivation: !LV_TQI_MON_ACT_MON

Function description



SDA_SRS / SDA V 5.2.7 26−Apr−2007

Figure 22:

11.10.1 Initialization

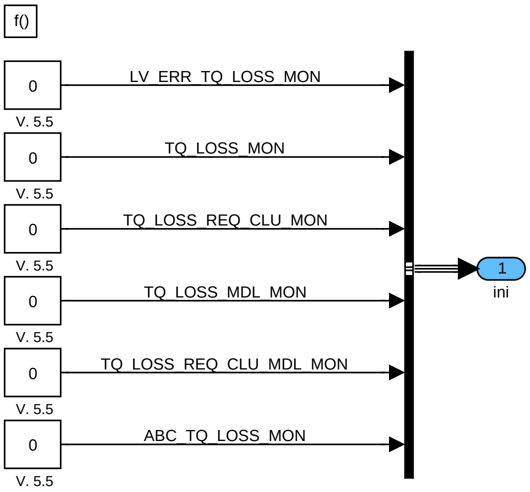


Figure 23:

11.10.2 Formula section

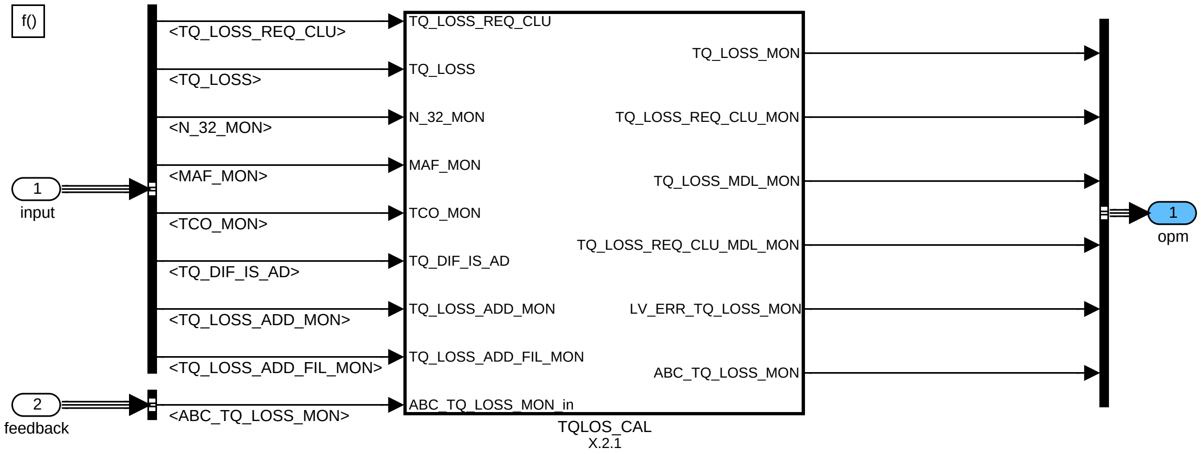


Figure 24:

11.10.2.1 Torque Loss calculation

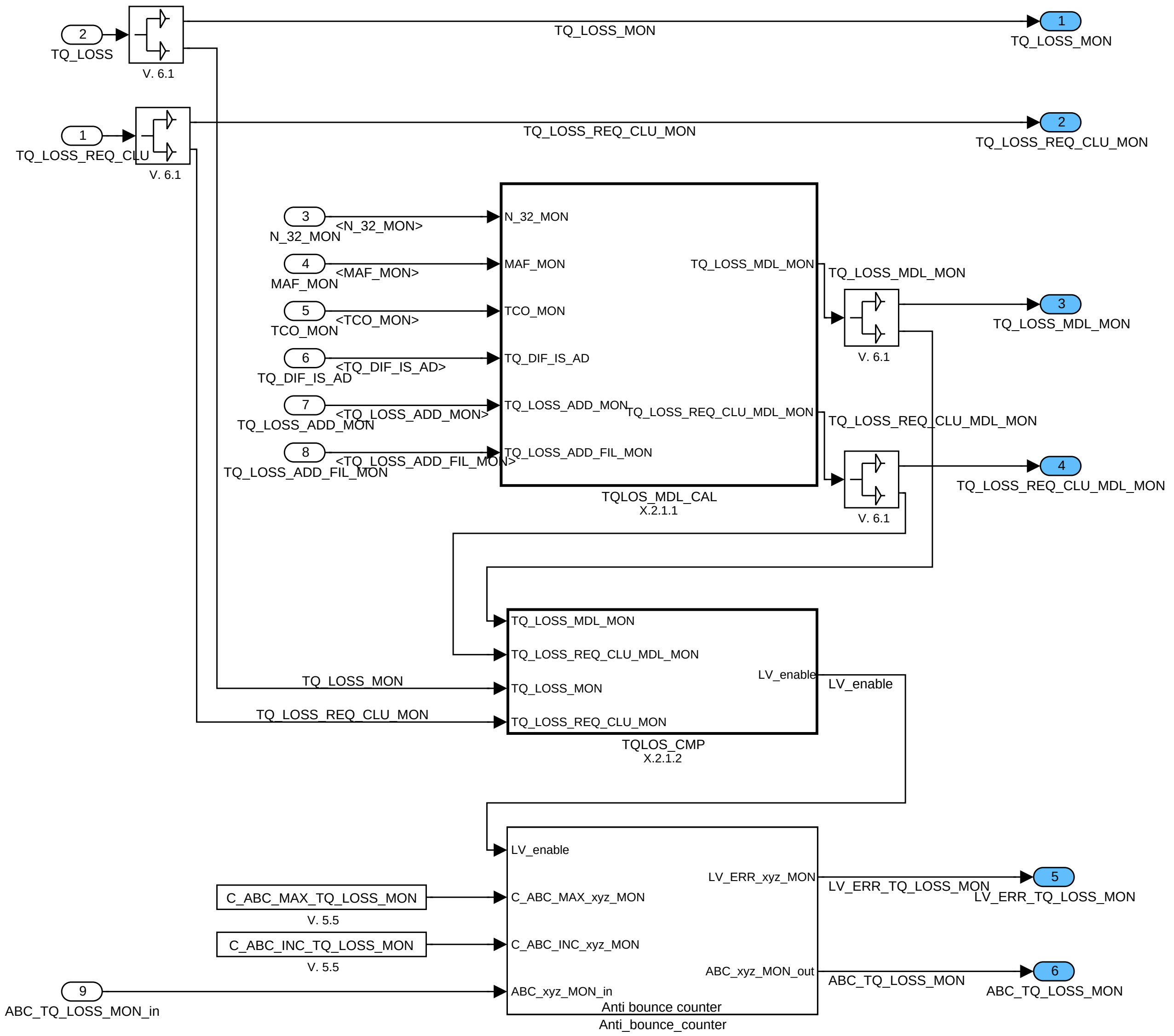


Figure 25:

11.10.2.1.1 Torque Loss - Modeled

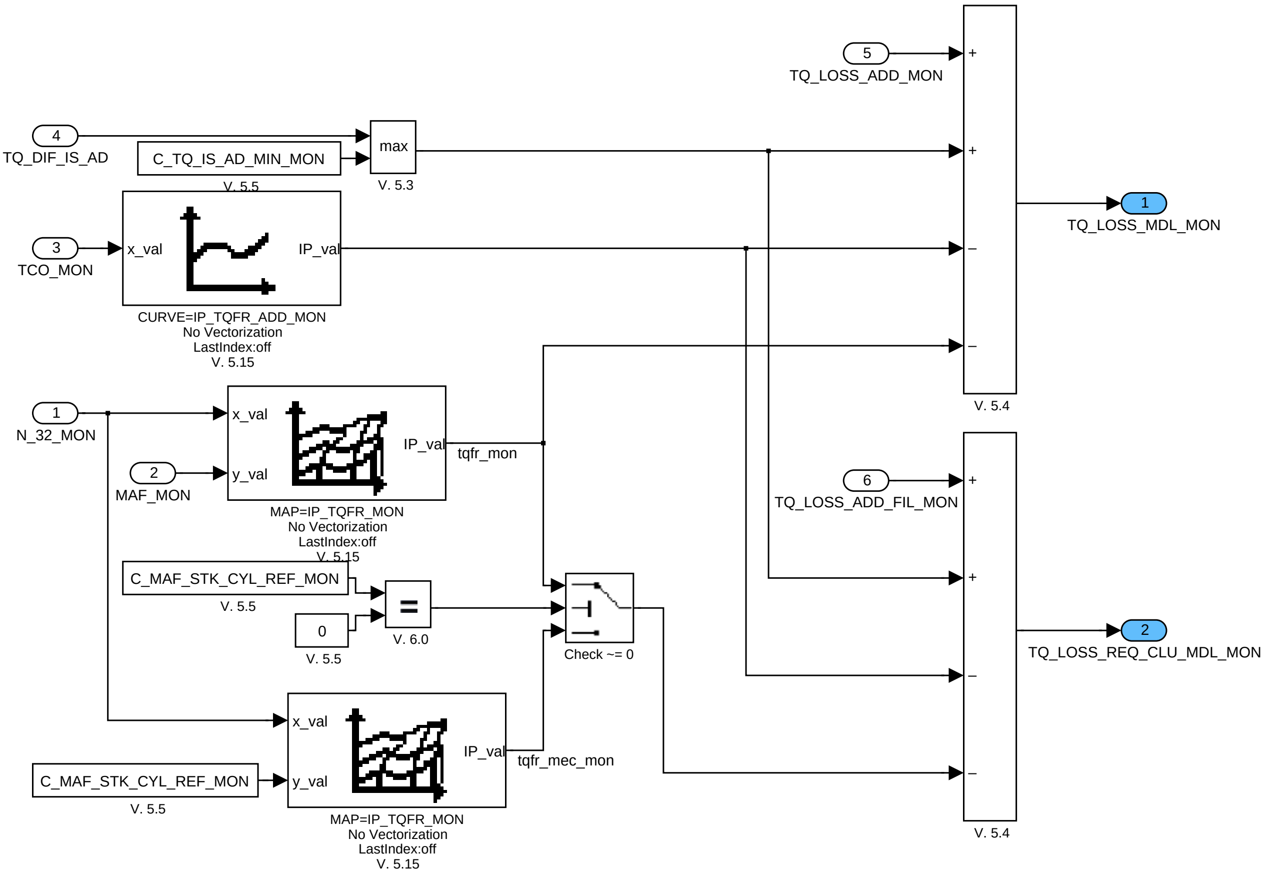


Figure 26:

11.10.2.1.2 Torque Loss detection flag

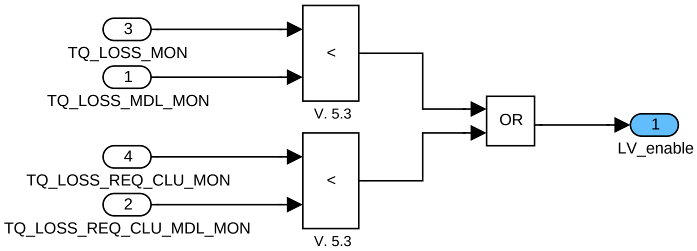


Figure 27:

11.10.3 Call Action

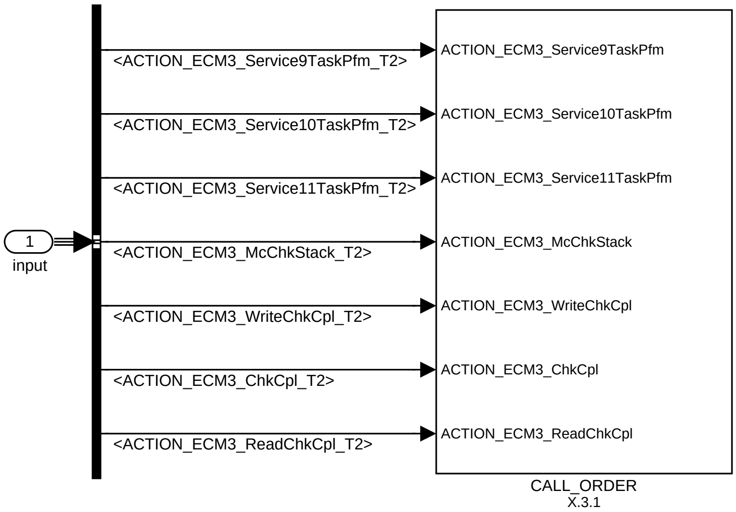


Figure 28:

11.10.3.1 Call order

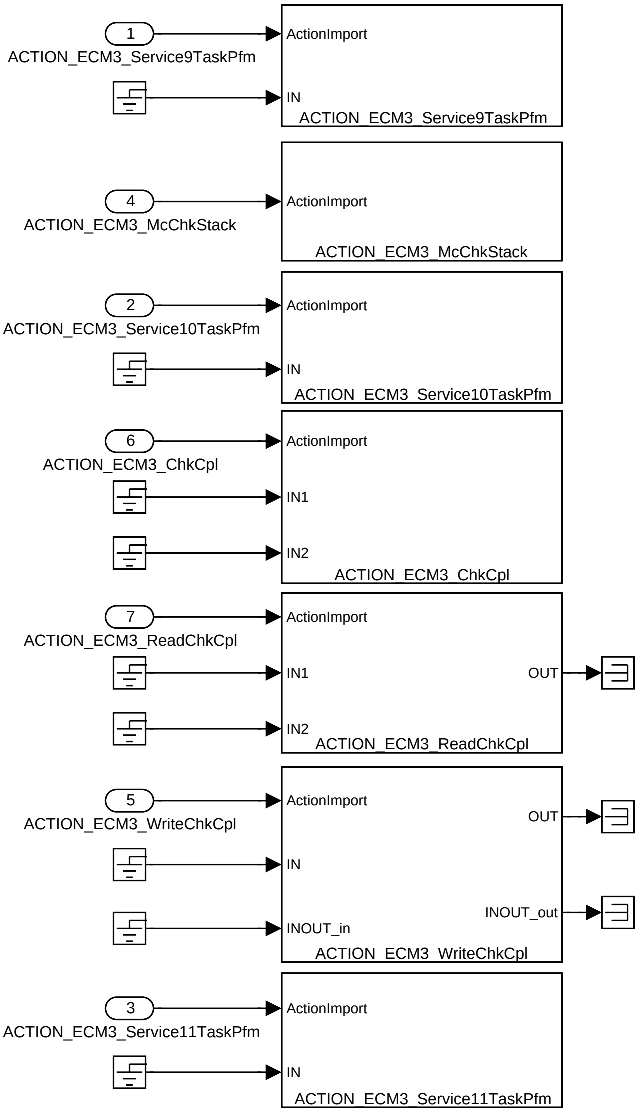


Figure 29:

11.11 Monitoring of minimum torque at clutch

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TQ_MIN_CLU_MON	V/O	0...1H	0...1	1	[-]
Fault currently present in torque generation, symptom "minimum torque at clutch"					
TQ_MIN_CLU_MON	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
monitored minimum torque at clutch					
ABC_TQ_MIN_CLU_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for monitoring of minimum torque at clutch					
TQ_ST_MON	V	0...7FFFH	0...1023.96875	0.03125	[Nm]
modelled value of start torque					
CTR_DLY_TQ_ST_MON	V	0...FFH	0...10.2	0.04	[s]
Delay time counter for holding start torque influence					
FAC_N_SP_IS_RATIO_MON	V	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Model of control factor for engine self-stabilising effect					
STATE_ERR_TQ_MIN_CLU_MON	V/O	0H 1H 2H	NO_ERROR ERROR_PATH1 ERROR_PATH2	1	[-]
State variable indicating origin of error resulting from monitoring of L1 torque value					
TQ_MIN_CLU_DIF_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
difference of monitored and modelled minimum torque at clutch					
TQ_MIN_CLU_MDL_2_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Temporary variable 2 for calculation of modelled minimum torque at clutch					
TQ_MIN_CLU_MDL_5_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Temporary variable 4 for calculation of modelled minimum torque at clutch					

Input data:

N_32_MON	TCO_MON	N_SP_IS_RATIO_MON	TQ_MIN_CLU
TQ_CONV_MAX_MON	TQ_DIF_I_IS_MON	TQ_ADD_MIN_CLU_MON	TQ_LOSS_MON
TQ_LOSS_REQ_CLU_MON	STATE_ERR_DET_TQ_MIN_MON	LV_SWI_TQ_MIN_CLU_MON	LV_TQI_MON_ACT_MON
LV_TQ_MIN_CLU			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TQ_MIN_CLU_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_TQ_MIN_CLU_MON	1	1...FFH	1...255	1	[-]
Maximum of anti bounce counter					
C_N_ST_MAX_MON	1	0...FFH	0...8160	32	[rpm]
maximum start end speed					
C_DLY_TQ_ST_MON	1	0...FFH	0...10.2	0.04	[s]
Time delay to hold influence of start torque on TQ_MIN_CLU model value					
C_TQ_MIN_CLU_MDL_OFS_MON	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Offset for L2 model value of minimum torque at clutch					
IP_FAC_N_SP_IS_RATIO_MON	5	0...FFFFH	-8...7.99975	0.2441e-3	[-]
LDP_N_SP_IS_RATIO_MON_IP_MON	5	0...FFFFH	0...7.99987	0.1221e-3	[-]
Idle self stabilising factor					
IP_TQI_MIN_REQ_PU_MON	6*4	0...7FFFH	0...1023.96875	0.03125	[Nm]
LDP_N_32_MON_IP_TQI_MIN_MON	6	0...FFH	0...8160	32	[rpm]
LDP_TCO_MON_IP_TQI_MIN_PU_MON	4	0...FEH	-48...142.5	0.75	[°C]
Minimum requested indicated torque in PU phase					

11.11.1 FUNCTION PART: ECM2_DTSYSTQMIN

Import actions:

ACTION_ECM3_Service12TaskPfm(IN <>)

ACTION_ECM3_Service13TaskPfm(IN <>)

ACTION_ECM3_Service14TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The argument has the value 3.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service12TaskPfm() and ACTION_ECM3_Service13TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT<>, OUT<>, IN <>)

ACTION_ECM3_ChkCpl(IN<>, IN<>)

ACTION_ECM3_ReadChkCpl(OUT <>, IN<>, IN<>)

Note: These actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

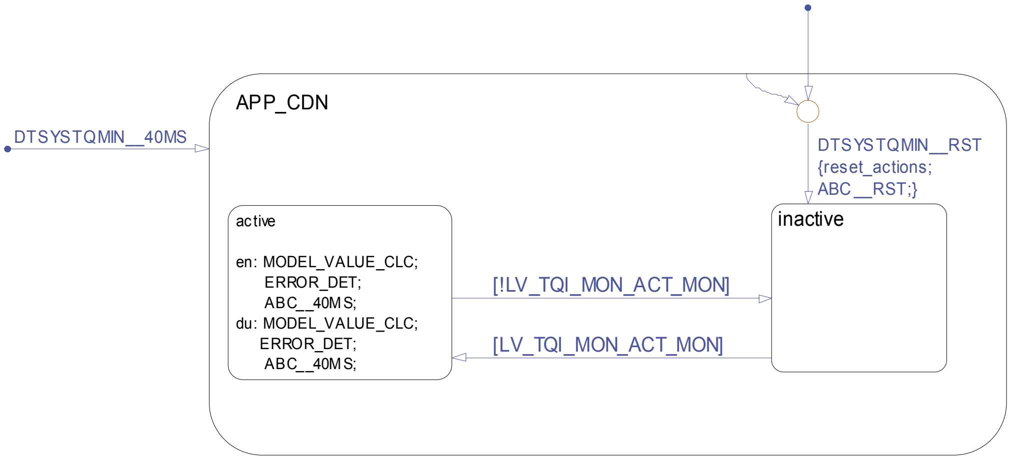
General information:

The objective of this function is the monitoring of the minimum torque at clutch. The model takes into account that a great part of the minimum torque at clutch could be the integral part of the ISC. Therefore the difference of monitored minimum torque at clutch and I-part of ISC is compared to a torque *threshold* which is dependent on:**1)** the engine self stabilising factor (function of ratio between engine speed and idle speed setpoint) **2)** TQ_LOSS_REQ_CLU_MON and **3)** TQ_CONV_MAX_MON. For engine speeds above idle speed setpoint this difference must be negative. For speeds below idle speed setpoint great values for minimum torque at clutch are allowed.

One exception of the above mentioned rule is at engine start. If the engine is in start phase, an constant additive term is taken into account for the threshold (see also "Chapter D – Minimum Torque at Clutch"). To monitor this start torque term, the actual engine speed is compared to an upper threshold for the engine start end-speed. In addition to the I-part of ISC, the two other factors influencing the minimum torque at clutch are hydraulic torque losses inside the converter of an automatic gearbox - TQ_CONV_MAX_MON (which depends on coolant temperature and idle speed setpoint) and the minimum requested indicated torque in the PU phase - TQI_MIN_REQ_PU_MON. However, if the engine is in PU or PUC mode TQ_MIN_ CLU_MDL is set to minimum possible torque that the driver can request i.e. TQ_LOSS_MON.

The plausiblisation of the L1 value, TQ_MIN_CLU, against a L2 model value is done based on current engine state. The variable STATE_ERR_DET_TQ_MIN_MON (calculated in the module ‘Application incidences of process monitoring’) is used to switch between the different error-detection paths. If the engine is in PU or PUC and the L1 minimum torque at clutch value is greater than TQ_LOSS_MON, then an error state is set. If however the engine is neither in PU or PUC, then the L1 value is compared to the L2 model value TQ_MIN_CLU_MDL_4_MON. If any of the error states are set, a fault is detected after debouncing.

Application Conditions



Initialisation(DTSYSTQMIN__RST): for condition see ‘Application Incidences of Process Monitoring’

Function Description

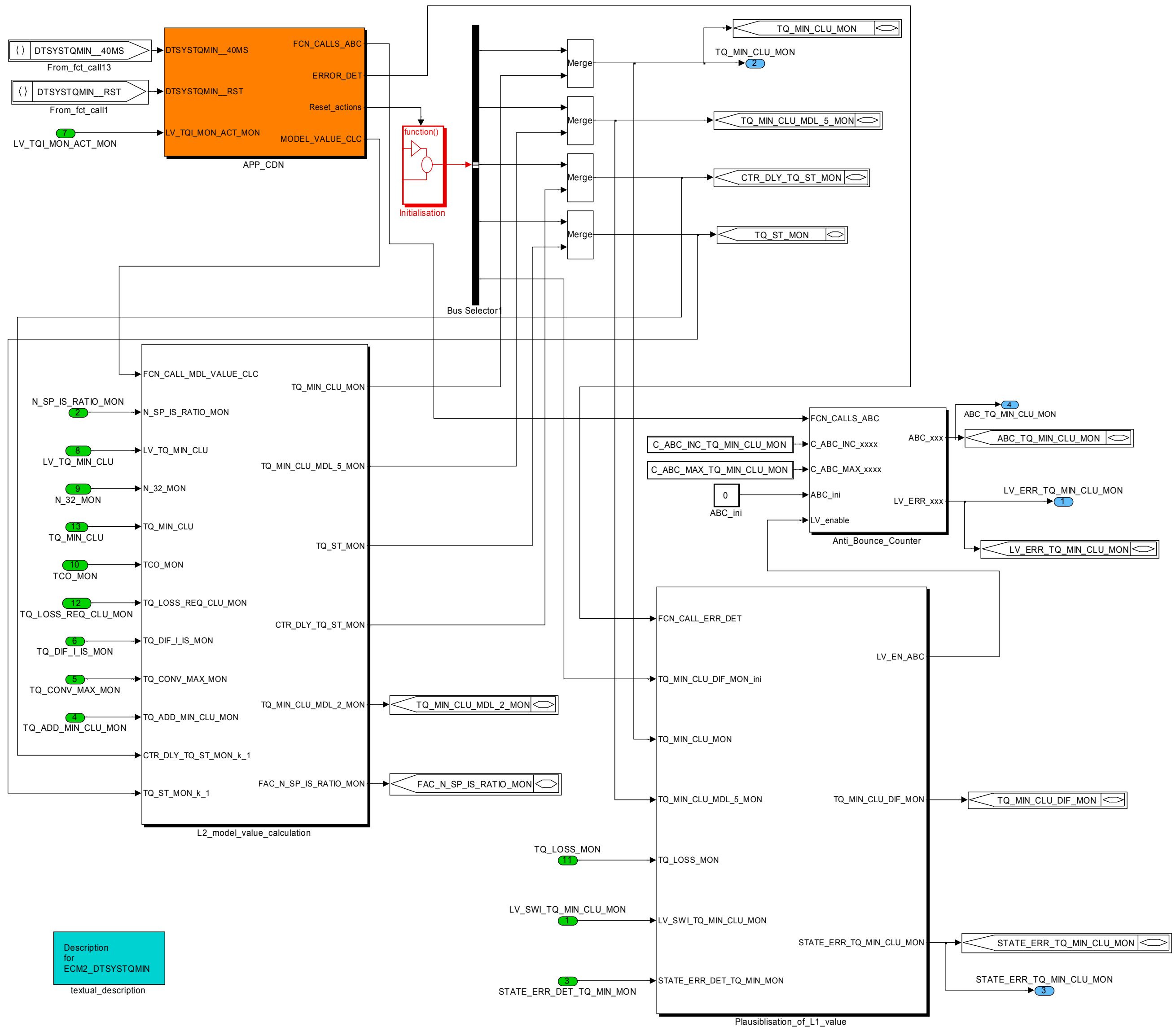


Figure 30 ECM2_DTSYSTQMIN

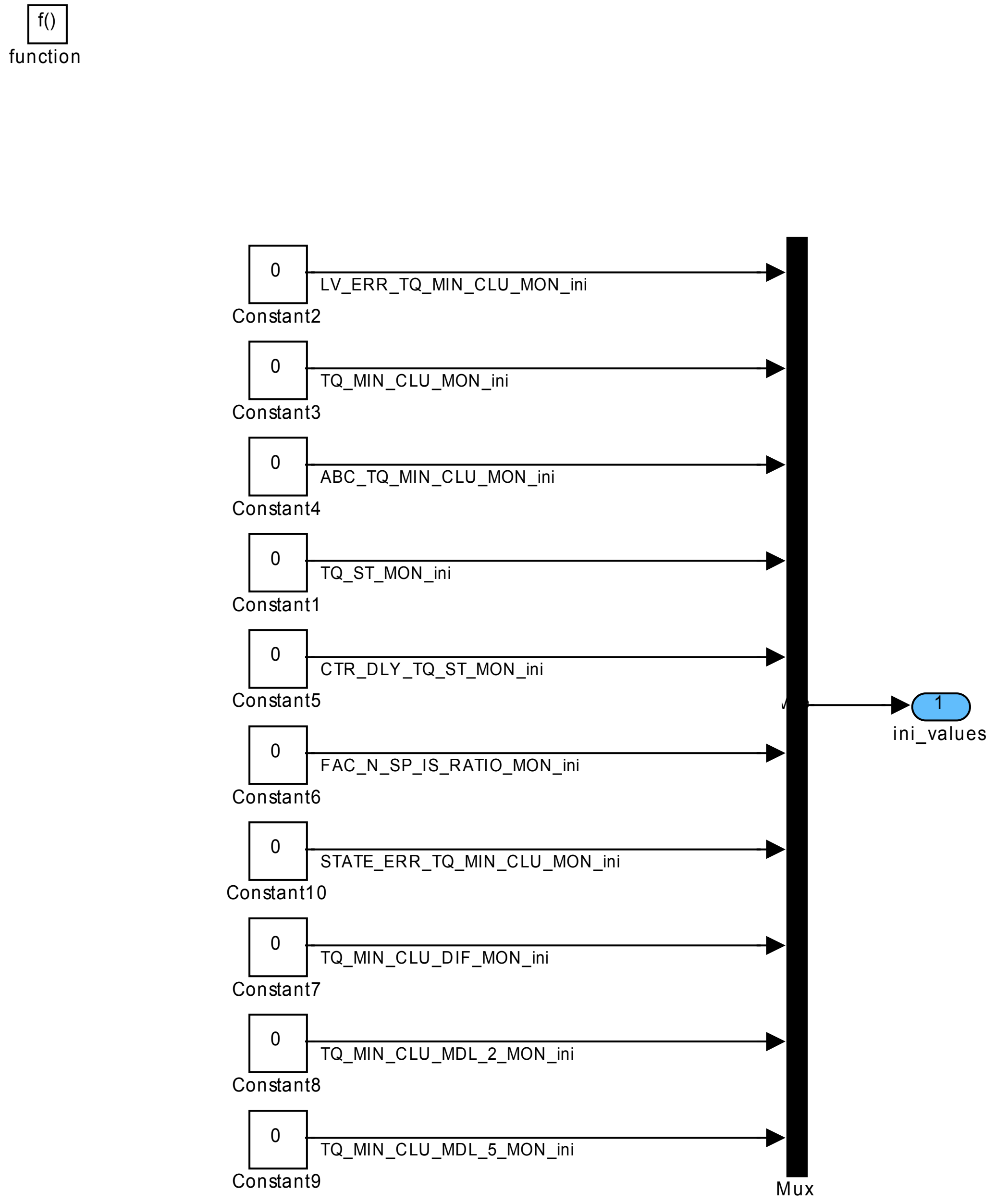


Figure 31 ECM2_DTSYSTQMIN/ Initialisation

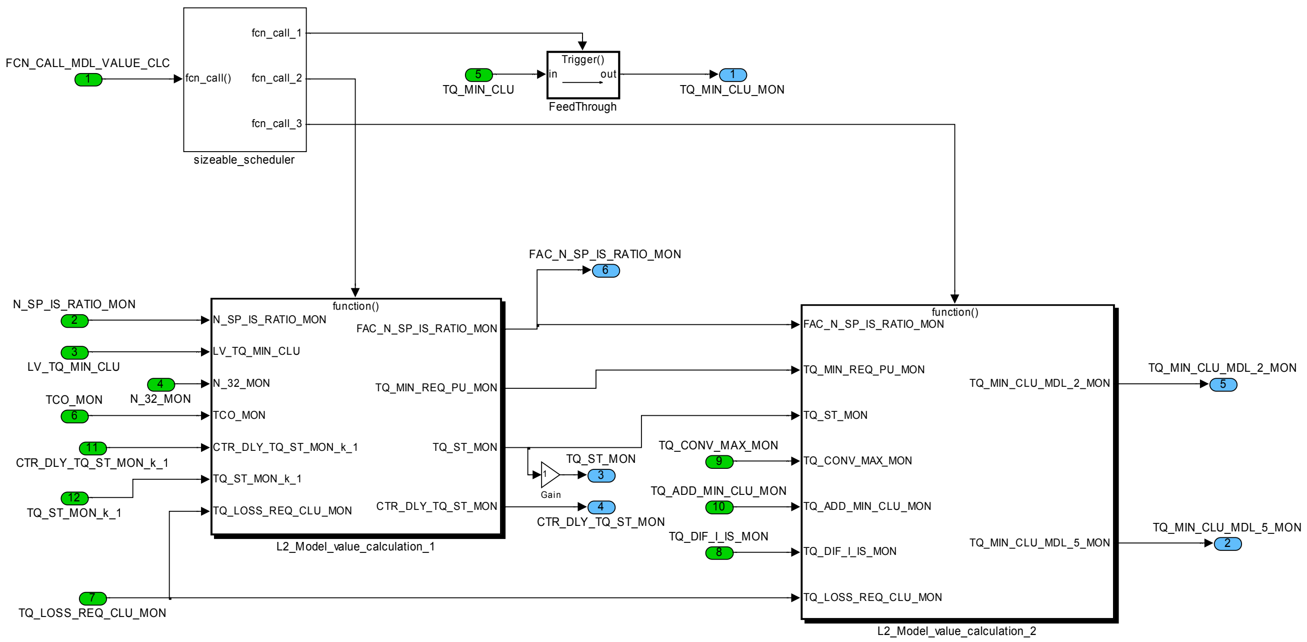


Figure 32 ECM2_DTSYSTQMIN/ L2_model_value_calculation

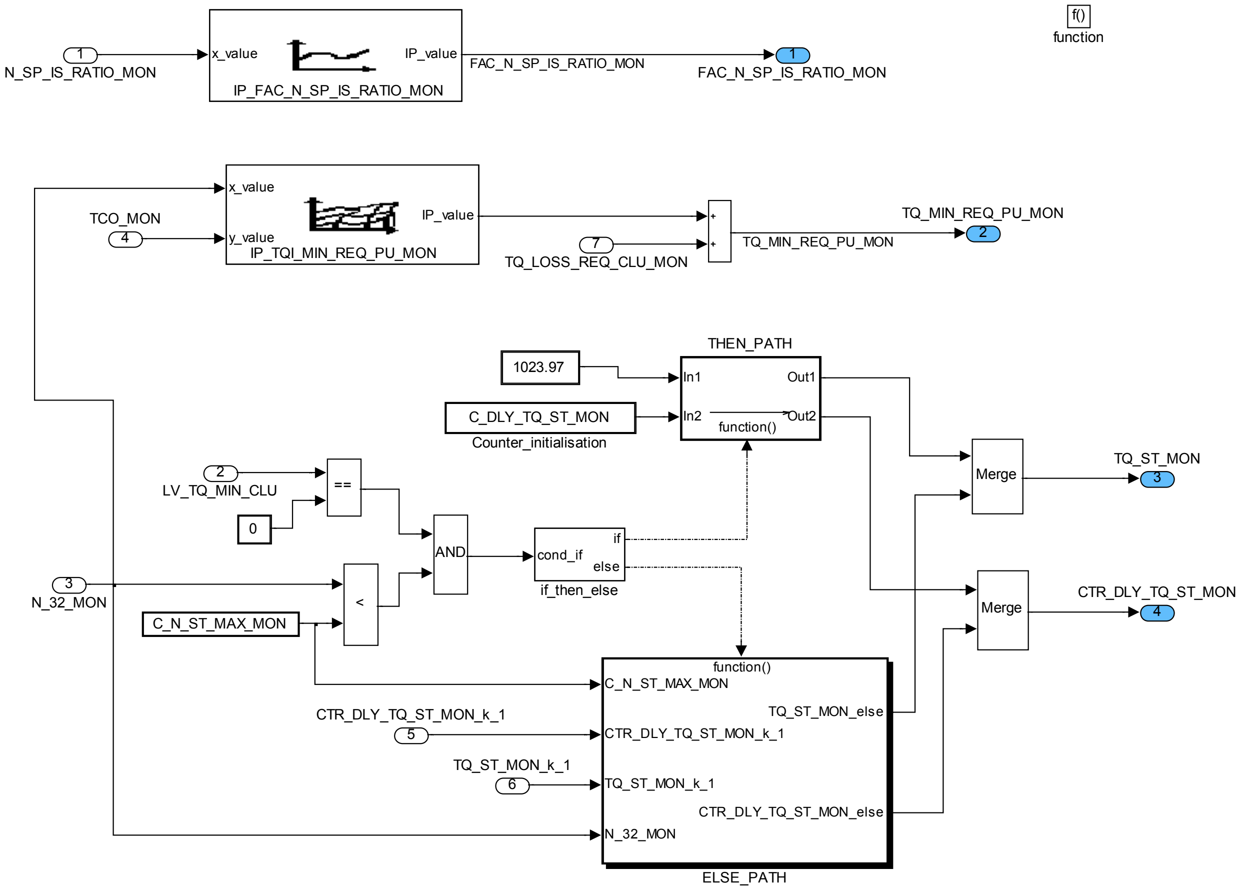


Figure 33 ECM2_DTSYSTQMIN/ L2_Model_value_calculation / L2_Model_value_calculation_1

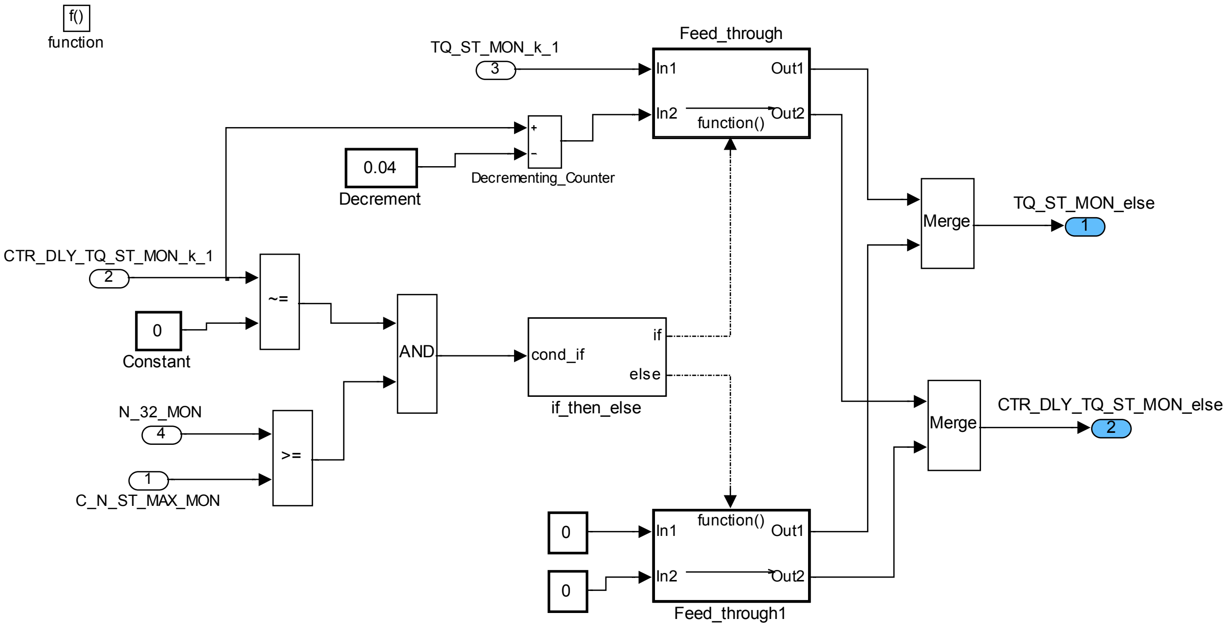


Figure 34 ECM2_DTSYSTQMIN/ L2_Model_value_calculation / L2_Model_value_calculation_1/ ELSE_path

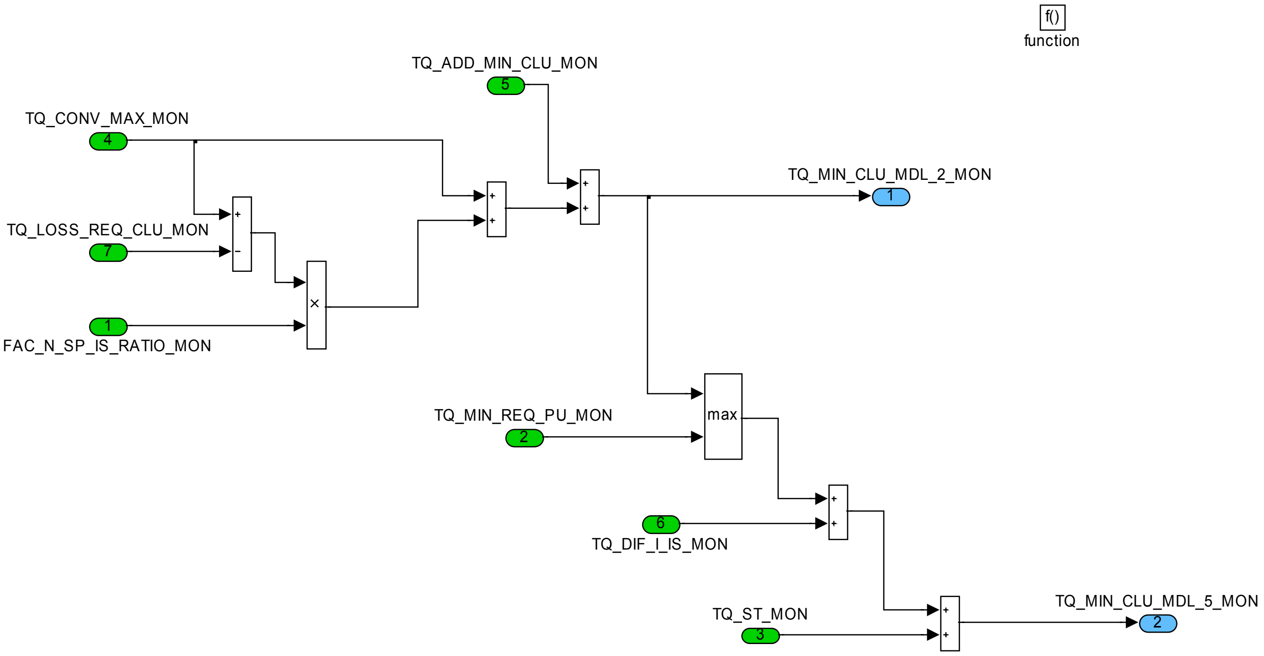


Figure 35 ECM2_DTSYSTQMIN/ L2_model_value_calculation/ L2_Model_value_calculation_2

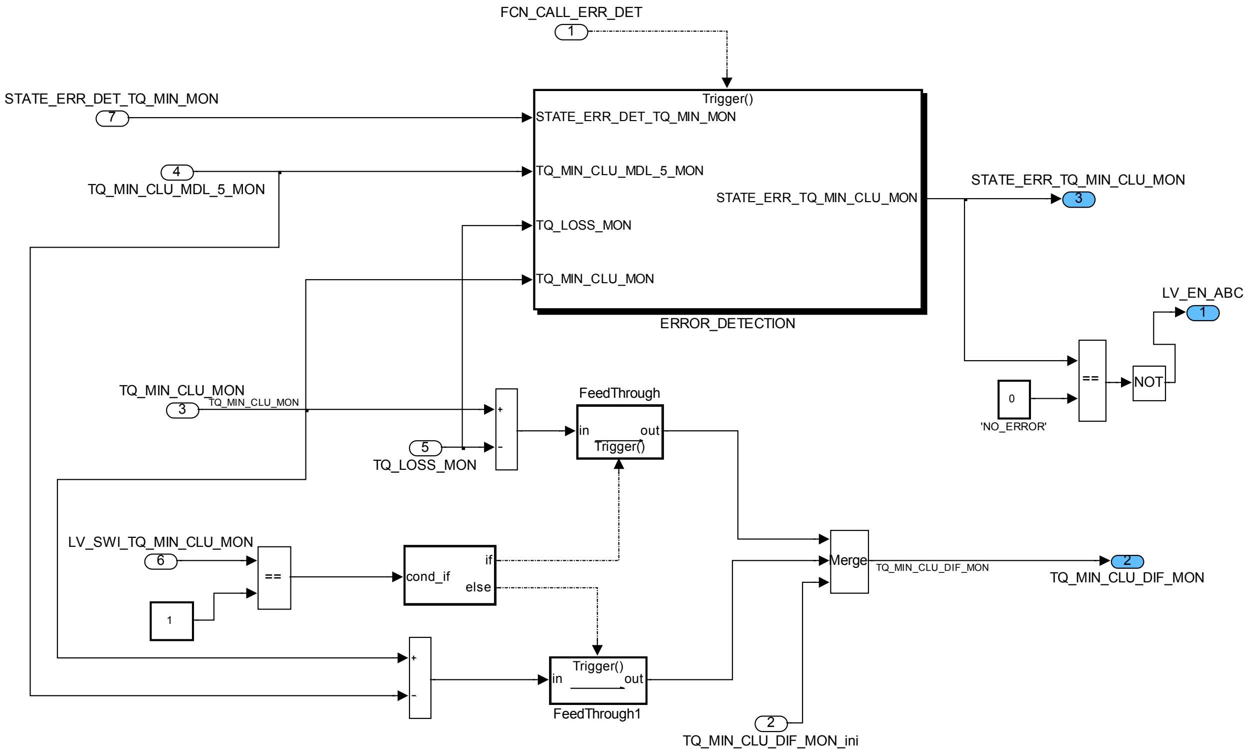


Figure 36 ECM2_DTSYSTQMIN/ Plausibilisation_of_L1_value

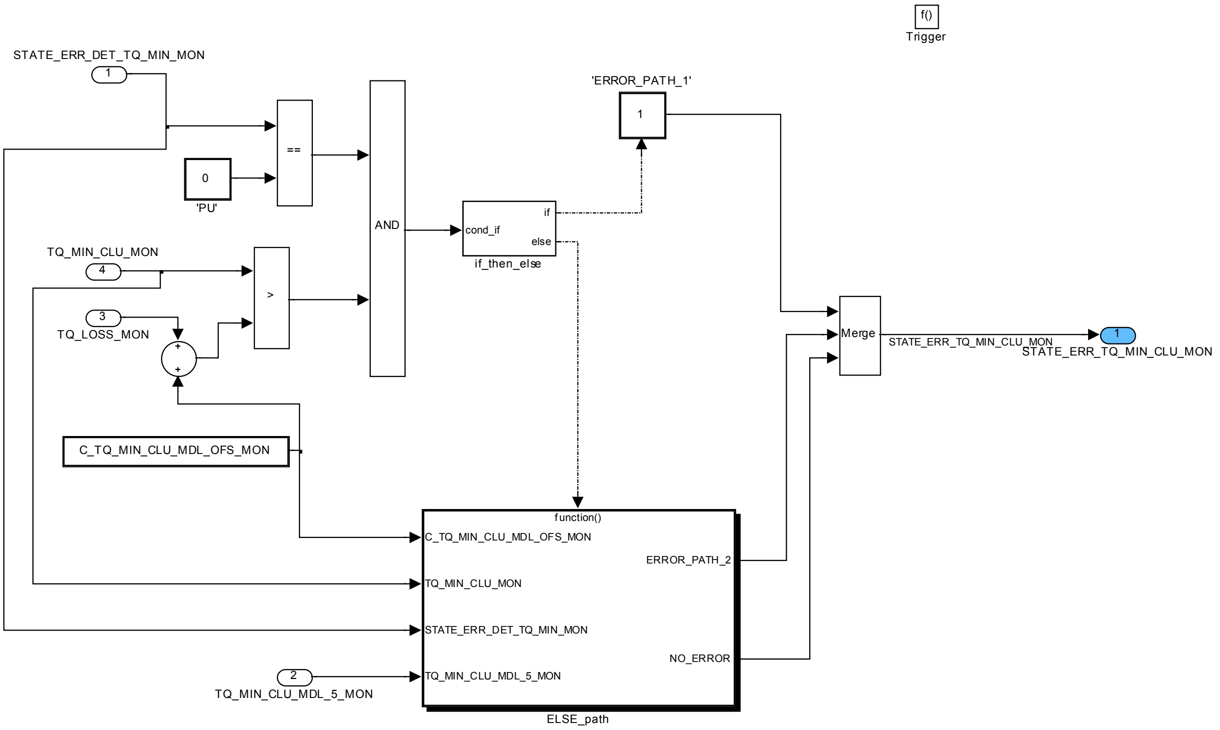


Figure 37 ECM2_DTSYSTQMIN/ Plausibilisation_of_L1_value / ERROR_DETECTION

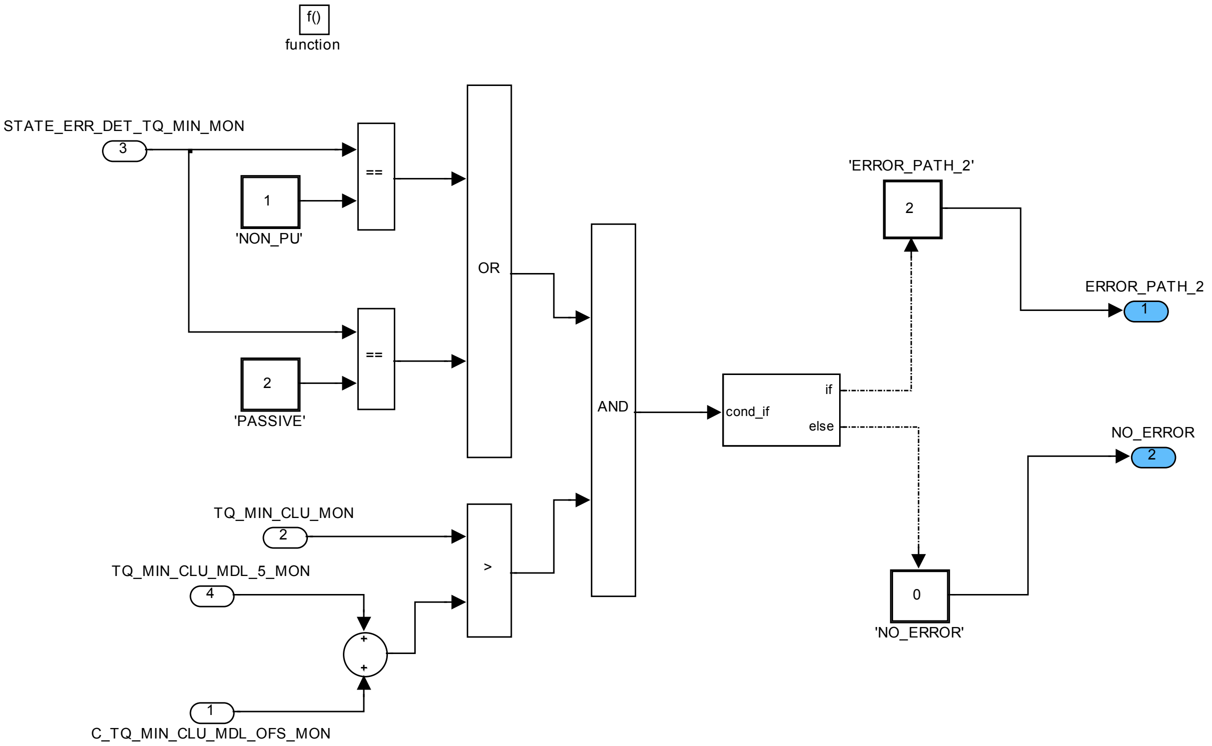


Figure 38 ECM2_DTSYSTQMIN/ Plausiblisation_of_L1_value/ ERROR_DETECTION/ ELSE_path

11.12 Monitoring of maximum torque at clutch

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TQ_MAX_CLU_MON	V/O	0...1H	0...1	1	[-]
Fault currently present in torque generation, symptom "maximum torque at clutch"					
TQ_MAX_CLU_MON	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
monitored maximum torque at clutch					
ABC_TQ_MAX_CLU_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for monitoring of maximum torque at clutch					
TQ_MAX_CLU_DIF_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Difference between monitored and modelled maximum torque at clutch					
TQ_MAX_CLU_MAX_MON	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Model value for maximum torque at clutch					
FAC_CHA_MAX_MON	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Ratio between the optimal and real air efficiency					
FAC_CUS_TQ_MAX_MON	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Model value of the ouput factor from the crossfade calculation					

Input data:

LV_TQI_MON_ACT_MON	N_32_MON	TQ_LOSS_REQ_CLU_MON	TQ_MAX_CLU
FAC_TQ_REQ_MON			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TQ_MAX_CLU_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_TQ_MAX_CLU_MON	1	1...FFH	1...255	1	[-]
Maximum of anti bounce counter					
C_OFS_TIA_MON	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Offset reflecting the sound velocity influence					
IP_TQI_REF_MAX_MON	7	0...7FFFH	0...1023.96875	0.03125	[Nm]
LDP_N_32_MON_IP_TQI_MAX_MON	7	0...FFH	0...8160	32	[rpm]
Maximum reference indicated engine torque					
IP_FAC_CHA_MAX_1_MON	11	0...FFFFH	-327.68...327.67	0.01	[%]
LDPM_N_32_MON__FAC_CHA_MON	11	0...FFH	0...8160	32	[rpm]
Maximum realtive cylinder filling by full load under normal conditions					
IP_FAC_CHA_MAX_2_MON	11	0...FFFFH	-327.68...327.67	0.01	[%]
LDPM_N_32_MON__FAC_CHA_MON	11	0...FFH	0...8160	32	[rpm]
Maximum realtive cylinder filling					
IP_FAC_CHA_MAX_FAC_TQ_REQ_MON	6	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_FAC_TQ_REQ_MON__TQ_MAX_MON	6	0...FFH	0...1.99218	0.0078125	[-]
Weighting factor for TQ_MAX_MON correction					

11.12.1 ECM2_DTSYSTQMAX

Import actions:

ACTION_ECM3_Service12TaskPfm(IN <>)
ACTION_ECM3_Service13TaskPfm(IN <>)
ACTION_ECM3_Service14TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The argument has the value 2.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service12TaskPfm() and ACTION_ECM3_Service13TaskPfm().

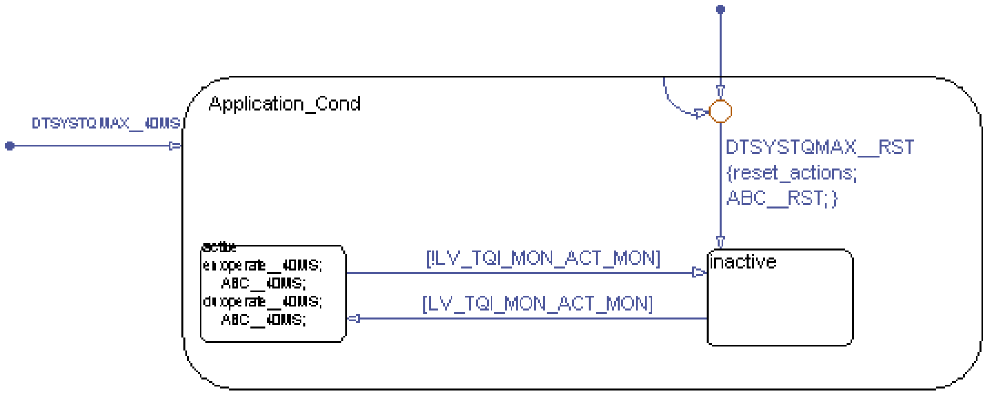
ACTION_ECM3_WriteChkCpl (INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN<>, IN<>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: These actions are defined in chapter "Processor Monitroing", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

General information:

The objective of this function is the monitoring of the maximum torque at clutch. The maximum torque at clutch is compared to a model threshold value (TQ_MAX_CLU_MAX_MON). The factors influencing this threshold are: IP_TQI_REF_MAX_MON (the worst case condition i.e. maximum possible fresh air entering in the cylinder is taken into account inside this map), TQ_LOSS_REQ_CLU_MON and driver request influence as a weighting factor in a crossfade calculation. A fault is detected after debouncing, if the maximum torque at clutch exceeds this threshold.

Application Condition



***Initialisation(DTSYSTQMAX__RST)*:** for condition see ‘Application Incidences of Process Monitoring’

Function Description

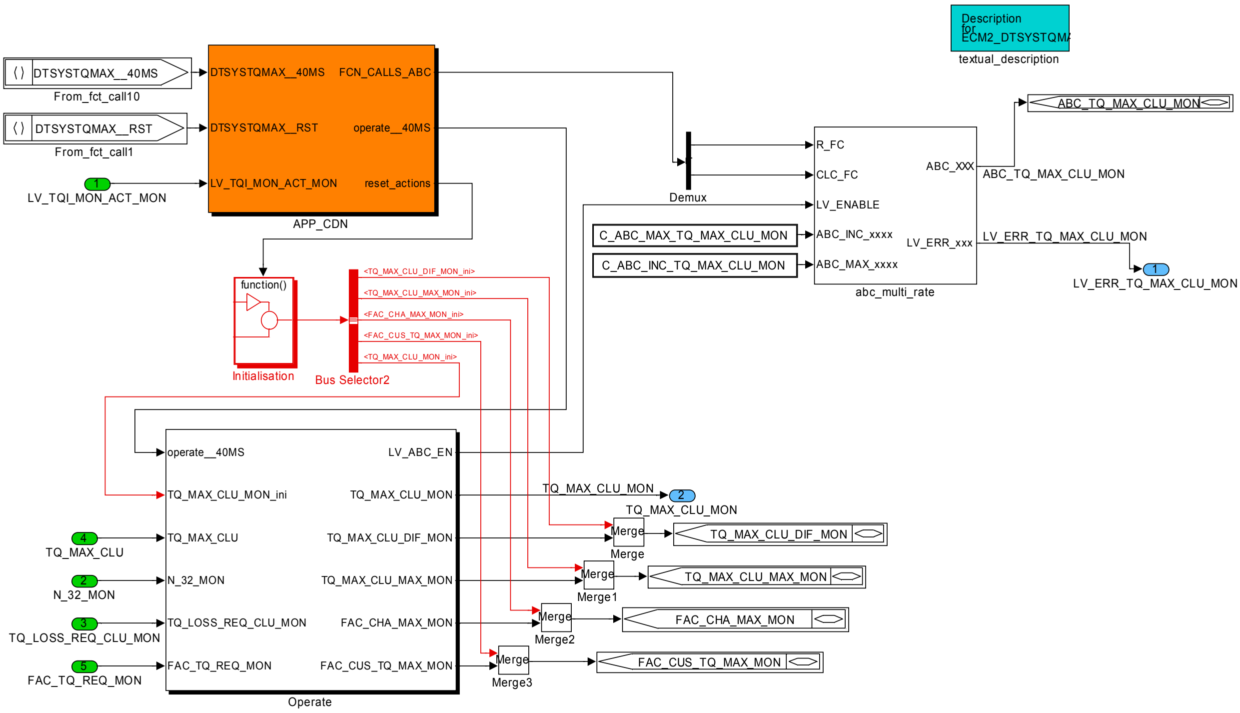


Figure 39 ECM2_DTSYSTQMAX

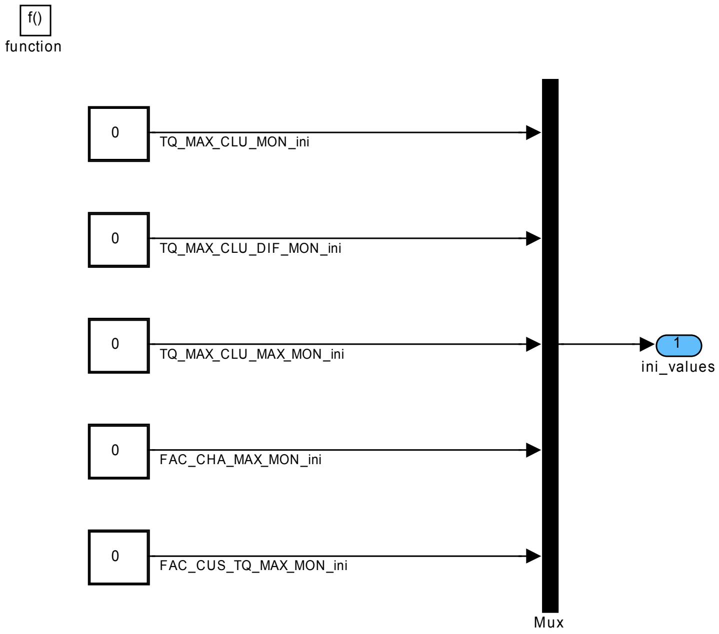


Figure 2 ECM2_DTSYSTQMAX/Initialisation

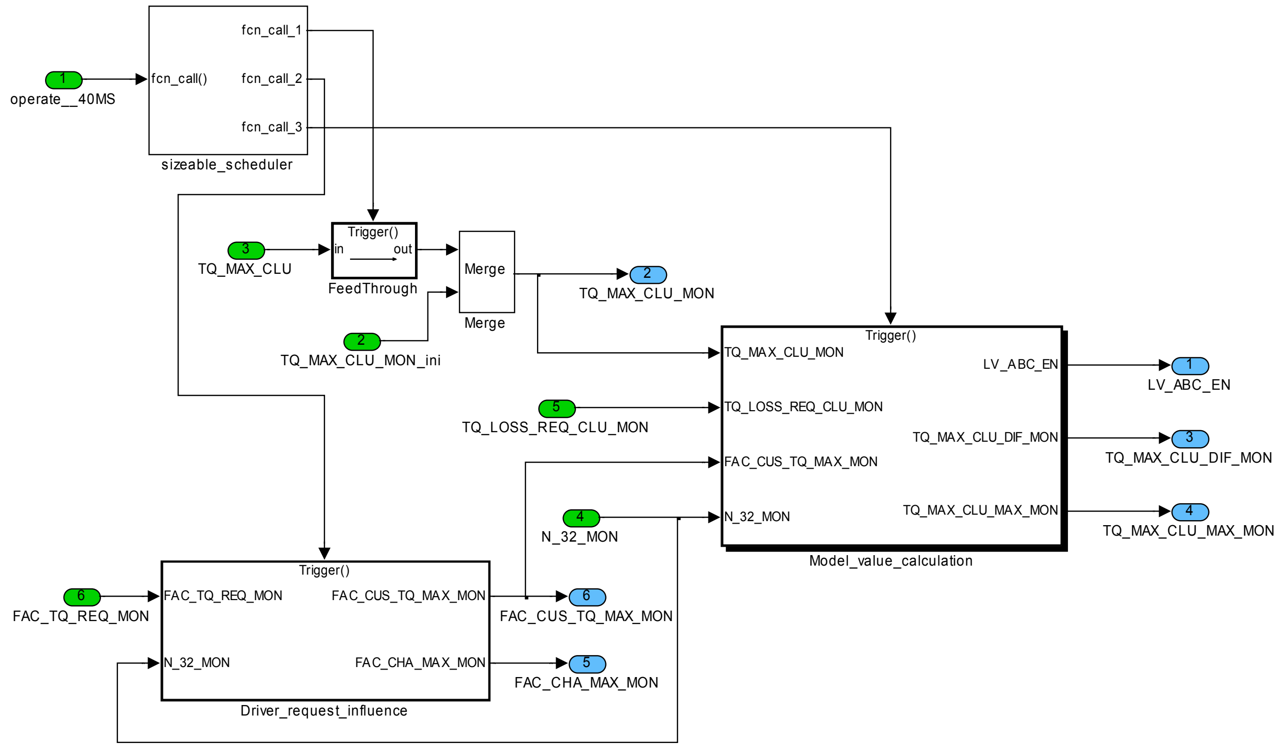


Figure 3 ECM2_DTSYSTQMAX/ Operate

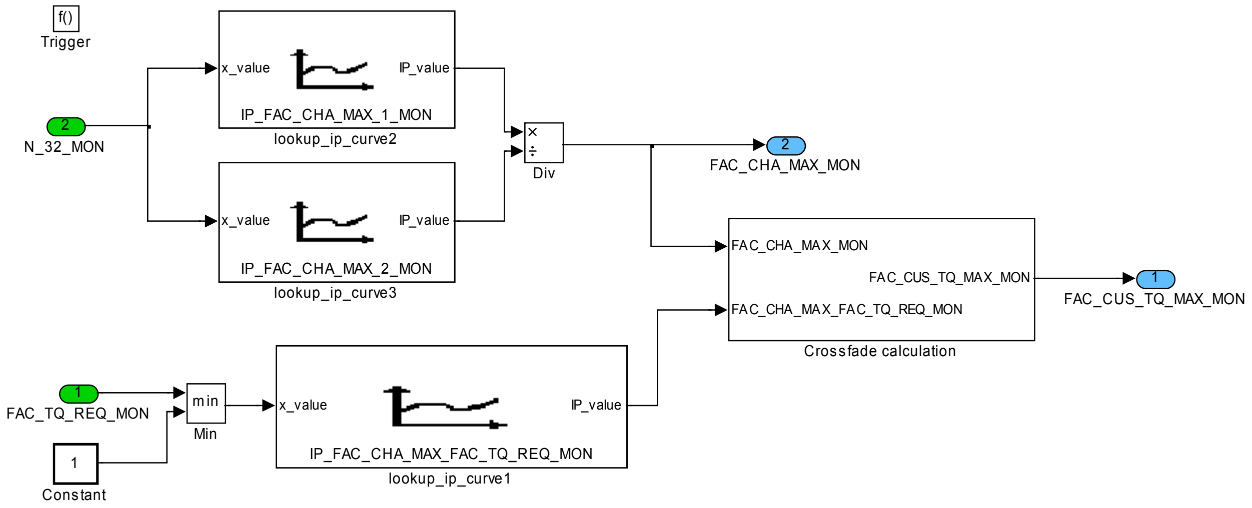


Figure 4 ECM2_DTSYSTQMAX/ Operate/ Driver_request_influence

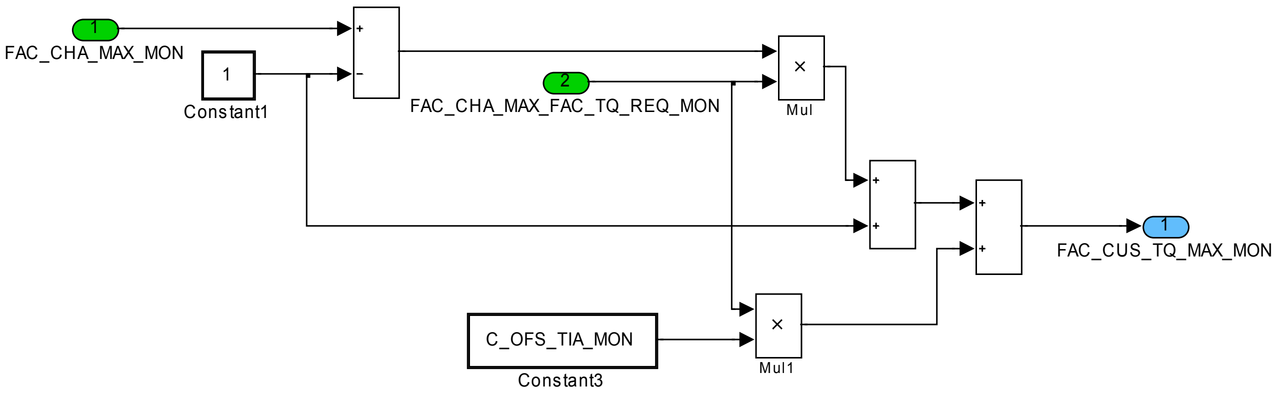


Figure 5 ECM2_DTSYSTQMAX/ Operate/ Driver_request_influence/Crossfade calculation

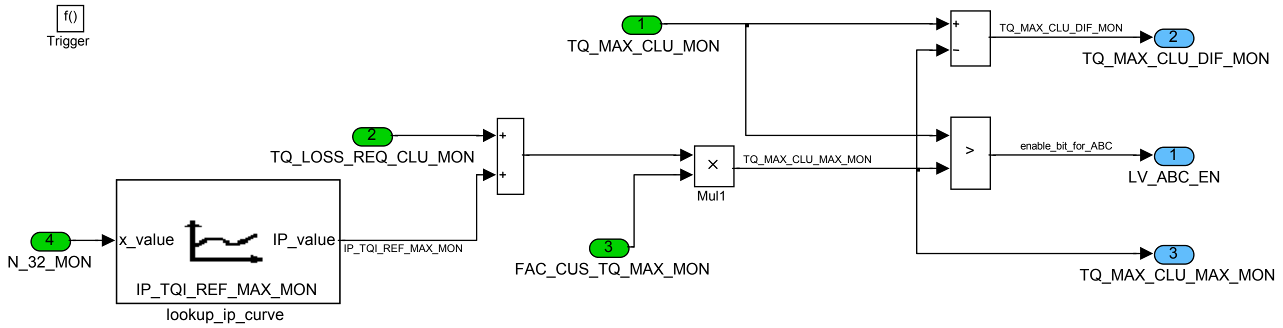


Figure 6 ECM2_DTSYSTQMAX/ Operate/ Model_value_calculation

11.13 Monitoring CAN-signals

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQI_MSR_MON	O/V	0...FFH	0...510	2	[Nm]
Torque demand from drag control CAN 11H					
LV_ERR_MSR_INH_MON	O/V	0...1H	0...1	1	[-]
Logical variable for fault in MSR present					
T_MSR_INH_MON	O/V	0...FFH	0...10.2	0.04	[s]
Timer for inhibition of MSR torque demand in monitoring level					
ABC_MSR_INH_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for fault in MSR present					
TQ_MSR_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque demand from drag control BN 2000					
TQ_WHEEL_LDM_MON	O/V	8000...7FFFH	-32768...32767	1	[Nm]
LDM torque request for monitoring					
LV_ERR_LDM_INH_MON	O/V	0...1H	0...1	1	[-]
Logical variable for fault in LDM present					
T_LDM_INH_MON	V	0...FFH	0...10.2	0.04	[s]
Timer for inhibition of LDM torque demand in monitoring level					
ABC_LDM_INH_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for fault in LDM present					
LV_ERR_LDM_MON	V	0...1H	0...1	1	[-]
LDM error mon					
LV_LDM_OFF_ECU_MON	V	0...1H	0...1	1	[-]
LDM off mon					
LV_LDM_OFF_ECU_1_MON	V	0...1H	0...1	1	[-]
LDM Irreversible off mon					
LV_LDM_OFF_ECU_2_MON	V	0...1H	0...1	1	[-]
LDM reversible off mon					
T_LDM_BRAKE_DET_MON	V	0...FFH	0...10.2	0.04	[s]
Brake delay timer mon					
TQ_DCC_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
DCC torque request for monitoring					
LV_ERR_DCC_INH_MON	O/V	0...1H	0...1	1	[-]
Logical variable for fault in DCC present					
T_DCC_INH_MON	V	0...FFH	0...10.2	0.04	[s]
Timer for inhibition of DCC torque demand in monitoring level					
ABC_DCC_INH_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for fault in DCC present					
LV_ERR_DCC_REV_MON	V	0...1H	0...1	1	[-]
Reversible DCC error mon					
LV_ERR_DCC_IRREV_MON	V	0...1H	0...1	1	[-]
Irreversible DCC error mon					
LV_DCC_OFF_ECU_MON	V	0...1H	0...1	1	[-]
Irreversible DCC error mon					
T_DCC_BRAKE_DET_MON	V	0...FFH	0...10.2	0.04	[s]
Brake delay timer mon					
TQI_AMT_MON	O/V	0...FFH	0...510	2	[Nm]
Torque demand from AMT control CAN 11H					
LV_ERR_AMT_INH_MON	O/V	0...1H	0...1	1	[-]
Logical variable for fault in MSR present					
T_AMT_INH_MON	O/V	0...FFH	0...10.2	0.04	[s]
Timer for inhibition of AMT torque demand in monitoring level					
ABC_AMT_INH_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for fault in AMT present					

TQ_AMT_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque demand from drag control BN 2000					
STATE_ERR_AMT_MON	V	0...FFH	0...255	1	[-]
State of AMT Error intervention					
T_STATE_AMT_1_DIAG_MON	V	0...FFH	0...10.2	0.04	[s]
Intervention in shift phase 1					
T_STATE_AMT_2_DIAG_MON	V	0...FFH	0...10.2	0.04	[s]
Intervention in shift phase 2					
T_STATE_AMT_3_DIAG_MON	V	0...FFH	0...10.2	0.04	[s]
Intervention in shift phase 2					
T_STATE_CLU_AMT_DIAG_MON	V	0...FFH	0...10.2	0.04	[s]
Timer clutch state 3					
STATE_AMT_OLD_MON	V	0...3H	0...3	1	[-]
Previos shift phase3					
T_TQI_AMT_DIAG_MON	V	0...FFH	0...10.2	0.04	[s]
Intervention timer 1 or 3					
TQI_AMT_PLAUS_MON	V	0...FFH	0...510	2	[Nm]
Plausibility torque					
TQI_GS_MON	O/V	0...FFH	0...510	2	[Nm]
Torque demand from drag control CAN 11H					
LV_ERR_GS_INH_MON	O/V	0...1H	0...1	1	[-]
Logical variable for fault in GS present					
T_GS_INH_MON	O/V	0...FFH	0...10.2	0.04	[s]
Timer for inhibition of GS torque demand in monitoring level					
ABC_GS_INH_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for fault in GS present					
TQ_GS_MON	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque demand from drag control BN 2000					

Input data:

	ERR_SYM_BN_ETCU	ERR_SYM_BN_TQ_AMT	ERR_SYM_BN_TQ_DCC
ERR_SYM_TOUT_AMT_1		LV_AMT_ES	LV_AMT_INC_ACT
LV_AT	LV_BRAKE_MON	LV_DCC_INC_ACT	LV_DCC_OFF_ACK
LV_DCC_OFF_ECU	LV_ERR_BLS_PLAUS	LV_ERR_BN_ETCU	LV_ERR_BN_TCS
LV_ERR_BN_TQ_AMT	LV_ERR_BN_TQ_DCC	LV_ERR_BN_TQ_ETCU	LV_ERR_BN_TQ_TCS
LV_ERR_BN_VS_TCS	LV_ERR_CAN_BOFF	LV_ERR_CRK	LV_ERR_CS
LV_ERR_TOUT_AMT_1	LV_ERR_TOUT_ASR_1	LV_ERR_TOUT_ASR_3	LV_ERR_TOUT_ETCU_1
LV_ERR_TOUT_ETCU_2	LV_TQ_AMT_DEC_REQ	LV_ES	LV_GS
LV_GS_ENA_INC	LV_GS_INC_ACT	LV_IGK_MON	LV_MSR_ACT
LV_TQ_AMT_INC_REQ	LV_TQ_ASR_REQ	LV_TQ_GS_INC_REQ	LV_TQ_MSR_REQ
LV_TQI_MON_ACT_MON	LV_VAR_BN	LV_VAR_DCC	
LV_TQI_REQ_CAN_INH	STATE_AMT	STATE_DCC_INTV	STATE_DCC_OFF_REQ
STATE_DCC_PUC_INH	LV_CDN_VB_CAN_TQ_DIAG	LV_DI_TQ_REQ_CAN_MPI_GDI	TQ_AMT_FAST_BN
TQ_AMT_SLOW_BN	TQ_DCC_FAST_BN	TQ_DCC_SLOW_BN	TQ_GS_FAST_BN
TQ_GS_FAST_INC_BN	TQ_GS_SLOW_BN	TQ_LOSS_MON	TQ_TCS_FAST_BN
TQ_TCS_SLOW_BN	TQI_AMT_FAST_DEC	TQI_AMT_FAST_INC	TQI_AMT_REQ_CAN
TQI_ASR_FAST_CAN	TQI_DCC_FAST_INC	TQI_GS_FAST_INC	TQI_GS_FAST_REQ_CAN
TQI_MSR_CAN	TQI_MSR_FAST	TQI_REQ_FAST_SEL	TQI_REQ_TRA_FAST
VB	VS_FIL	LV_LDM_ACT	LV_ERR_BN_LDM
LV_LDM_LIH_CAN	LV_LDM_OFF_ECU	LV_TQ_WHEEL_LDM_REQ	TQ_WHEEL_LDM_BN
ERR_SYM_BN_LDM	TQ_SP_WHEEL	LV_LDM_ENA_PLAUS_ERR	LV_TQ_WHEEL_LDM_BN_ERR
GR_DT	LV_ERR_PVS	LV_ETCU_DISABLE_CAN	LV_DCC_LIH_CAN
LV_TCS_LIH_CAN	LV_ETCU_LIH_CAN	LV_LDM_CAN_INI	LV_VAR_TCT
LV_TCT_LIH_CAN			

Import actions:

ACTION_ECM3_Service6TaskPfm(IN <n>)
ACTION_ECM3_Service7TaskPfm(IN <n>)
ACTION_ECM3_Service8TaskPfm(IN <n>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The argument n has the value 3.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service6TaskPfm() and ACTION_ECM3_Service7TaskPfm().

Import actions:

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)

ACTION_ECM3_ChkCpl(IN <>, IN <>)

ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

11.13.1 MSR monitoring

FUNCTION DESCRIPTION:

General information:

The torque demands received by the CAN-bus are monitored.

The input signals are read direct from CAN.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1
Deactivation: otherwise
Initialization: TQI_MSR_MON = 0
LV_ERR_MSR_INH_MON = 0
T_MSR_INH_MON = 0
ABC_MSR_INH_MON = 0
TQ_MSR_MON = -1024 Nm
Update Rate: 40 msec

Formula section:

LV_VAR_BN = 0:

```
IF   LV_ERR_MSR_INH_MON = 0
THEN
     TQI_MSR_MON = C_TQI_STND_MON  * TQI_MSR_CAN
     IF     LV_MSR_ACT = 1                                    and
            LV_ERR_TOUT_ASR_1             = 0                 and
            LV_ERR_TOUT_ASR_3             = 0                 and
            LV_DI_TQ_REQ_CAN_MPI_GDI      = 0                 and
            LV_TQI_REQ_CAN_INH            = 0                 and
            LV_TCS_LIH_CAN                = 0                 and
            LV_CDN_VB_CAN_TQ_DIAG         = 1

     THEN
            Initialize Timer: T_MSR_INH_MON = C_T_MSR_INH_MON
            IF     (LV_TQ_ASR_REQ = 1)                        OR
```

```
                  (LV_TQ_MSR_REQ= 0)                          OR
                  (TQI_MSR_CAN = 0)                           OR
                  ((TQI_ASR_FAST_CAN + TQI_MSR_CAN) ≠ FFH)    OR
                  ((VB > C_CAN_VB_MIN_DIAG_MON) AND (LV_ERR_CAN_BOFF=1))
            THEN  after debouncing with ABC_MSR_INH_MON
                  Inhibit MSR non reversible until the initialization conditions for error bits
                  are fulfilled (see application incidences for process monitoring):
                  LV_ERR_MSR_INH_MON = 1
                  TQI_MSR_MON = 0
                  T_MSR_INH_MON = 0
            ENDIF
        ELSE
            IF    T_MSR_INH_MON = 0
            THEN  TQI_MSR_MON = 0
            ELSE  Decrement Timer T_MSR_INH_MON
            ENDIF
        ENDIF

    ELSE
        TQI_MSR_MON = 0
        T_MSR_INH_MON = 0
    ENDIF
```

LV_VAR_BN = 1:

```
IF   LV_ERR_MSR_INH_MON = 0
THEN
    TQ_MSR_MON = TQ_TCS_FAST_BN
    IF      LV_MSR_ACT = 1                              and
            LV_DI_TQ_REQ_CAN_MPI_GDI  = 0               and
            LV_TQI_REQ_CAN_INH = 0                      and
            LV_TCS_LIH_CAN = 0                          and
            LV_ERR_BN_TQ_TCS = 0                        and
            LV_ERR_BN_VS_TCS = 0                        and
            LV_ERR_BN_TCS = 0                           and
```

```
                LV_CDN_VB_CAN_TQ_DIAG       = 1

        THEN
                Initialize Timer: T_MSR_INH_MON = C_T_MSR_INH_MON
                IF      (LV_TQ_ASR_REQ = 1)                                  OR
                        (LV_TQ_MSR_REQ= 0)                                   OR
                        (TQ_TCS_FAST_BN = 8000H)                             OR
                        (TQ_TCS_SLOW_BN = 8000H)                             OR
                        ((VB > C_CAN_VB_MIN_DIAG_MON) AND (LV_ERR_CAN_BOFF = 1))
                THEN    after debouncing with ABC_MSR_INH_MON
                        Inhibit MSR non reversible until the initialization conditions for error bits
                        are fulfilled (see application incidences for process monitoring):
                        LV_ERR_MSR_INH_MON = 1
                        TQ_MSR_MON = -1024 Nm
                        T_MSR_INH_MON = 0
                ENDIF
        ELSE
                IF      (T_MSR_INH_MON = 0)
                THEN    TQ_MSR_MON = -1024 Nm
                ELSE    Decrement Timer T_MSR_INH_MON
                ENDIF
    ELSE
        TQ_MSR_MON = -1024 Nm
        T_MSR_INH_MON = 0
    ENDIF
```

Explanation of some MSR inhibit conditions listed above:

On an error free CAN 11H transmission: TQI_ASR_FAST_CAN is the same, but bitinverse value of TQI_MSR_CAN

The CAN-interface is directly checked for the CAN-Bus OFF fault

Anti-bounce counter:	ABC_MSR_INH_MON
Anti-bounce counter increment:	C_ABC_INC_MSR_INH_MON
Anti-bounce counter maximum:	C_ABC_MAX_MSR_INH_MON

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ABC_INC_MSR_INH_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_MSR_INH_MON	1	1...FFH	1...255	1	[-]
Maximum value of anti-bounce counter					
C_T_MSR_INH_MON	1	0...FFH	0...10.2	0.04	[s]
Delay time for disable of MSR torque demand in monitoring level					
C_CAN_VB_MIN_DIAG_MON	1	0...FFH	0...25.89843	0.1015625	[V]
Threshold of battery voltage for CAN diagnosis					
C_TQI_STND_MON	1	0...FFH	0...510	2	[Nm]
Maximum indicated engine torque (scaling factor for torque demands from CAN)					

11.13.2 DCC monitoring

11.13.2.1 DCCTorque monitoring

FUNCTION DESCRIPTION:

General information:

The torque demands received by the CAN-bus are monitored.

The input signals are read direct from CAN.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialization:
TQ_DCC_MON = -1024Nm
LV_ERR_DCC_INH_MON = 0
T_DCC_INH_MON = 0
ABC_DCC_INH_MON = 0

Update Rate: 40 msec

Formula section:

LV_VAR_DCC = 1:

```
IF   LV_ERR_DCC_INH_MON = 0
THEN
     TQ_DCC_MON = TQ_DCC_FAST_BN       (physical converted)
     IF     LV_DCC_INC_ACT = 1                                  and
            LV_ERR_BN_TQ_DCC = 0                                and
            LV_DCC_LIH_CAN = 0                                  and
            LV_ERR_CAN_BOFF = 0                                 and
            STATE_DCC_PUC_INH < 03H                             and
            STATE_DCC_OFF_REQ = 02 H                            and
            LV_DCC_OFF_ECU = 0                                  and
            LV_CDN_VB_CAN_TQ_DIAG = 1
     THEN
            Initialize Timer: T_DCC_INH_MON = C_T_DCC_INH_MON
            IF     (LV_TQ_DCC_INC_REQ= 0)                       OR
                   (TQ_DCC_FAST_BN = 8000H)                     OR
```

```
                (TQ_DCC_SLOW_BN = 8000H)                              OR
                LV_DCC_OFF_ECU_MON = 1                                OR
                ((VB > C_CAN_VB_MIN_DIAG_MON) AND (LV_ERR_CAN_BOFF = 1))
            THEN after debouncing with ABC_DCC_INH_MON
                Inhibit DCC non reversible until the initialization conditions for error bits
                are fulfilled (see application incidences for process monitoring):
                LV_ERR_DCC_INH_MON = 1
                TQ_DCC_MON = -1024 Nm
                T_DCC_INH_MON = 0
            ENDIF
        ELSE
            IF    T_DCC_INH_MON = 0
            THEN  TQ_DCC_MON = -1024 Nm
            ELSE  Decrement Timer T_DCC_INH_MON
            ENDIF
        ENDIF
    ELSE
        TQ_DCC_MON = -1024 Nm
        T_DCC_INH_MON = 0
    ENDIF
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ABC_INC_DCC_INH_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_DCC_INH_MON	1	1...FFH	1...255	1	[-]
Maximum value of anti-bounce counter					
C_T_DCC_INH_MON	1	0...FFH	0...10.2	0.04	[s]
Delay time for disable of DCC torque demand in monitoring level					

11.13.2.2 DCC Plausibility Monitoring

FUNCTION DESCRIPTION:

Description:

The Conditions for DCC in the function level are devided in reversible and irreversible conditions. This function is monitored in the DCC torque monitoring.

Application conditions:

Initialisation: all 0 at reset or LV_IGK 0 --> 1 or clear FMY

Recurrence: 40 ms

Activation: *LV_TQI_MON_ACT_MON = 1; LV_VAR_DCC = 1*

Deactivation: -

Formula section:

```
IF   ( LV_ERR_BN_TQ_DCC      = 1  AND  ERR_SYM_BN_TQ_DCC        = SYM_2 )  or
                                                                "only timeout"
       LV_ERR_BLS_PLAUS      = 1                                              or
       LV_ERR_CS             = 1                                              or
       LV_DI_TQ_REQ_CAN_MPI_GDI    =1                                         or
       LV_ERR_CRK = 1
Then LV_ERR_DCC_REV_MON = 1 after debounce
Else IF    LV_ERR_DCC_REV_MON = 0 ("not debounced yet")
     Then  reset counter  to 0                 // reset counter to 0, no rebouncing!
     Else  IF     LV_DCC_OFF_ACK = 1
           then   LV_ERR_DCC_REV_MON = 0       // reset error and counter to 0, no rebouncing!
                  reset counter to 0
           else unchanged
endif

IF         LV_BRAKE_MON = 1
Then       T_DCC_BRAKE_DET_MON$_n$ = T_DCC_BRAKE_DET_MON$_{n-1}$ + 40 ms
else       T_DCC_BRAKE_DET_MON$_n$ =0 ms
endif
```

```
IF   ( LV_ERR_BN_TQ_DCC  = 1 AND ERR_SYM_BN_TQ_DCC  = SYM_1 or SYM_3 ) or
     "ONLY ALIVE AND CHECKSUM ERRORS"
     LV_TQI_REQ_CAN_INH      = 1                                  or
     LV_ERR_PVS              = 1                                  or
     [ (T_DCC_BRAKE_DET_MON > C_T_MAX_BRAKE_DET_DCC_MON) and
     STATE_DCC_INTV == 01H ]
Then LV_ERR_DCC_IRREV_MON = 1 after debounce
Endif

IF          LV_ERR_DCC_IRREV_MON = 1 or LV_ERR_DCC_REV _MON = 1
Then        LV_DCC_OFF_ECU_MON = 1
Else IF     LV_DCC_OFF_ACK = 1
     Then   LV_DCC_OFF_ECU_MON = 0
     Else   unchanged
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_DCC_REV_MON	1	0...FFH	0...255	1	[-]
Increment counter					
C_ABC_MAX_DCC_REV_MON	1	1...FFH	1...255	1	[-]
Maximum counter					
C_ABC_INC_DCC_IRREV_MON	1	0...FFH	0...255	1	[-]
Increment counter					
C_ABC_MAX_DCC_IRREV_MON	1	1...FFH	1...255	1	[-]
Maximum counter					
C_T_MAX_BRAKE_DET_DCC_MON	1	0...FFH	0...10.2	0.04	[s]
Maximum Brake delay timer					

11.13.3 LDM monitoring

11.13.3.1 LDM Torque monitoring

FUNCTION DESCRIPTION:

General information:

The torque demands received by the CAN-bus are monitored.

The input signals are read direct from CAN.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1

Deactivation: otherwise

Initialization:

Initialization:	TQ_WHEEL_LDM_MON	= -32000 Nm
	LV_ERR_LDM_INH_MON	= 0
	T_LDM_INH_MON	= 0
	ABC_LDM_INH_MON	= 0

Update Rate: 40 msec

Formula section:

LV_VAR_BN_LDM = 1:

```
IF   LV_ERR_LDM_INH_MON = 0
THEN
     TQ_WHEEL_LDM_MON = TQ_WHEEL_LDM_BN
     IF      LV_LDM_ACT = 1                                        and
             LV_ERR_BN_LDM = 0                                     and
             LV_LDM_LIH_CAN = 0                                    and
             LV_ERR_CAN_BOFF = 0                                   and
             LV_LDM_OFF_ECU = 0                                    and
             LV_CDN_VB_CAN_TQ_DIAG        = 1

     THEN
             Initialize Timer: T_LDM_INH_MON = C_T_LDM_INH_MON
             IF      (LV_TQ_WHEEL_LDM_REQ= 0)                  OR
                     (LV_TQ_WHEEL_LDM_BN_ERR = 1)              OR
                     LV_LDM_OFF_ECU_MON = 1                    OR
```

```
            ((VB > C_CAN_VB_MIN_DIAG_MON) AND (LV_ERR_CAN_BOFF = 1))
        THEN after debouncing with ABC_LDM_INH_MON
            Inhibit LDM non reversible until the initialization conditions for error bits
            are fulfilled (see application incidences for process monitoring):
            LV_ERR_LDM_INH_MON = 1
            TQ_WHEEL_LDM_MON = -32000Nm
            T_LDM_INH_MON = 0
        ENDIF

    ELSE
        IF   T_LDM_INH_MON = 0
        THEN TQ_WHEEL_LDM_MON = -32000Nm
        ELSE Decrement Timer T_LDM_INH_MON
        ENDIF
    ENDIF

ELSE
    TQ_WHEEL_LDM_MON = -32000Nm
    T_LDM_INH_MON = 0
ENDIF
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ABC_INC_LDM_INH_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_LDM_INH_MON	1	1...FFH	1...255	1	[-]
Maximum value of anti-bounce counter					
C_T_LDM_INH_MON	1	0...FFH	0...10.2	0.04	[s]
Delay time for disable of LDM torque demand in monitoring level					

11.13.3.2 LDM Plausibility Monitoring

FUNCTION DESCRIPTION:

Description:

The Conditions for LDM in the function level are devided in reversible and irreversible conditions. This function is monitored in the LDM torque monitoring.

Application conditions:

Initialisation: all 0 at reset or LV_IGK 0 --> 1 or clear FMY

Recurrence: 40 ms

Activation: LV_TQI_MON_ACT_MON = 1 and
LV_VAR_BN_LDM = 1 **and**
LV_LDM_CAN_INI = 0

Deactivation: -

Formula section:

```
IF      LV_BRAKE_MON = 1
Then    $\text{T\_LDM\_BRAKE\_DET\_MON}_{n} = \text{T\_LDM\_BRAKE\_DET\_MON}_{n-1} + 40$ ms
else    $\text{T\_LDM\_BRAKE\_DET\_MON}_{n} = 0$ ms
endif


IF      [(T_LDM_BRAKE_DET_MON > C_T_MAX_BRAKE_DET_LDM_MON)
        and   STATE_LDM_INTV == 01H ]                    or
        (LV_LDM_ENA_PLAUS_ERR = 1
        and   LC_LDM_ENA_PLAUS_ERR = 1)
Then    LV_ERR_LDM_MON = 1 after debounce
Else    If    LV_ERR_LDM_MON = 0 ("not debounced yet")
        Then  rebounce counter to 0
        Else  LV_ERR_LDM_MON = 0
              after rebounce
        endif
Endif
```

Calculation of LDM irrversible off

```
IF   ( LV_ERR_BN_LDM  = 1 AND ERR_SYM_BN_LDM  = SYM_1 or SYM_3 ) or
       "ONLY ALIVE AND CHECKSUM ERRORS"
       LV_TQI_REQ_CAN_INH              = 1                        or
       LV_ERR_PVS                      = 1                        or
       LV_ERR_LDM_MON                  = 1
Then LV_LDM_OFF_ECU_1_MON = 1           //LDM  irreversible off
Endif
```

Calculation of reversible off

```
IF   ( LV_ERR_BN_LDM  = 1  AND  ERR_SYM_BN_LDM      = SYM_2 )             or
                                                                "only timeout"
       LV_ERR_BLS_PLAUS         = 1                                        or
       LV_ERR_CS                = 1                                        or
       LV_ERR_CRK               = 1                                        or
       LV_DI_TQ_REQ_CAN_MPI_GDI     = 1                                    or
       LV_ETCU_DISABLE_CAN      = 1

Then LV_LDM_OFF_ECU_2_MON       = 1     //LDM reversible LDM off
Else  LV_LDM_ÓFF_ECU_2_MON      = 0
Endif
```

```
IF          LV_LDM_OFF_ECU_1_MON = 1 or LV_LDM_OFF_ECU_2_MON = 1
Then        LV_LDM_OFF_ECU_MON = 1
Else        LV_LDM_OFF_ECU_MON = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_LDM_MON	1	0...FFH	0...255	1	[-]
Increment counter					
C_ABC_MAX_LDM_MON	1	1...FFH	1...255	1	[-]
Maximum counter					
C_T_MAX_BRAKE_DET_LDM_MON	1	0...FFH	0...10.2	0.04	[s]
Maximum Brake delay timer					

11.13.4 AMT monitoring

11.13.4.1 AMT torque monitoring

Application conditions:

Activation:	LV_TQI_MON_ACT_MON = 1
Deactivation:	otherwise
Initialization:	TQI_AMT_MON = 0 LV_ERR_AMT_INH_MON = 0 T_AMT_INH_MON = 0 ABC_AMT_INH_MON = 0 TQ_AMT_MON = -1024 Nm
Update Rate:	40 msec

LV_VAR_BN = 0:

```
IF   LV_ERR_AMT_INH_MON = 0
THEN
     TQI_AMT_MON = C_TQI_STND_MON  * TQI_AMT_REQ_CAN
     IF      LV_AMT_INC_ACT = 1                                          and
             LV_TQI_REQ_CAN_INH = 0                                      and
             LV_ERR_TOUT_AMT_1 = 0                                       and
             LV_AMT_LIH_CAN = 0                                          and
             LV _DI_TQ_REQ_CAN_MPI_GDI   = 0                             and
             LV_CDN_VB_CAN_TQ_DIAG = 1

     THEN
             Initialize Timer: T_AMT_INH_MON = C_T_AMT_INH_MON
             IF      (LV_TQ_AMT_INC_REQ = 0)                          OR
                     TQI_AMT_REQ_CAN = FFH                            OR
                     (STATE_ERR_AMT_MON> 00H
                     for C_T_DLY_AMT_INH_MON)                         OR
                     ((VB > C_CAN_VB_MIN_DIAG_MON) AND (LV_ERR_CAN_BOFF=1))
             THEN    after debouncing with ABC_AMT_INH_MON
                     Inhibit AMT non reversible until the initialization conditions for error bits
                     are fulfilled (see application incidences for process monitoring):
```

```
                LV_ERR_AMT_INH_MON = 1
                TQI_AMT_MON = 0
                T_AMT_INH_MON = 0
            ENDIF

        ELSE
            IF    (T_AMT_INH_MON = 0)
            THEN  TQI_AMT_MON = 0
            ELSE  Decrement Timer T_AMT_INH_MON
            ENDIF
        ENDIF

    ELSE
        TQI_AMT_MON = 0
        T_AMT_INH_MON = 0
    ENDIF
```

LV_VAR_BN = 1:

```
    IF  LV_ERR_AMT_INH_MON = 0
    THEN
        TQ_AMT_MON = TQ_AMT_FAST_BN
        IF      LV_AMT_INC_ACT = 1                              and
                LV_ERR_BN_TQ_AMT = 0                            and
                LV_ERR_BN_ETCU = 0                              and
                LV_TQI_REQ_CAN_INH = 0                          and
                LV_AMT_LIH_CAN = 0                              and
                LV_DI_TQ_REQ_CAN_MPI_GDI    = 0                 and
                LV_CDN_VB_CAN_TQ_DIAG       = 1

        THEN
                Initialize Timer: T_AMT_INH_MON = C_T_AMT_INH_MON
                IF      (LV_TQ_AMT_INC_REQ = 0)                 OR
                        (TQ_AMT_FAST_BN = 8000H)                OR
                        (TQ_AMT_SLOW_BN = 8000H)                OR
```

```
                (STATE_ERR_AMT_MON> 00H
                for C_T_DLY_AMT_INH_MON)                    OR
                ((VB > C_CAN_VB_MIN_DIAG_MON) AND (LV_ERR_CAN_BOFF = 1))
            THEN after debouncing with ABC_AMT_INH_MON
                Inhibit AMT non reversible until the initialization conditions for error bits
                are fulfilled (see application incidences for process monitoring):
                LV_ERR_AMT_INH_MON = 1
                TQ_AMT_MON = -1024 Nm
                T_AMT_INH_MON = 0
            ENDIF

        ELSE
            IF   (T_AMT_INH_MON = 0)
            THEN TQ_AMT_MON = -1024 Nm
            ELSE Decrement Timer T_AMT_INH_MON
            ENDIF
        ENDIF

    ELSE
        TQ_AMT_MON = -1024 Nm
        T_AMT_INH_MON = 0
    ENDIF
```

The CAN-interface is directly checked for the CAN-Bus OFF fault

Anti-bounce counter:	ABC_AMT_INH_MON
Anti-bounce counter increment:	C_ABC_INC_AMT_INH_MON
Anti-bounce counter maximum:	C_ABC_MAX_AMT_INH_MON

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ABC_INC_AMT_INH_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_AMT_INH_MON	1	1...FFH	1...255	1	[-]
Maximum value of anti-bounce counter					
C_T_AMT_INH_MON	1	0...FFH	0...10.2	0.04	[s]
Delay time for disable of AMT torque demand in monitoring level					
C_T_DLY_AMT_INH_MON	1	0...FFH	0...10.2	0.04	[s]
Delay time for incrementing ABC-counter - AMT monitoring level					

11.13.4.2 AMT plausibility monitoring

FUNCTION DESCRIPTION:

General information:

To supress incorrect Torque request of transmission unit a monitoring of messages is done.

Application conditions:

Initialization: 0

Recurrence: 40 ms

Activation: LV_TQI_MON_ACT_MON = 1 and LV_IGK_MON = 1

Signal flow diagram:

The structure is like the Level 1 function "AMT intervention plausibility"

Formula section:

```
If LV_VAR_BN = 0
Then        TQI_AMT_PLAUS_MON = TQI_AMT_MON
Else IF     LV_TQ_AMT_DEC_REQ = 1
      Then  TQI_AMT_PLAUS_MON =   TQI_AMT_FAST_DEC
else        TQI_AMT_PLAUS_MON =   TQI_AMT_FAST_INC
endif
```

Plausibility of shift phase:

```
If    LV_GS = 0
Then if  STATE_AMT = 0 and (TQI_AMT_PLAUS_MON <
                                            C_TQI_AMT_COND_DIAG_MON)
                Then   STATE_ERR_AMT_MON = 00H
                Else   STATE_ERR_AMT_MON = 01H
      End if
Else if     STATE_AMT = 0
       Then  if     LV_AMT_ES > 0
            Then  if     TQI_AMT_PLAUS_MON ≤ C_TQI_AMT_CAN_ES_MON and
                         VS_FIL < C_VS_FIL_AMT
                  Then   STATE_ERR_AMT_MON = 00H ; LV_AMT_ES_REQ =1
                  Else   STATE_ERR_AMT_MON = 10H
```

```
                End if
            Else    STATE_ERR_AMT_MON = 11H
            End if
        Else    if      STATE_AMT = 1
            Then    if      STATE_AMT_OLD_MON = 1
                Then    T_STATE_AMT_1_DIAG_MON = T_STATE_AMT_1_DIAG_MON + 40 ms
                Else    T_STATE_AMT_1_DIAG_MON = 0
                        T_TQI_AMT_DIAG_MON = 0
                End if


            Else    if      STATE_AMT = 2
                Then    if      STATE_AMT_OLD_MON = 2
                    Then    T_STATE_AMT_2_DIAG_MON = T_STATE_AMT_2_DIAG_MON + 40 ms
                    Else    T_STATE_AMT_2_DIAG_MON = 0
                            T_STATE_CLU_AMT_DIAG_MON = 0
                    End if
                Else    if      STATE_AMT_OLD_MON = 3
                    Then    T_STATE_AMT_3_DIAG_MON = T_STATE_AMT_3_DIAG_MON + 40 ms
                    Else    T_STATE_AMT_3_DIAG_MON = 0
                            T_TQI_AMT_DIAG_MON = 0
                    End if


            If T_STATE_AMT_1_DIAG_MON > C_T_MAX_STATE_AMT_1_DIAG_MON  OR
              T_STATE_AMT_2_DIAG_MON > C_T_MAX_STATE_AMT_2_DIAG _MON OR
            T_STATE_AMT_3_DIAG_MON > C_T_MAX_STATE_AMT_3_DIAG_MON
            Then    STATE_ERR_AMT_MON = 20H
            Else    if      STATE_AMT = 1 or 3
                Then    if  TQI_AMT_PLAUS_MON >
                            MAX(TQI_REQ_TRA_FAST; -1* TQ_LOSS_MON; TQI_MSR_FAST; TQI_DCC_FAST_INC; (TQ_SP_WHEEL/GR_DT)) +
```

```
                                                      C_TQI_AMT_DIAG_MON
                    Then  T_TQI_AMT_PLAUS_MON_DIAG =
                                    T_TQI_AMT_DIAG_MON + 40 ms
                    else  T_TQI_AMT_PLAUS_MON_DIAG =
                                    T_TQI_AMT_DIAG_MON
                    End if


                    If  T_TQI_AMT_PLAUS_MON_DIAG >
                                    C_T_MAX_TQI_AMT_DIAG_MON
                    Then  STATE_ERR_AMT_MON = 21H
                    Else  STATE_ERR_AMT_MON = 00H
                    End if
                Else  if    STATE_CLU_AMT = 3
                    Then  T_STATE_CLU_AMT_DIAG_MON =
                                         T_STATE_CLU_AMT_DIAG_MON +
                                    40 ms
                    Else  T_STATE_CLU_AMT_DIAG_MON =
                                    T_STATE_CLU_AMT_DIAG_MON
                    End if
                    if    T_STATE_CLU_AMT_DIAG_MON >
                                    C_T_MAX_STATE_CLU_AMT_DIAG_MON
                    Then  STATE_ERR_AMT_MON = 22H
                    Else  STATE_ERR_AMT_MON = 00H
                    End if
                End if
            End if
        End if
  End if
  ------------------------------------------------------
  STATE_AMT_OLD_MON = STATE_AMT
```

The flag LV_AMT_ES_REQ will be set until the end of driving cycle

```
IF LV_ES = 1 and LV_IGK_MON = 0 to 1
Then LV_AMT_ES_REQ = 0
Endif
```

Plausibility of Torque intervention about message counter:

```
IF ERR_SYM_TOUT_AMT_1 or_BN_TQ_AMT or BN_ETCU = "SYM_2"
Then STATE_ERR_AMT_MON = 40H (timeout CAN)
```

```
Else if ERR_SYM_TOUT_AMT_1 or_BN_TQ_AMT or BN_ETCU = "SYM_1"
    Then  STATE_ERR_AMT_MON = 50H (timeout CAN)
    Else  if ERR_SYM_TOUT_AMT_1 or_BN_TQ_AMT or BN_ETCU = "SYM_3"
          then STATE_ERR_AMT_MON = 30H (timeout CAN)
          else  STATE_ERR_AMT_MON  remains
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_FIL_AMT_MON	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed threshold engine stop					
C_T_MAX_STATE_AMT_1_DIAG_MON	1	0...FFH	0...10.2	0.04	[s]
Max Intervention in shift phase 1					
C_T_MAX_STATE_AMT_2_DIAG_MON	1	0...FFH	0...10.2	0.04	[s]
Max Intervention in shift phase 2					
C_T_MAX_STATE_AMT_3_DIAG_MON	1	0...FFH	0...10.2	0.04	[s]
Max Intervention in shift phase 3					
C_TQI_AMT_DIAG_MON	1	0...FFH	0...510	2	[Nm]
Plausibility torque intervention hysteresis					
C_T_MAX_TQI_AMT_DIAG_MON	1	0...FFH	0...10.2	0.04	[s]
Max. timer shift intervention phase 1or 3					
C_TQI_AMT_COND_DIAG_MON	1	0...FFH	0...510	2	[Nm]
Torque at shift phase 0					
C_TQI_AMT_CAN_ES_MON	1	0...FFH	0...510	2	[Nm]
Torque at engine stop					
C_T_MAX_STATE_CLU_AMT_DIAG_MON	1	0...FFH	0...10.2	0.04	[s]
Max. time at clutch					

11.13.5 GS monitoring

11.13.5.1 GS torque monitoring

FUNCTION DESCRIPTION:

General information:

The torque demands received by the CAN-bus are monitored.

The input signals are read direct from CAN.

Application conditions:

Activation: LV_TQI_MON_ACT_MON = 1 **and** (LV_AT = 1 **or** LV_VAR_TCT = 1)

Deactivation: otherwise

Initialization:
TQI_GS_MON = 0
LV_ERR_GS_INH_MON = 0
T_GS_INH_MON = 0
ABC_GS_INH_MON = 0
TQ_GS_MON = -1024 Nm

Update Rate: 40 msec

Formula section:

LV_VAR_BN = 0:

```
IF   LV_ERR_GS_INH_MON = 0
THEN
     TQI_GS_MON = C_TQI_STND_MON  * TQI_GS_FAST_REQ_CAN
     IF     LV_GS_INC_ACT            = 1          and
            LV_DI_TQ_REQ_CAN_MPI_GDI    = 0       and
            LV_ETCU_LIH_CAN          = 0          and
            LV_ERR_TOUT_ETCU_1       = 0          and
            LV_ERR_TOUT_ETCU_2       = 0          and
            LV_TQI_REQ_CAN_INH       = 0          and
            LV_CDN_VB_CAN_TQ_DIAG  = 1

     THEN
            Initialize Timer: T_GS_INH_MON = C_T_GS_INH_MON
```

```
                IF      (LV_TQ_GS_INC_REQ = 0)                                    OR
                        (TQI_GS_FAST_REQ_CAN = FFH)                               OR
                        [(VB > C_CAN_VB_MIN_DIAG_MON) AND (LV_ERR_CAN_BOFF=1)]
                        for C_T_DLY_GS_INH_MON}
                THEN    after debouncing with ABC_GS_INH_MON
                        Inhibit GS non reversible until the initialization conditions for error bits are
                        fulfilled (see application incidences for process monitoring):
                        LV_ERR_GS_INH_MON = 1
                        TQI_GS_MON = 0
                        T_GS_INH_MON = 0
                ENDIF

        ELSE
                IF      T_GS_INH_MON = 0
                THEN    TQI_GS_MON = 0
                ELSE    Decrement Timer T_GS_INH_MON
                ENDIF
        ENDIF

    ELSE
        TQI_GS_MON = 0
        T_GS_INH_MON = 0
    ENDIF
```

LV_VAR_BN = 1:

```
    IF   LV_ERR_GS_INH_MON = 0
    THEN
        TQ_GS_MON = TQ_GS_FAST_BN
        IF      LV_GS_INC_ACT           = 1                 and
                LV_ERR_BN_TQ_ETCU       = 0                 and
                LV_ERR_BN_ETCU          = 0                 and
                LV_ETCU_LIH_CAN         = 0                 and
                LV_TCT_LIH_CAN          = 0                 and
```

```
            LV_TQI_REQ_CAN_INH      = 0                              and
            LV_DI_TQ_REQ_CAN_MPI_GDI    = 0                          and
            LV_CDN_VB_CAN_TQ_DIAG       = 1

        THEN
            Initialize Timer: T_GS_INH_MON = C_T_GS_INH_MON
            IF      (LV_TQ_GS_INC_REQ = 0)                           OR
                    (TQ_GS_FAST_BN = 8000H)                          OR
                    (TQ_GS_SLOW_BN = 8000H)                          OR
                    [(VB > C_CAN_VB_MIN_DIAG_MON) AND (LV_ERR_CAN_BOFF=1)]
                    for C_T_DLY_GS_INH_MON}
            THEN    after debouncing with ABC_GS_INH_MON
                    Inhibit GS non reversible until the initialization conditions for error bits are
                    fulfilled (see application incidences for process monitoring):
                    LV_ERR_GS_INH_MON = 1
                    TQ_GS_MON = -1024 Nm
                    T_GS_INH_MON = 0
            ENDIF

        ELSE
            IF      T_GS_INH_MON = 0
            THEN    TQ_GS_MON = -1024 Nm
            ELSE    Decrement Timer T_GS_INH_MON
            ENDIF
        ENDIF

    ELSE
        TQ_GS_MON = -1024 Nm
        T_GS_INH_MON = 0
    ENDIF
```

The CAN-interface is directly checked for the CAN-Bus OFF fault

Anti-bounce counter:	ABC_GS_INH_MON
Anti-bounce counter increment:	C_ABC_INC_GS_INH_MON
Anti-bounce counter maximum:	C_ABC_MAX_GS_INH_MON

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ABC_INC_GS_INH_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_GS_INH_MON	1	1...FFH	1...255	1	[-]
Maximum value of anti-bounce counter					
C_T_GS_INH_MON	1	0...FFH	0...10.2	0.04	[s]
Delay time for disable of GS torque demand in monitoring level					
C_T_DLY_GS_INH_MON	1	0...FFH	0...10.2	0.04	[s]
Delay time for incrementing ABC-counter - GS monitoring level					

11.14 Monitoring of engine speed

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_N_32_MON	V/O	0...1H	0...1	1	[-]
Logical variable for ratio error between N_32 and N_32_SUB_MON					
N_32_MON	V/O	0...FFH	0...8160	32	[rpm]
Engine speed (monitoring level)					
N_32_SUB_MON	V	0...FFH	0...8160	32	[rpm]
Engine speed substitute monitoring level					
T_SEG_SW_MON	V/O	0...FFFFFFFFH	0...268435455	1	[µs]
Timer for engine speed monitoring					
ABC_N_32_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter N_32 ratio error					
N_32_DIF_REL_MON	V	0...FFFFH	0...255	3.8911e-3	[-]
Relative engine speed deviation monitoring level					

Input data:

N_32			

Import actions:

ACTION_ECM3_Service9TaskPfm(IN <>)
ACTION_ECM3_Service10TaskPfm(IN <>)
ACTION_ECM3_Service11TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 1.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service9TaskPfm() and ACTION_ECM3_Service10TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

ACTION_INFR_GetTSEGSWMON(OUT <>)

Note: This action is defined in chapter "Basic SW Inputs and Outputs " (IRS).

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_32_DIF_REL_MAX_MON	1	0...FFFFH	0...255	3.8911e-3	[-]
Maximum relative difference between N_32 and N_32_SUB_MON					
C_N_32_LIM_MON	1	1...FFH	32...8160	32	[rpm]
Lower limit of denominator for relative error calculation (initial value: 32)					
C_N_32_MIN_MON	1	0...FFH	0...8160	32	[rpm]
Lower limit for using the relative error concept					
C_ABC_INC_N_32_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment (additive value in case of N_32 ratio error)					
C_ABC_MAX_N_32_MON	1	1...FFH	1...255	1	[-]
Value at which N_32 ratio error is recognized, when reached					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_T_SEG_N_32_CLC	-	0...FFFFFFFFH	0... 137438953440	32	[rpm*µs]
Non calibratable data for N_32_SUB_MON calculation					

11.14.1 FUNCTION PART: ECM2_DTSYSN

General information:

The comparison of the engine speed N_32 to an engine speed substitute N_32_SUB_MON, calculated in the monitoring level, performs the monitoring of the engine speed used in the monitoring level.

N_32_SUB_MON is calculated from a segment timing information that is generated by a software timer solution in the BSW (T_SEG_SW_MON). The software timer is triggered by crankshaft segments and produces time stamps for the trigger events. From these timing informations the signal T_SEG_SW_MON is determined (resolution: 1µs).

The engine speed signals N_32 and N_32_SUB_MON are compared to each other by means of a relative deviation. The ratio of the difference of N_32 and N_32_SUB_MON related to the minimum of the two values is not allowed to deviate more than the maximum difference C_N_32_DIF_REL_MAX_MON. Otherwise a process fault is assumed. Since for a small engine speed the relative error concept is leading to a rather small permitted absolute difference between N_32 and N_32_SUB_MON, the monitoring could be too tight here. Therefore, as soon as both engine speed signals are smaller than a limit C_N_32_MIN_MON, the relative error is calculated by relating the absolute difference to the constant C_N_32_LIM_MON.

On an error free system N_32 is copied to N_32_MON. If an engine speed error is detected, the maximum value of N_32 and N_32_SUB_MON is copied to N_32_MON to prevent from a sleeping fault in the module ‚Monitoring of engine speed limitation‘.

Note: This version of engine speed monitoring is for systems where a second hardware timer, which in other versions was delivering the signal SEG_T_MES, is not available on the ECU and where the missing timer is substituted by a software timer solution in the BSW.

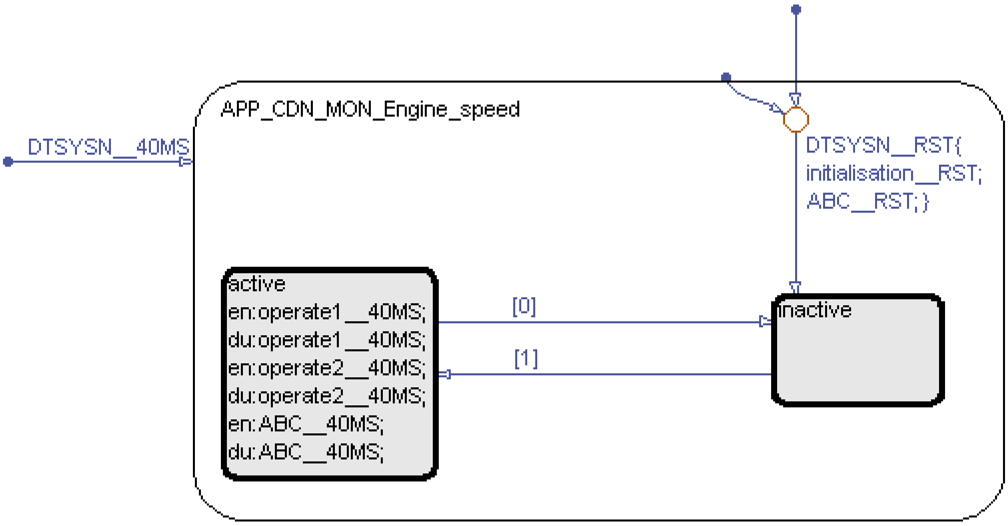
For configuration data:

NC_T_SEG_N_32_CLC is a constant for the calculation of N_32_SUB_MON from T_SEG_SW:

$$NC\_T\_SEG\_N\_32\_CLC = \frac{2*60*1000000}{NC\_CYL\_NR*\{resol.\ of\ \ N\_32\_SUB\_MON\}}$$

with NC_CYL_NR : number of cylinders

Application Condition



Note: DTSYSN__RST includes the function calls as defined in application incidences.

Function Description

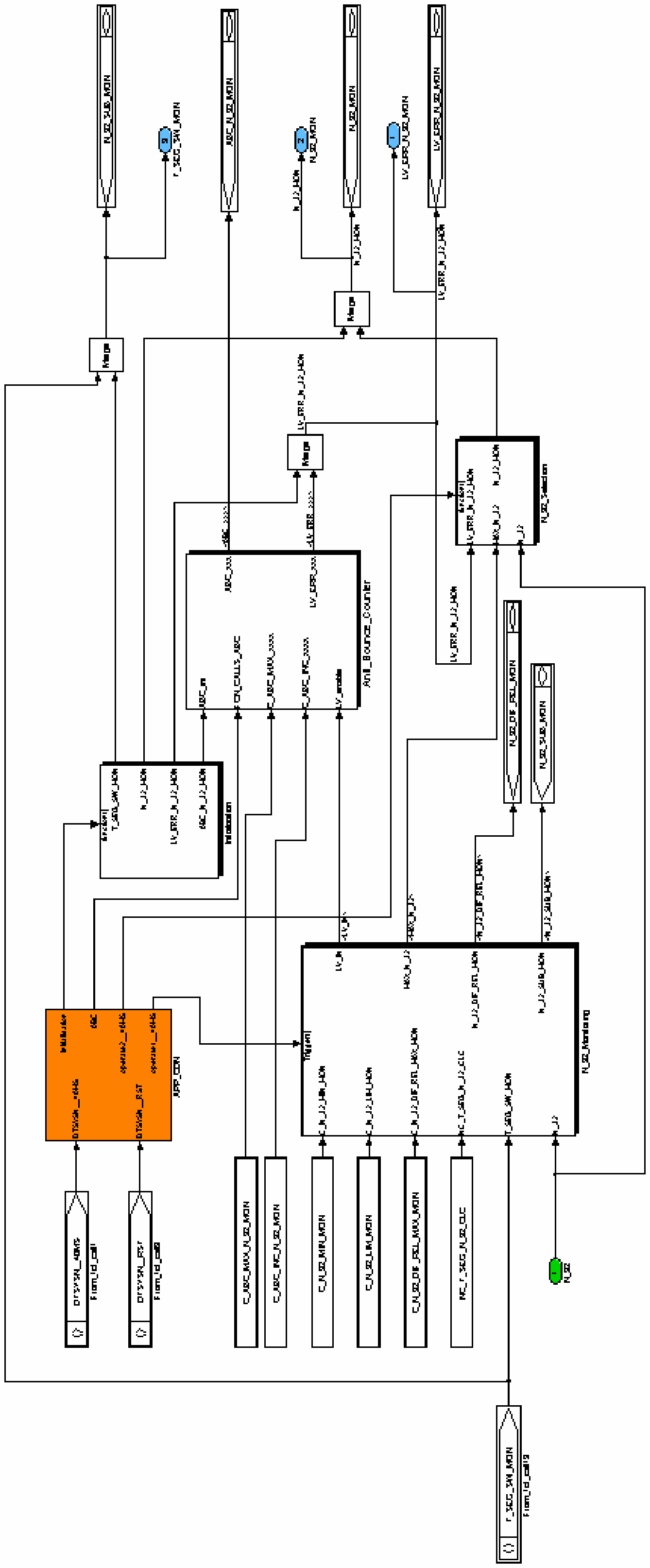


Figure 40 ECM2_DTSYSN

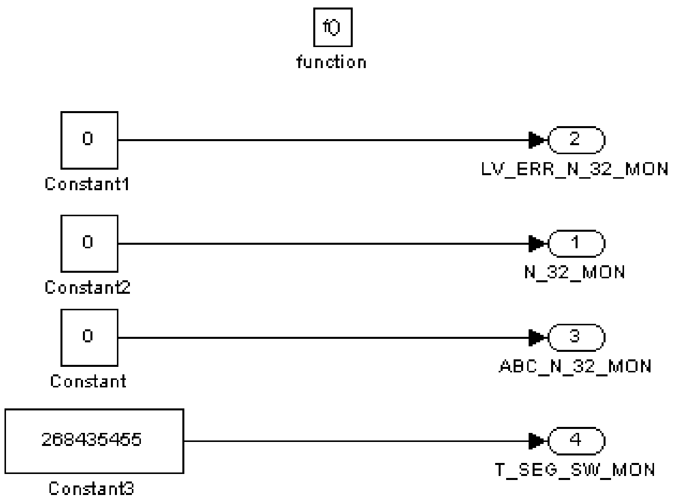


Figure 41 ECM2_DTSYSN/ INITIALISATION

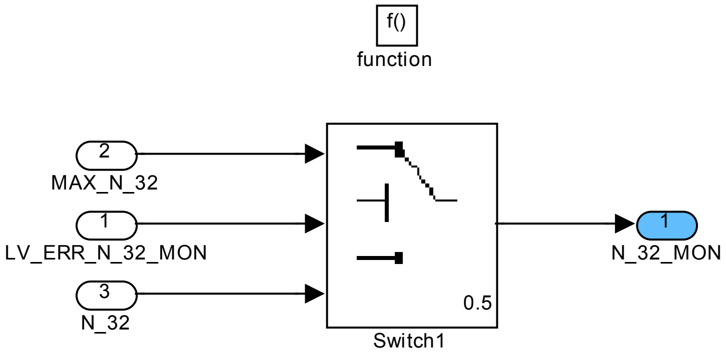


Figure 3 ECM2_DTSYSN/ N_32_MON_Selection

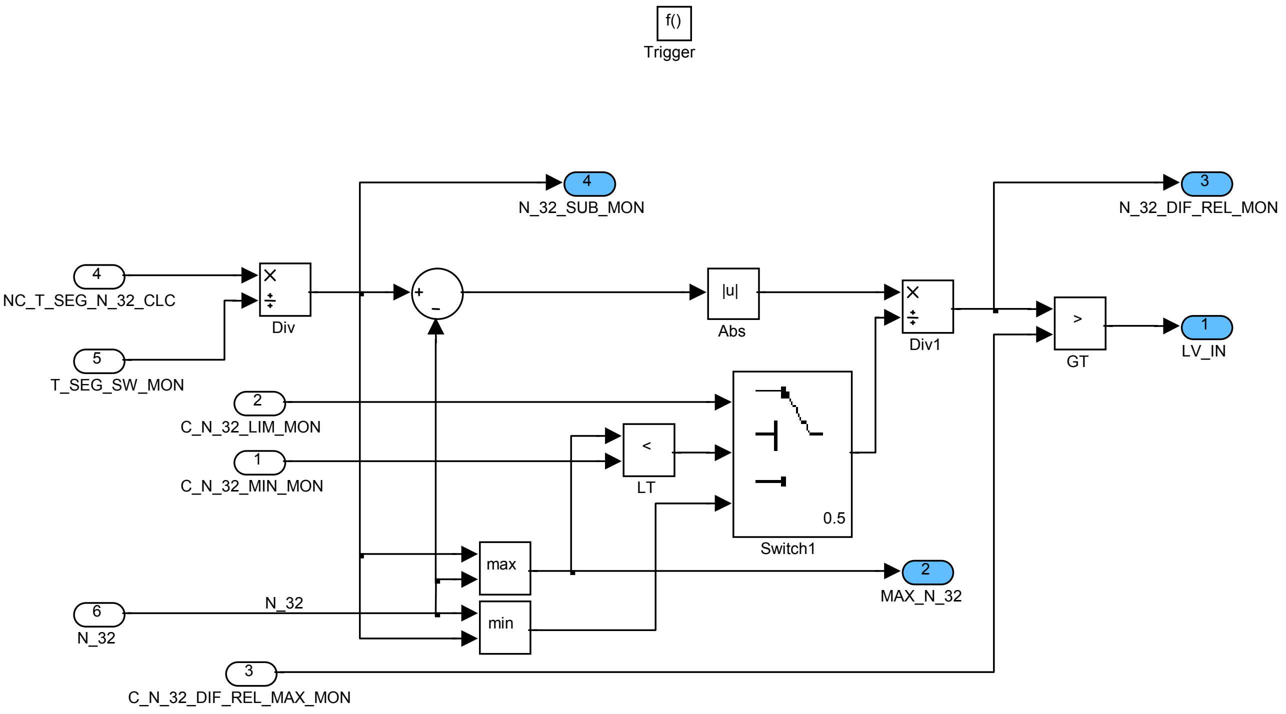


Figure 4 ECM2_DTSYSN/ N_32_MONITORING

11.15 Monitoring of pedal value signals(PVS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_PVS_MON	O/V	0...1H	0...1	1	-
Logical variable for different PVS diagnosis between L1 and L2					
LV_ERR_PVS_RATIO_MON	O/V	0...1H	0...1	1	-
Logical variable for ratio-error between PVS channel 1 and 2					
PV_AV_MON	O/V	0...FFH	0...99.609375	0.390625	%
Level 2 degree of activation of the accelerator pedal					
V_PVS_1_MON	O/V	0...3FFH	0...4.995117	0.00488281	V
Pedal value sensor 1 raw acquisition					
V_PVS_2_MON	O/V	0...3FFH	0...4.995117	0.00488281	V
Pedal value sensor 1 raw acquisition					
ABC_PVS_RATIO_MON	O/V	0...FFH	0...255	1	-
Anti bounce counter PVS ratio error					
PV_AV_1_MON	V	0...FFH	0...99.609375	0.390625	%
Degree of activation of the accelerator pedal (channel 1)					
PV_AV_2_MON	V	0...FFH	0...99.609375	0.390625	%
Degree of activation of the accelerator pedal (channel 2)					
FAC_V_SLOP_PVS_MON	V	0...FFH	0...3.984375	0.015625	-
Factor for correction of slop of channel 2 in ratio check					

Input data:

LV_BRAKE_MON	LV_ERR_VCC_PVS_1_MON	LV_ERR_VCC_PVS_2_MON	LV_ERR_VCC_PVS_MON
LV_ERR_PVS_1	LV_ERR_PVS_2	LV_ERR_PVS_RATIO	LV_IGK_MON
N_32_MON			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_PVS_RATIO_MON	1	0...FFH	0...255	1	-
Anti bounce counter increment (additive value in case of error)					
C_ABC_MAX_PVS_RATIO_MON	1	1...FFH	1...255	1	-
Value at which ratio error is recognized, when reached					
C_N_MIN_PVS_DIAG_MON	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for PVS diagnosis					
C_PV_LIH_MAX_MON	1	0...FFH	0...99.609375	0.390625	%
Max. pedal position on case of any PVS error					
C_V_PVS_IS_THD_1_MON	1	0...3FFH	0...4.995117	0.00488281	V
Threshold for idle speed PVS channel 1					
C_V_PVS_IS_THD_2_MON	1	0...3FFH	0...4.995117	0.00488281	V
Threshold for idle speed PVS channel 2					
C_V_PVS_SLOP_1_MON	1	0...FFH	0...79.6875	0.3125	%/V
Slope of the PVS channel 1					
C_V_PVS_SLOP_2_MON	1	0...FFH	0...159.375	0.625	%/V
Slope of the PVS channel 2					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_V_THD_RATIO_CHK_PVS_MON	6	0...FFH	0...4.98046875	0.01953125	V
LDP_V_MIN_PV_MON_IP_V_PVS_MON	6	0...FFH	0...4.98046875	0.01953125	V
Maximum PV ratio error allowed					

11.15.1 ECM2_DTSYSPVS

Import actions:

ACTION_ECM3_Service3TaskPfm(IN<>)

ACTION_ECM3_Service4TaskPfm(IN<>)

ACTION_ECM3_Service5TaskPfm(IN<>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 2.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service3TaskPfm() and ACTION_ECM3_Service4TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)

ACTION_ECM3_ChkCpl(IN <>, IN <>)

ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

ACTION_INFR_GetVPVSMON(OUT<V_PVS_1_MON>, OUT<V_PVS_2_MON>)

Note: These actions is defined in IRS, it is responible for providing voltages from PVS channel 1 && 2

General information:

The pedal value diagnosis is made in the monitoring level in order to check the V_PVS input value of the torque monitoring function (level 2).

After the diagnosis, a resulting pedal value PV_AV_MON is used as input for the torque monitoring function.

A malfunction of level 1 will be assumed, if a ratio error of the two redundant pedal values is diagnosed after debouncing, yet no PVS failure was detected previously by the PVS diagnosis function of level 1.

The following errors were detected:

The difference between the values V_PVS_1_MON and V_PVS_2_MON is too large.

The diagnosis system of level 2 detects a PVS ratio fault and no fault is detected by the diagnosis of level 1.

Any PVS error-BIT of the PVS diagnosis in level 1 has been reset without permission.

To ensure that the PVS diagnosis of level 1 detects a fault earlier than the PVS ratio diagnosis of level 2, the fault detection time of the diagnosis function in level 2 has to be adjusted that it exceeds the fault detection time of level 1. Therefore, all level 1 PVS diagnosis thresholds, all constants of the debounce counters and the cycle time have to be taken into account.

Application Condition

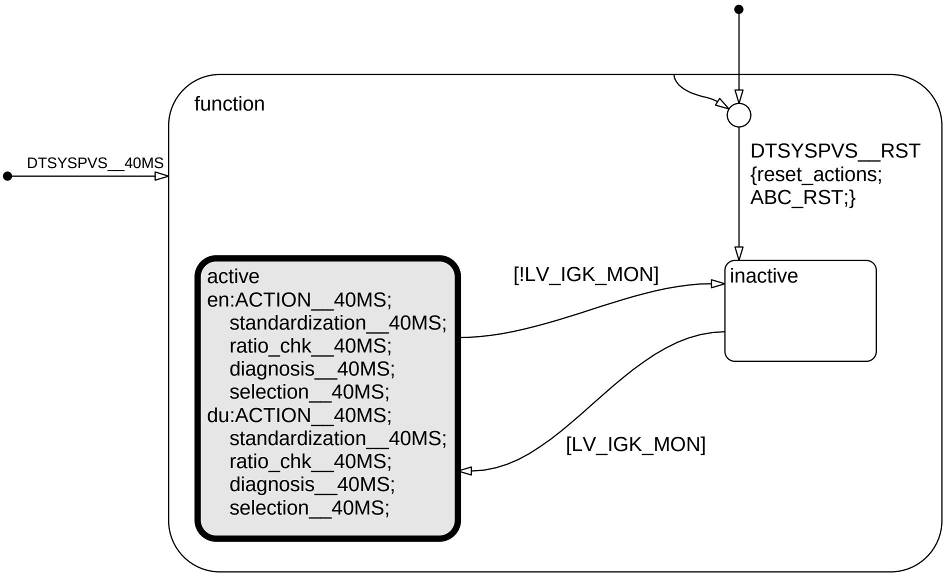
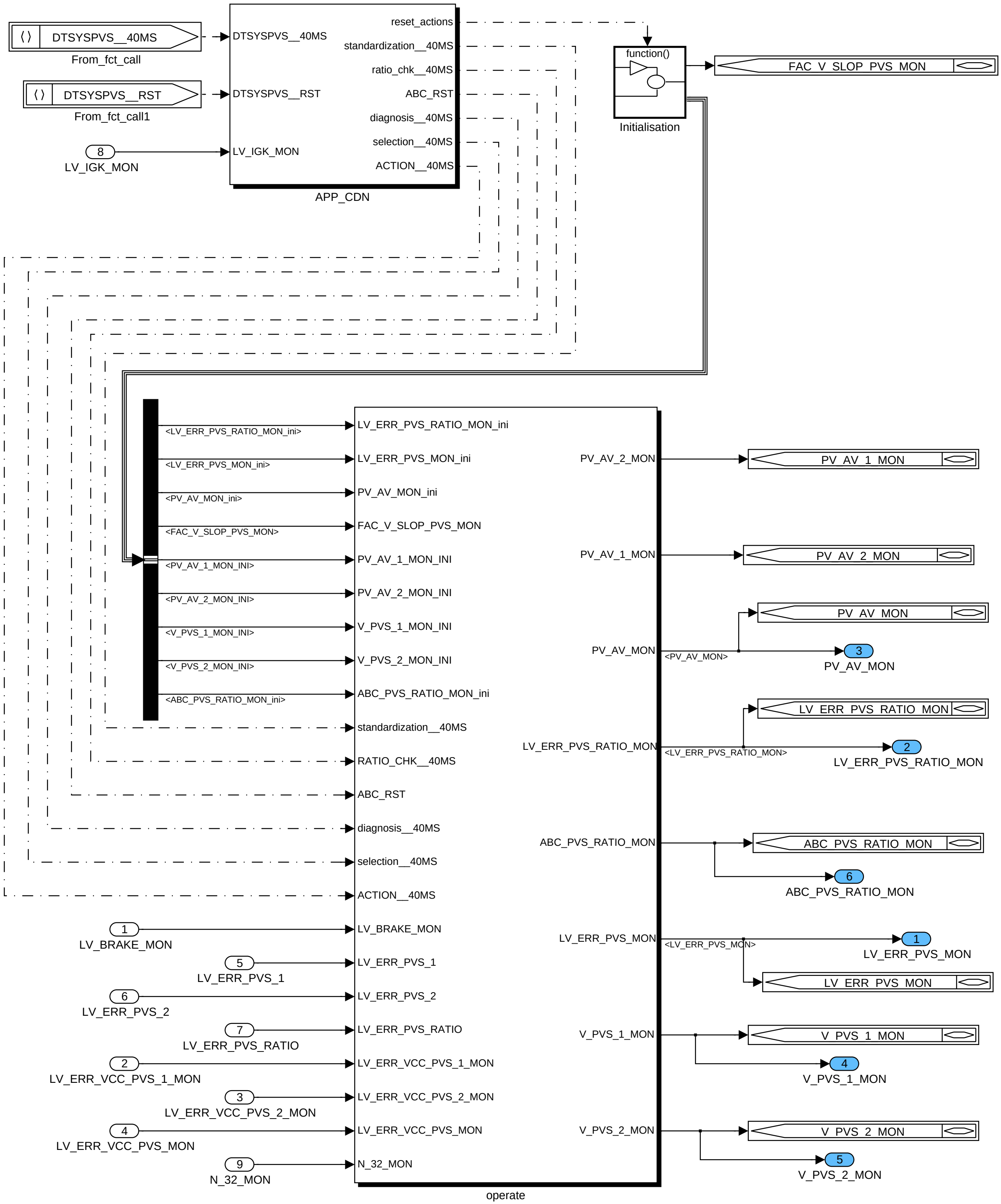


Figure 42 ECM2_DTSYSPVS/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 12−Jan−2007

Figure 43 ECM2_DTSYSPVS

11.15.1.1 SUBFUNCTION: Initialisation

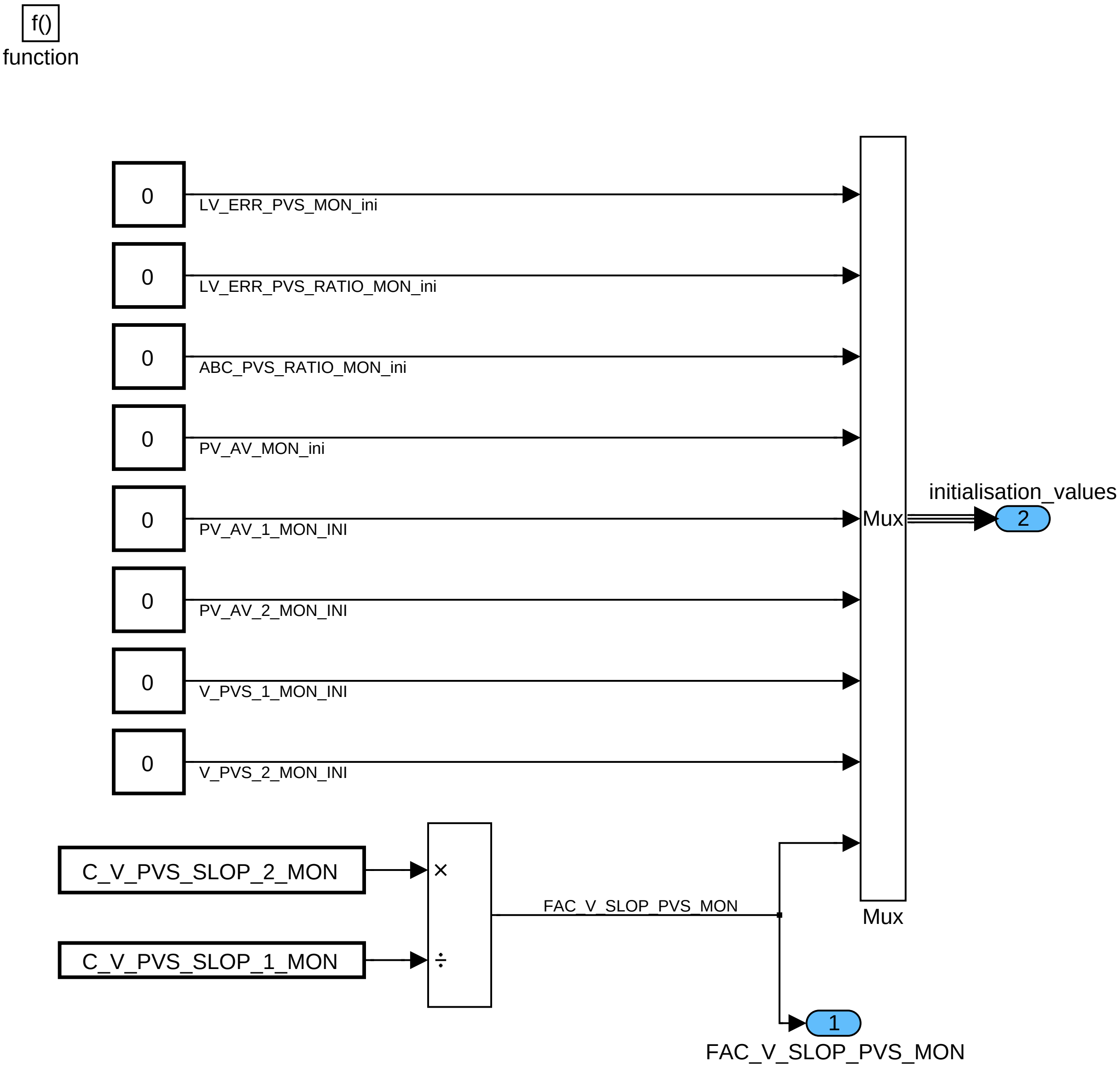


Figure 44 ECM2_DTSYSPVS/ Initialisation

11.15.1.2 SUBFUNCTION: operate

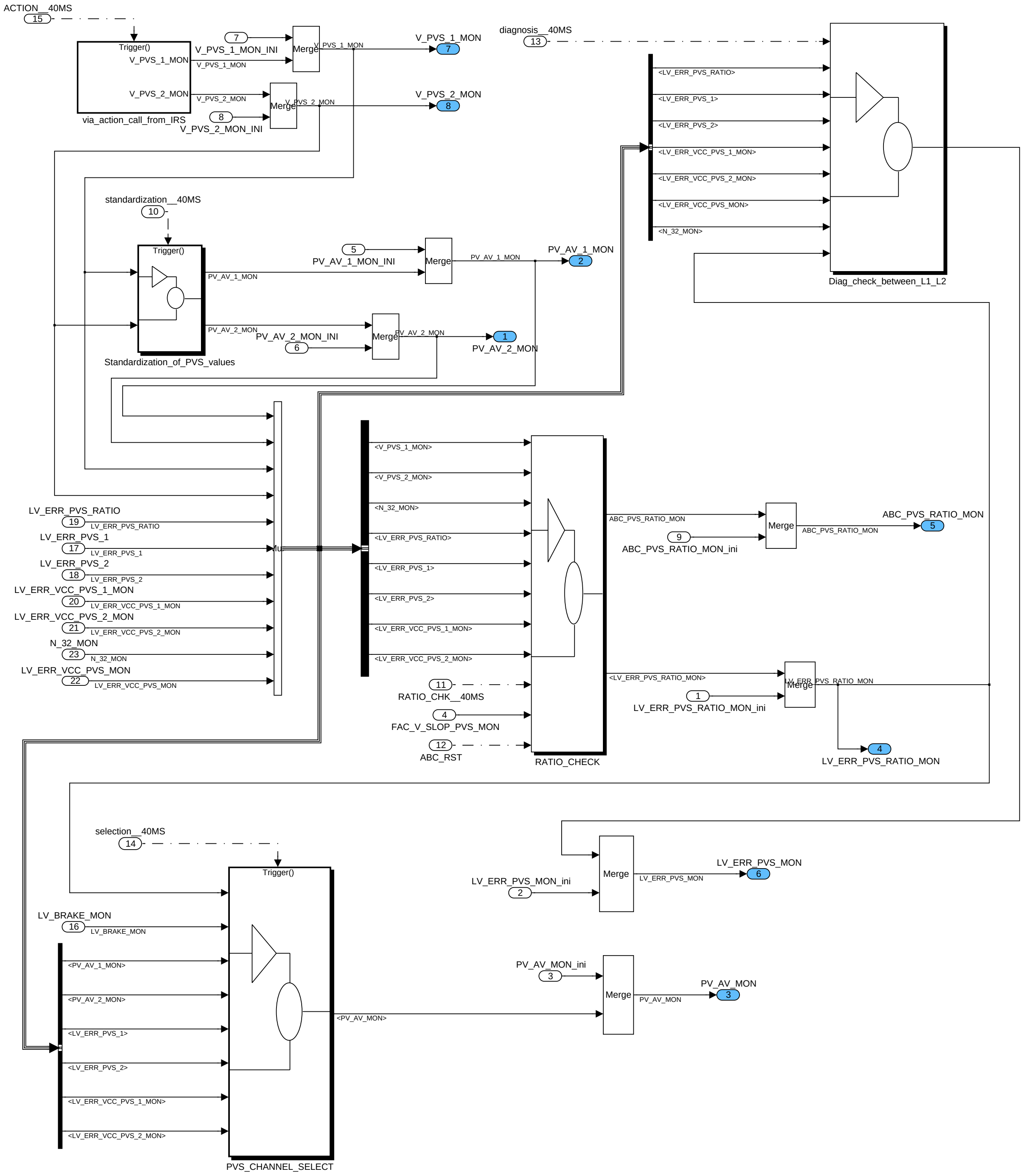


Figure 45 ECM2_DTSYSPVS/ operate

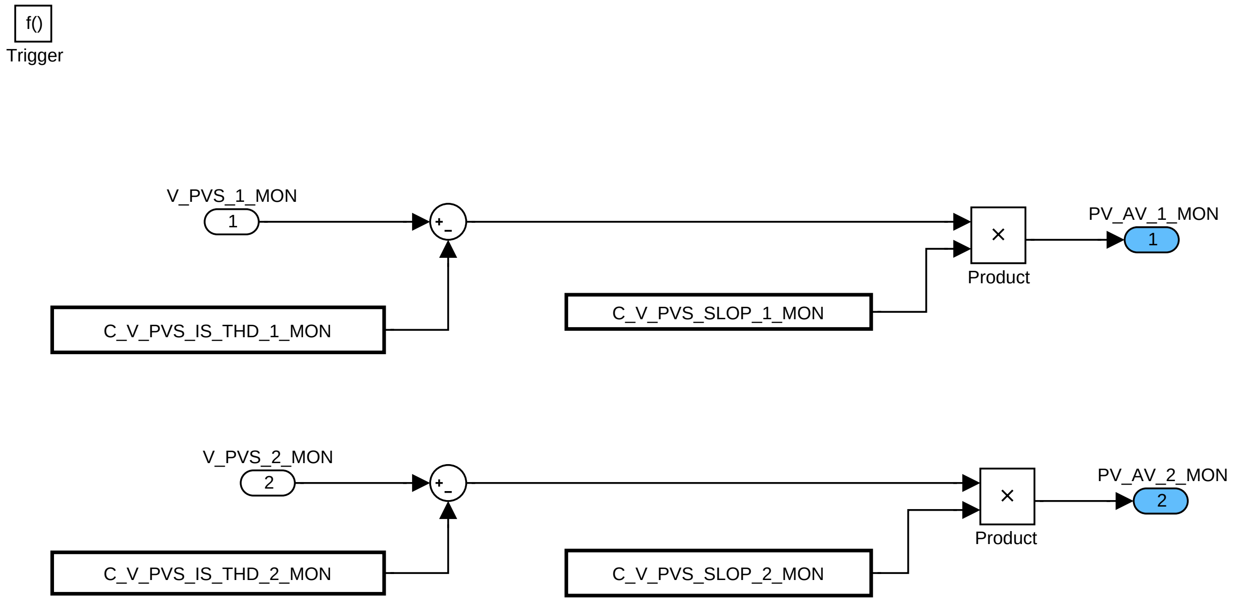


Figure 46 ECM2_DTSYSPVS/ operate/ Standardization_of_PVS_values

ECM2_DTSYSPVS/OPERATE/RATIO_CHECK

In case of an error free system (no siganl range error detected), the difference between V_PVS_1_MON and V_PVS_2_MON is not allowed to be greater than IP_V_THD_RATIO_CHK_PVS_MON.

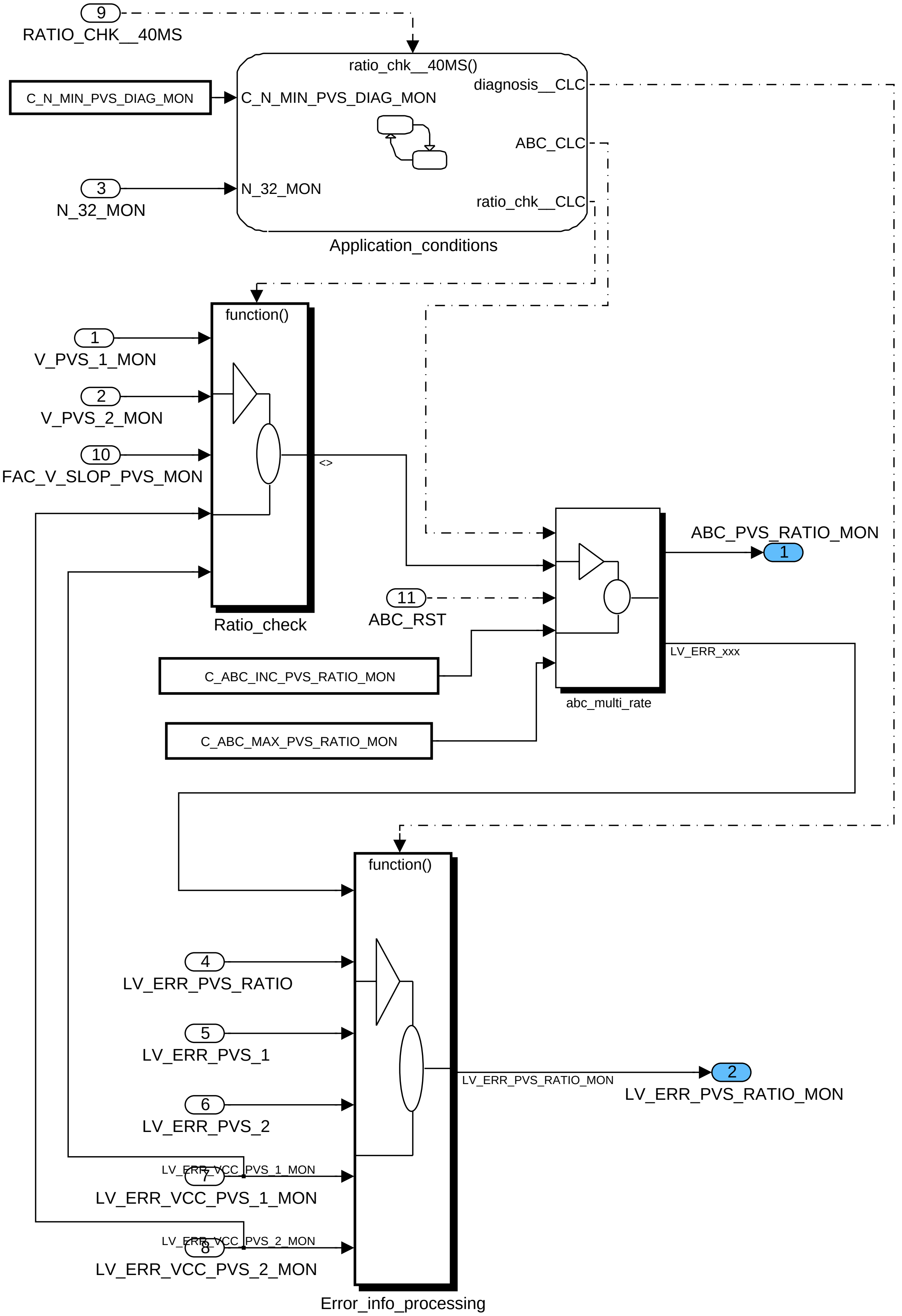

9
RATIO_CHK__40MS
ratio_chk__40MS()
C_N_MIN_PVS_DIAG_MON
C_N_MIN_PVS_DIAG_MON
diagnosis__CLC
ABC_CLC
3
N_32_MON
N_32_MON
ratio_chk__CLC
Application_conditions
function()
1
V_PVS_1_MON
2
V_PVS_2_MON
10
FAC_V_SLOP_PVS_MON
<>
Ratio_check
ABC_PVS_RATIO_MON
1
11
ABC_RST
LV_ERR_xxx
C_ABC_INC_PVS_RATIO_MON
abc_multi_rate
C_ABC_MAX_PVS_RATIO_MON
function()
4
LV_ERR_PVS_RATIO
5
LV_ERR_PVS_1
LV_ERR_PVS_RATIO_MON
2
LV_ERR_PVS_RATIO_MON
6
LV_ERR_PVS_2
LV_ERR_VCC_PVS_1_MON
7
LV_ERR_VCC_PVS_1_MON
LV_ERR_VCC_PVS_2_MON
8
LV_ERR_VCC_PVS_2_MON
Error_info_processing

Figure 47 ECM2_DTSYSPVS/ operate/ RATIO_CHECK

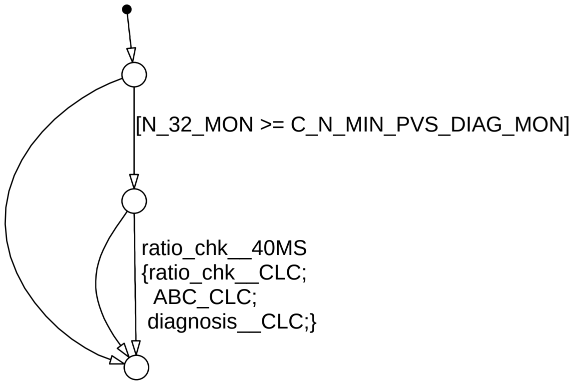


Figure 48 ECM2_DTSYSPVS/ operate/ RATIO_CHECK/ Application_conditions

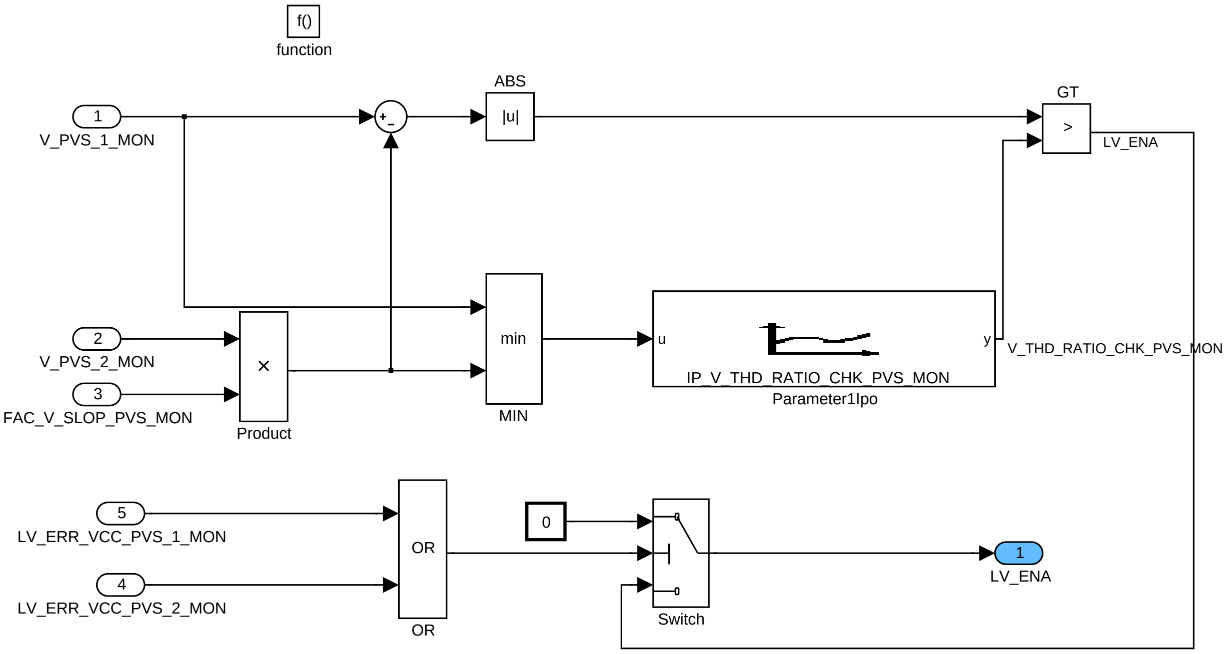


Figure 49 ECM2_DTSYSPVS/ operate/ RATIO_CHECK/ Ratio_check

ECM2_DTSYSPVS/OPERATE/RATIO_CHECK/ERROR_INFO_PROCESSING

The Bits LV_ERR_VCC_PVS_1_MON and LV_ERR_VCC_PVS_2_MON are reversible like the correspndent failure bits from level 1. That means, the output LV_ERR_PVS_RATIO_MON has to be reversible also depending of these two error bits.

All PVS error information of the function level is considered. This allows the different process levels to be based on the same error information:

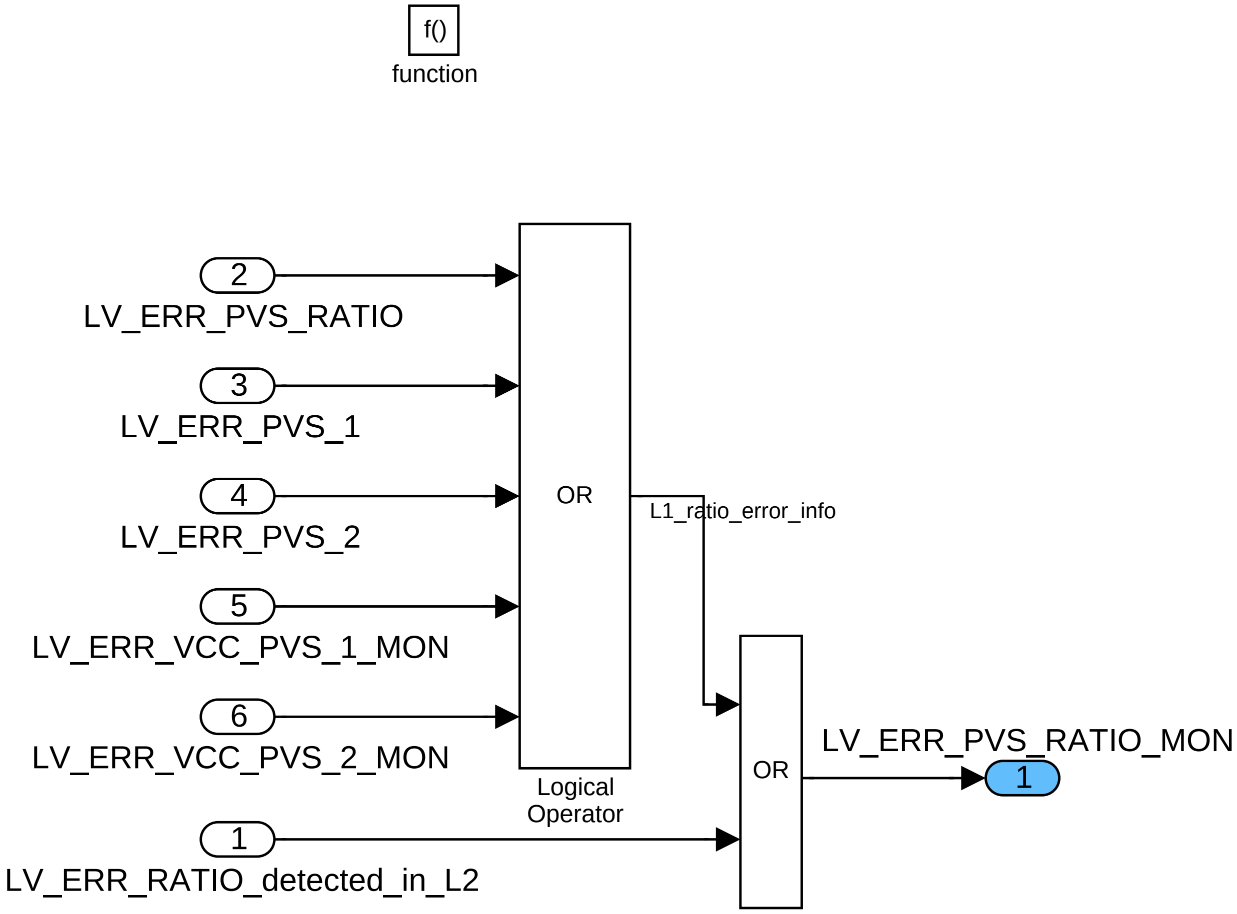


Figure 50 ECM2_DTSYSPVS/ operate/ RATIO_CHECK/ Error_info_processing

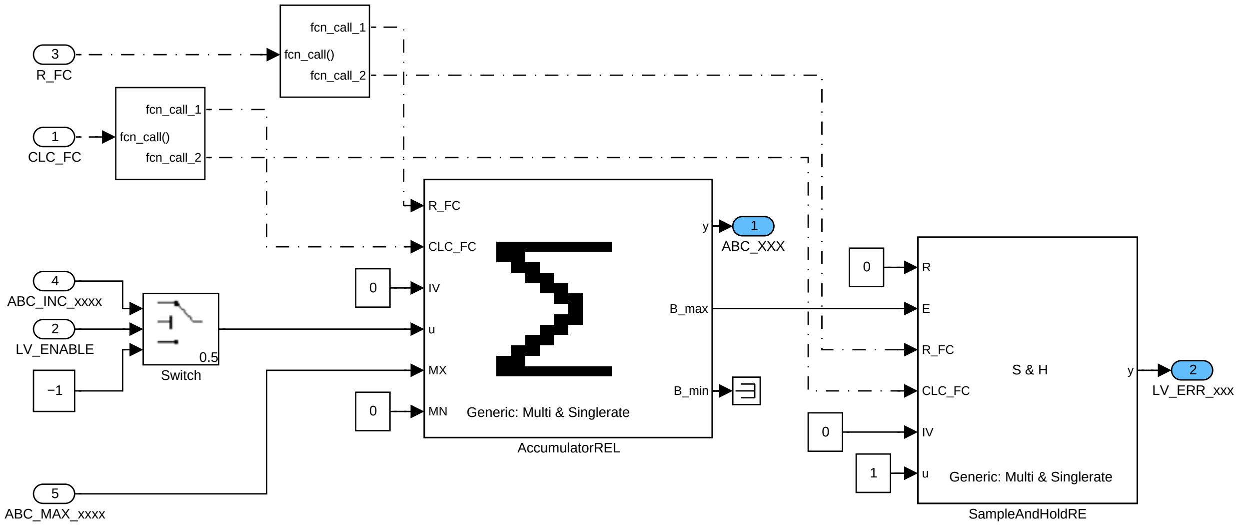


Figure 51 ECM2_DTSYSPVS/ operate/ RATIO_CHECK/ abc_multi_rate

ECM2_DTSYSPVS/OPERATE/DIAG_CHECK_BETWEEN_L1_L2

If any PVS ratio error is detected in level 2 and no PVS error is detected in level 1, a fault of the PVS diagnosis in level 1 will be assumed.

There is also a check, whether an error-BIT of the PVS diagnosis in level 1 was reset without permission. This is done by using the result of LV_ERR_PVS_RATIO_MON of the last recurrence.

LV_ERR_VCC_PVS_MON is assumed to be a PVS supply voltage error flag of a level 2 diagnosis in case of any PVS supply voltage error.

LV_ERR_VCC_PVS_1_MON and LV_ERR_VCC_PVS_2_MON are error flags that result from Level 1 diagnosis of PVS-1 and PVS-2 supply voltage and are assigned to the LV_ERR_VCC_PVS_1_MON and LV_ERR_VCC_PVS_1_MON respectively in the application incidences of process monitoring.

If an PVS?monitoring error is detected and the error bit LV_ERR_PVS_MON is set, then this bit muts not be reversible.

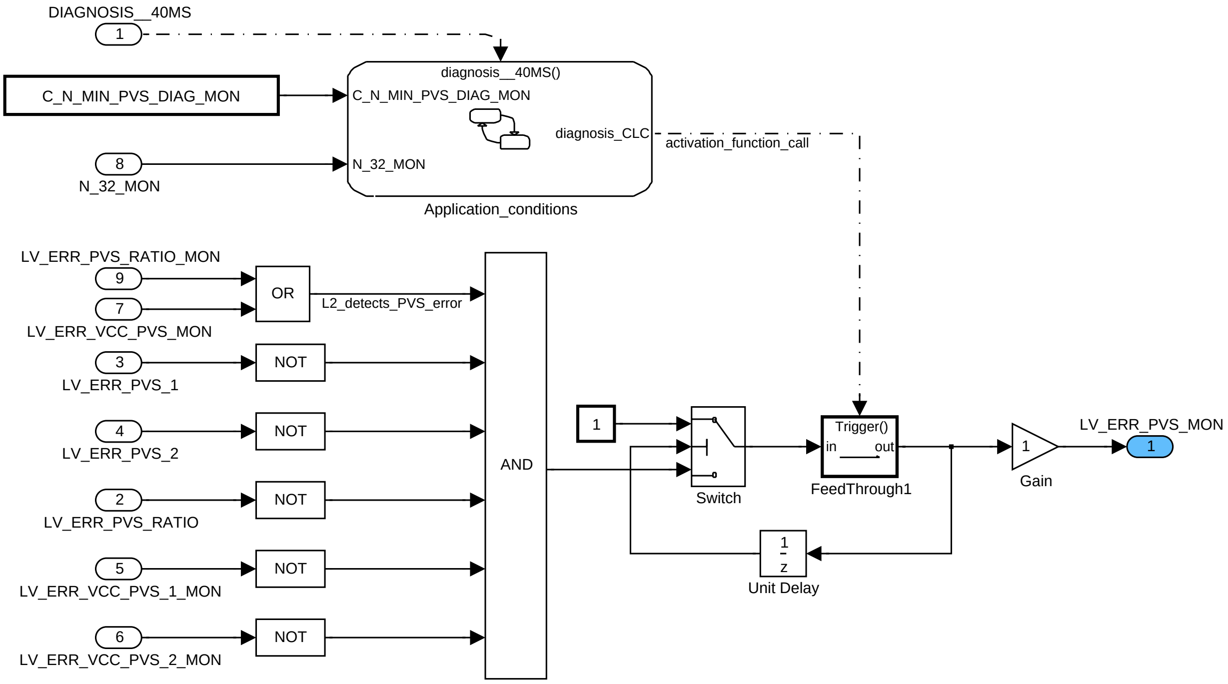


Figure 52 ECM2_DTSYSPVS/ operate/ Diag_check_between_L1_L2

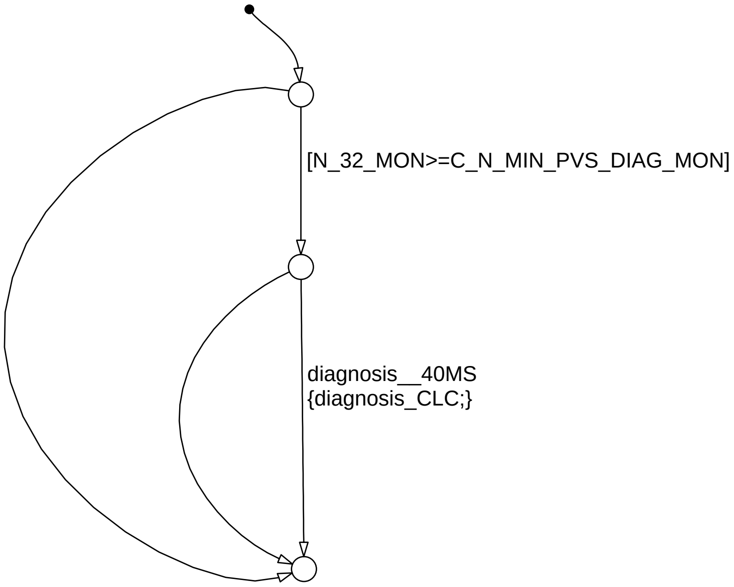


Figure 53 ECM2_DTSYSPVS/ operate/ Diag_check_between_L1_L2/ Application_conditions

ECM2_DTSYSPVS/OPERATE/PVS_CHANNEL_SELECT

Depending on the plausibility, the degree of pedal activation PV_AV_MON is defined by the signals PV_AV_1_MON and PV_AV_2_MON, as follows:

Error Level	Error bits set	Analysis, error handling
0	-	PV_AV_MON = MIN(PV_AV_1_MON,PV_AV_2_MON)
1	LV_ERR_PVS_RATIO_MON	PV_AV_MON = MIN(PV_AV_1_MON, PV_AV_2_MON) (minimum)
2	LV_ERR_PVS_1 **OR** LV_ERR_VCC_PVS_1_MON	PV_AV_MON = PV_AV_2_MON (selection)
3	LV_ERR_PVS_2 **OR** LV_ERR_VCC_PVS_2_MON	PV_AV_MON = PV_AV_1_MON (selection)

4	(LV_ERR_PVS_1 OR LV_ERR_VCC_PVS_1_MON) **AND** (LV_ERR_PVS_2 OR LV_ERR_VCC_PVS_2_MON)	PV_AV_MON = 0 driver demand not recoganizable

If any PVS error-BIT is set:

PV_AV_MON will be limited to C_PV_LIH_MAX_MON.

Use of the brake as a redundant driver demand:

IF (LV_BRAKE_MON = 1)

THEN

PV_AV_MON = 0 (same as Error level 4)

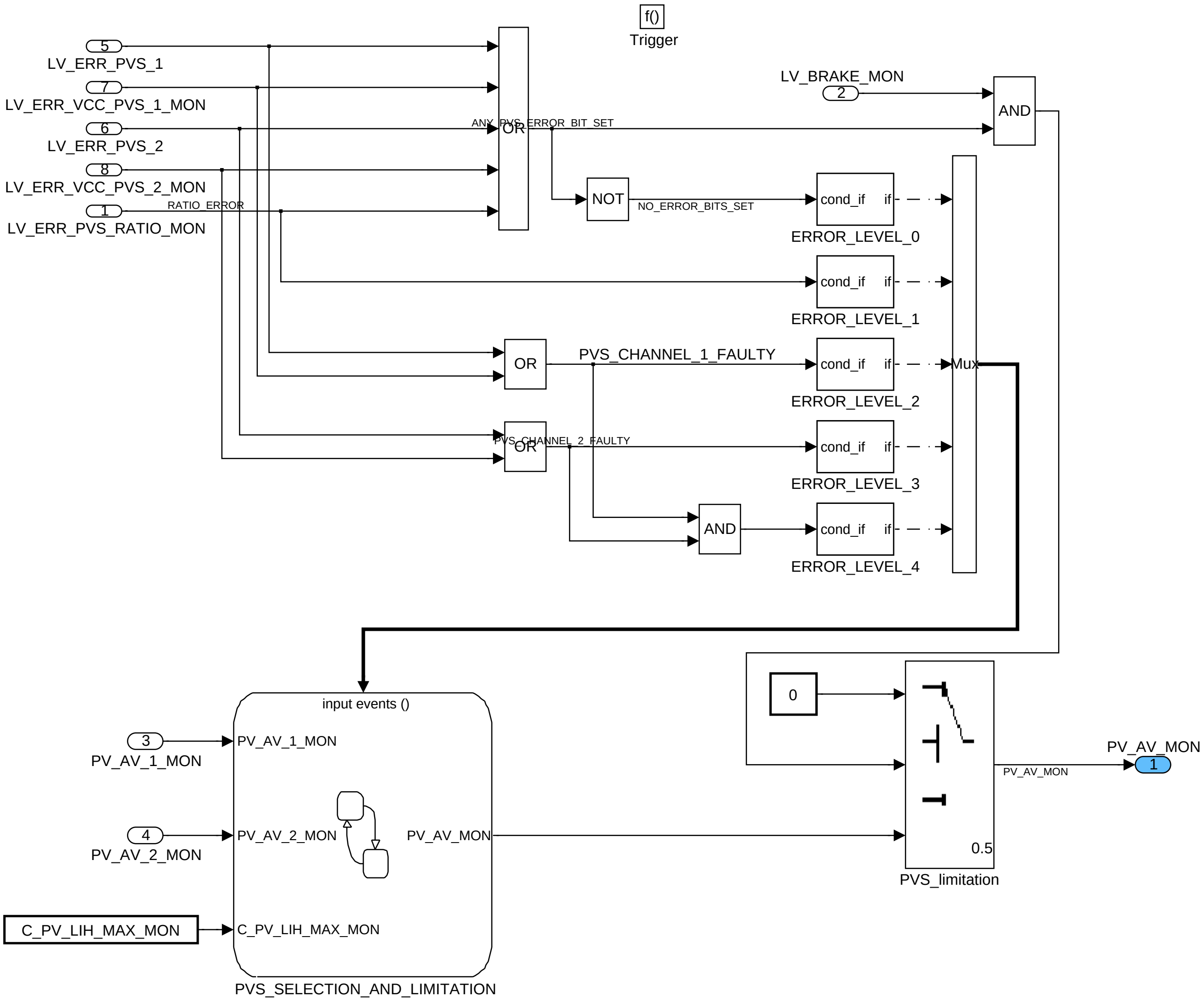


Figure 54 ECM2_DTSYSPVS/ operate/ PVS_CHANNEL_SELECT

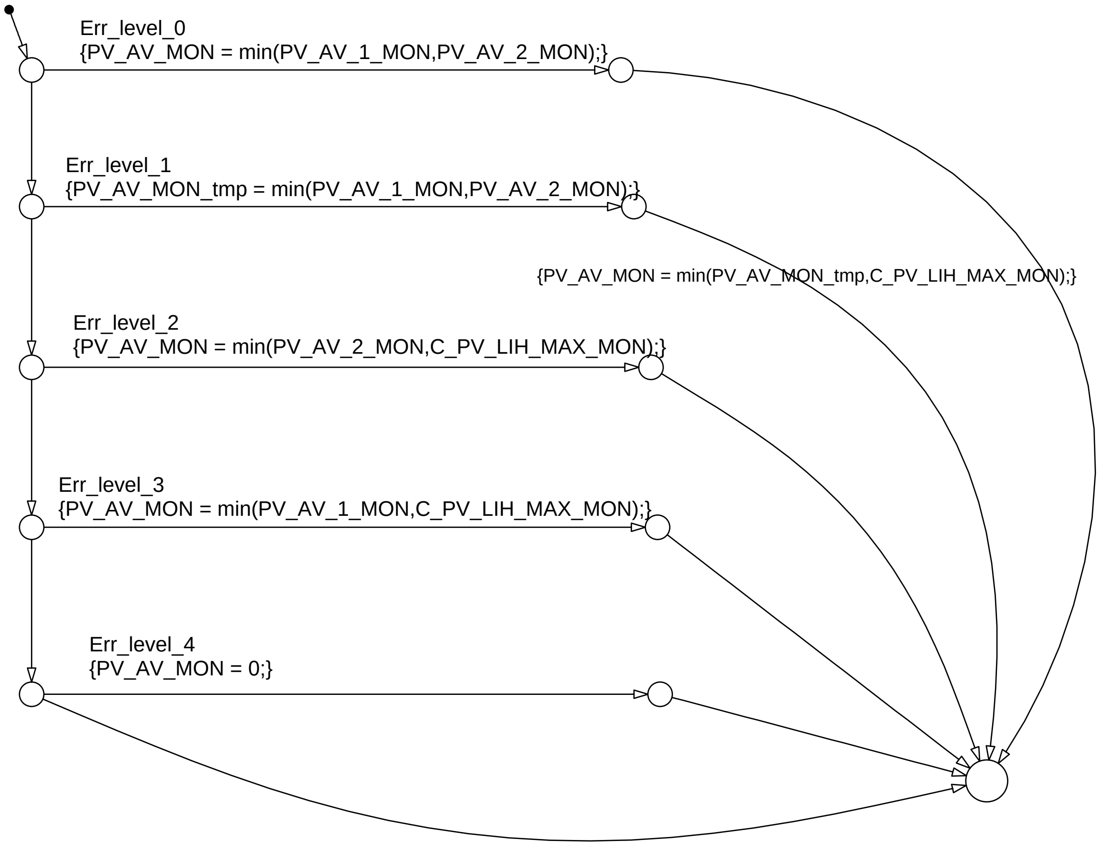


Figure 55 ECM2_DTSYSPVS/ operate/ PVS_CHANNEL_SELECT/ PVS_SELECTION_AND_LIMITATION

11.16 Monitoring of mass fuel flow (MFF_MON)

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
ABC_LAMB_DIF_NEG_MON	V	0... FFH	0... 255	1	[-]
Anti bounce counter LAMB_DIF too small					
ABC_LAMB_DIF_POS_MON	V	0... FFH	0... 255	1	[-]
Anti bounce counter LAMB_DIF too high					
ABC_MFF_NOT_AFS_MON	V	0... FFH	0... 255	1	[-]
Anti bounce for fault in MFF signal (not AFS-mode)					
ABC_TPS_RATIO_MON	V	0... FFH	0... 255	1	[-]
Anti bounce counter TPS ratio diagnosis error					
LAMB_DIF_FIL_MON	V	8000... 7FFFH	-32... 31.9990234375	976.563e-6	[-]
Deviation between actual and modeled Lambda (low pass filtered)					
LAMB_DIF_NEG_MON	V	8000... 7FFFH	-32... 31.9990234375	976.563e-6	[-]
allowed negative LAMB_DIF threshold					
LAMB_DIF_POS_MON	V	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
allowed positive LAMB_DIF threshold					
LV_ERR_MFF_MON	O/V	0... 1H	0... 1	1	[-]
Logical variable for error in mass fuel flow signal					
LV_ERR_TPS_MON	O/V	0... 1H	0... 1	1	[-]
Logical variable for different TPS diagnosis between L1 and L2					
LV_SWI_AFS_MFF_MAX_MON	V	0... 1H	0... 1	1	[-]
allowed maximum number of transitions to AFS mode exceeded					
LV_SWI_AFS_MFF_MON	O/V	0... 1H	0... 1	1	[-]
Flag for switching to AFS mode					
T_ACT_SWI_AFS_MFF_MON	V	0... FFFFH	0... 2621.4	0.04	[s]
active time for switch to AFS mode					
V_TPS_AD_1_MON	V	0... 3FFH	0... 4.9951171875	4.88281e-3	[V]
Adapted TPS signal channel 1					
V_TPS_AD_2_MON	V	0... 3FFH	0... 4.9951171875	4.88281e-3	[V]
Adapted TPS signal channel 2					

Input Data:

LV_TQI_MON_ACT_MON	OPM_AV_MON	LAMB_DIF_MON	LAMB_MDL_MON
LAMB_LS_UP [NC_CBK_EX_NR]	STATE_ERR_IV	LV_ERR_MAP	LV_ERR_FUP
LV_LAMB_LS_UP_VLD_MON [NC_CBK_EX_NR]	V_TPS_1_MON	V_TPS_2_MON	V_TPS_AD_EL_BOL_1
LV_ERR_TPS_RATIO	LV_ERR_TPS_1	LV_ERR_TPS_2	V_TPS_AD_EL_BOL_2
LV_ERR_TPS_MAF_1	LV_ERR_TPS_MAF_2	TQI_SP_MON	N_32_MON
MFF_LAMB_MON	MAF_MON	C_FAC_MFF_MAF_MON	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ABC_INC_LAMB_DIF_NEG_MON	1	0... FFH	0... 255	1	[-]
ABC increment in case of act. Lambda smaller than reference Lambda					

C_ABC_INC_LAMB_DIF_POS_MON	1	0... FFH	0... 255	1	[-]
ABC increment in case of act. Lambda greater than reference Lambda					
C_ABC_INC_MFF_NOT_AFS_MON	1	0... FFH	0... 255	1	[-]
Anti bounce counter increment in case of MFF error (not AFS-mode)					
C_ABC_INC_TPS_RATIO_MON	1	0... FFH	0... 255	1	[-]
Anti bounce counter increment (additive value in case of ratio diagnosis error)					
C_ABC_MAX_LAMB_DIF_NEG_AFS_MON	1	1... FFH	1... 255	1	[-]
Value at which negative LAMB_DIF error is recognised, when reached (AFS-mode)					
C_ABC_MAX_LAMB_DIF_NEG_S_MON	1	1... FFH	1... 255	1	[-]
Value at which negative LAMB_DIF error is recognised, when reached (not AFS-mode)					
C_ABC_MAX_LAMB_DIF_POS_MON	1	1... FFH	1... 255	1	[-]
Value at which positive LAMB_DIF error is recognised, when reached					
C_ABC_MAX_MFF_NOT_AFS_MON	1	1... FFH	1... 255	1	[-]
Value at which MFF error is recognised. when reached (not AFS-mode)					
C_ABC_MAX_TPS_RATIO_MON	1	1... FFH	1... 255	1	[-]
Value at which ratio diagnosis error is recognized. when reached					
C_CRLC_LAMB_DIF_MON	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
correlation constant for LAMB_DIF filter					
C_CTR_MAX_SWI_AFS_MFF_MON	1	0... FFH	0... 255	1	[-]
maximum number of transitions to AFS mode					
C_N_MIN_TPS_DIAG_MON	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for diagnosis of TPS ratio error					
C_T_ACT_SWI_AFS_MFF_MON	1	0... FFFFH	0... 2621.4	0.04	[s]
time out threshold for canceling AFS mode					
C_V_TPS_RATIO_HYS_MON	1	0... FFH	0... 4.98046875	0.0195313	[V]
Maximum TPS ratio error allowed					
IP_LAMB_DIF_ADD_1_MON	4*4	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDP_N_32_MON_IP_LAMB_DIF_MON	4	0... FFH	0... 8160	32	[rpm]
LDP_TQI_SP_MON_IP_LAMB_DIF_MON	4	0... FFH	0... 510	2	[Nm]
map for expanding the allowed LAMB_DIF threshold based on TQI_SP_MON and N_32_MON					
IP_LAMB_DIF_ADD_2_MON	4	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDP_MFF_MON_IP_LAMB_DIF_ADD_MON	4	0... FFFFH	0... 1389	0.0211948	[mg/stk]
map for expanding the allowed LAMB_DIF threshold based on MFF					
IP_LAMB_DIF_NEG_MON	8	0... FFFFH	-32... 31.9990234375	976.563e-6	[-]
LDPM_LAMB_MON_IP_LAMB_DIF_MON	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
map for allowed negative LAMB_DIF threshold					
IP_LAMB_DIF_POS_MON	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_LAMB_MON_IP_LAMB_DIF_MON	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
map for allowed positive LAMB_DIF threshold					
LC_LAMB_SEL_MON	1	0... 1H	0... 1	1	[-]
logical constant for selection of lambda value					
LC_RST_SWI_AFS_MFF_MON	1	0... 1H	0... 1	1	[-]
Switch to disable LV_SWI_AFS_MFF_MON					

General Information

Import actions:

ACTION_ECM3_Service18TaskPfm(IN<>)
ACTION_ECM3_Service19TaskPfm(IN<>)
ACTION_ECM3_Service20TaskPfm(IN<>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 2.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service18TaskPfm() and ACTION_ECM3_Service19TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

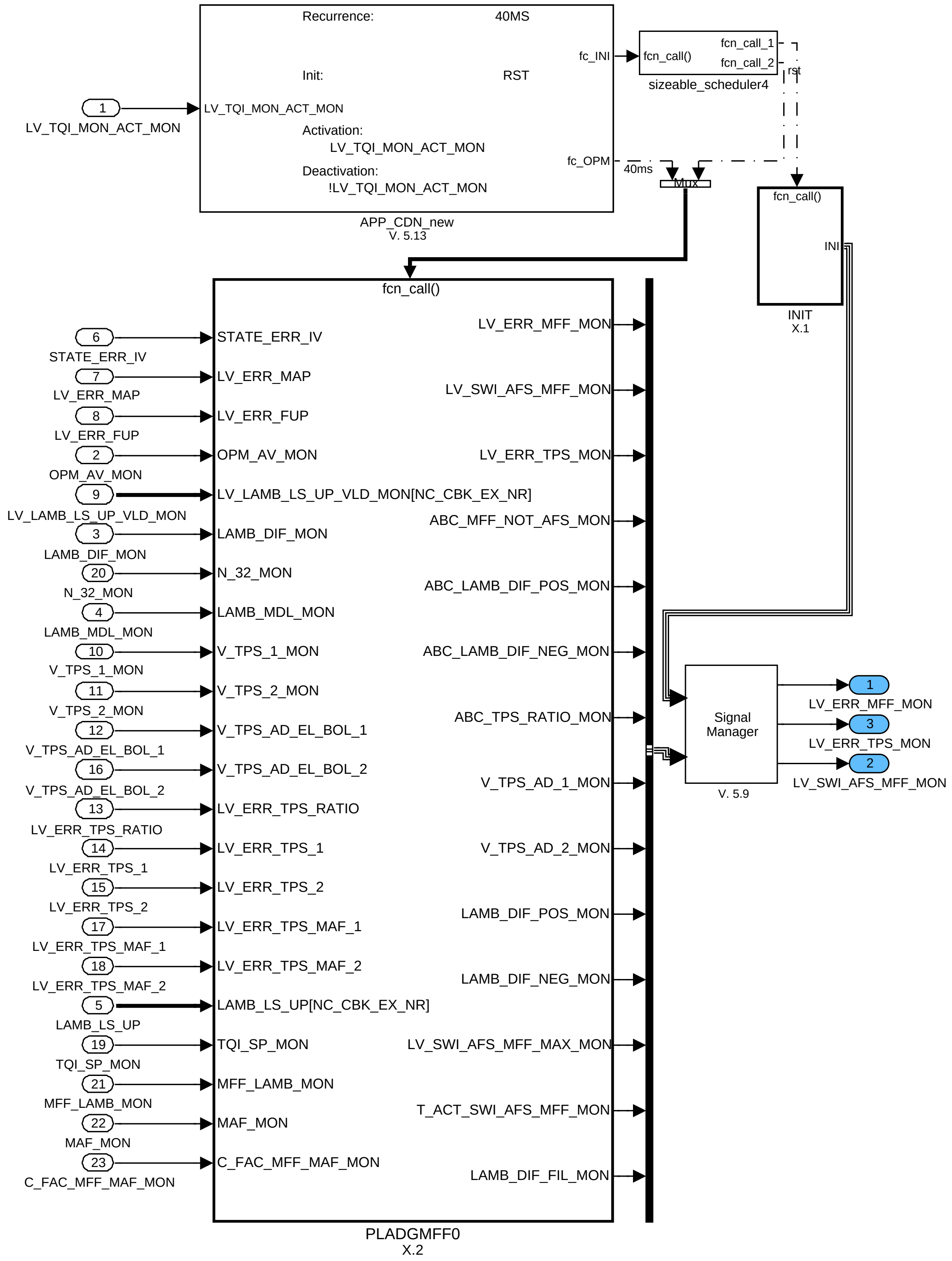
Note: These actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode O and history variables in modules of the Process Monitoring have to be checked by the cyclical RAM test.
General information:
The variable MFF_TQ_MON (represented via LAMB_DIF_MON) is needed for calculation of TQI_AV_MON. The modul checks the plausibility of MFF_TQ_MON (calculated in "Actual fuel mass flow"). Therefore LAMB_DIF_MON (actual Lambda reference Lambda) is checked against borders dependent on actual Lambda. The critical case will be, if LAMB_DIF_MON gets to small in nonAFS_mode (equivalent to acceleration). Additionally, LV_ERR_... coming from Level 1 are considered. If there is an implausible state during non AFSmode, OPM will be switched to AFS first.

An additional TPS ratio check is performed. This function checks if any TPS failure has been detected in level 1 when a TPS ratio error is recognized in level 2.

Hint!! Application Conditions - Initialisation: for condition see 'Application incidences of process monitoring'

Application Conditions

Initialization: RST
Recurrence: 40MS
Activation: LV_TQI_MON_ACT_MON
Deactivation: !LV_TQI_MON_ACT_MON



SDA_SRS / SDA V 5.0.4 20−Jun−2006

Figure 56:
Path: ECM2_PLADGMFF0

11.16.1 ECM2_PLADGMFF0/INIT

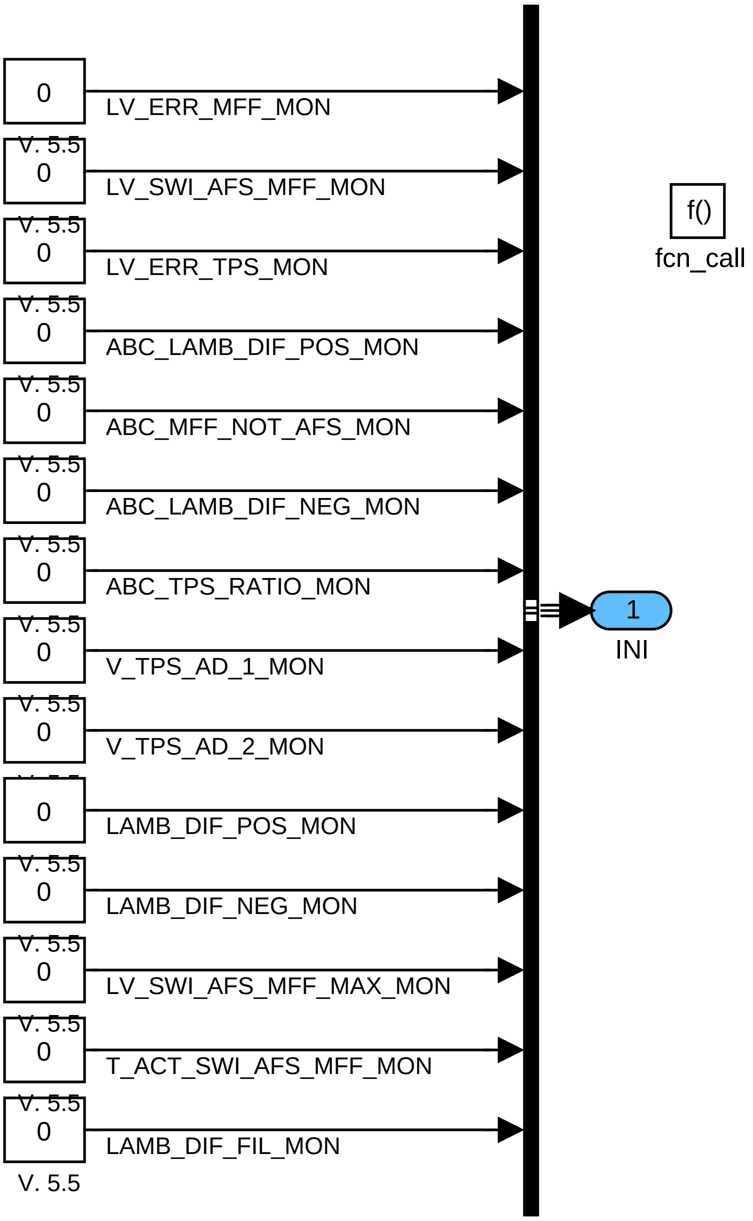


Figure 57:
Path: ECM2_PLADGMFF0/INIT

11.16.2 ECM2_PLADGMFF0/PLADGMFF0

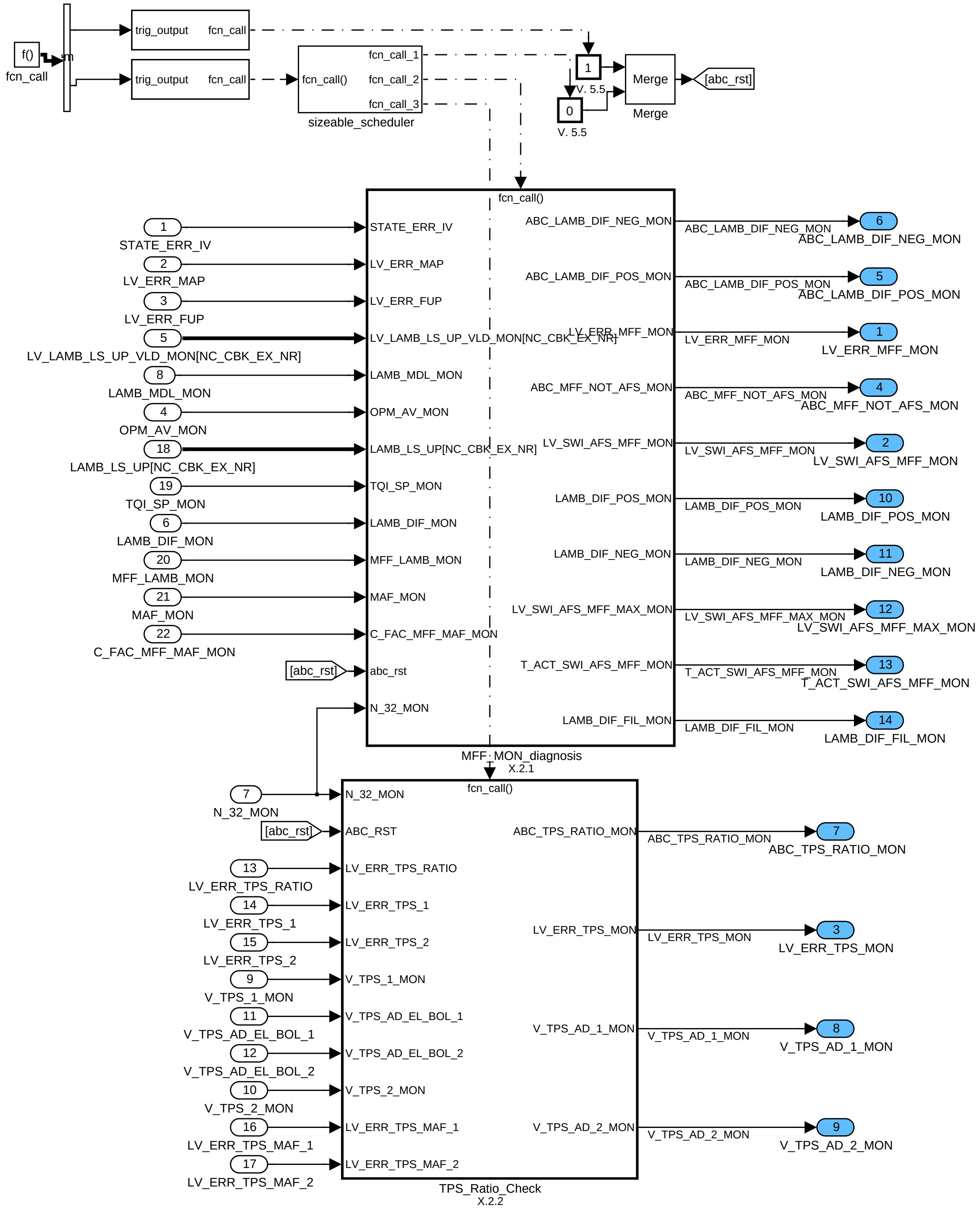


Figure 58:
Path: ECM2_PLADGMFF0/PLADGMFF0

11.16.2.1 ECM2_PLADGMFF0/PLADGMFF0/MFF_MON_DIAGNOSIS

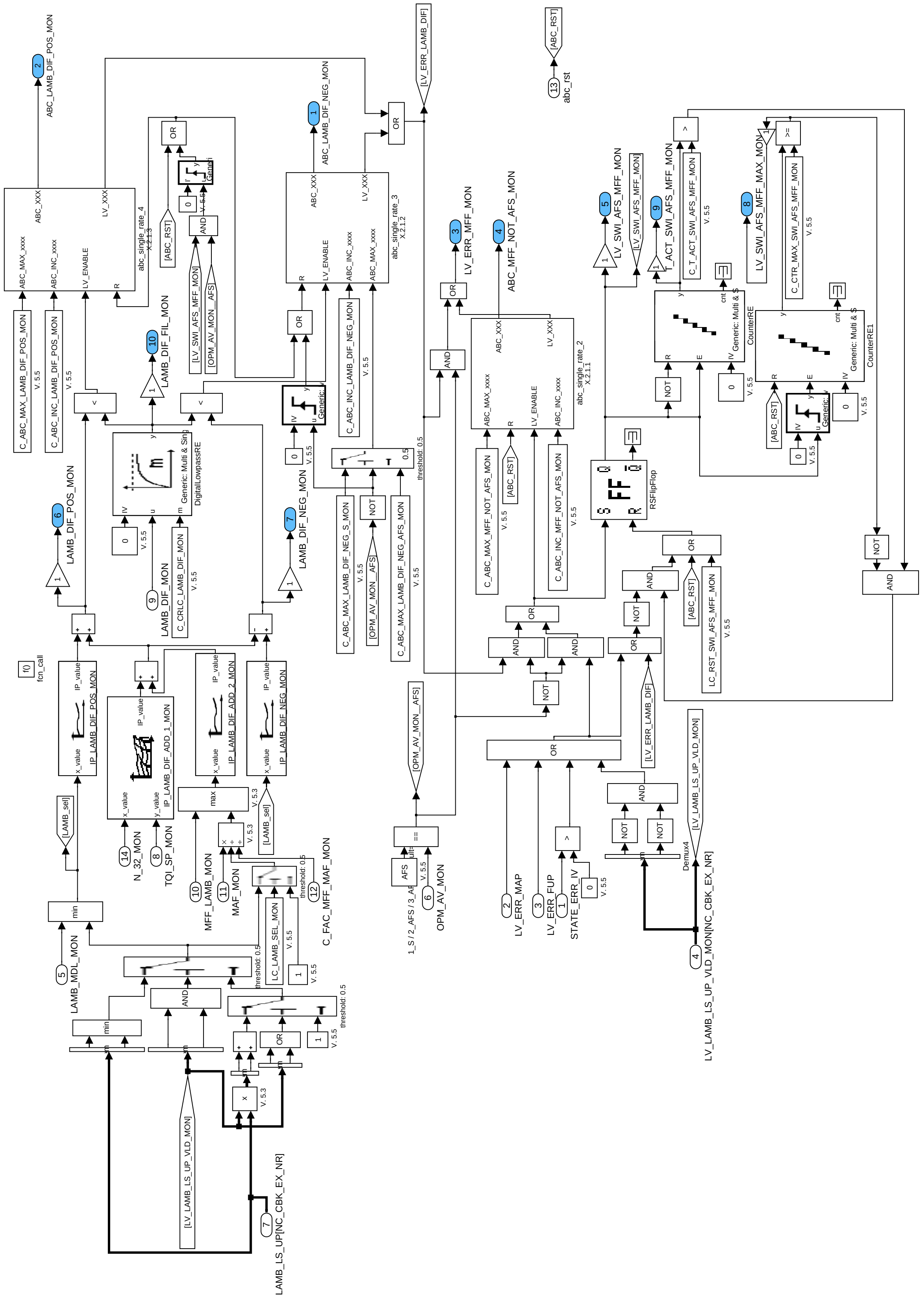


Figure 59:
Path: ECM2_PLADGMFF0/PLADGMFF0/MFF_MON_diagnosis

11.16.2.2 ECM2_PLADGMFF0/PLADGMFF0/TPS_RATIO_CHECK

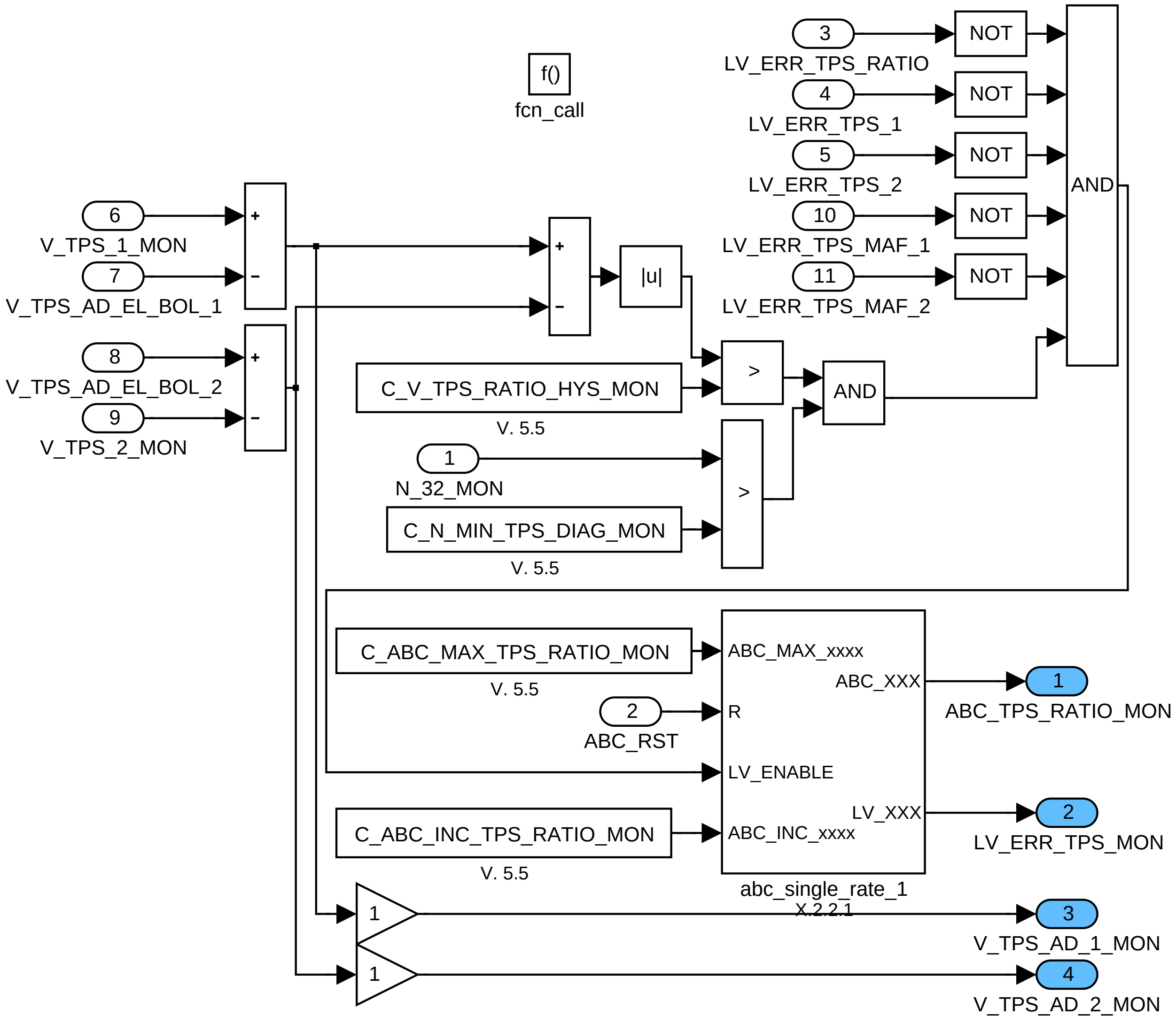


Figure 60:
Path: ECM2_PLADGMFF0/PLADGMFF0/TPS_Ratio_Check

11.17 Actual Lambda Deviation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_DIF_MON	O/V	8000...7FFFH	-32...31.9990234	9.76563E-4	-
Deviation between actual and modeled Lambda					
LAMB_MON	O/V	0...7FFFH	0...31.9990234	9.76563E-4	-
Modeled Lambda					
MAF_MON	O/V	0...FFH	0...1.389E+3	5.44705882	mg/stk
Mass air flow signal used monitoring level					
LAMB_MDL_MON	O/V	0...7FFFH	0...31.9990234	9.76563E-4	-
modeled lambda					
LAMB_DLY_MON	V	0...7FFFH	0...31.9990234	9.76563E-4	-
delayed LAMB_MON signal					
LAMB_MDL_REF_MON	V	0...7FFFH	0...31.9990234	9.76563E-4	-
Modeled lambda at reference conditions					
MAF_DLY_MON	V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Phase shifted MAF- signal					

Input data:

LV_TQI_MON_ACT_MON	OPM_AV_MON	MAF	MFF_LAMB_REF_MON
MFF_LAMB_MON	LAMB_LS_UP[NC_CBK_EX_NR]	T_DLY_SOI_LSL_POS[NC_CBK_EX_NR]	T1_LSL_UP_OPT[NC_CBK_EX_NR]
LV_LAMB_LS_UP_VLD_MON[NC_CBK_EX_NR]	MAF_EGR_NEUT_GAS	T_SEG_SW_MON	MAF_SCAV_EXT
EFF_SCC_MON			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DIF_LS_UP_LAMB_MON	1	8000...7FFFH	-32...31.9990234	9.76563E-4	-
difference offset added on LAMB_MON					
C_FAC_MFF_MAF_MON	1	0...FFFFH	0...15.9997559	2.44141E-4	-
Stoichiometric Constant for Lambda calculation					
C_FAC_T1_LSL_UP_MON	1	0...7FFFH	0...31.9990234	9.76563E-4	-
adaptation factor for first order time lag of sensor					
C_FAC_T_SEG_MON	1	0...FFH	0...15.9375	0.0625	-
Factor for multiple segment delay					
C_LAMB_REF_MON	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Adjustable lambda reference					
C_THD_EFF_SCC_MIN_MON	1	0...FFH	0...1.9921875	0.0078125	-
Threshold for minimal pattern efficiency					
C_THD_MAX_LAMB_MON	1	8000...7FFFH	-32...31.9990234	9.76563E-4	-
Maximum treshold for modeled lambda					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_THD_MAX_LAMB_SENS_MON	1	8000...7FFFH	-32...31.9990234	9.76563E-4	-
Maximum treshold for sensor synchronisation					
C_T_INJ_REAC_HOM_MON	1	0...FFH	0...10.2	0.04	s
maximal delay after reactivation at HOM					
C_T_INJ_REAC_MON	1	0...FFH	0...10.2	0.04	s
maximal delay after reactivation					
C_T_MIN_PUC_MON	1	0...FFH	0...10.2	0.04	s
Time for minimal PUC duration					
LC_LAMB_DLY_INI_MON	1	0...1H	0...1	1	-
Initialisation switch of Pade filter					
LC_PUC_LAMB_MON	1	0...1H	0...1	1	-
enable threshold- compare also out of PUC					
LC_T_MIN_PUC_MON	1	0...1H	0...1	1	-
Selector for minimal PUC duration					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_T_SAMPLE_MON	1	0...FFH	0...0.255	0.001	s
monitoring sample time					

11.17.1 ECM2_DTSYSLADIF

Import actions:

ACTION_ECM3_Service15TaskPfm(IN<>)
ACTION_ECM3_Service16TaskPfm(IN<>)
ACTION_ECM3_Service17TaskPfm(IN<>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 2.

ACTION_ECM3_McChkStack()

Note: These actions are defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service15TaskPfm() and ACTION_ECM3_Service16TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: These actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

General information:

The objective of this module is the determination of the deviation between the modelled Lambda and the sensor- signal during stratified combustion or the deviation between the modelled Lambda and the reference Lambda during homogeneous combustion.

Description:

Homogeneous combustion: The MFF_LAMB_MON signal has to be corrected by subtracting all enrichments (e.g. WUP, CAST, etc.) because after modelling the resulting Lambda by using MAF from Level1, a compare against the reference Lambda (normally 1) takes place. For this injection mode no Lambda sensor is necessary then.

Stratified combustion: The MFF_LAMB_MON signal has not to be corrected by enrichments, because the sensor signal is used here directly for the plausibility check. Due to physical transport delay of the exhaust gas the modelled Lambda signal has also to be delayed. This is done with a 'PADE- Filter'. Further the sensor behaviour itself has to be modelled by using a PT1- Filter. The resulting signal is compared with a mean value of both sensors or with the remaining, if one sensor is defect (in case of 2-bank engine).

Hint!! Application Conditions - Initialisation: for condition see 'Application incidences of process monitoring'

Application Condition

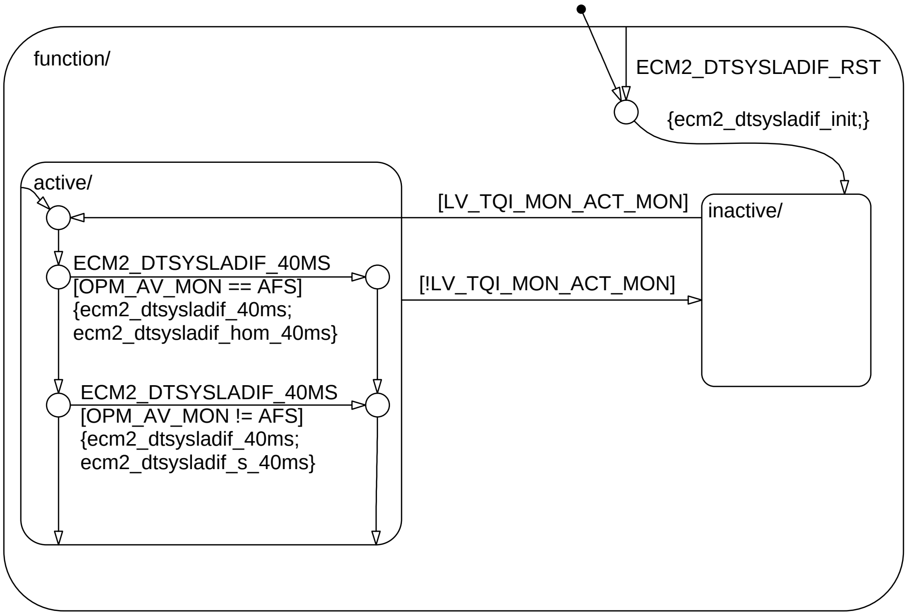
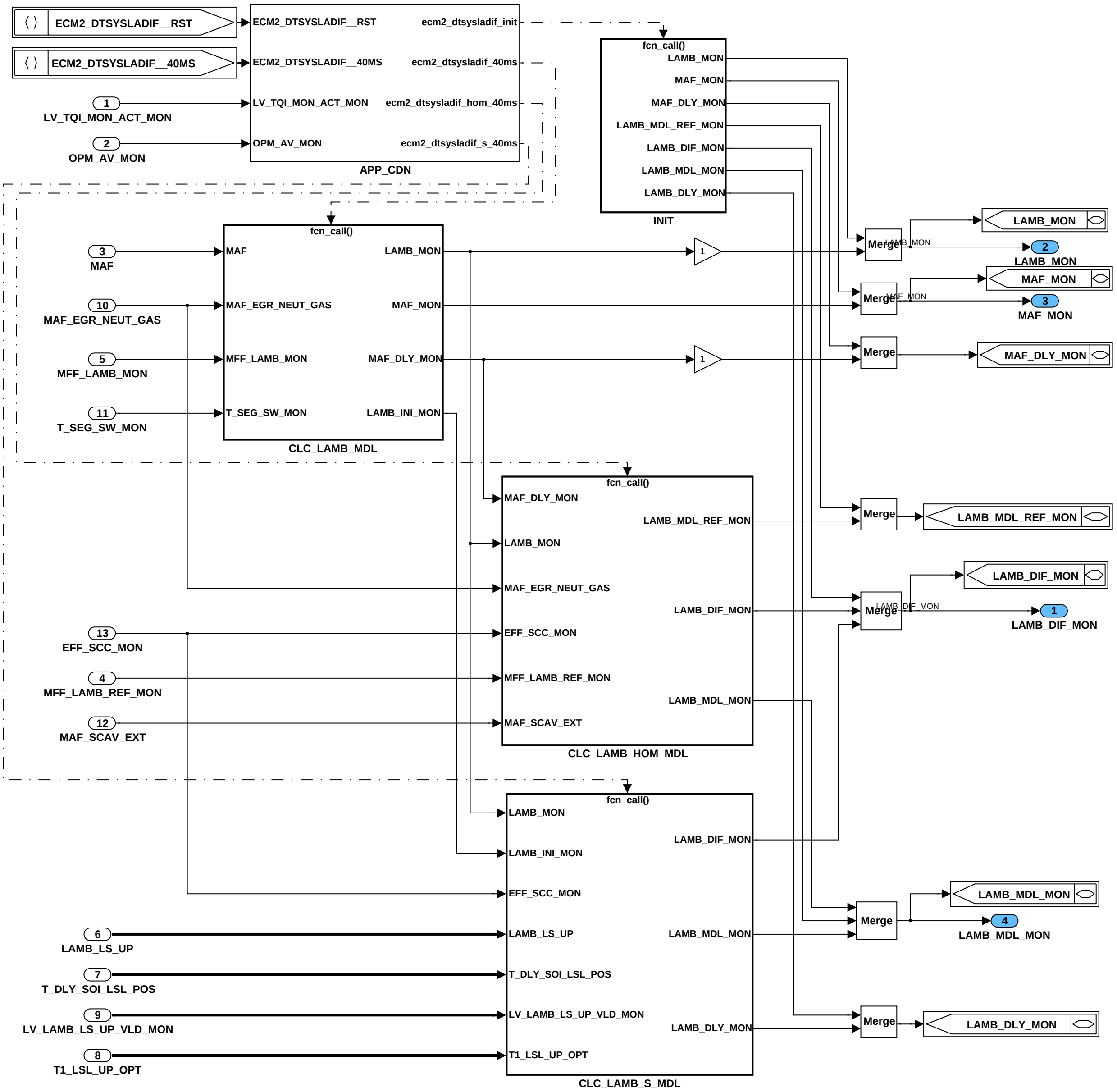


Figure 61 ECM2_DTSYSLADIF/ APP_CDN/ Chart



SDA_SRS / SDA 4.0 20−Jun−2006

Figure 62 ECM2_DTSYSLADIF

11.17.1.1 SUBFUNCTION: INIT

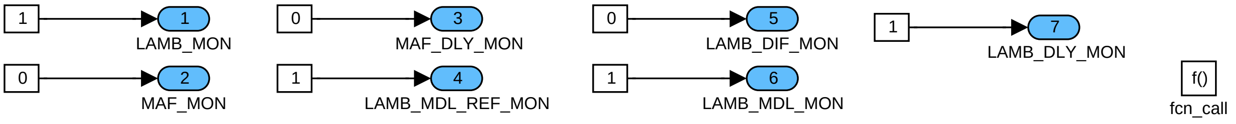


Figure 63 ECM2_DTSYSLADIF/ INIT

11.17.1.2 ECM2_DTSYSLADIF/CLC_LAMB_MDL

The Z-Block for the LAMB_MON delay should be initialized with 1 at every function activation.

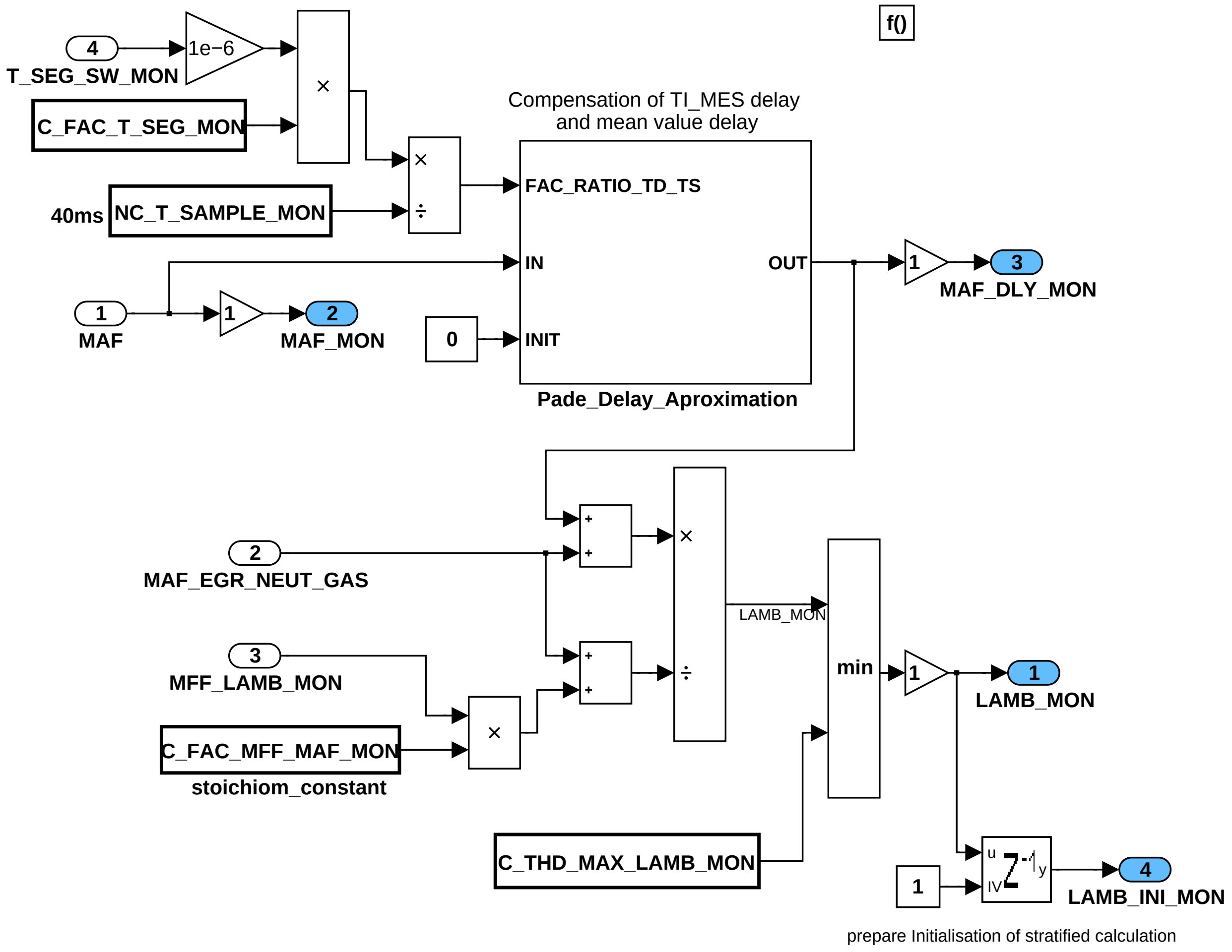


Figure 64 ECM2_DTSYSLADIF/ CLC_LAMB_MDL

11.17.1.3 ECM2_DTSYSLADIF/CLC_LAMB_HOM_MDL

Note: Once **at every activation of CLC_LAMB_HOM_MDL** the T_INJ_REAC_MON - timer should be initialized with C_T_INJ_REAC_HOM_MON.

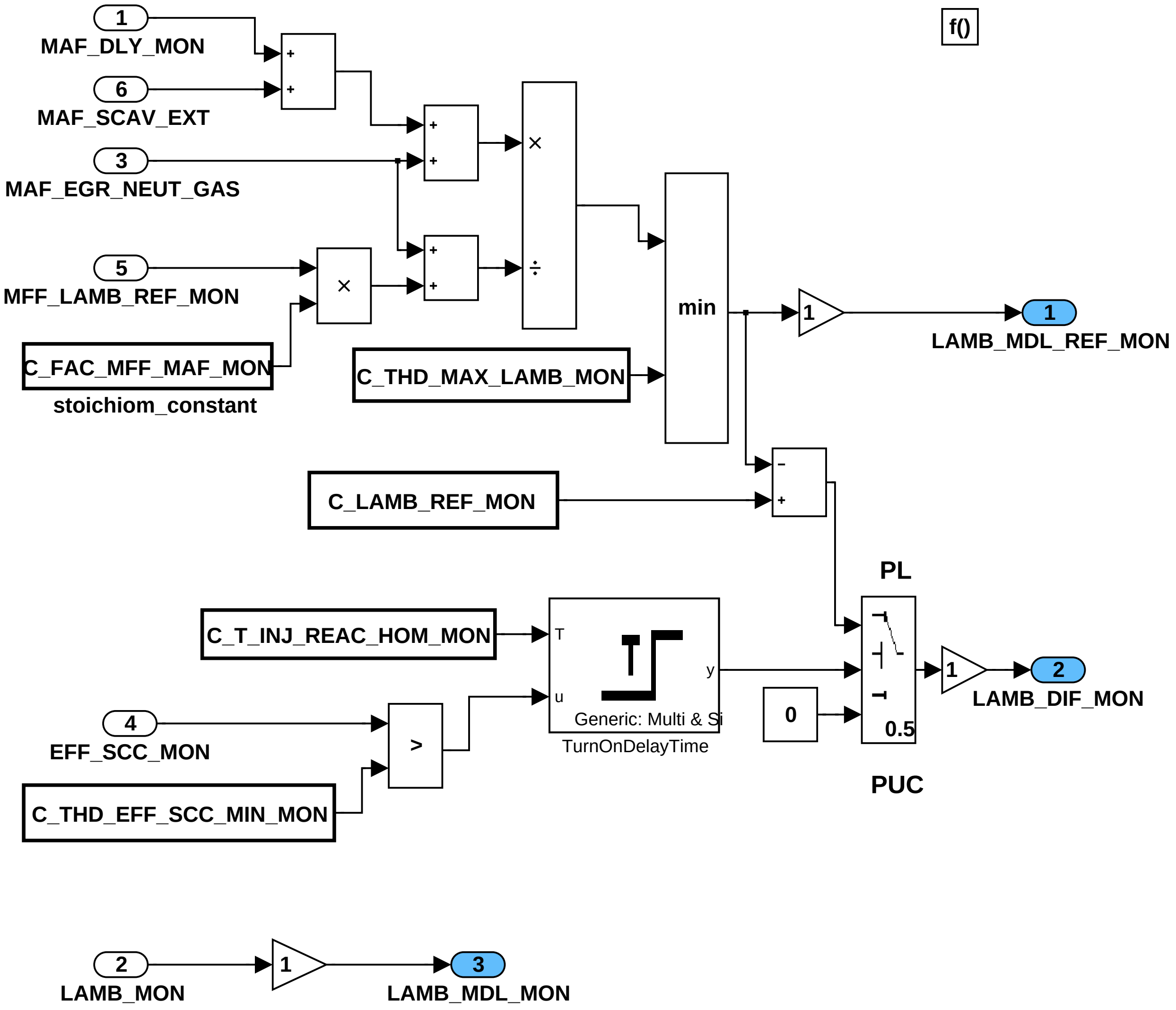


Figure 65 ECM2_DTSYSLADIF/ CLC_LAMB_HOM_MDL

11.17.1.4 ECM2_DTSYSLADIF/CLC_LAMB_S_MDL

Note: All internal values of the Pade-filter and the digital lowpass filter, shall be set to Init-value, once **at every activation of CLC_LAMB_S_MDL**. Also the T_INJ_REAC_MON -timer should be initialized with C_T_INJ_REAC_MON. Also the T_MIN_PUC_MON- timer should be set to 0, its output Y to 1. Further the Q output should be set to 1 at this time. The Reset input of the RSFlipFlop should have higher priority than the Set- input.

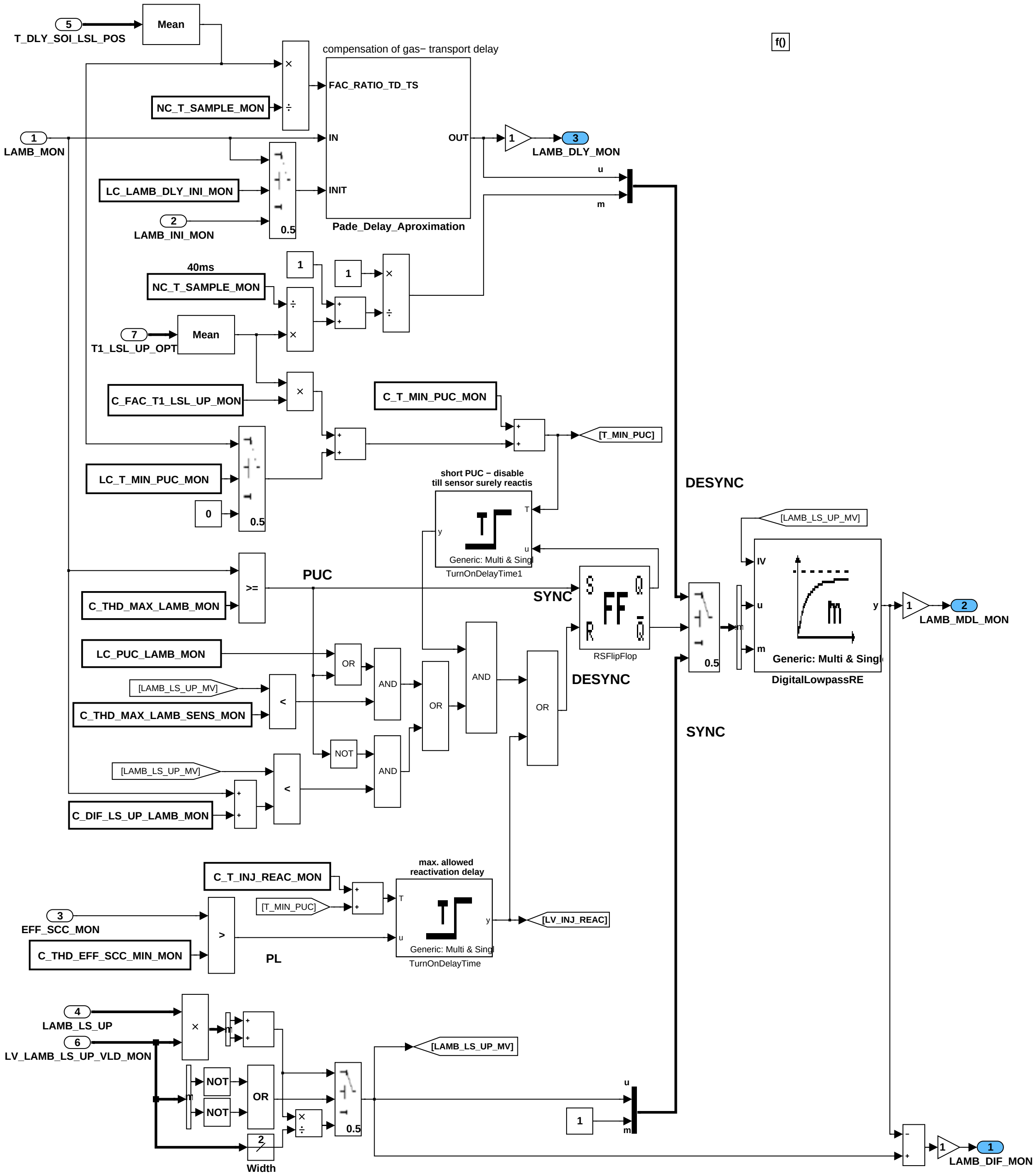


Figure 66 ECM2_DTSYSLADIF/ CLC_LAMB_S_MDL

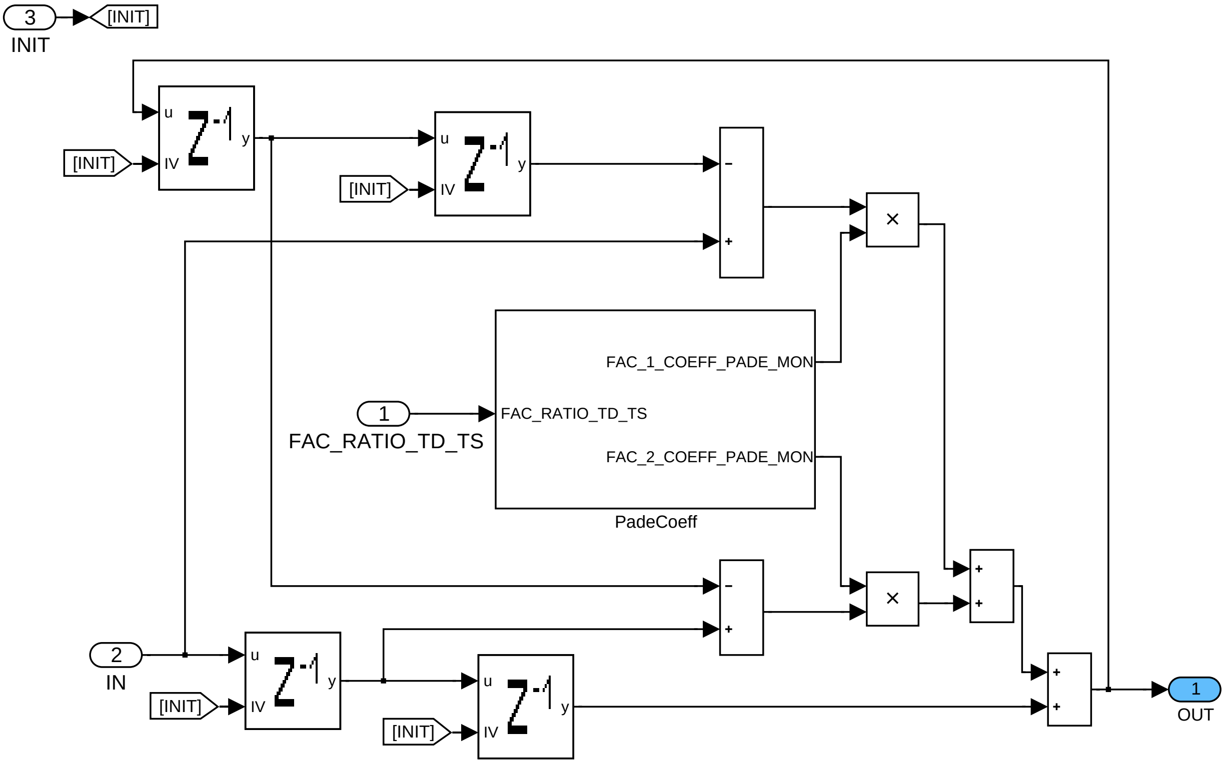


Figure 67 ECM2_DTSYSLADIF/ CLC_LAMB_S_MDL/ Pade_Delay_Aproximation

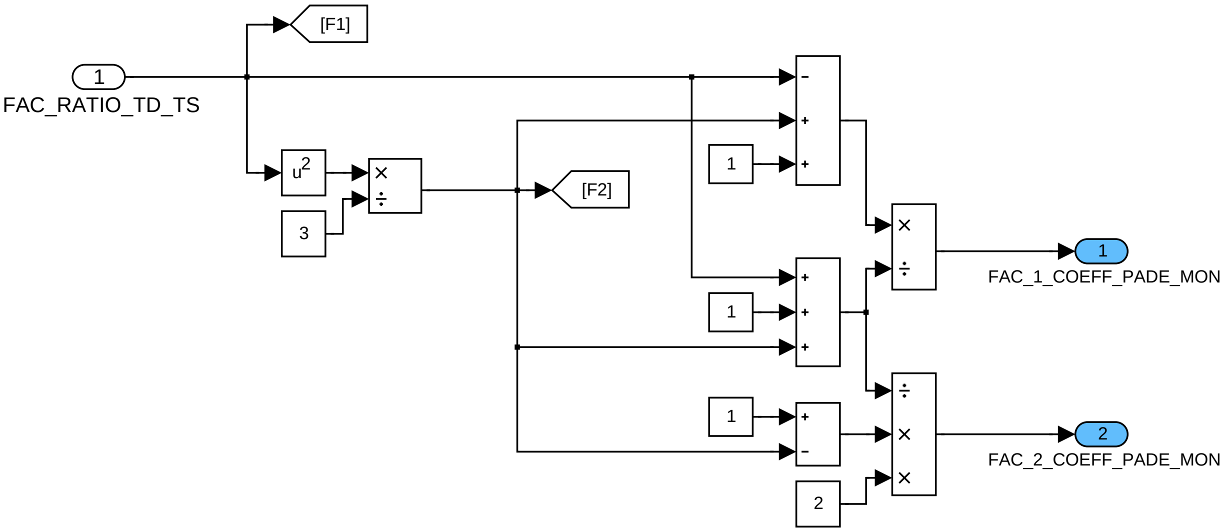


Figure 68 ECM2_DTSYSLADIF/ CLC_LAMB_S_MDL/ Pade_Delay_Aproximation/ PadeCoeff

11.18 Monitoring of conditions for cruise control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TQ_REQ_CRU_MON	O/V	0...FFH	0...1.99218	0.0078125	[-]
Scaling factor of cruise control torque demand in monitoring level					
LV_ERR_CRU_MON	O/V	0...1H	0...1	1	[-]
Fault in Cruise Control present					
LV_CRU_INH_MON	O/V	0...1H	0...1	1	[-]
logical variable for inhibition of cruise control					
ABC_CRU_TQ_MON	V	0...FFH	0...255	1	[-]
Anti bounce counter for monitoring of cruise control conditions					
T_CRU_INH_MON	V	0...FFH	0...10.2	0.04	[s]
Timer for inhibition of cruise control torque demand in monitoring level					

Input data:

LV_TQI_MON_ACT_MON	LV_CRU_ACT	FAC_TQ_REQ_CRU	LV_CRU_MAIN_SWI
LV_ERR_MSW_2	LV_ERR_MSW_3	LV_ERR_MSW_TOG	
VS_FIL	LV_BRAKE_MON		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_CRU_TQ_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment for cruise torque request					
C_ABC_MAX_CRU_TQ_MON	1	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter					
C_T_CRU_INH_MON	1	0...FFH	0...10.2	0.04	[s]
Delay time for disable of cruise control in monitoring level					
C_VS_MIN_CRU_MON	1	0...FFH	0...510	2	[km/h]
Minimum threshold for vehicle speed control active					

Import actions:

ACTION_ECM3_Service9TaskPfm(IN < >)
ACTION_ECM3_Service10TaskPfm(IN < >)
ACTION_ECM3_Service11TaskPfm(IN < >)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 3.

ACTION_ECM3_McChkStack( )

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service9TaskPfm() and ACTION_ECM3_Service10TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT < >, OUT < >, IN < >)
ACTION_ECM3_ChkCpl(IN < >, IN < >)
ACTION_ECM3_ReadChkCpl(OUT < >, IN < >, IN < >)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

ECM2 DTSYSCRU: General information:

The cruise control function is able to request a lot of torque and to accelerate the car up to high speed. In order to avoid dangerous situations caused by a fault of the cruise control function. the main activation/deactivation conditions are monitored within the etc safety concept. Therefore the basic signals from the master control (Hauptschalter) and the brake are evaluated.
In a cruise control function including active deceleration also the conditions for deceleration enabling have to be monitored.

Application Condition

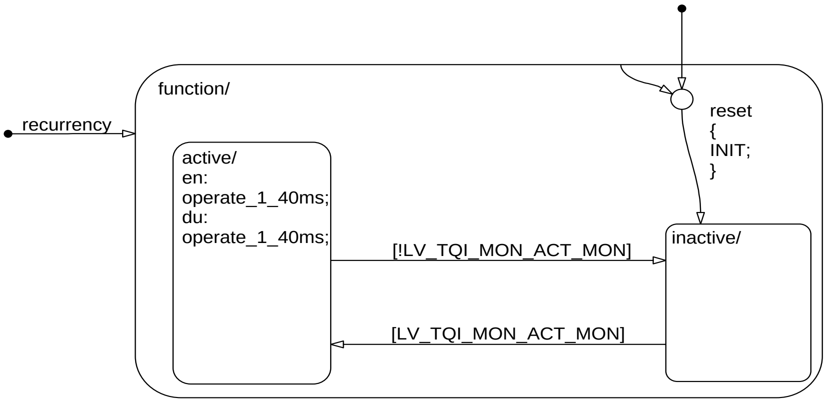


Figure 69 ECM2_DTSYSCRU/ APP_CDN/ Chart

11.18.1.1 SUBFUNCTION: INIT

For condition see 'Application Incidences of Process Monitoring'

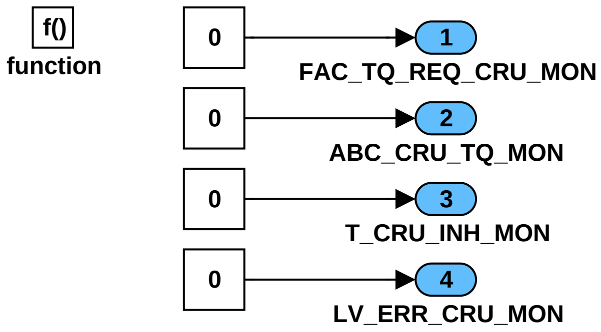
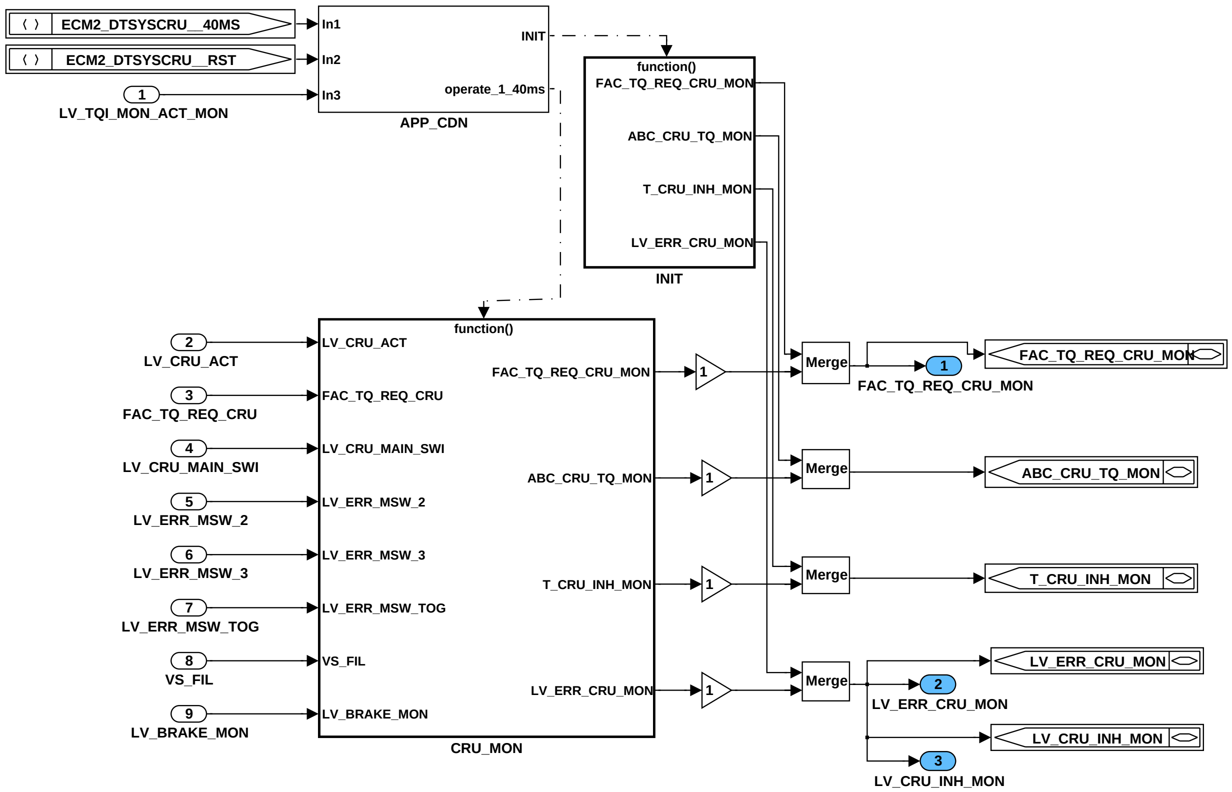


Figure 70 ECM2_DTSYSCRU / INIT

Function Description



SDA_SRS / SDA 4.0 14−Mar−2006

Figure 71 ECM2_DTSYSCRU

11.18.1.2 SUBFUNCTION: ME00B_CRU_MON

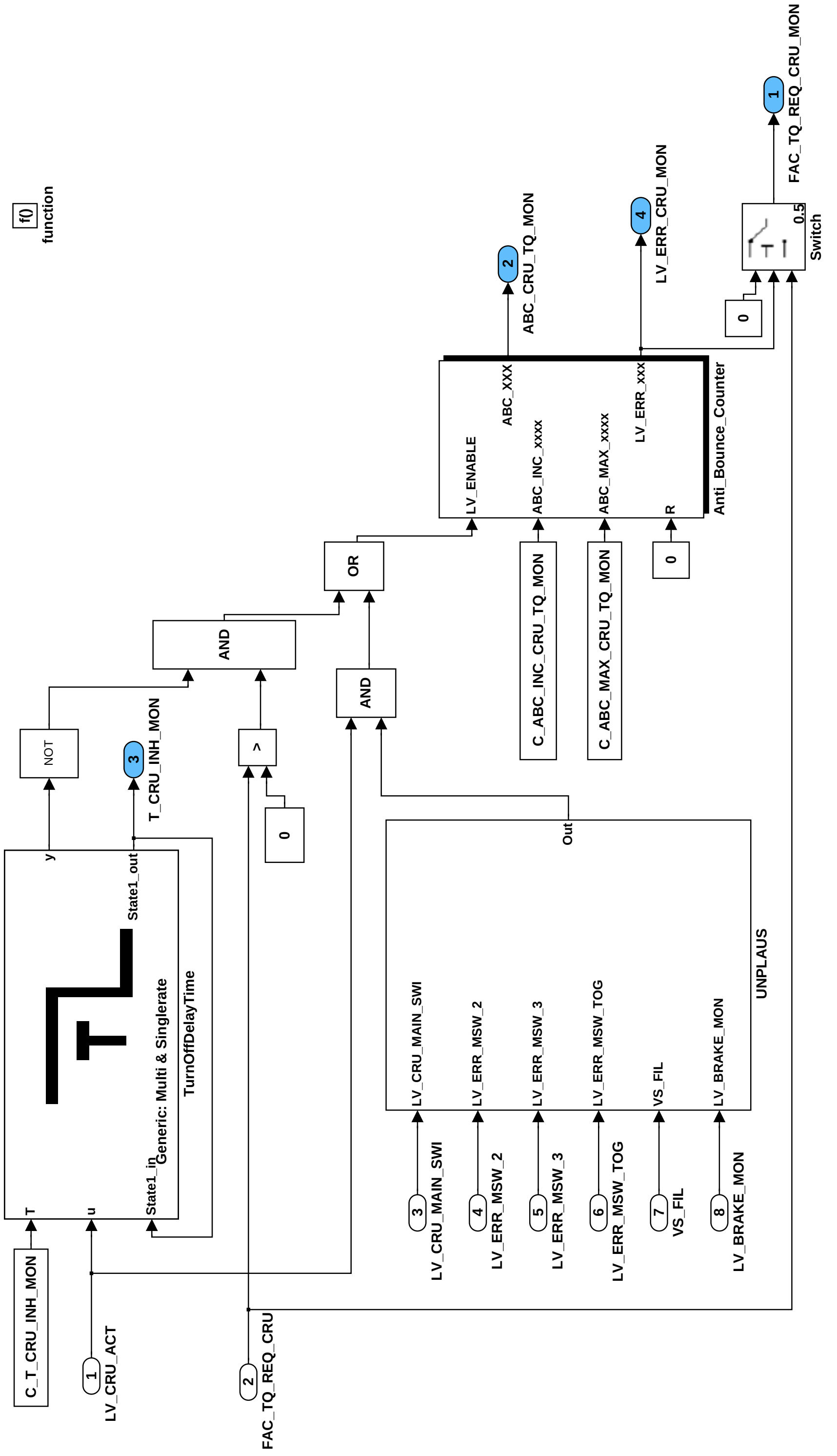


Figure 72 ECM2_DTSYSCRU / ME00B_CRU_MON

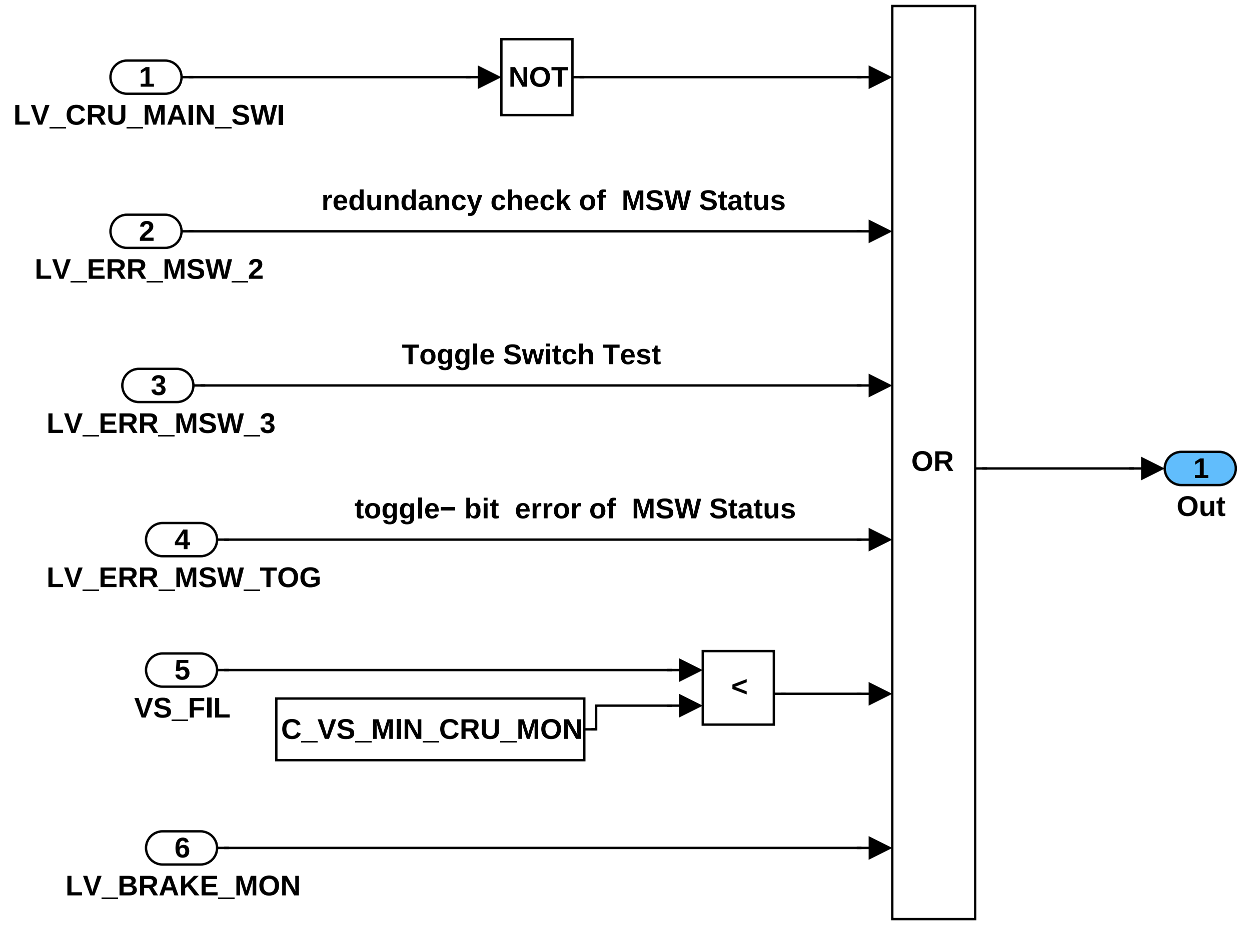


Figure 5 ECM2_DTSYSCRU / ME00B_CRU_MON/ UNPLAUS

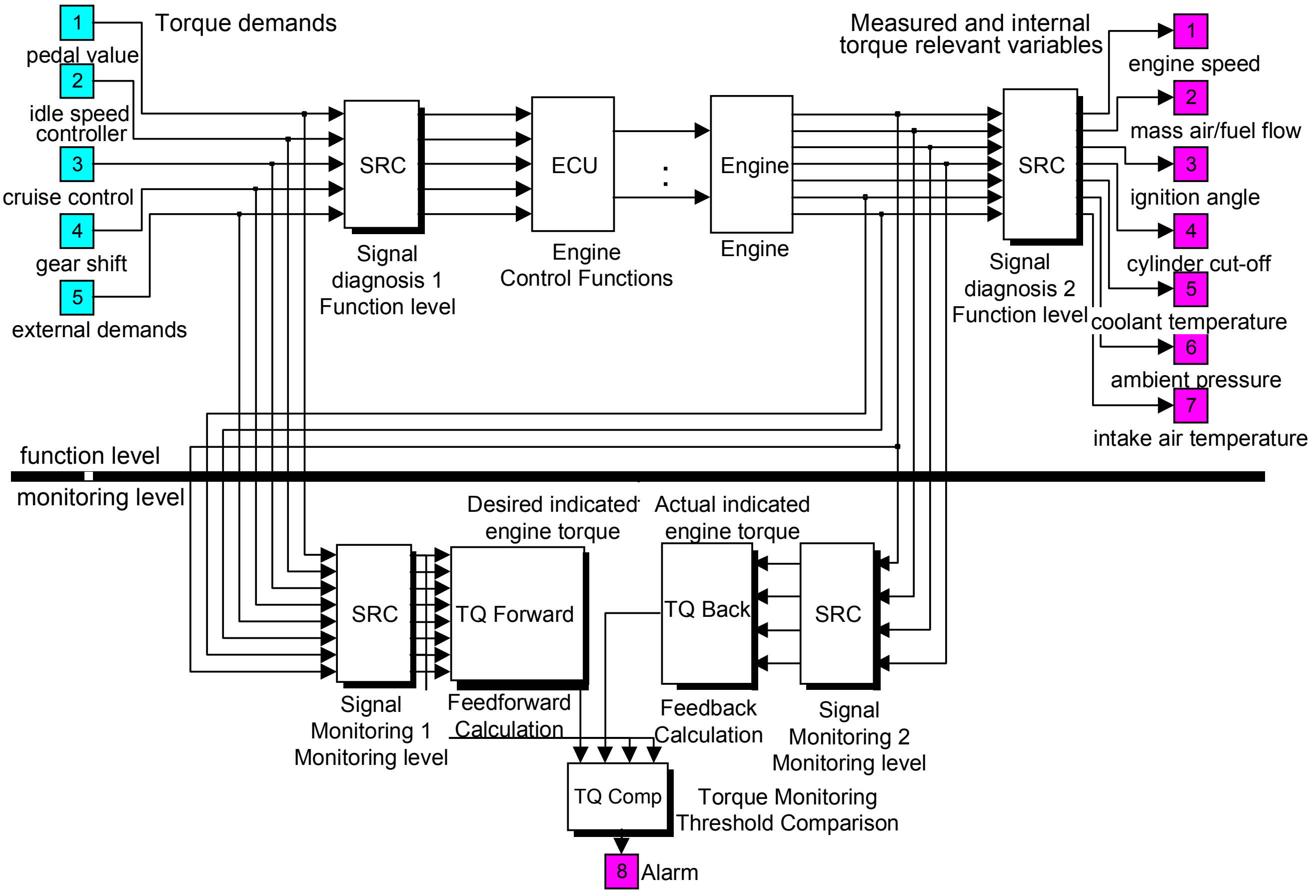
11.19 Torque Monitoring Overview

In the function level the various torque demands coming from the driver, the idle speed controller etc. are co-ordinated to a resulting quantfied charge setpoint. At the same time the actual indicated engine torque is calculated based on several measured and internal torque relevant variables.

The aim of the torque monitoring is to detect an increased actual indicated engine torque compared to the desired indicated engine torque. Therefore the co-ordination of the various torque demands and the calculation of the actual indicated engine torque done in the function level (level 1) is performed once again in the monitoring level (level 2), however with a simplified structure. Then the deviation between the actual (TQI_AV_MON) and the desired indicated engine torque (TQI_SP_MON) is checked for exceeding a threshold.

Signal flow diagram:

Overview



11.20 Desired indicated engine torque

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TQ_REQ_DRIV_MON	V/O	0...FFH	0...1.99218	0.0078125	[-]
Scaling factor for requested torque at clutch from driver					
FAC_TQ_REQ_MON	V/O	0...FFH	0...1.99218	0.0078125	[-]
Scaling factor for requested torque at clutch from driver and cruise control					
TQI_SP_MON	V/O	0...FFH	0...510	2	[Nm]
Desired indicated engine torque					
TQI_REQ_TOT_MON	V	0...FFH	0...510	2	[Nm]
Desired non-filtered indicated engine torque					
T_TQI_MON	V	0...FFH	0...10.2	0.04	[s]
dead time counter for torque decrease					

Input data:

PV_AV_AD_MON	TQI_INC_EXT_MON	TQI_MIN_PU_MON	TQ_DIF_P_D_IS_MON
TQ_MAX_CLU_MON	TQ_MIN_CLU_MON	FAC_TQ_REQ_CRU_MON	LV_TQI_MON_ACT_MON
N_32_MON	TQ_LOSS_MON	LV_SWI_FAC_TQ_REQ_MON	TQ_DROF_MON

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_TQI_MON	1	0...FFH	0...10.2	0.04	[s]
Dead time for torque decrease					
C_CRLC_TQI_SP_MON	1	0...FFH	0...0.99609	3.9063e-3	[-]
filter constant for realization of first order filter					
IP_FAC_TQ_REQ_DRIV_1_MON	8*8	0...FFH	0...1.99218	0.0078125	[-]
LDPM_N_32_MON_IP_FAC_TQ_MON	8	0...FFH	0...8160	32	[rpm]
LDPM_PV_AV_AD_MON_IP_FAC_MON	8	0...FFH	0...99.60937	0.390625	[%]
Scaling factor for requested torque at clutch from driver in other mode					
IP_FAC_TQ_REQ_DRIV_MON	8*8	0...FFH	0...1.99218	0.0078125	[-]
LDPM_N_32_MON_IP_FAC_TQ_MON	8	0...FFH	0...8160	32	[rpm]
LDPM_PV_AV_AD_MON_IP_FAC_MON	8	0...FFH	0...99.60937	0.390625	[%]
Scaling factor for requested torque at clutch from driver					

11.20.1 ECM2_DTSYSTQISP

Import actions:

ACTION_ECM3_Service6TaskPfm(IN <>)

ACTION_ECM3_Service7TaskPfm(IN <>)

ACTION_ECM3_Service8TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The argument has the value 4.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service6TaskPfm() and ACTION_ECM3_Service7TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT<>,OUT<>,IN<>)
ACTION_ECM3_ChkCpl(IN<>, IN<>)
ACTION_ECM3_ReadChkCpl(OUT<>,IN<>, IN<>)

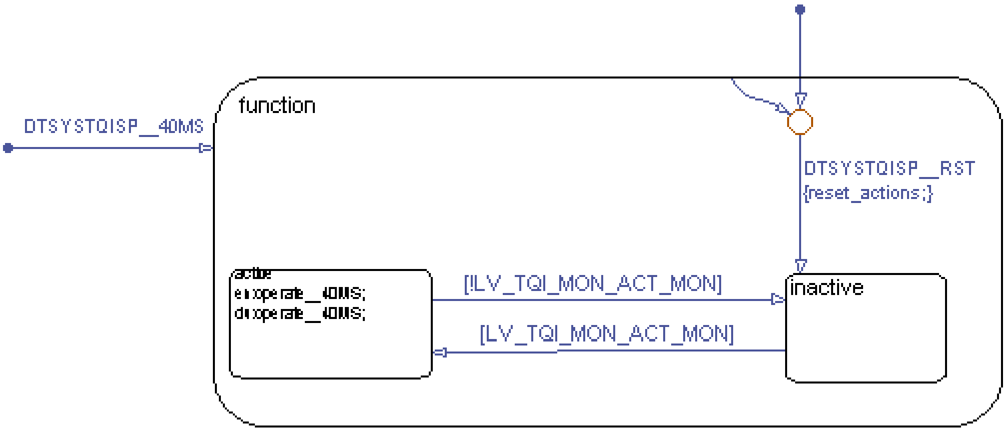
Note: This actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

General information:

The objective of this module is to calculate the desired indicated engine torque which derives from the function level (level 1). All torque increasing demands are taken into account. The different parts are:

- calculation of scaling factor for requested torque at clutch from driver (FAC_TQ_REQ_MON) in different operation modes, depending on engine speed (N_32_MON) and adapted accelerator pedal activation degree (PV_AV_AD_MON)
- calculation of requested torque at clutch from driver
- switch from brake torque to indicated torque demands
- adding the torque request for drive off assistance
- filtering the raw torque demand in falling direction to take into account the system response time
- maximum choice between internal and external torque demands
- If the engine is in the trailing throttle state i.e. no driver demand but limitation on indicated engine torque to ensure a safe combustion at low load conditions, then the torque setpoint in L2 is limited by TQI_MIN_PU_MON.
- adding internal increasing torque demands (TQ_DIF_P_D_IS_MON)

Application Condition



Initialisation(DTSYSTQISP__RST): for condition see 'Application Incidences of Process Monitoring'.

Function Description

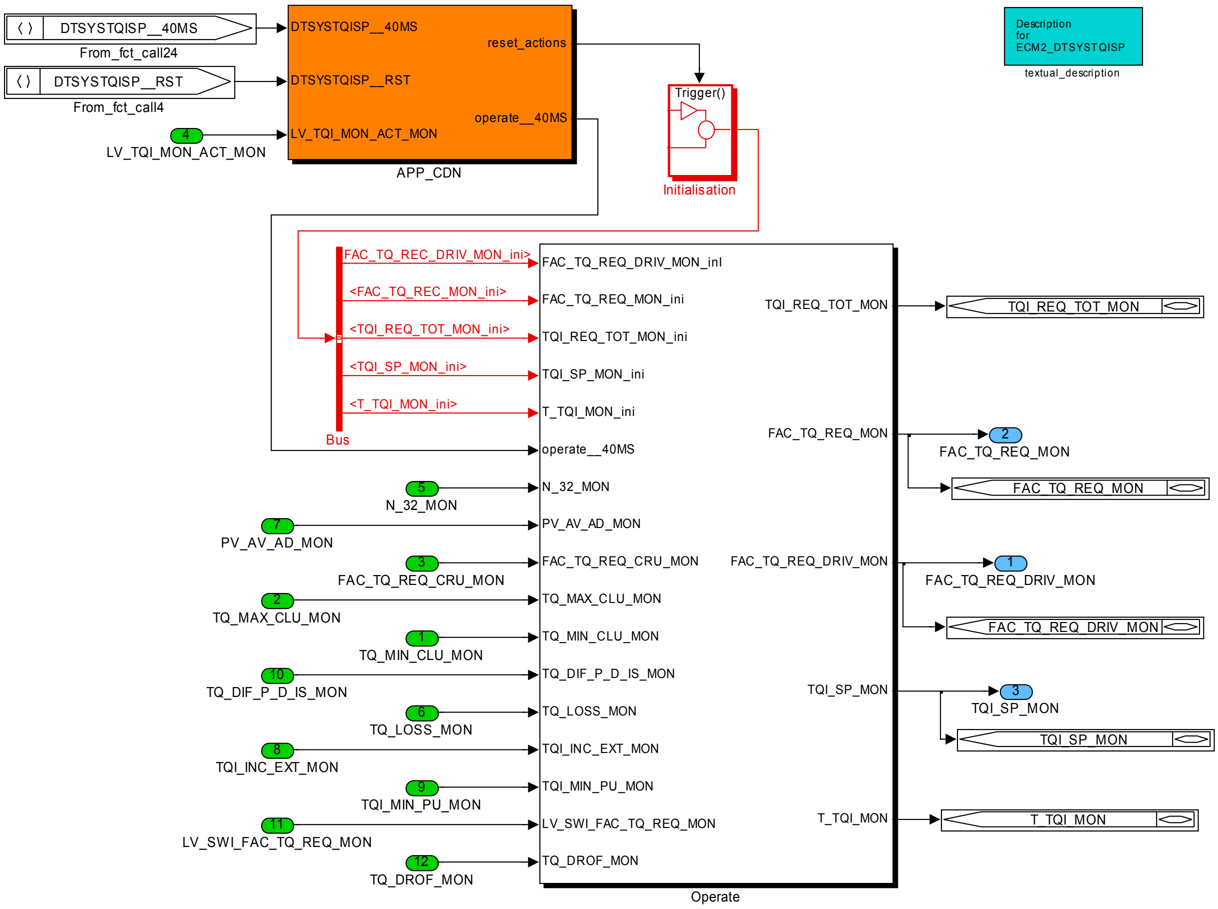


Figure 73 ECM2_DTSYSTQISP

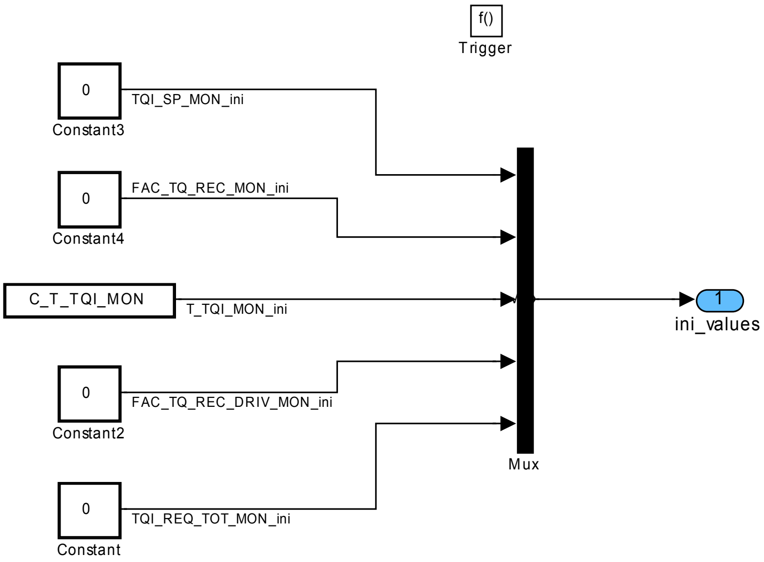


Figure 74 ECM2_DTSYSTQISP/ Initialisation

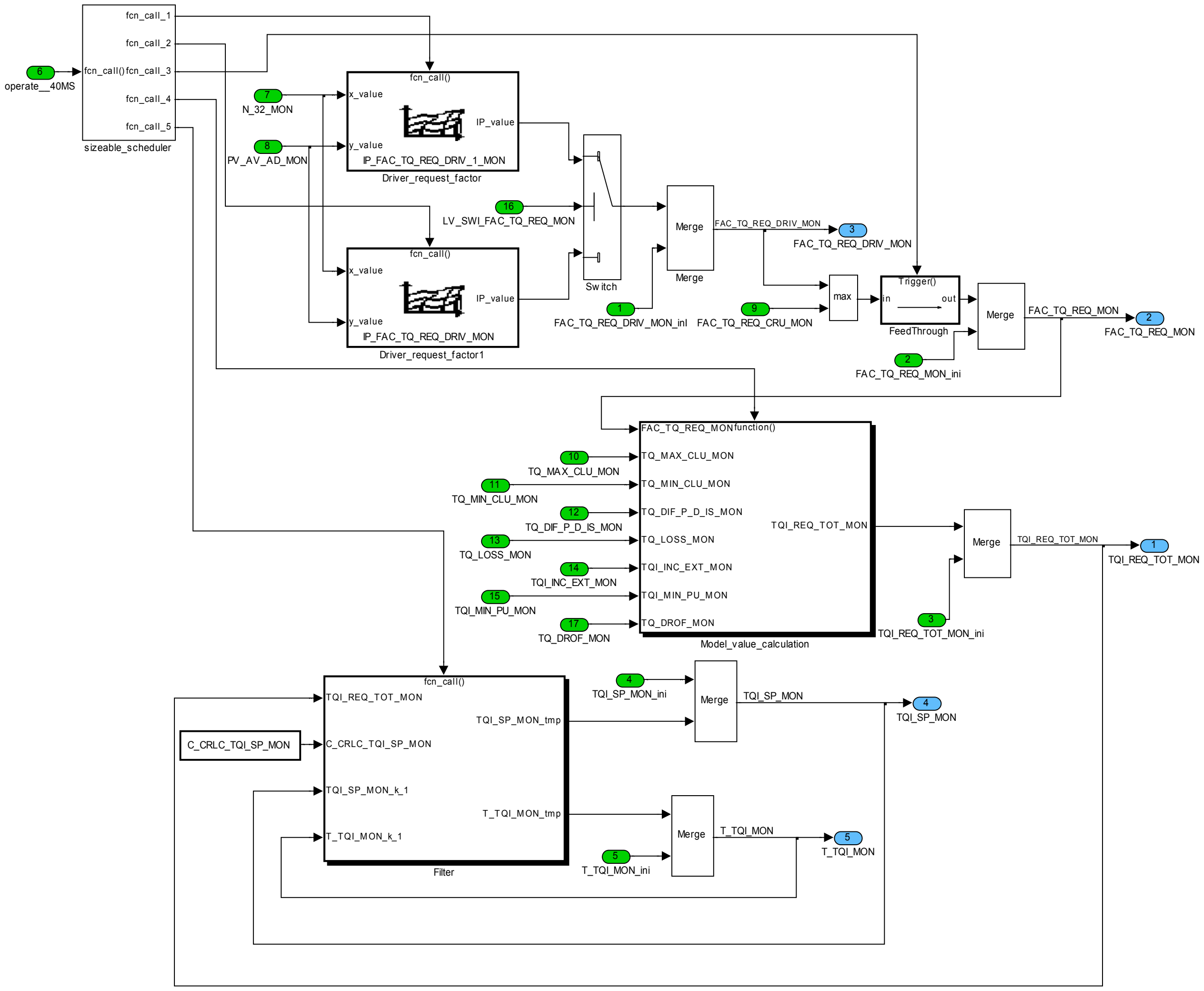


Figure 75 ECM2_DTSYSTQISP/ Operate

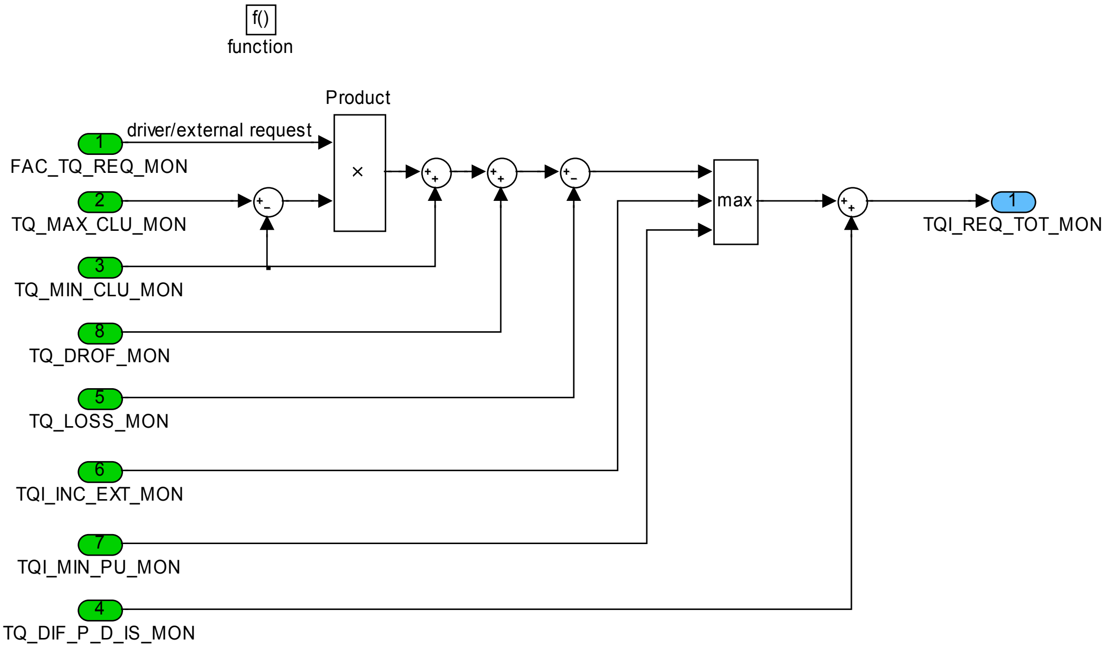


Figure 76 ECM2_DTSYSTQISP/ Operate/ Model_value_calculation

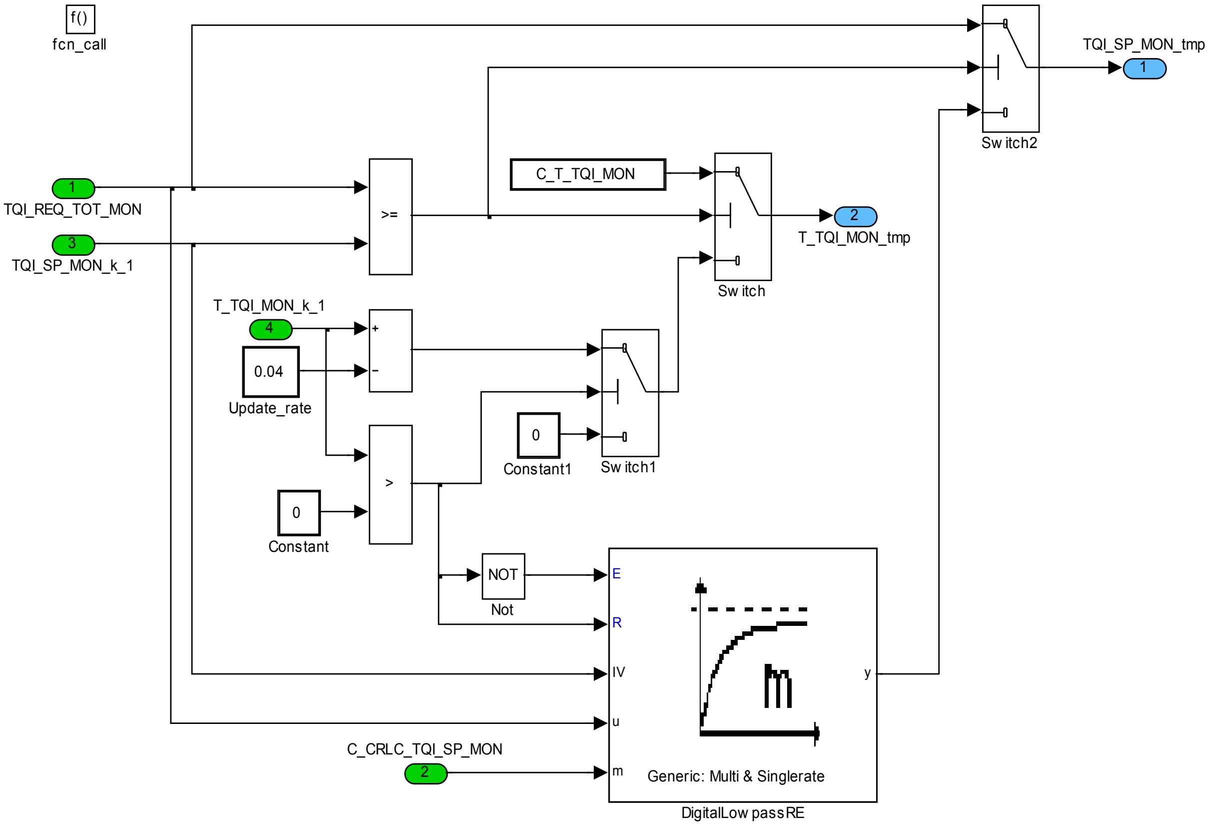


Figure 77 ECM2_DTSYSTQISP/ Operate/ Filter

11.21 Actual Fuel Mass Flow

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MFF_LAMB_REF_MON	O/V	0...FFFFH	0...1.389E+3	0.02119	mg/stk
Lambda relevant Mass Fuel Flow for stoichometric conditions					
MFF_LAMB_MON	O/V	0...FFFFH	0...1.389E+3	0.02119	mg/stk
Lambda relevant Mass Fuel Flow					
MFF_TQ_MON	O/V	0...FFFFH	0...1.389E+3	0.02119	mg/stk
Torque relevant Mass Fuel Flow					
EFF_SCC_MON	O/V	0...FFH	0...1.9921875	0.0078125	-
Actual efficiency of cylinder cut-off for torque monitoring					
LV_IDX_TI_PLAUS_MON	O/V	0...1H	0...1	1	-
Flag for unplausible injection index					
MFF_IV_1_MON	V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Mass Fuel Flow, first Pulse					
MFF_IV_2_MON	V	0...FFFFH	0...1.389E+3	0.02119	mg/stk
Mass Fuel Flow, second Pulse					
PRS_DIF_IV_2_MON	V	0...FFFFH	0...3.47776E+5	5.3067216	hPa
pressure difference at injector for second pulse					
PRS_DIF_IV_POST_MON	V	0...FFFFH	0...3.47776E+5	5.3067216	hPa
pressure difference at injector for post pulse					
LV_IV_EGY_POST_MON	V	0...1H	0...1	1	-
determination of energy level for post pulse					
TI_IDX_1_MON	V	0...FFFFH	0...65.535	0.001	ms
IDX- based, virtual injection time, first pulse					
TI_IDX_2_MON	V	0...FFFFH	0...65.535	0.001	ms
IDX- based, virtual injection time, second pulse					
TI_IDX_POST_MON	V	0...FFFFH	0...65.535	0.001	ms
IDX- based, virtual injection time, post injection pulse					
NR_CTR_IDX_MON	V	0...FFH	0...255	1	-
counter for maximal transient jitter					
TI_1_MES_IDX_MON	V	0...FFFFH	0...65.535	0.001	ms
Actual performed injection time for first pulse, based on IDX_TI					
MFF_IV_POST_MON	V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Mass Fuel Flow, post pulse					
TI_DIF_MES_IDX_MON	V	0...FFFFH	0...65.535	0.001	ms
difference between measured and indexed signal					
LV_ENA_GRD_TI_IDX_MON	V	0...1H	0...1	1	-
flag indicating stable dynamic conditions					
TI_IV_2_COR_MON	V	0...FFFFH	0...65.535	0.001	ms
standardized injection time for second pulse					
TI_IV_1_COR_MON	V	0...FFFFH	0...65.535	0.001	ms
standardized injection time for first pulse					
TI_IV_POST_COR_MON	V	0...FFFFH	0...65.535	0.001	ms
standardized injection time for post pulse					
PRS_DIF_IV_1_MON	V	0...FFFFH	0...3.47776E+5	5.3067216	hPa
pressure difference at injector for first pulse					

Input data:

LV_TQI_MON_ACT_MON	MAP_MES	FUP	PREV_STATE_IV

TI_TUN_ADD_IV[NC_CYL_NR]	TI_TUN_IV[NC_CYL_NR]	FAC_MFF_TFU	IDX_TI_1_MON
SOI_1_MES[NC_CYL_NR]	EOI_1_MES[NC_CYL_NR]	N_32_MON	IDX_TI_2_MON
SOI_2_MES[NC_CYL_NR]	EOI_2_MES[NC_CYL_NR]	TI_2_MES[NC_CYL_NR]	IDX_TI_POST_MON
EOI_POST_MES[NC_CYL_NR]	TI_POST_MES[NC_CYL_NR]	MFF_ADD_CYL_CP	TI_WUP
MFF_ADD_WF	FAC_LAM_AD_BAL[NC_CBK_EX_NR]	MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	TI_CAST
FAC_LAM_LIM_FIL[NC_CBK_EX_NR]	LV_IV_EGY_RNG_1	LV_IV_EGY_RNG_2	SOI_POST_MES[NC_CYL_NR]
EFF_SCC_AV	FAC_MFF_WUP_CUS_MON	LV_IV_EGY_RNG_3	MFF_SP_1_EXT_COR[NC_CYL_NR]
MFF_SP_2_EXT_COR[NC_CYL_NR]	MFF_SP_3_EXT_COR[NC_CYL_NR]	NC_CYL_NR	NC_CRK_INJ_REF_TDC
NC_CRK_INJ_BAS_REF	FAC_MFF_DIF_MV	TI_1_MES[NC_CYL_NR]	FAC_TI_PRS_COR_1
FAC_TI_PRS_COR_2	FAC_TI_PRS_COR_3	IGA_IGC_H_RNG[NC_CYL_NR]	FAC_LAM_PCTL[NC_CBK_EX_NR]
FAC_MFF_COR_INJ_MOD	NC_NR_IDX_TI_PRM_H	NC_NR_IDX_TI_PRM_L	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_IDX_TI_MIN_H_MON	1	0...FFFFH	0...65535	1	[-]
index position assigned to TI_MIN_H					
C_IDX_TI_MIN_L_MON	1	0...FFFFH	0...65535	1	[-]
index position assigned to TI_MIN_L					
C_THD_GRD_TI_IDX_MON	1	8000...7FFFH	-32.768...32.767	0.001	ms
gradient limit for stable dynamic conditions					
C_THD_MFF_AD_ADD_MAX_MON	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
Threshold for offset from Lambda adaptation					
C_THD_MFF_AD_FAC_MAX_MON	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Threshold for offset from Lambda adaptation					
C_THD_MIN_N_IDX_MON	1	0...FFH	0...8.16E+3	32	rpm
engine speed threshold for index plausibilisation					
C_THD_MIN_TI_MES_MON	1	0...FFFFH	0...65.535	0.001	ms
boundary for minimal injector opening					
C_THD_NR_CTR_IDX_MON	1	0...FFH	0...255	1	-
threshold for maximal transient jitter					
C_THD_SOI_POST_MON	1	FC40...3C0H	-360...360	0.375	°CRK
Threshold for torque participation of post pulse					
C_THD_TI_MES_MON	1	0...FFFFH	0...65.535	0.001	ms
THD for IDX switch					
C_TI_MES_OFS_MON	1	8000...7FFFH	-32.768...32.767	0.001	ms
Injection time offset for low needle lift pulses that exceed special timing parameter calculation.					
C_TI_OFS_H_MON	1	8000...7FFFH	-32.768...32.767	0.001	ms
Injection time offset for low needle lift pulses that exceed special timing parameter calculation.					
C_TI_OFS_L_MON	1	8000...7FFFH	-32.768...32.767	0.001	ms
Injection time offset for low needle lift pulses that exceed special timing parameter calculation.					
LC_IV_POST_EGY_RNG_MON	1	0...1H	0...1	1	-
Logical constant for high or low energy post pulse					
LC_THD_SOI_POST_MON	1	0...1H	0...1	1	-
Selection of the reference angle for post pulse recognition					
IP_FAC_TI_PRS_CYL_L_MON	12	0...FFFFH	0...15.9997559	2.44141E-4	-
LDPM_PRS_IP_FAC_TI_PRS_CYL_MON	12	0...FFFFH	0...3.47776E+5	5.3067216	hPa
correction of counter pressure on low needle lift					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_TI_PRS_CYL_MON	12	0...FFFFH	0...15.9997559	2.44141E-4	-
LDP_PRS_IP_FAC_TI_PRS_CYL_MON	12	0...FFFFH	0...3.47776E+5	5.3067216	hPa
correction of counter pressure on needle lift					
IP_THD_IDX_TI_MES_MON	6	0...FFFFH	0...65.535	0.001	ms
LDP_TI_MES_IP_THD_IDX_MON	6	0...FFFFH	0...65.535	0.001	ms
treshold for deviation between TI_MES and IDX_TI					
IP_MFF_TI_PRS_EGY_H_MON	18x12	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDP_TI_IP_MFF_EGY_H_MON	18	0...FFFFH	0...65.535	0.001	ms
LDPM_PRS_DEC_INJ_MON	12	0...FFFFH	0...3.47776E+5	5.3067216	hPa
injector characteristic at high energy level					
IP_MFF_TI_PRS_EGY_L_MON	18x12	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDP_TI_IP_MFF_EGY_L_MON	18	0...FFFFH	0...65.535	0.001	ms
LDPM_PRS_DEC_INJ_MON	12	0...FFFFH	0...3.47776E+5	5.3067216	hPa
injector characteristic at low energy level					
IP_PRS_INC_CMP_MON	12x16	0...FFFFH	0...31.99951	4.883E-4	-
LDP_EOI_IP_PRS_INC_CMP_MON	12	0...780H	0...720	0.375	°CRK
LDP_CRK_INJ_IP_PRS_INC_CMP_MON	16	0...780H	0...720	0.375	°CRK
influence of the varying pressure increase inside the cylinder					

11.21.1 ECM2_DTSYSMFFAC

This specification has to be coded according to the following method description: "M730204: Coding rules for process monitoring SW - ECM2.

Import actions:

ACTION_ECM3_Service15TaskPfm(IN<>)

ACTION_ECM3_Service16TaskPfm(IN<>)

ACTION_ECM3_Service17TaskPfm(IN<>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 1.

ACTION_ECM3_McChkStack()

Note: These actions are defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service15TaskPfm() and ACTION_ECM3_Service16TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)

ACTION_ECM3_ChkCpl(IN <>, IN <>)

ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: These actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

General information:

The ETC safety concept uses the mass-fuel-flow signal as a basis for the generation of a actual indicated engine torque. Therefore not the complete injection is seen as torque relevant. The post- pulse especially becomes weighted with a factor, which depends on its phasing. The sum of first, second and the weighted post- pulse (MFF_TQ_MON) is input for the determination of the actual engine torque. The un-weighted sum of all pulses (MFF_LAMB_MON) becomes corrected by elimination of all enrichments (e.g. WUP, CAST, etc.) with the aim to create a Lambda signal, which is comparable against the reference Lambda (see also 'Actual lambda deviation').

Description:

The fuel mass flow is determined by a summation of all injection pulses for every cylinder with an arithmetic mean- value calculation afterwards. From I/O- Software the feedback-signal for the opening duration of the injectors (TI_X_MES) is supplied to the monitoring level as well as the the charge CHA_IV_X_MES. The energy which was put into the injector's piezo- stack is taken as representative for the needle lift. After correction of these signals to standard conditions, a characteristic curve for a dedicated energy level, delivers the correspondence between injection time and mass-fuel-flow.

Hint!! Application Conditions - Initialisation: for condition see 'Application incidences of process monitoring'

Application Condition

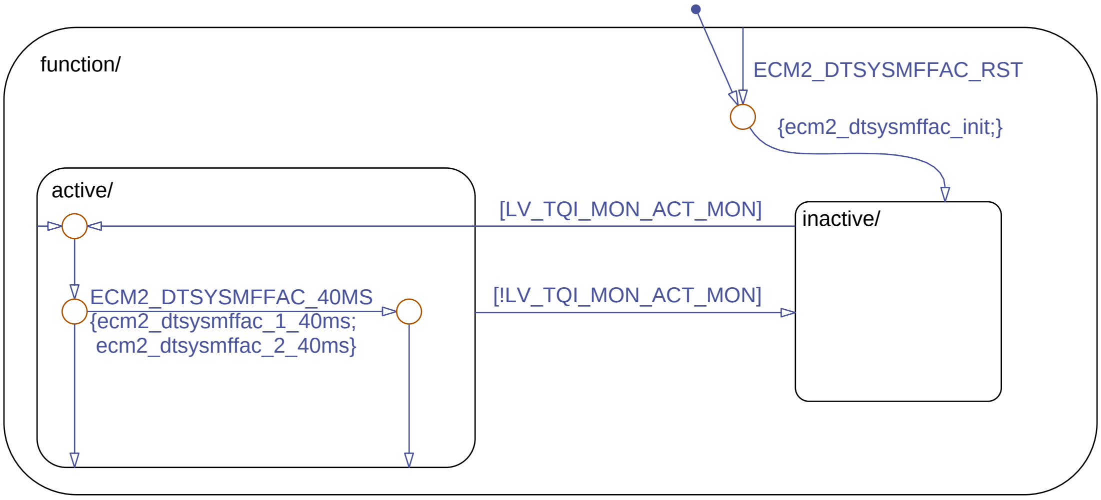
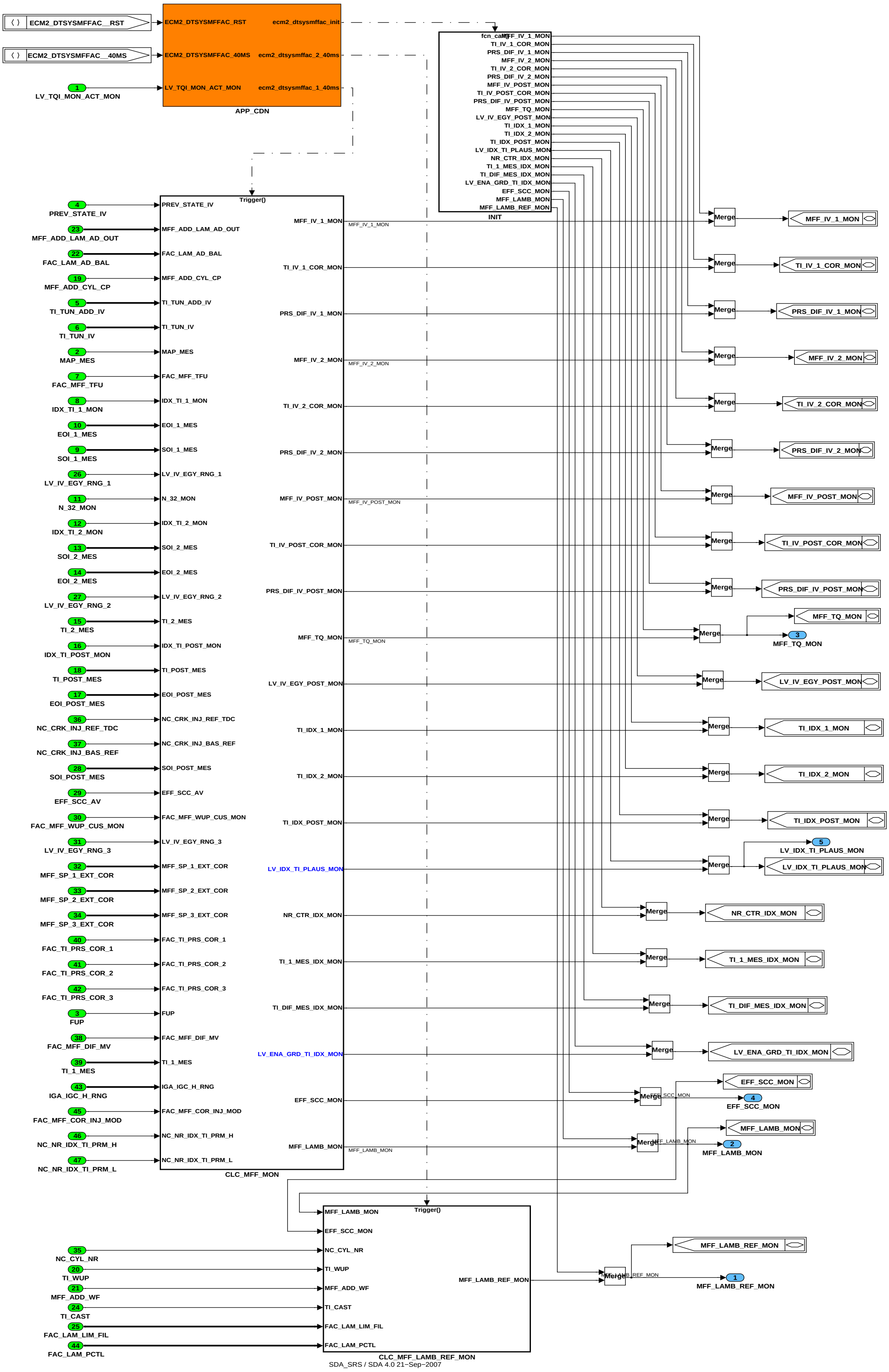


Figure 78 ECM2_DTSYSMFFAC/ APP_CDN/ Chart1

Function Description



SDA_SRS / SDA 4.0 21−Sep−2007

Figure 79 ECM2_DTSYSMFFAC

11.21.1.1 SUBFUNCTION: INIT

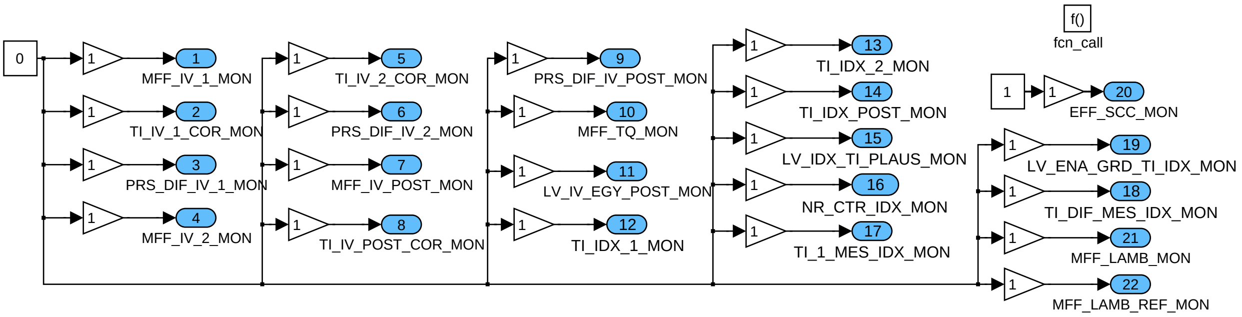


Figure 80 ECM2_DTSYSMFFAC/ INIT

11.21.1.2 SUBFUNCTION: CLC_MFF_LAMB_REF_MON

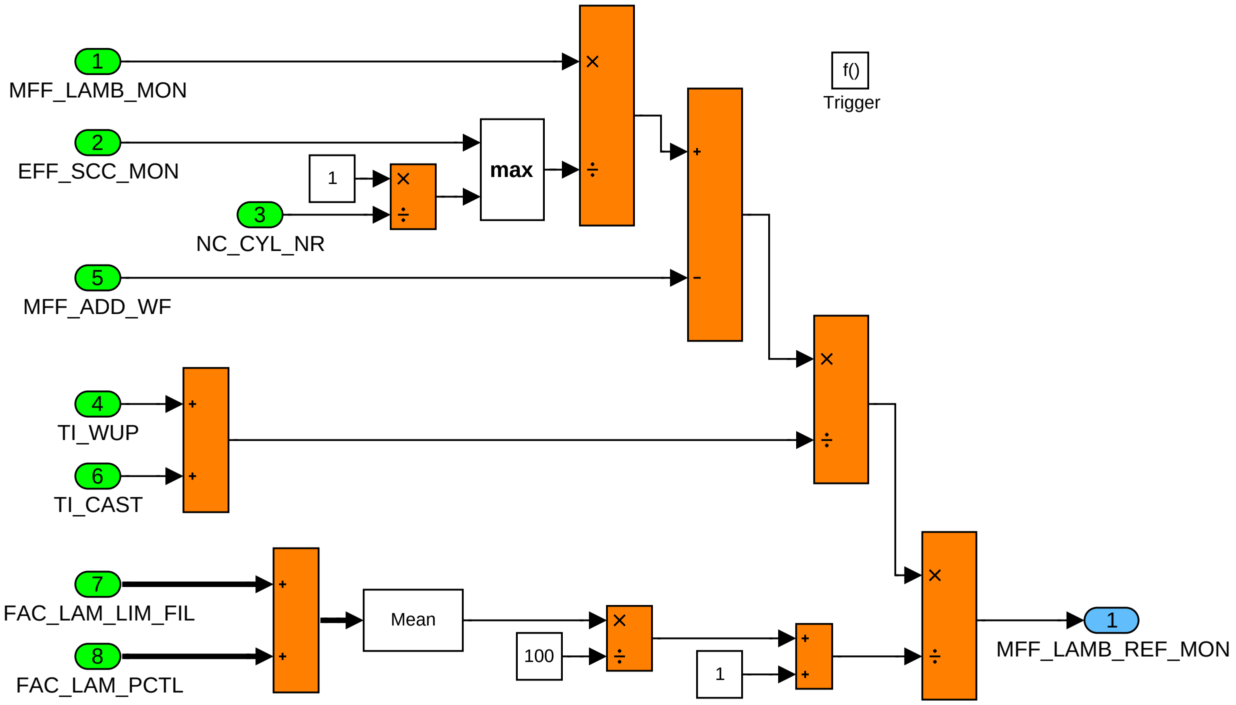


Figure 81 ECM2_DTSYSMFFAC/ CLC_MFF_LAMB_REF_MON

11.21.1.3 SUBFUNCTION: CLC_MFF_MON

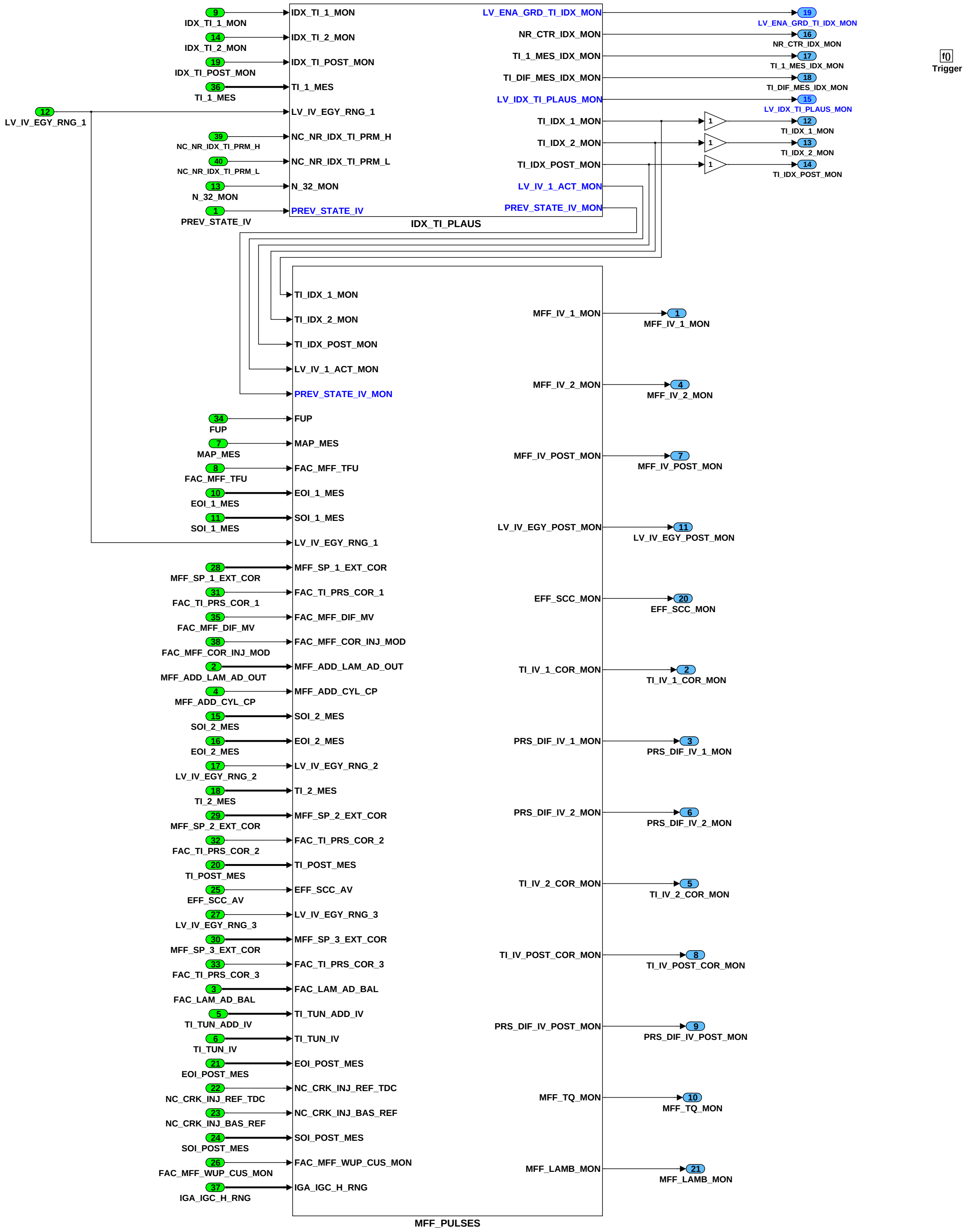
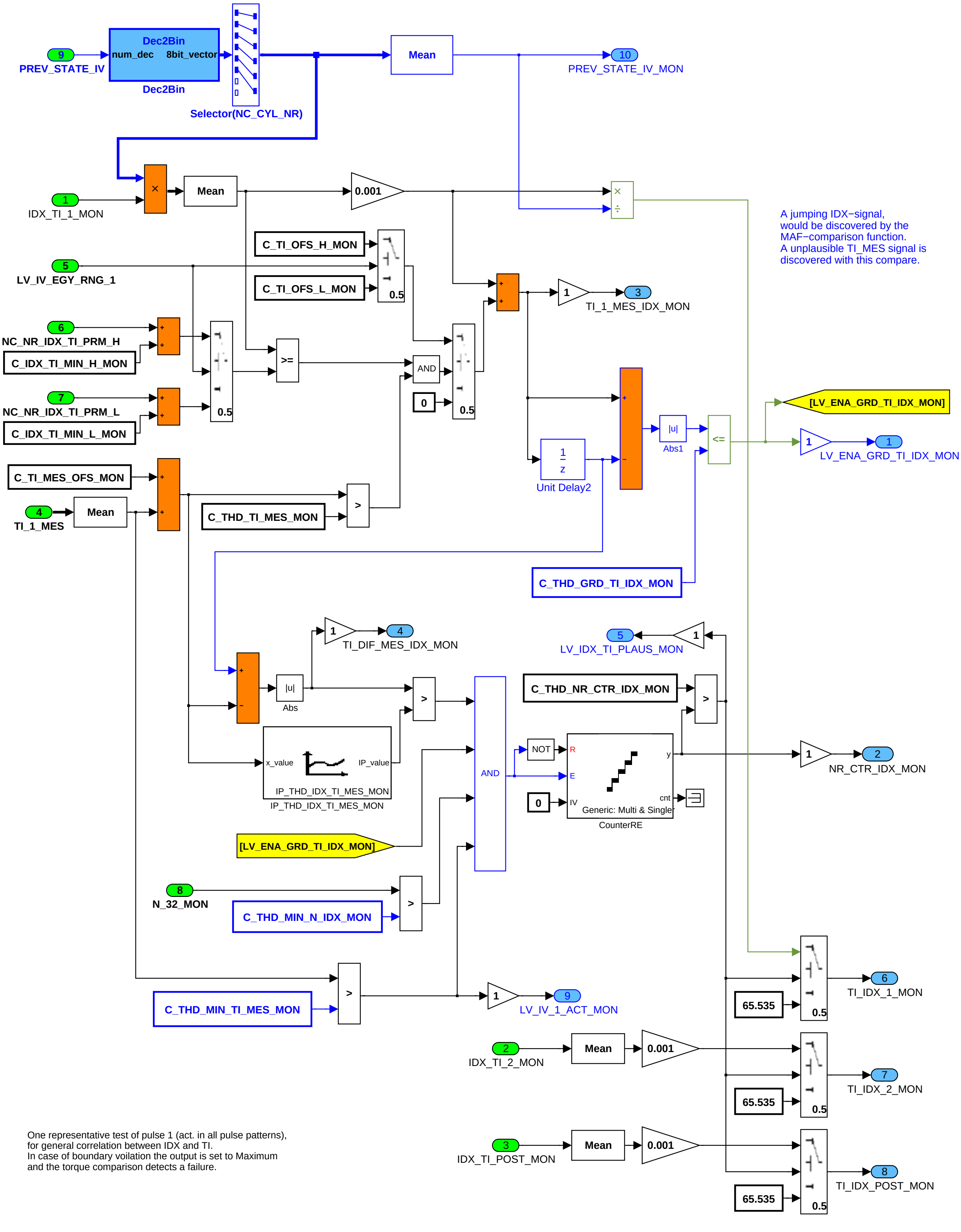


Figure 82 ECM2_DTSYSMFFAC/ CLC_MFF_MON



A jumping IDX−signal,
would be discovered by the
MAF−comparison function.
A unplausible TI_MES signal is
discovered with this compare.

One representative test of pulse 1 (act. in all pulse patterns),
for general correlation between IDX and TI.
In case of boundary voilation the output is set to Maximum
and the torque comparison detects a failure.

Figure 83 ECM2_DTSYSMFFAC/ CLC_MFF_MON/ IDX_TI_PLAUS

Chapter	Baseline		Include File
ECU Process Monitoring	4DC3940S		43E01601.00B
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Pages
1960 of 9643

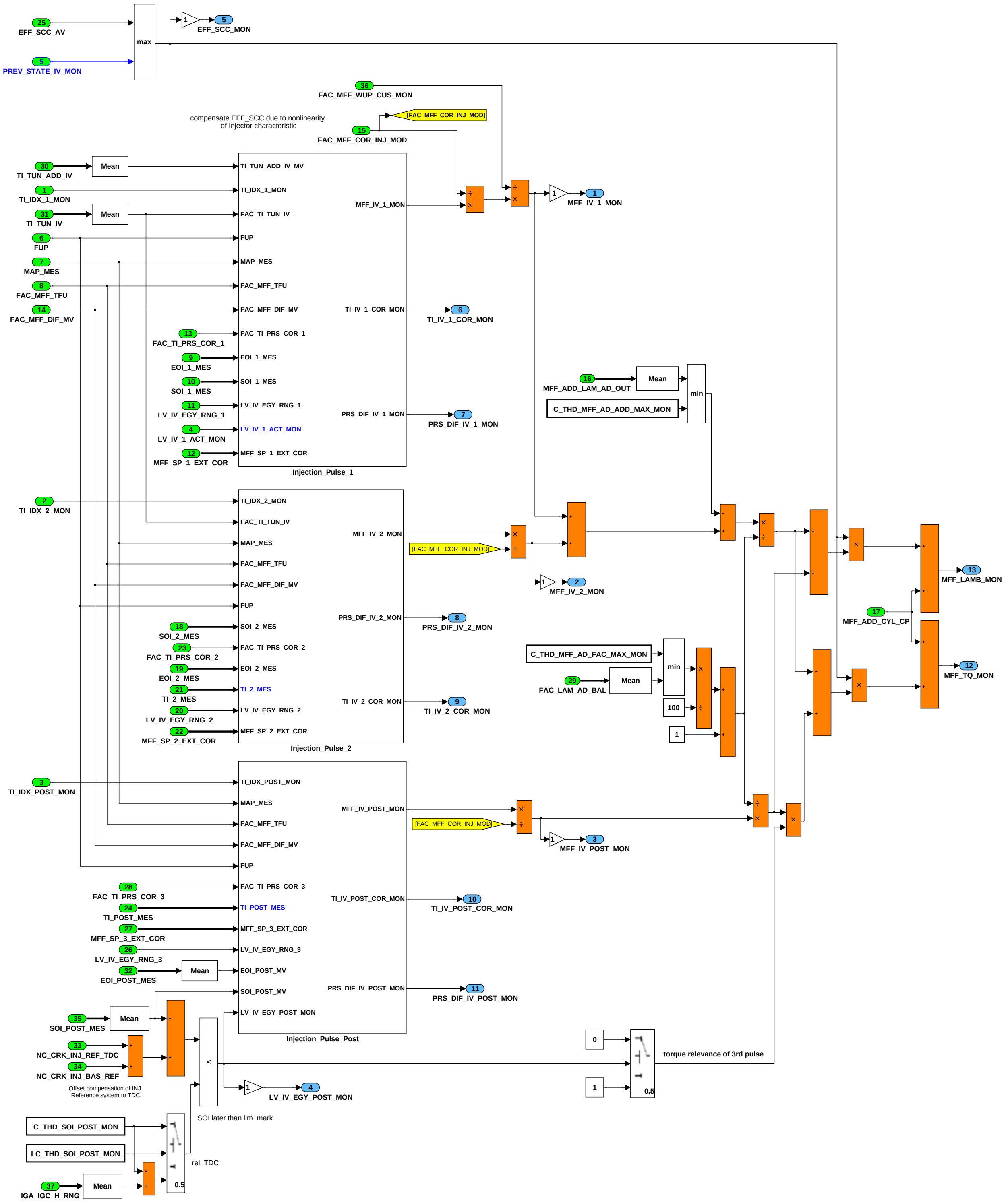


Figure 84 ECM2_DTSYSMFFAC/ CLC_MFF_MON/ MFF_PULSES

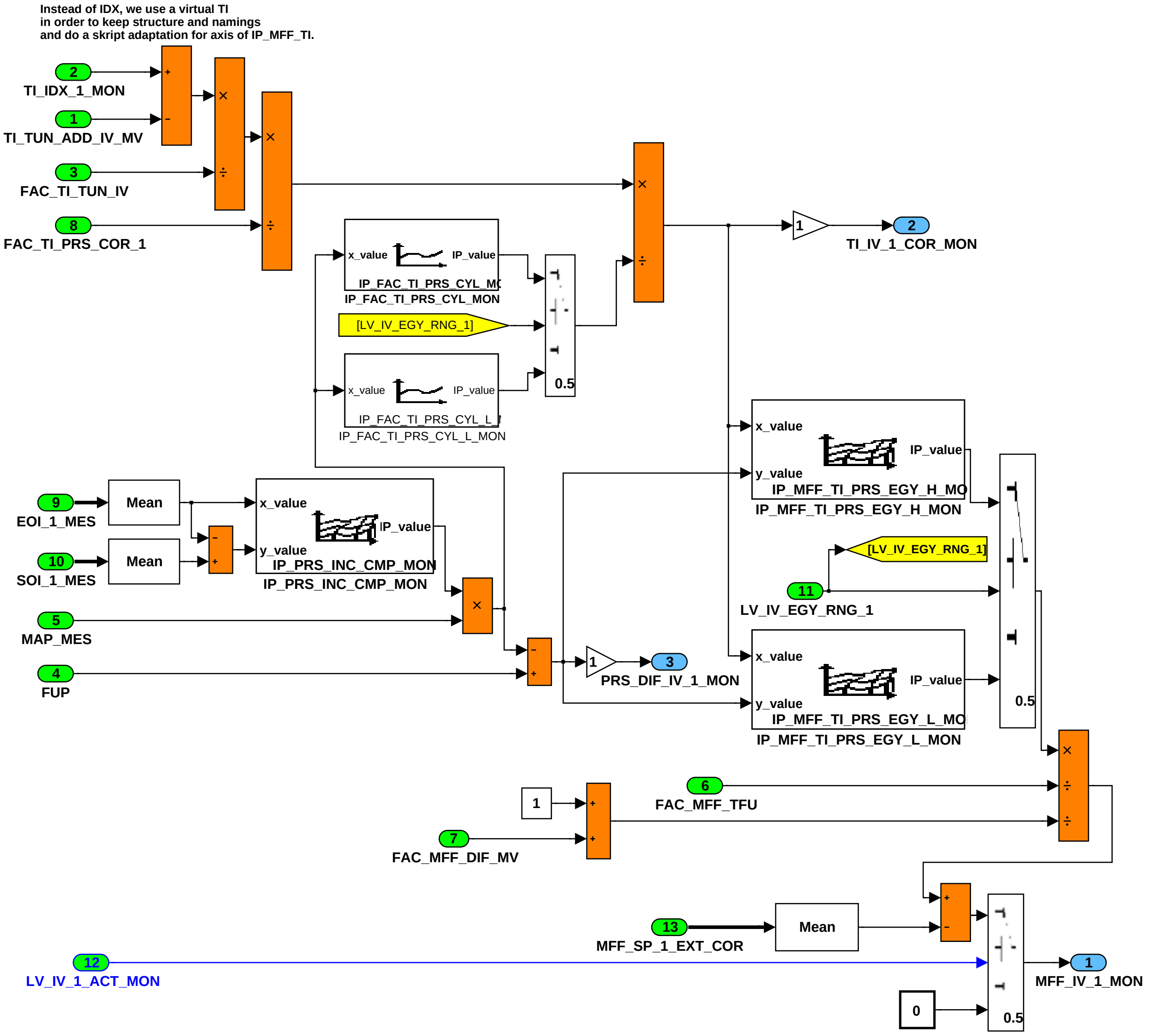


Figure 85 ECM2_DTSYSMFFAC/ CLC_MFF_MON/ MFF_PULSES/ Injection_Pulse_1

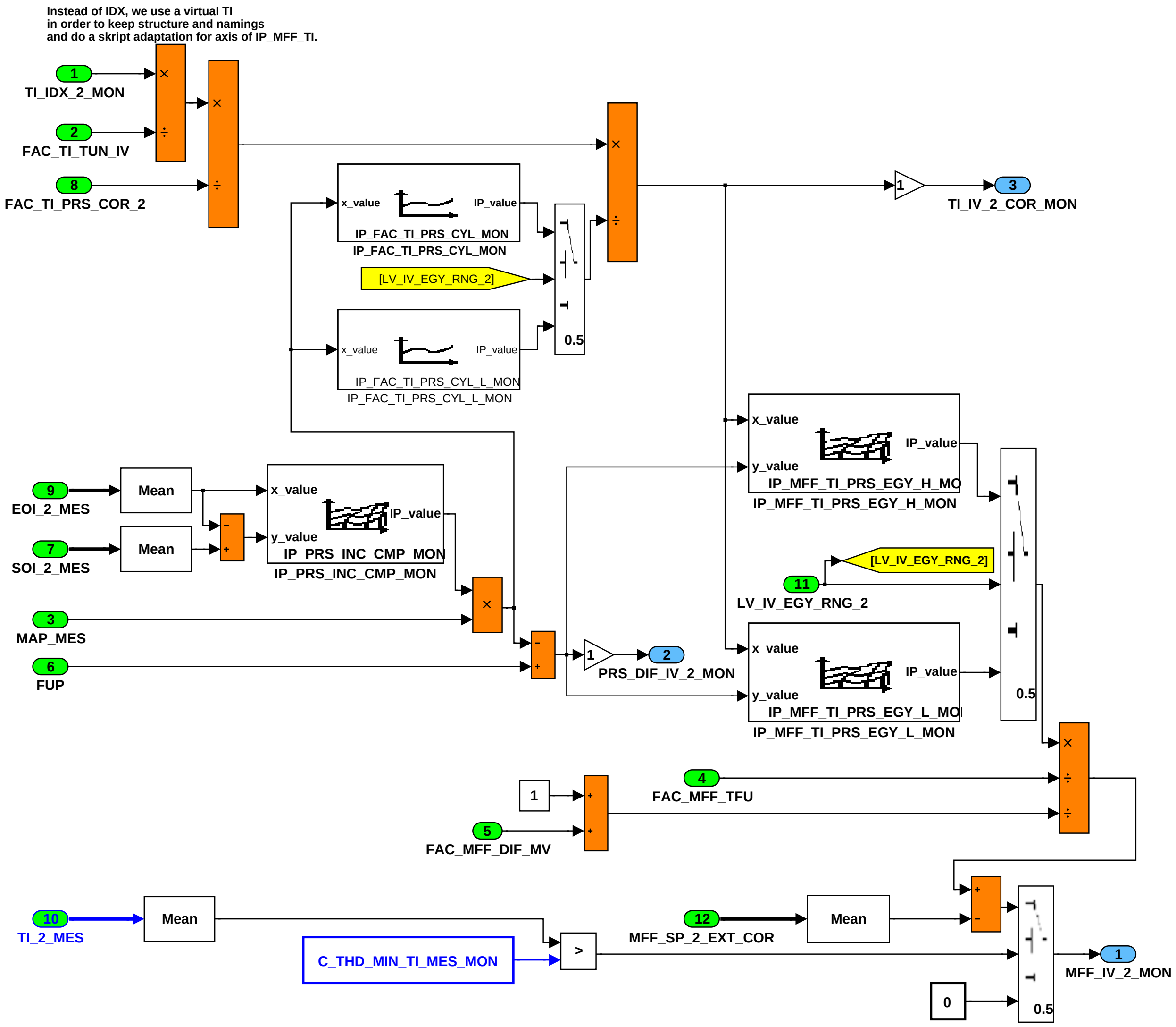


Figure 86 ECM2_DTSYSMFFAC/ CLC_MFF_MON/ MFF_PULSES/ Injection_Pulse_2

Instead of IDX, we use a virtual TI
in order to keep structure and namings
and do a skript adaptation for axis of IP_MFF_TI.

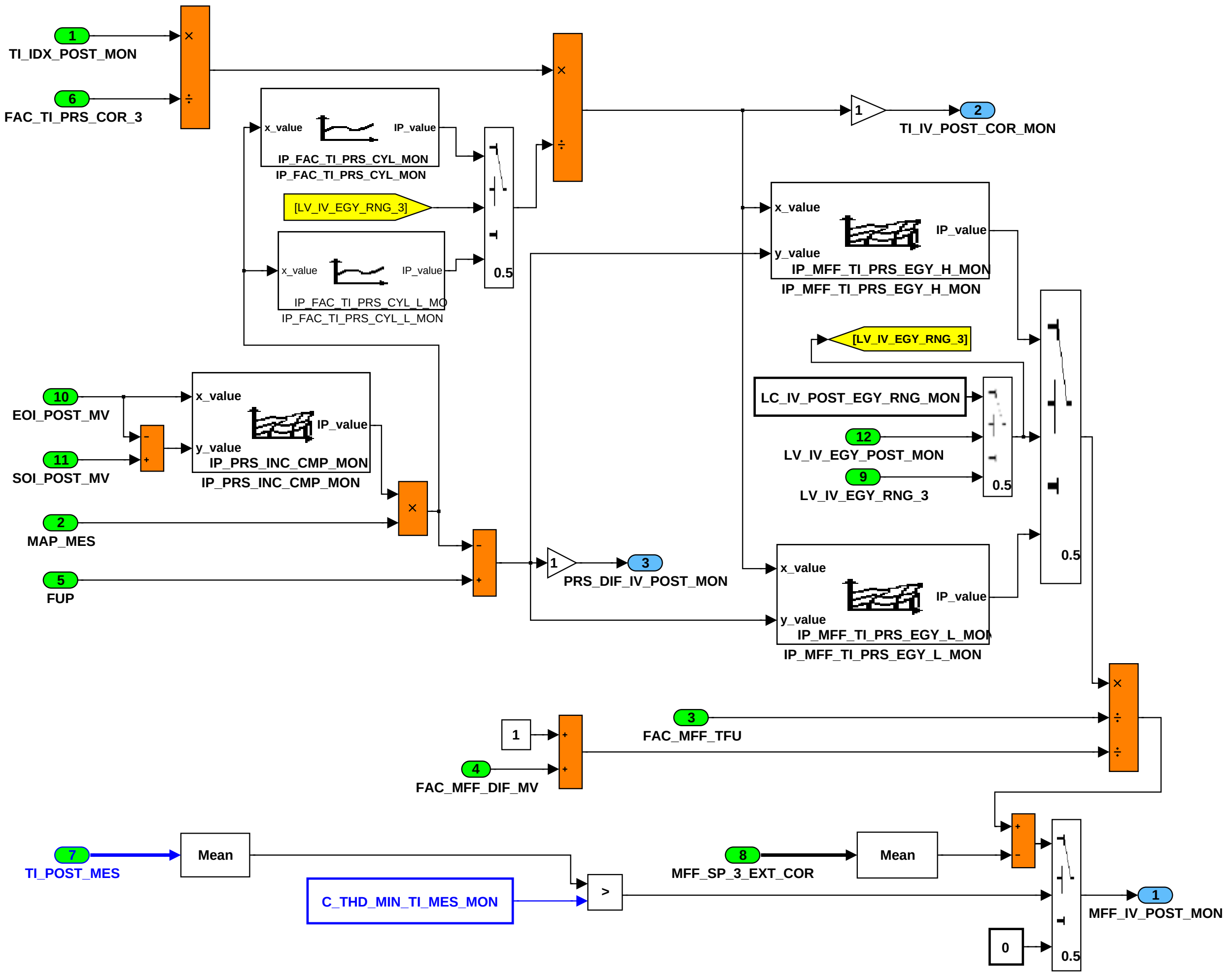


Figure 87 ECM2_DTSYSMFFAC/ CLC_MFF_MON/ MFF_PULSES/ Injection_Pulse_Post

11.22 Actual efficiencies

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
EFF_IGA_HOM_MON	V	0... FFH	0... 1.9921875	7.8125e-3	[-]
ignition efficiency in homogeneous (process monitoring)					
EFF_IGA_HOMS_MON	V	0... FFH	0... 1.9921875	7.8125e-3	[-]
ignition efficiency in multiple homogeneous-stratified (process monitoring)					
EFF_IGA_S_MON	V	0... FFH	0... 1.9921875	7.8125e-3	[-]
ignition efficiency in stratified (process monitoring)					
EFF_LAMB_HOM_MON	V	0... FFH	0... 1.9921875	7.8125e-3	[-]
lambda efficiency in homogeneous (process monitoring)					
EFF_LAMB_HOMS_MON	V	0... FFH	0... 1.9921875	7.8125e-3	[-]
lambda efficiency in homogeneous-stratified (process monitoring)					
EFF_LAMB_S_MON	V	0... FFH	0... 1.9921875	7.8125e-3	[-]
lambda efficiency in stratified homogeneous (process monitoring)					
EFF_TOT_MON	O/V	0... FFH	0... 1.9921875	7.8125e-3	[-]
total actual efficiencies					
IGA_DIF_AV_H_RNG_HOM_MON	V	0... B40H	0... -180	-0.0625	[°CRK]
difference from reference to actual ignition angle in homogeneous (process monitoring)					
IGA_DIF_AV_H_RNG_HOMS_MON	V	0... B40H	0... -180	-0.0625	[°CRK]
difference from reference to actual ignition angle in homogeneous_stratified (process monitoring)					
IGA_DIF_AV_HOM_MON	-	0... B40H	0... -180	-0.0625	[°CRK]
difference from reference to actual ignition angle in homogeneous (process monitoring)					

Input Data:

OPM_AV_MON	N_32_MON	MFF_LAMB_REF_IGA_MON	MFF_TQ_MON
IGA_REF_HOM_COR_EXT_MON	IGA_REF_HOMS_COR_EXT_MON	LV_TQI_MON_ACT_MON	IGA_IGC_H_RNG [NC_CYL_NR]
LAMB_MON			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
IP_EFF_IGA_HOM_MON	8	0... FFH	0... 99.609375	0.390625	[%]
LDPM_IGA_DIF_AV_MON_IP_EFF_MON	8	0... B40H	0... -180	-0.0625	[°CRK]
actual ignition efficiency in homogeneous (process monitoring)					
IP_EFF_IGA_HOMS_MON	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDPM_IGA_DIF_AV_MON_IP_EFF_MON	8	0... B40H	0... -180	-0.0625	[°CRK]
actual ignition efficiency in homogeneous_stratified (process monitoring)					
IP_EFF_LAMB_HOM_MON	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_LAMB_HOM_MON_IP_EFF_MON	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
lambda efficiency in homogeneous (process monitoring)					
IP_EFF_LAMB_HOMS_MON	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_LAMB_HOMS_MON_IP_EFF_MON	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
lambda efficiency in homogeneous-stratified (process monitoring)					
IP_EFF_LAMB_S_MON	8	0... FFH	0... 1.9921875	7.8125e-3	[-]

LDP_LAMB_S_MON_IP_EFF_MON	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
lambda efficiency in stratified (process monitoring)					
IP_FAC_EFF_IGA_MON	6*6	0... FFH	0... 15.9375	0.0625	[-]
LDP_N_32_MON_IP_FAC_EFF_IGA_MON	6	0... FFH	0... 8160	32	[rpm]
LDP_MFF_MON_IP_FAC_EFF_IGA_MON	6	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Factor ignition efficiency (process monitoring)					
IP_IGA_REF_HOM_MON	8*8	0... B40H	-90 ...90	0.0625	[°CRK]
LDPM_N_32_MON_IP_IGA_REF_MON	8	0... FFH	0... 8160	32	[rpm]
LDP_MFF_MON_IP_IGA_REF_HOM_MON	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
reference ignition angle in homogeneous (process monitoring)					
IP_IGA_REF_HOMS_MON	8*8	0... B40H	-90 ...90	0.0625	[°CRK]
LDPM_N_32_MON_IP_IGA_REF_MON	8	0... FFH	0... 8160	32	[rpm]
LDP_MFF_MON_IP_IGA_REF_HOMS_MON	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
reference ignition angle in homogeneous_stratified (process monitoring)					

General Information

Import actions:

ACTION_ECM3_Service15TaskPfm(IN<>)

ACTION_ECM3_Service16TaskPfm(IN<>)

ACTION_ECM3_Service17TaskPfm(IN<>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 3.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service15TaskPfm() and ACTION_ECM3_Service16TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)

ACTION_ECM3_ChkCpl(IN <>, IN <>)

ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: These actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode O and history variables in modules of the Process Monitoring have to be checked by the cyclical RAM test.

Hint Application Conditions Initialisation: for condition see 'Application incidences of process monitoring'

Application Conditions

Initialization: RST
Recurrence: 40MS
Activation: LV_TQI_MON_ACT_MON
Deactivation: !LV_TQI_MON_ACT_MON

Function description

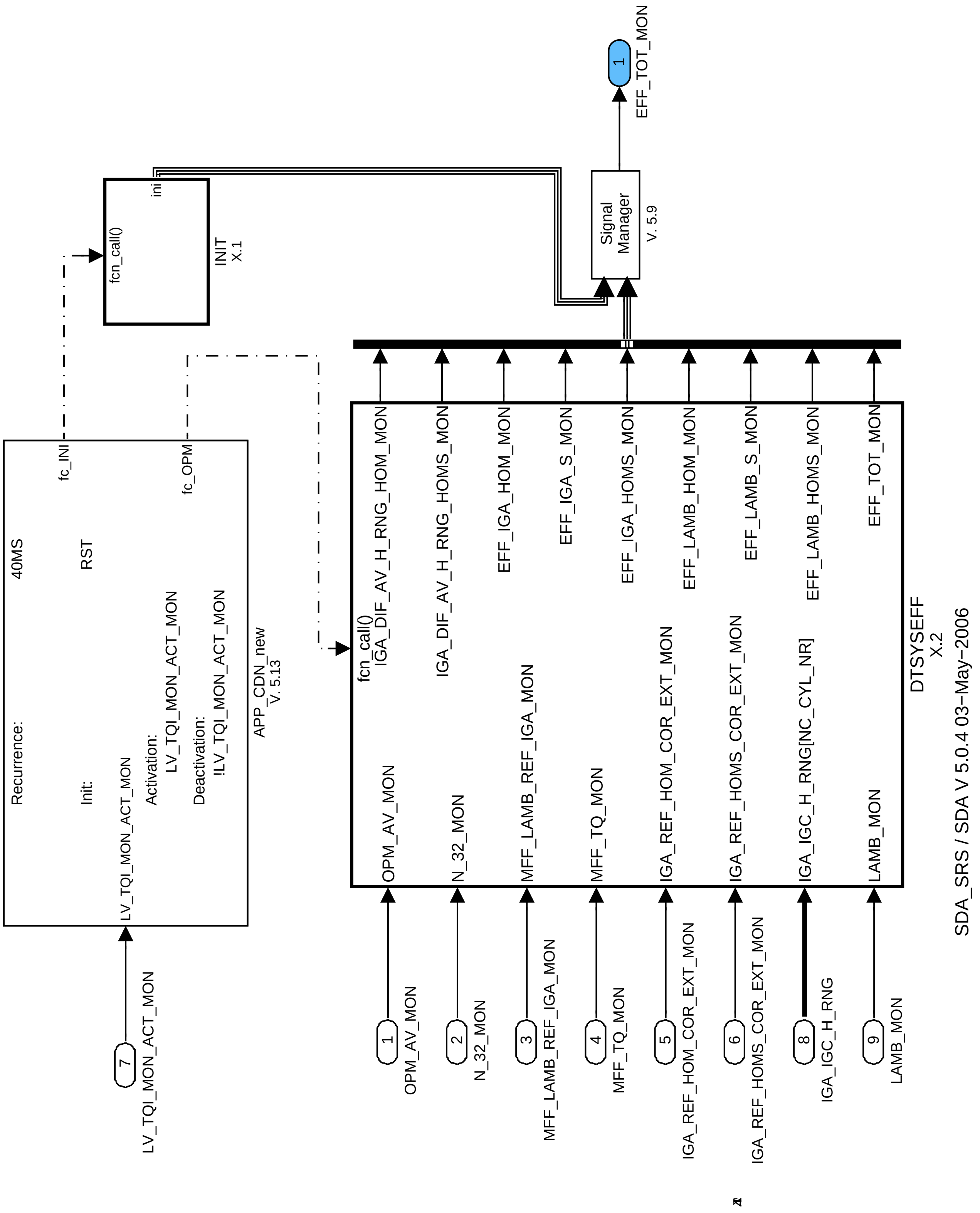


Figure 88:
Path: ECM2_DTSYSEFF

11.22.1 ECM2_DTSYSEFF/INIT

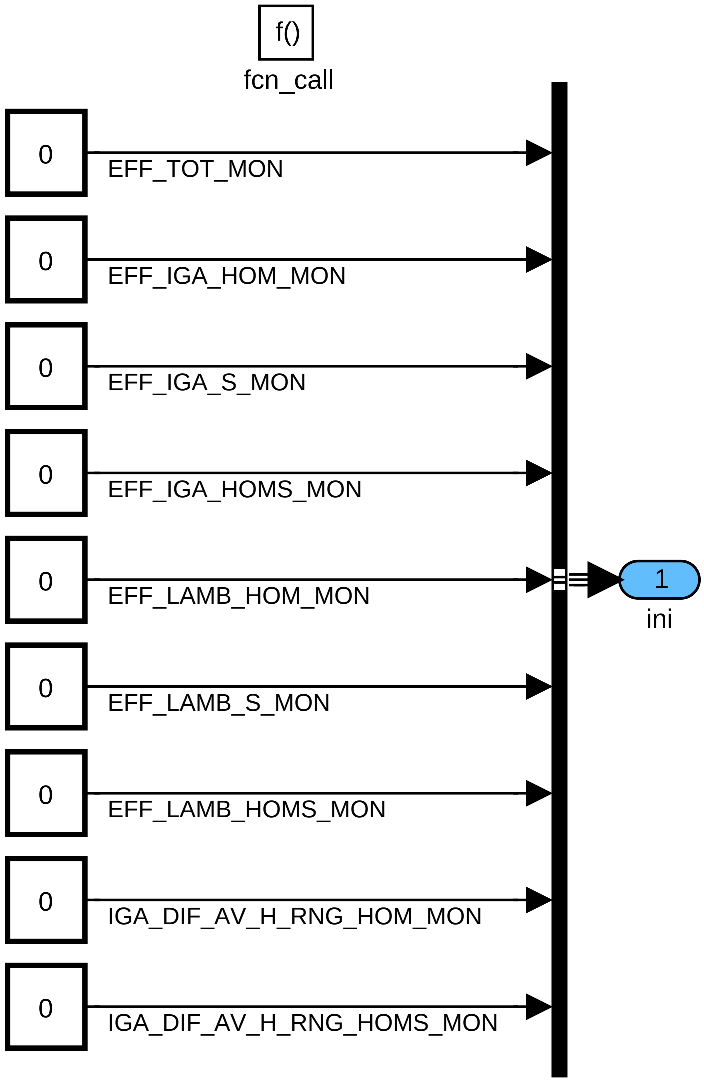


Figure 89:
Path: ECM2_DTSYSEFF/INIT

11.22.2 ECM2_DTSYSEFF/DTSYSEFF

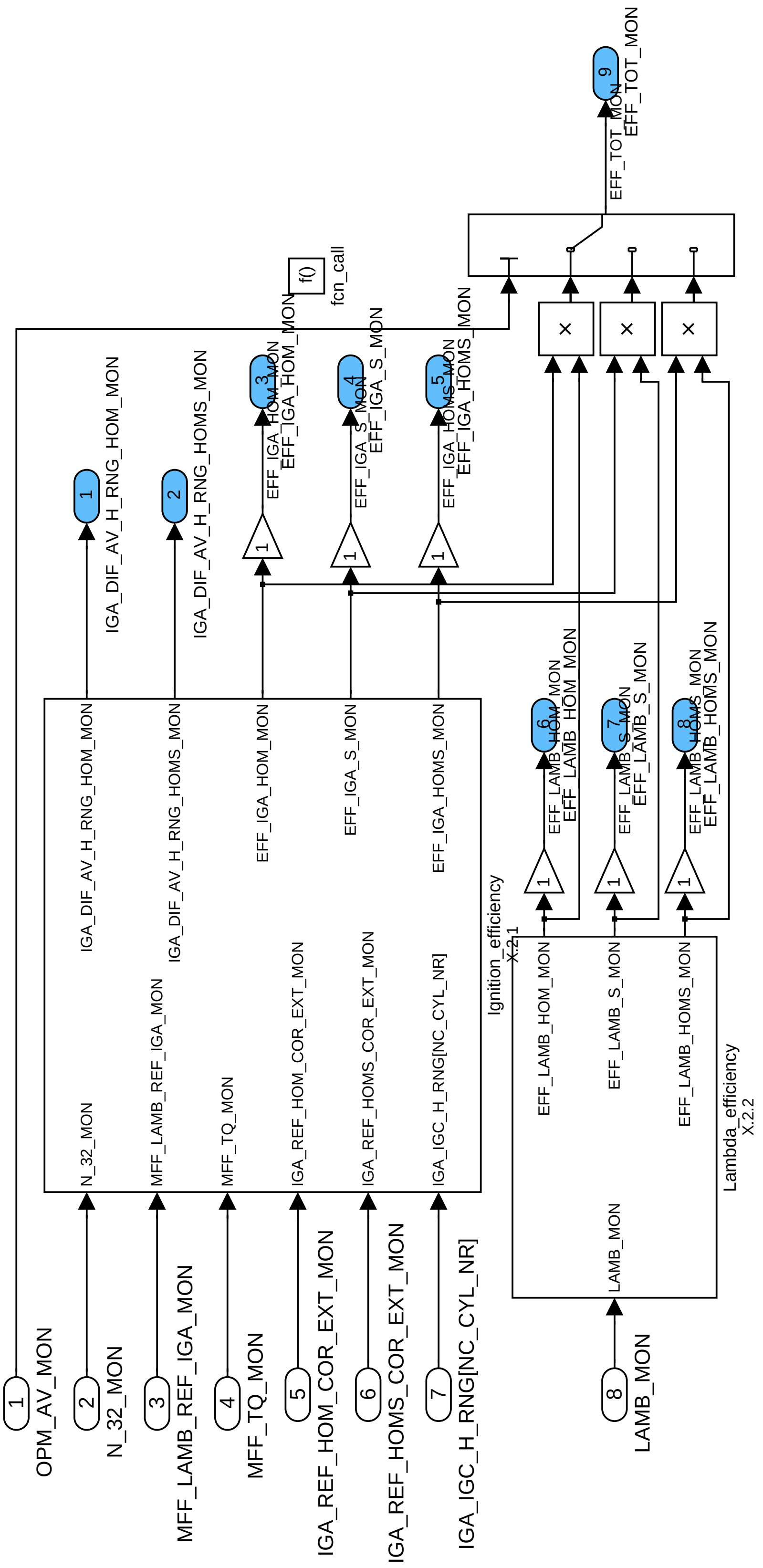


Figure 90:
Path: ECM2_DTSYSEFF/DTSYSEFF

11.22.2.1 ECM2_DTSYSEFF/DTSYSEFF/IGNITION_EFFICIENCY

<CONTENT VIEW="ANY">

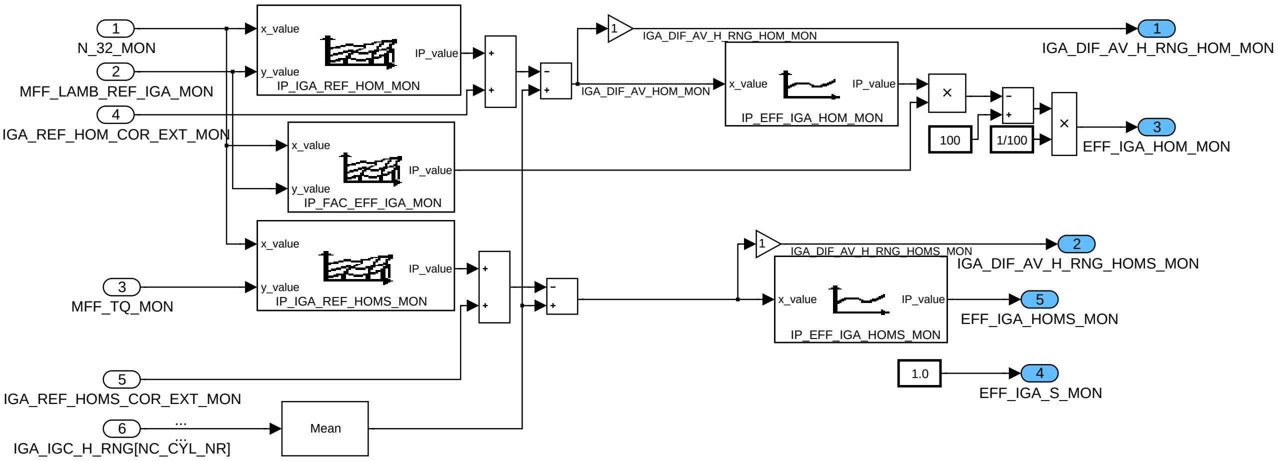


Figure 91:
Path: ECM2_DTSYSEFF/DTSYSEFF/Ignition_efficiency

11.22.2.2 ECM2_DTSYSEFF/DTSYSEFF/LAMBDA_EFFICIENCY

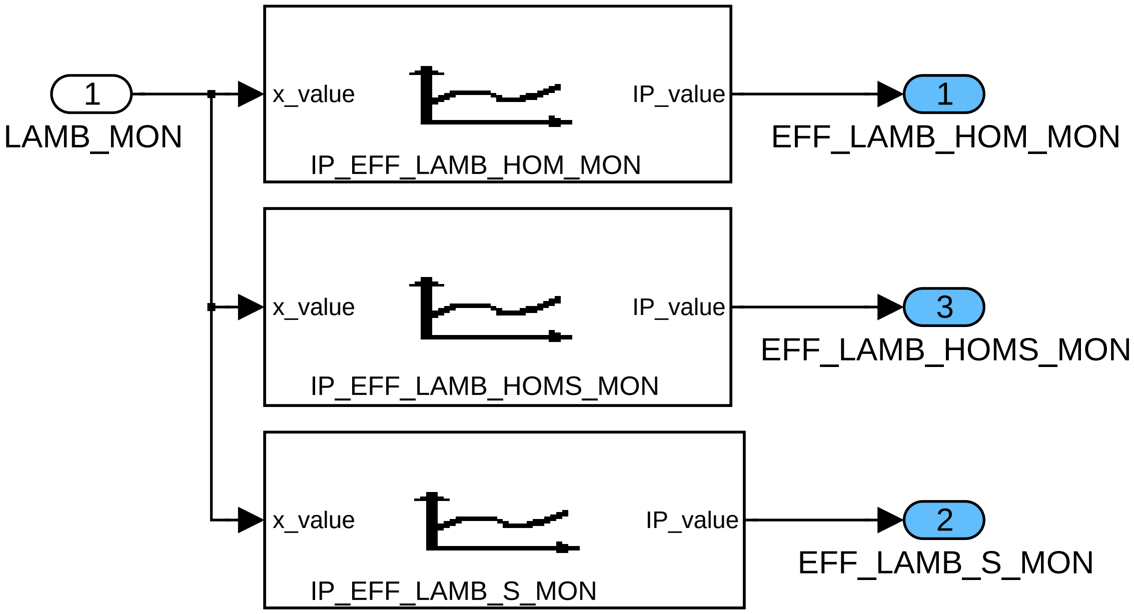


Figure 92:
Path: ECM2_DTSYSEFF/DTSYSEFF/Lambda_efficiency

11.23 Actual indicated engine torque

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQI_AV_MON	V/O	0...FFH	0...510	2	[Nm]
Actual indicated engine torque (process monitoring)					

Input data:

LV_TQI_MON_ACT_MON	MFF_TQ_MON	N_32_MON	EFF_TOT_MON
EFF_MFF_TQ_COR_MON			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQI_CUS_MON	6*8	0...FFH	0...510	2	[Nm]
LDP_N_32_MON_IP_TQI_CUS_MON	6	0...FFH	0...8160	32	[rpm]
LDP_MFF_MON_IP_TQI_CUS_MON	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Customer specific engine torque (process monitoring)					

11.23.1 ECM2_DTSYSTQIAV

Import actions:

ACTION_ECM3_Service3TaskPfm(IN<>)
ACTION_ECM3_Service4TaskPfm(IN<>)
ACTION_ECM3_Service5TaskPfm(IN<>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 4.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service3TaskPfm() and ACTION_ECM3_Service4TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: These actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

General information:

This module calculates the actual indicated engine torque TQI_AV_MON. The map represents a simplified versions of the original map of the corresponding functions in the control level.

Application Condition

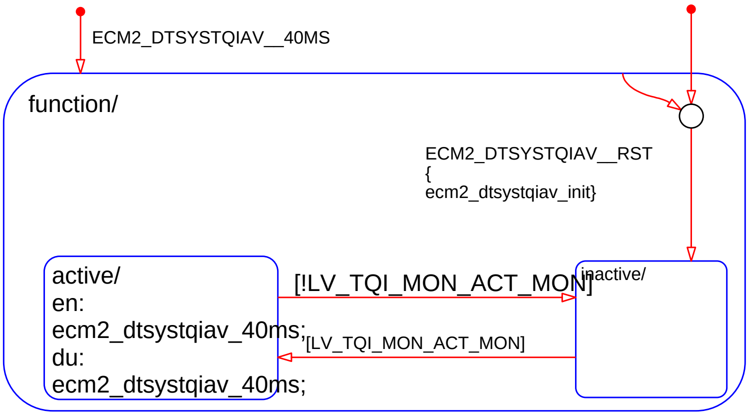


Figure 93 ECM2_DTSYSTQIAV/ APP_CDN/ Chart

Function Description

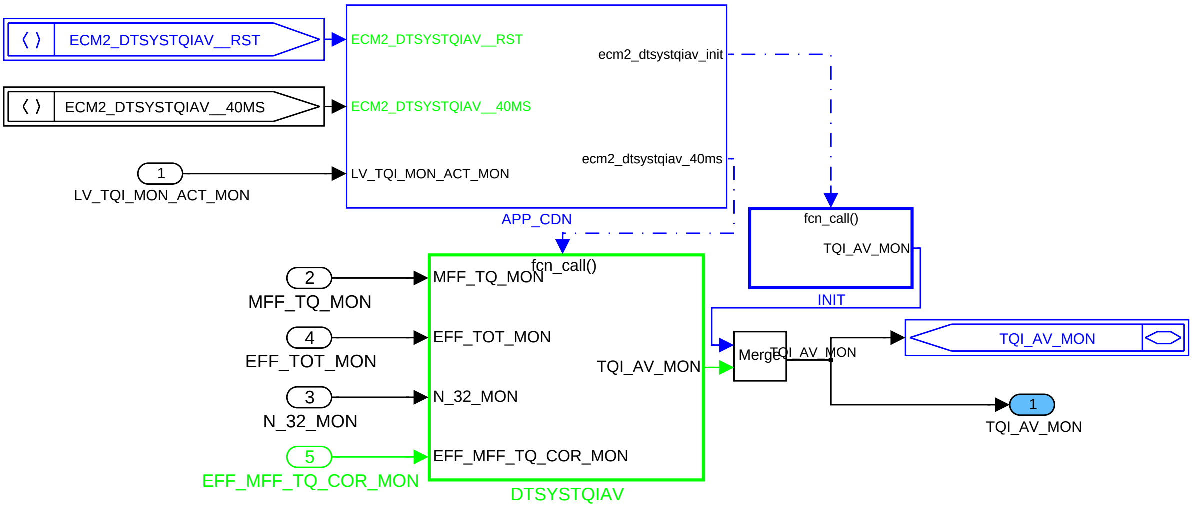


Figure 94 ECM2_DTSYSTQIAV

11.23.1.1 SUBFUNCTION: INIT

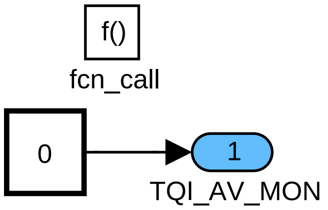


Figure 95 ECM2_DTSYSTQIAV/ INIT

11.23.1.2 SUBFUNCTION: DTSYSTQIAV

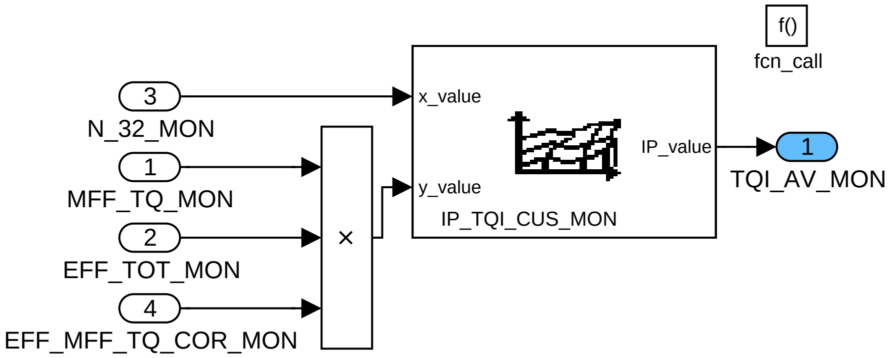


Figure 96 ECM2_DTSYSTQIAV/ DTSYSTQIAV

11.24 Monitoring of actual indicated engine torque

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TQI_AV_MON	V/O	0...1H	0...1	1	[-]
Fault currently present in torque generation					
LV_SWI_AFS_TQI_MON	V/O	0...1H	0...1	1	[-]
Switch to AFS (TQI_DIF above allowed threshold)					
TQI_DIF_SP_MON	V	0...FFH	0...510	2	[Nm]
deviation between torque setpoint and desired indicated engine torque					
TQI_DIF_AV_MON	V	0...FFH	0...510	2	[Nm]
deviation between actual and desired indicated engine torque					
TQI_DIF_MAX_MON	V	0...FFH	0...510	2	[Nm]
permitted deviation between actual and desired indicated engine torque					
ABC_TQI_AV_AFS_MON	V	0...FFH	0...255	1	[-]
Anti-bounce-counter for fault in torque generation (AFS-mode)					
ABC_SWI_AFS_TQI_MON	V	0...FFH	0...255	1	[-]
Anit-bounce-counter for request to AFS (TQI_DIF above allowed threshold)					

Input data:

LV_TQI_MON_ACT_MON	N_32_MON	OPM_AV_MON	TQI_AV_MON
TQI_SP	TQI_SP_MON		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SWI_AFS_TQI_MON	1	0...FFH	0...255	1	[-]
ABC increment for request to AFS (TQI_DIF above allowed threshold)					
C_ABC_INC_TQI_AV_AFS_MON	1	0...FFH	0...255	1	[-]
ABC increment at TQI_DIF above allowed treshold (AFS-mode)					
C_ABC_MAX_SWI_AFS_TQI_MON	1	1...FFH	1...255	1	[-]
Value at which OPM will be switched to AFS at TQI_DIF above allowed treshold					
C_ABC_MAX_TQI_AV_AFS_MON	1	1...FFH	1...255	1	[-]
Value at which TQI_AV_MON error is recognized. when reached (AFS-mode)					
C_GAIN_TQI_DIF_MAX_MON	1	0...FFH	0...63.75	0.25	[-]
Factor for threshold diagnosis with TQI_SP					
IP_TQI_DIF_MAX_MON	4*4	0...FFH	0...510	2	[Nm]
LDP_N_32_IP_TQI_DIF_MAX_MON	4	0...FFH	0...8160	32	[rpm]
LDP_TQI_SP_IP_TQI_DIF_MAX_MON	4	0...FFH	0...510	2	[Nm]
Threshold for diagnosis of engine torque overflow					

11.24.1 DTSYSTQMON

Import actions:

ACTION_ECM3_Service9TaskPfm(IN<>)
ACTION_ECM3_Service10TaskPfm(IN<>)
ACTION_ECM3_Service11TaskPfm(IN<>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 4.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service9TaskPfm() and ACTION_ECM3_Service10TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: These actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

General information:

The objective of this module is the comparison of the desired indicated engine torque TQI_SP_MON (feed-forward path) and the actual indicated engine torque TQI_AV_MON (feedback path). In the fault free case, the actual indicated engine torque should not exceed the desired indicated engine torque. A fault is detected after debouncing, when the actual indicated engine torque exceeds the setpoint value by more than a threshold value TQI_DIF_MAX_MON depending on engine speed and the desired torque. If the operation mode is not equal to AFS a request for AFS will be debounced.

TQI_AV_MON depends on the dynamics of the intake manifold and on the dynamics of the lambda sensor which means that a possible fault cannot be detected before a certain time. Hence for a fast diagnosis TQI_AV_MON is not suitable but can be replaced by TQI_SP which is compared to TQI_SP_MON. In case that TQI_SP_MON differs from TQI_SP by more than a certain value, the error bit LV_ERR_TQI_AV_MON is set.

Application Condition

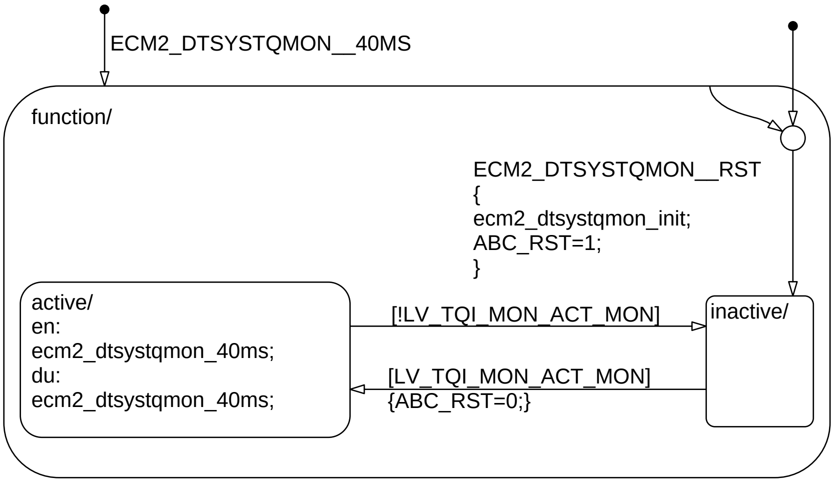


Figure 97 ECM2_DTSYSTQMON/ APP_CDN/ CHART

Function Description

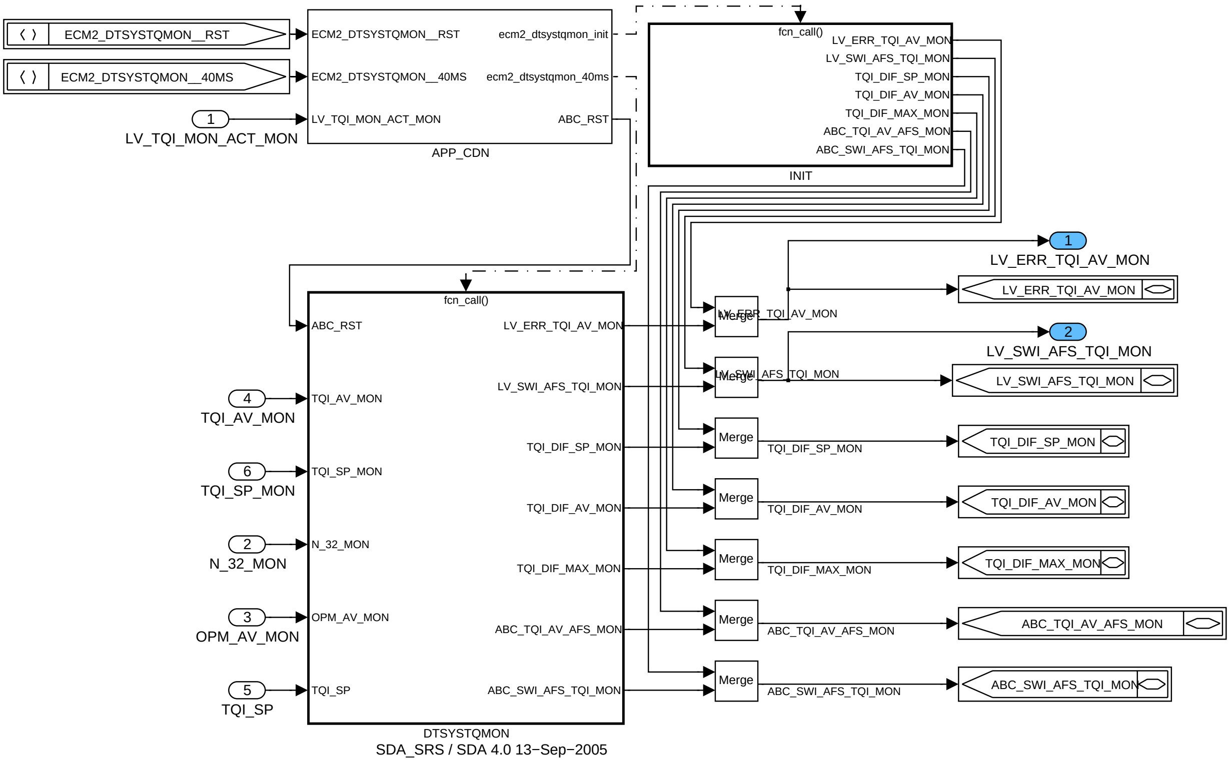


Figure 98 ECM2_DTSYSTQMON

11.24.1.1 SUBFUNCTION: INIT

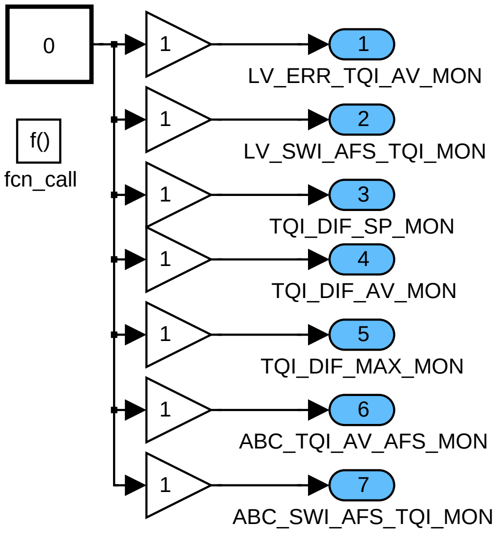


Figure 99 ECM2_DTSYSTQMON/ INIT

11.24.1.2 SUBFUNCTION: DTSYSTQMON

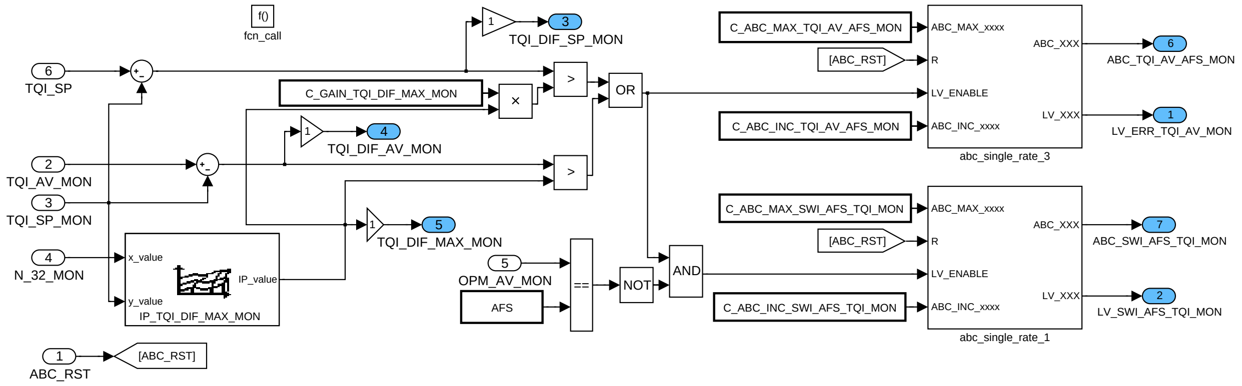


Figure 100 ECM2_DTSYSTQMON/ DTSYSTQMON

11.25 Combustion mode switch request

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SWI_AFS_MON	V/O	0...1H	0...1	1	[-]
Flag for switching to HOM_AFS due to non-HOM_AFS					

Input data:

LV_TQI_MON_ACT_MON	LV_SWI_AFS_TQI_MON	LV_SWI_AFS_MFF_MON	LV_SWI_AFS_LAMB_MON
LV_SWI_AFS_POST_INJ_MON			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_SWI_AFS_MON	1	0...1H	0...1	1	[-]
Switch to force engine into AFS mode (process monitoring)					

11.25.1 ECM2_DTSYSCMB

Import actions:

ACTION_ECM3_Service15TaskPfm(IN<>)
ACTION_ECM3_Service16TaskPfm(IN<>)
ACTION_ECM3_Service17TaskPfm(IN<>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 4.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service15TaskPfm() and ACTION_ECM3_Service16TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: These actions are defined in chapter "Processor Monitoring", subsection "RAM test on MC for processor monitoring". All variables with Mode “O” and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

General information:

This module combines all combustion mode switch requests from level 2 to switch to homogeneous stoichiometric mode in case of faults, which occured in non HOM_AFS mode.

11.25.2 FUNCTION PART: ECM2_DTSYSCMB

Application Condition

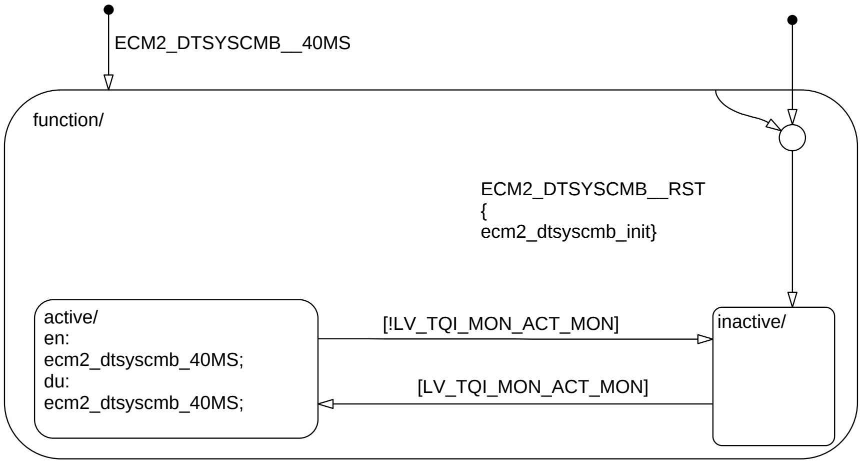
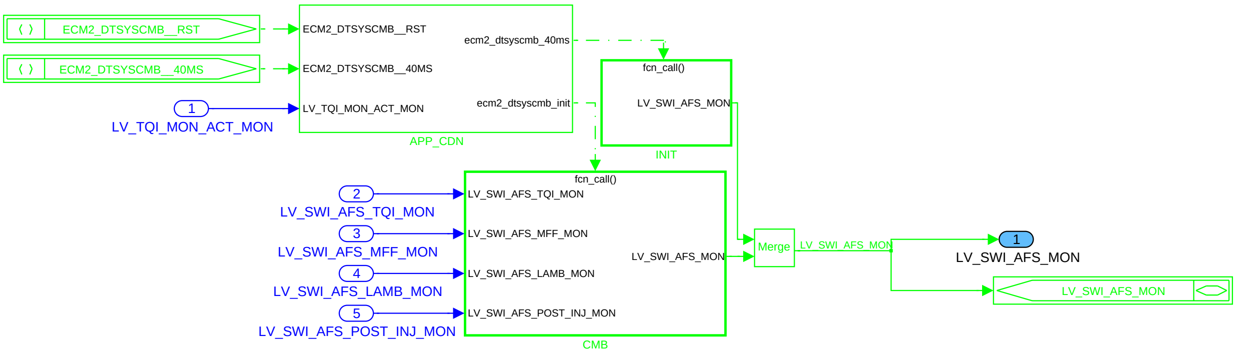


Figure 101 ECM2_DTSYSCMB/ APP_CDN/ CHART

Function Description



SDA_SRS / SDA 3.1 04−Oct−2004

Figure 102 ECM2_DTSYSCMB

11.25.2.1 SUBFUNCTION: INIT

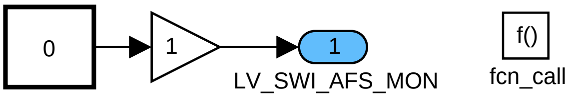


Figure 103 ECM2_DTSYSCMB/ INIT

11.25.2.2 SUBFUNCTION: CMB

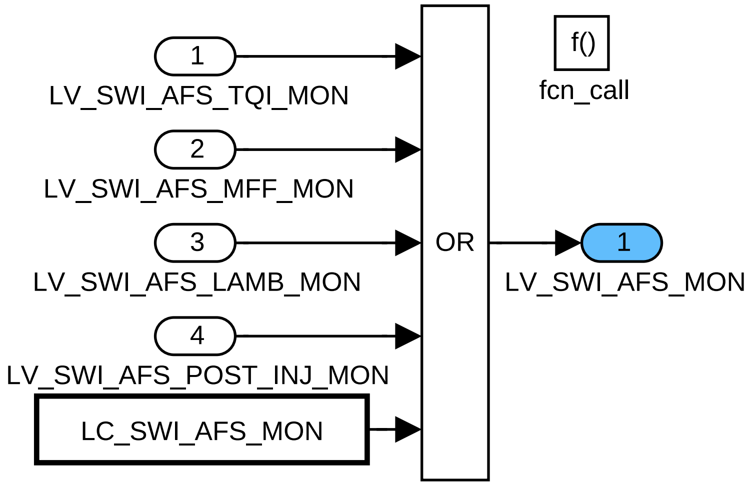


Figure 104 ECM2_DTSYSCMB/ CMB

11.26 Monitoring of engine speed limitation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TQI_N_MAX_MON	V/O	0...1H	0...1	1	[-]
Fault currently present in engine speed limitation					
ABC_TQI_N_MAX_MON	V/O	0...FFH	0...255	1	[-]
Anti-bounce-counter for fault in engine speed limitation					

Input data:

LV_N_LIM_ETC_MON	N_32_MON	PREV_STATE_IV	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TQI_N_MAX_MON	1	0...FFH	0...255	1	[-]
Anti bounce counter increment					
C_ABC_MAX_TQI_N_MAX_MON	1	1...FFH	1...255	1	[-]
Maximum value of anti-bounce counter					
C_N_MAX_MTC_LIH_THD_MON	1	0...FFH	0...8160	32	[rpm]
Speed threshold for the detection of a fault in the engine speed limitation					

Import actions:

ACTION_ECM3_Service3TaskPfm(IN <>)
ACTION_ECM3_Service4TaskPfm(IN <>)
ACTION_ECM3_Service5TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 3.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service3TaskPfm() and ACTION_ECM3_Service4TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

FUNCTION DESCRIPTION:

General information:

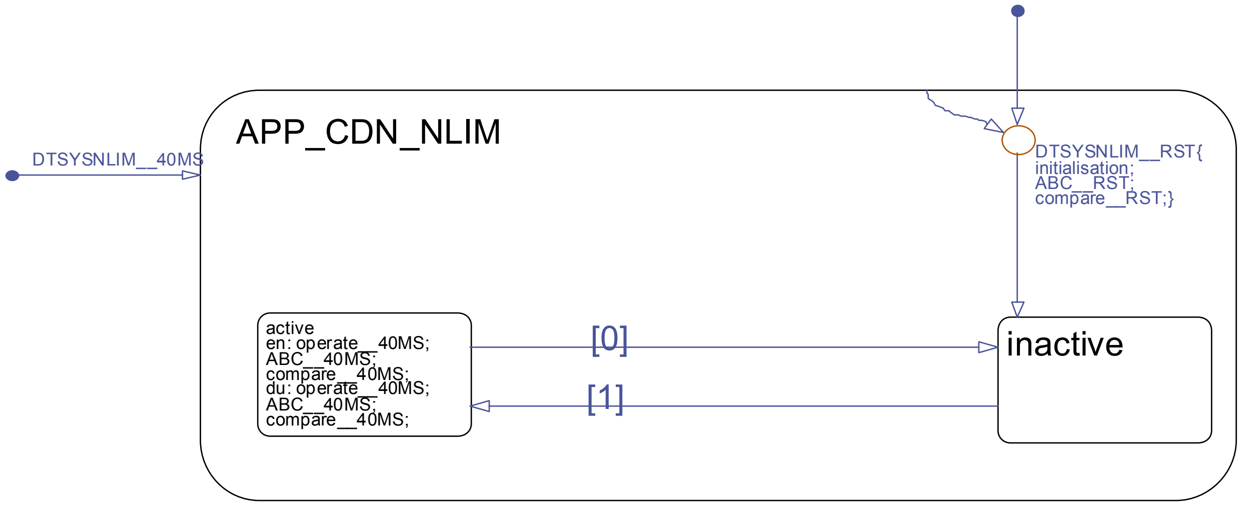
The objective of this module is to verify that the related level 1 function 'Engine speed limitation' works correctly. In case that the engine speed exceeds a certain speed level, the level 1 function activates ETC limp home mode (safety fuel cut-off) to protect the engine.

The level 2 function 'Monitoring of engine speed limitation' considers the status of the injection of all cylinders and the engine speed value and the ETC request flag (LV_N_LIM_ETC_MON). This flag is set to 1 if any function requests limp home activation.

In case that the engine speed exceeds the defined level 2 value C_N_MAX_MTC_LIH_THD_MON and limphome mode is requested (LV_N_LIM_ETC_MON = 1) and injection is active in at least one cylinder (PREV_STATE_IV ≠ 0; fuel cut-off is not active!), the related error bit LV_ERR_TQI_N_MAX_MON is set after having waited a certain debounce time defined by C_ABC_MAX_TQI_N_MAX_MON and C_ABC_INC_TQI_N_MAX_MON.
The injection status has to be checked to avoid that level 2 detects a failure in case that the engine speed exceeds the threshold although fuel cut-off has been put into action. This could happen if the vehicle is running down a hill and the engine speed increases because of the engine directly linked to the transmission.

11.26.1 FUNCTION PART: NLIM

Application Condition



Note: DTSYSNLIM__RST includes the function calls as defined in application incidences.

Function Description

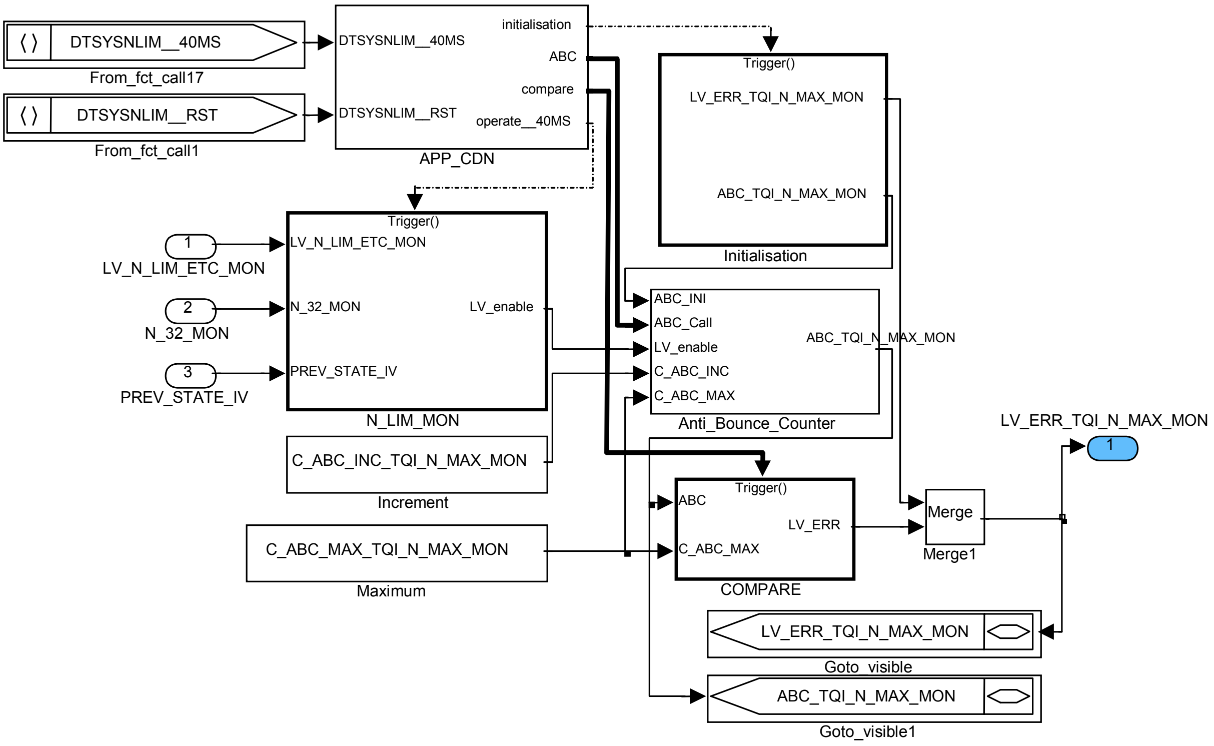


Figure 105 NLIM

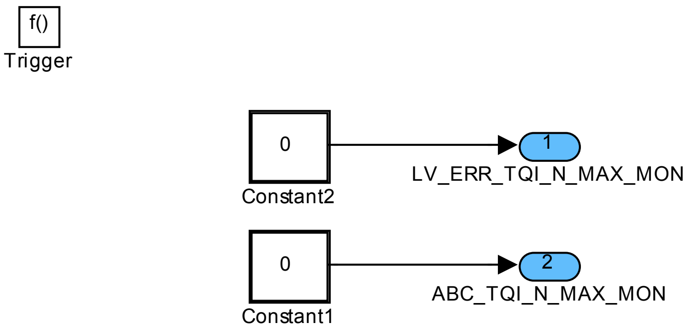


Figure 2 NLIM/ INITIALISATION

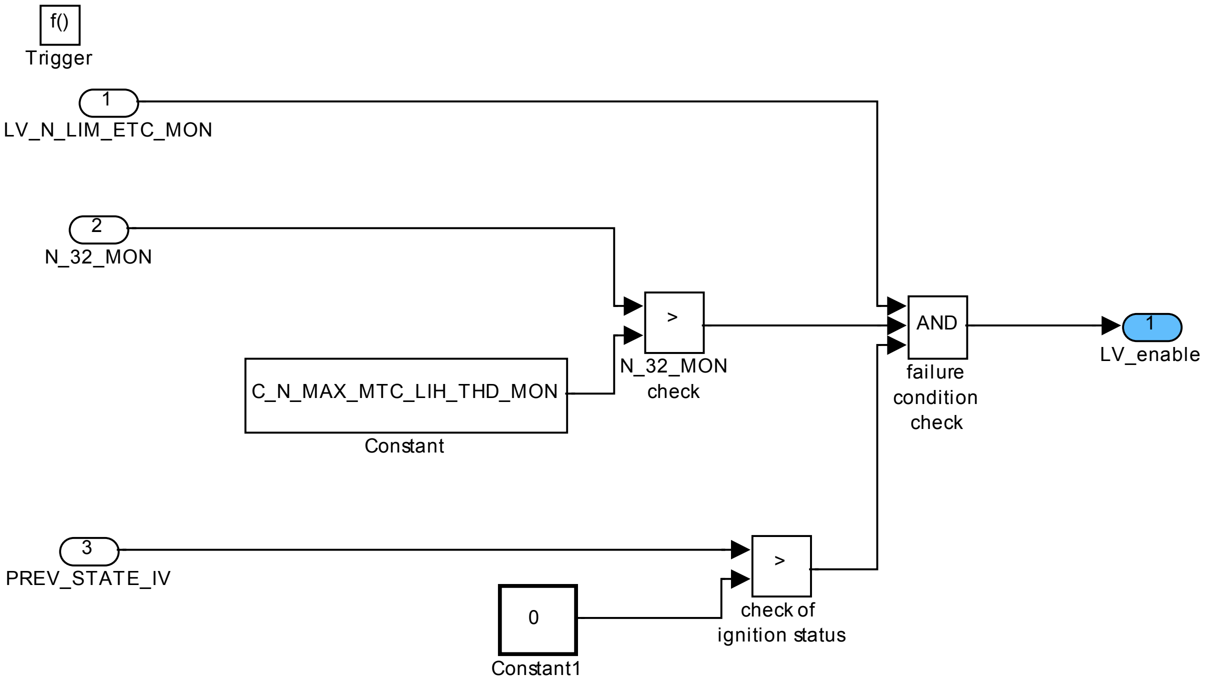


Figure 3 NLIM/ N_LIM_MON

11.27 Processor monitoring by copy of process monitoring (level 2')

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
RESP_MON2	O/V	0...FFFFH	0...6.5535E+4	1	-
Answer of copy of process monitoring (Level 2´)					
EFF_IGA_HOM_MON2	V	0...FFH	0...1.9921875	0.0078125	-
ignition efficiency in homogeneous (process monitoring)					
EFF_IGA_S_MON2	V	0...FFH	0...1.9921875	0.0078125	-
ignition efficiency in stratified (process monitoring)					
EFF_IGA_HOMS_MON2	V	0...FFH	0...1.9921875	0.0078125	-
ignition efficiency in multiple homogeneous-stratified (process monitoring)					
EFF_LAMB_HOMS_MON2	V	0...FFH	0...1.9921875	0.0078125	-
lambda efficiency in homogeneous-stratified (process monitoring)					
EFF_LAMB_HOM_MON2	V	0...FFH	0...1.9921875	0.0078125	-
lambda efficiency in homogeneous (process monitoring)					
EFF_LAMB_S_MON2	V	0...FFH	0...1.9921875	0.0078125	-
lambda efficiency in stratified homogeneous (process monitoring)					
EFF_TOT_MON2	V	0...FFH	0...1.9921875	0.0078125	-
total actual efficiencies					
EFF_MFF_TQ_COR_MON2	V	0...FFH	0...1.9921875	0.0078125	-
efficiency on torque relevant mass fuel flow					
TQI_AV_MON2	V	0...FFH	0...510	2	Nm
Actual indicated engine torque (process monitoring)					
N_32_MON2	V	0...FFH	0...8.16E+3	32	rpm
Engine speed					
TQI_DIF_MAX_MON2	V	0...FFH	0...510	2	Nm
Permitted deviation between actual and desired indicated engine torque					
ABC_TQI_AV_AFS_MON2	V	0...FFH	0...255	1	-
Anti-bounce-counter for fault in torque generation (AFS-mode)					
LV_ERR_TQI_AV_MON2	V	0...1H	0...1	1	-
Fault currently present in torque generation					
ABC_SWI_AFS_TQI_MON2	V	0...FFH	0...255	1	-
Anit-bounce-counter for request to AFS (TQI_DIF above allowed threshold)					
LV_ERR_OPM_AV_MON2	V	0...1H	0...1	1	-
Level2 error for wrong plausibilisation of engine operation mode					
LV_SWI_AFS_TQI_MON2	V	0...1H	0...1	1	-
Switch to AFS (TQI_DIF above allowed threshold)					
LV_DR_OFF_MU_MON2	V	0...1H	0...1	1	-
Request for wrong answers of the level 2´ test calcualtions to the monitoring unit in order to disable MTC and IV					
LV_ERR_TRAN_2_MON2	V	0...1H	0...1	1	-
Flag for temporary fault 2 (together with fault of L2) during last recurrency					
LV_ERR_TQI_N_MAX_MON2	V	0...1H	0...1	1	-
Fault currently present in engine speed limitation					
LV_ERR_TRAN_1_MON2	V	0...1H	0...1	1	-
Flag for temporary fault 1 (exclusive fault of L3) during last recurrency					
TQI_SP_MON_MON2	V	0...FFH	0...510	2	Nm
Desired indicated engine torque					
LV_OFF_IV_N_LIM_ETC_TMP_MON2	V	0...1H	0...1	1	-
Flag for temporary injection cut off during last recurrency					
PV_AV_AD_MON2	V	0...FFH	0...99.609375	0.390625	%
Pedal value adaptation					
FAC_TQ_REQ_CRU_MON2	V	0...FFH	0...1.9921875	0.0078125	-
Torque demand from cruise control					
TQ_MAX_CLU_MON2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Monitored maximum torque at clutch					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LF_4_MON2	V	0...FFH	0...255	1	-
Byte of logical starting values 4					
LV_N_LIM_REQ_MON2	V	0...1H	0...1	1	-
Request for engine speed limitation					
LV_ERR_TPS_MON2	V	0...1H	0...1	1	-
Fault in TPS present					
TQ_MIN_CLU_MON2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Monitored minimum torque at clutch					
TQ_DIF_P_D_IS_MON2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
P- and D- controller output for idle speed control					
TQ_LOSS_MON2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Torque losses					
TQI_REQ_TOT_MON2	V	0...FFH	0...510	2	Nm
Desired non-delayed indicated engine torque					
TQI_INC_EXT_MON2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
External torque request					
TQI_MIN_PU_MON2	V	0...7FFFH	0...1.02397E+3	0.03125	Nm
Torque for trailing throttle					
TQ_DROF_MON2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Torque demand from drive off assistant					
IGA_IGC_H_RNG_MON2[NC_CYL_NR]	V	FA60...5A0H	-90...90	0.0625	°CRK
Actual ignition angle high range					
MFF_TQ_MON2	V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Torque relevant Mass Fuel Flow					
TQI_SP_MON2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Setpoint indicated engine torque (function level)					
OPM_AV_MON2	V	1H 2H 3H 8H	S AFS AFL LIH	1	-
actual engine operation mode (process monitoring)					
PREV_STATE_IV_MON2	V	0...FFH	0...255	1	-
State of injection valves at previous cycle					
MFF_LAMB_REF_IGA_MON2	V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Lambda relevant Mass Fuel Flow for stoichometric conditions corrected by SCAV for EFF_IGA_MON					
LAMB_MON2	V	0...7FFFH	0...31.9990234	9.76563E-4	-
Modeled Lambda					
T_TQI_MON2	V	0...FFH	0...10.2	0.04	s
Dead time for delay of desired indicated engine torque					
IGA_REF_HOM_COR_EXT_MON2	V	FA60...5A0H	-90...90	0.0625	°CRK
Correction of reference ignition angle in homogeneous (for external functions e.g. VVT, EGR...)					
IGA_REF_HOMS_COR_EXT_MON2	V	FA60...5A0H	-90...90	0.0625	°CRK
Correction of reference ignition angle in homogeneous stratified (for external functions e.g. VVT, EGR...)					
LV_TQI_MON_ACT_MON2	V	0...1H	0...1	1	-
Logical variable for torque monitoring active					
LV_N_LIM_ETC_MON2	V	0...1H	0...1	1	-
Engine speed limitation request					
LV_ERR_CONV_MON2	V	0...1H	0...1	1	-
Fault in A/D-conversion present					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_N_32_MON2	V	0...1H	0...1	1	-
Fault in engine speed present					
LV_ERR_TQ_DIF_P_D_IS_MON2	V	0...1H	0...1	1	-
Fault currently present in torque generation, symptom "idle speed controller-PD-part"					
LV_ERR_TQ_DIF_I_IS_MON2	V	0...1H	0...1	1	-
Fault currently present in torque generation, symptom "idle speed controller - I-part"					
LV_ERR_TQ_LOSS_MON2	V	0...1H	0...1	1	-
Fault in torque losses present					
LV_ERR_TQ_MIN_CLU_MON2	V	0...1H	0...1	1	-
Fault currently present in torque generation, symptom `minimum torque at clutch¿					
FAC_TQ_REQ_MON2	V	0...FFH	0...1.9921875	0.0078125	-
Scaling factor for requested torque at clutch from driverand and cruise control					
LV_ERR_TQ_MAX_CLU_MON2	V	0...1H	0...1	1	-
Fault currently present in torque generation, symptom "maximum torque at clutch"					
LV_ERR_PVS_MON2	V	0...1H	0...1	1	-
Fault in pedal value signal present					
LV_ERR_MFF_MON2	V	0...1H	0...1	1	-
Logical variable for error in mass fuel flow signal					
LV_OFF_IV_N_LIM_ETC_MON2	V	0...1H	0...1	1	-
Reversible fuel Cut Off Switch for all cylinders					
LV_SWI_FAC_TQ_REQ_MON2	V	0...1H	0...1	1	-
logical variable for switching to other factors for requested torque at clutch					
LV_ERR_MU_MC_MON2	V	0...1H	0...1	1	-
Fault detected by processor monitoring (continuous)					
LV_ERR_TMP_MU_MC_MON2	V	0...1H	0...1	1	-
Fault detected by processor monitoring (temporary)					
FAC_TQ_REQ_DRIV_MON2	V	0...FFH	0...1.9921875	0.0078125	-
Scaling factor for requested torque at clutch from driver					
LV_OFF_IV_MON2	V	0...1H	0...1	1	-
Request for disable of IV power stage by main controller					
LV_OFF_MTC_MON2	V	0...1H	0...1	1	-
Request for disable of MTC power stage by main controller					
ABC_TQI_N_MAX_MON2	V	0...FFH	0...255	1	-
Anti bounce counter for fault in engine speed limitation					
TQI_DIF_SP_MON2	V	0...FFH	0...510	2	Nm
deviation between torque setpoint and desired indicated engine torque					
TQI_DIF_AV_MON2	V	0...FFH	0...510	2	Nm
deviation between actual and desired indicated engine torque					
IGA_DIF_AV_H_RNG_HOM_MON2	V	0...B40H	0...-180	-0.0625	°CRK
difference from reference to actual ignition angle in homogeneous (process monitoring)					
IGA_DIF_AV_H_RNG_HOMS_MON2	V	0...B40H	0...-180	-0.0625	°CRK
difference from reference to actual ignition angle in homogeneous-stratified (process monitoring)					

Input data:

NC_CYL_NR	NC_TEST_REC_IDX_MAX_MON2	TEST_REC_IDX_MON2	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SWI_AFS_TQI_MON2	1	0...FFH	0...255	1	-
ABC increment for request to AFS (TQI_DIF above allowed threshold)					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TQI_AV_AFS_MON2	1	0...FFH	0...255	1	-
ABC increment at TQI_DIF above allowed treshold (AFS-mode)					
C_ABC_INC_TQI_N_MAX_MON2	1	0...FFH	0...255	1	-
Anti bounce counter increment					
C_ABC_MAX_SWI_AFS_TQI_MON2	1	1...FFH	1...255	1	-
Value at which OPM will be switched to AFS at TQI_DIF above allowed treshold					
C_ABC_MAX_TQI_AV_AFS_MON2	1	1...FFH	1...255	1	-
Value at which TQI_AV_MON error is recognized. when reached (AFS-mode)					
C_ABC_MAX_TQI_N_MAX_MON2	1	1...FFH	1...255	1	-
Maximum value of anti-bounce counter					
C_CRLC_TQI_SP_MON2	1	0...FFH	0...0.99609375	0.00390625	-
filter constant for realization of first order filter					
C_GAIN_TQI_DIF_MAX_MON2	1	0...FFH	0...63.75	0.25	-
Factor for threshold diagnosis with TQI_SP					
C_N_MAX_MTC_LIH_THD_MON2	1	0...FFH	0...8.16E+3	32	rpm
Speed threshold for the detection of a fault in the engine speed limitation					
C_STATE_RESP_MON2_RST_ACT_MON2	1	0...FFH	0...255	1	-
Constant for reset activation via L2´					
C_T_TQI_MON2	1	0...FFH	0...10.2	0.04	s
Dead time for torque decrease					
ID_ABC_SWI_AFS_TQI_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...255	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Anit-bounce-counter for request to AFS (TQI_DIF above allowed threshold)					
ID_ABC_TQI_AV_AFS_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...255	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Anti-bounce-counter for fault in torque generation (AFS-mode)					
ID_ABC_TQI_N_MAX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...255	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Anti-bounce counter for monitoring of engine speed limitation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_EFF_MFF_TQ_COR_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...1.9921875	0.0078125	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
efficiency on torque relevant mass fuel flow					
ID_FAC_TQ_REQ_CRU_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...1.9921875	0.0078125	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Torque demand from cruise control					
ID_IGA_IGC_H_RNG_MON2	NC_TEST_REC_IDX_MAX_MON2	FA60...5A0H	-90...90	0.0625	°CRK
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Actual ignition angle					
ID_IGA_REF_HOMS_COR_EXT_MON2	NC_TEST_REC_IDX_MAX_MON2	FA60...5A0H	-90...90	0.0625	°CRK
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Correction of reference ignition angle in homogeneous stratified (for external functions e.g. VVT, EGR...)					
ID_IGA_REF_HOM_COR_EXT_MON2	NC_TEST_REC_IDX_MAX_MON2	FA60...5A0H	-90...90	0.0625	°CRK
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Correction of reference ignition angle in homogeneous (for external functions e.g. VVT, EGR...)					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_LAMB_MON2	NC_TEST_REC_IDX_MAX_MON2	0...7FFFH	0...31.9990234	9.76563E-4	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Modeled Lambda					
ID_LF_1_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...255	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Table of logical starting values 1					
ID_LF_2_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...255	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Table of logical starting values 2					
ID_LF_3_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...255	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Table of logical starting values 3					
ID_LF_4_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...255	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Table of logical starting values 4					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_MFF_LAMB_REF_IGA_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Lambda relevant Mass Fuel Flow for stoichometric conditions corrected by SCAV for EFF_IGA_MON					
ID_MFF_TQ_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Torque relevant Mass Fuel Flow					
ID_N_32_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...8.16E+3	32	rpm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Engine speed (monitoring level)					
ID_OPM_AV_MON2	NC_TEST_REC_IDX_MAX_MON2	1H 2H 3H 8H	S AFS AFL LIH	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
actual engine operation mode (process monitioring)					
ID_PREV_STATE_IV_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...255	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
State of injection valves at previous cycle					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_PV_AV_AD_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...99.609375	0.390625	%
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Pedal value adaptation					
ID_RESP_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFFFH	0...6.5535E+4	1	-
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Assignment table for answer decoder of copy of process monitoring					
ID_TQI_INC_EXT_MON2	NC_TEST_REC_IDX_MAX_MON2	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
External torque request					
ID_TQI_MIN_PU_MON2	NC_TEST_REC_IDX_MAX_MON2	0...7FFFH	0...1.02397E+3	0.03125	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Torque for trailing throttle					
ID_TQI_SP_MON2	NC_TEST_REC_IDX_MAX_MON2	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Setpoint indicated engine torque					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_TQI_SP_MON_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...510	2	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Resulting desired indicated engine torque, old value					
ID_TQ_DIF_P_D_IS_MON2	NC_TEST_REC_IDX_MAX_MON2	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
P- and D- controller output for idle speed control					
ID_TQ_DROF_MON2	NC_TEST_REC_IDX_MAX_MON2	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
TQ request for drive off assistance					
ID_TQ_LOSS_MON2	NC_TEST_REC_IDX_MAX_MON2	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Torque losses					
ID_TQ_MAX_CLU_MON2	NC_TEST_REC_IDX_MAX_MON2	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Monitored maximum torque at clutch					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_TQ_MIN_CLU_MON2	NC_TEST_REC_IDX_MAX_MON2	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Monitored minimum torque at clutch					
ID_T_TQI_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FFH	0...10.2	0.04	s
LDPM_TEST_REC_IDX_MON2	NC_TEST_REC_IDX_MAX_MON2	0...FH	0...15	1	-
Dead time for delay of desired indicated engine torque, old value					
IP_EFF_IGA_HOMS_MON2	8	0...FFH	0...1.9921875	0.0078125	-
LDPM_IGA_DIF_AV_MON2_IP_MON2	8	0...B40H	0...-180	-0.0625	°CRK
actual ignition efficiency in homogeneous_stratified (process monitoring)					
IP_EFF_IGA_HOM_MON2	8	0...FFH	0...99.609375	0.390625	%
LDPM_IGA_DIF_AV_MON2_IP_MON2	8	0...B40H	0...-180	-0.0625	°CRK
actual ignition efficiency in homogeneous (process monitoring)					
IP_EFF_LAMB_HOMS_MON2	8	0...FFH	0...1.9921875	0.0078125	-
LDP_LAMB_HOMS_MON2_IP_EFF_MON2	8	0...7FFFH	0...31.9990234	9.76563E-4	-
lambda efficiency in homogeneous-stratified (process monitoring)					
IP_EFF_LAMB_HOM_MON2	8	0...FFH	0...1.9921875	0.0078125	-
LDP_LAMB_HOM_MON2_IP_EFF_MON2	8	0...7FFFH	0...31.9990234	9.76563E-4	-
lambda efficiency in homogeneous (process monitoring)					
IP_EFF_LAMB_S_MON2	8	0...FFH	0...1.9921875	0.0078125	-
LDP_LAMB_S_MON2_IP_EFF_MON2	8	0...7FFFH	0...31.9990234	9.76563E-4	-
lambda efficiency in stratified (process monitoring)					
IP_FAC_EFF_IGA_MON2	6x6	0...FFH	0...15.9375	0.0625	-
LDP_N_32_MON2_IP_FAC_EFF_MON2	6	0...FFH	0...8.16E+3	32	rpm
LDP_MFF_MON2_IP_FAC_EFF_MON2	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Factor ignition efficiency (process monitoring)					
IP_FAC_TQ_REQ_DRIV_1_MON2	8x8	0...FFH	0...1.9921875	0.0078125	-
LDPM_N_32_MON2_IP_FAC_TQ_MON2	8	0...FFH	0...8.16E+3	32	rpm
LDPM_PV_AV_AD_MON2_IP_FAC_MON2	8	0...FFH	0...99.609375	0.390625	%
Scaling factor for requested torque at clutch from driver in other mode					
IP_FAC_TQ_REQ_DRIV_MON2	8x8	0...FFH	0...1.9921875	0.0078125	-
LDPM_N_32_MON2_IP_FAC_TQ_MON2	8	0...FFH	0...8.16E+3	32	rpm
LDPM_PV_AV_AD_MON2_IP_FAC_MON2	8	0...FFH	0...99.609375	0.390625	%
Scaling factor for requested torque at clutch from driver					
IP_IGA_REF_HOMS_MON2	8x8	0...B40H	-90...90	0.0625	°CRK
LDPM_N_32_MON2_IP_IGA_REF_MON2	8	0...FFH	0...8.16E+3	32	rpm
LDP_MFF_MON2_IP_IGA_HOMS_MON2	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
reference ignition angle in homogeneous-stratified (process monitoring)					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_IGA_REF_HOM_MON2	8x8	0...B40H	-90...90	0.0625	°CRK
LDPM_N_32_MON2_IP_IGA_REF_MON2	8	0...FFH	0...8.16E+3	32	rpm
LDP_MFF_MON2_IP_IGA_HOM_MON2	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
reference ignition angle in homogeneous (process monitoring)					
IP_TQI_CUS_MON2	6x8	0...FFH	0...510	2	Nm
LDP_N_32_MON2_IP_TQI_CUS_MON2	6	0...FFH	0...8.16E+3	32	rpm
LDP_MFF_MON2_IP_TQI_CUS_MON2	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Customer specific engine torque (process monitoring)					
IP_TQI_DIF_MAX_MON2	4x4	0...FFH	0...510	2	Nm
LDP_N_32_MON2_IP_TQI_DIF_MON2	4	0...FFH	0...8.16E+3	32	[rpm]
LDP_TQI_SP_MON_MON2_IP_MON2	4	0...FFH	0...510	2	[Nm]
Threshold for diagnosis of engine torque overflow					

11.27.1 ECM2_DTSYSCOPPRHPDI

Attention: All Level 2' calibration data are set to INVISIBLE and NOT TUNABLE epxect for C_STATE_RESP_MON2_RST_ACT_MON2. This is necessary in order to get a more robust calibration process in case of software or calibration data updates. C_STATE_RESP_MON2_RST_ACT_MON2 has to be visible and tunable in the calibration system in order to generate an L2' reset for test cases.

General information:

A copy of the main part of the process monitoring (level 2) is built (called level 2’) and used as a set of test calculations. The target is to verify the correct behaviour of the process monitoring at any time using predefined test data and comparing the resulting answer to the expected answer stored in a predefined table. Moreover there is a special calculation routine responsible for checking the correct data transfer between registers and stack in case of interrupts during the calculation.

This processor monitoring function includes the monitoring functions described in the table below. However, no real fault reaction like disabling of throttle etc. is released.

Since the answers of these test calculations will be compared to the predefined table ID_RESP_MON2 (see processor monitoring), it is important to generate a different answer for every index of starting values (NC_TEST_REC_IDX_MAX_MON2 different answers overall). The list of data points for the starting values must contain the indices 0...NC_TEST_REC_IDX_MAX_MON2-1 in ascending order. For details see also "Processor Monitoring" and "Processor Monitoring (Appl.Inc.)".

Table of software-code of level 2 modules that is included in level 2’:

Module:	**Specification:**
Desired indicated engine torque	`30E00K01.00I`
Actual efficiencies	`30E01401.00J`
Actual indicated engine torque	`30E00L01.00E`
Monitoring of actual indicated engine torque	`30E00M01.00E`
Monitoring of engine speed limitation	`30E00I01.00F`

Fault reaction of process monitoring	30E00G01.00J

The software-code of the level 2’ sub-modules is an exact copy of software-code of the level 2 modules listed in the table above, with exceptions described in the following sub chapters.

The following Application Condition takes into consideration that the level 2' module is triggered by Processor Monitoring (level 3).

Application Condition

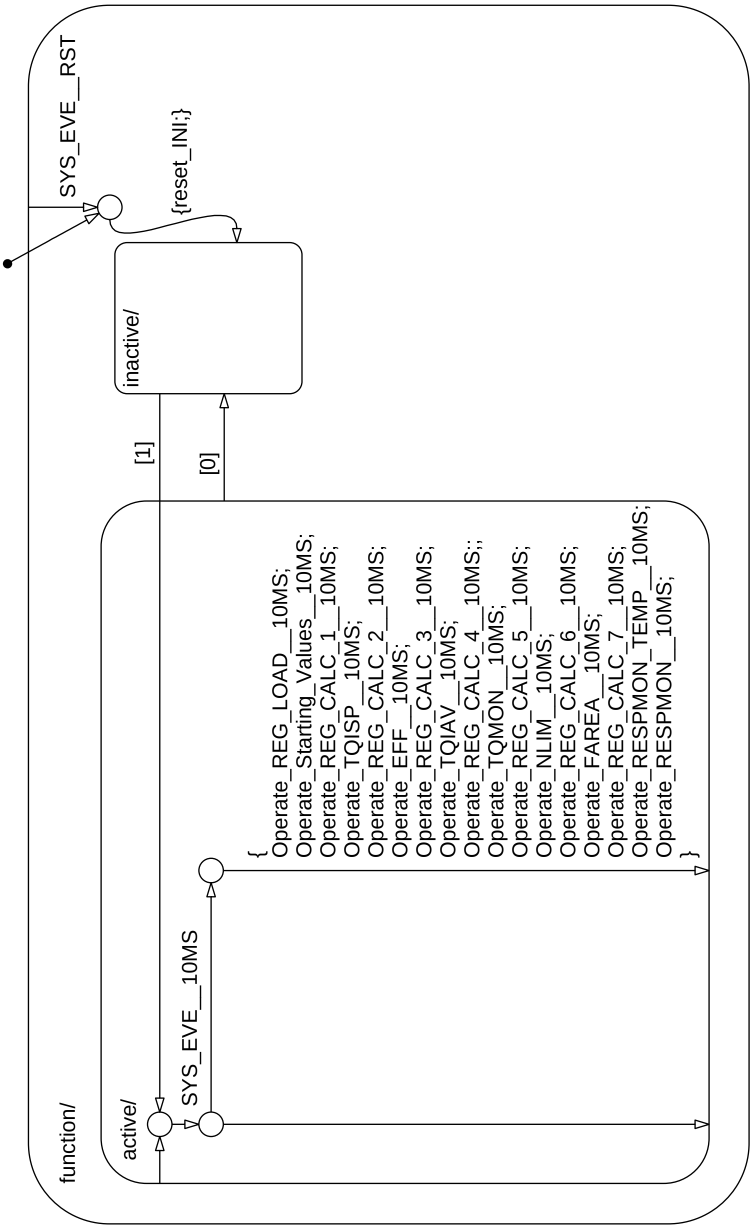
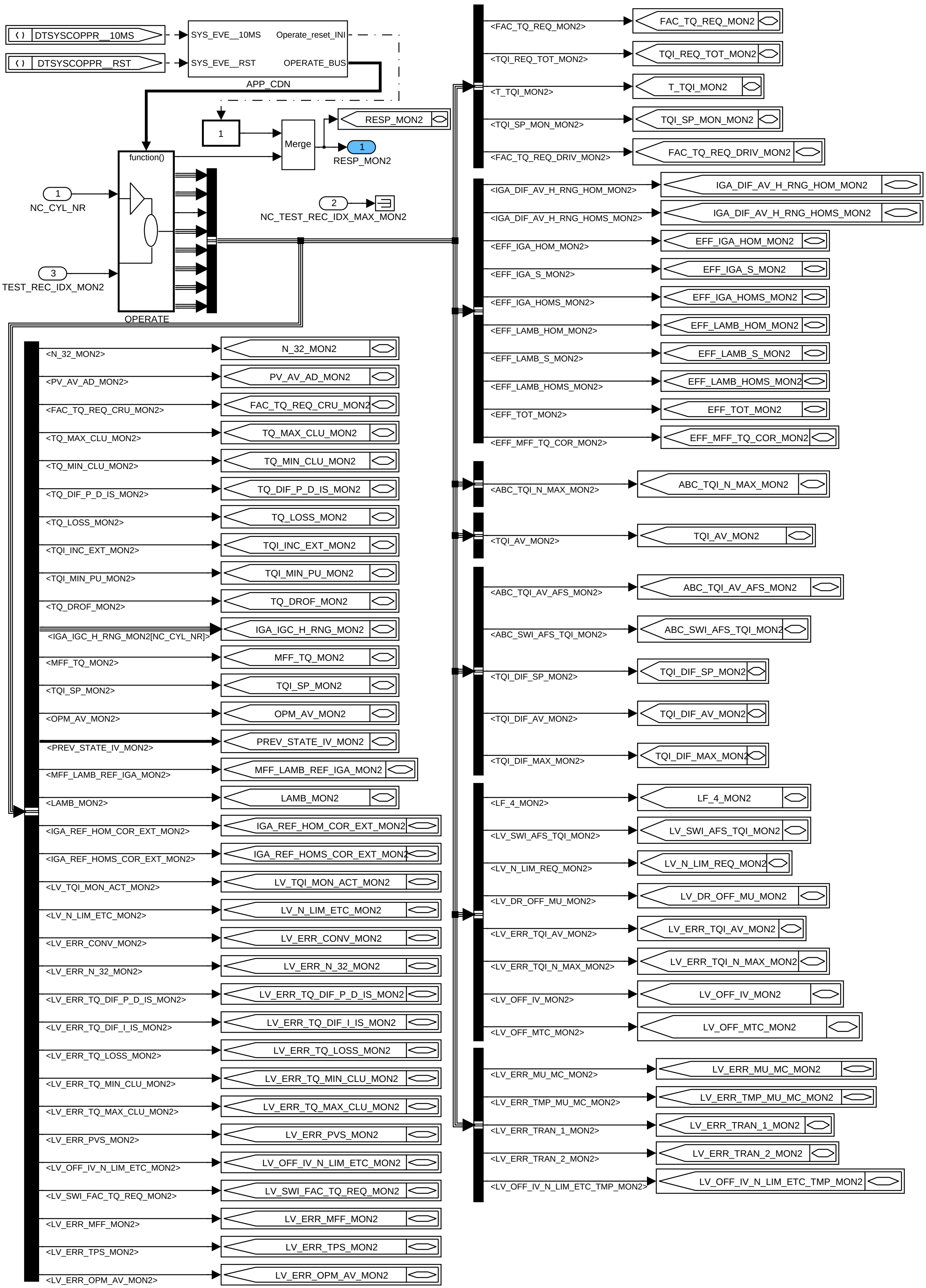


Figure 106 ECM2_DTSYSCOPPRHPDI/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 03−May−2006

Chapter	Baseline		Include File
ECU Process Monitoring	4DC3940S		30E00O02.00L
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Pages
2001 of 9643

Figure 107 ECM2_DTSYSCOPPRHPDI

11.27.1.1 OPERATE

General information:

Based on the number of the question asked by the monitoring unit a corresponding set of starting values for the test calculation is selected from the lists of starting values.

The values of LDPM_TEST_REC_IDX_MON2 are fixed. They depend on the maximum NC_TEST_REC_IDX_MAX_MON2 of the generated index TEST_REC_IDX_MON2. The hex and physical limits are defined with static values which depend on the possible number of questions in the HW.

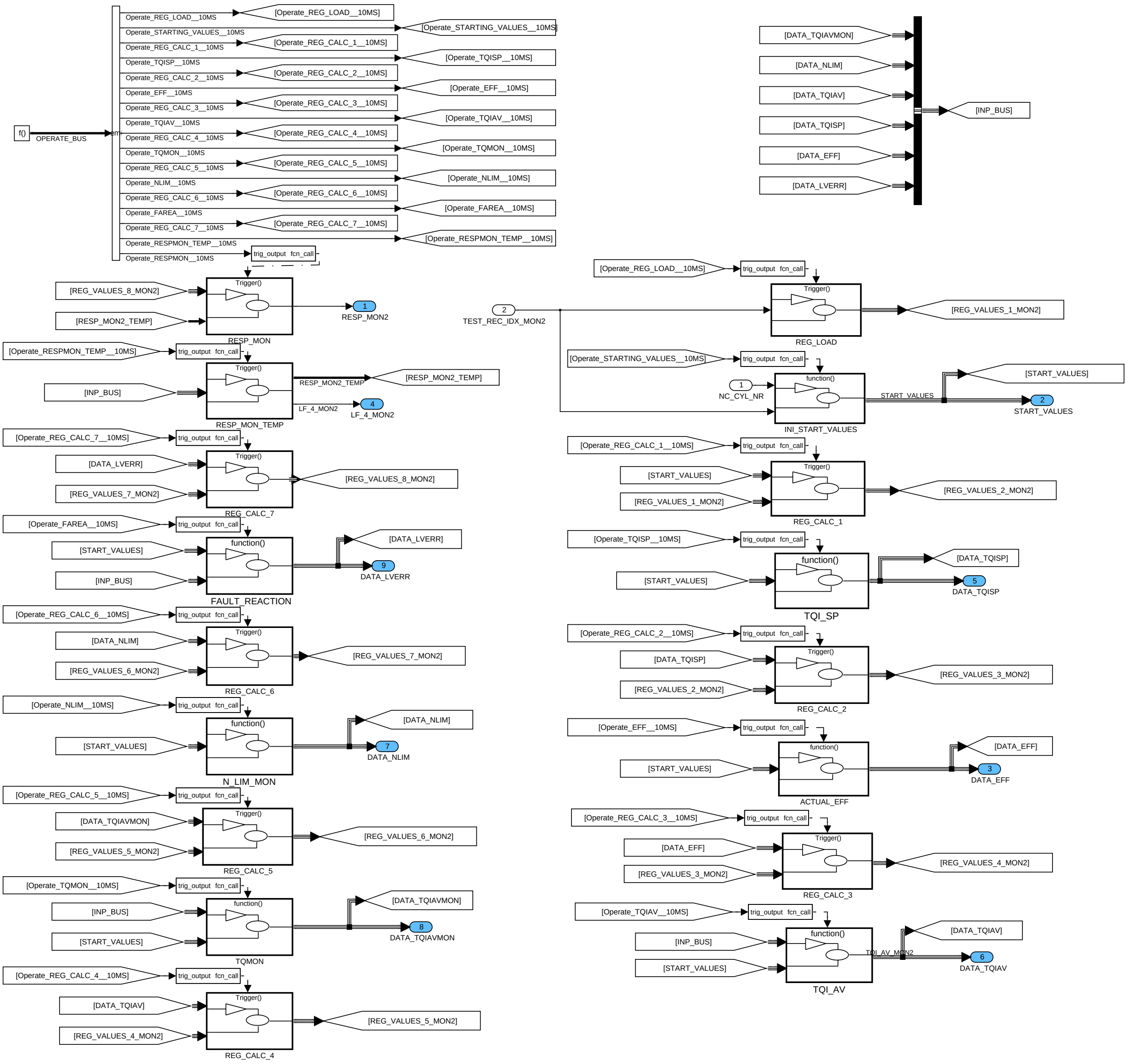


Figure 108 ECM2_DTSYSCOPPRHPDI/ OPERATE

Register LOAD for checking Interrupt routine

In order to simulate an interrupt at first all registers and some cells of the user stack (if available) will be loaded with some defined initial values (see figure below).

The data type for these values has to be signed 32 bit. It is not allowed to use arrays for the load values: after the first calculation every register or user stack value has to be stored separately.

After the calculation, the function call of the first module out of the L2 copied functions shall be carried out; thereby, the current register values will be copied onto the stack.

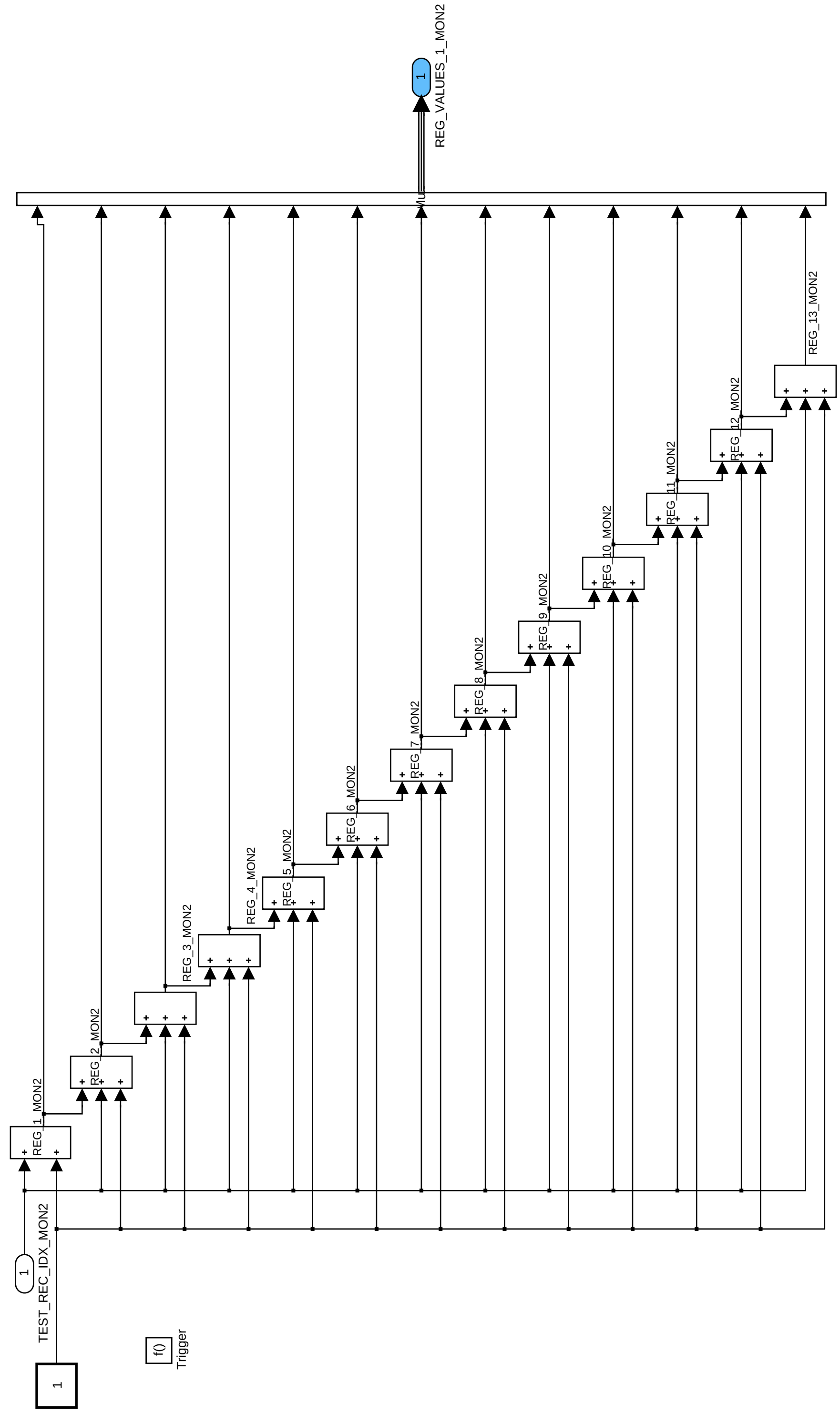


Figure 109 ECM2_DTSYSCOPPRHPDI/ OPERATE/ REG_LOAD

ECM2_DTSYSCOPPRHPDI/OPERATE/INI_START_VALUES

Initialization of variables from the lists of starting values

In order to ensure an unambiguous final result for RESP_MON2 the values

TQI_AV_MON2, TQI_REQ_TOT_MON2, EFF_TOT_MON2 and FAC_TQ_REQ_MON2 have to be initialized with zero.

All logical variables are stored in tables of status bytes. In every recurrence, the information is copied from the table into the status bytes LF_1_MON2, LF_2_MON2, LF_3_MON2 and LF_4_MON2 respectively.

The position of the logical variables LV_xyz_MON2 within the status bytes are shown in the figures before:

HINT: The curve ID_RESP_MON2 is calibrated in this module. But the values depending on TEST_REC_IDX_MON2 are used in L3 (processor monitoring). Therefore, the output of this ID is terminated in this simulation.

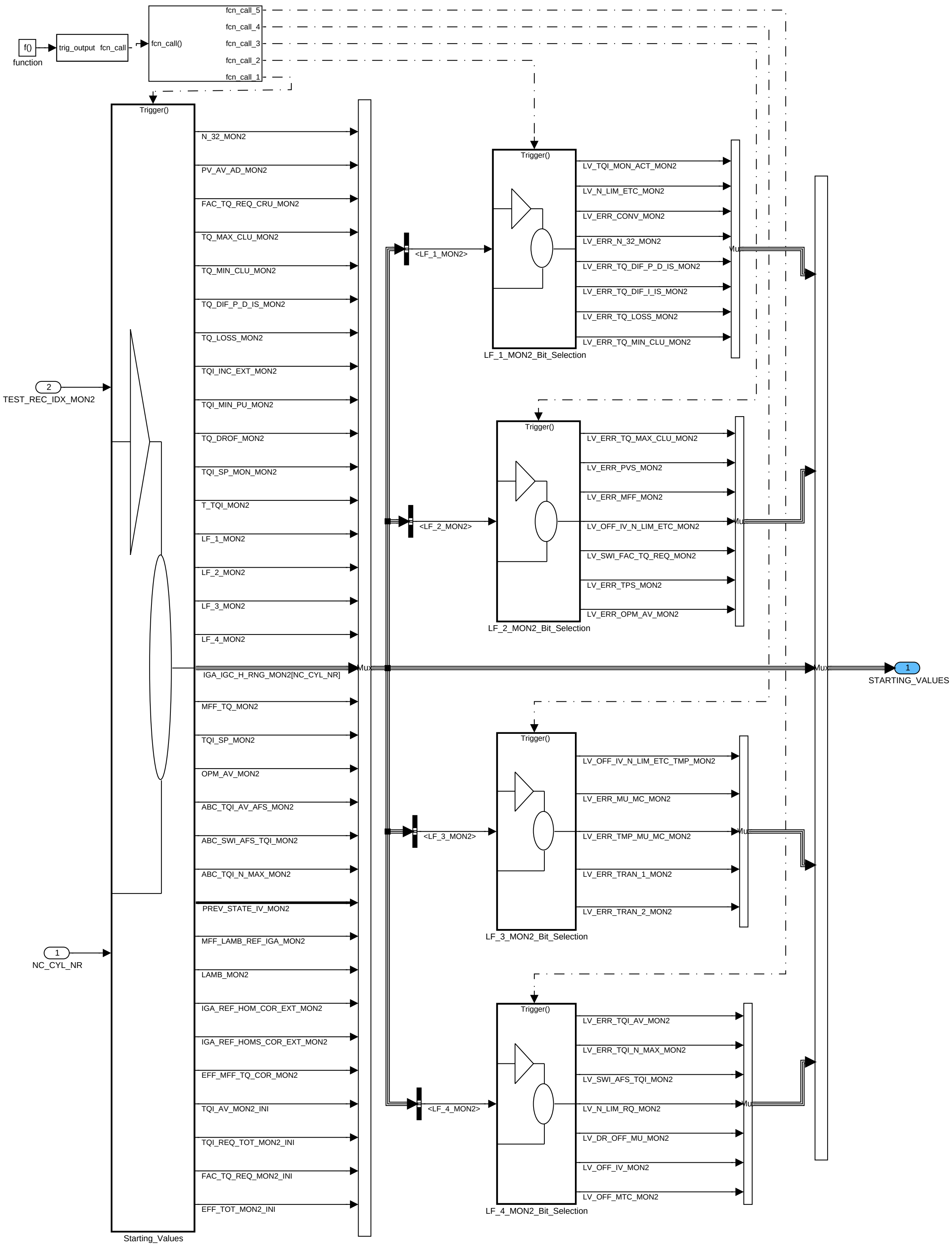


Figure 110 ECM2_DTSYSCOPPRHPDI/ OPERATE/ INI_START_VALUES

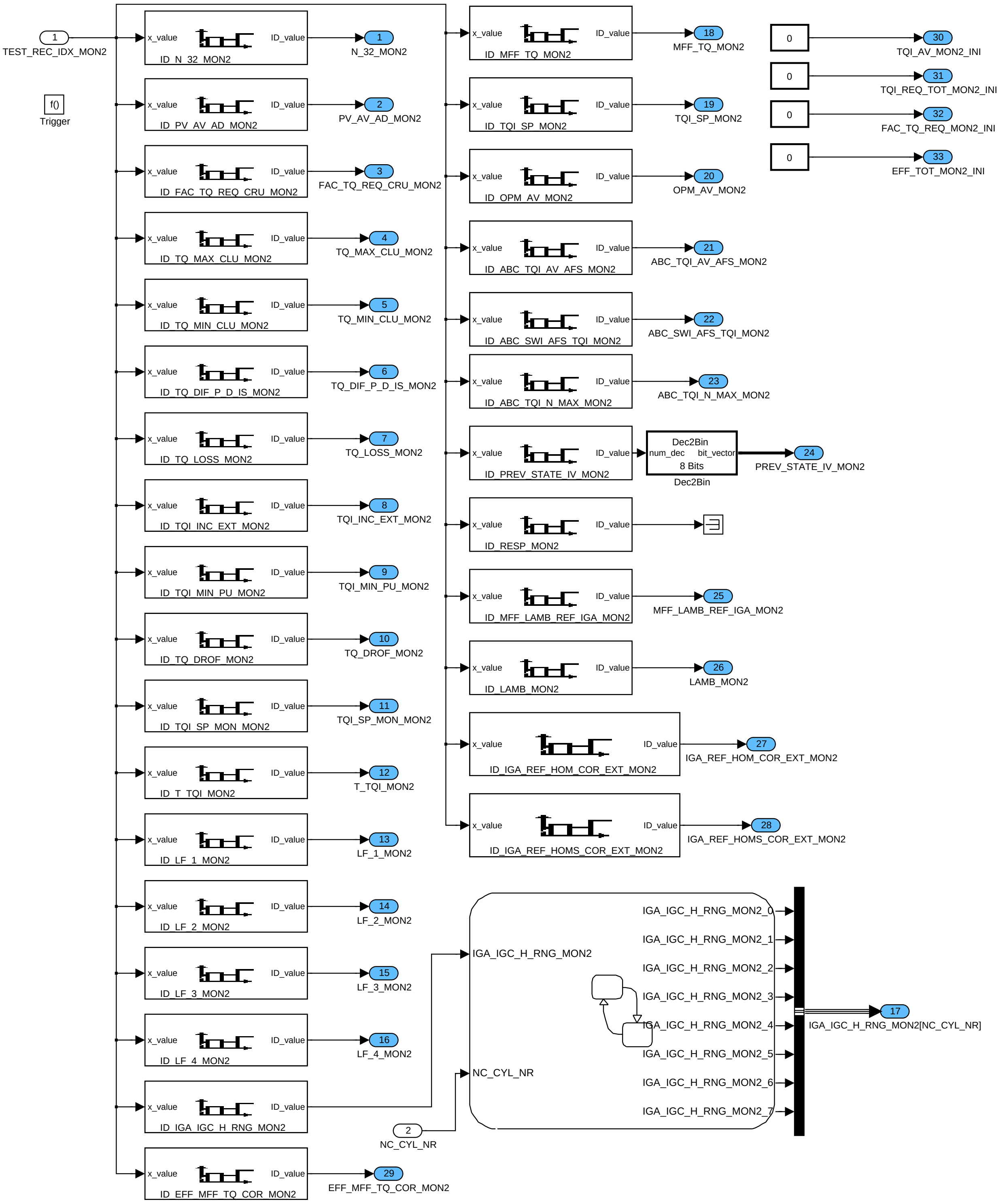


Figure 111 ECM2_DTSYSCOPPRHPDI/ OPERATE/ INI_START_VALUES/ Starting_Values

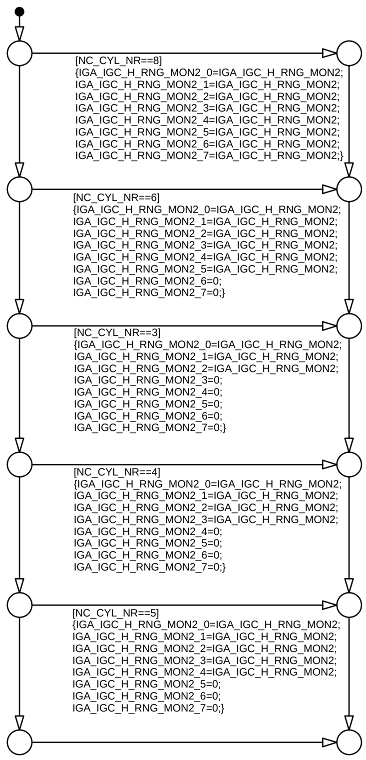


Figure 112 ECM2_DTSYSCOPPRHPDI/ OPERATE/ INI_START_VALUES/ Starting_Values/ Chart1

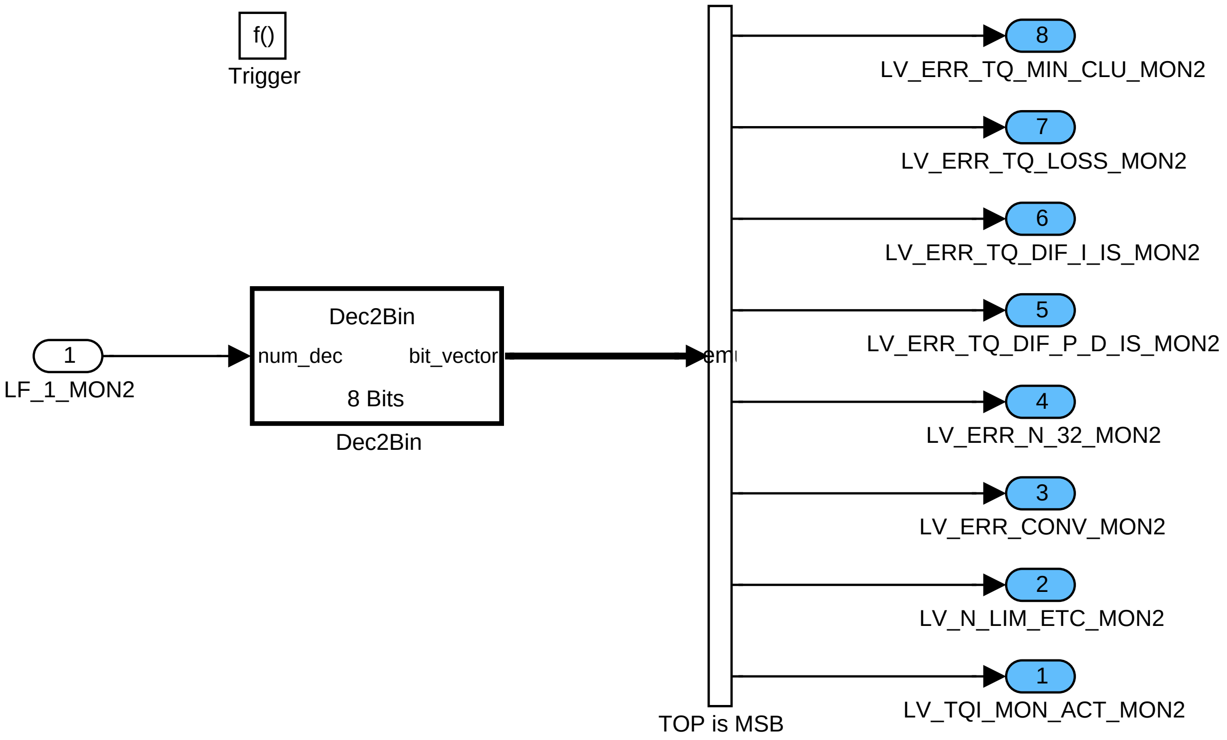


Figure 113 ECM2_DTSYSCOPPRHPDI/ OPERATE/ INI_START_VALUES/ LF_1_MON2_Bit_Selection

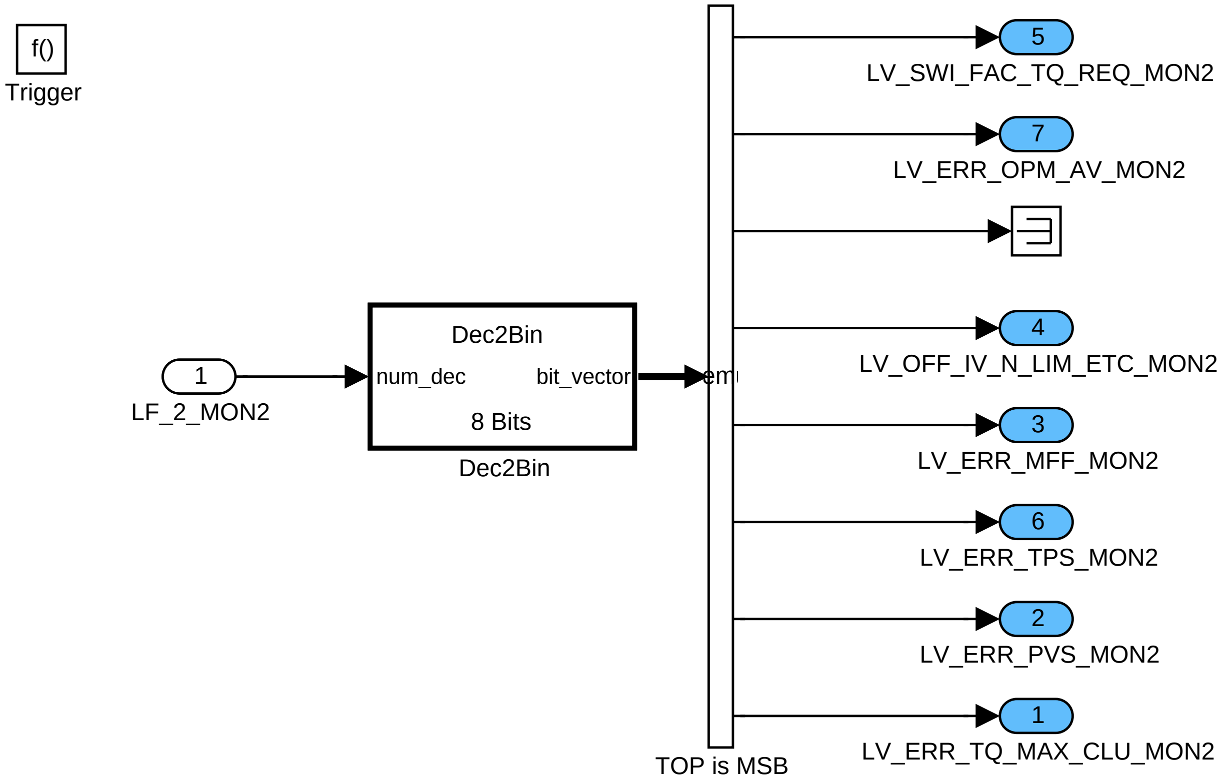


Figure 114 ECM2_DTSYSCOPPRHPDI/ OPERATE/ INI_START_VALUES/ LF_2_MON2_Bit_Selection

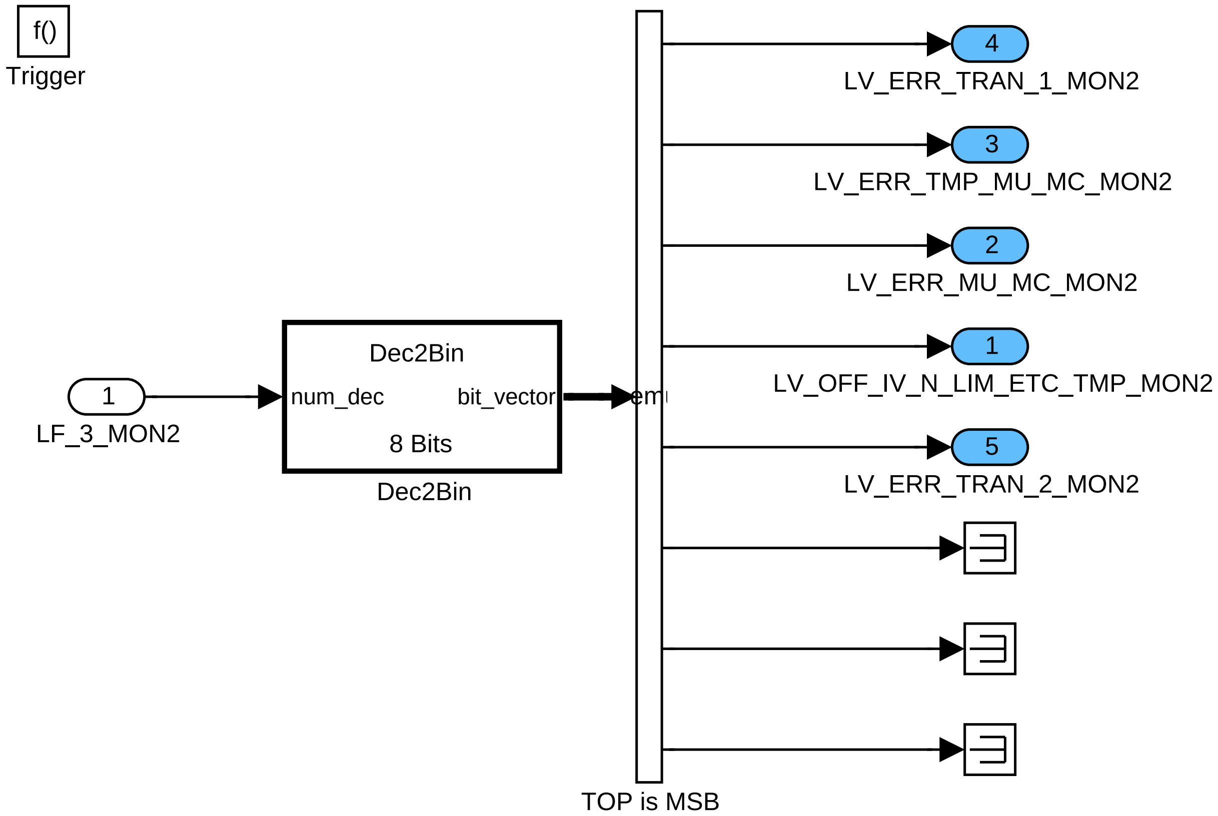


Figure 115 ECM2_DTSYSCOPPRHPDI/ OPERATE/ INI_START_VALUES/ LF_3_MON2_Bit_Selection

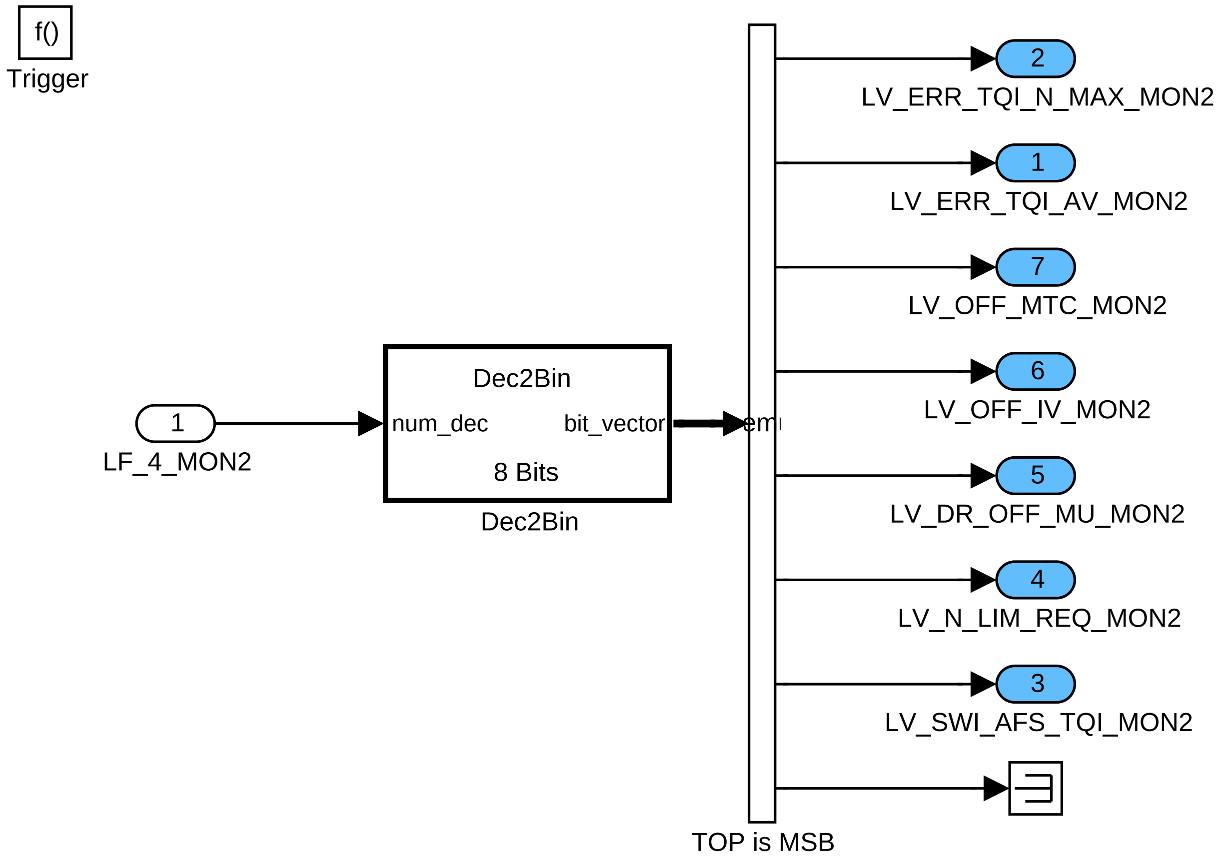


Figure 116 ECM2_DTSYSCOPPRHPDI/ OPERATE/ INI_START_VALUES/ LF_4_MON2_Bit_Selection

<u>Register calculation 1 for checking Interrupt routine</u>

After reloading the register values REG_x_MON2 from the stack, N_32_MON2 from the sub-module INI_START_VALUE shall be added to every REG_x_MON2.

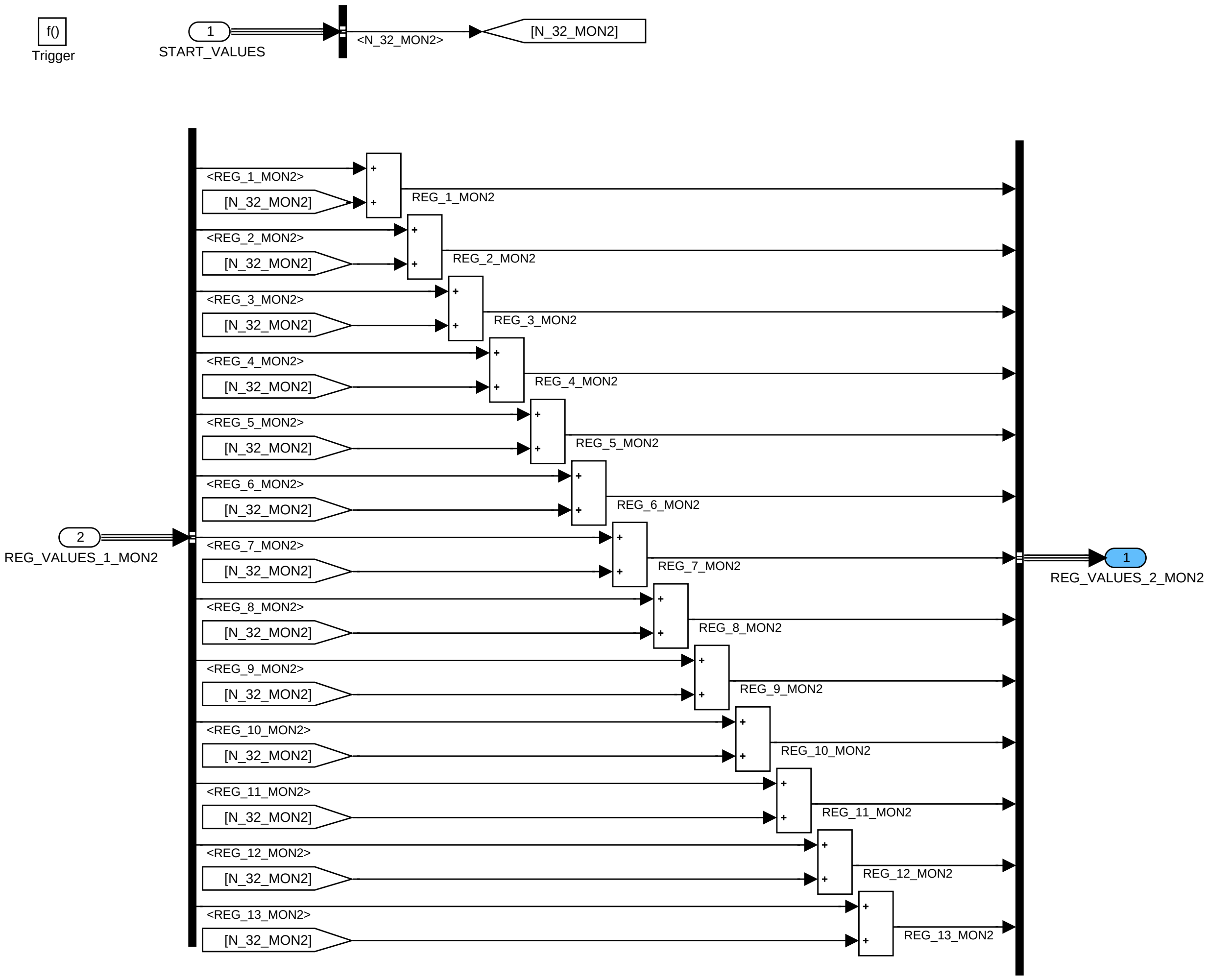


Figure 117 ECM2_DTSYSCOPPRHPDI/ OPERATE/ REG_CALC_1

ECM2_DTSYSCOPPRHPDI/OPERATE/TQI_SP

General information:

The module has exactly the same functionality than the corresponding module ‘Desired indicated engine torque’ of level 2 (`30E00K01.00I`) except:

No RAM test of output data performed

No initialisation section included

TQI_SP_MON (output) renamed to TQI_SP_MON_MON2

All input data, output data and calibration data renamed to level 2’ naming Example: xyz_MON renamed to xyz_MON2

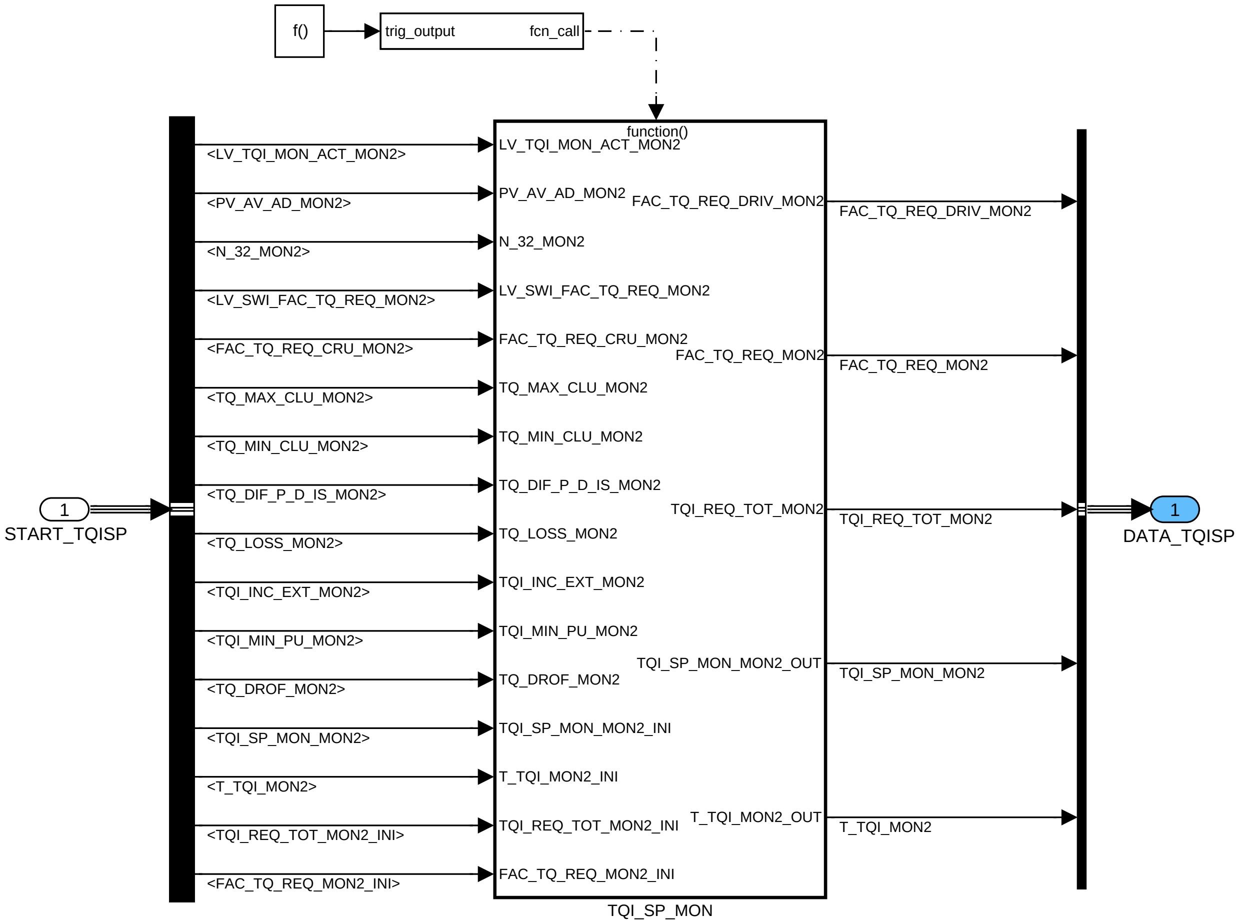


Figure 118 ECM2_DTSYSCOPPRHPDI/ OPERATE/ TQI_SP

<u>Register calculation 2 for checking Interrupt routine</u>

After reloading the register values REG_x_MON2 from the stack, TQI_SP_MON_MON2 from the sub-module TQI_SP shall be added to every REG_x_MON2.

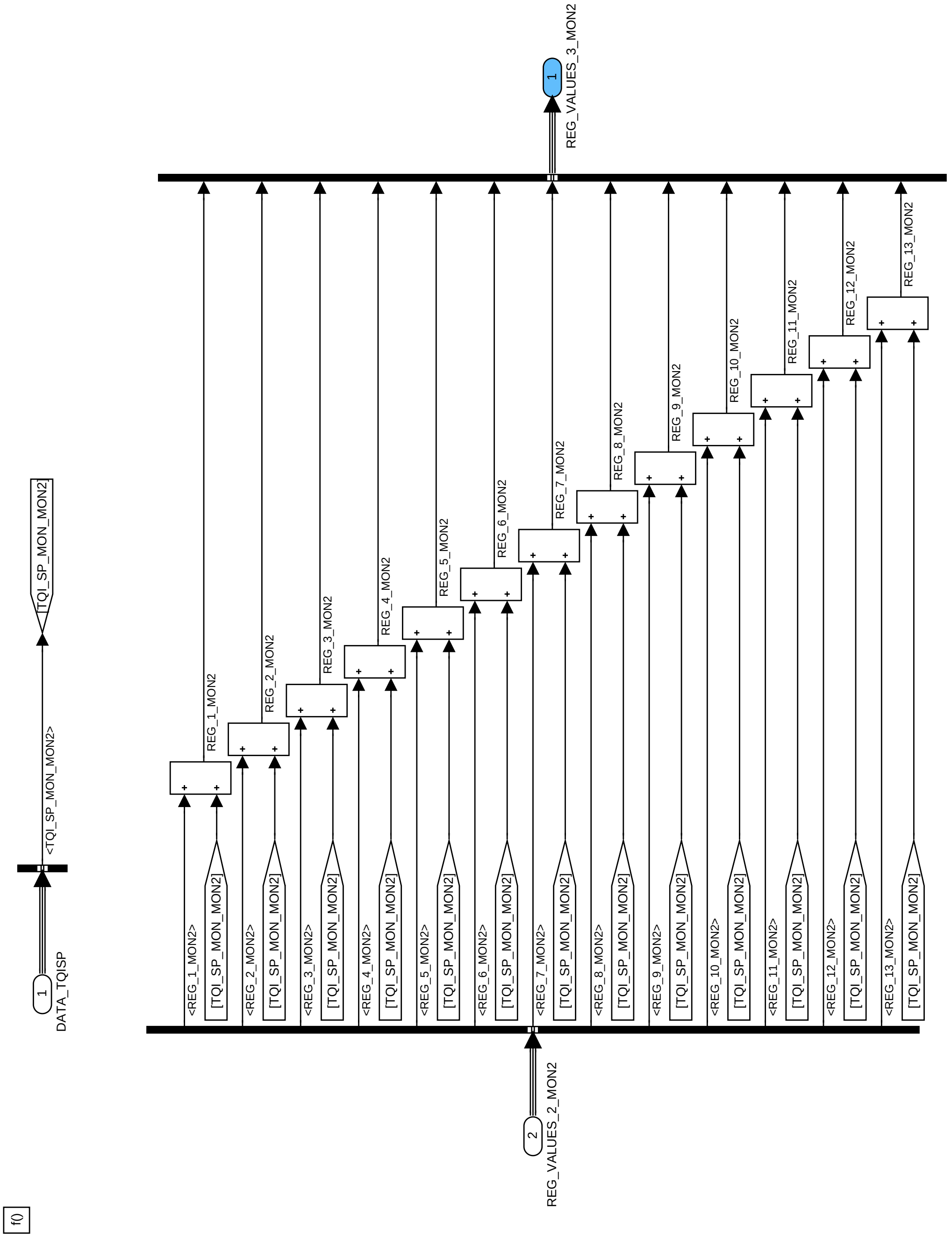


Figure 119 ECM2_DTSYSCOPPRHPDI/ OPERATE/ REG_CALC_2

ECM2_DTSYSCOPPRHPDI/OPERATE/ACTUAL_EFF

General information:

The module has exactly the same functionality than the corresponding module ‘Actual efficiencies’ of level 2 (30E01401.00J) except:

No RAM test of output data performed

No initialisation section included

All input data, output data and calibration data renamed to level 2' naming Example: xyz_MON renamed to xyz_MON2

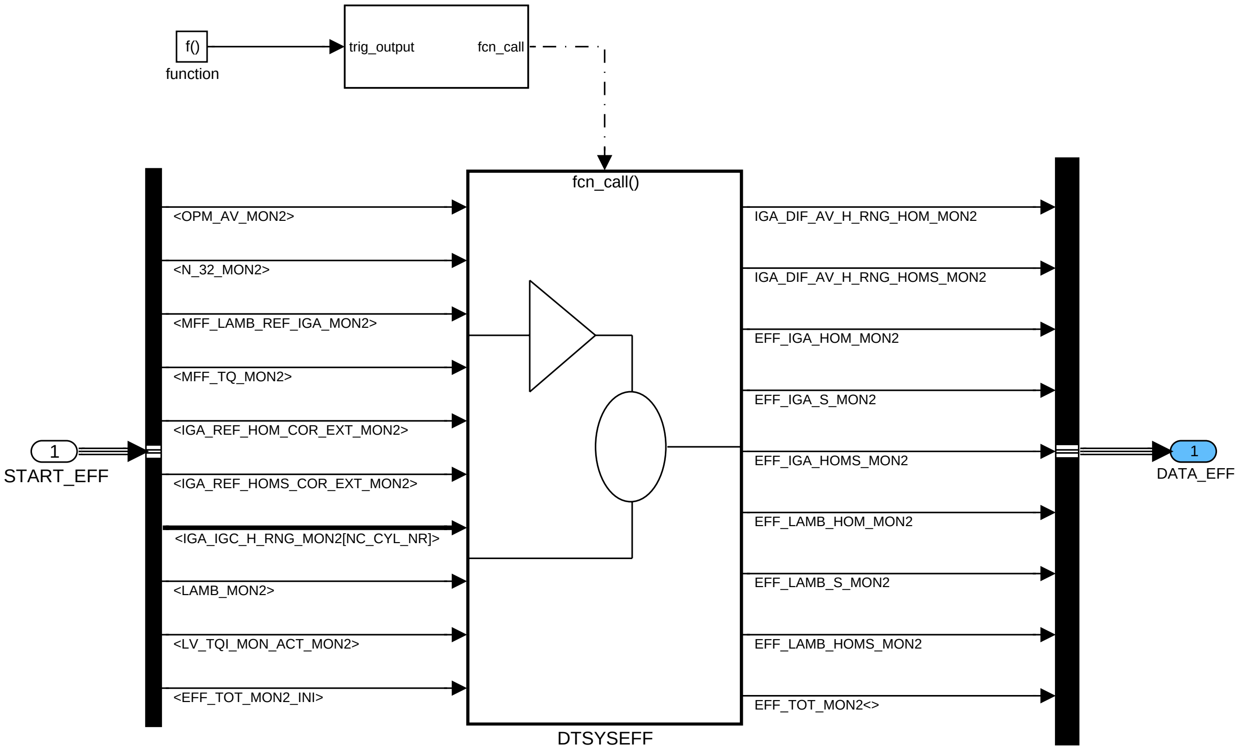


Figure 120 ECM2_DTSYSCOPPRHPDI/ OPERATE/ ACTUAL_EFF

Register calculation 3 for checking Interrupt routine

After reloading the register values REG_x_MON2 from the stack, EFF_TOT_MON2 from the sub-module ACTUAL_EFF shall be added to every REG_x_MON2.

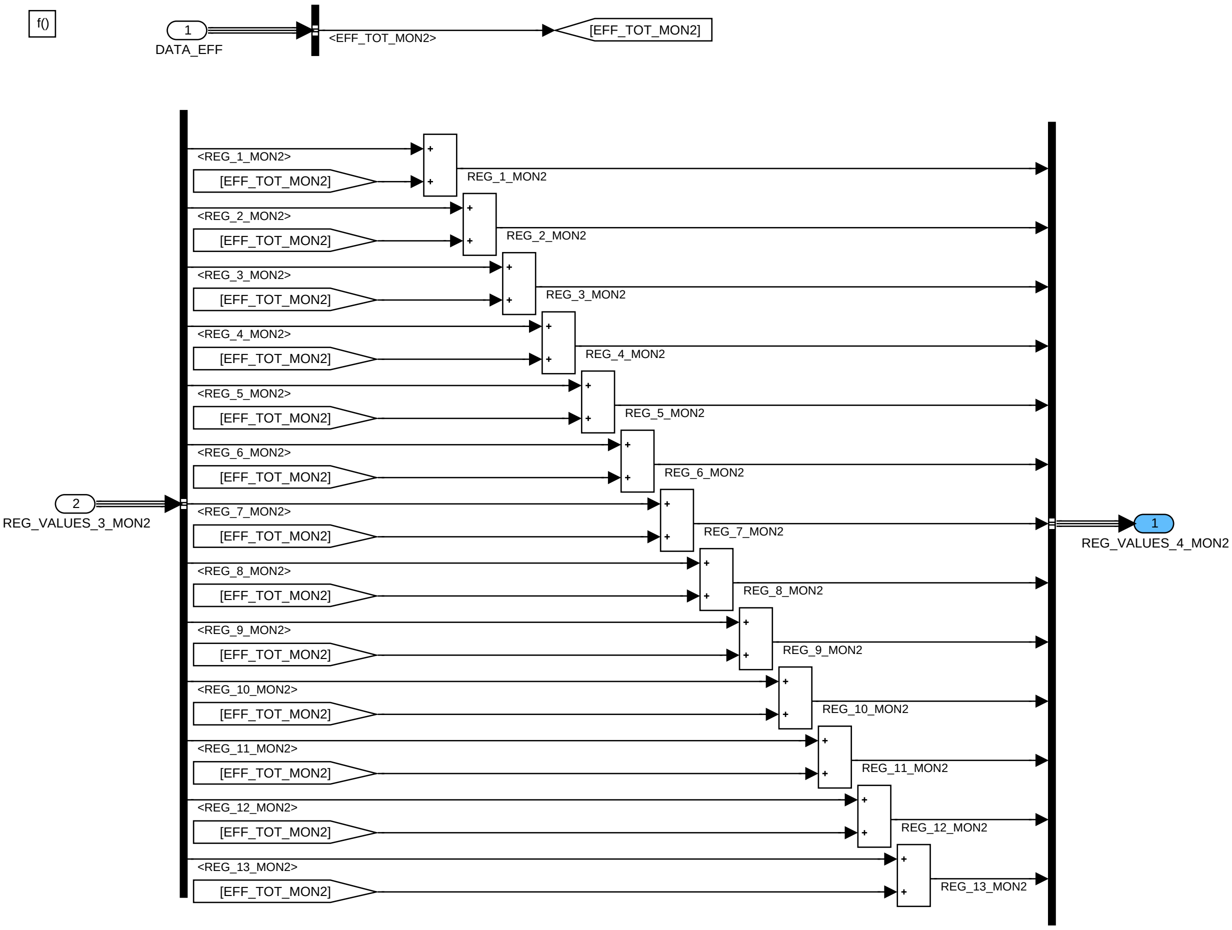


Figure 121 ECM2_DTSYSCOPPRHPDI/ OPERATE/ REG_CALC_3

ECM2_DTSYSCOPPRHPDI/OPERATE/TQI_AV

General information:

The module has exactly the same functionality than the corresponding module ‘Actual indicated engine torque’ of level 2 (`30E00L01.00E`) except:

No RAM test of output data performed

No initialisation section included

All input data, output data and calibration data renamed to level 2’ naming Example: xyz_MON renamed to xyz_MON2

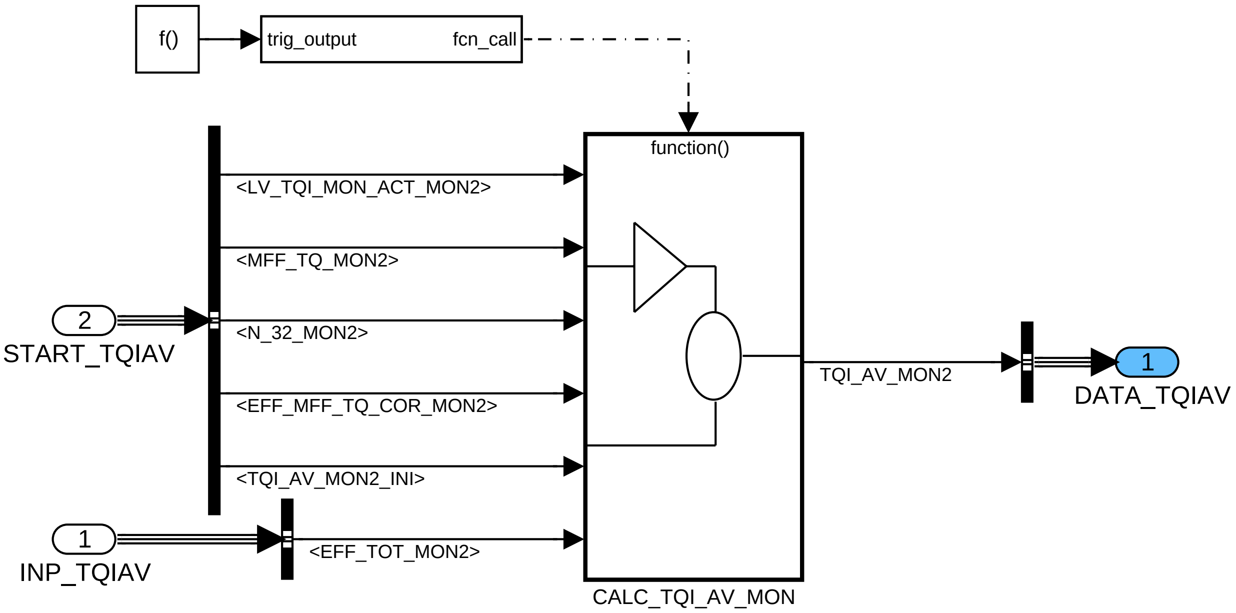


Figure 122 ECM2_DTSYSCOPPRHPDI/ OPERATE/ TQI_AV

Register calculation 4 for checking Interrupt routine

After reloading the register values REG_x_MON2 from the stack, TQI_AV_MON2 from the sub-module TQI_AV shall be added to every REG_x_MON2.

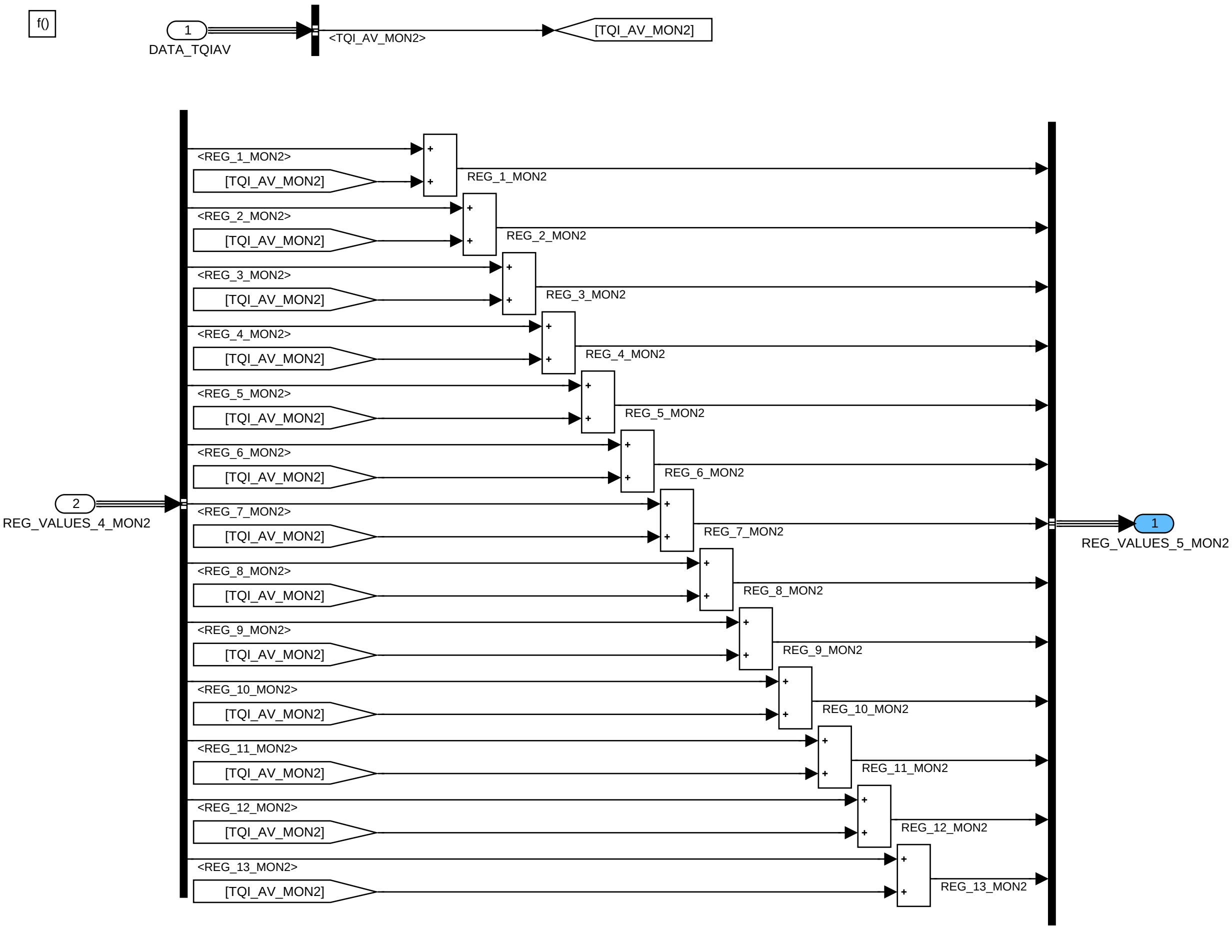


Figure 123 ECM2_DTSYSCOPPRHPDI/ OPERATE/ REG_CALC_4

ECM2_DTSYSCOPPRHPDI/OPERATE/TQMON

General information:

The module has exactly the same functionality than the corresponding module ‘Monitoring of actual indicated engine torque’ of level 2 (`30E00M01.00E`) except:

No RAM test of output data performed

No initialisation section included

TQI_SP_MON (input) renamed to TQI_SP_MON_MON2

All input data, output data and calibration data renamed to level 2’ naming Example: xyz_MON renamed to xyz_MON2

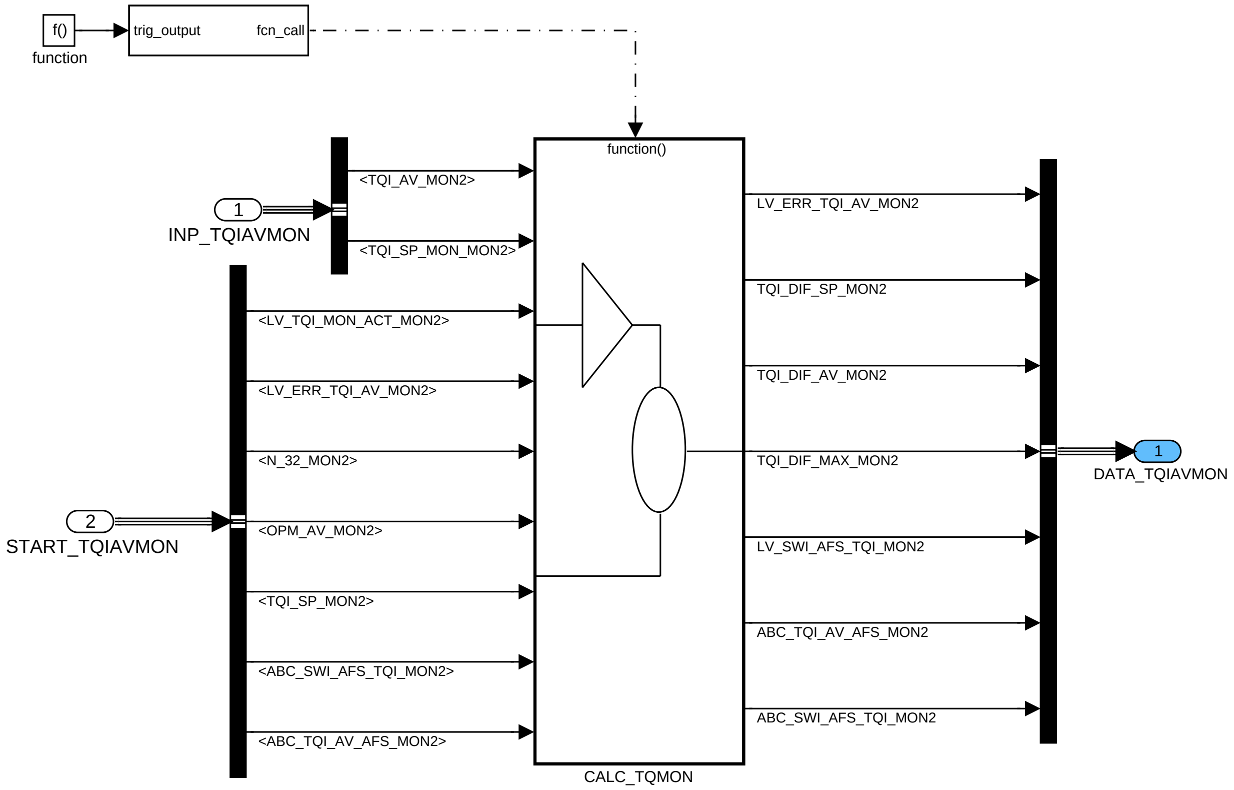


Figure 124 ECM2_DTSYSCOPPRHPDI/ OPERATE/ TQMON

Register calculation 5 for checking Interrupt routine

After reloading the register values REG_x_MON2 from the stack, ABC_TQI_AV_AFS_MON2 from the sub-module TQMON shall be added to every REG_x_MON2.

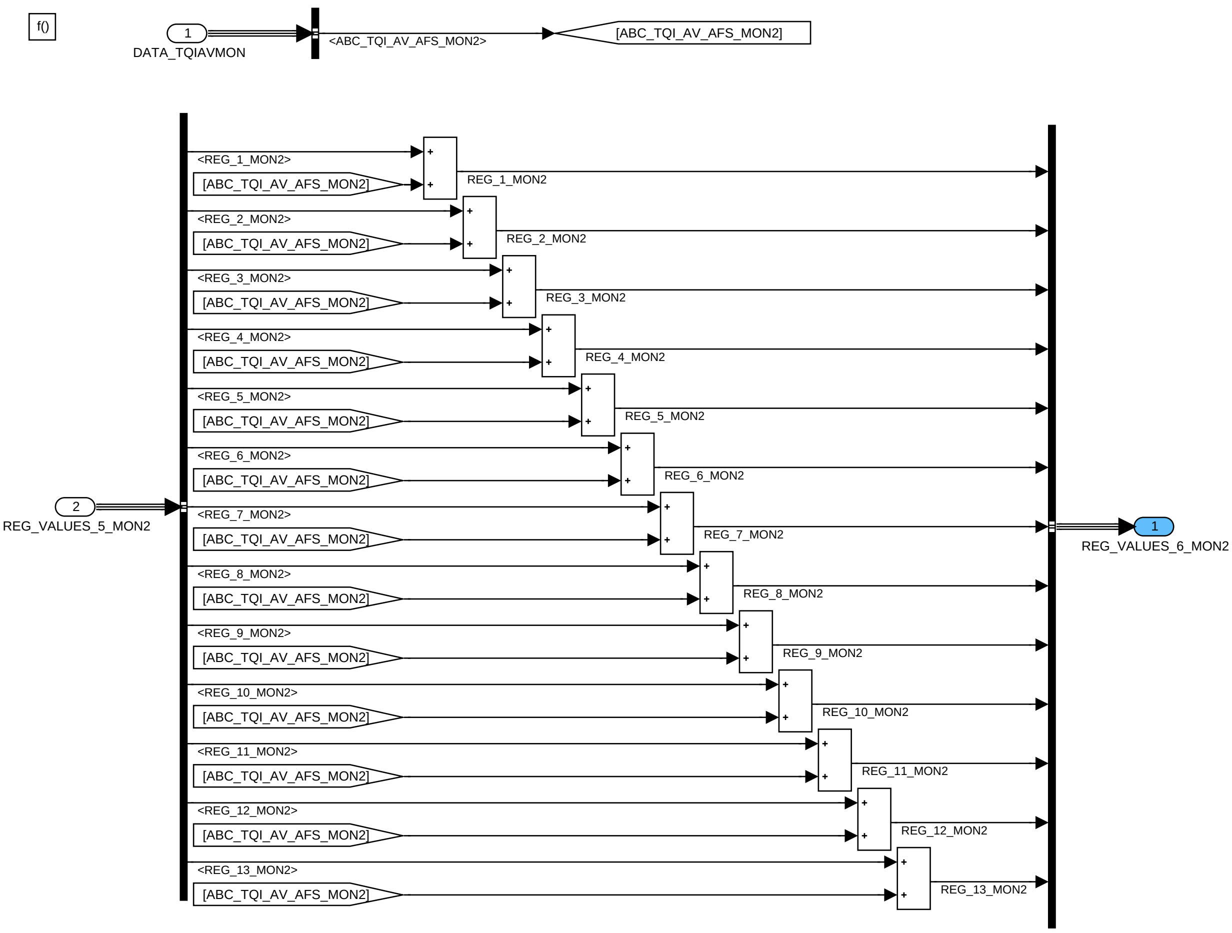


Figure 125 ECM2_DTSYSCOPPRHPDI/ OPERATE/ REG_CALC_5

ECM2_DTSYSCOPPRHPDI/OPERATE/N_LIM_MON

General information:

The module has exactly the same functionality than the corresponding module ‘Monitoring of engine speed limitation’ of level 2 (`30E00I01.00F`) except:

No RAM test of output data performed

No initialisation section included

All input data, output data and calibration data renamed to level 2’ naming Example: xyz_MON renamed to xyz_MON2

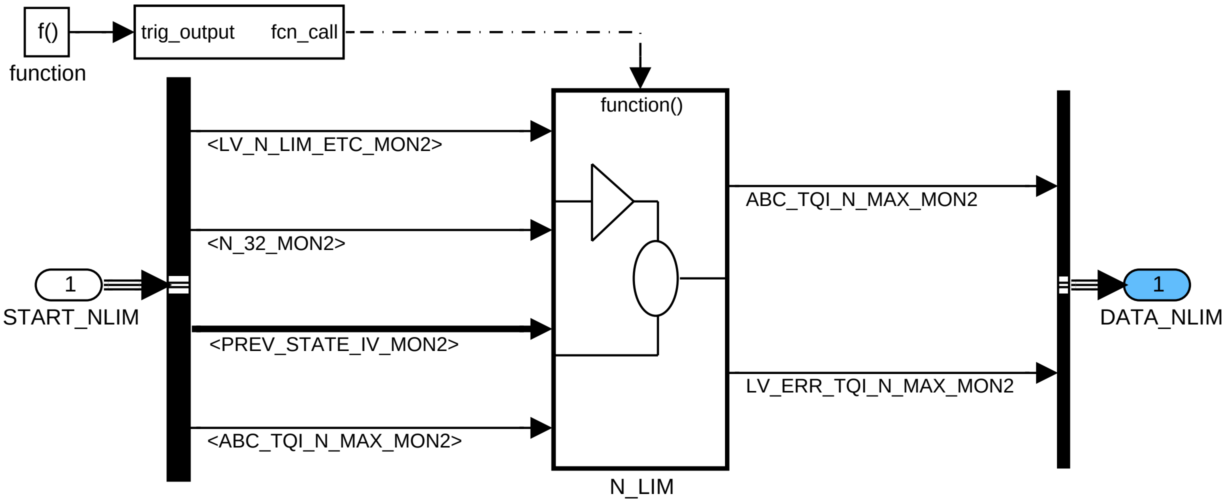


Figure 126 ECM2_DTSYSCOPPRHPDI/ OPERATE/ N_LIM_MON

Register calculation 6 for checking Interrupt routine

After reloading the register values REG_x_MON2 from the stack, ABC_TQI_N_MAX_MON2 from the sub-module N_LIM_MON shall be added to every REG_x_MON2.

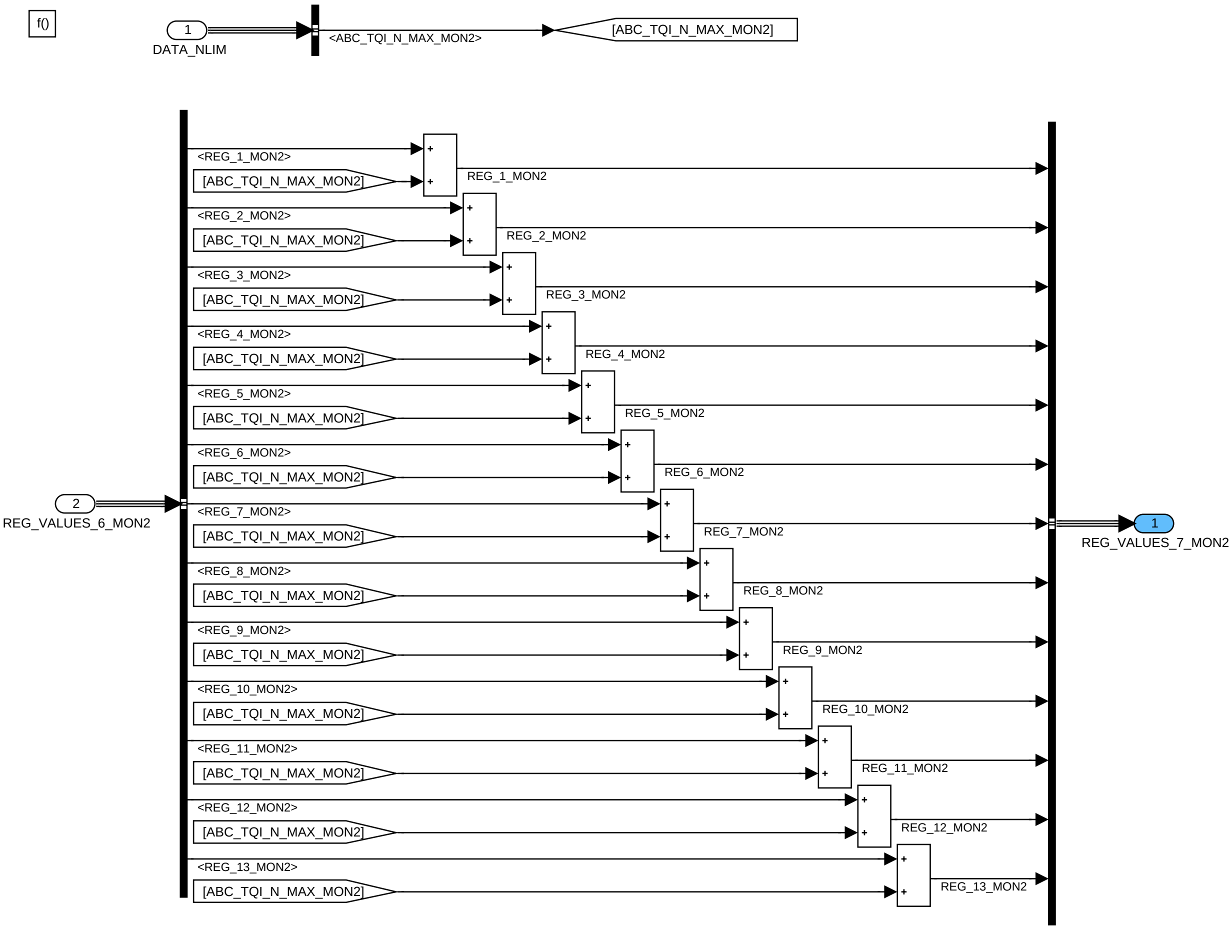


Figure 127 ECM2_DTSYSCOPPRHPDI/ OPERATE/ REG_CALC_6

ECM2_DTSYSCOPPRHPDI/OPERATE/FAULT_REACTION

General information:

The module has exactly the same functionality than the corresponding module 'Fault reaction of process monitoring' of level 2 (`30E00G01.00J`) except:

No RAM test of output data performed

No initialisation section included

All input data, output data and calibration data renamed to level 2' naming Example: xyz_MON renamed to xyz_MON2

The ACTION_ECM3_RedSwitchOffPath() is replaced, instead of the original action call LV_DR_OFF_MU_MON2 is set to 1.

The byte LF_4_MON2 is updated with the output flags after execution.

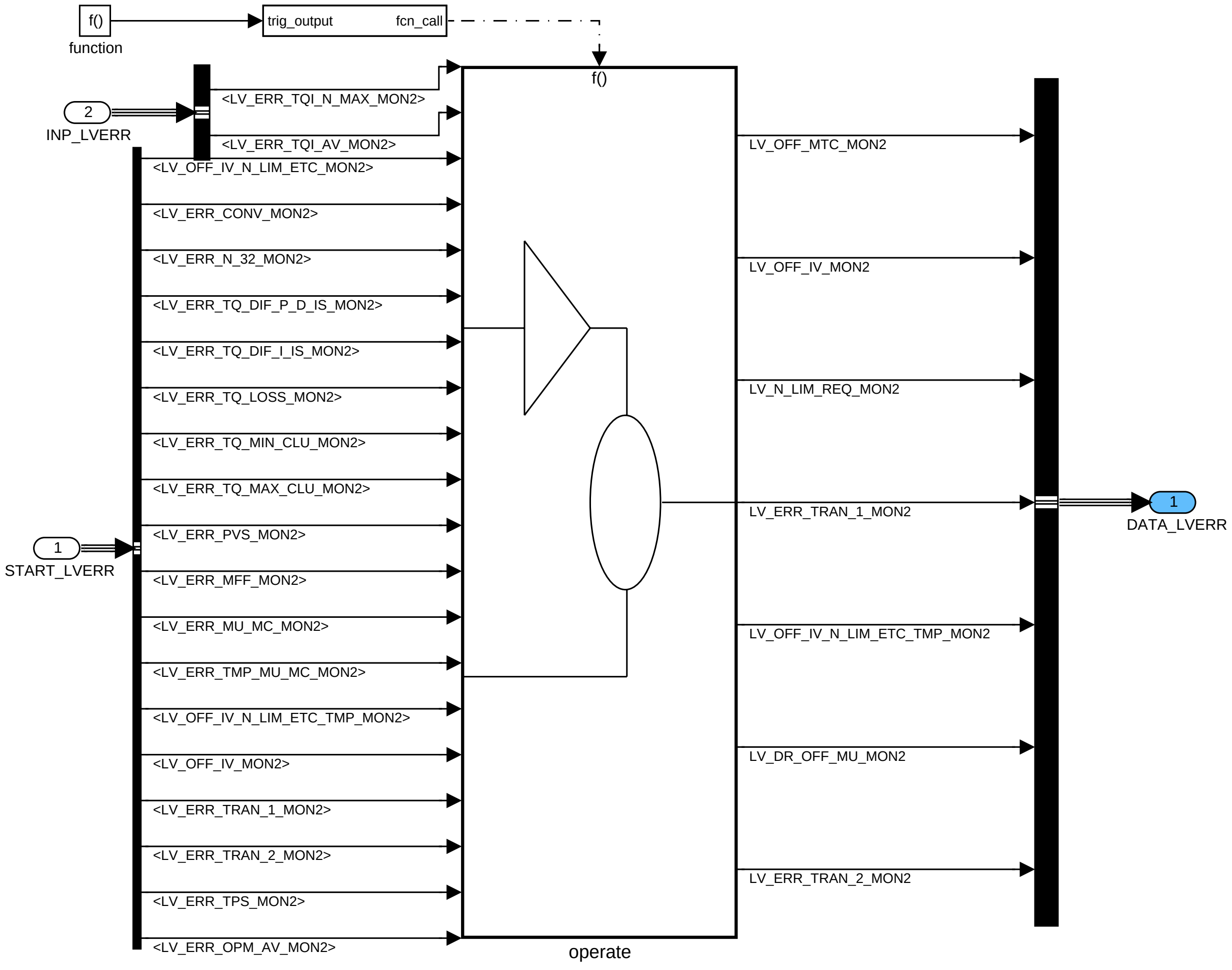


Figure 128 ECM2_DTSYSCOPPRHPDI/ OPERATE/ FAULT_REACTION

Register calculation 7 for checking Interrupt routine

After reloading the register values REG_x_MON2 from the stack, LV_OFF_IV_MON2 from the sub-module FAREA shall be added to every REG_x_MON2.

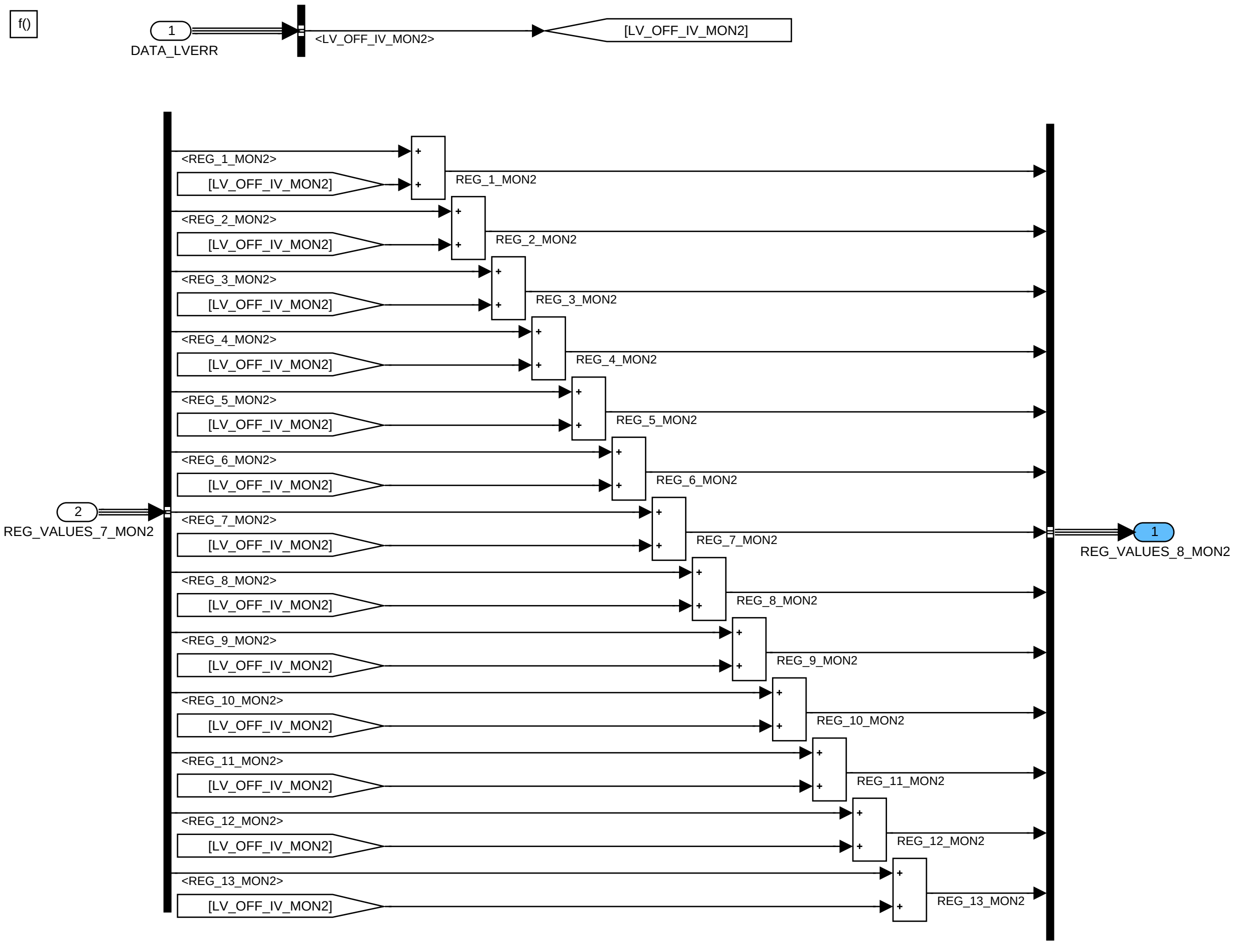


Figure 129 ECM2_DTSYSCOPPRHPDI/ OPERATE/ REG_CALC_7

ECM2_DTSYSCOPPRHPDI/OPERATE/RESP_MON_TEMP

General information:

The byte LF_4_MON2 is updated with the output flags after execution.

There is a possibility for faults, which leads to wrong interim results but to a correct final result of the test calculation. In order to ensure that such a kind of fault is detectable, the final result is composed of several intermediate ones coming out of different parts of the test calculation.

Formula section:

RESP_MON2_TEMP = (TQI_SP_MON_MON2
- TQI_AV_MON2
+ TQI_REQ_TOT_MON2
+ FAC_TQ_REQ_MON2
+ ABC_TQI_AV_AFS_MON2
+ 256 * ABC_TQI_N_MAX_MON2)

XOR (256 * LF_4_MON2)

All operations are performed with hexadecimal values only. The different units are not taken into account. The result of the operations and all intermediate results are 32 Bit word size.

ID_RESP_MON2 [TEST_REC_IDX_MON2] must fit to RESP_MON2 if the test calculation with index TEST_REC_IDX_MON2 has been executed correctly.

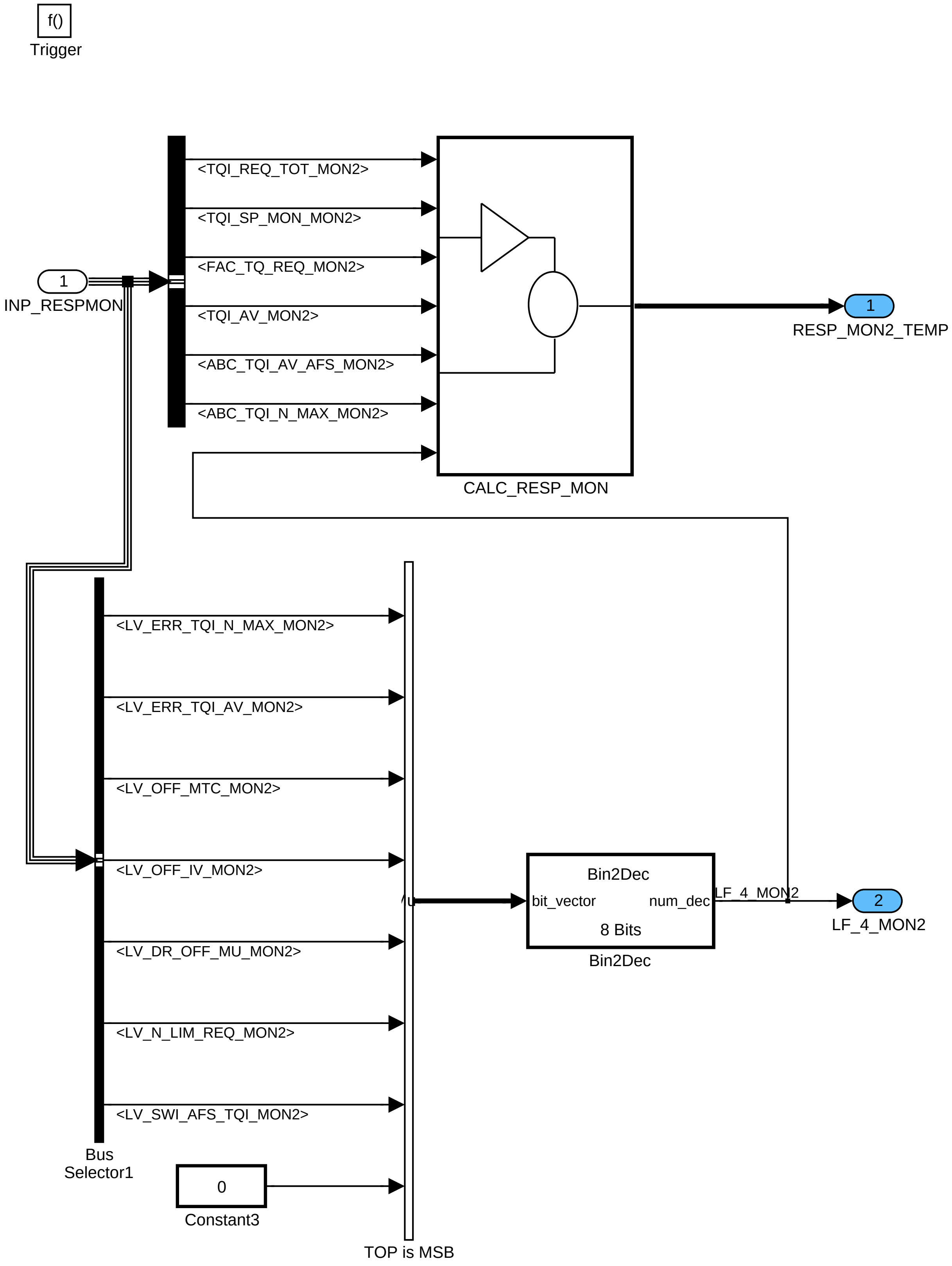


Figure 130 ECM2_DTSYSCOPPRHPDI/ OPERATE/ RESP_MON_TEMP

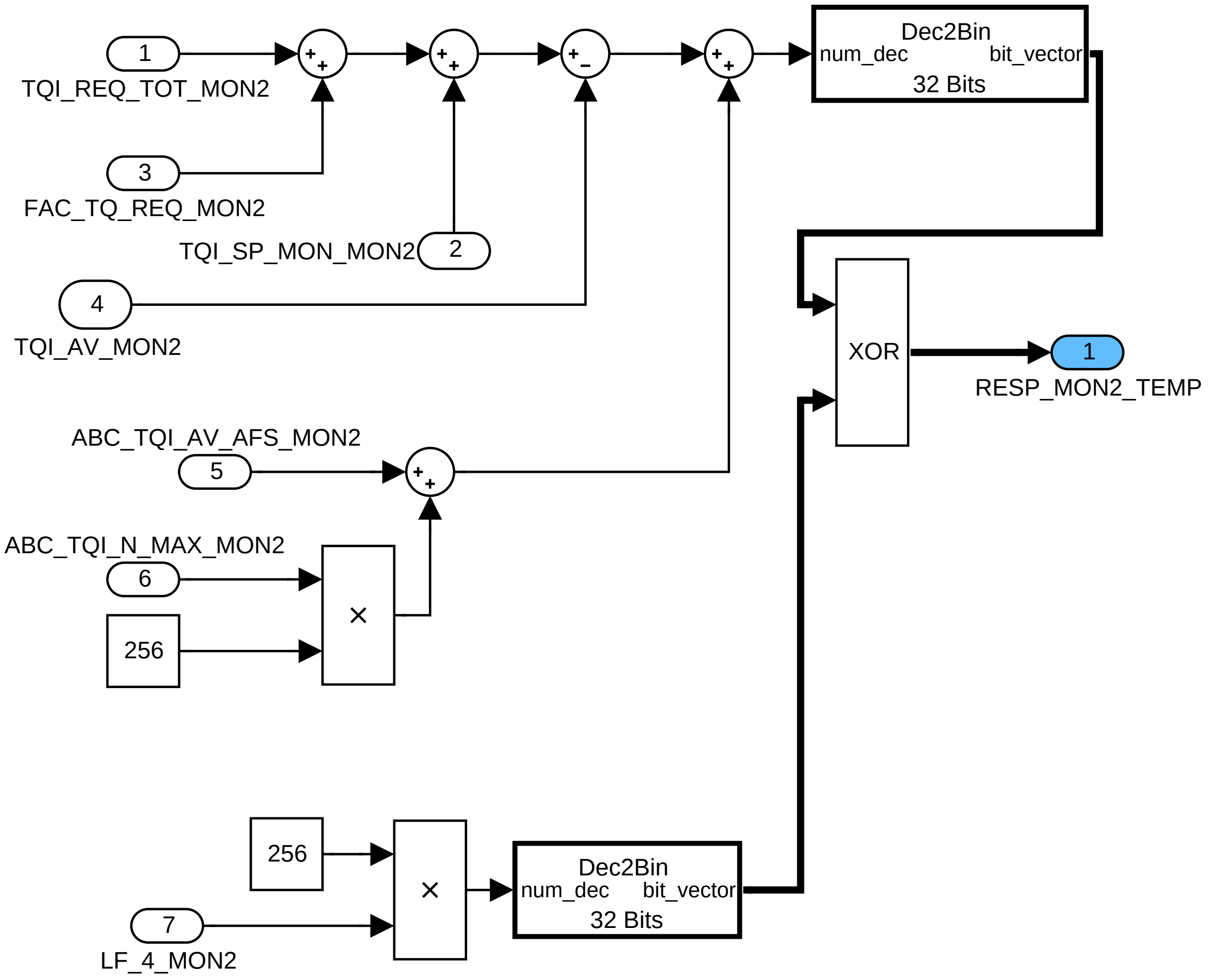


Figure 131 ECM2_DTSYSCOPPRHPDI/ OPERATE/ RESP_MON_TEMP/ CALC_RESP_MON

ECM2_DTSYSCOPPRHPDI/OPERATE/RESP_MON

For the final calculation of RESP_MON2, the 32 bit values used so far have to be encoded into a 16 bit value. Therefore, after adding the register values, the high and the low 16 bit values of the resulting sum shall be combined by an XOR. The same procedure shall be applied to the temporary result RESP_MON2_TEMP which shall the additionally also be XOR-ed with the final result of the register operations.

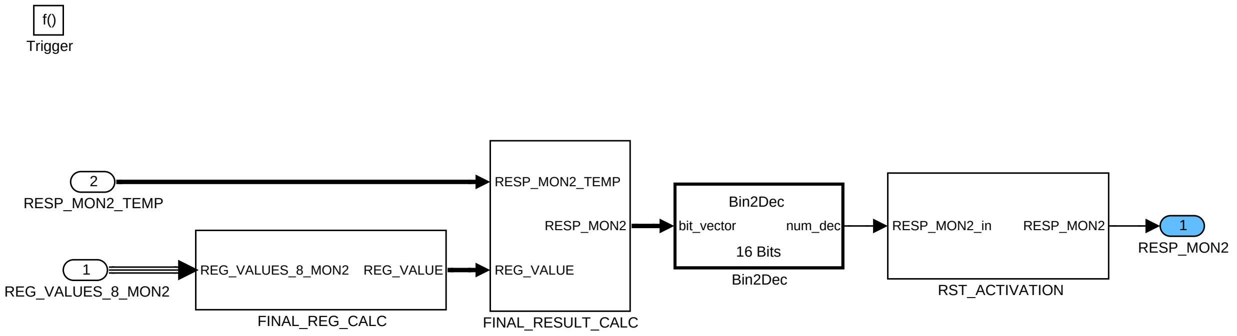


Figure 132 ECM2_DTSYSCOPPRHPDI/ OPERATE/ RESP_MON

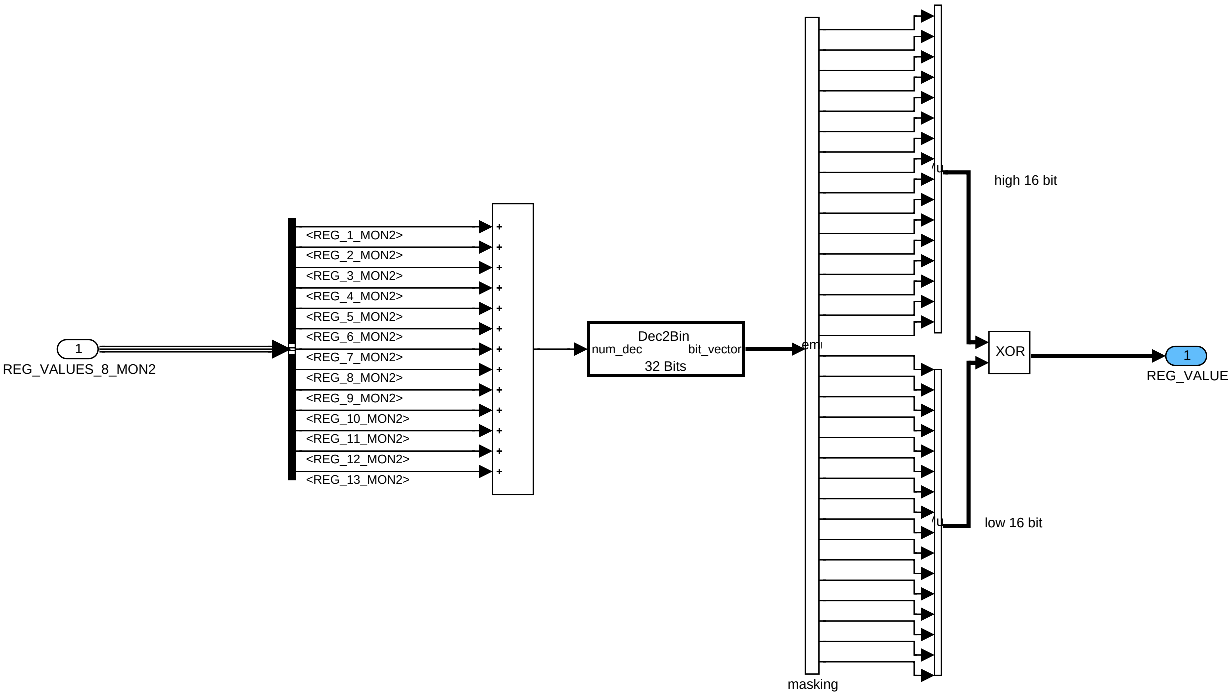


Figure 133 ECM2_DTSYSCOPPRHPDI/ OPERATE/ RESP_MON/ FINAL_REG_CALC

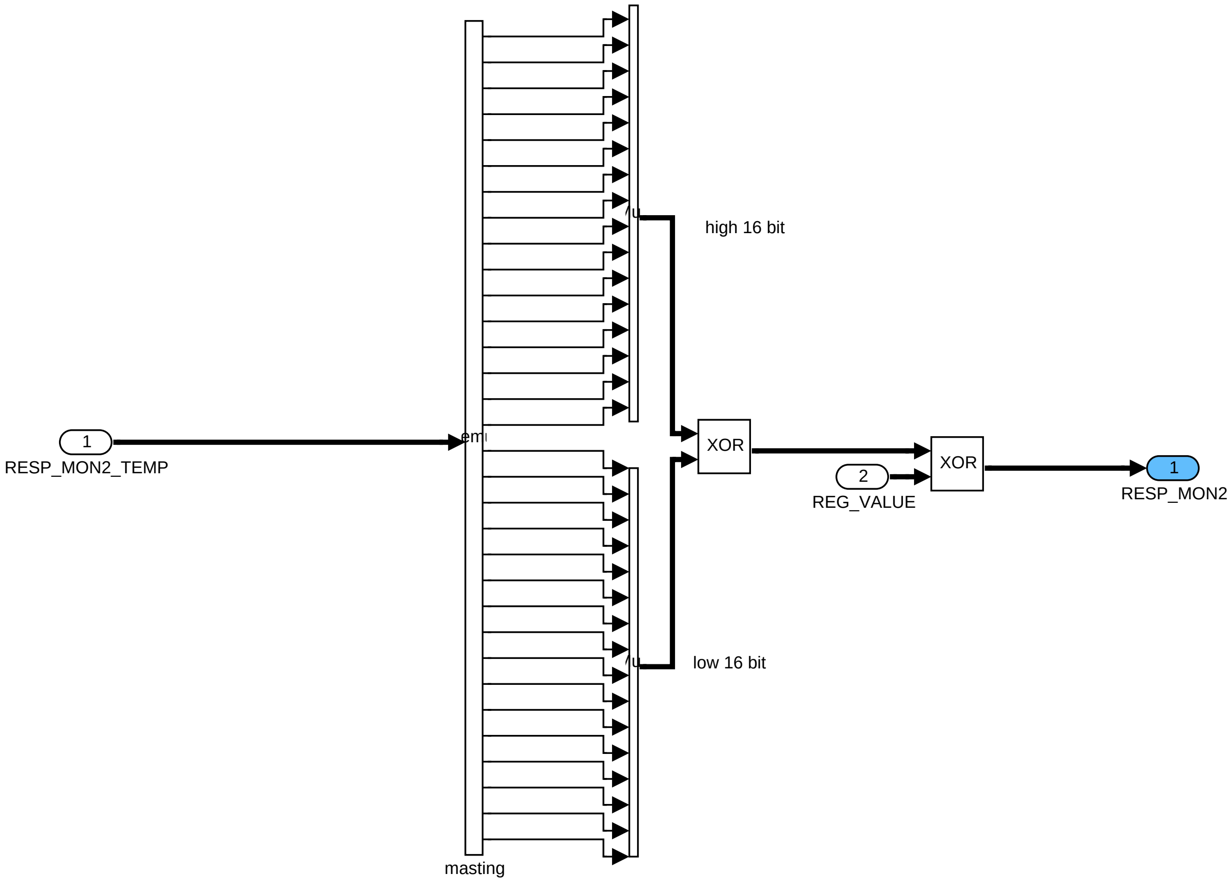


Figure 134 ECM2_DTSYSCOPPRHPDI/ OPERATE/ RESP_MON/ FINAL_RESULT_CALC

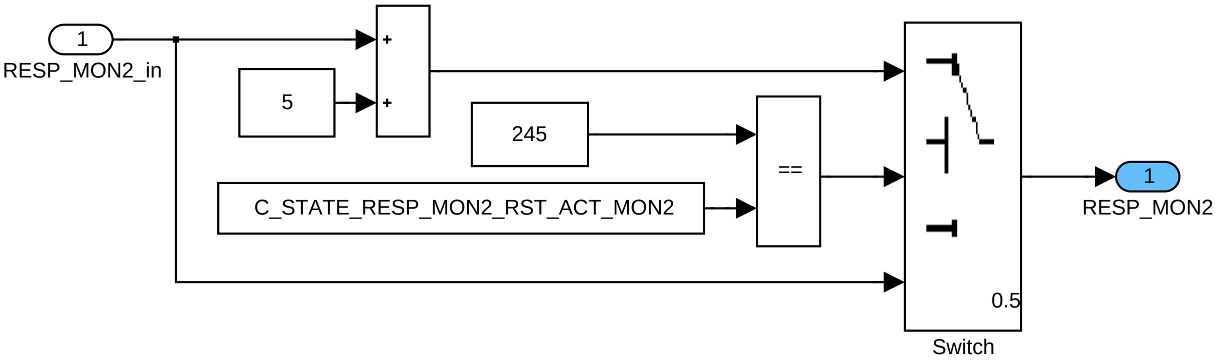


Figure 135 ECM2_DTSYSCOPPRHPDI/ OPERATE/ RESP_MON/ RST_ACTIVATION

11.28 Fault Reaction of Process Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_N_LIM_REQ_MON	O/V	0...1H	0...1	1	[-]
Request for engine speed limitation					
LV_OFF_IV_MON	O/V	0...1H	0...1	1	[-]
Request for disable of IV power stage by main controller					
LV_OFF_MTC_MON	O/V	0...1H	0...1	1	[-]
Request for disable of MTC power stage by main controller					
LV_ERR_TRAN_1_MON	V	0...1H	0...1	1	[-]
Flag for temporary fault 1 (exclusive fault of L3) during last recurrency					
LV_OFF_IV_N_LIM_ETC_TMP_MON	V	0...1H	0...1	1	[-]
Flag for temporary injection cut off during last recurrency					
LV_ERR_TRAN_2_MON	V	0...1H	0...1	1	[-]
Flag for temporary fault 2 (together with fault of L2) during last recurrency					

Input data:

LV_OFF_IV_N_LIM_ETC_MON	LV_ERR_CONV_MON	LV_ERR_N_32_MON	LV_ERR_TQ_DIF_P_D_IS_MON
LV_ERR_TQ_DIF_I_IS_MON	LV_ERR_TQ_LOSS_MON	LV_ERR_TQ_MIN_CLU_MON	LV_ERR_TQ_MAX_CLU_MON
LV_ERR_PVS_MON	LV_ERR_MFF_MON	LV_ERR_TQI_AV_MON	LV_ERR_TQI_N_MAX_MON
LV_ERR_MU_MC	LV_ERR_TMP_MU_MC	LV_ERR_TPS_MON	LV_ERR_OPM_AV_MON

Import actions:

ACTION_ECM3_Service12TaskPfm(IN <>)
ACTION_ECM3_Service13TaskPfm(IN <>)
ACTION_ECM3_Service14TaskPfm(IN <>)

Note: These actions are defined in chapter "Processor Monitoring". The first action shall be executed as first instruction in this module, the second after the stack check, and the third action as last instruction of this module. The arguments shall have value 4.

ACTION_ECM3_McChkStack()

Note: This action is defined in chapter "Processor Monitoring" and implements a stack check. It shall be executed between ACTION_ECM3_Service12TaskPfm() and ACTION_ECM3_Service13TaskPfm().

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)

Note: All variables with Mode O and history variables in modules of the *Process Monitoring* have to be checked by the cyclical RAM test.

ACTION_ECM3_RedSwitchOffPath()

Note: This action is defined in chapter "Processor Monitoring", subsection "The redundant switch off path".

Export Actions:

ACTION_ECM2_LockPws()
Disable power stages by process monitoring (used when no ECM2 tasks executed)
ACTION_ECM2_UnlockPws()
Enable power stages by process monitoring for exactly one time (used when ECM2 tasks executed)
ACTION_ECM2_ResetErrorFlags()
Enable power stages unless fault reaction already active

Description for ACTION_ECM2_LockPws():

Description:

Syntax: ACTION_ECM2_LockPws()

Formula section:

LV_OFF_IV_MON = 1;

LV_OFF_MTC_MON = 1;

/* Disable injection and H-bridge */

Description for ACTION_ECM2_UnlockPws():

Description:

Syntax: ACTION_ECM2_UnlockPws()

Formula section:

LV_OFF_IV_MON = 0;

LV_OFF_MTC_MON = 0;

/* Enable injection and H-bridge for exactly one time */

Description for ACTION_ECM2_ResetErrorFlags():

Description:

Syntax: ACTION_ECM2_ ResetErrorFlags()

Formula section:

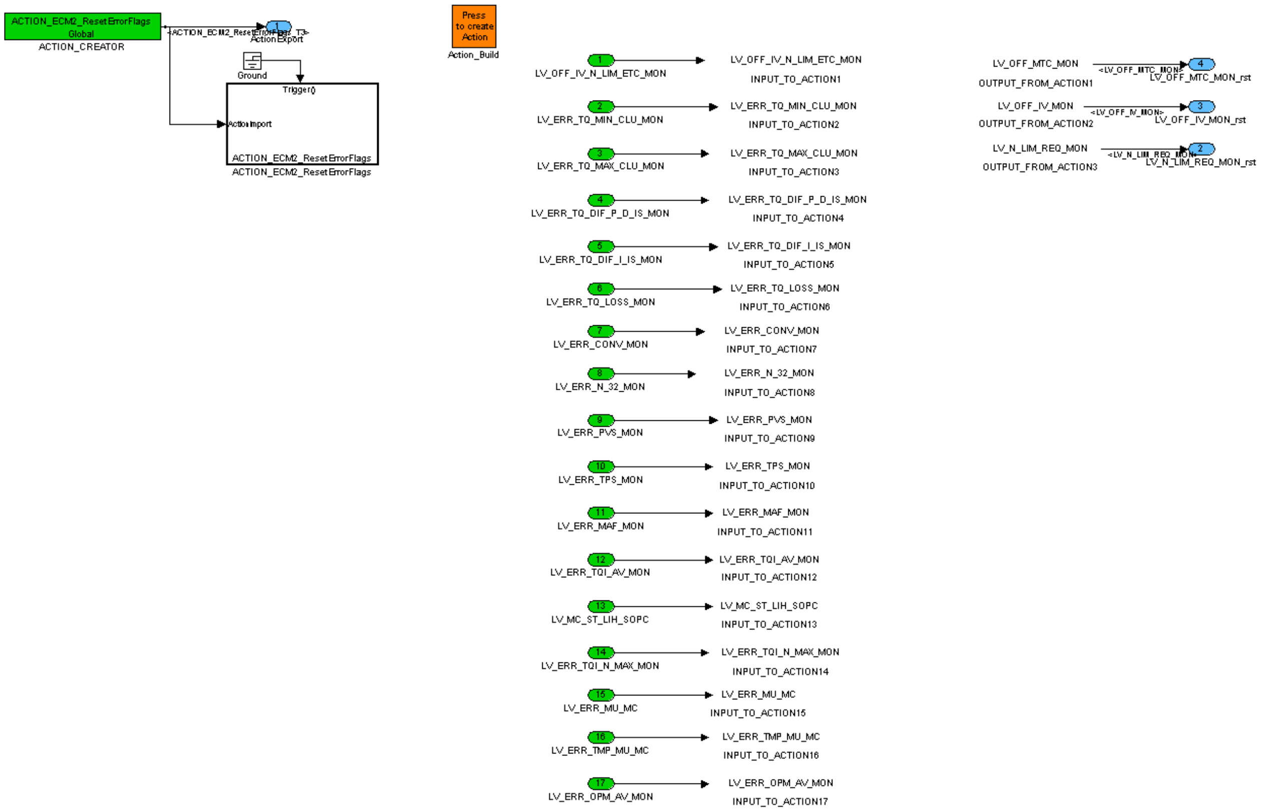


Figure 136 ECM2_DTSYSFAREA / ACTIONDEF_ECM2_ResetErrorFlags

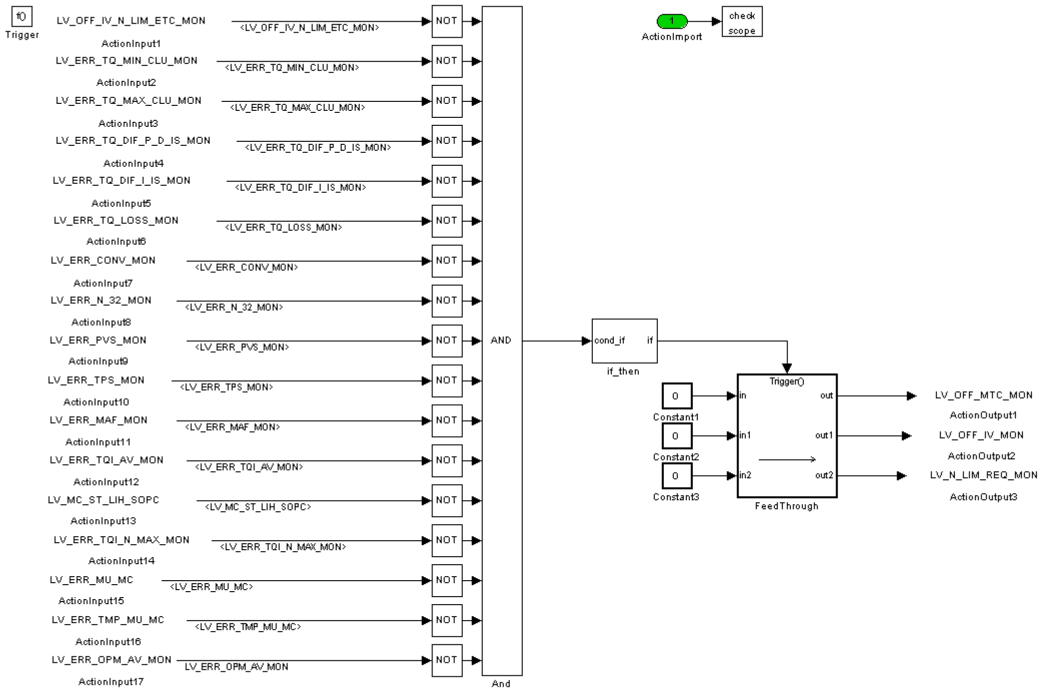


Figure 137 ECM2_DTSYSFAREA / ACTIONDEF_ECM2_ResetErrorFlags/ ACTION_ECM2_ResetErrorFlags

/* Reset of ECM2 error flags to enable power stages unless fault reaction already active */

General description:

The objective of this module is to react on faults detected by the process and processor monitoring of the etc-monitoring-concept. The following actions are introduced depending on the error level and the degree of degradation.

disable of throttle power stage (LV_OFF_MTC_MON)

disable of injection power stage (LV_OFF_IV_MON)

transmission of wrong answers from level 2’ to the monitoring processor in order to force the monitoring unit to reset the main controller and switch off the throttle and the injection power stages (ACTION_ECM3_RedSwitchOffPath())

The following table shows the fault reactions corresponding to the underlying faults. No fault reaction is reset until next ignition key on (transition from LV_IGK = 0 to 1). There are only two exceptions, 1.) for a temporary fault detected by the processor monitoring on the main controller (LV_ERR_TMP_MU_MC) and 2.) for injection cut off (LV_OFF_IV_MON) during engine speed limitation.

Faults	Fault reaction			
	Disable of MTC via main controller (LV_OFF_MTC_MON = 1)	Request engine speed limitation (LV_N_LIM_REQ_MON = 1)	Disable of IV via main controller (LV_OFF_IV_MON = 1)	Disable of MTC and IV via monitoring unit (ACTION_ECM3_RedSwitchOffPath())
LV_ERR_CONV_MON	X	X	Z	Z
LV_ERR_N_32_MON	X	X	Z	Z
LV_ERR_TQ_DIF_P_D_IS_MON	X	X	Z	Z
LV_ERR_TQ_DIF_I_IS_MON	X	X	Z	Z
LV_ERR_TQ_LOSS_MON	X	X	Z	Z
LV_ERR_TQ_MIN_CLU_MON	X	X	Z	Z
LV_ERR_TQ_MAX_CLU_MON	X	X	Z	Z
LV_ERR_OPM_AV_MON	X	X	Z	Z
LV_ERR_PVS_MON	X	X	Z	Z
LV_ERR_TPS_MON	X	X	Z	Z
LV_ERR_MFF_MON	X	X	Z	Z
LV_ERR_TQI_AV_MON	X	X	Z	Z
LV_OFF_IV_N_LIM_ETC_MON*	Z	Z	X/0	Z
LV_ERR_TQI_N_MAX_MON	X	Z	X	X
LV_ERR_MU_MC	X	Z	X	Z
LV_ERR_TMP_MU_MC*	X	Z	X	Z

* reversible during present driving cycle, for actions after reset see formulas or model description

The map shows resulting states of output bits. The proceeding is top-down, e.g. the undermost set fault bit of a column defines the resulting state. Symbol “x” means the related bit is set, “z” the former result of present recurrence will not be changed and “0 “ resets the related bit, if the fault is not set.

LV_OFF_IV_MON must be calculated first by LV_OFF_IV_N_LIM_ETC_MON and than by non reversible fault reactions to be sure that a wrong write operation with LV_OFF_IV_MON = 0 is less prior than fault reactions with permanent fuel cut-off.

Application Condition

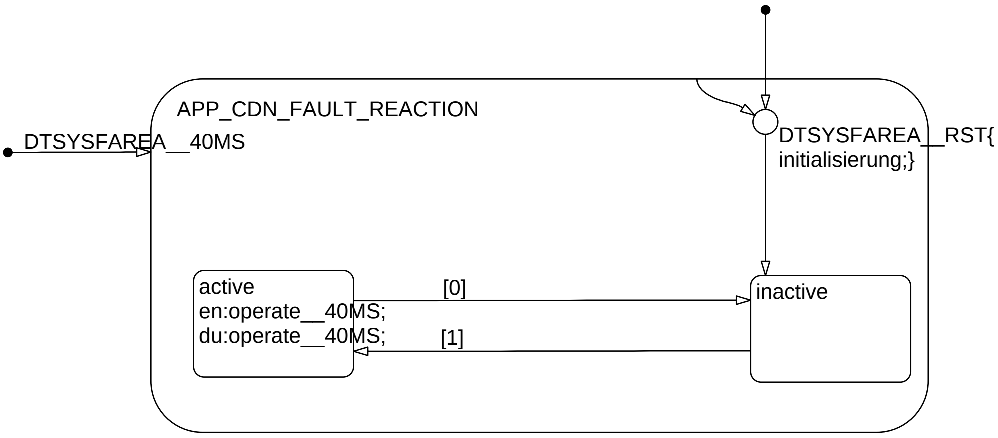
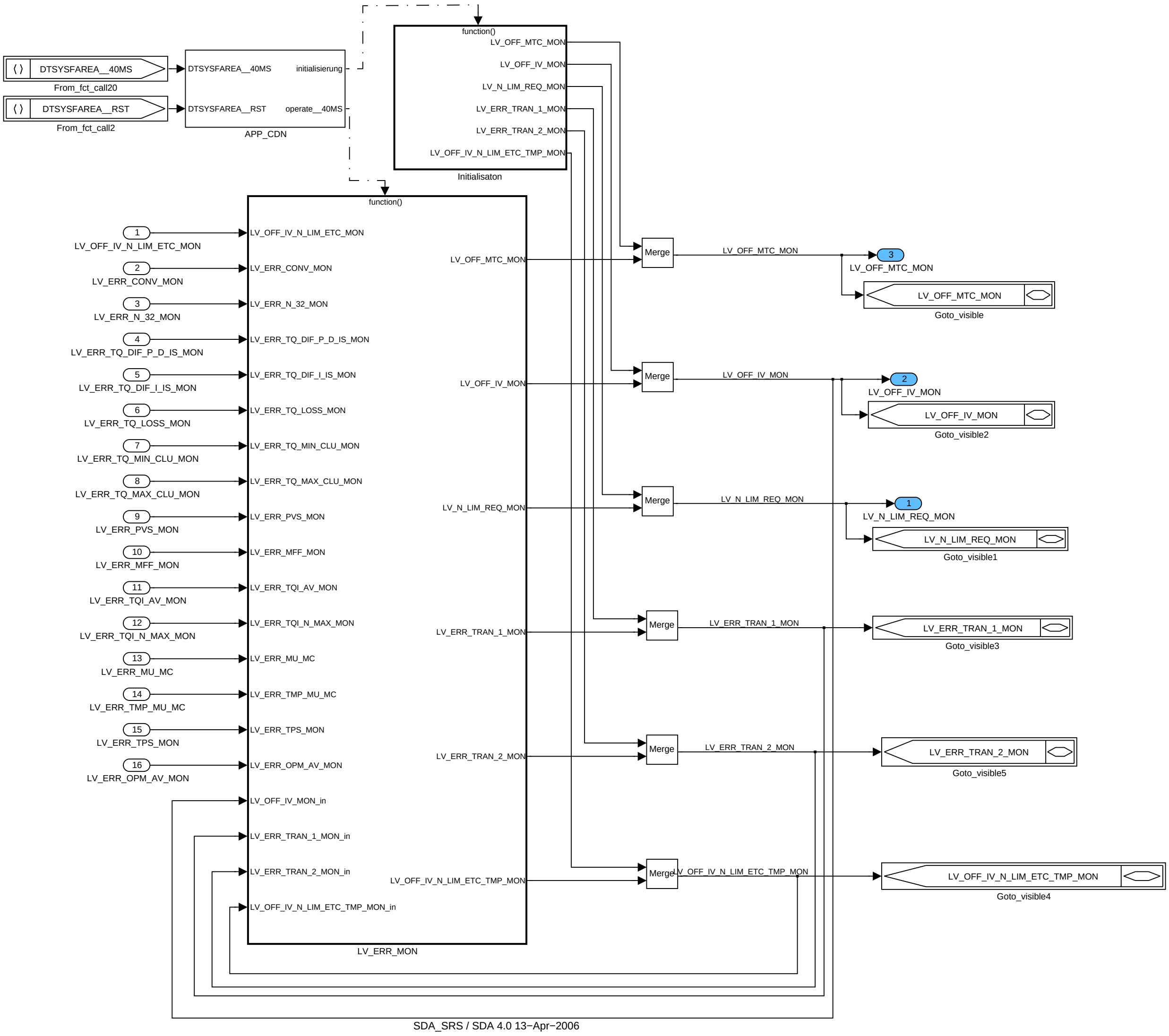


Figure 138 FAREA/ APP_CDN/ STATEFLOW_CHART

Function Description



SDA_SRS / SDA 4.0 13−Apr−2006

Figure 139 FAREA

11.28.1.1 SUBFUNCTION: Initialisaton

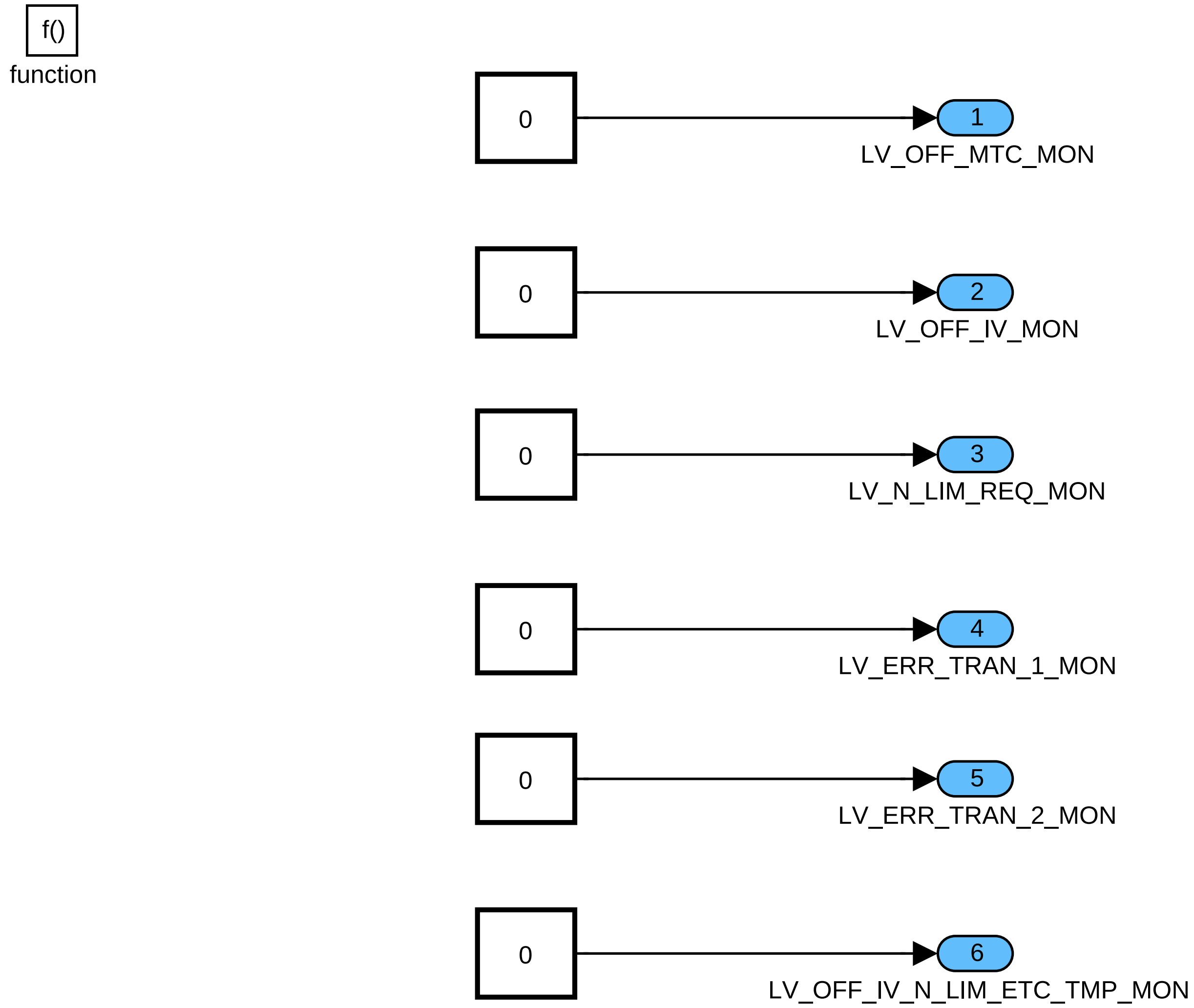


Figure 140 FAREA/ INITIALISATON

11.28.1.2 SUBFUNCTION: LV_ERR_MON

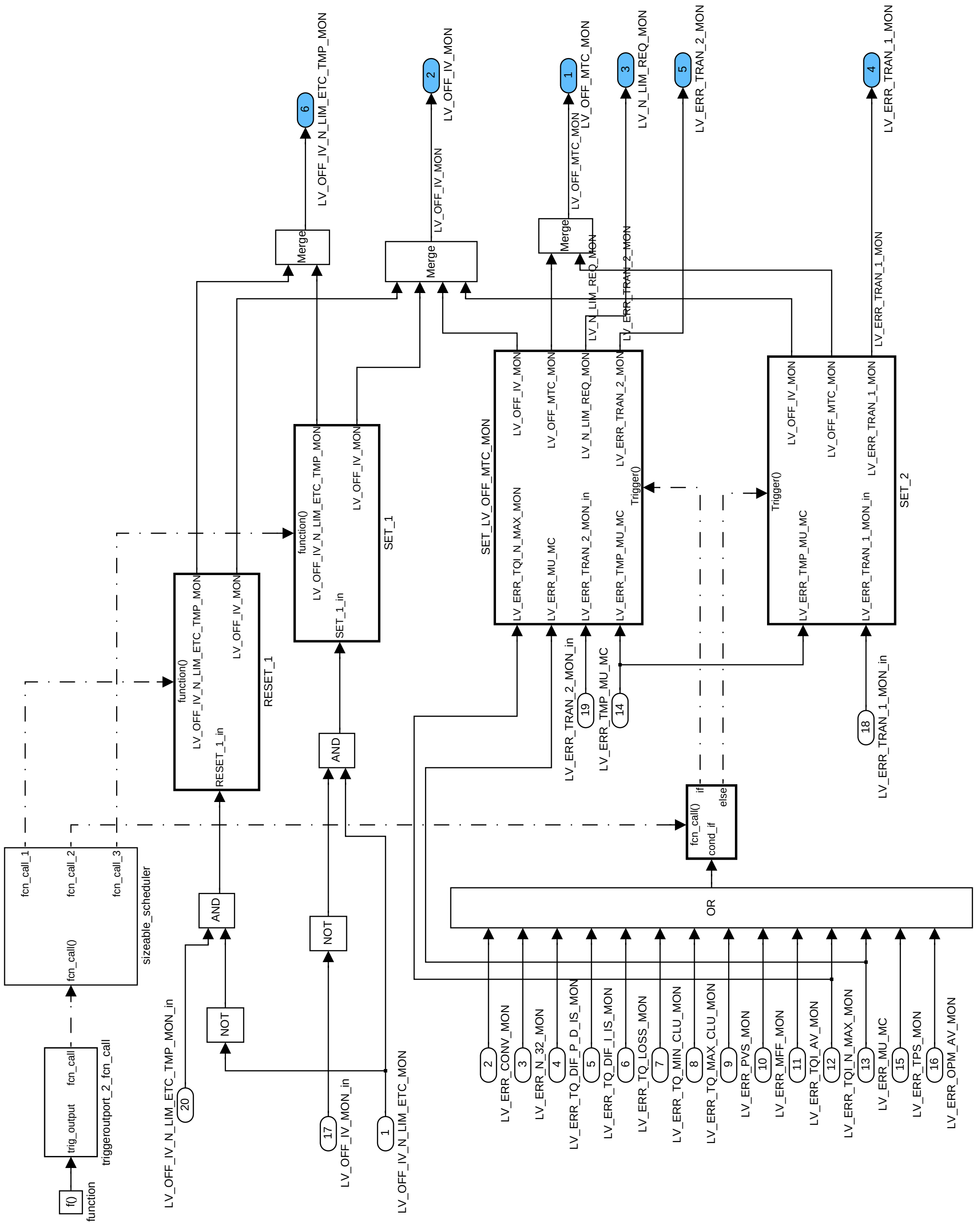


Figure 141 FAREA/ LV_ERR_MON

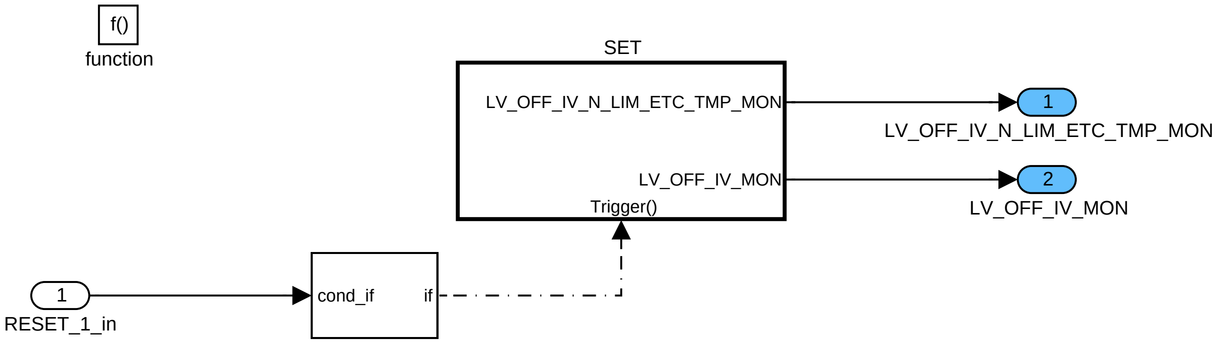


Figure 142 FAREA/ LV_ERR_MON/ RESET_1

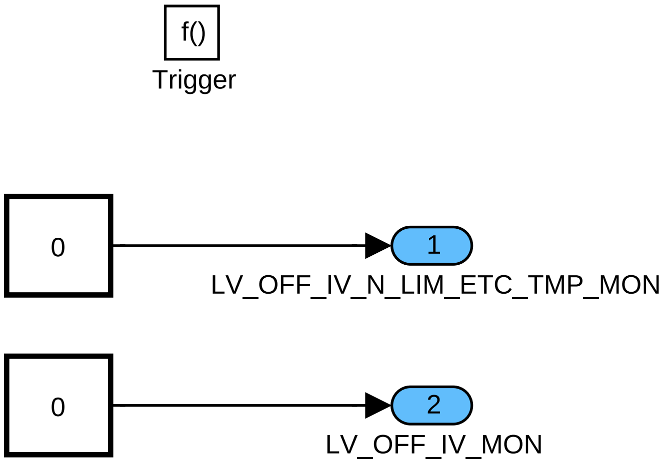


Figure 143 FAREA/ LV_ERR_MON/ RESET_1/ SET

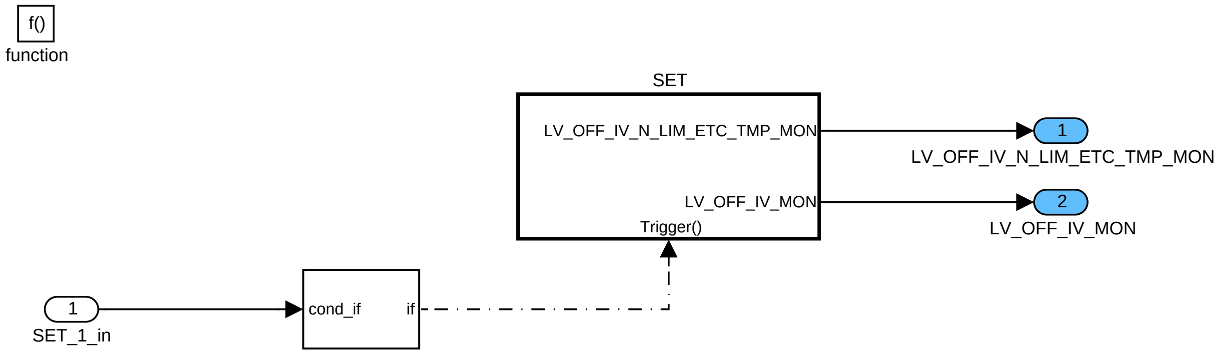


Figure 144 FAREA/ LV_ERR_MON/ SET_1

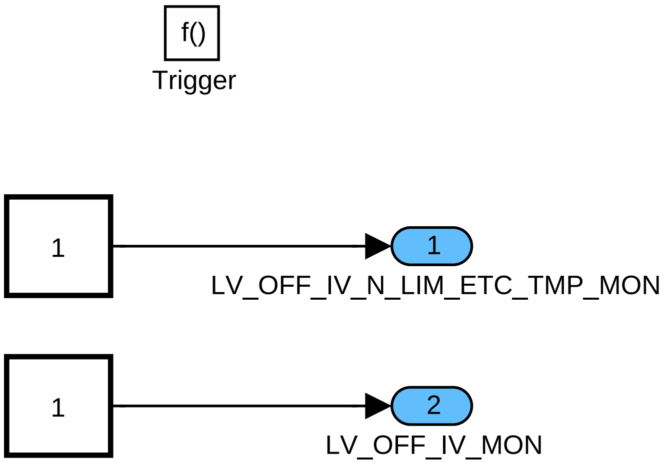


Figure 145 FAREA/ LV_ERR_MON/ SET_1/ SET

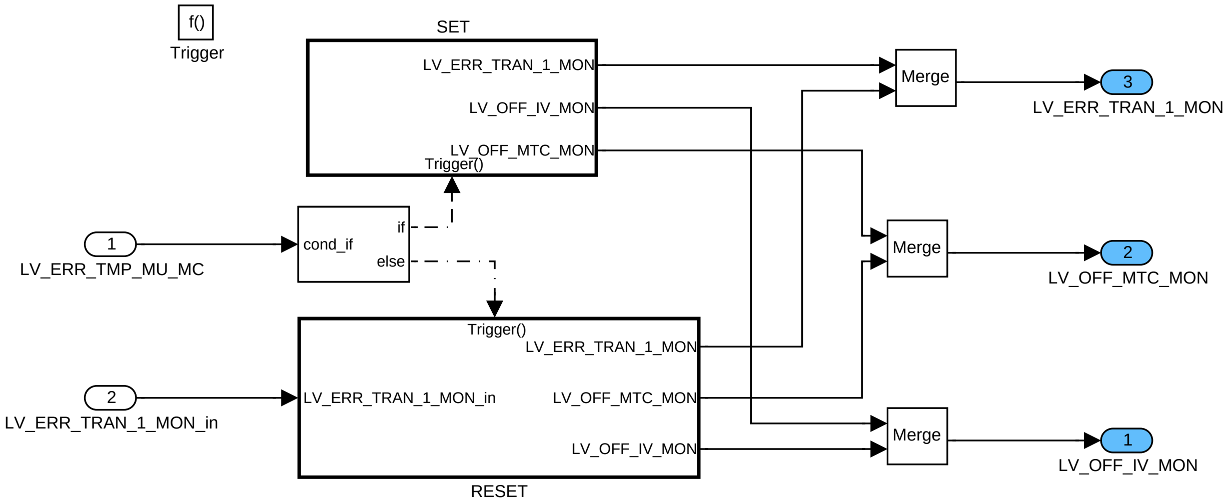


Figure 146 FAREA/ LV_ERR_MON/ SET_2

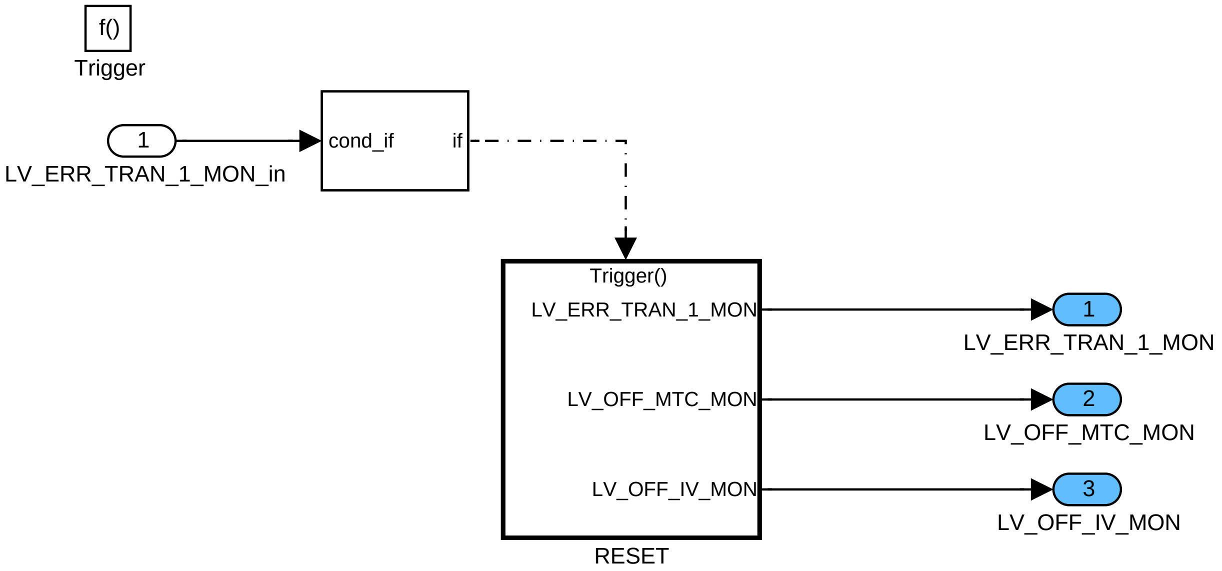


Figure 147 FAREA/ LV_ERR_MON/ SET_2/ RESET

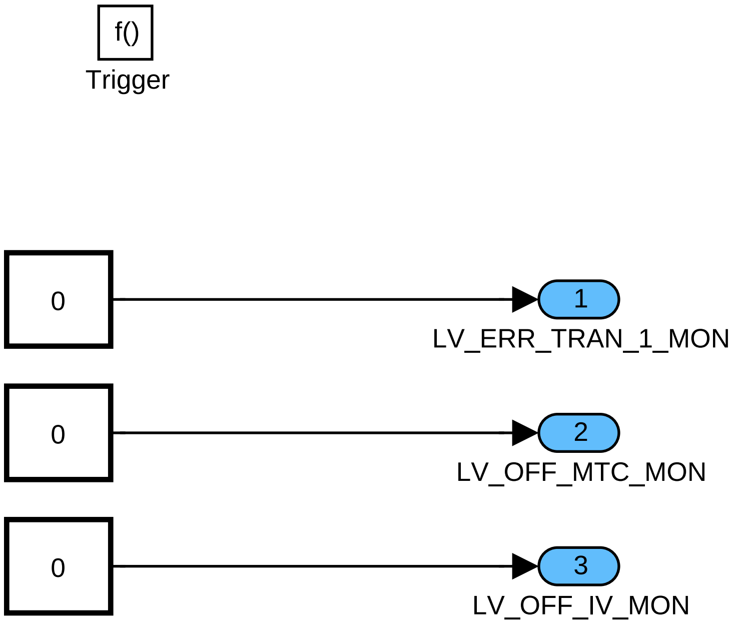


Figure 148 FAREA/ LV_ERR_MON/ SET_2/ RESET/ RESET

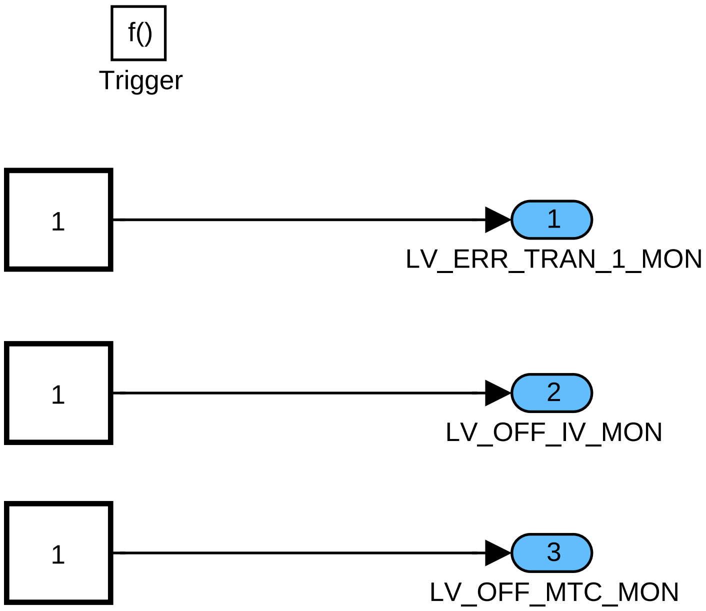


Figure 149 FAREA/ LV_ERR_MON/ SET_2/ SET

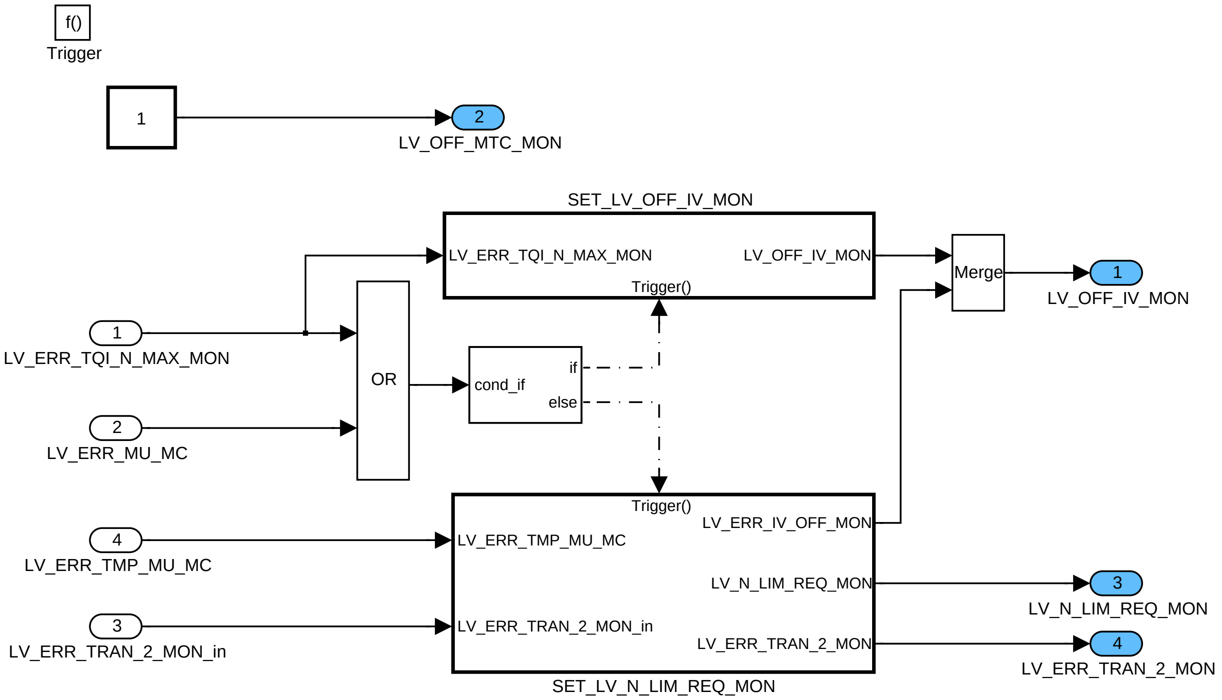


Figure 150 FAREA/ LV_ERR_MON/ SET_LV_OFF_MTC_MON

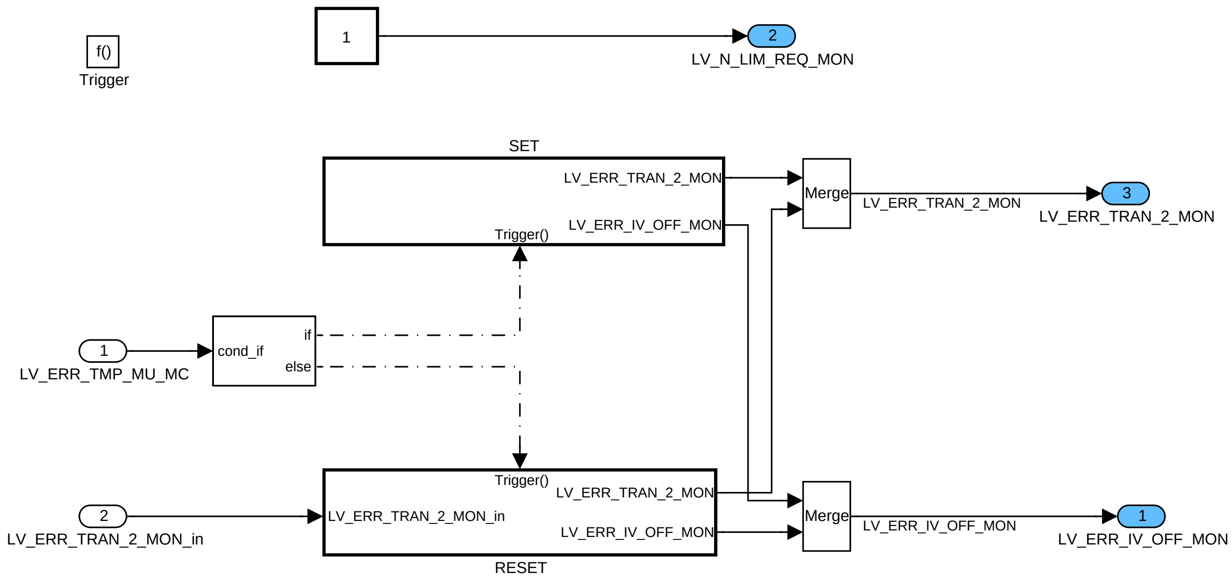


Figure 151 FAREA/ LV_ERR_MON/ SET_LV_OFF_MTC_MON/ SET_LV_N_LIM_REQ_MON

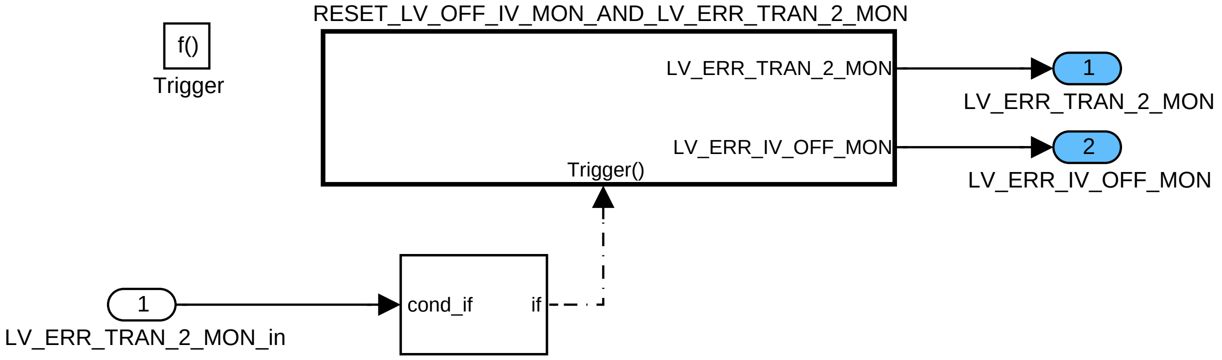


Figure 152 FAREA/ LV_ERR_MON/ SET_LV_OFF_MTC_MON/ SET_LV_N_LIM_REQ_MON/ RESET

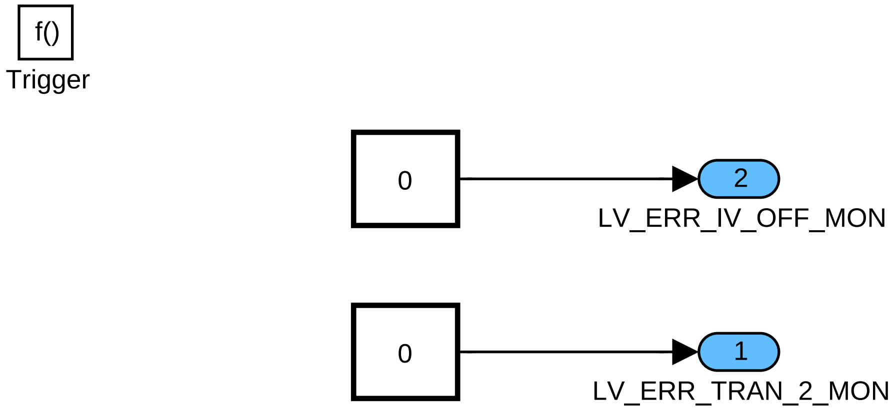


Figure 153 FAREA/ LV_ERR_MON/ SET_LV_OFF_MTC_MON/ SET_LV_N_LIM_REQ_MON/ RESET/ RESET_LV_OFF_IV_MON_AND_LV_ERR_TRAN_2_MON

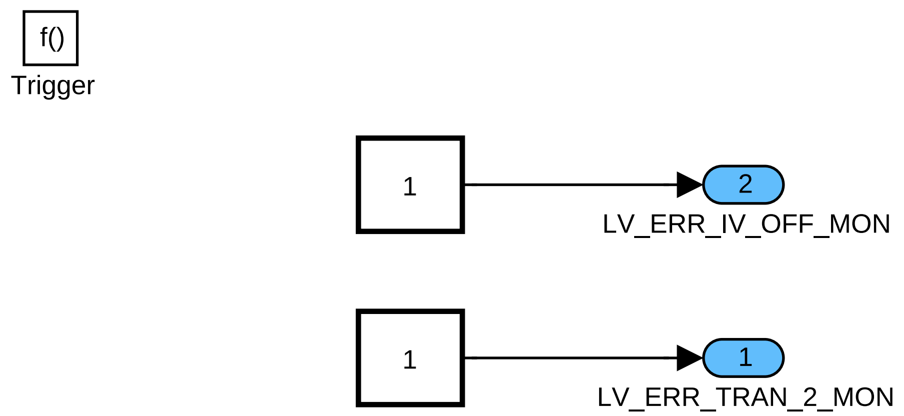


Figure 154 FAREA/ LV_ERR_MON/ SET_LV_OFF_MTC_MON/ SET_LV_N_LIM_REQ_MON/ SET

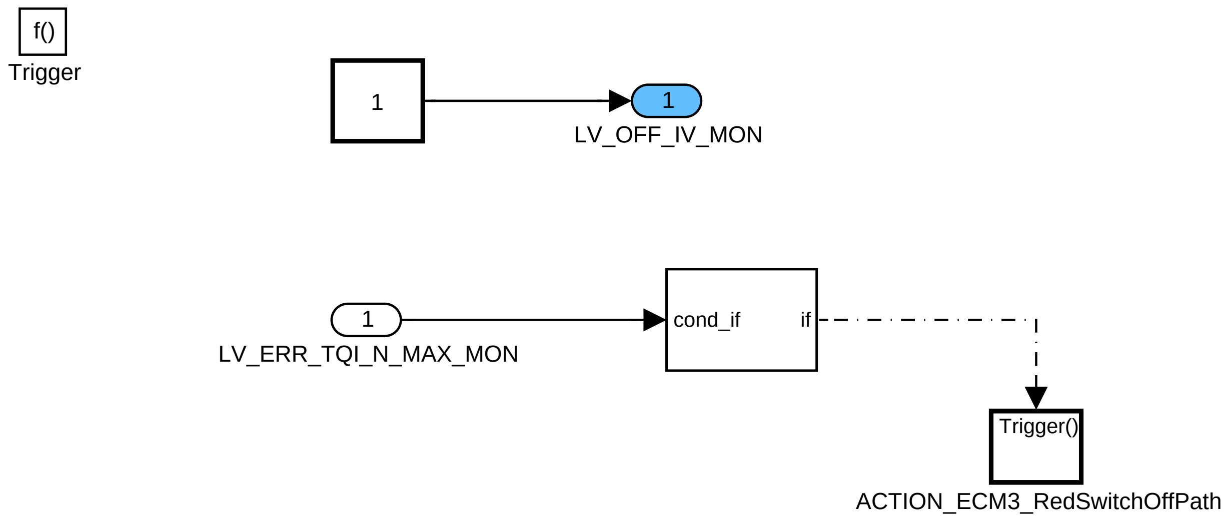


Figure 155 FAREA/ LV_ERR_MON/ SET_LV_OFF_MTC_MON/ SET_LV_OFF_IV_MON

11.29 Error memory management of process monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CONV_MON_1	V/O	0...1H	0...1	1	[-]
A/D conversion Monitoring error reflect to level 1					
LV_ERR_N_32_MON_1	V/O	0...1H	0...1	1	[-]
Engine speed Monitoring error reflect to level 1					
LV_ERR_TPS_MON_1	V/O	0...1H	0...1	1	[-]
TPS Monitoring error reflect to level 1					
LV_ERR_PVS_MON_1	V/O	0...1H	0...1	1	[-]
PVS Monitoring error reflect to level 1					
LV_ERR_CRU_INH_MON_1	V/O	0...1H	0...1	1	[-]
CRU Monitoring error reflect to level 1					
LV_ERR_TQI_AV_MON_1	V/O	0...1H	0...1	1	[-]
Actual indicated engine torque Monitoring error reflect to level 1					
LV_ERR_TQI_N_MAX_MON_1	V/O	0...1H	0...1	1	[-]
Torque limitation Monitoring error reflect to level 1					
LV_ERR_TQI_N_MAX_MON_1_SAVE	- /S	0...1H	0...1	1	[-]
N_LIM monitoring error saved reset resistant					
LV_ERR_MFF_MON_1	V/O	0...1H	0...1	1	[-]
Mass fuel flow monitoring error reflect to level 1					
LV_ERR_TQ_DIF_ISC_MON_1	V/O	0...1H	0...1	1	[-]
ISC Monitoring error reflect to level 1					
LV_ERR_TQ_REQ_MON_1	V/O	0...1H	0...1	1	[-]
Min torque at Clutch Monitoring error reflect to level 1					
LV_ERR_TQ_EXT_MON_1	V/O	0...1H	0...1	1	[-]
Max Torque at Clutch Monitoring error reflect to external torque level 1					
ERR_SYM_CONV_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom A/D conversion Monitoring error reflect to level 1					
ERR_SYM_N_32_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom Engine speed Monitoring error reflect to level 1					
ERR_SYM_TPS_MON_1	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom TPS Monitoring error reflect to level 1					
ERR_SYM_PVS_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom PVS Monitoring error reflect to level 1					
ERR_SYM_CRU_INH_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]

Error symptom CRU Monitoring error reflect to level 1					
ERR_SYM_TQI_AV_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom Actual indicated engine torque Monitoring error reflect to level 1					
ERR_SYM_TQI_N_MAX_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom Torque limitation Monitoring error reflect to level 1					
ERR_SYM_MFF_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom for mass fuel flow monitoring error					
ERR_SYM_TQ_DIF_ISC_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom ISC Monitoring error reflect to level 1					
ERR_SYM_TQ_REQ_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom Min torque at Clutch Monitoring error reflect to level 1					
ERR_SYM_TQ_EXT_MON_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom Max Torque at Clutch Monitoring error reflect to external torque level 1					
LV_END_DIAG_CONV_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis 1 A/D conversion Monitoring error reflect to level 1					
LV_END_DIAG_N_32_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis Engine speed Monitoring error reflect to level 1					
LV_END_DIAG_TPS_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis TPS Monitoring error reflect to level 1					
LV_END_DIAG_PVS_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis PVS Monitoring error reflect to level 1					
LV_END_DIAG_CRU_INH_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis CRU Monitoring error reflect to level 1					
LV_END_DIAG_TQI_AV_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis Actual indicated engine torque Monitoring error reflect to level 1					
LV_END_DIAG_TQI_N_MAX_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis Torque limitation Monitoring error reflect to level 1					
LV_END_DIAG_MFF_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis of mass fuel flow monitoring error reflect to level 1					
LV_END_DIAG_TQ_DIF_ISC_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis ISC Monitoring error reflect to level 1					
LV_END_DIAG_TQ_REQ_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis Min torque at Clutch Monitoring error reflect to level 1					
LV_END_DIAG_TQ_EXT_MON_1	V/O	0...1H	0...1	1	[-]
End of diagnosis Max Torque at Clutch Monitoring error reflect to external torque level 1					
LV_CDN_DIAG_CONV_MON_1	V	0...1H	0...1	1	[-]

Diagnosis condition A/D conversion Monitoring error reflect to level 1					
LV_CDN_DIAG_N_32_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition Engine speed Monitoring error reflect to level 1					
LV_CDN_DIAG_TPS_MON_1	V/O	0...1H	0...1	1	[-]
Diagnosis condition TPS Monitoring error reflect to level 1					
LV_CDN_DIAG_PVS_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition PVS Monitoring error reflect to level 1					
LV_CDN_DIAG_CRU_INH_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition CRU Monitoring error reflect to level 1					
LV_CDN_DIAG_TQI_AV_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition Actual indicated engine torque Monitoring error reflect to level 1					
LV_CDN_DIAG_TQI_N_MAX_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition Torque limitation Monitoring error reflect to level 1					
LV_CDN_DIAG_MFF_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition mass fuel flow monitoring error reflect to level 1					
LV_CDN_DIAG_TQ_DIF_ISC_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition ISC Monitoring error reflect to level 1					
LV_CDN_DIAG_TQ_REQ_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition Min torque at Clutch Monitoring error reflect to level 1					
LV_CDN_DIAG_TQ_EXT_MON_1	V	0...1H	0...1	1	[-]
Diagnosis condition Max Torque at Clutch Monitoring error reflect to external torque level 1					

Input data:

LV_ERR_CONV_MON	LV_ERR_MSR_INH_MON	LV_ERR_N_32_MON
LV_ERR_PVS_MON	LV_ERR_MFF_MON	LV_ERR_GS_INH_MON
LV_ERR_TQI_AV_MON	LV_ERR_TQ_DIF_P_D_IS_MON	LV_ERR_TQ_DIF_I_IS_MON
LV_ERR_TQ_LOSS_MON	LV_ERR_TQ_MIN_CLU_MON	LV_ERR_TQ_MAX_CLU_MON
LV_TQI_MON_ACT_MON	LV_ERR_TQI_N_MAX_MON	LV_ERR_CRU_MON
LV_ERR_SOF_INH_MON	LV_ERR_DCC_INH_MON	LV_ERR_AMT_INH_MON
LV_ERR_LDM_INH_MON	VB	LV_ERR_TPS_MON
LV_ERR_OPM_AV_MON		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VB_MIN_MON_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
VB Threshold for activation of monitoring diagnosis					

Import actions:

ACTION_ERRM_NoFilterSymptom(IN< XX >, IN< lv_cdn_diag_XX >, IN< err_sym_XX >, IN<lv_err_set_XX>, IN<lv_err_reset_XX>, IN< lv_end_diag_XX >, OUT< LV_ERR_XX >)

ACTION_ERRM_NoFilterReset(IN< XX >, OUT< LV_ERR_XX >)

FUNCTION DESCRIPTION:

General information:

When a fault is detected by the level 2 of the ETC-Monitoring concept and the reaction is released in the module ‘fault reaction of process monitoring’, the fault is transferred to the error memory management.

The memory area of all output data variables of this module is not permanently RAM tested, the memory area of this module is not cyclic ROM tested.

Description:

Since the different faults can occur in a successive way, they are stored in different locations of the error memory.

All faults LV_ERR_XX_MON_1 are stored in separate locations, in order to be able to analyze a scenario of successive faults. If more than one errors stored in one location, then it will be detected by different error symptoms.

The information whether a warning lamp (ETC MIL) is switched on in case of an error can be found in the general OBD error code table.

Application conditions:

Activation: at every engine state
Deactivation: -
Initialization:
- for condition see ‘Application Incidences of Process Monitoring chapter 1.1’

-ACTION_ERRM_NoFilterReset(IN<CONV_MON>, OUT<LV_ERR_CONV_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<N_32_MON>, OUT<LV_ERR_N_32_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<TPS_MON>, OUT<LV_ERR_TPS_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<PVS_MON>, OUT<LV_ERR_PVS_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<CRU_INH_MON>, OUT<LV_ERR_CRU_INH_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<MFF_MON>, OUT<LV_ERR_MFF_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<TQ_DIF_ISC_MON>, OUT<LV_ERR_TQ_DIF_ISC_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<TQ_REQ_MON>, OUT<LV_ERR_TQ_REQ_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<TQ_EXT_MON>, OUT<LV_ERR_TQ_EXT_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<TQI_AV_MON>, OUT<LV_ERR_TQI_AV_MON_1>)
-ACTION_ERRM_NoFilterReset(IN<TQI_N_MAX_MON>, OUT<LV_ERR_TQI_N_MAX_MON_1>)

(these actions erases filter data (LV_CDN_DIAG_xx = 0, LV_ERR_xx = 0, ERR_SYM_xx = NO_SYM

- For LV_ERR_N_MAX_MON_SAVE = 0, if a power up is recognized. A power up is detected, if the variables in the reset safe RAM are not consitend.

Update Rate: 40 ms

11.29.1 Error memory management of LV_ERR_CONV_MON_1

Formula section:

```
LV_CDN_DIAG_CONV_MON_1 = 1


If    (LV_ERR_CONV_MON = 1
Then
      lv_err_reset_CONV_MON_1 = 0;
      lv_err_set_CONV_MON_1 = 1;                {failure detected}
      ERR_SYM_CONV_MON_1 = SYM_3;
      LV_END_DIAG_CONV_MON_1 = 1;
Else
      lv_err_set_CONV_MON_1 = 0;
      lv_err_reset_CONV_MON_1 = 1;    {no failure detected}
      ERR_SYM_CONV_MON_1 = NO_SYM;
      If { delay 520ms } then
      /* After the level2 fault reaction time of 500ms the errors are updated */
            LV_END_DIAG_CONV_MON_1 = 1;
      Endif
Endif
```

For error management treatment the following action is called:

ACTION_ERRM_NoFilterSymptom(IN<CONV_MON>, IN<LV_CDN_DIAG_CONV_MON_1>, IN<ERR_SYM_CONV_MON_1>, IN<lv_err_set_CONV_MON_1>, IN<lv_err_reset_CONV_MON_1>, IN<LV_END_DIAG_CONV_MON_1>, OUT<LV_ERR_CONV_MON_1>)

11.29.2 Error memory management of LV_ERR_N_32_MON_1

Formula section:

```
LV_CDN_DIAG_N_32_MON_1 = 1


If    (LV_ERR_N_32_MON = 1
Then
      lv_err_reset_N_32_MON_1 = 0;
      lv_err_set_N_32_MON_1 = 1;                {failure detected}
      ERR_SYM_N_32_MON_1 = SYM_3;
```

```
        LV_END_DIAG_N_32_MON_1 = 1;
    Else
        lv_err_set_N_32_MON_1 = 0;
        lv_err_reset_N_32_MON_1 = 1;      {no failure detected}
        ERR_SYM_N_32_MON_1 = NO_SYM;
        If { delay 520ms } then
        /* After the level2 fault reaction time of 500ms the errors are updated */
                LV_END_DIAG_N_32_MON_1 = 1;
        Endif
    Endif
        For error management treatment the following action is called:
        ACTION_ERRM_NoFilterSymptom(IN<N_32_MON>,
        IN<LV_CDN_DIAG_N_32_MON_1>, IN<ERR_SYM_N_32_MON_1>,
        IN<lv_err_set_N_32_MON_1>, IN<lv_err_reset_N_32_MON_1>,
        IN<LV_END_DIAG_N_32_MON_1>, OUT<LV_ERR_N_32_MON_1>)
```

11.29.3 Error memory management of LV_ERR_TPS_MON_1

Formula section:

```
    If          LV_TQI_MON_ACT_MON = 1
    Then        LV_CDN_DIAG_TPS_MON_1 = 1

    If    (LV_ERR_TPS_MON = 1
    Then
        lv_err_reset_TPS_MON_1 = 0;
        lv_err_set_TPS_MON_1 = 1;                {failure detected}
        ERR_SYM_TPS_MON_1 = SYM_3;
        LV_END_DIAG_TPS_MON_1 = 1;
    Else
        lv_err_set_TPS_MON_1 = 0;
        lv_err_reset_TPS_MON_1 = 1;              {no failure detected}
        ERR_SYM_TPS_MON_1 = NO_SYM;
        If { delay 520ms } then
        /* After the level2 fault reaction time of 500ms the errors are updated */
                LV_END_DIAG_TPS_MON_1 = 1;
```

```
        Endif
    Endif
        For error management treatment the following action is called:
        ACTION_ERRM_NoFilterSymptom(IN<TPS_MON>,
        IN<LV_CDN_DIAG_TPS_MON_1>, IN<ERR_SYM_TPS_MON_1>,
        IN<lv_err_set_TPS_MON_1>, IN<lv_err_reset_TPS_MON_1>,
        IN<LV_END_DIAG_TPS_MON_1>, OUT<LV_ERR_TPS_MON_1>)
```

11.29.4 Error memory management of LV_ERR_PVS_MON_1

Formula section:

```
    LV_CDN_DIAG_PVS_MON_1 = 1

    If   (LV_ERR_PVS_MON = 1
    Then
        lv_err_reset_PVS_MON_1 = 0;
        lv_err_set_PVS_MON_1 = 1;                {failure detected}
        ERR_SYM_PVS_MON_1 = SYM_3;
        LV_END_DIAG_PVS_MON_1 = 1;
    Else
        lv_err_set_PVS_MON_1 = 0;
        lv_err_reset_PVS_MON_1 = 1;              {no failure detected}
        ERR_SYM_PVS_MON_1 = NO_SYM;
        If { delay 520ms } then
        /* After the level2 fault reaction time of 500ms the errors are updated */
                LV_END_DIAG_PVS_MON_1 = 1;
        Endif
    Endif
        For error management treatment the following action is called:
        ACTION_ERRM_NoFilterSymptom(IN<PVS_MON>,
        IN<LV_CDN_DIAG_PVS_MON_1>, IN<ERR_SYM_PVS_MON_1>,
        IN<lv_err_set_PVS_MON_1>, IN<lv_err_reset_PVS_MON_1>,
        IN<LV_END_DIAG_PVS_MON_1>, OUT<LV_ERR_PVS_MON_1>)
```

11.29.5 Error memory management of LV_ERR_CRU_INH_MON_1

Formula section:

```
If          LV_TQI_MON_ACT_MON = 1
Then        LV_CDN_DIAG_CRU_INH_MON_1 = 1;

If    (LV_ERR_CRU_MON = 1)
Then
      lv_err_reset_CRU_INH_MON_1 = 0;
      lv_err_set_CRU_INH_MON_1 = 1;           {failure detected}
      ERR_SYM_CRU_INH_MON_1 = SYM_0;
      LV_END_DIAG_CRU_INH_MON_1 = 1;
elseif  (LV_ERR_DCC_INH_MON = 1)
Then
      lv_err_reset_CRU_INH_MON_1 = 0;
      lv_err_set_CRU_INH_MON_1 = 1;           {failure detected}
      ERR_SYM_CRU_INH_MON_1 = SYM_1;
      LV_END_DIAG_CRU_INH_MON_1 = 1;
elseif  (LV_ERR_LDM_INH_MON = 1)
Then
      lv_err_reset_CRU_INH_MON_1 = 0;
      lv_err_set_CRU_INH_MON_1 = 1;           {failure detected}
      ERR_SYM_CRU_INH_MON_1 = SYM_2;
      LV_END_DIAG_CRU_INH_MON_1 = 1;
Else
      lv_err_set_CRU_INH_MON_1 = 0;
      lv_err_reset_CRU_INH_MON_1 = 1;                {no failure detected}
      ERR_SYM_CRU_INH_MON_1 = NO_SYM;
      If { delay 520ms } then
      /* After the level2 fault reaction time of 500ms the errors are updated */
            LV_END_DIAG_CRU_INH_MON_1 = 1;
      Endif
Endif
```

For error management treatment the following action is called:

ACTION_ERRM_NoFilterSymptom(IN<CRU_INH_MON>, IN<LV_CDN_DIAG_CRU_INH_MON_1>, IN<ERR_SYM_CRU_INH_MON_1>, IN<lv_err_set_CRU_INH_MON_1>, IN<lv_err_reset_CRU_INH_MON_1>, IN<LV_END_DIAG_CRU_INH_MON_1>, OUT<LV_ERR_CRU_INH_MON_1>)

11.29.6 Error memory management of LV_ERR_TQI_AV_MON_1

Formula section:

```
If        LV_TQI_MON_ACT_MON = 1
Then      LV_CDN_DIAG_TQI_AV_MON_1 = 1;

If   (LV_ERR_TQI_AV_MON = 1)
Then
    lv_err_reset_TQI_AV_MON_1 = 0;
    lv_err_set_TQI_AV_MON_1 = 1;               {failure detected}
    ERR_SYM_TQI_AV_MON_1 = SYM_3;
    LV_END_DIAG_TQI_AV_MON_1 = 1;
Else
    lv_err_set_TQI_AV_MON_1 = 0;
    lv_err_reset_TQI_AV_MON_1 = 1;             {no failure detected}
    ERR_SYM_TQI_AV_MON_1 = NO_SYM;
    If { delay 520ms } then
    /* After the level2 fault reaction time of 500ms the errors are updated */
                LV_END_DIAG_TQI_AV_MON_1 = 1;
    Endif
Endif
```

For error management treatment the following action is called:

ACTION_ERRM_NoFilterSymptom(IN<TQI_AV_MON>, IN<LV_CDN_DIAG_TQI_AV_MON_1>, IN<ERR_SYM_TQI_AV_MON_1>, IN<lv_err_set_TQI_AV_MON_1>, IN<lv_err_reset_TQI_AV_MON_1>, IN<LV_END_DIAG_TQI_AV_MON_1>, OUT<LV_ERR_TQI_AV_MON_1>)

11.29.7 Error memory management of LV_ERR_TQI_N_MAX_MON_1

The error LV_ERR_TQI_N_MAX_MON_1 triggers the redundant switch off path (see "Fault Reaction of Process Monitoring") during the actual duty cycle. This finally leads to a reset after debouncing (see "Processor Monitoring"). The entry into the error memory has to be performed past reset. Therefore it is needful to store the error information reset safe; LV_ERR_TQI_N_MAX_MON_1_SAVE is located in the reset safe RAM area. The entry into the error memory is done past reset by evaluation of the information stored before reset.

Application conditions:

Initialisation: LV_ERR_TQI_N_MAX_MON_1_SAVE = 0
only after power up, no initialization after reset

Formula section:

```
If          LV_IGK = 1 and
            VB > C_VB_MIN_MON_DIAG
Then        LV_CDN_DIAG_TQI_N_MAX_MON_1 = 1;


If(1)       (LV_ERR_TQI_N_MAX_MON_1_SAVE = 1)/* error detected last duty cycle */
Then(1)
   /* error not jet passed to ERRM =>  transfer error information to ERRM*/
   lv_err_reset_TQI_N_MAX_MON_1 = 0;
   lv_err_set_TQI_N_MAX_MON_1 = 1;
   ERR_SYM_TQI_N_MAX_MON_1 = SYM_3;
   /* environment data are already reset safe stored before reset */
   LV_END_DIAG_TQI_N_MAX_MON_1 = 1;
   /* avoid more than one entry into error memory */
   if (2) {delay 120ms}
   Then(2)
     /* this delay time periode depends on Processor Monitoring */
     /* After this time periode the reset has to be carried out */
     LV_ERR_TQI_N_MAX_MON_1_SAVE = 0;
   Endif(2)
Else(1)     /* error not detected last duty cycle */
     If (3)(LV_ERR_TQI_N_MAX_MON_1 = 1)
     Then(3)        /* error was stored last recurrence */
            lv_err_reset_TQI_N_MAX_MON_1 = 0;
            lv_err_set_TQI_N_MAX_MON_1 = 1;
            ERR_SYM_TQI_N_MAX_MON_1 = SYM_3;
```

```
            LV_END_DIAG_TQI_N_MAX_MON_1 = 1;
        Else(3)
            lv_err_set_TQI_N_MAX_MON_1 = 0;
            lv_err_reset_TQI_N_MAX_MON_1 = 1;          {no failure detected}
            ERR_SYM_TQI_N_MAX_MON_1 = NO_SYM;
          If (4){ delay 520ms }
          Then(4)
            /* After the level2 fault reaction time of 500ms the errors are updated */
            LV_END_DIAG_TQI_N_MAX_MON_1 = 1;
          Endif(4)
        Endif(3)
    Endif(1)
    If(5) (LV_ERR_TQI_N_MAX_MON = 1)
    Then(5)   /* error detected in actual duty cycle */
        If(6)   (LV_ERR_TQI_N_MAX_MON_1_SAVE = 0)
        Then(6)
        store ENVDs reset save --> refer to ERRM of process monitoring (Appl.Inc.)
        endif(6)
    /* store information reset save and avoid saving environment data next recurrence */
    LV_ERR_TQI_N_MAX_MON_1_SAVE =1;
    Endif(5)
```

For error management treatment the following action is called:

ACTION_ERRM_NoFilterSymptom(IN<TQI_N_MAX_MON>, IN<LV_CDN_DIAG_TQI_N_MAX_MON_1>, IN<ERR_SYM_TQI_N_MAX_MON_1>, IN<lv_err_set_TQI_N_MAX_MON_1>, IN<lv_err_reset_TQI_N_MAX_MON_1>, IN<LV_END_DIAG_TQI_N_MAX_MON_1>, OUT<LV_ERR_TQI_N_MAX_MON_1>)

11.29.8 Error memory management of LV_ERR_MFF_MON_1

Formula section:

```
If      LV_TQI_MON_ACT_MON = 1
Then    LV_CDN_DIAG_MFF_MON_1 = 1;

If   (LV_ERR_MFF_MON = 1)
```

```
Then
    lv_err_reset_MFF_MON_1 = 0;
    lv_err_set_MFF_MON_1 = 1;               {failure detected}
    ERR_SYM_MFF_MON_1 = SYM_3;
    LV_END_DIAG_MFF_MON_1 = 1;
Else if (LV_ERR_OPM_AV_MON = 1)
Then
    lv_err_reset_MFF_MON_1 = 0;
    lv_err_set_MFF_MON_1 = 1;               {failure detected}
    ERR_SYM_MFF_MON_1 = SYM_2;
    LV_END_DIAG_MFF_MON_1 = 1;
Else
    lv_err_set_MFF_MON_1 = 0;
    lv_err_reset_MFF_MON_1 = 1;             {no failure detected}
    ERR_SYM_MFF_MON_1 = NO_SYM;
    If { delay 520ms }
    Then
    /* After the level2 fault reaction time of 500ms the errors are updated */
            LV_END_DIAG_MFF_MON_1 = 1;
    Endif
Endif
```

<u>For error management treatment the following action is called:</u>

ACTION_ERRM_NoFilterSymptom(IN<MFF_MON>, IN<LV_CDN_DIAG_MFF_MON_1>, IN<ERR_SYM_MFF_MON_1>, IN<lv_err_set_MFF_MON_1>, IN<lv_err_reset_MFF_MON_1>, IN<LV_END_DIAG_MFF_MON_1>, OUT<LV_ERR_MFF_MON_1>)

11.29.9 Error memory management of LV_ERR_TQ_DIF_ISC_MON_1

Formula section:

```
If      LV_TQI_MON_ACT_MON = 1
Then    LV_CDN_DIAG_TQ_DIF_ISC_MON_1 = 1;

If  (LV_ERR_TQ_DIF_I_IS_MON = 1)
Then
```

```
        lv_err_reset_TQ_DIF_ISC_MON_1 = 0;
        lv_err_set_TQ_DIF_ISC_MON_1 = 1;              {failure detected}
        ERR_SYM_TQ_DIF_ISC_MON_1 = SYM_2;
        LV_END_DIAG_TQ_DIF_ISC_MON_1 = 1;
    Elseif (LV_ERR_TQ_DIF_P_D_IS_MON = 1)
    then
        lv_err_reset_TQ_DIF_ISC_MON_1 = 0;
        lv_err_set_TQ_DIF_ISC_MON_1 = 1;              {failure detected}
        ERR_SYM_TQ_DIF_ISC_MON_1 = SYM_3;
        LV_END_DIAG_TQ_DIF_ISC_MON_1 = 1;
    Else
        lv_err_set_TQ_DIF_ISC_MON_1 = 0;
        lv_err_reset_TQ_DIF_ISC_MON_1 = 1;            {no failure detected}
        ERR_SYM_TQ_DIF_ISC_MON_1 = NO_SYM;
            If { delay 520ms }
            then
            /* After the level2 fault reaction time of 500ms the errors are updated */
            LV_END_DIAG_TQ_DIF_ISC_MON_1 = 1;
            Endif
    Endif
```

For error management treatment the following action is called:

```
ACTION_ERRM_NoFilterSymptom(IN<TQ_DIF_ISC_MON>,
IN<LV_CDN_DIAG_TQ_DIF_ISC_MON_1>, IN<ERR_SYM_TQ_DIF_ISC_MON_1>,
IN<lv_err_set_TQ_DIF_ISC_MON_1>, IN<lv_err_reset_TQ_DIF_ISC_MON_1>,
IN<LV_END_DIAG_TQ_DIF_ISC_MON_1>, OUT<LV_ERR_TQ_DIF_ISC_MON_1>)
```

11.29.10 Error memory management of LV_ERR_TQ_REQ_MON_1

Formula section:

```
If      LV_TQI_MON_ACT_MON = 1
Then    LV_CDN_DIAG_TQ_REQ_MON_1 = 1;

If  (LV_ERR_TQ_MAX_CLU_MON = 1)
Then
    lv_err_reset_TQ_REQ_MON_1 = 0;
    lv_err_set_TQ_REQ_MON_1 = 1;          {failure detected}
```

```
        ERR_SYM_TQ_REQ_MON_1 = SYM_0;
        LV_END_DIAG_TQ_REQ_MON_1 = 1;
    Elseif (LV_ERR_TQ_MIN_CLU_MON = 1)
    then
        lv_err_reset_TQ_REQ_MON_1 = 0;
        lv_err_set_TQ_REQ_MON_1 = 1;            {failure detected}
        ERR_SYM_TQ_REQ_MON_1 = SYM_1;
        LV_END_DIAG_TQ_REQ_MON_1 = 1;
    Elseif (LV_ERR_TQ_LOSS_MON = 1)
    then
        lv_err_reset_TQ_REQ_MON_1 = 0;
        lv_err_set_TQ_REQ_MON_1 = 1;            {failure detected}
        ERR_SYM_TQ_REQ_MON_1 = SYM_2;
        LV_END_DIAG_TQ_REQ_MON_1 = 1;
    Elseif (LV_ERR_SOF_INH_MON = 1)
    then
        lv_err_reset_TQ_REQ_MON_1 = 0;
        lv_err_set_TQ_REQ_MON_1 = 1;            {failure detected}
        ERR_SYM_TQ_REQ_MON_1 = SYM_3;
        LV_END_DIAG_TQ_REQ_MON_1 = 1;
    Else
        lv_err_set_TQ_REQ_MON_1 = 0;
        lv_err_reset_TQ_REQ_MON_1 = 1;          {no failure detected}
        ERR_SYM_TQ_REQ_MON_1 = NO_SYM;
            If { delay 520ms }
            then
            /* After the level2 fault reaction time of 500ms the errors are updated */
            LV_END_DIAG_TQ_REQ_MON_1 = 1;
            Endif
    Endif
```

For error management treatment the following action is called:

```
ACTION_ERRM_NoFilterSymptom(IN<TQ_REQ_MON>,
IN<LV_CDN_DIAG_TQ_REQ_MON_1>, IN<ERR_SYM_TQ_REQ_MON_1>,
IN<lv_err_set_TQ_REQ_MON_1>, IN<lv_err_reset_TQ_REQ_MON_1>,
IN<LV_END_DIAG_TQ_REQ_MON_1>, OUT<LV_ERR_TQ_REQ_MON_1>)
```

11.29.11 Error memory management of LV_ERR_TQ_EXT_MON_1

Formula section:

```
If      LV_TQI_MON_ACT_MON = 1
Then    LV_CDN_DIAG_TQ_EXT_MON_1 = 1;

If  (LV_ERR_MSR_INH_MON = 1)
Then
    lv_err_reset_TQ_EXT_MON_1 = 0;
    lv_err_set_TQ_EXT_MON_1 = 1;            {failure detected}
    ERR_SYM_TQ_EXT_MON_1 = SYM_0;
    LV_END_DIAG_TQ_EXT_MON_1 = 1;
Elseif (LV_ERR_AMT_INH_MON = 1)
then
    lv_err_reset_TQ_EXT_MON_1 = 0;
    lv_err_set_TQ_EXT_MON_1 = 1;            {failure detected}
    ERR_SYM_TQ_EXT_MON_1 = SYM_2;
    LV_END_DIAG_TQ_EXT_MON_1 = 1;
Elseif (LV_ERR_GS_INH_MON = 1)
then
    lv_err_reset_TQ_EXT_MON_1 = 0;
    lv_err_set_TQ_EXT_MON_1 = 1;            {failure detected}
    ERR_SYM_TQ_EXT_MON_1 = SYM_3;
    LV_END_DIAG_TQ_EXT_MON_1 = 1;
Else
    lv_err_set_TQ_EXT_MON_1 = 0;
    lv_err_reset_TQ_EXT_MON_1 = 1;          {no failure detected}
    ERR_SYM_TQ_EXT_MON_1 = NO_SYM;
        If { delay 520ms }
        then
        /* After the level2 fault reaction time of 500ms the errors are updated */
        LV_END_DIAG_TQ_EXT_MON_1 = 1;
        Endif
Endif
```

For error management treatment the following action is called:

ACTION_ERRM_NoFilterSymptom(IN<TQ_EXT_MON>,
IN<LV_CDN_DIAG_TQ_EXT_MON_1>, IN<ERR_SYM_TQ_EXT_MON_1>,
IN<lv_err_set_TQ_EXT_MON_1>, IN<lv_err_reset_TQ_EXT_MON_1>,
IN<LV_END_DIAG_TQ_EXT_MON_1>, OUT<LV_ERR_TQ_EXT_MON_1>)

11.29.12 Diagnosis for AFS request by monitoring (Diagnostic file – STD_INI filter)

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
CTR_SWI_AFS_MON	V/O/S	0...FFH	0...255	1	[-]
Reset resistant counter for number of AFS requests by monitoring					
LV_ERR_SWI_AFS_MON	O/V	0...1H	0...1	1	[-]
Indicator: maximum number of AFS request by MFF-monitoring has been reached					
LV_END_DIAG_SWI_AFS_MON	V/O	0...1H	0...1	1	[-]
Diagnostic of AFS request by monitoring performed at least one time					
LV_CDN_DIAG_SWI_AFS_MON	V/O	0...1H	0...1	1	[-]
Diagnostic condition to start symptom detection (set to 1 when condition is fulfilled)					
ERR_SYM_SWI_AFS_MON	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
For each symptom : status of failure (set to 1 when failure symptom detected)					

Input data:

		LV_TQI_MON_ACT_MON	LV_SWI_AFS_MFF_MAX_MON

FUNCTION DESCRIPTION:

General information:

CTR_SWI_AFS_MON needs to be saved reset resistant AND in NVMY !

Description:

The MFF-monitoring-functionality is able to send requests for switching the engine to AFS-mode in case of detected unplausibilities in non-AFS-modes.

In order to gain information about the number of interventions of the MFF-monitoring-functionality during vehicle/ECU-lifetime, this diagnosis has been introduced.

The flag LV_SWI_AFS_MFF_MAX_MON represents that the maximum allowed number of requests by MFF-monitoring to switch to AFS-mode has been reached.

CTR_SWI_AFS_MON is used to count these events.

Additionally, the indicator flag LV_ERR_SWI_AFS_MON is set. It is used to call the ERRM and store the indicator with environmental data (e.g. with the value of CTR_SWI_AFS_MON) for further examination.

Error-symptoms and conditions: are defined to this diagnosis function as following

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> LV_SWI_AFS_MFF_MAX_MON detected        (= SYM_0)
 │ │ └───> not used                               (= SYM_1)
 │ └─────> not used                               (= SYM_2)
 └───────> not used                               (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_SWI_AFS_MON = STD_INI
LV_END_DIAG_SWI_AFS_MON = STD_INI
LV_ERR_SWI_AFS_MON = STD_INI
ERR_SYM_SWI_AFS_MON = STD_INI

at first power up/on saved RAM lost (or reprogramming): *CTR_SWI_AFS_MON = 0*

at reset: CTR_SWI_AFS_MON = restored from NVMY

Recurrence: 40ms

Activation:

```
If      LV_TQI_MON_ACT_MON = 1
Then    LV_CDN_DIAG_SWI_AFS_MON = 1
Endif
```

Deactivation:

When the activation condition is not fulfilled then LV_CDN_DIAG_SWI_AFS_MON = 0.

Formula section:

Symptoms calculation :

```
If      LV_SWI_AFS_MFF_MAX_MON changes from 0→1
Then
        ERR_SYM_SWI_AFS_MON = SYM_0
        CTR_SWI_AFS_MON = CTR_SWI_AFS_MON +1

Else
    ERR_SYM_SWI_AFS_MON = NO_SYM    { No failure has been detected }
Endif
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Configuration for diagnostic symptoms:

Diagnostic SWI_AFS_MON	Symptom description	Symptom	Filter type
AFS request by monitoring diagnosis	LV_SWI_AFS_MFF_MAX_MON	*SYM_0*	STD_INI
	not used	*SYM_1*	
	not used	*SYM_2*	
	not used	*SYM_3*	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SWI_AFS_MON	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_SWI_AFS_MON	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					

11.30 Error memory management of process monitoring (Appl.Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_32_MON_SAVE	O/V/S	0...FFH	0...8160	32	[rpm]
Engine speed (monitoring level, reset save)					
PREV_STATE_IV_SAVE	O/V/S	0...FFH	0...255	1	[-]
status word disable injectors (reset save)					
STATE_TQI_N_MAX_MON_1_1_SAVE	O/V/S	0...FFH	0...255	1	[-]
Statusbyte1 of LV_ERR_TQI_N_MAX_MON_1 (monitoring level, reset save)					
STATE_TQI_N_MAX_MON_1_2_SAVE	O/V/S	0...FFH	0...255	1	[-]
Statusbyte2 of LV_ERR_TQI_N_MAX_MON_1 (monitoring level, reset save)					
STATE_LV_ERR_MFF_MON_1	O/V	0...FFH	0...255	1	[-]
Statusbyte of LV_ERR_MFF_MON_1 (monitoring level)					
STATE_LV_ERR_TQ_DIF_ISC_MON_1	O/V	0...FFH	0...255	1	[-]
Statusbyte of LV_ERR_TQ_DIF_ISC_MON_1 (monitoring level)					
STATE_LV_ERR_CRU_INH_MON_1	O/V	0...FFH	0...255	1	[-]
Statusbyte of LV_ERR_CRU_INH_MON_1 (monitoring level)					

Input data:

LV_CRU_MAIN_SWI	PREV_STATE_IV	LV_CRU_ACT
LV_PAS_RAMP_ACT_P_D_CHG_MON	LV_TQ_DCC_INC_REQ	N_32_MON
PV_AV_MON	LV_PAS_RAMP_ACT_I_CHG_MON	LV_IGK_MON
LV_TQI_MON_ACT_MON	LV_ERR_TQI_AV_MON	LV_MTC_CUR_OFF
LV_IS	LV_ERR_PVS	LV_ERR_TQ_MIN_CLU_MON
LV_BRAKE_MON	LV_ERR_TQ_LOSS_MON	LV_ERR_MSR_INH_MON
LV_ERR_TPS	LV_ERR_CONV_MON	LV_ERR_TQ_DIF_I_IS_MON
LV_N_LIM_ETC_LIH	LV_ERR_TQ_DIF_P_D_IS_MON	LV_REQ_ISC
LV_ERR_TQ_MAX_CLU_MON	LV_ERR_OPM_AV_MON	Ba_ist
LV_ERR_N_32_MON	LV_ERR_PVS_MON	LV_LAMB_LS_UP_VLD_MON[NC_CBK_EX_NR]
LV_TQ_WHEEL_LDM_REQ	LV_ERR_MFF_MON	LV_ERR_TPS_MON

FUNCTION DESCRIPTION:

General information:

For every fault, the environmental conditions are stored in the environmental data variables. The error symptom is used for encoding the type of the fault if more than one level 2 error could nead to the errror. The variables are chosen such that the fault scenario can be analyzed.

The memory area of all output data variables of this module is not cyclical RAM tested,
the memory area of this module is not cyclic ROM tested.

Application conditions:

Initialization: -

Activation: *according to "Error Memory management of Process Monitoring"*

Update Rate: *according to "Error Memory management of Process Monitoring"*

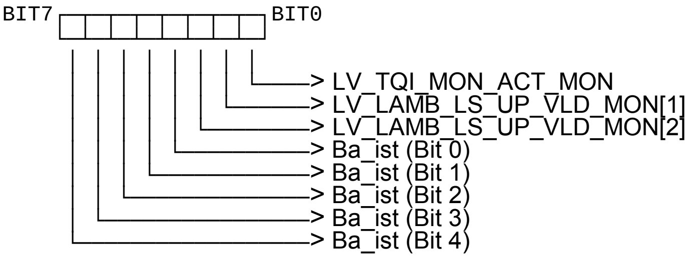
Formula section:

Since the different faults can occur in a successive way, they are stored in different locations of the failure memory.

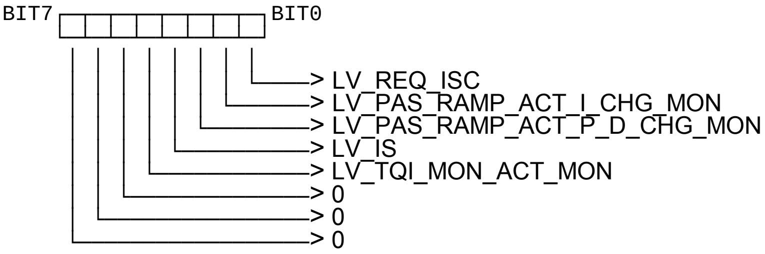
11.30.1 Definition of Statusbytes

The bit values for the ENVD of each error (that actually uses bit values within its ENVD) are represented by a statusbyte. The definition of these statusbytes is listed below.

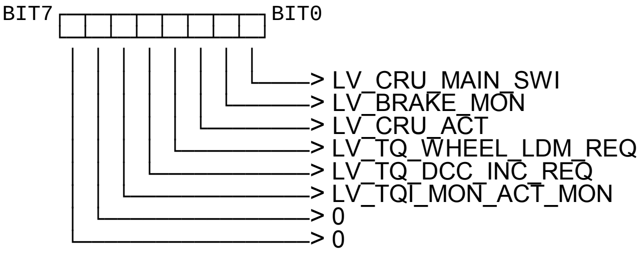
STATE_LV_ERR_MFF_MON_1 :



STATE_LV_ERR_TQ_DIF_ISC_MON_1 :



STATE_LV_ERR_CRU_INH_MON_1 :



11.30.2 Definition of non volantile ENVD for LV_ERR_TQI_N_MAX_MON_1

As the error LV_ERR_TQI_N_MAX_MON leads to a reset of the ECU, its environmental data need to be stored reset save before the reset. Therefore the ENVD of this error is copied to reset save labels. This is only done one time when the error occurs (refer to "ERRM of process monitoring": ERRM of LV_ERR_TQI_N_MAX_MON_1).

N_32_MON_SAVE = N_32_MON

PREV_STATE_IV_SAVE = PREV_STATE_IV

STATE_TQI_N_MAX_MON_1_1_SAVE :

```
BIT7 ┌┬┬┬┬┬┬┬┐ BIT0
     └┴┴┴┴┴┴┴┘
      │││││││└──────> LV_ERR_TQI_AV_MON
      ││││││└───────> LV_IGK_MON
      │││││└────────> LV_ERR_TPS
      ││││└─────────> LV_ERR_PVS
      │││└──────────> LV_MTC_CUR_OFF
      ││└───────────> (PV_AV_MON = 0)
      │└────────────> LV_N_LIM_ETC_LIH
      └─────────────> (LV_ERR_TQ_LOSS_MON=1 OR LV_ERR_TQ_MIN_CLU_MON=1 OR
                       LV_ERR_TQ_MAX_CLU_MON=1)
```

STATE_TQI_N_MAX_MON_1_2_SAVE :

```
BIT7 ┌┬┬┬┬┬┬┬┐ BIT0
     └┴┴┴┴┴┴┴┘
      │││││││└──────> LV_ERR_CONV_MON
      ││││││└───────> LV_ERR_MSR_INH_MON
      │││││└────────> LV_ERR_N_32_MON
      ││││└─────────> (LV_ERR_TQ_DIF_P_D_IS_MON=1 OR LV_ERR_TQ_DIF_I_IS_MON=1)
      │││└──────────> LV_ERR_PVS_MON
      ││└───────────> LV_ERR_TPS_MON
      │└────────────> LV_ERR_MFF_MON
      └────────────--> 0
```

To save some locations in failure memory some of the level 2 errors collect in one error ,location as different symtoms, this is shown in the next table. In addition to the information above, for every fault environmental data is stored. This datas are tuneable. The following table is only a recommendation of enviromental datas.

Error	SYM	Error Cause	Recommendation of environmental data			
Error bit level1		**Error bit level2**	**First byte**	**Second Byte**	**Third byte**	**Fourth byte**
LV_ERR_CONV_MON_1	3	LV_ERR_CONV_MON	N_32_MON	VP_MC_AN_DIG_MON_KWP	VP_MU_AN_DIG_MON_KWP	VB
LV_ERR_N_32_MON_1	3	LV_ERR_N_32_MON	N_32	N_32_MON	N_32_SUB_MON	T_SEG_SW_MON
LV_ERR_TPS_MON_1	3	LV_ERR_TPS_MON	TPS_AV	TPS_AV_MON	V_TPS_1_MON	V_TPS_2_MON
LV_ERR_PVS_MON_1	3	LV_ERR_PVS_MON	STATE_LV_ERR_CRU_INH_MON_1	PV_AV_MON	V_PVS_1_MON_KWP	V_PVS_2_MON_KWP
LV_ERR_CRU_INH_MON_1	0 or 1 or 2	LV_ERR_CRU_MON or LV_ERR_DCC_INH_MON or LV_ERR_LDM_INH_MON	STATE_LV_ERR_CRU_INH_MON_1	TQ_WHEEL_LDM_MON	FAC_TQ_REQ_CRU	TQ_DCC_MON
LV_ERR_TQI_AV_MON_1	3	LV_ERR_TQI_AV_MON	N_32_MON	TQI_SP_MON	TQI_AV_MON	TQI_SP_KWP
LV_ERR_TQI_N_MAX_MON_1	3	LV_ERR_TQI_N_MAX_MON	N_32_MON_SAVE	PREV_STATE_IV_SAVE	STATE_TQI_N_MAX_MON_1_1_SAVE	STATE_TQI_N_MAX_MON_1_2_SAVE
LV_ERR_MFF_MON_1	2 or 3	LV_ERR_OPM_AV_MON or LV_ERR_MFF_MON	STATE_LV_ERR_MFF_MON_1	MAF	MFF_SP_MV_KWP	LAMB_LS_UP_MV_KWP
LV_ERR_TQ_DIF_ISC_MON_1	2 or 3	LV_ERR_TQ_DIF_I_IS_MON or LV_ERR_TQ_DIF_P_D_IS_MON	STATE_LV_ERR_TQ_DIF_ISC_MON_1	N_DIF_SP_IS_MON	TQ_DIF_I_IS_MON	TQ_DIF_P_D_IS_MON
LV_ERR_TQ_REQ_MON_1	0 or 1 or 2 or 3	LV_ERR_TQ_MAX_CLU_MON or LV_ERR_TQ_MIN_CLU_MON or LV_ERR_TQ_LOSS_MON or LV_ERR_SOF_INH_MON	TQ_MAX_CLU_DIF_MON	TQ_MIN_CLU_DIF_MON	TQ_LOSS_DIF_MON	STATE_ENG
LV_ERR_TQ_EXT_MON_1	0 or 2 or 3	LV_ERR_MSR_INH_MON or LV_ERR_AMT_INH_MON or LV_ERR_GS_INH_MON	BIT 0: LV_MSR_PLAUS BIT 2: LV_TQ_AMT_INC_REQ BIT 3: LV_TQ_GS_INC_REQ	TQI_MSR_CAN	TQ_GS_MON	STATE_ERR_AMT_MON

The information whether a warning lamp (ETC MIL) is switched on in case of an error can be found in the general OBD error code table.

In case of more than one error cause occurs, always the first error (according to the table above; see BIT 0...4 of first Byte for the order) is taken into account.

11.30.3 List of error codes and symptoms

The list of error codes and the list of error symptoms can be found in the chapter "Table of failures".

Configuration diagnostic symptoms:

Diagnosis XX	Symptom	Nr	ABC type
Diagnosis description: A/D monitoring error	*no error*	*NO_SYM*	NO
CONV_MON	*A/D monitoring error*	*SYM_3*	
Diagnosis description: Engine speed monitoring error	*no error*	*NO_SYM*	NO
N_32_MON	*Engine speed monitoring error*	*SYM_3*	
Diagnosis description: TPS monitoring error	*no error*	*NO_SYM*	NO
TPS_MON	*TPS monitoring error*	*SYM_3*	
Diagnosis description: PVS monitoring error	*no error*	*NO_SYM*	NO
PVS_MON	*PVS monitoring error*	*SYM_3*	
Diagnosis description: Cruise control monitoring error	*no error*	*NO_SYM*	NO
CRU_INH_MON	*Cruise control monitoring error*	*SYM_0*	
	Distance cruise control monitoring error	*SYM_1*	
	LDM monitoring error	*SYM_2*	
Diagnosis description: Torque monitoring error	*no error*	*NO_SYM*	NO
TQI_AV_MON	*Torque monitoring error*	*SYM_3*	
Diagnosis description: N_LIM monitoring error	*no error*	*NO_SYM*	NO
TQI_N_MAX_MON	*N_LIM monitoring error*	*SYM_3*	
Diagnosis description: Mass fuel flow monitoring error	*no error*	*NO_SYM*	NO
MFF_MON	*Error in engine operation mode*	*SYM_2*	
	Mass fuel flow monitoring error	*SYM_3*	
Diagnosis description: ISC monitoring error	*no error*	*NO_SYM*	NO
ISC_MON	*ISC I part monitoring error*	*SYM_2*	

	ISC P-D part monitoring error	*SYM_3*	
Diagnosis description: *Torque req. monitoring error*	*no error*	*NO_SYM*	NO
TQ_REQ_MON	*TQ_MAX_CLU monitoring error*	*SYM_0*	
	TQ_MIN_CLU monitoring error	*SYM_1*	
	Torque loss monitoring error	*SYM_2*	
	Sound flap monitoring error	*SYM_3*	
Diagnosis description: *External torque req. control monitoring error*	*no error*	*NO_SYM*	NO
TQ_EXT_MON	*Deceleration slip control monitoring error*	*SYM_0*	
	Automated manual transmission monitoring error	*SYM_2*	
	Gear shift monitoring error	*SYM_3*	

11.31 Cus adap module: ECM2

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_dkpu	O/V	0...1H	0...1	1	[-]
LV for activation of engine speed limitation monitoring					
FAC_MFF_TQ_COR_SCAV	O/V	8000...7FFFH	-200...199.99389	6.1035e-3	[%]
Efficiency correction factor for scavenging					
TQ_LOSS_ADD_FIL	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Filtered additional torque losses					
TQ_REQ_CLU_GB	O/V	0...FFH	0...510	2	[Nm]
Torque request from manual gearbox (torque losses)					
TQI_N_CTL_TCT_FAST	O/V	0...FFH	0...510	2	[Nm]
Fast torque request of n-controller for TCT					
TQI_N_CTL_TCT_SLOW	O/V	0...FFH	0...510	2	[Nm]
Slow torque request of n-controller for TCT					

Input data:

LV_N_LIM_ETC_MON	Eta_md_uesp	Md_getriebe_hs	Md_na_ges_f
Mdi_nregs_plus	Mdi_nregl_plus		

11.31.1 Outputs for BMW functions which are not defined as ECM2 exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: 0

Recurrence : 40 ms

Activation: every engine state

Formula section:

B_dkpu = LV_N_LIM_ETC_MON

11.31.2 Outputs for SV functions which are defined as BMW exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation:
FAC_MFF_TQ_COR_SCAV =100%
TQ_REQ_CLU_GB = 0 Nm
TQI_N_CTL_TCT_FAST = 0 Nm
TQI_N_CTL_TCT_SLOW = 0 Nm
TQ_LOSS_ADD_FIL = 0 Nm

Recurrence : FAC_MFF_TQ_COR_SCAV:10ms
TQ_REQ_CLU_GB: 40ms
TQI_N_CTL_TCT_FAST: 40ms
TQI_N_CTL_TCT_SLOW: 40ms
TQ_LOSS_ADD_FIL: 40ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

FAC_MFF_TQ_COR_SCAV = Eta_md_uesp
TQ_REQ_CLU_GB = Md_getriebe_hs
TQI_N_CTL_TCT_FAST = Mdi_nregs_plus
TQI_N_CTL_TCT_SLOW = Mdi_nregl_plus
TQ_LOSS_ADD_FIL = Md_na_ges_f

12 ECU Processor Monitoring

Table of Contents

Index

12.1 ECM3 - MC requirements to infrastructure

Export actions:

ACTION_INFR_McChkCks(OUT <lv_mc_cks_ok>)
Check whether the checksum of the current communication was correctly transmitted.
ACTION_INFR_McChkPar(OUT <lv_mc_par_ok>)
Check whether the parity bits in the current communication are correctly set.
ACTION_INFR_McGetHd(OUT <state_mu>)
Get the header byte from the current communication.
ACTION_INFR_McGetErrByte(OUT <lf_mu_err_byte>)
Get error byte (containing error code and indication) from the current communication.
ACTION_INFR_McGetCompNr(OUT <nr_mu_comp>)
Get compatibility number transferred from the MU in the current communication.
ACTION_INFR_McGetVersNr(OUT <nr_mu_hw_vers>)
Get HW version number transferred from the MU in the current communication.
ACTION_INFR_McGetRstCtr(OUT <ctr_rst_mu>)
Get the reset counter of the MU transferred in the current communication.
ACTION_INFR_McGetConfData(OUT <lf_mu_conf_byte>)
Get the configuration data transferred from the MU in the current communication.
ACTION_INFR_McGetOvdFlag(OUT <lv_v_h_det_mu>)
This action returns the over-voltage detection flag sent by the MU.
ACTION_INFR_McGetUvdFlag(OUT <lv_v_l_det_mu>)
This action returns the under-voltage detection flag sent by the MU.
ACTION_INFR_McGetPfmByte(OUT <lf_mc_rcv_pfm_byte>)
Get the PFM data sent by the MU.
ACTION_INFR_McGetFsistIdx(OUT <idx_test_rcv_mc_fct_spc_ist>)
Get the FS-IST question sent by the MU.
ACTION_INFR_McGetFsistAbcVal(OUT <abc_rcv_mc_fct_spc_ist>)
Get the FS-IST anti-bounce counter value sent by the MU.
ACTION_INFR_McGetIgkMu(OUT <lv_mu_ign_key>)
Get the ignition key signal transmitted by the MU in the current communication.
ACTION_INFR_GetAnDigMonValMu(OUT <vp_mu_an_dig_mon>)
Get value of A/D conversion sent from the MU in the current communication.
ACTION_INFR_GetAnDigMonValMc(OUT <vp_mc_an_dig_mon>)
Get value of A/D conversion performed on the MC.
ACTION_INFR_McRestart()
Trigger a reset of the MC.
ACTION_INFR_McGetNdis0(IN <lv_mu_di_out_0>)
Read value of the NDIS_OUT0 pin sent by the MU.
ACTION_INFR_McGetNdis1(OUT <lv_mu_di_out_0>)
Read value of the NDIS_OUT1 pin sent by the MU.
ACTION_INFR_McStartCom(IN <state_mc>)
Trigger communication according to the given state.
ACTION_INFR_McRomChkReady(OUT <lv_mc_rom_chk_ready>)
This action returns a flag indicating whether the level 1 ROM check is ready or not.
ACTION_INFR_McRomChkOk(OUT <lv_mc_rom_chk_ok>)
This action returns a flag indicating whether the level 1 ROM check was completed successfully or not.
ACTION_INFR_McGetDiagPws0(OUT <state_mc_sopc_dr_0>)
This action returns status information of the power stage 0 (whether it is enabled (=1) or disabled (=0), or undefined (=3)).
ACTION_INFR_McGetDiagPws1(OUT <state_mc_sopc_dr_1>)
This action returns status information of the power stage 1 (whether it is enabled (=1) or disabled (=0), or undefined (=3)).
ACTION_INFR_McSendConfData(IN <lv_mu_mod>, IN <lv_mu_ign_key_acq>, IN <lv_mu_pow_off_trm_ena>, IN < lv_mu_inj_off_trm_ena >)
This action sends the configuration data to the infrastructure for transmission to the MU.
ACTION_INFR_McSendCompNr(IN <nr_mc_comp>)

This action sends the compatibility number of the MC to the infrastructure for transmission to the MU.
ACTION_INFR_McSendIgk(IN <lv_igk>)
This action sends the ignition key signal to the infrastructure for transmission to the MU.
ACTION_INFR_McSendErrInfo(IN <ctr_rst_mc>, IN <err_cod_mc>, IN <state_err_idc_mc>)
This action sends the error information (reset counter, error code & indication) to the infrastructure for transmission to the MU.
ACTION_INFR_McSendDrOff(IN <lv_mc_dr_off>)
This action sends the flag for the redundant switch-off path to the infrastructure for transmission to the MU.
ACTION_INFR_McSendPfmResp(IN <lf_mc_resp_pfm>)
This action sends the PFM response of the MC to the infrastructure for transmission to the MU.
ACTION_INFR_McSendFsistResp(IN <idx_resp_trm_mc_fct_spc_ist>)
This action sends the FS-IST answer of the MC to the infrastructure for transmission to the MU.
ACTION_INFR_McChkStack(OUT <PRM_LV_CHK_OK>)
This action returns the result of the stack check performed by the infrastructure.
ACTION_INFR_StartMuReadyChk()
This action triggers the start of the check for readiness of MU.
ACTION_INFR_MuReady(OUT <PRM_LV_MU_READY>)
This action fetches the result of the check for readiness of MU.

Input data:

NC_STATE_RESP_DR_DI	NC_STATE_RESP_DR_ENA	NC_STATE_RESP_DR_DFT	NC_VP_MC_AN_DIG_MON_CPL_ERR
NC_T_MC_COM_CYC	NC_STATE_MC_NOT_VLD	NC_STATE_MC_CONF	NC_STATE_MC_PRDR
NC_STATE_MC_NORM	NC_STATE_MC_DI		

Description for actions:

<table>
<tr><td colspan="6">ACTION_INFR_McChkCks(lv_mc_cks_ok)</td></tr>
<tr><td colspan="6">This action returns a flag indicating whether the XOR checksum of the received data was correctly transmitted; this is done comparing a freshly calculated checksum to the sent one; potential fault reactions are performed elsewhere (in the ASW).</td></tr>
<tr><th>Parameter</th><th>Type</th><th>Hex. Limits</th><th>Phys. Limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>lv_mc_cks_ok</td><td>OUT</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Flag indicating whether the checksum of the received data is correct</td></tr>
</table>

<table>
<tr><td colspan="6">ACTION_INFR_McChkPar(lv_mc_par_ok)</td></tr>
<tr><td colspan="6">This action returns a flag indicating whether the parity bits were proprly set in the current communication; potential fault reactions are performed elsewhere (in the ASW).</td></tr>
<tr><th>Parameter</th><th>Type</th><th>Hex. Limits</th><th>Phys. Limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>lv_mc_par_ok</td><td>OUT</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Flag indicating whether the parity of the received data is correct</td></tr>
</table>

<table>
<tr><td colspan="6">ACTION_INFR_McGetHd(state_mu)</td></tr>
<tr><td colspan="6">This action reads the header byte received from the MU in the current communication.</td></tr>
<tr><th>Parameter</th><th>Type</th><th>Hex. Limits</th><th>Phys. Limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>state_mu</td><td>OUT</td><td>00...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Header byte sent by MC in the last communication</td></tr>
</table>

<table>
<tr><td colspan="6">ACTION_INFR_McGetErrByte(lf_mu_err_byte)</td></tr>
<tr><td colspan="6">This action reads the error byte received from the MU in the current communication; this error byte encodes the error code of the MU and the error indication of the MU.</td></tr>
<tr><th>Parameter</th><th>Type</th><th>Hex. Limits</th><th>Phys. Limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>lf_mu_err_byte</td><td>OUT</td><td>00...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Byte sent by MC containing error code and error indication</td></tr>
</table>

ACTION_INFR_McGetCompNr(nr_mu_comp)

This action reads a byte containing compatibility information of the MU.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
nr_mu_comp	OUT	0...3H	0...3	1	[-]

Compatibility number sent by the MU

ACTION_INFR_McGetVersNr(nr_mu_hw_vers)

This action reads a byte containing compatibility information of the MU.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
nr_mu_hw_vers	OUT	00...1FH	0...31	1	[-]

HW version number sent by the MU

ACTION_INFR_McGetRstCtr(ctr_rst_mu)

This action reads the reset counter of the MU.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
ctr_rst_mu	OUT	0...FH	0...15	1	[-]

Reset counter of MU sent in the current communication

ACTION_INFR_McGetConfData(lf_mu_conf_byte)

This action reads a byte encoding configuration data received from the MU in the current communication.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lf_mu_conf_byte	OUT	00...FFH	0...255	1	[-]

Byte sent by the MU containing configuration information

ACTION_INFR_McGetOvdFlag(lv_v_h_det_mu)

This action reads the flag sent by the MU indicating whether over-voltage had been detected

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_v_h_det_mu	OUT	0...1H	0...1	1	[-]

Flag indicating whether over-voltage has been detected by the MU

ACTION_INFR_McGetUvdFlag(lv_v_l_det_mu)

This action reads the flag sent by the MU indicating whether under-voltage had been detected

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_v_l_det_mu	OUT	0...1H	0...1	1	[-]

Flag indicating whether under-voltage has been detected by the MU

ACTION_INFR_McGetPfmByte(lf_mc_rcv_pfm_byte)

This actions reads the program flow monitoring byte sent by the MU in the current communication.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lf_mc_rcv_pfm_byte	OUT	00...FFH	0...255	1	[-]

PFM byte sent by MU in the current communication: either an updated synchronisation byte or status information of MU PRM anti-bounce counters

ACTION_INFR_McGetFsistIdx(idx_test_rcv_mc_fct_spc_ist)

This action reads the MU question for the FS-IST.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
idx_test_rcv_mc_fct_spc_ist	OUT	00...3FH	0...63	1	[-]

FS-IST question sent by MU in the current communication

ACTION_INFR_McGetFsistAbcVal(abc_rcv_mc_fct_spc_ist)

This action reads the MU question for the FS-IST.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
abc_rcv_mc_fct_spc_ist	OUT	0...FH	0...15	1	[-]

FS-IST anti-bounce counter sent by MU in the current communication

ACTION_INFR_McGetIgkMu(lv_mu_ign_key)

This value reads the ignition key signal transmitted by the MU in the current communication.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_mu_ign_key	OUT	0...1H	0...1	1	[-]

Value of the ignition key (as obtained via the IGN_IN pin on the MU) sent from the MU in the current communication

ACTION_INFR_GetAnDigMonValMc(vp_mc_an_dig_mon_tmp)

This action reads the new value from the A/D conversion on the MC from the infrastructure.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
vp_mu_an_dig_mon_tmp	OUT	8000...7FFFH	-5...4.99984	0.1526e-3	[V]

Value of A/D conversion on the MC

ACTION_INFR_GetAnDigMonValMu(vp_mu_an_dig_mon)

This action reads the new value from the A/D conversion on the MU from the infrastructure.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
vp_mc_an_dig_mon	OUT	0...7FFFH	0...4.99984	0.1526e-3	[V]

Value of A/D conversion on the MU

ACTION_INFR_McRestart()

This action triggers a reset of the MC.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit

ACTION_INFR_McGetNdis0(lv_mu_di_out_0)

This action reads the value of the NDIS_OUT0 pin sent from the MU

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_mu_di_out_0	OUT	0...1H	0...1	1	[-]

Actual value of the NDIS_OUT0 pin

ACTION_INFR_McGetNdis1(lv_mu_di_out_1)

This action reads the value of the NDIS_OUT1 pin sent from the MU

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_mu_di_out_1	OUT	0...1H	0...1	1	[-]

Actual value of the NDIS_OUT1 pin

ACTION_INFR_McRomChkReady(lv_mc_rom_chk_ready)

This action returns a flag indicating whether the level 1 ROM check has been completed.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_mc_rom_chk_ready	OUT	0...1H	0...1	1	[-]

Flag indicating whether the level 1 ROM check has been completed

ACTION_INFR_McRomChkOk(lv_mc_rom_chk_ok)

This action returns a flag indicating whether the level 1 ROM check has been completed successfully or not.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_mc_rom_chk_ok	OUT	0...1H	0...1	1	[-]

Flag indicating whether the level 1 ROM check has been completed successfully

ACTION_INFR_McGetDiagPws0(state_mc_sopc_dr_0)

This action returns status information about power stage 0 required for the switch-off path check.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
state_mc_sopc_dr_0	OUT	0...3H	0...3	1	[-]

State indicating whether power stage 0 is enabled or disabled, or whether the diagnostic is not finished/unstable

ACTION_INFR_McGetDiagPws1(state_mc_sopc_dr_1)

This action returns status information about power stage 1 required for the switch-off path check.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
state_mc_sopc_dr_1	OUT	0...3H	0...3	1	[-]
State indicating whether power stage 1 is enabled or disabled, or whether the diagnostic is not finished/unstable					

ACTION_INFR_McSendConfData(lv_mu_mod, lv_mu_ign_key_acq, lv_mu_pow_off_tmr_ena, lv_mu_inj_off_tmr_ena)

This action sends the configuration data to the infrastructure; from there, they are transmitted to the MU.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_mu_mod	IN	0...1H	0...1	1	[-]
Mode of MU (0 = EMS, 1 = TCU)					
lv_mu_ign_key_acq	IN	0...1H	0...1	1	[-]
Mode for ignition key acquisition on MU (0 = SW, 1 = HW)					
lv_mu_pow_off_tmr_ena	IN	0...1H	0...1	1	[-]
Enable flag for power-off timer on MU					
lv_mu_inj_off_tmr_ena	IN	0...1H	0...1	1	[-]
Enable flag for injection-off timer on MU					

ACTION_INFR_McSendCompNr(nr_mc_comp)

This action sends the compatibility number of the MC to the infrastructure; from there, it is transmitted to the MU.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
nr_mc_comp	IN	0...3H	0...3	1	[-]
Compatibility number of MC					

ACTION_INFR_McSendIgk(lv_igk)

This action sends the ignition key signal to the infrastructure; from there, it is transmitted to the MU

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_igk	IN	0...1H	0...1	1	[-]
Value of ignition key signal on MC					

ACTION_INFR_McSendErrInfo(ctr_rst_mc, err_cod_mc, state_err_idc_mc)

This action sends the error informarion on the MC to the infrastructure; from there, it is transmitted to the MU.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
ctr_rst_mc	IN	0...FH	0...15	1	[-]
Reset counter on MC					
err_cod_mc	IN	00...1FH	0...31	1	[-]
Error code on MC					
state_err_idc_mc	IN	0...3H	0...3	1	[-]
Error indication on MC					

ACTION_INFR_McSendDrOff(lv_mc_dr_off)

This action sends the flag for the redundant switch-off path to the infrastructure; from there, it is transmitted to the MU.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_mc_dr_off	IN	0...1H	0...1	1	[-]
Flag indicating whether the redundant switch-off path was triggered or not					

ACTION_INFR_McSendPfmResp(lf_mc_resp_pfm)

This action sends the current PFM response on the MC to the infrastructure; from there, it is transmitted to the MU.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lf_mc_resp_pfm	IN	00...7FH	0...127	1	[-]
PFM response on the MC					

ACTION_INFR_McSendFsistResp(idx_resp_trm_mc_fct_spc_ist)					
This action sends the FS-IST answer on the MC to the infrastructure; from there, it is transmitted to the MU.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
idx_resp_trm_mc_fct_spc_ist	IN	00...3F	0...63	1	[-]
FS-IST answer on the MC					

ACTION_INFR_McStartCom(state_mc)					
This actions triggers a communication corresponding to the state given as argument.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
state_mc	IN	00...FFH	0...255	1	[-]
Header byte sent from MC					

ACTION_INFR_McChkStack(PRM_LV_CHK_OK)					
This actions triggers a communication corresponding to the state given as argument.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_LV_CHK_OK	OUT	0...1H	0...1	1	[-]
Result of stack check performed by infrastructure (1=OK, 0=NOK)					

ACTION_INFR_StartMuReadyChk()					
This action starts the check for MU readiness.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**

ACTION_INFR_MuReady(PRM_LV_MU_READY)					
This action fetches the result of the check for MU readiness.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_LV_MU_READY	OUT	0...1H	0...1	1	[-]
Result of MU readiness check performed by infrastructure (1=OK, 0=NOK)					

FUNCTION DESCRIPTION:

General information:

This document specifies the requirements of the processor monitoring ASW implemented on the main controller (MC) to the infrastructure: the requirements deal with the communication interface between the MC and MU (from the point of view of the MC) and the interface to the hardware.

Requirements for ACTION_INFR_McChkCks:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

This action shall incorporate requirements R44_3 and R44_4 of the following function.

12.1.1 G44 Communication checksum monitoring

R44_1: All transmitted data shall be protected by an XOR checksum of header byte and data bytes.

R44_2: The checksum shall always be the last data to be transmitted.

R44_3: The XOR checksum of the received data shall be checked.

R44_4: In case of a checksum error, this shall be indicated to the application SW by setting the return value lv_mc_cks_ok to 0.

Coincidence requirements: -

Requirements for ACTION_INFR_McChkPar:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

This action shall incorporate the following requirements.

12.1.2 G45 Parity-bit checking

R45_1:The most sensitive data shall be protected by a parity bit (these shall be the second and third byte in CONFIG and DISABLE communications).

R45_2: Sensitive data shall be checked for even parity.

R45_3: In case of a parity bit error, this shall be indicated to the application SW by setting the return value lv_mc_par_ok to 0.

Coincidence requirements: -

Requirements for ACTION_INFR_McRestart:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

The action shall trigger a system reset, including a reset of the MU.

Coincidence requirements: -

Requirements for ACTION_INFR_McGetDiagPws0:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

During the switch-off path check (SOPC), the current status (enabled/disabled) of the power stage is inspected by Processor Monitoring on the MC to check if the MU or the MC can disable the power stage. This leads to several requirements described in the following function.

12.1.3 G48 Requirements to IRS actions for switch-off path check

R48_1: The SOPC requires interfaces indicating the status of monitoring-relevant power stage(s) at certain times depending on the interaction of several Processor Monitoring functions.

R48_2: The status information shall be provided in a way independent from the actually used HW drivers for these monitoring-relevant power stage(s).

R48_3: The status information shall be acquired by calling the actions ACTION_ECM3_McGetDiagPws0() and ACTION_ECM3_McGetDiagPws1() inside the Processor Monitoring function "Switch-Off Path Check."

R48_4: Each of the two action calls shall return a three-valued status of the corresponding power stage: a return value of NC_STATE_RESP_DR_ENA (=1) shall mean " enabled", NC_STATE_RESP_DR_DI (=0) shall mean "disabled", NC_STATE_RESP_DR_DFT (=3) shall mean "undefined."

R48_5: If the status information for a power stage is unavailable, unclear, or unstable when requested, the returned status shall indicate "undefined"; this shall also be the default value after an acquisition of the status via one of the action calls (internal reset of the status to "undefined").

R48_6: Regardless of the way of its acquisition, the information status of a power stage indicated by the flag returned by any of the actions shall not be older than 20ms for GS/DS and 8ms for TCU when the interface is called (corresponding to the length of two communication cycles of Processor Monitoring for EMS or TCU mode); this may well have consequences on the frequency of SPI line checks and other mechanisms.

R48_7: It shall be possible to perform the switch-off path check at any time, regardless of the used power stage(s) and the current state of the system; in particular, this means that calling the function for retrieving the value of the power stage should be possible any time without disrupting the system's behaviour.

Coincidence requirements: -

Requirements for ACTION_INFR_McGetDiagPws1:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

During the switch-off path check, the current status (enabled/disabled) of the power stage is inspected by Processor Monitoring on the MC to check if the MU or the MC can disable the power stage. This leads to several requirements described in the following function.

G48 Requirements to IRS actions for switch-off path check

(see above)

Coincidence requirements: -

Requirements for ACTION_INFR_GetAnDigMonValMc:

Data acquisition					
Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment

Diagnosis:

The action shall incorporate the following requirements.

12.1.4 G53 Requirements to infrastructure interface for acquisition of values of A/D conversions on MC

R53_1: Directly after the end of each communication between MC and MU, an A/D conversion (of the same signal which is connected to the A/D converter of the MU) shall be performed.

R53_2: The result of the conversion shall be stored as value and complement until it is fetched via ACTION_INFR_GetAnDigMonValMc().

R53_3: When the action ACTION_INFR_GetAnDigMonValMc() is called, the stored value shall be checked for consistency with the stored complement:
R53_3.1: if value and complement are consistent, the actual converted value shall be returned;
R53_3.2: if value and complement are inconsistent, the constant NC_VP_MC_AN_DIG_MON_CPL_ERR shall be returned instead of the actual converted value.

Coincidence requirements: -

Requirements for ACTION_INFR_StartCom:

Data acquisition					
Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment

Diagnosis:

The action shall incorporate the following requirements.

12.1.5 G61 Requirements to infrastructure interface for triggering a communication with the MU

R61_1: This action shall incorporate the requirements R44_1 and R44_2 of the function "G44 Communication checksum monitoring" described above.

R61_2: Based on the state information passed as argument, parity bits shall be set for the appropriate data bytes (see description of the communication protocol) to achieve even parity (cf. "G45 Parity-bit checking" above).

R61_3: When the action is called, it shall invalidate the previous header-byte sent by the MU by copying NC_STATE_MC_NOT_VLD in its place.

R61_4: The state information passed as argument shall be included in the outgoing message to the MU.

R61_5: The action shall ensure that a communication with the MU is eventually performed (whether that means that a communication is triggered directly or just scheduled).

Coincidence requirements: -

Requirements for ACTION_INFR_SendConfData:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

The action shall incorporate the following functional requirements.

12.1.6 G46 Complement of configuration data

R46_1: In the CONFIG and PREDRIVE communication, the configuration data sent by the MC shall be protected by also sending the bit-wise complement of the configuration information (see specification of the communication protocol).

Coincidence requirements: -

Requirements for ACTION_INFR_McChkStack:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

The action shall incorporate the following functional requirements.

12.1.7 G50 Requirements to infrastructure interface for stack check

R50_1: The stack check shall have the ability to detect static stuck-at errors (like the standard RAM check).

R50_2: While the stack check is performed, interrupts shall be suppressed; consequently, there is an upper bound on the time available for the stack check (<= 2µs).

R50_3: The stack check shall test that slice of memory which would be used for saving the content of the registers used inside level 2 when an interrupt occurs during its execution.

R50_4: For the call and the execution of the stack check itself, no stack shall be used.

Coincidence requirements: -

Requirements for ACTION_INFR_StartMuReadyChk:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

The action shall incorporate the following functional requirements.

12.1.8 G58 Requirements to infrastructure interface for start of MU readiness check

R58_1: The check for readiness of MU shall be performed by sending a communication with more than 48bit length to the MU (the MU simply discards such communications).

R58_2: When the action is called, such a special communication shall be prepared and scheduled.

R58_3: When the action is called, it shall invalidate the previous header-byte sent by the MU by copying NC_STATE_MC_NOT_VLD in its place.

R58_4: The communication shall be completed at most NC_T_MC_COM_CYC ms later so that the result can be evaluated.

Coincidence requirements: -

Requirements for ACTION_INFR_MuReady:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

The action shall incorporate the following functional requirements.

12.1.9 G59 Requirements to infrastructure interface for evaluation of MU readiness check

R59_1: When the action is called, the MU response to the special communication used for the readiness check (cf. G58) shall be evaluated.

R59_2: The return value of the action shall be 1 if the MU response is different from only constant '0' and only constant '1' (i.e., at least two bits in the MU response are not equal to each other).

R59_3: Otherwise (i.e., all bits in the MU response are equal to '0' or all bits are equal to '1') it is assumed that the MU is not yet ready and the return value of the action shall be 0.

Note: also HW problems (shortcuts to ground or supply) of the SPI connection might lead to the MU being seen as not ready.

Coincidence requirements: -

12.1.10 Requirements to infrastructure interface for Standard ROM check on MC

FUNCTION DESCRIPTION:

General information:

In order to detect static ROM faults, the Standard ROM check is performed once after each power-up.

Description:

The function shall incorporate the following requirements:

G55 Requirements to infrastructure interface for Standard ROM check on MC

R55_1: For computing the checksum, the CRC algorithm shall be used.

R55_2: The areas checked by the Standard ROM check include all code, constant, and calibration areas with the exception of such ROM areas which might be changed during runtime (e.g., EEPROM simulation in the FLASH memory); furthermore excluded is the area where the expected checksum is stored.

R55_3: There shall be a configuration switch in the infrastructure routine for the ROM check with the effect that a ROM error in the calibration data area does not result in reporting an unsuccessful ROM check: this is done in order to prevent ROM check errors (and a subsequent fault reaction) while calibrations are performed.

R55_4: The Standard ROM check shall be computed at least once per driving cycle; this means if it has been completed once, it need not be restarted again for the driving cycle (in particular not after a reset); however, if it is interrupted by a reset while not yet completed, it shall be restarted after the reset.

12.1.11 Requirements to infrastructure interface for Standard RAM check on MC

FUNCTION DESCRIPTION:

General information:

The standard RAM check is performed after each power-up or reset.

By writing specific bit patterns to the RAM cells, static faults can be detected.

Description:

The action shall incorporate the following functional requirements.

G56 Requirements to infrastructure interface for Standard RAM check on MC

R56_1: During each start-up (after reset or power-up), all RAM areas shall be tested.

R56_2: The Standard RAM check shall at least provide the option to be destructive or non-destructive, as is appropriate for the tested areas; in case of destructive testing, the RAM cells shall afterwards be initialised to 0.

R56_3: The RAM check shall be able to detect static "stuck to 0" and "stuck to 1" errors in each of the tested RAM cells.

12.2 ECU-selftest

FUNCTION DESCRIPTION:

General information:

The purpose of the test is the check of differents memories (RAM, Flash, NVMY).

Remark:
- If failures are detected there are not-defined system reactions possible!
- If there is a failure in the Boot-software it it possible to have a passive ECU without error entry.

12.2.1 Flash test

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ECU_CKS	V/O	0...1H	0...1	1	[-]
Present ECU checksum error					
LV_CDN_DIAG_ECU_CKS	V/O	0...1H	0...1	1	[-]
ECU Diagnosis active (=1)					
ERR_SYM_ECU_CKS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected Symptom ECU					
LV_END_DIAG_ECU_CKS	V/O	0...1H	0...1	1	[-]
End of ECU Diagnosis					

Description:

The following three areas are checked by calculating the checksum:

- Boot software
- Applicative software
- Calibration data

For error healing one diagnosis cyle is necessary. If there is only transition key-ON / key OFF without end of diagnosis is reached then the error remains active.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Error in Boot - SW                (= SYM_0)
 │ │ └───> Error in Applic - SW              (= SYM_1)
 │ └─────> Error in Calibration data range   (= SYM_2)
 └───────> -                                 (= SYM_3)
```

Application conditions:

Initialisation: all 0 at reset **or** clearing error memory

Recurrence: Boot software test is applied once, recurrency of applicative software and calibration data test is depending on length of each test (some minutes possible).

Activation: at system start up after ignition-key-ON

(applicative software and calibration-data test remains active for the engine run)

Formula section:

Setting of condition diag.

LV_CDN_DIAG_ECU_CKS = 1 at reset

Error detection

```
If      one error symptom is detected
Then    ERR_SYM_ECU_CKS = SYM_0 or SYM_1 or SYM_2
        LV_ERR_ECU_CKS = 1                    //no debounce
        LV_END_DIAG_ECU_CKS = 1
Else    ERR_SYM_ECU_CKS = NO_SYM
        LV_ERR_ECU_CKS = 0
Endif
```

End of diagnosis with no error detection

after finishing each test once: LV_END_DIAG_ECU_CKS = 1

12.2.2 RAM test

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ECU_RAM	V/O	0...1H	0...1	1	[-]
Present ECU RAM error					
LV_CDN_DIAG_ECU_RAM	V/O	0...1H	0...1	1	[-]
ECU Diagnosis RAM active (=1)					
ERR_SYM_ECU_RAM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected Symptom ECU RAM					
LV_END_DIAG_ECU_RAM	V/O	0...1H	0...1	1	[-]
End of ECU Diagnosis RAM					

Description:

The following two areas are tested by ECU internal test routine:

- Internal - RAM
- Timing processing unit – RAM

- External -RAM

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> RAM error                    (= SYM_0)
 │  │  └────> -                            (= SYM_1)
 │  └───────> -                            (= SYM_2)
 └──────────> -                            (= SYM_3)
```

Remark: LV_ERR_ECU_RAM will only deliver a valid result if Level 3 (processor monitoring) is not implemented yet. Otherwise Level 3 will detect a failed RAM-test thru LV_ERR_MON_3 and the corresponding symptom: ERR_SYM_MON_3 = SYM_1 (RAM error on MC).

Application conditions:

Initialisation: all 0 at reset **or** clearing error memory

Recurrence: once

Activation: at reset

Formula section:

Setting of condition diag.

LV_CDN_DIAG_ECU_RAM = 1 at reset

Error detection

```
If      one error symptom is detected          (RAM test error detected by L1)
Then    ERR_SYM_ECU_RAM = SYM_0
        LV_ERR_ECU_RAM = 1                     (error is set for this engine run, no debounce)
        LV_END_DIAG_ECU_RAM = 1
Else    after finishing each test once LV_END_DIAG_ECU_RAM = 1
Endif
```

12.2.3 NVMY test

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ECU_NVMY	V/O	0...1H	0...1	1	[-]
Present ECU NVMY error					
LV_CDN_DIAG_ECU_NVMY	V/O	0...1H	0...1	1	[-]
ECU Diagnosis active (=1)					
ERR_SYM_ECU_NVMY	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected Symptom ECU					
LV_END_DIAG_ECU_NVMY	V/O	0...1H	0...1	1	[-]
End of ECU Diagnosis					
PWL_KRAM_SAVE	-	0...FFH	0...255	1	[-]
status of PWL					
PWL_KRAM_SAVE_CPL	-	0...FFH	0...255	1	[-]
complement of status of PWL					

Input data:

PWL_LOCK_CDN	LV_WARM_RST	ECU_STATE	V_IGK
C_VB_MIN_OBD1			

Description:

The NVMY blocks are restored after each reset during the initialization phase after start up. If one block is detected as faulty (see “Non volatile data” in chapter Basic SW general operation) only this datas are initialized.

Error symptoms are defined for this diagnosis function as:

```
☐☐☐☐
 │ │ │ └─> NVMY error             (= SYM_0)
 │ │ └───> -                      (= SYM_1)
 │ └─────> -                      (= SYM_2)
 └───────> -                      (= SYM_3)
```

Application conditions:

Initialisation: all 0 at reset **or** clearing error memory

Recurrence: 40 ms (see formula section)

Activation: at reset

Formula section:

at reset:

Error detection

```
If      V_IGK < C_VB_MIN_OBD1                                   or
        [ LV_WARM_RST = 1                                       and
        PWL_KRAM_SAVE == complement of PWL_KRAM_SAVE_CPL        and
        PWL_KRAM_SAVE = only BIT NVMY is set ]
Then    no diagnosis
Else    LV_CDN_DIAG_ECU_NVMY = 1
        If      error symptom is detected
        Then    ERR_SYM_ECU_NVMY = SYM_0
                LV_ERR_ECU_NVMY = 1         /*error is set for this engine run
                                                   no debounce*/
                LV_END_DIAG_ECU_NVMY = 1
        Else    after finishing the test once LV_END_DIAG_ECU_NVMY = 1
        Endif
        LV_CDN_DIAG_ECU_NVMY = 0            //reset to 0 after call of FMY
Endif
```

40ms:

```
If      (ECU_STATE == WAKE_UP           or
        ECU_STATE == PWL)               and
        T_PWL >= C_T_MIN_PWL            and
        PWL_LOCK_CDN Bit NVMY is set
Then    LV_ECU_NVMY_LOCK = 1
Endif
If      LV_ECU_NVMY_LOCK = 1
Then    PWL_KRAM_SAVE = PWL_LOCK_CDN
        PWL_KRAM_SAVE_CPL = complement of PWL_LOCK_CDN
Endif
```

12.3 Overview of Processor Monitoring

12.3.1 System Overview

Based on the recommendations of the VDA (as expressed in the version 2.0 of the paper of the VDA ETC working group "Arbeitskreis EGAS"), the monitoring concept for electronic control units (ECUs) uses three levels; this document provides a more detailed description of the *processor monitoring* level (level 3 in VDA terms).

Based on the VDA guidelines, the processor monitoring level consists of two *separate* hardware components: the main controller (MC) and the monitoring unit (MU). The former is generally realised by a micro controller (µC), while the latter might just as well be realised by an application-specific integrated circuit (ASIC), i.e., some specifically designed piece of hardware; this MU is almost exclusively used for processor monitoring functionality (the only exception is the A/D converter on the MU which is used for the monitoring of the A/D conversion on the MC; this functionality is as such not part of the processor monitoring but of the process monitoring, or "level 2" in VDA terms).

The monitoring concept described here is *generic* in the sense that it works regardless of how the MU is realised in terms of actual hardware (µC vs. ASIC), and for both engine management systems (EMSs) for gasoline/diesel engines and transmission control units (TCUs).

The two system components, MC and MU, *mutually* monitor each other (just to be clear about the fact that not only the MU monitors the MC). For instance, assume that there is a failure in the calculations of the ALU (arithmetic logic unit) of the MC. This could produce potentially dangerous results: e.g., due to a faulty calculation, the ECU might increase the engine speed too much for the driver to control the car. Additionally, the MC itself might not be able to detect such a fault and react appropriately (e.g., perform a reset, or switch off the engine): after all, it cannot even be guaranteed that the processor notices the fault, a (from the external point of view) faulty processor might ‘internally’ believe that the system, including itself, is functioning correctly.

In order to be able to detect such a case, the MU provides an *independent* (hardware) instance which can judge (as far as it is safety-relevant) whether the behaviour of the MC is appropriate according to actual environment conditions and demands (like the driver’s wish expressed, e.g., in terms of pushing the accelerator pedal); the behaviour of the MU, in turn, is then monitored by the MC in order to ascertain that the MU is still able to perform its monitoring duties.

All of these mutual monitoring activities are achieved by means of a repeated ‘*question and answer*’ (Q&A) game: the MU sends questions to the MC, and the MC sends back the respective answers. Based on the quality of the sent questions and received answers, *both* components can tell whether the respective other component is still working correctly.

If there are faults detected while the ECU is running (by whatever means available), appropriate fault reactions are carried out, either by the MC or by the MU, whichever detects the error. As most of the monitored functionalities are safety-critical, such fault reactions or processor monitoring necessitate (at least temporarily) disabling of the safety-relevant power stages in order to reach the so-called *safe state*. In order to be able to perform this type of fault reaction, there have to be corresponding connections (‘*switch-off paths*’) on the hardware level between MC/MU and the relevant power stages (e.g., H-bridge and injectors for gasoline engines). Note that, if the MU detects a fault on the MC, disables the power stages, and triggers a reset of the MC (the standard fault reaction), this reset is then fed back from the MC to the MU so that the *complete* system is restarted (potentially utilising external components like, e.g., power supply).

It should be stressed at this point that, as far as safety is concerned, the **primary goal** of the **fault reaction** is to **disable the power stages**: this prevents an error to result in an undesired acceleration of the car (or, correspondingly, unwanted shifting of gears or erroneous opening/closing of the clutch in TCU applications). The fact that (up to a certain maximal number) a reset is triggered after this disabling is only done in order to increase the **availability** of the car (i.e., to potentially enable the driver to continue his driving relatively undisturbed without having to use the ignition key)!

Coming back to the main task of processor monitoring, in order to set up the aforementioned Q&A game, the two components need to exchange data: this *communication interface* is usually implemented in the form of the S*erial* P*eripheral* I*nterface* (SPI). In this way, MC and MU *synchronise* their progress and, in doing so, can monitor the performed actions.

As an immediate consequence of the separation on the hardware level, since each component has its own parameter memory for variables and is running its own algorithms (in case the MU is realised by an ASIC, this refers to the circuitry that is hard-wired into the chip), variables on the MU are, a priori, not visible (i.e., available) on the MC, and vice versa, unless they are specifically transferred to the other component via the communication interface: variables are *local* to the MU or the MC. Moreover, even after being transmitted from one component to the other, variables on the MC and the MU will usually have different names since the communication is based on *values*, not *names* (despite ‘containing’ the same values!), and variable names reflect the content of the variables, which can very well be different on MC and MU.

It should furthermore be noted that any value exchanged between MC and MU can always only be a *snapshot* of a variable since it can only contain the actual value of the variable *at the time of the transmission*: consider the case a variable X is sent from the MU to the MC at a certain time t; if the variable is changed on the MU at time t+1, this update stays *local to the MU* in the sense that the MC only notices it if the changed X is re-transmitted to the MC after the update.

From a functional point of view, the main purpose of the processor monitoring is to monitor the proper functioning of the MC, thereby, continuing the layer-approach of the over-all system design (level 1 contains engine control functions and their diagnoses, level 2 monitors these functions and diagnoses by re-calculating the necessary values in an independent way to provide a means of judging their plausibility), in particular and most importantly, ensuring the proper functioning of the process monitoring. From this approach, the following main objectives of processor monitoring are derived:

- *memory checks on MC* (both RAM and ROM);
- *memory checks on MU* (RAM and ROM); this depends on the hardware realisation of the MU; on an ASIC, a RAM check shall be performed as usual; however, in place of a ROM check, there shall be a built-in self test (BIST) which tests the proper functioning of certain parts of the circuitry (mostly those parts responsible for triggering a fault reaction);
- *switch-off path check*: after each (re-)initialisation of the system, it must be checked whether the switch-off paths of the system work properly, viz., whether they can actually switch off the safety-relevant power stages should there be the need to do so later on;

- *program flow monitoring (PFM)*: it shall be checked that (monitoring-relevant) tasks (of level 2) on the MC are performed in the correct order and with the correct timing; here, “task” refers to a given and fixed sequence of modules running in the same nominal time grid (i.e., functionally connected parts of processor monitoring);
- *function-specific instruction set test (FS-IST)*: it shall be tested whether the processor correctly performs the (monitoring-relevant) calculations needed for the process monitoring on the MC; this shall be achieved by performing a number of *test calculations*; in order to avoid any effects of these (additional) test calculations on the actual monitoring functionality (and in line with VDA requirements), these test calculations are performed by a *copy* of some of the (monitoring-relevant) parts of the process monitoring located in separate memory on the MC which (being a copy of level 2 functionality) is also known as ”*level 2’*” (L2’).

Note that PFM and the FS-IST complement each other to achieve as complete a monitoring of the processor functionality as possible: while PFM checks whether the necessary tasks are executed in the *correct order* and with the *correct timing*, the FS-IST additionally checks that the required *calculations are performed correctly* (so that the tasks can properly execute and calculate the ‘right’ result - provided the employed algorithm is correct and correctly implemented: both PFM and L2’ can only achieve ‘formal’ or ‘syntactic’ correctness, ‘semantic’ correctness lies in the hands of the developer and cannot be tested (in this way)!).

A schematic overview of the system is presented in the following figure “Sketch of processor monitoring with two power stages,” including the required disable lines from the MC and the MU to the power stages; note that this picture does not include the redundant analogue input signal based on which the A/D conversion on the MU is performed and then sent to the MC in order to determine proper functioning of the A/D converter on the MC (by comparing the conversion result sent by the MU with the result obtained on the MC).

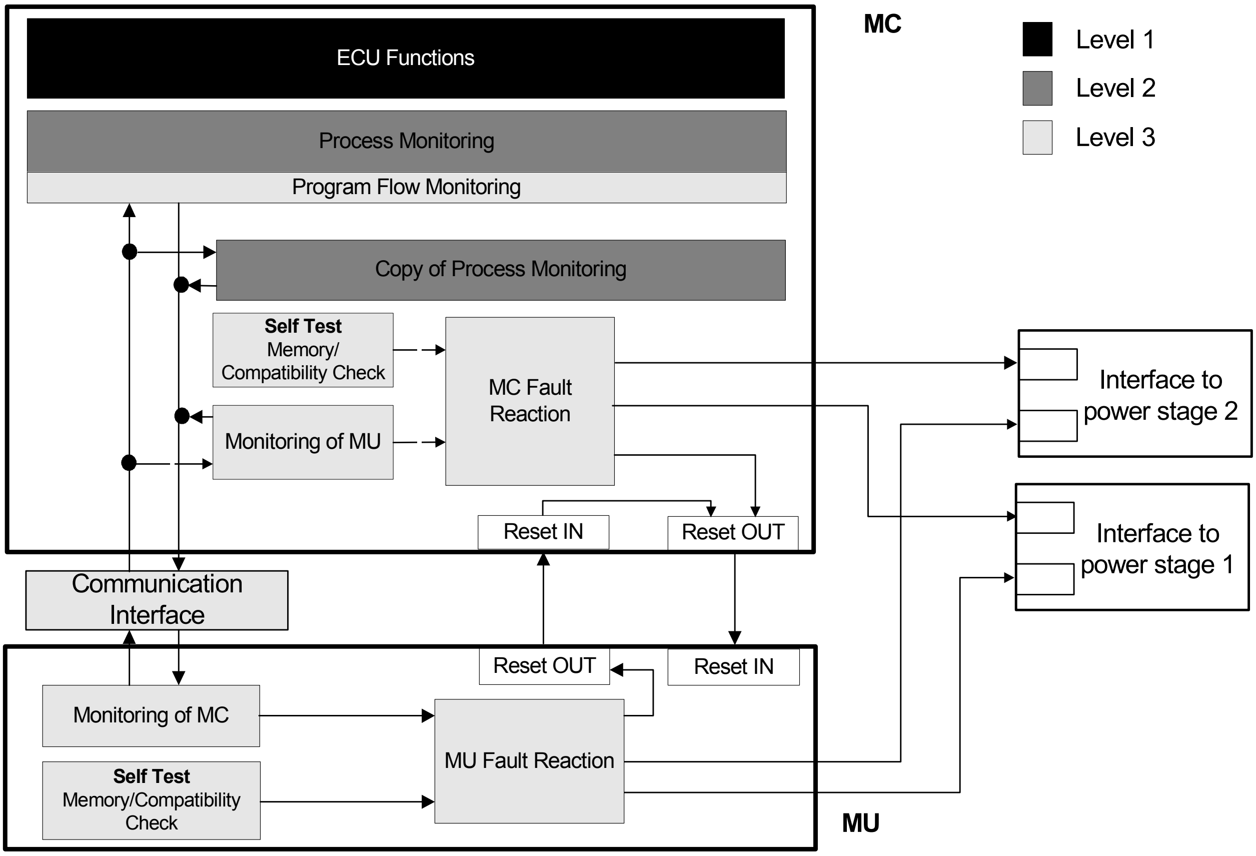


Figure 1: Sketch of processor monitoring with two power stages

12.3.2 States of the Components

The behavior of both components of the processor monitoring (i.e., MC and MU) is described by two *autonomous* state machines, reflecting the fact that both constitute separate components of a distributed system and as such should really have a separate state space.

Even so, the structure of the two state machines is basically identical, mostly because the two components have a common task, viz., perform the processor monitoring. Nevertheless, as the two parts run independently of each other, both could be in mismatching states; it is the task of the communication interface to prohibit such cases by ensuring that the components *synchronously* progress through their states in the correct order. The exception to this rule is the DISABLE state which is entered after a fault was detected (on the component which detected the fault); this knowledge can then only be propagated to the other component by means of a communication, e.g., it can be the case that, for one communication cycle, the MU is in DISABLE while the MC is still in NORMAL (after the next communication, the MC will also go to DISABLE).

The basic layout of the two state machines is described the figure “State machine layout.” The figure contains the basic transitions sequence through the different states. We shall now describe the actions which are performed in each of the states (for the time being without distinguishing between which actions are performed on MC and MU, respectively). Communications between MC and MU are an integral part of the system’s functionality and, in any case, information about the current state of a component is always part of the data exchanged by the communications.

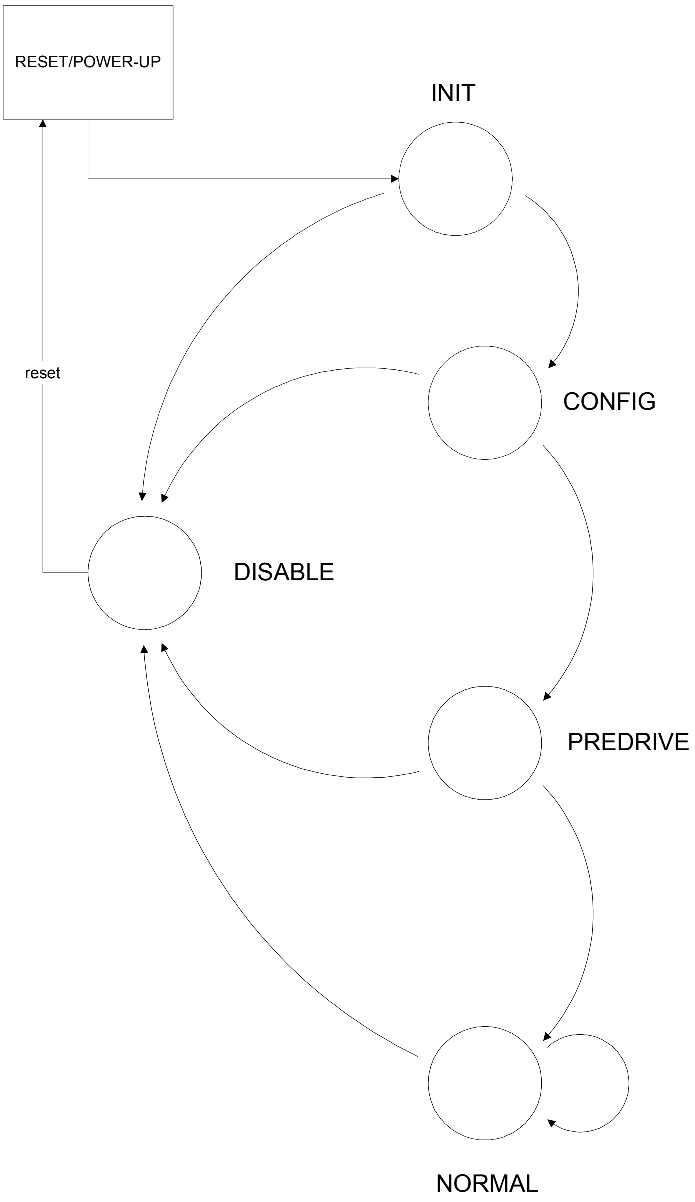


Figure 2: State machine layout

12.3.2.1 The INIT state

As indicated by its name, the INIT state is the initial state of the state machine, i.e., always after power-up or a reset, ‘start-up’ tasks like data initialisation and several self tests on both MC and MU are performed in INIT. Note that there is *no* communication performed while the two components are in their respective INIT states.

12.3.2.2 The CONFIG state

In the error free case, the next state is the CONFIG state which is mainly used to prepare and perform the first communication between MC and MU; consequently, the tasks carried out in this state are to prepare the communication, wait for it, and afterwards evaluate the received data. The exchanged data include error codes and reset counters, together with configuration and compatibility data; this communication is usually referred to as the

"CONFIG communication" because if everything is working normally, both components will be in their respective CONFIG states.

12.3.2.3 The PREDRIVE state

In the PREDRIVE state, another communication (the "PREDRIVE communication") is performed which serves the purpose to initialise the PFM (see section "Program Flow Monitoring" for both MC and MU) and the FS-IST (see module "Function-Specific Instruction Set Test" on both MC and MU) with the necessary data; also the configuration data transferred in the previous communication is sent back in order to double-check correct reception; additionally, from this state onwards, the monitored A/D conversion values are transferred from the MU to the MC so that the value obtained on the MC can be checked against the conversion values (of the same signal!) from the MU.

12.3.2.4 The NORMAL state

The NORMAL state is the state which is reached when all the previous actions (including communications) were performed successfully; it represents a properly functioning system. Since there are several tasks which have to be continuously performed (e.g., PFM, FS-IST), there are ongoing communications ("NORMAL communication") in the NORMAL state which continue the exchange of the necessary data for such continuous tasks.

12.3.2.5 The DISABLE state

The DISABLE state is reached if an error occurs during any the actions in the other states, see Figure 2; such errors include communication errors or errors detected during the Switch-off Path Check (see section "Switch-Off Path Check"). Consequently, since the system is malfunctioning, *the power stages are disabled*, by the component that enters its respective DISABLE state. In a corresponding communication ("DISABLE communication"), error information (like reset counters and error codes) are then exchanged.

12.3.3 Communication protocol between MC and MU

In the following, the communication protocol between MC and MU shall be described where the communications in each state are described on after the other.

The description for each communication is given in such a way that always the data flow in both directions (MC to MU, and MU to MC) is depicted in a separate column.

Note that in following description of the communication protocol, all variables are named as they are known on the MC, the MU is regarded as a black-box.

A row in the tables corresponds to one byte, and where more than one data is sent in one byte together this indicated accordingly (e.g., in the CONFIG state communication, ERR_COD_MC and STATE_ERR_IDC_MC are sent in a combined form as byte1 of the data sent from the MC to the MU).

The first byte of each communication in both directions is always a header byte reflecting the internal state of the sending component.

The last byte of each communication in both directions is always the (XOR) checksum of the previous 5 bytes (incl. header-byte) to detect communication errors.

12.3.3.1 CONFIG state communication

MC → MU	MU → MC

NC_STATE_MC_CONF	NC_STATE_MC_CONF
STATE_ERR_IDC_MC, ERR_COD_MC **(P)**	ERR_COD_MU, STATE_ERR_IDC_MU **(P)**
CTR_RST_MC **(P)**	LV_V_L_DET_MU, LV_V_H_DET_MU CTR_RST_MU **(P),**
Configuration data 1: NLC_MU_MOD NLC_MU_IGN_ACQ NLC_MU_INJ_OFF_TRM_ENA NLC_MU_POW_OFF_TRM_ENA Configuration data 2: 0 **(P)**	NR_MU_COMP, NR_MU_HW_VERS
Complement of configuration data 1 and 2 **(P)**	Empty (0x00)
Checksum	Checksum

(P) = Due to the high importance of the information, these data bytes are protected by a parity bit. A parity error (odd instead of even parity) is regarded as a communication error, see "G45 Parity-bit checking."

- MC and MU exchange error codes, error indications, and reset counters (the error code and error indication of the MU are sent as LF_MU_ERR_BYTE and then "decoded" to ERR_COD_MU and STATE_ERR_IDC_MU).
- The flags LV_V_L_DET_MU and LV_V_H_DET_MU are set if under- or over-voltage, respectively, has been detected on the MU.
- Based on the compatibility information NC_MU_COMP and HW version NC_MU_HW_VERS which is sent by the MU, the MC can check whether its software is compatible to the MU.
- By sending configuration data 1 and 2 during the CONFIG communication, the MU is configured by the MC: monitoring an EMS or a TCU; the acquisition of the ignition key signal can be configured to HW or SW mode; the IOT and POT functionality can be enabled; the 0 in Configuration data 2 stands for not performing EOLT.

12.3.3.2 PREDRIVE state communication

MC → MU	**MU → MC**
NC_STATE_MC_PRDR	NC_STATE_MC_PRDR
NR_MC_COMP	LF_MC_RCV_PFM_BYTE
Empty (0x00)	LV_MU_DI_OUT_0, LV_MU_DI_OUT_1, IDX_TEST_RCV_MC_FCT_SPC_IST
Configuration data 1: NLC_MU_MOD, NLC_MU_IGN_ACQ, NLC_MU_INJ_OFF_TRM_ENA, NLC_MU_POW_OFF_TRM_ENA Configuration data 2: 0 **(P)**	LV_MU_IGN_KEY, LF_MU_CONF_BYTE_PRDR (combining Configuration data 1 and Configuration data 2)
Complement of configuration data 1 and 2 **(P)**	VP_MU_AN_DIG_MON
Checksum	Checksum

(P) = Due to the high importance of the information, these data bytes are protected by a parity bit. A parity error (odd instead of even parity) is regarded as a communication error, see "G45 Parity-bit checking."

- The MU sends the first PFM synchronisation byte LF_MC_RCV_PFM_BYTE, the first FS-IST question IDX_TEST_RCV_MC_FCT_SPC_IST, and the converted analogue value VP_MU_AN_DIG_MON.
- The configuration data 1 and 2 are confirmed by the MC: now the MU is completely configured.
- To ensure that the MU is always working in the correct mode, it reads back all of its internally stored configuration data and sends it back to the MC as LF_MU_CONF_BYTE_PRDR.
- Based on the compatibility information NR_MC_COMP sent by the MC, the MU can check whether it is compatible to the MC.
- The MU shall send the status of its disable outputs in LV_MU_DI_OUT_0 and LV_MU_DI_OUT_1.

- The MU sends the latched valued of its input for the ignition key signal as LV_MU_IGN_KEY (regardeless of whether it was configured for SW or HW acquisition of said signal).

12.3.3.3 NORMAL state communication

MC → MU	MU → MC
NC_STATE_MC_NORM	NC_STATE_MC_NORM
LF_MC_RESP_PFM	LF_MC_RCV_PFM_BYTE
LV_MC_DR_OFF, LV_IGK, IDX_RESP_TRM_FCT_SPC_IST	LV_MU_DI_OUT_0, LV_MU_DI_OUT_1, IDX_TEST_RCV_MC_FCT_SPC_IST
Empty (0x00)	LV_MU_IGN_KEY, LF_MU_CONF_BYTE_NORM, ABC_RCV_MC_FCT_SPC_IST
Empty (0x00)	VP_MU_AN_DIG_MON
Checksum	Checksum

- The MU keeps on sending FS-IST questions IDX_TEST_RCV_MC_FCT_SPC_IST, and additionally the FS-IST anti-bounce counter ABC_RCV_MC_FCT_SPC_IST, and the monitored analogue value VP_MU_AN_DIG_MON.
- LF_MC_RCV_PFM_BYTE is either an updated PFM synchronisation byte or a status byte where a set bit indicates that the corresponding PFM anti-bounce counter on the MU has value >0.
- The MC sends the PFM answer byte LF_MC_RESP_PFM and the FS-IST answer IDX_RESP_TRM_FCT_SPC_IST.
- Additionally, it is indicated whether wrong FS-IST answers are sent because the redundant switch off path is activated via the flag LV_MC_DR_OFF.
- As the information whether the ignition key is switched on or off is important information, as it is used to determine whether the injection will be switched off (via the IOT), the MC sends its information LV_IGK to the MU.
- If the ignition key is connected to the CAN bus, the state of the ignition key cannot be acquired by the MU. In this case the IGK acquisition of the MU must be configured to software acquisition, i.e., LV_IGK is used to determine whether the IOT or the POT has to be started.

- To ensure that the MU is always working in the correct mode, it reads back its internally stored configuration data 1 and sends it back to the MC in the form of LF_MU_CONF_BYTE_NORM.
- The MU shall send the status of its disable outputs in LV_MU_DI_OUT_0 and LV_MU_DI_OUT_1.

- The MU sends the latched valued of its input for the ignition key signal as LV_MU_IGN_KEY (regardeless of whether it was configured for SW or HW acquisition of said signal).

12.3.3.4 DISABLE state communication

MC → MU	**MU → MC**
NC_STATE_MC_DI	NC_STATE_MC_DI
STATE_ERR_IDC_MC ERR_COD_MC **(P)**	LF_MU_ERR_BYTE **(P)** (Combining ERR_COD_MU and STATE_ERR_IDC_MU)
LV_IGK CTR_RST_MC **(P)**	CTR_RST_MU **(P)**
Empty (0x00)	LV_MU_IGN_KEY
Empty (0x00)	VP_MU_AN_DIG_MON
Checksum	Checksum

(P) = Due to the high importance of the information, these data bytes are protected by a parity bit. A parity error (odd instead of even parity) is regarded as a communication error, see "G45 Parity-bit checking."

- MC and MU exchange error codes, error indications, and reset counters (the error code and error indication of the MU are sent as LF_MU_ERR_BYTE and then "decoded" to ERR_COD_MU and STATE_ERR_IDC_MU).
- The MC sends its value of the ignition key signal, so does the MU.
- The MU sends the monitored analogue value VP_MU_AN_DIG_MON.

12.3.4 Table of error codes of MC and MU

The following table shows the meanings of the different error codes of MC in order to make it easier to analyse them in case of an error detected by Processor Monitoring.

Value	ERR_COD_MC	ERR_COD_MU
0	No error	No error
1	Header-byte error detected on MC	Header-byte error detected on MU
2	Checksum error detected on MC	Checksum error detected on MU
3	Parity error detected on MC	Parity error detected on MU
4	Wrong configuration data sent by MU	Wrong configuration data sent by MC
5	Error in switch-off path check	Error in complement check of configuration data
6	Error in compatibility check on MC	Error in compatibility check on MU
7	Error in Standard ROM check	Communication maximum time-out monitoring
8	Error in Standard RAM check	Communication minimum time-out monitoring
9	Error in Level 2 ROM check	Error detected by ignition key monitoring on MU
10	Error in Level 2 RAM check	NORMAL instead of EOLT header received
11	FS-IST: MU sent wrong question	FS-IST: MC sent wrong answer
12	FS-IST: MU does not react to deliberately wrong answer	Redundant switch-off path hsa been triggered
13	FS-IST: MU does not decrease its FS-IST anti-bounce counter	MU has detected under-voltage
14	Error in check of MU readiness	MU has detected over-voltage
15	- *(not used)*	MU has detected under- and over-voltage
16	PFM: MU does not toggle PFM bit 0 in time	PFM task 0 executed to slowly on MC
17	PFM: MU does not toggle PFM bit 1 in time	PFM task 1 executed to slowly on MC
18	PFM: MU does not toggle PFM bit 2 in time	PFM task 2 executed to slowly on MC
19	PFM: MU does not toggle PFM bit 3 in time	PFM task 3 executed to slowly on MC
20	PFM: MU does not toggle PFM bit 4 in time	PFM task 4 executed to slowly on MC
21	PFM: MU does not toggle PFM bit 5 in time	PFM task 5 executed to slowly on MC
22	PFM: MU does not toggle PFM bit 6 in time	PFM task 6 executed to slowly on MC
23	Invalid state transition or undefined state on MC	Invalid state transition or undefined state on MU
24	PFM: MU does not react to deliberately wrong PFM bit 0	- *(not used)*
25	PFM: MU does not react to deliberately wrong PFM bit 1	- *(not used)*
26	PFM: MU does not react to deliberately wrong PFM bit 2	- *(not used)*
27	PFM: MU does not react to deliberately wrong PFM bit 3	- *(not used)*

28	PFM: MU does not react to deliberately wrong PFM bit 4	- *(not used)*
29	PFM: MU does not react to deliberately wrong PFM bit 5	- *(not used)*
30	PFM: MU does not react to deliberately wrong PFM bit 6	- *(not used)*
31	- *(not used)*	- *(not used)*

12.4 MC Processor Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_MC	V/O	0...FFH	0...255	1	[-]
Internal state of the MC (also used as header byte in communications)					
LV_MC_ROM_CHK_READY	V/O/S	0...1H	0...1	1	[-]
Flag indicating whether the level 1 ROM check is ready					
LV_MC_ROM_CHK_READY_CPL	O/S	0...1H	0...1	1	[-]
Complement of LV_MC_ROM_CHK_READY					
LV_MC_ROM_CHK_OK	-	0...1H	0...1	1	[-]
Flag indicating whether the level 1 ROM check was successful					
LF_MU_CONF_BYTE	V	0...FFH	0...255	1	[-]
Local variable: configuration data sent by MU (in PREDRIVE and NORMAL communications)					
NR_MU_COMP	V/O	0...7H	0...7	1	[-]
Compatibility number sent by MU					
NR_MU_HW_VERS	V/O	0...1FH	0...31	1	[-]
Hardware version sent by MU					
VP_MC_AN_DIG_MON	V/O	0...7FFFH	0...4.99984	0.1526e-3	[V]
Result of A/D conversion on the MC (exported to process monitoring)					
VP_MC_AN_DIG_MON_CPL	V/O	0...7FFFH	0...4.99984	0.1526e-3	[V]
Complement of V_MC_AN_DIG_MON (exported to process monitoring)					
VP_MU_AN_DIG_MON	V/O	0...7FFFH	0...4.99984	0.1526e-3	[V]
Result of A/D conversion on the MU (exported to process monitoring)					
VP_MU_AN_DIG_MON_CPL	V/O	0...7FFFH	0...4.99984	0.1526e-3	[V]
Complement of V_MU_AN_DIG_MON (exported to process monitoring)					
LV_MU_IGN_KEY	O/V	0...1H	0...1	1	[-]
Ignition key signal redundantly acquired on MU (only useful in HW acquisition mode)					
VP_MC_AN_DIG_MON_TMP	-	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Local variable: result of A/D conversion as fetched from infrastructure					
VP_MC_AN_DIG_MON_TMP_CPL	-	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Local variable: complement of VP_MC_AN_DIG_MON_TMP					

Input data:

CTR_RST_MC	NC_CTR_RST_MC_THD	NC_ERR_COD_MC_CONF_DIF	NC_ERR_COD_MC_COMP_NR
LV_MC_IGN_KEY	NC_ERR_COD_MC_NOT_VLD_TRAN	NC_ERR_COD_MC_ROM_LEVEL_1	NC_ERR_COD_MC_ROM_LEVEL_2
NC_ERR_COD_MC_RAM_LEVEL_2	NC_ERR_COD_MC_CONF_DIF	NC_ABC_MC_CONF_DIF	LV_MC_COM_ERR
NC_ERR_COD_MC_COMP_NR	NC_ERR_COD_MC_RAM_LEVEL_1	CTR_RST_MC_CPL	LV_MC_COM_ERR_CPL
LV_MU_READY			

Export actions:

ACTION_ECM3_McChangeState(IN <PRM_STATE_MC_NEW>)
This action changes the state of the MC to the given new state; when already in DISABLE, nothing is done.
ACTION_ECM3_WriteChkCpl(INOUT <PRM_VAR>, OUT <PRM_VAR_CPL>, IN <PRM_VALUE>)
This action writes the value into the RAM and cross-checks the written value; the complement is generated, stored, and the corresponding variable is returned.
ACTION_ECM3_ChkCpl(IN <PRM_VAR>, IN <PRM_VAR_CPL>)
This action compares the value of var with its complement var_CPL.
ACTION_ECM3_ReadChkCpl(OUT <PRM_RESULT>, IN <PRM_VAR>, IN <PRM_VAR_CPL>)
This action checks that var and var_CPL are a valid value-complement pair, and if yes, stores the value in result so that the value can be used in successive calculations.

ACTION_ECM3_McReadChkCpl(OUT <PRM_RESULT>, IN <PRM_VAR>, IN <PRM_VAR_CPL>, IN <PRM_DFT_VALUE>)
This action checks that var and var_cpl are a valid value/complement pair; if no returns the supplied default value and triggers a fault reaction.
ACTION_ECM3_McChkStack()
This action triggers a check of the appropriate stack area by the infrastructure and evaluates its result.
ACTION_ECM3_McReadChkState (OUT <PRM_STATE_MC_TMP>)
This action checks the current state for being valid.
ACTION_ECM3_McResetRomChkFlags()
This action resets the flags for the Standard ROM check in case this becomes necessary.

Description for actions:

ACTION_ECM3_McChangeState(PRM_STATE_MC_NEW)

Change STATE_MC to new state.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
PRM_STATE_MC_NEW	IN	0...FFH	0...255	1	[-]
Representation of state to be entered					

ACTION_ECM3_WriteChkCpl(PRM_VAR, PRM_VAR_CPL, PRM_VALUE)

see description in section "RAM check service actions of Processor Monitoring on MC"

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
PRM_VAR	INOUT	*any*	*any*	*any*	*any*
Variable to be checked					
PRM_VAR_CPL	OUT	*any*		*any*	*any*
Variable for the complement of PRM_VAR					
PRM_VALUE	IN	*any*	*any*	*any*	*any*
Value to be stored					

ACTION_ECM3_ChkCpl(PRM_VAR, PRM_VAR_CPL)

see description in section "RAM check service actions of Processor Monitoring on MC"

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
PRM_VAR	IN	*any*	*any*	*any*	*any*
Variable to be checked					
PRM_VAR_CPL	IN	*any*		*any*	*any*
Variable for the complement of PRM_VAR					

ACTION_ECM3_ReadChkCpl(PRM_RESULT, PRM_VAR, PRM_VAR_CPL)

see description in section "RAM check service actions of Processor Monitoring on MC"

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
PRM_RESULT	OUT	*any*	*any*	*any*	*any*
Variable to be checked					
PRM_VAR	IN	*any*		*any*	*any*
Variable for the complement of PRM_VAR					
PRM_VAR_CPL	IN	*any*	*any*	*any*	*any*
Value to be stored					

ACTION_ECM3_McReadChkCpl(PRM_RESULT, PRM_VAR, PRM_VAR_CPL, PRM_DFT_VALUE)					
see description in section "RAM check service actions of Processor Monitoring on MC"					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_RESULT	OUT	*any*	*any*	*any*	*any*
Variable to be checked					
PRM_VAR	IN	*any*		*any*	*any*
Variable for the complement of PRM_VAR					
PRM_VAR_CPL	IN	*any*	*any*	*any*	*any*
Value to be stored					
PRM_DFT_VALUE	IN	*any*	*any*	*any*	*any*
Default value for PRM_VAR in case an inconsistency is detected					

ACTION_ECM3_McChkStack()					
The action triggers a stack check via an infrastructure action and evaluates the result of the check.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**

ACTION_ECM3_McReadChkState(<PRM_STATE_MC_TMP>)					
The action reads the internal representation of the current state (STATE_MC) and checks it for validity; in case of an inconsistency, the return value shall be NC_STATE_MU_DI (the representation for DISABLE state) and a fault reaction shall be triggered.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_MC_TMP	OUT	0...FFH	0...255	1	[-]
Short description of the given parameter					

ACTION_ECM3_McResetRomChkFlags()					
This action resets the flags LV_MC_ROM_CHK_READY and LV_MC_ROM_CHK_OK; this is used by DISABLE requests.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**

Import actions:

ACTION_ECM3_McFaultReaction(IN <>)
ACTION_ECM3_RedSwitchOffPath()
ACTION_ECM3_McIncAbc(IN <>, IN <>)
ACTION_ECM3_McDecAbc(IN <>)
ACTION_ECM3_McLockPwlResource()
ACTION_ECM3_McUnlockPwlResource()
ACTION_INFR_McRomChkReady(OUT <>)
ACTION_INFR_McRomChkOk(OUT <>)
ACTION_INFR_McGetConfData(OUT <>)
ACTION_INFR_McGetCompNr(OUT <>)
ACTION_INFR_McGetVersNr(OUT <>)
ACTION_INFR_GetAnDigMonValMc(OUT <>)
ACTION_INFR_GetAnDigMonValMu(OUT <>)
ACTION_INFR_McGetIgkMu(OUT <>)
ACTION_INFR_McSendConfData(IN <>, IN <>, IN <>, IN <>)
ACTION_INFR_McSendCompNr(IN <>)
ACTION_INFR_McSendIgk(IN <>)
ACTION_INFR_McChkStack(OUT <>)

FUNCTION DESCRIPTION:

General information:

This module describes the main function of processor monitoring on the MC, in particular the state machine.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ERR_SWI_MC_LEVEL_2_CAL	1	0...FFH	0...255	1	[-]
Switch to disregard ROM errors in the level 2 calibration data (when set to NC_ERR_SWI_MC_LEVEL_2_CAL)					
C_ERR_SWI_MC_LEVEL_2_COD	1	0...FFH	0...255	1	[-]
Switch to disregard ROM errors in the level 2 code (when set to NC_ERR_SWI_MC_LEVEL_2_COD)					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_ERR_SWI_MC_LEVEL_2_CAL	1	0...FFH	0...255	1	[-]
Value for the switch which allows to disregard ROM errors in the level 2 calibration area					
NC_STATE_MC_INI	1	0...FFH	0...255	1	[-]
Internal representation for INIT state					
NC_STATE_MC_CONF	1	0...FFH	0...255	1	[-]
Internal representation (and header byte) for CONFIG state					
NC_STATE_MC_PRDR	1	0...FFH	0...255	1	[-]
Internal representation (and header byte) for PREDRIVE state					
NC_STATE_MC_NORM	1	0...FFH	0...255	1	[-]
Internal representation (and header byte) for NORMAL state					
NC_STATE_MC_DI	1	0...FFH	0...255	1	[-]
Internal representation (and header byte) for DISABLE state					
NLC_MU_MOD	1	0...1H	0...1	1	[-]
Configuration of MU mode (monitoring of EMS=0 or TCU=1)					
NLC_MU_IGN_KEY_ACQ	1	0...1H	0...1	1	[-]
Configuration for acquisition of IGK signal on the MU (by software=0 or hardware=1)					
NLC_MU_POW_OFF_TMR_ENA	1	0...1H	0...1	1	[-]
Configuration of MU for enabling/disabling the power-off timer					
NLC_MU_INJ_OFF_TMR_ENA	1	0...1H	0...1	1	[-]
Configuration of MU for enabling/disabling the injection-off timer					
NC_NR_MC_COMP	1	0...7H	0...7	1	[-]
Compatibility information of MC					
NC_VP_MC_AN_DIG_MON_CPL_ERR	1	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Constant indicating a complement error of the value of the A/D conversion on the MC acquired in the infrastructure					
NC_ERR_SWI_MC_LEVEL_2_COD	1	0...FFH	0...255	1	[-]
Value for the switch which allows to disregard ROM errors in the level 2 code					
NC_STATE_MC_NOT_VLD	1	0...FFH	0...255	1	[-]
Internal representation for invalid state					

12.4.1 State machine on MC

FUNCTION DESCRIPTION:

General information:

This function represents the main control flow in the MC; based on the current state of the MC, other modules/functions are activated or deactivated. The internal state machine of the MC governs which actions on the MC shall be performed, e.g.:

- which data are sent to the MU;
- which actions should be taken upon receiving data, in particular, whether the received data are acceptable and the MU is still in sync with the MC;
- whether the power stages should be disabled or enabled.

As described schematically in the signal flow diagram, given a state, there are only a certain number of other states reachable.

Signal flow diagram:

The following diagram shows the state machine of the MC with all the possible transitions.

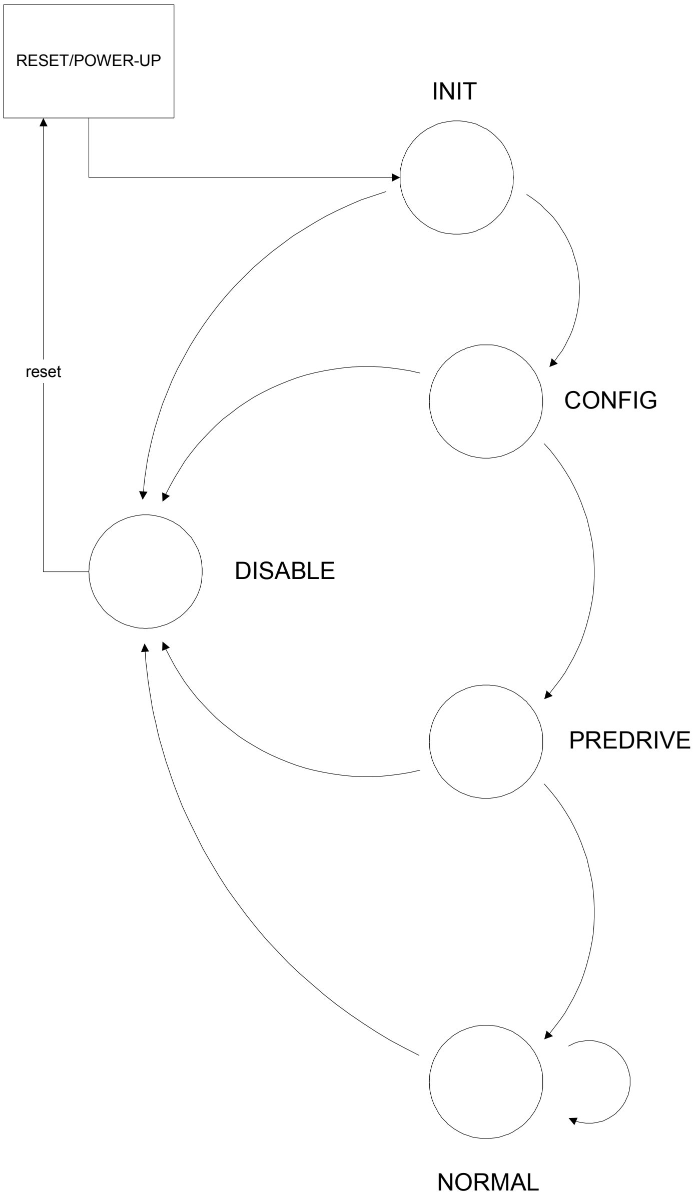


Figure 3 State machine of the MC

Description:

This function shall incorporate the following requirements:

G51 State machine on MC

R51_1: All state changes shall be performed using the action ACTION_ECM3_McChangeState().

R51_2: The possible state changes are described in the signal flow diagram above; no other transition between the states shall be performed.

R51_3: A valid transition from one state to the other (according to the signal flow diagram) shall be performed when all tasks of the given state are finished; an exception is the transition to DISABLE state: this shall be performed immediately when a transition condition is given (e.g., when a fault reaction on the MC is carried out, or when the MU sends a DISABLE header).
Note: a rough sequencing of tasks and state transitions can be given as follows

- evaluation of data from previous communication
- state transition (if necessary)
- preparation of new communication
- triggering of new communication

R51_4: The internal representations of states/header bytes shall be chosen in such a way that the minimum of the pair-wise hamming distances is as high as possible, i.e., in order to change from one valid representation of a state into another valid representation of a different state, several bits have to toggle at once.

R51_5: The same bit patterns shall be used for internal representation of states and for the corresponding header bytes sent to the MU.

R51_6: Concerning the transition from INIT to CONFIG:
R51_6.1: there shall be a project-specific transition condition (cf. G52);
R51_6.2: the transition shall only be performed if all tasks in INIT have been completed (cf. activation conditions) AND LV_MU_READY = 1 AND the project-specific condition is satisfied.
Note: the actual state change from INIT to CONFIG can be performed by the function G52 evaluating the project-specific transition condition; however, R51_6.2 then has to be ensured by the control flow (e.g., the project-specific condition is only evaluated if all other conditions like LV_MU_READY = 1 are already satisfied).

Description for ACTION_ECM3_McChangeState:

ACTION_ECM3_McChangeState(PRM_STATE_MC_NEW)					
Change STATE_MC to new state.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_MC_NEW	IN	0...FFH	0...255	1	[-]
Representation of state to be entered					

Formula section:

```
/* When already in DISABLE, no transition shall be possible: see signal flow diagram */
IF (STATE_MC <> NC_STATE_MC_DI)
THEN
    STATE_MC = PRM_STATE_MC_NEW;
```

FI

Application conditions:

Initialisation: *`/* Has the reset counter exceeded its threshold: if so, perform start-up in DISABLE; otherwise in INIT (note: also CTR_RST_MC_CPL has to be checked, using the actions from "RAM check service actions of processor monitoring on MC") */`*

```
STATE_MC = NC_STATE_MC_INI;
IF (CTR_RST_MC > NC_CTR_RST_MC_THD)
THEN
    STATE_MC = NC_STATE_MC_DI;
ELSE
```

Recurrence: The **initialisation** performed **after each power-up or reset**.

Afterwards, the function is **called**.

Activation: -

Deactivation: -

12.4.2 Monitoring of state machine

FUNCTION DESCRIPTION:

Description:

This function shall incorporate the following requirements:

G9 Monitoring of state machine

R9_1: The state machine on the MC shall be monitored in by testing whether STATE_MC is still valid (i.e., that it is equal to one of NC_STATE_MC_INI, NC_STATE_MC_CONF, NC_STATE_MC_PRDR, NC_STATE_MC_NORM, NC_STATE_MC_DI).

R9_2: If it is detected that the MC is an undefined state:
R9_2.1: it shall be assumed to be in DISABLE state;
R9_2.2: an immediate fault reaction shall be performed by calling ACTION_ECM3_McFaultReaction() with error location NC_ERR_COD_MC_NOT_VLD_TRAN.

R9_3: The function shall be incorporated into the action ACTION_ECM3_McReadChkState() as described in the formula section below.

Description for ACTION_ECM3_McReadChkState:

ACTION_ECM3_McReadChkState(<PRM_STATE_MC_TMP>)					
The action reads the internal representation of the current state (STATE_MC) and checks it for validity; in case of an inconsistency, the return value shall be NC_STATE_MU_DI (the representation for DISABLE state) and a fault reaction shall be triggered.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_MC_TMP	OUT	0...FFH	0...255	1	[-]
Short description of the given parameter					

Formula section:

```
IF ("STATE_MC is valid") /* see R9_1 */
THEN
    PRM_STATE_MC_TMP = STATE_MC
ELSE /* invalid state */
    PRM_STATE_MC_TMP = NC_STATE_MC_DI;
    ACTION_ECM3_McFaultReaction (IN        NC_ERR_COD_MC_NOT_VLD_TRAN);
FI
```

Application conditions:

Initialisation: -

Recurrence: The function is called via ACTION_ECM3_McReadChkState().

Activation: -

Deactivation: -

12.4.3 ROM checks of processor monitoring on MC

FUNCTION DESCRIPTION:

General information:

On the MC, there are two different ROM checks:

- the *standard ROM check* which checks the *whole* ROM (code, constants, calibration data with the exception of areas which can be changed during runtime like FLASH or EEPROM);
- the *cyclic ROM check for process monitoring* which checks the ROM of process monitoring (consisting of the code, constants, and calibration data of process monitoring, plus the code of all library functions called in the code of process monitoring).

The difference between the two ROM checks is the fact that the standard ROM check is performed only *once* after power-up (after reset only if the reset occurred before its successful completion) while the ROM check for process monitoring is *cyclic*, i.e., it is restarted after completion.

12.4.3.1 Standard ROM check on MC

FUNCTION DESCRIPTION:

General information:

This sub-function describes the standard ROM check (a.k.a. "level 1 ROM check") on the MC which is performed once per driving cycle directly after power-up or reset.

The two actions ACTION_ECM3_McLockPwlResource() and ACTION_ECM3_McUnlockPwlResource() are used to ensure that the ECU is not powered down before the ROM check has been completed at least once in the corresponding driving cycle.

Description:

The function incorporates the following requirements:

G8.1 Standard ROM check on MC

R8.1_1: Every 1000ms:
R8.1_1.1: the action ACTION_INFR_McRomChkReady() shall be called;
R8.1_1.2: its return value shall be stored in the flag LV_MC_ROM_CHK_READY (indicating whether the Standard ROM check has been finished or not); the flag shall be stored in a reset-safe manner and together with a complement, LV_MC_ROM_CHK_READY_CPL which shall also be stored in a reset-safe manner; LV_MC_ROM_CHK_READY shall be accessed (read/write) using the actions ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() defined in the section "RAM check service actions of processor monitoring on MC"; its default value in case of a RAM error shall be 0;
R8.1_1.3: if LV_MC_ROM_CHK_READY = 1:
R8.1_1.3.1: the corresponding power-latch resource shall be unlocked by calling ACTION_ECM3_McUnlockPwlResource();
R8.1_1.3.2: the action ACTION_INFR_McRomChkOk() shall be called;
R8.1_1.3.3: its return value shall be stored in the flag LV_MC_ROM_CHK_OK (indicating whether the checksum calculated by the standard ROM check is equal to the expected one);
R8.1_1.3.4: if LV_MC_ROM_CHK_OK = 0:
R8.1_1.3.4.1: LV_MC_ROM_CHK_READY shall be reset to 0 using the RAM check service actions;
R8.1_1.3.4.2: an immediate fault reaction shall be performed by calling ACTION_ECM3_McFaultReaction() with error code NC_ERR_COD_MC_ROM_LEVEL_1.

R8.1_2: This function shall be under PFM control using the 1000ms PFM task (note that this does only refer to the processor monitoring function which only evaluates the result of the standard ROM check; the actual checksum calculation itself is part of the infrastructure and as such cannot be controlled by PFM).

Description for ACTION_ECM3_McResetRomChkFlags:

ACTION_ECM3_McResetRomChkFlags()					
This action resets the flags LV_MC_ROM_CHK_READY and LV_MC_ROM_CHK_OK; this is used by DISABLE requests.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**

Formula section:

LV_MC_ROM_CHK_OK = 0;

/ Using the RAM check service actions */*

LV_MC_ROM_CHK_READY = 0;

Application conditions:

Initialisation: After power-up:

/ Using the RAM check service actions */*
LV_MC_ROM_CHK_READY = 0;
/ Lock power-latch resource */*
ACTION_ECM3_McLockPwlResource();

After power-up or reset:

LV_MC_ROM_CHK_OK = 0;
/ Lock power-latch resource */*
ACTION_ECM3_McLockPwlResource();

Recurrence: Every 1000ms

Activation: (STATE_MC <> NC_STATE_MC_DI) **AND**
(LV_MC_ROM_CHK_READY = 0)
/ using the RAM check service actions */*
(not in DISABLE state and not yet completed)

Deactivation: otherwise;
at deactivation, call ACTION_ECM3_McUnlockPwlResource()
/ after transition to DISABLE or after completion of ROM check: unlock power-latch resource */*

12.4.3.2 Cyclic ROM check on MC for process monitoring

FUNCTION DESCRIPTION:

General information:

The cyclic ROM check for process monitoring ("level 2 ROM check") checks all level 2 ROM areas, which are different from the level 1 ROM areas, as well as the corresponding constant and calibration data, very much like the standard ROM check for the level 1 ROM areas. However, there are two differences between the two ROM checks.

First of all, while the level 1 ROM check is only executed *once* per driving cycle, the level 2 ROM check is performed cyclically, i.e., once it is finished, it starts again.

Secondly, the level 2 ROM check also takes into account all library functions (e.g., from the mathematics library) which might potentially called from within any level 2 function.

This can be achieved, e.g., by performing a *complete* check of those external libraries or by the use of so-called *inline functions* where only the code of those functions *which are actually called* is tested: the code of inline functions is directly copied to the place where the function is called, instead of performing a function call; concretely, the code of an inline function from the mathematics library called from a level 2 function thus becomes part of the level 2 code and is consequently tested by the level 2 code ROM check.

Note that the two mentioned ways to check external library functions were just chosen as examples: there might very well be other ways to include such libraries in the level 2 ROM check, it just has to be ensured that **each function called inside a level 2 function is included in the level 2 ROM check** (one way or another).

Description:

The function incorporates the following requirements:

G8.2 Cyclic ROM Check on MC for Process Monitoring

R8.2_1: The checksum shall be calculated by an addition checksum.

R8.2_2: Both the calculated checksum and its expected value shall be unsigned values obtained by adding unsigned values.

R8.2_3: The areas to be checked are comprised of the **level 2 code** area, the **level 2 constant data**, the **level 2 calibration data**, and the **code of all functions from external libraries contained in the level 2 code**.

R8.2_4: For the level 2 code area, the following special mechanism shall be used:
R8.2_4.1: there shall be two sections of level 2 code;
R8.2_4.2: for one of the sections, there shall be a calibration switch C_ERR_SWI_MC_LEVEL_2_COD which, **for application software only**, allows ignoring errors in the level 2 code area if its value is set to NC_ERR_SWI_MC_LEVEL_2_COD; **for production software, this switch must be ignored!**
R8.2_4.3: for the second section of the code area, C_ERR_SWI_MC_LEVEL_2_COD must not have any effect: a checksum error in this section shall always lead to a fault reaction;
R8.2_4.4: the ROM areas where C_ERR_SWI_MC_LEVEL_2_COD and the expected checksum for the code area are stored shall not be part of the level 2 ROM check.

R8.2_5: For the level 2 calibration area, the following special mechanism shall be used:
R8.2_5.1: there shall be a calibration switch C_ERR_SWI_MC_LEVEL_2_CAL which, **for application software only**, allows ignoring errors in the level 2 calibration ROM area if its value is set to NC_ERR_SWI_MC_LEVEL_2_CAL; **for production software, this switch must be ignored!**
R8.2_5.2: the ROM areas where C_ERR_SWI_MC_LEVEL_2_CAL and the expected checksum for the calibration data area are stored shall not be part of the level 2 ROM check.

R8.2_6: In case a level 2 ROM error is detected, i.e., in case the calculated checksum does not agree with the expected one:
R8.2_6.1: the redundant switch-off path shall be triggered by calling the action ACTION_ECM3_RedSwitchOffPath();

R8.2_6.2: an immediate fault reaction shall be performed by calling the action ACTION_ECM3_McFaultReaction() with the error code NC_ERR_COD_MC_ROM_LEVEL_2;
R8.2_6.3: the flag LV_MC_ROM_CHK_READY shall be reset (using the RAM check service actions).

R8.2_7: The level 2 ROM check shall run in a fixed time grid and shall be called every **10ms**; the meaning of "running in a fixed time grid" is that the level 2 ROM check is activated with a fixed recurrence; when activated, it continues where it left off when called the time before, i.e., this does not mean that it has to be finished between two calls but simply that it is continued when called for the next time.

R8.2_8: One level 2 ROM check cycle shall be finished:
R8.2_8.1: in **EMS** mode, within the time for temporary disabling of power stages (see "G22 Disabling of power stages");
R8.2_8.2: in **TCU** mode, within 500ms.

R8.2_9: The cyclic ROM check for process monitoring shall be under PFM control (viz., the 40ms PFM task with task number 5)

Application conditions:

Initialisation:	The checksum shall be set to a configurable initial value.
Recurrence:	The **initialisation** shall be performed **after each power-up or reset** and **after one cycle has been completed**. The **function itself** shall be calculated **every 10ms**.
Activation:	-
Deactivation:	-

12.4.4 RAM checks of processor monitoring on MC

FUNCTION DESCRIPTION:

General information:

Several RAM checks have to be performed on the MC to test the integrity of the RAM:

- there shall be a non-destructive RAM check for those data that have to be stored in a reset-safe way, and those RAM areas that are shared between BOOT and ECU software;
- for all remaining RAM areas, there shall be a destructive RAM check which also performs an initialisation to 0;
- additionally, there shall be a cyclic RAM check for the level 2 RAM areas and for some of the RAM areas of the processor monitoring: this is done by storing those variables as pair of value and complement and performing consistency checks when accessing the variables;
- in order to prevent faults due to interrupt handling while level 2 is active, also a stack check is implemented which checks the relevant memory area on the stack where the level 2 register content will be stored in case of an interrupt.

The first two kinds of RAM checks have to be performed in such a way as to be able to detect static errors like stuck bits/cells. The last two kinds of RAM check are used to detect dynamic errors like a bit toggling during runtime or one (process/processor monitoring) module corrupting the memory area of another one, or problems with the stack memory.

12.4.4.1 Standard RAM check on MC

FUNCTION DESCRIPTION:

General information:

Those variables which are to be stored in the reset-safe RAM area have to checked without destroying their contents.

Description:

The function incorporates the following requirements:

G6 Standard RAM check on MC (destructive/non-destructive)

R6_1: In case an error has been detected during the Standard RAM check, an immediate fault reaction shall be performed; however, the Standard RAM check is active at a time when calling ACTION_ECM3_McFaultReaction() with the error code NC_ERR_COD_MC_RAM_LEVEL_1 is not possible; therefore, the effect of the action shall be mimicked as much as possible (increase reset counter, set error code, trigger reset if reset counter has not yet passed a certain threshold etc.).

Application conditions:

Initialisation:	-
Recurrence:	Once before all other functions are started
Activation:	activated by the Standard RAM check
Deactivation:	-

12.4.4.2 RAM check service actions of processor monitoring on MC

FUNCTION DESCRIPTION:

General information:

In addition to the RAM checks described above which are only executed after power-up/reset, the processor monitoring on the MC also provides RAM check functionality which is used to protect data which is repeatedly used in both process and processor monitoring.

The functionality of the process monitoring (level 2) is calculated cyclically. The calculation and the storage of intermediate results of the process monitoring functions is performed using the registers of the processor as far as possible. The registers are tested by the function-specific instruction set test (see “MC function-specific instruction set test”).

Description:

The function, via the exported actions, incorporates the following requirements:

G7 RAM check service actions of processor monitoring on MC

R7_1: All variables of the process monitoring which are located in the RAM and used in the next recurrence of a task (e.g. anti-bounce-counters) or in a different successive task are stored as pairs consisting of the value and its complement.

R7_2: After execution of a level 2 module, its results have to be copied from registers into variables located in the RAM. After storing a value from an internal register in the RAM, together with its complement in a variable of the name var_CPL if the original variable was named var (example: LV_OFF_IV_MON_CPL contains the complement of LV_OFF_IV_MON), the value is read back from the RAM and compared with the original value of the register. This procedure is performed by the action ACTION_ECM3_WriteChkCpl() described below.

R7_3: At the beginning of the following recurrence of the task, the two complementary variables are checked for plausibility. Also intermediate results, which cannot be completely handled by the internal registers, are stored in value-complement pairs and checked for plausibility. This is done at the beginning of a level 2 module using the action ACTION_ECM3_ChkCpl() or ACTION_ECM3_ReadChkCpl(); the difference between these two actions is that the action ACTION_ECM3_ChkCpl() only performs a plausibility check for a variable and its complement variable without loading the actual value into a register, while ACTION_ECM3_ReadChkCpl() also loads a register with the value of the given variable.

R7_4: If a RAM fault is detected by these functions, an immediate fault reaction shall be performed via ACTION_ECM3_McFaultReaction(), using the error code NC_ERR_COD_MC_RAM_LEVEL_2.

R7_5: Additionally, there is also a variant of the level 2 RAM check service action ACTION_ECM3_ReadChkCpl() named ACTION_ECM3_McReadChkCpl() used for variables of processor monitoring; the difference to the level 2 service action is the presence of a default return value which shall be returned in case an inconsistency is detected (this is only meaningful for those variables which are stored in the reset-safe area, the others are initialised by the reset following the fault reaction).

R7_6: All RAM check service actions have to be implemented as macros to avoid function calls in the code of level 2 and processor monitoring.

R7_7: All RAM check service actions have to be implemented in such a way that they are universally usable, in particular regardless of the type of their parameters.

Description for ACTION_ECM3_WriteChkCpl

ACTION_ECM3_WriteChkCpl(PRM_VAR, PRM_VAR_CPL, PRM_VALUE)					
see description in section "RAM check service actions of Processor Monitoring on MC"					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_VAR	INOUT	*any*	*any*	*any*	*any*
Variable to be checked					
PRM_VAR_CPL	OUT	*any*		*any*	*any*
Variable for the complement of PRM_VAR					
PRM_VALUE	IN	*any*	*any*	*any*	*any*
Value to be stored					

Formula section:

```
PRM_VAR = PRM_VALUE;
PRM_VAR_CPL = complement(PRM_VALUE);
IF (PRM_VAR != PRM_VALUE)
THEN
    ACTION_ECM3_McFaultReaction(
            IN      NC_ERR_COD_MC_RAM_LEVEL_2);
FI
```

Description for ACTION_ECM3_ChkCpl

ACTION_ECM3_ChkCpl(PRM_VAR, PRM_VAR_CPL)					
see description in section "RAM check service actions of Processor Monitoring on MC"					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_VAR	IN	*any*	*any*	*any*	*any*
Variable to be checked					
PRM_VAR_CPL	IN	*any*		*any*	*any*
Variable for the complement of PRM_VAR					

Formula section:

```
IF (PRM_VAR != complement(PRM_VAR_CPL))
THEN
            ACTION_ECM3_McFaultReaction(
                    IN      NC_ERR_COD_MC_RAM_LEVEL_2);
FI
```

Description for ACTION_ECM3_ReadChkCpl

ACTION_ECM3_ReadChkCpl(PRM_RESULT, PRM_VAR, PRM_VAR_CPL)					
see description in section "RAM check service actions of Processor Monitoring on MC"					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_RESULT	OUT	*any*	*any*	*any*	*any*
Variable to be checked					
PRM_VAR	IN	*any*		*any*	*any*
Variable for the complement of PRM_VAR					
PRM_VAR_CPL	IN	*any*	*any*	*any*	*any*
Value to be stored					

Formula section

```
PRM_RESULT = PRM_VAR;
```

```
IF (PRM_VAR != complement(PRM_VAR_CPL))
THEN
            ACTION_ECM3_McFaultReaction(
                    IN      NC_ERR_COD_MC_RAM_LEVEL_2);
FI
```

Description for ACTION_ECM3_McReadChkCpl

ACTION_ECM3_McReadChkCpl(PRM_RESULT, PRM_VAR, PRM_VAR_CPL, PRM_DFT_VALUE)					
see description in section "RAM check service actions of Processor Monitoring on MC"					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_RESULT	OUT	*any*	*any*	*any*	*any*
Variable to be checked					
PRM_VAR	IN	*any*		*any*	*any*
Variable for the complement of PRM_VAR					
PRM_VAR_CPL	IN	*any*	*any*	*any*	*any*
Value to be stored					
PRM_DFT_VALUE	IN	*any*	*any*	*any*	*any*
Default value for PRM_VAR in case an inconsistency is detected					

Formula section:

```
IF (PRM_VAR != complement(PRM_VAR_CPL))
THEN
            PRM_RESULT = PRM_DFT_VALUE;
            ACTION_ECM3_McFaultReaction(
                    IN      NC_ERR_COD_MC_RAM_LEVEL_2);
ELSE
            PRM_RESULT = PRM_VAR;
FI
```

Application conditions:

Initialisation:	Before the first test of the complementary variables, all appropriate variables must be initialized properly using the action ACTION_ECM3_WriteChkCpl().
Recurrence:	All the actions in this function are called.
Activation:	-
Deactivation:	-

12.4.4.3 Stack check service action of processor monitoring on MC

FUNCTION DESCRIPTION:

General information:

In addition to the RAM checks described above (the start-up checks described in G6 and the cyclic level 2 RAM check described in G7), there is another memory test to ensure the integrity of memory areas used by the stack. The motivation for the function is as follows.

While a process monitoring (level 2) module is executed, an interrupt might occur. Prompted by the interrupt, the current content of the registers, plus "logistic" data like a return address, are stored on the stack. Once the interrupt handling is finished, the register content is restored from the stack and program execution continues where it left off before the interrupt (the address is also determined from the data saved on the stack).
However, if the used stack memory is corrupted, the level 2 register content might be corrupted without being detected; this could either lead to a dormant fault (level 2 might not catch an existing error) or reduce the availability of the vehicle (a level 2 fault reaction might get triggered unnecessarily).
Finally, if the return address is corrupted by being stored on the stack, the level 2 module might not be completed (leading to a problem with program flow monitoring (PFM), again reducing availability), or some code passages of the level 2 module might be omitted (again leading to a dormant fault).
Consequently, at the beginning of each level 2 module, the relevant stack areas have to be checked so that in case of an interrupt during the execution of the module, the necessary data can be correctly stored and recovered.

Description:

The function incorporates the following requirements:

G49 Stack check service action of processor monitoring on MC

R49_1: The interface to level 2 shall be provided via the exported action ACTION_ECM3_McChkStack().

Remark:

- the action shall be called at the very beginning of each level 2 module in order to ensure that the relevant stack area is OK in case the level 2 module is interrupted;
- the action shall be "sandwiched" in between two PFM actions in order to ensure that the stack check action is entered and left properly since its code might not be completely ROM-checked.

R49_2: When ACTION_ECM3_McChkStack() is called, the infrastructure action ACTION_INFR_McChkStack() shall be called in order to perform the stack check; the requirements to this action shall be described in the IRS of Processor Monitoring.

R49_3: ACTION_INFR_McChkStack() shall return a flag indicating whether the stack check was successful or not.

R49_4: If ACTION_INFR_McChkStack() indicates that the stack check was unsuccessful, an immediate fault reaction shall be performed via ACTION_ECM3_McFaultReaction(), using the error code NC_ERR_COD_MC_RAM_LEVEL_2.

R49_5: The call and the execution of ACTION_ECM3_McChkStack() shall use no stack.

Application conditions:

Initialisation: -

Recurrence: The function is called via ACTION_ECM3_McChkStack().

Activation: -

Deactivation: -

12.4.5 Configuration of MU

FUNCTION DESCRIPTION:

General information:

The MU can be configured in various ways, e.g., whether it is used for monitoring of EMS or TCU application (a choice which has implications on timing etc.). The act of configuring is one of the tasks of the MC which sends these data in the first communication to the MU (and re-sends them in the second so that the data can be rendered plausible on the MU).

Since the configuration has far-reaching influences on the behaviour of the MU and the whole system (e.g., by enabling or disabling the injection-off timer) and each configuration information is simply a bit information, the MU keeps on sending its configuration to the MC in NORMAL communications so that a change in the configuration of the MU can be detected as quickly as possible.

Description:

The function incorporates the following requirements:

G10 Configuration of MU by MC

R10_1: In the CONFIG communication, the MC sends configuration information to the MU; the MU can be configured with respect to:
- general mode (EMS or TCU, NLC_MU_MOD)
- hardware (via the dedicated input pin) or software (only via information received from the MC) acquisition of ignition key signal (NLC_MU_IGN_KEY_ACQ)
- power-off timer enabled or disabled (NLC_MU_POW_OFF_TMR_ENA)
- injection-off timer enabled or disabled (NLC_MU_POW_OFF_TMR_ENA)
- end-of-line test enabled or not (always disabled)

This is achieved by calling the action ACTION_INFR_McSendConfData() with the above arguments; note that the end-of-line test configuration is not part of the argument list since it is always disabled.

R10_2: In the PREDRIVE communication, the MC shall re-send these data in order to confirm the previously sent data (note that, in case a discrepancy between the two sets of configuration data is detected, an immediate fault reaction on the MU shall be the consequence); since the arguments have already been sent to the infrastructure, there is no need to re-send them.

R10_3: The MC shall obtain the complete set of configuration information (including the end-of-line test configuration) from the MU after the PREDRIVE communication so that the MC knows what the MU believes to be configured for; the information shall be stored in LF_MU_CONF_BYTE acquired by calling the action ACTION_INFR_McGetConfData(), and compared to the data from R10_1 and R10_2; note that LF_MU_CONF_BYTE is a compressed representation where the bit values for the configuration data have to be extracted from.

R10_4: The MC shall obtain the configuration information (excluding the end-of-line test configuration) from the MU after NORMAL communications; the information shall again be stored in LF_MU_CONF_BYTE acquired by calling the action ACTION_INFR_McGetConfData(), and compared to the data from R10_1 and R10_2.

R10_5: In case a configuration mismatch is detected on the MC:
R10_5.1: in PREDRIVE, there shall be an immediate fault reaction by calling ACTION_ECM3_McFaultReaction() with error code NC_ERR_COD_MC_CONF_DIF;
R10_5.2: in NORMAL, there shall be a debounced fault reaction, using the anti-bounce counter NC_ABC_MC_CONF_DIF, the error code NC_ERR_COD_MC_CONF_DIF, and the two actions ACTION_ECM3_McIncAbc() and ACTION_ECM3_McDecAbc().

Application conditions:

Initialisation:	LF_MU_CONF_BYTE = 0;
Recurrence:	The **initialisation** shall be performed **after each power-up or reset**. R10_1 shall be executed before the CONFIG communication is triggered; R10_2 does not require any activity; R10_3 shall be executed after the PREDRIVE communication and the corresponding communication monitoring function; R10_4 shall be executed after NORMAL communications and the corresponding communication monitoring function; R10_5 shall be executed after R10_3 and R10_4, as appropriate.
Activation:	For R10_3 and R10_4, LV_MC_COM_ERR = 0 must hold *(they are only executed if no communication error was detected; note that a consistency check with LV_MC_COM_ERR_CPL is necessary)*
Deactivation:	Otherwise

12.4.6 Version compatibility check

FUNCTION DESCRIPTION:

General information:

This functions describes when and how compatibility information is exchanged between MC and MU.

Description:

This function incorporates the following requirements:

G11 Version compatibility check

R11_1: After the CONFIG communication, if communication monitoring has not found an error, i.e., if LV_MC_COM_ERR = 0 (note that a consistency check with LV_MC_COM_ERR_CPL is necessary), the MC shall obtain the compatibility information sent by the MU (NR_COMP_MU; the hardware version NR_MU_HW_VERS is only sent for information purposes) by calling the actions ACTION_INFR_McGetCompNr() and ACTION_INFR_McGetVersNr().

R11_2: If the compatibility information from the MU does not match the information on the MC (NC_NR_MC_COMP), an immediate fault reaction shall be triggered by calling ACTION_ECM3_McFaultReaction with error code NC_ERR_COD_MC_COMP_NR.

R11_3: In the PREDRIVE communication, the MC shall send compatibility information the MU (NC_NR_MC_COMP) by calling the action ACTION_INFR_McSendCompNr() before the PREDRIVE communication is triggered.

Application conditions:

Initialisation:	NR_MU_COMP = 0; NR_MU_HW_VERS = 0;
Recurrence:	The **initialisation** shall be performed **after each power-up or reset**. R1 shall be executed after the CONFIG communication is completed and no communication error was detected; R2 shall be executed after R1, if necessary; R3 shall be executed before the PREDRIVE communication is triggered.
Activation:	For R1, LV_MC_COM_ERR = 0 must hold *(only executed if no communication error was no detected; note that a consistency check with LV_MC_COM_ERR_CPL is necessary)*
Deactivation:	Otherwise

12.4.7 Acquisition of values of A/D conversions

FUNCTION DESCRIPTION:

General information:

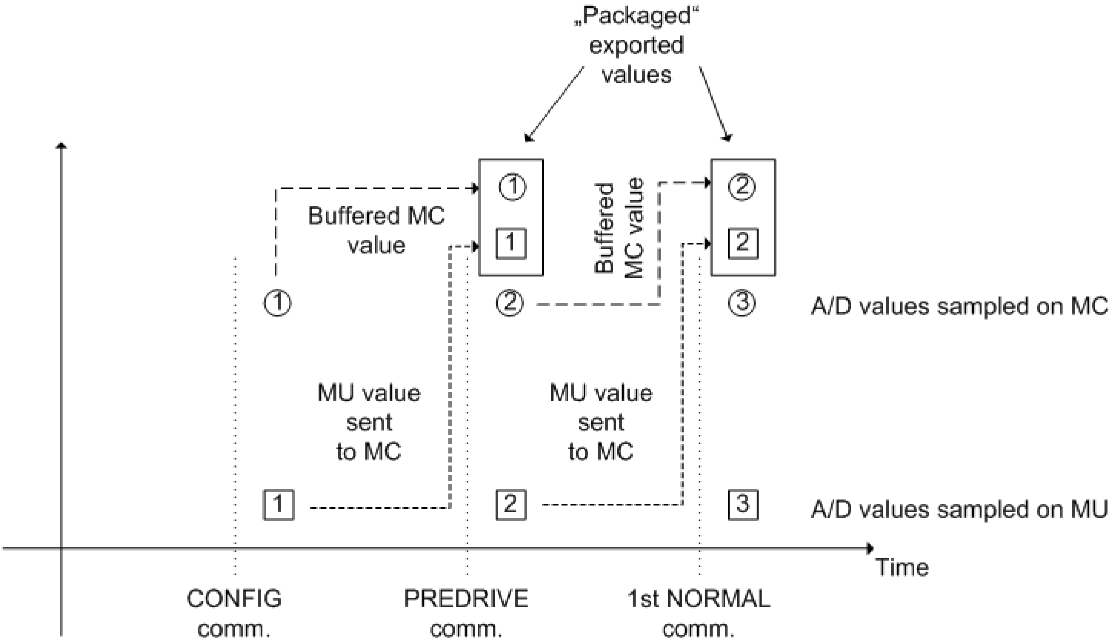
This functions describes when and how the result of the A/D conversions (both on MC and MU) shall be acquired and how it is exported to process monitoring.

A/D conversions are performed on both MC and MU, in both cases after a communication, starting with the end of the CONFIG communication.

However, since the value acquired on the MU can only be transmitted to the MC in the next communication, the data have to be synchronised: for example, the value acquired on the MU after the CONFIG communication is only transmitted to the MC in the PREDRIVE communication and hence only available on the MC after the end of the PREDRIVE communication; in the meantime, the MC also has acquired A/D values after the CONFIG and PREDRIVE communications; so it has to be ensured that the value from the MU sampled after the CONFIG communication is exported ("packaged") together with the value sampled on the MC after the CONFIG communication.

This can easily be achieved by simply buffering the value sampled on the MC until after the next communication (see signal flow diagram).

Signal flow diagram:



Description:

This function incorporates the following requirements:

G23 Acquisition of values of A/D conversions

R23_1: After each communication:
R23_1.1: the buffered A/D value VP_MC_AN_DIG_MON_TMP shall be copied into VP_MC_AN_DIG_MON;
R23_1.2: VP_MC_AN_DIG_MON_TMP and VP_MC_AN_DIG_MON shall be stored together with their complements VP_MC_AN_DIG_MON_TMP_CPL and VP_MC_AN_DIG_MON_CPL and only be accessed using the actions ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() from section "RAM check service actions of processor monitoring on MC";
R23_1.3: their default values in case of a RAM error shall be 0.

R23_2: After storing the buffered value into the actual interface, the new value of the A/D conversion on the MC shall be fetched from the infrastructure by calling the action ACTION_INFR_GetAnDigMonValMc(), storing the result in VP_MC_AN_DIG_MON_TMP.

R23_3: If VP_MC_AN_DIG_MON_TMP is equal to NC_VP_MC_AN_DIG_MON_CPL_ERR:
R23_3.1: VP_MC_AN_DIG_MON_TMP shall be reset to 0 (using the RAM check actions);
R23_3.2: an immediate fault reaction shall be performed via ACTION_ECM3_McFaultReaction(), using the error code NC_ERR_COD_MC_RAM_LEVEL_2;
R23_3.3: otherwise, no action shall be performed.

R23_4: After each successful PREDRIVE, NORMAL, and DISABLE communication (i.e., LV_MC_ERR_COM = 0):

R23_4.1: the freshly transmitted value of the A/D conversion on the MU shall be fetched from the infrastructure via the action ACTION_INFR_GetAnDigMonValMu(), storing the result in VP_MU_AN_DIG_MON;
R23_4.2: VP_MU_AN_DIG_MON shall be stored together with its complement VP_MU_AN_DIG_MON_CPL and only be accessed using the actions ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() from section "RAM check service actions of processor monitoring on MC";
R23_4.3: its default value in case of a RAM error shall be 0.

Note: the MU value VP_MU_AN_DIG_MON is only updated after a *successful* communication while the MC value is acquired after *each* communication; in the extreme case of no successful communications, the MU value will therefore freeze to the value obtained after the last correct communication, eventually leading to a fault reaction (of process monitoring, where the MC and MU values are compared); however, unsuccessful communications will (much faster) lead to a fault reaction of processor monitoring (by the communication monitoring functions).

Application conditions:

Initialisation: */* Using the RAM check service actions */*

VP_MC_AN_DIG_MON = 0;
VP_MU_AN_DIG_MON = 0;
VP_MC_AN_DIG_MON_TMP = 0;

Recurrence: The **initialisation** shall be performed **after each power-up or reset**.

The **function itself** shall be executed **after the communication monitoring functions**.

Activation: For R23_1, R23_2, R23_3:

Always.

For R23_4:

((STATE_MC = NC_STATE_MC_PRDR) **OR**
(STATE_MC = NC_STATE_MC_NORM) **OR**
(STATE_MC = NC_STATE_MC_DI)) **AND**
(LV_MC_COM_ERR = 0)
(active in PREDRIVE, NORMAL, and DISABLE state but only if the preceding communication was successful; note that a consistency check with LV_MC_COM_ERR_CPL is necessary)

Deactivation: Otherwise

12.4.8 Exchange of ignition key information between MC and MU

FUNCTION DESCRIPTION:

General information:

This function describes when and how information regarding the state of the ignition key is exchanged between MC and MU.

Description:

This function incorporates the following requirements:

G25 Exchange of ignition key information between MC and MU

R25_1: In each NORMAL and DISABLE communication, the MC sends the value of LV_MC_IGN_KEY to the MU; this is achieved by calling ACTION_INFR_McSendIgk() with argument LV_MC_IGN_KEY before the corresponding communications are triggered.

R25_2: If the MU is configured for hardware acquisition of the ignition key signal, after each PREDRIVE, NORMAL, and DISABLE communication, the value of the ignition key input IGK_IN on the MU shall be acquired by calling the action ACTION_INFR_McGetIgkMu(), storing the result in LV_MU_IGN_KEY, but only if no communication error was detected, i.e., LV_MC_COM_ERR = 0 (note that a consistency check with LV_MC_COM_ERR_CPL is necessary).

Application conditions:

Initialisation:	LV_MU_IGN_KEY = 0;
Recurrence:	The **initialisation** shall be performed **after each reset or power-up**. The **function** shall be **executed after the communication monitoring functions**.
Activation:	((STATE_MC = NC_STATE_MC_PRDR) **OR** (STATE_MC = NC_STATE_MC_NORM) **OR** (STATE_MC = NC_STATE_MC_DI)) *(only active in PREDRIVE, NORMAL, and DISABLE state)*
Deactivation:	Otherwise

12.5 MC Fault Reaction

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_RST_MC	V/O/S	0...FH	0...15	1	[-]
Reset counter on MC					
CTR_RST_MC_CPL	O/S	0...FH	0...15	1	[-]
Complement of CTR_RST_MC					
ERR_COD_MC	V/O/S	0...1FH	0...31	1	[-]
Error code of the MC					
STATE_ERR_IDC_MC	V/O/S	0...3H	0...3	1	[-]
Error indication on MC					
ABC_MC_CKS	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for checksum monitoring					
ABC_MC_HD	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for header-byte monitoring					
ABC_MC_CONF_DIF	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for receiving wrong configuration data from MU in NORMAL state					
ABC_MC_WRG_IDX	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not sending expected FS-IST test index (question)					
ABC_MC_WRG_RESP	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not reacting correctly to wrong FS-IST answers					
ABC_MC_NOT_DEC	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not decreasing the anti-bounce counter for FS-IST answers correctly					
ABC_MC_PFM_0_TOG	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not toggling the PFM bit of task 0 in time					
ABC_MC_PFM_1_TOG	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not toggling the PFM bit of task 1 in time					
ABC_MC_PFM_2_TOG	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not toggling the PFM bit of task 2 in time					
ABC_MC_PFM_3_TOG	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not toggling the PFM bit of task 3 in time					
ABC_MC_PFM_4_TOG	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not toggling the PFM bit of task 4 in time					
ABC_MC_PFM_5_TOG	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not toggling the PFM bit of task 5 in time					
ABC_MC_PFM_6_TOG	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not toggling the PFM bit of task 6 in time					
ABC_MC_PFM_0_RESP	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not reacting to wrongly transmitted PFM bit of task 0					
ABC_MC_PFM_1_RESP	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not reacting to wrongly transmitted PFM bit of task 1					
ABC_MC_PFM_2_RESP	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not reacting to wrongly transmitted PFM bit of task 2					
ABC_MC_PFM_3_RESP	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not reacting to wrongly transmitted PFM bit of task 3					
ABC_MC_PFM_4_RESP	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not reacting to wrongly transmitted PFM bit of task 4					
ABC_MC_PFM_5_RESP	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not reacting to wrongly transmitted PFM bit of task 5					
ABC_MC_PFM_6_RESP	- /S	0...FH	0...15	1	[-]
Local variable: anti-bounce counter for MU not reacting to wrongly transmitted PFM bit of task 6					
LV_MC_DR_OFF	V/O	0...1H	0...1	1	[-]
Flag indicating whether the redundant switch-off path was triggered					
LV_MC_DR_OFF_CPL	O	0...1H	0...1	1	[-]
Complement of LV_MC_DR_OFF					
CTR_COM_CYC_MC_DEC_RST	-	0...3FFH	0...1023	1	[-]
Local variable: communication cycle counter for healing of reset counter and error information					
LV_ERR_MU_MC	V/O	0...1H	0...1	1	[-]

Error flag requesting and indicating permanent disabling of power stages by Processor Monitoring (MU and MC)					
LV_ERR_MU_MC_CPL	O	0...1H	0...1	1	[-]
Complement of LV_ERR_MU_MC					
LV_ERR_TMP_MU_MC	V/O	0...1H	0...1	1	[-]
Error flag requesting and indicating temporary disabling of power stages by Processor Monitoring (MU and MC)					
LV_ERR_TMP_MU_MC_CPL	O	0...1H	0...1	1	[-]
Complement of LV_ERR_TMP_MU_MC					
LV_MU_DI_OUT_0	V/O	0...1H	0...1	1	[-]
Flag indicating (in inverse logic!) whether the MU disables power stage 0 or not					
LV_MU_DI_OUT_1	V/O	0...1H	0...1	1	[-]
Flag indicating (in inverse logic!) whether the MU disables power stage 1 or not					
CTR_MU_DI	-	0...FH	0...15	1	[-]
Local variable: counter for time-out of disable request (counts number of communications since the DISABLE header was sent to the MU)					
LV_MU_DI_ACT	-	0...1H	0...1	1	[-]
Local variable: flag indicating whether the DISABLE request is active (prompting a manipulated value for the reset counter to be sent to MU)					
LV_V_L_DET_MU	V/O	0...1H	0...1	1	[-]
Flag indicating that under-voltage was detected by the MU					
LV_V_H_DET_MU	V/O	0...1H	0...1	1	[-]
Flag indicating that over-voltage was detected by the MU					
CTR_RST_MU	V/O/S	0...FH	0...15	1	[-]
Value of reset counter of MU					
ERR_COD_MU	V/O/S	0...1FH	0...31	1	[-]
Error code of the MU					
STATE_ERR_IDC_MU	V/O/S	0...3H	0...3	1	[-]
Error indication of MU					
CTR_COM_CYC_MC_DR_DI_TMP	V	0...FFH	0...255	1	[-]
Communication cycle counter used for temporary disabling of power stages by Processor Monitoring					
LV_MC_INH_SOPC	V/O	0...1H	0...1	1	[-]
Flag indicating that the MC-part of the SOPC has to be delayed since the MU is still disabling					
MU_DI_STATE	V/O	0H 1H 2H 3H	MU_DI_NOT_REQ MU_DI_BUSY MU_DI_ERR MU_DI_OK	1	[-]
State of MU Disable request					
LV_ABC_MC_ERR_COM	V/O	0...1H	0...1	1	[-]
Flag indicating that an anti-bounce counter for communication monitoring is greater than 0 after reset					
LV_ABC_MC_ERR_PFM	V/O	0...1H	0...1	1	[-]
Flag indicating that an anti-bounce counter for PFM tasks 0...5 (not for task 6!) is greater than 0 after reset					
LV_ABC_MC_ERR_FCT_SPC_IST	V/O	0...1H	0...1	1	[-]
Flag indicating that an anti-bounce counter for FS-IST is greater than 0 after reset					
LV_ABC_MC_ERR_PFM_6	V/O	0...1H	0...1	1	[-]
Flag indicating that an anti-bounce counter for PFM task 6 is greater than 0 after reset					
LV_INJ_OFF_TMR_INJ_ENA_TMP	O/V	0...1H	0...1	1	[-]
Temporary flag indicating enabling of injection by the injection off timer of MU					

Note: data with mode "S" shall only be stored in a reset-safe manner, not in NVMY!

Input data:

STATE_MC	NC_STATE_MC_INI	NC_STATE_MC_CONF	NC_STATE_MC_PRDR
NC_STATE_MC_NORM	NC_STATE_MC_DI	LV_MC_ROM_CHK_READY	LV_MC_ROM_CHK_READY_CPL
LV_MC_SOPC_ACT	LV_MC_COM_ERR	LV_MC_COM_ERR_CPL	LV_MC_PFM_WRG_RESP_ACT_6
NLC_MU_INJ_OFF_TMR_ENA	LV_MU_IGN_KEY		

Export actions:

ACTION_ECM3_McFaultReaction(IN <err_cod>)
This action immediately performs the central fault reaction storing the given error code.
ACTION_ECM3_McIncAbc(IN <abc_id>, IN <err_cod>)
This action increases the value of the indicated anti-bounce counter by its corresponding increment; if the maximal value is reached, a fault reaction is triggered with the given error code as parameter.
ACTION_ECM3_McDecAbc(IN <abc_id>)
This action decreases the value of the indicated anti-bounce counter by 1.
ACTION_ECM3_RedSwitchOffPath()
This action sets the flag LV_MC_DR_OFF and, in doing so, activates the redundant switch-off path.
ACTION_ECM3_DisableReq(OUT<flag>)
This action changes the state of processor monitoring on both MC and MU into DISABLE state without triggering a restart of the ECU (exported to ASW); the flag returns "false" in case the request has not been completed (with or without error).
ACTION_ECM3_ShutDown(OUT <flag>)
This action changes the state of processor monitoring on both MC and MU into DISABLE state, eventually triggering a restart of the ECU (exported to ASW) ; the flag returns "false" in case the request has not been completed (with or without error).

Import actions:

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)
ACTION_ECM3_McReadChkCpl(OUT <>, IN <>, IN <>, IN <>)
ACTION_ECM3_McChangeState(IN <>)
ACTION_ECM3_McSetErrmUpdateFlag(IN <>)
ACTION_ECM3_McCancelSopc()
ACTION_ECM3_McResetRomChkFlags()
ACTION_ECM3_McReadChkState(OUT <>)
ACTION_ECM3_McUnlockPwlResource()
ACTION_INFR_McRestart()
ACTION_INFR_McGetHd(OUT <>)
ACTION_INFR_McGetErrByte(OUT <>)
ACTION_INFR_McGetRstCtr(OUT <>)
ACTION_INFR_McGetNdis0(OUT <>)
ACTION_INFR_McGetNdis1(OUT <>)
ACTION_INFR_McGetUvdFlag(OUT <>)
ACTION_INFR_McGetOvdFlag(OUT <>)
ACTION_INFR_McSendErrInfo(IN <>, IN <>, IN <>)
ACTION_ECM2_ResetErrorFlags()
ACTION_ECM2_LockPws()

FUNCTION DESCRIPTION:

General information:

This module contains several functions related to the fault reaction of processor monitoring on the MC.
Note that several functions might change the value of several of the output variables (e.g., the value of the reset counter of the MC, CTR_RST_MC, is changed by both the central fault reaction and the healing of reset counter and error information) but that always, at any given time, only **one** of the sub-functions is active so the variables simply have the value of the last update that they received (thereby acting as a kind of “merge” block).

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CTR_RST_MC_THD	1	0...FH	0...15	1	[-]
Threshold for reset counter on MU, RST_CTR_MC, the reaching of which shall prompt the MU to irreversibly (for the current driving cycle) disable the power stages					
NC_ERR_COD_MC_NOT_ERR	1	0...1FH	0...31	1	[-]
Error code for "no error"					
NC_ERR_COD_MC_HD	1	0...1FH	0...31	1	[-]
Error code for "header byte invalid or disallowed transition					
NC_ERR_COD_MC_CKS	1	0...1FH	0...31	1	[-]
Error code for "checksum of received data is wrong"					
NC_ERR_COD_MC_PAR	1	0...1FH	0...31	1	[-]
Error code for "parity error in received data"					
NC_ERR_COD_MC_CONF_DIF	1	0...1FH	0...31	1	[-]
Error code for "configuration data different from previously sent data"					
NC_ERR_COD_MC_SOPC	1	0...1FH	0...31	1	[-]
Error code for "error in switch-off path check"					
NC_ERR_COD_MC_COMP_NR	1	0...1FH	0...31	1	[-]
Error code for "MU is incompatible to MC software"					
NC_ERR_COD_MC_ROM_LEVEL_1	1	0...1FH	0...31	1	[-]
Error code for "error in standard ROM check (for level 1)"					
NC_ERR_COD_MC_RAM_LEVEL_1	1	0...1FH	0...31	1	[-]
Error code for "error in standard RAM check (for level 1)"					
NC_ERR_COD_MC_ROM_LEVEL_2	1	0...1FH	0...31	1	[-]
Error code for "error in cyclic ROM check for level 2"					
NC_ERR_COD_MC_RAM_LEVEL_2	1	0...1FH	0...31	1	[-]
Error code for "error in cyclic RAM check for level 2"					
NC_ERR_COD_MC_WRG_IDX	1	0...1FH	0...31	1	[-]
Error code for "MU does not send expected test index (question)"					
NC_ERR_COD_MC_WRG_RESP	1	0...1FH	0...31	1	[-]
Error code for "MU does not react to wrong FS-IST answers"					
NC_ERR_COD_MC_NOT_DEC	1	0...1FH	0...31	1	[-]
Error code for "MU does not decrease the anti-bounce counter for FS-IST correctly"					
NC_ERR_COD_MC_PFM_0_TOG	1	0...1FH	0...31	1	[-]
Error code for "MU does not toggle bit 0 in the PFM synchronisation byte with the correct recurrence"					
NC_ERR_COD_MC_PFM_1_TOG	1	0...1FH	0...31	1	[-]
Error code for "MU does not toggle bit 1 in the PFM synchronisation byte with the correct recurrence"					
NC_ERR_COD_MC_PFM_2_TOG	1	0...1FH	0...31	1	[-]
Error code for "MU does not toggle bit 2 in the PFM synchronisation byte with the correct recurrence"					
NC_ERR_COD_MC_PFM_3_TOG	1	0...1FH	0...31	1	[-]
Error code for "MU does not toggle bit 3 in the PFM synchronisation byte with the correct recurrence"					
NC_ERR_COD_MC_PFM_4_TOG	1	0...1FH	0...31	1	[-]
Error code for "MU does not toggle bit 4 in the PFM synchronisation byte with the correct recurrence"					
NC_ERR_COD_MC_PFM_5_TOG	1	0...1FH	0...31	1	[-]
Error code for "MU does not toggle bit 5 in the PFM synchronisation byte with the correct recurrence"					
NC_ERR_COD_MC_PFM_6_TOG	1	0...1FH	0...31	1	[-]
Error code for "MU does not toggle bit 6 in the PFM synchronisation byte with the correct recurrence"					
NC_ERR_COD_MC_NOT_VLD_TRAN	1	0...1FH	0...31	1	[-]
Error code for "invalid or more than one state active (invalid transition) on MC"					
NC_ERR_COD_MC_PFM_0_RESP	1	0...1FH	0...31	1	[-]
Error code for "MU does not respond to wrongly toggled bit 0 in PFM response"					
NC_ERR_COD_MC_PFM_1_RESP	1	0...1FH	0...31	1	[-]
Error code for "MU does not respond to wrongly toggled bit 1 in PFM response"					
NC_ERR_COD_MC_PFM_2_RESP	1	0...1FH	0...31	1	[-]
Error code for "MU does not respond to wrongly toggled bit 2 in PFM response"					
NC_ERR_COD_MC_PFM_3_RESP	1	0...1FH	0...31	1	[-]
Error code for "MU does not respond to wrongly toggled bit 3 in PFM response"					
NC_ERR_COD_MC_PFM_4_RESP	1	0...1FH	0...31	1	[-]

Error code for "MU does not respond to wrongly toggled bit 4 in PFM response"					
NC_ERR_COD_MC_PFM_5_RESP	1	0...1FH	0...31	1	[-]
Error code for "MU does not respond to wrongly toggled bit 5 in PFM response"					
NC_ERR_COD_MC_PFM_6_RESP	1	0...1FH	0...31	1	[-]
Error code for "MU does not respond to wrongly toggled bit 6 in PFM response"					
NC_STATE_ERR_IDC_MC_NOT_PRES	1	0...3H	0...3	1	[-]
Error indication "no error present"					
NC_STATE_ERR_IDC_MC_REST_REQ	1	0...3H	0...3	1	[-]
Error indication "ECU restart request"					
NC_STATE_ERR_IDC_MC_DI_REQ	1	0...3H	0...3	1	[-]
Error indication "DISABLE request"					
NC_STATE_ERR_IDC_MC_PREV_REST	1	0...3H	0...3	1	[-]
Error indication "ECU restart was requested"					
NC_ABC_MC_HD	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to header-byte monitoring					
NC_ABC_MC_CKS	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to checksum monitoring					
NC_ABC_MC_CONF_DIF	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to receiving different configuration data from MU					
NC_ABC_MC_WRG_IDX	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not sending the expected FS-IST test index (question)					
NC_ABC_MC_WRG_RESP	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter for MU not reacting correctly to wrong FS-IST answers					
NC_ABC_MC_NOT_DEC	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter for MU not decreasing the anti-bounce counter for FS-IST answers correctly					
NC_ABC_MC_PFM_0_TOG	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not toggling the PFM bit of task 0 in time					
NC_ABC_MC_PFM_1_TOG	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not toggling the PFM bit of task 1 in time					
NC_ABC_MC_PFM_2_TOG	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not toggling the PFM bit of task 2 in time					
NC_ABC_MC_PFM_3_TOG	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not toggling the PFM bit of task 3 in time					
NC_ABC_MC_PFM_4_TOG	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not toggling the PFM bit of task 4 in time					
NC_ABC_MC_PFM_5_TOG	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not toggling the PFM bit of task 5 in time					
NC_ABC_MC_PFM_6_TOG	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not toggling the PFM bit of task 6 in time					
NC_ABC_MC_PFM_0_RESP	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 0					
NC_ABC_MC_PFM_1_RESP	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 1					
NC_ABC_MC_PFM_2_RESP	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 2					
NC_ABC_MC_PFM_3_RESP	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 3					
NC_ABC_MC_PFM_4_RESP	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 4					
NC_ABC_MC_PFM_5_RESP	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 5					
NC_ABC_MC_PFM_6_RESP	1	0...1FH	0...31	1	[-]
Indicator for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 6					
NC_ABC_MC_HD_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to header-byte monitoring					
NC_ABC_MC_CKS_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to checksum monitoring					
NC_ABC_MC_CONF_DIF_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to receiving different configuration data from MU					
NC_ABC_MC_WRG_IDX_INI	1	0...FH	0...15	1	[-]

Initial value for anti-bounce counter corresponding to MU not sending the expected FS-IST test index (question)					
NC_ABC_MC_WRG_RESP_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter for MU not reacting correctly to wrong FS-IST answers					
NC_ABC_MC_NOT_DEC_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter for MU not decreasing the anti-bounce counter for FS-IST answers correctly					
NC_ABC_MC_PFM_0_TOG_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 0 in time					
NC_ABC_MC_PFM_1_TOG_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 1 in time					
NC_ABC_MC_PFM_2_TOG_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 2 in time					
NC_ABC_MC_PFM_3_TOG_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 3 in time					
NC_ABC_MC_PFM_4_TOG_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 4 in time					
NC_ABC_MC_PFM_5_TOG_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 5 in time					
NC_ABC_MC_PFM_6_TOG_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 6 in time					
NC_ABC_MC_PFM_0_RESP_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 0					
NC_ABC_MC_PFM_1_RESP_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 1					
NC_ABC_MC_PFM_2_RESP_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 2					
NC_ABC_MC_PFM_3_RESP_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 3					
NC_ABC_MC_PFM_4_RESP_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 4					
NC_ABC_MC_PFM_5_RESP_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 5					
NC_ABC_MC_PFM_6_RESP_INI	1	0...FH	0...15	1	[-]
Initial value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 6					
NC_ABC_MC_HD_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to header-byte monitoring					
NC_ABC_MC_CKS_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to checksum monitoring					
NC_ABC_MC_CONF_DIF_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to receiving different configuration data from MU					
NC_ABC_MC_WRG_IDX_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not sending the expected FS-IST test index (question)					
NC_ABC_MC_WRG_RESP_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter for MU not reacting correctly to wrong FS-IST answers					
NC_ABC_MC_NOT_DEC_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter for MU not decreasing the anti-bounce counter for FS-IST answers correctly					
NC_ABC_MC_PFM_0_TOG_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not toggling the PFM bit of task 0 in time					
NC_ABC_MC_PFM_1_TOG_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not toggling the PFM bit of task 1 in time					
NC_ABC_MC_PFM_2_TOG_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not toggling the PFM bit of task 2 in time					
NC_ABC_MC_PFM_3_TOG_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not toggling the PFM bit of task 3 in time					
NC_ABC_MC_PFM_4_TOG_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not toggling the PFM bit of task 4 in time					
NC_ABC_MC_PFM_5_TOG_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not toggling the PFM bit of task 5 in time					
NC_ABC_MC_PFM_6_TOG_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not toggling the PFM bit of task 6 in time					
NC_ABC_MC_PFM_0_RESP_INC	1	0...FH	0...15	1	[-]

Increment for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 0					
NC_ABC_MC_PFM_1_RESP_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 1					
NC_ABC_MC_PFM_2_RESP_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 2					
NC_ABC_MC_PFM_3_RESP_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 3					
NC_ABC_MC_PFM_4_RESP_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 4					
NC_ABC_MC_PFM_5_RESP_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 5					
NC_ABC_MC_PFM_6_RESP_INC	1	0...FH	0...15	1	[-]
Increment for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 6					
NC_ABC_MC_HD_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to header-byte monitoring					
NC_ABC_MC_CKS_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to checksum monitoring					
NC_ABC_MC_CONF_DIF_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to receiving different configuration data from MU					
NC_ABC_MC_WRG_IDX_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not sending the expected FS-IST test index (question)					
NC_ABC_MC_WRG_RESP_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter for MU not reacting correctly to wrong FS-IST answers					
NC_ABC_MC_NOT_DEC_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter for MU not decreasing the anti-bounce counter for FS-IST answers correctly					
NC_ABC_MC_PFM_0_TOG_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 0 in time					
NC_ABC_MC_PFM_1_TOG_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 1 in time					
NC_ABC_MC_PFM_2_TOG_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 2 in time					
NC_ABC_MC_PFM_3_TOG_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 3 in time					
NC_ABC_MC_PFM_4_TOG_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 4 in time					
NC_ABC_MC_PFM_5_TOG_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 5 in time					
NC_ABC_MC_PFM_6_TOG_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not toggling the PFM bit of task 6 in time					
NC_ABC_MC_PFM_0_RESP_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 0					
NC_ABC_MC_PFM_1_RESP_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 1					
NC_ABC_MC_PFM_2_RESP_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 2					
NC_ABC_MC_PFM_3_RESP_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 3					
NC_ABC_MC_PFM_4_RESP_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 4					
NC_ABC_MC_PFM_5_RESP_MAX	1	0...FH	0...15	1	[-]
Maximal value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 5					
NC_ABC_MC_PFM_6_RESP_MAX	1	0...FH	0...15	1	[-]

Maximal value for anti-bounce counter corresponding to MU not reacting to wrongly transmitted PFM bit of task 6					
NC_CTR_COM_CYC_MC_DEC_RST_INI	1	0...3FFH	0...1023	1	[-]
Initial value for communication cycle counter for healing of reset counter and error information					
NC_CTR_MU_DI_MAX	1	0...FH	0...15	1	[-]
Maximal value for counter for disable request					
NC_CTR_RST_MC_DI	1	0...FH	0...15	1	[-]
Value of reset counter to be sent to MU in case of a disable request					
NC_CYCNR_MC_DR_DI_TMP	1	1...FFH	1...255	1	[-]
Number of communication cycles (<>0) for temporary disabling after the PREDRIVE communication					
NC_ERR_COD_MC_MU_READY	1	0...1FH	0...31	1	[-]
Error code for "MU not ready"					
NC_ERR_COD_MU_V_L	1	0...1FH	0...31	1	[-]
Error code for "MU has detected under-voltage"					
NC_ERR_COD_MU_V_H	1	0...1FH	0...31	1	[-]
Error code for "MU has detected over-voltage"					
NC_ERR_COD_MU_V_L_H	1	0...1FH	0...31	1	[-]
Error code for "MU has detected both under- and over-voltage"					
NC_ERR_COD_MU_FCT_SPC_IST	1	0...1FH	0...31	1	[-]
Error code for "MU has detected a wrong FS-IST answer"					

Note: The last four error codes are MU error codes and are only used in connection with error memory and the under-/over-voltage flags sent from the MU (see section "Acquisition of under-/over-voltage flags").

12.5.1 Central fault reaction

FUNCTION DESCRIPTION:

General information:

This function contains the central fault reaction of processor monitoring on the MC. Since this function is not activated with a fixed recurrence but only called when an error has actually occurred, the complete functionality is encapsulated in the action ACTION_ECM3_McFaultReaction() which can be imported and called by other functions, should they require to trigger an immediate fault reaction (debounced fault reactions are handled via G21 "Debounce mechanism").

In order to avoid recursive calls of the fault reaction:

- **no monitoring of state machine shall be performed**
- **no complement checks shall be performed, only the complements shall be written**
- **for all calls of external functions (e.g. for disabling power stages), it has to be ensured that they cannot lead to recursive calls of the fault reaction**

Description:

The function, via the action ACTION_ECM3_McFaultReaction(), incorporates the following requirements:

G1 Central fault reaction

R1_1: Concerning the activation of the fault reaction:

- **a fault reaction shall only be performed in a state different from DISABLE (i.e., STATE_MC <> NC_STATE_MC_DI);**
- **if STATE_MC = NC_STATE_MC_DI when the fault reaction is triggered, nothing shall be done;**

- **therefore, the following requirements only refer to the case that the fault reaction is actually performed (i.e., STATE_MC <> NC_STATE_MC_DI).**

R1_2: The action ACTION_ECM3_McCancelSopc() shall be called in order to ensure that the power stages can be disabled.

R1_3: The power stages shall be disabled, see G22 "Disabling of power stages"; **this shall only be done if the fault reaction was activated in a state different from DISABLE (i.e., STATE_MC <> NC_STATE_MC_DI) to avoid recursive calls to the fault reaction (via ACTION_ECM2_LockPws())**.

R1_4: A transition into the DISABLE state shall be performed, using ACTION_ECM3_McChangeState(); this shall be done before the power stages are disabled (due to R22_8, the power stages will anyway be disabled in DISABLE, even if there is a problem during the execution of the fault reaction).

R1_5: If the reset counter CTR_RST_MC on the MC is strictly less than the threshold NC_CTR_RST_MC_THD:
R1_5.1: the reset counter shall be incremented by 1; this value shall be stored with complement CTR_RST_MC_CPL in a reset-safe area; the default value for CTR_RST_MC shall be NC_CTR_RST_MC_THD + 1 (in case of data corruption, assume permanent disabling).
R1_5.2: an MC error code which corresponds to the error location shall be stored, together with the indicator ”ECU restart request”; this value shall be stored in a reset-safe area.
R1_5.3: There shall be an attempt to restart the ECU, using the action ACTION_INFR_McRestart().

R1_6: If the reset counter is greater than or equal to the threshold NC_CTR_RST_MC_THD:
R1_6.1: the reset counter shall be set to NC_CTR_RST_MC_THD + 1; this value shall be stored with complement CTR_RST_MC_CPL in a reset-safe area; the default value for CTR_RST_MC shall be NC_CTR_RST_MC_THD + 1 (in case of data corruption, assume permanent disabling).
R1_6.2: an MC error code which corresponds to the error location shall be stored, together with the indicator ”DISABLE request”; this value shall be stored in a reset-safe area.
R1_6.3: the update flag for MC error memory information shall be set by calling ACTION_ECM3_McSetErrmUpdateFlag(1).
R1_6.4: there shall be no attempt to restart the MC.

R1_7: No monitoring of state machine or complement checks using the RAM check service actions shall be performed in order to avoid recursive calls of the fault reaction; for complements, the check shall be performed manually and in case of inconsistency, the default value shall be used.

R1_8: The power latch resource for the Standard ROM check shall be unlocked by calling ACTION_ECM3_McUnlockPwlResource().

Description of ACTION_ECM3_McFaultReaction

Description:

Syntax:	ACTION_ECM3_McFaultReaction(IN <err_cod>)	
Parameter (IN):	err_cod	error code indicating the cause of the fault reaction
Short description:	This action performs the central fault reaction of processor monitoring on MC (as specified in "G1 Central fault reaction" above).	

Application conditions:

Initialisation: /* Using the RAM check actions (where applicable) */

CTR_RST_MC = 0;
ERR_COD_MC = NC_ERR_COD_MC_NOT_ERR;

Recurrence: The **initialisation** shall be **performed after each power-up**.

The **function** itself is **called when required**.

Activation: -

Deactivation: -

12.5.2 Healing of reset counter and error information

FUNCTION DESCRIPTION:

General information:

If the MC is in NORMAL state (which indicates that there is no error present) for a certain time (4s), then the reset counter is decreased by 1 since the system has been stable long enough to assume that there is currently no problem present; if this decreasing results in a reset counter of 0, also the error code and the error indication shall be cleared.

Description:

This function incorporates the following requirements:

G2 Healing of reset counter and error code

R2_1: The reset counter shall be decreased by 1 after 4000ms have passed.

R2_2: If the reset counter is decremented to 0, the error code shall be cleared, i.e., set to 0, and the error indication shall be changed to “no error present.”

R2_3: The function shall only be active in NORMAL state, and if LV_MC_ROM_CHK_READY is set (i.e., after the standard ROM check has been completed and no reset or transition to DISABLE has occurred, hence the ROM check must have been successful); LV_MC_ROM_CHK_READY has to be accessed together with its complement LV_MC_ROM_CHK_READY_CPL, using the actions from "RAM check service actions of processor monitoring on MC."

R2_4: The time measurement shall be performed by counting communication cycles, rather than actually measuring elapsed time, counting backwards starting from NC_CTR_COM_CYC_MC_DEC_RST_INI (see initialisation below).

R2_5: Additionally, the healing shall only take place if LV_MC_PFM_WRG_RESP_ACT_6 is set to one indicating that the wrong response test for PFM task 6 has been performed at least once (to avoid endless resets due to wrong PFM in task 6).

R2_6: While LV_MC_SOPC_ACT = 1, no healing shall take place (to prohibit healing while the SOPC has not yet been completed).

Application conditions:

Initialisation:	*/* Initialise communication cycle counter */* CTR_COM_CYC_MC_DEC_RST = NC_ CTR_COM_CYC_MC_DEC_RST_INI;
Recurrence:	The **initialisation** shall be performed **after each power-up or reset**. The **function itself** shall be **activated after the data from NORMAL communications have been evaluated successfully**.
Activation:	(STATE_MC = NC_STATE_MC_NORM) **AND** (LV_MC_ROM_CHK_READY = 1) **AND** (LV_MC_PFM_WRG_RESP_ACT_6 = 1) **AND** (LV_MC_SOPC_ACT = 0) *(only active in NORMAL AND after standard ROM check was completed AND wrong response test for PFM task 6 has been performed once AND SOPC has been completed)*
Deactivation:	Otherwise

12.5.3 State of fault reaction and up-to-dateness of error code

FUNCTION DESCRIPTION:

General information:

The error indication describes the current state of the fault reaction in the sense that, if the fault reaction is triggered and it is decided that the ECU should be restarted, the error indication is set to “ECU restart request”, while an (irreversible for the current driving cycle) DISABLE request leads to the error indication “DISABLE request” (the decision between these two choices depends of the value of the reset counter CTR_RST_MC, see section “Central fault reaction”).

Description:

This function incorporates the following requirements.

G3 State of fault reaction and up-to-dateness of error code

R3_1: In addition to the error code, there shall be the error indication STATE_ERR_IDC_MC which indicates the state of the fault reaction and the up-to-dateness of the error code:

- “No error present” shall be indicated while no error is present (including the case that the reset counter is being decremented);
- “ECU restart request” shall be indicated if fault reaction currently requests ECU restart (i.e., the reset counter has not yet reached a certain defined value);
- “DISABLE request” shall be indicated if fault reaction requests DISABLE instead of ECU restart (i.e., the reset counter has reached the defined value);
- “ECU restart was requested” (“previous restart request”) shall be indicated after ECU has been restarted due to an ECU restart request from the fault reaction.

R3_2: The error indication shall be stored in a reset-safe RAM area.

R3_3: After a reset but not after power-up, if the current error indication is “ECU restart request,” this shall be changed to “ECU restart was requested”: a restart must have actually been performed because otherwise the INIT state could not have been reached (see initialisation below).

R3_4: In INIT (i.e., STATE_MC = NC_STATE_MC_INI), the update flag for MC error memory information shall always be set by calling ACTION_ECM3_McSetErrmUpdateFlag(1); this shall be done after the potential change of the error indication described by R3_3.

R3_5: Once the CONFIG communication has been prepared, i.e., after ACTION_INFR_McSendErrInfo() has been called, if the state of the fault reaction on the MC indicated "ECU restart was requested", it shall be changed to "No error present" by setting STATE_ERR_IDC_MC to NC_STATE_ERR_IDC_MC_NOT_PRES.

Application conditions:

Initialisation: After power-up:

/ After power-up, initialise the error indication to "no error present" */*

```
STATE_ERR_IDC_MC =
        NC_STATE_ERR_IDC_MC_NOT_PRES;
```

After reset:

/ After reset, if "ECU restart request" is indicated, this shall be changed to "restart was requested": the restart request has been carried out successfully, see R3_3 above */*

```
IF (STATE_ERR_IDC_MC =
    NC_STATE_ERR_IDC_MC_REST_REQ)
THEN
    STATE_ERR_IDC_MC =
        NC_STATE_ERR_IDC_MC_PREV_REST;
FI
```

Recurrence: The **initialisation** shall be performed **after each power-up or reset**.

The **function itself** shall be **called**.

Activation: -

Deactivation: -

12.5.4 Transmission of error information from MC to MU

FUNCTION DESCRIPTION:

General information:

This function describes how the error information (reset counter, error code, and error indication) are sent from MC to MU.

Description:

This function incorporates the following requirements:

G4.1 Transmission of error information from MC to MU

R4.1_1: The MC shall send reset counter, error code, and error indication to the MU in the communications in the states CONFIG and DISABLE using the action call ACTION_INFR_McSendErrInfo(), before the respective communications are triggered, with arguments ERR_COD_MC, STATE_ERR_IDC_MC, and CTR_RST_MC.

R4.1_2: When preparing the data for a DISABLE communication, i.e., if STATE_MC = NC_STATE_MC_DI, and LV_MU_DI_ACT is set (i.e., if a disable request for the MU is still active, see "Disable request by external software"), the value NC_CTR_RST_MU_DI shall be used as one argument to ACTION_INFR_McSendErrInfo(), instead of CTR_RST_MC; NC_CTR_RST_MU_DI must be different from 0.

Application conditions:

Initialisation:	-
Recurrence:	The **function** shall be **executed before a communication is triggered**.
Activation:	(STATE_MC = NC_STATE_MC_CONF) **OR** (STATE_MC = NC_STATE_MC_DI) *(only active in CONFIG and DISABLE state)*
Deactivation:	Otherwise

12.5.5 Evaluation of error information from MU on MC

FUNCTION DESCRIPTION:

General information:

This function describes when and how the error information (reset counter, error code, and error indication) sent from the MU is evaluated on the MC.

Description:

This function incorporates the following requirements:

G4.2 Evaluation of error information from MU on MC

R4.2_1: In CONFIG and DISABLE, if communication monitoring indicates no error, i.e., LV_MC_COM_ERR = 0 (and its complement LV_MC_COM_ERR_CPL=1):
R4.2_1.1: the MC shall obtain the reset counter CTR_RST_MU of the MU using the action ACTION_INFR_McGetRstCtr();
R4.2_1.2: the MC shall obtain the error code ERR_COD_MU of the MU using the action ACTION_INFR_McGetErrCode();
R4.2_1.3: the MC shall obtain the error indication STATE_ERR_IDC_MU of the MU using the action ACTION_INFR_McGetErrIdc().

R4.2_2: The update flag for MU error memory information shall be set by calling ACTION_EMC3_McSetErrmUpdateFlag(0).

Application conditions:

Initialisation:	*/* Initialise MU error information */* CTR_RST_MU = 0;

ERR_COD_MU = NC_ERR_COD_MC_NOT_ERR;
STATE_ERR_IDC_MU = NC_STATE_ERR_IDC_MC_NOT_PRES;

Recurrence: The **initialisation** shall be performed **after each power-up or reset**.

The **function itself** shall be executed **after the communication monitoring functions** and **after G5.1 "Disable request by MU."**

Activation: ((STATE_MC = NC_STATE_MC_CONF) **OR**
(STATE_MC = NC_STATE_MC_DI)) **AND**
(LV_MC_COM_ERR = 0)
(only active after CONFIG or DISABLE communications, and if no communication error was detected)

Deactivation: Otherwise

12.5.6 Disable request by MU

FUNCTION DESCRIPTION:

General information:

This function deals with the case when the MU sends a DISABLE header. Receiving a DISABLE header from the MU always causes a transition to DISABLE and disabling of power stages.

Description:

This function incorporates the following requirements:

G5.1 Disable request by MU

R5.1_1: If the MC is in a state different from DISABLE (i.e., STATE_MC <> NC_STATE_MC_DI) and a DISABLE header is received from the MU (i.e., ACTION_ECM3_McGetHd() returns NC_STATE_MC_DI):
R5.1_1.1: a transition into DISABLE shall be performed, using ACTION_ECM3_McChangeState();
R5.1_1.2: the power-latch resource for the Standard ROM check shall be unlocked by calling ACTION_ECM3_McUnlockPwlResource();
R5.1_1.3: **for production SW**, if the MU sends the error indication "ECU restart request" (i.e., if ACTION_ECM3_McGetErrIdc() returns NC_STATE_ERR_IDC_MC_REST_REQ), the MC shall attempt to restart the ECU by calling the action ACTION_INFR_McRestart() (the error indication expresses that the MU has performed a fault reaction which should have triggered a reset but the reset was not performed); **for development SW**, no restart shall be attempted.

R5.1_2: If the value of MU_DI_STATE is different from "MU_DI_ERR", it shall be set to "MU_DI_OK" (if a corresponding disable request had already timed out, this shall not be masked by the MU finally responding with a DISABLE header).

R5.1_3: No monitoring of state machine shall be performed (since anyway a change into DISABLE is the consequence).

Application conditions:

Initialisation: -

Recurrence: The **function** shall be **executed after the communication monitoring functions**.

Activation: always

Deactivation: -

12.5.7 Disable request by external software

FUNCTION DESCRIPTION:

General information:

This function provides an interface for the application software in order to force both MC and MU into DISABLE state (in particular to be able to re-program the ECU without having to worry about time-out monitoring, PFM, FS-IST etc.).

Description:

This function incorporates the following requirements:

G5.2 Disable request by external software

R5.2_1: Processor monitoring shall export an interface to the application software in order to force a transition of the processor monitoring on the MC into DISABLE state using ACTION_ECM3_McChangeState(); if already in DISABLE, the MC shall stay there.

R5.2_2: The MC shall clear its error information, i.e., CTR_RST_MC, ERR_COD_MC, and STATE_ERR_IDC_MC.

R5.2_3: The disable request has four different states, recorded in MU_DI_STATE:
- "MU_DI_NOT_REQ" meaning "no disable request"
- "MU_DI_BUSY" meaning "MU has not yet responded, the request is still busy"
- "MU_DI_ERR" meaning "MU response has timed out"
- "MU_DI_OK" meaning "MU has responded in time"

R5.2_4: The interface shall be provided in the form of the action call ACTION_ECM3_DisableReq()
R5.2_4.1: when ACTION_ECM3_DisableReq() is called, and MU_DI_STATE is equal to "MU_DI_NOT_REQ", it shall be set to "MU_DI_BUSY"; *hence, if MU_DI_STATE is different from "MU_DI_NOT_REQ", it shall remain unchanged*;
R5.2_4.2: the flag LV_MU_DI_ACT shall be set (prompting sending NC_CTR_RST_MU_DI instead of the value of CTR_RST_MC (which is 0, see R2) to the MU in subsequent DISABLE communications, see "Transmission of error information from MC to MU" R3);
R5.2_4.3: ACTION_ECM3_DisableReq() shall return "false" in case MU_DI_STATE is equal to "MU_DI_BUSY" and true otherwise.

R5.2_5: After each subsequent communication, in case the MU has not responded with a DISABLE header (which leads to MU_DI_STATE being set to "MU_DI_OK", see "Disable request by MU" R2), i.e., if MU_DI_STATE is equal to "MU_DI_BUSY", the counter CTR_MU_DI shall be increased by one.

R5.2_6: When the counter CTR_MU_DI reaches the value NC_CTR_MU_DI_MAX, this shall be considered as a time-out, i.e., STATE_MU_DI shall be set to "MU_DI_ERR" (which also counts as the disable request being finished, however unsuccessfully).

R5.2_7: The action ACTION_ECM3_McResetRomChkFlags() shall be called (resetting the flags for the Standard ROM check).

R5.2_8: In CONFIG, just before triggering the CONFIG communication, MU_DI_STATE shall be changed from "MU_DI_OK" (the initialisation value) to "MU_DI_NOT_REQ"; in this way, as long as the transition from INIT to CONFIG has not yet been performed, the return value of ACTION_ECM3_DisableReq() will be 1 (instead of 0, cf. R5.2_4.1 and R5.2_4.3).

R5.2_9: In case ACTION_ECM3_DisableReq() is called, the power-latch resource for the Standard ROM check shall be unlocked by calling ACTION_ECM3_McUnlockPwlResource().

Description of ACTION_ECM3_DisableReq

Description:

Syntax:	ACTION_ECM3_DisableReq()
Parameter (OUT):	flag Indication whether the disable request has been completed (perhaps with error, cf. R5.2_4.3 above)
Short description:	This action implements the disable request as specified above in "G5.2 Disable request by external software."

Application conditions:

Initialisation:	MU_DI_STATE = "MU_DI_OK"; CTR_MU_DI = 0; LV_MU_DI_ACT = 0;
Recurrence:	The **initialisation** shall be **performed after each power-up or reset**. The **function itself** shall be **called via ACTION_ECM3_DisableReq()**.
Activation:	-
Deactivation:	-

12.5.8 Shutdown request by external software

FUNCTION DESCRIPTION:

General information:

This function provides an interface for the application software in order to restart the ECU.

Description:

This function incorporates the following requirements:

G5.3 Shutdown request by external software

R5.3_1: Processor monitoring shall export an interface to the application software in order to force restart of the ECU via the MU.

R5.3_2: The interface shall be provided in the form of the action call ACTION_ECM3_ShutDown().

R5.3_3: It behaves exactly like ACTION_ECM3_DisableReq() with the one exception that the flag LV_MU_DI_ACT is not set, i.e., in subsequent DISABLE communications, the real value of CTR_RST_MC (which is 0 since it has been reset by the shutdown/disable request) is sent; this will first force the MU into DISABLE, clearing its error information in the process, and eventually to restart the ECU (unless the MU has not detected an error on its own, and also the state of the ignition key plays a role in how the MU reacts to the request).

Description of ACTION_ECM3_ShutDown

Description:

Syntax:	ACTION_ECM3_ShutDown()
Parameter (OUT):	flag Indication whether the shutdown request has been finished (with or without error, cf. R5.2_4.3 above)
Short description:	This action requests a restart of the ECU via the MU as specified above in "G5.3 Shutdown request by external software" (whether this is actually done depends on the internal state of the MU).

Application conditions:

Initialisation:	-
Recurrence:	The **function itself** shall be **called via ACTION_ECM3_ShutDown()**.
Activation:	-
Deactivation:	-

12.5.9 The redundant switch-off path

FUNCTION DESCRIPTION:

General information:

In case process or processor monitoring on the MC detect a fatal error and attempt to disable the safety-relevant power stages, they can also use the switch-off paths on the monitoring unit to redundantly perform the switching-off. This is done via an action call (from within process monitoring) which sets a flag in the processor monitoring that forces the MC to deliberately send wrong FS-IST answers to the MU which in turn, potentially after some debouncing, leads to a fault reaction on the MU and in particular to disabling the safety-relevant power stages (hence the name “redundant switch-off path” as the MU is eventually prompted to also disable the power stages).

Description:

This function incorporates the following requirements:

G20 Redundant switch-off path

R20_1: There shall be an interface between process and processor monitoring which, when used, causes the processor monitoring on the MC to permanently send wrong FS-IST answers to the MU, in this way eventually (after debouncing) resulting in a fault reaction of the MU.

R20_2: This interface shall be realised by providing the action call ACTION_ECM3_RedSwitchOffPath() to process monitoring which shall set the flag LV_MC_DR_OFF; LV_MC_DR_OFF shall be stored as together with its complement LV_MC_DR_OFF_CPL and shall only be accessed via the actions ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() from "RAM check service actions of processor monitoring on MC"; its default value in case of a RAM error shall be 1, indicating that the redundant switch-off path has been triggered.

Description of ACTION_ECM3_RedSwitchOffPath

Description:

Syntax:	ACTION_ECM3_RedSwitchOffPath()
Short description:	This action triggers the redundant switch-off path by setting the flag LV_MC_DR_OFF using the action ACTION_ECM3_WriteChkCpl() (see “RAM check service actions of processor monitoring on MC”).

Application conditions:

Initialisation:	*/* Using the RAM check service actions */* LV_MC_DR_OFF = 0;
Recurrence:	The **initialisation** shall be performed **after each power-up or reset**. The **function itself** shall be called via ACTION_ECM3_RedSwitchOffPath().
Activation:	-
Deactivation:	-

12.5.10 Debounce mechanism

FUNCTION DESCRIPTION:

General information:

Debouncing is a very common task for many functions of processor monitoring. Hence this module exports actions which perform the two necessary tasks to do with debouncing: increasing one of these anti-bounce counters by a predefined increment, and decreasing such a counter with a predefined decrement. Note that increasing an anti-bounce counter can trigger a fault reaction (viz., if its maximal value is reached).

Note that values of anti-bounce counters are not exported directly!

Description:

In the following, let xxx range over all possible identifiers for anti-bounce counters which are used on the MC (e.g., one possible value is HD for header-byte monitoring). The function incorporates the following requirements:

G21 Debounce mechanism

R21_1: If an error has been detected the anti-bounce counter of the corresponding error location shall be increased by the corresponding increment NC_ABC_MC_xxx_INC without creating an overflow; this shall be achieved by calling ACTION_ECM3_McIncAbc().

R21_2: If the anti-bounce counter reaches a corresponding maximum value NC_ABC_MC_xxx_MAX a fault reaction shall be triggered.

R21_3: If no error is detected and the anti-bounce counter of the corresponding error location is greater than 0, the anti-bounce counter shall be decreased by the corresponding decrement NC_ABC_MC_xxx_DEC without creating an underflow; this shall be achieved by calling ACTION_ECM3_McDecAbc().

R21_4: In general, if not specified otherwise, the following configuration shall be used: NC_ABC_MC_xxx_INI = 0, NC_ABC_MC_xxx_INC = 2, NC_ABC_MC_xxx_DEC = 1, NC_ABC_MC_xxx_MAX = 6.

R21_5: After a reset, in INIT state, i.e., STATE_MC = NC_STATE_MC_INI:
R21_5.1: if any anti-bounce counter of communication monitoring (xxx=HD, CKS, CONF_DIF) is greater than 0, the flag LV_ABC_MC_ERR_COM shall be set to 1, otherwise it shall be set to 0;
R21_5.2: if any anti-bounce counter of PFM **except for task 6** (xxx=PFM_*k*_TOG, PFM_*k*_RESP **for 0 <=*k*<= 5**) is greater than 0, the flag LV_ABC_MC_ERR_PFM shall be set to 1, otherwise it shall be set to 0;
R21_5.3: if one of the two anti-bounce counters PFM_6_TOG or PFM_6_RESP **for task 6** is greater than 0, the flag LV_ABC_MC_ERR_PFM_6 shall be set to 1, otherwise it shall be set to 0;
R21_5.4: if any anti-bounce counter of FS-IST is greater than 0 (xxx=WRG_IDX, WRG_RESP) or greater than 2 (in case of xxx=NOT_DEC), the flag LV_ABC_MC_ERR_FCT_SPC_IST shall be set to 1, otherwise it shall be set to 0;
R21_5.5: after evaluation, the anti-bounce counters shall be initiatialised to their corresponding INI values NC_ABC_MC_xxx_INI.

Description of ACTION_ECM3_McIncAbc

Description:

Syntax:	ACTION_ECM3_McIncAbc(IN <abc_id>, IN <err_cod>)	
Parameter (IN):	abc_id	identifier for the anti-bounce counter to be increased
Parameter (IN):	err_cod	error code for a potential fault reaction
Short description:	This action increases the value of the specified anti-bounce counter by the corresponding increment without creating an overflow; if the corresponding maximal value is reached, a fault reaction shall be triggered by calling ACTION_ECM3_McFaultReaction().	

Description of ACTION_ECM3_McDecAbc

Description:

Syntax: ACTION_ECM3_McDecAbc(IN <abc_id>)

Parameter (IN): abc_id identifier for the anti-bounce counter to be increased

Short description: This action decreases the value of the specified anti-bounce counter by 1 without creating an underflow.

Application conditions:

Initialisation: After each power-up:

All anti-bounce counters are initialised to their corresponding INI values.

After each power-up and reset:

LV_ ABC_MC_ERR_COM = 0;
LV_ ABC_MC_ERR_PFM = 0;
LV_ ABC_MC_ERR_PFM_6 = 0;
LV_ ABC_MC_ERR_FCT_SPC_IST = 0;

Recurrence: The **initialisation** shall be performed **after each power-up and reset** (see above).

Afterwards, **increasing** or **decreasing** of anti-bounce counters shall be done **via action calls**.

Activation: in all states *(in particular in in INIT state!)*

Deactivation: -

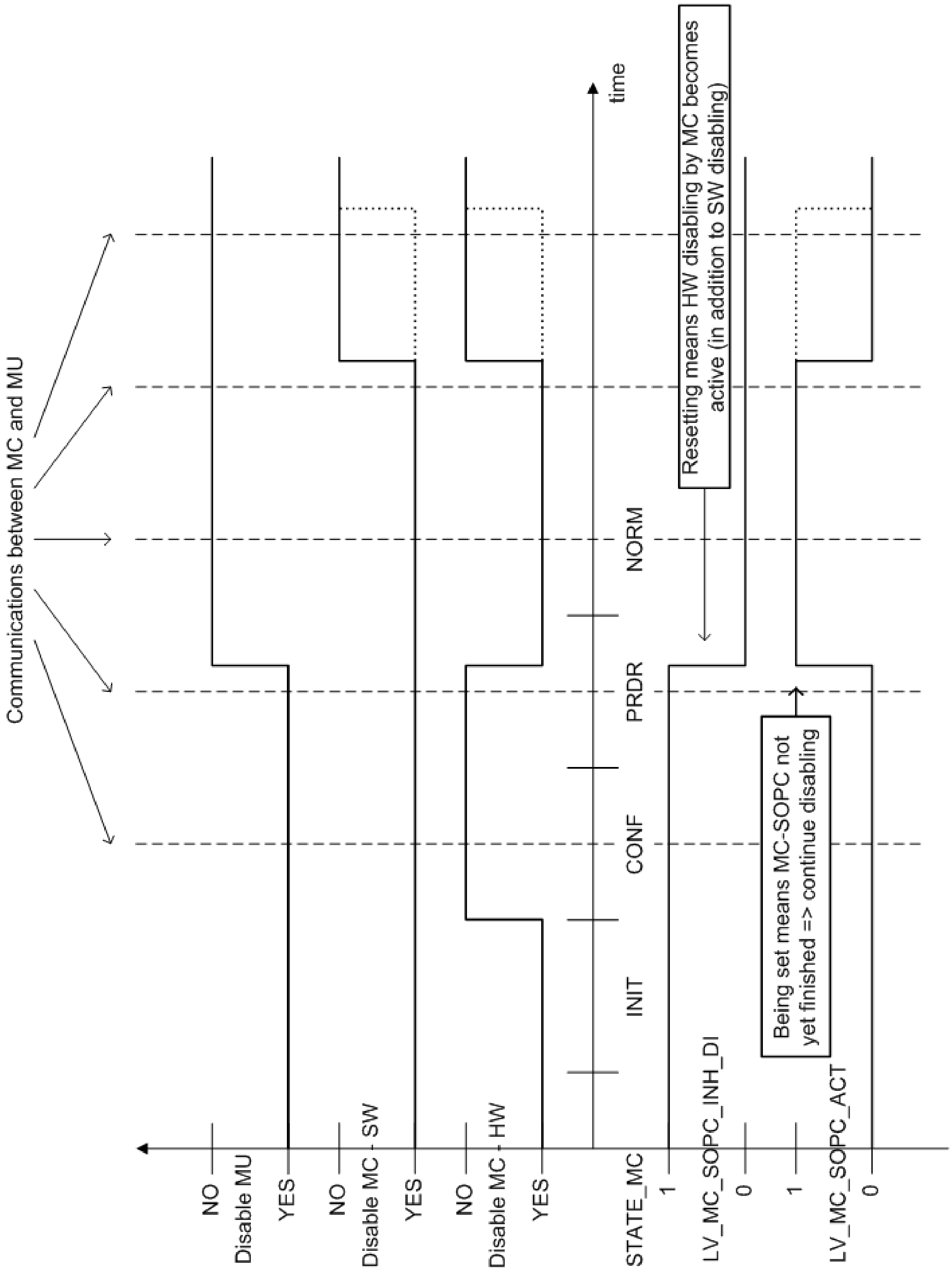
12.5.11 Disabling of power stages

FUNCTION DESCRIPTION:

General information:

This function describes how and when the MC disables the power stages it is connected to, separately for each state.
The phrase "enabling of power stages" shall be used to denote the fact that the MC no longer *disables*; however, this does not imply anything regarding *actively using* the power stages.
Note that on the MC, there are two ways of disabling: one is by SW done via flags which lead to no active signals being generated (e.g., no injection patterns); the second one is via HW where dedicated HW lines are used to disable power stages (the MU only has such HW disable lines).

Signal flow diagram:



Note that the diagram only shows the case when no temporary disabling is performed; the distinction between SW and HW disabling on the MC is managed by the switch-off path check.

Description:

The complete function satisfies the following requirements:

G22 Disabling of power stages

R22_1: The MC shall:
R22_1.1: permanently disable the power stages by setting LV_ERR_MU_MC = 1 (it has to be ensured that LV_ERR_MU_MC, after being set once, is not reset again), plus additionally calling ACTION_ECM2_LockPws() (in order to ensure that process monitoring also disables the power stages);
R22_1.2: temporarily disable the power stages by setting LV_ERR_TMP_MU_MC = 1, plus additionally calling ACTION_ECM2_LockPws() (in order to ensure that process monitoring also disables the power stages);
R22_1.3: enable the power stages by setting LV_ERR_TMP_MU_MC = 0 and LV_ERR_MU_MC = 0 (cf. remark above!), plus additionally calling the action ACTION_ECM2_ResetErrorFlags() (in order to ensure that process monitoring also stops disabling the power stages if appropriate).

R22_2: Both error flags shall be stored together with their respective complements, LV_ERR_MU_MC_CPL and LV_ERR_TMP_MU_MC_CPL, and shall be accessed using the imported RAM check service actions ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() (see "RAM check service actions of processor monitoring on MC"); their default values in case of a RAM error shall be 1, i.e., indicating that the power stages shall be disabled.

R22_3: The power stages shall be disabled depending on the reset counters of both MC and MU, and the current state of the monitoring concept.

R22_4: In INIT, the MC shall temporarily disable the power stages.

R22_5: In CONFIG, the MC shall temporarily disable the power stages.

R22_6: In PREDRIVE:
R22_6.1: the MC shall temporarily disable the power stages;
R22_6.2: before the PREDRIVE communication, if any of the two reset counters (CTR_RST_MC, or CTR_RST_MU) is > 1, the MC shall set CTR_COM_CYC_MC_DR_DI_TMP to NC_CYCNR_MC_DR_DI_TMP (this counter shall be used to temporarily disable the power stages until in total NC_CYCNR_MC_DR_DI_TMP + 2 communication cycles have been performed).

R22_7: In NORMAL:
R22_7.1: if CTR_COM_CYC_MC_DR_DI_TMP is not active (i.e., = 0), the MC shall enable the power stages (see signal flow diagram); this only holds if no pre-condition of any other requirement is satisfied (cf. in particular R22_9 and R22_10).
R22_7.2.: if CTR_COM_CYC_MC_DR_DI_TMP is active (i.e., <> 0):
R22_7.2.1: the MC shall temporarily disable the power stages;
R22_7.2.2: CTR_COM_CYC_MC_DR_DI_TMP shall be decreased by one without creating an underflow.

R22_8: In DISABLE, the MC shall permanently and temporarily disable the power stages.

R22_9: In PREDRIVE and NORMAL, the MC shall consider the status of the disable outputs of the MU regarding disabling the power stages:
R22_9.1: the MC shall call the actions ACTION_INFR_McGetNdisOut0() and ACTION_INFR_McGetNdisOut1() to obtain the status informationand store the results in LV_MU_DI_OUT_0 and LV_MU_DI_OUT_1 (note that disabling means status = 0, enabling means status = 1);

R22_9.2: the information about the status of the MU disable outputs shall be used as follows:
R22_9.2.1: if LV_MU_DI_OUT_0 = 0 and LV_MU_DI_OUT_1 = 1 and LV_MU_IGN_KEY = 0 and NLC_MU_INJ_OFF_TMR_ENA = 1, the injection-off timer (IOT) in the MU shall be assumed to be active; hence, the following assignment shall be performed:

- LV_INJ_OFF_TMR_INJ_ENA_TMP = 0 *(IOT active, no enabling of injection)*;

R22_9.2.2: if LV_MU_DI_OUT_0 = 0 and LV_MU_DI_OUT_1 = 0, the MU disables both its disable outputs; hence, the following assignment shall be performed:

- LV_ERR_TMP_MU_MC = 1 *(temporary disabling)*

R22_9.2.3: if LV_MU_DI_OUT_0 = 1 and LV_MU_DI_OUT_1 = 1, the MU enables its disable outputs; hence, the IOT must be inactive and the following assignment shall be performed:

- LV_INJ_OFF_TMR_INJ_ENA_TMP = 1 *(IOT inactive, enabling of injection)*;

R22_9.2.4: otherwise, the following assignment shall be performed:

- LV_ERR_TMP_MU_MC = 1 *(temporary disabling)*

R22_10: In NORMAL, while LV_MC_SOPC_ACT = 1 (i.e., the MC-part of the SOPC has not yet been finished), the MC shall temporarily disable the power stages.

R22_11: In PREDRIVE and NORMAL, if the MU indicates disabling any of its power stages (i.e., LV_MU_DI_OUT_0 or LV_MU_DI_OUT_1 = 0), the flag LV_MC_INH_SOPC shall be set to 1; otherwise, it shall be set to 0; the flag being set consequently indicates that the MU still disables the power stages (and therefore prompts a delay the MC-part of the SOPC).

Application conditions:

Initialisation: *(/* Initialise communication cycle counter for temporary disabling to "no temporary disabling" */)*

CTR_COM_CYC_MC_DR_DI_TMP = 0;

/ Delay of MC-part of SOPC (not yet) necessary */*

LV_MC_INH_SOPC = 0;

/ Using the RAM check service actions: temp. disabling */*

LV_ERR_MU_MC = 0;
LV_ERR_TMP_MU_MC = 1;
LV_MU_DI_OUT_0 = 0;
LV_MU_DI_OUT_1 = 0;

/ Power stages disabled after reset */*

LV_INJ_OFF_TMR_ENA_INJ_TMP = 0;

Recurrence: The **initialisation** shall be performed **after each power-up or reset**.

The **function itself** is **active in all states**.

Activation: always

Deactivation: -

12.5.12 Acquisition of under-/over-voltage flags

FUNCTION DESCRIPTION:

General information:

The MU contains detection mechanisms for both under- and over-voltage conditions in case its voltage leaves its operating range.
This is, amongst others, prompts a reset (which is held for as long as either condition persists) of MC and MU.
If the under- or over-voltage condition is present long enough, one of the flags LV_V_L_DET_MU or LV_V_H_DET_MU (as appropriate) is set. Once the voltage returns to its normal range, the system starts up and the two flags (which have been stored in a reset-safe area) are sent to the MC.

Description:

The function incorporates the following requirement:

G24 Acquisition of under-/over-voltage flags

R24_1: If no errors were detected in the CONFIG communication, i.e., if LV_MC_COM_ERR = 0 (and its complement LV_MC_COM_ERR_CPL = 1), the flags LV_V_L_DET_MU and LV_V_H_DET_MU shall be assigned the values sent by the MU, using the actions ACTION_INFR_McGetUvdFlag() and ACTION_INFR_McGetOvdFlag(), respectively.

Application conditions:

Initialisation:	LV_V_H_DET_MU = 0; LV_V_L_DET_MU = 0;
Recurrence:	The **initialisation** shall be performed **after each power-up or reset**. The **function itself** shall be executed **after the communication monitoring functions.**
Activation:	(STATE_MC = NC_STATE_MC_CONF) **AND** (LV_MC_COM_ERR = 0) *(only active in CONFIG and if no communication error was detected)*
Deactivation:	Otherwise

12.6 MC Communication Monitoring between MC and MU

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MC_CKS_OK	-	0...1H	0...1	1	[-]
Local variable: flag indicating whether the (XOR) checksum of the data received from the MU is correct					
LV_MC_CKS_OK_CPL	-	0...1H	0...1	1	[-]
Local variable: complement of LV_MC_CKS_OK					
LV_MC_HD_OK	-	0...1H	0...1	1	[-]
Local variable: flag indicating whether the received header was correct or not					
LV_MC_HD_OK_CPL	-	0...1H	0...1	1	[-]
Local variable: complement of LV_MC_HD_OK					
LV_MC_PAR_OK	-	0...1H	0...1	1	[-]
Local variable: flag indicating whether the data received from the MU had the correct (even) parity					
LV_MC_PAR_OK_CPL	-	0...1H	0...1	1	[-]
Local variable: complement of LV_MC_PAR_OK					
STATE_MU_TMP	V/O/S	0...FFH	0...255	1	[-]
Header byte of MU as sent to MC (stored in reset-safe manner)					
LV_MC_COM_ERR	V/O	0...1H	0...1	1	[-]
Error flag indicating any communication error (checksum, header, or parity)					
LV_MC_COM_ERR_CPL	V/O	0...1H	0...1	1	[-]
Complement of LV_MC_COM_ERR					
LV_MU_READY	V/O	0...1H	0...1	1	[-]
Flag indicating whether MU is ready (=1) or not (=0)					
CTR_MU_READY	-	0...FFH	0...255	1	[-]
Local variable: counter for attempts of check of MU readiness					
STATE_MU_TMP_CPL	O/S	0...FFH	0...255	1	[-]
Complement of STATE_MU_TMP (stored in reset-safe manner)					

Input data:

STATE_MC	NC_STATE_MC_CONF	NC_STATE_MC_PRDR	NC_STATE_MC_NORM
NC_STATE_MC_DI	NC_ERR_COD_MC_CKS	NC_ERR_COD_MC_HD	NC_ERR_COD_MC_PAR
NC_ABC_MC_CKS	NC_ABC_MC_HD	NC_STATE_MC_INI	NC_ERR_COD_MC_MU_READY
NC_STATE_MC_NOT_VLD			

Import actions:

ACTION_ECM3_McFaultReaction(IN <>)
ACTION_ECM3_McIncAbc(IN <>, IN <>)
ACTION_ECM3_McDecAbc(IN <>)
ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)
ACTION_ECM3_McReadChkCpl(OUT <>, IN <>, IN <>, IN <>)
ACTION_ECM3_McReadChkState(OUT <>)
ACTION_INFR_McStartCom(IN <>)
ACTION_INFR_McChkCks(OUT <>)
ACTION_INFR_McGetHd(OUT <>)
ACTION_INFR_McChkPar(OUT <>)
ACTION_INFR_StartMuReadyChk()
ACTION_INFR_MuReady(OUT <>)

FUNCTION DESCRIPTION:

General information:

This module describes all functions related to monitoring of communications on the MC.

The complete function is split according to the states where the communication occurs, and also there is a second split between triggering a communication, and monitoring the data from a preceding communication; functions for communication monitoring are triggered after the communication between MC and MU is finished.

Note that header-byte monitoring, in case a DISABLE header is received from the MU, only performs the consecutive parity-bit checking; the transition to DISABLE, which is also a consequence of the received header, is specified in the function "G5.1 Disable request by MU" (cf. R5.1_1.1).

A noteworthy point for communications in the states NORMAL and DISABLE is that the preparation of the next communication needs to be performed at a certain point *when all the necessary other functions have been executed and completed* as they prepare the data to be sent in the next communication; this can be contrasted to the two functions for CONFIG and PREDRIVE which are essentially triggered by state changes (with the additional point that certain data have to be sent to the infrastructure so that the correct values are transmitted to the MU).

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_T_MC_COM_CYC	1	0...FFH	0...255	1	[-]
Time between triggering two communications					
NC_CTR_MU_READY_MAX	1	0...FFH	0...255	1	[-]
Maximal number how often the check for MU readiness shall be performed before triggering a fault reaction					

12.6.1 Check of MU readiness

FUNCTION DESCRIPTION:

General information:

The start-up on MC and MU is done asynchronously, so potentially the MC could already be willing to trigger the first communication while the MU has not yet completed its start-up. A communication error would be the consequence, even if the communication itself was performed without any problem – the MU simply was not yet ready and had, e.g., not yet properly prepared the data for the first communication.

In order to avoid such situations, the MC – as the master of the communication – shall first check whether the MU is ready. If the MU is ready, the first communication shall be triggered. Otherwise, the MU readiness check shall be repeated for a certain number of times, and if the MU is still not ready after a configurable number of attempts, a fault reaction shall be performed.

The configuration can be adjusted in such a way that the readiness check shall be performed forever without ever becoming actually ready; this is intended for such situations where no MU is available in the HW.

Description:

The function shall incorporate the following requirements:

G57 Check of MU readiness

R57_1: The MC shall trigger the readiness check by calling the action ACTION_INFR_StartMuReadyChk().

R57_2: In the next recurrence, the MC shall fetch the result of the MU readiness check by calling the action ACTION_INFR_MuReady() with argument LV_MU_READY.

R57_3: If LV_MU_READY = 1, the MU is ready; this information is used as one of the conditions for enabling the transition from INIT to CONFIG (where the first communication will then be triggered, cf. G12).

R57_4: If LV_MU_READY = 0, CTR_MU_READY shall be increased by 1.

R57_5: If CTR_MU_READY > NC_CTR_MU_READY_MAX, the central fault reaction shall be triggered by calling ACTION_ECM3_McFaultReaction() with argument NC_ERR_COD_MC_MU_READY.

R57_6: If CTR_MU_READY <= NC_CTR_MU_READY_MAX, the sequence shall be started again from R57_1 onwards, i.e., calling ACTION_INFR_StartMuReadyChk() still in the same recurrence.

Application conditions:

Initialisation:	LV_MU_READY = 0; CTR_MU_READY = 0;
Recurrence:	The **initialisation** shall be performed **after each power-up or reset**. The **function itself** shall be executed **every NC_T_MC_COM_CYC ms**.
Activation:	(STATE_MC = NC_STATE_MC_INI) **AND** (LV_MU_READY <> 1) *(only active in INIT state and when the check has not yet been finished)*
Deactivation:	Otherwise

12.6.2 Trigger CONFIG communication

FUNCTION DESCRIPTION:

General information:

This function describes the triggering of the CONFIG communication by the MC.

Description:

The function incorporates the following requirements:

G12 Trigger CONFIG communication

R12_1: In the CONFIG communication, the following data are sent from the MC to the MU:

- error code;
- error indication;
- reset counter;
- configuration data for MU (plus their complements).

(See the description of the communication protocol; further note that these data are *not* imported here because sending them to the infrastructure so that they are available for the communication is the task of the corresponding modules where the data are exported)

R12_2: The communication is triggered by the MC once these data have been sent to the infrastructure by the responsible modules ("Central fault reaction", "State of fault reaction and up-to-dateness of error code", "Configuration of MU"); this shall be achieved by calling the action ACTION_INFR_McStartCom() with argument STATE_MC.

Application conditions:

Initialisation:	-
Recurrence:	The **function** shall be **activated after the relevant data** (see R12_1 above) **have been transported to the infrastructure**.
Activation:	STATE_MC = NC_STATE_MC_CONF *(only active in CONFIG state)*
Deactivation:	Otherwise

12.6.3 Communication monitoring in CONFIG state

FUNCTION DESCRIPTION:

General information:

This function describes how the CONFIG communication is monitored on the MC in the CONFIG state. Note that this really only refers to communication monitoring but not to evaluation of the received data: in case an error was encountered, an error flag (LV_MC_COM_ERR) shall be set which is subsequently exported to other processor monitoring functions so that those do not get activated in case of a communication error.

Description:

The function incorporates the following requirements:

G13 Communication monitoring in CONFIG state

R13_1: The sequence of checks shall be communication checksum monitoring, followed by communication header-byte monitoring, and finally parity-bit checking.

R13_2: For communication checksum monitoring:
R13_2.1: the action ACTION_INFR_McChkCks() shall be called returning the flag LV_MC_CKS_OK (which is set if the checksum was correct);
R13_2.2: the variable LV_MC_CKS_OK shall be stored together with its complement LV_MC_CKS_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0;
R13_2.3: in case of a checksum error:

R13_2.3.1: an immediate fault reaction shall be carried out by calling ACTION_ECM3_McFaultReaction() with error code NC_ERR_COD_MC_CKS;
R13_2.3.2: LV_MC_COM_ERR shall be set;
R13_2.3.3: no further communication monitoring shall be performed.

R13_3: For communication header-byte monitoring:
R13_3.1: the action ACTION_INFR_McGetHd() shall be called returning the header byte STATE_MU_TMP sent by the MU in the preceding communication; STATE_MU_TMP shall be stored together with its complement STATE_MU_TMP_CPL; both variables shall be stored in a reset-safe manner;
R13_3.2: the only acceptable header bytes are CONFIG and DISABLE;
R13_3.3: in case a CONFIG header is received, LV_MC_HD_OK shall be set and the communication monitoring shall continue (with parity-bit checking);
R13_3.4: in case a DISABLE header is received, the flag LV_MC_HD_OK shall be set and parity-bit checking shall be performed as for DISABLE communications (see section "Communication monitoring in DISABLE state" R4);
R13_3.5: in case of receiving a header-byte other than CONFIG or DISABLE:
R13_3.5.1: an immediate fault reaction shall be triggered by calling ACTION_ECM3_McFaultReaction with error code NC_ERR_COD_MC_HD;
R13_3.5.2: the flag LV_MC_HD_OK shall be reset;
R13_3.5.3: LV_MC_COM_ERR shall be set;
R13_3.5.4: no further communication monitoring shall be performed;
R13_3.6: the variable LV_MC_HD_OK shall be stored together with its complement LV_MC_HD_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0.

R13_4: For parity-bit checking:
R13_4.1: the action ACTION_INFR_McChkPar() shall be used which returns the flag LV_MC_PAR_OK (which is set of the parity was correct);
R13_4.2: the variable LV_MC_PAR_OK shall be stored together with its complement LV_MC_PAR_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0;
R13_4.3 in case of a parity error:
R13_4.3.1: an immediate fault reaction shall be triggered by calling ACTION_ECM3_McFaultReaction with error code NC_ERR_COD_MC_PAR;
R13_4.3.2: LV_MC_COM_ERR shall be set.

R13_5: The variable LV_MC_COM_ERR shall be stored together with its complement LV_MC_COM_ERR_CPL and accessed using the actions ACTION_ECM3_WriteChkCpl(),ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() from section "RAM check service actions of processor monitoring on MC"; its default value shall be 1.

Application conditions:

Initialisation: *\/* After each power-up *\/*

STATE_MU_TMP = NC_STATE_MC_INI;

/ After each power-up or reset, using the RAM check service actions */*

LV_MC_CKS_OK = 0;

LV_MC_HD_OK = 0;
LV_MC_PAR_OK = 0;
LV_MC_COM_ERR = 0;

Recurrence: The **initialisation** shall be performed **after each power-up or reset (as indicated above)**.

The **function itself** shall be triggered **after the communication between MC and MU is finished**.

Activation: STATE_MC = NC_STATE_MC_CONF *(only active in CONFIG state)*

Deactivation: Otherwise

12.6.4 Trigger PREDRIVE communication

FUNCTION DESCRIPTION:

General information:

This function describes the triggering of the PREDRIVE communication by the MC.

Description:

The function incorporates the following requirements:

G14 Trigger PREDRIVE communication

R14_1: In the PREDRIVE communication, the following data are sent from the MC to the MU:

- compatibility number;
- configuration data for MU (plus their complements).

(See the description of the communication protocol; further note that these data are *not* imported here because sending them to the infrastructure so that they are available for the communication is the task of the corresponding modules where the data are exported)

R14_2: The communication is triggered by the MC once these data have been sent to the infrastructure by the responsible modules ("Version compatibility check", "Configuration of MU"); this shall be achieved by calling the action ACTION_INFR_McStartCom() with argument STATE_MC.

Application conditions:

Initialisation: -

Recurrence: The **function** shall be **activated after the relevant data** (see R1 above) have **been transported to the infrastructure** and **if NC_T_MC_COM_CYC ms have passed since the last communication**.

Activation: STATE_MC = NC_STATE_MC_PRDR *(only active in PREDRIVE state)*

Deactivation: Otherwise

12.6.5 Communication monitoring in PREDRIVE state

FUNCTION DESCRIPTION:

General information:

This function describes how the PREDRIVE communication is monitored on the MC in the PREDRIVE state. Note that this really only refers to communication monitoring but not to evaluation of the received data: in case an error was encountered, an error flag (LV_MC_COM_ERR) shall be set which is subsequently exported to other processor monitoring functions so that those do not get activated in case of a communication error.

Description:

The function incorporates the following requirements:

G15 Communication monitoring in PREDRIVE state

R15_1: The sequence of checks shall be communication checksum monitoring, followed by communication header-byte monitoring.

R15_2: For communication checksum monitoring:
R15_2.1: the action ACTION_INFR_McChkCks() shall be called returning the flag LV_MC_CKS_OK (which is set if the checksum was correct);
R15_2.2: the variable LV_MC_CKS_OK shall be stored together with its complement LV_MC_CKS_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0;
R15_2.3: in case of a checksum error:
R15_2.3.1: an immediate fault reaction shall be carried out by calling ACTION_ECM3_McFaultReaction with error code NC_ERR_COD_MC_CKS;
R15_2.3.2: LV_MC_COM_ERR shall be set;
R15_2.3.3: no further communication monitoring shall be performed.

R15_3: For communication header-byte monitoring:
R15_3.1: the action ACTION_INFR_McGetHd() shall be called returning the header byte STATE_MU_TMP sent by the MU in the preceding communication; STATE_MU_TMP shall be stored together with its complement STATE_MU_TMP_CPL; both variables shall be stored in a reset-safe manner;
R15_3.2: the only acceptable header bytes are PREDRIVE and DISABLE;
R15_3.3: in case a PREDRIVE header is received, LV_MC_HD_OK shall be set and communication monitoring is finished;
R15_3.4: in case a DISABLE header is received, LV_MC_HD_OK shall be set and parity-bit checking shall be performed as for DISABLE communications (see section "Communication monitoring in DISABLE state" R19_4);
R15_3.5: in case of receiving a header-byte other than PREDRIVE or DISABLE:
R15_3.5.1: an immediate fault reaction shall be triggered by calling ACTION_ECM3_McFaultReaction with error code NC_ERR_COD_MC_HD;
R15_3.5.2: the flag LV_MC_HD_OK shall be reset;
R15_3.5.3: LV_MC_COM_ERR shall be set;
R15_3.5.4: no further communication monitoring shall be performed;
R15_3.5.5: the variable LV_MC_HD_OK shall be stored together with its complement LV_MC_HD_OK_CPL and it shall be accessed using the actions from section "RAM check

service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0.

R15_4: The variable LV_MC_COM_ERR shall be stored together with its complement LV_MC_COM_ERR_CPL and accessed using the actions ACTION_ECM3_WriteChkCpl(),ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() from section "RAM check service actions of processor monitoring on MC."

Application conditions:

Initialisation:	*/* Using the RAM check actions */* LV_MC_COM_ERR = 0;
Recurrence:	The **initialisation** shall be performed **before the PREDRIVE communication is triggered**. The **function itself** shall be triggered **after the communication between MC and MU is finished**.
Activation:	STATE_MC = NC_STATE_MC_PRDR *(only active in PREDRIVE state)*
Deactivation:	Otherwise

12.6.6 Trigger NORMAL communication

FUNCTION DESCRIPTION:

General information:

This function describes the triggering of the NORMAL communication by the MC.

Description:

The function incorporates the following requirements:

G16 Trigger NORMAL communication

R16_1: In the NORMAL communication, the following data are sent from the MC to the MU:
- the PFM response;
- the FS-IST response;
- the value of the ignition key;
- the flag for the redundant switch-off path.

(See the description of the communication protocol; further note that these data are *not* imported here because sending them to the infrastructure so that they are available for the communication is the task of the corresponding modules where the data are exported)

R16_2: The communication is triggered by the MC once these data have been sent to the infrastructure by the responsible modules ("Post-processing of PFM response before communications", "Post-processing of the FS-IST data on MC for transmission to MU", "Exporting the value of the ignition key", "The redundant switch-off path"); this shall be achieved by calling the action ACTION_INFR_McStartCom() with argument STATE_MC (note that the function can be called either shortly after the transition PREDRIVE→NORMAL, or after all other functions in NORMAL have been executed in case the NORMAL state is not left).

Application conditions:

Initialisation: -

Recurrence: The **function** shall be **activated after the relevant data** (see R16_1 above) **have been transported to the infrastructure** and **if NC_T_MC_COM_CYC ms have passed since the last communication**.

Activation: STATE_MC = NC_STATE_MC_NORM *(only active in NORMAL state)*

Deactivation: Otherwise

12.6.7 Communication monitoring in NORMAL state

FUNCTION DESCRIPTION:

General information:

This function describes how the NORMAL communications are monitored on the MC in the NORMAL state. Note that this really only refers to communication monitoring but not to evaluation of the received data: in case an error was encountered, an error flag (LV_MC_COM_ERR) shall be set which is subsequently exported to other processor monitoring functions so that those do not get activated in case of a communication error.

Description:

The function incorporates the following requirements:

G17 Communication monitoring in NORMAL state

R17_1: The sequence of checks shall be communication checksum monitoring, followed by communication header-byte monitoring.

R17_2: For communication checksum monitoring:
R17_2.1: the action ACTION_INFR_McChkCks() shall be called which returns the flag LV_MC_CKS_OK (which is set if the checksum was correct);
R17_2.2: the variable LV_MC_CKS_OK shall be stored together with its complement LV_MC_CKS_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0;
R17_2.3: in case of a checksum error, a debounced fault reaction shall be carried out:
R17_2.3.1: in case the checksum is wrong, the corresponding anti-bounce counter shall be increased (potentially triggering a fault reaction) by calling ACTION_ECM3_McIncAbc() with anti-bounce counter identifier NC_ABC_MC_CKS and error code NC_ERR_COD_MC_CKS;
R17_2.3.2: LV_MC_COM_ERR shall be set;
R17_2.3.3: no further communication monitoring shall be performed;
R17_2.4: in case the checksum is correct, the corresponding anti-bounce counter shall be decreased by calling ACTION_ECM3_McDecAbc() with anti-bounce counter identifier NC_ABC_MC_CKS.

R17_3: For communication header-byte monitoring:
R17_3.1: the action ACTION_INFR_McGetHd() shall be called returning the header byte STATE_MU_TMP sent by the MU in the preceding communication; STATE_MU_TMP shall be stored together with its complement STATE_MU_TMP_CPL; both variables shall be stored in a reset-safe manner;
R17_3.2: the only acceptable header bytes are NORMAL and DISABLE;
R17_3.3: in case a NORMAL header is received, LV_MC_HD_OK shall be set and communication monitoring is finished;
R17_3.4: in case a DISABLE header is received, LV_MC_HD_OK shall be set and parity bit checking shall be performed as for DISABLE communications (see section "Communication monitoring in DISABLE state" R19_4);
R17_3.5: in case of receiving a header-byte different from NORMAL or DISABLE, a debounced fault reaction shall be carried out:
R17_3.5.1: in case the header-byte is wrong, the corresponding anti-bounce counter shall be increased (potentially triggering a fault reaction) by calling ACTION_ECM3_McIncAbc() with anti-bounce counter identifier NC_ABC_MC_HD and error code NC_ERR_COD_MC_HD;
R17_3.5.2: the flag LV_MC_COM_ERR shall be set;
R17_3.5.3: no further communication monitoring shall be performed;
R17_3.5.4: the flag LV_MC_HD_OK shall be reset (in case the fault reaction is not correctly performed);
R17_3.6: in case the header-byte is NORMAL, the corresponding anti-bounce counter shall be decreased by calling ACTION_ECM3_McDecAbc() with anti-bounce counter identifier NC_ABC_MC_HD;
R17_3.7: the variable LV_MC_HD_OK shall be stored together with its complement LV_MC_HD_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0.

R17_4: The variable LV_MC_COM_ERR shall be stored together with its complement LV_MC_COM_ERR_CPL and accessed using the actions ACTION_ECM3_WriteChkCpl(),ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() from section "RAM check service actions of processor monitoring on MC."

Application conditions:

Initialisation:	*/* Initialise LV_MC_COM_ERR using the RAM check actions */* LV_MC_COM_ERR = 0;
Recurrence:	The **initialisation** shall be performed **before a NORMAL communication is triggered**. The **function itself** shall be triggered **after the communication between MC and MU is finished**.
Activation:	STATE_MU = NC_STATE_MU_NORM *(only active in NORMAL state)*
Deactivation:	Otherwise

12.6.8 Trigger DISABLE communication

FUNCTION DESCRIPTION:

General information:

This function describes the triggering of the DISABLE communication by the MC.

Description:

The function incorporates the following requirements:

G18 Trigger DISABLE communication

R18_1: In the DISABLE communication, the following data are sent from the MC to the MU:
- error code;
- error indication;
- reset counter;
- the value of the ignition key;

(See the description of the communication protocol; further note that these data are *not* imported here because sending them to the infrastructure so that they are available for the communication is the task of the corresponding modules where the data are exported)

R18_2: The communication is triggered by the MC once these data have been sent to the infrastructure by the responsible modules ("Exporting the value of the ignition key", "The redundant switch-off path"); this shall be achieved by calling the action ACTION_INFR_McStartCom() with argument STATE_MC (note that the function can be called either shortly after the transition into DISABLE from a different state, or after all other functions in DISABLE have been executed in case the DISABLE state is not left).

Application conditions:

Initialisation:	-
Recurrence:	The **function** shall be acti**vated after the relevant data** (see R18_1 above) **have been transported to the infrastructure** and **if NC_T_MC_COM_CYC ms have passed since the last communication**.
Activation:	STATE_MC = NC_STATE_MC_DI *(only active in DISABLE state)*
Deactivation:	Otherwise

12.6.9 Communication monitoring in DISABLE state

FUNCTION DESCRIPTION:

General information:

This function describes how the DISABLE communication is monitored on the MC in the DISABLE state. Note that this really only refers to communication monitoring but not to evaluation of the received data: in case an error was encountered, an error flag (LV_MC_COM_ERR) shall be set which is subsequently exported to other processor monitoring functions so that those do not get activated in case of a communication error.

Description:

The function incorporates the following requirements:

G19 Communication monitoring in DISABLE state

R19_1: The sequence of checks shall be communication checksum monitoring, followed by communication header-byte monitoring, and finally parity-bit checking.

R19_2: For communication checksum monitoring:
R19_2.1: the action ACTION_INFR_McChkCks() shall be called which returns the flag LV_MC_CKS_OK (which is set if the checksum was correct);
R19_2.2: the variable LV_MC_CKS_OK shall be stored together with its complement LV_MC_CKS_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0;
R19_2.3: in case of a checksum error, no fault reaction shall be carried out, only LV_MC_COM_ERR shall be set and no further communication monitoring shall be performed.

R19_3: For communication header-byte monitoring:
R19_3.1: the action ACTION_INFR_McGetHd() shall be called returning the header byte STATE_MU_TMP sent by the MU in the preceding communication; STATE_MU_TMP shall be stored together with its complement STATE_MU_TMP_CPL; both variables shall be stored in a reset-safe manner;
R19_3.2: the only acceptable header bytes is DISABLE;
R19_3.3: in case a DISABLE header is received, LV_MC_HD_OK shall be set and the communication monitoring shall continue (with parity-bit checking);
R19_3.4: in case of receiving a header-byte different from DISABLE, no fault reaction shall be carried out, only LV_MC_COM_ERR shall be set and no further communication monitoring shall be performed; furthermore, also the flag LV_MC_HD_OK shall be reset;
R19_3.5: the variable LV_MC_HD_OK shall be stored together with its complement LV_MC_HD_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring" ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl(); its default value shall be 0.

R19_4: For parity-bit checking:
R19_4.1: the action ACTION_INFR_McChkPar() shall be used which returns the flag LV_MC_PAR_OK (which is set of the parity was correct);
R19_4.2: the variable LV_MC_PAR_OK shall be stored together with its complement LV_MC_PAR_OK_CPL and it shall be accessed using the actions from section "RAM check service actions of Processor Monitoring"; its default value shall be 0 ACTION_ECM3_WriteChkCpl(), ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl();
R19_4.3: in case of a parity error, no fault reaction shall be triggered, only LV_MC_COM_ERR shall be set, and no further communication monitoring shall be performed.

R19_5: The variable LV_MC_COM_ERR shall be stored together with its complement LV_MC_COM_ERR_CPL and accessed using the actions ACTION_ECM3_WriteChkCpl(),ACTION_ECM3_ChkCpl(), ACTION_ECM3_ReadChkCpl(), and ACTION_ECM3_McReadChkCpl() from section "RAM check service actions of processor monitoring on MC."

Application conditions:

Initialisation: *`/* Initialise LV_MC_COM_ERR using the RAM check actions */`*

LV_MC_COM_ERR = 0;

Recurrence: The **initialisation** shall be performed **before a DISABLE communication is triggered**.

The **function itself** shall be triggered **after the communication between MC and MU is finished**.

Activation: STATE_MU = NC_STATE_MU_DI *(only active in DISABLE state)*

Deactivation: Otherwise

12.7 MC Switch-Off Path Check

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MC_SOPC_INH_DI	V/O	0...1H	0...1	1	[-]
Flag indicating to other functions not to disable the power stages via HW disable lines because the MU-part of the SOPC is active					
LV_MC_SOPC_INH_DI_CPL	O	0...1H	0...1	1	[-]
Complement of LV_MC_SOPC_INH_DI					
STATE_MC_RESP_SOPC	V	0...FFH	0...255	1	[-]
Local variable: final result of the SOPC					
LV_MC_SOPC_ACT	V/O	0...1H	0...1	1	[-]
Flag indicating whether MC SOPC is still active (=1, i.e., not finished) or not (=0)					
IDX_STATE_SOPC	V/O/S	0...FH	0...15	1	[-]
Index into lookup table to determine the final result of the SOPC (stored in reset-safe manner)					
IDX_STATE_SOPC_CPL	O/S	0...FH	0...15	1	[-]
Complement of IDX_STATE_SOPC (stored in reset-safe manner)					

Note: mode "S" shall mean storage in reset-safe, but not necessarily non-volatile memory.

Input data:

CTR_RST_MC	CTR_RST_MC_CPL	LV_MC_INH_SOPC	NC_STATE_MC_INI
NC_STATE_MC_CONF	NC_STATE_MC_PRDR	NC_STATE_MC_NORM	NC_STATE_MC_DI
STATE_MC			

Export actions:

ACTION_ECM3_McCancelSopc()
This action cancels the SOPC.

Description for actions:

ACTION_ECM3_McCancelSopc()

When called, this action resets LV_MC_SOPC_INH_DI in order to ensure that the power stages can be properly disabled by the MC; this is used inside G1 "Central Fault Reaction" which could be activated while the SOPC is not yet finished.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit

Import actions:

ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>, IN <>)
ACTION_ECM3_McReadChkCpl(OUT <>, IN <>, IN <>, IN <>)
ACTION_ECM3_McReadChkState(OUT <>)
ACTION_INFR_McGetDiagPws0(OUT <>)
ACTION_INFR_McGetDiagPws1(OUT <>)

FUNCTION DESCRIPTION:

General information:

The switch-off path check (SOPC, previously also known as "Predrive Check") is part of the standard monitoring duties performed by the processor monitoring functions.

Its objective is to determine whether the system's *switch-off paths*, i.e., dedicated physical connections with the purpose to disable the safety-relevant power stages are fully functional, in case an error should necessitate this action,.

Both MC and MU can have up to two such switch-off paths each connected to MC and MU.

The check itself is performed during the system's start-up (see signal flow diagram below).

The principle idea of the check is that one of the two system components (MC or MU) attempts to disable the power stages it is connected to and then, after obtaining status information from the power stages, it is judged whether they are indeed disabled.

The status of the power stages during the SOPC shall be collected on the MC, simply because the MU cannot read back the status of the power stages.

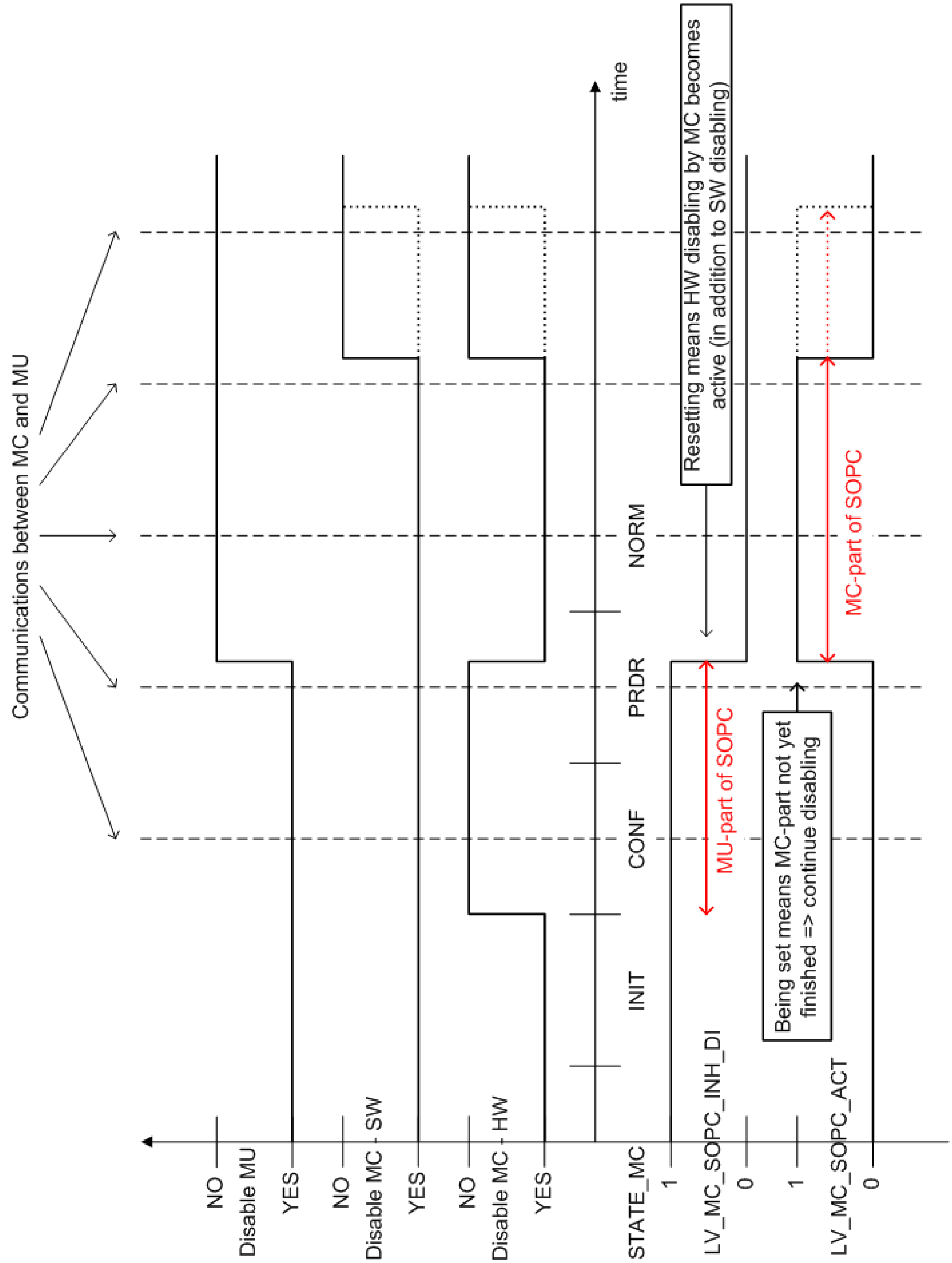
It is clear that the SOPC has to be synchronised between the MC and the MU in order to avoid conflicts stemming from the need to unequivocally decide whether a given switch-off path works or not (e.g., the MC and MU paths must not be tested at the same time because it might not be possible to determine which component is actually responsible for the disabling or, should the disabling fail, which switch-off path to which component is broken).

Since communications constitute the main synchronisation mechanism between MC and MU, the SOPC is synchronised according to different communications between MC and MU.

This specification describes the generic procedure; any project-specific deviation or extension shall be specified in the application incidences, function "G27 Configuration for switch-off path check."

Signal flow diagram:

The sequencing of the switch-off path check with respect to the different communications is shown in the following diagram (note that the diagram only shows the case when no temporary disabling is performed):


Communications between MC and MU
time
Resetting means HW disabling by MC becomes active (in addition to SW disabling)
INIT
CONF
PRDR
NORM
MU-part of SOPC
MC-part of SOPC
Being set means MC-part not yet finished => continue disabling
NO
Disable MU
YES
NO
Disable MC - SW
YES
NO
Disable MC - HW
YES
STATE_MC
1
LV_MC_SOPC_INH_DI
0
1
LV_MC_SOPC_ACT
0

Description:

The function shall incorporate the following requirements (always refer to the signal flow diagram):

G26 Switch-off path check

R26_1: The status information of a power stage shall be retrieved using the actions ACTION_INFR_McGetDiagPws0() and ACTION_INFR_McGetDiagPws1(); the requirements to these actions are described in the section "ECM3 - MC requirements to infrastructure interface."

R26_2: For the check of the MU switch-off paths:
R26_2.1: it shall be performed after the transition from INIT to CONFIG but before triggering the PREDRIVE communication (since, under regular circumstances, the MU stops disabling the power stages after the PREDRIVE communication, see signal flow diagram);
R26_2.2: if any of the two actions returns "undefined" as status information from a power stage, it shall be assumed that the corresponding switch-off path is not working;
R26_2.3: the result of the SOPC for the MU paths shall be stored as the two lower bits in the 4-bit vector IDX_STATE_SOPC, which shall be stored together with its complement IDX_STATE_SOPC_CPL (see "RAM check service actions of processor monitoring on MC"), its default value being NC_IDX_STATE_SOPC_DFCT (= no switch-off path working);
R26_2.4: the diagnostic result for power stage 0 shall be the MSB of the two bits reserved for the MU results;
R26_2.5: a set bit shall indicate that the switch-off path is working;
R26_2.6: when storing the two MU results, the upper two bits shall be reset (since IDX_STATE_SOPC is stored in reset-safe memory, these might still contain old data from a previous SOPC).

R26_3: The flag LV_MC_SOPC_INH_DI shall be handled as follows:
R26_3.1: it shall be set for the duration of the MU-part of the SOPC (from initialisation to shortly before the PREDRIVE communication);
R26_3.2: it shall be reset after the MU-part of the SOPC **but before the PREDRIVE communication**;
R26_3.3: it shall be stored together with its complement LV_MC_SOPC_INH_DI_CPL pair and accessed using the actions from "RAM check service actions of processor monitoring on the MC", with default value 0, indicating inactive SOPC.

R26_4: For the check of the MC switch-off paths:
R26_4.1: it shall be performed at some point after triggering the PREDRIVE communication (under regular circumstances, the MU then no longer disables the power stages);
R26_4.2: if any of the two actions returns "undefined" as status information from a power stage, it shall be assumed that the corresponding switch-off path is not working;
R26_4.3: the result of the SOPC for the MC paths shall be stored as the two upper bits in the 4-bit vector IDX_STATE_SOPC, which shall be stored together with its complement IDX_STATE_SOPC_CPL (see "RAM check service actions of processor monitoring on MC"), its default value being NC_IDX_STATE_SOPC_DFCT (= no switch-off path working);
R26_4.4: the diagnostic result for power stage 0 shall be the MSB of the two bits reserved for the MC results;
R26_4.5: a set bit shall indicate that the switch-off path is working.

R26_5: As for the exact timing of the MC-part of the SOPC:

R26_5.1: after the MU-part of the SOPC **but before the PREDRIVE communication**, LV_MC_SOPC_ACT shall be set to 1 in order to indicate that the MC-part of the SOPC has not yet been completed (this prompts disabling of power stages by the MC, cf. R22_10);
R26_5.2: the MC-part of the SOPC can only be performed if LV_MC_INH_SOPC = 0 (this indicates that the power stages are no longer temporarily disabled and that the MU no longer disables the power stages); under regular circumstances, this leaves time between after the PREDRIVE communication and before triggering the second NORMAL communication (cf. R22_7.1 and R22_9) to perform the check;
R26_5.3: otherwise, if LV_MC_INH_SOPC = 1, the MC-part of the SOPC has to be postponed until R26_5.2 becomes applicable (note that LV_MC_SOPC_ACT remains set!);
R26_5.4: after the MC-part of the SOPC has been finished, LV_MC_SOPC_ACT shall be reset (allowing the power stages to be enabled, cf. R22_10).

R26_6: By the previous requirements R26_2 and R26_4, the index IDX_STATE_SOPC for the lookup in one of the result tables is generated by concatenating the results of MC and MU paths as depicted below:

Bit 3	Bit 2	Bit 1	Bit 0
MC result PSW0	MC result PWS1	MU result PWS0	MU result PWS1

R26_7: For determining the final result of the SOPC, depending on the value of CTR_RST_MC (for which a consistency check with its complement CTR_RST_MC_CPL using ACTION_ECM3_ReadChkCpl() has to be performed), the following cases shall be distinguished:
R26_7.1: if CTR_RST_MC < NC_CTR_RST_MC_THD_SOPC, the final result of the switch-off path check STATE_MC_RESP_SOPC shall be obtained by a lookup in the table NC_RESP_SOPC_1[] at index IDX_STATE_SOPC;
R26_7.2: if CTR_RST_MC ≥ NC_CTR_RST_MC_THD_SOPC, the final result of the switch-off path check STATE_MC_RESP_SOPC shall be obtained by a lookup in the table NC_RESP_SOPC_2[] at index IDX_STATE_SOPC;
R26_7.3: the configuration of the lookup tables shall be described in the application incidences; in this way, the number of switch-off paths are connected to each component can be taken into account by putting dummy values into the lookup tables at appropriate positions.

R26_8: For the final result STATE_MC_RESP_SOPC:
R26_8.1: it shall be obtained by a lookup in one of the two tables NC_RESP_SOPC_x as described in R26_7;
then shall determine the behaviour of the system as a reaction to the SOPC results; it shall be stored as a value/complement pair and be accessed by the actions from "RAM check service actions of processor monitoring on MC", its default value being NC_STATE_RESP_SOPC_DFCT (indicating a complete failure of the SOPC); the two simple cases are: allowing the engine to start-up as usual in case the result is NC_STATE_RESP_SOPC_OK (the switch-off path check was completely successful); performing the central fault reaction of processor monitoring in case NC_STATE_RESP_SOPC_DFCT (in case the status of the switch-off paths would not permit reaching the safe state of the system in case of a safety-critical error); since other reactions might be desired, there can be up to 256 different values which (potentially) could correspond to 256 different engine start-up behaviours (but it should be noted that there are only up to 16 different configurations of the status information of the power stages); the considered values (in addition to the default cases NC_STATE_RESP_SOPC_DFCT and NC_STATE_RESP_SOPC_OK), plus the corresponding reactions, are also described in the application incidences.

R26_9: In DISABLE, the SOPC shall be cancelled; this means that LV_MC_SOPC_INH_DI shall be reset to 0 (and its complement set to 1) to indicate the end of the MU-part, and that LV_MC_SOPC_ACT shall be reset to 0 to indicate the end of the MC-part.

R26_10: If LV_MC_INH_SOPC performs a transition from 0 to 1 during the MC-part of the SOPC (including the determination of the result), the MC-part has to be restarted (see R26_5; the MC-part can only start anew when LV_MC_INH_SOPC = 0).

Description for action ACTION_ECM3_McCancelSopc:

ACTION_ECM3_McCancelSopc()

When called, this action resets LV_MC_SOPC_INH_DI in order to ensure that the power stages can be properly disabled by the MC; this is used inside G1 "Central Fault Reaction" which could be activated while the SOPC is not yet finished.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit

Formula section:

/* NOT using the RAM check service actions (to avoid recursion in fault reaction)! */

LV_MC_SOPC_INH_DI = 0;
LV_MC_SOPC_INH_DI_CPL = complement(0);

Application conditions:

Initialisation: **After each power-up:**

/ Using the RAM check service actions */*

IDX_STATE_SOPC = NC_IDX_STATE_SOPC_DFCT;

After each power-up or reset:

/ Using the RAM check service actions */*

LV_MC_SOPC_INH_DI = 1;
STATE_MC_RESP_SOPC = NC_STATE_RESP_SOPC_DFCT;

/ MC-part of the SOPC not yet active, cf. R26_5.1*/*

LV_MC_SOPC_ACT = 0;

Recurrence: The **initialisation** shall be performed **after each power-up or reset (see above)**.

The **function itself** is **spread over the states INIT, CONFIG, PREDRIVE, and NORMAL**, as depicted in the signal flow diagram.

Activation: STATE_MC <> NC_STATE_MC_DI *(inactive in DISABLE state)*

Deactivation: STATE_MC = NC_STATE_MC_DI *(inactive in DISABLE state)*

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CTR_RST_MC_THD_SOPC	1	0...FH	0...15	1	[-]
SOPC threshold for reset counter determining from which lookup table the final result shall be obtained					
NC_NR_RESP_SOPC	1	0...FFH	0...255	1	[-]
Size of arrays for switch-off path check					
NC_RESP_SOPC_1[NC_NR_RESP_SOPC]	1	0...FFH	0...255	1	[-]
Lookup table for final result if reset counter < SOPC threshold					
NC_RESP_SOPC_2[NC_NR_RESP_SOPC]	1	0...FFH	0...255	1	[-]
Lookup table for final result if reset counter >= SOPC threshold NC_CTR_RST_MC_THD_SOPC					
NC_IDX_STATE_SOPC_DFCT	1	0...FH	0...15	1	[-]
Constant for lookup index indicating that no corresponding switch-off path is working (all defective)					
NC_STATE_RESP_SOPC_DFCT	1	0...FFH	0...255	1	[-]
Constant indicating that the SOPC completely failed					
NC_STATE_RESP_SOPC_OK	1	0...FFH	0...255	1	[-]
Constant indicating that the SOPC was completely successful					
NC_STATE_RESP_DR_DI	1	0...3H	0...3	1	[-]
Diagnostic result of power stage (driver): driver disabled (i.e., switch-off path is working)					
NC_STATE_RESP_DR_ENA	1	0...3H	0...3	1	[-]
Diagnostic result of power stage (driver): driver enabled (i.e., switch-off path is not working)					
NC_STATE_RESP_DR_DFT	1	0...3H	0...3	1	[-]
Diagnostic result of power stage (driver): default value "undefined" (i.e., no check could be performed)					

12.8 MC Program Flow Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LF_MC_RCV_PFM_BYTE_MON	-	0...FFH	0...255	1	[-]
Local variable: "raw" data sent by MU obtained via infrastructure (used for monitoring of MU)					
LF_MC_PFM_SYN_BYTE_MON	V	0...7FH	0...127	1	[-]
Local variable: PFM synchronisation byte which is updated synchronously to communications and used for monitoring the PFM data sent by the MU (timely updates, wrong response test)					
LF_MC_RCV_PFM_BYTE_TOG	-	0...FFH	0...255	1	[-]
Local variable: "raw" data sent by MU obtained via infrastructure (used for PFM service actions)					
LF_MC_PFM_SYN_BYTE_TOG	V	0...7FH	0...127	1	[-]
Local variable: PFM synchronisation byte which is updated synchronously with monitored functionality (modified by PFM service actions)					
LF_STATE_ABC_PFM_MU	-	0...7FH	0...127	1	[-]
Local variable: "status byte" recording, for each PFM anti-bounce counter on the MU, whether the counter has currently value 0 or not (updated using LF_MC_RCV_PFM_BYTE received from MU)					
LF_MC_PFM_SYN_BYTE_CHG	-	0...7FH	0...127	1	[-]
Local variable: "change marker" recording the differences between a newly received synchronisation bytes and the previously received one (precisely those bits are set where the two synchronisation bytes differ)					
LF_MC_PFM_SYN_BYTE_CHG_OLD	-	0...7FH	0...127	1	[-]
Local variable: saved value of the previous "change marker"					
STATE_MC_PFM_CTL_BYTE[NC_NR_MC_PFM]	-	0...FFH	0...255	1	[-]
Local variable: array of PFM control bytes					
TCC_MC_PFM_TOG[NC_NR_MC_PFM_TCC]	-	0...FFH	0...255	1	[-]
Local variable: array of test cycle counters for monitoring the timely toggling of bits in the synchronisation byte by the MU for tasks 0 to 5					
TCC_MC_PFM_TOG_6	-	0...FFFFH	0...65535	1	[-]
Local variable: test cycle counter for PFM task 6 (requires higher resolution than the others)					
LF_MC_WRG_RESP_PFM_ACT	-	0...7FH	0...127	1	[-]
Local variable: bit field where a set bit k means that a wrong PFM response shall be sent out for task k					
CTR	-	0...7H	0...7	1	[-]
Local variable: counter for PFM tasks					
LV_MC_PFM_WRG_RESP_ACT_6	V/O	0...1H	0...1	1	[-]
Flag indicating that the wrong response test for PFM task 6 has been performed at least once (needed for healing of reset counter and error information on the MC)					
LF_MC_RESP_PFM	-	0...7FH	0...127	1	[-]
The PFM response of the MC (to be sent to the MU via the infrastructure)					
LV_MC_CKS_OK_PFM	-	0...1H	0...1	1	[-]
Local variable: flag indicating whether the (XOR) checksum of the data received from the MU is correct					
STATE_MU_TMP_PFM	-	0...FFH	0...255	1	[-]
Local variable: header byte of MU as sent to MC					

Input data:

STATE_MC	NC_STATE_MC_PRDR	NC_STATE_MC_NORM	NC_ABC_MC_PFM_0_TOG
NC_ABC_MC_PFM_1_TOG	NC_ABC_MC_PFM_2_TOG	NC_ABC_MC_PFM_3_TOG	NC_ABC_MC_PFM_4_TOG
NC_ABC_MC_PFM_5_TOG	NC_ABC_MC_PFM_6_TOG	NC_ERR_COD_MC_PFM_0_TOG	NC_ERR_COD_MC_PFM_1_TOG
NC_ERR_COD_MC_PFM_2_TOG	NC_ERR_COD_MC_PFM_3_TOG	NC_ERR_COD_MC_PFM_4_TOG	NC_ERR_COD_MC_PFM_5_TOG
NC_ERR_COD_MC_PFM_6_TOG	NC_ABC_MC_PFM_0_RESP	NC_ABC_MC_PFM_1_RESP	NC_ABC_MC_PFM_2_RESP
NC_ABC_MC_PFM_3_RESP	NC_ABC_MC_PFM_4_RESP	NC_ABC_MC_PFM_5_RESP	NC_ABC_MC_PFM_6_RESP

NC_ERR_COD_MC_PFM_0_RESP	NC_ERR_COD_MC_PFM_1_RESP	NC_ERR_COD_MC_PFM_2_RESP	NC_ERR_COD_MC_PFM_3_RESP
NC_ERR_COD_MC_PFM_4_RESP	NC_ERR_COD_MC_PFM_5_RESP	NC_ERR_COD_MC_PFM_6_RESP	LV_MC_COM_ERR
LV_MC_COM_ERR_CPL			

Export actions:

ACTION_ECM3_Service0TaskPfm(IN <k>)
Service action for "inspection point" 0 in task *k*.
ACTION_ECM3_Service1TaskPfm(IN <k>)
Service action for "inspection point" 1 in task *k*.
ACTION_ECM3_Service2TaskPfm(IN <k>)
Service action for "inspection point" 2 in task *k*.
ACTION_ECM3_Service3TaskPfm(IN <k>)
Service action for "inspection point" 3 in task *k*.
ACTION_ECM3_Service4TaskPfm(IN <k>)
Service action for "inspection point" 4 in task *k*.
ACTION_ECM3_Service5TaskPfm(IN <k>)
Service action for "inspection point" 5 in task *k*.
ACTION_ECM3_Service6TaskPfm(IN <k>)
Service action for "inspection point" 6 in task *k*.
ACTION_ECM3_Service7TaskPfm(IN <k>)
Service action for "inspection point" 7 in task *k*.
ACTION_ECM3_Service8TaskPfm(IN <k>)
Service action for "inspection point" 8 in task *k*.
ACTION_ECM3_Service9TaskPfm(IN <k>)
Service action for "inspection point" 9 in task *k*.
ACTION_ECM3_Service10TaskPfm(IN <k>)
Service action for "inspection point" 10 in task *k*.
ACTION_ECM3_Service11TaskPfm(IN <k>)
Service action for "inspection point" 11 in task *k*.
ACTION_ECM3_Service12TaskPfm(IN <k>)
Service action for "inspection point" 12 in task *k*.
ACTION_ECM3_Service13TaskPfm(IN <k>)
Service action for "inspection point" 13 in task *k*.
ACTION_ECM3_Service14TaskPfm(IN <k>)
Service action for "inspection point" 14 in task *k*.
ACTION_ECM3_Service15TaskPfm(IN <k>)
Service action for "inspection point" 15 in task *k*.
ACTION_ECM3_Service16TaskPfm(IN <k>)
Service action for "inspection point" 16 in task *k*.
ACTION_ECM3_Service17TaskPfm(IN <k>)
Service action for "inspection point" 17 in task *k*.
ACTION_ECM3_Service18TaskPfm(IN <k>)
Service action for "inspection point" 18 in task *k*.
ACTION_ECM3_Service19TaskPfm(IN <k>)
Service action for "inspection point" 19 in task *k*.
ACTION_ECM3_Service20TaskPfm(IN <k>)
Service action for "inspection point" 20 in task *k*.
ACTION_ECM3_Service21TaskPfm(IN <k>)
Service action for "inspection point" 21 in task *k*.
ACTION_ECM3_Service22TaskPfm(IN <k>)
Service action for "inspection point" 22 in task *k*.
ACTION_ECM3_Service23TaskPfm(IN <k>)
Service action for "inspection point" 23 in task *k*.
ACTION_ECM3_Service24TaskPfm(IN <k>)
Service action for "inspection point" 24 in task *k*.

ACTION_ECM3_Service25TaskPfm(IN <k>)
Service action for "inspection point" 25 in task *k*.
ACTION_ECM3_Service26TaskPfm(IN <k>)
Service action for "inspection point" 26 in task *k*.
ACTION_ECM3_Service27TaskPfm(IN <k>)
Service action for "inspection point" 27 in task *k*.
ACTION_ECM3_Service28TaskPfm(IN <k>)
Service action for "inspection point" 28 in task *k*.
ACTION_ECM3_Service29TaskPfm(IN <k>)
Service action for "inspection point" 29 in task *k*.
ACTION_ECM3_ServicePfm(IN <k>)
This action checks whether all PFM service actions in task *k* have been executed in the correct order by comparing the control byte associated to task *k* to a (configurable) value and, if that is the case, sets the value of the corresponding bit in the PFM response appropriately (=negated value of the same bit in the current synchronisation byte).

Import actions:

ACTION_ECM3_McIncAbc(IN <>, IN <>)
ACTION_ECM3_McDecAbc(IN <>)
ACTION_ECM3_McReadChkState(OUT <>)
ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)
ACTION_INFR_McGetPfmByte(OUT <>)
ACTION_INFR_McChkCks(OUT <>)
ACTION_INFR_McGetHd(OUT <>)
ACTION_INFR_McSendPfmResp(IN <>)

FUNCTION DESCRIPTION:

General information:

Program flow monitoring (PFM) is one of the ways in which the processor monitoring functionality monitors whether the MC-SW, most importantly the process monitoring, is still executed “properly.” From the perspective of PFM, “properly” means that the safety-relevant functions of process monitoring are executed with the correct recurrence and in the correct sequence. The basic algorithm of PFM works as follows.

The MU sends a bit pattern, the so-called *PFM synchronisation byte*, to the MC: this synchronisation byte contains one bit for each of 7 *PFM tasks* (numbered from 0 to 6) to be monitored. Here, “PFM task” denotes a collection of software modules (=functions), in particular of process monitoring, which have to be executed in a defined order and which have to be executed in the same time grid. There can be up to 30 "inspection points" in each of the monitored tasks (e.g., if 3 inspections per module are performed, this means up to 10 modules per task).

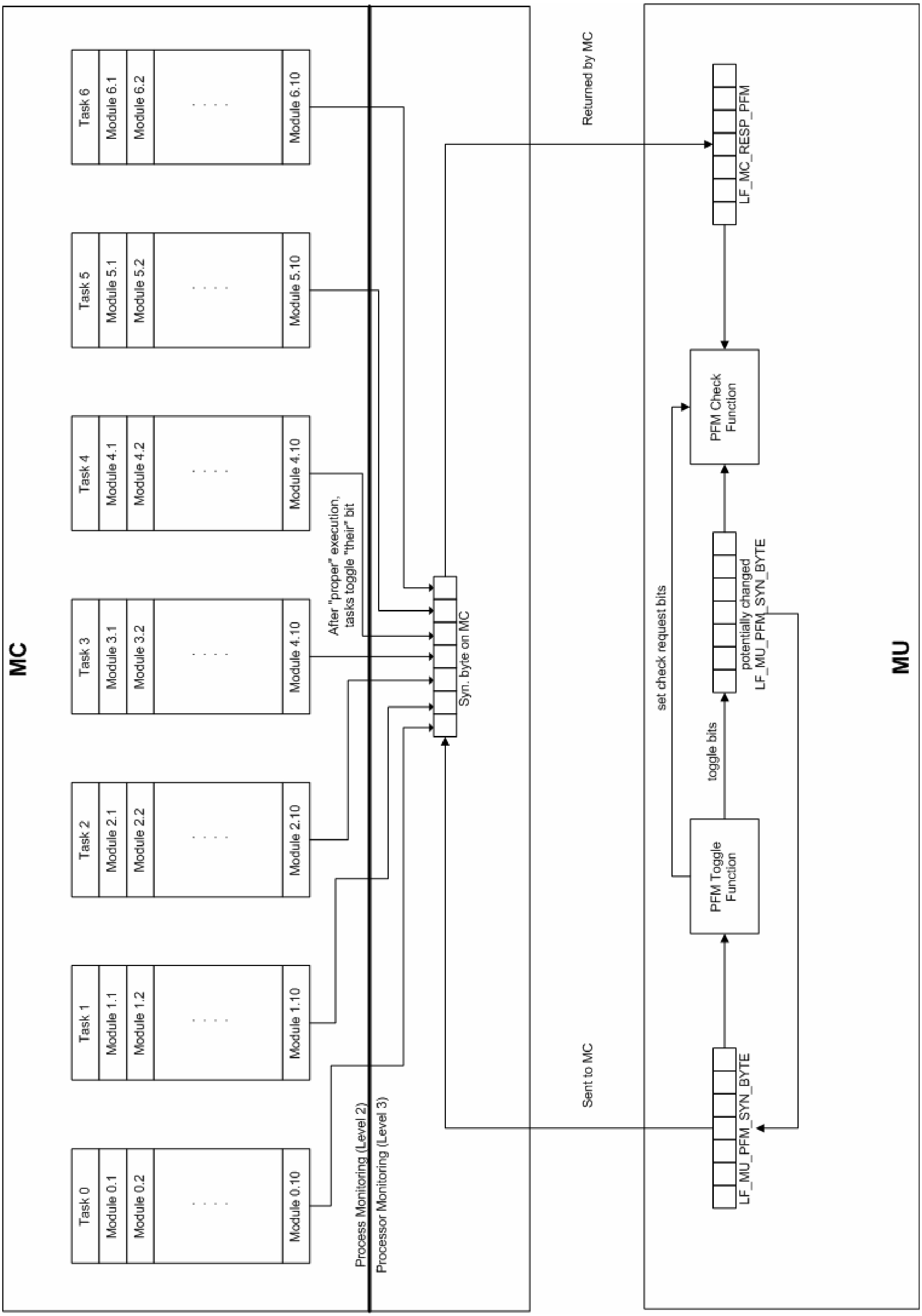
Having been received by the MC, the synchronisation byte is used to obtain the so-called *PFM response byte*: if the monitored modules (of process monitoring or others) from one task *are executed* and, furthermore, if they are *executed in the correct order* (note that these are two different constraints!), they manipulate the bit in the response byte corresponding to ‘their’ task in such a way that it is finally toggled. The (thus potentially updated) response byte is sent back to the MU. In how far the sent response byte (MC → MU) differs from the originally transmitted synchronisation byte (MU → MC) depends solely on which monitored modules were executed until the communication where the response byte was transmitted.

It should be noted that the MU sends two kinds of information: synchronisation bytes and status bytes, and that there is at least one status byte sent between two synchronisation bytes because of the toggle frequency on the MU. Such a status byte contains information about PFM anti-bounce counters on the MU: a set bit indicates that the corresponding counter has value > 0.

Meanwhile, on the MU, the synchronisation byte that was previously sent to the MC, is also manipulated: specific bits are toggled in a way in line with what is expected from the MC in terms of toggling the bits, assuming that the process monitoring tasks are executed “properly” in the sense explained above. When the PFM response of the MC (= a potential update by the MC of the original synchronisation byte) is received, it can thus be compared with the (now also on the MU manipulated) synchronisation byte on the MU.

Since the way the MU manipulates the synchronisation byte is the reference for how the MC should do the same task, a mismatch between the synchronisation byte and the PFM response of the MC indicates that the program execution on the MC is not performed in the prescribed way; as a consequence, a debounced fault reaction will be performed, using separate anti-bounce counters for each of the 7 tasks.

Signal flow diagram:



Description:

The PFM on the MC has the following main tasks:

1. decoding of PFM byte sent by the MU;
2. PFM service actions;
3. monitoring of timely updates of the synchronisation byte by the MU;
4. test MU reaction to wrong PFM responses;
5. post-processing of PFM response before communications.

These sub-tasks of PFM shall, in the following, be represented as the sub-functions of this module.

A special case arises for the first sub-function. It is possible that the monitored functionality (e.g., process monitoring) is running asynchronously to the communications between MC and MU. Therefore, the sub-function must be split in two: one function running synchronously with the communications dealing with testing the MU responses (timely updates and reaction to deliberately wrong responses); another one running synchronously with the fastest monitored task acquiring the synchronisation byte for toggling by the PFM service actions in the same time grid as the modules under PFM inspection.

The last sub-function which is responsible for preparing the PFM response sent to the MU only takes into account that synchronisation byte which used for the service actions since only this can provide the information which modules of the monitored tasks have been executed.

The sequencing of these sub-functions shall also be described (which is basically in the order in which they are presented above with the exception of the service actions which are independently executed by the modules/tasks which are put under the supervision of PFM).

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_MC_PFM	1	0...FFH	0...255	1	[-]
Number of PFM tasks					
NC_NR_MC_PFM_TCC	1	0...FFH	0...255	1	[-]
Size of arrays TCC_MC_PFM_TOG[] and NC_TCC_MC_PFM_TOG_MAX[]					
NC_TCC_MC_PFM_TOG_MAX[NC_NR_MC_PFM_TCC]	1	0...FFH	0...255	1	[-]
Array containing the maximal number of communication cycles for the MU to perform the timely toggling of PFM bits for PFM tasks 0...NC_NR_MC_PFM_TCC					
NC_TCC_MC_PFM_TOG_6_MAX	1	0...FFFFH	0...65535	1	[-]
Maximal number of communication cycles for the MU to perform the toggling of the PFM bit for task 6					
NLF_MC_PFM_SYN_BYTE_INI	1	0...7FH	0...127	1	[-]
Initial value for the PFM synchronisation byte used for toggling by the service actions					

12.8.1 Decoding of PFM byte sent by the MU used for monitoring of MU

FUNCTION DESCRIPTION:

General information:

This sub-function describes how the "raw" data received from the MU (via the infrastructure interface) are pre-processed for PFM on the MC.

Note that this sub-function deals with that instance of the synchronisation byte which is only evaluated to detect whether the MU is still processing its PFM functions properly.

Description:

The function incorporates the following requirements:

G28 Decoding of PFM byte sent by the MU used for monitoring of MU

R28_1: In PREDRIVE, i.e., STATE_MC = NC_STATE_MC_PRDR, in case no communication error was detected (i.e., LV_MC_COM_ERR = 0 and its complement LV_MC_COM_ERR_CPL = 1, checked using the RAM check service actions):
R28_1.1: the "raw" PFM data sent by the MU are obtained by calling the action ACTION_INFR_McGetPfmByte() and the return value shall be stored in the variable LF_MC_RCV_PFM_BYTE_MON;
R28_1.2: since the MU always sends a synchronisation byte as the first PFM byte, bits 0...6 of LF_MC_RCV_PFM_BYTE_MON shall be copied to the synchronisation byte on the MC, LF_MC_PFM_SYN_BYTE_MON.

R28_2: In NORMAL, i.e., STATE_MC = NC_STATE_MC_NORM, in case no communication error was detected (i.e., LV_MC_COM_ERR = 0 and its complement LV_MC_COM_ERR_CPL = 1, checked using the RAM check service actions):
R28_2.1: the "raw" PFM data sent by the MU are obtained by calling the action ACTION_INFR_McGetPfmByte() and stored in the variable LF_MC_RCV_PFM_BYTE_MON.
R28_2.2: the change marker LF_MC_PFM_SYN_BYTE_CHG is saved in LF_MC_PFM_SYN_BYTE_CHG_OLD.
R28_2.3: depending on the value of the MSB (Most Significant Bit) of LF_MC_RCV_PFM_BYTE_MON, two cases arise:
R28_2.3.1: **MSB = 0** means the MU has sent an (updated) *synchronisation byte* as bits 0...6 of LF_MC_RCV_PFM_BYTE_MON.
R28_2.3.1.1: the change marker LF_MC_PFM_SYN_BYTE_CHG is updated, setting bit *k* if bit *k* in LF_MC_PFM_SYN_BYTE_MON <> bit *k* in LF_MC_RCV_PFM_BYTE_MON, and clearing bit *k* otherwise;
R28_2.3.1.2: bits 0...6 of LF_MC_RCV_PFM_BYTE_MON shall be copied to LF_MC_PFM_SYN_BYTE_MON;
R28_2.3.1.3: LF_STATE_ABC_PFM_MU shall not be updated.
R28_2.3.2: **MSB = 1** means the MU has sent a *status byte* of the PFM anti-bounce counters on the MU: bit 0 <= *k* <= 6 in LF_MC_RCV_PFM_BYTE_MON being set indicates that the value of the anti-bounce counter for PFM task *k* on the MU is strictly greater than 0.
R28_2.3.2.1: the change marker LF_MC_PFM_SYN_BYTE_CHG shall be reset to 0;
R28_2.3.2.2: bits 0...6 of LF_MC_RCV_PFM_BYTE_MON shall be copied to LF_STATE_ABC_PFM_MU.
R28_2.3.2.3: LF_MC_PFM_SYN_BYTE_MON shall not be updated.

Note:
Requirement R28_2.3.2.1 is based on the following equivalence:

any of the two "change markers" LF_MC_PFM_SYN_BYTE_CHG and LF_MC_PFM_SYN_BYTE_CHG_OLD is equal to 0

if and only if

the PFM byte from the corresponding communication (either the current one or the one before that) was a status byte.

This holds because the MU sends a PFM synchronisation byte *only if* it was actually updated, compared to the previously sent synchronisation byte: at least one bit must have changed and hence, in turn, at least one set bit in the corresponding change marker would have to be set on the MC.

Application conditions:

Initialisation:	*LF_MC_RCV_PFM_BYTE_MON = 0;* *LF_MC_PFM_SYN_BYTE_MON = 0;* *LF_STATE_ABC_PFM_MU = 0;* *LF_MC_PFM_SYN_BYTE_CHG = 0;* *LF_MC_PFM_SYN_BYTE_CHG_OLD = 0;* *CTR = 0;*
Recurrence:	*The **initialisation** shall be **executed after each power-up or reset.*** *The **function itself** shall be **executed after the modules** "Communication monitoring in PREDRIVE state" or "Communication monitoring in NORMAL state", depending on the current state* (i.e., synchronously to the communications between MC and MU).
Activation:	*((STATE_MC = NC_STATE_MC_PRDR) **OR*** *(STATE_MC = NC_STATE_MC_NORM)) **AND*** *(LV_MC_COM_ERR = 0)* (active in both PREDRIVE and NORMAL, but only if no communication error was detected)
Deactivation:	*Otherwise*

12.8.2 Decoding of PFM byte sent by the MU used for toggling by the PFM service actions

FUNCTION DESCRIPTION:

General information:

This sub-function describes how the "raw" data received from the MU (via the infrastructure interface) are pre-processed for PFM on the MC when used for the PFM service actions.

The function shall be executed **synchronously to, but before** the fastest monitored task: in case there has been an update of the synchronisation byte just before the execution of the monitored task, the PFM service actions will use the updated value of the synchronisation byte when determining the relevant bits in the PFM response LF_MC_RESP_PFM sent to the MU.

Since this function potentially runs asynchronously to the communications between MC and MU, it is necessary to perform the necessary communication monitoring (checksum, header-byte) specifically inside this function. In case an error is detected, no fault reaction shall be carried out, only the received data shall be ignored: eventually the "standard" communication monitoring will be performed and will detect the error and perform the defined reaction.

Description:

The function incorporates the following requirements:

G60 Decoding of PFM byte sent by the MU used for toggling by the PFM service actions

R60_1: In NORMAL, i.e., if MC_STATE = NC_STATE_MC_NORM:
R60_1.1: for **communication checksum monitoring**, the action ACTION_INFR_McChkCks() shall be called returning the flag LV_MC_CKS_OK_PFM (which is set if the checksum was correct);
R60_1.2: in case of a checksum error:

R60_1.2.1: no further communication monitoring shall be performed;
R60_1.2.2: no update of LF_MC_RCV_PFM_BYTE_TOG shall be performed;
R60_1.3: in case of correct checksum, for **communication header-byte monitoring**, the action ACTION_INFR_McGetHd() shall be called returning the header byte STATE_MU_TMP_PFM sent by the MU in the latest communication.
R60_1.4: If STATE_MU_TMP_PFM <> NC_STATE_MC_NORM, no update of LF_MC_RCV_PFM_BYTE_TOG shall be performed.
R60_1.5: If STATE_MU_TMP_PFM = NC_STATE_MC_NORM, the action ACTION_INFR_McGetPfmByte() shall be called and the return value shall be stored in the variable LF_MC_RCV_PFM_BYTE_TOG.
R60_1.6: Depending on the value of the MSB (Most Significant Bit) of LF_MC_RCV_PFM_BYTE_TOG, two cases arise:
R60_1.6.1: **MSB = 0** means the MU has sent an (updated) *synchronisation byte* as bits 0...6 of LF_MC_RCV_PFM_BYTE_TOG, which shall hence be copied to LF_MC_PFM_SYN_BYTE_TOG;
R60_1.6.2: **MSB = 1** means the MU has sent a *status byte* of the PFM anti-bounce counters on the MU, hence no update of LF_MC_PFM_SYN_BYTE_TOG shall be performed.

Application conditions:

Initialisation: *LF_MC_PFM_SYN_BYTE_TOG = NLF_MC_PFM_SYN_BYTE_INI;*
LF_MC_RCV_PFM_BYTE_TOG = 0;
STATE_MU_TMP_PFM = 0;
LV_MC_CKS_OK_PFM = 0;
Recurrence: *The **initialisation** shall be **executed after each power-up or reset.***
*The **function itself** shall be **executed synchronously to but before the fastest monitored task*** (i.e., potentially asynchronously to the communications between MC and MU).
Activation: *(STATE_MC = NC_STATE_MC_NORM)* (only active in NORMAL)
Deactivation: *Otherwise*

12.8.3 PFM service actions

FUNCTION DESCRIPTION:

General information:

This sub-function describes the services to supervise the program flow in relevant modules/tasks (in particular those of process monitoring).

Description:

The function incorporates the following requirements:

G29 PFM service actions

In general, PFM works by providing "markers" in the form of actions to be called at certain points in the supervised modules/tasks, thus enabling the PFM function to analyse the correctness of the program flow inside those modules/tasks.

In the following, let *k* always range over 0 to 6, with the intended meaning that *k* corresponds to one of the 7 different PFM tasks, and let *j* range over 0 to 29, for maximally 30 "inspection points" spread over (the modules in any of) the 7 tasks.

The PFM inspection for a fixed task *k* works as follows.

There is a *PFM control byte* STATE_MC_PFM_CTL_BYTE[*k*] which, before the first PFM synchronisation byte is transmitted from the MU to the MC (in the PREDRIVE communication), is initialised to *nc_ctl_ini* (one initial value for the control bytes of all tasks).

The PFM service actions ACTION_ECM3_Service*j*TaskPfm(*k*) and ACTION_ECM3_ServicePfm(*k*) subsequently manipulate STATE_MC_PFM_CTL_BYTE[*k*], allowing the PFM function to track the program flow in the supervised tasks.

The idea behind the actions ACTION_ECM3_Service*j*TaskPfm(*k*) is to record the correct sequencing inside a task: STATE_MC_PFM_CTL_BYTE[*k*] is manipulated such that

STATE_MC_PFM_CTL_BYTE[*k*] = *nc_ctl_end_k*
if and only if
all "inspection points" in the task were reached in their correct sequence *j* = 0,1,2,...,*max*
(*max* <= 29 and depends on the supervised task)

This yields the following error cases:

- **"inspection points" were reached in an incorrect order (e.g., 0,1,3,2,...),**
- **"inspection points" were skipped (e.g., 0,2,3,...),**
- **"inspection points" were reached repeatedly (e.g., 0,1,1,2,...),**
- **or arbitrary combinations thereof.**

In all of these cases, the control byte shall have a value different from *nc_ctl_end_k*, indicating an incorrect program flow in task *k*.

The action ACTION_ECM3_ServicePfm(*k*) shall implement a "*end of task*" marker by checking the current value of STATE_MC_PFM_CTL_BYTE[*k*] for equality with the expected end value *nc_ctl_end_k*.

This implies that this action must be executed *after all "inspection point" actions of task k have been executed,* i.e., it must be *the last PFM-relevant activity of the task k* under supervision; usually, it really is the last activity before the end of the task.

If the test is positive, i.e., all "inspection points" inside task *k* have been reached exactly once and in the correct sequence, bit *k* in the *PFM response* LF_MC_RESP_PFM shall be set to the negated value of bit *k* in the current synchronisation byte LF_MC_PFM_SYN_BYTE_TOG. Consequently, since the PFM response will eventually be transmitted to the MU, the MU will receive a bit which is toggled when compared to its original value in the synchronisation byte as sent from the MU.

If the test is negative, bit *k* in the synchronisation byte LF_MC_PFM_SYN_BYTE_TOG shall simply be copied to bit *k* in the PFM response (since the MU expects to receive this bit in its toggled state, this will eventually prompt a reaction by the MU).

Regardless of the test result, the control byte for task *k* shall be re-initialised to its start value *nc_ctl_ini* in order to prepare a new execution of task *k*.

Note: the common start value *nc_ctl_ini* and the individual end values *nc_ctl_end_k* for each of the control bytes are configurable so that they can be adapted to project-specific needs and NOT declared here.

Note: the PFM response is exclusively determined from the instance of the PFM synchronisation byte LF_MC_PFM_SYN_BYTE_TOG acquired synchronously with the fastest monitored task since only this instance should record the effect of any monitored modules being executed.

Description of ACTION_ECM3_Service(j)TaskPfm

Description:

Syntax:	ACTION_ECM3_Service(j)TaskPfm(IN <k>)
Parameter (IN):	k task number
Short description:	This action manipulates the control byte STATE_MC_PFM_CTL_BYTE[*k*] of task *k* in order to indicate the fact that the *j*-th "inspection point" in the task has been reached

Description of ACTION_ECM3_ServicePfm

Description:

Syntax:	ACTION_ECM3_ServicePfm(IN <k>)
Parameter (IN):	k task number
Short description:	This action provides an "*end of task*" marker, see formula section.

Formula section:

```
IF (STATE_MC_PFM_CTL_BYTE[k] = nc_ctl_end_k)
THEN /* all "inspection points" reached correctly, stored inverted bit in PFM response */
        LF_MC_RESP_PFM.k = invert(LF_MC_PFM_SYN_BYTE_TOG.k)
ELSE /* wrong program flow, store un-inverted bit in PFM response */
        LF_MC_RESP_PFM.k = LF_MC_PFM_SYN_BYTE_TOG.k
FI
/* Reset control byte for task k */
STATE_MC_PFM_CTL_BYTE[k] = nc_ctl_ini;
```

Application conditions:

Initialisation: /* Set control bytes to their common start value */

```
FOR k = 0 TO NC_NR_MC_PFM
DO
        STATE_MC_PFM_CTL_BYTE[k] =      nc_ctl_ini;
OD
```

Recurrence: *The **initialisation** shall be **performed after each power-up or reset**.*
*The **actions** shall be **called from inside the supervised modules**.*

Activation: -

Deactivation: -

12.8.4 Monitoring of timely updates of the synchronisation byte by the MU

FUNCTION DESCRIPTION:

General information:

This sub-function describes the mechanism by which the MC monitors whether the MU sends, in the corresponding intervals of time, synchronisation bytes where specific bits have been toggled at the right rate.
In this way it is possible to detect whether the MU is still running in "real-time" compared to the MC (in some ways, this is the dual of PFM which serves the purpose to detect whether the modules/tasks of process monitoring still run in "real-time").

Description:

The function incorporates the following requirements:

G30 Monitoring of timely updates of the synchronisation byte by the MU

R30_1: From the point of view of the MC, the MU is supposed to send updated synchronisation bytes at certain rates; the rate is determined by the bit toggling function on the MU, additionally influenced by the communication recurrence.

R30_2: In order to check whether this toggling process on the MU still works correctly, there is the array TCC_MC_PFM_TOG[] (of size 6) plus TCC_MC_PFM_TOG_6 (for PFM task 6 which, due to its low recurrence, needs higher resolution), on the MC, for counting communication cycles between two updates of the corresponding bit in the synchronisation bytes sent by the MU.

R30_3: For each of the counters, there is a corresponding maximal value contained in the array NC_TCC_MC_PFM_TOG_MAX[], plus NC_TCC_MC_PFM_TOG_6_MAX, which is used for detecting whether an update of the synchronisation byte by the MU was performed too late.

Application conditions:

Initialisation: /* Initialise the required data */

```
FOR CTR = 0 TO NC_NR_MC_PFM_TCC
DO
    TCC_MC_PFM_TOG[CTR] = 0;
OD
TCC_MC_PFM_TOG_6 = 0;
```

Recurrence: *The **initialisation** shall be **performed after each power-up or reset.** The **formula section** shall be **executed after the PFM sub-function "Decoding of PFM byte sent by MU used for monitoring of MU."***

Activation: *(STATE_MC = NC_STATE_MC_NORM)* (only performed in NORMAL after the PFM byte sent from the MU has been decoded; in PREDRIVE, the check does not make sense since there, the MU sends a synchronisation byte for the first time)

Deactivation: -

Formula section:

```
FOR CTR = 0 TO 5
DO
    /* Has the bit in the synchronisation byte been updated? */
```

```
        IF (LF_MC_PFM_SYN_BYTE_CHG.CTR = 1)
    THEN /* Yes: check value of test cycle counter */
            IF (TCC_MC_PFM_TOG[CTR] <
                NC_TCC_MC_PFM_TOG_MAX[CTR])
            THEN /* Bit toggled in time: decrease corresponding anti-bounce counter; reset test
            cycle counter*/
                ACTION_ECM3_McDecAbc(
                        IN      NC_ABC_MC_PFM_[CTR]_TOG);
                TCC_MC_PFM_TOG[CTR] = 0;
            ELSE /* Bit not toggled in time: debounced fault reaction; reset test cycle counter */
                ACTION_ECM3_McIncAbc(
                    IN      NC_ABC_MC_PFM_[CTR]_TOG,
                    IN      NC_ERR_COD_MC_PFM_[CTR]_TOG);
                TCC_MC_PFM_TOG[CTR] = 0;
            FI
      /* Bit not updated; has the test cycle counter reached its maximal value? */
    ELSEIF (TCC_MC_PFM_TOG[CTR] ≥
                        NC_TCC_MC_PFM_TOG_MAX[CTR])
      THEN /* Yes: debounced fault reaction; reset test cycle counter */
            ACTION_ECM3_McIncAbc(
                IN      NC_ABC_MC_PFM_[CTR]_TOG,
                IN      NC_ERR_COD_MC_PFM_[CTR]_TOG);
            TCC_MC_PFM_TOG[CTR] = 0;
    ELSE /* No: increase test cycle counter */
            TCC_MC_PFM_TOG[CTR] = TCC_MC_PFM_TOG[CTR] +1;
      FI
OD
/* Perform the same algorithm for task 6, using TCC_MC_PFM_TOG_6 and
NC_TCC_MC_PFM_TOG_6_MAX */
```

12.8.5 Test MU reaction to wrong PFM responses

General information:

If a correctly working MU receives a wrong PFM response (i.e., at least one bit in the PFM response sent by the MC is different from the corresponding bit in the synchronisation byte kept on the MU), it performs a debounced fault reaction: it increases its corresponding anti-bounce counter and, if the maximal value for that counter is reached, triggers a fault reaction. This sub-function aims at testing this mechanism by, under the appropriate circumstances, *deliberately* sending a wrong PFM response to the MU and the checking the eventual reaction to the wrong response. This presupposes the presence of a mechanism on the MC to "notice" the fact that the MU has increased a PFM anti-bounce counter after a wrong PFM response. Furthermore, the robustness of the system must not be affected by a deliberately wrong PFM response, in particular such a wrong answer must not trigger a fault reaction.
For implementing this mechanism, it is crucial that the MU, whenever its synchronisation byte has not changed, sends status bytes establishing which of its PFM anti-bounce counters have value 0 and which do not. It follows that those with value 0 can be tested because (very importantly) there, a wrong answer is *guaranteed* not to trigger a fault reaction (for the others, this need not be true) and, equally important, the next time a status byte is sent, it can be checked whether the anti-bounce counter is still equal to 0 (meaning that the MU has *not* reacted to the wrong response) or not (meaning that the MU has increased the corresponding anti-bounce counter and hence reacted to the wrong response).
Note that this check cannot detect an anti-bounce counter on the MU which is stuck to a value between 0 and the maximal value: the status byte will always indicate that the anti-bounce counter

is different from 0 so that there will never be a deliberately wrong PFM response for the corresponding task.

Description:

The function incorporates the following requirements:

G31 Test MU reaction to wrong PFM responses

The algorithm is described in the formula section the following points should be pointed out (note that these considerations do not take into account the presence of communication errors).

When activated for the first time (after the first NORMAL communication), the initial status byte of the MU anti-bounce counters has just been received: in the PREDRIVE communication, the MU always sends the initial synchronisation byte; consequently, in the first NORMAL communication, the current status of the PFM anti-bounce counters on the MU is transmitted to the MC.

Moreover, LF_MC_PFM_SYN_BYTE_CHG_OLD currently has value 0 since, after the PREDRIVE communication, LF_MC_PFM_SYN_BYTE_CHG is initialised to 0 and this value is stored in LF_MC_PFM_SYN_BYTE_CHG_OLD after the first NORMAL communication.

Consequently, the first condition is false for every *k*, and nothing happens until the test of LF_STATE_ABC_PFM_MU. If bit *k* of LF_STATE_ABC_PFM_MU is set, we do nothing (anti-bounce counter on MU not equal to 0, cannot risk sending a wrong response as this might trigger a fault reaction). If bit *k* is not set, this means that the anti-bounce counter on the MU for task *k* has value 0 so we activate sending wrong responses; this will result in a wrong bit in the PFM response in the next communication (in this case: the second NORMAL communication) sent out after the post-processing function has been executed. So a set bit in LF_MC_WRG_RESP_PFM_ACT indicates that a wrong response was sent or will be sent, depending on when the value is observed.

Now assume that normal operation has continued for some time. When the function gets activated, the same test as described above is performed, viz., it is checked whether the corresponding bit in LF_MC_WRG_RESP_PFM_ACT is set, which implies that in the previous communication, a wrong response has been sent out, and whether in the previous communication an updated synchronisation byte was transmitted by the MU; so the status of an MU anti-bounce counter is only checked after a bit was toggled on the MU and sent to the MC.

This test being successful implies that in the current communication, a status byte was sent and so it can be checked whether the status of the MU anti-bounce counter for a task *k* is not equal to 0.

If this is the case the previously sent wrong response was detected by the MU (since a wrong response is only sent out if the anti-bounce counter is equal to 0) and the corresponding anti-bounce counter on the MC can be decreased. However, if the value of the MU anti-bounce counter is equal to 0, this means that the previously sent wrong response was not detected and consequently, the corresponding anti-bounce counter on the MC must be increased, eventually leading to a debounced fault reaction.

If this increase of the MC anti-bounce counter has not yet triggered a fault reaction, it is then checked whether the MU anti-bounce counter status indicates a value of 0. If that is the case (i.e., either when really no error is present on the MU or when previously a wrong answer was sent out but the MU has not (yet) reacted to the wrong response), the flag corresponding bit in LF_MC_WRG_RESP_PFM_ACT is set in order to send a wrong PFM response.

Note that this implies sending a wrong PFM response until one communication after the MU next updates its synchronisation byte and sends it to the MC because before that it is not possible to de-activate sending of wrong responses, unless the MU anti-bounce counter is no longer equal to 0 which also implies that the MC must have updated its synchronisation byte.

Sending a wrong response for any task is done for the first time as soon as the corresponding status byte received from the MU indicates that the MU anti-bounce counter has value 0. Due to its slow recurrence and the described algorithmic pre-condition, a long time has to elapse before the wrong response test for task 6 can be performed (also due to the fact that the corresponding anti-bounce counter is initialised to a non-zero value to speed up fault detection for this task). On the other hand, the reset counter on the MC heals after a certain time period which should last longer than the duration all possible diagnostic tests inside Processor Monitoring. Consequently, the healing can only be performed after the wrong response test for PFM task 6 has been performed at least once. This is fact shall be indicated to the function "Healing of reset counter and error code" (G2) by the flag LV_MC_PFM_WRG_RESP_ACT_6: it shall be set when sending a wrong PFM response for task 6 is activated.

Application conditions:

Initialisation: /* Deactivate sending wrong PFM responses for all tasks */
LF_MC_WRG_RESP_PFM_ACT = 0;
LV_MC_PFM_WRG_RESP_ACT_6 = 0;
Recurrence: *The* ***initialisation*** *shall be* ***performed after each power-up or reset****.*
The ***formula section*** *shall be* ***executed after the PFM sub-function "Decoding of PFM byte sent by MU used for monitoring of MU."***
Activation: *(STATE_MC = NC_STATE_MC_NORM)* (only performed in NORMAL after the PFM byte sent from the MU has been decoded)
Deactivation: *Otherwise*

Formula section:

```
FOR CTR = 0 TO NC_NR_MC_PFM
DO /* Check whether sending of wrong responses for task k is active and whether the last communication, the bit for task k was toggled (=in the last communication, a synchronisation byte was sent so that now a status byte must have been received) */
        IF ((LF_MC_WRG_RESP_PFM_ACT.CTR = 1) AND
        (LF_MC_PFM_SYN_BYTE_CHG_OLD.CTR = 1))
        THEN /* Has MU-anti-bounce counter reacted to wrong response? */
                IF (LF_STATE_ABC_PFM_MU.CTR = 1)
                THEN /* Yes: decrease MC anti-bounce counter */
                        ACTION_ECM3_McDecAbc(
                                IN      NC_ABC_MC_PFM_[CTR]_RESP);
                        IF (CTR = 6)
                        THEN /* Sending of wrong responses for task 6 has been performed at least once: indicate this to the function "Healing of reset counter and error code" (G2) */
                                LV_MC_PFM_WRG_RESP_ACT_6 = 1;
                        FI
                ELSE /* No: increase MC anti-bounce counter */
                        ACTION_ECM3_McIncAbc(
                                IN      NC_ABC_MC_PFM_[CTR]_RESP,
                                IN      NC_ERR_COD_MC_PFM_[CTR]_RESP);
                FI
        FI
        /* Check if sending of wrong responses can be activated/stay active: is the MU anti-bounce counter equal to 0? */
        IF (LF_STATE_ABC_PFM_MU.CTR = 0)
        THEN /* Yes: activate sending of wrong responses */
                LF_MC_WRG_RESP_PFM_ACT.CTR = 1;
        ELSE /* No: deactivate sending of wrong responses; this also applies if previously a deliberately wrong answer was sent */
                LF_MC_WRG_RESP_PFM_ACT.CTR = 0;
        FI
OD
```

12.8.6 Post-processing of PFM response before communications

General information:

This sub-functions sets the value of the PFM response to be sent to the MU in the following (NORMAL) communication.

Description:

The function incorporates the following requirements:

G32 Post-processing of PFM response before communications

Before the new PFM response can be sent to the MU, some post-processing of its current value might be necessary, viz., if sending of wrong PFM responses is active for a certain task, it must be ensured that the effect of the service actions (which, if executed properly, would result in a correct PFM response for the corresponding task) is overridden and explicitly a wrong bit in the PFM response is set.

By the way the checking mechanism on the MU works, this means that the bit in the PFM response has to be *equal* to the value of the respective in the current synchronisation byte on the MC (as the MU expects this bit to be sent back with its value *toggled*).

Finally, the (potentially modified) PFM response LF_MC_PFM_RESP has to be sent to the infrastructure so that it can be transmitted to the MU when the next communication is triggered.

To ensure this behaviour is precisely the task of this sub-function.

Application conditions:

Initialisation: *LF_MC_RESP_PFM = 0;*
Recurrence: *The **initialisation** shall be **performed after each power-up or reset**.*
*The **formula section** shall be **executed just before the next communication is triggered**.*
Activation: *(STATE_MC = NC_STATE_MC_NORM)* (active in NORMAL state)
Deactivation: *Otherwise*

Formula section:

```
FOR CTR = 0 TO 6
DO /* Shall a wrong response bit shall be sent for task k? */
    IF (LF_MC_WRG_RESP_PFM_ACT.CTR = 1)
        THEN /* Yes: set value of the bit in PFM response to the value of the corresponding bit in
        the current synchronisation byte used for toggling by PFM service actions */
                LF_MC_RESP_PFM.CTR = LF_MC_PFM_SYN_BYTE_TOG.CTR;
    FI /* Otherwise leave bit in PFM response as it is */
DO
ACTION_INFR_McSendPfmResp(IN        LF_MC_RESP_PFM);
```

12.9 MC Function-specific Instruction Set Test

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IDX_NR_RESP_MC_FCT_SPC_IST	-	0...1FH	0...31	1	[-]
Local variable: answer number of answer from level 2'					
ABC_RCV_MC_FCT_SPC_IST	-	0...FH	0...15	1	[-]
Local variable: value of FS-IST anti-bounce counter on MU (obtained via infrastructure)					
ABC_LID_MC_FCT_SPC_IST	V	0...FH	0...15	1	[-]
Local variable: value of MU anti-bounce counter for FS-IST (obtained from ABC_RCV_MC_FCT_SPC_IST)					
IDX_NR_LID_MC_FCT_SPC_IST	-	0...1FH	0...31	1	[-]
Local variable: question number corresponding to MU question (IDX_TEST_RCV_MC_FCT_SPC_IST)					
IDX_NR	-	0...1FH	0...31	1	[-]
Local variable: index for decoding of MU question					
LV_MC_FCT_SPC_IST_ACT	-	0...1H	0...1	1	[-]
Local variable: flag indicating whether FS-IST is active in current communication cycle					
TCC_MC_FCT_SPC_IST_WRG_RESP	-	0...1FH	0...31	1	[-]
Local variable: test cycle counter for sending wrong FS-IST answers to MU					
LV_MC_FCT_SPC_IST_WRG_RESP_ACT	V	0...1H	0...1	1	[-]
Local variable: flag indicating whether sending of wrong FS-IST answers is active or not					
ABC_LID_MC_FCT_SPC_IST_OLD	-	0...FH	0...15	1	[-]
Local variable: saved value of MU anti-bounce counter for FS-IST from preceding communication cycle					
IDX_NR_LID_MC_FCT_SPC_IST_PRED	-	0...1FH	0...31	1	[-]
Local variable: predicted question number (for monitoring of question sent by MU)					
IDX_NR_LID_MC_FCT_SPC_IST_OLD	-	0...1FH	0...31	1	[-]
Local variable: saved question number from previous communication cycle					
IDX_NR_RESP_MC_FCT_SPC_IST	-	0...1FH	0...31	1	[-]
Local variable: answer number of answer from level 2'					
TEST_REC_IDX_MON2	V/O	0...1FH	0...31	1	[-]
Question number ("test index") exported to the copy of process monitoring					
IDX_TEST_RCV_MC_FCT_SPC_IST	-	0...1FH	0...31	1	[-]
Local variable: question sent by MU (obtained via infrastructure; to be decoded into question number)					
IDX_NR_ERR_MC_FCT_SPC_IST	V	0...FFH	0...255	1	[-]
Local variable: "marker" of index for which second calculation of copy of process monitoring failed					

Input data:

STATE_MC	NC_STATE_MC_PRDR	NC_STATE_MC_NORM	NC_ABC_MC_NOT_DEC
NC_ERR_COD_MC_NOT_DEC	NC_ABC_MC_WRG_RESP	NC_ERR_COD_MC_WRG_RESP	NC_ABC_MC_WRG_IDX
NC_ERR_COD_MC_WRG_IDX	LV_MC_DR_OFF	LV_MC_DR_OFF_CPL	RESP_MON2
ID_RESP_MON2	LDPM_TEST_REC_IDX_MON2		

Import actions:

ACTION_ECM3_McIncAbc(IN <>, IN<>)
ACTION_ECM3_McDecAbc(IN <>)
ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)
ACTION_ECM3_McReadChkCpl(OUT <>, IN <>, IN <>, IN <>)
ACTION_ECM3_McReadChkState(OUT <>)
ACTION_INFR_McGetFsistIdx(OUT <>)
ACTION_INFR_McGetFsistAbcVal(OUT <>)
ACTION_INFR_McSendFsistResp(IN <>)

ACTION_INFR_McSendDrOff()

FUNCTION DESCRIPTION:

General information:

The function-specific instruction set test (FS-IST) complements the program flow monitoring (PFM) functionality in an effort to obtain as complete a monitoring of the proper functioning of the main controller (MC) as possible. The goal of the FS-IST is to check whether the MC is capable of correctly performing the calculations necessary for the most important safety-relevant functions in the *process monitoring* ("VDA level 2") - hence the name *function-specific* instruction-set test: only computations with relevance for the monitoring concept (the ‘function’ under consideration) are tested, as opposed to a complete instruction-set test.
As already stated, the FS-IST checks whether certain safety-relevant calculations are performed correctly. These so-called test calculations are performed by a copy of the relevant modules of process monitoring (so as not to ‘disturb’ the action process monitoring by the test calculations); these copied level 2 modules constitute the so-called copy of process monitoring, or level 2’. Although conceptually part of processor monitoring, the level 2’ are in practise more closely associated to process monitoring than to processor monitoring: each change of a process monitoring module a copy of which is part of level 2’ has to be carried out also for the corresponding level 2’ module.
Since the test calculations are also performed on the MC, i.e., by the same processor that performs the actual (level 2) calculations, the monitoring unit (MU) is used to initiate the FS-IST and check whether the calculations arrive at the correct results (otherwise no reasonable way of detecting faults is possible).
Concretely, the FS-IST is performed in the following way (there are some points to note about the timing which will be explained after the description of the basic algorithm):

- on the MU, there exists an array of test indices, or questions; these questions are simply bit patterns which are cyclically sent to the MC, i.e., at the beginning, the first entry of the array is sent, then the second, and so on (in the error-free case, the case of errors will be described later); for these question, there are corresponding answers, or responses, which are also stored in an array on the MU
- these test indices, once they have been received by the MC-part of processor monitoring, are then sent to the level 2’ via an interface which, depending on the question from the MU, determines the starting values (i.e., values of inputs, etc.) for the level 2’ modules;
- using this chosen starting values, level 2’ calculates according to the modules, depending on the ability of the processor, yielding a specific actual output value that is then forwarded to the MC-part of processor monitoring, again via the interface, together with the expected result for this particular test index;
- the MC-part of processor monitoring then determines (by comparing the actual with the expected output value) whether the test calculation was performer successfully before sending a response, or answer, back to the MU; note that this answer is also influenced by other factors (e.g., if a fault by the MC-side of processor monitoring was detected, this answer is forced to be false from the expected answer on the MU)
- the MU receives the answer of the MC-part of processor monitoring and compares that to the entry in its array of expected results at the position corresponding to the sent test index;
- if this comparison has a positive result, and anti-bounce counter shall be decreased (without creating an underflow, i.e., if its value is already 0 nothing shall happen, see “G21 Debounce mechanism”) and the procedure shall continue as described before;
- if the comparison shows that a wrong answer has been received, an anti-bounce counter shall be increased; upon reaching its maximal value, the central fault reaction (“G1

central fault reaction”) shall be triggered; if the maximal value is not reached when increasing the anti-bounce counter, the particular question that was wrongly answered shall be re-sent to the MC for as long as the anti-bounce has not reached 0, i.e., if a question was answered wrongly and the value of the anti-bounce counter has value 4, the same question is sent (at least) 4 times before the next question is sent.

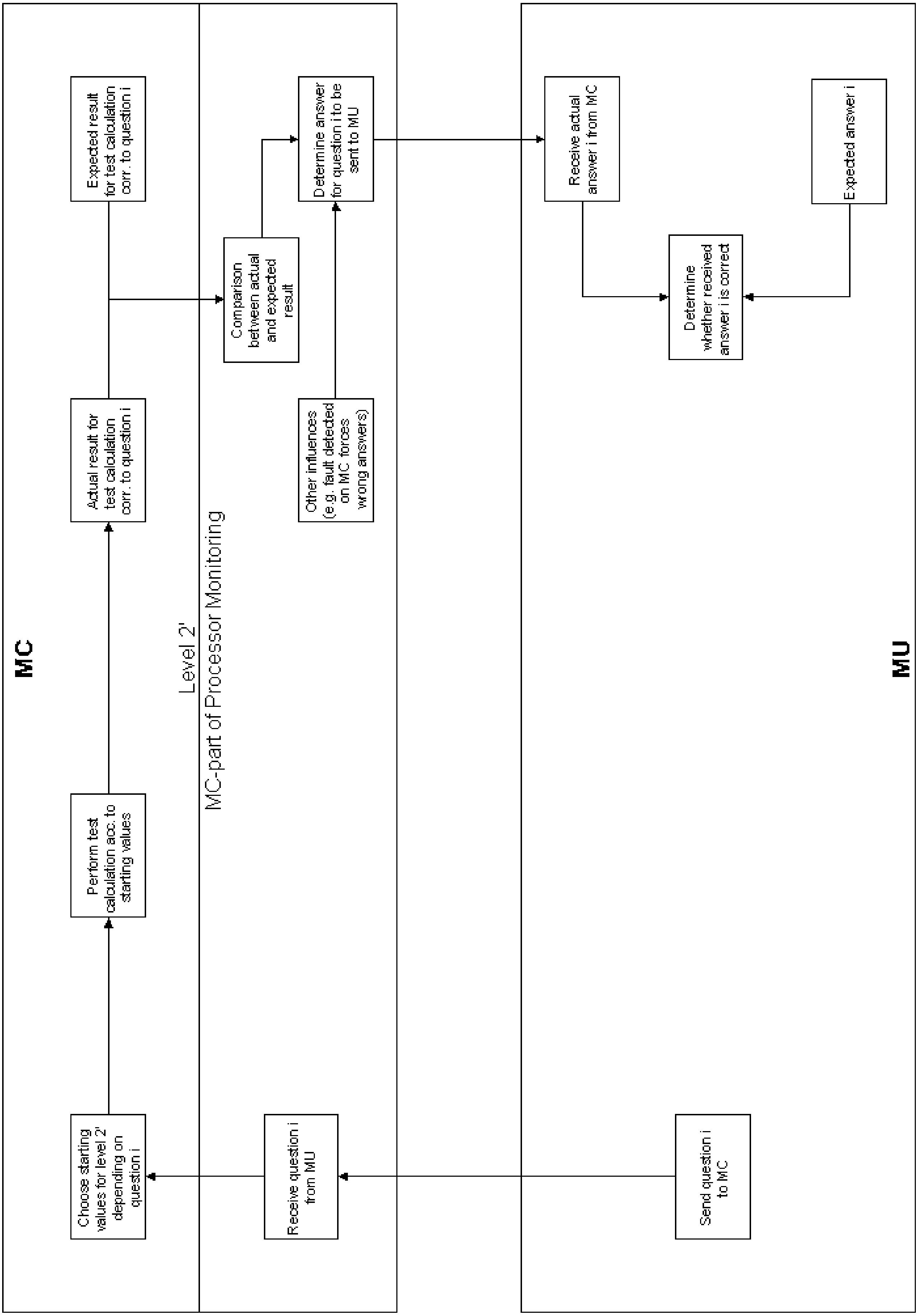
Note the possibility to distinguish between questions and question numbers: in this way, the algorithm for obtaining the new test index can be very simple, viz., the question number simply needs to be increased by with every correct answer (or set to if all possible numbers have been run through), while the actual bit patterns denoting the questions (to be found in a predefined array) can be chosen in such a way as to decrease (if not completely eliminate) the probability of corruption during the communication and also to be able to distinguish between question and answers in EMS and TCU mode of the MU. As stated before, some words regarding timing are necessary at this point:

- when the MU is used for monitoring of EMSs, the level 2’ answers are only checked every second communication cycle;
- when the MU is used for monitoring of TCUs, the level 2’ answers are checked every communication cycle;
- in NORMAL state, whenever the MU sends any question to the MC, the response received in that very communication (regardless of the chosen variant, EMS or TCU) is the answer corresponding to a previous question, i.e., in order to find out whether a question has been answered wrongly, the freshly received answer must be compared to the stored answer corresponding to the previous question;
- if an answer was wrong, the previously described timing situation leads to the fact that, before repeating the wrongly answered question, there will be one transmission of a new question (viz., in the communication where the wrong answer was received); for instance, assuming that both questions and question numbers are simply 0,1,2,... and question 3 is permanently answered wrongly by the MC, the following sequence of questions arises: 0,1,2,3,4,3,3,3... trigger “G1 Central fault reaction”, i.e., question 3 repeated until increasing the anti-bounce counter triggers a fault reaction; the answer to question 4 is simply ignored because the fact that the answer to question 3 was wrong only becomes apparent after question 4 has already been transmitted to the MC.

On the MC, for sending the response, there is a similar split into "answer" and "answer number" in order to facilitate the distinction between the responses from level 2' and for the MU.

Signal flow diagram:

A schematic overview of the FS-IST functionality is given in the following diagram (note that the picture only depicts the control flow, one should not forget that time passes when the communications are performed, the level 2’ calculations are performed etc.):


MC
Level 2'
MC-part of Processor Monitoring
MU
Choose starting values for level 2' depending on question i
Perform test calculation acc. to starting values
Actual result for test calculation corr. to question i
Expected result for test calculation corr. to question i
Comparison between actual and expected result
Determine answer for question i to be sent to MU
Other influences (e.g. fault detected on MC forces wrong answers)
Receive question i from MU
Receive actual answer i from MC
Expected answer i
Determine whether received answer i is correct
Send question i to MC

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_TEST_REC_IDX_MAX_MON2	1	0...1FH	0...31	1	[-]
(NC_TEST_REC_IDX_MAX_MON2-1) is the maximal value for the question number exported to the test calculations in level 2'					
NC_IDX_TEST_MC_FCT_SPC_IST[NC_TEST_REC_IDX_MAX_MON2]	-	0...3FH	0...63	1	[-]
Array of valid questions; compared to actual questions sent from MU					
NC_IDX_RESP_MC_FCT_SPC_IST[NC_TEST_REC_IDX_MAX_MON2]	-	0...3FH	0...63	1	[-]
Array of answers to be sent to MU					
NC_IDX_RESP_WRG_MC_FCT_SPC_IST	-	0...3FH	0...63	1	[-]
Constant for invalid answer					
NC_IDX_NR_WRG_MC_FCT_SPC_IST	-	0...1FH	0...31	1	[-]
Constant for invalid question number (used in the decoding processes)					
NC_TCC_MC_IST_WRG_RESP_REQ	-	0...1FH	0...31	1	[-]
Value of test cycle counter when sending of wrong answer should be requested					
NC_TCC_MC_IST_WRG_RESP_CHK	-	0...1FH	0...31	1	[-]
Value of test cycle counter when MU reaction to sending of wrong answer should be checked					

12.9.1 Pre-processing of the FS-IST data received from MU

FUNCTION DESCRIPTION:

General information:

This function pre-processes the data sent by the MU.

In PREDRIVE and NORMAL communications, the MU sends an FS-IST question and, only in NORMAL communications, additionally also the value of the anti-bounce counter for wrong FS-IST answers on the MU (it is not necessary to send that value in PREDRIVE because then its value is 0).

Description:

The function incorporates the following requirements:

G33 Pre-processing of the FS-IST data received from MU

The inputs to this function are IDX_TEST_RCV_MC_FCT_SPC_IST, the question as it was sent from the MU (via the infrastructure) and, in NORMAL communications, ABC_RCV_MC_FCT_SPC_IST, the value of the anti-bounce counter on the MU used for debouncing wrongly answered FS-IST questions.

Since the first MU anti-bounce counter value is only sent in NORMAL communications but the initial value of 0 is already known and used in PREDRIVE, it is necessary to use the local identifier ABC_LID_MC_FCT_SPC_IST to be used in all other MC FS-IST functions both in PREDRIVE and NORMAL; this variable is initialised to 0 before the PREDRIVE communication and then updated with the actual value sent by the MU after NORMAL communications.

As for the question sent by the MU, it is more efficient to instead use the corresponding *question number* (which is nothing but the appropriate index into the array NC_IDX_TEST_MC_FCT_SPC_IST[] of questions on the MC), in particular to simplify handing data to level 2'. Therefore, a decode step has to be taken after received the question on the MC: the received question IDX_TEST_RCV_MC_FCT_SPC_IST is simply compared to all entries in the array NC_IDX_TEST_MC_FCT_SPC_IST[] in order to find whether the transmitted question does indeed correspond to a known index on the MC, which is then stored in IDX_NR_LID_MC_FCT_SPC_IST; in order to detect when this decoding step fails, IDX_NR_LID_MC_FCT_SPC_IST is initialised to a specific value, NC_IDX_NR_WRG_MC_FCT_SPC_IST, indicating an invalid question number, and if no corresponding question number is found, this remains the value of IDX_NR_LID_MC_FCT_SPC_IST.

In this way, after both PREDRIVE and NORMAL communication(s), all data are always present for the other FS-IST functions.

Application conditions:

Initialisation: */* Initialise the value of the MU anti-bounce counter sent by the MU */*

ABC_RCV_MC_FCT_SPC_IST = 0;

/ Initialise the value of the MU anti-bounce counter on the MC */*

ABC_LID_MC_FCT_SPC_IST = 0;

/ Initialise question number for question sent by MU to wrong number */*

IDX_NR_LID_MC_FCT_SPC_IST = NC_IDX_NR_WRG_MC_FCT_SPC_IST;

/ Initialise index number */*

IDX_NR = 0;

/ Initialise question sent by MU */*

IDX_TEST_RCV_MC_FCT_SPC_IST = 0;

Recurrence: The **initialisation** shall be performed **after each power-up or reset**.

The **formula section** shall be **executed after the functions “Communication monitoring and evaluation of received data in PREDRIVE state” and “Communication monitoring and evaluation of received data in NORMAL state” have been finished**.

Activation: ((STATE_MC = NC_STATE_MC_PRDR) **OR**
(STATE_MC = NC_STATE_MC_NORM)) **AND**
"no errors during the evaluation following the communications"

Deactivation: Otherwise

Formula section:

/ Get question sent by MU from infrastructure */*

ACTION_INFR_McGetFsistIdx(*OUT* IDX_TEST_RCV_MC_FCT_SPC_IST);

/ Initialise local identifier for question number */*

IDX_NR_LID_MC_FCT_SPC_IST = NC_IDX_NR_WRG_MC_FCT_SPC_IST;

```
/* In NORMAL, change the value of the MU anti-bounce counter to the received value */
IF (STATE_MC = NC_STATE_MC_NORM)
THEN /* In PREDRIVE, MC value of MU anti-bounce counter is set in initialisation*/
    /* Get value of anti-bounce counter on MU from infrastructure */
    ACTION_INFR_McGetFsistAbcVal(OUT     ABC_RCV_MC_FCT_SPC_IST);
    ABC_LID_MC_FCT_SPC_IST = ABC_RCV_MC_FCT_SPC_IST;
FI
/* Check whether received question corresponds to a question number; if yes, set local identifier to the decoded question number */
FOR IDX_NR = 0 TO NC_TEST_REC_IDX_MAX_MON2 - 1
DO /* Check all defined questions */
    IF (NC_IDX_TEST_MC_FCT_SPC_IST[IDX_NR] =
    IDX_TEST_LID_MC_FST_SPC_IST)
    THEN /* Found the corresponding question number */
            IDX_NR_LID_MC_FCT_SPC_IST = IDX_NR;
    FI
OD
/* If no matching question number is found, IDX_NR_LID_MC_FCT_SPC_IST will have value NC_IDX_NR_WRG_MC_FCT_SPC_IST */
/* Has the number of the received question changed with this communication? */
IF (IDX_NR_LID_MC_FCT_SPC_IST <> TEST_REC_IDX_MON2)
THEN
    LV_MC_FCT_SPC_IST_ACT = 1;
FI /* Otherwise, LV_MC_FCT_SPC_IST_ACT remains unchanged */
```

12.9.2 Test of MU reaction to wrong FS-IST answers

FUNCTION DESCRIPTION:

General information:

This sub-function, under certain conditions, indicates (to the post-processing sub-function) a deliberate sending of wrong FS-IST answers to the MU, similar to PFM, with the goal to test whether the MU, after an appropriate delay, reacts to the received wrong answer by sending an increased anti-bounce counter and repeating the wrongly answered question.

This sub-function is executed after the pre-processing function.

Description:

The function incorporates the following requirements:

G34 Test of MU reaction to wrong FS-IST answers

After the data received from the MU have been decoded and pre-processed, the newly received information (in particular about the value of the FS-IST anti-bounce counter on the MU) is used to determine whether it is possible to send a deliberately wrong FS-IST answer in the current system state.

In addition to the anti-bounce counter information from the MU, this system state includes the value of a test cycle counter TCC_MC_FCT_SPC_IST_WRG_RESP counting the number of times the function gets activated. There are two significant values of this counter, viz., NC_TCC_MC_IST_WRG_RESP_REQ and NC_TCC_MC_IST_WRG_RESP_CHK: the former means that sending a wrong answer is requested (this will eventually prompt setting a flag influencing the post-processing sub-function to export a bad answer); the latter value means that it is time to check whether the MU has reacted to a previously sent deliberately wrong answer (one has to allow for some delay because the MU only sends a new question every second communication cycle; consequently, the two values must necessarily be different). If the counter reaches a certain number then it is re-started, eventually triggering a new cycle of another wrong answer, check MU reaction to that and so on.

The sub-function consists of several sequential sub-tasks:

1. find out whether the MU anti-bounce counter sticks to a non-zero value (deploying a saved copy of its previous value);
2. test whether the MU has reacted (i.e., increased the anti-bounce counter accordingly) if the test cycle counter has reached value NC_TCC_MC_IST_WRG_RESP_CHK; this test also checks whether the MU anti-bounce counter sticks to 0 (complementing the previous test);
3. initiate sending a deliberately wrong FS-IST answer if the test cycle counter has reached value NC_TCC_MC_IST_WRG_RESP_REQ;
4. increase the value of the test cycle counter (including re-starting the counter when it is has reached its end value).

These sub-tasks shall be performed in this order; note that the second and third task cannot be activated at the same time anyway because their activation conditions are that the test cycle counter reaches two distinct values. As a consequence, they will be specified in one common formula section.

Note that for sub-task 3, (potentially) initiating the sending of a wrong FS-IST answer to the MU, is only performed of the MU anti-bounce counter is equal to 0, much like in the corresponding PFM functionality: sending a deliberately wrong answer must not trigger fault reaction. As a consequence, if the test cycle counter indicates that the MU should have reacted to a wrong answer (sub-task 2), it is tested whether the MU anti-bounce counter has value 2 which corresponds to exactly one increase of the MU anti-bounce counter (caused by the *deliberately* wrong answer).

It should furthermore be noted that, if during sub-task 2, it is found that the anti-bounce counter does not have the expected value, a wrong answer shall again be sent (this is achieved by setting the flag LV_MC_FCT_SPC_IST_WRG_RESP_ACT) and the test cycle counter shall be decrease by 1; the latter action is done because in sub-task 4, the test cycle counter is *unconditionally* increase by 1 so in the mentioned error case, the counter will, after the increase, still have value NC_TCC_MC_IST_WRG_RESP_CHK, i.e., after the next activation of the FS-IST, it will again be checked whether the MU has reacted to the wrong answer.

Application conditions:

Initialisation: *[/* FS-IST to be activated */]*

```
/* FS-IST to be activated */
LV_MC_FCT_SPC_IST_ACT = 1;
/* Initialise saved value of MU anti-bounce counter */
ABC_LID_MC_FCT_SPC_IST_OLD = 0;
```

/ Initially deactivate sending of wrong responses */*

LV_MC_FCT_SPC_IST_WRG_RESP_ACT = 0;

/ Initially set test cycle counter to requesting a wrong response */*

TCC_MC_FCT_SPC_IST_WRG_RESP =
NC_TCC_MC_IST_WRG_RESP_REQ

Recurrence: The **initialisation** shall be **performed after each power-up or reset**.

The **formula section** shall be **executed after the FS-IST sub-function "Pre-processing of FS-IST data received from MU" has been finished.**

Activation: (LV_MC_FCT_SPC_IST_ACT = 1) **OR** ("MU configured for TCU mode")

Deactivation: Otherwise

Formula section:

```
/* Sub-task 1: checking whether the MU anti-bounce counter sticks to a non-zero value */
IF (ABC_LID_MC_FCT_SPC_IST > 0)
THEN /* Does the MU anti-bounce counter stick to a non-zero value? */
    IF (ABC_LID_MC_FCT_SPC_IST = ABC_LID_MC_FCT_SPC_IST_OLD)
    THEN /* Yes: increase MC anti-bounce counter */
        ACTION_ECM3_McIncAbc( IN    NC_ABC_MC_NOT_DEC,
                              IN    NC_ERR_COD_MC_NOT_DEC);
    ELSE /*Has the anti-bounce counter been decreased? */
        IF (ABC_LID_MC_FCT_SPC_IST < ABC_LID_MC_FCT_SPC_IST_OLD)
        THEN /* Yes: decrease MC anti-bounce counter */
            ACTION_ECM3_McDecAbc(IN    NC_ABC_MC_NOT_DEC);
        FI /* Otherwise: no action as this behaviour of the MU anti-bounce counter might
        be caused by a (really, as opposed to deliberately) wrong level 2' response */
    FI /* Note that sticking to 0 is tested later */
ELSE /* No, MU anti-bounce counter is equal to 0: decrease MC anti-bounce counter */
    ACTION_ECM3_McDecAbc(IN    NC_ABC_MC_NOT_DEC);
FI
/* Sub-task 2: does the test cycle counter indicate checking the MU reaction? */
IF (TCC_MC_FCT_SPC_IST_WRG_RESP = NC_TCC_MC_IST_WRG_RESP_CHK)
THEN /* Yes: check value of MU anti-bounce counter (should be equal to 2)*/
    IF (ABC_LID_MC_FCT_SPC_IST = 2)
    THEN /* MU anti-bounce counter has the expected value: decrease corresponding MC
    anti-bounce counter */
        ACTION_ECM3_McDecAbc(IN    NC_ABC_MC_WRG_RESP);
```

```
        ELSE /* MU anti-bounce counter does not have expected value: increase corresponding
        MC anti-bounce counter */
            ACTION_ECM3_McIncAbc(IN     NC_ABC_MC_WRG_RESP,
                                 IN     NC_ERR_COD_MC_WRG_RESP);
            /* Does the MU anti-bounce counter stick to 0? */
            IF (ABC_LID_MC_FCT_SPC_IST = 0)
            THEN /* Yes: re-activate sending of wrong answer, decrease test cycle counter
    */
                LV_MC_FCT_SPC_IST_WRG_RESP_ACT = 1;
                TCC_MC_FCT_SPC_IST_WRG_RESP =
                                    TCC_MC_FCT_SPC_IST_WRG_RESP – 1;
            FI
        FI
    FI
    /* Sub-task 3: request sending a wrong answer? */
    IF ((TCC_MC_FCT_SPC_IST_WRG_RESP = NC_TCC_MC_IST_WRG_RESP_REQ)
    THEN /* Yes: can we actually send a wrong answer (MU anti-bounce counter = 0)?*/
        IF (ABC_LID_MC_FCT_SPC_IST = 0)
        THEN /* Yes: set flag */
            LV_MC_FCT_SPC_IST_WRG_RESP_ACT = 1;
        ELSE /* No: try again in the next communication cycle */
            TCC_MC_FCT_SPC_IST_WRG_RESP =
                TCC_MC_FCT_SPC_IST_WRG_RESP – 1;
        FI
    FI
    /* Sub-task 4: increase test cycle counter (including re-starting the counter) */
    TCC_MC_FCT_SPC_IST_WRG_RESP =
                        (TCC_MC_FCT_SPC_IST_WRG_RESP + 1)
                MOD     NC_TEST_REC_IDX_MAX_MON2 + 4;
    /* Note that still LV_MC_FCT_SPC_IST_ACT = 1, i.e., the FS-IST is still active! */
```

12.9.3 Monitoring of FS-IST questions sent by MU

FUNCTION DESCRIPTION:

General information:

This sub-function monitors the sending of questions by the MU by comparing the question number corresponding to the actually sent question (in case it could be decoded) with a *predicted* question number (obtained as a function of the previous question number, plus the current and the saved state of the MU anti-bounce counter). In case of a discrepancy, a debounced fault reaction will be the result.

Description:

The function incorporates the following requirements:

G35 Monitoring of FS-IST questions sent by MU

After the data received from the MU have been pre-processed, in particular the newly sent question has been translated (if possible) into the matching question number on the MC, this function first predicts the expected question number; inputs for this are the previous, saved question number IDX_NR_LID_MC_FCT_SPC_IST_OLD, the current state of the MU anti-bounce counter ABC_LID_MC_FCT_SPC_IST, and the saved value ABC_LID_MC_FCT_SPC_IST_OLD of the MU anti-bounce counter, and the result of the prediction is stored in IDX_NR_LID_MC_FCT_SPC_IST_PRED.

Once the predicted question number has been obtained, a range check is performed for this decoded question number; in case it is found to be out of range (which means that the decode process detected an invalid question to which no matching question number could be found), a debounced fault reaction shall be performed and the *predicted* question number shall be saved (it is no use to save the invalid question number because the next prediction has to be based on a valid question number; moreover, the predicted question number is the one that should have been sent in the first place).

If the range check was successful, the predicted question number is compared with the decoded question number. If the two numbers disagree, a debounced fault reaction shall be performed, using the actions ACTION_ECM3_McIncAbc and ACTION_ECM3_McDecAbc, the anti-bounce counter NC_ABC_MC_WRG_IDX and (potentially) the error code NC_ERR_COD_MC_WRG_IDX. The difference to the previous case with the failed range check is the fact that in this case, the decoded question number will, despite not being the expected one, still be used as the basis for the next prediction: after all, it is a valid question number sent by the MU and it reflects the internal state of the MU (at least from the point of view of FS-IST). If the prediction for the next question number is to have any chance of being accurate, the decoded value has to be used because also the MU bases its internal calculation on it.

Note that the saved question number is initialised to the maximal question number because then it is only necessary to increase it by 1 (modulo the maximal question number, i.e., 0) to obtain the predicted question number in case the MU anti-bounce counter has current value 0 - as is the case in PREDRIVE when the function is active for the first time, and so the initial case does not require special treatment but can be handled just like any other case.

Now back to the calculation of the predicted question number. There are several cases, all distinguished by the interplay between saved value (ABC_LID_MC_FCT_SPC_IST_OLD) and newly transmitted value (ABC_LID_MC_FCT_SPC_IST) of the MU anti-bounce counter:

1. **ABC_LID_MC_FCT_SPC_IST ≤ 1:**

   Then no error is present on the MU, hence the MU should send the next question, i.e.,

   IDX_NR_LID_MC_FCT_SPC_IST_PRED =
   (IDX_NR_LID_MC_FCT_SPC_IST_OLD + 1)
   MOD NC_TEST_REC_IDX_MAX_MON2,

   as always during normal operation (the modulo-operation is necessary to avoid overflows of the predicted question number); the fact that less than or equal to 1 is tested is connected to the fact that the MU repeats a wrongly answered question until this condition for its FS-IST anti-bounce counter is satisfied;

2. **(ABC_LID_MC_FCT_SPC_IST_OLD = 0) AND (ABC_LID_MC_FCT_SPC_IST = 2):**

   This case expresses the fact that an error was detected on the MU between the last and the current activation of the FS-IST on the MC; consequently, it is expected that the MU simply repeats the wrongly answered question number, i.e.,

   IDX_NR_LID_MC_FCT_SPC_IST_PRED =
   (IDX_NR_LID_MC_FCT_SPC_IST_OLD - 1)
   (for IDX_NR_LID_MC_FCT_SPC_IST_OLD > 0)

   and

   IDX_NR_LID_MC_FCT_SPC_IST_PRED =
   NC_TEST_REC_IDX_MAX_MON2 - 1;
   (for IDX_NR_LID_MC_FCT_SPC_IST_OLD = 0)

   repeating the wrongly answered question means expecting the predecessor of the current question number because of the timing of fault detection in the FS-IST: in a certain communication, the MU receives an FS-IST answer *A(i)* from the MC to a question *Q(i)* with number *i* and, due to the bidirectional nature of the communication interface between MC and MU, in the very same communication it sends out the next question *Q(i+1)* (as the MU must assume that the received answer is correct because of system robustness issues); consequently, after the next activation of the FS-IST functions on the MC after this communication, *i+1* is stored in IDX_NR_LID_MC_FCT_SPC_IST_OLD; in the meantime, the MU has detected that *A(i)* was wrong: a debounced fault reaction is the consequence (responsible for the checked increase of the MU anti-bounce counter from 0 to 2), and in the next communication, the MU will repeat the question *Q(i)*, i.e., exactly as expressed in the formula for the predicted question number above;

3. **Otherwise (meaning that the current value is >1 and the saved value is >0):**

   This means that the MU will repeat the same question number as the saved one, i.e., IDX_NR_LID_MC_FCT_SPC_IST_PRED = IDX_NR_LID_MC_FCT_SPC_IST_OLD, and the explanation is similar to the previous case: the constellation of saved and current value for the MU anti-bounce counter can only be present if there has been an error on the MU but not in the last communication because otherwise, the save value of the MU anti-bounce counter would be 0 (the previous case); consequently, the MU has already started to repeat a wrongly answered question, and *will continue to do so* until its anti-bounce counter reaches 1 (cf. the first case); hence it is expected to receive the same question number as in the communication before.

Application conditions:

Initialisation: *\/* Initialise predicted question number *\/*

IDX_NR_LID_MC_FCT_SPC_IST_PRED = 0;

/ Initialise saved question number to last question number (see above) */*

IDX_NR_LID_MC_FCT_SPC_IST_OLD =
NC_TEST_REC_IDX_MAX_MON2;

Recurrence: The **initialisation** shall be performed **after each power-up or reset**.

The **function** shall be performed **after the function "Pre-processing of FS-IST data received from MU" has been finished** *(it may run in parallel to the test whether the MU reacts to wrongs FS-IST answers)*.

Activation: (LV_MC_FCT_SPC_IST_ACT = 1) **OR** ("MU configured for TCU mode")

Deactivation: Otherwise

12.9.4 Export of the question number to level 2'

FUNCTION DESCRIPTION:

General information:

This sub-function provides the question number to level 2' after all the required sub-functions have been activated; before calling level 2', a range check for the to-be-exported data is performed, and level 2' is not called in case the range check fails.

Description:

This sub-function incorporates the following requirements:

G36 Export of the question number to level 2'

R36_1: The current value ABC_LID_MC_FCT_SPC_IST of the MU anti-bounce counter shall be saved as ABC_LID_MC_FCT_SPC_IST_OLD for the next FS-IST activation.

R36_2: The current question number IDX_NR_LID_MC_FCT_SPC_IST shall be assigned to the variable TEST_REC_IDX_MON2 to be exported to level 2'.

R36_3: A range check on the latter value shall be performed (in order to avoid an invalid value to be exported to level 2', in particular if it was not possible to decode the question sent by the MU; cf. G33).

R36_4: If the range check was successful, TEST_REC_IDX_MON2 shall be handed over to level 2', otherwise nothing is done: level 2' need not be activated with an invalid value (unnecessary computations), and there is no default value for it either (it is not clear what the MU expects in answer to a corrupted question).

Application conditions:

Initialisation: */* Initialise exported question number to invalid value */*

TEST_REC_IDX_MON2 = NC_IDX_NR_WRG_MC_FCT_SPC_IST;

Recurrence: The **initialisation** shall be performed **after each power-up or reset**

. The **function itself** shall be performed **after both FS-IST functions "Test of MU reaction to wrong FS-IST answers" and "Monitoring of FS-IST questions sent by MU" have been finished**.

Activation: (LV_MC_FCT_SPC_IST_ACT = 1) **OR** ("MU configured for TCU mode")

Deactivation: Otherwise

12.9.5 Export of the question number to level 2' (second calculation)

FUNCTION DESCRIPTION:

General information:

This sub-function provides the question number to level 2' after all the required sub-functions have been activated; before calling level 2', a range check for the to-be-exported data is performed, and level 2' is not called in case the range check fails.

Description:

This sub-function incorporates the following requirements:

G62 Export of the question number to level 2' (second calculation)

R62_1: The current question number IDX_NR_LID_MC_FCT_SPC_IST shall be assigned to the variable TEST_REC_IDX_MON2 to be exported to level 2'.

R62_2: A range check on the latter value shall be performed (in order to avoid an invalid value to be exported to level 2', in particular if it was not possible to decode the question sent by the MU; cf. G33).

R62_3: If the range check was successful, TEST_REC_IDX_MON2 shall be handed over to level 2', otherwise nothing is done: level 2' need not be activated with an invalid value (unnecessary computations), and there is no default value for it either (it is not clear what the MU expects in answer to a corrupted question).

Application conditions:

Initialisation:	-
Recurrence:	The **function** shall be executed **after both FS-IST functions "Test of MU reaction to wrong FS-IST answers" and "Monitoring of FS-IST questions sent by MU" have been finished (even if they were not activated)**.
Activation:	(LV_MC_FCT_SPC_IST_ACT = 0) **AND** ("MU not configured for TCU mode") */* note that this function is executed in those recurrences of Processor Monitoring where G36 is **not** executed */*
Deactivation:	Otherwise

12.9.6 Decoding of the result of the level 2' test calculation

FUNCTION DESCRIPTION:

General information:

This sub-function decodes the result of the test calculation from level 2' (a 16bit value) into an answer number that can then be used for generating the actual FS-IST answer sent to the MU (a 6bit value).

Description:

The sub-function incorporates the following requirements:

G37 Decoding of the result of the level 2' test calculation

R37_1: After finishing, the test calculation of level 2' shall export its result in the 16bit value RESP_MON2 (note: this is an *answer*, not an *answer number*, similar to the distinction between question and question number; RESP_MON2 cannot be sent to the MU directly: the 16bit value would not fit into the 6bit reserved in the communication protocol for the FS-IST answer from the MC; consequently, a decoding process, like the decoding of the MU question, has to be performed before the actual FS-IST answer can be obtained).

R37_2: The decoding process is performed utilising the 'conversion table' ID_RESP_MON2 and its axis LDPM_TEST_REC_IDX_MON2:
R37_2.1: the answer number IDX_NR_RESP_MC_FCT_SPC_IST shall be initialised to the invalid number NC_IDX_NR_WRG_MC_FCT_SPC_IST;
R37_2.2: then IDX_NR_RESP_MC_FCT_SPC_IST runs through the table ID_RESP_MON2 (using LDPM_TEST_REC_IDX_MON2) and an index is sought such that a lookup at that index in the conversion table ID_RESP_MON2 yields RESP_MON2 as the result;
R37_2.3: in this way, if an index into ID_RESP_MON2 corresponding to RESP_MON2 is found, IDX_NR_RESP_MC_FCT_SPC_IST is assigned this value or, if the level 2' response could not be decoded (i.e., if there is no matching table entry), IDX_NR_RESP_MC_FCT_SPC_IST is not updated and keeps the invalid number NC_IDX_NR_WRG_MC_FCT_SPC_IST as value (this information can then be used in the subsequent post-processing sub-function).

Application conditions:

Initialisation: */* Initialise answer number to wrong number */*

IDX_NR_RESP_MC_FCT_SPC_IST =
NC_IDX_NR_WRG_MC_FCT_SPC_IST;

Recurrence: The **initialisation** shall be performed **after each power-up or reset**.

The **function itself** shall be executed **after the FS-IST sub-function "Export of the question number to level 2'" has been finished**.

Activation: (LV_MC_FCT_SPC_IST_ACT = 1) **OR** ("MU configured for TCU mode")

Deactivation: Otherwise

12.9.7 Decoding of the result of the level 2' test calculation (second calculation)

FUNCTION DESCRIPTION:

General information:

This sub-function decodes the result of the test calculation from level 2' (a 16bit value) into an answer number that can then be used for generating the actual FS-IST answer sent to the MU (a 6bit value).

This is done for the "second calculation" with the same question number. It is necessary to store whether the result was correct or not; for this, we have to ensure that

TEST_REC_IDX_MON2 is valid since otherwise level2' would not have been activated (hence we have to perform a range check on TEST_REC_IDX_MON2).

Description:

The sub-function incorporates the following requirements:

G63 Decoding of the result of the level 2' test calculation (second calculation)

R63_1: After finishing, the test calculation of level 2' shall export its result in the 16bit value RESP_MON2 (note: this is an *answer*, not an *answer number*, similar to the distinction between question and question number; RESP_MON2 cannot be sent to the MU directly: the 16bit value would not fit into the 6bit reserved in the communication protocol for the FS-IST answer from the MC; consequently, a decoding process, like the decoding of the MU question, has to be performed before the actual FS-IST answer can be obtained).

R63_2: The decoding process is performed utilising the 'conversion table' ID_RESP_MON2 and its axis LDPM_TEST_REC_IDX_MON2:
R63_2.1: the answer number IDX_NR_RESP_MC_FCT_SPC_IST shall be initialised to the invalid number NC_IDX_NR_WRG_MC_FCT_SPC_IST;
R63_2.2: then IDX_NR_RESP_MC_FCT_SPC_IST runs through the table ID_RESP_MON2 (using LDPM_TEST_REC_IDX_MON2) and an index is sought such that a lookup at that index in the conversion table ID_RESP_MON2 yields RESP_MON2 as the result;
R63_2.3: in this way, if an index into ID_RESP_MON2 corresponding to RESP_MON2 is found, IDX_NR_RESP_MC_FCT_SPC_IST is assigned this value or, if the level 2' response could not be decoded (i.e., if there is no matching table entry), IDX_NR_RESP_MC_FCT_SPC_IST is not updated and keeps the invalid number NC_IDX_NR_WRG_MC_FCT_SPC_IST as value (this information can then be used in the subsequent post-processing sub-function).

R63_3: After the decoding, if

- TEST_REC_IDX_MON2 is in range *(level 2' has in fact been activated with a defined question number)*, **AND**
- IDX_NR_RESP_MC_FCT_SPC_IST <> TEST_REC_IDX_MON2 *(the answer number from level 2' does not correspond to current (valid!) question, e.g., if the level2' answer RESP_MON2 could not be found in ID_RESP_MON2)*, **AND**
- IDX_NR_ERR_MC_FCT_SPC_IST is out of range, i.e., it has a value not between 0 and NC_TEST_REC_IDX_MAX_MON2-1 *(no "second calculation" has failed previously; this is done to repeat the* first *wrongly answered "second calculation")*,

IDX_NR_ERR_MC_FCT_SPC_IST shall be set to TEST_REC_IDX_MON2 *(storing the (valid!) question number for which the second calculation failed for later reference; by the previous conditions, this is the smallest/earliest question number for which the "second calculation" failed)*; otherwise, nothing shall be done *(in particular, if IDX_NR_ERR_MC_FCT_SPC_IST is in range, it shall stay unchanged in order to remember the* first *question number for which the "second calculation" failed)*

Application conditions:

Initialisation: *\/* Initialise index for wrong second calculation to invalid number *\/*

```
IDX_NR_ERR_MC_FCT_SPC_IST =
        NC_IDX_NR_WRG_MC_FCT_SPC_IST;
```

Recurrence: The **initialisation** shall be performed **after each power-up or reset**.

The **function itself** shall be executed **after the FS-IST sub-function "Export of the question number to level 2' (second calculation)" has been finished**.

Activation: (LV_MC_FCT_SPC_IST_ACT = 0) **AND** ("MU not configured for TCU mode") */* note that this function is executed in those recurrences of Processor Monitoring where G37 is **not** executed */*

Deactivation: Otherwise

12.9.8 Post-processing of the FS-IST data on MC for transmission to MU

FUNCTION DESCRIPTION:

General information:

This sub-function performs the necessary post-processing steps regarding the FS-IST on the MC, most importantly sending of a wrong answer if this has previously been requested (whether deliberately, because the redundant switch-off path was triggered, or because any of the decoding steps has failed), and toggling the activation condition for the FS-IST (which is only activated every second communication cycle).

Description:

The sub-function incorporates the following requirements:

G38 Post-processing of the FS-IST data on MC for transmission to MU

R38_1: If any of the following errors is active:
- the redundant switch-off path has been triggered (LV_MC_DR_OFF = 1, LV_MC_DR_OFF_CPL = 0, checked using the RAM check service actions), or
- a deliberately wrong answer shall be sent to the MU (LV_MC_FCT_SPC_IST_WRG_RESP_ACT = 1), or
- the current MU question could not be decoded properly (TEST_REC_IDX_MON2 = NC_IDX_NR_WRG_MC_FCT_SPC_IST), or
- the current level 2' answer could not be decoded properely (IDX_NR_RESP_MC_FCT_SPC_IST = NC_IDX_NR_WRG_MC_FCT_SPC_IST), or
- the current question has been answered wrongly during the last "second calculation" (IDX_NR_RESP_MC_FCT_SPC_IST = IDX_NR_ERR_MC_FCT_SPC_IST),

then a wrong answer shall be sent to the MU, calling the infrastructure action ACTION_INFR_McSendFsistResp() with NC_IDX_RESP_WRG_MC_FCT_SPC_IST as parameter.

R38_2: Otherwise, in case no error has been detected, the corresponding entry in the "answer array" NC_IDX_RESP_MC_FCT_SPC_IST[] shall be sent to the MU, calling the infrastructure action ACTION_INFR_McSendFsistResp() with NC_IDX_RESP_MC_FCT_SPC_IST[IDX_NR_RESP_MC_FCT_SPC_IST] as parameter.

R38_3: Finally, the active flag LV_MC_FCT_SPC_IST_ACT shall be toggled so that a new answer is only sent in the communication after the next communication cycle.

R38_4: The value of LV_MC_DR_OFF shall be written to the infrastructure using the action ACTION_INFR_McSendDrOff() so that it can be transmitted to the MU in NORMAL communications.

R38_5: The flag LV_MC_FCT_SPC_IST_WRG_RESP_ACT shall be reset to stop deliberately sending a wrong response.

R38_6: If IDX_NR_ERR_MC_FCT_SPC_IST = IDX_NR_RESP_MC_FCT_SPC_IST, IDX_NR_ERR_MC_FCT_SPC_IST shall be re-set to NC_IDX_NR_WRG_MC_FCT_SPC_IST *(a wrong answer for the previously "remembered" question number where the "second calculation" failed shall be sent to MU so the question will be repeated by the MU and hence the "marker" can be re-set to its initial value)*.

Application conditions:

Initialisation:	-
Recurrence:	The **function** shall be **executed after all other FS-IST sub-functions have been finished**.
Activation:	(LV_MC_FCT_SPC_IST_ACT = 1) **OR** ("MU configured for TCU mode")
Deactivation:	Otherwise

12.10 MC Error Memory Management of Processor Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_RST_ERR_MEM_ACT	V/S	0...FFH	0...255	1	[-]
Local variable: current value of reset counter for error memory management, plus potentially additional information					
CTR_RST_ERR_MEM_OLD	V/S	0...FFH	0...255	1	[-]
Local variable: old value of reset counter for error memory management, plus potentially additional information					
ERR_COD_ERR_MEM_ACT	V/S	0...1FH	0...31	1	[-]
Local variable: current value of error code for error memory management					
LV_MC_CPT_ERR_MEM_ACT	V/S	0...1H	0...1	1	[-]
Local variable: flag indicating whether the current error code for error memory management belongs to MC (=1) or MU(=0)					
ERR_COD_ERR_MEM_OLD	V/S	0...1FH	0...31	1	[-]
Local variable: old value of error code for error memory management					
LV_MC_CPT_ERR_MEM_OLD	V/S	0...1H	0...1	1	[-]
Local variable: flag indicating whether the old error code for error memory management belongs to MC (=1) or MU(=0)					
STATE_ERR_IDC_ERR_MEM_ACT	V/S	0...3H	0...3	1	[-]
Local variable: current error indication for error memory management					
STATE_ERR_IDC_ERR_MEM_OLD	V/S	0...3H	0...3	1	[-]
Local variable: old error indication for error memory management					
LV_MC_UPD_ERR_MEM_MC	-	0...1H	0...1	1	[-]
Local variable: update flag for MC entries into error history (used as activation condition)					
LV_MC_UPD_ERR_MEM_MU	-	0...1H	0...1	1	[-]
Local variable: update flag for MU entries into error history (used as activation condition)					

Note: data with mode "S" shall only be stored in a reset-safe manner, not in NVMY!

Input data:

NC_ERR_COD_MC_NOT_ERR	STATE_ERR_IDC_MC	CTR_RST_MC	ERR_COD_MC
NC_STATE_ERR_IDC_MC_NOT_PRES	CTR_RST_MU	ERR_COD_MU	STATE_ERR_IDC_MU
LV_V_L_DET_MU	LV_V_H_DET_MU	NC_ERR_COD_MU_V_L	NC_ERR_COD_MU_V_H
NC_ERR_COD_MU_V_L_H	STATE_MC	CTR_RST_MC_CPL	NC_STATE_MC_CONF
NC_STATE_MC_DI	NC_CTR_RST_MC_THD	IDX_STATE_SOPC	IDX_STATE_SOPC_CPL
NC_ERR_COD_MC_SOPC	STATE_MU_TMP	NC_ERR_COD_MC_HD	STATE_MU_TMP_CPL
NC_STATE_MC_PRDR	NC_STATE_MC_NORM	NC_STATE_MC_NOT_VLD	LV_ABC_MC_ERR_COM
LV_ABC_MC_ERR_PFM	LV_ABC_MC_ERR_FCT_SPC_IST	LV_ABC_MC_ERR_PFM_6	

Export actions:

ACTION_ECM3_McSetErrmUpdateFlag(IN <cpt>)
This action sets the update flag LV_MC_UPD_ERR_MEM_cpt (=0 means MU, =1 means MC) for error memory information.

Import actions:

ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)
ACTION_ECM3_McReadChkCpl(OUT <>, IN <>, IN <>, IN <>)
ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_ECM3_McReadChkState(OUT <>)
ACTION_ECM3_McErrmService(IN <>, IN <>, IN <>, IN <>, IN <>, IN <>, IN <>, IN <>)

Note: the last action is defined in the application incidences and hence its functional content can be adjusted to project-specific needs/requirements.

FUNCTION DESCRIPTION:

General information:

This module serves as the interface between the processor monitoring and the error memory management: as such, it provides an action call by which the current error information (error code, error indication, reset counter, and whether the data stem from MC or MU) get stored in a history-like fashion; furthermore, an action is imported (from the application incidences of processor monitoring to allow project-specific adaptations to different error memory concepts) and called after the saved values have been updated; this imported action performs the actual triggering of the error memory management with the current failure state.

Description:

The module provides the action ACTION_ECM3_McSetErrmUpdateFlag() which sets one of the update flags LV_MC_UPD_ERR_MEM_MC and LV_MC_UPD_ERR_MEM_MU, depending on the value of its argument.

This flag then, in turn, serves as (one of) the activation condition(s) for the functions described here, which then create/continue an error history using the variables in the output table.

The error history records which component has triggered the update of error memory information: the MC in INIT and after a fault reaction leading to permanent disabling of the power stages; the MU after a successful CONFIG communication and after receiving a DISABLE header from the MU in a state different from DISABLE.

The history stores the current value of reset counter, error code, error indication, and component (MC or MU), plus the corresponding old values (the length of the history is only two).

After the history is potentially updated, the project-specific function ACTION_ECM3_McErrmService(), which is imported from the application incidences (where it is defined), shall be called; this is taken care of by another sub-function, since the action shall be called *always* even when none of the update functions is actually executed due to unsatisfied activation conditions: this is necessary to inform the error memory management not only in the case that an error is present but also in the case that no error is present.

Description of ACTION_ECM3_McSetErrmUpdateFlag

Description:

Syntax:	ACTION_ECM3_McSetErrmUpdateFlag (IN <cpt>)	
Parameter (IN):	cpt	Component: 0 means MU, 1 means MC
Short description:	This action sets one of the two flags LV_MC_UPD_ERR_MEM_MU (if cpt = 0) or LV_MC_UPD_ERR_MEM_MC (if cpt = 1); note that only setting the flag is prompted by other modules, the actual update of the data is done in this module (using the flags as activation conditions).	

Application conditions:

Initialisation: After power-up:

/ Initialise the information for error memory management */*

```
CTR_RST_ERR_MEM_ACT = 0;
CTR_RST_ERR_MEM_OLD = 0;
ERR_COD_ERR_MEM_ACT = NC_ERR_COD_MC_NOT_ERR;
ERR_COD_ERR_MEM_OLD = NC_ERR_COD_MC_NOT_ERR;
STATE_ERR_IDC_ERR_MEM_ACT =
                              NC_STATE_IDC_MC_NOT_PRES;
STATE_ERR_IDC_ERR_MEM_OLD =
                              NC_STATE_IDC_MC_NOT_PRES;
LV_MC_CPT_ERR_MEM = 0;
LV_MC_CPT_ERR_MEM_OLD = 0;
LV_MC_UPD_ERR_MEM_MC = 0;
LV_MC_UPD_ERR_MEM_MU = 0;
```

After reset:

/ No initialisation of error memory data necessary, update will be performed, if necessary, by means of an action call */*

```
LV_MC_UPD_ERR_MEM_MC = 0;
LV_MC_UPD_ERR_MEM_MU = 0;
```

Recurrence: The **initialisation** shall be **performed after each power-up or reset**.

Activation: -

Deactivation: -

12.10.1 Update of error memory information with MU data

FUNCTION DESCRIPTION:

General information:

This sub-function describes when and how the error history for the error memory gets updated with new data from the MU.

There are two cases for this:

1. after a successful CONFIG communication, when the error information sent from the MU have been evaluated on the MC (including the treatment of under- and over-voltage having been detected on the MU);
2. after a successful communication in which the MU sent a DISABLE header while the MC was (prior to the communication) not yet in DISABLE, and the error information sent from the MU has been evaluated on the MC.

Since the subsequent actions are slightly different and would only lead to complicated case distinctions, the two cases will be represented by two sub-functions.

12.10.1.1 Update of error memory information with MU data after CONFIG communication

FUNCTION DESCRIPTION:

General information:

This sub-function describes how the error memory gets updated with new data from the MU after a successful CONFIG communication, taking into account potentially set under-/over-voltage flags.

Description:

The function incorporates the following requirements:

G39 Update of error memory information with MU data after CONFIG communication

R39_1: If

- **LV_MC_V_L_DET_MU =1** or **LV_MC_V_L_DET_MU =1** *(i.e., a voltage monitoring error has been detected by the MU before the previous reset)*, and
- the _ACT-variables *(describing the most recently stored error)* indicate an error which is **different from the same (combination of) voltage monitoring error(s) as currently detected by the MU** (*i.e., LV_MC_CPT_ERR_MEM_ACT <> 0 or ERR_COD_ERR_MEM_ACT not equal to NC_ERR_COD_MU_V_L, NC_ERR_COD_MU_V_H, or NC_ERR_COD_MU_V_L_H, respectively, depending on the three possible combinations where at least one of the two flags is set)*:

R39_1.1: the values of the _ACT-variables of error memory information shall be saved in the corresponding _OLD-variables;
R39_1.2: ERR_COD_ERR_MEM_ACT shall be set to NC_ERR_COD_MU_V_L, NC_ERR_COD_MU_V_H, or NC_ERR_COD_MU_V_L_H, respectively, depending on the three possible combinations where at least one of the two flags is set;
R39_1.3: CTR_RST_ERR_MEM_ACT shall be set to 0;
R39_1.4: STATE_ERR_IDC_ERR_MEM_ACT shall be set to NC_STATE_IDC_MC_NOT_PRES;
R39_1.5: LV_MC_CPT_ERR_MEM_ACT shall be set to 0, i.e., the active component shall be set to MU.

R39_2: If R39_1 does not apply but:

- **LV_MC_CPT_ERR_MEM_ACT <> 0** *(most recently stored error not an MU error)*, or
- **ERR_COD_ERR_MEM_ACT <> ERR_COD_MU** *(most recently stored error has different error code)*, or
- **CTR_RST_ERR_MEM_ACT (four LSBs) > CTR_RST_MU** *(most recently stored reset counter greater than current reset counter on MU, i.e., even though potentially the same error has occurred again, the decreased actual value of the reset counter indicates that the reset counter must have healed in between)*:

R39_2.1: the values of the _ACT-variables of error memory information shall be saved in the corresponding _OLD-variables;
R39_2.2: the values of ERR_COD_MU, STATE_ERR_IDC_MU, and CTR_RST_MU shall be saved to the _ACT-variables, and the active component shall be set to MU, i.e., LV_MC_CPT_ERR_MEM_ACT shall be set to 0.

R39_3: If R39_1 and R39_2 both do not apply but **CTR_RST_ERR_MEM_ACT (four LSBs) < CTR_RST_MU** *(i.e., the same MU error has occurred at least one more time which led to the increment of the reset counter since the storage)*, then the _ACT-variables shall be updated with the values of ERR_COD_MU, STATE_ERR_IDC_MU, and CTR_RST_MU, and LV_MC_CPT_ERR_MEM_ACT shall be set to 0, i.e., the active component shall be set to MU.

R39_4: The update flag LV_MC_UPD_ERR_MEM_MU shall be reset.

Application conditions:

Initialisation:	-
Recurrence:	The function shall be executed after the function "Evaluation of error information from MU on MC" (when potentially new data from the MU are available on the MC).
Activation:	(STATE_MC = NC_STATE_MC_CONF) **AND** (LV_MC_UPD_ERR_MEM_MU = 1) **AND** ((STATE_ERR_IDC_MU <> NC_STATE_ERR_IDC_MC_NOT_PRES) **OR** (LV_V_L_DET_MU = 1) **OR** (LV_V_H_DET_MU = 1)) *(only active in CONFIG if update flag for MU is set and if any of the following conditions is true: the error indication of the MU is different from "no error present"; the UVD flag is set; the OVD flag is set)*
Deactivation:	Otherwise

12.10.1.2 Update of error memory information with MU data after DISABLE communication

FUNCTION DESCRIPTION:

General information:

This sub-function describes how the error memory gets updated with new data from the MU after a successful communication in which the MU sent a DISABLE header but the MC was, prior to the communication, not in DISABLE state (meanwhile, it is also in DISABLE).

Description:

The function incorporates the following requirements:

G40 Update of error memory information with MU data after DISABLE communication

R40_1: If
- **LV_MC_CPT_ERR_MEM_ACT <> 0** *(most recently stored error not an MU error)*, or
- **ERR_COD_ERR_MEM_ACT <> ERR_COD_MU** *(most recently stored error has different error code)*, or
- **CTR_RST_ERR_MEM_ACT (four LSBs) > CTR_RST_MU** *(most recently stored reset counter greater than counter on MU, i.e., even though potentially the same error has occurred again, the decreased actual value of the reset counter indicates that the reset counter must have healed in between)*

R40_1.1: the values of the _ACT-variables of error memory information shall be saved in the corresponding _OLD-variables;
R40_1.2: the values of ERR_COD_MU, STATE_ERR_IDC_MU, and CTR_RST_MU shall be saved to the _ACT-variables, and the active component shall be set to MU, i.e., LV_MC_CPT_ERR_MEM_ACT shall be set to 0.

R40_2: If R40_1 does not apply, but **CTR_RST_ERR_MEM_ACT (four LSBs) < CTR_RST_MU** *(i.e., the same MU error has occurred at least one more time which led to the increment of the reset counter since the storage)*, then the _ACT-variables shall be updated with the values of ERR_COD_MU, STATE_ERR_IDC_MU, and CTR_RST_MU, and LV_MC_CPT_ERR_MEM_ACT shall be set to 0, i.e., the active component shall be set to MU.

R40_3: The update flag LV_MC_UPD_ERR_MEM_MU shall be reset.

Application conditions:

Initialisation: -

Recurrence: The function shall be executed after the function "Evaluation of error information from MU on MC" (when potentially new data from the MU are available on the MC).

Activation: (STATE_MC = NC_STATE_MC_DI) **AND**
(LV_MC_UPD_ERR_MEM_MU = 1) **AND**
(STATE_ERR_IDC_MU <> NC_STATE_ERR_IDC_MC_NOT_PRES)
(only active in DISABLE if update flag for MU is set and error indication different from "no error present")

Deactivation: Otherwise

12.10.2 Update of error memory information with MC data

FUNCTION DESCRIPTION:

General information:

This sub-function describes when and how the error history for the error memory gets updated with new data from the MC.

There are two cases for this:
1. in INIT (potentially after the error indication has been changed from "ECU restart request" to "ECU restart was requested");
2. after a fault reaction on the MC leading to permanent disabling (i.e., CTR_RST_MC = NC_CTR_RST_MC_THD + 1, error indication "disable request").

In the first case, the update function is only called if the error indication is different from "no error present". Note that this condition is also satisfied in the second case, and in both cases, the update flag LV_MC_UPD_ERR_MEM_MC has been set by the corresponding modules ("State of fault reaction and up-to-dateness of error code" and "Central fault reaction", respective).

Consequently, the activation conditions for the function are that LV_MC_UPD_ERR_MEM_MC is set and that the error indication is different from "no error present."

Description:

The function incorporates the following requirements:

G41 Update of error memory information with MC data

R41_1: If
- **LV_MC_CPT_ERR_MEM_ACT <> 1** *(most recently stored error not an MC error)*, or
- **ERR_COD_ERR_MEM_ACT <> ERR_COD_MC** *(most recently stored error has different error code)*, or
- **CTR_RST_ERR_MEM_ACT (four LSBs) > CTR_RST_MC** *(most recently stored reset counter greater than counter on MC, i.e., even though potentially the same error has occurred again, the decreased actual value of the reset counter indicates that the reset counter must have healed in between)*:

R41_1.1: the values of the _ACT-variables of error memory information shall be saved in the corresponding _OLD-variables;
R41_1.2: the values of ERR_COD_MC, STATE_ERR_IDC_MC, and CTR_RST_MC (after a consistency check with its complement CTR_RST_MC_CPL using ACTION_ECM3_ReadChkCpl()) shall be saved to the _ACT-variables, and the active component shall be set to MC, i.e., LV_MC_CPT_ERR_MEM_ACT shall be set to 1;
R41_1.3: if ERR_COD_MC = NC_ERR_COD_MC_SOPC *(i.e., an error in the switch-off path check was detected)*, IDX_STATE_SOPC shall be encoded into the four MSBs of CTR_RST_ERR_MEM_ACT (CTR_RST_MC uses only the four LSBs); IDX_STATE_SOPC shall be accessed using the RAM check service actions (complement check with IDX_STATE_SOPC_CPL);
R41_1.4: if ERR_COD_MC = NC_ERR_COD_MC_HD *(i.e., an invalid header-byte has been sent by the MU)*, the value of STATE_MU_TMP shall be encoded into the four MSBs of CTR_RST_ERR_MEM_ACT (CTR_RST_MC uses only the four LSBs); the value encoded in CTR_RST_ERR_MEM_ACT shall be as shown in the following table:

Value of STATE_MU_TMP	Encoded value (dec.)
NC_STATE_MC_CONF	1
NC_STATE_MC_PRDR	2
NC_STATE_MC_NORM	3
NC_STATE_MC_DI	4
NC_STATE_MC_NOT_VLD	5
00H	6
FFH	7

R41_1.5: if ERR_COD_MC <> NC_ERR_COD_MC_SOPC and ERR_COD_MC <> NC_ERR_COD_MC_HD *(i.e., neither an SOPC nor a header error has occurred)*, the flags LV_ABC_MC_ERR_COM, LV_ABC_MC_ERR_PFM, LV_ABC_MC_ERR_PFM_6 and LV_ABC_MC_ERR_FCT_SPC_IST shall be encoded into the four MSBs of CTR_RST_ERR_MEM_ACT (CTR_RST_MC uses only the four LSBs) as follows:

Bit 3	Bit 2	Bit 1	Bit 0
LV_ABC_MC_ERR_FCT_SPC_IST	LV_ABC_MC_ERR_PFM_6	LV_ABC_MC_ERR_PFM	LV_ABC_MC_ERR_COM

R41_2: If R41_1 does not apply but **CTR_RST_ERR_MEM_ACT (four LSBs) < CTR_RST_MC** *(i.e., the same MC error has occurred at least one more time which led to the increment of the reset counter since the storage)*:
R41_2.1: the _ACT-variables shall be updated with the values of ERR_COD_MC, STATE_ERR_IDC_MC, and CTR_RST_MC (after a consistency check with its complement CTR_RST_MC_CPL using ACTION_ECM3_ReadChkCpl()), and LV_MC_CPT_ERR_MEM_ACT shall be set to 1, i.e., the active component shall be set to 1;
R41_2.2: if ERR_COD_MC = NC_ERR_COD_MC_SOPC *(i.e., an error in the switch-off path check was detected)*, IDX_STATE_SOPC shall be encoded into the four MSBs of CTR_RST_ERR_MEM_ACT (CTR_RST_MC uses only the four LSBs); IDX_STATE_SOPC shall be accessed using the RAM check service actions (complement check with IDX_STATE_SOPC_CPL);
R41_2.3: if ERR_COD_MC = NC_ERR_COD_MC_HD *(i.e., an invalid header-byte has been sent by the MU)*, the value of STATE_MU_TMP shall be encoded into the four MSBs of CTR_RST_ERR_MEM_ACT (CTR_RST_MC uses only the four LSBs); the value encoded in CTR_RST_ERR_MEM_ACT shall be as shown in above table;

R41_2.4: if ERR_COD_MC <> NC_ERR_COD_MC_SOPC and ERR_COD_MC <> NC_ERR_COD_MC_HD *(i.e., neither an SOPC nor a header error has occurred)*, the flags LV_ABC_MC_ERR_COM, LV_ABC_MC_ERR_PFM, LV_ABC_MC_ERR_PFM_6 and LV_ABC_MC_ERR_FCT_SPC_IST shall be encoded into the four MSBs of CTR_RST_ERR_MEM_ACT (CTR_RST_MC uses only the four LSBs) as shown in the table above.

R41_3: The update flag LV_MC_UPD_ERR_MEM_MC shall be reset.

Application conditions:

Initialisation:	-
Recurrence:	After the functions "State of fault reaction and up-to-dateness of error code" and "Central fault reaction"
Activation:	(LV_MC_UPD_ERR_MEM_MC = 1) **AND** (STATE_ERR_IDC_MC <> NC_STATE_ERR_IDC_NOT_PRES)
Deactivation:	Otherwise

12.10.3 Activation of project-specific function calling the error memory management

FUNCTION DESCRIPTION:

General information:

This sub-function simply calls the project-specific function ACTION_ECM3_McErrmService() which uses hands error history to the error memory management and sets the necessary diagnosis flags.

The action shall be called every time when potentially an update of the error history could have been performed, regardless of an actual update; this is done because the error memory also needs to be informed of the case where *no* error is present/detected.

Description:

The sub-function incorporates the following requirements:

G42 Activation of project-specific function calling the error memory management

R42_1: The project-specific action ACTION_ECM3_McErrmService() shall be called in CONFIG or DISABLE with the complete error history after any of the update functions above has been called ***regardless of whether the update function actually was activated***.

R42_2: In DISABLE, ACTION_ECM3_McErrmService() shall only be called once.

Application conditions:

Initialisation: -

Recurrence: The function shall be called each time after one of the update functions could be have been activated, regardless of the actual activation of the update function.

Activation: (STATE_MC = NC_STATE_MC_CONF) **OR**
(STATE_MC = NC_STATE_MC_DI) **OR**
("LV_MC_UPD_ERR_MEM_MC or LV_MC_UPD_ERR_MEM_MU were previously set")
(activated only in CONFIG and DISABLE after an update function could have been called)

Deactivation: -

12.11 MC Processor Monitoring (Application Incidences)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_MON_3	O/V	0...1H	0...1	1	[-]
Error detected by processor monitoring					
LV_CDN_DIAG_MON_3	V	0...1H	0...1	1	[-]
Diagnosis condition for processor monitoring error					
LV_END_DIAG_MON_3	V	0...1H	0...1	1	[-]
Indicates that diagnosis has been finished					
ERR_SYM_MON_3	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom extracted from ERR_COD_MC and ERR_COD_MU					
ENVD_0_MON_3	O/V/S	0...FFH	0...255	1	[-]
Environmental data 0 for general processor monitoring error					
ENVD_1_MON_3	O/V/S	0...FFH	0...255	1	[-]
Environmental data 1 for general processor monitoring error					
ENVD_2_MON_3	O/V/S	0...FFH	0...255	1	[-]
Environmental data 2 for general processor monitoring error					
ENVD_3_MON_3	O/V/S	0...FFH	0...255	1	[-]
Environmental data 3 for general processor monitoring error					
LV_MC_IGN_KEY	O/V	0...1H	0...1	1	[-]
Ignition key information to be sent from MC to MU					
LV_MC_ST_LIH_SOPC	V/O	0...1H	0...1	1	[-]
Request to start in limp-home mode based on result of the switch-off path check					
LV_MC_ST_LIH_SOPC_CPL	O	0...1H	0...1	1	[-]
Complement of LV_MC_ST_LIH_SOPC_CPL					
LV_ERR_RAM_MON_3	V/O	0...1H	0...1	1	[-]
RAM test error detected by processor monitoring					
LV_INJ_OFF_TMR_INJ_ENA	O/V	0...1H	0...1	1	[-]
Indicates enabling of injection by injection off timer of MU					
LV_MU_IGN_KEY_OLD	V	0...1H	0...1	1	[-]
Flag used for storing value of ignition key signal on MU from last recurrence (needed for IOT handling)					

Note: storage ENVD_x_MON_3 as non-volatile shall be ensured via the Error Memory Management.

Input data:

NC_STATE_MC_PRDR	STATE_MC	NC_STATE_MC_INI	NC_STATE_MC_CONF
NC_STATE_MC_NORM	NC_STATE_MC_DI	NLC_MU_MOD	NLC_MU_IGN_KEY_ACQ
NLC_MU_POW_OFF_TMR_ENA	NLC_MU_INJ_OFF_TMR_ENA	NC_NR_MC_COMP	NC_ERR_SWI_MC_LEVEL_2_CAL
NC_ERR_COD_MC_NOT_ERR	NC_ERR_COD_MC_HD	NC_ERR_COD_MC_CKS	NC_ERR_COD_MC_PAR
NC_ERR_COD_MC_CONF_DIF	NC_ERR_COD_MC_SOPC	NC_ERR_COD_MC_COMP_NR	NC_ERR_COD_MC_ROM_LEVEL_1
NC_ERR_COD_MC_RAM_LEVEL_1	NC_ERR_COD_MC_ROM_LEVEL_2	NC_ABC_MC_PFM_3_TOG	NC_ERR_COD_MC_WRG_IDX
NC_ERR_COD_MC_WRG_RESP	NC_ERR_COD_MC_NOT_DEC	NC_ERR_COD_MC_PFM_0_TOG	NC_ERR_COD_MC_PFM_1_TOG
NC_ERR_COD_MC_PFM_2_TOG	NC_ERR_COD_MC_PFM_3_TOG	NC_ERR_COD_MC_PFM_4_TOG	NC_ERR_COD_MC_PFM_5_TOG
NC_ERR_COD_MC_PFM_6_TOG	NC_ERR_COD_MC_NOT_VLD_TRAN	NC_ERR_COD_MC_PFM_0_RESP	NC_ERR_COD_MC_PFM_1_RESP
NC_ERR_COD_MC_PFM_2_RESP	NC_ERR_COD_MC_PFM_3_RESP	NC_ERR_COD_MC_PFM_4_RESP	NC_ERR_COD_MC_PFM_5_RESP
NC_ERR_COD_MC_PFM_6_RESP	NC_ERR_COD_MU_FCT_SPC_IST	NC_ERR_COD_MU_V_L	NC_ERR_COD_MU_V_H
NC_ERR_COD_MU_V_L_H	NC_STATE_ERR_IDC_MC_NOT_PRES	NC_STATE_ERR_IDC_MC_REST_REQ	NC_STATE_ERR_IDC_MC_DI_REQ
NC_STATE_ERR_IDC_MC_PREV_REST	NC_ABC_MC_HD	NC_ABC_MC_CKS	NC_ABC_MC_CONF_DIF
NC_ABC_MC_WRG_IDX	NC_ABC_MC_WRG_RESP	NC_ABC_MC_NOT_DEC	NC_ABC_MC_PFM_0_TOG
NC_ABC_MC_PFM_1_TOG	NC_ABC_MC_PFM_2_TOG	NC_ABC_MC_PFM_3_TOG	NC_ABC_MC_PFM_4_TOG
NC_ABC_MC_PFM_5_TOG	NC_ABC_MC_PFM_6_TOG	NC_ABC_MC_PFM_0_RESP	NC_ABC_MC_PFM_1_RESP
NC_ABC_MC_PFM_2_RESP	NC_ABC_MC_PFM_3_RESP	NC_ABC_MC_PFM_4_RESP	NC_ABC_MC_PFM_5_RESP
NC_ABC_MC_PFM_6_RESP	NC_ABC_MC_HD_INI	NC_ABC_MC_CKS_INI	NC_ABC_MC_CONF_DIF_INI
NC_ABC_MC_WRG_IDX_INI	NC_ABC_MC_WRG_RESP_INI	NC_ABC_MC_NOT_DEC_INI	NC_ABC_MC_PFM_0_TOG_INI
NC_ABC_MC_PFM_1_TOG_INI	NC_ABC_MC_PFM_2_TOG_INI	NC_ABC_MC_PFM_3_TOG_INI	NC_ABC_MC_PFM_4_TOG_INI
NC_ABC_MC_PFM_5_TOG_INI	NC_ABC_MC_PFM_6_TOG_INI	NC_ABC_MC_PFM_0_RESP_INI	NC_ABC_MC_PFM_1_RESP_INI
NC_ABC_MC_PFM_2_RESP_INI	NC_ABC_MC_PFM_3_RESP_INI	NC_ABC_MC_PFM_4_RESP_INI	NC_ABC_MC_PFM_5_RESP_INI
NC_ABC_MC_PFM_6_RESP_INI	NC_ABC_MC_HD_INC	NC_ABC_MC_CKS_INC	NC_ABC_MC_CONF_DIF_INC
NC_ABC_MC_WRG_IDX_INC	NC_ABC_MC_WRG_RESP_INC	NC_ABC_MC_NOT_DEC_INC	NC_ABC_MC_PFM_0_TOG_INC
NC_ABC_MC_PFM_1_TOG_INC	NC_ABC_MC_PFM_2_TOG_INC	NC_ABC_MC_PFM_3_TOG_INC	NC_ABC_MC_PFM_4_TOG_INC
NC_ABC_MC_PFM_5_TOG_INC	NC_ABC_MC_PFM_6_TOG_INC	NC_ABC_MC_PFM_0_RESP_INC	NC_ABC_MC_PFM_1_RESP_INC
NC_ABC_MC_PFM_2_RESP_INC	NC_ABC_MC_PFM_3_RESP_INC	NC_ABC_MC_PFM_4_RESP_INC	NC_ABC_MC_PFM_5_RESP_INC
NC_ABC_MC_PFM_6_RESP_INC	NC_ABC_MC_HD_MAX	NC_ABC_MC_CKS_MAX	NC_ABC_MC_CONF_DIF_MAX
NC_ABC_MC_WRG_IDX_MAX	NC_ABC_MC_WRG_RESP_MAX	NC_ABC_MC_NOT_DEC_MAX	NC_ABC_MC_PFM_0_TOG_MAX
NC_ABC_MC_PFM_1_TOG_MAX	NC_ABC_MC_PFM_2_TOG_MAX	NC_ABC_MC_PFM_3_TOG_MAX	NC_ABC_MC_PFM_4_TOG_MAX

NC_ABC_MC_PFM_5_TOG_MAX	NC_ABC_MC_PFM_6_TOG_MAX	NC_ABC_MC_PFM_0_RESP_MAX	NC_ABC_MC_PFM_1_RESP_MAX
NC_ABC_MC_PFM_2_RESP_MAX	NC_ABC_MC_PFM_3_RESP_MAX	NC_ABC_MC_PFM_4_RESP_MAX	NC_ABC_MC_PFM_5_RESP_MAX
NC_ABC_MC_PFM_6_RESP_MAX	NC_CTR_COM_CYC_MC_DEC_RST_INI	NC_NR_MC_PFM	NC_NR_MC_PFM_TCC
NC_TCC_MC_PFM_TOG_6_MAX	NC_IDX_TEST_MC_FCT_SPC_IST[NC_TEST_REC_IDX_MAX_MON2]	NC_IDX_NR_WRG_MC_FCT_SPC_IST	NC_TEST_REC_IDX_MAX_MON2
NC_TCC_MC_IST_WRG_RESP_REQ	NC_TCC_MC_IST_WRG_RESP_CHK	NC_IDX_RESP_MC_FCT_SPC_IST[NC_TEST_REC_IDX_MAX_MON2]	NC_IDX_RESP_WRG_MC_FCT_SPC_IST
NC_T_MC_COM_CYC	NC_CTR_MU_DI_MAX	NC_STATE_RESP_SOPC_OK	NC_STATE_RESP_DR_DI
NC_STATE_RESP_DR_ENA	NC_STATE_RESP_DR_DFT	NC_CTR_RST_MC_DI	NC_ERR_SWI_MC_LEVEL_2_COD
NC_ERR_COD_MC_MU_READY	NC_NR_RESP_SOPC	NC_CTR_RST_MC_THD_SOPC	NC_RESP_SOPC_1[NC_NR_RESP_SOPC]
NC_RESP_SOPC_2[NC_NR_RESP_SOPC]	NC_CYCNR_MC_DR_DI_TMP	ERR_COD_MC	ERR_COD_MU
NC_CTR_RST_MC_THD	NC_ERR_COD_MC_RAM_LEVEL_2	NC_IDX_TEST_MC_FCT_SPC_IST[NC_TEST_REC_IDX_MAX_MON2]	NC_IDX_RESP_MC_FCT_SPC_IST[NC_TEST_REC_IDX_MAX_MON2]
NC_VP_MC_AN_DIG_MON_CPL_ERR	NC_TCC_MC_PFM_TOG_MAX[NC_NR_MC_PFM_TCC]	LV_RLY_MAIN_DLY_ERR	NLF_MC_PFM_SYN_BYTE_INI
NC_IDX_STATE_SOPC_DFCT	NC_STATE_RESP_SOPC_DFCT	IDX_STATE_SOPC	NC_CTR_MU_READY_MAX
NC_STATE_MC_NOT_VLD	NC_PWL_LOCK_CDN_ROM_CHK	LV_DIAG_END_RLY_MAIN_DLY	LV_MU_IGN_KEY
PWL_LOCK_CDN	LV_ERR_TMP_MU_MC	LV_INJ_OFF_TMR_INJ_ENA_TMP	

Export actions:

ACTION_ECM3_McErrmService(IN <cpt_act>, IN <err_cod_act>, IN <err_idc_act>, IN <ctr_rst_act>, IN <cpt_old>, IN <err_cod_old>, IN <err_idc_old>, IN <ctr_rst_old>)

This action receives all the available data from the error memory management of processor monitoring and which shall then trigger the error memory management to prepare saving the provided values in non-volatile memory for later analysis.

ACTION_ÊCM3_McLockPwlResource()

This action shall lock the power latch resource for the Standard ROM check to extend the power latch phase at least until the Standard ROM check has been completed.

ACTION_ECM3_McUnlockPwlResource()

This action shall unlock the power latch resource for the Standard ROM check.

Import actions:

ACTION_ECM3_McFaultReaction(IN <>)
ACTION_ECM3_McChangeState(IN <>)
ACTION_ECM3_McReadChkState(OUT <>)
ACTION_ECM3_ReadChkCpl(OUT <>, IN <>, IN <>)
ACTION_ECM3_McReadChkCpl(OUT <>, IN <>, IN <>, IN <>)
ACTION_ECM3_WriteChkCpl(INOUT <>, OUT <>, IN <>)
ACTION_ECM3_ChkCpl(IN <>, IN <>)
ACTION_SetRstInfo_Ecm3

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CTR_RST_THD_ERR_MEM	1	0...FH	0...15	1	[-]
Threshold for reset counter of MC and MU when error memory entries shall be generated					

12.11.1 Generic configuration – Value assignment for configuration data

FUNCTION DESCRIPTION:

General information:

This section describes the value assignment for the generic (i.e., project-independent) configuration data of Processor Monitoring (e.g., header-bytes, error codes etc.) for engine management systems (EMSs).

THESE ASSIGNMENTS CANNOT BE CHANGED BY PROJECTS!

Value assignment for configuration data of processor monitoring on MC:

```
/* State machine */

NC_STATE_MC_INI = 241;        /* F1H */
NC_STATE_MC_CONF = 146;   /* 92H */
NC_STATE_MC_PRDR = 199;   /* C7 */
NC_STATE_MC_NORM = 173;  /* ADH */
NC_STATE_MC_DI = 59;          /* 3BH */
NC_STATE_MC_NOT_VLD = 2; /* 02H */

/* Compatibility information of MC */

NC_NR_MC_COMP = 0;

/* Value for calibration switch so that no fault reaction will be performed after a ROM error for
level 2 calibration data (NOT TAKEN INTO ACCOUNT IN PRODUCTION SOWFTARE!) */

NC_ERR_SWI_MC_LEVEL_2_CAL = 165 /* A5H */

/* Value for calibration switch so that no fault reaction will be performed after a ROM error for
level 2 code (NOT TAKEN INTO ACCOUNT IN PRODUCTION SOWFTARE!) */

NC_ERR_SWI_MC_LEVEL_2_COD = 165 /* A5H */

/* Threshold for reset counter for irreversibly disabling of power stages */

NC_CTR_RST_MC_THD = 7;

/* Error codes on MC*/

NC_ERR_COD_MC_NOT_ERR = 0;
NC_ERR_COD_MC_HD = 1;
NC_ERR_COD_MC_CKS = 2;
NC_ERR_COD_MC_PAR = 3;
NC_ERR_COD_MC_CONF_DIF = 4;
NC_ERR_COD_MC_SOPC = 5;
NC_ERR_COD_MC_COMP_NR = 6;
NC_ERR_COD_MC_ROM_LEVEL_1 = 7;
NC_ERR_COD_MC_RAM_LEVEL_1 = 8;
NC_ERR_COD_MC_ROM_LEVEL_2 = 9;
```

```
NC_ERR_COD_MC_RAM_LEVEL_2 = 10;
NC_ERR_COD_MC_WRG_IDX = 11;
NC_ERR_COD_MC_WRG_RESP = 12;
NC_ERR_COD_MC_NOT_DEC = 13;
NC_ERR_COD_MC_MU_READY = 14;
NC_ERR_COD_MC_PFM_0_TOG = 16;
NC_ERR_COD_MC_PFM_1_TOG = 17;
NC_ERR_COD_MC_PFM_2_TOG = 18;
NC_ERR_COD_MC_PFM_3_TOG = 19;
NC_ERR_COD_MC_PFM_4_TOG = 20;
NC_ERR_COD_MC_PFM_5_TOG = 21;
NC_ERR_COD_MC_PFM_6_TOG = 22;
NC_ERR_COD_MC_NOT_VLD_TRAN = 23;
NC_ERR_COD_MC_PFM_0_RESP = 24;
NC_ERR_COD_MC_PFM_1_RESP = 25;
NC_ERR_COD_MC_PFM_2_RESP = 26;
NC_ERR_COD_MC_PFM_3_RESP = 27;
NC_ERR_COD_MC_PFM_4_RESP = 28;
NC_ERR_COD_MC_PFM_5_RESP = 29;
NC_ERR_COD_MC_PFM_6_RESP = 30;


/* Error codes on MU (for evaluation of error information: only used for error memory
management) */

NC_ERR_COD_MU_FCT_SPC_IST = 11; /* Wrong FS-IST answer */
NC_ERR_COD_MU_V_L = 13; /* Under-voltage detected on MU */
NC_ERR_COD_MU_V_H = 14; /* Over-voltage detected on MU */
NC_ERR_COD_MU_V_L_H = 15; /* Under- and over-voltage detected on MU */


/* Error indications */

NC_STATE_ERR_IDC_MC_NOT_PRES = 0;
NC_STATE_ERR_IDC_MC_REST_REQ = 1;
NC_STATE_ERR_IDC_MC_DI_REQ = 2;
NC_STATE_ERR_IDC_MC_PREV_REST= 3;

/* Debounce mechanism: indicators for anti-bounce counters */

NC_ABC_MC_HD = 0;
NC_ABC_MC_CKS = 1;
NC_ABC_MC_CONF_DIF = 2;
NC_ABC_MC_WRG_IDX = 3;
NC_ABC_MC_WRG_RESP = 4;
NC_ABC_MC_NOT_DEC = 5;
NC_ABC_MC_PFM_0_TOG = 6;
NC_ABC_MC_PFM_1_TOG = 7;
NC_ABC_MC_PFM_2_TOG = 8;
NC_ABC_MC_PFM_3_TOG = 9;
NC_ABC_MC_PFM_4_TOG = 10;
NC_ABC_MC_PFM_5_TOG = 11;
NC_ABC_MC_PFM_6_TOG = 12;
NC_ABC_MC_PFM_0_RESP = 13;
NC_ABC_MC_PFM_1_RESP = 14;
NC_ABC_MC_PFM_2_RESP = 15;
NC_ABC_MC_PFM_3_RESP = 16;
```

```
NC_ABC_MC_PFM_4_RESP = 17;
NC_ABC_MC_PFM_5_RESP = 18;
NC_ABC_MC_PFM_6_RESP = 19;

/* Debounce mechanism: initial values for anti-bounce counters */

NC_ABC_MC_HD_INI = 0;
NC_ABC_MC_CKS_INI = 0;
NC_ABC_MC_CONF_DIF_INI = 0;
NC_ABC_MC_WRG_IDX_INI = 0;
NC_ABC_MC_WRG_RESP_INI = 0;
NC_ABC_MC_NOT_DEC_INI = 0;
NC_ABC_MC_PFM_0_TOG_INI = 0;
NC_ABC_MC_PFM_1_TOG_INI = 0;
NC_ABC_MC_PFM_2_TOG_INI = 0;
NC_ABC_MC_PFM_3_TOG_INI = 0;
NC_ABC_MC_PFM_4_TOG_INI = 0;
NC_ABC_MC_PFM_5_TOG_INI = 0;
NC_ABC_MC_PFM_6_TOG_INI = 4;
NC_ABC_MC_PFM_0_RESP_INI = 0;
NC_ABC_MC_PFM_1_RESP_INI = 0;
NC_ABC_MC_PFM_2_RESP_INI = 0;
NC_ABC_MC_PFM_3_RESP_INI = 0;
NC_ABC_MC_PFM_4_RESP_INI = 0;
NC_ABC_MC_PFM_5_RESP_INI = 0;
NC_ABC_MC_PFM_6_RESP_INI = 0;

/* Debounce mechanism: increments for anti-bounce counters */

NC_ABC_MC_HD_INC = 2;
NC_ABC_MC_CKS_INC = 2;
NC_ABC_MC_CONF_DIF_INC = 2;
NC_ABC_MC_WRG_IDX_INC = 2;
NC_ABC_MC_WRG_RESP_INC = 2;
NC_ABC_MC_NOT_DEC_INC = 2;
NC_ABC_MC_PFM_0_TOG_INC = 2;
NC_ABC_MC_PFM_1_TOG_INC = 2;
NC_ABC_MC_PFM_2_TOG_INC = 2;
NC_ABC_MC_PFM_3_TOG_INC = 2;
NC_ABC_MC_PFM_4_TOG_INC = 2;
NC_ABC_MC_PFM_5_TOG_INC = 2;
NC_ABC_MC_PFM_6_TOG_INC = 2;
NC_ABC_MC_PFM_0_RESP_INC = 2;
NC_ABC_MC_PFM_1_RESP_INC = 2;
NC_ABC_MC_PFM_2_RESP_INC = 2;
NC_ABC_MC_PFM_3_RESP_INC = 2;
NC_ABC_MC_PFM_4_RESP_INC = 2;
NC_ABC_MC_PFM_5_RESP_INC = 2;
NC_ABC_MC_PFM_6_RESP_INC = 2;

/* Debounce mechanism: maximal values for anti-bounce counters */

NC_ABC_MC_HD_MAX = 6;
NC_ABC_MC_CKS_MAX = 6;
NC_ABC_MC_CONF_DIF_MAX = 6;
NC_ABC_MC_WRG_IDX_MAX = 6;
NC_ABC_MC_WRG_RESP_MAX = 6;
```

```
NC_ABC_MC_NOT_DEC_MAX = 6;
NC_ABC_MC_PFM_0_TOG_MAX = 6;
NC_ABC_MC_PFM_1_TOG_MAX = 6;
NC_ABC_MC_PFM_2_TOG_MAX = 6;
NC_ABC_MC_PFM_3_TOG_MAX = 6;
NC_ABC_MC_PFM_4_TOG_MAX = 6;
NC_ABC_MC_PFM_5_TOG_MAX = 6;
NC_ABC_MC_PFM_6_TOG_MAX = 6;
NC_ABC_MC_PFM_0_RESP_MAX = 6;
NC_ABC_MC_PFM_1_RESP_MAX = 6;
NC_ABC_MC_PFM_2_RESP_MAX = 6;
NC_ABC_MC_PFM_3_RESP_MAX = 6;
NC_ABC_MC_PFM_4_RESP_MAX = 6;
NC_ABC_MC_PFM_5_RESP_MAX = 6;
NC_ABC_MC_PFM_6_RESP_MAX = 6;

/* Initial value for communication cycle counter for healing of reset counter */

NC_CTR_COM_CYC_MC_DEC_RST_INI = 400;        /* 190H */

/* Maximal value for counter for disable request */

NC_CTR_MU_DI_MAX = 6;

/* Value of reset counter to be sent to MU in case of a disable request */

NC_CTR_RST_MC_DI = 15;

/* Number of communication cycles for temporary disabling after PREDRIVE communication
*/

NC_CYCNR_MC_DR_DI_TMP = 50; /* 520ms of disabling in total */

/* 10ms time between two communications */

NC_T_MC_COM_CYC = 10;

/* Number of PFM tasks */

NC_NR_MC_PFM = 7;

/* Number of PFM test cycle counters */

NC_NR_MC_PFM_TCC = NC_NR_MC_PFM - 1;


/* Array containing the maximal number of communication cycles after which toggling of bits
by the MU for the corresponding PFM tasks must have happened */

NC_TCC_MC_PFM_TOG_MAX[NC_NR_MC_PFM_TCC] = [4, 8, 8, 8, 8, 8];
NC_TCC_MC_PFM_TOG_6_MAX = 428;

/* Initialisation value for the PFM synchronisation byte used for toggling by PFM service
actions */

NLF_MC_PFM_SYN_BYTE_INI = 37;             /* 25H */

/* FS-IST: maximal question/answer number */

NC_TEST_REC_IDX_MAX_MON2 = 15;

/* FS-IST: array of expected questions */

NC_IDX_TEST_MC_FCT_SPC_IST[NC_TEST_REC_IDX_MAX_MON2] = [0, 33, 34, 3, 36,
5, 6, 39, 40, 9, 10, 43, 12, 45, 46];
```

/ FS-IST: array of corresponding answers */*

NC_IDX_RESP_MC_FCT_SPC_IST[NC_TEST_REC_IDX_MAX_MON2] = [19, 4, 21, 22, 7, 8, 25, 26, 11, 28, 13, 14, 31, 32, 49];

/ FS-IST: constant for invalid answer */*

NC_IDX_RESP_WRG_MC_FCT_SPC_IST = 62; */* 3EH */*

/ FS-IST: constant for invalid question number */*

NC_IDX_NR_WRG_MC_FCT_SPC_IST = 31;

/ FS-IST: value of test cycle counter when sending a wrong response shall be requested */*

NC_TCC_MC_IST_WRG_RESP_REQ = 1;

/ FS-IST: value of test cycle counter when reaction to wrong response shall be checked */*

NC_TCC_MC_IST_WRG_RESP_CHK = 3;

/ constant indicating a complement error of the value of the A/D conversion on the MC acquired in the infrastructure */*

NC_VP_MC_AN_DIG_MON_CPL_ERR = -0,000152587890625; */* FFFFH*/*

12.11.2 Project-specific configuration – Value assignment for configuration data

<u>FUNCTION DESCRIPTION:</u>

<u>General information:</u>

This section describes the value assignment for the project-specific configuration data of Processor Monitoring (e.g., configuration information for MU, etc.).

<u>Value assignment for configuration data of Processor Monitoring:</u>

/ Configuration */*

NLC_MU_MOD = 0; */* EMS */*
NLC_MU_IGN_KEY_ACQ = 1; */* HW acquisition */*
NLC_MU_POW_OFF_TMR_ENA = 0; */* POT inactive */*
NLC_MU_INJ_OFF_TMR_ENA = 1; */* IOT active */*

/ Common threshold for reset counter of MU and MC determining when error memory information shall be updated */*

NC_CTR_RST_THD_ERR_MEM = 1;

/ Maximal value for counter in check of MU readiness */*

NC_CTR_MU_READY_MAX = 3;

12.11.3 Project-specific configuration – Configuration of switch-off path check

<u>FUNCTION DESCRIPTION:</u>

<u>General information:</u>

This "sub-function" describes the configuration of the switch-off path check.

Description:

The function incorporates the following requirements:

G27 Configuration of switch-off path check

R27_1: The following system configuration is assumed:

- the **MC is only connected to power stage 1 (H-bridge)**; the check result for its switch-off path to power stage 0 (injection) shall always be 0;
- the **MU is connected to both power stages 0 and 1**.

R27_2: There are two final results:

- "**no engine start**" with representation NC_STATE_RESP_SOPC_DFCT (00H = 0) which shall simply trigger a fault reaction of processor monitoring by calling ACTION_ECM3_McFaultReaction() with error code NC_ERR_COD_MC_SOPC
- "**engine start**" with representation NC_STATE_RESP_SOPC_OK (FFH = 255) which shall prompt no reaction, i.e., the system starts up as usual.

R27_3 : The SOPC threshold NC_CTR_RST_MC_THD_SOPC for the reset counter is set to 0: the lookup for the final result shall always be performed in NC_RESP_SOPC_2[].

R27_4: The values in the array NC_RESP_SOPC_2[] shall correspond to the following start-up behaviour:

Index	**MC**		**MU**		**Result**
	PWS0	**PWS1**	**PWS0**	**PWS1**	
0	0	0	0	0	**no start**
1	0	0	0	1	**no start**
2	0	0	1	0	**no start**
3	0	0	1	1	**no start**
4	0	1	0	0	**no start**
5	0	1	0	1	**no start**
6	0	1	1	0	**start**
7	0	1	1	1	**start**
8	1	0	0	0	**no start**
9	1	0	0	1	**no start**
10	1	0	1	0	**no start**
11	1	0	1	1	**no start**
12	1	1	0	0	**no start**
13	1	1	0	1	**no start**
14	1	1	1	0	**no start**
15	1	1	1	1	**no start**

In the above table, the first column represents the index IDX_STATE_SOPC, while the next 4 columns represent the feedback from the two power stages (in the way that they are used to obtain IDX_STATE_SOPC), and the last column described the entry in NC_RESP_SOPC_2[] at index IDX_STATE_SOPC.

R27_5: The MU-SPOC result for power stage 0 obtained before triggering the PREDRIVE communication has to be re-checked before triggering the first NORMAL communication: the status of this particular disable line can only be determined unequivocally if tested when disabled and enabled; this does not cause any conflict with the MC-part of the SOPC since the MC is not connected to power stage 0. The

final check result of power stage 0 for the MU has to take into account both checks: only if the first result was "disabled" and the second was "enabled", the complete check was successful.

Value assignment for configuration data of SOPC:

/ Size of arrays for switch-off path check */*
NC_NR_RESP_SOPC = 16;
/ always use NC_RESP_SOPC_2, never NC_RESP_SOPC_1 for lookup of the final result */*
NC_CTR_RST_MC_THD_SOPC = 0;
/ SOPC result: no engine start possible */*
NC_STATE_RESP_SOPC_DFCT = 0;
/ SOPC result regular engine start possible */*
NC_STATE_RESP_SOPC_OK= 255;
/ not used: all entries are equal to NC_STATE_RESP_SOPC_DFCT */*
NC_RESP_SOPC_1[NC_NR_RESP_SOPC] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
/ see table and NCs for results above */*
NC_RESP_SOPC_2[NC_NR_RESP_SOPC] = [0, 0, 0, 0, 0, 0, 255, 255, 0, 0, 0, 0, 0, 0, 0, 0]
/ Default value for IDX_STATE_SOPC: no switch-off path working */*
NC_IDX_STATE_SOPC_DFCT = 0;
/ Diagnostic result from power stage "driver disabled" */*
NC_STATE_RESP_DR_DI = 0;
/ Diagnostic result from power stage "driver enabled" */*
NC_STATE_RESP_DR_ENA = 1;
/ Default diagnostic result from power stage ("status undefined") */*
NC_STATE_RESP_DR_DFT = 3;

Application conditions:

Initialisation:	*/* Using the RAM check service actions */* LV_MC_ST_LIH_SOPC = 0;
Recurrence:	The **initialisation** shall be **executed after each power-up or reset**.

12.11.4 Project-specific configuration – Error memory service function

General information:

The function incorporates the following requirements:

G43 Error memory service function

The action ACTION_ECM3_McErrmService() shall be called when all the necessary information is available. When the action is called, it is first tested whether an update of the error history was performed right before the action was called. Since this is only done in case there currently is an error present, this means if no update was done the error symptom ERR_SYM_MON_3 can be set to "NO_SYM"; ERR_SYM_MON_3 consists of four bits:

Bit3 (SYM_3)	Bit2 (SYM_2)	Bit1 (SYM_1)	Bit0 (SYM_0, LSB)
8	4	2	1
FS-IST error on MU	SOPC error/ unspecific level 3 error	RAM error on MC	ROM error on MC

For every fault, environmental conditions are stored in the environmental data variables ENVD_x_MON_3; the stored content is chosen to help analysing the fault scenario afterwards; more information can be found in the formula section below.

Application conditions:

Initialisation: */* Initialise the interfaces to ERRM */*

LV_ERR_MON_3 = 0;
LV_CDN_DIAG_MON_3 = 0;
LV_END_DIAG_MON_3 = 0;
ERR_SYM_MON_3 = NO_SYM */* No symptom */*;
ENVD_0_MON_3 = 0;
ENVD_1_MON_3 = 0;
ENVD_2_MON_3 = 0;
ENVD_3_MON_3 = 0;

Recurrence: The **initialisation** shall be **performed after each power-up and reset** and **at FMY reset**.

Activation: This **sub-function** is **called**.

Description of ACTION_ECM3_McErrmService

Description:

Syntax: ACTION_ECM3_McErrmService(

IN <cpt_act>,
IN <err_cod_act>,
IN <err_idc_act>,
IN <ctr_rst_act>,
IN <cpt_old>,
IN <err_cod_old>,
IN <err_idc_old>,
IN <ctr_rst_old>
)

Parameter (IN):	cpt_act	component for current error information (1=MC, 0=MU)
Parameter (IN):	err_cod_act	current error code
Parameter (IN):	err_idc_act	current error indication
Parameter (IN):	ctr_rst_act	current value of reset counter
Parameter (IN):	cpt_old	component for previous error information (1=MC, 0=MU)
Parameter (IN):	err_cod_old	previous error code
Parameter (IN):	err_idc_old	previous error indication

Parameter (IN): ctr_rst_old previous value of reset counter

Short description: This action receives the current error history and provides the interface to the error memory management.

Formula section:

```
IF ( (ctr_rst_act > 0) OR ((cpt_act = 0) AND ("err_cod_act indicates voltage monitoring
error")) ) /* reset counter of current error greater than 0 OR current error is an MU error and
the current error is a voltage monitoring error */
THEN
        ACTION_SetRstInfo_Ecm3;          /* set reset-cause -> RST_DET */
FI
IF (LV_ERR_MON_3 = 1)          /* previously detected an error */
THEN /* Initialise data to be able to potentially update the error symptom */
        LV_ERR_MON_3 = 0;
        ERR_SYM_MON_3 = NO_SYM;
        LV_CDN_DIAG_MON_3 = 1;
        LV_END_DIAG_MON_3 = 1;
        Deliver the result to Error Management;
FI
IF ("no update of the error history has been performed before this call")
    /* can be checked by comparing env. data to actual parameters */
THEN
       ERR_SYM_MON_3 = NO_SYM;
/* Otherwise attempt to set filtered error information ERR_SYM_MON_3 according
evaluation of the given parameters */
ELSEIF ((cpt_act = 1) AND (ctr_rst_act > NC_CTR_RST_THD_ERR_MEM ))
/* current error is an MC error and reset counter above threshold */
THEN
    IF ("err_cod_act indicates a ROM error (level 1 or level 2)")
    THEN
         ERR_SYM_MON_3 = SYM_0;          /* ROM error on MC */
    ELSEIF ("err_cod_act indicates a RAM error (level 1 or level 2)")
    THEN
         ERR_SYM_MON_3 = SYM_1;          /* RAM error on MC */
         LV_ERR_RAM_MON_3 = 1;
    ELSE /* Unspecific error */
         ERR_SYM_MON_3 = SYM_2
    FI
```

```
ELSEIF ((cpt_act = 0) AND ((ctr_rst_act > NC_CTR_RST_THD_ERR_MEM) OR
("err_cod_act indicates over-voltage ")) /* current error is an MU error and reset counter
above threshold OR over-voltage was detected */

THEN

     IF ("err_cod_act indicates FS-IST error detected by the MU")

     THEN

              ERR_SYM_MON_3 = SYM_3;        /* FS-IST error on MU */

     ELSE

              ERR_SYM_MON_3 = SYM_2;        /* unspecific error */

     FI

  FI

  /* Now flags for error memory management can be set */

  IF (ERR_SYM_MON_3 = NO_SYM)

     LV_ERR_MON_3 = 0; /* no error is present */

  ELSE

     LV_ERR_MON_3 = 1; /* error signaled by ERR_SYM_MON_3 is present */

     /* Set environmental data, see table for exact bit positions */

     ENVD_0_MON_3 = "combine cpt_act, err_idc_act and err_cod_act into one byte";
     ENVD_1_MON_3 = ctr_rst_act; /* this could include encoded additional information */
     ENVD_2_MON_3 = "combine cpt_old, err_idc_old and err_cod_old into one byte";
     ENVD_3_MON_3 = ctr_rst_old; /* this could include encoded additional information */
```

	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
ENVD_0_MON_3	Current error: 0 for MU error 1 for MC error	Current error: error indication 0 = no error 1 = restart request 2 = DISABLE request 3 = restart was requested		Current error: error code (see tables in "Processor monitoring general")				
ENVD_1_MON_3	Current error: additional information (test results of switch-off paths, encoded wrong header byte, ABC status, cf. G41 "Update of error memory information with MC data")				Current error: reset counter			
ENVD_2_MON_3	Old error: 0 for MU error 1 for MC error	Old error: error indication 0 = no error 1 = restart request 2 = DISABLE request 3 = restart was requested		Old error: error code (see tables in "Processor monitoring general")				
ENVD_3_MON_3	Old error: additional information (test results of switch-off paths, encoded wrong header byte, ABC status, cf. G41 "Update of error memory information with MC data")				Old error: reset counter			

```
  FI

  LV_CDN_DIAG_MON_3 = 1;

  LV_END_DIAG_MON_3 = 1;                 { Diagnosis is finished }

  Deliver the result to Error Management;
```

Configuration for diagnostic symptoms:

Diagnostic MON_3	Symptom description	Symptom	Filter type
Diagnoses of Processor Monitoring	ROM error on MC	*SYM_0*	NO
	RAM error on MC	*SYM_1*	
	SOPC or unspecific error	*SYM_2*	
	FS-IST error on MU	*SYM_3*	

Note: it is recommended to configure MON_3 to "CARB CC"
Note: it is recommended to calibrate ENVD_x_MON_3 to be stored in the error memory when a MON_3 error occurs (ID_ERR_ENVD_MON_3).

12.11.5 Project-specific configuration – Initialisation of power stages after disabling by Processor Monitoring

FUNCTION DESCRIPTION:

General information:

Not used since this is done via polling in the relevant functions.

12.11.6 Project-specific configuration – Project-specific transition condition for the transition from INIT to CONFIG state

FUNCTION DESCRIPTION:

General information:

In order to prevent a start of Processor Monitoring SW while the "environmental condition" inside the ECU (e.g., the supply voltage for the H-bridge, which is supplied via the main relay) are not yet in a defined state, the transition from INIT to CONFIG (where the first communication is performed) can only be performed under specific conditions.

This functions describes a project-specific condition for the transition to be possible; other (generic) conditions might be defined in other specifications.

Description:

This function incorporates the following requirements:

G52 Project-specific transition condition for the transition from INIT to CONFIG state

R52_1: If LV_DIAG_END_RLY_MAIN_DLY = 1 **AND** LV_RLY_MAIN_DLY_ERR = 0, the transition from INIT to CONFIG shall be possible.

R52_2: Otherwise, the transition shall not be possible.

Application conditions:

Initialisation: -

Recurrence: The **function** shall be **called by G51 "State machine on MC" for deciding if the transition from INIT to CONFIG can be performed**.

Activation: STATE_MC = NC_STATE_MC_INI *(only active in INIT state)*

Deactivation: Otherwise

12.11.7 Project-specific configuration – Acquisition of ignition key information on MC

FUNCTION DESCRIPTION:

General information:

In each NORMAL and DISABLE communication, the MC sends information about the status of the ignition key signal to the MU.

Usually, the ignition key information on the MC is available in LV_IGK; however, in some instances, a pre-processing step could possible/necessary (e.g., if the signal is available redundantly, it can be checked for plausibility and then the monitored value can be sent to the MU; another application could be to return the MU value in HW acquisition mode back to the MU so that only the pin value on the MU will determine the MU-internal ignition key information).

To support any such pre-processing, this project-specific function allows to freely define the value of LV_MC_IGN_KEY before it is sent to the MU.

Description:

This function incorporates the following requirements:

G54 Acquisition of ignition key information on MC

R54_1: The value of LV_MC_IGN_KEY shall be obtained by copying LV_MU_IGN_KEY.

Application conditions:

Initialisation: LV_MC_IGN_KEY = 0

Recurrence: The **initialisation** shall be **performed after each power-up and reset**.

The **function itself** shall be activated before G25 in NORMAL and DISABLE.

Activation: (STATE_MC = NC_STATE_MC_NORM) **OR**
(STATE_MC = NC_STATE_MC_DI)
(active in NORMAL and DISABLE state)

Deactivation: *Otherwise*

Formula section:

/ send back MU value to MU to avoid problems with wrong detection thresholds */*

LV_MC_IGN_KEY = LV_MU_IGN_KEY

12.11.8 Project-specific configuration – Extension of power-latch phase for Standard ROM check

FUNCTION DESCRIPTION:

General information:

Since the Standard ROM check (G8.1) is calculated in the background, it is not pre-determined how long it will take to complete; however, there is the requirement that the ROM check has to be completed at least once per driving cycle. Consequently, in order to fulfil the mentioned requirement, the power latch phase (the phase after the ignition has been switched off, KL15 = 0, but before the ECU is switched off) shall be extended variably for (at least) as long the duration of one complete checksum calculation.

Description:

This function incorporates the following requirements:

G64 Extension of power-latch phase for Standard ROM check

Application conditions:

Initialisation: -

Recurrence: The **function itself** shall be activated via the action exported calls ACTION_ECM3_McLockPwlResources() and ACTION_ECM3_McUnlockPwlResources().

Description of ACTION_ECM3_McLockPwlResource

Description:

Syntax: ACTION_ECM3_McLockPwResource()

Formula section:

"set power-latch lock condition corresponding to NC_PWL_LOCK_CDN_ROM_CHK in PWL_LOCK_CDN"

Description of ACTION_ECM3_McUnlockPwlResource

Description:

Syntax: ACTION_ECM3_McUnlockPwResource()

Formula section:

"reset power-latch lock condition corresponding to NC_PWL_LOCK_CDN_ROM_CHK in PWL_LOCK_CDN"

12.11.9 Project-specific configuration – Handling of injection-off timer functionality

FUNCTION DESCRIPTION:

General information:

The injection-off timer inside the MU can disable the injection power stage. This fact shall be indicated to other SW parts via the flag LV_INJ_OFF_TMR_INJ_ENA.

Description:

This function incorporates the following requirements:

G65 Handling of injection-off timer functionality

R65_1: The value of LV_INJ_OFF_TMR_INJ_ENA shall be obtained by checking a 0->1 transition of LV_MU_IGN_KEY between the last two recurrences, see formula section below (*due to unreliable disable information from the MU, the only reliable decision base is the ignition key information*).

NOTE: the ignition key signal is already low while the timer is active but not yet disabling the injection so the injection will be disabled immediately after the 1->0 transition of the ignition key signal!

NOTE: this is only a temporary solution and should be changed once the latest version of the MU is consistently used in the HW; then it will be sufficient to copy the value of LV_INJ_OFF_TMR_ENA_INJ_TMP to LV_INJ_OFF_TMR_ENA_INJ!

Application conditions:

Initialisation:	LV_INJ_OFF_TMR_ENA_INJ = 0 LV_MU_IGN_KEY_OLD = 0
Recurrence:	The **initialisation** shall be **performed after each power-up and reset**. The **function itself** shall be activated after G22.
Activation:	(STATE_MC = NC_STATE_MC_CONF) **OR** (STATE_MC = NC_STATE_MC_PRDR) **OR** (STATE_MC = NC_STATE_MC_NORM) */* only active in CONFIG; PREDRIVE, or NORMAL state */*
Deactivation:	otherwise */* inactive in INIT and DISABLE state: here, LV_ERR_TMP_MU_MC is set anyway and so the power stages are disabled regardless of the IOT */*

Formula section:

/ Enable injection if value of ignition key signal on MU, LV_MU_IGN_KEY, of current and last recurrence is high because then the IOT has been reliably reset inside the MU (wait-state needed due to delay stemming from SPI communication) */*

LV_INJ_OFF_TMR_INJ_ENA = LV_MU_IGN_KEY_OLD **AND** LV_MU_IGN_KEY;

/ Save current value for next recurrence */*

LV_MU_IGN_KEY_OLD = LV_MU_IGN_KEY;

13 ECU Memory

Table of Contents

Index

13.1 Non volatile data

Input data:

ABC_LAM_CYL_SEL_ERR	Absch_korr	ANG_PSTE_END_LE_AD	ANG_PSTE_END_RI_AD
B_lvs_neustart			
Batt_class	Bosbtvfbk	Bosconf	Bosfxid2
Bosfxid2r	Bosid2	Bosid2r	Bosjziel
Bosjzielr	Bosmziel	Bosmzielr	Bosprog2
Bosprog2r	Bosres	Bosrlsm	Bosrw2
Bosrw2r	Bszsi	C_ANG_PSTE_END_LE_INI	C_ANG_PSTE_END_RI_INI
CHK_STAMP	CL_MMV_SAE	CONF_SOF_SWI	CPPWM_ADD_AD_MEM
CRK_CYL_LAM_DELTA_INI[NC_CBK_EX_NR]	CTR_CHK_FUC_OPEN	CTR_CNL_ROUGH_LEAK_MES_FUC	CTR_CNL_SMALL_LEAK_MES
CTR_CNL_SMALL_LEAK_MES_VIRT	CTR_COLD_ST	CTR_COLD_ST_NT	CTR_COLD_ST_NT
CTR_CUR_DMTL_REF_DIF_MAX	CTR_DC_ENA_CHK_FUC	CTR_ERR_OBD_DIAG_CYL_BAL_ER	CTR_ERR_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]
CTR_IGA_DIF_MAX_KNK[NC_CYL_NR]	CTR_KM_LAM_ADJ_LAM_AD_CUS[NC_CBK_EX_NR]	CTR_KM_LAM_ADJ_TMP_LAM_AD_CUS[NC_CBK_EX_NR]	CTR_KM_N_MAX
CTR_MIS_DC_CYL[NC_CYL_NR]	CTR_MIS_DC_MMV_CYL[NC_CYL_NR]	CTR_MIS_DET	CTR_MIS_DET_CYL[NC_CYL_NR]
CTR_MIS_TOT_DC	CTR_MIS_TOT_NVMY	CTR_N_MAX	CTR_NT_AGI_AD_CMPL_SUM
CTR_NT_AGI_AD_CMPL_SUM	CTR_NT_AGI_SO2P_FQ	CTR_PHA_SHIFT_AD_TRIG[NC_CBK_EX_NR]	CTR_REP_ROUGH_LEAK_MES_FUC
CTR_SAVE_SWT_LS_DOWN[NC_CBK_EX_NR]	CTR_SEG_AD_ER	CTR_SEG_MIS_ACT	CTR_ST_CHK_CHK_LIH_1_ERR
CTR_ST_CHK_CHK_LIH_1_READY	CTR_STC_TECU_1	CTR_STC_TECU_2	CTR_STC_TECU_3
CTR_STC_TECU_4	CTR_STC_TECU_5	CTR_STC_TECU_6	CTR_STC_TECU_7
CTR_STC_TECU_8	CTR_STOP_FSD	CTR_SWI_AFS_MON	CTR_TCO_ST_DMTL
CTR_TOIL_MAX	CTR_TOT_CAT_DIAG[NC_CBK_EX_NR]	CTR_TOT_MIS	CTR_TPS_JAM_DET_ACT
CTR_VIMPWM_1_EDGE	CTR_VIMPWM_2_EDGE	CUR_DMTL_COR_FIL_EOL	CUR_DMTL_REF_LEAK_EOL
CUR_DMTL_ROUGH_LEAK_MIN_EOL	CYL_ID_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	D_soc	DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]
DELTA_CRK_DIF_MAX_ER	Dfds[16]	DIST_DMTL	DIST_O2_STC_OLD
DIST_TOIL_MAX	ECU_STATE	EFF_CAT_DIAG[NC_CBK_EX_NR]	EFF_CAT_DIAG_HOM[NC_CBK_EX_NR]
ER_STD_MMV_BAL[NC_CYL_NR]	ER_STND_MMV_BAL[NC_CYL_NR]	ER_STND_MMV_STD_BAL[NC_CYL_NR]	F_atlad
FAC_ACR_SLOP	FAC_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	FAC_DYN_LSL_DIAG_SAE[NC_CBK_EX_NR]	FAC_DYN_LSL_DIAG_TOL_SAE[NC_CBK_EX_NR]
FAC_EGY_PWM_AD[NC_CYL_NR]	FAC_FCO_KWP	FAC_GR_MMV_AV	FAC_H_RNG_LAM_AD[NC_CBK_EX_NR]
FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	FAC_LAM_CYL_SEL_ADJ_CST[NC_CYL_NR]	FAC_LAM_CYL_SEL_ADJ_H_RNG[NC_CYL_NR]	FAC_LAM_CYL_SEL_ADJ_L_RNG[NC_CYL_NR]
FAC_LAM_CYL_SEL_LAM_AD_CUS[NC_CYL_NR]	FAC_LAM_TCO_A[NC_CBK_EX_NR]	FAC_LAM_TCO_B[NC_CBK_EX_NR]	FAC_LAM_TCO_C[NC_CBK_EX_NR]
FAC_LAM_TCO_D[NC_CBK_EX_NR]	FAC_LAM_TCO_E[NC_CBK_EX_NR]	FAC_LAM_TCO_MIN[NC_CBK_EX_NR]	FAC_LSL_GAIN_AD[NC_CBK_EX_NR]
FAC_MFF_ADD_FAC_LAM_AD[NC_CBK_EX_NR]	FAC_POW_MNG_VST_CNS[15]	FAC_TI_AD_ER_BAL[NC_CYL_NR]	FCO_AV_DMTL

FTL_AV	FTL_OLD	GEAR_EF_N_MAX	GEAR_TOIL_MAX
Ibsderrs1	Ibsderrs2	Ibspreco2	Igenk
Igrinfo[30]	Ikurz_ogr	Ikurz_ugr	IPLSL_MMV_VLD_FCUT[NC_CBK_EX_NR]
IPLSL_VARI_VLD_FCUT[NC_CBK_EX_NR]	Iwakeupgr	Km_st_1	Ktupcsctr
LAMB_DELTA_AD_LAM_ADJ[NC_CBK_EX_NR]	LAMB_THD_VPLSL_LIM[NC_CBK_EX_NR]	Leminfo[40]	LV_ACR_AD_REQ
LV_AT	LV_CHK_DMTL_CUR_DMTL_VLD_VIRT	LV_CHK_FUC_OPEN	LV_CHK_FUC_OPEN_ACT
LV_CRU_OFF_IRR	LV_CS_N_MAX	LV_CS_TOIL_MAX	LV_CTR_CNL_SMALL_LEAK_MES
LV_CTR_DC_CHK_FUC	LV_CUR_DMTL_REF_DIF_MAX	LV_CUR_DMTL_THD_DIF_MES	LV_CYL_BAL_ER_AD_ADD_EOL
LV_CYL_BAL_ER_AD_FAC_EOL	LV_CYL_BAL_LAM_AD_ADD[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_ADD_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_DC
LV_CYL_BAL_LAM_AD_EOL	LV_CYL_BAL_LAM_AD_FAC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_FAC_H_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_FAC_L_DC[NC_CBK_EX_NR]
LV_CYL_BAL_LAM_SEL_AD_COLD[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_SEL_AD_COLD_DC	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL	LV_CYL_BAL_LAM_SEL_AD_HOT_DC
LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	LV_CYL_BAL_LAM_SEL_AD_RNG_H[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_SEL_AD_RNG_L
LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[NC_CBK_EX_NR]	LV_DET_FUC_OPEN	LV_DET_NO_ROUGH_LEAK
LV_DET_NO_SMALL_LEAK	LV_DET_REFU	LV_DET_SMALL_LEAK	LV_DIST_DET_ROUGH_LEAK
LV_DMTL_STOP	LV_ECRAS_DOWN_DET_SET	LV_ENA_CHK_FUC_REFU	LV_ENA_FUC
LV_ENA_SMALL_LEAK_MES	LV_ERR_DET_DMTL_MAX	LV_ERR_DET_DMTL_MIN	LV_ERR_DET_ROUGH_LEAK_MIN
LV_ERR_FTL	LV_ERR_FTL_PLAUS	LV_ERR_LAM_CYL_SEL	LV_ERR_SIG_DMTL
LV_ERR_TCO_2_PREL	LV_ERR_TCO_PREL_DET	LV_FTL_DIAG	LV_FUC_OPEN
LV_IGK	LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[NC_CBK_EX_NR]	LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[NC_CBK_EX_NR]	LV_LSL_FIRST_GAIN_AD[NC_CBK_EX_NR]
LV_NT_AFS_REQ_AGI	LV_NT_AFS_REQ_AGI_TMP_3	LV_NT_O2_STC_LIM	LV_NT_O2_STC_REQ
LV_NT_O2_STC_REQ	LV_NT_O2_STC_VLD	LV_NT_O2_STC_VLD	LV_REFU_DMTL
LV_REP_CHK_FUC	LV_ROUGH_LEAK_SUSP_CHK_FUC	LV_ROUGH_LEAK_SUSP_SET	LV_SDR_DMTL
LV_SEG_AD_AVL_ER	LV_SEG_AD_LIM_ER	LV_SENS_BAT_SMT_DET	LV_SO2P_REQ_1
LV_SO2P_REQ_1	LV_SO2P_REQ_2	LV_SO2P_REQ_2	LV_SO2P_REQ_FQ
LV_STALL	LV_T_DMTL_MAX	LV_TPS_AD_REQ	LV_VAR_4WD
LV_VAR_ACIN	LV_VAR_AEB	LV_VAR_AMT	LV_VAR_ARS
LV_VAR_ASR	LV_VAR_ASR_3	LV_VAR_ASR_4	LV_VAR_BN_EFP
LV_VAR_BN_GEAR_REV	LV_VAR_BN_LDM	LV_VAR_BN_LTG_HDLP_L	LV_VAR_BN_MSW
LV_VAR_BN_TRL	LV_VAR_DCC	LV_VAR_EBOX_CFA	LV_VAR_ECRAS_DOWN
LV_VAR_ECRAS_UP	LV_VAR_EF	LV_VAR_EFP_CRASH	LV_VAR_ETCU
LV_VAR_ETCU_3	LV_VAR_ETCU_SPT	LV_VAR_LSH_DOWN	LV_VAR_LSH_UP
LV_VAR_MAF	LV_VAR_MAF_LEARNT	LV_VAR_MSW	LV_VAR_PBR
LV_VAR_PSTE	LV_VAR_PSTE_2	LV_VAR_PSTE_3	LV_VAR_RLY_ACCOUT
LV_VAR_RLY_ST	LV_VAR_SOF	LV_VAR_TCT	LV_VB_DIF_MAX
M6_CTR_CNL_SMALL_LEAK_MES	M6_CUR_DMTL_COR_FIL_CID18	M6_CUR_DMTL_COR_FIL_CID19	M6_CUR_DMTL_DMTLS_TEST
M6_CUR_DMTL_REF_LEAK	M6_CUR_DMTL_ROUGH_LEAK_END	M6_CUR_DMTL_ROUGH_LEAK_LEN_END	M6_CUR_DMTL_SMALL_LEAK_END

M6_CUR_DMTL_THD_DMTLS_TEST	M6_CUR_DMTL_THD_ROUGH_LEAK	M6_CUR_DMTL_THD_ROUGH_LEAK_LEN	MAF_DIAGCPS_SAE
MAF_DIAGCPS_THD_SAE	MAF_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	MFF_ADD_CYL_LAM_COR[NC_CYL_NR]	MFF_ADD_LAM_AD[NC_CBK_EX_NR]
Msa_arravrs[60]	Msa_indexrs	Msainfo[50]	Msastz
Msastzmsa	Msnlgofs_tmp	N_GRD_N_MAX	N_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]
N_KWP_OFS_ACC_DRI_NVMY	N_KWP_OFS_DRI_NVMY	N_KWP_OFS_NVMY	N_KWP_OFS_VB_NVMY
N_MAX	N_TOIL_MAX	NOX_AD_FAC	NOX_AD_FAC_SNG[NC_NT_NR]
NOX_OFS_LOAD[NC_NOX_SENS_CONF]	NOX_OFS_PUC[NC_NOX_SENS_CONF]	NR_PAT_SCC_N_MAX	NT_AGI
NT_AGI	NT_AGI_SO2P_FQ_SUM	NT_AGI_SUL	NT_AGI_SUL
NT_AGI_SUL_SNG[NC_NT_NR]	NT_AGI_THERMO	NT_AGI_THERMO	NT_AGI_THERMO_SNG[NC_NT_NR]
NT_O2_STC_AD	NT_O2_STC_AD	NT_O2_STC_OFS	NT_O2_STC_OFS
NT_SUL_32[NC_NT_NR]	NT_SUL_32[NC_NT_NR]	NT_SUL_H_32[NC_NT_NR]	NTL
Oz_kvbog	Oz_kvbsm_ul	Oz_lf1c	Oz_lf1t
Oz_lf2c	Oz_lf2t	Oz_nivlangt	Oz_oelkm
Oz_permbog	Oz_permex	Oz_permlow	Pminfo1[37]
Pminfo2[29]	Pmsv[15]	PSN_EDGE_AD_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	PSN_EDGE_AD_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]
PV_AV_N_MAX	PV_AV_TOIL_MAX	Qv_cdherst_1	Qv_cdherst_2
Qv_cdherst_3	Qv_cdherst_4	Qv_cdherst_5	Qv_cdherst_6
Qv_cdherst_7	Qv_cdherst_8	Qv_h2o	Qv_h2oquali
Qv_h2ostatus	Qv_nv_ezm	Qv_nv_start	Qv_nv_zh
Qv_out_1	Qv_out_2	Qv_out_3	Qv_out_4
Qv_out_5	Qv_out_m	Qv_quali_1	Qv_quali_2
Qv_quali_3	Qv_quali_4	Qv_quali_5	Qv_quali_m
Qv_status	Qv_td1	Qv_td2	Qv_td3
Qv_td4	Qv_td5	Qvc_status_1	Qvc_status_2
Qvc_status_3	Qvc_status_4	R_IT_OBD_LSH_DOWN[NC_CBK_EX_NR]	R_IT_THD_OBD_LSH_DOWN[NC_CBK_EX_NR]
SEG_AD_MMV_ER[NC_CYL_NR]	St_bns	St_deavns2	St_devxdetec
St_dgenerrst_md1	St_dgenerrst_md2	St_dgengrenz1	St_dgengrenz1_md1
St_dgengrenz1_md2	St_dgengrenz2	St_dgengrenz2_md1	St_dgengrenz2_md2
St_dgengrenzerr	St_dgengrenzerr_md1	St_dgengrenzerr_md2	St_dgengrenznz
St_dgengrenznz_md1	St_dgengrenznz_md2	St_dgenub1	St_dgenub1_md1
St_dgenub1_md2	St_dgenub2	St_dgenub2_md1	St_dgenub2_md2
St_dgenuberr	St_dgenuberr_md1	St_dgenuberr_md2	St_dgenubnz
St_dgenubnz_md1	St_dgenubnz_md2	St_dgenugen1	St_dgenugen1_md1
St_dgenugen1_md2	St_dgenugen2	St_dgenugen2_md1	St_dgenugen2_md2
St_dgenugenerr	St_dgenugenerr_md1	St_dgenugenerr_md2	St_dgenugennz
St_dgenugennz_md1	St_dgenugennz_md2	St_dibs0	St_ecostat2
St_eisyad_read	St_elu	St_igrinnv	St_igroutnv
St_lvs_in_nv			
St_ngang	St_pmi_nv	St_spa2	St_ubw
STATE_CMB_CTL_NVMY_KWP	STATE_CRU_OFF_IRR	STATE_CRU_OFF_REV	STATE_DMTL_EOL
STATE_LAMB_CYL_SEL_CQ_SLOP[NC_CBK_EX_NR]	STATE_TBL_ALTER[7][5][5]	SUM_DIAG_DIAGCPS_SAE	SUM_NTL_CYC
SUM_NTL_CYC	SUM_TQI_REQ_LIM	T_AFL_SUM	T_AST_REST
T_CYL_BAL_LAM_AD_DC	T_CYL_BAL_LAM_SEL_DC	T_N_MAX	T_REL_CAN_ES
T_REL_CAN_ES_2	T_STATE_TBL_ALTER	T_SUM_N_MAX	T_TQI_REQ_LIM_MAX

TAM_TOIL_MAX	TCO_A_LAM_AD_WUP[NC_CBK_EX_NR]	TCO_B_LAM_AD_WUP[NC_CBK_EX_NR]	TCO_C_LAM_AD_WUP[NC_CBK_EX_NR]
TCO_D_LAM_AD_WUP[NC_CBK_EX_NR]	TCO_E_LAM_AD_WUP[NC_CBK_EX_NR]	TCO_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	TCO_MIN_LAM_AD_WUP[NC_CBK_EX_NR]
TCO_REST	TCO_STOP	TCO_STOP	TCO_TOIL_MAX
TEG_DYN_STOP	TEG_DYN_STOP	TEMP_MDL_IGC	TIA_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]
Tmot_abstell	Tn_abstellm	TOIL_MAX_WARN	TOIL_STOP
TOIL_THD_TOIL_MAX	TPS_AD_SLOP_GAIN_1	TPS_LIH_1	TPS_LIH_2
TQ_DIF_IS_AD	TQ_DIF_IS_AD_ACC	TQ_DIF_IS_AD_ACC_1	TQ_DIF_IS_AD_ACC_1_OPM_1
TQ_DIF_IS_AD_ACC_1_OPM_2	TQ_DIF_IS_AD_ACC_CONV	TQ_DIF_IS_AD_ACC_CONV_OPM_1	TQ_DIF_IS_AD_ACC_CONV_OPM_2
TQ_DIF_IS_AD_ACC_NEUT	TQ_DIF_IS_AD_ACC_NEUT_OPM_1	TQ_DIF_IS_AD_ACC_NEUT_OPM_2	TQ_DIF_IS_AD_CONV
TQ_DIF_IS_AD_CONV_1	TQ_DIF_IS_AD_CONV_1_OPM_1	TQ_DIF_IS_AD_CONV_1_OPM_2	TQ_DIF_IS_AD_CONV_NEUT
TQ_DIF_IS_AD_CONV_NEUT_OPM_1	TQ_DIF_IS_AD_CONV_NEUT_OPM_2	TQ_DIF_IS_AD_OPM_1	TQ_DIF_IS_AD_OPM_2
TQI_AV_TOIL_MAX	TQI_REQ_LIM_DIF_MAX	TRT	TRT_N_MAX
TTIP_MES_LS_UP_OBD_LSH_UP[NC_CBK_EX_NR]	Tumg_abst	Tumg_battab	Tvneutral
V_ACR_AD_BOL	V_ACR_AD_BOL_0	V_ACR_AD_TOL	V_MAP_AD
V_TPS_AD_EL_BOL_1	V_TPS_AD_EL_BOL_2	V_TPS_AD_LIH_1	V_TPS_AD_LIH_2
VLS_DIF_SAVE_DIAG_LS_DOWN[NC_CBK_EX_NR]	VLS_DOWN_PUC_SAVE[NC_CBK_EX_NR]	VS_MAX_SEL_EXT_REQ	VS_N_MAX
VS_STEP[15]	VS_TOIL_MAX	Vsa_adp_ext	Vse_adp_ext
Zbibs	Zr_lvs_0	Zr_lvs_1	Zr_lvs_2
Zr_lvs_3			
Zr_lvs_ll_reakt	Zr_lvssekt_0	Zr_lvssekt_1	Zr_lvssekt_2
Zr_lvssekt_3	Zr_lvssekt_4	Zr_lvssekt_5	Zr_lvssekt_6
Zr_lvssekt_7	Zr_lvssekt_8	Zrbosmld	

Import action

void klann_clrad(void)

FUNCTION DESCRIPTION:

General information:

Non volatile datas are stored in the E2PROM. Non volatile datas are:

- All adaptation values
- The debounced errors in the dynamic error memory (max. 10)
- Some datas from the last engine run

This specification describes the non volatile data handling in connection with the E2PROM and how to initialize those values by calibration system.
The input values for Linear-Lambda functions and Error memory (see:"Selective adaptation ranges") are not considered in the Input data above.

Application conditions:

Initialisation: see "READ Non-Volatile-Data to the E2PROM"

Recurrence: at state READ: Once
at state WRITE: Once

	at state	DELETE:	1s
Activation:	at state	READ:	after ECU reset
	at state	WRITE:	ECU_STATE = PWL
	at state	DELETE:	LV_IGK = 1

13.1.1 READ Non-Volatile-Datas from the E2PROM

Description:

The update of the non-volatile-datas will be executed at the system startup after Ignition-key-ON.

Every functionality will be read seperately from the E2PROM. If a checksum fault or a fail of hardware communication (SPI) occurs, then all non-volatile-datas will be initialized (Adaptation values with default or calibratable values and all debounced and active errors with 0).

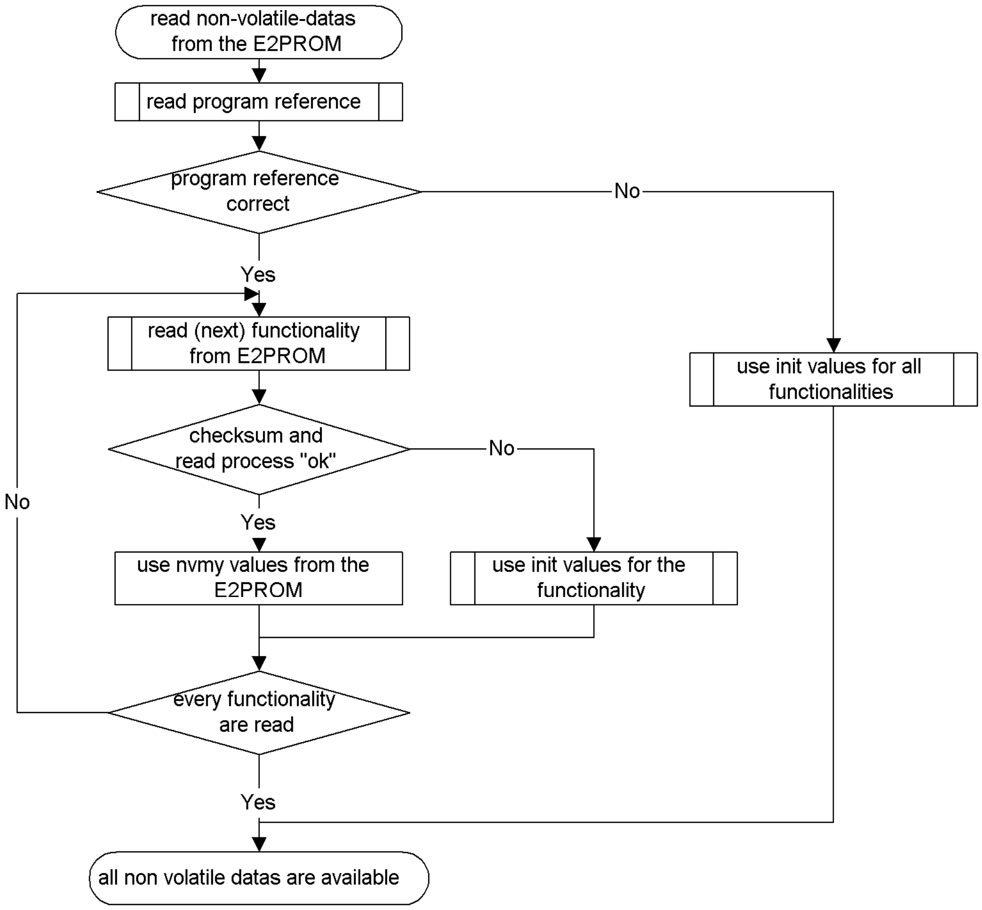


Figure 1 flow chart of the READ process

13.1.2 WRITE Non-Volatile-Datas to the E2PROM

Description:

The write process to the E2PROM starts at the end of the PWL (power latch phase).

Every individual functionality will be saved seperately with the current values for the non-volatile-datas in the E2PROM. Moreover a checksum will be calculated, which will also be stored. The actual program reference will be saved in the same way.

13.1.3 INITIALIZE Non-Volatile-Datas

Description:

Non-volatile-datas can be initialized by calibration system.

The following Non-Volatile-Datas can be initialized by a calibration constant:

- All Non-Volatile-Datas and all Adaptation-Values
- Selective adaptive range values

The initialization will be executed once per transition, if the calibration constant transits from 0 -> 1.

Remark: The following variables can not be initialized by a calibration constant:

TRT (Total runtime counter)

VS_MAX_SEL_EXT_REQ (Request of maximum vehicle speed limitation)

Formula section:

Initialize all Adaptation-Datas:

```
If      LC_AD_CLR = 1
Then    All Selective Adaptive Range Values and all Non-Volatile-Datas are initialized (without Error memory)
Endif
```

Initialize selectiv Non-Volatile-Datas:

```
If      LC_AD_CLR_XXXX = 1      (XXXX = selective adaptation range)
Then    Selective adaptive values are initialized
Endif
```

Clear KLANN-Adaptation values:

```
If      LC_AD_CLR_LACO 0 -> 1
Then    call klann_clrad(void)
Endif
```

13.1.4 Selective adaptation ranges

Description:

The following adaptation values/datas from the last engine run are considered in selective adaptation ranges:

Adaptation range:		Initialization:
		(0x = HEX)
ALTER		
	STATE_TBL_ALTER	All 0
	T_STATE_TBL_ALTER	0
ATI		
	TCO_REST	0xFF
	LV_STALL	0x0
	T_AST_REST	0xFFFF
LONG_LAM	all LONG_LAM nonvolatile variables are **never cleared**	
	CTR_KM_DELTA_LAM_ADJ_LAM_AD_CUS	0
	CTR_KM_LAM_ADJ_LAM_AD_CUS	0
	CTR_LAM_ADJ_LAM_AD_CUS	0
	FAC_LAM_CYL_SEL_LAM_AD_CUS[NC_CYL_NR]	0x0
TQLO		
	TQ_DIF_IS_AD	0x0
	TQ_DIF_IS_AD_CONV	0x0
	TQ_DIF_IS_AD_ACC	0x0
	TQ_DIF_IS_AD_OPM_1	0x0
	TQ_DIF_IS_AD_OPM_2	0x0
	TQ_DIF_IS_AD_ACC_1_OPM_1	0x0
	TQ_DIF_IS_AD_ACC_1_OPM_2	0x0
	TQ_DIF_IS_AD_ACC_CONV_OPM_1	0x0
	TQ_DIF_IS_AD_ACC_CONV_OPM_2	0x0
	TQ_DIF_IS_AD_CONV_NEUT	0x0
	TQ_DIF_IS_AD_CONV_1	0x0
	TQ_DIF_IS_AD_ACC_NEUT	0x0
	TQ_DIF_IS_AD_ACC_CONV	0x0
	TQ_DIF_IS_AD_ACC_1	0x0
	TQ_DIF_IS_AD_CONV_1_OPM_1	0x0
	TQ_DIF_IS_AD_CONV_1_OPM_2	0x0
	TQ_DIF_IS_AD_ACC_NEUT_OPM_1	0x0
	TQ_DIF_IS_AD_ACC_NEUT_OPM_2	0x0
	TQ_DIF_IS_AD_CONV_NEUT_OPM_1	0x0
	TQ_DIF_IS_AD_CONV_NEUT_OPM_2	0x0
TPS		
	V_TPS_AD_EL_BOL_1	C_V_TPS_AD_BOL_INI_1
	V_TPS_AD_EL_BOL_2	C_V_TPS_AD_BOL_INI_2
	V_TPS_AD_LIH_1	C_V_TPS_SP_LIH_1
	V_TPS_AD_LIH_2	C_V_TPS_SP_LIH_2
	TPS_AD_SLOP_GAIN_1	C_TPS_AD_SLOP_GAIN_INI_1
	LV_TPS_AD_REQ	0x1
	CTR_ST_CHK_CHK_LIH_1_READY	0x0
	CTR_ST_CHK_CHK_LIH_1_ERR	0x0
	CTR_TPS_JAM_DET_ACT	0x0
	V_TPS_AD_SLOP_GAIN_1	C_V_TPS_AD_SLOP_GAIN_INI_1
	TPS_LIH_1	TPS_LIH_INI
	TPS_LIH_2	TPS_LIH_INI
TOIL		
	TOIL_STOP	0x0

TOIL_MAX		
	TOIL_MAX_WARN	0x0
	VS_TOIL_MAX	0x0
	N_TOIL_MAX	0x0
	GEAR_TOIL_MAX	0x0
	TQI_AV_TOIL_MAX	0x0
	TAM_TOIL_MAX	0x0
	TCO_TOIL_MAX	0x0
	CTR_TOIL_MAX	0x0
	PV_AV_TOIL_MAX	0x0
	DIST_TOIL_MAX	0x0
	LV_CS_TOIL_MAX	0x0
	TOIL_THD_TOIL_MAX	0x0
SOF		
	CONF_SOF_SWI	0x0
	LV_VAR_SOF	0x0
CAN		
	T_REL_CAN_ES	0x0
	T_REL_CAN_ES_2	0x0
	CTR_KM_CAN	0x0
POW		
	FAC_POW_MNG_VST_CNS[15]	0
N_SP_IS		
	N_KWP_OFS_ACC_DRI_NVMY	0X0
	N_KWP_OFS_DRI_NVMY	0X0
	N_KWP_OFS_NVMY	0X0
	N_KWP_OFS_ACC_NVMY	0X0
	N_KWP_OFS_VB_NVMY	0X0
CRU		
	STATE_CRU_OFF_IRR	0x0
	STATE_CRU_OFF_REV	0x0
	VS_STEP[15]	0x0
FMY		
	CTR_ERR_DYN_NUM	
	CTR_ERR_HMEM	
	Freezeframe	
	Dynamic-error-memory-information [NC_NR_ERR_DYN] {SW: dtc, diag_inst, ctr_frc, ctr_wup_cyc, ctr_dc, idx_frf}	
	Static-error-memory-information [NC_NR_ERR_DYN] {SW: ctr_abc, err_sym_N, first_err_info, err_info_obd}	
	Remark: (NC_MASK_ERR_PRES_DC, NC_MASK_ERR_DC, NC_MASK_ERR_END_DIAG, NC_MASK_ERR_DET are not saved)	
	Readiness-code[NC_READINESS_SIZE]	

	History-memory-information [NC_NR_ERR_HIS] {SW: CDK, status_of_dtc, ctr_frc, ctr_km, fmy_class}	
TRT		
	TRT	0
N_MAX		
	TRT_N_MAX	0
	N_MAX	0
	CTR_N_MAX	0
	CTR_KM_N_MAX	0
	N_GRD_N_MAX	0
	VS_N_MAX	0
	PV_AV_N_MAX	0
	GEAR_EF_N_MAX	0
	LV_CS_N_MAX	0
	T_SUM_N_MAX	0
	T_N_MAX	0
	NR_PAR_SCC_N_MAX	0
OBD2		
	CTR_STOP_FSD	0
	EFF_CAT_DIAG[NC_CBK_EX_NR]	0
	CTR_TOT_CAT_DIAG[NC_CBK_EX_NR]	0
	EFF_CAT_DIAG_HOM	0
IMOB		
	confidential	
TECU		
	CTR_STC_TECU_1	0x0
	CTR_STC_TECU_2	0x0
	CTR_STC_TECU_3	0x0
	CTR_STC_TECU_4	0x0
	CTR_STC_TECU_5	0x0
	CTR_STC_TECU_6	0x0
	CTR_STC_TECU_7	0x0
	CTR_STC_TECU_8	0x0
VAR		
	LV_VAR_ASR	FALSE
	LV_VAR_ASR_3	FALSE
	LV_VAR_ASR_4	FALSE
	LV_VAR_AMT	FALSE
	LV_VAR_ACIN	FALSE
	LV_AT	FALSE
	LV_VAR_MSW	0x0
	LV_VAR_EFP_CRASH	FALSE
	LV_VAR_PSTE	0
	LV_VAR_PSTE_2	0
	LV_VAR_PSTE_3	0
	LV_VAR_ECRAS_UP	0
	LV_VAR_ECRAS_DOWN	0
	LV_ECRAS_DOWN_DET_SET	0
	LV_VAR_PBR	0
	LV_VAR_TCT	FALSE

	LV_VAR_DCC	0
	LV_VAR_ARS	0
	LV_VAR_BN_MSW	0
	LV_VAR_BN_EFP	0
	LV_VAR_BN_GEAR_REV	0
	LV_VAR_BN_LTG_HDLP_L	0
	LV_VAR_BN_LDM	0
	LV_VAR_BN_TRL	0
	LV_VAR_ETCU	0
	LV_VAR_ETCU_3	0
	LV_VAR_EBOX_CFA	0
	LV_VAR_EF	0
	LV_VAR_LSH_UP	0
	LV_VAR_LSH_DOWN	0
	LV_VAR_RLY_ACCOUT	0
	LV_VAR_RLY_ST	0
	LV_VAR_MAF	0
	FAC_GR_MMV_AV	0x8000
	LV_SENS_BAT_SMT_DET	0
	LV_VAR_AEB	0
	LV_VAR_4WD	0
	LV_VAR_ETCU_SPT	0
	LV_VAR_MAF_LEARNT	0
DMTL		
	CTR_CNL_ROUGH_LEAK_MES_FUC	0
	CTR_REP_ROUGH_LEAK_MES_FUC	0
	LV_ENA_FUC	0
	LV_CTR_CNL_SMALL_LEAK_MES	0
	CTR_CNL_SMALL_LEAK_MES	0
	LV_DMTL_STOP	0
	LV_FUC_OPEN	0
	LV_ERR_DET_DMTL_MAX	0
	LV_ERR_DET_DMTL_MIN	0
	LV_SDR_DMTL	0
	LV_ERR_SIG_DMTL	0
	LV_VB_DIF_MAX	0
	CTR_CUR_DMTL_REF_DIF_MAX	0
	LV_DET_FUC_OPEN	0
	LV_DET_REFU	0
	LV_T_DMTL_MAX	0
	LV_CUR_DMTL_THD_DIF_MES	0
	DIST_DMTL	0
	LV_ERR_DET_ROUGH_LEAK_MIN	0
	LV_DIST_DET_ROUGH_LEAK	0
	LV_ROUGH_LEAK_SUSP_SET	0
	LV_REFU_DMTL	0
	LV_CUR_DMTL_REF_DIF_MAX	0
	FCO_AV_DMTL	0
	LV_ENA_SMALL_LEAK_MES	0
	FTL_INI	0
	LV_CTR_DC_CHK_FUC	0
	LV_ENA_CHK_FUC_REFU	0
	CTR_DC_ENA_CHK_FUC	0
	CTR_CHK_FUC_OPEN	0
	LV_CHK_FUC_OPEN	0
	LV_CHK_FUC_OPEN_ACT	0
	LV_REP_CHK_FUC	0
	LV_ROUGH_LEAK_SUSP_CHK_FUC	0
	LV_ERR_FTL_PLAUS	0

	LV_CHK_DMTL_CUR_DMTL_VLD_VIRT	0
	LV_DET_NO_ROUGH_LEAK	0
	LV_DET_SMALL_LEAK	0
	LV_DET_NO_SMALL_LEAK	0
	CTR_TCO_ST_DMTL	0
	CUR_DMTL_REF_LEAK_EOL	0
	CUR_DMTL_ROUGH_LEAK_MIN_EOL	0
	LV_ERR_FTL	0
	M6_CUR_DMTL_ROUGH_LEAK_END	0
	M6_CUR_DMTL_THD_ROUGH_LEAK_LEN	0
	M6_CUR_DMTL_ROUGH_LEAK_LEN_END	0
	M6_CUR_DMTL_THD_ROUGH_LEAK	0
	M6_CUR_DMTL_COR_FIL_CID18	0
	M6_CUR_DMTL_COR_FIL_CID19	0
	M6_CUR_DMTL_DMTLS_TEST	0
	M6_CUR_DMTL_THD_DMTLS_TEST	0
	M6_CUR_DMTL_SMALL_LEAK_END	0
	M6_CUR_DMTL_REF_LEAK	0
	M6_CTR_CNL_SMALL_LEAK_MES	0
	CTR_CNL_SMALL_LEAK_MES_VIRT	0
	FTL_OLD	0
	LV_FTL_DIAG	0
	CUR_DMTL_COR_FIL_EOL	0
	STATE_DMTL_EOL	0
ENRD	all ENRD nonvolatile variables are **never cleared**	
	DELTA_CRK_DIF_MAX_ER	NC_CYL_NR * 4 * C_CRK_DIF_MAX_ER
	CTR_SEG_AD_ER	0
	LV_SEG_AD_AVL_ER	0
	SEG_AD_MMV_ER[NC_CYL_NR]	0
	LV_SEG_AD_LIM_ER	0
LIF		
	FAC_FCO_KWP	0
	LV_ALTER_CTL_EXT_ADJ	0
	STATE_CMB_CTL_NVMY_KWP	0
	CHK_STAMP	0
ENTE		
	TCO_STOP	0x0
	LV_ERR_TCO_PREL_DET	0
	LV_ERR_TCO_2_PREL	0
LACO	all LACO nonvolatile variables are **never cleared**	
	FAC_MFF_ADD_FAC_LAM_AD[NC_CBK_EX_NR]	0
	FAC_L_RNG_LAM_AD[NC_CBK_EX_NR] FAC_H_RNG_LAM_AD[NC_CBK_EX_NR] MFF_ADD_LAM_AD[NC_CBK_EX_NR] LAMB_DELTA_AD_LAM_ADJ[NC_CBK_EX_NR]	0
	FAC_LAM_TCO_A[NC_CBK_EX_NR] FAC_LAM_TCO_B[NC_CBK_EX_NR] FAC_LAM_TCO_C[NC_CBK_EX_NR] FAC_LAM_TCO_D[NC_CBK_EX_NR] FAC_LAM_TCO_E[NC_CBK_EX_NR] FAC_LAM_TCO_MIN[NC_CBK_EX_NR]	0
	TCO_A_LAM_AD_WUP[NC_CBK_EX_NR] TCO_B_LAM_AD_WUP[NC_CBK_EX_NR] TCO_C_LAM_AD_WUP[NC_CBK_EX_NR] TCO_D_LAM_AD_WUP[NC_CBK_EX_NR] TCO_E_LAM_AD_WUP[NC_CBK_EX_NR]	0

	TCO_MIN_LAM_AD_WUP[NC_CBK_EX_NR]	
TQI_LIM		
	SUM_TQI_REQ_LIM	0
	T_TQI_REQ_LIM_MAX	0
	TQI_REQ_LIM_DIF_MAX	0
EGCP		0
	R_IT_THD_OBD_LSH_DOWN[NC_CBK_EX_NR]	0
	R_IT_OBD_LSH_DOWN[NC_CBK_EX_NR]	0
	CTR_SAVE_SWT_LS_DOWN[NC_CBK_EX_NR]	0
	VLS_DIF_SAVE_DIAG_LS_DOWN[NC_CBK_EX_NR]	0
	FAC_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	0
	FAC_DYN_LSL_DIAG_SAE	0
	FAC_DYN_LSL_DIAG_TOL_SAE	0x7FFF
	IPLSL_MMV_VLD_FCUT[NC_CBK_EX_NR]	C_IPLSL_NOM_PURGE
	IPLSL_VARI_VLD_FCUT[NC_CBK_EX_NR]	0
	FAC_LSL_GAIN_AD[NC_CBK_EX_NR]	0x8000
	LV_LSL_FIRST_GAIN_AD	1
	LAMB_DELTA_I_SAVE_DIAG[NC_CBK_EX_NR]	0
	TTIP_MES_LS_UP_OBD_LSH_UP[NC_CBK_EX_NR]	0
	LAMB_THD_VPLSL_LIM[NC_CBK_EX_NR]	0
	VLS_DOWN_PUC_SAVE[NC_CBK_EX_NR]	0
	CRK_CYL_LAM_DELTA_INI [NC_CBK_EX_NR]	0
EVAC		
	CPPWM_ADD_AD_MEM	C_CPPWM_ADD_AD_MEM_NVMY_INI
MIS		
	CTR_TOT_MIS	0x0
	CTR_SEG_MIS_ACT	0x0
	CTR_MIS_DET	0x0
	CTR_MIS_DET_CYL[NC_CYL_NR]	0x0
	CTR_MIS_TOT_NVMY	0x0
	CTR_MIS_TOT_DC	0x0
	CTR_MIS_DC_CYL[NC_CYL_NR]	0x0
	CTR_MIS_DC_MMV_CYL[NC_CYL_NR]	0x0
PSTE		
	ANG_PSTE_END_RI_AD	C_ANG_PSTE_END_RI_INI
	ANG_PSTE_END_LE_AD	C_ANG_PSTE_END_LE_INI
ENSD		
	PSN_EDGE_AD_CAM_IN_i[NC_NR_EDGE_CAM_IN]	
	PSN_EDGE_AD_CAM_EX_i[NC_NR_EDGE_CAM_EX]	
EXTD		
	TEG_DYN_STOP	0x0
	CTR_COLD_ST	0
ECM2		
	CTR_SWI_AFS_MON	0
RON	CTR_IGA_DIF_MAX_KNK[NC_CYL_NR]	0
	N_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	0x0
	MAF_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	0x0
	TCO_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	0x40
	TIA_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	0x40
	CYL_ID_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	0x0
VIM		

	CTR_VIMPWM_1_EDGE	0 (never cleared, like TRT)
	CTR_VIMPWM_2_EDGE	0 (never cleared, like TRT)
MSD70 specific		
NOxCTR_COLD_ST_NT0	EGY_DEW_NT_STOP	0
CTR_NT_AGI_AD_CMPL_SUM 0		
	DIST_O2_STC_OLD	0
	LV_NT_AFS_REQ_AGI	0
	LV_NT_O2_STC_LIM	1
	LV_NT_O2_STC_REQ	0
	LV_NT_O2_STC_VLD	0
	LV_NT_AFS_REQ_AGI_TMP_3	0x0
	LV_SO2P_REQ_FQ	0x0
	CTR_NT_AGI_SO2P_FQ	0x0
	NT_AGI_SO2P_FQ_SUM	0x0
	NT_AGI_SUL_SNG[NC_NT_NR]	0xFFFFFFFF
	NT_AGI_THERMO_SNG[NC_NT_NR]	0xFFFF
	LV_SO2P_REQ_1	0
	LV_SO2P_REQ_2	0
	NOX_AD_FAC_SNG[NC_NT_NR]	0x80
	NOX_OFS_LOAD[NC_NOX_SENS_CONF]	0
	NOX_OFS_PUC[NC_NOX_SENS_CONF]	0
	NT_AGI	0xFFFF
	NT_AGI_SUL	0xFFFF
	NT_AGI_THERMO	0xFFFF
	NT_O2_STC_AD	0x80
	NT_O2_STC_OFS	0
	NT_SUL_32	0
	NT_SUL_H_32	0
	SUM_NTL_CYC	0
	T_AFL_SUM	0
	NTL	0
BAL	all BAL nonvolatile variables are **never cleared**	
	FAC_TI_AD_ER_BAL[NC_CYL_NR]	0x0
	MFF_ADD_AD_ER_BAL[NC_CYL_NR]	
	ER_STND_MMV_BAL[NC_CYL_NR]	
	ER_STND_MMV_STD_BAL[NC_CYL_NR]	
	ER_STD_MMV_BAL[NC_CYL_NR]	
	LV_CYL_BAL_LAM_AD_EOL	0x0
	LV_CYL_BAL_LAM_SEL_AD_COLD_EOL	0x0
	LV_CYL_BAL_LAM_SEL_AD_HOT_EOL	0x0
	LV_CYL_BAL_ER_AD_FAC_EOL	0x0
	LV_CYL_BAL_ER_AD_ADD_EOL	0x0
	LV_CYL_BAL_LAM_SEL_AD_COLD[NC_CBK_EX_NR]	0x0
	LV_CYL_BAL_LAM_AD_FAC[NC_CBK_EX_NR]	0x0
	LV_CYL_BAL_LAM_AD_ADD[NC_CBK_EX_NR]	0x0
	LV_CYL_BAL_LAM_SEL_AD_RNG_L[NC_CBK_EX_NR]	0x0
	LV_CYL_BAL_LAM_SEL_AD_RNG_H[NC_CBK_EX_NR]	0x0
	T_CYL_BAL_LAM_SEL_DC	0x0
	T_CYL_BAL_LAM_AD_DC	0x0
	LV_CYL_BAL_LAM_AD_DC	0x0
	LV_CYL_BAL_LAM_SEL_AD_COLD_DC	0x0
	LV_CYL_BAL_LAM_SEL_AD_HOT_DC	0x0
	LV_CYL_BAL_LAM_SEL_RNG_COLD_DC[NC_CBK_EX_NR]	0x0
	LV_CYL_BAL_LAM_SEL_AD_RNG_L_DC[NC_CBK_EX_NR]	0x0

	LV_CYL_BAL_LAM_SEL_AD_RNG_H_DC[NC_CBK_EX_NR]	0x0
	LV_CYL_BAL_LAM_AD_ADD_DC[NC_CBK_EX_NR]	0x0
	LV_CYL_BAL_LAM_AD_FAC_L_DC[NC_CBK_EX_NR]	0x0
	LV_CYL_BAL_LAM_AD_FAC_H_DC[NC_CBK_EX_NR]	0x0
EGRC_FAC_ACR_SLOP		
	FAC_ACR_SLOP	C_V_ACR_AD_BOL_INI
	V_ACR_AD_BOL	C_V_ACR_AD_TOL_INI
	V_ACR_AD_TOL	1
	LV_ACR_AD_REQ	C_V_ACR_AD_BOL_INI
	V_ACR_AD_BOL_0	
INSY		0x0
	V_MAP_AD	
INJR		
	FAC_EGY_PWM_AD[NC_CYL_NR]	C_FAC_EGY_PWM_MAN
CILC	all CILC nonvolatile variables are **never cleared**	
	FAC_LAM_CYL_SEL_ADJ_L_RNG[NC_CYL_NR]	0x0
	FAC_LAM_CYL_SEL_ADJ_H_RNG[NC_CYL_NR]	0x0
	FAC_LAM_CYL_SEL_ADJ_CST[NC_CYL_NR]	0x0
	MFF_ADD_CYL_LAM_COR[NC_CYL_NR]	0x0
	STATE_LAMB_CYL_SEL_CQ_SLOP[NC_CBK_EX_NR]	0x0
	LV_LAM_CYL_SEL_ADJ_H_RNG_VLD[NC_CBK_EX_NR]	0x0
	LV_LAM_CYL_SEL_ADJ_L_RNG_VLD[NC_CBK_EX_NR]	0x0
	DELTA_CRK_CYL_LAM[NC_CBK_EX_NR]	0x0
	CTR_PHA_SHIFT_AD_TRIG[NC_CBK_EX_NR]	0x0
IGRE		
	CTR_ERR_OBD_DIAG_CYL_BAL_LAM[NC_CBK_EX_NR]	0x0
	CTR_ERR_OBD_DIAG_CYL_BAL_ER	0x0
EVAM	CL_MMV_SAE	0xFFFFH; Cleared only during re-programming
	MAF_DIAGCPS_SAE	0x0H; Cleared only during re-programming
	MAF_DIAGCPS_THD_SAE	0xFFFFH; Cleared only during re-programming
	SUM_DIAG_DIAGCPS_SAE	0x0H; Cleared only during re-programming
Customer_fr	all Customer_fr nonvolatile variables are **never cleared** and must be **master copy**	
	Tvneutral	0x00
Customer		Standardcase: 0x00
	Absch_korr	
	B_lvs_neustart	
	Batt_class	
	Bosbtvfbk	
	Bosconf	
	Bosfxid2	
	Bosfxid2r	0xFFFF
	Bosid2	0xFF
	Bosid2r	0xFF
	Bosjziel	0xFF
	Bosjzielr	0x0F
	Bosmziel	0x0F
	Bosmzielr	0xFF
	Bosprog2	0xFF
	Bosprog2r	
	Bosres	0xFF
	Bosrlsm	
	Bosrw2	0xFF

	Bosrw2r	0xFF
	Bszsi	
	D_soc	
	D_soc	
	Dfds[16]	
	F_atlad	
	Ibsderrs1	
	Ibsderrs2	
	Ibspreco2	
	Igenk	
	Igrinfo[30]	
	Ikurz_ogr	
	Ikurz_ugr	
	Iwakeupgr	
	Km_st_1	
	Ktupcsctr	
	Leminfo[40]	
	Msa_arravrs[60]	
	Msa_indexrs	
	Msainfo[50]	
	Msastz	
	Msastzmsa	
	Msnlgofs_tmp	
	Oz_kvbog	
	Oz_kvbsm_ul	
	Oz_lf1c	
	Oz_lf1t	
	Oz_lf2c	
	Oz_lf2t	
	Oz_nivlangt	
	Oz_oelkm	
	Oz_permbog	
	Oz_permex	
	Oz_permlow	
	Pminfo1[37]	
	Pminfo2[29]	
	Pmsv[15]	
	Qv_cdherst_1	
	Qv_cdherst_2	
	Qv_cdherst_3	
	Qv_cdherst_4	
	Qv_cdherst_5	
	Qv_cdherst_6	
	Qv_cdherst_7	
	Qv_cdherst_8	
	Qv_h2o	
	Qv_h2oquali	
	Qv_h2ostatus	
	Qv_nv_ezm	
	Qv_nv_start	
	Qv_nv_zh	
	Qv_out_1	
	Qv_out_2	
	Qv_out_3	
	Qv_out_4	
	Qv_out_5	

	Qv_out_m	
	Qv_quali_1	
	Qv_quali_2	
	Qv_quali_3	
	Qv_quali_4	
	Qv_quali_5	
	Qv_quali_m	
	Qv_status	
	Qv_td1	
	Qv_td2	
	Qv_td3	
	Qv_td4	
	Qv_td5	
	Qvc_status_1	
	Qvc_status_2	
	Qvc_status_3	
	Qvc_status_4	
	St_bns	
	St_deavns2	
	St_devxdetec	
	St_dgenerrst_md1	
	St_dgenerrst_md2	
	St_dgengrenz1	
	St_dgengrenz1_md1	
	St_dgengrenz1_md2	
	St_dgengrenz2	
	St_dgengrenz2_md1	
	St_dgengrenz2_md2	
	St_dgengrenzerr	
	St_dgengrenzerr_md1	
	St_dgengrenzerr_md2	
	St_dgengrenznz	
	St_dgengrenznz_md1	
	St_dgengrenznz_md2	
	St_dgenub1	
	St_dgenub1_md1	
	St_dgenub1_md2	
	St_dgenub2	
	St_dgenub2_md1	
	St_dgenub2_md2	
	St_dgenuberr	
	St_dgenuberr_md1	
	St_dgenuberr_md2	
	St_dgenubnz	
	St_dgenubnz_md1	
	St_dgenubnz_md2	
	St_dgenugen1	
	St_dgenugen1_md1	
	St_dgenugen1_md2	
	St_dgenugen2	
	St_dgenugen2_md1	
	St_dgenugen2_md2	
	St_dgenugenerr	
	St_dgenugenerr_md1	
	St_dgenugenerr_md2	
	St_dgenugennz	

	St_dgenugennz_md1	
	St_dgenugennz_md2	
	St_dibs0	
	St_ecostat2	
	St_eisyad_read	
	St_elu	
	St_igrinnv	
	St_igroutnv	
	St_lvs_in_nv	
	St_ngang	
	St_pmi_nv	
	St_spa2	
	St_ubw	
	Tmot_abstell	
	Tn_abstellm	
	Tumg_abst	
	Tumg_battab	
	Vsa_adp_ext	
	Vse_adp_ext	
	Zbibs	
	Zr_lvs_0	
	Zr_lvs_1	
	Zr_lvs_2	
	Zr_lvs_3	
	Zr_lvs_ll_reakt	
	Zr_lvssekt_0	
	Zr_lvssekt_1	
	Zr_lvssekt_2	
	Zr_lvssekt_3	
	Zr_lvssekt_4	
	Zr_lvssekt_5	
	Zr_lvssekt_6	
	Zr_lvssekt_7	
	Zr_lvssekt_8	
	Zrbosmld	

Hint:
To supress start problems after reprogramming it´s necessary to save ´never cleared´ variables for ´lifetime´ of ECU.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_AD_CLR	1	0...1H	0...1	1	[-]
Logical constant for initializing all selective adaption range values					
LC_AD_CLR_ALTER	1	0...1H	0...1	1	[-]
Logical constant for initializing the alternator profile nvmy data					
LC_AD_CLR_ATI	1	0...1H	0...1	1	[-]
Logical constant for initializing ATI nvmy data					
LC_AD_CLR_BAL	1	0...1H	0...1	1	[-]
Logical constant for initializing the balancing nvmy data					
LC_AD_CLR_CAN	1	0...1H	0...1	1	[-]
Logical constant for initializing CAN nvmy data					
LC_AD_CLR_CILC	1	0...1H	0...1	1	[-]
Logical constant for initializing the cyl. ind. lambda controller (CILC) nvmy data					
LC_AD_CLR_CRU	1	0...1H	0...1	1	[-]
Logical constant for initializing CRU nvmy data					
LC_AD_CLR_DMTL	1	0...1H	0...1	1	[-]
Logical constant for initializing DMTL nvmy data					
LC_AD_CLR_ECM2	1	0...1H	0...1	1	[-]
Logical constant for initializing the ECM2 nvmy data					
LC_AD_CLR_EGCP	1	0...1H	0...1	1	[-]
Logical constant for initializing EGCP nvmy data					
LC_AD_CLR_EGR	1	0...1H	0...1	1	[-]
Logical constant for initializing EGR nvmy data					
LC_AD_CLR_ENRD	1	0...1H	0...1	1	[-]
Logical constant for initializing the ENRD nvmy data					
LC_AD_CLR_ENSD	1	0...1H	0...1	1	[-]
Logical constant for initializing ENSD nvmy data					
LC_AD_CLR_ENTE	1	0...1H	0...1	1	[-]
Logical constant for initializing the ENTE nvmy data					
LC_AD_CLR_EVAC	1	0...1H	0...1	1	[-]
Logical constant for initializing EVAC nvmy data					
LC_AD_CLR_EXTD	1	0...1H	0...1	1	[-]
Logical constant for initializing the EXTD nvmy data					
LC_AD_CLR_INJR	1	0...1H	0...1	1	[-]
Logical constant for initializing INJR nvmy data					
LC_AD_CLR_INSY	1	0...1H	0...1	1	[-]
Logical constant for initializing the INSY nvmy data					
LC_AD_CLR_IVVT	1	0...1H	0...1	1	[-]
Logical constant for initializing IVVT nvmy data					
LC_AD_CLR_LACO	1	0...1H	0...1	1	[-]
Logical constant for initializing the LACO nvmy data					
LC_AD_CLR_LAM	1	0...1H	0...1	1	[-]
Logical constant for initializing LAMBDA nvmy data					
LC_AD_CLR_LIF	1	0...1H	0...1	1	[-]
Logical constant for initializing LIF nvmy data					
LC_AD_CLR_LONG_LAM_1	1	0...1H	0...1	1	[-]
Logical constant for initializing Long Term Lambda Adaptation (Bk1) nvmy data					
LC_AD_CLR_LONG_LAM_2	1	0...1H	0...1	1	[-]
Logical constant for initializing Long Term Lambda Adaptation (Bk2) nvmy data					
LC_AD_CLR_MIS	1	0...1H	0...1	1	[-]
Logical constant for initializing MISFIRE nvmy data					
LC_AD_CLR_N_MAX	1	0...1H	0...1	1	[-]
Logical constant for initializing the N_MAX nvmy data					
LC_AD_CLR_N_SP_IS	1	0...1H	0...1	1	[-]
Logical constant for initializing N_SP_IS nvmy data					
LC_AD_CLR_NOX	1	0...1H	0...1	1	[-]
Logical constant for initializing the NOx nvmy data					

LC_AD_CLR_OBD2	1	0...1H	0...1	1	[-]
Logical constant for initializing OBDII nvmy data					
LC_AD_CLR_POW	1	0...1H	0...1	1	[-]
Logical constant for initializing the POW nvmy data					
LC_AD_CLR_PSTE	1	0...1H	0...1	1	[-]
Logical constant for initializing PSTE nvmy data					
LC_AD_CLR_RON	1	0...1H	0...1	1	[-]
Logical constant for initializing RON nvmy data					
LC_AD_CLR_SOF	1	0...1H	0...1	1	[-]
Logical constant for initializing the sound flap / sport switch					
LC_AD_CLR_TECU	1	0...1H	0...1	1	[-]
Logical constant for initializing TECU nvmy data					
LC_AD_CLR_TOIL	1	0...1H	0...1	1	[-]
Logical constant for initializing TOIL nvmy data					
LC_AD_CLR_TOIL_MAX	1	0...1H	0...1	1	[-]
Logical constant for initializing TOIL_MAX nvmy data					
LC_AD_CLR_TPS	1	0...1H	0...1	1	[-]
Logical constant for initializing TPS nvmy data					
LC_AD_CLR_TQI_LIM	1	0...1H	0...1	1	[-]
Logical constant for initializing TQI_LIM nvmy data					
LC_AD_CLR_TQLO	1	0...1H	0...1	1	[-]
Logical constant for initializing IDLE nvmy data					
LC_AD_CLR_VAR	1	0...1H	0...1	1	[-]
Logical constant for initializing VAR nvmy data					

13.2 Non Volatile Data Management – General Functionality

13.2.1 Introduction

This document specifies the high level SW specification of the NVMY function. It allows storing and restoring some data manipulated by the Engine Control Unit as adaptative values, error flags for the diagnosis, and engine environments on several driving cycles in case of failure.

The non volatile data management shall provide:

- a common interface between the aggregates in the ECU and the function NVMY i.e. the mechanism to access to the memory for writing and reading stored data,
- different services provided to the aggregates,
- a safety data storage

13.2.2 Abbreviations and terms

HW	HardWare
SW	SoftWare
ECU	Engine Control Unit
CRC	Cyclic Redundancy Checksum
EEPROM	Electrically Erasable PROM
NVMY	Non Volatile MemorY

FLASH EEPROM emulation means that NVMY data are stored into the flash device.

In the rest of this document the term memory designs either EEPROM or FLASH EEPROM emulation.

13.2.3 General description

Usually the manager provides the capability to initialise, memorize and restore data. The NVMY manager shall ensure the NVMY data recording, and also secure the NVMY data integrity.

The NVMY manager shall support:

- a scheduler to handle the priority of the tasks,
- a supervisor, called regularly by the operating system, handles the real time,
- an executor of the services,
- the return of error cases to the others aggregates,
- the initialisation of the data and the integrity of the data in the memory structure.

13.2.4 Hardware constraints

The NVMY manager is hardware independent and shall administrate the NVMY data on EEPROM and/or FLASH EEPROM emulation.

EEPROM:

Writing of NVMY data could be done “in live” (write on "fly") while the ECU is running. This operation must be limited in order to save the EEPROM life duration regarding the hardware write/erase cycles.

Also, the application needs to take in account the EEPROM size capability.

FLASH EEPROM emulation:

Both accesses to flash data are available regarding flash hardware definition:

- One bank: the operating system is blocking while erase/write operation. The write/erase operation has to ensure that the SW is in a safe ECU mode (shutdown phase). Application has to guaranty that all NVMY data must be store at shutdown phase.
- Dual banks: the operating system is not blocking while erase/write operation. There is the possibility to store NVMY data while applicative SW is running.

Two specific flash segments are requested (at least).

Recommendation:

The writing of NVMY data on "fly" must be limited to one per ECU cycle in order to save the memory life duration.

Pre-emptive operation:

Application must ensure to not perform operation on memory while it is used by the NVMY manager.

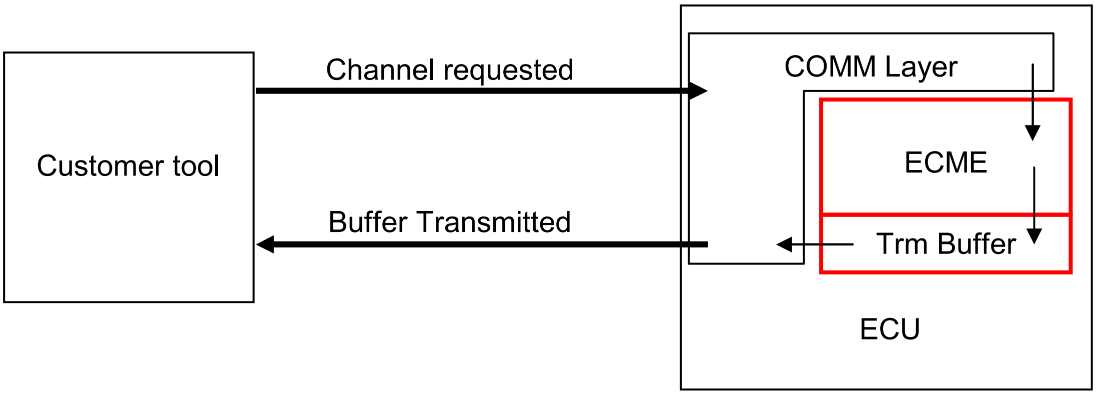
13.2.5 NVMY system events

Three related system events are defined:

- NVMRES: read-out (restore) non volatile data from the memory. It happens before the reset event. In case of error, NVMINI happens afterwards
- NVMINI: initialize the non-volatile data with defined default values. It happens after NVMRES and before the reset event in case of error during NVMRES
- NVMSTO: store the non-volatile data into the memory. It happens in a safe ECU mode (ECU shutdown). By extension, the write on "fly" functionality uses the same system event.

13.2.6 Flight recorder on customer tool request

On special communication request, a configured channel can be sent to a customer tool.



Via the communication media, the tool first asks for one specific channel.

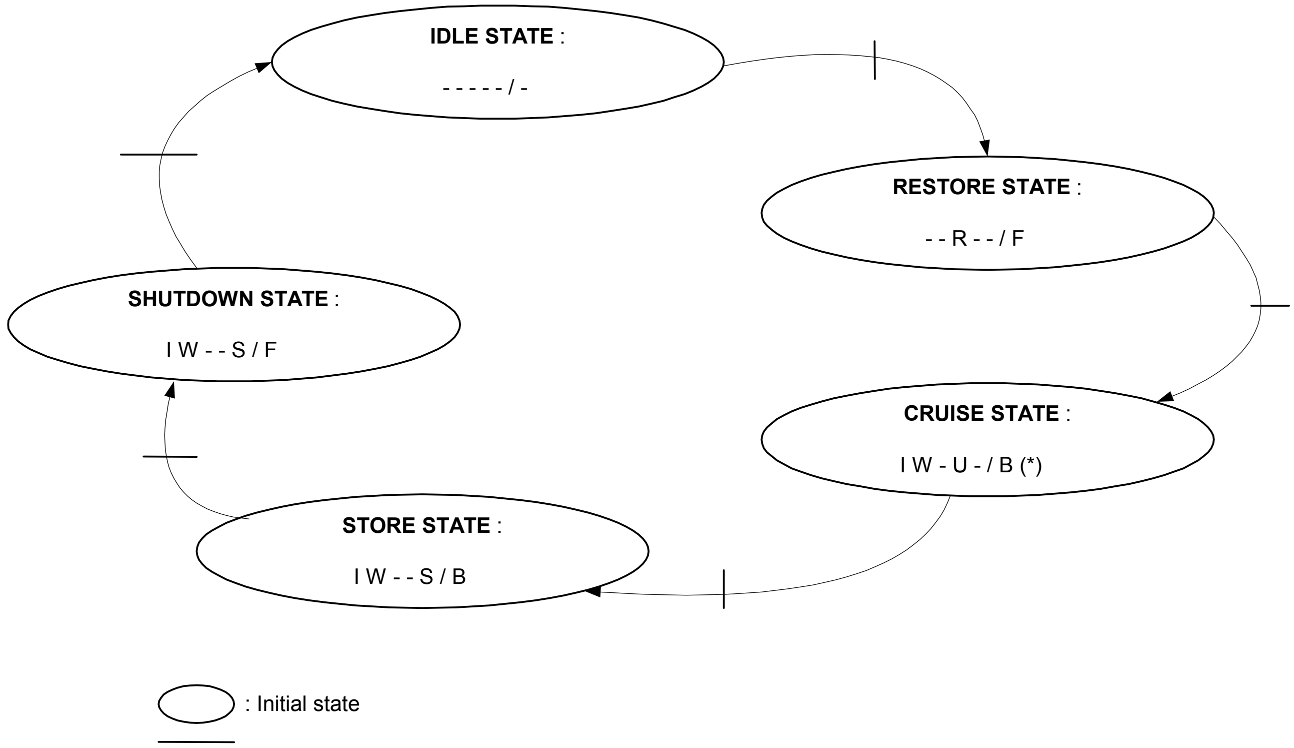
On reception of this message, ECME scheduler will prepare itself to read out the asked channel. As soon as the channel is available, a transmission buffer to the communication media is filled. After this, the channel can be sent to the customer tool.

13.3 Non Volatile Memory – Manager description

Basically, this module contains the scheduler and the supervisor related to NVMY functionality.
The supervisor is a state machine that contains the way that NVMY channel's operation are performed in addition to the type of operations enabled depending on the current state.
Furthermore, the supervisor only decides when NVMY operations should be performed.
Indeed, depending on the state of engine control management, the NVMY operations are performed either in BACKGROUND or FOREGROUND mode.
Foreground mode means that ECME's aggregate keeps "hand" till operation is finished.
On the contrary, BACKGROUND mode intends that NVMY operations are performed step by step while engine maybe running.

13.3.1 NVMY Manager

The picture below, presents all manager states :



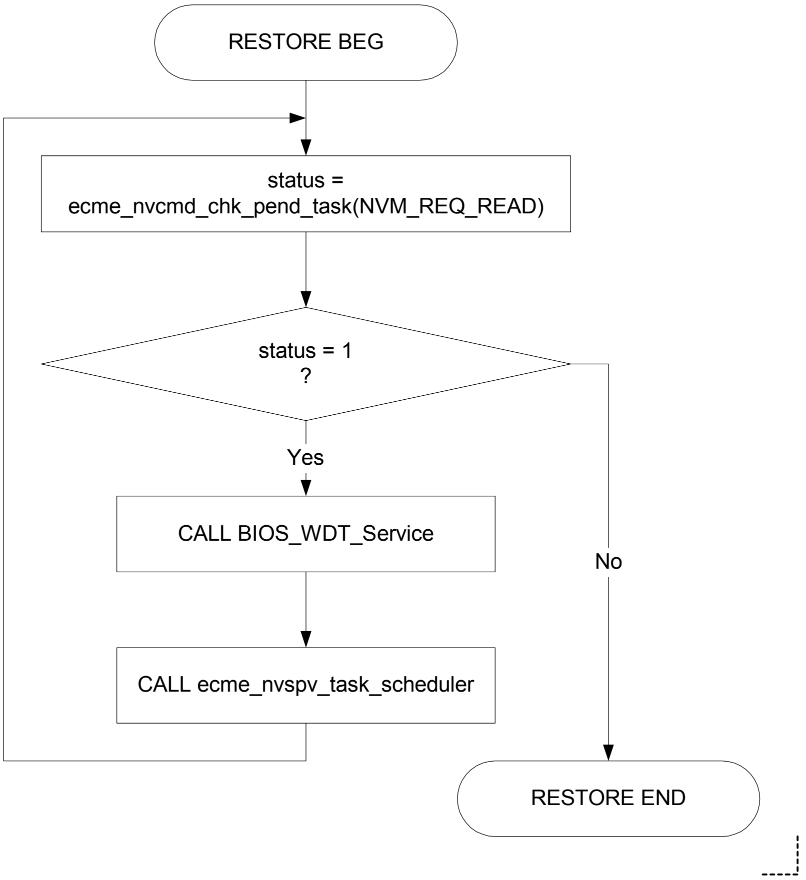
Active tasks :
- : disabled
I : Init task enabled
W : Write task enabled
R : Read task enabled
U : Update task enabled
S : Synchronisation task enabled

Operation mode :
F : Foreground
B : Background

13.3.2 STATES description

- IDLE state
  The supervisor simply does nothing.

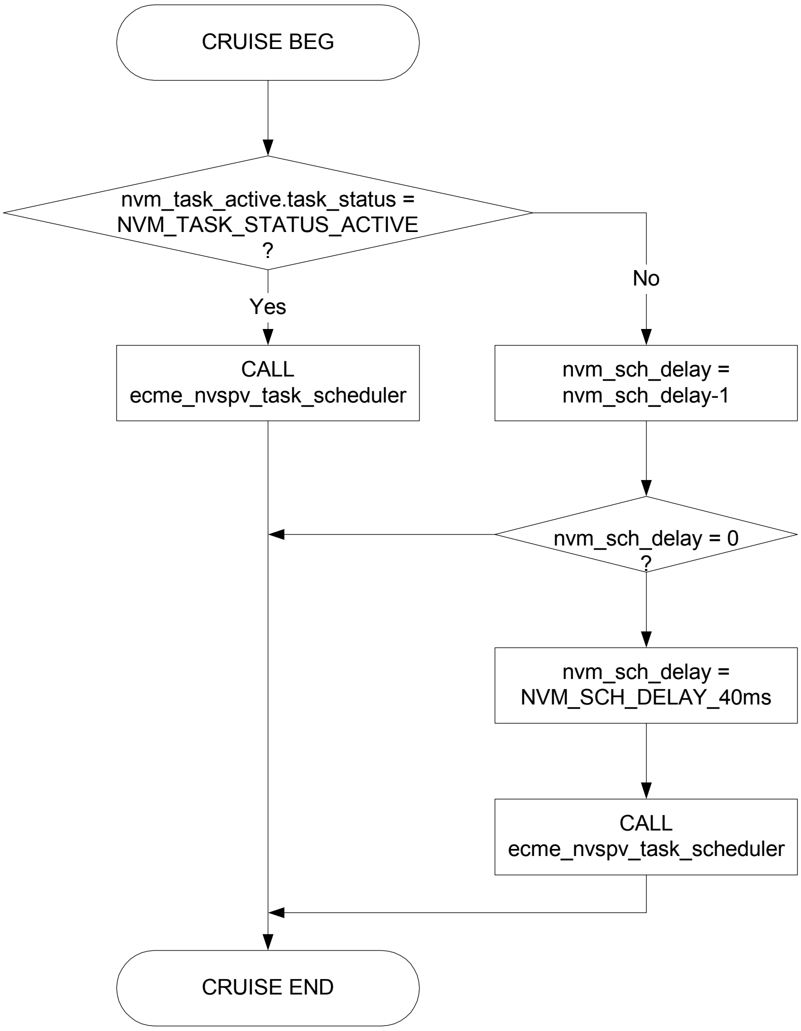
- RESTORE state :



This state is working in FOREGROUND. the NVMY scheduler is called till READ requests do not remain. As a complete RESTORE may take a long time, the Watchdog shall be rearmed periodically.

- CRUISE state :

This state is working in BACKGOUND. Each NVMY task should be performed step by step. The time base of 5ms for calling the supervisor is the good compromise according to the FLASH and EEPROM hardware data sheets.
Nevertheless, a new NVMY task shall be started 40ms after the end of the previous one, in order to minimize the load on SPI bus since engine maybe running.
Concerning watchdog, it is rearmed within OS operation.



- STORE state :

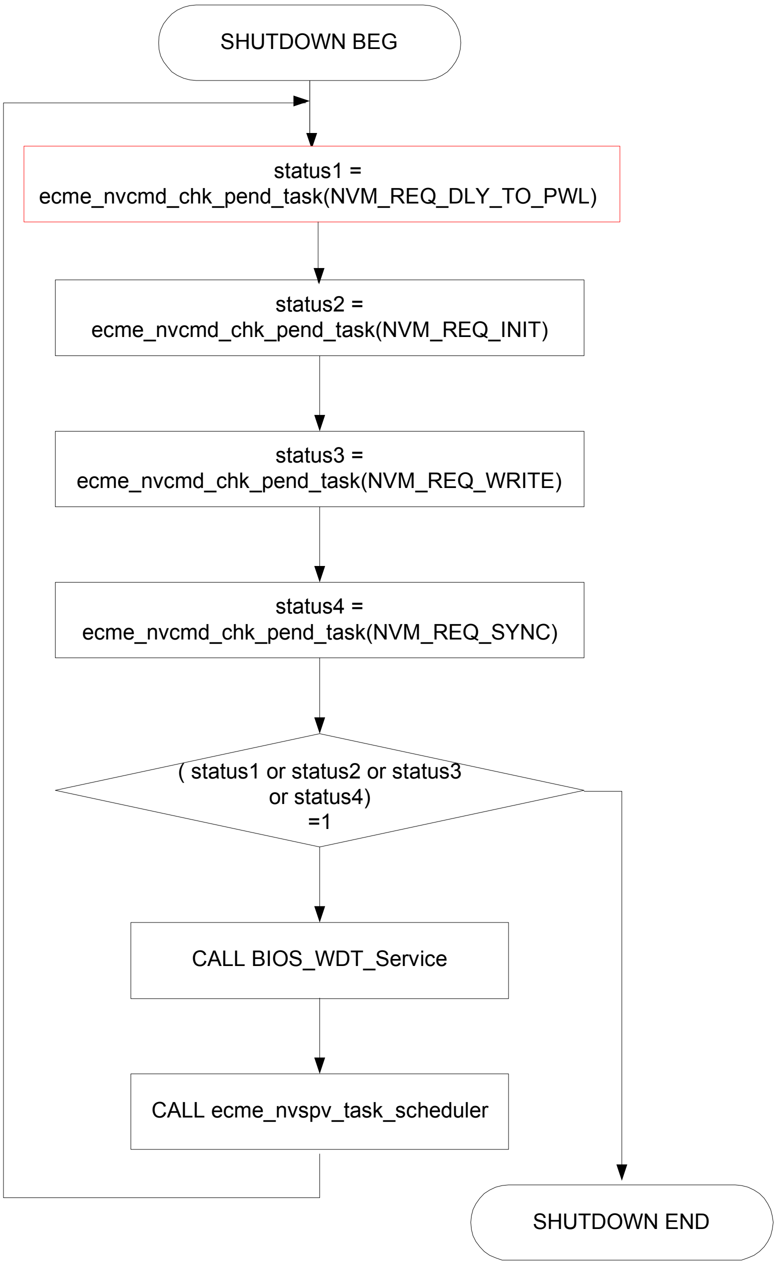
The NVMY scheduler (ecme_nvspv_task_scheduler) is simply called within this state.

REMARK :

Though this state is no more than a call, it may be found useless. Nevertheless, it might be used in future release of NVMY, in the case that STORE operations are started at the beginning of the power-latch.

- SHUTDOWN STATE

This state is working in FOREGROUND. This state is mainly used to complete STORE operations. The scheduler is either called till WRITE requests or INIT requests or SYNCHRONISATION requests are not pending.
As RESTORE operations, SHUTDOWN may take a long time. Therefore, Watchdog shall be periodically rearmed.



13.4 Non Volatile Data Management – Services Definition

This module especially contains all services offered to external aggregate.
These services are used to request operation from outside ECME aggregate

13.4.1 RESTORE service (FOREGROUND)

This service is in charge to restore a list of channels defined by parameter.
In addition, it may proceed with different kind of initialisation:

- First power on initialization
- Tuning initialization

This service is only available during RESTORE state according to NVMY supervisor.
It implies that it acts in Foreground only during ECU start-up. That is to say, OS is not active.

Therefore, it must be used during ECU start-up only.

13.4.1.1 Implementation

ECME_NVCMD				
ECME_NVMCMD_RESTORE_LST				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	list_name	U8 *	-	list name
OUTPUT	-	void	-	-

13.4.2 CLEAR INIT request (FOREGROUND)

This service is in charge to cancel an INIT request if the concerned channel is detected as corrupted at RESTORE stage.

It only acts in foreground during ECU’s start-up.

13.4.2.1 Implementation

ECME_NVCMD				
ECME_NVMCMD_CLEAR_INIT_REQ				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	nvmsetname	-	-	channel name
OUTPUT	-	void	-	-

13.4.3 READ service (FOREGROUND)

This service is in charge to read a single channel defined by parameter.
It only acts in foreground during ECU’s start-up.

13.4.3.1 Implementation

ECME_NVCMD				
ECME_NVMCMD_READ_CH				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	nvmsetname	-	-	channel name
OUTPUT	-	void	-	-

13.4.4 READ FOR DIAG service (BACKGROUND)

This service is in charge to read a single channel when a diagnosis tool requires it.
It only acts in background during ECU's cruise operation.
It is only supported by specific channel configuration.

13.4.4.1 Implementation

ECME_NVCMD				
ECME_NVMCMD_READ_FOR_DIAG_CH				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	nvmsetname	-	-	channel name
OUTPUT	-	void	-	-

13.4.5 WRITE service (BACKGROUND)

This service is in charge to write a single channel defined by parameter.
It only acts in foreground during ECU's cruise operation.

However, it may not work with flash that does not support basic operation while CPU fetches code (example: flash with simple bank). In this case the write operation is delay to the STORE ECME operation (done in the power latch).

Moreover, NVMY channel writing execution is delayed to running state, if writing is ordered during ECU start-up.

13.4.5.1 Implementation

ECME_NVCMD				
ECME_NVMCMD_WRITE_CH				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	nvmsetname	-	-	channel name
OUTPUT	-	void	-	-

13.4.6 STORE service (BACKGROUND)

This service is in charge to prepare the store of all channels.
This service only acts in background during ECU's store operation.

13.4.6.1 Implementation

ECME_NVCMD				
ecme_nvcmd_write_ch_req				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	-	-	-	-
OUTPUT	-	-	-	-

13.4.7 SHUTDOWN service (FOREGROUND)

This service is in charge to store all channels.
This service only acts in background during ECU's shutdown operation.

13.4.7.1 Implementation

ECME_NVCMD				
ecme_nvcmd_shutdown				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	-	-	-	-
OUTPUT	-	-	-	-

13.4.8 NVMY coherency error services (FOREGROUND)

ECME aggregate provides two services for the NVMY coherency check functionality in order to set or clear coherency error.

These services are based on two macros defined below. Basically, these macros respectively set nvm_sw_coherency_err to 1 or 0.

13.4.8.1 Implementation

ECME_NVCMD				
ECME_NVMCMD_VLD_CHK_OK				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	-	-	-	-
OUTPUT	-	-	-	-

ECME_NVCMD				
ECME_NVMCMD_VLD_CHK_NOK				
Parameters	**Name**	**Type**	**Value(s)**	**Comment**
INPUT	-	-	-	-
OUTPUT	-	-	-	-

13.5 Non Volatile Memory – Mechanisms for Data initialization

These section deals with all cases of NVMY channel initialization.

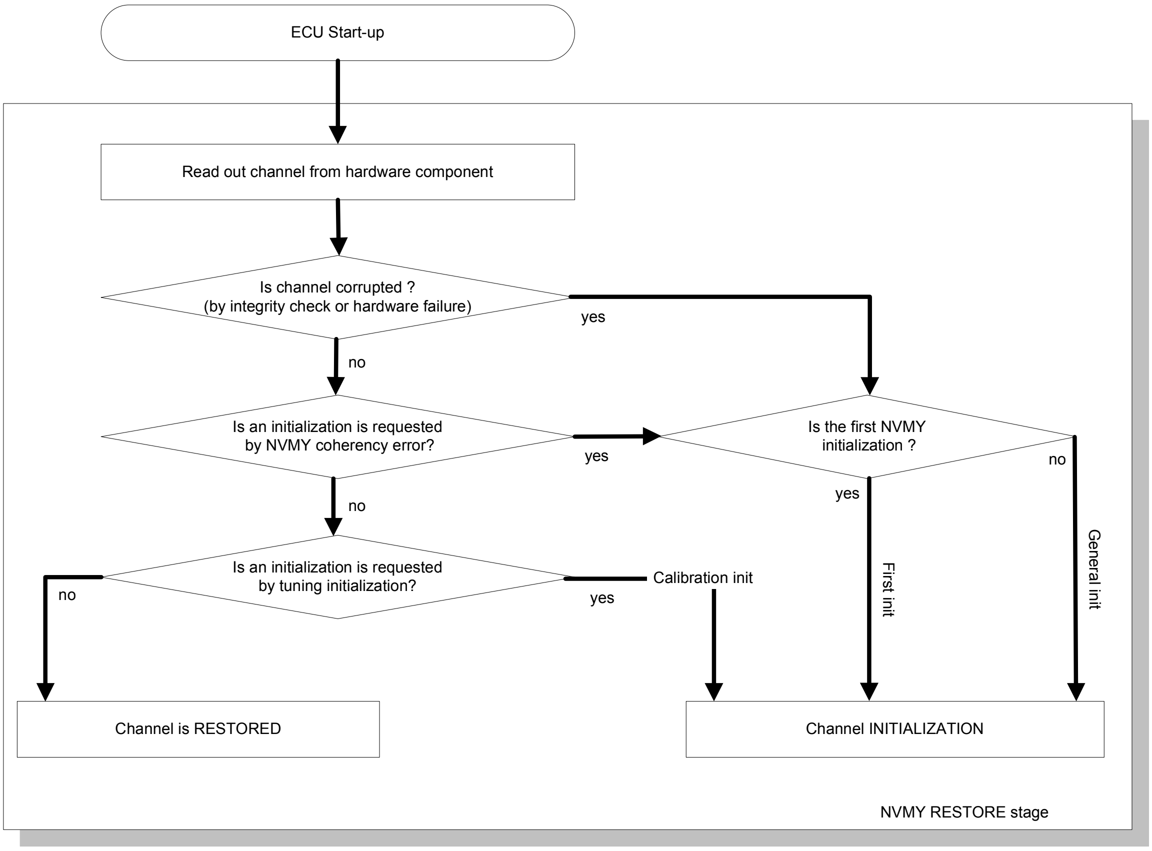
The initialization consists in initializing channel(s) with their default values.

The Channels set may be either partially or completely initialized depending on the case.

In addition, we can distinguish 3 kind of initialization:
- First initialization
- General initialization
- Calibration initialization

NVMY data initialization is performed during the NVMY RESTORE stage.

The following diagram depicts all kind of NVMY channel initialization:



13.5.1 Initialization in case of corrupted channel

Basically, this initialization is requested by the NVMY executor in case of a reading error occurred toward a single channel. Reading failure can be generated by a hardware failure or integrity channel problem (according to the channel attribute configuration).

13.5.2 Initialization in case of NVMY coherency error

During RESTORE stage, NVMY functionality should ensure that restore data match with Applicative software functionalities, which are using NVMY data.

NVMY coherency mainly relies on a check at RESTORE stage in order to determine whether NVMY contents (all channels) is compatible or not with applicative software.
Usually, problem may occur if ECU is restarted after it has been previously reprogrammed, because channel definition may have change in between from one software version to another. Without coherency check, it may lead Applicative software to run with NVMY data that are not correct, so coherency check is mandatory before restoring whole NVMY contents.

<u>Strategy:</u>
The NVMY coherency check relies on two channels (named SWVER_FIX and SWVER), which contains data related to coherency check (ECU-SW coherence identifier).
In the same way, Applicative software contains the same data. At restore stage, both channels are read before applicative channel reading. If NVMY content doesn't fit with current applicative software, a status notification is sent to the NVMY manager.

The below picture defines the channel allocation.

SWVER_FIX is the first written channel.
SWVER is the last written channel.

13.5.3 Initialization in case of tuning request

NVMY function provides the possibility to force channel initialization during <u>RESTORE stage</u>. This is <u>only</u> restricted to the ECU development phase.

This kind of initialization is limited and can only be applied to a limited number of channels, pre-defined with configuration. Therefore channels are grouped according to 3 possible levels of initialization.

Tuning calibration:

The following calibrations are defined under DDS and allow channel initialization:

Calibration name	Value	Purpose
lc_req_ini_nvm	PASSIVE	No request for initialization
	ACTIVE	Initialization is required
c_req_ini_nvm_grp	0	unused
	1	channels that belong to group level 1 should be initialized
	2	channels that belong to group level 2 should be initialized
	3	channels that belong to group level 3 should be initialized
	4	reserved for future purpose
	5	reserved for future purpose
	6	reserved for future purpose
	7	reserved for future purpose

13.5.4 Initialization in case of "First power-on"

In case of brand new ECU, the initialization of all channels with their default values is called initialization "first power-on".

Usually, this kind of initialization may occur only one time during ECU's lifetime.

This procedure relies on one channel (named ECME_1_FIX), which contains data related to the first power-on detection. Since this data is not set, all channels are supposed to be initialized. It implies that this data is false when ECU is delivered at the customer.

13.5.5 Hard initialization

Hardware means that channel is initialized with default values within hardware component.
Since these initialization may be performed during engine running (ECME CRUISE stage), it saves up writing time at the end of the power-latch.
Though this task is applied to EEPROM and FLASH, this initialization is only relevant for EEPROM.

13.6 Non Volatile Data Management, configuration

This section gives a summary of ECME features in order to advise you in the way that you define your data into channels organisation.

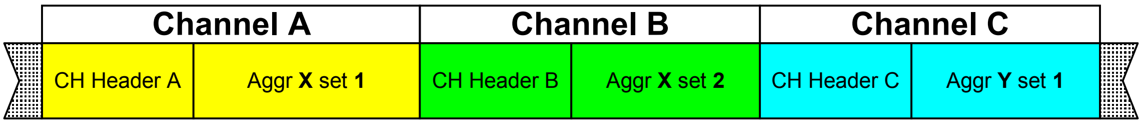
Each aggregate that want to use some Non Volatile data must create a channel containing all data to be saved.

13.6.1 Channel organisation

The NVMY is divided into channels that physically represent a part of the hardware component.

Each channel is divided into two parts. The first part so-called Header contains the data related to the saving criteria. In other words, the data that ensures saving data integrity. The second part only contains the saved data that belongs to the same aggregate.

Basically, the contents of non-volatile memory looks like as follow:



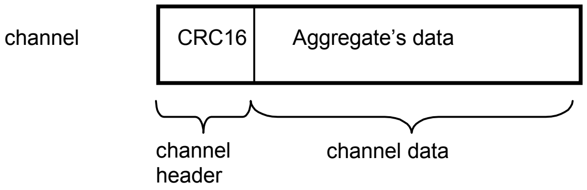
Each aggregate (or function) can define one or several channels depending on its requirements. Nevertheless two different Aggregates could not share their channel.

One aggregate can hold one or more channels but can not share any channel with a different Aggregate

13.6.2 Saving criteria

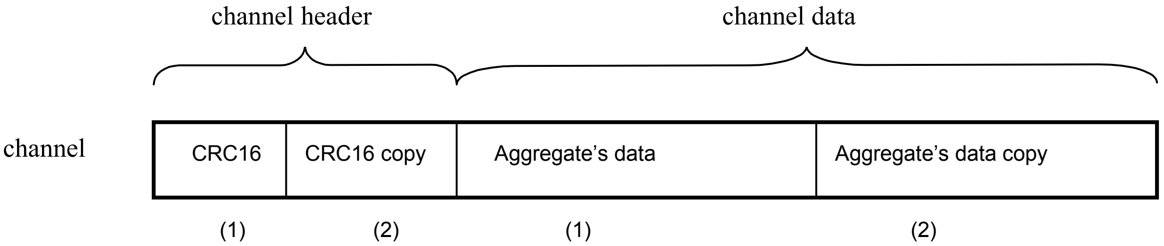
Data that belongs to the same channel are secured by a single saving criterion taken from the list below:

CRC 16 : channel integrity is covered by checksum calculation CRC16



If the CRC16 coherency check with the aggregate's data is failed, the channel is declared as corrupted: initialization needs to be performed.

MASTER and COPY : aggregate's data and CRC16 are duplicated in a zone called COPY



(1) master block
(2) copy block

Process of restoration:

Master check block (*)	Copy block check (*)	Result block	comment
ok	ok	master	last updated information
ok	nok	master	last updated information
nok	ok	copy	last updated information
nok	nok	initialization has to be done	both blocs are corrupted

(*) master/copy check block is ok means that CRC16 coherency check is coherent with the aggregate's data

This method is only available for EEPROM hardware component.

NONE: there is not saving criterion to guaranty the channel data integrity.

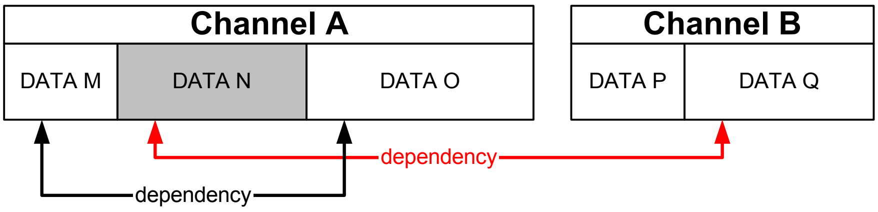
Non volatile memory data that should be secured with two different criteria can not share the same channel. Nevertheless, you may take the strongest criteria in order to group these data.

NVMY data within a channel are secured with the same saving criteria

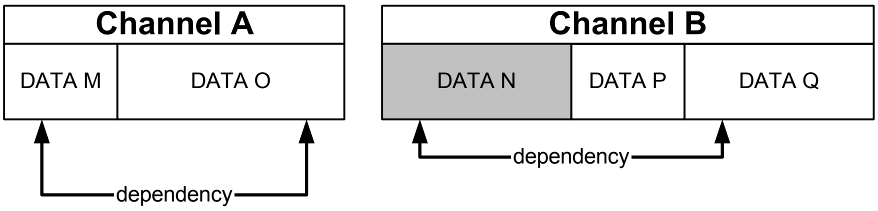
13.6.3 NVMY data coherency

As NVMY channels are separately handled by ECME aggregate, you should take care in the way you group data. Indeed, you should not separate data that are linked together. That is to say data that might be initialised at the same time in case of failure.

WRONG ORGANIZATION



CORRECT ORGANIZATION



DATA of the same Aggregate that are functionally linked should be grouped in the same channel.

13.6.4 Channel management

13.6.4.1 STANDARD channel

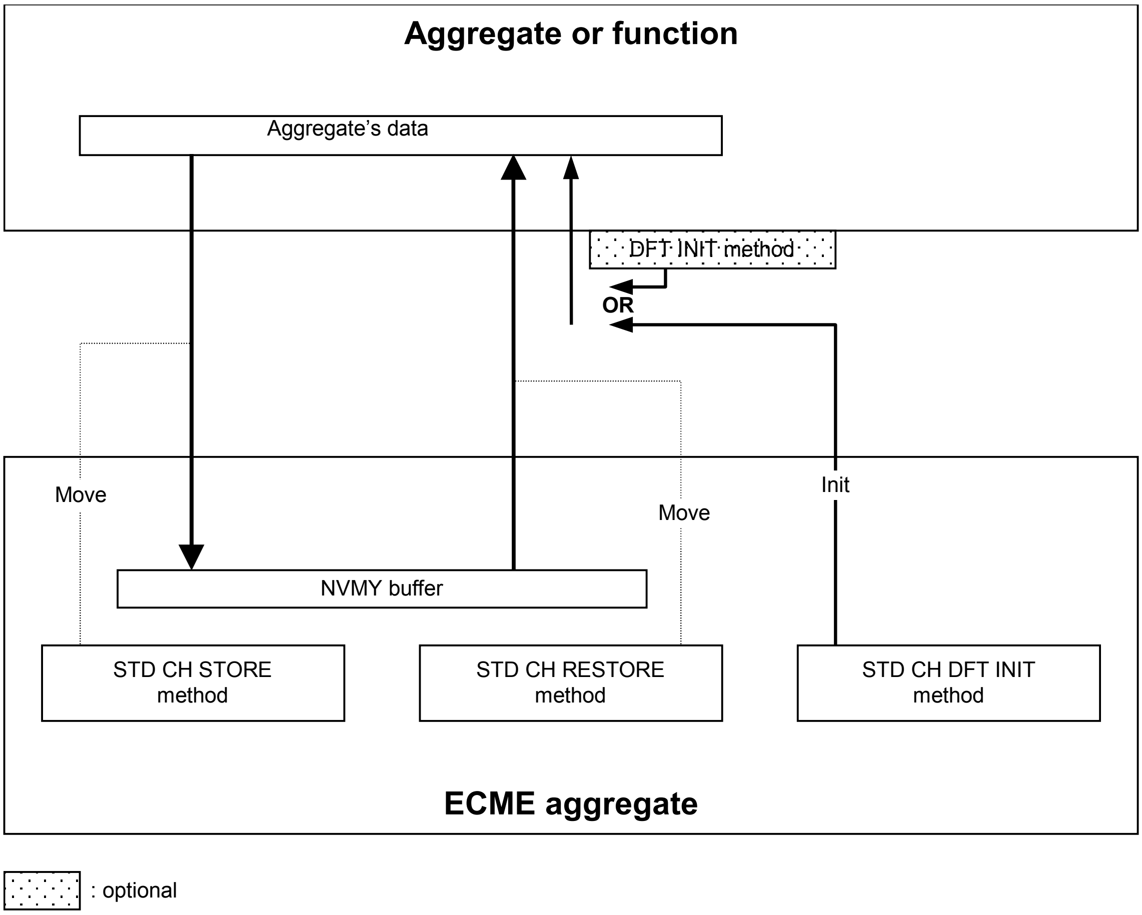
Aggregate do not have to provide any routine for NVMY data management except an initialisation routine if the default value does not match the Aggregate requirement (*).

In this case, the Aggregate relies on the ECME services for following purposes:

RESTORE
from Hardware component to Aggregate's data
STORE
from Aggregate's data to Hardware component

- DEFAULT INITIALIZATION [**optional** *]

The following picture describes the way that non-volatile data are managed in case of a STANDARD channel:



REMARK:

Aggregate's data memory location is known by the method boxes.
Accordingly, ECME aggregate should know of Aggregate's data declaration.

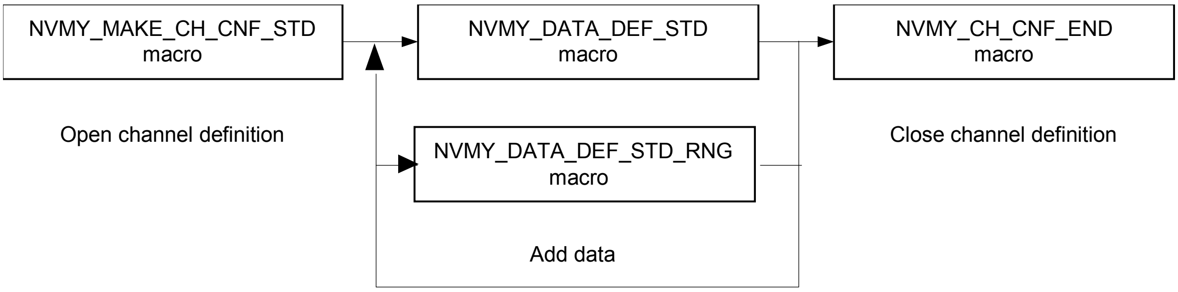
The use of STANDARD channel should be limited in order to avoid constant management overhead.

STANDARD Channel should be used for a few data only. Therefore, you should consider that a STANDARD channel can not hold more than 10 data.

13.6.4.1.1 Channel configuration

An ECME aggregate provides several MACROs that help the designer in order to define Channel configuration.
This configuration takes place into the ECME_ANVM.CFG component that is described in later section of this document.

Standard channels are defined according to the following scheme:



13.6.4.1.2 MACROs description

NVMY_MAKE_CH_CNF_STD (CH_ID,
TGT_ID,
VERSION_CHK,
TUNING_LVL,
SAVING_METH,
DFT_INIT_METH)

This macro opens a standard channel definition.
Besides, it helps to define all channel characteristics.

Parameters definition:

Parameter	**Value**	**Purpose**	**Comment**
CH_ID	NAME_N[_FIX] **(1)**	Channel identifier	Used to order ECME operation on a channel
TGT_ID	E2P	EEprom is the Preferred hardware location for storing this channel.	**This channel is stored to Eeprom if component is available. Otherwise, it will be stored to FLASH.**
	SIM_E2P	FLASH is the Preferred hardware location for storing this channel.	**This channel is stored to Eeprom if component is available. Otherwise, it will be stored to EEPROM.**
VERSION_CHK	CHK	Channel is sensitive to NVMY coherency check	If the NVMY contents coherency checking failed, the channel's data should be initialized with default values
	NO_CHK	Channel is not sensitive to NVMY coherency check	Channel's data is not initialized even if NVMYcontents coherency checking failed
TUNING_LVL	LV0 **(2)**	Channel is part of level 0 group for tuning initialization	Protected group : ECME's Tuning calibration has no effect on this group
	LV1	Channel is part of level 1 group for tuning initialization	Channel's contents may be initialized with default values if tuning calibration is set to level 1
	LV2	Channel is part of level 2 group for tuning initialization	Channel's contents may be initialized with default values if tuning calibration is set to level 2
	LV3	Channel is part of level 3 group for tuning initialization	Channel's contents may be initialized with default values if tuning calibration is set to level 3
ORE	NO_STRQ (4)	Synchronization for store operation	Copy data from previous block and don't take in account data from current driving cycle.
	NO_SYNC (5)	No operation done at Store phasis	Synchronisation and write operation is forbidden.
	END_PWL	Store at the end of Powerlatch	END_PWL and BEG_PWL do the same operations : store is done at the end of powerlatch
	BEG_PWL	Store at the beginning of powerlatch	
SAVING_METH	CRC16	CRC16 method	Channel's data are secured with CRC16 method
	MACOPY	MACOPY method	Channel's data are secured with MACOPY method
	NONE	NONE method	Channel's data are secured with NONE method
DFT_INIT_METH	NVMY_DATA_INIT_DFT	Aggregate uses the ECME's default init method	Channel data are initialized by ECME's method in case of troubles.
	Default Init **function name (3)**	specific method provided by Aggregate	Aggregate provides a method in order to initialize channel's data.

Remarks:

(1) : According to the “C coding guideline”, the channel identifier name is limited to **17** characters.
For convenience, **NAME** shall refer to Aggregate’s short name.
N is a digit that may be used in the case that Aggregates holds more than one channel.
The **FIX** may be added to channel identifier name if channel is not sensitive to NVMY data coherency checking.

(2) : This level is basically used to protect channels against tuning initialisation.
For instance, the channel dedicated to NVMY data coherency is protected.

(3): Basically, this field refers to the function name defined by Aggregate.

(4) : The channels that have the NO_STRQ parameter are only synchronized at powerlatch. For the first powerlatch, they are stored but in the case that the first powerlatch is interrupted the channels are not stored and will be initialized at every power on until they are stored into flash. The aggregate should take this into account and store the channel separately in case of initialization.

(5) : The channels that have NO_SYNC parameter are never stored nor synchronized to memory. The aggregate should take this into account and store the channel separately.

NVMY_DATA_DEF_STD(nvmy_data)

This macro helps to add a single data to the channel.
The nvmy_data corresponds to the real name of the NVMY data.
The name of the data must be limited to 15 characters.

NVMY_DATA_DEF_STD_RNG(name, address, size)

This macro helps to add a data to the channel.
The name must be limited to 15 characters.
Address corresponds to the NVMY data address begin.
Size is the length of the NVMY data in byte and must be limited in order to not increase the channel size excessively.

NVMY_CH_CNF_END(CH_ID)

This macro closes a standard channel definition.

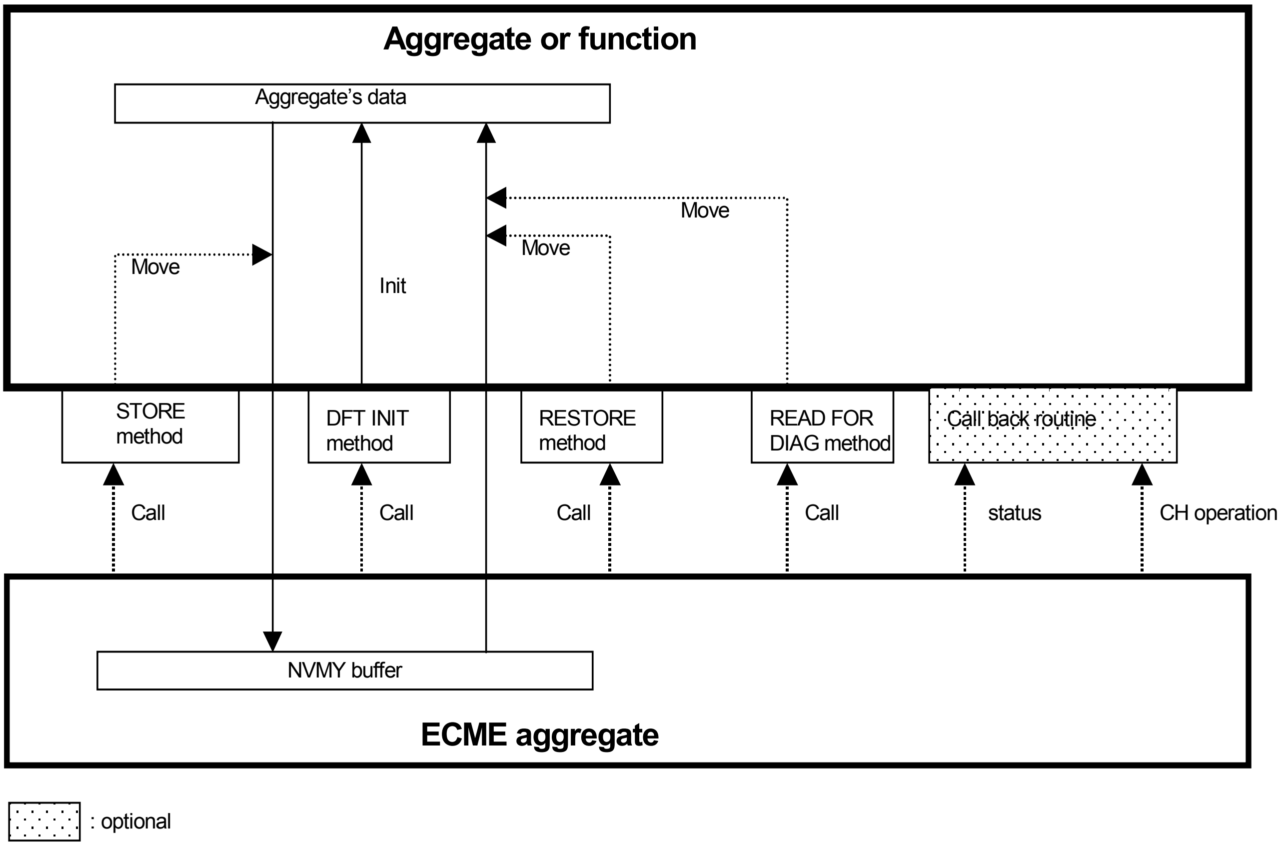
CH_ID is the same as the one defined by NVMY_MAKE_CH_CNF_STD macro.

13.6.4.2 SPECIFIC channel

Aggregates provide all methods for NVMY data management as follows:

RESTORE
from Hardware component to Aggregate's data
STORE
from Aggregate's data to Hardware component
DEFAULT INITIALIZATION
Aggregate's data initialization with default value

- CALLBACK routine [optional]
  Call back routine called by ECME aggregate after a store operation in order to get its status.
- READ FOR DIAGNOSIS
  From hardware component to the data used by the Read for diagnosis service



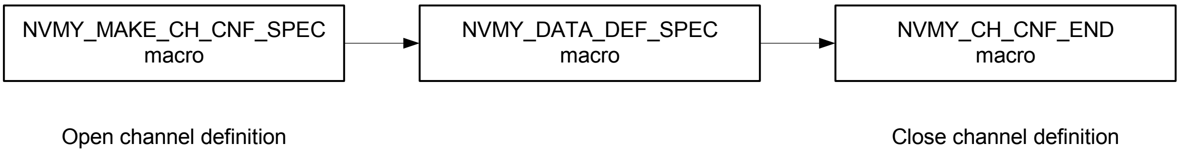
REMARK:

Aggregate's data memory location is known by the method boxes.
Thus, ECME aggregate does not have to know Aggregate's data memory location.

13.6.4.2.1 Channel configuration

An ECME aggregate provides several MACROs that help the designer in order to define Channel configuration.
This configuration takes place into the ECME_ANVM.CFG component that is described in later section of this document.

Standard channel are defined according to the following scheme:



13.6.4.2.2 MACROs description

```
NVMY_MAKE_CH_CNF_SPEC (   CH_ID,
                          TGT_ID,
                          VERSION_CHK,
                          TUNING_LVL,
                          SAVING_METH,
                          STORE_METH,
                          RESTORE_METH,
                          CBACK_METH,
                          DFT_INIT_METH,
                          READ_FOR_DIAG_METH)
```

This macro opens a standard channel definition.
Besides, it helps to define all channel characteristics.

Parameters definition:

Parameter	Value	Purpose	Comment
CH_ID	NAME_N[_FIX] **(1)**	Channel identifier	Used to order ECME operation on a channel
TGT_ID	E2P	EEprom is the Preferred hardware location for storing this channel.	**This channel is stored to Eeprom if component is available. Otherwise, it will be stored to FLASH.**
	SIM_E2P	FLASH is the Preferred hardware location for storing this channel.	**This channel is stored to Eeprom if component is available. Otherwise, it will be stored to EEPROM.**
VERSION_CHK	CHK	Channel is sensitive to NVMY coherency check	If the NVMY contents coherency checking failed, the channel's data should be initialized with default values
	NO_CHK	Channel is not sensitive to NVMY coherency check	Channel's data is not initialized even if NVMYcontents coherency checking failed
TUNING_LVL	LV0 **(2)**	Channel is part of level 0 group for tuning initialization	Protected group : ECME's Tuning calibration has no effect on this group
	LV1	Channel is part of level 1 group for tuning initialization	Channel's contents may be initialized with default values if tuning calibration is set to level 1
	LV2	Channel is part of level 2 group for tuning initialization	Channel's contents may be initialized with default values if tuning calibration is set to level 2
	LV3	Channel is part of level 3 group for tuning initialization	Channel's contents may be initialized with default values if tuning calibration is set to level 3

Parameter	Value	Purpose	Comment
ORE	NO_STRQ (4)	Synchronization for store operation	Copy data from previous block and don't take in account data from current driving cycle.
	NO_SYNC (5)	No operation done at Store phasis	Synchronisation and write operation is forbidden.
	END_PWL	Store at the end of Powerlatch	END_PWL and BEG_PWL do the same operations : store is done at the end of powerlatch
	BEG_PWL	Store at the beginning of powerlatch	
SAVING_METH	CRC16	CRC16 method	Channel's data are secured with CRC16 method
	MACOPY	MACOPY method	Channel's data are secured with MACOPY method
	NONE	NONE method	Channel's data are secured with NONE method
STORE_METH	Store **function name** **(3)**	specific method provided by Aggregate	Aggregate provides a method in order to copy NVMY data from ram to nvmy buffer (ECME)
RESTORE_METH	Restore **function name** **(3)**	specific method provided by Aggregate	Aggregate provides a method in order update its NVMY data thanks to NVMY buffer
CBACK_METH	NVMY_DFT_CBACK	ECME call-back default method	Call back is not active for this channel.
	call back **function name** **(3)**	specific method provided by Aggregate	Aggregate provides a method that will be called by ECME in case of writing on fly
DFT_INIT_METH	NVMY_DATA_INIT_DFT	Aggregate uses the ECME's default init method	Channel data are initialized by ECME's method in case of troubles.
	Default Init **function name** **(3)**	specific method provides by Aggregate	Aggregate provides a method in order to initialize channel's data.
READ_FOR_DIAG_METH	NVMY_DFT_READ_FOR_DIAG	Aggregate uses the ECME's Read For Diag default method	Read For Diagnosis is not active for this channel
	Read for diag **function name** **(3)**	specific method provides by Aggregate	Aggregate provides a method in order to Read For Diagnosis channel's data.

Remarks:

(1): According to the "C coding guideline", the channel identifier name is limited to **17** characters.
For convenience, **NAME** shall refer to Aggregate's short name.
N is a digit that may be used in the case that Aggregates holds more than one channel.
The **FIX** may be added to channel identifier name if channel is not sensitive to NVMY data coherency checking.

(2): This level is basically used to protect channels against tuning initialisation.
For instance, the channel dedicated to NVMY data coherency is protected.

(3): Basically, this field refers to the function name defined by Aggregate.

(4): The channels that have the NO_STRQ parameter are only synchronized at powerlatch. For the first powerlatch, they are stored but in the case that the first powerlatch is interrupted the channels are not stored and will be initialized at every power on until they are stored into flash. The aggregate should take this into account and store the channel separately in case of initialization.

(5): The channels that have NO_SYNC parameter are never stored nor synchronized to memory. The aggregate should take this into account and store the channel separately.

NVMY_DATA_DEF_SPEC(aggr_typedef_struct)

This macro is used to add a package of NVMY data to channel.
This package is defined through the typedef structure provided by Aggregate

NVMY_CH_CNF_END(CH_ID)

This macro closes a specific channel definition.

CH_ID is the same as the one defined by NVMY_MAKE_CH_CNF_SPEC macro.

13.6.5 Tuning Initialisation

This section treats all tuning aspects related to NVMY functionality.
During life of project, developers may need to initialise some channels with their default values.
In this way ECME provides 2 calibrations that are used in this purpose.

Nevertheless, channels should be correctly configured. That is to say, channel’s level should be defined according to the way that developers intend to use tuning calibrations.

Afterwards, these calibrations may be used to initialise a set of channels defined by the user.

Besides, ECME provides a safe way for channel initialisation. Indeed, initialisations are made during ECU’s start-up only.

13.6.5.1 Calibration description

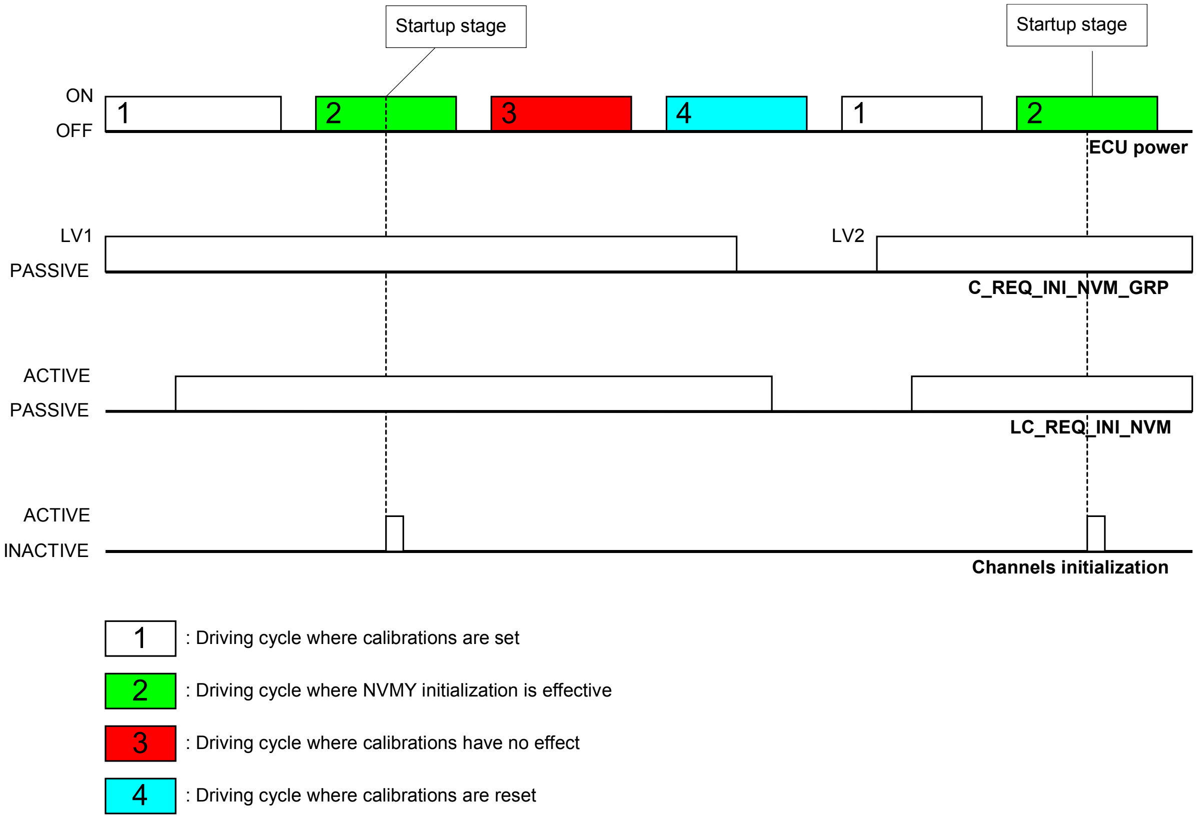
From SAM 2000 point of view, these calibrations belong to the ECME_NVM_DEV group.
In addition, the values are given according to SAM view.

Calibration name	Value	Purpose
lc_req_ini_nvm	PASSIVE	No request for initialization
	ACTIVE	Initialization is required
c_req_ini_nvm_grp	0	unused
	1	channel that belong to group level 1should be initialized
	2	channel that belong to group level 2should be initialized
	3	channel that belong to group level 3 should be initialized

Lc_req_ini_nvm is the trigger for NVMY initialisation, whereas c_req_ini_nvm_grp is used to identify the channel group.

You should notice that initialisation is only effective when lc_req_ini_nvm is set is changed from PASSIVE to ACTIVE. In addition, Initialisations are only done at start-up.

These calibrations should be used according to the following diagram:



13.6.5.2 First Power On

After the first power on, the Non Volatile Memory must operate its own initialisation.

So, an important delay may occur between the first power on of the Ecu and the availability of the whole NVMY strategy.

In consequence, all aggregate using NVMY functionality must first wait for the end of this first initialisation.

13.7 Aggregate ECME adaptation

13.7.1 Outputs for BMW functions which are not defined as ECME exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_pwf	V/O	0...1	0...1	1	-
0: reading of customer adaptation block in the E2PROM was successful // 1: not sucessful					

Input data:

LV_NVMY_CUS_WRG			

FUNCTION DESCRIPTION:

General information:

There are customer provided adaptation values stored in the E2Prom. Therefore the information has to be created whether the re-loading of this values out of the NVMY was successful or not. The bit B_pwf is containing this information (0 = upload was successful // 1 = upload failed).

Application conditions:

Initialisation: B_pwf = 0 at reset

Recurrence: once after up-loading of the adaptation values out of the E2Prom

Activation: after up-loading the customer adaptation values out of the E2Prom

Deactivation: after set of B_pwf

Formula section:

B_pwf = LV_NVMY_CUS_WRG if non-volatile stored data of storage class .Nvram (all customer adaptation values) could not be re-loaded error free out of the E2PROM

14 ECU Operation

Table of Contents

Index

14.1 ECU Warm-reset detection, reset counter, debug-info

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_WARM_RST	V/O	0...1H	0...1	1	[-]
flag shows, that a reset during current ECU-run has occured					
LV_RST_DET	V/S	0...1H	0...1	1	[-]
Warm-reset during ECU-lifetime (until clearing FMY) occured					
RST_TYP	V/O	0...FFH	0...255	1	[-]
Status-byte: reset type during current ECU-run					
RST_SEC	V/O	0...FFH	0...255	1	[-]
Status-byte: security info for atypical reset (reset-save memory)					
RST_INFO_CTR	V/O	0...FFH	0...255	1	[-]
Number of atypical warm-resets since last power-up (BSW)					
RST_CLAS_TYP[NC_RST_DBG_LIST_SIZE]	V/O/S	0...FFH	0...255	1	[-]
Reset type					
RST_CLAS_SEC[NC_RST_DBG_LIST_SIZE]	V/O/S	0...FFH	0...255	1	[-]
Security info					
RST_INFO_ADD[NC_RST_DBG_LIST_SIZE]	V/O/S	0...FFFFFFFFH	0...4294967295	1	[-]
Additional reset info (cause)					
RST_DBG_BACK_INFO_VLD[NC_RST_DBG_LIST_SIZE]	V/O/S	0...1H	0...1	1	[-]
background info for last reset valid					
RST_DBG_BACKTRACE_ADDRESS[NC_RST_DBG_LIST_SIZE][NC_RST_DBG_BACKTRACE_SIZE]	V/O/S	0...FFFFFFFFH	0...4294967295	1	[-]
Info last available caller addresses (default 0)					
RST_DBG_BACK_UPPER_CONTEXT[NC_RST_DBG_LIST_SIZE][15]	- /S	0...FFFFFFFFH	0...4294967295	1	[-]
Background info upper context (default 0)					
RST_DBG_BACK_PROC_STATUS[NC_RST_DBG_LIST_SIZE][9]	- /S	0...FFFFFFFFH	0...4294967295	1	[-]
Background info processor status (default 0)					
RST_DBG_BACK_ENVIRONMENT[NC_RST_DBG_LIST_SIZE][23]	- /S	0...FFFFFFFFH	0...4294967295	1	[-]
Background info environment (default 0)					
CTR_ADC_OUT	V	0...FFH	0...255	1	[-]
Counter for the number of ADC interface error messages					
ERR_SYM_WARM_RST	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
detected error symtom ECU-reset					
LV_ERR_WARM_RST	V/O	0...1H	0...1	1	[-]
error bit of ECU-Reset detection					
LV_CDN_DIAG_WARM_RST	V/O	0...1H	0...1	1	[-]
diagnosis condition ECU-reset					
LV_END_DIAG_WARM_RST	V/O	0...1H	0...1	1	[-]
end of diagnosis ECU-reset					
CPU_LOAD_MAX_RST_DET[NC_RST_DBG_LIST_SIZE]	V/O/S	0...400H	0...100	0.0976563	[%]
Max. CPU load from reset detection					
N_CPU_LOAD_MAX_RST_DET[NC_RST_DBG_LIST_SIZE]	V/O/S	0...1FE0H	0...8160	1	[rpm]
Engine speed at max. cpu load from reset detection					

CPU_LOAD_RST_DET[NC_RST_DBG_LIST_SIZE]	V/O/S	0...400H	0...100	0.0976563	[%]
CPU load at reset					
N_RST_DET[NC_RST_DBG_LIST_SIZE]	V/O/S	0...1FE0H	0...8160	1	[rpm]
Engine speed at reset					
DIST_RST_DET[NC_RST_DBG_LIST_SIZE]	V/O/S	0...FFFFFFFFH	0... 429496729500	100	[m]
Mileage counter at reset					
TRT_RST_DET[NC_RST_DBG_LIST_SIZE]	V/O/S	0...FFFFFFFFH	0... 119304.64708	0.0278e-3	[h]
Total runtime at reset					
LV_RST_DET_DISP	-	0...1H	0...1	1	[-]
ECU-Warm-Reset displayed in error management					

Input data:

TRT		DIST	N
CPU_LOAD	CPU_LOAD_MAX		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_WARM_RST_ENA	1	0...1H	0...1	1	[-]
Entry in error management enabled (=1)					
C_N_RST_DET	1	0...1FE0H	0...8160	1	[rpm]
Engine speed for activation of reset detection in errror-management					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_RST_DBG_LIST_SIZE	1	0...FFH	0...255	1	[-]
Number of bytes in info array					
NC_RST_DBG_BACKTRACE_SIZE	1	0...FFH	0...255	1	[-]
Number of byte in backtrace info array					
NC_MAX_ADC_OUT	1	0...FFH	0...255	1	[-]
Maximum number timeout ADC					
NC_CTR_MAX_ADC_OUT	1	0...FFH	0...255	1	[-]
Maximum counter for ADC interface error messages					

Export actions:

ACTION_SetRstInfo_ECM3	

Action sets reset information of ecm3 security system

Import actions:

ACTION_INFR_GetBiosAdcStatus (OUT <Bios_adc_status >)
This action provides the digitized status of analogous signals

General information:

The goal is to memorise the number and the reason of irregular ECU-resets (not at the normal power-up !!), in order to support the ECU-development.
The reset-information will NOT be saved in E^{2}PROM for evaluation at the car-service-station.
Related hardware: TC1796 / system Units32-Bit / Single-Chip Microcontroller

The new debug driver is called by debug events which are typically resets. After a debug event it is necessary to ensure that the event- and the diagnosis-information is stored.

In case of E2PROM error or standby function call all informations are cleared.

The following figures show, which different values and additional info can be evaluated and stored for later access:

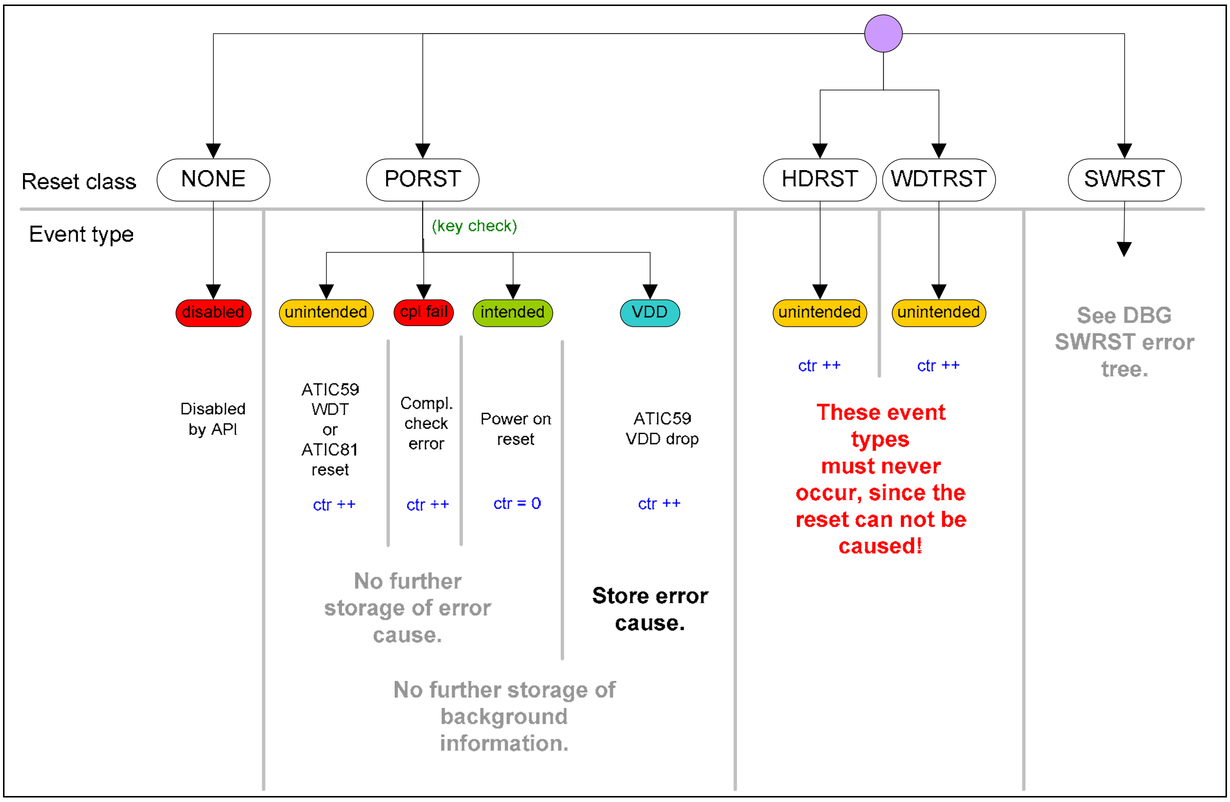


Figure1: DBG main reset class

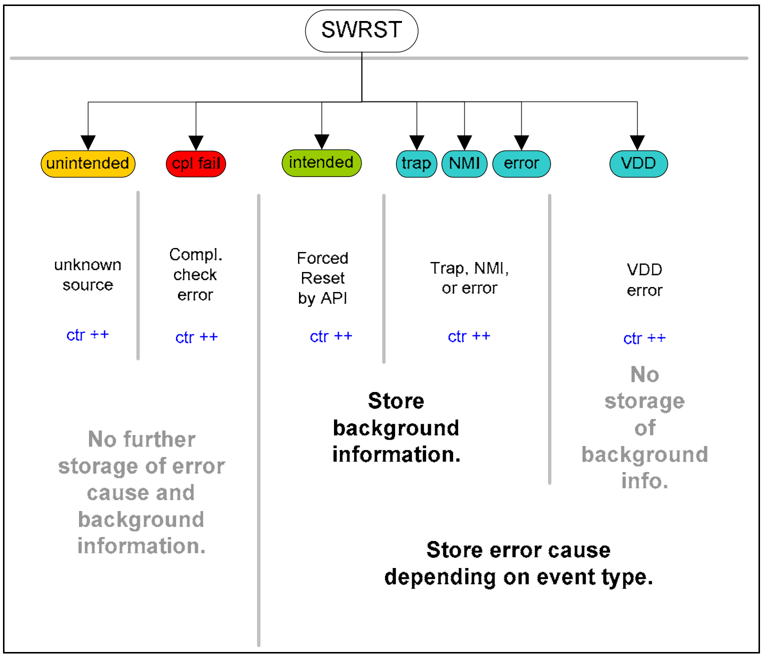


Figure 2: DBG SWRST reset tree

As described in figure1 the reset class represents a main category of reset types and the described event. The type represents a more detailed description of an occured reset, only the event type is used here as RST_TYP.

The error cause is more detailled or is set to 0 if not valid. It is used here as RST_INFO_ADD.

The background information is only stored/updated for event types as shown above. Only some backtrace information is provided at the moment, named RST_DBG_BACKTRACE_ADDRESS. The information is marked valid or invalid through RST_DBG_BACK_INFO_VLD. The complete info is set to 0 if not valid.

In a further step a warm/cold-Reset-detection is done by following mechanism:
A reset-save key which is written at ECU power-up and destroyed at regular ECU power-down. Checking this key against its init-value allows to detect a warm-reset.
By comparing the debug driver and the reset-save key it is possible to distinguish between internal or external MC-resets.

Reset-detection is done by evaluating low level functionality at startup:

RST_TYP
This value represents, whether an intended Power-on reset (cold start, 12h) or any other reset is detected at startup.

Hex-val	Description
01h	Error at evaluation (ERROR_TYPE)
02h	Driver disabled (DISABLED_TYPE)
10h	Unintended Power-On Reset (WDT or ATIC81, PORST_TYPE_UNINTENDED)
11h	Complement error at Power-On Reset (PORST_TYPE_CPL_FAIL)
12h	Intended Power-On Reset (Cold Start, PORST_TYPE_INTENDED)
13h	VDD Error at Power-On Reset (PORST_TYPE_VDD_ERROR)
20h	Hardware Reset (HDRST_TYPE_UNINTENDED)
30h	Watchdog Reset (WDTRST_TYPE_UNINTENDED)
40h	Unintended Software Reset (SWRST_TYPE_UNINTENDED)
41h	Complement error at Software Reset (SWRST_TYPE_CPL_FAIL)
42h	Intended Software Reset (SWRST_TYPE_INTENDED)
43h	VDD Error at Software Reset (SWRST_TYPE_VDD_ERROR)
44h	Trap (detected trap, SWRST_TYPE_TRAP)
45h	NMI (detected NMI/Trap class 7, SWRST_TYPE_NMI)
46h	Error at Reset (SWRST_TYPE_ERROR)

This value represents, whether a Power-on reset or any other reset is detected at startup.

RST_SEC
This value represents, whether the implemented security mechanism ECM3 has notified the module after restart through the interface ACTION_SetRstInfo_Ecm3 (see below) about an executed reset.

Hex-val	Description
00h	No additional Security info
01h	Additional Security info registered since last power-up

RST_INFO_CTR

Hex-val	Description
00h	Unspecified / Additional Reset info invalid
xxh	Number of atypical warm resets since last power-up

This value represents the number of atypical warm resets since last power-up, if additional reset info is valid. On Power-on reset this value is reset to 0.

Except for an intended Power-on reset (cold start, 12h) some additional informations may be provided as a list of the last detected resets. The list entry [0] represents the last/newest one, [1] that one before the last and so on up to the maximum size of the list with the length of NC_RST_DBG_LIST_SIZE. Before evaluating a new reset the whole list is shifted by one step and therefore the oldest list entry [NC_RST_DBG_LIST_SIZE-1] is lost.

Information lists are defined as arrays:
NC_RST_DBG_LIST_SIZE (see module configuration data: e.g. MSx80=2, MSD85=1)
NC_RST_DBG_BACKTRACE_SIZE = 10

RST_CLAS_TYP[NC_RST_DBG_LIST_SIZE]

This list value represents the same value as RST_TYP (see above), if reset differs from an intended Power-on reset at startup.

RST_CLAS_SEC[NC_RST_DBG_LIST_SIZE]

This list value represents the same value as RST_SEC (see above), if reset differs from an intended Power-on reset at startup.

RST_INFO_ADD[NC_RST_DBG_LIST_SIZE]

This list value represents an additional info on the cause of the reset.

RST_INFO_ADD at RST_CLAS_TYP = 42h (Intended Software Reset)

If an intended software reset is triggered by the following cases, the equivalent value is stored:

Hex-val	Description
00000000h	no info available
00000100h	Caused by Immobilizer (IMOB)
00000200h	Caused by Reprogr. (REPROG learn)
00000201h	Caused by Reprogr. (REPROG kwp)
00000300h	Caused by Security Control (SEC_MC)
00000400h	Caused by Memory Support (SUPRAM)
00000401h	Caused by Memory Support (NVMY)
00000402h	Caused by ADC driver (BIOS_ADC)
00000404h	Caused by injection (INJR)
00000500h	Caused by VIgk Peak (VDD_IGK)

RST_INFO_ADD at RST_CLAS_TYP = 44h (Trap)

If a software reset by trap occured, the equivalent value is stored in format:

Hex-val	Description
xxxx0000h	Trap value (high word)
0000yyyyh	Tin value (low word)

The trap classes represent more in detail, which type of trap occurred.

A trap occurs as a result of an event such as a Non-Maskable Interrupt (NMI), an instruction exception, memory-management exception or an illegal access. Traps are always active; i.e. they cannot be disabled by software action.
The TriCore architecture specifies general classes for traps. Each class has its own trap handler, accessed through a trap vector of 32 bytes per entry, indexed by the hardware-defined trap class number.

Hex-val	Description
00FFh	No trap deteced
0000h	memory management unit
0001h	Internal Protection Traps
0002h	Instruction Error
0003h	Context Management
0004h	System bus and peripherals error
0005h	Assertion traps

Within each trap class, specific traps are distinguished by a Trap Identification Number (TIN) that is loaded by hardware into a register before the first instruction of the trap handler is executed.

Note: The TC1796 does not include a MMU, no traps in class 0 will be generated.

The following Tin information is available depending on the trap class:

Info TIN in trap class 0

Hex-val	Description
FFh	no tin deteced
00h	Virtual Address Fill
01h	Virtual Address Protection

Info TIN in trap class 1

Hex-val	Description
FFh	no tin deteced
01h	Privileged Instruction
02h	Memory Protection: Read Access
03h	Memory Protection: Write Access
04h	Memory Protection: Execution Access
05h	Memory Protection: Peripheral Access
06h	Memory Protection: Null Address
07h	Global Register Write Protection

Info TIN in trap class 2

Hex-val	Description
FFh	no tin deteced
01h	Privileged Instruction
02h	Memory Protection: Read Access
03h	Memory Protection: Write Access
04h	Memory Protection: Execution Access
05h	Memory Protection: Peripheral Access
06h	Memory Protection: Null Address
07h	Global Register Write Protection

Info TIN in trap class 3

Hex-val	Description
01h	Free context list depleted (FCX == LCX)
02h	Call depth overflow
03h	Call depth underflow
04h	Free context list underflow (FCX == 0)
05h	Call stack underflow (PCX == 0)
06h	Context type error (PCXI.UL wrong)
07h	Nesting error: RFE with non-zero call depth

Info TIN in trap class 4

Hex-val	Description
01h	Program fetch bus error
02h	Data load bus error
03h	Data store bus error

Info TIN in trap class 5

Hex-val	Description
01h	Arithmetic overflow
02h	Sticky arithmetic overflow

RST_INFO_ADD at RST_CLAS_TYP = 45h (NMI)

If an NMI is the reason one of the following cases with their equivalent value is stored:

Hex-val	Description
00000100h	External NMI
00000200h	NMI caused by WDT
00000300h	PLL loss of lock
00000400h	Parity error in DMI memory

00000401h	Parity error in SPRAM or Instruction cache
00000402h	Parity error in PMI memory
00000403h	Parity error in DMU memory
00000404h	Parity error in PCP RAM memory
00000405h	Parity error in PCP ROM memory
00000406h	Parity error in CAN memory

RST_DBG_BACK_INFO_VLD[NC_RST_DBG_LIST_SIZE]

If additional background information is available, this info shows, whether these info is valid. In other cases 0.

Hex-val	Description
00h	Additional Background infos invalid
01h	Additional Background infos valid

RST_DBG_BACKTRACE_ADDRESS[NC_RST_DBG_LIST_SIZE][NC_RST_DBG_BACKTRACE_SIZE]

If additional background information is valid, a list of NC_RST_DBG_BACKTRACE_SIZE 32-bit values is provided for further investigation purposes.

Hex-val	Description
xxxxxxxxh	Backtrace address 0
yyyyyyyyh	Backtrace address 1
zzzzzzzzh	...

These additional background informations are stored, but not provided at the moment:

RST_DBG_LAST_UPPER_CONTEXT[NC_RST_DBG_LIST_SIZE][15]

RST_DBG_PROC_STATUS[NC_RST_DBG_LIST_SIZE][9]

RST_DBG_ENVIRONMENT[NC_RST_DBG_LIST_SIZE][23]

Before a SW-reset is explicitely executed, it is ensured that the debug driver is activated by calling the DBG routine. This enables the storage of the following bytes in the reset safe memory as valid debug informations.

All additional reset-debug-informations are saved in EEPROM and contain the cause for the last warm-resets.

Activation:

once, directly after each ECU-reset; before starting the operation-system

Formula section:

Evaluate controller information:

RST_TYP

```
RST_SEC = 0 (reset to 0)

If (1)    RST_TYP = 12h (Intended Power-On Reset)
Then (1)
    Power-On-Reset detected
    LV_WARM_RST = 0
Else (1)
    warm-Reset detected
    RST_INFO_CTR = Reset Counter from Debug driver
    Shift all list entries by one, eg.              (first in first out of stored informations)
    RST_CLASS_TYP[NC_RST_DBG_LIST_SIZE-2]
            -> RST_CLASS_TYP[NC_RST_DBG_LIST_SIZE-1]
                    ->
    RST_CLASS_TYP[0]
            -> RST_CLASS_TYP[1]
    Erase newest list entries, eg.
    RST_CLASS_TYP[0] = 0
    Evaluate and store actual values in list
    RST_CLASS_TYP[0] = RST_TYP
    RST_INFO_ADD[0] = Cause from Debug driver     (Additional Info)
    CPU_LOAD_MAX_RST_DET[0] = CPU_LOAD_MAX at reset
    N_CPU_LOAD_MAX_RST_DET[0] = N_MAX at CPU_LOAD_MAX
    CPU_LOAD_RST_DET[0] = CPU_LOAD at reset
    N_RST_DET[0] = N at reset
    DIST_RST_DET[0] = DIST at reset
    TRT_RST_DET[0] = TRT at reset
    If (2) RST_CLASS_TYP[0] = 42h (Intended Software Reset)
            RST_LPRC_RESET_ADDRESS[0] = Last  LPRC caller address
    Else (2)
            RST_LPRC_RESET_ADDRESS[0] = 0
    Endif (2)
    RST_DBG_BACK_INFO_VLD[0] = Validation info from Debug driver
    If (2) RST_DBG_BACK_INFO_VLD[0] = 1
            RST_DBG_BACKTRACE_ADDRESS[0][ ] = Backtrace from Debug driver
            RST_DBG_BACK_UPPER_CONTEXT[ ] = Context from Debug driver
            RST_DBG_BACK_PROC_STATUS[ ] = Processor status from Debug driver
            RST_DBG_BACK_ENVIRONMENT[ ] = Environment from Debug driver
    Endif (2)
    LV_WARM_RST = 1
    LV_ RST_DET = 1
```

Endif (1)

Later notification from security system:

Activation: **Action at notification from ECM3 (ACTION_SetRstInfo_Ecm3)**

Formula section:

LV_RST_DET_DISP = 0 (reset flag until error reported)
RST_SEC = 1 (add ECM3 security info)

The safety-concept will submit the information after a reset, when rst-information from the MU is available. As it is not known, whether or when such a reset can be detected by security system, the trap reset can set error SYM_0 before it will be set to SYM_1 by security system.

Clearing of stored data can be done on demand:

Activation: **Action at clearing FMY (e.g. by service-tester)**

Formula section:

RST_TYP = 0
RST_INFO_CTR = 0 Clear Reset counter from Debug driver also
RST_SEC = 0
LV_RST_DET_DISP = 0
LV_RST_DET = 0
LV_WARM_RST = 0

Clear all list entries, eg.
RST_CLASS_TYP[NC_RST_DBG_LIST_SIZE-2] = 0
...
RST_CLASS_TYP[0] = 0

Activation: **Action for status ADC from interface (ACTION_GetBiosAdcStatus)**

called every 10ms

Hint:
The Adc_watchdog_timeout describes the timeout of ADC interface in BIOS.

When the Adc_watchdog_timeout >= NC_MAX_ADC_OUT the interface is detected as out of operation and debug driver is called.

Formula section:

NC_MAX_ADC_OUT = 50 (Configuration BIOS: adc_watchdog_max_timeout)

```
NC_CTR_MAX_ADC_OUT = 50

ACTION_INFR_GetBiosAdcStatus (Bios_adc_status)

set counter CTR_ADC_OUT according STD_INI

IF      ACTION_INFR_GetBiosAdcStatus delivers ADC error
THEN    CTR_ADC_OUT$_n$ = CTR_ADC_OUT$_{n-1}$ + 1
ELSE    IF CTR_ADC_OUT$_n$ >0
        THEN CTR_ADC_OUT$_n$ = CTR_ADC_OUT$_{n-1}$ – 1
        ELSE CTR_ADC_OUT$_n$ = 0
        END
END

IF              CTR_ADC_OUT >= NC_CTR_MAX_ADC_OUT
THEN            call Reset Debug driver    (execute Reset caused by ADC interface)
                CTR_ADC_OUT = 0
END
```

14.1.1 Handling of ECU-Reset detection in error management

FUNCTION DESCRIPTION:

The following section describes the mechanism of failure handling in error management. Related to the reset type the error management carries out a failure symptom.

Application conditions:

after each ECU-reset
As information from the safety concept is only available late after restart, it has to be checked cyclic for changes (see later). So it is supposed first, that errors will be cleared first.

An entry in error management has occurred if the reset counter RST_INFO_CTR is incremented.

In this case actually the byte RST_CLAS_TYP[0] which include the causes of resets is additionally sent to error management.

Error symptoms are defined to this diagnosis function as follows:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 │  │  │  └─> -   Intended software reset (SWRST)    (= SYM_0)
 │  │  └────> -   Reset by safety-concept            (= SYM_1)
 │  └───────> -   Reset unintended by software       (= SYM_2)
 └──────────> -   Reset unintended by hardware       (= SYM_3)
```

Reset class	RST_TYPE	RST_INFO_ADD	RST_SEC	Symptom in ERRM
None	disabled #02h			
PORST	unintended #10h		0 1	SYM_3 SYM_1
	cpl fail #11h			SYM_3
	intended #12h			SYM_3
	VDD #13h			SYM_3
HDRST	unintended #20h			SYM_3
WDTRST	unintended #30h			SYM_3
SWRST	unintended #40h			SYM_2
	cpl fail #41h			SYM_3
	intended #42h	000h 300h (SEC_MC) 400h 401h (NVMY) 402h (ADC) 404h (INJR) 500h (VDD) 100h 200h 201h other (not expected)		SYM_0 SYM_1 SYM_0 SYM_0 SYM_0 SYM_0 SYM_3 NO SYM NO SYM NO SYM SYM_0
	trap #44h			SYM_2
	NMI #45h			SYM_2
	error #46h			SYM_2
	VDD #43h			SYM_3

at clearing FMY

The error is present until clearing FMY by application constant or flash. Additionally an information about current presence of failure is performed.

Hint: As described also in ASW the CPU_LOAD_MAX is reset with LC_CPU_LOAD_MAX_RST.

cyclic data collection

recurrency: 10 ms

In the warm-reset memory several data are cyclic saved every 10ms. CPU_LOAD_RST_DET is updated with CPU_LOAD. In case of an occured warm-reset the memory is read at initialisation of ECU. CPU_LOAD_RST_DET represents the CPU_LOAD at warm-reset. N_RST_DET is the related engine speed at this warm-reset. The CPU_LOAD_MAX_RST_DET is the occured maximum and saved in non-volatile memory. It gives information about several cycles. N_CPU_LOAD_MAX_RST_DET is the related engine speed at this maximum.

Related to warm-reset memory DIST_RST_DET represents the distance and TRT_RST_DET represents the total runtime at the last occured warm-reset.

By preparing the freeze frame it is necessary to hand over the relevant values. So the configured freeze frame is read out of warm-reset memory.

The data listed in the table are enabled to configure in freeze frame as environment data.
If a reset occurs and send to error-management the freeze frame is saved. Via tester the data are readable on the diagnostic communication line.

No.	**Environment data**
1.	RST_TYP
2.	RST_INFO_CTR
3.	RST_CLAS_TYP[0]
4.	RST_CLAS_SEC[0]
5.	RST_INFO_ADD[0]
6.	RST_DBG_BACK_INFO_VLD[0]
7.	RST_DBG_BACKTRACE_ADDRESS[0][0]
8.	CPU_LOAD_MAX_RST_DET[0]
9.	N_CPU_LOAD_MAX_RST_DET[0]
10.	CPU_LOAD_RST_DET[0]
11.	N_RST_DET[0]
12.	DIST_RST_DET[0]
13.	TRT_RST_DET[0]

```
Cyclic stored CPU_LOAD = CPU_LOAD
Cyclic stored N = N
Cyclic stored DIST = DIST
Cyclic stored TRT = TRT
if    CPU_LOAD_MAX > Cyclic stored CPU_LOAD_MAX
then
      Cyclic stored CPU_LOAD_MAX = CPU_LOAD_MAX
      Cyclic stored N_MAX = N
Endif
```

Application conditions:

Recurrence: 100ms

Initialisation: after clearing FMY
LV_CDN_DIAG_WARM_RST = 1
LV_RST_DET_DISP = 0

Activation: (*LV_WARM_RST = 1 AND*
LC_WARM_RST_ENA = 1 **AND**
N_RST_DET[0] > C_N_RST_DET) //cyclic saved engine speed//

Deactivation: (*LV_WARM_RST = 0 OR*
LC_WARM_RST_ENA = 0 **OR**
N_RST_DET <= C_N_RST_DET

Action at clearing FMY (e.g. by service-tester)

Cyclic stored CPU_LOAD_MAX = 0
Cyclic stored N_MAX = 0
Cyclic stored CPU_LOAD = 0
Cyclic stored N = 0
Cyclic stored DIST = 0
Cyclic stored TRT = 0

Formula section:

```
ERR_SYM_WARM_RST = NO_SYM
LV_ERR_WARM_RST = 0          (actual error flag is reset by error management)
IF(1)       LV_RST_DET_DISP = 0
THEN(1)
            LV_ERR_WARM_RST = 1             no debounce, set by error manager
            RST_CLAS_SEC[0] = RST_SEC       copy to actual list entry
            IF(2)           (RST_CLAS_TYP[0] = 01h
                    or      RST_CLAS_TYP[0] = 10h)
            THEN(2)
                    IF(3)           RST_CLAS_SEC[0] = 1
                    THEN(3)
                            ERR_SYM_WARM_RST = SYM_1
                    ELSE(3)
                            ERR_SYM_WARM_RST = SYM_3
                    ENDIF(3)
            ELSEIF(2)       (RST_CLAS_TYP[0] = F0h
                    or      RST_CLAS_TYP[0] = 12h (should not at LV_WARM_RST = 1)
                    or      RST_CLAS_TYP[0] = 13h
                    or      RST_CLAS_TYP[0] = 20h
                    or      RST_CLAS_TYP[0] = 30h)
            THEN(2)
                    ERR_SYM_WARM_RST = SYM_3
            ELSEIF(2)       RST_CLAS_TYP[0] = 40h (unintended software reset)
            THEN(2)
                    ERR_SYM_WARM_RST = SYM_2
            ELSEIF(2)       (RST_CLAS_TYP[0] = 11h (Complement err. at Power-On Reset)
                    or      RST_CLAS_TYP[0] = 41h (Complement error at Software Reset)
            THEN(2)
                    ERR_SYM_WARM_RST = SYM_3
            ELSEIF(2)       RST_CLAS_TYP[0] = 42h (intended software reset)
            THEN(2)
                    IF(3)           (RST_INFO_ADD[0] = 00000000h (no info available)
                            OR      RST_INFO_ADD[0] = 00000400h (SUPRAM)
                            OR      RST_INFO_ADD[0] = 00000401h (NVMY)
                            OR      RST_INFO_ADD[0] = 00000402h (ADC)
```

```
                    OR       RST_INFO_ADD[0] = 00000404h (INJR)
               THEN(3)
                    ERR_SYM_WARM_RST = SYM_0
               ELSEIF(3)     RST_INFO_ADD[0] = 00000300h (SEC_MC)
               THEN(3)
                    ERR_SYM_WARM_RST = SYM_1
               ELSEIF(3)     (RST_INFO_ADD[0] = 00000100h (IMOB)
                    OR       RST_INFO_ADD[0] = 00000200h (REPROG learn)
                    OR       RST_INFO_ADD[0] = 00000201h (REPROG kwp)
               THEN(3)
                    Nothing (no error)
               ELSEIF(3)     RST_ INFO_ADD[0] = 00000500h (VDD_IGK)
               THEN(3)
                    ERR_SYM_WARM_RST = SYM_3
               ELSE(3)
                    ERR_SYM_WARM_RST = SYM_0 (should not happen)
               ENDIF(3)
          ELSEIF(2)     RST_CLAS_TYP[0] = 43h (VDD Error at Software Reset)
          THEN(2)
               ERR_SYM_WARM_RST = SYM_3
          ELSEIF(2)     RST_CLAS_TYP[0] = 44h (TRAP)
          THEN(2)
               ERR_SYM_WARM_RST = SYM_2
          ELSEIF(2)     RST_CLAS_TYP[0] = 45h (NMI)
          THEN(2)
               ERR_SYM_WARM_RST = SYM_2
          ELSEIF(2)     RST_CLAS_TYP[0] = 46h (Error at Reset)
          THEN(2)
               ERR_SYM_WARM_RST = SYM_2
          ELSE(2)
               ERR_SYM_WARM_RST = SYM_3 (should not happen)
          ENDIF(2)
          LV_RST_DET_DISP = 1 (error reporting done)
      LV_END_DIAG_WARM_RST = 1
ENDIF(1)
```

The anti-bounce-algorithm is not used. The sympton is set and reset immediately.

14.2 ECOP – Requirements to infrastructure interface

Export actions:

ACTION_INFR_GetVpTecu (OUT < Vp_tecu_sens **>)**
This action provides the digitized voltage value of the ECU temperature sensor
ACTION_INFR_GetVTecuMc (OUT <V_tecu_mc_sens>)
This action provides the digitized voltage value of the microcontroller temperature sensor
ACTION_INFR_GetBiosAdcStatus (OUT <Bios_adc_status>)
This action provides the digitized status of analogous signals

Description for actions:

ACTION_INFR_GetVpTecu (Vp_tecu_sens**)**

This action provides the digitized voltage value of the ECU temperature sensor in a single value delivered from the infrastructure. The AD conversion is performed autonomously by the infrastructure. When the action is called the gathered information will be provided to the application software level.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Vp_tecu_sens	OUT	0...7FFFH	0...4.99984	0.1526e-03	V
ECU temperature sensor voltage value					

ACTION_INFR_GetVTecuMc (V_tecu_mc_sens**)**

This action provides the digitized voltage value of the microcontroller temperature sensor in a single value delivered from the infrastructure. The AD conversion is performed autonomously by the infrastructure. When the action is called the gathered information will be provided to the application software level.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
V_tecu_mc_sens	OUT	0...7FFFH	0...4.99984	0.1526e-3	V
Microcontroller temperature sensor voltage					

ACTION_INFR_GetBiosAdcStatus (Bios_adc_status**)**

This action provides the digitized status of analogous signals in a single value delivered from the infrastructure. The AD conversion is performed autonomously by the infrastructure. When the action is called the gathered information will be provided to the application software level.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Bios_adc_status	OUT	0...1H	0...1	1	[-]
Status ADC in BIOS					

FUNCTION DESCRIPTION:

General information:

The following action is used for the acquisition of the digitized voltage values of the adressed ECU temperature sensors in two single values. The action delivers the ECU temperature sensors voltages values to the application software level from the standard AD converter queue.

- The AD conversions are performed autonomously by the infrastructure, the returned values are not older than 500 ms (in case of TECU sensor) and 10 ms (in case of the TECU sensor of the microcontroller).
- The voltage values are gathered in the infrastructure until the application reads out the information by calling the action, old values are replaced by new values.

Requirements for ACTION_INFR_GetVpTecu:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Vp_tecu_sens			10 Bit	Vp_tecu_sens is digitized at point in time t_k. For the returned array the maximum allowed time periode $\| t_k - t_i \|$ must less than 500ms.	No comment

Diagnosis: no electrical diagnosis done here

Coincidence requirements: When calling this action, all returned voltages must be within the last 500 ms.

Requirements for ACTION_INFR_GetVTecuMc:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
V_tecu_mc_sens			10 Bit		

Diagnosis: no electrical diagnosis done here

Coincidence requirements: When calling this action, all returned voltages must be within the last 500 ms.

Requirements for ACTION_INFR_GetBiosAdcStatus:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Bios_adc_status			1 Bit		

Diagnosis: no electrical diagnosis done here

Coincidence requirements:

When calling this action, all returned results must be within the last **10 ms**.

14.3 ECU State Detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ECU_STATE	V/O	0H 1H 2H 3H 4H 5H 6H	ENG_STOP RUN_ENG SYN_ENG_IGK_ON SYN_ENG_IGK_OFF PWL ENG_LOCK WAKE_UP	0	[-]
information on ECU state					
FAST_ECU_TRAN_PWL_CTR	V	0...FFFFH	0...655.35	0.01	[s]
fast transition counter for transition from SYN_ENG_IGK_OFF to PWL despite of engine running					

Input data:

LV_INH_PWL_TRAN_ES_EL	ECU_LOCK_REQ	
LV_STOP_ENG	LV_IGK	LV_SYN_ENG
LV_FIRST_VLD_TOOTH	LV_RUN_ENG	LV_CS
T_DLY_RLY_MAIN_DIAG	LV_RLY_MAIN	VB
C_VB_RLY_MAIN_DIAG		

Application conditions:

Recurrence: If LV_ES = 1 → segment-synchronous

If LV_ES = 0 → 10ms

FUNCTION DESCRIPTION:

General information:

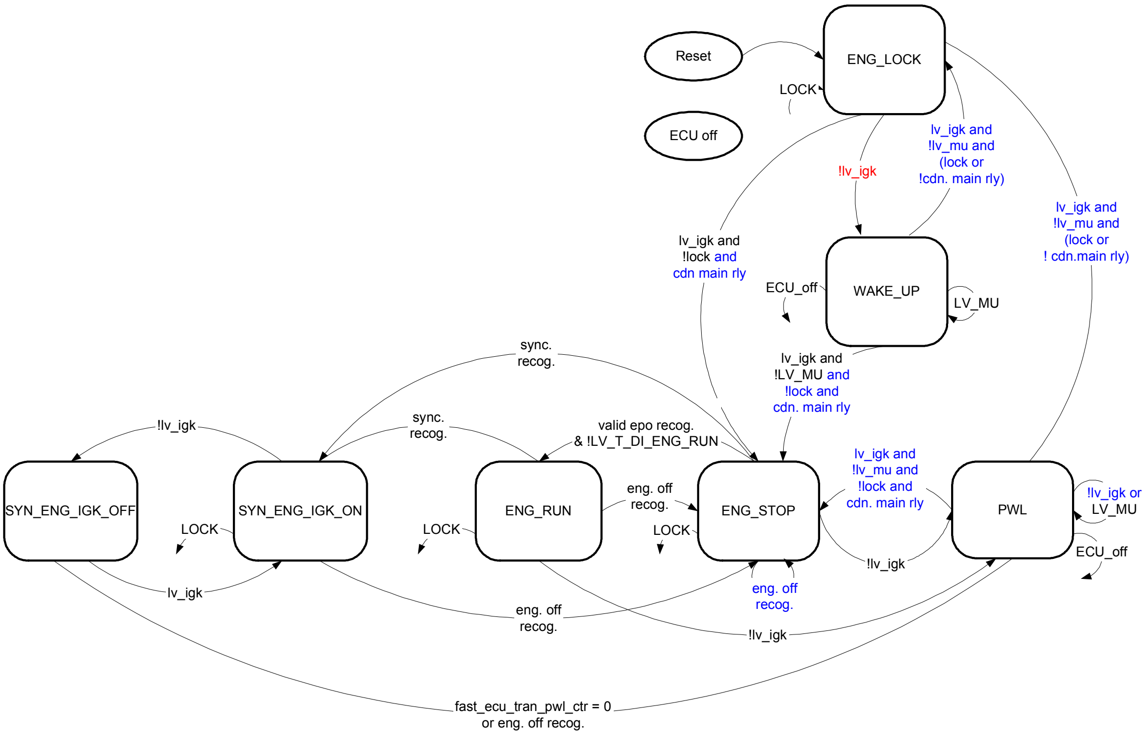
The function ECU-state manages detection of the operating states of the ECU and transitions between them referring to the main inputs key detection and crankshaft synchronization.

The ECU-state offers system events corresponding to those transitions that can be used by other functions (e.g. ignition function that asks for ignition reset on the transition engine running to engine stop).

Those actions are described in the specification chapter of the corresponding functions.

An important value to determine the ECU-state is the value ECU_LOCK_REQ. If it becomes 1, the ECU-state will become ENG_LOCK (exeption is PWL and WAKE_UP).

Signal flow diagram:



Details:
-> LOCK = ecu_lock_req
-> LV_MU = lv_inh_pwl_tran_es_el
-> eng. off recog. = lv_stop_eng
-> sync. recog. = lv_syn_eng
-> valid epo recog. = lv_run_eng and !lv_syn_eng
-> cdn. main rly = [lv_rly_main and (vb >= c_vb_rly_main_diag or t_dly_rly_main_diag = 0)]

Detection of first valid tooth:

If LV_STOP_ENG = 0 **and**

LV_FIRST_VLD_TOOTH 0 -> 1

Then offer an interface for project specific activities on this state transition

Endif

ENGINE LOCK (ENG_LOCK)

This state is dedicated to a special operating state (e.g. engine locked by immobilizer) where all essential outputs are inhibited. An interface for project specific activities on this state ENG_LOCK does exist. The ECU state is initialized with this state after reset.

ENG_LOCK => WAKE_UP

```
If      LV_IGK = 0
Then    ENG_LOCK => WAKE_UP
Endif
```

ENG_LOCK => ENG_STOP

```
If      ECU_LOCK_REQ = 0                and
        LV_IGK = 1                      and
        LV_RLY_MAIN = 1                 and
        (VB >= C_VB_RLY_MAIN_DIAG       or
        T_DLY_RLY_MAIN_DIAG = 0)
Then    offer an interface for project specific activities on this state transition
        ECU_STATE = ENG_STOP
Endif
```

ENGINE STOP (ENG_STOP)

The crankshaft does not turn (no valid crank- or cam-edges are recognized), ignition key is ON and no engine lock condition is set. An interface for project specific activities on this state does exist.

ENG_STOP => PWL

```
If      LV_IGK = 0
Then    offer an interface for project specific activities
        LV_T_DI_ENG_RUN  =  0
        ECU_STATE = PWL
Endif
```

ENG_STOP => ENG_LOCK

```
If      ECU_LOCK_REQ > 0        and
        LV_IGK = 1
Then    offer an interface for project specific activities
        ECU_STATE = ENG_LOCK
Endif
```

ENG_STOP => ENG_STOP

```
If          LV_STOP_ENG = 1
Then        offer an interface for project specific activities
            ECU_STATE = ENG_STOP
            Decrement  timer_disable_engine_run  till 0
            If      timer_disable_engine_run = 0
            Then    LV_T_DI_ENG_RUN  =  0
            Endif
Endif
```

ENG_STOP => ENG_RUN

```
If          LV_RUN_ENG = 1              and
            LV_SYN_ENG = 0              and
            LV_T_DI_ENG_RUN = 0
Then        offer an interface for project specific activities
            ECU_STATE = RUN_ENG
Endif
```

ENG_STOP => SYN_ENG_IGK_ON

```
If          LV_RUN_ENG = 1    and
            LV_SYN_ENG = 1
Then        offer an interface for project specific activities for transition of state RUN_ENG
            offer a second interface for project specific activities on the state transition in
            state SYN_ENG_IGK_ON
            ECU_STATE = SYN_ENG_IGK_ON
Endif
```

RUNNING ENGINE (ENG_RUN)

This state corresponds to a turning engine without synchronization on crankshaft.

An interface for project specific activities on this state does exist.

ENG_RUN => PWL

```
If          LV_IGK = 0
Then        offer an interface for project specific activities on this state transition.
            ECU_STATE = PWL
Endif
```

ENG_RUN => ENGINE LOCK

```
If      ECU_LOCK_REQ > 0
Then    offer an interface for projekt specific activities,
        ECU_STATE = ENG_LOCK
Endif
```

ENG_RUN => ENG_STOP

```
If      LV_STOP_ENG = 1
Then    offer an interface for project specific activities,
        ECU_STATE = ENG_STOP
Endif
```

ENG_RUN => SYN_ENG_IGK_ON

```
If      LV_SYN_ENG = 1
Then    offer an interface for project specific activities
        ECU_STATE = SYN_ENG_IGK_ON
Endif
```

SYNCHRON ENGINE: IGNITION KEY ON (SYN_ENG_IGK_ON)

This state corresponds to a synchronized engine and ignition key on.

SYN_ENG_IGK_ON => ENG_LOCK

```
If      ECU_LOCK_REQ > 0
Then    offer an interface for project specific activities
        ECU_STATE = ENG_LOCK
Endif
```

SYN_ENG_IGK_ON => ENGINE STOP

```
If      LV_STOP_ENG = 1
Then    offer an interface for project specific activities
        If      LV_CS = 1
        Then    start  timer_disable_engine_run  with C_T_DI_ENG_RUN
                LV_T_DI_ENG_RUN  =  1
        Endif
        ECU_STATE = ENG_STOP
Endif
```

SYN_ENG_IGK_ON => SYN_ENG_IGK_OFF

```
If      LV_IGK = 0
Then    offer an interface for project specific activities,
        ECU_STATE = SYN_ENG_IGK_OFF
Endif
```

Fast transition counter FAST_ECU_TRAN_PWL_CTR

The transition SYN_ENG_IGK_OFF => PWL FAST_ECU_TRAN_PWL_CTR is done if engine is stopped or if LV_IGK = 0 and fast transition counter FAST_ECU_TRAN_PWL_CTR = 0. The fast transition counter is started at transition from SYN_ENG_IGK_ON => SYN_ENG_IGK_OFF and is decremented at every recurrence by 10 ms.

```
If      transition ECU_STATE = SYN_ENG_IGK_OFF => ECU_STATE = PWL
Then    FAST_ECU_TRAN_PWL_CTR = FAST_ECU_TRAN_PWL_CTR - 10 ms
Else    FAST_ECU_TRAN_PWL_CTR = NC_FAST_TRAN_PWL
Endif
```

SYNCHRON ENGINE: IGNITION KEY OFF (SYN_ENG_IGK_OFF)

This state corresponds to a synchronized engine and ignition key off.

SYN_ENG_IGK_OFF => SYN_ENG_IGK_ON

```
If      LV_IGK = 1
Then    offer an interface for project specific activities
        ECU_STATE = SYN_ENG_IGK_ON
Endif
```

SYN_ENG_IGK_OFF => PWL

```
If      LV_STOP_ENG = 1                    or
        FAST_ECU_TRAN_PWL_CTR = 0
Then    offer an interface for project specific activities
        ECU_STATE = PWL
Endif
```

POWER LATCH (PWL)

Non volatile data are stored and ECU shutdown is prepared. An interface for project specific activities on this state does exist.

PWL => ENG_LOCK

```
If      LV_IGK = 1                      and
        LV_INH_PWL_TRAN_ES_EL = 0       and
        (ECU_LOCK_REQ > 0               or
        LV_RLY_MAIN = 0                 or
        (VB < C_VB_RLY_MAIN_DIAG        and
        T_DLY_RLY_MAIN_DIAG > 0))
Then    ECU_STATE = ENG_LOCK
Endif
```

PWL => ENG_STOP

```
If      LV_IGK = 1                      and
        LV_INH_PWL_TRAN_ES_EL = 0       and
        ECU_LOCK_REQ = 0                and
        LV_RLY_MAIN = 1                 and
        (VB >= C_VB_RLY_MAIN_DIAG       or
        T_DLY_RLY_MAIN_DIAG = 0)
Then    ECU_STATE = ENG_STOP
Endif
```

PWL => PWL

```
If      LV_ IGK=0                       or
        LV_INH_PWL_TRAN_ES_EL = 1
Then    The condition LV_INH_PWL_TRAN_ES_EL = 1 indicates that the monitoring
        unit disabled the powerstages of MTC and injection outputs. IGK ON recognition
        is not accepted without reset.
        ECU_STATE = PWL
Endif
```

WAKE UP MODE (WAKE_UP)

The Wake UP Mode can only be reached after ECU_Reset (Shut Down Mode).
This mode indicates that the LV_WAKE_UP was set and the Ignition Key is off.

WAKE_UP => ENG_LOCK

```
If      LV_IGK  = 1                  and
        LV_INH_PWL_TRAN_ES_EL = 0    and
        (ECU_LOCK_REQ > 0            or
        LV_RLY_MAIN = 0              or
        (VB < C_VB_RLY_MAIN_DIAG     and
        T_DLY_RLY_MAIN_DIAG > 0))
Then    ECU_STATE = ENG_LOCK
Endif
```

WAKE_UP => ENG_STOP

```
If      LV_IGK = 1                   and
        LV_INH_PWL_TRAN_ES_EL = 0    and
        ECU_LOCK_REQ = 0             and
        LV_RLY_MAIN = 1              and
        (VB >= C_VB_RLY_MAIN_DIAG    or
        T_DLY_RLY_MAIN_DIAG = 0)
Then    ECU_STATE = ENG_STOP
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DI_ENG_RUN	1	0...FFFFH	0...655.35	0.01	[s]
Time of disabled ENG_RUN after engine stall during drive off					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_FAST_TRAN_PWL	1	0...FFFFH	0...655.35	0.01	[s]
maximum delay for transition from SYN_ENG_IGK_OFF to PWL despite engine running					

14.4 ECU Temparature

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TECU	O/V	0...FEH	-48...142.5	0.75	°C
ECU temperature					
CTR_STC_TECU_1	O/V/S	0...FFFFFFFFH	0...4.29497E+9	1	-
statistic counter 1 for TECU monitoring					
CTR_STC_TECU_2	O/V/S	0...FFFFFFFFH	0...4.29497E+9	1	-
statistic counter 2 for TECU monitoring					
CTR_STC_TECU_3	O/V/S	0...FFFFFFFFH	0...4.29497E+9	1	-
statistic counter 3 for TECU monitoring					
CTR_STC_TECU_4	O/V/S	0...FFFFFFFFH	0...4.29497E+9	1	-
statistic counter 4 for TECU monitoring					
CTR_STC_TECU_5	O/V/S	0...FFFFFFFFH	0...4.29497E+9	1	-
statistic counter 5 for TECU monitoring					
CTR_STC_TECU_6	O/V/S	0...FFFFFFFFH	0...4.29497E+9	1	-
statistic counter 6 for TECU monitoring					
CTR_STC_TECU_7	O/V/S	0...FFFFFFFFH	0...4.29497E+9	1	-
statistic counter 7 for TECU monitoring					
CTR_STC_TECU_8	O/V/S	0...FFFFFFFFH	0...4.29497E+9	1	-
statistic counter 8 for TECU monitoring					
VP_TECU	O/V	0...7FFFH	0...4.99984741	1.52588E-4	V
Voltage of ECU-temperature-sensor					
LV_V_TECU_VLD	O/V	0...1H	0...1	1	-
Flag indicating that a valid VP_TECU value is available at MU					
TECU_MC	O/V	0...FEH	-48...142.5	0.75	°C
Microcontroller temperature					
V_TECU_MC	O/V	0...7FFFH	0...4.99984741	1.52588E-4	V
Voltage of microcontroller temperature sensor					

Input data:

LV_ERR_TECU	LV_ES	LV_IGK	TRT

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TECU_SUB	1	0...FEH	-48...142.5	0.75	°C
TECU substitute value					
C_TECU_THD_1	1	0...FEH	-48...142.5	0.75	°C
ETCU threshold 1 (init. Value: 40°C)					
C_TECU_THD_2	1	0...FEH	-48...142.5	0.75	°C
ETCU threshold 2 (init. Value: 50°C)					
C_TECU_THD_3	1	0...FEH	-48...142.5	0.75	°C
ETCU threshold 3 (init. Value: 60°C)					
C_TECU_THD_4	1	0...FEH	-48...142.5	0.75	°C
ETCU threshold 4 (init. Value: 70°C)					
C_TECU_THD_5	1	0...FEH	-48...142.5	0.75	°C
ETCU threshold 5 (init. Value: 85°C)					
C_TECU_THD_6	1	0...FEH	-48...142.5	0.75	°C
ETCU threshold 6 (init. Value: 100°C)					
C_TECU_THD_7	1	0...FEH	-48...142.5	0.75	°C
ETCU threshold 7 (init. Value: 115°C)					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TRT_DIF_TECU	1	0...FFFFFFFFH	0...1.19305E+5	2.77778E-5	h
TRT difference for ETCU statistic (init. value: 0,08333h)					
LC_TECU_SWI	1	0...1H	0...1	1	-
Selection of TECU source					
IP_TECU	16	0...FEH	-48...142.5	0.75	°C
LDP_VP_TECU__TECU	16	0...7FFFH	0...4.99984	1.526E-4	[V]
NTC characteristic					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_TECU_MC_OFS	1	0...7FFFH	0...1.02397E+3	0.03125	°C
Offset of conversion from voltage to temperature of microcontroller					
NC_TECU_MC_SLOP	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Slope of conversion from voltage to temperature of microcontroller					

Import actions:

ACTION_INFR_GetVTecuMc(OUT <V_TECU_MC>)
This action provides the digitized voltage value of the microcontroller temperature sensor
ACTION_INFR_GetVpTecu(OUT <VP_TECU>)
This action provides the digitized voltage value of the ECU temperature sensor

14.4.1 ECU temperature TECU:

The ECU temperature is measured through an A/D converter. The software then is generating a modified sensor signal (VP_TECU), which is requested by the application software for further execution.

The conversion into physical values is made through the non-linear characteristic IP_TECU. In case of a present fault of the TECU-sensor a substitute value will be used.

By request (LC_TECU_SWI) the source for the TECU calculation can switch between IP_TECU(VP_TECU) and TECU_MC.

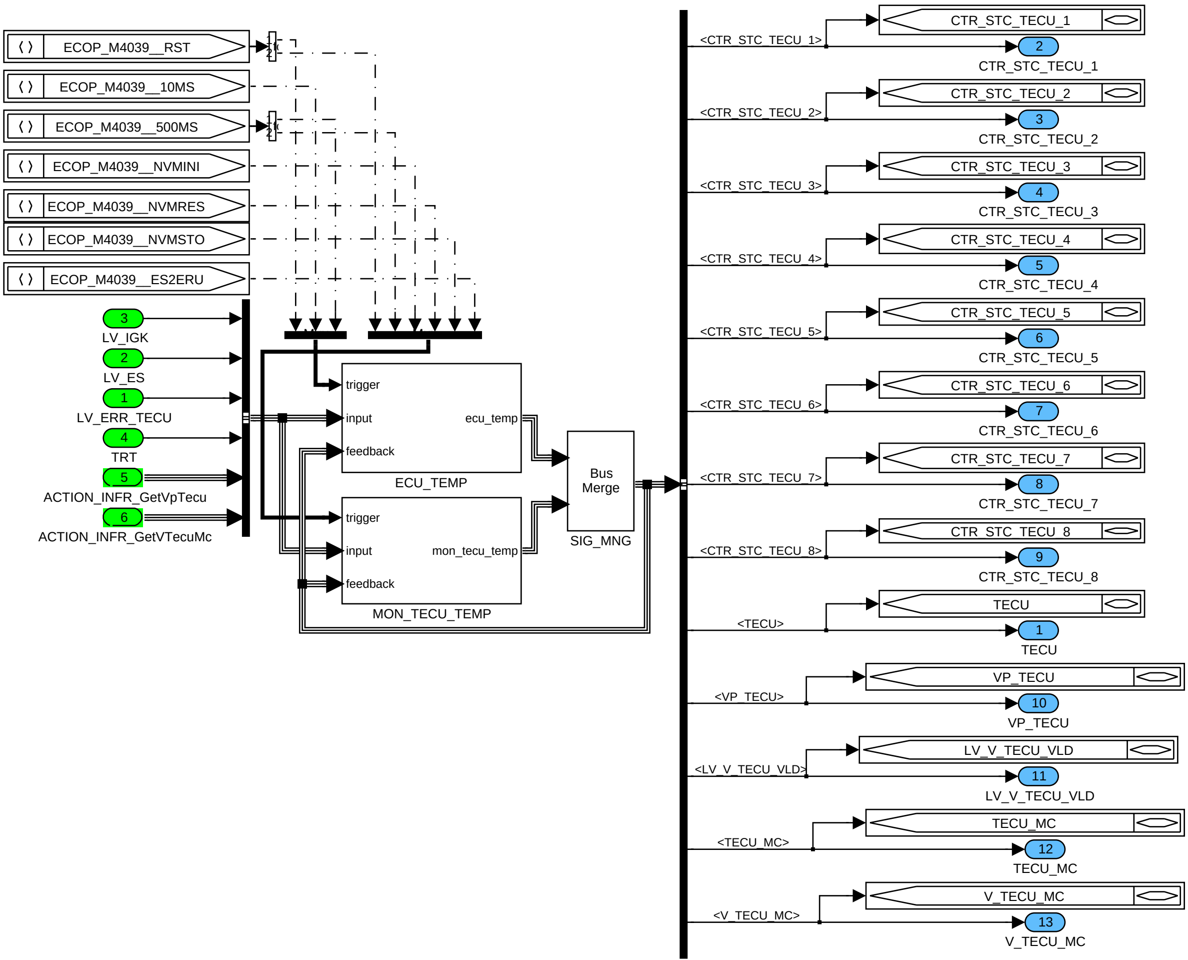
Configuration data

Here are listed the configuration data:

Data	**Value**
NC_TECU_MC_SLOP	190 °C/V *
NC_TECU_MC_OFS	231.9 °C

*Note: For SW calculation purposes NC_TECU_MC_SLOP = 0.028992584

Function Description



SDA_SRS / SDA 4.0 05−Dec−2005

Figure 1 ECOP_M4039

14.4.1.1 ECU Temperature Calculations

Determination of TECU_MES in case of an/no TECU error

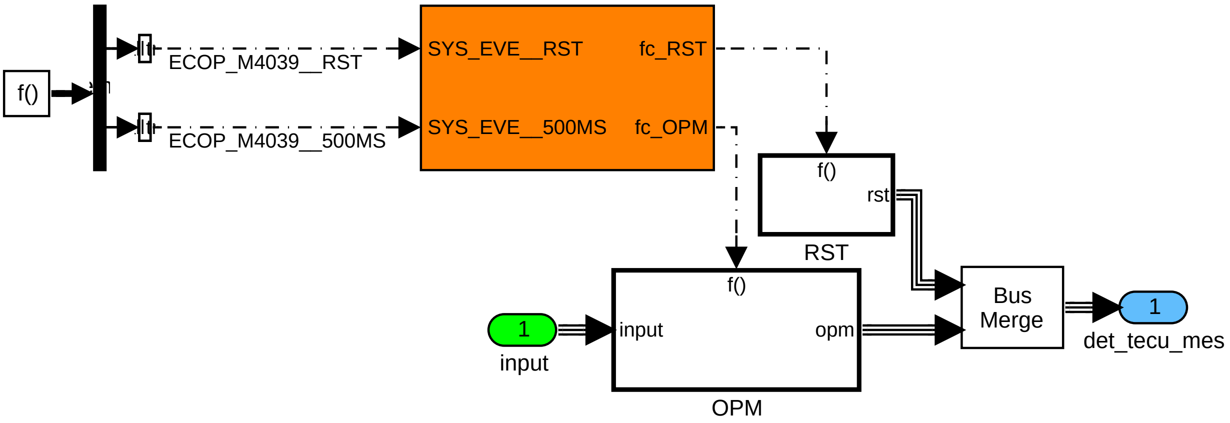


Figure 2 ECOP_M4039/ ECU_TEMP/ DET_TECU_MES

Application Condition

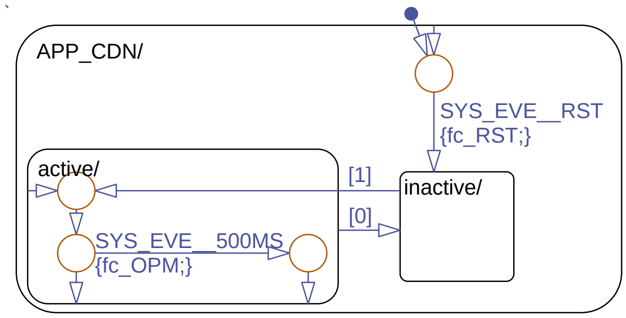


Figure 3 ECOP_M4039/ ECU_TEMP/ DET_TECU_MES/ CHAPTER_1/ Chart

Initialisation at "reset" event

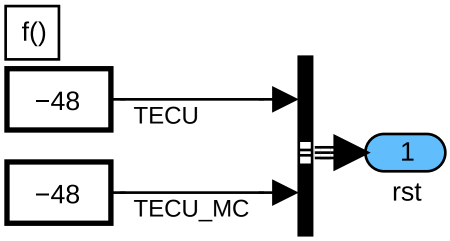


Figure 4 ECOP_M4039/ ECU_TEMP/ DET_TECU_MES/ RST

Formula Section

Calculation V_TECU_MC , VP_TECU, TECU and TECU_MC

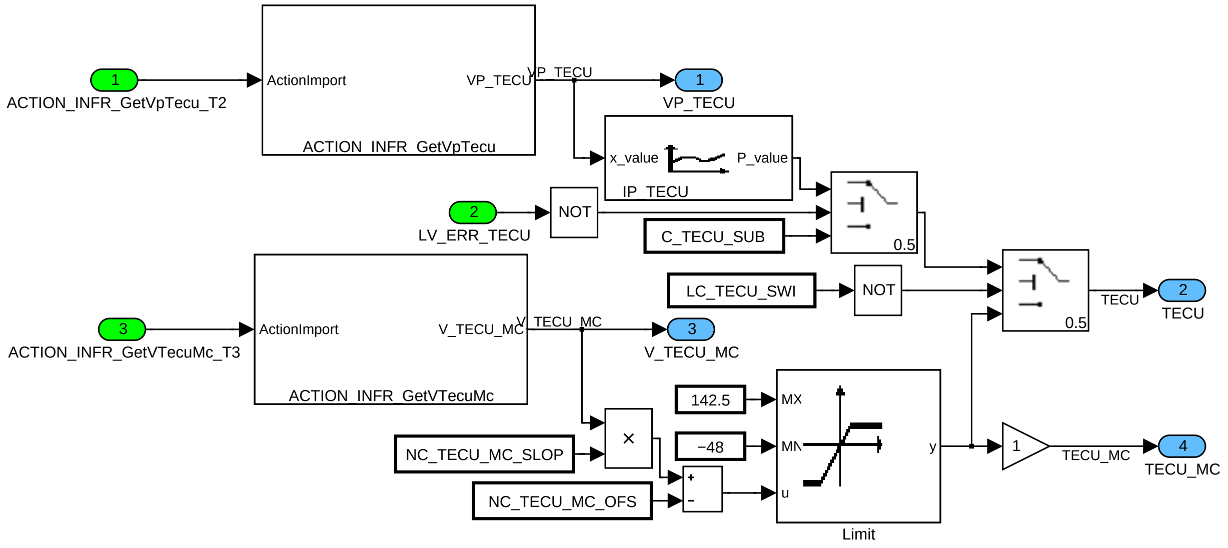


Figure 5 ECOP_M4039/ ECU_TEMP/ DET_TECU_MES/ OPM/ CLC

Detection of valid value of V_TECU (as an interface to VVL-functionalites)

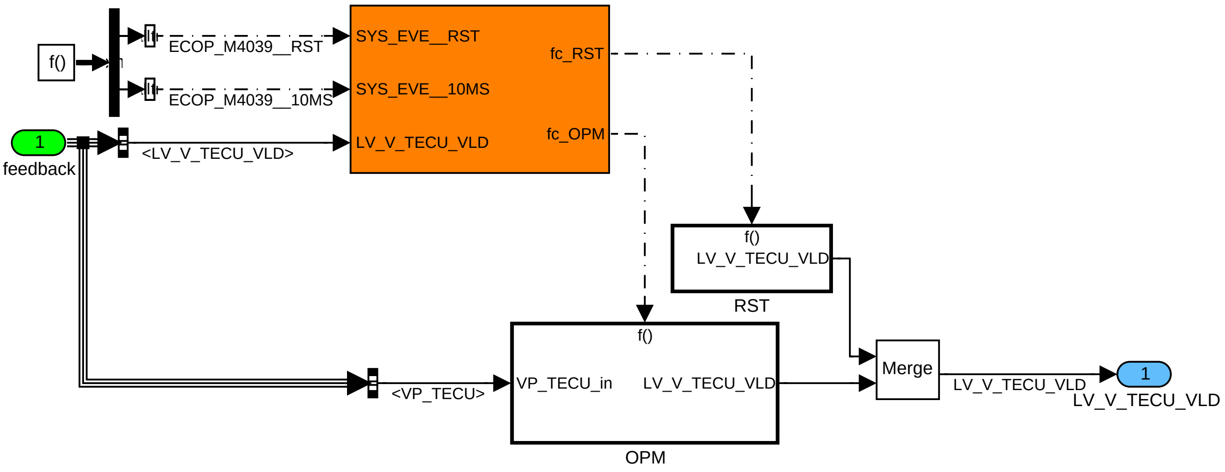


Figure 6 ECOP_M4039/ ECU_TEMP/ DET_VLD_V_TECU

Application condition

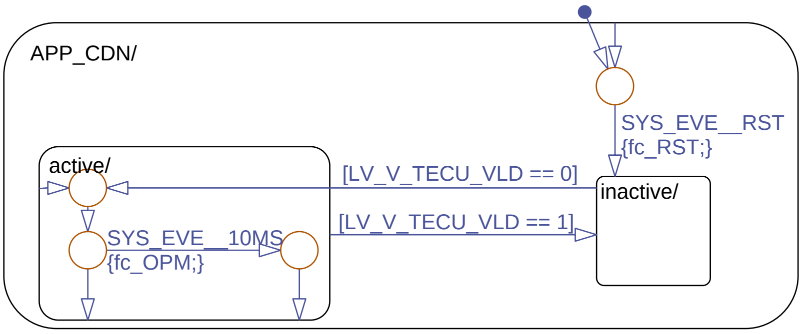


Figure 7 ECOP_M4039/ ECU_TEMP/ DET_VLD_V_TECU/ CHAPTER_2/ Chart

Initialisation at "reset" event

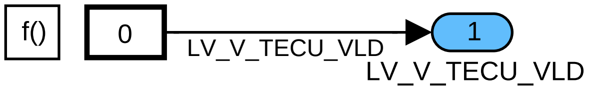


Figure 8 ECOP_M4039/ ECU_TEMP/ DET_VLD_V_TECU/ RST

Formula Section : Calculation of LV_V_TECU_VLD

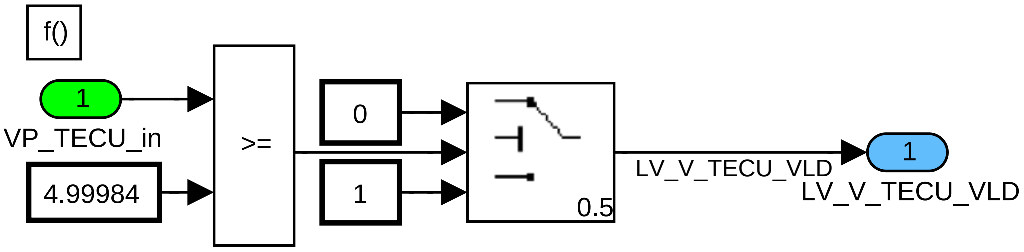


Figure 9 ECOP_M4039/ ECU_TEMP/ DET_VLD_V_TECU/ OPM

14.4.1.2 Monitoring of the ECU – Temperature

This function should provide a statistical statement, how often the ECU temperature is in a defined temperature band. After a calibratable time the ECU temperature will be checked in which temperature band is the ECU. Then, the corresponding counter is incremented.

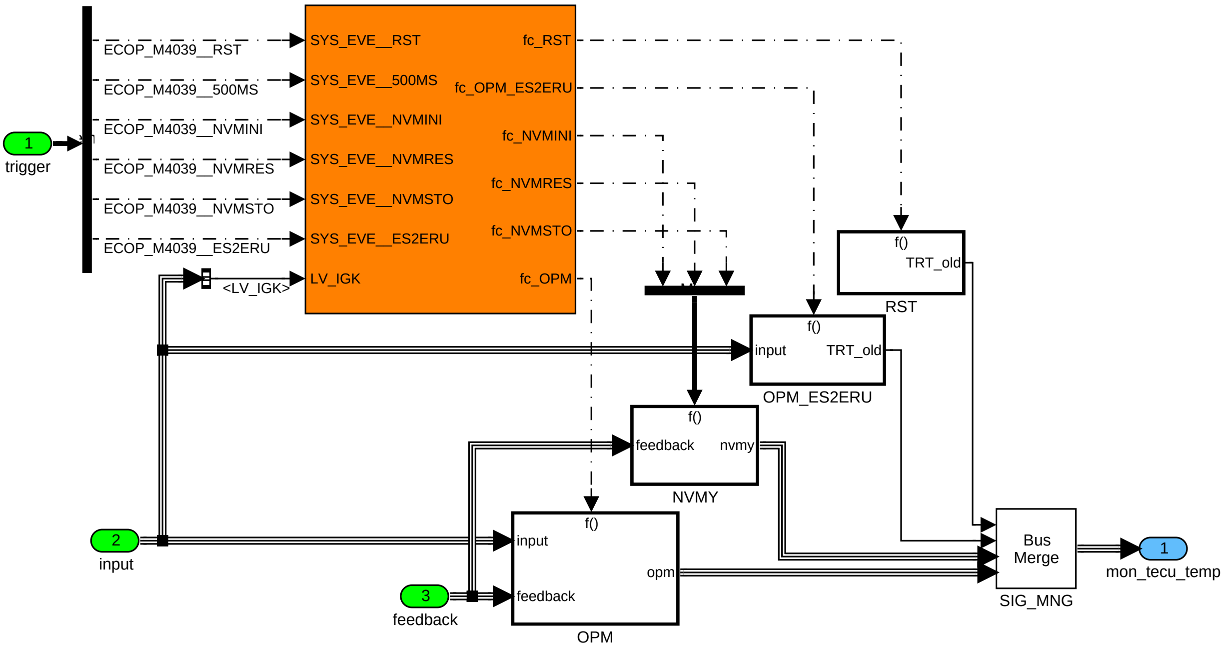


Figure 10 ECOP_M4039/ MON_TECU_TEMP

Application condition

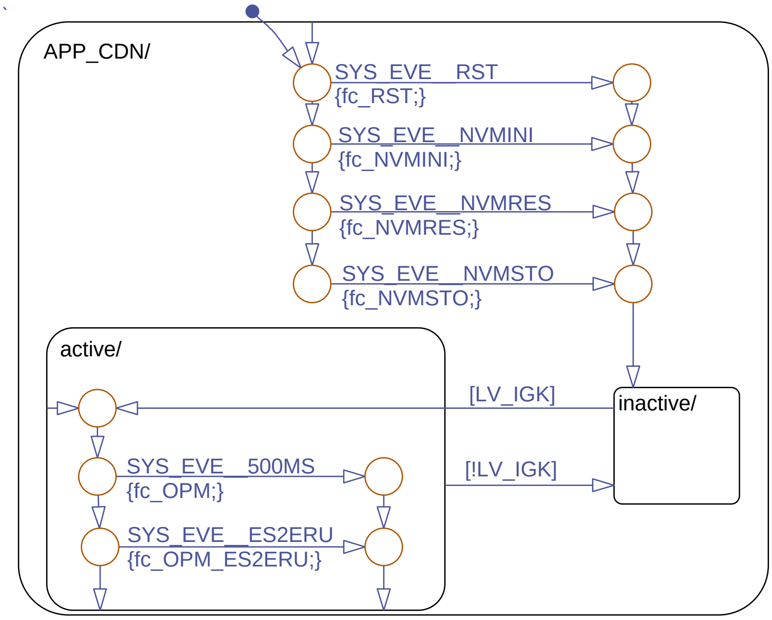


Figure 11 ECOP_M4039/ MON_TECU_TEMP/ ECOP_3/ Chart

Initialisation of TRT_old

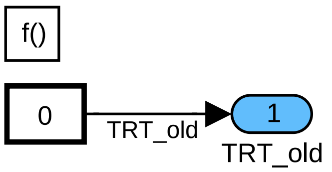


Figure 12 ECOP_M4039/ MON_TECU_TEMP/ RST

Initialisation at system event ES2ERU

Assignment of TRT to TRT_old at system event ES2ERU

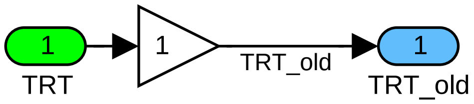


Figure 13 ECOP_M4039/ MON_TECU_TEMP/ OPM_ES2ERU/ INI

Initialisations at system events related to NVMY

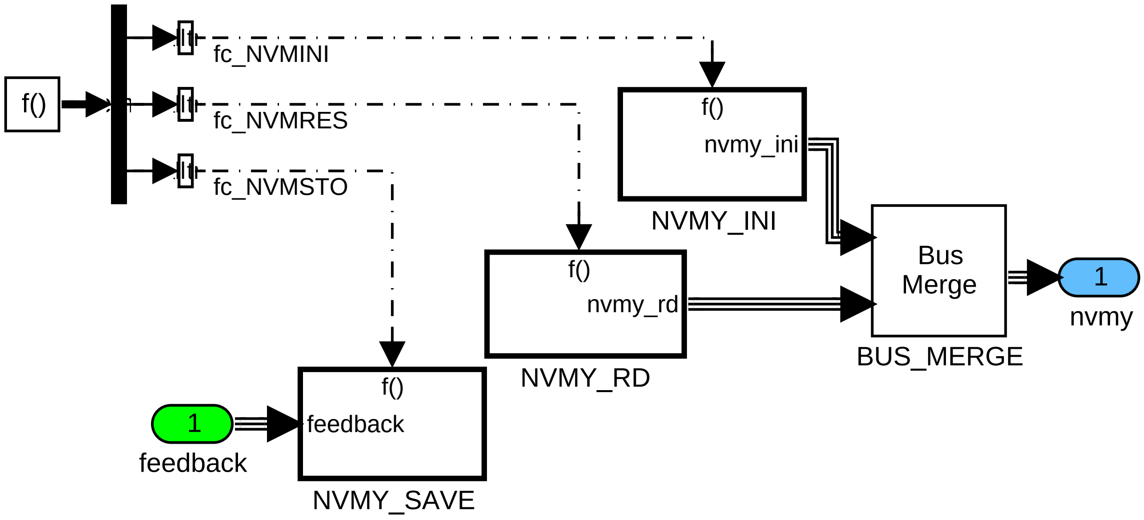


Figure 14 ECOP_M4039/ MON_TECU_TEMP/ NVMY

Formula Section

Condition check for the counter calculations

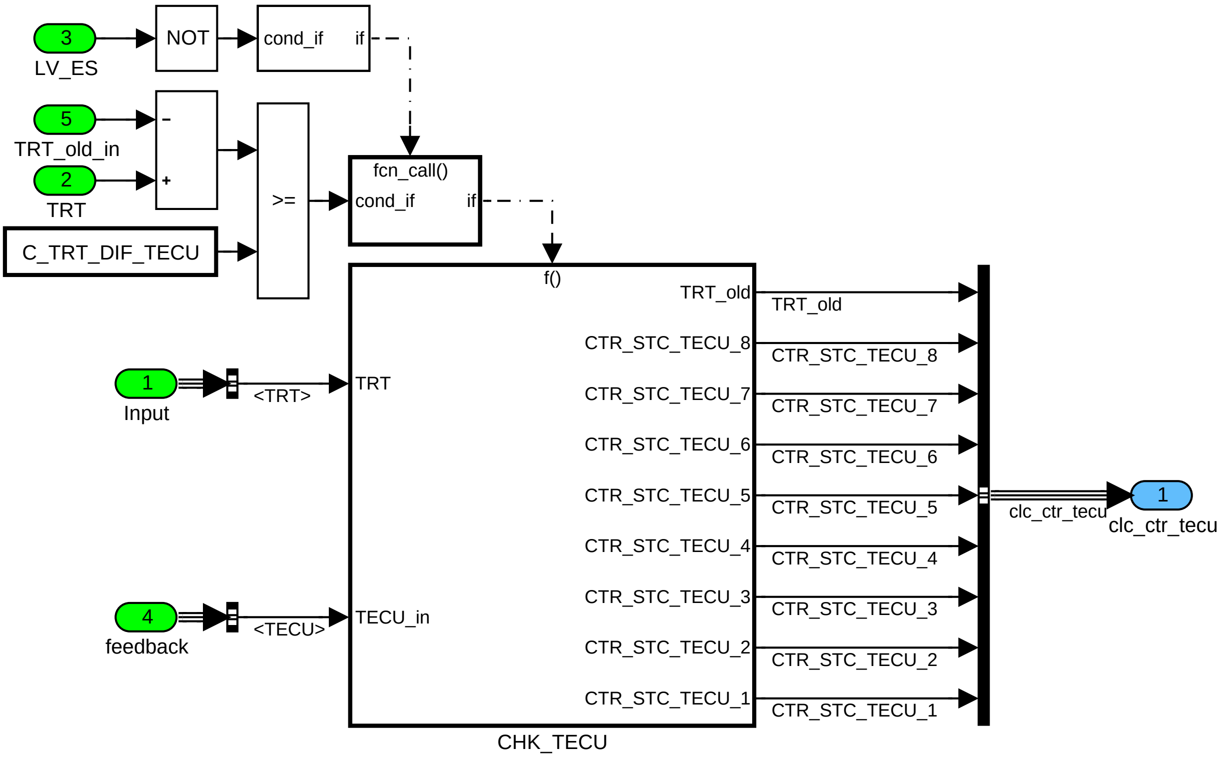


Figure 15 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU

Calculation of statistical counters

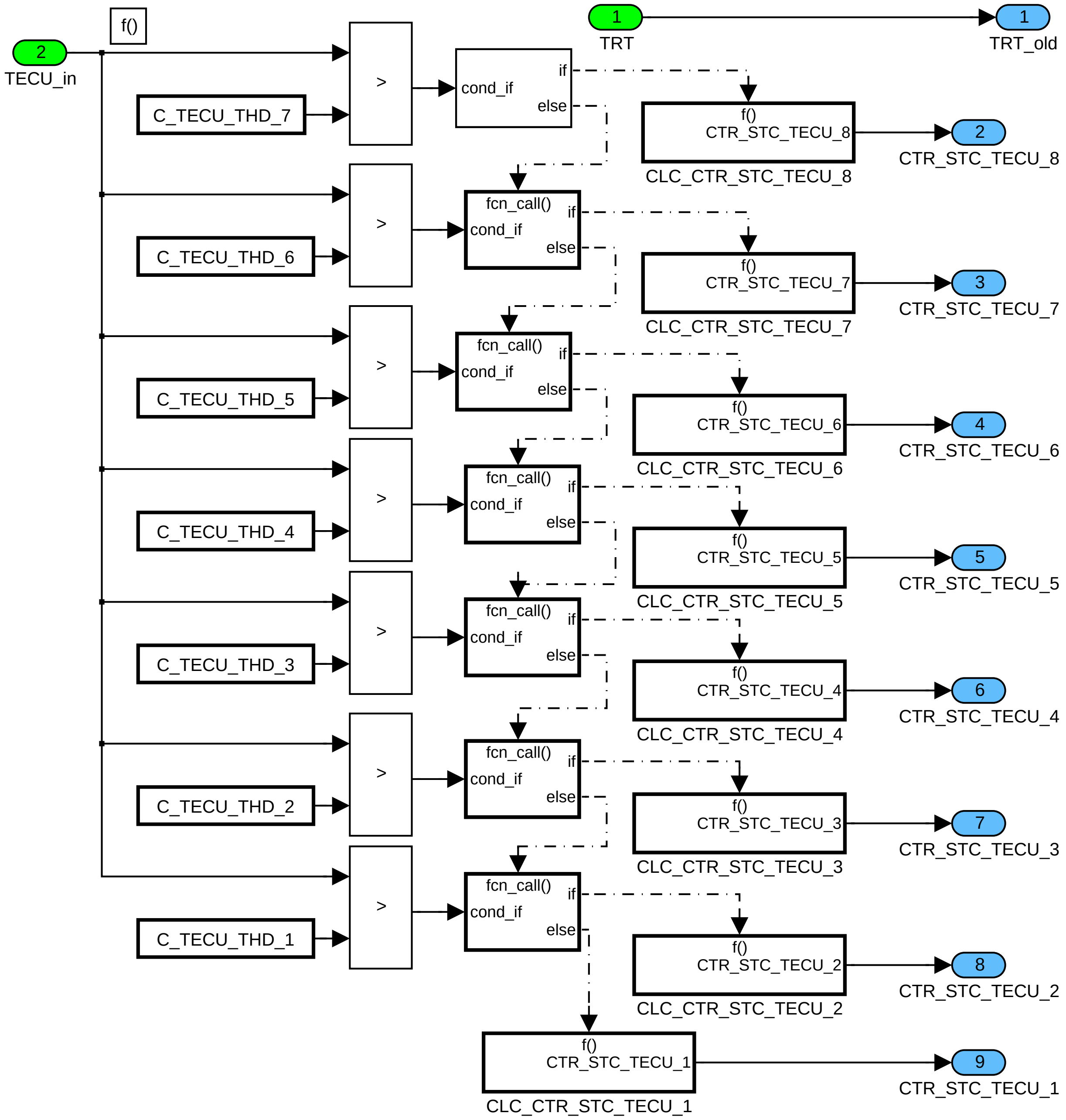


Figure 16 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU

Calculation of statistical counter 8

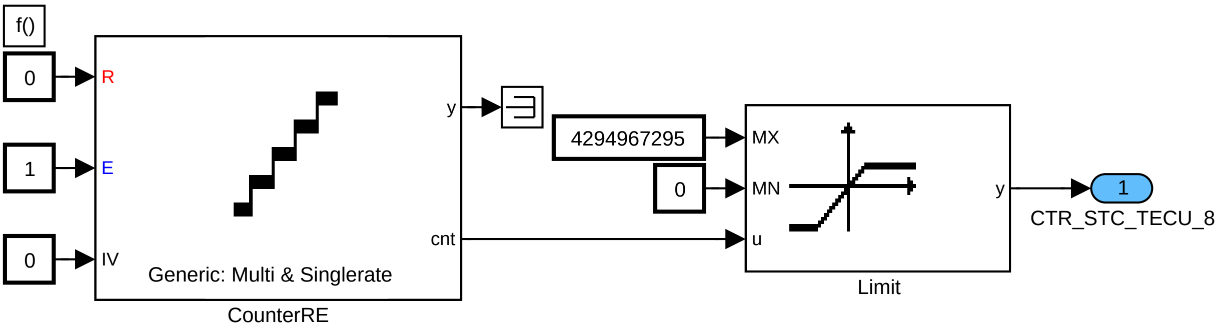


Figure 17 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU/ CLC_CTR_STC_TECU_8

Calculation of statistical counter 7

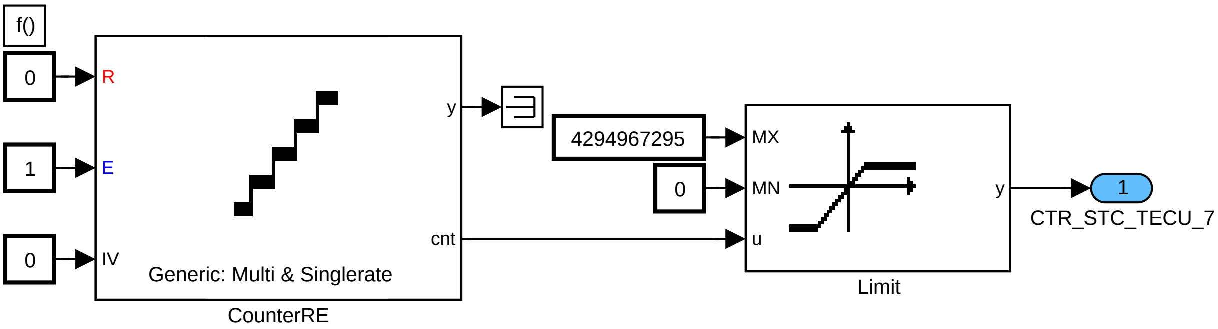


Figure 18 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU/ CLC_CTR_STC_TECU_7

Calculation of statistical counter 6

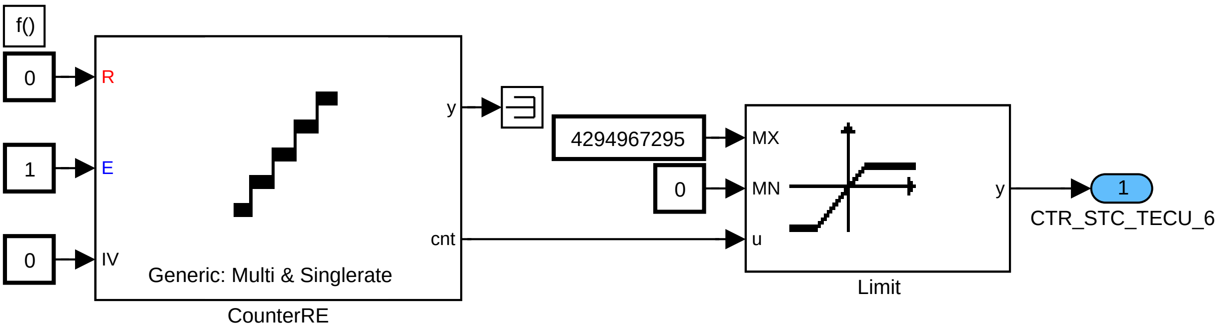


Figure 19 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU/ CLC_CTR_STC_TECU_6

Calculation of statistical counter 5

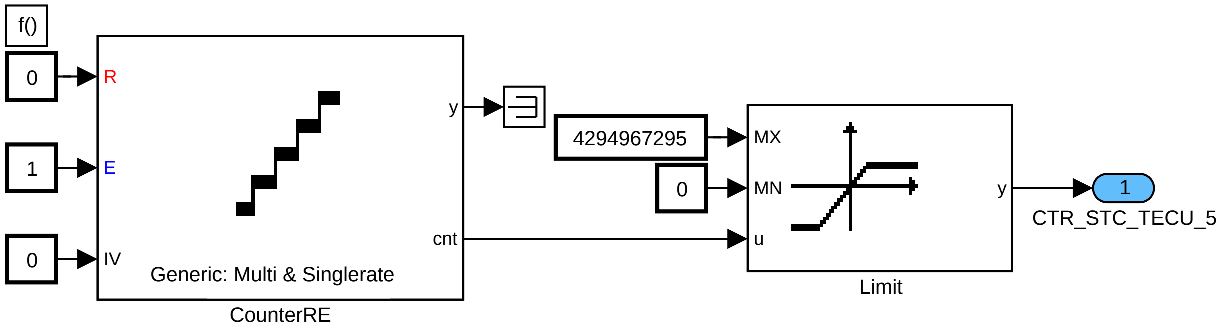


Figure 20 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU/ CLC_CTR_STC_TECU_5

Calculation of statistical counter 4

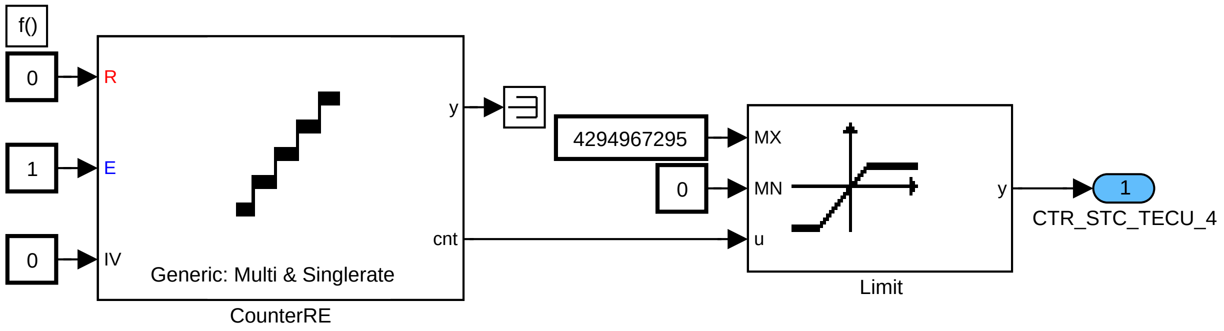


Figure 21 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU/ CLC_CTR_STC_TECU_4

Calculation of statistical counter 3

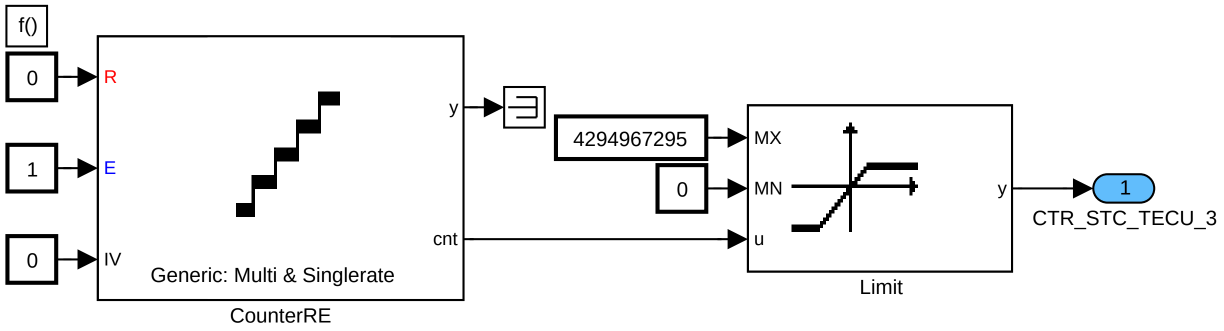


Figure 22 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU/ CLC_CTR_STC_TECU_3

Calculation of statistical counter 2

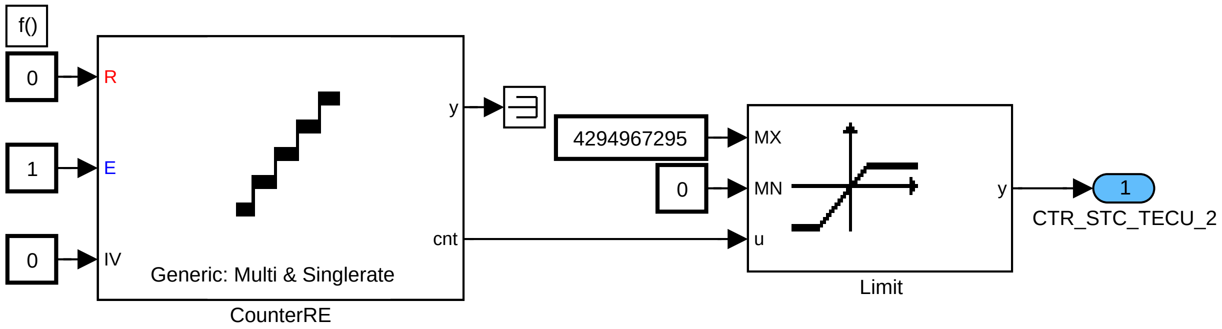


Figure 23 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU/ CLC_CTR_STC_TECU_2

Calculation of statistical counter 1

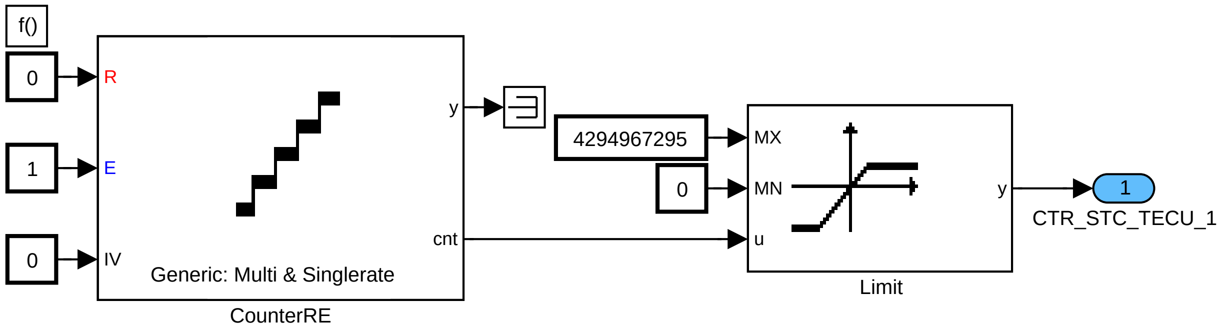


Figure 24 ECOP_M4039/ MON_TECU_TEMP/ OPM/ CLC_CTR_TECU/ CHK_TECU/ CLC_CTR_STC_TECU_1

14.5 ECU temperature sensor diagnosis (LV_ERR_TECU)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_TECU	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom TECU diagnosis					
LV_CDN_DIAG_TECU	O/V	0...1H	0...1	1	-
Diagnosis condition TECU diagnosis					
LV_END_DIAG_TECU	O/V	0...1H	0...1	1	-
End of diagnosis TECU diagnosis					
LV_ERR_TECU	O/V	0...1H	0...1	1	-
TECU error flag					

Input data:

LV_IGK	TIA	T_AST_DIAG	VP_TECU
NC_IDX_DIAG_TECU			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TECU	1	0...FFH	0...255	1	-
Anti-bounce increment					
C_ABC_MAX_TECU	1	1...FFH	1...255	1	-
Anti-bounce maximum					
C_TIA_MAX_TECU_DIAG	1	0...FEH	-48...142.5	0.75	°C
TIA threshold - TECU diagnosis					
C_T_DLY_TECU_DIAG	1	0...FFFFH	0...3.27675E+4	0.5	s
Delay time for diagnosis activation					
C_VP_TECU_MAX_DIAG	1	0...7FFFH	0...4.99984741	1.52588E-4	V
TECU_MES threshold for short to VB or line break - TECU diagnosis					
C_VP_TECU_MIN_DIAG	1	0...7FFFH	0...4.99984741	1.52588E-4	V
TECU_MES threshold for short to ground - TECU diagnosis					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetErrSym(IN <IDX_DIAG>, OUT <ERR_SYM>)
Action that returns the symptom of the failure
ACTION_ERRM_GetLvCdnDiag(IN <IDX_DIAG>, OUT <LV_CDN_DIAG>)
Action that returns the diagnostic condition
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the status of the failure availability
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

14.5.1 FUNCTION DESCRIPTION:

The purpose of the diagnosis shall be to detect electrical faults as defined by OBD I requirements. The input signal is analog from a NTC.

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 |  |  |  └─> Short circuit to Vbatt or open load      (= SYM_0)
 |  |  └───> Short circuit to GND                       (= SYM_1)
 |  └─────> -                                           (= SYM_2)
 └───────> -                                            (= SYM_3)
```

Application Condition

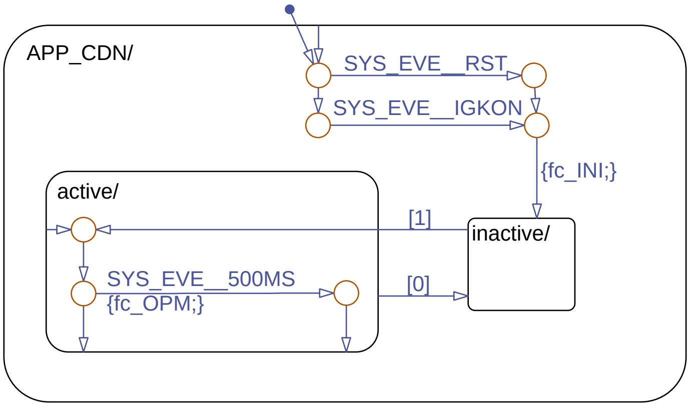


Figure 25 ECOP_MA0BI/ APP_CDN/ Chart

Function Description

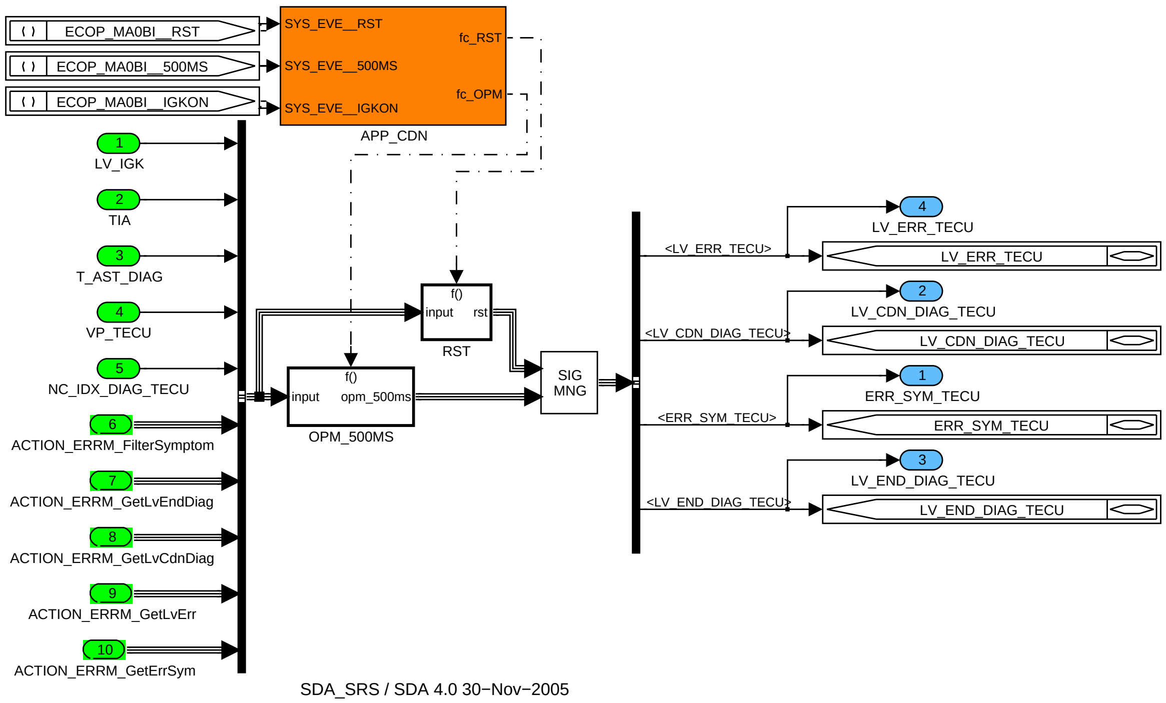


Figure 26 ECOP_MA0BI

14.5.1.1 Initilisation

According ABC configuration **STD_INI**

ABC datas are initialized at transition LV_IGK 0->1)

Initialisation

Initialisation at system event RESET and IGKON.

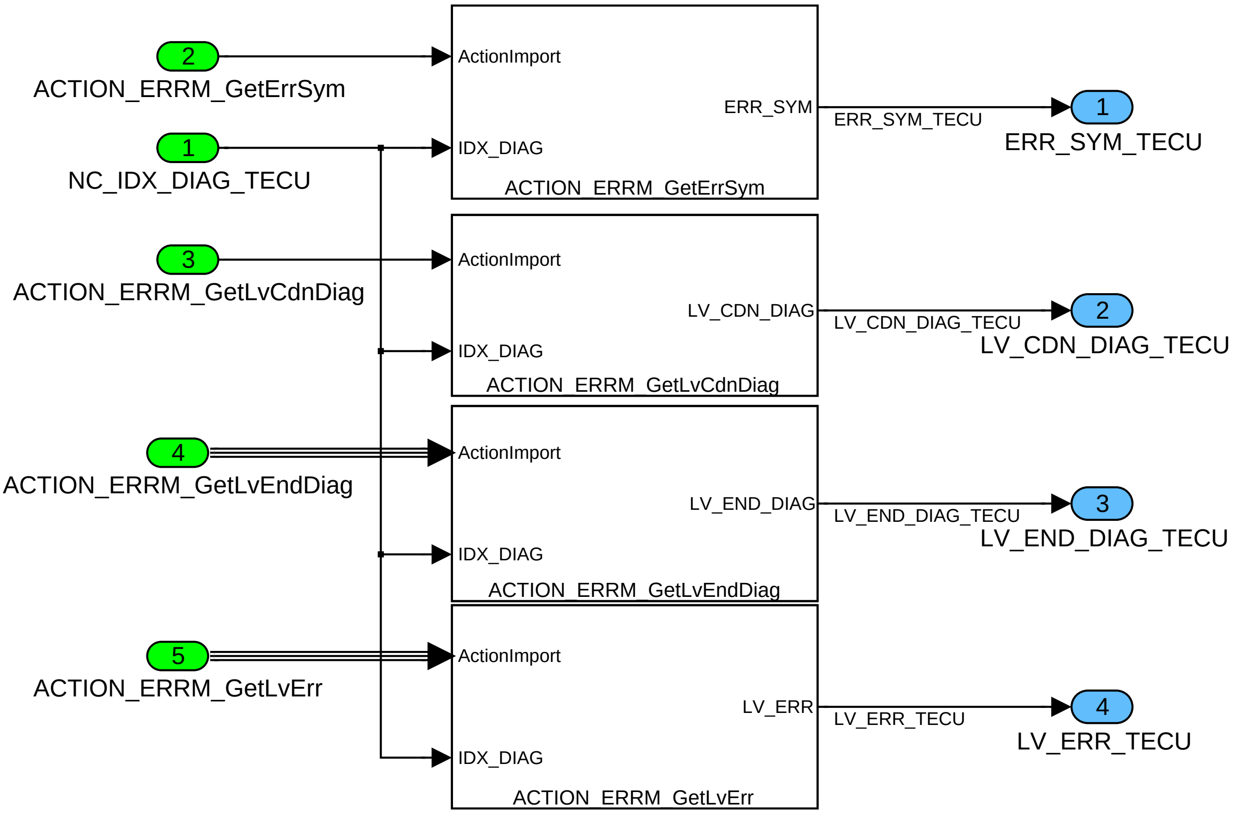


Figure 27 ECOP_MA0BI/ RST/ CLC

14.5.1.2 Formula Section

Activation Condition Check

If LV_IGK =1 then the error symptoms are calculated and if false Diagnosis bit is set to 0.

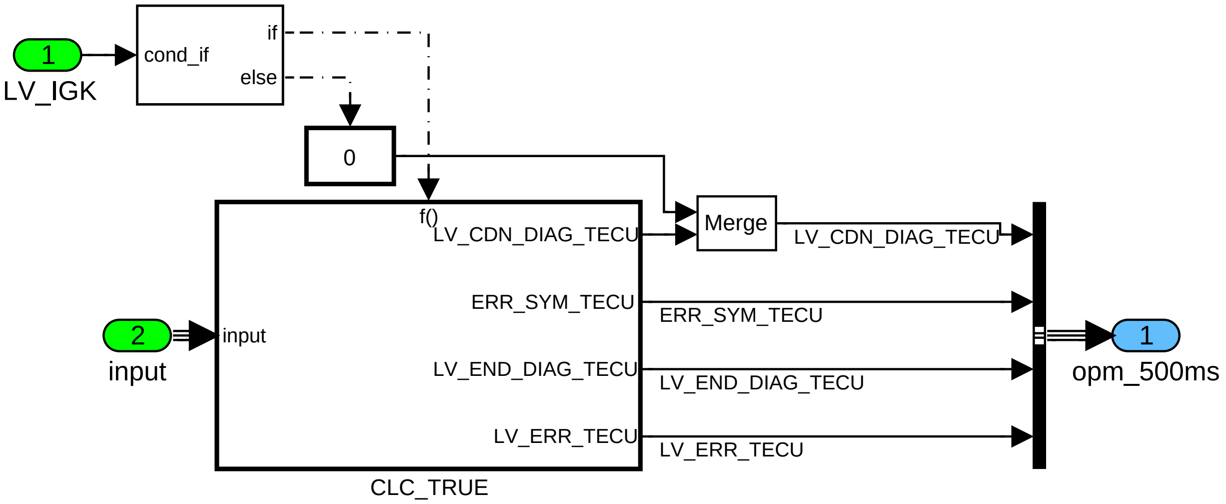


Figure 28 ECOP_MA0BI/ OPM_500MS/ CLC

Calculation of Error Management Variables

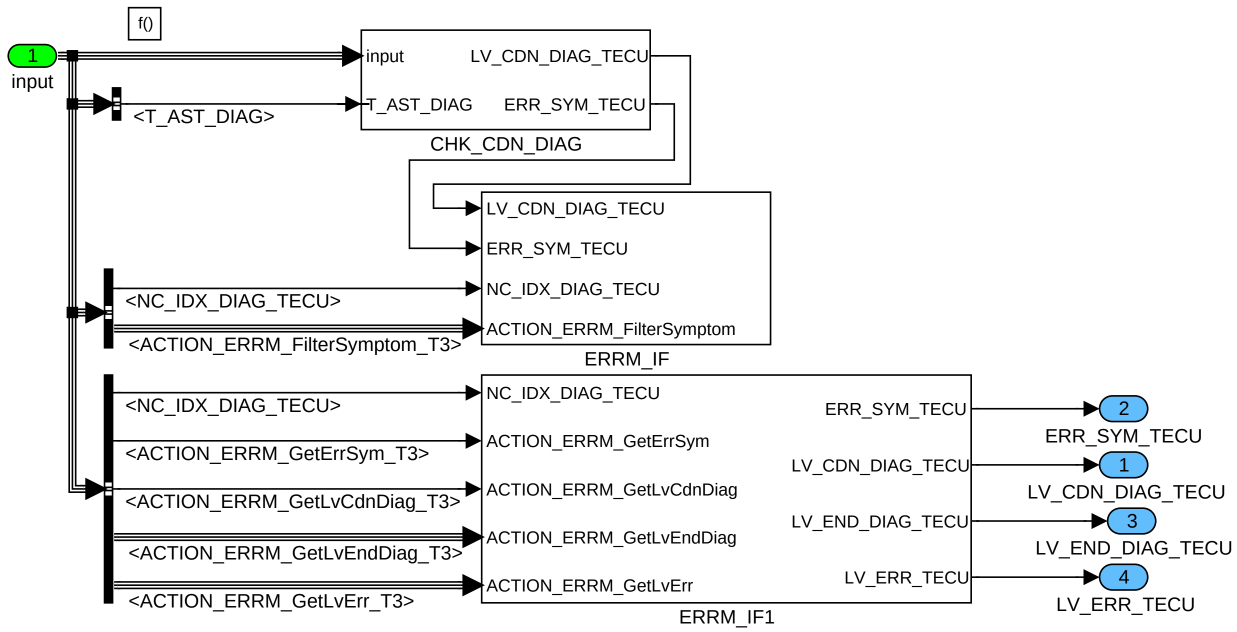


Figure 29 ECOP_MA0BI/ OPM_500MS/ CLC/ CLC_TRUE

Check Condition

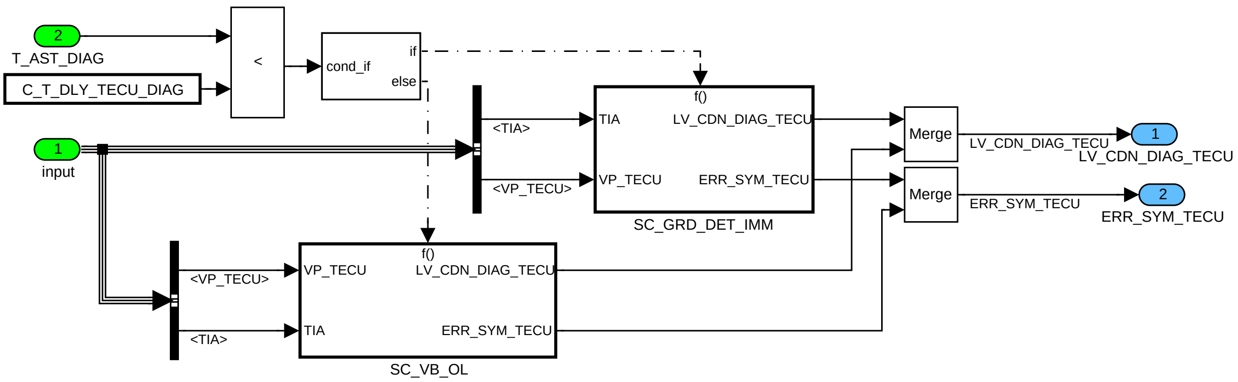


Figure 30 ECOP_MA0BI/ OPM_500MS/ CLC/ CLC_TRUE/ CHK_CDN_DIAG

Short circuit to ground can be detected immediately:

ERR_SYM and LV_CDN are calculated in this subsystem

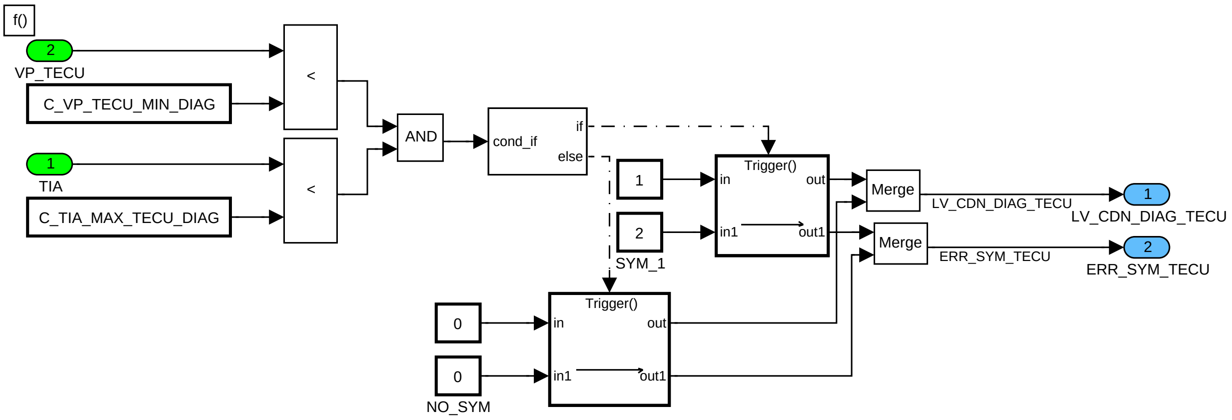


Figure 31 ECOP_MA0BI/ OPM_500MS/ CLC/ CLC_TRUE/ CHK_CDN_DIAG/ SC_GRD_DET_IMM

Short circuit to VBatt or open load:

ERR_SYM and LV_CDN are calculated in this subsystem

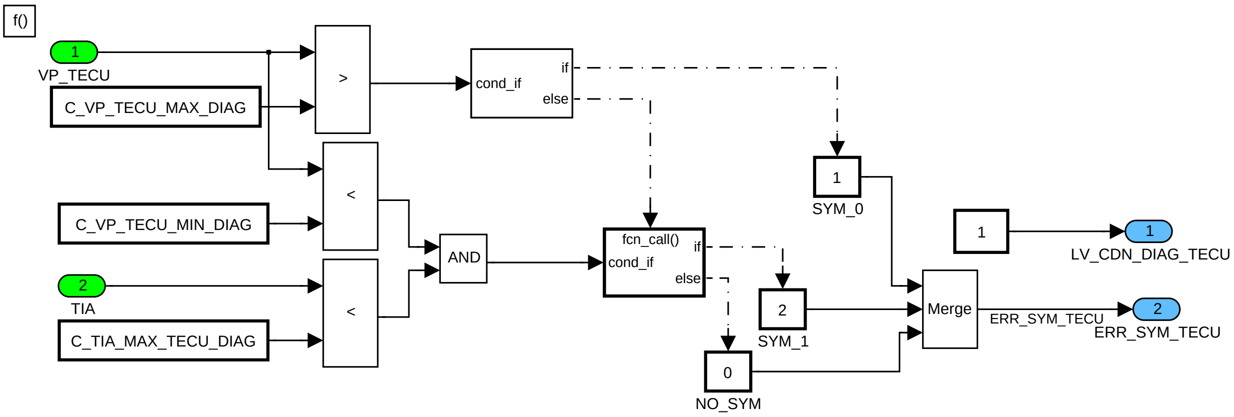


Figure 32 ECOP_MA0BI/ OPM_500MS/ CLC/ CLC_TRUE/ CHK_CDN_DIAG/ SC_VB_OL

Storage of LV_ERR

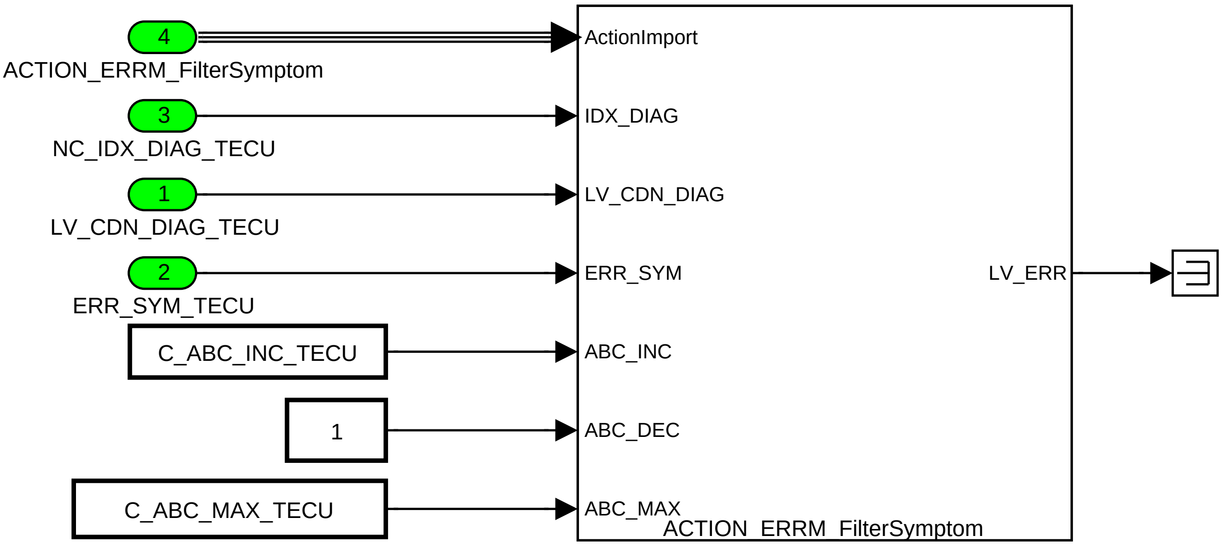


Figure 33 ECOP_MA0BI/ OPM_500MS/ CLC/ CLC_TRUE/ ERRM_IF

Storage of Error Management Varables

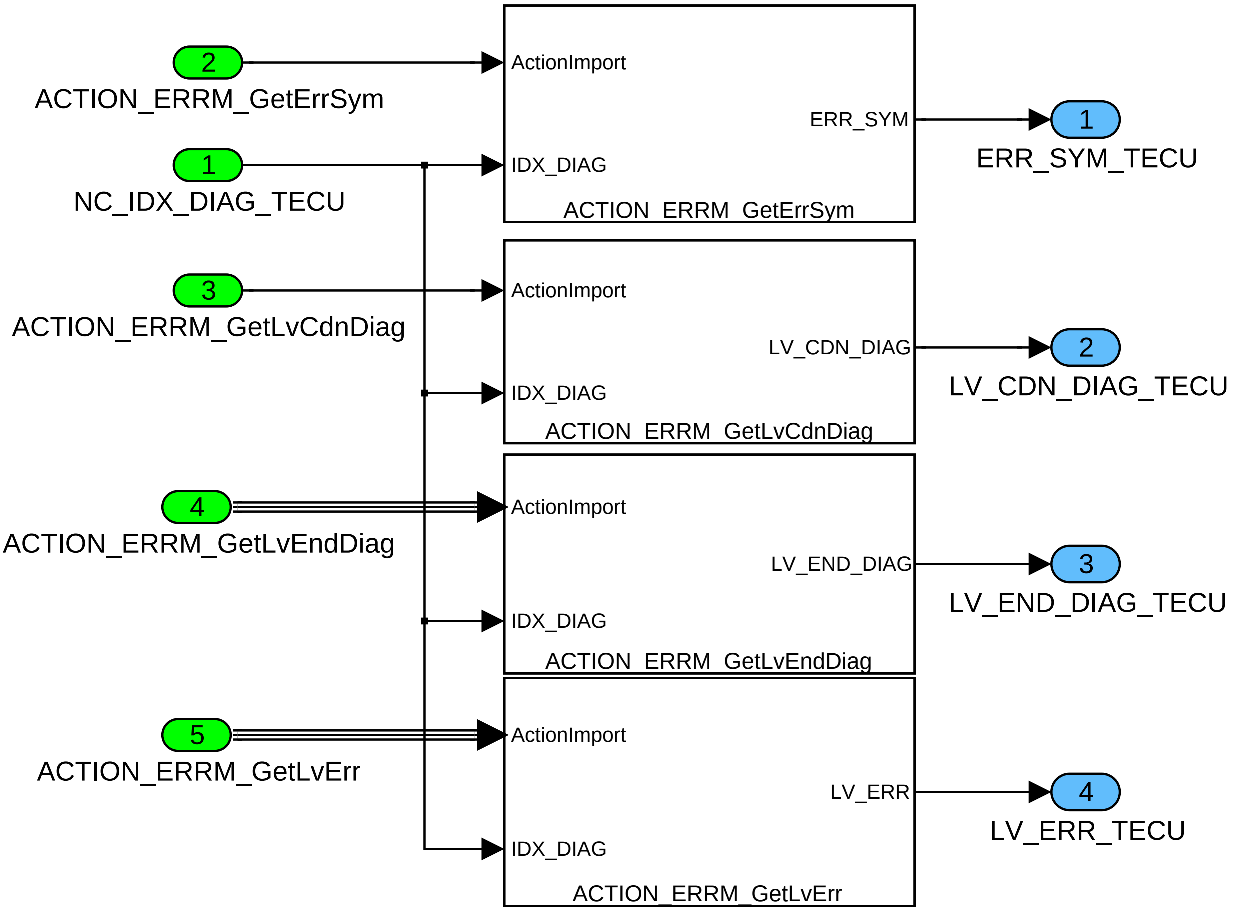


Figure 34 ECOP_MA0BI/ OPM_500MS/ CLC/ CLC_TRUE/ ERRM_IF1

15 ECU Reprogramming

Table of Contents

Index

Error! No index entries found.

15.1 ECU reprogramming, general functionality

15.1.1 Introduction

This document describes the flash reprogramming concept. The concept is always applied.

It is a generic specification. For the customer particular requirements another document called “ECU reprogramming, SW specification” describes the software specification of the reprogramming function.

In addition to this current document the customer specific parts (diagnosis protocol, logistics and historic field) have to be described in other files referring to this document.

This document does not describe the first programming of an ECU at the Siemens production line or the first programming of prototypes.

15.1.2 General description

The usage of suitable segmented flash memories, with the opportunity to delete and (re-) program parts of the module separately, enables the development of a strategy called reprogramming of the Flash memory at the customer.

The car manufacturers use this feature in order to delete the Calibration data and/or ECU-SW and to program them again. With this method it is possible to avoid the exchange of the ECU when functional changes or errors of the ECU-SW occur.

The result is a considerable decrease in expenses of the after-sales service for the customer and the supplier.

The activation of the function and the communication actions are done through a unique communication line (diagnosis line) without applying an external voltage to a specific pin of the ECU connector.

The ECU memory is divided into three principal and functional independent parts : Boot sector, ECU-SW sector and Calibration data sector.

The reprogramming function needs always an existing Boot-SW because all the software routines for a reprogramming session are located in the Boot sector (Boot-SW).

The fundamental restriction is that this sector is not erasable.

The effects resulting from the user actions during a reprogramming session (e.g. key off/key on, false use of the tool, manipulation) and external disruptions on the communication line are not predictable. So the ECU has to ensure that unauthorized actions or failures in the reprogramming session have no impact on the operation of the Boot-SW / ECU-SW.

The following principle requirements have to be taken into consideration for the implementation of the reprogramming function :

- The reprogramming function can be used for the end of line and the field programming.
- The routines for the programming / erasing of the flash memory must be executed from outside the Flash memory itself, because a read access to the flash memory is not possible during programming / erasing.
- For the activation of the reprogramming function and the communication during the session only one physical layer (e.g. K-line) is used.
- For the reprogramming function the customer specific diagnostic protocol is used.
- The software is divided into three independent parts (Boot-SW, ECU-SW, and Calibration data).
- Each part of the SW is protected with its own, separate checksum.
- The reprogramming function needs always an existing Boot-SW.
- For the customer it is not possible to reprogram the Boot-SW.
- Security concept to protect the ECUs against unauthorized manipulation.
- Security concept to avoid the “recycling ECUs” when the reprogramming is not correct.
- Check mechanism for compatibility between Boot-SW, ECU-SW and calibration data.
- Information about the reprogramming status of the ECU easily available for the tool.

15.1.3 Communication between reprogramming Tool and ECU

The communication between the reprogramming tool and the ECU is carried out through a unique bi-directional data link (in general K-line).

The reprogramming services are part of the diagnosis services. The basic services are listed below :

- Communication initialization,
- ECU identification (read logistics information),
- Reprogramming Tool identification (security access),
- Erase action (Calibration data sector and/or ECU-SW sector),
- Transfer actions (read / write Calibration data and/or ECU-SW),
- Reprogramming control and report (checksum, coherence Boot-SW / ECU-SW / Calibration data),
- Historic update (write bytes in Boot sector),
- End communication.

All the diagnostic services that are needed for a reprogramming session must be stored in the Boot-SW (non-erasable sector).

The performance (duration) of a reprogramming session depends on the performance of the communication protocol, except for the Flash memory erase sequence, that depends on the Flash memory type.

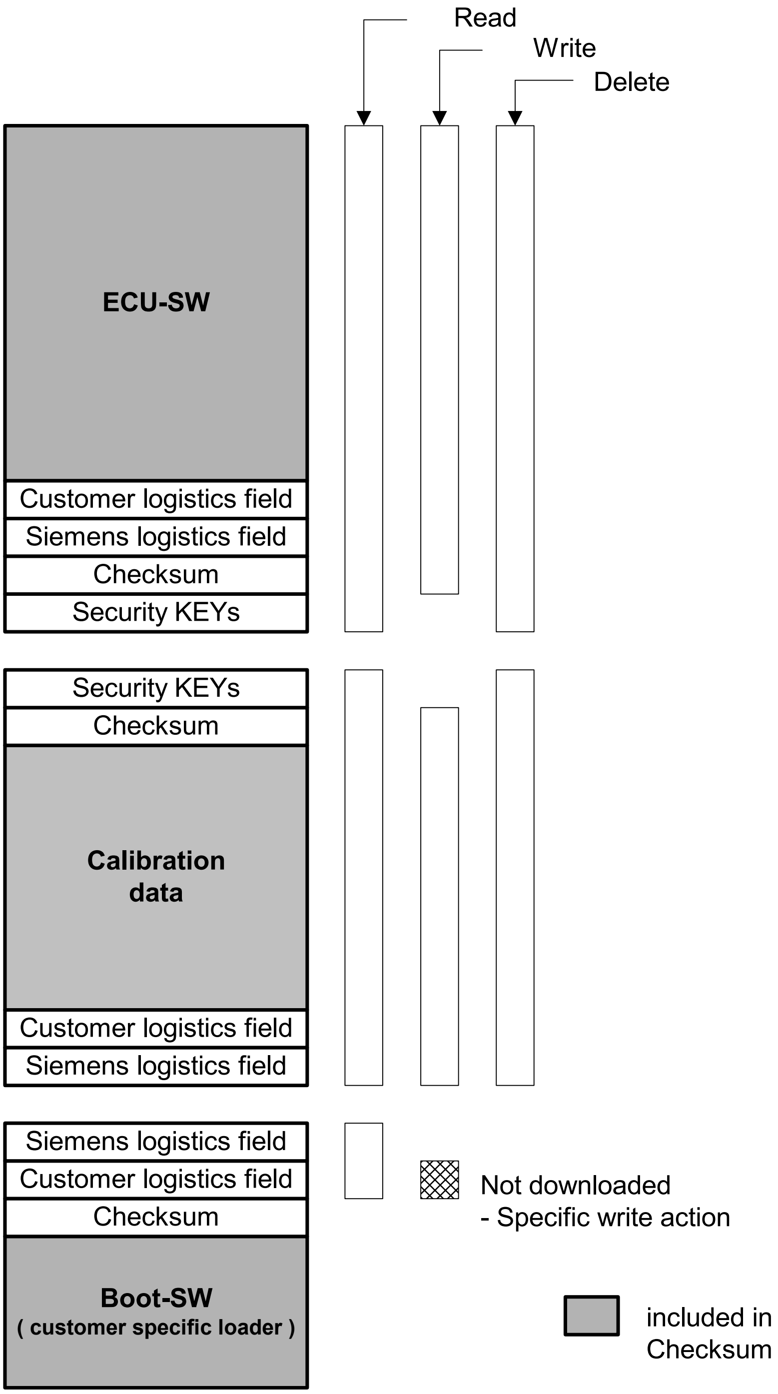
The diagnosis protocol and services have to be described in other documents.

15.1.4 Functional memory layout

The software is divided into three parts :

Boot-SW	cannot be reprogrammed through the customer protocol, size depends on the used flash memory.
ECU-SW	reprogrammable, size depends on the used flash memory.
Calibration data	reprogrammable, size depends on the used flash memory.

The following picture shows the principle architecture.



15.1.4.1 Boot-SW

This sector is not erasable through the customer protocol; it contains all routines used at start-up of the system and all diagnosis services for a reprogramming session.

A field is reserved for customer logistics information.

Contents :

- Interface for ECU-SW,
- Minimum operating system,
- RAM-Test,
- Key_on / Key_off recognition mechanism,
- Customer specific communication protocol layer,
- Diagnosis communication services for reprogramming and logistics,
- Reprogramming routines (delete, start, transfer, exit and check) for the flash memory (please report to chapter 15.1.7 -Reprogramming states),
- Flash access routines for erasing and reprogramming,
- Siemens logistics information field (test results of the Siemens production line, logistics data, identifiers, checksum),
- Customer logistics information field (e.g. the historic of the reprogramming actions stored at the end of reprogramming session),
- Security mechanisms (checksum calculation, coherence and security keys check),
- Seed / Key procedure.

15.1.4.2 ECU-SW

The ECU-SW size depends on the functional system requirements. The limitation to a maximum size depends on the used flash device.

This part contains all routines of the applicative SW of the system.

The reprogramming actions affect the complete area, except the security fields (please report to chapter 15.1.5 - Security mechanisms).

Contents :

- Interface for Boot-SW,
- Basic SW functions which are not necessary for the reprogramming,
- All I/O and applicative SW routines,
- Communication protocol routines,
- Seed / Key procedure,
- Security keys for ECU-SW which guarantee the correct reprogramming,
- Logistic information field (Siemens and Customer),
- Interrupt vector table.

The logistic information field of the ECU-SW, the checksum information and the interrupt vector table are located at fixed addresses in the same way as the interface table for the Boot-SW.

15.1.4.3 Calibration data

The calibration data contains all information for engine tuning.

The reprogramming actions affect the complete area, except the security fields (please report to chapter 15.1.5 -Security mechanisms).

Contents :

- All constants and calibration maps tuned via application tool,
- Security keys for the calibration data which guarantee the correct reprogramming,
- Identification field (comparable to logistic information field within the software),
- Customer logistics field (if necessary).

The checksum information has a definite size and is located at a fixed address within the calibration data.

15.1.5 Security mechanisms

The effects resulting from user actions during a reprogramming session or caused by unauthorized manipulation are not predictable. So a protection is necessary to avoid erroneous flash contents as a consequence of such an influence.

The SW has to ensure a high level of security so that the above mentioned cases have no impact on the operation of the ECU. Therefore special security mechanisms are implemented. Some for protection and the other for a safe detection if correct reprogrammed.

- Security keys in flash memory and RAM to secure the reprogramming session,
- Status word in RAM to guaranty the proper sequence of the reprogramming states,
- Seed/Key procedure (according to SAE J2186 / J2190) via the data link to allow a reprogramming session,
- When an ECU-SW reprogramming session is active, only reprogramming within the ECU-SW memory address range is allowed.
- When a calibration data session is active, only reprogramming within the calibration memory address range is allowed.
- ADD-ON programming for logistics data is only allowed, when no ECU-SW or Calibration data reprogramming is active.
- Checksum test to detect data transmission errors on the data link,
- Check mechanism using a coherence system,
- All logistics data in the BOOT-SW cannot be deleted over the lifetime of the ECU.
- As long as a reprogramming session is not successfully completed, the ECU remains in the reprogramming state (only activation of Boot-SW); this is necessary to avoid “recycling ECUs”.

15.1.5.1 Security Access

To avoid unauthorized manipulation (read / erase / write actions) of the ECU data, security accesses (Siemens / Customer) are implemented to allow / forbid access to the content of the flash memory.

It is not possible to reprogram the ECU-SW and/or the calibration data if the ECU is not unlocked by a successfully security access phase (according to SAE J2186) between the reprogramming tool and the ECU.

The access to higher security levels (or unlocked state) is realized by two successfully handshake messages (request / response) between the tool and the ECU. Thus the ECU certifies the tool.

15.1.5.2 Reprogramming Keys in RAM

To avoid unpredictable erase / write actions of the ECU data, the basic routines for erase and write actions are inhibited if the Reprogramming Keys are not initialized.

The Reprogramming Keys, 4 RAM bytes, are initialized at each erase / write request that concerns the reprogramming function. These keys are tested by the basic routines (erase / write).

At the end of each erase / write routine and on ECU reset, these bytes are set to 00h.

15.1.5.3 Checksum calculation

To guaranty the validity of the reprogrammed data transmitted through the communication line an independent checksum value is stored in each area (ECU-SW, Calibration data).

This checksum is tested at the end of a reprogramming session. If the calculated values do not fit with the stored values in ECU-SW area and Calibration data area, the applicative software is inhibited and there is no engine operation.

15.1.5.4 Coherence System

The Coherence System ensures that the three independent SW-parts fit together. The coherence identifiers are implemented in the Boot-SW, the ECU-SW and the Calibration data and are checked at the end of a reprogramming session and at start-up of the ECU.

If the three identifiers do not fit together, the applicative software is inhibited and there is no engine operation.

15.1.5.5 Security Keys in flash memory

To guaranty that a reprogramming session is successfully completed, a four bytes Security Key is implemented in each part, which can be reprogrammed (ECU-SW, Calibration data).

These keys are updated at the end of a reprogramming session, and tested at start-up of the ECU. If one of the Security Keys is not valid, the applicative software is inhibited and there is no engine operation.

Functional description of the Security Keys :

- After a reprogramming request, the Security Key is set to 00h.
- During the erase operation the Security Key is automatically set to default value of the flash.
- At the end of the reprogramming operation the checksum and the Coherence System are checked, in case of a proper reprogramming session the Security Key is set to a fixed value.

15.1.5.6 Reprogramming session status

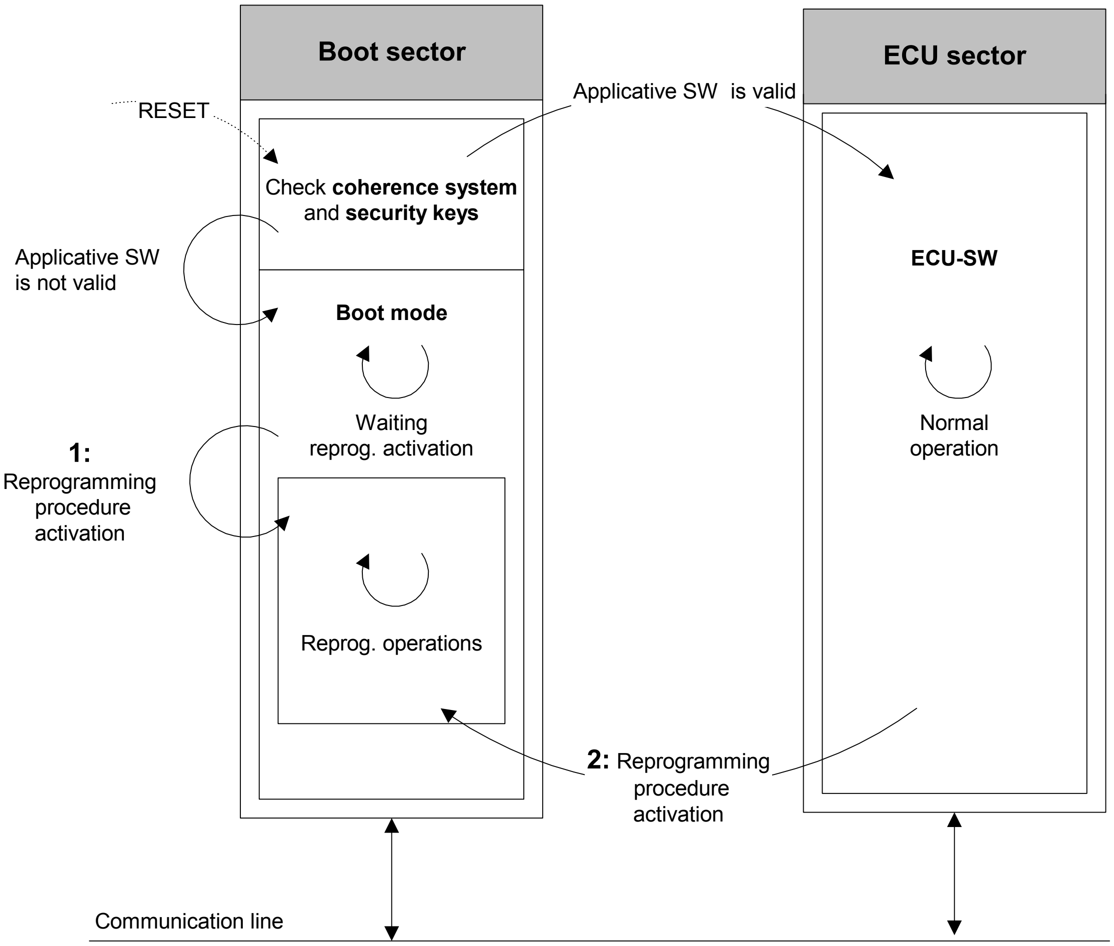
To guaranty the proper sequence of the reprogramming session, a session status is tested and updated in each basic action (please report to chapter 15.1.7 -Reprogramming states).

This mechanism avoids to skip a phase of the reprogramming states.

15.1.6 Reprogramming procedure activation

The following picture shows the principle mechanisms to activate a reprogramming session :

1. In that case, the applicative SW is inhibited; the Boot-SW controlled the communication line interface and the diagnosis services for the reprogramming activation.
2. In that case, ECU is in applicative mode. It should switch to Boot mode before to launch the reprogramming function.



The applicative SW is only activated if all parts are correct reprogrammed and fit together.
If one part is missing or not correct reprogrammed the system actives the Boot-SW.

The following conditions are possible at start-up (Key-On, Reset) of the system :

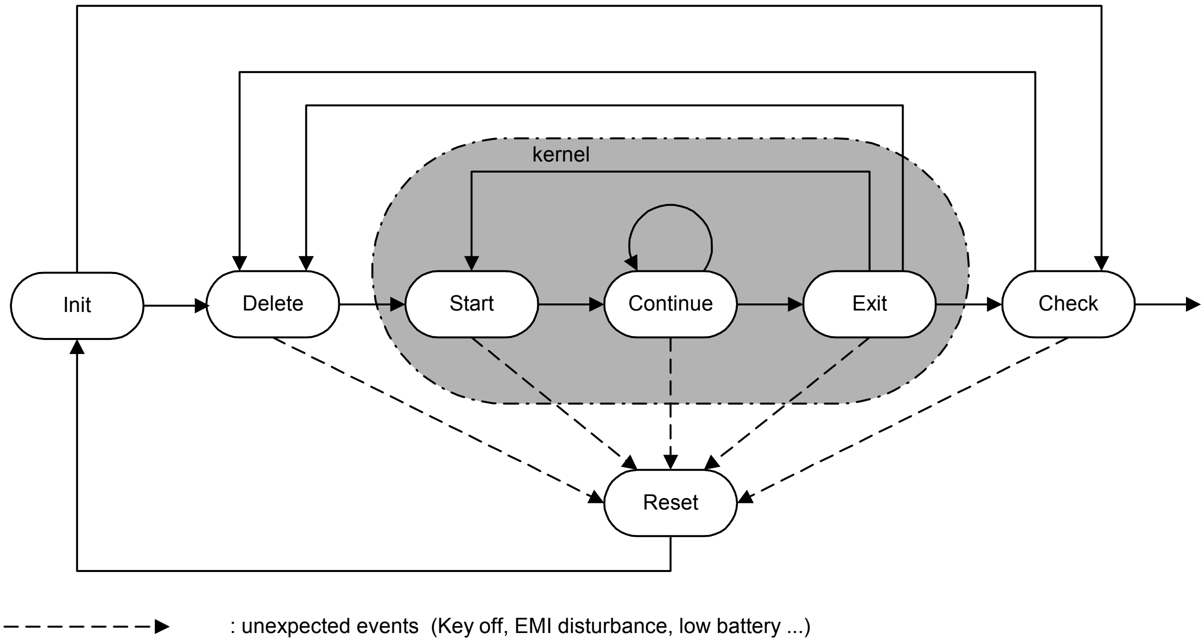
Boot-SW	ECU-SW	Calib. data	ECU action
X	[-]	[-]	ECU activates only Boot-SW, no engine operation
X	[-]	X	ECU activates only Boot-SW, no engine operation
X	[X]	X	ECU activates only Boot-SW, no engine operation
X	[X]	[X]	ECU activates only Boot-SW, no engine operation
X	X	[-]	ECU activates only Boot-SW, no engine operation
X	X	[X]	ECU activates only Boot-SW, no engine operation
X	X	X	ECU normal operation = engine operation

(X = available, [X] = wrong reprogrammed or coherence system value not correct, [-] = erased)

15.1.7 Reprogramming states

These reprogramming states are only possible when the ECU is unlocked by the tool.

REPROGRAMMING STATES



15.1.7.1 State “Init”

First step to the flash access, only necessary once during an ECU reprogramming session.

- Prepare a reprogramming session
    - The diagnostic session has to be opened.
    - The timings of the protocol have to be set to higher limits in order to reduce the reprogramming duration.
- Unlock ECU for a reprogramming session

15.1.7.2 State “Delete”

- The only action is to delete either calibration data or ECU-SW.
- This state also activates the Boot SW mode.

15.1.7.3 State "Start"

- Open a reprogramming session for calibration data or ECU-SW.
- No data are deleted.
- No reprogramming of data.
- Check if address area is allowed.

15.1.7.4 State "Continue"

- An active reprogramming session for calibration data or ECU-SW is continued.
- Reprogramming of data will be done.
- No data are deleted.

Note: If a session for reprogramming of calibration data is opened, reprogramming of ECU-SW is not possible and vice versa.

15.1.7.5 State "Exit"

- Close a reprogramming session for calibration data or ECU-SW.
- No checks are done.
- No reprogramming of data.

15.1.7.6 State "Check"

- The calibration data and ECU-SW are checked (coherence system, checksums).
- No reprogramming of data (except the Security Keys if necessary to write).
- Update of the reprogramming status word; please report to the chapter 15.1.8 - Reprogramming Status Word definition.

15.1.7.7 State "Reset"

- Reset the status information for reprogramming. Useful when the communication end or a break was detected.

15.1.8 Reprogramming Status Word definition

The Reprogramming Status Word informs users on the validity of the applicative SW.

This word content is a dynamic information (RAM) and is only valid if the check of reprogramming have been previously performed.

At reset this information is lost.

Bit #	Bit definition	Group definition
#15	Checksum of Calibration data is correct	Calibration data information
#14	Security Keys for Calibration data are not written	
#13	Security Keys for Calibration data are correct	
#12	Calibration data is correct	
#11	Checksum of ECU-SW is correct	ECU-SW information
#10	Security Keys for ECU-SW are not written	
#9	Security Keys for ECU-SW are correct	
#8	ECU-SW is correct	
#7	Reprogramming of ECU is successfully completed	General ECU information
#6	ECU is not at the end of reprogramming session	
#5	Coherence identifiers fit together	Coherence system information
#4	Calibration data does not fit to ECU-SW	
#3	ECU-SW does not fit to Boot-SW	
#2	Coherence identifier in Calibration data is erroneous	
#1	Coherence identifier in ECU-SW is erroneous	
#0	Coherence identifier in Boot-SW is erroneous	

When the bit definition is right, the bit value is set to 1.

e.g. just at the end of successful reprogramming session, the value is BBh A0h.

15.2 Flash reprogramming

15.2.1 The message flow of flash-reprogramming with the WinProF Tool (aftersales)

Time	client (Tester)	Server (ECU)
W5	idle Time (300ms)	
WuP	Wake-up-Pattern (50ms)	
	startCommunication.Req	
P2		startCommunication.PosRsp[KB1,KB2]
		{Normal Timing is enabled by the keybytes}
P3	startDiagnosticSession[ECUProgrammingMode]	
P2	+ optional setting of the baudrate	startDiagnosticSession Pos.Rsp
P3	accessTimingParameterReq[setParameters]	
P2		accessTimingParameterPosRsp[setParameters]
P3	startRoutineByLocalIdentifier	
P2	[Erase Flash 'Data-area' (ID #02h)]	startRoutineByLocalIdentifier Pos.Rsp
P3	StartRoutineByLocalIdentifier	
P2	[Erase Flash 'Code-area' (ID #02h)]	startRoutineByLocalIdentifier Pos.Rsp
P3	RequestDownloadRequestServiceId	
P2	(Download 'Code-area internal flash 1')	requestDownload Pos.Rsp.
P3	transferData Request Service Id [#1]	
P2		transferData Pos.Rsp.[#1]
P3	transferData Request Service Id [#2]	
P2		transferData Pos.Rsp. [#2]
:	:	:
P3	transferData Request Service Id [#n]	
P2		transferData Pos.Rsp. [#n]
P3	RequestDownloadRequestServiceId	
P2	(Download 'Code-area internal flash 2')	requestDownload Pos.Rsp.
P3	transferData Request Service Id [#1]	
P2		transferData Pos.Rsp.[#1]
P3	transferData Request Service Id [#2]	
P2		transferData Pos.Rsp. [#2]
:	:	:
P3	transferData Request Service Id [#n]	
P2		transferData Pos.Rsp. [#n]
P3	RequestDownloadRequestServiceId	
P2	(Download 'Code-area external flash')	requestDownload Pos.Rsp.
P3	transferData Request Service Id [#1]	
P2		transferData Pos.Rsp.[#1]
P3	transferData Request Service Id [#2]	
P2		transferData Pos.Rsp. [#2]
:	:	:
P3	transferData Request Service Id [#n]	
P2		transferData Pos.Rsp. [#n]
P3	RequestTransferExit	
P2		requestTransferExit Pos.Rsp.
P3	RequestDownloadRequestServiceId	
P2	(Download 'Data-area')	requestDownload Pos.Rsp.

P3 P2 P3 P2 : P3 P2	transferData Request Service Id [#1] transferData Request Service Id [#2] : transferData Request Service Id [#n]	 transferData Pos.Rsp.[#1] transferData Pos.Rsp. [#2] : transferData Pos.Rsp. [#n]
P3 P2	RequestTransferExit	 requestTransferExit Pos.Rsp.
P3 P2	StartRoutineByLocalIdentifier [Check reprogramming dependings(ID #01h)]	 startRoutineByLocalIdentifier Pos.Rsp
P3 P2	StartRoutineByLocalIdentifier [Report reprogramming status(ID #0Ah)]	 startRoutineByLocalIdentifier Pos.Rsp
P3 P2	EcuReset	 ecuReset Pos.Rsp

Table Message FlowFlash programming

16 Exhaust gas composition

Table of Contents

Index

16.1 EGCP Configuration Data

Input data:

NC_FAC_R_REF_LS_DOWN	NC_R_REF_LS_DOWN	NC_VLS_DOWN_CUR_PUMP_REF	NC_LSHPWM_BOL_LSH_DOWN
NC_LSHPWM_TOL_LSH_DOWN	NC_LSHPWM_BOL_LSH_UP	NC_LSHPWM_TOL_LSH_UP	NC_FAC_10_SAMPLE
NC_FAC_5_SAMPLE	NC_NR_BUF_SAMPLE_LEN	NC_NR_RAW_BUF_LEN	NC_N_MIN_VLS_UP_MV
NC_T_SEG_LEN_MIN	NC_NR_BUF_SUM_LEN	NC_PSD_DLY_LSH_UP	NC_PSD_DLY_LSH_DOWN

16.1.1 Global configuration data

Here are listed the configuration data specific for the projects using the Tricore processor and which can be used in other aggregates:

ATTENTION: NC_CBK_EX_NR has the value 2 for 4 and 6 cylinder engines. For N43 UL the second bank is virtual to fit the system.

Data	**Value**
NC_CBK_EX_NR	2
NC_STATE_LSL_UP_IF	1
NC_STATE_VLS_UP_SIG_ACQ	1
NC_FRQ_LSHPWM_UP	100 Hz
NC_FRQ_LSHPWM_DOWN	10 Hz
NC_PSD_DLY_LSH_UP	2
NC_PSD_DLY_LSH_DOWN	2
NC_FAC_R_REF_LS_DOWN	0.07957
NC_R_REF_LS_DOWN	3740 Ohm
NC_VLS_DOWN_CUR_PUMP_REF	3.22 V

16.1.2 Local configuration data

Here are listed the configuration data, which are used only in the EGCP aggregate.

Data	**Value**
NC_LSHPWM_BOL_LSH_DOWN	1 %
NC_LSHPWM_TOL_LSH_DOWN	99 %
NC_LSHPWM_BOL_LSH_UP	0 %
NC_LSHPWM_TOL_LSH_UP	97.265625 %
NC_FAC_10_SAMPLE	2
NC_FAC_5_SAMPLE	4
NC_NR_BUF_SAMPLE_LEN	12

Data	Value
NC_NR_RAW_BUF_LEN	10
NC_N_MIN_VLS_UP_MV	500 rpm
NC_T_SEG_LEN_MIN	0.01 s
NC_NR_BUF_SUM_LEN	24
NLC_LSH_RLY_EFP	0
NC_NR_CONF_LSH_RLY	0
NC_FAC_MAF_INT_20	20/3.6 msec

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CBK_EX_NR	1	1...4H	1...4	1	[-]
Number of exhaust cylinder banks					
NC_STATE_LSL_UP_IF	1	0...7H	0...7	1	[-]
Interface to lambda sensor linear upstream					
NC_STATE_VLS_UP_SIG_ACQ	1	0...7H	0...7	1	[-]
Configuration switch for optional signal acquisition					
NC_FRQ_LSHPWM_UP	1	0...FFFFH	0...4095.9375	0.0625	[Hz]
PWM frequency for the upstream oxygen sensor heater					
NC_FRQ_LSHPWM_DOWN	1	0...FFFFH	0...4095.9375	0.0625	[Hz]
PWM frequency for the downstream oxygen sensor heater					
NLC_LSH_RLY_EFP	1	0...1H	0...1	1	[-]
Configuration switch to indicate whether lambda sensor heater is connected to EFP or main relay					
NC_NR_CONF_LSH_RLY	1	0...7H	0...7	1	[-]
Numerical constant which describes the lambda sensor heater wiring					
NC_FAC_MAF_INT_20	1	0...FFFFH	0...0.89236	0.0136e-3	[s]
unit adaptation factor					

Notes:

- *The configuration switch NC_STATE_VLS_UP_SIG_ACQ denotes whether signal filtering shall be applied to the linear lambda sensor's raw signal provided by the BSW prior to further signal processing. If signal filtering shall be applied NC_STATE_VLS_UP_SIG_ACQ shall be set to 1. If not or if binary upstream sensor is used NC_STATE_VLS_UP_SIG_ACQ shall be set to 0.*

- *The PWM frequency for the oxygen sensor heater must be adjusted to values larger than 5Hz, i.e.* NC_FRQ_LSHPWM_UP >= 5Hz and NC_FRQ_LSHPWM_DOWN >= *5Hz. Usually no values larger than 100 Hz are applied, the exact value has to be derived from the corresponding sensor specification.*

16.2 EGCP - Requirements to Infrastructure Interface

Input data:

NC_CBK_EX_NR			

Export actions:

ACTION_INFR_CSetPwmLshUp(IN <i>, **IN** <ducy>**)**
This action sets the lambda heater upstream driver
ACTION_INFR_CSetPwmLshDown(IN <i>, **IN** <ducy>**)**
This action sets the lambda heater downstream driver
ACTION_INFR_CGetVAdcLsUp(IN <I>, **OUT** <v_adc>**)**
This action gets the raw value from the upstream oxygen sensor voltage
ACTION_INFR_CGetVAdcLsDown(IN <i>, **OUT** <v_adc>**)**
This action gets the oxygen sensor voltage downstream
ACTION_INFR_CGetVAdcTempLsUp(IN <i>, **OUT** <v_adc>**)**
This action gets the oxygen sensor voltage upstream
ACTION_INFR_CEnableCilc()
This action sets the activation of CILC (cylinder individuel lambda control)
ACTION_INFR_CDisableCilc()
This action sets the deactivation of CILC
ACTION_INFR_CGetCilc(IN <x>, **OUT** <v_adc, **OUT** <ctr>**)**
Signal acquisition for CILC
ACTION_INFR_CSetCilc(IN <x>, **IN** <crk>**)**
This action provide actual phase displacement for each cylinder
ACTION_INFR_CGetElDiagLshUp(IN <i>, **OUT** <cdn_diag>, **OUT** <err_diag>**)**
This action reads the failure and the conditions information for each symptom from the infrastructure
ACTION_INFR_CGetElDiagLshDown(IN <i>, **OUT** <cdn_diag>, **OUT** <err_diag>**)**
This action reads the failure and the conditions information for each symptom from the infrastructure
ACTION_INFR_CGetWrafControl(IN <i>, **OUT** <lv_lsl_deac>, **OUT** <state_err_vnlsl>, **OUT** <state_err_vglsl>, **OUT** <state_err_viplsl>, **OUT** <state_err_vrclsl>, **OUT** <state_err_iplsl>, **OUT** <state_err_lsl>, **OUT** <state_lsl_if_spi_rd>, **OUT** <state_lsl_if_conf_spi_rd>, **OUT** <frq_r_it_osc_lsl_if_spi_rd>, **OUT** <ctr_err_lsl_if_spi_wr>, **OUT** <ctr_err_lsl_if_spi_it_osc_ena>, **OUT** <lv_vplsl_lim>, **OUT** <lv_vnlsl_lim>**)**
This action reads the values from the WRAF controller
ACTION_INFR_CSetWrafControl(IN <i>, **IN** <state_lsl_if_spi_wr>, **IN** <state_lsl_if_conf_spi_wr>, **IN** <frq_r_it_osc_lsl_if_spi_wr>, **IN** <lv_lsl_if_rst>, **IN** <lv_lsl_up_spi_com_inh>, **IN** <lv_ctr_err_lsl_if_spi_man_inc>**)**
This action sets the values of the WRAF controller
ACTION_INFR_CSetCmdLsUp(IN <i>, **IN** <lv_swi_ls_up>**)**
This action starts the internal resistance determination by "pumping" the sensor.
ACTION_INFR_CSetCmdLsDown(IN <i>, **IN** <lv_swi_ls_down>**)**
This action starts the internal resistance determination by "pumping" the sensor.
ACTION_INFR_CSetHeatCoupling(IN <i>, **IN** <t_dly_trig_lsh_up>, **IN** <lamb_sp_fil_hom>**)**
This action sets the value of the WRAF controller for the heater coupling diagnosis.
ACTION_INFR_CGetHeatCoupling(IN <i>, **OUT** <lamb_sp_fil_delta_rise>, **OUT** <lamb_sp_fil_delta_fall>, **OUT** <vls_up_delta_rise>, **OUT** <vls_up_delta_fall>**)**
This action gets the value from the WRAF controller for the heater coupling diagnosis.

General information:

a) **i = NC_CBK_EX_NR** (Number of exhaust gas cylinder banks)

b) **x = NC_CYL_NR** (Number of engine cylinders)

Description for actions:

ACTION_INFR_CSetPwmLshUp(i, ducy **)**					
This action sets the lambda heater upstream driver					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
ducy	IN	0...FFH	0...99.609375	0.390625	%
Up Heater driver PWM duty cycle					

ACTION_INFR_CSetPwmLshDown(i, ducy **)**					
This action sets the lambda heater downstream driver					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
ducy	IN	0…FFH	0…99.609375	0.390625	%
Down Heater driver PWM duty cycle					

ACTION_INFR_CGetVAdcLsUp(i, v_adc **)**					
This action gets the raw value from the upstream oxygen sensor voltage.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
v_adc	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Raw value of upstream oxygen sensor voltage					

ACTION_INFR_CGetVAdcLsDown(i, v_adc **)**					
This action gets the oxygen sensor voltage upstream					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
v_adc	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Downstream oxygen sensor voltage					

ACTION_INFR_CGetVAdcTempLsUp(i, v_adc **)**					
This action sets the upstream lambda sensor temperature voltage					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
v_adc	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Upstream lambda sensor temperature voltage					

ACTION_INFR_CEnableCilc()					
This action sets the activation of CILC (cylinder individuel lambda control)					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

ACTION_INFR_CDisableCilc()					
This action sets the deactivation of CILC					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

ACTION_INFR_CGetCilc(x, v_adc, ctr **)**					
Signal acquisition for CILC					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
x	IN	**0…7H**	**0…7**	**1**	**-**
logical index of cylinder to be acquired					
v_adc	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
upstream oxygen sensor acquisition (cylinder individual)					
Ctr	OUT	0…FFH	0…255	1	-
cylinder selective update counter for trigger event					

ACTION_INFR_CSetCilc(x, crk **)**					
This action provide actual phase displacement for each cylinder. Remark: The maximum angle delay which can be realized by the BSW is 720°CRK. Larger values shall be calculated beginning with 0°CRK.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
x	IN	**0…7H**	**0…7**	**1**	**-**
logical index of cylinder to be changed					
crk	IN	0…F0H	0…1440	6	°CRK
Value of phase displacement					

ACTION_INFR_CGetElDiagLshUp(i**,** cdn_diag, err_diag **)**					
This action reads the failure and the conditions information for each symptom from the infrastructure The readout of the power stage is performed autonomous and the information is gathered. When calling this Action the information inside the Infrastructure is reset after returning the OUT parameters.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
cdn_diag	OUT	0...7H	0...7	1	-
Bit 0: Set, if error symptom SCP (SYM_0) is detected Bit 1: Set, if error symptom SCG (SYM_1) is detected Bit 2: Set, if error symptom OC (SYM_2) is detected					
err_diag	OUT	0...7H	0...7	1	-
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

ACTION_INFR_CGetElDiagLshDown(i**,** cdn_diag, err_diag **)**					
This action reads the failure and the conditions information for each symptom from the infrastructure The readout of the power stage is performed autonomous and the information is gathered. When calling this Action the information inside the Infrastructure is reset after returning the OUT parameters.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
cdn_diag	OUT	0...7H	0...7	1	-
Bit 0: Set, if error symptom SCP (SYM_0) is detected Bit 1: Set, if error symptom SCG (SYM_1) is detected Bit 2: Set, if error symptom OC (SYM_2) is detected					
err_diag	OUT	0...7H	0...7	1	-
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

ACTION_INFR_CGetWrafControl(i, lv_lsl_deac, state_err_vnlsl, state_err_vglsl, state_err_viplsl, state_err_vrclsl, state_err_iplsl, state_err_lsl, state_lsl_if_spi_rd, state_lsl_if_conf_spi_rd, frq_r_it_osc_lsl_if_spi_rd, ctr_err_lsl_if_spi_wr, ctr_err_lsl_if_spi_it_osc_ena, lv_vplsl_lim, lv_vnlsl_lim **)**
This action reads the values from the WRAF controller.

Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
lv_lsl_deac	OUT	0...1H	0...1	1	-
Boolean flag indicating OBDI signal fault pending					
state_err_vnlsl	OUT	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Nernst Cell positive pin.					
state_err_vglsl	OUT	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Virtual Ground pin.					
state_err_viplsl	OUT	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Pump Cell pin.					
state_err_vrclsl	OUT	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Calibration Resistance pin.					
state_err_iplsl	OUT	0H 1H 2H	NO_FAULT CURRENT_LOW CURRENT_HIGH	1	[-]
Diagnosis status of the Pumping Current.					
state_err_lsl	OUT	0H 1H	NO_FAULT FAULT	1	[-]
OBDI General Status					
state_lsl_if_spi_rd	OUT	0...FFH	0...255	1	[-]
Bit-field, mapping verified contents of WRAF controller Configuration Register					
state_lsl_if_conf_spi_rd	OUT	0...FFH	0...255	1	[-]
Content of the configuration register comprising sensor disconnection time delay and Icp value					
frq_r_it_osc_lsl_if_spi_rd	OUT	0...7FH	0...127	1	[-]
Content of the configuration register comprising Ri-measurement frequency and bit for pin selective disabling					
ctr_err_lsl_if_spi_wr	OUT	0...FFH	0...255	1	[-]
Counter indicating number of invalid INIT_REG_x write validation cycles which have been detected					
ctr_err_lsl_if_spi_it_osc_ena	OUT	0...FFH	0...255	1	[-]
Counter indicating number of invalid transmissions of “R_IT_OSC_ENA” bit					
lv_vplsl_lim	OUT	0…1H	0…1	1	[-]
Boolean flag indicating that Pump Cell voltage has reached its limits					
lv_vnlsl_lim	OUT	0...1H	0...1	1	[-]
Boolean flag indicating overvoltage at Nernst Cell positive pin (reverse Icp cut-off)					

ACTION_INFR_CSetWrafControl(i, state_lsl_if_spi_wr, state_lsl_if_conf_spi_wr, frq_r_it_osc_lsl_if_spi_wr, lv_lsl_if_rst, lv_lsl_up_spi_com_inh, lv_ctr_err_lsl_if_spi_man_inc**)**

This action sets the values of the WRAF controller

Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
state_lsl_if_spi_wr	IN	0...FFH	0...255	1	-
Bit-field mapping of contents to be written to WRAF controller INIT_REG_1					
state_lsl_if_conf_spi_wr	IN	0...FFH	0...255	1	-
Bit-field mapping of contents to be written to WRAF controller INIT_REG_1					
frq_r_it_osc_lsl_if_spi_wr	IN	0...7FH	0...127	1	-
Mapping of contents to be written to WRAF controller INIT_REG_1					
lv_lsl_if_rst	IN	0...1H	0...1	1	-
Boolean flag to reset the WRAF controller					

lv_lsl_up_spi_com_inh	IN	0...1H	0...1	1	-
Inhibit bit for the SPI-communication related to electrical diagnosis on BSW level					
lv_ctr_err_lsl_if_spi_man_inc	IN	0...1H	0...1	1	-
BSW interface for manual increase of value of SPI communication error counter					

ACTION_INFR_CSetCmdLsUp(i, lv_swi_ls_up **)**					
This action starts the internal resistance determination by "pumping" the the sensor.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
lv_swi_ls_up	IN	0H 1H	Switch off Switch on	1	-
Digital command parameter for the setting of pump current for binary upstream lambda sensor Remark: Switch on equal to activation (high or low level)					

ACTION_INFR_CSetCmdLsDown(i, lv_swi_ls_down **)**					
This action starts the internal resistance determination by "pumping" the the sensor.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
lv_swi_ls_down	IN	0H 1H	Switch off Switch on	1	-
Digital command parameter for the setting of pump current for binary downstream lambda sensor Remark: Switch on equal to activation (high or low level)					

ACTION_INFR_CSetHeatCoupling(i, t_dly_trig_lsh_up, lamb_sp_fil_hom **)**					
This action sets the value of the WRAF controller for the heater coupling diagnosis.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
t_dly_trig_lsh_up	IN	0...FFH	0...0.255	0.001	s
Time delay for acquisition of oxygen sensor signal and lambda model					
lamb_sp_fil_hom	IN	0...7FFFH	0...1.999939	6.10351573229163 E-5	-
output signal of the WRAF sensor model for homogeneous mode					

ACTION_INFR_CGetHeatCoupling(i, lamb_sp_fil_delta_rise, lamb_sp_fil_delta_fall, vls_up_delta_rise, vls_up_delta_fall **)**					
This action gets the value from the WRAF controller for the heater coupling diagnosis.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
lamb_sp_fil_delta_rise	OUT	0...7FFFH	0...1.99993896484375	6.10351562E-5	-
Acquired lambda model value at the rising edge of heater PWM					
lamb_sp_fil_delta_fall	OUT	0...7FFFH	0...1.99993896484375	6.10351562E-5	-
Acquired lambda model value at the falling edge of heater PWM					
vls_up_delta_rise	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Acquired oxygen sensor voltage at the rising edge of heater PWM					
vls_up_delta_fall	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Acquired oxygen sensor voltage at the falling edge of heater PWM					

FUNCTION DESCRIPTION:

16.2.1 Control of the up- and downstream LSH Driver

General information:

The goal of the heater control shall be to control the oxygen sensor heating such that the optimal operating temperature is reached in the shortest time possible.

Requirements for ACTION_INFR_CSetPwmLshUp:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i				Index parameter i is related to parameter: ducy	
ducy	-	-	0.390625 %		. By the initialization the PWM value set to 0%.

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The Action has to be called synchronous (e.g. 10 ms) according the recurrences in the specification

Requirements for ACTION_INFR_CSetPwmLshDown:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i				Index parameter i is related to parameter: ducy	
ducy	-	-	0.390625 %		By the initialization the PWM value set to 0%.

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The Action has to be called synchronous (e.g. 10 ms) according the recurrences in the specification

16.2.2 Exhaust gas composition – Requirements for up- and downstream lambda sensor voltage

General information:

Voltages of lambda sensors offered by Basic SW are transformed to application system specific variables.

Requirements for ACTION_INFR_CGetVAdcLsUp:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

i			1	Index parameter i is related to parameter: v_adc	
v_adc		0,1%	1.52588E-4 V		

Diagnosis:

Coincidence requirements: The AD conversion is performed autonomously by the infrastructure all 10 ms (state =0) or 1 ms (state=1)

Requirements for ACTION_INFR_CGetVAdcLsDown:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i			1	Index parameter i is related to parameter: v_adc	
v_adc		0,1 %	1.52588E-4 V		

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The AD conversion is performed autonomously by the infrastructure all 10 ms.

Requirements for ACTION_INFR_CGetVAdcTempLsUp:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i				Index parameter i is related to parameter: v_adc	
v_adc		0,1 %	1.52588E-4 V		

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The AD conversion is performed autonomously by the infrastructure all 10 ms.

16.2.3 Signal acquisition for cylinder individual lambda control (CILC)

General information:

Additionally basic informaton about requirements for the infrastructure can be found in the module 201P with the title **" Signal acquisition for cylinder individual lambda control".**

```
if two separate cylinder banks are concerned, then
    i = 1, for cylinder bank 1 ( x = 0, 2, 4, ... )
    i = 2, for cylinder bank 2 ( x = 1, 3, 5, ... )
else
i = 1 ( x = 0 ... 3 or 4 )
end
x: logical cylinder (NC_CYL_NR)
```

The function shall serve as a detector for a precise signal acquisition of the Wide Range Air-Fuel (WRAF) sensor in accordance with the identified cylinder.

The calculation shall be done for x = 0 : NC_CYL_NR –1.

Requirements for ACTION_INFR_CEnableCilc:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

Coincidence requirements:

Requirements for ACTION_INFR_CDisableCilc:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

Coincidence requirements:

Requirements for ACTION_INFR_CGetCilc:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
x			1	Index parameter x is related to parameter: v_adc, ctr	
v_adc		0,1 %	1.52588E-4 V		
ctr		0,4 %	1		

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The AD conversion is performed autonomously by the infrastructure all 1 ms. This Action has to be called asynchronous

Requirements for ACTION_INFR_CSetCilc:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
x			1	Index parameter x is related to parameter: crk	
crk			6 °CRK		

Diagnosis: no electrical diagnosis done here

Coincidence requirements: This Action has to be called asynchronous

16.2.4 Upstream oxygen sensor heater OBDI diagnosis

General information:

The error detection is performed via the ECU hardware. Error informations are gathered until read out by the application software ASW. At the moment when ASW is reading data from I/O software, the I/O software data will be reset. (It is possible to have more than one electrical failure symptom reported to ASW.)
Following errors may be detected:

1. short cut to battery (scp)
2. short cut to ground (scg)
3. open curcuit (ol)

This action returns the result of the electrical diagnosis of the RCL Actuator Electrical Power Stage

- The device readout is performed autonomously by the Infrastructure. It has to be ensured that the infrastructure interface provides the result before the ASW readout which is defined by the diagnosis frequency.
- The error information are gathered in the Infrastructure until the Application reads out the information by calling ACTION_INFR_CGetElDiagLshUp() / ACTION_INFR_CGetElDiagLshDown().
- After having read out the information by calling ACTION_INFR_CGetElDiagLshUp() / ACTION_INFR_CGetElDiagLshDown(), the data inside the Infrastructure are reset.

The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements.

All symptoms of the current error code are handeled by anti-bouncing.
The error detection is effected via the ECU hardware.

The purpose of the diagnosis shall be to detect electrical faults within the oxygen sensor heater circuit as defined by OBD I requirements.

The function is intended for use in a hardware configuration where the heater driver shall carry out the electrical fault diagnosis and provide the electrical status via status flags. Usually the ASW diagnosis frequency is much lower than the HW diagnosis frequency for PWM signals. Therefore several symptoms may be provided to the ASW within one diagnosis cycle. In this case the following priorization shall be considered.

- *Short circuit to Vbatt by Over-temperature **or** Overcurrent* (**SCP**)
- *Short circuit to GND* (SCG)
- *Open load* (**OC**)

The electrical status flags for each heater driver shall be read out and decoded by the appropriate function and made available to the Application SW in the form of driver and fault specific flags. It must be considered that the PWM frequency shall be at least twice the diagnosis frequency to avoid oversampling

Requirements for ACTION_INFR_CGetElDiagLshUp:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i			1	cdn_diag err_diag	

cdn_diag	4 bit	-	<bit coded>	i err_diag	Diagnosis condition for each symptom **bit 0:** diagnosis condition for symptom SCP (SYM_0) **bit 1:** diagnosis condition for symptom SCG (SYM_1) **bit 2:** diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
err_diag	4 bit	-	<bit coded>	i cdn_diag	Bit coded result of each symptom **0h** = NO_SYM **1h** = SYM_0 – SCP **2h** = SYM_1 – SCG **4h** = SYM_2 – OC The relevant bit is set, if the error has been detected.

Diagnosis: ACTION_INFR_CGetElDiagLshUp returns the electric diagnosis of the corresponding power stage.

Coincidence requirements: This Action has to be called asynchronous

Requirements for ACTION_INFR_CGetElDiagLshDown:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i			1	cdn_diag err_diag	
cdn_diag	4 bit	-	<bit coded>	err_diag	Diagnosis condition for each symptom **bit 0:** diagnosis condition for symptom SCP (SYM_0) **bit 1:** diagnosis condition for symptom SCG (SYM_1) **bit 2:** diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
err_diag	4 bit	-	<bit coded>	cdn_diag	Bit coded result of each symptom **0h** = NO_SYM **1h** = SYM_0 – SCP **2h** = SYM_1 – SCG **4h** = SYM_2 – OC The relevant bit is set, if the error has been detected.

Diagnosis: ACTION_INFR_CGetElDiagLshDown returns the electric diagnosis of the corresponding power stage

Coincidence requirements: This Action has to be called asynchronous

16.2.5 Wide Range Air-Fuel sensor interface (BSW)

General information:

Additionally basic informaton about requirements for the infrastructure can be found in the module 202V with the title **"Wide Range Air-Fuel sensor interface (BSW) for the ATIC42".**

general specification

The function shall describe the interface between the Wide Range Air-Fuel (WRAF) sensor, its controll device and ECU.

The following information can be requested by the ECU from the WRAF sensor:

- [Read from the two IC identity registers IDENT_REG_1[i] and IDENT_REG_2[i] – currently not implemented in generic SW!]
- Configuration and initialization of the 1-byte long **initialisation register**, to be taken from the corresponding BSW Specification
- Configure and initialise the **second configuration register** containing the 5 bit setting for the Icp-current value, and the 3-bit register for protection time prior to sensor disconnecting in case of electrical failure.
- Configure and initialise the 7-bit register containing Ri measurement frequency and the activation switch for selective pin disabling during sensor protection time.
- Read out the sensor pin electrical diagnostic register.
- Read out the general status register
- [Reset the WRAF controller – currently not implemented in generic SW!]

The individual timing requirements and the general description of the SPI may be found in the WRAF controller Component Specification and shall not be repeated here. Only those points that serve to clarify the interface between the WRAF controller and the microcontroller will be elaborated upon.

Requirements for ACTION_INFR_CGetWrafControl:

Data acquisition

Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment
i			1	Index parameter **i** is related to all parameter:	
lv_lsl_deac			1		
state_err_vnlsl			1		
state_err_vglsl			1		
state_err_viplsl			1		
state_err_vrclsl			1		
state_err_iplsl			1		
state_err_lsl			1		
state_lsl_if_spi_rd			1		
state_lsl_if_conf_spi_rd			1		
frq_r_it_osc_lsl_if_spi_rd			1		
ctr_err_lsl_if_spi_wr			1		
ctr_err_lsl_if_spi_it_osc_ena			1		
lv_vplsl_lim			1		
lv_vnlsl_lim			1		

Diagnosis:

Coincidence requirements: This Action has to be called asynchronous

Requirements for ACTION_INFR_CSetWrafControl:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i			1	Index parameter **i** is related to all parameter:	
state_lsl_if_spi_wr			1		
state_lsl_if_conf_spi_wr			1		
frq_r_it_osc_lsl_if_spi_wr			1		
lv_lsl_if_rst			1		
lv_lsl_up_spi_com_inh			1		
lv_ctr_err_lsl_if_spi_man_inc			1		

Diagnosis:

Coincidence requirements: This Action has to be called asynchronous

16.2.6 Up and downstream oxygen sensor internal resistance determination.

General information:

Internal resistance determination is used for binary oxygen sensor temperature determination

Requirements for ACTION_INFR_CSetCmdLsUp:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i			1	Index parameter i is related to parameter: lv_swi_ls_up	
lv_swi_ls_up			1		

Diagnosis:

Coincidence requirements:

Requirements for ACTION_INFR_CSetCmdLsDown:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
i			1	Index parameter i is related to parameter: lv_swi_ls_down	
Lv_swi_ls_down			1		

Diagnosis:

Coincidence requirements:

16.2.7 Acquisition of signal voltage and lambda model values for the Heater Coupling Diagnosis

General information:

Additionally basic informaton about requirements for the infrastructure can be found in the module 202Y with the title **"Acquisition of signal voltage and lambda model values for the Heater Coupling Diagnosis with UEGO Sensors and ATIC42"**

This function shall deliver the inputs needed by the heater coupling diagnosis and describe how the sensor signal and corresponding model values shall be acquired synchronously with the heater signal.

The BSW shall guarantee the edge detection of the PWM signal up to a PWM frequency of 100Hz.
Remarks:
- High and low phase of a PWM signal: minimum 2 ms (which corresponds to the current feasibility of BSW).
- The maximum delay time for edge detection must be equal to or smaller than 2 ms.
- Further requirements have to be clearified with BSW.

Requirements for ACTION_INFR_CSetHeatCoupling:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
I			1	Index parameter **i** is related to all parameter:	
t_dly_trig_lsh_up			0.001 s		
lamb_sp_fil_hom			6.10351573229163 E-5		

Diagnosis:

Coincidence requirements: This Action has to be called asynchronous

Requirements for ACTION_INFR_CGetHeatCoupling:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
I			1	Index parameter **i** is related to all parameter:	
lamb_sp_fil_delta_rise			6.10351562E-5		
lamb_sp_fil_delta_fall			6.10351562E-5		
vls_up_delta_rise		0,1 %	1.52588E-4 V		
vls_up_delta_fall		0,1 %	1.52588E-4 V		

Diagnosis:

Coincidence requirements:

16.3 EGCP – Configuration of Infrastructure Interface

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NR_CONF_CBK_EX	V/O	0...FFH	0...255	1	[-]
current configuration of the technical structure of the exhaust bank with its components.					
NR_CBK_EX	V/O	1...4H	1...4	1	[-]
current number of exhaust gas cylinder banks					

Input data:

NC_CBK_EX_NR	VLS_NS[NC_NOX_SENS_CONF]	CDN_DIAG_LSH_UP[NC_CBK_EX_NR]	ERR_DIAG_LSH_UP[NC_CBK_EX_NR]
CDN_DIAG_LSH_DOWN[NC_CBK_EX_NR]	ERR_DIAG_LSH_DOWN[NC_CBK_EX_NR]		

Export actions:

ACTION_INFR_SetPwmLshUp(IN <i>, **IN** <ducy>**)**
This action sets the lambda heater upstream driver
ACTION_INFR_SetPwmLshDown(IN <i>, **IN** <ducy>**)**
This action sets the lambda heater downstream driver
ACTION_INFR_GetVAdcLsUp(IN <i>, **OUT** <v_adc>**)**
This action gets the raw value from the upstream oxygen sensor voltage
ACTION_INFR_GetVAdcLsDown(IN <i>, **OUT** <v_adc>**)**
This action gets the oxygen sensor voltage downstream
ACTION_INFR_GetVAdcTempLsUp(IN <i>, **OUT** <v_adc>**)**
This action gets the oxygen sensor voltage upstream
ACTION_INFR_EnableCilc()
This action sets the activation of CILC (cylinder individuel lambda control)
ACTION_INFR_DisableCilc()
This action sets the deactivation of CILC
ACTION_INFR_GetCilc(IN <x>, **OUT** <v_adc, **OUT** <ctr>**)**
Signal acquisition for CILC
ACTION_INFR_SetCilc(IN <x>, **IN** <crk>**)**
This action provide actual phase displacement for each cylinder
ACTION_INFR_GetElDiagLshUp(IN <i>, **OUT** <cdn_diag>, **OUT** <err_diag>**)**
This action reads the failure and the conditions information for each symptom from the infrastructure
ACTION_INFR_GetElDiagLshDown(IN <i>, **OUT** <cdn_diag>, **OUT** <err_diag>**)**
This action reads the failure and the conditions information for each symptom from the infrastructure
ACTION_INFR_GetWrafControl(IN <i>, **OUT** <lv_lsl_deac>, **OUT** <state_err_vnlsl>, **OUT** <state_err_vglsl>, **OUT** <state_err_viplsl>, **OUT** <state_err_vrclsl>, **OUT** <state_err_iplsl>, **OUT** <state_err_lsl>, **OUT** <state_lsl_if_spi_rd>, **OUT** <state_lsl_if_conf_spi_rd>, **OUT** <frq_r_it_osc_lsl_if_spi_rd>, **OUT** <ctr_err_lsl_if_spi_wr>, **OUT** <ctr_err_lsl_if_spi_it_osc_ena>, **OUT** <lv_vplsl_lim>, **OUT** <lv_vnlsl_lim>**)**
This action reads the values from the WRAF controller
ACTION_INFR_SetWrafControl(IN <i>, **IN** <state_lsl_if_spi_wr>, **IN** <state_lsl_if_conf_spi_wr>, **IN** <frq_r_it_osc_lsl_if_spi_wr>, **IN** <lv_lsl_if_rst>, **IN** <lv_lsl_up_spi_com_inh>, **IN** <lv_ctr_err_lsl_if_spi_man_inc>**)**
This action sets the values of the WRAF controller
ACTION_INFR_SetCmdLsUp(IN <i>, **IN** <lv_swi_ls_up>**)**
This action starts the internal resistance determination by "pumping" the sensor.
ACTION_INFR_SetCmdLsDown(IN <i>, **IN** <lv_swi_ls_down>**)**
This action starts the internal resistance determination by "pumping" the sensor.

ACTION_INFR_SetHeatCoupling(IN <i>, **IN** <t_dly_trig_lsh_up>, **IN** <lamb_sp_fil_hom>**)**
This action sets the value of the WRAF controller for the heater coupling diagnosis.
ACTION_INFR_GetHeatCoupling(IN <i>, **OUT** <lamb_sp_fil_delta_rise>, **OUT** <lamb_sp_fil_delta_fall>, **OUT** <vls_up_delta_rise>, **OUT** <vls_up_delta_fall>**)**
This action gets the value from the WRAF controller for the heater coupling diagnosis.

Import actions:

ACTION_INFR_CSetPwmLshUp(IN <i>, **IN** <ducy>**)**
ACTION_INFR_CSetPwmLshDown(IN <i>, **IN** <ducy>**)**
ACTION_INFR_CGetVAdcLsUp(IN <i>, **OUT** <v_adc>**)**
ACTION_INFR_CGetVAdcLsDown(IN <i>, **OUT** <v_adc>**)**
ACTION_INFR_CGetVAdcTempLsUp(IN <i>, **OUT** <v_adc>**)**
ACTION_INFR_CEnableCilc()
ACTION_INFR_CDisableCilc()
ACTION_INFR_CGetCilc(IN <x>, **OUT** <v_adc, **OUT** <ctr>**)**
ACTION_INFR_CSetCilc(IN <x>, **IN** <crk>**)**
ACTION_INFR_CGetElDiagLshUp(IN <i>, **OUT** <cdn_diag>, **OUT** <err_diag>**)**
ACTION_INFR_CGetElDiagLshDown(IN <i>, **OUT** <cdn_diag>, **OUT** <err_diag>**)**
ACTION_INFR_CGetWrafControl(IN <i>, **OUT** <lv_lsl_deac>, **OUT** <state_err_vnlsl>, **OUT** <state_err_vglsl>, **OUT** <state_err_viplsl>, **OUT** <state_err_vrclsl>, **OUT** <state_err_iplsl>, **OUT** <state_err_lsl>, **OUT** <state_lsl_if_spi_rd>, **OUT** <state_lsl_if_conf_spi_rd>, **OUT** <frq_r_it_osc_lsl_if_spi_rd>, **OUT** <ctr_err_lsl_if_spi_wr>, **OUT** <ctr_err_lsl_if_spi_it_osc_ena>, **OUT** <lv_vplsl_lim>, **OUT** <lv_vnlsl_lim>**)**
ACTION_INFR_CSetWrafControl(IN <i>, **IN** <state_lsl_if_spi_wr>, **IN** <state_lsl_if_conf_spi_wr>, **IN** <frq_r_it_osc_lsl_if_spi_wr>, **IN** <lv_lsl_if_rst>, **IN** <lv_lsl_up_spi_com_inh>, **IN** <lv_ctr_err_lsl_if_spi_man_inc>**)**
ACTION_INFR_CSetCmdLsUp(IN <i>, **IN** <lv_swi_ls_up>**)**
ACTION_INFR_CSetCmdLsDown(IN <i>, **IN** <lv_swi_ls_down>**)**
ACTION_INFR_CSetHeatCoupling(IN <i>, **IN** <t_dly_trig_lsh_up>, **IN** <lamb_sp_fil_hom>**)**
ACTION_INFR_CGetHeatCoupling(IN <i>, **OUT** <lamb_sp_fil_delta_rise>, **OUT** <lamb_sp_fil_delta_fall>, **OUT** <vls_up_delta_rise>, **OUT** <vls_up_delta_fall>**)**

Description for actions:

ACTION_INFR_SetPwmLshUp(i, ducy **)**

This action sets the lambda heater upstream driver

Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
Ducy	IN	0...FFH	0...99.609375	0.390625	%
Up Heater driver PWM duty cycle					

ACTION_INFR_SetPwmLshDown(i, ducy **)**

This action sets the lambda heater downstream driver

Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
Ducy	IN	0…FFH	0…99.609375	0.390625	%
Down Heater driver PWM duty cycle					

ACTION_INFR_GetVAdcLsUp(i, v_adc **)**					
This action gets the raw value from the upstream oxygen sensor voltage.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
v_adc	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Raw value of upstream oxygen sensor voltage					

ACTION_INFR_GetVAdcLsDown(i, v_adc **)**					
This action gets the oxygen sensor voltage upstream					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
v_adc	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Downstream oxygen sensor voltage					

ACTION_INFR_GetVAdcTempLsUp(i, v_adc **)**					
This action sets the upstream lambda sensor temperature voltage					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
v_adc	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Upstream lambda sensor temperature voltage					

ACTION_INFR_EnableCilc()					
This action sets the activation of CILC (cylinder individuel lambda control)					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

ACTION_INFR_DisableCilc()					
This action sets the deactivation of CILC					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

ACTION_INFR_GetCilc(x, v_adc, ctr **)**					
Signal acquisition for CILC					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
X	IN	**0…7H**	**0…7**	**1**	**-**
logical index of cylinder to be acquired					
v_adc	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
upstream oxygen sensor acquisition (cylinder individual)					
Ctr	OUT	0…FFH	0…255	1	-
cylinder selective update counter for trigger event					

ACTION_INFR_SetCilc(x, crk **)**					
This action provide actual phase displacement for each cylinder. Remark: The maximum angle delay which can be realized by the BSW is 720°CRK. Larger values shall be calculated beginning with 0°CRK.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

X	IN	**0…7H**	**0…7**	**1**	**-**
logical index of cylinder to be changed					
Crk	IN	0…F0H	0…1440	6	°CRK
Value of phase displacement					

ACTION_INFR_GetElDiagLshUp(i, cdn_diag, err_diag **)**

This action reads the failure and the conditions information for each symptom from the infrastructure
The readout of the power stage is performed autonomous and the information is gathered.
When calling this Action the information inside the Infrastructure is reset after returning the OUT parameters.

Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
cdn_diag	OUT	0...7H	0...7	1	-
Bit 0: Set, if error symptom SCP (SYM_0) is detected Bit 1: Set, if error symptom SCG (SYM_1) is detected Bit 2: Set, if error symptom OC (SYM_2) is detected					
err_diag	OUT	0...7H	0...7	1	-
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

ACTION_INFR_GetElDiagLshDown(i**,** cdn_diag, err_diag **)**

This action reads the failure and the conditions information for each symptom from the infrastructure
The readout of the power stage is performed autonomous and the information is gathered.
When calling this Action the information inside the Infrastructure is reset after returning the OUT parameters.

Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
cdn_diag	OUT	0...7H	0...7	1	-
Bit 0: Set, if error symptom SCP (SYM_0) is detected Bit 1: Set, if error symptom SCG (SYM_1) is detected Bit 2: Set, if error symptom OC (SYM_2) is detected					
err_diag	OUT	0...7H	0...7	1	-
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

ACTION_INFR_GetWrafControl(i, lv_lsl_deac, state_err_vnlsl, state_err_vglsl, state_err_viplsl, state_err_vrclsl, state_err_iplsl, state_err_lsl, state_lsl_if_spi_rd, state_lsl_if_conf_spi_rd, frq_r_it_osc_lsl_if_spi_rd, ctr_err_lsl_if_spi_wr, ctr_err_lsl_if_spi_it_osc_ena, lv_vplsl_lim, lv_vnlsl_lim **)**

This action reads the values from the WRAF controller.

Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
lv_lsl_deac	OUT	0...1H	0...1	1	-
Boolean flag indicating OBDI signal fault pending					
state_err_vnlsl	OUT	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Nernst Cell positive pin.					
state_err_vglsl	OUT	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Virtual Ground pin.					
state_err_viplsl	OUT	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Pump Cell pin.					
state_err_vrclsl	OUT	0H 1H	NO_FAULT SCG	1	[-]

		2H	SCB		
Diagnosis status of the Calibration Resistance pin.					
state_err_iplsl	OUT	0H 1H 2H	NO_FAULT CURRENT_LOW CURRENT_HIGH	1	[-]
Diagnosis status of the Pumping Current.					
state_err_lsl	OUT	0H 1H	NO_FAULT FAULT	1	[-]
OBDI General Status					
state_lsl_if_spi_rd	OUT	0...FFH	0...255	1	[-]
Bit-field, mapping verified contents of WRAF controller Configuration Register					
state_lsl_if_conf_spi_rd	OUT	0...FFH	0...255	1	[-]
Content of the configuration register comprising sensor disconnection time delay and Icp value					
frq_r_it_osc_lsl_if_spi_rd	OUT	0...7FH	0...127	1	[-]
Content of the configuration register comprising Ri-measurement frequency and bit for pin selective disabling					
ctr_err_lsl_if_spi_wr	OUT	0...FFH	0...255	1	[-]
Counter indicating number of invalid INIT_REG_x write validation cycles which have been detected					
ctr_err_lsl_if_spi_it_osc_ena	OUT	0...FFH	0...255	1	[-]
Counter indicating number of invalid transmissions of “R_IT_OSC_ENA” bit					
lv_vplsl_lim	OUT	0…1H	0…1	1	[-]
Boolean flag indicating that Pump Cell voltage has reached its limits					
lv_vnlsl_lim	OUT	0...1H	0...1	1	[-]
Boolean flag indicating overvoltage at Nernst Cell positive pin (reverse Icp cut-off)					

ACTION_INFR_SetWrafControl(i, state_lsl_if_spi_wr, state_lsl_if_conf_spi_wr, frq_r_it_osc_lsl_if_spi_wr, lv_lsl_if_rst, lv_lsl_up_spi_com_inh, lv_ctr_err_lsl_if_spi_man_inc**)**					
This action sets the values of the WRAF controller					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
state_lsl_if_spi_wr	IN	0...FFH	0...255	1	-
Bit-field mapping of contents to be written to WRAF controller INIT_REG_1					
state_lsl_if_conf_spi_wr	IN	0...FFH	0...255	1	-
Bit-field mapping of contents to be written to WRAF controller INIT_REG_1					
frq_r_it_osc_lsl_if_spi_wr	IN	0...7FH	0...127	1	-
Mapping of contents to be written to WRAF controller INIT_REG_1					
lv_lsl_if_rst	IN	0...1H	0...1	1	-
Boolean flag to reset the WRAF controller					
lv_lsl_up_spi_com_inh	IN	0...1H	0...1	1	-
Inhibit bit for the SPI-communication related to electrical diagnosis on BSW level					
lv_ctr_err_lsl_if_spi_man_inc	IN	0...1H	0...1	1	-
BSW interface for manual increase of value of SPI communication error counter					

ACTION_INFR_SetCmdLsUp(i, lv_swi_ls_up **)**					
This action starts the internal resistance determination by "pumping" the the sensor.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
i	IN	0H 1H	1 2	1	**-**
Index of exhaust cylinder banks					
lv_swi_ls_up	IN	0H 1H	Switch off Switch on	1	-
Digital command parameter for the setting of pump current for binary upstream lambda sensor Remark: Switch on equal to activation (high or low level)					

ACTION_INFR_SetCmdLsDown(i, lv_swi_ls_down **)**					
This action starts the internal resistance determination by "pumping" the the sensor.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

i	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
lv_swi_ls_down	IN	0H 1H	Switch off Switch on	1	-
Digital command parameter for the setting of pump current for binary downstream lambda sensor Remark: Switch on equal to activation (high or low level)					

ACTION_INFR_SetHeatCoupling(i, t_dly_trig_lsh_up, lamb_sp_fil_hom **)**					
This action sets the value of the WRAF controller for the heater coupling diagnosis.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
t_dly_trig_lsh_up	IN	0...FFH	0...0.255	0.001	s
Time delay for acquisition of oxygen sensor signal and lambda model					
lamb_sp_fil_hom	IN	0...7FFFH	0...1.999939	6.10351573229163 E-5	-
output signal of the WRAF sensor model for homogeneous mode					

ACTION_INFR_GetHeatCoupling(i, lamb_sp_fil_delta_rise, lamb_sp_fil_delta_fall, vls_up_delta_rise, vls_up_delta_fall **)**					
This action gets the value from the WRAF controller for the heater coupling diagnosis.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
I	IN	0H 1H	1 2	1	-
Index of exhaust cylinder banks					
lamb_sp_fil_delta_rise	OUT	0...7FFFH	0...1.99993896484375	6.10351562E-5	-
Acquired lambda model value at the rising edge of heater PWM					
lamb_sp_fil_delta_fall	OUT	0...7FFFH	0...1.99993896484375	6.10351562E-5	-
Acquired lambda model value at the falling edge of heater PWM					
vls_up_delta_rise	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Acquired oxygen sensor voltage at the rising edge of heater PWM					
vls_up_delta_fall	OUT	0...7FFFH	0...4.99984741	1.52588E-4	V
Acquired oxygen sensor voltage at the falling edge of heater PWM					

FUNCTION DESCRIPTION:

General information:

a) **i = NC_CBK_EX_NR** (Number of exhaust gas cylinder banks)

b) **x = NC_CYL_NR** (Number of engine cylinders)

Depending on the value of NC_CBK_EX_NR and the states of the variable **NR_CONF_CBK_EX** following rules are defined which have to be used for all actions.

The symbolic item param_list represents the complete dedicated action parameters, exept the explicitly given one.

RULE 1: *%% used in general for upstream SET-ACTION*
IF NC_CBK_EX_NR = 2
THEN

```
    Case selection      NR_CBK_EX
    = 1:
            for i = 1:  ACTION_INFR_Command(IN < i = 1 >, param_list) =
                                    ACTION_INFR_CCommand( IN < i = 1 >, param_list)
            for i = 2:  no ACTION activity
    = 2:
            ACTION_INFR_Command(IN <i>, param_list) =
                        ACTION_INFR_CCommand ( IN < i>, param_list)
# ELSE
    ACTION_INFR_Command(IN <i>, param_list) =
                ACTION_INFR_CCommand ( IN < i>, param_list)
# ENDIF
```

RULE 2: *%% used in general for downstream SET-ACTION*

```
# IF        NC_CBK_EX_NR = 2
# THEN
    Case selection      NR_CONF_CBK_EX
    = 1:
            for i = 1:  ACTION_INFR_Command(IN < i = 1 >, param_list) =
                                    ACTION_INFR_CCommand( IN < i = 1 >, param_list)
            for i = 2:      no ACTION activity
    = 2:
            ACTION_INFR_Command(IN < i >, param_list) =
                        ACTION_INFR_CCommand( IN < i >, param_list)
    = 3:
            for i = 1:      no ACTION activity
            for i = 2: ACTION_INFR_Command(IN < i = 2 >, param_list) =
                                ACTION_INFR_CCommand( IN < 2 >, param_list)
    = 4:
            for i = 1: ACTION_INFR_Command(IN < i = 1 >, param_list) =
                                    ACTION_INFR_CCommand( IN < i = 1 >, param_list)
            for i = 2:      no ACTION activity
# ELSE
    ACTION_INFR_Command(IN < i >, param_list) =
                ACTION_INFR_CCommand( IN < i >, param_list)
# ENDIF
```

RULE 3: *%% used in general for upstrem GET-ACTION*

```
# IF        NC_CBK_EX_NR = 2
# THEN
    Case selection      NR_CBK_EX
    = 1:
            for i = 1:  ACTION_INFR_Command(IN < i = 1 >, param_list) =
                                ACTION_INFR_CCommand( IN < i = 1 >, param_list)
            for i = 2:  ACTION_INFR_Command(IN < i = 2 >, param_list) =
                                ACTION_INFR_CCommand( IN < i = 1 >, param_list)
    = 2:
            ACTION_INFR_Command(IN < i >, param_list) =
                        ACTION_INFR_CCommand( IN < i>, param_list)
# ELSE
```

```
        ACTION_INFR_Command(IN < i >, param_list) =
                    ACTION_INFR_CCommand( IN < i >, param_list)
# ENDIF
```

RULE 4: *%% used in general for downstream GET-ACTION*

```
# IF        NC_CBK_EX_NR = 2
# THEN
    Case selection        NR_CONF_CBK_EX
    = 1:
        for i = 1:  ACTION_INFR_Command(IN < i = 1 >, param_list) =
                        ACTION_INFR_CCommand( IN < i = 1 >, param_list)
        for i = 2:  ACTION_INFR_Command(IN < i = 2 >, param_list) =
                        ACTION_INFR_CCommand( IN < i = 1 >, param_list)
    = 2:
        ACTION_INFR_Command(IN < i >, param_list) =
                        ACTION_INFR_CCommand( IN < i>, param_list)
    = 3:
        for i = 1: ACTION_INFR_Command(IN < i = 1 >, param_list) =
                        ACTION_INFR_Command(IN < i = 1 >, OUT < VLS_NS[1] >)
                        % physical value of VLS_NS shall be output as VLS_DOWN[1]
        for i = 2: ACTION_INFR_Command(IN < i = 2 >, param_list) =
                        ACTION_INFR_CCommand( IN < i = 2 >, param_list)
    = 4:
        for i = 1: ACTION_INFR_Command(IN < i = 1 >, param_list) =
                        ACTION_INFR_CCommand( IN < i =  1 >, param_list)
        for i = 2: ACTION_INFR_Command(IN < i = 2 >, param_list) =
                        ACTION_INFR_Command(IN < i = 2 >, OUT < VLS_NS[1] >)
                        % physical value of VLS_NS shall be output as VLS_DOWN[2]
# ELSE
    ACTION_INFR_Command(IN < i >, param_list) =
                ACTION_INFR_CCommand( IN < i >, param_list)
# ENDIF
```

RULE 5:

```
ACTION_INFR_Command(param_list)  =  ACTION_INFR_CCommand(param_list)
```

RULE 6:

%% used for upstream Heater diagnosis with ATIC39.
*%% relies on the assumption that a **call for bank 1 made immediately before the call for bank 2.***

```
# IF        NC_CBK_EX_NR = 2
# THEN
    Case selection        NR_CBK_EX
    = 1:
        for i = 1:  ACTION_INFR_Command(IN < i = 1 >, param_list) =
                        ACTION_INFR_CCommand( IN < i = 1 >, param_list)
        for i = 2:  ACTION_INFR_Command(IN < i = 2 >, param_list) =
                        ACTION_INFR_Command( IN < i = 2 >,
                        OUT < cdn_diag = CDN_DIAG_LSH_UP[1] >,
                        OUT < err_diag = ERR_DIAG_LSH_UP[1] >)
```

```
    = 2:
            ACTION_INFR_Command(IN < i >, param_list) =
                            ACTION_INFR_CCommand( IN < i>, param_list)
# ELSE
    ACTION_INFR_Command(IN < i >, param_list) =
                    ACTION_INFR_CCommand( IN < i >, param_list)
# ENDIF
```

RULE 7:

%% used for downstream Heater diagnosis with ATIC39

*%% relies on the assumption that a **call for bank 1 made immediately before the call for bank 2**.*

```
# IF        NC_CBK_EX_NR = 2
# THEN
    Case selection        NR_CBK_EX
    = 1:
            for i = 1:  ACTION_INFR_Command(IN < i = 1 >, param_list) =
                                    ACTION_INFR_CCommand( IN < i = 1 >, param_list)
            for i = 2:  ACTION_INFR_Command(IN < i = 2 >, param_list) =
                                    ACTION_INFR_Command( IN < i = 2 >,
                                    OUT < cdn_diag = CDN_DIAG_LSH_DOWN[1] >,
                                    OUT < err_diag = ERR_DIAG_LSH_DOWN[1] >)
    = 2:
            ACTION_INFR_Command(IN < i >, param_list) =
                            ACTION_INFR_CCommand( IN < i>, param_list)
# ELSE
    ACTION_INFR_Command(IN < i >, param_list) =
                    ACTION_INFR_CCommand( IN < i >, param_list)
# ENDIF
```

Application conditions:

Initialisation: *at reset:* NR_CONF_CBK_EX = C_NR_CONF_CBK_EX

NR_CBK_EX = C_NR_CBK_EX

Recurrence: - (the assignment of NR_CONF_CBK_EX is only performed once during initialization)

Activation: always, under all conditions and states

Deactivation: ---

Formula section:

ACTION	RULE
ACTION_INFR_SetPwmLshUp(IN <i>, **IN** <ducy>**)**	**1**
ACTION_INFR_SetPwmLshDown(IN <i>, **IN** <ducy>**)**	**2**
ACTION_INFR_GetVAdcLsUp(IN <i>, **OUT** <v_adc>**)**	**3**
ACTION_INFR_GetVAdcLsDown(IN <i>, **OUT** <v_adc>**)**	**4**
ACTION_INFR_GetVAdcTempLsUp(IN <i>, **OUT** <v_adc>**)**	**3**
ACTION_INFR_EnableCilc()	**5**
ACTION_INFR_DisableCilc()	**5**
ACTION_INFR_GetCilc(IN <x>, **OUT** <v_adc, **OUT** <ctr>**)**	**5**
ACTION_INFR_SetCilc(IN <x>, **IN** <crk>**)**	**5**
ACTION_INFR_GetElDiagLshUp(IN <i>, **OUT** <cdn_diag>, **OUT** <err_diag>**)**	**6**
ACTION_INFR_GetElDiagLshDown(IN <i>, **OUT** <cdn_diag>, **OUT** <err_diag>**)**	**7**
ACTION_INFR_GetWrafControl(IN <i>, **OUT** <lv_lsl_deac>, **OUT** <state_err_vnlsl>, **OUT** <state_err_vglsl>, **OUT** <state_err_viplsl>, **OUT** <state_err_vrclsl>, **OUT** <state_err_iplsl>, **OUT** <state_err_lsl>, **OUT** <state_lsl_if_spi_rd>, **OUT** <state_lsl_if_conf_spi_rd>, **OUT** <frq_r_it_osc_lsl_if_spi_rd>, **OUT** <ctr_err_lsl_if_spi_wr>, **OUT** <ctr_err_lsl_if_spi_it_osc_ena>, **OUT** <lv_vplsl_lim>, **OUT** <lv_vnlsl_lim>**)**	**3**
ACTION_INFR_SetWrafControl(IN <i>, **IN** <state_lsl_if_spi_wr>, **IN** <state_lsl_if_conf_spi_wr>, **IN** <frq_r_it_osc_lsl_if_spi_wr>, **IN** <lv_lsl_if_rst>, **IN** <lv_lsl_up_spi_com_inh>, **IN** <lv_ctr_err_lsl_if_spi_man_inc>**)**	**1**
ACTION_INFR_SetCmdLsUp(IN <i>, **IN** <lv_swi_ls_up>**)**	**1**
ACTION_INFR_SetCmdLsDown(IN <i>, **IN** <lv_swi_ls_down>**)**	**2**
ACTION_INFR_SetHeatCoupling(IN <i>, **IN** <t_dly_trig_lsh_up>, **IN** <lamb_sp_fil_hom>**)**	**1**
ACTION_INFR_GetHeatCoupling(IN <i>, **OUT** <lamb_sp_fil_delta_rise>, **OUT** <lamb_sp_fil_delta_fall>, **OUT** <vls_up_delta_rise>, **OUT** <vls_up_delta_fall>**)**	**3**

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NR_CONF_CBK_EX	-	0...FFH	0...255	1	[-]
Define the technical structure of the exhaust bank with its components					
C_NR_CBK_EX	-	1...4H	1...4	1	[-]
Define the number of exhaust gas cylinder banks					

- *The configuration calibration data C_NR_CONF_CBK_EX define the hardware configuration of the exhaust gas as describe below:*
  - = 1: standard one bank
  - = 2: standard two banks
  - = 3: two bank system in a Y-configuration without lambda sensor downstream at position bank 1 and with one NOx-sensor after the NOx catalyst (NC_NOX_SENS_CONF = 1).
  - = 4: two bank system in a Y-configuration without lambda sensor downstream at position bank 2 and with one NOx-sensor after the NOx catalyst (NC_NOX_SENS_CONF = 1).

16.4 Wide Range Air-Fuel sensor interface (BSW) for the ATIC42

16.4.1 Acquire raw data from the Infrastructure (BSW)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LSL_DEAC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating OBDI signal fault pending					
STATE_ERR_VNLSL[NC_CBK_EX_NR]	V/O	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Nernst Cell positive pin.					
STATE_ERR_VGLSL[NC_CBK_EX_NR]	V/O	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Virtual Ground pin.					
STATE_ERR_VIPLSL[NC_CBK_EX_NR]	V/O	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Pump Cell pin.					
STATE_ERR_VRCLSL[NC_CBK_EX_NR]	V/O	0H 1H 2H	NO_FAULT SCG SCB	1	[-]
Diagnosis status of the Calibration Resistance pin.					
STATE_ERR_IPLSL[NC_CBK_EX_NR]	V/O	0H 1H 2H	NO_FAULT CURRENT_LOW CURRENT_HIGH	1	[-]
Diagnosis status of the Pumping Current.					
STATE_ERR_LSL[NC_CBK_EX_NR]	V/O	0H 1H	NO_FAULT FAULT	1	[-]
OBDI General Status					
STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Bit-field, mapping verified contents of ATIC42 Configuration Register					
STATE_LSL_IF_CONF_SPI_RD[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Content of the configuration register comprising sensor disconnection time delay and Icp value					
FRQ_R_IT_OSC_LSL_IF_SPI_RD[NC_CBK_EX_NR]	V/O	0...7FH	0...127	1	[-]
Content of the configuration register comprising Ri-measurement frequency and bit for pin selective disabling					
CTR_ERR_LSL_IF_SPI_WR[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Counter indicating number of invalid INIT_REG_x write validation cycles which have been detected					
CTR_ERR_LSL_IF_SPI_IT_OSC_ENA[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter indicating number of invalid transmissions of "R_IT_OSC_ENA" bit					
LV_VPLSL_LIM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that Pump Cell voltage has reached its limits					
LV_VNLSL_LIM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating overvoltage at Nernst Cell positive pin (reverse Icp cut-off)					

Input data:

LV_LSL_UP_SPI_COM_INH[NC_CBK_EX_NR]	LV_CTR_ERR_LSL_IF_SPI_MAN_INC[NC_CBK_EX_NR]	STATE_LSL_IF_CONF_SPI_WR[NC_CBK_EX_NR]	NC_CBK_EX_NR
FRQ_R_IT_OSC_LSL_IF_SPI_WR[NC_CBK_EX_NR]	STATE_LSL_IF_SPI_WR[NC_CBK_EX_NR]	LV_LSL_IF_RST[NC_CBK_EX_NR]	LV_IGK

Import actions:

ACTION_INFR_GetWrafControl(IN <i>, **OUT** <lv_lsl_deac>, **OUT** <state_err_vnlsl>, **OUT** <state_err_vglsl>, **OUT** <state_err_viplsl>, **OUT** <state_err_vrclsl>, **OUT** <state_err_iplsl>, **OUT** <state_err_lsl>, **OUT** <state_lsl_if_spi_rd>, **OUT** <state_lsl_if_conf_spi_rd>, **OUT** <frq_r_it_osc_lsl_if_spi_rd>, **OUT** <ctr_err_lsl_if_spi_wr>, **OUT** <ctr_err_lsl_if_spi_it_osc_ena>, **OUT** <lv_vplsl_lim>, **OUT** <lv_vnlsl_lim>**)**
This action reads the values from the WRAF controller
ACTION_INFR_SetWrafControl(IN <i>, **IN** <state_lsl_if_spi_wr>, **IN** <state_lsl_if_conf_spi_wr>, **IN** <frq_r_it_osc_lsl_if_spi_wr>, **IN** <lv_lsl_if_rst>, **IN** <lv_lsl_up_spi_com_inh>, **IN** <lv_ctr_err_lsl_if_spi_man_inc>**)**
This action sets the values of the WRAF controller

FUNCTION DESCRIPTION:

General information:

Application conditions:

Initialisation: at reset the actual input data will be send to and get from the infrastructure through calling the Actions

ACTION_INFR_GetWrafControl(

IN	i
OUT	LV_LSL_DEAC[i]
OUT	STATE_ERR_VNLSL[i]
OUT	STATE_ERR_VGLSL[i]
OUT	STATE_ERR_VIPLSL[i]
OUT	TATE_ERR_VRCLSL[i]
OUT	STATE_ERR_IPLSL[i]
OUT	STATE_ERR_LSL[i]
OUT	STATE_LSL_IF_SPI_RD[i]
OUT	STATE_LSL_IF_CONF_SPI_RD[i]
OUT	FRQ_R_IT_OSC_LSL_IF_SPI_RD[i]
OUT	CTR_ERR_LSL_IF_SPI_WR[i]
OUT	CTR_ERR_LSL_IF_SPI_IT_OSC_ENA[i]
OUT	LV_VPLSL_LIM[i]
OUT	LV_VNLSL_LIM[i])

ACTION_INFR_SetWrafControl(

IN	i
IN	STATE_LSL_IF_SPI_WR[i]
IN	STATE_LSL_IF_CONF_SPI_WR[i]
IN	FRQ_R_IT_OSC_LSL_IF_SPI_WR[i]
IN	LV_LSL_IF_RST[i]

IN LV_LSL_UP_SPI_COM_INH[i]

IN LV_CTR_ERR_LSL_IF_SPI_MAN_INC[i])

Recurrence:

T_SAMPLE = 10 ms.

Activation:

--

Deactivation:

--

Formula section:

has to be executed before all other functions of EGCP which have the same recurrence.

ACTION_INFR_GetWrafControl(...)

has to be executed after all other functions of EGCP which have the same recurrence.

ACTION_INFR_SetWrafControl(...)

16.4.2 Descripton of the interaction between infrastruture and the WRAF controller

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

The function shall describe the interface between the Wide Range Air-Fuel (WRAF) sensor, its controll device, the ATIC42 in its final development step (2nd Silicon, B5), and ECU.

The digital control part of the interface is realised in hardware by means of a Serial Peripheral Interface (SPI) bus between the ECU and the ATIC42. Latter acts as slave while the microcontroller acts as master. Via the SPI it shall be possible to carry out the following tasks:

- [Read from the two IC identity registers IDENT_REG_1[i] and IDENT_REG_2[i] – currently not implemented in generic SW!]

- Configure and initialise the 1-byte long **initialisation register**, which comprehends the following switches:

bit position	description
0	Set Ip measurement gain (16 or 8)
1	Enable/disable pumping current (Ip) offset measurement
2	Enable/disable reference pumping current (Icp)
3	Enable/disable Ip controller
4	Enable/disable Nernst Cell inner resistance (Ri) calibration
5	Start/stop oscillator for Ri measurement
6	Enable/disable sensor
7	Set Ri measurement range (range2, range1)

- Configure and initialise the **second configuration register** containing the 5 bit setting for the Icp-current value, and the 3-bit register for protection time prior to sensor disconnecting in case of electrical failure.

- Configure and initialise the 7-bit register containing Ri measurement frequency and the activation switch for selective pin disabling during sensor protection time.

- Read out the sensor pin electrical diagnostic register.

- Read out the general status register

- [Reset the ATIC42 – currently <u>not</u> implemented in generic SW!]

The individual timing requirements and the general description of the SPI may be found in the ATIC42 Component Specification and shall not be repeated here. Only those points that serve to clarify the interface between the ATIC42 and the microcontroller will be elaborated upon.

16.4.3 BSW Description

General remark regarding handling of read (RD), and write (WR) variables of SPI interface: Access to status bits which are part of STATE_LSL_IF... SPI_*RD[*i] variables has to be realized by directly addressing the individual bit "STATE_LSL_IF ... SPI_*RD*[i].bitX". The contents of STATE_LSL_IF ... SPI_*WR[*i] variables can either be accessed directly via "STATE_LSL_IF ... SPI_*WR[*i].bitX" or by corresponding logical variables, e.g. LV_LSL_ENA_LSL_IF[i].

16.4.3.1 SPI command sequencing

Immediately after a HW-reset or after ECU RESET, the standard command sequence stated below shall be executed. All the initialization information shall be sent once. Afterwards the sequence is repeated at every recurrence of BSW, i.e. every T_SAMPLE, containing up-to-date control information (command identifiers according to ATIC42 specification dated 24/07/02 are stated in brackets):

RD_DIAG_1(M1) + RD_DIAG_2(M2)
+ WR_CONF_INIT_1(F) + WR_CONF_INIT_2(I)

+ WR_INIT_R_FRQ_1(K) + WR_INIT_R_FRQ_2(L)
+ WR_INIT_1(C) + WR_INIT_2(D)
[+ RD_IDENT_1(A1) “dummy” command]

In order to receive the answer to the last write command sent, which is evaluated for communication monitoring purposes, WR_INIT_2(D), a “dummy” read command - has been added to the sequence.
Remark: With transmission of command “n”, response to command “n-1” is received in communication cycle “n”.

WR_INIT_2(D) command is sent at the end of each communication cycle to allow for all configuration data to be updated before setting sensor status (enable/disable).

16.4.3.2 Deactivation of WRAFsensor

Decoupling of WRAF sensor due to presence of electrical fault is handled by ASW/BSW interface (Appl Inc), 403B.

With LV_LSL_UP_SPI_COM_INH[i] = 1, ATIC42 SPI-responses related to electrical diagnosis status (contents of General Status Register and Pin Diagnosis Register) are no longer used to update contents of corresponding interface variables STATE_ERR_xxx[i], LV_LSL_VP_LIM[i], and LS_VNLSL_LIM[i]. The value of these variables thus preserves OBDI error status at the very moment of successful error debouncing for the remainder of the current driving cycle.

16.4.3.3 Reading the identification registers (currently not implemented in generic SW!)

If required, the identification registers can be read via the commands RD_IDENT_1(A1) and RD_IDENT_2(A2).

The command RD_IDENT_2(A2) reads the 8 MSB of the IC code while the command RD_IDENT_1(A1) reads the 3 LSB of the IC code, the 2 bit supplier index, and the 3 bit IC version number. The received information could be placed in a 16-bit output variable STATE_LSL_IF_IC_INFO[i] (V, 0...FFFFH, 0...65535, 1, -).

16.4.3.4 Reading the diagnostic registers

Diagnostic information regarding electrical sensor failures can be retrieved with the command sequence RD_DIAG_1(M1) + RD_DIAG_2(M2) which addresses the ATIC42 status registers.

Upon reading the general status register contents , the contents of both diagnosis registers are deleted. Therefore the two commands have to be performed in the appropriate sequence specified above: first the single pin diagnosis status has to be evaluated (RD_DIAG_1(M1)), then the general status register can be read (RD_DIAG_2(M2)) - deleting all electrical diagnosis information the very same moment!

The pin diagnosis register DIAG_REG_1 contains information regarding the type and location of electrical sensor faults. The following failure categories can be detected and for each fault a diagnosis bit is latched:

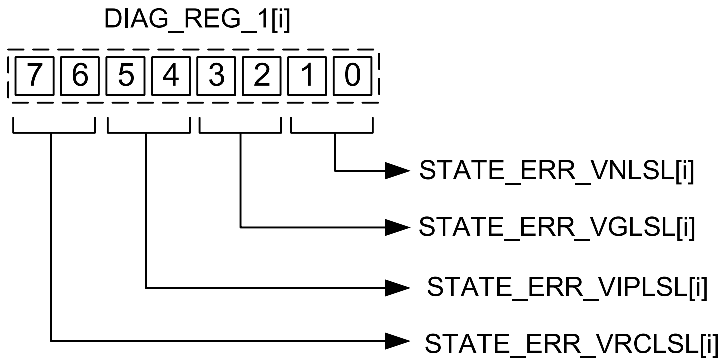
bit position	description
0	Short circuit to Ground (SCG) at pin RCD
1	Short circuit to Battery (SCB) at pin RCD
2	SCG at pin VIP
3	SCB at pin VIP
4	SCG at pin VG
5	SCB at pin VG
6	SCG at pin VN
7	SCB at pin VN

Two bits are associated with every sensor pin (RCD, VIP, VG, VN). Their combination provides four possible failure states (00 = no failure, 01 = SCG, 10 = SCB, 11 = presently not used).

DIAG_REG_2 contains information regarding general failure status, Pump Cell voltage and pumping current limit status, and overvoltage at VN pin. The following diagnoses can be evaluated :

1. General status of diagnosis.
2. Pump cell voltage at allowed limits. A special bit is set but not latched – as soon as the limitation is over, this bit is reset.
3. Pumping current (Ip) at its limits. A special bit is set but not latched – as soon as the violation disappears, this bit is reset.
4. Overvoltage at Nernst Cell positive pin which might impair voltage headroom for generation of reference pumping current. Therefore Icp generation is disabled by ATIC42 HW to prevent reverse Icp. Diagnosis bit is set but not latched.

As faults may occur while reading the diagnostic registers, it is advised to verify the contents of the diagnosis registers regularly. This shall be carried out by comparing the responses related to two consecutive RD_DIAG commands.



If after two consecutive RD_DIAG_x[i] command sequences a fault persists, the function shall decode the contents of both DIAG_REG_x[i] registers. The contents of the DIAG_REG_1[i] shall be placed in the appropriate output data variables, as it is shown in the picture above.

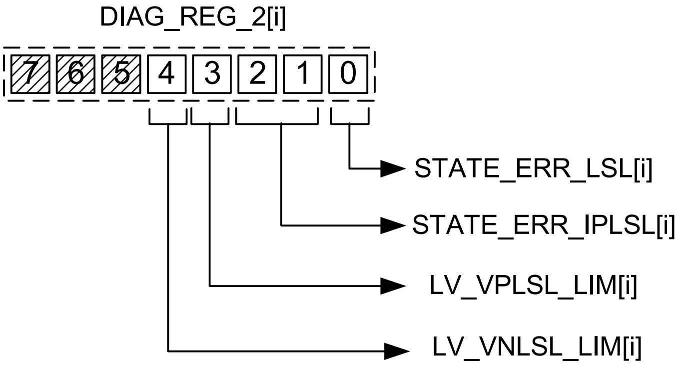
STATE_ERR_VNLSL[i] mirrors the diagnosis of Nernst cell positive pin, STATE_ERR_VGLSL[i] the diagnosis of virtual ground pin, STATE_ERR_VIPLSL[i] the diagnosis of Pump Cell positive pin, and STATE_ERR_VRCLSL[i] the diagnosis of the RCD pin.

The following diagnosis are decoded in each STATE-variable:

00H means no fault,

01H means short circuit to ground

02H means short circuit to battery.



The contents of the DIAG_REG_2[i] shall be placed in the appropriate output data variables:

STATE_ERR_LSL[i] gives the general status of the diagnosis in the ATIC42, where 00H means no fault and 01H means a fault is present. LV_VPLSL_LIM[i] shall be activated if the pump cell voltage reaches allowed bounds. STATE_ERR_IPLSL[i] gives a diagnosis when the pump cell current is out of range, where 00H means no fault, 01H means Ip current too low and 02H means Ip current too high. LV_VNLSL_LIM[i] signals detection of overvoltage at positive Nernst Cell pin which leads to disabling of reference pumping current in order to avoid reverse current flow.

Additionally, a Boolean flag LV_LSL_DEAC[i] shall be set if at any time a single valid RD_DIAG_x data byte returns an electrical fault at one sensor pin (excluding Pump Cell voltage at allowed bonds, VN overvoltage, and Ip limitation). The flag shall only be reset should the function receive two valid consecutive RD_DIAG command return data bytes indicating that the specific fault is no longer present.

16.4.3.5 Writing to the initialisation register (WR_INIT)

The application software (ASW) shall be permitted to write to both configuration registers of the ATIC42 to perform control of its different functionalities.

The interface to the ASW shall be facilitated by defining input data variables STATE_LSL_IF_SPI_WR[i], STATE_LSL_IF_CONF_SPI_WR[i], FRQ_R_IT_OSC_LSL_IF_SPI_WR[i] and the communication error counter CTR_ERR_LSL_IF_SPI_RD[i].

The BSW function shall service the ASW as follows:

The WRITE command sequence shall be carried out every recurrence, independently of any new information made available by the ASW via the interfaces STATE_LSL_IF_SPI_WR[i], STATE_LSL_IF_CONF_SPI_WR[i], FRQ_R_IT_OSC_LSL_IF_SPI_WR[i], and of the previous write command being incorrectly executed.

The function shall latch the current content of the three input data variables STATE_LSL_IF_xxx_SPI_WR[i](n) to STATE_LSL_IF_xxx_SPI_WR[i](n-1). Thus any further changes to the information during a WRITE cycle are prevented.

Next, the STATE_LSL_IF_xxx_SPI_WR[i](n-1) information shall be sent via the SPI to the ATIC42. The content written shall thereupon be verified via the corresponding answer which is received when sending the next command of the current SPI sequence.

Next, the STATE_LSL_IF_xxx_SPI_WR[i](n-1) information shall be sent via the SPI to the ATIC42. The content written shall thereupon be verified via the corresponding answer which is received when sending the next command of the current SPI sequence.

Should any WRITE command within current SPI transmission cycle be invalid or the written content not match the read content, the counter CTR_ERR_LSL_IF_SPI_WR[i] shall be incremented once.

For write validation procedure, sensor enable bit LV_LSL_ENA_LSL_IF[i] in the initialisation register is NOT considered because it is automatically reset by ATIC42 HW upon detection of an electrical sensor failure, regardless of the last transmitted SPI command. This asynchronuous change in bit value assignment does not represent a communication error.

In addition, bit6 of FRQ_R_IT_OSC_LSL_IF_SPI_xx[i] which represents the status of the “Pin Selective Disabling” feature shall also not be evaluated for SPI communication monitoring at present. Due to an ATIC42 HW error, command execution is performed but not monitored in ASIC response.

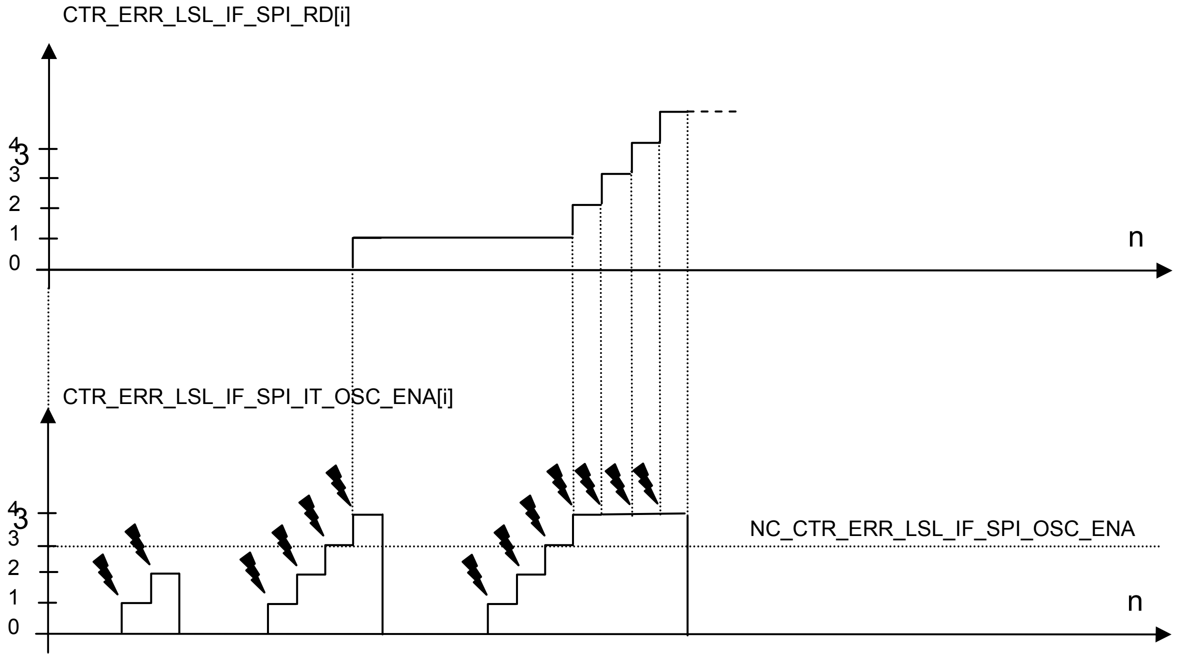
For manual increment of SPI communication error counter an interface to ASW (403B) is provided. This feature is introduced for purpose of SW test.

As the ATIC42 has a serial interface, the response to the currently transmitted command represents the answer to the previously executed command. As example, in the ATIC42 standard command sequence, the response received with command WR_CONF_INIT_1 corresponds to the answer to the preceding command RD_DIAG_2(M2). The BSW shall accordingly transfer the command responses to the correspondent output data variable (STATE_LSL_IF_SPI_RD[i], STATE_LSL_IF_CONF_SPI_RD[i], or FRQ_R_IT_OSC_LSL_IF_SPI _RD[i], resp.).

Up to ATIC42, 2nd Silicon, B5 step, a workaround for SPI transmission of “R_IT_OSC_ENA” bit has to be implemented. The ASIC shows random deviation between SPI request to set this bit and its physical reaction. If ATIC42 fails in switching on the oscillator for Ri determination, its ‘correct’ response does not fit the transmitted request.

Since this behaviour does not represent a communication error, its direct evaluation shall be excluded for the regular communication error counter CTR_ERR_LSL_IF_SPI_RD[i]. To be able to distinguish between HW based mismatch of request and response and SPI communication error, an additional cascaded error counter is introduced (CTR_ERR_LSL_IF_SPI_IT_OSC_ENA[i]) to deal with the “R_IT_OSC_ENA” bit. Its value is limited to NC_CTR_ERR_LSL_IF_SPI_OSC_ENA = 3. This threshold represents maximum number of sequential retries for setting “R_IT_OSC_ENA” bit before SPI communication error is recognized and reported to global error counter CTR_ERR_LSL_IF_SPI_RD[i].

Should transmission be successfull before supplementary counter reaches threshold value, it is reset. Once CTR_ERR_LSL_IF_SPI_IT_OSC_ENA[i] reaches threshold value, this value is kept until next successfull processing of “R_IT_OSC_ENA” bit and every additional erroneous transmission leads to direct incrementation of global error counter (see also Drawing 1).



Drawing 1: Relation between global and cascaded communication error counter

16.4.3.6 Performing a SW reset at the ATIC42 IC (currently not implemented in generic SW!)

The command RESET_IC(O) shall be executed if there exists a request to reset the ASIC via SW (LV_LSL_IF_RST[i] = 1). The activation of this flag as well as the priority definition should be handled in the “WRAF sensor interface (ASW-BSW)” Appl Inc, 403B. With the reset command the internal reset line is activated and all SPI registers are set to their default state provided by ATIC42 HW!

In order to deal with daisy chain communication comprehending multiple WRAF controllers (requiring identical number of commands provided to each controller but individual data contents), two alternative reset sequences can be thought of and should be discussed with the specific project prior to implementation:

- Repeated transmission of reset commands - equaling number of commands in regular sequence - to WRAF controller which is to be reset.
- Reset performance on all connected WRAF controllers.

<u>**Application conditions:**</u>

Initialisation:

NC_CTR_ERR_LSL_IF_SPI_OSC_ENA = 3

The following variable initialisation shall take place at the activation of the BSW:

CTR_ERR_LSL_IF_SPI_WR[i] = 0

CTR_ERR_LSL_IF_SPI_IT_OSC_ENA_i = 0

LV_LSL_DEAC[i] = 0

LV_VPLSL_LIM[i] = 0

LV_VNLSL_LIM[i] = 0

STATE_LSL_IF_SPI_RD[i] = 0

STATE_LSL_IF_CONF_SPI_RD[i] = 0

FRQ_R_IT_OSC_LSL_IF_SPI_RD[i] = 0

STATE_ERR_LSL[i] = NO FAULT (0H)

STATE_ERR_VNLSL[i] = NO FAULT (0H)

STATE_ERR_VGLSL[i] = NO FAULT (0H)

STATE_ERR_VIPLSL[i] = NO FAULT (0H)

STATE_ERR_VRCLSL[i] = NO FAULT (0H)

STATE_ERR_IPLSL[i] = NO FAULT (0H)

Note:

The interface variables STATE_LSL_IF_SPI_WR[i] and STATE_LSL_IF_CONF_SPI_WR[i] shall be initialized in the “WRAF sensor interface” (ASW – BSW). The corresponding ASW (“Oxygen sensor temperature”, 402A) shall do the initialization of FRQ_R_IT_OSC_LSL_IF_SPI_RD[i].

Recurrence:

T_SAMPLE = 10 ms.

Activation:

After every ECU RESET.

Deactivation:

--

Formula section:

```
If      LV_CTR_ERR_LSL_IF_SPI_MAN_INC[i] == 1
then    CTR_ERR_LSL_IF_SPI_WR[i] = CTR_ERR_LSL_IF_SPI_WR[i] + 1
endif
If      LV_CTR_ERR_LSL_IF_SPI_MAN_INC[i] == 1 -> 0
then    CTR_ERR_LSL_IF_SPI_WR[i] = 0
endif
```

Valid SPI-Commands:

The following SPI-commands consist of a 4 bit command plus 4 bit data partition each:

RD_IDENT_1(A1)	0H + 1H
RD_IDENT_2(A2)	0H + 2H
WR_INIT_1(C)	2H + D0D1D2D3 (see Table 3)

WR_INIT_2(D)	3H + D4D5D6D7 (see Table 3)
WR_CONF_INIT_1(F)	5H + D0D1D2D3 (see Table 4)
WR_CONF_INIT_2(I)	8H + D4D5D6D7 (see Table 4)
WR_ INIT_R_FRQ_1(K)	AH + D0D1D2D3
WR_ INIT_R_FRQ_2(L)	BH + D4D5D6D7
RD_DIAG_1(M1)	CH + 1H (read pin diagnosis register)
RD_DIAG_2(M2)	CH + 2H (read general status register)
RESET_IC(O)	FH + FH

Following a RD_IDENT_2(A2) command, the data byte of the returned message is defined as follows:

Bit Pos.	Bit Name	Description
0	STATE_LSL_IF_IC_INFO[i].bit_8	8 MSB of IC Code
1	STATE_LSL_IF_IC_INFO[i].bit_9	Ditto
2	STATE_LSL_IF_IC_INFO[i].bit_10	Ditto
3	STATE_LSL_IF_IC_INFO[i].bit_11	Ditto
4	STATE_LSL_IF_IC_INFO[i].bit_12	Ditto
5	STATE_LSL_IF_IC_INFO[i].bit_13	Ditto
6	STATE_LSL_IF_IC_INFO[i].bit_14	Ditto
7	STATE_LSL_IF_IC_INFO[i].bit_15	Ditto

Table 1: Data byte equivalence of the response to command RD_IDENT_2(A2) and the output variable STATE_LSL_IF_IC_INFO[i]

Similarly the answer to RD_IDENT_1(A1) is defined as:

Bit Pos.	Bit Name	Description
0	STATE_LSL_IF_IC_INFO[i].bit_0	Bits 0 through 2, 3 LSB of IC Code
1	STATE_LSL_IF_IC_INFO[i].bit_1	Ditto
2	STATE_LSL_IF_IC_INFO[i].bit_2	Ditto
3	STATE_LSL_IF_IC_INFO[i].bit_3	Bits 4 through 5, Supplier index
4	STATE_LSL_IF_IC_INFO[i].bit_4	Ditto
5	STATE_LSL_IF_IC_INFO[i].bit_5	Bits 6 through 7, Chip version number *)
6	STATE_LSL_IF_IC_INFO[i].bit_6	Ditto
7	STATE_LSL_IF_IC_INFO[i].bit_7	Ditto

Table 2: Data byte equivalence between the response to RD_IDENT_1 and STATE_LSL_IF_IC_INFO[i]

*) 000 for A1, A2, B1 samples; 001 for B2 samples!

Output data STATE_LSL_IF_SPI_RD[i] shall be defined to be a byte mirroring the content of the register INIT_REG_1, which represents a collection of individual bits as shown in the following table:

Bit Pos.	Bit Name	Description
D0	SWI_GAIN	Set Ip measurement gain to 8 (0) or 16 (1)
D1	VLS_OFS_ADJ_ENA	IP-offset-measurement (0) disable (1) enable
D2	ICPLSL_ENA	Icp current: (0) disable (1) enable
D3	IPLSL_CTL_ENA	Ip controller: (0) disable (1) enable
D4	R_IT_CAL_ENA	Ri calibration mode: (0) disabled, (1) enabled
D5	R_IT_OSC_ENA	Oscilator: (0) off, (1) on
D6	LSL_ENA	Enable / Disable linear lambda sensor
D7	R_IT_SWI_RNG	Ri measurement gain: (0) Range 1, (1) Range 2

***Table 3**:* Data byte for command WR_INIT_1(C) and WR_INIT_2(D)

Similarly the output data STATE_LSL_IF_CONF_SPI_RD[i] shall be defined to be a byte mirroring the content of the WR_CONF_INIT_1(F) and WR_CONF_INIT_2(I) as follows:

Bit Pos.	Bit Name	Description
0	ICPLSL[i].bit_0	Bits 0 through 4, Icp current
1	ICPLSL[i].bit_1	
2	ICPLSL[i].bit_2	
3	ICPLSL[i].bit_3	
4	ICPLSL[i].bit_4	
5	T_DLY_DI[i].bit_0	Bits 5 through 7, sensor protection time
6	T_DLY_DI[i].bit_1	
7	T_DLY_DI[i].bit_2	

Table 4: Data byte for command WR_CONF_INIT_1(F) and WR_CONF_INIT_2(I)

Following a RD_DIAG_1(M1) command, the data byte of the returned message is defined as follows:

Bit Pos.	Bit Name	Description
0	STATE_ERR_VNLSL[i].bit_0	VN -short circuit to gnd (0) no (1) yes
1	STATE_ERR_VNLSL[i].bit_1	VN -short circuit to battery (0) no (1) yes
2	STATE_ERR_VGLSL[i].bit_0	VG -short circuit to gnd (0) no (1) yes
3	STATE_ERR_VGLSL[i].bit_1	VG -short circuit to battery (0) no (1) yes
4	STATE_ERR_VIPLSL[i].bit_0	VIP -short circuit to gnd (0) no (1) yes
5	STATE_ERR_VIPLSL[i].bit_1	VIP -short circuit to battery (0) no (1) yes
6	STATE_ERR_VRCLSL[i].bit_0	RCD -short circuit to gnd (0) no (1) yes
7	STATE_ERR_VRCLSL[i].bit_1	RCD -short circuit to battery (0) no (1) yes

Table 5: Data byte equivalence between the register DIAG_REG_1[i] and the diagnosis output variables

Following a RD_DIAG_2(M2) command, the data byte of the returned message is defined as follows:

Bit Pos.	Bit Name	Description
0	STATE_ERR_LSL[i]	General diagnosis status: (0) no fault, (1) fault
1	STATE_ERR_IPLSL[i].bit_0	IP current low: (0) no (1) yes
2	STATE_ERR_IPLSL[i].bit_1	IP current high: (0) no (1) yes
3	LV_VPLSL_LIM[i]	VP at the allowed bounds: (0) no (1) yes
4	LV_VNLSL_LIM[i]	Overvoltage at VN pin: (0) no (1) yes

Table 6: Data byte equivalence between the register DIAG_REG_2[i] and the corresponding diagnosis output variables

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CTR_ERR_LSL_IF_SPI_OSC_ENA	-	0...FFH	0...255	1	[-]
Limit of counter indicating number of invalid transmissions of "R_IT_OSC_ENA" bit					

16.5 Exhaust gas composition – Requirements to I/O BSW

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_UP_RAW[NC_CBK_EX_NR]	O	0...3FFH	0...4.99511	4.8828e-3	[V]
Raw value of upstream oxygen sensor voltage					
VLS_DOWN[NC_CBK_EX_NR]	O	0...3FFH	0...4.99511	4.8828e-3	[V]
Downstream sensor voltage measured					
V_TEMP_LS_UP[NC_CBK_EX_NR]	O	0...3FFH	0...4.99511	4.8828e-3	[V]
Upstream lambda sensor temperature voltage					
VP_LS_UP_RAW	-	0...7FFFH	0...4.99984	0.1526e-3	[V]
Temporary storage for conversion of VLS_UP ADC value to resolution required in EGCP					
VP_LS_DOWN	-	0...7FFFH	0...4.99984	0.1526e-3	[V]
Temporary storage for conversion of VLS_DOWN ADC value to resolution required in EGCP					
VP_TEMP_LS_UP	-	0...7FFFH	0...4.99984	0.1526e-3	[V]
Temporary storage for conversion of V_TEMP_LS_UP ADC value to resolution required in EGCP					

Input data:

NC_CBK_EX_NR	LV_R_IT_REQ_LS_UP[NC_CBK_EX_NR]	LV_R_IT_REQ_LS_DOWN[NC_CBK_EX_NR]	NC_STATE_VLS_UP_SIG_ACQ
NC_STATE_LSL_UP_IF			

Import actions:

ACTION_INFR_GetVAdcLsUp(IN <i>, **OUT** <v_adc>**)**
Acquisition of the raw value from the upstream oxygen sensor voltage
ACTION_INFR_GetVAdcLsDown(IN <i>, **OUT** <v_adc>**)**
This action gets the oxygen sensor voltage downstream
ACTION_INFR_GetVAdcTempLsUp(IN <i>, **OUT** <v_adc>**)**
This action gets the oxygen sensor voltage upstream

FUNCTION DESCRIPTION:

General information:

Name	Acquisition recurrence	Acquisition precision	Notes
VLS_UP_RAW[i]	10ms	10bit	NC_STATE_VLS_UP_SIG_ACQ = 0
	1ms	10bit	NC_STATE_VLS_UP_SIG_ACQ = 1
V_TEMP_LS_UP[i]	10ms	10bit	-
VLS_DOWN[i]	10ms	10bit	-

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

If the lambda sensor signal shall be filtered prior to further processing NC_STATE_VLS_UP_SIG_ACQ has to be set to 1, otherwise to 0.

Description:

Voltages of lambda sensors offered by Basic SW are read from infrastructure in standard ECU06 resolution and transformed to application system specific variables.

Application conditions:

Activation: at every engine operating state
Deactivation: -
Update rate:
VLS_UP_RAW[i]: 10 ms for NC_STATE_VLS_UP_SIG_ACQ = 0
1 ms for NC_STATE_VLS_UP_SIG_ACQ = 1
VLS_DOWN[i] : 10 ms
V_TEMP_LS_UP[i]: 10 ms
Initialization: at reset: VLS_UP_RAW[i] = 0
VLS_DOWN[i] = 0
V_TEMP_LS_UP[i] = 0

Formula section:

```
# If      NC_STATE_LSL_UP_IF = 0          % Masking shall only be done for binary sensor
# then    If     LV_R_IT_REQ_LS_UP[i] = 0
          then   acquire VLS_UP_RAW[i]
                 ACTION_INFR_GetVAdcLsUp(
                             IN          i
                             OUT         VP_LS_UP_RAW)
                 VLS_UP_RAW[i] = VP_LS_UP_RAW
          else   $VLS\_UP\_RAW[i]_N = VLS\_UP\_RAW[i]_{N-1}$
          endif
# else    acquire VLS_UP_RAW[i]
          ACTION_INFR_GetVAdcLsUp(
                      IN          i
                      OUT         VP_LS_UP_RAW)
          VLS_UP_RAW[i] = VP_LS_UP_RAW
          acquire V_TEMP_LS_UP[i]
          ACTION_INFR_GetVAdcTempLsUp(
                      IN          i
                      OUT         VP_TEMP_LS_UP)
          V_TEMP_LS_UP[i] = VP_TEMP_LS_UP
```

```
# endif

If      LV_R_IT_REQ_LS_DOWN[i] = 0
then    acquire VLS_DOWN[i]
        ACTION_INFR_GetVAdcLsDown(
                IN      i
                OUT     VP_LS_DOWN)
        VLS_DOWN[i] = VP_LS_DOWN
else    $VLS\_DOWN[i]_{N} = VLS\_DOWN[i]_{N-1}$
endif
```

16.6 Acquisition of signal voltage and lambda model values for the Heater Coupling Diagnosis with UEGO Sensorsand ATIC42

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_SP_FIL_DELTA_RISE[NC_CBK_EX_NR]	O	0...7FFFH	0...1.99993	0.061e-3	[-]
Acquired lambda model value at the rising edge of heater PWM					
LAMB_SP_FIL_DELTA_FALL[NC_CBK_EX_NR]	O	0...7FFFH	0...1.99993	0.061e-3	[-]
Acquired lambda model value at the falling edge of heater PWM					
VLS_UP_DELTA_RISE[NC_CBK_EX_NR]	O	0...3FFH	0...4.99511	4.8828e-3	[V]
Acquired oxygen sensor voltage at the rising edge of heater PWM					
VLS_UP_DELTA_FALL[NC_CBK_EX_NR]	O	0...3FFH	0...4.99511	4.8828e-3	[V]
Acquired oxygen sensor voltage at the falling edge of heater PWM					
VP_LS_UP_DELTA_RISE	-	0...7FFFH	0...4.99984	0.1526e-3	[V]
Temporary storage for conversion of VLS_UP ADC readout at PWM rise to resolution required in EGCP					
VP_LS_UP_DELTA_FALL	-	0...7FFFH	0...4.99984	0.1526e-3	[V]
Temporary storage for conversion of VLS_UP ADC readout at PWM fall to resolution required in EGCP					

Input data:

VLS_UP[NC_CBK_EX_NR]	LSHPWM_UP[NC_CBK_EX_NR]	LAMB_SP_FIL_HOM[NC_CBK_EX_NR]	NC_CBK_EX_NR
T_DLY_TRIG_LSH_UP[NC_CBK_EX_NR]	LV_LSL_DIAG_LSH_LSL_UP_DEAC		

Import actions:

ACTION_INFR_SetHeatCoupling(IN <i>, **IN** <t_dly_trig_lsh_up>, **IN** <lamb_sp_fil_hom>**)**
This action sets the value of the WRAF controller for the heater coupling diagnosis.
ACTION_INFR_GetHeatCoupling(IN <i>, **OUT** <lamb_sp_fil_delta_rise>, **OUT** <lamb_sp_fil_delta_fall>, **OUT** <vls_up_delta_rise>, **OUT** <vls_up_delta_fall>**)**
This action gets the value from the WRAF controller for the heater coupling diagnosis

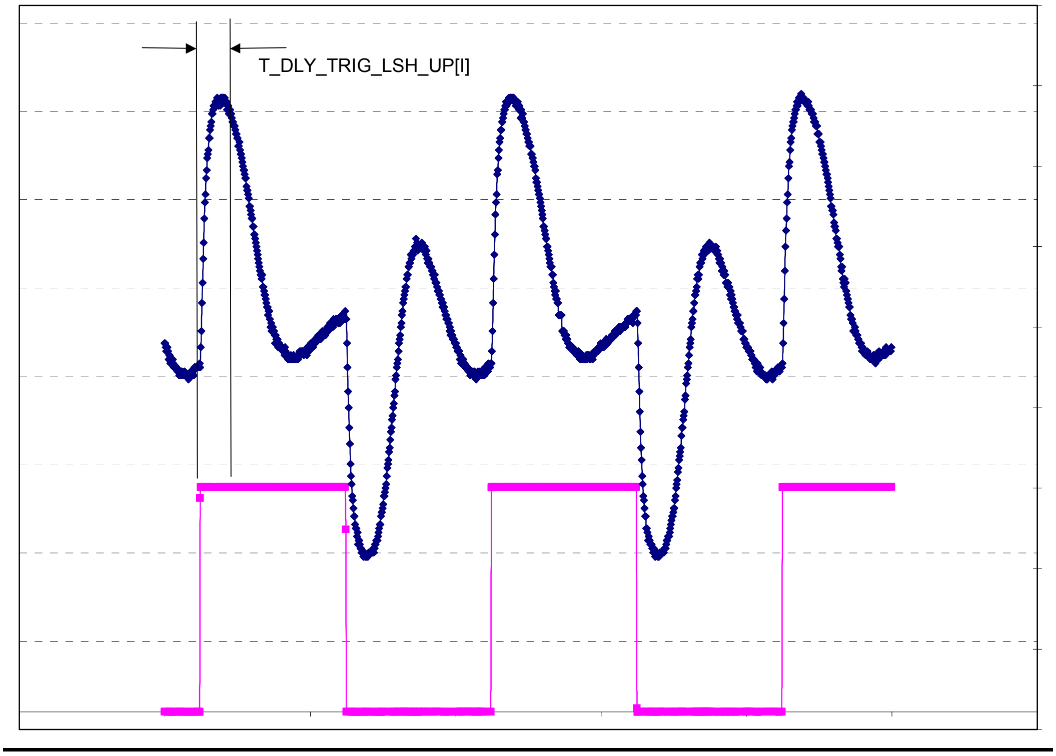
FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

This function shall deliver the inputs needed by the heater coupling diagnosis specific for UEGO Sensors controlled with the ATIC42. This functions describes how the upstream sensor signal and the lambda model shall be acquired synchronously with the heater signal.

Signal flow diagram:



Description:

In case a rising edge of the lambda sensor heater PWM is detected, the delay timer shall be started. After the delay timer expired its maximim value, the sensor signal VLS_UP[i] and the λ-model LAMB_SP_FIL[i] shall be acquired and transfered to the variables VLS_UP_DELTA_RISE[i] and LAMB_SP_FIL_DELTA_RISE[i] respectively.

Analogously, after a lambda sensor heater PWM falling edge occurred and the delay is expired the variables VLS_UP_DELTA_FALL[i] and LAMB_SP_FIL_DELTA_FALL[i] shall be updated.

The delay timer maximum value comprehends the value of T_DLY_TRIG_LSH_UP[i] updated at the last ***falling*** edge and shall be valid for one LSHPWM_UP[i] period at least. In case an edge comes before the time delay counter is expired, neither the lambda signal nor the lambda model shall be acquired. In this case the respective variables shall retain their old values until a new update is possible.

Also at the every falling edge the variables VLS_UP_DELTA_RISE[i], VLS_UP_DELTA_FALL[i], LAMB_SP_FIL_DELTA_RISE[i] and LAMB_SP_FIL_DELTA_FALL[i] shall be handed over to the ASW.

Application conditions:

Initialisation: at reset

VLS_UP_DELTA_RISE[i] = 0
VLS_UP_DELTA_FALL[i] = 0
LAMB_SP_FIL_DELTA_RISE[i] = 0
LAMB_SP_FIL_DELTA_FALL[i] = 0

Recurrence: 10 ms

Activation: **--**

Deactivation: **--**

Formula section:

```
If      LV_LSL_DIAG_LSH_LSL_UP_DEAC = 0
then    write data  to infrastructure
        ACTION_INFR_SetHeatCoupling(
                IN    i
                IN    T_DLY_TRIG_LSH_UP[i]
                IN    LAMB_SP_FIL_HOM[i])


        get data from infrastructure
        ACTION_INFR_GetHeatCoupling(
                IN    i
                OUT   LAMB_SP_FIL_DELTA_RISE[i]
                OUT   LAMB_SP_FIL_DELTA_FALL[i]
                OUT   VP_LS_UP_DELTA_RISE
                OUT   VP_LS_UP_DELTA_FALL)
        VLS_UP_DELTA_RISE[i] = VP_LS_UP_DELTA_RISE
        VLS_UP_DELTA_FALL[i] = VP_LS_UP_DELTA_FALL
endif
```

16.7 Signal acquisition for cylinder individual lambda control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_UP_CYL_SEL[NC_CYL_NR]	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
upstream oxygen sensor acquisition (cylinder individual)					
CTR_VLS_UP_CYL_SEL_TRIG[NC_CYL_NR]	V/O	0...FFH	0...255	1	[-]
cylinder selective update counter for trigger event					
VP_LS_CYL_SEL	-	0...7FFFH	0...4.99984	0.1526e-3	[V]
Temporary storage for conversion of infrastructure data to resolution required in EGCP					

Input data:

CRK_CYL_LAM[NC_CYL_NR]	NC_CYL_NR	LV_LAM_CYL_ENA[NC_CBK_EX_NR]

Import actions:

ACTION_INFR_EnableCilc()
This action sets the activation of CILC (cylinder individuel lambda control)
ACTION_INFR_DisableCilc()
This action sets the deactivation of CILC
ACTION_INFR_GetCilc(IN <x>, **OUT** <v_adc, **OUT** <ctr>**)**
Signal acquisition for CILC
ACTION_INFR_SetCilc(IN <x>, **IN** <crk>**)**
This action provide actual phase displacement for each cylinder

FUNCTION DESCRIPTION:

General information:

The function shall serve as a detector for a precise signal acquisition of the Wide Range Air-Fuel (WRAF) sensor in accordance with the identified cylinder. Therefore, the phase displacement, CRK_CYL_LAM[x] shall be considered. Since for dual-bank engines, the signal acquisition is dual-port, the phase shift shall be allocated to the right cylinder and cylinder bank. The phase displacement CRK_CYL_LAM[x] is computed once every combustion cycle (ASW) and shall be considered for all cylinders.

The calculation shall be done for x = 0 : NC_CYL_NR –1.

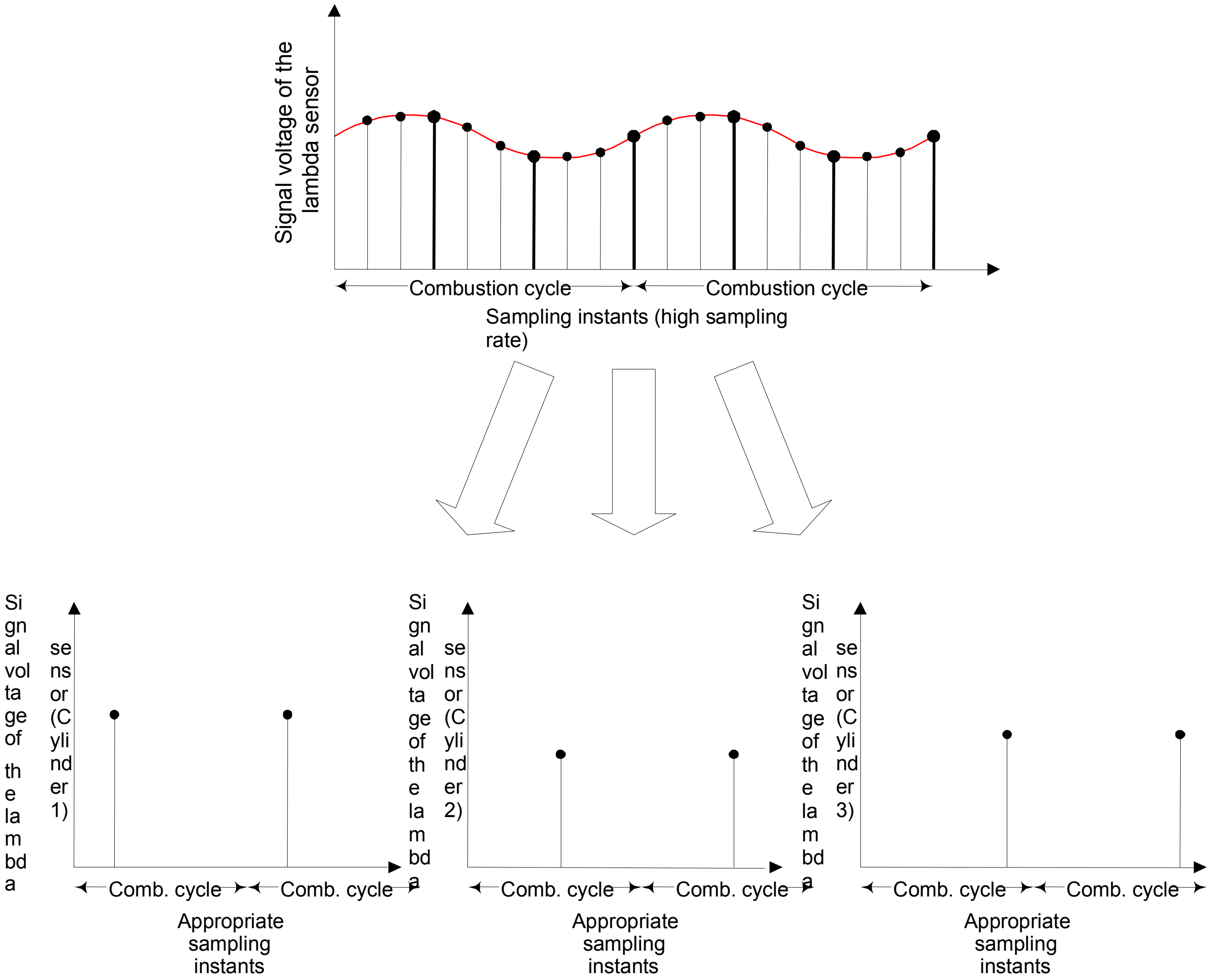


Figure 1: Cylinder individual signal acquisition of WRAF sensor

Application conditions:

Recurrence: segment synchronous, synchron to cylinder individual lambda control (ASW)

Activation: LV_LAM_CYL_ENA[i] = 0 -> 1

ACTION_INFR_EnableCilc()

Deactivation: LV_LAM_CYL_ENA[i] = 1 -> 0

ACTION_INFR_DisableCilc()

Formula section:

write data asynchronous to infrastructure

```
ACTION_INFR_SetCilc(
                IN      x
                IN      CRK_CYL_LAM[x])
```

get data asynchronous to infrastructure

```
ACTION_INFR_GetCilc(
                IN      x
                OUT     VP_LS_CYL_SEL
                OUT     CTR_VLS_UP_CYL_SEL_TRIG[x])
VLS_UP_CYL_SEL[x] = VP_LS_CYL_SEL
```

Description:

Signal acquisition and trigger event

In coordination with TDC the port selected signal voltage of the WRAF sensor shall be sampled (VLS_UP_CYL_SEL[x]) after the phase shift CRK_CYL_LAM[x] for each segment and the trigger counter for respective cylinder (CTR_VLS_UP_CYL_SEL_TRIG[x]) shall be incremented.

TDC **->** CRK_CYL_LAM[x] **->** VLS_UP_CYL_SEL[x] **->** CTR_VLS_UP_CYL_SEL_TRIG[x] ++

Remark:

For instance, see figure 2, which represents the sensor signal acquisition for a dual-bank engine (6 cylinders).

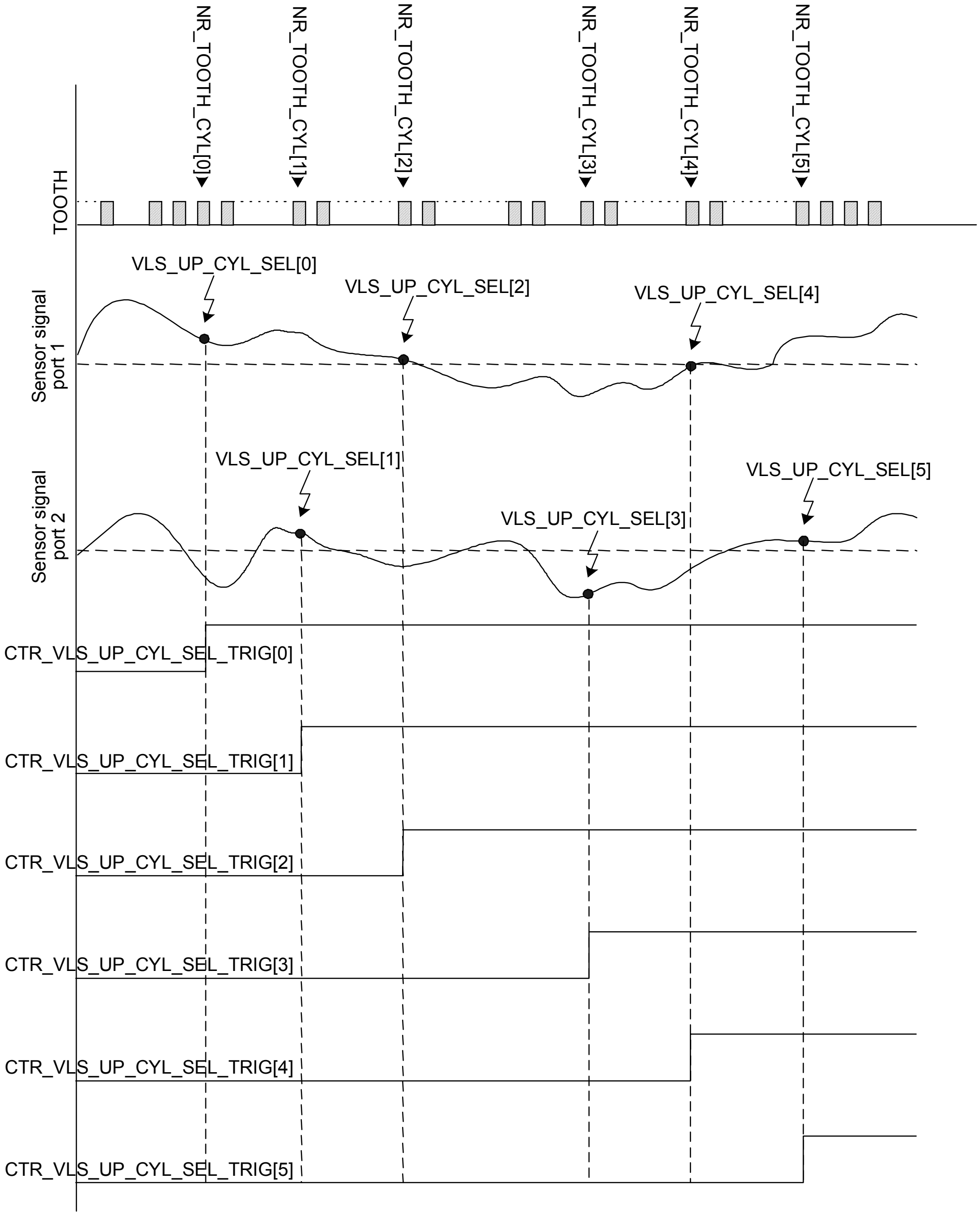


Figure 2: WRAF sensor signal acquisition of a dual-bank catalyst system (6 cylinders)

16.8 AGGR EGCP adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_1	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda signal value of the WRAF sensor of exhaust bank 1					
LAMB_2	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda signal value of the WRAF sensor of exhaust bank 2					
LAMB_MIN	V/O	0 ... 7FFFH	0 ... 31.999023	9.7656E-4	-
Minimum lambda in multiple-branched exhaust gas lines					
LAMB_SP_1	V/O	0 ... 7FFFH	0 ... 31.999023	9.7656E-4	-
Lambda set point of exhaust gas line - Bk1					
LAMB_SP_2	V/O	0 ... 7FFFH	0 ... 31.999023	9.7656E-4	-
Lambda set point of exhaust gas line – Bk2					
LAMB_SP_BEG_LS_ACT_TEST_1	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Initial value of lambda set point of exhaust bank 1					
LAMB_SP_BEG_LS_ACT_TEST_2	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Initial value of lambda set point of exhaust bank 2					
LV_LAMB_PULS_DYN_DIAG	V/O	0...1H	0...1	1	[-]
Request forced stimulation parameters specific for the WRAF sensor dynamic					
LV_UP_LS_1	V/O	0...1H	0...1	1	[-]
Logical variable for operative readiness of upstream oxygen sensor bank 1					
LV_UP_LS_2	V/O	0...1H	0...1	1	[-]
Logical variable for operative readiness of upstream oxygen sensor bank 2					
LV_DOWN_LS_1	V/O	0...1H	0...1	1	[-]
Logical variable for operative readiness of downstream oxygen sensor bank 1					
LV_DOWN_LS_2	V/O	0...1H	0...1	1	[-]
Logical variable for operative readiness of downstream oxygen sensor bank 2					
LV_ERR_LSH_DOWN_1	V/O	0...1H	0...1	1	[-]
Error flag for downstream heater diagnosis OBD I after debouncing					
LV_ERR_LSH_DOWN_2	V/O	0...1H	0...1	1	[-]
Error flag for downstream heater diagnosis OBD I after debouncing Bk2					
LV_ERR_LS_UP_1	V/O	0...1H	0...1	1	[-]
Final diagnostic of the upstream oxygen sensor ¿ Bk1					
LV_ERR_LS_UP_2	V/O	0...1H	0...1	1	[-]
Final diagnostic of the upstream oxygen sensor Bk2					
LV_ERR_LS_DOWN_1	V/O	0...1H	0...1	1	[-]
Final diagnostic of the downstream oxygen sensor Bk1					
LV_ERR_LS_DOWN_2	V/O	0...1H	0...1	1	[-]
Final diagnostic of the downstream oxygen sensor Bk2					
LV_ERR_LS_UP_DYN_VLD_1	V/O	0...1H	0...1	1	[-]
Boolean flag indicating the validation of correspondent symptom (transmitted to the error management) Bk1					
LV_ERR_LS_UP_DYN_VLD_2	V/O	0...1H	0...1	1	[-]
Boolean flag indicating the validation of correspondent symptom (transmitted to the error management) Bk2					

LV_ERR_OBD_LSH_DOWN_1	V/O	0...1H	0...1	1	[-]
Lambda Sensor Heater Error in Bk1					
LV_ERR_OBD_LSH_DOWN_2	V/O	0...1H	0...1	1	[-]
Lambda Sensor Heater Error in Bk2					
LV_ERR_VLS_DOWN_1	V/O	0...1H	0...1	1	[-]
Lambda Sensor Electrical in Bk1					
LV_ERR_VLS_DOWN_2	V/O	0...1H	0...1	1	[-]
Lambda Sensor Electrical Error in Bk2					
VLS_DOWN_1	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Downstream sensor voltage measured Bk1					
VLS_DOWN_2	-	0...3FFH	0...4.99511	4.8828e-3	[V]
Downstream sensor voltage measured Bk2					
LV_REQ_LAMB_SP_LS_ACT_TEST[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Lambda coordination request to change set point for active plausibility test					
LAMB_SP_LS_ACT_TEST[NC_CBK_EX_NR]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda set point for active plausibility test					
TEG_DYN_LS_UP[NC_CBK_EX_NR]	V/O	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Exhaust gas temperatures at the lambda sensor upstream catalyst					
TEG_DYN_LS_DOWN[NC_CBK_EX_NR]	V/O	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Exhaust gas temperatures at the lambda sensor downstream catalyst					
LV_TEMP_DEW_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag indicating that dew point is passed at lambda sensor up catalyst					
LV_TEMP_DEW_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag indicating that dew point is passed at lambda sensor down catalyst					
LV_INH_WIN_DET_DELTA_I_LAM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag to inhibit window calculation					
LAMB_MV	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Global lambda mean value					
LV_ERR_LSH_UP_1	O/V	0...1H	0...1	1	[-]
Error flag for upstream heater diagnosis OBD I after debouncing					
LV_ERR_LSH_UP_2	O/V	0...1H	0...1	1	[-]
Error flag for upstream heater diagnosis OBD I after debouncing ¿ Bk2					
LV_LSL_OFS_PULS_1	O/V	0...1H	0...1	1	[-]
State of the offset measurement for lambda stimulation Bk1					
LV_LSL_OFS_PULS_2	O/V	0...1H	0...1	1	[-]
State of the offset measurement for lambda stimulation Bk2					

Input data:

LV_LAMB_PLS_SWI_OFF[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]
LAMB_LS_UP_MIN	LAMB_SP_BEG_DIAG_LS_UP_DOWN[NC_CBK_EX_NR]
NC_CBK_EX_NR	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_LAMB_PLS_REQ_DYN_LSL_UP[NC_CBK_EX_NR]
LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_LS_UP_READY[NC_CBK_EX_NR]
LAMB_LS_UP_MV	LAMB_SP[NC_CBK_EX_NR]
LV_LS_DOWN_READY[NC_CBK_EX_NR]	LV_ERR_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_EL_LS_DOWN[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]
LV_LAMB_SP_REQ_DIAG_ACT[NC_CBK_EX_NR]	LAMB_SP_DIAG_LS_UP_DOWN[NC_CBK_EX_NR]
TEG_CAT_UP_MDL[NC_CBK_EX_NR]	TEG_CAT_DOWN_MDL[NC_CBK_EX_NR]
LV_TEG_MIN_THD[NC_CBK_EX_NR]	LV_TEG_CAT_DOWN_MIN_THD[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

Adaptation to aggregate environment.

i = 1 ... NC_CBK_EX_NR

Application conditions:

Initialisation:	0
Recurrence :	same recurrence as corresponding input data
Activation:	every engine operating state

Formula section:

LAMB_1 = LAMB_LS_UP[1]

LAMB_2 = LAMB_LS_UP[2]

LAMB_MIN = LAMB_LS_UP_MIN

LAMB_MV = LAMB_LS_UP_MV

LAMB_SP_BEG_LS_ACT_TEST_1 = LAMB_SP_BEG_DIAG_LS_UP_DOWN[1]

LAMB_SP_BEG_LS_ACT_TEST_2 = LAMB_SP_BEG_DIAG_LS_UP_DOWN[2]

LV_LAMB_PULS_DYN_DIAG = LV_LAMB_PLS_REQ_DYN_LSL_UP[1] **or** LV_LAMB_PLS_REQ_DYN_LSL_UP[2]

LV_UP_LS_1 = LV_LS_UP_READY[1]

LV_UP_LS_2 = LV_LS_UP_READY[2]

LV_ERR_LS_UP_1 = LV_ERR_LS_UP[1]

LV_ERR_LS_UP_2 = LV_ERR_LS_UP[2]

LV_ERR_LSH_UP_1 = LV_ERR_LSH_UP[1]

LV_ERR_LSH_UP_2 = LV_ERR_LSH_UP[2]

LV_ERR_LS_UP_DYN_VLD_1 = LV_ERR_DYN_VLD_LS_UP[1]

LV_ERR_LS_UP_DYN_VLD_2 = LV_ERR_DYN_VLD_LS_UP[2]

LAMB_SP_1 = LAMB_SP[1]

LAMB_SP_2 = LAMB_SP[2]

LV_DOWN_LS_1 = LV_LS_DOWN_READY[1]

LV_DOWN_LS_2 = LV_LS_DOWN_READY[2]

LV_ERR_LS_DOWN_1 = LV_ERR_LS_DOWN[1]

LV_ERR_LS_DOWN_2 = LV_ERR_LS_DOWN[2]

LV_ERR_LSH_DOWN_1 = LV_ERR_LSH_DOWN[1]

LV_ERR_LSH_DOWN_2 = LV_ERR_LSH_DOWN[2]

LV_ERR_OBD_LSH_DOWN_1 = LV_ERR_OBD_LSH_DOWN[1]

LV_ERR_OBD_LSH_DOWN_2 = LV_ERR_OBD_LSH_DOWN[2]

LV_ERR_VLS_DOWN_1 = LV_ERR_EL_LS_DOWN[1]
LV_ERR_VLS_DOWN_2 = LV_ERR_EL_LS_DOWN[2]
VLS_DOWN_1 = VLS_DOWN[1]
VLS_DOWN_2 = VLS_DOWN[2]
LV_REQ_LAMB_SP_LS_ACT_TEST[i] = LV_LAMB_SP_ REQ_DIAG_ACT[i]
LAMB_SP_LS_ACT_TEST[i] = LAMB_SP_DIAG_LS_UP_DOWN[i]
TEG_DYN_LS_UP[i] = TEG_CAT_UP_MDL[i]
TEG_DYN_LS_DOWN[i] = TEG_CAT_DOWN_MDL[i]
LV_TEMP_DEW_LS_UP[i] = LV_TEG_MIN_THD[i]
LV_TEMP_DEW_LS_DOWN[i] = LV_TEG_CAT_DOWN_MIN_THD[i]
LV_INH_WIN_DET_DELTA_I_LAM[i] = 0
LV_LSL_OFS_PULS_1 = LV_LAMB_PLS_SWI_OFF[1]
LV_LSL_OFS_PULS_2 = LV_LAMB_PLS_SWI_OFF[2]

16.9 Wide Range Air-Fuel Sensor Interface BSW-ASW for the ATIC42

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_LSL_IF_SPI_WR[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Bit-field mapping of contents to be written to ATIC42 INIT_REG_1					
STATE_LSL_IF_CONF_SPI_WR[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Mapping of contents to be written to ATIC42 INIT_REG_1					
LV_LSL_IF_SPI_RST_END[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating general status of ATIC42 initialisation after RESET					

Input data:

LV_SWI_GAIN_LSL_IF[NC_CBK_EX_NR]	LV_VLS_OFS_ADJ_ENA_LSL_IF[NC_CBK_EX_NR]	LV_ICPLSL_ENA_LSL_IF[NC_CBK_EX_NR]
LV_R_IT_SWI_RNG_LSL_IF[NC_CBK_EX_NR]	LV_R_IT_CAL_ENA_LSL_IF[NC_CBK_EX_NR]	LV_R_IT_OSC_ENA_LSL_IF[NC_CBK_EX_NR]
LV_LSL_ENA_LSL_IF[NC_CBK_EX_NR]	STATE_LSL_IF_CONF_SPI_RD[NC_CBK_EX_NR]	STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]
T_DLY_DI_LSL_IF[NC_CBK_EX_NR]	ICPLSL_LSL_IF[NC_CBK_EX_NR]	FRQ_R_IT_OSC_LSL_IF_SPI_RD[NC_CBK_EX_NR]
FRQ_R_IT_OSC_LSL_IF_SPI_WR[NC_CBK_EX_NR]	CTR_ERR_LSL_IF_SPI_WR[NC_CBK_EX_NR]	LV_IPLSL_CTL_ENA_LSL_IF[NC_CBK_EX_NR]
NC_CBK_EX_NR		

FUNCTION DESCRIPTION:

General information:

This function shall describe the interface between application software (ASW) and the basic software (BSW) that helps to initialise and to configure the ATIC42.

Furthermore, the function shall collect information about the status of the initialisations performed by itself and by other application software and send out a general status of the initialisation via the flag LV_LSL_IF_SPI_RST_END[i], which also indicates the readiness of the ATIC42.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

Description:

The function shall initialize the WRAF sensor interface **after RESET or upon erase of error management** by mirroring the contents of the calibration data C_STATE_LSL_IF_SPI_WR_INI into STATE_LSL_IF_SPI_WR[i] and C_STATE_LSL_IF_CONF_SPI_WR_INI into the variable STATE_LSL_IF_CONF_SPI_WR[i].

Table 1: Contents of the initialization variable C_STATE_LSL_IF_SPI_WR_INI

Bit	Bit Name	Meaning (value “1”/”0”)
0	SWI_GAIN	IP gain 16 / 8
1	VLS_OFS_ADJ_ENA	IP offset measurement ON / OFF
2	ICPLSL_ENA	Enable / disable Icp
3	IPLSL_CTL_ENA	Enable / disable Ip control loop (2nd silicon)
4	R_IT_CAL_ENA	Enable / disable Ri calibration
5	R_IT_OSC_ENA	Ri oscillator ON / OFF
6	LSL_ENA	Sensor enable / disable
7	R_IT_SWI_RNG	Ri measurement range 2 / range 1

After this the function shall bitwise compare the contents of STATE_LSL_IF_SPI_WR[i] and STATE_LSL_IF_SPI_RD[i], and of STATE_LSL_IF_CONF_SPI_WR[i] and STATE_LSL_IF_CONF_SPI_RD[i]. The function shall also check the initialization of the Ri-measurement frequency (done by the correspondent function) by bitwise comparing FRQ_R_IT_OSC_LSL_IF_SPI_WR[i] with FRQ_R_IT_OSC_LSL_IF_SPI_RD[i].

Remark:

For write validation procedure, sensor enable bit LV_LSL_IF_LSL_ENA_i in the initialisation register is NOT considered because it is automatically reset by ATIC42 HW upon detection of an electrical sensor failure, regardless of the last transmitted SPI command. This asynchronuous change in bit value assignment does not represent a communication error.

In addition, bit6 of FRQ_R_IT_OSC_LSL_IF_SPI_xx[i] which represents the status of the “Pin Selective Disabling” feature shall also not be evaluated for SPI communication monitoring at present. Due to an ATIC42 HW error, command execution is performed but not monitored in ASIC response.

Additionally the actual value of error counter CTR_ERR_LSL_IF_SPI_WR[i]$_{(N)}$ must be compared to its precedent value CTR_ERR_LSL_IF_SPI_WR[i]$_{(N-1)}$. If all comparisons show coincidence the flag LV_LSL_IF_SPI_RST_END[i] is set to 1.

The BSW function is defined elsewhere (Basic inputs & outputs, WRAF sensor interface). The BSW essentially observes the interface byte for changes. Should a change occur due to an ASW function, the new information is sent to the ATIC42 and the write cycle is then verified.

The function shall collect information from 8 logical variables and map these to the single byte variable STATE_LSL_IF_SPI_WR[i], which then acts as an interface to the BSW. The bit-field mapping is depicted in the following Table 2 and Figure 1.

Table 2: Bit-field mapping of contents of STATE_LSL_IF_SPI_WR[i]

Bit Pos.	Bit Name	Associated logical variable
0	SWI_GAIN	LV_SWI_GAIN_LSL_IF[i]
1	VLS_OFS_ADJ_ENA	LV_VLS_OFS_ADJ_ENA_LSL_IF[i]
2	ICPLSL_ENA	LV_ICPLSL_ENA_LSL_IF[i]
3	IPLSL_CTL_ENA	LV_IPLSL_CTL_ENA_LSL_IF[i]

4	R_IT_CAL_ENA	LV_R_IT_CAL_ENA_LSL_IF[i]
5	R_IT_OSC_ENA	LV_R_IT_OSC_ENA_LSL_IF[i]
6	LSL_ENA	LV_LSL_ENA_LSL_IF[i]
7	R_IT_SWI_RNG	LV_R_IT_SWI_RNG_LSL_IF[i]

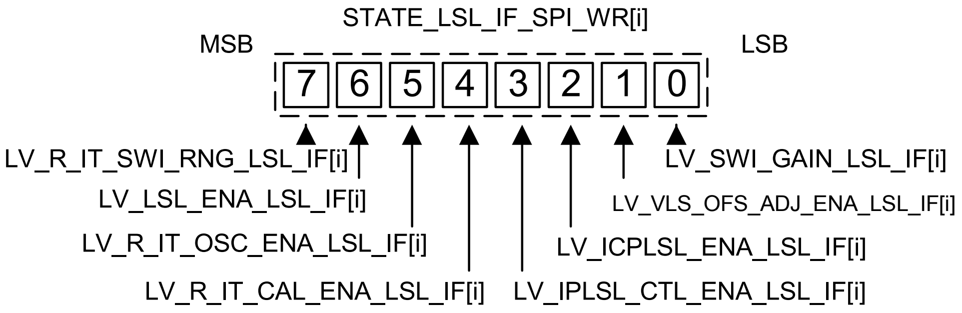


Figure 1: Bit-field mapping of contents of STATE_LSL_IF_SPI_WR[i]

Similarly the function shall also collect the value of the 5-bit variable ICPLSL_LSL_IF[i] and the value of the 3-bit variable T_DLY_DI_LSL_IF[i] and map these into the single byte variable STATE_LSL_IF_CONF_SPI_WR[i], which also acts as an interface to the BSW – see Figure 2.

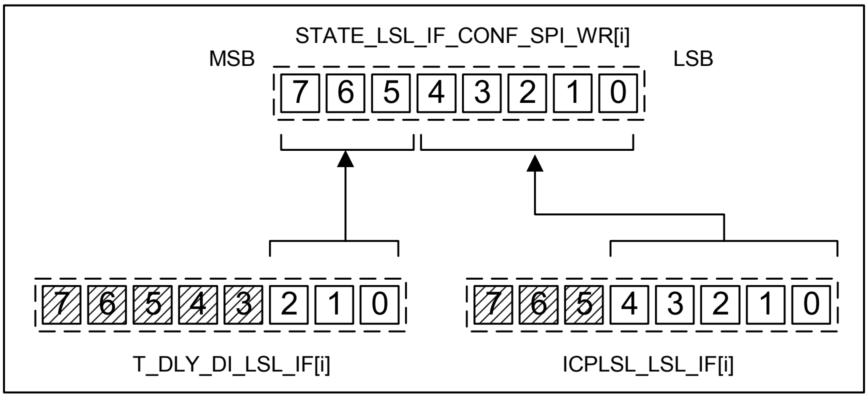


Figure 2: Bit-field mapping of contents of STATE_LSL_IF_CONF_SPI_WR[i]

<u>**Application conditions:**</u>

Initialization:

After a RESET or upon erase of error management the following initialization shall take place:
STATE_LSL_IF_SPI_WR[i] = C_STATE_LSL_IF_SPI_WR_INI
STATE_LSL_IF_CONF_SPI_WR[i] = C_STATE_LSL_IF_CONF_SPI_WR_INI
LV_LSL_IF_SPI_RST_END[i] = 0.
Store CTR_ERR_LSL_IF_SPI_WR_i$_{(N)}$ – 1.

Counter content is decremented at function initialization prior to storage to ensure that BSW is executed once the least before LV_LSL_IF_SPI_RST_END is set - and thus might falsely indicate execution of a successful write cycle for initialization. The simulated communication error (stored counter value differs from actual value) replaces time flow synchronization mechanism between ASW-BSW interface and BSW function which are executed at same recurrence rate. This workaround assures that BSW is able to perform one write cycle minimum regardless of contents of ATIC42 initialization registers.

Note: after a successfull initialization of the ATIC42 the flag LV_LSL_IF_SPI_RST_END shall be activated.

Recurrence:

T_SAMPLE = 10 ms.

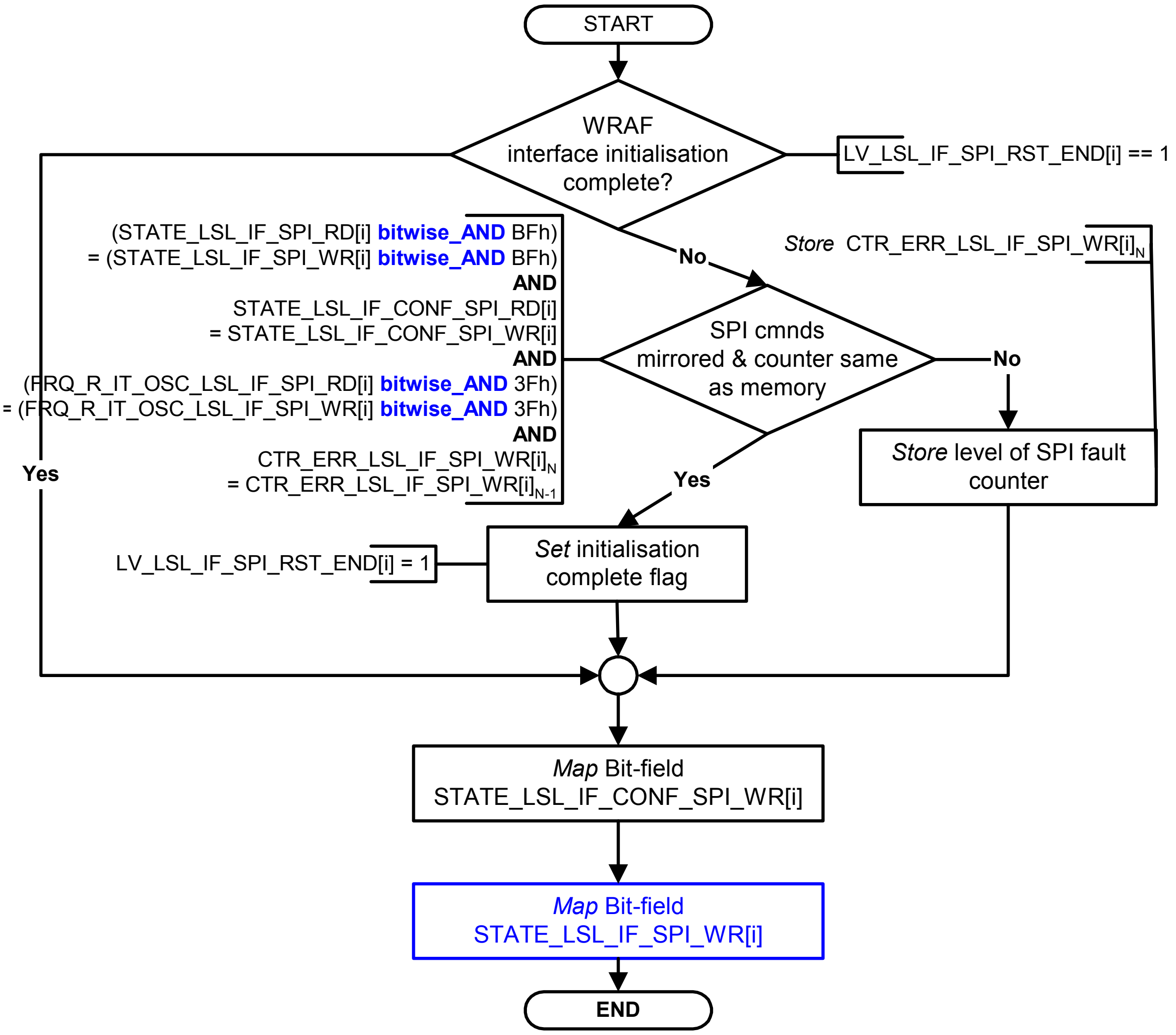
Activation:

In all Engine States

Deactivation:

--

Formula section:



Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_LSL_IF_SPI_WR_INI	1	0...FFH	0...255	1	[-]
Initialisation data of SPI register INIT_REG_1					
C_STATE_LSL_IF_CONF_SPI_WR_INI	1	0...FFH	0...255	1	[-]
Initialisation data of SPI register INIT_REG_2					

16.9.1 Application incidences for the ATIC42 ASW-BSW interface

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LSL_ENA_LSL_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag to enable/disable WRAF sensor connection					
LV_LSL_IF_RST[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag to reset the ATIC42					
LV_CTR_ERR_LSL_IF_SPI_MAN_INC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
BSW interface for manual increase of value of SPI communication error counter					
T_DLY_DI_LSL_IF[NC_CBK_EX_NR]	V/O	0...7H	0...7	1	[-]
Configuration of the Sensor Protection Time					

Input data:

NC_CBK_EX_NR	LV_IGK	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

Specification defines and initializes variables and boolean flags not defined and initialized elsewhere.

In case of an electric fault (incl. Open Circuit) which is detected by WRAF sensor electric diagnosis, sensor is decoupled as long as corresponding error flags are present.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1, for exhaust cylinder bank 1
 i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1, for single exhaust cylinder bank.

Description:

This function shall initialise the unused WRAF sensor interface flags.

Once OBDI sensor error is debounced (LV_ERR_EL_LSL_UP[i] == 1), WRAF sensor is decoupled (LV _LSL_ENA_LSL_IF [i] = 0) as long as error flag is present. Same action is taken once debounced OC failure is present (LV_ERR_OC_LSL_UP[i] == 1).

The following configuration data shall be used:

C_T_DLY_DI_LSL_IF = 010 = 2H (corresponds to 1ms)

Application conditions:

Initialisation:

The following variable initialisation shall take place *after RESET*.

T_DLY_DI_LSL_IF[i] = C_T_DLY_DI_LSL_IF

LV_LSL_IF_RST[i] = 0

LV_LSL_ENA_LSL_IF[i] = 1

The following variable initialisation shall take place *at LV_IGK = 0 -> 1, or upon erase of error management*.

LV_LSL_ENA _LSL_IF[i] = 1

NOTE: The logical variable LV_LSL_ENA_LSL_IF[i] is mapped in the interface byte STATE_LSL_IF_SPI_WR[i].
Its initialization shall be done according to the corresponding bit value of C_STATE_LSL_IF_SP_WR_INI in 'ATIC42 ASW<->BSW Interface', 4033.
T_DLY_DI_LSL_IF[i] is mapped in the configuration byte STATE_LSL_IF_CONF_SPI_WR[i].

Recurrence:

T_SAMPLE = 10ms

Activation:

In all engine states

Deactivation:

\---

Formula section:

LV_CTR_ERR_LSL_IF_SPI_MAN_INC[i] = LC_CTR_ERR_LSL_IF_SPI_MAN_INC

```
IF      LV_IGK == 1
THEN    IF      LV_ERR_EL_LSL_UP[i] == 1
        OR      LV_ERR_OC_LSL_UP[i] == 1
        THEN    LV_LSL_ENA_LSL_IF[i] = 0
        ELSE    LV _LSL_ENA_LSL_IF[i] = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_CTR_ERR_LSL_IF_SPI_MAN_INC	1	0...1H	0...1	1	[-]
Switch to manually increase value of SPI communication error counter within BSW					
C_T_DLY_DI_LSL_IF	1	0...7H	0...7	1	[-]
Configuration of sensor protection time					

16.10 Oxygen sensor ceramic temperature

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TTIP_MES_LS_UP[NC_CBK_EX_NR]	V/O	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
Oxygen sensor ceramic temperature					
R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...16383.75	0.25	[ohm]
Oxygen sensor element internal resistance					
STATE_TTIP_MES_LS_UP[NC_CBK_EX_NR]	V	0H 1H 2H 3H 4H	OFF MES_OFS ADJ_OFS ADJ_GAIN MES	1	[-]
State variable indicating current state of function					
V_OFS_MES_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFCH	0...4.99511	1.22E-03	[V]
Internal resistance determination circuit offset in resistance measurement mode					
V_OFS_ADJ_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFCH	0...4.99511	1.22E-03	[V]
Internal resistance determination circuit offset in resistance adjust mode					
FAC_GAIN_H_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...23.9996	3.66E-04	[-]
High gain of oxygen sensor internal resistance determination circuit					
FAC_GAIN_L_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...23.9996	3.66E-04	[-]
Low gain of oxygen sensor internal resistance determination circuit					
FRQ_R_IT_OSC_LSL_IF_SPI_WR[NC_CBK_EX_NR]	V/O	0...7FH	0...127	1	[-]
Variable indicating coded information for requested Ri determination oscillator frequency					
T_DLY_R_IT_IPLSL_CTL_ACT[NC_CBK_EX_NR]	V	0...FFH	0...2.55	0.01	[s]
Delay timer for freeze of Ri signal upon activation of Ip loop					
TCC_V_OFS_MES_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter incremented for every measured offset voltage in resistance measurement mode					
TCC_V_OFS_ADJ_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter incremented for every offset voltage in resistance adjust mode					
TCC_FAC_GAIN_H_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter incremented for every high gain adjust measurement					
TCC_FAC_GAIN_L_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter incremented for every low gain adjust measurement					
TCC_ERR_V_OFS_MES_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter incremented for every invalid offset voltage in resistance measurement mode					
TCC_ERR_V_OFS_ADJ_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter incremented for every invalid offset voltage in resistance adjust mode					
TCC_ERR_FAC_GAIN_H_R_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter incremented for every invalid high gain adjust measurement					
TCC_ERR_FAC_GAIN_L_R_LS_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter incremented for every invalid low gain adjust measurement					
LV_V_OFS_MES_VLD_R_IT_LS_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Boolean flag indicating offset in resistance measurement mode valid					
LV_V_OFS_ADJ_VLD_R_IT_LS_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating offset in resistance adjust mode valid					
LV_FAC_GAIN_H_VLD_R_IT_LS_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating measured high gain valid					
LV_FAC_GAIN_L_VLD_R_IT_LS_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating measured low gain valid					
LV_TTIP_MES_VLD_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating validity of oxygen sensor ceramic temperature TTIP_MES_LS_UP					
LV_V_REF_VLD_R_IT_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating internal resistance determination circuit references valid					
LV_R_IT_OSC_ENA_LSL_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag requesting WRAF interface internal oscillator to be enabled					
LV_R_IT_CAL_ENA_LSL_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag requesting WRAF interface internal resistance adjust mode to be enabled					
LV_R_IT_REQ_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that conditions have been met and resistance determination requested					
LV_R_IT_SWI_RNG_LSL_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag requesting WRAF interface to select internal resistance gain (range)					

Input data:

LV_IGK	V_TEMP_LS_UP[NC_CBK_EX_NR]	CTR_ERR_LSL_IF_SPI_WR[NC_CBK_EX_NR]
LV_LSL_OFS_ADJ_ACT[NC_CBK_EX_NR]	LV_LS_UP_READY_RAW[NC_CBK_EX_NR]	LV_LSL_IF_SPI_RST_END[NC_CBK_EX_NR]
LV_LS_UP_READY_CDN[NC_CBK_EX_NR]	STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]	FRQ_R_IT_OSC_LSL_IF_SPI_RD[NC_CBK_EX_NR]
LV_LSL_DEAC[NC_CBK_EX_NR]	LV_LS_UP_READY[NC_CBK_EX_NR]	LV_INH_TTIP_LS_UP[NC_CBK_EX_NR]
NC_CBK_EX_NR	LV_LSH_UP_MAN_ACT[NC_CBK_EX_NR]	

FUNCTION DESCRIPTION:

General information:

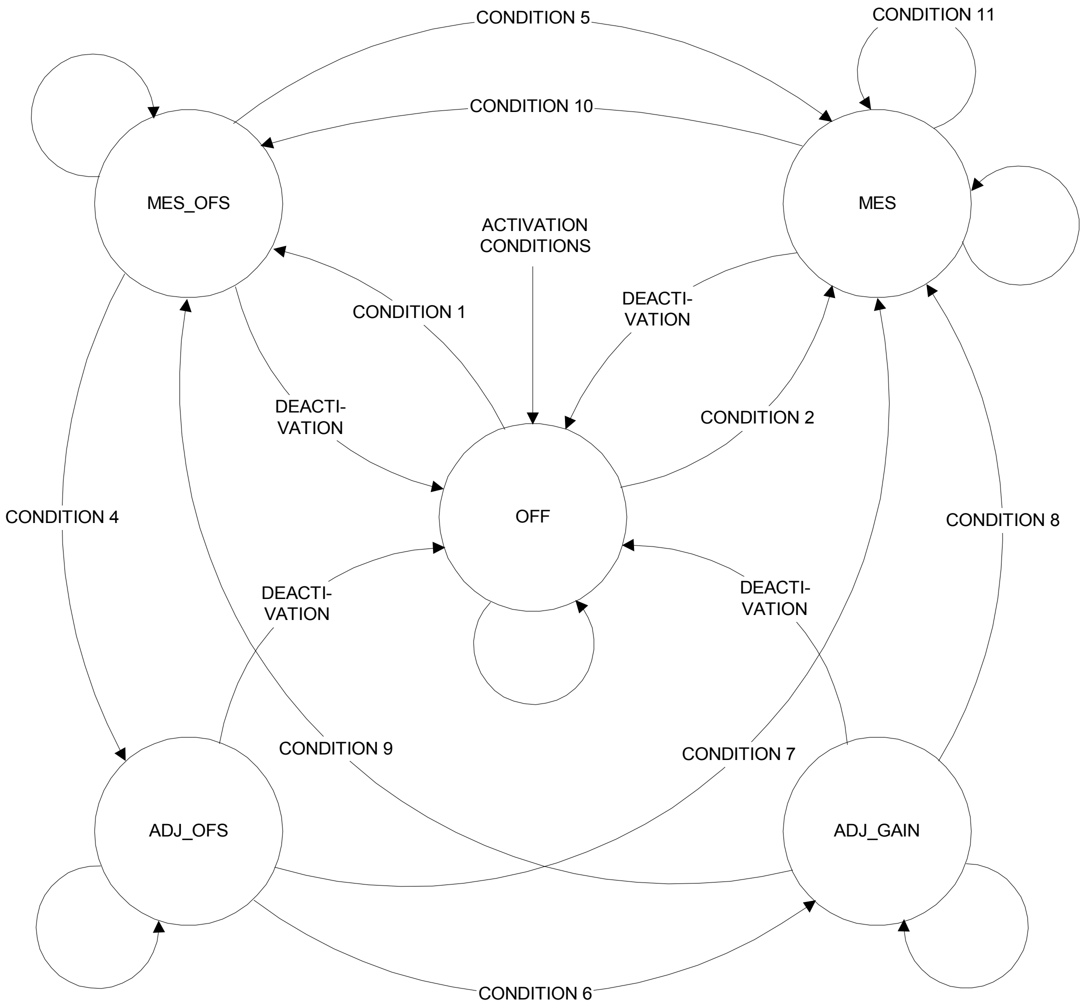
The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

The purpose of this function shall be to:

- Initialise the WRAF sensor interface with respect to all resistance determination parameters, i.e. switches and oscillator frequency.

- Switch between internal resistance ranges upon detection of operative readiness or when manually requested.
- Convert an acquired voltage (V_TEMP_LS_UP[i]) into an oxygen sensor element ceramic temperature and internal resistance.
- Periodically measure the offset and gain of the internal resistance determination circuit in order to minimise errors in the measured resistance and therefore ceramic temperature determination.

Signal flow diagram:



Description:

The function, once activated, shall provide the ceramic tip temperature of the oxygen sensor every 100 ms derived from the measured internal resistance. This temperature may then be used for temperature control and diagnosis purposes. In order to increase the system accuracy, the equation linking the measured voltage (measure of internal resistance) to the internal resistance may be compensated for drifts in offset and gain of the measurement circuit. The compensation shall be carried out periodically, when triggered by the input variable LV_LSL_OFS_ADJ_ACT[i] being set and when switching between internal resistance ranges. This former condition ensures that the oscillator, used to measure the internal resistance, is off when the lambda output offset measurement is carried out. The latter condition ensures that the different gain associated with the different range is re-measured.

The offsets shall be measured in both the measurement mode (oxygen sensor attached to WRAF sensor interface) and adjustment mode (reference calibration resistor attached to WRAF sensor interface) by switching the oscillator off and measuring the voltage output of the circuit. The gain shall be measured by switching to the adjust mode and taking the voltage reading with the calibration resistor in place. With reading and the knowledge of the value of the calibration resistor, stimulus resistor and previously measured offset in adjustment mode, the function shall compute the gain. This gain value of the internal amplifier is the same for both adjustment mode and measurement mode. However, the offsets may differ from one another due to CM (common mode) errors in the internal amplifier and due to the differing potentials at the VN amplifier input in adjustment mode (VGND) and measurement mode (VGND + VS). Hence the gain is calculated with the adjustment mode offset and the measured internal resistance is calculated with the measurement offset.

The **initialisation** of the function output data variables shall be carried out as follows:

- The SW initialisation of the WRAF sensor interface shall be carried out directly after a reset. FRQ_R_IT_OSC_LSL_IF_SPI_WR[i] shall be set to C_FRQ_R_IT_OSC_LSL_IF. The ASW interface function shall then determine whether the WRAF sensor interface has been successfully initialised by comparing the information returned from the WRAF interface ASIC with the corresponding initialisation request. Once all initialisations have been completed, the flag LV_LSL_IF_SPI_RST_END[i] is set indicating that the complete initialisation has been completed.
- The internal resistance R_IT_LS_UP[i] shall be initialised to its maximum value and the ceramic temperature TTIP_MES_LS_UP[i] to 0, i.e. cold sensor.
- The state machine shall be initialised to the OFF state and the temperature validity flag LV_TTIP_MES_VLD_LS_UP[i] shall be reset.
- The validity flags, LV_V_REF_VLD_R_IT_LS_UP[i], LV_V_OFS_MES_VLD_R_IT_LS_UP[i], LV_V_OFS_ADJ_VLD_R_IT_LS_UP[i], LV_FAC_GAIN_H_VLD_R_IT_LS_UP[i] & LV_FAC_GAIN_L_VLD_R_IT_LS_UP[i] shall be initialised dependent on the state of activation logical constant LC_V_REF_R_IT_LS_UP. Should the function be set to be active via this constant, then the flags shall be initialised with 0. Hence should the function be set to be inactive, the flags shall be set to 1, i.e. show that the offset and gain measurements are valid. This permits the function to be disabled via LC_V_REF_R_IT_LS_UP without affecting the functions that use the ceramic temperature information.
- The test cycle and frequency counters used to measure the number of reference measurements carried out (TCC_...) and the number of reference measurements out of bounds (FRC_...) shall be initialised after completion of engine start.
- The offset variables V_OFS_MES_R_IT_LS_UP[i] & V_OFS_ADJ_R_IT_LS_UP[i] and gain variables FAC_GAIN_H_R_IT_LS_UP[i] & FAC_GAIN_L_R_IT_LS_UP[i] shall be initialised with the appropriate C_... constant.

Upon **deactivation** of the function, when the engine start flag is reset, the function shall indicate to other functions that the ceramic temperature is no longer valid by setting the STATE_TTIP_MES_LS_UP[i] to OFF, resetting the general validity flag LV_TTIP_MES_VLD_LS_UP[i], the internal resistance measurement oscillator shall be switched off, the internal resistance measurement mode set to normal measurement mode and the resistance range shall be set the value LC_R_RNG_REQ_LS_UP_LSL_IF.

The function shall be **activated** with LV_IGK = 1, no inhibition bit set (LV_INH_TTIP_LS_UP[i] == 1), no electrical OBD fault have been detected for the WRAF sensor which leads to sensor decoupling (STATE_LSL_IF_SPI_RD[i] (*bit* LSL_ENA) == 1), and either the raw or conditional operative readiness has been detected (LV_LS_UP_READY_RAW[i] = 1 or LV_LS_UP_READY_CDN[i] = 1), or forced sensor pre-heating has been activated (LV_LSH_UP_MAN_ACT[i] == 1).

It shall be **deactivated** for LV_IGK = 0, at presence of function inhibition through corresponding application incidence, upon detection of an electrical OBD fault for the WRAF sensor resulting in sensor decoupling, or the raw and conditional operative readiness be reset together with passivated forced sensor pre-heating.

The validity of the ceramic temperature measurement shall be computed every recurrence and indicated by the Boolean flag LV_V_REF_VLD_R_IT_LS_UP[i]. This flag shall be set when both offset measurements have been determined to be valid, as indicated by LV_V_OFS_MES_VLD_R_IT_LS_UP[i] and LV_V_OFS_ADJ_VLD_R_IT_LS_UP[i], and when the gain has been determined to be valid, as indicated by LV_FAC_GAIN_H_VLD_R_IT_LS_UP[i] or LV_FAC_GAIN_L_VLD_R_IT_LS_UP[i] depending on the selected resistance range (LV_R_IT_SWI_RNG_LSL_IF[i]).

The function shall be implemented as a state machine with five states. The state of the state machine shall be indicated by the variable STATE_TTIP_MES_LS_UP[i]. The state machine may be described as follows:

STATE_TTIP_MES_LS_UP[i] = OFF

In state OFF, the ceramic temperature validity flag LV_TTIP_MES_VLD_LS_UP[i] shall be set to 0, indicating that the temperature shall not be used by other functions. The state machine may exit to one of either two states:

Exit to measurement mode offset measurement, **MES_OFS**: Shall occur when the SPI interface has been determined to have been initialised via flag LV_LSL_IF_SPI_RST_END[i] and should the calibration flag LC_V_REF_R_IT_LS_UP be set, thus permitting the calibration procedure to take place. Should these conditions be met, then *TIMER_1* & *TIMER_2* shall be reset, *TIMER_1* started. Additionally, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored, the state STATE_TTIP_MES_LS_UP[i] set to MES_OFS, the oscillator switched off via LV_R_IT_OSC_ENA_LSL_IF[i] = 0 and the measurement mode activated via LV_R_IT_CAL_ENA_LSL_IF[i] = 0.

Exit to normal resistance measurement mode, **MES**: Shall occur when the SPI interface has been determined to have been initialised via flag LV_LSL_IF_SPI_RST_END[i] and should the calibration flag LC_V_REF_R_IT_LS_UP be reset, thus preveting the calibration procedure from taking place. Should these conditions be met, then *TIMER_2* shall be reset, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored, the state STATE_TTIP_MES_LS_UP[i] set to MES, the oscillator switched on via LV_R_IT_OSC_ENA_LSL_IF[i] = 1 and the measurement mode activated via LV_R_IT_CAL_ENA_LSL_IF[i] = 0.

STATE_TTIP_MES_LS_UP[i] = MES_OFS

In state MES_OFS, the offset shall be determined as follows:

The function shall determine whether a fault in the SPI communication has occurred since the WRAF sensor interface control byte flags LV_R_IT_OSC_ENA_LSL_IF[i], LV_R_IT_CAL_ENA_LSL_IF[i], & LV_R_IT_SWI_RNG_LSL_IF[i] were programmed. This is facilitated by comparing the value of the SPI fault counter with the value stored prior to setting the flags. Additionally the status of the sensor coupling shall be determined. Should a communication fault have occurred either sending the information to the interface or should the information have changed in the interface control byte since programming, CTR_ERR_LSL_IF_SPI_WR[i] shall be incremented by the appropriate SPI communication function. This difference or the decoupling of the sensor, signalled by LV_LSL_DEAC[i] = 1 or STATE_LSL_IF_SP_RD[i].LSL_ENA = 0 shall cause the *TIMER_2* to be reset and the new value of the counter to be stored.

The SPI communication function operates an automatic retry strategy, thus if the desired contents could not be programmed or should the contents read back are not those programmed, then the SPI communication function shall continue to re-send the information until either it is successful or a debounced fault is recognised and further communication halted (check of FRQ_R_IT_OSC_LSL_IF_SP_RD[i]).

Should no SPI fault or decoupling of the sensor be determined to have occurred, the function shall verify that the desired contents have been programmed by reading the appropriate bits (IT_OSC_ENA, R_LS_CAL_ENA & R_LS_RNG) of the interface byte STATE_LSL_IF_SP_RD[i]. This byte mirrors the current contents of the WRAF sensor interface control byte. Due to the finite time that is required to program the WRAF sensor interface, the programming may not have taken place. In this instance the function shall await the next recurrence. If the programmed contents can be verified, the function shall wait until delay C_T_DLY_OFS_MES_R_IT_LS_UP has passed facilitated by *TIMER_2* and then take 4 V_TEMP_LS_UP[i] samples over the current and next 3 recurrences.

The absolute difference between the average of the 4 samples and the constant C_V_OFS_REF_R_IT_LS_UP shall then be compared to C_V_DIF_MAX_OFS_MES_R_IT_LS_UP to determine whether the measured offset is in bounds. If this is the case, the appropriate validity flag LV_V_OFS_MES_VLD_R_IT_LS_UP[i] set. If not the case, the frequency counter TCC_ERR_V_OFS_MES_R_IT_LS_UP[i] shall be incremented, indicating a faulty offset measurement, and LV_V_OFS_MES_VLD_R_IT_LS_UP[i] shall be reset. V_OFS_MES_R_IT_LS_UP[i] shall take on the value of the average, within bounds or not.

Test cycle counter TCC_V_OFS_MES_R_IT_LS_UP[i] shall be incremented, indicating that another measurement mode offset measurement has been carried out.

The state machine may exit to one of either two states:

Exit to adjustment mode offset measurement, **ADJ_OFS**: Shall occur when the offset measurement has been determined to have been completed. Should this condition be met, then the flag LV_R_IT_CAL_ENA_LSL_IF[i] shall be set, switching to adjust mode (calibration resistor), *TIMER_1* & *TIMER_2* shall be reset, *TIMER_1* started. Additionally, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored and the state STATE_TTIP_MES_LS_UP[i] set to ADJ_OFS.

Exit to normal resistance measurement mode, **MES**: Shall occur when the time limit C_T_MAX_V_OFS_MES_R_IT_LS_UP has been determined to have been exceeded, facilitated by *TIMER_1*. This provides a time out to ensure that the function does not lock up should SPI communication faults continuously prevent the offset measurement from being completed. Should these conditions be met, then the oscillator shall be switched on by setting flag LV_R_IT_CAL_ENA_LSL_IF[i], *TIMER_2* shall be reset, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored and the state STATE_TTIP_MES_LS_UP[i] set to MES.

STATE_TTIP_MES_LS_UP[i] = ADJ_OFS

In state ADJ_OFS, the offset shall be determined by exactly the same method as described for state MES_OFS, the only differences being:

- When checking that the programmed information has been correctly mirrored, bit STATE_LSL_IF_SP_RD[i].IT_OSC_ENA shall be 0, STATE_LSL_IF_SP_RD[i].R_LS_CAL_ENA shall be 1 and STATE_LSL_IF_SP_RD[i].R_LS_RNG shall equal LV_R_IT_SWI_RNG_LSL_IF[i].
- The *TIMER_2* delay shall be C_T_DLY_OFS_ADJ_R_IT_LS_UP.
- The absolute difference between the sample averages and constant C_V_OFS_REF_R_IT_LS_UP shall be compared to C_V_DIF_MAX_OFS_ADJ_R_IT_LS_UP.
- The adjustment mode offset variable shall be V_OFS_ADJ_R_IT_LS_UP[i], the validity flag shall be LV_V_OFS_ADJ_VLD_R_IT_LS_UP[i], the frequency counter TCC_ERR_V_OFS_ADJ_R_IT_LS_UP[i] and test cycle counter TCC_V_OFS_ADJ_R_IT_LS_UP[i].

The state machine may exit to one of either two states:

Exit to adjustment mode gain measurement, **ADJ_GAIN**: Shall occur when the offset measurement has been determined to have been completed and the lambda offset calibration has been completed, thus ensuring that the oscillator being off does not disturb the lambda output. Should these conditions be met, then the flag LV_R_IT_OSC_ENA_LSL_IF[i] shall be set, switching the oscillator on, *TIMER_1* & *TIMER_2* shall be reset and *TIMER_1* started. Additionally, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored and the state STATE_TTIP_MES_LS_UP[i] set to ADJ_GAIN.

Exit to normal resistance measurement mode, **MES**: Shall occur when the time limit C_T_MAX_V_OFS_ADJ_R_IT_LS_UP has been determined to have been exceeded, facilitated by *TIMER_1* providing a time out as described and the lambda output offset compensation has been completed as determined by LV_LSL_OFS_ADJ_ACT[i] = 0. Should these conditions be met, then the oscillator shall be switched on by setting flag LV_R_IT_CAL_ENA_LSL_IF[i], the mode set to measurement mode by resetting LV_R_IT_CAL_ENA_LSL_IF[i], *TIMER_2* shall be reset, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored and the state STATE_TTIP_MES_LS_UP[i] set to MES.

STATE_TTIP_MES_LS_UP[i] = ADJ_GAIN

In state ADJ_GAIN, the gain shall be determined by similarly to the method as described for state MES_OFS, the only differences being:

- When checking that the programmed information has been correctly mirrored, bit STATE_LSL_IF_SP_RD[i].IT_OSC_ENA shall be 1, STATE_LSL_IF_SP_RD[i].R_LS_CAL_ENA shall be 1 and STATE_LSL_IF_SP_RD[i].R_LS_RNG shall equal LV_R_IT_SWI_RNG_LSL_IF[i].
- The *TIMER_2* delay shall be C_T_DLY_FAC_GAIN_R_IT_LS_UP.
- After having taken the 4 samples, the gain shall be computed using the equation in the formula section. The gain calculated and the constants used in the calculation shall depend on the selected resistance range.

- The absolute difference between the gain and constant C_GAIN_TEMP_LS_UP shall be compared to C_GAIN_TEMP_LS_UP_DIF.
- The adjustment mode gain variable shall be FAC_GAIN_H_R_IT_LS_UP[i], the validity flag shall be LV_FAC_GAIN_H_VLD_R_IT_LS_UP[i], the frequency counter TCC_ERR_FAC_GAIN_H_R_LS_UP[i] and test cycle counter TCC_FAC_GAIN_H_R_IT_LS_UP[i] or FAC_GAIN_L_R_IT_LS_UP[i], the validity flag shall be LV_FAC_GAIN_L_VLD_R_IT_LS_UP[i], the frequency counter TCC_ERR_FAC_GAIN_L_R_LS_UP[i] and test cycle counter TCC_FAC_GAIN_L_R_IT_LS_UP[i] dependent on the selected resistance range (LV_R_IT_SWI_RNG_LSL_IF[i]).

The state machine may exit to one of either two states:

Exit to normal internal resistance measurement mode, **MES**: Shall occur when the gain measurement has been determined to have been successfully completed (LV_V_REF_VLD_R_IT_LS_UP[i] = 1) or when the time limit C_T_MAX_FAC_GAIN_R_IT_LS_UP has been determined to have been exceeded, facilitated by *TIMER_1* providing a time out as described previously. Should these conditions be met, then the mode shall be set to measurement mode by resetting LV_R_IT_CAL_ENA_LSL_IF[i], *TIMER_2* shall be reset, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored and the state STATE_TTIP_MES_LS_UP[i] set to MES.

Exit to measurement mode offset measurement, **MES_OFS**: Shall occur should the gain measurement have been unsuccessfully completed (LV_V_REF_VLD_R_IT_LS_UP[i] = 0) or if the lambda output offset compensation be requested (LV_LSL_OFS_ADJ_ACT[i] = 1). Should either of these conditions be met, then the oscillator shall be switched off via LV_R_IT_OSC_ENA_LSL_IF[i] = 0 and the measurement mode activated via LV_R_IT_CAL_ENA_LSL_IF[i] = 0, *TIMER_1* & *TIMER_2* shall be reset, *TIMER_1* started. Additionally, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored, the state STATE_TTIP_MES_LS_UP[i] set to MES_OFS.

STATE_TTIP_MES_LS_UP[i] = MES

In state MES, the internal resistance shall be determined as follows:

The function shall determine whether a fault in the SPI communication or if the sensor has been decoupled as described previously. Should either of these conditions be true, the ceramic temperature validity flag LV_TTIP_MES_VLD_LS_UP[i] shall be reset, thus suspending further use of the temperature information, *TIMER_2* shall be reset and the new value of the counter to be stored.

Should no SPI fault or decoupling of the sensor be determined to have occurred, function checks for <u>rising</u> edge of ATIC42 pump current control enable bit. To avoid erroneous Ri readings due to electrical disturbance on Ri measurement circuit during Ip activation, Ri value is frozen for calibratable delay time C_T_DLY_R_IT_IPLSL_CTL_ACT triggered by rising edge detection. Same holds true for <u>falling</u> edge. During signal freeze LV_TTIP_MES_VLD_LS_UP[i] shall be reset.

Next the function shall verify that the desired control byte contents have been programmed by reading the appropriate bits (IT_OSC_ENA = 1, R_LS_CAL_ENA = 0, R_LS_RNG = LV_R_IT_SWI_RNG_LSL_IF[i]) of the interface byte STATE_LSL_IF_SP_RD. If not the case the function shall await the next recurrence. If the programmed contents can be verified, the function shall wait until delay C_T_DLY_CLC_R_IT_LS_UP has passed facilitated by *TIMER_2* and then compute the internal resistance from the measurement mode offset, V_OFS_MES_R_IT_LS_UP[i], the average of up to 8 V_TEMP_LS_UP[i] samples, the value of the stimulus resistor, C_R_IT_REF_LS_UP and dependent on the value of LV_R_IT_SWI_RNG_LSL_IF[i] either FAC_GAIN_H_R_IT_LS_UP[i] or FAC_GAIN_L_R_IT_LS_UP[i].

The ceramic temperature validity flag LV_TTIP_MES_VLD_LS_UP[i] shall be set and the ceramic temperature TTIP_MES_LS_UP[i] shall be determined from the map IP_TTIP_MES_LS_UP.

The state machine may exit to one state:

Exit to state, measurement mode offset measurement, **MES_OFS**: Shall occur should the calibration flag LC_V_REF_R_IT_LS_UP be set, thus permitting the calibration procedure to take place and either when the lambda offset correction has been determined to have been requested, LV_LSL_OFS_ADJ_ACT[i] = 1 or should the operative readiness be detected or finally should a manual switching of the gain be requested. The non-operative readiness condition ensures that should any of the time-outs occur in the states MES_OFS, ADJ_OFS or ADJ_GAIN, and the operative readiness not yet have been detected, then the function may restart the calibration process. Should these conditions be met, then, dependent on whether a manual range switch was requested, the resistance range shall be set to the Boolean constant LC_R_RNG_REQ_LS_UP_LSL_IF or to the negation of LV_LS_UP_READY[i], the ceramic temperature validity flag LV_TTIP_MES_VLD_LS_UP[i] shall be reset, the oscillator shall be switched off via LV_R_IT_OSC_ENA_LSL_IF[i] = 0, *TIMER_1* & *TIMER_2* shall be reset, *TIMER_1* started. Additionally, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored, the state STATE_TTIP_MES_LS_UP[i] set to MES_OFS.

Should the calibration flag LC_V_REF_R_IT_LS_UP be reset, thus disabling the calibration procedure to take place but the operative readiness be detected or should a manual switching of the gain be requested then , dependent on whether a manual range switch was requested, the resistance range shall be set to the Boolean constant LC_R_RNG_REQ_LS_UP_LSL_IF or to the negation of LV_LS_UP_READY[i], the ceramic temperature validity flag LV_TTIP_MES_VLD_LS_UP[i] shall be reset, *TIMER_1* & *TIMER_2* shall be reset, *TIMER_1* started. Additionally, the current value of the SPI fault counter CTR_ERR_LSL_IF_SPI_WR[i] shall be stored, but the state STATE_TTIP_MES_LS_UP[i] shall **remain in MES**.

In addition to the aforementioned state conditions, the state machine may **exit to state OFF from all other states**. This shall occur should the function be deactivated. In this case certain variables may be reset as described previously in the section regarding the initialisation.

Application conditions:

Initialisation:

The following initialisation shall take place directly **after RESET**:

FRQ_R_IT_OSC_LSL_IF_SPI_WR[i] = C_FRQ_R_IT_OSC_LSL_IF

The following variable initialisation shall take place **after RESET, at LV_IGK = 0 -> 1**, or **upon erase of error management**:

TTIP_MES_LS_UP[i] = 0

R_IT_LS_UP[i] = 16383.75 (i.e. FFFFH)

STATE_TTIP_MES_LS_UP[i] = OFF

LV_V_OFS_MES_VLD_R_IT_LS_UP[i] = **!** LC_V_REF_R_IT_LS_UP

LV_V_OFS_ADJ_VLD_R_IT_LS_UP[i] = **!** LC_V_REF_R_IT_LS_UP

LV_FAC_GAIN_H_VLD_R_IT_LS_UP[i] = **!** LC_V_REF_R_IT_LS_UP

LV_FAC_GAIN_L_VLD_R_IT_LS_UP[i] = **!** LC_V_REF_R_IT_LS_UP

LV_TTIP_MES_VLD_LS_UP[i] = 0

LV_V_REF_VLD_R_IT_LS_UP[i] = ! LC_V_REF_R_IT_LS_UP

TCC_V_OFS_MES_R_IT_LS_UP[i] = 0

TCC_V_OFS_ADJ_R_IT_LS_UP[i] = 0

TCC_FAC_GAIN_H_R_IT_LS_UP[i] = 0

TCC_FAC_GAIN_L_R_IT_LS_UP[i] = 0

TCC_ERR_V_OFS_MES_R_IT_LS_UP[i] = 0

TCC_ERR_V_OFS_ADJ_R_IT_LS_UP[i] = 0

TCC_ERR_FAC_GAIN_H_R_LS_UP[i] = 0

TCC_ERR_FAC_GAIN_L_R_LS_UP[i] = 0

V_OFS_MES_R_IT_LS_UP[i] = C_V_OFS_REF_R_IT_LS_UP

V_OFS_ADJ_R_IT_LS_UP[i] = C_V_OFS_REF_R_IT_LS_UP

FAC_GAIN_H_R_IT_LS_UP[i] = C_FAC_GAIN_H_R_IT_LS_UP

FAC_GAIN_L_R_IT_LS_UP[i] = C_FAC_GAIN_L_R_IT_LS_UP

LV_R_IT_REQ_LS_UP[i] = 0

T_DLY_R_IT_IPLSL_CTL_ACT[i] = 0

Reset internal timers TIMER_1 & TIMER_2

Recurrence:

The function shall run at a recurrence of 10 ms.

The computation of a new internal resistance value shall be carried out once every 100 ms provided the applicable conditions have been met.

Activation:

The functions "*General validity of ceramic temperature measurement*", "*Reset statistical counters after diagnosis completion*" & "*State Machine*" shall be carried out according to the following conditions:

LV_INH_TTIP_LS_UP[i] == 0
and LV_IGK == 1
and STATE_LSL_IF_SPI_RD[i] (*bit* LSL_ENA) == 1

```
and     ( LV_LS_UP_READY_RAW[i] == 1
or      LV_LS_UP_READY_CDN[i] == 1
or      LV_LSH_UP_MAN_ACT[i] == 1 )
```

Deactivation:

```
        LV_INH_TTIP_LS_UP[i] ==1
or      LV_IGK == 0
or      STATE_LSL_IF_SPI_RD[i] ( bit LSL_ENA ) == 0
or      ( LV_LS_UP_READY_RAW[i] == 0
and     LV_LS_UP_READY_CDN[i] == 0
and     LV_LSH_UP_MAN_ACT[i] == 0 )
```

The following variable initialisation shall take place upon deactivation:

STATE_TTIP_MES_LS_UP[i] = OFF

LV_TTIP_MES_VLD_LS_UP[i] = 0

In addition upon deactivation, the following variables shall be conditionally initialised:

```
If      LV_LSL_IF_SPI_RST_END[i] == 1
then    LV_R_IT_CAL_ENA_LSL_IF[i] = 0
        LV_R_IT_OSC_ENA_LSL_IF[i] = 0
        If      LC_R_RNG_REQ_SWI_LS_UP_LSL_IF == 1
        then    LV_R_IT_SWI_RNG_LSL_IF[i] = LC_R_RNG_REQ_LS_UP_LSL_IF
        else    LV_R_IT_SWI_RNG_LSL_IF[i] = ! LV_LS_UP_READY[i]
        endif
endif
```

Formula section:

General validity of ceramic temperature measurement:

```
If      LV_R_IT_SWI_RNG_LSL_IF[i] == 0
then    LV_V_REF_VLD_R_IT_LS_UP[i]
        =       LV_V_OFS_MES_VLD_R_IT_LS_UP[i]
                &       LV_V_OFS_ADJ_VLD_R_IT_LS_UP[i]
                &       LV_FAC_GAIN_H_VLD_R_IT_LS_UP[i]
else    LV_V_REF_VLD_R_IT_LS_UP[i]
        =       LV_V_OFS_MES_VLD_R_IT_LS_UP[i]
                &       LV_V_OFS_ADJ_VLD_R_IT_LS_UP[i]
                &       LV_FAC_GAIN_L_VLD_R_IT_LS_UP[i]
```

endif

State Machine

The state machine shall remain in its current state and carry out the actions specified to occur within that state once per recurrence unless otherwise specified. The state machine shall only move to another state when one of the conditions has been determined to be met.

Notes:
1. The priorities of the conditions to change between states shall be defined by the order in which these conditions are listed within the appropriate state as described below.
2. TIMER_1 & TIMER_2 refer to SW internal variables. TIMER_1 & TIMER_2 are not visible output variables.
3. The assignment store CTR_ERR_LSL_IF_SPI_WR[i]$_N$ denotes that the current counter value shall be noted by the SW until the next assignment occurs.

STATE_TTIP_MES_LS_UP[i] = “OFF”

Actions:

LV_TTIP_MES_VLD_LS_UP[i] = 0

Note: The above listed actions must be carried out at least once when the state has been entered for the first time or from any other state!

Condition 1: “OFF” to “MES_OFS”

LV_LSL_IF_SPI_RST_END[i] = 1 **&**
LC_V_REF_R_IT_LS_UP = 1

Transition actions:

Reset *TIMER_1*
Start *TIMER_1*
Reset *TIMER_2*
Store CTR_ERR_LSL_IF_SPI_WR[i]$_N$
STATE_TTIP_MES_LS_UP[i] = MES_OFS
LV_R_IT_OSC_ENA_LSL_IF[i] = 0
LV_R_IT_CAL_ENA_LSL_IF[i] = 0

Condition 2: “OFF” to “MES”

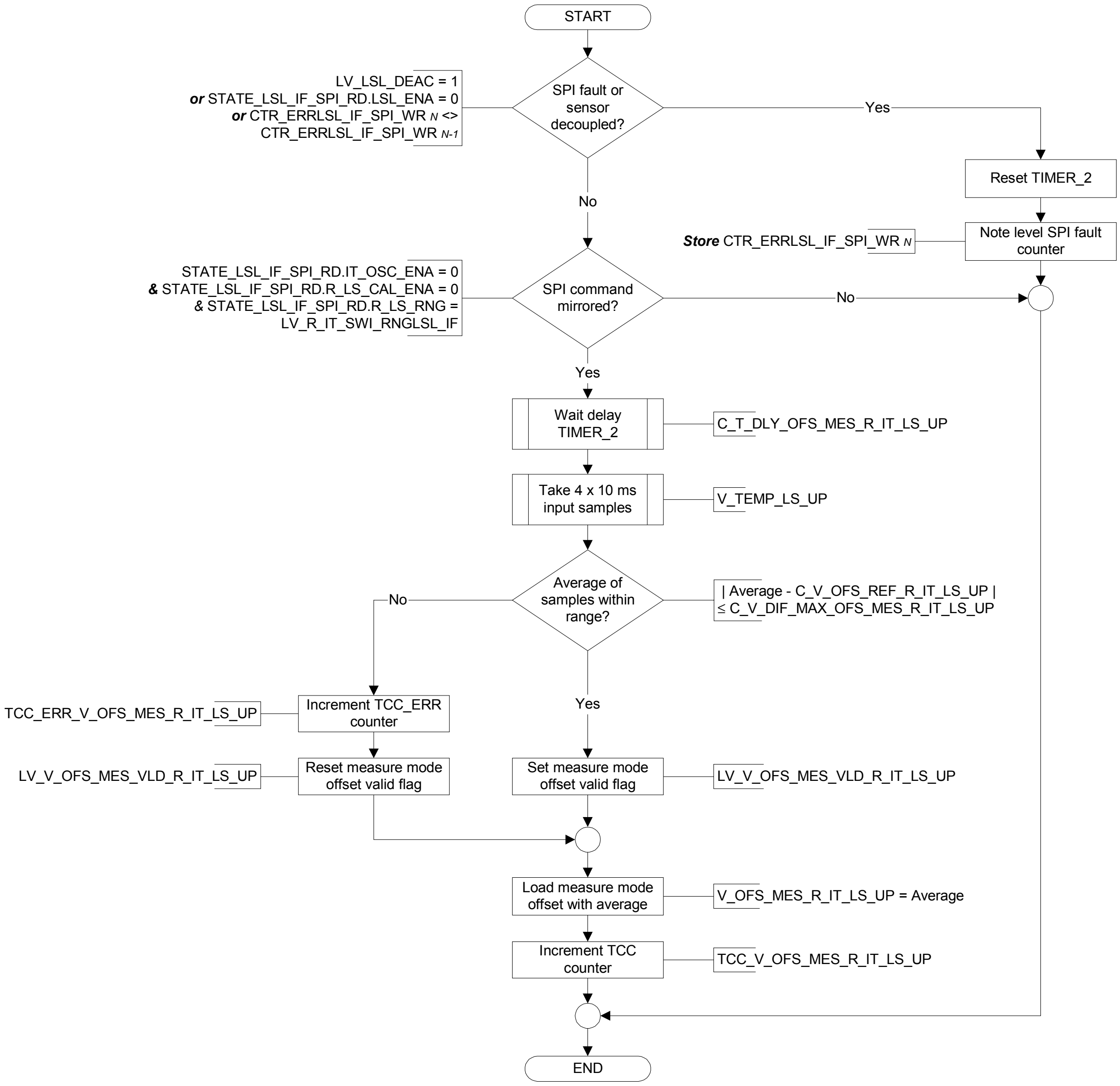
LV_LSL_IF_SPI_RST_END[i] = 1 **&**
LC_V_REF_R_IT_LS_UP = 0

Transition actions:

Reset *TIMER_2*
Store CTR_ERR_LSL_IF_SPI_WR[i]$_N$
STATE_TTIP_MES_LS_UP[i] = MES
LV_R_IT_OSC_ENA_LSL_IF[i] = 1
LV_R_IT_CAL_ENA_LSL_IF[i] = 0

STATE_TTIP_MES_LS_UP[i] = “MES_OFS”

Actions:



Note: TCC_V_OFS_MES_R_IT_LS_UP[i] & TCC_ERR_V_OFS_MES_R_IT_LS_UP[i] shall not be permitted to overflow.

Condition 4: *“MES_OFS” to “ADJ_OFS”*

MES_OFS procedure complete (valid or invalid)

Transition actions:

LV_R_IT_CAL_ENA_LSL_IF[i] = 1
Reset *TIMER_1*
Start *TIMER_1*
Reset *TIMER_2*
Store CTR_ERR_LSL_IF_SPI_WR[i]$_N$
STATE_TTIP_MES_LS_UP[i] = ADJ_OFS

Condition 5: *“MES_OFS” to “MES”*

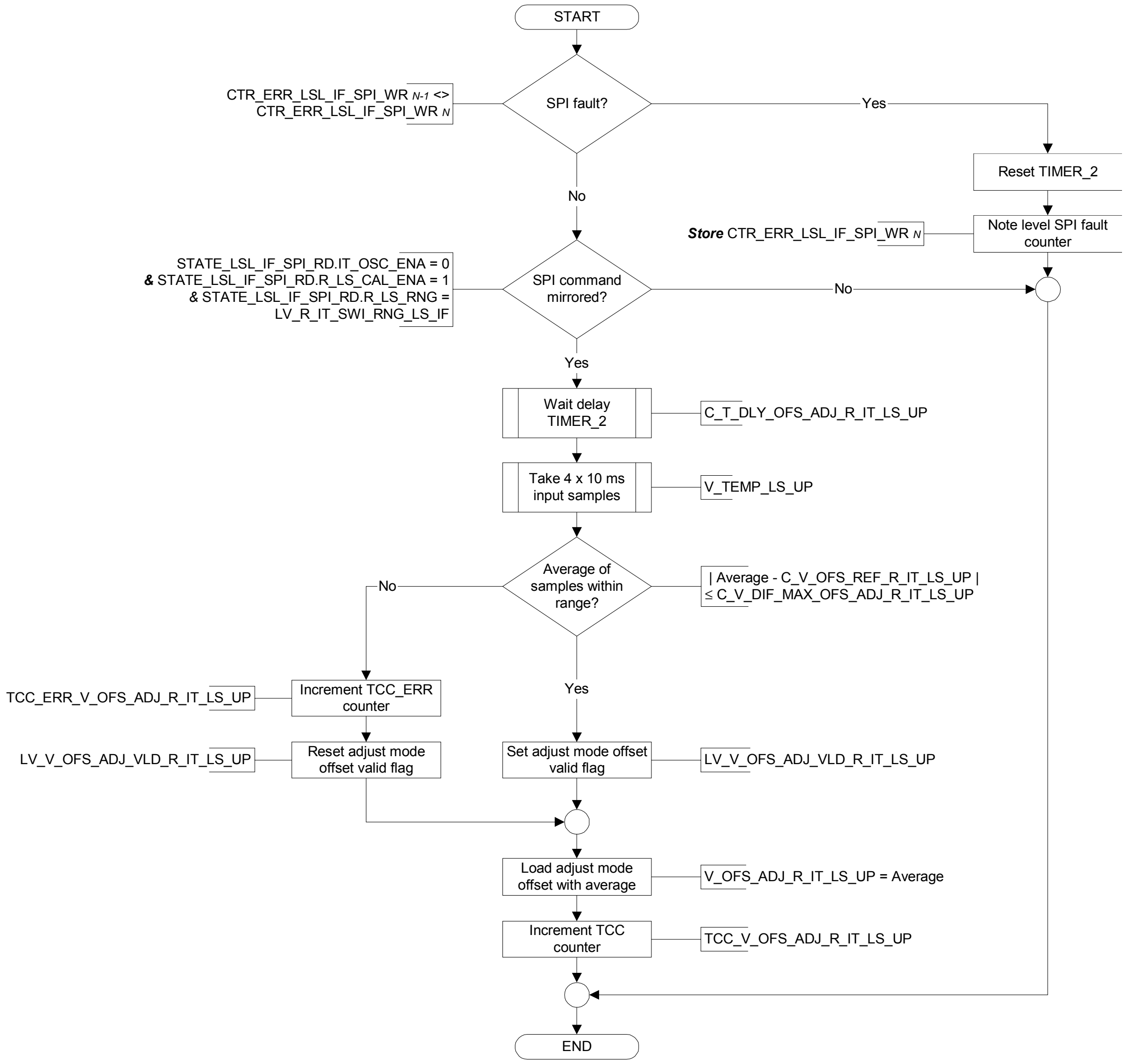
TIMER_1 ≥ C_T_MAX_V_OFS_MES_R_IT_LS_UP

Transition actions:

LV_R_IT_OSC_ENA_LSL_IF[i] = 1
Reset *TIMER_2*
Store CTR_ERR_LSL_IF_SPI_WR[i]$_N$
STATE_TTIP_MES_LS_UP[i] = MES

STATE_TTIP_MES_LS_UP[i] = “ADJ_OFS”

Actions:



Note: TCC_V_OFS_ADJ_R_IT_LS_UP[i] & TCC_ERR_V_OFS_ADJ_R_IT_LS_UP[i] shall not be permitted to overflow.

<u>Condition 6:</u> “ADJ_OFS” to “ADJ_GAIN”

ADJ_OFS procedure complete (valid or invalid) **&**
LV_LSL_OFS_ADJ_ACT[i] = 0

Transition actions:

LV_R_IT_OSC_ENA_LSL_IF[i] = 1
Reset *TIMER_1*
Start *TIMER_1*
Reset *TIMER_2*
Store CTR_ERR_LSL_IF_SPI_WR[i]$_N$
STATE_TTIP_MES_LS_UP[i] = ADJ_GAIN

<u>Condition 7:</u> “ADJ_OFS” to “MES”

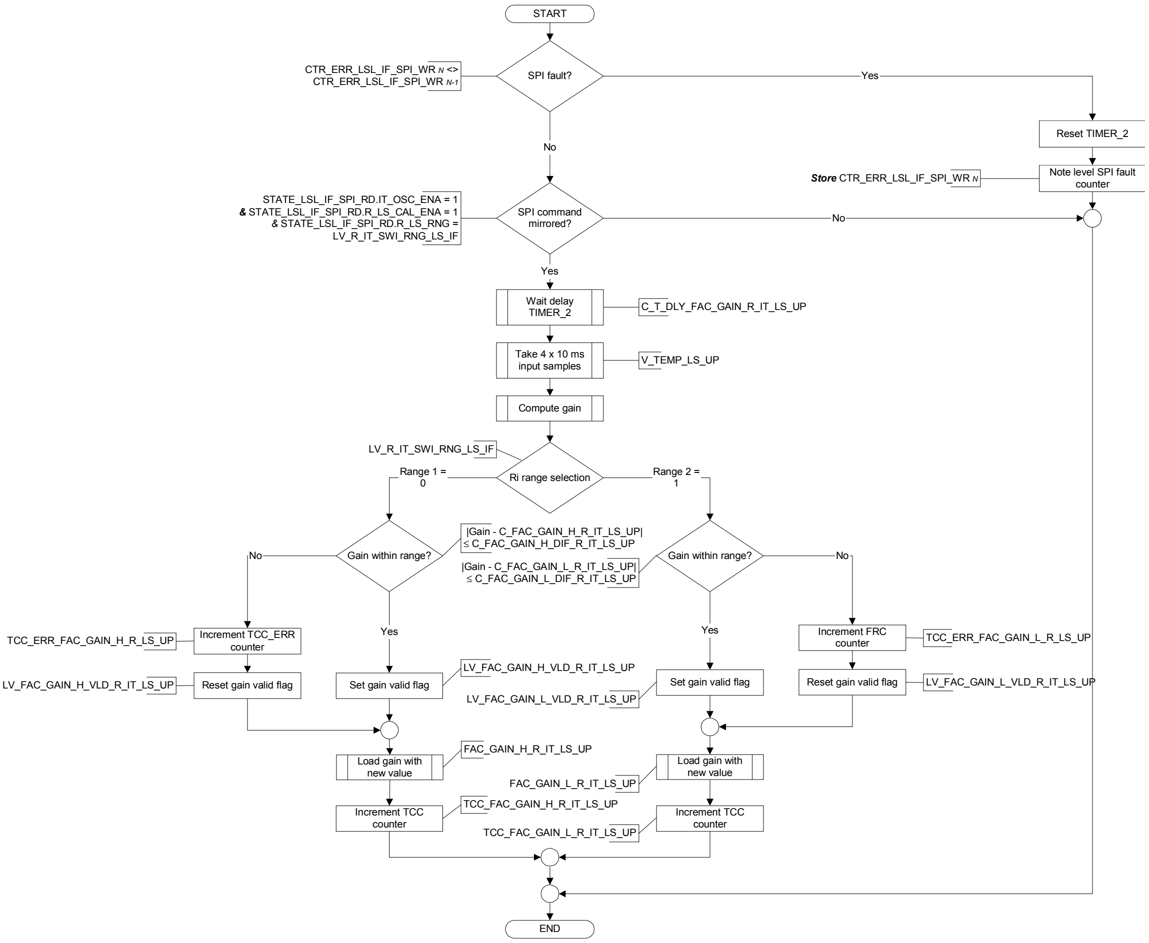
TIMER_1 ≥ C_T_MAX_V_OFS_ADJ_R_IT_LS_UP **&**
LV_LSL_OFS_ADJ_ACT[i] = 0

Transition actions:

```
LV_R_IT_OSC_ENA_LSL_IF[i] = 1
LV_R_IT_CAL_ENA_LSL_IF[i] = 0
Reset TIMER_2
Store CTR_ERR_LSL_IF_SPI_WR[i]_N
STATE_TTIP_MES_LS_UP[i] = MES
```

STATE_TTIP_MES_LS_UP[i] = “ADJ_GAIN”

Actions:



Note: TCC_FAC_GAIN_L_R_IT_LS_UP[i], TCC_ERR_FAC_GAIN_L_R_LS_UP[i], TCC_FAC_GAIN_H_R_IT_LS_UP[i] & TCC_ERR_FAC_GAIN_H_R_LS_UP[i] shall not be permitted to overflow.

Compute gain of internal resistance measurement circuit:

The gain shall be computed from the measurement circuit offset and the average of four (4) V_TEMP_LS_UP[i] samples in the adjustment mode using the following formula:

```
If      LV_R_IT_SWI_RNG_LSL_IF[i] == 0
then    FAC_GAIN_H_R_IT_LS_UP[i]
                = ( AV( V_TEMP_LS_UP[i] ) - V_OFS_ADJ_R_IT_LS_UP[i] )
                        * C_FAC_GAIN_R_IT_LS_UP
else    FAC_GAIN_L_R_IT_LS_UP[i]
                = ( AV( V_TEMP_LS_UP[i] ) - V_OFS_ADJ_R_IT_LS_UP[i] )
                        * C_FAC_GAIN_R_IT_LS_UP
endif
```

Where AV(V_TEMP_LS_UP[i]) is the average of the 4 previously samples taken at 10 ms intervals as shown in the appropriate flow chart.

Note: The difference in voltages (V_TEMP_LS_UP[i] – V_OFS_ADJ_R_IT_LS_UP[i]) shall be limited to 0 to prevent negative values.

Condition 9: "ADJ_GAIN" to "MES_OFS"

LV_LSL_OFS_ADJ_ACT[i] = 1 **or**
(ADJ_GAIN procedure complete **&**
LV_V_REF_VLD_R_IT_LS_UP[i] = 0)

Transition actions:

LV_R_IT_OSC_ENA_LSL_IF[i] = 0
LV_R_IT_CAL_ENA_LSL_IF[i] = 0
***Reset** TIMER_1*
***Start** TIMER_1*
***Reset** TIMER_2*
Store $\text{CTR\_ERR\_LSL\_IF\_SPI\_WR[i]}_N$
STATE_TTIP_MES_LS_UP[i] = MES_OFS

Condition 8: "ADJ_GAIN" to "MES"

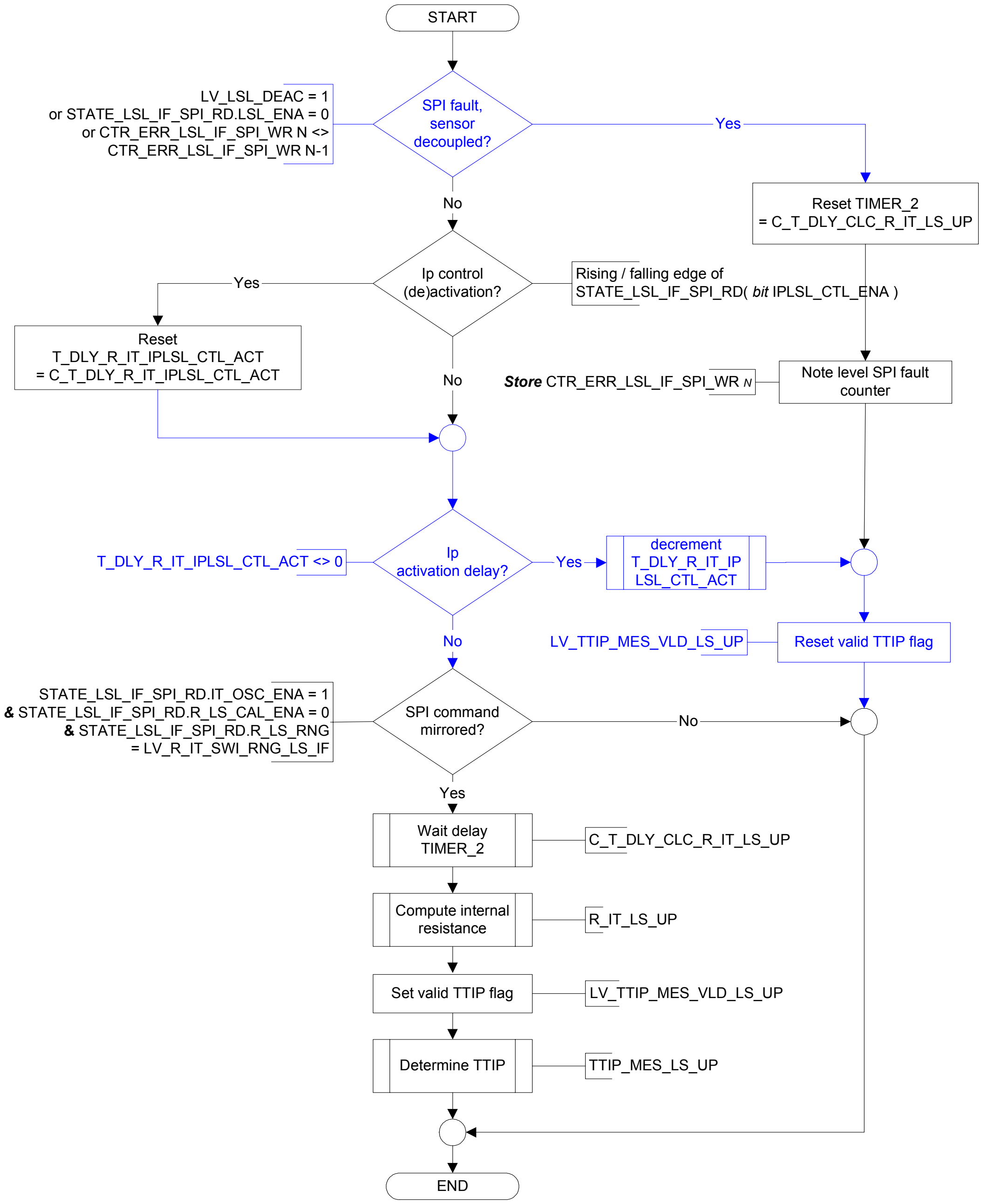
TIMER_1 ≥ C_T_MAX_FAC_GAIN_R_IT_LS_UP **or**
(ADJ_GAIN procedure complete **&**
LV_V_REF_VLD_R_IT_LS_UP[i] = 1)

Transition actions:

LV_R_IT_CAL_ENA_LSL_IF[i] = 0
***Reset** TIMER_2*
Store $\text{CTR\_ERR\_LSL\_IF\_SPI\_WR[i]}_N$
STATE_TTIP_MES_LS_UP[i] = MES

*STATE_TTIP_MES_LS_*UP[i] = *“MES”*

Actions:



Compute internal resistance of oxygen sensor element: (Deviating recurrence, see above)

The internal resistance shall be computed from the offset voltage determined in the measurement mode with the oscillator off, the measurement circuit gain determined in the adjust mode and AV(V_TEMP_LS_UP[i]). Once the active Ri determination conditions have been met as denoted above, the resistance shall be determined from the following formula.

```
If      LV_R_IT_SWI_RNG_LSL_IF[i] == 0
then    R_IT_LS_UP[i] = C_R_IT_REF_LS_UP
                * { AV( V_TEMP_LS_UP[i] ) - V_OFS_MES_R_IT_LS_UP[i] }
                / { 5V * FAC_GAIN_H_R_IT_LS_UP[i]
                        - ( AV( V_TEMP_LS_UP[i] ) - V_OFS_MES_R_IT_LS_UP[i] ) }
else    R_IT_LS_UP[i] = C_R_IT_REF_LS_UP
                * { AV( V_TEMP_LS_UP[i] ) - V_OFS_MES_R_IT_LS_UP[i] }
                / { 5V * FAC_GAIN_L_R_IT_LS_UP[i]
                        - ( AV( V_TEMP_LS_UP[i] ) - V_OFS_MES_R_IT_LS_UP[i] ) }
endif
```

Where AV(V_TEMP_LS_UP[i]) is either the average of the 8 previously samples taken at 10 ms intervals as shown in the appropriate flow chart or should the 100 ms task, in which the resistance is computed, be called before 8 samples could be taken then the latest V_TEMP_LS_UP[i] value shall be used. Should the timing be such that no current V_TEMP_LS_UP[i] are available then the resistance and hence ceramic temperature shall be calculated in the next 100 ms recurrence. Additionally, in this instance LV_TTIP_MES_VLD_LS_UP[i] would remain reset until the next recurrence.

Note: The difference in voltages (V_TEMP_LS_UP[i] – V_OFS_MES_R_IT_LS_UP[i]) shall be limited to 0 to prevent negative values.

Should internal resistance determination be suspended due to an adjustment measurement request, the value R_IT_LS_UP[i] shall be maintained at the value computed prior to suspension but the variable(s) used to compute the average shall be reinitialised.

Determine ceramic temperature from mapped characteristic oxygen sensor element temperature vs. internal resistance:

TTIP_MES_LS_UP[i] = IP_TTIP_MES_LS_UP (R_IT_LS_UP[i])

<u>*Condition 10:*</u> *“MES” to “MES_OFS”*

LC_V_REF_R_IT_LS_UP = 1 **&**
(LV_LSL_OFS_ADJ_ACT[i] = 1 **or**
LV_LS_UP_READY[i] *transition* 0 → 1 **or**
LC_R_RNG_REQ_SWI_LS_UP_LSL_IF *transition* 0 → 1)

Transition actions:

If (LC_R_RNG_REQ_SWI_LS_UP_LSL_IF = 1)

then

```
    LV_R_IT_SWI_RNG_LSL_IF[i] = LC_R_RNG_REQ_LS_UP_LSL_IF
else
    LV_R_IT_SWI_RNG_LSL_IF[i] = ! LV_LS_UP_READY[i]
endif.
LV_TTIP_MES_VLD_LS_UP[i] = 0
LV_R_IT_OSC_ENA_LSL_IF[i] = 0
Reset TIMER_1
Start TIMER_1
Reset TIMER_2
Store CTR_ERR_LSL_IF_SPI_WR[i]$_N$
STATE_TTIP_MES_LS_UP[i] = MES_OFS
```

Condition 11: *“MES” to “MES”*

LC_V_REF_R_IT_LS_UP = 0 **&**
(LV_LS_UP_READY[i] *transition* 0 → 1 **or**
LC_R_RNG_REQ_SWI_LS_UP_LSL_IF *transition* 0 → 1)

Transition actions:

```
If (LC_R_RNG_REQ_SWI_LS_UP_LSL_IF = 1)
then
    LV_R_IT_SWI_RNG_LSL_IF[i] = LC_R_RNG_REQ_LS_UP_LSL_IF
else
    LV_R_IT_SWI_RNG_LSL_IF[i] = ! LV_LS_UP_READY[i]
endif.
LV_TTIP_MES_VLD_LS_UP[i] = 0
Reset TIMER_2
Store CTR_ERR_LSL_IF_SPI_WR[i]$_N$
```

Additional variable / constant definitions

FRQ_R_IT_OSC_LSL_IF_SPI_WR[i] and C_FRQ_R_IT_OSC_LSL_IF shall be defined as follows:

Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
X	D6	D5	D4	D3	D2	D1	D0

Where:

X	denotes don’t care
D0 through D5	denotes the 6 bit coded oscillator frequency information. A value of 00H represents 2.0 kHz and each step 125 Hz. Hence 3F represents 9.875 kHz.
D6	denotes the bit used to selectively disable pins inside of the protection time under fault conditions. 0 pins will be decoupled after the protection time has passed. 1 the pin at which the fault is detected will immediately be decoupled, i.e. inside the protection time before the remaining pins are decoupled after the passing of the protection time.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TTIP_MES_LS_UP	8	0...7FFFH	0...2047.9375	0.0625	[°C]
LDP_R_IT_LS_UP_IP_TTIP_LS_UP	8	0...FFFFH	0...16383.75	0.25	[ohm]
Oxygen sensor temperature vs. internal resistance characteristic					
C_T_DLY_OFS_MES_R_IT_LS_UP	1	0...FFH	0...2.55	0.01	[s]
Delay required from switching oscillator off or internal resistance measurement to measurement offset mode					
C_T_DLY_OFS_ADJ_R_IT_LS_UP	1	0...FFH	0...2.55	0.01	[s]
Delay required from switching from internal resistance measurement offset to adjust offset mode					
C_T_DLY_FAC_GAIN_R_IT_LS_UP	1	0...FFH	0...2.55	0.01	[s]
Delay required from switching from internal resistance adjust offset to adjust gain mode					
C_T_DLY_CLC_R_IT_LS_UP	1	0...FFH	0...2.55	0.01	[s]
Delay required from switching from internal resistance adjust gain to measurement mode					
C_T_MAX_V_OFS_MES_R_IT_LS_UP	1	0...FFFFH	0...655.35	0.01	[s]
Maximum permitted time for offset measurement in measurement mode					
C_T_MAX_V_OFS_ADJ_R_IT_LS_UP	1	0...FFFFH	0...655.35	0.01	[s]
Maximum permitted time for offset measurement in adjust mode					
C_T_MAX_FAC_GAIN_R_IT_LS_UP	1	0...FFFFH	0...655.35	0.01	[s]
Maximum permitted time for gain adjust measurement					
C_V_DIF_MAX_OFS_MES_R_IT_LS_UP	1	0...FFCH	0...4.99511	1.22E-03	[V]
Maximum permitted deviation of V_OFS_MES_R_IT_LS_UP [NC_CBK_EX_NR] to nominal reference value					
C_V_DIF_MAX_OFS_ADJ_R_IT_LS_UP	1	0...FFCH	0...4.99511	1.22E-03	[V]
Maximum permitted deviation of V_OFS_ADJ_R_IT_LS_UP [NC_CBK_EX_NR] to nominal reference value					
C_FAC_GAIN_H_DIF_R_IT_LS_UP	1	0...FFFFH	0...23.9996	3.66E-04	[-]
Maximum permitted deviation of measured FAC_GAIN_H_R_IT_LS_UP [NC_CBK_EX_NR] from nominal reference value					
C_FAC_GAIN_L_DIF_R_IT_LS_UP	1	0...FFFFH	0...23.9996	3.66E-04	[-]
Maximum permitted deviation of measured FAC_GAIN_L_R_IT_LS_UP [NC_CBK_EX_NR] from nominal reference value					
LC_R_RNG_REQ_SWI_LS_UP_LSL_IF	1	0...1H	0...1	1	[-]
Boolean flag enabling manual switching of internal resistance determination circuit gain					
LC_R_RNG_REQ_LS_UP_LSL_IF	1	0...1H	0...1	1	[-]
Boolean flag selecting internal resistance determination circuit gain when manual switching enabled					
LC_V_REF_R_IT_LS_UP	1	0...1H	0...1	1	[-]
Boolean flag enabling internal resistance determination circuit reference measurement					
C_T_DLY_R_IT_IPLSL_CTL_ACT	1	0...FFH	0...2.55	0.01	[s]
Delay time for freeze of Ri signal upon activation of Ip loop					
C_FAC_GAIN_R_IT_LS_UP	1	0...FFFFH	0...76.79882	1.17E-03	[1/V]
Factor used when determining high gain = (C_R_IT_REF_LS_UP+R5+R6)/(5V*(R5+R6)					
C_R_IT_REF_LS_UP	1	0...FFFFH	0...65535	1	[ohm]
Reference current pulse resistor for internal resistance determination circuit = (R2*R1)/(R2+R1)					
C_FAC_GAIN_H_R_IT_LS_UP	1	0...FFFFH	0...23.9996	3.66E-04	[-]
Reference high gain for internal resistance determination circuit					
C_FAC_GAIN_L_R_IT_LS_UP	1	0...FFFFH	0...23.9996	3.66E-04	[-]
Reference low gain for internal resistance determination circuit					
C_V_OFS_REF_R_IT_LS_UP	1	0...FFCH	0...4.99511	1.22E-03	[V]
Reference offset voltage for internal resistance determination circuit					
C_FRQ_R_IT_OSC_LSL_IF	1	0...7FH	0...127	1	[-]
Coded information for Ri determination oscillator frequency					

16.10.1 Application specific conditions for oxygen sensor ceramic temperature

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_TTIP_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that upstream oxygen sensor ceramic temperature determination shall be inhibited					

Input data:

LV_DIAG_OC_IF_OSC_DI_LSL_UP[NC_CBK_EX_NR]	LV_ST_END	NC_CBK_EX_NR	LV_ES
STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]			

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

The Boolean flag LV_INH_TTIP_LS_UP[i] shall indicate whether the upstream oxygen sensor ceramic temperature determination shall be permitted to take place. When set, no temperature determination shall take place.

Description:

Application conditions:

Initialisation:

The following initialisation shall be carried out after RESET and upon leaving the engine state Engine Stop (LV_ES).

LV_INH_TTIP_LS_UP[i] = 0

Recurrence: 1s

Activation:

LV_ST_END == 1

Deactivation:

LV_ST_END == 0

Formula section:

```
If      LV_DIAG_OC_IF_OSC_DI_LSL_UP[i] == 1
or      STATE_LSL_IF_SPI_RD[i] bit LSL_ENA == 0
then    LV_INH_TTIP_LS_UP[i] = 1
else    LV_INH_TTIP_LS_UP[i] = 0
endif
```

16.11 Oxygen Sensor Signal Processing   Upstream

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_UP[NC_CBK_EX_NR]	O/V	0...3FFH	0...4.99511719	0.00488281	V
Upstream oxygen sensor voltage - Bank 1					
LV_VLS_UP_VLD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Boolean flag indicating whether clipping of the WRAF sensor signal is taking place (1 - No clipping, 0 - clipping)					
VLS_UP_10_RAW[NC_CBK_EX_NR]	O/V	0...3FFH	0...4.99511719	0.00488281	V
Raw value of WRAF sensor voltage signal - Recurrence 10.0 msec					
LV_VLS_UP_INIT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Boolean flag indicating the buffer initialisation after switching between Gain 16 & Gain 8 values of WRAF Controller or after Offset adjustment of WRAF sensor					
VLS_UP_TOT_SUM_MV[NC_CBK_EX_NR]	V	0...FFFFFFFFH	0...2.09715E+7	0.00488281	V
Total sum of raw signal voltages taken for mean value calculation					
NR_SAMPLE_TOT_MV[NC_CBK_EX_NR]	V	0...FFH	0...255	1	-
Total number of samples taken for mean value calculation					
T_SEG_LEN	V	0...FFH	0...0.255	0.001	s
Actual segment time measured in msec					
NR_VALUE_BUF_SUM	V	0...FFH	0...255	1	-
Number of values to be taken from VLS_BUF_SUM for mean value calculation					
NR_SAMPLE_1_SUM	V	0...FFH	0...255	1	-
Number of samples taken for Intermediate Sum 1					
NR_VLS_UP_RAW_BUF_IDX	V	0...FFH	0...255	1	-
Index for the ring buffer VLS_UP_RAW_BUF as well as the Counter to trigger the mean value calculation of WRAF sensor signal					
NR_VLS_UP_RAW_SAMPLE_MV	V	0...FFH	0...255	1	-
Total number of samples required for mean value calculation					
STATE_LSL_IF_SPI_RD_OLD[NC_CBK_EX_NR]	V	0...FFH	0...255	1	-
past initialization status register information for WRAF controller					
NR_VALUE_BUF_SAMPLE	V	0...FFH	0...255	1	-
Number of values to be taken from NR_BUF_SAMPLE for mean value calculation					
VLS_UP_MV[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511719	0.00488281	V
Mean value of Upstream Oxygen sensor voltage					
NR_SAMPLE_2_SUM	V	0...FFH	0...255	1	-
Number of samples taken for Intermediate Sum 2					
NR_VLS_UP_BUF_IDX[NC_CBK_EX_NR]	V	0...FFH	0...255	1	-
Index for the ring buffer NR_BUF_SAMPLE					
VLS_UP_INTER_1_SUM[NC_CBK_EX_NR]	V	0...FFFFH	0...319.995117	0.00488281	V
Intermediate sum 1					
VLS_UP_INTER_2_SUM[NC_CBK_EX_NR]	V	0...FFFFH	0...319.995117	0.00488281	V
Intermediate sum 2					
T_DLY_VLS_UP_INIT[NC_CBK_EX_NR]	V	0...FFH	0...2.55	0.01	s
reset timer for LV_VLS_UP_INIT					
STATE_LSL_IF_SPI_WR_OLD[NC_CBK_EX_NR]	V	0...FFH	0...255	1	-
past initialization write status register information for WRAF controller					
NR_BUF_SAMPLE[NC_NR_BUF_SAMPLE_LEN][NC_CBK_EX_NR]	-	0...FFH	0...255	1	-
Ring buffer for saving the number of samples taken for intermediate sum.					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_BUF_SUM[NC_NR_BUF_SUM_LEN][NC_CBK_EX_NR]	-	0...FFFFH	0...319.995117	0.00488281	V
Ring buffer for saving the intermediate sums of raw sensor signal					
VLS_UP_RAW_BUF[NC_NR_RAW_BUF_LEN][NC_CBK_EX_NR]	-	0...3FFH	0...4.99511719	0.00488281	V
Ring Buffer for storing the raw signal of WRAF sensor every 1 msec					

Input data:

LV_IPLSL_VLD[NC_CBK_EX_NR]	VLS_UP_RAW[NC_CBK_EX_NR]	NC_CYL_NR	N
NC_CBK_EX_NR	STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]	STATE_LSL_IF_SPI_WR[NC_CBK_EX_NR]	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_VLS_UP_MV	1	0...1FE0H	0...8.16E+3	1	rpm
Maximum engine speed threshold for calculating mean value (background : save CPU load)					
C_T_DLY_VLS_UP_INIT	1	1...FFH	0.01...2.55	0.01	s
reset timer for LV_VLS_UP_INIT, Initial value = 1					
C_T_DLY_VLS_UP_INIT_MAX	1	1...FFH	0.01...2.55	0.01	s
initial value for delay timer in case of gain-switching or offset adjustment					
C_VLS_UP_THD_BOL	1	0...3FFH	0...4.995117	0.00488281	V
Lower voltage threshold for WRAF sensor signal					
C_VLS_UP_THD_TOL	1	0...3FFH	0...4.995117	0.00488281	V
Upper voltage threshold for WRAF sensor signal					
LC_VLS_UP_MV_ON	1	0...1H	0...1	1	-
Calibration flag to switch between raw value and filtered value (1- filtered, 0 - raw)					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_FAC_10_SAMPLE	1	0...FFH	0...255	1	-
Factor for multiplication, if segment time is less than NC_T_SEG_LEN_MIN					
NC_FAC_5_SAMPLE	1	0...FFH	0...255	1	-
Factor for multiplication, if segment time is less than 5 msec					
NC_NR_BUF_SAMPLE_LEN	1	0...FFH	0...255	1	-
Buffer size for storing the number of samples taken for intermediate sum 1 (for minimum speed of 500 RPM)					
NC_NR_BUF_SUM_LEN	1	0...FFH	0...255	1	-
Buffer size for storing the intermediate sums of sensor signal every 10 ms (for minimum speed of 500 RPM)					
NC_NR_RAW_BUF_LEN	1	0...FFH	0...255	1	-
Ring buffer size for storing the raw sensor signal every 1 msec					
NC_N_MIN_VLS_UP_MV	1	0...1FE0H	0...8.16E+3	1	rpm
Minimum engine speed threshold for calculating the mean value					
NC_T_SEG_LEN_MIN	1	0...FFH	0...0.255	0.001	s
Minimum segment time required for mean value calculation					

16.11.1 EGCP_SIGCVLSL3

The WRAF sensor’s signal is influenced by the pressure fluctuations in the exhaust manifold. The dynamic pressure peaks in exhaust gas results in momentary increase of oxygen concentration inside the reference cell of WRAF sensor & the same is refelected at the sensor output signal. Pressure fluctuation comes mainly from segment timing, quick open & close of exhaust valve, misfiring and may be some other parameter. Out of all these most dominant influence comes from segment timing. To remove segment timing influence from from the signal a mean value is evaluated.

The calculation shall be done for all exhaust cylinder banks.

NC_CBK_EX_NR defines the number of exhaust banks. For vector elements, the variable extension „_i“ is used in the model instead of „[i]“ as found in the textual description.

To evaluate segment synchronous mean value, the raw voltage signal is captured every 1 ms (from BSW) and stored in a ring buffer. The size of this ring buffer is defined by NC_NR_RAW_BUF_LEN and it is **fixed** at 10. All other calculations in this function is triggered soon after the 10th value is written in to the buffer. The segment time in m-sec is calculated every 10 ms.

In order to save the memory space, 2 intermediate sums is built out of the earlier 10 values and stored in a second ring buffer. At the same time, the number of samples taken for first intermediate sum calculation is stored in the third ring buffer. The number of samples required for mean value as well as intermediate sums is calculated dynamically based on the segment time.

The final mean value is calculated by reading the buffers filled with intermediate sums and corresponding number of samples. The values are filled in all the three ring buffers from bottom to top.

It is possible to switch off this function using the switch LC_VLS_UP_MV_ON, in which case only the raw sensor signal is delivered at 10 ms recurrence.

The ring buffer size for storing the intermediate sums and number of samples is calculated for a minimum engine speed of 500 rpm (Engine having 4 cyl and 2 exhaust bank – worst case). Hence the default values are

NC_N_MIN_VLS_UP_MV = 500 rpm

NC_NR_BUF_SAMPLE_LEN = 12

NC_NR_BUF_SUM_LEN = 2 * NC_NR_BUF_SAMPLE_LEN = 24

Application Condition

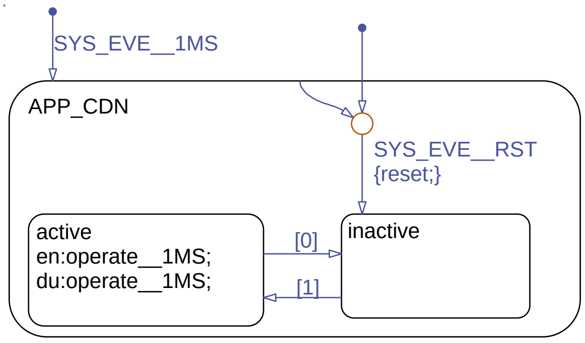


Figure 3 EGCP_SIGCVLSL3/ APP_CDN/ Chart

Function Description

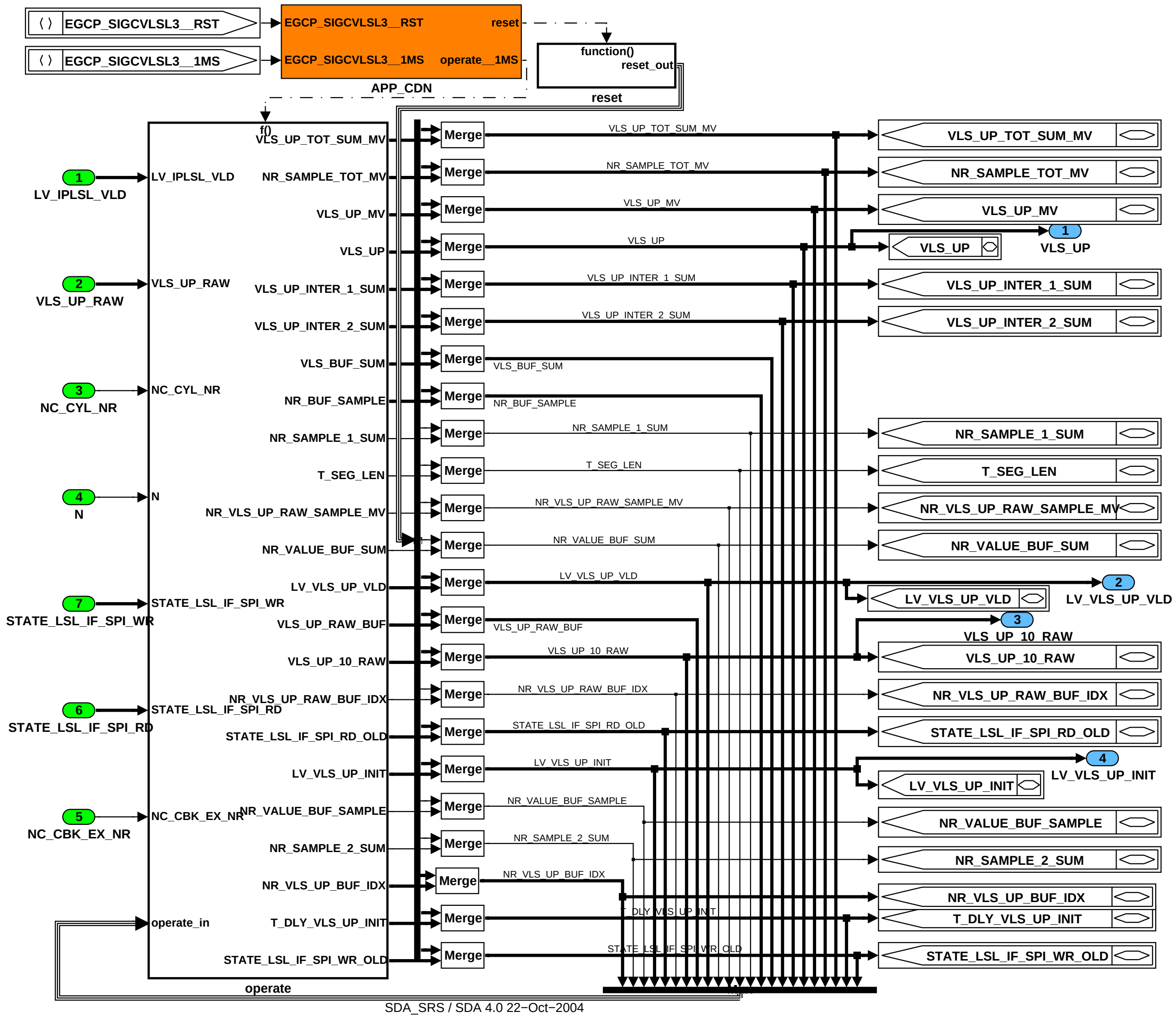


Figure 4 EGCP_SIGCVLSL3

16.11.1.1 SUBFUNCTION: operate

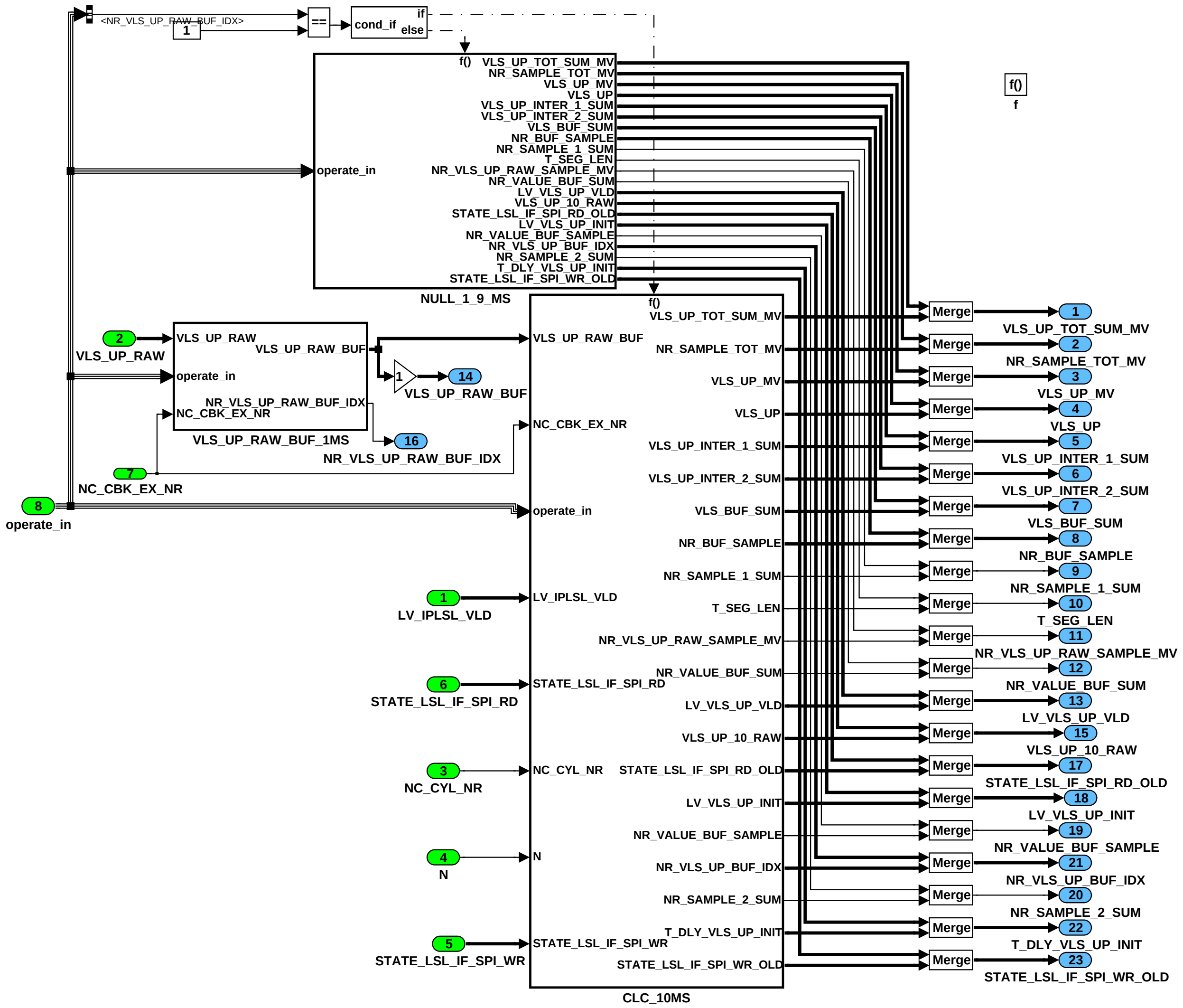


Figure 5 EGCP_SIGCVLSL3/ operate

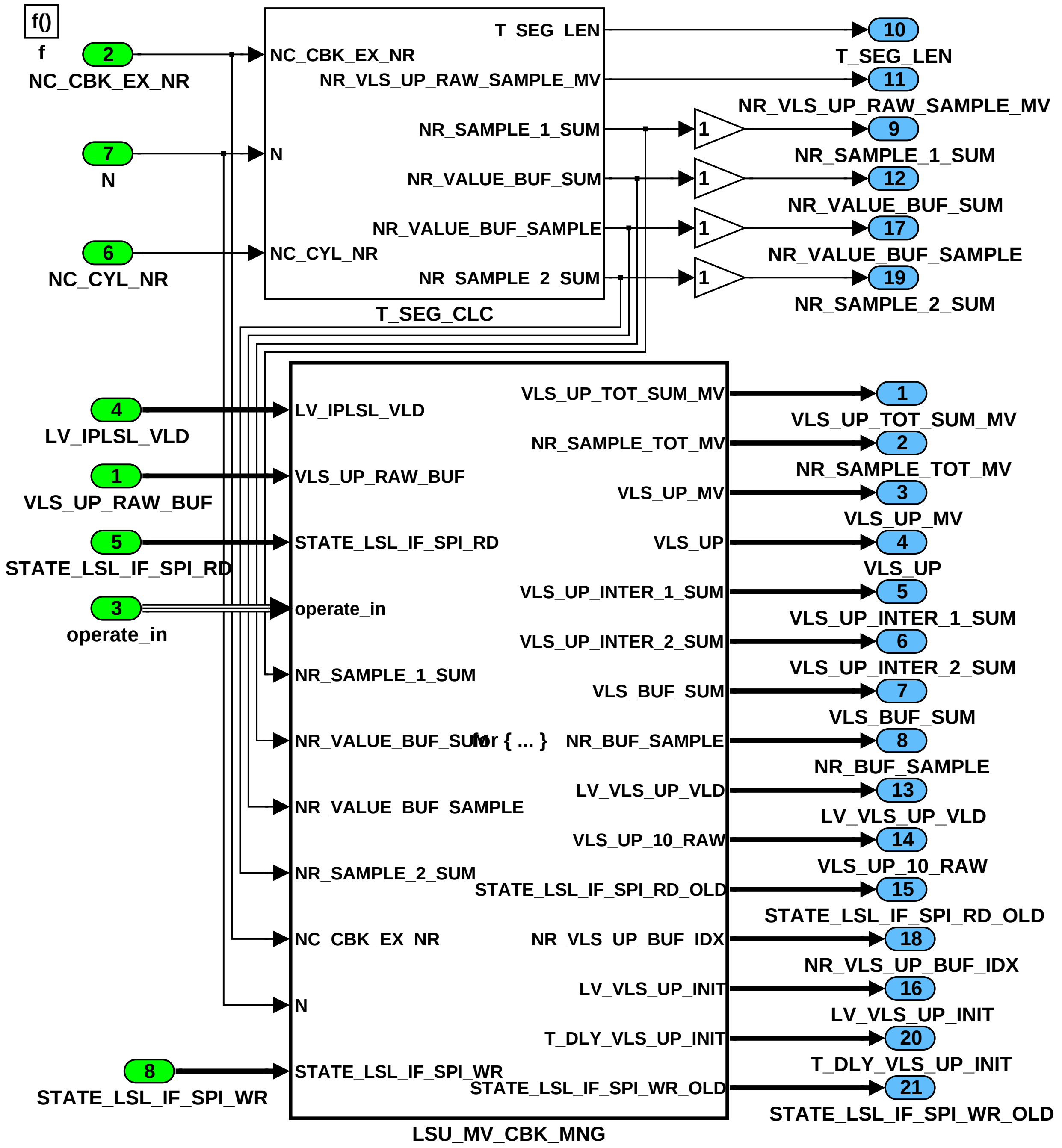


Figure 6 EGCP_SIGCVLSL3/ operate/ CLC_10MS

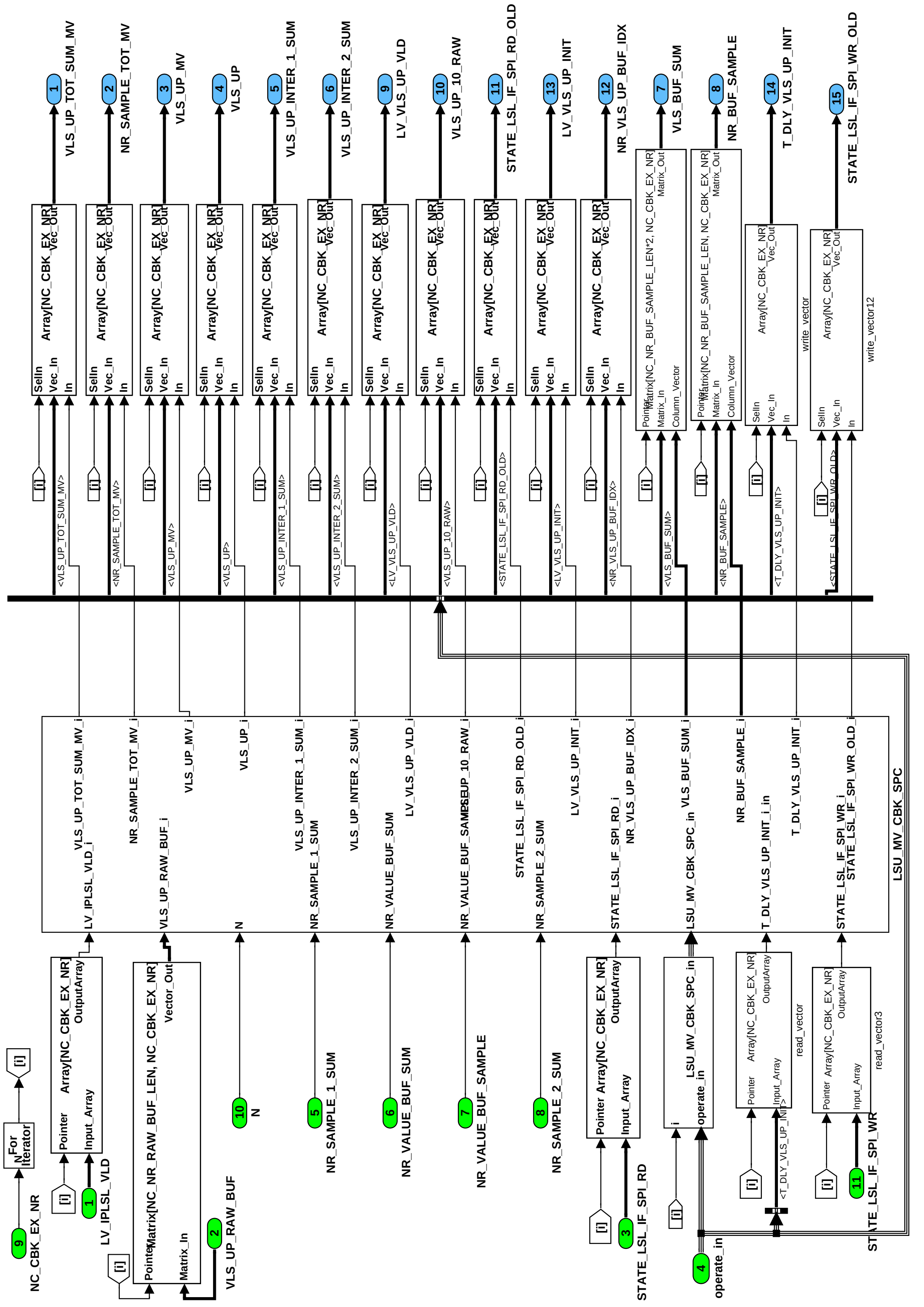


Figure 7 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG

Re-initialisation of buffers

The raw signal voltage is always compared with limitting values and clipped if any. Accordingly the flag LV_VLS_UP_VLD is set.

The bit LV_VLS_UP_INIT is set to 1 whenever any one of the 2 LSBs of STATE_LSL_IF_SPI_WR toggles. T_DLY_VLS_UP_INIT is initialized to calibration value whenever any one of the 2 LSBs of STATE_LSL_IF_SPI_WR toggles. The bit LV_VLS_UP_INIT reset to 0 after expiry of timer T_DLY_VLS_UP_INIT.

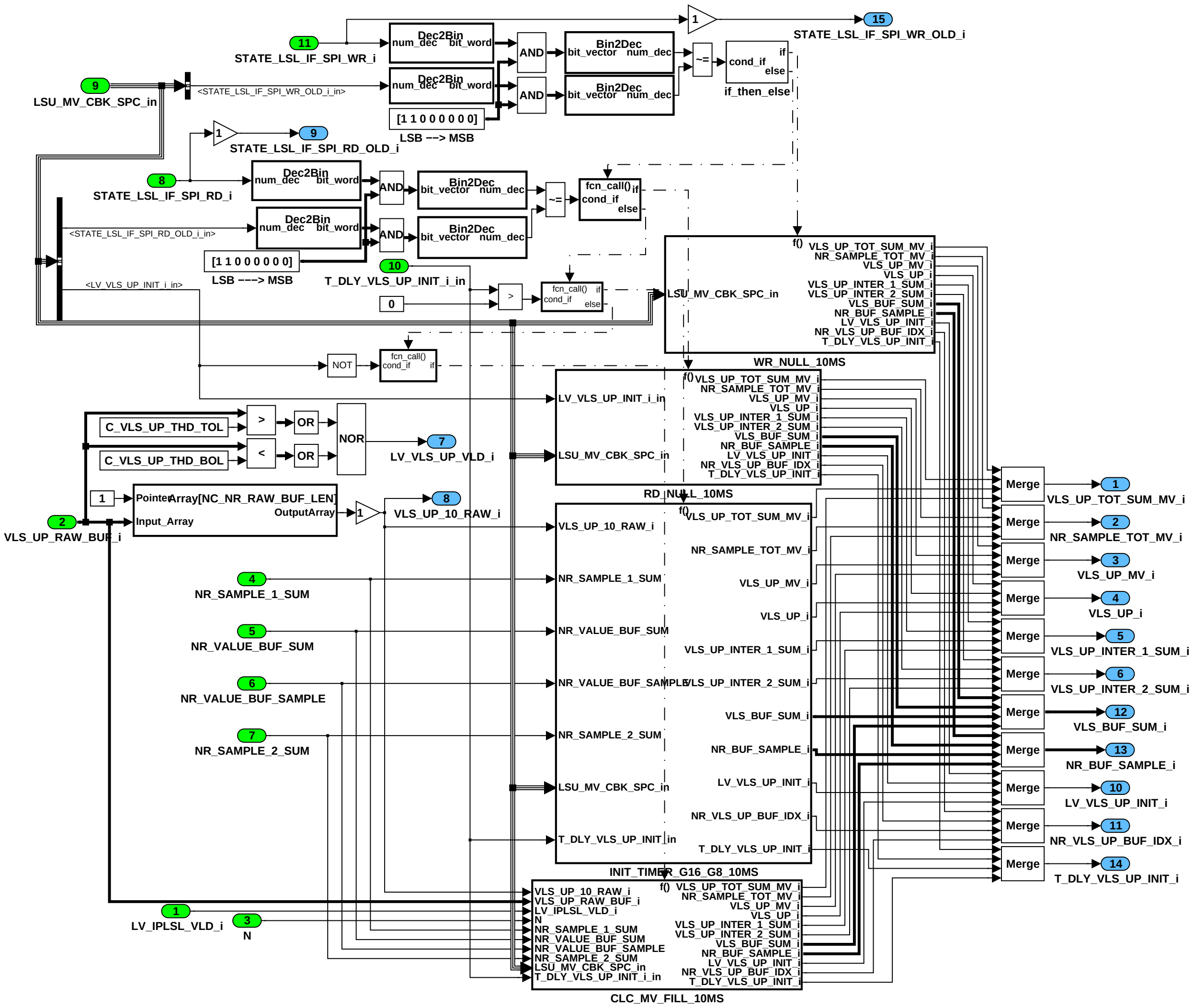


Figure 8 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC

Filling ring buffers with current values

During engine starting, the buffers are filled with current sensor values till the conditions for mean value calculation are fulfilled. In this case, VLS_UP would be same as VLS_UP_10_RAW (every 10th raw signal value stored in VLS_UP_RAW_BUF).

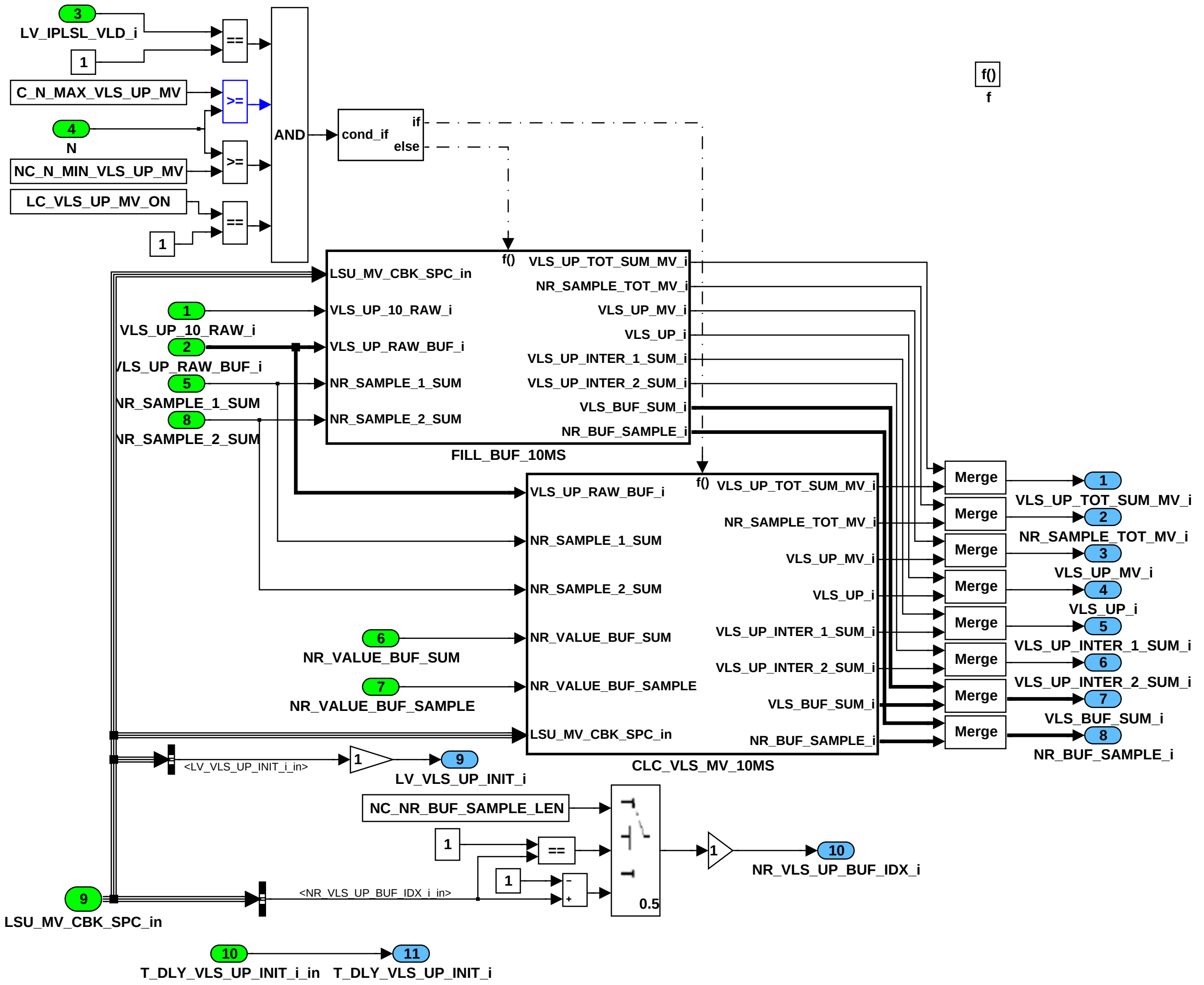


Figure 9 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS

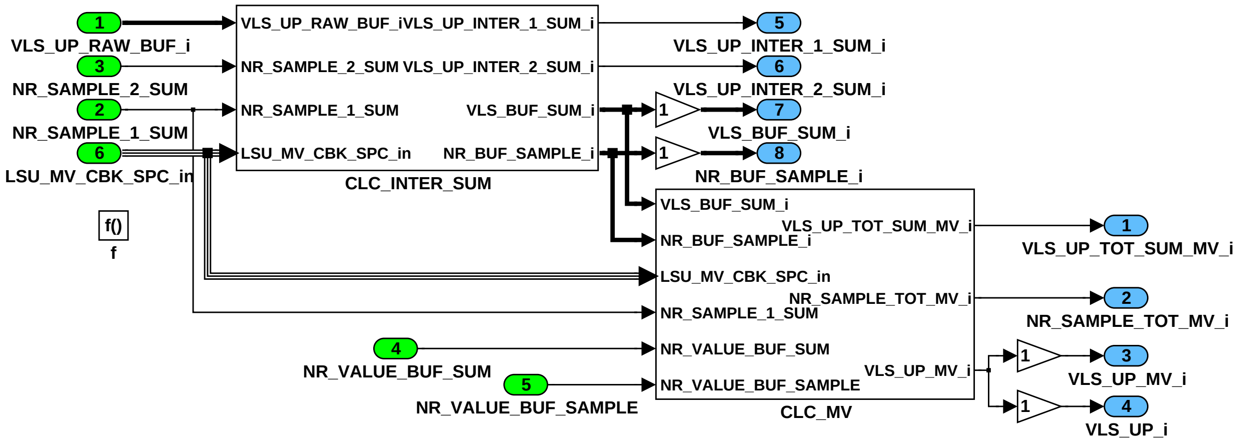


Figure 10 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ CLC_VLS_MV_10MS

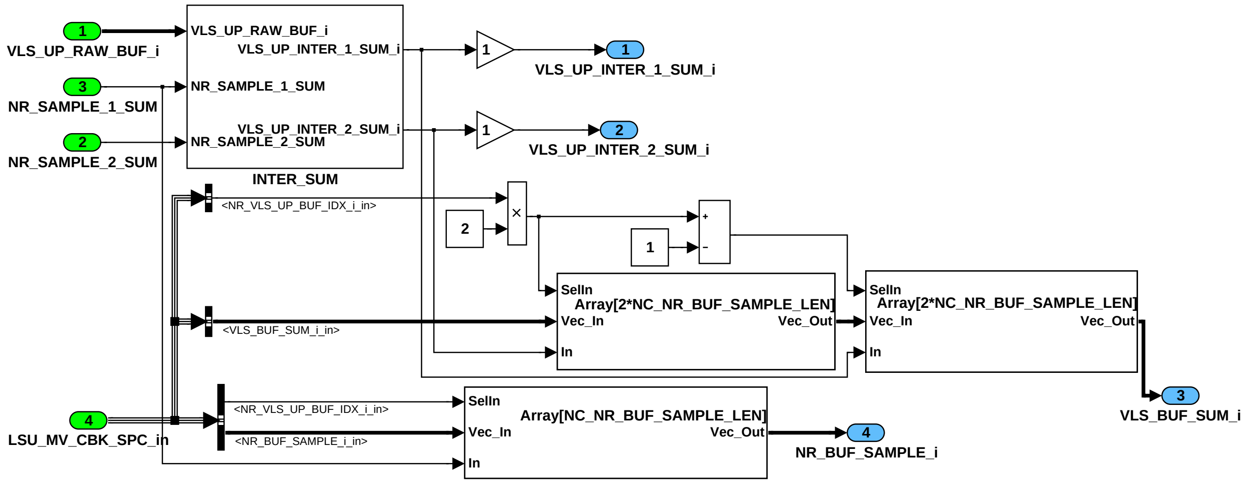


Figure 11 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ CLC_VLS_MV_10MS/ CLC_INTER_SUM

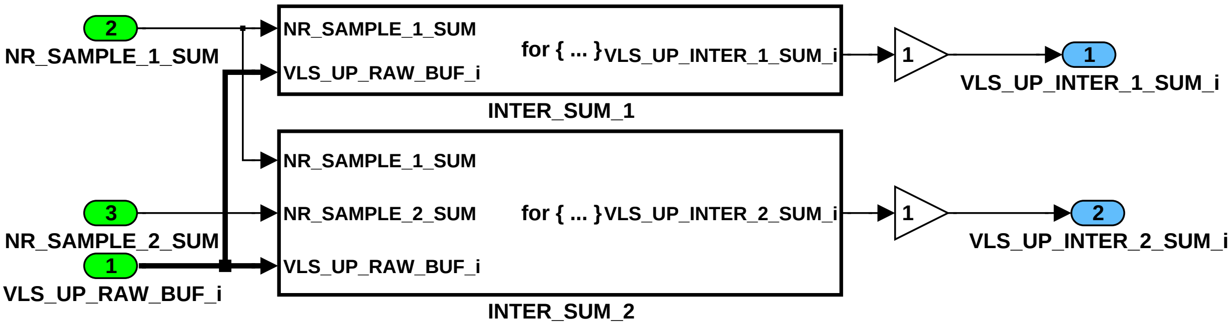


Figure 12 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ CLC_VLS_MV_10MS/ CLC_INTER_SUM/ INTER_SUM

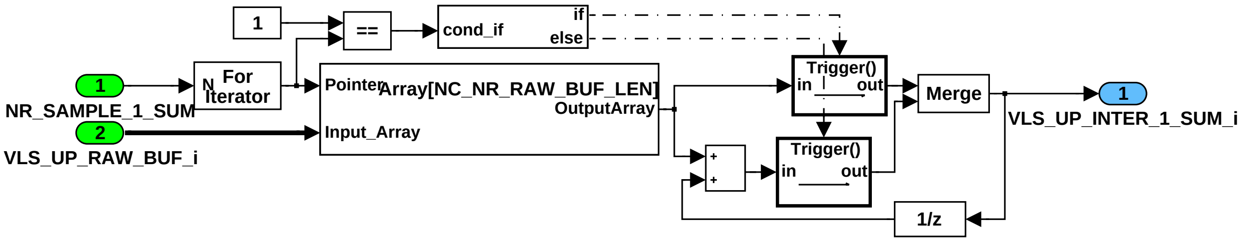


Figure 13 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ CLC_VLS_MV_10MS/ CLC_INTER_SUM/ INTER_SUM/ INTER_SUM_1

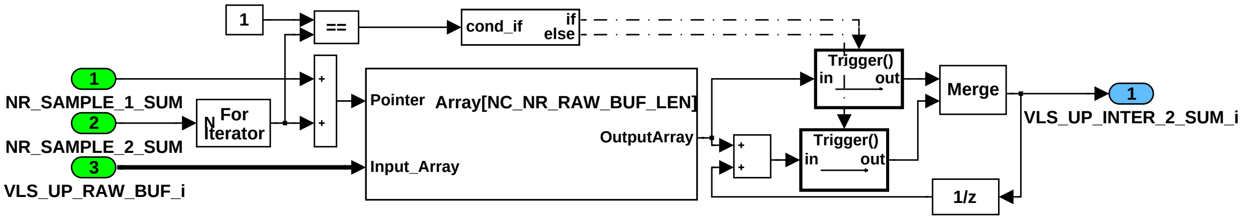


Figure 14 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ CLC_VLS_MV_10MS/ CLC_INTER_SUM/ INTER_SUM/ INTER_SUM_2

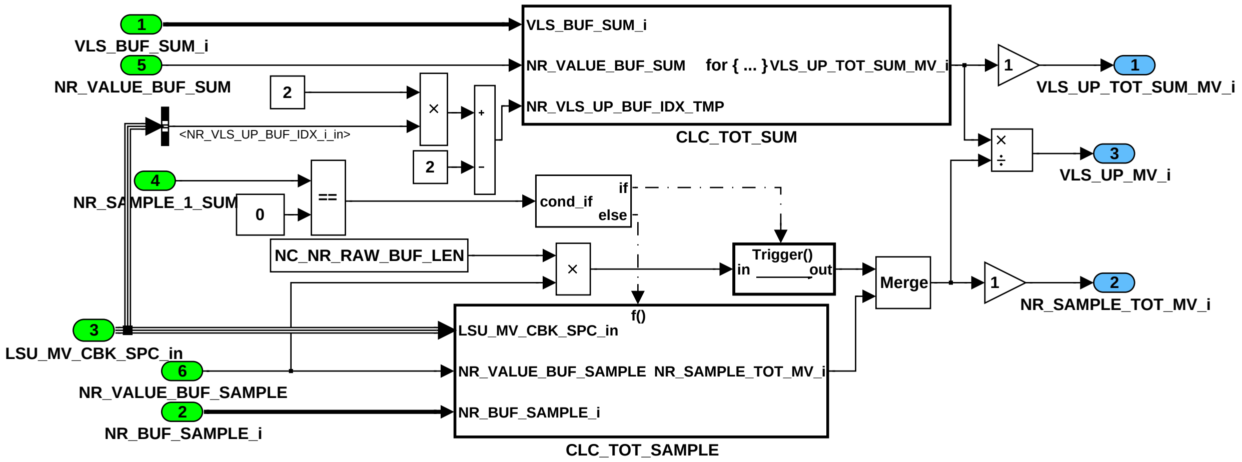


Figure 15 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ CLC_VLS_MV_10MS/ CLC_MV

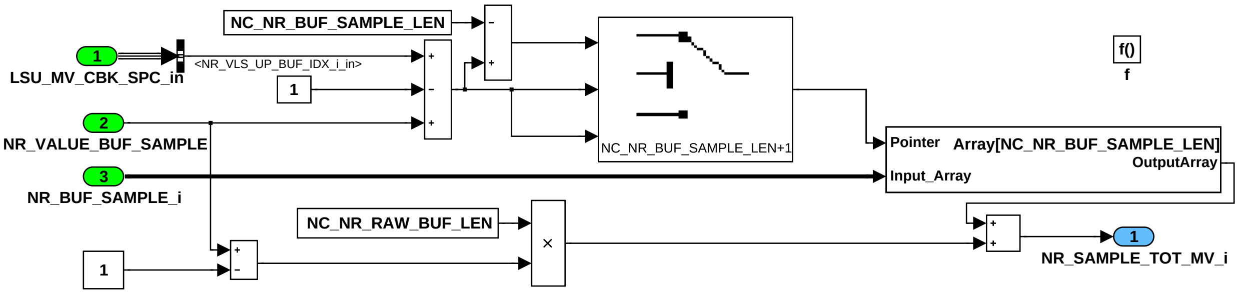


Figure 16 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ CLC_VLS_MV_10MS/ CLC_MV/ CLC_TOT_SAMPLE

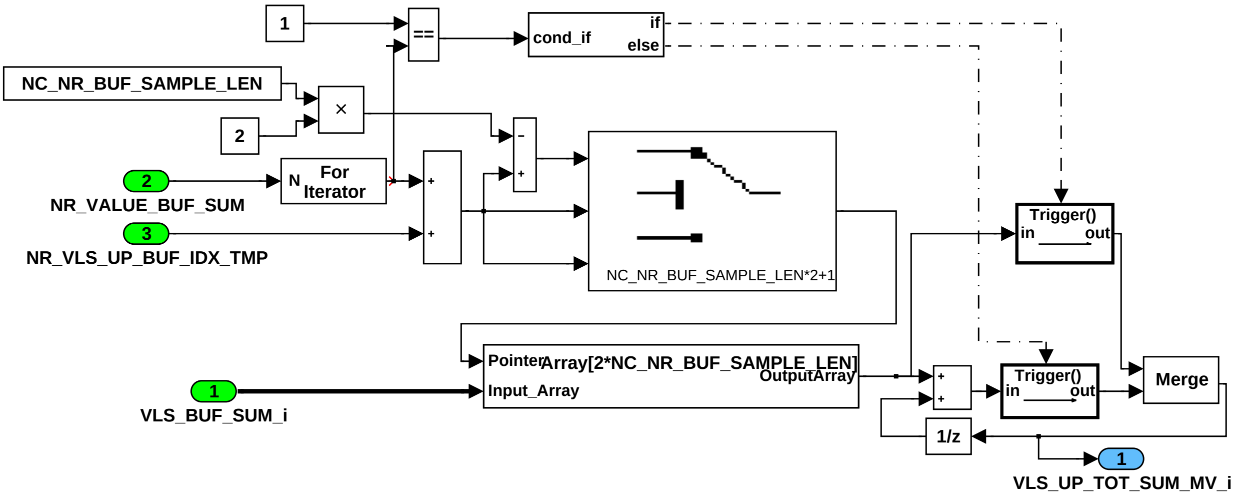


Figure 17 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ CLC_VLS_MV_10MS/ CLC_MV/ CLC_TOT_SUM

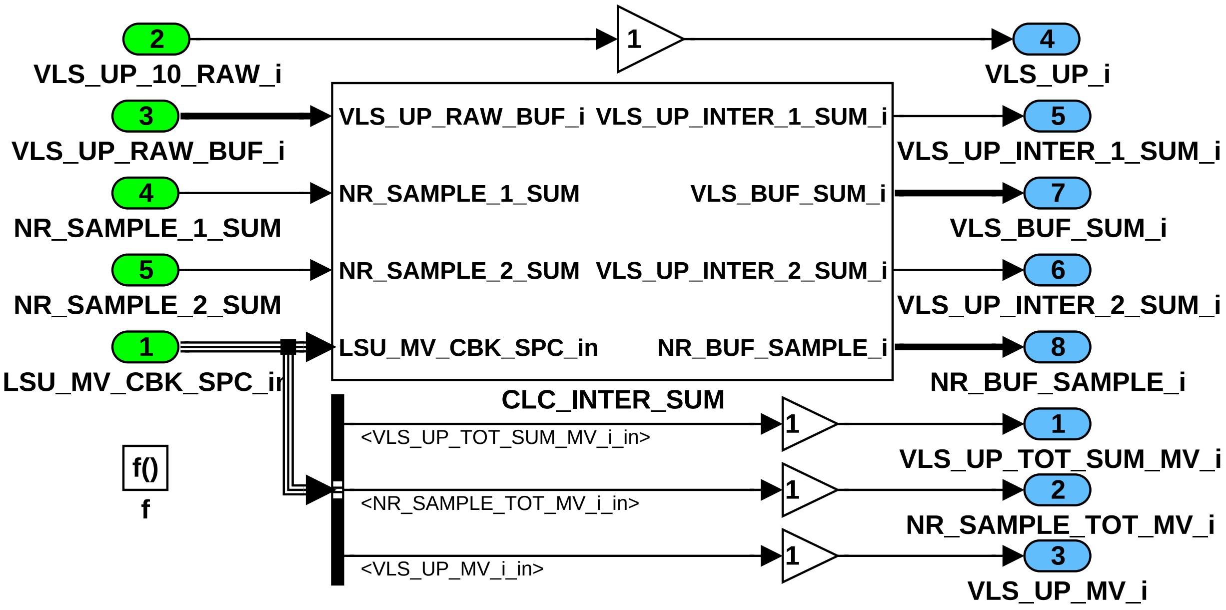


Figure 18 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ FILL_BUF_10MS

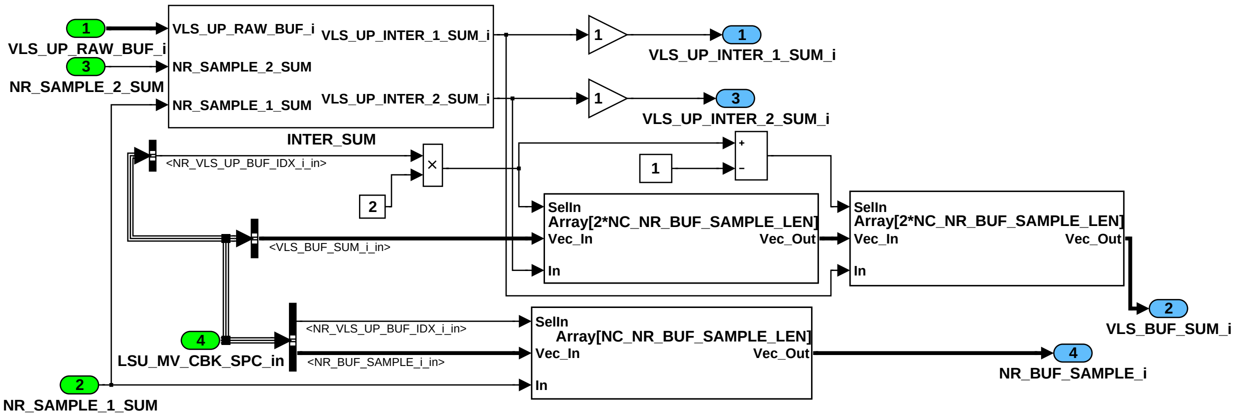


Figure 19 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ FILL_BUF_10MS/ CLC_INTER_SUM

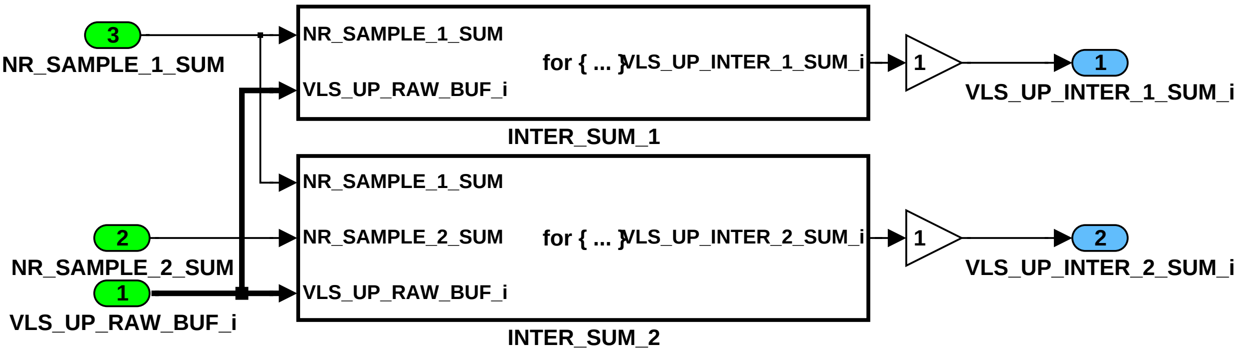


Figure 20 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ FILL_BUF_10MS/ CLC_INTER_SUM/ INTER_SUM

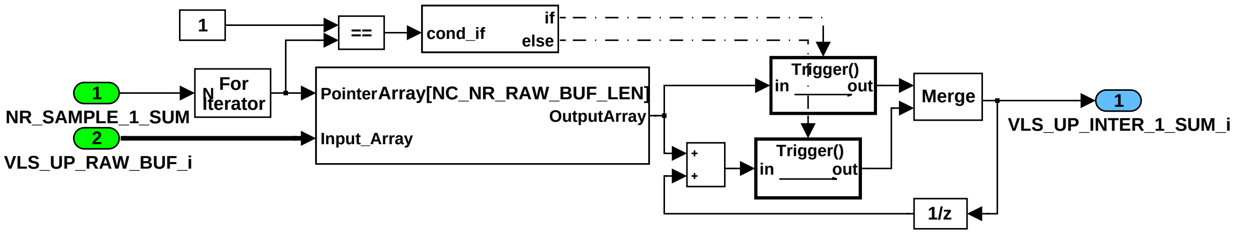


Figure 21 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ FILL_BUF_10MS/ CLC_INTER_SUM/ INTER_SUM/ INTER_SUM_1

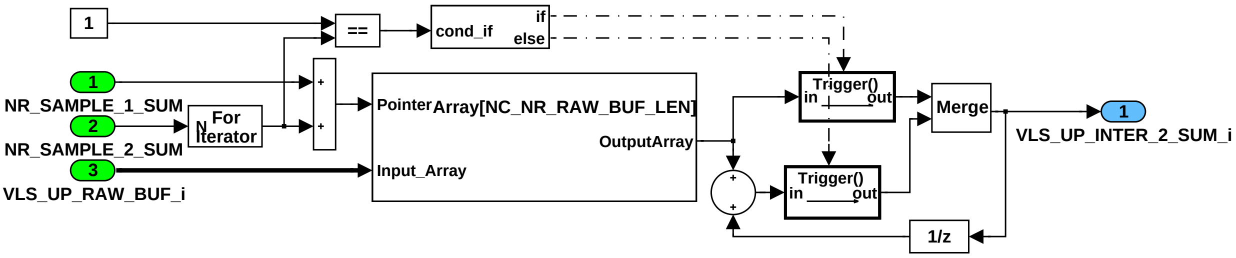


Figure 22 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ CLC_MV_FILL_10MS/ FILL_BUF_10MS/ CLC_INTER_SUM/ INTER_SUM/ INTER_SUM_2

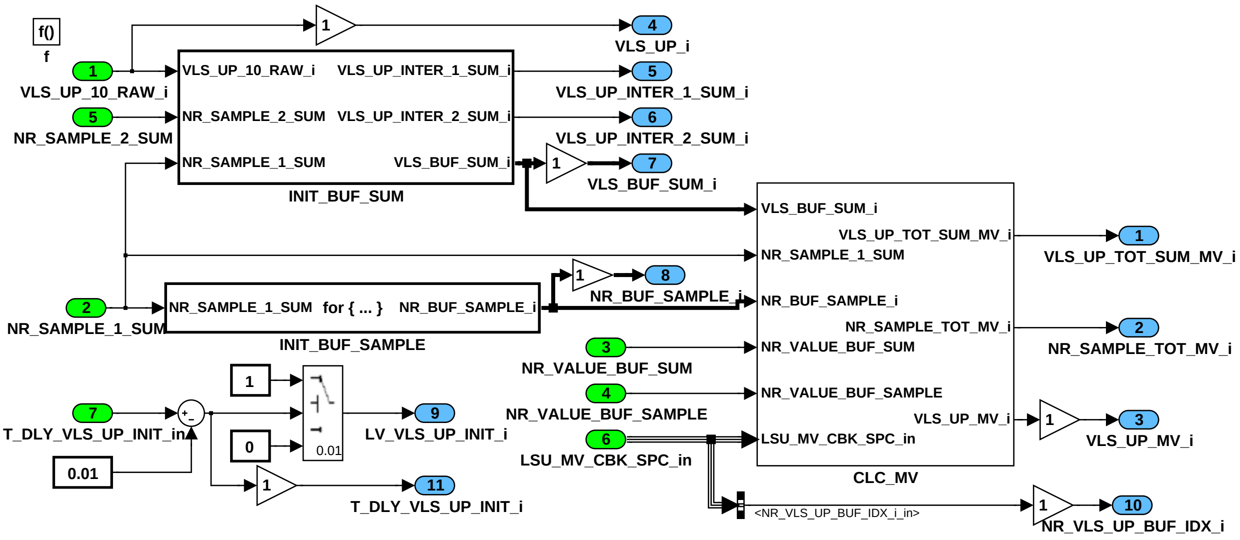


Figure 23 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ INIT_TIMER_G16_G8_10MS

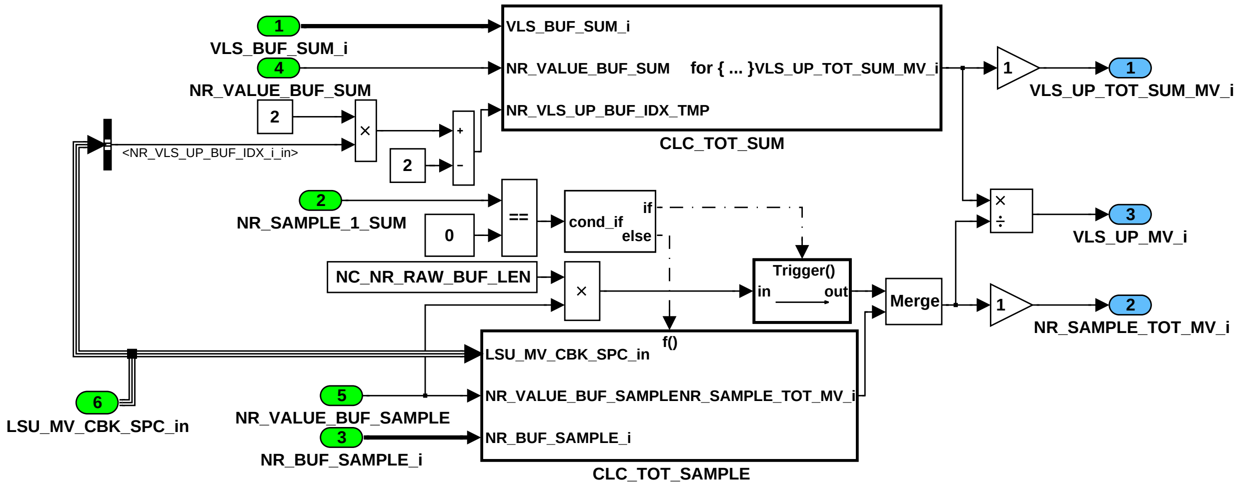


Figure 24 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ INIT_TIMER_G16_G8_10MS/ CLC_MV

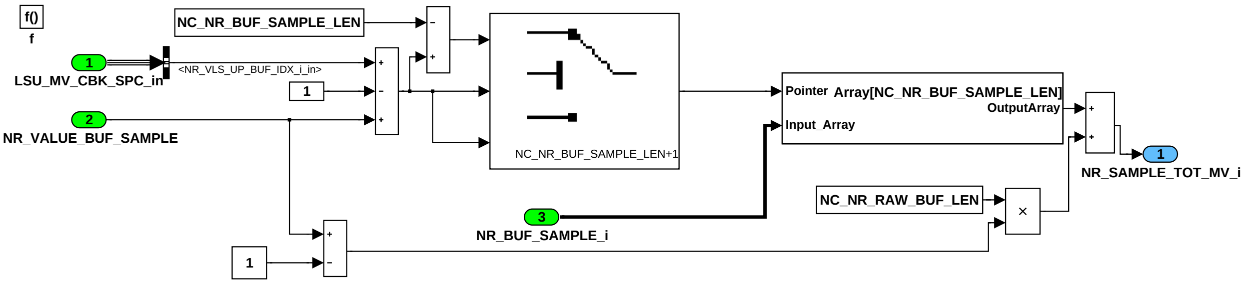


Figure 25 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ INIT_TIMER_G16_G8_10MS/ CLC_MV/ CLC_TOT_SAMPLE

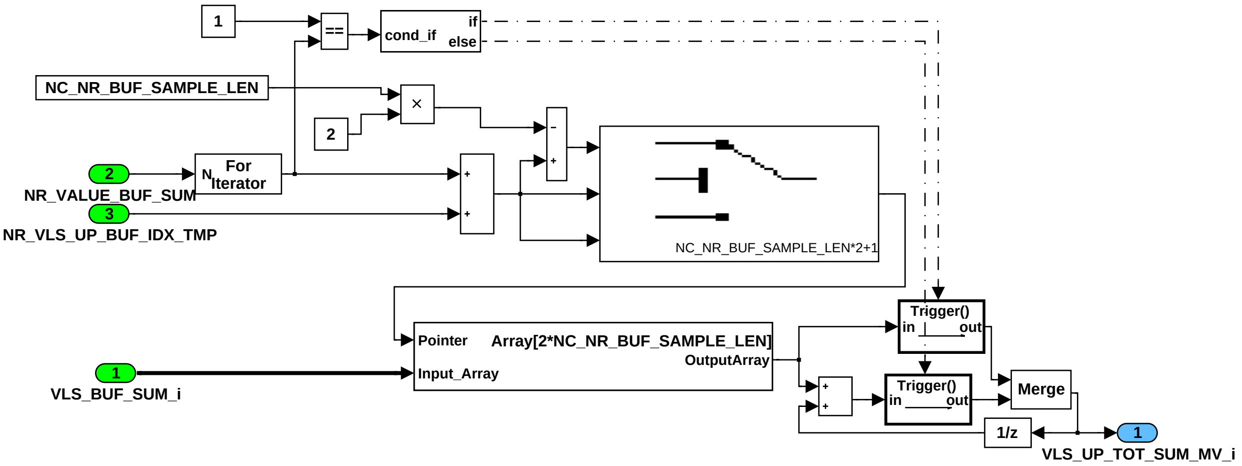


Figure 26 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ INIT_TIMER_G16_G8_10MS/ CLC_MV/ CLC_TOT_SUM

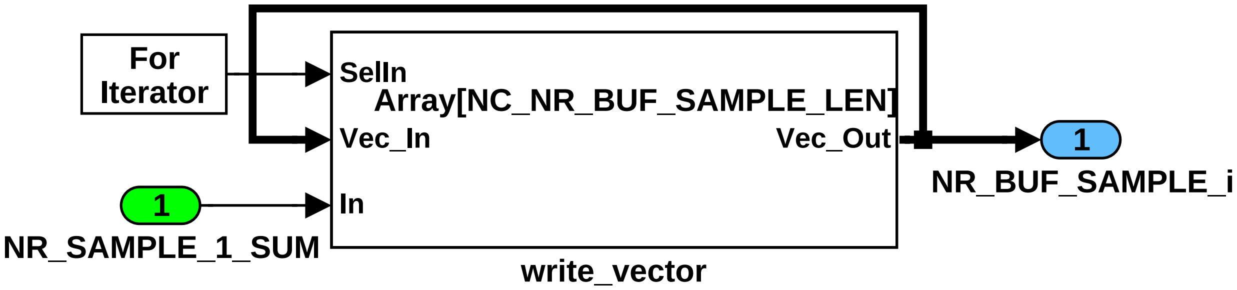


Figure 27 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ INIT_TIMER_G16_G8_10MS/ INIT_BUF_SAMPLE

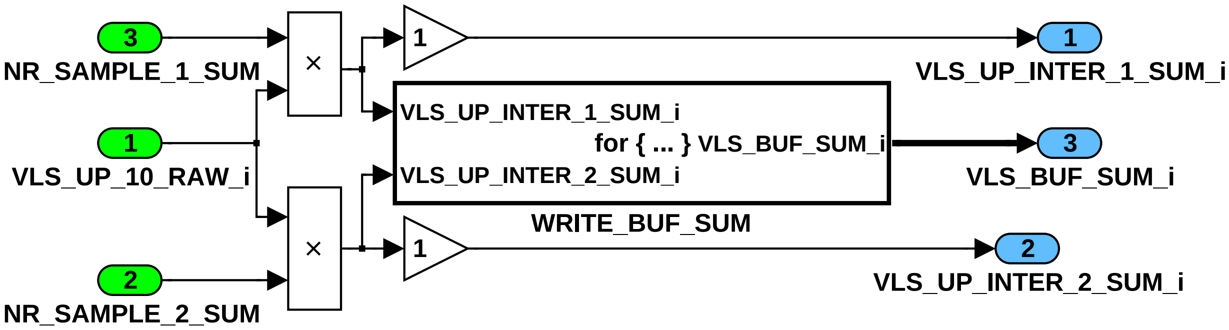


Figure 28 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ INIT_TIMER_G16_G8_10MS/ INIT_BUF_SUM

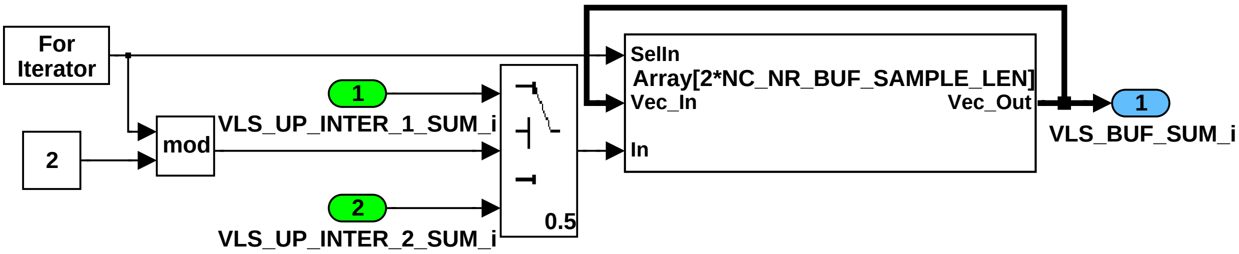


Figure 29 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ INIT_TIMER_G16_G8_10MS/ INIT_BUF_SUM/ WRITE_BUF_SUM

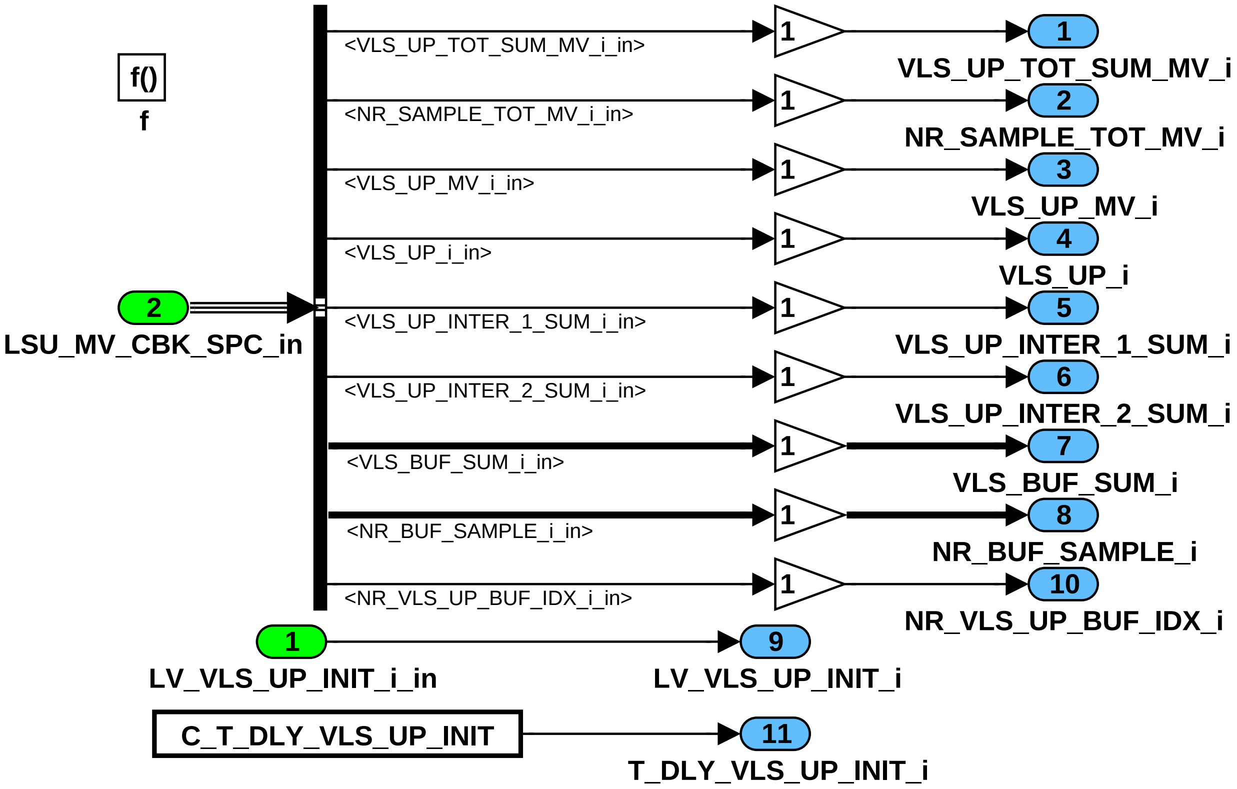


Figure 30 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ RD_NULL_10MS

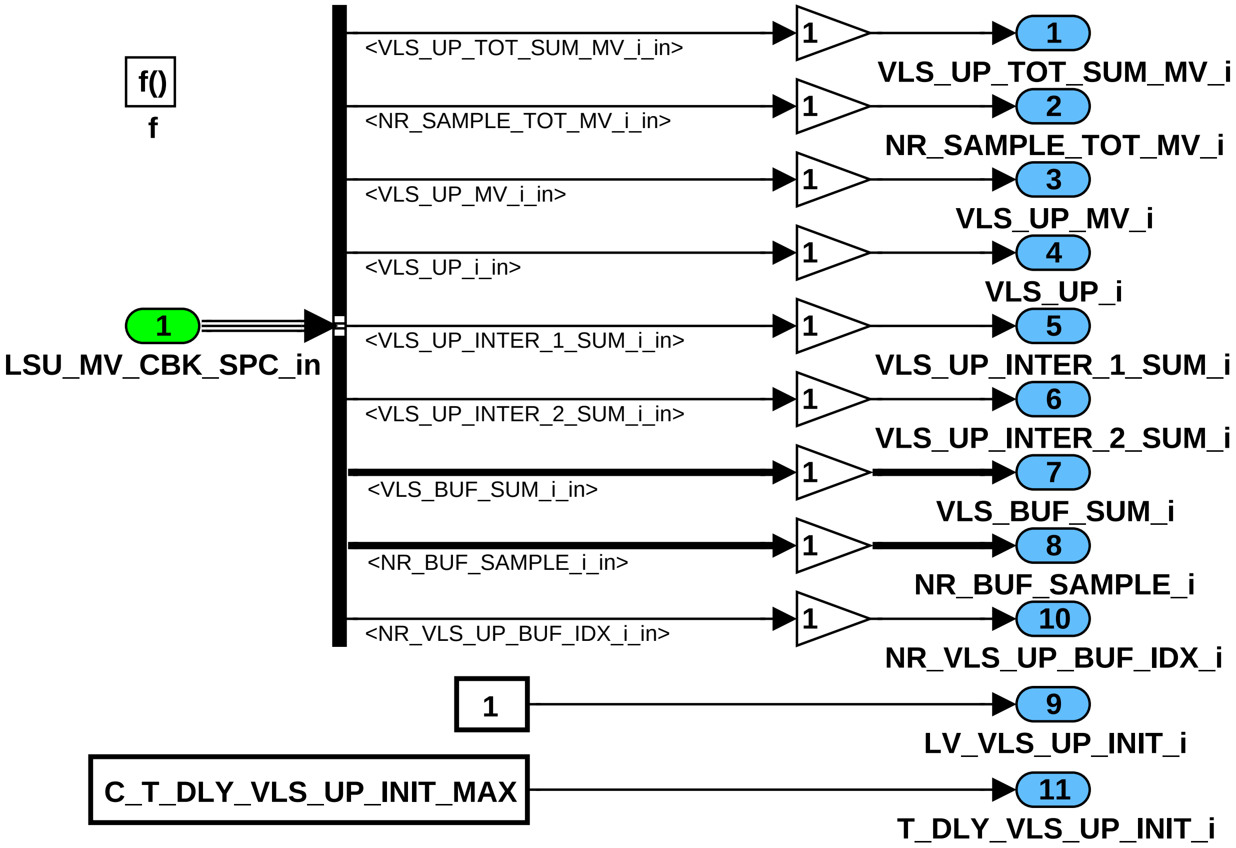


Figure 31 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_CBK_SPC/ WR_NULL_10MS

Figure 32 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ LSU_MV_CBK_MNG/ LSU_MV_SIG_CBK_MNG

Calculation of segment time and number of samples for mean value

The segment time will be calculated after every 10th raw signal value is written in to the buffer. The number of samples required for mean value calculation depend on the segment time. The minimum segment time is fixed using NC_T_SEG_LEN_MIN. If the measured segment time is less than the minimum, then number of samples will be calculated by corresponding multiplication factors.

The default values are

NC_T_SEG_LEN_MIN = 0.01 s

NC_FAC_10_SAMPLE = 2

NC_FAC_5_SAMPLE = 4

2 * 60 * 1000 * no_of_cyl_bank

T_SEG_LEN = --- (ms)

N_in_rpm * no_of_cyl

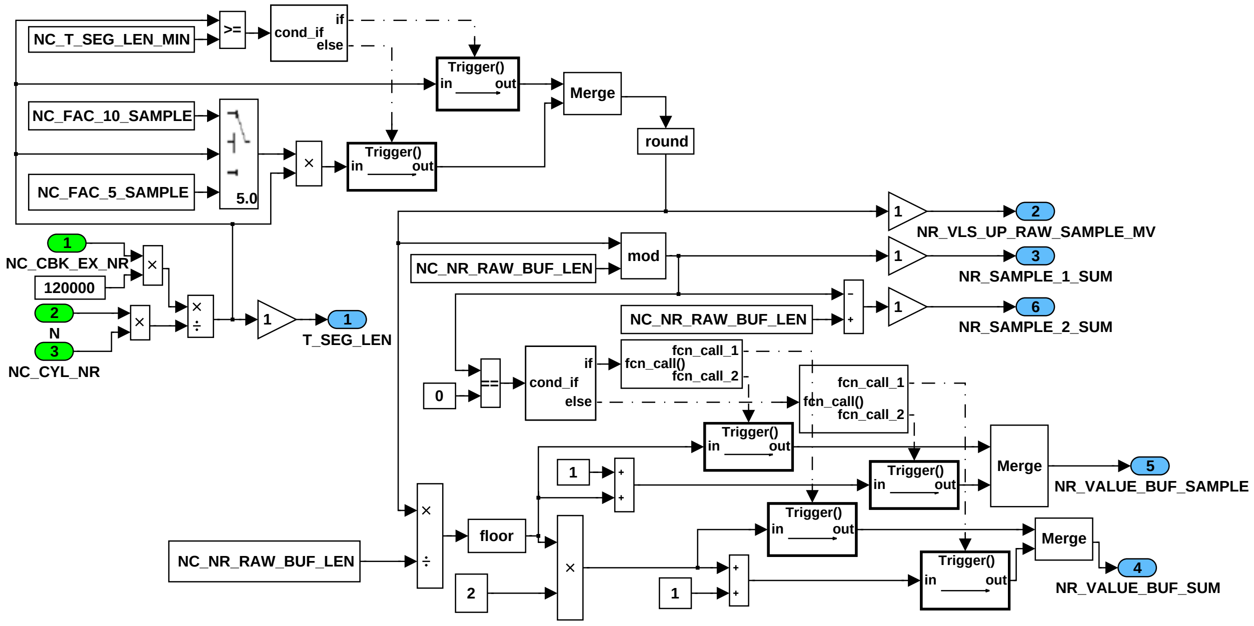


Figure 33 EGCP_SIGCVLSL3/ operate/ CLC_10MS/ T_SEG_CLC

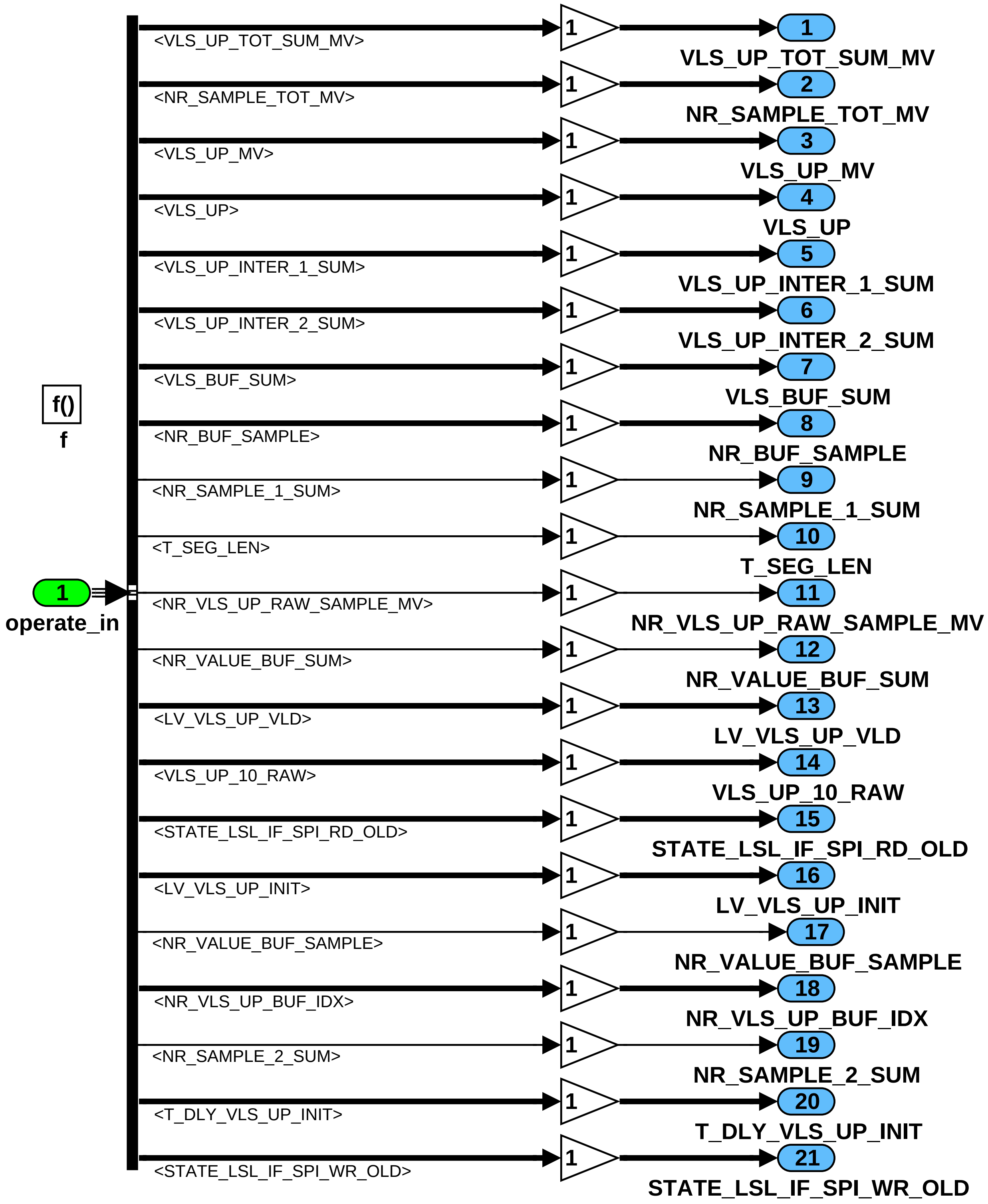


Figure 34 EGCP_SIGCVLSL3/ operate/ NULL_1_9_MS

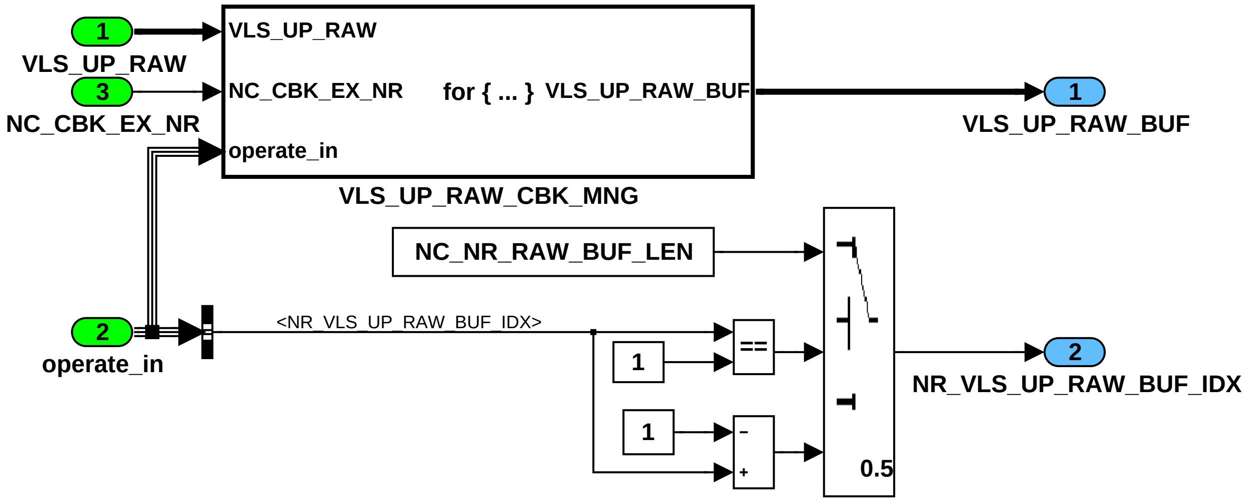


Figure 35 EGCP_SIGCVLSL3/ operate/ VLS_UP_RAW_BUF_1MS

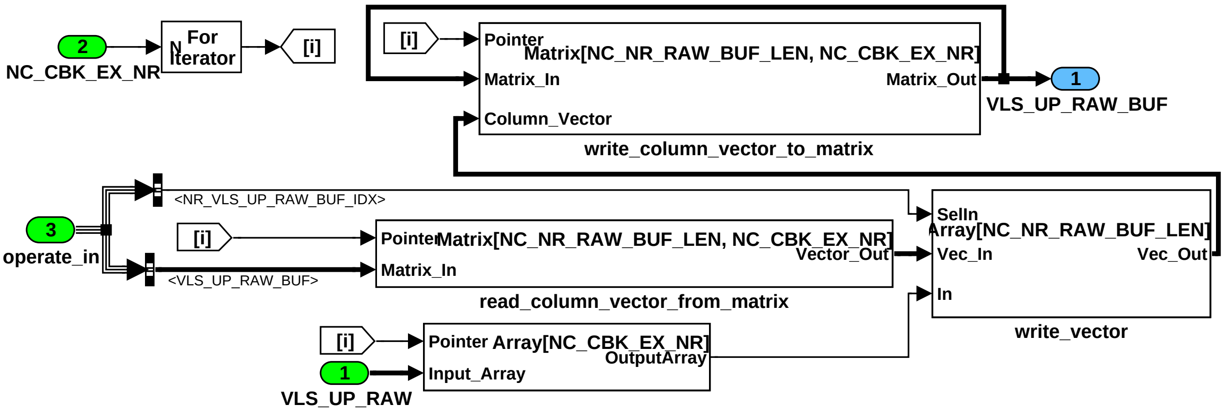


Figure 36 EGCP_SIGCVLSL3/ operate/ VLS_UP_RAW_BUF_1MS/ VLS_UP_RAW_CBK_MNG

16.11.1.2 SUBFUNCTION: reset

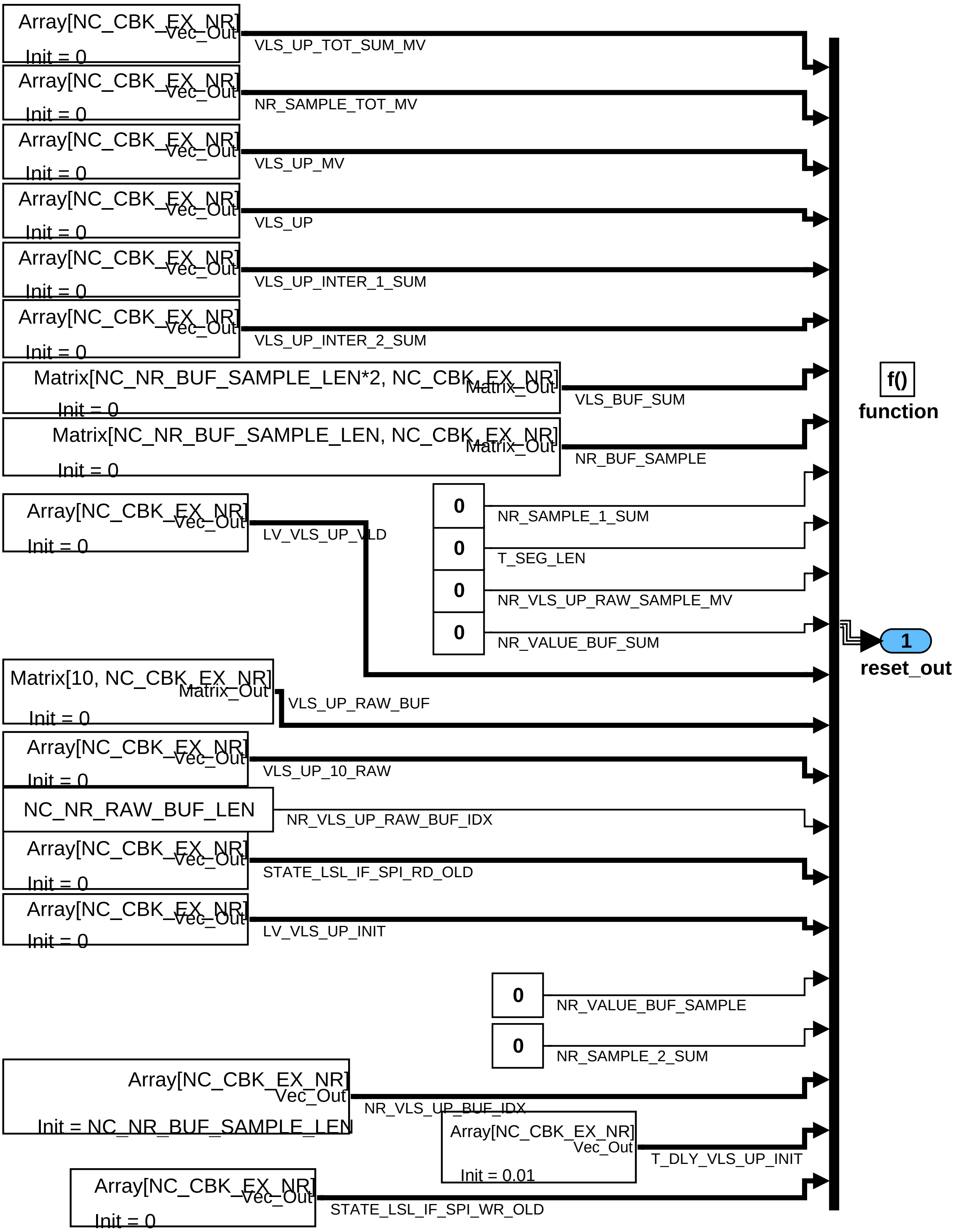


Figure 37 EGCP_SIGCVLSL3/ reset

16.12 Downstream oxygen sensor internal resistance determination

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
R_IT_LS_DOWN[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[Ohm]
Internal resistance of downstream oxygen sensor (Released)					
R_IT_MES_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[Ohm]
Internal resistance of downstream oxygen sensor (Measured)					
R_IT_MDL_LS_DOWN_NEW[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[Ohm]
Internal resistance of downstream oxygen sensor (Modelled for new sensor)					
POW_MMV_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...84.49871	1.2894e-3	[W]
Moving mean value of downstream oxygen sensor heater power					
TTIP_REF_MDL_MMV_LS_DOWN[NC_CBK_EX_NR]	V	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Reference of downstream oxygen sensor ceramic temperature (Modelled for new sensor)					
VLS_DOWN_CUR_PUMP_ON[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Downstream oxygen sensor voltage C_T_CUR_PUMP_ON_LS_DOWN after switching on pump current switch					
VLS_DOWN_CUR_PUMP_OFF[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Downstream oxygen sensor voltage prior to switching on pump current switch					
VP_LS_DOWN_TMP	-	0...7FFFH	0...4.99984	0.1526e-3	[V]
Temporary VLS_DOWN measurement from ADC for conversion to range required in EGCP					
CTR_CYCNR_R_IT_LS_DOWN_VLD[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Handshake counter indicating the number of valid resistance values determined since initial activation					
LV_VLS_DOWN_DRV1_CDN_R_LS[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating that low gradient condition met					
LV_VLS_DOWN_RNG_CDN_R_LS[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating that VLS_DOWN at low gradient condition met					
LV_R_IT_REQ_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that conditions have been met and resistance determination requested					
LV_R_IT_DET_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating that resistance has already been determined (lock out)					
VLS_DOWN_MMV_DRV1_THD_MIN[NC_CBK_EX_NR]	V	8000...7FFFH	-2.5...2.4999	0.0763e-3	[V/10ms]
Lower variable VLS_DOWN gradient threshold					
VLS_DOWN_MMV_DRV1_THD_MAX[NC_CBK_EX_NR]	V	8000...7FFFH	-2.5...2.4999	0.0763e-3	[V/10ms]
Upper variable VLS_DOWN gradient threshold					
T_AFL_CYC_HLD_R_IT_LS_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...2.55	0.01	[s]
Lean mixture cycle time (measured upstream), held until next threshold crossing downstream					
FAC_T_VLS_DOWN_MMV_DRV1[NC_CBK_EX_NR]	V	0...FFH	0...63.75	0.25	[-]
Multiplicative factor modifies DRV1_VLS_DOWN_MMV_MIN_THD dependent on T_VLS_CYC_AFL_HLD					
FAC_T_VLS_DOWN_MMV_DRV1_ABS[NC_CBK_EX_NR]	V	0...FFH	0...15.9375	0.0625	[-]
Multiplicative factor modifies C_VLS_DOWN_MMV_DRV1_ABS_MAX dependent on T_VLS_CYC_AFL_HLD					
TTIP_MES_LS_DOWN[NC_CBK_EX_NR]	V/O	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Downstream oxygen sensor ceramic temperature					
TTIP_MES_TOL_LS_DOWN[NC_CBK_EX_NR]	V	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Downstream oxygen sensor ceramic temperature, Upper limit, Unfiltered					

TTIP_MES_BOL_LS_DOWN[NC_CBK_EX_NR]	V	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Downstream oxygen sensor ceramic temperature, Lower limit, Unfiltered					
TTIP_MES_TOL_MMV_LS_DOWN[NC_CBK_EX_NR]	V	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Downstream oxygen sensor ceramic temperature, Upper limit, Filtered					
TTIP_MES_BOL_MMV_LS_DOWN[NC_CBK_EX_NR]	V	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Downstream oxygen sensor ceramic temperature, Lower limit, Filtered					
T_SAMPLE_R_IT_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Time measured between internal resistance samples					

Input data:

LV_ST_END	VLS_DOWN[NC_CBK_EX_NR]	VLS_DOWN_DRV1_MMV[NC_CBK_EX_NR]	LV_INH_R_IT_LS_DOWN[NC_CBK_EX_NR]
LV_ES	VLS_DOWN_MMV_MIN[NC_CBK_EX_NR]	VLS_DOWN_DRV1_MMV_MIN[NC_CBK_EX_NR]	LV_VLS_DOWN_MMV_LIM[NC_CBK_EX_NR]
TEG_DYN_LS_DOWN[NC_CBK_EX_NR]	VLS_DOWN_MMV_MAX[NC_CBK_EX_NR]	VLS_DOWN_DRV1_ABS_MAX[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]
MAF_KGH	VLS_DOWN_BOL[NC_CBK_EX_NR]	T_AFL_CYC_HLD[NC_CBK_EX_NR]	STATE_LSH_DOWN[NC_CBK_EX_NR]
V_EFC_LSH_DOWN[NC_CBK_EX_NR]	VLS_DOWN_TOL[NC_CBK_EX_NR]	CTR_AFL_CYC[NC_CBK_EX_NR]	LV_VLS_DOWN_MMV_ACT
	NC_CBK_EX_NR		

Import actions:

ACTION_INFR_SetCmdLsDown(IN <i>, **IN** <lv_swi_ls_down>**)**
This action starts the internal resistance determination by "pumping" the sensor.
ACTION_INFR_GetVAdcLsDown(IN <i>, **OUT** <v_adc>**)**
This action sets the oxygen sensor voltage downstream

FUNCTION DESCRIPTION:

General information:

The function shall provide the output variable R_IT_LS_DOWN[i], the internal resistance of the oxygen sensor. The resistance is dependent on the operating temperature of the sensor and may be used for oxygen sensor diagnosis.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1, for exhaust cylinder bank 1
 i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1, for single exhaust cylinder bank.

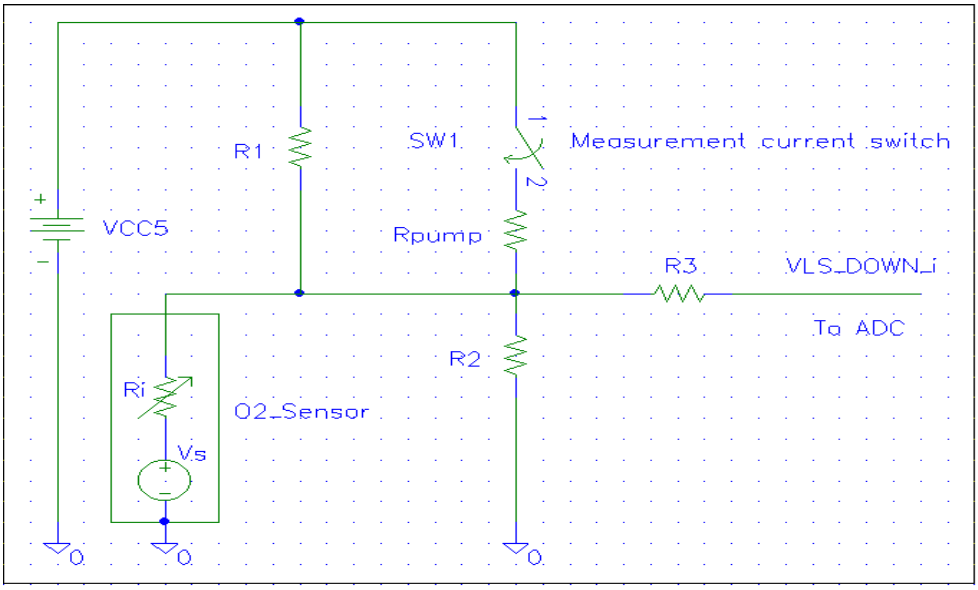


Figure 1 Simplified measurement circuit schematic, Internal resistance determination

Signal flow diagram:

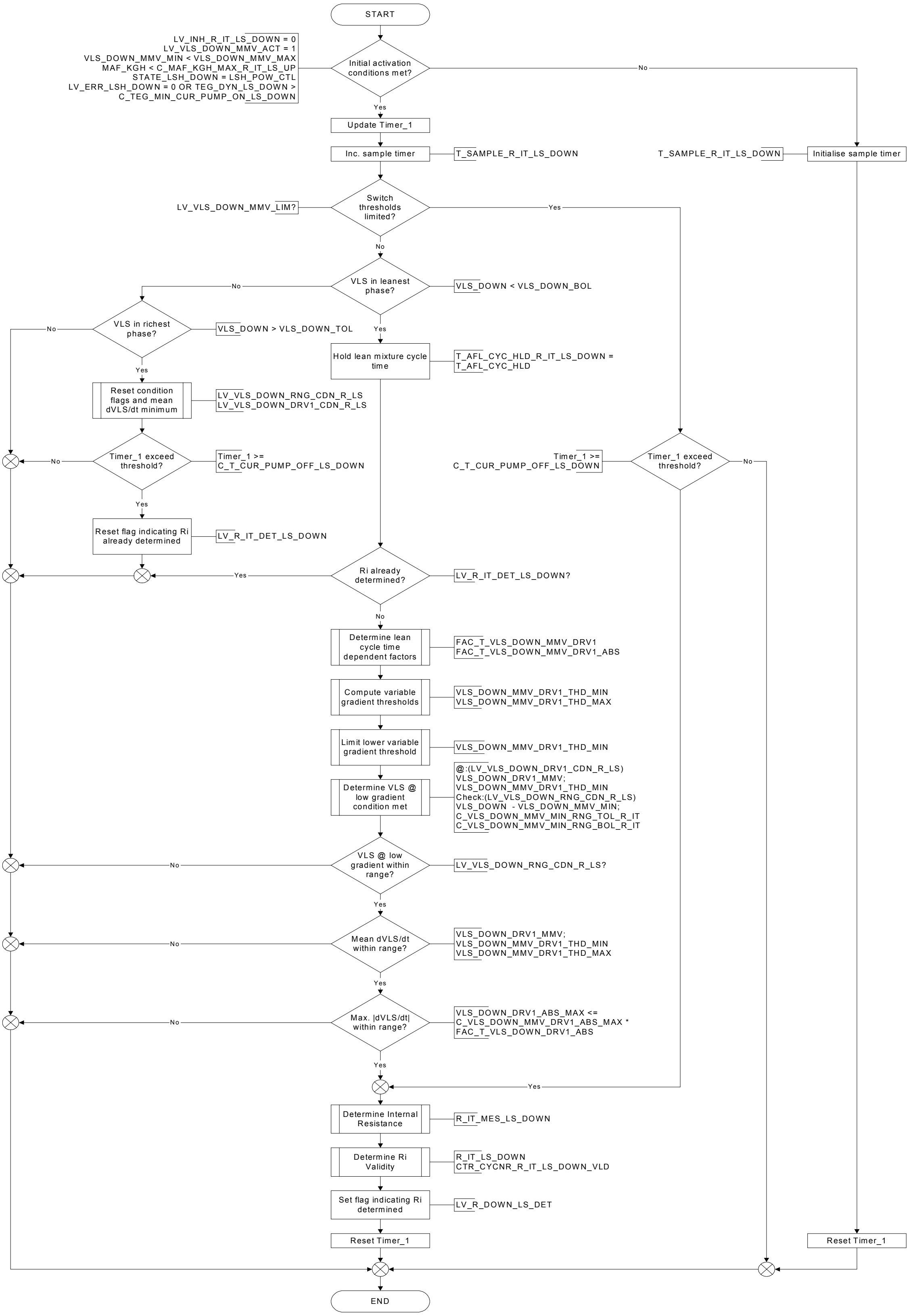


Figure 2 Function flow chart, Internal resistance determination

Description:

The internal resistance of the oxygen sensor shall be determined by use of the oxygen sensor interface circuit *Figure 1* (Capacitors not shown). The internal resistance provides information as to the tip temperature of the sensor and may be used for diagnosis purposes and operative readiness determination.

Under normal conditions, the switch SW1 shall be open and the oxygen sensor signal voltage (VLS_DOWN[i]) is measured by the microcontroller.

The potential divider (resistors R_1 & R_2) shall be permit the operation readiness of the oxygen sensor to be determined and enable electrical fault conditions to be detected.

Periodically, for a limited amount of time, the switch SW1 shall be closed and a voltage of approx. 4.8 V connected through resistor R_{pump} to the oxygen sensor signal connection. This shall cause current to flow into the sensor and through the internal resistance R_i, thereby “pumping” the sensor. The resultant voltage acquired at the microcontroller input shall permit the internal resistance of the sensor to be calculated by use of a formula shown in the formula section below.

Prior to permitting the determination of the internal resistance, a number of conditions shall be determined to be met:

The initial activation conditions shall require that: No function inhibit is requested (LV_INH_R_IT_LS_DOWN[i] = 0); Delay since deactivation since last overrun fuel cut-off passed (LV_VLS_DOWN_MMV_ACT); The initial post initialisation transients have passed for the VLS_DOWN_MMV_xxx[i] inputs (VLS_DOWN_MMV_MIN[i] < VLS_DOWN_MMV_MAX[i]); The engine load threshold has not been exceeded (MAF_KGH < C_MAF_KGH_MAX_R_IT_LS_DOWN); The oxygen sensor heater state LSH_POW_CTL has been reached; No heater OBDI error has been detected in the relevant heater driver (LV_ERR_LSH_DOWN[i]) or should a heater driver fault have been detected, the modelled exhaust gas temperature at downstream sensor position TEG_DYN_LS_DOWN[i] shall exceed the threshold C_TEG_MIN_CUR_PUMP_ON_LS_DOWN.

Once the above conditions have been met, an internal timer (*TIMER_1*) shall be started and timer T_SAMPLE_R_IT_LS_DOWN[i] shall be updated. The latter timer shall be initialised with its maximum value of 655.35 s. This ensures that the temperature limits for the validity function are correctly initialised. The further procedure shall depend on the status of the oxygen sensor signal, as determined by LV_VLS_DOWN_MMV_LIM. Should the VLS_DOWN_MMV_xxx[i] signals be limited, i.e. no or little oxygen sensor amplitude is present, the function shall determine the internal resistance every (C_T_CUR_PUMP_OFF_LS_DOWN + C_T_CUR_PUMP_ON_LS_DOWN + C_T_DLY_CUR_PUMP_LS_DOWN). Otherwise the oxygen sensor signal shall be determined to have significant amplitude and the internal resistance determination shall only be carried out in lean phases where the oxygen sensor signal is relatively flat and stable. This is achieved by the following conditions:

If the oxygen sensor signal is lean (VLS_DOWN[i] < VLS_DOWN_BOL[i]) and neither the resistance determination has been carried out yet in this lean phase nor the determination be locked out due to the minimum off time (LV_R_IT_DET_LS_DOWN[i] = 0) then the function shall determine the lean cycle time dependent factors FAC_T_VLS_DOWN_MMV_DRV1[i] & FAC_T_VLS_DOWN_MMV_DRV1_ABS[i] from the appropriate maps IP_FAC_T_DRV1_VLS_DOWN_MMV & IP_FAC_T_VLS_DOWN_MMV_DRV1_ABS respectively. These factors shall be dependent on the held lean cycle time from the previous lean cycle (T_AFL_CYC_HLD_R_IT_LS_DOWN[i]). Should the cycle time decrease then it is intended that the factors shall increase and vice versa.

The thresholds for the gradient condition shall then be computed. These thresholds (VLS_DOWN_MMV_DRV1_THD_MAX[i] & VLS_DOWN_MMV_DRV1_THD_MIN[i]) are directly proportional to the amplitude of the difference between VLS_DOWN_MMV_xxx[i] signals. Furthermore, VLS_DOWN_MMV_DRV1_THD_MIN[i] shall be modified by factor FAC_T_VLS_DOWN_MMV_DRV1[i]. The factor FAC_T_VLS_DOWN_MMV_DRV1[i] shall be set to 1 unless C_CTR_AFL_CYC_MIN_RI_IT_LS_DOWN AF cycles have passed since the last lambda controller activation, as counted by CTR_AFL_CYC[i]. This ensures that T_AFL_CYC_HLD_R_IT_LS_DOWN[i] is valid.

The variable threshold VLS_DOWN_MMV_DRV1_THD_MIN[i] shall be limited so that its value is not less than that of VLS_DOWN_DRV1_MMV_MIN[i] * C_FAC_VLS_DOWN_MMV_DRV1_THD_MIN.

The function shall then wait for the oxygen sensor signal gradient to exceed the minimum threshold (VLS_DOWN_DRV1_MMV[i] ≥ VLS_DOWN_MMV_DRV1_THD_MIN[i]) and then determine whether the absolute oxygen sensor signal voltage is within a particular range of the VLS_DOWN_MMV_MIN[i] signal (VLS_DOWN_MMV_MIN[i] - C_VLS_DOWN_MMV_MIN_RNG_BOL_R_IT ≤ VLS_DOWN[i] ≤ VLS_DOWN_MMV_MIN[i] + C_VLS_DOWN_MMV_MIN_RNG_TOL_R_IT). If this is the case, the VLS_DOWN[i] shall be considered to be consistent with previous lean phase amplitudes and resistance determination may be carried out. This condition shall only be carried out once per lean phase. i.e. at transition from rich to lean. The fulfilment of the gradient condition is indicated by LV_VLS_DOWN_DRV1_CDN_R_LS[i] and the fulfilment of the range condition by LV_VLS_DOWN_RNG_CDN_R_LS[i].

If the range condition has been met, the gradient is in a particular range (VLS_DOWN_MMV_DRV1_THD_MIN[i] ≤ VLS_DOWN_DRV1_MMV[i] ≤ VLS_DOWN_MMV_DRV1_THD_MAX[i]) and the oxygen sensor signal is determined to be stable in the short term (VLS_DOWN_DRV1_ABS_MAX[i] ≤ C_VLS_DOWN_MMV_DRV1_ABS_MAX * FAC_T_VLS_DOWN_MMV_DRV1_ABS[i]) then resistance determination is requested (LV_R_IT_REQ_LS_DOWN[i] = 1). The factor FAC_T_VLS_DOWN_MMV_DRV1_ABS[i] shall be set to 1 unless C_CTR_AFL_CYC_MIN_RI_IT_LS_DOWN AF cycles have passed since the last lambda controller activation, as counted by CTR_AFL_CYC[i]. This ensures that T_AFL_CYC_HLD_R_IT_LS_DOWN[i] is valid.

Once internal resistance determination has been requested, the Basic SW shall take a single VLS_DOWN[i] sample and store as VLS_DOWN_CUR_PUMP_OFF[i]. SW1 shall be switched ON, at time C_T_CUR_PUMP_ON_LS_DOWN[i] and the VLS_DOWN[i] shall be masked. A second oxygen sensor signal voltage sample shall be taken and stored as VLS_DOWN_CUR_PUMP_ON[i]. SW1 shall then be switched OFF and masking shall continue for a time C_T_DLY_CUR_PUMP_LS_DOWN, after which the VLS_DOWN[i] shall return to normal following of the oxygen sensor signal voltage. Masking shall be discontinued if the oxygen sensor signal voltage exceeds VLS_DOWN_CUR_PUMP_ON[i] + C_VLS_HYS_R_IT_LS_DOWN during the time C_T_DLY_CUR_PUMP_LS_DOWN after switching off SW1. Adding the voltage hysteresis shall prevent the function to discard lambda sensor voltage masking if the first VLS_DOWN[i] value after switching off the pump current is larger than the last VLS_DOWN[i] value while pumping due to signal noise.

The internal resistance R_IT_MES_LS_DOWN[i] shall be determined using the formula stated with the main variables being oxygen sensor voltage prior to pumping, VLS_DOWN_CUR_PUMP_OFF[i], and signal voltage during pumping, VLS_DOWN_CUR_PUMP_ON[i].

The internal resistance shall be calculated every time the oxygen sensor is pumped. For each new resistance value detected, a counter CTR_CYCNR_R_IT_LS_DOWN_VLD[i] shall be incremented to indicate to other functions that a new resistance value is available.

NOTE: During the time where SW1 is switched on, (LV_R_IT_REQ_LS_DOWN[i]=1) the output voltage of the oxygen sensor (VLS_DOWN[i]) does not represent the true A/F ratio and is therefore invalid and shall not be used as such. The basic software shall mask the response of the sensor to pumping by keeping the voltage constant during this time by using the last VLS_DOWN[i] sample prior to activating SW1.

The internal resistance shall be calculated every time the oxygen sensor is pumped. For each new resistance value detected, the validity of the resistance shall be checked:

Should the validity function not be inhibited (LC_R_IT_VLD_INH_LS_DOWN = 0), then the tip temperature of the ceramic shall be determined from a map. TTIP_MES_LS_DOWN[i] represents the nominal temperature, TTIP_MES_TOL_LS_DOWN[i] represents the maximum temperature (R_IT_MES_LS_DOWN[i] * (1 – C_FAC_R_IT_ERR_NEG_LS_DOWN)), i.e. temperature for resistance with negative measurement error and TTIP_MES_BOL_LS_DOWN[i] represents the minimum temperature (R_IT_MES_LS_DOWN[i] * (1 + C_FAC_R_IT_ERR_NEG_LS_DOWN)), i.e. temperature for resistance with positive measurement error. A low pass filter shall then be applied to the upper and lower temperature limits to represent the maximum possible temperature gradient between resistance samples. Dependent on the direction of change of resistance between two measurements, either constant C_CRLC_TTIP_RISE_R_IT_LS_DOWN (rising temp.) or C_CRLC_TTIP_FALL_R_IT_LS_DOWN (falling temp.) shall be applicable. The final filter constant used, limited to a maximum of 1, shall be the aforementioned constant multiplied by the time between resistance samples (T_SAMPLE_R_IT_LS_DOWN[i]). This shall be to compensate for the irregular sample time.

Should then the ceramic tip temperature fall within the temperature limit bounds, the measured resistance shall be considered to be valid, shall be passed to the output variable R_IT_LS_DOWN[i], the counter CTR_CYCNR_R_IT_LS_DOWN_VLD[i] shall be incremented to indicate to other functions that a new resistance value is available and the T_SAMPLE_R_IT_LS_DOWN[i] shall be reset. If the above temperature condition is not met, R_IT_LS_DOWN[i] shall be maintained at its last value and the filtered temperature limits shall be reset to their previous values, i.e. the resistance value shall be ignored.

Should the validity function be inhibited then for each new resistance the R_IT_MES_LS_DOWN[i] shall be passed to R_IT_LS_DOWN[i] and the counter CTR_CYCNR_R_IT_LS_DOWN_VLD[i] shall be incremented.

In the case of significant oxygen sensor amplitude (LV_VLS_DOWN_MMV_LIM = 0) and the sensor signal be in the rich phase (VLS_DOWN[i] > VLS_DOWN_TOL[i]), the flags LV_VLS_DOWN_RNG_CDN_R_LS[i] & LV_VLS_DOWN_DRV1_CDN_R_LS[i] shall be reset. Furthermore, if these conditions have been met and the timer *TIMER_1* exceeds C_T_CUR_PUMP_OFF_LS_DOWN then LV_R_IT_DET_LS_DOWN[i] shall be reset, thereby permitting the internal resistance to be determined in the next cycle should the appropriate conditions be met.

For reference reasons of OBD II diagnosis the value R_IT_MDL_LS_DOWN_NEW[i] is calculated out of heating power, exhaust gas temperature and MAF_KGH.

Application conditions:

Recurrence: *T_SAMPLE_1 = 10 ms*

The function shall be carried out once every 10 ms. The internal resistance value shall only be calculated after “pumping” the oxygen sensor. See recurrence there. Reference internal rsistance shall be calculated with recurrence 1 s.

Initialisation:

at reset:

```
ACTION_INFR_SetCmdLsDown(
                        IN      i
                        IN      0)
```

The following initialisation shall be carried out after a RESET and upon leaving the engine state Engine Stop (LV_ES).

R_IT_LS_DOWN[i] = 65535
R_IT_MES_LS_DOWN[i] = 65535
R_IT_MDL_LS_DOWN_NEW[i] = 65535
POW_MMV_LSH_DOWN[i] = 0
VLS_DOWN_CUR_PUMP_ON[i] = 0
VLS_DOWN_CUR_PUMP_OFF[i] = 0
CTR_CYCNR_R_IT_LS_DOWN_VLD[i] = 0
LV_VLS_DOWN_DRV1_CDN_R_LS[i] = 0
LV_VLS_DOWN_RNG_CDN_R_LS[i] = 0
LV_R_IT_REQ_LS_DOWN[i] = 0
LV_R_IT_DET_LS_DOWN[i] = 0
VLS_DOWN_MMV_DRV1_THD_MIN[i] = 0
VLS_DOWN_MMV_DRV1_THD_MAX[i] = 0
T_AFL_CYC_HLD_R_IT_LS_DOWN[i] = 0
FAC_T_VLS_DOWN_MMV_DRV1[i] = 0
FAC_T_VLS_DOWN_MMV_DRV1_ABS[i] = 0
TTIP_REF_MDL_MMV_LS_DOWN[i] = 0
TTIP_MES_LS_DOWN[i] = 0
TTIP_MES_TOL_LS_DOWN[i] = 0
TTIP_MES_BOL_LS_DOWN[i] = 0
TTIP_MES_TOL_MMV_LS_DOWN[i] = 0
TTIP_MES_BOL_MMV_LS_DOWN[i] = 0
T_SAMPLE_R_IT_LS_DOWN[i] = 655.35

NOTE: All “N-1” variables shall also be initialised to the associated value specified above.

Activation / Deactivation:

```
If      LV_ST_END = 1
then    Function activated
else    Function deactivated
endif
```

Formula section:

```
If          LV_INH_R_IT_LS_DOWN[i] = 0
and         LV_VLS_DOWN_MMV_ACT = 1
```

```
and     VLS_DOWN_MMV_MIN[i] < VLS_DOWN_MMV_MAX[i]
and     MAF_KGH < C_MAF_KGH_MAX_R_IT_LS_DOWN
and     STATE_LSH_DOWN[i] = "LSH_POW_CTL"
and     (LV_ERR_LSH_DOWN[i] = 0
or      TEG_DYN_LS_DOWN[i] > C_TEG_MIN_CUR_PUMP_ON_LS_DOWN)
then    Update TIMER_1
        Increment T_SAMPLE_R_IT_LS_DOWN[i]
        If      LV_R_IT_REQ_LS_DOWN[i] = 0
        then    If      LV_VLS_DOWN_MMV_LIM[i] = 0
                then    If      VLS_DOWN[i] < VLS_DOWN_BOL[i]
                        then    T_AFL_CYC_HLD_R_IT_LS_DOWN[i] =
                                T_AFL_CYC_HLD[i]
                                If      LV_R_IT_DET_LS_DOWN[i] = 0
                                then    "Determine lean cycle time dependent factors"
                                        "Compute variable gradient thresholds"
                                        "Limit lower variable gradient threshold"
                                        "Determine VLS_DOWN at low gradient condition met"
                                        If      LV_VLS_DOWN_RNG_CDN_R_LS[i] = 1
                                        and     VLS_DOWN_DRV1_MMV[i] ≥
                                                VLS_DOWN_MMV_DRV1_THD_MIN[i]
                                        and     VLS_DOWN_DRV1_MMV[i] ≤
                                                VLS_DOWN_MMV_DRV1_THD_MAX[i]
                                        and     VLS_DOWN_DRV1_ABS_MAX[i] ≤
                                                C_VLS_DOWN_MMV_DRV1_ABS_MAX *
                                                FAC_T_VLS_DOWN_MMV_DRV1_ABS[i])
                                        then    LV_R_IT_REQ_LS_DOWN[i] = 1
                                                VLS_DOWN_CUR_PUMP_OFF[i] =
                                                VLS_DOWN[i]
                                                Switch SW1 ON
                                                "Determine internal resistance" started
                                                ACTION_INFR_SetCmdLsDown(
                                                            IN      i
                                                            IN      1)

                                        endif
                                endif
```

```
                        else    If      VLS_DOWN[i] > VLS_DOWN_TOL[i]
                                then    LV_VLS_DOWN_RNG_CDN_R_LS[i] = 0
                                        LV_VLS_DOWN_DRV1_CDN_R_LS[i] = 0
                                        If      TIMER_1 ≥
                                                C_T_CUR_PUMP_OFF_LS_DOWN
                                        then    LV_R_IT_DET_LS_DOWN[i] = 0
                                        endif
                                endif
                        endif
                else    If      (TIMER_1 ≥ C_T_CUR_PUMP_OFF_LS_DOWN)
                        then    LV_R_IT_REQ_LS_DOWN[i] = 1
                                VLS_DOWN_CUR_PUMP_OFF[i] = VLS_DOWN[i]
                                Switch SW1 ON
                                "Determine internal resistance" started
                                ACTION_INFR_SetCmdLsDown(
                                                IN      i
                                                IN      1)

                        endif
                endif
        endif
        If      LV_R_IT_REQ_LS_DOWN[i] = 1 -> 0
        then    "Calculation of Internal resistance"
                LV_R_IT_DET_LS_DOWN[i] = 1
                Reset TIMER_1
        endif
else    Reset TIMER_1
        T_SAMPLE_R_IT_LS_DOWN[i] = 655.35 (i.e. FFFFH)
        LV_R_IT_DET_LS_DOWN[i] = 0
        LV_VLS_DOWN_DRV1_CDN_R_LS[i] = 0
        LV_VLS_DOWN_RNG_CDN_R_LS[i] = 0
endif
"Calculation of internal resistance reference"
```

NOTES:

1. ***TIMER_1 refers to a SW internal timer and not a visible output data variable.***
2. ***T_SAMPLE_R_LS_DOWN[i] shall not be permitted to overflow. When the specified maximum value has been reached, this shall be held until the timer is reset by the function.***

Calculation of internal resistance reference:

% This calculation is carried out with the recurrence: T_SAMPLE_2 = 1 s.

POW_MMV_LSH_DOWN[i]$_n$ = POW_MMV_LSH_DOWN[i]$_{n-1}$ *
(1 - C_CRLC_POW_MMV_LSH_DOWN) +
C_CRLC_POW_MMV_LSH_DOWN *
V_EFC_LSH_DOWN[i]^2 * T_SAMPLE_2 * [W/(s*V$^{2)}$]

TTIP_REF_MDL_MMV_LS_DOWN[i]$_n$ = TTIP_REF_MDL_MMV_LS_DOWN[i]$_{n-1}$ *
(1 - IP_CRLC_TTIP_REF_MDL_LS_DOWN) +
IP_CRLC_TTIP_REF_MDL_LS_DOWN * TEG_DYN_LS_DOWN[i]
% IP (MAF_KGH)

R_IT_MDL_LS_DOWN_NEW[i] = IP_R_IT_MDL_LS_DOWN_NEW
% IP (POW_MMV_LSH_DOWN[i]; TTIP_REF_MDL_MMV_LS_DOWN[i])

Determine lean cycle time dependent factors:

```
If   (CTR_AFL_CYC[i] ≥ C_CTR_AFL_CYC_MIN_RI_IT_LS_DOWN)
then FAC_T_VLS_DOWN_MMV_DRV1[i] =
     IP_FAC_T_DRV1_VLS_DOWN_MMV (T_AFL_CYC_HLD_R_IT_LS_DOWN[i])
     FAC_T_VLS_DOWN_MMV_DRV1_ABS[i] =
     IP_FAC_T_VLS_DOWN_MMV_DRV1_ABS (T_AFL_CYC_HLD_R_IT_LS_DOWN[i])
else FAC_T_VLS_DOWN_MMV_DRV1[i] = 1
     FAC_T_VLS_DOWN_MMV_DRV1_ABS[i] = 1
endif.
```

Compute variable gradient thresholds:

VLS_DOWN_MMV_DRV1_THD_MAX[i] = C_VLS_DOWN_MMV_DRV1_THD_MAX *
C_FAC_VLS_DOWN_MMV_DRV1_THD *
(VLS_DOWN_MMV_MAX[i] -
VLS_DOWN_MMV_MIN[i])

VLS_DOWN_MMV_DRV1_THD_MIN[i] = C_VLS_DOWN_MMV_DRV1_THD_MIN *
C_FAC_VLS_DOWN_MMV_DRV1_THD *
(VLS_DOWN_MMV_MAX[i] -
VLS_DOWN_MMV_MIN[i]) *
FAC_T_VLS_DOWN_MMV_DRV1[i]

Limit lower variable gradient threshold:

VLS_DOWN_MMV_DRV1_THD_MIN[i] = MAX(VLS_DOWN_MMV_DRV1_THD_MIN[i],
VLS_DOWN_DRV1_MMV_MIN[i] *
C_FAC_VLS_DOWN_MMV_DRV1_THD_MIN)

Determine VLS_DOWN at low gradient condition met:

```
If  (LV_VLS_DOWN_DRV1_CDN_R_LS[i] = 0) &
    (VLS_DOWN_DRV1_MMV[i] ≥ VLS_DOWN_MMV_DRV1_THD_MIN[i])
```

```
then LV_VLS_DOWN_DRV1_CDN_R_LS[i] = 1
     If  (VLS_DOWN[i] ≤
         (VLS_DOWN_MMV_MIN[i] + C_VLS_DOWN_MMV_MIN_RNG_TOL_R_IT) ) &
         (VLS_DOWN[i] ≥
         (VLS_DOWN_MMV_MIN[i] - C_VLS_DOWN_MMV_MIN_RNG_BOL_R_IT) )
     then LV_VLS_DOWN_RNG_CDN_R_LS[i] = 1
     else LV_VLS_DOWN_RNG_CDN_R_LS[i] = 0
     endif
endif.
```

NOTE: VLS_DOWN_MMV_MIN[i] + C_VLS_DOWN_MMV_MIN_RNG_TOL_R_IT shall not cause an overflow and VLS_DOWN_MMV_MIN[i] - C_VLS_DOWN_MMV_MIN_RNG_BOL_R_IT shall not cause an underflow.

Determine internal resistance:

Recurrence: T_SAMPLE_3 = 1 ms

```
If        LV_R_IT_REQ_LS_DOWN[i] = 1
then
```

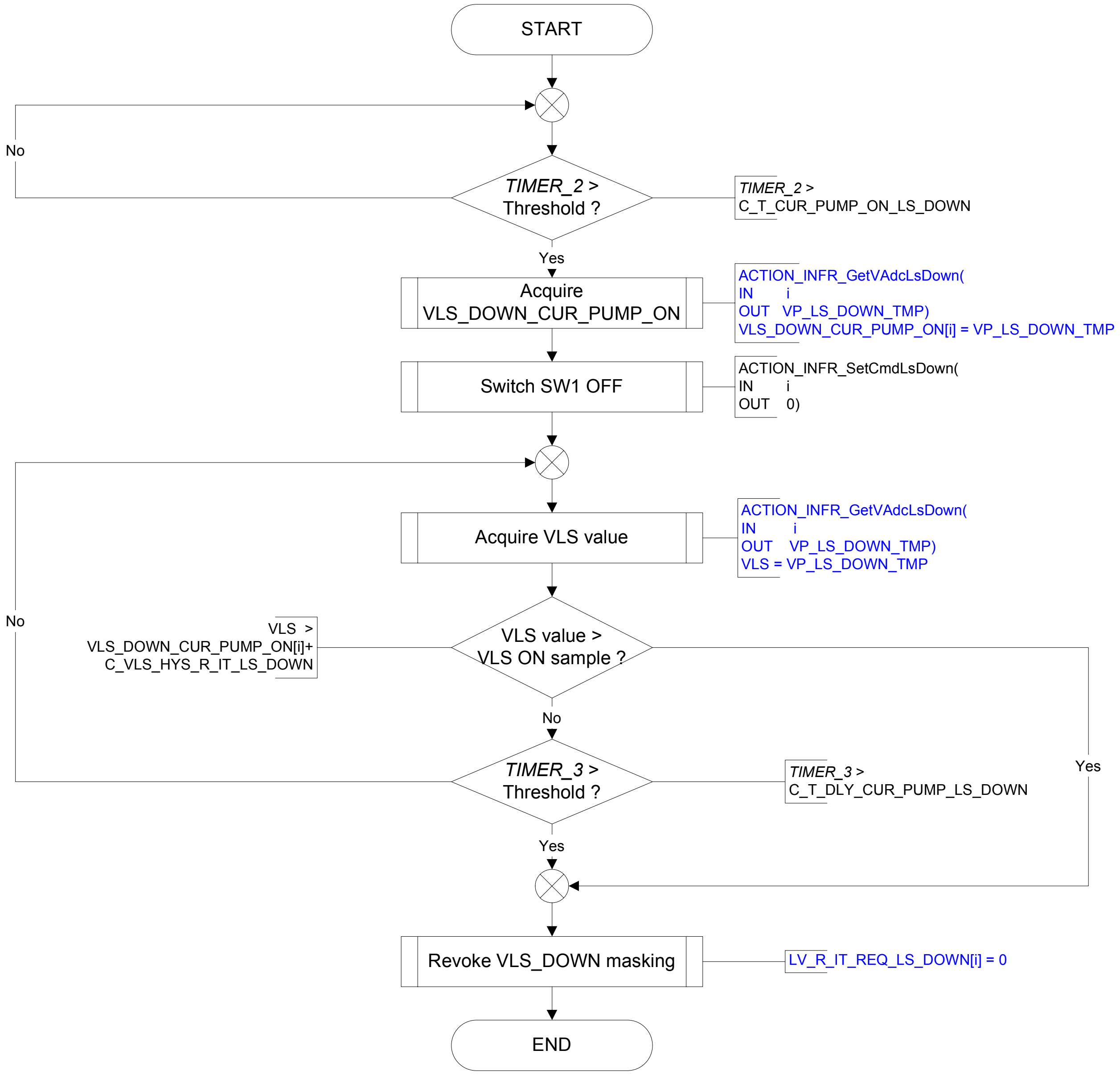


Figure 3 Function flow chart, Internal resistance determination

endif

Calculation of Internal resistance:

The internal resistance shall be determined as follows:

R_IT_MES_LS_DOWN[i] = NC_R_REF_LS_DOWN * (VLS_DOWN_CUR_PUMP_ON[i] - VLS_DOWN_CUR_PUMP_OFF[i]) / {NC_VLS_DOWN_CUR_PUMP_REF - (VLS_DOWN_CUR_PUMP_ON[i] - VLS_DOWN_CUR_PUMP_OFF[i])* NC_FAC_R_REF_LS_DOWN - VLS_DOWN_CUR_PUMP_ON[i]}

Determine internal resistance validity:

The following function shall be carried once out for every new resistance value determined.

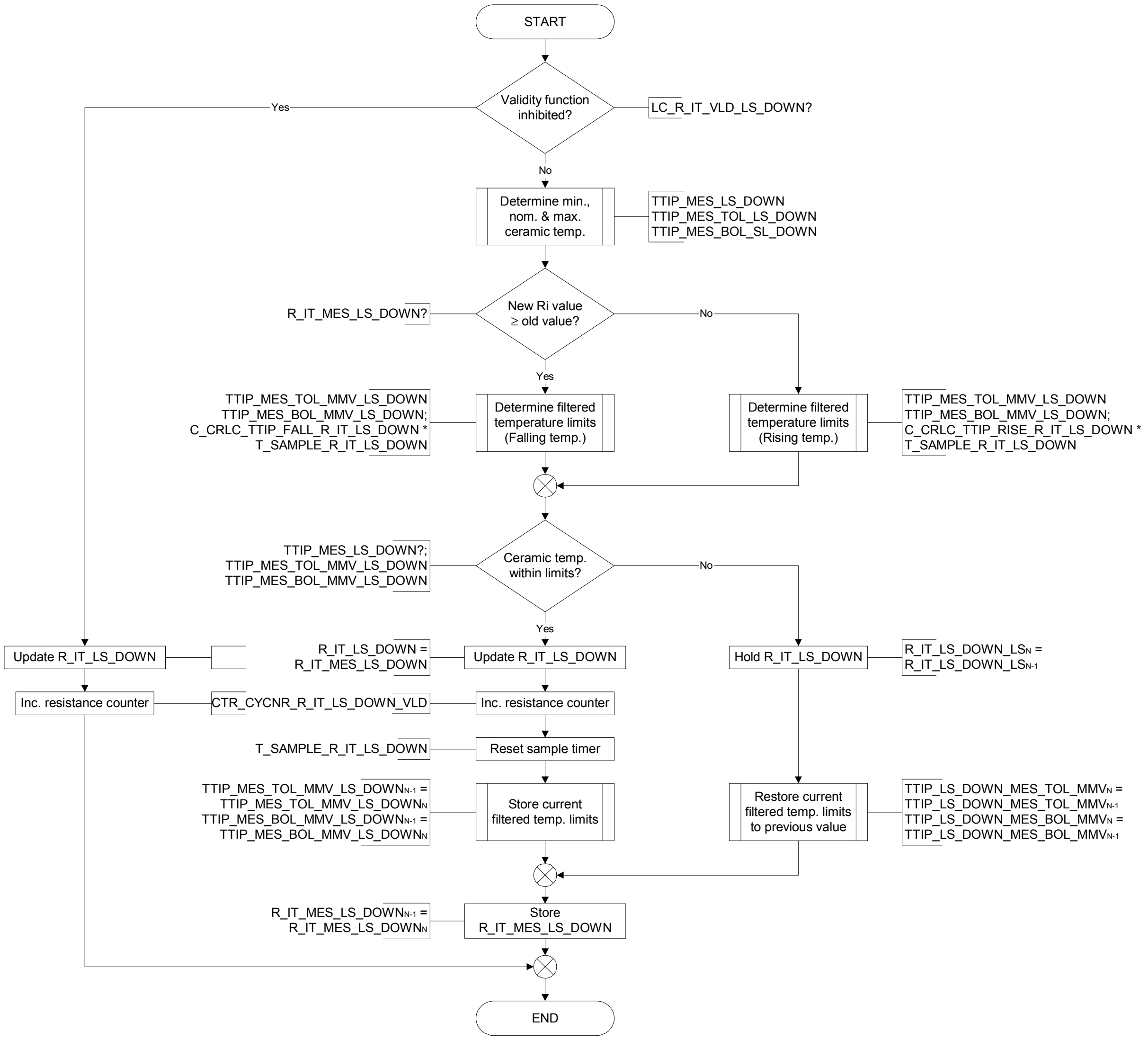


Figure 4 Function flow chart, Internal resistance validity determination

```
If (LC_R_IT_VLD_INH_LS_DOWN = 0)
then
    TTIP_MES_LS_DOWN[i] = IP_TTIP_MES_LS_DOWN (R_IT_MES_LS_DOWN[i])
    TTIP_MES_TOL_LS_DOWN[i] = IP_TTIP_MES_LS_DOWN (R_IT_MES_LS_DOWN[i] *
    (1 - C_FAC_R_IT_ERR_NEG_LS_DOWN))
    TTIP_MES_BOL_LS_DOWN[i] = IP_TTIP_MES_LS_DOWN (R_IT_MES_LS_DOWN[i] *
    (1+ C_FAC_R_IT_ERR_POS_LS_DOWN))
    If  (R_IT_MES_LS_DOWN[i]$_N$ ≥ R_IT_MES_LS_DOWN[i]$_{N-1}$)
    then
        TTIP_MES_TOL_MMV_LS_DOWN[i]$_N$ = TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N-1}$ +
        (TTIP_MES_TOL_LS_DOWN[i] - TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N-1}$) *
        MIN(1, C_CRLC_TTIP_FALL_R_IT_LS_DOWN * T_SAMPLE_R_IT_LS_DOWN[i])
```

```
        TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N}$ = TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N-1}$ +
        (TTIP_MES_BOL_LS_DOWN[i] - TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N-1}$) *
        MIN(1, C_CRLC_TTIP_FALL_R_IT_LS_DOWN * T_SAMPLE_R_IT_LS_DOWN[i])
    else
        TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N}$ = TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N-1}$ +
        (TTIP_MES_TOL_LS_DOWN[i] - TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N-1}$) *
        MIN(1, C_CRLC_TTIP_RISE_R_IT_LS_DOWN * T_SAMPLE_R_IT_LS_DOWN[i])

        TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N}$ = TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N-1}$ +
        (TTIP_MES_BOL_LS_DOWN[i] - TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N-1}$) *
        MIN(1, C_CRLC_TTIP_RISE_R_IT_LS_DOWN * T_SAMPLE_R_IT_LS_DOWN[i])
    endif
    If  ((TTIP_MES_LS_DOWN[i] $\le$ TTIP_MES_TOL_MMV_LS_DOWN[i]) &
         (TTIP_MES_LS_DOWN[i] $\ge$ TTIP_MES_BOL_MMV_LS_DOWN[i]))
    then
        R_IT_LS_DOWN[i] = R_IT_MES_LS_DOWN[i]$_{N}$
        Increment CTR_CYCNR_R_IT_LS_DOWN_VLD[i]
        Reset T_SAMPLE_R_IT_LS_DOWN[i]
        TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N-1}$ = TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N}$
        TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N-1}$ = TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N}$
    else
        R_IT_LS_DOWN[i]$_{N}$ = R_IT_LS_DOWN[i]$_{N-1}$
        TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N}$ = TTIP_MES_TOL_MMV_LS_DOWN[i]$_{N-1}$
        TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N}$ = TTIP_MES_BOL_MMV_LS_DOWN[i]$_{N-1}$
    endif
    R_IT_MES_LS_DOWN[i]$_{N-1}$ = R_IT_MES_LS_DOWN[i]$_{N}$
else
    R_IT_LS_DOWN[i] = R_IT_MES_LS_DOWN[i]$_{N}$
    Increment CTR_CYCNR_R_IT_LS_DOWN_VLD[i]
endif.
```

NOTES:

1. ***The difference (VLS_DOWN_CUR_PUMP_ON[i] - VLS_DOWN_CUR_PUMP_OFF[i]) shall be limited to 0 for negative values as the voltage during pumping should be greater than that without.***
2. ***If the incrementation would cause the counter CTR_CYCNR_R_IT_LS_DOWN_VLD[i] to overflow then the counter shall be reset to 0.***
3. ***The basic software (BSW) shall mask the VLS_DOWN[i] signal during resistance determination by retaining the value of the last sample prior to activating the resistance determination. This shall be indicated by setting LV_R_IT_REQ_LS_DOWN[i]. Upon revoking the masking the last oxygen sensor signal voltage VLS value determined by the BSW shall be used for the VLS_DOWN[i] signal and LV_R_IT_REQ_LS_DOWN[i] shall be reset.***
4. ***Should C_T_CUR_PUMP_ON_LS_DOWN = 0, SW1 shall not be activated.***

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEG_MIN_CUR_PUMP_ON_LS_DOWN	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Temperature threshold for the activation of sensor pumping					
C_T_CUR_PUMP_ON_LS_DOWN	1	0...8H	0...0.008	0.001	[s]
Duration for pump current active					
C_T_CUR_PUMP_OFF_LS_DOWN	1	0...FFFFH	0...655.35	0.01	[s]
Minimum delay between consecutive pump current activation					
C_T_DLY_CUR_PUMP_LS_DOWN	1	0...FFH	0...0.255	0.001	[s]
Duration in which VLS_DOWN[NC_CBK_EX_NR] signal masked directly after Ri determination					
C_VLS_HYS_R_IT_LS_DOWN	1	0...3FFH	0...4.99511	4.88E-03	[V]
Voltage hysteresis for evaluation if VLS_DOWN[i] shall be masked after Ri determination					
C_VLS_DOWN_MMV_MIN_RNG_TOL_R_IT	1	0...FFFFH	0...4.99992	7.63E-05	[V]
Permitted range of VLS_DOWN[NC_CBK_EX_NR] above VLS_DOWN_MMV_MIN[NC_CBK_EX_NR] at low gradient for Ri determination					
C_VLS_DOWN_MMV_MIN_RNG_BOL_R_IT	1	0...FFFFH	0...4.99992	7.63E-05	[V]
Permitted range of VLS_DOWN[NC_CBK_EX_NR] below VLS_DOWN_MMV_MIN[NC_CBK_EX_NR] at low gradient for Ri determination					
C_VLS_DOWN_MMV_DRV1_THD_MIN	1	8000...7FFFH	-2.5...2.4999	7.63E-05	[V/10ms]
Minimum threshold for mean 1st derivative of VLS_DOWN[NC_CBK_EX_NR] signal for Ri determination					
C_VLS_DOWN_MMV_DRV1_THD_MAX	1	8000...7FFFH	-2.5...2.4999	7.63E-05	[V/10ms]
Maximum threshold for mean 1st derivative of VLS_DOWN[NC_CBK_EX_NR] signal for Ri determination					
C_VLS_DOWN_MMV_DRV1_ABS_MAX	1	0...FFFFH	0...4.99992	7.63E-05	[V/10ms]
Threshold for absolute maximum 1st derivative of VLS_DOWN[NC_CBK_EX_NR] signal for Ri determination					
C_FAC_VLS_DOWN_MMV_DRV1_THD	1	0...FFH	0...1.99218	0.0078125	[-]
Multiplicative factor used to weight gradient thresholds dependent on VLS_DOWN_MMV_xxx[NC_CBK_EX_NR] amplitude					
C_FAC_VLS_DOWN_MMV_DRV1_THD_MIN	1	0...FFH	0...0.99609	3.91E-03	[-]
Hysteresis factor of DRV1_VLS_DOWN_MMV_MIN governing limiting of DRV1_VLS_DOWN_MMV_MIN_THD					
IP_FAC_T_DRV1_VLS_DOWN_MMV	8	0...FFH	0...63.75	0.25	[-]
LDPM_T_AFL_CYC_HLD_1_EGCP	8	0...FFH	0...2.55	0.01	[s]
Multiplicative factor modifies DRV1_VLS_DOWN_MMV_MIN_THD dependent on T_VLS_CYC_AFL_HLD					
IP_CRLC_TTIP_REF_MDL_LS_DOWN	8	0...FFH	0...0.99609	3.91E-03	[-]
LDPM_MAF_KGH_1_EGCP	8	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correlation map for calculation of reference downstream tip temperature (modelled for new sensor)					
IP_R_IT_MDL_LS_DOWN_NEW	8*8	0...FFFFH	0...65535	1	[ohm]
LDP_POW_MMV_LSH_DOWN_IP_LS_DOWN	8	0...FFFFH	0...84.49871	1.2894e-3	[W]
LDP_TTIP_REF_MDL_MMV_LS_DOWN_R	8	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Internal resistance of a new downstream oxygen sensor (modelled)					
IP_FAC_T_VLS_DOWN_MMV_DRV1_ABS	8	0...FFH	0...15.9375	0.0625	[-]
LDPM_T_AFL_CYC_HLD_1_EGCP	8	0...FFH	0...2.55	0.01	[s]
Multiplicative factor modifies C_VLS_DOWN_MMV_DRV1_ABS_MAX dependent on T_VLS_CYC_AFL_HLD					
C_CTR_AFL_CYC_MIN_RI_IT_LS_DOWN	1	0...FFFFH	0...65535	1	[-]
Min. no. AF cycles post lambda controller activation to permit use of T_VLS_CYC_AFL_HLD dependency					
C_MAF_KGH_MAX_R_IT_LS_DOWN	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Maximum MAF_KGH threshold for activation of internal resistance determination					
C_FAC_R_IT_ERR_POS_LS_DOWN	1	0...FFH	0...0.99609	3.91E-03	[-]
Internal resistance measurement positive error (i.e. 0% to 100%)					
C_FAC_R_IT_ERR_NEG_LS_DOWN	1	0...FFH	0...0.99609	3.91E-03	[-]
Internal resistance measurement negative error (i.e. 100% to 0%)					
IP_TTIP_MES_LS_DOWN	8	0...7FFFH	0...2047.9375	0.0625	[°C]
LDP_R_IT_MES_LS_DOWN_IP_LS_DOWN	8	0...FFFFH	0...65535	1	[ohm]
Oxygen sensor temperature vs. internal resistance characteristic					

C_CRLC_POW_MMV_LSH_DOWN	1	0...FFH	0...0.99609	3.91E-03	[-]
Correlation (filter) constant for moving mean value of downstream heater power					
C_CRLC_TTIP_FALL_R_IT_LS_DOWN	1	0...FFH	0...0.99609	3.91E-03	[1/s]
Correlation (filter) constant for falling ceramic temperature, limit calculation					
C_CRLC_TTIP_RISE_R_IT_LS_DOWN	1	0...FFH	0...0.99609	3.91E-03	[1/s]
Correlation (filter) constant for rising ceramic temperature, limit calculation					
LC_R_IT_VLD_INH_LS_DOWN	1	0...1H	0...1	1	[-]
Boolean flag inhibiting internal resistance determination validity function					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_R_REF_LS_DOWN	1	0...FFFFH	0...65535	1	[ohm]
Reference resistance for pump current (RPump)					
NC_FAC_R_REF_LS_DOWN	1	0...FFFFH	0...3.99993	6.10E-05	[-]
Pump current limiting to pull down resistance ratio (RPump/R2)					
NC_VLS_DOWN_CUR_PUMP_REF	1	0...3FFH	0...4.99511	4.88E-03	[V]
Reference voltage value (4.82 V)					

16.12.1 Application specific conditions for oxygen sensor internal resistance

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_R_IT_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	-
Boolean flag indicating that downstream sensor internal resistance determination shall be inhibited when set					

Input data:

NC_CBK_EX_NR	LV_ES	LV_VAR_LSH_DOWN	TCO

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.

The Boolean flag(s) LV_INH_R_IT_LS_DOWN[NC_CBK_EX_NR] shall indicate whether the downstream oxygen sensor internal resistance determination shall be permitted to take place. When set, no resistance determination shall take place.

Application conditions:

Initialisation: The following initialisation shall be carried out upon leaving the engine state Engine Stop (LV_ES).

LV_INH_R_IT_LS_DOWN[NC_CBK_EX_NR] = 0

Recurrence: 1s

Activation: at every engine operating state

Deactivation: -

Formula section:

```
If   LV_VAR_LSH_DOWN = 1   and   TCO > C_TCO_CUR_ON_LS_DOWN_MIN
  then    LV_INH_R_IT_LS_DOWN[NC_CBK_EX_NR]  =  0
  else    LV_INH_R_IT_LS_DOWN[NC_CBK_EX_NR]  =  1
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_CUR_ON_LS_DOWN_MIN	1	0 ... FEH	-48 ... 142.5	0.75	°C
TCO threshold for detection of R_IT_LS_DOWN					

16.13 Signal acquisition and compensation of WRAF sensor

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_UP_COR[NC_CBK_EX_NR]	V/O	0...7FFH	0...9.99511	4.8828e-3	[V]
Compensated output signal of lambda sensor					
VLS_UP_COR_H_GAIN[NC_CBK_EX_NR]	V/O	0...7FFH	0...9.99511	4.8828e-3	[V]
Ip gain16 equivalent of compensated output signal of lambda sensor					
VLS_COR_H_RES_LSL[NC_CBK_EX_NR]	V	0...FFFFH	0...9.99984	0.1526e-3	[V]
Compensated output signal of lambda sensor with extended resolution					
VLS_OFS_LSL[NC_CBK_EX_NR]	V	8000...7FFFH	-5...4.99984	1.53E-04	[V]
Output signal offset of linear lambda sensor for Ip gain16					
VLS_OFS_LSL_L_GAIN[NC_CBK_EX_NR]	V	8000...7FFFH	-5...4.99984	1.53E-04	[V]
Output signal offset of linear lambda sensor for Ip gain8					
VLS_OFS_TMP_LSL[NC_CBK_EX_NR]	V	8000...7FFFH	-5...4.99984	1.53E-04	[V]
Intermediate storage variable for calculated offset value for Ip gain16					
VLS_OFS_TMP_OLD_LSL[NC_CBK_EX_NR]	V	8000...7FFFH	-5...4.99984	1.53E-04	[V]
Storage variable for calculated offset value for Ip gain16					
VLS_OFS_TMP_OLD_LSL_L_GAIN[NC_CBK_EX_NR]	V	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Storage variable for calculated offset value for Ip gain8					
VLS_UP_DIAG[NC_CBK_EX_NR]	V/O	8000...7FFFH	-160...159.99511	4.88E-03	[V]
Offset and exhaust pressure compensated VLS_UP[i] which is used for sensor diagnoses					
VLS_MV_LSL[NC_CBK_EX_NR]	V	0...7FFFH	0...4.99984	1.53E-04	[V]
Mean value of output signal of linear lambda sensor					
VLS_MV_LSL_BUF[NC_CBK_EX_NR]	V	0...FFFFH	0...319.99511	4.88E-03	[V]
Buffer for mean value calculation					
CTR_VLS_MV_LSL_NR_SAMPLE[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Sample counter for mean value calculation					
VLS_MMV_LSL[NC_CBK_EX_NR]	V	0...7FFFH	0...4.99984	1.53E-04	[V]
Moving mean value of output signal of linear lambda sensor for Ip gain16					
VLS_MMV_LSL_L_GAIN[NC_CBK_EX_NR]	V	0...7FFFH	0...4.99984	1.53E-04	[V]
Moving mean value of output signal of linear lambda sensor for Ip gain8					
VLS_DELTA_TMP_LSL[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-327680...327679.99984	1.53E-04	[V]
Controller setpoint free, gain and exhaust pressure corrected output signal					
LV_LSL_OFS_ADJ_ACT[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
State of the offset measurement (active / not active)					
LV_LSL_OFS_ACT_AST[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Switch to chose activation conditions directly after engine start					
LV_LAMB_STI_OFF_LSL_OFS_ADJ[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Request for disabling of forced lambda stimulation, or keeping it disabled, resp.					
LV_LAMB_PLS_SWI_OFF[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Indicator for need to switch off forced lambda stimulation					
LV_LSL_OFS_ADJ_IS[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Offset measurement carried out during current idle phase					
IPLSL_COR[NC_CBK_EX_NR]	V/O	8000...7FFFH	-12.596...12.59561	3.84E-04	[mA]
Compensated nominal pumping current (after sensor / sensor dispersion compensation for new / aged sensor)					
STATE_LSL_OFS_ADJ[NC_CBK_EX_NR]	V	0H 1H 2H 3H 4H	WAIT PREPARE ADJUST_H ADJUST_L RETURN	1	[-]

State of WRAF controller lambda output voltage offset adjustment					
LV_VLS_OFS_LIM_LSL[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating limitation of output voltage offset adjustment value for Ip gain16					
LV_VLS_OFS_LIM_LSL_L_GAIN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating limitation of output voltage offset adjustment value for Ip gain8					
LV_VLS_OFS_ADJ_ENA_LSL_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag requesting enabling of offset adjustment mode of WRAF sensor interface for lambda signal output voltage					
LV_SWI_GAIN_LSL_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag configuring the WRAF sensor interface gain-selection for Ip measurement (‘0’ = g8, ‘1’ = g16)					
CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Storage variable for SPI communication error counter value of WRAF controller					
T_DLY_LSL_OFS_ADJ[NC_CBK_EX_NR]	V	0...FFH	0...2.55	0.01	[s]
Wait timer to delay acquisition of measurement values upon activation of adjustment					
T_WAIT_LSL_OFS_ADJ[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Wait timer to define minimum interval between two subsequent offset adjustment cycles					
CTR_CYC_T_WAIT_OFS_ADJ_RED[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter of cycles for application of reduced repetition rate for offset adjustment during engine warm-up					
T_DLY_LAMB_STI_OFF_OFS_ADJ[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Wait timer to allow for lambda signal settling upon deactivation of forced stimulation					
T_DLY_LSL_OFS_ADJ_IS[NC_CBK_EX_NR]	V	0...FFH	0...2.55	0.01	[s]
Wait timer to delay activation of offset adjustment upon entering engine state idle speed					
LV_VLS_OFS_ADJ_CMPL[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating successful completion of offset adjustment cycle					
LV_LSL_OFS_GAIN_SWI_CYC[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating active Ip gain range for alternating offset adjustment					
T_DLY_LSL_OFS_ADJ_PUC_END[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Wait timer to delay activation of offset adjustment after leaving fuel cut-off phase					
LAMB_LS_UP[NC_CBK_EX_NR]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda signal value of the WRAF sensor					
LAMB_LS_UP_MV	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Mean value of lambda in multiple-branched exhaust gas lines					
LAMB_LS_UP_MV_TMP	-	0...FFFFFFFFH	0... 4194303.99902	0.9766e-3	[-]
Temporary storage for calculation of mean value of lambda in multiple-branched exhaust gas lines					
LAMB_LS_UP_MIN	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Minimum lambda in multiple-branched exhaust gas lines					
VLS_COR_TMP_LSL[NC_CBK_EX_NR]	V	0...FFFFH	0...9.99984	0.1526e-3	[V]
Temporary storage value of compensated output signal of lambda sensor					
VLS_COR_MV_LSL[NC_CBK_EX_NR]	V	0...FFFFH	0...9.99984	0.1526e-3	[V]
Optional average over two samples of compensated output signal of lambda sensor					
LV_SWI_CLC_LSL_OFS_ADJ[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical variable for switching of lambda calculation routines					
T_INI_DLY_LSL_SIG_ACQ[NC_CBK_EX_NR]	V	0...FFH	0...2.55	0.01	[s]
Delay of aquisition after freeze of signal. Assures that segment synchronous meanvalue calc has reached final value.					
LV_LSL_UP_SHO_PER_REQ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Trigger the short period stop mode of Lambda controller during offset adjustment and gain switching					

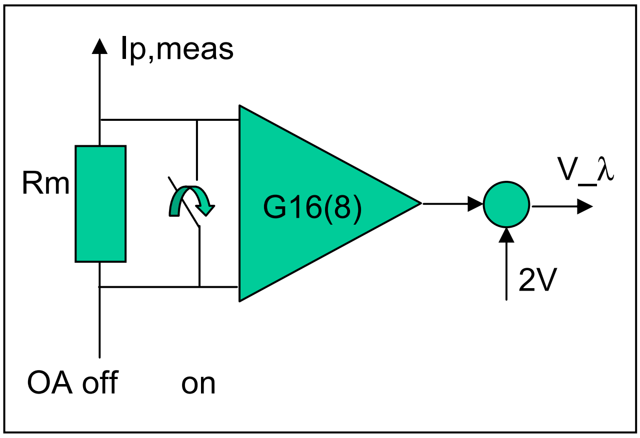
Input data:

VLS_UP[NC_CBK_EX_NR]	LV_LS_UP_READY[NC_CBK_EX_NR]	LV_ST_END	LV_IS
LV_PU	PRS_EX_PCAT_UP	LV_LAMB_PLS_ACT[NC_CBK_EX_NR]	FAC_LSL_GAIN_AD[NC_CBK_EX_NR]
VLS_UP_10_RAW[NC_CBK_EX_NR]	STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]	CTR_ERR_LSL_IF_SPI_WR[NC_CBK_EX_NR]	LV_LSL_IF_SPI_RST_END[NC_CBK_EX_NR]
NC_CBK_EX_NR	LAMB_SP[NC_CBK_EX_NR]	LV_IPLSL_VLD[NC_CBK_EX_NR]	STATE_CAT_DIAG[NC_CBK_EX_NR]
LV_VLS_UP_VLD[NC_CBK_EX_NR]	LV_VLS_UP_INIT[NC_CBK_EX_NR]	LV_PUC	LV_INH_LSL_OFS_ADJ[NC_CBK_EX_NR]
TTIP_MES_LS_UP[NC_CBK_EX_NR]	LV_TTIP_MES_VLD_LS_UP[NC_CBK_EX_NR]	LV_LS_UP_READY_CDN[NC_CBK_EX_NR]	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]
VB	LV_LSH_UP_MAN_ACT[NC_CBK_EX_NR]	STATE_LSH_UP[NC_CBK_EX_NR]	

FUNCTION DESCRIPTION:

OA = Offset adjustment.

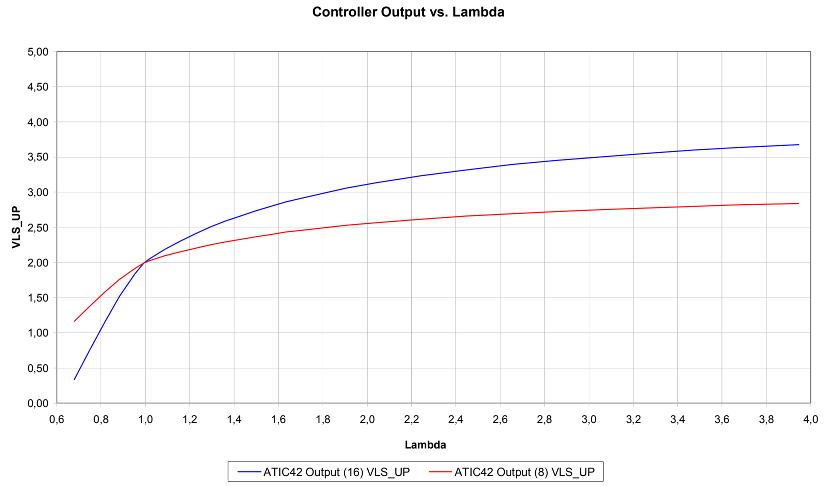
Based on short circuiting of Ip measurement amplifier input.



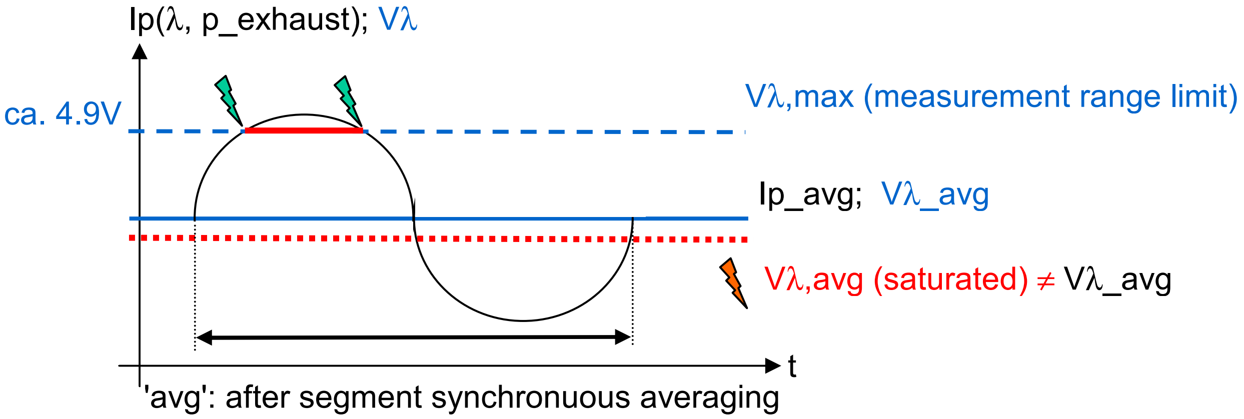
Ip measurement gain16 = High Gain ('1')

Ip measurement gain8 = Low Gain ('0')

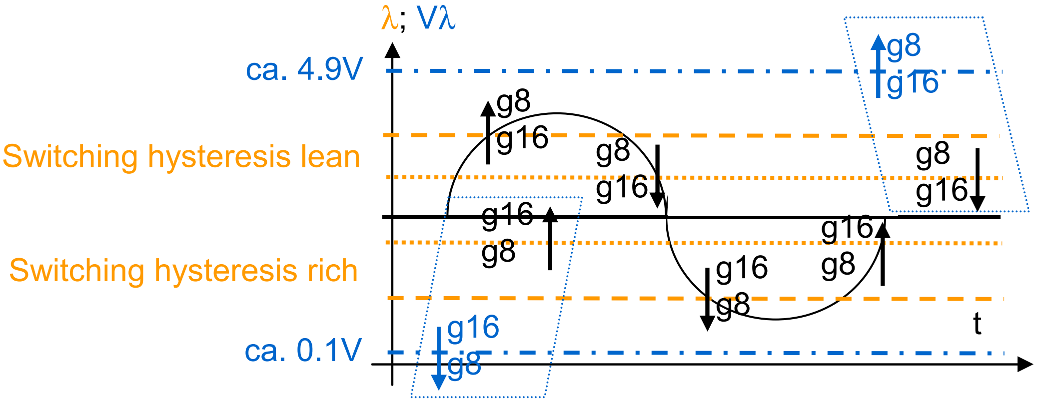
	Use cases			
LOGICAL CONSTANT **Effect ('0' / '1')**	**gain16 only**	**gain8 only**	**gain16&8 simultaneously**	**gain16/8 alternating**
LC_LSL_OFS_ADJ_GAIN_SEL **Defines default Ip measurment gain**	**1**	**0**	**1**	**1**
LC_LSL_OFS_ADJ_H_L_GAIN **OA in default gain only (second range is approximated) / for both gains**	**0**	**0**	**1**	**1**
LC_LSL_OFS_GAIN_SWI_CYC **Simultaneous / alternating OA for both gains**	**0**	**0**	**0**	**1**
LC_IPLSL_GAIN_SWI_ACT **Gain switching during Ip measurement active**	**0 / 1**	**0 / 1**	**0 / 1**	**0 / 1**



Ip measurement range vs. resolution (exemplary).



Range switching triggered by λ / Vλ thresholds to avoid saturation effects of Ip measurement.



Switching thresholds/hysteresis

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned
(NC_CBK_EX_NR = 2), then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation: The following variable initialisation shall take place at **LV_IGK 0->1**, **after RESET,** or **upon erase of error management**:

VLS_UP_COR[i] = 0

VLS_COR_H_RES_LSL[i] = 0

VLS_UP_COR_H_GAIN[i] = 0

VLS_DELTA_TMP_LSL[i] = 0

VLS_OFS_LSL[i] = 0

VLS_OFS_TMP_LSL[i] = 0

VLS_OFS_TMP_OLD_LSL[i] = 0

VLS_OFS_LSL_L_GAIN[i] = 0

VLS_OFS_TMP_OLD_LSL_L_GAIN[i] = 0

IPLSL_COR[i] = 0

LV_VLS_OFS_LIM_LSL[i] = 0

LV_VLS_OFS_LIM_LSL_L_GAIN[i] = 0

VLS_MV_LSL[i] = 0

LAMB_LS_UP[i] = 1

LAMB_LS_UP_MV = 1

LAMB_LS_UP_MIN = 1

VLS_COR_TMP_LSL[i] = C_VLS_OFS_SP_LSL

VLS_COR_MV_LSL[i] = C_VLS_OFS_SP_LSL

LV_SWI_CLC_LSL_OFS_ADJ[i] = 0

VLS_MMV_LSL[i] $_{i-1}$ = C_VLS_OFS_SP_LSL

VLS_MMV_LSL_L_GAIN[i] $_{i-1}$ = C_VLS_OFS_SP_LSL

CTR_ CYC_T_WAIT_OFS_ADJ_RED[i] = 0

LV_LSL_UP_SHO_PER_REQ[i] = 0

LV_LSL_OFS_ACT_AST[i] = 1

```
LV_LSL_OFS_ADJ_ACT[i] = 0
LV_LSL_OFS_ADJ_IS[i] = 0
LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] = 0
LV_LAMB_PLS_SWI_OFF[i] = 0
T_WAIT_LSL_OFS_ADJ[i] = 0
CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[i]
    = CTR_ERR_LSL_IF_SPI_WR[i]
STATE_LSL_OFS_ADJ[i] = " WAIT "
T_DLY_LSL_OFS_ADJ_IS[i] = C_T_DLY_LSL_OFS_ADJ_IS
LV_SWI_GAIN_LSL_IF[i] = LC_LSL_OFS_ADJ_GAIN_SEL
LV_VLS_OFS_ADJ_CMPL[i] = 0
LV_LSL_OFS_GAIN_SWI_CYC[i] = LC_LSL_OFS_ADJ_GAIN_SEL
T_DLY_LSL_OFS_ADJ_PUC_END[i] = 0
T_INI_DLY_SIG_ACQ[i] = 0
```

The following variable initialisation shall take place according to below condition:

```
If  ( LV_LSH_UP_MAN_ACT[i] == 0 ->1 and LV_ST_END == 0 )
or  ( LV_LSH_UP_MAN_ACT[i] == 0 and LV_ST_END == 0 ->1 )
then
    LV_LSL_OFS_ACT_AST[i] = 1
    LV_LSL_OFS_ADJ_ACT[i] = 0
    LV_LSL_OFS_ADJ_IS[i] = 0
    LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] = 0
    LV_LAMB_PLS_SWI_OFF[i] = 0
    T_WAIT_LSL_OFS_ADJ[i] = 0
    CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[i]
            = CTR_ERR_LSL_IF_SPI_WR[i]
    STATE_LSL_OFS_ADJ[i] = " WAIT "
    T_DLY_LSL_OFS_ADJ_IS[i] = C_T_DLY_LSL_OFS_ADJ_IS
    LV_SWI_GAIN_LSL_IF[i] = LC_LSL_OFS_ADJ_GAIN_SEL
    LV_VLS_OFS_ADJ_CMPL[i] = 0
    LV_LSL_OFS_GAIN_SWI_CYC[i] = LC_LSL_OFS_ADJ_GAIN_SEL
    T_DLY_LSL_OFS_ADJ_PUC_END[i] = 0

endif
```

Recurrence: T_SAMPLE = 10ms

16.13.1 Compensation of WRAF controller electrical offset voltage and system gain error

Activation:

To avoid disturbances to the WRAF interface due to EMC interference of the starter, the function shall be activated after engine start.

```
If      ( LV_ST_END == 1
or      LV_LSH_UP_MAN_ACT[i] == 1 )
and     LV_LSL_IF_SPI_RST_END[i] == 1
        ( Initialisation of WRAF controller ASIC completed successfully )
and     LV_ERR_OFS_LSL_UP[i] == 0
then    " function enabled "
else    " function disabled "
        The following variable initialisation shall take place after function deactivation:
        LV_VLS_OFS_ADJ_ENA_LSL_IF[i] = 0
        LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] = 0
        LV_LAMB_PLS_SWI_OFF[i] = 0
        LV_LSL_OFS_ADJ_ACT[i] = 0
        LV_LSL_OFS_ADJ_IS[i] = 0
endif
```

```
If      LV_LSL_OFS_ACT_AST[i] == 1
then    " 1.1.1 Conditions for offset adjustment directly after engine start "
else    " 1.1.2 Conditions for offset adjustment during normal engine operation "
endif
```

Deactivation: --

16.13.1.1 Conditions for offset adjustment directly after engine start

Description:

Formula section:

```
If      ( LV_LS_UP_READY[i] == 0
or      LV_LSH_UP_MAN_ACT[i] == 1 )
and     LV_INH_LSL_OFS_ADJ[i] == 0
and     VB > C_VB_MIN_OFS_ADJ
and     STATE_LSL_IF_SPI_RD[i] ( bit LSL_ENA ) == 1
and     CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[i] == CTR_ERR_LSL_IF_SPI_WR[i]
then    LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] = 1
        If      LC_LSL_OFS_ADJ_H_L_GAIN == 1
        and     LV_LSL_OFS_ADJ_ACT[i] == 0
        then    LV_SWI_GAIN_LSL_IF[i] = 1
        endif
else    CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[i] = CTR_ERR_LSL_IF_SPI_WR[i]
        LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] = 0
        LV_SWI_GAIN_LSL_IF[i] = LC_LSL_OFS_ADJ_GAIN_SEL
        If      LV_VLS_OFS_ADJ_ENA_LSL_IF[i] == 1 ( first measurement aborted )
        or      LV_LS_UP_READY[i] == 1 ( regular engine operation reached )
        then    LV_LSL_OFS_ACT_AST[i] = 0  ( leave after start phase )
                STATE_LSL_OFS_ADJ[i] = “ WAIT ”
        endif
        LV_VLS_OFS_ADJ_ENA_LSL_IF[i] = 0
endif
```

16.13.1.2 Conditions for offset adjustment during normal engine operation

Description:

Formula section:

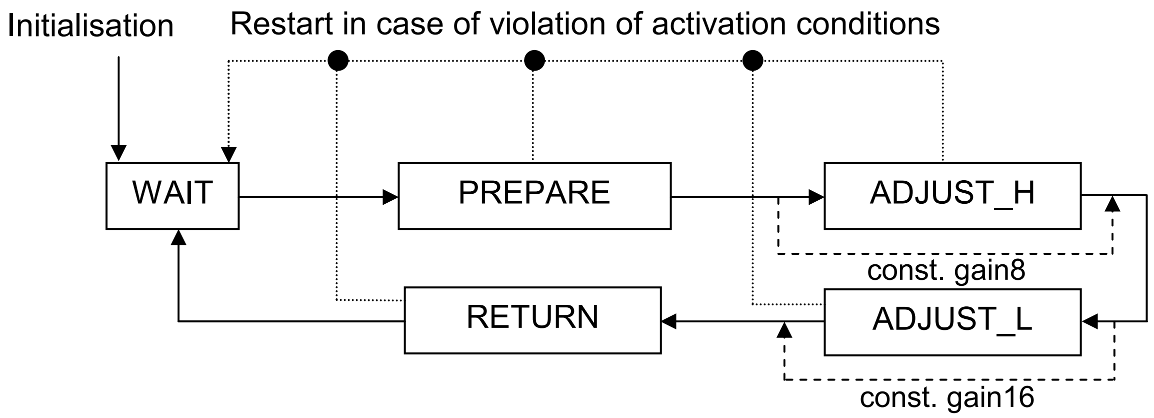
```
T_WAIT_LSL_OFS_ADJ[i] = T_WAIT_LSL_OFS_ADJ[i] – T_SAMPLE

If      LV_IS == 0
then    T_DLY_LSL_OFS_ADJ_IS[i] = C_T_DLY_LSL_OFS_ADJ_IS
else    T_DLY_LSL_OFS_ADJ_IS[i] = T_DLY_LSL_OFS_ADJ_IS[i] – T_SAMPLE
endif
```

```
If      LV_PUC == 1
then    T_DLY_LSL_OFS_ADJ_PUC_END[i] = C_T_DLY_LSL_OFS_ADJ_PUC_END
else    T_DLY_LSL_OFS_ADJ_PUC_END[i]
                = T_DLY_LSL_OFS_ADJ_PUC_END[i] – T_SAMPLE
endif

If      T_WAIT_LSL_OFS_ADJ[i] == 0
and     LV_INH_LSL_OFS_ADJ[i] == 0
and     LV_LS_UP_READY[i] == 1
and     LV_LSH_UP_MAN_ACT[i] == 0
and     VB > C_VB_MIN_OFS_ADJ
and     LV_PUC == 0
and     T_DLY_LSL_OFS_ADJ_PUC_END[i] == 0
and     LV_PU == 0
and     ( LC_LSL_OFS_ADJ_ENA == 1
        or      ( LV_IS == 1
        and T_DLY_LSL_OFS_ADJ_IS[i] == 0 ) )
and     LV_LSL_OFS_ADJ_IS[i] == 0
and     ( C_LAMB_LS_UP_MIN_OFS_ADJ < LAMB_SP[i]
            < C_LAMB_LS_UP_MAX_OFS_ADJ
or      LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] == 1 )
and     ( LC_CAT_DIAG_ACT_WAIT == 0
        or      STATE_CAT_DIAG[i] == “PASSIVE”
        or      STATE_CAT_DIAG[i] == “END” )
and     STATE_LSL_IF_SPI_RD[i] ( bit LSL_ENA ) == 1
and     CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[i] == CTR_ERR_LSL_IF_SPI_WR[i]
then    If      ( LC_LSL_OFS_ADJ_H_L_GAIN == 1
                and     LV_SWI_GAIN_LSL_IF[i] == 1
                and     STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN ) == 1 )
                ( OA for both ranges active: starts once gain16  is selected in
                measurement mode, is performed for gain16 and gain8
                simultaneously / alternating, return to gain16 -> measurement mode )
        or      ( LC_LSL_OFS_ADJ_H_L_GAIN == 0
                and     LV_SWI_GAIN_LSL_IF[i] == LC_LSL_OFS_ADJ_GAIN_SEL
                and     STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN )
                                == LC_LSL_OFS_ADJ_GAIN_SEL )
```

```
                ( OA is performed for default range only )
        then    If      LV_LAMB_PLS_ACT[i] == 1
                and     LC_LAMB_PLS_ACT_NEUT == 0
                then    LV_LAMB_PLS_SWI_OFF[i] = 1
                endif
                LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] = 1
        endif
else    CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[i] = CTR_ERR_LSL_IF_SPI_WR[i]
        LV_LAMB_PLS_SWI_OFF[i] = 0
        LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] = 0
        If      LV_IS == 0
        then    LV_LSL_OFS_ADJ_IS[i] = 0
        endif
        STATE_LSL_OFS_ADJ[i] = " WAIT "
        LV_VLS_OFS_ADJ_ENA_LSL_IF[i] = 0
endif
```

16.13.1.3 Offset calculation

Description:



Formula section:

Initial state upon (re)start of state machine

STATE “ WAIT ”:

```
LV_VLS_OFS_ADJ_CMPL[i] = 0
If    STATE_LSL_IF_SPI_RD[i] ( bit VLS_OFS_ENA ) == 0
```

```
then        LV_LSL_OFS_ADJ_ACT[i] = 0
            If      LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] == 1
            and     ( LV_LAMB_PLS_ACT[i] == 0
                    or      LC_LAMB_PLS_ACT_NEUT == 1 )
            then    If      LV_LAMB_PLS_SWI_OFF[i] == 1
                    and     LC_LAMB_PLS_ACT_NEUT == 0
                    then    T_DLY_LAMB_STI_OFF_OFS_ADJ[i]
                                    = C_T_DLY_LAMB_STI_OFF_OFS_ADJ
                    else    T_DLY_LAMB_STI_OFF_OFS_ADJ[i] = 0
                    endif
                    STATE_LSL_OFS_ADJ[i]  = " PREPARE "
            endif
endif
```

STATE " PREPARE ":

```
If          T_DLY_LAMB_STI_OFF_OFS_ADJ[i] == 0
then        LV_LSL_OFS_ADJ_ACT[i] = 1
            LV_VLS_OFS_ADJ_ENA_LSL_IF[i] = 1
            If      LC_LSL_OFS_GAIN_SWI_CYC[i] == 1
            and     LV_LSL_OFS_GAIN_SWI_CYC[i] == 0
            ( alternating OA, gain8 recurrence )
            then    LV_SWI_GAIN_LSL_IF[i] = 0
            endif
            If      LC_CTR_ERR_LSL_IF_SAVE_MAN_INC == 1
            then    CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[i]
                            = CTR_ERR_LSL_IF_SPI_OFS_ADJ_SAVE[i] + 1
            endif
            If      STATE_LSL_IF_SPI_RD[i] ( bit VLS_OFS_ENA ) == 1
            ( ATIC42 OA mode enabled )
            then    T_DLY_LSL_OFS_ADJ[i] = C_T_DLY_LSL_OFS_ADJ
                    CTR_VLS_MV_LSL_NR_SAMPLE[i] = 0
                    VLS_MV_LSL_BUF[i] = 0
                    If      ( LC_LSL_OFS_ADJ_H_L_GAIN == 0
                    and     LC_LSL_OFS_ADJ_GAIN_SEL == 0 )
                    ( OA for gain8 only )
                    or      LV_LSL_OFS_GAIN_SWI_CYC[i] == 0
```

```
                    ( Alternating OA gain16/8 )
                    then    STATE_LSL_OFS_ADJ[i]  = " ADJUST_L "
                    else    STATE_LSL_OFS_ADJ[i]  = " ADJUST_H "
                    endif
            endif
else        T_DLY_LAMB_STI_OFF_OFS_ADJ[i]
                = T_DLY_LAMB_STI_OFF_OFS_ADJ[i] - T_SAMPLE
endif

STATE " ADJUST_H ":
If          STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN ) == 1  ( gain16 )
then        If      T_DLY_LSL_OFS_ADJ[i] == 0
            then    " 1.1.3.1 Measurement acquisition "
                    If      CTR_VLS_MV_LSL_NR_SAMPLE[i] >= 4
                    then    VLS_MV_LSL[i] = VLS_MV_LSL_BUF[i] / 4
                            " 1.1.3.2 Offset adjustment High Gain "
                            If      LC_LSL_OFS_ADJ_H_L_GAIN == 1
                            ( OA for both ranges active )
                            then    If      LC_LSL_OFS_GAIN_SWI_CYC == 0
                                    ( Simultaneous OA gain16&8 )
                                    or      LV_LSL_OFS_ACT_AST[i] == 1
                                    ( Initial OA in both gain16&8 )
                                    then    T_DLY_LSL_OFS_ADJ[i]
                                                    = C_T_DLY_LSL_OFS_ADJ_CYC
                                            CTR_VLS_MV_LSL_NR_SAMPLE[i] = 0
                                            VLS_MV_LSL_BUF[i] = 0
                                            LV_SWI_GAIN_LSL_IF[i] = 0
                                            STATE_LSL_OFS_ADJ[i] = " ADJUST_L "
                                    else    ( Alternating OA gain16/8 )
                                            LV_LSL_OFS_GAIN_SWI_CYC = 0
                                            STATE_LSL_OFS_ADJ[i] = " RETURN "
                                    endif
                            else    ( OA for gain16 only )
                                    VLS_OFS_TMP_LSL[i]
                                    = ( ( VLS_OFS_TMP_LSL[i] – C_VLS_OFS_DELTA_ADD )
                                            / 2 ) + C_VLS_OFS_DELTA_ADD
```

```
                              " 1.1.3.3. Offset adjustment Low Gain "
                              ( calculation of corresponding offset for gain8 )
                              STATE_LSL_OFS_ADJ[i] = " RETURN "
                    endif
               endif
          else T_DLY_LSL_OFS_ADJ[i] = T_DLY_LSL_OFS_ADJ[i] – T_SAMPLE
          endif
endif

STATE " ADJUST_L ":
If        STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN ) == 0  ( gain8 )
then      If   T_DLY_LSL_OFS_ADJ[i] == 0
          then " 1.1.3.1 Measurement acquisition "
               If   CTR_VLS_MV_LSL_NR_SAMPLE[i] >= 4
               then VLS_MV_LSL[i] = VLS_MV_LSL_BUF[i] / 4
                    " 1.1.3.3 Offset adjustment Low Gain "
                    If   LC_LSL_OFS_ADJ_H_L_GAIN == 1
                         ( OA for both ranges active )
                    then LV_SWI_GAIN_LSL_IF[i] = 1
                    else ( OA for gain8 only )
                         VLS_OFS_TMP_LSL[i] = ( ( VLS_OFS_TMP_LSL[i]
                                   – C_VLS_OFS_DELTA_ADD ) * 2 )
                                   + C_VLS_OFS_DELTA_ADD
                         " 1.1.3.2 Offset adjustment High Gain "
                         ( calculation of corresponding offset for gain16 )
                    endif
                    If   LC_LSL_OFS_GAIN_SWI_CYC == 1
                    then LV_LSL_OFS_GAIN_SWI_CYC[i] = 1
                    endif
                    STATE_LSL_OFS_ADJ[i] = " RETURN "
               endif
          else T_DLY_LSL_OFS_ADJ[i] = T_DLY_LSL_OFS_ADJ[i] – T_SAMPLE
          endif
endif

STATE " RETURN ":
```

```
LV_VLS_OFS_ADJ_ENA_LSL_IF[i] = 0
If      LC_LSL_OFS_ADJ_H_L_GAIN == 0
( OA in default range only: no gain switching has taken place during calibration)
or      STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN ) == 1
( OA in both ranges: wait for return to gain16 prior to exiting to measurement mode )
then    If      LV_LSL_OFS_ACT_AST[i] == 0
        then    If      CTR_ CYC_T_WAIT_OFS_ADJ_RED[i]
                                < C_NR_CYC_T_WAIT_LSL_OFS_ADJ_RED
                then    T_WAIT_LSL_OFS_ADJ[i]
                                = C_T_WAIT_LSL_OFS_ADJ_RED
                        CTR_ CYC_T_WAIT_OFS_ADJ_RED[i]
                                = CTR_ CYC_T_WAIT_OFS_ADJ_RED[i] + 1
                else    T_WAIT_LSL_OFS_ADJ[i] = C_T_WAIT_LSL_OFS_ADJ
                endif
                If      LV_IS == 1
                then    LV_LSL_OFS_ADJ_IS[i] = 1
                endif
        else    LV_LSL_OFS_ACT_AST[i] = 0
        endif
        STATE_LSL_OFS_ADJ[i]  = " WAIT "
        LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] = 0
        LV_VLS_OFS_ADJ_CMPL[i] = 1
endif
```

16.13.1.3.1 Measurement acquisition

Description:

Formula section:

```
If      CTR_VLS_MV_LSL_NR_SAMPLE[i] < 4
then    VLS_MV_LSL_BUF[i] = VLS_MV_LSL_BUF[i] + VLS_UP_10_RAW[i]
        CTR_VLS_MV_LSL_NR_SAMPLE[i] =
                CTR_VLS_MV_LSL_NR_SAMPLE[i] + 1
endif
```

16.13.1.3.2 Offset adjustment High Gain

Description:

Formula section:

```
If      LC_LSL_OFS_ADJ_H_L_GAIN == 1
( skip gain16 measurement acquisition for 'OA in gain8 only'
  -> calculation of corresponding offset for gain16 )
or      LC_LSL_OFS_ADJ_GAIN_SEL == 1
then    If      LV_LSL_OFS_ACT_AST[i] == 1
        then    VLS_MMV_LSL[i] = VLS_MV_LSL[i]
        else    VLS_MMV_LSL[i]$_{i}$ = VLS_MMV_LSL[i]$_{i-1}$ + C_CRLC_VLS_UP_OFS_ADJ
                                         * ( VLS_MV_LSL[i] – VLS_MMV_LSL[i]$_{i-1}$ )
        endif
        VLS_OFS_TMP_LSL[i] = C_VLS_OFS_SP_LSL - VLS_MMV_LSL[i]
endif
If      I VLS_OFS_TMP_LSL[i] - VLS_OFS_TMP_OLD_LSL[i] I
                < C_VLS_OFS_DIF_OK_LSL
or      LV_LSL_OFS_ACT_AST[i] == 1
then    If      | VLS_OFS_TMP_LSL[i] | < C_VLS_OFS_MAX_ABSV_LSL
        then    LV_VLS_OFS_LIM_LSL[i] = 0
                VLS_OFS_LSL[i] = VLS_OFS_TMP_LSL[i]
        ( offset adjustment valid; VLS_OFS_LSL[i] is changed )
        else    LV_VLS_OFS_LIM_LSL[i] = 1
                If      VLS_OFS_TMP_LSL[i] > 0
                then    VLS_OFS_LSL[i] = + C_VLS_OFS_MAX_ABSV_LSL
                else    VLS_OFS_LSL[i] = – C_VLS_OFS_MAX_ABSV_LSL
        ( set offset voltage to threshold value which has been reached,
          taking into account the correct sign )
                endif
        endif
endif
VLS_OFS_TMP_OLD_LSL[i] = VLS_OFS_TMP_LSL[i]
```

16.13.1.3.3 Offset adjustment Low Gain

Description:

Formula section:

```
If          LC_LSL_OFS_ADJ_H_L_GAIN == 1
( skip gain8 measurement acquisition for 'OA in gain16 only'
   -> calculation of corresponding offset for gain8 )
or          LC_LSL_OFS_ADJ_GAIN_SEL == 0
then        If      LV_LSL_OFS_ACT_AST[i] == 1
            then    VLS_MMV_LSL_L_GAIN[i] = VLS_MV_LSL[i]
            else    VLS_MMV_LSL_L_GAIN[i]$_{i}$ = VLS_MMV_LSL_L_GAIN[i]$_{i-1}$
                            + C_CRLC_VLS_UP_OFS_ADJ_L_GAIN
                            * ( VLS_MV_LSL[i] – VLS_MMV_LSL_L_GAIN[i]$_{i-1}$ )
            endif
            VLS_OFS_TMP_LSL[i] = C_VLS_OFS_SP_LSL - VLS_MMV_LSL_L_GAIN[i]
endif
If          I VLS_OFS_TMP_LSL[i] - VLS_OFS_TMP_OLD_LSL_L_GAIN[i] I
                    < C_VLS_OFS_DIF_OK_LSL_L_GAIN
or          LV_LSL_OFS_ACT_AST[i] == 1
then        If      | VLS_OFS_TMP_LSL[i] | < C_VLS_OFS_MAX_ABSV_LSL_L_GAIN
            then    LV_VLS_OFS_LIM_LSL_L_GAIN[i] = 0
                    VLS_OFS_LSL_L_GAIN[i] = VLS_OFS_TMP_LSL[i]
            ( offset adjustment valid; VLS_OFS_LSL_L_GAIN[i] is changed )
            else    LV_VLS_OFS_LIM_LSL_L_GAIN[i] = 1
                    If      VLS_OFS_TMP_LSL[i] > 0
                    then    VLS_OFS_LSL_L_GAIN[i] =
                                    + C_VLS_OFS_MAX_ABSV_LSL_L_GAIN
                    else    VLS_OFS_LSL_L_GAIN[i] =
                                    – C_VLS_OFS_MAX_ABSV_LSL_L_GAIN
                    ( set offset voltage to threshold value which has been reached,
                      taking into account the correct sign )
                    endif
            endif
endif
VLS_OFS_TMP_OLD_LSL_L_GAIN[i] = VLS_OFS_TMP_LSL[i]
```

16.13.2 Signal acquisition and conversion for WRAF sensor

FUNCTION DESCRIPTION:

Description:

Activation:

If	STATE_LSH_UP[i] <> OFF
and	LV_LSL_IF_SPI_RST_END[i] == 1
	(Initialisation of WRAF controller ASIC completed successfully)
then	*" function enabled "*
else	*" function disabled "*

16.13.2.1 WRAF sensor interface gain-switching for Ip measurement

Description:

NOTE: The logical variable LV_SWI_GAIN_LSL_IF[i] is mapped in the interface byte STATE_LSL_IF_SPI_WR[i].
Its initialization at ECU RESET shall be done according to the corresponding bit value of C_STATE_LSL_IF_SP_WR_INI in 'ASW-BSW Interface', 4033.

Formula section:

```
If      LC_IPLSL_GAIN_SWI_ACT == 1    (Ip gain switching active)
and     LV_IPLSL_VLD[i] == 1
and     LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] == 0
then    If      STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN  ) == 1 (gain16)
        and     LV_SWI_GAIN_LSL_IF[i] == 1
        then    If      LV_VLS_UP_VLD[i] == ( 1 -> 0 )
                and     ( ( LAMB_LS_UP[i] > 1
                and     LC_VLS_UP_THD_SWI_GAIN_AFL_ACT == 1 )
                or      ( LAMB_LS_UP[i] < 1
                and     LC_VLS_UP_THD_SWI_GAIN_AFR_ACT == 1 ) )
                or      ( LC_SWI_GAIN_LSL_IF_PUC_ACT == 1
                and     LV_PUC == 1 )
                or      LAMB_LS_UP[i] > C_LAMB_LS_UP_GAIN_SWI_MAX
                or      LAMB_LS_UP[i] < C_LAMB_LS_UP_GAIN_SWI _MIN
                then    LV_SWI_GAIN_LSL_IF[i] = 0 ( gain16 -> gain8 )
                endif
        else    ( gain8 )
                If      STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN ) == 0
                and     LV_SWI_GAIN_LSL_IF[i] == 0
                and     ( C_LAMB_LS_UP_VLS_UP_H_GAIN_MIN
                                < LAMB_LS_UP[i]
                                < C_LAMB_LS_UP_VLS_UP_H_GAIN_MAX )
                or      ( LC_SWI_GAIN_LSL_IF_PUC_ACT == 1
                and     LV_PUC == 1 )
                then    LV_SWI_GAIN_LSL_IF[i] = 1
                        ( gain8 -> gain16, including hysteresis )
                endif
        endif
```

```
else    If      LV_LAMB_STI_OFF_LSL_OFS_ADJ[i] == 0
        then    LV_SWI_GAIN_LSL_IF[i] = LC_LSL_OFS_ADJ_GAIN_SEL
                ( stay in default gain mode )
        endif
endif
```

16.13.2.2 Calculation of compensated WRAF controller output voltage and WRAF sensor pumping current

Description:

Formula section:

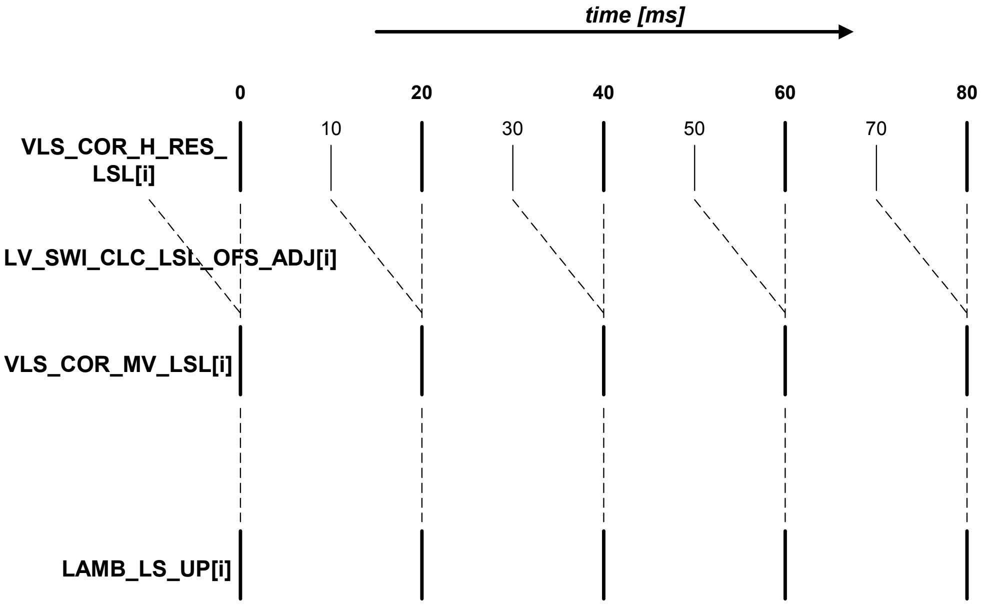
```
If      LV_VLS_UP_INIT[i] == 0      ( no gain switching currently ongoing )
and     LV_SWI_GAIN_LSL_IF[i] == STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN )
and     LV_LSL_OFS_ADJ_ACT[i] == 0
        ( OA mode of WRAF controller not active )
then    If      STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN ) == 1 ( gain16 )
        then    VLS_DELTA_TMP_LSL[i]
                        = ( VLS_UP[i] + VLS_OFS_LSL[i] - C_VLS_OFS_SP_LSL )
                If      C_VLS_OFS_DELTA_CBK_EX_MAN <> 0
                ( request for simulated diametrical misalignment of cylinder banks
                  for diagnostic test purpose )
                then    If      i == 1
                        then    VLS_DELTA_TMP_LSL[i] = VLS_DELTA_TMP_LSL[i]
                                        + C_VLS_OFS_DELTA_CBK_EX_MAN
                        else    VLS_DELTA_TMP_LSL[i] = VLS_DELTA_TMP_LSL[i]
                                        - C_VLS_OFS_DELTA_CBK_EX_MAN
                        endif
                endif
        else    ( gain8 )
                VLS_DELTA_TMP_LSL[i] = ( VLS_UP[i] + VLS_OFS_LSL_L_GAIN[i]
                        - C_VLS_OFS_SP_LSL ) * 2
        endif
        If      VLS_DELTA_TMP_LSL[i] >= 0
        then    VLS_DELTA_TMP_LSL[i] = VLS_DELTA_TMP_LSL[i]
                        * IP_FAC_COR_IPLSL_PRS_EX
```

```
            else    VLS_DELTA_TMP_LSL[i] = VLS_DELTA_TMP_LSL[i]
                            * IP_FAC_COR_IPLSL_AFR_PRS_EX
            endif
            VLS_UP_DIAG[i] = VLS_DELTA_TMP_LSL[i]  + C_VLS_OFS_SP_LSL
            If      LV_LS_UP_READY[i] == 1
            and     LV_LS_UP_READY_CDN[i] == 0
            then    VLS_DELTA_TMP_LSL[i] = VLS_DELTA_TMP_LSL[i]
                            * IP_FAC_COR_IPLSL_TTIP_LS_UP
            endif
            VLS_DELTA_TMP_LSL[i] = VLS_DELTA_TMP_LSL[i] * FAC_LSL_GAIN_AD[i]
                    + ( C_ICPLSL_COR_ADD / C_IPLSL_GAIN )
            IPLSL_COR[i] = VLS_DELTA_TMP_LSL[i]  * C_IPLSL_GAIN
else        VLS_DELTA_TMP_LSL[i]$_n$ = VLS_DELTA_TMP_LSL[i]$_{n-1}$
            VLS_UP_DIAG[i]$_n$ = VLS_UP_DIAG[i]$_{n-1}$
            IPLSL_COR[i]$_n$ = IPLSL_COR[i]$_{n-1}$
endif

% Low pass filtering of compensated signal.
If   C_FAC_LPF_VLS_COR_LSL > 0
and  LV_VLS_UP_INIT = 0
then
     VLS_COR_H_RES_LSL[i]$_n$ = VLS_COR_H_RES_LSL[i]$_{n-1}$
          + (VLS_DELTA_TMP_LSL[i] + C_VLS_OFS_SP_LSL
             - VLS_COR_H_RES_LSL[i]$_{n-1}$) * C_FAC_LPF_VLS_COR_LSL
else
     VLS_COR_H_RES_LSL[i]$_n$ = VLS_DELTA_TMP_LSL[i] + C_VLS_OFS_SP_LSL
endif


VLS_UP_COR[i] = VLS_COR_H_RES_LSL[i]
VLS_UP_COR_H_GAIN[i] = VLS_COR_H_RES_LSL[i]
```

16.13.2.3 Conversion into lambda signal

Description:

Signal flow diagram for activated averaging (LC_VLS_COR_MV_LSL_ACT == 1):



Formula section:

Calculation in case of LC_VLS_COR_MV_LSL_ACT = 1

```
If    LV_VLS_UP_INIT[i] = ( 1 -> 0 )       ( initialization of current Ip range complete )
and         LV_LSL_OFS_ADJ_ACT[i] == 0
            ( OA mode of WRAF controller not active )
then        VLS_COR_TMP_LSL[i] = VLS_COR_H_RES_LSL[i]
            VLS_COR_MV_LSL[i] = VLS_COR_H_RES_LSL[i]
            LV_LSL_UP_SHO_PER_REQ[i] = 0
            LAMB_LS_UP[i] = IP_LAMB_LS_UP
else        If     LV_VLS_UP_INIT[i] == 1 ( gain switching is currently taking place )
            or     LV_SWI_GAIN_LSL_IF[i] ≠ STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN )
            or     LV_LSL_OFS_ADJ_ACT[i] == 1
                   ( OA mode of WRAF controller active )
            then   VLS_COR_TMP_LSL[i]$_{n}$ = VLS_COR_TMP_LSL[i]$_{n-1}$
                   VLS_COR_MV_LSL[i]$_{n}$ = VLS_COR_MV_LSL[i]$_{n-1}$
                   LV_LSL_UP_SHO_PER_REQ[i] = 1
                   LAMB_LS_UP[i]$_{n}$ = LAMB_LS_UP[i]$_{n-1}$
                   LV_SWI_CLC_LSL_OFS_ADJ[i] = 0
            else   If     LV_SWI_CLC_LSL_OFS_ADJ[i] == 0
                   then   LV_SWI_CLC_LSL_OFS_ADJ[i] = 1
                          VLS_COR_TMP_LSL[i] = VLS_COR_H_RES_LSL[i]
                   else   LV_SWI_CLC_LSL_OFS_ADJ[i] = 0
                          VLS_COR_MV_LSL[i] = ( VLS_COR_H_RES_LSL[i]
                                                      + VLS_COR_TMP_LSL[i] ) / 2
                          LV_LSL_UP_SHO_PER_REQ[i] = 0
                          LAMB_LS_UP[i] = IP_LAMB_LS_UP
                   endif
            endif
endif
```

(end of calculations in case of LC_VLS_COR_MV_LSL_ACT = 1)

Calculation in case of LC_VLS_COR_MV_LSL_ACT = 0

```
If      LV_VLS_UP_INIT[i] == 1 ( gain switching is currently taking place )
or      LV_SWI_GAIN_LSL_IF[i] ≠ STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN )
or      LV_LSL_OFS_ADJ_ACT[i] == 1 ( OA mode of WRAF controller active )
then    $VLS\_COR\_TMP\_LSL[i]_n = VLS\_COR\_TMP\_LSL[i]_{n-1}$
        $VLS\_COR\_MV\_LSL[i]_n = VLS\_COR\_MV\_LSL[i]_{n-1}$
        LV_LSL_UP_SHO_PER_REQ[i] = 1
        $LAMB\_LS\_UP[i]_n = LAMB\_LS\_UP[i]_{n-1}$
        T_INI_DLY_LSL_SIG_ACQ[i] = C_T_INI_DLY_LSL_SIG_ACQ
else
        if      T_INI_DLY_LSL_SIG_ACQ[i] > 0
        then    decrement T_INI_DLY_LSL_SIG_ACQ[i]
                $VLS\_COR\_TMP\_LSL[i]_n = VLS\_COR\_TMP\_LSL[i]_{n-1}$
                $VLS\_COR\_MV\_LSL[i]_n = VLS\_COR\_MV\_LSL[i]_{n-1}$
                LV_LSL_UP_SHO_PER_REQ[i] = 1
                $LAMB\_LS\_UP[i]_n = LAMB\_LS\_UP[i]_{n-1}$
        else    VLS_COR_TMP_LSL[i] = VLS_COR_H_RES_LSL[i]
                VLS_COR_MV_LSL[i] = VLS_COR_H_RES_LSL[i]
                LV_LSL_UP_SHO_PER_REQ[i] = 0
                LAMB_LS_UP[i] = IP_LAMB_LS_UP
endif
```

(end of calculations in case of LC_VLS_COR_MV_LSL_ACT = 1)

Set LAMB_LS_UP_MIN_TMP = 32, and LAMB_LS_UP_MV_TMP = 0, i_ctr_tmp = 0, prior to execution of FOR loop depending on [i].

LAMB_LS_UP_MIN_TMP = **MIN** (LAMB_LS_UP[i], LAMB_LS_UP_MIN_TMP)

LAMB_LS_UP_MIN = LAMB_LS_UP_MIN_TMP

LAMB_LS_UP_MV_TMP = LAMB_LS_UP_MV_TMP + LAMB_LS_UP[i]

increment i_ctr_tmp

LAMB_LS_UP_MV = LAMB_LS_UP_MV_TMP / i_ctr_tmp

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_LAMB_LS_UP_MIN_OFS_ADJ	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Minimum lambda threshold for the offset calculation					
C_LAMB_LS_UP_MAX_OFS_ADJ	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Maximum lambda threshold for the offset calculation					
C_T_WAIT_LSL_OFS_ADJ	1	0...FFFFH	0...655.35	0.01	[s]
Lock time after an offset calculation prior to the next calculation for warm engine					
C_T_WAIT_LSL_OFS_ADJ_RED	1	0...FFFFH	0...655.35	0.01	[s]
Lock time after an offset calculation prior to the next calculation during engine warm-up					
C_NR_CYC_T_WAIT_LSL_OFS_ADJ_RED	1	0...FFH	0...255	1	[-]
Number of cycles for application of reduced repetition rate for offset adjustment during engine warm-up					
C_CRLC_VLS_UP_OFS_ADJ	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Correlation constant for filtering of mean value of WRAF sensor lambda signal for Ip gain16					
C_CRLC_VLS_UP_OFS_ADJ_L_GAIN	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Correlation constant for filtering of mean value of WRAF sensor lambda signal for Ip gain8					
C_VLS_OFS_DIF_OK_LSL	1	0...7FFFH	0...4.99984	1.53E-04	[V]
Maximum permitted deviation of two successive offset values for Ip gain16					
C_VLS_OFS_DIF_OK_LSL_L_GAIN	1	0...7FFFH	0...4.99984	1.53E-04	[V]
Maximum permitted deviation of two successive offset values for Ip gain8					
C_VLS_OFS_SP_LSL	1	0...7FFFH	0...4.99984	1.53E-04	[V]
Nominal WRAF controller output offset voltage (valid for Ip = 0 or lambda = 1)					
C_T_DLY_LSL_OFS_ADJ_IS	1	0...FFH	0...2.55	0.01	[s]
Delay time for activation of offset adjustment upon entering engine state idle speed					
C_T_DLY_LSL_OFS_ADJ	1	0...FFH	0...2.55	0.01	[s]
Delay time for the offset calculation after activating offset adjustment					
C_T_DLY_LAMB_STI_OFF_OFS_ADJ	1	0...FFFFH	0...655.35	0.01	[s]
Delay time for the offset calculation after switching-off forced lambda stimulation					
C_VLS_OFS_MAX_ABSV_LSL	1	0...7FFFH	0...4.99984	1.53E-04	[V]
Limit value for the offset as a maximum value (absolute value) for Ip gain16					
C_VLS_OFS_MAX_ABSV_LSL_L_GAIN	1	0...7FFFH	0...4.99984	1.53E-04	[V]
Limit value for the offset as a maximum value (absolute value) for Ip gain8					
LC_LSL_OFS_ADJ_ENA	1	0...1H	0...1	1	[-]
Switch to enable offset adjustment in every engine state					
LC_LSL_OFS_ADJ_H_L_GAIN	1	0...1H	0...1	1	[-]
Switch to enable offset adjustment in both Ip measurement ranges					
LC_LSL_OFS_ADJ_GAIN_SEL	1	0...1H	0...1	1	[-]
Switch to choose Ip measurement range in which singular offset adjustment shall be performed (‘0’ = gain8, ‘1’ = gain16)					
LC_IPLSL_GAIN_SWI_ACT	1	0...1H	0...1	1	[-]
Switch to enable gain switching for measurement of pumping current					
LC_CAT_DIAG_ACT_WAIT	1	0...1H	0...1	1	[-]
Switch to grant catalyst diagnosis priority to be completed without interruption by OA					
LC_LAMB_PLS_ACT_NEUT	1	0...1H	0...1	1	[-]
Switch to perform OA regardless of current status of forced lambda stimulation					
LC_CTR_ERR_LSL_IF_SAVE_MAN_INC	1	0...1H	0...1	1	[-]
Switch enabling simulation of SPI communication error with WRAF controller					
C_LAMB_LS_UP_GAIN_SWI_MIN	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Rich lambda threshold for switching of Ip gain16 -> gain8					
C_LAMB_LS_UP_GAIN_SWI_MAX	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Lean lambda threshold for switching of Ip gain16 -> gain8					
C_LAMB_LS_UP_VLS_UP_H_GAIN_MIN	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Bottom limit of ‘Ip gain16’ band in terms of lambda					
C_LAMB_LS_UP_VLS_UP_H_GAIN_MAX	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Upper limit of ‘Ip gain16’ band in terms of lambda					
C_IPLSL_GAIN	1	0...FFFFH	0...5.0384	7.69E-05	[mA/V]

Total gain of WRAF sensor interface circuit (Ip / (V_LMBD_IP - C_VLS_OFS_SP_LSL)) for Ip gain16Ip: for NTK UEGO = pumping current through sensor, for BOSCH LSU = current through measurement resistor					
IP_FAC_COR_IPLSL_PRS_EX	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_PRS_EX_IP_FAC_COR_IPLSL	8	0...FFFFH	0...5434	0.0829175	[hPa]
Factor for exhaust gas pressure correction at lean conditions					
IP_FAC_COR_IPLSL_AFR_PRS_EX	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_PRS_EX_IP_FAC_COR_IPLSL	8	0...FFFFH	0...5434	0.0829175	[hPa]
Factor for exhaust gas pressure correction at rich conditions					
IP_FAC_COR_IPLSL_TTIP_LS_UP	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_TTIP_LS_UP_IP_FAC_COR_IPLSL	8	0...FFFFH	-2048... 2047.9375	0.0625	[°C]
Factor for correction of tip temperature influence					
C_VB_MIN_OFS_ADJ	1	0...FFH	0...25.89843	0.1015625	[V]
Minimum battery voltage for accurate offset adjustment based on related reference voltages					
LC_VLS_UP_THD_SWI_GAIN_AFR_ACT	1	0...1H	0...1	1	[-]
Boolean flag for activation of Ip gain switching due to possible VSL_UP_RAW[i] rich side signal saturation					
LC_LSL_OFS_GAIN_SWI_CYC	1	0...1H	0...1	1	[-]
Switch to choose alternating offset adjustment in Ip gain16 / gain8 mode					
C_T_DLY_LSL_OFS_ADJ_PUC_END	1	0...FFFFH	0...655.35	0.01	[s]
Delay time for activation of offset adjustment after leaving fuel cut-off phase					
C_VLS_OFS_DELTA_ADD	1	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Additive share of correlation between offset value in Ip gain16, and Ip gain8					
C_T_DLY_LSL_OFS_ADJ_CYC	1	0...FFH	0...2.55	0.01	[s]
Delay time for second offset determination after switching Ip ranges					
LC_SWI_GAIN_LSL_IF_PUC_ACT	1	0...1H	0...1	1	[-]
Switch to activate Ip gain switching gain16 -> gain8 during PUC					
C_VLS_OFS_DELTA_CBK_EX_MAN	1	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Diametrical shift of offset values for different exhaust banks for diagnostic test purpose					
LC_VLS_UP_THD_SWI_GAIN_AFL_ACT	1	0...1H	0...1	1	[-]
Boolean flag for activation of Ip gain switching due to possible VLS_UP_RAW[i] lean side signal saturation					
IP_LAMB_LS_UP	32	0...7FFFH	0...31.99902	0.9766e-3	[-]
LDP_VLS_COR_MV_IP_LAMB_LS_UP	32	0...FFFFH	0...9.99984	0.1526e-3	[V]
Characteristic curve of the lambda sensor (LAMB_LS_UP[i] vs. compensated VLS_COR_MV_LSL[i])					
LC_VLS_COR_MV_LSL_ACT	1	0...1H	0...1	1	[-]
Switch for (de)activation of averaging of VLS_COR_H_RES_LSL[i] prior to conversion					
C_ICPLSL_COR_ADD	1	8000...7FFFH	-12.596... 12.59561	0.3844e-3	[mA]
Additive correction of pump current due to adjusted value of pumped reference current					
C_FAC_LPF_VLS_COR_LSL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Low pass filter constant for filtering of corrected output signal of WRAF sensor					
C_T_INI_DLY_LSL_SIG_ACQ	1	0...FFH	0...2.55	0.01	[s]
Delay of acquisition after freeze of signal. Assures that segment synchronous mean value calc has reached final value.					

16.14 Application Incidences for ‘Offset adjustment of oxygen sensor (linear control)’

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_LSL_OFS_ADJ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Boolean flag indicating that upstream oxygen sensor Ip offset adjustment shall be inhibited					
T_HOM_ACT_OFS_ADJ	O/V	0...FFFFH	0...655.35	0.01	[s]
Time in homogenous mode for enabling offset adjustment					

Input data:

STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]	LV_LAM_AD_INJ_ACT	C_T_DLY_TTIP_MES_OBD_LSH_UP	C_T_DLY_TTIP_RES_OBD_LSH_UP
LV_HOM_ACT			

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned
(NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

The Boolean flag LV_INH_LSL_OFS_ADJ[i] shall indicate whether the upstream oxygen sensor Ip offset adjustment shall be permitted to take place.

When set, Ip offset adjustment shall be inhibited.

Description:

Application conditions:

Initialisation:

The following initialisation shall be carried out **after RESET**, **after engine start**, or **upon erase of error management**.

LV_INH_LSL_OFS_ADJ [i] = 0

T_HOM_ACT_OFS_ADJ = max phys. value

Recurrence:

T_SAMPLE = 10ms

Activation:

LV_ST_END == 1

Deactivation:

--

Formula section:

```
IF      LV_HOM_ACT = 0
THEN    T_HOM_ACT_OFS_ADJ = 0
ELSE    increment T_HOM_ACT_OFS_ADJ  (timer shall saturate at max phys. limit)
ENDIF

If      STATE_LSL_IF_SPI_RD[i] bit LSL_ENA == 0
or      LV_LAM_AD_INJ_ACT = 1
or      (T_HOM_ACT_OFS_ADJ <= (C_T_DLY_TTIP_MES_OBD_LSH_UP + C_T_DLY_TTIP_RES_OBD_LSH_UP))
then    LV_INH_LSL_OFS_ADJ[i] = 1
else    LV_INH_LSL_OFS_ADJ[i] = 0
endif
```

16.15 Operability detection for the linear lambda sensor

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LS_UP_READY[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Logical variable for operative readiness of upstream oxygen sensor					
LV_LS_UP_READY_CDN[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Logical variable for forced operative readiness of upstream oxygen sensor					
LV_LS_UP_READY_RAW[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Logical variable for raw operative readiness of upstream oxygen sensor					
LV_LSL_UP_ERR_SUSP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
error suspicion of upstream oxygen sensor signal, set reduced limitation of Lambda controller					
LAMB_DE_INT_LS_UP_READY[NC_CBK_EX_NR]	V	0...FFFFH	0...63.9990234	9.76563E-4	-
Value as a measure of the sensor signal variation vs. time					
T_TOUT_LS_UP_READY[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Timer for forced activation of linear lambda sensor					
LV_LSL_UP_SUSP_ENA[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Enable setting of LV_LSL_UP_ERR_SUSP once the WRAF sensor is ready					
LAMB_MV_LS_UP_READY[NC_CBK_EX_NR]	V	0...7FFFH	0...31.9990234	9.76563E-4	-
Mean value of the converted upstream oxygen sensor signal in terms of A/F-ratio					
LV_TEG_MIN_DLY[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Logical variable indicating sufficient delay after dew point detection prior to forced sensor activation					
T_DLY_CHK_LS_UP_READY[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Delay timer for start of readiness detection after engine start					
T_DLY_LS_UP_READY_TEG_MIN[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Delay time after dew point detection prior to forced sensor activation					
T_LAMB_DE_INT_RST[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Timer for duration of summation of sensor signal dispersion up to reset thereof					

Input data:

LAMB_LS_UP[NC_CBK_EX_NR]	LV_PUC	LV_ST_END	LV_TEMP_DEW_LS_UP[NC_CBK_EX_NR]
LV_TTIP_MES_VLD_LS_UP[NC_CBK_EX_NR]	LV_V_REF_VLD_R_IT_LS_UP[NC_CBK_EX_NR]	MAF_INT_PUC_ACT	NC_CBK_EX_NR
STATE_LSH_UP[NC_CBK_EX_NR]	TCO_ST	TTIP_MES_LS_UP[NC_CBK_EX_NR]	LV_IPLSL_VLD[NC_CBK_EX_NR]
LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]	STATE_ERR_IPLSL[NC_CBK_EX_NR]
LV_LSL_DEAC[NC_CBK_EX_NR]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_LAMB_LS_UP_READY	1	0...FFH	0...0.99609375	0.00390625	-
Correlation constant for mean value calculation of lambda sensor output signal					
C_LAMB_DE_INT_THD_LS_UP_READY	1	0...FFFFH	0...63.9990234	9.76563E-4	-
Sensor signal dispersion threshold for readiness detection					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_THD_MAX_LS_UP_READY	1	0...7FFH	0...1.99902344	9.76563E-4	-
Upper threshold of sensor signal for readiness detection					
C_LAMB_THD_MIN_LS_UP_READY	1	0...7FFH	0...1.99902344	9.76563E-4	-
Lower threshold of sensor signal for readiness detection					
C_MAF_INT_PUC_MAX_LS_UP_READY	1	0...FFFFH	0...2.91267E+3	0.04444444	g
Maximum MAF integral value during PUC used for reset of operability					
C_TTIP_MIN_LS_UP_READY	1	8000...7FFFH	-2.048E+3 ... 2.04794E+3	0.0625	°C
Minimum oxygen sensor ceramic temperature for save operation					
C_T_DLY_LS_UP_READY_TEG_MIN	1	0...FFFFH	0...1.3107E+3	0.02	s
Threshold of delay time after dew point detection prior to forced sensor activation					
C_T_LAMB_DE_INT_RST	1	0...FFFFH	0...1.3107E+3	0.02	s
Time for sensor dispersion calculation up to reset					
LC_TEG_MIN_DEAC	1	0...1H	0...1	1	-
Test switch to disable dependance of readiness detection on dew point detection					
LC_TEG_MIN_THD_DEAC_TEST	1	0...1H	0...1	1	-
Test switch to check function behaviour at revocation of dew point					
ID_T_DLY_CHK_LS_UP_READY	8	0...FFFFH	0...1.3107E+3	0.02	s
LDP_TCO_ST_ID_T_DLY_CHK_LS_UP	8	0...FEH	-48...142.5	0.75	[°C]
Delay time for begin of readiness detection after engine start					
ID_T_TOUT_LS_UP_READY	4	0...FFFFH	0...1.3107E+3	0.02	s
LDP_TCO_ST_ID_T_TOUT_LS_UP	4	0...FEH	-48...142.5	0.75	[°C]
Time-out constant for readiness detection -> forced readiness detection					

16.15.1 FUNCTION DESCRIPTION

General information:

This is a particular **version for linear oxygen sensors to be used if the oxygen sensor ceramic temperature is determined by internal resistance measurement of the Nernst cell**. It requires that the oscillator can be switched on and off depending on LV_LS_UP_READY_RAW[i] which is determined in this function.

The operability check for the lambda sensor shall detect the moment at which the sensor reaches principal ability to generate plausible output signals (once pump current control loop has successfully been activated!). It shall analyse the sensor signal and distinguish between two stages of operative readiness. The criterion for raw sensor operability may be the variation of the sensor signal as well as the crossing of a maximum or minimum threshold by the sensor signal or simply when a delay has been exceeded. When the raw operative readiness has been sensed, the full operative readiness shall be determined by waiting until the minimum oxygen sensor ceramic temperature for safe sensor operation has been reached.

The calculation shall be done for all exhaust cylinder banks.

For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then

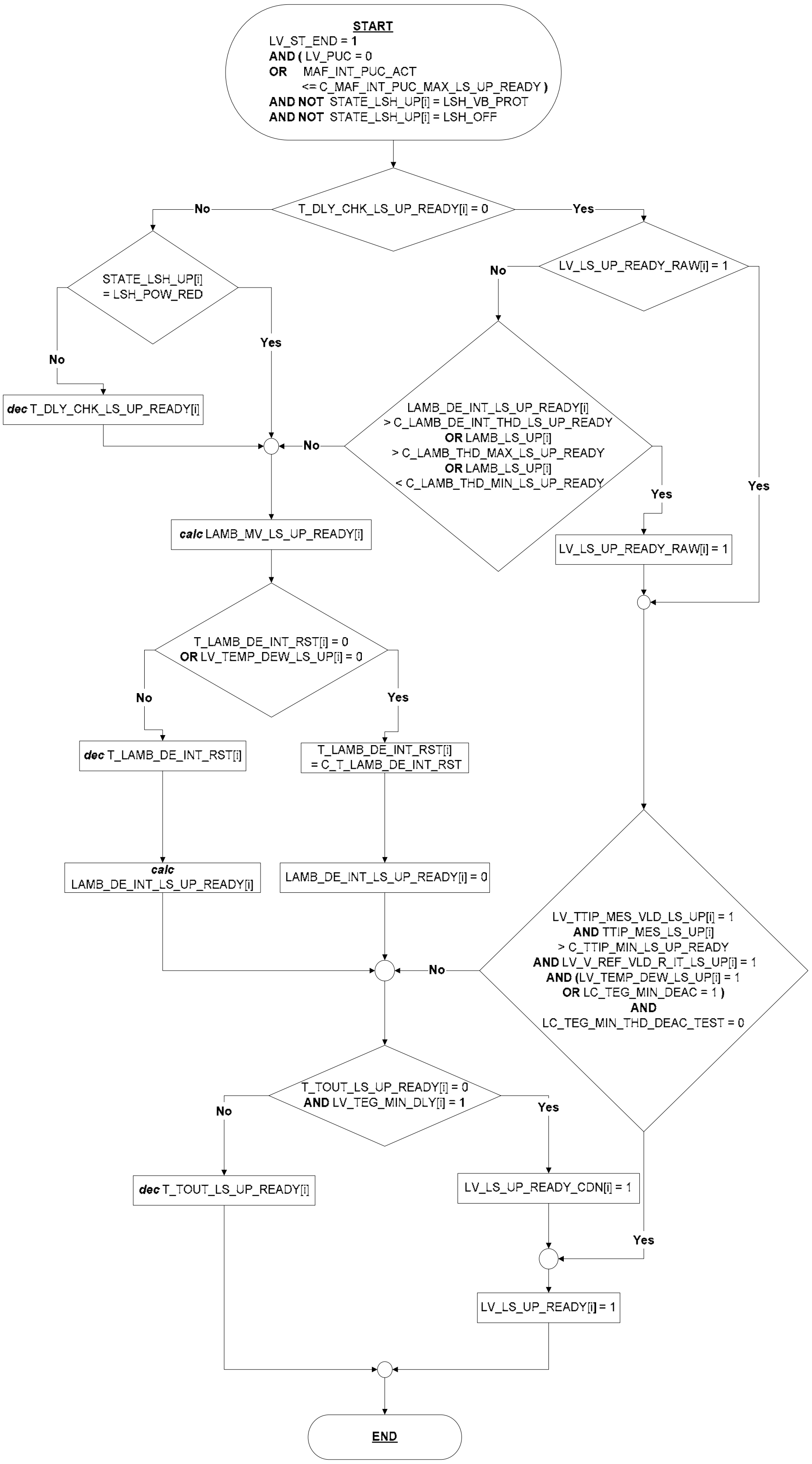
i = 1, for exhaust cylinder bank 1

i = 2, for exhaust cylinder bank 2

otherwise (NC_CBK_EX_NR = 1)

i = 1, for single exhaust cylinder bank.

Signal flow diagram:



Description:

The function shall be activated with the following conditions being fulfilled:

Engine start is left, i.e. "exit START" (LV_ST_END = 1) *and*

Upstream heater has left the battery-voltage protection mode (STATE_LSH_UP[i] ≠ LSH_VB_PROT) *and*

Upstream heater is <u>not</u> in off mode (STATE_LSH_UP[i] ≠ LSH_OFF) *and*

Threshold value of MAF integral is <u>not</u> exceeded during PUC.

The function shall be deactived once any one of these conditions is not fulfilled anymore.

Function re-start shall also take place upon erase of error management.

Precondition for starting the operability check is that the delay time ID_T_DLY_CHK_LS_UP_READY has expired.

The sensor is subsequently recognized as being <u>fully</u> operable (LV_LS_UP_READY[i] = 1) if

(

the value LAMB_DE_INT_LS_UP_READY[i] exceeds the threshold C_LAMB_DE_INT_THD_LS_UP_READY once

OR if the sensor output signal LAMB_LS_UP[i] exceeds the window defined by the limiting values C_LAMB_THD_MIN_LS_UP_READY and C_LAMB_THD_MAX_LS_UP_READY

< This state is called <u>raw</u> sensor operability (LV_LS_UP_READY_RAW[i] = 1) >

AND

the value for ceramic temperature is valid (LV_TTIP_MES_VLD_LS_UP[i] = 1)

AND the value of the reference voltage is valid (LV_V_REF_VLD_R_IT_LS_UP[i] = 1)

AND the temperature of the sensor is higher than a threshold (TTIP_MES_LS_UP[i] > C_TTIP_MIN_LS_UP_READY)

AND

the dew point was exceeded (LV_TEMP_DEW_LS_UP[i] = 1)

OR LC_TEG_MIN_DEAC = 1 (only for test purpose in conjunction with forced sensor heating prior to engine start!)

AND

LC_TEG_MIN_THD_DEAC_TEST = 0

)

OR

the timer for forced activation has expired (T_TOUT_LS_UP_READY[i] = 0)

AND dew point was exceeded (LV_TEMP_DEW_LS_UP[i] = 1) for certain time delay (LV_TEG_MIN_DLY[i] = 1)

< This state is called forced sensor operability (LV_LS_UP_READY_CDN[i] = 1) >

The logical value LV_LS_UP_READY_RAW[i] detects changes in sensor signal as basis for recognition of the raw sensor operability.

REMARK: The CARB has defined that the sensor must be recognized as being operable after a certain time upon engine start. This requirement is realized using the timer T_TOUT_LS_UP_READY[i].

If the timer has expired the sensor will be set operable (LV_LS_UP_READY[i] = 1) and in addition the conditional readiness bit (LV_LS_UP_READY_CDN[i]) will be set to “1”, initializing the forced activation of sensor.

The flag LV_LSL_UP_ERR_SUSP[i] indicates that a fault may be present in the sensor, which is not yet debounced but may lead to wrong lambda measurement. It may be used to restrict the usage of the sensor signal by functions like lambda control or adaptation or diagnosis functions. The detection of these conditions is effectively masked with LV_LSL_UP_SUSP_ENA[i], which ensures that LV_LSL_UP_ERR_SUSP[i] is not set during sensor warmup.

.

Application Condition

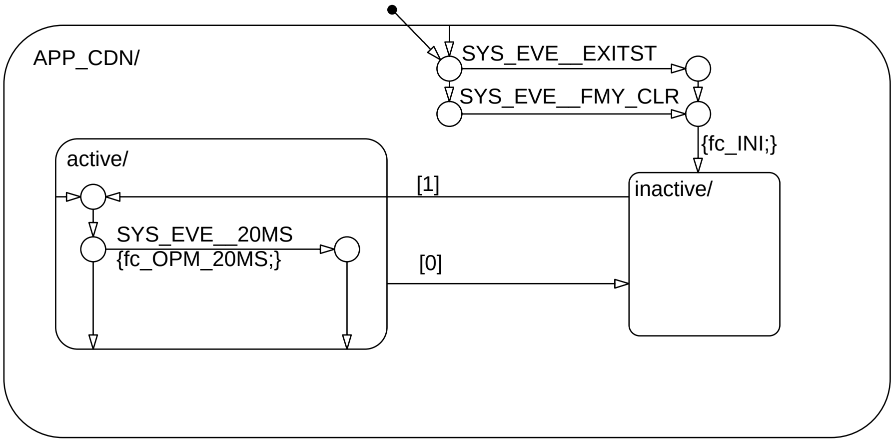
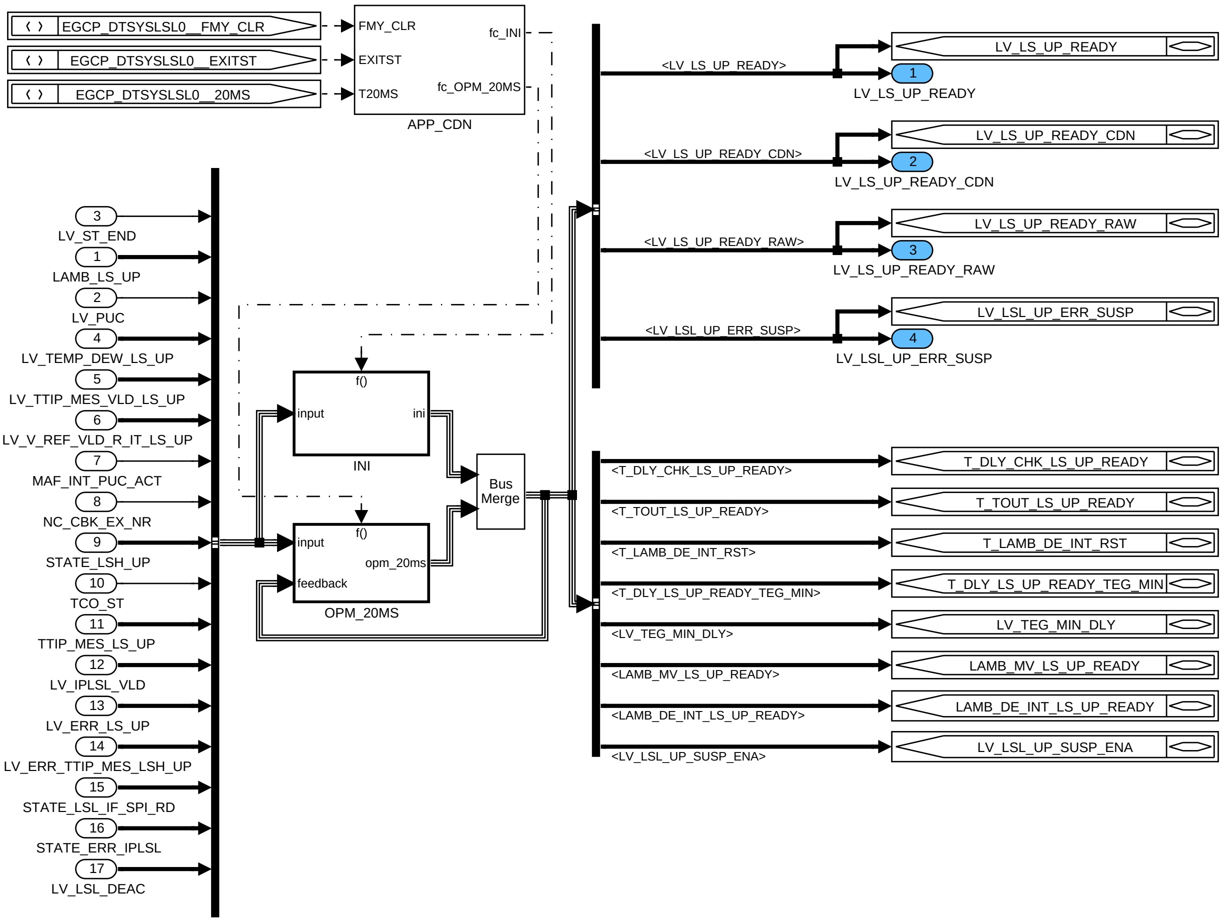


Figure 38 EGCP_DTSYSLSL0/ APP_CDN/ Chart



SDA_SRS / SDA 4.0 14−Mar−2006

Figure 39 EGCP_DTSYSLSL0

16.15.1.1 INITIALIZATION

Initialization at reset and cleaning of fault memory

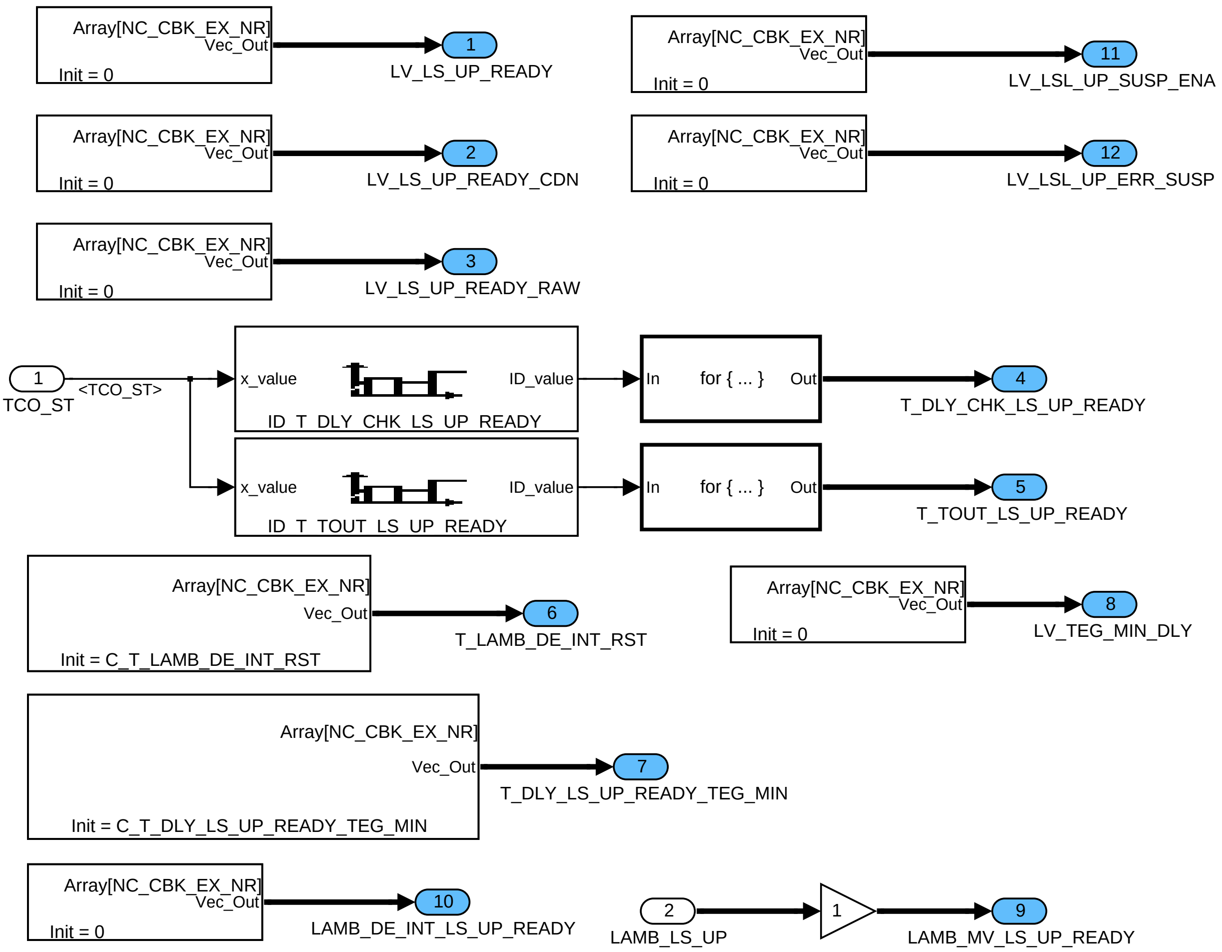


Figure 40 EGCP_DTSYSLSL0/ INI/ INI

16.15.1.2 FORMULA SECTION

Content of the formula section:

SECTION 1: Check of the general condition for initialization or activation

SECTION 2: Calculation active (FCN_ACT)

SECTION 2.1: Check of full operative readiness

SECTION 2.2: Check of sensor readiness

SECTION 2.2.1: Readiness detection (READ_DET)

SECTION 2.2.1.1: Check of the delay time for start of readiness

SECTION 2.2.1.2: Calculation of mean value and sensor signal variation

SECTION 2.2.1.3: Calculation of the delay time after dew point detection

SECTION 2.2.1.4: Forced operative readiness

SECTION 2.3: Generation of signal quality flags

SECTION 3: Initialization due to MAF integral exceeding during PUC (INI_MAF_INT_EXC)

SECTION 4: Initialization in case calculation not active (INI_VL_PROT)

SECTION 1: Check of the general condition for initialization or activation

fc_FCN_ACT triggers the calculation of whole Section 2

fc_INI_MAF_INT_EXC triggers the calculation of Section 3

fc_INI_VL_PROT triggers the calculation of Section 4

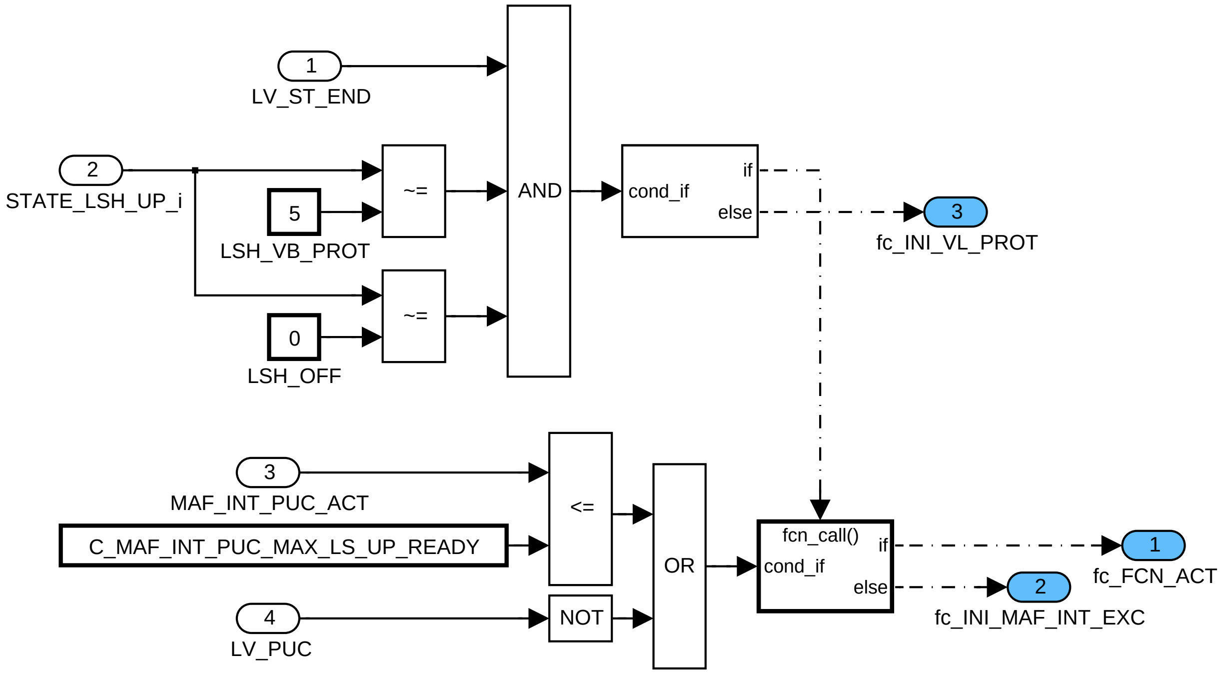


Figure 41 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CHK_ACT_CDN

SECTION 2: Calculation active (fc_FCN_ACT)

SECTION 2.1: Check of full operative readiness

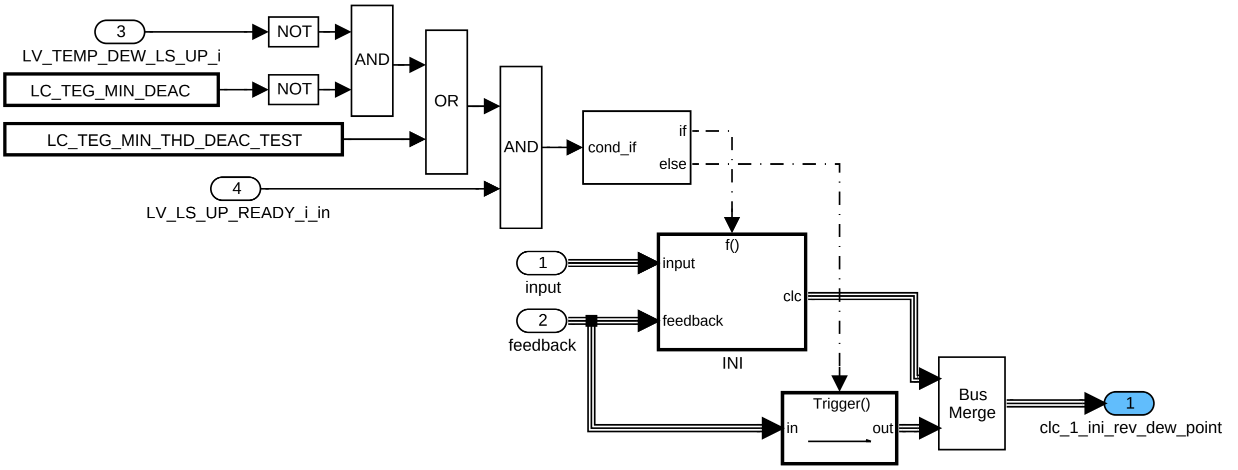


Figure 42 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_1_INI_REV_DEW_POINT

Initialization due to revocation of dew point upon successful initial detection of full operative readiness

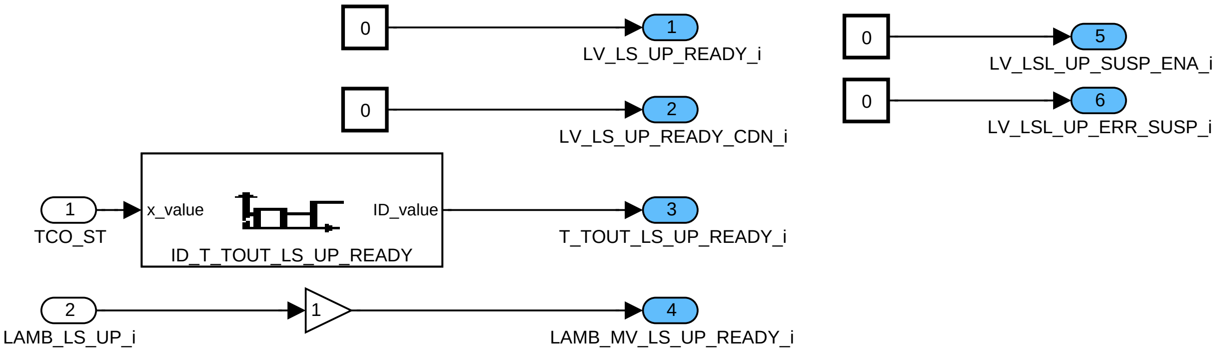


Figure 43 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_1_INI_REV_DEW_POINT/ INI/ INI

SECTION 2.2: Check of sensor readiness

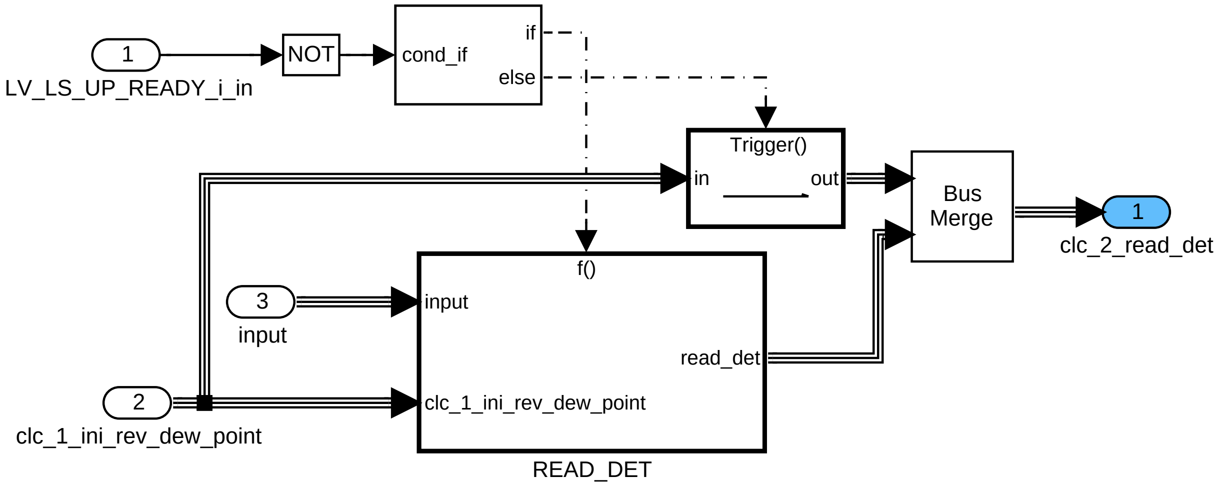


Figure 44 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET

SECTION 2.2.1: Readiness detection (READ_DET)

SECTION 2.2.1.1: Check of the delay time for start of readiness

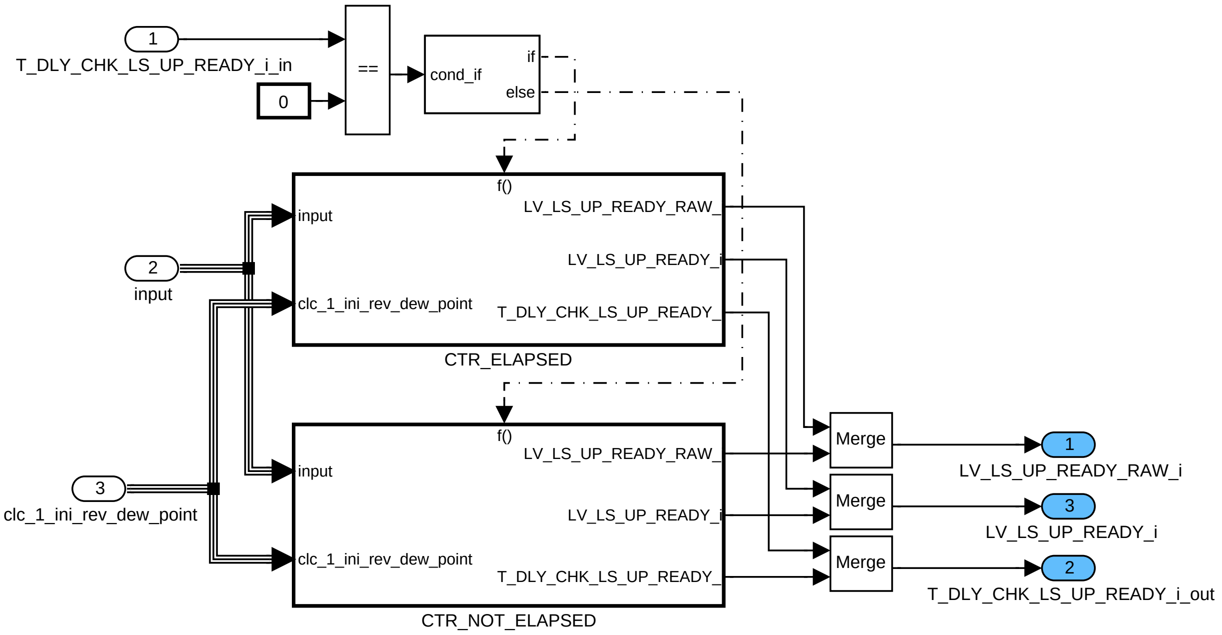


Figure 45 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_1_T_DLY_CHK

Delay time elapsed (CTR_ELAPSED) – recognition of sensor operability

The detection of sensor operability is done in two calculations:

Calculation 1: RAW sensor operability detection

Calculation 2: Full sensor operability detection

Calculation 1: RAW sensor operability detection

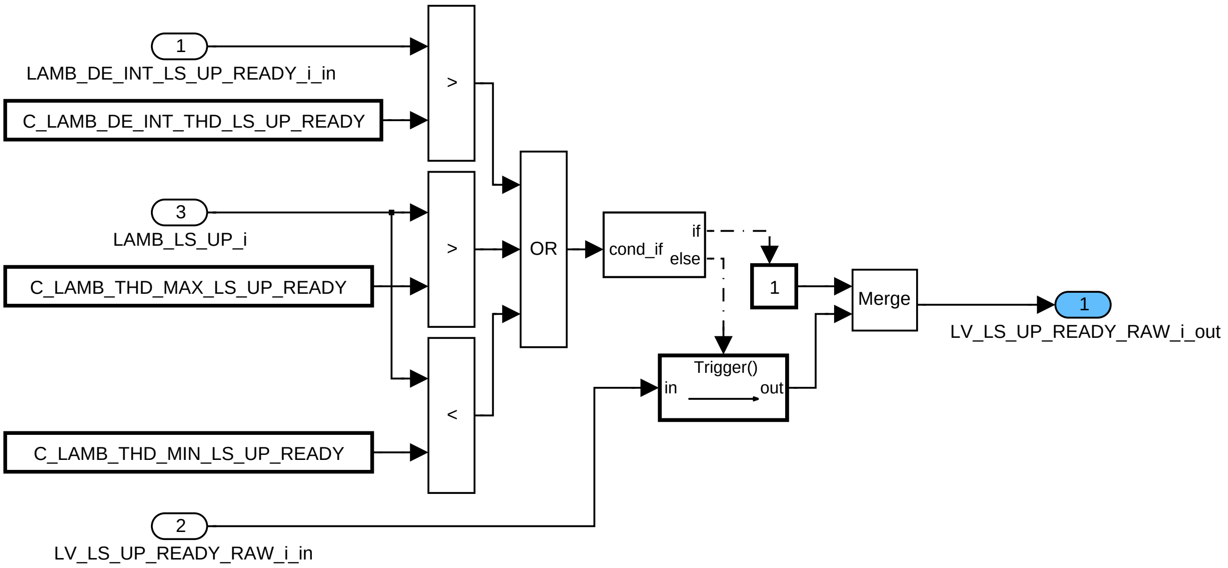


Figure 46 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_1_T_DLY_CHK/ CTR_ELAPSED/ CTR_ELAPSED_1

Calculation 2: Full sensor operability detection

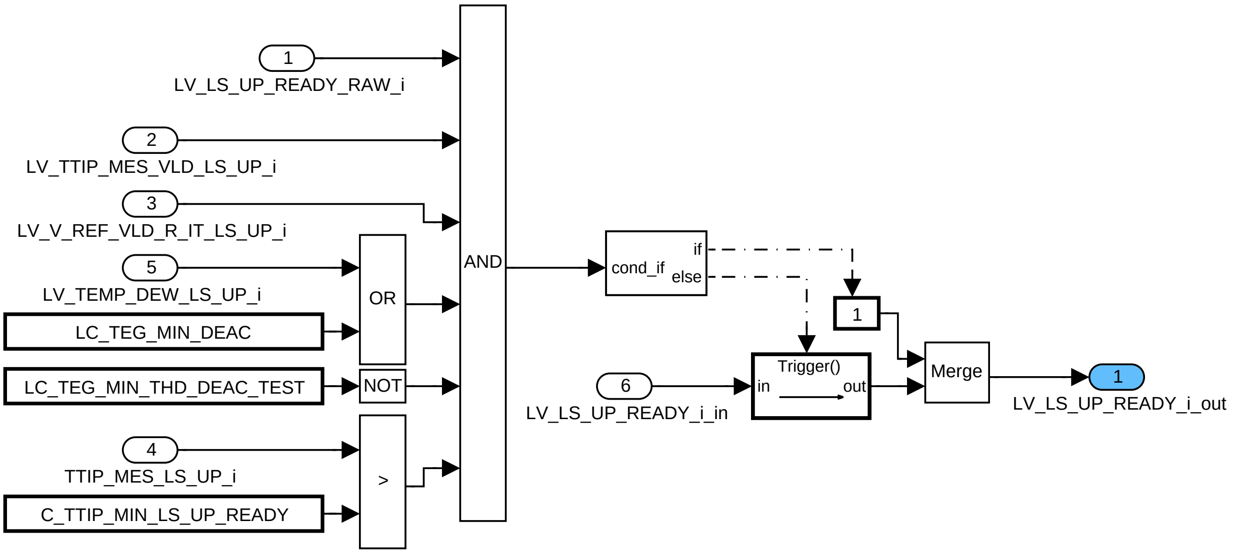


Figure 47 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_1_T_DLY_CHK/ CTR_ELAPSED/ CTR_ELAPSED_2

Delay time not elapsed (CTR_NOT_ELAPSED)

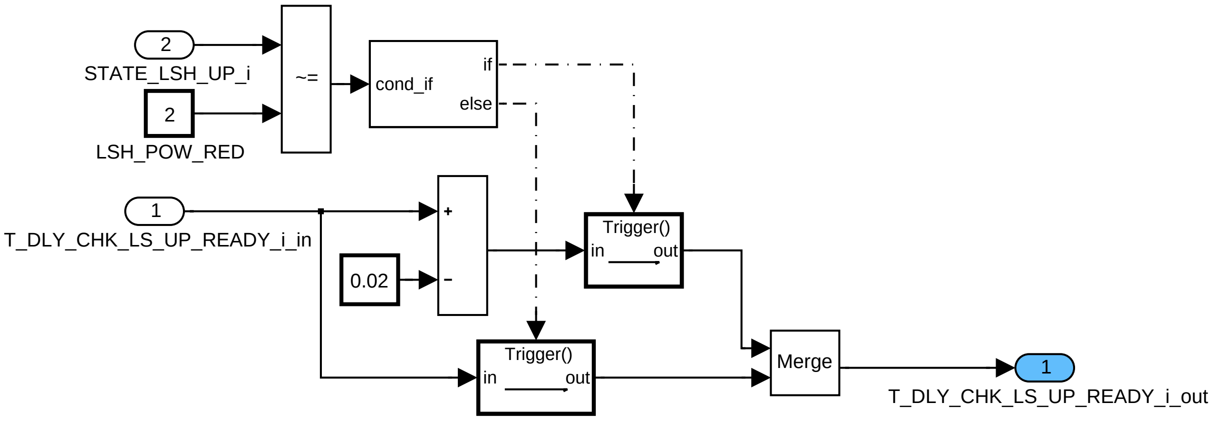


Figure 48 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_1_T_DLY_CHK/ CTR_NOT_ELAPSED/ CTR_NOT_ELAPSED

SECTION 2.2.1.2: Calculation of mean value and sensor signal variation

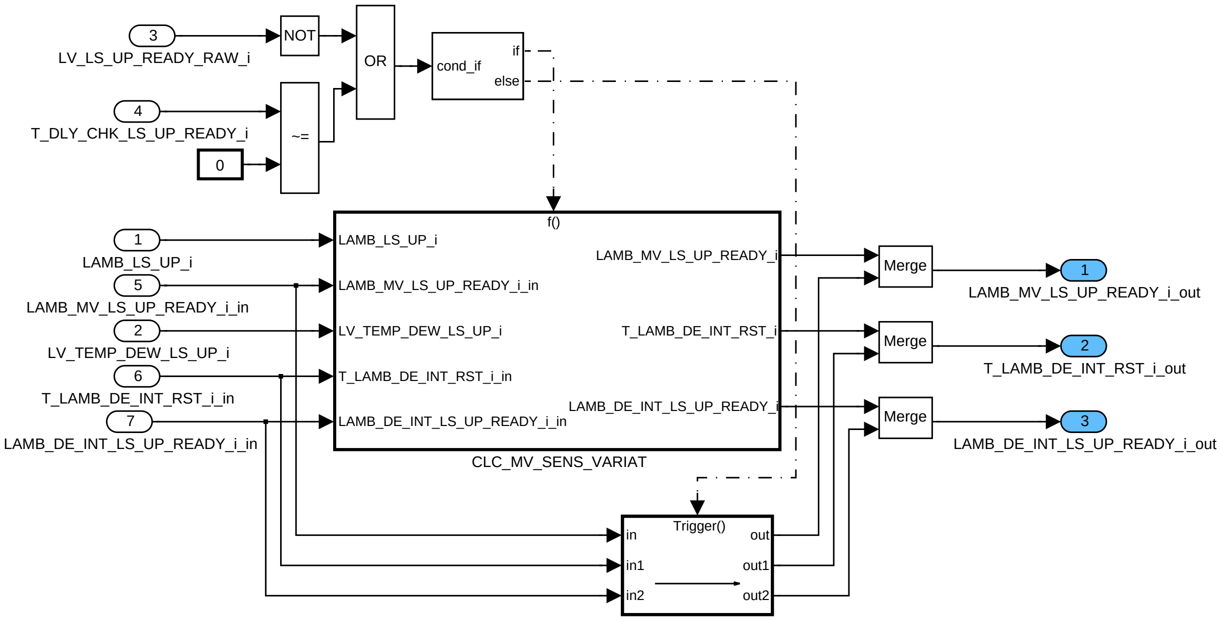


Figure 49 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_2_MV_SENS_SIG

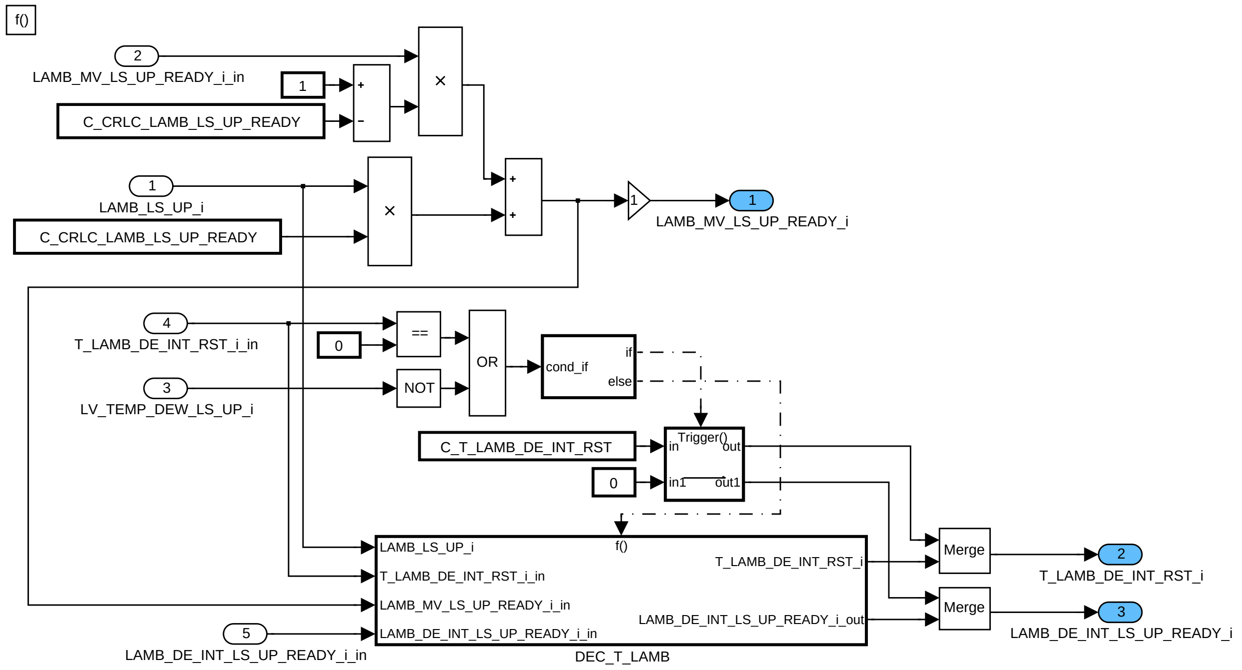


Figure 50 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_2_MV_SENS_SIG/ CLC_MV_SENS_VARIAT

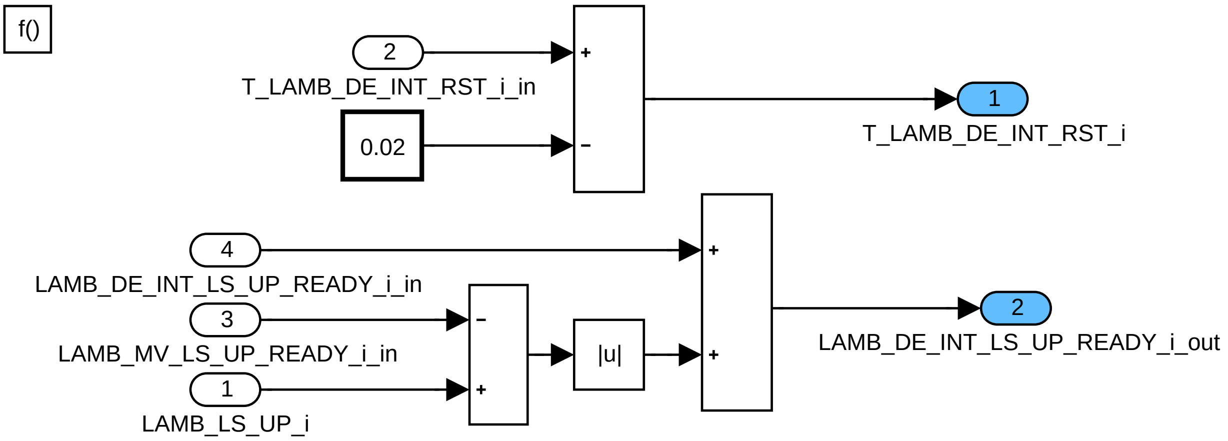


Figure 51 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_2_MV_SENS_SIG/ CLC_MV_SENS_VARIAT/ DEC_T_LAMB

SECTION 2.2.1.3: Calculation of the delay time after dew point detection

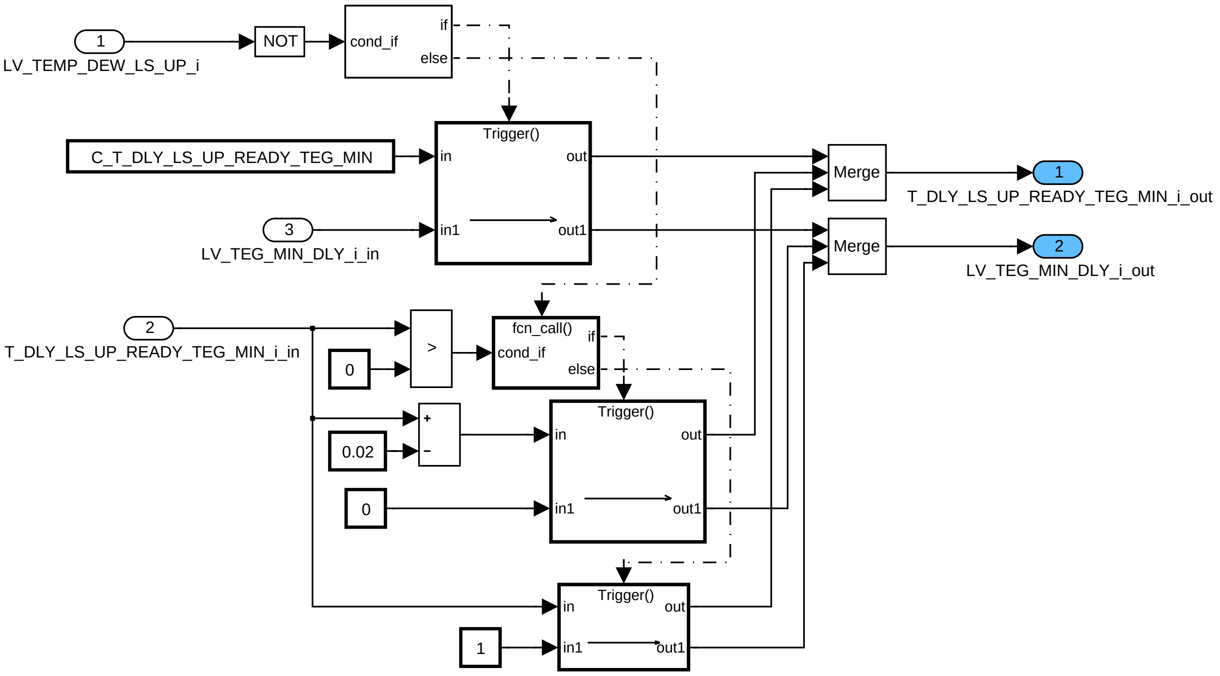


Figure 52 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_3_DLY_TIME

SECTION 2.2.1.4: Forced operative readiness

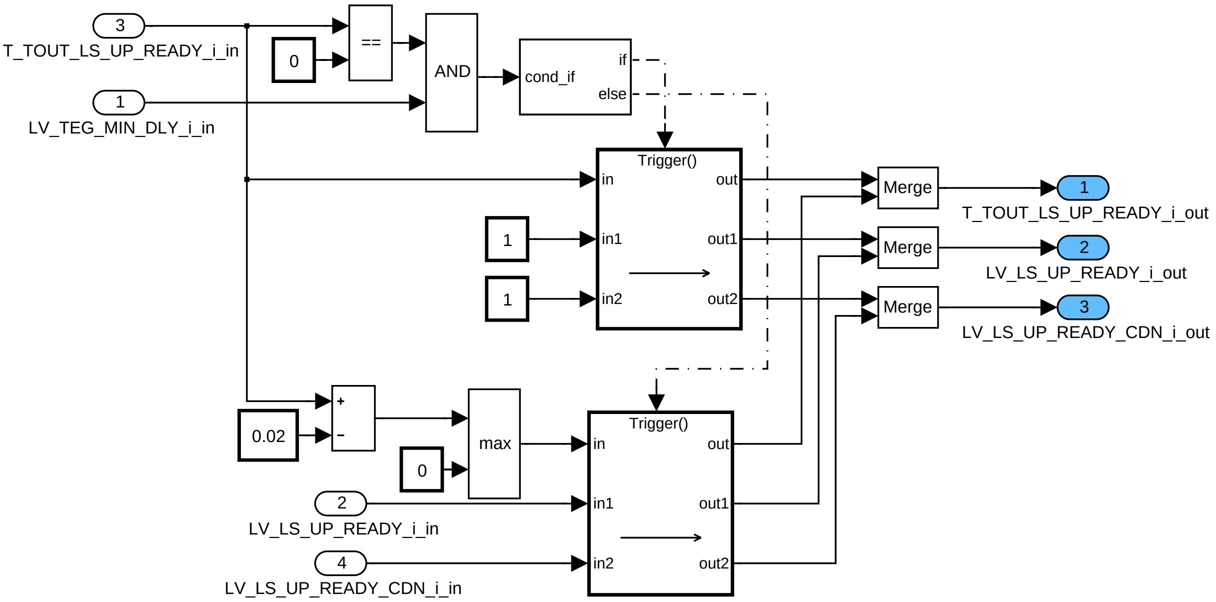


Figure 53 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_2_READ_DET/ READ_DET/ CLC_4_CHK_FORCE_ACT_SENS

SECTION 2.3: Generation of signal quality flags

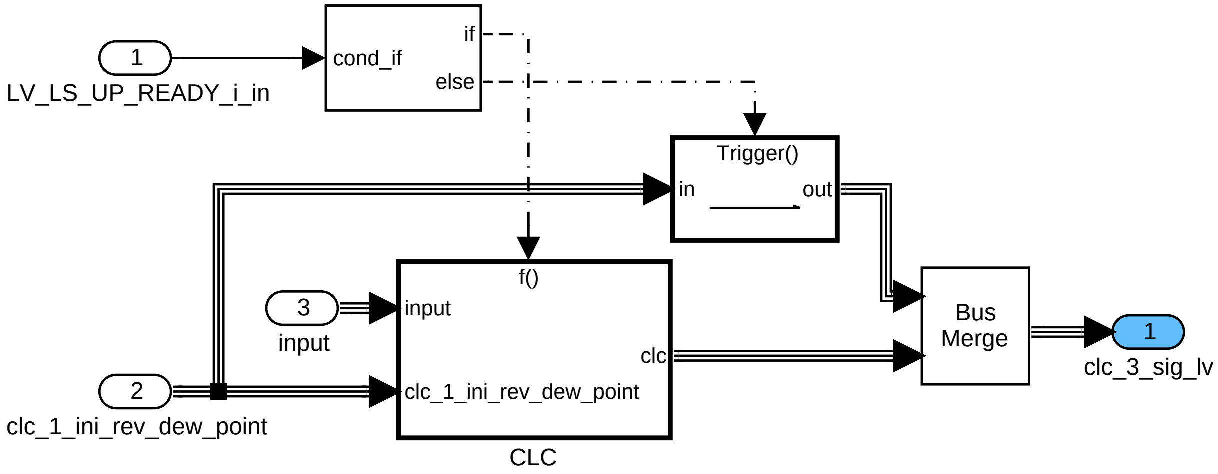


Figure 54 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_3_SIG_LV

Calculation active (CLC)

Unlock the error suspicion flag

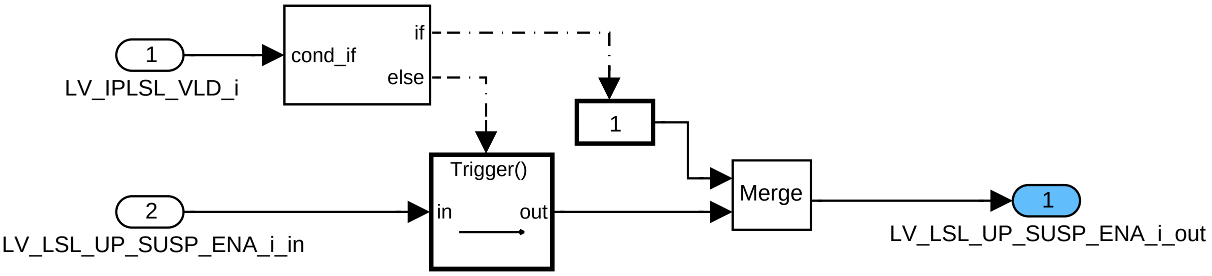


Figure 55 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_3_SIG_LV/ CLC/ CLC_1_IPLSL_VLD

Error suspicion present

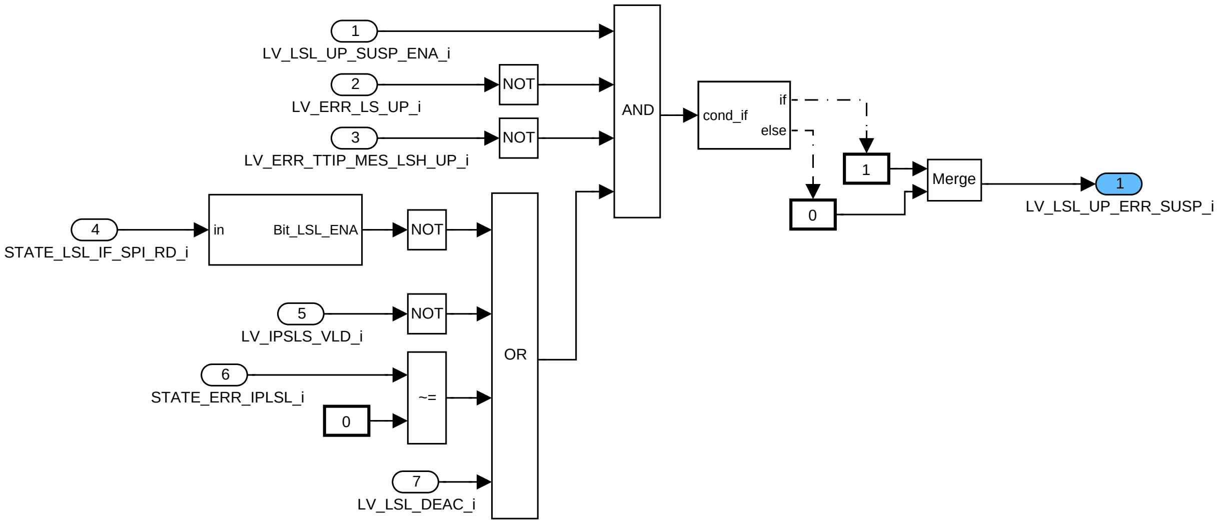


Figure 56 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ FCN_ACT/ CLC_3_SIG_LV/ CLC/ CLC_2_ERR_SUSP

SECTION 3: Initialization due to MAF integral exceeding during PUC (fc_INI_MAF_INT_EXC)

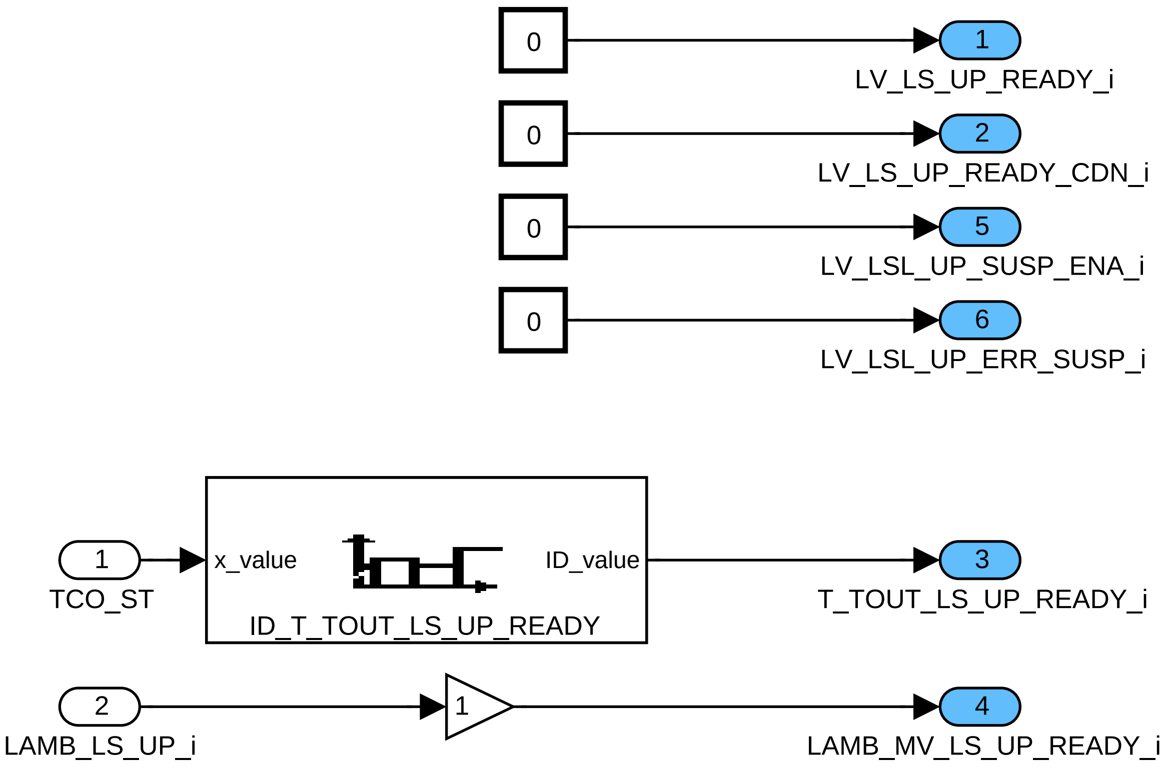


Figure 57 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ INI_MAF_INT_EXC/ INI_MAF_INT_EXC

SECTION 4: Initialization in case calculation not active (fc_INI_VL_PROT)

Initialization due to "Voltage protection" or "off" mode or "not yet left engine start"

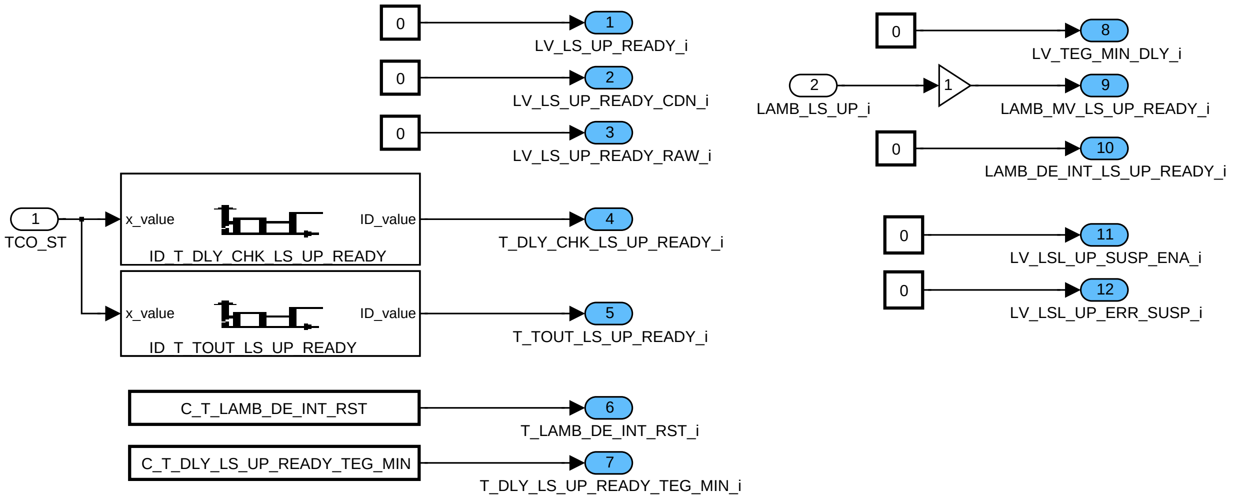


Figure 58 EGCP_DTSYSLSL0/ OPM_20MS/ FLP/ OPM/ CLC_OPER_DET_FOR_LLS/ INI_VL_PROT/ INI_VL_PROT

16.16 WRAF sensor control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ICPLSL_LSL_IF[NC_CBK_EX_NR]	V/O	0...1FH	0...31	1	[-]
Value of the pumping current applied for filling of WRAF sensor oxygen reference area (Icp)					
LV_ICPLSL_ENA_LSL_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag requesting activation / deactivation of reference pumping current (Icp)					
T_ICPLSL_DLY_PWL[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Delay timer for limited Icp supply during ECU power latch					
LV_IPLSL_CTL_ENA_LSL_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag requesting activation / deactivation of pumping current (Ip) controller					
R_IT_IPLSL[NC_CBK_EX_NR]	V	0...FFFFH	0...16383.75	0.25	[ohm]
Calculated value of internal DC resistance of WRAF sensor pump cell					
VPLSL[NC_CBK_EX_NR]	V	80...7FH	-3.224...3.19881	0.0251875	[V]
Calculated voltage across WRAF sensor pump cell (Vp = VIP-VG)					
CTR_ICPLSL[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Counter used to approximate filling & emptying of WRAF sensor pumped reference area					
CTR_ICPLSL_MAX[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
“Delay” for filling & emptying of WRAF sensor pumped reference area					
CTR_IPLSL_CTL_ENA_NR_CYC[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Cycle counter for pulsed activation of pump current					
CTR_IPLSL_CTL_ENA_PWM_ON[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter for duration of “Ip on” phase during pulsed activation of pump current					
CTR_IPLSL_CTL_ENA_PWM_OFF[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter for duration of “Ip off” phase during pulsed activation of pump current					
LV_IPLSL_CTL_ENA_PLS_ACT[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that pulsed activation of pump current is active					
LV_ICPLSL_VLD[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating that WRAF sensor pumped reference area is completely filled					
LV_IPLSL_VLD[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that WRAF sensor pump current is valid					
T_IPLSL_NOT_VLD_VB_L[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Delay timer for possible occurence of increased heater coupling due to permanent battery undervoltage					
T_IPLSL_NOT_VLD_VB_L_THD[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Upper limit of delay timer for possible occurence of increased heater coupling					
LV_IPLSL_NOT_VLD_VB_L[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that increased heater-coupling due to permanent battery undervoltage might be present					
LV_LAMB_LS_UP_VLD[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that lambda signal from WRAF system is reliable and up-to-date					
LV_IPLSL_CTL_INH_VNLSL_LIM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that pump current control is inhibited by active Icp cut-off / Vn overvoltage conditions					
LV_VPLSL_LIM_ACT[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating active pumping voltage (Vp) limitation conditions					
LV_IPLSL_CTL_INH_VPLSL_LIM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that pump current control is inhibited by active active pumping voltage (Vp) limitation conditions					
LAMB_THD_TMP_VPLSL_LIM[NC_CBK_EX_NR]	V	0...7FFFH	0...31.99902	9.77E-04	[-]

Intermediate storage variable of threshold for Vp limitation in terms of lambda					
LAMB_THD_VPLSL_LIM[NC_CBK_EX_NR]	V/O/S	0...7FFFH	0...31.99902	9.77E-04	[-]
Threshold for Vp limitation in terms of lambda					
T_DLY_IPLSL_CTL_ENA[NC_CBK_EX_NR]	V	0...FFH	0...2.55	0.01	[s]
Delay timer for validation of Ip after enabling of Ip control loop, used to allow for signal settling					
T_DLY_VNLSL_LIM_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Timer for limitation of duration of Ip control loop inhibition due to Icp cut-off / Vn overvoltage conditions					
T_IPLSL_CTL_LSH_UP_MAN_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Timer for delay in activation of pump current control upon engine start in case of forced sensor pre-heating					
LV_LAMB_LS_UP_ESTIM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Logical variable indicating limited accuracy of measured lambda signal					

Input data:

LV_ST_END	LAMB_SP[NC_CBK_EX_NR]	R_IT_LS_UP[NC_CBK_EX_NR]	CTR_ERR_LSL_IF_SPI_WR[NC_CBK_EX_NR]
LV_LSL_DEAC[NC_CBK_EX_NR]	STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]	IPLSL_COR[NC_CBK_EX_NR]	LV_LSL_IF_SPI_RST_END[NC_CBK_EX_NR]
LV_LS_UP_READY_CDN[NC_CBK_EX_NR]	LV_LS_UP_READY[NC_CBK_EX_NR]	TTIP_MES_LS_UP[NC_CBK_EX_NR]	STATE_ERR_IPLSL[NC_CBK_EX_NR]
LV_ERR_LS_UP[NC_CBK_EX_NR]	NC_CBK_EX_NR	STATE_LSL_IF_CONF_SPI_RD[NC_CBK_EX_NR]	LV_VLS_OFS_ADJ_ENA_LSL_IF[NC_CBK_EX_NR]
LV_VNLSL_LIM[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]	LV_VPLSL_LIM[NC_CBK_EX_NR]	LV_LS_UP_READY[NC_CBK_EX_NR]
LV_LSL_OFS_ADJ_ACT[NC_CBK_EX_NR]	SAF_KGH	LV_SAP	LV_SAV
VB	LV_IGK	LV_DIAG_OBD_SYM_LSH_UP[NC_CBK_EX_NR]	LV_VLS_UP_VLD[NC_CBK_EX_NR]
LV_VLS_UP_INIT[NC_CBK_EX_NR]	LV_LAMB_STI_OFF_LSL_OFS_ADJ[NC_CBK_EX_NR]	STATE_LSH_UP[NC_CBK_EX_NR]	LV_LSH_UP_MAN_ACT[NC_CBK_EX_NR]
LV_SWI_GAIN_LSL_IF[NC_CBK_EX_NR]	MAF_KGH	N_32	

FUNCTION DESCRIPTION:

General information:

The purpose of this function is the management of supply of pumped reference current (Icp), and pump current (Ip) control for ATIC42 WRAF sensor interface.

The Icp current is required by certain WRAF sensor types to provide a pumped oxygen reference for the Nernst cell sensing portion, as this is not provided by venting air into a chamber within the sensor. The value of the Icp current is programmed according to certain operating states of the WRAF system, and the Icp current is enabled or disabled.

Ip current is a measure of lambda. The Ip controller shall be required to meet stringent sensor protection requirements. For certain critical operating states of WRAF sensor, it may be undesirable to have the Ip control active and hence the function shall enable / disable the Ip control as required.

Both danger of implausible lambda measurement, e.g. due to signal saturation effects, and physical sensor destruction have to be detected and appropriate measures are taken.

Ip control is activated with a repeated pulse routine to avoid excessive pumping voltage (Vp) supply at point of activation. During activation, some of the provided sensor protection mechanisms are inhibited because they would produce inappropriate results due to transient behaviour.

Validity of pumping current signal is jugded depending on current state of WRAF system. Certain operating conditions lead to temporary revocation of validity flag, e.g. danger of increased heater-coupling on Ip signal circuit due to longer term battery undervoltage.

In case of pumping voltage (Vp) across sensor pump cell violating certain bounds or of reverse Icp cut-off, Ip control also has to be inhibited to avoid physical destruction of sensor element, i.e. by blackening effect of zirconia dioxyde ceramic.

Two logical variables stating short term profound reliability of measured lambda signal, and signal accuracy level (regular / limited), resp., shall be provided.

Description:

The function shall be activated once the transition from engine start to engine run has taken place and once the initialisation of the WRAF sensor 'ASW<=>BSW' interface has successfully been carried out. It shall stay active up to the end of ECU power latch.

Alternative activation of certain chapters shall take place in case of forced sensor pre-heating prior to engine start.

The function is split into ten main sections:

1. Activation of pumped reference current (Icp)
2. Value of pumped reference current ("Set Icp programming value")
3. Determination of Icp cut-off conditions ("Overvoltage at sensor Vn pin")
4. Determination of required status of pump current (Ip) control loop
5. Activation of pump current (Ip) control loop
6. Possibility of increased heater-coupling due to permanent battery undervoltage
7. Validity of pump current (Ip)
8. Pumping voltage (Vp) limitation strategy
9. Provision of a cumulative logical variable LV_LAMB_LS_UP_VLD_i for judgement of short term reliability of WRAF system lambda output signal.
10. Indication of regular / limited accuracy of measurement signal.

Remark: Below description of functional units within 'WRAF sensor control' module aims to provide a general survey on physical backgrounds of sensor behaviour and derived actions for operational strategy. For related **variable names** please respect corresponding diagrams provided within **formula section**.

1. Activation of pumped reference current

This section shall provide the 'Icp enable' and appropriate programming value information to the WRAF sensor interface and indicate whether the pumped reference has been filled via corresponding validity flag.

Should the sensor type being used not have a pumped oxygen reference or a general WRAF sensor error has occured, then this section shall be disabled, i.e. status flag for Icp validity is revoked, Icp is programmed to its minimum value, and Icp supply is deactived.

If none of these two conditions is present, this section behaves as follows:

During pulsed Ip control activation only programming value of Icp is evaluated and no other change in Icp supply conditions is performed.

In case of permanent operation of Ip loop with no SPI fault present, sufficiently high battery voltage for accurate operation of ATIC42, and no Vn overvoltage in conjuction with under-run of rich lambda setpoint threshold present (occurence is limited to rich atmosphere, forced Icp cut-off is automatically performed by ATIC42 HW), routine waits for successful mirroring of latest Icp status and programming value commands sent via SPI bus.

Once command transfer has been successful, Icp flow is requested, sensor element temperature has exceeded calibratible threshold, counter limit for filling time of pumped reference is determined. Latter condition can be replaced by OBD2 heater fault present in conjunction with operative sensor readiness (-> tip temperature not reliable), or in case of conditional operative readiness (-> forced sensor activation).

If repeated evaluation of corresponding index table (which depends on sensed element temperature and Icp programming value) leads to a higher value than adjusted before and pumped reference has not been filled completely yet, counter limit is increased. To force initial determination of counter limit this action is also performed when entering the condition with a filling counter value of zero.

Once Icp counter limit is reached the Icp valid flag is set, otherwise counter for Icp filling status is incremented.

General Icp activation is treated depending on engine condition (Icp active with engine running), status of forced sensor heating, and value of power latch delay timer. Latter provides for extended Icp supply after engine stop.

Next Subroutine “Value of pumped reference current” is executed to determine appropriate Icp programming value.

Should any of the following conditions hold true, filling status of pumped reference has to be verified: SPI command transfer fault, low battery voltage, Vn overvoltage in conjunction with under-run of rich lambda setpoint threshold, revocation of Icp activation, tip temperature drop below threshold.

If pumped reference area has been judged to be completely filled before, Icp counter limit is set to calibratable threshold which represents delay for emptying prior to revocation of 'Icp valid' status bit - to take into account for transient disturbances without immedate reset. Revocation takes place once counter has been decremented to zero.

In case of forced sensor pre-heating prior to engine start, simplified filling procedure of pumped reference is applied which bypasses certain conditions.

2. Value of pumped reference current

Icp value to be programmed is determined for four different operating states taking into account dependency of provided reference pumping current on (estimated) exhaust lambda and on sensor element temperature:

- Power Latch phase (engine stalled and forced sensor pre-heating inactive): Exhaust system filled with air; Ip control loop disabled. Icp supply is limited to calibratable delay time.
- Ip control active or pulsed activation thereof: Exhaust lambda provided by WRAF sensor.

- Ip control inactive, cold sensor: Exhaust lambda derived from <u>combustion</u> lambda setpoint (LAMB_SP[i]) and additional secondary air flow (SAF_KGH). For calibration it has to be taken into account that LAMB_SP[i] does not represent composition of exhaust atmosphere present at linear oxygen sensor during activation of secondary air; under these conditions lean conditions prevail at sensor mounting location regardless of LAMB_SP[i]!
- Ip control inactive, warm sensor, or OBD2 heater fault present in conjunction with operative sensor readiness (-> tip temperature not reliable), or in case of conditional operative readiness (-> forced sensor activation): same as above.

The influence of secondary air on exhaust lambda at sensor mounting position is taken into account for disabled Ip control loop.

Current status of Ip control loop is judged by evaluation of corresponding ATIC42 interface bit.

Distinction between “cold” and “warm” sensor is made by evaluation of calibratable ceramic temperature threshold.

As long as Ip control is in disabled state, Electromagnetic force (EMF) across sensor nernst cell adjusts itself in a range between 0 ... 1V depending on surrounding exhaust gas atmosphere – comparable to the behaviour of a binary type oxygen sensor. With enabled loop, EMF is controlled to be approx. 450mV constantly.

As a rule of thumb for conversion of Icp programming value to actual value of Icp current provided by ATIC42 (which is split between sensor nernst cell and shunt resistance within ATIC42 module which serves as voltage limitation for cold sensor) the following formula can be applied:

Icp = (k + 1) * 5uA with k: Icp programming value.

3. Determination of Icp cut-off conditions

Under worst case tolerances and rich conditions, the potential at sensor pin VN may exceed the threshold up to which Icp generation can be guaranteed for sensors of pumped reference type. Under such circumstances the Icp may temporarily be reduced or even cut-off by ATIC42 protection mechanism. This may lead to lean Ip misindication and also to sensor damage.

ATIC42 2nd silicon provides for automatic Icp cut-off in case of Vn overvoltage and signals the presence of such a condition through its interface.

Hence when Icp cut-off is signaled to occur by ATIC42, Ip control shall be disabled reducing the potential Vp across the pump cell and thereby also reducing the potential at pin Vn.

The voltage Vp across the pump cell shall be modelled by multiplying the measured pump current Ip with the pump cell internal DC resistance obtained from the relationship between the internal resistance of the Nernst cell and the pump cell.

The pump cell DC internal resistance (Rip) shall be obtained from a mapping dependent on value of nernst cell inner resistance (Ri, representing sensor temperature), and lambda setpoint. Both internal resistances are linked by sensor temperature and design. Additionally the pump cell resistance is dependent on the sensor lambda environment, hence the lambda setpoint dependence. Preliminary Rip value is then multiplied by a factor and a constant is added. Both parameters permit the dependency of the resistances to be modified to take worst case ageing into account.

The pump cell voltage Vp shall be calculated by multiplying modelled Rip by corrected value of pump current Ip.

Providing that WRAF sensor is of pumped reference type, the function shall determine whether Icp cut-off conditions are already active. If true, then the condition shall only be deemed to no longer persist when Rip is less than the value stored once Icp cut-off conditions were determined (i.e. sensor elememt got hotter), or lambda setpoint is greater than (i.e. leaner) than the value stored once Icp cut-off conditions were determined. Or, if maximum duration of active Vp limitation is calibrated to be non-zero, when the internal timer TIMER_2 has expired. In addition, for all three cases, VN overvoltage shall no longer be indicated by ATIC42. If all this holds true, then TIMER_2 shall be stopped, the stored Rip and lambda setpoint values shall be reset and the Ip control inhibition flag revoked. Otherwise TIMER_2 is decremented.

If the Icp cut-off conditions are inactive, and pulsed Ip control activation has been completed, calculated Vp voltage shall be compared to a minimum threshold. If this ‘rich’ threshold is exceeded or Vn overvoltage is sensed by ATIC42 and rich environment is present at sensor (i.e. rich lambda setpoint threshold is under-run), Icp cut-off conditions may occur. Should they be present for longer than estimated emptying delay of pumped reference, TIMER_2 shall be initialized, the current Rip and lambda setpoint values shall be stored, and Ip control inhibition flag shall be set.

4. Determination of required status of pump current (Ip) control loop

This section shall collect the various requests for activation / deactivation of the Ip control loop and provide the information to the WRAF sensor interface.

The function shall provide for manual switching of the Ip control loop by means of logical constants. If manual activation is requested, subroutine “Activation of pump current control loop” is executed.

In case of automatic switching, no general WRAF sensor error present and running engine (or with activated forced sensor pre-heating), the function checks if pumped reference of a sensor requiring Icp is not completely filled yet, or if Icp cut-off or Vp limitation sensor protection routine is active (and forced sensor pre-heating is inactive). Either one of these conditions leads to Ip passivation by resetting corresponding ATIC42 interface bit. At the same time all variables needed for next “Activation of pump current control loop” are initialized.

Next the function shall determine whether the conditional operative readiness is set or pulsed Ip control activation is running. If so then the evaluation of further conditions related to sensor readiness and tolerated lambda setpoint range for Ip activation shall be skipped and “Activation of pump current control loop” shall be executed.

If former condtions are not present, then the function shall determine whether operative readiness is set (or forced sensor pre-heating is activated). If not, then the Ip control loop shall be deactivated.

Once oxygen sensor ceramic temperature is greater or equal to a calibratable threshold, Ip loop shall be enabled once lambda setopint is located within calibratable bounds, or secondary air is active (providing for lean exhaust gas atmosphere), and Ip offset adjustment is not pending. The same action is taken once OBD2 heater fault ist present (-> tip temperature not reliable). Otherwise Ip loop shall stay disabled.

Thus pumping voltage saturation because of high pumping current requirements for λ << >> 1 can be avoided at point of Ip control loop activation. During secondary air phase, output signal of WRAF sensor is used for monitoring purposes. Thus Ip loop has to be activated even at very lean surrounding atmosphere which is present at that time. In general pumping voltage saturation in lean direction is not critical for WRAF sensor operation in conjunction with ATIC42 as it is for rich atmosphere.

5. Activation of pump current (Ip) control

In order to avoid excessive supply of pumping voltage Vp to WRAF sensor during switch-on of Ip control loop (which might lead to sensor destruction due to blackening effect for applied ‘rich’ voltages Vp < 2V), a pulsed activation with PWM-behaviour is provided.

For a calibratable number of cycles, ON- and OFF- phases of variable length are applied to WRAF sensor interface. Calibration can also be used to skip sensor protection mechanisms during one-time Ip switch-on for a certain time. In this case OFF-phase is of zero length.

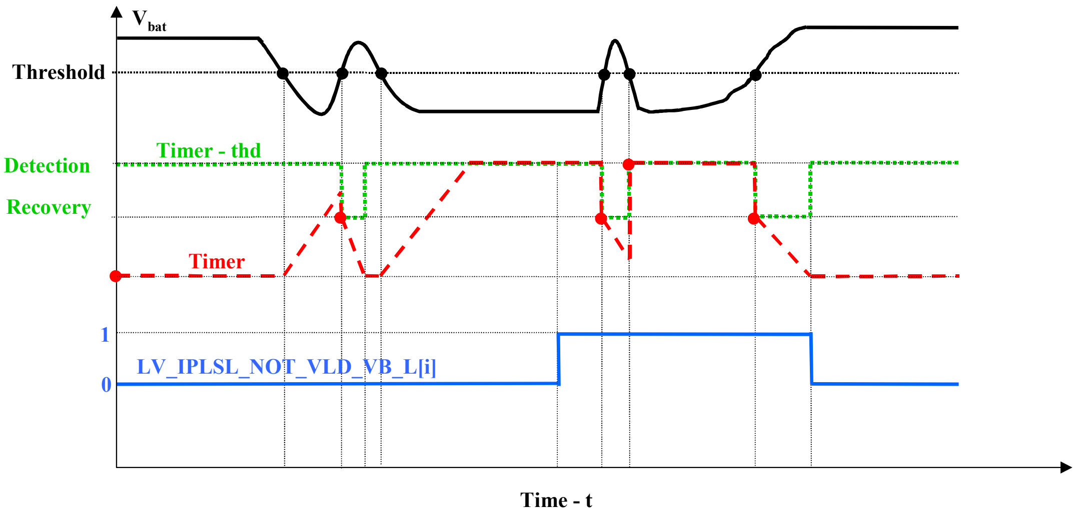
6. Indication of possibility for increased heater-coupling due to permanent battery undervoltage

For the linear oxygen sensor BOSCH LSU4.9, up to and including Äp3.1, there is a danger that for longer term battery undervoltage in combination with maximum heater PWM, increased heater-coupling to the pumping current signal circuit of the sensor might appear.

This effect is related to ion migration within the sensor element due to long term application of a certain ion activation potential to negative sensor heater electrode which is of ‘active low’ type.

Once regular battery voltage conditions are restored the exaggerated heater coupling vanishes within several seconds and returns to regular values.

A timer based detection strategy for possibility of increased heater coupling has been developed and is depticted in below figure.



Once the critical battery undervoltage condition is met for a certain delay time after leaving engine start - usually also leading to high PWM values – pumping current validity shall be revoked.

Since detection and recovery might be linked to different delay times, timer threshold is switched depending on presence of battery undervoltage condition.

If recovery delay time is interrupted by repeated undervoltage condition the delay timer is set to and kept at detection threshold value.

Should time based thresholds for detection and recovery be calibrated identically, latter ist incremented for one step to avoid ambiguities within algorithm execution.

7. Validity of pump current (Ip)

This section shall model the basic validity of the Ip current. For short term reliability of ATIC42 lambda output signal a special indicator bit is introduced in chapter 9.

The function shall check if a general WRAF sensor error is present or if a SPI error has occurred since the last function recurrence. Latter information is retrieved by comparison of corresponding error counter with its previously stored value. If the values differ from each other then current value shall be stored for next comparison, the internal timer TIMER_1 and the Ip validity flag shall be reset.

If sufficient battery voltage for power supply of ATIC42 is available, function shall wait for the sent Ip control enable / disable information to be mirrored by comparing to ATIC42 SPI response byte.

In case of forced sensor pre-heating additional delay time **after engine start** can be calibrated. If corresponding timer threshold is calibrated to be zero, validity of pump current shall also be detectable prior to engine start.

Next the function checks if pulsed activation of pump current control loop is active. If so, Ip validity is revoked.

If not, the function shall determine whether the Ip control loop has been requested to be active. In addition the danger of increased heater coupling due to permanent battery undervoltage has to be excluded.

If all validation conditions hold true, a delay shall be exceeded prior to setting the Ip validity flag. If false, then the internal timer TIMER_1 and the Ip validity flag shall be reset.

8. Pumping voltage (Vp) limitation strategy

Since it is crucial for pumping voltage (Vp) across WRAF sensor pump cell not to exceed certain bounds depending on sensor temperature and ehxaust lambda (roughly $|Vp| \leq 2V$ for sensor at operating temperature, less during sensor warm-up), compliance with these limits has to be ensured under all operating conditions. Exceeding these limits may lead to physical sensor destruction by “blackening” of zirconia sensor element especially under rich conditions.

Sensor ageing (increase in pump cell inner resistance), insufficient sensor operating temperature, or high pressure conditions, resp., may lead to restriction of measurable lambda range within permitted pumping voltage range.

Since physical Vp threshold for WRAF sensors can be extended to Vp > 2V in lean area, HW based analog Vp limitation provided by ATIC42 offers sufficient protection. For rich area, evaluation of digital limitation threshold LV_LSL_VP_LIM_i enables strict limitation of Vp to fixed threshold of Vp = -2V also under dynamic conditions by means of SW intervention on status of Ip control loop.

Gradual increase in limitation of operational lambda range in rich environment due to long term effects shall lead to adaptation of corresponding threshold which is stored in non-volatile ECU memory.

Once digital Vp limitation is indicated by ATIC42, and rich lambda setpoint limitation threshold is under-run, and secondary air / pulsed Ip control activation both are not active, current measured lambda value is temporarily stored as new threshold for reactivation of Ip control. Since digital limitation might already limit pumping current, measured lambda value is equal or greater actual lambda which provides additional security margin for reactivation of Ip control loop.

Should Lambda signal be indicated not to be reliable at that time, current lambda setpoint value is adopted as new temporary threshold.

To guarantee minimum operational range in rich direction, e.g. for Ip activation during sensor warm-up, a calibratable maximum value for Vp limitation lambda threshold has to be respected in addition.

Vp limitation threshold is assigned with either latter value or previously determined temporary threshold.

Upon storage of Vp limitation threshold, Ip control is inhibited by setting corresponding flag and Vp limitation status bit is set thus preventing second evaluation of limitation threshold during current Vp limitation phase.

Should Vp limitation not be indicated any longer, but Vp limitation still be active, lambda setpoint is crosschecked with actual Vp limitation threshold. Once setpoint value exceeds latter, Vp limitation status is reset and inhibition of Ip control loop is revoked.

With Vp limitation inactive and reliable lambda value exceeding limitation threshold in rich direction an 'improvement' in WRAF sensor behaviour is detected (due to sensor change or vanishing of temporary effects which led to an increase in Vp requirements, like e.g. elevated exhaust pressure or decreased sensor temperature) and Vp limitation threshold is relaxed.

9. Logical variable for judgement of WRAF sensor lambda signal reliability

Various conditions for operative readiness of WRAF system, consisting of WRAF sensor (including heater) & WRAF controller, have to be fullfilled and no diagnostic contraindications are allowed to be present to judge WRAF lambda signal as being *completely* reliable (for details see formula section). Evaluation of this variable has to take place at start of Vp limitation check.

10. Logical variable for judgement of lambda sensor signal accuracy level

Certain environmental conditions or functional limitations might lead to decrease in signal accuracy level (regular -> limited) due to need for additional compensative measures which are subject to tolerances, or limited functional performance.

Presence of any of these conditions is indicated by cumulative flag.

Application conditions:

Initialisation:

The following initialisation shall take place for each newly-flashed ECU or upon detection of EEPROM error:

LAMB_THD_VPLSL_LIM[i] = 0

The following initialisation shall take place directly after RESET:

ICPLSL_LSL_IF[i] = 0

T_IPLSL_NOT_VLD_VB_L_THD[i] = C_T_IPLSL_NOT_VLD_VB_L_INC

T_IPLSL_NOT_VLD_VB_L[i] = 0

The following initialisation shall take place after ECU RESET, at LV_IGK = 0->1, or upon erase of error management:

R_IT_IPLSL[i] = 16383.75 (i.e. FFFFH)

VPLSL[i] = 0

CTR_ICPLSL[i] = 0

CTR_ICPLSL_MAX[i] = 0

T_ICPLSL_DLY_PWL[i] = 0

LV_ICPLSL_VLD[i] = 0

LV_IPLSL_VLD[i] = 0

LV_IPLSL_CTL_ENA_PLS_ACT[i] = 0

CTR_IPLSL_CTL_ENA_NR_CYC[i] = 0

CTR_IPLSL_CTL_ENA_PWM_ON[i] = 0

CTR_IPLSL_CTL_ENA_PWM_OFF[i] = 0

LV_IPLSL_CTL_INH_VNLSL_LIM[i] = 0

LV_VPLSL_LIM_ACT[i] = 0

LAMB_THD_TMP_VPLSL_LIM[i] = 0

LV_IPLSL_CTL_INH_VPLSL_LIM[i] = 0

LV_LAMB_LS_UP_VLD[i] = 0

LV_IPLSL_NOT_VLD_VB_L[i] = 0

T_IPLSL_CTL_LSH_UP_MAN_ACT[i] = C_T_IPLSL_CTL_LSH_UP_MAN_ACT

LV_LAMB_LS_UP_ESTIM[i] = 1

R_IT_IPLSL[i] ***Store*** = 16383.75 (i.e. FFFFH)

LAMB_SP[i] ***Store*** = 0

Reset internal timers T_DLY_IPLSL_CTL_ENA[i] & T_DLY_VPLSL_LIM_ACT [i]

Recurrence:

T_SAMPLE = 10ms.

Activation:

For chapters 3. / 6. / 8. / 9.: Transition from engine start to run and SPI initialisation complete.

LV_LSL_IF_SPI_RST_END[i] == 1
and LV_ST_END == 1

For chapters 1. / 2. / 4. / 5. / 7. **/ 10.**: Extended supply of pumped reference current and activation of Ip control loop prior to engine start in case of forced sensor pre-heating.

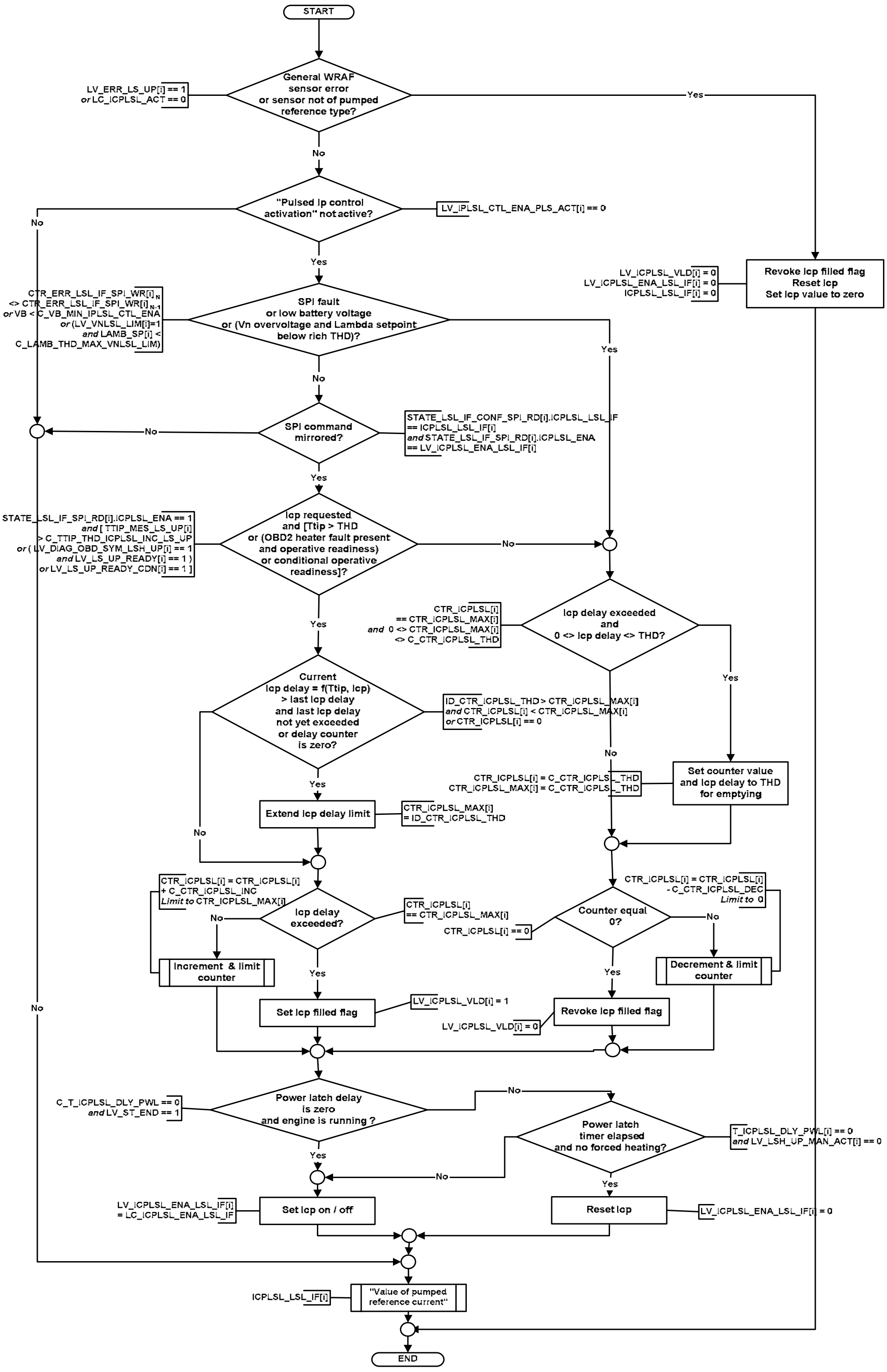
LV_LSL_IF_SPI_RST_END[i] == 1
and **(** LV_ST_END == 1
or LV_LSH_UP_MAN_ACT[i] == 1 **)**

Deactivation:

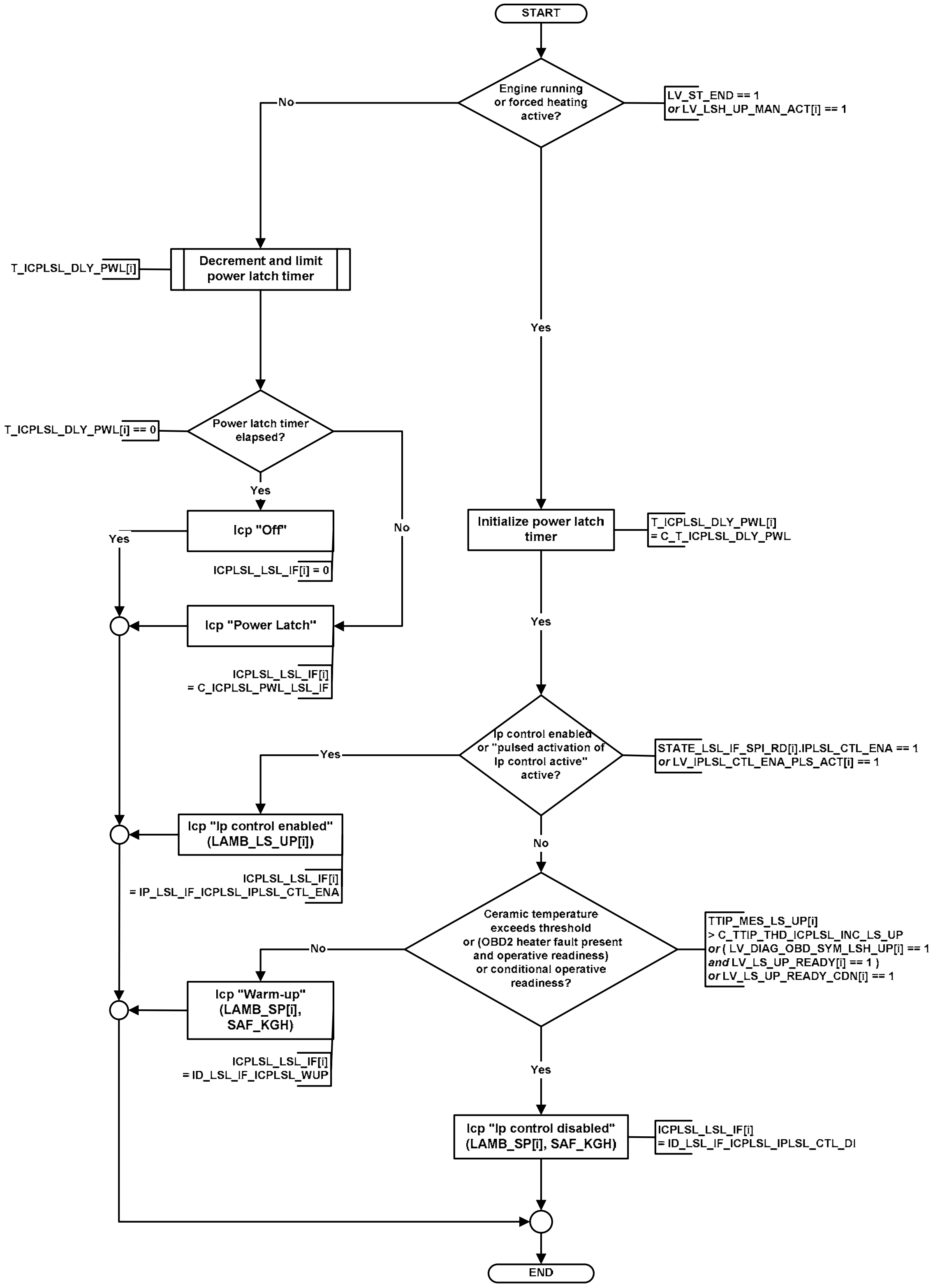
At the end of ECU power latch.

Formula section:

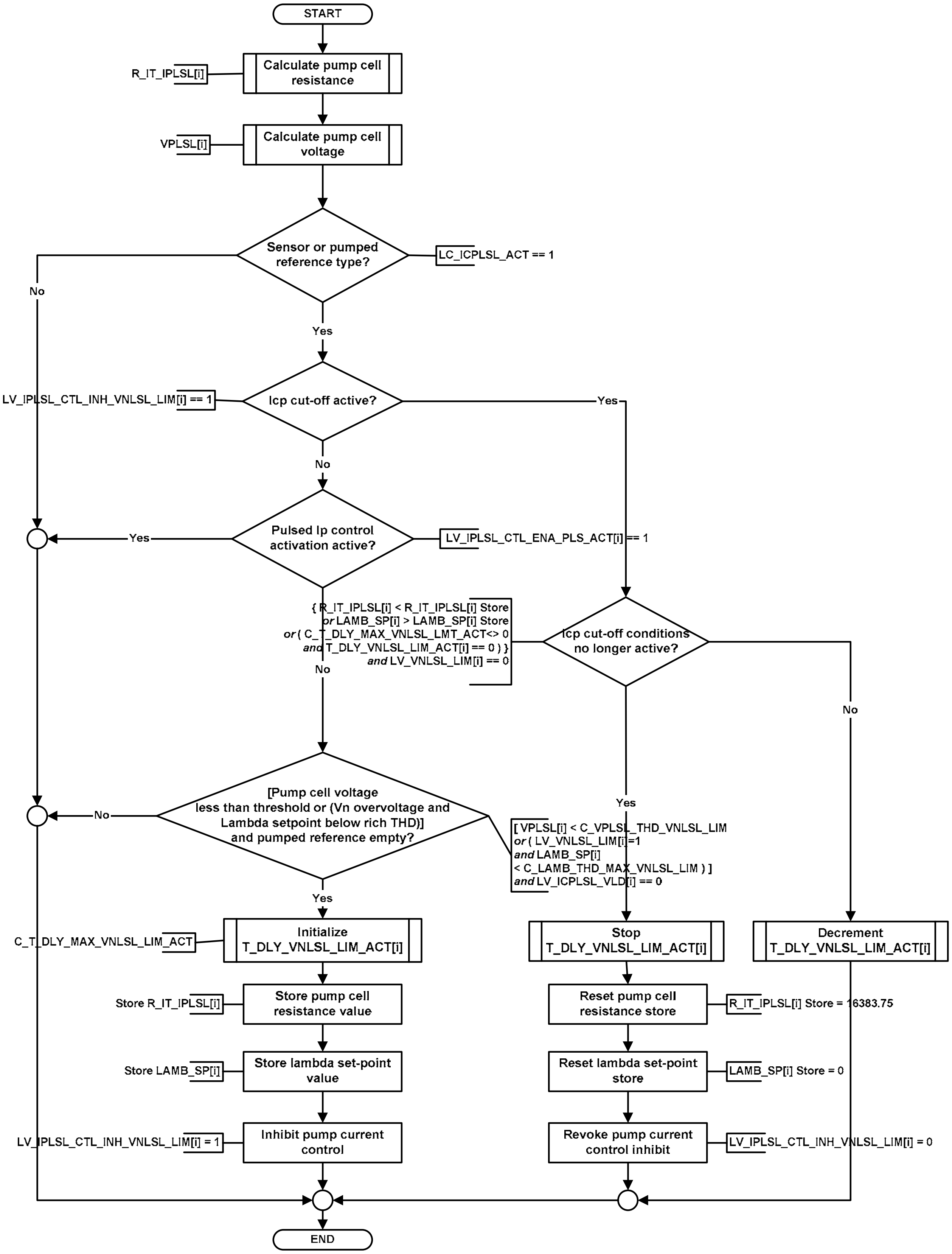
1. Activation of pumped reference current



2. Value of pumped reference current



3. Determination of Icp cut-off conditions



Calculation of modelled WRAF sensor pump cell DC resistance

See determination of Icp cut-off conditions above for use.

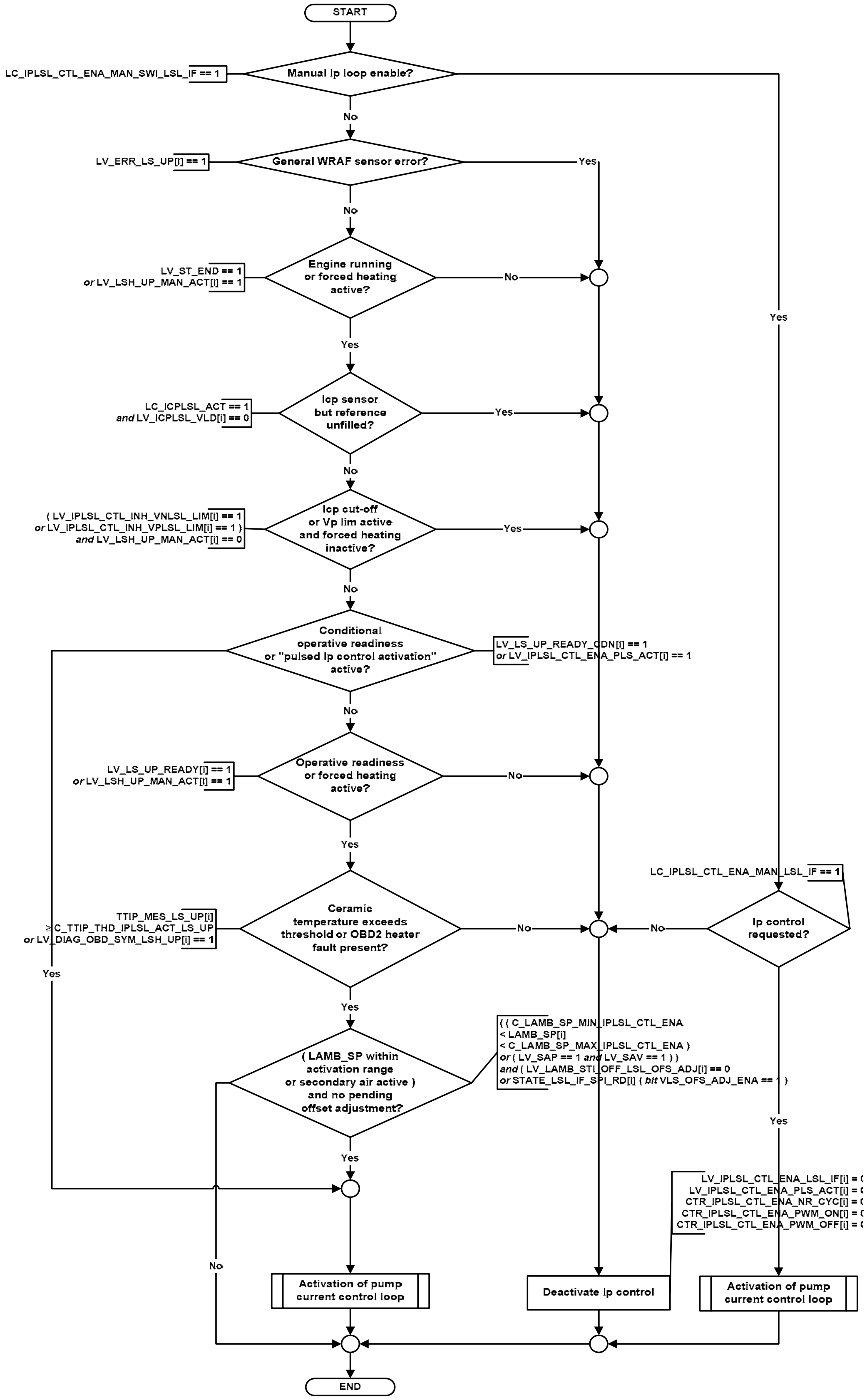
R_IT_IPLSL[i] = (IP_R_IT_IPLSL (R_IT_LS_UP[i], LAMB_SP[i]) * C_R_IT_IPLSL_AGI_FAC) + C_R_IT_IPLSL_AGI_ADD

Calculation of modelled voltage across WRAF sensor pump cell

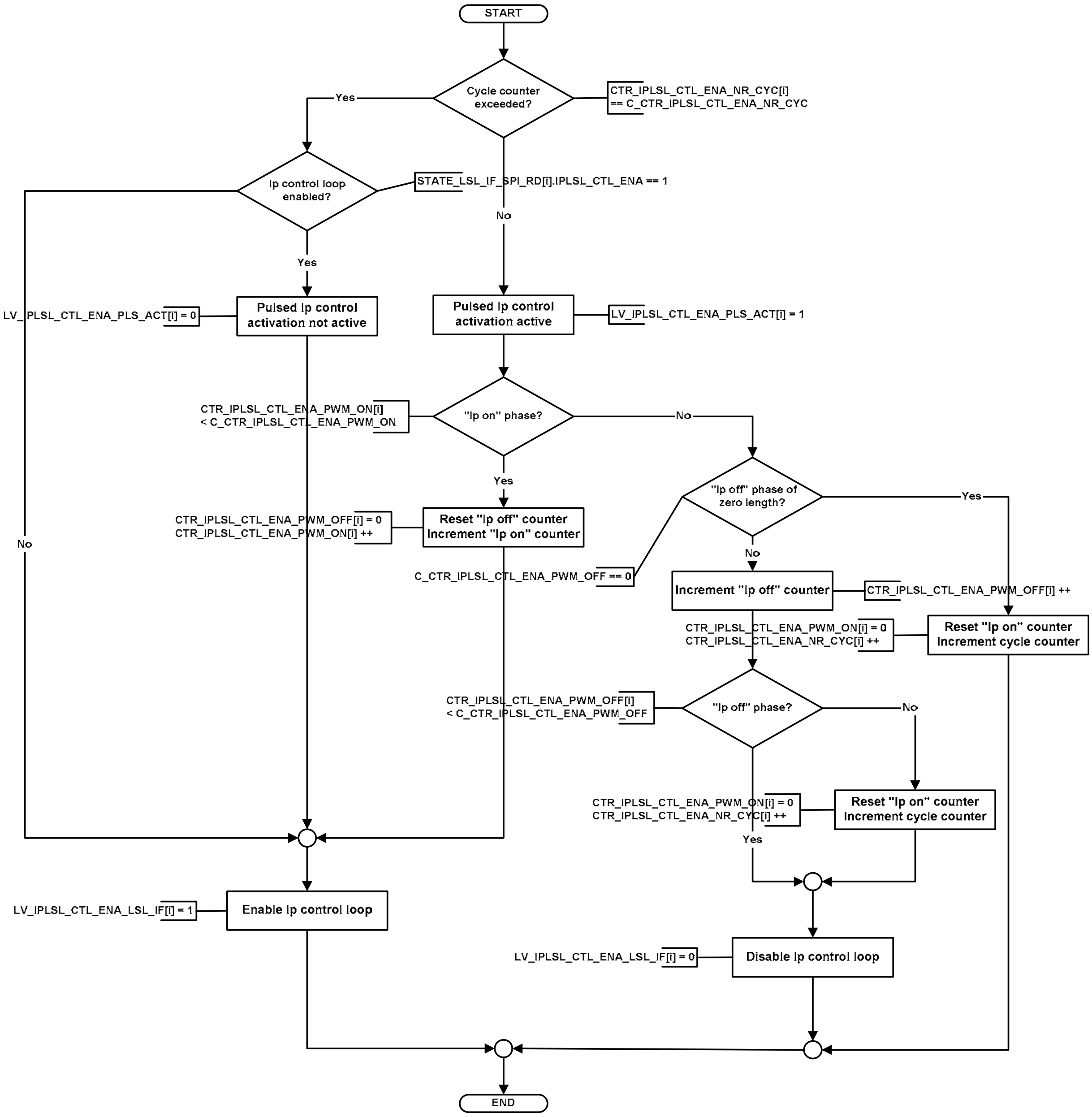
See determination of Icp cut-off conditions above for use.

VPLSL[i] = R_IT_IPLSL[i] * IPLSL_COR[i]

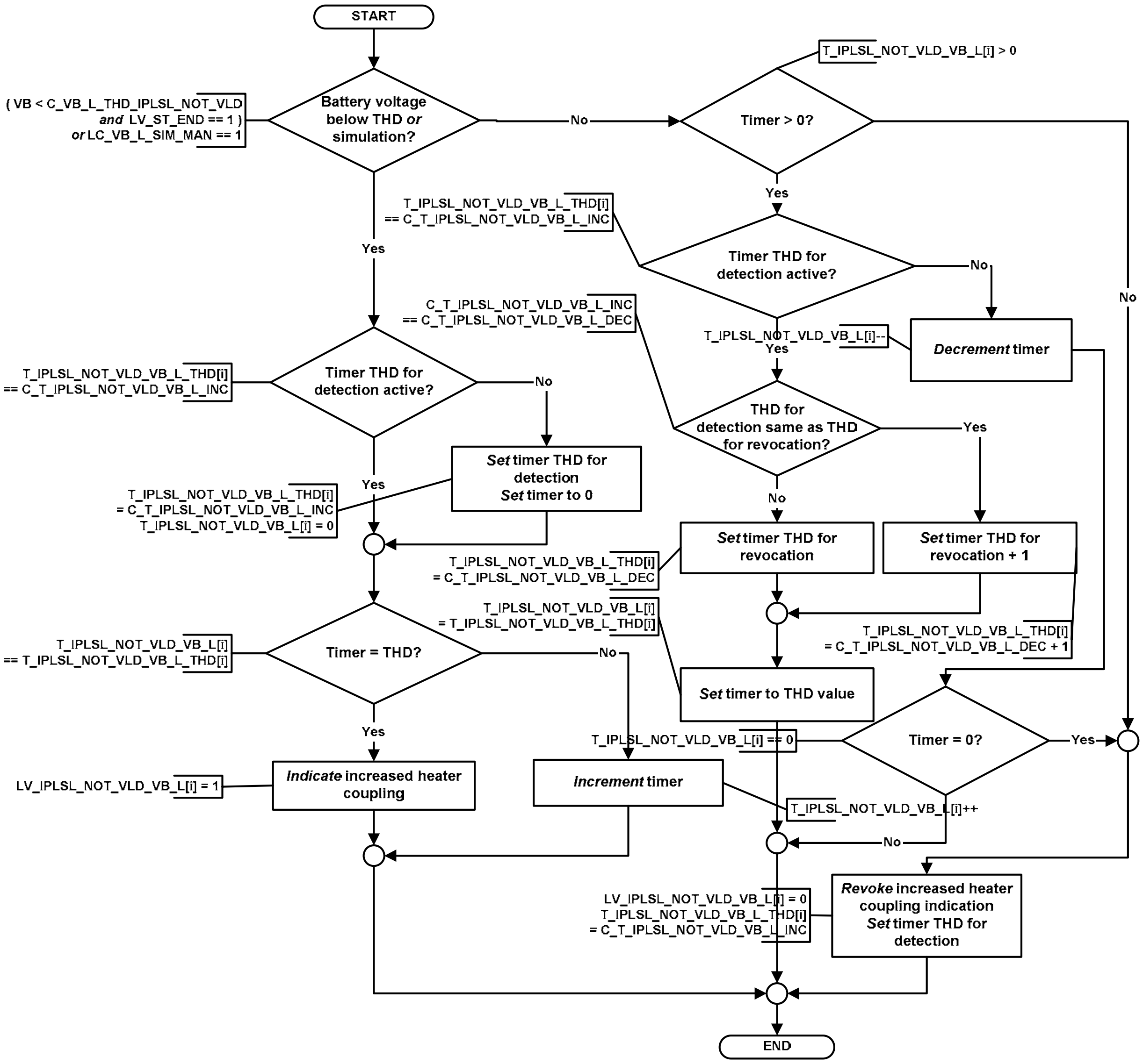
4. Determine required status of pump current (Ip) control loop



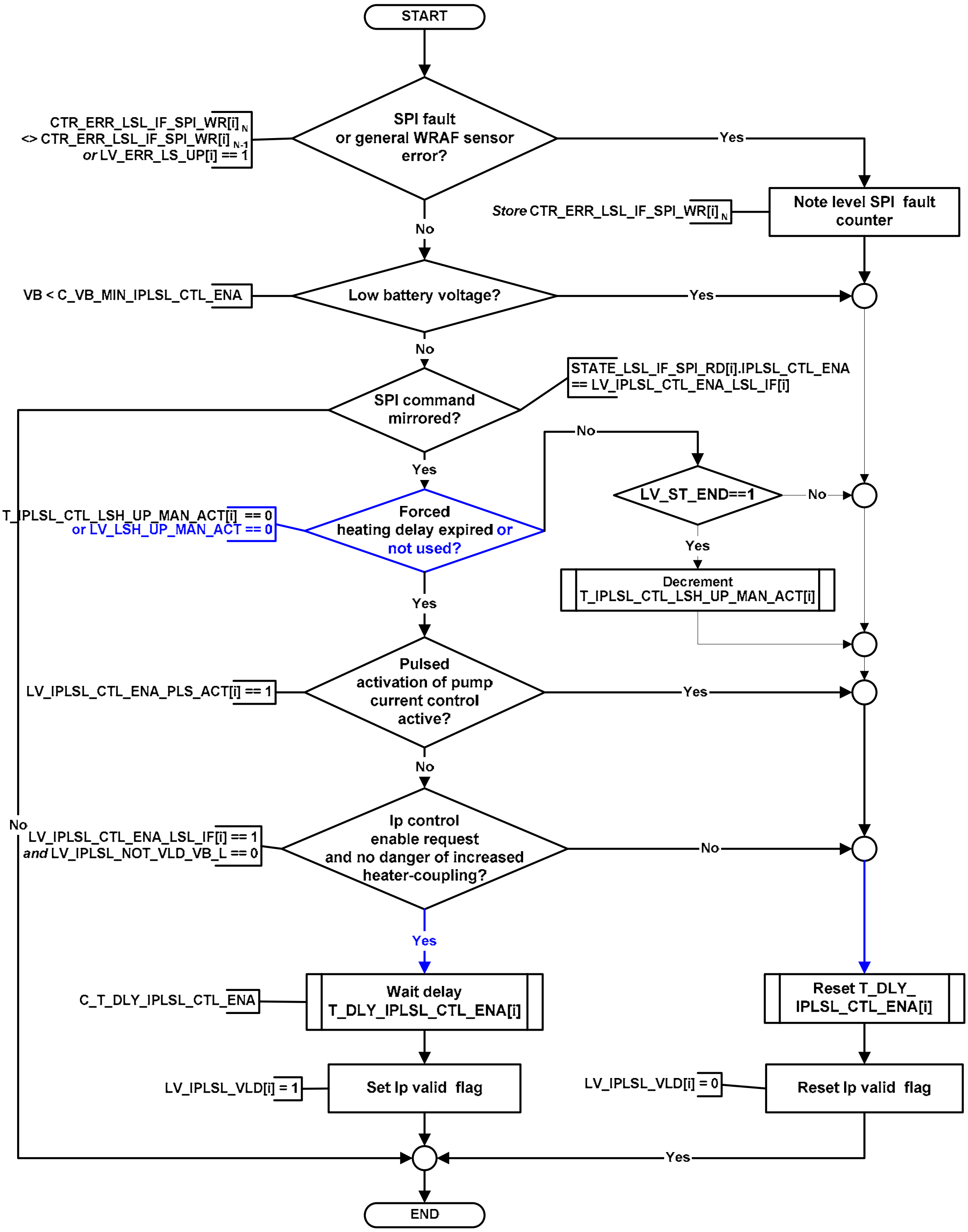
5. Activation of pump current (Ip) control loop



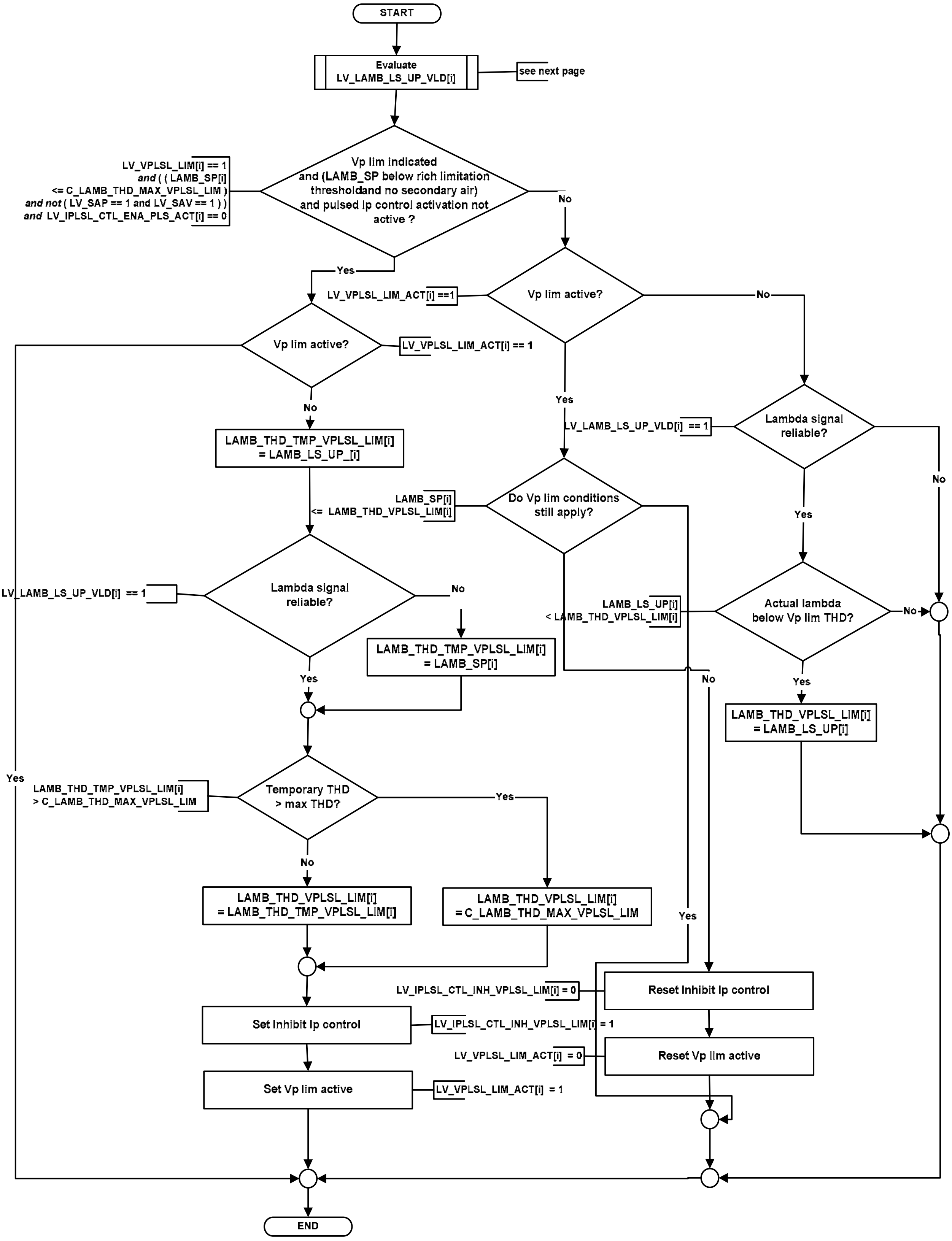
6. Indication of possibility for increased heater-coupling due to permanent battery undervoltage



7. Validity of pump current (Ip)



8. Pumping voltage (Vp) limitation strategy



9. Logical variable for judgement of short term WRAF sensor lambda signal reliability

```
If      LV_LS_UP_READY[i] == 1
        /* WRAF sensor full operative readiness detected */
and     LV_LSL_OFS_ADJ_ACT[i] == 0
and     LV_VLS_OFS_ADJ_ENA_LSL_IF[i] == 0
        /* WRAF controller electrical offset adjustment not active */
and     STATE_LSL_IF_SPI_RD[i] (bit LSL_ENA) == 1
        /* WRAF sensor coupled */
and     ( LV_ICPLSL_VLD[i] == 1 or LC_ICPLSL_ACT == 0 )
        /* Reference pumping current valid or sensor not of pumped reference type */
and     LV_IPLSL_VLD[i] == 1
        /* WRAF sensor pumping current valid */
and     LV_VLS_UP_VLD[i] == 1
        /* No clipping of ATIC42 output voltage present */
and     LV_VLS_UP_INIT[i] == 0
and     LV_SWI_GAIN_LSL_IF[i] == STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN )
        /* Initialisation of VLS_UP[i] has successfully been performed
          upon Ip gain-switching */
and     STATE_ERR_IPLSL[i] == 0
        /* Pumping current within allowed bounds */
and     LV_LSL_DEAC[i] == 0
        /* No OBD1 signal fault pending */
and     LV_ERR_LS_UP[i] == 0
        /* No WRAF system error present */
then    LV_LAMB_LS_UP_VLD[i] = 1
        /* WRAF system lambda signal reliable */
else    LV_LAMB_LS_UP_VLD[i] = 0
        /* WRAF system lambda signal NOT reliable */
endif
```

10. Logical variable for judgement of lambda sensor signal accuracy level

```
If      LV_IPLSL_VLD[i] == 0
        /* WRAF sensor pump current is valid in general */
or      TTIP_MES_LS_UP[i] < C_TTIP_MIN_LAMB_LS_UP_ESTIM
or      MAF_KGH < C_MAF_KGH_MIN_LAMB_LS_UP_ESTIM
or      N_32 > C_N_32_MAX_LAMB_LS_UP_ESTIM
then    LV_LAMB_LS_UP_ESTIM[i] = 1
        /* Reduced accuracy of linear lambda signal */
else    LV_LAMB_LS_UP_ESTIM[i] = 0
        /* No constraint on general pump current validity */
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_R_IT_IPLSL	8*8	0...FFFFH	0...16383.75	0.25	[ohm]
LDP_R_IT_LS_UP_IP_R_IT_IPLSL	8	0...FFFFH	0...16383.75	0.25	[ohm]
LDP_LAMB_SP_IP_R_IT_IPLSL	8	0...7FFFH	0...31.99902	9.77E-04	[-]
Internal DC resistance of WRAF sensor pump cell					
ID_CTR_ICPLSL_THD	3*3	0...FFFFH	0...65535	1	[-]
LDP_TTIP_LS_IP_CTR_ICPLSL_THD	3	0...FFFFH	-2048...2047.9375	0.0625	[°C]
LDP_ICPLSL_IP_CTR_ICPLSL_THD	3	0...1FH	0...31	1	[-]
Variable threshold of counter used to approximate filling & emptying of WRAF sensor pumped reference area					
IP_LSL_IF_ICPLSL_IPLSL_CTL_ENA	3	0...1FH	0...31	1	[-]
LDP_LAMB_IP_ICPLSL_IPLSL_ENA	3	0...7FFFH	0...31.99902	9.77E-04	[-]
Configuration of current to pump reference air (Icp) with Ip control enabled					
ID_LSL_IF_ICPLSL_IPLSL_CTL_DI	3*3	0...1FH	0...31	1	[-]
LDP_LAMB_SP_IP_ICPLSL_IPLSL_DI	3	0...7FFFH	0...31.99902	9.77E-04	[-]
LDP_SAF_KGH_IP_ICPLSL_IPLSL_DI	3	0...FFFFH	0...1023.98437	0.015625	[kg/h]
Configuration of current to pump reference air (Icp) with Ip control disabled and ‘warm’ WRAF sensor					
ID_LSL_IF_ICPLSL_WUP	3*3	0...1FH	0...31	1	[-]
LDP_LAMB_SP_IP_ICPLSL_WUP	3	0...7FFFH	0...31.99902	9.77E-04	[-]
LDP_SAF_KGH_IP_ICPLSL_WUP	3	0...FFFFH	0...1023.98437	0.015625	[kg/h]
Configuration of current to pump reference air (Icp) with Ip control disabled and ‘cold’ WRAF sensor					
C_R_IT_IPLSL_AGI_ADD	1	0...FFFFH	0...16383.75	0.25	[ohm]
Additive ageing constant for WRAF sensor internal pump cell DC resistance					
C_R_IT_IPLSL_AGI_FAC	1	0...FFH	0...7.96875	0.03125	[-]
Multiplicative ageing factor for WRAF sensor internal pump cell DC resistance					
C_ICPLSL_PWL_LSL_IF	1	0...1FH	0...31	1	[-]
Configuration of current to pump reference air (Icp) during power latch phase, or forced sensor pre-heating, resp.					
C_CTR_ICPLSL_INC	1	0...FFH	0...255	1	[-]
Increment of counter used to approximate filling of WRAF sensor oxygen reference area					
C_CTR_ICPLSL_DEC	1	0...FFH	0...255	1	[-]
Decrement of counter used to approximate emptying of WRAF sensor oxygen reference area					
C_CTR_ICPLSL_THD	1	0...FFFFH	0...65535	1	[-]
Threshold of counter used to approximate filling state of WRAF sensor oxygen reference area during discharge					
C_T_ICPLSL_DLY_PWL	1	0...FFFFH	0...655.35	0.01	[s]
Delay time for limited Icp supply during ECU power latch					
C_T_IPLSL_NOT_VLD_VB_L_INC	1	0...FFFFH	0...655.35	0.01	[s]
Delay time for possible occurence of increased heater coupling due to permanent battery undervoltage					
C_T_IPLSL_NOT_VLD_VB_L_DEC	1	0...FFFFH	0...655.35	0.01	[s]
Delay time for revocation of indicated increased heater coupling					
C_VB_L_THD_IPLSL_NOT_VLD	1	0...FFH	0...25.89843	0.1015625	[V]
Battery undervoltage threshold for possible increase of heater-coupling					
LC_VB_L_SIM_MAN	1	0...1H	0...1	1	[-]
Switch for simulation of battery undervoltage which can be used for function testing					
C_CTR_IPLSL_CTL_ENA_NR_CYC	1	0...FFH	0...255	1	[-]
Number of cycles for pulsed activation of pump current					
C_CTR_IPLSL_CTL_ENA_PWM_ON	1	0...FFH	0...255	1	[-]
Threshold for duration of “Ip on” phase during pulsed activation of pump current					
C_CTR_IPLSL_CTL_ENA_PWM_OFF	1	0...FFH	0...255	1	[-]
Threshold for duration of “Ip off” phase during pulsed activation of pump current					
C_VPLSL_THD_VNLSL_LIM	1	80...7FH	-3.224...3.19881	0.0251875	[V]
Pumping voltage (Vp) threshold below which Icp cut-off / Vn overvoltage conditions are detected					
C_TTIP_THD_IPLSL_ACT_LS_UP	1	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
Ceramic temperature above which Ip control loop may be activated					
C_VB_MIN_IPLSL_CTL_ENA	1	0...FFH	0...25.89843	0.1015625	[V]
Threshold value of battery voltage for accurate operation of ATIC42					

C_TTIP_THD_ICPLSL_INC_LS_UP	1	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
Ceramic temperature above which Icp can be switched to elevated values (‘warm’ sensor)					
C_T_DLY_IPLSL_CTL_ENA	1	0...FFH	0...2.55	0.01	[s]
Delay to be exceeded from switching on of pump current control to setting validity flag					
C_T_DLY_MAX_VNLSL_LIM_ACT	1	0...FFFFH	0...655.35	0.01	[s]
Delay from determining Icp cut-off / Vn overvoltage to forced revokation of Ip control loop disable flag					
LC_ICPLSL_ENA_LSL_IF	1	0...1H	0...1	1	[-]
Boolean flag permitting manual activation / deactivation of pumped-reference current (Icp)					
LC_ICPLSL_ACT	1	0...1H	0...1	1	[-]
Boolean flag indicating whether upstream WRAF sensors are of type pumped-reference or not					
C_LAMB_THD_MAX_VNLSL_LIM	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Maximum value of lambda threshold for Vn overvoltage / reverse Icp cut-off monitoring					
C_LAMB_THD_MAX_VPLSL_LIM	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Maximum value of lambda threshold for Vp limitation					
C_LAMB_SP_MIN_IPLSL_CTL_ENA	1	0...7FFFH	0...31.99902	9.77E-04	[-]
MInimum value of lambda setpoint for activation of Ip control loop					
C_LAMB_SP_MAX_IPLSL_CTL_ENA	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Maximum value of lambda setpoint for activation of Ip control loop					
LC_IPLSL_CTL_ENA_MAN_SWI_LSL_IF	1	0...1H	0...1	1	[-]
Boolean flag permitting manual activation / deactivation of pump current control					
LC_IPLSL_CTL_ENA_MAN_LSL_IF	1	0...1H	0...1	1	[-]
Boolean flag requesting activation / deactivation of pump current control during manual control					
C_T_IPLSL_CTL_LSH_UP_MAN_ACT	1	0...FFFFH	0...655.35	0.01	[s]
Delay time for activation of pump current control after engine start in case of forced sensor pre-heating					
C_TTIP_MIN_LAMB_LS_UP_ESTIM	1	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
Minimum elememt temperature for accurate compensation of measured lambda signal					
C_MAF_KGH_MIN_LAMB_LS_UP_ESTIM	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Minimum MAF_KGH to guarantee sufficient sensor signal dynamics lambda-wise					
C_N_32_MAX_LAMB_LS_UP_ESTIM	1	0...FFH	0...8160	32	[rpm]
N threshold above which limited accuracy of measured lambda signal is expected					

16.17 WRAF Sensor Gain Adaptation at fuel cut-off

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IPLSL_COR_MMV_FCUT[NC_CBK_EX_NR]	V	8000...7FFFH	-12.596... 12.59561	0.3844e-3	[mA]
Moving mean value of corrected pumping current during fuel cut					
LV_IPLSL_COR_MMV_FCUT_VLD[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag stating validity of latest moving mean value of corrected pumping current during PUC					
IPLSL_PURGE[NC_CBK_EX_NR]	V	8000...7FFFH	-12.596... 12.59561	0.3844e-3	[mA]
Compensated (excluding factor for gain adaptation) pumping current during fuel cut at purged condition					
IPLSL_MV_PURGE[NC_CBK_EX_NR]	V	8000...7FFFH	-12.596... 12.59561	0.3844e-3	[mA]
Arithmetic mean of compensated pumping current during fuel cut at purged condition					
IPLSL_COR_BEG_PURGE[NC_CBK_EX_NR]	V	8000...7FFFH	-12.596... 12.59561	0.3844e-3	[mA]
Corrected pumping current during fuel cut at begin of purged condition					
IPLSL_COR_CHG_PURGE[NC_CBK_EX_NR]	V	8000...7FFFH	-12.596... 12.59561	0.3844e-3	[mA]
Change of corrected pumping current during acquisition at purged condition					
IPLSL_VARI_PURGE[NC_CBK_EX_NR]	V	0...FFFFFFFFH	0...634.63686	0.1478e-6	[(mA)²]
Variance of corrected pumping current during fuel cut at purged condition					
IPLSL_MMV_VLD_FCUT[NC_CBK_EX_NR]	V/O/S	8000...7FFFH	-12.596... 12.59561	0.3844e-3	[mA]
Moving mean of valid samples of pumping current during fuel cut					
IPLSL_VARI_VLD_FCUT[NC_CBK_EX_NR]	V/O/S	0...FFFFFFFFH	0...634.63686	0.1478e-6	[(mA)²]
Moving variance of valid samples of pumping current during fuel cut					
T_ACQ_IPLSL_PURGE[NC_CBK_EX_NR]	V	0...FFH	0...25.5	0.1	[s]
Timer for acquisition interval of compensated pumping current during fuel cut at purged condition					
T_DLY_IPLSL_COR_MMV_FCUT_VLD[NC_CBK_EX_NR]	V	0...FFH	0...25.5	0.1	[s]
Delay timer for retry of gain adaptation upon detection of possible saturation effects on Ip measurement					
STATE_IPLSL_AD_FCUT[NC_CBK_EX_NR]	V	0H 1H 2H 3H 4H	INIT WAIT PURGE ACQ ADAPT	1	[-]
State of fuel cut gain adaptation procedure					
STATE_IPLSL_GAIN_AD[NC_CBK_EX_NR]	V	0H 1H 2H 3H	IN_RANGE LOW_LIMIT HIGH_LIMIT IMPOSSIBLE	1	[-]
State of WRAF sensor gain adaptation factor					
LV_LSL_GAIN_AD_REQ[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Request for fuel cut Ip measurement					
LV_LSL_GAIN_AD_REQ_VLD[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Validation of request for fuel cut Ip measurement					
LV_IPLSL_VLD_FCUT[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Valid Ip sample has been acquired during current fuel cut					
LV_LSL_FCUT_MES_REP_INH[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Status flag indicating successful fuel cut Ip measurement during current PUC phase					
LV_LSL_FCUT_MES_NOT_PLAUS[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Actual fuel cut Ip measurement not plausible					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LSL_GAIN_AD_FCUT_NOT_OK[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Factor for gain adaptation out of valid range					
LV_FAC_LSL_GAIN_AD_LIM_MAX[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status flag for limitation of WRAF gain adaptation factor to maximum					
LV_FAC_LSL_GAIN_AD_LIM_MIN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status flag for limitation of WRAF gain adaptation factor to minimum					
LV_MAP_PUC_LIM_REQ_LSL_GAIN_AD	V/O	0...1H	0...1	1	[-]
Status flag for request of limitation of MAP during next PUC phase from torque management					
FAC_LSL_GAIN_AD[NC_CBK_EX_NR]	V/O/S	0...FFFFH	0...1.99996	0.0305e-3	[-]
Adaptive factor for WRAF sensor gain					
LV_LSL_FIRST_GAIN_AD[NC_CBK_EX_NR]	V/S	0...1H	0...1	1	[-]
Bit indicating that Gainadaptation is calculated for the first time					
MAF_INT_PUC_MIN_LSL_GAIN_AD[NC_CBK_EX_NR]	V	0...FFFFH	0...2912.66666	0.0444444	[g]
treshold for MAF-integral as transition condition from PURGE to ACQ					

Input data:

TCO	N_32	TTIP_MES_LS_UP[NC_CBK_EX_NR]	LV_PUC
IPLSL_COR[NC_CBK_EX_NR]	AMP	LV_INH_LSL_GAIN_AD[NC_CBK_EX_NR]	TAM
LV_LS_UP_READY[NC_CBK_EX_NR]	LV_V_REF_VLD_R_IT_LS_UP[NC_CBK_EX_NR]	MAF_KGH	LV_MAP_PUC_LIM_ACT_LSL_GAIN_AD
GEAR	LV_IGK	NC_CBK_EX_NR	LV_LAMB_LS_UP_VLD[NC_CBK_EX_NR]
LV_VPLSL_LIM[NC_CBK_EX_NR]	LV_IPLSL_VLD[NC_CBK_EX_NR]	STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]	LV_SWI_GAIN_LSL_IF[NC_CBK_EX_NR]
LV_VLS_UP_INIT[NC_CBK_EX_NR]	LV_VLS_UP_VLD[NC_CBK_EX_NR]	LV_CP_CLOSE_ACT	CTR_STOP_FSD
MAF_INT_PUC_ACT	LAMB_LS_UP[NC_CBK_EX_NR]		

FUNCTION DESCRIPTION:

During fuel cut-off the engine purges the exhaust line with air. This condition is the best suitable engine operating condition to check the WRAF sensor’s static gain.
The function comprises:

1. the recognition of a condition to be stable enough for a significant pumping current measurement.
2. pumping current correction due to interfering impacts as: exhaust pressure etc.
3. current acquisition and averaging during fuel cut.
4. evaluation of adaptation conditions
5. WRAF sensor gain adaptation

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Application conditions

Activation: *at LV_IGK = 1*

Deactivation: *at LV_IGK = 0*

Initialization **at first ECU run :**

FAC_LSL_GAIN_AD[i] = 1

IPLSL_MMV_VLD_FCUT[i] = C_IPLSL_NOM_PURGE

IPLSL_VARI_VLD_FCUT[i] = 0

LV_LSL_FIRST_GAIN_AD[i] = 1

Initialization at every transition **LV_IGK = 0 ->** 1, **ECU Reset, or upon erase of error management**:

STATE_IPLSL_AD_FCUT[i] = INIT

16.17.1 Conditions for WRAF Sensor fuel cut-off Ip determination

FUNCTION:

These conditions must be verified the whole time during the fuel cut-off Ip determination.

Request of MAP limitation during next PUC phase by any arbitrary exhaust cylinder bank shall lead to corresponding request to torque management.

Recurrence: T_SAMPLE = 100ms.

Perform prior to execution of FOR loop depending on [i].

```
If      LV_PUC == 0
then    LV_MAP_PUC_LIM_REQ_LSL_GAIN_AD = 0
endif
```

Fuel cut-off Ip determination is requested only:

```
If      ( LV_LS_UP_READY [i] == 1
and     LV_IPLSL_VLD[i] == 1
and     LV_CP_CLOSE_ACT == 1
and     CTR_STOP_FSD <= C_CTR_STOP_FSD_THD
and     TCO > C_TCO_MIN_LSL_GAIN_AD
and     TAM < C_TAM_MAX_LSL_GAIN_AD
and     ( C_N_32_MIN_LSL_GAIN_AD < N_32 < C_N_32_MAX_LSL_GAIN_AD
or      C_N_32_BOL_LSL_GAIN_AD < N_32  < C_N_32_TOL_LSL_GAIN_AD )
and     C_TTIP_MIN_LSL_GAIN_AD < TTIP_MES_LS_UP[i]
```

```
                        < C_TTIP_MAX_LSL_GAIN_AD
    and         AMP > C_AMP_MIN_LSL_GAIN_AD
    and         LV_V_REF_VLD_R_IT_LS_UP[i] == 1
                ( tip temperature of upstream sensor is valid )
    and         LC_LSL_GAIN_AD_ENA == 1
    and         GEAR ≥ C_NR_GEAR_MIN_LSL_GAIN_AD
    and         LV_INH_LSL_GAIN_AD[i] == 0
    and         LC_FAC_LSL_GAIN_AD_MAN_ADJ_ENA == 0 )
    or          LC_LSL_GAIN_AD_TEST == 1
    then        ( general conditions fulfilled )
                If      LV_PUC == 0
                then    If      LC_MAP_PUC_LIM_REQ_LSL_GAIN_AD == 1
                        then    LV_MAP_PUC_LIM_REQ_LSL_GAIN_AD = 1
                        endif
                        ( MAP limitation during next PUC is requested )
                        LV_LSL_GAIN_AD_REQ_VLD[i] = 1
                        ( request of gain adaptation will be seen valid )
                        LV_LSL_GAIN_AD_REQ[i] = 0
                else    If      LV_LSL_GAIN_AD_REQ_VLD[i] == 1
                        then    LV_LSL_GAIN_AD_REQ[i] = 1
                                ( gain adaptation is requested )
                                If      LV_LAMB_LS_UP_VLD[i] == 1
                                and     LV_VPLSL_LIM[i] == 0
                                and     LV_VLS_UP_VLD[i] == 1
                                and     T_DLY_IPLSL_COR_MMV_FCUT_VLD[i] == 0
                                then    LV_IPLSL_COR_MMV_FCUT_VLD[i] = 1
                                else    decrement T_DLY_IPLSL_COR_MMV_FCUT_VLD[i]
                                endif
                        endif
                endif
    else        If      LC_FAC_LSL_GAIN_AD_MAN_ADJ_ENA == 1
                then    FAC_LSL_GAIN_AD[i] = C_FAC_LSL_GAIN_AD_MAN_ADJ
                        LV_LSL_GAIN_AD_FCUT_NOT_OK[i] = 0
                        LV_FAC_LSL_GAIN_AD_LIM_MAX[i] = 0
                        LV_FAC_LSL_GAIN_AD_LIM_MIN[i] = 0
                endif
```

```
                LV_LSL_GAIN_AD_REQ[i] = 0
                LV_LSL_GAIN_AD_REQ_VLD[i] = 0
    endif
```

16.17.2 Low Pass Filtering of Pumping Current during fuel cut

FUNCTION:

Recurrence: T_SAMPLE = 10 ms.

Remark: Index ‘k’ is linked to T_SAMPLE = 10ms.

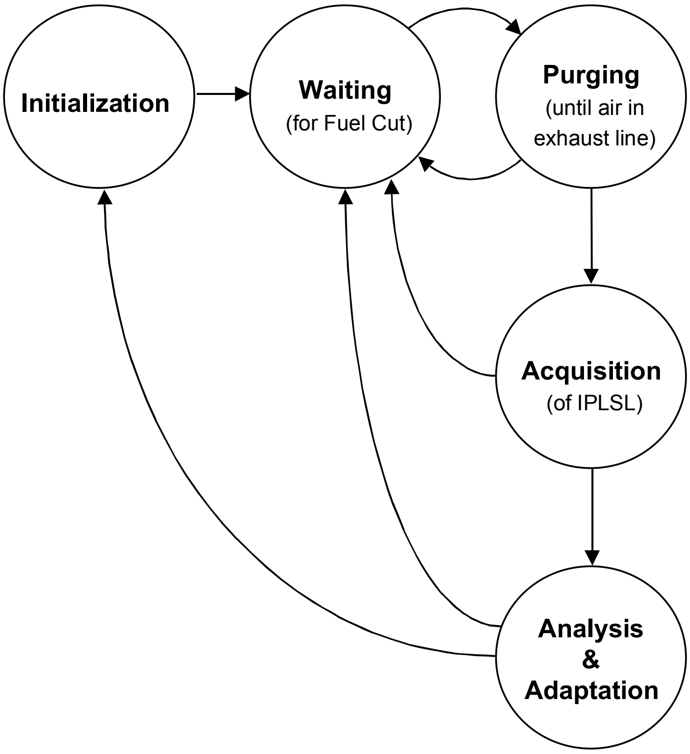
```
If      LV_LSL_GAIN_AD_REQ[i] == 1
then    If      LV_LAMB_LS_UP_VLD[i] = 1
        and     LV_VPLSL_LIM[i] = 0
        and     LV_VLS_UP_VLD[i] = 1
        then    $IPLSL\_COR\_MMV\_FCUT[i]_{k} = IPLSL\_COR\_MMV\_FCUT[i]_{k-1}$
                        $+ C\_CRLC\_IPLSL\_COR\_FCUT$
                        $* ( IPLSL\_COR[i] - IPLSL\_COR\_MMV\_FCUT[i]_{k-1} )$
        else    LV_IPLSL_COR_MMV_FCUT_VLD[i] = 0
                T_DLY_IPLSL_COR_MMV_FCUT_VLD[i]
                        = C_T_DLY_IPLSL_COR_MMV_FCUT_VLD
                IPLSL_COR_MMV_FCUT[i] = IPLSL_COR[i]
        endif
else    IPLSL_COR_MMV_FCUT[i] = IPLSL_COR [i]
endif
```

16.17.3 Sequence for fuel cut-off Ip determination:

FUNCTION:

Recurrence: T_SAMPLE = 100 ms.

Remark: Index ‘n’ is linked to T_SAMPLE = 100ms.



16.17.3.1 State ”INIT”:

Actions:

LV_LSL_GAIN_AD_REQ[i] = 0
LV_LSL_FCUT_MES_REP_INH[i] = 0
LV_LSL_FCUT_MES_NOT_PLAUS[i] = 0
LV_LSL_GAIN_AD_FCUT_NOT_OK[i] = 0
LV_IPLSL_COR_MMV_FCUT_VLD[i] = 1
T_DLY_IPLSL_COR_MMV_FCUT_VLD[i] = 0

Transition to state ”WAIT”:

Conditions:

none

Actions during transition:

STATE_IPLSL_AD_FCUT[i] = WAIT

16.17.3.2 State ”WAIT” :

STATE_IPLSL_AD_FCUT[i] remains in this wait state as long as the conditions for fuel cut Off Ip determination are not fulfilled or not fulfilled anymore. Furthermore it is taken care that only one measurement per fuel cut is done.

Actions:

```
If      LV_PUC == 0
then    LV_LSL_FCUT_MES_REP_INH[i] = 0
endif
```

Transition to state ”PURGE":

Conditions:

```
If      LV_LSL_GAIN_AD_REQ[i] == 1
and     LV_LSL_FCUT_MES_REP_INH[i] == 0
        ( LV_LSL_FCUT_MES_REP_INH[i] == 1 prevents a second measurement
              during the same fuel cut )
and     ( LV_MAP_PUC_LIM_ACT_LSL_GAIN_AD == 1
or      LC_MAP_PUC_LIM_ACT_NOT_VLD == 1 )
and     LV_SWI_GAIN_LSL_IF[i] == LC_IPLSL_GAIN_SEL_LSL_GAIN_AD
and     STATE_LSL_IF_SPI_RD[i] ( bit SWI_GAIN )
              == LC_IPLSL_GAIN_SEL_LSL_GAIN_AD
and     LV_VLS_UP_INIT[i] == 0
        ( Confirmation that required gain for pump current measurement is active )
then    " transition to PURGE "
endif
```

Actions during transition:

STATE_IPLSL_AD_FCUT[i] = PURGE

IPLSL_COR_MMV_FCUT[i] = IPLSL_COR[i]

$\text{IPLSL\_COR\_MMV\_FCUT[i]}_{n-1} = \text{IPLSL\_COR\_MMV\_FCUT[i]}_{n}$

LV_IPLSL_VLD_FCUT[i] = 0

MAF_INT_PUC_MIN_LSL_GAIN_AD[i]
= IP_MAF_INT_PUC_MIN_LSL_GAIN_AD(LAMB_LS_UP[i])

16.17.3.3 State: ”PURGE”:

During fuel cut STATE_IPLSL_AD_FCUT[i] waits until the exhaust system is purged and the pumping current is stable. The stability is reached if:

- the gradient of the averaged pumping current is below a threshold
- and the absolute pumping current is higher than a given threshold.

Actions:

none

Transition to state ”WAIT”:

Conditions:

```
If      LV_LSL_GAIN_AD_REQ[i] == 0
then    " transition to WAIT "
endif
```

Actions during transition:

STATE_IPLSL_AD_FCUT[i] = WAIT

Transition to state ”Acquisition":

Conditions:

```
If      LV_LSL_GAIN_AD_REQ[i] == 1
and     ( $IPLSL\_COR\_MMV\_FCUT[i]_n$ - $IPLSL\_COR\_MMV\_FCUT[i]_{n-1}$ )
                < C_IPLSL_COR_GRD_THD_FCUT
and     $IPLSL\_COR\_MMV\_FCUT[i]_n$
                > C_FAC_IPLSL_COR_THD_FCUT * C_IPLSL_NOM_PURGE
and     MAF_INT_PUC_ACT > MAF_INT_PUC_MIN_LSL_GAIN_AD[i]
then    " transition to ACQUISITION "
else    $IPLSL\_COR\_MMV\_FCUT[i]_{n-1}$ = $IPLSL\_COR\_MMV\_FCUT[i]_n$
        “ stay in PURGE ”
endif
```

Actions during transition:

STATE_IPLSL_AD_FCUT[i] = ACQUISITION

IPLSL_VARI_PURGE[i] = 0

T_ACQ_IPLSL_PURGE[i] = 0

IPLSL_COR_BEG_PURGE[i] = IPLSL_COR_MMV_FCUT[i]

16.17.3.4 State ”Acquisition” :

As input the pumping current (IPLSL_COR_MMV_FCUT[i]) is compensated for actual impacts of:

- ambient pressure
- engine speed
- MAF_KGH
- (and WRAF sensor TTIP_MES_LS_UP[i])

Afterwards the real current (IPLSL_PURGE[i]) is calculated by division with the actual compensation factor (FAC_LSL_GAIN_AD[i]).

During the acquisition time (C_T_ACQ_IPLSL_PURGE) the actual real current (at fuel cut and at purged exhaust line) is averaged (IPLSL_MV_PURGE[i]) and its variance (IPLSL_VARI_PURGE[i]) is calculated. At the end of the acquisition a final compensation of residual gas purging is done by multiplying with IP_FAC_COR_IPLSL_CHG_PURGE with the latter depending on the Ip change from start to end of acquisition.

Actions:

```
If      LV_IPLSL_COR_MMV_FCUT_VLD[i] = 1
then    If      T_ACQ_IPLSL_PURGE[i] < C_T_ACQ_IPLSL_PURGE
        then    IPLSL_PURGE[i] = IPLSL_COR_MMV_FCUT[i]
                        * IP_FAC_COR_IPLSL_MAF_KGH * IP_FAC_COR_IPLSL_N_32
                          / FAC_LSL_GAIN_AD[i]
                IPLSL_MV_PURGE[i]$_n$ = ( T_ACQ_IPLSL_PURGE[i]
                        * IPLSL_MV_PURGE[i]$_{n-1}$ + T_SAMPLE * IPLSL_PURGE[i] )
                        / ( T_ACQ_IPLSL_PURGE[i] + T_SAMPLE )
                IPLSL_VARI_PURGE[i]$_n$ =
                        ( T_ACQ_IPLSL_PURGE[i] * IPLSL_VARI_PURGE[i]$_{n-1}$
                        + T_SAMPLE * ( IPLSL_MV_PURGE[i] - IPLSL_PURGE[i] ) ^2 )
                        / ( T_ACQ_IPLSL_PURGE[i] + T_SAMPLE )
                T_ACQ_IPLSL_PURGE[i]$_n$  = T_ACQ_IPLSL_PURGE[i]$_{n-1}$ + T_SAMPLE
        endif
else    IPLSL_COR_BEG_PURGE[i] = IPLSL_COR_MMV_FCUT[i]
        T_ACQ_IPLSL_PURGE[i] = 0
        IPLSL_VARI_PURGE[i] = 0
endif
If      T_ACQ_IPLSL_PURGE[i] == C_T_ACQ_IPLSL_PURGE
and     LV_LSL_FCUT_MES_REP_INH[i] == 0
then    LV_LSL_FCUT_MES_REP_INH[i] = 1
        IPLSL_COR_CHG_PURGE[i] =
                IPLSL_COR_MMV_FCUT[i] - IPLSL_COR_BEG_PURGE[i]
        IPLSL_MV_PURGE[i] =
                IPLSL_MV_PURGE[i] * IP_FAC_COR_IPLSL_CHG_PURGE
endif
```

Transition to state ”WAIT":

Conditions:

```
    If      LV_LSL_GAIN_AD_REQ[i] == 0
    then    " transition to WAIT "
    endif
```

Actions during transition:

```
    STATE_IPLSL_AD_FCUT[i] = WAIT
```

Transition to state ”ADAPT":

Conditions:

```
If      LV_LSL_GAIN_AD_REQ[i] == 1
and     LV_LSL_FCUT_MES_REP_INH[i] == 1
then    " transition to ADAPT "
endif
```

Actions during transition:

```
STATE_IPLSL_AD_FCUT[i] = ADAPT
```

16.17.3.5 State ”ADAPT”:

The fuel cut pumping current sample is checked for sensor operability and stability during the previous acquisition phase. Therefore the variance of the output must be between allowed limits. If this condition or the conditions for ”fuel cut-off Ip determination” are not fulfilled anymore the STATE_IPLSL_AD_FCUT[i] switches back to ”WAIT”.

If the conditions are fulfilled it will be declared as ”valid sample”. Next a MMV value and the variance of the valid Ip samples are calculated.

If the gain adaptation takes place for the first time (in case of a new sensor or new ECU) the MMV value is initialized with the mean-value of the first measurement. The variance stays on the initial value (zero) in this case. The calculation of MMV and variance is started as soon as one successful gain adaptation was carried out (by setting LV_FIRST_GAIN_AD = 0).

A new adaptive factor (FAC_LSL_GAIN_AD[i]) for the WRAF sensor’s gain is calculated if the following conditions for adaptation are fulfilled:

1. Variance of previous valid samples in permitted range.
2. Adaptive factor (FAC_LSL_GAIN_AD[i]) in permitted range

Since the new adaptive factor is calculated based on the MMV value of the valid samples it follows the average of the last measurements and not direct every particular fuel cut sample. The response time of the adaptation is defined by the correlation constant (C_CRLC_IPLSL_MV_VLD).

Actions:

```
If      LV_IPLSL_VLD_FCUT[i] == 0
        ( LV_IPLSL_VLD_FCUT[i] == 1 prevents a second evaluation and
        updating of IPLSL_MMV_VLD_FCUT[i] during the same fuel cut )
and     C_IPLSL_VARI_THD_MIN_PURGE <= IPLSL_VARI_PURGE[i]
                < C_IPLSL_VARI_THD_MAX_PURGE
then    LV_LSL_FCUT_MES_NOT_PLAUS[i] = 0
        LV_IPLSL_VLD_FCUT[i] = 1

        Average of recent valid fuel cuts & Variance of recent valid fuel cuts:
        If      (LV_LSL_FIRST_GAIN_AD[i] == 1 and LC_LSL_FIRST_GAIN_AD == 1)
```

```
then    IPLSL_MMV_VLD_FCUT[i]$_{i}$ = IPLSL_MV_PURGE[i]$_{i}$
        IPLSL_VARI_VLD_FCUT[i]$_{i}$ = IPLSL_VARI_VLD_FCUT[i]$_{i-1}$

else    IPLSL_MMV_VLD_FCUT[i]$_{i}$ = IPLSL_MMV_VLD_FCUT[i]$_{i-1}$
        + C_CRLC_IPLSL_MV_VLD * ( IPLSL_MV_PURGE[i]
        –   IPLSL_MMV_VLD_FCUT[i]$_{i-1}$ )

        IPLSL_VARI_VLD_FCUT[i]$_{i}$ = IPLSL_VARI_VLD_FCUT[i]$_{i-1}$
        + C_CRLC_IPLSL_MV_VLD * (( IPLSL_MV_PURGE[i]
        - IPLSL_MMV_VLD_FCUT[i]$_{i}$ ) ^2 - IPLSL_VARI_VLD_FCUT[i]$_{i-1}$ )

end
```

Adaptation conditions

```
If      IPLSL_VARI_VLD_FCUT[i] > C_IPLSL_MV_VLD_VARI_MAX
then    STATE_IPLSL_GAIN_AD[i] = 3               ( impossible )
        ( adaptation impossible due variance limit of valid fuel cut
        measurements exceeded )
else    If      ( C_IPLSL_NOM_PURGE / IPLSL_MMV_VLD_FCUT[i] )
                        > C_FAC_LSL_GAIN_AD_LIM_MAX
        then    STATE_IPLSL_GAIN_AD[i] = 1       ( low Ip limit )
                If      LC_FAC_LSL_GAIN_AD_LIM_RST == 1
                then    FAC_LSL_GAIN_AD[i] = 1
                else    FAC_LSL_GAIN_AD[i]
                                = C_FAC_LSL_GAIN_AD_LIM_MAX
                endif
                LV_FAC_LSL_GAIN_AD_LIM_MIN[i] = 0
                LV_FAC_LSL_GAIN_AD_LIM_MAX[i] = 1
                ( upper adapt. factor limit exceeded
                since pumping current lower than allowed limit )
        else    LV_FAC_LSL_GAIN_AD_LIM_MAX[i] = 0
                If      ( C_IPLSL_NOM_PURGE / IPLSL_MMV_VLD_FCUT[i] )
                                < C_FAC_LSL_GAIN_AD_LIM_MIN
                then    STATE_IPLSL_GAIN_AD[i] = 2       ( high Ip limit )
                        If      LC_FAC_LSL_GAIN_AD_LIM_RST = 1
```

```
                        then    FAC_LSL_GAIN_AD[i] = 1
                        else    FAC_LSL_GAIN_AD[i]
                                        = C_FAC_LSL_GAIN_AD_LIM_MIN
                        endif
                        LV_FAC_LSL_GAIN_AD_LIM_MIN[i] = 1
                        ( lower adaptation factor limit exceeded,
                        since pumping current higher than allowed limit )
                else    STATE_IPLSL_GAIN_AD[i] = 0       ( in range )
                        LV_FAC_LSL_GAIN_AD_LIM_MIN[i] = 0
                endif
        endif
endif

Adaptation
If      STATE_IPLSL_GAIN_AD[i] == 0     ( in range )
then    LV_LSL_GAIN_AD_FCUT_NOT_OK[i] = 0
        If      LC_FAC_LSL_GAIN_AD_ENA == 1
        then    FAC_LSL_GAIN_AD[i] =
                        C_IPLSL_NOM_PURGE / IPLSL_MMV_VLD_FCUT[i]
                ( New adaptation factor = nominal current divided by MMV
                of uncompensated current )
                LV_LSL_FIRST_GAIN_AD[i] = 0
        else    FAC_LSL_GAIN_AD[i] = 1
        endif
else    LV_LSL_GAIN_AD_FCUT_NOT_OK[i] = 1
endif
```

```
else    LV_LSL_FCUT_MES_NOT_PLAUS[i] = 1
endif
```

Transition to state ”WAIT":

Conditions:

```
If      LV_LSL_FCUT_MES_NOT_PLAUS[i] == 1
or      LV_LSL_GAIN_AD_FCUT_NOT_OK[i] == 1
then    " transition to WAIT ”
endif
```

Actions during transition:

STATE_IPLSL_AD_FCUT[i] = WAIT

<u>Transition to state ”INIT":</u>

Conditions:

```
    If      LV_IPLSL_VLD_FCUT[i] == 1
    and     LV_PUC == 0
    then    " transition to INIT "
    endif
```

Actions during transition:

STATE_IPLSL_AD_FCUT[i] = INIT

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_MIN_LSL_GAIN_AD	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant temperature to allow fuel cut Ip determination					
C_TAM_MAX_LSL_GAIN_AD	1	0...FEH	-48...142.5	0.75	[°C]
Maximum ambient temperature to allow fuel cut Ip determination					
C_AMP_MIN_LSL_GAIN_AD	1	0...FFFFH	0...5434	8.29E-02	[hPa]
Minimum ambient pressure to allow fuel cut Ip determination					
C_TTIP_MIN_LSL_GAIN_AD	1	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
Minimum tip temperature to allow fuel cut Ip determination					
C_TTIP_MAX_LSL_GAIN_AD	1	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
Maximum tip temperature to allow fuel cut Ip determination					
C_N_32_BOL_LSL_GAIN_AD	1	0...FFH	0...8160	32	[rpm]
Additional bottom limit above which fuel cut Ip determination is allowed to take place					
C_N_32_TOL_LSL_GAIN_AD	1	0...FFH	0...8160	32	[rpm]
Additional top limit below which fuel cut Ip determination is allowed to take place					
C_N_32_MIN_LSL_GAIN_AD	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed to allow fuel cut Ip determination					
C_N_32_MAX_LSL_GAIN_AD	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed to allow fuel cut Ip determination					
C_CRLC_IPLSL_COR_FCUT	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Correlation constant for filtering of corrected pumping current					
C_CRLC_IPLSL_MV_VLD	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Correlation constant for filtering of valid Ip samples (is also time constant for adaptation)					
C_IPLSL_COR_GRD_THD_FCUT	1	8000...7FFFH	-12.596...12.59561	3.84E-04	[mA/100ms]
Maximum gradient of low pass filtered pumping current to detect stable conditions (end of PURGE)					
C_FAC_IPLSL_COR_THD_FCUT	1	0...FFH	0...0.99609	3.91E-03	[-]
Factor for minimum threshold of the low pass filtered pumping current to detect stable conditions (end of PURGE)					
C_IPLSL_VARI_THD_MAX_PURGE	1	0...FFFFFFFFH	0...634.63686	1.48E-07	[mA²]
Maximum variance of the low pass filtered pumping current to confirm stable conditions					
C_IPLSL_VARI_THD_MIN_PURGE	1	0...FFFFFFFFH	0...634.63686	1.48E-07	[mA²]
Minimum variance of the low pass filtered pumping current to confirm that sensor is operable					
C_T_ACQ_IPLSL_PURGE	1	1...FFH	0.1...25.5	0.1	[s]
Duration of Ip acquisition and averaging					
C_T_DLY_IPLSL_COR_MMV_FCUT_VLD	1	1...FFH	0.1...25.5	0.1	[s]
Delay time for retry of gain adaptation upon detection of possible saturation effects on Ip measurement					
IP_FAC_COR_IPLSL_MAF_KGH	12	0...FFFFH	0...1.99996	3.05E-05	[-]
LDP_MAF_KGH_IP_FAC_COR_IPLSL	12	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Factor for MAF correction					
IP_FAC_COR_IPLSL_N_32	16	0...FFFFH	0...1.99996	3.05E-05	[-]
LDP_N_32_IP_FAC_COR_IPLSL	16	0...FFH	0...8160	32	[rpm]
Factor for engine speed correction					
IP_FAC_COR_IPLSL_CHG_PURGE	4	0...FFFFH	0...1.99996	3.05E-05	[-]
LDP_IPLSL_CHG_PURGE_IP_IPLSL	4	0...FFFFH	-12.596...12.59561	3.84E-04	[mA]
Factor for residual gas after purging correction					
C_IPLSL_NOM_PURGE	1	8000...7FFFH	-12.596...12.59561	3.84E-04	[mA]
Nominal pumping current at fuel cut					
LC_FAC_LSL_GAIN_AD_MAN_ADJ_ENA	1	0...1H	0...1	1	[-]
Enable manual adjustment of WRAF sensor gain adaptation factor					
C_FAC_LSL_GAIN_AD_MAN_ADJ	1	0...FFFFH	0...1.99996	3.05E-05	[-]

Value for manual adjustment of WRAF sensor gain adaptation factor					
C_FAC_LSL_GAIN_AD_LIM_MIN	1	0...FFFFH	0...1.99996	3.05E-05	[-]
Minimum threshold for WRAF sensor gain adaptation					
C_FAC_LSL_GAIN_AD_LIM_MAX	1	0...FFFFH	0...1.99996	3.05E-05	[-]
Maximum threshold for WRAF sensor gain adaptation					
LC_FAC_LSL_GAIN_AD_LIM_RST	1	0...1H	0...1	1	[-]
Switch for reset of adaptation factor to one when exceeding thresholds, or for adoption of threshold value, resp.					
C_IPLSL_MV_VLD_VARI_MAX	1	0...FFFFFFFFH	0...634.63686	1.48E-07	[mA²]
Maximum permitted variance of valid Ip sample to allow adaptation					
C_NR_GEAR_MIN_LSL_GAIN_AD	1	0...FFH	0...255	1	[-]
Minimum gear to allow fuel cut Ip determination					
LC_FAC_LSL_GAIN_AD_ENA	1	0...1H	0...1	1	[-]
Enable adoption of WRAF sensor gain adaptation factor					
LC_LSL_GAIN_AD_ENA	1	0...1H	0...1	1	[-]
Enable WRAF sensor gain adaptation					
LC_IPLSL_GAIN_SEL_LSL_GAIN_AD	1	0...1H	0...1	1	[-]
Determines which Ip gain has to be active during execution of gain adaptation (‘0’ = gain8, ‘1’ = gain16)					
LC_MAP_PUC_LIM_REQ_LSL_GAIN_AD	1	0...1H	0...1	1	[-]
Enable request to torque management for limitation of MAP during PUC					
LC_MAP_PUC_LIM_ACT_NOT_VLD	1	0...1H	0...1	1	[-]
Flag to ignore activation of limitation of MAP during PUC by torque management					
LC_LSL_GAIN_AD_TEST	1	0...1H	0...1	1	[-]
Manual enable of LV_LSL_GAIN_AD_REQ_i for test purposes					
C_CTR_STOP_FSD_THD	1	0...FFH	0...255	1	[-]
threshold for CTR_STOP_FSD to allow gain adaptation					
LC_LSL_FIRST_GAIN_AD	1	0...1H	0...1	1	[-]
1: no filtering for first calculation of correction value after exchange of ECU or sensor					
IP_MAF_INT_PUC_MIN_LSL_GAIN_AD	6	0...FFFFH	0...2912.66666	0.0444444	[g]
LDP_LAMB_LS_UP_IP_MAF_INT	6	0...7FFFH	0...31.99902	0.9766e-3	[-]
Treshold for MAF-integral as transition condition from PURGE to ACQ					

16.18 Activation conditions for WRAF sensor gain adaptation at fuel cut off

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_LSL_GAIN_AD[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
WRAF sensor Gain Adaptation: application incidences for inhibition					

Input data:

OPM_AV	LV_ERR_MAF	STATE_ERR_IV	LV_ERR_IGC
LV_IGK	LV_ERR_TPS	LV_ERR_LOAD_TPS_PLAUS	NC_CBK_EX_NR
LV_ERR_IVVT	LV_ERR_AMP	LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_ERR_AMP_PLAUS
LV_ERR_TAM_PLAUS	LV_ST_END	LV_ERR_CAM_CUS	LV_ERR_CHG_LS_UP
LV_ERR_TAM_CAN	LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]
LV_ERR_EL_CPS	LV_ERR_DIAGCPS	LV_ERR_SA_SAV_LSL	LV_ERR_SAV
LV_ERR_SA_SAV	LV_ERR_SAP	LV_MIS_STATE_B	LV_MIS_STATE_A
LV_ERR_CRK	LV_ERR_TCO	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]
LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS	

FUNCTION DESCRIPTION:

General information:

If one of the below mentioned conditions is not fulfilled, the inhibit bit interrupts the WRAF sensor gain adaptation at fuel cut-off.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation: at every LV_IGK = 0 ⇒ 1 and reset all output variables are initialized with 0

Recurrence: T_SAMPLE = 100 ms.

Activation: at LV_IGK = 1

Deactivation: at LV_IGK = 0

Formula section:

```
IF      LV_ERR_MAF = 1                              mass air flow sensor                      OR
    -   STATE_ERR_IV <> 0                           injection valve                           OR
    -   LV_ERR_TPS = 1                              Throttle position sensor                  OR
    -   LV_ERR_IGC = 1                                                                        OR
    -   LV_ERR_FSD_LAM_LIM [i] = 1                                                            OR
    -   LV_ERR_TTIP_MES_LSH_UP[i] = 1                                                         OR
    -   LV_ERR_LOAD_TPS_PLAUS = 1                   (MAF, ISA, AMP)  PlausiCheck              OR
    -   LV_ERR_IVVT = 1                             infinitaly valve timing                   OR
    -   LV_ERR_LS_UP[NC_CBK_EX_NR] = 1              WRAF sensor                               OR
    -   LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR] = 1            trimm controller diag.                OR
    -   LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR] = 1                                                 OR
    -   LV_ERR_CHG_LS_UP = 1                        swapped WRAF sensor                       OR
    -   LV_ERR_EL_CPS = 1                           Canister purge valve                      OR
    -   LV_ERR_DIAGCPS = 1                      Canister purge valve mechanical blocked       OR
    -   LV_ERR_FSD[NC_CBK_EX_NR] = 1                 fuel system diagnosis                    OR
    -   LV_ERR_SA_SAV = 1                       Secondary air valve error detected            OR
    -   LV_ERR_SA_SAV_LSL = 1                                                                 OR
    -   LV_ERR_SAV = 1                              Secondary air valve                       OR
    -   LV_ERR_SAP = 1                              Secondary air pumpe                       OR
    -   LV_MIS_STATE_B = 1                          Misfire State B detected                  OR
    -   LV_MIS_STATE_A = 1                          Misfire State A detected                  OR
    -   LV_ERR_CRK = 1                              crankshaft sensor                         OR
    -   LV_ERR_CAM_CUS = 1                                                                    OR
    -   LV_ERR_TCO = 1                              Coolant temperature sensor                OR
    -   LV_ERR_TAM_CAN = 1                                                                    OR
    -   LV_ERR_TAM_PLAUS = 1                                                                  OR
    -   LV_ERR_AMP = 1                                                                        OR
    -   LV_ERR_AMP_PLAUS = 1                                                                  OR
    -   LV_ERR_OFS_LSL_UP[i] = 1                                                              OR
    -   LV_ERR_CTL_LSL_UP[i] = 1                                                              OR
    -   LV_ERR_MAP_TPS_PLAUS = 1                                                              OR
    -    (LC_LSL_GAIN_AD_ENA_ALL_OPM = 0 and OPM_AV <> 1)
THEN
   LV_INH_LSL_GAIN_AD[NC_CBK_EX_NR]  =  1
ELSE
   LV_INH_LSL_GAIN_AD[NC_CBK_EX_NR]  =  0
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_LSL_GAIN_AD_ENA_ALL_OPM	1	0...1H	0...1	1	[-]
Logical constant to enable gain adaption in all engine operation modes					

16.19 Upstream oxygen sensor heater management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LSHPWM_UP[NC_CBK_EX_NR]	V/O	0...FFH	0...99.60937	0.390625	[%]
Heater driver PWM duty cycle, acquired also by BSW					
V_EFC_LSH_UP[NC_CBK_EX_NR]	V/O	0...FFH	0...25.89843	0.1015625	[V]
Effective final output value of heater voltage					
V_EFC_CLC_LSH_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...25.99960	3.97E-04	[V]
Effective calculated heater voltage value prior to overvoltage protection					
V_EFC_CTL_ADD_LSH_UP[NC_CBK_EX_NR]	V	80...7FH	-13...12.89843	0.1015625	[V]
Additive effective heater voltage from closed loop controller					
V_EFC_CTL_P_LSH_UP[NC_CBK_EX_NR]	V	8000...7FFFH	-13...12.99960	3.97E-04	[V]
Additive effective heater voltage from closed loop controller, P term					
V_EFC_CTL_I_LSH_UP[NC_CBK_EX_NR]	V	8000...7FFFH	-13...12.99960	3.97E-04	[V]
Additive effective heater voltage from closed loop controller, I term					
V_EFC_CTL_P_LSH_UP_TMP[NC_CBK_EX_NR]	-	8000...7FFFH	-13...12.99960	0.3967e-3	[V]
Intermediate additive effective heater voltage from closed loop controller, P term					
V_EFC_CTL_I_LSH_UP_TMP[NC_CBK_EX_NR]	-	8000...7FFFH	-13...12.99960	0.3967e-3	[V]
Intermediate additive effective heater voltage from closed loop controller, I term					
TTIP_MES_DIF_LS_UP[NC_CBK_EX_NR]	V	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
Difference between actual and set oxygen sensor tip temperature					
T_POW_RISE_LSH_UP[NC_CBK_EX_NR]	V/O	0...FFFFH	0...6553.5	0.1	[s]
Timer indicating the duration of the pre-heating and post dew point heating phases					
T_V_EFC_TOL_LSH_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer indicating permitted effective heater voltage overvoltage duration					
T_V_EFC_LIM_LSH_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer indicating the duration of the effective heater voltage limitation preventing overvoltage					
LV_LSH_CTL_CLL_LSH_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean variable indicating closed loop control mode active					
LV_V_EFC_LIM_BOL_LSH_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating effective heater voltage limited due to sum of voltage components falling below zero.					
LV_V_EFC_LIM_TOL_LSH_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating effective heater voltage limited due to exceeding threshold for period of time.					
LV_V_EFC_LIM_MAX_LSH_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating effective heater voltage limited due to exceeding absolute maximum voltage spec.					
LV_V_EFC_LIM_PROT_VB_LSH_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean variable indicating effective heater voltage limited due to excessive battery voltage level					
LV_LSH_UP_MAN_ACT[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean variable indicating that forced lambda probe heating before engine start is active					
STATE_LSH_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 3H 4H 5H 6H	LSH_OFF LSH_POW_RISE LSH_POW_RED LSH_POW_FALL LSH_POW_CTL LSH_VB_PROT LSH_TEMP_PROT	1	[-]
Present heater state					

Input data:

LV_INH_LSH_UP[NC_CBK_EX_NR]	LV_LS_UP_READY_CDN[NC_CBK_EX_NR]	TEMP_INI_LS_UP[NC_CBK_EX_NR]	LV_INH_LSH_CTL_CLL_LSH_UP[NC_CBK_EX_NR]
LV_ST_END	VS	TTIP_MES_LS_UP[NC_CBK_EX_NR]	LV_TEMP_DEW_LS_UP[NC_CBK_EX_NR]
VB	N_32	TEG_DYN_LS_UP[NC_CBK_EX_NR]	LV_TTIP_MES_VLD_LS_UP[NC_CBK_EX_NR]
NC_CBK_EX_NR	T_AST	LSHPWM_EXT_LS_UP[NC_CBK_EX_NR]	LV_V_REF_VLD_R_IT_LS_UP[NC_CBK_EX_NR]
LV_ES	LV_STST_STOP_CYC		

Import actions:

ACTION_INFR_SetPwmLshUp(IN <i>, IN <ducy>)
This action sets the lambda heater upstream driver

FUNCTION DESCRIPTION:

General information:

If two separate cylinder banks are concerned, then
i = 1, for cylinder bank 1
i = 2, for cylinder bank 2

The heater control shall be realised by use of a state machine. Each upstream oxygen sensor shall have its own state machine and each state machine shall run independently of the other.

Lambda probe heating after engine start (normal case)

The goal of the heater control shall be to control the oxygen sensor heating such that the optimal operating temperature is reached in the shortest time possible. At the same time, the maximum permissible temperature gradient (possible damage to ceramic due to thermal stress), the possible occurrence of water splash under the dew point (possible damage to ceramic due to thermal shock) and the absolute maximum ratings specified for the sensor shall be taken into account.

The function shall permit oxygen sensor heating under open and closed loop control.

Signal flow:

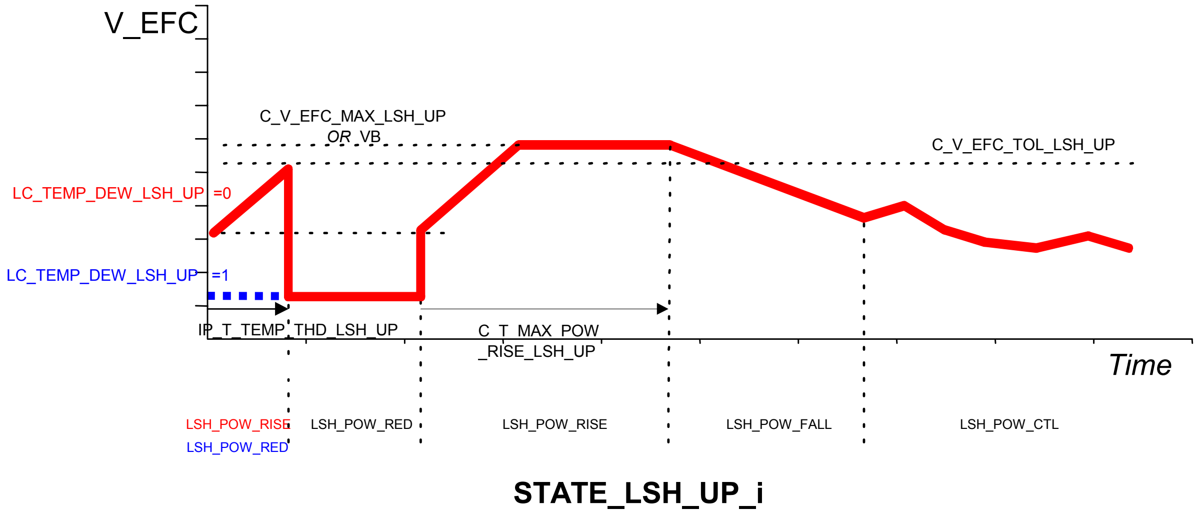


Figure 1: Course of effective heater voltage over time / state

The heater control shall be initialised to the LSH_OFF state at the start of a new driving cycle. Hence the oxygen sensor will not initially be electrically heated if not otherwise required (see sub chapter "forced lambda probe heating before engine start").

In the normal case (no forced heating before engine start) heating shall be started when the engine has transitioned from the start state (LV_ST_END = 1) and the heater management inhibit bit for the correspopnding bank is not set (LV_INH_LSH_UP[i] = 0). Depending on the lambda sensor requirement specification following two methods of initial lambda sensor heating are possible as judged by calibration of LC_TEMP_DEW_LSH_UP (see figure 1). This is due to the fact that for lambda sensors until dew point recognition either a certain temperature limit or a defined voltage limit must not be exceeded as the lambda sensor may be damaged by water splash or excessive temperature gradient otherwise. Depending on the setting of LC_TEMP_DEW_LSH_UP the heater management state machine shall thus switch from LSH_OFF to LSH_POW_RISE (LC_TEMP_DEW_LSH_UP = 0) or to LSH_POW_RED (LC_TEMP_DEW_LSH_UP = 1). In case of direct dew point recognition after hot restart the state machine shall switch to LSH_POW_RISE.

1.) Transition from LSH_OFF to LSH_POW_RED before dew point recognition

In case of transition from LSH_OFF to LSH_POW_RED the initial heater power is set by setting the effective heater voltage to C_V_EFC_RED_LSH_UP. The reduced heating state where water splash may occur shall persist until the dew point in the exhaust gas system has been recognized, i.e. until LV_TEMP_DEW_LS_UP[i] = 1, at which point the heater state shall switch to LSH_POW_RISE.

The effective heater voltage shall be set to the effective heater voltage as determined by IP_V_EFC_INI_LSH_UP, which may be modified by the calibration data C_V_EFC_STEP_LSH_UP, and may further be incremented with a gradient of C_V_EFC_INC_LSH_UP without danger of damaging the oxygen sensor. Additionally the timer T_POW_RISE_LSH_UP[i] shall be reinitialised with the IP_T_TEMP_THD_LSH_UP determined at function start.

2.) Transition from LSH_OFF to LSH_POW_RISE before dew point recognition

In case of transition from heater state LSH_OFF to LSH_POW_RISE the initial value for the effective heater voltage shall be determined by obtaining the oxygen sensor temperature at start (TEMP_INI_LS_UP[i]) from input data. This shall then be used to obtain the appropriate effective heater starting voltage from map IP_V_EFC_INI_ LSH_UP. In the state LSH_POW_RISE, the effective heater voltage shall be incremented with the gradient of C_V_EFC_INC_LSH_UP. The continuation of the heating gradient shall depend on whether the dew point at sensor location has been recognized.

Should the dew point not have been exceeded (LV_TEMP_DEW_LS_UP[i] = 0) but timer T_POW_RISE_LSH_UP[i] has equalled or exceeded IP_T_TEMP_THD_LSH_UP, it shall be assumed that condensed water splash at the oxygen sensor location may occur. In order to protect the oxygen sensor by keeping the temperature of the ceramic lower than the specified critical temperature, the heater power shall be reduced immediately by setting the effective heater voltage to C_V_EFC_RED_LSH_UP. In this case, the heater state shall be set to LSH_POW_RED (see above).

The reduced heating state shall persist until the dew point in the exhaust gas system has been recognized, i.e. until LV_TEMP_DEW_LS_UP[i] = 1, at which point the heater state shall return to LSH_POW_RISE. For further operation please refer to case 1.)

Should the initial exhaust gas temperature at start have been determined to exceed the dew point (LV_TEMP_DEW_LS_UP[i] = 1), the heater control shall not enter the reduced heater power state but continue to incrementally increase the heater power within the LSH_POW_RISE state.

Dependent on whether closed loop control is permitted, the heater management shall enter either LSH_POW_CTL or LSH_POW_FALL. Should closed loop be permitted (LV_INH_LSH_CTL_CLL_LSH_UP[i] = 0) and the ceramic temperature reference be valid (LV_V_REF_VLD_R_IT_LS_UP[i] = 1) and ceramic temperature exceed the set temperature (TTIP_MES_LS_UP[i] ≥ C_TTIP_SP_LS_UP[i]) then the state machine shall change to LSH_POW_CTL and the timer shall be frozen. This state change shall permit the controller to take over heating when the operative readiness has been detected in order to prevent under- & overshoot of the ceramic temperature. Should the conditional operative readiness be set (LV_LS_UP_READY_CDN[i] = 1) then the state machine shall also enter state LSH_POW_CTL and stop the timer. This shall permit the heater stability diagnosis to be carried out even when the heater is sufficiently defective to prevent the set temperature from being reached. Should the closed loop controller be inhibited or the ceramic temperature calibration be faulty, and T_POW_RISE_LSH_UP[i] be determined to have equalled or exceeded the value C_T_MAX_POW_RISE_LSH_UP (irrespective of the dew point having been exceeded or not), the heater state shall change to LSH_POW_FALL and the timer shall be frozen.

In the LSH_POW_FALL state, the effective heater voltage shall be reduced with a gradient of C_V_EFC_DEC_LSH_UP until such time that it falls below or equals the value set for the open loop control at the current engine operating point. For cylinder bank 1, this value is determined by map IP_V_EFC_CTL_LSH_UP alone. For cylinder bank 2, this value is determined by IP_V_EFC_CTL_LSH_UP * IP_FAC_V_EFC_TEG_LSH_UP. The latter factor shall be necessary to accommodate differences in heat transfer rates between cylinder banks at the oxygen sensor location, particularly for transversal mounted V6-engines. At such time the effective heater voltage falls below or equals the respective threshold, the heater state shall change to LSH_POW_CTL.

Should, however, the Boolean flag LV_TEMP_DEW_LS_UP[i] be reset whilst in the state LSH_POW_FALL, then the closed loop controller terms shall be reset, the Boolean flags to indicate closed loop control and effective heater voltage limitation shall be reset and the heater control shall resume at state LSH_POW_RED following the procedures outlined above.

The control of the effective heater voltage may be carried out in open loop or closed loop mode. Unless either the closed loop control inhibit bit (LV_INH_LSH_CTL_CLL_LSH_UP[i]) is set or the reference voltage validity bit (LV_V_REF_VLD_R_IT_LS_UP[i]) is reset, the effective heater voltage shall be controlled in closed loop mode. Closed loop control conditions met shall be indicated by setting LV_LSH_CTL_CLL_LSH_UP[i]. Should at any time one of the above conditions no longer be met, the heater controller shall revert to open loop control, reset the closed loop variables and reset LV_LSH_CTL_CLL_LSH_UP[i].

In the open loop heater control mode, the effective heater voltage shall be obtained from map IP_V_EFC_CTL_LSH_UP and is dependent upon the engine speed and the modelled exhaust gas temperature. For cylinder bank 2, an additional multiplicative factor (IP_FAC_V_EFC_TEG_LSH_UP) may be required for the reasons outlined above.

In the closed loop heater control mode, a PI controller, that affects the open loop effective heater voltage in an additive manner, shall regulate the effective heater voltage such that the temperature specified by C_TTIP_SP_LS_UP[i] is maintained. The closed loop control may be further described as follows:

The current temperature of the oxygen sensor (TTIP_MES_LS_UP[i]) shall be obtained and the difference between this temperature and the set temperature computed to obtain the input variable to the controller (TTIP_MES_DIF_LS_UP[i]). Note that this variable shall be computed in both open and closed loop control modes but shall only be used in the closed loop control mode.

The controller shall then determine the appropriate additive voltage (V_EFC_CTL_ADD_LSH_UP[i]) by computing the P and I terms separately and then adding the two terms.

The P term (V_EFC_CTL_P_LSH_UP[i]) shall calculate a voltage proportional to the temperature difference and thus converting a temperature delta into voltage. The P term shall only be calculated when the absolute temperature difference (TTIP_MES_DIF_LS_UP[i]) equals or exceeds a constant (C_TTIP_DIF_MIN_CTL_P_LS_UP). This shall be achieved via the map IP_FAC_xxx_P_LSH_UP_CTL, where xxx stands for either POS or NEG dependent on the sign of the temperature difference. The map values are dependent on engine speed and exhaust gas temperature. A further dimensionless constant C_FAC_CTL_P_LSH_UP shall permit global parameter changes to be made.

The I term (V_EFC_CTL_I_LSH_UP[i]) shall calculate an integral of a voltage that shall be proportional to the temperature difference and thus converting a temperature delta into voltage. This shall be achieved via the map IP_FAC_xxx_I_LSH_UP_CTL, where xxx stands for either POS or NEG dependent on the sign of the temperature difference. The map values are dependent on engine speed and exhaust gas temperature. A further dimensionless constant C_FAC_CTL_I_LSH_UP shall permit global parameter changes to be made.

Once the additive voltage has been determined, this shall then be added to the open loop control voltage along with a calibration system constant (C_V_EFC_AS_LSH_UP) to permit a global change in the total effective heater voltage to be made. Thus the calculated effective heater voltage (V_EFC_CLC_LSH_UP[i]) shall be computed from the open loop value, the closed loop additive term and the global correction value.

The closed loop control computations shall be suspended should either the Boolean flag LC_LSH_CTL_CLL_INH_LSH_UP[i] be set or the ceramic temperature no longer be valid, for example due to an active reference voltage measurement, as indicated by LV_TTIP_MES_VLD_LS_UP[i] being reset. This shall freeze the P and I terms at their last calculated value. Upon revoking the suspension, the controller shall continue to compute from where the point immediately prior to the suspension, i.e. shall not start anew. Furthermore should the sum of the components used to calculate the effective heater voltage be negative, LV_V_EFC_LIM_BOL_LSH_UP[i] shall be set. This shall prevent the controller from decreasing the effective heater voltage further but not prevent it from increasing the voltage. When the sum of the components is no longer negative, LV_V_EFC_LIM_BOL_LSH_UP[i] shall be reset. Similarly, should either of the overvoltage limitation Boolean flags LV_V_EFC_LIM_TOL_LSH_UP[i] & LV_V_EFC_LIM_MAX_LSH_UP[i] be set, as described later, the controller shall be prevented from increasing the effective heater voltage further but not prevent it from decreasing the voltage. When the overvoltage flags are no longer set and the other applicable conditions are met, the controller shall resume normal operation.

Should, however, the Boolean flag LV_TEMP_DEW_LS_UP[i] be reset whilst in the state LSH_POW_CTL, then the closed loop controller terms shall be reset, the Boolean flags to indicate closed loop control and effective heater voltage limitation shall be reset and the heater control shall resume at state LSH_POW_RED following the procedures outlined above.

Should at any time the battery voltage VB exceed the threshold C_VB_MAX_PROT_LSH_UP, as determined by the state of logical variable LV_V_EFC_LIM_PROT_VB_LSH_UP[i], the oxygen sensor heater power shall be reduced by reducing the effective heater voltage used to calculate the driver duty cycle to C_V_EFC_PROT_VB_LSH_UP. The heater state shall change to LSH_VB_PROT on the next recurrence of the state machine and the effective heater voltage set to the aforementioned limit. This shall prevent damage to the sensor due to excessive heater voltages e.g. due to jump start from 24 V.

Should the battery voltage recover when in the state LSH_VB_PROT, as indicated by LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0, T_POW_RISE_LSH_UP[i] shall be reset, V_EFC_LSH_UP[i] shall be initialised to IP_V_EFC_INI_LSH_UP calculated after the start, the closed loop controller terms shall be reset, the Boolean flags to indicate closed loop control and effective heater voltage limitation shall be reset and the heater control shall resume at state LSH_POW_RISE following the procedures outlined above.

Short term voltage excursions shall cause the heater power to be limited but will only cause the state machine to change state should the voltage excursion persist until the next recurrence of the state machine. The state machine shall not be able to enter the LSH_VB_PROT state directly from LSH_OFF but shall be required to pass through LSH_POW_RISE.

The sensor shall also be protected from over-temperature. This shall be achieved by evaluating the value of the modelled exhaust gas temperature TEG_DYN_LS_UP[i] in all states other than LSH_OFF. Should the exhaust gas temperature exceed the threshold C_TEG_DYN_MAX_LSH_UP, the calculated effective heater voltage shall be reduced to 0, the state shall change to LSH_TEMP_PROT thus suspending closed loop control, if active.

Should the exhaust gas temperature fall below the threshold, the state shall change dependent on the state active immediately prior to the over-temperature protection. If the state was LSH_POW_CTL, the state machine shall return to this state and the effective heater voltage shall take the new value as computed by the open / closed loop control. The closed loop controller shall not be started anew. If the state was any other than LSH_POW_CTL, the state shall change to LSH_POW_RISE. In this case T_POW_RISE_LSH_UP[i] shall be reset, V_EFC_LSH_UP[i] shall be initialised to IP_V_EFC_INI_LSH_UP calculated after the start, the closed loop controller terms shall be reset, the Boolean flags to indicate closed loop control and effective heater voltage limitation shall be reset and the heater control shall resume at state LSH_POW_RISE following the procedures outlined above.

Furthermore, in order to protect the sensor heater against long term high voltage stress it shall be necessary to limit the effective heater voltage should the value of V_EFC_CLC_LSH_UP[i] equal or exceed the threshold C_V_EFC_TOL_LSH_UP.

In this instance, the timer T_V_EFC_TOL_LSH_UP[i] shall be started. Should the overvoltage condition persist, such that this timer equals or exceeds the threshold C_T_MAX_V_EFC_TOL_LSH_UP, timer T_V_EFC_LIM_LSH_UP[i] shall be started and the Boolean flag LV_V_EFC_LIM_TOL_LSH_UP[i] shall be set. This flag shall prevent the controller from increasing effective the voltage further. If however the value of V_EFC_CLC_LSH_UP[i] should fall below the threshold C_V_EFC_TOL_LSH_UP at any time, LV_V_EFC_LIM_TOL_LSH_UP[i] shall be reset.

Once timer T_V_EFC_LIM_LSH_UP[i] has been started, Boolean flag LV_V_EFC_LIM_TOL_LSH_UP[i] shall be set or reset depending on the value of V_EFC_CLC_LSH_UP[i]. Once the timer T_V_EFC_LIM_LSH_UP[i] equals or exceeds threshold C_T_MAX_V_EFC_LIM_LSH_UP, both timers shall be reset and timer T_V_EFC_TOL_LSH_UP[i] must exceed the threshold C_T_MAX_V_EFC_TOL_LSH_UP again in order to permit further limiting to take place.

To prevent damage to the sensor, V_EFC_CLC_LSH_UP[i] shall not be permitted to exceed the lower of the values C_V_EFC_MAX_LSH_UP, C_V_EFC_MAX_ST_LSH_UP or VB dependent on the value of T_POW_RISE_LSH_UP[i].

If T_POW_RISE_LSH_UP[i] equals or exceeds the constant C_T_MAX_V_EFC_MAX_ST_LSH_UP STEP or conditions regarding T_AST and STATE_LSH_UP[i] then V_EFC_CLC_LSH_UP[i] shall be checked against VB & C_V_EFC_MAX_LSH_UP. If either of the values is exceeded then Boolean flag LV_V_EFC_LIM_MAX_LSH_UP[i] shall be set and V_EFC_CLC_LSH_UP[i] limited to the lower of VB & C_V_EFC_MAX_LSH_UP otherwise LV_V_EFC_LIM_MAX_LSH_UP[i] shall be reset.

If T_POW_RISE_LSH_UP[i] is lower then constant C_T_MAX_V_EFC_MAX_ST_LSH_UP then V_EFC_CLC_LSH_UP[i] shall be checked against VB & C_V_EFC_MAX_ST_LSH_UP. If either of the values is exceeded then Boolean flag LV_V_EFC_LIM_MAX_LSH_UP[i] shall be set and V_EFC_CLC_LSH_UP[i] limited to the lower of VB & C_V_EFC_MAX_ST_LSH_UP otherwise LV_V_EFC_LIM_MAX_LSH_UP[i] shall be reset.

The flag LV_V_EFC_LIM_MAX_LSH_UP[i] shall prevent the controller from increasing effective the voltage further and permit the final effective heater voltage to be limited, as described later.

Should timer T_V_EFC_LIM_LSH_UP[i] be unequal to zero, i.e. overvoltage no longer tolerated as it has been present for certain time, V_EFC_LSH_UP[i] shall be the lower of the values V_EFC_CLC_LSH_UP[i] & C_V_EFC_LIM_LSH_UP otherwise V_EFC_LSH_UP[i] shall be set to V_EFC_CLC_LSH_UP[i].

Once the effective heater voltage has been determined throughout the various heater states other than LSH_OFF, it shall be used to compute the PWM duty cycle (LSHPWM_UP[i]) that shall control the appropriate heater driver. The heater power shall be corrected to take into account deviations in the measured battery voltage and limited to the range bounded by NC_LSHPWM_TOL_LSH_UP & NC_LSHPWM_BOL_LSH_UP. The evaluation of excess battery voltage shall be determined at the same recurrence as the computation of the duty cycle as described above. The PWM value may also be modified by use of constant C_FAC_V_EFC_AS_LSH_UP.

Should at any time an inhibit flag be set (LV_INH_LSH_UP[i] = 1), the corresponding oxygen sensor heater shall be switched off and the STATE_LSH_UP[i] set to LSH_OFF. This shall allow project specific application conditions to turn off oxygen sensor heating according to project philosophy. Should the heater function be inhibited, then the PWM duty cycle shall be set to input LSHPWM_EXT_LS_UP[i]. On a transition of the inhibit bit LV_INH_LSH_UP[i] from 1 → 0, the heater function shall start from anew.

Forced lambda probe heating before engine start (for application purpose)

Additionally to normal sensor heating after engine start setting LC_LSH_UP_MAN_ACT[i] = 1 allows forced pre-heating of the lambda sensor before engine start. The aim is to ramp up the sensor element to operating temperature in order to achieve a reliable lambda sensor signal already at the time when the engine is started. It has to be emphasized that this method covers a high risk for sensor damage by water splash as operating temperature is already reached before dew-point recognition. Therefore this feature shall only be used for application purpose.

If forced heating is activated, i.e. condition 15 being fulfilled, the state machine shall enter the state LSH_POW_RISE and a heater voltage C_V_EFC_INI_LSH_UP_MAN_ACT shall be applied. Before engine start no forced convection and consequently no lambda sensor cooling by the exhaust gas is present. Therefore the initial heater voltage may be reduced to C_V_EFC_INI_LSH_UP_MAN_ACT in order to prevent excessive temperature gradients and overheating. Additionally the STATE_LSH_UP[i] shall be set to LSH_POW_RISE, the timer T_POW_RISE_LSH_UP[i] shall be incremented and the status flag LV_LSH_UP_MAN_ACT[i] indicating activated forced heating shall be set. Forced pre-heating activation by evaluating the transition of LC_LSH_UP_MAN_ACT[i] from 0 to 1 ensures that forced heating may not become active in field application even if LC_LSH_UP_MAN_ACT[i] remains 1 for the final series data.

In the state LSH_POW_RISE the lambda sensor heater voltage shall further be increased with the same voltage ramp as for inactive forced heating. It should be considered that too high ramp values may lead to excessive temperature gradients. The heater voltage during this initial phase is limited to C_V_EFC_MAX_LSH_UP_MAN_ACT. Resetting LC_LSH_UP_MAN_ACT[i] shall also reset LV_LSH_UP_MAN_ACT[i]. If T_POW_RISE_LSH_UP[i] exceeds the threshold C_T_MAX_CDN_LSH_UP_MAN_ACT the state machine shall change to state LSH_POW_CTL.

In the state LSH_POW_CTL while forced heating is active the element temperature shall be controlled in open loop. The corresponding effective heater voltage is derived from the same map IP_V_EFC_CTL_LSH_UP as for inactive forced heating, reduced by the factor C_FAC_V_EFC_LSH_UP_MAN_ACT. If forced heating is activated the lambda sensor protection states LSH_VB_PROT and LSH_TEMP_PROT may be reached. Should the corresponding conditions no longer be met, the state LSH_POW_RISE shall be re-entered.

Application conditions:

Activation:

At any engine operating state.

Deactivation:

-

Time recurrence:

The state machine, sensor protection and effective heater voltage definition shall be carried out once every 100 ms.

The determination of the oxygen sensor heater driver duty cycle and associated excessive battery voltage protection shall be carried out once every 10 ms.

Initialisation:

The following variable initialisation shall take place at LV_IGK = 0→1:

STATE_LSH_UP[i] = LSH_OFF
T_POW_RISE_LSH_UP[i] = 0
T_V_EFC_TOL_LSH_UP[i] = 0
T_V_EFC_LIM_LSH_UP[i] = 0
LSHPWM_UP[i] = 0
V_EFC_CTL_P_LSH_UP[i] = 0
V_EFC_CTL_I_LSH_UP[i] = 0
V_EFC_CTL_ADD_LSH_UP[i] = 0
V_EFC_LSH_UP[i] = 0
V_EFC_CLC_LSH_UP[i] = 0
LV_LSH_CTL_CLL_LSH_UP[i] = 0
LV_LSH_UP_MAN_ACT[i] = 0
LV_V_EFC_LIM_BOL_LSH_UP[i] = 0
LV_V_EFC_LIM_TOL_LSH_UP[i] = 0
LV_V_EFC_LIM_MAX_LSH_UP[i] = 0
LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0
TTIP_MES_DIF_LS_UP[i] = 0

At reset:

ACTION_INFR_SetPwmLshUp(

IN i

IN LSHPWM_UP[i])

NOTE: Projects not making use of the inhibit flags LV_INH_LSH_UP[i] & LV_INH_LSH_CTL_CLL_LSH_UP[i] shall ensure that they are initialised with and remain 0, thus never disabling the oxygen sensor heater function or thus never disabling the oxygen sensor heater closed loop control function respectively

Formula section:

The following formula shall be evaluated before the state machine is calculated:

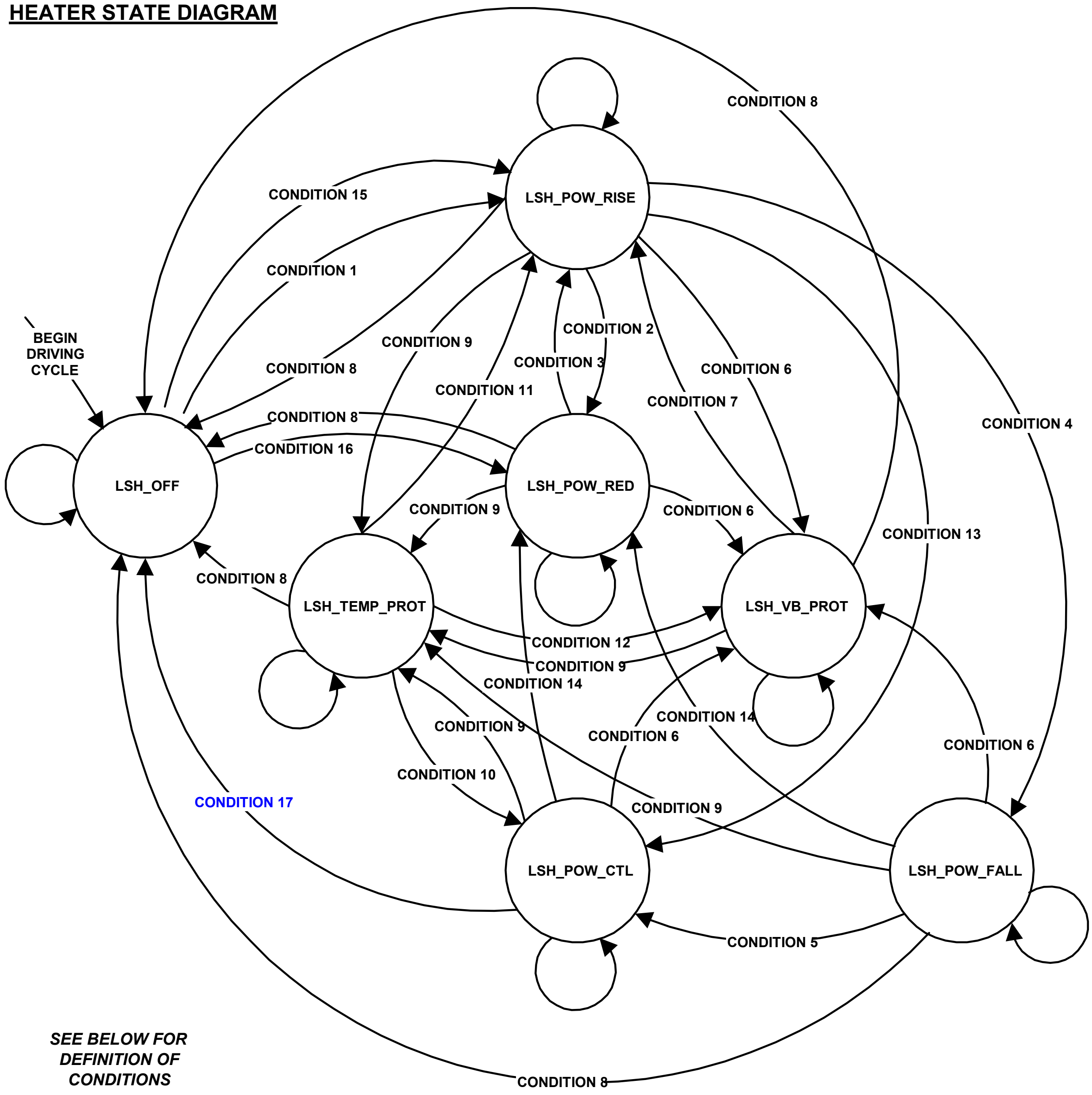
If LC_LSH_UP_MAN_ACT[i] = 1→0
then LV_LSH_UP_MAN_ACT[i] = 0
endif

16.19.1 Heater management state machine

The state machine shall remain in its current state and carry out the actions specified to occur within that state once per recurrence unless otherwise specified. The state machine shall only move to another state when one of the conditions has been determined to be met.

Note: The priorities of the conditions to change between states shall be defined by the order in which these conditions are listed within the appropriate state as described below.

HEATER STATE DIAGRAM



SEE BELOW FOR DEFINITION OF CONDITIONS

Figure 2: Heater management state diagram

STATE_LSH_UP[i] "LSH_OFF"

Actions:

T_POW_RISE_LSH_UP[i] = 0
T_V_EFC_TOL_LSH_UP[i] = 0
T_V_EFC_LIM_LSH_UP[i] = 0
V_EFC_CTL_P_LSH_UP[i] = 0
V_EFC_CTL_I_LSH_UP[i] = 0
V_EFC_CTL_ADD_LSH_UP[i] = 0
V_EFC_LSH_UP[i] = 0
V_EFC_CLC_LSH_UP[i] = 0
LV_LSH_CTL_CLL_LSH_UP[i] = 0
LV_LSH_UP_MAN_ACT[i] = 0
LV_V_EFC_LIM_BOL_LSH_UP[i] = 0
LV_V_EFC_LIM_TOL_LSH_UP[i] = 0
LV_V_EFC_LIM_MAX_LSH_UP[i] = 0
LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0
TTIP_MES_DIF_LS_UP[i] = 0

Note: The above listed actions must be carried out at least once when the state has been entered for the first time or from any other state!

Condition 1:"LSH_OFF to LSH_POW_RISE"

LV_ST_END = 1 **&**
LV_INH_LSH_UP[i] = 0 **&**
(LV_TEMP_DEW_LS_UP[i] = 1 **or**
LC_TEMP_DEW_LSH_UP = 0)

Transition actions:

V_EFC_CLC_LSH_UP[i] = IP_V_EFC_INI_LSH_UP (TEMP_INI_LS_UP[i])
Timer T_POW_RISE_LSH_UP[i] shall be started
STATE_LSH_UP[i] = LSH_POW_RISE

Condition 16:"LSH_OFF to LSH_POW_RED"

LV_ST_END = 1 **&**
LV_INH_LSH_UP[i] = 0 **&**
LV_TEMP_DEW_LS_UP[i] = 0 **&**
LC_TEMP_DEW_LSH_UP = 1

Transition actions:

V_EFC_CLC_LSH_UP[i] = C_V_EFC_RED_LSH_UP

Condition 15: Forced transition "LSH_OFF to LSH_POW_RISE" before engine start.

LV_ES = 1 **&** LC_LSH_UP_MAN_ACT[i] = 0→1

Transition actions:

V_EFC_CLC_LSH_UP[i] = C_V_EFC_INI_LSH_UP_MAN_ACT
Timer T_POW_RISE_LSH_UP[i] shall be started
STATE_LSH_UP[i] = LSH_POW_RISE
LV_LSH_UP_MAN_ACT[i] = 1

STATE_LSH_UP[i] "LSH_POW_RISE"

Actions:

```
If      LV_LSH_UP_MAN_ACT[i] = 1 & LV_ST_END = 0
then    $V\_EFC\_CLC\_LSH\_UP[i]_{N} = MIN(V\_EFC\_CLC\_LSH\_UP[i]_{N-1} + C\_V\_EFC\_INC\_LSH\_UP * 100\ ms, C\_V\_EFC\_MAX\_LSH\_UP\_MAN\_ACT)$
else    $V\_EFC\_CLC\_LSH\_UP[i]_{N} = V\_EFC\_CLC\_LSH\_UP[i]_{N-1} + C\_V\_EFC\_INC\_LSH\_UP * 100\ ms$
endif
```

Note: See section "General oxygen sensor voltage protection" for notes on limiting the applied heater voltage.

Condition 8:"LSH_POW_RISE to LSH_OFF"

(LV_ST_END = 0 **&** LV_LSH_UP_MAN_ACT[i] = 0) **or** LV_INH_LSH_UP[i] = 1
Transition actions:

STATE_LSH_UP[i] = LSH_OFF
V_EFC_LSH_UP[i] = 0

Condition 9:"LSH_POW_RISE to LSH_TEMP_PROT"

TEG_DYN_LS_UP[i] > C_TEG_DYN_MAX_LSH_UP **&**
LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0

Transition actions:

STATE_LSH_UP[i] = LSH_TEMP_PROT

Condition 6:"LSH_POW_RISE to LSH_VB_PROT"

LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_VB_PROT

Condition 2:"LSH_POW_RISE to LSH_POW_RED"

(T_POW_RISE_LSH_UP[i] ≥ IP_T_TEMP_THD_LSH_UP (TEMP_INI_LS_UP[i]) **&**
LV_TEMP_DEW_LS_UP[i] = 0) **&** LV_LSH_UP_MAN_ACT[i] = 0

Transition actions:

STATE_LSH_UP[i] = LSH_POW_RED

Condition 13:"LSH_POW_RISE to LSH_POW_CTL"

(LV_INH_LSH_CTL_CLL_LSH_UP[i] = 0 **&**
LV_V_REF_VLD_R_IT_LS_UP[i] = 1 **&**
TTIP_MES_LS_UP[i] ≥ C_TTIP_SP_LS_UP[i] - ID_TTIP_SP_DIF_LSH_UP **&**
LV_TTIP_MES_VLD_LS_UP[i] = 1) **or**
LV_LS_UP_READY_CDN[i] = 1 **or**
(LV_LSH_UP_MAN_ACT[i] = 1 **&**
T_POW_RISE_LSH_UP[i] ≥ C_T_MAX_CDN_LSH_UP_MAN_ACT)

Transition actions:

Stop timer T_POW_RISE_LSH_UP[i] and freeze value
STATE_LSH_UP[i] = LSH_POW_CTL

<u>Condition 4</u>:"LSH_POW_RISE to LSH_POW_FALL"

((LV_INH_LSH_CTL_CLL_LSH_UP[i] = 1 **or**
LV_V_REF_VLD_R_IT_LS_UP[i] = 0) **&**
T_POW_RISE_LSH_UP[i] ≥ C_T_MAX_POW_RISE_LSH_UP +
IP_T_TEMP_THD_LSH_UP (TEMP_INI_LS_UP[i])) **&** LV_LSH_UP_MAN_ACT[i] = 0

Transition actions:

Stop timer T_POW_RISE_LSH_UP[i] and freeze value
STATE_LSH_UP[i] = LSH_POW_FALL

STATE_LSH_UP[i] "LSH_POW_RED"

Actions:

V_EFC_CLC_LSH_UP[i] = C_V_EFC_RED_LSH_UP

<u>Condition 8</u>:"LSH_POW_RED to LSH_OFF"

(LV_ST_END = 0 **&** LV_LSH_UP_MAN_ACT[i] = 0) **or** LV_INH_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_OFF
V_EFC_LSH_UP[i] = 0

<u>Condition 9</u>:"LSH_POW_RED to LSH_TEMP_PROT"

TEG_DYN_LS_UP[i] > C_TEG_DYN_MAX_LSH_UP **&**
LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0

Transition actions:

STATE_LSH_UP[i] = LSH_TEMP_PROT

<u>Condition 6</u>:"LSH_POW_RED to LSH_VB_PROT"

LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_VB_PROT

<u>Condition 3</u>:"LSH_POW_RED to LSH_POW_RISE"

LV_TEMP_DEW_LS_UP[i] = 1

Transition actions:

V_EFC_CLC_LSH_UP[i] = IP_V_EFC_INI_LSH_UP (TEMP_INI_LS_UP[i]) *(same value as at start)* + C_V_EFC_STEP_LSH_UP
Re-initialise timer T_POW_RISE_LSH_UP[i] with IP_T_TEMP_THD_LSH_UP (TEMP_INI_LS_UP[i]) *(same value as at start)*
STATE_LSH_UP[i] = LSH_POW_RISE

Note: See section "General oxygen sensor voltage protection" for notes on limiting the applied heater voltage. The function shall prevent the effective heater voltage from becoming negative by limiting to 0 where necessary.

STATE_LSH_UP[i] "LSH_POW_FALL"

Actions:

$V\_EFC\_CLC\_LSH\_UP[i]_N = V\_EFC\_CLC\_LSH\_UP[i]_{N-1} - C\_V\_EFC\_DEC\_LSH\_UP * 100$ ms

Note: See section “General oxygen sensor voltage protection” for notes on limiting the applied heater voltage. The function shall prevent the effective heater voltage from becoming negative by limiting to 0 where necessary.

Condition 8:“LSH_POW_FALL to LSH_OFF”

(LV_ST_END = 0 **&** LV_LSH_UP_MAN_ACT[i] = 0) **or** LV_INH_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_OFF
V_EFC_LSH_UP[i] = 0

Condition 9:“LSH_POW_FALL to LSH_TEMP_PROT”

TEG_DYN_LS_UP[i] > C_TEG_DYN_MAX_LSH_UP **&**
LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0

Transition actions:

STATE_LSH_UP[i] = LSH_TEMP_PROT

Condition 6:“LSH_POW_FALL to LSH_VB_PROT”

LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_VB_PROT

Condition 14:“LSH_POW_FALL to LSH_POW_RED”

LV_TEMP_DEW_LS_UP[i] = 0

Transition actions:

T_V_EFC_TOL_LSH_UP[i] = 0
T_V_EFC_LIM_LSH_UP[i] = 0
V_EFC_CTL_P_LSH_UP[i] = 0
V_EFC_CTL_I_LSH_UP[i] = 0
V_EFC_CTL_ADD_LSH_UP[i] = 0
LV_LSH_CTL_CLL_LSH_UP[i] = 0
LV_V_EFC_LIM_BOL_LSH_UP[i] = 0
LV_V_EFC_LIM_TOL_LSH_UP[i] = 0
LV_V_EFC_LIM_MAX_LSH_UP[i] = 0
TTIP_MES_DIF_LS_UP[i] = 0
STATE_LSH_UP[i] = LSH_POW_RED

Condition 5:“LSH_POW_FALL to LSH_POW_CTL”

For Bank 1; i = 1

V_EFC_LSH_UP[i] ≤ IP_V_EFC_CTL_LSH_UP (N_32, TEG_DYN_LS_UP[i])

For Bank 2; i = 2

V_EFC_LSH_UP[i] ≤ IP_V_EFC_CTL_LSH_UP (N_32, TEG_DYN_LS_UP[i]) *
IP_FAC_V_EFC_TEG_LSH_UP

Transition actions:

STATE_LSH_UP[i] = LSH_POW_CTL

STATE_LSH_UP[i] “LSH_POW_CTL”

Actions:

Determine deviation between desired operating temperature and measured temperature:

TTIP_MES_DIF_LS_UP[i] = C_TTIP_SP_LS_UP[i] - TTIP_MES_LS_UP[i]

Note: TTIP_MES_DIF_LS_UP[i] shall be computed in both open loop and closed loop control modes.

Determine whether closed loop control permitted, if not revert to open loop control:

```
If   (LV_INH_LSH_CTL_CLL_LSH_UP[i] = 0) &
     (LV_V_REF_VLD_R_IT_LS_UP[i] = 1)
then Closed loop control:
         LV_LSH_CTL_CLL_LSH_UP[i] = 1
         Case A: Closed loop control
else Open loop control:
         LV_LSH_CTL_CLL_LSH_UP[i] = 0

         PI controller shall be re-initialised, i.e.:
         V_EFC_CTL_P_LSH_UP[i] = 0
         V_EFC_CTL_I_LSH_UP[i] = 0
         V_EFC_CTL_ADD_LSH_UP[i] = 0
endif.
```

***Case A**: Closed loop control:*

Determine conditions for computation of closed loop control variables:

```
If   (LC_LSH_CTL_CLL_INH_LSH_UP[i] = 1) or
     (LV_TTIP_MES_VLD_LS_UP[i] = 0)
then General suspension and re-initialisation of closed loop control computation, i.e.:
         V_EFC_CTL_ADD_LSH_UP[i] = 0
         V_EFC_CTL_P_LSH_UP[i] = 0
         V_EFC_CTL_I_LSH_UP[i] = 0
else Normal closed loop control computation:
         Case B: Closed loop control & Not generally suspended:
endif.
```

***End Case A**: Closed loop control:*

***Case B**: Closed loop control & Not generally suspended:*

```
If   (TTIP_MES_LS_UP[i] > C_TTIP_SP_LS_UP[i])    i.e. Temperature difference negative
then     Compute I terms of controller:
         V_EFC_CTL_I_LSH_UP_TMP[i]$_{N}$ = V_EFC_CTL_I_LSH_UP[i]$_{N-1}$ +
         TTIP_MES_DIF_LS_UP[i] *
         IP_FAC_NEG_CTL_I_LSH_UP (N_32, TEG_DYN_LS_UP[i])*
         C_FAC_CTL_I_LSH_UP
         If      ( | TTIP_MES_DIF_LS_UP[i] | ≥ C_TTIP_DIF_MIN_CTL_P_LS_UP)
         then Compute P term of controller:
```

```
                V_EFC_CTL_P_LSH_UP_TMP[i] = TTIP_MES_DIF_LS_UP[i] *
                IP_FAC_NEG_CTL_P_LSH_UP (N_32, TEG_DYN_LS_UP[i])*
                C_FAC_CTL_P_LSH_UP
        else    V_EFC_CTL_P_LSH_UP_TMP[i] = 0
        endif
        if      (LV_V_EFC_LIM_BOL_LSH_UP[i] = 0) i.e. no lower limiting active
        then    V_EFC_CTL_I_LSH_UP[i]_N = V_EFC_CTL_I_LSH_UP_TMP[i]
                V_EFC_CTL_P_LSH_UP[i]_N = V_EFC_CTL_P_LSH_UP_TMP[i]
        else    V_EFC_CTL_I_LSH_UP[i]_N = MAX(V_EFC_CTL_I_LSH_UP[i]_N-1 ,
                                              V_EFC_CTL_I_LSH_UP_TMP[i])
                V_EFC_CTL_P_LSH_UP[i]_N = MAX(V_EFC_CTL_P_LSH_UP[i]_N-1 ,
                                              V_EFC_CTL_P_LSH_UP_TMP[i])
        endif
else    Temperature difference positive or zero; compute I term of controller:
        V_EFC_CTL_I_LSH_UP_TMP[i]_N = V_EFC_CTL_I_LSH_UP[i]_N-1 +
        TTIP_MES_DIF_LS_UP[i] *
        IP_FAC_POS_CTL_I_LSH_UP (N_32, TEG_DYN_LS_UP[i])*
        C_FAC_CTL_I_LSH_UP
        If      ( | TTIP_MES_DIF_LS_UP[i] | ≥ C_TTIP_DIF_MIN_CTL_P_LS_UP)
        then    Compute P term of controller:
                V_EFC_CTL_P_LSH_UP_TMP[i] = TTIP_MES_DIF_LS_UP[i] *
                IP_FAC_POS_CTL_P_LSH_UP (N_32, TEG_DYN_LS_UP[i])*
                C_FAC_CTL_P_LSH_UP
        else    V_EFC_CTL_P_LSH_UP_TMP[i] = 0
        endif
        if      LV_V_EFC_LIM_MAX_LSH_UP[i] = 0 &
                LV_V_EFC_LIM_TOL_LSH_UP[i] = 0 i.e. no upper limiting active
        then    V_EFC_CTL_I_LSH_UP[i]_N = V_EFC_CTL_I_LSH_UP_TMP[i]
                V_EFC_CTL_P_LSH_UP[i]_N = V_EFC_CTL_P_LSH_UP_TMP[i]
        else    V_EFC_CTL_I_LSH_UP[i]_N = MIN(V_EFC_CTL_I_LSH_UP[i]_N-1 ,
                                              V_EFC_CTL_I_LSH_UP_TMP[i])
                V_EFC_CTL_P_LSH_UP[i]_N = MIN(V_EFC_CTL_P_LSH_UP[i]_N-1 ,
                                              V_EFC_CTL_P_LSH_UP_TMP[i])
        endif
endif
```

Note: In cases where the effective heater voltage has been limited as shown by the state of the Boolean flag LV_V_EFC_LIM_BOL_LSH_UP[i], the controller shall only be permitted to increase the effective voltage. In the case of limiting via Boolean flags LV_V_EFC_LIM_TOL_LSH_UP[i] & LV_V_EFC_LIM_MAX_LSH_UP[i] the closed loop control shall only be permitted to decrease the effective heater voltage. In all other cases, the controller shall effectively be suspended, i.e. P & I terms shall remain unchanged but applied, until the applicable limitation is revoked. Upon revoking the limitation, the function shall resume to compute the P & I terms according to the conditions specified above.

Combine P and I terms:
V_EFC_CTL_ADD_LSH_UP[i] =
V_EFC_CTL_P_LSH_UP[i] + V_EFC_CTL_I_LSH_UP[i]

Note: Although variables V_EFC_CTL_P_LSH_UP[i] and V_EFC_CTL_I_LSH_UP[i] are defined as words, the summation shall be carried out as a 16 bit addition and the resultant 16 bit value converted to an 8 bit value, by taking the high byte, and placed in V_EFC_CTL_ADD_LSH_UP[i]. This shall prevent controller from stopping short of the target temperature.

End Case B: *Closed loop control; Not suspended:*

Compute effective heater voltage from open loop control and closed loop delta voltages:

If LV_LSH_UP_MAN_ACT[i] = 1 **&** LV_ST_END = 0

then V_EFC_CLC_LSH_UP[i] = IP_V_EFC_CTL_LSH_UP (N_32, TEG_DYN_LS_UP[i]) *
C_FAC_V_EFC_LSH_UP_MAN_ACT

else

For Bank 1; i = 1

V_EFC_CLC_LSH_UP[i] = IP_V_EFC_CTL_LSH_UP (N_32, TEG_DYN_LS_UP[i]) +
V_EFC_CTL_ADD_LSH_UP[i] + C_V_EFC_AS_LSH_UP

Note: If (IP_V_EFC_CTL_LSH_UP + V_EFC_CTL_ADD_LSH_UP[i] + C_V_EFC_AS_LSH_UP) < 0 , i.e. would cause the effective heater voltage to be negative, V_EFC_CLC_LSH_UP[i] shall be limited to 0 and further reduction of the effective heater voltage via by the closed loop controller P & I terms shall be prevented by setting LV_V_EFC_LSH_UP_LIM_BOL[i], until the sum less than zero condition is no longer met at which point LV_V_EFC_LSH_UP_LIM_BOL[i] shall be reset.

For Bank 2; i = 2

V_EFC_CLC_LSH_UP[i] = IP_V_EFC_CTL_LSH_UP (N_32, TEG_DYN_LS_UP[i])*
IP_FAC_V_EFC_TEG_LSH_UP + V_EFC_CTL_ADD_LSH_UP[i] + C_V_EFC_AS_LSH_UP

Note: If (IP_V_EFC_CTL_LSH_UP* IP_FAC_V_EFC_TEG_LSH_UP + V_EFC_CTL_ADD_LSH_UP[i] + C_V_EFC_AS_LSH_UP) < 0 , i.e. would cause the effective heater voltage to be negative, V_EFC_CLC_LSH_UP[i] shall be limited to 0 and further reduction of the effective heater voltage via by the closed loop controller P & I terms shall be prevented by setting LV_V_EFC_LSH_UP_LIM_BOL[i], until the sum less than zero condition is no longer met at which point LV_V_EFC_LSH_UP_LIM_BOL[i] shall be reset.

endif

<u>Condition 17</u>:"LSH_POW_CTL to LSH_OFF"

(LV_ST_END = 0 **&** LV_LSH_UP_MAN_ACT[i] = 0 **&** LV_STST_STOP_CYC = 0) **or**
LV_INH_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_OFF
V_EFC_LSH_UP[i] = 0

<u>Condition 9</u>:"LSH_POW_CTL to LSH_TEMP_PROT"

TEG_DYN_LS_UP[i] > C_TEG_DYN_MAX_LSH_UP **&**
LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0

Transition actions:

STATE_LSH_UP[i] = LSH_TEMP_PROT

<u>Condition 6</u>:"LSH_POW_CTL to LSH_VB_PROT"

LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_VB_PROT

Condition 14:“LSH_POW_CTL to LSH_POW_RED”

LV_TEMP_DEW_LS_UP[i] = 0 **&** LV_LSH_UP_MAN_ACT[i] = 0

Transition actions:

T_V_EFC_TOL_LSH_UP[i] = 0
T_V_EFC_LIM_LSH_UP[i] = 0
V_EFC_CTL_P_LSH_UP[i] = 0
V_EFC_CTL_I_LSH_UP[i] = 0
V_EFC_CTL_ADD_LSH_UP[i] = 0
LV_LSH_CTL_CLL_LSH_UP[i] = 0
LV_V_EFC_LIM_BOL_LSH_UP[i] = 0
LV_V_EFC_LIM_TOL_LSH_UP[i] = 0
LV_V_EFC_LIM_MAX_LSH_UP[i] = 0
TTIP_MES_DIF_LS_UP[i] = 0
STATE_LSH_UP[i] = LSH_POW_RED

STATE_LSH_UP[i] “LSH_VB_PROT”

Actions:

V_EFC_CLC_LSH_UP[i] = C_V_EFC_PROT_VB_LSH_UP

Condition 8:“LSH_VB_PROT to LSH_OFF”

(LV_ST_END = 0 **&** LV_LSH_UP_MAN_ACT[i] = 0) **or** LV_INH_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_OFF
V_EFC_LSH_UP[i] = 0

Condition 9:“LSH_VB_PROT to LSH_TEMP_PROT”

TEG_DYN_LS_UP[i] > C_TEG_DYN_MAX_LSH_UP **&**
LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0

Transition actions:

STATE_LSH_UP[i] = LSH_TEMP_PROT

Condition 7:“LSH_VB_PROT to LSH_POW_RISE”

LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0

Transition actions:

```
T_POW_RISE_LSH_UP[i] = 0
If      LV_LSH_UP_MAN_ACT[i] = 0
then V_EFC_CLC_LSH_UP[i] = IP_V_EFC_INI_LSH_UP (TEMP_INI_LS_UP[i])
else  V_EFC_CLC_LSH_UP[i] = C_V_EFC_INI_LSH_UP_MAN_ACT
endif
```

T_V_EFC_TOL_LSH_UP[i] = 0
T_V_EFC_LIM_LSH_UP[i] = 0
V_EFC_CTL_P_LSH_UP[i] = 0
V_EFC_CTL_I_LSH_UP[i] = 0
V_EFC_CTL_ADD_LSH_UP[i] = 0
LV_LSH_CTL_CLL_LSH_UP[i] = 0
LV_V_EFC_LIM_BOL_LSH_UP[i] = 0
LV_V_EFC_LIM_TOL_LSH_UP[i] = 0
LV_V_EFC_LIM_MAX_LSH_UP[i] = 0
TTIP_MES_DIF_LS_UP[i] = 0
STATE_LSH_UP[i] = LSH_POW_RISE

STATE_LSH_UP[i] “LSH_TEMP_PROT”

Actions:

V_EFC_CLC_LSH_UP[i] = 0

Condition 8:“LSH_TEMP_PROT to LSH_OFF”

(LV_ST_END = 0 **&** LV_LSH_UP_MAN_ACT[i] = 0) **or** LV_INH_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_OFF
V_EFC_LSH_UP[i] = 0

Condition 12:“LSH_TEMP_PROT to LSH_VB_PROT”

LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 1

Transition actions:

STATE_LSH_UP[i] = LSH_VB_PROT

Condition 10:“LSH_TEMP_PROT to LSH_POW_CTL”

TEG_DYN_LS_UP[i] ≤ C_TEG_DYN_MAX_LSH_UP **&**
Previous state of STATE_LSH_UP[i] = LSH_POW_CTL

Transition actions:

STATE_LSH_UP[i] = LSH_POW_CTL

Condition 11:“LSH_TEMP_PROT to LSH_POW_RISE”

TEG_DYN_LS_UP[i] ≤ C_TEG_DYN_MAX_LSH_UP **&**
Previous state of STATE_LSH_UP[i] ≠ LSH_POW_CTL

Transition actions:

T_POW_RISE_LSH_UP[i] = 0
If LV_LSH_UP_MAN_ACT[i] = 0
then V_EFC_CLC_LSH_UP[i] = IP_V_EFC_INI_LSH_UP (TEMP_INI_LS_UP[i])
else V_EFC_CLC_LSH_UP[i] = C_V_EFC_INI_LSH_UP_MAN_ACT
endif

```
T_V_EFC_TOL_LSH_UP[i] = 0
T_V_EFC_LIM_LSH_UP[i] = 0
V_EFC_CTL_P_LSH_UP[i] = 0
V_EFC_CTL_I_LSH_UP[i] = 0
V_EFC_CTL_ADD_LSH_UP[i] = 0
LV_LSH_CTL_CLL_LSH_UP[i] = 0
LV_V_EFC_LIM_BOL_LSH_UP[i] = 0
LV_V_EFC_LIM_TOL_LSH_UP[i] = 0
LV_V_EFC_LIM_MAX_LSH_UP[i] = 0
TTIP_MES_DIF_LS_UP[i] = 0
STATE_LSH_UP[i] = LSH_POW_RISE
```

Note: Should none of the conditions have been determined to be met, the state machine shall remain in the same state.

16.19.2 Oxygen sensor heater voltage protection

Prolonged exposure to heater voltages that exceed the absolute maximum ratings specified shall be prevented. The preventative measures shall be applicable to all heater states other than LSH_OFF.

```
If      STATE_LSH_UP[i] ≠ LSH_OFF
then    If      (V_EFC_CLC_LSH_UP[i] ≥ C_V_EFC_TOL_LSH_UP)
        then    If      (T_V_EFC_TOL_LSH_UP[i] ≥ C_T_MAX_V_EFC_TOL_LSH_UP)
                then    If      T_V_EFC_LIM_LSH_UP[i] ≥
                                        C_T_MAX_V_EFC_LIM_LSH_UP
                        then    LV_V_EFC_LIM_TOL_LSH_UP[i] = 0
                                Reset timer T_V_EFC_LIM_LSH_UP[i]
                                Reset timer T_V_EFC_TOL_LSH_UP[i]
                        else    LV_V_EFC_LIM_TOL_LSH_UP[i] = 1
                                Increment timer T_V_EFC_LIM_LSH_UP[i]
                        endif
                else    LV_V_EFC_LIM_TOL_LSH_UP[i] = 0
                        Reset timer T_V_EFC_LIM_LSH_UP[i]
                        Increment timer T_V_EFC_TOL_LSH_UP[i]
                endif
        else    LV_V_EFC_LIM_TOL_LSH_UP[i] = 0

                If      T_V_EFC_LIM_LSH_UP[i] = 0
                or      T_V_EFC_LIM_LSH_UP[i] ≥
                                C_T_MAX_V_EFC_LIM_LSH_UP
                then    Reset timer T_V_EFC_LIM_LSH_UP[i]
```

```
                        Reset timer T_V_EFC_TOL_LSH_UP[i]
                else    Increment timer T_V_EFC_LIM_LSH_UP[i]
                endif
        endif
        If      T_POW_RISE_LSH_UP[i] ≥ C_T_MAX_V_EFC_MAX_ST_LSH_UP
        or      (T_AST ≥ C_T_AST_MAX_LSH_UP
        and     STATE_LSH_UP[i] = "LSH_POW_CTL")
        then    If      V_EFC_CLC_LSH_UP[i] > VB
                or      V_EFC_CLC_LSH_UP[i] > C_V_EFC_MAX_LSH_UP
                then    LV_V_EFC_LIM_MAX_LSH_UP[i] = 1
                        V_EFC_CLC_LSH_UP[i] = MIN (VB, C_V_EFC_MAX_LSH_UP)
                else    LV_V_EFC_LIM_MAX_LSH_UP[i] = 0
                endif
        else    If      V_EFC_CLC_LSH_UP[i] > VB
                or      V_EFC_CLC_LSH_UP[i] > C_V_EFC_MAX_ST_LSH_UP
                then    LV_V_EFC_LIM_MAX_LSH_UP[i] = 1
                        V_EFC_CLC_LSH_UP[i] =
                                        MIN (VB, C_V_EFC_MAX_ST_LSH_UP)
                else    LV_V_EFC_LIM_MAX_LSH_UP[i] = 0
                endif
        endif
        If      T_V_EFC_LIM_LSH_UP[i] <> 0
        then    V_EFC_LSH_UP[i] =
                                MIN (V_EFC_CLC_LSH_UP[i], C_V_EFC_LIM_LSH_UP)
        else    V_EFC_LSH_UP[i] = V_EFC_CLC_LSH_UP[i]
        endif
endif
```

16.19.3 Oxygen sensor heater driver duty cycle computation

The PWM duty cycle for the appropriate heater driver shall continually be calculated in all heater states at the specified recurrence.

```
If    (STATE_LSH_UP[i] = LSH_OFF & LC_LSH_UP_MAN_ACT[i] = 0)
then        LSHPWM_UP[i] = LSHPWM_EXT_LS_UP[i]
else        If      (VB ≥ C_VB_MAX_PROT_LSH_UP)
            then    LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 1
                    LSHPWM_UP[i] =      $C\_V\_EFC\_PROT\_VB\_LSH\_UP^2 *100 / VB_N^2$
            else    LV_V_EFC_LIM_PROT_VB_LSH_UP[i] = 0
                    LSHPWM_UP[i] = $V\_EFC\_LSH\_UP[i]_N^2$ *
                                        $C\_FAC\_V\_EFC\_AS\_LSH\_UP *100 / VB_N^2$
                    If      LSHPWM_UP[i] ≥ NC_LSHPWM_TOL_LSH_UP
                    then    LSHPWM_UP[i] = NC_LSHPWM_TOL_LSH_UP
                    else    If      LSHPWM_UP[i] ≤ NC_LSHPWM_BOL_LSH_UP
                            then    LSHPWM_UP[i] = NC_LSHPWM_BOL_LSH_UP
                            else    LSHPWM_UP[i] = LSHPWM_UP[i]
                            endif
                    endif
            endif
endif.
ACTION_INFR_SetPwmLshUp(
                    IN      i
                    IN      LSHPWM_UP[i])
```

The value C_FAC_V_EFC_AS_LSH_UP is a correction signal generated in the application system.

NOTE: The recurrence of the heater driving duty cycle shall occur at the same recurrence as the Vbatt acquisition. This shall prevent short duration voltage drop-outs during engine start from causing long duration Vbatt corrections being made where the Vbatt has since recovered. This may otherwise cause excessive current to flow in the heater driver.

The computation of the duty cycle shall be checked to ensure that the result remains positive (i.e. ≥ 0) at all times and that now under- or overflows are caused by the values of the calibration system constants.

The multiplication of the corrected voltage ratio by 100 to obtain a unit of percent (%) may not necessarily be implemented in the SW but in the data bank definition of the variables.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_V_EFC_INI_LSH_UP	6	0...FFH	0...25.89843	0.1015625	[V]
LDPM_TEMP_INI_LS_UP_1_EGCP	6	0...FFFFH	-2048...2047.9375	0.0625	[°C]
Initial effective heater voltage					
IP_V_EFC_CTL_LSH_UP	6*6	0...FFH	0...25.89843	0.1015625	[V]
LDPM_N_32_1_EGCP	6	0...FFH	0...8160	32	[rpm]
LDPM_TEG_DYN_LS_UP_1_EGCP	6	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Open loop control effective heater voltage					
IP_T_TEMP_THD_LSH_UP	6	0...FFH	0...25.5	0.1	[s]
LDPM_TEMP_INI_LS_UP_1_EGCP	6	0...FFFFH	-2048...2047.9375	0.0625	[°C]
Duration until critical water splash sensor temperature reached					
IP_FAC_V_EFC_TEG_LSH_UP	6	0...FFH	0...3.98437	0.015625	[-]
LDP_VS_IP_FAC_V_EFC_TEG_LSH_UP	6	0...FFH	0...255	1	[km/h]
Factor for the heat transfer rate, bank 2 only					
C_V_EFC_INC_LSH_UP	1	0...FFH	0...1.625	6.37E-03	[V/100ms]
Effective heater voltage increment to raise temperature					
C_V_EFC_DEC_LSH_UP	1	0...FFH	0...1.625	6.37E-03	[V/100ms]
Effective heater voltage decrement to lower temperature					
C_V_EFC_RED_LSH_UP	1	0...FFH	0...25.89843	0.1015625	[V]
Reduced effective heater voltage during danger of water splash damage					
C_V_EFC_STEP_LSH_UP	1	80...7FH	-13...12.89843	0.1015625	[V]
Additional effective heater voltage step on transition from LSH_POW_RED to LSH_POW_RISE					
C_T_AST_MAX_LSH_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time after start at which step in max. effective heater voltage applied, if LSH_POW_CTL active (T_POW_RISE_LSH_UP[NC_CBK_EX_NR] stopped)					
C_T_MAX_POW_RISE_LSH_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Duration of open loop pre-heating in LSH_POW_RISE from dew-point to set-point temperature					
C_T_MAX_V_EFC_TOL_LSH_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum permitted duration where excessive effective heater voltage tolerated					
C_T_MAX_V_EFC_LIM_LSH_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum duration of effective heater voltage limiting					
C_T_MAX_V_EFC_MAX_ST_LSH_UP	1	0...FFH	0...25.5	0.1	[s]
Time after start at which step in max. effective heater voltage applied					
C_V_EFC_TOL_LSH_UP	1	0...FFH	0...25.89843	0.1015625	[V]
Effective heater voltage threshold above which voltage limiting may occur if threshold persistently exceeded					
C_V_EFC_LIM_LSH_UP	1	0...FFH	0...25.89843	0.1015625	[V]
Effective heater voltage limit under persistent excessive heater voltage condition					
C_VB_MAX_PROT_LSH_UP	1	0...FFH	0...25.89843	0.1015625	[V]
Maximum permitted battery voltage threshold, over which overvoltage detected					
C_V_EFC_PROT_VB_LSH_UP	1	0...FFH	0...25.89843	0.1015625	[V]
Effective heater voltage in the case of battery voltage overvoltage condition					
C_V_EFC_MAX_LSH_UP	1	0...FFH	0...25.89843	0.1015625	[V]
Absolute maximum permitted effective heater voltage after initial phase; threshold & limit					
C_V_EFC_MAX_ST_LSH_UP	1	0...FFH	0...25.89843	0.1015625	[V]
Absolute maximum permitted effective heater voltage during initial phase; threshold & limit					
C_TTIP_SP_LS_UP[NC_CBK_EX_NR]	1	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
Oxygen sensor set temperature for closed loop control					
IP_FAC_NEG_CTL_P_LSH_UP	6*6	0...FFH	0...0.255	0.001	[V/K]
LDPM_N_32_1_EGCP	6	0...FFH	0...8160	32	[rpm]
LDPM_TEG_DYN_LS_UP_1_EGCP	6	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Closed loop controller P-term factor for temperature too high; decrease heater voltage					
IP_FAC_POS_CTL_P_LSH_UP	6*6	0...FFH	0...0.255	0.001	[V/K]
LDPM_N_32_1_EGCP	6	0...FFH	0...8160	32	[rpm]
LDPM_TEG_DYN_LS_UP_1_EGCP	6	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
Closed loop controller P-term factor for temperature too low; increase heater voltage					
C_FAC_CTL_P_LSH_UP	1	0...FFH	0...6.348	2.49E-02	[-]
Multiplicative scaling factor for closed loop P-term					
C_TTIP_DIF_MIN_CTL_P_LS_UP	1	0...7FFFH	0...2047.9375	0.0625	[°C]
Minimum required temperature difference in order to calculate P term of controller					
IP_FAC_NEG_CTL_I_LSH_UP	6*6	0...FFH	0...0.255	0.001	[V/K]
LDPM_N_32_1_EGCP	6	0...FFH	0...8160	32	[rpm]
LDPM_TEG_DYN_LS_UP_1_EGCP	6	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Closed loop controller I-term factor for temperature too high; decrease heater voltage					
IP_FAC_POS_CTL_I_LSH_UP	6*6	0...FFH	0...0.255	0.001	[V/K]
LDPM_N_32_1_EGCP	6	0...FFH	0...8160	32	[rpm]
LDPM_TEG_DYN_LS_UP_1_EGCP	6	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Closed loop controller I-term factor for temperature too low; increase heater voltage					
ID_TTIP_SP_DIF_LSH_UP	4	8000...7FFFH	-2048...2047.9375	0.0625	[°C]
LDP_T_POW_RISE_LSH_UP_ID_TTIP	4	0...FFFFH	0...6553.5	0.1	[s]
Reducing value for TTIP threshold of transition conditions LSH_POW_RISE to LSH_POW_CTL					
C_FAC_CTL_I_LSH_UP	1	0...FFH	0...6.348	2.49E-02	[-]
Multiplicative scaling factor for closed loop I-term					
C_V_EFC_AS_LSH_UP	1	80...7FH	-13...12.89843	0.1015625	[V]
Effective heater voltage calibration system additive factor					
C_FAC_V_EFC_AS_LSH_UP	1	0...FFH	0...1.99218	0.0078125	[-]
Multiplicative correction factor to effective heater voltage					
C_TEG_DYN_MAX_LSH_UP	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Maximum permitted exhaust gas temperature above which LSH switched off					
LC_LSH_CTL_CLL_INH_LSH_UP[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Boolean constant to permit suspension of closed loop heater control					
LC_LSH_UP_MAN_ACT[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Boolean constant to permit forced lambda sensor heating before engine start					
C_FAC_V_EFC_LSH_UP_MAN_ACT	1	0...FFH	0...1.99218	0.0078125	[-]
Multiplicative correction factor to effective heater voltage with activated forced heating before engine start for state LSH_POW_CTL					
C_V_EFC_INI_LSH_UP_MAN_ACT	1	0...FFFFH	0...25.99960	0.3967e-3	[V]
Initial effective heater voltage when forced heating before engine start activated					
C_V_EFC_MAX_LSH_UP_MAN_ACT	1	0...FFFFH	0...25.99960	0.3967e-3	[V]
Absolute maximum permitted effective heater voltage during forced heating phase before engine start					
C_T_MAX_CDN_LSH_UP_MAN_ACT	1	0...FFFFH	0...6553.5	0.1	[s]
Time threshold for transition from state POW_RISE to POW_CTL with activated forced heating before engine start					
LC_TEMP_DEW_LSH_UP	1	0...1H	0...1	1	[-]
Logical calibration data to force LSH_OFF to LSH_POW_RED transition if no dew point recognized					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_LSHPWM_TOL_LSH_UP	1	0...FFH	0...99.60937	0.390625	[%]
Heater driver PWM duty cycle upper limit					
NC_LSHPWM_BOL_LSH_UP	1	0...FFH	0...99.60937	0.390625	[%]
Heater driver PWM duty cycle lower limit					

16.19.4 Application incidences for upstream oxygen sensor heater management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating project specific application conditions for heating not met					
LV_INH_LSH_CTL_CLL_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating project specific application conditions closed loop heater control not met					
TEMP_INI_LS_UP[NC_CBK_EX_NR]	V/O	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Oxygen sensor temperature at engine start					
LSHPWM_EXT_LS_UP[NC_CBK_EX_NR]	V/O	0...FFH	0...99.60937	0.390625	[%]
Heater driver PWM duty cycle – external adjustment					
LSHPWM_UP_MAN	V	0...FFH	0...99.60937	0.390625	[%]
Heater driver PWM duty cycle – manual adjustment via application system					
V_EFC_LSH_UP_MAN	V	0...FFH	0...25.89843	0.1015625	[V]
Effective output value of heater voltage via application system					

Input data:

LV_IGK	LSHPWM_UP_EXT_ADJ[NC_CBK_EX_NR]	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]
NC_CBK_EX_NR	LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]
LV_ST_END	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]
LV_VAR_LSH_UP	LV_LSHPWM_UP_EXT_ADJ[NC_CBK_EX_NR]	TEG_WALL_CAT_UP_MDL[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinderbanks.

i = 1 ... NC_CBK_EX_NR

Description:

The Boolean constant LC_LSH_UP[i] shall permit the appropriate oxygen sensor to be heated. When set, the Boolean inhibit flag LV_INH_LSH_UP[i] shall be reset. When reset, LV_INH_LSH_UP[i] shall be set.

The Boolean flag LV_INH_LSH_CTL_CLL_LSH_UP[i] shall indicate whether closed loop heating shall be inhibited or not. Should either the Boolean constant LC_LSH_UP_CLL_CTL[i] be reset or one of several upstream sensor error bits be set then the Boolean inhibit flag LV_INH_LSH_CTL_CLL_LSH_UP[i] shall be set (open loop heater control), otherwise LV_INH_LSH_CTL_CLL_LSH_UP[i] shall be reset (closed loop heater control permitted).

Application conditions:

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Recurrence: 100 ms

Initialisation: LV_INH_LSH_UP[i] = 0

LV_INH_LSH_CTL_CLL_LSH_UP[i] = 0

LSHPWM_EXT_LS_UP[i] = 0%

V_EFC_LSH_UP_MAN = C_V_EFC_MAN

TEMP_INI_LS_UP[i] = TEG_WALL_CAT_UP_MDL[i]

Formula section:

Calculation of LSHPWM during external adjustment, manual adjustment via application system or not learnt variant

```
If(1)       LV_LSHPWM_UP_EXT_ADJ[i]= 0                  // no adjustment by tester
Then(1)
    If(2)       ( LC_LSHPWM_UP_MAN_ADJ[i] = 1       and
                'not serial software' )          // adjustment by application system
    Then(2)     LV_INH_LSH_UP[i] = 1
        If(3)   V_EFC_LSH_UP_MAN < C_V_EFC_MAN
        Then(3)
        $V\_EFC\_LSH\_UP\_MAN_{N} = V\_EFC\_LSH\_UP\_MAN_{N-1} + C\_INC\_V\_EFC\_MAN$
        $LSHPWM\_UP\_MAN = V\_EFC\_LSH\_UP\_MAN_{N}^{2} * 100/ VB_{N}^{2}$          // in %
        Else(3)     $V\_EFC\_LSH\_UP\_MAN_{N} = C\_V\_EFC\_MAN$
                    $LSHPWM\_UP\_MAN = C\_ V\_EFC\_MAN^{2} * 100/ VB_{N}^{2}$      // in %
        Endif(3)
        LSHPWM_EXT_LS_UP[i] = LSHPWM_UP_MAN
    Else(2)
        If(3)           LV_VAR_LSH_UP = 0
        Then(3)     LV_INH_LSH_UP[i] = 1
                    LSHPWM_EXT_LS_UP[i] = 5%
        Else(3)     LV_INH_LSH_UP[i] = ! LC_LSH_UP[i]
                    LSHPWM_EXT_LS_UP[i] = 0
                    LSHPWM_UP_MAN = 0
    Endif(2)
Else(1)     LV_INH_LSH_UP[i] = 1
            LSHPWM_EXT_LS_UP[i] =  LSHPWM_UP_EXT_ADJ[i]
Endif(1)
```

Calculation of Oxygen sensor temperature at engine start

One time determination after finishing engine state start

```
If    LV_ST_END = 1  for first time after LV_ST_END = 0
  Then       TEMP_INI_LS_UP[i]  =  TEG_WALL_CAT_UP_MDL[i]
Endif
```

Inhibition of closed loop heater control

```
If    LC_LSH_UP_CLL_CTL[i] = 0          or
      LV_ERR_OC_LSL_UP[i] = 1           or
      LV_ERR_LSL_UP_IF[i] = 1           or
      LV_ERR_EL_LSL_UP[i] = 1           or
      LV_ERR_OBD_VLD_LSH_UP[i] = 1or
      LV_VAR_LSH_UP = 0                 or
      LV_ERR_LSH_UP[i] = 1
  Then       LV_INH_LSH_CTL_CLL_LSH_UP[i]  =  1
  Else       LV_INH_LSH_CTL_CLL_LSH_UP[i]  =  0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_LSH_UP[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Boolean variable to enable oxygen sensor heating					
LC_LSH_UP_CLL_CTL[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Boolean variable to enable closed loop oxygen sensor heater control					
LC_LSHPWM_UP_MAN_ADJ[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Boolean variable to enable oxygen sensor heating via application system					
C_INC_V_EFC_MAN	1	0...FFH	0...1.625	6.37E-03	[V/100ms]
Effective heater voltage increment to raise temperature via application system					
C_V_EFC_MAN	1	0...FFH	0...25.89843	0.1015625	[V]
Maximum permitted effective heater voltage via application system					

16.20 Downstream oxygen sensor signal voltage

16.20.1 Signal evaluation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_DOWN_DRV1_MMV[NC_CBK_EX_NR]	V/O	8000...7FFFH	-2.5...2.4999	7.63E-05	[V/10ms]
Mean of buffer containing 1st derivative of VLS_DOWN[i] signal					
VLS_DOWN_DRV1_MMV_MIN[NC_CBK_EX_NR]	V/O	8000...7FFFH	-2.5...2.4999	7.63E-05	[V/10ms]
Minimum value of mean of buffer containing 1st derivative of VLS_DOWN[i] signal in lean phase					
VLS_DOWN_DRV1_ABS_MAX[NC_CBK_EX_NR]	V/O	0...FFFFH	0...4.99992	7.63E-05	[V/10ms]
Absolute maximum value within buffer containing 1st derivative of VLS_DOWN[i] signal					
VLS_DOWN_MMV_MIN[NC_CBK_EX_NR]	V/O	0...FFFFH	0...4.99992	7.63E-05	[V]
Mean oxygen sensor lean voltage					
VLS_DOWN_MMV_MAX[NC_CBK_EX_NR]	V/O	0...FFFFH	0...4.99992	7.63E-05	[V]
Mean oxygen sensor rich voltage					
VLS_DOWN_BOL[NC_CBK_EX_NR]	V/O	0...3FFH	0...4.99511	4.88E-03	[V]
Variable switching threshold for calculation of VLS_DOWN_MMV_MIN[i]					
VLS_DOWN_TOL[NC_CBK_EX_NR]	V/O	0...3FFH	0...4.99511	4.88E-03	[V]
Variable switching threshold for calculation of VLS_DOWN_MMV_MAX[i]					
VLS_DOWN_MMV_HYS[NC_CBK_EX_NR]	V	0...3333H	0...0.99998	7.63E-05	[V]
Variable hysteresis used to determine switching thresholds					
LV_VLS_DOWN_MMV_LIM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating VLS_DOWN_MMV_xxx[i] signals being limited when set					
LV_VLS_DOWN_MMV_ACT	V/O	0...1H	0...1	1	[-]
Boolean flag indicating calibrateable delay since PUC deactivation passed					

Input data:

LV_ST_END	LV_ES	LV_PUC	NC_CBK_EX_NR
VLS_DOWN[NC_CBK_EX_NR]			

FUNCTION DESCRIPTION:

General information:

The function shall evaluate the oxygen sensor signal voltage and compute a number of resultant values that may be used by other functions. These values shall include:

1. Moving mean of sensor voltage first derivative, i.e. average signal gradient
2. Minimum value of moving mean of sensor voltage first derivative in lean phase
3. Absolute maximum of sensor voltage first derivative over defined number of previous samples, i.e. measure of sensor voltage stability
4. Moving mean value of lean sensor voltage
5. Moving mean value of rich sensor voltage.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2

otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank

Signal flow diagram:

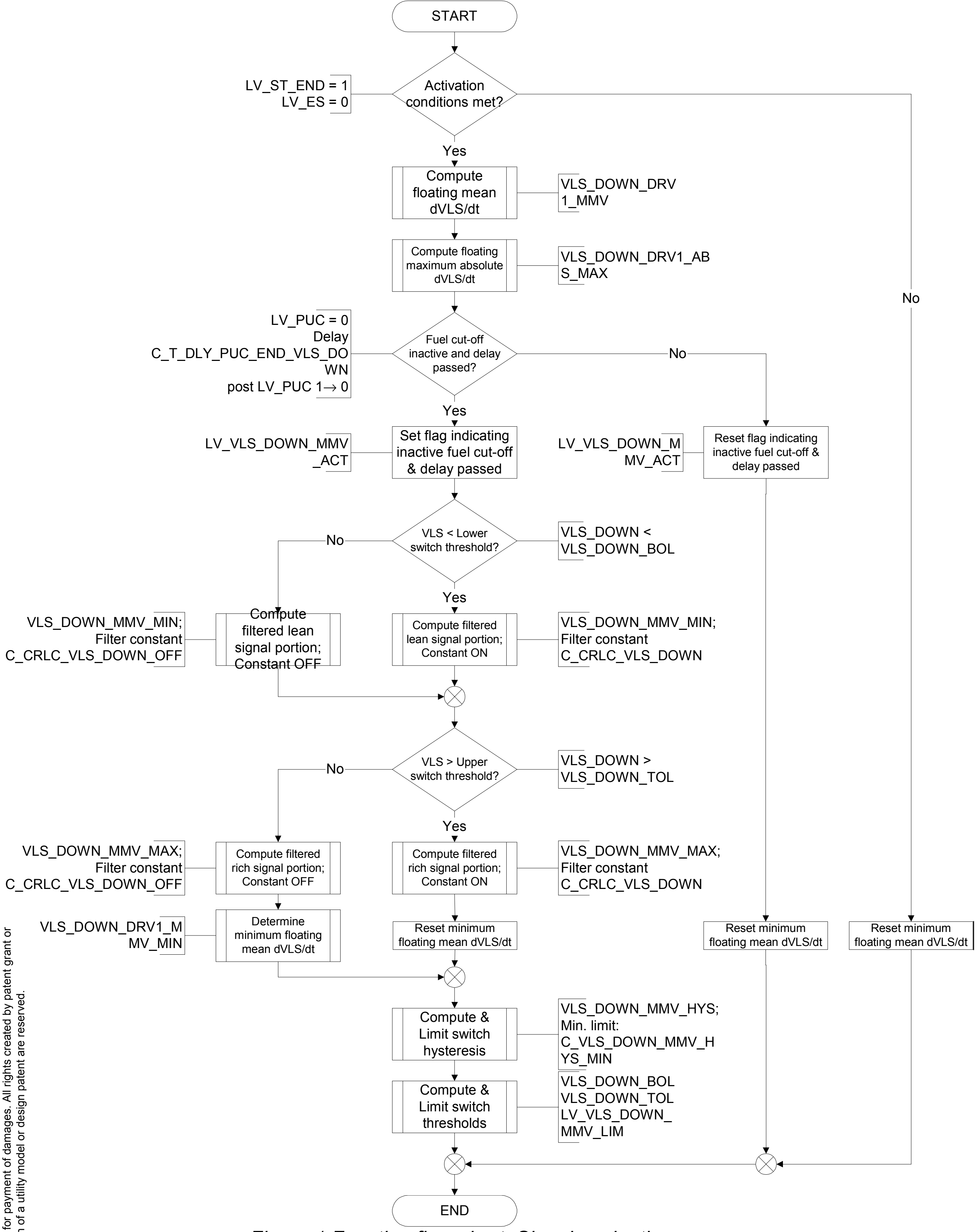


Figure 1 Function flow chart, Signal evaluation

Description:

The function shall evaluate the downstream oxygen sensor signal VLS_DOWN[i] and provide variables quantifying the signal characteristics: Floating mean gradient, Floating maximum signal deviation between samples, Filtered mean rich signal voltage & Filtered mean lean signal voltage (voltage will deviate from stable level signal with ageing of the catalyst). This shall be carried out as follows:

The function shall be activated should the engine state transition out of start (LV_ST_END = 1) and shall be deactivated should the engine stop state (LV_ES = 1) be entered.

Once the activation conditions have been met the function may be described in the following blocks:

1. Computation of moving mean of first derivative VLS_DOWN_DRV1_MMV[i]: The moving mean value of oxygen sensor signal voltage first derivative VLS_DOWN_DRV1_MMV[i] shall be computed by averaging the last differences between consecutive VLS_DOWN[i] samples, i.e. the most current difference (VLS_DOWN$[i]_N$ - VLS_DOWN$[i]_{N-1}$) and the previous 5 differences. This shall indicate a measure for the signal gradient.

2. Computation of absolute maximum of first derivative VLS_DOWN_DRV1_ABS_MAX[i]: The absolute maximum of the oxygen sensor voltage first derivative VLS_DOWN_DRV1_ABS_MAX[i] shall be computed by taking the maximum of the absolute values of the last differences between consecutive VLS_DOWN[i] samples, i.e. the most current difference (VLS_DOWN$[i]_N$ - VLS_DOWN$[i]_{N-1}$) and the previous 5 differences. This shall indicate a measure for the signal stability.

3. Computation of moving mean value of lean and rich sensor voltages, VLS_DOWN_MMV_xxx[i]: The computation of the moving mean value of lean and rich sensor voltages, associated switch thresholds and switch hysteresis shall only be carried out when the overrun fuel cut-off flag is reset (LV_PUC = 0) and the time C_T_DLY_PUC_END_VLS_DOWN has passed since the deactivation of the last occurring engine state PUC (i.e. LV_PUC 1 → 0). The lean and rich mean signal voltages shall be implemented by use of first order low pass filtering and floating switching thresholds. For example:

Should the VLS_DOWN[i] signal exceeds the threshold VLS_DOWN_TOL[i], the rich mean signal voltage VLS_DOWN_MMV_MAX[i] shall be computed by filtering the VLS_DOWN[i] signal with filter constant C_CRLC_VLS_DOWN. This shall continue until threshold VLS_DOWN_TOL[i] is no longer exceeded, after which VLS_DOWN_MMV_MAX[i] shall be computed by filtering with filter constant C_CRLC_VLS_DOWN_OFF. This constant is much smaller than C_CRLC_VLS_DOWN and the effect is to cause the VLS_DOWN_MMV_MAX[i] to follow the rich peaks. The OFF constant shall permit the VLS_DOWN_MMV_MAX[i] signal to follow the VLS_DOWN[i] signal for sudden decreases in signal amplitude, i.e. maximum VLS_DOWN[i] lower than VLS_DOWN_TOL[i].

The same procedure shall be carried out for the computation of VLS_DOWN_MMV_MIN[i], i.e. where VLS_DOWN[i] falls below VLS_DOWN_BOL[i], VLS_DOWN_MMV_MIN[i] shall be computed with filter constant C_CRLC_VLS_DOWN and where VLS_DOWN[i] no longer falls below VLS_DOWN_BOL[i], the filter constant C_CRLC_VLS_DOWN_OFF shall be used.

4. Determination of the minimum value of the moving mean of first derivative VLS_DOWN_DRV1_MMV_MIN[i]: The same conditions as for the computation of VLS_DOWN_MMV_xxx[i] shall apply. Should VLS_DOWN[i] fall below or equal VLS_DOWN_TOL[i], VLS_DOWN_DRV1_MMV_MIN$[i]_N$ shall be equal to the lower of VLS_DOWN_DRV1_MMV_MIN$[i]_{N-1}$ & VLS_DOWN_DRV1_MMV[i]. When VLS_DOWN[i]

exceeds VLS_DOWN_TOL[i], the LV_PUC condition in conjunction with delay C_T_DLY_PUC_END_VLS_DOWN is not met or the function is deactivated, VLS_DOWN_DRV1_MMV_MIN[i] shall be reset to 0.

5. Compute & limit switch hysteresis: The value of the hysteresis, VLS_DOWN_MMV_HYS[i], used to determine the switching thresholds VLS_DOWN_BOL[i] & VLS_DOWN_TOL[i] shall be made proportional using factor C_FAC_VLS_DOWN_MMV_HYS to the oxygen sensor signal amplitude, as determined by taking the absolute difference of the moving mean lean / rich variables. The hysteresis shall also be limited so that it may not fall below the calibrateable threshold C_VLS_DOWN_MMV_HYS_MIN.

6. Compute & limit switch thresholds: The function shall limit the minimum amplitude of the difference between the moving mean lean / rich variables (VLS_DOWN_MMV_MAX[i] - VLS_DOWN_MMV_MIN[i]) to twice that of the switch hysteresis, VLS_DOWN_MMV_HYS[i].

Should the amplitude of the aforementioned difference fall below the said threshold, both the switch thresholds VLS_DOWN_BOL[i] & VLS_DOWN_TOL[i] shall be set to the average of the moving mean lean / rich variables, the moving mean lean / rich variables are recalculated based on the limited switching thresholds and hysteresis and the flag LV_VLS_DOWN_MMV_LIM[i] shall be set to indicate that switch threshold limiting is active.

Should no limiting be required, the switch thresholds VLS_DOWN_BOL[i] & VLS_DOWN_TOL[i] shall be calculated based on the switch hysteresis, moving mean lean variable and moving mean rich variable respectively. The flag LV_VLS_DOWN_MMV_LIM[i] shall be reset to indicate that no switch threshold limiting is being carried out.

The above blocks shall be cycled through once every function recurrence.

Application conditions:

Recurrence:

The function shall be carried out once every oxygen sensor acquisition, i.e. every 10 ms.

Initialisation:

The following initialisation shall be carried out after a RESET and upon leaving the engine state Engine Stop (LV_ES).

VLS_DOWN_MMV_MIN[i] = C_VLS_DOWN_INI_MIN
VLS_DOWN_MMV_MAX[i] = C_VLS_DOWN_INI_MAX
VLS_DOWN_BOL[i] = C_VLS_DOWN_INI_MAX
VLS_DOWN_TOL[i] = C_VLS_DOWN_INI_MIN
VLS_DOWN_MMV_HYS[i] = C_VLS_DOWN_MMV_HYS_MIN
LV_VLS_DOWN_MMV_LIM[i] = 0
LV_VLS_DOWN_MMV_ACT = 0
VLS_DOWN_DRV1_MMV[i] = 0
VLS_DOWN_DRV1_MMV_MIN[i] = 0
VLS_DOWN_DRV1_ABS_MAX[i] = 0

Buffer contents from which VLS_DOWN_DRV1_MMV[i] & VLS_DOWN_DRV1_ABS_MAX[i] computed = 0.

Activation:

If LV_ST_END = 1

then ***Function activated***

endif.

Deactivation:

```
If  LV_ST_END = 0
    then  Function deactivated
        VLS_DOWN_DRV1_MMV_MIN[i] = 0
endif.
```

Formula section:

The following function shall be carried out in the order listed and as indicated by the function flow chart:

Computation of moving mean of first derivative VLS_DOWN_DRV1_MMV[i]

$$\text{VLS\_DOWN\_DRV1\_MMV[i]} = \frac{\sum_{x=0}^{5}\left(\text{VLS\_DOWN[i]}_{N-x} - \text{VLS\_DOWN[i]}_{N-(x+1)}\right)}{6 * 10\text{ms}}$$

The above may be facilitated for example by use of a 6 value ring buffer. The most current difference, (VLS_DOWN[i]$_{N}$ - VLS_DOWN[i]$_{N-1}$), would overwrite the oldest difference in the buffer, (VLS_DOWN[i]$_{N-5}$ - VLS_DOWN[i]$_{N-6}$), and the average of the buffer contents recalculated.

Computation of absolute maximum of first derivative number of previous samples VLS_DOWN_DRV1_ABS_MAX[i]

$$\text{VLS\_DOWN\_DRV1\_ABS\_MAX[i]} = \text{MAXIMUM}\left|{}_{x=0}^{5}\left(\text{VLS\_DOWN[i]}_{N-x} - \text{VLS\_DOWN[i]}_{N-(x+1)}\right)\right|$$

The above may be facilitated for example by use of the same 6 value ring buffer, as proposed above for the computation of VLS_DOWN_DRV1_MMV[i]. The function would calculate the absolute values of the buffer contents and take the maximum of these values.

Computation of moving mean value of lean and rich sensor voltages, VLS_DOWN_MMV_xxx[i]

```
If  (LV_PUC = 0) &
    (delay since last LV_PUC 1 → 0 transition ≥ C_T_DLY_PUC_END_VLS_DOWN)
then  LV_VLS_DOWN_MMV_ACT = 1
    If  (VLS_DOWN[i] < VLS_DOWN_BOL[i])
    then  Compute moving mean lean signal, filter constant ON
    else  Compute moving mean lean signal, filter constant OFF
    endif
    If  (VLS_DOWN[i] > VLS_DOWN_TOL[i])
    then  Compute moving mean rich signal, filter constant ON
          VLS_DOWN_DRV1_MMV_MIN[i] = 0
    else  Compute moving mean rich signal, filter constant OFF
          Determine minimum of moving mean of first derivative
    endif
    Compute & limit switch hysteresis
```

```
        Compute & limit switch thresholds
else    LV_VLS_DOWN_MMV_ACT = 0
        Reset timer used for delay C_T_DLY_PUC_END_VLS_DOWN
        VLS_DOWN_DRV1_MMV_MIN[i] = 0
endif.
```

Compute moving mean lean signal, filter constant ON

The moving mean of the lean signal voltage shall be realised by use of a 1st order low pass digital filter with the filter constant C_CRLC_VLS_DOWN.

VLS_DOWN_MMV_MIN[i]$_N$ = VLS_DOWN_MMV_MIN[i]$_{N-1}$ * (1-C_CRLC_VLS_DOWN) +
VLS_DOWN[i]$_N$ * C_CRLC_VLS_DOWN

Compute moving mean lean signal, filter constant OFF

The moving mean of the lean signal voltage shall be realised by use of a 1st order low pass digital filter with the filter constant C_CRLC_VLS_DOWN_OFF.

VLS_DOWN_MMV_MIN[i]$_N$ = VLS_DOWN_MMV_MIN[i]$_{N-1}$ *
(1-C_CRLC_VLS_DOWN_OFF) +
VLS_DOWN[i]$_N$ * C_CRLC_VLS_DOWN_OFF

Compute moving mean rich signal, filter constant ON

The moving mean of the rich signal voltage shall be realised by use of a 1st order low pass digital filter with the filter constant C_CRLC_VLS_DOWN.

VLS_DOWN_MMV_MAX[i]$_N$ = VLS_DOWN_MMV_MAX[i]$_{N-1}$ * (1-C_CRLC_VLS_DOWN) +
VLS_DOWN[i]$_N$ * C_CRLC_VLS_DOWN

Compute moving mean rich signal, filter constant OFF

The moving mean of the rich signal voltage shall be realised by use of a 1st order low pass digital filter with the filter constant C_CRLC_VLS_DOWN_OFF.

VLS_DOWN_MMV_MAX[i]$_N$ = VLS_DOWN_MMV_MAX[i]$_{N-1}$ *
(1-C_CRLC_VLS_DOWN_OFF) +
VLS_DOWN[i]$_N$ * C_CRLC_VLS_DOWN_OFF

Determine minimum of moving mean of first derivative VLS_DOWN_DRV1_MMV_MIN[i]

The determination of the minimum of VLS_DOWN_DRV1_MMV[i] shall take place when the condition VLS_DOWN[i] ≤ VLS_DOWN_TOL[i] has been met. In this case:

VLS_DOWN_DRV1_MMV_MIN[i] = MIN(VLS_DOWN_DRV1_MMV_MIN[i],
VLS_DOWN_DRV1_MMV[i])

Should the above condition no longer met, the LV_PUC with delay condition not be met or the initial activation conditions not be met then VLS_DOWN_DRV1_MMV_MIN[i] shall be reset to 0.

Compute & limit switch hysteresis

VLS_DOWN_MMV_HYS[i] = | (VLS_DOWN_MMV_MAX[i] - VLS_DOWN_MMV_MIN[i]) | *
C_FAC_VLS_DOWN_MMV_HYS

```
If    (VLS_DOWN_MMV_HYS[i] < C_VLS_DOWN_MMV_HYS_MIN)
then  VLS_DOWN_MMV_HYS[i] = C_VLS_DOWN_MMV_HYS_MIN
endif.
```

Compute & limit switch thresholds

```
If   ( (VLS_DOWN_MMV_MAX[i] - VLS_DOWN_MMV_MIN[i])
       ≤ 2 * VLS_DOWN_MMV_HYS[i] )
then   VLS_DOWN_BOL[i] = (VLS_DOWN_MMV_MAX[i] + VLS_DOWN_MMV_MIN[i]) / 2
       VLS_DOWN_TOL[i] = (VLS_DOWN_MMV_MAX[i] + VLS_DOWN_MMV_MIN[i]) / 2

       VLS_DOWN_MMV_MIN[i] = VLS_DOWN_BOL[i] - VLS_DOWN_MMV_HYS[i]
       VLS_DOWN_MMV_MAX[i] = VLS_DOWN_TOL[i] + VLS_DOWN_MMV_HYS[i]

       LV_VLS_DOWN_MMV_LIM[i] = 1
else   VLS_DOWN_BOL[i] = VLS_DOWN_MMV_MIN[i] + VLS_DOWN_MMV_HYS[i]
       VLS_DOWN_TOL[i] = VLS_DOWN_MMV_MAX[i] - VLS_DOWN_MMV_HYS[i]

       LV_VLS_DOWN_MMV_LIM[i] = 0
endif.
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VLS_DOWN_MMV_HYS_MIN	1	0...3333H	0...0.99998	7.63E-05	[V]
Minimum hysteresis for the computation of the thresholds VLS_DOWN_TOL[i] and VLS_DOWN_BOL[i]					
C_VLS_DOWN_INI_MIN	1	0...3FFH	0...4.99511	4.88E-03	[V]
Lower initialisation value for VLS_DOWN_TOL[i]. VLS_DOWN_MMV_MIN[i]					
C_VLS_DOWN_INI_MAX	1	0...3FFH	0...4.99511	4.88E-03	[V]
Upper initialisation value for VLS_DOWN_BOL[i]. VLS_DOWN_MMV_MAX[i]					
C_CRLC_VLS_DOWN	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Filter constant for determination of VLS_DOWN_MMV_xxx[i] values when respective threshold passed					
C_CRLC_VLS_DOWN_OFF	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Filter constant for determination of VLS_DOWN_MMV_xxx[i] values when respective threshold not passed					
C_FAC_VLS_DOWN_MMV_HYS	1	0...FFH	0...0.99609	3.91E-03	[-]
Multiplicative factor governing amplitude of hysteresis					
C_T_DLY_PUC_END_VLS_DOWN	1	0...FFFFH	0...655.35	0.01	[s]
Delay between completion of engine state PUC and computation of VLS_DOWN_MMV_xxx[i]					

16.21 O2 sensor (bin, down) operability detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LS_DOWN_READY[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Logical variable for operative readiness of downstream oxygen sensor(s)					
T_TEMP_DEW_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Time after dew point detection for downstream lambda probe					
LV_LS_DOWN_READY_DEAC[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical variable indicating that operative readiness of downstream oxygen sensor(s) has been reset					
NR_CBK_EX_LS_DOWN	-	0...4H	0...4	1	[-]
exhaust gas bank with downstream sensor (for Y configuration with NOx trap)					
NR_CBK_EX_NOT_LS_DOWN	-	0...4H	0...4	1	[-]
exhaust gas bank without downstream sensor (for Y configuration with NOx trap)					
MAF_INT_NOT_STAT	V	0...FFFFH	0...2912.66666	0.0444444	[g]
MAF integral during non stationary condition at downstream sensor (for Y configuration with NOx trap)					
LV_VLS_DOWN_NOT_STAT	V	0...1H	0...1	1	[-]
Logical variable to indicate non stationary conditions at downstream sensor (for Y configuration with NOx trap)					
MAF_INT_DLY_EG_NOX_EQU_DOWN	V	0...FFFFH	0...2912.66666	0.0444444	[g]
MAF integral to consider gas flow delay between downstream sensor and NOx sensor (for Y configuration)					

Input data:

LV_PUC	MAF_INT_PUC_ACT	LV_ES	NC_CBK_EX_NR
VLS_DOWN[NC_CBK_EX_NR]	TEG_DYN_LS_DOWN[NC_CBK_EX_NR]	LV_TEMP_DEW_LS_DOWN[NC_CBK_EX_NR]	STATE_LSH_DOWN[NC_CBK_EX_NR]
MAF_INT_PUC_NOT_ACT	NR_CONF_CBK_EX	NT_AGI	LV_VLS_NS_VLD[NC_NOX_SENS_CONF]
VLS_DIF_LAM_ADJ[NC_CBK_EX_NR]	MAF_CYL	LV_ERR_LS_DOWN[NC_CBK_EX_NR]	

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1, for exhaust cylinder bank 1
 i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1, for single exhaust cylinder bank.

The operative readiness of the downstream oxygen sensor(s) shall be determined by this function to allow other functions to act accordingly dependent upon the readiness.

Description:

The operative readiness shall only be determined for a downstream oxygen sensor which is not operatively ready. The oxygen sensor shall be set as inoperable after a ECU reset, in the state *Engine Stop*, when during PUC the MAF integral exceeds a calibrated threshold or when the dew point LV_TEMP_DEW_LS_DOWN[i] switches back to 0.

The operative readiness shall be determined by analysing the downstream oxygen sensor voltage (VLS_DOWN[i]) and determining whether the voltage has left a pre-determined voltage band after the lambda probe heater state machine has reached LSH_POW_CTL.

A second condition shall permit the forced operative readiness state, should the voltage level condition not be met, by assuming that when the modelled exhaust gas temperature at lambda probe position has reached a pre-determined threshold, that the downstream oxygen sensor is also at operating temperature and functioning. The forced readiness detection may additionally be delayed until dew point detection or even longer until the calibratable threshold C_T_MIN_TEMP_DEW_LS_DOWN has been reached, depending on its calibration.

Should the sensor's operative readiness be reset due to a sufficiently long fuel cut-off phase or reset of dew-point recognition a certain mass air flow integral threshold after leaving fuel cut-off shall be awaited until operability detection is started again. This shall ensure that the sensor heater has sufficient time to raise the temperature up to normal operating conditions prior to start of readiness detection in order to increase its reliability.

The operative readiness of the downstream oxygen sensor(s) shall be recorded in the variable LV_LS_DOWN_READY[i].

Application conditions:

Initialisation:

At reset: reset all variables to 0

Recurrence: T_SAMPLE = 100 ms

Activation: In all engine states

Formula section:

% only relevant for Y configurations with one downstream sensor and 1 NOx sensor

%SW-hint: The following if-then-else calculation can be done outside the for-loop, since calculation is independent of exhaust bank.

```
#if  NC_CBK_EX_NR = 2
#then
    if  NR_CONF_CBK_EX = 3              % downstream sensor on bank 1 is missing
    then  NR_CBK_EX_LS_DOWN = 2
          NR_CBK_EX_NOT_LS_DOWN = 1
    elseif  NR_CONF_CBK_EX = 4          % downstream sensor on bank 2 is missing
        NR_CBK_EX_LS_DOWN = 1
        NR_CBK_EX_NOT_LS_DOWN = 2
    else                                % no downstream sensor is missing
        NR_CBK_EX_LS_DOWN = 0
        NR_CBK_EX_NOT_LS_DOWN = 0
    endif
#else                                   % NC_CBK_EX_NR = 1
        NR_CBK_EX_LS_DOWN = 0
```

```
        NR_CBK_EX_NOT_LS_DOWN = 0
#endif


If    LV_ES = 1
then
      LV_LS_DOWN_READY[i] = 0
      T_TEMP_DEW_LS_DOWN[i] = 0
      LV_LS_DOWN_READY_DEAC[i] = 0
else
      if      i = NR_CBK_EX_NOT_LS_DOWN
      % only relevant for Y configurations with one downstream sensor and 1 NOx sensor
      then    % calculation for the exhaust bank without any downstream sensor
         if   VLS_DIF_LAM_ADJ[NR_CBK_EX_LS_DOWN] >=
                          C_VLS_DIF_ MAX_LAM_ADJ
         or   VLS_DIF_LAM_ADJ[NR_CBK_EX_LS_DOWN] <=
                          C_VLS_DIF_MIN_LAM_ADJ
         then
              if      MAF_INT_NOT_STAT < C_MAF_MAX_INT_NOT_STAT
              then    $MAF\_INT\_NOT\_STAT_n = MAF\_INT\_NOT\_STAT_{n-1}$ + MAF_CYL *
                          T_SAMPLE[ms] * 1/3600 [g*h/(kg*ms)]
                      % increment MAF_INT_NOT_STAT
              endif
         else         $MAF\_INT\_NOT\_STAT_n = MAF\_INT\_NOT\_STAT_{n-1}$ - MAF_CYL *
                          T_SAMPLE[ms] * 1/3600 [g*h/(kg*ms)]
                      % decrement MAF_INT_NOT_STAT
         endif
         if           MAF_INT_NOT_STAT >= C_MAF_MAX_INT_NOT_STAT
         then         % deviation on bank with downstream of untolerable magnitude
                      LV_VLS_DOWN_NOT_STAT = 1
         else         LV_VLS_DOWN_NOT_STAT = 0
         endif
         if           LV_LS_DOWN_READY[NR_CBK_EX_LS_DOWN] = 1
         and LV_ERR_LS_DOWN[NR_CBK_EX_LS_DOWN] = 0
         and LV_VLS_DOWN_NOT_STAT = 0
         then         % wait for gas flow delay from downstream sensor to NOx sensor
                      $MAF\_INT\_DLY\_EG\_NOX\_EQU\_DOWN_n$ =
                              $MAF\_INT\_DLY\_EG\_NOX\_EQU\_DOWN_{n-1}$ + MAF_CYL *
                              T_SAMPLE[ms] * 1/3600 [g*h/(kg*ms)]
```

```
                    % increment MAF_INT_DLY_EG_NOX_EQU_DOWN
        else        MAF_INT_DLY_EG_NOX_EQU_DOWN = 0
        endif
        if          MAF_INT_DLY_EG_NOX_EQU_DOWN >
                                IP_MAF_MAX_DLY_EG_FAC_NT_AGI
                    % maximum gas flow delay is dependent on NOx trap ageing
        and         LV_VLS_NS_VLD[1] = 1
        then        % deviation on bank with downstream sensor is low enough to neglect influence
                    LV_LS_DOWN_READY[i] = 1
        else        LV_LS_DOWN_READY[i] = 0
        endif
      else % all other configurations
        if      (LV_PUC = 1
        and     MAF_INT_PUC_ACT > C_MAF_INT_PUC_MAX_DOWN_LS_READY
        and         MAF_INT_PUC_ACT$_{n}$ > MAF_INT_PUC_ACT$_{n-1}$)
        or   LV_TEMP_DEW_LS_DOWN[i] = 1 -> 0
        then        LV_LS_DOWN_READY[i] = 0
                    LV_LS_DOWN_READY_DEAC[i] = 1
        endif
        If   LV_LS_DOWN_READY[i] = 0
        and  LV_LS_DOWN_READY_DEAC[i] = 0
        then        execute section "Readiness detection"
        else
            if      MAF_INT_PUC_NOT_ACT >=
                    C_MAF_INT_PUC_NOT_LS_DOWN_READY
            and     MAF_INT_PUC_NOT_ACT$_{n}$ > MAF_INT_PUC_NOT_ACT$_{n-1}$
            then    LV_LS_DOWN_READY_DEAC[i] = 0
                    T_TEMP_DEW_LS_DOWN[i] = 0
            endif
        endif
      endif
  endif
```

Readiness detection

% all other configurations

```
if      LV_TEMP_DEW_LS_DOWN[i] = 1
then    T_TEMP_DEW_LS_DOWN[i] = T_TEMP_DEW_LS_DOWN[i] + T_SAMPLE
else    T_TEMP_DEW_LS_DOWN[i] = 0
endif

if      ( (C_VLS_DOWN_AFR_THD_READY < VLS_DOWN[i]    or
        VLS_DOWN[i] < C_VLS_DOWN_AFL_THD_READY)    and
        STATE_LSH_DOWN[i] = LSH_POW_CTL )
or      ( TEG_DYN_LS_DOWN[i] > C_TEG_DYN_MIN_LS_DOWN_READY    and
        T_TEMP_DEW_LS_DOWN[i] >= C_T_MIN_TEMP_DEW_LS_DOWN )
then    LV_LS_DOWN_READY[i] = 1
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEG_DYN_MIN_LS_DOWN_READY	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Temperature threshold above which O2 sensor recognised as operatively ready					
C_VLS_DOWN_AFL_THD_READY	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lean VLS_DOWN[i] threshold for operative readiness detection					
C_VLS_DOWN_AFR_THD_READY	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Rich VLS_DOWN[i] threshold for operative readiness detection					
C_T_MIN_TEMP_DEW_LS_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time threshold after dew point recognition for lambda probe readiness detection					
C_MAF_INT_PUC_MAX_DOWN_LS_READY	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Maximum MAF integral during PUC for operability reset					
C_MAF_INT_PUC_NOT_LS_DOWN_READY	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Minimum MAF integral after operability reset for restart of readiness detection					
C_VLS_DIF_MAX_LAM_ADJ	1	FC00...3FFH	-5...4.99511	4.8828e-3	[V]
maximum positive setpoint deviation tolerable to use NOx sensor signal instead of downstream sensor					
C_VLS_DIF_MIN_LAM_ADJ	1	FC00...3FFH	-5...4.99511	4.8828e-3	[V]
maximum negative setpoint deviation tolerable to use NOx sensor signal instead of downstream sensor					
C_MAF_MAX_INT_NOT_STAT	1	0...FFFFH	0...2912.66666	0.0444444	[g]
MAF integral threshold above which non stationary condition at downstream sensor apply					
IP_MAF_MAX_DLY_EG_FAC_NT_AGI	6	0...FFFFH	0...2912.66666	0.0444444	[g]
LDP_NT_AGI_IP_MAF_MAX_DLY_EG	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
MAF integral threshold for consideration of gas delay flow (dependent on NOx trap ageing)					

16.22 Downstream oxygen sensor heater management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LSHPWM_DOWN[NC_CBK_EX_NR]	V/O	0...FFH	0...99.60937	0.390625	[%]
Heater driver PWM duty cycle; acquired also by BSW					
V_EFC_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...FFH	0...25.89843	0.1015625	[V]
Effective final output value of heater voltage					
V_EFC_CLC_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...25.99960	3.97E-04	[V]
Effective calculated heater voltage value prior to overvoltage protection					
T_POW_RISE_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...FFFFH	0...6553.5	0.1	[s]
Timer indicating the duration of the pre-heating and post dew point heating phases					
T_V_EFC_TOL_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...25.5	0.1	[s]
Timer indicating permitted effective heater voltage overvoltage duration					
T_V_EFC_LIM_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...25.5	0.1	[s]
Timer indicating the duration of the effective heater voltage limitation preventing overvoltage					
LV_V_EFC_LIM_BOL_LSH_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating effective heater voltage limited due to sum of voltage components falling below zero.					
LV_V_EFC_LIM_TOL_LSH_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating effective heater voltage limited due to exceeding threshold for period of time.					
LV_V_EFC_LIM_MAX_LSH_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating effective heater voltage limited due to exceeding absolute maximum voltage spec.					
LV_V_EFC_LIM_PROT_VB_LSH_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean variable indicating effective heater voltage limited due to excessive battery voltage level					
STATE_LSH_DOWN[NC_CBK_EX_NR]	V/O	0H 1H 2H 3H 4H 5H 6H	LSH_OFF LSH_POW_RISE LSH_POW_RED LSH_POW_FALL LSH_POW_CTL LSH_VB_PROT LSH_TEMP_PROT	1	[-]
Present heater state					

Input data:

LV_INH_LSH_DOWN[NC_CBK_EX_NR]	LV_LS_DOWN_READY[NC_CBK_EX_NR]	TEMP_INI_LS_DOWN[NC_CBK_EX_NR]	LSHPWM_EXT_LS_DOWN[NC_CBK_EX_NR]
LV_ST_END	VS	TEG_DYN_LS_DOWN[NC_CBK_EX_NR]	LV_TEMP_DEW_LS_DOWN[NC_CBK_EX_NR]
VB	N_32	T_AST	NC_CBK_EX_NR

Import actions:

ACTION_INFR_SetPwmLshDown(IN <i>, IN <ducy>)
This action sets the lambda heater downstream driver

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

The goal of the heater control shall be to control the oxygen sensor heating such that the optimal operating temperature is reached in the shortest time possible. At the same time, the maximum permissible temperature gradient (possible damage to ceramic due to thermal stress), the possible occurrence of water splash under the dew point (possible damage to ceramic due to thermal shock) and the absolute maximum ratings specified for the sensor shall be taken into account. The function shall permit oxygen sensor heating under open loop control.

Signal flow:

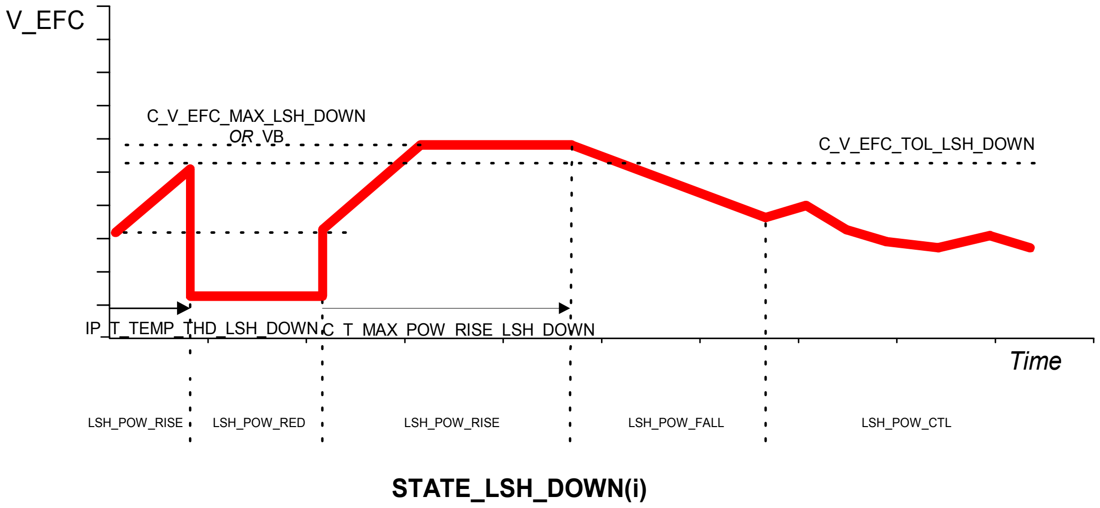


Figure 1: Course of effective heater voltage over time / state

Description:

The heater control shall be realised by use of a state machine. Each downstream oxygen sensor shall have its own state machine and each state machine shall run independently of the other.

The heater control shall be initialised to the LSH_OFF state at the start of a new driving cycle. Hence the oxygen sensor will not initially be electrically heated.

Heating shall commence should the activation conditions listed below be met. i.e. when the engine has transitioned from the start state (LV_ST_END = 1) and the heater management inhibit bit for the corresponding bank is not set (LV_INH_LSH_DOWN[i] = 0). The appropriate heater state shall change to LSH_POW_RISE and the initial value for the effective heater voltage shall be determined by obtaining the oxygen sensor temperature at start (TEMP_INI_LS_DOWN[i]) from input data. This shall then be used to obtain the appropriate effective heater starting voltage from map IP_V_EFC_INI_LSH_DOWN.

In the state LSH_POW_RISE, the effective heater voltage shall be incremented with the gradient of C_V_EFC_INC_LSH_DOWN.

The continuation of the heating gradient shall depend on whether the dew point at the sensor location has been determined to have been exceeded, i.e. the temperature of the inner manifold wall at the sensor location has exceeded the temperature threshold where water in the exhaust gas may condense, as determined by LV_TEMP_DEW_LS_DOWN[i].

Should the dew point not have been exceeded (LV_TEMP_DEW_LS_DOWN[i] = 0) but timer T_POW_RISE_LSH_DOWN[i] has equalled or exceeded IP_T_TEMP_THD_LSH_DOWN, it shall be assumed that condensed water splash at the oxygen sensor location may occur. In order to protect the oxygen sensor by keeping the temperature of the ceramic lower than the specified critical temperature, the heater power shall be reduced immediately by setting the effective heater voltage to C_V_EFC_RED_LSH_DOWN. In this case, the heater state shall be set to LSH_POW_RED.

The reduced heating state shall persist until the dew point in the exhaust gas system has been determined to have been exceeded, i.e. until LV_TEMP_DEW_LS_DOWN[i] = 1, at which point the heater state shall return to LSH_POW_RISE. The effective heater voltage shall be set to the same effective heater voltage as determined at start (IP_V_EFC_INI_LSH_DOWN), which may be modified by the calibration data C_V_EFC_STEP_LSH_DOWN, may be further incremented with a gradient of C_V_EFC_INC_LSH_DOWN without danger of damaging the oxygen sensor and the timer T_POW_RISE_LSH_DOWN[i] shall be reinitialised with the IP_T_TEMP_THD_LSH_DOWN determined at function start.

Should the initial exhaust gas temperature at start have been determined to exceed the dew point (LV_TEMP_DEW_LS_DOWN[i] = 1), the heater control shall not enter the reduced heater power state but continue to incrementally increase the heater power within the LSH_POW_RISE state.

Should T_POW_RISE_LSH_DOWN[i] be determined to have equalled or exceeded the value C_T_MAX_POW_RISE_LSH_DOWN (irrespective of the dew point having been exceeded or not), the heater state shall change to LSH_POW_FALL and the timer shall be frozen.

In the LSH_POW_FALL state, the effective heater voltage shall be reduced with a gradient of C_V_EFC_DEC_LSH_DOWN until such time that it falls below or equals the value set for the open loop control at the current engine operating point. For cylinder bank 1, this value is determined by map IP_V_EFC_CTL_LSH_DOWN alone. For cylinder bank 2, this value is determined by IP_V_EFC_CTL_LSH_DOWN * IP_FAC_V_EFC_TEG_LSH_DOWN. The latter factor shall be necessary to accommodate differences in heat transfer rates between cylinder banks at the oxygen sensor location, particularly for transversal mounted V6-engines. At such time the effective heater voltage falls below or equals the respective threshold, the heater state shall change to LSH_POW_CTL.

Should, however, the Boolean flag LV_TEMP_DEW_LS_DOWN[i] be reset whilst in the state LSH_POW_FALL, the heater control shall resume at state LSH_POW_RED following the procedures outlined above.

The control of the effective heater voltage is carried out in open loop mode.

In the open loop heater control mode, the effective heater voltage shall be obtained from map IP_V_EFC_CTL_LSH_DOWN and is dependent upon the engine speed and the modelled exhaust gas temperature. For cylinder bank 2, an additional multiplicative factor (IP_FAC_V_EFC_TEG_LSH_DOWN) may be required for the reasons outlined above.

To the open loop control voltage a calibration system constant (C_V_EFC_AS_LSH_DOWN) can be added to permit a global change in the total effective heater voltage to be made. Thus the calculated effective heater voltage (V_EFC_CLC_LSH_DOWN[i]) shall be computed from the open loop value and the global correction value.

Furthermore should the sum of the components used to calculate the effective heater voltage be negative, LV_V_EFC_LIM_BOL_LSH_DOWN[i] shall be set. This shall prevent the controller from decreasing the effective heater voltage further but not prevent it from increasing the voltage. When the sum of the components is no longer negative, LV_V_EFC_LIM_BOL_LSH_DOWN[i] shall be reset. Similarly, should either of the overvoltage limitation Boolean flags LV_V_EFC_LIM_TOL_LSH_DOWN[i] & LV_V_EFC_LIM_MAX_LSH_DOWN[i] be set, as described later, the controller shall be prevented from increasing the effective heater voltage further but not prevent it from decreasing the voltage. When the overvoltage flags are no longer set and the other applicable conditions are met, the controller shall resume normal operation.

Should, however, the Boolean flag LV_TEMP_DEW_LS_DOWN[i] be reset whilst in the state LSH_POW_CTL, then the open loop controller terms shall be reset, the Boolean flags to indicate open loop control and effective heater voltage limitation shall be reset and the heater control shall resume at state LSH_POW_RED following the procedures outlined above.

Should at any time the battery voltage VB exceed the threshold C_VB_MAX_PROT_LSH_DOWN, as determined by the state of logical variable LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i], the oxygen sensor heater power shall be reduced by reducing the effective heater voltage used to calculate the driver duty cycle to C_V_EFC_PROT_VB_LSH_DOWN. The heater state shall change to LSH_VB_PROT on the next recurrence of the state machine and the effective heater voltage set to the aforementioned limit. This shall prevent damage to the sensor due to excessive heater voltages e.g. due to jump start from 24 V.

Should the battery voltage recover when in the state LSH_VB_PROT, as indicated by LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0, T_POW_RISE_LSH_DOWN[i] shall be reset, V_EFC_LSH_DOWN[i] shall be initialised to IP_V_EFC_INI_LSH_DOWN calculated after the start, the heater control shall resume at state LSH_POW_RISE following the procedures outlined above.

Short term voltage excursions shall cause the heater power to be limited but will only cause the state machine to change state should the voltage excursion persist until the next recurrence of the state machine. The state machine shall not be able to enter the LSH_VB_PROT state directly from LSH_OFF but shall be required to pass through LSH_POW_RISE.

The sensor shall also be protected from over-temperature. This shall be achieved by evaluating the value of the modelled exhaust gas temperature TEG_DYN_LS_DOWN[i] in all states other than LSH_OFF. Should the exhaust gas temperature exceed the threshold C_TEG_DYN_MAX_LSH_DOWN, the calculated effective heater voltage shall be reduced to 0, the state shall change to LSH_TEMP_PROT thus suspending open loop control, if active.

Should the exhaust gas temperature fall below the threshold, the state shall change dependent on the state active immediately prior to the over-temperature protection. If the state was LSH_POW_CTL, the state machine shall return to this state and the effective heater voltage shall take the new value as computed by the open loop control. If the state was any other than LSH_POW_CTL, the state shall change to LSH_POW_RISE. In this case T_POW_RISE_LSH_DOWN[i] shall be reset, V_EFC_LSH_DOWN[i] shall be initialised to IP_V_EFC_INI_LSH_DOWN calculated after the start, effective heater voltage limitation shall be reset and the heater control shall resume at state LSH_POW_RISE following the procedures outlined above.

Furthermore, in order to protect the sensor heater against long term high voltage stress it shall be necessary to limit the effective heater voltage should the value of V_EFC_CLC_LSH_DOWN[i] equal or exceed the threshold C_V_EFC_TOL_LSH_DOWN.

In this instance, the timer T_V_EFC_TOL_LSH_DOWN[i] shall be started. Should the overvoltage condition persist, such that this timer equals or exceeds the threshold C_T_MAX_V_EFC_TOL_LSH_DOWN, timer T_V_EFC_LIM_LSH_DOWN[i] shall be started and the Boolean flag LV_V_EFC_LIM_TOL_LSH_DOWN[i] shall be set. This flag shall prevent the controller from increasing effective the voltage further. If however the value of V_EFC_CLC_LSH_DOWN[i] should fall below the threshold C_V_EFC_TOL_LSH_DOWN at any time, LV_V_EFC_LIM_TOL_LSH_DOWN[i] shall be reset.

Once timer T_V_EFC_LIM_LSH_DOWN[i] has been started, Boolean flag LV_V_EFC_LIM_TOL_LSH_DOWN[i] shall be set or reset depending on the value of V_EFC_CLC_LSH_DOWN[i]. Once the timer T_V_EFC_LIM_LSH_DOWN[i] equals or exceeds threshold C_T_MAX_V_EFC_LIM_LSH_DOWN, both timers shall be reset and timer T_V_EFC_TOL_LSH_DOWN[i] must exceed the threshold C_T_MAX_V_EFC_TOL_LSH_DOWN again in order to permit further limiting to take place.

To prevent damage to the sensor, V_EFC_CLC_LSH_DOWN[i] shall not be permitted to exceed the lower of the values C_V_EFC_MAX_LSH_DOWN, C_V_EFC_MAX_ST_LSH_DOWN or VB dependent on the value of T_POW_RISE_LSH_DOWN[i].

If T_POW_RISE_LSH_DOWN[i] equals or exceeds the constant C_T_MAX_V_EFC_MAX_ST_LSH_DOWN or conditions regarding T_AST and STATE_LSH_DOWN[i] then V_EFC_CLC_LSH_DOWN[i] shall be checked against VB & C_V_EFC_MAX_LSH_DOWN. If either of the values is exceeded then Boolean flag LV_V_EFC_LIM_MAX_LSH_DOWN[i] shall be set and V_EFC_CLC_LSH_DOWN[i] limited to the lower of VB & C_V_EFC_MAX_LSH_DOWN otherwise LV_V_EFC_LIM_MAX_LSH_DOWN[i] shall be reset.

If T_POW_RISE_LSH_DOWN[i] is lower then constant C_T_MAX_V_EFC_MAX_ST_LSH_DOWN then V_EFC_CLC_LSH_DOWN[i] shall be checked against VB & C_V_EFC_MAX_ST_LSH_DOWN. If either of the values is exceeded then Boolean flag LV_V_EFC_LIM_MAX_LSH_DOWN[i] shall be set and V_EFC_CLC_LSH_DOWN[i] limited to the lower of VB & C_V_EFC_MAX_ST_LSH_DOWN otherwise LV_V_EFC_LIM_MAX_LSH_DOWN[i] shall be reset.

The flag LV_V_EFC_LIM_MAX_LSH_DOWN[i] shall prevent the controller from increasing effective the voltage further and permit the final effective heater voltage to be limited, as described later.

Should timer T_V_EFC_LIM_LSH_DOWN[i] be unequal to zero, i.e. overvoltage no longer tolerated as it has been present for certain time, V_EFC_LSH_DOWN[i] shall be the lower of the values V_EFC_CLC_LSH_DOWN[i] & C_V_EFC_LIM_LSH_DOWN otherwise V_EFC_LSH_DOWN[i] shall be set to V_EFC_CLC_LSH_DOWN[i].

Once the effective heater voltage has been determined throughout the various heater states other than LSH_OFF, it shall be used to compute the PWM duty cycle (LSHPWM_DOWN[i]) that shall control the appropriate heater driver. The heater power shall be corrected to take into account deviations in the measured battery voltage and limited to the range bounded by NC_LSHPWM_TOL_LSH_DOWN & NC_LSHPWM_BOL_LSH_DOWN. The evaluation of excess battery voltage shall be determined at the same recurrence as the computation of the duty cycle as described above. The PWM value may also be modified by use of constant C_FAC_V_EFC_AS_LSH_DOWN.

Should at any time an inhibit flag be set (LV_INH_LSH_DOWN[i] = 1), the corresponding oxygen sensor heater shall be switched off and the STATE_LSH_DOWN[i] set to LSH_OFF. This shall allow project specific application conditions to turn off oxygen sensor heating according to project philosophy. Should the heater function be inhibited, then the PWM duty cycle shall be set to input LSHPWM_EXT_LS_DOWN[i]. On a transition of the inhibit bit LV_INH_LSH_DOWN[i] from 1 → 0, the heater function shall start from anew.

Application conditions:

Activation:

At any engine operting state.

Deactivation:

-

Time recurrence:

The state machine, sensor protection and effective heater voltage definition shall be carried out once every 100 ms. T_SAMPLE_1 = 100ms

The determination of the oxygen sensor heater driver duty cycle and associated excessive battery voltage protection shall be carried out once every 10 ms. T_SAMPLE_2 = 10ms

Initialisation:

The following variable initialisation shall take place at the beginning of a new driving cycle:

STATE_LSH_DOWN[i] = LSH_OFF

T_POW_RISE_LSH_DOWN[i] = 0

T_V_EFC_TOL_LSH_DOWN[i] = 0

T_V_EFC_LIM_LSH_DOWN[i] = 0

LSHPWM_DOWN[i] = 0

V_EFC_LSH_DOWN[i] = 0

V_EFC_CLC_LSH_DOWN[i] = 0

LV_V_EFC_LIM_BOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 0

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0

At reset:

ACTION_INFR_SetPwmLshDown(

IN i

IN LSHPWM_DOWN[i])

NOTE: Projects not making use of the inhibit flags LV_INH_LSH_DOWN[i] shall ensure that they are initialised with and remain 0, thus never disabling the oxygen sensor heater function or thus never disabling the oxygen sensor heater open loop control function respectively

Formula section:

16.22.1 Heater management state machine

The state machine shall remain in its current state and carry out the actions specified to occur within that state once per recurrence unless otherwise specified. The state machine shall only move to another state when one of the conditions has been determined to be met.

Note: The priorities of the conditions to change between states shall be defined by the order in which these conditions are listed within the appropriate state as described below.

HEATER STATE DIAGRAM

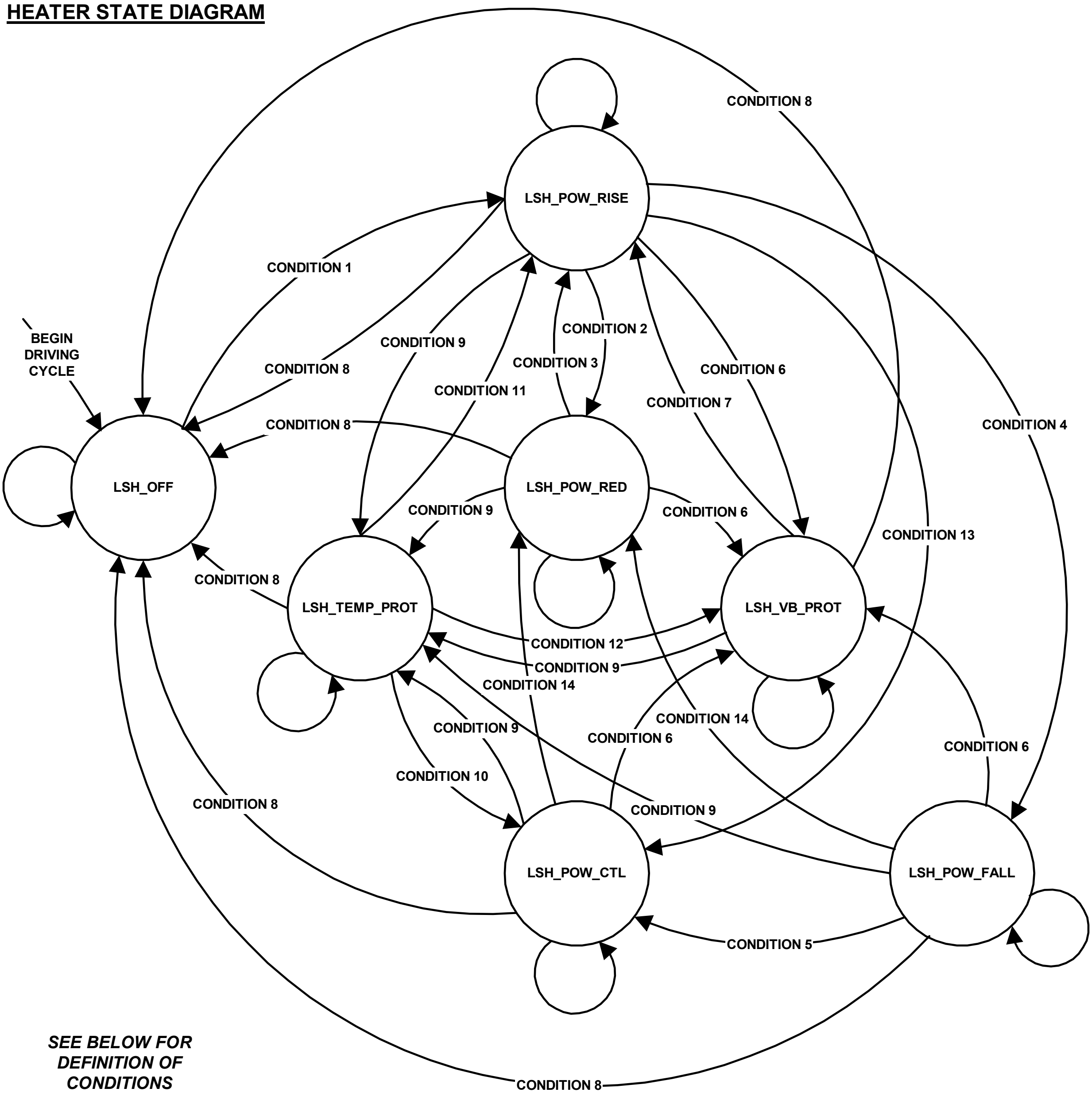


Figure 2: Heater management state diagram

STATE_LSH_DOWN[i] "LSH_OFF"

Actions:

T_POW_RISE_LSH_DOWN[i] = 0

T_V_EFC_TOL_LSH_DOWN[i] = 0

T_V_EFC_LIM_LSH_DOWN[i] = 0

V_EFC_LSH_DOWN[i] = 0

V_EFC_CLC_LSH_DOWN[i] = 0

LV_V_EFC_LIM_BOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 0

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0

Note: The above listed actions must be carried out at least once when the state has been entered for the first time or from any other state!

<u>Condition </u>1:“LSH_OFF to LSH_POW_RISE”

LV_ST_END = 1 **&**

LV_INH_LSH_DOWN[i] = 0

Transition actions:

V_EFC_CLC_LSH_DOWN[i] = IP_V_EFC_INI_LSH_DOWN (TEMP_INI_LS_DOWN[i])

Timer T_POW_RISE_LSH_DOWN[i] shall be started

STATE_LSH_DOWN[i] = LSH_POW_RISE

STATE_LSH_DOWN[i] “LSH_POW_RISE”

Actions:

V_EFC_CLC_LSH_DOWN[i]$_{N}$ = V_EFC_CLC_LSH_DOWN[i]$_{N-1}$ + C_V_EFC_INC_LSH_DOWN

T_SAMPLE_1

Increment timer T_POW_RISE_LSH_DOWN[i]

Note: See section “Oxygen sensor heater voltage protection” for notes on limiting the applied heater voltage.

<u>Condition 8</u>:“LSH_POW_RISE to LSH_OFF”

LV_ST_END = 0 **or** LV_INH_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_OFF

<u>Condition 9</u>:“LSH_POW_RISE to LSH_TEMP_PROT”

TEG_DYN_LS_DOWN[i] > C_TEG_DYN_MAX_LSH_DOWN **&**

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0

Transition actions:

STATE_LSH_DOWN[i] = LSH_TEMP_PROT

<u>Condition 6</u>:“LSH_POW_RISE to LSH_VB_PROT”

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_VB_PROT

<u>Condition 2</u>:“LSH_POW_RISE to LSH_POW_RED”

T_POW_RISE_LSH_DOWN[i] ≥ IP_T_TEMP_THD_LSH_DOWN (TEMP_INI_LS_DOWN[i]) **&**

LV_TEMP_DEW_LS_DOWN[i] = 0

Transition actions:

STATE_LSH_DOWN[i] = LSH_POW_RED

Condition 13:"LSH_POW_RISE to LSH_POW_CTL"

LV_LS_DOWN_READY[i] = 1

Transition actions:

Stop timer T_POW_RISE_LSH_DOWN[i] and freeze value

STATE_LSH_DOWN[i] = LSH_POW_CTL

Condition 4:"LSH_POW_RISE to LSH_POW_FALL"

T_POW_RISE_LSH_DOWN[i] ≥ C_T_MAX_POW_RISE_LSH_DOWN + IP_T_TEMP_THD_LSH_DOWN

% IP_ ... (TEMP_INI_LS_DOWN[i])

Transition actions:

Stop timer T_POW_RISE_LSH_DOWN[i] and freeze value

STATE_LSH_DOWN[i] = LSH_POW_FALL

STATE_LSH_DOWN[i] "LSH_POW_RED"

Actions:

V_EFC_CLC_LSH_DOWN[i] = C_V_EFC_RED_LSH_DOWN

Condition 8:"LSH_POW_RED to LSH_OFF"

LV_ST_END = 0 **or** LV_INH_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_OFF

Condition 9:"LSH_POW_RED to LSH_TEMP_PROT"

TEG_DYN_LS_DOWN[i] > C_TEG_DYN_MAX_LSH_DOWN **&**

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0

Transition actions:

STATE_LSH_DOWN[i] = LSH_TEMP_PROT

Condition 6:"LSH_POW_RED to LSH_VB_PROT"

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_VB_PROT

Condition 3:"LSH_POW_RED to LSH_POW_RISE"

LV_TEMP_DEW_LS_DOWN[i] = 1

Transition actions:

V_EFC_CLC_LSH_DOWN[i] = IP_V_EFC_INI_LSH_DOWN

+ C_V_EFC_STEP_LSH_DOWN

% IP_ ... (TEMP_INI_LS_DOWN[i]) *same value as at start*

T_POW_RISE_LSH_DOWN[i] = IP_T_TEMP_THD_LSH_DOWN

% TEMP_INI_LS_DOWN[i] *same value as at start*

STATE_LSH_DOWN[i] = LSH_POW_RISE

Note: See section "Oxygen sensor heater voltage protection" for notes on limiting the applied heater voltage. The function shall prevent the effective heater voltage from becoming negative by limiting to 0 where necessary.

STATE_LSH_DOWN[i] "LSH_POW_FALL"

Actions:

$V\_EFC\_CLC\_LSH\_DOWN[i]_N = V\_EFC\_CLC\_LSH\_DOWN[i]_{N-1} - C\_V\_EFC\_DEC\_LSH\_DOWN * T\_SAMPLE\_1$

Note: See section "Oxygen sensor heater voltage protection" for notes on limiting the applied heater voltage. The function shall prevent the effective heater voltage from becoming negative by limiting to 0 where necessary.

Condition 8:"LSH_POW_FALL to LSH_OFF"

LV_ST_END = 0 **or** LV_INH_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_OFF

Condition 9:"LSH_POW_FALL to LSH_TEMP_PROT"

TEG_DYN_LS_DOWN[i] > C_TEG_DYN_MAX_LSH_DOWN **&**

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0

Transition actions:

STATE_LSH_DOWN[i] = LSH_TEMP_PROT

Condition 6:"LSH_POW_FALL to LSH_VB_PROT"

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_VB_PROT

Condition 14:"LSH_POW_FALL to LSH_POW_RED"

LV_TEMP_DEW_LS_DOWN[i] = 0

Transition actions:

T_V_EFC_TOL_LSH_DOWN[i] = 0

T_V_EFC_LIM_LSH_DOWN[i] = 0

LV_V_EFC_LIM_BOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 0

STATE_LSH_DOWN[i] = LSH_POW_RED

Condition 5:"LSH_POW_FALL to LSH_POW_CTL"

For bank 1; i = 1

V_EFC_LSH_DOWN[i]≤IP_V_EFC_CTL_LSH_DOWN(N_32,TEG_DYN_LS_DOWN[i])

For bank 2; i = 2

V_EFC_LSH_DOWN[i]≤IP_V_EFC_CTL_LSH_DOWN(N_32,TEG_DYN_LS_DOWN[i])

* IP_FAC_V_EFC_TEG_LSH_DOWN

Transition actions:

STATE_LSH_DOWN[i] = LSH_POW_CTL

STATE_LSH_DOWN[i] “LSH_POW_CTL”

Actions:

Compute effective heater voltage from open loop control and closed loop delta voltages:

For Bank 1; i = 1

V_EFC_CLC_LSH_DOWN[i] = IP_V_EFC_CTL_LSH_DOWN + C_V_EFC_AS_LSH_DOWN

% IP_ ... (N_32, TEG_DYN_LS_DOWN[i])

Note: If (IP_V_EFC_CTL_LSH_DOWN + C_V_EFC_AS_LSH_DOWN) < 0 , i.e. would cause the effective heater voltage to be negative, V_EFC_CLC_LSH_DOWN[i] shall be limited to 0 and LV_V_EFC_LIM_BOL_LSH_DOWN[i] shall be set, until the sum less than zero condition is no longer met at which point LV_V_EFC_LIM_BOL_LSH_DOWN[i] shall be reset.

For Bank 2; i = 2

V_EFC_CLC_LSH_DOWN[i] = IP_V_EFC_CTL_LSH_DOWN * IP_FAC_V_EFC_TEG_LSH_DOWN

\+ C_V_EFC_AS_LSH_DOWN

% IP_ ... (N_32, TEG_DYN_LS_DOWN[i])

Note: If (IP_V_EFC_CTL_LSH_DOWN * IP_FAC_V_EFC_TEG_LSH_DOWN + C_V_EFC_AS_LSH_DOWN) < 0 , i.e. would cause the effective heater voltage to be negative, V_EFC_CLC_LSH_DOWN[i] shall be limited to 0 and LV_V_EFC_LIM_BOL_LSH_DOWN[i] shall be set, until the sum less than zero condition is no longer met at which point LV_V_EFC_LIM_BOL_LSH_DOWN[i] shall be reset.

Condition 8:“LSH_POW_CTL to LSH_OFF”

LV_ST_END = 0 **or** LV_INH_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_OFF

Condition 9:“LSH_POW_CTL to LSH_TEMP_PROT”

TEG_DYN_LS_DOWN[i] > C_TEG_DYN_MAX_LSH_DOWN **&**

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0

Transition actions:

STATE_LSH_DOWN[i] = LSH_TEMP_PROT

Condition 6:“LSH_POW_CTL to LSH_VB_PROT”

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_VB_PROT

Condition 14:“LSH_POW_CTL to LSH_POW_RED”

LV_TEMP_DEW_LS_DOWN[i] = 0

Transition actions:

T_V_EFC_TOL_LSH_DOWN[i] = 0

T_V_EFC_LIM_LSH_DOWN[i] = 0

LV_V_EFC_LIM_BOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 0

STATE_LSH_DOWN[i] = LSH_POW_RED

STATE_LSH_DOWN[i] "LSH_VB_PROT"

Actions:

V_EFC_CLC_LSH_DOWN[i] = C_V_EFC_PROT_VB_LSH_DOWN

Condition 8:"LSH_VB_PROT to LSH_OFF"

LV_ST_END = 0 **or** LV_INH_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_OFF

Condition 9:"LSH_VB_PROT to LSH_TEMP_PROT"

TEG_DYN_LS_DOWN[i] > C_TEG_DYN_MAX_LSH_DOWN **&**

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0

Transition actions:

STATE_LSH_DOWN[i] = LSH_TEMP_PROT

Condition 7:"LSH_VB_PROT to LSH_POW_RISE"

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0

Transition actions:

T_POW_RISE_LSH_DOWN[i] = 0

V_EFC_CLC_LSH_DOWN[i] = IP_V_EFC_INI_LSH_DOWN

% IP_ ... (TEMP_INI_LS_DOWN[i]) *same value as at start*

T_V_EFC_TOL_LSH_DOWN[i] = 0

T_V_EFC_LIM_LSH_DOWN[i] = 0

LV_V_EFC_LIM_BOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 0

STATE_LSH_DOWN[i] = LSH_POW_RISE

STATE_LSH_DOWN[i] "LSH_TEMP_PROT"

Actions:

V_EFC_CLC_LSH_DOWN[i] = 0

Condition 8:"LSH_TEMP_PROT to LSH_OFF"

LV_ST_END = 0 **or** LV_INH_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_OFF

<u>Condition 12</u>:"LSH_TEMP_PROT to LSH_VB_PROT"

LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 1

Transition actions:

STATE_LSH_DOWN[i] = LSH_VB_PROT

<u>Condition 10</u>:"LSH_TEMP_PROT to LSH_POW_CTL"

TEG_DYN_LS_DOWN[i] ≤ C_TEG_DYN_MAX_LSH_DOWN **&**

Previous state of STATE_LSH_DOWN[i] = LSH_POW_CTL

Transition actions:

STATE_LSH_DOWN[i] = LSH_POW_CTL

<u>Condition 11</u>:"LSH_TEMP_PROT to LSH_POW_RISE"

TEG_DYN_LS_DOWN[i] ≤ C_TEG_DYN_MAX_LSH_DOWN **&**

Previous state of STATE_LSH_DOWN[i] ≠ LSH_POW_CTL

Transition actions:

T_POW_RISE_LSH_DOWN[i] = 0

V_EFC_CLC_LSH_DOWN[i] = IP_V_EFC_INI_LSH_DOWN (TEMP_INI_LS_DOWN[i])

% IP_ ... (TEMP_INI_LS_DOWN[i]) *same value as at start*

T_V_EFC_TOL_LSH_DOWN[i] = 0

T_V_EFC_LIM_LSH_DOWN[i] = 0

LV_V_EFC_LIM_BOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0

LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 0

STATE_LSH_DOWN[i] = LSH_POW_RISE

Note: Should none of the conditions have been determined to be met, the state machine shall remain in the same state.

16.22.2 Oxygen sensor heater voltage protection

Prolonged exposure to heater voltages that exceed the absolute maximum ratings specified shall be prevented. The preventative measures shall be applicable to all heater states other than LSH_OFF.

```
If      STATE_LSH_DOWN[i] ≠ LSH_OFF
then    If      V_EFC_CLC_LSH_DOWN[i] ≥ C_V_EFC_TOL_LSH_DOWN
        then    If      T_V_EFC_TOL_LSH_DOWN[i] ≥
                                C_T_MAX_V_EFC_TOL_LSH_DOWN
                then    If      T_V_EFC_LIM_LSH_DOWN[i] ≥
                                C_T_MAX_V_EFC_LIM_LSH_DOWN
                        then    LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0
```

```
                        Reset timer T_V_EFC_LIM_LSH_DOWN[i]
                        Reset timer T_V_EFC_TOL_LSH_DOWN[i]
                else    LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 1
                        Increment timer T_V_EFC_LIM_LSH_DOWN[i]
                endif
        else    LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0
                Reset timer T_V_EFC_LIM_LSH_DOWN[i]
                Increment timer T_V_EFC_TOL_LSH_DOWN[i]
        endif
else    LV_V_EFC_LIM_TOL_LSH_DOWN[i] = 0
        If      T_V_EFC_LIM_LSH_DOWN[i] = 0
        or      T_V_EFC_LIM_LSH_DOWN[i] ≥
                        C_T_MAX_V_EFC_LIM_LSH_DOWN
        then    Reset timer T_V_EFC_LIM_LSH_DOWN[i]
                Reset timer T_V_EFC_TOL_LSH_DOWN[i]
        else    Increment timer T_V_EFC_LIM_LSH_DOWN[i]
        endif
endif
If      T_POW_RISE_LSH_DOWN[i] ≥
                C_T_MAX_V_EFC_MAX_ST_LSH_DOWN
or      (T_AST ≥ C_T_AST_MAX_LSH_DOWN
and     STATE_LSH_DOWN[i] = "LSH_POW_CTL")
then    If      V_EFC_CLC_LSH_DOWN[i] > VB
        or      V_EFC_CLC_LSH_DOWN[i] > C_V_EFC_MAX_LSH_DOWN
        then    LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 1
                V_EFC_CLC_LSH_DOWN[i] =
                                MIN (VB; C_V_EFC_MAX_LSH_DOWN)
        else    LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 0
        endif
else    If      V_EFC_CLC_LSH_DOWN[i] > VB
        or      V_EFC_CLC_LSH_DOWN[i] > C_V_EFC_MAX_ST_LSH_DOWN
        then    LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 1
                V_EFC_CLC_LSH_DOWN[i] =
                                MIN (VB, C_V_EFC_MAX_ST_LSH_DOWN)
        else    LV_V_EFC_LIM_MAX_LSH_DOWN[i] = 0
        endif
```

```
        endif
        If      T_V_EFC_LIM_LSH_DOWN[i] <> 0
        then    V_EFC_LSH_DOWN[i] =
                        MIN (V_EFC_CLC_LSH_DOWN[i], C_V_EFC_LIM_LSH_DOWN)
        else    V_EFC_LSH_DOWN[i] = V_EFC_CLC_LSH_DOWN[i]
        endif
endif
```

16.22.3 Oxygen sensor heater driver duty cycle computation

The PWM duty cycle for the appropriate heater driver shall continually be calculated in all heater states at the specified recurrence.

```
If      STATE_LSH_DOWN[i] = LSH_OFF
then    LSHPWM_DOWN[i] = LSHPWM_EXT_LS_DOWN[i]
else    If      VB ≥ C_VB_MAX_PROT_LSH_DOWN
        then    LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 1
                LSHPWM_DOWN[i] = C_V_EFC_PROT_VB_LSH_DOWN$^2$ * 100 / VB$_N^2$
        else    LV_V_EFC_LIM_PROT_VB_LSH_DOWN[i] = 0
                LSHPWM_DOWN[i] = V_EFC_LSH_DOWN[i]$_N^2$ *
                                C_FAC_V_EFC_AS_LSH_DOWN *100 / VB$_N^2$
                If      LSHPWM_DOWN[i] ≥ NC_LSHPWM_TOL_LSH_DOWN
                then    LSHPWM_DOWN[i] = NC_LSHPWM_TOL_LSH_DOWN
                else    If      LSHPWM_DOWN[i] ≤ NC_LSHPWM_BOL_LSH_DOWN
                        then    LSHPWM_DOWN[i] = NC_LSHPWM_BOL_LSH_DOWN
                        else    LSHPWM_DOWN[i] = LSHPWM_DOWN[i]
                        endif
                endif
        endif
endif.
ACTION_INFR_SetPwmLshDown(
                IN      i
                IN      LSHPWM_DOWN[i])
```

The value C_FAC_V_EFC_AS_LSH_DOWN is a correction signal generated in the application system.

NOTE: The recurrence of the heater driving duty cycle shall occur at the same recurrence as the Vbatt acquisition. This shall prevent short duration voltage drop-outs during engine start

from causing long duration Vbatt corrections being made where the Vbatt has since recovered. This may otherwise cause excessive current to flow in the heater driver.

The computation of the duty cycle shall be checked to ensure that the result remains positive (i.e. $\geq$ 0) at all times and that now under- or overflows are caused by the values of the calibration system constants.

The multiplication of the corrected voltage ratio by 100 to obtain a unit of percent (%) may not necessarily be implemented in the SW but in the data bank definition of the variables.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_V_EFC_INI_LSH_DOWN	6	0...FFH	0...25.89843	0.1015625	[V]
LDPM_TEMP_INI_LS_DOWN_1_EGCP	6	0...FFFFH	-2048...2047.9375	0.0625	[°C]
Initial effective heater voltage					
IP_V_EFC_CTL_LSH_DOWN	6*6	0...FFH	0...25.89843	0.1015625	[V]
LDPM_N_32_3_EGCP	6	0...FFH	0...8160	32	[rpm]
LDPM_TEG_DYN_LS_DOWN_1_EGCP	6	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Open loop control effective heater voltage					
IP_T_TEMP_THD_LSH_DOWN	6	0...FFFFH	0...6553.5	0.1	[s]
LDPM_TEMP_INI_LS_DOWN_1_EGCP	6	0...FFFFH	-2048...2047.9375	0.0625	[°C]
Duration until critical water splash sensor temperature reached					
IP_FAC_V_EFC_TEG_LSH_DOWN	6	0...FFH	0...3.98437	0.015625	[-]
LDPM_VS_1_EGCP	6	0...FFH	0...255	1	[km/h]
Factor for the heat transfer rate, bank 2 only					
C_V_EFC_INC_LSH_DOWN	1	0...FFH	0...1.625	6.37E-03	[V/100ms]
Effective heater voltage increment to raise temperature					
C_V_EFC_DEC_LSH_DOWN	1	0...FFH	0...1.625	6.37E-03	[V/100ms]
Effective heater voltage decrement to lower temperature					
C_V_EFC_RED_LSH_DOWN	1	0...FFH	0...25.89843	0.1015625	[V]
Reduced effective heater voltage during danger of water splash damage					
C_V_EFC_STEP_LSH_DOWN	1	80...7FH	-13...12.89843	0.1015625	[V]
Additional effective heater voltage step on transition from LSH_POW_RED to LSH_POW_RISE					
C_T_MAX_POW_RISE_LSH_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Duration of open loop pre-heating in LSH_POW_RISE from dew-point to set-point temperature					
C_T_MAX_V_EFC_TOL_LSH_DOWN	1	0...FFH	0...25.5	0.1	[s]
Maximum permitted duration where excessive effective heater voltage tolerated					
C_T_MAX_V_EFC_LIM_LSH_DOWN	1	0...FFH	0...25.5	0.1	[s]
Maximum duration of effective heater voltage limiting					
C_T_AST_MAX_LSH_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Time after start at which step in max. effective heater voltage applied, if LSH_POW_CTL active (T_POW_RISE_LSH_DOWN[i] stopped)					
C_T_MAX_V_EFC_MAX_ST_LSH_DOWN	1	0...FFH	0...25.5	0.1	[s]
Time after start at which step in max. effective heater voltage applied					
C_V_EFC_TOL_LSH_DOWN	1	0...FFH	0...25.89843	0.1015625	[V]
Effective heater voltage threshold above which voltage limiting may occur if threshold persistently exceeded					
C_V_EFC_LIM_LSH_DOWN	1	0...FFH	0...25.89843	0.1015625	[V]
Effective heater voltage limit under persistent excessive heater voltage condition					
C_VB_MAX_PROT_LSH_DOWN	1	0...FFH	0...25.89843	0.1015625	[V]
Maximum permitted battery voltage threshold, over which overvoltage detected					
C_V_EFC_PROT_VB_LSH_DOWN	1	0...FFH	0...25.89843	0.1015625	[V]
Effective heater voltage in the case of battery voltage overvoltage condition					
C_V_EFC_MAX_LSH_DOWN	1	0...FFH	0...25.89843	0.1015625	[V]
Absolute maximum permitted effective heater voltage after initial phase; threshold & limit					
C_V_EFC_MAX_ST_LSH_DOWN	1	0...FFH	0...25.89843	0.1015625	[V]
Absolute maximum permitted effective heater voltage during initial phase; threshold & limit					
C_V_EFC_AS_LSH_DOWN	1	80...7FH	-13...12.89843	0.1015625	[V]

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
Effective heater voltage calibration system additive factor					
C_FAC_V_EFC_AS_LSH_DOWN	1	0...FFH	0...1.99218	0.0078125	[-]
Multiplicative correction factor to effective heater voltage					
C_TEG_DYN_MAX_LSH_DOWN	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Maximum permitted exhaust gas temperature above which LSH switched off					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_LSHPWM_TOL_LSH_DOWN	1	0...FFH	0...99.60937	0.390625	[%]
Heater driver PWM duty cycle upper limit					
NC_LSHPWM_BOL_LSH_DOWN	1	0...FFH	0...99.60937	0.390625	[%]
Heater driver PWM duty cycle lower limit					

16.22.4 Application incidences for downstream oxygen sensor heater management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating project specific application conditions for heating not met					
LSHPWM_EXT_LS_DOWN[NC_CBK_EX_NR]	V/O	0...FFH	0...99.60937	0.390625	[%]
Downstream heater PWM duty cycle preset from external tester					
TEMP_INI_LS_DOWN[NC_CBK_EX_NR]	V/O	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Oxygen sensor temperature at start					

Input data:

LV_VAR_LSH_DOWN	NC_CBK_EX_NR	TEG_WALL_CAT_DOWN_MDL[NC_CBK_EX_NR]	LV_IGK
LV_LSHPWM_DOWN_EXT_ADJ[NC_CBK_EX_NR]	LSHPWM_DOWN_EXT_ADJ[NC_CBK_EX_NR]	NC_LSHPWM_BOL_LSH_DOWN	

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Description:

The Boolean constant LC_LSH_DOWN[i] shall permit the appropriate oxygen sensor to be heated. When set, the Boolean inhibit flag LV_INH_LSH_DOWN[i] shall be reset. When reset, LV_INH_LSH_DOWN[i] shall be set.

The oxygen sensor temperature at start TEMP_INI_LS_DOWN[i] shall be obtained from the input variable TEG_WALL_CAT_DOWN_MDL[i] (this is the exhaust duct temperature downstream) every time an engine start is finished.

All actions beside the initialisation routine are done only, if the variant ‘with monitor lambda sensors’ is on the table.

Application conditions:

Activation:	LV_IGK = 1
Deactivation:	LV_IGK = 0
Recurrence:	1 s

Initialisation:

The following variable initialisation shall take place at the beginning of a new driving cycle:

LV_INH_LSH_DOWN[i] = 0
LSHPWM_EXT_LS_DOWN[i] = 0
TEMP_INI_LS_DOWN[i] = 0 °C

Formula section:

```
IF      LV_LSHPWM_DOWN_EXT_ADJ[i] = 0
THEN
    if      LV_VAR_LSH_DOWN = 0
    then    LV_INH_LSH_DOWN[i]  =  1
            LSHPWM_EXT_LS_DOWN[i]  = NC_LSHPWM_BOL_LSH_DOWN
    else    LV_INH_LSH_DOWN[i]  =  ! LC_LSH_DOWN[i]
            LSHPWM_EXT_LS_DOWN[i]  =  0
ELSE    LV_INH_LSH_DOWN[i]  =  1
        LSHPWM_EXT_LS_DOWN[i]  = LSHPWM_DOWN_EXT_ADJ[i]
ENDIF
```

Once at every function activation:

TEMP_INI_LS_DOWN[i] = TEG_WALL_CAT_DOWN_MDL[i]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_LSH_DOWN[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Boolean variable to enable oxygen sensor heating					

16.23 O2 sensor Lambda set point intervention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAMB_SP_REQ_DIAG_ACT[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Lambda coordination request to change set point for active plausibility test					
LAMB_SP_DIAG_LS_UP_DOWN[NC_CBK_EX_NR]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda set point for active plausibility test					

Input data:

LAMB_SP_OSC_CHK[NC_CBK_EX_NR]	LAMB_SP_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	LAMB_SP_DIAG_OPL_LS_UP_DOWN[NC_CBK_EX_NR]	LV_ST_END

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

This function provides an internal coordination of lambda setpoint requests from functions within EGCP.

Description:

The function shall perform a priority ranking for several lambda setpoint requests established inside the EGCP aggregate and forward one request to functions outside EGCP. The flag LV_LAMB_SP_REQ_DIAG_ACT[i] shall indicate that at least one setpoint request is active. No check should be performed whether the setpoint is reached, nor should a request for open or closed loop lambda control be output. The possible sources of lambda setpoint requests are:

- plausibility check of upstream and downstream O2 sensor,
- monitor for downstream O2 sensor during oscillation check,
- monitor for upstream O2 sensor during open circuit check.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset or at clearing error memory initialize all variables and bits with 0, with the expection of:
LAMB_SP_DIAG_LS_UP_DOWN[i] which is initialized with 1.

Recurrence:

This function shall be carried out every 100 ms.

Activation:

LV_ST_END = 1

Formula section:

```
% priority is open loop check > oscillation check > open circuit check
If    LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] > 0
then
      LAMB_SP_DIAG_LS_UP_DOWN[i] = LAMB_SP_DIAG_OPL_LS_UP_DOWN[i]
      LV_LAMB_SP_REQ_DIAG_ACT[i] = 1

elseif        LAMB_SP_OSC_CHK[i] > 0
then
      LAMB_SP_DIAG_LS_UP_DOWN[i] = LAMB_SP_OSC_CHK[i]
      LV_LAMB_SP_REQ_DIAG_ACT[i] = 1

elseif        LAMB_SP_DIAG_OC_LSL_UP[i] > 0
and           LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] = 0 for all i
then
      LAMB_SP_DIAG_LS_UP_DOWN[i] = LAMB_SP_DIAG_OC_LSL_UP[i]
      LV_LAMB_SP_REQ_DIAG_ACT[i] = 1

else
      LAMB_SP_DIAG_LS_UP_DOWN[i] = 1
      LV_LAMB_SP_REQ_DIAG_ACT[i] = 0
endif
```

16.24 Air fuel cycle evaluation and calculation of dwell times

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_AFL[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating detection of lean Air-Fuel mixture when set					
LV_AFL_OLD_1_EGCP[NC_CBK_EX_NR]	-	0...1H	0...1	1	[-]
Auxiliary flag indicating lean Air-Fuel mixture status of previous sample step					
CTR_AFL_CYC[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Counter indicating number of air fuel cycles					
T_AFR_CYC[NC_CBK_EX_NR]	V/O	0...FFFFH	0...655.35	0.01	[s]
Rich mixture cycle time					
T_AFL_CYC[NC_CBK_EX_NR]	V/O	0...FFFFH	0...655.35	0.01	[s]
Lean mixture cycle time					
T_AFL_CYC_HLD[NC_CBK_EX_NR]	V/O	0...FFFFH	0...655.35	0.01	[s]
Lean mixture cycle time, held until next new value available					
T_SUM_AFL_AFR_CYC[NC_CBK_EX_NR]	V/O	0...FFFFH	0...655.35	0.01	[s]
Lean plus rich mixture cycle time; time of last forced lambda stimulation period					

Input data:

LV_ST_END	NC_CBK_EX_NR
LAMB_PLS[NC_CBK_EX_NR]	LV_LAMB_PLS_ACT[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.

For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2,

otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Signal flow diagram:

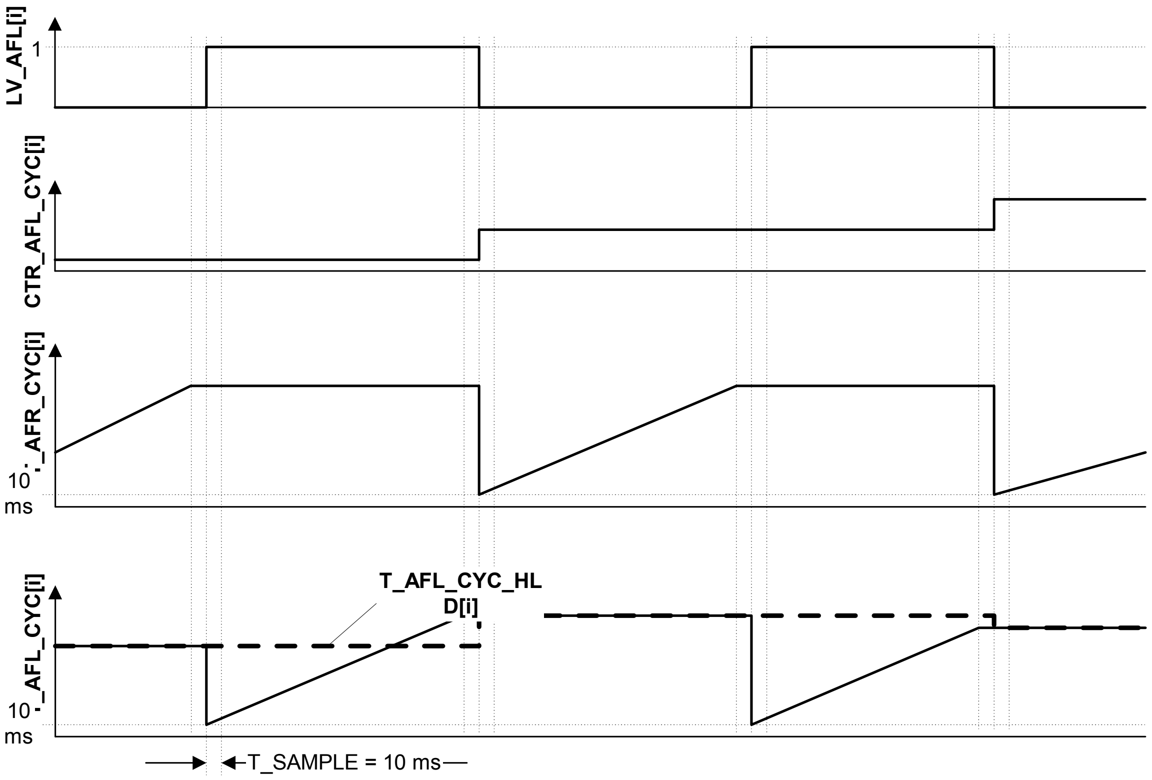


Figure 59: Definition of dwell times

Description:

The logical variable LV_AFL_i that indicates a lean air fuel ratio (lambda set point is lean) is calculated based on the output of the forced lambda stimulation. In case of a positive LAMB_PLS[i] the flag is set to one. The lean dwell time T_AFL_CYC[i] is calculated in that case. If LAMB_PLS[i] is negative or zero LV_AFL[i] is 0. In that case the rich dwell time T_AFL_CYC[i] is calculated.

When LV_AFL[i] switches from 0 to 1 T_AFL_CYC[i] is reset. When LV_AFL[i] switches from 1 to 0 T_AFR_CYC[i] is reset and the current value of T_AFL_CYC[i] is stored in the variable T_AFL_CYC_HLD[i]

The function is only active when the forced lambda stimulation is active (LV_LAMB_PLS_ACT[i] = 1). For further explanation see Figure 59.

T_SUM_AFL_AFR_CYC[i] is the period time of the forced lambda stimulation. It is updates each rising edge of LV_AFL[i].

Application conditions:

Initialisation: at engine start (LV_ST_END = 0 -> 1) and at each function activation:

LV_AFL[i] = 0
LV_AFL_OLD_1_EGCP[i] = 0
CTR_AFL_CYC[i] = 0
T_AFR_CYC[i] = T_SAMPLE
T_AFL_CYC[i] = T_SAMPLE
T_AFL_CYC_HLD[i] = T_SAMPLE
T_SUM_AFL_AFR_CYC[i] = T_SAMPLE

Recurrence: T_SAMPLE = 10 ms

Activation: LV_LAMB_PLS_ACT[i] = 1

Deactivation: -

Formula section:

```
IF LAMB_PLS[i] > 0    /* lean */
THEN
   LV_AFL[i] = 1
   IF LV_AFL_OLD_1_EGCP[i] = 0    /* rising edge LV_AFL[i] = 0 -> 1 */
   THEN
      T_SUM_AFL_AFR_CYC[i] = T_AFL_CYC_HLD[i] + T_AFR_CYC[i]
      T_AFL_CYC[i] = T_SAMPLE
   ELSE
      T_AFL_CYC[i] = T_AFL_CYC[i] + T_SAMPLE
   ENDIF
ELSE    /* rich */
   LV_AFL[i] = 0
   IF LV_AFL_OLD_1_EGCP[i] = 1    /* falling edge LV_AFL[i] = 1 -> 0 */
   THEN
      T_AFR_CYC[i] = T_SAMPLE
      T_AFL_CYC_HLD[i] = T_AFL_CYC[i]
      CTR_AFL_CYC[i] = CTR_AFL_CYC[i] + 1
   ELSE
      T_AFR_CYC[i] = T_AFR_CYC[i] + T_SAMPLE
   ENDIF
ENDIF

LV_AFL_OLD_1_EGCP[i] = LV_AFL[i]
```

16.25 Application Incidencies of the “WRAF sensor electric diagnosis”

16.25.1 Application Incidences of “Electrical Fault Detection”

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_EL_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "HW-Based Diagnosis"					

Input data:

LV_ST_END	NC_CBK_EX_NR	LV_IGK	LV_CDN_VB_OBD1
LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	LV_LS_UP_OBD_1_MAN_DEAC[NC_CBK_EX_NR]	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]
STATE_LSL_IF_SPI_RD[NC_CBK_EX_NR]			

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Application conditions:

Initialization:

The flag shall be initialized as follow during the transition LV_IGK = 0 to 1.

LV_INH_DIAG_EL_LSL_UP[i] = 0

Recurrence:

The function shall be carried out once every 10ms.

Activation:

Transition LV_IGK = 0 ->1

Deactivation:

Transition LV_IGK = 1 ->0. LV_INH_DIAG_EL_LSL_UP[i] shall be set to 1.

Formula section:

```
If    LV_CDN_VB_OBD1 = 0                        or
      LV_ERR_LSL_UP_IF[i] = 1                   or
      LV_LS_UP_OBD_1_MAN_DEAC[i] = 1            or
      LV_ERR_OFS_LSL_UP[i] = 1                  or
      LV_ERR_CTL_LSL_UP[i] = 1                  or
      STATE_LSL_IF_SPI_RD[i] bit LSL_ENA = 0
Then
      LV_INH_DIAG_EL_LSL_UP[i] = 1
Else
      LV_INH_DIAG_EL_LSL_UP[i] = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_DIAG_EL_LSL_UP	1	0...FFH	0...255	1	[-]
Increment of OBD-I Diagnosis debouncer counter					
C_ABC_MAX_DIAG_EL_LSL_UP	1	1...FFH	1...255	1	[-]
Maximum threshold of OBD-I Diagnosis debouncer counter					

Application assistances:

The following list shows the influenced functions if a failure is detected.

Direct error reactions:

The following functions have to be direct deactivated:

- catalyst efficiency diagnosis
- diagnosis of the canister purge system (flow rate diagnosis)
- secondary air diagnosis (flow rate)
- secondary air valve diagnosis (mechanical)
- secondary air pump
- upstream oxygen sensor temperature acquisition
- WRAF Sensor controller diagnosis
- WRAF Sensor Plausibility Diagnosis
- WRAF Sensor dynamic diagnosis
- WRAF Sensor not mounted diagnosis
- WRAF Sensor plausibility diagnosis during PUC
- WRAF Sensor false start diagnosis
- WRAF Sensor heater coupling diagnosis
- WRAF Sensor OBD2 heater diagnosis

The following function has to be activated :

- canister purge function (min. operation)

Diagnostic XX	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
EL_LSL_UP[NC_CBK_EX_NR]	Short Circuit to Plus	*SYM_0*					MIL_ON / LAW_M	CARB_LS
	Short Circuit to Ground	*SYM_1*						

DTC are defined in the table of failures

16.25.2 WRAF Sensor Pump Current Controller Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_IPLSL_DIAG_INH_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "HW-Based Diagnosis with the CJ120"					

Input data:

LV_ST_END	NC_CBK_EX_NR	LV_IGK	LV_CDN_VB_OBD1

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1, for exhaust cylinder bank 1
 i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1, for single exhaust cylinder bank.

Application conditions:

Initialization:

The flag shall be initialized as follow during the transition LV_IGK = 0 to 1.

LV_IPLSL_DIAG_INH_LSL_UP[i] = 0

Recurrence:

The function shall be carried out once every 10ms.

Activation:

Transition LV_IGK = 0 ->1

Deactivation:

Transition LV_IGK = 1 ->0. LV_IPLSL_DIAG_INH_LSL_UP[i] shall be set to 1.

Formula section:

```
If   LV_CDN_VB_OBD1 = 0
Then
    LV_IPLSL_DIAG_INH_LSL_UP[i] = 1
Else
    LV_IPLSL_DIAG_INH_LSL_UP[i] = 0
Endif
```

16.26 WRAF Sensor Controller Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Present error in the CJ120 WRAF sensor controller					
LV_CDN_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Conditions to run diagnosis of CJ120 WRAF sensor controller are fullfiled					
LV_END_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Diagnostic of CJ120 WRAF sensor controller available					
ERR_SYM_LSL_UP_IF[NC_CBK_EX_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symtom of CJ120 WRAF sensor controller					
CTR_ERR_SAVE_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Number of communication errors during the WRAF sensor controller initialization					
CTR_DELTA_LSL_UP_IF[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Increment of error counter since initialization respectively since last increment of error counter					

Input data:

LV_INH_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	LV_IGK	LV_LSL_IF_SPI_RST_END[NC_CBK_EX_NR]	CTR_ERR_LSL_IF_SPI_WR[NC_CBK_EX_NR]
NC_CBK_EX_NR			

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

This function shall detect an error during the initialization and/or operation of a WRAF sensor controller which uses a SPI comunication, like the CJ120 or ATIC42. For this task informations from the Basic Software (BSW) concerning the initialization and the communication between ASW and the controller are used to judge if the ASIC works properly.

After an ECU reset, the WRAF sensor controller is started and the diagnosis determines the time until the initialization of the ASIC has been performed (LV_LSL_IF_SPI_RST_END[i] = 1). If this doesn't happen after a certain time (C_T_RST_MAX_DIAG_LSL_UP_IF[I]) an error is present and the flag LV_ERR_LSL_UP_IF[i] shall be set. If the initialization occurs successfully, the difference is checked between the present error counter (CTR_ERR_LSL_IF_SPI_WR[i]) and the stored value of this error counter (CTR_ERR_SAVE_DIAG_LSL_UP_IF[i], stored at ECU reset, LV_IGK switch from 0 -> 1 or

at clearing error memory and also after each function call in case that a difference between both counters was found). If there is a difference, another counter is incremented.
If this counter is bigger than a threshold, a SPI communication error is entered.

In any case the flag LV_ERR_LSL_UP_IF[i] = 1 the operation of the WRAF sensor shall be inhibited.

Application conditions:

Initialisation:

At transition LV_IGK = 0 -> 1, at clearing error memory or at ECU reset,

LV_ERR_LSL_UP_IF[i] = 0
LV_END_DIAG_LSL_UP_IF[i] = 0
ERR_SYM_LSL_UP_IF[i] = 0
CTR_DELTA_LSL_UP_IF[i] = 0
CTR_ERR_SAVE_DIAG_LSL_UP_IF[i] = CTR_ERR_LSL_IF_SPI_WR[i]

Start TIMER_1 and TIMER_2

Recurrence:

The function shall run every 10 ms

Activation:

At LV_IGK=1 & LV_INH_DIAG_LSL_UP_IF[i]=0 the function shall be started and LV_CDN_DIAG_LSL_UP_IF[i] = 1

Deactivation:

The deactivation occures if the activation conditions above are not fulfilled.

Formula section:

```
If  LV_LSL_IF_SPI_RST_END[i] = 1        /init. of WRAF sensor controller done successfully
Then
    Stop TIMER_1

    If      CTR_ERR_LSL_IF_SPI_WR[i] =! CTR_ERR_SAVE_DIAG_LSL_UP_IF[i]
    then    CTR_DELTA_LSL_UP_IF[i]$_{n}$ = CTR_DELTA_LSL_UP_IF[i]$_{n-1}$ + 1
            CTR_ERR_SAVE_DIAG_LSL_UP_IF[i] = CTR_ERR_LSL_IF_SPI_WR[i]

    If      CTR_DELTA_LSL_UP_IF[i]  > C_CTR_MAX_SPI_COM_LSL_UP[I]
    Then
            ERR_SYM_LSL_UP_IF[i] = “sym_1”
            LV_ERR_LSL_UP_IF[i] = 1
            LV_END_DIAG_LSL_UP_IF[i] = 1
    Endif
Else
```

```
        If  TIMER_1 > C_T_RST_MAX_DIAG_LSL_UP_IF[I]
        Then
                ERR_SYM_LSL_UP_IF[i] = “sym_0”          /no init. of WRAF sensor controller done
                LV_ERR_LSL_UP_IF[i] = 1
                LV_END_DIAG_LSL_UP_IF[i] = 1
        Else
                Increment TIMER_1
        Endif
Endif

If  LV_END_DIAG_LSL_UP_IF[i] = 0
Then
        If  TIMER_2 < C_T_END_DIAG_LSL_UP_IF[I]
        Then Increment TIMER_2
        Else
                LV_END_DIAG_LSL_UP_IF[i] = 1            /for the diagnosis readines
                Stop TIMER_2
        Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_MAX_SPI_COM_LSL_UP[NC_CBK_EX_NR]	1	0...FFH	0...255	1	[-]
Max. allowed number of SPI comunication errors					
C_T_RST_MAX_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	1	0...FFFFH	0...655.35	0.01	[s]
Max. waiting time for the initialization of the WRAF sensor controller					
C_T_END_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	1	0...FFFFH	0...655.35	0.01	[s]
Time to diagnostic availability					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
LSL_UP_IF[NC_CBK_EX_NR]	wrong initialisation of WRAF Sensor controller	SYM_0	NO
	SPI communication error	SYM_1	

16.27 Application Incidences of “WRAF Sensor Controller Diagnosis”

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibit diagnosis					

Input data:

LV_IGK	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	NC_CBK_EX_NR
LV_LS_UP_OBD_1_MAN_DEAC[NC_CBK_EX_NR]			

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

This function shall set the inhibit bit LV_INH_DIAG_LSL_UP_IF[i] in case an electrical fault of the WRAF sensor. In this case, i.e., when LV_ERR_EL_LSL_UP[I] = 1, the communication between sensor and ASIC is stopped, therefore doesn’t makes sense to run the ASIC diagnosis.

Application conditions:

Initialisation:

At the transition LV_IGK = 0 -> 1
LV_INH_DIAG_LSL_UP_IF[i] = 0

Recurrence:

Every 10 ms

Activation:

At LV_IGK = 1

Deactivation:

If LV_ERR_LSL_UP_IF[i] = 1

Formula section:

```
If  LV_ERR_EL_LSL_UP[I] = 1
or LV_LS_UP_OBD_1_MAN_DEAC[i] = 1
Then
      LV_INH_DIAG_LSL_UP_IF[i] = 1
Endif
```

Configuration data:

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
WRAF Sensor Controller Diagnosis	*Sym_0*	*0*	P 0606	--	---		MIL CARB_H
	Sym_1	*1*		--			
IF_LSL_UP[i]							

16.28 O2 sensor (lin) open circuit diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Debounced open circuit failure detected					
LV_CDN_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Bit indicating the necessary conditions to run the diagnosis are fullfiled					
LV_END_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Bit indicating the diagnostic is valid					
ERR_SYM_OC_LSL_UP[NC_CBK_EX_NR]	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom indicating open circuit failure in one of the lines was detected					
LV_VLS_UP_MIN_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Bit indicating LSL undervoltage					
LV_VLS_UP_AFS_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Bit indicating LSL stoichiometric voltage					
STATE_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	O/V	0H 1H 2H	PASSIVE ACTIVE END	1	[-]
Status of the LSL Open Circuit Diagnosis					
T_VLS_UP_MIN_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for LSL signal under voltage					
T_VLS_UP_AFS_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for lambda signal being in stoichiometric band while it shouldn't					
T_DIAG_OC_RST_OSC_DI[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Time to reactivate oscillator for upstream tip temperature measurement					
T_WAIT_SYM_OBD_2_LSH_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Waiting time to check if OBD2 upstream heater error is present (to distinguish OC on Vip from Vg)					
LV_DIAG_OC_IF_OSC_DI_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Request stop oscillator used to determine sensor internal resistance					
T_DIAG_OC_RST_LAMB_SP_REQ[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Time to finish the active check for OC in VIP / VG					
T_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer to finish OC diagnosis in case of OBD2 upstream heater error symptom active and conditions for plausibility diagnosis fulfilled					
LAMB_SP_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	O/V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Requested lambda setpoint for active part of linear sensor open circuit check					
LAMB_SP_DELTA_OC_LSL_UP[NC_CBK_EX_NR]	V	8000...7FFFH	-32...31.99902	0.9766e-3	[-]
Lambda setpoint ramp increment for active part of linear sensor open circuit check					

Input data:

LV_DIAG_OBD_SYM_LSH_UP[NC_CBK_EX_NR]	LV_DIAG_OBD_END_LSH_UP[NC_CBK_EX_NR]	LV_DIAG_RAW_ACT_END_LSL_UP[NC_CBK_EX_NR]	LV_DIAG_ACT_END_LSL_UP[NC_CBK_EX_NR]
LV_DIAG_PUC_SYM_LSL_UP[NC_CBK_EX_NR]	LV_DIAG_PUC_END_LSL_UP[NC_CBK_EX_NR]	STATE_SYM_DIAG_PUC_LSL_UP[NC_CBK_EX_NR]	
LV_DIAG_ACT_SYM_LSL_UP[NC_CBK_EX_NR]	LV_IPLSL_VLD[NC_CBK_EX_NR]	LV_INH_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	LV_IGK
VLS_UP_DIAG[NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_ES	LV_DIAG_EOL_END_LS_UP_DOWN[NC_CBK_EX_NR]
LV_DIAG_RAW_ACT_SYM_LSL_UP[NC_CBK_EX_NR]	LAMB_SP[NC_CBK_EX_NR]	LV_DIAG_OBD_CDN_LSH_UP[NC_CBK_EX_NR]	LV_R_IT_OSC_ENA_LSL_IF[NC_CBK_EX_NR]
		LV_DIAG_ACT_SHO_CDN_LSL_UP[NC_CBK_EX_NR]	

FUNCTION DESCRIPTION:

General information:

This function shall use some debounced OBD2 lambda sensor faults to evaluate if an open circuit in any of the four electric lines (VN, VG, VIP and VRc) is present. The matrix below shows reaction of sensor/controller to each open line.

Open Circuit on Line	Measured Sensor Temp	Lambda Sensor
VN	**V_Ri = 5 V (Ttip = 560 °C)**	**U_Lambda = 0 V** **(for NTK sensor only if R_I oscillator is turned off)**
VG	**V_Ri = 0 V (Ttip = 880 °C)** **(only applicable for some senosor types)**	**U_Lambda = 2.0 V** **(sensor stuck)**
VIP	**V_Ri plausible**	**U_Lambda = 2.0 V** **(sensor stuck)**
VRc	**V_Ri plausible**	**U_Lambda > 4.8 V in air**

This function shall run only if at least one of the following diagnoses has set its readiness bit: Activity Check, Plausibility during PUC and Sensor Heater OBD2. The function shall go to the state = “active” if one of the above diagnoses debounced a fault.

The end flag for this diagnosis is set if all the related diagnosis functions have set their end flags or after changing to the STATE “end” with detection of an open circuit error.

OC in the VN line (LSU and ZFAS-Ux sensors) or VG line (ZFAS-Ux only) cause an OBD2 heater symptom. Therefore an OC error is assumed if a heater symptom is accompanied by other indications of these OC errors. For OC in VN, a very low VLS_UP_DIAG[i] will occur (for ZFAS-Ux, only if the Ri measurement oscillator is turned off), while for OC in VG this value will be around 2.0V (i.e. stuck at lambda=1). These conditions are tested with an open loop lambda setpoint shift requested here, monitoring of VLS_UP_DIAG[i] is coordinated with the detection of shifted lambda setpoint and disabled Ri measurement oscillator. If the lambda shift request is not fulfilled, the diagnosis looks for an activity symptom instead to decide if the VG/VIP line is open, but this probably takes much more time.

If VLS_UP_DIAG[i] is stuck at 2.0V, as detected by the PUC plausibility or activity diagnoses, OC in VG or VIP may be present. In ZFAS-Ux sensors, OC in VG would also cause a heater symptom, so further fault isolation is possible using T_WAIT_SYM_OBD_2_LSH_UP[i] to wait for this symptom. If C_T_WAIT_SYM_OBD_2_LSH_UP is set to 0, all faults causing VLS_UP_DIAG[i]= 2.0V are regarded as OC in VIP, even if the test was triggered by an OBD2 heater symptom.

Quite independently, OC in VRC shows up as a VLS_H_OC symptom in the PUC plausibility diagnosis and is consumed here. It will be overridden however by any other OC fault.

If any open circuit symptom could be detected the flag LV_ERR_OC_LSL_UP[i] shall be set to 1.

If two separate catalyst systems are concerned, then
    i = 1, for cylinder bank 1
    i = 2, for cylinder bank 2,
else
    i = 1.

Description:

Application conditions:

Initialisation: At LV_IGK = 0 to 1, reset ***or*** *at clearing error memory:*
Reset all variables

Recurrence: This function shall run every 100 ms.

Activation / Deactivation:

```
If  ( LV_DIAG_ACT_END_LSL_UP[i] = 1      or
      LV_DIAG_OBD_END_LSH_UP[i] = 1      or
      LV_DIAG_PUC_END_LSL_UP[i] = 1)
and LV_INH_DIAG_OC_LSL_UP[i] = 0
and LV_IPLSL_VLD[i] = 1
then
   LV_CDN_DIAG_OC_LSL_UP[i] = 1
   % activate function ( = carry out formula section)
else
   LV_CDN_DIAG_OC_LSL_UP[i] = 0
   LAMB_SP_DIAG_OC_LSL_UP[i] = 0
   LV_DIAG_OC_IF_OSC_DI_LSL_UP[i] = 0
   % deactivate function
endif
```

Formula section:

```
%Readiness bit setting in case no failure is present
If   % during normal driving cycle
   { LV_DIAG_OBD_SYM_LSH_UP[i] = 0   &
     LV_DIAG_ACT_SYM_LSL_UP[i] = 0    &
     LV_DIAG_PUC_SYM_LSL_UP[i] = 0    &
     LV_DIAG_ACT_END_LSL_UP[i] = 1    &
     LV_DIAG_OBD_END_LSH_UP[i] = 1   &
     (LV_DIAG_PUC_END_LSL_UP[i] = 1 or LC_DIAG_PUC_END_MAN_LSL_UP = 1)            &
     LV_END_DIAG_OC_LSL_UP[i] = 0 }
Or % during EOL test
     { LV_DIAG_EOL_END_LS_UP_DOWN[i] = 1 & LV_DIAG_PUC_END_LSL_UP[i] = 1 }
Then
        ERR_SYM_OC_LSL_UP[i] = NO_SYM
        LV_END_DIAG_OC_LSL_UP[i] = 1
Endif
```

DEFINITION OF THE STATES

State “PASSIVE”

Action:

None

Transition to other states

Condition 1: to state “ACTIVE”

```
If      LV_DIAG_OBD_SYM_LSH_UP[i] = 1 or
        LV_DIAG_ACT_SYM_LSL_UP[i] = 1 or
        LV_DIAG_PUC_SYM_LSL_UP[i] = 1
then
     STATE_DIAG_OC_LSL_UP[i] = “ACTIVE”
endif
```

State “ACTIVE”

Action:

```
%Waiting timer for a heater OBD diagnostic
If      ( ( LV_DIAG_ACT_SYM_LSL_UP[i] = 1 or LV_DIAG_PUC_SYM_LSL_UP[i] = 1)
         and LV_DIAG_OBD_CDN_LSH_UP[i] = 1 )
then
         increment T_WAIT_SYM_OBD_2_LSH_UP[i]
endif
```

```
% If a heater OBD2 symptom is present, turn off Ri oscillator and check sensor signal
% with an active lambda shift.
If(1)   LV_DIAG_OBD_SYM_LSH_UP[i] = 1
then
% Check for additional symptoms of OC faults.
    % Request to turn off Ri oscillator, if the test has not timed out:
    If(2a)      T_DIAG_OC_RST_OSC_DI[i] < C_T_DIAG_OC_RST_OSC_DI
    then
                LV_DIAG_OC_IF_OSC_DI_LSL_UP[i] = 1
    Else
                LV_DIAG_OC_IF_OSC_DI_LSL_UP[i] = 0
    Endif(2a)
    % If Ri oscillator is off, do test for Vn symptom
    If(2b)      LV_R_IT_OSC_ENA_LSL_IF[i] = 0
    then
        increment T_DIAG_OC_RST_OSC_DI[i]
        If(3b) VLS_UP_DIAG[i] < C_VLS_UP_DIAG_MIN_OC_LSL_UP
        then % VLS_UP_DIAG symptom for OC@VN
            If(4b)      T_VLS_UP_MIN_DIAG_OC_LSL_UP[i] >=
                        C_T_VLS_UP_MIN_DIAG_OC_LSL_UP
            then
                        LV_VLS_UP_MIN_DIAG_OC_LSL_UP[i] = 1
            Else
                        increment T_VLS_UP_MIN_DIAG_OC_LSL_UP[i]
            Endif (4b)
        Else (3a)
            reset T_VLS_UP_MIN_DIAG_OC_LSL_UP[i]
        Endif (3b)
    Endif (2b)
    %Request lambda setpoint shift, if the test has not timed out:
    If(2c)      T_DIAG_OC_RST_LAMB_SP_REQ[i] < C_T_DIAG_OC_RST_OSC_DI
    then
                execute Section "Moving Lambda setpoint"
    Else
                LAMB_SP_DIAG_OC_LSL_UP[i] = 0
    Endif(2c)
    % If lambda setpoint is shifted far enough, check for open loop test symptom of OC@VG/VIP
    If(2d)      (C_LAMB_SP_MAX_DIAG_OC_LSL_UP < LAMB_SP[i] or
                 LAMB_SP[i] < C_LAMB_SP_MIN_DIAG_OC_LSL_UP)
    then
        increment T_DIAG_OC_RST_LAMB_SP_REQ[i]
        If(3d) C_VLS_UP_MIN_OC_AFS_LSL_UP < VLS_UP_DIAG[i] and
               VLS_UP_DIAG[i] < C_VLS_UP_MAX_OC_AFS_LSL_UP
        then
```

```
                If(4d)      T_VLS_UP_AFS_DIAG_OC_LSL_UP[i] >=
                            C_T_VLS_UP_MIN_DIAG_OC_LSL_UP
                then
                            LV_VLS_UP_AFS_DIAG_OC_LSL_UP[i] = 1
                Else (4d)
                    increment T_VLS_UP_AFS_DIAG_OC_LSL_UP[i]
                Endif (4d)
            Endif (3d)
        Endif (2d)

        % if sensor is active, there cannot be an OC in VG/VIP, even if the lambda setpoint is not shifted

        If(2e)      C_VLS_UP_MIN_OC_AFS_LSL_UP >= VLS_UP_DIAG[i] or
                    VLS_UP_DIAG[i] >= C_VLS_UP_MAX_OC_AFS_LSL_UP
        then
                reset T_VLS_UP_AFS_DIAG_OC_LSL_UP[i]
        Endif (2e)

        % abort open loop if lambda setpoint shift did not occur and activity test would have seen the fault
        % T_DIAG_OC_LSL_UP approximates the time until the activity diagnosis would come to its final decision.

        If (2f)     LV_DIAG_ACT_SHO_CDN_LSL_UP[i] = 1
        then
                    increment T_DIAG_OC_LSL_UP[i]
        Else
                    reset T_DIAG_OC_LSL_UP[i]
        Endif (2f)
    Endif (1)
```

<u>**Transition to other states**</u>

***Condition 2: to state "END"** (Open Circ in line Vn detected)*

```
If      LV_DIAG_OBD_SYM_LSH_UP[i] = 1 and
        LV_VLS_UP_MIN_DIAG_OC_LSL_UP[i] = 1
then
    STATE_DIAG_OC_LSL_UP[i] = "END"
    ERR_SYM_OC_LSL_UP[i]  =  "SYM_0"   / Open Circ in UN"
endif
```

***Condition 3: to state "END"** (Open Circ in line VIP detected)*

```
If      (T_WAIT_SYM_OBD_2_LSH_UP[i] ≥ C_T_WAIT_SYM_OBD_2_LSH_UP) and
        (LV_DIAG_ACT_SYM_LSL_UP[i] = 1 or
          LV_VLS_UP_AFS_DIAG_OC_LSL_UP[i] = 1 or
          (LV_DIAG_PUC_SYM_LSL_UP[i] = 1 and
            STATE_DIAG_PUC_LSL_UP[i] = "VLS_AFS_OC"))
```

```
then
% wait if OBD2 upstream heater error symptom is present – to distinguish between OC on Vip and Vg.
% Only applicable for some sensor types, for others the timer limit should be set zero so that all pump
% circuit faults are presented as OC in VIP.
    STATE_DIAG_OC_LSL_UP[i] = "END"
    ERR_SYM_OC_LSL_UP[i]  =  "SYM_2"  /  Open Circ in VIP"
Endif
```

Condition 4: to state "END" *(Open Circ in line VG detected)*

```
If      LV_DIAG_OBD_SYM_LSH_UP[i] = 1 and
        (LV_VLS_UP_AFS_DIAG_OC_LSL_UP[i] = 1 or
          LV_DIAG_ACT_SYM_LSL_UP[i] = 1 or
          (LV_DIAG_PUC_SYM_LSL_UP[i] = 1 &
            STATE_SYM_DIAG_PUC_LSL_UP[i] = "VLS_AFS_OC"))
then
    STATE_DIAG_OC_LSL_UP[i] = "END"
    ERR_SYM_OC_LSL_UP[i]  =  "SYM_1"  / Open Circ in VG"
endif
```

Condition 5: to state "END" *(Open Circ in line VRc detected)*

```
If     LV_DIAG_PUC_SYM_LSL_UP[i] = 1 and
        STATE_SYM_DIAG_PUC_LSL_UP[i] = "VLS_H_OC"
then
    STATE_DIAG_OC_LSL_UP[i] = "END"
    ERR_SYM_OC_LSL_UP[i]  =  "SYM_3"    / Open Circ in VRc
endif
```

Condition 6: to state "end"

```
% finish diagnosis with "no open circuit error found"
% catch timeouts and secondary symptoms which are proven absent

If      (LV_DIAG_OBD_SYM_LSH_UP[i] = 1 and
         T_DIAG_OC_RST_OSC_DI[i] > C_T_DIAG_OC_RST_OSC_DI and
         (T_DIAG_OC_RST_LAMB_SP_REQ[i] > C_T_DIAG_OC_RST_OSC_DI or
          T_DIAG_OC_LSL_UP[i] > C_T_DIAG_OC_LSL_UP))
        % Heater symptom present, but additional symptoms of OC in VN or VG proven to be absent.
        % This means the heater symptom is caused by a real weak heater.
        or
        T_WAIT_SYM_OBD_2_LSH_UP[i] > C_T_WAIT_SYM_OBD_2_LSH_UP
        % Diagnosis triggered by something other than OBD2 LSH and no other condition applied
        % Should happen only if Plaus/PUC triggered the OC test with a symptom of VLS_L
then
    STATE_DIAG_OC_LSL_UP[i] = "END"
    ERR_SYM_OC_LSL_UP[i]  =  "NO_SYM"
Endif
```

<u>Transition action for all change from State "active" to "end"(Conditions 2 to 6)</u>

```
If      ERR_SYM_OC_LSL_UP[i]  = "NO_SYM"
then
    LV_ERR_OC_LSL_UP[i] = 0
else
```

```
    LV_ERR_OC_LSL_UP[i] = 1
endif
LV_END_DIAG_OC_LSL_UP[i] = 1
```

State “end”

Action:

```
% clean up.
T_WAIT_SYM_OBD_2_LSH_UP[i] = 0
T_DIAG_OC_LSL_UP[i] = 0
T_DIAG_OC_RST_OSC_DI[i] = 0
T_DIAG_OC_RST_LAMB_SP_REQ[i] = 0
T_VLS_UP_MIN_DIAG_OC_LSL_UP[i] = 0
T_VLS_UP_AFS_DIAG_OC_LSL_UP[i] = 0
LAMB_SP_DIAG_OC_LSL_UP[i] = 0
LV_DIAG_OC_IF_OSC_DI_LSL_UP[i] = 0
```

Transition to other states

```
% return to passive state.
If      ERR_SYM_OC_LSL_UP[i] = "NO_SYM"
then
        STATE_DIAG_OC_LSL_UP[i] = "PASSIVE"
endif
```

Section "Moving Lambda setpoint"

This section shall produce a lambda set point that will be forwarded as a request to the lambda setpoint coordination.

```
% if currently not active, then determine the ramp
If      LAMB_SP_DIAG_OC_LSL_UP[i] = 0
then
        LAMB_SP_DELTA_OC_LSL_UP[i] = (C_LAMB_SP_OC_CHK_LSL_UP -
        LAMB_SP[i]) / C_LAMB_SP_STEP_OC_CHK_LSL_UP
        if      LAMB_SP_DELTA_OC_LSL_UP[i] = 0
        then
                if      C_LAMB_SP_OC_CHK_LSL_UP >= LAMB_SP[i]
                then
                        LAMB_SP_DELTA_OC_LSL_UP[i] = +1H
                Else
                        LAMB_SP_DELTA_OC_LSL_UP[i] = -1H
                endif
        endif
```

```
        LAMB_SP_DIAG_OC_LSL_UP[i] = LAMB_SP[i]
endif
% move the set point
increment LAMB_SP_DIAG_OC_LSL_UP[i] with LAMB_SP_DELTA_OC_LSL_UP[i]
if      (LAMB_SP_DELTA_OC_LSL_UP[i] > 0 and
        LAMB_SP_DIAG_OC_LSL_UP[i] > C_LAMB_SP_OC_CHK_LSL_UP)
or      (LAMB_SP_DELTA_OC_LSL_UP[i] < 0 and
        LAMB_SP_DIAG_OC_LSL_UP[i] < C_LAMB_SP_OC_CHK_LSL_UP)
Then
        LAMB_SP_DIAG_OC_LSL_UP[i] = C_LAMB_SP_OC_CHK_LSL_UP
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VLS_UP_DIAG_MIN_OC_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
LSL undervoltage threshold					
C_VLS_UP_MIN_OC_AFS_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
LSL stoichiometric voltage lower threshold					
C_VLS_UP_MAX_OC_AFS_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
LSL stoichiometric voltage upper threshold					
C_T_VLS_UP_MIN_DIAG_OC_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Max time to debounce open circuit in VN or VG symptom					
LC_DIAG_PUC_END_MAN_LSL_UP	1	0...1H	0...1	1	[-]
Boolean switch to bypass activation of LV_DIAG_PUC_END_LSL_UP[i]					
C_T_WAIT_SYM_OBD_2_LSH_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Waiting time to check if OBD2 upstream heater error is present (to distinguish OC on Vip from Vg)					
C_T_DIAG_OC_RST_OSC_DI	1	0...FFFFH	0...6553.5	0.1	[s]
Time to reactivate oscillator for upstream tip temperature measurement					
C_T_DIAG_OC_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Timer to finish OC diagnosis in case of OBD2 upstream heater error symptom active and conditions for plausibility diagnosis fulfilled					
C_LAMB_SP_OC_CHK_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda setpoint target for active test for open circuit in VG / VIP					
C_LAMB_SP_STEP_OC_CHK_LSL_UP	1	1...FFH	1...255	1	[-]
Number of recurrences to achieve the desired lambda set point during open-circuit check					
C_LAMB_SP_MIN_DIAG_OC_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lower end of blind region for lambda setpoint for active test for open circuit in VG / VIP					
C_LAMB_SP_MAX_DIAG_OC_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Upper end of blind region for lambda setpoint for active test for open circuit in VG / VIP					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
OC_LSL_UP[NC_CBK_EX_NR]	Open Circuit on Nernst Cell (Vn)	SYM_0	NO
	Open Circuit on Virtual Ground (Vg)	SYM_1	
	Open Circuit on Pump Cell (Vip)	SYM_2	
	Open Circuit on comp. resistor (Vrc)	SYM_3	

16.29 Application Incidencies of the “O2 sensor (lin) open circuit diagnosis”

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "Open Circuit Diagnosis"					

Input data:

LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_IGK	LV_VB_CDN_OBD_2
LV_ERR_TPS	LV_LS_UP_OBD_1_MAN_DEAC[NC_CBK_EX_NR]	LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]
LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_MAP_TPS_PLAUS			

16.29.1 General

Application conditions:

Initialization: At transition LV_IGK = 0 -> 1 or ECU reset LV_INH_DIAG_OC_LSL_UP[i] = 0.

at every transition LV_IGK = 1->0 LV_INH_DIAG_OC_LSL_UP[i] = 1

Recurrence:

```
If      NC_4D4 = true  (SW Variant = 5  MSD80 4Cyl.)
then    100 ms
else    10 ms
endIf
```

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

The following part shall be carried out once every 10ms:

```
If   LV_ERR_LSL_UP_IF[i] = 1            or
     LV_ERR_TPS = 1                     or
     LV_ERR_FSD[i] = 1                  or
     LV_ERR_FSD_LAM_LIM[i] = 1          or
     LV_ERR_FUP = 1                     or
     LV_ERR_FUP_MFP_PLAUS = 1           or
     LV_ERR_H_PRS_SYS = 1               or
     LV_ERR_VCV = 1                     or
     LV_ERR_FUP_ORNG = 1                or
```

```
    LV_ERR_FUP_ST = 1                    or
    LV_LS_UP_OBD_1_MAN_DEAC[i] = 1       or
    LV_VB_CDN_OBD_2 = 0                  or
    LV_ERR_OFS_LSL_UP[i] = 1             or
    LV_ERR_CTL_LSL_UP[i] = 1             or
    LV_ERR_MAP_TPS_PLAUS = 1

Then
    LV_INH_DIAG_OC_LSL_UP[i] = 1
Else
    LV_INH_DIAG_OC_LSL_UP[i] = 0
Endif
```

Configuration for diagnostic symptoms:

Diagnostic OC_LSL_UP[i]	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
Detection of Open Circuit	Open Circuit on Vn	*SYM_0*					MIL_ON / LAW_L	CARB_LS
	Open Circuit on Vg	*SYM_1*						
	Open Circuit on Vip	*SYM_2*						
	Open Circuit on Vrc	*SYM_3*						

DTCs are defined in the table of failures.

16.30 Upstream oxygen sensor heater OBDI diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Error flag for upstream heater diagnosis OBD I after debouncing					
LV_CDN_DIAG_LSH_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Diagnosis condition for upstream heater diagnosis OBD 1					
LV_END_DIAG_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
End of diagnosis for upstream heater diagnosis OBD I					
LV_LSH_SCG_ACT_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that debounce for upstream heater OBDI SCG diagnosis active					
ERR_DIAG_LSH_UP[NC_CBK_EX_NR]	V	0H 1H 2H 4H	NO_SYM SYM_0 SYM_1 SYM_2	1	[-]
Raw value of error symptom for upstream heater diagnosis OBD I					
ERR_SYM_LSH_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom for upstream heater diagnosis OBD I filterd with CDN_DIAG_LSH_UP					
CDN_DIAG_LSH_UP[NC_CBK_EX_NR]	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of upstream heater diagnosis OBD I					

Input data:

LV_IGK	STATE_LSH_UP[NC_CBK_EX_NR]	**LV_VB_CDN_OBD_1**	TEG_DYN_LS_UP[NC_CBK_EX_NR]
LSHPWM_UP[NC_CBK_EX_NR]	LV_INH_DIAG_LSH_UP[NC_CBK_EX_NR]	NC_CBK_EX_NR	

Import actions:

ACTION_ERRM_FilterMulticondition(IN <*XX* >, IN< CDN_DIAG_ XX>, IN< ERR_DIAG_XX>, IN <C_ABC_INC_XX>, IN <C_ABC_MAX_XX>, OUT<LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result
ACTION_INFR_GetElDiagLshUp(IN <i>, OUT < cdn_diag>, OUT <err_diag>)
This action reads the failure and the conditions information for each symptom from the infrastructure

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

All symptoms of the current error code are handeled by anti-bouncing.
We assume that only one symptom of an error code can be active at the same time.

Description:

The error detection is effected via the ECU hardware.

The purpose of the diagnosis shall be to detect electrical faults within the oxygen sensor heater circuit as defined by OBD I requirements.

Error-symptoms and conditions: are defined to this diagnosis function as following

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 │  │  │  └─> Short circuit to battery (SCP)          (= SYM_0)
 │  │  └────> Short circuit to ground (SCG)           (= SYM_1)
 │  └───────> Open circuit (OC)               (= SYM_2)
 └──────────> -                                       (= SYM_3)
```

The function is intended for use in a hardware configuration where the heater driver shall carry out the electrical fault diagnosis and provide the electrical status via status flags.

- *Short circuit to Vbatt by Over-temperature* ***or*** *Overcurrent* (**SCP**)
- *Short circuit to GND* (SCG)
- *Open load* (**OC**)

The heater drivers named are capable of detecting an **OC** and **SCP** fault when in the ON state and only capable of detecting a SCG fault in the OFF state. As the heater power is controlled by a PWM signal, the driver will be placed alternately in an ON and then OFF state.

The diagnosis shall be activated should the following conditions be met:

- Engine leaving the engine start state as denoted by LV_IGK = 1
- Battery voltage falls between pre-set thresholds to protect against false errors at battery voltage extremes as indicated by **LV_VB_CDN_OBD_1**.
- CDN depending on duty cycle of LSHPWM_UP[i]
- Inhibit flag not set (LV_INH_DIAG_LSH_UP[i])

In exceptionally rare extreme environmental conditions for certain system configurations, it is possible that the heater driver may be driven into over-temperature during initial oxygen sensor heating. In this case, the driver automatically turns off the output stage in self-protection and sets the **SCP** electrical fault bit. In order to mask an incorrect oxygen sensor heater **SCP** fault, the function shall only allow the **SCP** to set a fault during the initial heater phase, if the exhaust gas temperature has exceeded a pre-determined calibrateable threshold.

The oxygen sensor heater fault detection shall be debounced by use of an anti-bounce counter. This shall be handled by the Dynamic Error Management.

Each of the above listed faults shall be described individually below:

Otherwise the diagnosis has to be suppressed (CDN_DIAG_LSH_UP[i] = 0). This is managed within the “Diagnosis condition calculation.

Application conditions:

Initialization: at reset CDN_DIAG_LSH_UP[i] = 0, ERR_DIAG_LSH_UP[i] = 0

All other variables are initialized according filter type

Activation : -

Deactivation: -

Recurrence: 100 ms

NOTE: The heater PWM frequency shall be at least 2 times faster than the diagnosis frequency to ensure that an oversampling effect does not occur!

Formula section:

```
If      LV_IGK = 1
and     LV_VB_CDN_OBD_1 = 1
and     LV_INH_DIAG_LSH_UP[i] = 0
then
```

Usage of the diagnosis information (failure symptoms (raw value) **ERR_DIAG_LSH_UP[i]** and basic diagnosis conditions **CDN_DIAG_LSH_UP[i]**) received from the infrastructure:

ACTION_INFR_GetElDiagLshUp(IN <i>, **OUT**<CDN_DIAG_LSH_UP[i]>, **OUT**<ERR_DIAG_LSH_UP[i]>**)**

Basic diagnosis conditions are set according infrastructure information: CDN_DIAG_LSH_UP[i]. Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_LSH_UP[i].

```
If      Activation conditions are met for the NC_PSD_DLY_LSH_UP recurrence
Then
        If      LSHPWM_UP[i] < C_LSHPWM_MIN_DIAG_LSH_UP
        or      [(STATE_LSH_UP[i] = LSH_POW_RISE
        or      STATE_LSH_UP[i] = LSH_POW_RED)
        and     TEG_DYN_LS_UP[i]< C_TEG_DYN_MIN_DIAG_LSH_UP]
        then    bit 0 of CDN_DIAG_LSH_UP[i] = 0 (SCP can not be detected))
        endif
        If      LSHPWM_UP[i] > C_LSHPWM_MAX_DIAG_LSH_UP
        then    bit 1 of CDN_DIAG_LSH_UP[i] = 0  (SCG can not be detected)
        endif
        if      LSHPWM_UP[i] < C_LSHPWM_MIN_DIAG_LSH_UP
        or      LSHPWM_UP[i]  > C_LSHPWM_MAX_DIAG_LSH_UP
```

```
                then    bit 2 of CDN_DIAG_LSH_UP[i] = 0  (OC can not be detected)
                endif
        Else
                CDN_DIAG_LSH_UP[i] = 0
        Endif
Else
                CDN_DIAG_LSH_UP[i] = 0
endif
```

Error Symptom calculation (raw value from I/O SW)

```
If      bit 1 of ERR_DIAG_LSH_UP[i] = 1
then    LV_LSH_SCG_ACT_LSH_UP[i] = 1
endif
```

```
If      abc-counter = 0
then    LV_LSH_SCG_ACT_LSH_UP[i] = 0
endif
```

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_LSH_UP[i] and ERR_DIAG_LSH_UP[i].

ACTION_ERRM_FilterMulticondition(IN <*XX*>, IN< CDN_DIAG_LSH_UP[i]>, IN< ERR_DIAG_LSH_UP[i]>, IN <C_ABC_INC_LSH_UP>, IN <C_ABC_MAX_LSH_UP>, OUT<LV_ERR_LSH_UP[i]>)

This algorithm determines:

ERR_SYM_LSH_UP[i]	(= raw value ERR_DIAG_LSH_UP[i] filtered with CDN_DIAG_LSH_UP[i], detected error symptom for lambda sensor heater upstream [i] diagnosis)
LV_ERR_ LSH_UP[i]	(Error flag for debounced error of for lambda sensor heater upstream [i])
LV_CDN_DIAG_ LSH_UP[i]	(Diagnosis condition information)
LV_END_DIAG_ LSH_UP[i]	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LSHPWM_MIN_DIAG_LSH_UP	1	0...FFH	0...99.60937	0.390625	[%]
Minimum O2-Sensor heater duty cycle to perform the electrical diagnosis					
C_LSHPWM_MAX_DIAG_LSH_UP	1	0...FFH	0...99.60937	0.390625	[%]
Maximum O2-Sensor heater duty cycle to perform the electrical diagnosis					
C_TEG_DYN_MIN_DIAG_LSH_UP	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Minimum exhaust gas temperature threshold to permit SCP to set fault during LSH power rise state					
C_ABC_INC_LSH_UP	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_LSH_UP	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
Downstream oxygen sensor heater OBDI diagnosis	*Short circuit to battery*	*SYM_0*	*MPL_STD_INI*
	Short circuit to GND	*SYM_1*	
LSH_UP[i]	*Open circuit*	*SYM_2*	
	4th symptom description	*SYM_3*	

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_LSH_UP	-	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (typical 2)					

16.30.1 Application incidences for upstream oxygen sensor heater OBDI diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibit flag for upstream oxygen sensor heater OBDI diagnosis					

Input data:

LV_ST_END	NC_CBK_EX_NR	LV_CDN_VB_OBD1	LV_ERR_SPI_MPS
LV_LS_UP_OBD_1_MAN_DEAC[NC_CBK_EX_NR]			

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

Description:

The diagnosis is inhibited due to low battery voltage, present error or if sensors are not learnt.

Application conditions:

Initialisation:	LV_INH_DIAG_LSH_UP[i] = 1 at reset
Recurrence:	The function shall be carried out once every 100 ms.
Activation:	LV_ST_END = 1
Deactivation:	LV_ST_END = 0 , set LV_INH_DIAG_LSH_UP[i] = 1

Formula section:

```
If      LV_CDN_VB_OBD1 = 0                    or
        LV_ERR_SPI_MPS = 1                    or
        LV_LS_UP_OBD_1_MAN_DEAC[i] = 1              //sensors not learnt
Then    LV_INH_DIAG_LSH_UP[i] = 1
Else    LV_INH_DIAG_LSH_UP[i] = 0
Endif
```

16.31 O2 sensor (bin, down) signal circuit diag

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_EL_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean error flag, electrical fault currently present on downstream oxygen sensor signal					

Input data:

LV_ERR_SCG_LS_DOWN [NC_CBK_EX_NR]	LV_IGK	LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

According the exhaust gas system shall the variables be defined for [i]=1 to NC_CBK_EX_NR.

The VLS_DOWN[i] value used for diagnosis purposes shall not be affected should the sensor be “pumped”. The value prior to pumping shall be used until the sensor returns to normal operation (no pumping).
Once an electrical fault has been detected and debounced, the plausibility diagnosis shall not be carried out until the electrical fault has been diagnosed as being no longer present. The electrical diagnosis shall remain unaffected by the state of the plausibility diagnosis.
Once an electrical fault has been diagnosed and debounced, the application assistances (Functions that are defined to be affected by the state of the oxygen sensor signal) shall remain active for the remainder of the driving cycle. The restarting of the affected functions during the next driving cycle shall be carried out according to the project philosophy.
The measurement of the oxygen sensor voltage shall not be affected by the detection of a signal fault unless explicitly deactivated.

Description:

The oxygen sensor signal electrical diagnosis shall detect the following faults:
A. Short circuit of the oxygen sensor signal line to GND
B. Short circuit of the oxygen sensor signal line to Vbatt
C. Open circuit (line break) in the connection to oxygen sensor element

Each fault may set the boolean flag (LV_ERR_SCG_LS_DOWN[i]; LV_ERR_SCP_LS_DOWN[i]; LV_ERR_OC_LS_DOWN[i]) to signal that the particular fault is currently active. Should one of the electrical fault flags (A-C) be set, then a global error flag (LV_ERR_EL_LS_DOWN[i]) shall be set to indicate that an electrical fault currently exists on the particular oxygen sensor.
Each of the above listed erro shall be described individually in further detail below.

Application conditions:

Initialisation: all variables shall be initialised to 0 with LV_IGK = 0 -> 1 or reset or at clearing the error memory

Recurrence: 100ms

Activation:

If LV_IGK = 1

Deactivation:

If LV_IGK = 0

Formula section:

```
If      LV_ERR_SCG_LS_DOWN[i] = 1
or      LV_ERR_SCP_LS_DOWN[i] = 1
or      LV_ERR_OC_LS_DOWN[i] = 1
then    LV_ERR_EL_LS_DOWN[i] = 1
else    LV_ERR_EL_LS_DOWN[i] = 0
endif
```

16.31.1 Short circuit of oxygen sensor signal to GND.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean error flag, short to ground fault currently present on downstream oxygen sensor signal					
ERR_SYM_SCG_LS_DOWN[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Boolean symptom flag, short to ground fault currently present on downstream oxygen sensor signal					
LV_CDN_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status conditions of oxygen sensor short to ground downstream diagnosis					
LV_END_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Bit to indicate end of short to ground downstream oxygen sensor signal diagnosis					
T_ACT_PLS_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Delay time for final diagnosis activation					
CTR_SYM_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Detection time for SCG					
LV_DIAG_ACT_REQ_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Request for starting the downstream sensor active plausibility check					
CTR_R_UPD_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Value of handshake counter for last valid resistance value used for short circuit to ground diagnosis					

Input data:

VLS_DOWN[NC_CBK_EX_NR]	LV_CAT_PURGE_ACT[NC_CBK_EX_NR]	LV_IGK	LV_INH_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]
R_IT_LS_DOWN[NC_CBK_EX_NR]	CTR_CYCNR_R_IT_LS_DOWN_VLD[NC_CBK_EX_NR]	LV_DIAG_CYC_END_LS_DOWN[NC_CBK_EX_NR]	MAF_KGH

Description:

According the exhaust gas system shall the variables be defined for [i]=1 to NC_CBK_EX_NR.

In order to determine the electrical fault, short circuit to GND, the oxygen sensor signal voltage is compared to a calibrateable threshold (C_VLS_THD_DIAG_SCG_LS_DOWN) in conjunction with certain activation conditions.
Should a hard short to GND (i.e. approx. 0 Ω) exist on the oxygen sensor signal line, the signal voltage will tend to 0 V. This condition may also occur should the air-fuel mixture (AFR) become lean due to a fault in the injection system, due to an air leak, intentional or not, or due to a normal operating conditions. Hence the following additional fault conditions shall be required:
The ignition key shall be determined to be on (LV_IGK = 1) and the appropriate inhibit flag shall not be set (LV_INH_DIAG_SCG_LS_DOWN[i]).

Each time that timer T_ACT_PLS_DIAG_SCG_LS_DOWN[i] exceeded the minimum duration (C_T_DLY_DIAG_SCG_LS_DOWN) for the first time, e.g. first time after engine start or after timer T_ACT_PLS_DIAG_SCG_LS_DOWN[i] has been reset, the CTR_R_UPD_DIAG_SCG_LS_DOWN[I] shall be updated once only with the value CTR_CYCNR_R_IT_LS_DOWN_VLD[i]. This shall ensure that only new internal resistance values be used for diagnosis purposes that have been determined since all the activation conditions have been met.

It shall be determined whether a new internal resistance has been determined by comparing the contents of counter CTR_R_UPD_DIAG_SCG_LS_DOWN[i] with that of CTR_CYCNR_R_IT_LS_DOWN_VLD[i]. Should the counters be unequal, then a new value is available and CTR_R_UPD_DIAG_SCG_LS_DOWN[i] shall be updated with CTR_CYCNR_R_IT_LS_DOWN_VLD[i]. If this conditions and all the above conditions be met, the diagnosis shall be considered to be active.

Should the diagnosis be determined to be active, the oxygen sensor signal voltage shall be compared to the threshold. Should the voltage fall below the threshold and the resistance condition R_IT_LS_DOWN[i] less than threshold be met, a pre - detection counter is incremented. The counter is reset to 0 if the above mentioned conditions are no more fulfilled. If the pre – detection counter counter reached the maximum value then the active check is triggered or if finished the error “Short to GND” is detected. Additionally the triggered active diagnosis can be switched of with LC_DIAG_ACT_CHK_INH = 1.

<u>Application conditions:</u>

Initialisation: all variables shall be initialised to 0 with LV_IGK = 0 -> 1 or reset or at clearing the error memory

Recurrence: T_SAMPLE = 100ms

Activation / Deactivation:

```
If$^{0}$          LV_IGK = 1
and          LV_INH_DIAG_SCG_LS_DOWN[i] = 0
and          LV_CAT_PURGE_ACT[i] = 0
and          MAF_KGH > C_MAF_KGH_MIN_DIAG_SCG_LS_DOWN
then$^{0}$ If$^{1}$    T_ACT_PLS_DIAG_SCG_LS_DOWN[i] $\geq$ C_T_DLY_DIAG_SCG_LS_DOWN
    then$^{1}$ One-off update CTR_R_UPD_DIAG_SCG_LS_DOWN[i]
            If$^{2}$      CTR_CYCNR_R_IT_LS_DOWN_VLD[i] $\neq$
                     CTR_R_UPD_DIAG_SCG_LS_DOWN[i]
            Then$^{2}$    “Diagnosis available”
                     LV_CDN_DIAG_SCG_LS_DOWN[i] = 1
                     CTR_R_UPD_DIAG_SCG_LS_DOWN[i] =
                                    CTR_CYCNR_R_IT_LS_DOWN_VLD[i]
            Else$^{2}$    “Diagnosis NOT available”
                     LV_CDN_DIAG_SCG_LS_DOWN[i] = 0
            Endif$^{2}$
    Else$^{1}$ LV_CDN_DIAG_SCG_LS_DOWN[i] = 0
```

```
                    T_ACT_PLS_DIAG_SCG_LS_DOWN[i] =
                                        T_ACT_PLS_DIAG_SCG_LS_DOWN[i] + T_SAMPLE
            Endif1
    Else0 "Diagnosis NOT available"
                    T_ACT_PLS_DIAG_SCG_LS_DOWN[i] = 0
                    CTR_SYM_DIAG_SCG_LS_DOWN[I] = 0
                    LV_CDN_DIAG_SCG_LS_DOWN[i] = 0
                    LV_DIAG_ACT_REQ_LS_DOWN[i] = 0

    endif0
```

Formula section:

```
IF0          LV_CDN_DIAG_SCG_LS_DOWN[i] = 1
THEN0 If1          VLS_DOWN[i] < C_VLS_THD_DIAG_SCG_LS_DOWN
       and         R_IT_LS_DOWN[i] < C_R_IT_MAX_DIAG_SCG_LS_DOWN
    then1 IF2 CTR_SYM_DIAG_SCG_LS_DOWN[i] > C_CTR_MIN_DIAG_SCG_LS_DOWN
          THEN2 If3     LV_DIAG_CYC_END_LS_DOWN[i] = 1      Active check finished
                or      LC_DIAG_ACT_CHK_INH = 1
                then3
                        LV_DIAG_ACT_REQ_LS_DOWN[i] = 0
                        ERR_SYM_SCG_LS_DOWN[i] = SYM_1
                        % Error management automatically increment ABC counter
                else3
                        LV_DIAG_ACT_REQ_LS_DOWN[i] = 1      Request active check
                endif3
          ELSE2 CTR_SYM_DIAG_SCG_LS_DOWN[i] =
                 CTR_SYM_DIAG_SCG_LS_DOWN[i] +1
                ERR_SYM_SCG_LS_DOWN[i] = NO_SYM
                % Error management automatically decrement ABC counter
          ENDIF2
      else1 LV_DIAG_ACT_REQ_LS_DOWN[i] = 0
          % Error management automatically decrement ABC counter
          ERR_SYM_SCG_LS_DOWN[i] = NO_SYM
          CTR_SYM_DIAG_SCG_LS_DOWN[i] = 0
      endif1
ENDIF0
```

Calibration data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
C_VLS_THD_DIAG_SCG_LS_DOWN	1	0...3FFH	0...4.99511	4.88E-03	[V]
Oxygen sensor voltage threshold under which short to GND may be present					
C_R_IT_MAX_DIAG_SCG_LS_DOWN	1	0...FFFFH	0...65535	1	[ohm]
Maximum resistance threshold oxygen sensor internal resistance during short to ground sensor					
C_CTR_MIN_DIAG_SCG_LS_DOWN	1	0...FFH	0...255	1	[-]
Minimum duration required for continious low VLS_DOWN[i] detection					
C_T_DLY_DIAG_SCG_LS_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum duration required for low VLS_DOWN[i] detection					
C_MAF_KGH_MIN_DIAG_SCG_LS_DOWN	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Minimum MAF_KGH flow required for continious low VLS_DOWN[i] detection					
LC_DIAG_ACT_CHK_INH	1	0…1H	0..1	1	[-]
Inhibition bit for the triggered active diagnosis					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
SCG_LS_DOWN[NC_CBK_EX_NR]	short circuit to ground bank i	SYM_1	MEM

16.31.2 Short circuit of oxygen sensor signal to VBATT

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean error flag, short to plus fault currently present on downstream oxygen sensor signal					
ERR_SYM_SCP_LS_DOWN[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Boolean symptom flag, short to plus fault currently present on downstream oxygen sensor signal					
LV_CDN_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status conditions of oxygen sensor short to plus downstream diagnosis					
LV_END_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Bit to indicate end of short to plus downstream oxygen sensor signal diagnosis					

Input data:

VLS_DOWN[NC_CBK_EX_NR]	LV_INH_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_IGK	

Description:

According the exhaust gas system shall the variables be defined for [i]=1 to NC_CBK_EX_NR.

Should a hard short to Vbatt (i.e. approx. 0 Ω) exist on the oxygen sensor signal line, the signal voltage will tend to Vbatt and the analogue digital converter value will tend to 5 V. This effect does not occur under normal conditions as the sensor delivers typically 1 V for rich AFR at the sensor location and may be used for short to Vbatt detection.

In order to determine the electrical fault, short circuit to Vbatt, the oxygen sensor signal voltage (VLS_DOWN[i]) is compared to a calibrateable threshold (C_VLS_THD_DIAG_SCP_LS_DOWN) in conjunction with the herein mentioned activation conditions.

The ignition key shall be determined to be on (LV_IGK = 1) and the appropriate inhibit flag shall not be set (LV_INH_DIAG_SCP_LS_DOWN[i]).

Should the oxygen signal voltage exceed the said threshold, a signal short to Vbatt shall be determined to be present and the debounce counter shall be incremented otherwise the debounce counter shall be decremented.

The error flag LV_ERR_SCP_LS_DOWN[i] shall be set once the anti-bounce counter has reached its maximum value and reset should the anti-bounce counter reach 0.

Application conditions:

Initialisation: all variables shall be initialised to 0 with LV_IGK = 0 -> 1 or reset or at clearing the error memory

Recurrence: 100ms

Activation / Deactivation:

```
If      LV_IGK = 1
and     LV_INH_DIAG_SCP_LS_DOWN[i] = 0
then    "Diagnosis available"
        LV_CDN_DIAG_SCP_LS_DOWN[i] = 1
else    "Diagnosis NOT available"
        LV_CDN_DIAG_SCP_LS_DOWN[i] = 0
endif
```

Formula section:

```
If      LV_CDN_DIAG_SCP_LS_DOWN[i] = 1
then    If      VLS_DOWN[i] > C_VLS_THD_DIAG_SCP_LS_DOWN
        then    "Short to Vbatt fault present"
                % Error management automatically increment ABC counter
                ERR_SYM_SCP_LS_DOWN[i] = SYM_0            %SHORT TO PLUS

        else    ERR_SYM_SCP_LS_DOWN[i] = "NO_SYM"
                % Error management automatically decrement ABC counter
```

endif

endif

Calibration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
C_VLS_THD_DIAG_SCP_LS_DOWN	1	0...3FFH	0...4.99511	4.88E-03	[V]
Oxygen sensor voltage threshold over which short to Vbatt present					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
SCP_LS_DOWN[NC_CBK_EX_NR]	short circuit to plus bank i	SYM_0	MEM

16.31.3 Open circuit (line break) in connection to oxygen sensor element.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean error flag, open circuit fault currently present on downstream oxygen sensor signal					
ERR_SYM_OC_LS_DOWN[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Variable indicating an open circuit fault is currently present on downstream oxygen sensor signal					
LV_T_CDN_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Status conditions of VLS range for the downstream oxygen sensor open circuit diagnosis					
LV_R_IT_VLD_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Valid conditions of resistance of downstream oxygen sensor open circuit diagnosis					
LV_CDN_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status conditions of downstream oxygen sensor open circuit diagnosis					
LV_END_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Bit to indicate end of downstream oxygen sensor open circuit diagnosis					
T_ACT_PLS_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Delay time for final diagnosis activation					
T_CDN_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Detection time for status OC before diagnosis					
CTR_R_UPD_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Value of handshake counter for last valid resistance value used for diagnosis					

Input data:

VLS_DOWN[NC_CBK_EX_NR]	TEG_DYN_LS_DOWN[NC_CBK_EX_NR]	LV_IGK	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]
STATE_LSH_DOWN[NC_CBK_EX_NR]	LV_INH_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	CTR_CYCNR_R_IT_LS_DOWN_VLD[NC_CBK_EX_NR]	R_IT_LS_DOWN[NC_CBK_EX_NR]
LV_LS_DOWN_READY[NC_CBK_EX_NR]	NC_CBK_EX_NR		

Description:

According the exhaust gas system shall the variables be defined for [i]=1 to NC_CBK_EX_NR.

Should a complete open circuit occur in the oxygen sensor signal or return line, the measured voltage at the input to the microcontroller will tend to the voltage determined by the operative readiness potential divider (approx. 450 mV) and the measured internal resistance will be abnormally high.
In order to determine an open circuit electrical fault, the oxygen sensor internal resistance (R_IT_LS_DOWN[i]) is compared to a calibrateable threshold (C_R_IT_MIN_DIAG_OC_LS_DOWN) in conjunction with certain activation conditions shown below:

The ignition key shall be determined to be on (LV_IGK = 1) and the appropriate inhibit flag shall not be set (LV_INH_DIAG_OC_LS_DOWN[i]).

The sensor shall be determined to be in a state of operative readiness (LV_LS_DOWN_READY[i] = 1, possibly thorough forced activation) and the temperature operating conditions of the sensor shall be determined to be sufficient to permit normal operation, i.e. No fault shall be determined to be present in the appropriate downstream oxygen sensor heater (LV_ERR_LSH_DOWN[i] = 0) or if so that the operating temperature be determined to be sufficient for sensor operability (TEG_DYN_LS_DOWN[i] > C_TEG_DYN_MIN_DIAG_OC_LS_DOWN), and the appropriate heater shall be determined to being controlled normally (STATE_LSH_DOWN[i] = LSH_POW_CTL).

Should the above initial conditions be met, a timer (T_ACT_PLS_DIAG_OC_LS_DOWN[i]) shall be incremented. Should the conditions no longer be met, both timers shall be reset and flag LV_T_CDN_DIAG_OC_LS_DOWN[i] reset.

The oxygen sensor signal shall be determined to lie in a calibrateable voltage band (C_VLS_AFL_THD_DIAG_OC_LS_DOWN < VLS_DOWN[i] < C_VLS_AFR_THD_DIAG_OC_LS_DOWN).

Should the above listed pre-activation conditions be determined to have been met, a timer T_CDN_DIAG_OC_LS_DOWN[i] shall be started. The timer shall be required to equal or exceed a minimum calibrateable duration (C_T_CDN_DIAG_OC_LS_DOWN). When no longer met, T_CDN_DIAG_OC_LS_DOWN[i] and LV_T_CDN_DIAG_OC_LS_DOWN[i] shall be reset.

Each time that the T_CDN_DIAG_OC_LS_DOWN[i] has exceeded the minimum duration for the first time, e.g. first time after engine start or after the T_CDN_DIAG_OC_LS_DOWN[i] has been reset, the CTR_R_UPD_DIAG_OC_LS_DOWN[i] shall be updated once only with the value CTR_CYCNR_R_IT_LS_DOWN_VLD[i]. This shall ensure that only new internal resistance values be used for diagnosis purposes that have been determined since all the activation conditions have been met.

It shall be determined whether a new internal resistance has been determined by comparing the contents of counter CTR_R_UPD_DIAG_OC_LS_DOWN[i] with that of CTR_CYCNR_R_IT_LS_DOWN_VLD[i]. Should the counters be unequal, then a new value is available and CTR_R_UPD_DIAG_OC_LS_DOWN[i] shall be updated with CTR_CYCNR_R_IT_LS_DOWN_VLD[i] and a flag LV_R_IT_VLD_DIAG_OC_LS_DOWN[I] be set, thus indicating that the voltage lies within the set band and that a new resistance value is available.

If either LV_T_CDN_DIAG_OC_LS_DOWN[i] is set or T_ACT_PLS_DIAG_OC_LS_DOWN[i] equals or exceeds threshold C_T_DLY_DIAG_OC_LS_DOWN, the diagnosis shall be considered to be available.

Should the diagnosis be determined to be available and the condition LV_R_IT_VLD_DIAG_OC_LS_DOWN[I] has been met, the oxygen sensor internal resistance shall be compared to the said threshold. Should the resistance exceed the threshold, a signal open circuit shall be determined to be present and the debounce counter shall be incremented otherwise the debounce counter shall be decremented.

The error flag LV_ERR_OC_LS_DOWN[i] shall be set once the anti-bounce counter has reached its maximum and reset should the anti-bounce counter reach 0.

Application conditions:

Initialisation: all variables shall be initialised to 0 with LV_IGK = 0 -> 1 or reset or at clearing the error memory

Recurrence: T_SAMPLE = *100ms*

Activation / Deactivation:

```
If      LV_IGK = 1
and     LV_INH_DIAG_OC_LS_DOWN[i] = 0
and     LV_LS_DOWN_READY[i] = 1
and     (LV_ERR_LSH_DOWN[i] = 0
or      TEG_DYN_LS_DOWN[i] > C_TEG_DYN_MIN_DIAG_OC_LS_DOWN)
and     STATE_LSH_DOWN[i] = LSH_POW_CTL
then
        T_ACT_PLS_DIAG_OC_LS_DOWN[i] = T_ACT_PLS_DIAG_OC_LS_DOWN[i] +
                                                        T_SAMPLE

    One-off update CTR_R_UPD_DIAG_OC_LS_DOWN[i]
    If CTR_CYCNR_R_IT_LS_DOWN_VLD[i] ≠ CTR_R_UPD_DIAG_OC_LS_DOWN[i]
    then  CTR_R_UPD_DIAG_OC_LS_DOWN[i]
                             = CTR_CYCNR_R_IT_LS_DOWN_VLD[i]
          LV_R_IT_VLD_DIAG_OC_LS_DOWN[I] = 1
    else  LV_R_IT_VLD_DIAG_OC_LS_DOWN[I] = 0
    endif
    If    C_VLS_AFL_THD_DIAG_OC_LS_DOWN < VLS_DOWN[i]
                < C_VLS_AFR_THD_DIAG_OC_LS_DOWN
    then
          If    T_CDN_DIAG_OC_LS_DOWN[i] ≥ C_T_CDN_DIAG_OC_LS_DOWN
          then  LV_T_CDN_DIAG_OC_LS_DOWN[i] = 1
          else  LV_T_CDN_DIAG_OC_LS_DOWN[i] = 0
                T_CDN_DIAG_OC_LS_DOWN[i] =
                        T_CDN_DIAG_OC_LS_DOWN[i] + T_SAMPLE
          endif
    else  T_CDN_DIAG_OC_LS_DOWN[i] = 0
          LV_T_CDN_DIAG_OC_LS_DOWN[i] = 0
    endif
else    T_CDN_DIAG_OC_LS_DOWN[i] = 0
        T_ACT_PLS_DIAG_OC_LS_DOWN[i] = 0
        LV_R_IT_VLD_DIAG_OC_LS_DOWN[i] = 0
        LV_T_CDN_DIAG_OC_LS_DOWN[i] = 0
endif
If      (T_ACT_PLS_DIAG_OC_LS_DOWN[i] ≥ C_T_DLY_DIAG_OC_LS_DOWN
```

```
or      LV_T_CDN_DIAG_OC_LS_DOWN[i] = 1)
and     LV_R_IT_VLD_DIAG_OC_LS_DOWN[i] = 1
then    "Diagnosis available"
        LV_CDN_DIAG_OC_LS_DOWN[i] = 1
else    "Diagnosis NOT available"
        LV_CDN_DIAG_OC_LS_DOWN[i] = 0
endif
```

NOTE: The variable CTR_CYCNR_R_VLS_DOWN_DIAG_i shall be updated with the current value of the input variable CTR_CYCNR_R_IT_LS_DOWN_VLD[i] on each occasion that the TIMER_3 exceeds the minimum for the first time. For further function calls where the TIMER_3 condition is met, the one-off update shall not be carried out. This shall continue until the TIMER_3 condition is no longer met.

Formula section:

```
If      LV_CDN_DIAG_OC_LS_DOWN[i] = 1
then    If      R_IT_LS_DOWN[i] > C_R_IT_MIN_DIAG_OC_LS_DOWN
        and     LV_T_CDN_DIAG_OC_LS_DOWN[i] = 1
        then    "Open circuit fault present"
                % Error management automatically iccrement ABC counter
                ERR_SYM_OC_LS_DOWN[i] = sym_2  %"OPEN LINE"

        else    ERR_SYM_OC_LS_DOWN[i] = "NO_SYM"
                % Error management automatically decrement ABC counter
        endif
endif
```

Calibration data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
C_R_IT_MIN_DIAG_OC_LS_DOWN	1	0...FFFFH	0...65535	1	[ohm]
Minimum resistance threshold oxygen sensor internal resistance during open circuit sensor					
C_T_CDN_DIAG_OC_LS_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum required duration where voltage lies outside of the limits for operable sensor to detect fault					
C_T_DLY_DIAG_OC_LS_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum required duration where voltage lies outside of the limits for operable sensor to detect fault					
C_TEG_DYN_MIN_DIAG_OC_LS_DOWN	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Minimum requested exhaust gas temperature for diagnosis					
C_VLS_AFL_THD_DIAG_OC_LS_DOWN	1	0...3FFH	0...4.99511	4.88E-03	[V]
Oxygen sensor voltage-lean mixture threshold for diagnosis					
C_VLS_AFR_THD_DIAG_OC_LS_DOWN	1	0...3FFH	0...4.99511	4.88E-03	[V]
Oxygen sensor voltage-rich mixture threshold for diagnosis					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
OC_LS_DOWN[NC_CBK_EX_NR]	Open circuit bank i	SYM_2	MEM

16.32 O2 sensor (bin, down) signal circuit diag (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_ PUC_LS_DOWN[NC_CBK_EX_NR]	V/O	0...7H	0...7	1	[-]
Interface of PUC_LS_DOWN					
LV_INH_DIAG_RBM_PUC_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0..1	1	[-]
Deactivation of Rate based monitoring of down stream sensor (PUC_LS_DOWN) by mal function.					
LV_INH_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating project specific application conditions for downstream open line diagnosis not met					
LV_INH_DIAG_PUC_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating project specific application conditions for downstream PUC diagnosis not met					
LV_INH_DIAG_FL_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating project specific application conditions for downstream FL diagnosis not met					
LV_INH_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating project specific application conditions for downstream SCG diagnosis not met					
LV_INH_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating project specific application conditions for downstream short to plus diagnosis not met					

Input data:

LV_VB_CDN_OBD_1	LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]
LV_ERR_AMP	LV_ERR_AMP_PLAUS	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_CAM_CUS	LV_ERR_MAP_DIP_SENS	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_EL_CPS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_CRK	LV_ERR_MAP_DIP_SHIFT	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_DIAGCPS	LV_ERR_IVVT	
LV_ERR_FSD[NC_CBK_EX_NR]	LV_VAR_LSH_DOWN	LV_ERR_LS_UP[NC_CBK_EX_NR]
LV_ERR_MAF	NC_CBK_EX_NR	LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]
		LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]
LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]	LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_LSH_LSL_UP[NC_CBK_EX_NR]
STATE_ERR_IV	LV_ERR_CHG_LS_UP	LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_LAM_LSCL[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]
LV_ERR_TCO	LV_FAC_LAM_LIM_MIN[NC_CBK_EX_NR]	LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_TPS	LV_FAC_LAM_LIM_MAX[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]
LV_IGK	LV_DIAG_ACT_REQ_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]
LV_MIS_STATE_A	LV_ERR_FUP	LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]
LV_MIS_STATE_B	LV_ERR_FUP_MFP_PLAUS	LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_VCV	LV_ERR_H_PRS_SYS	LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_CHG_LS_DOWN	LV_ERR_FUP_ORNG	LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_TCO_STUCK	LV_ERR_TCO_PLAUS	LV_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_IN	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]
R_IT_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS	LV_END_DIAG_PUC_LS_DOWN

FUNCTION DESCRIPTION:

General information:

According the exhaust gas system the variables shall be defined from [i]=1 to NC_CBK_EX_NR.

Application conditions:

Initialisation: all variables are initialised to 1 at reset and every LV_IGK = 0→1

Recurrence: 100 ms

Activation: LV_IGK = 1 **and** LV_VAR_LSH_DOWN = 1

Deactivation: LV_IGK = 0 **or** LV_VAR_LSH_DOWN = 0

Before deactivation all variables shall be reset to 1.

Formula section:

Calculation of LV_INH_DIAG_OC_LS_DOWN[i]

```
If   LV_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR] = 1         or
     LV_ERR_LSH_DOWN[NC_CBK_EX_NR] = 1                   or
     LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR] = 1                or
     LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR] = 1                or
     LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR] = 1               or
     LV_VB_CDN_OBD_1 = 0

  then     LV_INH_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]  =  1
  else     LV_INH_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]  =  0
endif
```

Calculation of LV_INH_DIAG_PUC_LS_DOWN[i]

```
If    ~~LV_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR] = 1~~        ~~or~~
      STATE_ERR_IV <> 0                                      or
      LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR] = 1                   or
      LV_ERR_FUP = 1                                         or
      LV_ERR_FUP_MFP_PLAUS = 1                               or
      LV_ERR_H_PRS_SYS = 1                                   or
      LV_ERR_VCV[NC_CBK_HPP_NR] = 1                          or
      LV_ERR_CHG_LS_DOWN = 1                                 or
      LV_ERR_FUP_ORNG = 1                                    or
      LV_ERR_FUP_ST = 1                                      or
      LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR] = 1               or
      LV_ERR_LS_UP [NC_CBK_EX_NR] = 1                        or
      LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR] = 1                    or
      LV_ERR_LSH_DOWN[NC_CBK_EX_NR] = 1                      or
      LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR] = 1                   or
      LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR] = 1                   or
      LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR] = 1                    or
      LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR] = 1                   or
      LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR] = 1                   or
      LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR] = 1                  or
      LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR] = 1                   or
      LV_ERR_EL_CPS  = 1                                     or
      LV_ERR_DIAGCPS = 1                                     or
      LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR] = 1                  or
      LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR] = 1                   or
      LV_ERR_CHG_LS_UP = 1                                   or
      LV_ERR_FSD [NC_CBK_EX_NR] = 1                          or
      LV_ERR_MAF = 1                                         or
      LV_ERR_MAP_DIP_SENS = 1                                or
      LV_ERR_MAP_DIP_PLAUS = 1                               or
      LV_ERR_MAP_DIP_SHIFT = 1                               or
      ~~LV_VB_CDN_OBD_1 = 0~~                                ~~or~~
      LV_ERR_OFS_LSL_UP[i] = 1                               or
      LV_ERR_CTL_LSL_UP[i] = 1                               or
      LV_ERR_MAP_TPS_PLAUS = 1                               or
      ~~R_IT_LS_DOWN[i] > C_R_IT_MAX_DIAG_MPL_LS_DOWN~~

then      LV_INH_DIAG_RBM_PUC_LS_DOWN[NC_CBK_EX_NR]  =  1
  else    LV_INH_DIAG_RBM_PUC_LS_DOWN[NC_CBK_EX_NR]  =  0
endif

If    LV_INH_DIAG_RBM_PUC_LS_DOWN [NC_CBK_EX_NR] = 1    or
      LV_LS_DOWN_OBD_1_MAN_DEAC [NC_CBK_EX_NR] = 1      or
      LV_VB_CDN_OBD_1 = 0                               or
      R_IT_LS_DOWN[i] > C_R_IT_MAX_DIAG_MPL_LS_DOWN

Then  LV_INH_DIAG_PUC_LS_DOWN = 1
else  LV_INH_DIAG_PUC_LS_DOWN = 0
```

Calculation of LV_INH_DIAG_FL_LS_DOWN[i]

```
IF      LV_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR] = 1   OR
     -  LV_ERR_AMP = 1                                 OR
     -  LV_ERR_AMP_PLAUS = 1                           OR
     -   LV_ERR_CRK = 1                                OR
     -   LV_ERR_CAM_CUS = 1                            OR
     -   LV_ERR_TCO = 1                                OR
     -   LV_ERR_FUP = 1                                OR
     -   LV_ERR_FUP_MFP_PLAUS = 1                      OR
     -   LV_ERR_H_PRS_SYS = 1                          OR
     -   LV_ERR_VCV = 1                                OR
     -   LV_ERR_CHG_LS_DOWN = 1                        OR
     -   LV_ERR_FUP_ORNG = 1                           OR
     -   LV_ERR_FUP_ST = 1                             OR
     -   LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR] = 1      OR
     -   LV_ERR_TPS = 1                                OR
     -   LV_ERR_MAF = 1                                OR
     -   LV_ERR_MAP_DIP_SENS = 1                       OR
     -   LV_ERR_MAP_DIP_PLAUS = 1                      OR
     -   LV_ERR_MAP_DIP_SHIFT = 1                      OR
     -   LV_ERR_EL_CPS = 1                             OR
     -   LV_ERR_DIAGCPS = 1                            OR
     -   LV_ERR_SA_SAV = 1                             OR
     -   LV_ERR_SA_SAV_LSL = 1                         OR
     -   LV_ERR_SA_SAP = 1                             OR
     -   LV_ERR_SA_SYS = 1                             OR
     -   LV_ERR_SAV = 1                                OR
     -   LV_ERR_SAP = 1                                OR
     -   LV_MIS_STATE_A = 1                            OR
     -   LV_MIS_STATE_B = 1                            OR
     -   LV_ERR_FSD[NC_CBK_EX_NR] = 1                  OR
     -   LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR] = 1          OR
     -   LV_ERR_CHG_LS_UP = 1                          OR
     -   LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR] = 1         OR
     -   LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR] = 1          OR
     -   LV_ERR_LOAD_TPS_PLAUS = 1                     OR
     -   LV_ERR_LS_UP[NC_CBK_EX_NR] = 1                OR
     -       LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR] = 1      OR
     -       LV_ERR_LSH_DOWN[NC_CBK_EX_NR] = 1         OR
     -       LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR] = 1 OR
     -       LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR] = 1      OR
     -       LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR] = 1       OR
     -       LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR] = 1      OR
     -       LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR] = 1OR
     -       LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR] = 1     OR
     -       LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR] = 1      OR
     -   LV_ERR_IVVT = 1                               OR
     -   LV_VB_CDN_OBD_1 = 0                           OR
     -   LV_ERR_SLV_IVVT_IN = 1                        OR
     -   LV_ERR_OFS_LSL_UP[i] = 1                      OR
     -   LV_ERR_CTL_LSL_UP[i] = 1                      OR
```

```
    -   LV_ERR_MAP_TPS_PLAUS = 1

 then      LV_INH_DIAG_FL_LS_DOWN[NC_CBK_EX_NR]  =  1
 else      LV_INH_DIAG_FL_LS_DOWN[NC_CBK_EX_NR]  =  0
endif
```

Calculation of LV_INH_DIAG_SCG_LS_DOWN[i]

```
If   LV_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR] = 1             OR
     LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR] = 1                     OR
     LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR] = 1                    OR
     LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR] = 1                   OR
     LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR] = 1                    OR
     LV_ERR_CHG_LS_DOWN = 1                                  OR
     LV_ERR_LSH_DOWN[NC_CBK_EX_NR] = 1                       OR
     LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR] = 1                    OR
     LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR] = 1                     OR
     LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR] = 1                    OR
     LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR] = 1                    OR
     LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR] = 1                   OR
     LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR] = 1                    OR
     LV_VB_CDN_OBD_1 = 0                                     OR
     (LV_DIAG_ACT_REQ_LS_DOWN[i] = 0
           and          (LV_LAM_LSCL[i] = 0
                         or    LV_FAC_LAM_LIM_MIN[i] =  1
                         or    LV_ FAC_LAM_LIM_MAX[i] = 1))

then       LV_INH_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]  =  1
else       LV_INH_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]  =  0
endif
```

Calculation of LV_INH_DIAG_SCP_LS_DOWN[i]

```
If   LV_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR] = 1             or
     LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR] = 1               or
     LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR] = 1                     or
     LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR] = 1                   or
     LV_ERR_LSH_DOWN[NC_CBK_EX_NR] = 1                       or
     LV_VB_CDN_OBD_1 = 0

 then      LV_INH_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR]  =  1
 else      LV_INH_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR]  =  0
endif
```

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
Downstream Oxygen Sensor Signal	*Short to ground (bank 1/2)*	*0*	*see table of failires*			100ms	
SCG_LS_DOWN[i]							

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
Downstream Oxygen Sensor Signal	*Short to plus (bank 1/2)*	*0*	*see table of failires*			100ms	
SCP_LS_DOWN[i]							

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
Downstream Oxygen Sensor Signal	*Open line (bank 1/2)*	*0*	*see table of failires*			100ms	
OC_LS_DOWN[i]							

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
Downstream Oxygen Sensor Signal Plausibility	*Signal plausibility fault (bank 1/2)*	*3*	*see table of failires*			100ms	
PUC_LS_DOWN[i]							

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
Downstream Oxygen Sensor Signal Plausibility	*Signal plausibility fault (bank 1/2)*	*3*	*see table of failires*			100ms	
FL_LS_DOWN[i]							

16.32.1 Interface for Rate – Based - Monitoring

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the PUC_LS_DOWN[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_PUC_LS_DOWN[i] data.

Within STATE_RBM_PUC_LS_DOWN[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

Application conditions:

Initialisation: *at ECU reset:*
bit 0, bit 1 and bit 2 of STATE_RBM_ PUC_LS_DOWN[i] = 0
LV_DC 0 ->1 transition:
bit 0 and bit 1 of STATE_RBM_ PUC_LS_DOWN [i] = 0
on failure memory reset:
bit 1 of STATE_RBM_STATE_RBM_ PUC_LS_DOWN [i] = 0

Recurrence: 1 s
Activation: LV_DC = 1

Formula section:

At LV_DC 0 ->1 transition

bit 2 of STATE_RBM_ PUC_LS_DOWN [i] = 1

The pending status of the following failures has to be checked only once :

LV_ERR_IV[NC_CYL_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_FUP
LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV[NC_CBK_HPP_NR]
LV_ERR_CHG_LS_DOWN	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]
LV_ERR_EL_LSL_UP[i]	LV_ERR_OC_LSL_UP[i]	LV_ERR_LSL_UP_IF[i]
LV_ERR_LSH_UP[i]	LV_ERR_OBD_VLD_LSH_UP[i]	LV_ERR_LSH_LSL_UP[i]
LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]	LV_ERR_PUC_VLD_LS_UP[i]	LV_ERR_SHIFT_AFL_LSL_UP[i]
LV_ERR_SHIFT_AFR_LSL_UP[i]	LV_ERR_DYN_VLD_LS_UP[i]	LV_ERR_CHG_LS_UP
LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_EL_CPS	LV_ERR_DIAGCPS	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]
LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS	LV_ERR_FSD[NC_CBK_EX_NR]
LV_ERR_MAF	LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS
LV_ERR_FUP_ORNG	LV_ERR_MAP_DIP_SHIFT	LV_ERR_OFS_LSL_UP[i]
LV_ERR_CTL_LSL_UP[i]		

```
If(1)       { CPU optimization at LV_DC 0 ->1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While  bit 1 of STATE_RBM_ PUC_LS_DOWN [i] = 0 do
                   with each XX failure of the above list  (depending on configuration, see
                   below):
                   ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                   SYNCHRONIZATION<CALL>)

                   If(2)    XX has a pending status
                   Then(2)         bit 1 of STATE_RBM_ PUC_LS_DOWN[i] = 1
                   Endif(2)
            Endwhile
Else(1)     { the dynamic failure memory is empty }      No action
Endif(1)
```

Every 1 s :

```
If    bit 0 of STATE_RBM_PUC_LS_DOWN [i] = 0
Then
            If       LV_END_DIAG_PUC_LS_DOWN [i] = 1
            Then     bit 0 of STATE_RBM_PUC_LS_DOWN[i] = 1
            Endif
Endif


If    bit 1 of STATE_RBM_ PUC_LS_DOWN [i] = 0
Then
            If       LV_INH_DIAG_RBM_ PUC_LS_DOWN [i] = 1
            Then     bit 1 of STATE_RBM_ PUC_LS_DOWN [i] = 1
            Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_MAX_SCG_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter maximum for short to ground electrical check					
C_ABC_INC_SCG_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter increment for short to ground electrical check					
C_ABC_MAX_SCP_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter maximum for short to plus electrical check					
C_ABC_INC_SCP_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter increment for short to plus electrical check					
C_ABC_MAX_OC_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter maximum for open line electrical check					
C_ABC_INC_OC_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter increment for open line electrical check					
C_ABC_MAX_PUC_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter maximum for PUC plausibility					
C_ABC_INC_PUC_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter increment for PUC plausibility					
C_ABC_MAX_FL_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter maximum for FL plausibility					
C_ABC_INC_FL_LS_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter increment for FL plausibility					
C_R_IT_MAX_DIAG_MPL_LS_DOWN	1	0...FFFFH	0...65535	1	[Ohm]
Maximum allowed internal resistance of the downstream sensor for downstream dynamic diagnosis					

16.33 Signal plausibility monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_CDN_PLAUS_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag, plausibility check on downstream oxygen sensor signal possible					
LV_ERR_PLAUS_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag, error plausibility check on downstream oxygen sensor signal					

Input data:

LV_ERR_EL_LS_DOWN[NC_CBK_EX_NR]	LV_LS_DOWN_READY[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]
NC_CBK_EX_NR	STATE_LSH_DOWN[NC_CBK_EX_NR]	LV_IGK	LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

Due to possible oxygen sensor defects (e.g. reference air poisoning) or faults in the injection system (e.g. leaking fuel injector), the oxygen sensor may not provide the expected lean or rich AFR signal level during overrun fuel cut-off (PUC) or full load (FL) condition. Hence, the oxygen sensor signal shall be checked for plausibility during this engine operating states.

Description:

If the following conditions are met, then the general conditions for plausibility check are determined to be present:

A. Ignition key shall be determined to be on (LV_IGK = 1)
B. No electrical oxygen sensor fault shall be determined to be present on the sensor being diagnosed (LV_ERR_EL_LS_DOWN[i] = 0).
C. Sensor shall be in a state of operative readiness (LV_LS_DOWN_READY[i] =1).
D. The sensor shall be determined to be at operating temperature (STATE_LSH_DOWN[i] = LSH_POW_CTL & LV_ERR_LSH_DOWN[i] = 0).

The corresponding flag LV_DIAG_CDN_PLAUS_LS_DOWN[i] shall then be set.

For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation:

At the transition LV_IGK = 0 -> 1, reset or clearing the error memory, reset of all variables and bits to 0

Recurrence: **T_sample 10 ms**

Activation / Deactivation:

```
If      LV_IGK = 1
and     LV_ERR_EL_LS_DOWN[i] = 0
and     LV_LS_DOWN_READY[i] = 1
and     LV_ERR_LSH_DOWN[i] = 0
and     STATE_LSH_DOWN[i] = LSH_POW_CTL
then    "Diagnosis available"
        LV_DIAG_CDN_PLAUS_LS_DOWN[i] = 1
else    "Diagnosis NOT available"
        LV_DIAG_CDN_PLAUS_LS_DOWN[i] = 0
endif
```

Formula section:

```
If      LV_ERR_PUC_LS_DOWN[i] = 1
or      LV_ERR_FL_LS_DOWN[i] = 1
then    LV_ERR_PLAUS_LS_DOWN[i] = 1
else    LV_ERR_PLAUS_LS_DOWN[i] = 0
endif
```

16.33.1 Overrun fuel cut-off (PUC) oxygen sensor signal plausibility

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Boolean error flag, PUC plausibility fault currently present on downstream oxygen sensor signal					
ERR_SYM_PUC_LS_DOWN[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Boolean error flag, fault currently present on downstream oxygen sensor signal					
LV_CDN_DIAG_PUC_LS_DOWN[NC_CBK_EX_NR]	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Status conditions of oxygen sensor downstream diagnosis					
LV_END_DIAG_PUC_LS_DOWN[NC_CBK_EX_NR]	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Bit to indicate end of complete downstream oxygen sensor signal diagnosis					
MAF_INT_PUC_NOT_LS_DOWN_DIAG[NC_CBK_EX_NR]	V	0...FFFFH	0...2912.66666	0.0444444	[g]
MAF integral out of PUC in case of certain load threshold exceeded					
MAF_INT_PUC_ACT_OLD	-	0...FFFFH	0...2912.66666	0.0444444	[g]
MAF integral during PUC from previous recurrence					
LV_DIAG_PUC_CHG_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Reminder flag for fuel cut off was active					
VLS_DOWN_PUC_OK[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Last VLS_DOWN value satisfying PUC plausibility					
VLS_DOWN_PUC_ERR[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Last VLS_DOWN value not plausible for PUC conditions					
VLS_DOWN_PUC_SAVE[NC_CBK_EX_NR]	V/O/S	0...3FFH	0...4.99511	4.8828e-3	[V]
Mode 6 value describing last VLS_DOWN PUC plausibility result					
VLS_DOWN_PUC_PLAUS_VLD[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
VLS_DOWN[i] value before entering fuel-cut phase					
LV_VLS_DOWN_PUC_AFR_VLD[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical variable indicating that rich voltage threshold has been exceeded before entering PUC phase					

Input data:

VLS_DOWN[NC_CBK_EX_NR]	LV_PUC	MAF_INT_PUC_ACT	LV_DIAG_CDN_PLAUS_LS_DOWN[NC_CBK_EX_NR]
LV_INH_DIAG_PUC_LS_DOWN[NC_CBK_EX_NR]	LV_IGK	NC_CBK_EX_NR	MAF_KGH

FUNCTION DESCRIPTION:

General information:

Description:

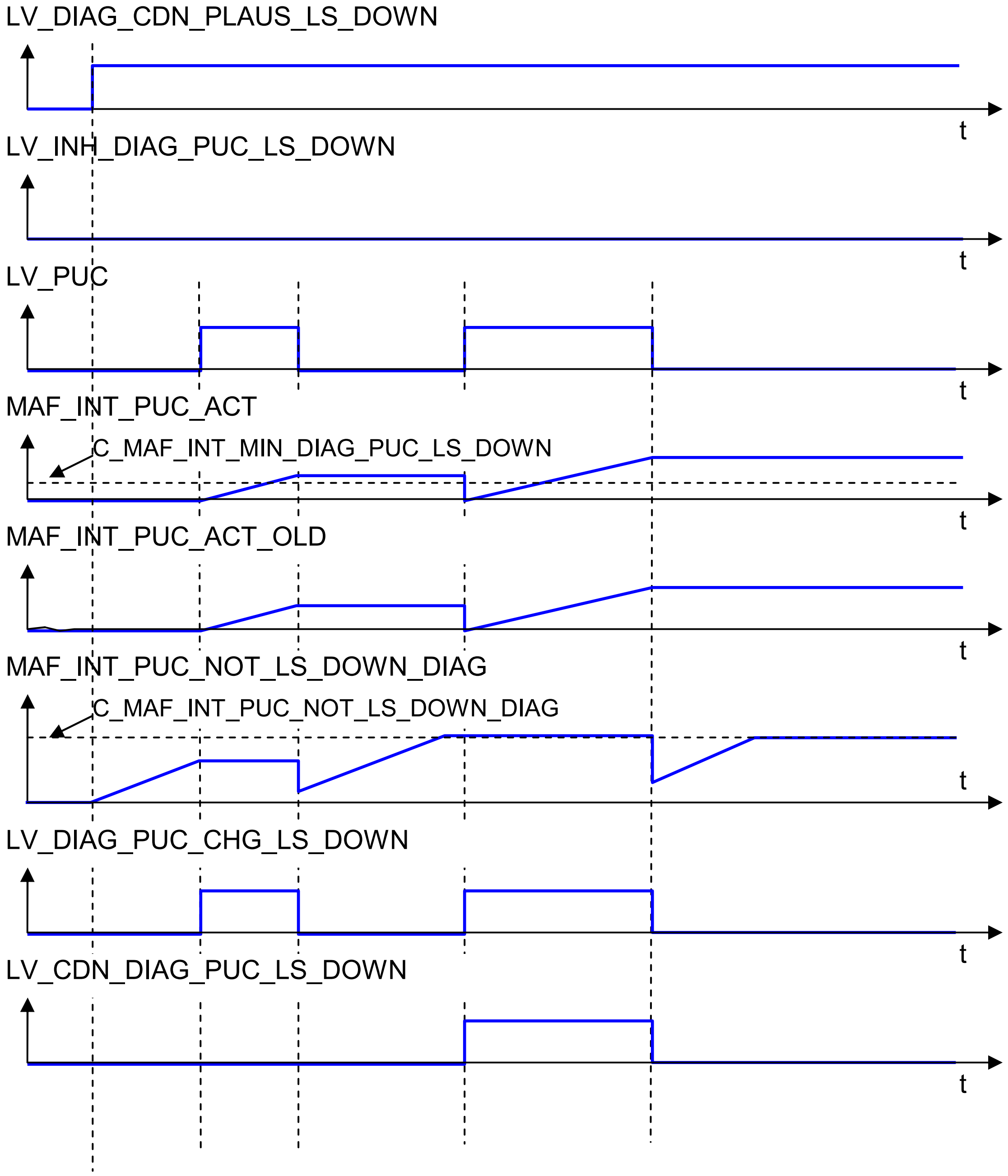
Should the oxygen sensor signal voltage (VLS_DOWN[i]) exceed the calibrateable thresholds (C_VLS_MIN_DIAG_PUC_LS_DOWN) during engine operating state PUC and the following additional conditions are met, then a fault shall be determined to be present:

A. The appropriate inhibit flag shall not be set (LV_INH_DIAG_PUC_LS_DOWN[i]).
B. Plausibility check set to be possible (LV_DIAG_CDN_PLAUS_LS_DOWN[i] = 1)
C. The PUC state (LV_PUC = 1) shall be determined to be stable, as determined by comparing the mass air flow integral calculated after activation (MAF_INT_PUC_ACT) with the threshold range (C_MAF_INT_MIN_VLS_DOWN_DIAG; C_MAF_INT_MAX_VLS_DOWN_DIAG).
D. The energy transfer from the exhaust gas to the oxygen sensor shall exceed the calibratable threshold (C_MAF_INT_PUC_NOT_LS_DOWN_DIAG).
E. The reset of the MAF integral from the last fuel cut off phase must have taken place before activating the diagnosis. Otherwise there is a risk to set the conditions to 1 right at the beginning of the fuel cut off phase. At this time, the lambda sensor voltage downstream is still high and the error symptom would be set for one recurrence.
F. Prior to the PUC phase the VLS_DOWN[i] must have exceeded the voltage thresholds C_VLS_DOWN_PUC_PLAUS_VLD and C_VLS_DOWN_PUC_AFR_VLD.

The error flag LV_ERR_PUC_LS_DOWN[i] shall be set once the anti-bounce counter has reached its maximum value and reset should the anti-bounce counter reach 0.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

Correlation between MAF integrals MAF_INT_PUC_ACT, MAF_INT_PUC_ACT_OLD and MAF_INT_PUC_NOT_LS_DOWN_DIAG and diagnosis activation :



Application conditions:

Initialisation:

At the transition LV_IGK = 0 -> 1, reset or clearing the error memory, reset of all variables and bits to 0, except VLS_DOWN_PUC_SAVE. VLS_UP_PUC_SAVE[i] shall be initialised to 0 at new ECU and in case of NVMY checksum errors.

Recurrence: **T_sample 100ms**

Activation / Deactivation: -

Formula section:

Conditions to start the downstream lambda sensor signal observation

```
If (1)      LV_PUC = 0
And(1)      VLS_DOWN[i] > C_VLS_DOWN_PUC_AFR_VLD
Then(1)     LV_VLS_DOWN_PUC_AFR_VLD[i] = 1
Elseif(1)   LV_PUC = 1 --> 0
Then(1)     LV_VLS_DOWN_PUC_AFR_VLD[i] = 0
Endif(1)
If(1)       LV_PUC = 0 --> 1
Then(1)     VLS_DOWN_PUC_PLAUS_VLD[i] = VLS_DOWN[i]
Endif(1)

If(1)       LV_DIAG_CDN_PLAUS_LS_DOWN[i] = 1
And(1)      LV_INH_DIAG_PUC_LS_DOWN[i] = 0
And(1)      LV_PUC = 1
Then(1)     LV_DIAG_PUC_CHG_LS_DOWN[i] = 1
    If(2)       C_MAF_INT_MAX_DIAG_PUC_LS_DOWN > MAF_INT_PUC_ACT > C_MAF_INT_MIN_DIAG_PUC_LS_DOWN
    And(2)      MAF_INT_PUC_NOT_LS_DOWN_DIAG[i] >= C_MAF_INT_PUC_NOT_LS_DOWN_DIAG
                % diagnosis active after a certain energy transfer out of fuel cut off
    And(2)      MAF_INT_PUC_ACT > MAF_INT_PUC_ACT_OLD
                % MAF integral during PUC from last DC must be reset before starting the diagnosis
    And(2)      VLS_DOWN_PUC_PLAUS_VLD[i] > C_VLS_DOWN_PUC_PLAUS_VLD
    And(2)      LV_VLS_DOWN_PUC_AFR_VLD[i] = 1

    Then(2)     "Diagnosis available"
                LV_CDN_DIAG_PUC_LS_DOWN[i] = 1
    Else(2)     "Diagnosis NOT available"
                LV_CDN_DIAG_PUC_LS_DOWN[i] = 0
    Endif(2)
Else(1)     LV_CDN_DIAG_PUC_LS_DOWN[i] = 0
            VLS_DOWN_PUC_PLAUS_VLD[i] = 0
```

```
        If(2a)        LV_PUC = 0
        And(2a)       LV_DIAG_CDN_PLAUS_LS_DOWN[i] = 1
        And(2a)       LV_INH_DIAG_PUC_LS_DOWN[i] = 0
        And(2a)       MAF_KGH > C_MAF_KGH_MIN_LS_DOWN_DIAG
        Then(2a)
            If(3a)        LV_DIAG_PUC_CHG_LS_DOWN[i] = 1 (fuel cut off was active)
            Then(3a)      $MAF\_INT\_PUC\_NOT\_LS\_DOWN\_DIAG[i]_N =$
                          $MAF\_INT\_PUC\_NOT\_LS\_DOWN\_DIAG[i]_{N-1} –$
                          C_FAC_MAF_INT_LS_DOWN_DIAG * MAF_INT_PUC_ACT
                          LV_DIAG_PUC_CHG_LS_DOWN[i] = 0
            Endif(3a)
            If(3b)        MAF_INT_PUC_NOT_LS_DOWN_DIAG[i] <=
                          C_MAF_INT_PUC_NOT_LS_DOWN_DIAG
            Then(3b)      $MAF\_INT\_PUC\_NOT\_LS\_DOWN\_DIAG[i]_N =$
                          $MAF\_INT\_PUC\_NOT\_LS\_DOWN\_DIAG[i]_{N-1}$ + MAF_KGH*
                          T_SAMPLE[ms] * 1/3600[(g*h) / (kg*ms)]

            Else(3b)      $MAF\_INT\_PUC\_NOT\_LS\_DOWN\_DIAG[i]_N =$
                          $MAF\_INT\_PUC\_NOT\_LS\_DOWN\_DIAG[i]_{N-1}$
            Endif(3b)
        Endif(2a)
    Endif(1)
    MAF_INT_PUC_ACT_OLD = MAF_INT_PUC_ACT
```

Conditions to start debouncing

```
If          LV_CDN_DIAG_PUC_LS_DOWN[i] = 1
then        If      VLS_DOWN[i] > C_VLS_MIN_DIAG_PUC_LS_DOWN
            then    “PUC signal plausibility fault present”
                    ERR_SYM_PUC_LS_DOWN[i] = sym_3   /"PUC PLAUSIBILITY ERROR"
                    /Error Manager shall debounce this symptom
                    VLS_DOWN_PUC_ERR[i] = VLS_DOWN[i]
            else
                    ERR_SYM_PUC_LS_DOWN[i] = "NO_SYM"
                    /Error Manager shall decrement the debounce counter
                    VLS_DOWN_PUC_OK[i] = VLS_DOWN[i]
            endif
endif
```

% At this point, update of the ERRM shall be completed.

```
% In the following part, current values of LV_END_DIAG_PUC_LS_DOWN[i] and LV_ERR_PUC_LS_DOWN[i]
% (after updating ERRM with the diagnostic result of this recurrence) shall be used:

if    LV_END_DIAG_PUC_LS_DOWN[i] = 1
then
      if LV_ERR_PUC_LS_DOWN[i] = 1
      then
            VLS_DOWN_PUC_SAVE[i] = VLS_DOWN_PUC_ERR[i]
      else
            VLS_DOWN_PUC_SAVE[i] = VLS_DOWN_PUC_OK[i]
      endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MAF_INT_MAX_DIAG_PUC_LS_DOWN	1	0...FFFFH	0...2912.66666	4.44E-02	[g]
Maximum threshold for MAF integral to end downstream PUC signal plausibility					
C_MAF_INT_MIN_DIAG_PUC_LS_DOWN	1	0...FFFFH	0...2912.66666	4.44E-02	[g]
Minimum threshold for MAF integral required to start downstream PUC signal plausibility					
C_VLS_MIN_DIAG_PUC_LS_DOWN	1	0...3FFH	0...4.99511	4.88E-03	[V]
VLS voltage threshold for PUC fault detection					
C_MAF_INT_PUC_NOT_LS_DOWN_DIAG	1	0...FFFFH	0...2912.66666	4.44E-02	[g]
Minimum threshold for MAF integral out of PUC required to start downstream PUC signal plausibility					
C_FAC_MAF_INT_LS_DOWN_DIAG	1	0...FFH	0...31.875	0.125	[-]
Factor for impact of fuel cut off phase to the sensor temperature					
C_MAF_KGH_MIN_LS_DOWN_DIAG	1	0...FFFFH	0...2047.96875	3.125E-2	kg/h
Minimum MAF_KGH to start MAF integral					
C_VLS_DOWN_PUC_PLAUS_VLD	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Rich voltage threshold for activation of PUC plausibility diagnosis					
C_VLS_DOWN_PUC_AFR_VLD	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Rich voltage monitoring threshold between fuel-cut phases					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
PUC_LS_DOWN[NC_CBK_EX_NR]	Signal high during fuel cut off	SYM_3	MEM

16.33.2 Full load (FL) oxygen sensor signal plausibility

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean error flag, FL plausibility fault currently present on downstream oxygen sensor signal					
ERR_SYM_FL_LS_DOWN[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Boolean error flag, fault currently present on downstream oxygen sensor signal					
LV_CDN_DIAG_FL_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status conditions of oxygen sensor downstream diagnosis					
LV_END_DIAG_FL_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Bit to indicate end of complete downstream oxygen sensor signal diagnosis					

Input data:

VLS_DOWN[NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_IGK	LV_DIAG_CDN_PLAUS_LS_DOWN[NC_CBK_EX_NR]
LV_INH_DIAG_FL_LS_DOWN[NC_CBK_EX_NR]	MAF_INT_FL[NC_CBK_EX_NR]		

FUNCTION DESCRIPTION:

General information:

Description:

Should the oxygen sensor signal voltage (VLS_DOWN[i]) exceed the calibrateable thresholds (C_VLS_MAX_DIAG_FL_LS_DOWN) during engine operating state FL and the following additional conditions are met, then a fault shall be determined to be present:

A. The appropriate inhibit flag shall not be set (LV_INH_DIAG_FL_LS_DOWN[i]).
B. Plausibility check set to be possible (LV_DIAG_CDN_PLAUS_LS_DOWN[i] = 1)
C. The FL state (LV_FL = 1) shall be determined to be stable, as determined by comparing the mass air flow integral calculated after activation (MAF_INT_FL[i]) with the threshold range (C_MAF_INT_MIN_DIAG_FL_LS_DOWN; C_MAF_INT_MAX_DIAG_FL_LS_DOWN).

The error flag LV_ERR_FL_LS_DOWN[i] shall be set once the anti-bounce counter has reached its maximum value and reset should the anti-bounce counter reach 0.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation:

At the transition LV_IGK = 0 -> 1, reset or clearing the error memory, reset of all variables and bits to 0

Recurrence: **T_sample 100ms**

Activation / Deactivation:

```
If      LV_DIAG_CDN_PLAUS_LS_DOWN[i] = 1
and     LV_INH_DIAG_FL_LS_DOWN[i] = 0
and     C_MAF_INT_MAX_DIAG_FL_LS_DOWN > MAF_INT_FL[i]
        > C_MAF_INT_MIN_DIAG_FL_LS_DOWN
then    "Diagnosis available"
        LV_CDN_DIAG_FL_LS_DOWN[i] = 1
else    "Diagnosis NOT available"
        LV_CDN_DIAG_FL_LS_DOWN[i] = 0
endif
```

Formula section:

```
If      LV_CDN_DIAG_FL_LS_DOWN[i] = 1
then    If      VLS_DOWN[i] < C_VLS_MAX_DIAG_FL_LS_DOWN
        then    "FL signal plausibility fault present"
                ERR_SYM_FL_LS_DOWN[i] = sym_3 / "FL PLAUSIBILITY ERROR"
                /Error Manager shall debounce this symptom
        else
                ERR_SYM_FL_LS_DOWN[i] = "NO_SYM"
                /Error Manager shall decrement the debounce counter
        endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_MAF_INT_MAX_DIAG_FL_LS_DOWN	1	0...FFFFH	0...1820.42	0.0277778	[g]
Maximum threshold for MAF integral to end downstream FL signal plausibility					
C_MAF_INT_MIN_DIAG_FL_LS_DOWN	1	0...FFFFH	0...1820.42	0.0277778	[g]
Minimum threshold for MAF integral required to start downstream FL signal plausibility					
C_VLS_MAX_DIAG_FL_LS_DOWN	1	0...3FFH	0...4.99511	4.8828e-3	[V]
VLS voltage threshold for FL fault detection					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
FL_LS_DOWN[NC_CBK_EX_NR]	Signal low during full load	SYM_3	MEM

16.33.3 Calculation of MAF_INT_FL

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_INT_FL[NC_CBK_EX_NR]	V/O	0...FFFFH	0...1820.42	0.0277778	[g]
integral of air mass flow while FL active					

Input data:

LV_FL	MAF_CYL	LV_ST_END	LV_IGK
VLS_UP[NC_CBK_EX_NR]			

FUNCTION DESCRIPTION:

General information:

Description:

The calculation of MAF_INT_FL[i] expires in 20ms raster as long as either the operation status FL is fulfilled or VLS_UP[i] indicates rich conditions, depending on the setting of LC_MAF_INT_FL_CDN_AFR. The integral is calculated even when the application conditions for the plausibility check function are not fulfilled. MAF_INT_FL[i] is limited to the maximum value if necessary in case of a longer FL (or rich) phase.
The value of MAF_INT_FL[i] should reset to 0 if the transition LV_FL = 1 -> 0 takes place resp. if VLS_UP[i] no longer indicates rich.

Remarks:
Although the upstream sensor signal is used here if LC_MAF_INT_FL_CDN_AFR = 1, the calculation is still usable for both linear and binary upstream sensors. Only one of the calibrateable VLS_UP limits should be calibrated for a given type: the upper limit for linear sensors, and the lower limit for binary sensors. The other one is not meaningful and should be set to the physical limit.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation:

At the transition LV_IGK = 0 -> 1 or reset reset of all variables and bits to 0

Recurrence: **Tsample = 20ms**

Activation / Deactivation:

```
If      LV_ST_END = 1
then    calculation enabled
else    calculation disabled
endif
```

Formula section:

```
If   (LC_MAF_INT_FL_CDN_AFR = 0 and LV_FL = 1) or
     (LC_MAF_INT_FL_CDN_AFR = 1 and
     VLS_UP[i] ≥ C_VLS_UP_MIN_MAF_INT_AFR and
     VLS_UP[i] ≤ C_VLS_UP_MAX_MAF_INT_AFR)
then
     $MAF\_INT\_FL[i]_n \ [g] = MAF\_INT\_FL[i]_{n-1} \ [g] + MAF\_CYL * T\_SAMPLE \ [ms] * 1/3600 \ [(g*h) / (kg*ms)]$
else
     MAF_INT_FL[i] = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VLS_UP_MIN_MAF_INT_AFR	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Upstream minimum VLS to enable MAF_INT_AFR integration					
C_VLS_UP_MAX_MAF_INT_AFR	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Upstream maximum VLS to enable MAF_INT_AFR integration					
LC_MAF_INT_FL_CDN_AFR	1	0...1H	0...1	1	[-]
Flag to indicate that upstream rich signal instead of LV_FL should enable the MAF_INT_FL integral					

16.34 Downstream oxygen sensor heater OBDI diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Error flag for downstream heater diagnosis OBD I after debouncing					
LV_CDN_DIAG_LSH_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Diagnosis condition for downstream heater diagnosis OBD I					
LV_END_DIAG_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
End of diagnosis for downstream heater diagnosis OBD I					
ERR_DIAG_LSH_DOWN[NC_CBK_EX_NR]	V	0H 1H 2H 4H	NO_SYM SYM_0 SYM_1 SYM_2	1	[-]
Raw value of error symptom for downstream heater diagnosis OBD I					
ERR_SYM_LSH_DOWN[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom for downstream heater diagnosis OBD I filterd with CDN_DIAG_LSH_DOWN					
CDN_DIAG_LSH_DOWN[NC_CBK_EX_NR]	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of downstream heater diagnosis OBD I					

Input data:

LV_IGK	STATE_LSH_DOWN[NC_CBK_EX_NR]	**LV_VB_CDN_OBD_1**	TEG_DYN_LS_DOWN[NC_CBK_EX_NR]
LSHPWM_DOWN[NC_CBK_EX_NR]	LV_INH_DIAG_LSH_DOWN[NC_CBK_EX_NR]	NC_CBK_EX_NR	

Import actions:

ACTION_ERRM_FilterMulticondition(IN <*XX* >, IN< CDN_DIAG_ XX>, IN< ERR_DIAG_XX>, IN <C_ABC_INC_XX>, IN <C_ABC_MAX_XX>, OUT<LV_ERR_XX>)
This action computes the elementary antibounce filter for one failure treatment and returns filter result
ACTION_INFR_GetElDiagLshDown(IN <i>, OUT < cdn_diag>, OUT <err_diag>)
This action reads the failure and the conditions information for each symptom from the infrastructure

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.
All symptoms of the current error code are handeled by anti-bouncing.
We assume that only one symptom of an error code can be active at the same time.

Description:

The error detection is effected via the ECU hardware.

The purpose of the diagnosis shall be to detect electrical faults within the oxygen sensor heater circuit as defined by OBD I requirements.

Error-symptoms and conditions: are defined to this diagnosis function as following

```
☐☐☐☐
 │ │ │ └─> Short circuit to battery (SCP)          (= SYM_0)
 │ │ └───> Short circuit to ground (SCG)           (= SYM_1)
 │ └─────> Open circuit          (OC)          (= SYM_2)
 └───────> -                                       (= SYM_3)
```

The function is intended for use in a hardware configuration where the heater driver shall carry out the electrical fault diagnosis and provide the electrical status via status flags.

- *Short circuit to Vbatt by Over-temperature* ***or*** *Overcurrent* (**SCP**)
- *Short circuit to GND* (SCG)
- *Open load* (**OC**)

The heater drivers named are capable of detecting an **OC** and **SCP** fault when in the ON state and only capable of detecting a SCG fault in the OFF state. As the heater power is controlled by a PWM signal, the driver will be placed alternately in an ON and then OFF state.

The diagnosis shall be activated should the following conditions be met:

- Engine leaving the engine start state as denoted by LV_IGK = 1
- Battery voltage falls between pre-set thresholds to protect against false errors at battery voltage extremes as indicated by **LV_VB_CDN_OBD_1**.
- CDN depending on duty cycle of LSHPWM_DOWN[i]
- Inhibit flag not set (LV_INH_DIAG_LSH_DOWN[i])

In exceptionally rare extreme environmental conditions for certain system configurations, it is possible that the heater driver may be driven into over-temperature during initial oxygen sensor heating. In this case, the driver automatically turns off the output stage in self-protection and sets the **SCP** electrical fault bit. In order to mask an incorrect oxygen sensor heater **SCP** fault, the function shall only allow the **SCP** to set a fault during the initial heater phase, if the exhaust gas temperature has exceeded a pre-determined calibrateable threshold.

The oxygen sensor heater fault detection shall be debounced by use of an anti-bounce counter. This shall be handled by the Dynamic Error Management.

Each of the above listed faults shall be described individually below:

Otherwise the diagnosis has to be suppressed (CDN_DIAG_LSH_DOWN[i] = 0). This is managed within the “Diagnosis condition calculation.

Application conditions:

Initialization: at reset CDN_DIAG_LSHDOWN[i] = 0,

ERR_DIAG_LSH_DOWN[i] = 0

All other variables are initialized according filter type

Activation : -

Deactivation: -

Recurrence: 100 ms

NOTE: The heater PWM frequency shall be at least 2 times faster than the diagnosis frequency to ensure that an oversampling effect does not occur!

<u>**Formula section:**</u>

```
If      LV_IGK = 1
        And   LV_VB_CDN_OBD_1 = 1
        and   LV_INH_DIAG_LSH_DOWN[i] = 0
then
```

<u>Usage of the diagnosis information (failure symptoms (raw value)</u> **ERR_DIAG_LSH_DOWN[i]** <u>and basic diagnosis conditions</u> **CDN_DIAG_LSH_DOWN[i]**) <u>received from the infrastructure:</u>

ACTION_INFR_GetElDiagLshDown (IN <i>, **OUT**<CDN_DIAG_LSH_DOWN[i]>, **OUT**<ERR_DIAG_LSH_DOWN[i]>**)**

Basic diagnosis conditions are set according infrastructure information: CDN_DIAG_LSH_DOWN[i]. Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_LSH_DOWN[i].

```
    If      Activation conditions are met for the NC_PSD_DLY_LSH_DOWN recurrence
    Then
            If      LSHPWM_DOWN[i] < C_LSHPWM_MIN_DIAG_LSH_DOWN
            or      [(STATE_LSH_DOWN[i] = LSH_POW_RISE
            or      STATE_LSH_DOWN[i] = LSH_POW_RED)
            and     TEG_DYN_LS_DOWN[i] <
                    C_TEG_DYN_MIN_DIAG_LSH_DOWN]
            then    bit 0 of CDN_DIAG_LSH_DOWN[i] = 0 (SCP can not be detected))
            endif
            If      LSHPWM_DOWN[i] > C_LSHPWM_MAX_DIAG_LSH_DOWN
            then    bit 1 of CDN_DIAG_LSH_DOWN[i] = 0  (SCG can not be detected)
            endif
            If      LSHPWM_DOWN[i] < C_LSHPWM_MIN_DIAG_LSH_DOWN
            or      LSHPWM_DOWN[i]  > C_LSHPWM_MAX_DIAG_LSH_DOWN
            then    bit 2 of CDN_DIAG_LSH_DOWN[i] = 0  (OC can not be detected)
            endif
```

```
        Else
                CDN_DIAG_LSH_DOWN[i] = 0
        Endif
    Else
                CDN_DIAG_LSH_DOWN[i] = 0
    endif
```

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_LSH_DOWN[i] and ERR_DIAG_LSH_DOWN[i].

ACTION_ERRM_FilterMulticondition(IN <*XX*>, IN< CDN_DIAG_LSH_DOWN[i]>, IN< ERR_DIAG_LSH_DOWN[i]>, IN <C_ABC_INC_LSH_DOWN>, IN <C_ABC_MAX_LSH_DOWN>, OUT<LV_ERR_LSH_DOWN[i]>)

This algorithm determines:

ERR_SYM_LSH_DOWN[i]	(= raw value ERR_DIAG_LSH_DOWN[i] filtered with CDN_DIAG_LSH_DOWN[i], detected error symptom for lambda sensor heater upstream [i] diagnosis)
LV_ERR_ LSH_DOWN[i]	(Error flag for debounced error of for lambda sensor heater upstream [i])
LV_CDN_DIAG_ LSH_DOWN[i]	(Diagnosis condition information)
LV_END_DIAG_ LSH_DOWN[i]	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LSHPWM_MIN_DIAG_LSH_DOWN	1	0...FFH	0...99.60937	0.390625	[%]
Minimum O2-Sensor heater duty cycle to perform the electrical diagnosis					
C_LSHPWM_MAX_DIAG_LSH_DOWN	1	0...FFH	0...99.60937	0.390625	[%]
Maximum O2-Sensor heater duty cycle to perform the electrical diagnosis					
C_TEG_DYN_MIN_DIAG_LSH_DOWN	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Minimum exhaust gas temperature threshold to permit SCP to set fault during LSH power rise state					
C_ABC_INC_LSH_DOWN	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_LSH_DOWN	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
Downstream oxygen sensor heater OBDI diagnosis	*Short circuit to battery*	*SYM_0*	*MPL_STD_INI*
	Short circuit to GND	*SYM_1*	
LSH_DOWN[i]	*Open circuit*	*SYM_2*	

	4th symptom description	SYM_3	

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_LSH_DOWN	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (typical 2)					

16.34.1 Application incidences for downstream oxygen sensor heater OBDI diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibit flag for downstream oxygen sensor heater OBDI diagnosis					

Input data:

LV_ST_END	NC_CBK_EX_NR	LV_CDN_VB_OBD1	LV_ERR_SPI_MPS
LV_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR]			

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

Description:

The diagnosis is inhibited due to low battery voltage, present error or if sensors are not learnt.

Application conditions:

Initialisation: LV_INH_DIAG_LSH_DOWN[i] = 1 at reset

Recurrence: The function shall be carried out once every 100 ms.

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0 , set LV_INH_DIAG_LSH_DOWN[i] = 1

Formula section:

If LV_CDN_VB_OBD1 = 0 **or**
LV_ERR_SPI_MPS = 1 **or**
LV_LS_DOWN_OBD_1_MAN_DEAC[i] = 1
Then LV_INH_DIAG_LSH_DOWN[i] = 1
Else LV_INH_DIAG_LSH_DOWN[i] = 0
Endif

16.35 WRAF Sensor Electric Diagnosis with the ATIC42

16.35.1 HW-based Diagnosis with the ATIC42 (WRAF Sensor Controller)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounced diagnosis value caused by an OBD-I error at the WRAF-Sensor					
LV_LSL_UP_SPI_COM_INH[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibit bit for the SPI-communication related to electrical diagnosis on BSW level					
ERR_SYM_EL_LSL_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom of electrical failure on the upstream WRAF sensor					
LV_CDN_DIAG_EL_LSL_UP[NC_CBK_EX_NR]	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Condtions to run the electrical diagnosis of the upstream WRAF sensor					
LV_DIAG_ACT_EL_LSL_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical Value for diagnosis active (=1) or not (=0)					
LV_END_DIAG_EL_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Result of present diagnostis status is valid					
STATE_ERR_EL_LSL_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH	NO_FAULT SCG_LINE_RCD SCG_LINE_VIP SCG_LINE_VG SCG_LINE_VN SCG SCBAT_LINE_RCD SCBAT_LINE_VIP SCBAT_LINE_VG SCBAT_LINE_VN SCBAT	1	[-]
State to distinguish the OBD-I failures					
T_DLY_DIAG_EL_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Delay timer to start the electrical failure diagnosis					

Input data:

LV_ES	LV_LSL_IF_SPI_RST_END[NC_CBK_EX_NR]	LV_ST_END	LV_INH_DIAG_EL_LSL_UP[NC_CBK_EX_NR]
STATE_ERR_VNLSL[NC_CBK_EX_NR]	STATE_ERR_VGLSL[NC_CBK_EX_NR]	STATE_ERR_VIPLSL[NC_CBK_EX_NR]	STATE_ERR_VRCLSL[NC_CBK_EX_NR]
NC_CBK_EX_NR	LV_IGK		

FUNCTION DESCRIPTION:

General information:

This function shall debounce OBD-I errors at the Wide Range Air/Fuel (WRAF) Sensor detected by the ATIC42 and define a variable containing the diagnosis register of the controller.

If the error is recognised as debounced, the function shall generate an error flag and inhibit the SPI-communication between basic software (BSW) and ATIC42. This is necessary, because in case an electrical failure, the ATIC42 disconnect and reconnect all the PINs of the sensor by itself every recurrence of the BSW, by reading of the Diagnosis-register. This is harmful for the sensor and must be prevented.

According the exhaust gas system the variable shall be defined from [i]=1 to NC_CBK_EX_NR.

Error Symptoms are defined as follows:

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │ │ │ └─> Short circuit to battery “SCP”   ( = SYM_0)
  │ │ └───> Short circuit to Ground “SCG”  ( = SYM_1)
  │ └─────> (not used)
  └───────> (not used)
```

Application conditions:

Initialisation:

The following variables, related to the diagnosis manager shall be initialized according to the filter type “MEM”.
(all 0 at LV_IGK 0->1 or at ECU reset or clearing error memory)

LV_ERR_EL_LSL_UP[i] = 0

LV_CDN_DIAG_EL_LSL_UP[i] = 0

LV_END_DIAG_EL_LSL_UP[i] = 0

ERR_SYM_EL_LSL_UP[i] = “NO_SYM”

The following data shall be set to 0 at LV_IGK 0->1 or at ECU reset or at clearing error memory

STATE_ERR_EL_LSL_UP[i] = ‘No Fault’

LV_LSL_UP_SPI_COM_INH[i] = 0

T_DLY_DIAG_EL_LSL_UP[i] = 0

Recurrence:

The function shall be carried out once every 10ms.

Activation / Deactivation:

If

LV_ST_END = 1 **and**
LV_ES = 0 **and**
LV_ERR_EL_LSL_UP[i] = 0 **and**
LV_LSL_IF_SPI_RST_END[i] = 1 **and**

```
     LV_INH_DIAG_EL_LSL_UP[i] = 0
Then
        LV_DIAG_ACT_EL_LSL_UP[i] = 1
        The diagnosis shall be carried out
Else
     LV_ DIAG_ACT_EL_LSL_UP[i] = 0
        The diagnosis shall be deactivated
Endif
```

Formula section:

The variable STATE_ERR_EL_LSL_UP[i] is usefull during the development phase of the ASIC and shall be built up by the failure-bits STATE_ERR_VNLSL[i], STATE_ERR_VGLSL[i], STATE_ERR_VIPLSL[i], STATE_ERR_VRCLSL[i], delivered from the BSW. This variable shall have one of the following values according to the following table:

STATE_ERR_ VRCLSL[i]	STATE_ERR_ VIPLSL[i]	STATE_ERR_ VGLSL[i]	STATE_ERR_ VNLSL[i]	Value for STATE_ERR_ EL_LSL_UP
0	0	0	0	0H
1	0	0	0	1H
0	**1**	0	0	2H
0	0	**1**	0	3H
0	0	0	**1**	4H
"1" in more then one Variable				5H
2	0	0	0	6H
0	**2**	0	0	7H
0	0	**2**	0	8H
0	0	0	**2**	9H
"2" in more then one Variable				AH

Setting the symptoms:

```
If (1)   LV_DIAG_ACT_EL_LSL_UP[i] = 1
Then (1)
     If (2) T_DLY_DIAG_EL_LSL_UP[i] > C_T_DLY_DIAG_EL_LSL_UP
     Then
          LV_CDN_DIAG_EL_LSL_UP[i] = 1
          If (3) 1H ≤ STATE_ERR_EL_LSL_UP[i] ≤ 5H
          Then (3)
             ERR_SYM_EL_LSL_UP[i] = sym_1  /% "Short to ground"
          Else (3)
             If  (4)
                        6H ≤ STATE_ERR_EL_LSL_UP[i] ≤ AH
             Then (4)
                       ERR_SYM_EL_LSL_UP[i] = sym_0   / % "Short to plus"
             Else (4)
                       ERR_SYM_EL_LSL_UP[i] = NO_SYM   /% no symptom
```

```
            Endif (4)
        Endif (3)
    Else (2)
      Increment T_DLY_DIAG_EL_LSL_UP[i]
    Endif (2)
Endif (1)
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

```
Apply filter on current symptoms
If filtering result available  (after debounce)
then
    LV_ERR_EL_LSL_UP[i] = filtering result
    LV_END_DIAG_EL_LSL_UP[i] = 1
    Delivery of the result to the Error Management
Endif
```

LV_LSL_UP_SPI_COM_INH[i] = LV_ERR_EL_LSL_UP[i]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_DIAG_EL_LSL_UP	1	0...FFFFH	0...655.35	0.01	[s]
Time delay to start the electrical failure diagnosis					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
EL_LSL_UP[i]	Short to plus	SYM_0	MEM
	Short to ground	SYM_1	

16.35.2 WRAF Sensor Pump Current Controller Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_IPLSL_SYM_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Pump Current Controller in Limitation for too long time					
LV_DIAG_IPLSL_CDN_LSL_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Conditions to run the WRAF sensor Pump Current Controller Diagnosis					
IPLSL_SYM_DIAG_LSL_UP[NC_CBK_EX_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the WRAF sensor Pump Current Controller Diagnosis					
CTR_IPLSL_SYM_DIAG_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Error debounce counter					

Input data:

STATE_ERR_IPLSL[NC_CBK_EX_NR]	ERR_SYM_EL_LSL_UP[NC_CBK_EX_NR]	STATE_LSH_UP[NC_CBK_EX_NR]	NC_CBK_EX_NR
LV_IPLSL_DIAG_INH_LSL_UP[NC_CBK_EX_NR]	LC_ICPLSL_ACT	LV_ICPLSL_ENA_LSL_IF[NC_CBK_EX_NR]	LV_IPLSL_CTL_ENA_LSL_IF[NC_CBK_EX_NR]
STATE_SYM_OBD_LSL_LSH_UP[NC_CBK_EX_NR]	LAMB_SP[NC_CBK_EX_NR]	LV_IGK	

FUNCTION DESCRIPTION:

General information:

This function shall monitor the *pump current controller* inside the WRAF sensor controller ATIC42 in order to protect the oxygen sensor against operation with high pump current (Ip) during excessive time which can be harmfull to the sensor.

High pump current for long time is allowed only during the warm up phase of the WRAF sensor, when its ceramic temperature not high enough, but the Ip controller activaton is needed to detect the sensor operability as fast as possible. To avoid false error messages this function shall run only after the sensor heater achieves its steady state after the warm up phase, when it starts to work in closed loop.

To avoid any double diagnostic it shall be monitored if no sensor electrical failure exists (short circuit to ground or to battery voltage) or if the heater management works properly. In both cases it is possible due to a failure to flow a high Ip current through the sensor. Similarly it must be asked if the sensor requires an *reference air pump current* and in positive case, if this is active.

If a *pump current controller* failure symptom is detected, the use of the sensor is not more possible and the lambda controller shall be deactivated immediately. Should this symptom be completely debounced, the sensor shall not be used for any purpose during the current driving cycle and also the *pump current controller* shall be deactivated in order to avoid any sensor damage.

Description:

This diagnosis serves to debounce the information concerning the conditions of the *pump current controller* which *is* the delivered by the ATIC42 via its Basic Software.
As long as the Ip current is at their bounds, which is defined *per* hardware [1], the ATIC42 limits the *pump current controller* output and writes in a warning message in its diagnosis register, indicating that Ip current is too low or too high. The Basic Software dedicated to the ATIC42 reads out every 10 ms the diagnosis register and save this information in the variable STATE_ERR_IPLSL[i]. This shall be evaluated here and in failure case the symptom debounced.

[1] For more informations please see the ATIC42 specification.

Application conditions:

Initialisation:

At the transition LV_IGK = 0 -> 1, by ECU-reset or at clearing the error memory shall all variables be reseted.

Recurrence:

The function shall be carried out once every 100ms.

Note: Altough the variable STATE_ERR_IPLSL[i] is updated every 10 ms, it is not necessary to evaluate it so frequently. The decision to stop the Ip controller must not be done with a resolution of miliseconds but seconds. It can be taken advantage to that point allowing to filter transient Ip-controller conditions.

Activation / Deactivation:

```
If      (LAMB_SP[i] < C_LAMB_SP_AFR_IPLSL_DIAG_LSL_UP  or
        LAMB_SP[i] > C_LAMB_SP_AFL_IPLSL_DIAG_LSL_UP)            and

    LV_IPLSL_CTL_ENA_LSL_IF[i] = 1                               and
    (Ip controller active)
    LV_IPLSL_DIAG_INH_LSL_UP[i] = 0                              and
    (no restriction to run this diagnosis)
        ERR_SYM_EL_LSL_UP[i] = “NO SYM”                          and
        (no sensor electrical failure)
        STATE_LSH_UP[i] = “LSH_POW_CTL”                          and
        (sensor is heated properly)
    STATE_SYM_OBD_LSL_LSH_UP[i] = “NO SYM”                       and
    (no sensor under- or overheating)
        [ LV_ICPLSL_ENA_LSL_IF[i] = 1 or
     (LV_ICPLSL_ENA_LSL_IF[i] = 0 and LC_ICPLSL_ACT = 0)]
    (ref. Air is properly pumped if it is needed)
then
    LV_DIAG_IPLSL_CDN_LSL_UP[i] = 1
    (activate this function)
else
    LV_DIAG_IPLSL_CDN_LSL_UP[i] = 0
    (deactivate this function)
endif
```

Formula section:

```
If  STATE_ERR_IPLSL[i] = ”current high”
Then
     IPLSL_SYM_DIAG_LSL_UP[i] = “sym_0”
Else
          If  STATE_ERR_IPLSL[i] = ”current low”
     Then
               IPLSL_SYM_DIAG_LSL_UP[i] = “sym_1”
     Else
               IPLSL_SYM_DIAG_LSL_UP[i] = “no sym”
          Endif
Endif

% Debouncing error symptom
If  IPLSL_SYM_DIAG_LSL_UP[i] ≠ “no_symptom”
Then
          If  CTR_IPLSL_SYM_DIAG_LSL_UP[i]  > C_CTR_IPLSL_SYM_DIAG_LSL_UP
          Then
          LV_DIAG_IPLSL_SYM_LSL_UP[i] = 1
          Else
               Increment CTR_IPLSL_SYM_DIAG_LSL_UP[i]
          Endif
Else
     LV_DIAG_IPLSL_SYM_LSL_UP[i] = LC_IPLSL_SYM_DIAG_LSL_UP
     CTR_IPLSL_SYM_DIAG_LSL_UP[i] = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_IPLSL_SYM_DIAG_LSL_UP	1	0...FFFFH	0...65535	1	[-]
Max. error debounce counter					
C_LAMB_SP_AFR_IPLSL_DIAG_LSL_UP	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Max. rich lambda set point to activate the IP diagnosis					
C_LAMB_SP_AFL_IPLSL_DIAG_LSL_UP	1	0...7FFFH	0...31.99902	9.77E-04	[-]
Min. lean lambda set point to activate the IP diagnosis					
LC_IPLSL_SYM_DIAG_LSL_UP	1	0...1H	0...1	1	[-]
Logical variable to force IPLSL symptom flag (only for test purpose)					

16.36 O2 sensor (up, down) diag management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Final diagnostic of the upstream oxygen sensor					
LV_LS_UP_DIAG_END[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Sum end bit of upstream oxygen sensor diagnosis functions					
LV_LS_UP_OBD_1_MAN_DEAC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Logical variable indicating manual deactivation of all OBD1 diagnosis for upstream oxygen sensor					
LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Logical variable indicating manual deactivation of all OBD2 diagnosis for upstream oxygen sensor					
LV_ERR_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Final diagnostic of the downstream oxygen sensor					
LV_LS_DOWN_DIAG_END[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Collecting end flag for the diagnosis functions of the downstream oxygen sensor					
LV_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Logical variable indicating manual deactivation of all OBD1 diagnosis for downstream oxygen sensor					
LV_LS_DOWN_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Logical variable indicating manual deactivation of all OBD2 diagnosis for downstream oxygen sensor					

Input data:

LV_ST_END	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_LSH_LSL_UP[NC_CBK_EX_NR]
LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]	LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]
NC_CBK_EX_NR	LV_IGK		
LV_END_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]	LV_END_DIAG_EL_LSL_UP[NC_CBK_EX_NR]	LV_END_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	LV_END_DIAG_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_END_DIAG_LSH_LSL_UP[NC_CBK_EX_NR]	LV_END_DIAG_LSH_UP[NC_CBK_EX_NR]	LV_END_DIAG_PUC_VLD_LS_UP[NC_CBK_EX_NR]	LV_END_DIAG_LSL_UP_IF[NC_CBK_EX_NR]
LV_END_DIAG_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	LV_END_DIAG_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	LV_END_DIAG_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	LV_LS_UP_OBD_MAN_DEAC_EXT
LV_ERR_CHG_LS_UP	LV_END_DIAG_CHG_LS_UP		
LV_ERR_EL_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_PLAUS_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_CHG_LS_DOWN
LV_END_DIAG_SWT_LS_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_CHK_LS_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_FL_LS_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_LSH_DOWN[NC_CBK_EX_NR]
LV_END_DIAG_PUC_LS_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_OBD_LSH_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]
LV_END_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_OC_LS_DOWN[NC_CBK_EX_NR]	LV_LS_DOWN_OBD_MAN_DEAC_EXT	LV_ERR_CHG_LS_DOWN

16.36.1 Upstream O2 sensor final diagnosis

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

This function shall summarize the diagnostic status of the upstream oxygen sensor delivered from the OBD I & II functions (heater and sensor) through the flag LV_ERR_LS_UP[i].

The ultimate error flag LV_ERR_LS_UP[i] is a component error and no DTC shall be sended to the Error Management.

To reduce the number of variables to be checked by other diagnosis functions, the collecting flag LV_LS_UP_DIAG_END is introduced. It provides the possibility to combine the end flags from almost all the upstream linear lambda sensor diagnosis functions. By a bit mask variable it is calibratable which end flag contributes to the collecting end flag.

Note:
1. If two separate Banks are concerned, then i=1 for cylinder bank 1 and i=2 for cylinder bank 2.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset **or** at clearing error memory:

LV_ERR_LS_UP[i] = 0
LV_LS_UP_OBD_1_MAN_DEAC[i] = 0
LV_LS_UP_OBD_2_MAN_DEAC[i] = 0
LV_LS_UP_DIAG_END[i] = 0

Recurrence:

The function should be carried out once every 100 ms.

Activation/ Deactivation:

The diagnosis will be carried out if LV_ST_END = 1

Formula section:

```
If      LV_LS_UP_OBD_MAN_DEAC_EXT = 1
Then    LV_LS_UP_OBD_1_MAN_DEAC[i] = 1
        LV_LS_UP_OBD_2_MAN_DEAC[i] = 1
Else
```

```
              LV_LS_UP_OBD_1_MAN_DEAC[i] = LC_LS_UP_OBD_1_MAN_DEAC[i];
              LV_LS_UP_OBD_2_MAN_DEAC[i] = LC_LS_UP_OBD_2_MAN_DEAC[i];
Endif
```

The linear lambda sensor error flag shall be set under the following conditions:

```
IF
   LC_LS_UP_DIAG_ACT[i] = 1  % activation of final diagnosis                AND
  ( LV_ERR_EL_LSL_UP[i] = 1    % electrical failure                         OR
  LV_ERR_OC_LSL_UP[i] = 1   % open circuit failure                          OR
  LV_ERR_LSL_UP_IF[i] = 1     % linear sensor controller failure            OR

  LV_ERR_LSH_UP[i] = 1                  % heater electrical failure         OR
  LV_ERR_OBD_VLD_LSH_UP[i] = 1   % heater controller failure                OR
  LV_ERR_LSH_LSL_UP[i] = 1            % heater to sensor signal coupling failure   OR

  LV_ERR_AIR_LSL_UP[i] = 1   % sensor not mounted in exhaust pipe           OR
  LV_ERR_PUC_VLD_LS_UP[i]= 1      % sensor charact. line slope failure      OR
  LV_ERR_SHIFT_AFL_LSL_UP[i]= 1  % sensor charact. line shift to lean failure   OR
  LV_ERR_SHIFT_AFR_LSL_UP[i]= 1 % sensor charact. line shift to rich failure    OR

  LV_ERR_DYN_VLD_LS_UP[i]=1 & LC_ERR_DYN_VLD_LS_UP = 1                     OR
    % sensor dynamic failure (allowed by the project to stop closed loop control)
  LV_ERR_CHG_LS_UP =1 & LC_ERR_CHG_LS_UP = 1                               OR
    % sensor interchanged failure (allowed by the project to stop closed loop control)

   LC_LS_UP_ERR[i] = 1)          % manual activation of lambda sensor error
THEN
     LV_ERR_LS_UP[i] = 1
ELSE
     LV_ERR_LS_UP[i] = 0
ENDIF
```

The general end flag is created according to the following logic:

If the corresponding bit is one, the LV_END_DIAG_XX flag shall be considered for the final end flag.

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit mask variable (example) C_NR_END_DIAG_LSL_UP[i]	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	Not used	Not used	Not used	Not used	LV_END_DIAG_CHG_LS_UP	LV_END_DIAG_AIR_LSL_UP[i]	LV_END_DIAG_LSH_LSL_UP[i]	LV_END_DIAG_PUC_VLD_LS_UP[i]	LV_END_DIAG_SHIFT_AFR_LSL_UP[i]	LV_END_DIAG_SHIFT_AFL_LSL_UP[i]	LV_END_DIAG_DYN_VLD_LS_UP[i]	LV_END_DIAG_OBD_VLD_LSH_UP[i]	LV_END_DIAG_LSL_UP_IF[i]	LV_END_DIAG_OC_LSL_UP[i]	LV_END_DIAG_EL_LSL_UP[i]	LV_END_DIAG_LSH_UP[i]

For this proposal, the number would be 0000011111111111 binary = 2047 decimal. All end flags except the one from the interchanged diagnosis must be set to 1 to also set the final end flag to 1.

The bitmask shall be combined with the single end flags as follows:

If [(C_NR_END_DIAG_LSL_UP[i].bit 0 = 0 **or** LV_END_DIAG_LSH_UP[i] = 1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 1 = 0 **or** LV_END_DIAG_EL_LSL_UP[i] = 1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 2 = 0 **or** LV_END_DIAG_OC_LSL_UP[i] = 1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 3 = 0 **or** LV_END_DIAG_LSL_UP_IF[i] = 1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 4 = 0 **or** LV_END_DIAG_OBD_VLD_LSH_UP[i]= 1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 5 = 0 **or** LV_END_DIAG_DYN_VLD_LS_UP[i] = 1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 6 = 0 **or** LV_END_DIAG_SHIFT_AFL_LSL_UP[i]=1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 7 = 0 **or** LV_END_DIAG_SHIFT_AFR_LSL_UP[i]=1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 8 = 0 **or** LV_END_DIAG_PUC_VLD_LS_UP[i] = 1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 9 = 0 **or** LV_END_DIAG_LSH_LSL_UP[i] = 1)
and
(C_NR_END_DIAG_LSL_UP[i].bit 10 = 0 **or** LV_END_DIAG_AIR_LSL_UP[i] = 1)

```
    and
    (C_NR_END_DIAG_LSL_UP[i].bit 11 = 0 or LV_END_DIAG_CHG_LS_UP = 1)
then
    LV_LS_UP_DIAG_END[i] = 1
else
    LV_LS_UP_DIAG_END[i] = 0
endif
```

16.36.2 Downstream O2 sensor final diagnosis

FUNCTION DESCRIPTION:

General information:

This function shall summarize the diagnostic status of the downstream oxygen sensor delivered from the OBD I & II (heater and sensor) functions through the flag LV_ERR_LS_DOWN[i].

The ultimate error flag LV_ERR_LS_DOWN[i] is a System Error Flag. It should neither be saved in the Error Management nor a DTC code should be given.

To reduce the number of variables to be checked by other diagnosis functions, the collecting flag LV_LS_DOWN_DIAG_END is introduced. It provides the possibility to combine the end flags from almost all the downstream lambda sensor diagnosis functions. By a bit mask variable it is calibratable which end flag contributes to the collecting end flag.

Note:

1. If two separate Banks are concerned, then i=1 for cylinder bank 1 and i=2 for cylinder bank 2.
2. The diagnosis for interchanged lambda sensors is not included in the final error and end flag as this is mainly related to the End of Line test or to a final check after a repair in the garage.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset **or** at clearing error memory:

```
LV_ERR_LS_DOWN[i] = 0
LV_LS_DOWN_OBD_1_MAN_DEAC[i] = 0
LV_LS_DOWN_OBD_2_MAN_DEAC[i] = 0
LV_LS_DOWN_DIAG_END[i] = 0
```

Recurrence:

The function should be carried out once every 100 ms.

Activation / Deactivation:

The diagnosis will be carried out only if LV_ST_END = 1

Formula section:

```
If      LV_LS_DOWN_OBD_MAN_DEAC_EXT = 1
Then    LV_LS_DOWN_OBD_1_MAN_DEAC[i] = 1
        LV_LS_DOWN_OBD_2_MAN_DEAC[i] = 1
Else
    LV_LS_DOWN_OBD_1_MAN_DEAC[i] = LC_LS_DOWN_OBD_1_MAN_DEAC[i]
    LV_LS_DOWN_OBD_2_MAN_DEAC[i] = LC_LS_DOWN_OBD_2_MAN_DEAC[i]
Endif
```

Calculation of sum error bit for downstream oxygen sensor diagnosis

```
IF
    LC_LS_DOWN_DIAG_ACT[i] = 1     % final diagnosis active                  AND
    ( LV_ERR_EL_LS_DOWN[i]= 1      % electrical failure                      OR

    LV_ERR_SWT_LS_DOWN[i] = 1      % switching time rich-to-lean failure     OR
    LV_ERR_PUE_LS_DOWN[i] = 1      % sensor signal after PUC phase failure   OR

    LV_ERR_CHK_LS_DOWN[i] = 1      % error detected by the active test       OR
    LV_ERR_PLAUS_LS_DOWN[i] = 1  % error detected by the passive test
           OR

    LV_ERR_LSH_DOWN[i] = 1         % heater electrical failure               OR
    LV_ERR_OBD_LSH_DOWN[i] = 1   % heater obd2 failure                       OR
    ( LV_ERR_CHG_LS_DOWN = 1 & LC_ERR_CHG_LS_DOWN = 1 )                      OR
    % sensor interchanged failure

    LC_LS_DOWN_ERR[i] = 1 )        % manual activation of sensor error
THEN
     LV_ERR_LS_DOWN[i] = 1
```

```
ELSE
    LV_ERR_LS_DOWN[i] = 0
ENDIF
```

The general end flag is created according to the following logic:

If the corresponding bit is one, the LV_END_DIAG_XX flag shall be considered for the final end flag.

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit mask variable (example) C_NR_END_DIAG_LS_DOWN[i]	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
	Not used	Not used	Not used	Not used	Not used	LV_END_DIAG_CHG_LS_DOWN	LV_END_DIAG_PUE_LS_DOWN[i]	LV_END_DIAG_SWT_LS_DOWN[i]	LV_END_DIAG_CHK_LS_DOWN[i]	LV_END_DIAG_PUC_LS_DOWN[i]	LV_END_DIAG_FL_LS_DOWN[i]	LV_END_DIAG_OC_LS_DOWN[i]	LV_END_DIAG_SCG_LS_DOWN[i]	LV_END_DIAG_SCP_LS_DOWN[i]	LV_END_DIAG_OBD_LSH_DOWN[i]	LV_END_DIAG_LSH_DOWN[i]

For this proposal, the number would be 0000000111011111 binary = 479 decimal.

The bitmask shall be combined with the single end flags as follows:

```
If  [(C_NR_END_DIAG_LS_DOWN[i].bit 0 = 0 or LV_END_DIAG_LSH_DOWN[i] = 1)
    and
    (C_NR_END_DIAG_LS_DOWN[i].bit 1 = 0 or LV_END_DIAG_OBD_LSH_DOWN[i] = 1)
    and
    (C_NR_END_DIAG_LS_DOWN[i].bit 2 = 0 or LV_END_DIAG_SCP_LS_DOWN[i] = 1)
    and
    (C_NR_END_DIAG_LS_DOWN[i].bit 3 = 0 or LV_END_DIAG_SCG_LS_DOWN[i] = 1)
    and
    (C_NR_END_DIAG_LS_DOWN[i].bit 4 = 0 or LV_END_DIAG_OC_LS_DOWN[i] = 1)
    and
    (C_NR_END_DIAG_LS_DOWN[i].bit 5 = 0 or LV_END_DIAG_FL_LS_DOWN[i] = 1)
    and
    (C_NR_END_DIAG_LS_DOWN[i].bit 6 = 0 or LV_END_DIAG_PUC_LS_DOWN[i] = 1)
    and
    (C_NR_END_DIAG_LS_DOWN[i].bit 7 = 0 or LV_END_DIAG_CHK_LS_DOWN[i] = 1)
    and
```

```
        (C_NR_END_DIAG_LS_DOWN[i].bit 8 = 0 or LV_END_DIAG_SWT_LS_DOWN[i] = 1)
        and
        (C_NR_END_DIAG_LS_DOWN[i].bit 9 = 0 or LV_END_DIAG_PUE_LS_DOWN[i] = 1)
        and
        (C_NR_END_DIAG_LS_DOWN[i].bit 10 = 0 or LV_END_DIAG_CHG_LS_DOWN = 1)
    then
        LV_LS_DOWN_DIAG_END[i] = 1
    else
        LV_LS_DOWN_DIAG_END[i] = 0
    endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_ERR_DYN_VLD_LS_UP	1	0...1H	0...1	1	[-]
Allow Sensor Dynamic Fault to be considered as Sensor Failure (stops closed loop AF control)					
C_NR_END_DIAG_LSL_UP[NC_CBK_EX_NR]	1	0...FFFFH	0...65535	1	[-]
Bit mask to select the consideration of a certain end flag for the final end flag					
LC_LS_UP_OBD_1_MAN_DEAC[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Logical calibration data for manual deactivation of all OBD1 diagnosis for upstream oxygen sensor					
LC_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Logical calibration data for manual deactivation of all OBD2 diagnosis for upstream oxygen sensor					
LC_LS_UP_DIAG_ACT[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Activation bit for upstream oxygen sensor final diagnosis					
LC_LS_UP_ERR[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Opportunity for manual setting of LV_ERR_LS_UP[i] = 1					
LC_ERR_CHG_LS_UP	1	0...1H	0...1	1	[-]
Allow Sensor Interchanged Fault to be considered as Sensor Failure (stops closed loop AF control)					
C_NR_END_DIAG_LS_DOWN[NC_CBK_EX_NR]	1	0...FFFFH	0...65535	1	[-]
Bit mask to select the consideration of a certain end flag for the final end flag					
LC_LS_DOWN_DIAG_ACT[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Activation bit for downstream oxygen sensor final diagnosis					
LC_LS_DOWN_OBD_1_MAN_DEAC[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Logical calibration data for manual deactivation of all OBD1 diagnosis for downstream oxygen sensor					
LC_LS_DOWN_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Logical calibration data for manual deactivation of all OBD2 diagnosis for downstream oxygen sensor					
LC_LS_DOWN_ERR[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Opportunity for manual setting of LV_ERR_LS_DOWN[i] = 1					
LC_ERR_CHG_LS_DOWN	1	0...1H	0...1	1	[-]
Allow Sensor Interchanged Fault to be considered as Sensor Failure (in case of 1)					

16.37 Application Incidences of the Oxygen Sensor Diagnosis Management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LS_UP_OBD_MAN_DEAC_EXT	V/O	0...1H	0...1	1	[-]
Upstream WRAF Sensor Diagnosis deactivation for all banks					
LV_LS_DOWN_OBD_MAN_DEAC_EXT	V/O	0...1H	0...1	1	[-]
Downstream Oxygen Sensor Diagnosis deactivation for all banks					

Input data:

LV_VAR_LSH_UP	LV_VAR_LSH_DOWN		

16.37.1 General inhibition of all upstream oxygen sensor diagnosis for systems without lambda-sensors

FUNCTION DESCRIPTION:

General information:

This function serves to be able to deactivate all WRAF Sensor Diagnosis functions with one flag. If the variant "Lambda-sensors" is not learnt then all upstream-diagnosis are inhibited.

Application conditions:

Initialisation after reset:

LV_LS_UP_OBD_MAN_DEAC_EXT **=!** LV_VAR_LSH_UP

Recurrence: The function should be carried out once every 1000 ms.

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

LV_LS_UP_OBD_MAN_DEAC_EXT **=!** LV_VAR_LSH_UP

16.37.2 General inhibition of all downstream oxygen sensor diagnosis for systems without lambda-sensors

FUNCTION DESCRIPTION:

General information:

This function serves to be able to deactivate all Downstream Oxygen Sensor Diagnosis functions with one flag. If the variant "Lambda-sensors" is not learnt then all downstream-diagnosis are inhibited.

Application conditions:

Initialisation after reset:

LV_LS_DOWN_OBD_MAN_DEAC_EXT **=!** LV_VAR_LSH_DOWN

Recurrence:

The function should be carried out once every 1000 ms.

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

LV_LS_DOWN_OBD_MAN_DEAC_EXT **=!** LV_VAR_LSH_DOWN

16.38 WRAF Controller diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounced diagnosis value caused by an offset correction error of the WRAF Sensor controller					
ERR_SYM_OFS_LSL_UP[NC_CBK_EX_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom of offset correction error on the upstream WRAF sensor					
LV_CDN_DIAG_OFS_LSL_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Condtions to run the offset diagnosis of the upstream WRAF sensor controller					
LV_END_DIAG_OFS_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Result of present diagnostis status is valid					
T_DIAG_ACT_OFS[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
time offset adjustment is active					
CTR_OFS_ADJ_CMPL[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
number of completed offset adjustments					
LV_LSL_OFS_ADJ_DIAG_OFS_OLD[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Value of LV_LS_OFS_ADJ_ACT from last recurrence of the WRAF controller offset diagnosis					
LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounced result of pump current controller diagnosis					
ERR_DIAG_CTL_LSL_UP[NC_CBK_EX_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom of pump current controller fault on the upstream WRAF sensor controller					
CDN_DIAG_CTL_LSL_UP[NC_CBK_EX_NR]	V	0...FH	0...15	1	[-]
Condtions to run the pump current controller diagnosis for the WRAF sensor controller					
LV_END_DIAG_CTL_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
End flag of the pump current controller diagnosis for WRAF sensor controller					

Input data:

LV_IGK	LV_VLS_OFS_ADJ_CMPL[NC_CBK_EX_NR]	LV_VLS_OFS_LIM_LSL_L_GAIN[NC_CBK_EX_NR]	LV_VLS_OFS_LIM_LSL[NC_CBK_EX_NR]
NC_CBK_EX_NR	LV_LSL_OFS_ADJ_ACT[NC_CBK_EX_NR]	LV_INH_DIAG_OFS_LSL[NC_CBK_EX_NR]	STATE_ERR_IPLSL[NC_CBK_EX_NR]
LV_VPLSL_LIM[NC_CBK_EX_NR]	LV_VNLSL_LIM[NC_CBK_EX_NR]	LV_IPLSL_CTL_ENA_LSL_IF[NC_CBK_EX_NR]	LV_ICPLSL_ENA_LSL_IF[NC_CBK_EX_NR]
CDN_DIAG_CTL_LSL_INH[NC_CBK_EX_NR]	CDN_DIAG_CTL_LSL_DI[NC_CBK_EX_NR]	TTIP_MES_LS_UP[NC_CBK_EX_NR]	VLS_UP_DIAG[NC_CBK_EX_NR]
LV_IPLSL_CTL_ENA_PLS_ACT[NC_CBK_EX_NR]	LAMB_SP[NC_CBK_EX_NR]	LV_TTIP_MES_VLD_LS_UP[NC_CBK_EX_NR]	

Import actions:

ACTION_ERRM_FilterSymptom(IN<IDX_DIAG>, IN< LV_CDN_DIAG_XX >, IN< ERR_SYM_XX >, IN< C_ABC_INC_XX >, IN< C_ABC_DEC_XX >, IN< C_ABC_MAX_XX >, OUT< LV_ERR_XX >)
ACTION_ERRM_GetLvErr(IN<IDX_DIAG>, OUT< LV_ERR_XX >)
ACTION_ERRM_GetLvEndDiag(IN<IDX_DIAG>, OUT< LV_END_DIAG_XX >)
ACTION_ERRM_FilterMulticonditio(IN<XX>, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>, OUT<LV_ERR_XX>)

FUNCTION DESCRIPTION:

Application conditions:

Initialisation:

At reset and LV_IGK 0 → 1 transition, do the following synchronization with ERRM:

ACTION_ERRM_GetLvErr (in OFS_LSL_UP_i,
out LV_ERR_OFS_LSL_UP[i])

ACTION_ERRM_GetLvEndDiag (in OFS_LSL_UP_i,
out LV_END_DIAG_OFS_LSL_UP[i])

ACTION_ERRM_GetLvErr (in CTL_LSL_UP_i,
out LV_ERR_CTL_LSL_UP[i])

ACTION_ERRM_GetLvEndDiag (in CTL_LSL_UP_i,
out LV_END_DIAG_CTL_LSL_UP[i])

When clearing the error memory, all ERRM related data shall be reset to 0.

Additionally, the following initialization shall be done on all these occasions:

CDN_DIAG_CTL_LSL_UP[i] = 0

ERR_DIAG_CTL_LSL_UP[i] = 0

CTR_OFS_ADJ_CMPL[i] = 0

T_DIAG_ACT_OFS[i] = 0

LV_LSL_OFS_ADJ_DIAG_OFS_OLD[i] = 0

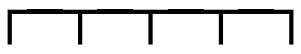
For activation conditions and recurrences see the individual function blocks.

16.38.1 WRAF Controller Offset Diagnosis

General information:

This part of the function shall check the performance of the offset adjustment carried out by the WRAF controller. The measured offset should fall within a calibrateable band, and the time taken by an offset adjustment should be shorter than a calibrateable upper limit.

Error Symptoms are defined as follows:



```
  └─┴─┴─┴─┘
   │  │  │  └─> Limitation of offset correction in low gain         ( = SYM_0)
   │  │  └────> Limitation of offset correction in high gain        ( = SYM_1)
   │  └───────> Exceeded maximum duration of offset correction      ( = SYM_2)
   └──────────> (not used)
```

Recurrence:

The function shall be carried out once every 10ms.

Activation: LV_INH_DIAG_OFS_LSL[i] = 0 **and**
(LV_LSL_OFS_ADJ_DIAG_OFS_OLD[i] = 1 **or**
LV_LSL_OFS_ADJ_ACT[i] = 1)

Formula section:

```
If(1)      LV_LSL_OFS_ADJ_ACT[i] = 1
then(1)    Increment  T_DIAG_ACT_OFS[i]
else(1)    Reset  T_DIAG_ACT_OFS[i]
endif(1)
LV_LSL_OFS_ADJ_DIAG_OFS_OLD[i] = LV_LSL_OFS_ADJ_ACT[i]
```

Setting the symptoms:

```
if (1)    LV_VLS_OFS_ADJ_CMPL[i]  0 -> 1
then (1)     LV_CDN_DIAG_OFS_LSL_UP[i] = 1
          Reset T_DIAG_ACT_OFS[i]
          Increment CTR_OFS_ADJ_CMPL[i]
          if (2)  LV_VLS_OFS_LIM_LSL_L_GAIN[i] = 1
          then (2) ERR_SYM_OFS_LSL_UP[i] = SYM_0    % "offset correction limitation low gain"
          elseif (2) LV_VLS_OFS_LIM_LSL[i] = 1
          then (2) ERR_SYM_OFS_LSL_UP[i] = SYM_1  % "offset correction limitation high gain"
          else (2)  ERR_SYM_OFS_LSL_UP[i] = NO_SYM  % "offset correction works normal"
          endif (2)
elseif (1)   T_DIAG_ACT_OFS[i] >= C_T_DIAG_ACT_OFS_MAX
then (1)     LV_CDN_DIAG_OFS_LSL_UP[i] = 1
          ERR_SYM_OFS_LSL_UP[i] = SYM_2    % "offset correction maximum duration exceeded"
else (1)LV_CDN_DIAG_OFS_LSL_UP[i] = 0
endif (1)
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Calling the generic error management for anti-bounce filtering and update of the outputs:

```
ACTION_ERRM_FilterSymptom (      in     OFS_LSL_UP_i,
                                 in     LV_CDN_DIAG_OFS_LSL_UP[i],
                                 in     ERR_SYM_OFS_LSL_UP[i],
                                 in     C_ABC_INC_DIAG_OFS_LSL_UP[i],
                                 in     1,
                                 in     C_ABC_MAX_DIAG_OFS_LSL_UP[i],
                                 out    LV_ERR_OFS_LSL_UP[i])
ACTION_ERRM_GetLvEndDiag (       in     OFS_LSL_UP_i,
                                 out    LV_END_DIAG_OFS_LSL_UP[i])
```

16.38.2 WRAF Controller Limitation Diagnosis

General information:

This part of the function shall check the performance of the pump current controller and (if used) the reference pump current generator in the WRAF sensor controller. For this, the pump current, pump voltage and nernst overvoltage diagnostic flags read out from the controller by the BSW are checked.

Since the flags are also set if there is an open circuit in one or more of the sensor lines, the diagnosis must only operate under conditions that exclude an open circuit. Hence the checks for tip temperature and VLS_UP_DIAG limits.

To disable monitoring for the individual flags based on driving conditions, the corresponding bit in CDN_DIAG_CTL_LSL_INH[i] should be set. To turn off a monitor permanently, the bit in CDN_DIAG_CTL_LSL_DI[i] must be set instead, otherwise the diagnosis end bit will never be set.

Error Symptoms are defined as follows:

```
 |_|_|_|_|
  | | | └─> Limitation of pump current                             ( = SYM_0)
  | | └───> Limitation of pump voltage                             ( = SYM_1)
  | └─────> VN voltage too high for reference pump current generation  ( = SYM_2)
  └───────> (not used)
```

Recurrence:

The function shall be carried out once every 10ms.

Activation: LV_IPLSL_CTL_ENA_LSL_IF[i] = 1 **and**
(CDN_DIAG_CTL_LSL_INH[i].bit_x = 0 and
CDN_DIAG_CTL_LSL_DI[i].bit_x = 0) for any x = 0...2

Formula section:

If LV_IPLSL_CTL_ENA_PLS_ACT[i] = 0

```
and   C_LAMB_SP_MIN_DIAG_CTL_LSL_UP <= LAMB_SP[i]
and   LAMB_SP[i] <= C_LAMB_SP_MAX_DIAG_CTL_LSL_UP

and   LV_TTIP_MES_VLD_LS_UP[i]
and   TTIP_MES_LS_UP[i] > C_TTIP_MIN_DIAG_CTL_LSL_UP
and   C_TTIP_MAX_DIAG_CTL_LSL_UP > TTIP_MES_LS_UP[i]

and   (VLS_UP_DIAG[i] < C_VLS_UP_MIN_DIAG_CTL_LSL_UP
or      C_VLS_UP_MAX_DIAG_CTL_LSL_UP < VLS_UP_DIAG[i])
then (1)
% Exclude conditions that may indicate open circuit faults to avoid false CTL_LSL_UP entries.
    If      CDN_DIAG_CTL_LSL_DI[i].bit_0 = 0
    and     STATE_ERR_IPLSL [i] != NO_SYM
    then (2a)
            Set "SYM_0" in ERR_DIAG_CTL_LSL_UP[i]
            %pump current above limits allowed by hardware
    Else (2a)
            Clear "SYM_0" in ERR_DIAG_CTL_LSL_UP[i]
            %pump current OK
    Endif (2a)

    If      CDN_DIAG_CTL_LSL_DI[i].bit_1 = 0
    and     LV_VPLSL_LIM[i] = 1
    then (2b)
            Set "SYM_1 in ERR_DIAG_CTL_LSL_UP[i]"
            %pump voltage above limits allowed by hardware
    Else (2b)
            Clear "SYM_1" in ERR_DIAG_CTL_LSL_UP[i]
            %pump voltage OK
    Endif (2b)

    If      CDN_DIAG_CTL_LSL_DI[i].bit_2 = 0
    and     LV_VNLSL_LIM[i] = 1
    and     LV_ICPLSL_ENA_LSL_IF[i] = 1
    then (2c)
            Set "SYM_2" in ERR_DIAG_CTL_LSL_UP[i]
            %voltage at VN too high for correct generation of reference pump current
    Else (2c)
            Clear "SYM_2" in ERR_DIAG_CTL_LSL_UP[i]
            %voltage at VN OK
    Endif (2c)

    % now generate the conditions for the ERRM multi-condition filter call.  Bits 0, 1 and 2 in
    % CDN_DIAG_CTL_LSL_DI/INH[i] correspond to the condition for SYM_0, SYM_1
    % and SYM_2 respectively, as denoted by the suffix _x below.

    If      CDN_DIAG_CTL_LSL_DI[i].bit_x = 1
    or      CDN_DIAG_CTL_LSL_INH[i].bit_x = 0
    then (2d)
            Diagnostic conditions for SYM_x are fulfilled
    Else (2d)
            Diagnostic conditions for SYM_x are not fulfilled
    Endif (2d)

Else (1)
%no diagnostic result is available
```

Diagnostic conditions for SYM_0, SYM_1 and SYM_2 not fulfilled

Endif (1)

Calling the generic error management for anti-bounce filtering and update of the outputs:

```
ACTION_ERRM_FilterMulticonditio (   in    CTL_LSL_UP_i,
                                    in    CDN_DIAG_CTL_LSL_UP[i],
                                    in    ERR_DIAG_CTL_LSL_UP[i],
                                    in    C_ABC_INC_DIAG_CTL_LSL_UP[i],
                                    in    C_ABC_MAX_DIAG_CTL_LSL_UP[i],
                                    out   LV_ERR_CTL_LSL_UP[i])
ACTION_ERRM_GetLvEndDiag (          in    CTL_LSL_UP_i,
                                    out   LV_END_DIAG_CTL_LSL_UP[i])
```

Calibration data:

Name	**Dim**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
C_T_DIAG_ACT_OFS_MAX	1	0...FFFFH	0...655.35	0.01	[s]
Maximum allowed duration of offset adjustment					
C_LAMB_SP_MIN_DIAG_CTL_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Minimum lambda setpoint to execute pump current controller diagnosis					
C_LAMB_SP_MAX_DIAG_CTL_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Maximum lambda setpoint to execute pump current controller diagnosis					
C_TTIP_MIN_DIAG_CTL_LSL_UP	1	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Minimum sensor tip temperature to allow pump current controller diagnosis					
C_TTIP_MAX_DIAG_CTL_LSL_UP	1	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Maximum sensor tip temperature to allow pump current controller diagnosis					
C_VLS_UP_MIN_DIAG_CTL_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Lower end of blind spot for pump current controller diagnosis					
C_VLS_UP_MAX_DIAG_CTL_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Upper end of blind spot for pump current controller diagnosis					
C_ABC_INC_DIAG_OFS_LSL_UP[NC_CBK_EX_NR]	1	0...FFH	0...255	1	[-]
Antibounce counter increment of offset adjustment diagnosis					
C_ABC_MAX_DIAG_OFS_LSL_UP[NC_CBK_EX_NR]	1	1...FFH	1...255	1	[-]
Maximumvalue for antibounce counter of offset adjustment diagnosis					
C_ABC_INC_DIAG_CTL_LSL_UP[NC_CBK_EX_NR]	1	0...FFH	0...255	1	[-]
Antibounce counter increment of pump current controller diagnosis					
C_ABC_MAX_DIAG_CTL_LSL_UP[NC_CBK_EX_NR]	1	1...FFH	1...255	1	[-]
Maximumvalue for antibounce counter of pump current controller diagnosis					

Configuration for diagnostic symptoms :

Diagnostic OFS_LSL_UP[i]	Symptom description	Symptom	Filter type
Diagnostic description	limitation of offset correction in low gain	SYM_0	STD_INI
	limitation of offset in high gain	SYM_1	
	maximum duration of offset correction exeeded	SYM_2	

Diagnostic CTL_LSL_UP[i]	Symptom description	Symptom	Filter type
Diagnostic description	Pump current limitation	SYM_0	MPL_STD_INI
	Pump voltage limitation	SYM_1	
	Reference pump current cutoff because of VN overvoltage	SYM_2	

16.39 O2 sensor (lin, up) controller offset diag (Appl. Inc.)

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_OFS_LSL[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibition condition for the Offset adjustment diagnosis					
LV_INH_DIAG_CTL_LSL_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Inhibition condition for the pump current controller diagnosis					
CDN_DIAG_CTL_LSL_DI[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Permanently disabled monitors for the pump current controller diagnosis					
CDN_DIAG_CTL_LSL_INH[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Temporarily inhibited monitors for the pump current controller diagnosis					

Input data:

LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	LV_VB_CDN_OBD_2	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]
LV_ERR_LSH_UP[NC_CBK_EX_NR]			

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_OFS_LSL allows to deactivate the corresponding diagnostic.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

Description:

Application conditions:

Initialisation:
The following data shall be set to 0 at LV_IGK 0->1 or at ECU reset or at clearing error memory:
LV_INH_DIAG_OFS_LSL[i] = 0
LV_INH_DIAG_CTL_LSL_UP[i] = 0
CDN_DIAG_CTL_LSL_DI[i] = C_CDN_DIAG_CTL_LSL_DI
CDN_DIAG_CTL_LSL_INH[i] = 0

The following data shall be set to 1 at LV_IGK 1->0:
LV_INH_DIAG_OFS_LSL[i] = 1

LV_INH_DIAG_CTL_LSL_UP[i] = 1

Recurrence: 10ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

```
IF      LV_LS_UP_OBD_2_MAN_DEAC[i] = 1     OR
        LV_VB_CDN_OBD_2 = 0                OR
        LV_ERR_LSL_UP_IF[i] = 1            OR
        LV_ERR_EL_LSL_UP[i] = 1
THEN    LV_INH_DIAG_OFS_LSL[i] = 1
ELSE    LV_INH_DIAG_OFS_LSL[i] = 0
ENDIF
```

```
If  LV_LS_UP_OBD_2_MAN_DEAC[i] = 1     OR
    LV_VB_CDN_OBD_2 = 0                OR
    LV_ERR_EL_LSL_UP[i] = 1            OR
    LV_ERR_LSL_UP_IF[i] = 1            OR
    LV_ERR_OFS_LSL_UP[i] = 1           OR
    LV_ERR_OBD_VLD_LSH_UP[i] = 1       OR
    LV_ERR_OC_LSL_UP[i] = 1            OR
    LV_ERR_LSH_UP[i] = 1
then
    LV_INH_DIAG_CTL_LSL_UP[i] = 1
Else
    LV_INH_DIAG_CTL_LSL_UP[i] = 0
Edif
```

```
If  LV_INH_DIAG_CTL_LSL_UP[i] = 1
then
    % inhibition may be changed dynamically here.
    % Disable flags should never be changed, since this would fake a premature end of the diagnosis!
    CDN_DIAG_CTL_LSL_INH[i] = $00000111_{bin}$
ELSE
    CDN_DIAG_CTL_LSL_INH[i] = 0
Endif
```

Configuration for diagnostic symptoms :

Diagnostic XX	Symptoms	Ref.	DTC Symptom SAE	DTC Symptom Customer	DTC Global SAE	DTC Global Customer	Failure class A/B	Readiness code
Diagnostic description	1st symptom description	*SYM_0*			P2A00 P2A01			CARB_LS
	2nd symptom description	*SYM_1*						
	3rd symptom description	*SYM_2*						

Diagnostic XX	Symptoms	Ref.	DTC Symptom SAE	DTC Symptom Customer	DTC Global SAE	DTC Global Customer	Failure class A/B	Readiness code
Diagnostic description	1st symptom description	*SYM_0*			P21xx P21xx			CARB_LS
	2nd symptom description	*SYM_1*						
	3rd symptom description	*SYM_2*						

Fields information (For more information refer to Error Management file “Table of Failure”) :

- DTC Symptom : give for each symptom the corresponding DTC. According to the configuration one or two fields are possible (In our table one is for CARB the other for customer)
- DTC Global : recommended DTC value, it may be the code defined by the law. According to the configuration two fields are possible (In our table one is for CARB the other for customer)
- Failure class parameter for calibrating the behaviour of the Error management for this failure. The failure class values are described in Error management aggregate.
- Readiness code : put the corresponding readiness group code.

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_CDN_DIAG_CTL_LSL_DI	1	0...FFH	0...255	1	[-]
Constant for permanently disable monitors for the pump current controller diagnosis					

16.40 Diagnosis of Heater and Sensor Signals Coupling (ATIC42 / UEGO Sensor)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_LSH_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounced heater coupling failure					
ERR_SYM_LSH_LSL_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom of the heater coupling diagnosis					
LV_CDN_DIAG_LSH_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Heater coupling diagnosis condition					
LV_END_DIAG_LSH_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating a valid diagnostic is available					
CYCNR_DIAG_LSH_LSL_UP[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter of forced stimulation periods investigated					
CTR_SYM_DIAG_LSH_LSL_UP[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter of forced stimulation periods with a present symptom					
MAF_INT_DIAG_LSH_LSL_UP	V	0...FFFFH	0...1820.41666	0.0277778	[g]
Integral of the air flow, for the activation of the heater coupling diagnosis					
LAMB_DELTA_RISE_LPF[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993	6.10E-05	[-]
Mean value for heater coupling (rising edge)					
LAMB_DELTA_FALL_LPF[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993	6.10E-05	[-]
Mean value for heater coupling (falling edge)					
LAMB_SYN_LSHPWM_RISE[NC_CBK_EX_NR]	V	0...7FFFH	0...1.99993	6.10E-05	[-]
Lambda value acquied after lshpwm signal rising edge					
LAMB_SYN_LSHPWM_FALL[NC_CBK_EX_NR]	V	0...7FFFH	0...1.99993	6.10E-05	[-]
Lambda value acquied after lshpwm signal falling edge					
LAMB_DELTA_RISE[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993	6.10E-05	[-]
Variation of lambda value before and after lshpwm signal rising edge					
LAMB_DELTA_FALL[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993	6.10E-05	[-]
Variation of lambda value before and after lshpwm signal falling edge					
T_DLY_TRIG_LSH_UP[NC_CBK_EX_NR]	V/O	0...FFH	0...0.255	0.001	[s]
Time delay for acquisition of oxygen sensor signal and lambda model					

Input data:

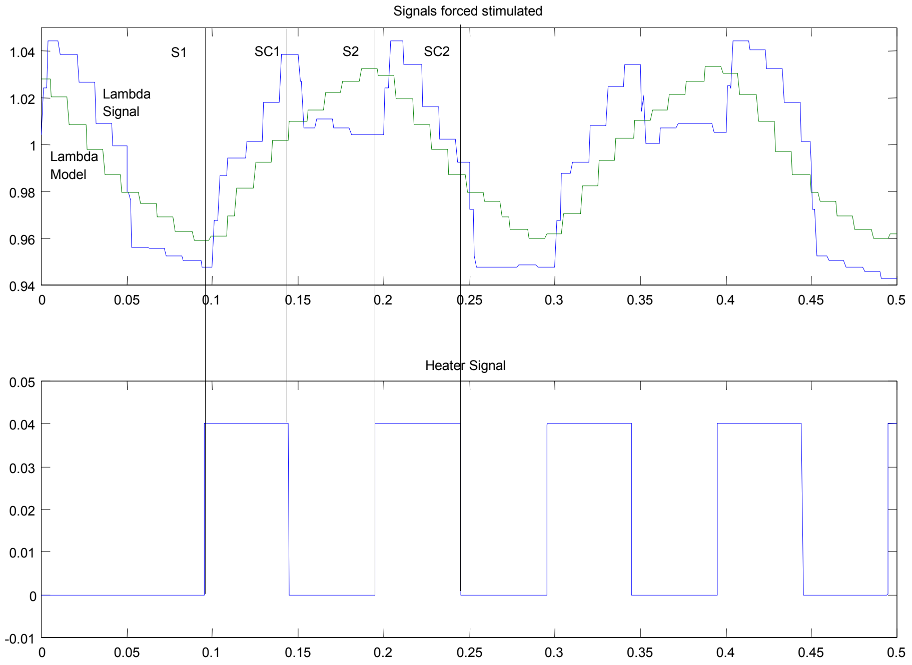
LV_LS_UP_READY[NC_CBK_EX_NR]	LV_LAMB_PLS_ACT[NC_CBK_EX_NR]	LAMB_SP_DE_PLS	LV_PUC
LV_LDC_LAM_ADJ[NC_CBK_EX_NR]	TEG_DYN_LS_UP[NC_CBK_EX_NR]	LSHPWM_UP[NC_CBK_EX_NR]	MAF_KGH
LAMB_SP_FIL_DELTA_FALL[NC_CBK_EX_NR]	LAMB_SP_FIL_DELTA_RISE[NC_CBK_EX_NR]	VLS_UP_DELTA_RISE[NC_CBK_EX_NR]	VLS_UP_DELTA_FALL[NC_CBK_EX_NR]
T_SUM_AFL_AFR_CYC[NC_CBK_EX_NR]	LV_LAM_STOP[NC_CBK_EX_NR]	LV_INH_DIAG_LSH_LSL_UP[NC_CBK_EX_NR]	IP_LAMB_LS_UP
NC_CBK_EX_NR	LV_IGK	LV_LSL_DIAG_LSH_LSL_UP_DEAC	

FUNCTION DESCRIPTION:

General information:

The diagnosis of heater coupling on the sensor signal shall detect an increase in the leakage current from heater to sensor circuit, which can be originated by the reduction of insulation resistance, and causes a crosstalk from the heater signal to the sensor circuit. Due to the heater coupling the measured signal of the wide range A/F sensor is perturbed and the lambda regulation influenced. The diagnosis criterion is the difference in the slope between measured sensor signal and model synchronised with the rising and falling edges of the heater PWM-signal. This is showed in the following picture, where a heater frequence of 10 Hz is admitted.

Sx … sampling of the Lambda value
SCx … sampling of the Lambda value and handover from Basic software



The values LAMB_SP_FIL_DELTA_FALL[i], LAMB_SP_FIL_DELTA_RISE[i], VLS_UP_DELTA_RISE[i] and VLS_UP_DELTA_FALL[i] are delivered from the BASIC-Software and represent the lambda model and the oxygen sensor voltage values acquired synchronised with the rising and falling edges of the heater PWM-signal.

Description:

At the beginning of every diagnosis period, the counter CTR_SYM_DIAG_LSH_LSL_UP[i] and the cycle counter CYCNR_DIAG_LSH_LSL_UP[i] shall be initialised by „0“.

After the hand over of VLS_UP_DELTA_RISE[i], VLS_UP_DELTA_FALL[i], LAMB_SP_FIL_DELTA_RISE[i] and LAMB_SP_FIL_DELTA_FALL[i] from the Basic-software (at the time SCx) this function has to start with the calculation of LAMB_DELTA_RISE/FALL[i].

For a better separation of the forced stimulation and the heater coupling (the forced stimulation has the same symptom as the heater coupling), it is necessary to include the sensor signal model in the diagnosis calculation. This will be realised by a subtraction from the relative deviation of the sensor signal model ($(\lambda_{SC1} - \lambda_{S1})_{model}$) and the relative deviation of sensor signal ($(\lambda_{SC1} - \lambda_{S1})_{meas}$). This results gets absolute value. Then it will be filtered by a first order lag element (for mean value). This values will be named to LAMB_DELTA_RISE_LPF[i] (calculated at time point S1 and SC1) and LAMB_DELTA_FALL_LPF[i] ($(\lambda_{SC1} - \lambda_{S2})_{model}$- $(\lambda_{SC1} - \lambda_{S2})_{meas}$).

This values will be compared with the threshold C_LAMB_DELTA_LPF_THD. Is the calculated value higher than the threshold, the error-counter (CTR_SYM_DIAG_LSH_LSL_UP[i]) will be incremented by “1”.

After a resumption of the diagnosis it must be guaranteed that LAMB_DELTA_FALL_LPF[i] contains a current value. If LAMB_DELTA_FALL_LPF[i] and LAMB_DELTA_RISE_LPF[i] are measured, then the cycle counter CYCNR_DIAG_LSH_LSL_UP[i] shall be incremented. If the necessary quantity of diagnosis for one cycle has been carried out, i.e. if CYCNR_DIAG_LSH_LSL_UP[i] > C_CYCNR_MAX_DIAG_LSH_LSL_UP is valid, the diagnosis result must be determined.

By the comparison of the diagnosis results to the limit value, the heater coupling is diagnosed, i.e. if CTR_SYM_DIAG_LSH_LSL_UP[i] > C_CTR_SYM_DIAG_LSH_LSL_UP is true, a heater coupling (ERR_SYM_LSH_LSL_UP[i] = 1) is recognised. If no error is diagnosed, than ERR_SYM_LSH_LSL_UP[i] = 0 is valid. After the reaching of the required number of diagnosis steps C_CYCNR_MAX_DIAG_LSH_LSL_UP, a new diagnosis cycle is started.
If an error could be detected, then the error flag LV_ERR_LSH_LSL_UP[i] = 1 is set, otherwise LV_ERR_LSH_LSL_UP[i] is set to 0.

Remark: The range of lambda sensor heater duty cycle must be limited, through the sampling of the wide range A/F sensor signal with an period of 10 ms. A heater coupling recognitionis possible only in a range of 15% up to 85% of the duty cycle.

At the function reactivation after LV_PUC=1 the MAF-integral shall be calculated. If the MAF-integral exceeds a defined threshold the heater coupling diagnosis starts.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset or at clearing error memory:
CYCNR_DIAG_LSH_LSL_UP[i] = 0
CTR_SYM_DIAG_LSH_LSL_UP[i] = 0
MAF_INT_DIAG_LSH_LSL_UP = C_MAF_INT_DIAG_LSH_LSL_UP
LV_ERR_LSH_LSL_UP[i] = 0
LV_CDN_DIAG_LSH_LSL_UP[i] = 0
LV_END_DIAG_LSH_LSL_UP[i] = 0
ERR_SYM_ LSH_LSL_UP[i] = “NO SYMPTOM”
LV_ERR_ LSH_LSL_UP[i] = 0
LAMB_DELTA_RISE[i] = 0
LAMB_DELTA_FALL[i] = 0
LAMB_DELTA_RISE_LPF[i] = 0
LAMB_DELTA_FALL_LPF[i] = 0
LAMB_SYN_LSHPWM_RISE[i] = 0
LAMB_SYN_LSHPWM_FALL[i] = 0
T_DLY_TRIG_LSH_UP[i] = 0

Recurrence: 20 ms.

Activation:

```
IF LV_PUC = 0
AND LV_LSL_DIAG_LSH_LSL_UP_DEAC = 0
THEN
   IF MAF_INT_DIAG_LSH_LSL_UP < C_MAF_INT_DIAG_LSH_LSL_UP THEN
      calculate MAF-integral
   ELSE
      IF LV_LAM_STOP[i]= 0                                    AND
         LV_LDC_LAM_ADJ[i] = 1                                AND
         LV_LS_UP_READY[i] = 1                                AND
         TEG_DYN_LS_UP[I] > C_TEG_MIN_DIAG_LSH_LSL_UP         AND
         LV_INH_DIAG_LSH_LSL_UP[i]  = 0                       AND

         { (LAMB_SP_DE_PLS < C_LAMB_SP_DE_DIAG_LSH_LSL_UP     &
           T_SUM_AFL_AFR_CYC[i] < C_T_AFL_AFR_DIAG_LSH_LSL_UP)
           OR
           LV_LAMB_PLS_ACT[i] = 0 }                           AND

           C_LSHPWM_MIN_DIAG_LSH_LSL_UP ≤ LSHPWM_UP[i] ≤
                                   C_LSHPWM_MAX_DIAG_LSH_LSL_UP
      THEN
         LV_CDN_DIAG_LSH_LSL_UP[i]  = 1
                  /activate function
      ELSE
         LV_CDN_DIAG_LSH_LSL_UP[i]  = 0
           /deactivate function
           LAMB_DELTA_RISE_LPF[i]   = 0
           LAMB_DELTA_FALL_LPF[i]   = 0
      ENDIF
   ENDIF
ELSE
   MAF_INT_DIAG_LSH_LSL_UP = 0
   LAMB_DELTA_RISE_LPF[i]   = 0
   LAMB_DELTA_FALL_LPF[i]   = 0
ENDIF
```

Formula section:

1. Calculation of the MAF-integral:

Recurrence: The MAF-Integral calculation shall be incremented every T_SAMPLE =100 ms.

$$\text{MAF\_INT\_DIAG\_LSH\_LSL\_UP}_{N} = \text{MAF\_INT\_DIAG\_LSH\_LSL\_UP}_{N-1} + \text{MAF\_KGH} * \text{T\_SAMPLE [ms]} * 1/3600\ [(g*h) / (kg*ms)]$$

2. Monitoring the electrical coupling between heater PWM and sensor signal

Recurrence: Every time the BSW hands over new values, i.e. if one of the conditions below is true:

```
IF(1) LAMB_SP_FIL_DELTA_FALL[i]$_{N}$ ≠ LAMB_SP_FIL_DELTA_FALL[i]$_{N-1}$
OR(1) VLS_UP_DELTA_FALL[i]$_{N}$ ≠ VLS_UP_DELTA_FALL[i]$_{N-1}$
THEN(1)
increment CYCNR_DIAG_LSH_LSL_UP[i]
```

```
% Determination of the lambda value synchronized with the heater PWM
LAMB_SYN_LSHPWM_RISE[i] = IP_LAMB_LS_UP(VLS_UP_DELTA_RISE[i])
LAMB_SYN_LSHPWM_FALL[i] = IP_LAMB_LS_UP(VLS_UP_DELTA_FALL[i])
```

Note:

1) Before the determination of LAMB_SYN_LSHPWM_RISE[i], please adapt the resolution of VLS_UP_DELTA_RISE[i] to the axis of "IP_LAMB_LS_UP". The analogous shall be done to LAMB_SYN_LSHPWM_FALL[i].

2) The Map "IP_LAMB_LS_UP" is used in other functions, i.e. any changes in this map will influence other functions.

3) At the first function call the values of LAMB_DELTA_RISE/FALL should not be calculated because at this time no old values (N-1) are avaliable.

```
LAMB_DELTA_RISE[i]$_{N}$ = (LAMB_SP_FIL_DELTA_FALL[i]$_{N}$ - LAMB_SP_FIL_DELTA_RISE[i]$_{N}$) –
                           (LAMB_SYN_LSHPWM_FALL[i]$_{N}$ - LAMB_SYN_LSHPWM_RISE[i]$_{N}$)

LAMB_DELTA_FALL[i]$_{N}$ = (LAMB_SP_FIL_DELTA_FALL[i]$_{N-1}$ - LAMB_SP_FIL_DELTA_RISE[i]$_{N}$) –
                           (LAMB_SYN_LSHPWM_FALL[i]$_{N-1}$ - LAMB_SYN_LSHPWM_RISE[i]$_{N}$)

LAMB_DELTA_RISE_LPF[i]$_{N}$ = LAMB_DELTA_RISE_LPF[i]$_{N-1}$
                           + (LAMB_DELTA_RISE[i]$_{N-1}$ – LAMB_DELTA_RISE_LPF[i]$_{N-1}$)
                           * C_CRLC_LPF_RISE_DIAG_LSH_LSL_UP

LAMB_DELTA_FALL_LPF[i]$_{N}$ = LAMB_DELTA_FALL_LPF[i]$_{N-1}$
                           + (LAMB_DELTA_FALL[i]$_{N}$ – LAMB_DELTA_FALL_LPF[i]$_{N-1}$)
                           * C_CRLC_LPF_FALL_DIAG_LSH_LSL_UP
```

```
IF abs(LAMB_DELTA_RISE_LPF[i]) OR abs(LAMB_DELTA_FALL_LPF[i]) >
                                                        C_LAMB_DELTA_LPF_THD
THEN
    increment CTR_SYM_DIAG_LSH_LSL_UP[i]
ENDIF

IF CYCNR_DIAG_LSH_LSL_UP[i] > C_CYCNR_MAX_DIAG_LSH_LSL_UP
THEN
```

```
    IF CTR_SYM_DIAG_LSH_LSL_UP[i] > C_CTR_SYM_DIAG_LSH_LSL_UP
    THEN
        ERR_SYM_ LSH_LSL_UP[i] = “Sym_0”
        LV_ERR_LSH_LSL_UP[i] = 1
        LV_END_DIAG_LSH_LSL_UP[i] = 1
    ELSE
        ERR_SYM_LSH_LSL_UP[i]  = “NO SYMPTOM”
        LV_END_DIAG_LSH_LSL_UP[i]= 1
    ENDIF
    CYCNR_DIAG_LSH_LSL_UP[i] = 0
    CTR_SYM_DIAG_LSH_LSL_UP[i] = 0
ENDIF

    % Adjusting the acquisition delay of sensor signal VLS_UP[i]  and the modeled sensor signal
    % LAMB_SP_FIL_HOM[i]

    T_DLY_TRIG_LSH_UP[i]  = IP_T_DLY_TRIG_LSH_UP (LSHPWM_UP[i])

ENDIF(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_DELTA_LPF_THD	1	0...7FFFH	0...1.99993	6.10E-05	[-]
Threshold for the maximum transition of the sensor signal					
C_CYCNR_MAX_DIAG_LSH_LSL_UP	1	0...FFH	0...255	1	[-]
Quantity of valid diagnosis values in one function cycle					
C_CTR_SYM_DIAG_LSH_LSL_UP	1	0...FFH	0...255	1	[-]
Limit value of the diagnosis results to diagnose the heater coupling					
C_LAMB_SP_DE_DIAG_LSH_LSL_UP	1	0...7FH	0...0.12402	9.77E-04	[-]
Maximum value of the forced excitation amplitude to activate the diagnosis					
C_TEG_MIN_DIAG_LSH_LSL_UP	1	0...7FFFH	-273.15 ... 1774.7875	0.0625	[°C]
Minimum value of the exhaust temperature to activate the function					
C_MAF_INT_DIAG_LSH_LSL_UP	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Threshold for the activation of the MAF-Integral					
C_T_AFL_AFR_DIAG_LSH_LSL_UP	1	0...FFFFH	0...655.35	0.01	[s]
Threshold of the MAF-Integral to activate the heater-coupling diagnosis					
C_CRLC_LPF_FALL_DIAG_LSH_LSL_UP	1	0...FFH	0...0.99609	3.91E-03	[-]
Filter constant for Fall					
C_CRLC_LPF_RISE_DIAG_LSH_LSL_UP	1	0...FFH	0...0.99609	3.91E-03	[-]
Filter constant for Rise					
C_LSHPWM_MIN_DIAG_LSH_LSL_UP	1	0...FFH	0...99.60937	0.390625	[%]
minimum heater PWM value allowed to carry out the heater coupling diagnosis					
C_LSHPWM_MAX_DIAG_LSH_LSL_UP	1	0...FFH	0...99.60937	0.390625	[%]
maximum heater PWM value allowed to carry out the heater coupling diagnosis					
IP_T_DLY_TRIG_LSH_UP	4	0...FFH	0...0.255	0.001	[s]
LDP_LSHPWM_UP_IP_T_DLY_ACQ_VLS	4	0...FFH	0...99.60937	0.390625	[%]
Time delay to trigger oxygen sensor signal and lambda model					

Configuration for diagnostic symptoms:

Diagnosis	Symptom	Nr	ABC type
Diagnosis of the heater coupling in the sensor signal	Heater Coupling Failure Symptom	*0*	NO
LSH_LSL_UP[NC_CBK_EX_NR]			

16.41 Application Incidencies of the “Diagnosis of the Heater Coupling in the Sensor”

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_LSH_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "Diagnosis of the Heater Coupling in the Sensor Signal"					
LV_LSL_DIAG_LSH_LSL_UP_DEAC	V/O	0...1H	0...1	1	[-]
Manual deactivation of "Diagnosis of the Heater Coupling in the Sensor Signal¿					

Input data:

LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]
LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]
LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LAM_ADJ[NC_CBK_EX_NR]
LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSH_UP[NC_CBK_EX_NR]
LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_CHG_LS_UP

Application conditions:

Recurrence: 100 ms.
Initialisation: 0, at LV_IGK = 0 to 1 or ECU-Reset:

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Description:

The function “Diagnosis of the Heater Coupling in the Sensor Signal” has to be deactivated, if at least one of the following errors / special modes

Formula section:

```
IF      LV_ERR_LSL_UP_IF[NC_CBK_EX_NR] = 1                OR
        LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR] = 1            OR
        LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR] = 1            OR
        LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]        = 1        OR
        LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR] = 1         OR
        LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR] = 1           OR
        LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR] = 1               OR
```

```
          LV_ERR_EL_LSL_UP[NC_CBK_EX_NR] = 1              OR
          LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]  = 1             OR
          LV_ERR_LAM_ADJ [NC_CBK_EX_NR] = 1               OR
          LV_ERR_LSH_UP[NC_CBK_EX_NR] = 1                 OR
          LV_ERR_CHG_LS_UP

THEN
    LV_INH_DIAG_LSH_LSL_UP[i] = 1
ELSE
    LV_INH_DIAG_LSH_LSL_UP[i] = 0
ENDIF
```

LV_LSL_DIAG_LSH_LSL_UP_DEAC = LC_LSL_DIAG_LSH_LSL_UP_DEAC

Calibration data:

Name	**Dim**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
LC_LSL_DIAG_LSH_LSL_UP_DEAC	1	0...1H	0...1	1	[-]
Logical calibration data for manual deactivation of heater coupling diagnosis					

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
Diagnosis of the heater coupling in the sensor signal	*Symptom detected*	*0*	P0130 / P0150	P2231 / P2234	---		LAMP CARB_M
LSH_LSL_UP[i]							

16.42 Oxygen Sensor Interchanged Signal Diagnosis

16.42.1 Upstream Sensor Banks 1 and 2

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CHG_LS_UP	V/O	0...1H	0...1	1	[-]
Debounced error: upstream sensor of banks 1and 2 interchanged					
LV_CDN_DIAG_CHG_LS_UP	V	0...1H	0...1	1	[-]
Condition to run the diagnosis are fullfiled					
LV_END_DIAG_CHG_LS_UP	V/O	0...1H	0...1	1	[-]
Diagnostic available					
ERR_SYM_CHG_LS_UP	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom					
MAF_INT_SYM_DIAG_CHG_LS_UP	V	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral for detetction of interchanged upstream sensors					
MAF_INT_END_DIAG_CHG_LS_UP	V	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral for end of upstream interchanged sensors detection in no failure case					

Input data:

FAC_LAM_LIM[NC_CBK_EX_NR]	LV_LAM_STOP[NC_CBK_EX_NR]	LV_LAM_LSCL[NC_CBK_EX_NR]	MAF_CYL
LV_IGK	LV_INH_DIAG_CHG_LS_UP	NC_CBK_EX_NR	LV_LAM_STOP_SHO_PER_CDN[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

This function was written for exhaust gas systems having two banks, where a sensor interchange cannot be avoided by hardware measures. In case the upstream sensors were interchanged it is expected that both lambda signals will drift from the set point in different directions and the lambda controllers achieve their maximum.

A symptom can be detected if the lambda controller correction of one bank crosses the min threshold whereas the other one crosses the max threshold.

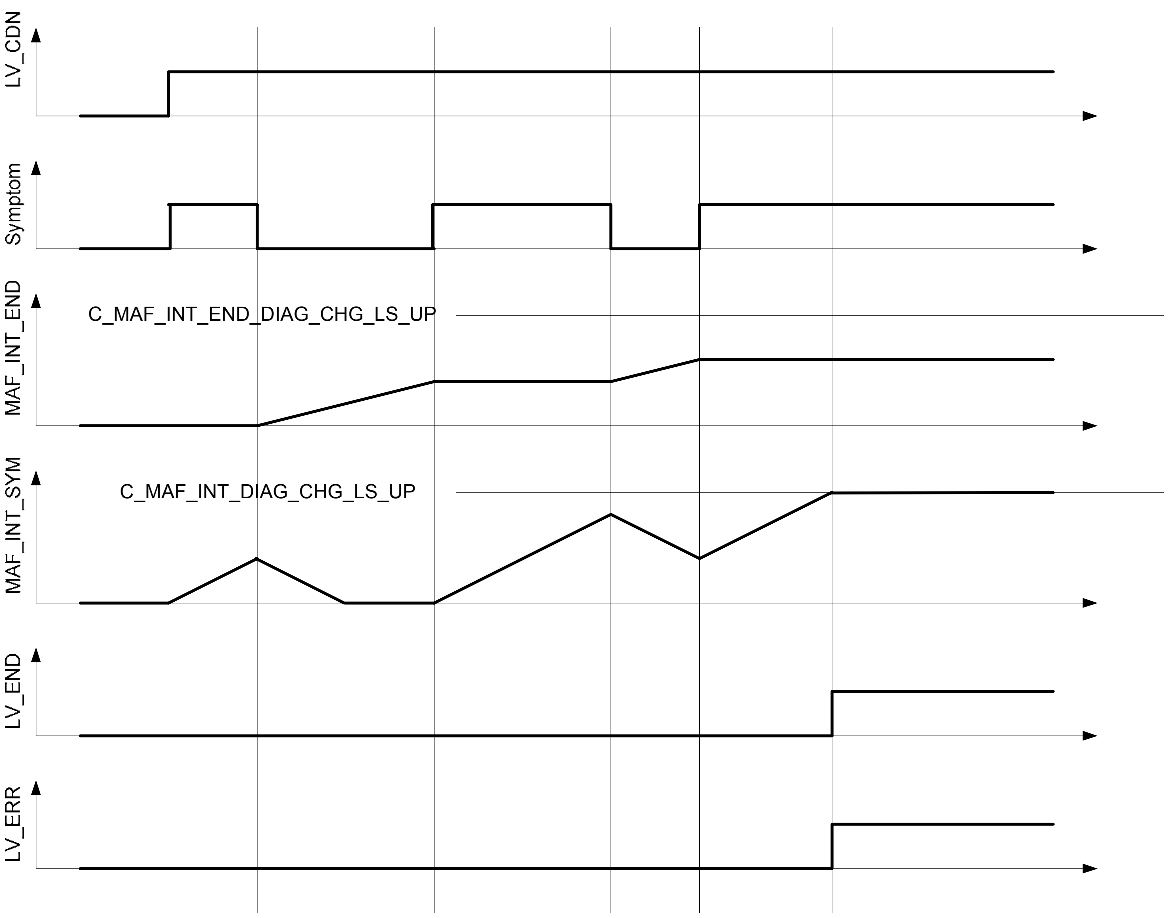


Figure 1: Basic principle of applied debounce algorithm.

Application conditions:

Initialisation:

At the transition LV_IGK = 0 -> 1, reset or clearing error memory reset all variables.

Recurrence:

This function shall run every T_SAMPLE = 100 ms.

Activation/Deactivation:

```
IF      LV_LAM_LSCL[i] = 1        for all [i]
AND         (LV_LAM_STOP[1] = 0          OR LV_LAM_STOP_SHO_PER_CDN[1] = 1)
AND         (LV_LAM_STOP[2] = 0 OR LV_LAM_STOP_SHO_PER_CDN[2] = 1)
AND  LV_INH_DIAG_CHG_LS_UP = 0
AND  LV_END_DIAG_CHG_LS_UP = 0
THEN
     LV_CDN_DIAG_CHG_LS_UP = 1
ELSE
    LV_CDN_DIAG_CHG_LS_UP = 0
ENDIF
```

Formula section:

```
IF(1a)      LV_CDN_DIAG_CHG_LS_UP = 1
THEN(1a)
   IF(2a)
     LV_END_DIAG_CHG_LS_UP = 0 &
          MAF_INT_END_DIAG_CHG_LS_UP < C_MAF_INT_END_DIAG_CHG_LS_UP
   THEN(2a)
         IF(3a) ERR_SYM_CHG_LS_UP = "NO_SYM"
         THEN(3a)
               MAF_INT_END_DIAG_CHG_LS_UP $_n$ =
                    MAF_INT_END_DIAG_CHG_LS_UP $_{n-1}$ + MAF_CYL* T_SAMPLE [ms]   *
               1/3600 [(g*h) / (kg*ms)]
         ELSE(3a)
               MAF_INT_END_DIAG_CHG_LS_UP $_n$ =
                    MAF_INT_END_DIAG_CHG_LS_UP $_{n-1}$
         ENDIF(3a)
   ELSE(2a)
         LV_END_DIAG_CHG_LS_UP = 1
   ENDIF(2a)

   IF(2b)
      (FAC_LAM_LIM[1] >  C_FAC_LAM_LIM_MAX_CHG_LS_UP &
       FAC_LAM_LIM[2] <  C_FAC_LAM_LIM_MIN_CHG_LS_UP)
   OR(2b)
      (FAC_LAM_LIM[1] <  C_FAC_LAM_LIM_MIN_CHG_LS_UP &
       FAC_LAM_LIM[2] >  C_FAC_LAM_LIM_MAX_CHG_LS_UP)
   THEN(2b)
         ERR_SYM_CHG_LS_UP = “SYM_3”
         IF(3b)  MAF_INT_SYM_DIAG_CHG_LS_UP  < C_MAF_INT_DIAG_CHG_LS_UP
         THEN(3b)
               MAF_INT_SYM_DIAG_CHG_LS_UP $_n$ =
                            MAF_INT_SYM_DIAG_CHG_LS_UP $_{n-1}$ +
                    MAF_CYL* C_FAC_POS_CHG_LS_UP *
                    T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]
         ELSE(3b)
               LV_ERR_CHG_LS_UP = 1
               LV_END_DIAG_CHG_LS_UP = 1
         ENDIF(3b)
   ELSE(2b)
     ERR_SYM_CHG_LS_UP = “NO_SYM”
     MAF_INT_SYM_DIAG_CHG_LS_UP $_n$ =
            MAF_INT_SYM_DIAG_CHG_LS_UP $_{n-1}$
            - MAF_CYL* C_FAC_NEG_CHG_LS_UP * T_SAMPLE [ms]
            * 1/3600 [(g*h) / (kg*ms)]
   ENDIF(2b)

ELSE(1a)
   MAF_INT_SYM_DIAG_CHG_LS_UP = 0
   MAF_INT_END_DIAG_CHG_LS_UP = 0
ENDIF(1a)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_LAM_LIM_MIN_CHG_LS_UP	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Min. Threshold to detect a interchanged sensor					
C_FAC_LAM_LIM_MAX_CHG_LS_UP	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Max. Threshold to detect a interchanged sensor					
C_MAF_INT_DIAG_CHG_LS_UP	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Max. MAF integral value when ERR_SYM_... ("NO_SYM"					
C_MAF_INT_END_DIAG_CHG_LS_UP	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Max. MAF integral value to have a valid diagnostic result					
C_FAC_POS_CHG_LS_UP	1	0...FFH	0...1.99218	0.0078125	[-]
Gain factor when MAF integral is increased					
C_FAC_NEG_CHG_LS_UP	1	0...FFH	0...1.99218	0.0078125	[-]
Gain factor when MAF integral decreased					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
CHG_LS_UP	1st symptom description	SYM_3	NO

16.43 Appl. Incidences of “Oxygen Sensor Interchanged Signal Diagnosis”

16.43.1 “Upstream sensor banks 1 and 2”

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_CHG_LS_UP	V/O	0...1H	0...1	1	[-]
Inhibit diagnosis interchanged upstream sensors					
LV_LAM_STOP_SHO_PER_CDN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
substitute to LV_LAM_STOP_SHO_PER in intechanged signal diagnosis					

Input data:

NC_CBK_EX_NR	LV_ERR_EL_CPS	LV_ERR_SA_SYS
NC_CYL_NR		LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]
NC_INJ_CONF	LV_ERR_SAV	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_IGK	LV_ERR_SAP	LV_LAM_STOP_SHO_PER[NC_CBK_EX_NR]
LV_ST_END	LV_ERR_SA_SAV	LV_ERR_DIAGCPS
LV_ERR_SA_SAP	LV_ERR_SA_SAV_LSL	LV_CDN_VB_OBD2
LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_MAP_DIP_SHIFT
LV_ERR_MAF	LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS

General Information:

The calculation shall be done for all exhaust cylinder banks.

Application conditions:

Initialisation:

At the transition LV_IGK = 0 -> 1 or ECU-Reset:

LV_INH_DIAG_CHG_LS_UP = 0
LV_LAM_STOP_SHO_PER_CDN[i] = 0

At the transition LV_IGK = 1 -> 0
LV_INH_DIAG_CHG_LS_UP = 1
LV_LAM_STOP_SHO_PER_CDN[i] = 1

Recurrence:

100 ms

Activation:

LV_IGK = 1

Deactivation

LV_IGK = 0

Formula section:

```
LV_LAM_STOP_SHO_PER_CDN[i] = LV_LAM_STOP_SHO_PER[i]
% prevent frequent deactivation of diagnosis caused by active sensor offset adjustment

IF
          LV_CDN_VB_OBD2 = 0                                  OR
          LV_ERR_DYN_VLD_LS_UP[i] = 1          for any [i]    OR
          LV_ERR_EL_CPS = 1                                   OR
          LV_ERR_DIAGCPS = 1                                  OR
          LV_ERR_SAV = 1                                      OR
          LV_ERR_SAP = 1                                      OR
          LV_ERR_SA_SAV = 1                                   OR
          LV_ERR_SA_SAV_LSL = 1                               OR
          LV_ERR_SA_SYS = 1                                   OR
          LV_ERR_SA_SAP = 1                                   OR
          LV_LS_UP_OBD_2_MAN_DEAC[i] = 1       for any [i]    OR
          LV_ERR_MAP_DIP_SENS = 1                             OR
          LV_ERR_MAP_DIP_PLAUS = 1                            OR
          LV_ERR_MAP_DIP_SHIFT = 1                            OR
          LV_ERR_MAF = 1                                      OR
          LV_ERR_FSD[i] = 1                                   OR
          LV_ERR_MAP_TPS_PLAUS = 1
THEN
     LV_INH_DIAG_CHG_LS_UP = 1
ELSE
     LV_INH_DIAG_CHG_LS_UP = 0
ENDIF
```

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurrence	Failure class A/B
Plausibility check of sensor signal	*Symptom detected*	*0*	P040	---	---		NO_LAMP CARB_M
CHG_LS_UP							

16.44 O2 sensor (lin, up) not mounted diag

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounced diagnosis result of upstream sensor signal monitoring					
LV_CDN_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Diagnosis condition					
ERR_SYM_AIR_LSL_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
diagnosis symptom of upstream sensor signal monitoring					
LV_END_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Result of diagnosis exist					
T_SYM_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer to debounce an error					
T_END_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer to have a diagnostic available					

Input data:

VLS_UP_DIAG[NC_CBK_EX_NR]	LV_IPLSL_VLD[NC_CBK_EX_NR]	LV_PL	LV_FL
LV_INH_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_IGK	LV_DIAG_EOL_END_LS_UP_DOWN[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1, for exhaust cylinder bank 1
 i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1, for single exhaust cylinder bank.

Description:

This function shall detect a mechanically not attached sensor to the exhaust gas pipe. To do this the WRAF sensor output signal must be monitored during the engine operation , when power is generated. This shall be done by comparing the sensor signal VLS_UP_DIAG[i] with a threshold.
The function shall be activated during Partial Load or Full Load, if the sensor was recognised as operable, i.e. LV_IPLSL_VLD[i] = 1. A symptom shall be detected if the engine runs, i.e.

it generates power, and at the same time the sensor signal is very high, indicating a very lean combustion mixture.

Application Conditions:

Initialisation:

At LV_IGK = 0 to 1, reset **or** at clearing error memory reset all variables.

Recurrence:

The function shall be carried out every 100 ms.

Activation:

```
IF
  LV_IPLSL_VLD[i] = 1                    AND
  (LV_PL = 1  OR  LV_FL = 1)                   AND
  LV_INH_DIAG_AIR_LSL_UP[i] = 0
THEN
    LV_CDN_DIAG_AIR_LSL_UP[i]  = 1
    /activate function
ELSE
    LV_CDN_DIAG_AIR_LSL_UP[i]  = 0
    /deactivate function
ENDIF
```

Formula section:

```
IF  VLS_UP_DIAG[i]  > C_VLS_UP_DIAG_MAX_AIR_LSL_UP
THEN
    ERR_SYM_AIR_LSL_UP[i]  = "SYM_0"
    IF T_SYM_DIAG_AIR_LSL_UP[i] >  C_T_SYM_DIAG_AIR_LSL_UP
    THEN
            LV_ERR_AIR_LSL_UP[i]   = 1
            LV_END_DIAG_AIR_LSL_UP[i]     = 1
    ELSE
            Increment T_SYM_DIAG_AIR_LSL_UP[i]
    ENDIF
ELSE
    ERR_SYM_AIR_LSL_UP[i]  = "NO_SYM"
    T_SYM_DIAG_AIR_LSL_UP[i] = 0
ENDIF
```

Minimum time to have a valid diagnostic available

```
IF(a) LV_END_DIAG_AIR_LSL_UP[i]  = 0
THEN(a)
      IF(b)   T_END_DIAG_AIR_LSL_UP[i] > C_T_END_DIAG_AIR_LSL_UP
          OR(b) LV_DIAG_EOL_END_LS_UP_DOWN[i] = 1
```

```
        THEN(b)
            LV_END_DIAG_AIR_LSL_UP[i]  = 1
        ELSE(b)
            Increment T_END_DIAG_AIR_LSL_UP[i]
        ENDIF(b)
ENDIF(a)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VLS_UP_DIAG_MAX_AIR_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Upstream sensor signal max. threshold during engine operation (PUC excepted)					
C_T_SYM_DIAG_AIR_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time filter to detect a symptom					
C_T_END_DIAG_AIR_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time to have a diagnostic available					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
AIR_LSL_UP[NC_CBK_EX_NR]	Sensor on air	SYM_0	NO

16.45 Application Incidences for "O2 sensor (lin, up) not mounted diag"

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_RBM_AIR_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of Rate Based Monitoring of ¿Monitoring Upstream Sensor Signal ¿ by malfunction					
LV_INH_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "Monitoring Upstream Sensor Signal "					
STATE_RBM_AIR_LSL_UP[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Interface of AIR_LSL_UP [NC_CBK_EX_NR] monitor with the Rate-Based Monitoring statistics					
LV_DIAG_EOL_AIR_LSL_UP_DC[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical Value as reminder for End of Line test within this driving cycle					
LV_DIAG_EOL_AIR_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Activation of short trip for upstream sensor signal monitoring					

Input data:

LV_IGK	LV_CDN_VB_OBD2	LV_ERR_MAF	LV_ERR_AMP
LV_ERR_SA_SAV	LV_ERR_AMP_PLAUS	LV_ERR_SA_SYS	LV_ERR_EL_CPS
LV_ERR_SAV	LV_ERR_CHG_LS_UP	LV_MIS_STATE_A	LV_ERR_DIAGCPS
LV_ERR_TPS	LV_DC	LV_MIS_STATE_B	LV_ERR_SAP
LV_ERR_IVVT	NC_CBK_EX_NR	LV_ERR_MAP_DIP_SHIFT	
LV_ERR_SA_SAP	LV_ERR_MAP_DIP_SENS	LV_ERR_SA_SAV_LSL	
LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MAP_DIP_PLAUS	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	LV_ERR_FSD[NC_CBK_EX_NR]
CTR_ERR_DYN_NR	LV_HOM_ACT	LV_ERR_IGC	LV_SCC
	LV_END_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]	STATE_ERR_IV	LV_ERR_FTL_MIN
LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_IN	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS
LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]			

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation: at every transition LV_IGK = 0->1or reset all variables shall be reset to 0.

at every transition LV_IGK = 1->0

LV_INH_DIAG_AIR_LSL_UP[NC_CBK_EX_NR] = 1

Recurrence: the functions shall be carried out once every 1s.

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

This function has to be deactivated, if at least one of the following errors / special modes is active.

IF LV_ERR_MAP_DIP_SENS =1 **OR**
- LV_ERR_MAP_DIP_PLAUS = 1 **OR**
- LV_ERR_MAP_DIP_SHIFT = 1 **OR**
- LV_ERR_MAF = 1 **OR**
- STATE_ERR_IV > 0 **OR**
- LV_ERR_LOAD_TPS_PLAUS = 1 **OR**
- LV_ERR_TPS = 1 **OR**
- LV_ERR_IVVT = 1 **OR**
- LV_ERR_CHG_LS_UP = 1 **OR**
- LV_ERR_EL_CPS = 1 **OR**
- LV_ERR_DIAGCPS = 1 **OR**
- LV_ERR_FSD[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_SA_SAP = 1 **OR**
- LV_ERR_SA_SAV = 1 **OR**
- LV_ERR_SA_SAV_LSL= 1 **OR**
- LV_ERR_SA_SYS = 1 **OR**
- LV_ERR_SAP = 1 **OR**
- LV_ERR_SAV = 1 **OR**
- LV_MIS_STATE_A = 1 **OR**
- LV_MIS_STATE_B = 1 **OR**
- LV_ERR_AMP = 1 **OR**
- LV_ERR_AMP_PLAUS = 1 **OR**
- LV_ERR_IGC = 1 **OR**
- LV_ERR_FTL_MIN = 1 **OR**
- LV_ERR_FUP = 1 **OR**
- LV_ERR_FUP_MFP_PLAUS = 1 **OR**
- LV_ERR_H_PRS_SYS = 1 **OR**
- LV_ERR_VCV = 1 **OR**
- LV_ERR_FUP_ORNG = 1 **OR**
- LV_ERR_FUP_ST = 1 **OR**
- LV_ERR_TTIP_MES_LSH_UP[i] = 1 **OR**
- LV_ERR_FSD_LAM_LIM[i] = 1 **OR**
- LV_ERR_SLV_IVVT_IN = 1 **OR**
- LV_ERR_OFS_LSL_UP[i] = 1 **OR**
- LV_ERR_CTL_LSL_UP[i] = 1 **OR**
- LV_ERR_MAP_TPS_PLAUS = 1 **OR**
- LV_ERR_EL_LSL_UP[i] = 1 **OR**

```
        -   LV_ERR_OC_LSL_UP[i] = 1                          OR
        -   LV_ERR_LSL_UP_IF[i] = 1                          OR
        -   LV_ERR_LSH_UP[i]  = 1                            OR
        -   LV_ERR_OBD_VLD_LSH_UP[i] = 1                         OR
        -   LV_ERR_PUC_VLD_LS_UP[i] = 1                      OR
        -   LV_ERR_SHIFT_AFL_LSL_UP[i] = 1                   OR
        -   LV_ERR_SHIFT_AFR_LSL_UP[i] = 1                   OR
        -   LV_ERR_DYN_VLD_LS_UP[i] = 1

  THEN     LV_INH_DIAG_RBM_AIR_LSL_UP[NC_CBK_EX_NR]  =  1
  ELSE     LV_INH_DIAG_RBM_AIR_LSL_UP[NC_CBK_EX_NR]  =  0

ENDIF

IF       LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]  = 1          OR
     -   LV_INH_DIAG_RBM_AIR_LSL_UP[NC_CBK_EX_NR] = 1        OR
     -   LV_CDN_VB_OBD2 = 0                                  OR
     -   LV_SCC[NC_CBK_EX_NR] = 1                            OR
     -   LV_HOM_ACT = 0

  THEN     LV_INH_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]  =  1

  ELSE     LV_INH_DIAG_AIR_LSL_UP[NC_CBK_EX_NR]  =  0

ENDIF
```

Data :

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
Monitoring upstream sensor	*Sym_0*	*0*	*see table of failires*	*see table of failires*	---		NO_LAMP CARB_M
AIR_LSL_UP[i]							

16.45.1.1 Interface for Rate – Based – Monitoring AIR_LSL_UP[i]

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the AIR_LSL_UP[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_AIR_LSL_UP[i] data.

Within STATE_RBM_AIR_LSL_UP[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for catalyst diagnosis)

Application conditions:

Initialisation: *at ECU reset:*

bit 0, bit 1 and bit 2 of STATE_RBM_AIR_LSL_UP[i] = 0

LV_DIAG_EOL_AIR_LSL_UP_DC[i] = 0

LV_DC 0 → 1 transition:

bit 0 and bit 1 of STATE_RBM_AIR_LSL_UP[i] = 0

LV_DIAG_EOL_AIR_LSL_UP_DC[i] = 0

on failure memory reset:

bit 1 of STATE_RBM_STATE_RBM_AIR_LSL_UP[i] = 0

LV_DIAG_EOL_AIR_LSL_UP_DC[i] = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 -> 1 transition

bit 2 of STATE_RBM_AIR_LSL_UP[i] = 1

The pending status of the following failures has to be checked only once :

	LV_ERR_AMP	LV_ERR_LOAD_TPS_PLAUS
LV_ERR_MIS[NC_CYL_NR]	LV_ERR_EL_CPS	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
	LV_ERR_DIAGCPS	LV_ERR_FSD[NC_CBK_EX_NR]
LV_ERR_MEC_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_MAF	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_MEC_IVVT_IN[NC_NR_CBK_IVVT]	LV_ERR_SA_SAP	LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_SA_SAV	LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]
	LV_ERR_SA_SYS	
LV_ERR_MAP_DIP_SENS	LV_ERR_SAP	LV_ERR_LSH_UP[NC_CBK_EX_NR]
LV_ERR_MAP_DIP_PLAUS	LV_ERR_SAV	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]
LV_ERR_MAP_DIP_SHIFT	LV_ERR_SA_SAV_LSL	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]
LV_ERR_CHG_LS_UP	LV_ERR_IV[NC_CYL_NR]	LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_AMP_PLAUS		LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_FUP[NC_CBK_HPP_NR]	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]
LV_ERR_VCV[NC_CBK_HPP_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_TPS_2
LV_ERR_TPS_1	LV_ERR_TPS_AD	LV_ERR_TPS_RATIO
LV_ERR_TPS_MAF_1	LV_ERR_TPS_MAF_2	LV_ERR_TPS_ST_CHK_2
LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3	LV_ERR_MTC_DR
LV_ERR_TPS_AD_BOL	LV_ERR_SLV_IVVT_IN	
LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS

```
If(1)       { CPU optimization at LV_DC 0 -> 1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While  bit 1 of STATE_RBM_AIR_LSL_UP[i] = 0 do
                   with each XX failure of the above list  (depending on configuration, see below):
                   ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>, SYNCHRONIZATION<CALL>)

                   If(2)     XX has a pending status
                   Then(2)        bit 1 of STATE_RBM_AIR_LSL_UP[i] = 1
                   Endif(2)
            Endwhile
Else(1)     { the dynamic failure memory is empty }      No action
Endif(1)
```

Every 1 s :

```
If          LV_DIAG_EOL_AIR_LSL_UP[i] = 1
Then        LV_DIAG_EOL_AIR_LSL_UP_DC[i] = 1
Endif


If    bit 0 of STATE_RBM_AIR_LSL_UP[i] = 0
Then
            If      LV_END_DIAG_AIR_LSL_UP[i] = 1        and
                    LV_DIAG_EOL_AIR_LSL_UP_DC[i] = 0
            Then    bit 0 of STATE_RBM_AIR_LSL_UP[i] = 1
            Endif
Endif


If    bit 1 of STATE_RBM_AIR_LSL_UP[i] = 0
Then
            If      LV_INH_DIAG_RBM_AIR_LSL_UP[i] = 1
            Then    bit 1 of STATE_RBM_AIR_LSL_UP[i] = 1
            Endif
Endif
```

16.45.2 Activation of short trip Monitoring AIR_LSL_UP[i]

FUNCTION DESCRIPTION:

General information:

Not yet defined

Application conditions:

Initialisation: LV_DIAG_EOL_AIR_LSL_UP[i] = 0 *(never changed)*

16.46 O2 sensor (lin, up) dynamic diag

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_DYN_SYM_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounce diagnosis value of the sensor dynamics					
LV_DIAG_DYN_CDN_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Diagnosis condition					
LV_DIAG_DYN_END_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Result of diagnosis exist					
LV_LAMB_PLS_REQ_DYN_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Request forced stimulation parameters specific for the WRAF sensor dynamic					
LV_LAM_ADJ_REQ_DYN_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Request dynamic trim controller parameters specific for the WRAF sensor dynamic					
LV_CYCNR_DYN_LSL_UP_VLD[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Number of carried out single valid amplitude calculation					
LAMB_LPF_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda value filtered with a Low Pass Filter					
LAMB_MIN_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Minimum value of the lambda sensor signal (per sample period)					
LAMB_MAX_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Maximum value of the lambda sensor signal (per sample period)					
LAMB_AMPL_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FH	0...0.12402	0.9766e-3	[-]
carried out single sensor signal amplitude measurement					
LAMB_AMPL_MIN_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FH	0...0.12402	0.9766e-3	[-]
Minimum acceptable carried out single sensor signal amplitude measurement					
LAMB_AMPL_MAX_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FH	0...0.12402	0.9766e-3	[-]
Maximum acceptable carried out single sensor signal amplitude measurement					
LAMB_AMPL_INT_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Integral over carried out single sensor signal amplitude measurement					
LAMB_AMPL_MV_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FH	0...0.12402	0.9766e-3	[-]
sensor signal amplitude mean value					
CTR_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Number of carried out single valid amplitude measurements					
CTR_MIN_DIAG_DYN_LSL_UP	V	0...FFH	0...255	1	[-]
Minimum number of carried out single valid amplitude measurements					
CTR_WAIT_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Number of forced stimulation periods before carrying out amplitude measurements					
CYCNR_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
number of cycles necessary to finish the diagnosis					

STATE_DYN_DIAG[NC_CBK_EX_NR]	V	0H 1H 2H 3H	PASSIVE WAIT ACQUISITION END	1	[-]
State shows the condition of the forced stimulation					
FAC_VALUE_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Normalised single sensor signal amplitude					
FAC_SUM_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Summation over all normalised single sensor signal amplitude					
FAC_MV_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	O/V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Mean value of normalised single sensor signal amplitude					
FAC_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	O/V/S	0...400H	0...1	0.9766e-3	[-]
Adaption factor to the modeled sensor time constant					
T_DYN_DIAG_AFL[NC_CBK_EX_NR]	V	0...FFFFH	0...1310.7	0.02	[s]
Timer for maximum amplitude measurement duration after LV_AFL 0 -> 1					
T_DYN_DIAG_AFR[NC_CBK_EX_NR]	V	0...FFFFH	0...1310.7	0.02	[s]
Timer for maximum amplitude measurement duration after LV_AFL 1 -> 0					
LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating the validation of correspondent symptom (transmitted to the error management)					
ERR_SYM_DYN_VLD_LS_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom to be transmitted to the error management					
LV_END_DIAG_DYN_VLD_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating end of correspondent diagnosis					
LV_CDN_DIAG_DYN_VLD_LS_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating condition of correspondent diagnosis					
FAC_DYN_LSL_DIAG_SAE[NC_CBK_EX_NR]	O/V/S	8000...7FFFH	-327.68...327.67	0.01	[-]
Mode6 output of diagnostic result. Either from Dynamic diag or from Dynamic diag after NOx cat regeneration					
FAC_DYN_LSL_DIAG_TOL_SAE[NC_CBK_EX_NR]	O/V/S	8000...7FFFH	-327.68...327.67	0.01	[-]
Mode6 Output : Maximum test limit. Either from Dynamic diag (homogen) or Dynamic diag after Nox-Cat regeneration.					

Input data:

LV_LAMB_PLS_ACT[NC_CBK_EX_NR]	LV_IPLSL_VLD[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]	LV_AFL[NC_CBK_EX_NR]
STATE_CAT_DIAG[NC_CBK_EX_NR]	LV_DIAG_EOL_DYN_LS_UP[NC_CBK_EX_NR]	LV_DIAG_DYN_INH_LSL_UP[NC_CBK_EX_NR]	LV_INH_DIAG_LS_UP_DYN_CP
LV_FAC_LAM_LIM_MAX[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MIN[NC_CBK_EX_NR]	MAF	N_32
MAF_HB	LV_CAT_PURGE_ACT[NC_CBK_EX_NR]	LV_CAT_LDC[NC_CBK_EX_NR]	NC_CBK_EX_NR
LV_DIAG_DYN_S_LSL_UP_END[NC_CBK_EX_NR]	LV_DIAG_DYN_S_LSL_UP_ERR[NC_CBK_EX_NR]	NC_IDX_DIAG_DYN_VLD_LS_UP	FAC_MV_DIAG_DYN_S_LSL_UP[NC_CBK_EX_NR]
C_FAC_DIAG_DYN_S_LSL_UP_MAX			

Import actions:

ACTION_ERRM_NoFilterSymptom(IN<IDX_DIAG>, IN< LV_CDN_DIAG_XX >, IN< ERR_SYM_XX >, IN< LV_ERR_SET_XX >, IN< LV_ERR_RST_XX >, IN< LV_END_DIAG_XX >, OUT< LV_ERR_XX >)
ACTION_ERRM_NoFilterReset(IN<IDX_DIAG>, OUT< LV_ERR_XX >)

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

This function shall evaluate the Linear Oxygen Sensor dynamic behaviour once per driving cycle by measuring the sensor output signal damping during the time, the forced stimulation is active with the special set of parameters for the Dynamic Diagnosis or for the Catalyst Efficiency Diagnosis. During this time the air-fuel ratio is stimulated with a defined amplitude and period.

The air-fuel oscillation observed with a new sensor and with the aged limit sensor shall be measured, using the above mentioned forced stimulation parameters in combination with the eventually shorter period time IP_T_DYN_DIAG. The results shall be saved in the maps IP_LAMB_AMPL_MIN_DYN_LSL_UP (for the aged limit sensor) and IP_LAMB_AMPL_MAX_DYN_LSL_UP (for a new sensor). The aged limit sensor has to be defined by the project in order to fulfil the legal emission requirements and/or to avoid the cross effect on any other function, for example on Catalyst Efficiency Diagnosis.

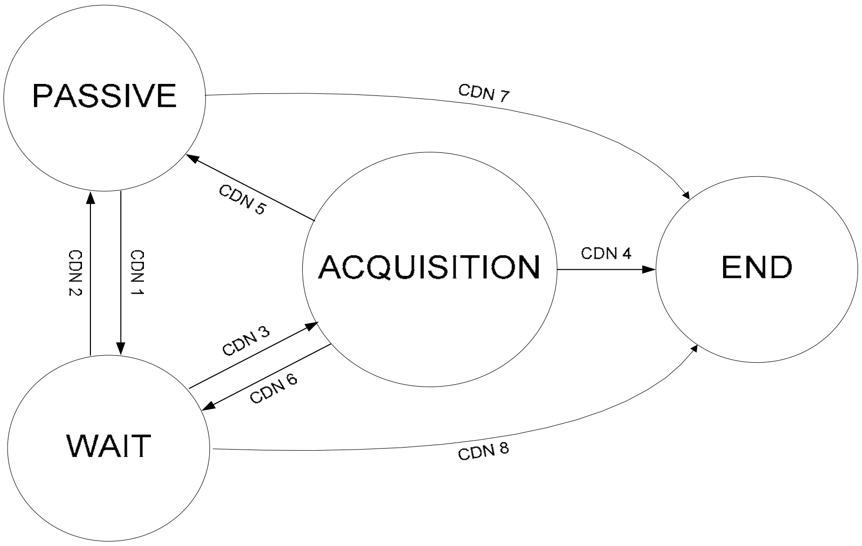
Once the diagnosis is active, after every forced stimulation period, the lambda sensor signal amplitude is calculated. This amplitude is normalised versus engine speed and load. After a calibrateable number of normalised amplitude values are available, a mean value is determined. The lambda amplitude mean value is transformed into the relative damping factor FAC_MV_DIAG_DYN_LSL_UP[i] by comparing with the amplitude of a new sensor with the value for the limit aged sensor.

Once the relative damping factor FAC_MV_DIAG_DYN_LSL_UP[i] is determined for the current driving cycle, it is compared with a calibrateable threshold. If the sensor dynamic response is considered to be over the acceptable limits, then LV_DIAG_DYN_SYM_LSL_UP[i] = 0, otherwise this flag is set to 1.

The factor FAC_MV_DIAG_DYN_LSL_UP[i] can also be used to adapt the sensor time constant model T1_LSL_UP[i] by setting the Boolean switch LC_T1_AD_DIAG_DYN_LSL_UP = 1. In that case a simple linear interpolation is done between the time constant for a new sensor and that for the aged limit sensor, which are saved in the correspondent maps IP_T1_LSL_UP_AGI_LIM[i] and IP_T1_LSL_UP[i]. For more information please refer to the function specification *Lambda control (Fuel mass).*

To help the calibration of this function a tuning guide is available.

Signal flow diagram:



Application conditions:

Initialisation:

At the transition LV_IGK = 0 ->1, by ECU reset, by clearing the Error Memory or by triggering the EOL test (transition LV_DIAG_EOL_DYN_LS_UP[i] = 0 -> 1) iinitialise the variables as indicated below:

- LAMB_MIN_DIAG_DYN_LSL_UP[i] = 32
- T_DYN_DIAG_AFL[i] = 1310.7
- T_DYN_DIAG_AFR[i] = 1310.7
- FAC_DIAG_DYN_LSL_UP[i] , FAC_DYN_LSL_DIAG_SAE[i] and FAC_DYN_LSL_DIAG_TOL_SAE[i] with the correspondent value saved on the non-volatile memory.
- all other variables with zero.

At the initialisation of new ECU or at a EEPROM checksum error, initialize

- FAC_DIAG_DYN_LSL_UP[i] with 0
- FAC_DYN_LSL_DIAG_SAE[i] with 0
- FAC_DYN_LSL_DIAG_TOL_SAE[i] with MAX

ACTION_ERRM_NoFilterReset(IN<NC_IDX_DIAG_ DYN_VLD_LS_UP[i]>, OUT<LV_ERR_ DYN_VLD_LS_UP[i]>)

Recurrence: 20 ms.

Activation:

```
If(1) LV_DIAG_DYN_END_LSL_UP[i] = 0
Then(1)
    If(2) LV_DIAG_DYN_S_LSL_UP_END[i] = 1
    Then(2)
    % force diagnosis to enter result from stratified diagnosis function
        LV_DIAG_DYN_CDN_LSL_UP[i] = 1
```

```
        else(2)
            If(3) LV_DIAG_DYN_INH_LSL_UP[i] = 0
            Then(3)
                If(4) LC_DIAG_DYN_LSL_UP_CAT_SYN = 1
                % diagnosis carried out together with the catalyst efficiency diagnosis
                Then(4)
                    If(5)     STATE_CAT_DIAG[i]  = “passive”
                          OR ( STATE_CAT_DIAG[i] = "END" &
                              CYCNR_DIAG_DYN_LSL_UP[i]
                                  < C_CYCNR_MIN_DIAG_DYN_LSL_UP )
                    Then(5)
                        LV_DIAG_DYN_CDN_LSL_UP[i] = 0
                    Else(5)
                        LV_DIAG_DYN_CDN_LSL_UP[i] = 1
                    Endif(5)
                Else(4)
                % diagnosis carried out after or before the catalyst efficiency diagnosis
                    If(5) LV_IPLSL_VLD[i] = 1                                   &
                        LV_LAMB_PLS_ACT[i] = 1                          &
                        LV_FAC_LAM_LIM_MAX[i] = 0                               &
                        LV_FAC_LAM_LIM_MIN[i] = 0                               &
                        C_N_MIN_DIAG_DYN_LSL_UP < N_32 <
                                            C_N_MAX_DIAG_DYN_LSL_UP     &
                        C_MAF_MIN_DIAG_DYN_LSL_UP < MAF <
                                            C_MAF_MAX_DIAG_DYN_LSL_UP&
                        (STATE_CAT_DIAG[i]  =  “passive”  or  “end”)            &
                        LV_CAT_LDC[i] = 1                               &
                        LV_CAT_PURGE_ACT[i] = 0                                 &
                        % catalyst purge function not active
                        LV_INH_DIAG_LS_UP_DYN_CP = 0
                        % canister purge stable with low load (refer to the Appl. Inc.)
                    Then(5)
                        LV_DIAG_DYN_CDN_LSL_UP[i] = 1
                    Else(5)
                        LV_DIAG_DYN_CDN_LSL_UP[i] = 0
                    Endif(5)
                Endif(4)
            Else(3)
                LV_DIAG_DYN_CDN_LSL_UP[i] = 0
            Endif(3)
        Endif(2)
    Else(1)
        LV_DIAG_DYN_CDN_LSL_UP[i] = 0
    Endif(1)
```

Activation by a short test:

```
IF
  LV_DIAG_EOL_DYN_LS_UP[i] = 1                    &
  LV_IPLSL_VLD[i] = 1                                     &

  LV_LAMB_PLS_ACT[i] = 1                          &
```

```
    LV_DIAG_DYN_INH_LSL_UP[i] = 0                    &
    LV_FAC_LAM_LIM_MAX[i] = 0                        &
    LV_FAC_LAM_LIM_MIN[i] = 0                        &
    [(STATE_CAT_DIAG[i]  = passive  or end) or LC_DIAG_DYN_LSL_UP_CAT_SYN = 1]
                                                     &
    LV_INH_DIAG_LS_UP_DYN_CP = 0
    % canister purge operating stable with low load (refer to the Appl. Inc.)
  THEN
    LV_DIAG_DYN_CDN_LSL_UP[i] = 1
  ELSE
    LV_DIAG_DYN_CDN_LSL_UP[i] = 0
  ENDIF
```

Formula section:

1. PRE-CALCULATIONS

1.1 FILTERING HIGH FREQUENCE PERTUBATIONS

LAMB_LPF_DIAG_DYN_LSL_UP[i] $_{(new)}$ =
LAMB_LPF_DIAG_DYN_LSL_UP[i] $_{(old)}$ *(1-C_CRLC_LPF_ LAMB_DIAG_DYN_LSL_UP) +
C_CRLC_LPF_LAMB_DIAG_DYN_LSL_UP * LAMB_LS_UP[i]

1.2 CALCULATION OF SENSOR SIGNAL MIN and MAX VALUES

```
If          LV_AFL[i] = 0 -> 1
then        T_DYN_DIAG_AFL[i] = IP_T_DYN_DIAG
else        decrement T_DYN_DIAG_AFL[i]
endif

If          T_DYN_DIAG_AFL[i] > 0
And         LAMB_MAX_DIAG_DYN_LSL_UP[i] < LAMB_LPF_DIAG_DYN_LSL_UP[i]
Then        LAMB_MAX_DIAG_DYN_LSL_UP[i] = LAMB_LPF_DIAG_DYN_LSL_UP[i]
            /set maximum Lambda value
else        freeze LAMB_MAX_DIAG_DYN_LSL_UP[i]
endif

If          LV_AFL[i] = 1 -> 0
then        T_DYN_DIAG_AFR[i] = IP_T_DYN_DIAG
else        decrement T_DYN_DIAG_AFR[i]
endif

If          T_DYN_DIAG_AFR[i] > 0
and         LAMB_MIN_DIAG_DYN_LSL_UP[i] > LAMB_LPF_DIAG_DYN_LSL_UP[i]
then        LAMB_MIN_DIAG_DYN_LSL_UP[i] = LAMB_LPF_DIAG_DYN_LSL_UP[i]
            /set minimum Lambda value
else        freeze LAMB_MIN_DIAG_DYN_LSL_UP[i]
endif
```

2. STATE MACHINE

In the state machine, the transitions shall be handled first and the current state be changed if required. Then the state actions of the new current state shall be executed.

2.1 STATE_DYN_DIAG[i] = "passive"

Actions: None

Condition 7: from STATE_DYN_DIAG[i] = "passive" to "end"

LV_DIAG_DYN_S_LSL_UP_END[i] = 1

Transition actions: STATE_DYN_DIAG[i] = "end"

Condition 1: from STATE_DYN_DIAG[i] = "passive" to "wait"

LV_DIAG_DYN_CDN_LSL_UP[i] = 1

Transition actions: STATE_DYN_DIAG[i] = "wait"

2.2 STATE_DYN_DIAG[i] = "wait"

Actions:

```
If    LV_AFL[i] = 0 -> 1 & LC_DIAG_DYN_LSL_UP_CAT_SYN = 0
Then
    If LV_LAMB_PLS_REQ_DYN_LSL_UP[i] = 0
    Then
            LV_LAMB_PLS_REQ_DYN_LSL_UP[i] = 1
            % set the forced stimulation parameters specially for this diagnosis
            LV_LAM_ADJ_REQ_DYN_LSL_UP[i] = 1
            % the current value of the trim controller I-share shall be saved. Moreover its parameters shall be
            % temporarily changed to the one originally defined for the Catalyst Eff. Diag.   After the end of the
            % current diagnosis the trim ctr. I-share shall be re-initialised with its old value.
    Else
            Increment CTR_WAIT_DIAG_DYN_LSL_UP[i]
    Endif
% forced stimulation and trim ctr. I-share parameters set by the Catalyst Eff. Diag
Endif
LV_CYCNR_DYN_LSL_UP_VLD[i] = 0
```

Condition 8: from STATE_DYN_DIAG[i] = "wait" to "end"

LV_DIAG_DYN_S_LSL_UP_END[i] = 1

Transition actions: STATE_DYN_DIAG[i] = "end"

Condition 2: from _STATE_DYN_DIAG[i] = "wait" to "passive"

```
LV_DIAG_DYN_CDN_LSL_UP[i] = 0

Transition actions:   STATE_DYN_DIAG[i] = “passive”
                      LV_LAMB_PLS_REQ_DYN_LSL_UP[i] = 0
                      LV_LAM_ADJ_REQ_DYN_LSL_UP[i] = 0
                      CTR_WAIT_DIAG_DYN_LSL_UP[i] = 0
```

Condition 3: from STATE_DYN_DIAG[i] = “wait” to “acquisition”

```
[LC_DIAG_DYN_LSL_UP_CAT_SYN = 1  &
STATE_CAT_DIAG = “cyc_lean“  or   “cyc_rich” ]
Or
 [LC_DIAG_DYN_LSL_UP_CAT_SYN = 0  &
CTR_WAIT_DIAG_DYN_LSL_UP[i] > C_CTR_WAIT_DIAG_DYN_LSL_UP]
```

Transition actions:

```
If  LV_DIAG_EOL_DYN_LS_UP[i] = 1
Then
  CTR_MIN_DIAG_DYN_LSL_UP = C_CTR_MIN_DIAG_EOL_DYN_LSL_UP
Else
  CTR_MIN_DIAG_DYN_LSL_UP = C_CTR_MIN_DIAG_DYN_LSL_UP
Endif

LAMB_AMPL_INT_DIAG_DYN_LSL_UP[i] = 0
CTR_DIAG_DYN_LSL_UP[i] = 0
STATE_DYN_DIAG[i] = “acquisition”
```

2.3 STATE_DYN_DIAG[i] = “acquisition”

Actions:

```
If(1) LV_AFL[i] = 0 -> 1
% transition of the forced stimulation half period rich to lean
Then(1)
    If(2)   LV_CYCNR_DYN_LSL_UP_VLD[i] = 1 &
            LAMB_MAX_DIAG_DYN_LSL_UP[i]  ≤ C_LAMB_MAX_DIAG_DYN_LSL_UP   &
            LAMB_MIN_DIAG_DYN_LSL_UP[i]   ≥  C_LAMB_MIN_DIAG_DYN_LSL_UP
    % Purpose of this test is to avoid  any shift in the lambda set point .
    % Narrow calibration of those constants can reduce strongly the number of valid measured amplitudes.
    % Moreover, the current lean/rich cycle must be marked as valid via LV_CYCNR_DYN_LSL_UP_VLD[i].

    Then(2)

       LAMB_AMPL_DIAG_DYN_LSL_UP[i] =
            LAMB_MAX_DIAG_DYN_LSL_UP[i] - LAMB_MIN_DIAG_DYN_LSL_UP[i]

       LAMB_AMPL_MIN_DIAG_DYN_LSL_UP[i] =
            IP_LAMB_AMPL_MIN_DYN_LSL_UP - C_LAMB_AMPL_MIN_DYN_LSL_UP

       LAMB_AMPL_MAX_DIAG_DYN_LSL_UP[i] =
```

```
                IP_LAMB_AMPL_MAX_DYN_LSL_UP + C_LAMB_AMPL_MAX_DYN_LSL_UP

        % Purpose of this test is, if wanted, to avoid flat line signals or excessive sensor gain.
        % C_LAMB_AMPL_MAX/MIN can be viewed as are amplitude tolerances necessary due to frequent
        % amplitude dispersion.

        If(3)
           LAMB_AMPL_MIN_DIAG_DYN_LSL_UP[i] ≤ LAMB_AMPL_DIAG_DYN_LSL_UP[i]
           ≤ LAMB_AMPL_MAX_DIAG_DYN_LSL_UP[i]
        Then(3)
              LAMB_AMPL_INT_DIAG_DYN_LSL_UP[i] $_{n}$ =
                       LAMB_AMPL_INT_DIAG_DYN_LSL_UP[i] $_{n-1}$ +
                                                         LAMB_AMPL_DIAG_DYN_LSL_UP[i]

              Increment CTR_DIAG_DYN_LSL_UP[i]

           If(4)  CTR_DIAG_DYN_LSL_UP[i] ≥ CTR_MIN_DIAG_DYN_LSL_UP
           Then(4)
              % Short term mean value of the current engine speed and load condition.
              LAMB_AMPL_MV_DIAG_DYN_LSL_UP[i] =
                       LAMB_AMPL_INT_DIAG_DYN_LSL_UP[i] / CTR_DIAG_DYN_LSL_UP[i]

              % Normalisation of amplitude value to reduce engine speed, load sensibility
              FAC_VALUE_DIAG_DYN_LSL_UP[i] =
              (IP_LAMB_AMPL_MAX_DYN_LSL_UP -
              LAMB_AMPL_MV_DIAG_DYN_LSL_UP[i]) /
              (IP_LAMB_AMPL_MAX_DYN_LSL_UP - IP_LAMB_AMPL_MIN_DYN_LSL_UP)

              FAC_SUM_DIAG_DYN_LSL_UP[i] =
                 FAC_SUM_DIAG_DYN_LSL_UP[i] + FAC_VALUE_DIAG_DYN_LSL_UP[i]

              %Number cycles (short-term measurements)
              increment CYCNR_DIAG_DYN_LSL_UP[i]

              LAMB_AMPL_INT_DIAG_DYN_LSL_UP[i] = 0
              CTR_DIAG_DYN_LSL_UP[i] = 0
           Endif(4)
         Endif(3)
      Endif(2)
      LV_CYCNR_DYN_LSL_UP_VLD[i] = 1
      LAMB_MAX_DIAG_DYN_LSL_UP[i] = 0
      LAMB_MIN_DIAG_DYN_LSL_UP[i] = 32
Endif(1)
```

<u>Condition 4</u>: from STATE_DYN_DIAG[i] = “acquisition” to “end”

```
   LV_DIAG_EOL_DYN_LS_UP[i] = 0            &
   LC_DIAG_DYN_LSL_UP_CAT_SYN = 1          &
   ( CYCNR_DIAG_DYN_LSL_UP[i] ≥ C_CYCNR_END_DIAG_DYN_LSL_UP or
     (STATE_CAT_DIAG = "end" &
     CYCNR_DIAG_DYN_LSL_UP[i] ≥ C_CYCNR_MIN_DIAG_DYN_LSL_UP) )
Or
   LV_DIAG_EOL_DYN_LS_UP[i] = 1            &
   LC_DIAG_DYN_LSL_UP_CAT_SYN = 1          &
```

```
    ( CYCNR_DIAG_DYN_LSL_UP[i] ≥ C_CYCNR_EOL_DIAG_DYN_LSL_UP or
      (STATE_CAT_DIAG = "end" &
      CYCNR_DIAG_DYN_LSL_UP[i] ≥ C_CYCNR_MIN_DIAG_DYN_LSL_UP) )
Or
    LV_DIAG_EOL_DYN_LS_UP[i] = 0              &
    LC_DIAG_DYN_LSL_UP_CAT_SYN = 0            &
    CYCNR_DIAG_DYN_LSL_UP[i] ≥ C_CYCNR_END_DIAG_DYN_LSL_UP
Or
    LV_DIAG_EOL_DYN_LS_UP[i] = 1              &
    CYCNR_DIAG_DYN_LSL_UP[i] ≥ C_CYCNR_EOL_DIAG_DYN_LSL_UP
Or
    LV_DIAG_DYN_S_LSL_UP_END[i] = 1
```

Transition actions:

Diagnosis of sensor dynamic:

```
FAC_MV_DIAG_DYN_LSL_UP[i]  = FAC_SUM_DIAG_DYN_LSL_UP[i] /
CYCNR_DIAG_DYN_LSL_UP[i]

If FAC_MV_DIAG_DYN_LSL_UP[i] ≥ C_FAC_MV_THD_DIAG_DYN_LSL_UP
Then
    % symptom is present
    LV_DIAG_DYN_SYM_LSL_UP[i] = 1
Endif

% return trim ctr and forced stimulation to their original parameters
LV_LAMB_PLS_REQ_DYN_LSL_UP[i] = 0
LV_LAM_ADJ_REQ_DYN_LSL_UP[i] = 0
CTR_WAIT_DIAG_DYN_LSL_UP[i] = 0


STATE_DYN_DIAG[i] = “end”
LV_DIAG_DYN_END_LSL_UP[i] = 1
```

<u>Condition 5</u>: from_STATE_DYN_DIAG[i] = “acquisition” to “passive”

```
LV_DIAG_DYN_CDN_LSL_UP[i] = 0
```

Transition actions:

```
STATE_DYN_DIAG[i] = “passive”
LV_LAMB_PLS_REQ_DYN_LSL_UP[i] = 0
LV_LAM_ADJ_REQ_DYN_LSL_UP[i] = 0
CTR_WAIT_DIAG_DYN_LSL_UP[i] = 0
```

<u>Condition 6</u>: from STATE_DYN_DIAG[i] = “acquisition” to “wait”

```
LC_DIAG_DYN_LSL_UP_CAT_SYN = 1  & STATE_CAT_DIAG = “wait”
```

Transition actions:

```
STATE_DYN_DIAG[i] = “wait”
LAMB_AMPL_INT_DIAG_DYN_LSL_UP[i] = 0
CTR_DIAG_DYN_LSL_UP[i] = 0
```

2.4 STATE_DYN_DIAG[i] = "end"

Actions:

```
LV_CDN_DIAG_DYN_VLD_LS_UP[i] = 1

% scaling of result from dynamic diagnosis after NoX-cat-regeneration to same range as
dynamic diagnosis (homogen)

IF LV_DIAG_DYN_S_LSL_UP_END[i] = 1
THEN
            FAC_MV_DIAG_DYN_LSL_UP[i] =
            IP_FAC_SCA_FAC_MV_DIAG_DYN ( FAC_MV_DIAG_DYN_S_LSL_UP )

            %negative values FAC_MV_DIAG_DYN_S_LSL_UP shall be scaled to zero. This is
            forced by datatype definition of FAC_MV_DIAG_DYN_LSL_UP

ELSE        no change on FAC_MV_DIAG_DYN_LSL_UP[i]
ENDIF
```

Adaptation of sensor time constant:

```
If(1)  LC_T1_AD_DIAG_DYN_LSL_UP = 1                                    &
   LV_DIAG_EOL_DYN_LS_UP[i] = 0                                        &
   FAC_MV_DIAG_DYN_LSL_UP[i] ≥ C_FAC_AD_THD_DIAG_DYN_LSL_UP            &
Then(1)
% update FAC_DIAG_DYN_LSL_UP[i].
% To be consistent with the physical meaning of this variable, its maximum value shall be less than 1
%  see Output Table.
     FAC_DIAG_DYN_LSL_UP[i](new)  =
            FAC_DIAG_DYN_LSL_UP[i](old) + C_CRLC_LPF_FAC_AD_DYN_LSL_UP
            * ( FAC_MV_DIAG_DYN_LSL_UP[i] - FAC_DIAG_DYN_LSL_UP[i](old) )
Else(1)
     If(2)    FAC_MV_DIAG_DYN_LSL_UP[i] < C_FAC_AD_RST_DIAG_DYN_LSL_UP
     Then(2)       FAC_DIAG_DYN_LSL_UP[i](new) = 0
     Else(2)       % FAC_DIAG_DYN_LSL_UP[i] shall retain its value saved in the memory
            FAC_DIAG_DYN_LSL_UP[i](new)  = FAC_DIAG_DYN_LSL_UP[i](old)
     Endif(2)
Endif(1)


IF LV_DIAG_DYN_S_LSL_UP_END[i] = 1
THEN % end forced by stratified diagnosis function
     % force old "diagosis ready" indication and set error flag depending on stratified result
     LV_DIAG_DYN_END_LSL_UP[i] = 1
     FAC_DYN_LSL_DIAG_SAE[i] = FAC_MV_DIAG_DYN_S_LSL_UP[i]
     FAC_DYN_LSL_DIAG_TOL_SAE[i] = C_FAC_DIAG_DYN_S_LSL_UP_MAX
     IF LV_DIAG_DYN_S_LSL_UP_ERR[i] = 0
     THEN % no failure present
            ERR_SYM_DYN_VLD_LS_UP[i] = "NO_SYM"
            LV_ERR_DYN_VLD_LS_UP[i] =  0
```

```
        ELSE  % a failure is present
                ERR_SYM_DYN_VLD_LS_UP[i] = "SYM_1"
                LV_ERR_DYN_VLD_LS_UP[i] = 1
        ENDIF
    ELSE % set error flag depending on homogenous result
        FAC_DYN_LSL_DIAG_SAE[i] = FAC_MV_DIAG_DYN_LSL_UP[i]
        FAC_DYN_LSL_DIAG_TOL_SAE[i] = C_FAC_MV_THD_DIAG_DYN_LSL_UP
        IF LV_DIAG_DYN_SYM_LSL_UP [i] = 0
        THEN % no failure present
                ERR_SYM_DYN_VLD_LS_UP[i] = "NO_SYM"
                LV_ERR_DYN_VLD_LS_UP[i] =  0
        ELSE  % a failure is present
                ERR_SYM_DYN_VLD_LS_UP[i] = "SYM_0"
                LV_ERR_DYN_VLD_LS_UP[i] = 1
        ENDIF
    ENDIF
    % no matter how the end was reached and what the result is, set the diagnosis ready
    LV_END_DIAG_DYN_VLD_LS_UP[i] = 1
```

For failure and error management treatment the anti-bounce mechanism is called :

% if LV_ERR_XX=1 then LV_ERR_SET=1 or if LV_ERR_XX=0 then LV_ERR_RST=1

ACTION_ERRM_NoFilterSymptom(IN<NC_IDX_DIAG_ DYN_VLD_LS_UP[i]>, IN<LV_CDN_DIAG_ DYN_VLD_LS_UP[i]>, IN< ERR_SYM_ DYN_VLD_LS_UP[i] >, IN< LV_ERR_SET_ DYN_VLD_LS_UP[i] >, IN<LV_ERR_RST_ DYN_VLD_LS_UP[i]>, IN<LV_END_DIAG_DYN_VLD_LS_UP[i]>, OUT<LV_ERR_ DYN_VLD_LS_UP[i]>)

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_CTR_MIN_DIAG_DYN_LSL_UP	1	0...FFH	0...255	1	[-]
Number of forced stimulation periods needed to calculate the short term mean value					
C_CTR_MIN_DIAG_EOL_DYN_LSL_UP	1	0...FFH	0...255	1	[-]
Number of forced stimulation periods needed to calculate the short term mean value (EOL test)					
C_CTR_WAIT_DIAG_DYN_LSL_UP	1	0...FFH	0...255	1	[-]
Number of forced stimulation periods during the diagnosis status transition "wait" to "acquisition"					
C_CYCNR_MIN_DIAG_DYN_LSL_UP	1	0...FFH	0...255	1	[-]
Minimum number of short term mean values to calculate the long term mean value					
C_CYCNR_END_DIAG_DYN_LSL_UP	1	0...FFH	0...255	1	[-]
Number of long term mean values to deliver the sensor dynamic diagnostic					
C_CYCNR_EOL_DIAG_DYN_LSL_UP	1	0...FFH	0...255	1	[-]
Number of long term mean values to deliver the sensor dynamic diagnostic (EOL test)					
C_N_MIN_DIAG_DYN_LSL_UP	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed to activate the diagnosis					
C_N_MAX_DIAG_DYN_LSL_UP	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed to activate the diagnosis					
C_MAF_MAX_DIAG_DYN_LSL_UP	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Maximum mass airflow rate to activate the diagnosis function					
C_MAF_MIN_DIAG_DYN_LSL_UP	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Minimum mass airflow rate to activate the diagnosis function					
C_CRLC_LPF_LAMB_DIAG_DYN_LSL_UP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation factor for low pass filtered lambda value					
C_LAMB_MIN_DIAG_DYN_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lowest allowed sensor signal min value					
C_LAMB_MAX_DIAG_DYN_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Highest allowed sensor signal max value					
C_LAMB_AMPL_MIN_DYN_LSL_UP	1	0...7FH	0...0.12402	0.9766e-3	[-]
Calibration data for adjust the expected amplitude: low range					
C_LAMB_AMPL_MAX_DYN_LSL_UP	1	0...7FH	0...0.12402	0.9766e-3	[-]
Calibration data for adjust the expected amplitude: low range					
C_FAC_MV_THD_DIAG_DYN_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Limited diagnosis value for the error diagnosis					
C_FAC_AD_THD_DIAG_DYN_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Diagnosis value threshold for begin adaptation of sensor time constant model					
C_FAC_AD_RST_DIAG_DYN_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Diagnosis value threshold to reset adaptation of sensor time constant model					
C_CRLC_LPF_FAC_AD_DYN_LSL_UP	1	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation factor for low pass filtered (sensor time constant) adaptation factor					
IP_LAMB_AMPL_MAX_DYN_LSL_UP	8*8	0...7FH	0...0.12402	0.9766e-3	[-]
LDPM_N_32_LS_UP_DYN	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_LS_UP_DYN	8	0...FFH	0...1389	5.4470588	[mg/stk]
Nominal sensor expected amplitude during the Dynamic Diagnosis Rem.: The same points in LDPM_N_32_2_EGTR and LDPM_MAF_HB_2_EGTR shall be used					
IP_LAMB_AMPL_MIN_DYN_LSL_UP	8*8	0...7FH	0...0.12402	0.9766e-3	[-]
LDPM_N_32_LS_UP_DYN	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_LS_UP_DYN	8	0...FFH	0...1389	5.4470588	[mg/stk]
Limit sensor expected amplitude during the Dynamic Diagnosis Rem.: The same points in LDPM_N_32_2_EGTR and LDPM_MAF_HB_2_EGTR shall be used					
IP_T_DYN_DIAG	8*8	0...FFFFH	0...1310.7	0.02	[s]
LDPM_N_32_LS_UP_DYN	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_LS_UP_DYN	8	0...FFH	0...1389	5.4470588	[mg/stk]
Amplitude measurement time: same points than in LDPM_N_32_2_EGTR and LDPM_MAF_HB_2_EGTR shall be used					
LC_DIAG_DYN_LSL_UP_CAT_SYN	1	0...1H	0...1	1	[-]
Boolean switch to synchronise the dynamic diagnosis with the catalyst efficiency diagnosis					
LC_T1_AD_DIAG_DYN_LSL_UP	1	0...1H	0...1	1	[-]
Boolean switch to allow the adaptation of the sensor time constant model T1_LSL_UP[i]					

IP_FAC_SCA_FAC_MV_DIAG_DYN	4	0...7FFFH	0...31.99902	0.9766e-3	[-]
LDP_FAC_MV_DIAG_DYN_S_LSL_UP	4	0...7FFFH	0...127.99609	3.9062e-3	[-]
Scaling of result from dynamic diagnosis after NoX-cat-regeneration to fit result from dynamic diagnosis (homogen)					

Configuration for diagnostic symptoms:

Diagnostic XX	**Symptom description**	**Symptom**	**Filter type**
DYN_VLD_LS_UP[NC_CBK_EX_NR]	too slow dynamic (homogenous mode)	SYM_0	NO
	too slow dynamic (stratified mode)	SYM_1	

16.47 O2 sensor (lin, up) dynamic diag (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_RBM_DYN_VLD_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of Rate Based Monitoring of "Diagnosis of WRAF Sensor Dynamic" by malfunction					
LV_DIAG_DYN_INH_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "Diagnosis of the WRAF Sensor Dynamic"					
LV_INH_DIAG_LS_UP_DYN_CP	V/O	0...1H	0...1	1	[-]
Canister Purge Load too high					
STATE_RBM_DYN_VLD_LS_UP[NC_CBK_EX_NR]	V/O	0...7H	0...7	1	[-]
Interface of DYN_VLD_LS_UP					
LV_DIAG_EOL_DYN_LS_UP_DC[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical Value as reminder for End of Line test within this driving cycle					
LV_DIAG_EOL_DYN_LS_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status variable to activate the short test trip at the end of line					

Input data:

LV_IGK	LV_CDN_VB_OBD2	LV_ERR_MAF	LV_ERR_AMP
LV_ERR_SA_SAV	LV_ERR_AMP_PLAUS	LV_ERR_SA_SYS	LV_ERR_CAM_CUS
LV_ERR_SAV	LV_ERR_CHG_LS_UP	LV_ERR_TCO	LV_ERR_EL_CPS
LV_ERR_TPS	LV_ERR_CRK	LV_MIS_STATE_B	LV_ERR_DIAGCPS
LV_ERR_IVVT	NC_CBK_EX_NR	LV_ERR_MAP_DIP_SHIFT	LV_ERR_SAP
LV_ERR_SA_SAP	LV_ERR_MAP_DIP_SENS	LV_ERR_SA_SAV_LSL	LV_DC
CTR_ERR_DYN_NR	LV_ERR_MAP_DIP_PLAUS	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	
LV_ERR_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]
STATE_CP	LV_ERR_FSD[NC_CBK_EX_NR]	CL_MMV	LV_CL_CLC_AVL
LV_END_DIAG_DYN_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_IN	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]		

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation: at every transition LV_IGK = 0->1or reset all variables shall be reset to 0.

at every transition LV_IGK = 1->0 LV_DIAG_DYN_INH_LSL_UP[i] = 1

Recurrence: the functions shall be carried out once every 1s.

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

The function "Diagnosis of the WRAF Sensor Dynamic" has to be deactivated, if at least one of the following errors is active.

IF - LV_ERR_AMP = 1 **OR**
- LV_ERR_AMP_PLAUS = 1 **OR**
- LV_ERR_CRK = 1 **OR**
- LV_ERR_CAM_CUS = 1 **OR**
- LV_ERR_TCO = 1 **OR**
- LV_ERR_TPS = 1 **OR**
- LV_ERR_MAF = 1 **OR**
- LV_ERR_MAP_DIP_SENS = 1 **OR**
- LV_ERR_MAP_DIP_PLAUS = 1 **OR**
- LV_ERR_MAP_DIP_SHIFT = 1 **OR**
- LV_ERR_EL_CPS = 1 **OR**
- LV_ERR_DIAGCPS = 1 **OR**
- LV_ERR_SA_SAV = 1 **OR**
- LV_ERR_SA_SAV_LSL = 1 **OR**
- LV_ERR_SA_SAP = 1 **OR**
- LV_ERR_SA_SYS = 1 **OR**
- LV_ERR_SAV = 1 **OR**
- LV_ERR_SAP = 1 **OR**
- LV_ERR_IVVT = 1 **OR**
- LV_MIS_STATE_B = 1 **OR**
- LV_ERR_FSD[i] = 1 **OR**
- LV_ERR_CHG_LS_UP = 1 **OR**
- LV_ERR_LOAD_TPS_PLAUS = 1 **OR**
- LV_ERR_EL_LSL_UP[i] = 1 **OR**
- LV_ERR_OC_LSL_UP[i] = 1 **OR**
- LV_ERR_LSL_UP_IF[i] = 1 **OR**
- LV_ERR_LSH_UP[i] = 1 **OR**
- LV_ERR_OBD_VLD_LSH_UP[i]= 1 **OR**
- LV_ERR_PUC_VLD_LS_UP[i] = 1 **OR**
- LV_ERR_SHIFT_AFL_LSL_UP[i] = 1 **OR**
- LV_ERR_SHIFT_AFR_LSL_UP[i] = 1 **OR**
- LV_ERR_AIR_LSL_UP[i] = 1 **OR**
- LV_ERR_LS_DOWN[i] = 1 **OR**
- LV_ERR_DELTA_I_LAM[i] = 1 **OR**
- LV_ERR_VLS_DOWN_DIF[i] = 1 **OR**
- LV_ERR_FUP = 1 **OR**
- LV_ERR_FUP_MFP_PLAUS = 1 **OR**
- LV_ERR_H_PRS_SYS = 1 **OR**
- LV_ERR_VCV = 1 **OR**

```
    -   LV_ERR_FUP_ORNG = 1                                   OR
    -   LV_ERR_FUP_ST = 1                                     OR
    -   LV_ERR_TTIP_MES_LSH_UP[i]  = 1                        OR
    -   LV_ERR_FSD_LAM_LIM[i]  = 1                            OR
    -   LV_ERR_SLV_IVVT_IN = 1                                OR
    -   LV_ERR_OFS_LSL_UP[i] = 1                              OR
    -   LV_ERR_CTL_LSL_UP[i] = 1                              OR
    -   LV_ERR_MAP_TPS_PLAUS = 1
    -

THEN     LV_INH_DIAG_RBM_DYN_VLD_LS_UP[i]  =  1
ELSE     LV_INH_DIAG_RBM_DYN_VLD_LS_UP[i]  =  0
ENDIF
```

The function “Diagnosis of the WRAF Sensor Dynamic” has to be deactivated, if at least one of the special modes is active.

```
IF   LV_LS_UP_OBD_2_MAN_DEAC[i]  = 1                         OR
     LV_CDN_VB_OBD2 = 0                                      OR
     LV_INH_DIAG_RBM_DYN_VLD_LS_UP[i] = 1
THEN     LV_DIAG_DYN_INH_LSL_UP[i]  =  1
ELSE     LV_DIAG_DYN_INH_LSL_UP[i]  =  0
ENDIF
```

<u>% Monitoring Canister Purge</u>

```
IF   LV_DIAG_EOL_DYN_LS_UP[i] = 0
  THEN                                      % short test not required
     IF      STATE_CP = NO_PURGE  OR
             STATE_CP = CP_NOT_ACT OR
             STATE_CP = WAIT_RAMP_OPEN  OR
             (STATE_CP = MIN_PURGE & LV_CL_CLC_AVL  = 1)   OR
             (STATE_CP = MAX_PURGE & CL_MMV ≤ C_CL_MMV_LS_UP_DIAG_MAX)
      THEN         LV_INH_DIAG_LS_UP_DYN_CP  =  0
      ELSE         LV_INH_DIAG_LS_UP_DYN_CP  =  1
     ENDIF
  ELSE                                      % short test required
     IF  (STATE_CP = MAX_PURGE) & (CL_MMV ≤ C_CL_MMV_LS_UP_DIAG_MAX)) OR
         STATE_CP = NO_PURGE
      THEN         LV_INH_DIAG_LS_UP_DYN_CP  =  0
      ELSE         LV_INH_DIAG_LS_UP_DYN_CP  =  1
     ENDIF
ENDIF
```

16.47.1 Interface for Rate – Based - Monitoring

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the DYN_VLD_LS_UP[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_DYN_VLD_LS_UP[i] data.

Within STATE_RBM_DYN_VLD_LS_UP[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for catalyst diagnosis)

Application conditions:

Initialisation: *at ECU reset:*

bit 0, bit 1 and bit 2 of STATE_RBM_DYN_VLD_LS_UP[i] = 0
LV_DIAG_EOL_DYN_LS_UP_DC[i] = 0

LV_DC 0 ->1 transition:

bit 0 and bit 1 of STATE_RBM_DYN_VLD_LS_UP[i] = 0
LV_DIAG_EOL_DYN_LS_UP_DC[i] = 0

on failure memory reset:

bit 1 of STATE_RBM_STATE_RBM_DYN_VLD_LS_UP[i] = 0
LV_DIAG_EOL_DYN_LS_UP_DC[i] = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 ->1 transition

bit 2 of STATE_RBM_DYN_VLD_LS_UP[i] = 1

The pending status of the following failures has to be checked only once :

LV_ERR_MEC_IVVT_IN[NC_NR_CBK_IVVT]	LV_ERR_MAF	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]
LV_ERR_MEC_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_SA_SAV	LV_ERR_LOAD_TPS_PLAUS
	LV_ERR_SA_SAV_LSL	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_SA_SAP	LV_ERR_FSD[NC_CBK_EX_NR]
LV_ERR_CRK_TOOTH	LV_ERR_SA_SYS	LV_ERR_REF_CRK_CAM_IN[NC_NR_CAM_CBK]
LV_ERR_CRK_PLAUS	LV_ERR_SAV	LV_ERR_REF_CRK_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_CRK_TOOTH_PER	LV_ERR_SAP	LV_ERR_SYN_CAM_IN[NC_NR_CAM_CBK]
		LV_ERR_PLAUS_CAM_IN[NC_NR_CAM_CBK]
LV_ERR_CRK_SYN	LV_ERR_MIS[NC_CYL_NR]	LV_ERR_PLAUS_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_MAP_DIP_SHIFT	LV_ERR_AMP	LV_ERR_PER_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_MAP_DIP_SENS	LV_ERR_CHG_LS_UP	LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_EL_CPS	LV_ERR_DIAGCPS	LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_TCO_EL	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]
LV_ERR_TCO_GRD	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	
LV_ERR_TCO_PLAUS	LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]
LV_ERR_TCO_STUCK	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]
LV_ERR_MAP_DIP_PLAUS	LV_ERR_AMP_PLAUS	LV_ERR_PER_CAM_IN[NC_NR_CAM_CBK]
LV_ERR_SYN_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_TPS_AD_BOL	LV_ERR_TCO_STUCK_RNG
LV_ERR_FUP[NC_CBK_HPP_NR]	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]
LV_ERR_VCV[NC_CBK_HPP_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_TPS_2
LV_ERR_TPS_1	LV_ERR_TPS_AD	LV_ERR_TPS_RATIO
LV_ERR_TPS_MAF_1	LV_ERR_TPS_MAF_2	LV_ERR_TPS_ST_CHK_2
LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3	LV_ERR_MTC_DR
LV_ERR_MAP_TPS_PLAUS		LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_IN		LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]

```
If(1)     { CPU optimization at LV_DC 0 ->1 transition }
          CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
          While bit 1 of STATE_RBM_DYN_VLD_LS_UP[i] = 0 do
                with each XX failure of the above list  (depending on configuration, see below):
                ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>, SYNCHRONIZATION<CALL>)

                If(2)     XX has a pending status
                Then(2)        bit 1 of STATE_RBM_DYN_VLD_LS_UP[i] = 1
                Endif(2)
          Endwhile
Else(1)   { the dynamic failure memory is empty }      No action
```

```
    Endif(1)

  Every 1 s :
  If        LV_DIAG_EOL_DYN_LS_UP[i] = 1
  Then      LV_DIAG_EOL_DYN_LS_UP_DC[i] = 1
  Endif

  If   bit 0 of STATE_RBM_ DYN_VLD_LS_UP[i] = 0
  Then
            If      LV_END_DIAG_DYN_VLD_LS_UP[i] = 1
            And     LV_DIAG_EOL_DYN_LS_UP_DC[i] = 0
            Then    bit 0 of STATE_RBM_DYN_VLD_LS_UP[i] = 1
            Endif
  Endif

  If   bit 1 of STATE_RBM_DYN_VLD_LS_UP[i] = 0
  Then
            If      LV_INH_DIAG_RBM_DYN_VLD_LS_UP[i] = 1
            Then    bit 1 of STATE_RBM_DYN_VLD_LS_UP[i] = 1
            Endif
  Endif
```

16.47.2 Activation of short trip for WRAF sensor dynamic diagnosis

FUNCTION DESCRIPTION:

General information:

Not yet defined

Application conditions:

Initialisation: LV_DIAG_EOL_DYN_LS_UP[NC_CBK_EX_NR] = 0 *(never changed)*

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CL_MMV_LS_UP_DIAG_MAX	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum degree of saturation of the evaporative canister to activate the function					

16.48 O2 sensor (lin, up) dynamic diagnosis after NOx cat regeneration

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_DYN_S_LSL_UP_END[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
End flag from stratified combustion lambda sensor dynamics diagnosis					
LV_DIAG_DYN_S_LSL_UP_ERR[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
error signalization from stratified dynamics diagnosis					
STATE_SYM_DIAG_DYN_S_LSL_UP[NC_CBK_EX_NR]	O/V	0...FFH	0...255	1	-
error symptom from stratified dynamics diagnosis					
FAC_MV_DIAG_DYN_S_LSL_UP[NC_CBK_EX_NR]	O/V	8000...7FFFH	-128 ... 127.996094	0.00390625	-
final diagnostic value of stratified dynamics diagnosis					
CONC_O2_MES_DIAG_DYN_S[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993896	6.10352E-5	-
O2 concentration derived from measured lambda value from sensor					
LV_DIAG_DYN_S_LSL_UP_SAMPLE[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
signals that a new diagnostic sample should be calculated and included into the average					
CTR_DIAG_DYN_S_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6.5535E+4	1	-
number of valid diagnostic samples in the current driving cycle					
FAC_SUM_DIAG_DYN_S_LSL_UP[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-8.3886E+6 ... 8.38861E+6	0.00390625	-
sum of all diagnostic values for stratified dynamics diagnosis sampled in the current driving cycle					
CONC_O2_CLC_DIAG_DYN_S[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993896	6.10352E-5	-
O2 concentration derived from lambda value of sensor model					
LV_DIAG_DYN_S_LSL_UP_STOP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
signals that the diagnosis shall sleep until reactivated by another homogenous combustion phase					
CONC_O2_CLC_INT_DIAG_DYN_S[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993896	6.10352E-5	-
integral of modelled O2 concentration in stratified dynamics diagnosis					
FAC_VALUE_DIAG_DYN_S_LSL_UP[NC_CBK_EX_NR]	V	8000...7FFFH	-128 ... 127.996094	0.00390625	-
last sample of stratified dynamics diagnosis value					
CONC_O2_DIF_INT_DIAG_DYN_S[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993896	6.10352E-5	-
integral of the difference between modelled and measured O2 concentration in stratified dynamics diagnosis					
CONC_O2_MES_MAX_DIAG_DYN_S[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993896	6.10352E-5	-
measured O2 concentration at time of last transition out of homogenous combustion mode					
CONC_O2_MES_DIAG_DYN_S_ST[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993896	6.10352E-5	-
measured O2 concentration at time of last transition out of homogenous combustion mode					
CONC_O2_CLC_DIAG_DYN_S_ST[NC_CBK_EX_NR]	V	8000...7FFFH	-2...1.99993896	6.10352E-5	-
O2 concentration from sensor model at the last transition out of homogenous combustion mode					
MAF_INT_DIAG_DYN_S_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...2.91267E+3	0.04444444	g
MAF integral since transition out of homogenous combustion mode					

Input data:

LV_HOM_ACT	LV_INH_DYN_DIAG_S_LS	LAMB_SP_FIL_S[NC_CBK	LAMB_LS_UP[NC_CBK_E

	L_UP[NC_CBK_EX_NR]	_EX_NR]	X_NR]
N_32	MAF_CYL	TQI_SP_S	NC_CBK_EX_NR
LV_DIAG_DYN_S_SWI_CMB			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CONC_O2_CLC_HOM_MAX_DYN_S	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
maximum allowable modelled O2 concentration for stratified dynamics diagnosis					
C_CONC_O2_DIF_MAX_DIAG_DYN_S	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
maximum allowable difference between measured and modelled O2 concentration for stratified dynamics diagnosis					
C_CONC_O2_DIF_ST_MAX_DIAG_DYN_S	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
maximum difference between measured and modelled O2 concentration to allow stratified dynamics diagnosis					
C_CONC_O2_MES_DEC_MAX_DYN_S	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
maximum allowable difference between measured and modelled O2 concentration for stratified dynamics diagnosis					
C_CTR_DIAG_DYN_S_LSL_UP	1	0...FFH	0...255	1	-
Number of valid regeneration cycles needed for averaging in stratified dynamics diagnosis					
C_FAC_CONC_O2_COR_DIAG_DYN_S	1	0...FFH	0...1.9921875	0.0078125	-
scaling factor in stratified dynamics diagnosis for correction of measured O2 concentration					
C_FAC_DIAG_DYN_S_LSL_UP_MAX	1	8000...7FFFH	-128 ... 127.996094	0.00390625	-
maximum dynamics characterization value from stratified diagnosis allowable for a "good" lambda sensor					
C_MAF_INT_END_DIAG_DYN_S	1	0...FFFFH	0...2.91267E+3	0.04444444	g
air mass integral after end of regeneration phase after which to one diagnosis cycle ends					
C_MAF_INT_ST_DIAG_DYN_S	1	0...FFFFH	0...2.91267E+3	0.04444444	g
air mass integral to wait for before starting integration of differences in stratified dynamics diagnosis					
C_N_MAX_STOP_DIAG_DYN_S	1	0...FFH	0...8.16E+3	32	rpm
maximum engine speed to conduct stratified dynamics diagnosis					
C_N_MIN_STOP_DIAG_DYN_S	1	0...FFH	0...8.16E+3	32	rpm
minimum engine speed to conduct stratified dynamics diagnosis					
C_TQI_MAX_DIAG_DYN_S	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
maximum torque request allowable for stratified dynamics diagnosis					
C_TQI_MIN_DIAG_DYN_S	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
minimum torque request allowable for stratified dynamics diagnosis					
LC_DIAG_DYN_S_LSL_UP_MOD	1	0...1H	0...1	1	-
switch between integral and difference as basis for dynamics diagnosis					
IP_FAC_COR_CONC_O2_DIF_DYN_HOMS	8x8	0...FFH	0...0.06225586	2.44141E-4	-
LDPM_N_32_LS_UP_DYN	8	0...FFH	0...8.16E+3	32	rpm
LDP_TQI_SP_S_IP_CONC_O2_DYN_S	8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
weighting factor for dynamics diagnosis result					
IP_FAC_COR_CONC_O2_DIF_DYN_S	8x8	0...FFH	0...0.06225586	2.44141E-4	-
LDPM_N_32_LS_UP_DYN	8	0...FFH	0...8.16E+3	32	rpm
LDP_TQI_SP_S_IP_CONC_O2_DYN_S	8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
weighting factor for dynamics diagnosis result					

16.48.1 EGCP_PLADGLSL6

This is an alternative diagnosis for the sensor dynamics of WRAF sensors in stratified combustion engines. It expects NOx trap regeneration or other short phases of homogenous combustion in a normally stratified or other lean engine operation. The function interprets the transition from homogenous to stratified mode as a step-like input to the sensor and monitors the sensor's signal response as it rises from around lambda=1 to higher values (typically above lambda=2).

The difference between this response and the modelled behaviour of a good sensor is integrated, start and end of the integral can be calibrated in terms of a MAF integral to focus on the parts of the signal trace that show large and reproducible differences between new and aged sensors. To get more even weighting of differences in the high and low lambda ranges, the modelled and measured lambda values are transformed into O_2 concentrations and the integrals are calculated with these concentrations. A map is used to correct for influences of mass air flow and engine speed on the sensor response.

Application Condition

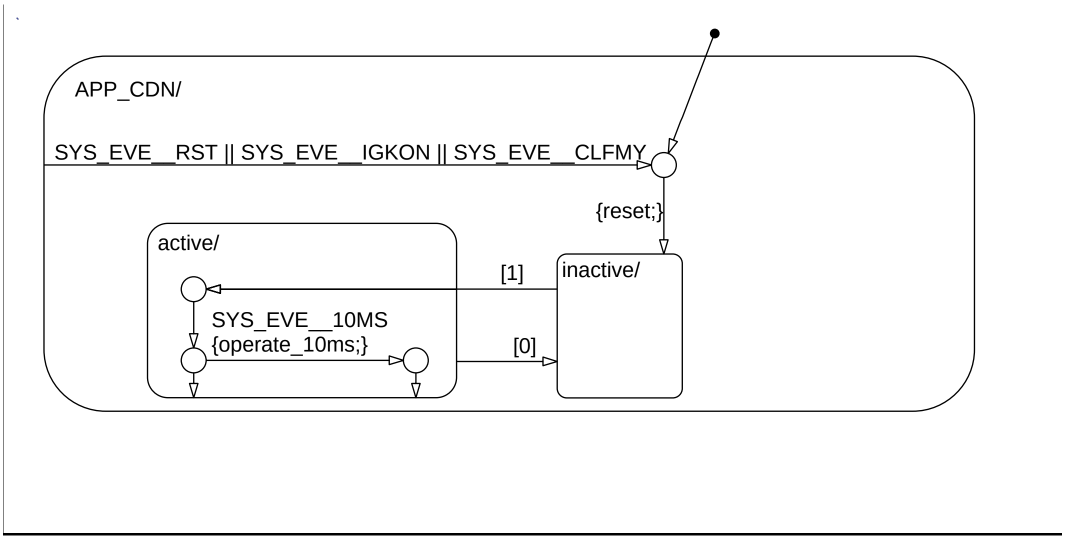


Figure 60 EGCP_PLADGLSL6/ APP_CDN/ Chart

Function Description

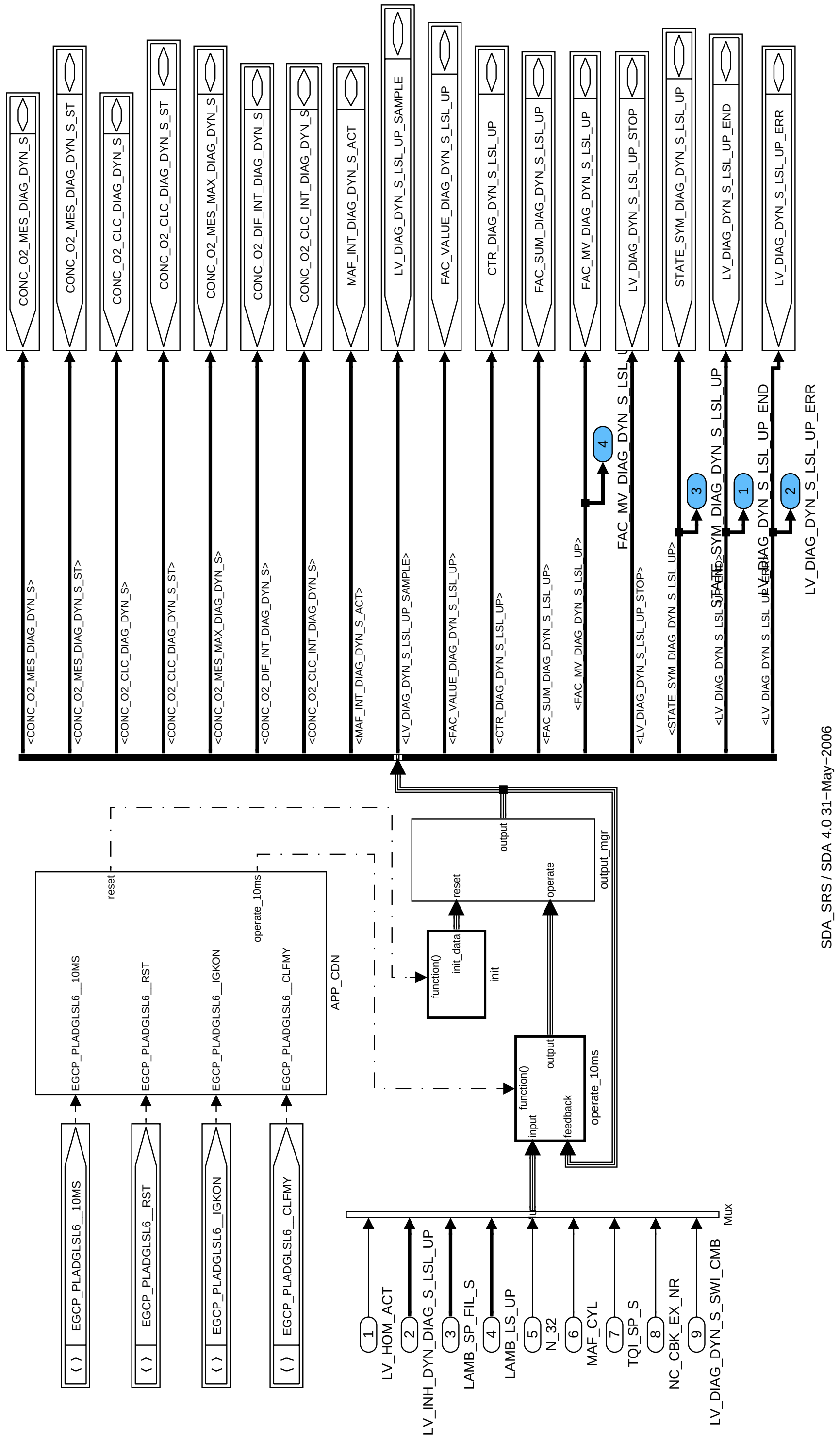


Figure 61 EGCP_PLADGLSL6

16.48.1.1 EGCP_PLADGLSL6/INIT

Initialization of the online data takes place at reset, when clearing the error memory and at the beginning of a driving cycle.

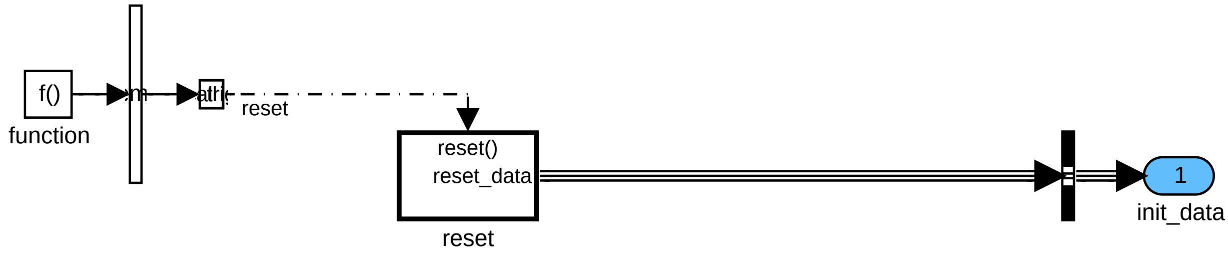


Figure 62 EGCP_PLADGLSL6/ init

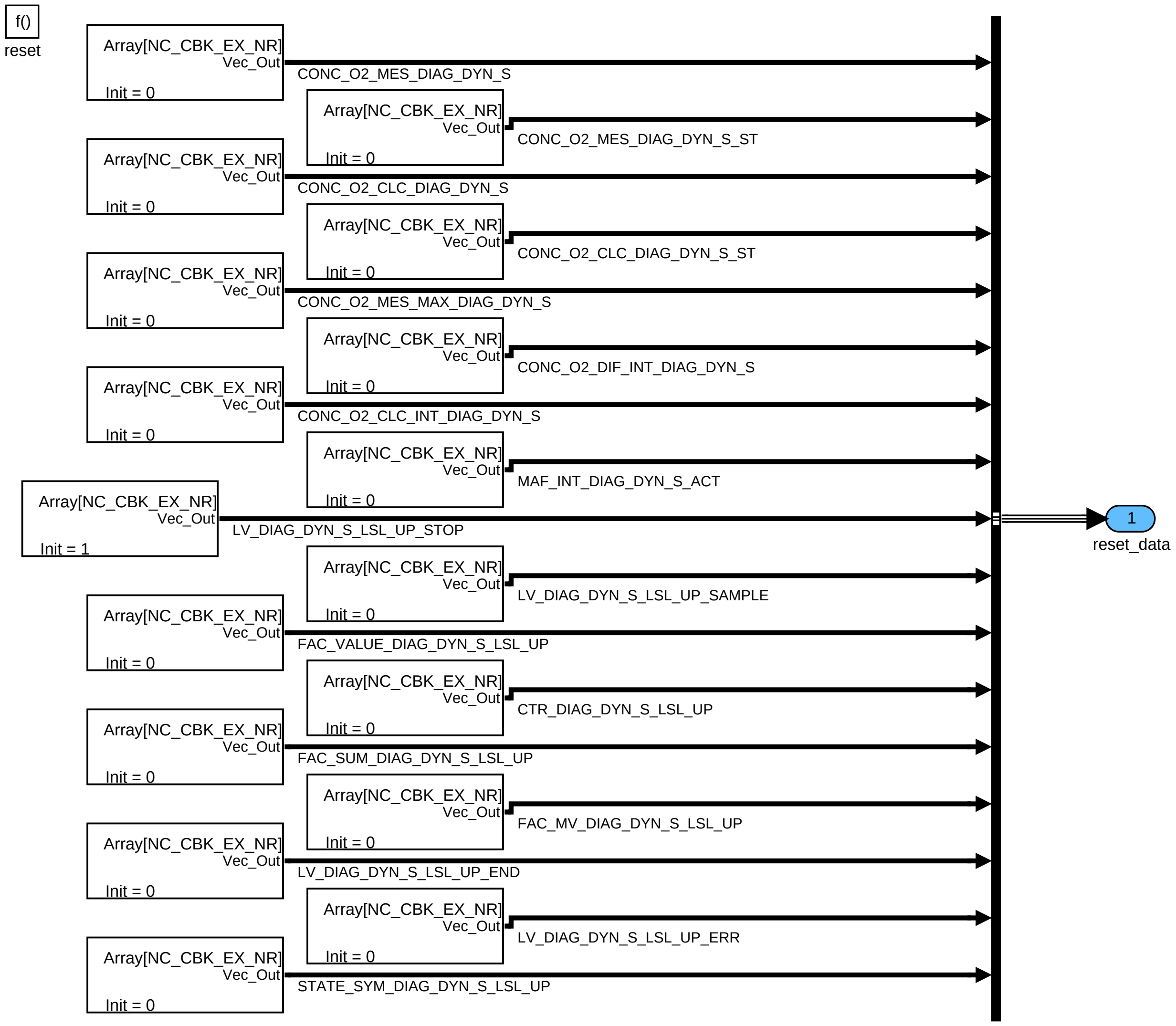


Figure 63 EGCP_PLADGLSL6/ init/ reset

16.48.1.2 SUBFUNCTION: operate_10ms

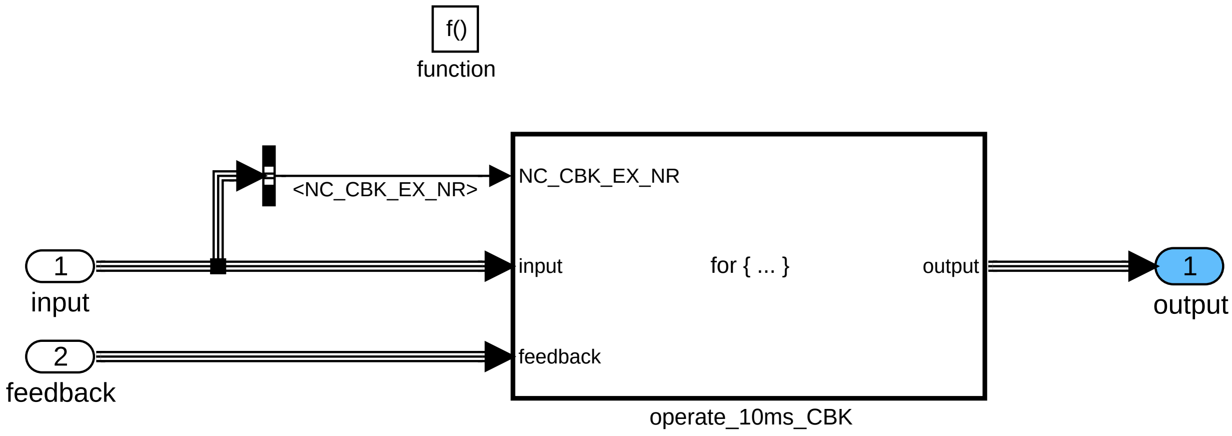


Figure 64 EGCP_PLADGLSL6/ operate_10ms

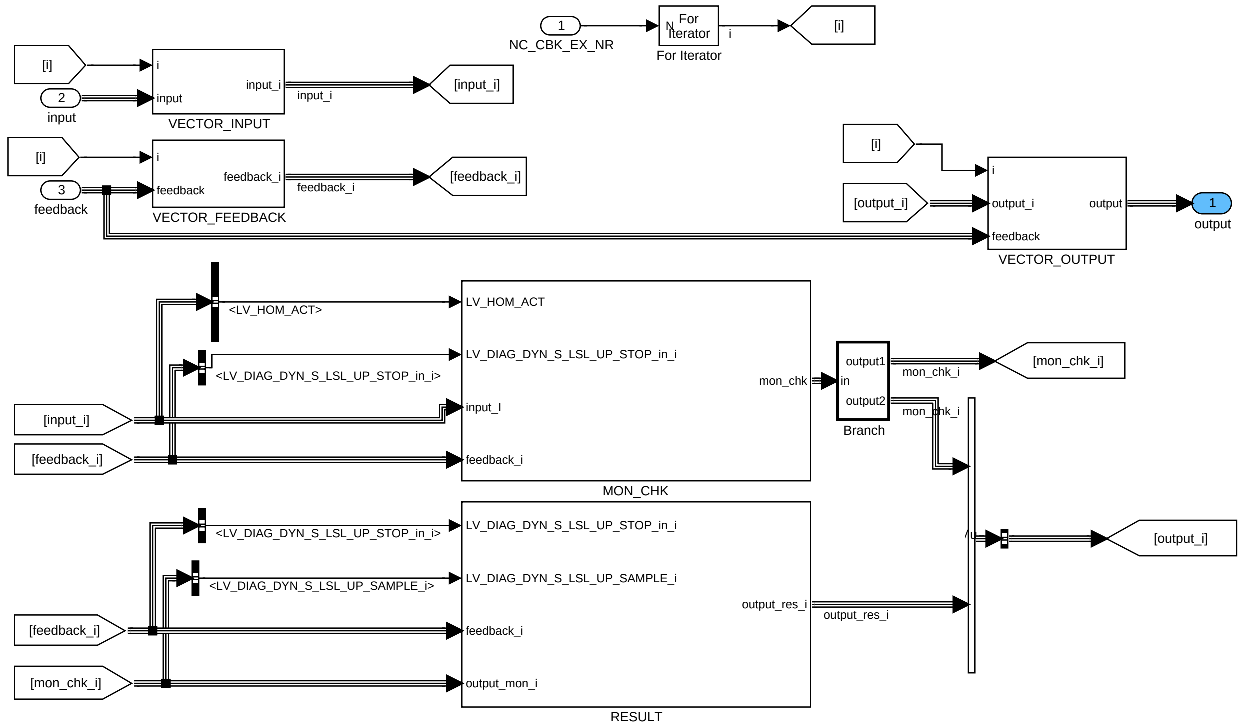


Figure 65 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK

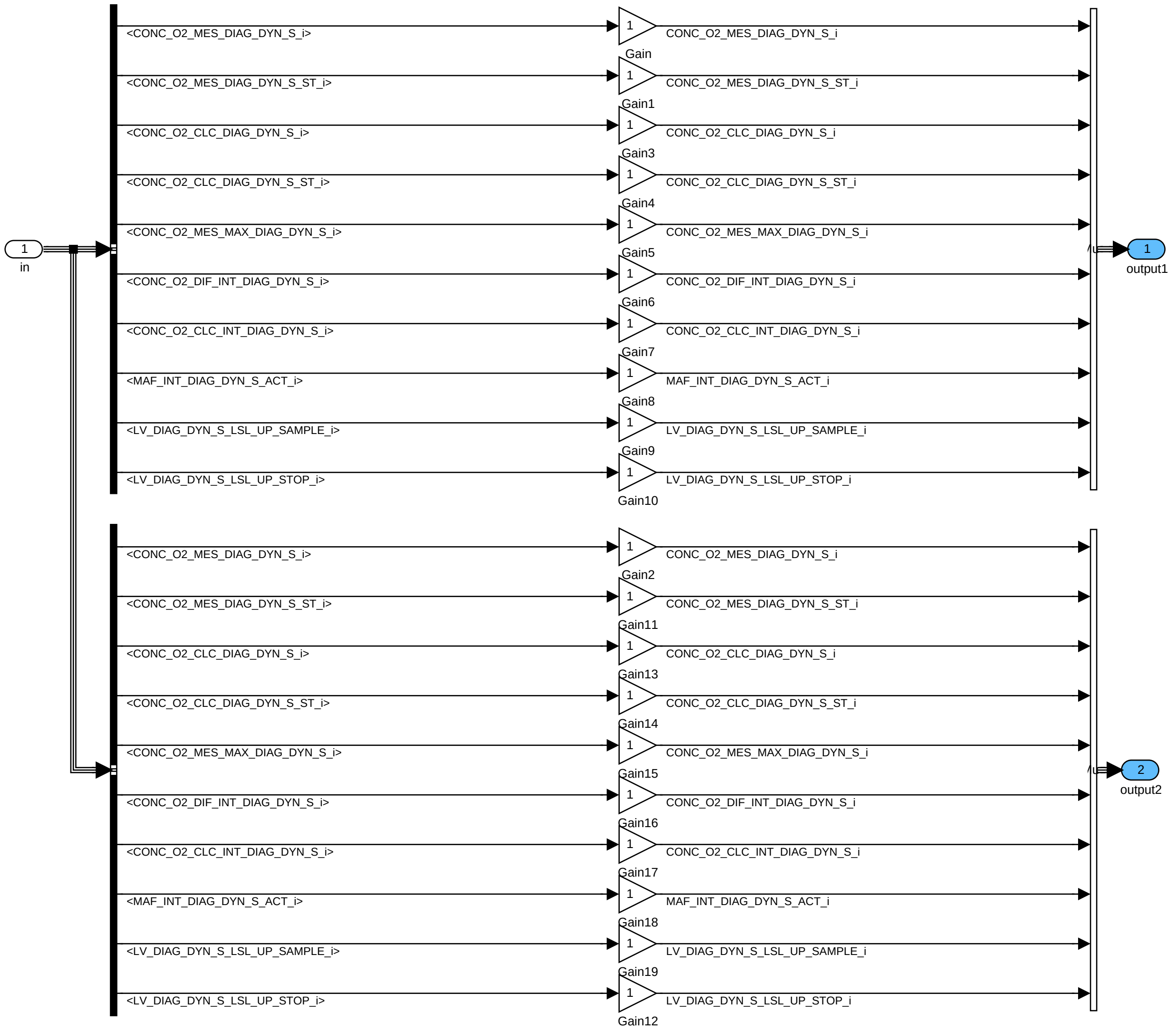


Figure 66 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ Branch

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/MON_CHK

This is the first part of the diagnosis, which continuously monitors the WRAF sensor signal and calculates several integrals (see below for further explanation). It starts if the engine is running in homogenous combustion mode and continues into the start of the following phase of stratified combustion until aborted or finished.

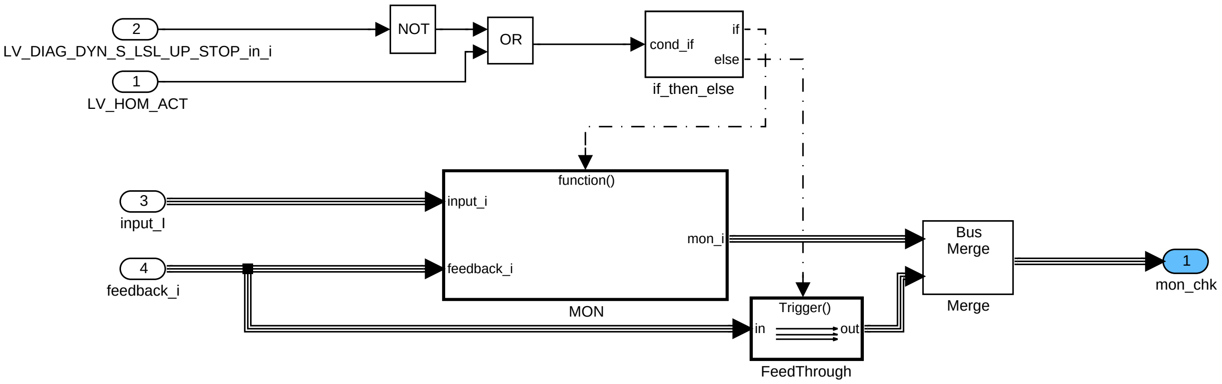


Figure 67 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK

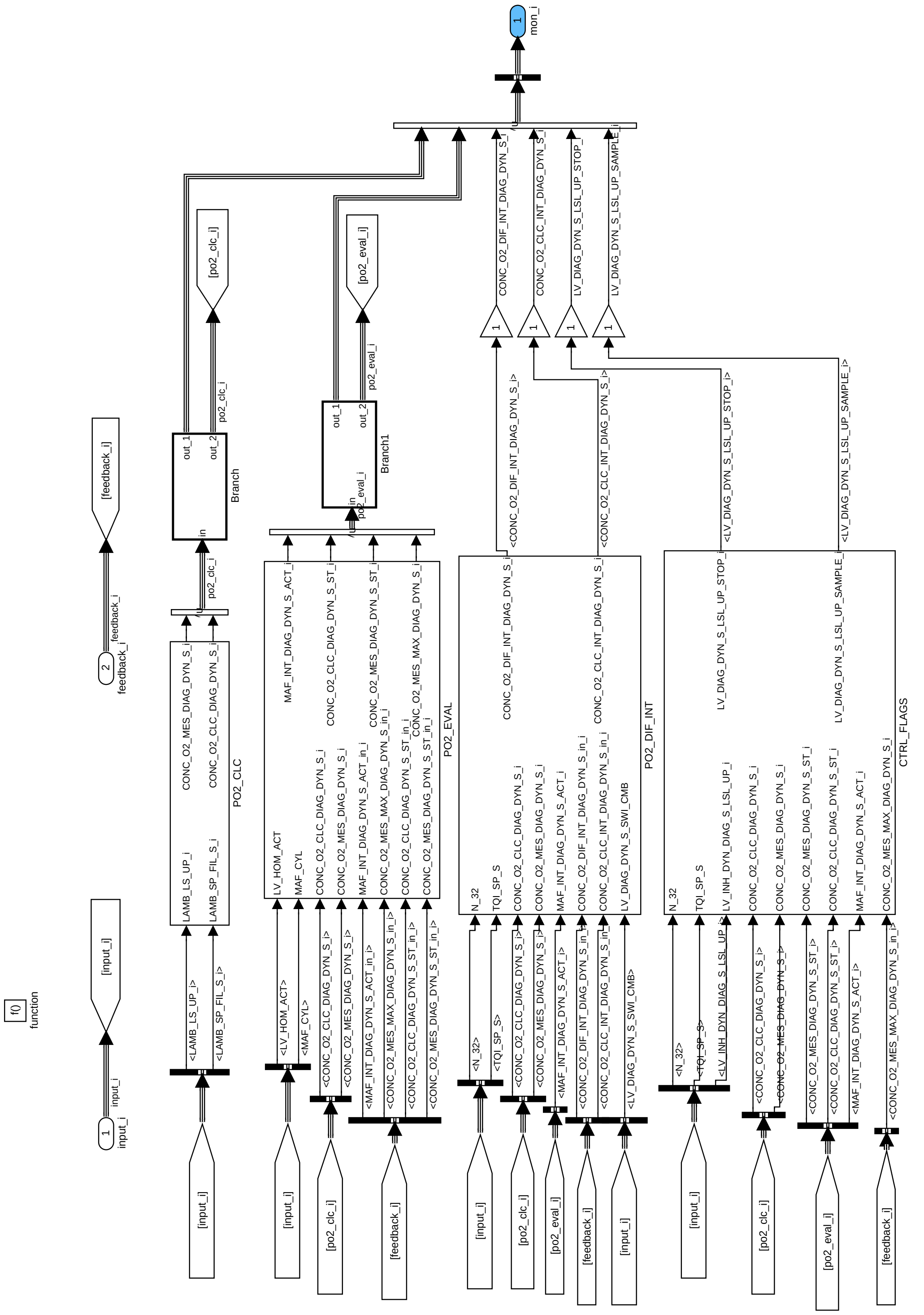


Figure 68 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK/ MON

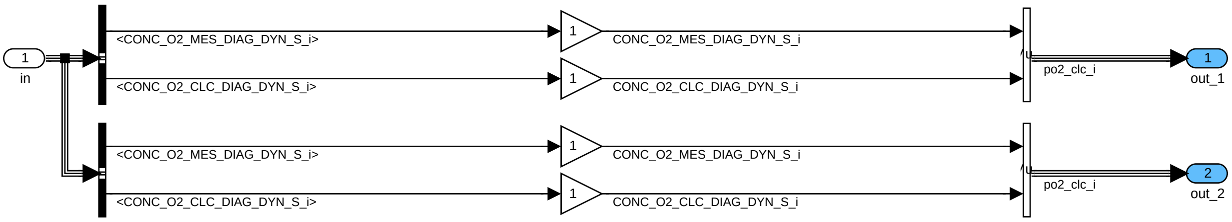


Figure 69 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK/ MON/ Branch

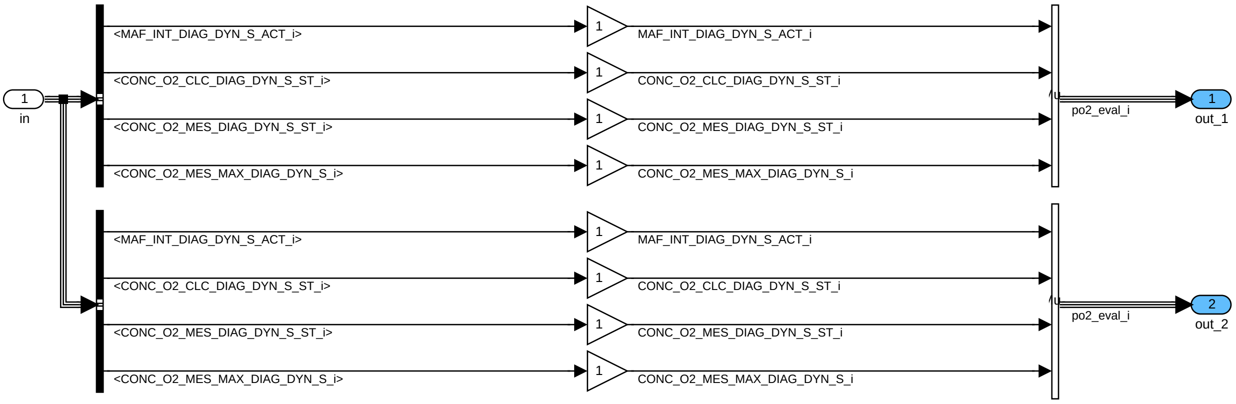


Figure 70 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK/ MON/ Branch1

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/MON_CHK/MON/CTRL_FLAGS

This section shall generate two flags. LV_DIAG_DYN_S_LSL_UP_SAMPLE[i] triggers the calculation of a result at the end of the current diagnostic cycle. It will be set to 1 in the last recurrence before the function deactivates itself, and reset to 0 at the start of the next cycle.

LV_DIAG_DYN_S_LSL_UP_STOP[i] is expected to be set to 0 during a homogenous combustion phase to indicate that a diagnostic cycle can be started at the next transition to stratified combustion. It will be set to 1 if the cycle should be stopped, either because a result is available or because required conditions are not fulfilled anymore.

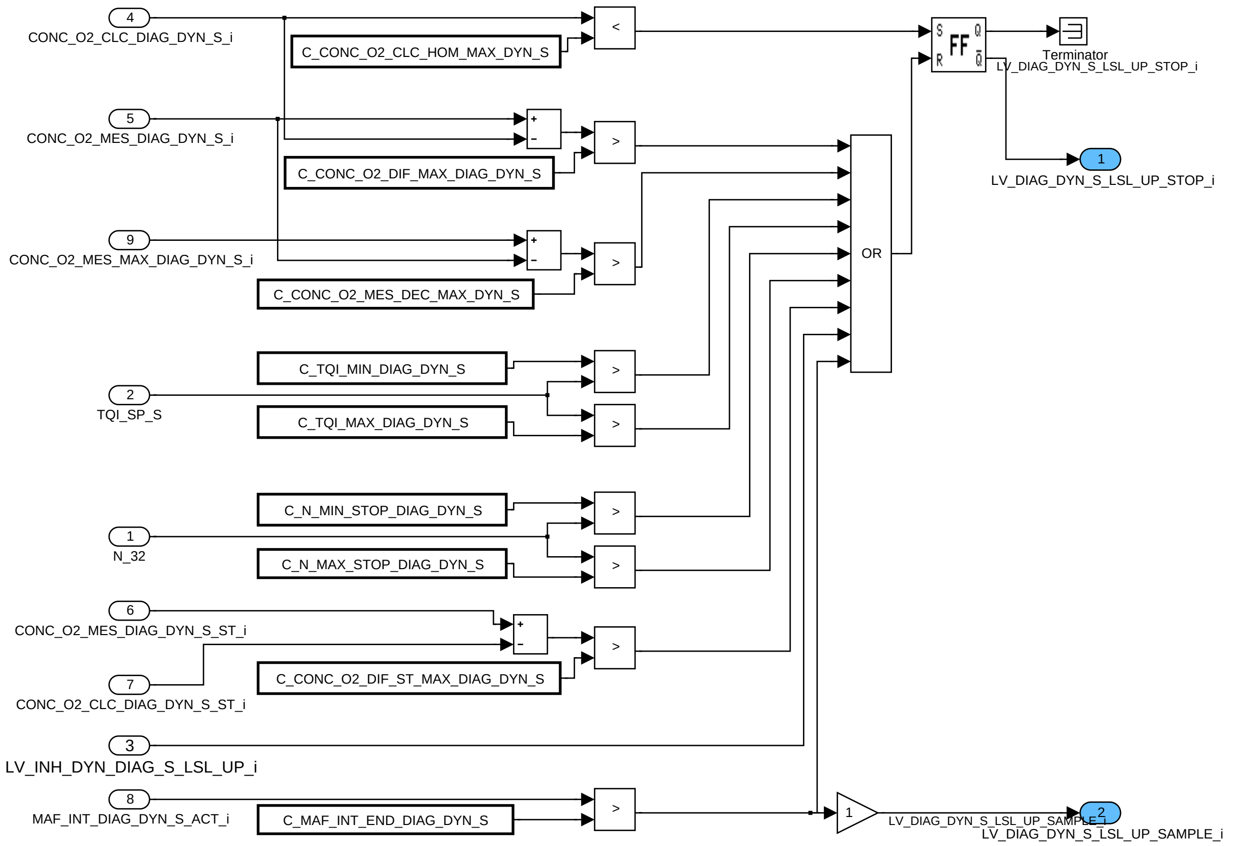


Figure 71 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK/ MON/ CTRL_FLAGS

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/MON_CHK/MON/PO2_CLC

This block calculates the O2 concentrations CONC_O2_MES_DIAG_DYN_S[i] from LAMB_LS_UP[i] and CONC_O2_CLC_DIAG_DYN_S[i] from LAMB_SP_FIL_S[i]. CONC_O2_MES_DIAG_DYN_S[i] can be scaled with a constant factor to improve diagnostic results for good sensors. The CONC_O2 values are used throughout the diagnosis because accuracy and dynamics are more similar across the different lambda ranges than for lambda itself.

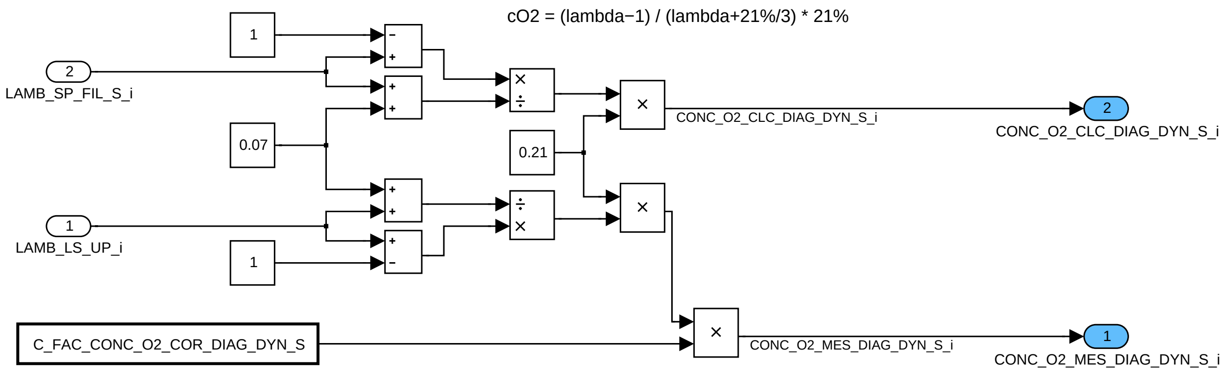


Figure 72 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK/ MON/ PO2_CLC

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/MON_CHK/MON/PO2_DIF_INT

Here, two integrals are produced which are used to calculate the diagnostic value. In CONC_O2_DIF_INT_DIAG_DYN_S[i] integrates the differences between modelled and measured O2 concentration, while CONC_O2_CLC_INT_DIAG_DYN_S[i] integrates the modelled oxygen concentration itself.

Both integrals start only after a calibrateable air mass has passed since the homogenous combustion mode was left. This can be used to avoid signal features shortly after the transition and to focus the diagnostic value on parts of the signal trace that show large and clear deviations between a modelled (fast) and a slow sensor. The integrals are stopped implicitly by calculating a result from them once after a certain air mass was consumed by the engine. After this, the function deactivates itself until the engine enters homogenous combustion mode again.

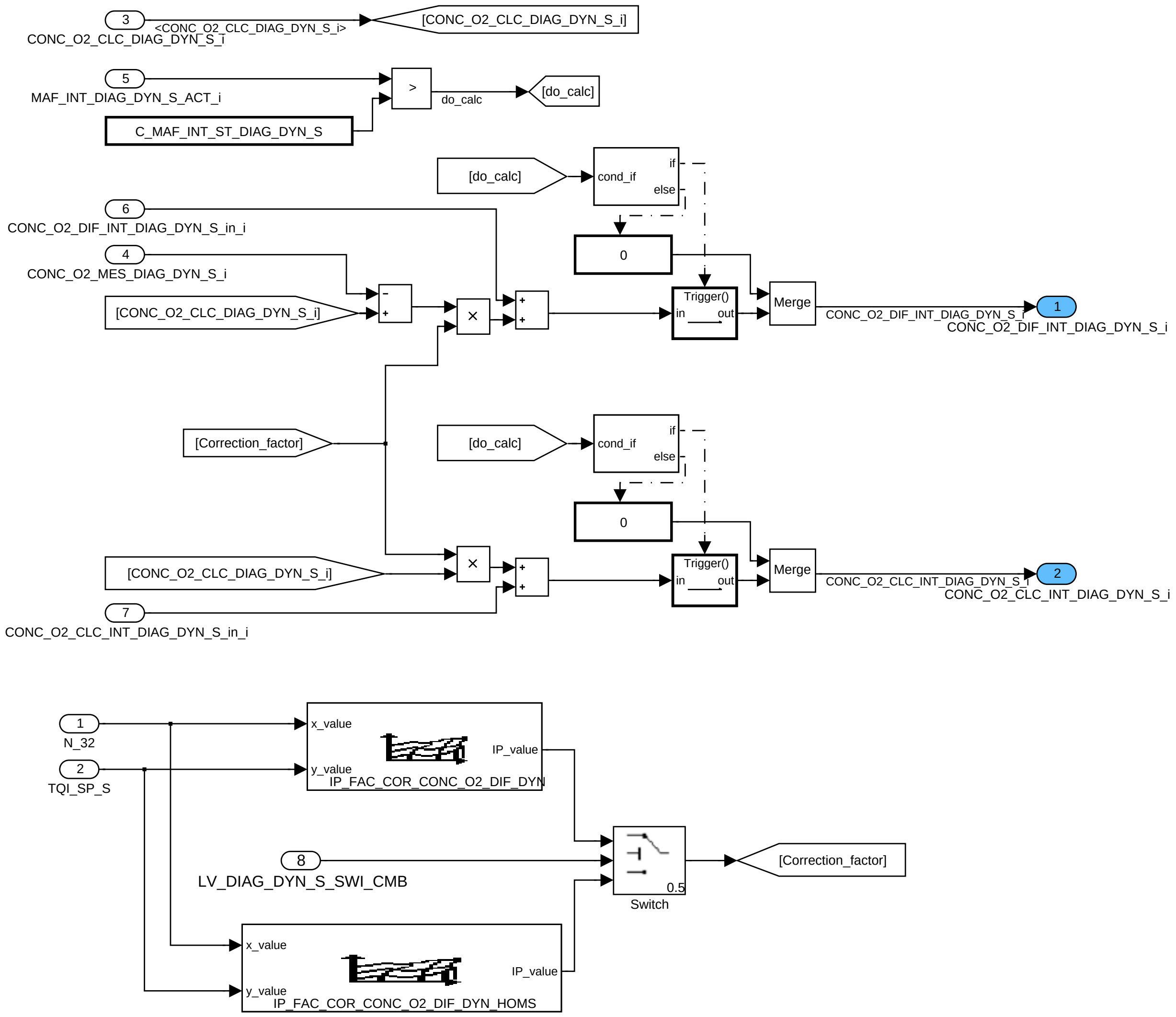


Figure 73 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK/ MON/ PO2_DIF_INT

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/MON_CHK/MON/PO2_EVAL

Here, a mass flow integral is calculated, which starts at the transition out of homogenous combustion and governs start and end points for the calculation of the diagnostic value. CONC_O2_MES_DIAG_DYN_S_ST[i] and CONC_O2_CLC_DIAG_DYN_S_ST[i] receive the values of CONC_O2_MES_DIAG_DYN_S[i] and CONC_O2_CLC_DIAG_DYN_S[i] at the time of the transition, they are used to calculate and validate the diagnostic result of the current transition. CONC_O2_MES_MAX_DIAG_DYN_S[i] is the highest O2 concentration found in the current diagnostic cycle, it is used in validation of the result.

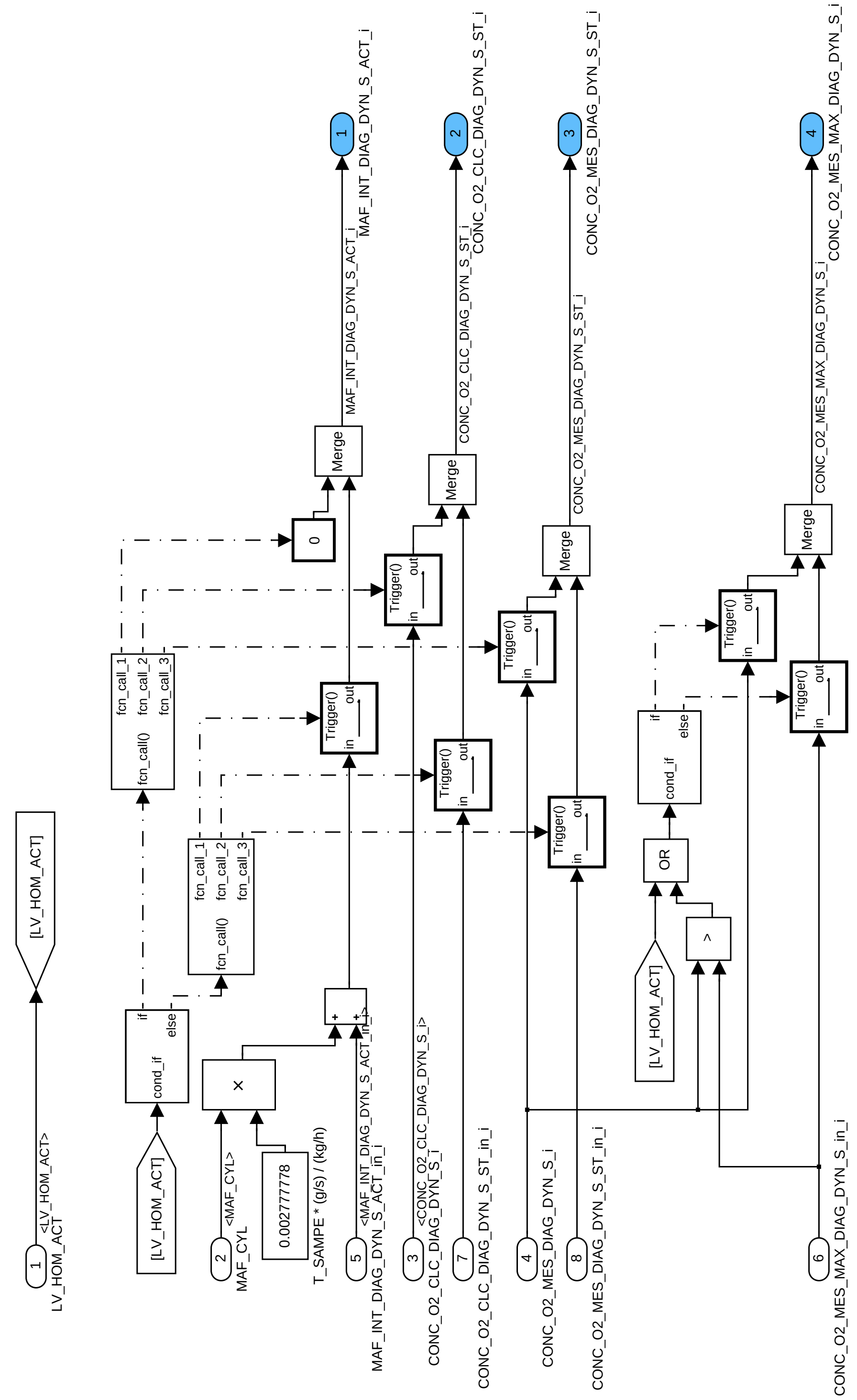


Figure 74 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK/ MON/ PO2_EVAL

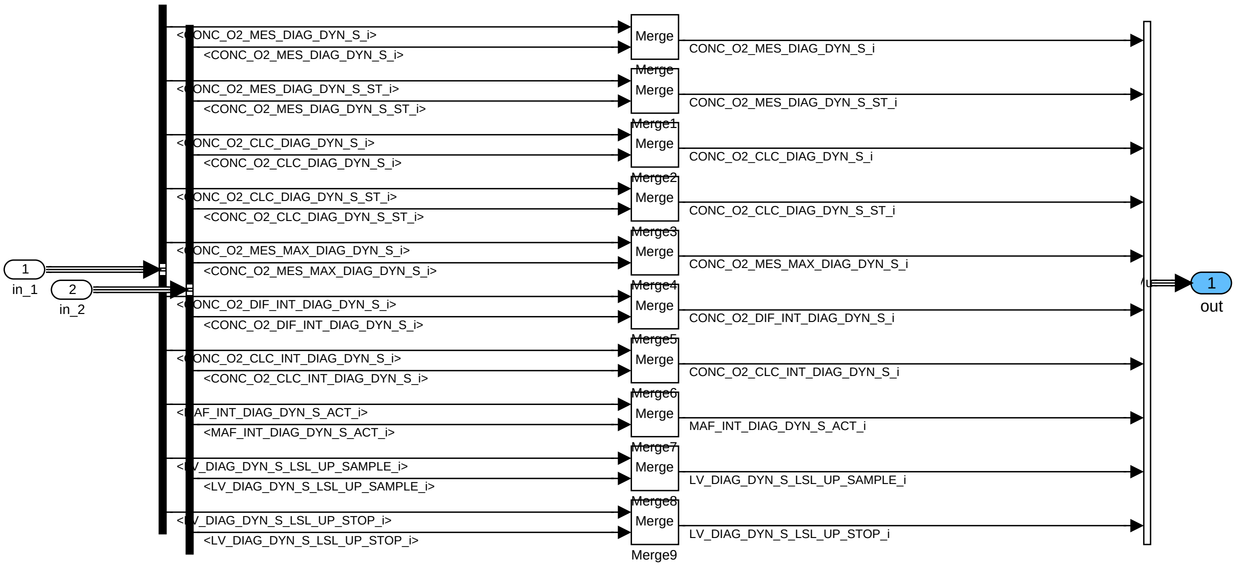


Figure 75 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ MON_CHK/ Merge

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/RESULT

The logic shown here triggers the calculation of a result once after a diagnostic cycle has completed successfully.

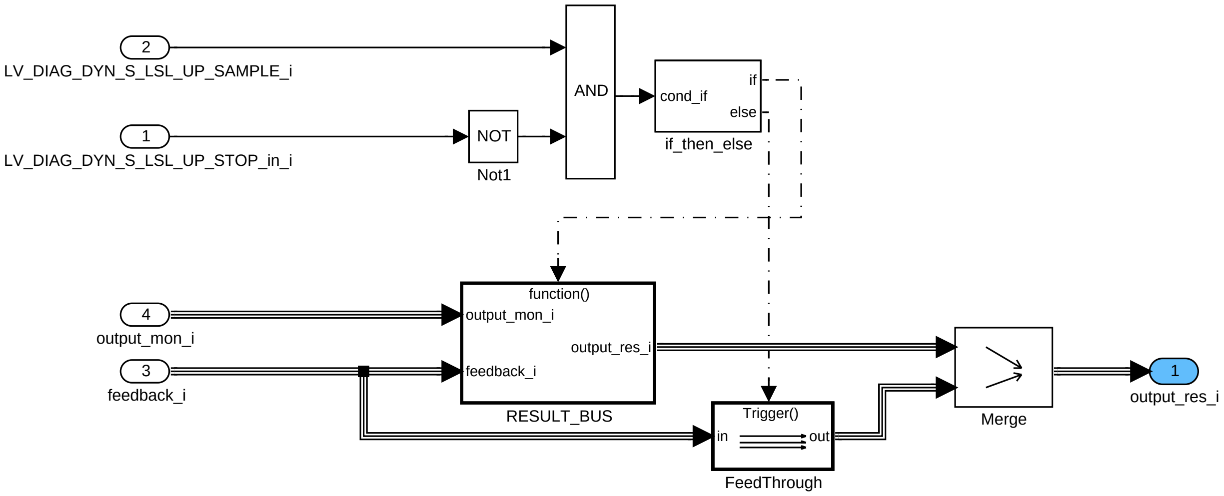


Figure 76 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ RESULT

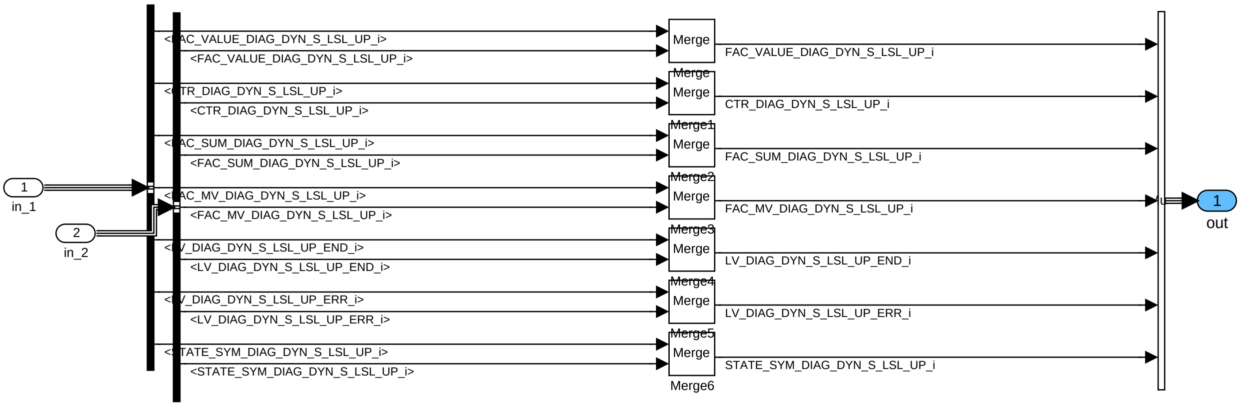


Figure 77 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ RESULT/ Merge

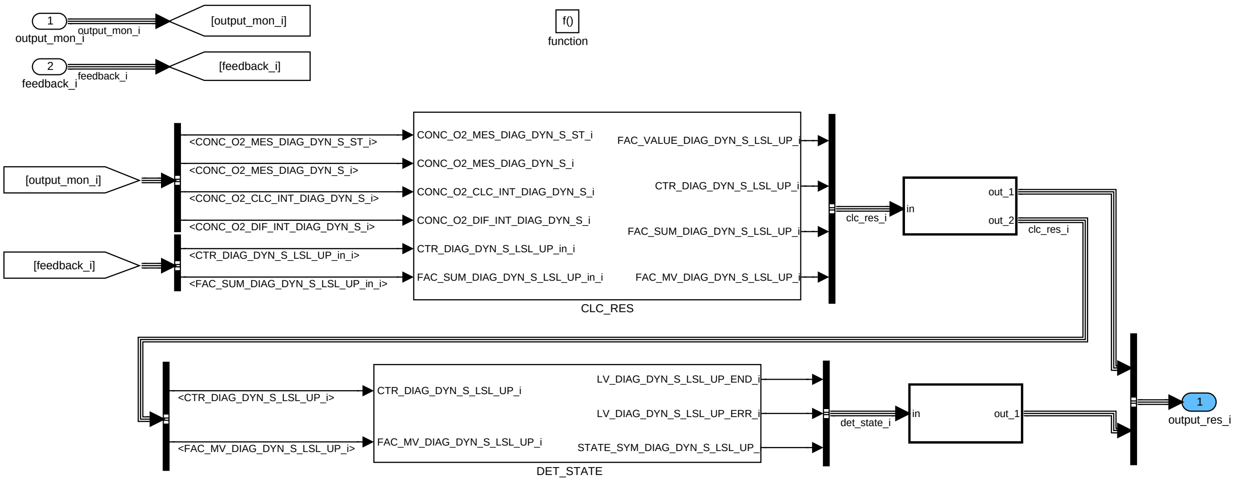


Figure 78 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ RESULT/ RESULT_BUS

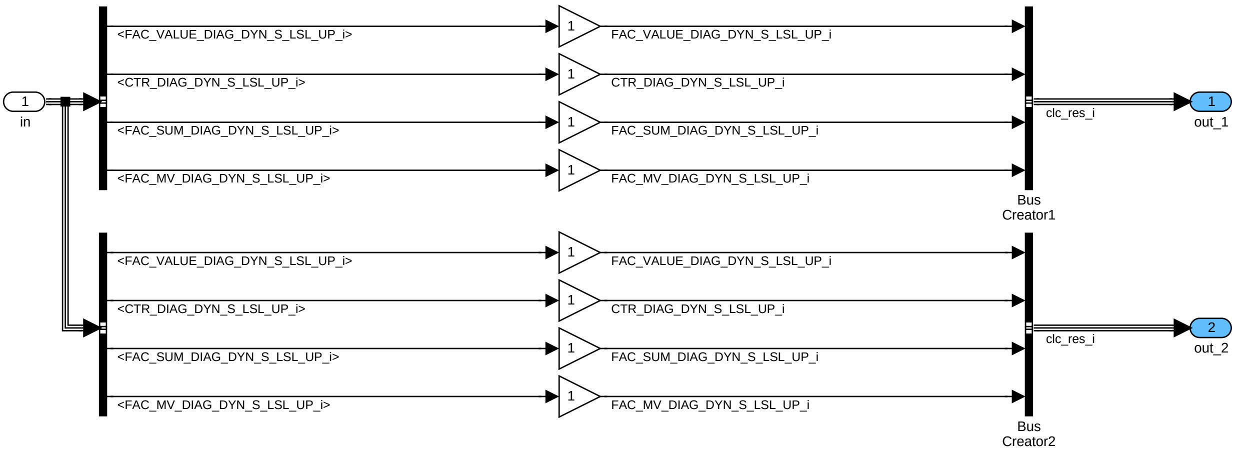


Figure 79 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ RESULT/ RESULT_BUS/ Branch

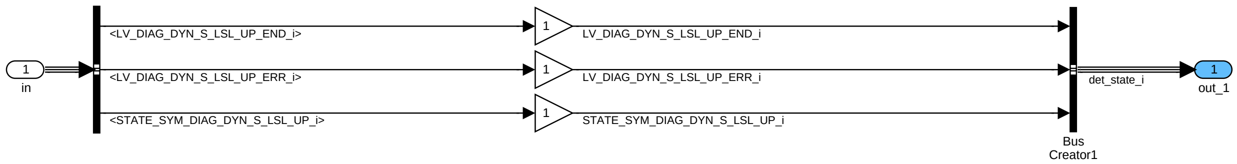


Figure 80 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ RESULT/ RESULT_BUS/ Branch1

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/RESULT/RESULT_BUS/CLC_RES

The result of one single diagnostic cycle is calculated in one of two ways and stored in FAC_VALUE_DIAG_DYN_S_LSL_UP[i]. It resembles the difference between modelled (i.e. "good") and actual sensor response to the lambda change in the transition from homogenous to lean combustion. If LC_DIAG_DYN_S_LSL_UP_MOD = 1, the difference is normalized to the signal change of the sensor and resembles the time constant of the sensor response. Otherwise it is normalized to the integrated signal change in the model, which makes the result more stable for slow sensors.

FAC_MV_DIAG_DYN_S_LSL_UP[i] is a cumulated average for the current driving cycle, whic is the basis for the decision whether the sensor is good or not.

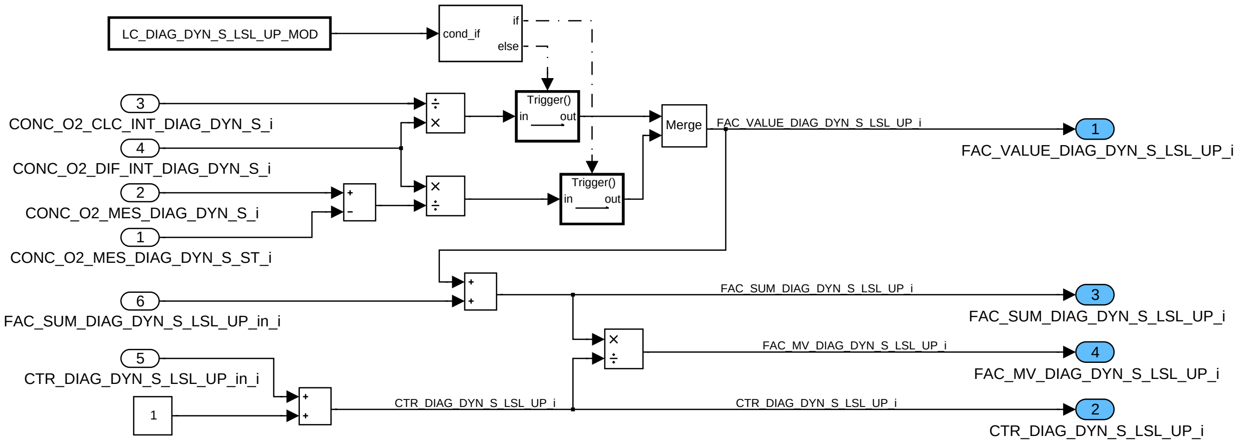


Figure 81 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ RESULT/ RESULT_BUS/ CLC_RES

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/RESULT/RESULT_BUS/DET_STATE

The block determines the diagnostic decision. With every cycle, the good/bad status in LV_DIAG_DYN_S_LSL_UP_ERR[i] and STATE_SYM_DIAG_DYN_S_LSL_UP[i] is updated, and if enough cycles have been completed, the end of diagnosis is indicated via LV_DIAG_DYN_S_LSL_UP_END[i]. This information is not transmitted to the error memory directly.

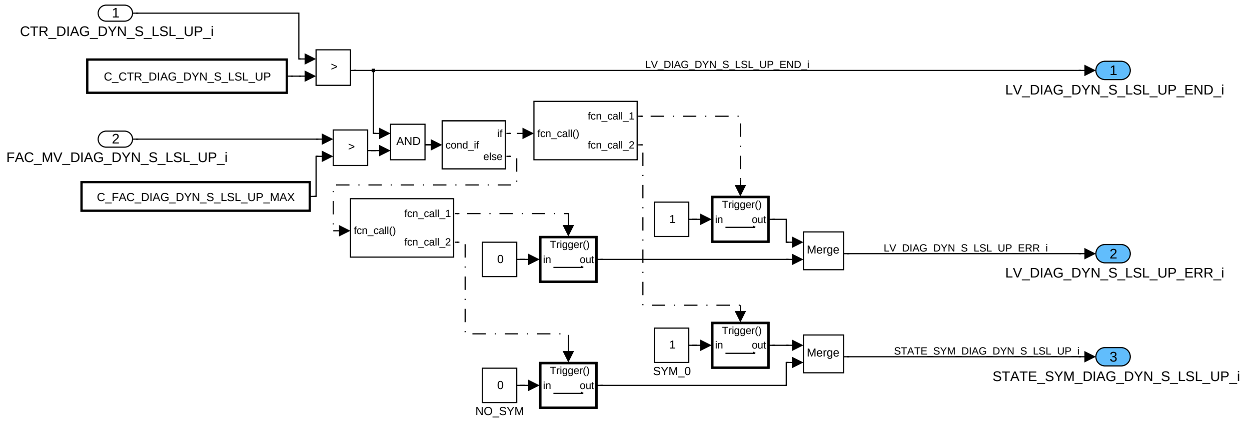


Figure 82 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ RESULT/ RESULT_BUS/ DET_STATE

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/VECTOR_FEEDBACK

This block selects required values from the last recurrence into a vector used in the functions for the current cylinder bank.

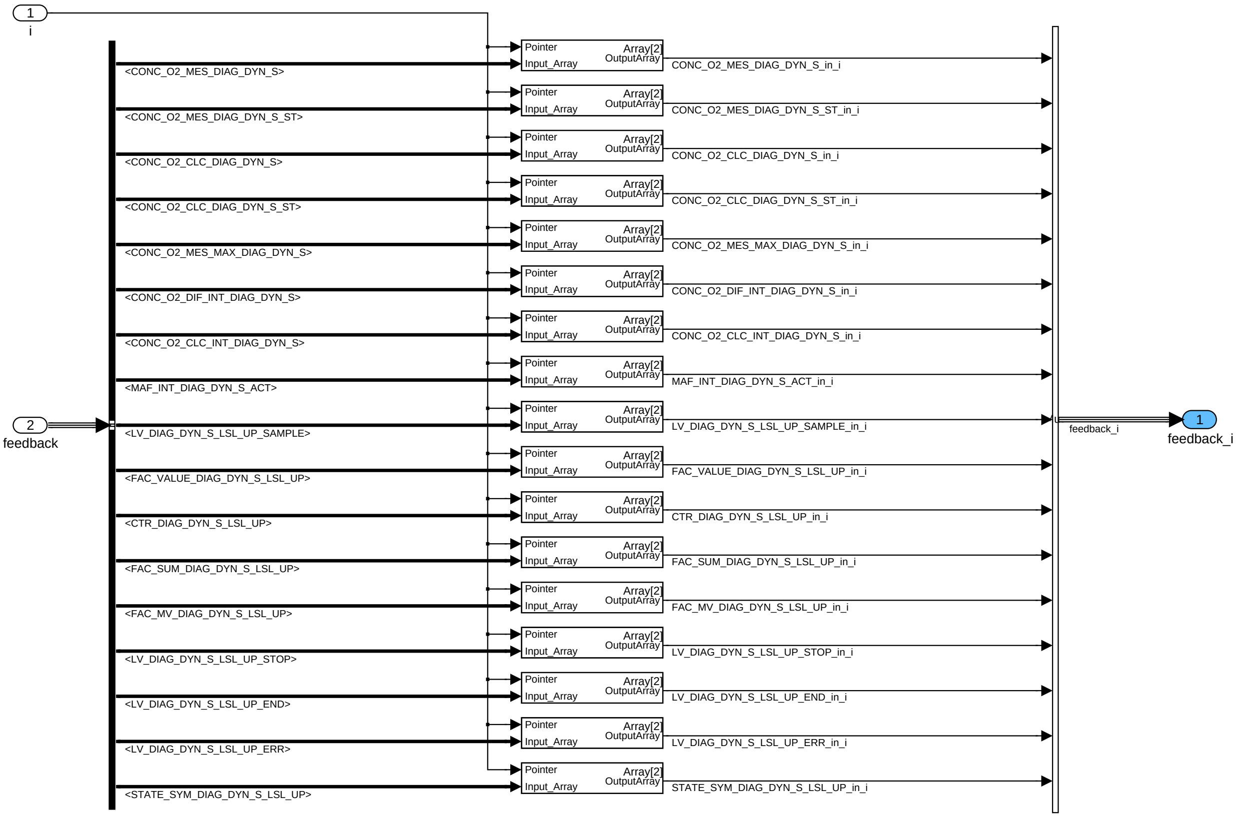


Figure 83 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ VECTOR_FEEDBACK

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/VECTOR_INPUT

The result of this block is a vector with the current input needed for the cylinder bank to be calculated.

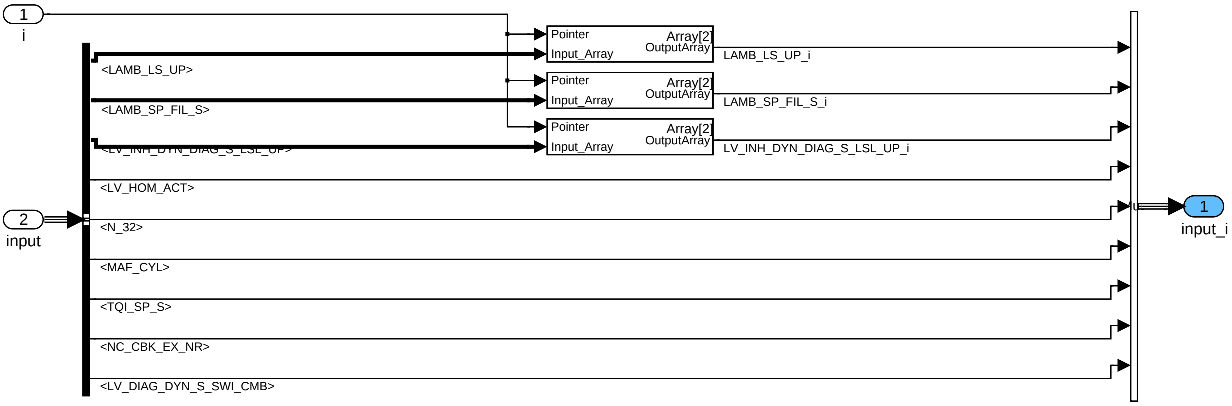


Figure 84 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ VECTOR_INPUT

EGCP_PLADGLSL6/OPERATE_10MS/OPERATE_10MS_CBK/VECTOR_OUTPUT

The output block assembles the model output vector from the results for the individual cylinder banks.

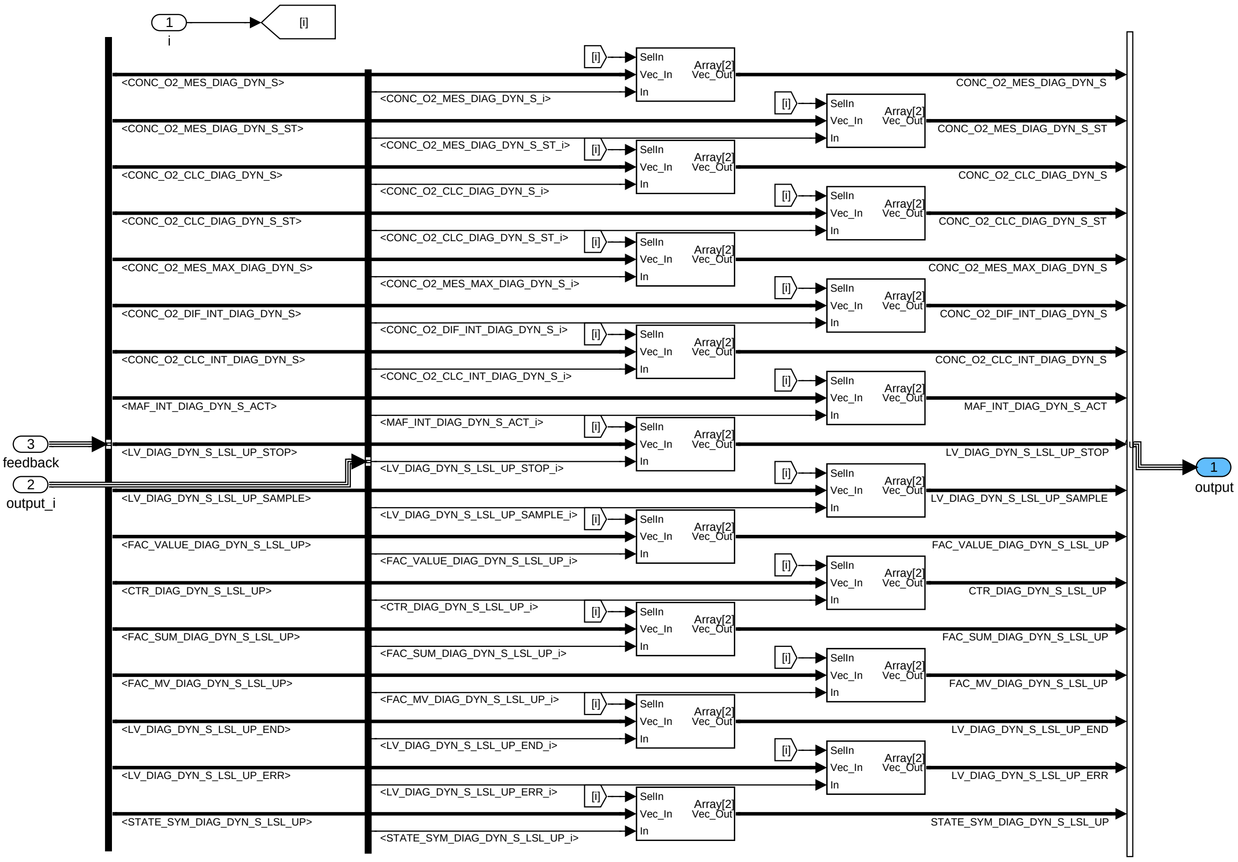


Figure 85 EGCP_PLADGLSL6/ operate_10ms/ operate_10ms_CBK/ VECTOR_OUTPUT

16.48.1.3 EGCP_PLADGLSL6/OUTPUT_MGR

The output manager merges the results of init and operate blocks into the functions' output vector.

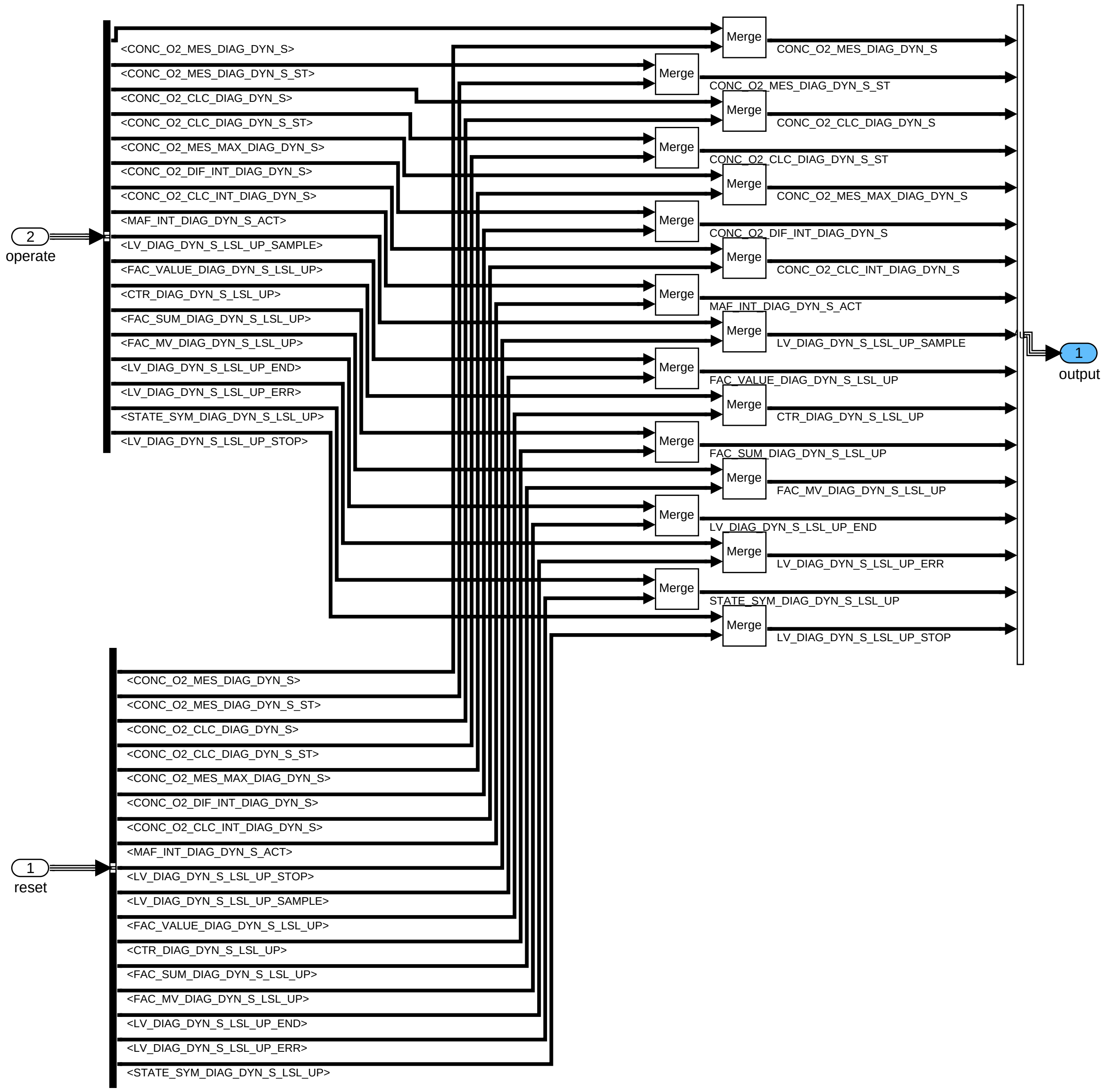


Figure 86 EGCP_PLADGLSL6/ output_mgr

16.49 Application Incidencies for “O2 sensor (lin, up) dyn diag after NOx trap regeneration"

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DYN_DIAG_S_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibition of linear lambda sensor dynamic diagnosis for stratified combustion					
LV_DIAG_DYN_S_SWI_CMB	O/V	0...1H	0...1	1	[-]
Logical variable to switch weighting factor depending on cumbustion mode					

Input data:

LV_DIAG_DYN_INH_LSL_UP[NC_CBK_EX_NR]	OPM_AV	LV_ST_END	LV_IGK
LV_END_DIAG_DYN_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_SLV_IVVT_IN	LV_ERR_MAP_TPS_PLAUS	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]
LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]			

16.49.1 General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

With C_DIAG_DYN_INH_CMB_TRAN the diagnosis can be interrupted due to switching of the combustion mode.

Interruption in case of:
Switching CBM from AFL to S and C_DIAG_DYN_INH_CMB_TRAN = 1
Switching CBM from S to AFL and C_DIAG_DYN_INH_CMB_TRAN = 2
Switching CBM from S to AFL and AFL to S and C_DIAG_DYN_INH_CMB_TRAN = 3

Application conditions:

Initialisation: At transition LV_IGK = 0 -> 1 or reset LV_INH_DYN_DIAG_S_LSL_UP[i] shall be set to 1.

Recurrence: 20ms.

Activation: -

Formula section:

```
If  LV_ST_END = 0                      or
    LV_IGK = 0                         or
    LV_ERR_SLV_IVVT_IN = 1             or
    LV_ERR_OFS_LSL_UP[i] = 1           or
    LV_ERR_CTL_LSL_UP[i] = 1           or
    LV_ERR_MAP_TPS_PLAUS = 1           or
```

```
        LV_DIAG_DYN_INH_LSL_UP[i] = 1         or
        LV_END_DIAG_DYN_VLD_LS_UP[i] = 1 or
        LC_INH_DYN_DIAG_S_LSL_UP[i] = 1       or
        (OPM_AV = 3h -->1h and
        C_DIAG_DYN_INH_CMB_TRAN =  1)         or
        (OPM_AV = 1h --> 3h and
        C_DIAG_DYN_INH_CMB_TRAN = 2)          or
        ((OPM_AV = 3h --> 1h or
        OPM_AV = 1h --> 3h) and
        C_DIAG_DYN_INH_CMB_TRAN = 3))

       % Inhibitions are the same as for homogenous dynamics diagnosis, plus inhibition of stratified mode
       % and manual deactivation.
Then
       LV_INH_DYN_DIAG_S_LSL_UP[i] = 1
Else
       LV_INH_DYN_DIAG_S_LSL_UP[i] = 0
Endif



If     OPM_AV = 3h
       % Engine runs in  mode AFL

Then
       LV_DIAG_DYN_S_SWI_CMB = 0
Else
       LV_DIAG_DYN_S_SWI_CMB = 1
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_DYN_DIAG_S_LSL_UP[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Manual inhibition of WRAF sensor dynamic diagnosis in stratified operation					
C_DIAG_DYN_INH_CMB_TRAN	1	0...3H	0...3	1	[-]
Inhibition of DYN_DIAG due to switching combustion mode					

16.50 O2 sensor (lin, up) PUC plausibility diag

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Debounced failure of the signal plausibility from the upstream sensor during PUC					
LV_END_DIAG_PUC_VLD_LS_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
End of diagnosis of signal plausibility from the upstream sensor during PUC					
ERR_SYM_PUC_VLD_LS_UP[NC_CBK_EX_NR]	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
symptom of signal plausibility from the upstream sensor during PUC					
LV_CDN_DIAG_PUC_VLD_LS_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Diagnosis condition for error symptom of signal plausibility of upstream sensor during PUC					
LV_DIAG_PUC_SYM_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Debounced failure symptom of the signal plausibility from the upstream sensor during PUC					
STATE_SYM_DIAG_PUC_LSL_UP[NC_CBK_EX_NR]	O/V	0H 1H 2H 4H	VLS_OK VLS_L VLS_H_OC VLS_AFS_OC	1	[-]
Symptom of the signal plausibility from the upstream sensor during PUC					
LV_DIAG_CDN_PUC_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Diagnosis condition					
LV_DIAG_PUC_END_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Result of diagnosis exist					
T_DIAG_END_PUC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Timer to have a diagnostic available					
T_MAX_DIAG_END_PUC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Time to set the END_DIAG bit					
T_DLY_DIAG_PUC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Delay Time to activate the diagnosis					
T_MAX_DLY_DIAG_PUC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Maximimum delay time to activate the diagnosis					
T_DIAG_SYM_PUC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Symptom debouncer timer					
T_MAX_DIAG_SYM_PUC_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Symptom debouncer time					
VLS_UP_DIAG_SAVE_PUC_LSL_UP[NC_CBK_EX_NR]	O/V/S	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Upstream sensor voltage - value used during the diagnosis and transmitted to the Scan Tool in mode06					

Input data:

VLS_UP_DIAG[NC_CBK_EX_NR]	LV_PUC	LV_IPLSL_VLD[NC_CBK_EX_NR]	TEG_DYN_LS_UP[NC_CBK_EX_NR]
LV_INH_DIAG_PUC_LSL_UP[NC_CBK_EX_NR]	LV_IGK	STATE_LSH_UP[NC_CBK_EX_NR]	NC_CBK_EX_NR
LV_DIAG_EOL_PUC_LSL_UP[NC_CBK_EX_NR]	NC_IDX_DIAG_PUC_VLD_LS_UP[NC_CBK_EX_NR]		

Import actions:

ACTION_ERRM_NoFilterSymptom(IN<IDX_DIAG>, IN< LV_CDN_DIAG_XX >, IN< ERR_SYM_XX >, IN< LV_ERR_SET_XX >, IN< LV_ERR_RST_XX >, IN< LV_END_DIAG_XX >, OUT< LV_ERR_XX >)
ACTION_ERRM_NoFilterReset(IN<IDX_DIAG>, OUT< LV_ERR_XX >)

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1, for exhaust cylinder bank 1
 i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1, for single exhaust cylinder bank.

Description:

This function shall test the plausibility of the upstream sensor signal during the Pull Fuel Cut-off phase (PUC). The variable LV_DIAG_SYM_PUC_LSL_UP[i] indicates an open circit on Vg or Vip (“VLS_AFS_OC) or on Pin Vrc (“VLS_H_OC”). This variable is considered only as a symptom and therefore shall not be transmitted to the Error Management.

The failure flag LV_ERR_PUC_VLD_LS_UP[i] is directly transfered to the error management. The failure is entered, if the lambda sensor voltage during fuel cut off is higher than the voltage in case of Open Circuit on Pin Vg or Vip but lower than the standard voltage.

The variable VLS_UP_DIAG_SAVE_PUC_LSL_UP[i] shall retain its value until an updated one is present. It should be reset only in case the Error Management memory is deleted.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset, LV_DIAG_EOL_PUC_LSL_UP[i] = 0 -> 1 or at clearing error memory reset all variables. The variable saved in the non-volatile memory shall be reset only in case the error memory is cleared.
ACTION_ERRM_NoFilterReset(IN< NC_IDX_DIAG_PUC_VLD_LS_UP[i]>, OUT<LV_ERR_PUC_VLD_LS_UP[i]>)

Recurrence:

This function shall be executed every 100 ms.

Activation:

```
IF    LV_PUC = 1                                              AND
      LV_IPLSL_VLD[i] = 1                                     AND
      TEG_DYN_LS_UP[i] > C_TEG_MIN_DIAG_PUC_LSL_UP            AND
      LV_INH_DIAG_PUC_LSL_UP[i] = 0                           AND
      STATE_LSH_UP[i] = LSH_POW_CTL
THEN
      LV_DIAG_CDN_PUC_LSL_UP[i] = 1
ELSE
      LV_DIAG_CDN_PUC_LSL_UP[i] = 0
      LV_CDN_DIAG_PUC_VLD_LS_UP[i] = 0
      IF       LV_DIAG_PUC_SYM_LSL_UP[i] = 0
      THEN  STATE_SYM_DIAG_PUC_LSL_UP[i] = “VLS_OK”
      ENDIF
      reset T_DLY_DIAG_PUC_LSL_UP[i]
ENDIF
```

Formula section:

```
IF    LV_DIAG_EOL_PUC_LSL_UP[i] = 0
THEN
      T_MAX_DLY_DIAG_PUC_LSL_UP = C_T_DLY_DIAG_PUC_LSL_UP
      T_MAX_DIAG_SYM_PUC_LSL_UP = C_T_DIAG_SYM_PUC_LSL_UP
      T_MAX_DIAG_END_PUC_LSL_UP = C_T_DIAG_END_PUC_LSL_UP
ELSE
      T_MAX_DLY_DIAG_PUC_LSL_UP = C_T_DLY_DIAG_EOL_PUC_LSL_UP
      T_MAX_DIAG_SYM_PUC_LSL_UP = C_T_DIAG_SYM_EOL_PUC_LSL_UP
      T_MAX_DIAG_END_PUC_LSL_UP = C_T_DIAG_END_EOL_PUC_LSL_UP
ENDIF
```

```
IF(1)   T_DLY_DIAG_PUC_LSL_UP[i] >  T_MAX_DLY_DIAG_PUC_LSL_UP
THEN(1)    LV_CDN_DIAG_PUC_VLD_LS_UP[i] = 1
      IF(2)    VLS_UP_DIAG[i] < C_VLS_UP_DIAG_AFS_PUC_LSL_UP
      THEN(2)
               STATE_SYM_DIAG_PUC_LSL_UP[i] = “VLS_AFS_OC”
      ELSE(2)
               IF(3)   VLS_UP_DIAG[i]  < C_VLS_UP_DIAG_MIN_PUC_LSL_UP
               THEN(3)
                       STATE_SYM_DIAG_PUC_LSL_UP[i] = “VLS_L”
               ELSE(3)
                       IF(4)    VLS_UP_DIAG[i] > C_VLS_UP_DIAG_MAX_PUC_LSL_UP
                       THEN(4)
                               STATE_SYM_DIAG_PUC_LSL_UP[i] = “VLS_H_OC”
                       ELSE(4)
```

```
                        STATE_SYM_DIAG_PUC_LSL_UP[i] = “VLS_OK”
                ENDIF(4)
            ENDIF(3)
    ENDIF(2)

    IF(2a)  STATE_SYM_DIAG_PUC_LSL_UP[i] = “VLS_AFS_OC”
            or      STATE_SYM_DIAG_PUC_LSL_UP[i] = “VLS_H_OC”
    THEN(2a)
            IF(3a) T_DIAG_SYM_PUC_LSL_UP[i] >  T_MAX_DIAG_SYM_PUC_LSL_UP
            THEN(3a)
                    LV_DIAG_PUC_SYM_LSL_UP[i] = 1
                    LV_DIAG_PUC_END_LSL_UP[i] = 1
                    LV_END_DIAG_PUC_VLD_LS_UP[i] = 1
                    VLS_UP_DIAG_SAVE_PUC_LSL_UP[i] = VLS_UP_DIAG[i]
            ELSE(3a)
                    Increment T_DIAG_SYM_PUC_LSL_UP[i]
            ENDIF(3a)
    ELSE(2a)
                    IF(3a1) STATE_SYM_DIAG_PUC_LSL_UP[i] = “VLS_L”
                    THEN(3a1)
                    IF(4a)  T_DIAG_SYM_PUC_LSL_UP[i] >
                            T_MAX_DIAG_SYM_PUC_LSL_UP
                    THEN(4a)
                                    ERR_SYM_PUC_VLD_LS_UP[i] = “SYM_0”
                            LV_ERR_PUC_VLD_LS_UP[i] = 1
                            LV_END_DIAG_PUC_VLD_LS_UP[i] = 1
                            LV_DIAG_PUC_END_LSL_UP[i] = 1
                            VLS_UP_DIAG_SAVE_PUC_LSL_UP[i] = VLS_UP_DIAG[i]
                    ELSE(4a)
                            Increment T_DIAG_SYM_PUC_LSL_UP[i]
                    ENDIF(4a)
            ELSE(3a1) T_DIAG_SYM_PUC_LSL_UP[i] = 0
            ENDIF(3a1)
    ENDIF(2a)

    %Minimum time to have a valid diagnostic available

    IF(2b)     LV_DIAG_PUC_END_LSL_UP[i]  = 0
    AND(2b) LV_END_DIAG_PUC_VLD_LS_UP[i] = 0
    THEN(2b)
            IF(3b) T_DIAG_END_PUC_LSL_UP[i] > T_MAX_DIAG_END_PUC_LSL_UP
            THEN(3b)
                    LV_DIAG_PUC_END_LSL_UP[i] = 1
                    LV_END_DIAG_PUC_VLD_LS_UP[i] = 1
                    VLS_UP_DIAG_SAVE_PUC_LSL_UP[i] = VLS_UP_DIAG[i]
            ELSE(3b)
                    Increment T_DIAG_END_PUC_LSL_UP[i]
            ENDIF(3b)
    ENDIF(2b)
ELSE(1)
    Increment T_DLY_DIAG_PUC_LSL_UP[i]
    LV_CDN_DIAG_PUC_VLD_LS_UP[i] = 0
ENDIF(1)
```

For failure and error management treatment the anti-bounce mechanism is called :

% if LV_ERR_XX=1 then LV_ERR_SET=1 or if LV_ERR_XX=0 then LV_ERR_RST=1

ACTION_ERRM_NoFilterSymptom(IN<NC_IDX_DIAG_PUC_VLD_LS_UP[i]>, IN<LV_CDN_DIAG_PUC_VLD_LS_UP[i]>, IN< ERR_SYM_PUC_VLD_LS_UP[i] >, IN< LV_ERR_SET_PUC_VLD_LS_UP[i] >, IN<LV_ERR_RST_PUC_VLD_LS_UP[i]>, IN<LV_END_DIAG_PUC_VLD_LS_UP[i]>, OUT<LV_ERR_PUC_VLD_LS_UP[i]>)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DIAG_SYM_PUC_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time filter to detect a symptom					
C_T_DIAG_SYM_EOL_PUC_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time filter to detect a symptom (EOL test)					
C_T_DLY_DIAG_PUC_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Waiting time after beginning of PUC to test sensor signal					
C_T_DLY_DIAG_EOL_PUC_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Waiting time after beginning of PUC to test sensor signal (EOL test)					
C_VLS_UP_DIAG_AFS_PUC_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Upstream sensor output voltage in PUC corresponding to Open Circuit on WRAF Sensor					
C_VLS_UP_DIAG_MIN_PUC_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Minimum admissible upstream sensor output voltage in PUC					
C_VLS_UP_DIAG_MAX_PUC_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Upstream sensor output voltage in PUC corresponding to Open Circuit on WRAF Sensor					
C_TEG_MIN_DIAG_PUC_LSL_UP	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Minimum admissible exhaust gas temperature in PUC					
C_T_DIAG_END_PUC_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time to have a diagnostic available					
C_T_DIAG_END_EOL_PUC_LSL_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time to have a diagnostic available (EOL test)					

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
PUC_VLD_LS_UP[NC_CBK_EX_NR]	low voltage during fuel cut off	SYM_0	NO

16.51 Application Incidences for "O2 sensor (lin, up) PUC plausibility diag"

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_RBM_PUC_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of ¿Monitoring Upstream Sensor Signal during PUC¿ by other failures					
LV_INH_DIAG_PUC_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "Monitoring Upstream Sensor Signal during PUC"					
LV_DIAG_EOL_PUC_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Activation of short trip for Monitoring Upstream Sensor during PUC					

Input data:

LV_IGK	STATE_ERR_IV	LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]
LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]	LV_ERR_CHG_LS_UP	LV_ERR_EL_CPS	LV_ERR_DIAGCPS
LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_SAV	LV_MIS_STATE_B	LV_ERR_FTL_MIN
LV_ERR_TCO	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	LV_CDN_VB_OBD2	NC_CBK_EX_NR
LV_ERR_MAP_DIP_SHIFT	LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_MAF
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]
LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV
LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS	

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1, for exhaust cylinder bank 1
 i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation: at every transition LV_IGK = 0->1or reset all variables shall be reset to 0.

at every transition LV_IGK = 1->0

LV_INH_DIAG_PUC_LSL_UP[NC_CBK_EX_NR] = 1

Recurrence: the functions shall be carried out once every 1s.

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

This function has to be deactivated, if at least one of the following errors / special modes is active.

```
IF          STATE_ERR_IV > 0                                   OR
            LV_ERR_LS_UP[i] = 1                                OR
            LV_ERR_DELTA_I_LAM[i] = 1                          OR
            LV_ERR_VLS_DOWN_DIF[i] = 1                         OR
            LV_ERR_CHG_LS_UP = 1                               OR
            LV_ERR_EL_CPS = 1                                  OR
            LV_ERR_DIAGCPS = 1                                 OR
            LV_ERR_FSD[i] = 1                                  OR
            LV_ERR_SAV = 1                                     OR
            LV_MIS_STATE_B = 1                                 OR
            LV_ERR_FTL_MIN = 1                                 OR
            LV_ERR_TCO = 1                                     OR
            LV_ERR_MAF = 1                                     OR
            LV_ERR_MAP_DIP_SENS = 1                            OR
            LV_ERR_MAP_DIP_PLAUS = 1                           OR
            LV_ERR_MAP_DIP_SHIFT = 1                           OR
            LV_ERR_FSD_LAM_LIM [NC_CBK_EX_NR] = 1              OR
            LV_ERR_FUP = 1                                     OR
            LV_ERR_FUP_MFP_PLAUS = 1                           OR
            LV_ERR_H_PRS_SYS = 1                               OR
            LV_ERR_VCV = 1                                     OR
            LV_ERR_FUP_ORNG = 1                                OR
            LV_ERR_FUP_ST = 1                                  OR
            LV_ERR_TTIP_MES_LSH_UP [NC_CBK_EX_NR] = 1          OR
            LV_ERR_OFS_LSL_UP[i] = 1                           OR
            LV_ERR_CTL_LSL_UP[i] = 1                           OR
            LV_ERR_MAP_TPS_PLAUS = 1

  THEN    LV_INH_DIAG_RBM_PUC_LSL_UP[i]  =  1
  ELSE    LV_INH_DIAG_RBM_PUC_LSL_UP[i]  =  0
ENDIF

IF        LV_LS_UP_OBD_2_MAN_DEAC[i]  = 1                  OR
          LV_CDN_VB_OBD2 = 0                               OR
          LV_INH_DIAG_RBM_PUC_LSL_UP[i] = 1

    THEN  LV_INH_DIAG_PUC_LSL_UP[i]  =  1
    ELSE  LV_INH_DIAG_PUC_LSL_UP[i]  =  0
ENDIF
```

16.51.1 Activation of short trip for Monitoring Upstream Sensor during PUC

FUNCTION DESCRIPTION:

General information:

Not yet defined

Application conditions:

Initialisation: LV_DIAG_EOL_PUC_LSL_UP[i] = 0 *(never changed)*

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
WRAF Sensor diagnosis during PUC	*Sym_0*	*0*	*see table of failires*	*see table of failires*	---		See Table of failure
PUC_VLD_LS_UP[i]							

16.52 Upstream oxygen sensor heater OBDII monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_OBD_CDN_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that conditions for diagnosis met					
STATE_SYM_OBD_LSL_LSH_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H	NO_SYM TTIP_ERR READY_ERR TTIP_MES_ERR	1	[-]
Variable indicating status of each symptom, updated each recurrence & at diagnosis completion					
LV_DIAG_OBD_SYM_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that fault is present and has been debounced					
LV_DIAG_OBD_END_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that diagnosis has been completed					
LV_LAMB_PLS_REQ_DIAG_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag requesting exaggerated forced stimulation					
TCC_END_DIAG_OBD_LSH_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Counter for end flag determination of TTIP_MES diagnosis					
TCC_ERR_OBD_LSH_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Number of diagnosis cycles in which temperature deviation exceeding limits present					
TTIP_MES_LS_UP_OBD_LSH_UP[NC_CBK_EX_NR]	V/O/S	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Most current oxygen sensor temperature (Mode 6 information)					
LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating the validation of correspondent symptom (transmitted to the error management)					
ERR_SYM_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom to be transmitted to the error management					
LV_CDN_DIAG_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	-	0...1H	0...1	1	[-]
Boolean flag indicating condition of correspondent diagnosis					
LV_END_DIAG_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating end of correspondent diagnosis					
LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating the validation of general tip temperature plausibility symptom					
ERR_SYM_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom for general tip temperature plausibility					
LV_CDN_DIAG_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating availability of general tip temperature plausibility diagnosis					
LV_END_DIAG_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating end of general tip temperature plausibility diagnosis					

Input data:

LV_ST_END	LV_INH_DIAG_OBD_LSH_UP[NC_CBK_EX_NR]	LSHPWM_UP[NC_CBK_EX_NR]	LV_VB_CDN_OBD_2
STATE_LSH_UP[NC_CBK_EX_NR]	LV_TTIP_MES_VLD_LS_UP[NC_CBK_EX_NR]	TTIP_MES_LS_UP[NC_CBK_EX_NR]	LV_IGK
T_POW_RISE_LSH_UP[NC_CBK_EX_NR]	LV_TEMP_DEW_LS_UP[NC_CBK_EX_NR]		LV_LS_UP_READY[NC_CBK_EX_NR]
TEMP_INI_LS_UP[NC_CBK_EX_NR]	LV_V_REF_VLD_R_IT_LS_UP[NC_CBK_EX_NR]	NC_CBK_EX_NR	TEG_DYN_LS_UP[NC_CBK_EX_NR]
LV_END_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	STATE_DIAG_OC_LSL_UP[NC_CBK_EX_NR]	C_TTIP_THD_IPLSL_ACT_LS_UP

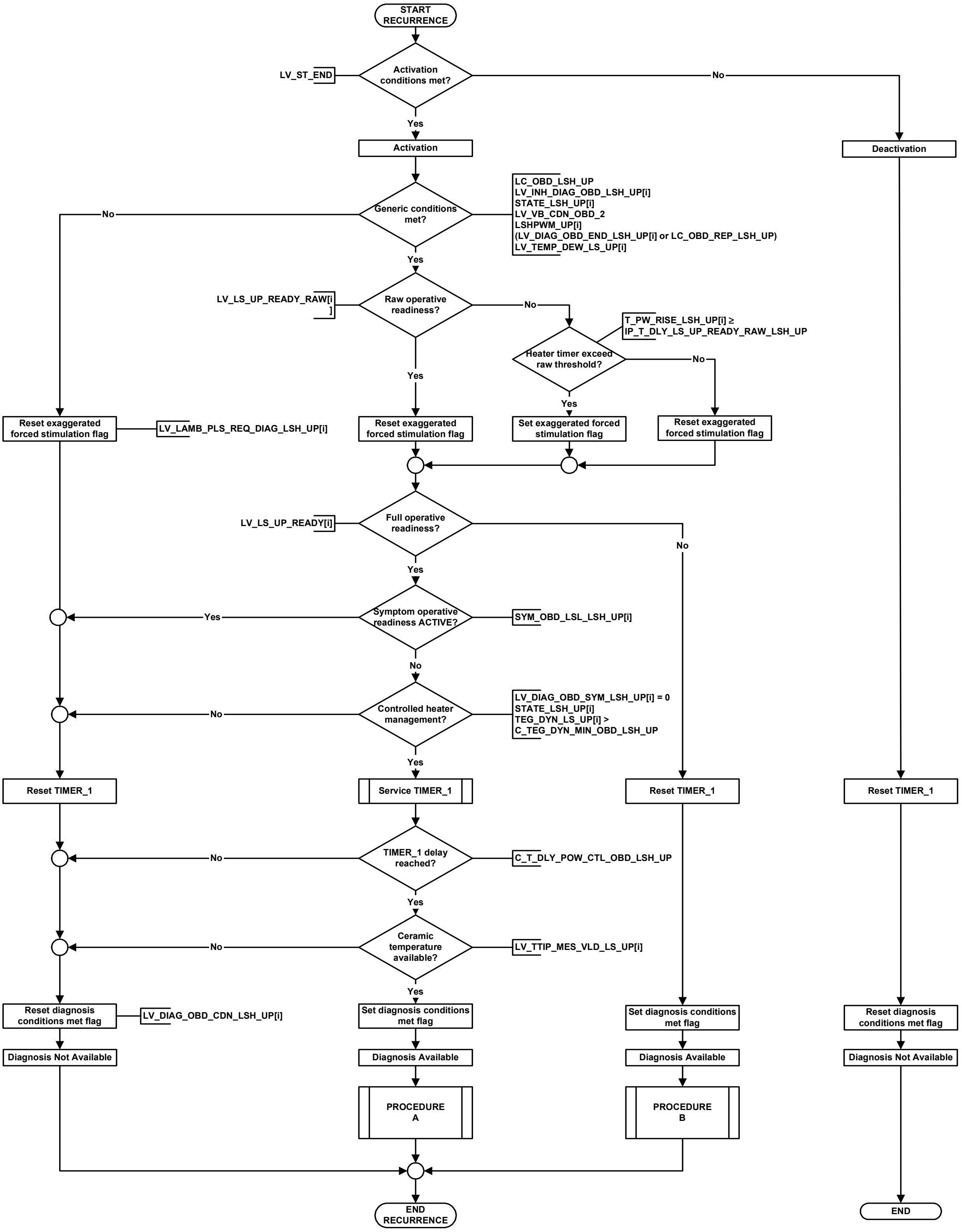
FUNCTION DESCRIPTION:

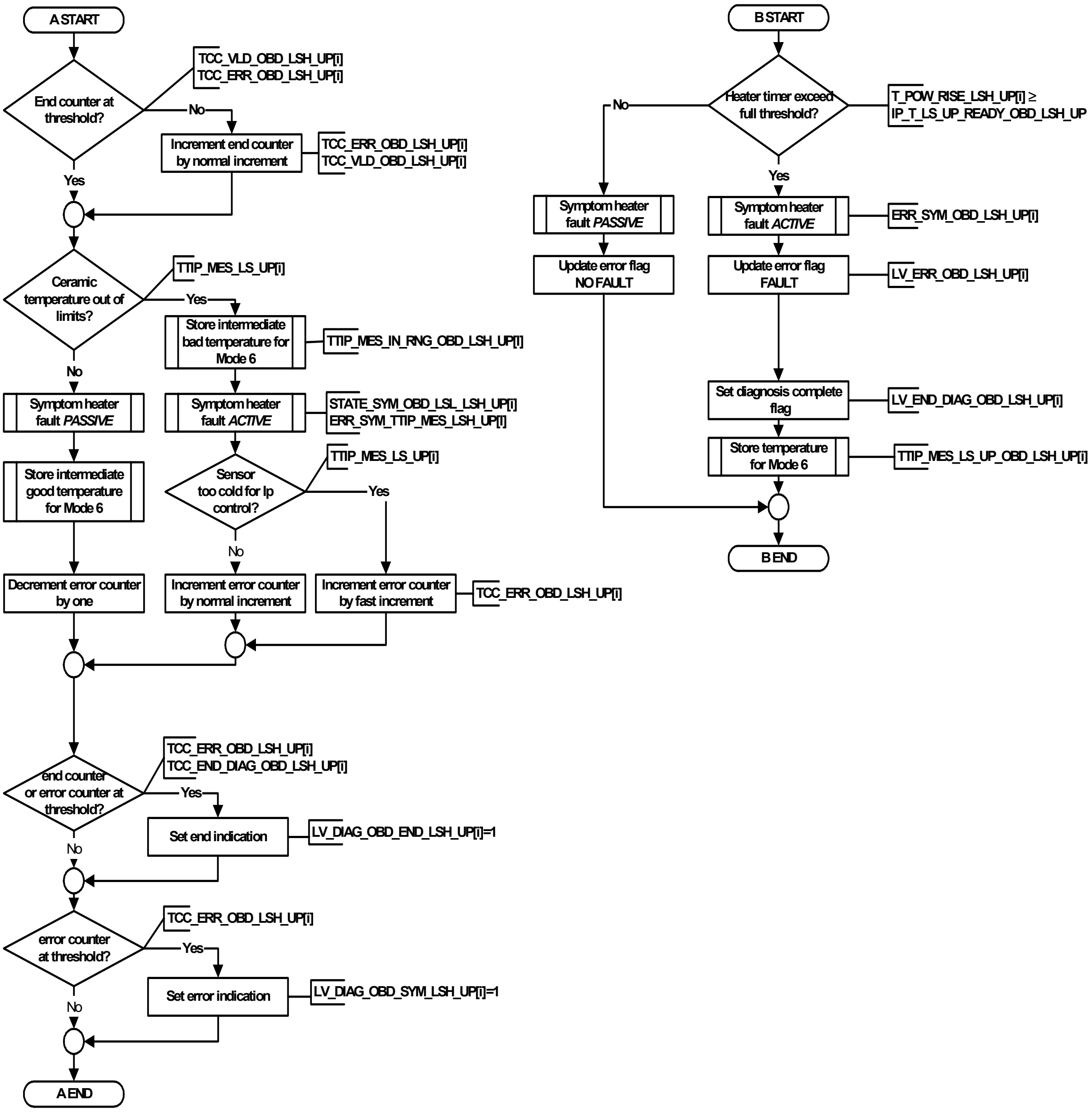
General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1, for exhaust cylinder bank 1
 i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1, for single exhaust cylinder bank.

This function shall facilitate the detection of oxygen sensor heater failures that would lead to an increase in emissions beyond the thresholds stated in the appropriate regulations. The diagnosis shall be carried out by determining whether the operative readiness of the sensor exceeds a time threshold or whether the measured oxygen sensor ceramic temperature exceeds or falls below set bounds over a number of measurement cycles, defined to be the actions carried out once per recurrence. The latter evaluation over the diagnosis cycle, defined to be after the completion of the set number of monitoring cycles, shall be carried out statistically.

Signal flow diagram:





Description:

Deviations in the oxygen sensor ceramic temperature or the oxygen sensor not being operatively ready in a timely manner can lead to an increase in emissions above the applicable standards or prevent the sensor signal from being used as a diagnostic system monitoring device.

The deviations may occur due to, for example, ageing of the heater element, defective wiring, increased heater circuit connector contact resistance, defective heater driver etc.

The hereafter mentioned diagnosis strategy is based on the detection via one of two methods, both cases being emissions relevant. Firstly the time to operative readiness shall be checked and secondly a statistical evaluation of the oxygen sensor ceramic temperature over a pre-defined number of monitoring cycles shall be carried out, whereby if the temperature should fall out of pre-set bounds a calibrateable number of times during the complete diagnosis cycle, a fault shall be detected as being present.

The oxygen sensor ceramic temperature shall be obtained indirectly via the measured internal resistance of the sensor.

The functionality of the diagnosis may be described in further detail below:

The diagnosis shall be carried out at least once per driving cycle or may be permitted to repeat, dependent on the status of flag LC_OBD_REP_LSH_UP. It shall be clear that only the temperature deviation detection method can repeat as the sensor warm-up phase is usually only carried out once per driving cycle. The diagnosis shall require a number of monitoring cycles to be carried out to permit statistical evaluation of the results. The function may set

The function shall be activated and deactivated dependent on the engine start state, denoted by flag LV_ST_END. Should the function be deactivated, the diagnosis conditions flags LV_DIAG_OBD_CDN_LSH_UP[i], LV_CDN_DIAG_OBD_VLD_LSH_UP[i] and LV_CDN_:DIAG_TTIP_MES_LSH_UP[i] and the SW internal timer *TIMER_1* shall be reset and the diagnosis shall be considered to be “Not Available”.

Additional application conditions shall be met prior to permitting monitoring cycles to take place:

- LC_OBD_LSH_UP shall permit the diagnosis to be enabled or disabled from within the calibration system.
- LV_INH_DIAG_OBD_LSH_UP[i] shall facilitate the interfacing of project specific application conditions.
- STATE_LSH_UP[i] shall neither be in LSH_OFF nor LSH_VB_PROT. See following description for explanation.
- VB shall remain within bounds as indicated by LV_VB_CDN_OBD_2. This shall provide the facility to ensure that the battery voltage is sufficient to carry out OBD II diagnosis.
- LSHPWM_UP[i] shall remain above or equal to C_LSHPWM_MIN_LSH_UP. This shall provide the facility to limit the performing of the diagnosis in a region where the sensor temperature can be controlled. No upper limit is given as should the heater become defective then the closed loop heater voltage controller will tend to drive the LSHPWM_UP[i] to its maximum limit.
- The condition (LV_DIAG_OBD_END_LSH_UP[i] = 0 **or** LC_OBD_REP_LSH_UP = 1) shall permit the diagnosis to be carried out once only per driving cycle (LC_OBD_REP_LSH_UP = 0) or continuously over the whole driving cycle (LC_OBD_REP_LSH_UP = 1).
- The dew point shall have passed as denoted by LV_TEMP_DEW_LS_UP[i] = 1.

The function shall determine whether the full operative readiness LV_LS_UP_READY[i] has been detected. If not, then the time to operative readiness detection method shall be carried out. Internal TIMER_1 shall be reset, the diagnosis conditions flag LV_DIAG_OBD_CDN_LSH_UP[i] shall be set and the diagnosis shall be declared as being “Available”. Timer T_POW_RISE_LSH_UP[i] shall be checked against threshold IP_T_DLY_LS_UP_READY_OBD_LSH_UP. Should it equal or exceed the threshold then the operative readiness has not been detected in a timely manner and the symptoms shall be set to READY_ERR, fault flag LV_DIAG_OBD_SYM_LSH_UP[i] and the error flags shall be set and the end of diagnosis flags shall be set. If T_POW_RISE_LSH_UP[i] does not equal or exceed the threshold then all fault indicators shall be reset.

In order to meet Mode 6 diagnosis requirements, the current oxygen sensor ceramic temperature shall be stored in TTIP_MES_LS_UP_OBD_LSH_UP[i]. (This is used for the detection of the full operative readiness and hence may be used to meet the requirements).

Once the full operative readiness has been detected, further conditions shall be met prior to determining whether the oxygen sensor temperature falls within bounds, i.e. detection method two.

The function shall determine whether a time to operative readiness fault has been detected by observing STATE_SYM_OBD_LSL_LSH_UP[i]. Provided no error symptom is active, the function shall then wait for STATE_LSH_UP[i] to reach LSH_POW_CTL. Additionally the exhaust gas temperature shall exceed the limit C_TEG_DYN_MIN_OBD_LSH_UP to ensure that in long PUC phases sensor cooling does not lead to false error detection. After this condition has been met, the function shall await the delay C_T_DLY_POW_CTL_OBD_LSH_UP. The delay provides the heater controller sufficient time to regulate the ceramic temperature to the set-point. Furthermore, this condition excludes states LSH_VB_PROT & LSH_TEMP_PROT. In both of these states, the heater power is reduced to prevent damage to the sensor. Under such conditions, it cannot be expected that the sensor temperature remain stable and hence OBDII heater diagnosis shall not be carried out in these instances. The SW internal timer used to await the delay shall be reset whenever STATE_LSH_UP[i] is no longer in LSH_POW_CTL or when the other above listed conditions are no longer met.

Additionally LV_TTIP_MES_VLD_LS_UP[i] shall be set, indicating that the ceramic temperature computation is active. This is the case when either the normal operative readiness has been detected via the raw operative readiness and validated via the ceramic temperature or the conditional operative readiness has been detected (forced operative readiness). These conditions permit the use of the measured ceramic temperature (TTIP_MES_LS_UP[i]) and not its initialised value (0 °C), i.e. the system tries to measure the ceramic temperature even if the heater fails sufficiently such that the normal operative readiness cannot be detected, where the oscillator used to measure the internal resistance and hence the ceramic temperature is only switched on when the sensor shows raw operative readiness. Note that during ceramic temperature reference measurements, the flag LV_TTIP_MES_VLD_LS_UP[i] will be revoked until the reference measurement cycle has been completed. The latter shall not cause TIMER_1 to be reset.

Once the application conditions have been met, this shall be indicated by setting the flag LV_DIAG_OBD_CDN_LSH_UP[i] and the function shall consider the diagnosis to be "Available". Should the conditions not be met, then LV_DIAG_OBD_CDN_LSH_UP[i] shall be reset and the diagnosis shall be considered to be "Not Available".

The function shall implement a statistical approach to detect a weak sensor heater: If the temperature is outside calibrateable limits, an error accumulator shall be incremented either with normal steps or with large steps, depending on the temperature being above or below a limit where the sensor is not functional. If the temperature is in range, the accumulator shall be decremented slowly. If the accumulator has reached a limit, the diagnosis shall indicate that an error is present. A second increment-only accumulator shall run in parallel to indicate that the diagnosis is done in the "good" case, i.e. that a fault would have been detected if it was permanently present. The diagnosis shall stop once an error has been detected. The last measured ceramic temperature shall be copied to a variable located in NVMY for Mode $06 scan tool output.

If the repeat flag LC_OBD_REP_LSH_UP_i is not set, the diagnosis shall stop as soon as the increment-only accumulator has reached or exceeded the required value and the end flag is set. If not, the diagnosis keeps running, but the result shall not change anymore once an error has been indicated.

Application conditions:

Initialisation:

The following variables shall be initialised after reset, at the transition of ignition key on (i.e. LV_IGK = 0 -> 1) and upon clearing of the error memory:

STATE_SYM_OBD_LSL_LSH_UP[i] = 0
LV_DIAG_OBD_SYM_LSH_UP[i] = 0
LV_DIAG_OBD_CDN_LSH_UP[i] = 0
LV_DIAG_OBD_END_LSH_UP[i] = 0
LV_LAMB_PLS_REQ_DIAG_LSH_UP[i] = 0
TCC_ERR_OBD_LSH_UP[i] = 0
TCC_END_DIAG_OBD_LSH_UP[i] = 0
LV_ERR_OBD_VLD_LSH_UP[i] = 0
ERR_SYM_OBD_VLD_LSH_UP[i] = “No symptom”
LV_CDN_DIAG_OBD_VLD_LSH_UP[i] = 0
LV_ERR_TTIP_MES_LSH_UP[i] = 0
ERR_SYM_TTIP_MES_LSH_UP[i] = “NO_SYM”
LV_CDN_DIAG_TTIP_MES_LSH_UP[i] = 0

***Reset** all timers*

NOTE: timers refer to internal SW timer and not to output data variable.

Recurrence:

The monitoring cycles shall occur every 100 ms providing the conditions are met.

Activation: "in all engine states"

Formula section:

If(1) LV_ST_END = 1
and LC_OBD_LSH_UP = 1
and LV_INH_DIAG_OBD_LSH_UP[i] = 0
and STATE_LSH_UP[i] <> LSH_OFF
and STATE_LSH_UP[i] <> LSH_VB_PROT
and STATE_LSH_UP[i] <> LSH_TEMP_PROT
and LV_VB_CDN_OBD_2 = 1
and C_LSHPWM_MIN_LSH_UP ≤ LSHPWM_UP[i]
and (LV_DIAG_OBD_END_LSH_UP[i] = 0
or LC_OBD_REP_LSH_UP = 1)
and LV_TEMP_DEW_LS_UP[i] = 1
then(1)

% recognize both errors for delayed readiness

LV_CDN_DIAG_TTIP_MES_LSH_UP[i] = 1
LV_CDN_DIAG_OBD_VLD_LSH_UP[i] = 1

```
If(2)   LV_LS_UP_READY[i] = 0
then(2) Reset TIMER_1
        LV_DIAG_OBD_CDN_LSH_UP[i] = 1
        Diagnosis Available
        If(3)   T_POW_RISE_LSH_UP[i] ≥
                IP_T_DLY_LS_UP_READY_OBD_LSH_UP
        then(3) STATE_SYM_OBD_LSL_LSH_UP[i] = READY_ERR
                Symptom “Oxygen sensor heater fault READY_ERR” active
                LV_DIAG_OBD_SYM_LSH_UP[i] = 1
                LV_DIAG_OBD_END_LSH_UP[i] = 1
                TTIP_MES_LS_UP_OBD_LSH_UP[i] = TTIP_MES_LS_UP[i]
        else(3) Reset STATE_SYM_OBD_LSL_LSH_UP[i]
                Symptom “Oxygen sensor heater fault READY_ERR” passive
                LV_DIAG_OBD_SYM_LSH_UP[i] = 0
        endif(3)
else(2) If(3)   LV_DIAG_OBD_SYM_LSH_UP[i] = 0
        and     STATE_LSH_UP[i] = LSH_POW_CTL
        and     TEG_DYN_LS_UP[i] > C_TEG_DYN_MIN_OBD_LSH_UP
        then(3) increment TIMER_1
                If(4)   TIMER_1 ≥ C_T_DLY_POW_CTL_OBD_LSH_UP
                then(4) LV_DIAG_OBD_CDN_LSH_UP[i] = 1
                        Diagnosis Available
                        If(5)   LV_TTIP_MES_VLD_LS_UP[i] = 1
                        and     LV_V_REF_VLD_R_IT_LS_UP[i] = 1
                        then(5)
                                If        TCC_END_DIAG_OBD_LSH_UP[i]
                                          < C_TCC_THD_ERR_OBD_LSH_UP
                                Then(6a) % don't exceed the threshold
                                          increment TCC_END_DIAG_OBD_LSH_UP[i]
                                          with C_TCC_ERR_INC_OBD_LSH_UP
                                Endif(6a)
                                If        (C_TTIP_MAX_OBD_LSH_UP < TTIP_MES_LS_UP[i]
                                or         TTIP_MES_LS_UP[i] < C_TTIP_MIN_OBD_LSH_UP)
                                then(6b) % symptom  found.  set state, store "bad" TTIP, push error ctr.
                                          STATE_SYM_OBD_LSL_LSH_UP[i] = TTIP_ERR
                                          Symptom “Oxygen sensor heater fault TTIP_ERR” active
```

```
        TTIP_MES_LS_UP_OBD_LSH_UP[i] = TTIP_MES_LS_UP[i]
        If      TTIP_MES_LS_UP[i] < C_TTIP_THD_IPLSL_ACT_LS_UP
        then(7a) % pump current will be off, accelerated error recognition
            increment TCC_ERR_OBD_LSH_UP[i]
            with C_TCC_ERR_INC_OBD_LSH_UP_FAST
            % freeze at upper limit, don't wrap
        Else(7a) % normal error recognition
            increment TCC_ERR_OBD_LSH_UP[i]
            with C_TCC_ERR_INC_OBD_LSH_UP
            % freeze at upper limit, don't wrap
        Endif(7a)
    Else(6b) % no symptom.  clear state, store "good" TTIP, roll back error ctr.
        TTIP_MES_LS_UP_OBD_LSH_UP[i] = TTIP_MES_LS_UP[i]
        reset STATE_SYM_OBD_LSL_LSH_UP[i]
        Symptom "Oxygen sensor heater fault TTIP_ERR" passive
        decrement TCC_ERR_OBD_LSH_UP[i]
        % freeze at zero limit, don't wrap
    Endif(6b)
    If      TCC_END_DIAG_OBD_LSH_UP[i]
              >= C_TCC_THD_ERR_OBD_LSH_UP
    or      TCC_ERR_OBD_LSH_UP[i] >= C_TCC_THD_ERR_OBD_LSH_UP
    Then(6c) % we have a result available
        LV_DIAG_OBD_END_LSH_UP[i] = 1
    Endif(6c)
    If      TCC_ERR_OBD_LSH_UP[i] >= C_TCC_THD_ERR_OBD_LSH_UP
    Then(6c) % an error was debounced
        LV_DIAG_OBD_SYM_LSH_UP[i] = 1
    Endif(6c)
else(5)Symptom "Oxygen sensor heater fault TTIP_MES_ERR"
    active
    STATE_SYM_OBD_LSL_LSH_UP[i] = TTIP_MES_ERR
    Increment TIMER_2
    If(6)   TIMER_2 ≥ C_T_DLY_TTIP_MES_OBD_LSH_UP
    then(6) LV_DIAG_OBD_SYM_LSH_UP[i] = 1
        LV_DIAG_OBD_END_LSH_UP[i] = 1
```

```
                                        TTIP_MES_LS_UP_OBD_LSH_UP[i] =
                                        TTIP_MES_LS_UP[i]
                                endif(6)
                        endif(5)
                        If(5)   (LV_TTIP_MES_VLD_LS_UP[i] = 1->0
                        and     Timer_3 = 0)
                        or      Timer_3 <> 0
                        then(5)Increment TIMER_3
                                If(6)   TIMER_3 ≥ C_T_DLY_TTIP_RES_OBD_LSH_UP
                                then(7)Reset TIMER_2
                                        Reset TIMER_3
                                endif(7)
                        endif(5)
                else(4)Diagnosis Not Available
                        LV_DIAG_OBD_CDN_LSH_UP[i] = 0
                endif(4)
        else(3)Reset TIMER_1
                LV_DIAG_OBD_CDN_LSH_UP[i] = 0
                Diagnosis Not Available
        endif(3)
    endif(2)
else(1)
    Reset TIMER_1
    LV_DIAG_OBD_CDN_LSH_UP[i] = 0
  % recognize both errors for delayed readiness
  LV_CDN_DIAG_TTIP_MES_LSH_UP[i] = 0
  LV_CDN_DIAG_OBD_VLD_LSH_UP[i] = 0
    Diagnosis Not Available
endif(1)

Symptom Validation
% decide about general error.  never reset in one driving cycle
If          LV_ERR_TTIP_MES_LSH_UP[i]= 0
and LV_DIAG_OBD_END_LSH_UP[i] = 1
then(1b) % general tip temperature plausibility diagnosis is finished
            LV_END_DIAG_TTIP_MES_LSH_UP[i] = 1
```

```
        If      LV_DIAG_OBD_SYM_LSH_UP[i] = 1
        then(2b) % a failure is present
                LV_ERR_TTIP_MES_LSH_UP[i] = 1
                ERR_SYM_TTIP_MES_LSH_UP[i] = STATE_SYM_OBD_LSL_LSH_UP[i]
        endif(2b)
endif(1b)
% now, if the OC diagnosis also has a result, decide about real heater error entry.
If      LV_END_DIAG_OC_LSL_UP[i] = 1
and STATE_DIAG_OC_LSL_UP[i] <> "active"
then(1c) %hold back if open circuit debounce is going on
        LV_END_DIAG_OBD_VLD_LSH_UP[i] = 1
        If      LV_ERR_TTIP_MES_LSH_UP[i] = 0
        or      LV_ERR_OC_LSL_UP[i] = 1
        then(2c) % no failure present or failure is due to an open circuit
                LV_ERR_OBD_VLD_LSH_UP[i] = 0
                ERR_SYM_OBD_VLD_LSH_UP[i] = “No symptom”
        else(2c) % a real heater failure is present
                LV_ERR_OBD_VLD_LSH_UP[i] = 1
                ERR_SYM_OBD_VLD_LSH_UP[i] = STATE_SYM_OBD_LSL_LSH_UP[i]
        endif(2c)
endif(1c)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LSHPWM_MIN_LSH_UP	1	0...FFH	0...99.60937	0.390625	[%]
Minimum permitted heater effective voltage PWM for upstream LSH OBD2 diagnosis					
C_T_DLY_POW_CTL_OBD_LSH_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum delay prior to checking status of OBD1 heater /signal faults					
C_TTIP_MIN_OBD_LSH_UP	1	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Lower diagnosis threshold for oxygen sensor ceramic temperature					
C_TTIP_MAX_OBD_LSH_UP	1	8000...7FFFH	-2048... 2047.9375	0.0625	[°C]
Upper diagnosis threshold for oxygen sensor ceramic temperature					
C_TEG_DYN_MIN_OBD_LSH_UP	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Minimum exhaust temperature to activate oxygen sensor ceramic temperature diagnosis					
C_TCC_THD_ERR_OBD_LSH_UP	1	0...FFFFH	0...65535	1	[-]
Fault detection threshold, when exceeded, heater is determined to be defective during driving cycle					
C_TCC_ERR_INC_OBD_LSH_UP	1	0...FFH	0...255	1	[-]
Increment of heater fault counter in case that Ttip is below diagnostic threshold					
C_TCC_ERR_INC_OBD_LSH_UP_FAST	1	0...FFH	0...255	1	[-]
Fast increment of heater fault counter in case that Ttip is below IP control activation limit					
IP_T_DLY_LS_UP_READY_OBD_LSH_UP	6	0...FFFFH	0...6553.5	0.1	[s]
LDPM_TEMP_INI_LS_UP_1_EGCP	6	0...FFFFH	-2048... 2047.9375	0.0625	[°C]
Time threshold after heater management start where fault detected, if no full operative readiness observed					
LC_OBD_REP_LSH_UP	1	0...1H	0...1	1	[-]
Logical constant permitting dignosis to be repeated within onde driving cycle					
LC_OBD_LSH_UP	1	0...1H	0...1	1	[-]
Logical constant permitting dignosis to be carried out					
C_T_DLY_TTIP_MES_OBD_LSH_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time delay after TTIP_ERR_MES symptom valid to set error flag					
C_T_DLY_TTIP_RES_OBD_LSH_UP	1	0...FFFFH	0...6553.5	0.1	[s]
Time delay to reset TTIP_ERR_MES timer					

The calibration of the anti-bounce is defined in the chapter “Diagnosis information”.

Configuration for diagnostic symptoms:

Diagnostic XX	**Symptom description**	**Symptom**	**Filter type**
OBD_VLD_LSH_UP[NC_CBK_EX_NR]	“sensor temperature out of range”	SYM_0	NO
	“sensor readiness delayed”	SYM_1	
	“invalid sensor temperature”	SYM_2	

Diagnostic XX	**Symptom description**	**Symptom**	**Filter type**
TTIP_MES_LSH_UP [NC_CBK_EX_NR]	“sensor temperature out of range”	SYM_0	NO
	“sensor readiness delayed”	SYM_1	
	“invalid sensor temperature”	SYM_2	

16.53 Application incidences for oxygen sensor heater OBDII monitoring

Output data

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_OBD_LSH_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean inhibit flag for O2 sensor heater OBDII monitoring					
STATE_RBM_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Interface of OBD_VLD_LSH_UP[NC_CBK_EX_NR] monitor with the Rate-Based Monitoring statisticsBit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1)Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1)Bit 2					
LV_INH_DIAG_RBM_OBD_LSH_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean inhibit flag for Rate Based Monitoring of O2 sensor heater OBDII diagnosis					

Input data:

LV_VB_CDN_OBD_2	LV_MIS_STATE_A	LV_ERR_CHG_LS_UP
LV_ST_END	LV_MIS_STATE_B	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]
LV_VAR_LSH_UP	LV_ERR_CHG_LS_UP	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_MIS[NC_CYL_NR]	LV_IGK	LV_ERR_LSH_UP[NC_CBK_EX_NR]
CTR_ERR_DYN_NR	NC_CBK_EX_NR	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]
NC_CYL_NR	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]
LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

16.53.1 Upstream sensor heater OBD2 diagnosis: Inhibition of diagnosis

Description:

This function shall evaluate the application conditions applicable to the oxygen sensor ceramic temperature OBDII monitoring and provide the result in a single boolean flag. This flag shall represent the interface to the main diagnosis function.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2), then
    i = 1, for cylinder bank 1
    i = 2, for cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank 1.

Application conditions:

Initialisation: at every transition LV_IGK = 0 -> 1or reset all variables shall be reset to 0, except STATE_RBM_OBD_VLD_LSH_UP[i] *//for init of this variable see section below*

at every transition LV_IGK = 1 -> 0

- LV_INH_DIAG_OBD_LSH_UP[NC_CBK_EX_NR] = 1

Recurrence: 1 s

Activation: LV_IGK = 1

Formula section:

```
If      LV_ERR_TTIP_MES_LSH_UP[i] = 1          or
        LV_ERR_LSL_UP_IF[i] = 1                or
        LV_ERR_AIR_LSL_UP[i] = 1               or
        LV_ERR_EL_LSL_UP[i] = 1                or
        LV_ERR_OC_LSL_UP[i] = 1                or
        LV_ERR_LSH_UP[i] = 1                   or
        LV_ERR_CHG_LS_UP = 1                   or
        LV_MIS_STATE_A = 1                     or
        LV_MIS_STATE_B = 1                     or
        LV_ERR_OFS_LSL_UP[i] = 1               or
        LV_ERR_CTL_LSL_UP[i] = 1

then    LV_INH_DIAG_RBM_OBD_LSH_UP[i]  =  1
else    LV_INH_DIAG_RBM_OBD_LSH_UP[i]  =  0
endif

If      LV_LS_UP_OBD_2_MAN_DEAC[i] = 1         or
        LV_INH_DIAG_RBM_OBD_LSH_UP[i]  =  1or
        LV_VB_CDN_OBD_2 = 0                    or
        LV_VAR_LSH_UP = 0

then    LV_INH_DIAG_OBD_LSH_UP[i]  =  1
else    LV_INH_DIAG_OBD_LSH_UP[i]  =  0
endif
```

16.53.2 Upstream sensor heater OBD2 diagnosis: Interface for rate – based – monitoring

Description:

With this module the interface between the OBD_VLD_LSH_UP[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_OBD_VLD_LSH_UP[i] data.

Within STATE_RBM_OBD_VLD_LSH_UP[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for heater diagnosis)

Application conditions:

Initialisation: *at ECU reset:*

bit 0, bit 1 and bit 2 of STATE_RBM_OBD_VLD_LSH_UP[i] = 0

LV_DC 0 --> 1 transition:

bit 0 and bit 1 of STATE_RBM_OBD_VLD_LSH_UP[i] = 0

bit 2 of STATE_RBM_OBD_VLD_LSH_UP[i] = 1

on failure memory reset:

bit 1 of STATE_RBM_OBD_VLD_LSH_UP[i] = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 -->1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_CHG_LS_UP	LV_ERR_MIS[NC_CYL_NR]	LV_ERR_LSH_UP[NC_CBK_EX_NR]
LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]
	LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]	

```
If(1)     { CPU optimization at LV_DC 0 --> 1 transition }
          CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
          While  bit 1 of STATE_RBM_OBD_VLD_LSH_UP[i] = 0 do
```

```
                with each XX failure of the above list  (depending on configuration, see
                below):
                ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                SYNCHRONIZATION<CALL>)
                If(2)     XX has a pending status
                Then(2)        bit 1 of STATE_RBM_OBD_VLD_LSH_UP[i] = 1
                Endif(2)
            Endwhile
    Else(1)   { the dynamic failure memory is empty }      No action
    Endif(1)
```

Every 1 s :

```
If    bit 0 of STATE_RBM_OBD_VLD_LSH_UP[i] = 0
Then
            If      LV_END_DIAG_OBD_VLD_LSH_UP[i] = 1
            Then    bit 0 of STATE_RBM_OBD_VLD_LSH_UP[i] = 1
            Endif
Endif
If    bit 1 of STATE_RBM_OBD_VLD_LSH_UP[i] = 0
Then
            If      LV_INH_DIAG_RBM_OBD_LSH_UP[i] = 1
            Then    bit 1 of STATE_RBM_OBD_VLD_LSH_UP[i] = 1
            Endif
Endif
```

16.53.3 ERRM configuration

Diagnosis	Symptom	Nr	ABC type
Oxygen sensor upstream heater monitoring plausibility	*TTIP_ERR*	*0*	NO
	READY_ERR	*1*	
	TTIP_MES_ERR	*2*	
OBD_VLD_LSH_UP[i]		*3*	

16.54 O2 sensor (lin, up) activity diag

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_RAW_ACT_SYM_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Counter to set the raw symptom of sensor no-activity check (first part)					
CTR_RAW_ACT_END_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Counter to have valid diagnostic result from the no-activity check (first part)					
LV_DIAG_RAW_ACT_SYM_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating a detected raw symptom of sensor non-activity (from first test)					
LV_DIAG_RAW_ACT_END_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating end of raw symptom test of no-activity check (from first test)					
LAMB_DE_LSL[NC_CBK_EX_NR]	V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Deviation of the instantaneous lambda value from the mean moving value					
LAMB_DE_INT_ACT_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Cumulating value of the lambda deviation					
LAMB_DE_TOT_ACT_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Total cumulated lambda deviation					
LAMB_MMV_ACT_LSL_UP[NC_CBK_EX_NR]	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda moving mean value during the time the forced stimulation is active					
LV_DIAG_ACT_CDN_LSL_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean flag indicating the conditions to activate the second test for sensor non-activity are fulfilled					
LV_DIAG_ACT_SHO_CDN_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating the short term conditions to activate the second test for sensor non-activity are fulfilled					
CTR_SYM_ACT_LSL_UP[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Debounce counter of the sensor non-activity symptom					
CTR_END_ACT_LSL_UP[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter to have valid diagnostic result from the no-activity check (second part)					
CTR_LAMB_DE_INT_ACT_LSL_UP[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter of forced stimulation periods					
LV_DIAG_ACT_SYM_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating a detected poor upstream sensor activity (from second test)					
LV_DIAG_ACT_END_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating end of detailed test of no-activity check (from second test)					

Input data:

LV_IPLSL_VLD[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]	LV_DIAG_ACT_INH_LSL_UP[NC_CBK_EX_NR]	LV_VLS_UP_VLD[NC_CBK_EX_NR]
FAC_MV_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	NC_CBK_EX_NR	FAC_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	VLS_UP_DIAG[NC_CBK_EX_NR]
LV_LAMB_PLS_ACT[NC_CBK_EX_NR]	T_SUM_AFL_AFR_CYC[NC_CBK_EX_NR]	LAMB_SP_DE_PLS	LV_IGK
MAF_KGH	LV_AFL[NC_CBK_EX_NR]	N_32	
LV_LS_DOWN_READY[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]		

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

This function shall facilitate the upstream sensor non-activity detection, which can be caused by a extremely cold sensors or an open circuit some ATIC42 or CJ125 lines.

Description:

In the first part of this test the sensor is assumed to have no activity if VLS_UP_DIAG remains stuck in a tight band around stoichiometric values. If this is true for C_CTR_RAW_ACT_SYM_LSL_UP recurrences, then the raw symptom flag LV_DIAG_RAW_ACT_SYM_LSL_UP is set.
Rate Based Monitoring (RBM) requires similar time spans for the diagnosis to come to an end with or without a failure present. Therefore, if VLS_UP_DIAG is not within the mentioned band (thus a failure is unlikely) the counter CTR_RAW_ACT_END_LSL_UP is incremented until the threshold C_CTR_RAW_ACT_END_LSL_UP is reached. At this point the end flag LV_DIAG_RAW_ACT_END_LSL_UP is set. VLS_UP_DIAG of a faultless sensor might pass through or come to lie within the critical band. In these cases the end counter is frozen and the incrementation is resumed, once the critical band is left again.
In the failure case CTR_RAW_ACT_SYM_LSL_UP is incremented until the threshold C_CTR_RAW_ACT_SYM_LSL_UP is reached, where the end flag and the raw symptom flag are set at the same instance. In case of intermittent faults, excursion from the critical band should not lead to an immediate reset of the raw symptom flag. Therefore, the raw symptom flag is taken back only if the raw symptom counter drops below C_CTR_RAW_ACT_RST_LSL_UP.

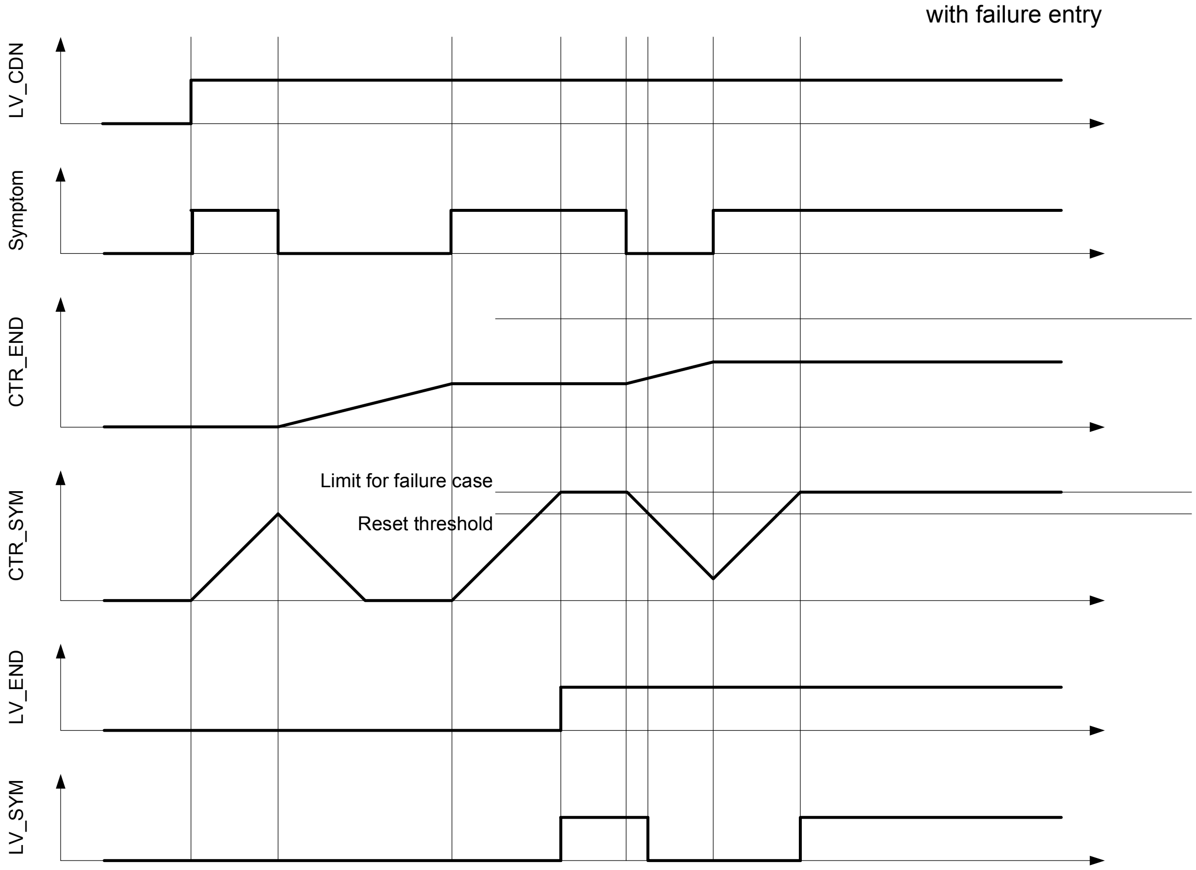


Figure 1: Basic principle of debounce algorithm for first part of activity check (LV_DIAG_RAW_ACT_END/SYM_LSL_UP)

The second part is started once the first part has come to an end and the following conditions are met: The forced stimulation is active and additionally engine speed, engine load, amplitude and period of the forced stimulation are sufficient to provoke the fluctuation of the sensor output. If these conditions are fulfilled then the following procedure is carried out. The difference between LAMB_LS_UP[i] and the moving mean value LAMB_MMV_ACT_LSL_UP[i] is calculated and integrated. After C_CTR_LAMB_DE_INT_ACT_LSL_UP forced stimulation periods the result is saved in the variable LAMB_DE_TOT_ACT_LSL_UP[i].
While integrating the difference between LAMB_LS_UP[i] and the moving mean value two different gain factors are taken into account. First, ID_LAMB_GAIN_LS_UP_DYN considers an aged sensor. Second, with ID_LAMB_GAIN_VLS_DOWN the speed of integration is reduced, if VLS_DOWN[i] shows either very rich or very lean conditions. The background here is that the combination of an upstream signal near stoichiometry and a very rich or very lean downstream signal strongly points to a not active upstream signal.
If the value of the deviation integral after C_CTR_LAMB_DE_INT_ACT_LSL_UP forced stimulation cycles is lower than the threshold C_LAMB_DE_TOT_ACT_LSL_UP, then the symptom counter shall be incremented, otherwise it shall be decremented. If the symptom counter has achieved the maximum C_CTR_SYM_MAX_ACT_LSL_UP cycles, then the symptom flag LV_DIAG_ACT_SYM_LSL_UP[i] is set, indicating that the sensor activity is very poor. This flag shall trigger the Open Circuit Diagnosis to narrow down the source of the failure. To have a valid end flag in case of a faultless sensor the counter C_CTR_END_ACT_LSL_UP is incremented once the end flag of the first part is set and no raw symptom was detected. The end flag of the Activity Check is set when the end counter reaches the sum of both loops needed for the fault detection.

Application conditions:

Initialisation:

At the transition LV_IGK = 0 to 1, reset or at clearing error memory reset all variables.

At the transition LV_DIAG_ACT_CDN_LSL_UP[i] = 0 -> 1 initialize as follow:

CTR_SYM_ACT_LSL_UP[i] = 0
CTR_END_ACT_LSL_UP[i] = 0
CTR_RAW_ACT_SYM_LSL_UP[i] = 0
CTR_RAW_ACT_END_LSL_UP[i] = 0
CTR_LAMB_DE_INT_ACT_LSL_UP[i] = 0
LAMB_DE_INT_ACT_LSL_UP[i] = 0
LAMB_DE_TOT_ACT_LSL_UP[i] = 0
LAMB_MMV_ACT_LSL_UP[i] = 1

Recurrence: 20 ms

Activation:

LV_ST_END = 1

Deactivation:

LV_ST_END = 0

Formula section:

```
IF [LV_IPLSL_VLD[i] = 1 OR (LV_IPLSL_VLD[i] = 0 AND LV_VLS_UP_VLD[i] = 0)]
AND  LV_DIAG_ACT_INH_LSL_UP[i] = 0
AND  FAC_MV_DIAG_DYN_LSL_UP[i] < C_FAC_DYN_INT_ACT_LSL_UP
THEN
    LV_DIAG_ACT_CDN_LSL_UP[i] = 1
ELSE
    LV_DIAG_ACT_CDN_LSL_UP[i] = 0
ENDIF



% First Part: Fast detection of no sensor activity
% counter management

IF(1b) LV_DIAG_ACT_CDN_LSL_UP[i] = 1
    IF(2a)    VLS_UP_DIAG[i] < C_VLS_UP_AFS_MIN_PLAUS_LSL_UP
    OR(2a)  VLS_UP_DIAG[i] > C_VLS_UP_AFS_MAX_PLAUS_LSL_UP
    THEN(2a) % no raw symptom is present
        IF(3a) CTR_RAW_ACT_END_LSL_UP[i] < C_CTR_RAW_ACT_END_LSL_UP
        THEN(3a)
            increment CTR_RAW_ACT_END_LSL_UP[i] with ID_CTR_INC_RAW_ACT_LSL_UP
        ENDIF(3a)
        decrement CTR_RAW_ACT_SYM_LSL_UP[i] with ID_CTR_INC_RAW_ACT_LSL_UP
    ELSE(2a) % raw symptom is present
        IF(3b) CTR_RAW_ACT_SYM_LSL_UP[i] < C_CTR_RAW_ACT_SYM_LSL_UP
        THEN(3b)
            increment CTR_RAW_ACT_SYM_LSL_UP[i] with ID_CTR_INC_RAW_ACT_LSL_UP
        ENDIF(3b)
    ENDIF(2a)


    % end flag and symptom flag management
    % diagnosis has come to an end without any raw symptom present
    IF(2b) CTR_RAW_ACT_END_LSL_UP[i] >= C_CTR_RAW_ACT_END_LSL_UP
    THEN(2b)
        LV_DIAG_RAW_ACT_END_LSL_UP[i] = 1
    END(2b)
    % raw symptom is detected and sensor non-activity detailed check is triggered
    IF(2c) CTR_RAW_ACT_SYM_LSL_UP[i] >= C_CTR_RAW_ACT_SYM_LSL_UP
```

```
        THEN(2c)
            LV_DIAG_RAW_ACT_SYM_LSL_UP[i] = 1
            LV_DIAG_RAW_ACT_END_LSL_UP[i] = 1
        ELSEIF(2c)
            % short excursion from critical window should not lead to an immediate reset of the raw symptom flag
            CTR_RAW_ACT_SYM_LSL_UP[i] < C_CTR_RAW_ACT_RST_LSL_UP
        THEN(2c)
            LV_DIAG_RAW_ACT_SYM_LSL_UP[i] = 0
        END(2c)
    END(1a)

        %  Second Part: Monitoring upstream sensor integral deviation
        % Checking the long term conditions enabling the integral deviation monitor
    IF(1b) LV_DIAG_ACT_CDN_LSL_UP[i] = 1 AND LV_DIAG_RAW_ACT_END_LSL_UP[i] = 1
    THEN(1b)
        % Checking the short term conditions enabling the integral deviation monitor
        IF(2d)   LV_LAMB_PLS_ACT[i] = 1
        AND(2d)          T_SUM_AFL_AFR_CYC[i] > C_T_SUM_AFL_AFR_ACT_LSL_UP
        AND(2d)          LAMB_SP_DE_PLS > C_LAMB_SP_DE_PLS_ACT_LSL_UP
        AND(2d)          MAF_KGH > C_MAF_KGH_MIN_ACT_LSL_UP
        AND(2d)          N_32 > C_N_MIN_ACT_LSL_UP
        AND(2d)          LV_LS_DOWN_READY[i] = 1
        THEN(2d)
              LV_DIAG_ACT_SHO_CDN_LSL_UP[i] = 1
        ELSE(2d)
              LV_DIAG_ACT_SHO_CDN_LSL_UP[i] = 0
        ENDIF(2d)

        % reset counter
        IF(2e)   LV_DIAG_RAW_ACT_SYM_LSL_UP[i] 0 -> 1
        OR(2e) LV_DIAG_RAW_ACT_SYM_LSL_UP[i] 1 -> 0
        THEN(2e)
           reset CTR_LAMB_DE_INT_ACT_LSL_UP[i]
           reset CTR_SYM_ACT_LSL_UP[i]
           reset LV_DIAG_ACT_SYM_LSL_UP[i]
           reset LAMB_DE_INT_ACT_LSL_UP[i]
        ENDIF(2e)
```

```
% activation of the integral deviation monitor
LAMB_MMV_ACT_LSL_UP[i] (new) =
        LAMB_MMV_ACT_LSL_UP[i] (old) * (1-C_CRLC_ACT_LSL_UP) +
        C_CRLC_ACT_LSL_UP * LAMB_LS_UP[i]

IF (2f)  LV_DIAG_ACT_SHO_CDN_LSL_UP[i] = 1
THEN (2f)
    IF(3f) LV_AFL[i] = 0 -> 1
    THEN(3f)
        IF(4f) LV_DIAG_RAW_ACT_SYM_LSL_UP[i] = 0
        THEN(4f)        % no raw symptom present => no failure case
            IF(5f) CTR_END_ACT_LSL_UP[i] <
                        (C_CTR_LAMB_DE_INT_ACT_LSL_UP + 1) *
        C_CTR_SYM_MAX_ACT_LSL_UP/C_CTR_SYM_INC_ACT_LSL_UP
                % to mirror the time span needed to detect a sensor with no activity, the time delay of the first part has to be considered here, too (RBM require ment).
                THEN(5f)
                        increment CTR_END_ACT_LSL_UP[i]
            ELSE(5f)
                        LV_DIAG_ACT_END_LSL_UP = 1
            ENDIF(5f)
        ELSE(4f) % raw symptom present => failure case
            % freeze CTR_END_ACT_LSL_UP[i]
            IF(5g) CTR_LAMB_DE_INT_ACT_LSL_UP[i] <
                C_CTR_LAMB_DE_INT_ACT_LSL_UP
            THEN(5g)
                        increment CTR_LAMB_DE_INT_ACT_LSL_UP[i]
            ELSE(5g)
                % analysis of the deviation integral is only performed after the raw symptom is constantly present for C_CTR_LAMB_DE_INT_ACT_LSL + 1 times.
                        LAMB_DE_TOT_ACT_LSL_UP[i]                    =
        LAMB_DE_INT_ACT_LSL_UP[i]
                LAMB_DE_INT_ACT_LSL_UP[i] = 0
                CTR_LAMB_DE_INT_ACT_LSL_UP[i] = 0
                IF(6f)  LAMB_DE_TOT_ACT_LSL_UP[i]  ≤
                                C_LAMB_DE_TOT_ACT_LSL_UP
                THEN(6f) % deviation integral is low => possible failure
                        IF(7f) CTR_SYM_ACT_LSL_UP[i] <
```

```
                                        C_CTR_SYM_MAX_ACT_LSL_UP
                              THEN(7f)
                                  increment CTR_SYM_ACT_LSL_UP[i] with
                                              C_CTR_SYM_INC_ACT_LSL_UP
                              ENDIF(7f)
                          ELSE(6f) %deviation integral is sufficiently high => failure unlikely
                              IF(7g) CTR_SYM_ACT_LSL_UP[i] >
                                        C_CTR_SYM_DEC_ACT_LSL_UP
                              THEN(7g)
                                  decrement CTR_SYM_ACT_LSL_UP[i] with
                                              C_CTR_SYM_DEC_ACT_LSL_UP
                              ENDIF(7g)
                          ENDIF(6f)
                          IF(6g)  CTR_SYM_ACT_LSL_UP[i] <=
                                  C_CTR_SYM_DEC_ACT_LSL_UP
                          THEN(6g)
                              LV_DIAG_ACT_SYM_LSL_UP[i] = 0
                          ELSEIF(6g) CTR_SYM_ACT_LSL_UP[i] >=
                                  C_CTR_SYM_MAX_ACT_LSL_UP
                              LV_DIAG_ACT_SYM_LSL_UP[i] = 1
                              LV_DIAG_ACT_END_LSL_UP[i] = 1
                          ENDIF(6g)
                  ENDIF(5g)
              ENDIF(4f)
          ELSE(3f)
              LAMB_DE_LSL[i]  =  | LAMB_LS_UP[i] – LAMB_MMV_ACT_LSL_UP[i] |
              LAMB_DE_INT_ACT_LSL_UP[i] (new)  = LAMB_DE_INT_ACT_LSL_UP[i](old) +
              LAMB_DE_LSL[i] * ID_LAMB_GAIN_VLS_DOWN * ID_LAMB_GAIN_LS_UP_DYN
          ENDIF(3f)
      ELSE(2f) %short term conditions for activation of the integral deviation monitor not fulfilled
          Freeze CTR_LAMB_DE_INT_ACT_LSL_UP[i]
          Freeze LAMB_DE_INT_ACT_LSL_UP[i]
      ENDIF(2f)
  ENDIF(1b)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_DYN_INT_ACT_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Relative sensor signal damping admitted to monitor the sensor activity(see WRAF Sensor Dynamic Diagnosis)					
C_N_MIN_ACT_LSL_UP	1	0...FFH	0...8160	32	[rpm]
Lower engine speed admitted to monitor the sensor activity					
C_T_SUM_AFL_AFR_ACT_LSL_UP	1	0...FFFFH	0...655.35	0.01	[s]
Lower forced stimulation period admitted to monitor the sensor activity					
C_LAMB_SP_DE_PLS_ACT_LSL_UP	1	0...7FFH	0...0.12493	0.061e-3	[-]
Lower forced stimulation amplitude admitted to monitor the sensor activity					
C_MAF_KGH_MIN_ACT_LSL_UP	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Lower mass airflow in Kg/h admitted to monitor the sensor activity					
C_CTR_LAMB_DE_INT_ACT_LSL_UP	1	0...FFH	0...255	1	[-]
Number of forced stimulation periods to calculate the total cumulated lambda deviation					
C_LAMB_DE_TOT_ACT_LSL_UP	1	0...FFFFH	0...63.99902	0.9766e-3	[-]
Threshold of the total cumulated lambda deviation to debounce the sensor non-activity symptom					
C_CTR_SYM_MAX_ACT_LSL_UP	1	0...FFH	0...255	1	[-]
Counter threshold to set the sensor non-activity symptom flag					
C_CTR_SYM_INC_ACT_LSL_UP	1	0...FFH	0...255	1	[-]
Counter increment to set the sensor non-activity symptom flag					
C_CTR_SYM_DEC_ACT_LSL_UP	1	0...FFH	0...255	1	[-]
Counter decrement to set the sensor non-activity symptom flag					
C_CRLC_ACT_LSL_UP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation factor for the moving mean value					
ID_CTR_INC_RAW_ACT_LSL_UP	8	0...FFH	0...255	1	[-]
LDP_MAF_KGH_ID_CTR_ACT_LSL_UP	8	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Counter increment to detect raw symptom of upstream sensor non-activity					
ID_LAMB_GAIN_VLS_DOWN	8	0...FFH	0...1.99218	0.0078125	[-]
LDP_VLS_DOWN_ID_LAMB_GAIN	8	0...3FFH	0...4.99511	4.8828e-3	[V]
Gain/Damping factor to calculate LAMB_DE_INT_ACT_LSL_UP to double-check with VLS_DOWN					
ID_LAMB_GAIN_LS_UP_DYN	8	0...FFH	0...31.875	0.125	[-]
LDP_FAC_DIAG_DYN_LSL_UP_ID_LAMB	8	0...400H	0...1	0.9766e-3	[-]
Gain factor to consider an aged linear lambda sensor					
C_VLS_UP_AFS_MIN_PLAUS_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Lower limit of upstream sensor signal window to detect the sensor non-activity					
C_VLS_UP_AFS_MAX_PLAUS_LSL_UP	1	8000...7FFFH	-160...159.99511	4.8828e-3	[V]
Upper limit of upstream sensor signal window to detect the sensor non-activity					
C_CTR_RAW_ACT_END_LSL_UP	1	0...FFFFH	0...65535	1	[-]
Counter threshold to have valid diagnostic result from the no-activity check (first part)					
C_CTR_RAW_ACT_RST_LSL_UP	1	0...FFFFH	0...65535	1	[-]
Counter threshold to reset symptom counter if raw symptom has disappeared					
C_CTR_RAW_ACT_SYM_LSL_UP	1	0...FFFFH	0...65535	1	[-]
Counter threshold to set the raw symptom flag of the no-activity check (first part)					

16.55 Application Incidencies for “O2 sensor (lin, up) activity diag"

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_ACT_INH_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "Activity Diagnosis"					

Input data:

LV_IGK	LV_ERR_MAP	LV_ERR_MAF	STATE_ERR_IV
LV_ERR_IGC	LV_ERR_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_SAP
LV_ERR_RATIO_CHK	LV_ERR_EL_CPS	LV_ERR_TPS	LV_N_LIM_ETC_LIH
LV_ERR_TCO	LV_SAV	LV_MIS_STATE_A	LV_MIS_STATE_B4
NC_CBK_EX_NR	LV_ERR_MEC_OPEN_CPS	LV_ERR_SA_SAV	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]
LV_ERR_AMP	LV_ERR_AMP_PLAUS	LV_ERR_TH	LV_ERR_ECT_EL
LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_DIAGCPS	LV_ERR_SLV_IVVT_IN	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]
LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS		

FUNCTION DESCRIPTION:

16.55.1 General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation:

at every transition LV_IGK = 0 -> 1or reset all variables shall be reset to 0.

at every transition LV_IGK = 1 -> 0 LV_DIAG_ACT_INH_LSL_UP[i] = 1

Recurrence:

The functions shall be carried out once every 1s.

Activation:

LV_IGK = 1

Deactivation:

LV_IGK = 0

Formula Section:

general specification

This function has to be deactivated, if at least one of the following errors / special modes is active.

```
IF      LV_LS_UP_OBD_2_MAN_DEAC[i] = 1                                              OR
    -   LV_ERR_MAF = 1                 mass air flowsensor                          OR
    -   LV_ERR_MAP = 1                 intake manifold pressure sensor              OR
    -   STATE_ERR_IV > 0               injection valve 1...n malfunction            OR
    -   LV_ERR_IGC = 1                 ignition error                               OR
    -   LV_ERR_LS_UP[i] = 1            Upstream WRAF sensor                         OR
    -   LV_ERR_LS_DOWN[i]   = 1        binary oxygen sensor downstream              OR
    -   LV_ERR_RATIO_CHK = 1           mass-air / throttle position-relationship    OR
        LV_ERR_EL_CPS = 1              Canisterpurge valve                          OR
    -   LV_ERR_MEC_OPEN_CPS = 1        Canisterpurge valve mechanical blocked       OR
    -   LV_ERR_TPS = 1                 Throttle position sensor                     OR
    -   LV_ERR_TCO = 1                 Coolant temperature sensor                   OR
    -   LV_MIS_STATE_A = 1             Misfire State A                              OR
    -   LV_MIS_STATE_B4 = 1            Misfire State B4                             OR
    -   LV_N_LIM_ETC_LIH = 1           Limp home                                    OR
    -   LV_SAV = 1                     Secondary air valve active                   OR
    -   LV_SAP = 1                     Secondary air pump active                    OR
    -   LV_ERR_AMP = 1                 AMP error present                            OR
    -   LV_ERR_AMP_PLAUS = 1           Ambient pressure plausibility                OR
    -   LV_ERR_TH = 1                  coolant thermostat error                     OR
    -   LV_ERR_ECT_EL = 1              electrical thermostat diagnosis              OR
    -   LV_ERR_FSD[i] = 1              Fuel system failure                          OR
    -   LV_ERR_DIAGCPS = 1                                                          OR
    -   LV_ERR_SLV_IVVT_IN = 1                                                      OR
    -   LV_ERR_OFS_LSL_UP[i] = 1                                                    OR
    -   LV_ERR_CTL_LSL_UP[i] = 1                                                    OR
    -   LV_ERR_MAP_TPS_PLAUS = 1

THEN
    LV_DIAG_ACT_INH_LSL_UP[i] = 1
ELSE
    LV_DIAG_ACT_INH_LSL_UP[i]= 0
ENDIF
```

16.56 O2 sensor (up, down) plausibility diag

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_DIAG_PLAUS_SYM_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounced diagnosis result of the signal plausibility for the wide range A/F sensor					
STATE_SYM_DIAG_PLAUS_LSL_UP[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Oxygen Sensor Plausibility Check Symptom					
LV_DIAG_CDN_PLAUS_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Oxygen Sensor Plausibility Check condition					
LV_DIAG_PLAUS_END_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Oxygen Sensor Plausibility Check result available					
LV_DIAG_AFR_SYM_PLAUS_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Oxygen Sensor Plausibility Check - not debounced symptom "SENS_AFR"					
LV_DIAG_AFL_SYM_PLAUS_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Oxygen Sensor Plausibility Check - not debounced symptom -SENS_AFL-					
MAF_SYM_RAW_PLAUS_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral for the plausibility symptom (compensation of the catalyst delay)					
MAF_END_PLAUS_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral for valid diagnostic result					
MAF_SYM_PLAUS_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral for detection of mutual not plausible sensor signals					

Input data:

LV_IPLSL_VLD[NC_CBK_EX_NR]	LV_LS_DOWN_READY[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]
TEG_DYN_LS_DOWN[NC_CBK_EX_NR]	LV_IGK	MAF_CYL	MAF_KGH
LAMB_DELTA_LAM_ADJ[NC_CBK_EX_NR]	LV_DIAG_PLAUS_INH_LSL_UP[NC_CBK_EX_NR]	LV_LAMB_SP_AFR_REQ_DIAG_ACT[NC_CBK_EX_NR]	LV_LAMB_SP_AFL_REQ_DIAG_ACT[NC_CBK_EX_NR]
NC_CBK_EX_NR	N_32	LV_VLS_DIF_DLY_LDC[NC_CBK_EX_NR]	LV_CAT_PURGE_ACT[NC_CBK_EX_NR]
TCO			

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)

i = 1, for single exhaust cylinder bank.

Description:

The plausibility check between the upstream oxygen sensor (WRAF sensor) and the downstream oxygen sensor (binary) signal is done once the expected driving conditions are fulfilled. A non-plausibility symptom detected by this diagnosis indicates that a failure is present in one of both sensors.

Two different symptoms can be detected:

- STATE_SYM_DIAG_PLAUS_LSL_UP[i] = *"SYM_0"*, if the upstream sensor indicates a rich lambda value, while the downstream sensor signal lean.
- STATE_SYM_DIAG_PLAUS_LSL_UP[i] = *"SYM_1"*, if the upstream sensor indicates a lean lambda value, while the downstream sensor signal rich.

Once the down- and upstream sensor signals are not mutual plausible first the mass flow integral MAF_SYM_RAW_PLAUS_LSL_UP[i] is calculated to take into account the delay time caused by the oxygen storage capability of the catalyst. If this integral reaches the threshold C_MAF_SYM_RAW_PLAUS_LSL_UP[i] the status in STATE_SYM_DIAG_PLAUS_LSL_UP[i] is set to the respective symptom. If either LV_DIAG_AFR_SYM_PLAUS_LSL_UP[i] or LV_DIAG_AFS_SYM_PLAUS_LSL_UP[i] disappears the MAF integral is decremented. When the integral reaches C_MAF_SYM_RAW_RST_PLAUS_LSL_UP[i] the status flag is taken back.

Second, after STATE_SYM_DIAG_PLAUS_LSL_UP[i] is set to the respective symptom the debounce algorithm is started. If both LV_DIAG_AFR/AFS_SYM_PLAUS_LSL_UP[i] and STATE_SYM_DIAG_PLAUS_LSL_UP[i] are active MAF_SYM_PLAUS_LSL_UP is integrated. In contrast if both symptom indicators are inactive (thus no failure is expected) the integral MAF_END_PLAUS_LSL_UP[i] is integrated. Should the status of flag differ from eachother, then both integrals are frozen. MAF_SYM_PLAUS_LSL_UP is reset when the status of STATE_SYM_DIAG_PLAUS_LSL_UP[i] changes. There is not reset condition for the end integral in order to have a valid end flag also in case of a jittering signal.

LV_DIAG_PLAUS_SYM_LSL_UP[i] is set to one once the symptom integral exceeds C_MAF_SYM_PLAUS_LSL_UP[i] is exceeded. LV_DIAG_PLAUS_END_LSL_UP[i] is set when the end integral exceeds sum of C_MAF_SYM_PLAUS_LSL_UP[i] and C_MAF_SYM_RAW_PLAUS_LSL_UP[i] or when the LV_DIAG_PLAUS_SYM_LSL_UP[i] is set.

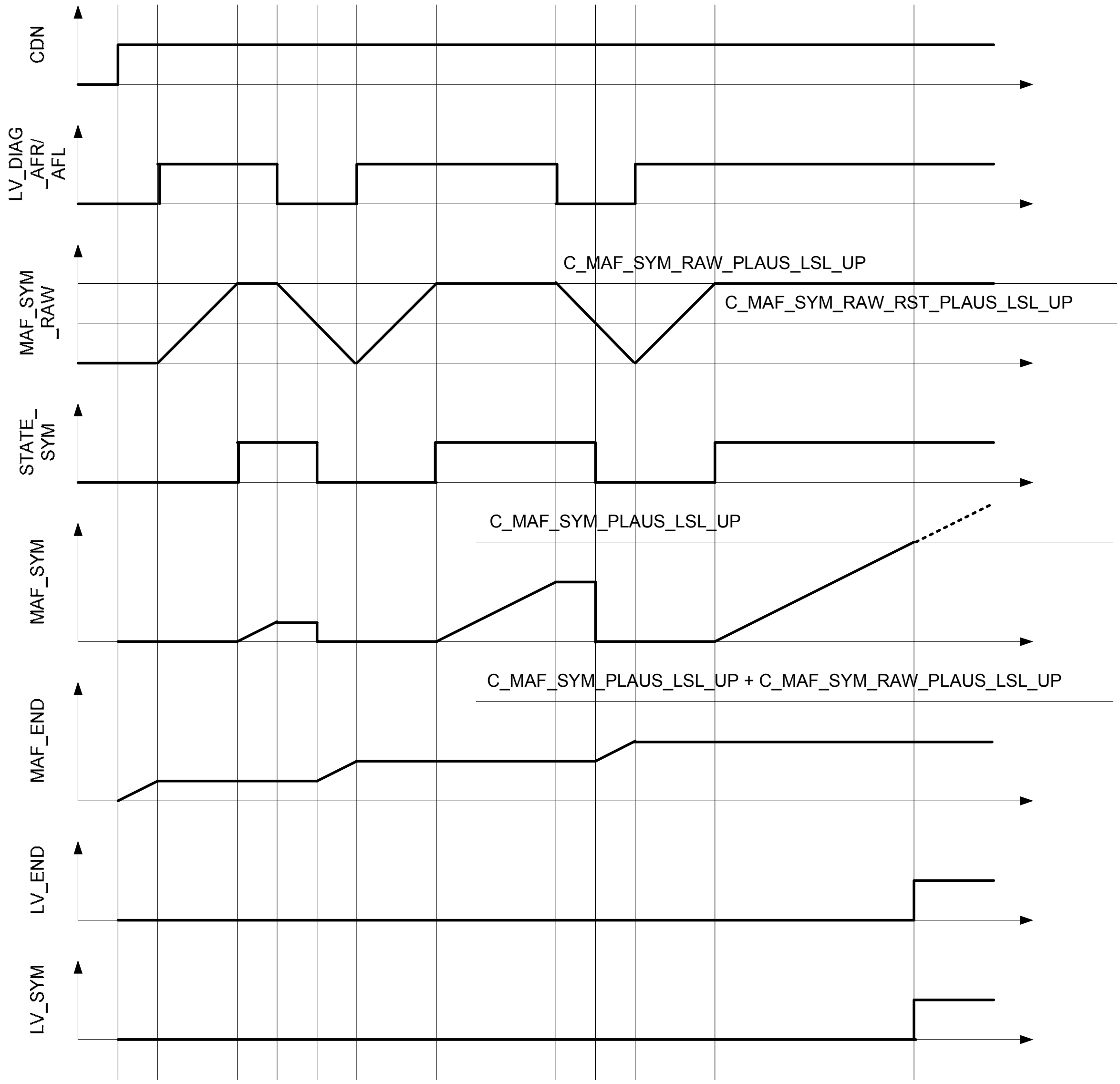


Figure 1: Basic principle of applied debounce algorithm.

The symptom flag LV_DIAG_PLAUS_SYM_LSL_UP is exported to *“Sensor Characteristic Line Shift to Lean”* and *“Shift to Rich”* functions to facilitate the detection of the faulty sensor and failure cause.

The calibration of this function is described in its Tunning Guide.

Application conditions:

Initialisation:

At transition LV_IGK = 0 to 1, reset or at clearing error memory reset all variables.

Recurrence: 100 ms

Activation:

IF LV_DIAG_PLAUS_SYM_LSL_UP[i] = 0 **AND**

```
        LV_IPLSL_VLD[i] = 1                                         AND
        LV_LS_DOWN_READY[i] = 1                                     AND
        LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 0                          AND
        LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 0                          AND
        N_32 > C_N_MIN_DIAG_PLAUS_LSL_UP                            AND
        C_MAF_KGH_MIN_DIAG_PLAUS_LSL_UP < MAF_KGH <
                      C_MAF_KGH_MAX_DIAG_PLAUS_LSL_UP               AND
        TCO ≥ C_TCO_MIN_DIAG_PLAUS_LSL_UP                           AND
        C_TEG_MIN_PLAUS_LSL_UP[i] < TEG_DYN_LS_DOWN[i] <
                     C_TEG_MAX_PLAUS_LSL_UP[i]                      AND
        LV_DIAG_PLAUS_INH_LSL_UP[i] = 0                             AND
        LV_CAT_PURGE_ACT[i] = 0                                     AND
        LV_VLS_DIF_DLY_LDC[i] = 1
    THEN  % activate function
        LV_DIAG_CDN_PLAUS_LSL_UP[i] = 1
    ELSE  % deactivate function
        LV_DIAG_CDN_PLAUS_LSL_UP[i] =0
        % freeze STATE_SYM_DIAG_PLAUS_LSL_UP[i]
        % freeze MAF_SYM_PLAUS_LSL_UP[i]
    ENDIF
```

Formula section:

```
    % Testing signal shift to rich:
    IF(1a) VLS_DOWN[i] <= C_VLS_DOWN_AFL_PLAUS_LSL_UP[i]
        % downstream sensor signal lean
    AND(1a)  LAMB_LS_UP[i] + LAMB_DELTA_LAM_ADJ[i] <
               C_LAMB_AFR_PLAUS_LSL_UP[i]
               % upstream sensor output rich
    THEN(1a) LV_DIAG_AFR_SYM_PLAUS_LSL_UP[i] = 1
        % upstream sensor probably too rich
    ELSE(1a)
        % Testing signal shift to lean:
        LV_DIAG_AFR_SYM_PLAUS_LSL_UP[i] = 0
        IF(2a)   VLS_DOWN[i] >= C_VLS_DOWN_AFR_PLAUS_LSL_UP[i]
                     % downstream sensor signal rich
        AND(2a) LAMB_LS_UP[i] + LAMB_DELTA_LAM_ADJ[i] >
                     C_LAMB_AFL_PLAUS_LSL_UP[i]
                     % upstream sensor output lean
```

```
        THEN(2a) LV_DIAG_AFL_SYM_PLAUS_LSL_UP[i] = 1
                % upstream sensor probably too lean
        ELSE(2a)  LV_DIAG_AFL_SYM_PLAUS_LSL_UP[i] = 0
        ENDIF(2a)
    ENDIF(1a)
```

% First anti-bouncing filter (to eliminate the catalyst oxygen buffer effect):

```
    IF(1b) LV_DIAG_AFR_SYM_PLAUS_LSL_UP[i] = 1 OR
                        LV_DIAG_AFL_SYM_PLAUS_LSL_UP[i] = 1
    THEN(1b)
        IF(2b) MAF_SYM_RAW_PLAUS_LSL_UP[i]  < C_MAF_SYM_RAW_PLAUS_LSL_UP[i]
        THEN(2b)
            $MAF\_SYM\_RAW\_PLAUS\_LSL\_UP[i]_n = MAF\_SYM\_RAW\_PLAUS\_LSL\_UP[i]_{n-1} + MAF\_CYL * T\_SAMPLE[ms] * 1/3600 [g*h/(kg*ms)]$
        ELSE(2b)
            IF(3b) LV_DIAG_AFR_SYM_PLAUS_LSL_UP[i] = 1
            THEN(3b)
                STATE_SYM_DIAG_PLAUS_LSL_UP[i] = “SYM_0”
            ELSE(3b)
                STATE_SYM_DIAG_PLAUS_LSL_UP[i] = “SYM_1”
            ENDIF(3b)
        ENDIF(2b)
    ELSE(1b)
        $MAF\_SYM\_RAW\_PLAUS\_LSL\_UP[i]_n = MAF\_SYM\_RAW\_PLAUS\_LSL\_UP[i]_{n-1} - MAF\_CYL * T\_SAMPLE[ms] * 1/3600 [g*h/(kg*ms)]$
    ENDIF(1b)


    IF(1c) MAF_SYM_RAW_PLAUS_LSL_UP[i] <=
                        C_MAF_SYM_RAW_RST_PLAUS_LSL_UP[i]
    THEN(1c)
        STATE_SYM_DIAG_PLAUS_LSL_UP[i] = “NO_SYM”
    ENDIF(1c)
```

% Anti-bounce Counter and minimum time to have a valid diagnostic:

```
    IF(1a) LV_DIAG_AFR_SYM_PLAUS_LSL_UP[i] = 0
```

```
AND(1a) LV_DIAG_AFL_SYM_PLAUS_LSL_UP[i] = 0
THEN(1a) % "raw symptom" not present => no failure case
    IF(2a) STATE_SYM_DIAG_PLAUS_LSL_UP[i] =  “NO_SYM”
    THEN(2a)
            MAF_END_PLAUS_LSL_UP[i]$_{n}$ = MAF_END_PLAUS _LSL_UP[i]$_{n-1}$ + MAF_CYL
            * T_SAMPLE[ms]  * 1/3600 [g*h/(kg*ms)]
    ELSE(2a)
            MAF_END_PLAUS_LSL_UP[i]$_{n}$ = MAF_END_PLAUS_LSL_UP[i]$_{n-1}$
            MAF_SYM_PLAUS_LSL_UP[i]$_{n}$ = MAF_SYM_PLAUS_LSL_UP[i]$_{n-1}$
    ENDIF(2a)
ELSEIF(1a) LV_DIAG_AFR_SYM_PLAUS_LSL_UP[i] = 1
OR(1a) LV_DIAG_AFL_SYM_PLAUS_LSL_UP[i] = 1
THEN(1a) % "raw symptom" is present => failure case
    IF(2b) STATE_SYM_DIAG_PLAUS_LSL_UP[i] =  “NO_SYM”
    THEN(2b) % symptom flag is not yet set
            MAF_END_PLAUS_LSL_UP[i]$_{n}$ = MAF_END_PLAUS_LSL_UP[i]$_{n-1}$
            MAF_SYM_PLAUS_LSL_UP[i]$_{n}$ = MAF_SYM_PLAUS_LSL_UP[i]$_{n-1}$
    ELSE(2b) % symptom flag has been set
            MAF_SYM_PLAUS_LSL_UP[i]$_{n}$ = MAF_SYM_PLAUS_LSL_UP[i]$_{n-1}$ +
                MAF_CYL * T_SAMPLE[ms]  * 1/3600 [g*h/(kg*ms)]
    ENDIF(2b)
ENDIF(1a)


% Reset of MAF_SYM_PLAUS_LSL_UP
IF(1b)
    STATE_SYM_DIAG_PLAUS_LSL_UP[i]$_{n}$ <> STATE_SYM_DIAG_PLAUS_LSL_UP[i]$_{n-1}$
THEN(1b)
    MAF_SYM_PLAUS_LSL_UP[i] = 0
ENDIF(1b)
STATE_SYM_DIAG_PLAUS_LSL_UP[i]$_{n-1}$ = STATE_SYM_DIAG_PLAUS_LSL_UP[i]$_{n}$


% Failure determination and setting of end flag
IF(1c) MAF_END_PLAUS_LSL_UP[i] > C_MAF_SYM_PLAUS_LSL_UP[i] +
    C_MAF_SYM_RAW_PLAUS_LSL_UP[i]
    % To assure time symmetry between good (no failure) case and failure case, for the
good case the MAF integral has to reach the sum of the thresholds of the anti-bounce filters.
THEN(1c) % End-Bit is set in case no symptom present
    LV_DIAG_PLAUS_END_LSL_UP[i] = 1
```

```
ENDIF(1c)

IF(1d) MAF_SYM_PLAUS_LSL_UP[i] > C_MAF_SYM_PLAUS_LSL_UP[i]
THEN(1d) % End-bit is set in case symptom present
    LV_DIAG_PLAUS_SYM_LSL_UP[i] = 1
    LV_DIAG_PLAUS_END_LSL_UP[i] = 1
ENDIF(1d)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEG_MIN_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Minimum temperature of the catalyst to carry out the plausibility					
C_TEG_MAX_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Maximum temperature of the catalyst to carry out the plausibility					
C_MAF_KGH_MIN_DIAG_PLAUS_LSL_UP	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Minimum mass air flow to carry out the plausibility					
C_MAF_KGH_MAX_DIAG_PLAUS_LSL_UP	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Maximum mass air flow to carry out the plausibility					
C_N_MIN_DIAG_PLAUS_LSL_UP	1	0...FFH	0...8160	32	[rpm]
Lower engine speed admitted to carry out the plausibility					
C_VLS_DOWN_AFL_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lean-mixture threshold of the downstream sensor voltage					
C_VLS_DOWN_AFR_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Rich-mixture threshold of the downstream sensor voltage					
C_LAMB_AFL_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lean-mixture threshold of the upstream sensor voltage					
C_LAMB_AFR_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Rich-mixture threshold of the upstream sensor voltage					
C_TCO_MIN_DIAG_PLAUS_LSL_UP	1	0...FEH	-48...142.5	0.75	[°C]
Minimum threshold of the cooling-water temperature to activate the function					
C_MAF_SYM_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral threshold for detection of mutual not plausible sensor signals					
C_MAF_SYM_RAW_RST_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral threshold for reset of raw symptom					
C_MAF_SYM_RAW_PLAUS_LSL_UP[NC_CBK_EX_NR]	1	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral threshold for compensation of the catalyst delay					

16.57 Application Incidences for "O2 sensor (up, down) plausibility diag"

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_PLAUS_INH_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of "Plausibility check of the sensor signal"					
LV_INH_DIAG_RBM_PLAUS_LSL_UP[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Deactivation of Rate Base Monitoring of "Plausibility check of the sensor signal"					

Input data:

LV_IGK	LV_CDN_VB_OBD2	LV_ERR_MAF	LV_ERR_AMP
LV_ERR_SA_SAV	LV_ERR_AMP_PLAUS	LV_ERR_SA_SYS	LV_ERR_CAM_CUS
LV_ERR_SAV	LV_ERR_CHG_LS_UP	LV_ERR_TCO	LV_ERR_EL_CPS
LV_ERR_TPS	LV_ERR_CRK	LV_MIS_STATE_B	LV_ERR_DIAGCPS
LV_ERR_IVVT	NC_CBK_EX_NR	LV_ERR_MAP_DIP_SHIFT	LV_ERR_SAP
LV_ERR_SA_SAP	LV_ERR_MAP_DIP_SENS	LV_ERR_SA_SAV_LSL	LV_ERR_ECT_EL
LV_ERR_ECT_MEC	LV_ERR_MAP_DIP_PLAUS	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LOAD_TPS_PLAUS	LV_CAT_PURGE_ACT[NC_CBK_EX_NR]	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]
LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]
LV_ERR_LSH_LSL_UP[NC_CBK_EX_NR]	LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]
LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_TH	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_FUP
LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV	LV_ERR_FUP_ORNG
LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_SLV_IVVT_IN	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]
LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS		

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation: at every transition LV_IGK = 0->1or reset all variables shall be reset to 0.

at every transition LV_IGK = 1->0

LV_DIAG_PLAUS_INH_LSL_UP[NC_CBK_EX_NR] = 1

Recurrence: the functions shall be carried out once every 1s.

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

This function has to be deactivated, if at least one of the following errors / special modes is active.

IF
- LV_ERR_AMP = 1 **OR**
- LV_ERR_AMP_PLAUS = 1 **OR**
- LV_ERR_CRK = 1 **OR**
- LV_ERR_CAM_CUS = 1 **OR**
- LV_ERR_TCO = 1 **OR**
- LV_ERR_TPS = 1 **OR**
- LV_ERR_MAF = 1 **OR**
- LV_ERR_MAP_DIP_SENS = 1 **OR**
- LV_ERR_MAP_DIP_PLAUS = 1 **OR**
- LV_ERR_MAP_DIP_SHIFT = 1 **OR**
- LV_ERR_EL_CPS = 1 **OR**
- LV_ERR_DIAGCPS = 1 **OR**
- LV_ERR_SA_SAV = 1 **OR**
- LV_ERR_SA_SAV_LSL = 1 **OR**
- LV_ERR_SA_SAP = 1 **OR**
- LV_ERR_SA_SYS = 1 **OR**
- LV_ERR_SAV = 1 **OR**
- LV_ERR_SAP = 1 **OR**
- LV_ERR_IVVT = 1 **OR**
- LV_MIS_STATE_B = 1 **OR**
- LV_ERR_CHG_LS_UP = 1 **OR**
- LV_ERR_LOAD_TPS_PLAUS = 1 **OR**
- LV_ERR_LS_DOWN[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_ECT_EL = 1 **OR**
- LV_ERR_ECT_MEC = 1 **OR**
- LV_ERR_EL_LSL_UP[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_OC_LSL_UP[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_LSL_UP_IF[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_LSH_UP[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_LSH_LSL_UP[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR] = 1 **OR**
- LV_ERR_TH = 1 **OR**
- LV_ERR_DYN_VLD_LS_UP[i] = 1 **OR**
- LV_ERR_FSD_LAM_LIM[i] = 1 **OR**
- LV_ERR_FUP = 1 **OR**

```
      -  LV_ERR_FUP_MFP_PLAUS = 1                              OR
      -  LV_ERR_H_PRS_SYS = 1                                  OR
      -  LV_ERR_VCV  = 1                                       OR
      -  LV_ERR_FUP_ORNG = 1                                   OR
      -  LV_ERR_FUP_ST = 1                                     OR
      -  LV_ERR_TTIP_MES_LSH_UP[i]  = 1
      -
  THEN    LV_INH_DIAG_RBM_PLAUS_LSL_UP[NC_CBK_EX_NR]  =  1
  ELSE    LV_INH_DIAG_RBM_PLAUS_LSL_UP[NC_CBK_EX_NR]  =  0
 ENDIF

 IF   -  LV_INH_DIAG_RBM_PLAUS_LSL_UP[NC_CBK_EX_NR] = 1            OR
      -  LV_CAT_PURGE_ACT[NC_CBK_EX_NR] = 1                        OR
      -  LV_CDN_VB_OBD2 = 0                                        OR
      -  LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR] = 1                 OR
      -  LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR] = 1                 OR
      -  LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR] = 1                 OR
      -  LV_ERR_SLV_IVVT_IN = 1                                    OR
      -  LV_ERR_OFS_LSL_UP[i] = 1                                  OR
      -  LV_ERR_CTL_LSL_UP[i] = 1                                  OR
      -  LV_ERR_MAP_TPS_PLAUS = 1

 THEN     LV_DIAG_PLAUS_INH_LSL_UP[NC_CBK_EX_NR]  =  1
 ELSE     LV_DIAG_PLAUS_INH_LSL_UP[NC_CBK_EX_NR]  =  0
 ENDIF
```

16.58 Dynamic Fuel Trim Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_LAM_ADJ[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the lambda trim control errors					
LV_ERR_LAM_ADJ[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status diagnostic result of lambda trim control					
LV_END_DIAG_LAM_ADJ[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Diagnostic cycle for lambda trim control complete					
LV_CDN_DIAG_DELTA_I_LAM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean Flag to indicate the activation of trim controller limit diagnosis					
ERR_SYM_DELTA_I_LAM[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom of trim controller limit error					
LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Trim controller limit error					
LV_END_DIAG_DELTA_I_LAM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
First cycle of trim controller limit diagnosis finished					
LV_END_DIAG_WIN_DELTA_I_LAM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
End of diagnosis cycle for similar conditions window					
T_SUM_MAX_DELTA_I_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Total duration over C_LAMB_DELTA_I_MAX_DIAG					
T_SUM_MIN_DELTA_I_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Total duration over C_LAMB_DELTA_I_MIN_DIAG					
T_SUM_END_DIAG_WIN_DELTA_I_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter to set LV_END_DIAG_WIN_DELTA_I_LAM[i] for similar conditions window					
T_SUM_END_READY_DELTA_I_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Second time counter to set LV_END_DIAG_DELTA_I_LAM[i]					
T_SUM_RST_DELTA_I_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter to reset T_SUM_MAX_DELTA_I_LAM[i] and T_SUM_MIN_DELTA_I_LAM[i].					
LV_DET_SYM_EVE_DELTA_I_LAM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Bit for second time counter DELTA_I_LAM (T_SUM_END_READY_DELTA_I_LAM[i])					

LAMB_DELTA_I_SAVE_DIAG[NC_CBK_EX_NR]	V/O/S	F800...800H	-0.125...0.125	0.061e-3	[-]
I-share from trim controller - to be transmitted to the Scan Tool mode06					
LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Trim controller deviation error					
ERR_SYM_VLS_DOWN_DIF[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom of trim controller deviation error					
LV_CDN_DIAG_VLS_DOWN_DIF[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Boolean Flag to indicate the activation of trim controller deviation diagnosis					
LV_END_DIAG_VLS_DOWN_DIF[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Trim controller deviation diagnosis available					
MAF_INT_VLS_DIF_DLY_MAX[NC_CBK_EX_NR]	V	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral for upper VLS_DIF_DLY deviation					
MAF_INT_VLS_DIF_DLY_MIN[NC_CBK_EX_NR]	V	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral for lower VLS_DIF_DLY deviation					
MAF_INT_RST_VLS_DOWN_DIF_DIAG[NC_CBK_EX_NR]	V	0...FFFFH	0...1820.41666	0.0277778	[g]
Integral to start a new diagnostic cycle and to set the end flag					
LV_VLS_DIF_DLY_LDC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Status limited dynamic for the function "Monitoring Trim Controller Deviation"					
LV_VLS_DIF_DLY_LDC_N	V	0...1H	0...1	1	[-]
Status limited dynamic N					
LV_VLS_DIF_DLY_LDC_MAF	V	0...1H	0...1	1	[-]
Status limited dynamic MAF					
LV_VLS_DIF_DLY_LDC_LAM_MV[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Status limited dynamic FAC_LAM_MV					
MAF_INT_VLS_DIF_DLY_LDC[NC_CBK_EX_NR]	V	0...FFFFH	0...1389	0.0211948	[g]
integral of mass air flow since limited dynamic conditions fulfilled					
N_MMV_VLS_DIF_DLY_LDC	V	0...1FE0H	0...8160	1	[rpm]
floating mean value for limited engine speed					
MAF_MMV_VLS_DIF_DLY_LDC	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
floating mean value for limited MAF					
LAM_MMV_VLS_DIF_DLY_LDC[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
floating mean value for limited lambda deviation					
MAF_INT_OPM_STAT	V	0...FFFFH	0...1820.41666	0.0277778	[g]
Integral to start trim controller deviation diagnosis after changing in homogeneous combustion mode					

Input data:

LV_IGK	LV_ES	LV_ST_END	NC_CBK_EX_NR
N	MAF_CYL	MAF	MAF_KGH
LV_DIAG_EOL_END_LS_UP_DOWN[NC_CBK_EX_NR]	LV_INH_DIAG_DLY_LAM[NC_CBK_EX_NR]	LV_CP_RAMP_OPEN_ACT	LV_LAM_ADJ_I_ACT[NC_CBK_EX_NR]
LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]	LAMB_DELTA_AD_LAM_ADJ[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]	LV_LAM_ADJ_P_ACT[NC_CBK_EX_NR]
LV_INH_DIAG_DLY_LAM[NC_CBK_EX_NR]	EFF_CAT_DIAG[NC_CBK_EX_NR]	FAC_LAM_MV[NC_CBK_EX_NR]	OPM_AV

FUNCTION DESCRIPTION:

General information:

This function shall summarize the results of both dynamic fuel trim diagnosis specified in this module. Although this function has variable names which are according to the error management naming convention, it does not include an interface to the error manager.

Error Symptoms are defined as follows:

```
 ┌─┬─┬─┬─┐
 │ │ │ │ │
 └─┴─┴─┴─┘
  │  │  │  └─> AF too RICH      ( = SYM_0)
  │  │  └────> AF too LEAN      ( = SYM_1)
  │  └───────> (not used)
  └──────────> (not used)
```

Application conditions:

Initialisation: At transition LV_IGK = 0 to 1, ECU reset or at clearing error memory reset all variables to zero, except for the following:

T_SUM_RST_DELTA_I_LAM[i] = C_T_SUM_RST_MAX_DELTA_I_LAM[i]
T_SUM_END_DIAG_WIN_DELTA_I_LAM[i] = C_T_SUM_RST_MAX_DELTA_I_LAM[i]
T_SUM_END_READY_DELTA_I_LAM[i] = C_T_SUM_READY_DELTA_I_LAM[i]
LAMB_DELTA_I_SAVE_DIAG[i] shall not be initialized

Recurrence: 100 ms

Activation: -

Formula section:

```
If   LV_ERR_DELTA_I_LAM[i] = 1 or  LV_ERR_VLS_DOWN_DIF[i] = 1
Then
    If   (ERR_SYM_DELTA_I_LAM[i] = “SYM_0” &
          ERR_SYM_VLS_DOWN_DIF[i] <> “SYM_1”)
    Or   (ERR_SYM_DELTA_I_LAM[i] <> “SYM_1” &
          ERR_SYM_VLS_DOWN_DIF[i] = “SYM_0”)
    Then
            LV_ERR_LAM_ADJ[i]  = 1
            ERR_SYM_LAM_ADJ[i] = “SYM_0”
    Else
            If  (ERR_SYM_DELTA_I_LAM[i] = “SYM_1” &
                 ERR_SYM_VLS_DOWN_DIF[i] <> “SYM_0”)
```

```
                Or (ERR_SYM_DELTA_I_LAM[i] <> "SYM_0" &
                    ERR_SYM_VLS_DOWN_DIF[i] = "SYM_1")
                Then
                        LV_ERR_LAM_ADJ[i]  = 1
                        ERR_SYM_LAM_ADJ[i] = "SYM_1"
            Else
                        ERR_SYM_LAM_ADJ[i] = "NO_SYM"
                Endif
        Endif
    Endif


    If      LV_END_DIAG_DELTA_I_LAM[i] = 1 or
            LV_END_DIAG_VLS_DOWN_DIF[i]  = 1 or
        LV_DIAG_EOL_END_LS_UP_DOWN[i] = 1
    Then
        LV_END_DIAG_LAM_ADJ[i] = 1
    Endif
```

16.58.1 Monitoring trim controller I-Share

FUNCTION DESCRIPTION:

General information:

To fulfil the legal requirements of the OBDII lambda sensor diagnosis, the trim of the lambda control by the lambda sensor signal downstream of the catalytic converter must be monitored in addition to the diagnosis of the lambda sensor upstream of the catalytic converter. In the case of a trim control error, a plausibility check is performed to assign the faulty behavior to the control sensor or the monitoring sensor.
Faulty behavior of the lambda sensors upstream or downstream from the catalytic converter exists if one of the emission components exceeds the standard emissions in the emission test cycle by a certain factor.

Error Symptoms are defined as follows:

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │  │  │  └─> AF too RICH     ( = SYM_0)
  │  │  └───> AF too LEAN     ( = SYM_1)
  │  └─────> (not used)
  └───────> (not used)
```

Application conditions:

Initialisation: -- (see first chapter) --

Recurrence: 100ms

Activation / Deactivation:

```
If      LV_INH_DIAG_DLY_LAM[i] = 0
and     LV_CP_RAMP_OPEN_ACT  = 0
and     LV_LAM_ADJ_I_ACT[i] = 1
and     LV_ERR_DELTA_I_LAM[i] = 0
and     C_LAMB_DELTA_I_MAX_DIAG > C_LAMB_DELTA_I_MIN_DIAG
then    LV_CDN_DIAG_DELTA_I_LAM[i] = 1
           “diagnosis enabled”
else    LV_CDN_DIAG_DELTA_I_LAM[i] = 0
           “diagnosis disabled”
endif
```

Description:

16.58.2 Reset of time counters and set of end of diagnosis flags

After a calibrateable time during which the diagnosis has been activated and no error has been set, the timers for detection of too rich or lean system are reset or respectively reinitialised for the next diagnosis check. This must be done e.g. to avoid to sum up intermittend failures over a too long period.
Beside the items mentioned above the end of diagnosis flag can be set by two different time counters also. (T_SUM_END_DIAG_WIN_DELTA_I_LAM[i] and T_SUM_END_READY_DELTA_I_LAM[i])
After a calibrateable time during which the diagnosis has been activated the end of diagnosis flag is set. The flag of temporary end of diagnosis is a trigger for the similar conditions check and therefore always set after every completed DELTA_I_LAM cycle check (reset next recurrence).
At engine stop all time counters are reset respectively reinitialised.

Formula section:

```
If      LV_CDN_DIAG_DELTA_I_LAM[i] = 1
then    % Reset of time counters and set of end of diagnosis flags
        If      ERR_SYM_DELTA_I_LAM[i] = SYM_0 or SYM_1
        and     T_SUM_RST_DELTA_I_LAM[i] = C_T_SUM_RST_MAX_DELTA_I_LAM[i]
        then    T_SUM_END_DIAG_WIN_DELTA_I_LAM[i] =
                                        C_T_SUM_RST_MAX_DELTA_I_LAM[i]
```

```
            T_SUM_END_READY_DELTA_I_LAM[i] =
                                C_T_SUM_READY_DELTA_I_LAM[i]
        endif
        If   ERR_SYM_DELTA_I_LAM[i] = SYM_0 or SYM_1
        or   T_SUM_RST_DELTA_I_LAM[i] < C_T_SUM_RST_MAX_DELTA_I_LAM[i]
        then decrement T_SUM_RST_DELTA_I_LAM[i]
        endif
        decrement T_SUM_END_DIAG_WIN_DELTA_I_LAM[i]
        If   T_SUM_END_DIAG_WIN_DELTA_I_LAM[i] = 0
        then LV_END_DIAG_WIN_DELTA_I_LAM[i] = 1
             LV_END_DIAG_DELTA_I_LAM[i] = 1
             T_SUM_END_DIAG_WIN_DELTA_I_LAM[i] =
                                C_T_SUM_RST_MAX_DELTA_I_LAM[i]
        else LV_END_DIAG_WIN_DELTA_I_LAM[i] = 0
        endif
        If   T_SUM_MAX_DELTA_I_LAM[i] > C_T_SUM_MAX_ERR_DELTA_I_LAM[i]
        or   T_SUM_MIN_DELTA_I_LAM[i] > C_T_SUM_MAX_ERR_DELTA_I_LAM[i]
        then LV_DET_SYM_EVE_DELTA_I_LAM[i] = 1
        else LV_DET_SYM_EVE_DELTA_I_LAM[i] = 0
        endif
        decrement T_SUM_END_READY_DELTA_I_LAM[i]
        If   T_SUM_END_READY_DELTA_I_LAM[i] = 0
        and  LV_DET_SYM_EVE_DELTA_I_LAM[i] = 0
        and  LV_END_DIAG_DELTA_I_LAM[i] = 0
        then LV_END_DIAG_DELTA_I_LAM[i] = 1
             LV_END_DIAG_WIN_DELTA_I_LAM[i] = 1
             T_SUM_END_DIAG_WIN_DELTA_I_LAM[i] =
                                C_T_SUM_RST_MAX_DELTA_I_LAM[i]
        endif
        If   T_SUM_RST_DELTA_I_LAM[i] = 0
        and  LV_ERR_DELTA_I_LAM[i] = 0
        then T_SUM_MAX_DELTA_I_LAM[i] = 0
             T_SUM_MIN_DELTA_I_LAM[i] = 0
             T_SUM_RST_DELTA_I_LAM[i] = C_T_SUM_RST_MAX_DELTA_I_LAM[i]
        endif
   else freeze T_SUM_RST_DELTA_I_LAM[i]
        freeze T_SUM_END_DIAG_WIN_DELTA_I_LAM[i]
```

```
        freeze T_SUM_END_READY_DELTA_I_LAM[i]
        LV_END_DIAG_WIN_DELTA_I_LAM[i] = 0
        If      LV_ES = 1
        then    T_SUM_MAX_DELTA_I_LAM[i] = 0
                T_SUM_MIN_DELTA_I_LAM[i] = 0
                T_SUM_RST_DELTA_I_LAM[i] = C_T_SUM_RST_MAX_DELTA_I_LAM[i]
                T_SUM_END_DIAG_WIN_DELTA_I_LAM[i] =
                                        C_T_SUM_RST_MAX_DELTA_I_LAM[i]
                T_SUM_END_READY_DELTA_I_LAM[i] =
                                        C_T_SUM_READY_DELTA_I_LAM[i]
         endif
endif
```

16.58.3 Error detection for rich/lean A/F ratio deviation

If the total amount of I-share of the trim controller (LAMB_DELTA_I_LAM_ADJ[i] + LAMB_DELTA_AD_LAM_ADJ[i]) exceeds the diagnostic threshold C_LAMB_DELTA_I_MAX/MIN_DIAG[i], T_SUM_MAX/MIN_DELTA_I_LAM[i] is incremented as long as the diagnostic threshold is exceeded. If (LAMB_DELTA_I_LAM_ADJ[i] + LAMB_DELTA_AD_LAM_ADJ[i]) falls below the diagnostic threshold, then T_SUM_MAX/MIN_DELTA_I_LAM[i] is stopped. A failure is detected as soon as T_SUM_MAX/MIN_DELTA_I_LAM[i] is above the threshold C_T_SUM_MAX/MIN_MAX_DELTA_I_LAM[i]. Incrementation restarts, if the I-share of the trim controller exceeds the limit again.
If the engine is stopped the time counters shall be reset.

```
If      LV_CDN_DIAG_DELTA_I_LAM[i] = 1
then    % Error detection for rich/lean A/F ratio deviation
         If    ( LAMB_DELTA_I_LAM_ADJ[i] + LAMB_DELTA_AD_LAM_ADJ[i] ) >
                                              C_LAMB_DELTA_I_MAX_DIAG
         then       increment T_SUM_MAX_DELTA_I_LAM[i]
                    ERR_SYM_DELTA_I_LAM[i] = “SYM_1”       /AF too LEAN
         else       freeze T_SUM_MAX_DELTA_I_LAM[i]
                    If      ( LAMB_DELTA_I_LAM_ADJ[i] +
                     LAMB_DELTA_AD_LAM_ADJ[i] ) < C_LAMB_DELTA_I_MIN_DIAG
                    then    increment T_SUM_MIN_DELTA_I_LAM[i]
                            ERR_SYM_DELTA_I_LAM[i] = “SYM_0”      /AF too RICH
                    else    freeze T_SUM_MIN_DELTA_I_LAM[i]
                            ERR_SYM_DELTA_I_LAM[i] = “NO_SYM”  /no Symptom
                    endif
         endif
```

```
          If      T_SUM_MAX_DELTA_I_LAM[i] >
                                 C_T_SUM_MAX_MAX_DELTA_I_LAM[i]
          or      T_SUM_MIN_DELTA_I_LAM[i] >
                                 C_T_SUM_MIN_MAX_DELTA_I_LAM[i]
          then    LV_ERR_DELTA_I_LAM[i] = 1
                  LV_END_DIAG_DELTA_I_LAM[i] = 1
                  LV_END_DIAG_WIN_DELTA_I_LAM[i] = 1
                  LAMB_DELTA_I_SAVE_DIAG[i] = ( LAMB_DELTA_I_LAM_ADJ[i] +
                                                LAMB_DELTA_AD_LAM_ADJ[i] )
          endif
  else
        freeze ERR_SYM_DELTA_I_LAM[i]
        freeze T_SUM_MAX_DELTA_I_LAM[i]
        freeze T_SUM_MIN_DELTA_I_LAM[i]
  endif


  If    LV_END_DIAG_DELTA_I_LAM[i] = 0 -> 1
  then  LAMB_DELTA_I_SAVE_DIAG[i] = ( LAMB_DELTA_I_LAM_ADJ[i] +
                  LAMB_DELTA_AD_LAM_ADJ[i] )
  endif
```

Application hint: the data for each individual bank should fulfill the following conditions
1. C_T_SUM_RST_MAX_DELTA_I_LAM **>>** C_T_SUM_READY_DELTA_I_LAM
2. C_T_SUM_MIN_MAX_DELTA_I_LAM and C_T_SUM_MAX_MAX_DELTA_I_LAM **>>** C_T_SUM_MAX_ERR_DELTA_I_LAM

16.58.4 Monitoring of trim controller deviation

FUNCTION DESCRIPTION:

General information:

In order to get a fast decision of error location in case of oxygen sensor failure, trim controller input signal is monitored. Insufficient trim control over long period can be obtained out of this function. In conjunction with activity check of the up- and downstream sensors it is decided which sensor is damaged (up- or downstream).

Description:

The downstream sensor signal deviation to setpoint (VLS_DIF_LAM_ADJ[i]) is observed during trim controller diagnosis phase. This implements the activation of trim control.
If VLS_DIF_LAM_ADJ[i] exceeds either maximum or minimum voltage thresholds corresponding MAF integrals are started. As long as the diagnostic threshold is exceeded the MAF integral is incremented, else it is stopped. A failure is detected as soon as the MAF integral exceeds the threshold (MAF_INT_VLS_DIF_DLY_MAX/MIN). Then symptom flag LV_ERR_VLS_DOWN_DIF and the corresponding symptom (MIN or MAX) are set.
MAF_INT_RST_VLS_DOWN_DIF_DIAG[i] is incremented if no end flag is set so far, or if a symptom was detected at least once in the current diagnostic cycle, it seves two purposes: It causes a "good" diagnostic result if no symptom is detected after sufficient time has elapsed, and it aborts and restarts the diagnosis if a symptom was present, but occurs only rarely (e.g. due to dynamic driving conditions).

Error Symptoms are defined as follows:

```
 ┌──┬──┬──┬──┐
 └──┴──┴──┴──┘
  |  |  |  └─> AF too RICH     ( = SYM_0)
  |  |  └────> AF too LEAN     ( = SYM_1)
  |  └───────> (not used)
  └──────────> (not used)
```

Application conditions:

Initialisation: -- (see first chapter) –

Recurrence: 100ms

Activation:

```
If    $OPM\_AV_{n}$ <> $OPM\_AV_{n-1}$
then        MAF_INT_OPM_STAT = C_MAF_INT_OPM_STAT
else        $MAF\_INT\_OPM\_STAT_{n}$ =
            $MAF\_INT\_OPM\_STAT_{n-1}$
            - MAF_KGH* T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]
endif
```

```
If    LV_INH_DIAG_DLY_LAM[i] = 0
and   LV_CP_RAMP_OPEN_ACT  = 0
and   LV_LAM_ADJ_P_ACT[i] = 1
```

```
and LV_VLS_DIF_DLY_LDC[i]  = 1
and EFF_CAT_DIAG[i] < C_CAT_DIAG_MAX_DELTA_I_LAM[i]
and MAF_INT_OPM_STAT = 0
then
    LV_CDN_DIAG_VLS_DOWN_DIF[i] = 1
    “diagnosis enabled”
else
    LV_CDN_DIAG_VLS_DOWN_DIF[i] = 0
    “diagnosis disabled”
    “freeze all variables”
endif
```

Formula section:

```
If (1)MAF_INT_RST_VLS_DOWN_DIF_DIAG[i] <
          C_MAF_INT_RST_VLS_DOWN_DIF_DIAG[i]
then (1) % current diag cycle not too long without result
    If (2a)   VLS_DOWN[i] ≤ C_VLS_DOWN_MIN_DIAG_LAM_ADJ[i]
    then (2a) % too lean after catalyst
        ERR_SYM_VLS_DOWN_DIF[i] = “SYM_1”       /AF too LEAN
        If (3a)  MAF_INT_VLS_DIF_DLY_MAX[i] > C_MAF_INT_VLS_DIF_DLY_MAX[i]
        then (3a) % symptom present long enough – store the error
            LV_ERR_VLS_DOWN_DIF[i] = 1
            LV_END_DIAG_VLS_DOWN_DIF[i] = 1
        else (3b) % symptom present – set the symptom and count up, but don't store an error
            MAF_INT_VLS_DIF_DLY_MAX[i]$_{n}$ =
                MAF_INT_VLS_DIF_DLY_MAX[i]$_{n-1}$ +
                MAF_CYL* T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]
        endif (3a/b)
    else (2b) % not too lean after catalyst – do not touch the lean case, but check if it's too rich
        freeze  MAF_INT_VLS_DIF_DLY_MAX[i]
        if (3b)  VLS_DOWN[i] ≥ C_VLS_DOWN_MAX_DIAG_LAM_ADJ[i]
        then (3b) % too rich after catalyst
            ERR_SYM_VLS_DOWN_DIF[i] = “SYM_0”       /AF too RICH
            If  (4a) MAF_INT_VLS_DIF_DLY_MIN[i] >
            C_MAF_INT_VLS_DIF_DLY_MIN[i]
            then (4a) % symptom present long enough – store the error
                LV_ERR_VLS_DOWN_DIF[i] = 1
                LV_END_DIAG_VLS_DOWN_DIF[i] = 1
            else (4b) % symptom present – set the symptom and count up, but don't store an error
```

```
                MAF_INT_VLS_DIF_DLY_MIN[i]$_{n}$ =
                    MAF_INT_VLS_DIF_DLY_MIN[i]$_{n-1}$ +
                    MAF_CYL* T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]

            endif (4a/b)

        else (3b) % not too lean and not too rich – set the symptom (counting is done below)

            freeze MAF_INT_VLS_DIF_DLY_MIN[i]

            If (4c)  LV_ERR_VLS_DOWN_DIF[i] = 0
            Then (4c)

                ERR_SYM_VLS_DOWN_DIF[i] = “NO_SYM”

            endif (4c)

        endif (3a/b)

    endif (2a/b)

    % count a MAF integral to generate an end flag for the good case.

    If (2c)  LV_END_DIAG_VLS_DOWN_DIF[i] = 0
    or       MAF_INT_RST_VLS_DOWN_DIF_DIAG[i] > 0
    or       ERR_SYM_VLS_DOWN_DIF[i] <> “NO_SYM”
    then (2c)

        MAF_INT_RST_VLS_DOWN_DIF_DIAG[i]$_{n}$ =
            MAF_INT_RST_VLS_DOWN_DIF_DIAG[i]$_{n-1}$ +
            MAF_CYL* T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]

    else (2d)

        freeze MAF_INT_RST_VLS_DOWN_DIF_DIAG[i]

    endif (2c/d)

else (1)
% running too long without coming to an end.  Cancel a running diagnosis cycle by resetting the MAF integrals
(assuming a symptom is either not present at all or intermittent but rare).  Also indicate a "good" diag result, if no
end was reached so far.

    MAF_INT_RST_VLS_DOWN_DIF_DIAG[i] = 0

    MAF_INT_VLS_DIF_DLY_MIN[i] = 0

    MAF_INT_VLS_DIF_DLY_MAX[i] = 0

    If (2e) LV_END_DIAG_VLS_DOWN_DIF[i] = 0
    then (2e)

        LV_END_DIAG_VLS_DOWN_DIF[i] = 1

        LV_ERR_VLS_DOWN_DIF[i] = 0

        ERR_SYM_VLS_DOWN_DIF[i] = “NO_SYM”

    endif (2e)

endif (1)
```

16.58.5 Limited dynamic for the function "Monitoring the Trim Controller Deviation"

FUNCTION DESCRIPTION:

General information:

As it is specified in the function "Monitoring Trim Controller Deviation" contained in this module, this shall be carried out every time the Trim Controller P-share is beeing calculated. This is active most of the time to maintain the the air fuel ratio inside the catalyst window also if any deviation happens. Under such unstable condition its not proper to carry out the trim controller diagnosis and this shall be inhibit until the system achieves the steady state. This shall be facilitated by the present function which shall summarize informations concerning the dynamic of engine speed, mass air flow and air fuel ratio.

Description:

The floating mean value N_MMV_VLS_DIF_DLY_LDC is computed using the averaging constant C_N_CRLC_LDC_VLS_DIF_DLY. The limited dynamics condition only exist while the engine speed N lies within the dynamics window around N_MMV_VLS_DIF_DLY_LDC.
If the above limited dynamics condition is violated, the floating mean value N_MMV_VLS_DIF_DLY_LDC is set to the current engine speed in order to reach the limited dynamics condition faster:

The floating mean value MAF_MMV_VLS_DIF_DLY_LDC is computed using the averaging constant C_MAF_CRLC_LDC_VLS_DIF_DLY. The limited dynamics condition only exist while the air mass MAF stays within the dynamics window around MAF_MMV_VLS_DIF_DLY_LDC. If the above limited dynamics condition is violated, the floating mean value MAF_MMV_VLS_DIF_DLY_LDC is set to the current air-mass value in order to reach the limited dynamics condition faster:

The floating mean value LAM_MMV_VLS_DIF_DLY_LDC[i] is computed using the averaging constant C_LAM_MV_CRLC_LDC_VLS_DIF_DLY. The limited dynamics condition continues to exist while the mean oxygen value FAC_LAM_MV[i] stays within the dynamics window around LAM_MMV_VLS_DIF_DLY_LDC[i]. If the above limited dynamics condition is violated, the floating mean value LAM_MMV_VLS_DIF_DLY_LDC[i] is set to the current mean oxygen value in order to reach the limited dynamics condition faster.

Application conditions:

Initialisation: -- (see first chapter) –

Recurrence: T_SAMPLE = 100ms

Activation: If (LV_IGK = 1 and LV_ST_END = 1)

Deactivation: *the above conditions are not fullfiled*

Initialization at deactivation:
N_MMV_VLS_DIF_DLY_LDC = N
MAF_MMV_VLS_DIF_DLY_LDC = MAF
LAM_MMV_VLS_DIF_DLY_LDC[i] = FAC_LAM_MV[i]

Formula section:

Limited engine speed (N) dynamics:

```
N_MMV_VLS_DIF_DLY_LDC =
            N_MMV_VLS_DIF_DLY_LDC * (1 - C_N_CRLC_LDC_VLS_DIF_DLY) +
                                       C_N_CRLC_LDC_VLS_DIF_DLY * N

IF | N - N_MMV_VLS_DIF_DLY_LDC | < C_N_LDC_VLS_DIF_DLY
THEN  LV_VLS_DIF_DLY_LDC_N = 1
ELSE  LV_VLS_DIF_DLY_LDC_N = 0
      N_MMV_VLS_DIF_DLY_LDC = N
ENDIF
```

Limited mass air flow (MAF) dynamics:

```
MAF_MMV_VLS_DIF_DLY_LDC =
         MAF_MMV_VLS_DIF_DLY_LDC * (1 - C_MAF_CRLC_LDC_VLS_DIF_DLY) +
                                       C_MAF_CRLC_LDC_VLS_DIF_DLY * MAF

IF | MAF - MAF_MMV_VLS_DIF_DLY_LDC | < C_MAF_LDC_VLS_DIF_DLY

THEN  LV_VLS_DIF_DLY_LDC_MAF = 1
ELSE  LV_VLS_DIF_DLY_LDC_MAF = 0
        MAF_MMV_VLS_DIF_DLY_LDC = MAF
ENDIF
```

Limited mean oxygen value (FAC_LAM_MV[i]) dynamics:

```
LAM_MMV_VLS_DIF_DLY_LDC[i]  =
           LAM_MMV_VLS_DIF_DLY_LDC[i] * (1 -
C_LAM_MV_CRLC_LDC_VLS_DIF_DLY)
                 +  C_LAM_MV_CRLC_LDC_VLS_DIF_DLY * FAC_LAM_MV[i]


IF | FAC_LAM_MV[i] - LAM_MMV_VLS_DIF_DLY_LDC[i] | <
                                            C_LAM_MV_LDC_VLS_DIF_DLY
THEN  LV_VLS_DIF_DLY_LDC_LAM_MV[i] = 1
ELSE  LV_VLS_DIF_DLY_LDC_LAM_MV[i] = 0
      LAM_MMV_VLS_DIF_DLY_LDC[i]  = FAC_LAM_MV[i]
ENDIF
```

Summarizing limited dynamic conditions:

```
IF      LV_VLS_DIF_DLY_LDC_N = 1   AND   LV_VLS_DIF_DLY_LDC_MAF = 1   AND
           LV_VLS_DIF_DLY_LDC_LAM_MV[i] = 1
THEN
      IF     MAF_INT_VLS_DIF_DLY_LDC[i] ≤  C_MAF_INT_MIN_VLS_DIF_DLY_LDC[i]
      THEN
           MAF_INT_VLS_DIF_DLY_LDC[i]$_n$ =

                      MAF_INT_VLS_DIF_DLY_LDC[i]$_{n-1}$

                      + MAF_KGH* T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]
```

```
        ELSE
            freeze MAF_INT_VLS_DIF_DLY_LDC[i]
                LV_VLS_DIF_DLY_LDC[i] = 1
            ENDIF
    ELSE
            LV_VLS_DIF_DLY_LDC[i]                       =  0
            MAF_INT_VLS_DIF_DLY_LDC[i]                  =  0
    ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_DELTA_I_MAX_DIAG	1	F800...800H	-0.125...0.125	0.061e-3	[-]
Maximum limit of LAMB_DELTA_I_LAM_ADJ[i] + LAMB_DELTA_AD_LAM_ADJ[i]					
C_LAMB_DELTA_I_MIN_DIAG	1	F800...800H	-0.125...0.125	0.061e-3	[-]
Minimum limit of LAMB_DELTA_I_LAM_ADJ[i] + LAMB_DELTA_AD_LAM_ADJ[i]					
C_T_SUM_MAX_MAX_DELTA_I_LAM[NC_CBK_EX_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
Duration of T_SUM_MAX_DELTA_I_LAM[i] to detect an error					
C_T_SUM_MIN_MAX_DELTA_I_LAM[NC_CBK_EX_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
Duration of T_SUM_MIN_DELTA_I_LAM[i] to detect an error					
C_T_SUM_RST_MAX_DELTA_I_LAM[NC_CBK_EX_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
Time counter to reset T_SUM_MIN/MAX_DELTA_I_LAM[i]					
C_T_SUM_MAX_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
Time counter for maximum time of trim control over limit to set LV_END_DIAG_DELTA_I_LAM[i]					
C_T_SUM_READY_DELTA_I_LAM[NC_CBK_EX_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
Time counter for end of diagnosis					
C_VLS_DOWN_MAX_DIAG_LAM_ADJ[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Rich limit for VLS_DOWN for trim control diagnosis					
C_VLS_DOWN_MIN_DIAG_LAM_ADJ[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lean limit for VLS_DOWN for trim control diagnosis					
C_MAF_INT_VLS_DIF_DLY_MIN[NC_CBK_EX_NR]	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Maximum MAF integral of VLS_DIF_DLY deviation (rich side)					
C_MAF_INT_VLS_DIF_DLY_MAX[NC_CBK_EX_NR]	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Maximum MAF integral of VLS_DIF_DLY deviation (lean side)					
C_MAF_INT_RST_VLS_DOWN_DIF_DIAG[NC_CBK_EX_NR]	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Threshold for integral to start a new diagnostic cycle and to set the end flag					
C_CAT_DIAG_MAX_DELTA_I_LAM[NC_CBK_EX_NR]	1	0...FFH	0...1.99218	0.0078125	[-]
Maximum threshold for catalyst diagnosis result to perform the diagnosis					
C_N_LDC_VLS_DIF_DLY	1	0...1FE0H	0...8160	1	[rpm]
threshold limited dynamic engine speed					
C_MAF_LDC_VLS_DIF_DLY	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
threshold limited dynamic engine MAF					
C_LAM_MV_LDC_VLS_DIF_DLY	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
threshold limited dynamic lambda					
C_N_CRLC_LDC_VLS_DIF_DLY	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
correlation constant floating mean value calculation engine speed					
C_MAF_CRLC_LDC_VLS_DIF_DLY	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
correlation constant floating mean value calculation MAF					
C_LAM_MV_CRLC_LDC_VLS_DIF_DLY	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
correlation constant floating mean value calculation lambda					
C_MAF_INT_MIN_VLS_DIF_DLY_LDC[NC_CBK_EX_NR]	1	0...FFFFH	0...1389	0.0211948	[g]
Maximum value of the mass air flow integral after limited dynamic conditions were fulfilled					
C_MAF_INT_OPM_STAT	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Minimum value of mass air flow after changing in homogeneous combustion mode to allow diagnosis (VLS_DOWN_DIF)					

Configuration for diagnostic symptoms:

Diagnosis	Symptom	Nr	ABC type
Dynamic fuel trim I-share	*AF too RICH*	*0*	NO
	AF too LEAN	*1*	
DELTA_I_LAM[i]			

Diagnosis	Symptom	Nr	ABC type
Trim controller deviation	*AF too RICH*	*0*	NO
	AF too LEAN	*1*	
VLS_DOWN_DIF[i]			

16.59 O2 sensor fuel trim diag (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_RBM_DLY_LAM[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Inhibition of the dynamic fuel trim diagnosis due to failures					
LV_INH_DIAG_DLY_LAM[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibition condition for the dynamic fuel trim diagnosis					
STATE_RBM_VLS_DOWN_DIF[NC_CBK_EX_NR]	V/O	0...7H	0...7	1	[-]
RBM - Interface of VLS_DOWN_DIF monitor					
MAF_INT_SCAV_NOT_ACT	V	0...FFFFH	0...1820.41666	0.0277778	[g]
Integral to start fuel trim diagnosis after scavenging (Turbo)					

Input data:

LV_VB_CDN_OBD_2	LV_ERR_SA_SYS	LV_ERR_CAT_DIAG[NC_CBK_EX_NR]	
LV_DC	LV_ERR_SAP	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]	
LV_ERR_AMP	LV_ERR_TCO	LV_ERR_MAP_DIP_SENS	
LV_ERR_EL_CPS	LV_ERR_AMP_PLAUS	STATE_DIAG_ACT_LS_UP_DOWN[NC_CBK_EX_NR]	
LV_ERR_DIAGCPS	LV_IGK	LV_DIAG_EOL_REQ_LS_UP_DOWN[NC_CBK_EX_NR]	
NC_CBK_EX_NR	LV_MIS_STATE_B	LV_ERR_MAP_DIP_PLAUS	
LV_ERR_MAF	LV_VAR_LSH_DOWN	LV_ERR_MAP_DIP_SHIFT	
LV_ERR_SA_SAP	LV_NT_RGN_REQ	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	
LV_MIS_STATE_A	CTR_ERR_DYN_NR	LV_END_DIAG_VLS_DOWN_DIF[NC_CBK_EX_NR]	
LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS
MAF_SCAV_EXT	N	MAF_KGH	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_DIAG_DLY_LAM[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Manual inhibition of the dynamic fuel trim diagnosis (bank individual)					
C_MAF_SCAV_MAX_INH_DLY_LAM	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Maximum allowed MAF during scavenging to inhibit fuel trim diagnosis					
C_MAF_INT_SCAV_NOT_ACT	1	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral threshold to allow fuel trim diagnosis after scavenging					
C_N_MIN_DIAG_DLY_LAM	1	0...1FE0H	0...8160	1	[rpm]
Minimum engine speed to allow fuel trim diagnosis					

16.59.1 Inhibtion of “Dynamic fuel trim diagnosis”

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks. For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2), then:

i = 1 for exhaust cylinder bank 1
i = 2 for exhaust cylinder bank 2,
otherwise (NC_CBK_EX_NR = 1)
i = 1 for single exhaust cylinder bank.

Application conditions:

Initialisation: at reset all variables shall be set to zero;

at LV_IGK = 0 -> 1 all variables except *STATE_RBM_VLS_DOWN_DIF[i]* shall be set to zero;

at every LV_IGK = 1 -> 0 LV_INH_DIAG_DLY_LAM[i] = 1

Recurrence: 1 s

Activation: LV_IGK = 1

Dectivation: LV_IGK = 0

Formula section:

```
If      MAF_SCAV_EXT > C_MAF_SCAV_MAX_INH_DLY_LAM
then    MAF_INT_SCAV_NOT_ACT = C_MAF_INT_SCAV_NOT_ACT
else    MAF_INT_SCAV_NOT_ACT$_{n}$ =
        MAF_INT_SCAV_NOT_ACT$_{n-1}$
        - MAF_KGH* T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]
endif
```

% Inhibition of the diagnosis during scavenging (especially for turbocharged engines)

```
If          LV_ERR_MAP_DIP_SENS = 0
    and     LV_ERR_MAP_DIP_PLAUS = 0
    and     LV_ERR_MAP_DIP_SHIFT = 0
    and     LV_ERR_CAT_DIAG[i] = 0
    and     LV_ERR_SAP = 0
    and     LV_ERR_SA_SAP = 0
    and     LV_ERR_SA_SYS = 0
    and     LV_ERR_EL_CPS = 0
    and     LV_ERR_DIAGCPS = 0
    and     LV_ERR_TCO = 0
    and     LV_ERR_MAF = 0
    and     LV_MIS_STATE_B = 0
    and     LV_MIS_STATE_A = 0
    and     LV_ERR_FUP = 0
    and     LV_ERR_FUP_MFP_PLAUS = 0
    and     LV_ERR_H_PRS_SYS = 0
    and     LV_ERR_VCV = 0
    and     LV_ERR_FUP_ORNG = 0
    and     LV_ERR_FUP_ST = 0
    and     LV_ERR_TTIP_MES_LSH_UP[i] = 0
    and     LV_ERR_AMP = 0
    and     LV_ERR_AMP_PLAUS = 0
    and     LV_ERR_DYN_VLD_LS_UP[i] = 0
```

```
      and     LV_ERR_MAP_TPS_PLAUS = 0
  then        LV_INH_DIAG_RBM_DLY_LAM[i]  =  0
  else        LV_INH_DIAG_RBM_DLY_LAM[i]  =  1
endif

If            LV_VAR_LSH_DOWN = 1
      and     LC_INH_DIAG_DLY_LAM[i]  =  0
      and     LV_NT_RGN_REQ = 0
      and     LV_VB_CDN_OBD_2 = 1
      and     LV_LS_UP_OBD_2_MAN_DEAC[i] = 0
      and     LV_INH_DIAG_RBM_DLY_LAM[i] = 0
      and     STATE_DIAG_ACT_LS_UP_DOWN[i] = PASSIVE  or CHECK_FINSHED for all i
      and     LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 0  for all i
      and     MAF_INT_SCAV_NOT_ACT = 0
      and     N >= C_N_MIN_DIAG_DLY_LAM
  then        LV_INH_DIAG_DLY_LAM[i]  =  0
  else        LV_INH_DIAG_DLY_LAM[i]  =  1
endif
```

16.59.2 Interface for Rate – Based – Monitoring

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the VLS_DOWN_DIF[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_VLS_DOWN_DIF[i] data.

Within STATE_RBM_VLS_DOWN_DIF[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for trim diagnosis)

Application conditions:

Initialisation:

at LV_DC 0 -> 1 transition:

bit 0 , bit 1 of STATE_RBM_VLS_DOWN_DIF[i] = 0

bit 2 shall be set to 1.

at ECU reset:

all bits of STATE_RBM_VLS_DOWN_DIF[i] = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 -> 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_MAP_DIP_SHIFT
LV_ERR_SA_SAP	LV_ERR_SAP	LV_ERR_CAT_DIAG[NC_CBK_EX_NR]
LV_ERR_SA_SYS	LV_ERR_EL_CPS	LV_ERR_DIAGCPS
LV_ERR_TCO_EL	LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS
LV_ERR_TCO_STUCK	LV_ERR_MAF	LV_ERR_MIS[NC_CYL_NR]
LV_ERR_AMP	LV_ERR_AMP_PLAUS	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_TPS	LV_MIS_STATE_A	LV_MIS_STATE_B

LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TCO_STUCK_RNG	LV_ERR_TTIP_MES_LSH_UP[i]
LV_ERR_MAP_TPS_PLAUS			

```
If(1)     { CPU optimization at LV_DC 0 -> 1 transition }
          CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
          While  bit 1 of STATE_RBM_VLS_DOWN_DIF[i] = 0 do
                 with each XX failure of the above list  (depending on configuration, see
                 below):
                 ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                 SYNCHRONIZATION<CALL>)

                 If(2)        XX has a pending status
                 Then(2)      bit 1 of STATE_RBM_VLS_DOWN_DIF[i] = 1
                 Endif(2)
          Endwhile
Else(1)   { the dynamic failure memory is empty }      No action
Endif(1)
```

Every 1 s :

```
If    bit 0 of STATE_RBM_VLS_DOWN_DIF[i] = 0
Then
```

```
            If    LV_END_DIAG_VLS_DOWN_DIF[i] = 1
            Then  bit 0 of STATE_RBM_VLS_DOWN_DIF[i] = 1
            Endif
    Endif
    If    bit 1 of STATE_RBM_VLS_DOWN_DIF[i] = 0
    Then
            If    LV_INH_DIAG_RBM_DLY_LAM[i] = 1
            Then  bit 1 of STATE_RBM_VLS_DOWN_DIF[i] = 1
            Endif
    Endif
```

16.60 O2 sensor open loop diag

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>STATE_DIAG_ACT_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0H
1H
2H
3H</td><td>PASSIVE
RICH_CHECK
LEAN_CHECK
CHECK_ FINISHED</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">State of check of downstream oxygen sensor signal for lambda sensor error manager</td></tr>
<tr><td>MAF_INT_DIAG_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V</td><td>0...FFFFH</td><td>0...1820.41666</td><td>0.0277778</td><td>[g]</td></tr>
<tr><td colspan="6">integral of air mass flow while check is active</td></tr>
<tr><td>LV_DIAG_ACT_END_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Boolean flag, end of active check of the oxygen sensors</td></tr>
<tr><td>LV_DIAG_EOL_END_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Boolean flag, end of active check of the oxygen sensors (EOL test)</td></tr>
<tr><td>LV_DIAG_AFR_CDN_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Boolean flag indicating the condition to execute RICH active check are met</td></tr>
<tr><td>LV_DIAG_AFR_END_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Boolean flag indicating the end of RICH active check</td></tr>
<tr><td>LV_DIAG_AFL_CDN_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Boolean flag indicating the condition to execute LEAN active check are met</td></tr>
<tr><td>LV_DIAG_AFL_END_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Boolean flag indicating the end of LEAN active check</td></tr>
<tr><td>LAMB_SP_BEG_DIAG_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0...7FFFH</td><td>0...31.99902</td><td>0.9766e-3</td><td>[-]</td></tr>
<tr><td colspan="6">Inicial value of lambda set point</td></tr>
<tr><td>LV_LAMB_SP_AFR_REQ_DIAG_ACT[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Trigger function to change lambda set point for AFR active plausibility test</td></tr>
<tr><td>LV_LAMB_SP_AFL_REQ_DIAG_ACT[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Trigger function to change lambda set point for AFL active plausibility test</td></tr>
<tr><td>T_DIAG_ACT_AFR_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V</td><td>0...FFFFH</td><td>0...6553.5</td><td>0.1</td><td>[s]</td></tr>
<tr><td colspan="6">Timer for plausibility test under rich AF conditions</td></tr>
<tr><td>T_DIAG_ACT_AFL_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V</td><td>0...FFFFH</td><td>0...6553.5</td><td>0.1</td><td>[s]</td></tr>
<tr><td colspan="6">Timer for plausibility test under lean AF conditions</td></tr>
<tr><td>LV_LAM_DI_REQ[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Request disable close loop operation of the lambda controller</td></tr>
<tr><td>LV_DIAG_ACT_CDN_LS_UP_DOWN[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Conditions to perform rich/lean check of open loop diag are fulfilled - not relevant for soft check</td></tr>
<tr><td>LV_DIAG_AFR_CHK_SYM_LSL_UP[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Debounced failure in the upstream sensor during the RICH active check</td></tr>
<tr><td>LV_DIAG_AFL_CHK_SYM_LSL_UP[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Debounced failure in the upstream sensor during the LEAN active check</td></tr>
<tr><td>LV_DIAG_ACT_CHK_CDN_LSL_UP[NC_CBK_EX_NR]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Conditions to run active test of upstream sensor fulfilled</td></tr>
</table>

LV_DIAG_ACT_CHK_END_LSL_UP[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
End active test of upstream sensor					
CTR_SUM_RST_AFR_CHK_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Reset counter for the rich active test					
CTR_SUM_RST_AFL_CHK_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Reset counter for the rich active test					
CTR_AFR_CHK_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Debounce counter for the rich test					
CTR_AFL_CHK_LSL_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Debounce counter for the lean test					
LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounced failure in the downstream sensor					
ERR_SYM_CHK_LS_DOWN[NC_CBK_EX_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom in the downstream sensor					
LV_CDN_DIAG_CHK_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Conditions to run active test of downstream sensor fulfilled					
LV_DIAG_CYC_END_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
End of cycle of active test of downstream sensor					
LV_END_DIAG_CHK_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
End active test of downstream sensor					
LV_ERR_CHG_LS_DOWN	V/O	0...1H	0...1	1	[-]
Debounced failure of downstream lambda sensors interchanged					
ERR_SYM_CHG_LS_DOWN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom for downstream lambda sensors interchanged					
LV_CDN_DIAG_CHG_LS_DOWN	V	0...1H	0...1	1	[-]
Condition to run the diagnosis are fullfiled					
LV_END_DIAG_CHG_LS_DOWN	V	0...1H	0...1	1	[-]
End of downstream oxygen sensors interchanged test					
T_END_DIAG_ACT_LS_DOWN	V	0...FFFFH	0...6553.5	0.1	[s]
Timer for the end of the downstream sensor active test in the no failure case					
LV_SYM_AFL_CHK_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Symptom of oxygen sensor downstream too lean in case of 2 bank system during end of line test or sensor interchanged test					
LV_SYM_AFR_CHK_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Symptom of oxygen sensor downstream too rich in case of 2 bank system during end of line test or sensor interchanged test					
T_AFR_CHK_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Debounce counter for the rich test					
T_AFL_CHK_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Debounce counter for the lean test					
LV_DIAG_EOL_DC_LS_UP_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag indicating eol test was active in this driving cycle.					
LAMB_SP_DELTA_LS_UP_DOWN[NC_CBK_EX_NR]	V	8000...7FFFH	-32...31.99902	0.9766e-3	[-]
Increment of of lambda set point					

LAMB_SP_DIAG_OPL_LS_UP_DOWN[NC_CBK_EX_NR]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
lambda setpoint requested by open-loop test for upstream and downstream lambda sensors					

Input data:

LV_IGK	LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[NC_CBK_EX_NR]	LV_DIAG_OPL_REQ_CBK[NC_CBK_EX_NR]	LV_DIAG_EOL_REQ_LS_UP_DOWN[NC_CBK_EX_NR]
NC_CBK_EX_NR	LV_DIAG_ACT_INH_LS_UP_DOWN[NC_CBK_EX_NR]	MAF_CYL	LV_END_DIAG_LAM_ADJ[NC_CBK_EX_NR]
LAMB_SP[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]	LV_LAM_LSCL[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]
LV_INH_DIAG_CHG_LS_DOWN	LV_ST_END		

16.60.1 State machine

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Due to possible oxygen sensor defects (e.g. reference air poisoning) or faults in the injection system (e.g. leaking fuel injector), the oxygen sensors may not provide the expected lean or rich AF signal level during definite conditions (lean or rich). Hence, the oxygen sensor signals shall be checked for plausibility during this engine operating states. The handling is requested and results are utilized by a separate manager for oxygen sensors error.

Description:

If the following conditions are met, then the conditions for plausibility check at lean conditions are determined to be present:
A. The Fuel Trim Diagnosis has detected an error (transmitted by LV_DIAG_OPL_REQ_CBK[1]).
B. No general inhibition observed (LV_DIAG_ACT_INH_LS_UP_DOWN[i]) and no specific inihibition exists (LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[i]= 0).
C. The lean state shall be determined to be stable, as determined by comparing the mass air flow integral calculated after activation (MAF_INT_DIAG_LS_UP_DOWN[i]) with the threshold range (C_MAF_INT_DIAG_LS_UP_DOWN[i]).
D. The lean condition will be maintained until a timer is elapsed or an error was found on the upstream sensor or in case of a 1 bank system also on the downstream sensor.

The rich check follows always the lean test. The procedure is analogous to the lean check.

In case of a two bank system and finished open loop test on bank 1 and if there was no upstream linear lambda sensor fault detected, the bank 2 will be checked (triggered by LV_DIAG_OPL_REQ_CBK[2] which is set in application incidences).

In case an EOL test the timers shall be initialized with a non-zero calibrateable value to reduce the minimum required time. The same is valid for the initialization of the MAF integral.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, by reset, by clearing error memory or by triggering the EOL test (LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 0 -> 1; while LV_DIAG_ACT_END_LS_UP_DOWN[i] = 0) initialise the variables as follows:

STATE_DIAG_ACT_LS_UP_DOWN[i] = 0
MAF_INT_DIAG_LS_UP_DOWN[i] = 0
LV_DIAG_ACT_END_LS_UP_DOWN[i] = 0
LV_DIAG_EOL_END_LS_UP_DOWN[i] = 0
LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 0
LV_DIAG_AFR_END_LS_UP_DOWN[i] = 0
LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 0
LV_DIAG_AFL_END_LS_UP_DOWN[i] = 0
LAMB_SP_BEG_DIAG_LS_UP_DOWN[i] = 1
LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 0
LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 0
T_DIAG_ACT_AFR_LS_UP_DOWN[i] = 0
T_DIAG_ACT_AFL_LS_UP_DOWN[i] = 0
LV_LAM_DI_REQ[i] = 0

Recurrence: *T_SAMPLE* = 100ms

> *Note:*
> *The functions contained in this module shall be carried out according the following sequence:*
> 1. *active test upstream sensor*
> 2. *active test downstream sensor*
> 3. *state machine*

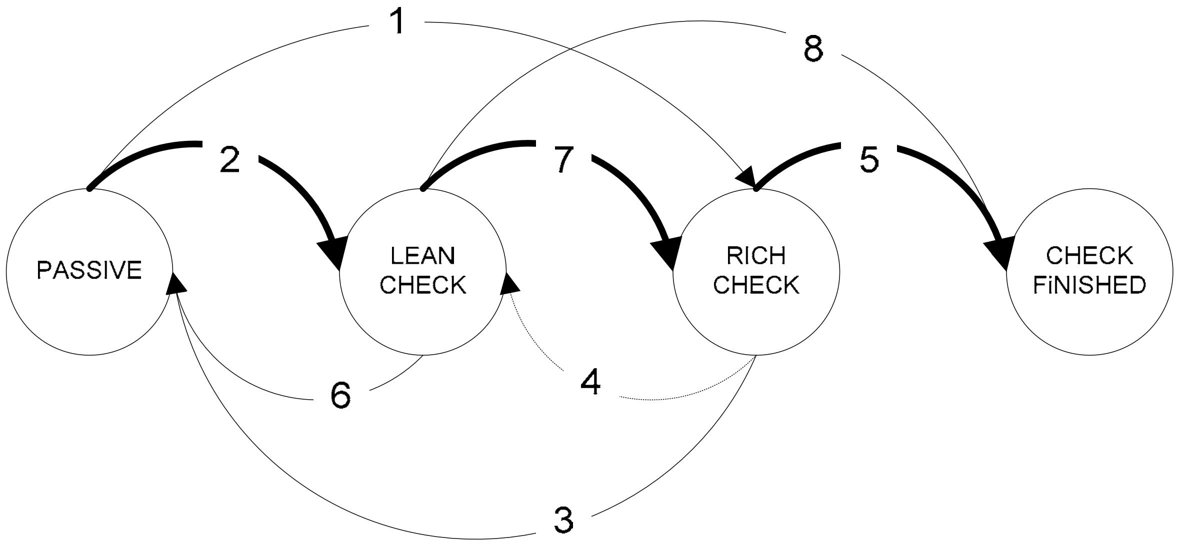
Activation:

LV_ST_END = 1

16.60.1.1 Sequence for check of signal plausibility monitoring (state machine)

The state machine shall remain in its current state and carry out the actions specified to occur within that state once per recurrence unless otherwise specified. The state machine shall only move to another state when one of the conditions has been determined to be met.

The priority of the conditions to change between states shall be defined by the order in which these conditions are listed within the appropiate state as described below.



STATE machine diagram:

```
If      LV_DIAG_ACT_INH_LS_UP_DOWN[i] = 0
and     LV_DIAG_ACT_END_LS_UP_DOWN[i] = 0
and     [ ( LV_DIAG_OPL_REQ_CBK[i] = 1 & LC_DIAG_AUTH_CHK_LS_UP_DOWN = 1)
        or
        LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1 ]
then
        LV_DIAG_ACT_CDN_LS_UP_DOWN[i] = 1
else
        LV_DIAG_ACT_CDN_LS_UP_DOWN[i] = 0
endif
```

16.60.1.2 State "PASSIVE"

Actions:

none

Condition 1: Transition to state "RICH CHECK"

```
If      LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[i] = 0
and     LV_DIAG_AFR_END_LS_UP_DOWN[i] = 0
and     LV_DIAG_AFL_END_LS_UP_DOWN[i] = 1
and     LV_DIAG_ACT_CDN_LS_UP_DOWN[i] = 1
then
```

```
        STATE_DIAG_ACT_LS_UP_DOWN[i] = "RICH CHECK"
        LAMB_SP_BEG_DIAG_LS_UP_DOWN[i] = LAMB_SP[i]
        LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 1
        LV_LAM_DI_REQ[i] = 1
        If  LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
        Then
                MAF_INT_DIAG_LS_UP_DOWN[i]  = C_MAF_INT_EOL_DIAG_LS_UP_DOWN[i]
                T_DIAG_ACT_AFL_LS_UP_DOWN[i]   = C_T_DIAG_EOL_LS_UP_DOWN
        Else
                T_DIAG_ACT_AFL_LS_UP_DOWN[i]   = 0
                MAF_INT_DIAG_LS_UP_DOWN[i] = 0
        endif
    endif
```

Condition 2: Transition to state "LEAN CHECK"

```
    If          LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[i] = 0
    and         LV_DIAG_AFL_END_LS_UP_DOWN[i] = 0
    and ( LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1        or
                LV_DIAG_OPL_REQ_CBK[i] = 1 )
    and LV_DIAG_ACT_CDN_LS_UP_DOWN[i] = 1
    then
        STATE_DIAG_ACT_LS_UP_DOWN[i] = "LEAN CHECK"
        LAMB_SP_BEG_DIAG_LS_UP_DOWN[i] = LAMB_SP[i]
        LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 1
        LV_LAM_DI_REQ[i] = 1        % disable lambda ctr. close loop operation
        If  LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
        Then
                MAF_INT_DIAG_LS_UP_DOWN[i]  = C_MAF_INT_EOL_DIAG_LS_UP_DOWN[i]
                T_DIAG_ACT_AFL_LS_UP_DOWN[i]   = C_T_DIAG_EOL_LS_UP_DOWN
        Else
                T_DIAG_ACT_AFL_LS_UP_DOWN[i]   = 0
                MAF_INT_DIAG_LS_UP_DOWN[i] = 0
        endif
    endif
```

16.60.1.3 State "RICH CHECK"

Description:

The MAF integral runs. After exceeding the calibrateable threshold for MAF the signal voltage is observed and the check is complete. The corresponding diagnosis flag is set in case of not sufficient voltage value.
The calculation of MAF_INT_DIAG_LS_UP_DOWN[i] expires in 100ms raster as long as the operation state is fulfilled. MAF_INT_DIAG_LS_UP_DOWN[i] is limited to the maximum value if necessary in case of a longer rich phase.

Actions:

```
If     LV_LAM_LSCL[i] = 0
and LAMB_SP[i]  > C_LAMB_AFR_MIN_DIAG_LS_UP_DOWN
and LAMB_SP[i]  < C_LAMB_AFR_MAX_DIAG_LS_UP_DOWN
then
    MAF_INT_DIAG_LS_UP_DOWN[i] $_{n}$ = MAF_INT_DIAG_LS_UP_DOWN[i] $_{n-1}$ + MAF_CYL* T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]
else
    If  LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
    Then
            MAF_INT_DIAG_LS_UP_DOWN[i]  = C_MAF_INT_EOL_DIAG_LS_UP_DOWN[i]
    else
            MAF_INT_DIAG_LS_UP_DOWN[i] = 0
    endif
endif


If    MAF_INT_DIAG_LS_UP_DOWN[i]  >  C_MAF_INT_DIAG_LS_UP_DOWN[i]
then        keep MAF_INT_DIAG_LS_UP_DOWN[i]
    If      T_DIAG_ACT_AFR_LS_UP_DOWN[i] < C_T_MAX_DIAG_AFR_LS_UP_DOWN
    then    active test runs
            LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 1
            If    LV_ERR_CHK_LS_DOWN[i] = 1
            or  LV_DIAG_AFR_CHK_SYM_LSL_UP[i] = 1
            then
                    LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 0
                    go to "transition to check finished"
            else
                    Increment T_DIAG_ACT_AFR_LS_UP_DOWN[i]
            endif
    else    active test timed out
            LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 0
            LV_DIAG_AFR_END_LS_UP_DOWN[i] = 1
            keep T_DIAG_ACT_AFR_LS_UP_DOWN[i]

            if    LV_DIAG_AFL_END_LS_UP_DOWN[i] = 1
            then   go to "transition to check finished"
```

```
                else    go to "transition to lean check"
                endif
        endif
else
        LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 0
        If      T_DIAG_ACT_AFR_LS_UP_DOWN[i]  < C_T_MAX_DIAG_AFR_LS_UP_DOWN
        Then
                If  LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
                Then
                        T_DIAG_ACT_AFR_LS_UP_DOWN[i]  = C_T_DIAG_EOL_LS_UP_DOWN
                else
                        T_DIAG_ACT_AFR_LS_UP_DOWN[i]  = 0
                endif
        else
                Keep T_DIAG_ACT_AFR_LS_UP_DOWN[i]
        endif
endif
```

<u>Condition 3:</u> Transition to state "PASSIVE"

```
If  LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[i] = 1
or  LV_DIAG_ACT_CDN_LS_UP_DOWN[i] = 0
then
        LV_LAM_DI_REQ[i] = 0        % re-enable lambda ctr. close loop operation
        LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 0
        LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 0
        STATE_DIAG_ACT_LS_UP_DOWN[i] = " PASSIVE"
endif
```

<u>Condition 4:</u> Transition to state "LEAN CHECK"

```
If transition to lean test
then
        MAF_INT_DIAG_LS_UP_DOWN[i]  = 0
        LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 0
        LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 1
        LAMB_SP_BEG_DIAG_LS_UP_DOWN[i] = LAMB_SP[i]
        STATE_DIAG_ACT_LS_UP_DOWN[i] = "LEAN CHECK "
endif
```

Condition 5: Transition to state "CHECK FINISHED"

```
If  transition to check finished
Then
    LV_LAM_DI_REQ[i] = 0        % re-enable lambda ctr. close loop operation
    LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 0
    LV_DIAG_ACT_END_LS_UP_DOWN[i] = 1
    STATE_DIAG_ACT_LS_UP_DOWN[i] = " CHECK FINISHED "
    If  LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
    then
        LV_DIAG_EOL_END_LS_UP_DOWN[i] = 1
    endif
endif
```

16.60.1.4 State "LEAN CHECK"

Description:

The MAF integral runs. After exceeding the calibrateable threshold for MAF the signal voltage is observed and the check is complete. The corresponding diagnosis flag is set in case of not sufficient voltage value.
The calculation of MAF_INT_DIAG_LS_UP_DOWN[i] expires in 100ms raster as long as the operation state is fulfilled. MAF_INT_DIAG_LS_UP_DOWN[i] is limited to the maximum value if necessary in case of a longer rich phase.

Actions

```
If      LV_LAM_LSCL[i] = 0
and LAMB_SP[i]  > C_LAMB_AFL_MIN_DIAG_LS_UP_DOWN
and LAMB_SP[i]  < C_LAMB_AFL_MAX_DIAG_LS_UP_DOWN
then
MAF_INT_DIAG_LS_UP_DOWN[i]$_n$ = MAF_INT_DIAG_LS_UP_DOWN[i]$_{n-1}$ + MAF_CYL * T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]
else
    If  LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
    Then
        MAF_INT_DIAG_LS_UP_DOWN[i]  = C_MAF_INT_EOL_DIAG_LS_UP_DOWN[i]
    else
        MAF_INT_DIAG_LS_UP_DOWN[i] = 0
    endif
endif
```

```
If    MAF_INT_DIAG_LS_UP_DOWN[i]  > C_MAF_INT_DIAG_LS_UP_DOWN[i]
then        keep MAF_INT_DIAG_LS_UP_DOWN[i]
      If    T_DIAG_ACT_AFL_LS_UP_DOWN[i] < C_T_MAX_DIAG_AFL_LS_UP_DOWN
      then  active test runs
            LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 1
            If   LV_ERR_CHK_LS_DOWN[i] = 1
            or  LV_DIAG_AFL_CHK_SYM_LSL_UP[i] = 1
            then
                  LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 0
                  go to "transition to check finished"
            else
                  Increment T_DIAG_ACT_AFL_LS_UP_DOWN[i]
            endif
      else  active test timed out
            LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 0
            LV_DIAG_AFL_END_LS_UP_DOWN[i] = 1
            keep T_DIAG_ACT_AFL_LS_UP_DOWN[i]

            If   LV_DIAG_AFR_END_LS_UP_DOWN[i] = 1
            then   go to "transition to check finished"
            else    go to "transition to rich check"
            endif
      endif
else
      LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 0
      If        T_DIAG_ACT_AFL_LS_UP_DOWN[i] < C_T_MAX_DIAG_AFL_LS_UP_DOWN
      then
            If  LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
            Then
                  T_DIAG_ACT_AFL_LS_UP_DOWN[i]  = C_T_DIAG_EOL_LS_UP_DOWN
            else
                  T_DIAG_ACT_AFL_LS_UP_DOWN[i]  = 0
            endif
      else
            Keep T_DIAG_ACT_AFL_LS_UP_DOWN[i]
      endif
endif
```

```
Condition 6: Transition to state "PASSIVE"
If  LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[i] = 1
or   LV_DIAG_ACT_CDN_LS_UP_DOWN[i] = 0
then
     LV_LAM_DI_REQ[i] = 0       % re-enable lambda ctr. close loop operation
     LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 0
     LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 0
     STATE_DIAG_ACT_LS_UP_DOWN[i] = "PASSIVE"
endif

Condition 7: Transition to state "RICH CHECK"
If transition to rich test
then
     MAF_INT_DIAG_LS_UP_DOWN[i] = 0
     LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 0
     LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 1
     LAMB_SP_BEG_DIAG_LS_UP_DOWN[i] = LAMB_SP[i]
     STATE_DIAG_ACT_LS_UP_DOWN[i] = " RICH CHECK "
endif

Condition 8: Transition to state "CHECK FINISHED"
If  transition to check finished
Then
     LV_LAM_DI_REQ[i] = 0       % re-enable lambda ctr. close loop operation
     LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 0
     LV_DIAG_ACT_END_LS_UP_DOWN[i] = 1
     STATE_DIAG_ACT_LS_UP_DOWN[i] = "CHECK FINISHED"
     If  LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
     Then
          LV_DIAG_EOL_END_LS_UP_DOWN[i] = 1
     endif
endif
```

16.60.1.5 State "CHECK FINISHED"

Action:

none

16.60.2 O2 Sensor (up) open loop diag

FUNCTION DESCRIPTION:

This function shall do a plausibility test of the upstream oxygen sensor after the lambda set point was shifted to rich and/or lean. The failure entry will be done by the O2 Sensor diagnosis management (characteristic shift down diagnosis). Consequently the flags LV_ERR_.... and LV_END_DIAG_... are defined there.

Description:

Application conditions:

Initialisation:

At LV_IGK change 0 -> 1, reset or at clearing error memory initialize the variables as follows:

LV_DIAG_AFR_CHK_SYM_LSL_UP[i] = 0
LV_DIAG_AFL_CHK_SYM_LSL_UP[i] = 0
LV_DIAG_ACT_CHK_CDN_LSL_UP[i] = 0
LV_DIAG_ACT_CHK_END_LSL_UP[i] = 0
CTR_SUM_RST_AFR_CHK_LSL_UP[i] = 0
CTR_SUM_RST_AFL_CHK_LSL_UP[i] = 0
CTR_AFR_CHK_LSL_UP[i] = 0
CTR_AFL_CHK_LSL_UP[i] = 0

Recurrence:

This function shall be carried out every 100 ms.

Activation:

```
If LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 1 or LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 1
then
    LV_DIAG_ACT_CHK_CDN_LSL_UP[i] = 1
    Activate function
Else
    LV_DIAG_ACT_CHK_CDN_LSL_UP[i] = 0
    Deactivate function
endif
```

Formula section:

```
If LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 1
And CTR_SUM_RST_AFR_CHK_LSL_UP[i] < C_CTR_SUM_RST_AFR_CHK_LSL_UP
then
    If      LAMB_LS_UP[i] > C_LAMB_AFR_THD_DIAG_ACT_LSL_UP
    Then
            If CTR_AFR_CHK_LSL_UP[i] < C_ABC_MAX_CHK_LSL_UP
            Then
                    Increment  CTR_AFR_CHK_LSL_UP[i] with C_ABC_INC_CHK_LSL_UP
            Else
                    LV_DIAG_AFR_CHK_SYM_LSL_UP[i]= 1
                    % sensor shows a shift to lean
                    LV_DIAG_ACT_CHK_END_LSL_UP[i]= 1
            Endif
    Else
            stop CTR_AFR_CHK_LSL_UP[i]
    Endif

    If CTR_AFR_CHK_LSL_UP[i] > 0
    Then
            Increment CTR_SUM_RST_AFR_CHK_LSL_UP_i
    Endif
Else
   CTR_SUM_RST_AFR_CHK_LSL_UP[i]= 0
   CTR_AFR_CHK_LSL_UP[i] = 0
Endif


If    LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 1
And CTR_SUM_RST_AFL_CHK_LSL_UP[i] < C_CTR_SUM_RST_AFL_CHK_LSL_UP
then
    If      LAMB_LS_UP[i] < C_LAMB_AFL_THD_DIAG_ACT_LSL_UP
    Then
            If CTR_AFL_CHK_LSL_UP[i] < C_ABC_MAX_CHK_LSL_UP
            Then
                    Increment  CTR_AFL_CHK_LSL_UP[i] with C_ABC_INC_CHK_LSL_UP
            Else
                    LV_DIAG_AFL_CHK_SYM_LSL_UP[i]= 1
                    % sensor shows a shift to lean
                    LV_DIAG_ACT_CHK_END_LSL_UP[i]= 1
            Endif
    Else
            stop CTR_AFL_CHK_LSL_UP[i]
    Endif

    If CTR_AFL_CHK_LSL_UP[i] > 0
    Then
            Increment CTR_SUM_RST_AFL_CHK_LSL_UP[i]
    Endif
Else
```

```
        CTR_SUM_RST_AFL_CHK_LSL_UP[i]= 0
        CTR_AFL_CHK_LSL_UP[i] = 0
    Endif
```

16.60.3 O2 sensor (down) open loop diag

FUNCTION DESCRIPTION:

General information:

This function shall do a plausibility test of the downstream oxygen sensor after the lambda set point was shifted to rich and/or lean. Once the test is complete on all cylinder banks, the symptoms are combined to produce appropriate error entries. The evaluation is different for 1 bank and 2 bank systems, more than 2 banks are not supported.

Here the dynamic error managemer is used to set the flags LV_ERR_.... and LV_END_DIAG_.... LV_DIAG_CYC_END_LS_DOWN[i] is defined as dummy for global interface reasons.

Description:

Error Symptoms are defined as follows:

```
ERR_SYM_CHK_LS_DOWN
  □□□□
   │ │ │ └─> downstream oxygen sensor too rich          ( = SYM_0)
   │ │ └───> downstream oxygen sensor too lean          ( = SYM_1)
   │ └─────> (not used)                                         ( = SYM_2)
   └───────> (not used)                                         ( = SYM_3)

ERR_SYM_CHG_LS_DOWN
  □□□□
   │ │ │ └─> (not used)                                         ( = SYM_0)
   │ │ └───> (not used)                                         ( = SYM_1)
   │ └─────> (not used)                                         ( = SYM_2)
   └───────> downstream oxygen sensors interchanged   ( = SYM_3)
```

Application conditions:

Initialisation:

At LV_IGK change 0 -> 1, reset or at clearing error memory initialize the variables as follows:

LV_ERR_CHK_LS_DOWN[i] = 0
ERR_SYM_CHK_LS_DOWN[i] = 0
LV_CDN_DIAG_CHK_LS_DOWN[i] = 0
LV_DIAG_CYC_END_LS_DOWN[i] = 1
LV_END_DIAG_CHK_LS_DOWN[i] = 0
LV_ERR_CHG_LS_DOWN = 0
ERR_SYM_CHG_LS_DOWN = 0
LV_CDN_DIAG_CHG_LS_DOWN = 0
LV_END_DIAG_CHG_LS_DOWN = 0
T_END_DIAG_ACT_LS_DOWN = 0
LV_SYM_AFL_CHK_LS_DOWN[i] = 0
LV_SYM_AFR_CHK_LS_DOWN[i] = 0
T_AFR_CHK_LS_DOWN[i] = 0

```
T_AFL_CHK_LS_DOWN[i] = 0
LV_DIAG_EOL_DC_LS_UP_DOWN[i] = 0
```

Recurrence:

This function shall be carried out every 100 ms.

Activation:

```
If    LV_END_DIAG_LAM_ADJ[i] = 1
or    LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
or    LV_DIAG_EOL_DC_LS_UP_DOWN[i] = 1
or    LV_DIAG_OPL_REQ_CBK[i] = 1 for any i
then
      LV_CDN_DIAG_CHK_LS_DOWN[i] = 1

      Activate the function
      If      LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 1
      then    LV_DIAG_EOL_DC_LS_UP_DOWN[i] =
                          LV_DIAG_EOL_REQ_LS_UP_DOWN[i]
      endif
Else
      LV_CDN_DIAG_CHK_LS_DOWN[i] = 0
      Deactivate the function
Endif
```

Formula section:

```
If(1a) LV_INH_DIAG_CHG_LS_DOWN = 0
then(1a)
  If(2a)    LV_DIAG_ACT_END_LS_UP_DOWN[i] = 0 for any i
  % at least one bank still not complete: Symptom acquisition
  then(2a)
      If(3a)      LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 1
      then(3a)
          If(4a)             VLS_DOWN[i] <
          C_VLS_AFR_THD_DIAG_ACT_LS_DOWN[i]
          then(4a)
              If(5a)         T_AFR_CHK_LS_DOWN[i] < C_T_MAX_CHK_LS_DOWN
              then(5a)
                  Increment T_AFR_CHK_LS_DOWN[i]
              else(5b)
                  LV_SYM_AFL_CHK_LS_DOWN[i] = 1
              endif(5a/b)
          endif(4a)
```

```
    elseif(3b)LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 1
    then(3b)

        If(4b)                VLS_DOWN[i] >
        C_VLS_AFL_THD_DIAG_ACT_LS_DOWN[i]
        then(4b)

            If(5c)          T_AFL_CHK_LS_DOWN[i] <
            C_T_MAX_CHK_LS_DOWN[i]
            then(5c)

                Increment T_AFL_CHK_LS_DOWN[i]

            else(5d)

                LV_SYM_AFR_CHK_LS_DOWN[i] = 1

            endif(5c/d)

        endif(4b)

    endif(3b)

else(2b)

% check on all banks complete: Symptom evaluation and error flag generation

    # If(3c) (NC_CBK_EX_NR == 2)
    # then(3c) % 2 banks configuration: check for rich/lean shifts and interchanged symptom

        LV_CDN_DIAG_CHG_LS_DOWN = 1

        If(4c)  ( LV_SYM_AFL_CHK_LS_DOWN[1] = 1
        and LV_SYM_AFR_CHK_LS_DOWN[2] = 0
        and LV_SYM_AFL_CHK_LS_DOWN[2] = 0 )
        then(4c)

            ERR_SYM_CHK_LS_DOWN[1] = “SYM_1”                %signal too low
            bank 1

            LV_ERR_CHK_LS_DOWN[1] = 1

        elseif(4d) ( LV_SYM_AFL_CHK_LS_DOWN[2] = 1
        and LV_SYM_AFR_CHK_LS_DOWN[1] = 0
        and LV_SYM_AFL_CHK_LS_DOWN[1] = 0 )
         then(4d)

            ERR_SYM_CHK_LS_DOWN[2] = “SYM_1”                %signal too low
            bank 2

            LV_ERR_CHK_LS_DOWN[2] = 1

        elseif(4e) ( LV_SYM_AFR_CHK_LS_DOWN[1] = 1
        and LV_SYM_AFR_CHK_LS_DOWN[2] = 0
        and LV_SYM_AFL_CHK_LS_DOWN[2] = 0 )
         then(4e)

            ERR_SYM_CHK_LS_DOWN[1] = “SYM_0”                %signal too high
            bank 1

            LV_ERR_CHK_LS_DOWN[1] = 1

        elseif(4f) ( LV_SYM_AFR_CHK_LS_DOWN[2] = 1
        and LV_SYM_AFR_CHK_LS_DOWN[1] = 0
```

```
                and LV_SYM_AFL_CHK_LS_DOWN[1] = 0 )
                then(4f)
                    ERR_SYM_CHK_LS_DOWN[2] = "SYM_0"          %signal too high bank 2
                    LV_ERR_CHK_LS_DOWN[2] = 1
                elseif(4g)  (( LV_SYM_AFR_CHK_LS_DOWN[2] = 1
                            or  LV_SYM_AFL_CHK_LS_DOWN[2] = 1 )
                    and     ( LV_SYM_AFR_CHK_LS_DOWN[1] = 1
                            or   LV_SYM_AFL_CHK_LS_DOWN[1] = 1 ))
                then(4g)
                    ERR_SYM_CHG_LS_DOWN = "SYM_3"
                                                    %downstream oxygen sensors interchanged
                    LV_ERR_CHG_LS_DOWN = 1
                else(4h)
                    ERR_SYM_CHK_LS_DOWN[i] = "NO_SYM" for [1] and [2]
                    ERR_SYM_CHG_LS_DOWN = "NO_SYM"
                endif(4c-h)
            # elseif(3d) (NC_CBK_EX_NR == 1)
            # then(3d) % 1 bank configuration: check for rich/lean shifts only
                If(4i) (LV_SYM_AFL_CHK_LS_DOWN[1] = 1)
                then(4i)
                    ERR_SYM_CHK_LS_DOWN[1] = "SYM_1"          %signal too low
                    LV_ERR_CHK_LS_DOWN[1] = 1
                elseif(4j) (LV_SYM_AFR_CHK_LS_DOWN[1] = 1)
                then(4j)
                    ERR_SYM_CHK_LS_DOWN[1] = "SYM_0"          %signal too high
                    LV_ERR_CHK_LS_DOWN[1] = 1
                endif(4i/j)
            # endif(3c/d)
        endif(2a/b)
    else(1b) % LV_INH_DIAG_CHG_LS_DOWN = 1 – downstream interchanged sensors not possible
        If(2c)     LV_DIAG_AFR_CDN_LS_UP_DOWN[i] = 1
        then(2c)
            If(3e)            VLS_DOWN[i] < C_VLS_AFR_THD_DIAG_ACT_LS_DOWN[i]
            then(3e)
                If(4k)            T_AFR_CHK_LS_DOWN[i] < C_T_MAX_CHK_LS_DOWN
                then(4k)
                    Increment T_AFR_CHK_LS_DOWN[i]
                else(4l)
                    LV_SYM_AFL_CHK_LS_DOWN[i] = 1
```

```
        ERR_SYM_CHK_LS_DOWN[i] = "SYM_1"                %signal too low
        LV_ERR_CHK_LS_DOWN[i] = 1
      endif(4k/l)
    endif(3e)
  elseif(2d)    LV_DIAG_AFL_CDN_LS_UP_DOWN[i] = 1
  then(2d)
    If(3f)          VLS_DOWN[i] > C_VLS_AFL_THD_DIAG_ACT_LS_DOWN[i]
    then(3f)
      If(4m)            T_AFL_CHK_LS_DOWN[i] < C_T_MAX_CHK_LS_DOWN
      then(4m)
        Increment T_AFL_CHK_LS_DOWN[i]
      else(4m)
        LV_SYM_AFR_CHK_LS_DOWN[i] = 1
        ERR_SYM_CHK_LS_DOWN[i] = "SYM_0"                %signal too high
        LV_ERR_CHK_LS_DOWN[i] = 1
      endif(4m/n)
    endif(3f)
  endif(2d)
end(1a/b)
```

End flag generation

```
If          T_END_DIAG_ACT_LS_DOWN[i] >= C_T_END_DIAG_ACT_LS_DOWN[i] for all i
or          LV_DIAG_ACT_END_LS_UP_DOWN[i] = 1 for all i
or          LV_ERR_CHK_LS_DOWN[i] = 1 for any i
or          LV_ERR_CHG_LS_DOWN = 1
then        LV_END_DIAG_CHG_LS_DOWN = 1
else        %active test was not triggered – no failure case
     If     LV_DIAG_OPL_REQ_CBK[i] = 0 for all i
     and    LV_END_DIAG_LAM_ADJ[i] = 1
     and    LV_DIAG_ACT_INH_LS_UP_DOWN[i] = 0
     and    LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 0

     then       increment T_END_DIAG_ACT_LS_DOWN[i]
     else       freeze T_END_DIAG_ACT_LS_DOWN[i]
     endif
endif

if          LV_END_DIAG_CHG_LS_DOWN = 1
or          (T_END_DIAG_ACT_LS_DOWN[i] >= C_T_END_DIAG_ACT_LS_DOWN[i]
            and LV_INH_DIAG_CHG_LS_DOWN = 1)
then        LV_END_DIAG_CHK_LS_DOWN[i] = 1
endif
```

Hint: The constant C_T_END_DIAG_ACT_LS_DOWN[i] must be bigger equal
2* (C_T_MAX_DIAG_AFL_LS_UP_DOWN + C_T_MAX_DIAG_AFR_LS_UP_DOWN + maximum time for the MAF integral C_MAF_INT_DIAG_LS_UP_DOWN[i] times 2).
The reason is to be in line with Rate Based Monitoring requirements (duration of diagnosis in the no failure case not shorter than in the failure case).

16.60.4 Moving Lambda Set Point

FUNCTION DESCRIPTION:

This functions shall move the lambda set point to a calibratable value, in order to do a plausibility test with the upstream and the downstream oxygen sensors separately. To avoid a jump in the lambda value and consequently any problem concernig driving confort, the shift in the lambda set point should be done through a ramp function with calibratable slope. The slope is defined as the diference between desired lambda set point and its value before the active test divided by the number of recurrences to achieve the final set point value, which is calibratable.

While the function is moving the lambda set point the flag LV_LAMB_SP_REQ_DIAG_ACT[i] is active (set in module 708A) to import this new value with high priority to the lambda coordination. To avoid any unbounded value of LAMB_SP[i], it’s monitored while the movemnt occurs. As bounds for the lambda set point, it is used the desired AFR and AFL calibration data.

Description:

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset or at clearing error memory initialize all variables as follows:

LAMB_SP_DELTA_LS_UP_DOWN[i] = 0
LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] = 0

Recurrence:

This function shall be carried out every 100 ms.

Activation:

This function shall be activated when
LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 1 **or**
LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 1

Otherwise this function shall be deactivated and *LAMB_SP_DIAG_OPL_LS_UP_DOWN[i]* shall be set to zero.

Formula section:

Shift lambda set point to rich

/One-off update by the activation of the rich test

If LV_LAMB_SP_AFR_REQ_DIAG_ACT$_{n-1}$[i] = 0

And LV_LAMB_SP_AFR_REQ_DIAG_ACT$_{n}$[i] = 1

```
Then   //Calc. only for the first time
    LAMB_SP_DELTA_LS_UP_DOWN[i] =
            (C_LAMB_SP_AFR_DIAG_LS_UP_DOWN  -
            LAMB_SP_BEG_DIAG_LS_UP_DOWN[i]) /
            C_LAMB_SP_STEP_DIAG_LS_UP_DOWN
    If LAMB_SP_DELTA_LS_UP_DOWN[i] = 0
    then
        LAMB_SP_DELTA_LS_UP_DOWN[i] =  - 1H
    endif
    Update LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] =
            LAMB_SP_BEG_DIAG_LS_UP_DOWN[i]
Endif


/moving lambda set point
If   LV_LAMB_SP_AFR_REQ_DIAG_ACT[i] = 1
then
    If        LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] >
                            C_LAMB_SP_AFR_DIAG_LS_UP_DOWN
    then
            increment LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] with
                    LAMB_SP_DELTA_LS_UP_DOWN [i]
    else
            LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] =
                    C_LAMB_SP_AFR_DIAG_LS_UP_DOWN
    endif
endif
```

<u>Shift lambda set point to lean</u>

```
/One-off update by the activation of the lean test
If   LV_LAMB_SP_AFL_REQ_DIAG_ACT$_{n-1}$[i] = 0
And LV_LAMB_SP_AFL_REQ_DIAG_ACT$_{n}$[i] = 1
Then //Calc. only for the first time
    LAMB_SP_DELTA_LS_UP_DOWN[i] =
            (C_LAMB_SP_AFL_DIAG_LS_UP_DOWN  -
            LAMB_SP_BEG_DIAG_LS_UP_DOWN[i]) /
            C_LAMB_SP_STEP_DIAG_LS_UP_DOWN
    If LAMB_SP_DELTA_LS_UP_DOWN[i] = 0
    then
            LAMB_SP_DELTA_LS_UP_DOWN[i] =  +1H
```

```
        endif
        Update LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] =
                LAMB_SP_BEG_DIAG_LS_UP_DOWN[i]
    endif
    /moving lambda set point
    If  LV_LAMB_SP_AFL_REQ_DIAG_ACT[i] = 1
    then
        If      LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] <
                                C_LAMB_SP_AFL_DIAG_LS_UP_DOWN
        then
                increment LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] with
                        LAMB_SP_DELTA_LS_UP_DOWN[i]
        else
                LAMB_SP_DIAG_OPL_LS_UP_DOWN[i] =
                        C_LAMB_SP_AFL_DIAG_LS_UP_DOWN
        endif
    endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MAF_INT_DIAG_LS_UP_DOWN[NC_CBK_EX_NR]	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Minimum threshold for MAF integral required to start active test					
C_T_MAX_DIAG_AFR_LS_UP_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Timer threshold for plausibility test under rich AF conditions					
C_T_MAX_DIAG_AFL_LS_UP_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Timer threshold for plausibility test under lean AF conditions					
C_T_DIAG_EOL_LS_UP_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Initialization of timer in case EOL to reduce the required minimum activity test time					
C_MAF_INT_EOL_DIAG_LS_UP_DOWN[NC_CBK_EX_NR]	1	0...FFFFH	0...1820.41666	0.0277778	[g]
Initialization of MAF integral to reduce required minimum MAF value to start active test					
C_LAMB_AFL_MIN_DIAG_LS_UP_DOWN	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Min. lambda set point for lean fault detection					
C_LAMB_AFL_MAX_DIAG_LS_UP_DOWN	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Max. lambda set point for lean fault detection					
C_LAMB_AFR_MIN_DIAG_LS_UP_DOWN	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Min. lambda set point for rich fault detection					
C_LAMB_AFR_MAX_DIAG_LS_UP_DOWN	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Max. lambda set point for rich fault detection					
LC_DIAG_AUTH_CHK_LS_UP_DOWN	1	0...1H	0...1	1	[-]
Boolean switch allowing the active test					
C_LAMB_AFR_THD_DIAG_ACT_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda value threshold for rich fault detection (upstream sensor)					
C_LAMB_AFL_THD_DIAG_ACT_LSL_UP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda value threshold for lean fault detection (upstream sensor)					
C_ABC_INC_CHK_LSL_UP	1	0...FFH	0...255	1	[-]
Increment for debounce algorithmus					
C_ABC_MAX_CHK_LSL_UP	1	0...FFFFH	0...65535	1	[-]
Maximum value for debounce algorithmus					
C_CTR_SUM_RST_AFR_CHK_LSL_UP	1	0...FFFFH	0...65535	1	[-]
Reset debounce counter for Richt test					
C_CTR_SUM_RST_AFL_CHK_LSL_UP	1	0...FFFFH	0...65535	1	[-]
Reset debounce counter for Lean test					
C_VLS_AFR_THD_DIAG_ACT_LS_DOWN[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511	4.8828e-3	[V]
VLS voltage threshold for rich fault detection (downstream sensor)					
C_VLS_AFL_THD_DIAG_ACT_LS_DOWN[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511	4.8828e-3	[V]
VLS voltage threshold for lean fault detection (downstream sensor)					
C_T_MAX_CHK_LS_DOWN	1	0...FFFFH	0...6553.5	0.1	[s]
Time to debounce the error					
C_T_END_DIAG_ACT_LS_DOWN[NC_CBK_EX_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
Timer threshold for the end of the downstream sensor active test in the no failure case					
C_LAMB_SP_AFR_DIAG_LS_UP_DOWN	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda set point for rich fault detection					
C_LAMB_SP_AFL_DIAG_LS_UP_DOWN	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda set point for lean fault detection					
C_LAMB_SP_STEP_DIAG_LS_UP_DOWN	1	1...FFH	1...255	1	[-]
Number of recurrences to achieve the desired lambda set point					

16.61 O2 sensor open loop diag (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_ACT_INH_LS_UP_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
General inhibition Oxygen Sensor Active Plausibility Check					
LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibition of open loop diag while interferring functions are deactivated in failure case					
LV_DIAG_OPL_REQ_CBK[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Request open loop diagnosis					
STATE_RBM_CHK_LS_DOWN[NC_CBK_EX_NR]	V/O	0...7H	0...7	1	[-]
RBM - Interface of CHK_LS_DOWN monitor					
LV_DIAG_EOL_REQ_LS_UP_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Start EOL active test					
LV_INH_DIAG_CHG_LS_DOWN	V/O	0...1H	0...1	1	[-]
Inhibit diagnosis interchanged downstream sensors					
LV_DIAG_ACT_INH_RBM_LS_UP_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Inhibit- flag for RBM for Oxygen Sensor Active Plausibility Check					

Input data:

CPPWM	LV_ERR_TCO	LV_DIAG_ACT_END_LS_UP_DOWN[NC_CBK_EX_NR]	LV_VB_CDN_OBD_2
LV_ERR_TPS	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_AMP	LV_FL
LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_EL_CPS	LV_IGK	LV_ERR_LOAD_TPS_PLAUS
LV_ERR_CRK	LV_IND_FCUT	LV_ERR_MAF	LV_ERR_DIAGCPS
LV_LAMB_COP[NC_CBK_EX_NR]	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	LV_ERR_MAP_DIP_PLAUS	LV_ERR_CRK
LV_ERR_AMP_PLAUS	LV_ERR_SA_SAV	LV_MIS_STATE_B	LV_ERR_MAP_DIP_SENS
LV_ERR_SA_SYS	LV_PUC		LV_ERR_SA_SAV_LSL
LV_SAP	LV_ERR_MAP_DIP_SHIFT	LV_ERR_SA_SAP	LV_SAV
LV_ERR_SAV	LV_SAWUP	LV_ERR_IVVT	LV_ERR_CAM_CUS
LV_ERR_SAP	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_DIAG_ACT_END_LS_UP_DOWN[NC_CBK_EX_NR]	LV_DC
CTR_ERR_DYN_NR	LV_END_DIAG_CHK_LS_DOWN[NC_CBK_EX_NR]	NR_CBK_EX	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]
LV_ERR_MAP_TPS_PLAUS	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_FUP[NC_CBK_HPP_NR]	LV_ERR_FUP_MFP_PLAUS
LV_ERR_H_PRS_SYS	LV_ERR_VCV	LV_ERR_SLV_IVVT_IN	LV_ERR_FUP_ORNG
LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_DIAG_CHG_LS_DOWN_INH	1	0...1H	0...1	1	[-]
Manual inhibitation of symptom evaluation for downstream sensor interchanged error					

FUNCTION DESCRIPTION:

General information:

In this Appl. Incidence the execution order of the AF lean and rich active tests are defined. Also some criteria are defined, which must inhibit each test.
i = 1 ... NC_CBK_EX_NR

Application conditions:

Initialisation: **If** LV_IGK = 0→1 **or** reset **or** clearing the error memory **then**

LV_DIAG_ACT_INH_LS_UP_DOWN[i] = 0
LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[i] = 0
LV_DIAG_OPL_REQ_CBK[i] = 0
LV_INH_DIAG_CHG_LS_DOWN = 0
LV_DIAG_ACT_INH_RBM_LS_UP_DOWN[i] = 0
LV_DIAG_EOL_REQ_LS_UP_DOWN[i] = 0

***If** LV_IGK = 1→0*
LV_INH_DIAG_CHG_LS_DOWN = 1
LV_DIAG_ACT_INH_LS_UP_DOWN[i] = 1

Recurrence: 100 ms

Activation: LV_IGK = 1

Deactivation: LV_DIAG_ACT_END_LS_UP_DOWN[i] = 1 or LV_IGK = 0

Formula section:

If the following conditions are true, the active test shall wait until the interferring functions are deactivated in the failure case.

```
If      CPPWM <> 0%        canister purge valve is open

then    LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[i]  =  1
else    LV_DIAG_AFR_AFL_INH_LS_UP_DOWN[i]  =  0
endif
```

If the following errors/special mode are true the whole function shall be inhibited.

If LV_ERR_MAP_DIP_SENS = 1 **OR**
- LV_ERR_MAP_DIP_PLAUS = 1 **OR**
- LV_ERR_MAP_DIP_SHIFT = 1 **OR**

```
    -  LV_ERR_IVVT = 1                                                          OR
    -  LV_ERR_FSD_LAM_LIM[i] = 1                                                OR
    -  LV_ERR_FUP = 1                                                           OR
    -  LV_ERR_FUP_MFP_PLAUS = 1                                                 OR
    -  LV_ERR_H_PRS_SYS = 1                                                     OR
    -  LV_ERR_VCV = 1                                                           OR
    -  LV_ERR_FUP_ORNG = 1                                                      OR
    -  LV_ERR_FUP_ST = 1                                                        OR
    -  LV_ERR_TTIP_MES_LSH_UP[i] = 1                                            OR
    -  LV_ERR_CRK = 1                  crankshaft sensor                        OR
    -  LV_ERR_CAM_CUS = 1                                                       OR
    -  LV_ERR_MAF = 1                  mass air flow sensor                     OR
    -  LV_ERR_LOAD_TPS_PLAUS = 1       (MAF, ISA, AMP)  PlausiCheck             OR
    -  LV_ERR_AMP = 1                  ambient pressure                         OR
    -  LV_ERR_AMP_PLAUS = 1                                                     OR
    -  LV_ERR_DYN_VLD_LS_UP[i] = 1     WRAF sensor dynamic diag.                OR
    -  LV_ERR_FSD[i] = 1               fuel system diagnosis                    OR
    -  LV_ERR_SCP_LS_DOWN[i] = 1       short circuit plus                       OR
    -  LV_ERR_SCG_LS_DOWN[i] = 1       short circuit ground                     OR
    -  LV_ERR_OC_LS_DOWN[i] = 1        open circuit                             OR
    -  LV_ERR_SWT_LS_DOWN[i] = 1       switch time rich-to-lean                 OR
    -  LV_ERR_PUE_LS_DOWN[i] = 1       sensor signal after PUC phase            OR
    -  LV_ERR_FL_LS_DOWN[i] = 1        full load error                          OR
    -  LV_ERR_PUC_LS_DOWN[i] = 1       failure during PUC phase                 OR
    -  LV_ERR_LSH_DOWN[i] = 1          heater electrical failure                OR
    -  LV_ERR_OBD_LSH_DOWN[i] = 1      heater OBD2 failure                      OR
    -  LV_ERR_EL_CPS = 1               Canister purge valve                     OR
    -  LV_ERR_DIAGCPS = 1              Can. purge valve mechanical blocked      OR
    -  LV_ERR_TPS = 1                  Throttle position sensor                 OR
    -  LV_ERR_TCO = 1                  Coolant temperature sensor               OR
    -  LV_MIS_STATE_B = 1              Misfire State B detected                 OR
    -  LV_ERR_SA_SAP = 1                                                        OR
    -  LV_ERR_SA_SAV_LSL = 1                                                    OR
    -  LV_ERR_SAP = 1                                                           OR
    -  LV_ERR_SAV = 1                  Secondary air valve                      OR
    -  LV_ERR_SA_SAV = 1               Secondary air valve error detected       OR
    -  LV_ERR_SA_SYS = 1               Secondary air pump error detected        OR
    -  LV_ERR_SLV_IVVT_IN = 1                                                   OR
    -  LV_ERR_MAP_TPS_PLAUS = 1                                                 OR
    -  LV_ERR_TTIP_MES_LSH_UP[i] = 1   ceramic temperatur too low

     then   LV_DIAG_ACT_INH_RBM_LS_UP_DOWN[i]  =  1
     else   LV_DIAG_ACT_INH_RBM_LS_UP_DOWN[i]  =  0
    endif

 if         LV_DIAG_ACT_INH_RBM_LS_UP_DOWN[i]  =  1                             OR
            LV_LS_UP_OBD_2_MAN_DEAC[i] = 1                                      OR
            LV_VB_CDN_OBD_2 = 0     battery voltage is too low                  OR
```

```
            LV_PUC = 1            pull fuel cut off                        OR
            LV_FL = 1             full load                                OR
            LV_SAV = 1            secondary air valve active               OR
            LV_SAP = 1            secondary air pump active                OR
            LV_SAWUP = 1          secondary air is active                  OR
            LV_IND_FCUT = 1       cylinder cut off is active               OR
            LV_LAMB_COP_i = 1     catalyst overheating protection active

then        LV_DIAG_ACT_INH_LS_UP_DOWN[i]  =  1
else        LV_DIAG_ACT_INH_LS_UP_DOWN[i]  =  0
endif
```

% Inhibitation of symptom evaluation for downstream sensor interchanged error

```
IF          LC_DIAG_CHG_LS_DOWN_INH = 1         OR
            LV_LS_UP_OBD_2_MAN_DEAC[2] = 1
THEN        LV_INH_DIAG_CHG_LS_DOWN = 1
ELSE        LV_INH_DIAG_CHG_LS_DOWN = 0
ENDIF
```

% Start and control of open loop test

```
    If      LV_ERR_DELTA_I_LAM[i] = 1       OR
            LV_ERR_VLS_DOWN_DIF[i] = 1      for any i
    Then    LV_DIAG_OPL_REQ_CBK[1] = 1
    Endif

    # If    NC_CBK_EX_NR = 2
    # Then
            If      LV_DIAG_OPL_REQ_CBK[1] = 1              and
                    (LV_DIAG_ACT_END_LS_UP_DOWN[1] = 1 or NR_CBK_EX = 1)
            Then    LV_DIAG_OPL_REQ_CBK[2] = 1
            Endif
    # Endif

    If      LV_DIAG_ACT_END_LS_UP_DOWN[i] = 1 for all i
    Then    LV_DIAG_OPL_REQ_CBK[i] = 0 for all i
    Endif
      //reset of LV_DIAG_OPL_REQ_CBK[i] after open loop test to re- enable e.g. stratified combustion
```

Data :

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
Downstream sensors	See Table of failure		*see table of failires*	*see table of failires*	---		See Table

Bank1 and 2 interchanged							of failure
CHG_LS_DOWN		3					

16.61.1 Interface for Rate – Based – Monitoring

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the CHK_LS_DOWN[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_CHK_LS_DOWN[i] data.

Within STATE_RBM_CHK_LS_DOWN[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for active- check)

Application conditions:

Initialisation:

at LV_DC 0 → 1 transition:

bit 0 , bit 1 of STATE_RBM_CHK_LS_DOWN[i] = 0

bit 2 of STATE_RBM_CHK_LS_DOWN[i] = 1

on failure memory reset:

bit 1 of STATE_RBM_CHK_LS_DOWN[i] = 0

at ECU reset all bits of STATE_RBM_CHK_LS_DOWN[i] shall be set to zero.

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_MAP_DIP_SHIFT
LV_ERR_MEC_IVVT_IN[NC_NR_CBK_IVVT]	LV_ERR_MEC_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_SLV_IVVT_EX[NC_NR_CBK_IVVT]
LV_ERR_CRK_PLAUS	LV_ERR_CRK_TOOTH_PER	LV_ERR_CRK_SYN
LV_ERR_CRK_TOOTH	LV_ERR_PLAUS_CAM_IN_1	LV_ERR_PLAUS_CAM_EX_1
LV_ERR_PER_CAM_IN_1	LV_ERR_PER_CAM_EX_1	LV_ERR_REF_CRK_CAM_IN_1
LV_ERR_REF_CRK_CAM_EX_1	LV_ERR_SYN_CAM_EX_1	LV_ERR_SYN_CAM_IN_1
LV_ERR_MAF	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_AMP
LV_ERR_AMP_PLAUS	LV_ERR_EL_CPS	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_DIAGCPS	LV_ERR_TCO_EL	LV_ERR_FSD[NC_CBK_EX_NR]
LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS	LV_ERR_TCO_STUCK
LV_ERR_MIS[NC_CYL_NR]	LV_ERR_SAP	LV_ERR_SAV
LV_ERR_SA_SAP	LV_ERR_SA_SYS	LV_ERR_TPS_1
LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]

LV_ERR_TPS_2	LV_ERR_TPS_AD	LV_ERR_TPS_AD_BOL	LV_ERR_TPS_MAF_1
LV_ERR_TPS_MAF_2	LV_ERR_TPS_ST_CHK_2	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MTC_CTL_2
LV_ERR_MTC_CTL_3	LV_ERR_MTC_DR	LV_MIS_STATE_B	LV_ERR_FSD_LAM_LIM[i]
LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV
	LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TCO_STUCK_RNG
LV_ERR_TTIP_MES_LSH_UP[i]	LV_ERR_SYN_CRK_CAM_IN_1	LV_ERR_SYN_CRK_CAM_EX_1	LV_ERR_TOOTH_OFF_IN_1
LV_ERR_TOOTH_OFF_EX_1	LV_ERR_SLV_IVVT_IN	LV_ERR_OFS_LSL_UP[i]	LV_ERR_CTL_LSL_UP[i]
LV_ERR_MAP_TPS_PLAUS			

```
If(1)       { CPU optimization at LV_DC 0 → 1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While  bit 1 of STATE_RBM_CHK_LS_DOWN[i] = 0 do
                   with each XX failure of the above list  (depending on configuration, see below):
                   ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>, SYNCHRONIZATION<CALL>)

                   If(2)         XX has a pending status
                   Then(2)       bit 1 of STATE_RBM_CHK_LS_DOWN[i] = 1
                   Endif(2)
            Endwhile
Else(1)     { the dynamic failure memory is empty }      No action
Endif(1)
```

Every 1 s :

```
If    bit 0 of STATE_RBM_CHK_LS_DOWN[i] = 0
Then
          If      LV_END_DIAG_CHK_LS_DOWN[i] = 1
          Then    bit 0 of STATE_RBM_CHK_LS_DOWN[i] = 1
          Endif
Endif

If    bit 1 of STATE_RBM_CHK_LS_DOWN[i] = 0
Then
          If      LV_DIAG_ACT_INH_RBM_LS_UP_DOWN[i] = 1
          Then    bit 1 of STATE_RBM_CHK_LS_DOWN[i] = 1
          Endif
Endif
```

16.62 O2 sensor (lin, up) characteristic line diag

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_DLY_DIAG_SHIFT_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Delay timer for the end flag of the shift down diagnosis in the no failure case (RBM purpose)					
LV_T_DLY_DIAG_SHIFT_DOWN[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
End flag of the shift down diagnosis in the no failure case (RBM purpose)					
LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
WRAF Sensor characteristic line shift to lean ERROR detected					
ERR_SYM_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
WRAF Sensor characteristic line shift to lean SYMPTOM detected					
LV_END_DIAG_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Diagnostic available for the WRAF Sensor characteristic line shift to lean					
LV_CDN_DIAG_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Condition to run the Diagnosis WRAF Sensor characteristic line shift to lean					
LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
WRAF Sensor characteristic line shift to rich ERROR detected					
ERR_SYM_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
WRAF Sensor characteristic line shift to rich SYMPTOM detected					
LV_END_DIAG_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Diagnostic available for the WRAF Sensor characteristic line shift to rich					
LV_CDN_DIAG_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Condition to run the Diagnosis WRAF Sensor characteristic line shift to rich					

Input data:

LV_ST_END	NC_CBK_EX_NR	LV_IGK	
LV_END_DIAG_LAM_ADJ[NC_CBK_EX_NR]	LV_ERR_LAM_ADJ[NC_CBK_EX_NR]		
LV_DIAG_AFR_CHK_SYM_LSL_UP[NC_CBK_EX_NR]	LV_DIAG_AFL_CHK_SYM_LSL_UP[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MAX[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MIN[NC_CBK_EX_NR]
STATE_SYM_DIAG_PLAUS_LSL_UP[NC_CBK_EX_NR]	LV_DIAG_PLAUS_SYM_LSL_UP[NC_CBK_EX_NR]	LV_DIAG_PLAUS_END_LSL_UP[NC_CBK_EX_NR]	LV_DIAG_EOL_END_LS_UP_DOWN[NC_CBK_EX_NR]
STATE_DIAG_ACT_LS_UP_DOWN[NC_CBK_EX_NR]			NC_IDX_DIAG_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]
NC_IDX_DIAG_SHIFT_AFR_LSL_UP			

Import actions:

ACTION_ERRM_NoFilterSymptom(IN<IDX_DIAG>, IN< LV_CDN_DIAG_XX >, IN< ERR_SYM_XX >, IN< LV_ERR_SET_XX >, IN< LV_ERR_RST_XX >, IN< LV_END_DIAG_XX >, OUT< LV_ERR_XX >)
ACTION_ERRM_NoFilterReset(IN<IDX_DIAG>, OUT< LV_ERR_XX >)

16.62.1 Delay time calculation after finishing fuel trim diagnosis without failure

FUNCTION DESCRIPTION:

General information:

This function shall deliver information in the no failure case, indicating that the sensor characteristic shift down diagnosis could have finished in the failure case.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset **or** at clearing error memory:

T_DLY_DIAG_SHIFT_DOWN[i] = 0
LV_T_DLY_DIAG_SHIFT_DOWN[i] = 0

Recurrence:

The function should be carried out once every 100 ms.

Activation / Deactivation:

The function will be carried out in case of

LV_ST_END = 1

Formula section:

```
If      T_DLY_DIAG_SHIFT_DOWN[i] >= C_T_DLY_DIAG_SHIFT_DOWN[i]
then    LV_T_DLY_DIAG_SHIFT_DOWN[i] = 1
else    If      LV_END_DIAG_LAM_ADJ[i] = 1
        and     LV_ERR_LAM_ADJ[i] = 0
        then    increment T_DLY_DIAG_SHIFT_DOWN[i]
        endif
endif
```

16.62.2 Sensor Characteristic Line Shift to Lean (CSD)

FUNCTION DESCRIPTION:

General information:

This function shall deliver information indicating that the sensor characteristic line has a shift to lean **(CSD)**, which shall be done by summarizing all similar failure symptoms of this kind.

It is assumed such deviation from the nominal behavior can be occur due to the poisoning of sensor reference air channel with fuel, but also due to a leak current inside the sensor, a very low reference air pump current (used in some types of wide range oxygen sensors) or a relatively high impedance short circuit to the battery voltage or ground in one of the four sensor lines.

In dependence of the shift strengh that are three different paths followed by this function:

1. Strong shift to lean: lambda controller go to its limit and the plausibility check recognises this error.
2. Middle strong shift to lean: the trim controller goes to its limit but the lambda controller not, the trim controller deviation diagnosis recognises that the system has a problem and the active test detects that the problem is in the upstream sensor, this showing a characteristic line shift to lean.
3. Mild shift to lean: the trim controller goes to its limit but the lambda controller not. The trim controller I-share runs to its limit, whereas the trim ctr. deviation diagnosis are not able to recognise that the system has a problem. The trim ctr I-share diagnosis triggers the active test which detects that the problem is in the upstream sensor, this showing a characteristic line shift to lean.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset **or** at clearing error memory:

```
ACTION_ERRM_NoFilterReset( IN<NC_IDX_DIAG_SHIFT_AFL_LSL_UP[i]>,
OUT<LV_ERR_SHIFT_AFL_LSL_UP[i]>)
```

Recurrence:

The function should be carried out once every 100 ms.

Activation / Deactivation:

The diagnosis will be carried out

```
If  LV_ST_END = 1 and LV_ERR_SHIFT_AFL_LSL_UP[i] = 0
Then
    LV_CDN_DIAG_SHIFT_AFL_LSL_UP[i] = 1        / activation
Else
    LV_CDN_DIAG_SHIFT_AFL_LSL_UP[i] = 0        / deactivation
Endif
```

Formula section:

```
IF   LV_DIAG_AFR_CHK_SYM_LSL_UP[i] = 1
     % failure detected during active test
     OR
     LV_DIAG_PLAUS_SYM_LSL_UP[i] = 1 &
     STATE_SYM_DIAG_PLAUS_LSL_UP[i] = “SYM_1 ” &
     LV_FAC_LAM_LIM_MAX[i] = 1
     % failure detected trhrough plausibility test and lambda ctr. in limitation
THEN
     LV_ERR_SHIFT_AFL_LSL_UP[i] = 1
     ERR_SYM_SHIFT_AFL_LSL_UP[i] = “SYM_0”
     LV_END_DIAG_SHIFT_AFL_LSL_UP[i] = 1
ELSE
     % normal driving cycle, no error detected
     IF     ( LV_T_DLY_DIAG_SHIFT_DOWN[i] = 1
            OR     STATE_DIAG_ACT_LS_UP_DOWN[i] = "check finished" )
     AND    LV_DIAG_PLAUS_END_LSL_UP[i] = 1
     THEN   LV_END_DIAG_SHIFT_AFL_LSL_UP[i] = 1
     ELSE
            % EOL Test, no error detected
            IF LV_DIAG_EOL_END_LS_UP_DOWN[i] = 1
            THEN
                   LV_END_DIAG_SHIFT_AFL_LSL_UP[i] = 1
            ENDIF
     ENDIF
ENDIF
```

For failure and error management treatment the anti-bounce mechanism is called :

% if LV_ERR_XX=1 then LV_ERR_SET=1 or if LV_ERR_XX=0 then LV_ERR_RST=1

ACTION_ERRM_NoFilterSymptom(IN<NC_IDX_DIAG_SHIFT_AFL_LSL_UP[i]>, IN<LV_CDN_DIAG_SHIFT_AFL_LSL_UP[i]>, IN< ERR_SYM_SHIFT_AFL_LSL_UP[i] >, IN< LV_ERR_SET_SHIFT_AFL_LSL_UP[i] >, IN<LV_ERR_RST_SHIFT_AFL_LSL_UP[i]>, IN<LV_END_DIAG_SHIFT_AFL_LSL_UP[i]>, OUT<LV_ERR_SHIFT_AFL_LSL_UP[i]>)

16.62.3 Sensor Characteristic Line Shift to Rich

FUNCTION DESCRIPTION:

General information:

This function shall deliver information indicating that the sensor characteristic line has a shift to rich, which shall be done by summarizing all similar failure symptoms of this kind.

It is assumed such deviation from the nominal behavior can be occur due to a leak current inside the sensor, as well as a relatively high impedance short circuit to the battery voltage or ground in one of the four sensor lines.

In dependence of the shift strengh there are three different paths followed by this function:

4. Strong shift to rich: lambda controller go to its limit and the plausibility check recognises this error.
5. Middle strong shift to rich: the trim controller goes to its limit but the lambda controller not: The trim controller deviation diagnosis recognises that the system has a problem and the active test detects that the problem is in the upstream sensor showing a characteristic line shift to rich.
6. Mild shift to rich: the trim controller goes to its limit but the lambda controller not. The trim controller I-share runs to its limit, whereas the trim ctr. deviation diagnosis are not able to recognise that the system has a problem. The trim ctr I-share diagnosis triggers the active test which detects that the problem is in the upstream sensor, this showing a characteristic line shift to rich.

Application conditions:

Initialisation:

At LV_IGK = 0 to 1, reset **or** at clearing error memory:

ACTION_ERRM_NoFilterReset(IN< NC_IDX_DIAG_SHIFT_AFR_LSL_UP[i]>, OUT<LV_ERR_SHIFT_AFR_LSL_UP[i]>)

Recurrence:

The function should be carried out once every 100 ms.

Activation / Deactivation:

The diagnosis will be carried out

```
 If  LV_ST_END = 1 and LV_ERR_SHIFT_AFR_LSL_UP[i] = 0
Then
    LV_CDN_DIAG_SHIFT_AFR_LSL_UP[i] = 1         / activation
Else
    LV_CDN_DIAG_SHIFT_AFR_LSL_UP[i] = 0         / deactivation
Endif
```

Formula section:

```
IF    LV_DIAG_AFL_CHK_SYM_LSL_UP[i] = 1
    % failure detected during active test
    OR
      LV_DIAG_PLAUS_SYM_LSL_UP[i] = 1  &
      STATE_SYM_DIAG_PLAUS_LSL_UP[i] = "SYM_0" &
      LV_FAC_LAM_LIM_MIN[i] = 1
    % failure detected through plausibility test. No open circuit failure detected.
THEN
    LV_ERR_SHIFT_AFR_LSL_UP[i] = 1
    ERR_SYM_SHIFT_AFR_LSL_UP[i] = "SYM_0"
    LV_END_DIAG_SHIFT_AFR_LSL_UP[i] = 1
ELSE
```

```
        % normal driving cycle, no error detected

        IF      (LV_T_DLY_DIAG_SHIFT_DOWN[i] = 1
        OR      STATE_DIAG_ACT_LS_UP_DOWN[i] = "check finished" )
        AND     LV_DIAG_PLAUS_END_LSL_UP[i] = 1
        THEN    LV_END_DIAG_SHIFT_AFR_LSL_UP[i] = 1
        ELSE
                % EOL Test, no error detected
                IF LV_DIAG_EOL_END_LS_UP_DOWN[i] = 1
                THEN
                        LV_END_DIAG_SHIFT_AFR_LSL_UP[i] = 1
                ENDIF

        ENDIF
ENDIF
```

<u>For failure and error management treatment the anti-bounce mechanism is called :</u>

% if LV_ERR_XX=1 then LV_ERR_SET=1 or if LV_ERR_XX=0 then LV_ERR_RST=1

```
ACTION_ERRM_NoFilterSymptom( IN<NC_IDX_DIAG_SHIFT_AFR_LSL_UP[i]>,
IN<LV_CDN_DIAG_SHIFT_AFR_LSL_UP[i]>,  IN< ERR_SYM_SHIFT_AFR_LSL_UP[i] >,
IN< LV_ERR_SET_SHIFT_AFR_LSL_UP[i] >, IN<LV_ERR_RST_SHIFT_AFR_LSL_UP[i]>,
IN<LV_END_DIAG_SHIFT_AFR_LSL_UP[i]>, OUT<LV_ERR_SHIFT_AFR_LSL_UP[i]> )
```

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
SHIFT_AFL_LSL_UP[i]	Characteristic Line Shift to Lean	SYM_0	NO
SHIFT_AFR_LSL_UP[i]	Characteristic Line Shift to Rich	SYM_0	NO

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_DIAG_SHIFT_DOWN[NC_CBK_EX_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
Threshold for the "end flag" delay timer in the no failure case					

16.63 O2 sensor (lin, up) characteristic line diag (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Interface of SHIFT_AFL_LSL_UP[NC_CBK_EX_NR] monitor with the Rate-Based Monitoring statistics					
STATE_RBM_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Interface of SHIFT_AFR_LSL_UP[NC_CBK_EX_NR] monitor with the Rate-Based Monitoring statistics					

Input data:

LV_IGK	NC_CBK_EX_NR	NC_CYL_NR	
LV_END_DIAG_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]	LV_END_DIAG_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	CTR_ERR_DYN_NR	LV_INH_DIAG_RBM_PLAUS_LSL_UP[NC_CBK_EX_NR]

16.63.1 Upstream sensor lean/rich – shift diagnosis: Interface for rate – based – monitoring

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the SHIFT_AFL_LSL_UP[i] and SHIFT_AFR_LSL_UP[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_SHIFT_AFL_LSL_UP[i] and STATE_RBM_SHIFT_AFR_LSL_UP[i] data.

Within STATE_RBM_SHIFT_AFL/AFR[i], three different informations are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for lean/rich shift diagnosis)

Application conditions:

Initialisation: *at ECU reset:*

bit 0, bit 1 and bit 2 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 0
bit 0, bit 1 and bit 2 of STATE_RBM_SHIFT_AFR_LSL_UP[i] = 0

LV_DC = 0 --> 1 transition:

bit 0 and bit 1 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 0
bit 0 and bit 1 of STATE_RBM_SHIFT_AFR_LSL_UP[i] = 0

bit 2 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 1
bit 2 of STATE_RBM_SHIFT_AFR_LSL_UP[i] = 1

on failure memory reset:

bit 1 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 0
bit 1 of STATE_RBM_SHIFT_AFR_LSL_UP[i] = 0

Recurrence: 1 s

Activation: LV_DC = 1

Deactivation: LV_DC = 0

Formula section:

At LV_DC 0 --> 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_MEC_IVVT_IN	LV_ERR_MAF	LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_MEC_IVVT_EX	LV_ERR_SA_SAV	LV_ERR_LOAD_TPS_PLAUS
LV_ERR_SLV_IVVT_IN	LV_ERR_SA_SAV_LSL	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_EX	LV_ERR_SA_SAP	LV_ERR_FSD[NC_CBK_EX_NR]
LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_SA_SYS	LV_ERR_REF_CRK_CAM_IN[NC_CBK_EX_NR]
LV_ERR_CRK_PLAUS	LV_ERR_SAV	LV_ERR_REF_CRK_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_CRK_TOOTH_PER	LV_ERR_SAP	LV_ERR_SYN_CAM_IN[NC_NR_CAM_CBK]
LV_ERR_CRK_TOOTH	LV_ERR_ECT_EL	LV_ERR_SYN_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_TH	LV_ERR_TPS	LV_ERR_PLAUS_CAM_IN[NC_NR_CAM_CBK]
LV_ERR_CRK_SYN	LV_ERR_MIS[NC_CYL_NR]	LV_ERR_PLAUS_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_MAP_DIP_PLAUS	LV_ERR_AMP_PLAUS	LV_ERR_PER_CAM_IN[NC_NR_CAM_CBK]
LV_ERR_MAP_DIP_SHIFT	LV_ERR_AMP	LV_ERR_PER_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_MAP_DIP_SENS	LV_ERR_CHG_LS_UP	LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_EL_CPS	LV_ERR_DIAGCPS	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_TCO_EL	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]
LV_ERR_TCO_GRD	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSH_LSL_UP[NC_CBK_EX_NR]
LV_ERR_TCO_PLAUS	LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]
LV_ERR_TCO_STUCK	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]		LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR]		

```
If(1)      { CPU optimization at LV_IGK 0 --> 1 transition }
           CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
           While  bit 1 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 0 do
                  with each XX failure of the above list  (depending on configuration, see
                  below):
                  ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                  SYNCHRONIZATION<CALL>)

                  If(2)     XX has a pending status
                  Then(2)         bit 1 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 1
                  Endif(2)
           Endwhile
           bit 1 of STATE_RBM_SHIFT_AFR_LSL_UP[i] =
                                       bit 1 of STATE_RBM_SHIFT_AFL_LSL_UP[i]
```

```
    Else(1)   { the dynamic failure memory is empty }      No action
    Endif(1)

Every 1 s :
If    bit 0 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 0
Then
              If      LV_END_DIAG_SHIFT_AFL_LSL_UP[i] = 1
              Then    bit 0 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 1
              Endif
Endif

If    bit 0 of STATE_RBM_SHIFT_AFR_LSL_UP[i] = 0
Then
              If      LV_END_DIAG_SHIFT_AFR_LSL_UP[i] = 1
              Then    bit 0 of STATE_RBM_SHIFT_AFR_LSL_UP[i] = 1
              Endif
Endif
If    bit 1 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 0
Then
              If      LV_INH_DIAG_RBM_PLAUS_LSL_UP[i] = 1
              Then    bit 1 of STATE_RBM_SHIFT_AFL_LSL_UP[i] = 1
                      bit 1 of STATE_RBM_SHIFT_AFR_LSL_UP[i] = 1
              Endif
Endif
```

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
O2 Sensor characteristic line shift to rich	See Table of failure	0	See Table of failure	See Table of failure			See Table of failure
SHIFT_AFR_LSL_UP[i]							

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Hardware config	Recurence	Failure class A/B
O2 Sensor characteristic line shift to lean	See Table of failure	0	See Table of failure	See Table of failure			See Table of failure
SHIFT_AFL_LSL_UP[i]							

16.64 Oscillation Check Lambda sensor down

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_OSC_CHK[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Debounced failure of oscillation diagnosis for downstream sensor					
LV_END_DIAG_OSC_CHK[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
End of oscillation diagnosis for downstream sensor					
LAMB_SP_OSC_CHK[NC_CBK_EX_NR]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
lambda setpoint requested by oscillation check for downstream lambda sensor					

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Description:

This is a stub module to provide signals consumed in other EGCP modules.

Application conditions:

Initialisation: The output shall be set to the following values, which are never changed:
LV_ERR_OSC_CHK[i] = 0
LV_END_DIAG_OSC_CHK[i] = 1
LAMB_SP_OSC_CHK[i] = 0

Recurrence: *not applicable*

Activation: *not applicable*

Formula section:

- This function does not execute any actions or calculations. -

16.65 Diagnosis of monitor sensors

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAG_MPL_CDN_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Status of general activation conditions for diagnosis					
LV_CDN_DIAG_SWT_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Status of permission for diagnosis of switching time check					
VLS_DOWN_TRAN_PUC[NC_CBK_EX_NR]	O/V	0...3FFH	0...4.99511	4.8828e-3	[V]
Monitor sensor voltage on activating the trailing throttle fuel cut-off, Bank i					
CTR_SWT_ACT_LS_DOWN[NC_CBK_EX_NR]	O/V	0...FFH	0...5100	20	[ms]
Cycle counter for switching time determination, Bank i					
CTR_SWT_LS_DOWN[NC_CBK_EX_NR]	O/V	0...FFH	0...255	1	[-]
Number of valid switching times from rich to lean					
CTR_QUO_SUM_SWT_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...511.99219	0.0078125	[-]
Sum of weighted CTR_SWT_ACT_LS_DOWN[NC_CBK_EX_NR], Bank i					
CTR_CYCNR_SWT_LS_DOWN[NC_CBK_EX_NR]	O/V	0...FFH	0...1.99218	0.0078125	[-]
Average of weighted CTR_SWT_ACT_LS_DOWN[NC_CBK_EX_NR], Bank i					
CTR_SAVE_SWT_LS_DOWN[NC_CBK_EX_NR]	O/V/S	0...FFH	0...1.99218	0.0078125	[-]
Mean value of the ratio between the monitor sensor switching time and the threshold value for scantool					
CTR_QUO_SWT_LS_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...1.99218	0.0078125	[-]
weighted value of CTR_SWT_ACT_LS_DOWN[NC_CBK_EX_NR]					
MAF_KGH_MIN_PUC	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Minimum value of MAF_KGH while switching time calculation					
LV_END_DIAG_SWT_LS_DOWN[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
bit to indicate end of switching time diagnosis					
LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Error bit of the switching time diagnosis					
LV_SWT_DIAG_VLD_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Validity bit of the current switching time check					
ERR_SYM_SWT_LS_DOWN[NC_CBK_EX_NR]	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Determined error of the switching time diagnosis					
CTR_SWT_ACT_RBM_LS_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...5100	20	[ms]
Cycle counter for RBM switching time determination, Bank i					
CTR_QUO_SWT_RBM_LS_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...1.99218	0.0078125	[-]
weighted value of CTR_SWT_ACT_RBM_LS_DOWN[NC_CBK_EX_NR] for RBM purpose					
LV_CDN_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Status of permission for diagnosis of leaving PUC check					
VLS_DOWN_PUE[NC_CBK_EX_NR]	O/V	0...3FFH	0...4.99511	4.8828e-3	[V]
Monitor sensor voltage at the time of leaving PUC, Bank i (i=1.2)					
LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Debounced diagnosis result of pull end plausibility					

LV_DIAG_PUE_VLD_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical variable indicating validity of previous diagnostic cycle					
ERR_SYM_PUE_LS_DOWN[NC_CBK_EX_NR]	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom of pull end plausibility					
LV_END_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Bit to indicate end of pull end plausibility diagnosis					
CTR_SYM_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter for anti-bounce of PUE_LS_DOWN diagnosis					
CTR_END_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Counter to set LV_END_DIAG_VLS_DOWN_PUE_i					
VLS_DIF_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Difference of VLS_DOWN_i values from the beginning to the end of the MAF_INT_PUC_NOT_ACT calculation					
VLS_DIF_SAVE_DIAG_LS_DOWN[NC_CBK_EX_NR]	O/V/S	0...3FFH	0...4.99511	4.8828e-3	[V]
Difference of VLS_DOWN_i values from the beginning to the end of the MAF_INT_PUC_NOT_ACT calculation for scantool					
VLS_DIF_MIN_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR][1][1]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Minimum value of VLS_DIF_DIAG_PUE_LS_DOWN[i] until MAF threshold outside PUC is exceeded					
VLS_DIF_MAX_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR][1][1]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Maximum value of VLS_DIF_DIAG_PUE_LS_DOWN[i] until MAF threshold outside PUC is exceeded					
LV_MAF_INT_PUC_NOT_ACT_RST[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical variable indicating that MAF_INT_PUC_NOT_ACT integral has been reset					
CTR_QUO_SWT_LS_DOWN_MIN[NC_CBK_EX_NR]	V	0...FFH	0...1.99218	0.0078125	[-]
Minimum measured weighted value of CTR_SWT_ACT_LS_DOWN[NC_CBK_EX_NR]					
CTR_QUO_SWT_LS_DOWN_MAX[NC_CBK_EX_NR]	V	0...FFH	0...1.99218	0.0078125	[-]
Maximum measured weighted value of CTR_SWT_ACT_LS_DOWN[NC_CBK_EX_NR]					
LV_DIAG_SWT_LS_DOWN_S_ENA[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Conditions for SWT diagnosis after rich-lean combustion mode transition fulfilled					
MAF_INT_LAMB_AFR_SWT_S_ENA[NC_CBK_EX_NR]	V	0...FFFFH	0...2912.66666	0.0444444	[g]
Air mass with definite rich conditions for stratified SWT diagnosis (upstream lambda criterion)					
MAF_INT_VLS_AFR_SWT_S_ENA[NC_CBK_EX_NR]	V	0...FFFFH	0...2912.66666	0.0444444	[g]
Air mass with definite rich conditions for stratified SWT diagnosis (downstream lambda criterion)					
T_LAMB_NOT_AFR_SWT_S_ENA[NC_CBK_EX_NR]	V	0...FFFFH	0...1310.7	0.02	[s]
Time after leaving definite rich conditions for stratified SWT diagnosis (upstream lambda criterion)					
T_VLS_NOT_AFR_SWT_S_ENA[NC_CBK_EX_NR]	V	0...FFFFH	0...1310.7	0.02	[s]
Time after leaving definite rich conditions for stratified SWT diagnosis (downstream lambda criterion)					
MAF_INT_S_MAX_SWT_S[NC_CBK_EX_NR]	V	0...FFFFH	0...2912.66666	0.0444444	[g]
Air mass after entering lean combustion mode for stratified SWT diagnosis					

Input data:

LV_IGK	LV_PUC	NC_CBK_EX_NR	TCO

LV_LS_DOWN_READY[NC_CBK_EX_NR]	VS	TEMP_CAT_DYN_MDL[NC_CBK_EX_NR]	MAF_INT_PUC_ACT
MAF_KGH	VLS_DOWN[NC_CBK_EX_NR]	MAF_INT_PUC_NOT_ACT	LV_INH_DIAG_SWT_LS_DOWN[NC_CBK_EX_NR]
LV_INH_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	LV_DIAG_RGN_NT_LS_DOWN_ERR[NC_CBK_EX_NR]	LV_DIAG_RGN_NT_LS_DOWN_END[NC_CBK_EX_NR]	LV_S_ACT
LAMB_SP[NC_CBK_EX_NR]	LAMB_SP_FIL_S[NC_CBK_EX_NR]		

FUNCTION DESCRIPTION:

General information:

The monitor sensor diagnosis consists of three separate tests. These tests are carried out at the beginning or at the end of the pull fuel cut-off (PUC) or after switching from stoichiometric to lean combustion. In detail there are the tests monitoring of the rich-lean switching time, monitoring of the signal voltage transition during PUC or after switching to lean combustion and monitoring of the signal voltage motion after PUC, the last one being complemented by an alternative function providing the error flags LV_DIAG_RGN_NT_LS_DOWN_ERR[i] and the completition flags LV_DIAG_RGN_NT_LS_DOWN_END[i]. The purpose of these tests is the recognition of emission relevant sensor errors caused by changes in the sensor behaviour or by electrical errors like open circuits.

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
  i = 1, for exhaust cylinder bank 1
  i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
  i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation:

If At LV_IGK 0->1 **or** reset **or** at clearing error memory

then reset of all variables and bits to 0

endif

Recurrence: 20 ms

Activation / Deactivation:

If TCO > C_TCO_MIN_DIAG_MPL_LS_DOWN

and LV_LS_DOWN_READY[i] = 1

(Operating readiness of the monitor sensor must be detected)

and C_VS_MIN_DIAG_MPL_LS_DOWN ≤ VS ≤ C_VS_MAX_DIAG_MPL_LS_DOWN

(Vehicle speed in range)

```
and     TEMP_CAT_DYN_MDL[i] > C_TEMP_CAT_MIN_DIAG_MPL_LS_DOWN
        (Catalytic converter at operating temperature)
then
        LV_DIAG_MPL_CDN_LS_DOWN[i] = 1
Else
        LV_DIAG_MPL_CDN_LS_DOWN[i] = 0
endif

If      LV_DIAG_MPL_CDN_LS_DOWN[i] = 1
and     (LV_END_DIAG_PUE_LS_DOWN[i] = 0
or      LV_END_DIAG_SWT_LS_DOWN[i] = 0)
then
        LV_DIAG_MPL_CDN_LS_DOWN[i] = 1
else
        LV_DIAG_MPL_CDN_LS_DOWN[i] = 0
endif
```

16.65.1 Monitoring the rich-lean switching times in the trailing throttle fuel cut-off

FUNCTION DESCRIPTION:

General information:

Changes in the dynamic behaviour of the monitor sensor can be detected by the recognition of the rich-lean switching time after fuel cut-off or after changing the combustion mode from homogenous mode to stratified mode.

Description:

After detection of the PUC engine operating state, the rich-lean switching time of the lambda sensor is determined and weighted depending upon the mass air flow in the intake system. The functioning of the sensor is diagnosed by comparison of the mean value of the weighted switching time with a limit. The functioning is guaranteed in this test if the switching time is below the limit.
The switching time diagnosis shall be finished for the current driving cycle, if C_CTR_SWT_DIAG_LS_DOWN valid switching times are calculated or if the alternative function indicates its completition via LV_DIAG_RGN_NT_LS_DOWN_END[i] = 1. Whenever one of the two functions completes, its diagnostic result is used for the current driving cycle and the result of the other function is ignored.
The rich-lean switching time of the sensor is determined in defined ranges of the mass air flow. A prerequisite for performing the diagnosis is that the voltage of the monitor sensor at the time of activating the trailing throttle fuel cut-off VLS_DOWN_TRAN_PUC[i] is above the threshold C_VLS_PUC_MIN_DIAG_SWT_LS_DOWN. The voltage value of the sensor at the time of trailing throttle fuel cut-off VLS_DOWN_TRAN_PUC[i] is required for the calculation of the dynamic limits for switching point determination and must be stored for the diagnostic period.
If the sensor voltage drops below the value C_FAC_VLS_MAX_DIAG_SWT_LS_DOWN * VLS_DOWN_TRAN_PUC[i], the switching time determination starts and the cycle counter CTR_SWT_ACT_LS_DOWN[i] has to be started. The counting process is ended if the limit C_FAC_VLS_MIN_DIAG_SWT_LS_DOWN * VLS_DOWN_TRAN_PUC[i] is exceeded downwards. The value of the cycle counter CTR_SWT_ACT_LS_DOWN[i] is a measure of the rich-lean switching time of the monitor sensor.
If a rich peak which exceeds C_VLS_HYS_DIAG_SWT_LS_DOWN occurs during determination of the switching time, its determination is interrupted and the relevant switching time is no longer further processed. The determination of the sensor switching times must also be interrupted if PUC is left.
Simultaneously to incrementation of CTR_SWT_ACT_LS_DOWN[i] the counter CTR_SWT_ACT_RBM_LS_DOWN[i] shall be incremented until the corresponding ratio CTR_QUO_SWT_RBM_LS_DOWN[i] exceeds the threshold for a defective lambda probe C_CTR_SWT_LS_DOWN. After this condition has been fulfilled the valid cycle counter CTR_SWT_LS_DOWN[i] shall be incremented. This procedure ensures that within each diagnostic cycle and after completion of the required total number of diagnostic cycles a probe malfunction can be detected as is demanded by ARB.
A valid value of the switching times for the monitor sensor must be weighted depending upon the minimum value for the mass flow during the determination of the switching time of the lambda sensor MAF_KGH_MIN_PUC. The weighting factor is determined via the map IP_FAC_CYCNR_MAX_LS_DOWN_ACT.
If for the determined switching time of the monitor sensor doesn't exceed the limit the sensor is diagnosed as functioning, otherwise as malfunctioning. Depending on LC_SWT_LS_DOWN_MV the average value of the switching time determination or the

respective minimum value shall be evaluated. The minimum or maximum measured diagnosis values are saved for the scantool SAE 1979 accordingly.

Application conditions:

Initialisation:

```
If      At LV_IGK 0->1 or reset or at clearing error memory
then    reset of all variables and bits to 0, except:
        MAF_KGH_MIN_PUC = "max phys. limit"
        CTR_SAVE_SWT_LS_DOWN[i] which shall be initialized with its saved
        Value.  Reset only at clearing error memory.
        CTR_QUO_SWT_LS_DOWN_MIN[i] which shall be initialized with its maximum
        value
endif
```

Recurrence: 20 ms

Activation / Deactivation:

```
If      ((LV_DIAG_MPL_CDN_LS_DOWN[i] =1
and        LV_INH_DIAG_SWT_LS_DOWN[i] = 0)
or        LV_DIAG_RGN_NT_LS_DOWN_END[i] = 1)
and     LV_END_DIAG_SWT_LS_DOWN[i] = 0
then    LV_CDN_DIAG_SWT_LS_DOWN[i] =1
else    LV_CDN_DIAG_SWT_LS_DOWN[i] =0
        (switching time check disabled)
        VLS_DOWN_TRAN_PUC[i] = 0
        LV_SWT_DIAG_VLD_LS_DOWN[i] = 0
        CTR_SWT_ACT_LS_DOWN[i] = 0
        CTR_SWT_ACT_RBM_LS_DOWN[i] = 0
        MAF_KGH_MIN_PUC = "max phys. limit"
        LV_DIAG_SWT_LS_DOWN_S_ENA[i] = 0
endif
```

Formula section:

% Ensure that either upstream or downstream sensor indicates rich conditions for a minimum contiguous air
% mass integral. After such conditions, leaner conditions are allowed for a limited time. The upstream criterion
% is preferred, because a slow downstream sensor could make it hard to detect short rich phases, thus
% inhibiting its own detection. Fuel cut is a blocking criterion on its own, because lambda setpoint is
% not a reliable indicator in this condition.

```
If          LV_PUC = 0
then   If   LAMB_SP[i]$_{n}$ < C_LAMB_SP_AFR_MAX_SWT_S
            then   If     LAMB_SP[i]$_{n-1}$ >= C_LAMB_SP_AFR_MAX_SWT_S
                   then   reset MAF_INT_LAMB_AFR_SWT_S_ENA[i]
```

```
                        endif
                        MAF_INT_LAMB_AFR_SWT_S_ENA[i] + MAF_KGH * 0.02/3.6
                        reset T_LAMB_NOT_AFR_SWT_S_ENA[i]

                else    increment T_LAMB_NOT_AFR_SWT_S_ENA[i]
                        if      T_LAMB_NOT_AFR_SWT_S_ENA[i] >=
                                C_T_MAX_LAMB_NOT_AFR_SWT_S[NC_CBK_EX_NR]

                        then    reset MAF_INT_LAMB_AFR_SWT_S_ENA[i]
                        endif
                endif

                If      VLS_DOWN[i]$_n$ > C_VLS_DOWN_AFR_MIN_SWT_S

                then    If      VLS_DOWN[i]$_{n-1}$ <= C_VLS_DOWN_AFR_MIN_SWT_S
                        then    reset MAF_INT_VLS_AFR_SWT_S_ENA[i]
                        endif

                then    MAF_INT_VLS_AFR_SWT_S_ENA[i] + MAF_KGH * 0.02/3.6
                        reset T_VLS_NOT_AFR_SWT_S_ENA[i]

                else    increment T_VLS_NOT_AFR_SWT_S_ENA[i]
                        if      T_VLS_NOT_AFR_SWT_S_ENA[i] >=
                                C_T_MAX_VLS_NOT_AFR_SWT_S

                        then    reset MAF_INT_VLS_AFR_SWT_S_ENA[i]
                        endif
                endif
        else    reset MAF_INT_VLS_AFR_SWT_S_ENA[i]
                reset MAF_INT_LAMB_AFR_SWT_S_ENA[i]
        endif

        If   MAF_INT_VLS_AFR_SWT_S_ENA[i] > C_MAF_INT_MIN_VLS_AFR_SWT_S
        or   MAF_INT_LAMB_AFR_SWT_S_ENA[i] > C_MAF_INT_MIN_LAMB_AFR_SWT_S

        then    LV_DIAG_SWT_LS_DOWN_S_ENA[i] = 1
        endif

        If      (LV_PUC = 0 -> 1    and    LV_S_ACT = 0)

        then    VLS_DOWN_TRAN_PUC[i] = VLS_DOWN[i]

        endif

        % Check whether stable rich conditions have been reached before transition to lean combustion mode and
        % calculate a lean air mass integral for the last lean combustion period.
        If      LV_S_ACT = 0->1
        then    If      LC_DIAG_SWT_LS_DOWN_S_ENA = 1
                and     LV_DIAG_SWT_LS_DOWN_S_ENA[i] = 1

                then    VLS_DOWN_TRAN_PUC[i] = VLS_DOWN[i]

                else    VLS_DOWN_TRAN_PUC[i] = 0
                endif

                reset MAF_INT_S_MAX_SWT_S[i]
                LV_DIAG_SWT_LS_DOWN_S_ENA[i] = 0

        endif
```

```
If      LV_S_ACT = 1
then    MAF_INT_S_MAX_SWT_S[i] + MAF_KGH * 0.02/3.6
endif

% Lean combustion is an alternate way to start the diagnosis, if apropriate conditions are met (see above)
% In lean mode, stop the diagnosis if the sensor does not show the lean transition after a certain air
% mass consumption.
% Additionally, require a lean enough upstream lambda when the downstream transition occurs to ensure
% that the signal does not float around at stoichiometric.  If upstream is too rich, abort the current cycle.

If      (LV_PUC = 1
or       (LV_S_ACT = 1 and LC_DIAG_SWT_LS_DOWN_S_ENA = 1 and
          MAF_INT_S_MAX_SWT_S[i] < C_MAF_INT_S_MAX_SWT_S))

and     VLS_DOWN_TRAN_PUC[i] > C_VLS_PUC_MIN_DIAG_SWT_LS_DOWN

and     C_MAF_PUC_MIN_DIAG_SWT_LS_DOWN < MAF_KGH
                                 < C_MAF_PUC_MAX_DIAG_SWT_LS_DOWN

and     (VLS_DOWN[i]$_n$ < VLS_DOWN[i]$_{n-1}$ + C_VLS_HYS_DIAG_SWT_LS_DOWN
   or    (LV_S_ACT = 1      and      VLS_DOWN[i] >=
          C_FAC_VLS_MAX_DIAG_SWT_LS_DOWN * VLS_DOWN_TRAN_PUC[i]))
        % allow rising signal for stratified case, if it occurs before start of SWT measurement

then    If      VLS_DOWN[i]<C_FAC_VLS_MAX_DIAG_SWT_LS_DOWN *
                VLS_DOWN_TRAN_PUC[i]
        then    increment CTR_SWT_ACT_RBM_LS_DOWN[i]
                If      C_FAC_VLS_MIN_DIAG_SWT_LS_DOWN *
                                VLS_DOWN_TRAN_PUC[i] < VLS_DOWN[i]
                then    increment CTR_SWT_ACT_LS_DOWN[i]
                        MAF_KGH_MIN_PUC = MIN(MAF_KGH; MAF_KGH_MIN_PUC)
                else    freeze CTR_SWT_ACT_LS_DOWN[i]
                        MAF_KGH_MIN_PUC$_N$ = MAF_KGH_MIN_PUC$_{N-1}$
                        CTR_QUO_SWT_RBM_LS_DOWN[i] =
                                CTR_SWT_ACT_RBM_LS_DOWN[i] /
                                IP_FAC_CYCNR_MAX_LS_DOWN_ACT
                        If      (CTR_QUO_SWT_RBM_LS_DOWN[i] >=
                                        C_CTR_SWT_LS_DOWN)
                        and     LV_SWT_DIAG_VLD_LS_DOWN[i] = 0
                        then    LV_SWT_DIAG_VLD_LS_DOWN[i] = 1
                                increment CTR_SWT_LS_DOWN[i]
                                 CTR_QUO_SWT_LS_DOWN[i] =
                                        CTR_SWT_ACT_LS_DOWN[i] /
                                        IP_FAC_CYCNR_MAX_LS_DOWN_ACT
                                CTR_QUO_SWT_LS_DOWN_MIN[i] =
                                        MIN(CTR_QUO_SWT_LS_DOWN_MIN[i],
```

```
                                        CTR_QUO_SWT_LS_DOWN[i])
                                CTR_QUO_SWT_LS_DOWN_MAX[i] =
                                        MAX(CTR_QUO_SWT_LS_DOWN_MAX[i],
                                        CTR_QUO_SWT_LS_DOWN[i])
                                CTR_QUO_SUM_SWT_LS_DOWN[i] =
                                        CTR_QUO_SUM_SWT_LS_DOWN[i] +
                                        CTR_QUO_SWT_LS_DOWN[i]
                                CTR_CYCNR_SWT_LS_DOWN[i] =
                                        CTR_QUO_SUM_SWT_LS_DOWN[i] /
                                        CTR_SWT_LS_DOWN[i]
                        endif
                endif
                If      (LV_PUC = 0
                and     LAMB_SP_FIL_S < C_LAMB_SP_MIN_AFL_SWT_S)
                then    VLS_DOWN_TRAN_PUC[i] = 0
                endif
        endif
else    VLS_DOWN_TRAN_PUC[i] = 0
        LV_SWT_DIAG_VLD_LS_DOWN[i] = 0
        CTR_SWT_ACT_LS_DOWN[i] = 0
        CTR_SWT_ACT_RBM_LS_DOWN[i] = 0
        MAF_KGH_MIN_PUC = "max phys. limit"
endif
If      CTR_SWT_LS_DOWN[i] ≥ C_CTR_SWT_DIAG_LS_DOWN
then    If      (CTR_CYCNR_SWT_LS_DOWN[i] < C_CTR_SWT_LS_DOWN
        and     LC_SWT_LS_DOWN_MV = 1)
        or      (CTR_QUO_SWT_LS_DOWN_MIN[i]<C_CTR_SWT_LS_DOWN
        and     LC_SWT_LS_DOWN_MV = 0)
        then    LV_ERR_SWT_LS_DOWN[i] = 0
                ERR_SYM_SWT_LS_DOWN[i] = "NO_SYM"
                CTR_SAVE_SWT_LS_DOWN[i] = CTR_QUO_SWT_LS_DOWN_MIN[i]
        else    LV_ERR_SWT_LS_DOWN[i] = 1
                ERR_SYM_SWT_LS_DOWN[i] = " SYM _3"
                CTR_SAVE_SWT_LS_DOWN[i] = CTR_QUO_SWT_LS_DOWN_MAX[i]
        endif
        LV_END_DIAG_SWT_LS_DOWN[i] = 1
else
```

```
        if      LV_DIAG_RGN_NT_LS_DOWN_END[i] = 1
        then
                if      LV_DIAG_RGN_NT_LS_DOWN_ERR[i] = 0
                then    LV_ERR_SWT_LS_DOWN[i] = 0
                        ERR_SYM_SWT_LS_DOWN[i] = "NO_SYM"
                else    LV_ERR_SWT_LS_DOWN[i] = 1
                        ERR_SYM_SWT_LS_DOWN[i] = "SYM_2"
                Endif
                LV_END_DIAG_SWT_LS_DOWN[i] = 1
        else    % Switching time diagnosis going on
        endif
endif
```

Filtering :

```
Apply filter on current symptoms
If filtering result available  (after internal debounce; see Appl. Inc. ABC type: No)
    Delivery the result to Error Management
Endif
```

16.65.2 Monitoring the sensor voltage after leaving trailing throttle fuel cut-off

FUNCTION DESCRIPTION:

General information:

With level monitoring of the monitor sensor after leaving PUC unplausible voltage values due to short circuit to ground can be detected and a plausibility check upstream / downstream sensor can be made.

Description:

If the general conditions for activating the diagnostic function are fulfilled and If MAF_INT_PUC_NOT_ACT exceeds the value C_MAF_INT_DIAG_PUE_LS_DOWN, then the diagnosis must be made.

The maximum difference between the voltage value of the monitor sensor at the time of leaving PUC, VLS_DOWN_PUE[i] and the voltage value of the monitor sensor at the time at which the mass air flow MAF_INT_PUC_NOT_ACT exceeds the value C_MAF_INT_DIAG_PUE_LS_DOWN must be formed. Additionally the corresponding maximum and minimum values VLS_DIF_MIN_DIAG_PUE_LS_DOWN[i] and VLS_DIF_MAX_DIAG_PUE_LS_DOWN[i] shall be calculated. If the maximum difference lies below the limit C_VLS_DIF_MIN_DIAG_PUE_LS_DOWN, unplausible lambda sensor voltage of the post-Cat sensor after PUC is detected (ERR_SYM_PUE_LS_DOWN[i] = "SYM_3").
LV_ERR_PUE_LS_DOWN[i] = 1 as well as LV_END_DIAG_PUE_LS_DOWN[i] = 1 shall be set, if the counter reaches C_CTR_SYM_MAX_DIAG_PUE_LS_DOWN. Then also VLS_DIF_SAVE_DIAG_LS_DOWN[i] shall be set equal to VLS_DIF_MAX_DIAG_PUE_LS_DOWN[i].
If no unplausible lambda sensor voltage of the downstream sensor after PUC is detected ERR_SYM_PUE_LS_DOWN[i] = "NO_SYM" shall be set and the counter is decremented. After calibratable number of checks C_CTR_END_DIAG_PUE_LS_DOWN the end of diagnosis flag LV_END_DIAG_PUE_LS_DOWN[i] shall be set to 1 and VLS_DIF_SAVE_DIAG_LS_DOWN[i] equal to VLS_DIF_MIN_DIAG_PUE_LS_DOWN[i].

Application conditions:

Initialisation:

If At LV_IGK 0->1 **or** reset **or** at clearing error memory

then reset of all variables and bits to 0

except

- VLS_DIF_SAVE_DIAG_LS_DOWN which shall be initialized with its saved value. Reset only at clearing error memory.
- VLS_DIF_MIN_DIAG_PUE_LS_DOWN[i] = 4,99511V
- LV_DIAG_PUE_VLD_LS_DOWN[i] = 1

endif

Recurrence: 20 ms

Activation / Deactivation:

If LV_DIAG_MPL_CDN_LS_DOWN[i] =1

```
and     LV_INH_DIAG_PUE_LS_DOWN[i] = 0
and     LV_END_DIAG_PUE_LS_DOWN[i] = 0
then    LV_CDN_DIAG_PUE_LS_DOWN[i]  =1
        (leaving PUC check enabled)
else    LV_CDN_DIAG_PUE_LS_DOWN[i]  =0
        (leaving PUC check disabled)
        LV_DIAG_PUE_VLD_LS_DOWN[i] = 1
endif
```

Formula section:

```
If      LV_PUC = 1 ⇒ 0
then    VLS_DOWN_PUE[i] = VLS_DOWN[i]
        LV_DIAG_PUE_VLD_LS_DOWN[i] = 0
        VLS_DIF_DIAG_PUE_LS_DOWN[i] = 0V
endif
If      $\text{MAF\_INT\_PUC\_NOT\_ACT}_n < \text{MAF\_INT\_PUC\_NOT\_ACT}_{n-1}$
Then    LV_MAF_INT_PUC_NOT_ACT_RST[i] = 1
Endif
If      LV_PUC = 0
and     MAF_INT_PUC_ACT > C_MAF_INT_PUC_DIAG_PUE_LS_DOWN
and     LV_DIAG_PUE_VLD_LS_DOWN[i] = 0
then
        VLS_DIF_DIAG_PUE_LS_DOWN[i] = MAX(VLS_DOWN[i]- VLS_DOWN_PUE[i],
        VLS_DIF_DIAG_PUE_LS_DOWN[i])
        If      MAF_INT_PUC_NOT_ACT > C_MAF_INT_DIAG_PUE_LS_DOWN
        and     LV_MAF_INT_PUC_NOT_ACT_RST[i] = 1
        then    LV_MAF_INT_PUC_NOT_ACT_RST[i] = 0
                CTR_END_DIAG_PUE_LS_DOWN[i] =
                                CTR_END_DIAG_PUE_LS_DOWN[i] + 1
                LV_DIAG_PUE_VLD_LS_DOWN[i] = 1
                VLS_DIF_MAX_DIAG_PUE_LS_DOWN[i] =
                        MAX(VLS_DIF_MAX_DIAG_PUE_LS_DOWN[i],
                        VLS_DIF_DIAG_PUE_LS_DOWN[i])
                VLS_DIF_MIN_DIAG_PUE_LS_DOWN[i] =
                        MIN(VLS_DIF_MIN_DIAG_PUE_LS_DOWN[i],
                        VLS_DIF_DIAG_PUE_LS_DOWN[i])
```

```
                if      VLS_DIF_DIAG_PUE_LS_DOWN[i] >
                                C_VLS_DIF_MIN_DIAG_PUE_LS_DOWN
                then    CTR_SYM_DIAG_PUE_LS_DOWN[i] =
                                        CTR_SYM_DIAG_PUE_LS_DOWN[i] – 1
                else    CTR_SYM_DIAG_PUE_LS_DOWN[i] =
                                        CTR_SYM_DIAG_PUE_LS_DOWN[i] +
                                        C_CTR_SYM_INC_DIAG_PUE_LS_DOWN
                endif
        endif
endif


If      CTR_END_DIAG_PUE_LS_DOWN[i] ≥
                                        C_CTR_END_DIAG_PUE_LS_DOWN
then    LV_END_DIAG_PUE_LS_DOWN[i] = 1
        if      CTR_SYM_DIAG_PUE_LS_DOWN[i] ≥
                                        C_CTR_SYM_MAX_DIAG_PUE_LS_DOWN
        then    ERR_SYM_PUE_LS_DOWN[i] =  "SYM_3"
                LV_ERR_PUE_LS_DOWN[i] = 1
                VLS_DIF_SAVE_DIAG_LS_DOWN[i] =
                        VLS_DIF_MIN_DIAG_PUE_LS_DOWN[i]
        else    ERR_SYM_PUE_LS_DOWN[i] = "NO_SYM"
                LV_ERR_PUE_LS_DOWN[i] = 0
                VLS_DIF_SAVE_DIAG_LS_DOWN[i] =
                        VLS_DIF_MAX_DIAG_PUE_LS_DOWN[i]
        endif
endif
```

Filtering :

```
Apply filter on current symptoms
If filtering result available  (after internal debounce; see Appl. Inc. ABC type: No)
    Deliver the result to Error Management
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_MIN_DIAG_MPL_LS_DOWN	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant temperature for diagnosis of the monitor sensors					
C_VS_MIN_DIAG_MPL_LS_DOWN	1	0...FFH	0...255	1	[km/h]
Minimum vehicle speed for diagnosis of the monitor sensors					
C_VS_MAX_DIAG_MPL_LS_DOWN	1	0...FFH	0...255	1	[km/h]
Maximum vehicle speed for diagnosis of the monitor sensors					
C_TEMP_CAT_MIN_DIAG_MPL_LS_DOWN	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Catalyst temperature threshold for diagnosis of the monitor sensors					
C_MAF_INT_PUC_DIAG_PUE_LS_DOWN	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Threshold for activation of the diagnosis					
C_MAF_INT_DIAG_PUE_LS_DOWN	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Threshold for start of the diagnosis					
C_CTR_END_DIAG_PUE_LS_DOWN	1	0...FFFFH	0...65535	1	[-]
Counter threshold for CTR_END_DIAG_PUE_LS_DOWN_[NC_CBK_EX_NR]					
C_CTR_SYM_INC_DIAG_PUE_LS_DOWN	1	0...FFH	0...255	1	[-]
Counter increment for PUE_VLS_DOWN diagnosis					
C_CTR_SYM_MAX_DIAG_PUE_LS_DOWN	1	0...FFH	0...255	1	[-]
Maximum value of counter for PUE_VLS_DOWN diagnosis					
C_VLS_DIF_MIN_DIAG_PUE_LS_DOWN	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Threshold for VLS_DIF_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]					
C_MAF_PUC_MIN_DIAG_SWT_LS_DOWN	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Mass air flow threshold for diagnosis of the switching times					
C_MAF_PUC_MAX_DIAG_SWT_LS_DOWN	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Mass air flow threshold for diagnosis of the switching times					
C_VLS_PUC_MIN_DIAG_SWT_LS_DOWN	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage threshold for monitor sensor diagnosis					
C_FAC_VLS_MAX_DIAG_SWT_LS_DOWN	1	0...FFH	0...0.99609	3.9063e-3	[-]
Reduction factor for start of switching time determination					
C_FAC_VLS_MIN_DIAG_SWT_LS_DOWN	1	0...FFH	0...0.99609	3.9063e-3	[-]
Reduction factor for end of switching time determination					
C_VLS_HYS_DIAG_SWT_LS_DOWN	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Monitor sensor voltage hysteresis					
C_CTR_SWT_DIAG_LS_DOWN	1	0...FFH	0...255	1	[-]
Condition for the start of the lambda sensor switching time check					
C_CTR_SWT_LS_DOWN	1	0...FFH	0...1.99218	0.0078125	[-]
Diagnostic threshold for CTR_CYCNR_SWT_LS_DOWN[NC_CBK_EX_NR]					
IP_FAC_CYCNR_MAX_LS_DOWN_ACT	8	0...FFH	0...5100	20	[ms]
LDP_MAF_KGH_IP_FAC_LS_DOWN	8	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Weighting factor for CTR_SWT_ACT_LS_DOWN[NC_CBK_EX_NR]					
LC_SWT_LS_DOWN_MV	1	0...1H	0...1	1	[-]
Logical calibration data for activation of mean value evaluation of switching time diagnosis					
LC_DIAG_SWT_LS_DOWN_S_ENA	1	0...1H	0...1	1	[-]
Logical calibration data for activating downstream SWT diagnosis on rich-lean combustion mode transistions					
C_LAMB_SP_AFR_MAX_SWT_S	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Maximum lambda to determine rich conditions for stratified SWT diagnosis (upstream lambda criterion)					
C_MAF_INT_MIN_LAMB_AFR_SWT_S	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Minimum air mass to determine rich conditions for stratified SWT diagnosis (upstream lambda criterion)					
C_T_MAX_LAMB_NOT_AFR_SWT_S[NC_CBK_EX_NR]	1	0...FFFFH	0...1310.7	0.02	[s]
Maximum time after leaving rich conditions for stratified SWT diagnosis (upstream lambda criterion)					
C_VLS_DOWN_AFR_MIN_SWT_S	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Minimum signal to determine rich conditions for stratified SWT diagnosis (downstream lambda criterion)					
C_MAF_INT_MIN_VLS_AFR_SWT_S	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Minimum air mass to determine rich conditions for stratified SWT diagnosis (downstream lambda criterion)					
C_T_MAX_VLS_NOT_AFR_SWT_S	1	0...FFFFH	0...1310.7	0.02	[s]
Maximum time after leaving rich conditions for stratified SWT diagnosis (downstream lambda criterion)					
C_MAF_INT_S_MAX_SWT_S	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Air mass limit after which diagnosis cycle is aborted if no signal transition occured					

C_LAMB_SP_MIN_AFL_SWT_S	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Minimum upstream lambda at the moment of signal transition to validate a switching time measurement					

16.66 Diagnosis of monitor sensors (Applic. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_SWT_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag indicating inhibition of lambda sensor switching time diagnosis downstream					
LV_INH_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag indicating inhibition of lambda sensor PUC-end diagnosis downstream					
STATE_RBM_SWT_LS_DOWN[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Interface of SWT_LS_DOWN monitor with the Rate-Based Monitoring statistics					
LV_INH_DIAG_RBM_SWT_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag indicating inhibition of RBM for lambda sensor switching time diagnosis downstream					
LV_INH_DIAG_TMP_LS_DOWN	V	0...1H	0...1	1	[-]
Flag indicating inhibition for PUE_ and SWT_LS_DOWN					

Input data:

LV_VB_CDN_OBD_2	CTR_ERR_DYN_NR	LV_ERR_LS_UP[NC_CBK_EX_NR]
LV_ERR_AMP	LV_ERR_AMP_PLAUS	LV_ERR_FSD[NC_CBK_EX_NR]
LV_ERR_CHG_LS_UP	LV_ERR_MAP_DIP_SENS	
LV_ERR_EL_CPS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_LOAD_TPS_PLAUS
LV_ERR_CRK	LV_ERR_MAP_DIP_SHIFT	LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_DIAGCPS	LV_ERR_IVVT	LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SA_SAV_LSL	LV_IGK	LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_MAF	LV_MIS_STATE_A	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_SA_SAP	LV_MIS_STATE_B	LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SA_SAV	LV_ST_END	LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SA_SYS	NC_CBK_EX_NR	LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SAP	LV_ERR_TCO	LV_ERR_SWT_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SAV	LV_ERR_TPS	LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_CAM_CUS	LV_DC	LV_END_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]	LV_END_DIAG_SWT_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_FUP	LV_LS_UP_OBD_2_MAN_DEAC[NC_CBK_EX_NR]

LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV	LV_ERR_CHG_LS_DOWN
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_CAT_DIAG[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_IN	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS
R_IT_LS_DOWN[NC_CBK_EX_NR]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit

LC_INH_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Manual deactivation of the PUE diagnosis(when = 1)					
LC_INH_DIAG_SWT_LS_DOWN[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Manual deactivation of the SWT diagnosis (when = 1)					
LC_INH_LS_DOWN_MAN_DEAC	1	0...1H	0...1	1	[-]
Manual deactivation of inhibit conditions (when = 1)					
C_R_IT_MAX_DIAG_MPL_LS_DOWN	1	0...FFFFH	0...65535	1	[Ohm]
Maximum allowed internal resistance of the downstream sensor for downstream dynamic diagnosis					

16.66.1 Inhibition of diagnosis of the monitor sensors

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then

i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2

otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

If one of the below mentioned conditions is not fulfilled, the inhibit bit interrupts the downstream oxygen sensor diagnosis.

Application conditions:

Initialisation: at every transition LV_IGK = 0 -> 1or reset all variables shall be reset to 0, except STATE_RBM_SWT_LS_DOWN[i] *//for init of this variable see section below*

at every transition LV_IGK = 1 -> 0

- LV_INH_DIAG_SWT_LS_DOWN[NC_CBK_EX_NR] = 1
- LV_INH_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR] = 1

Recurrence: 100 ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

If LV_ERR_AMP = 0
and LV_ERR_AMP_PLAUS = 0
and LV_ERR_CRK = 0
and LV_ERR_CAM_CUS = 0
and LV_ERR_TCO = 0
and LV_ERR_MAF = 0
and LV_ERR_MAP_DIP_SENS = 0
and LV_ERR_MAP_DIP_PLAUS = 0

```
and     LV_ERR_MAP_DIP_SHIFT = 0
and     LV_ERR_EL_CPS = 0
and     LV_ERR_DIAGCPS = 0
and     LV_ERR_SA_SAV = 0
and     LV_ERR_SA_SAV_LSL = 0
and     LV_ERR_SA_SAP = 0
and     LV_ERR_SA_SYS = 0
and     LV_ERR_SAV = 0
and     LV_ERR_SAP = 0
and     LV_MIS_STATE_A = 0
and     LV_MIS_STATE_B = 0
and     LV_ERR_FSD[i] = 0
and     LV_ERR_CHG_LS_UP = 0
and     LV_ERR_DELTA_I_LAM[i] = 0
and     LV_ERR_VLS_DOWN_DIF[i] = 0
and     LV_ERR_FSD_LAM_LIM[i] = 0
and     LV_ERR_FUP = 0
and     LV_ERR_FUP_MFP_PLAUS = 0
and     LV_ERR_H_PRS_SYS = 0
and     LV_ERR_VCV = 0
and     LV_ERR_CHG_LS_DOWN = 0
and     LV_ERR_FUP_ORNG = 0
and     LV_ERR_FUP_ST = 0
and     LV_ERR_TTIP_MES_LSH_UP[i] = 0
and     LV_ERR_LOAD_TPS_PLAUS = 0
and     LV_ERR_LS_UP[i] = 0
and     LV_ERR_FL_LS_DOWN[i] = 0
and     LV_ERR_PUC_LS_DOWN[i] = 0
and     LV_ERR_LSH_DOWN[i] = 0
and     LV_ERR_SCG_LS_DOWN[i] = 0
and     LV_ERR_SCP_LS_DOWN[i] = 0
and     LV_ERR_OC_LS_DOWN[i] = 0
and     LV_ERR_OBD_LSH_DOWN[i] = 0
and     LV_ERR_CHK_LS_DOWN[i] = 0
and     LV_ERR_IVVT = 0
then    LV_INH_DIAG_TMP_LS_DOWN[i] = 0
else    LV_INH_DIAG_TMP_LS_DOWN [i] = 1
endif

If      LV_INH_DIAG_TMP_LS_DOWN = 1        or
        LV_ERR_TPS = 1                     or
        LV_ERR_CAT_DIAG[i] = 1             or
        LV_ERR_SLV_IVVT_IN = 1             or
        LV_ERR_OFS_LSL_UP[i] = 1           or
        LV_ERR_CTL_LSL_UP[i] = 1           or
        LV_ERR_MAP_TPS_PLAUS = 1

then    LV_INH_DIAG_RBM_SWT_LS_DOWN = 1
else    LV_INH_DIAG_RBM_SWT_LS_DOWN = 0
endif
```

```
If      ((LV_INH_DIAG_RBM_SWT_LS_DOWN [i] = 1                or
        LV_VB_CDN_OBD_2 = 0                                   or
        R_IT_LS_DOWN[i] > C_R_IT_MAX_DIAG_MPL_LS_DOWN         or
        LV_LS_UP_OBD_2_MAN_DEAC[i] = 1)                       and
        LC_INH_LS_DOWN_MAN_DEAC = 0)                          or
        LC_INH_DIAG_SWT_LS_DOWN[i] = 1                        or

then    LV_INH_DIAG_SWT_LS_DOWN[i] = 1
else    LV_INH_DIAG_SWT_LS_DOWN[i] = 0
endif
```

```
If      ((LV_INH_DIAG_TMP_LS_DOWN [i] = 1    or
        LV_VB_CDN_OBD_2 = 0                   or
        LV_LS_UP_OBD_2_MAN_DEAC[i] = 1)       and
        LC_INH_LS_DOWN_MAN_DEAC = 0)          or
        LC_INH_DIAG_PUE_LS_DOWN[i] = 1        or
        LV_ERR_SLV_IVVT_IN = 1                or
        LV_ERR_OFS_LSL_UP[i] = 1              or
        LV_ERR_CTL_LSL_UP[i] = 1              or
        LV_ERR_MAP_TPS_PLAUS = 1
then    LV_INH_DIAG_PUE_LS_DOWN[i] = 1
else    LV_INH_DIAG_PUE_LS_DOWN[i] = 0
endif
```

16.66.2 ERRM configuration for monitor diagnosis

Diagnosis	Symptom	Nr	ABC type
Oxygen sensor downstream rich lean switch time check		*0*	NO
		1	
		2	
SWT_LS_DOWN	*SYM_3*	*3*	

Diagnosis	Symptom	Nr	ABC type
Oxygen sensor downstream monitoring sensor voltage after PUC check		*0*	NO
		1	
		2	
PUE_LS_DOWN	*SYM_3*	*3*	

16.66.3 RBM – Interface for monitor sensor diagnosis

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the SWT_LS_DOWN[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_SWT_LS_DOWN[i] data.

Within STATE_RBM_SWT_LS_DOWN[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for SWT_LS_DOWN[i] diagnosis)

Application conditions:

Initialisation :

at ECU reset all bits of STATE_RBM_SWT_LS_DOWN[i] shall be initialized with 0

at LV_DC 0 ->1 transition :

bit 0 and bit 1 of STATE_RBM_SWT_LS_DOWN[i] = 0

bit 2 of STATE_RBM_SWT_LS_DOWN[i] = 1

on failure memory reset :

bit 1 of STATE_RBM_SWT_LS_DOWN[i] = 0

Recurrence: 1 s

Activation: LV_DC 0 -> 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 -> 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_MAP_DIP_SHIFT
LV_ERR_MEC_IVVT_IN[NC_NR_CBK_IVVT]	LV_ERR_MEC_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_SLV_IVVT_EX[NC_NR_CBK_IVVT]
	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]
LV_ERR_CRK_PLAUS	LV_ERR_CRK_TOOTH_PER	LV_ERR_CRK_SYN
LV_ERR_CRK_TOOTH	LV_ERR_PLAUS_CAM_IN_1	LV_ERR_PLAUS_CAM_EX_1
LV_ERR_PER_CAM_IN_1	LV_ERR_PER_CAM_EX_1	LV_ERR_REF_CRK_CAM_IN_1
LV_ERR_REF_CRK_CAM_EX_1	LV_ERR_SYN_CAM_EX_1	LV_ERR_SYN_CAM_IN_1
LV_ERR_MAF	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_AMP
LV_ERR_AMP_PLAUS	LV_ERR_EL_CPS	LV_ERR_DYN_VLD_LS_UP[NC_CBK_EX_NR]
LV_ERR_DIAGCPS	LV_ERR_TCO_EL	LV_ERR_FSD[NC_CBK_EX_NR]
LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS	LV_ERR_TCO_STUCK
LV_ERR_MIS[NC_CYL_NR]	LV_ERR_SAP	LV_ERR_SAV
LV_ERR_SA_SAP	LV_ERR_SA_SYS	
LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_AIR_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_PUC_VLD_LS_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]
LV_ERR_LSH_LSL_UP[NC_CBK_EX_NR]	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]
LV_ERR_LSH_UP[NC_CBK_EX_NR]	LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_SA_SAV_LSL

LV_ERR_TPS_RATIO	LV_ERR_TPS_AD	LV_ERR_TPS_AD_BOL	LV_ERR_TPS_MAF[i]
LV_ERR_TPS_ST_CHK_2	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3
LV_ERR_MTC_DR	LV_ERR_LSL_UP_IF[i]	LV_ERR_DYN_VLD_LS_UP[i]	LV_ERR_PUC_VLD_LS_UP[i]
LV_ERR_SHIFT_AFL_LSL_UP[i]	LV_ERR_SHIFT_AFR_LSL_UP[i]	LV_ERR_LSH_LSL_UP[i]	LV_ERR_OBD_VLD_LSH_UP[i]
LV_ERR_AIR_LSL_UP[i]	LV_ERR_EL_LSL_UP[i]	LV_ERR_OC_LSL_UP[i]	LV_ERR_LSH_UP[i]
LV_ERR_TCO_STUCK_RNG	LV_ERR_SYN_CRK_CAM_IN_1	LV_ERR_SYN_CRK_CAM_EX_1	LV_ERR_TPS_1
LV_ERR_TPS_2	LV_ERR_TOOTH_OFF_EX_1	LV_ERR_TOOTH_OFF_IN_1	LV_ERR_SLV_IVVT_IN
LV_ERR_OFS_LSL_UP[i]	LV_ERR_CTL_LSL_UP[i]	LV_ERR_MAP_TPS_PLAUS	

```
If(1)       { CPU optimization at LV_DC 0 -> 1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While  bit 1 of STATE_RBM_SWT_LS_DOWN[i] = 0 do
                   with each XX failure of the above list  (depending on configuration, see
                   below):
                   ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                   SYNCHRONIZATION<CALL>)

                   If(2)          XX has a pending status
```

```
                    Then(2)     bit 1 of
                                    STATE_RBM_SWT_LS_DOWN[i] = 1
                    Endif(2)
               Endwhile
   Else(1)     { the dynamic failure memory is empty }      No action
   Endif(1)
```

Every 1 s :

```
If    bit 0 of STATE_RBM_SWT_LS_DOWN[i] = 0
Then        If      LV_END_DIAG_SWT_LS_DOWN[i] = 1
            Then    bit 0 of STATE_RBM_SWT_LS_DOWN[i] = 1
            Endif
Endif

If    bit 1 of STATE_RBM_SWT_LS_DOWN[i] = 0
Then
      If      LV_INH_DIAG_RBM_SWT_LS_DOWN[i]= 1
      Then    bit 1 of STATE_RBM_SWT_LS_DOWN[i] = 1
      Endif
Endif
```

16.67 O2 sensor (bin, down) dynamic diagnosis after NOx cat regeneration

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_DOWN_RGN_NT_END[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Maximum monitor sensor voltage during last regeneration phase of NOx trap, Bank i (i=1.2)					
VLS_DOWN_RGN_NT_MIN_END[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Minimum monitor sensor voltage during MAF integral calculation in stratified mode					
VLS_DOWN_RGN_NT_MIN_TMP[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Temporary minimum monitor sensor voltage during MAF integral calculation in stratified mode					
VLS_DOWN_RGN_NT_MAX_TMP[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Temporary maximum monitor sensor voltage during MAF integral calculation in stratified mode					
LV_DIAG_RGN_NT_LS_DOWN_ERR[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Debounced diagnosis result for monitoring lean to rich switching times after Nox cat regeneration					
LV_DIAG_ACT_RGN_NT_LS_DOWN[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Logical variable indicating that diagnosis conditions are fulfilled					
LV_DIAG_RGN_NT_LS_DOWN_END[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
End bit for monitoring lean to rich switching times after Nox cat regeneration					
MAF_INT_RGN_NT_S_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...2912.66666	0.0444444	[g]
Mass air flow which has passed since activation of stratified mode					
MAF_INT_RGN_NT_END[NC_CBK_EX_NR]	V	0...FFFFH	0...2912.66666	0.0444444	[g]
Mass air flow which has passed since last regeneration cycle					
T_RGN_NT_END[NC_CBK_EX_NR]	V	0...FFFFH	0...1310.7	0.02	[s]
Time which has passed after leaving last regeneration cycle					
VLS_DOWN_RGN_NT_DIF[NC_CBK_EX_NR]	V	FC00...3FFH	-5...4.99511	4.8828e-3	[V]
Measured Voltage difference used for error detection					
CTR_END_DIAG_RGN_NT_LS_DOWN[NC_CBK_EX_NR]	V	0...FFH	0...255	1	[-]
Counter for end bit calculation					
CTR_SYM_DIAG_RGN_NT_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Counter for internal anti-bounce of error symptom					
VLS_DOWN_RGN_NT_MES_SAE[NC_CBK_EX_NR]	O/V/S	0...FFFFH	0...7.99987	0.1221e-3	[V]
Test value for diagnosis result evaluation for mode 6 communication					
VLS_DOWN_RGN_NT_MIN_SAE[NC_CBK_EX_NR]	O/V	0...FFFFH	0...7.99987	0.1221e-3	[V]
Minimum voltage threshold for diagnosis result evaluation for mode 6 communication					

Input data:

LV_IGK	LV_S_ACT	LV_NT_RGN_REQ	LV_NT_RGN_REQ_AD
VLS_DOWN[NC_CBK_EX_NR]	MAF_KGH	LV_INH_DIAG_RGN_NT_LS_DOWN[NC_CBK_EX_NR]	TEMP_CAT_DYN_MDL[NC_CBK_EX_NR]
LV_LS_DOWN_READY[NC_CBK_EX_NR]	TCO	STATE_RGN[NC_CBK_EX_NR]	NC_CBK_EX_NR

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

With level monitoring of the monitor sensor after leaving regeneration phase of NOx trap unplausible voltage values can be detected and a plausibility check of the sensor can be made.

Description:

If the general conditions for activating the diagnostic function are fulfilled and if during a regeneration phase either the state STATE_RGN[i] is set to L_AMPL or LC_STATE_RGN_L_AMPL_DEAC = 1 (withdrawal of monitoring STATE_RGN[i] depending on regeneration strategy) the lambda probe voltage and its previous value shall be stored.

Upon leaving the regeneration phase the previously measured voltage shall be compared to the threshold which, when exceeded, allows to activate the diagnosis as indicated by LV_DIAG_ACT_RGN_NT_LS_DOWN[i] = 1. Additionally the timer T_RGN_NT_END[i] and the mass air flow integral MAF_INT_RGN_NT_END[i] are calculated until the next stratified combustion mode is activated.

Provided that the stratified mode is activated before the timer T_RGN_NT_END[i] or the MAF integral MAF_INT_RGN_NT_END[i] exceed their respective threshold values monitoring the lambda probe voltage after leaving the regeneration phase shall be started and the MAF integral MAF_INT_RGN_NT_S_ACT[i] shall be calculated. After exceeding the MAF integral threshold C_MAF_INT_RGN_NT_S_ACT, i.e. if the stratified mode has been active long enough, the voltage difference between leaving the regeneration phase and exceeding the MAF integral threshold shall be calculated as expressed by VLS_DOWN_RGN_NT_DIF[i].

Comparison of the voltage difference to a threshold value allows to recognize a sensor malfunction. Each time the voltage comparison is made the counter for end of diagnosis recognition shall be increased. Provided that the measured voltage difference does not exceed its respective threshold C_VLS_DOWN_RGN_NT_DIF_THD the symptom counter CTR_SYM_DIAG_RGN_NT_LS_DOWN[i] shall be incremented by the calibratable increment C_CTR_INC_DIAG_RGN_NT_LS_DOWN and decremented otherwise with a decrement of 1.

The final diagnosis result is drawn after the end cycle counter has reached the calibratable minimum number of necessary diagnostic cycles C_CTR_END_DIAG_RGN_NT_LS_DOWN. In this case the end indication flag shall be set and the error indication flag shall be set following an n-out-of-m statistical evaluation for the case that the symptom counter has exceeded the corresponding threshold C_CTR_SYM_MAX_RGN_NT_ LS_DOWN. The result shall not be transmitted to the ERRM directly, since another diagnosis is expected to include it in its error and end state considerations.

Error-symptoms and conditions:

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │ │ │ └─> (not used)                                          (= SYM_0)
  │ │ └───> (not used)                                          (= SYM_1)
  │ └─────> (not used)                                          (= SYM_2)
  └───────> (lean voltage not reached after NOx cat. regeneration)  (= SYM_3)
```

Application conditions:

Initialisation:

At reset or the transition of LV_IGK = 0 -> 1 or at clearing the error memory all variables shall be reset to 0, except VLS_DOWN_RGN_NT_MES_SAE[i] and VLS_DOWN_RGN_NT_ MIN_SAE[i].

Initialisation in case of new ECU and NVMY checksum errors:
VLS_DOWN_RGN_NT_MES_SAE[i] = 0V

Initialisation in case of reset:

VLS_DOWN_RGN_NT_MIN_SAE[i] = C_VLS_DOWN_RGN_NT_DIF_THD

Recurrence: 20 ms

Activation: At any engine operating state

Formula section:

Check diagnosis availability

```
If      LV_INH_DIAG_RGN_NT_LS_DOWN[i] = 0
and     LV_DIAG_RGN_NT_LS_DOWN_END[i] = 0
and     TCO > C_TCO_MIN_DIAG_RGN_NT_LS_DOWN
and     LV_LS_DOWN_READY[i] = 1
and     C_MAF_MIN_DIAG_RGN_NT_LS_DOWN <= MAF_KGH
                                  <= C_MAF_MAX_DIAG_RGN_NT_LS_DOWN
and     TEMP_CAT_DYN_MDL[i] > C_TEMP_MIN_DIAG_RGN_NT_LS_DOWN
then    % diagnosis available
        if      LV_NT_RGN_REQ = 0 -> 1
        then    VLS_DOWN_RGN_NT_END[i] = 0  % reinitialize for a new cycle
        endif
        if      LV_NT_RGN_REQ = 1
        and     (STATE_RGN[i] = L_AMPL
        or      LC_STATE_RGN_L_AMPL_DEAC = 1)
        and     LV_NT_RGN_REQ_AD = 0
```

```
        then  VLS_DOWN_RGN_NT_END[i] =
              MAX (VLS_DOWN[i]$_{(n-1)}$, VLS_DOWN_RGN_NT_END[i])
        endif
        if    LV_NT_RGN_REQ = 1 -> 0
        and   VLS_DOWN_RGN_NT_END[i] > C_VLS_MIN_DIAG_RGN_LS_DOWN
        then  VLS_DOWN_RGN_NT_MIN_END[i] = physical maximum limit
              LV_DIAG_ACT_RGN_NT_LS_DOWN[i] = 1
              diagnosis available
        endif
else    % diagnosis not available
        LV_DIAG_ACT_RGN_NT_LS_DOWN[i] = 0
        VLS_DOWN_RGN_NT_END[i] = 0
        VLS_DOWN_RGN_NT_DIF[i] = 0
        MAF_INT_RGN_NT_END[i] = 0
        MAF_INT_RGN_NT_S_ACT[i] = 0
        T_RGN_NT_END[i] = 0
endif
```

Conditions for activation of diagnosis fulfilled

```
If      LV_DIAG_ACT_RGN_NT_LS_DOWN[i] = 1
then    if    LV_S_ACT = 0
        then  Check if stratified mode activated in time
              MAF_INT_RGN_NT_S_ACT[i] = 0
              if    ( MAF_INT_RGN_NT_END[i] >= C_MAF_INT_RGN_NT_END
              or    T_RGN_NT_END[i] >= C_T_RGN_NT_END )
              then  LV_DIAG_ACT_RGN_NT_LS_DOWN[i] = 0
                    MAF_INT_RGN_NT_END[i] = 0
                    T_RGN_NT_END[i] = 0
                    VLS_DOWN_RGN_NT_END[i] = 0
              else  MAF_INT_RGN_NT_END[i] =
                    MAF_INT_RGN_NT_END[i] + MAF_KGH * 0.02/3.6
                    T_RGN_NT_END[i] = T_RGN_NT_END[i] + T_SAMPLE
              endif
        else  Monitoring of lambda probe voltage after NOx catalyst regeneration
              MAF_INT_RGN_NT_END[i] = 0
              T_RGN_NT_END[i] = 0
```

```
                if      MAF_INT_RGN_NT_S_ACT[i] > C_MAF_INT_RGN_NT_S_ACT
                then    VLS_DOWN_RGN_NT_DIF[i] = VLS_DOWN_RGN_NT_END[i] –
                        VLS_DOWN_RGN_NT_MIN_END[i]
                        LV_DIAG_ACT_RGN_NT_LS_DOWN[i] = 0
                        CTR_END_DIAG_RGN_NT_LS_DOWN[i] =
                        CTR_END_DIAG_RGN_NT_LS_DOWN[i] + 1
                        VLS_DOWN_RGN_NT_END[i] = 0
                        MAF_INT_RGN_NT_S_ACT[i] = 0
                        If      VLS_DOWN_RGN_NT_DIF[i] <
                                C_VLS_DOWN_RGN_NT_DIF_THD
                        then    CTR_SYM_DIAG_RGN_NT_LS_DOWN[i] =
                                CTR_SYM_DIAG_RGN_NT_LS_DOWN[i] +
                                C_CTR_INC_DIAG_RGN_NT_LS_DOWN
                                VLS_DOWN_RGN_NT_MIN_TMP[i] =
                                                VLS_DOWN_RGN_NT_DIF[i]
                        else    CTR_SYM_DIAG_RGN_NT_LS_DOWN[i] =
                                CTR_SYM_DIAG_RGN_NT_LS_DOWN[i] - 1
                                VLS_DOWN_RGN_NT_MAX_TMP[i] =
                                                VLS_DOWN_RGN_NT_DIF[i]
                        endif
                else    MAF_INT_RGN_NT_S_ACT[i] =
                        MAF_INT_RGN_NT_S_ACT[i] + MAF_KGH * 0.02/3.6
                        VLS_DOWN_RGN_NT_MIN_END[i] =
                        MIN(VLS_DOWN[i], VLS_DOWN_RGN_NT_MIN_END[i]_{n-1})
                endif
        endif
endif
```

Evaluation of final diagnosis result

```
If      CTR_END_DIAG_RGN_NT_LS_DOWN[i] >
        C_CTR_END_DIAG_RGN_NT_LS_DOWN
then    LV_DIAG_RGN_NT_LS_DOWN_END[i] = 1
        if      CTR_SYM_DIAG_RGN_NT_LS_DOWN[i] >
                C_CTR_SYM_MAX_RGN_NT_LS_DOWN
        then
                LV_DIAG_RGN_NT_LS_DOWN_ERR[i] = 1
                VLS_DOWN_RGN_NT_MES_SAE[i] = VLS_DOWN_RGN_NT_MIN_TMP[i]
```

```
        else
                LV_DIAG_RGN_NT_LS_DOWN_ERR[i] = 0
                VLS_DOWN_RGN_NT_MES_SAE[i] = VLS_DOWN_RGN_NT_MAX_TMP[i]
        endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_MIN_DIAG_RGN_NT_LS_DOWN	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant temperature for activation of RGN_NT_LS_DOWN[i] diagnosis					
C_MAF_MIN_DIAG_RGN_NT_LS_DOWN	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Minimum mass air flow for activation of RGN_NT_LS_DOWN[i] diagnosis					
C_MAF_MAX_DIAG_RGN_NT_LS_DOWN	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Maximum mass air flow for activation of RGN_NT_LS_DOWN[i] diagnosis					
C_TEMP_MIN_DIAG_RGN_NT_LS_DOWN	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Catalyst temperature threshold for activation of RGN_NT_LS_DOWN[i] diagnosis					
LC_STATE_RGN_L_AMPL_DEAC	1	0...1H	0...1	1	[-]
Logical calibration data to skip check of STATE_RGN[i] depending on regeneration strategy					
C_VLS_DOWN_RGN_NT_DIF_THD	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Measured Voltage difference used for error detection					
C_VLS_MIN_DIAG_RGN_LS_DOWN	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Minimum voltage during regeneration phase to enable diagnosis					
C_MAF_INT_RGN_NT_S_ACT	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Mass air flow which has passed since activation of stratified mode					
C_MAF_INT_RGN_NT_END	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Mass air flow which has passed since last regeneration cycle					
C_T_RGN_NT_END	1	0...FFFFH	0...1310.7	0.02	[s]
Threshold for time which has passed after leaving last regeneration cycle					
C_CTR_INC_DIAG_RGN_NT_LS_DOWN	1	0...FFH	0...255	1	[-]
Counter increment for present error symptom					
C_CTR_SYM_MAX_RGN_NT_LS_DOWN	1	0...FFFFH	0...65535	1	[-]
Threshold for detection of downstream sensor dynamic error after NOx cat regeneration					
C_CTR_END_DIAG_RGN_NT_LS_DOWN	1	0...FFH	0...255	1	[-]
Counter threshold for setting of diagnosis end bit					

16.68 O2 sensor (bin, down) dyn diag after NOx cat reg (Appl. Inc.)

1.1. Calculation of inhibition for diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_RGN_NT_LS_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Logical variable indicating inhibition of downstream oxygen sensor diagnosis after NOx cat regeneration					

Input data:

LV_ERR_DIAGCPS	LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_MAF	LV_ERR_IV[NC_CYL_NR]
LV_ERR_CAT_DIAG[NC_CBK_EX_NR]	LV_ERR_EL_CPS	LV_ERR_MEC_OPEN_CPS	LV_ERR_TCO
LV_ERR_TPS	LV_LS_DOWN_OBD_2_MAN_DEAC[NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_ERR_IVVT
	LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS	LV_ERR_MAP_DIP_SHIFT
LV_ERR_CHG_LS_UP	LV_ERR_DIAGCPS	LV_MIS_STATE_A	LV_MIS_STATE_B
LV_ERR_CRK	LV_ERR_CAM_CUS	LV_ERR_AMP	LV_ERR_AMP_PLAUS
LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_FUP
LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV	LV_ERR_CHG_LS_DOWN
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST		LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]
LV_ERR_SLV_IVVT_IN	LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_DIAG_RGN_NT_LS_DOWN[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Manual deactivation of the RGN_NT_LS_DOWN diagnosis (when = 1)					

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
    i = 1, for exhaust cylinder bank 1
    i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
    i = 1, for single exhaust cylinder bank.

Application conditions:

Initialization: *at reset all variables shall be initialized with 1*

At the transition LV_IGK = 0 -> 1 or ECU-Reset:

LV_INH_DIAG_RGN_NT_LS_DOWN[i] = 0
At the transition LV_IGK = 1 -> 0
LV_INH_DIAG_RGN_NT_LS_DOWN[i] = 1

Recurrence: *20 ms*

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

```
IF  LV_ERR_LS_DOWN[i] = 1                                          OR
    LV_ERR_FSD[i] = 1                                              OR
    LV_ERR_MAF = 1                                                 OR
    LV_ERR_MAP_DIP_SENS = 1                                        OR
    LV_ERR_MAP_DIP_PLAUS = 1                                       OR
    LV_ERR_MAP_DIP_SHIFT = 1                                       OR
    LV_ERR_CHG_LS_UP = 1                                           OR
    LV_ERR_DIAGCPS = 1                                             OR
    LV_MIS_STATE_A = 1                                             OR
    LV_MIS_STATE_B = 1                                             OR
    LV_ERR_CRK = 1                                                 OR
    LV_ERR_CAM_CUS = 1                                             OR
    LV_ERR_AMP = 1                                                 OR
    LV_ERR_AMP_PLAUS = 1                                           OR
    LV_ERR_DELTA_I_LAM[i] = 1                                      OR
    LV_ERR_VLS_DOWN_DIF[i] = 1                                     OR
    LV_ERR_FSD_LAM_LIM[i] = 1                                      OR
    LV_ERR_FUP = 1                                                 OR
    LV_ERR_FUP_MFP_PLAUS = 1                                       OR
    LV_ERR_H_PRS_SYS = 1                                           OR
    LV_ERR_VCV = 1                                                 OR
    LV_ERR_CHG_LS_DOWN = 1                                         OR
    LV_ERR_FUP_ORNG = 1                                            OR
    LV_ERR_FUP_ST = 1                                              OR
    LV_ERR_TTIP_MES_LSH_UP[i] = 1                                  OR
    LV_ERR_IV[x] = 1  (x = 0,2,4 for Bank1; x=1,3,5 for Bank2)     OR
    LV_ERR_IVVT = 1                                                OR
    LV_ERR_CAT_DIAG[i] = 1                                         OR
    LV_ERR_EL_CPS = 1                                              OR
    LV_ERR_MEC_OPEN_CPS = 1                                        OR
    LV_ERR_TCO = 1                                                 OR
    LV_ERR_TPS = 1                                                 OR
    LV_LS_DOWN_OBD_2_MAN_DEAC[i] = 1                               OR
    LC_INH_DIAG_RGN_NT_LS_DOWN[i]  = 1                             OR
    LV_ERR_SLV_IVVT_IN = 1                                         OR
    LV_ERR_OFS_LSL_UP[i] = 1                                       OR
    LV_ERR_CTL_LSL_UP[i] = 1                                       OR
    LV_ERR_MAP_TPS_PLAUS = 1
THEN
    LV_INH_DIAG_RGN_NT_LS_DOWN[i] = 1
ELSE
    LV_INH_DIAG_RGN_NT_LS_DOWN[i] = 0
ENDIF
```

16.69 Downstream oxygen sensor heater OBDII monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CDN_DIAG_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that conditions for diagnosis met					
ERR_SYM_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Variable indicating status of each symptom, updated each recurrence & at diagnosis completion					
LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that fault is present and has been debounced					
LV_END_DIAG_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean flag indicating that diagnosis has been completed					
STATE_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0H 1H 2H 3H 4H 5H 6H 7H 8H	DIAG_OFF DIAG_INIT LS_READY OBD1_DLY OBD1_CHK TEG_THD POW_INT DIAG_ACT DIAG_END	1	[-]
Indicates current phase of oxygen sensor heater OBDII diagnosis					
TEMP_DIF_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0...7FFH	0...2047	1	[K]
Indicates difference in temperature between set operating temperature & exhaust gas at sensor location					
POW_INT_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[J]
Integral indicates measure of cooling energy of exhaust gas at sensor location					
TCC_VLD_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0...3FFH	0...1023	1	[-]
Number of completed valid diagnosis cycles					
TCC_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0...3FFH	0...1023	1	[-]
Number of faults detected during valid diagnosis cycles					
CTR_CYCNR_R_IT_OBD_LS_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[-]
Value of handshake counter for last valid resistance value used for diagnosis					
R_IT_THD_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O/S	0...FFFFH	0...65535	1	[Ohm]
Threshold for most current downstream oxygen sensor internal resistance used for diagnosis cycle, Mode 6 information					
R_IT_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O/S	0...FFFFH	0...65535	1	[Ohm]
Most current downstream oxygen sensor internal resistance used for diagnosis cycle, Mode 6 information					
T_DLY_OBD_LSH_DOWN_TMP	-	0...FFH	0...255	1	[s]
Minimum delay time prior to checking status of OBDI heater / signal faults					
POW_INT_MIN_OBD_LSH_DOWN_TMP	-	0...FFFFH	0...65535	1	[J]
Minimum required cooling energy required at downstream sensor prior to checking Ri threshold					
TCC_THD_OBD_LSH_DOWN_TMP	-	0...3FFH	0...1023	1	[-]
Number of valid downstream LSH OBDII diagnosis cycles after which statistic evaluation is carried out					
TCC_THD_ERR_OBD_LSH_DOWN_TMP	-	0...3FFH	0...1023	1	[-]
Fault detection threshold for determination of defective heater					
R_IT_L_MES_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[Ohm]
Latest measured downstream oxygen sensor internal resistance value for no error case, mode 6 information					

R_IT_L_REF_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[Ohm]
Latest reference value for measured internal resistance downstream in error case, mode 6 information					
R_IT_H_MES_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[Ohm]
Latest measured downstream oxygen sensor internal resistance value for error case, mode 6 information					
R_IT_H_REF_OBD_LSH_DOWN[NC_CBK_EX_NR]	V	0...FFFFH	0...65535	1	[Ohm]
Latest reference value for measured internal resistance downstream in error case, mode 6 information					

Input data:

LV_ST_END	R_IT_LS_DOWN[NC_CBK_EX_NR]	STATE_LSH_DOWN[NC_CBK_EX_NR]	LV_INH_DIAG_OBD_LSH_DOWN[NC_CBK_EX_NR]
MAF_KGH	LV_LS_DOWN_READY[NC_CBK_EX_NR]	LV_VB_CDN_OBD_2	LSHPWM_DOWN[NC_CBK_EX_NR]
LV_IGK	LV_ERR_EL_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	CTR_CYCNR_R_IT_LS_DOWN_VLD[NC_CBK_EX_NR]
R_IT_MDL_LS_DOWN_NEW[NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_DIAG_EOL_REQ_OBD_LSH_DOWN[NC_CBK_EX_NR]	TEG_DYN_LS_DOWN[NC_CBK_EX_NR]
LV_DIAG_EOL_END_LS_UP_DOWN[NC_CBK_EX_NR]	NC_STATE_LSL_UP_IF	LV_ERR_LS_DOWN[NC_CBK_EX_NR]	

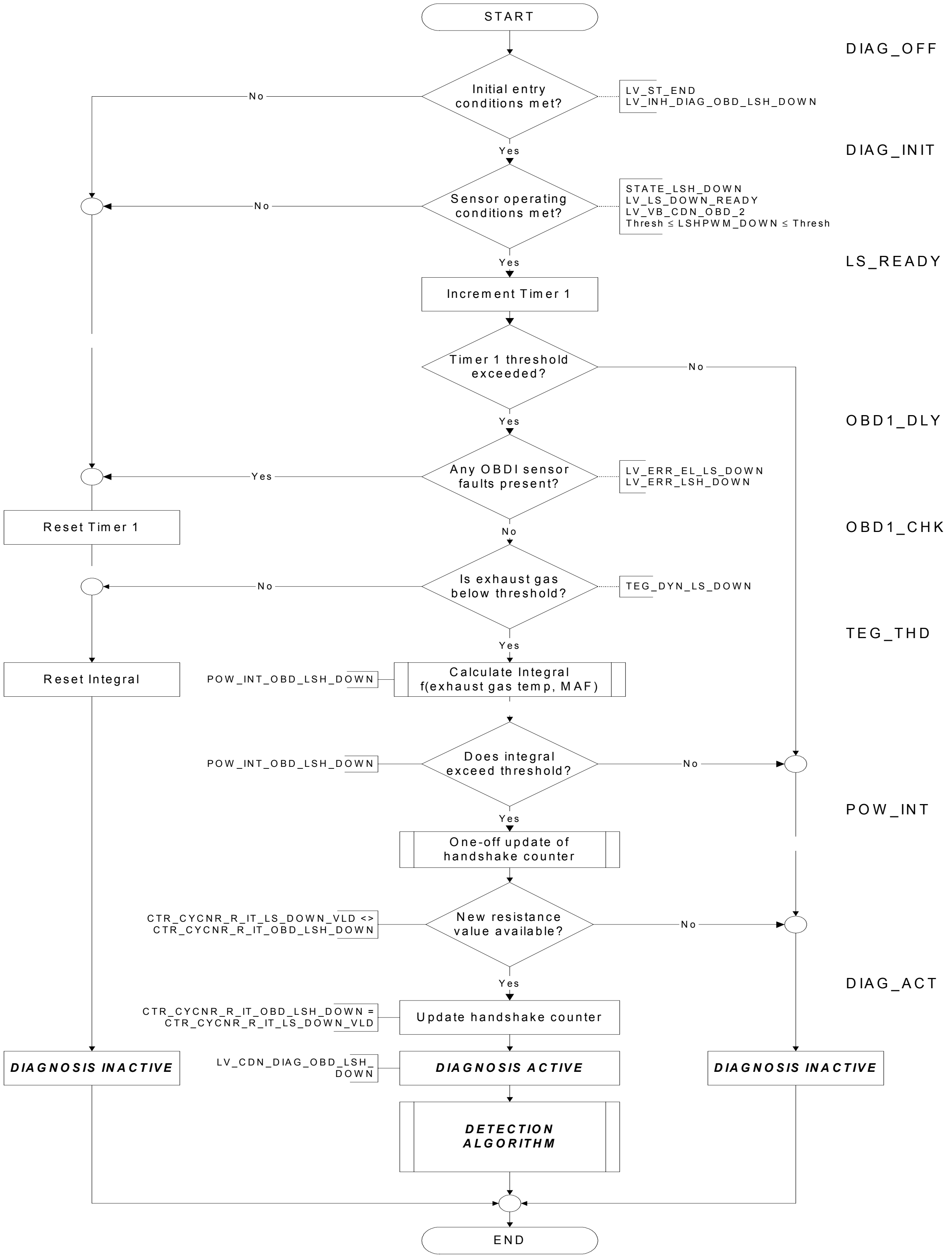
FUNCTION DESCRIPTION:

General information:

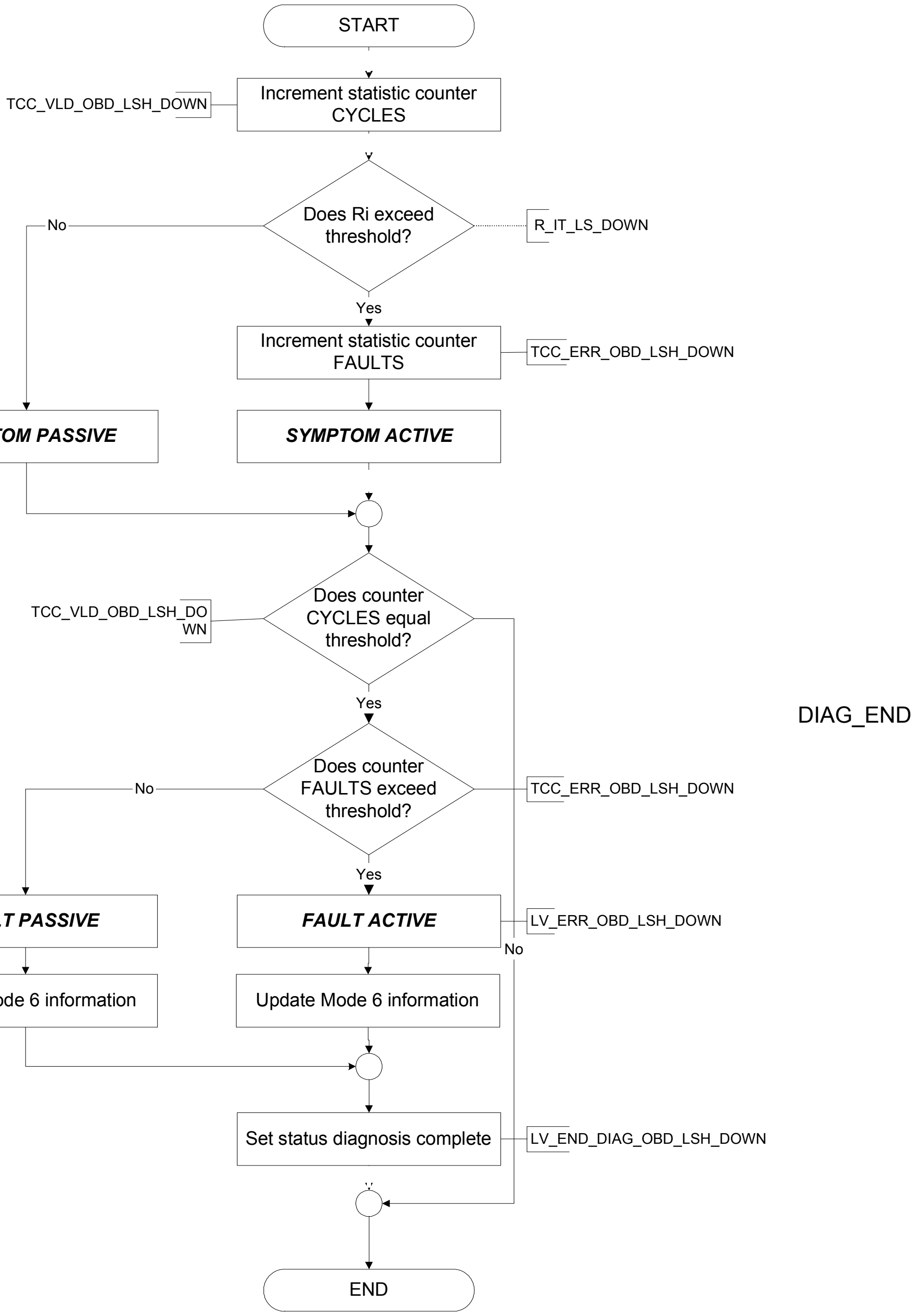
The calculation shall be done for all exhaust cylinder banks.
For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Signal flow diagram:

Activation conditions:



Detection Algorithm:



Description:

The downstream oxygen sensor heater circuit shall detect any loss in heater power that would cause a drop in the sensor operating temperature, thereby possibly causing exhaust gas emissions to rise above the applicable standards or prevent the sensor signal from being used as a diagnostic system monitoring device.

Losses in heater power may occur due to, for example, ageing of the heater element, defective wiring, increased heater cct. connector contact resistance, defective heater driver etc.

The hereafter mentioned diagnosis strategy is based on the comparison of the oxygen sensor internal resistance to an absolute threshold during operating conditions where the exhaust gas temperature has been determined to be sufficiently low as to cause the sensor ceramic temperature to fall outside normal operating levels, in cases where a power is insufficient.

The functionality of the diagnosis may be described in further detail below:

The initial diagnosis application condition; LV_ST_END shall ensure that the engine is running and has left the start phase but not yet entered the engine stop phase, LV_INH_DIAG_OBD_LSH_DOWN[i] shall ensure that the project specific application conditions (Application Incidences) have been met and LV_END_DIAG_OBD_LSH_DOWN[i] shall ensure that the diagnosis only be carried out once per driving cycle.

In order to ensure that the sensor has reached its normal operating temperature, the heater management state STATE_LSH_DOWN[i] shall be determined to be in the open loop control state LSH_POW_CTL, the sensor shall be determined to be in a state of operative readiness via LV_LS_DOWN_READY[i], the battery voltage VB shall be in a range determined to be fit for carrying out OBD2 diagnosis shown by LV_VB_CDN_OBD_2 and the heater duty-cycle LSHPWM_DOWN[i] shall be determined to be within calibrateable range C_LSHPWM_MIN_LSH_DOWN & C_LSHPWM_MAX_LSH_DOWN.

Should the above initial and operative conditions have been determined to be met, a timer, denoted herein as *TIMER_1*, shall be started. The timer shall run until the threshold T_DLY_OBD_LSH_DOWN_TMP is equalled or exceeded. Once this delay has passed, the status of the OBDI error bits LV_ERR_EL_LS_DOWN[i] & LV_ERR_LSH_DOWN[i] shall be read to ensure that no signal electrical or plausibility faults and no heater electrical faults are present in the sensor cct. under test. The delay shall ensure that sufficient time passes to permit completion of the OBDI monitoring prior to the start of OBDII monitoring.

Should no OBDI faults be present, the modelled exhaust gas temperature TEG_DYN_LS_DOWN[i] shall be compared to a calibrateable threshold C_TEG_DYN_MAX_OBD_LSH_DOWN. Due tolerances in the determination of the sensor internal resistance and in the case of a functional heater, the maximum resistance value may be given by a minimum tolerance sensor at a much lower temperature than for a maximum tolerance sensor at a higher temperature. Therefore, in the case of diagnosis based on an absolute internal resistance threshold, a certain temperature tolerance exists. The above mentioned temperature threshold shall lie below the minimum temperature determined from analysis of the internal resistance tolerances and be sufficient to cool the sensor ceramic element to below normal operating conditions, in the case of a heater fault.

Should the exhaust gas temperature be determined to equal or fall below the threshold, the monitoring strategy shall ensure that the sensor be cooled sufficiently long by the exhaust gas to enable differentiation between a functional and non-functional heater cct. (The time required is dependent on the thermal capacity of the system at the sensor location and the forced convection cooling of the sensor location). This shall be achieved by integrating the heat flux of the forced convection thermal transfer between sensor and exhaust gas, this being a measure for the cooling energy of the exhaust gas. The heat flux shall be calculated by determining the difference between the typical set sensor operating temperature C_TEMP_SP_OBD_LSH_DOWN and the modelled exhaust gas temperature TEG_DYN_LS_DOWN[i] and multiplying the result by the factor IP_FAC_MAF_OBD_LSH_DOWN representing the heat transfer coefficient and surface area concerned in the thermal transfer. As the heat transfer coefficient is a function of the velocity of the exhaust gas, the factor shall be mapped and dependent on the current mass airflow MAF_KGH.

The resultant integral POW_INT_OBD_LSH_DOWN[i] shall be compared to the threshold POW_INT_MIN_OBD_LSH_DOWN_TMP. Each time that the threshold be exceeded for the first time, e.g. first time after engine start or after the power integral has been reset, the CTR_CYCNR_R_IT_OBD_LS_DOWN[i] shall be updated once only with the value CTR_CYCNR_R_IT_LS_DOWN_VLD[i]. This shall ensure that only new internal resistance values be used for diagnosis purposes that have been determined since all the activation conditions have been met.

It shall be determined whether a new internal resistance has been determined by comparing the contents of counter CTR_CYCNR_R_IT_OBD_LS_DOWN[i] with that of CTR_CYCNR_R_IT_LS_DOWN_VLD[i]. Should the counters be unequal, then a new value is available and CTR_CYCNR_R_IT_OBD_LS_DOWN[i] shall be updated with CTR_CYCNR_R_IT_LS_DOWN_VLD[i]. If this conditions and all the above conditions be met, LV_CDN_DIAG_OBD_LSH_DOWN[i] shall be set and the diagnosis shall be considered to be active and the current monitoring cycle valid.

Once the current monitoring cycle has been determined to be valid, a counter TCC_VLD_OBD_LSH_DOWN[i] shall be incremented, denoting the number of valid monitoring cycles carried out to date. The internal resistance R_IT_LS_DOWN[i] shall be compared to a calibrateable threshold IP_R_IT_THD_OBD_LSH_DOWN according to modelled tip temperature of a new sensor (including safety margin). Should the resistance equal or exceed the threshold, a heater fault shall be determined to be present in the current cycle and a counter TCC_ERR_OBD_LSH_DOWN[i] shall be incremented, denoting the number of heater faults within the valid monitoring cycles carried out to date.

In order to comply with OBDII requirements, the current R_IT_LS_DOWN[i] value used shall be made available to an external tester. This information is Mode 6 information. Hence once the comparison to the threshold has been carried out, the latest measured internal resistance and threshold values (R_IT_L_MES_OBD_LSH_DOWN[i] and R_IT_L_REF_OBD_LSH_DOWN[i]) shall be copied to the respective Mode 6 interface variables R_IT_OBD_LS_DOWN[i] and R_IT_THD_OBD_LSH_DOWN[i] for the no error case. In the error case the variables R_IT_H_MES_OBD_LSH_DOWN[i] and R_IT_H_REF_OBD_LSH_DOWN[i] shall be handed out.

Once the number of valid monitoring cycles TCC_THD_OBD_LSH_DOWN_TMP has been determined to have been carried out, the diagnosis shall carry out a n-out-of-m statistical evaluation of the results. Should the number of determined heater faults exceed the threshold TCC_THD_ERR_OBD_LSH_DOWN_TMP, the oxygen sensor heater under test shall be considered to be faulty for the current driving cycle and flag LV_ERR_OBD_LSH_DOWN[i] shall be set.

If the number of faults does not exceed the threshold, LV_ERR_OBD_LSH_DOWN[i] shall be reset.

Once the statistical evaluation has been carried out, no further monitoring cycles shall be executed in the remaining driving cycle. This is achieved by setting LV_END_DIAG_OBD_LSH_DOWN[i].

Should, according to project philosophy, the diagnosis be activated for an end of line (EOL) test, specific threshold conditions for the diagnosis delay timer, the power integral, the required number of test and error indicating cycles may be chosen to shorten the time until recognition of diagnosis end.

NOTES:

A. The integral POW_INT_OBD_LSH_DOWN[i] shall be reset to 0 should the exhaust gas temperature exceed the applicable threshold. This would cause the sensor to be re-heated from the exhaust gas once again and a certain delay would be required prior to the sensor being cooled long enough to permit valid diagnosis to occur.

B. The timer *TIMER_1* and integral POW_INT_OBD_LSH_DOWN[i] shall be reset should any of the initial entry or operative readiness conditions no longer be met or an OBDI fault be detected in the sensor under test. This may cause the sensor to leave its normal operating conditions and the OBDII diagnosis shall be required to start from the beginning as the sensor operating conditions would require time to stabilise should the conditions be met again.

C. The statistic counters TCC_VLD_OBD_LSH_DOWN[i] and TCC_ERR_OBD_LSH_DOWN[i] shall <u>not</u> be reset by changing conditions and therefore shall only increment or remain frozen during any given monitoring cycle. The counters shall only be reset at the beginning of a new driving cycle.

D. Projects that provide for an internal statistical error management shall use their internal error management and apply this wherever reference to TCC_VLD_OBD_LSH_DOWN[i], TCC_ERR_OBD_LSH_DOWN[i], TCC_THD_OBD_LSH_DOWN_TMP & TCC_THD_ERR_OBD_LSH_DOWN_TMP is made.

E. The variable STATE_OBD_LSH_DOWN[i] shall indicate, at the end of the each monitoring cycle, the status of the diagnosis within that cycle. The following correlation of variable content to description shall apply:

0:	DIAG_OFF	Initialisation state; no conditions met
1:	DIAG_INIT	Initial conditions & application incidences met only
2:	LS_READY	Sensor operatively ready conditions met + 1,
3:	OBD1_DLY	OBDI monitoring completion delay exceeded + 1 & 2
4:	OBD1_CHK	No OBDI faults present +1, 2 & 3
5:	TEG_THD	Exhaust gas below threshold + 1, 2, 3 & 4
6:	POW_INT	Power integral exceeded + 1,2,3,4 & 5
7:	DIAG_ACT	***Diagnosis detection active***
8:	DIAG_END	***Diagnosis detection complete***

<u>Application conditions:</u>

Initialisation:

The following variables shall be initialised at reset, at the transition ignition key on (i.e. LV_IGK = 0 -> 1) and at clearing the error memory:

LV_CDN_DIAG_OBD_LSH_DOWN[i] = 0
ERR_SYM_OBD_LSH_DOWN[i] = 0
LV_ERR_OBD_LSH_DOWN[i] = 0
LV_END_DIAG_OBD_LSH_DOWN[i] = 0
STATE_OBD_LSH_DOWN[i] = 0
TEMP_DIF_OBD_LSH_DOWN[i] = 0
POW_INT_OBD_LSH_DOWN[i] = 0
TCC_VLD_OBD_LSH_DOWN[i] = 0
TCC_ERR_OBD_LSH_DOWN[i] = 0
CTR_CYCNR_R_IT_OBD_LS_DOWN[i] = 0
R_IT_L_MES_OBD_LSH_DOWN[i] = 0
R_IT_L_REF_OBD_LSH_DOWN[i] = 0
R_IT_H_MES_OBD_LSH_DOWN[i] = 0
R_IT_H_REF_OBD_LSH_DOWN[i] = 0

Reset *TIMER_1*

R_IT_OBD_LSH_DOWN[i] and R_IT_THD_OBD_LSH_DOWN[i] shall be initialised with the last value from the previous driving cycle stored in non-volatile memory after RESET.

NOTE: TIMER_1 refers to a program specific timer and is not intended to represent a variable name!

Recurrence:

The OBDII oxygen sensor heater diagnosis shall be carried out once per driving cycle. The diagnosis shall require a number of monitoring cycles to be carried out to permit statistical evaluation of the results. These monitoring cycles shall occur every 1 s, until a calibrateable number of valid diagnosis cycles has been exceeded.

Once this threshold has been completed, the OBDII oxygen sensor heater diagnosis shall no longer be carried out until a new driving cycle is initiated.

Should an OBDII heater fault be detected, the application assistances (Functions that are defined to be affected by the state of the oxygen sensor signal) shall remain active for the remainder of the driving cycle. The restarting of the affected functions during the next driving cycle shall be carried out according to the project philosophy.

Activation:

```
If  (LV_ST_END = 1) &
    (LV_INH_DIAG_OBD_LSH_DOWN[i] = 0)
then If   LV_DIAG_EOL_REQ_OBD_LSH_DOWN[i] = 0
     then T_DLY_OBD_LSH_DOWN_TMP = C_T_DLY_OBD_LSH_DOWN
          POW_INT_MIN_OBD_LSH_DOWN_TMP =
                                  C_POW_INT_MIN_OBD_LSH_DOWN
          TCC_THD_OBD_LSH_DOWN_TMP = C_TCC_THD_OBD_LSH_DOWN
          TCC_THD_ERR_OBD_LSH_DOWN_TMP =
                                  C_TCC_THD_ERR_OBD_LSH_DOWN
     else T_DLY_OBD_LSH_DOWN_TMP = C_T_DLY_OBD_LSH_DOWN_EOL
          POW_INT_MIN_OBD_LSH_DOWN_TMP =
                                  C_POW_INT_MIN_OBD_LSH_DOWN_EOL
          TCC_THD_OBD_LSH_DOWN_TMP = C_TCC_THD_OBD_LSH_DOWN_EOL
          TCC_THD_ERR_OBD_LSH_DOWN_TMP =
                                  C_TCC_THD_ERR_OBD_LSH_DOWN_EOL
     endif
     If   (LV_LS_DOWN_READY[i] = 1) &
          (STATE_LSH_DOWN[i] = LSH_POW_CTL) &
          (LV_VB_CDN_OBD_2 = 1) &
          (C_LSHPWM_MIN_LSH_DOWN ≤ LSHPWM_DOWN[i]) &
          (LSHPWM_DOWN[i] ≤ C_LSHPWM_MAX_LSH_DOWN)
     then Increment TIMER_1
          If  (TIMER_1 ≥ T_DLY_OBD_LSH_DOWN_TMP)
          then If  (LV_ERR_EL_LS_DOWN[i] = 0) &
                   (LV_ERR_LSH_DOWN[i] = 0)
```

```
            then If  (TEG_DYN_LS_DOWN[i] ≤
                     C_TEG_DYN_MAX_OBD_LSH_DOWN)
                then Calculate POW_INT_OBD_LSH_DOWN[i]
                     If (POW_INT_OBD_LSH_DOWN[i] ≥
                        POW_INT_MIN_OBD_LSH_DOWN_TMP)
                     then One-off update CTR_CYCNR_R_IT_OBD_LS_DOWN[i]
                          % see note below
                          If  (CTR_CYCNR_R_IT_LS_DOWN_VLD[i] <>
                              CTR_CYCNR_R_IT_OBD_LS_DOWN[i])
                          then LV_CDN_DIAG_OBD_LSH_DOWN[i] = 1
                               CTR_CYCNR_R_IT_OBD_LS_DOWN[i] =
                               CTR_CYCNR_R_IT_LS_DOWN_VLD[i]
                               „Diagnosis detection active“
                          else STATE_OBD_LSH_DOWN[i] = POW_INT
                               LV_CDN_DIAG_OBD_LSH_DOWN[i] = 0
                          endif
                     else STATE_OBD_LSH_DOWN[i] = TEG_THD
                          LV_CDN_DIAG_OBD_LSH_DOWN[i] = 0
                     endif
                else STATE_OBD_LSH_DOWN[i] = OBD1_CHK
                     Reset POW_INT_OBD_LSH_DOWN[i]
                     LV_CDN_DIAG_OBD_LSH_DOWN[i] = 0
                endif
            else STATE_OBD_LSH_DOWN[i] = OBD1_DLY
                 Reset TIMER_1
                 Reset POW_INT_OBD_LSH_DOWN[i]
                 LV_CDN_DIAG_OBD_LSH_DOWN[i] = 0
            endif
        else STATE_OBD_LSH_DOWN[i] = LS_READY
             LV_CDN_DIAG_OBD_LSH_DOWN[i] = 0
        endif
    else STATE_OBD_LSH_DOWN[i] = DIAG_INIT
         Reset TIMER_1
         Reset POW_INT_OBD_LSH_DOWN[i]
         LV_CDN_DIAG_OBD_LSH_DOWN[i] = 0
    endif
else STATE_OBD_LSH_DOWN[i] = DIAG_OFF
     Reset TIMER_1
     Reset POW_INT_OBD_LSH_DOWN[i]
     LV_CDN_DIAG_OBD_LSH_DOWN[i] = 0
endif.
```

Deactivation:

```
If  (LV_END_DIAG_OBD_LSH_DOWN[i] = 1)
then  „Diagnosis detection complete“
endif.
```

NOTES: The state variable STATE_OBD_LSH_DOWN[i] *shall only be validated at the completion of the current monitoring cycle, i.e. the state should not vary within the current 1 s monitoring cycle, but indicate at the end of the current cycle , the last state reached, dependant upon the branches.*

The variable CTR_CYCNR_R_IT_OBD_LS_DOWN[i] shall be updated with the current value of the input variable CTR_CYCNR_R_IT_LS_DOWN_VLD[i] on each occasion that the integral POW_INT_OBD_LSH_DOWN[i] exceeds the threshold for the first time. For further function calls where the integral condition is met, the one-off update shall not be carried out. This shall continue until the integral condition is no longer met.

Formula section:

Calculation of cooling energy integral:

Determine difference between typical sensor operating temperature and exhaust gas temperature:

TEMP_DIF_OBD_LSH_DOWN[i] = (C_TEMP_SP_OBD_LSH_DOWN - TEG_DYN_LS_DOWN[i])

(NOTE: Resolution of temperature difference is reduced from 0.0625 °C, resolution of the individual temperature variables, to 1 °C; The difference shall be limited to provide only positive values, i.e. ≥ 0)

The integral of heat flux (in Watts) provides a measure for the total cooling energy of the exhaust gas (in Joules):

POW_INT_OBD_LSH_DOWN[i]$_{(n)}$ = (POW_INT_OBD_LSH_DOWN[i]$_{(n-1)}$ + TEMP_DIF_OBD_LSH_DOWN[i] * IP_FAC_MAF_OBD_LSH_DOWN)

(NOTE: Integral shall be limited to its maximum hex limit, i.e. FFFFH or 65535 digits, to prevent overflow.)

Detection: (Assumes "Diagnosis detection active") **See also NOTE D above!**

```
Increment TCC_VLD_OBD_LSH_DOWN[i]
If  (R_IT_LS_DOWN[i] ≥ IP_R_IT_THD_OBD_LSH_DOWN)
then  Symptom "Oxygen sensor heater fault, bank i" active
      ERR_SYM_OBD_LSH_DOWN[i] = "SYM_0"
      R_IT_H_MES_OBD_LSH_DOWN[i] = R_IT_LS_DOWN[i]
      R_IT_H_REF_OBD_LSH_DOWN[i] = IP_R_IT_THD_OBD_LSH_DOWN
      Increment TCC_ERR_OBD_LSH_DOWN[i]
else  Symptom "Oxygen sensor heater fault, bank i" passive
      ERR_SYM_OBD_LSH_DOWN[i] = "NO_SYM"
      R_IT_L_MES_OBD_LSH_DOWN[i] = R_IT_LS_DOWN[i]
      R_IT_L_REF_OBD_LSH_DOWN[i] = IP_R_IT_THD_OBD_LSH_DOWN
endif.
If  (TCC_VLD_OBD_LSH_DOWN[i] ≥ TCC_THD_OBD_LSH_DOWN_TMP)
```

```
then STATE_OBD_LSH_DOWN[i] = DIAG_END
     If   (TCC_ERR_OBD_LSH_DOWN[i] ≥ TCC_THD_ERR_OBD_LSH_DOWN_TMP)
     then Symptom “Oxygen sensor heater fault, bank i” active
          LV_ERR_OBD_LSH_DOWN[i] = 1
          ERR_SYM_OBD_LSH_DOWN[i] = " SYM_0"
          R_IT_OBD_LSH_DOWN[i] = R_IT_H_MES_OBD_LSH_DOWN[i]
          R_IT_THD_OBD_LSH_DOWN[i] = R_IT_H_REF_OBD_LSH_DOWN[i]
     else Symptom “Oxygen sensor heater fault, bank i” passive
          LV_ERR_OBD_LSH_DOWN[i] = 0
          ERR_SYM_OBD_LSH_DOWN[i] = "NO_SYM"
          R_IT_OBD_LSH_DOWN[i] = R_IT_L_MES_OBD_LSH_DOWN[i]
          R_IT_THD_OBD_LSH_DOWN[i] = R_IT_L_REF_OBD_LSH_DOWN[i]
     endif
else STATE_OBD_LSH_DOWN[i] = DIAG_ACT
endif.
```

The diagnosis end bit shall be calculated each recurrence independently on STATE_OBD_LSH_DOWN[i] until diagnosis deactivation:

```
If      STATE_OBD_LSH_DOWN[i] = DIAG_END
or ...  #If     NC_STATE_LSL_UP_IF > 0
        #then   ... (LV_DIAG_EOL_END_LS_UP_DOWN[i] = 1
                and  LC_DIAG_EOL_END_LSH_DOWN[i] = 1
                and  LV_ERR_LS_DOWN[i] = 0)
        #endif
then    LV_END_DIAG_OBD_LSH_DOWN[i] = 1
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LSHPWM_MIN_LSH_DOWN	1	0...FFH	0...99.60937	0.390625	[%]
Minimum permitted heater effective voltage PWM for downstream LSH OBDII diagnosis					
C_LSHPWM_MAX_LSH_DOWN	1	0...FFH	0...99.60937	0.390625	[%]
Maximum permitted heater effective voltage PWM for downstream LSH OBDII diagnosis					
C_T_DLY_OBD_LSH_DOWN	1	0...FFH	0...255	1	[s]
Minimum delay prior to checking status of OBDI heater / signal faults					
C_T_DLY_OBD_LSH_DOWN_EOL	1	0...FFH	0...255	1	[s]
Minimum delay prior to checking status of OBDI heater / signal faults when EOL test required					
C_TEG_DYN_MAX_OBD_LSH_DOWN	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Temperature threshold under which downstream LSH OBDII diagnosis activated					
C_TEMP_SP_OBD_LSH_DOWN	1	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Set temperature for downstream LSH operation used to generate power integral					
C_POW_INT_MIN_OBD_LSH_DOWN	1	0...FFFFH	0...65535	1	[J]
Measure of minimum required cooling energy required at downstream sensor prior to checking Ri threshold					
C_POW_INT_MIN_OBD_LSH_DOWN_EOL	1	0...FFFFH	0...65535	1	[J]
Measure of minimum required cooling energy required at downstream sensor prior to checking Ri threshold when EOL test required					
IP_R_IT_THD_OBD_LSH_DOWN	4	0...FFFFH	0...65535	1	[Ohm]
LDP_R_IT_LS_DOWN_NEW_LSH_DOWN	4	0...FFFFH	0...65535	1	[Ohm]
Internal resistance threshold, when exceeded, heater is determined to be defective for current diagnosis cycle; including reference to modelled value for new sensor					
IP_FAC_MAF_OBD_LSH_DOWN	6	0...FFH	0...15.9375	0.0625	[J/K]
LDPM_MAF_KGH_3_EGCP	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Factor reflecting forced convection influence of cooling area, gas velocity and heat transfer coefficient					
C_TCC_THD_OBD_LSH_DOWN	1	0...3FFH	0...1023	1	[-]
Number of valid downstream LSH OBDII diagnosis cycles over which statistic evaluation to be carried out					
C_TCC_THD_OBD_LSH_DOWN_EOL	1	0...3FFH	0...1023	1	[-]
Number of valid downstream LSH OBDII diagnosis cycles over which statistic evaluation to be carried out when EOL test required					
C_TCC_THD_ERR_OBD_LSH_DOWN	1	0...3FFH	0...1023	1	[-]
Fault detection threshold, when exceeded, heater is determined to be defective during driving cycle					
C_TCC_THD_ERR_OBD_LSH_DOWN_EOL	1	0...3FFH	0...1023	1	[-]
Fault detection threshold, when exceeded, heater is determined to be defective during driving cycle when EOL test required					
LC_DIAG_EOL_END_LSH_DOWN[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Locigal calibration data to enforce end of diagnosis when EOL test has passed					

See also NOTE D above!

Configuration for diagnostic symptoms:

Diagnostic XX	Symptom description	Symptom	Filter type
OBD_LSH_DOWN[NC_CBK_EX_NR]	*Heater fault*	SYM_0	NO

16.70 Inhibition conditions for OBDII oxygen downstream heater diag.

16.70.1 1.1. Inhibition of diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Boolean inhibit flag for O2 sensor heater OBDII monitoring					
LV_INH_DIAG_RBM_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag to inhibit rate based monitoring for oxygen sensor heater OBDII diagnosis					
LV_DIAG_EOL_REQ_OBD_LSH_DOWN[NC_CBK_EX_NR]	O	0...1H	0...1	1	[-]
Logical variable indicating requested EOL test for downstream heater diagnosis					

Input data:

LV_IGK	LV_ST_END	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]
LV_VAR_LSH_DOWN	LV_CDN_VB_OBD2	LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]
NC_CBK_EX_NR		LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]
TCO		LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]
		LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]
		LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]
		LV_LS_DOWN_OBD_2_MAN_DEAC[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

LV_INH_DIAG_OBD_LSH_DOWN[NC_CBK_EX_NR] represents the influence by errors/conditions deactivating the O2 heater diagnosis. Additionally the interfaces for Rate Based Monitoring RBM are provided.

The calculation shall be done for all exhaust cylinder banks. For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
 i = 1 for exhaust cylinder bank 1
 i = 2 for exhaust cylinder bank 2
otherwise (NC_CBK_EX_NR = 1)
 i = 1 for single exhaust cylinder bank.

Application conditions:

Initialisation: 0 at LV_IGK = 0→1 or at reset

Recurrence: 1 s

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

```
If         LV_ERR_LSH_DOWN[i] = 1
     or    LV_ERR_SCP_LS_DOWN[i] = 1
     or    LV_ERR_SCG_LS_DOWN[i] = 1
     or    LV_ERR_OC_LS_DOWN[i] = 1
     or    LV_ERR_PUE_LS_DOWN[i] = 1
     or    LV_ERR_CHK_LS_DOWN[i] = 1

      then LV_INH_DIAG_RBM_OBD_LSH_DOWN[i]  =  1
  else     LV_INH_DIAG_RBM_OBD_LSH_DOWN[i]  =  0
endif

If   LV_VAR_LSH_DOWN = 0
     or    LV_CDN_VB_OBD2 = 0
     or    TCO < C_TCO_MIN_OBD2_LSH_DOWN
     or    LV_INH_DIAG_RBM_OBD_LSH_DOWN[i] = 1
     or    LV_LS_DOWN_OBD_2_MAN_DEAC[i] = 1
  then     LV_INH_DIAG_OBD_LSH_DOWN[i]  =  1
  else     LV_INH_DIAG_OBD_LSH_DOWN[i]  =  0
endif

LV_DIAG_EOL_REQ_OBD_LSH_DOWN[i] = 0
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_MIN_OBD2_LSH_DOWN	1	0...FEH	-48...142.5	0.75	[°C]
TCO threshold for OBD2 heater diagnosis downstream					

16.70.2 1.2. Interface for Rate Based Monitoring

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
STATE_RBM_OBD_LSH_DOWN[NC_CBK_EX_NR]	V/O	0...FFH	0...255	1	[-]
Interface of OBD_LSH_DOWN monitor with the Rate-Based Monitoring statistics					

Input data:

LV_DC	LV_END_DIAG_OBD_LSH_DOWN[NC_CBK_EX_NR]
CTR_ERR_DYN_NR	

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the OBD_LSH_DOWN[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_OBD_LSH_DOWN[i] data.

Within STATE_RBM_OBD_LSH_DOWN[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

Application conditions:

Initialisation :

at ECU reset : bit 0, bit 1 and bit 2 of STATE_RBM_OBD_LSH_DOWN[i] = 0

at LV_DC 0 → 1 transition : bit 0 and bit 1 of STATE_RBM_OBD_LSH_DOWN[i] = 0

on failure memory reset : bit 1 of STATE_RBM_OBD_LSH_DOWN[i] = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

bit 2 of STATE_RBM_OBD_LSH_DOWN[i] = 1

The pending status of the following failures has to be checked only once:

LV_ERR_LSH_DOWN[NC_CBK_EX_NR]
LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_PUE_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_CHK_LS_DOWN[NC_CBK_EX_NR]

```
If(1)     { CPU optimization at LV_DC 0 → 1 transition }
          CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
          While  bit 1 of STATE_RBM_OBD_LSH_DOWN[i] = 0 do
                 ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                 SYNCHRONIZATION<CALL>)
                 with all XX failures of the above list.

                 If(2)        XX has a pending status
                 Then(2)
                              bit 1 of STATE_RBM_OBD_LSH_DOWN[i] = 1
                 Endif(2)
          Endwhile
Else(1)
          { the dynamic failure memory is empty }
          No action
Endif(1)
```

Every 1 s :

```
If        (bit 0 of STATE_RBM_OBD_LSH_DOWN[i] = 0     and
          LV_END_DIAG_OBD_LSH_DOWN[i] = 1)
Then      bit 0 of STATE_RBM_OBD_LSH_DOWN[i] = 1
Endif


If        (bit 1 of STATE_RBM_OBD_LSH_DOWN[i] = 0     and
          LV_INH_DIAG_RBM_OBD_LSH_DOWN[i] = 1)
```

```
Then    bit 1 of STATE_RBM_OBD_LSH_DOWN[i] = 1
Endif
```

16.71 End of line test: Lambda sensors confused / wrong wired

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_VLS_EOL	V/O	0H	OFF	1	[-]
		1H	STEP_1		
		2H	STEP_2		
		3H	WAIT		
		4H	TIMEOUT		
		10H	FINISHED		
		11H	ERR_UP		
		12H	ERR_DOWN		
		13H	ERR_UP_DOWN		
		14H	ERR_UP_1		
		15H	ERR_UP_2		
		16H	ERR_DOWN_1		
		17H	ERR_DOWN_2		
		18H	NO_RESULT		
Status of lambda sensor EOL diagnos confused wired					
LAMB_SP_VLS_EOL[NC_CBK_EX_NR]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Bank selective setpoint (LS mix EOL test)					
LV_ACT_VLS_EOL	V/O	0...1H	0...1	1	[-]
Activation bit for lambda sensor EOL diagnos confused wired					
MAF_INT_MIN_VLS_EOL	V/O	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral for lambda sensor EOL diagnos confused wired					
VLS_DOWN_AFR_EOL[NC_CBK_EX_NR]	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Frozen VLS value for lambda sensor EOL diagnos confused wired					
VLS_DOWN_AFL_EOL[NC_CBK_EX_NR]	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Frozen VLS value for lambda sensor EOL diagnos confused wired					
LAMB_LS_UP_AFR_EOL[NC_CBK_EX_NR]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Frozen Lambda value for lambda sensor EOL diagnos confused wired					
LAMB_LS_UP_AFL_EOL[NC_CBK_EX_NR]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Frozen Lambda value for lambda sensor EOL diagnos confused wired					
LV_VLS_EOL_SHIFT_REP_INH	V	0...1H	0...1	1	[-]
Bit to prevent endless loop of lambda shifts					
T_WAIT_VLS_EOL_OPM_SWI	V	0...FFFFH	0...1310.7	0.02	[s]
Time waiting for operation mode switch					

Input data:

LAMB_LS_UP[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]	NC_FAC_MAF_INT_20	LV_IGK
MAF_CYL	LV_ACT_VLS_EOL_EXT_ADJ	OPM_REQ_CUS	C_OPM_EXT_REQ_VLS_EOL

FUNCTION DESCRIPTION:

General information:

This EOL test is to detect confused or wrong wired lambda sensors in order to prevent bad emmissions for a two cylinder bank system, therefore:

i = 1 for cylinder bank 1 and i = 2 for cylinder bank 2

Description:

This EOL-test is activated by customer-tool by setting LV_ACT_VLS_EOL_EXT_ADJ = 1. At the same time a switch to the correct engine operation mode (e.g. homogenous combustion mode) is triggered in another software module. The function goes to the state WAIT until the correct combustion mode is reached (indicated by OPM_REQ_CUS = C_OPM_EXT_REQ_VLS_EOL). Meanwhile a timer T_WAIT_VLS_EOL_OPM_SWI is incremented. If this timer exceeds the threshold C_T_WAIT_VLS_EOL_OPM_SWI, the switch to the correct engine operation mode failed. So the function goes to state TIMEOUT and the test is stopped. The test can only be restarted, when a stop command is sent from tester (LV_ACT_VLS_EOL_EXT_ADJ = 0) and the start command is sent again (LV_ACT_VLS_EOL_EXT_ADJ = 1).

When the engine is in correct operation mode, STEP-1 is started with shifting the Lambda-setpoint (bank 2 lean / bank 1 rich) for C_MAF_INT_MIN_VLS_EOL. After reaching the MAF integral there is the first freezing of VLS values. After this STEP-2 is started inverse (bank 2 rich / bank 1 lean). After reaching the MAF integral there is the second freezing of VLS values.
In STEP-3 it is checked, if the frozen values are valid. If YES the EOL result is calculated (resuming with STEP_4). If NO (the frozen values of the downstream sensors are in an undefined band between rich and lean), STEP_1 and STEP_2 are started again, to validate these values. The setting and checking of the flag LV_VLS_EOL_SHIFT_REP_INH makes sure, that this validation is done only one time and the test can not enter an endless loop of STEP_1 and STEP_2.

After this the EOL result STATE_VLS_EOL is calculated finally (STEP-4), by comparing the frozen values with calibratable thresholds.
The idea is, that at rich lambda setpoint LAMB_LS_UP and VLS_DOWN of the corresponding exhaust bank must show "rich" values. Analogous the same at lean lambda setpoint. If the frozen values are confused for bank 1 and 2 (lean value at rich setpoint and vice versa), then the corresponding sensors (upstream and/or downstream) are confused wired. If the frozen values do not cross the thresholds, the corresponding sensor signal is stucking.

The test is also used for failure analysis by the mechanics at the garage in the field.

Application conditions:

Initialisation: LAMB_SP_VLS_EOL[i] = 1 **and** remaining outputs = 0 at

initialization routine of start EOL test **or**

ECU reset **or**

Clearing error memory **or**

LV_IGK 0->1

Recurrence: 20ms

Activation:

If LV_ACT_VLS_EOL_EXT_ADJ 0 --> 1

Then Start function with WAIT

Else STATE_VLS_EOL = OFF

```
            Endif
Deactivation:     LV_ACT_VLS_EOL_EXT_ADJ = 0
      Action at deactivation: set LAMB_SP_VLS_EOL[i] = 1 and remaining outputs = 0
```

Formula section:

```
WAIT
IF      OPM_REQ_CUS = C_OPM_EXT_REQ_VLS_EOL
THEN    Continue with STEP_1
ELSE    IF    T_WAIT_VLS_EOL_OPM_SWI > C_T_WAIT_VLS_EOL_OPM_SWI
        THEN  STATE_VLS_EOL = TIMEOUT
              Stop calculation and deactivate function without reset of all variables
        ELSE  increment timer T_WAIT_VLS_EOL_OPM_SWI
        ENDIF
ENDIF
```

```
STEP-1, First bankwise lambdasetpoint shifting
STATE_VLS_EOL = STEP_1
Start calculation of MAF_INT_MIN_VLS_EOL
MAF_INT_MIN_VLS_EOL = MAF_INT_MIN_VLS_EOL + MAF_CYL*NC_FAC_MAF_INT_20
LV_ACT_VLS_EOL = 1                              for lambda coordination
LAMB_SP_VLS_EOL[1] = C_LAMB_SP_VLS_EOL_AFR      bank1 = rich
LAMB_SP_VLS_EOL[2] = C_LAMB_SP_VLS_EOL_AFL      bank2 = lean
If      MAF_INT_MIN_VLS_EOL >= C_MAF_INT_MIN_VLS_EOL
Then    LAMB_LS_UP_AFR_EOL[1] = LAMB_LS_UP[1]
        LAMB_LS_UP_AFL_EOL[2] = LAMB_LS_UP[2]
        VLS_DOWN_AFR_EOL[1] = VLS_DOWN[1]
        VLS_DOWN_AFL_EOL[2] = VLS_DOWN[2]
        LAMB_SP_VLS_EOL[1] = 1                  reset
        LAMB_SP_VLS_EOL[2] = 1                  reset
        MAF_INT_MIN_VLS_EOL = 0                 reset
        Continue with STEP-2
Else    Integration is still active
Endif
```

```
STEP-2, Second bankwise lambdasetpoint shifting
STATE_VLS_EOL = STEP_2
Start calculation of MAF_INT_MIN_VLS_EOL
MAF_INT_MIN_VLS_EOL = MAF_INT_MIN_VLS_EOL + MAF_CYL*NC_FAC_MAF_INT_20
```

```
LV_ACT_VLS_EOL = 1                                for lambda coordination
LAMB_SP_VLS_EOL[1] = C_LAMB_SP_VLS_EOL_AFL                    bank1 = lean
LAMB_SP_VLS_EOL[2] = C_LAMB_SP_VLS_EOL_AFR                    bank2 = rich
If       MAF_INT_MIN_VLS_EOL >= C_MAF_INT_MIN_VLS_EOL
Then     LAMB_LS_UP_AFL_EOL[1] = LAMB_LS_UP[1]
         LAMB_LS_UP_AFR_EOL[2] = LAMB_LS_UP[2]
         VLS_DOWN_AFL_EOL[1] = VLS_DOWN[1]
         VLS_DOWN_AFR_EOL[2] = VLS_DOWN[2]
         LAMB_SP_VLS_EOL[1] = 1                     reset
         LAMB_SP_VLS_EOL[2] = 1                     reset
         MAF_INT_MIN_VLS_EOL = 0                    reset
         Continue with STEP-3, "Validation"
Else     Integration is still active
Endif
```

STEP-3, Validation of froozen VLS values

```
If       (C_VLS_AFL_EOL < VLS_DOWN_AFR_EOL[1] < C_VLS_AFR_EOL    or
         C_VLS_AFL_EOL < VLS_DOWN_AFR_EOL[2] < C_VLS_AFR_EOL     or
         C_VLS_AFL_EOL < VLS_DOWN_AFL_EOL[1] < C_VLS_AFR_EOL     or
         C_VLS_AFL_EOL < VLS_DOWN_AFL_EOL[2] < C_VLS_AFR_EOL)    and
         LV_VLS_EOL_SHIFT_REP_INH = 0
Then     Repeat  STEP-1 for the second cycle
         LV_VLS_EOL_SHIFT_REP_INH = 1
Else     Continue with STEP-4 "Calculation"
         LV_ACT_VLS_EOL = 0                 reset
Endif
```

STEP-4, Calculation of test results STATE_VLS_EOL

Remark: After the first error status is detected the calculation is stopped

EOL test finished, all sensors O.K.

```
If       LAMB_LS_UP_AFR_EOL[1] < C_LAMB_LS_UP_AFR_EOL             and
         LAMB_LS_UP_AFR_EOL[2] < C_LAMB_LS_UP_AFR_EOL             and
         VLS_DOWN_AFR_EOL[1]   > C_VLS_AFR_EOL          and
         VLS_DOWN_AFR_EOL[2]   > C_VLS_AFR_EOL          and
         LAMB_LS_UP_AFL_EOL[1] > C_LAMB_LS_UP_AFL_EOL        and
         LAMB_LS_UP_AFL_EOL[2] > C_LAMB_LS_UP_AFL_EOL        and
```

```
          VLS_DOWN_AFL_EOL[1]    < C_VLS_AFL_EOL              and
          VLS_DOWN_AFL_EOL[2]    < C_VLS_AFL_EOL
Then      STATE_VLS_EOL = FINISHED
```

EOL test finished, UP sensors confused

```
Elseif    LAMB_LS_UP_AFR_EOL[1] > C_LAMB_LS_UP_AFR_EOL        and
          LAMB_LS_UP_AFR_EOL[2] > C_LAMB_LS_UP_AFR_EOL        and
          VLS_DOWN_AFR_EOL[1]    > C_VLS_AFR_EOL              and
          VLS_DOWN_AFR_EOL[2]    > C_VLS_AFR_EOL              and
          LAMB_LS_UP_AFL_EOL[1] < C_LAMB_LS_UP_AFL_EOL        and
          LAMB_LS_UP_AFL_EOL[2] < C_LAMB_LS_UP_AFL_EOL        and
          VLS_DOWN_AFL_EOL[1]    < C_VLS_AFL_EOL              and
          VLS_DOWN_AFL_EOL[2]    < C_VLS_AFL_EOL
Then      STATE_VLS_EOL = ERR_UP
```

EOL test finished, DOWN sensors confused

```
Elseif    LAMB_LS_UP_AFR_EOL[1] < C_LAMB_LS_UP_AFR_EOL        and
          LAMB_LS_UP_AFR_EOL[2] < C_LAMB_LS_UP_AFR_EOL        and
          VLS_DOWN_AFR_EOL[1]    < C_VLS_AFR_EOL              and
          VLS_DOWN_AFR_EOL[2]    < C_VLS_AFR_EOL              and
          LAMB_LS_UP_AFL_EOL[1] > C_LAMB_LS_UP_AFL_EOL        and
          LAMB_LS_UP_AFL_EOL[2] > C_LAMB_LS_UP_AFL_EOL        and
          VLS_DOWN_AFL_EOL[1]    > C_VLS_AFL_EOL              and
          VLS_DOWN_AFL_EOL[2]    > C_VLS_AFL_EOL
Then      STATE_VLS_EOL = ERR_DOWN
```

EOL test finished, UP&DOWN sensors confused

```
Elseif    LAMB_LS_UP_AFR_EOL[1] > C_LAMB_LS_UP_AFR_EOL        and
          LAMB_LS_UP_AFR_EOL[2] > C_LAMB_LS_UP_AFR_EOL        and
          VLS_DOWN_AFR_EOL[1]    < C_VLS_AFR_EOL              and
          VLS_DOWN_AFR_EOL[2]    < C_VLS_AFR_EOL              and
          LAMB_LS_UP_AFL_EOL[1] < C_LAMB_LS_UP_AFL_EOL        and
          LAMB_LS_UP_AFL_EOL[2] < C_LAMB_LS_UP_AFL_EOL        and
          VLS_DOWN_AFL_EOL[1]    > C_VLS_AFL_EOL          and
          VLS_DOWN_AFL_EOL[2]    > C_VLS_AFL_EOL
Then      STATE_VLS_EOL = ERR_UP_DOWN
```

```
EOL test finished, UP1 not plausible
Elseif    LAMB_LS_UP_AFR_EOL[1] > C_LAMB_LS_UP_AFR_EOL        or
          LAMB_LS_UP_AFL_EOL[1] < C_LAMB_LS_UP_AFL_EOL
Then      STATE_VLS_EOL = ERR_UP_1

EOL test finished, UP2  not plausible
Elseif    LAMB_LS_UP_AFR_EOL[2] > C_LAMB_LS_UP_AFR_EOL        or
          LAMB_LS_UP_AFL_EOL[2] < C_LAMB_LS_UP_AFL_EOL
Then      STATE_VLS_EOL = ERR_UP_2

EOL test finished, DOWN1 not plausible
Elseif    VLS_DOWN_AFR_EOL[1]   < C_VLS_AFR_EOL      or
          VLS_DOWN_AFL_EOL[1]   > C_VLS_AFL_EOL
Then      STATE_VLS_EOL = ERR_DOWN_1

EOL test finished, DOWN2 not plausible
Elseif    VLS_DOWN_AFR_EOL[2]   < C_VLS_AFR_EOL      or
          VLS_DOWN_AFL_EOL[2]   > C_VLS_AFL_EOL
Then      STATE_VLS_EOL = ERR_DOWN_2

EOL test finished, no result
Else      STATE_VLS_EOL = NO_RESULT

Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_MIN_VLS_EOL	1	0...FEH	-48...142.5	0.75	[°C]
TCO threshold for VLS EOL diagnosis					
C_TEG_CAT_DOWN_EOL	1	0...FFFFH	-48...1459.305	0.023	[°C]
TEG_CAT_DOWN threshold for VLS EOL diagnosis					
C_MAF_INT_MIN_VLS_EOL	1	0...FFFFH	0...1820.41666	0.0277778	[g]
MAF integral threshold for lambda sensor EOL diagnos confused wired					
C_LAMB_SP_VLS_EOL_AFR	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda setpoint AFR for lambda sensor EOL diagnos confused wired					
C_LAMB_SP_VLS_EOL_AFL	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda setpoint AFL for lambda sensor EOL diagnos confused wired					
C_VLS_AFR_EOL	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Rich VLS threshold for lambda sensor EOL diagnosis					
C_VLS_AFL_EOL	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lean VLS threshold for lambda sensor EOL diagnosis					
C_LAMB_LS_UP_AFR_EOL	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Rich Lambda threshold for lambda sensor EOL diagnosis					
C_LAMB_LS_UP_AFL_EOL	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lean Lambda threshold for lambda sensor EOL diagnosis					
C_T_WAIT_VLS_EOL_OPM_SWI	1	0...FFFFH	0...1310.7	0.02	[s]
Time threshold waiting for operation mode switch					

16.72 Acquisition of lambda sensor signal

16.72.1 O2 sensor (lin, up) SPI interface ASW – BSW

Initialisation of ATIC42 (= WRAF controller ASIC) status and configuration registers for SPI communication and control is basically handled within three modules: BSW (30202V01), ASW <-> BSW interface (403301), and corresponding Appl. Inc. (403B01). Configuration register for Ri measurement is initialised in 'O2 sensor (lin, up) element temperature determination' (402A01).

Certain calibration values depend on actual layout of generic ATIC42 HW module within target ECU and might be subject to change.

16.72.1.1 Calibration interfering functions

- Initialisation values for ATIC42 registers have to be chosen in accordance with calibration of 'O2 sensor (lin, up) operating strategy' (702Y01) regarding sensor warm-up behaviour, operation, and protection; chosen values have to comply with requirements and limits of specific sensor HW specification, ATIC42 HW module specification, and WRAF system operation guidelines.
- Cross dependencies / consistency with 'O2 sensor (lin, up) element temperature determination' (402A01), and 'O2 sensor (up) operability detection' (702O01) and have to be considered.

16.72.1.2 Calibration flowchart

There is no need to follow certain calibration sequence.

16.72.1.3 Calibration method

Supported SV ATIC42 WRAF System Configurations		
Configuration ID	**Combustion Principle**	**Sensor Type**
A	GS	NTK ZFAS-U1
B	GS	BOSCH LSU 4.9
C	DS	NTK ZFAS-U1
D	DS	BOSCH LSU 4.9

Parameter Source	
SS WRAF Sensor Specification	Fixed value, sensor HW specific.
OG WRAF System Operation Guidline	Fixed value, WRAF system (= sensor and controller module) specific.
CS WRAF Controller Specification + HW module layout	Fixed value, controller and HW module specific.
TC WRAF System Tolerance Calculation	Fixed value, based on WRAF system specifics.
TS Target System Specification	Project specific adaptation required.

Legend

- *All calibration values are described in decimal numbers if not stated otherwise!*
- *Physical representations (in brackets) are exemplary and depend on implemented WRAF system configuration.*
- *'do.' = [abbr.] ditto.*

C_STATE_LSL_IF_SPI_WR_INI	**Initialisation data of SPI register INIT_REG_1** ***Defines ATIC42 default behaviour after ECU RESET.***
Proposed Calibration Value	Unit
A: 192 B: do. C: do. D: do.	**[-]**
Source : OG / CS	

C_STATE_LSL_IF_CONF_SPI_WR_INI	**Initialisation data of SPI register INIT_REG_2** ***Defines ATIC42 default configuration after ECU RESET regarding total pumped reference current (25uA) and sensor protection time.***
Proposed Calibration Value	Unit
A: 68 B: do. C: do. D: do.	**[-]**
Source : OG / CS	

C_T_DLY_DI_LSL_IF	**Configuration of sensor protection time** ***Delay time for automatic sensor disconnection in case of electrical failure (2ms).***
Proposed Calibration Value	Unit
A: 2 B: do. C: do. D: do.	**[-]**
Source : TC / CS	

LC_CTR_ERR_LSL_IF_SPI_MAN_INC	**Switch to manually increase value of SPI communication error counter within BSW** ***Test switch for simulation of SPI communication error.***
Proposed Calibration Value	Unit
A: 0 B: do. C: do. D: do.	**[-]**

Source :	-

C_FRQ_R_IT_OSC_LSL_IF	**Coded information for Ri determination oscillator frequency** ***Oscillator frequency (3kHz) and activation switch for selective pin disabling during sensor protection time.***

Proposed Calibration Value								Unit
A:	72	B:	do.	C:	do.	D:	do.	**[-]**

Source : OG / CS

16.72.2 Downstream oxygen sensor (VLS_DOWN)

16.72.2.1 Calibration interfering functions

-

16.72.2.2 Calibration flowchart

Activation conditions

(LV_ST_END = 1& LV_ES = 0)

↓

Set

C_FAC_VLS_DOWN_MMV_HYS = 0.25
C_VLS_DOWN_MMV_HYS_MIN = 0.01999
C_VLS_DOWN_INI_MIN = 0.801
C_VLS_DOWN_INI_MAX = 0.098
C_CRLC_VLS_DOWN = 0.02
C_CRLC_VLS_DOWN_OFF = 0.002
C_T_DLY_PUC_END_VLS_DOWN = 2

↓

Initialisation check

VLS_DOWN_MMV_MIN[i] = C_VLS_DOWN_INI_MIN
VLS_DOWN_MMV_MAX[i] = C_VLS_DOWN_INI_MAX
VLS_DOWN_BOL[i] = C_VLS_DOWN_INI_MAX
VLS_DOWN_TOL[i] = C_VLS_DOWN_INI_MIN
VLS_DOWN_MMV_HYS[i] = C_VLS_DOWN_MMV_HYS_MIN
LV_VLS_DOWN_MMV_LIM[i] = 0
LV_VLS_DOWN_MMV_ACT = 0
VLS_DOWN_DRV1_MMV[i] = 0
VLS_DOWN_DRV1_MMV_MIN[i] = 0
VLS_DOWN_DRV1_ABS_MAX[i] = 0

↓

Functional check

16.72.2.3 Calibration method

C_VLS_DOWN_MMV_HYS_MIN	Minimum hysteresis for the computation of the thresholds VLS_DOWN_TOL[i] and VLS_DOWN_BOL[i]. Compute & limit switch hysteresis. The hysteresis shall also be limited so that it may not fall below the calibrateable threshold C_VLS_DOWN_MMV_HYS_MIN (**e.g. 0,02 V**).
C_VLS_DOWN_INI_MIN	Lower initialisation value for VLS_DOWN_TOL[i]. VLS_DOWN_MMV_MIN[i] The value of VLS_DOWN_MMV_MIN[i] (moving mean of the lean signal voltage) is initialised by C_VLS_DOWN_INI_MIN (**e.g. 0,8 V**)
C_VLS_DOWN_INI_MAX	Upper initialisation value for VLS_DOWN_BOL[i]. VLS_DOWN_MMV_MAX[i] The value of VLS_DOWN_MMV_MAX[i] (moving mean of the rich signal voltage) is initialised by C_VLS_DOWN_INI_MAX (**e.g. 0,1 V**)
C_CRLC_VLS_DOWN	Filter constant for determination of VLS_DOWN_MMV_MIN[i]/ VLS_DOWN_MMV_MAX[i] values when respective threshold passed (**e.g. 0,02**).
C_CRLC_VLS_DOWN_OFF	Filter constant for determination of VLS_DOWN_MMV_MIN[i]/ VLS_DOWN_MMV_MAX[i] values when respective threshold not passed (**e.g. 0,002**)
C_FAC_VLS_DOWN_MMV_HYS	Multiplicative factor governing amplitude of hysteresis (**e.g. 0,25**)
C_T_DLY_PUC_END_VLS_DOWN	Delay between completion of engine state PUC and computation of VLS_DOWN_MMV_MIN[i]/ VLS_DOWN_MMV_MAX[i] (**e.g. 2 s**)

16.73 Lambda Sensor Operation and Parameter Calculation

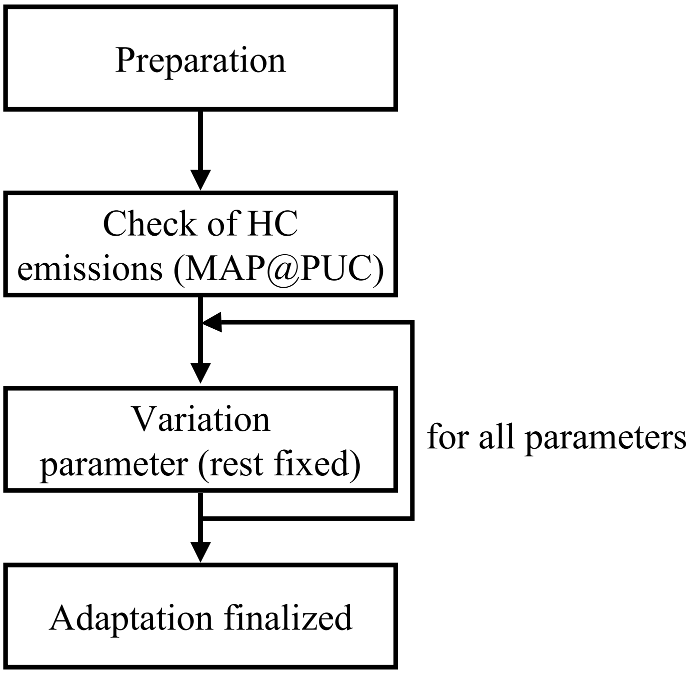
16.73.1 WRAF Sensor Gain Adaptation at Fuel Cut Off

16.73.1.1 Calibration interfering functions

- calculation of lambda value is affected in first order (part of offset adjustment - 702N)
- MAP limitation at PUC is handled at torque aggregate
- functionality's which have to be calibrated and running well in advance:
  - ❖ oxygen sensor temperature determination upstream
  - ❖ oxygen sensor heater management upstream
  - ❖ offset adjustment of oxygen sensor (linear control)
  - ❖ exhaust gas pressure calculation
- diagnosis's of oxygen sensor (heater and signal) are not mandatory to be activated (no direct impact for calibration)

16.73.1.2 Calibration flowchart

The sequence is also described by the order in calibration methods.



16.73.2 Calibration method

Preparation:

For preparation of the calibration all adaptation values have to be set to neutral values (=1), including the adaptation factor FAC_LS_GAIN_AD[i] = 1. All limiting thresholds should be set to minimum/maximum value to ensure non-impact.

general specification

The aim of gain adaptation calibration is to ensure best characteristic line of the linear oxygen sensor with focus on gain. Exerted influence on characteristic line by engine, environmental conditions and so on shall be reduced to a minimum. That increases accuracy and reliability.

C_TCO_MIN_LSL_GAIN_AD	<u>Description:</u> **Minimum coolant temperature to allow fuel cut Ip (pump current) determination** To ensure best and repeatable adaptation engine has to be at a warmed up status.
C_TAM_MAX_LSL_GAIN_AD	<u>Description:</u> **Maximum ambient temperature to allow fuel cut Ip (pump current) determination** To control impact of humidity (shift of characteristic due to absolute H_2O amount) the ambient temperature has to be limited for the adaptation
C_AMP_MIN_LSL_GAIN_AD	<u>Description:</u> **Minimum ambient pressure to allow fuel cut Ip (pump current) determination** To control impact of pressure at high altitude (shift of characteristic) the minimum ambient pressure has to be limited. Common conditions are supposed to be covered by exhaust gas pressure calculation as best as possible and correction within offset calculation. Therefore this is just a final threshold beyond which the inputs are not assumed to have suitable accuracy.
C_TTIP_MIN_LSL_GAIN_AD C_TTIP_MAX_LSL_GAIN_AD	<u>Description:</u> **Minimum/Maximum tip temperature to allow fuel cut Ip (pump current) determination** Within the defined range the heater management is assumed to run at controlled mode. Also shifts of signal characteristics are assumed to be quite low and can be handled within the correction applied at the adaptation (see IP_FAC_COR_IPLSL_TTIP_MES_LSL_UP below).
C_N_32_MIN_LSL_GAIN_AD C_N_32_MAX_LSL_GAIN_AD	<u>Description:</u> **Minimum/Maximum engine speed to allow fuel cut Ip (pump current) determination** Within the defined range the engine is running at common mode. Shifts of signal characteristics can be handled within the correction applied at the adaptation (see IP_FAC_COR_IPLSL_N_32 below).
C_N_32_BOL_LSL_GAIN_AD C_N_32_TOL_LSL_GAIN_AD	<u>Description:</u> **Additional minimum/maximum engine speed above/below which fuel cut Ip determination is allowed to take place.** Together with C_N_32_MIN_LSL_GAIN_AD and C_N_32_MAX_LSL_GAIN_AD the area within which gain adaptation is allowed can be shaped more flexible.
C_FAC_LSL_GAIN_AD_MAN_ADJ	<u>Description:</u> **Value for manual adjustment of WRAF sensor gain adaptation factor** Defined value for the adaptation factor, which can be set manually by calibration (see below).
LC_FAC_LSL_GAIN_AD_MAN_ADJ_ENA	<u>Description:</u> **Enable manual adjustment of WRAF sensor gain adaptation factor** Logical constant to set the adaptation factor to a defined value.
LC_LSL_GAIN_AD_ENA	<u>Description:</u> **Enable WRAF sensor gain adaptation** Logical constant to enable adaptation on principal. Can be used also as manual inhibition.

LC_LSL_GAIN_AD_ENA_TEST	**Description: Manual enable of LV_LSL_GAIN_AD_REQ[i] for test purposes** Logical constant to enable adaptation without respect to environmental conditions. The constant is used just for application purposes.
LC_MAP_PUC_LIM_REQ_LSL_GAIN_AD	**Description: Enable request to torque management for limitation of MAP during PUC** To ensure low HC evaporation into exhaust gas system during PUC phase MAP is increased via TQDR. This flag enables the announcement that this should be done.
LC_MAP_PUC_LIM_ACT_NOT_VLD	**Description: Flag to ignore activation of limitation of MAP during PUC by torque management** According to project philosophy adaptation can also take place even if TQDR does not limit MAP during PUC phase to ensure low HC emissions. This flag enables adaptation even without co-operation of TQDR.
LC_IPLSL_GAIN_SEL_LSL_GAIN_AD	**Description: Determines which Ip gain has to be active during execution of gain adaptation.** ' 0 ' = gain 8 ; ' 1 ' = gain 16.

Heck of HC emissions:

First of all HC emissions at fuel cut off (PUC) have to be checked. Depending on engine revolution maximum air mass flows must be determined, to limit HC emission out of engine (oil etc.) into exhaust gas system during PUC. High HC emission trouble the measurement (shift of characteristic). These maximum limits are handled by AGGR TQDR module D02V (Minimum indicated engine torque offset during PUC). For calibration please have a look there.

C_CTR_STOP_FSD_THD	**Description: Threshold for CTR_STOP_FSD to allow gain adaptation.** Caused by (unsuccessful) coldstarts in quick succession some amouts of fuel can reach into the oil. If the oil warms up the fuel may vaporise and additional HCs may be contained in the exhaust gas. This can lead to a wrong measurement of the pump current during PUC phase. The variable CTR_STOP_FSD (out of "fuel system diagnosis") is a measure for the dilution of the oil with fuel. Gain adaptation should be only allowed if CTR_STOP_FSD falls below a certain value.

Variation parameter:

In general based on a chosen base point, each of the parameters (TTIP; N_32, MAF, AMP) has to be varied on its own. The tests are proposed to run several times for each measurement point cycling between the observed PUC conditions (measurement) and PL conditions (long enough stabilisation phase). All other parameters are supposed to be fixed during variation of one parameter. Most corrections itself are quite small but may have influence if summed up. Please have a close look during calculation of the calibration parameters not to generate 1/factor instead of factor.

C_NR_GEAR_MIN_LSL_GAIN_AD	<u>Description:</u> **Minimum gear to allow fuel cut Ip (pump current) determination** Minimum gear to run the adaptation. Usually adaptations at lowest gear are not sufficient, as change of revolution during the acquisition is quite high. Valid adaptation therefore occurs quite rare.
C_CRLC_IPLSL_COR_FCUT	<u>Description:</u> **Correlation constant for filtering of corrected pumping current** The correlation constant ensures to smooth the signal in first order. Smoothing shouldn't be too much, not to extend the start of adaptation and to fulfil reasonable signal values.
C_IPLSL_NOM_PURGE	<u>Description:</u> **Nominal pumping current at fuel cut** This is the value the sensor should show at air. It is well defined by the specification of the sensor. For the calibration tests itself the actual signal value of the sensor used has to be checked (measurement at air) and calibrated here.
C_IPLSL_COR_GRD_THD_FCUT	<u>Description:</u> **Maximum gradient of low pass filtered pumping current to detect stable conditions (end of PURGE)** As one of the entry criteria for the start of an acquisition the gradient of the signal value has to be below a defined threshold. This guarantees stable, not increasing heavily signals.
C_FAC_IPLSL_COR_THD_FCUT	<u>Description:</u> **Factor for minimum threshold of the low pass filtered pumping current to detect stable conditions (end of PURGE)** As one of the entry criteria for the start of an acquisition the signal value has to reach at least a defined fraction of the nominal value. This guarantees not start adaptation to early and to dismiss low input signals, which might crawl and therefore would fulfil gradient criteria.
C_T_ACQ_IPLSL_PURGE	<u>Description:</u> **Duration of Ip (pump current) acquisition and averaging** The duration of the acquisition is being orientated towards the intended accuracy of the adaptation and the frequency and duration of PUC phases. The longer the acquisition takes place the better the factor is (less dispersion). The less and shorter PUC phases are expected the less valid adaptations will take place.
IP_FAC_COR_IPLSL_MAF_KGH	<u>Description:</u> **Factor for MAF correction** Map to realise the influence of the air mass flow. Usually this is the biggest impact on adaptation factors.
IP_FAC_COR_IPLSL_N_32	<u>Description:</u> **Factor for engine speed correction** Map to realise the influence of the engine revolution. Usually the impact on adaptation values is quite low.
IP_FAC_COR_IPLSL_TTIP_MES_LSL_UP	<u>Description:</u> **Factor for tip temperature correction** Map to realise the influence of the tip temperature. If the heater management is well calibrated the map will not have big impact as temperature stays near setpoint (nearly no deviation).
IP_FAC_COR_IPLSL_CHG_PURGE	<u>Description:</u> **Factor for residual gas after purging correction** The map is currently used to consider systematic shifts or failures. Therefore It can be calibrated after

	the primary calibration to bring adapted factors, which are stable but not near the neutral values (=1), to the neutral value.
IP_FAC_COR_IPLSL_AMP	**Description: Factor for AMP correction** Map to realise the influence of the ambient pressure. If the exhaust gas pressure model is well calibrated the map is superfluous and can be calibrated to neutral values (=1). Calibration can be done/checked far at the end at altitude trip/chamber.

Adaptation finalised:

After calibration phase of the signal values, some framework conditions have to be defined to ensure best accurate results of adaptation.

C_IPLSL_VARI_THD_MIN _PURGE C_IPLSL_VARI_THD_MAX _PURGE	**Description: Maximum variance of the low pass filtered pumping current to confirm stable conditions - Minimum variance of the low pass filtered pumping current to confirm that sensor is operable** After acquisition of the signal the quality is evaluated. One criterion is the variance. Being to high means a signal was not quite stable. Also being to low means the sensor might not be alive and just delivers constant values.
C_CRLC_IPLSL_MV_VLD	**Description: Correlation constant for filtering of valid Ip (pump current) samples (is also time constant for adaptation)** The correlation constant ensures to smooth the values (signal and variance) of all valid fuel cuts in first order. Smoothing shouldn't be too much, not to elongate taking into account a permanent step of signal at PUC (change of environmental conditions or real change of sensor behaviour). Also correlation should not be to fast to avoid consideration of just single spikes of signals.
C_IPLSL_MV_VLD_VARI _MAX	**Description: Maximum permitted variance of valid Ip (pump current) sample to allow adaptation** After acquisition of the signal the quality of calculated values is evaluated. One criterion is the variance. Being to high means the valid signals spreading too much for the last measurements, meaning they are not quite stable and therefore shouldn't be considered.
C_FAC_LSL_GAIN_AD_LIM _MIN C_FAC_LSL_GAIN_AD_LIM _MAX	**Description: Minimum/Maximum threshold for WRAF sensor gain adaptation** Within the defined range the adapted factors are taken into account. To low or to high factors are limited or even set to neutral value.
LC_FAC_LSL_GAIN_AD_LIM _RST	**Description: Switch for reset of adaptation factor to one when exceeding thresholds, or for adoption of threshold value, resp.** With this flag the factors can be reseted to neutral value (=1), if they exceed the defined range.
LC_FAC_LSL_GAIN_AD_ENA	**Description: Enable adoption of WRAF sensor gain adaptation factor** With this flag the adaptation can be finalised. The calculated and valid factor is taken into account. If not set the factor is set to neutral value (=1), meaning that the adaptation will not have any impact.

16.74 O2 sensor (lin, up) operability detection

16.74.1 Calibration interfering functions

16.74.1.1 Prerequisites

The following functions must be working before the operability detection can be calibrated:

- Engine operating functions with impact on exhaust temperature (e.g. ignition, catalyst heating)
- Upstream exhaust temperature model, including dew point detection
- Lambda sensor heater management (at least the precontrol part)
- Lambda sensor temperature determination
- WRAF sensor control and sensor signal acquisition must produce sensible lambda values around lambda=1. Pressure/temperature correction maps and gain switching are not needed yet.

16.74.2 Calibration flowchart

It is suggested to first set all data to the values given below and then verify if the function works correctly. If not, data must be changed according to the directions given for the individual labels and verification must be repeated.

16.74.3 Calibration method

16.74.3.1 Ressources

Most of the calibration may be done on the road side and/or on an engine dyno. Some data must be measured in the cold (cold trip or climate chamber), a weak heater limit sensor is required for this and it must be possible to set the battery voltage to the lowest acceptable value. A slow dynamic limit sensor may be useful to calibrate parts of the raw readiness detection.

16.74.3.2 Constraints

Legal requirements concerning closed loop lambda control may limit the applicable values for some of the labels:

- The sensor may be required to be ready after a certain time. The maximum possible delay from engine start to sensor activation is either the sum of C_T_DLY_LS_UP_READY_TEG_MIN and the time until the exhaust model temperature reaches the dew point or ID_T_TOUT_LS_UP_READY, whichever is larger.
- C_MAF_INT_PUC_MAX_LS_UP_READY and LC_TEG_MIN_DEAC provide ways to revoke sensor readiness in the middle of a driving cycle. If sensor readiness must be kept once it was established, these ways must be blocked by calibration either here or in the exhaust temperature model.

16.74.4 Calibrateable data

16.74.4.1 Readiness detection

There are two alternative paths to detect raw readiness: A lambda band which must be left by the sensor signal and a deviation integral which must exceed a certain threshold. Once the raw readiness was detected, full operative readiness is established when the measured tip temperature exceeds a limit. The function can be calibrated such that raw readiness is set unconditionally after a delay (see below) has elapsed.

ID_T_DLY_CHK_LS_UP_READY	**This delay timer runs while the sensor heater is not in state LSH_POW_RED, raw readiness detection is blocked until it has elapsed.** **Use 10s as a start.** **Must be greater than the initial LSH_POW_RISE**

	phase to avoid starting readiness detection before the dew point was passed.
C_LAMB_THD_MIN_LS_UP_READY **C_LAMB_THD_MAX_LS_UP_READY**	**The labels describe a band which must be left by the sensor signal for raw readiness to be detected.** **Set minimum to 0.98 and maximum to 1.02 for a start. Make the band narrower if it is not left by the signal of a good sensor, make it wider if the signal of a sensor with open Vip line leaves it.** **Set minimum to 2 and maximum to 0 to set raw readiness after the delay with no regard to sensor signal.**
C_CRLC_LAMB_LS_UP_READY **C_T_LAMB_DE_INT_RST** **C_LAMB_DE_INT_THD_LS_UP_READY**	**C_CRLC_LAMB_LS_UP_READY is a correlation constant for calculation of a moving mean value of the sensor signal. The deviations between signal and mean value are integrated for the duration of the timer C_T_LAMB_DE_INT_RST. If the integral exceeds C_LAMB_DE_INT_THD_LS_UP_READY, raw readiness is detected.** **Apply 0.05 for the CRLC, 5s for the timer and 0.002 for the threshold as a starting point.** **The integral must reliably exceed the threshold even for a slow sensor. With Open circuit in Vip, the integral must stay below the threshold. Lowering the CRLC constant and the threshold and rising the timeout will make raw readiness detection via this path easier, and vice versa. Do not set the timer higher than the timeout**
C_TTIP_MIN_LS_UP_READY	**If the tip temperature exceeds this limit, full sensor readiness is assumed. Must be coordinated with the activation limits for Ip control in the WRAF sensor control.**

16.74.4.2 Readiness timeout monitoring

These timers run rather independently. If readiness was not detected when both of them timed out, forced readiness is set and fault detection is triggered. The weak heater limit sensor is needed and the battery voltage should be lowered to the acceptable limit. Measurements should take place in the cold.

C_T_DLY_LS_UP_READY_TEG_MIN	**This time must pass after dew point detection before the sensor readiness is checked. 120s is a safe starting point.**

ID_T_TOUT_LS_UP_READY	**This timeout runs after engine start, 200s may be used as a starting point.** **Final values should be much lower, sensors usually take 10 to 20s to get ready once full heater power is supplied. Worst case must be measured with above mentioned conditions and limit parts, and some safety should be added. Only one of the timers needs to be calibrated, the other one may be left at 0.**

16.74.4.3 Readiness revocation

These data control readiness revocation in conditions where the exhaust gas is too cold. If readiness is revoked, readiness detection and timeout monitor will restart once the condition is relieved. The weak heater limit sensor is needed and the battery voltage should be lowered to the acceptable limit. Measurements should take place in the cold.

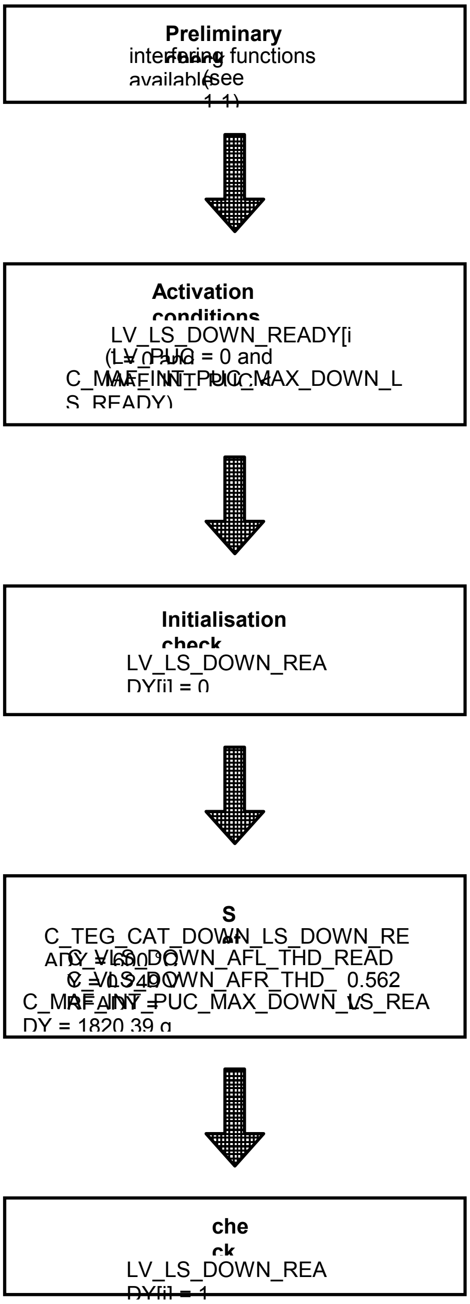
LC_TEG_MIN_THD_DEAC_TEST	**This is for test purposes only and must be set to 0**
LC_TEG_MIN_DEAC	**If set to 0, the readiness flag is revoked when the exhaust model temperature drops back below the dew point.**
C_MAF_INT_PUC_MAX_LS_UP_READY	**Minimum air mass which is sufficient to cool the sensor below minimum operating temperature during fuel cut off. Set to the maximum of 2912.67g, then check if the sensor becomes too cold during PUC and lower the value as required. Subtract a fair safety margin for cases like consecutive fuel cut off phases interrupted only by short part load phases (e.g. long descents in the mountains).** **Values far above 100g should be expected.** **Keep at maximum to disable the feature.**

16.75 Downstream oxygen sensor readiness detection

16.75.1 Calibration interfering functions

- Downstream oxygen sensor heater control
- Exhaust gas temperature model
- Lambda sensor voltage
- Catalyst purge

16.75.2 Calibration flowchart



16.75.3 Calibration method

_DOWN_READY	Temperature threshold above which O2 sensor recognised as operatively ready (**e.g. 600 °C**).
C_VLS_DOWN_AFL_THD_READY	Lean VLS_DOWN[i] threshold for operative readiness detection (**e.g. 0,249 V**)
C_VLS_DOWN_AFR_THD_READY	Rich VLS_DOWN[i] threshold for operative readiness detection (**e.g. 0,562 V**)
C_MAF_INT_PUC_MAX_DOWN_LS_READY	Maximum MAF integral during PUC for operability reset (**e.g. 1820,39 g** at the beginning).

16.76 Air-Fuel cycle evaluation

16.76.1 Calibration interfering functions

- Readiness detection of upstream oxygen sensor (LV_LS_UP_READY[NC_CBK_EX_NR])
- Lambda control (LV_LAM_LSCL[NC_CBK_EX_NR])
- Lambda sensor voltage (VLS_UP[NC_CBK_EX_NR])

16.76.2 Calibration flowchart

Preliminary check
interfering functions available (see 1.1)

Activation conditions
LV_ST_END = 1 and LV_ES = 0

Initialisation check
LV_AFL[i] = 0
T_AFL_CYC[i] = 0
T_AFL_CYC_HLD[i] = 0
T_AFR_CYC[i] = 0
T_AFL_LOCK[i] = 2.55 s (i.e. FFH)
CTR_AFL_CYC[i] = 0

Set
C_VLS_AFR_THD = 0,498 V
C_VLS_AFL_THD = 0,352 V
IP_T_AFL_LOCK = 0 all over

check for LV_LAM_LSCL[i] = 1

when VLS_UP[i] exceeds the threshold C_VLS_AFR_THD then the flag LV_AFL[i] shall be set to zero indicating that the air fuel mixture is rich. As long as LV_AFL[i] = 0 the timer T_AFR_CYC[i] has to be incremented. As soon as LV_AFL[i] changes from 1 to zero the timer T_AFR_CYC[i] shall be reset.

when VLS_UP[i] falls short of the threshold C_VLS_AFL_THD then the flag LV_AFL[i] shall be set to 1 indicating that the air fuel mixture is lean As long as LV_AFL[i] = 1 the timer T_AFL_CYC[i] has to be incremented. As soon as LV_AFL[i] changes from zero to 1 the timer T_AFL_CYC[i] shall be reset.

The Air-Fuel cycle counter CTR_AFL_CYC[i] shall be incremented for every lean to rich AF mixture transition (LV_AFL[i] 1 -> o), to indicate that a complete AF cyclehas passed.

Calibrate
IP_T_AFL_LOCK

16.76.3 Calibration method

The function shall detect a change in AF mixture from lean to rich or vice versa ignoring possible jitter in the oxygen sensor signal. This shall be facilitated by the following method: If the timer T_AFL_LOCK[i] falls below the threshold IP_T_AFL_LOCK, the timer shall be incremented and no further action shall take place in the current recurrence. Should, during following recurrence cycles, the timer equal or exceed the threshold and if the AF mixture flag is currently indicating lean, the function shall observe the VLS_UP[i] signal for a change to rich as indicated by VLS_UP[i] exceeding the threshold C_VLS_AFR_THD. Once this has occurred, the LV_AFL[i] flag shall be changed to indicate a rich mixture (LV_AFL[i] = 0), a counter CTR_AFL_CYC[i] shall be incremented to indicate the number of complete AF cycles since the last activation of the lambda controller, the output data T_AFL_CYC_HLD[i] shall be updated with T_AFL_CYC[i], the timer T_AFL_LOCK[i] shall be reset and a SW internal flag set to indicate the AF mixture transition. The same procedure, except incrementation of counter CTR_AFL_CYC[i], applies for a currently rich mixture. A transition to a lean mixture shall be detected by VLS_UP[i] falling below threshold C_VLS_AFL_THD.

C_VLS_AFR_THD	Threshold for detection of rich mixture (**e.g. 0,498 V**).
C_VLS_AFL_THD	Threshold for detection of lean mixture (**e.g. 0,352 V**).
IP_T_AFL_LOCK	LV_AFL[i] transition lock-out time after transition of AF mixture. The timer T_AFL_LOCK[i] and associated engine operating point dependent threshold IP_T_AFL_LOCK reduce the effect of rapid oxygen sensor signal changes, noise generated by cylinder to cylinder AF mixture deviations, that would cause the lambda controller to execute a P-jump. The mapped lock-out time represents the delay between injection of the AF mixture and the burnt AF mixture reaching and being detected by the oxygen sensor. The lock-out time shall not exceed the exhaust gas delay for any particular engine operating point otherwise the lambda controller frequency will be affected. Oxygen sensor signal changes occurring within this delay time may not be considered to be due to the change in controlled lambda but from noise and shall be ignored. (**e.g. see below**)

***Example for** IP_T_AFL_LOCK [s]*

MAF [mg/stk] **N [rpm]**	**704**	**960**	**1312**	**1760**	**2400**	**3264**	**4448**	**6016**
70.79	0.67	0.56	0.46	0.39	0.30	0.25	0.22	0.20
92.60	0.60	0.50	0.40	0.33	0.25	0.20	0.18	0.16
125.30	0.52	0.43	0.34	0.26	0.19	0.16	0.14	0.12
163.39	0.45	0.38	0.28	0.20	0.16	0.13	0.11	0.10
217.90	0.38	0.32	0.22	0.16	0.13	0.11	0.10	0.09
288.69	0.32	0.26	0.18	0.14	0.12	0.11	0.10	0.09
386.70	0.27	0.22	0.15	0.13	0.12	0.10	0.09	0.08
512.00	0.23	0.19	0.13	0.12	0.10	0.09	0.08	0.07

16.77 Lambda Sensor Diagnosis OBDI

16.77.1 WRAF Sensor electrical diagnosis

16.77.1.1 Calibration interfering functions

16.77.1.2 Before start of the calibration of the electrical diagnosis, the following other functions must be calibrated:

Electrical diagnosis by ATIC 42 controller

- Signal Acquisition of linear lambda sensor (controller)

Pump Current Limitation

- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor

Open Circuit Diagnosis

- Basic Engine Tuning (Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Wallfilm Compensation Model
- Evaporative Emission Control
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Lambda Control Parameter fixed
- Forced Stimulation
- Basic Lambda Setpoint
- Trim Control
- WRAF Sensor Plausibility Diagnosis during PUC
- WRAF Sensor Plausibility Diagnosis
- WRAF Sensor Activity Check
- WRAF Sensor Heater OBDII Diagnosis

16.77.1.3 Before calibration of the electrical diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.77.1.4 Calibration flowchart

The sequence is described by the order of constants and maps in the chapter “Calibration method” below.

16.77.1.5 Calibration method

Electrical Diagnosis by ATIC42 controller

C_T_DLY_DIAG_EL_LSL_UP	Time delay to start the electrical failure diagnosis; Check if there is any increment of the anti bounce counter in case of no delaytime. If nothing can be observed, choose 0 for this delaytime.

Pump Current Limitation

The purpose was to disable the IP loop for very rich or very lean A/F ratio to prevent a wrong A/F control in case of not reliable IP current. Due to the fact, that there is no MIL illuminated, its not possible to furthermore use this diagnosis. In fact it will not have any effect as the WRAF Sensor control no longer reacts to the symptom of this diagnosis. It will be changed or removed for some later revisions of this module. If there is really a problem on the pump current in an emission relevant operation area of the engine, it will lead to another failure entry (i.e. trim control setpoint deviation exceeded).

C_CTR_IPLSL_SYM_DIAG_LSL_UP	Max. error debounce counter; To prevent a wrong A/F ratio for a longer time due to unreliable IP current, the lambda control should be disabled within a few seconds. The function is processed all 100 ms. I.e. 30.
C_LAMB_SP_AFR_IPLSL_DIAG_LSL_UP	Max. rich lambda set point to activate the IP diagnosis; It must be significantly more rich than the richest AF ratio during emission test to not interfere into the emission test result. I.e. 0,8.
C_LAMB_SP_AFL_IPLSL_DIAG_LSL_UP	Min. lean lambda set point to activate the IP diagnosis; It must be significantly more lean than the leanest AF ratio during emission test to not interfere into the emission test result. I.e. 1,2.
LC_IPLSL_SYM_DIAG_LSL_UP	Logical variable to force IPLSL symptom flag (only for test purpose); Must be 0.

Open Circuit Diagnosis

Depending on the diagnosis functions which provide the INPUT for the Open Circuit diagnosis, it can be attempted to detect the Open Circuit in Idle Speed.
It´s important to check the timing during emission test.
The relevant failure cases are:
For each Pin (Vn, Vg, Vip, Vrc) test the failure entry in case that the open circuit happened

1. before the beginning of the driving cycle (damage of the wire during the night by an animal).
2. after operative readiness of the WRAF Sensor

Test case 1 and 2 in

- Idle Speed,
- Part Load,
- Dynamic Driving

and check out where the problems are to detect the open circuit.

In case of some WRAF Sensors, its not possible to distinguish open circuit on Vg and Vip, as the tip temperature measurement continues to work almost normal even with open circuit on Vg. That results in a failure entry "Open Circuit on Vip" even if in reality the line to

Vg is open. In this case, the waiting timer C_T_WAIT_SYM_OBD_2_LSH_UP can be calibrated to 1 sec.

C_VLS_UP_DIAG_MIN_OC_LSL_UP	LSL undervoltage threshold; Smaller than the smallest VLS_UP_DIAG for normal (rich) operation and bigger than 0 to cover some noise on the wire. I.e. 0,1 V.
C_T_VLS_UP_MIN_DIAG_OC_LSL_UP	Max allowed time for LSL undervoltage; As the reaction of VLS_UP in case of a real open circuit on Vn is very fast, this time can be short. I.e. 5 sec.
LC_DIAG_PUC_END_MAN_LSL_UP	Boolean switch to bypass activation of LV_DIAG_PUC_END_LSL_UP[i]; If there is no fuel cut off during the emission test cycle, a possible open circuit on Vrc would not be detected. The best way is to force fuel cut off as long as it is required to check the open circuit on Vrc. Otherwise, the authority must be informed that its not possible during emission test to perform the open circuit diagnosis on Vrc. To anyway set the end flag of the open circuit diagnosis even without fuel cut off, the flag could be set to 1.
C_T_WAIT_SYM_OBD_2_LSH_UP	Waiting time to check if OBD2 upstream heater error is present (to distinguish OC on Vip from Vg); Some seconds longer than C_TCC_THD_OBD_LSH_UP to be sure that a possible lambda sensor heater error could have been detected. E.g. 65 sec. In case of a WRAF Sensor which does not show a tip temperature error in case of open circuit on Vg, calibrate this time to 1 sec. (not 0 as there will occure another unwanted effect).
C_T_DIAG_OC_RST_OSC_DI	Time to reactivate oscillator for upstream tip temperature measurement; Must be longer than C_T_VLS_UP_MIN_DIAG_OC_LSL_UP.
C_T_DIAG_OC_LSL_UP	Timer to finish OC diagnosis in case of OBD2 upstream heater error symptom active and conditions for plausibility diagnosis fulfilled; It must be slightly longer than the time to debounce the “Sensor not active” check and the plausibility diagnosis: **(**IP_T_SYM_AFS_PLAUS + (1/10 * C_CTR_THD_DIAG_PLAUS_LSL_UP/ C_CTR_INC_DIAG_PLAUS_LSL_UP) + (C_CTR_SUM_RAW_ACT/(smallest value of ID_CTR_INC_RAW_ACT_LSL_UP * (1 sec / 20 ms))) + longest possible T_SUM_AFL_AFR_CYC[i] * (C_CTR_LAMB_DE_INT_ACT_LSL_UP + 1) * ((C_CTR_SYM_MAX_ACT_LSL_UP / C_CTR_SYM_INT_ACT_LSL_UP) + 1) **)**
C_T_DIAG_OC_LSL_UP_LSH_ERR	Timer to finish OC diagnosis in case of OBD2 upstream heater error symptom active and LAMB plausible; This is the time to ensure that LV_DIAG_RAW_ACT_SYM_LSL_UP could have been set to 1 in case of Open Circuit on Vg. It must be slighlty longer, i.e. 2 sec. more than (C_CTR_SUM_RAW_ACT/(smallest value of ID_CTR_INC_RAW_ACT_LSL_UP * (1 sec / 20 ms)).

16.77.2 WRAF Sensor controller diagnosis

16.77.2.1 Calibration interfering functions

16.77.2.1.1 Before start of the calibration of the WRAF Sensor controller diagnosis, following other functions must be calibrated:

- Signal Acquisition of linear lambda sensor (controller)

16.77.2.1.2 Before calibration of the electrical diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.77.2.2 Calibration flowchart

The sequence is described by the order of constants and maps in the chapter "Calibration method" below.

16.77.2.3 Calibration method

C_CTR_MAX_SPI_COM_LSL_UP[NC_CBK_EX_NR]	Max. allowed number of SPI communication errors; I.e. 5.
C_T_RST_MAX_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	Max. waiting time for the initialization of the WRAF sensor controller; I.e. 130 sec.
C_T_END_DIAG_LSL_UP_IF[NC_CBK_EX_NR]	Time to diagnostic availability; I.e. same duration than C_T_RST_MAX_DIAG_LSL_UP_IF.

16.78 Lambda sensor downstream probe diagnosis (plausibility)

16.78.1 Calibration interfering functions

16.78.1.1 Before start of the calibration of the downstream lambda sensor plausibility diagnosis, following other functions must be calibrated:

- Basic Engine Tuning (i.e. Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Basic Lambda Setpoint
- Full load correction
- Downstream lambda sensor heater calibration
-

16.78.1.2 Before calibration of the downstream lambda sensor plausibility diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.78.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below.

16.78.3 Calibration method

Plausibility Diagnosis during Fuel Cut Off

The calibration can be done on the road.
The calibration must be checked in emission test.
If no fuel cut off is allowed during emission test, the setting of the general end flag LV_LS_DOWN_DIAG_END must be adapted accordingly via C_NR_END_DIAG_LS_DOWN.

C_VLS_MIN_DIAG_PUC_LS_DOWN	VLS voltage threshold for PUC fault detection; This threshold must be smaller than the value for a sensor with open circuit minus tolerances (i.e. 0,37 V). And it must be bigger than the standard fuel cut off voltage plus tolerances. Besides the tolerances from ECU, the bigger part of the tolerances arise due to a weak heater, a cool sensor due to operation conditions or an engine which releases plenty of oil during fuel cut off. Take i.e. the voltage value from sensor specification for lambda = 1,10 or leaner. Take the Oxygen Sensor with limit low heater power and observe the sensor voltage during fuel cut off in all possible environment, especially very cold conditions (i.e. cold ambient temperature, fuel cut off right after operative readiness of the sensor). I.e. 100 mV.

C_MAF_INT_MAX_DIAG_PUC_LS_DOWN	Maximum threshold for MAF integral to end downstream PUC signal plausibility; Check the downstream lambda sensor voltage for long fuel cut off phases. Use the Oxygen Sensor with limit low heater power and a lifetime catalyst (i.e. 150 thousand miles aged for SULEV). Take a maximum maf integral which would lead to a significant voltage increase (i.e. 1/10 of the failure threshold) due to the cooling of the sensor. If no voltage change can be observed, the value can be as large than the MAF Integral for reset of the operative readiness of downstream Oxygen Sensor. Countercheck this calibration in very cold conditions.
C_MAF_INT_MIN_DIAG_PUC_LS_DOWN	Minimum threshold for MAF integral required to start downstream PUC signal plausibility; Air mass integral to purge the catalyst after start of fuel cut off. A catalyst with maximum O2 load is required (be careful with greening effect – better take a catalyst which was used for some hours). Measure the duration between start of fuel cut off and downstream signal voltage below C_VLS_MIN_DIAG_PUC_LS_DOWN. Multiply the measured time with a safety factor. This safety factor must cover a slow (aged) sensor which can still be used for trimming of A/F ratio and for catalyst diagnosis. Therefore its best to use a sensor which is already aged (i.e. artificially "aged" with small wholes in the protection tube), but still suitable for above mentioned functionality and to measure the delay with this sensor. To have no risk of wrong failure detection, anyway a safety factor should be considered. As there is the switching time diagnosis for the downstream oxygen sensor, the safety factor can be choosen large (i.e. above 100%). Check if the required duration of fuel cut off for a failure detection is shorter than the available fuel cut off time during emission test.
C_MAF_INT_PUC_NOT_LS_DOWN_DIAG	Minimum threshold for MAF integral out of PUC required to start downstream PUC signal plausibility; Useful for projects with a weak downstream O2 sensor heating. This MAF integral serves to prevent a wrong failure entry due to accumulated cooling energy in case of only briefly interrupted long fuel cut off phases. It goes in combination with a minimum MAF_KGH threshold, see C_MAF_KGH_MIN_LS_DOWN_DIAG. The MAF integral can be calibrated by cooling down the sensor during a long fuel cut off phase with high engine speed (worst case cooling). Another condition for the worst case is, that the whole exhaust line is still cool (start the measurements right after operative readiness - never going to high loads). Then go to part load and measure the time and the MAF integral until the sensor is warm enough (i.e. to show a lean signal). Make several measurements subsequently: long PUC until the sensor voltage increases in PUC (too cold), then PL for x sec. – increase the duration x for every measurement until the time in part load was long enough that the sensor is warmed up after it was cooled down.

	If the sensor does never cool down during long PUC phases, then calibrate this value to 0.
C_FAC_MAF_INT_LS_DOWN_DIAG	With this factor, it can be adjusted how big the impact of fuel cut off is to the lambda sensor temperature. If the sensor cools down very fast, the factor should be big. If the sensor almost never comes to a condition where it would be too cold, the factor should be small. If the factor is big, a short fuel cut off phase already requires a relatively long engine operation with positive torque generation afterwards to resume the diagnosis.
C_MAF_KGH_MIN_LS_DOWN_DIAG	The heat flow to the sensor is only considered, if the load is above this threshold.

<u>Full load (FL) oxygen sensor signal plausibility</u>

The calibration can be done on the road.
The diagnosis will in most vehicles not be active during the emission test as no full load occurs. In this case, the setting of the general end flag LV_LS_DOWN_DIAG_END must be independent from this diagnosis (calibration via C_NR_END_DIAG_LS_DOWN).

The sensor manufacturer gives just a guarantee for a rich voltage level which corresponds to a maximum aged lambda sensor.
In real life, the lambda sensors show this maximum guaranteed voltage already during standard operation at stoichiometric Air Fuel Ratio.

C_VLS_MAX_DIAG_FL_LS_DOWN	VLS voltage threshold for FL fault detection; There are two mainstreams which can be taken for calibration of this constant: 1. If the vehicle is rich mixture sensitive and the vehicle manufacturer agrees to change a sensor which shows still a voltage far over the specified minimum value in the sensor specification, then: Take a value which is significantly higher than the downstream fuel trim requires to decrement the fuel trim correction value. 2. If the vehicle manufacturer doesn´t agree to change a sensor which is still within the sensor specification, then: Take the guaranteed minimum rich voltage from sensor specification. In this case it must be checked if the sensor amplitude of such a sensor is enough for catalyst diagnosis malfunction detection. And it must be checked if for the vehicle lifetime the emission level can be kept. The vehicle manufacturer must agree.
C_MAF_INT_MAX_DIAG_FL_LS_DOWN	Maximum threshold for MAF integral to end downstream FL signal plausibility; The sensor signal is temperature dependent. In case of extended duration of full load it is possible that the sensor gets too hot and the sensor signal decreases. Take the oxygen sensor with maximum heater power. Stay in maximum full load and observe the voltage. If the voltage decreases although the lambda remains the same, then choose the mass air flow integral at this time to stop the diagnosis. E.g. 110 g.

C_MAF_INT_MIN_DIAG_FL_LS_DOWN	Minimum threshold for MAF integral required to start downstream FL signal plausibility; Air mass integral to purge the catalyst after start of full load. A catalyst with maximum O2 load is required. Measure the duration between start of Full Load (lowest full load possible) and downstream signal voltage above C_VLS_MAX_DIAG_FL_LS_DOWN. Multiply the measured time with a safety factor. This safety factor must cover a slow (aged) sensor which can still be used for trimming of A/F ratio and for catalyst diagnosis. Therefore its best to use a sensor which is already aged (i.e. artificially "aged" with small wholes in the protection tube), but still suitable for above mentioned functionality and to measure the delay with this sensor. To have no risk of wrong failure detection, anyway a safety factor should be considered. As there is the switching time diagnosis for the downstream oxygen sensor, the safety factor can be choosen large (i.e. above 200 %). Take care that the required duration of full load is reasonable during all day life operation. E.g. 20 g

16.79 Lambda Sensor Diagnosis OBDII

16.79.1 Active plausibility diagnosis

16.79.2 Calibration interfering functions

16.79.2.1 Before start of the calibration of active plausibility diagnosis, the following other functions must be calibrated:

- Basic Engine Tuning (i.e. Ignition, **Basic Injection**, Torque Model, Variable Valve Timing)
- **Intake Manifold Model (MAP System)**
- Wallfilm Compensation Model
- Evaporative Emission Control
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Basic Lambda Setpoint

16.79.2.2 Before calibration of the active plausibility diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.79.3 Calibration flowchart

The sequence is described by the order of constants and maps in the chapter “Calibration method” below.

16.79.4 Calibration method

Moving Lambda Setpoint

C_LAMB_SP_STEP_DIAG_LS_UP_DOWN	Number of recurrences to achieve the desired lambda set point; In case that the engine is robust against lambda changes, this value can be choosen quite small for a shorter duration of the whole test. I.e. 5. If time is not the matter, a lambda shift by 15 % within 1 second is reasonable -> then choose 10.
C_LAMB_SP_AFR_DIAG_LS_UP_DOWN	Lambda set point for rich fault detection; Two limits must be considered: 1. proper engine running must be ensured 2. enough margin for not perfectly balanced precontrol must be included Recommendation: at least 15 % rich if the engine runs proper at all operating points except FL or COP.
C_LAMB_SP_AFL_DIAG_LS_UP_DOWN	Lambda set point for lean fault detection; Two limits must be considered: 1. engine running without misfire must be ensured 2. enough margin for not perfectly balanced precontrol must be included Recommendation: 12 - 15 % lean if the engine is still running without misfire (under all operating points

except FL or COP)

State machine of active plausibility diagnosis

The calibration can be done on the road.
The calibration should be checked in emission test.
The failure can be caused i.e. by forcing a dynamic fuel trim diagnosis – too high difference between setpoint and actual downstream lambda sensor voltage :
adding + / - 10 % on IP_LAMB_LS_UP[i].
A new catalyst (not green but with maximum O2 load) is required.

The risk for this function is, that a wrong diagnosis result is found because of a wrong precontrol. It must be ensured, that the shift of the lambda setpoint always guarantees enough tolerance for precontrolled operation. The setpoint must be shifted strongly enough, to guarantee that a real rich or lean mixture is the result of the lambda setpoint shift.

The function makes sense if the fuel mass injection adaptation is disabled and reset to 0 at the time of starting the open loop test.

C_MAF_INT_DIAG_LS_UP_DOWN	Minimum threshold for MAF integral required to start active test; Choose an operation point with lowest engine speed / load area where the active test can run (slow reaction time of sensor). Measure the MAF integral until the downstream lambda sensor voltage shows the expected response to the lambda setpoint change.
C_MAF_INT_EOL_DIAG_LS_UP_DOWN	Initialization of MAF integral to reduce required minimum MAF value to start active test; Check the required MAF integral for the conditions of the EOL Test. Measure the MAF integral until the downstream lambda sensor voltage shows the expected effect to the lambda setpoint change.
C_LAMB_AFL_MIN_DIAG_LS_UP_DOWN	Min. lambda set point for lean fault detection; It is related to C_LAMB_SP_AFL_DIAG_LS_UP_DOWN and should be 2 % richer than this.
C_LAMB_AFL_MAX_DIAG_LS_UP_DOWN	Max. lambda set point for lean fault detection; In case that there is a leaner lambda setpoint due to any reason, the active test is interrupted by this lambda setpoint threshold. I.e. 2 % more than C_LAMB_SP_AFL_DIAG_LS_UP_DOWN.
C_LAMB_AFR_MIN_DIAG_LS_UP_DOWN	Min. lambda set point for rich fault detection; It is related to C_LAMB_SP_AFR_DIAG_LS_UP_DOWN and should be 2 % leaner than this.
C_LAMB_AFR_MAX_DIAG_LS_UP_DOWN	Max. lambda set point for rich fault detection; In case that there is a richer lambda setpoint due to any reason, the active test is interrupted by this lambda setpoint threshold. I.e. 2 % richer than C_LAMB_SP_AFR_DIAG_LS_UP_DOWN.
C_T_MAX_DIAG_AFR_LS_UP_DOWN	Timer threshold for plausibility test under rich AF conditions; Can be quite short as all waiting timers to get stable conditions have been waited for. I.e. 10 sec.
C_T_MAX_DIAG_AFL_LS_UP_DOWN	Timer threshold for plausibility test under lean AF conditions; Same than C_T_MAX_DIAG_AFR_LS_UP_DOWN.
C_T_DIAG_EOL_LS_UP	Initialization of timer in case of EOL to reduce the

_DOWN	required minimum activity test time; Check the available time during EOL test and don´t reduce the overall test time under rich or lean conditions below 5 seconds.
LC_DIAG_AUTH_CHK_LS_UP_DOWN	Boolean switch allowing the active test; 1 for activation of the active test.

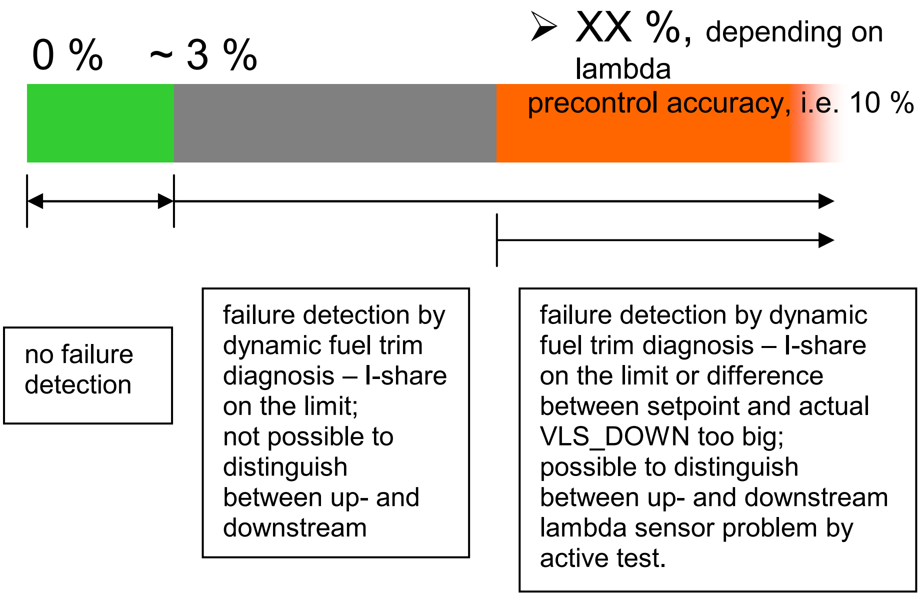
Active Test Upstream Sensor

C_LAMB_AFR_THD_DIAG_ACT_LSL_UP	Lambda value threshold for rich fault detection (upstream sensor); The closer this value is to lambda 1, the more deviation is possible for the precontrolled lambda to still detect a problem on the sensor. It must not be over the typical stoichiometric lambda setpoint. And it should not be smaller than the adjusted rich lambda setpoint during the active test plus the expected maximum deviation of real lambda from lambda setpoint. This value gives an additional opportunity to make the blind area (where no clear distinction between up- and downstream O2 Sensor error is possible) smaller than for the downstream O2 Sensor. The downstream O2 Sensor is always checked against its stoichiometric switching point, so just a big characteristic shift of C_LAMB_SP_AFR_DIAG_LS_UP_DOWN can be observed. Example: Fuel trim on the negative limit due to upstream O2 Sensor characteristic shift by – 10 % (-> LAMB_LS_UP = 1,1 in case of real lambda = 1,0). No failure detection by fuel system diagnosis or plausibility diagnosis. Lean test gives no problem for up- nor for downstream O2 Sensor. Rich test makes 15 % shift to rich. If the sensor would be able to follow this 15 % and the precontrol would be perfect, LAMB_LS_UP would be 0,95. If C_LAMB_AFR_THD_DIAG_ACT_LSL_UP was choosen to be 0,94 the problem will be detected. So the difference between C_LAMB_SP_AFR_DIAG_LS_UP_DOWN and C_LAMB_AFR_THD_DIAG_ACT_LSL_UP is approximately the characteristic shift over which an error could be detected. That also means, that at least between fuel trim limit and approximately 10% shift of characteristic its not possible to distinguish between up- and downstream lambda sensor error.
C_LAMB_AFL_THD_DIAG_ACT_LSL_UP	Lambda value threshold for lean fault detection (upstream sensor); See the explanation for C_LAMB_AFR_THD_DIAG_ACT_LSL_UP.
C_ABC_INC_CHK_LSL_UP	Increment for debounce algorithmus; C_ABC_MAX_CHK_LSL_UP / (C_ABC_INC_CHK_LSL_UP * 10 recurrences / sec) must be significantly shorter than the duration of the lean respectively the rich test (consider also the end

	of line active test duration which is even shorter : C_T_MAX_DIAG_AFL/AFR_LS_UP_DOWN – C_T_DIAG_EOL_LS_UP_DOWN).
C_ABC_MAX_CHK_LSL_UP	Maximum value for debounce algorithmus; See explanation for C_ABC_INC_CHK_LSL_UP.
C_CTR_SUM_RST_AFR_CHK_LSL_UP	Reset debounce counter for Rich test; Must be longer than the time to debounce an error via the anti bounce counter increment and maximum.
C_CTR_SUM_RST_AFL_CHK_LSL_UP	Reset debounce counter for Lean test; See explanation for C_CTR_SUM_RST_AFR_CHK_LSL_UP.

Active Test Downstream Sensor

C_VLS_AFR_THD_DIAG_ACT_LS_DOWN	VLS voltage threshold for rich fault detection (downstream sensor); Voltage slightly below the maximum guaranteed rich voltage of sensor supplier at lambda ~ 0,9 (see sensor specification).
C_VLS_AFL_THD_DIAG_ACT_LS_DOWN	VLS voltage threshold for lean fault detection (downstream sensor); Voltage slightly above the lean voltage at lambda 1,01.
C_T_MAX_CHK_LS_DOWN	Maximum time to debounce the error. It must be significantly shorter than the duration of the lean respectively the rich test. Consider also the end of line active test duration which is even shorter : C_T_MAX_DIAG_AFL/AFR_LS_UP_DOWN and C_T_DIAG_EOL_LS_UP_DOWN).
C_T_END_DIAG_ACT_LS_DOWN	Time for end of active test in case it was not triggered. It must be bigger or equal (C_T_MAX_DIAG_AFL_LS_UP_DOWN + C_T_MAX_DIAG_AFR_LS_UP_DOWN + twice the maximum time for the MAF integral C_MAF_INT_DIAG_LS_UP_DOWN). The reason is to be in line with Rate Based Monitoring requirements (duration of diagnosis in the no failure case not shorter than in the failure case).



0 %

~ 3 %

➢ XX %, depending on lambda precontrol accuracy, i.e. 10 %

no failure detection

failure detection by dynamic fuel trim diagnosis – I-share on the limit; not possible to distinguish between up- and downstream

failure detection by dynamic fuel trim diagnosis – I-share on the limit or difference between setpoint and actual VLS_DOWN too big; possible to distinguish between up- and downstream lambda sensor problem by active test.

16.80 Interchanged Lambda Sensor diagnosis

16.80.1 Calibration interfering functions

16.80.1.1 Before start of the calibration of the interchanged lambda sensor diagnosis, following other functions must be calibrated:

- Basic Engine Tuning (i.e. Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Wallfilm Compensation Model
- Evaporative Emission Control
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Lambda Control Parameter fixed (also limits)
- Forced Stimulation
- Basic Lambda Setpoint
- Trim Control

16.80.1.2 Before calibration of the interchanged lambda sensor diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.80.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below.

16.80.3 Calibration method

Interchanged upstream lambda sensor diagnosis

The calibration can be done on the road.
The failure is created most easily by swapping the sensors.

C_FAC_LAM_LIM_MIN_CHG_LS_UP	Min. Threshold to detect an interchanged sensor; Smaller than lambda control limit.
C_FAC_LAM_LIM_MAX_CHG_LS_UP	Max. Threshold to detect a interchanged sensor; Smaller than lambda control limit.
C_MAF_INT_DIAG_CHG_LS_UP	Max. MAF integral value when ERR_SYM_... ≠ “NO_SYM”; To exclude short term events by dynamic driving. Bigger than required for short term events and smaller as the maximum possible time of active lambda control (or limited lambda control by wallfilm) during emission test. I.e. 150.

C_T_END_DIAG_CHG_LS_UP	Max. time to set LV_END_... = 1; Check how long it takes at lowest possible load to debounce the symptom. Add a few seconds.
C_FAC_INT_CHG_LS_UP	Gain factor for the MAF integral; 1

Interchanged Downstream Lambda Sensors

The calibration can be done on the road.
The failure is created most easily by swapping the sensors.
Different operating points must be checked.
Even the detection in Idle Speed can be checked.

C_VLS_DOWN_MAX_CHG_LS_DOWN	Max VLS_DOWN_i threshold; Exchange the Sensors and observe the "lowest" rich voltage for the bank which is on the rich side. Define the threshold slightly lower than this observed value. The voltage must be outside of the guaranteed rich voltage of the lambda sensor specification, as the guaranteed sensor rich voltage is already reached during normal operation with active trim control. So this diagnosis is not guaranteed for the sensor lifetime.
C_VLS_DOWN_MIN_CHG_LS_DOWN	Min VLS_DOWN_i threshold; Exchange the Sensors and observe the "highest" lean voltage for the bank which is on the lean side. Define the threshold slightly higher than this observed value.
C_MAF_INT_DIAG_CHG_LS_DOWN	MAF integral threshold; It must be smaller than the integral to debounce the error for the fuel trim diagnosis (vls_dif_dly - C_MAF_INT_VLS_DIF_DLY_MIN). Otherwise, the failure is not correctly located on sensors exchanged.
C_FAC_INT_CHG_LS_DOWN	Gain factor for the MAF integral; 1

16.81 Linear Oxygen Sensor plausibility diagnosis

16.81.1 Calibration interfering functions

16.81.1.1 Before start of the calibration of the plausibility diagnosis, following other functions must be calibrated:

- Basic Engine Tuning (Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Wallfilm Compensation Model
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Lambda Control Parameter fixed
- Forced Stimulation
- Basic Lambda Setpoint
- Trim Control

16.81.1.2 Before calibration of the plausibility diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.81.2 Calibration flowchart

The sequence is described by the order of constants and maps in the chapter “Calibration method” below.

16.81.3 Calibration method

Plausibility Check

The calibration can be done on the road, except for calibration data to consider the catalyst delay. Here a chassis dynamometer is recommended.
A break out box is required.
The calibration must be checked in emission test.

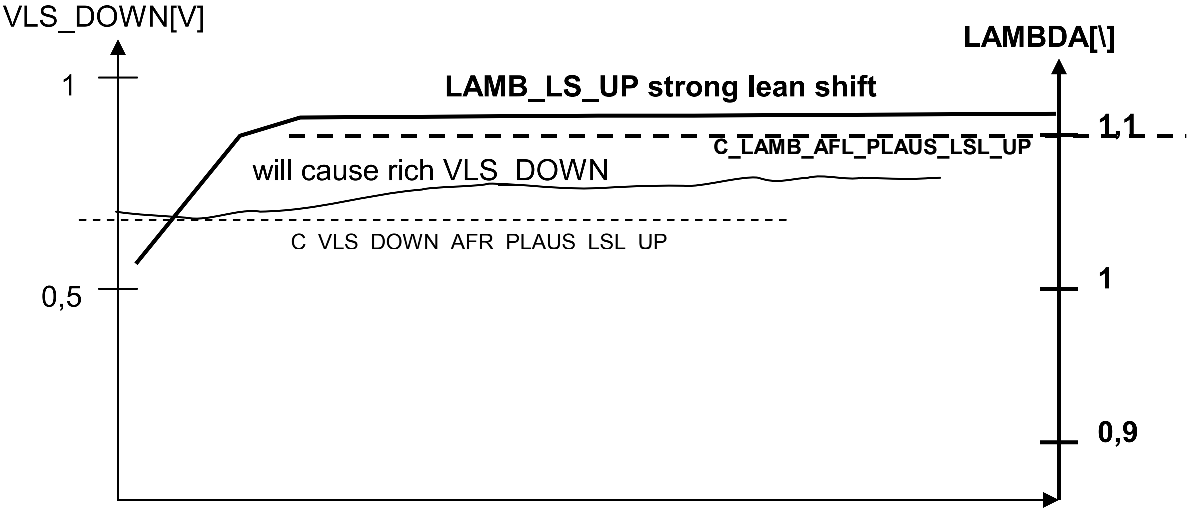
It's acceptable that the diagnosis runs only in Part load. The highest calibration level is reached, if the diagnosis works also in Idle Speed without risk of wrong failure detection.

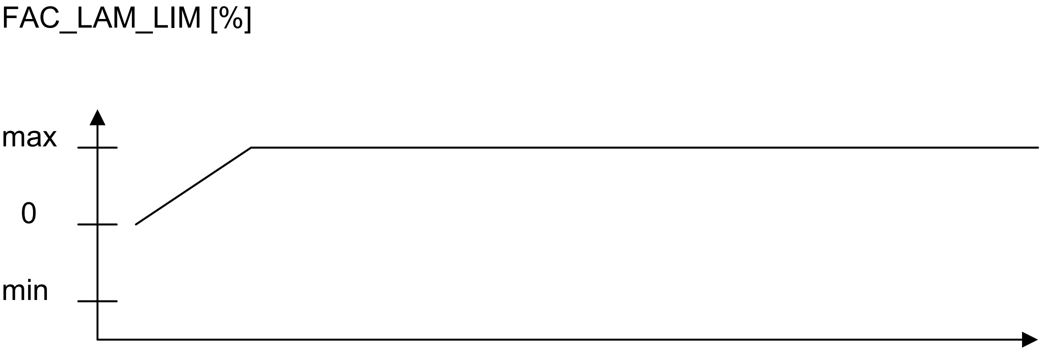
C_TEG_MIN_PLAUS_LSL_UP [NC_CBK_EX_NR]	Minimum temperature of the exhaust gas at the downstream sensor to carry out the plausibility check; The catalyst should be active to be able to smooth the downstream lambda sensor signal. I.e. 400 °C
C_TEG_MAX_PLAUS_LSL_UP[NC_CBK_EX_NR]	Maximum temperature of the exhaust gas at the downstream sensor to carry out the plausibility check; Deactivation threshold when the temperature of the

	exhaust line/exhaust gas is so hot, that the sensor signal is not anymore reliable. It might be a theoretical value in case that the sensor is reliable until the highest possible temperature. I.e. 900 °C
C_MAF_KGH_MIN_DIAG_PLAUS_LSL_UP	Minimum mass air flow to carry out the plausibility check; For very low load, the sensor signal could be like for an inactive sensor, so with this constant, low load can be excluded from the diagnosis range (i.e. IS at warm engine). i.e. 20 kg/h
C_MAF_KGH_MAX_DIAG_PLAUS_LSL_UP	Maximum mass air flow to carry out the plausibility check; To save runtime for high load. If this is not required, the threshold can be set to the max value. I.e. 500 kg/h
C_N_MIN_DIAG_PLAUS_LSL_UP	Lowest engine speed threshold to carry out the plausibility check; To define an rpm range above Idle Speed (in Idle Speed, the delay of the downstream catalyst A/F ratio is very long and the behaviour is hardly reproducible). I.e. 1000 rpm
C_TCO_MIN_DIAG_PLAUS_LSL_UP	Minimum threshold of the cooling-water temperature to activate the function; The precontrol of the engine will be better for higher TCO. As the plausibility check is however for quite strong characteristic shifts of the sensor, the TCO threshold can be chosen small, i.e. like the lambda control TCO activation threshold.
C_VLS_DOWN_AFL_PLAUS_LSL_UP[NC_CBK_EX_NR]	Lean-mixture threshold of the downstream sensor voltage; Voltage above 0 which can be easily reached even by an aged sensor.
C_VLS_DOWN_AFR_PLAUS_LSL_UP[NC_CBK_EX_NR]	Rich-mixture threshold of the downstream sensor voltage; Voltage slightly below the max. guaranteed rich voltage of sensor supplier at lambda ~ 0,9 (see sensor specification) I.e. 0,67 V
C_LAMB_AFL_PLAUS_LSL_UP[NC_CBK_EX_NR]	Lean-mixture threshold of the upstream sensor voltage; A significant rich value must be chosen, which is not met during emission test. I.e. 0,9
C_LAMB_AFR_PLAUS_LSL_UP[NC_CBK_EX_NR]	Rich-mixture threshold of the upstream sensor voltage; A significant lean value must be chosen which is not met during emission test. I.e. 1,1
C_MAF_SYM_PLAUS_LSL_UP[NC_CBK_EX_NR]	Second anti-bounce filter; In case of the need to speed up the diagnosis, this value can be reduced as it is a second debouncing filter.
C_MAF_SYM_RAW_PLAUS_LSL_UP[NC_CBK_EX_NR]	First anti-bounce filter: compensation of the catalyst delay, Force a mixture change (i.e. lambda = 0,90 to 1,10). Measure in steady state operation the delay time between sensor signal change upstream and downstream for different loads.

C_MAF_SYM_RAW_RST_PLAUS_LSL_UP[NC_CBK_EX_NR]	In case of intermittent failures, the raw symptom is not taken back immediately, but the first anti-bounce integral has to fall below this threshold.

Example for upstream lambda sensor lean shifted:





16.82 Linear Oxygen Sensor activity check

16.82.1 Calibration interfering functions

16.82.1.1 Before start of the calibration of the plausibility diagnosis, following other functions must be calibrated:

- Basic Engine Tuning (Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Wallfilm Compensation Model
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Lambda Control Parameter fixed
- Forced Stimulation
- Basic Lambda Setpoint
- Trim Control

16.82.1.2 Before calibration of the plausibility diagnosis, the following modifications on the calibration data should be done:

Sensor activity check:

For the check of the sensor activity with a maximum slow sensor, all functions which would give additional stimulation to the lambda sensor activity should be switched off:

Lambda Adaptation

Trim Control

Canister Purge

Because the target is to find the minimum activity for the system without open circuit.

16.82.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the "Calibration method" section below.

Two loops at least are required to separate the failure case from the "No failure case".

1. Failure Case
(Open Circuit on Pin Vg / Vip)

2. Ensure no failure detection in the
"no open circuit" case especially for
- low load
- slow dynamic of sensor

16.82.3 Calibration method

No activity check

The calibration can be done on the road.
A break out box is required.

For low load it's difficult to distinguish the normal sensor from the sensor with open line on Pin Vg or Vip. Therefore, some fade out criteria exist to prevent wrong diagnosis entries.

C_VLS_UP_AFS_MIN _PLAUS_LSL_UP	Lower limit of upstream sensor signal window to detect the sensor non-activity; Lowest VLS_UP_DIAG voltage which occurs in the system after open circuit on Vg **or** Vip minus ~ 10 mV (since revision K of the plausibility check, the offset correction is already considered in VLS_UP_DIAG – so just the minimum VLS_UP_DIAG must be considered). In case, that through some peaks on VLS_UP_DIAG, the plausibility diagnosis is interrupted, increase the 10 mV to any value required to ensure a stable debouncing.
C_VLS_UP_AFS_MAX _PLAUS_LSL_UP	Upper limit of upstream sensor signal window to detect the sensor non-activity; See one row above, take the highest observed VLS_UP_DIAG value in case of open circuit on Vg or Vip, consider open circuit present before engine start and open circuit created after start and after operative readiness of upstream WRAF Sensor was already finished.
C_CTR _RAW_ACT_SYM _LSL_UP	Counter threshold to set the raw symptom of sensor non-activity; Start with 5000 and check how often the counter starts to increment i.e. at low load and speed condition in the no failure case. If it is very often that the conditions for a not active sensor are met in the no failure case (i.e. low dynamic sensor in IS – worst case), then increase the limit or decrease the increment (see ID_CTR_INC_RAW_ACT_LSL_UP)
ID_CTR_INC_RAW_ACT_LSL _UP	Counter in- and decrement to detect raw symptom of upstream sensor non-activity; In principle, the counter can be increased faster for higher load as the probability for a good sensor to stay within the narrow lambda boundaries is lower for high load.
C_CTR _RAW_ACT_END _LSL_UP	Counter threshold to set end flag of first part of diagnosis in case of faultless sensor. In case of a faulty sensor, the end bit should not be set before the failure is detected. Therefore, chose threshold such that it represents the longest duration to detect a faulty sensor.
C_CTR _RAW_ACT_RST _LSL_UP	Intermittent failures shall not cause an immediate reset of the raw symptom. To reset the raw symptom, the counter has to fall below this threshold.
C_FAC_DYN_INT_ACT_LSL _UP	Relative sensor signal damping admitted to monitor the sensor activity;

	The Open Circuit diagnosis must work until the MIL would be turned on for the dynamic diagnosis. But if fuel cut off is frequently done in this project, the open circuit could also be detected via the plausibility diagnosis in PUC. In this case, the activity check could be switched off for a certain slow lambda sensor. In this case, it must be ensured that the Open Circuit diagnosis still works in the emission test.
C_N_MIN_ACT_LSL_UP	Lower engine speed to activate the sensor activity; To be able to deactivate the activity check below an engine speed where even a new sensor shows a very low activity. I.e. 600 rpm
C_T_SUM_AFL_AFR_ACT_LSL_UP	Lower forced stimulation period admitted to monitor the sensor activity; To be able to deactivate the activity check below a forced stimulation period time where even a new sensor shows a very low activity. I.e. shortest period time in the engine speed / load area which occurs during emission test.
C_LAMB_SP_DE_PLS_ACT_LSL_UP	Lower forced stimulation amplitude admitted to monitor the sensor activity; See C_T_SUM_AFL_AFR_ACT_LSL_UP.
C_MAF_KGH_MIN_ACT_LSL_UP	Lower mass airflow in Kg/h admitted to monitor the sensor activity; See C_N_MIN_ACT_LSL_UP; I.e. 20 kg/h to exclude Idle Speed for warm engine.
C_CRLC_ACT_LSL_UP	Correlation factor for the moving mean value; It must be chosen very small, to have an almost frozen mean value in case of a good lambda sensor signal. The smaller the value, the bigger the activity integral for a good sensor.
C_CTR_LAMB_DE_INT_ACT_LSL_UP	Number of forced stimulation periods to calculate the total cumulated lambda deviation; The smaller the difference is between the slow dynamic sensor and the sensor with open circuit on Vg or Vip, the longer the integration should be to get a better safety margin between good and malfunction system. I.e. 5.
C_CTR_SYM_MAX_ACT_LSL_UP	Counter threshold to set the sensor non-activity symptom flag; I.e. 3.
C_CTR_SYM_INC_ACT_LSL_UP	Counter increment to set the sensor non-activity symptom flag. For detection of intermittent errors this counter increment should be 2.
C_LAMB_DE_TOT_ACT_LSL_UP	Threshold of the total cumulated lambda deviation to debounce the sensor non-activity symptom; Install the limit slow dynamic sensor and measure the value of the activity integral at the lowest engine speed and load area where the plausibility diagnosis is still active. The constant should be chosen smaller than the result of this measurement. Open the WRAF Sensor lines Vg or Vip and observe the value for the activity integral. The constant must be bigger than the result of this measurement.
C_CTR_SYM_DEC_ACT_LSL_UP	Counter decrement to set the sensor non-activity symptom flag. For detection of intermitted errors this counter decrement should be 1.

ID_LAMB_GAIN_VLS_DOWN	Gain factor to emphasize no activity of the sensor if downstream sensor is far away from stoichiometric value. In the region around the downstream sensor set point the gain should be 1. If the downstream signal is far away from the set point, we have strong indication of no activity and the gain can be decreased below 1 to accelerate the failure detection.
ID_LAMB_GAIN_LS_UP_DYN	Gain factor to consider an aged linear lambda sensor; In case that the slow dynamic lambda sensor would result in an open circuit detection, this map can help to increase the activity integral. Take two different aging stages of the lambda sensor where the dynamic diagnosis already has a significant diagnosis value above 0. Calibrate this map to get the same activity integral (LAMB_DE_INT_ACT_LSL_UP) than for a new sensor.

16.83 Linear Oxygen Sensor dynamic diagnosis

16.83.1 Calibration interfering functions

16.83.1.1 Before start of the calibration of the dynamic diagnosis, the following other functions must be calibrated:

- Basic Engine Tuning (Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Wallfilm Compensation Model
- Evaporative Emission Control
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Lambda Control Parameter fixed
- Forced Stimulation
- Basic Lambda Setpoint
- Trim Control

16.83.1.2 Before calibration of the dynamic diagnosis, the following modifications on the calibration data should be done:

For the calibration of the minimum and maximum amplitude, all functions which would give additional stimulation should be switched off:

Lambda Adaptation

Trim Control

Canister Purge

Currently, only the synchron operation of catalyst diagnosis and dynamic diagnosis is used by the projects. The main reason is the available time during emission test.

For this mode, two ways of calibration are possible:

Make the catalyst diagnosis countinously running or copy the forced stimulation parameters from catalyst diagnosis to the separate maps for dynamic diagnosis.

The adaptation of the sensor time constant is not recommended. The PVE test would not be possible without modification of calibration data as a normal sensor is suddenly exchanged to a maximum aged sensor. So the recommendation is to take care that with the maximum aged lambda sensor, the lambda control is not too nervous so that the driveability is still ok and that the catalyst diagnosis can still be performed. If it is decided to not use the sensor time constant adaptation, it is not required to read the text below.

For the calibration of the T1_LAMB_AGI map, it is required to set the gain factors of the lambda control to 0 (just in the I-part of lambda control, 1 increment above 0 is required to keep the lambda control centered around 0).

With the adaptation of the sensor time constant, the limit sensor choice is not necessarily due to emission reasons, but can also be due to driveability impact or because the sensor reaction is “unreasonable” slow.

As during the calibration of the limit sensor amplitude, the adaptation of the sensor time constant is not yet done, the lambda control parameters must also be set to 0 (like for the calibration of the T1_LAMB_AGI map) to keep the result of lambda amplitude free from lambda control feedback.

16.83.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below.

1. C_CRLC_LPF_LAMB_DIAG_DYN_LSL_UP

2. New Sensors amplitude

3. Limit Sensors amplitude

4. Calibration of remaining items
- ✓ correlation constant
- ✓ counters
- ✓ adaptation
- ✓ failure limit

16.83.3 Calibration method

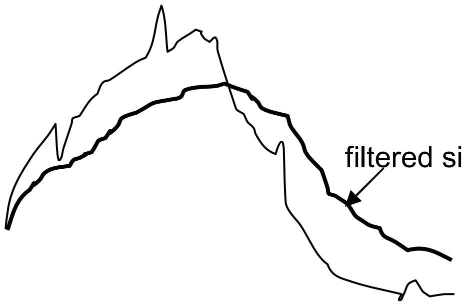
Dynamic Diagnosis

The calibration requires a chassis dyno roller.
Required parts: 1 standard new Sensor and 1 maximum aged Sensor or a Sensor with small wholes in the protection tube.
The calibration must be checked in emission test.

Hint for calibration of C_CRLC_LPF_LAMB_DIAG_DYN_LSL_UP:

good

too slow – loss of amplitude



too fast – no filter effect

LC_DIAG_DYN_LSL_UP_CAT_SYN	Boolean switch to synchronise the dynamic diagnosis with the catalyst efficiency diagnosis; I.e. 1 for synchronized processing of catalyst and dynamic diagnosis.
LC_T1_AD_DIAG_DYN_LSL_UP	Boolean switch to allow the adaptation of the sensor time constant model T1_LSL_UP[i]; I.e. 0 to forbid the adaptation.
C_CRLC_LPF_LAMB_DIAG_DYN_LSL_UP	Correlation factor for low pass filtered lambda value. Purpose is to filter out high frequent disturbances. The filtered lambda signal amplitude should not be significantly smaller than amplitude of the unfiltered lambda sensor signal. I.e. 0,3.
C_LAMB_MIN_DIAG_DYN_LSL_UP	Lowest allowed sensor signal min value; This value shall detect a shift of lambda setpoint in rich direction. Carefully handle this constant, as a too big value can lead to a significant reduction of diagnosis cycles. As during catalyst diagnosis no lambda setpoint shift is allowed, the value can be choosen similarly to C_LAMB_AMPL_MAX_DYN_LSL_UP: i.e. richest lambda during catalyst diagnosis in steady state condition (without disturbance) – 0,04.
C_LAMB_MAX_DIAG_DYN_LSL_UP	Highest allowed sensor signal max value; See C_LAMB_AMPL_MAX_DYN_LSL_UP: i.e. leanest lambda during catalyst diagnosis in steady state condition (without disturbance) + 0,04.
C_LAMB_AMPL_MIN_DYN_LSL_UP	Calibration data for adjust the expected amplitude: low range; This check can prevent a detection of a slow dynamic of upstream WRAF Sensor in case of Open Circuit on Pin Vg or Vip. As it is however not possible to run the catalyst diagnosis with open circuit conditions (lambda mean value conditions and O2 loading range not fulfilled), it can be set to maximum. With that calibration, the minimum condition will never be inhibit the diagnosis cycle.
C_LAMB_AMPL_MAX_DYN_LSL_UP	Calibration data for adjust the expected amplitude: high range; This constant is thought to exclude events with an exaggerate amplitude due to any event which disturbed the lambda control signal. But on the other hand side, it can significantly reduce the number of performed cycles. Therefore it should not be choosen too close to the standard new amplitude. The value depends on the signal noise on lambda which is very project / sensor specific. The biggest possible amplitude for a new sensor + tolerance must be choosen.
IP_LAMB_AMPL_MAX_DYN_LSL_UP	Nominal sensor expected amplitude during the Dynamic Diagnosis; Set C_CTR_MIN_DIAG_DYN_LSL_UP to 10. Adjust the relevant operation point from the map (constant on Chassis Dyno). Check LAMB_AMPL_MV_DIAG_DYN_LSL_UP[i] which is the amplitude. Enter this amplitude into the map. It is recommended to subtract a small value from the measured one to get really diagnosis result 0 for a new sensor (i.e. 0,005).

IP_LAMB_AMPL_MIN_DYN_LSL_UP	Limit sensor expected amplitude during the Dynamic Diagnosis; Fill the lambda control gain parameters with 0. Set C_CTR_MIN_DIAG_DYN_LSL_UP to 10. Adjust the relevant operation point from the map (constant on Chassis Dyno). Check LAMB_AMPL_MV_DIAG_DYN_LSL_UP[i] which is the amplitude. Enter this amplitude into the map.
C_CTR_MIN_DIAG_DYN_LSL_UP	Number of forced stimulation periods needed to calculate the short term mean value; As the engine load and speed correction is performed only once after C_CTR_MIN_DIAG_DYN_LSL_UP cycles, it should be set to 1.
C_CTR_MIN_DIAG_EOL_DYN_LSL_UP	Number of forced stimulation periods needed to calculate the short term mean value (EOL test). See C_CTR_MIN_DIAG_DYN_LSL_UP.
C_CYCNR_MIN_DIAG_DYN_LSL_UP	Minimum number of short term mean values to calculate the long term mean value; In synchronous mode: i.e. C_CTR_MIN_DIAG_DYN_LSL_UP * C_CYCNR_MIN_DIAG_DYN_LSL_UP should be 1/2 of the required catalyst diagnosis cycles.
C_CYCNR_EOL_DIAG_DYN_LSL_UP	Number of long term mean values to deliver the sensor dynamic diagnostic (EOL test); Number of possible dynamic diagnosis cycles for the end of line test. Customer specific.
C_CYCNR_END_DIAG_DYN_LSL_UP	Number of long term mean values to deliver the sensor dynamic diagnostic; End of dynamic diagnosis in synchronous mode triggered either by end of catalyst diagnosis or by number of cycles bigger than this treshold. I.e. 20.
C_FAC_AD_THD_DIAG_DYN_LSL_UP	Diagnosis value threshold for begin adaptation of sensor time constant model. As for the time beeing, the lambda control will force the amplitude of the sensor to the sensor model, an adaptation cycle would start as soon as the sensor model was adapted one time (i.e. by a measured amplitude which is slightly smaller than the max value). This is especially true for a fast (new) sensor which reacts fast to the lambda control. Therefore this threshold should be set to a significant value like i.e. 0,5.
C_FAC_AD_RST_DIAG_DYN_LSL_UP	Diagnosis value threshold to reset adaptation of sensor time constant model; This value must be smaller than C_FAC_AD_THD_DIAG_DYN_LSL_UP to realize a hysteresis. I.e. 0,1.
C_CRLC_LPF_FAC_AD_DYN_LSL_UP	Correlation factor for low pass filtered (sensor time constant) adaptation factor; I.e. 0,3.

C_FAC_MV_THD_DIAG_DYN_LSL_UP	Limited diagnosis value for the error diagnosis. This treshold should be determined with the slow dynamic sensor which causes either a driveability impact or an exceeding of emission threshold in emission test. Due to the adapted sensor time constant, it can be, that just the availability of a even slower sensor is not given (i.e. just a 0,2 mm whole in the protection tube can be still ok for emissions and driveability). In that case, the choice can be done just with the reason that the sensor is “unreasonable” slow
C_CTR_WAIT_DIAG_DYN_LSL_UP	Number of forced stimulation periods during the diagnosis status transition “wait” to “acquisition”; For the unsynchronous mode only. I.e. 2.
C_N_MIN_DIAG_DYN_LSL_UP	Minimum engine speed to activate the diagnosis; Only for unsynchronous mode. Choose values in relation to minimum load while steady state operation in emission test.
C_N_MAX_DIAG_DYN_LSL_UP	Maximum engine speed to activate the diagnosis; Only for unsynchronous mode. Choose values in relation to maximum load while steady state operation in emission test.
C_MAF_MAX_DIAG_DYN_LSL_UP	Maximum mass airflow rate to activate the diagnosis function; Only for unsynchronous mode. Choose values in relation to maximum load while steady state operation in emission test.
C_MAF_MIN_DIAG_DYN_LSL_UP	Minimum mass airflow rate to activate the diagnosis function; Only for unsynchronous mode. Choose values in relation to minimum load while steady state operation in emission test.
IP_T_DYN_DIAG	Time in one lean/rich cycle during which the pointer for maximum and minimum amplitude is updated; This map shall include the time for a new sensor to reach the maximum amplitude + 10% of the complete cycle time. The target is to make the diagnosis distance between new and aged sensor bigger in the low engine speed / load operation area where even the maximum slow sensor reaches the maximum amplitude. Measure LAMB_AMPL_MV_DIAG_DYN_LSL_UP (with C_CTR_MIN_DIAG_DYN_LSL_UP = 10 and C_CYCNR_MIN_DIAG_DYN_LSL_UP = 1) for every relevant operating point with IP_T_DYN_DIAG = IP_T_PER_LAMB_DE_CAT_DIAG (for synchronous operation) * 1, * 0,6, * 0,5, *0,45, * 0,4, * 0,35, * 0,3, * 0,25; Do that measurement for a new sensor and for an old sensor. Compare the result for LAMB_AMPL_MV_DIAG_DYN_LSL_UP for both sensors and every relevant operation point. Choose the value for IP_T_DYN_DIAG according to the factor on IP_T_PER_LAMB_DE_CAT_DIAG which gave almost no decrease of amplitude for the new sensor but already a significant reduction of amplitude for the limit aged sensor.

16.84 Linear Oxygen Sensor on Air diagnosis

16.84.1 Calibration interfering functions

16.84.1.1 Before start of the calibration of the Sensor on Air diagnosis, following other functions must be calibrated:

- Intake Manifold Model (MAP System)
- Temperature Model upstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of upstream lambda sensor

16.84.1.2 Before calibration of the Sensor on Air diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.84.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below.

16.84.3 Calibration method

Sensor on Air Diagnosis

The calibration can be done on street.
A sensor which is not mounted must be connected to the ECU (i.e. with breakout box or directly on the connector).

C_VLS_UP_DIAG_MAX_AIR_LSL_UP	Upstream sensor signal max. threshold during engine operation; Minimum voltage for the sensor on air out of the sensor specification (similar to the “low” threshold in the function lambda sensor plausibility diagnosis during fuel cut off - C_VLS_UP_DIAG_MIN_PUC_LSL_UP).
C_T_SYM_DIAG_AIR_LSL_UP	Time filter to detect a symptom; I.e. 25 sec.
C_T_END_DIAG_AIR_LSL_UP	Time to have a diagnostic available; I.e. 100 sec.

16.85 Linear Oxygen Sensor plausibility diagnosis during fuel cut off

16.85.1 Calibration interfering functions

16.85.1.1 Before start of the calibration of the WRAF Sensor plausibility diagnosis during fuel cut off, following other functions must be calibrated:

- Temperature Model upstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of upstream lambda sensor

16.85.1.2 Before calibration of the WRAF Sensor plausibility diagnosis during fuel cut off, the following modifications on the calibration data should be done:

Nothing.

16.85.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below.

16.85.3 Calibration method

WRAF Sensor plausibility diagnosis during fuel cut off

The calibration can be done on street.
But it is more simple on Chassis Dyno Roller.
The calibration must be checked in emission test.

C_T_DIAG_SYM_PUC_LSL_UP	Time filter to detect a symptom; Accumulated time in a driving cycle where the too high or low signal must be present to enter an error. This is very dependent from the possible fuel cut off time during the emission test.
C_T_DIAG_SYM_EOL_PUC_LSL_UP	Time filter to detect a symptom (EOL test); The debounce time can be choosen smaller to make the test possible at the end of line.
C_T_DLY_DIAG_PUC_LSL_UP	Waiting time after beginning of PUC to test sensor signal; Choose a low engine speed and make fuel cut off. Measure the time delay which allows the sensor to show the lean signal without further increasing. Check this also with a dynamically slow sensor.
C_T_DLY_DIAG_EOL_PUC_LSL_UP	Waiting time after beginning of PUC to test sensor signal (EOL test); See C_T_DLY_DIAG_PUC_LSL_UP.
C_VLS_UP_DIAG_AFS_PUC_LSL_UP	Upstream sensor output voltage in PUC corresponding to Open Circuit on WRAF Sensor; Corresponding to maximum possible VLS_UP_DIAG threshold in case of open circuit on Vg or Vip. Take this VLS_UP_DIAG and add 50 mV.

C_VLS_UP_DIAG_MIN_PUC_LSL_UP	Minimum admissible upstream sensor output voltage in PUC; Corresponding to a sensor specific voltage which must be calculated according to the sensor specification, including the tolerance chain from the Engine Control Unit. – e.g. 3,1 V.
C_VLS_UP_DIAG_MAX_PUC_LSL_UP	Upstream sensor output voltage in PUC corresponding to Open Circuit on WRAF Sensor; Corresponding to a sensor specific voltage which must be calculated according to the sensor specification, including the tolerance chain from the Engine Control Unit. – e.g. 4,9 V.
C_TEG_MIN_DIAG_PUC_LSL_UP	Minimum admissible exhaust gas temperature in PUC. To ensure no impact from a cooled down sensor. I.e. 300 °C.
C_T_DIAG_END_PUC_LSL_UP	Time to have a diagnosis available; Choose a time which is shorter than the available fuel cut off time in the emission test cycle.
C_T_DIAG_END_EOL_PUC_LSL_UP	Time to have a diagnosis available (EOL test); Choose a time which is shorter than the available fuel cut off time during the end of line test.

16.86 Linear Oxygen Sensor Heater Coupling Diagnosis

16.86.1 Calibration interfering functions

16.86.1.1 Before start of the calibration of the heater coupling diagnosis, following other functions must be calibrated:

- Basic Engine Tuning (Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Wallfilm Compensation Model
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Lambda Control Parameter fixed
- Forced Stimulation
- Basic Lambda Setpoint
- Trim Control
- Catalyst Purge

16.86.1.2 Before calibration of the heater coupling diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.86.2 Calibration flowchart

The sequence is described by the order of constants and maps in the chapter “Calibration method” below.

16.86.3 Calibration method

The calibration shall be done on chassis dynamometer.
A WRAF Sensor break out box and a Resistor Decade is required.
The calibration must be checked in emission test.

Determination of min. possible heater coupling resistance:

1. Connect the break out box between sensor and ECU plugs. The heater coupling failure can be simulated installing a variable resistor between the UN and the negative potential of Uheater – lines.
2. Adjust C_LAMB_DELTA_LPF_THD, C_CTR_SYM_DIAG_LSH_LSL_UP and C_CYCNR_MAX_DIAG_LSH_LSL_UP to their maximum values.
3. Carry out emission tests with different resistance values (i.e. 1 Mohm, 700Kohm, 300K, 200K and 100K) in order to determine, in which case the emissions are near the threshold defined by the legislation.

C_LAMB_SP_DE_DIAG_LSH_LSL_UP	Maximum value of the forced excitation amplitude to activate the diagnosis; I.e. maximum value.
C_TEG_MIN_DIAG_LSH_LSL_UP	Minimum value of the exhaust temperature to activate the function. The colder the lambda sensor environment is, the bigger can the heater coupling be. To see the impact, check heater coupling diagnosis result right after operative readiness after coldstart and see it with warmed up engine. I.e. 350 °C.
C_MAF_INT_DIAG_LSH_LSL_UP	Threshold for the activation of the MAF-Integral; 1. Adjust C_MAF_INT_DIAG_LSH_LSL_UP equal “0” to activate the function. 2. Measure some changes from PUC to PL. Evaluate the results of DELTA_LAMB_SP_FALL/RISE; i.e. at the transition from PUC <-> PL evaluate if the diagnosis value is greater than zero. Vary C_MAF_INT_DIAG_LSH_LSL_UP until no influence of PUC phase can be found. 3. Calibrate C_MAF_INT_DIAG_LSH_LSL_UP with the maximum measured value. I.e. 25 g.
C_T_AFL_AFR_DIAG_LSH_LSL_UP	Maximum forced stimulation cycle time to activate the heater-coupling diagnosis; I.e. maximum.
C_CRLC_LPF_FALL_DIAG_LSH_LSL_UP	Filter constant for Fall; I.e. 0,5.
C_CRLC_LPF_RISE_DIAG_LSH_LSL_UP	Filter constant for Rise; I.e. 0,5.
C_LSHPWM_MIN_DIAG_LSH_LSL_UP	minimum heater PWM value allowed to carry out the heater coupling diagnosis; The calibration depends on the delay time, calibrated in IP_T_DLY_TRIG_LSH_UP. If the counterflank of the currently active flank comes before the delay time is elapsed, the VLS_UP_RISE / FALL cannot be taken correctly. I.e. if the delay time is 6 ms, the minimum off respectively on time for the PWM is 6 + 2 (for polling) ms. For a heater PWM frequency of 10 Hz., 8 ms would correspond to 8 %. I.e. 10 %.
C_LSHPWM_MAX_DIAG_LSH_LSL_UP	maximum heater PWM value allowed to carry out the heater coupling diagnosis; See comment for minimum threshold above. I.e. 90 %.
IP_T_DLY_TRIG_LSH_UP	Time delay to trigger oxygen sensor signal and lambda model; Simulate the heater coupling with a resistor of ~ 500 kohm. Vary the delay time to obtain the highest value of CTR_SYM_DIAG_LSH_LSL_UP[i]. Repeat this procedure by different heater PWM values. I.e. 0,006 sec.

C_LAMB_DELTA_LPF_THD	Threshold for the maximum transition of the sensor signal; 1. Use the chassis dyno, adjust the variable resistor to the limit value to be detected. 2. Vary the load/engine speed according usual conditions during emission test cycle. At each condition proceed as described below. 3. Set C_CYCNR_MAX_DIAG_LSH_LSL_UP to a reasonable value, for example 50. 4. Observe if LAMB_DELTA_RISE/FALL_LPF[i] are relatively stable, otherwise increase C_CYCNR_MAX_DIAG_LSH_LSL_UP 5. Set C_LAMB_DELTA_LPF_THD to the value of LAMB_DELTA_RISE/FALL_LPF[i]. Evaluate the dispersion during each measurement. If the values are too low, check the calibration of map IP_T_DLY_ACQ_VLS_UP. 6. Observe the value of CTR_SYM_DIAG_LSH_LSL_UP[i]. 7. Set C_LAMB_DELTA_LPF_THD to the lowest value obtained over all load/engine conditions. I.e. 0,02.
C_CYCNR_MAX_DIAG_LSH_LSL_UP	Quantity of valid diagnosis values in one function cycle; See C_LAMB_DELTA_LPF_THD. I.e. 100.
C_CTR_SYM_DIAG_LSH_LSL_UP	Limit value of the diagnosis results to diagnose the heater coupling; See C_LAMB_DELTA_LPF_THD. I.e. 50.

16.87 Oxygen Sensor Downstream Diagnosis

16.87.1 Calibration interfering functions

16.87.1.1 Before start of the calibration of the OBD2 downstream oxygen sensor diagnosis, following other functions must be calibrated:

- Basic Engine Tuning (i.e. Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Wallfilm Compensation Model
- Evaporative Emission Control
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Lambda Control Parameter fixed
- Forced Stimulation
- Basic Lambda Setpoint
- Trim Control
- Catalyst Purge

16.87.1.2 Before calibration of the OBD2 downstream oxygen sensor diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.87.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below.

16.87.3 Calibration method

The calibration can be done on the road.
The calibration must be checked in emission test.
The diagnosis for switching time of the downstream sensor is responsible to ensure a fast enough switching to detect a possible catalyst damage.
It is very recommended to activate it.
If the vehicle has no or not sufficient fuel cut off phases, a possible deterioration of sensor switching must be detected with the fuel trim diagnosis which will not always be possible.
Conclusion is, that a fuel cut off phase is required for a proper diagnosis of the monitor sensors.

Application Conditions

C_TCO_MIN_DIAG_MPL_LS_DOWN	Minimum coolant temperature for diagnosis of the monitor sensors; The operative readiness for the downstream sensor is activated in an early stage. If after the operative readiness a fuel cut off follows, the sensor temperature could still be low and the dynamic low. Therefore its recommended to set the temperature according to a slightly warmed up engine. I.e. 50 °C.

C_VS_MIN_DIAG_MPL_LS_DOWN	Minimum vehicle speed for diagnosis of the monitor sensor;s; There is no special reason to exclude low vehicle speed. I.e. 0 km/h.
C_VS_MAX_DIAG_MPL_LS_DOWN	Maximum vehicle speed for diagnosis of the monitor sensors. There is no special reason to exclude high vehicle speed. It must be ensured that the temperature of the downstream sensor is still in the specified range for high speed / high load operation. Check the diagnosis behaviour for very high load and speed. I.e. 255 km/h (max.).
C_TEMP_CAT_MIN_DIAG_MPL_LS_DOWN	Catalyst temperature threshold for diagnosis of the monitor sensors; See explanation for C_TCO_MIN_DIAG_MPL_LS_DOWN. I.e. 350 °C.

Monitoring the rich-lean switching times in the trailing throttle fuel cut-off

C_MAF_PUC_MIN_DIAG_SWT_LS_DOWN	Mass air flow threshold for diagnosis of the switching times; Measure the behaviour of the switching time for the lowest MAF_KGH during PUC (low engine rpm). If that is still acceptable, adjust the constant to a value slightly smaller than the smallest MAF_KGH during PUC in emission test. In case that there is a bad behaviour of the switching time, check if there are fuel cut off phases during emission test with higher MAF_KGH and select the value slightly below this higher MAF_KGH.
C_MAF_PUC_MAX_DIAG_SWT_LS_DOWN	Mass air flow threshold for diagnosis of the switching times; To inhibit the switching time diagnosis for fuel cut off phases with high engine speed. Check the behaviour of the downstream diagnosis at entry to fuel cut off at high engine speed. If there is no problem, choose a MAF_KGH which corresponds to an engine speed of i.e. 5000 rpm.
C_VLS_PUC_MIN_DIAG_SWT_LS_DOWN	Voltage threshold for monitor sensor diagnosis; Minimum VLS_DOWN to start the diagnosis at start of fuel cut off phase. Subtract approximately 50 mV from the lowest fuel trim setpoint which occurs during emission test.
C_FAC_VLS_MAX_DIAG_SWT_LS_DOWN	Reduction factor for start of switching time determination; This factor should be small to avoid an unintended start of the diagnosis, i.e. by wallfilm effects due to release of the accelerator pedal. I.e. 0,6.
C_FAC_VLS_MIN_DIAG_SWT_LS_DOWN	Reduction factor for end of switching time determination; I.e. 0,1.
C_VLS_HYS_DIAG_SWT_LS_DOWN	Monitor sensor voltage hysteresis; Measure the downstream lambda sensor voltage at entry into fuel cut off in 10 ms recurrence at low load. Check the maximum peaks which occure due to cylinder to cylinder richness deviation. Take this as hysteresis, so that the diagnosis can run even with this (small peaks).
C_CTR_SWT_DIAG_LS_DOWN	Condition for the start of the lambda sensor switching time check; Measure the number of possible diagnosis cycles during emission test and chose the constant a little smaller. The minimum number of cycles should however not fall short of 3.
C_CTR_SWT_LS_DOWN	Diagnostic threshold for CTR_CYCNR_SWT_LS_DOWN[NC_CBK_EX_NR]; The map IP_FAC_CYCNR_MAX_LS_DOWN_ACT can be calculated in a way, that the failure threshold can be set to 1.

IP_FAC_CYCNR_MAX_LS_DOWN_ACT	Weighting factor for CTR_SWT_ACT_LS_DOWN[NC_CBK_EX_NR] Measure the standard switching time for a new sensor for different MAF_KGH. Take oxygen sensors with small wholes in the protection tube and equipp the vehicle with a limit catalyst which must be detected as malfunctioning. The oxygen sensor which is too slow for a good catalyst diagnosis result can be chosen as limit sensor. Measure the switching time for this sensor for different MAF_KGH. Adjust the map that the limit sensor results in a diagnosis result of 1.

Monitoring the sensor voltage after leaving trailing throttle fuel cut-off

C_MAF_INT_PUC_DIAG_PUE_LS_DOWN	Threshold for activation of the diagnosis; The duration of the fuel cut off phase must be long enough to ensure the fuel cut off voltage level of 0,00 V. Otherwise the increase of VLS_DOWN after end of PUC would not be detected and a failure is entered. Check this with a new catalyst.
C_VLS_DIF_MIN_DIAG_PUE_LS_DOWN	Threshold for VLS_DIF_DIAG_PUE_LS_DOWN[NC_CBK_EX_NR]; Measure the standard voltage right after PUC (after the first movement of the downstream sensor which comes immediately after reactivation of injection). Chose a value in the middle between this voltage and 0 V.
C_MAF_INT_DIAG_PUE_LS_DOWN	Threshold for start of the diagnosis; Take a new catalyst system (not green but with maximum O2 load capacity). Measure the MAF integral which is required for the limit slow Oxygen Sensor (see switching time diagnosis) to reach the first voltage plateau right after reactivation of injection. Multiply this with a safety factor of i.e. 2 or more (the smaller the MAF integral, the higher the safety factor must be). If the switching time diagnosis is activated as well an the fuel trim diagnosis is calibrated well, the MAF integral can be chosen much higher. In this case, the PUE diagnosis covers only the detection of a stuck sensor (this symptome would also be detected by fuel trim diagnosis later on depending on driving style).
C_CTR_SYM_INC_DIAG_PUE_LS_DOWN	Counter increment for PUE_VLS_DOWN diagnosis; Statistical evaluation, i.e. 1.
C_CTR_SYM_MAX_DIAG_PUE_LS_DOWN	Maximum value of counter for PUE_VLS_DOWN diagnosis; Statistical evaluation, i.e. 2. That means that in case of 2 malfunction events out of 3 diagnosis cycles, an error is entered.
C_CTR_END_DIAG_PUE_LS_DOWN	Counter threshold for CTR_END_DIAG_PUE_LS_DOWN_[NC_CBK_EX_NR]; Measure the number of possible diagnosis cycles during emission test and chose the constant a little smaller. The minimum number of cycles should however not fall short of 3.

16.88 O2 sensor (lin, up) dynamic diagnosis after NOx cat regeneration

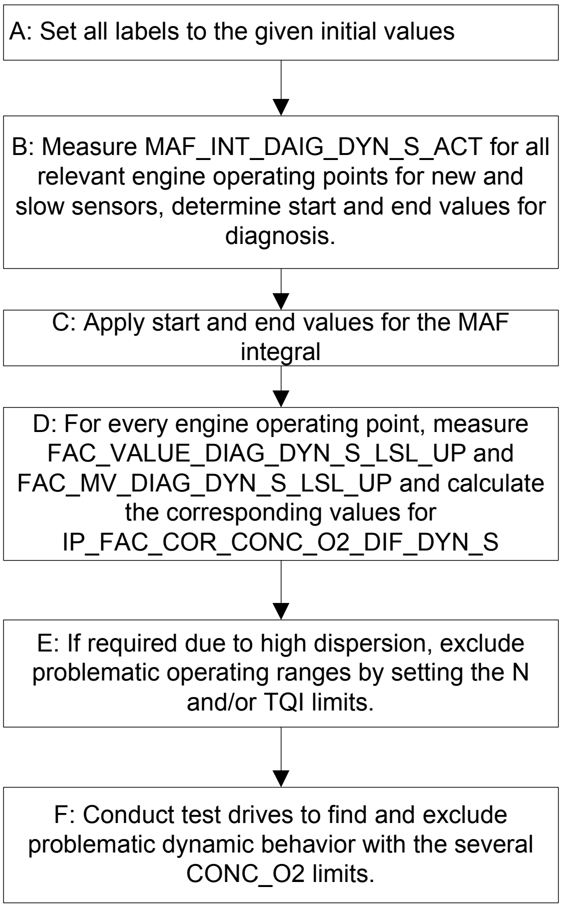
16.88.1 Calibration interfering functions

The following functions must be working before the operability detection can be calibrated:

- Intake manifold model
- lambda setpoint determination, including sensor model, coordination of lambda setpoint requests and lambda calculation for stratified combustion
- Engine operation functions, including combustion management
- NOx trap regeneration strategy
- Lambda sensor must be working and signal acquisition and correction must be fully calibrated.

16.88.2 Calibration flowchart

The following steps should be executed in the order shown:



16.88.3 Calibration method

16.88.3.1 Resources

Slow dynamics sensors are required, both a "good" limit sample and a "bad" sample. For calibration of MAF integral limits and correction maps, an engine or chassis dyno is required. Calibration of constraints should be done on the road with dynamic driving patterns.

16.88.3.2 Data overview (Step A)

The following table gives an overview of the data values. Initial data should be applied before calibration is started, they make sure that the diagnosis is not disabled by any means and that the required data can be recorded. The data will be changed during the calibration process.

Label	Initial value	Expected range
C_CONC_O2_CLC_DEC_MAX_DYN_S	1	<0.02
C_CONC_O2_CLC_HOM_MAX_DYN_S	1	<0.05
C_CONC_O2_CLC_MIN_DIAG_DYN_S	0	>0.05
C_CONC_O2_DIF_MAX_DIAG_DYN_S	1	<0.06
C_CONC_O2_DIF_ST_MAX_DIAG_DYN_S	1	<0.05
C_CRLC_CONC_O2_CLC_LPF_DYN_S	0.1	0.01...0.2
C_CTR_DIAG_DYN_S_LSL_UP	5	1...10
C_FAC_CONC_O2_COR_DIAG_DYN_S	1	0.9...1.1
C_FAC_DIAG_DYN_S_LSL_UP_MAX	127.996	Depends on other values and dynamic limit sensors
C_MAF_INT_ST_DIAG_DYN_S	0g	0...20g
C_MAF_INT_END_DIAG_DYN_S	1000g	20...100g
C_N_MIN_STOP_DIAG_DYN_S	0 rpm	0...1500
C_N_MAX_STOP_DIAG_DYN_S	8160 rpm	- in stratified combustion range -
C_TQI_MIN_DIAG_DYN_S	0 Nm	>0
C_TQI_MAX_DIAG_DYN_S	1000 Nm	- in stratified combustion range -
LC_DIAG_DYN_S_LSL_UP_MOD	1	Tbd
IP_FAC_COR_CONC_O2_DIF_DYN_S	0.03	0.01...0.3, depending on the MAF integral thresholds

16.88.3.3 Calibration of MAF integral thresholds (Step B/C)

Measure MAF_INT_DIAG_DYN_S_ACT[i], CONC_O2_CLC_DIAG_DYN_S[i] and CONC_O2_MES_DIAG_DYN_S[i] for the several engine operating points for both slow and fast sensors. After a regeneration phase, the CONC_O2 values will rise more or less slowly from around 0 to the steady state for stratified combustion.

C_MAF_INT_ST_DIAG_DYN_S	**Set to a MAF_INT_DIAG_DYN_S_ACT value where the CONC_O2_MES_DIAG_DYN_S value for a new sensor has (almost) reached its steady state for all relevant operating points. Setting this too low will increase dispersion for good sensors.**

C_MAF_INT_END_DIAG_DYN_S	**Set a bit lower than the lowest MAF_INT_DIAG_DYN_S_ACT value for which the CONC_O2_MES_DIAG_DYN_S value for a bad sensor has reached its steady state value for any operating point. Setting this too high will increase the influence of slight gain errors in the lambda sensor or in the realization of the fuel mass set point.**

16.88.3.4 Calibration of the weighting map and diagnostic limit (Step D/E)

LC_DIAG_DYN_S_LSL_UP_MOD is used to select one of two methods to calculate the diagnostic result. If set to 1, the diagnostic value more or less resembles the T1 constant of the PT1 behavior of the sensor, but the dispersion may be high for slow sensors. If set to 0, the value is more stable. If the flag is altered, all subsequent steps must be repeated.

For every operating point and with the bad sensor, clear the error memory and let the engine run until LV_DIAG_DYN_S_LSL_UP_END = 1 is reached. Record CONC_O2_DIF_INT_DIAG_DYN_S, FAC_VALUE_DIAG_DYN_S_LSL_UP and FAC_MV_DIAG_DYN_S_LSL_UP. If CONC_O2_DIF_INT_DIAG_DYN_S hits its limit, lower the initial values in the weighting map and start over.
If FAC_VALUE_DIAG_DYN_S_LSL_UP shows high dispersion already on the engine or chassis dyno, the corresponding operating points should be excluded via C_N_MIN/MAX_STOP_DIAG_DYN_S and C_TQI_MIN/MAX_DIAG_DYN_S. The diagnosis should also be blocked in the homogenous stratified combustion range, because the axes for the weighting map are N_32 and MAF_CYL and they uniquely describe an operating point only in one combustion mode.

After collecting the data, calculate the correct factors for the weighting map: Select a target diagnostic value X for the bad sensor and calculate the values from the current IP_FAC_COR_CONCO2_DIF_DYN_S$_{\mathrm{old}}$ values as

$$\mathrm{IP\_FAC\_COR\_CONC\_O2\_DIF\_DYN\_S}_{\mathrm{new}} = \frac{X \cdot \mathrm{IP\_FAC\_COR\_CONC\_O2\_DIF\_DYN\_S}_{\mathrm{old}}}{\mathrm{FAC\_MV\_DIAG\_DYN\_S\_LSL\_UP}_{\mathrm{old}}}$$

$$\mathrm{IP\_FAC\_COR\_CONC\_O2\_DIF\_DYN\_S}_{\mathrm{old}} \cdot \frac{\mathrm{CONC\_O2\_DIF\_INT\_DIAG\_DYN\_S}_{\mathrm{lim}}}{\mathrm{CONC\_O2\_DIF\_INT\_DIAG\_DYN\_S}_{\mathrm{old}}}$$

but make sure that the factor is not higher than
with CONC_O2_DIF_INT_DIAG_DYN_S$_{\mathrm{lim}}$ being the physical limit of the variable and CONC_O2_DIF_INT_DIAG_DYN_S$_{\mathrm{old}}$ the highest value found for the current operating point with the old IP value. If this limit is violated, the integrals may saturate to their physical limits for a bad sensor, giving too low diagnostic results. Select a lower target value X in this case. If there is room for a higher value, the value should be increased, as this can only improve accuracy.

The diagnostic limit C_FAC_DIAG_DYN_S_LSL_UP_MAX should be a bit lower than the target X to make sure that bad sensors are detected reliably.

16.88.3.5 Calibration of engine dynamics constraints (Step F)

The integrals in the diagnosis may take a few seconds for low engine loads. If the load changes rapidly while the integral runs, the result may be wrong. The following labels are available to block or abort the diagnosis in such conditions:

C_CONC_O2_CLC_DEC_MAX_DYN_S	Detects too high gradients in engine load, which in stratified mode usually lead to a rapid decrease in O_2 concentration and thus violate this limit.
C_CONC_O2_CLC_HOM_MAX_DYN_S	Maximum modeled O_2 concentration in homogenous mode, useful to exclude very short homogenous phases.
C_CONC_O2_CLC_MIN_DIAG_DYN_S	Minimum for the modeled and low pass filtered O_2 concentration in stratified mode, may be used to exclude too high engine load.
C_CRLC_CONC_O2_CLC_LPF_DYN_S	Used to do low pass filtering on modeled O2 concentration for the above limit. Decrease the value to increase the filtering effect.
C_CONC_O2_DIF_MAX_DIAG_DYN_S	Maximum difference between measured and modeled O_2 concentration, may be used to exclude instationary conditions.
C_CONC_O2_DIF_ST_MAX_DIAG_DYN_S	Same as above, but for the moment of homogenous-to-stratified transition, useful to exclude homogenous phases that are too short for slow sensors.
C_CTR_DIAG_DYN_S_LSL_UP	Minimum number of samples in average. Must reliably fit into the test driving cycle, but must be high enough to sufficiently reduce dispersion.
C_FAC_CONC_O2_COR_DIAG_DYN_S	May be set >1 to reduce the diagnostic value for good sensors and improve differentiation between good and bad sensors

16.89 O_2 sensor heater management

16.89.1 Upstream oxygen sensor heater management

16.89.1.1 Calibration interfering functions

16.89.1.1.1 Before start of the calibration of the upstream oxygen sensor heater management, following other functions must be calibrated:

- Basic engine tuning (ignition, basic injection, torque model, variable valve timing)
- Intake manifold model (MAP/MAF system)
- Exhaust gas temperature model and dew point recognition upstream catalyst
- Internal resistance determination (oxygen sensor temperature for linear lambda sensors)
- Signal acquisition of linear lambda sensor (controller, for linear lambda sensors only)
- Readiness detection of upstream lambda sensor
- Basic Lambda Setpoint

16.89.1.1.2 Before calibration of the upstream oxygen sensor heater management, the following modifications on the calibration data should be done:

Nothing.

16.89.1.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below. Parameters which are relevant for cold start may be calibrated independently on the remaining parameters for heater control and sensor protection.

Remark: If the same N_32 axis points are used for the interpolation maps IP_V_EFC_CTL_LSH_UP / IP_V_EFC_CTL_LSH_DOWN the calibration of the precontrol maps upstream and downstream can be done at the same time.

16.89.1.3 Calibration method

Calibration under cold start and hot restart conditions

The calibration shall be done in a climate chamber using a lambda sensor with tip-temperature measurement capability (TTS).

IP_V_EFC_INI_LSH_UP	Initial effective heater voltage. This map is dependent on the inital sensor temperature TEMP_INI_LS_UP[i] (differentiation between cold start and hot restart) and has main influence on the maximum temperature rise rates during POW_RISE. For calibration the following aspects are important: 1) The sensor specification for maximum intial voltage must be fulfilled.

2) Calibration with too large map entries may lead large temperature rise rates which violate the sensor specification. This has to be avoided.
3) In combination with too large map entries of IP_T_TEMP_THD_LSH_UP the maximum allowable tip temperature – if available – may exceed the specified limit before dew point recognition. This has to be avoided.
4) The data should be calibrated to achieve the lowest possible light-off time under consideration of 1-3.

The calibration should at best be done in a climate chamber in combination with TEMP_INI_LS_UP[i], IP_T_TEMP_THD_LSH_UP and C_V_EFC_INC_ LSH_UP using aTTS with strong limit heater (critical for temperature rise rates). On the other hand the lowest temperature during dew-point recognition should exceed the recommendation in the specification – if available – for weak limit heater.

IP_T_TEMP_THD_LSH_UP	Duration until critical water splash sensor temperature (typ. < 350°C, see sensor specification) reached under any conditions. Check must be done with TTS. Check on chassis dyno is not necessary. Calibration has to be done in accordance with IP_TEMP_INI_LS_UP
C_T_MAX_POW_RISE_LSH_UP	Maximum allowed time for state POW_RISE. Calibration has to be done with TTS. Under all conditions tip temperature has to reach a minimum temperature and should not exceed the maximum operating temperature (see sensor specification for operating range of temperature). Cirtical edge is of course warm start with long heating for maximum allowed POW_RISE time. E. g. 14s.
C_V_EFC_INC_LSH_UP	Effective heater voltage increment to raise temperature. Calibration shall be done under consideration of the maximum specification value. In general a larger value may shorten light-off time but may also increase the temperature rise rate above the specified limit (thermal stress). Too large values may also lead to a violation of the maximum specified temperature threshold unitl dew-point detection. The increment of heating voltage must be determined with a TTS sensor. In general smaller values are recommended. E.g. 0.4V/s.
C_V_EFC_DEC_LSH_UP	Effective heater voltage decrement to lower temperature. Calibration shall be done under consideration of the maximum value in the sensor specification. If no value is specified the decrement can be choosen same as increment. Because of risk for high thermal stress smaller values are recommended. E.g. 0.4V/s.
C_V_EFC_RED_LSH_UP	Reduced effective heater voltage during danger of water splash damage. Calibration should be done such that the temperature is kept below the specified value (<300°C for most sensors, with safety margin). For calibration a TTS with strong limit heater is recommended. E.g. 4V. Remark: According to sensor specification it may be required that until dew point recognition the effective heater voltage does not exceed a certain level instead of a certain temperature (see LC_TEMP_DEW_LSH_UP).
C_V_EFC_STEP_LSH_UP	Additional effective heater voltage step to IP_V_EFC_ INI_LSH_UP on transition from LSH_POW_RED to LSH_POW_RISE. The calibration has to be done with

	TTS if the data is not specified in the sensor specification anyway. Calibration of this data has to take into account the maximum permitted temperature gradient, if it is mentioned within sensor specification. E.g. 1V.
LC_TEMP_DEW_LSH_UP	Calibration of this data equal to 1 allows to skip the initial transition of the heater state machine from LSH_OFF to LSH_POW_RISE before entering LSH_POW_RED if dew point has not been recognized yet. Setting this logical calibration data is required to fulfill component specifications where a maximum effective heater voltage before dew point recognition is demanded. In other cases in which a maximum temperature (e.g. 350°C) until dew point determination is recommended this data should be calibrated equal to 0, i.e. the state machine enters LSH_POW_RISE after engine start in any case.

Calibration of lambda sensor heater control

The calibration shall be done on a chassis dyno. For binary lambda sensors the tip-temperature should be measured using a sensor with temperature measurement capability (TTS) or by evaluating the results of the internal resistance determination. For linear lambda sensors, the most accurate tip temperature is the one measured from the system; therefore it is best to rely on this temperature. After calibration the results should be checked under real conditions on road because of possible heat accumulation on chassis dyno.

IP_V_EFC_CTL_LSH_UP	Open loop control effective heater voltage. The map entries can be determined best on chassis dyno, as it is very sensible to geometry of the exhaust gas system. For each load as characterized by exhaust gas temperature and engine speed the necessary heating voltage to reach the setpoint temperature has to be determined. Cooling of the fan should be adjusted to medium level. If the system has two exhaust gas banks which differ in geometry just focus on bank 1 for calibration. For bank 2 see IP_FAC_V_EFC_TEG_ LSH_UP. For linear lambda sensor set the setpoint and observe the additive part of the lambda sensor heater power (V_EFC_CTL_ADD_LSH_UP[i]). Adjust IP_V_EFC_ CTL_LSH_UP according to an additive part of 0 V. The TTS Sensor measures up to 80 °C lower temperature than the measurement through the ECU. It is mainly useful for preheating. For binary sensors the setpoint temperature can be controlled by two ways: either by using a TTS or by evaluation of the determined internal resistance values and the corresponding conversion to element temperature (see engineering specification). For set-point temperature please refer to engineering specification. E.g. 650°C.
IP_FAC_V_EFC_TEG_LSH_UP	Factor for the heat transfer rate, bank 2 only. This interpolation map reflects the difference in geometry between the exhaust gas banks due to, e.g., different cooling by airstream. Tests are done best on road (highway etc.)

C_V_EFC_AS_LSH_UP	Effective heater voltage calibration system additive constant. This data shall only be used for application purposes and provides the opportunity for global change of heater voltage. Default value: 0V.
C_FAC_V_EFC_AS_LSH_UP	Multiplicative correction factor to effective heater voltage. Calibration data only for application system purposes, e.g. check system reaction on heater deactivation without leaving state LSH_POW_CTL. Default value: 1.

The following calibration is relevant only for a system with linear sensor upstream and binary sensor downstream.

LC_LSH_CTL_CLL_INH_LSH _UP[NC_CBK_EX_NR]	Boolean constant to permit suspension of closed loop heater control. It should be set = 1 for binary sensors (no closed loop control recommended). For system configurations with linear oxygen sensor upstream LC_LSH_CTL_CLL_INH_LSH_UP[i] = 0
C_TTIP_SP_LS_UP[NC_CBK_EX _NR]	Oxygen sensor set temperature for closed loop control. For calibration see data of sensor specification (best working point).
C_TTIP_DIF_MIN_CTL_P_LS_UP	Minimum required temperature difference in order to calculate P term of controller. If the closed cloop heater control is required to act also on small deviations between measured and set-point temperature this data shall be set to small values, e.g. 0°C.
C_FAC_CTL_P_LSH_UP	Multiplicative scaling factor for closed loop P-term. For common approach this data shall be set close to 1 and the calculation of the P share be achieved by calibration of the corresponding maps IP_FAC_POS_CTL_P LSH_UP and IP_FAC_NEG_CTL_P_LSH_UP. E.g. 1.
C_FAC_CTL_I_LSH_UP	Multiplicative scaling factor for closed loop I-term. For common approach this data shall be set close to 1 and the calculation of the I share be achieved by calibration of the corresponding maps IP_FAC_POS_CTL_I_ LSH_UP and IP_FAC_NEG_CTL_I_LSH_UP. E.g. 1.
IP_FAC_NEG_CTL_P_LSH_UP	Closed loop controller P-term factor for temperature too high; decrease heater voltage. The map values shall be calibrated such that the conroller may react sufficiently on steps of load and revolution number, i.e. to keep temperature fluctuations smaller than 15-20°C without tendency to temperature oscillations. As the dependence on revolution number and exhaust gas temperature is small the same values may be used for all map entries. A pragmatic way to calibrate the p-share would be first to find a certain p-value for a given load point where the temperature oscillates with constant amplitude while I controller is deactivated via corresponding map. This p-share value shall then multiplied by 0.45 and used as final calibration value (Ziegler-Nichols-method). E.g. 0.12V/K.
IP_FAC_POS_CTL_P_LSH_UP	Closed loop controller P-term factor for temperature too low; increase heater voltage. See IP_FAC_NEG_CTL_P_LSH_UP.

IP_FAC_NEG_CTL_I_LSH_UP	Closed loop controller I-term factor for temperature too high; decrease heater voltage. Calibration shall be done after IP_FAC_NEG_CTL_P_LSH_UP has been calibrated properly. An empirical way would be to start with high I-shares with the system in oscillating condition. As next step the I-share shall reduced sufficiently until the oscillations disappear. E.g. 0.002 V/K.
IP_FAC_POS_CTL_I_LSH_UP	Closed loop controller I-term factor for temperature too low; increase heater voltage. See IP_FAC_NEG_CTL_I_LSH_UP.
ID_TTIP_SP_DIF_LSH_UP	Time dependent reduction factor for TTIP threshold of transition conditions LSH_POW_RISE to LSH_POW_CTL. This calibration data facilitates the transition from state LSH_POW_RISE to LSH_POW_CTL step-wize when subtracted from C_TTIP_SP_LS_UP. The aim is to provide the activation condition for start of the OBDII heater diagnosis before forced operative readiness is set. Calibration should be done such that the TTIP setpoint difference C_TTIP_SP_LS_UP - C_TTIP_MIN_LS_UP _READY amounts to 0°C short before the time when T_TOUT_LS_UP_READY[i] for setting of forced readiness has expired (e.g. 118s). This means, e.g. for C_TTIP_SP_LS_UP = 780°C and C_TTIP_MIN_LS_UP = 610°C, the maximum map entry at 118s should be chosen as 170°C. The lowest map entry for 0s may be chosen same as C_TTIP_SP_LS_UP. The map values between lowest and largest time may be evenly spread over the whole range. This also holds for the time axis.

Calibration data referring to sensor protection

The calibration shall be done in a climate chamber using a lambda sensor with tip-temperature measurement capability (TTS).

C_V_EFC_MAX_LSH_UP	Maximum effective voltage the heater is allowed to run. For calibration see data of sensor specification (long run). Additionally the voltage drop of electrical wiring which has to be determined for each system individually shall be considered. E.g. 13.6V.
C_V_EFC_MAX_ST_LSH_UP	Maximum effective voltage at which the heater is allowed to run in the first start phase. For calibration see data of sensor specification (start phase), e.g. 10V. The variable belongs to C_T_MAX_V_EFC_MAX_ST_LSH_UP.
C_T_MAX_V_EFC_MAX_ST_LSH_UP	Primary corresponding timer threshold to C_V_EFC_MAX_ST_LSH_UP, e.g. 10s. For calibration see sensor specification (start phase).
C_T_AST_MAX_LSH_UP	Secondary corresponding timer threshold to C_V_EFC_MAX_ST_LSH_UP. Intention is a back up, if heater state POW_CTL is reached very early e.g. due to warm sensor. For calibration see sensor specification (start phase) and usual time to reach state POW_CTL including safety margin. e.g. 60s
C_V_EFC_TOL_LSH_UP	Effective heater voltage threshold above which the voltage may be limited if the threshold is exceeded persistently. For calibration see sensor specification, e.g. 13.4V.

C_T_MAX_V_EFC_TOL_LSH_UP	Maximum permitted duration where excessive heater voltage is tolerated for short-term, e.g. 10s. For calibration see sensor specification. This data refers to the voltage threshold C_V_EFC_TOL_LSH_UP.
C_V_EFC_LIM_LSH_UP	Voltage to which the effective heater voltage is limited if the heater voltage exceeds the threshold C_V_EFC_TOL_LSH_UP for longer than C_T_MAX_V_EFC_TOL_LSH_UP. For calibration see sensor specification, e.g. 12.8V. (Remark: may in some cases be choosen same as C_V_EFC_TOL_LSH_UP)
C_T_MAX_V_EFC_LIM_LSH_UP	Maximum duration while heater voltage limited to C_V_EFC_LIM_LSH_UP, e.g. 20s. For calibration see sensor specification.
C_TEG_DYN_MAX_LSH_UP	Maximum permitted exhaust gas temperature above which LSH is switched off. If exhaust gas temperature exceeds this threshold overheating may occur, therefore heating shall no more be allowed; see data of sensor specification for maximum exhaust gas temperature. For very high exhaust gas temperatures the ceramic temperature determination may run out of the sensible range. Therefore a check with TTS is recommended. E.g. 950°C
C_VB_MAX_PROT_LSH_UP	Maximum voltage at which the heater is allowed to run. This data refers to battery voltage VB and not to effective heater voltage. For calibration see data of sensor specification, e.g. 16.5V
C_V_EFC_PROT_VB_LSH_UP	Effective voltage at which the heater is allowed to run even with VB exceeding C_VB_MAX_PROT_LSH_UP. This means that the actual battery voltage is applied to the heater with the frequency of the heater control. For calibration see data of sensor specification, as safety calibration this data should be chosen as 0V.

Activation and calibration of forced lambda sensor heating before engine start

The following calibration data section is relevant only for cases in which the lambda sensor heater shall be switched on before engine start for application purpose in order to get a valid lambda signal (for linear sensor) or lambda sensor voltage (binary sensor) at engine start. If possible the calibration should be done once using a TTS (see above).

It shall be emphasized that during this procedure the dew point recognition as lambda sensor safety mechanism is disabled. Therefore a high risk for lambda sensor damage due to water splash is present (see also meaning of C_V_EFC_RED_LSH_UP). This feature is for application purpose only !

LC_LSH_UP_MAN_ACT[NC_CBK_EX_NR]	Setting this calibration data activates the initiation of lambda sensor forced heating before engine start without awaiting dew point recognition. Its activation is triggered by the transition of LC_LSH_UP_MAN_ACT[i] = 0 => 1 and can thus not be activated w/o application system for safety reasons. The activated forced heating is indicated by LV_LSH_UP_ MAN_ACT[i] which can be used as interface for other functions. HIGH RISK FOR SENSOR DAMAGE !!!
C_V_EFC_INI_LSH_UP_MAN_ACT	Initial effective heater voltage when forced lambda

	sensor heating before engine start is required. This data should be calibrated such that the temperature rising rate does not exceed the specified limits in order to prevent additional risk for sensor damage, e.g. 6V.
C_V_EFC_MAX_LSH_UP_MAN_ ACT	Absolute maximum permitted effective heater voltage during forced heating before engine start. Calibration should be done such that during the initial voltage rising phase the effective heater voltage does not take excessive values which may lead to overheating (additional risk for sensor damage should be avoided). E.g.: 12V
C_FAC_V_EFC_LSH_UP_MAN_ ACT	Factor which allows to reduce the effective heater voltage of the open loop value during POW_CTL as derived from IP_V_EFC_CTL_LSH_UP. Background is that during forced heating before engine start no lambda sensor cooling due to forced convection by cold exhaust gas is present. Therefore the heater voltage shall be reduced to prevent the sensor from overheating in order to avoid additional risk for sensor damage; e.g. 0.8.
C_T_MAX_CDN_LSH_UP_ MAN_ACT	Time threshold for transition from state POW_RISE to POW_CTL with activated forced heating before engine start. This calibration should be done such that the lambda sensor has sufficient time to reach nominal operating temperature before switching to open loop control in POW_CTL, e.g. 20s. When this time has passed and after an additional time span of a couple of seconds (e.g. 10s, in order to await stable temperature conditions) the lambda sensor may be considered as ready for signal acquisition at engine start or 2-4 sec after engine start for binary / linear lambda sensors, respectively.

16.89.2 Downstream oxygen sensor heater management

16.89.2.1 Calibration interfering functions

16.89.2.1.1 Before start of the calibration of the downstream oxygen sensor heater management, following other functions must be calibrated:

- Basic engine tuning (ignition, basic injection, torque model, variable valve timing)
- Intake manifold model (MAP/MAF system)
- Internal resistance determination
- Exhaust gas temperature model and dew point recognition downstream catalyst
- Readiness detection of downstream lambda sensor

16.89.2.1.2 Before calibration of the downostream oxygen sensor heater management, the following modifications on the calibration data should be done:

Nothing.

16.89.2.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below. Parameters which are relevant for cold start may be calibrated independently on the remaining parameters for heater control and sensor protection.

Remark: If the same N_32 axis points are used for the interpolation maps IP_V_EFC_CTL_LSH_UP / IP_V_EFC_CTL_LSH_DOWN the calibration of the precontrol maps upstream and downstream can be done at the same time.

16.89.2.3 Calibration method

Calibration under cold start and hot restart conditions

The calibration shall be done in a climate chamber using a lambda sensor with tip-temperature measurement capability (TTS).

IP_V_EFC_INI_LSH_DOWN	Initial effective heater voltage. This map is dependent on the inital sensor temperature TEMP_INI_LS_DOWN[i] (differentiation between cold start and hot restart) and has main influence on the maximum temperature rise rates during POW_RISE. For calibration the following aspects are important: 1) The sensor specification for maximum intial voltage must be fulfilled. 2) Calibration with too large map entries may lead large temperature rise rates which violate the sensor specification. This has to be avoided. 3) In combination with too large map entries of IP_T_TEMP_THD_LSH_DOWN the maximum allowable tip temperature may exceed the specified limit – if available – before dew point recognition.

This has to be avoided.

4) The data should be calibrated to achieve the lowest possible light-off time under consideration of 1-3.

The calibration should at best be done in a climate chamber in combination with TEMP_INI_LS_DOWN[i], IP_T_TEMP_THD_LSH_DOWN and C_V_EFC_INC_LSH_DOWN using aTTS with strong limit heater (critical for temperature rise rates). On the other hand the lowest temperature during dew-point recognition should exceed the recommended value in the specification – if available – for weak limit heater.

IP_T_TEMP_THD_LSH_DOWN	Duration until critical water splash sensor temperature (typ. < 300°C, see sensor specification) reached under any conditions. Check must be done with TTS. Check on chassis dyno is not necessary. Calibration has to be done in accordance with IP_TEMP_INI_LS_DOWN.
C_T_MAX_POW_RISE_LSH_DOWN	Maximum allowed time for state POW_RISE. Calibration has to be done with TTS. Under all conditions tip temperature has to reach a minimum temperature and should not exceed the maximum operating temperature (see sensor specification for operating range of temperature). Cirtical edge is of course warm start with long heating for maximum allowed POW_RISE time. E. g. 14s.
C_V_EFC_INC_LSH_DOWN	Effective heater voltage increment to raise temperature. Calibration shall be done under consideration of the maximum specification value. In general a larger value may shorten light-off time but may also increase the temperature rise rate above the specified limit (thermal stress). Too large values may also lead to a violation of the maximum specified temperature threshold unitl dew-point detection. The increment of heating voltage should be determined with a TTS sensor. In general smaller values are recommended. E.g. 0.4V/s.
C_V_EFC_DEC_LSH_DOWN	Effective heater voltage decrement to lower temperature. Calibration shall be done under consideration of the maximum value in the sensor specification. If no value is specified the decrement can be choosen same as increment. Because of risk for high thermal stress smaller values are recommended. E.g. 0.4V/s.
C_V_EFC_RED_LSH_DOWN	Reduced effective heater voltage during danger of water splash damage. Calibration should be done such that the temperature is kept below the specified value (<300°C for most sensors, with safety margin) or that the heater voltage does not exceed a specified limit. For calibration a TTS with strong limit heater is recommended. E.g. 4V.
C_V_EFC_STEP_LSH_DOWN	Additional effective heater voltage step to IP_V_EFC_INI_LSH_DOWN on transition from LSH_POW_RED to LSH_POW_RISE. The calibration has to be done with TTS if the data is not specified in the sensor specification anyway. Calibration of this data has to take into account the maximum permitted temperature gradient, if it is mentioned within sensor specification. E.g. 1V.

Calibration of lambda sensor heater control

The calibration shall be done on a chassis dyno. The tip-temperature should be measured using a sensor with temperature measurement capability (TTS) or by evaluating the results of the internal resistance determination. After calibration the results should be checked under real conditions on road because of possible heat accumulation on chassis dyno.

IP_V_EFC_CTL_LSH_DOWN	Open loop control effective heater voltage. The map entries can be determined best on chassis dyno, as it is very sensible to geometry of the exhaust gas system. For each load as characterized by exhaust gas temperature and engine speed the necessary heating voltage to reach the setpoint temperature has to be determined. The setpoint temperature can be controlled by two ways: either by using a TTS or by evaluation of the determined internal resistance values and the corresponding conversion to element temperature (see engineering specification). Cooling of the fan should be adjusted to medium level. If the system has two exhaust gas banks which differ in geometry just focus on bank 1 for calibration. For bank 2 see IP_FAC_V_EFC_TEG_ LSH_DOWN. For set-point temperature please refer to engineering specification. E.g. 650°C.
IP_FAC_V_EFC_TEG_LSH_DOWN	Factor for the heat transfer rate, bank 2 only. This interpolation map reflects the difference in geometry between the exhaust gas banks due to, e.g., different cooling by airstream. Tests are done best on road (highway etc.)
C_V_EFC_AS_LSH_DOWN	Effective heater voltage calibration system additive constant. This data shall only be used for application purposes and provides the opportunity for global change of heater voltage. Default value: 0V.
C_FAC_V_EFC_AS_LSH_DOWN	Multiplicative correction factor to effective heater voltage. Calibration data only for application system purposes, e.g. check system reaction on heater deactivation without leaving state LSH_POW_CTL. Default value: 1.

Calibration data referring to sensor protection

The calibration shall be done in a climate chamber using a lambda sensor with tip-temperature measurement capability (TTS).

C_V_EFC_MAX_LSH_DOWN	Maximum effective voltage the heater is allowed to run. For calibration see data of sensor specification (long run). Additionally the voltage drop of electrical wiring which has to be determined for each system individually shall be considered. E.g. 13.6V.
C_V_EFC_MAX_ST_LSH_DOWN	Maximum effective voltage at which the heater is allowed to run in the first start phase. For calibration see data of sensor specification (start phase), e.g. 10V. The variable belongs to C_T_MAX_V_EFC_MAX_ST_LSH_DOWN.
C_T_MAX_V_EFC_MAX_ST_LSH_DOWN	Primary corresponding timer threshold to C_V_EFC_MAX_ST_LSH_DOWN, e.g. 10s. For calibration see sensor specification (start phase).
C_T_AST_MAX_LSH_DOWN	Secondary corresponding timer threshold to C_V_EFC_MAX_ST_LSH_DOWN. Intention is a back up, if heater state POW_CTL is reached very early e.g. due to warm sensor. For calibration see sensor

	specification (start phase) and usual time to reach state POW_CTL including safety margin. e.g. 60s
C_V_EFC_TOL_LSH_DOWN	Effective heater voltage threshold above which the voltage may be limited if the threshold is exceeded persistently. For calibration see sensor specification, e.g. 13.4V.
C_T_MAX_V_EFC_TOL_LSH_DOWN	Maximum permitted duration where excessive heater voltage is tolerated for short-term, e.g. 10s. For calibration see sensor specification. This data refers to the voltage threshold C_V_EFC_TOL_LSH_DOWN.
C_V_EFC_LIM_LSH_DOWN	Voltage to which the effective heater voltage is limited if the heater voltage exceeds the threshold C_V_EFC_TOL_LSH_DOWN for longer than C_T_MAX_V_EFC_TOL_LSH_DOWN. For calibration see sensor specification, e.g. 12.8V. (Remark: may in some cases be choosen same as C_V_EFC_TOL_LSH_DOWN)
C_T_MAX_V_EFC_LIM_LSH_DOWN	Maximum duration while heater voltage limited to C_V_EFC_LIM_LSH_DOWN, e.g. 20s. For calibration see sensor specification.
C_TEG_DYN_MAX_LSH_DOWN	Maximum permitted exhaust gas temperature above which LSH is switched off. If exhaust gas temperature exceeds this threshold overheating may occur, therefore heating shall no more be allowed; see data of sensor specification for maximum exhaust gas temperature. For very high exhaust gas temperatures the ceramic temperature determination may run out of the sensible range. Therefore a check with TTS is recommended. E.g. 950°C
C_VB_MAX_PROT_LSH_DOWN	Maximum voltage at which the heater is allowed to run. This data refers to battery voltage VB and not to effective heater voltage. For calibration see data of sensor specification, e.g. 16.5V
C_V_EFC_PROT_VB_LSH_DOWN	Effective voltage at which the heater is allowed to run even with VB exceeding C_VB_MAX_PROT_LSH_DOWN. This means that the actual battery voltage is applied to the heater with the frequency of the heater control. For calibration see data of sensor specification, as safety calibration this data should be chosen as 0V.

16.90 Upstream oxygen sensor heater OBDI diagnosis

16.90.1 General Informations

This calibration hint refers to the following specification: A01M

The purpose of the diagnosis shall be to detect electrical faults within the oxygen sensor heater circuit as defined by OBD I requirements. The error detection is effected via the ECU hardware and is able to detect short circuit to battery voltage, short circuit to ground and open circuit.

16.90.1.1 Requirements

- The calibration shall be carried out with a common oxygen sensor.
- Additionally a break-out box is necessary to simulate the three possible electrical circuit faults
- A standard application system performing online variable recording with a minimum sample rate of 100 ms is required.

16.90.2 General calibration hint

16.90.2.1 Calibration interfering functions

- functionalities which have to be calibrated and running well in advance:
  - ❖ exhaust gas temperature model

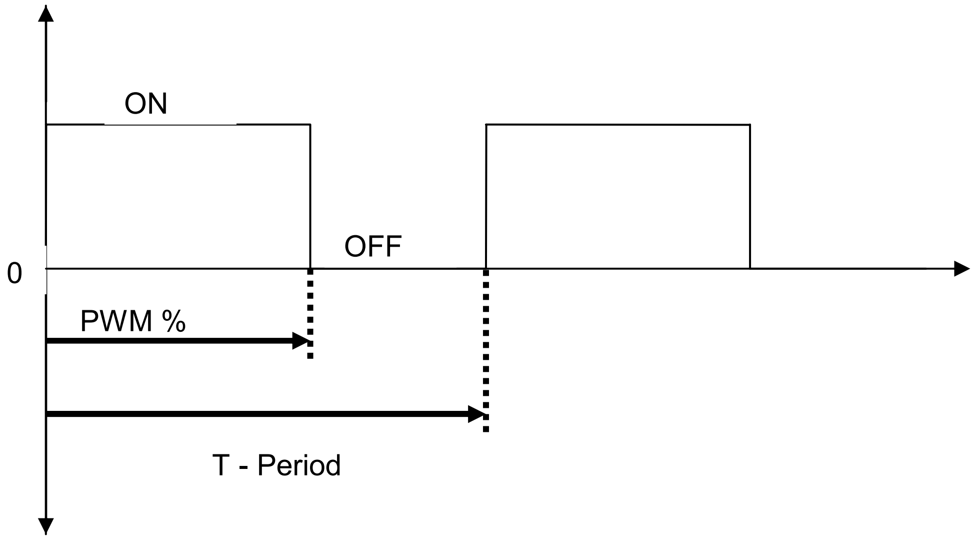
16.90.3 Calibration method

16.90.3.1 Calculation of LSHPWM limits

The upper and lower limits for the PWM thresholds (C_LSHPWM_MAX_DIAG_LSH_UP and C_LSHPWM_MIN_DIAG_LSH_UP) are dependent on the filter time of the heater driver and the PWM frequency used. For calibration the heater driver PWM frequency needs to be known.

Typically the filter time of the heater driver for the recognition

- of a short circuit to battery (SCB) is 0,05 ms (min. ON).
- of a short circuit to ground (SCG) is 0,25 ms (min. OFF).
- of open circuit (OC) is 0,40 ms (min. ON).



As an example a PWM frequency of 10 Hz is used which leads to a period of 100 ms.

C_LSHPWM_MIN_DIAG_LSH_UP	is the minimum permitted heater effective voltage PWM for upstream LSH OBD1 diagnosis. In order for an OC to be recognised, the heater driver must remain on for at least 0,4 ms. Hence this represents (100% * 0,40 ms / 100 ms) PWM = 0.4 %. As the PWM resolution is 0.3891 %, this value is chosen for the calibration constant. Additionally the calibration data C_LSHPWM_MIN_DIAG_LSH_UP must be lower or equal to NC_LSHPWM_BOL_LSH_UP.
C_LSHPWM_MAX_DIAG_LSH_UP	is the maximum permitted heater effective voltage PWM for upstream LSH OBD1 diagnosis. In order for a SCG to be recognised, the heater driver must remain off for at least 0,25 ms. Hence this represents (100% * (1- (0,25 ms / 100 ms))) PWM = 99.75 %. As the PWM resolution is 0.3891 %, 100.00 – 0.39 = 99.61 % value is chosen for the calibration constant. Additionally the calibration constant C_LSHPWM_MAX_DIAG_LSH_UP must be higer or equal to NC_LSHPWM_TOL_LSH_UP.

16.90.3.2 Calculation of exhaust gas temperature limit

C_TEG_DYN_MIN_DIAG_LSH_UP	is the minimum exhaust gas temperature threshold to permit SCB fault to be set during LSH preheating state. This constant shall be calibrated such that the exhaust gas temperature is sufficiently warm to heat the sensor to a temperature where no heater inrush current exceeding the driver over-current threshold (typ. 8 A) occurs. The inrush current transient will only occur in the initial few seconds of heating (1 to 5 seconds) and is dependent on the sensor type and environmental conditions.

16.90.3.3 Calibration data for error debouncing

C_ABC_INC_DIAG_LSH_UP	is the anti-bounce counter increment for upstream O2-sensor heater diagnosis. This constant shall be calibrated according to project philosophy. As a first hint the increment may be set to 1.

C_ABC_MAX_DIAG_LSH_UP	is the maximum value of the anti-bounce counter for upstream O2-sensor heater diagnosis. After having reached this number the error flag is being set. The constant shall be calibrated according to project philosophy. As a first hint C_ABC_MAX_DIAG_LSH_UP may be set to 50 which means that using a recurrence of 0,1 s the diagnosis will be finished after 5 s after the activation conditions have been met and the fault being present. Additionally this number should not be chosen too large since several OBD II functions are intented to start after the electrical diagnosis has finished and LV_END_DIAG_LSH_UP is set, e.g. the heater OBD II diagnosis.

16.90.4 Validation

The validation is done using a break-out box to simulate the possible electrical faults. Attention shall be paid to the SCG test connecting the heater cabling to ground since in case of a short in the battery – lambda sensor connection a fuse will blow and in case of a lambda sensor - µ-controller short cut the lambda sensor may overheat without the safety precautions being able to disrupt.

16.91 Downstream oxygen sensor heater OBDI diagnosis

16.91.1 General Informations

This calibration hint refers to the following specifications: A01N

The purpose of the diagnosis shall be to detect electrical faults within the oxygen sensor heater circuit as defined by OBD I requirements. The error detection is effected via the ECU hardware and is able to detect short circuit to battery voltage, short circuit to ground and open circuit.

16.91.1.1 Requirements

- The calibration shall be carried out with a common oxygen sensor.
- Additionally a break-out box is necessary to simulate the three possible electrical circuit faults
- A standard application system performing online variable recording with a minimum sample rate of 100 ms is required.

16.91.2 General calibration hint

16.91.2.1 Calibration interfering functions

- functionalities which have to be calibrated and running well in advance:
    - ❖ exhaust gas temperature model

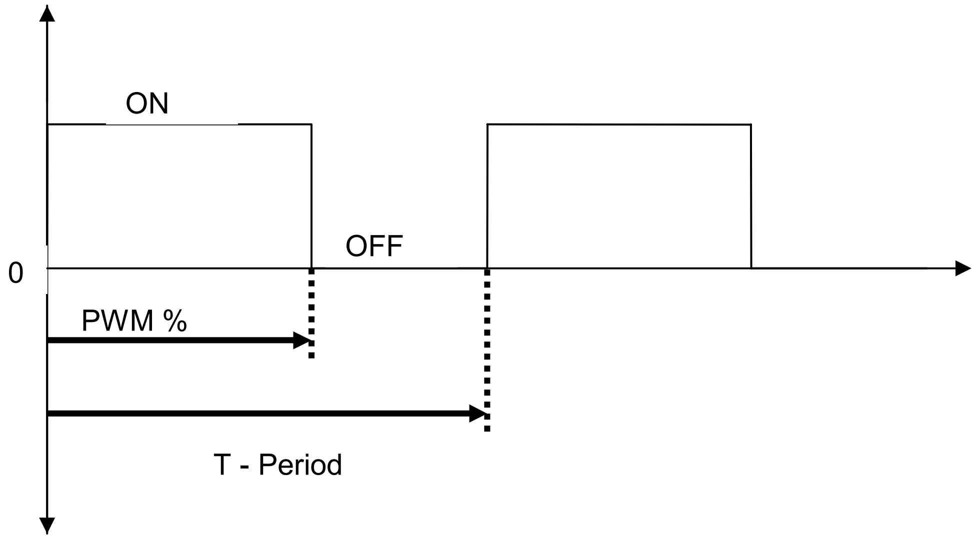
16.91.3 Calibration method

16.91.3.1 Calculation of LSHPWM limits

The upper and lower limits for the PWM thresholds (C_LSHPWM_MAX_DIAG_LSH_DOWN and C_LSHPWM_MIN_DIAG_LSH_DOWN) are dependent on the filter time of the heater driver and the PWM frequency used. For calibration the heater driver PWM frequency needs to be known.

Typically the filter time of the heater driver for the recognition

- of a short circuit to battery (SCB) is 0,05 ms (min. ON).
- of a short circuit to ground (SCG) is 0,25 ms (min. OFF).
- of open circuit (OC) is 0,40 ms (min. ON).



As an example a PWM frequency of 10 Hz is used which leads to a period of 100 ms.

C_LSHPWM_MIN_DIAG_LSH_DOWN	is the minimum permitted heater effective voltage PWM for upstream LSH OBD1 diagnosis. In order for an OC to be recognised, the heater driver must remain on for at least 0,4 ms. Hence this represents (100% * 0,40 ms / 100 ms) PWM = 0.4 %. As the PWM resolution is 0.3891 %, this value is chosen for the calibration constant. Additionally the calibration data C_LSHPWM_MIN_DIAG_LSH_DOWN must be lower or equal to NC_LSHPWM_BOL_LSH_DOWN.
C_LSHPWM_MAX_DIAG_LSH_DOWN	is the maximum permitted heater effective voltage PWM for upstream LSH OBD1 diagnosis. In order for a SCG to be recognised, the heater driver must remain off for at least 0,25 ms. Hence this represents (100% * (1- (0,25 ms / 100 ms))) PWM = 99.75 %. As the PWM resolution is 0.3891 %, 100.00 – 0.39 = 99.61 % value is chosen for the calibration constant. Additionally the calibration constant C_LSHPWM_MAX_DIAG_LSH_DOWN must be higer or equal to NC_LSHPWM_TOL_LSH_DOWN.

16.91.3.2 Calculation of exhaust gas temperature limit

C_TEG_DYN_MIN_DIAG_LSH_DOWN	is the minimum exhaust gas temperature threshold to permit SCB fault to be set during LSH preheating state. This constant shall be calibrated such that the exhaust gas temperature is sufficiently warm to heat the sensor to a temperature where no heater inrush current exceeding the driver over-current threshold (typ. 8 A) occurs. The inrush current transient will only occur in the initial few seconds of heating (1 to 5 seconds) and is dependent on the sensor type and environmental conditions.

16.91.3.3 Calibration data for error debouncing

C_ABC_INC_DIAG_LSH_DOWN	is the anti-bounce counter increment for upstream O2-sensor heater diagnosis. This constant shall be calibrated according to project philosophy. As a first hint the increment may be set to 1.
C_ABC_MAX_DIAG_LSH_DOWN	is the maximum value of the anti-bounce counter for upstream O2-sensor heater diagnosis. After having reached this number the error flag is being set. The constant shall be calibrated according to project philosophy. As a first hint C_ABC_MAX_DIAG_LSH_UP may be set to 50 which means that using a recurrence of 0,1 s the diagnosis will be finished after 5 s after the activation conditions have been met and the fault being present. Additionally this number should not be chosen too large since several OBD II functions are intented to start after the electrical diagnosis has finished and LV_END_DIAG_LSH_DOWN is set, e.g. the heater OBD II diagnosis.

16.91.4 Validation

The validation is done using a break-out box to simulate the possible electrical faults. Attention shall be paid to the SCG test connecting the heater cabling to ground since in case of a short cut in the battery – lambda sensor connection a fuse will blow and in case of a lambda sensor - µ-controller short cut the lambda sensor may overheat without the safety precautions being able to disrupt.

16.92 Downstream Fuel Trim diagnosis

16.92.1 Calibration interfering functions

16.92.1.1 Before start of the calibration of the downstream fuel trim, the following other functions must be calibrated:

- Basic Engine Tuning (i.e. Ignition, Basic Injection, Torque Model, Variable Valve Timing)
- Intake Manifold Model (MAP System)
- Wallfilm Compensation Model
- Evaporative Emission Control
- Temperature Model up- and downstream catalyst
- Signal Acquisition of linear lambda sensor (controller and heater calibration)
- Readiness detection of up- and downstream lambda sensor
- Lambda Control Parameter fixed
- Forced Stimulation
- Basic Lambda Setpoint
- Trim Control
- Catalyst Purge

16.92.1.2 Before calibration of the downstream fuel trim diagnosis, the following modifications on the calibration data should be done:

Nothing.

16.92.2 Calibration flowchart

The sequence is described by the order of constants and maps in the table of the “Calibration method” section below.

16.92.3 Calibration method

Monitoring of Trim Controller I-share

The calibration can be done on the road.
The calibration should be checked in emission test.
The failure can be caused i.e. by adding + / - 4 % on IP_LAMB_LS_UP[NC_CBK_EX_NR].

C_LAMB_DELTA_I_MAX_DIAG	Maximum limit of LAMB_DELTA_I; This value should be slightly lower than the maximum possible trim control limit. If 5 % for trim control limit is used, take I.e. 4 % for the diagnosis limit.
C_LAMB_DELTA_I_MIN_DIAG	Minimum limit of LAMB_DELTA_I; See explanation for C_LAMB_DELTA_I_MAX_DIAG. I.e. – 4 %.
C_T_SUM_MAX_MAX_DELTA_I	Total duration of trim control on the maximum limit;

_LAM	It must be shorter than the reset time, i.e. 30 % of it and shorter than the maximum available fuel trim activation time in the emission test. If the trim control I-share is calibrated quite slowly, the time can be even very short, i.e. 10 sec., as the I-share itself can be treated as a kind of debouncing. The time should be longer if it is known, that the trim control I-share is calibrated quite fast and / or the A/F precontrol is not precise.
C_T_SUM_MIN_MAX_DELTA_I_LAM	Total duration of trim control on the minimum limit; See at C_T_SUM_MAX_DIAG_LAM_ADJ. I.e. 10 sec.
C_T_SUM_RST_MAX_DELTA_I_LAM	Time counter to reset min and max counter; See also C_T_SUM_MAX_MAX_DELTA_I_LAM for further explanation.
C_T_SUM_MAX_ERR_DELTA_I_LAM	The value gives a kind of characteristic number for reliability. If a failure symptom during C_T_SUM_READY_DELTA_I_LAM was available for not longer than C_T_SUM_MAX_ERR_DELTA_I_LAM, than the end flag is set. If the symptom was available for a longer time, then it is decided to wait for one complete "reset cycle", which means it will take at least C_T_SUM_RST_MAX_DELTA_I_LAM seconds until the end flag will be set, except a failure is detected before.
C_T_SUM_READY_DELTA_I_LAM	Timer for fast end of trim diagnosis. The purpose is to set the end flag in a completely symptomless driving cycle earlier. If a failure symptom during C_T_SUM_READY_DELTA_I_LAM was available for not longer than C_T_SUM_MAX_ERR_DELTA_I_LAM, than the end flag is set. In general, to fulfill its purpose, it shall be shorter than C_T_SUM_RST_MAX_DELTA_I_LAM, i.e. half as long.

Monitoring of Trim Controller Deviation

The calibration can be done on the road.
The calibration should be checked in emission test.
A catalyst with maximum oxygen storage capacity and with no storage capacity shall be used for the final check of the calibration.
If the failure is simulated via adjustment of IP_LAMB_LS_UP +/- 10%, it depends on the calibration of the MAF integrals and the VLS_DOWN_DIF thresholds if this failure will be met at first in a driving cycle like the FTP75. Because of the dynamic driving, the VLS_DOWN_DIF thresholds might not be met during a long time, and then at first the trim control I-share diagnosis would be entered into the failure memory. Anyway, the calibration must be adjusted to be able to produce the failure in the FTP75 emission test. The demonstration with an external tool in the FTP75 test cycle might be possible, if a fix downstream sensor signal is applied only for the phase of constant driving and not for fuel cut off phases. Otherwise, other diagnosis functions would find an error. The failure MAF integral must be calibrated so that it is exceeded at the end of this constant driving phase. But be careful to check if the trim control P-share would be able to bring the VLS_DOWN signal away from the diagnosis threshold under all circumstances for a good sensor.
Another possibility to demonstrate it is to make the VLS_DOWN thresholds very aggressive (close to values which can occure with good sensor), but this is not recommended due to the risk of wrong failure entry.

For the PVE test it seems to be more simple. A stuck VLS_DOWN must be simulated (external tool) which would lead to exceeded thresholds. And then, a driving cycle must be performed, which does not include any fuel cut off.

The failure MAF integral should not be smaller than 2/3 of the reset MAF integral to prevent a failure entry due to a damaged catalyst (oscillating VLS_DOWN).

C_VLS_DIF_MAX_DIAG	Maximum threshold for VLS_DIF_DLY deviation; Lowest setpoint of trim controller – significant lean voltage (i.e. 0,05 V) which would not occure during normal operation with trim control P-share on (be careful if P-share has only small effect). I.e. 0,65 (setpoint) – 0,05 V (lean voltage) = 0,60 V for C_VLS_DIF_MAX_DIAG.
C_VLS_DIF_MIN_DIAG	Minimum threshold for VLS_DIF_DLY deviation; The problem with this value is, that it must be choosen so high, so that the lambda sensor supplier would never guarantee this voltage for an aged sensor. So this diagnosis is reliable only for a part of the downstream lambda sensor lifetime. C_VLS_DIF_MIN_DIAG must be choosen to a value, that the lowest setpoint of trim controller + C_VLS_DIF_MIN_DIAG is higher than the VLS_DOWN voltage which occurs under normal conditions (i.e. after full load or catalyst purge or catalyst overheat protection) at the time when the p-share of fuel trim is activated (take into account also the VLS_DOWN after warmstart when the sensor is still a little cold).
C_MAF_INT_VLS_DIF_DLY_MIN	Maxmium MAF integral of VLS_DIF_DLY deviation (rich_side); It must be less than the reset integral, but not too less to prevent a failure entry in case of a damaged catalyst (oscillating VLS_DOWN) i.e. 70 % and less than the maximum available fuel trim activation integral in the emission test. I.e. 500 g.
C_MAF_INT_VLS_DIF_DLY_MAX	Maximum MAF integral of VLS_DIF_DLY deviation (lean_side); See C_MAF_INT_VLS_DIF_DLY_MIN. I.e. 500 g.
C_MAF_INT_RST_VLS_DOWN_DIF_DIAG	Threshold for integral to start a new diagnostic cycle and to set the end flag; It must be possible to reach this maf integral during the emission test as also the end flag is set after exceeding first time this reset integral in the no failure case. I.e. 750 g.
C_CAT_DIAG_MAX_DELTA_I_LAM	Deactivation threshold for the diagnosis to prevent a wrong failure entry due to oscillating VLS_DOWN after the catalyst was damaged.

Limited dynamic for the function “Monitoring the Trim Controller Deviation”

A catalyst with maximum oxygen storage capacity shall be used for the calibration.
A chassis dyno roller is recommended.
Check the behaviour of VLS_DOWN while slowly changing the parameters N / MAF / LAM_MV. Check the impact on VLS_DOWN_DIF.
Find out the acceptable change quickness and calibrate the correlation constants accordingly. If even a very slow change of the parameter makes trouble, keep the correlation constant small.
If there is no trouble due to fast changes until a certain absolute difference between the mean value and the current value, then find out the maximum acceptable absolute difference for a fast change and calibrate the window accordingly.

C_N_LDC_VLS_DIF_DLY	If only a small change quickness is acceptable, the correlation constant and the acceptable window of the parameter must be choosen small. The bigger the acceptable change quickness is, the higher the correlation constant can be choosen.
C_MAF_LDC_VLS_DIF_DLY	See C_N_LDC_VLS_DIF_DLY;
C_LAM_MV_LDC_VLS_DIF_DLY	See C_N_LDC_VLS_DIF_DLY;
C_N_CRLC_LDC_VLS_DIF_DLY	See C_N_LDC_VLS_DIF_DLY;
C_MAF_CRLC_LDC_VLS_DIF_DLY	See C_N_LDC_VLS_DIF_DLY;
C_LAM_MV_CRLC_LDC_VLS_DIF_DLY	See C_N_LDC_VLS_DIF_DLY;
C_MAF_INT_MIN_VLS_DIF_DLY_LDC	See C_N_LDC_VLS_DIF_DLY;

16.93 Cus adap module: EGCP

16.93.1 Outputs for BMW which are defined as EGCP exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_lamson_ok1	O/V	0...1H	0...1	1	[-]
Lambdasonde VK, Bank1, bereit					
B_lamson_ok2	O/V	0...1H	0...1	1	[-]
Lambdasonde VK, Bank2, bereit					
B_pruef1	O/V	0...1H	0...1	1	[-]
Lambda coordination request to change set point for active plausibility test					
B_pruef2	O/V	0...1H	0...1	1	[-]
Lambda coordination request to change set point for active plausibility test					
B_sbblsu	O/V	0...1H	0...1	1	[-]
operability status of the lambda sensor (delayed), exhaust cylinderbank 1					
B_sbblsu2	O/V	0...1H	0...1	1	[-]
operability status of the lambda sensor (delayed), exhaust cylinderbank 2					
La_pruef1	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda set point for active plausibility test bank 1					
La_pruef2	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda set point for active plausibility test bank 2					
Lam_son1	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda signal value of the WRAF sensor					
Lam_son2	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda signal value of the WRAF sensor					
Ulams_1	O/V	0...7FFH	0...9.99511	4.8828e-3	[V]
corrected output signal of the lambda sensor, exhaust cylinderbank 1					
Ulams_2	O/V	0...7FFH	0...9.99511	4.8828e-3	[V]
corrected output signal of the lambda sensor, exhaust cylinderbank 2					

Input data:

LAMB_LS_UP[NC_CBK_EX_NR]	LAMB_SP_DIAG_LS_UP_DOWN[NC_CBK_EX_NR]	LV_INH_LSCL[NC_CBK_EX_NR]	LV_LAMB_LS_UP_VLD[NC_CBK_EX_NR]
LV_LAMB_SP_REQ_DIAG_ACT[NC_CBK_EX_NR]	NC_CBK_EX_NR	VLS_UP_COR[NC_CBK_EX_NR]	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

i = 1 ... NC_CBK_EX_NR

Application conditions:

Initialisation at reset or at exit power latch state:

B_pruef1, B_pruef2, B_sbblsu, B_sbblsu2

B_lamson_ok1, B_lamson_ok2 = 0
La_pruef1 = La_pruef2 = 1.00
Lam_son1, Lam_son2, Ulams_1, Ulams_2 : first calculation

Activation: every engine state, except power latch phase

Deactivation : at power latch phase

Values at power latch phase :

Lam_son1 = Lam_son2 = 1.0
La_pruef1 = La_pruef2 = 1.0
B_pruef1 = B_pruef2 = 0
B_lamson_ok1, B_lamson_ok2 = 0
B_sbblsu = B_sbblsu2 = 1
Ulams_1= Ulams_2 = 0 V

Recurrences:	10 ms	Lam_son1/2, Ulams_1/2, B_lamson_ok1, B_lamson_ok2
	100 ms	La_pruef1/2 , B_pruef1/2, B_sbblsu, B_sbblsu2

Formula section:

Lam_son1 = LAMB_LS_UP[1]

Lam_son2 = LAMB_LS_UP[2]

La_pruef1 = LAMB_SP_DIAG_LS_UP_DOWN[1]

La_pruef2 = LAMB_SP_DIAG_LS_UP_DOWN[2]

B_pruef1 = LV_LAMB_SP_REQ_DIAG_ACT[1]

B_pruef2 = LV_LAMB_SP_REQ_DIAG_ACT[2]

B_sbblsu = not(LV_INH_LSCL[1])

B_sbblsu2 = not(LV_INH_LSCL[2])

Ulams_1 = VLS_UP_COR[1]

Ulams_2 = VLS_UP_COR[2]

B_lamson_ok1 = LV_LAMB_LS_UP_VLD[1]

B_lamson_ok2 = LV_LAMB_LS_UP_VLD[2]

17 Exhaust Gas Recirculation

Table of Contents

Index

17.1 Actuator control - Requirements for infrastructure (SF mode)

Input data:

Export actions:

ACTION_INFR_GetVAcr(OUT <V_acr_bas>**)**
This action provides the standard signal voltages of the actuator measurement system.
ACTION_INFR_SetPwmDucyAcr(IN <Pwm_acr>**)**
This action adjusts the duty-cycle and the current direction of the actuator power stage output.
ACTION_INFR_SetPwmFrqAcr(IN <Frq_pwm_acr>**)**
This action adjusts the frequency of the PWM signal from the PWM signal generator
ACTION_INFR_GetElDiagAcrDr(OUT <Lv_state_acr_dr>**)**
This action reads the status flag information of the actuator power stage!
ACTION_INFR_DisableAcrDr( )
This action is able to switch-off the actuator power stage.
ACTION_INFR_EnableAcrDr( )
This action is able to switch-on the actuator power stage.

Description for actions:

ACTION_INFR_GetVAcr(V_acr_bas**)**					
This action reads the signal voltages of the position sensor delivered from the infrastructure. The AD conversion is performed autonomously every millisecond by the infrastructure. When the action is called the gathered information will be provided to the application software level.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
V_acr_bas	OUT	0...3FFH	0...4.9951	4.9e-3	V
Signal voltage of actuator position sensor					

ACTION_INFR_SetPwmDucyAcr(Pwm_acr**)**					
This action writes the PWM duty-cycle value for the actuator power stage in the infrastructure. The PWM update and the current direction of the actuator power stage are performed autonomously by the infrastructure. When the action is called old requests will be replaced by the new request.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Pwm_acr	IN	8000...7FFFH	-100...99.997	3.05e-3	%
Output of the actuator position controller included PWM duty-cycle and current direction					

ACTION_INFR_SetPwmFrqAcr(Frq_pwm_acr**)**					
This action writes the PWM frequency, which is tuneable, for the actuator PWM signal generator into the infrastructure.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Frq_pwm_acr	OUT	0...FFFFH	0...6553.5	0.1	1/s
PWM generator frequency					

ACTION_INFR_GetElDiagAcrDr(<Lv_state_acr_dr>**)**					
This action reads the status flag information of the actuator power stage. The readout of the SF is performed autonomously and the information is gathered. When calling this Action the information inside the Infrastructure is reset after returning the OUT parameters.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Lv_state_acr_dr	OUT	0...1H	0...1	1	-
Status flag information from the actuator power stage provided for the diagnosis					

ACTION_INFR_DisableAcrDr( )
The action is used for the deactivation of the actuator power stage via enable/disable line.

ACTION_INFR_EnableAcrDr( )
The action is used for the activation of the actuator power stage via enable/disable line.

FUNCTION DESCRIPTION:

17.1.1 Closed-loop actuator position control

General information:

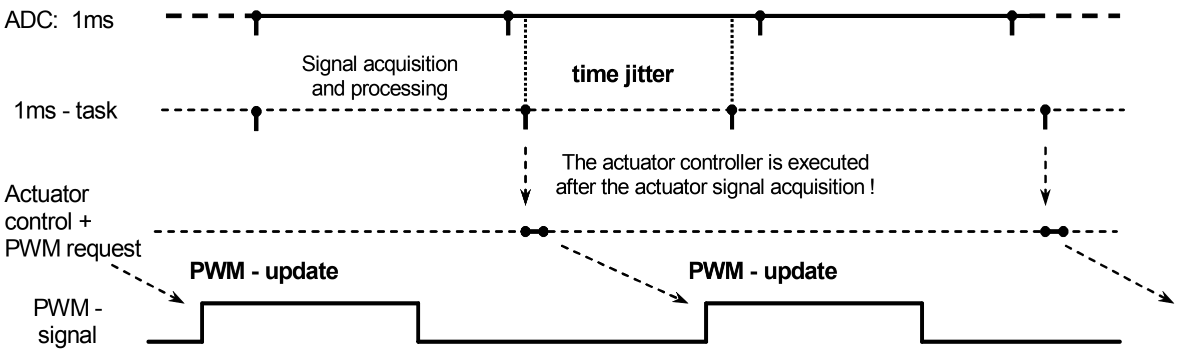
This chapter describes the time behaviour between actuator valve position signal acquisition, digital position controller calculation and the PWM output generation for the actuator power stage. The aim is to synchronise all functions for smallest time delay. The actuator valve position signal acquisition and the digital position controller are calculated in the 1 ms ASW task asynchronously to the AD queue converted every millisecond and the PWM generation (update is only possible after a complete duty-cycle). The maximum time delay between the AD conversion and PWM generation depend on time jitter between:

- AD conversion and actuator signal determination ($0 \leq T_1 \leq 1$ ms)
- Position controller calculation and PWM update ($0 \leq T_2 \leq 1$ / Frq_pwm_acr)

The maximum time delay T_{delay} is the summation of T_1 and T_2 !

($0 \leq T_{delay} \leq 3$ ms for a frequency of 500 Hz, $0 \leq T_{delay} \leq 2$ ms for a frequency of 1000 Hz)

Signal flow diagram



17.1.2 Actuator position sensor signal acquisition

General information:

The following actions are used for the acquisition of the signal voltage of the actuator valve position measurement system. The actions deliver the actuator valve position to the application software level from the standard AD queue converted every millisecond.

- The AD conversion is performed autonomously by the infrastructure every 1 ms.

Requirements for ACTION_INFR_GetVAcr:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
V_acr_bas	Not relevant	< 1% of reference supply voltage for sensor	10bit	-	-

Diagnosis: no electrical diagnosis done here

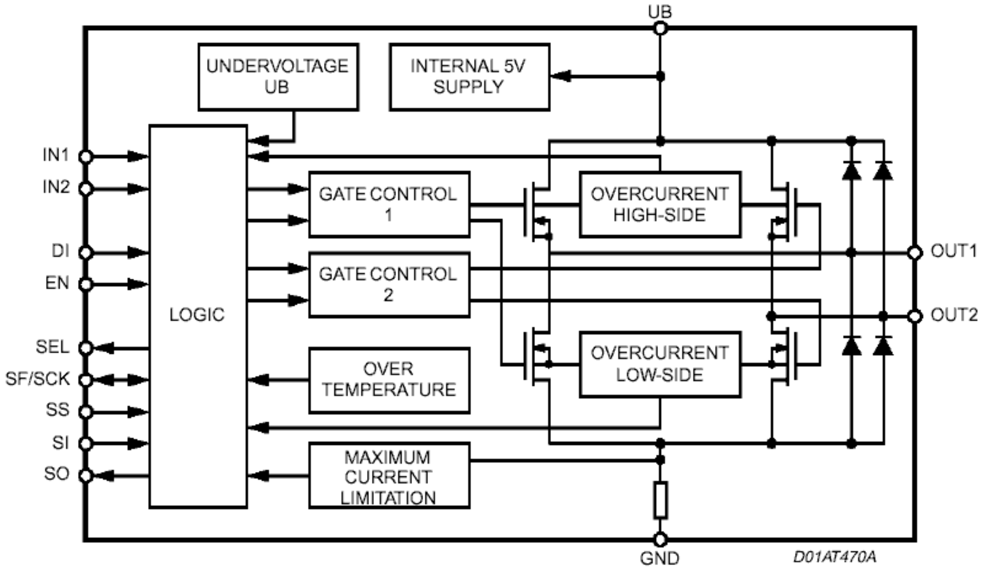
Coincidence requirements: The AD conversion is performed autonomously by the infrastructure every 1 ms.

17.1.3 Power stage for actuator position control

General information:

The used power stage is an full H-Bridge, designed for the control of DC motors in safety critical applications and under extreme environmental conditions. The power stage is protected against over temperature and short circuits and has an under voltage lockout for all the supply voltages (main DC power supply). All malfunctions cause the output stages to go tristate.

The following picture shows the internal structure of the used power stage:



Infineon TLE 7209R

IN1 / IN2 - Input pins for output power control
OUT1 / OUT2 - Output pins connected with the actuator
EN / DI - Enable and disable pin of the power stage

17.1.3.1 Control of the actuator power stage

General information:

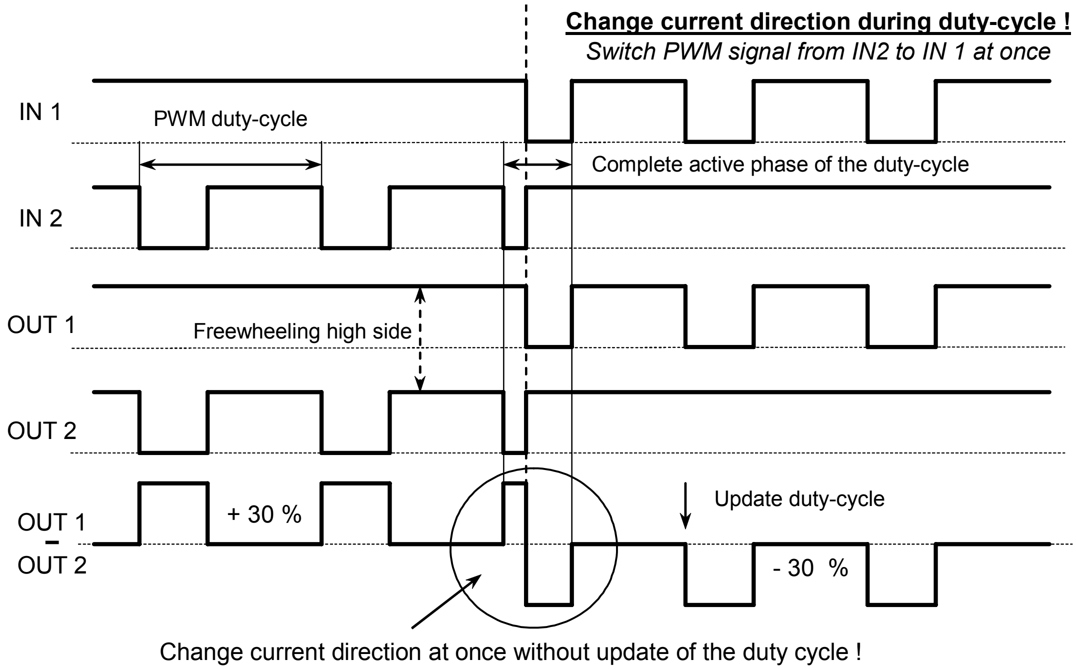
The power stage delivers the energy for the closed-loop electronic actuator valve position control. The power of the output stages (OUT1 and OUT2) should be managed by **only one** PWM signal from the ECU06 main controller (TriCore). The control strategy uses a special feature of the TriCore processor for the change of the current direction within the motor electrical circuit. The TriCore is able to switch the internal PWM source between the input pins IN1 and IN2 of the power stage, therefore only one PWM channel is necessary.

- The output stages of an active power stage change between the states "electric-power supply" and "electric freewheeling" permanently. It is absolutely necessary to manage the power stages with "**freewheeling high side**" to fulfill the high EMC requirements.

- The duty-cycle of the PWM signal starts always with an active phase "electric-power supply" and then the passive phase "electric freewheeling high side" follows.
- The current direction of the motor electrical circuit is determined autonomously by infrastructure means of the PWM request delivered from the actuator position controller. A positive PWM duty-cycle request means forward direction and a negative PWM duty-cycle request means backwards direction. The described behaviour can also be changed by configuration.
- The current direction is adjusted by the PWM source assignment asynchronously to the PWM duty-cycle, that means the current direction can be changed within the duty cycle, see signal flow diagram below.

The physical range of the duty cycle is 0 to 100% and the resolution 100% / 2^15 = 3.05e-3. After ignition on or reset the duty cycle is set to 0 %.

Signal flow diagram



Requirements for ACTION_INFR_SetPwmDucyAcr :

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Pwm_acr	-	-	3.05e - 3 %	-	After IGKON or RST the duty – cycle must be set to **0 %** !

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The time duration between set and update of the duty cycle has to be less than 1/Frq_pwm_acr. The PWM frequency is set by calling ACTION_INFR_SetPwmFrqAcr() in the application software.

Requirements for ACTION_INFR_SetPwmFrqAcr:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Frq_pwm_acr	-	-	0.1 Hz	-	Value is fixed once at RST or IGKON. **The value must not be changed while active PWM signal generator!**

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The value can be changed in accordance to hardware requirements. Default is 1000 Hz.

17.1.3.2 Actuator power stage diagnosis based on SF

General information:

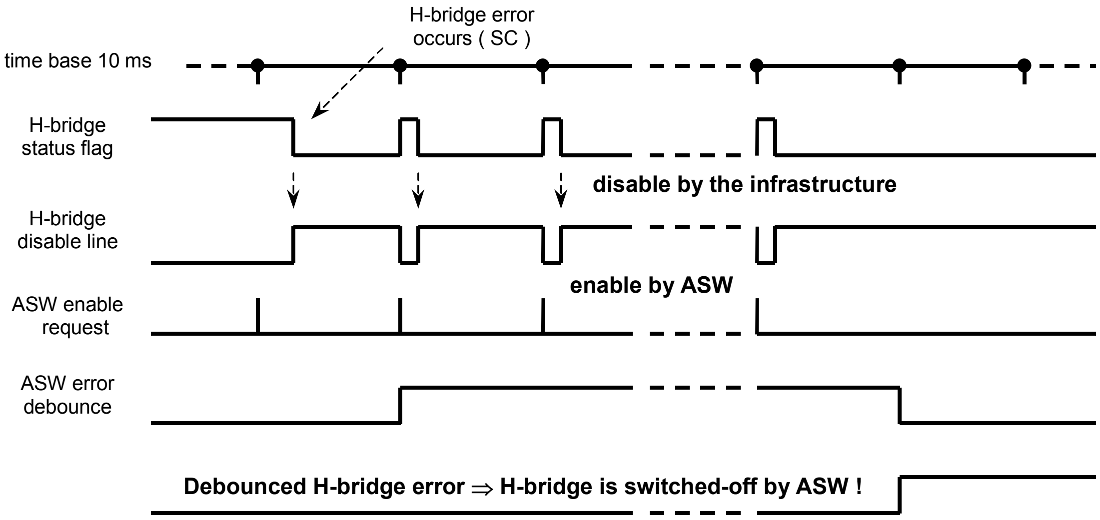
The used actuator power stage is an intelligent full H-Bridge with integrated diagnostic and must be configured in status flag mode. This action returns the result of the internal electrical diagnosis delivered by the status flag line of the actuator power stage. The following errors are detected by the internal H-bridge diagnosis:

- Overcurrent
- Overtemperature
- Undervoltage

In the status flag mode it is not possible to distinguish the errors by the software, that means only the information “error is present” will be indicated ! With detected error the power stage is switched-off automatically. The re-activation is only possible with a signal change on the enable/disable line of the power stage.

- The device readout is performed autonomously by the infrastructure.
- If the infrastructure detects an error on the power stage via status flag line, then the infrastructure must disable the H-bridge by enable/disable line autonomously.
- The error information are gathered in the infrastructure until the application reads out the information by calling ACTION_INFR_GetElDiagAcrDr().
- After having read out the information by calling ACTION_INFR_GetElDiagAcrDr(), the data inside the infrastructure are reset.

Signal flow diagram



Requirements for ACTION_INFR_GetElDiagAcrDr:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Lv_state_acr_dr	-	-	<bit coded>	-	The logical variable is set, if the error delivered by status flag of the actuator power stage has been detected.

Diagnosis:

ACTION_INFR_GetElDiagAcrDr returns the electric diagnosis of the actuator power stage.

Coincidence requirements:

The re-activation process of the actuator power stage during the error detection is a co-operation between application software and infrastructure, that means with error detection the infrastructure must disable the H-bridge (power stage) and the application software must be able to activate the power stage again, see signal flow diagram.

17.1.3.3 Actuator power stage activation/deactivation request

General information:

The actuator power stage is equipped with two separate digital IO-lines for the enable and disable of the output transistors. Both lines must be set to enable for the activation, otherwise the power stage is set to high-resistance.

One enable line is connected to the main controller and the other one is not used due to no safety relevance of actuator valve (i.e. it is connected to supply voltage via resistance).

These actions describe the access to the enable/disable line from the main controller to the actuator power stage.

Requirements for ACTION_INFR_DisableAcrDr:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
-	-	-	-	-	After the ECU reset the actuator power stage must be deactivated !

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The actuator power stage deactivation request has to be performed at once. The infrastructure must be able to deactivate the power stage via enable/disable line itself during the diagnosis.

Requirements for ACTION_INFR_EnableAcrDr:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
-	-	-	-	-	After the IGKON or RST the actuator power stage must be deactivated !

Diagnosis: no electrical diagnosis done here

Coincidence requirements: The actuator power stage activation request has to be performed at once. The power stage activation request may only be performed by the application software level !

17.2 Aggregate adaptation module, ACRC

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
AV_OPEN_EGR	V/O	0...FFH	0...99.60937	0.390625	[%]
Actual opening value of the EGR valve after adaptation					
LV_ERR_EGR_2	V/O	0...1H	0...1	1	[-]
Stuck valve or potentiometer failure					
STK_AV_EGR	V/O	0...FFH	0...99.60937	0.390625	[%]
Actual value of the stroke of EGR valve					
LV_EGR_ACT	V/O	0...1H	0...1	1	[-]
Logical value for EGR-control active					
LV_ER_FDOUT	O	0...1H	0...1	1	[-]
LV_ER_FDOUT					

Input data:

OPG_ACR	LV_PWM_ACR_OFF_REQ	STATE_ACR_CTL	

FUNCTION DESCRIPTION:

This function is used for the adaptation of the actuator control modules to the remaining software system. The Module has to be executed after the ACRC functionality has been calculated.

Application conditions:

Initialization:	at RST set all to zero and LV_ER_FDOUT = 0
Recurrences:	10 ms
Activation:	at all engine operating states
Deactivation:	no deactivation

Formula section:

AV_OPEN_EGR = OPG_ACR

STK_AV_EGR = OPG_ACR

LV_EGR_ACT = **(**STATE_ACR_CTL == PWM_CTL**)**

LV_ERR_EGR_2 = LV_PWM_ACR_OFF_REQ

17.3 ACRC position determination

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
OPG_ACR	O/V	0...FFFH	0...99.9756	0.02441	%
Measured opening of the actuator valve					
V_ACR	O/V	0...3FFH	0...4.995117	0.00488	V
Voltage value for actuator position control delivered from the measuring system					
V_ACR_MV	V	0...7FFH	0...4.997559	0.002441	V
Voltage value for actuator position control delivered from the measuring system with higher resolution					
V_ACR_BAS	V	0...3FFH	0...4.995117	0.00488	V
Raw value of the measuring system for actuator position control					

Input data:

FAC_ACR_SLOP	V_ACR_AD_BOL	LV_OPG_ACR_INI_REQ	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_OPG_ACR_INI	1	0...FFFH	0...99.9756	0.02441	%
Replacement value for sensor signal acquisition					

Import actions:

ACTION_INFR_GetVAcr(OUT <V_acr_bas>**)**
This action provides the standard signal voltages of the ACR measurement system.

17.3.1 General information

The electrical actuator is equipped with a linear position sensor system. The propose of this module is to determine the actual valve position for the closed-loop digital position control.

Application Condition

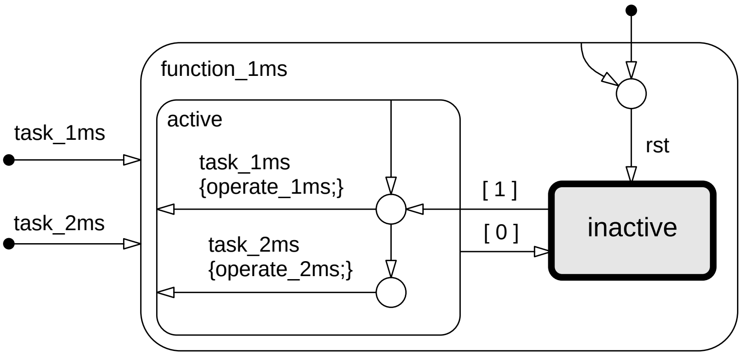


Figure 1 ACRC_position_determination/ APP_CDN/ APP_CDN

Function Description

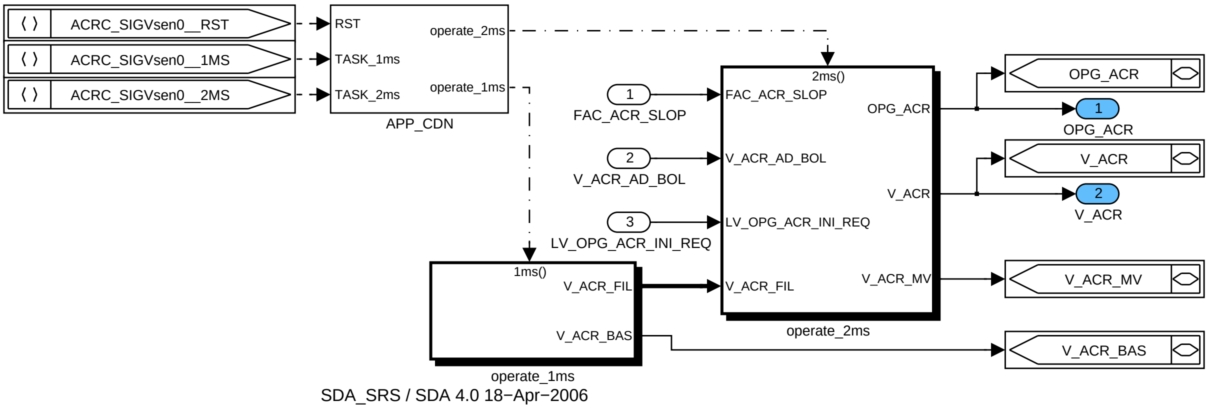


Figure 2 ACRC_POSITION_DETERMINATION

17.3.1.1 Signal pre processing

The described filter function is a part of signal processing. The AD value of the measuring system has a resolution of 10 bit (1024 increments) according to the AD converter. The smallest signification bit (one increment) is the fault of the AD conversion process and this bit can alternate from conversion to conversion. To compensate this behaviour the described bit filter is used, see below.

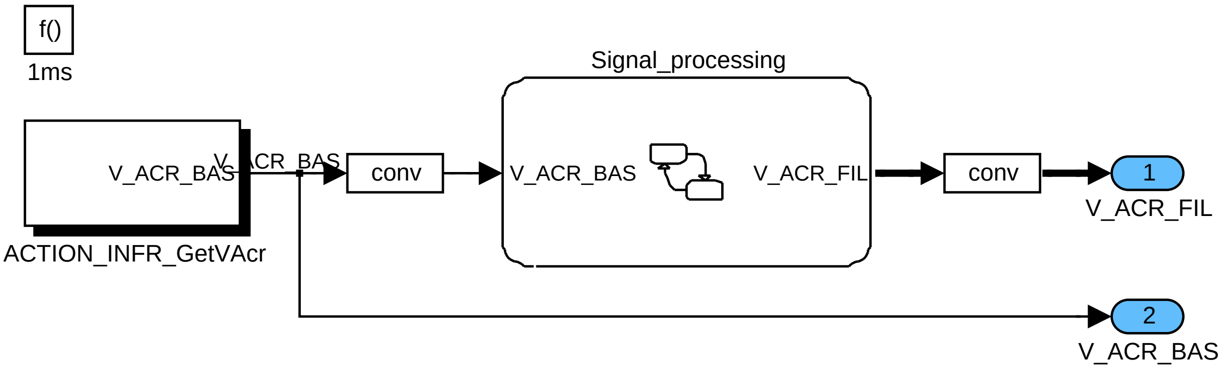


Figure 3 ACRC_POSITION_DETERMINATION/ OPERATE_1MS

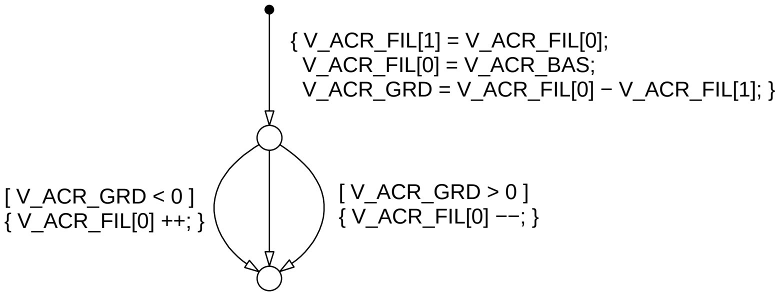


Figure 4 ACRC_position_determination/ operate_1ms/ Signal_processing

17.3.1.2 Signal standardization

The aim of this chapter is to provide the valve position for the digital position controller. The signal voltage of the measuring system is determined by over-sampling and averaging after the signal pre-processing. Afterwards the position signal is standardised by the electrical valve adaptation value and the standard slop of the measuring system and is limited in the range between 0 % and 100 %.

The output of the signal acquisition is set to C_OPG_ACR_INI if the replacement value for the sensor signal acquisition has been requested.

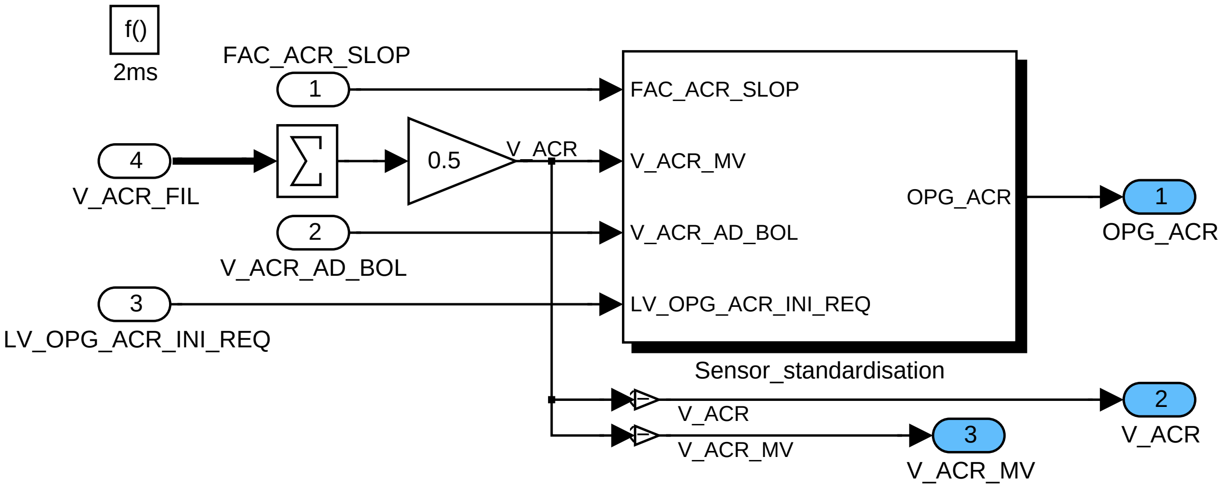


Figure 5 ACRC_POSITION_DETERMINATION/ OPERATE_2MS

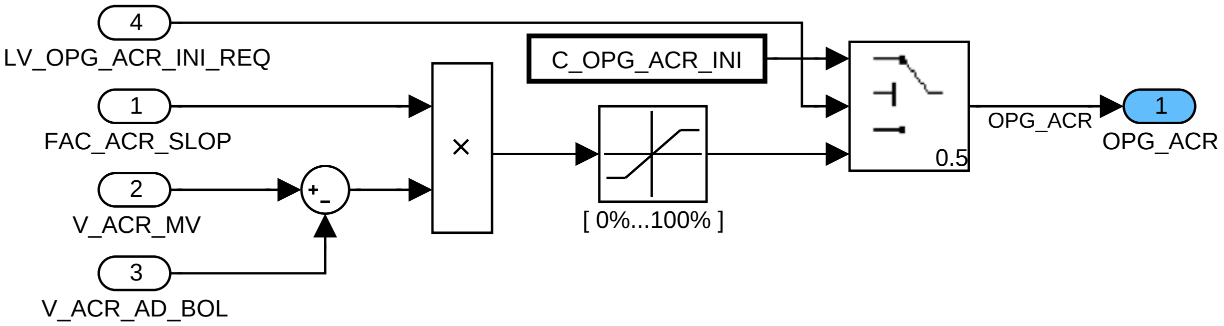


Figure 6 ACRC_POSITION_DETERMINATION/ OPERATE_2MS/ SENSOR_STANDARDISATION

17.4 System variables in the exhaust pipe

17.4.1 Calculation of the exhaust manifold pressure PRS_EX

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PRS_EX	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Pressure in the exhaust system					
PRS_EX_INC	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Increase of exhaust pressure in comparison to ambient pressure					
MAF_FLOW_EX	V/O	0000...FFFFH	0...2047.96875	0.03125	kg/h
air mass flow out of the exhaust manifold					
PRS_EX_EGR	O/V	0000...FFFFH	0...5434	0.083	hPa
Pressure in the exhaust system (for MAF_EGR-Calculation)					
PRS_EX_EGR_INC	O/V	0....FFFFH	0....5434	0.0829	hPa
increase of exhaust pressure in comparison to ambient pressure (for MAF_EGR-Calculation)					

Input data:

MAF_CYL	AMP	LV_RLY_MAIN	NC_CBK_EX_NR
N_32	MFF_SP[NC_CBK_EX_NR]	NC_MAF_FAC_CYL	NC_CYL_NR
LV_IGK			

FUNCTION DESCRIPTION:

There is an increase of the pressure in the exhaust pipe in comparison to the ambient pressure. The increase of the pressure is higher with increasing mass flow in the exhaust pipe. The cylinder mass air flow is direct proportional to the exhaust mass flow. Therefore the pressure increase in the exhaust pipe is a function of the cylinder mass flow.

The exhaust pressure PRS_EX is a function of the ambient pressure and the increase of the pressure in the exhaust pipe. which is a function of the mass flow in the exhaust pipe.

Application conditions:

Initialisation: all = 0 at reset

Recurrence: 1000ms

Activation: LV_IGK = 0 -> 1

Deactivation: LV_RLY_MAIN = 1 -> 0 (also calculated in PWL phase)

Formula section:

PRS_EX_INC = IP_PRS_EX_INC (MAF_CYL)

PRS_EX = PRS_EX_INC + AMP

MAF_FLOW_EX = MAF_CYL / NC_CBK_EX_NR +
(MFF_SP[NC_CBK_EX_NR = 2] *
NC_CYL_NR/NC_CBK_EX_NR * N / NC_MAF_FAC_CYL) –
MAF_EGR

PRS_EX_EGR_INC = IP_PRS_EX_EGR INC (MAF_FLOW_EX)

PRS_EX_EGR = PRS_EX_EGR_INC + AMP

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_PRS_EX_INC	1*8	0....FFFFH	0....5434	0.0829	hPa
LDPM_MAF_CYL_EX	8	0....FFFFH	0....2047.96875	0.03125	kg/h
pressure increase in the exhaust pipe					
IP_PRS_EX_EGR_INC	1*8	0....FFFFH	0....5434	0.0829	hPa
LDPM_MAF_FLOW_EGR	8	0....FFFFH	0....2047.96875	0.03125	kg/h
pressure increase in the exhaust pipe (for MAF_EGR-Calculation)					

17.5 Ignition angle correction due to EGR

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_BAS_EGR_COR	O/V	80...7FH	-48...47.625	0.375	[°CRK]
additive basic ignition angle correction due to EGR influence					
IGA_BAS_EGR_COR_HOM	V	80...7FH	-48...47.625	0.375	[°CRK]
additive basic ignition angle correction due to EGR influence in homogenous mode					
IGA_BAS_EGR_COR_S	V	80...7FH	-48...47.625	0.375	[°CRK]
additive basic ignition angle correction due to EGR influence in stratified mode					
EGR_RATIO_COR_2	V	0...FFFFH	0...99.99847	1.5259e-3	[%]
corrected EGR_RATIO for LDP of IP_IGA_BAS_EGR					

Input data:

MAF	N_32	LV_EGR_ACT	TCO
MFF_SP_S	LV_SP_RATE_CYL_EGR_SWI	EGR_RATIO_SP	LAMB_SP

FUNCTION DESCRIPTION:

General information:

Homogeneous mode:

In homogeneous mode, if EGR function is active, the basic ignition angle IGA_BAS_IVVT is improved by the additive EGR correction value IGA_BAS_EGR_COR. This value depends from N_32, MAF, TCO, EGR_RATIO_SP and LAMB_SP.

Stratified mode:

In stratified mode, if EGR function is active, the basic ignition angle IGA_BAS_S is improved by the additive EGR correction value IGA_BAS_EGR_COR. IGA_BAS_EGR_COR is calculated from ~~IP_IGA_EGR_S corrected by~~ the coolant temperature TCO (IP_IGA_EGR_S).

Application conditions:

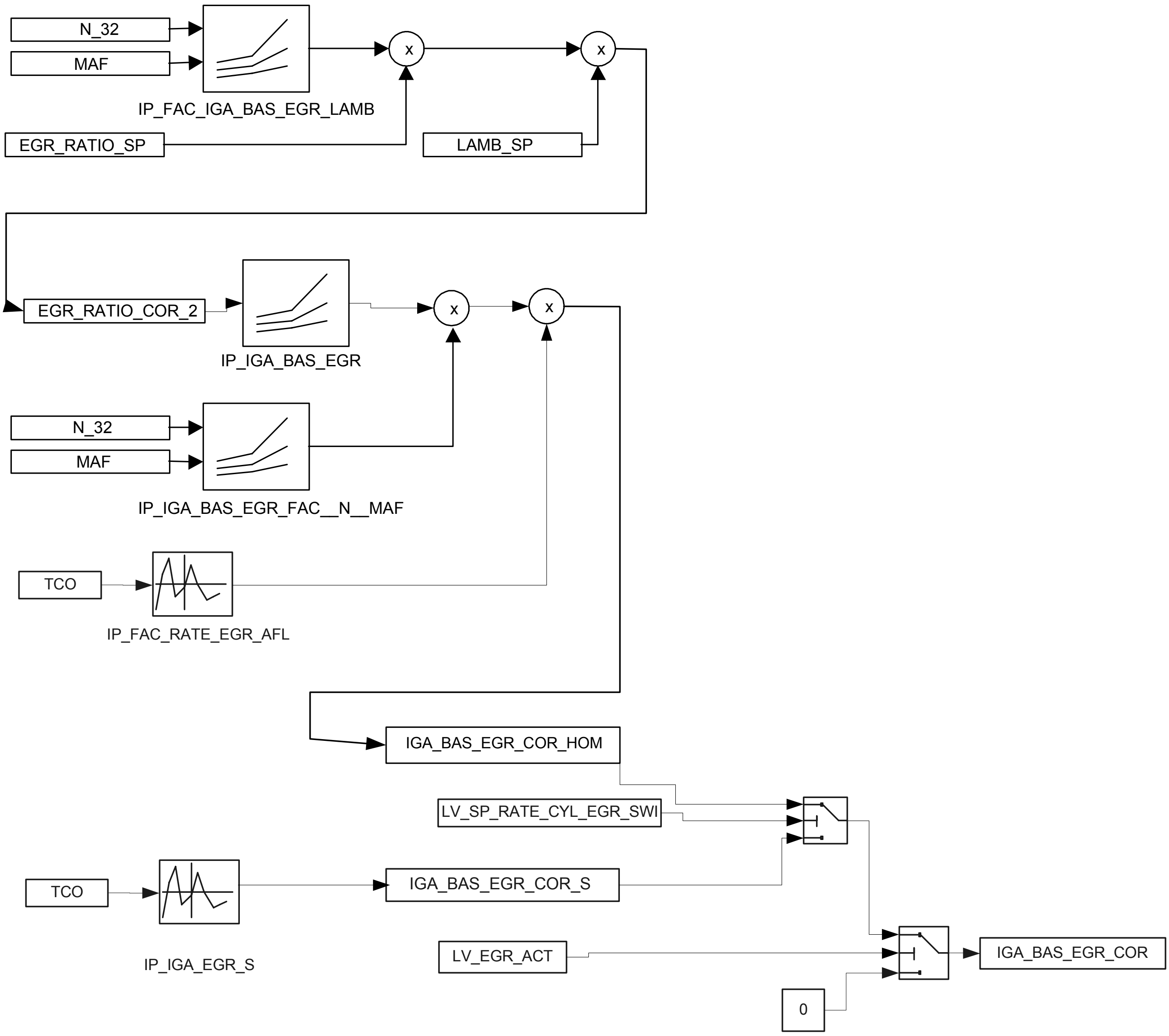
Activation: LV_EGR_ACT = 1

Deactivation: LV_EGR_ACT = 0

Update Rate: 10 ms

Initialization: IGA_BAS_EGR_COR = 0

Signal flow diagram:



Formula section:

EGR_RATIO_COR_2 = EGR_RATIO_SP * IP_FAC_IGA_BAS_EGR_LAMB * LAMB_SP

IGA_BAS_EGR_COR_HOM = IP_IGA_BAS_EGR *
IP_IGA_BAS_EGR_FAC__N__MAF *
IP_FAC_RATE_EGR_AFL

IGA_BAS_EGR_COR_S = ~~IP_IGA_EGR_S *~~ IP_IGA_EGR_S

```
if     LV_EGR_ACT = 0
  then       IGA_BAS_EGR_COR = 0
  elseif     LV_SP_RATE_CYL_EGR_SWI = 1
     then      IGA_BAS_EGR_COR = IGA_BAS_EGR_COR_HOM
     else      IGA_BAS_EGR_COR = IGA_BAS_EGR_COR_S
```

```
  endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_IGA_BAS_EGR_LAMB	12x12	0...FFH	0...0.996	0.0039063	-
LDPM_N_32_IP_EGR_RATIO	12	0...FFH	0...8160	32	rpm
LDPM_MAF_IP_EGR_RATIO	12	0...FFFFH	0...1389	0.0212	mg/stk
factor for EGR_RATIO list of datapoints					
IP_IGA_BAS_EGR	6x1	0...FFH	-48 ... 47.625	0.375	°CRK
LDP_EGR_RATIO_COR_2__IGA_BAS	6	0...FFFFH	0...99.998	1.525E-3	%
basis for EGR correction IGA_BAS					
IP_IGA_BAS_EGR_FAC__N__MAF	12x12	0...FFH	0...1.992	0.007813	-
LDPM_N_32_IP_EGR_RATIO	12	0...FFH	0...8160	32	rpm
LDPM_MAF_IP_EGR_RATIO	12	0...FFFFH	0...1389	0.0212	mg/stk
IP_FAC_RATE_EGR_AFL	5	0...FFH	0...1.992	0.007813	[-]
LDPM_TCO__IGA _EGR	5	0...FEH	-48...142.5	0.75	°C
temperature corrective factor					
IP_IGA_EGR_S	5	0...FFH	-48 ... 47.625	0.375	°CRK
LDPM_TCO__IGA_EGR	5	0...FEH	-48...142.5	0.75	[°C]
Temperature based correction (stratified mode)					

17.6 ACRC setpoint selection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
OPG_SP_ACR	O/V	0...FFFH	0...99.9756	0.02441	%
Setpoint request after limitation for actuator position control					
OPG_SP_SEL_ACR	V	0...FFFH	0...99.9756	0.02441	%
Setpoint request after setpoint selection for actuator position control					

Input data:

OPG_SP_REQ_ACR	LV_OPG_SP_LIH_ACR	OPG_SP_LIH_ACR	LV_OPG_SP_AD_ACR
OPG_SP_AD_ACR	LV_OPG_SP_EXT_ACR	OPG_SP_EXT_ACR	LV_OPG_SP_POP_ACR
OPG_SP_POP_ACR			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_OPG_SP_MAX_ACR	1	0...FFFH	0...99.9756	0.02441	%
Upper limitation threshold of the actuator position setpoint					
C_OPG_SP_MIN_ACR	1	0...FFFH	0...99.9756	0.02441	%
Lower limitation threshold of the actuator position setpoint					

17.6.1 General information

This module describes the generation of the actuator valve position setpoint for closed – loop digital position control. All setpoint requests delivered from the remaining software system are gathered, coordinated and filially limited in the physical limits of the actuator valve. At the end of the module the position setpoint is provided for the digital position controller directly.

Application Condition

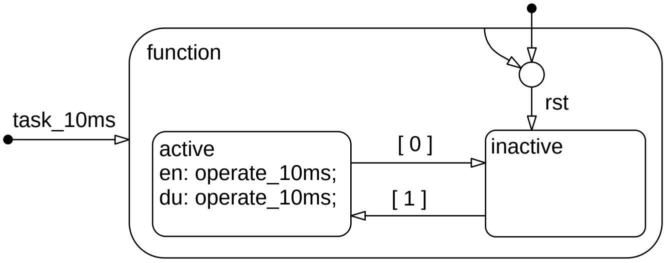


Figure 7 ACRC_setpoint_selection/ APP_CDN/ APP_CDN

Function Description

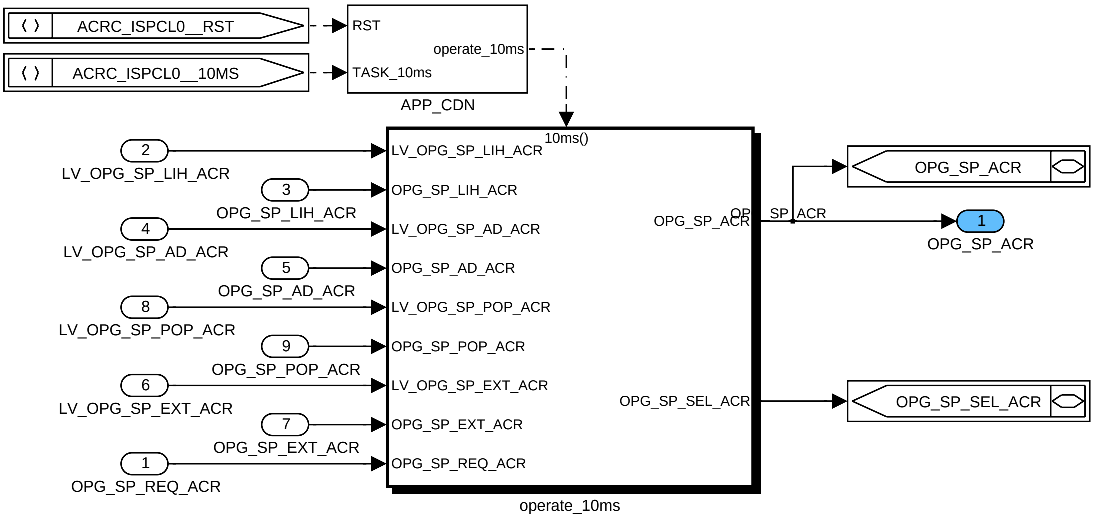


Figure 8 ACRC_setpoint_selection

17.6.1.1 ACRC setpoint functionality

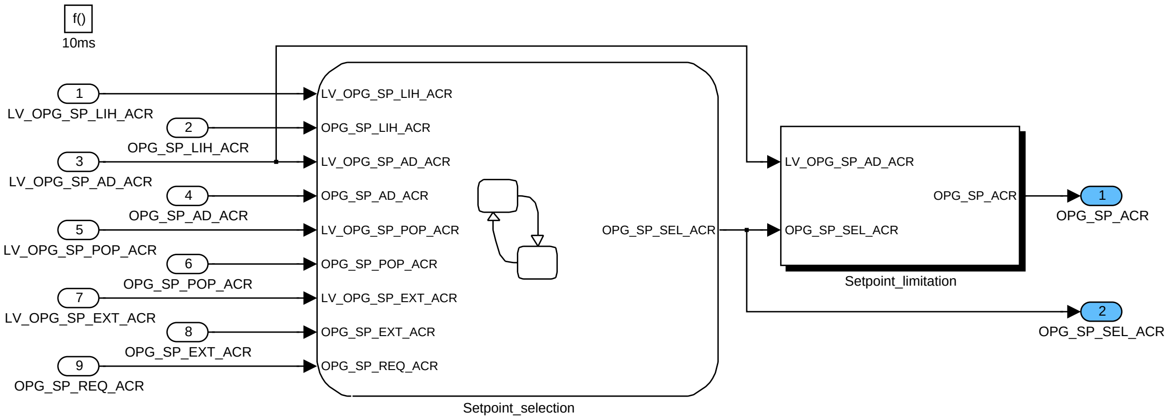


Figure 9 ACRC_setpoint_selection/ operate_10ms

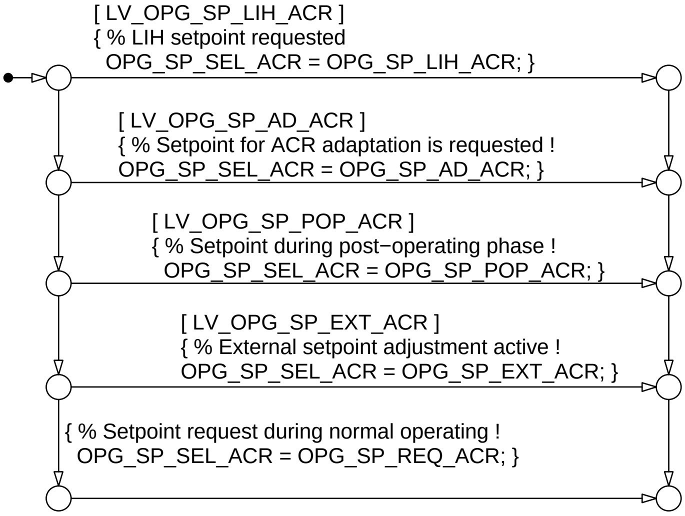


Figure 10 ACRC_setpoint_selection/ operate_10ms/ Setpoint_selection

Setpoint limitation

During the setpoint request from the actuator adaptation function the setpoint is not limited in increasing direction, the maximum value is 100 %.

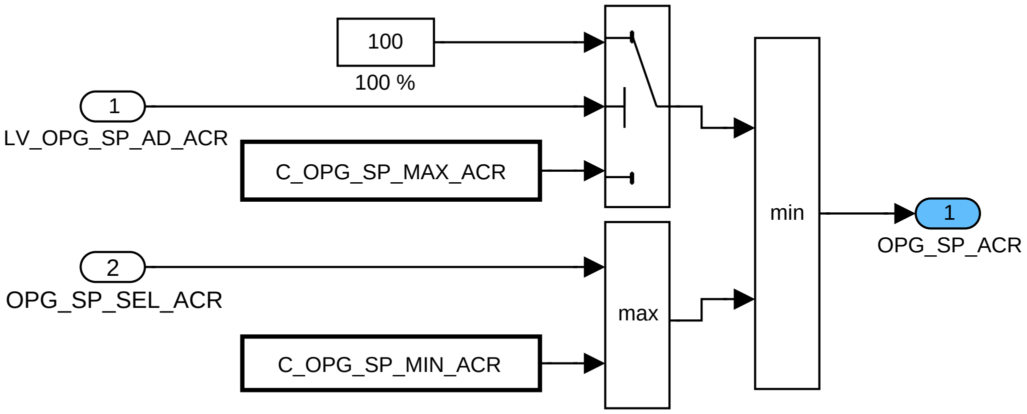


Figure 11 ACRC_setpoint_selection/ operate_10ms/ Setpoint_limitation

17.7 ACRC setpoint selection (appl. inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_OPG_SP_LIH_ACR	O/V	0...1H	0...1	1	-
Logical variable indicates a LIH setpoint request for actuator position control					
OPG_SP_LIH_ACR	O/V	0...FFFH	0...99.9755859	0.02441	%
Limp-home position setpoint for actuator position control					
LV_OPG_SP_EXT_ACR	O/V	0...1H	0...1	1	-
Logical variable indicates a external setpoint request for actuator position control					
OPG_SP_EXT_ACR	O/V	0...FFFH	0...99.9755859	0.02441	%
External position setpoint for actuator position control					
LV_OPG_SP_POP_ACR	O/V	0...1H	0...1	1	-
Logical variable indicates a setpoint request during the post operating phase for actuator position control					
OPG_SP_POP_ACR	O/V	0...FFFH	0...99.9755859	0.02441	%
Position setpoint during post operating phase for actuator position control					
STATE_ACR_CTL_REQ	O/V	0H 1H 2H	PWM_OFF PWM_STAT PWM_CTL	1	-
The state variable indicates the request for digital actuator control					

Input data:

OPG_SP_REQ_ACR	LV_IGK	LV_ES	LV_OPG_SP_AD_ACR
STATE_ACR_CTL	LV_INH_ACR_CTL_AD	LV_OPG_SP_ACR_EXT_ADJ	OPG_SP_ACR_EXT_ADJ

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_DLY_ACR_CTL	1	0...FFH	0...255	1	-
Time delay for the deactivation of the actuator position control					
C_CTR_OPG_SP_POP_ACR	1	0...FFH	0...255	1	-
Delay counter for the deactivation of the position controller in the post-operating phase					
C_OPG_SP_EXT_ACR	1	0...FFFH	0...99.9755859	0.02441	%
External position setpoint requested from the calibration system					
C_OPG_SP_LIH_ACR	1	0...FFFH	0...99.9755859	0.02441	%
Position setpoint request during limp-home mode has been activated					
C_OPG_SP_MIN_ACR_CTL	1	0...FFFH	0...99.9755859	0.02441	%
Deactivation threshold for actuator position control in normal operation					
C_OPG_SP_POP_ACR	1	0...FFFH	0...99.9755859	0.02441	%
Position setpoint request in the post-operating phase					
LC_OPG_SP_EXT_ACR	1	0...1H	0...1	1	-
Activation of the external setpoint adjustment made with the calibration system					

17.7.1 General information

The following module shall provide all remaining actuator valve position setpoint requests for all engine operating states. At the end of the function the power stage activation condition according to position setpoint request is calculated.

Application Condition

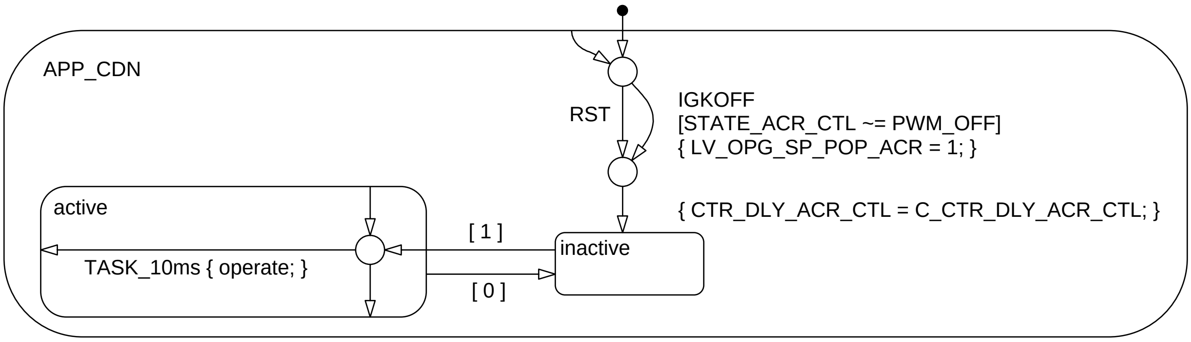


Figure 12 ACRC_setpoint_selection_ai/ APP_CDN/ APP_CDN

Function Description

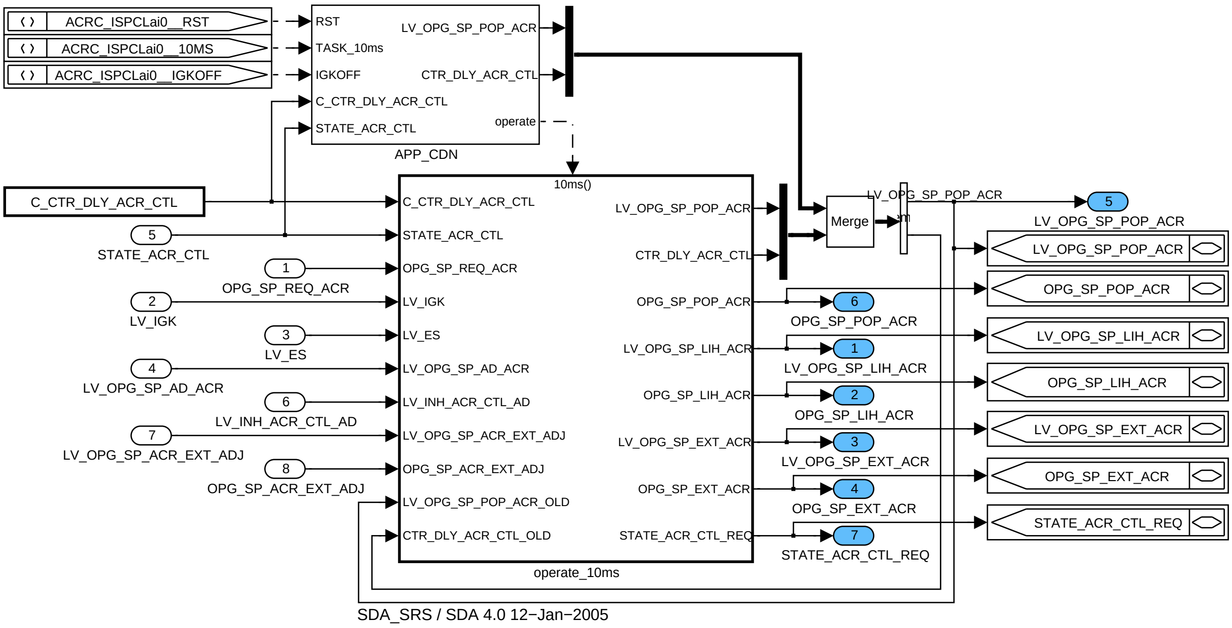


Figure 13 ACRC_setpoint_selection_ai

17.7.1.1 Setpoint generation

The module defines the following setpoint requests: Limp - home position setpoint request active with deactivated power stage, external position setpoint request made with the calibration system and the setpoint request setpoint during post - operating phase after ignition off. During the post - operating phase the setpoint is set to C_OPG_SP_POP_ACR and will be removed first after a defined calibration time.

At the end of the function the power stage activation request is performed. In normal operating the transition between active and passive closed - loop control is calculated here. Closed - loop control will be finished after a defined delay time and position setpoint requests lower than C_OPG_SP_EXT_ACR. With passive closed - loop control it is possible to deactivate the power stage via disable line or to close the actuator valve with output of a constant PWM duty - cycle. The decision will be done with the logical configuration constant LC_PWM_ACR_OFF_ENA !

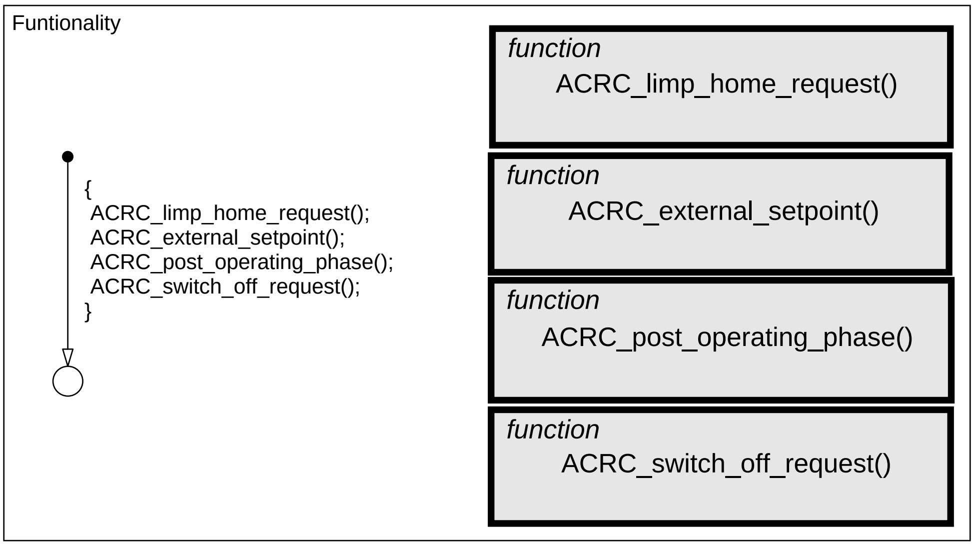


Figure 14 ACRC_setpoint_selection_ai/ operate_10ms/ Chart

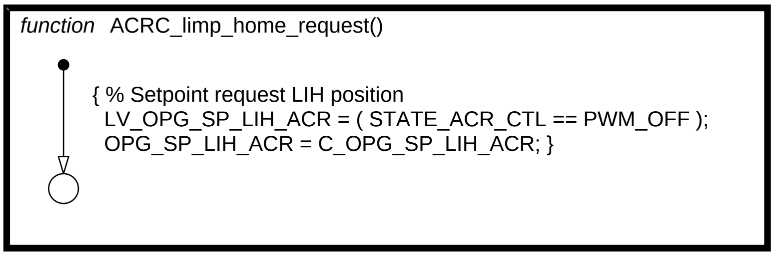


Figure 15 ACRC_setpoint_selection_ai/ operate_10ms/ Chart/ Funtionality/ ACRC_limp_home_request

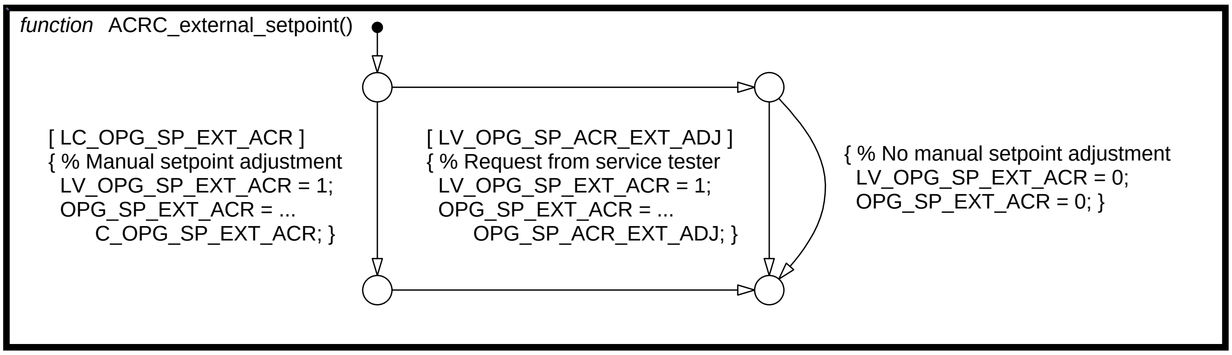


Figure 16 ACRC_setpoint_selection_ai/ operate_10ms/ Chart/ Funtionality/ ACRC_external_setpoint

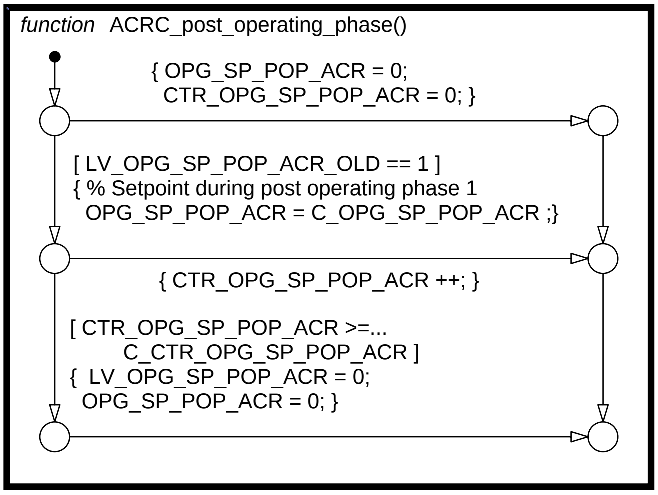


Figure 17 ACRC_setpoint_selection_ai/ operate_10ms/ Chart/ Funtionality/ ACRC_post_operating_phase

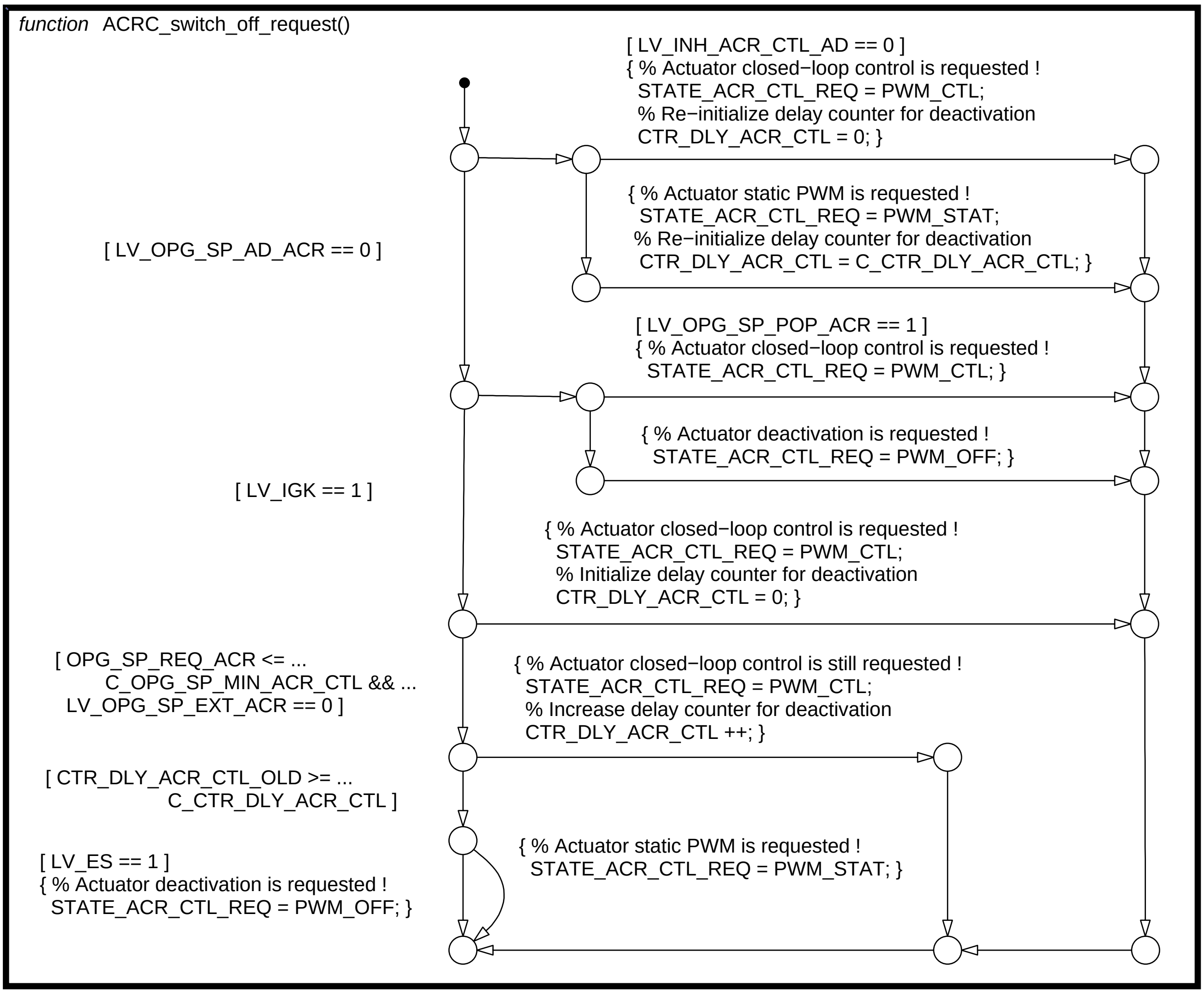


Figure 18 ACRC_setpoint_selection_ai/ operate_10ms/ Chart/ Funtionality/ ACRC_switch_off_request

17.8 ACRC position controller

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PWM_ACR	O/V	8000...7FFFH	-100 ... 99.9969	0.003052	%
Finally duty cycle of digital actuator control					
OPG_DIF_ACR	O/V	F000...FFFH	-100 ... 99.9756	0.02441	%
Control deviation of the digital actuator control					
FAC_VB_PWM_ACR	V	0...800H	0...2	9.7656E-4	-
Factor for battery voltage correction					
OPG_SP_CTL_ACR	V	0...FFFH	0...99.9755859	0.02441	%
Setpoint after prefilter for digital actuator control					

Input data:

OPG_SP_ACR	OPG_ACR	STATE_ACR_CTL	VB_MMV
LV_ACR_AD_ACT			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_C_1_ACR_CTL	1	8000...7FFFH	-4...3.99987793	1.2207E-4	-
Filter parameter c1 of the ACR position controller					
C_FAC_C_2_ACR_CTL	1	8000...7FFFH	-4...3.99987793	1.2207E-4	-
Filter parameter c2 of the ACR position controller					
C_FAC_C_3_ACR_CTL	1	8000...7FFFH	-4...3.99987793	1.2207E-4	-
Filter parameter c3 of the ACR position controller					
C_FAC_F_1_ACR_CTL	1	8000...7FFFH	-4...3.99987793	1.2207E-4	-
Filter parameter f1 of the ACR position controller					
C_FAC_F_2_ACR_CTL	1	8000...7FFFH	-4...3.99987793	1.2207E-4	-
Filter parameter f2 of the ACR position controller					
C_FAC_F_3_ACR_CTL	1	8000...7FFFH	-4...3.99987793	1.2207E-4	-
Filter parameter f3 of the ACR position controller					
C_FAC_K_FIL_ACR_CTL	1	0...FFFFH	0...0.99998474	1.5259E-5	-
Pre-filter amplification of the ACR position controller					
C_FAC_OPG_SP_ACR	1	0...FFH	0...0.99609375	0.003906	-
Weigthing factor for setpoint rate limitation of the ACR position controller					
C_FRQ_PWM_ACR	1	0...FFFFH	0...6.5535E+3	0.1	Hz
Frequency of the PWM signal used for closed-loop actuator control					
C_KP_ACR_CTL	1	0...400H	0...64	0.0625	-
Amplification factor of the ACR position controller					
C_PWM_ACR_CLOSE	1	8000...7FFFH	-100 ... 99.9969	0.003052	%
Static PWM-duty cycle for the closing of the actuator					
C_PWM_MAX_ACR_AD	1	8000...7FFFH	-100 ... 99.9969	0.003052	%
Maximum value of the position controller output during actuator adaptation					
C_PWM_MAX_ACR_CTL	1	8000...7FFFH	-100 ... 99.9969	0.003052	%
Upper limitation threshold of the ACR position controller					
C_PWM_MIN_ACR_CTL	1	8000...7FFFH	-100 ... 99.9969	0.003052	%
Lower limitation threshold of the ACR position controller					
C_RATE_OPG_SP_ACR	1	0...FFFH	0...99.9755859	0.02441	%
Setpoint rate limitation threshold of the ACR position controller					
IP_FAC_VB_PWM_ACR	4	0...800H	0...2	9.7656E-4	-
LDP_VB_MMV_IP_FAC_VB_PWM_ACR	4	0...FFH	0...25.8984375	0.1015625	V
Factor for battery voltage correction					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_PWM_ACR_STAT	4	0...FFFFH	-100 ... 99.9969	0.003052	%
LDP_OPG_SP_ACR_IP_PWM_ACR_STAT	4	0...FFFH	0...99.9755859	0.02441	%
Feedforward control of the ACR position controller					

Import actions:

ACTION_INFR_SetPwmDucyAcr (IN <Pwm_acr> **)**
This action adjusts the duty-cycle and the current direction of the ACR power stage output.
ACTION_INFR_SetPwmFrqAcr(IN <Frq_pwm_acr>**)**
This action adjusts the frequency of the PWM signal from the PWM signal generator

17.8.1 General information

This module describes the digital position controller used for the closed-loop control of an electrical actuator. The position controller is carried out as a digital robust position controller. Calibration data are determined according to SiemensVDO internal position controller design algorithms. The controller algorithm is split in a 2 ms functionality for closed-loop position control and a 10 ms functionality for static preload, battery voltage correction and position setpoint rate limitation.

Application Condition

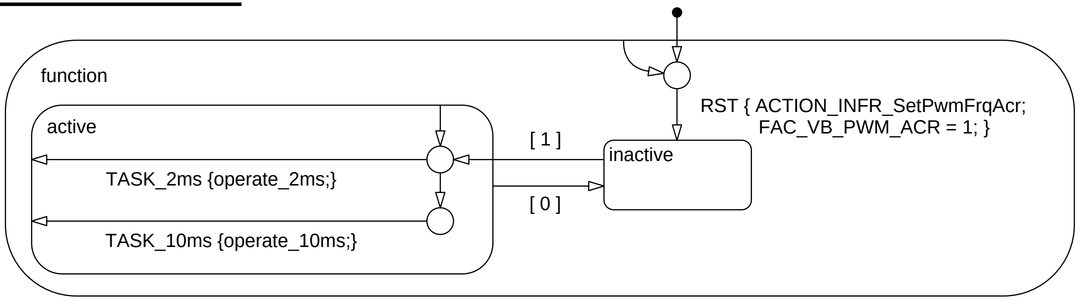


Figure 19 ACRC_position_controller/ APP_CDN/ APP_CDN

Function Description

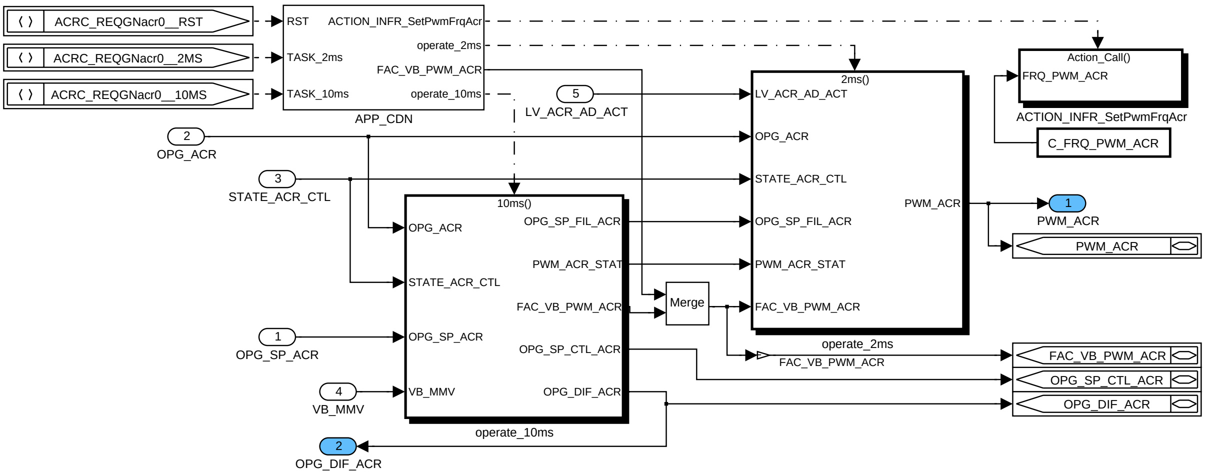


Figure 20 ACRC_position_controller

17.8.1.1 Functional part of 10ms

The 10 ms functionality of the position controller is only calculated if the closed-loop position control has been requested. Closed-loop position control will be indicated by the state PWM_CTL (= 2) of the state variable STATE_ACR_CTL.

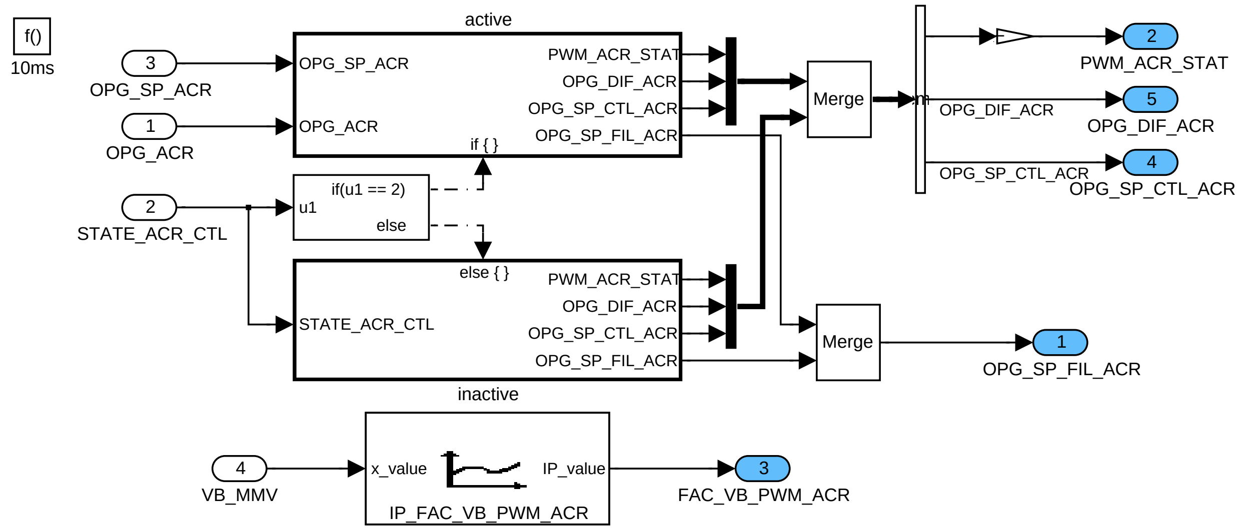


Figure 21 ACRC_position_controller/ operate_10ms

ACR position controller

This function part contains the calculation of the battery correction factor, the static pre - load of the position controller output and a final position setpoint filter function as a part of the closed - loop position control. In the inactive state of the position control algorithm the battery voltage correction factor is set to 1, the static preload is set to C_PWM_ACR_CLOSE and all other values are set to zero.

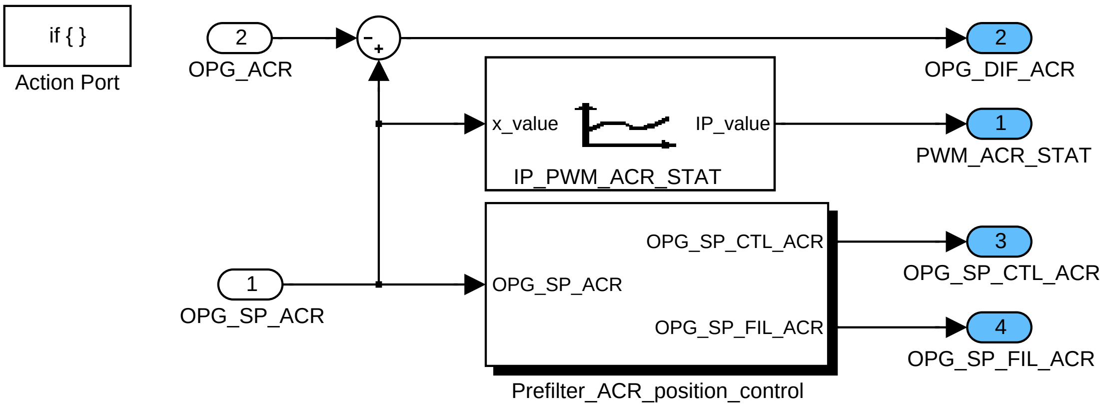


Figure 22 ACRC_position_controller/ operate_10ms/ active

Pre–filter functionality

The position setpoint filter function consists of a rate limitation functionality and a weighting between the described setpoint rate limitation and the direct setpoint request. The weighting can be chosen by the calibration constant C_FAC_OPG_SP_ACR. Finally the setpoint request for the digital position controller is corrected by the static transfer amplification factor C_FAC_K_FIL_ACR_CTL of the position controller filter function.

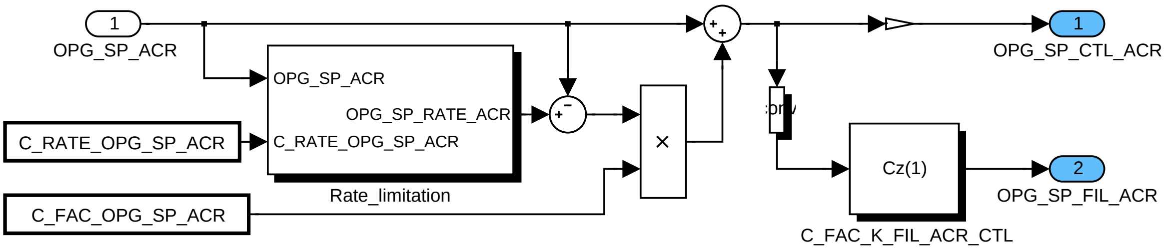


Figure 23 ACRC_position_controller/ operate_10ms/ active/ Prefilter_ACR_position_control

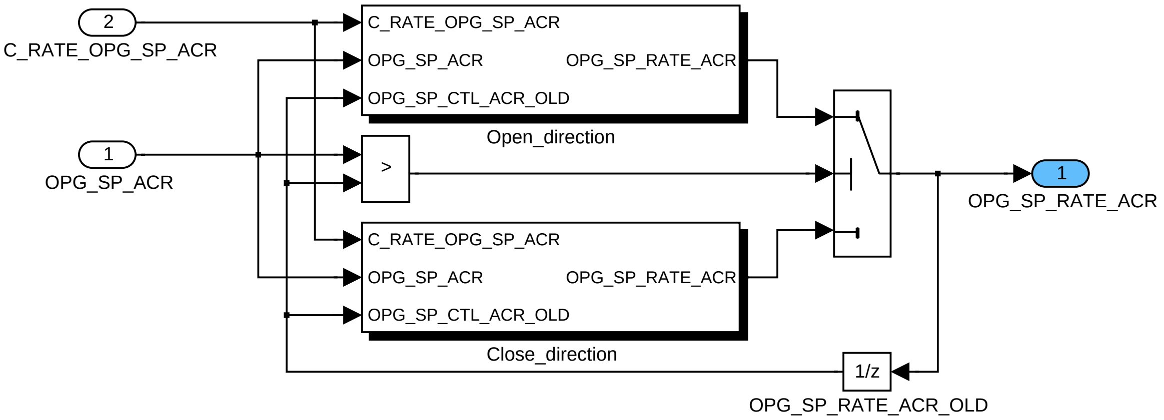


Figure 24 ACRC_position_controller/ operate_10ms/ active/ Prefilter_ACR_position_control/ Rate_limitation

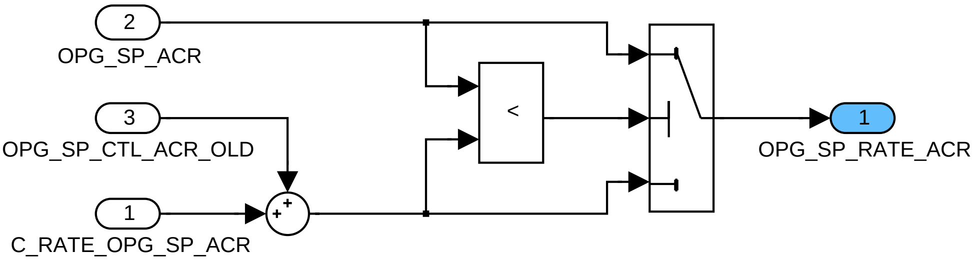


Figure 25 ACRC_position_controller/ operate_10ms/ active/ Prefilter_ACR_position_control/ Rate_limitation/ Open_direction

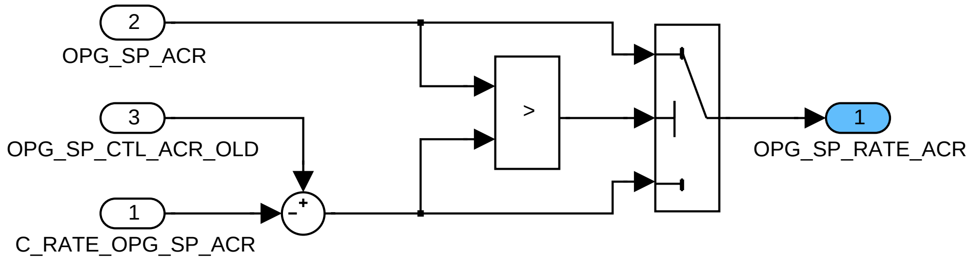
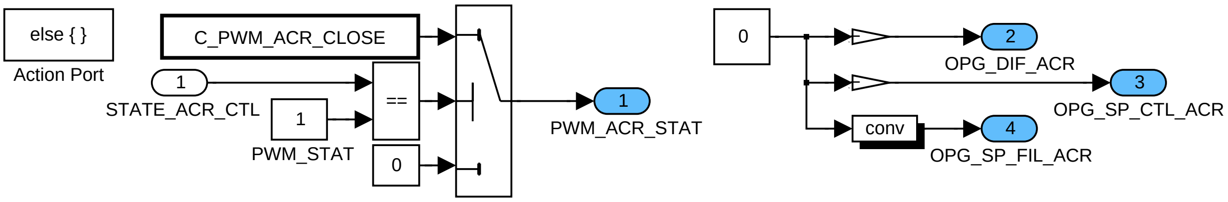


Figure 26 ACRC_position_controller/ operate_10ms/ active/ Prefilter_ACR_position_control/ Rate_limitation/ Close_direction



Note: The state variable OPG_SP_RATE_ACR_OLD from the active part is also set to zero !

Figure 27 ACRC_position_controller/ operate_10ms/ inactive

17.8.1.2 Functional part of 2ms

The 2 ms functionality of the position controller is only calculated if the closed-loop position control has been requested. Closed-loop position control will be indicated by the state PWM_CTL (= 2)of the state variable STATE_ACR_CTL.

After the calculation of the position controller the static preload and the battery voltage correction are considered. Finally the PWM duty-cycle of the position controller output will be limited between -100 % and 100 %.

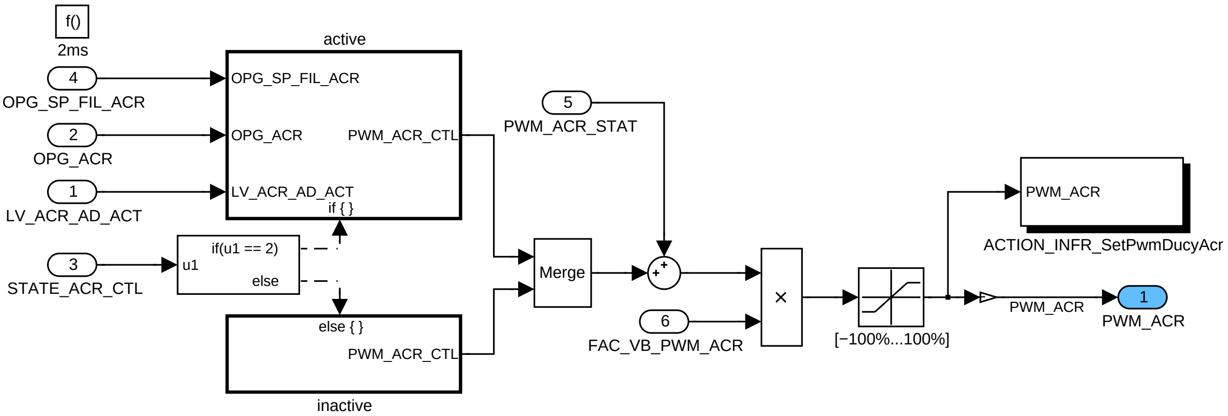
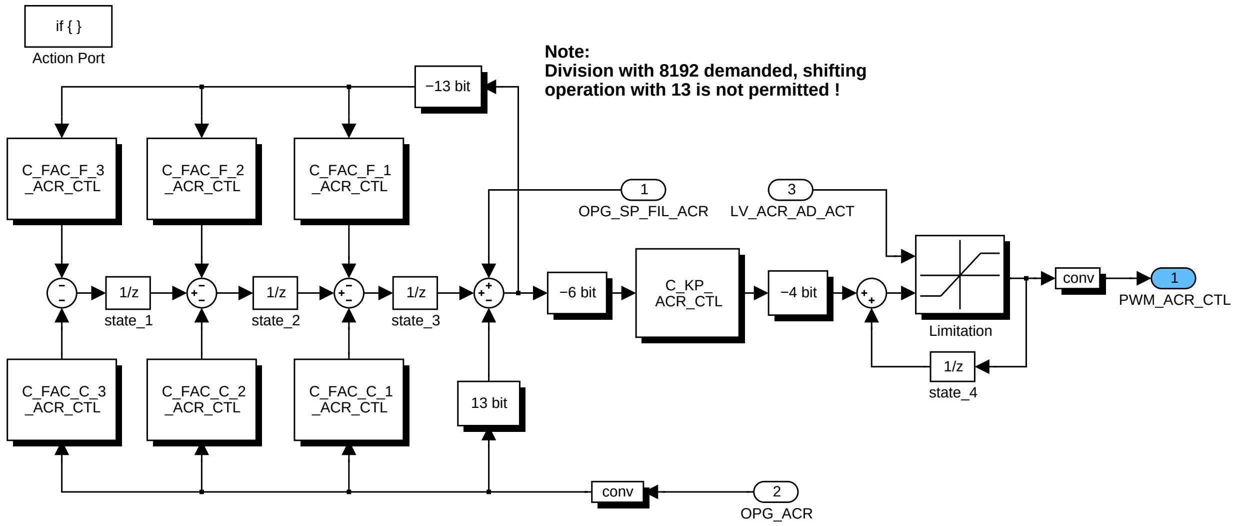


Figure 28 ACRC_position_controller/ operate_2ms

ACR position controller

The position control algorithm is a robust digital position controller and based on the pole placement design. The position controller consists of a filter function third order in backwards direction and a amplification factor with integrated integrator in forward direction. The output of the integrator is limited between a lower and upper limitation threshold. In the inactive state of the controller algorithm all state variables (state_x, x = 1...4) and the controller output are set to zero.

Calibration data are determined according to SiemensVDO internal position controller design algorithms.



Note:
Division with 8192 demanded, shifting operation with 13 is not permitted !

PWM_ACR = KP * z / (z − 1) * [C(1) / F(1) * OPG_SP_ACR − C(z) / F(z) * OPG_ACR]

C_FAC_x_y_ACR_CTL : HEX = 8000...7FFFH, DEC = −4...4, RES = 2^−13 = 1.2207e−4

C_KP_ACR_CTL : HEX = 0...400H, DEC = 0...64, RES = 2^−4 = 0.0625

Figure 29 ACRC_position_controller/ operate_2ms/ active

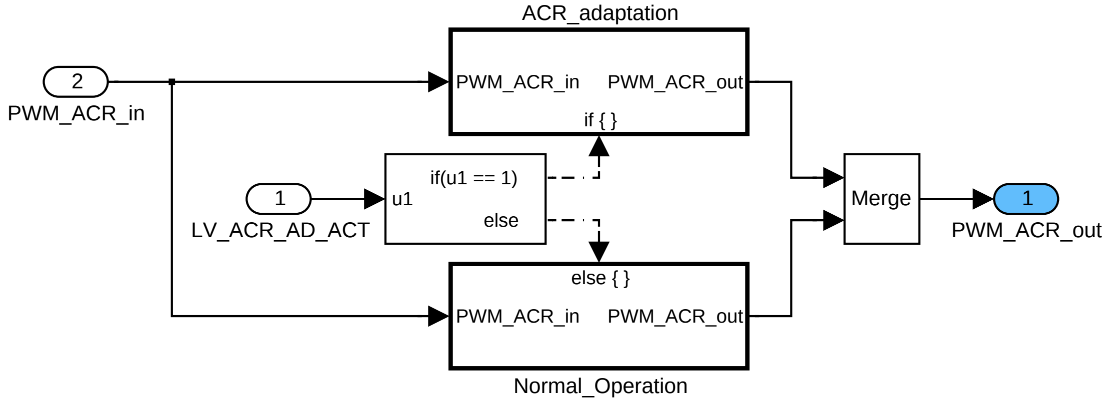


Figure 30 ACRC_position_controller/ operate_2ms/ active/ Limitation

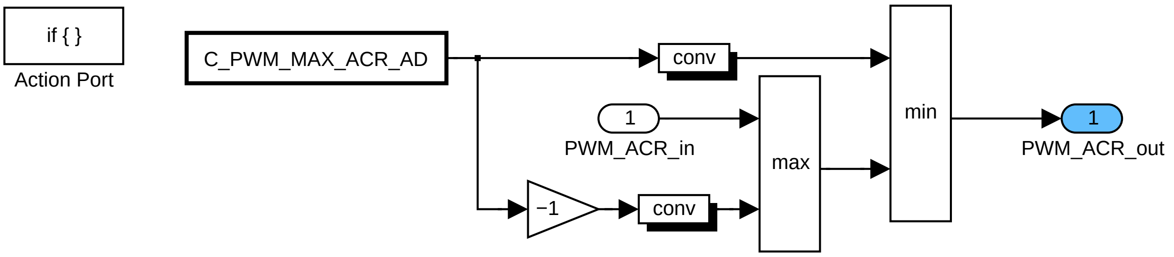


Figure 31 ACRC_position_controller/ operate_2ms/ active/ Limitation/ ACR_adaptation

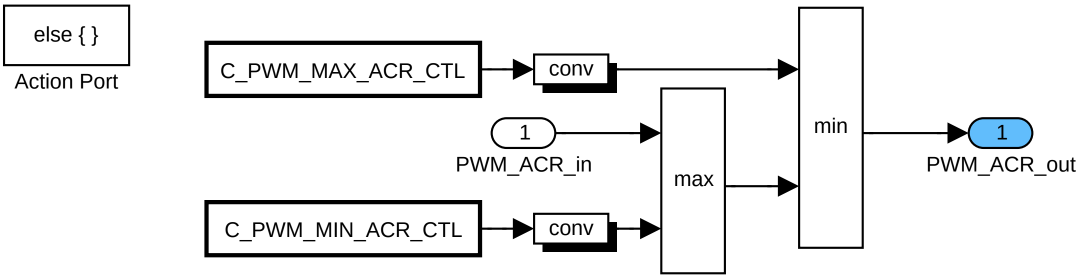
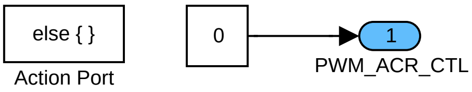


Figure 32 ACRC_position_controller/ operate_2ms/ active/ Limitation/ Normal_Operation



Note: All state variables (state_1...4) from the ACR position controller are set to zero in the inactive state !

Figure 33 ACRC_position_controller/ operate_2ms/ inactive

17.9 ACRC position controller (appl. inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_ACR_CTL	O/V	0H 1H 2H	PWM_OFF PWM_STAT PWM_CTL	1	-
The state variable indicates the state of the ACR position control					

Input data:

STATE_ACR_CTL_REQ	LV_PWM_ACR_OFF_REQ		

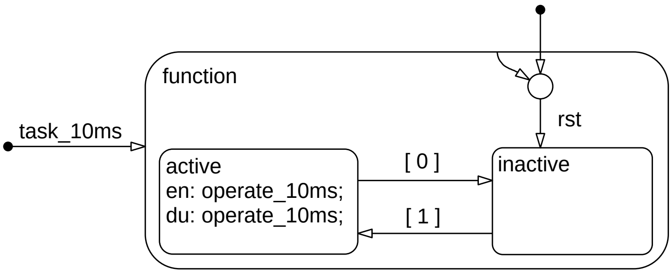
Import actions:

ACTION_INFR_DisableAcrDr( )
This action is able to switch-off the ACR power stage.
ACTION_INFR_EnableAcrDr( )
This action is able to switch-on the ACR power stage.

17.9.1 General information

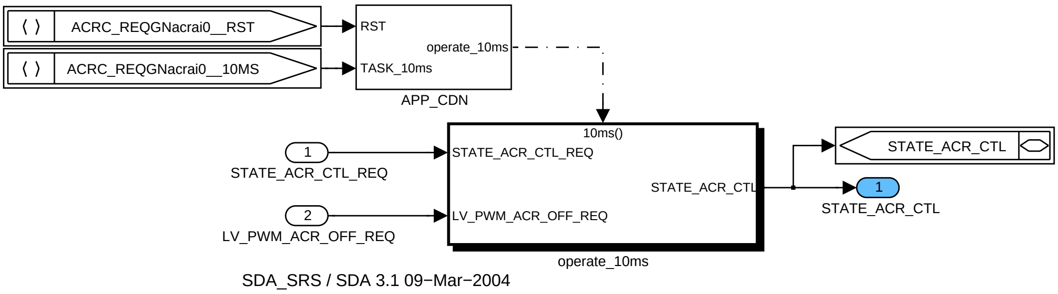
The propose of this module is to coordinate all states of closed – loop position control. The state variable STATE_ACR_CTL indicates the current state of the actuator systems. The operating can be divided in the power stage is deactivated via disable line, the actuator is controlled means of static PWM duty-cycle for active valve closing and the closed – loop position control.

Application Condition



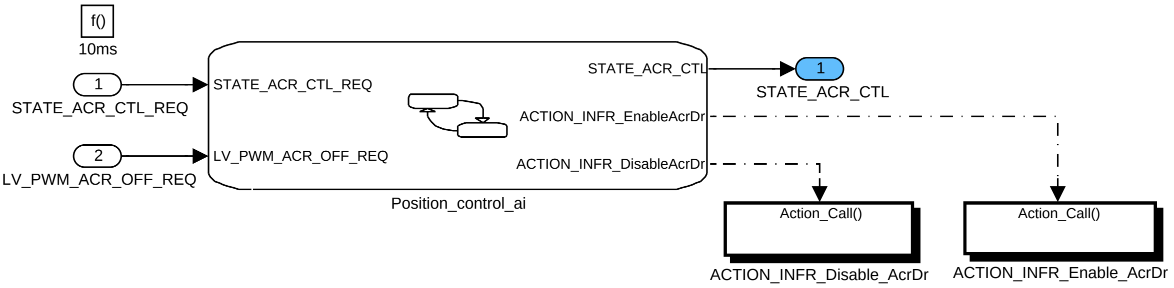
ACRC_position_controller_ai/APP_CDN/APP_CDN

Function Description

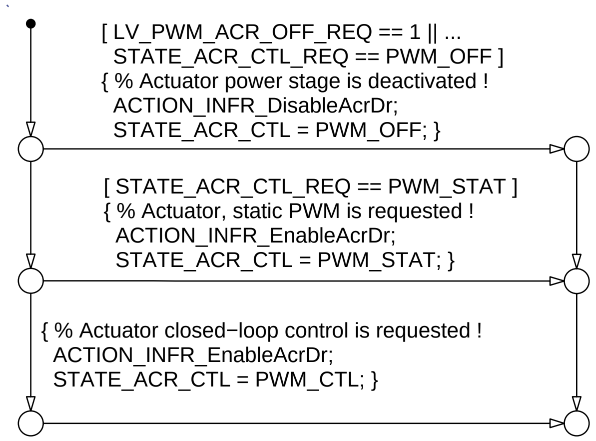


ACRC_position_controller_ai

17.9.1.1 Power stage coordination



ACRC_position_controller_ai/operate_10ms



ACRC_position_controller_ai/operate_10ms/Position_control_ai

17.10 Exhaust gas recirculation (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ACR_CTL_ENA	O/V	0...1H	0...1	1	[-]
Logical variable for the release of ACR valve digital position control					

Input data:

TCO	LV_FL	LV_IS	T_AST
LV_IS	LV_PL	N	LV_ERR_MAF
LV_ERR_MAP	LV_ERR_KNKS[NC_NR_SENS_KNK]	LV_PWM_ACR_OFF_REQ	LV_ERR_MAP_DIP_SHIFT
LV_ERR_TPS_1	LV_ERR_TPS_2	LV_ERR_TPS_RATIO	LV_ERR_TPS_AD
LV_ERR_TPS_AD_BOL	LV_ERR_TPS_MAF_1	LV_ERR_TPS_MAF_2	LV_ERR_TPS_ST_CHK_2
LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3	LV_ERR_MTC_DR
LV_ERR_ACR_AD	LV_ERR_ACR_CTL	LV_ERR_ACR_DR	LV_ERR_MAP_TPS_PLAUS

General information:

The EGR-Control is activated (LV_ACR_CTL_ENA = 1), if the coolant temperature TCO is higher than a threshold value C_TCO_MIN_EGR and the engine is running at part load or idle speed (LV_FL=0). If idle speed is detected the EGR-function is activated after the time T_AST> C_T_AST_EGR.

Application conditions:

Initialisation: at RST set LV_ACR_CTL_ENA to zero !

Recurrence: 20ms

Activation: at all engine operating states

Deactivation: no deactivation

Formula section:

```
if      TCO > C_TCO_MIN_EGR                          and
        LV_ERR_MAF                == 0               and
        LV_ERR_MAP                == 0               and
        LV_ERR_KNKS_i             == 0               and
        LV_FL                     == 0               and
        LV_PWM_ACR_OFF_REQ        == 0               and
        LV_ERR_MAP_DIP_SHIFT      == 0               and
        LV_ERR_TPS_1              == 0               and
        LV_ERR_TPS_2              == 0               and
        LV_ERR_TPS_RATIO          == 0               and
        LV_ERR_TPS_AD             == 0               and
        LV_ERR_TPS_AD_BOL         == 0               and
        LV_ERR_TPS_MAF_1          == 0               and
        LV_ERR_TPS_MAF_2          == 0               and
```

```
           LV_ERR_TPS_ST_CHK_2      == 0                          and
           LV_ERR_LOAD_TPS_PLAUS == 0                             and
           LV_ERR_MTC_CTL_2          == 0                         and
           LV_ERR_MTC_CTL_3          == 0                         and
           LV_ERR_MTC_DR             == 0                         and
           LV_ERR_ACR_AD             == 0                         and
           LV_ERR_ACR_CTL            == 0                         and
           LV_ERR_ACR_DR             == 0                         and
           LV_ERR_MAP_TPS_PLAUS   == 0                            and
           [ (LV_IS == 1 and T_AST > C_T_AST_EGR) or LV_PL==1 ]   or
           LC_ACR_CTL_ENA== 1
then       LV_ACR_CTL_ENA = 1
else       LV_ACR_CTL_ENA = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_MIN_EGR	1	0...FEH	-48...142.5	0.75	[°C]
Minimum temperature for EGR-control active					
C_T_AST_EGR	1	0...FFFFH	0...6553.5	0.1	[s]
Time delay for EGR active after start					
LC_ACR_CTL_ENA	1	0...1H	0...1	1	[-]
Logical constant for the activation of actuator position control					

17.11 ACRC adaptation and diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_ACR_AD	O/V	0H 1H 2H 3H 4H 5H	ACR_AD_INIT ACR_AD_BOL ACR_GO_TOL ACR_AD_TOL ACR_GO_BOL ACR_AD_END	1	-
Variable indicate the state of the key on actuator adaptation					
LV_INH_ACR_CTL_AD	O/V	0...1H	0...1	1	-
Logical variable to inhibit actuator position control					
V_ACR_AD_BOL	O/V/S	0...3FFH	0...4.99511719	0.00488	V
Adaptation value for the lower mechanical stop of the actuator valve					
V_ACR_AD_TOL	O/V/S	0...3FFH	0...4.99511719	0.00488	V
Adaptation value for the upper mechanical stop					
FAC_ACR_SLOP	O/V/S	0...FFFFH	0...99.9984741	0.001526	%/V
Signal slop of the actuator measuring system					
LV_OPG_SP_AD_ACR	O/V	0...1H	0...1	1	-
Logical variable indicates a setpoint request for the actuator adaptation					
OPG_SP_AD_ACR	O/V	0...FFFH	0...99.9755859	0.02441	%
Actuator adaptation setpoint request for actuator position control					
LV_ERR_ACR_AD	O/V	0...1H	0...1	1	-
Logical variable is set, if an adaptation error occurs					
CTR_DIAG_ACR_AD	V	0...FFH	0...255	1	-
Diagnostic counter used at the continuously BOL adaptation					
LV_END_DIAG_ACR_AD	V	0...1H	0...1	1	-
Logical variable is set, if end of diagnosis condition for the actuator adaptation is fulfilled					
LV_CDN_DIAG_ACR_AD	V	0...1H	0...1	1	-
Logical variable is set, if the diagnosis condition for actuator adaptation is fulfilled					
ERR_SYM_ACR_AD	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the actuator adaptation					
V_ACR_AD_BOL_0	V/S	0...3FFH	0...4.99511719	0.00488	V
Adaptation value for the lower mechanical stop after first adaptation					
CTR_DLY_ACR_AD	V	0...FFH	0...255	1	-
Delay counter for the activation of the actuator adaptation					

Input data:

LV_ACR_AD_ACT	LV_ACR_AD_REQ	LV_INH_ACR_AD	LV_INH_ACR_AD_BOL
V_ACR			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_ACR_AD	1	0...FFH	0...0.99609375	0.003906	-
Correlation constant for the determination of the actuator adaptation value					
C_CTR_ACR_AD	1	0...FFH	0...255	1	-
Maximum time for the adaptation of the lower mechanical stop					
C_CTR_DIAG_ACR_AD	1	0...FFH	0...255	1	-
Diagnosis counter threshold for the adaptation of the lower mechanical stop					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_DIAG_END_ACR_AD	1	0...FFH	0...255	1	-
Constant to define the end of diagnosis of the continuously BOL adaptation					
C_CTR_DLY_ACR_AD	1	0...FFH	0...255	1	-
Delay time before adaptation of the lower and upper mechanical stop will be started					
C_CTR_DLY_ACR_AD_MAX	1	0...FFH	0...255	1	-
Delay counter for the activation of the actuator adaptation					
C_CTR_DLY_ACR_AD_TOL	1	0...FFH	0...255	1	-
End of diagnosis counter for the actuator adaptation					
C_FAC_ACR_SLOP	1	0...FFFFH	0...99.9984741	0.001526	%/V
Signal slop of the actuator measuring system					
C_OPG_SP_AD_LGRD	1	0...FFFH	0...99.9755859	0.02441	%
Change limitation for the adaptation of the upper mechanical stop					
C_OPG_SP_MAX_AD_ACR	1	0...FFFH	0...99.9755859	0.02441	%
Start setpoint for the adaptation of the upper mechanical stop					
C_V_ACR_AD_BOL_INI	1	0...3FFH	0...4.99511719	0.00488	V
Initialisation value for the adaptation value of the lower mechanical stop					
C_V_ACR_AD_BOL_MAX	1	0...3FFH	0...4.99511719	0.00488	V
Upper limit for the plausibility check of the adaptation value for the lower mechanical stop					
C_V_ACR_AD_BOL_MIN	1	0...3FFH	0...4.99511719	0.00488	V
Lower limit for the plausibility check of the adaptation value for the lower mechanical stop					
C_V_ACR_AD_CHK_CTL	1	0...3FFH	0...4.99511719	0.00488	V
Threshold for the actuator function check during the upper mechanical stop adaptation					
C_V_ACR_AD_TOL_INI	1	0...3FFH	0...4.99511719	0.00488	V
Initialisation value of the upper mechanical stop					
C_V_ACR_AD_TOL_MAX	1	0...3FFH	0...4.99511719	0.00488	V
Upper limit for the plausibility check of the adaptation value for the upper mechanical stop					
C_V_ACR_AD_TOL_MIN	1	0...3FFH	0...4.99511719	0.00488	V
Lower limit for the plausibility check of the adaptation value for the upper mechanical stop					
LC_FAC_ACR_SLOP_AD_ENA	1	0...1H	0...1	1	-
Logical constant to enable the adaptation of the upper mechanical stop					

Import actions:

ACTION_ERRM_NoFilterSymptom(IN<*XX*>, IN<lv_cdn_diag_XX>, IN<err_sym_XX>, IN<lv_err_set_XX>, IN<lv_err_reset_XX>, IN<lv_end_diag_XX>, OUT<lv_err_XX> **)**
This action computes the elementary treatment case no filtering is used

17.11.1 General information

The actuator valve position is determined by a measuring system. In order to reduce measurement inaccuracy, the sensor signal voltage is referenced to its sensor supply voltage. The electrical value of the lower and upper mechanical stop depends on the actuator characteristic, the actuator temperature and the valve coking therefore the electrical valve characteristic has to be learnt. After the actuator adaptation the adaptation values of the lower, the upper mechanical stop and the sensor slope are stored at the end of the driving cycle as "non-volatile".

Application Condition

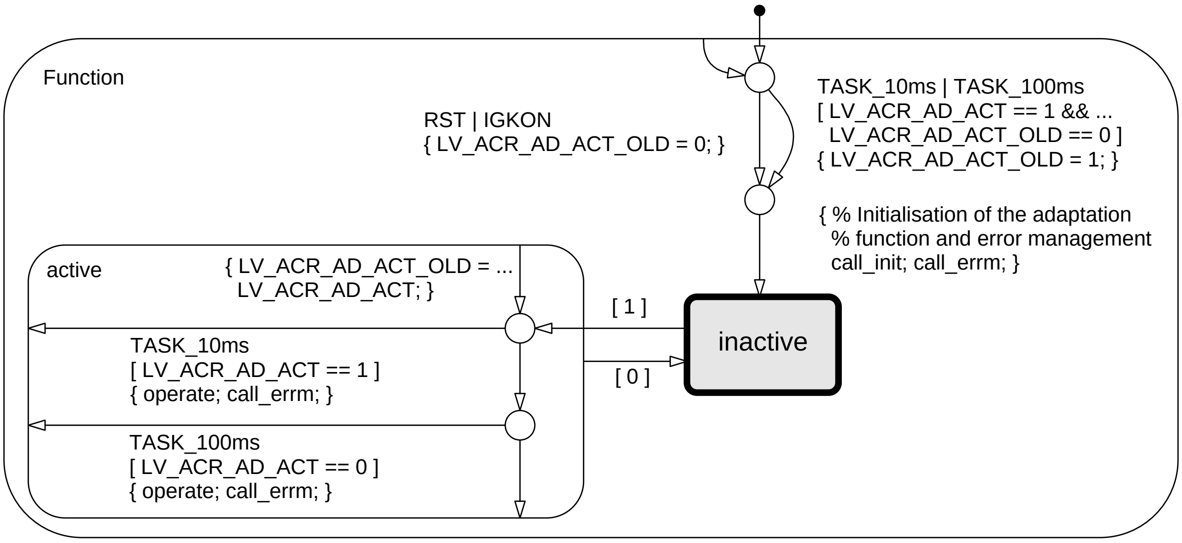
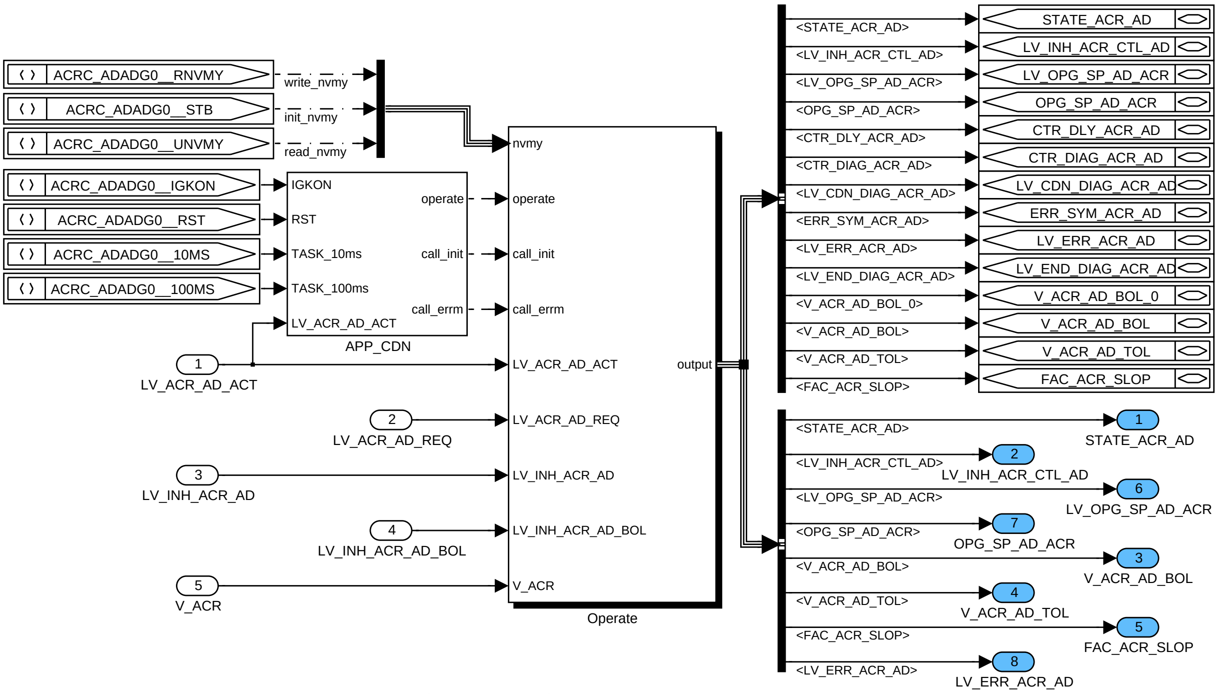


Figure 34 ACRC_adaptation_and_diagnosis/ APP_CDN/ APP_CDN

Function Description



SDA_SRS / SDA 4.0 19−Jul−2005

Figure 35 ACRC_adaptation_and_diagnosis

17.11.1.1 Adaptation functionalities

The adaptation functionality consists of a complete adaptation of the actuator valve and a continuously adaptation only for the lower mechanical valve stop. The adaptation algorithms are controlled by the variable LV_ACR_AD_ACT and the related inhibition conditions. The variables are provided in corresponding application incidence module.
After the reset of the functionality a complete adaptation of the actuator valve is started if the variable LV_ACR_AD_ACT has been set otherwise the continuously adaptation of the lower mechanical stop will be performed.

The adaptation module contain own diagnostic functions for the check of the adaptation values and the correct function of the actuator valve.

Error symptoms for the actuator adaptation are defined as follows :

SYM_0 – Adaptation conditions not fulfilled
SYM_1 – Lower adaptation value outside range
SYM_2 – Adaptation position not reached
SYM_3 – Upper adaptation value outside range

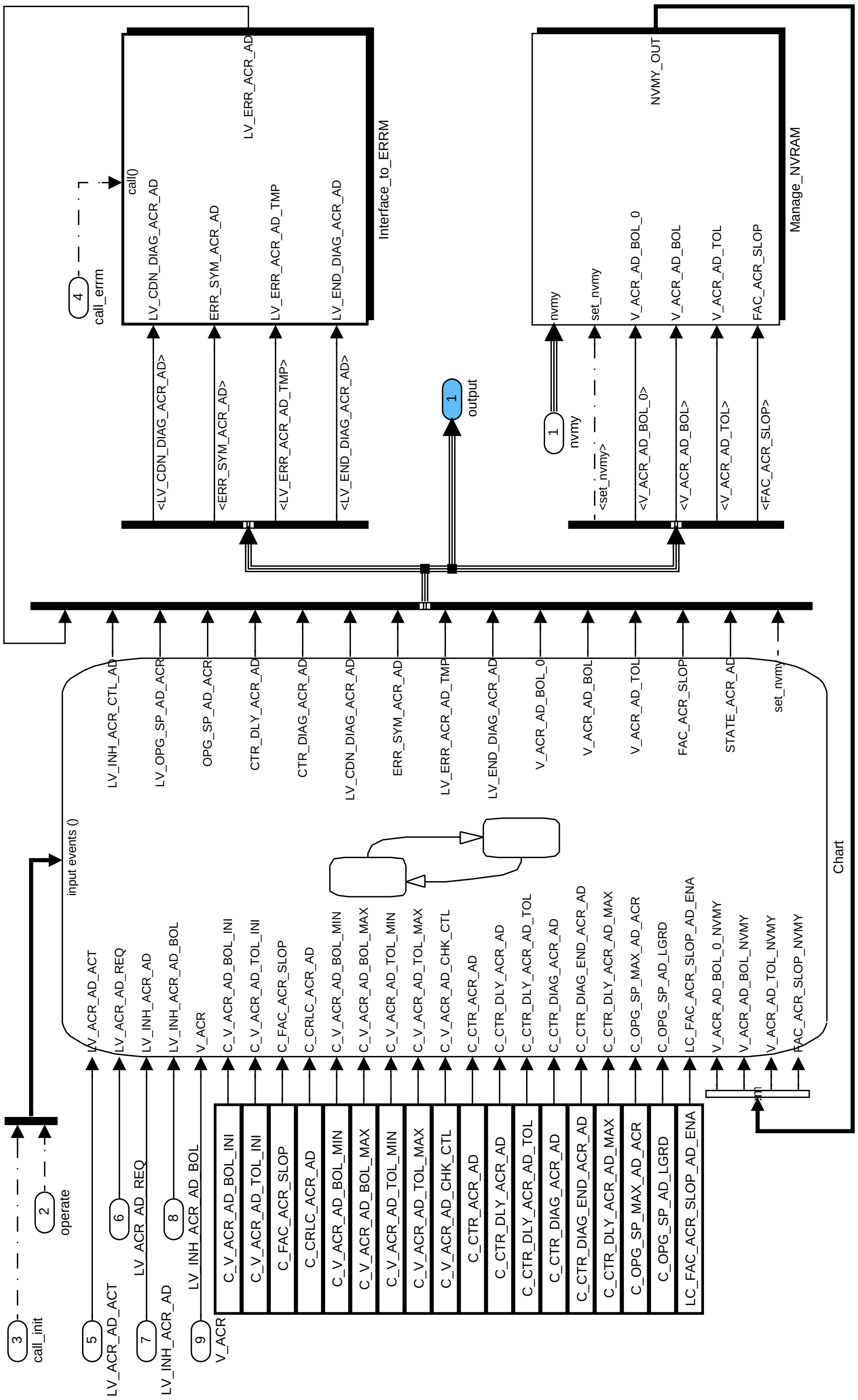


Figure 36 ACRC_adaptation_and_diagnosis/ Operate

State machine of the actuator adaptation

The state flow diagram of the actuator adaptation functionality contains the complete and the continuously adaptation functionality. The state machine is reset with a new reset event generated by the application conditions that means the functionality will be restarted on the default transition dependent on LV_ACR_AD_ACT.

The inhibition variable LV_INH_ACR_AD of the adaptation function is only check after the initialisation state has been finished. If the inhibition variable is set during the check then an error entry will be performed.

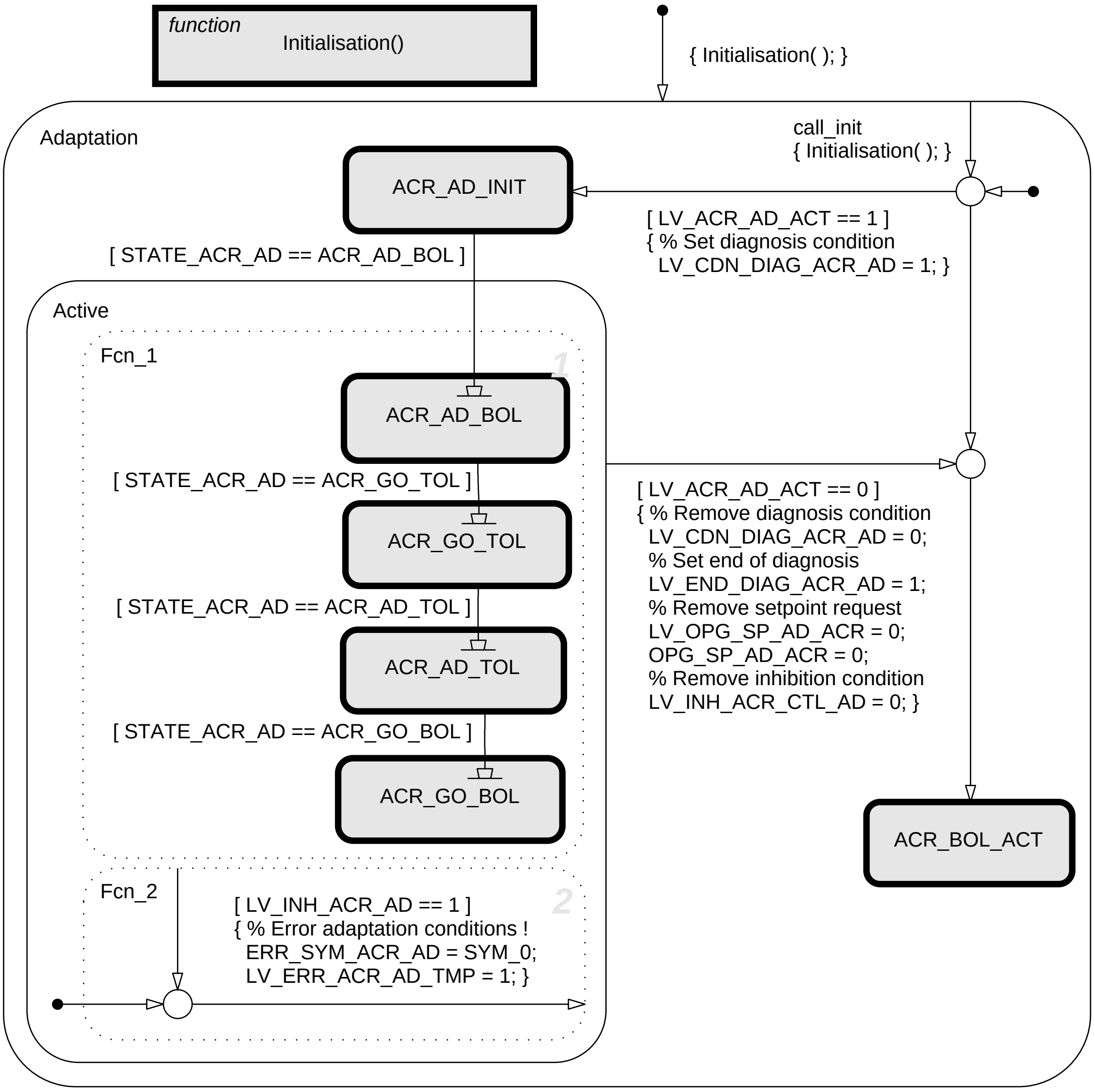
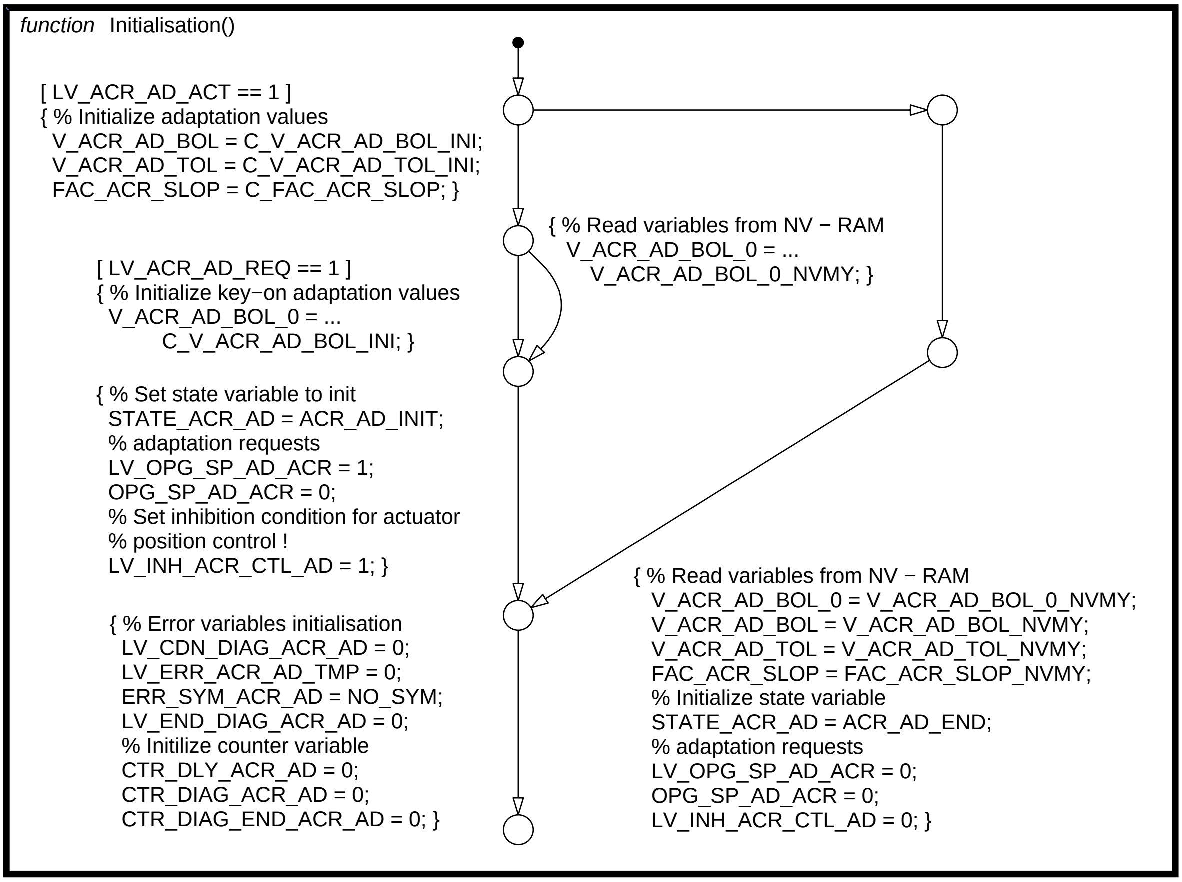


Figure 37 ACRC_adaptation_and_diagnosis/ Operate/ Chart

Initialisation function

The following function is used for the initialisation of the actuator adaptation. The function is called from the application condition module and shall initialise the adaptation values, the used error and auxiliary values.

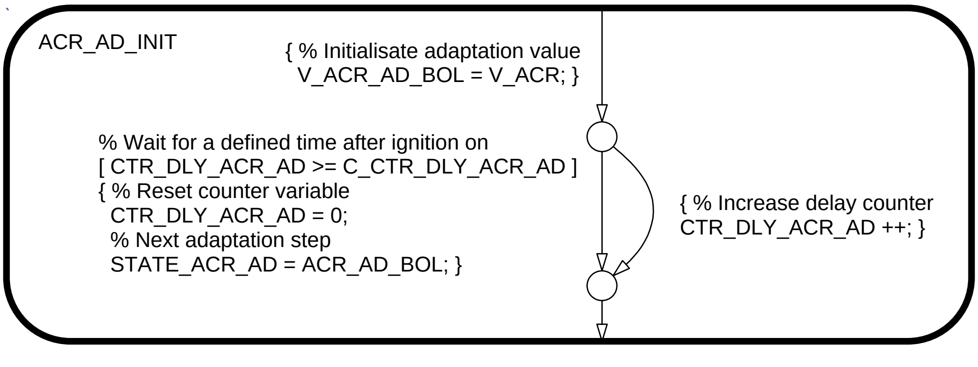


```
function  Initialisation()

[ LV_ACR_AD_ACT == 1 ]
{ % Initialize adaptation values
  V_ACR_AD_BOL = C_V_ACR_AD_BOL_INI;
  V_ACR_AD_TOL = C_V_ACR_AD_TOL_INI;
  FAC_ACR_SLOP = C_FAC_ACR_SLOP; }

{ % Read variables from NV − RAM
  V_ACR_AD_BOL_0 = ...
    V_ACR_AD_BOL_0_NVMY; }

[ LV_ACR_AD_REQ == 1 ]
{ % Initialize key−on adaptation values
  V_ACR_AD_BOL_0 = ...
       C_V_ACR_AD_BOL_INI; }

{ % Set state variable to init
  STATE_ACR_AD = ACR_AD_INIT;
  % adaptation requests
  LV_OPG_SP_AD_ACR = 1;
  OPG_SP_AD_ACR = 0;
  % Set inhibition condition for actuator
  % position control !
  LV_INH_ACR_CTL_AD = 1; }

{ % Read variables from NV − RAM
  V_ACR_AD_BOL_0 = V_ACR_AD_BOL_0_NVMY;
  V_ACR_AD_BOL = V_ACR_AD_BOL_NVMY;
  V_ACR_AD_TOL = V_ACR_AD_TOL_NVMY;
  FAC_ACR_SLOP = FAC_ACR_SLOP_NVMY;
  % Initialize state variable
  STATE_ACR_AD = ACR_AD_END;
  % adaptation requests
  LV_OPG_SP_AD_ACR = 0;
  OPG_SP_AD_ACR = 0;
  LV_INH_ACR_CTL_AD = 0; }

{ % Error variables initialisation
  LV_CDN_DIAG_ACR_AD = 0;
  LV_ERR_ACR_AD_TMP = 0;
  ERR_SYM_ACR_AD = NO_SYM;
  LV_END_DIAG_ACR_AD = 0;
  % Initilize counter variable
  CTR_DLY_ACR_AD = 0;
  CTR_DIAG_ACR_AD = 0;
  CTR_DIAG_END_ACR_AD = 0; }
```

Figure 38 ACRC_adaptation_and_diagnosis/ Operate/ Chart/ Initialisation

Initialisation state of the actuator adaptation

At the start of the actuator adaptation the following initialisation function is necessary. During the initialisation state the adaptation waits a defined time for valid adaptation conditions.



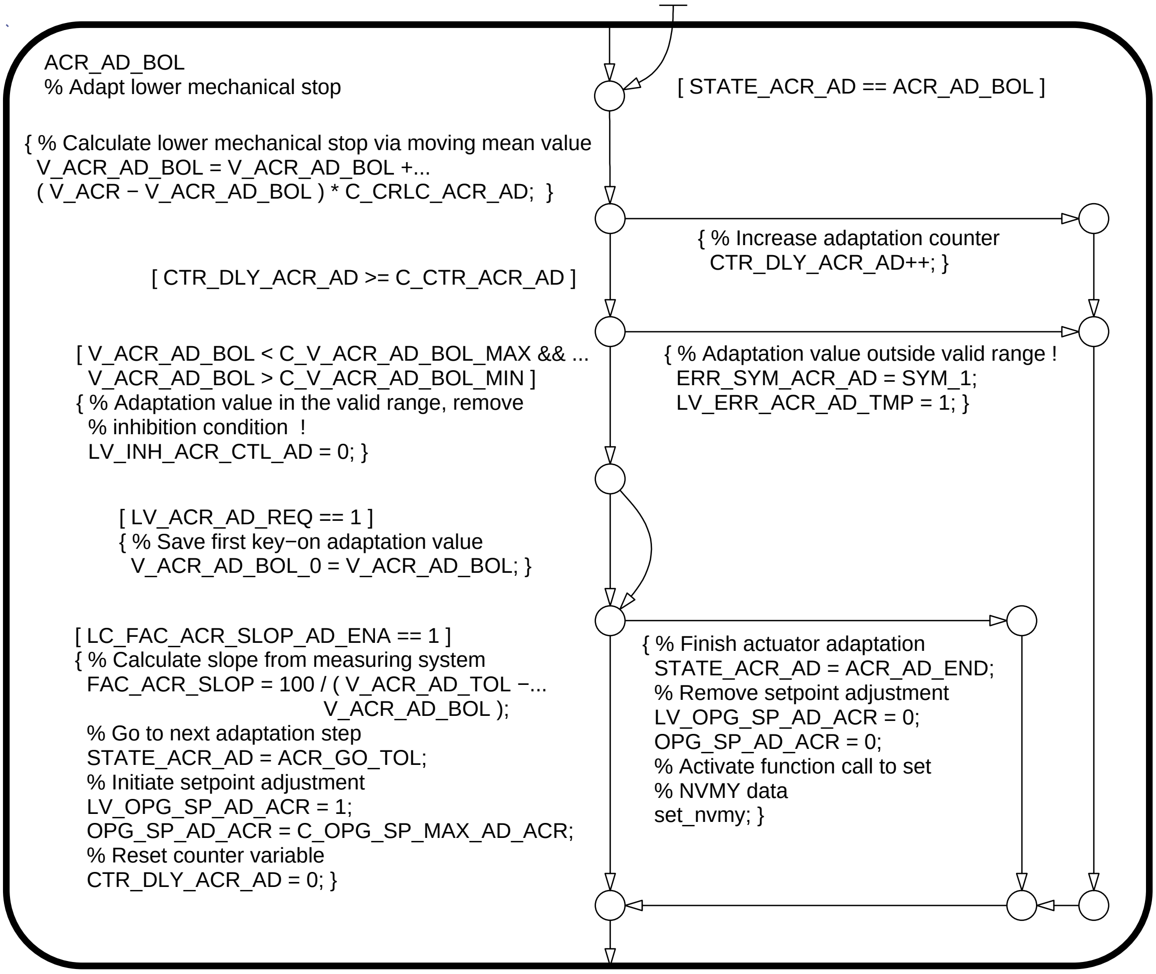
```
ACR_AD_INIT

{ % Initialisate adaptation value
  V_ACR_AD_BOL = V_ACR; }

% Wait for a defined time after ignition on
[ CTR_DLY_ACR_AD >= C_CTR_DLY_ACR_AD ]
{ % Reset counter variable
  CTR_DLY_ACR_AD = 0;
  % Next adaptation step
  STATE_ACR_AD = ACR_AD_BOL; }

{ % Increase delay counter
CTR_DLY_ACR_AD ++; }
```

Figure 39 ACRC_adaptation_and_diagnosis/ Operate/ Chart/ Adaptation/ ACR_AD_INIT

Adaptation of the lower mechanical stop

In the following state the lower mechanical stop of the actuator valve is learnt. The adaptation value is determined as moving mean value of the sensor signal voltage. The corresponding correlation constant C_CRLC_ARC_AD defines the calculation of the adaptation value. The adaptation is performed only for a defined adaptation time to get a valid adaptation value. At the end of the algorithm the adaptation value is check of validity. If the adaptation value is outside of the valid range than an error entry will be performed.

Additionally the further adaptation functionality, the adaptation of the upper mechanical stop, can be chosen via configuration data LC_FAC_ACR_SLOP_AD_ENA. If the constant is set then the upper mechanical stop and the sensor slope will be learnt, otherwise the adaptation will be finished and the initialisation values will be used !



```
ACR_AD_BOL
% Adapt lower mechanical stop

[ STATE_ACR_AD == ACR_AD_BOL ]

{ % Calculate lower mechanical stop via moving mean value
  V_ACR_AD_BOL = V_ACR_AD_BOL +...
  ( V_ACR − V_ACR_AD_BOL ) * C_CRLC_ACR_AD;  }

{ % Increase adaptation counter
  CTR_DLY_ACR_AD++; }

[ CTR_DLY_ACR_AD >= C_CTR_ACR_AD ]

[ V_ACR_AD_BOL < C_V_ACR_AD_BOL_MAX && ...
  V_ACR_AD_BOL > C_V_ACR_AD_BOL_MIN ]
{ % Adaptation value in the valid range, remove
  % inhibition condition  !
  LV_INH_ACR_CTL_AD = 0; }

{ % Adaptation value outside valid range !
  ERR_SYM_ACR_AD = SYM_1;
  LV_ERR_ACR_AD_TMP = 1; }

[ LV_ACR_AD_REQ == 1 ]
{ % Save first key−on adaptation value
  V_ACR_AD_BOL_0 = V_ACR_AD_BOL; }

[ LC_FAC_ACR_SLOP_AD_ENA == 1 ]
{ % Calculate slope from measuring system
  FAC_ACR_SLOP = 100 / ( V_ACR_AD_TOL −...
                         V_ACR_AD_BOL );
  % Go to next adaptation step
  STATE_ACR_AD = ACR_GO_TOL;
  % Initiate setpoint adjustment
  LV_OPG_SP_AD_ACR = 1;
  OPG_SP_AD_ACR = C_OPG_SP_MAX_AD_ACR;
  % Reset counter variable
  CTR_DLY_ACR_AD = 0; }

{ % Finish actuator adaptation
  STATE_ACR_AD = ACR_AD_END;
  % Remove setpoint adjustment
  LV_OPG_SP_AD_ACR = 0;
  OPG_SP_AD_ACR = 0;
  % Activate function call to set
  % NVMY data
  set_nvmy; }
```

Figure 40 ACRC_adaptation_and_diagnosis/ Operate/ Chart/ Adaptation/ Active/ ACR_AD_BOL

Drive actuator in the upper mechanical stop

The following state drives the actuator valve in the upper mechanical stop via setpoint adjustment. At first the setpoint for closed-loop position control is set to a defined start value and the power stage is switched on. Afterwards the setpoint is increased via change limitation to 100 %.

Additionally the state checks the correct function of the closed-loop position control system. If the setpoint is set to 100 % then the actuator valve must exceed a diagnostic threshold otherwise an adaptation error is indicated and the adaptation is stopped.

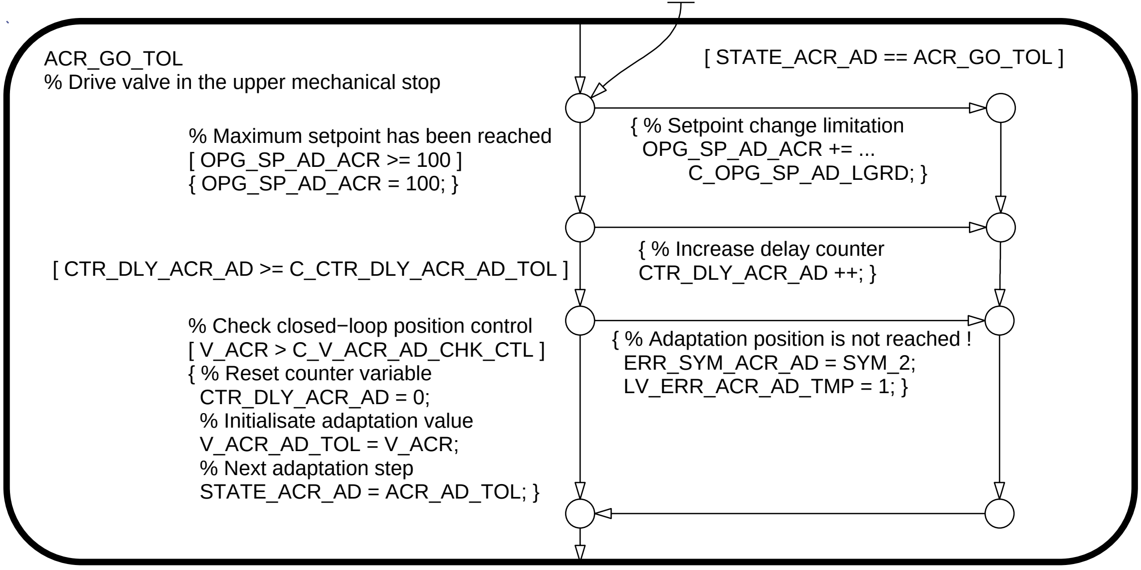


Figure 41 ACRC_adaptation_and_diagnosis/ Operate/ Chart/ Adaptation/ Active/ ACR_GO_TOL

Adaptation of the upper mechanical stop

In the next adaptation state the upper mechanical stop of the actuator valve is learnt. The adaptation value of the upper mechanical stop is determined as moving mean value of the sensor signal voltage. The corresponding correlation constant C_CRLC_ARC_AD defines the calculation of the adaptation value. The adaptation is performed only for a defined adaptation time to get a valid adaptation value. At the end of the algorithm the adaptation value of the upper mechanical stop is check of validity. If the adaptation value is outside of the valid range than an error entry will be performed.

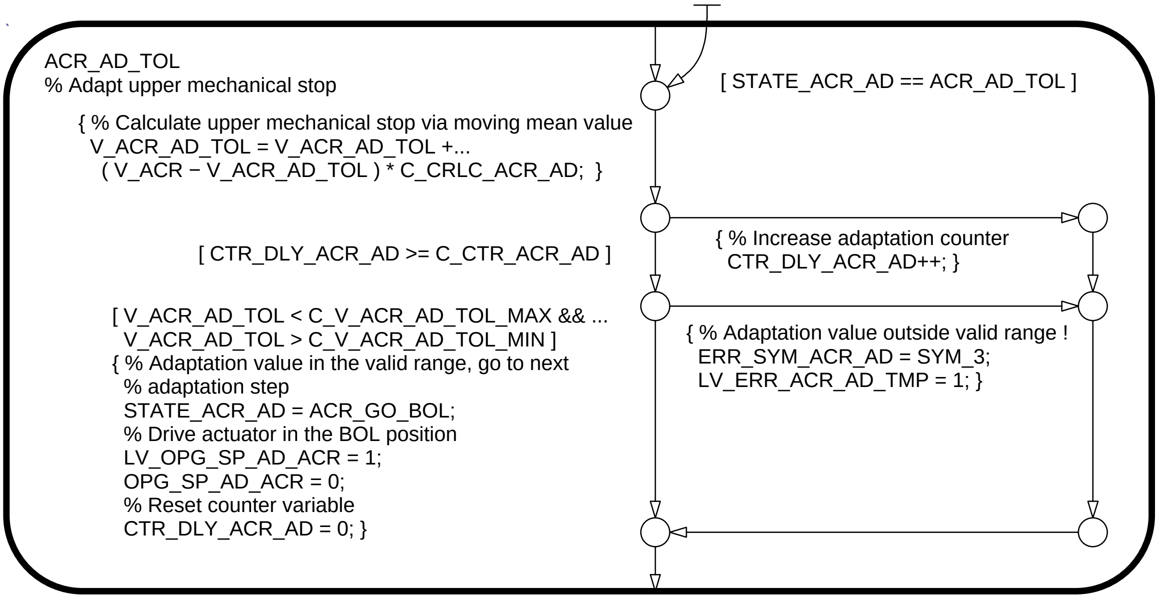
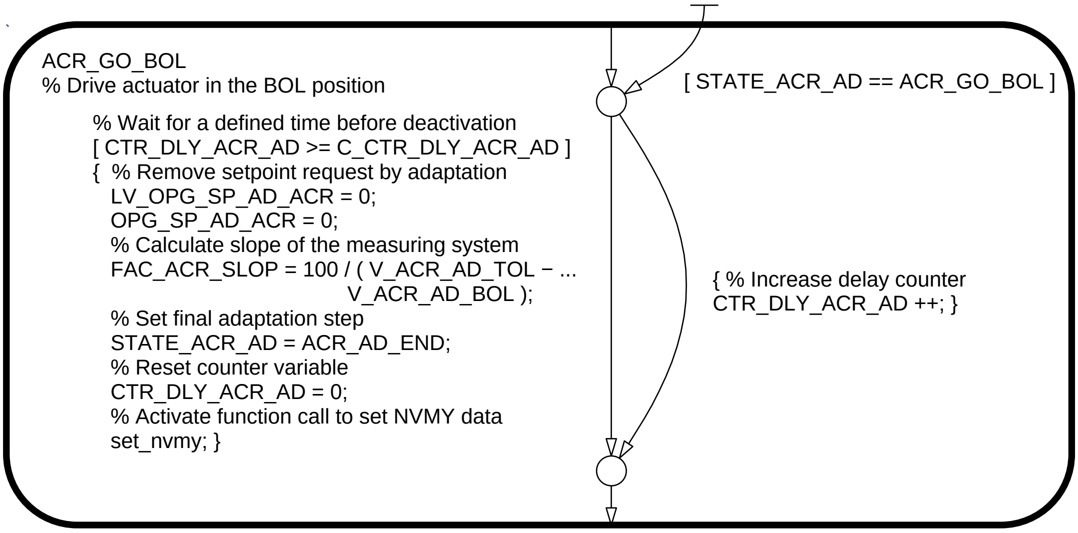


Figure 42 ACRC_adaptation_and_diagnosis/ Operate/ Chart/ Adaptation/ Active/ ACR_AD_TOL

Finish actuator adaptation

The last state of the adaptation shall drive the actuator valve in the lower mechanical stop and calculate the sensor signal slop of the measuring system. The adaptation state will be left if a defined waiting time is elapsed.



```
ACR_GO_BOL
% Drive actuator in the BOL position
```

```
[ STATE_ACR_AD == ACR_GO_BOL ]
```

```
% Wait for a defined time before deactivation
[ CTR_DLY_ACR_AD >= C_CTR_DLY_ACR_AD ]
{  % Remove setpoint request by adaptation
   LV_OPG_SP_AD_ACR = 0;
   OPG_SP_AD_ACR = 0;
   % Calculate slope of the measuring system
   FAC_ACR_SLOP = 100 / ( V_ACR_AD_TOL − ...
                              V_ACR_AD_BOL );
   % Set final adaptation step
   STATE_ACR_AD = ACR_AD_END;
   % Reset counter variable
   CTR_DLY_ACR_AD = 0;
   % Activate function call to set NVMY data
   set_nvmy; }
```

```
{ % Increase delay counter
CTR_DLY_ACR_AD ++; }
```

Figure 43 ACRC_adaptation_and_diagnosis/ Operate/ Chart/ Adaptation/ Active/ ACR_GO_BOL

Continuously adaptation of the lower mechanical stop

If the actuator valve is in closing position, then the adaptation of the lower mechanical stop will be started after a defined delay time. A closed actuator valve is indicated by the state variable STATE_ACR_CTL of the position control software. The adaptation value is determined as moving mean value of the sensor signal voltage. The corresponding correlation constant C_CRLC_ARC_AD defines the calculation of the adaptation value.

At the end of the algorithm the adaptation value is check of validity. If the adaptation value is outside of the valid range than an error entry will be performed.

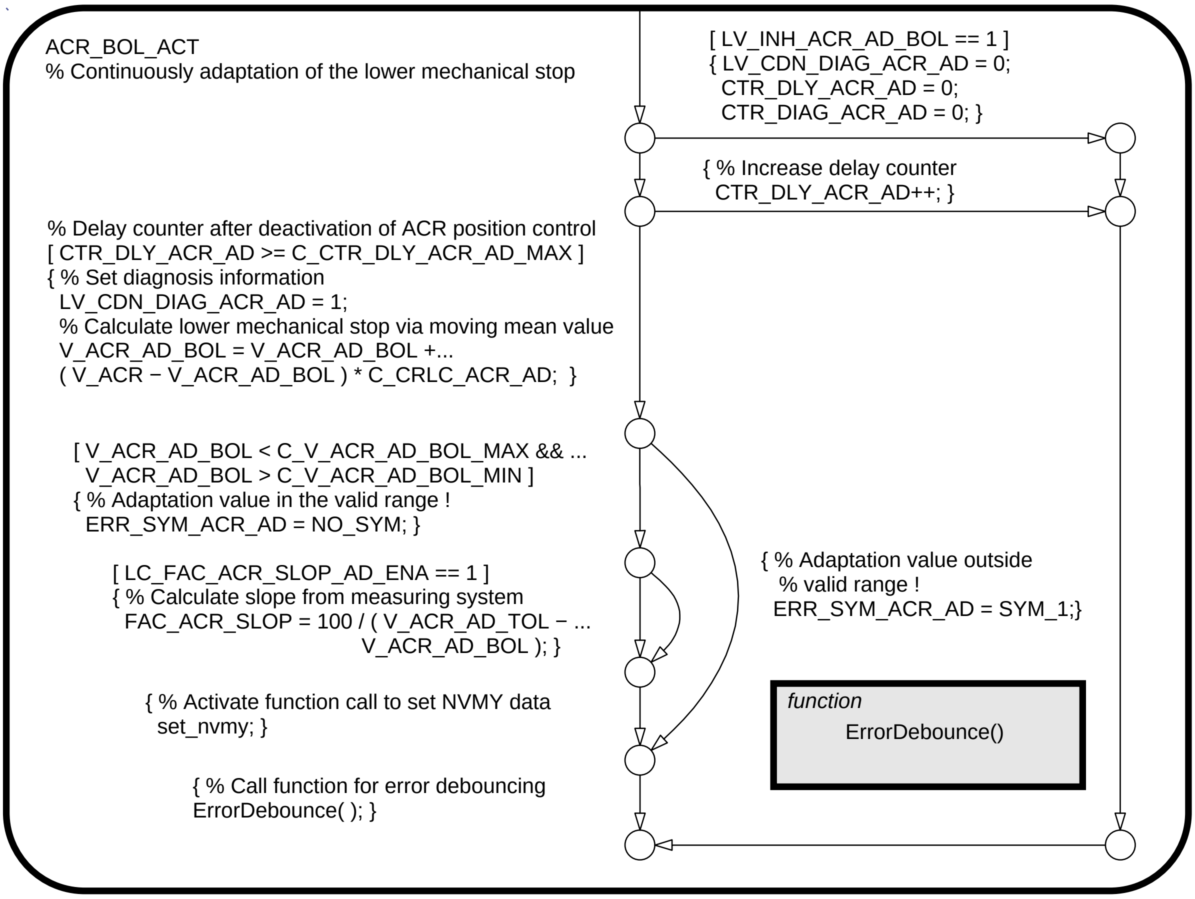


Figure 44 ACRC_adaptation_and_diagnosis/ Operate/ Chart/ Adaptation/ ACR_BOL_ACT

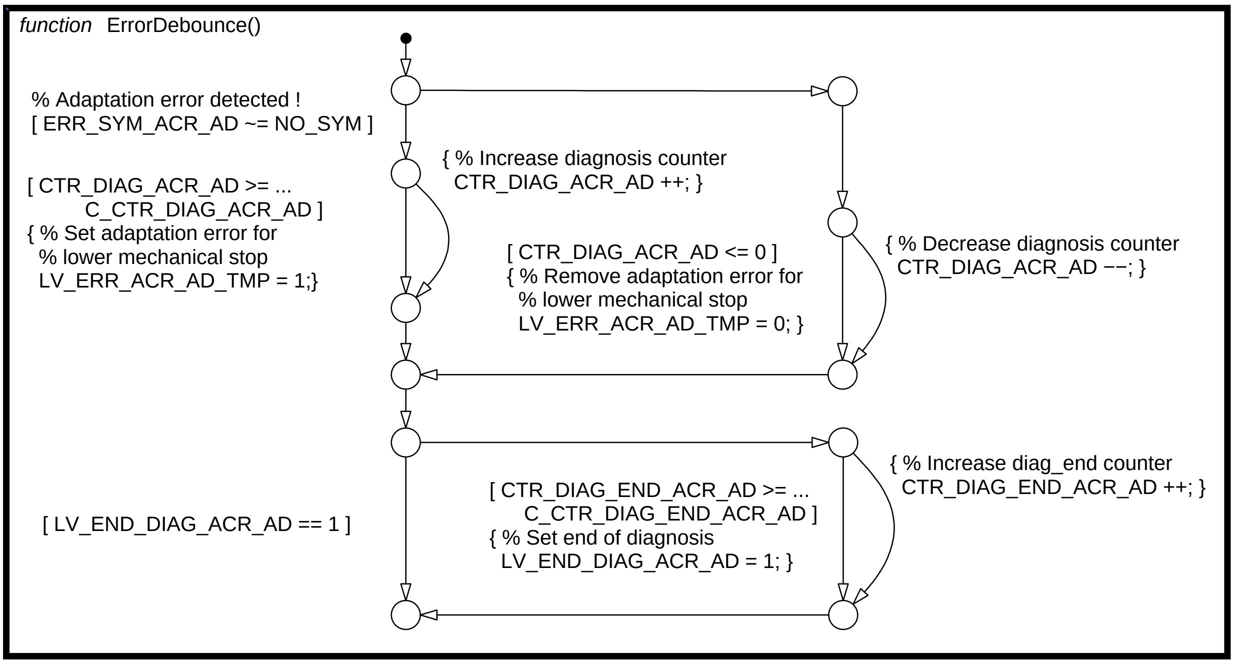


Figure 45 ACRC_adaptation_and_diagnosis/ Operate/ Chart/ Adaptation/ ACR_BOL_ACT/ ErrorDebounce

Interface to the error management

The following algorithm set the detected error and the end of diagnosis information according the diagnostic result and delivers the result to Error Management.

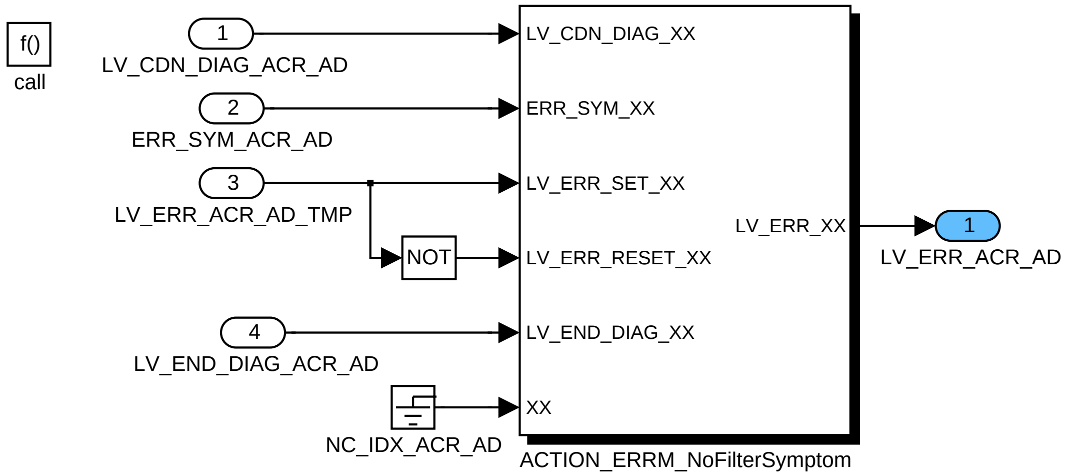


Figure 46 ACRC_adaptation_and_diagnosis/ Operate/ Interface_to_ERRM

Manage non-volatile variables

The last section describes the data handling of variables defined in the non-volatile memory. Three function calls are available for the initialisation, the readout and the update of the non-volatile memory. A special feature has been implemented for the data update. New adaptation values are only stored for an error free system and a successful adaptation end. Otherwise the last valid adaptation values will be used again for the NVRAM update!

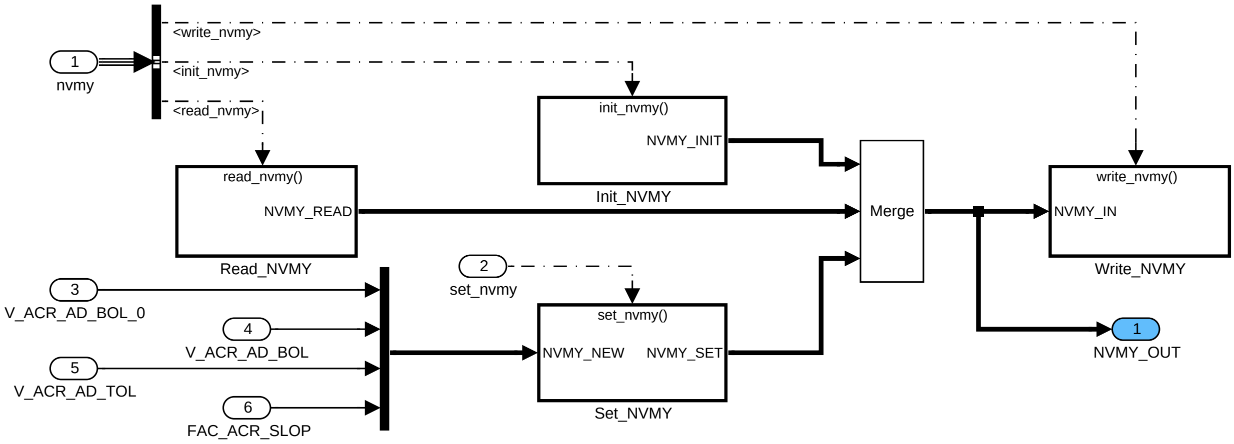


Figure 47 ACRC_adaptation_and_diagnosis/ Operate/ Manage_NVRAM

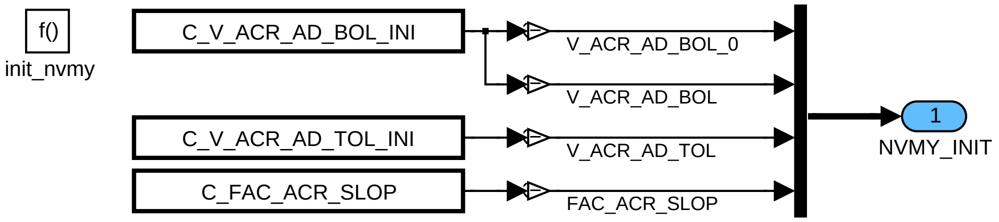


Figure 48 ACRC_adaptation_and_diagnosis/ Operate/ Manage_NVRAM/ Init_NVMY

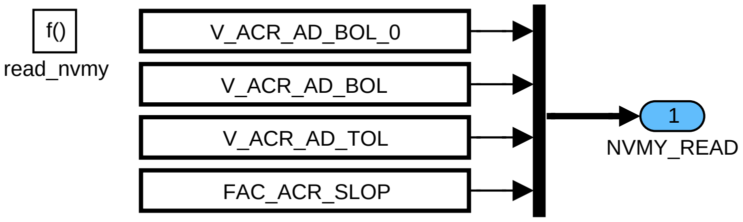


Figure 49 ACRC_adaptation_and_diagnosis/ Operate/ Manage_NVRAM/ Read_NVMY

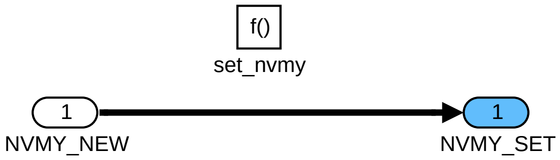


Figure 50 ACRC_adaptation_and_diagnosis/ Operate/ Manage_NVRAM/ Set_NVMY

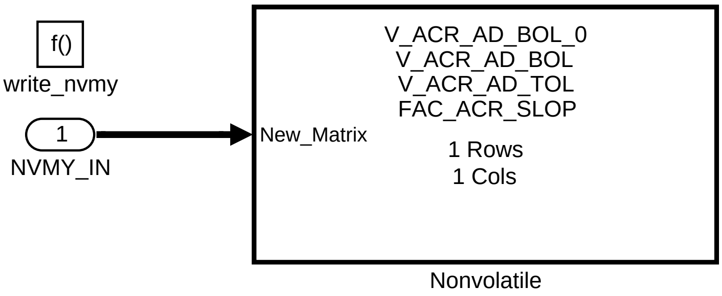


Figure 51 ACRC_adaptation_and_diagnosis/ Operate/ Manage_NVRAM/ Write_NVMY

17.12 ACRC actuator adaptation and diagnosis (appl. inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ACR_AD_ACT	O/V	0...1H	0...1	1	-
Logical variable to indicate an active actuator adaptation					
LV_ACR_AD_REQ	O/V/S	0...1H	0...1	1	-
Logical variable to indicate an actuator adaptation request					
LV_INH_ACR_AD	O/V	0...1H	0...1	1	-
Inhibition condition for the actuator adaptation and diagnosis					
LV_INH_ACR_AD_BOL	O/V	0...1H	0...1	1	-
Inhibition condition for the countinuous actuator adaptation and diagnosis					
LV_ACR_AD_POP_INH	V	0...1H	0...1	1	-
Logical variable to inhibit the actuator adaptation for post operating phase during IGKON					

Input data:

STATE_ACR_CTL	LV_IGK	LV_ES	TCO
VB	LV_PWM_ACR_OFF_REQ	STATE_ACR_AD	LV_ERR_ACR_AD

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ACR_AD_TCO_MIN	1	0...FEH	-48...142.5	0.75	°C
Temperature threshold for the activation of the actuator adaptation					
C_ACR_AD_VB_MIN	1	0...FFH	0...25.8984375	0.1015625	V
Battery voltage threshold for the activation of the actuator adaptation					
LC_ACR_AD_POP_ENA	1	0...1H	0...1	1	-
Logical variable to enable the actuator adaptation in the post operating phase					
LC_ACR_AD_REQ	1	0...1H	0...1	1	-
Logical variable to set a new adaptation request					

Configuration for diagnostic symptoms:

Diagnostic ACR_AD	Symptom description	Symptom	Filter type
ACR actuator adaptation and diagnosis	*Adaptation conditions not fulfilled*	*SYM_0*	*NO*
	BOL adaptation value outside range	*SYM_1*	
	Could not reached TOL position	*SYM_2*	
	TOL adaptation value outside range	*SYM_3*	

17.12.1 General information

This module describes the coordination of the actuator adaptation with corresponding diagnosis functions. The actuator adaptation can be split in a standard adaptation during IGKON and post operation phase and a continuously adaptation only for the adaptation of the lower mechanical stop. Separate inhibition conditions for the described adaptations are available.

Application Condition

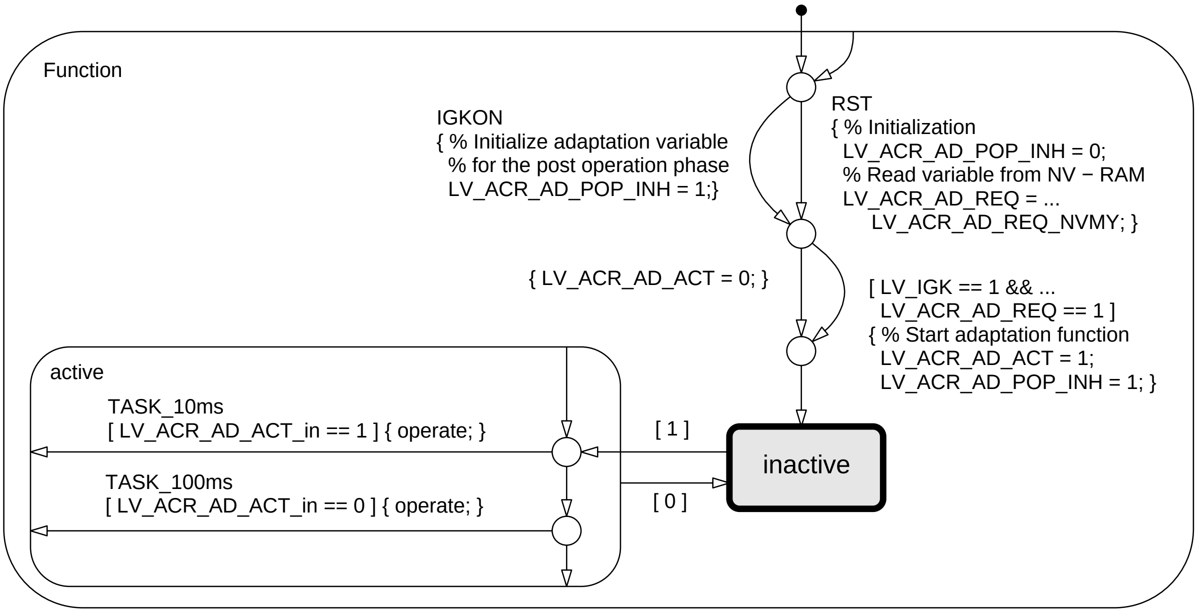
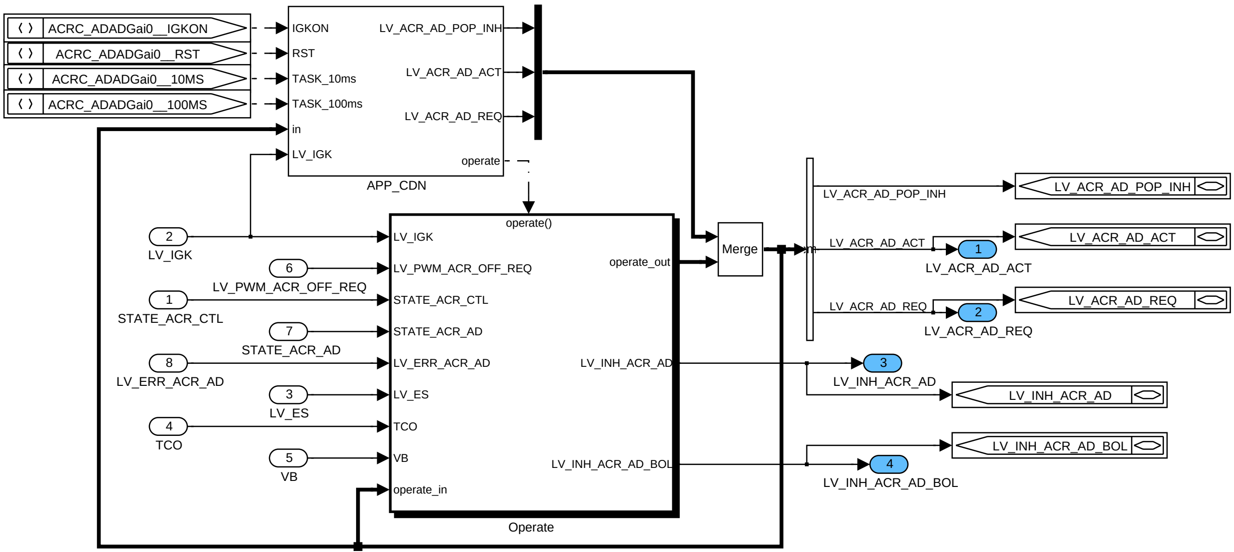


Figure 52 ACRC_adaptation_and_diagnosis_ai/ APP_CDN/ APP_CDN

Function Description



SDA_SRS / SDA 4.0 02−Mar−2005

Figure 53 ACRC_adaptation_and_diagnosis_ai

17.12.1.1 Coordination of the adaptation functionality

The adaptation request LV_ACR_AD_REQ is checked every time the control unit is switched on. If an adaptation is requested then a "first key on" adaptation will be performed with LV_ACR_AD_ACT = 1. The actuator adaptation is necessary in the following cases:

- In initial start (first engine run or control unit exchange), that means no adaptation values are stored in the non volatile memory
- Check sum error (NVMY – error) is current
- Adaptation request bit LV_ACR_AD_REQ is set

Each of these points ends with a new adaptation request. After the activation of the "key on" actuator adaptation the corresponding inhibition condition will be checked continuously.

A second part of the adaptation functionality is the actuator adaptation in the post operating phase. After ignition off and engine stop the corresponding inhibition variable is once checked. If all activation conditions are fulfilled then the actuator adaptation will be started, the inhibition variable is **not** checked during the adaptation to avoid possible error entries.

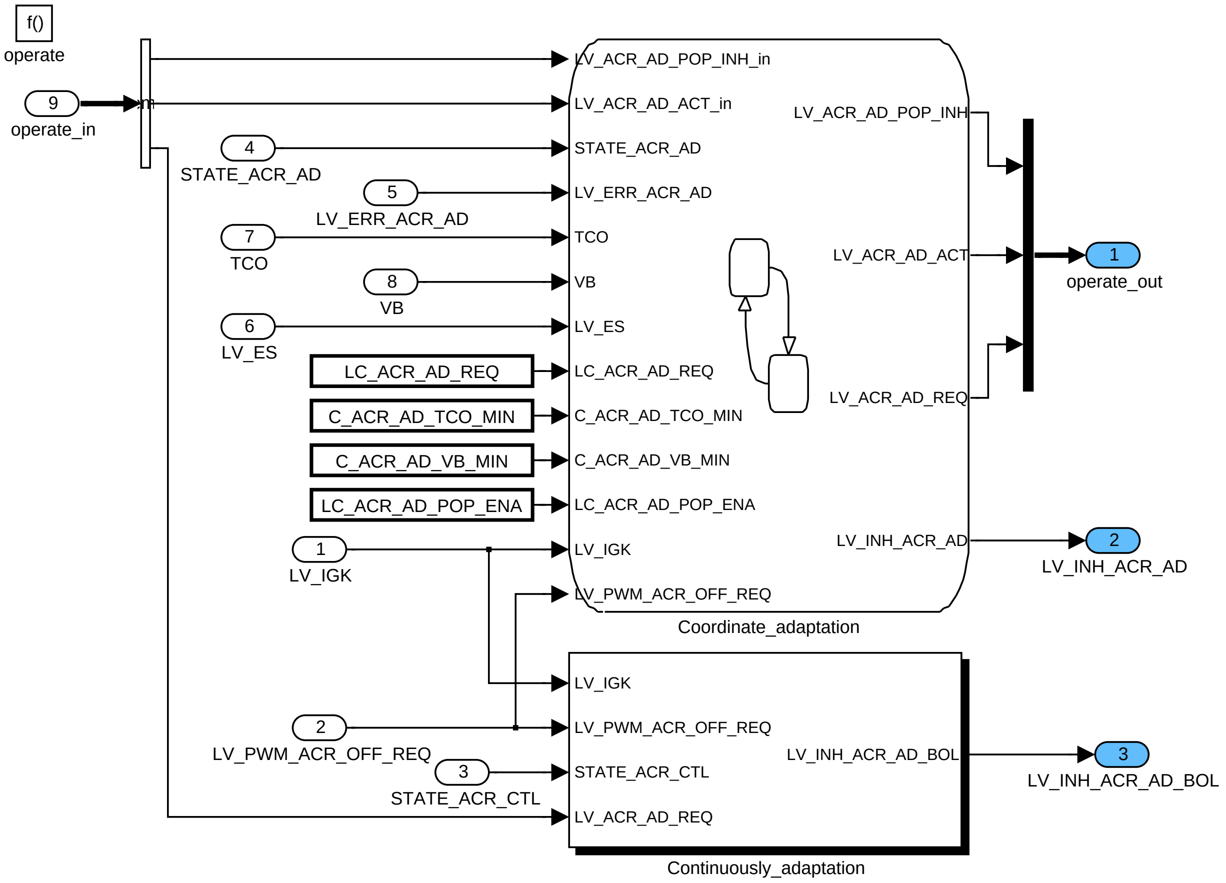


Figure 54 ACRC_adaptation_and_diagnosis_ai/ Operate

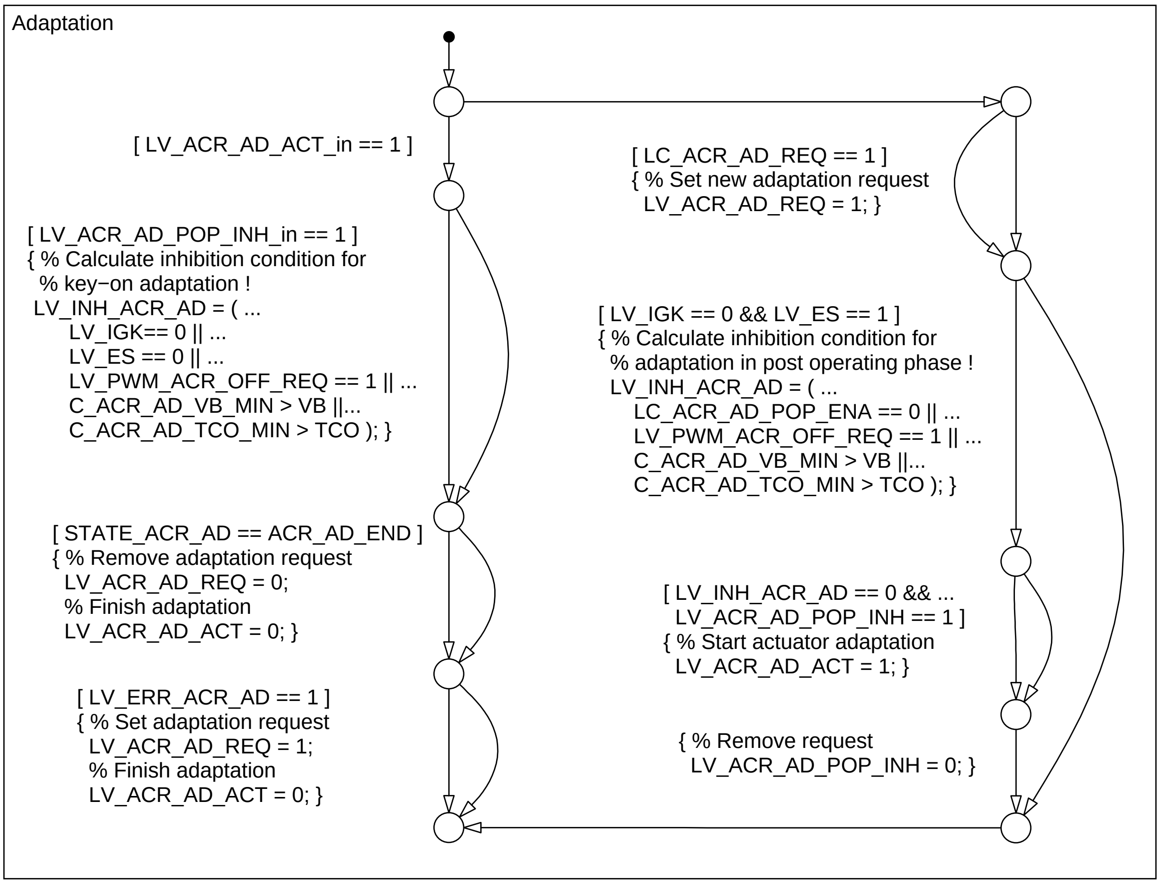


Figure 55 ACRC_adaptation_and_diagnosis_ai/ Operate/ Coordinate_adaptation

Inhibition condition for the continuous BOL adaptation

This chapter describes the inhibition condition of the continuously adaptation for the lower mechanical stop.

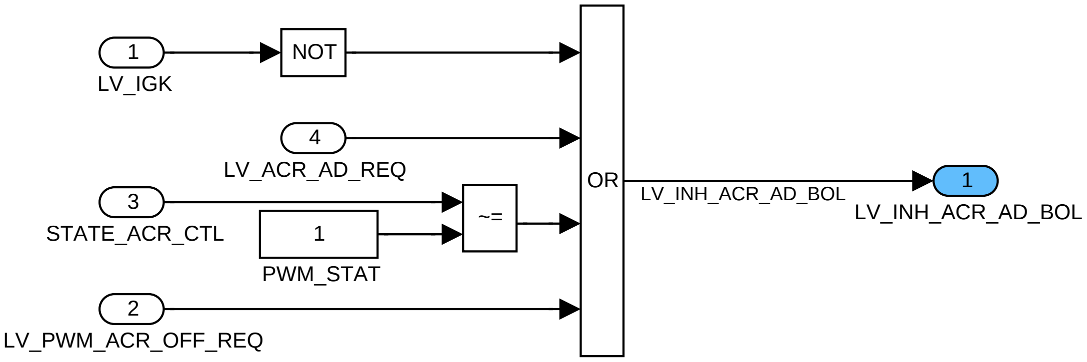


Figure 56 ACRC_adaptation_and_diagnosis_ai/ Operate/ Continuously_adaptation

17.13 ACRC position sensor diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SENS_ACR	O/V	0...1H	0...1	1	-
Present failure for sensor signal range check after error debounce					
ERR_SYM_SENS_ACR	O/V	0H 1H 2H	NO_SYM SYM_0 SYM_1	1	-
Error symptom for sensor signal range check					
LV_CDN_DIAG_SENS_ACR	V	0...1H	0...1	1	-
Diagnosis condition for sensor signal range check					
LV_END_DIAG_SENS_ACR	V	0...1H	0...1	1	-
End of diagnosis information for sensor signal range check					

Input data:

LV_INH_DIAG_SENS_ACR	V_ACR		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SENS_ACR	1	0...FFH	0...255	1	-
Anti bounce counter increment for signal range check					
C_ABC_MAX_SENS_ACR	1	1...FFH	1...255	1	-
Maximum value for anti bounce counter for signal range check					
C_V_MAX_SENS_ACR	1	0...3FFH	0...4.995117	0.00488	V
Upper diagnostic threshold for the signal range check					
C_V_MIN_SENS_ACR	1	0...3FFH	0...4.995117	0.00488	V
Lower diagnostic threshold for the signal range check					

Import actions:

ACTION_ERRM_FilterSymptom(IN<*XX*>, IN<lv_cdn_diag_XX>, IN< err_sym_XX>, IN<c_abc_inc_XX>, IN<c_abc_dec_XX>, IN<c_abc_max_XX>, OUT<lv_err_XX> **)**
This action computes the elementary anti-bounce filter for one failure treatment and returns filter result

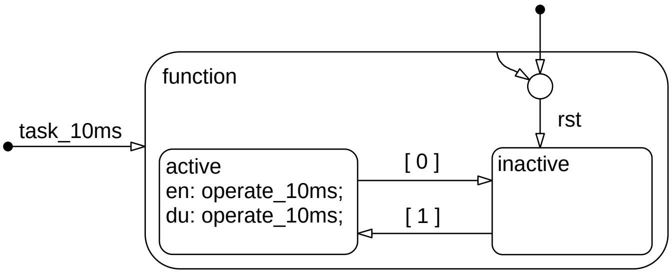
17.13.1 General information

This diagnosis function is used to monitor the admissible input voltage range of the actuator valve position sensor system. The diagnostic is able to detect short circuit to battery, open circuit and short circuit to ground for the position sensor input according to the pull resistor at the input of the AD – converter.

Error symptoms are defined to this function as following:

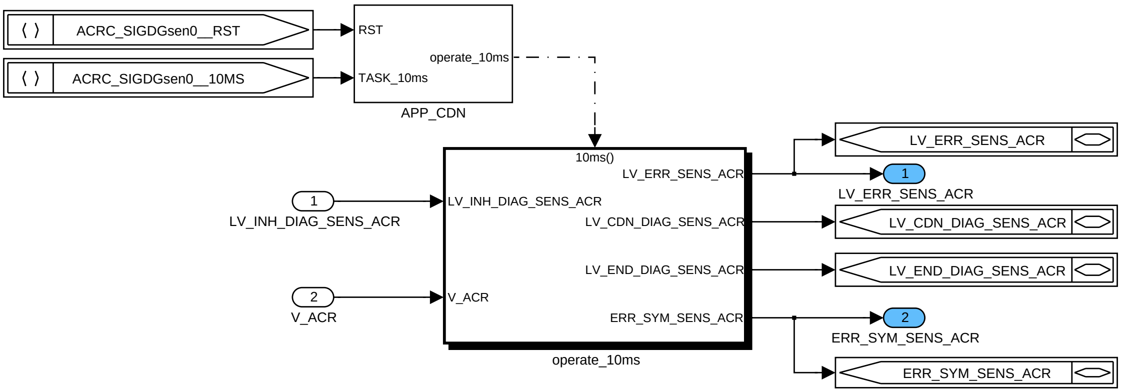
- SYM_0 - Short circuit to battery or open load
- SYM_1 - Short circuit to ground

Application Condition



ACRC_position_sensor_diagnosis/APP_CDN/APP_CDN

Function Description

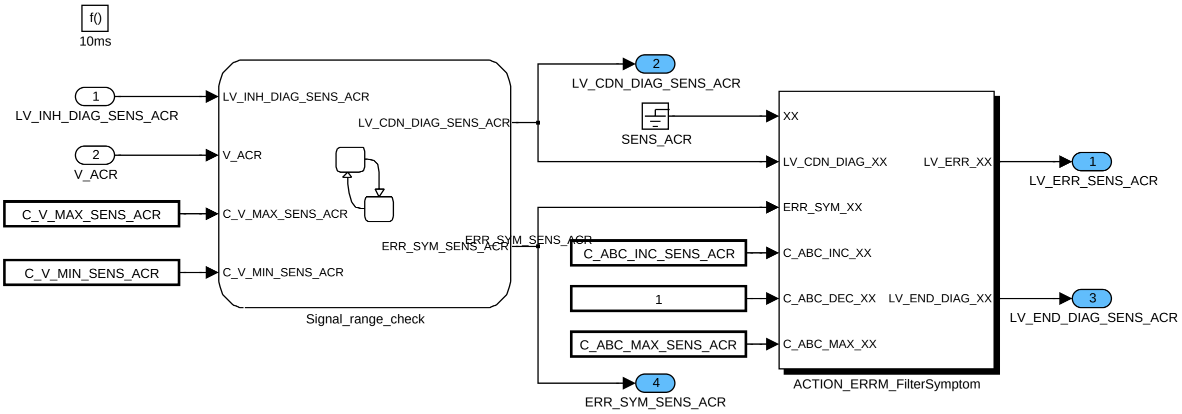


SDA_SRS / SDA 3.1 11−Mar−2004

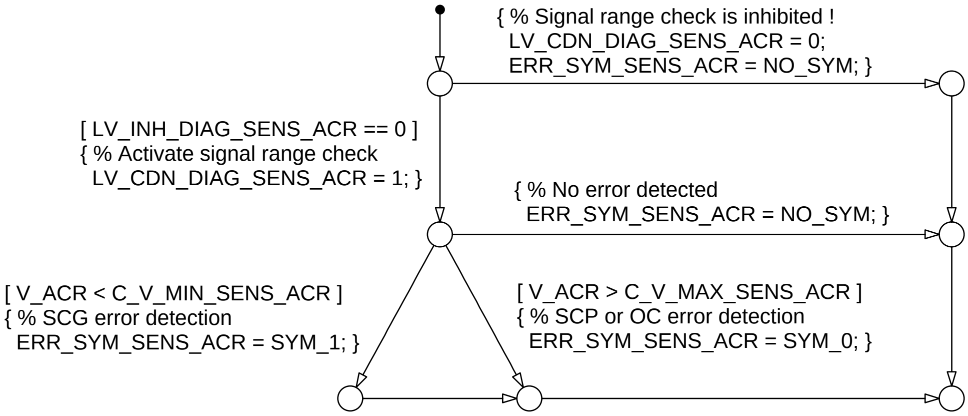
ACRC_position_sensor_diagnosis

17.13.1.1 Sensor signal range check

The anti-bounce filter algorithm from the generic error management is called at the end of the module.



ACRC_position_sensor_diagnosis/operate_10ms



ACRC_position_sensor_diagnosis/operate_10ms/Signal_range_check

17.14 ACRC position sensor diagnosis (appl. inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_SENS_ACR	O/V	0...1H	0...1	1	-
Inhibition condition for the position sensor range check					
LV_ERR_VCC_ACR	O/V	0...1H	0...1	1	-
Actuator sensor supply voltage error, variable is not supported by ERRM					

Input data:

LV_IGK	LV_ERR_V_REF_1		

Configuration for diagnostic symptoms:

Diagnostic **SENS_ACR**	**Symptom description**	**Symptom**	**Filter type**
ACRC position sensor diagnosis	*Short circuit to battery or open load*	*SYM_0*	**STD_INI**
	Short circuit to ground	*SYM_1*	
	not used	*SYM_2*	
	not used	*SYM_3*	

17.14.1 General information

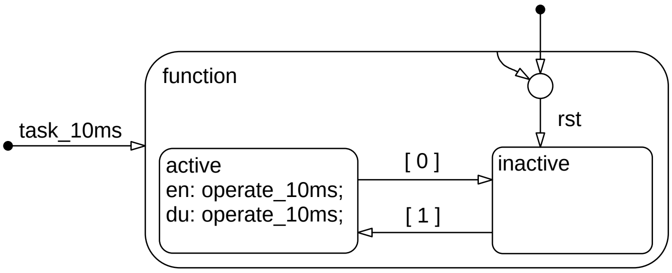
The flag LV_INH_DIAG_SENS_ACR permit to deactivate the corresponding diagnostic. As projects have different diagnosis information for actuator valve position sensor supply voltage, several error treatments are possible:

If there are no Information available, the error bit is set to 0 generally.

LV_ERR_VCC_ACR = 0

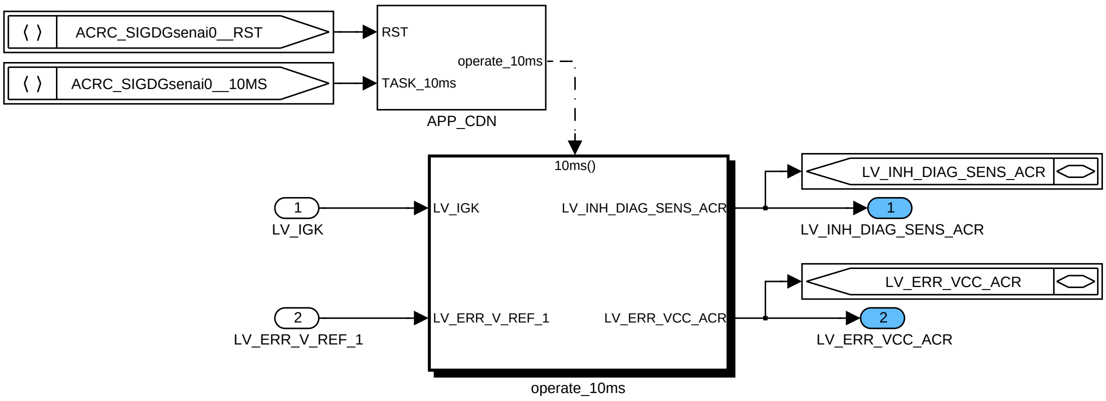
Otherwise the information bit has to be set to LV_ERR_VCC_ACR respectively.

Application Condition



ACRC_position_sensor_diagnosis_ai/APP_CDN/APP_CDN

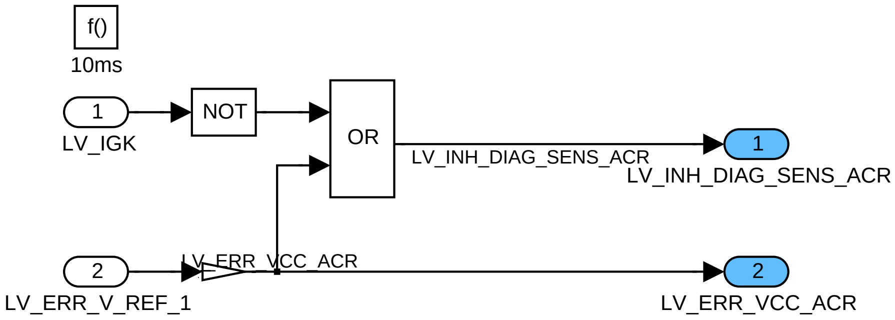
Function Description



ACRC_position_sensor_diagnosis_ai

17.14.1.1 Setting of the inhibition condition

This chapter describes the inhibition conditions for the sensor range check .



ACRC_position_sensor_diagnosis_ai/operate_10ms

17.15 ACRC power stage diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ACR_DR	O/V	0...1H	0...1	1	-
Present failure for power stage diagnosis after error debounce					
ERR_SYM_ACR_DR	V	0H 1H	NO_SYM SYM_0	1	-
Error symptom for the power stage diagnosis					
LV_CDN_DIAG_ACR_DR	V	0...1H	0...1	1	-
Diagnosis condition for power stage diagnosis					
LV_END_DIAG_ACR_DR	V	0...1H	0...1	1	-
End of diagnosis information for the power stage diagnosis					

Input data:

LV_INH_DIAG_ACR_DR

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_ACR_DR	1	0...FFH	0...255	1	-
Anti bounce counter increment for the power stage diagnosis					
C_ABC_MAX_ACR_DR	1	1...FFH	1...255	1	-
Maximum value for anti bounce counter for the power stage diagnosis					

Import actions:

ACTION_INFR_GetElDiagAcrDr(OUT <Lv_state_acr_dr> **)**
This action reads the status flag information of the ACR power stage!
ACTION_ERRM_FilterSymptom(IN<XX>, IN<lv_cdn_diag_XX>, IN< err_sym_XX>, IN<c_abc_inc_XX>, IN<c_abc_dec_XX>, IN<c_abc_max_XX>, OUT<lv_err_XX> **)**
This action computes the elementary anti-bounce filter for one failure treatment and returns filter result

17.15.1 General information

The used power stage is an intelligent full H-Bridge, designed for the control of DC and stepper motors in safety critical applications and under extreme environmental conditions. The H-Bridge is protected against over-temperature, short circuits and has an under voltage lockout for all supply voltages (main DC power supply). All malfunctions cause the output stages to go tristate.

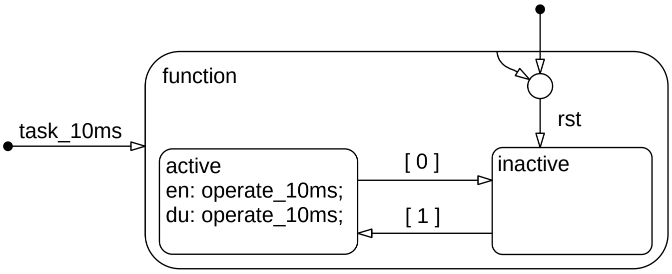
In this application the power stage is implemented in the status flag mode, that means indicated errors can not be distinguished by the software. The following errors can be detected only by the power stage:

Undervoltage / Overcurrent / Overtemperature / hardware disabled manually

Error-symptoms are defined to this diagnosis function as following:

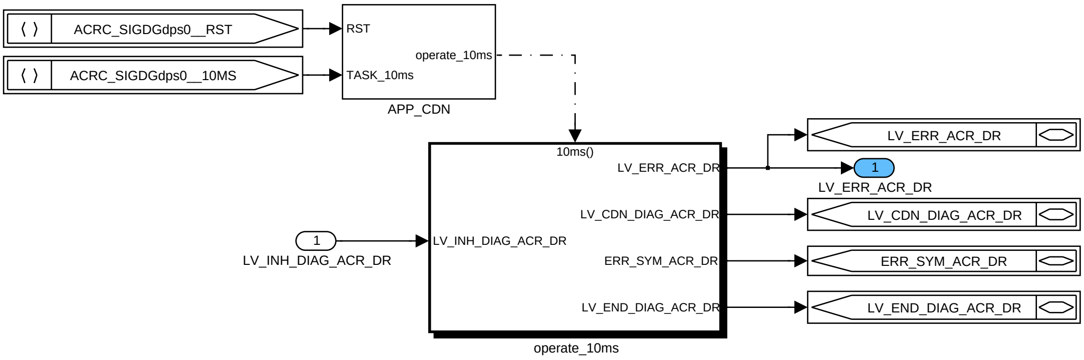
- SYM_0 - ACR power stage error

Application Condition



ACRC_power_stage_diagnosis/APP_CDN/APP_CDN

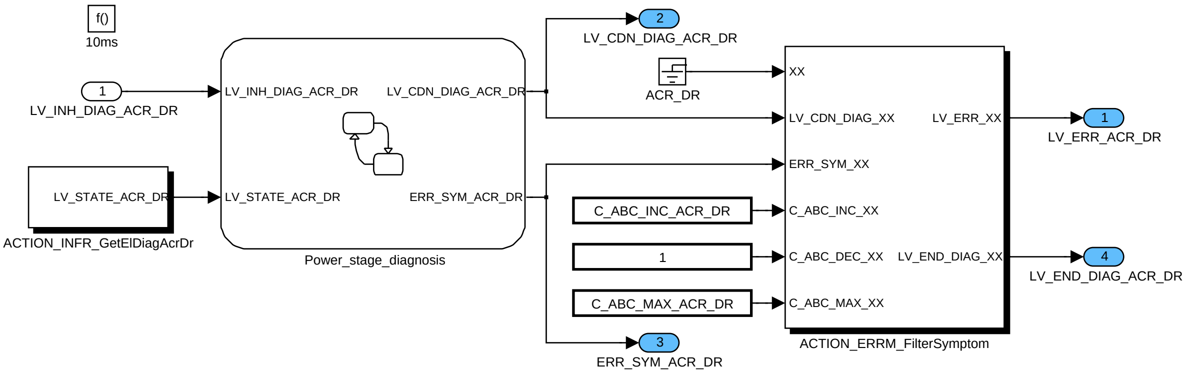
Function Description



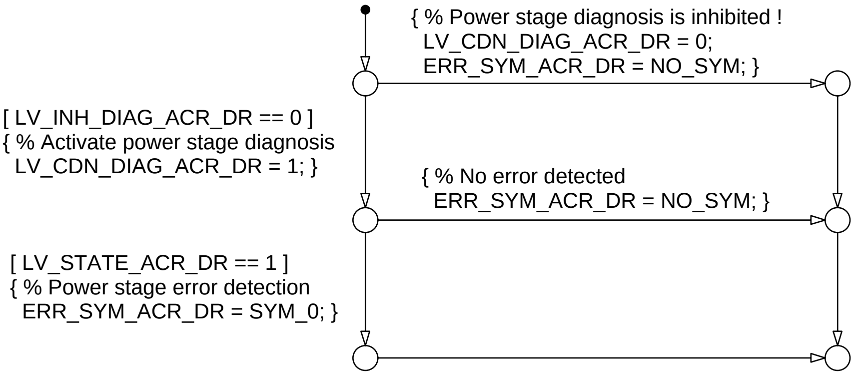
ACRC_power_stage_diagnosis

17.15.1.1 Power stage diagnosis

The status flag information from the power stage is read out with an infrastructure action at the beginning of the function and the anti-bounce filter algorithm from the generic error management is called as action at the end of the module.



ACRC_power_stage_diagnosis/operate_10ms



ACRC_power_stage_diagnosis/operate_10ms/Power_stage_diagnosis

17.16 ACRC power stage diagnosis based on SF (appl. inc.)

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
LV_INH_DIAG_ACR_DR	O/V	0...1H	0...1	1	-
Inhibition condition for the power stage diagnosis					

Input data:

LV_IGK	VB	STATE_ACR_CTL	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VB_MIN_ACR_DR_DIAG	1	0...FFH	0...25.8984	0.10156	V
Minimum threshold for the activation/deactivation of the power stage diagnosis					

Configuration for diagnostic symptoms:

Diagnostic **ACR_DR**	**Symptom description**	**Symptom**	**Filter type**
ACR power stage diagnosis	*ACR power stage error*	*SYM_0*	**STD_INI**
	not used	*SYM_1*	
	not used	*SYM_2*	
	not used	*SYM_3*	

17.16.1 General information

The flag LV_INH_DIAG_ACR_DR permits to deactivate the corresponding diagnostic.

Application Condition

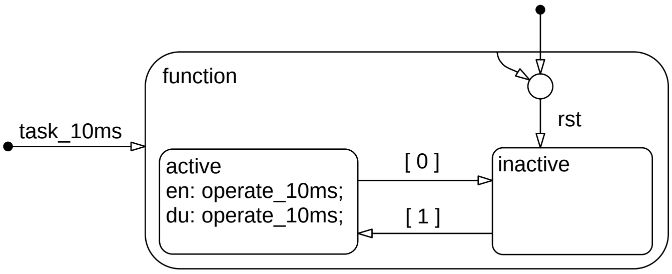


Figure 57 ACRC_power_stage_diag_SF_ai/ APP_CDN/ APP_CDN

Function Description

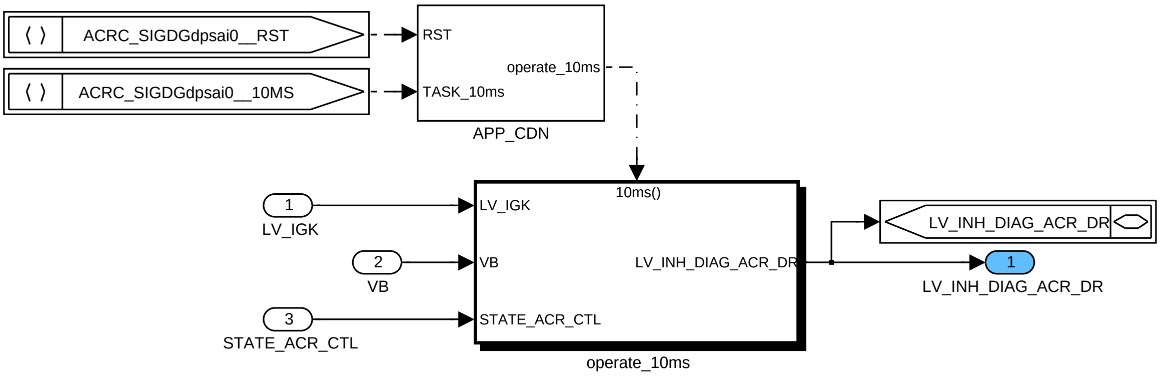


Figure 58 ACRC_power_stage_diag_SF_ai

17.16.1.1 Setting of the inhibition condition

This chapter describes the inhibition conditions for the power stage diagnosis.

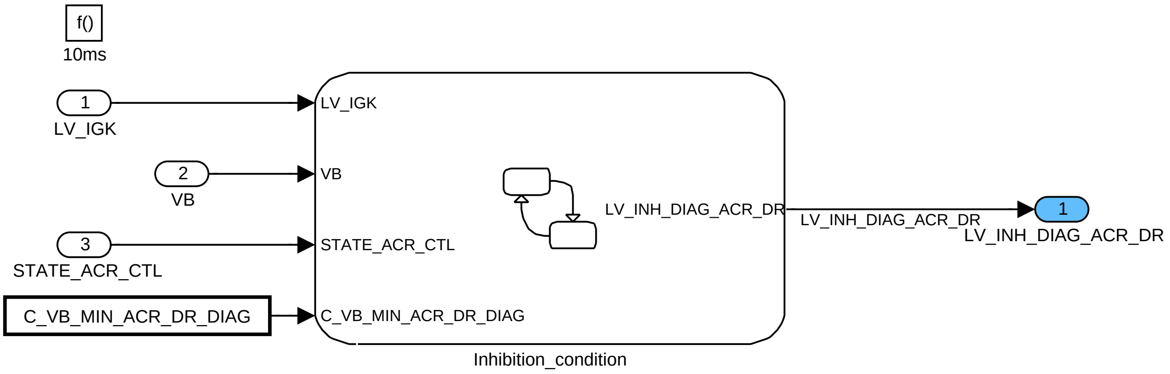


Figure 59 ACRC_power_stage_diag_SF_ai/ operate_10ms

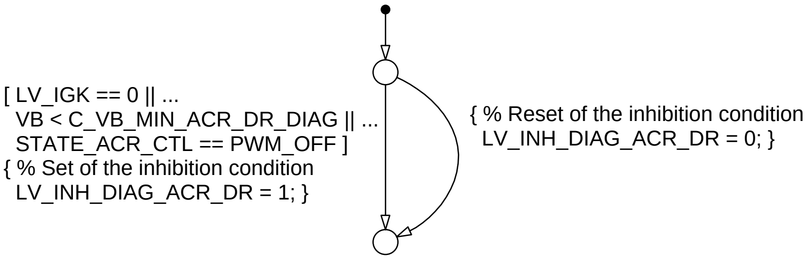


Figure 60 ACRC_power_stage_diag_SF_ai/ operate_10ms/ Inhibition_condition

17.17 ACRC actuator diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ACR_CTL	O/V	0...1H	0...1	1	-
Present failure for the actuator diagnosis after the error debounce					
CTR_DIAG_ACR_CTL	O/V	0...FFH	0...255	1	-
Anti bounce counter for the actuator diagnosis					
ERR_SYM_ACR_CTL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the actuator diagnosis					
LV_CDN_DIAG_ACR_CTL	V	0...1H	0...1	1	-
Diagnosis condition for the actuator diagnosis					
LV_END_DIAG_ACR_CTL	V	0...1H	0...1	1	-
End of diagnosis information for the actuator diagnosis					
PWM_ACR_MMV	V	0...7FFFH	0...99.9969482	0.003052	%
Moving mean value of the actuator position controller output					
OPG_SP_ACR_FIL	V	0...FFFH	0...99.9755859	0.024414	%
Actuator valve position setpoint after filtering by low pass first order					
OPG_SP_ACR_GRD	V	F000...FFFH	-100 ... 99.97559	0.024414	%
Gradient of the actuator valve position setpoint determined by low pass first order					
OPG_DIF_ACR_ERR_MAX	V	0...FFFH	0...99.9755859	0.024414	%
Maximum admissible control deviation of the closed-loop position control system					

Input data:

LV_INH_DIAG_ACR_CTL	VB_MMV	PWM_ACR	OPG_SP_ACR
OPG_DIF_ACR			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_ACR_CTL	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the actuator diagnosis					
C_ABC_INC_ACR_CTL	1	0...FFH	0...255	1	-
Anti bounce counter increment for the actuator diagnosis					
C_ABC_MAX_ACR_CTL	1	1...FFH	1...255	1	-
Maximum value for anti bounce counter for the actuator diagnosis					
C_CRLC_OPG_SP_ACR	1	0...FFFFH	0...0.99998474	1.5259E-5	-
Correlation constant for filtering of OPG_SP_ACR					
C_CRLC_PWM_ACR	1	0...FFFFH	0...0.99998474	1.52589-5	-
Correlation constant for filtering of PWM_ACR					
C_PWM_ACR_MAX_DIAG	1	0...7FFFH	0...99.9969482	0.003052	%
Maximum permitted PWM output of the ACR position controller					
C_VB_MIN_PWM_ACR_DIAG	1	0...FFH	0...25.8984375	0.1015625	V
Threshold depend on battery voltage for the activation of the PWM_ACR monitoring					
IP_OPG_DIF_ACR_ERR_MAX	4	0...FFFH	0...99.9755859	0.024414	%
LDP_OPG_SP_ACR_GRD_ABSV_IP_OPG	4	0...FFFH	0...99.9755859	0.024414	%
Maximum admissible control deviation dependent on the actuator valve setpoint gradient					

Import actions:

ACTION_ERRM_NoFilterSymptom(IN<XX>, IN<lv_cdn_diag_XX>, IN<err_sym_XX>, IN<lv_err_set_XX>, IN<lv_err_reset_XX>, IN<lv_end_diag_XX>, OUT<lv_err_XX> **)**
This action computes the elementary treatment case no filtering is used

17.17.1 General information

The task of this diagnosis is to detect an actuator valve error or a jammed actuator. The diagnostic observes the control deviation of the closed-loop position control system dependent on the gradient of the actuator valve position setpoint.

Additionally the position controller output is monitored by this diagnostic function. If the moving mean value of PWM_ACR exceeds a defined threshold than an error will be indicated. The calculation of the moving mean value is done by an low pass filter first order.

Application Condition

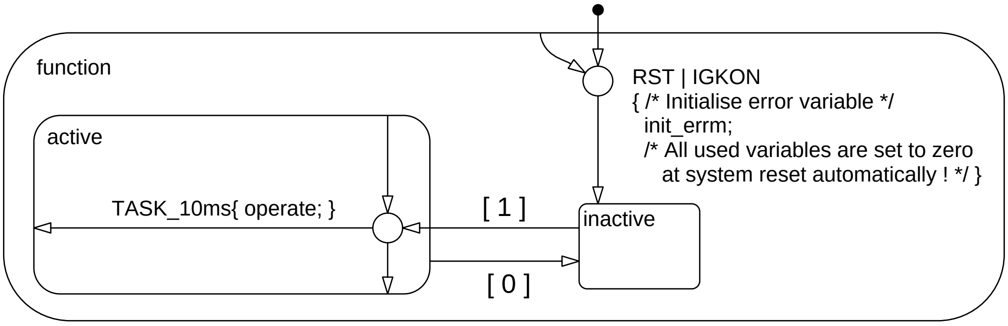


Figure 61 ACRC_actuator_diagnosis/ APP_CDN/ APP_CDN

Function Description

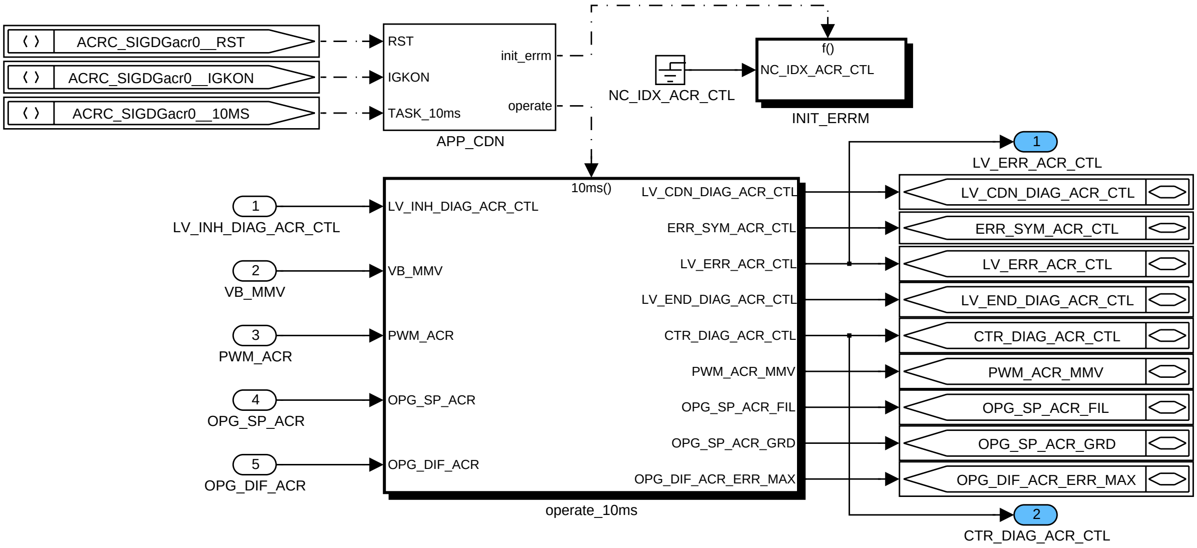


Figure 62 ACRC_actuator_diagnosis

17.17.1.1 Monitoring of the actuator valve function

The anti-bounce of the error detection is done at the end of the module without generic debounce algorithm. The increment and decrement value of the implemented algorithm can be calibrated.

The following error symptoms can be distinguished:

- PWM error of the closed loop position control system

- System deviation outside the admissible range

Error symptoms are defined to this diagnosis function as following:

SYM_0 – System deviation outside range

SYM_1 – PWM_ACR outside range

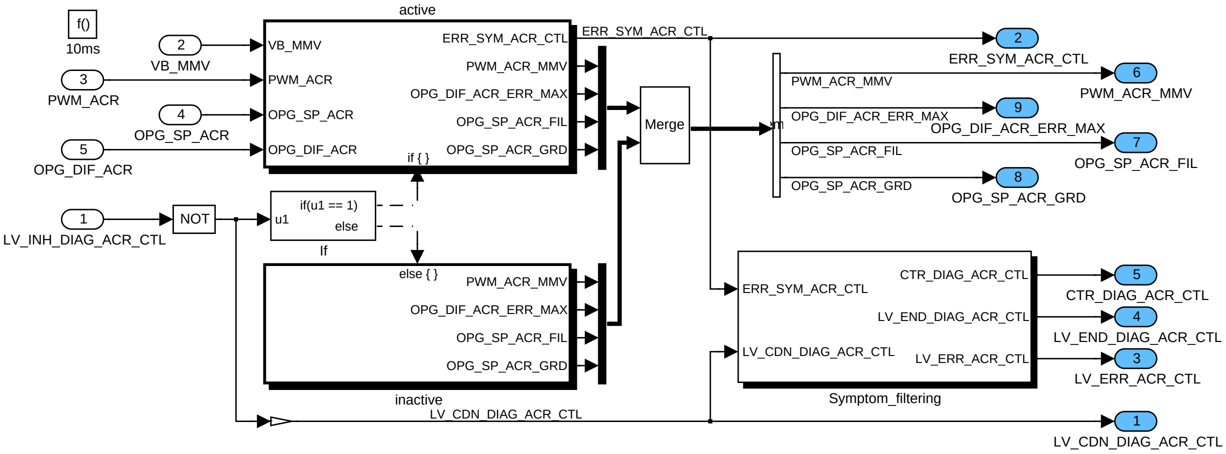


Figure 63 ACRC_actuator_diagnosis/ operate_10ms

Actuator failure detection

This chapter calculates the admissible error of the control deviation and the moving mean value of the position controller output, after than the error symptom determination will be performed !

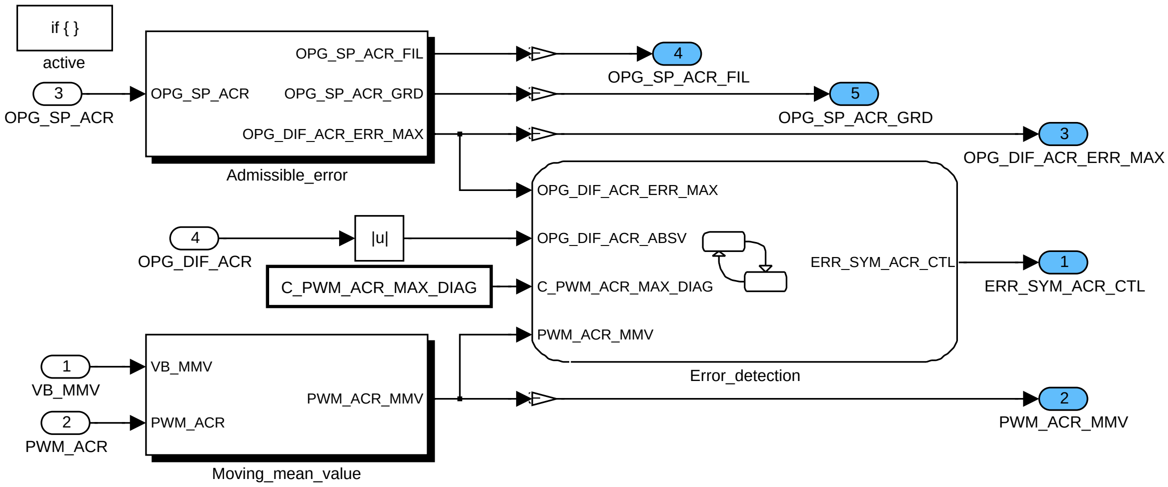


Figure 64 ACRC_actuator_diagnosis/ operate_10ms/ active

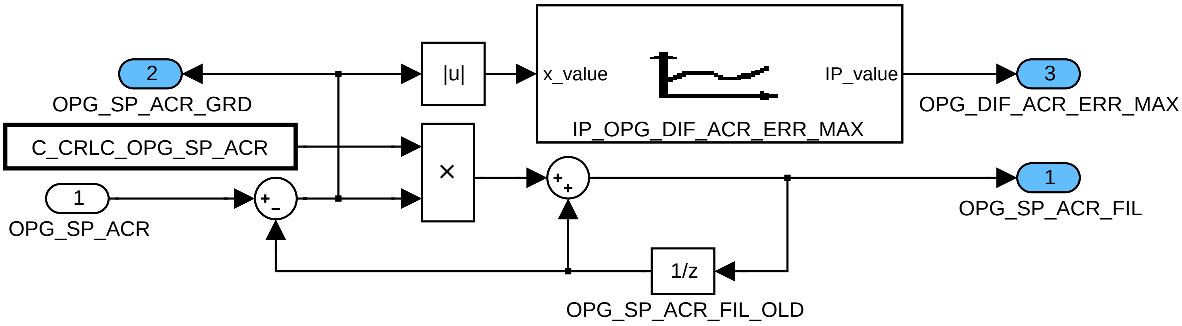


Figure 65 ACRC_actuator_diagnosis/ operate_10ms/ active/ Admissible_error

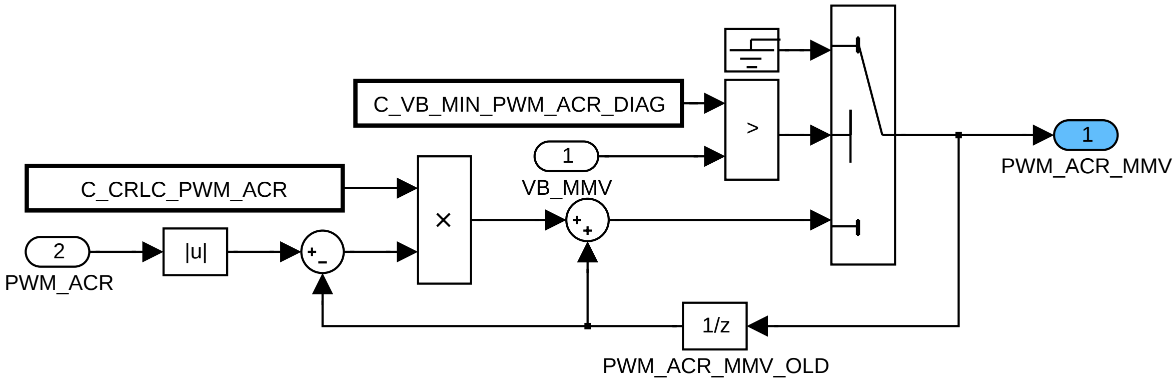


Figure 66 ACRC_actuator_diagnosis/ operate_10ms/ active/ Moving_mean_value

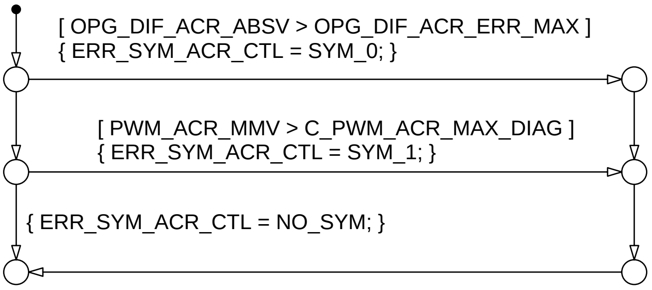
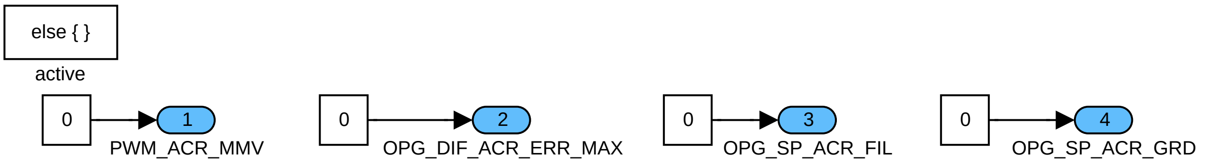


Figure 67 ACRC_actuator_diagnosis/ operate_10ms/ active/ Error_detection



The state variables PWM_ACR_MMV_OLD and OPG_SP_ACR_FIL_OLD from the active state are also set to zero

Figure 68 ACRC_actuator_diagnosis/ operate_10ms/ inactive

Error symptom filtering

Finally the anti-bounce algorithm for the determined error symptoms and the end of diagnosis information are calculated. In case of errors the debounced error variable can be set for the driving cycle with the corresponding inhibition condition.

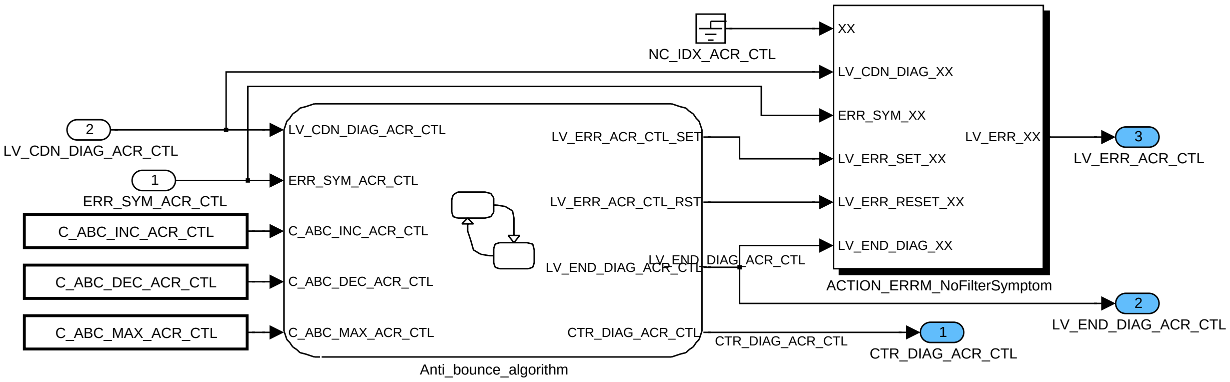


Figure 69 ACRC_actuator_diagnosis/ operate_10ms/ Symptom_filtering

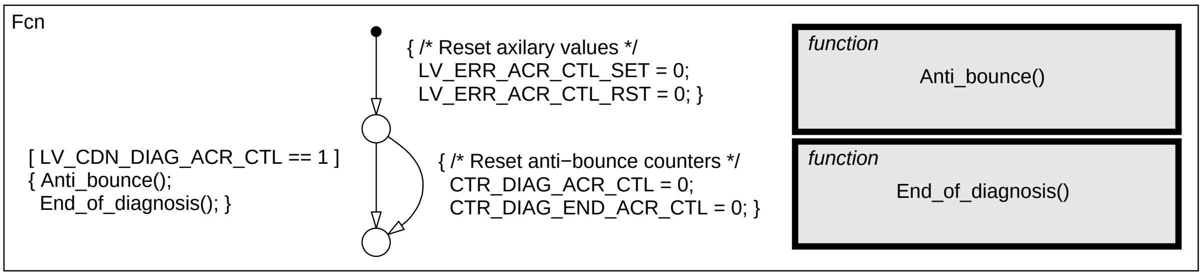


Figure 70 ACRC_actuator_diagnosis/ operate_10ms/ Symptom_filtering/ Anti_bounce_algorithm

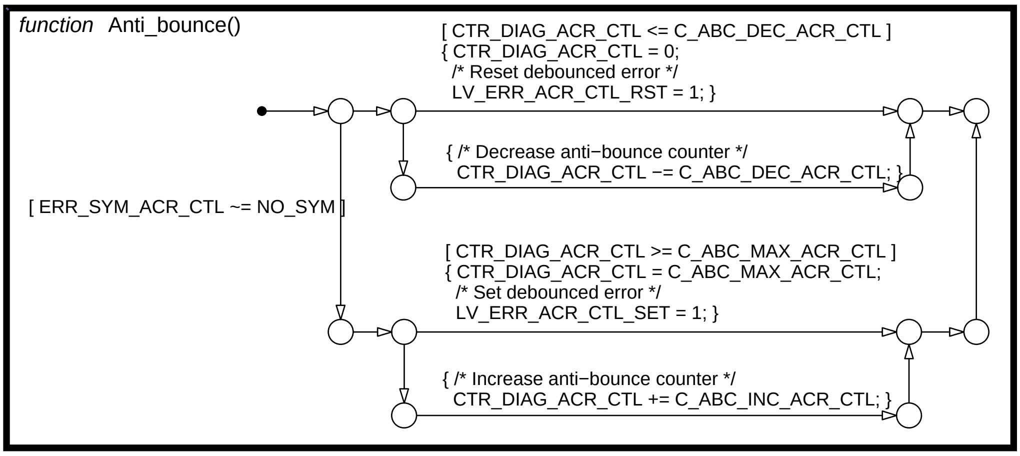


Figure 71 ACRC_actuator_diagnosis/ operate_10ms/ Symptom_filtering/ Anti_bounce_algorithm/ Fcn/ Anti_bounce

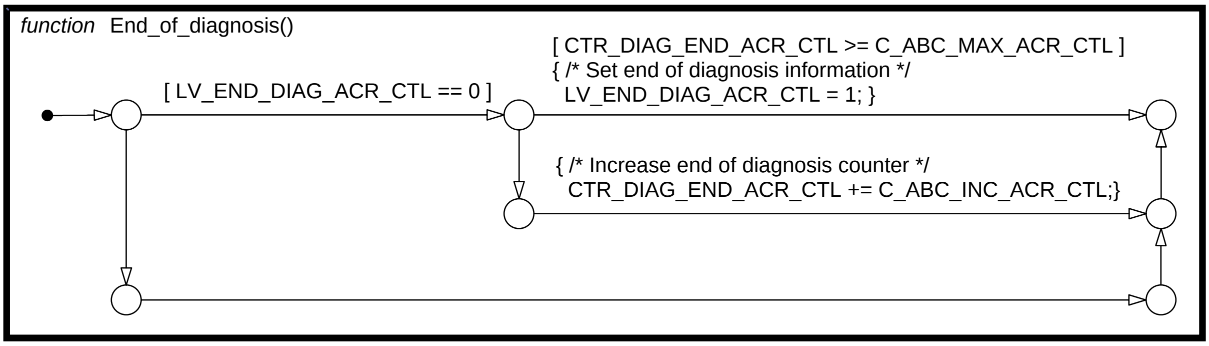


Figure 72 ACRC_actuator_diagnosis/ operate_10ms/ Symptom_filtering/ Anti_bounce_algorithm/ Fcn/ End_of_diagnosis

17.18 ACRC actuator diagnosis (appl. inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_ACR_CTL	O/V	0...1H	0...1	1	-
Inhibition condition for the actuator diagnosis					

Input data:

LV_IGK	STATE_ACR_CTL	LV_ACR_AD_ACT	

Configuration for diagnostic symptoms:

Diagnostic ACR_CTL	Symptom description	Symptom	Filter type
ACR actuator diagnosis	*System deviation outside range*	*SYM_0*	*NO*
	Output position controller outside range	*SYM_1*	
	not used	*SYM_2*	
	not used	*SYM_3*	

17.18.1 General information

The flag LV_INH_DIAG_ACR_CTL permits to deactivate the corresponding diagnostic.

Application Condition

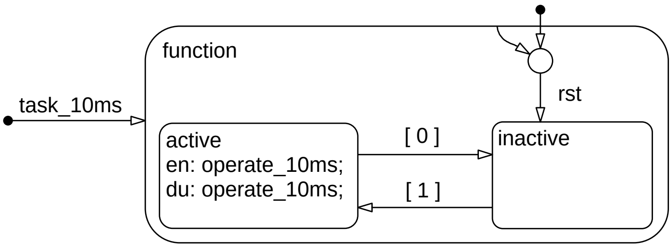


Figure 73 ACRC_actuator_diagnosis_ai/ APP_CDN/ APP_CDN

Function Description

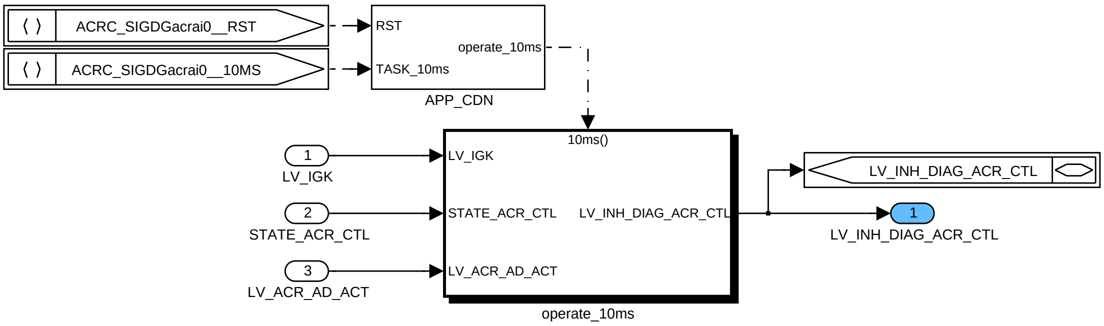


Figure 74 ACRC_actuator_diagnosis_ai

17.18.1.1 Setting of the inhibition condition

This chapter describes the inhibition condition for the actuator diagnosis.

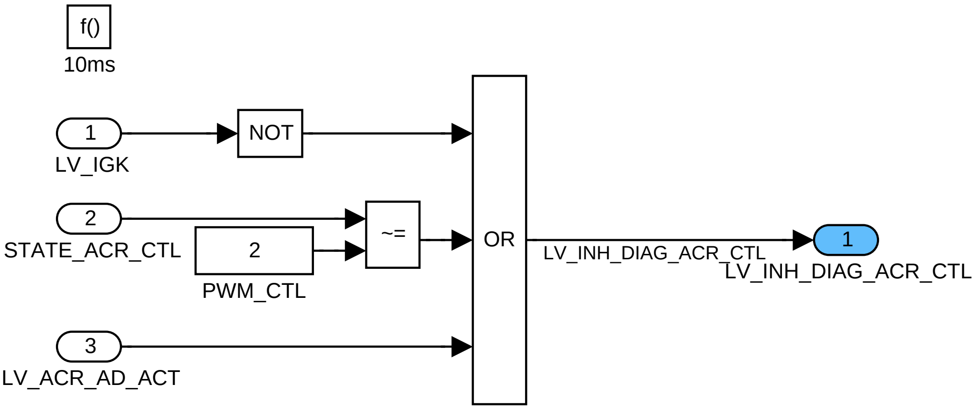


Figure 75 ACRC_actuator_diagnosis_ai/ operate_10ms

17.19 ACRC limp home management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_PWM_ACR_OFF_REQ	O/V	0...1H	0...1	1	-
ACR position control deactivation requested from monitoring level 1					
LV_OPG_ACR_INI_REQ	O/V	0...1H	0...1	1	-
Logical variable requests the replacement value for actuator signal acquisition					

Input data:

LV_ERR_VCC_ACR	LV_ERR_SENS_ACR	LV_ERR_ACR_DR	LV_ERR_ACR_CTL
LV_ERR_ACR_AD			

17.19.1 General information

The limp home management monitors the diagnostic functions of the closed-loop actuator control system. In case of monitoring level 1 errors the limp home function requests the deactivation of the power stage and actuator position control. The following diagnostic functions for ACRC position control are available:

- ACRC position sensor diagnosis
- ACRC power stage diagnosis
- ACRC actuator diagnosis
- ACRC adaptation diagnosis

Application Condition

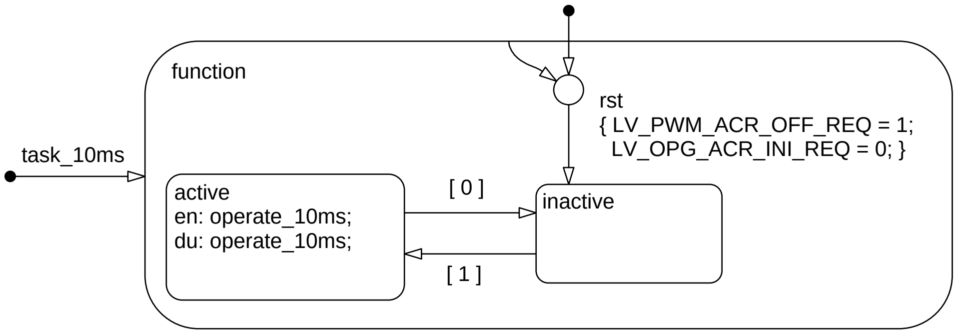


Figure 76 ACRC_limp_home_management/ APP_CDN/ APP_CDN

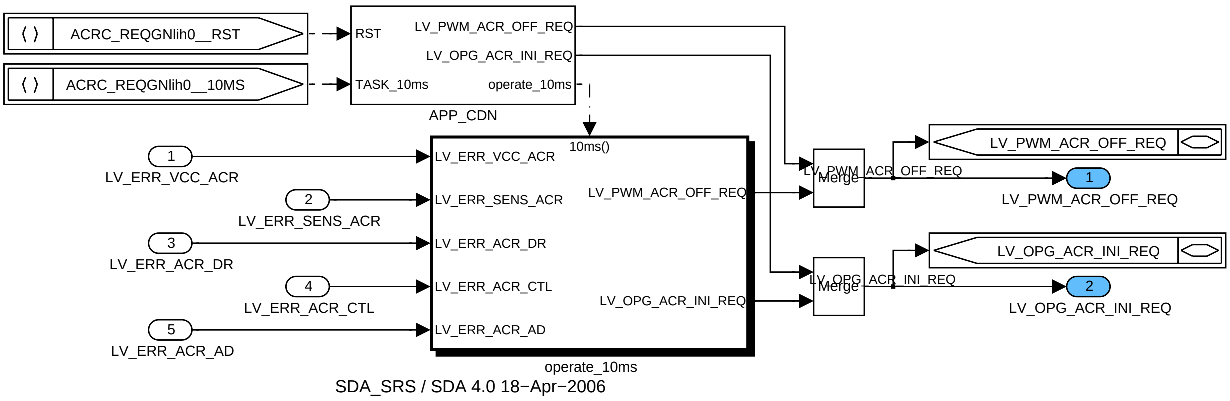


Figure 77 ACRC_LIMP_HOME_MANAGEMENT

17.19.2 Calculate switch-off request for ACR power stage

The following section calculates the power stage deactivation request with monitoring level 1 errors. Additionally the request for the replacement value of the sensor signal acquisition is determined.

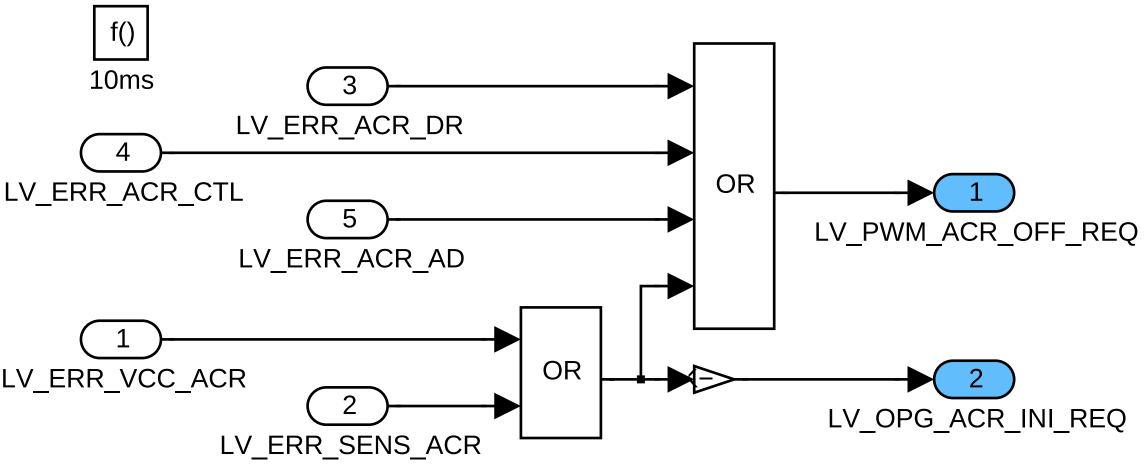


Figure 78 ACRC_LIMP_HOME_MANAGEMENT/ OPERATE_10MS

17.20 Calibration hints for actuator position control

17.20.1 Calibration interfering functions

No special requirements for the calibration with interfering functions are known.

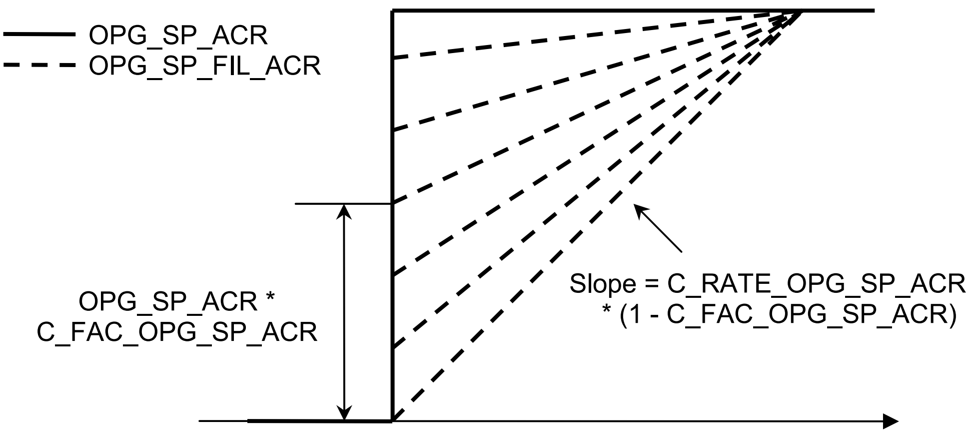
17.20.2 Calibration flowchart

The sequence is described by the order in calibrated methods.

17.20.3 Calibration method

Actuator position controller

C_FRQ_PWM_ACR	The calibration constant defines the frequency of the PWM output signal delivered from the position controller. The value is determined by the technical actuator specification and the ECU requirements (power loss and EMC of the actuator power stage). A recommend value is 1000 Hz!
C_RATE_OPG_SP_ACR C_FAC_OPG_SP_ACR	The setpoint filter function consists of a gradient limitation and a weighting between the setpoint gradient limitation and the direct setpoint request. The weighting can be chosen by the calibration constant C_FAC_OPG_SP_ACR see below



C_FAC_C_1_ACR_CTL C_FAC_C_2_ACR_CTL C_FAC_C_3_ACR_CTL C_FAC_F_1_ACR_CTL C_FAC_F_2_ACR_CTL C_FAC_F_3_ACR_CTL	The filter coefficients of the digital position controller are defined according to the internal SiemensVDO position controller design algorithm.
C_FAC_K_FIL_ACR_CTL	The static correction factor is calculated as follow:

$$\text{C\_FAC\_K\_FIL\_ACR\_CTL} = 1 + \sum_{i=1}^{3} \text{C\_FAC\_C\_i\_ACR\_CTL}$$

The correction factor can be fine-tuned with the external setpoint adjustment after successful operating of the position control system. A large setpoint of the control system shows the finally control deviation OPG_DIF. Increase or decrease the calibration value to correct the control deviation, finally OPG_DIF should be set to 0 % (a maximum difference of one AD converter increment is permitted).

C_KP_ACR_CTL	Usually the amplification factor of the digital position controller is determined also according to the internal SiemensVDO position controller design algorithm. But the position controller is a very robust controller algorithm that means the calibration value can be adapted to the requirements of the control system.
C_PWM_MIN_ACR_CTL C_PWM_MAX_ACR_CTL	Both variables limit the output of the digital position controller. Usually the operating range of the position controller is defined from -100 % to 100 % PWM duty-cycle.
C_PWM_MAX_ACR_AD	Limitation threshold during actuator valve adaptation. The value shall protect the actuator against destroyment. The limitation value has to be calibrated with respect of the battery voltage correction and the technical specifications made for the actuator.
IP_FAC_VB_PWM_ACR	The battery voltage correction is determined by a characteristic line. To avoid too high amplification values the data points are used to limit the correction value. With nominal value of the battery voltage the correction factor should be 1. Theoretical correction formula : $\mathrm{FAC\_VB\_PWM\_ACR} = \frac{\mathrm{VB\_MMV}_{\mathrm{nominal}}}{\mathrm{VB\_MMV}}$
IP_PWM_ACR_STAT	The characteristic line can be used for the calibration of the static preload of the actuator valve. Usually the characteristic line should be calibrated to zero, that means the position controller has to be able to compensate the static preload completely!
C_PWM_ACR_CLOSE	The calibration constant defines the static PWM duty-cycle to close the actuator valve with power after deactivation of the digital position controller. The value must be chosen lower than zero, a recommend value is -10 %.

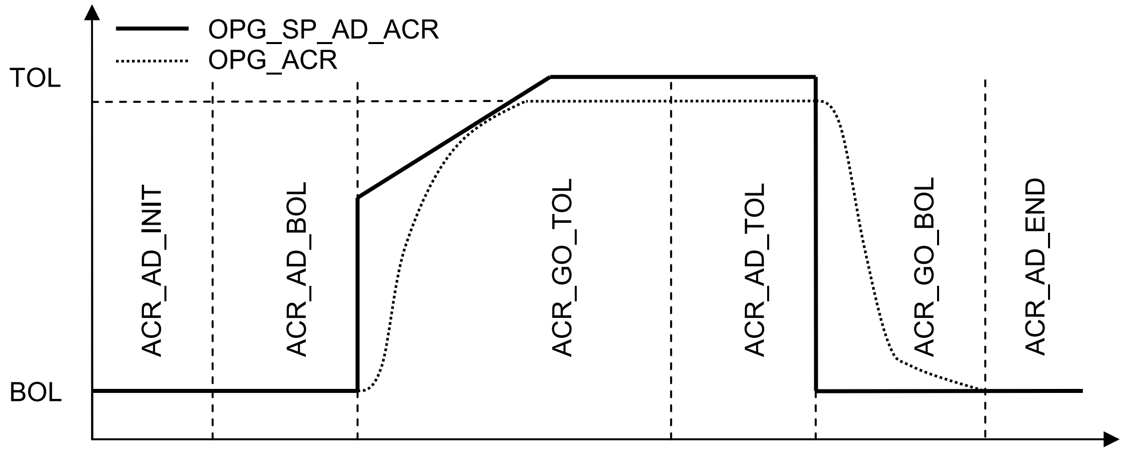
Actuator setpoint selection

C_OPG_SP_LIH_ACR	The limp-home position setpoint defines usually the setpoint request with deactivated control system. Normally the value is calibrated to 0 %.
C_OPG_SP_EXT_ACR	The calibration value is used for the setpoint adjustment requested from calibration system. The default calibration should be 0 %. **Note:** With active external setpoint request the position control system is not deactivated at small setpoint requests that means the position controller output can reach large PWM duty-cycles and can destroy the electrical actuator (monitor PWM_ACR_MMV).
C_OPG_SP_POP_ACR	The setpoint defines the setpoint request during the post operating phase of the electronic control unit (LV_IGK == 0). The setpoint request is used for the correct deactivation of the position control system. Calibrate the value with 0 %.
C_CTR_OPG_SP_POP_ACR	The counter variable describes the remaining activation of the closed-loop position control system with transition in the post-operating phase (LV_IGK == 0). Calibrate the value lower than 100 (that corresponded with 1 s).
C_OPG_SP_MIN_ACR_CTL	The described threshold is used to deactivate the position control system at small position setpoint requests. The position controller will be deactivated if the setpoint request

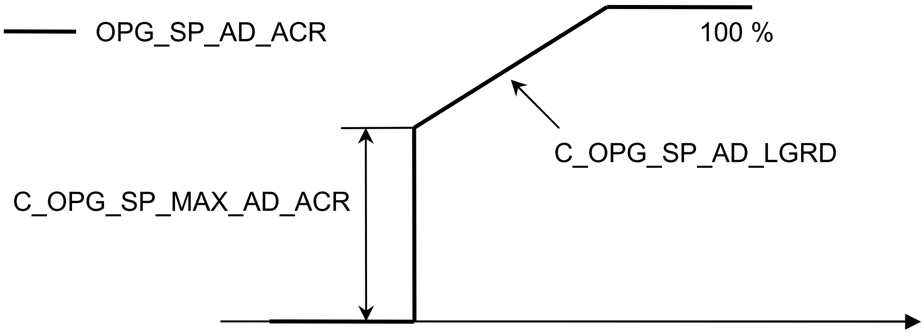
	falls below the threshold, recommend value is 0.3 %.
C_CTR_DLY_ACR_CTL	When the setpoint request falls below the above described threshold C_OPG_SP_MIN_ACR_CTL then a delay counter will be started and if the delay counter is elapsed then the digital position controller will be deactivated. The delay time has to be chosen with respect to actuator dynamic. That means the actuator valve must be able to reach the requested setpoint before deactivation. e.g. C_CTR_DLY_ACR_CTL = 25 that are 250 ms
C_OPG_SP_MIN_ACR	The lower setpoint limitation is used to limit the setpoint request in lower direction. The setpoint and the actual value of the control system have been implemented in unsigned integer. To get a negative control deviation OPG_DIF in each case the limitation value has to be calibrated larger than zero. Usually a recommend value is 0.2 %!
C_OPG_SP_MAX_ACR	The upper setpoint limitation is used to limit the setpoint request in upper direction. The aim is to protect the upper mechanical stop of the actuator and to get an additional sensor distance for the sensor range check diagnosis. The value should be chosen with respect to the tolerances of the ECU and the actuator measuring system.
LC_PWM_ACR_OFF_ENA	The configuration constant defines the state of the actuator power stage in normal operating with deactivated position controller. The algorithm can be distinguished two function modes. • LC_PWM_ACR_OFF_ENA = 0 The power stage is in the active state after deactivation of the digital position controller and the calibration constant C_PWM_ACR_CLOSE set the PWM duty-cycle for active valve closing. • LC_PWM_ACR_OFF_ENA = 1 The power stage will be deactivated via enable/disable line with deactivation of the digital position controller.

Actuator adaptation and diagnosis

LC_ACR_AD_REQ	The "first key on" actuator adaptation can be requested by a manual adaptation request via calibration system. Set the logical constant and an actuator adaptation will be performed with each reset.
LC_ACR_AD_POP_ENA	Logical constant to perform the actuator adaptation during post operation phase (LV_IGK = 1 ⇒ 0) ! The constant has to be set for an adaptation request.
C_ACR_AD_VB_MIN	Lowest admissible battery voltage to enable actuator valve adaptation. The calibration constant must be calibrated with respect to the position controller limitation threshold. Usually the value is set to 10 V according to battery voltage correction FAC_VB_PWM_ACR !
C_ACR_AD_TCO_MIN	Minimum engine temperature to enable actuator valve adaptation. The threshold shall inhibit the valve adaptation with iced actuator. Calibrate larger than 0 °C !



C_V_ACR_AD_BOL_INI	The calibration value should contain the smallest possible sensor voltage of the measuring system for the lower mechanical stop. The value is defined in the data sheet of the actuator, e.g. 0.2 V!
C_V_ACR_AD_TOL_INI	The position controller must reach the upper mechanical stop therefore the value must contain the largest possible sensor signal voltage for the upper mechanical stop, e.g. 4.9V and larger!
C_FAC_ACR_SLOP	The nominal value of the sensor slope is defined in the data sheet of the actuator valve. Usually the recommend value is 27.8 %/V!
C_CTR_ACR_AD	Counter variable for the adaptation of the lower and upper mechanical stop to get valid adaptation values. The adaptation interval must be calibrated with respect to the correlation constant C_CRLC_ACR_AD, see below. A value of 10 corresponds with 100 ms adaptation time.
C_CTR_DLY_ACR_AD	Delay counter for the activation of the lower and upper mechanical stop adaptation. A recommend value is 10.
C_CTR_DLY_ACR_AD_TOL	Delay counter for the adaptation of the upper mechanical stop. A recommend value is 10.
C_CTR_DLY_ACR_AD_MAX	Delay counter for the continuously adaptation of the lower mechanical stop. The value is used to get a valid sensor signal for the lower mechanical stop. Calibrate the value with 20 that corresponds with 2.0 s delay time.
C_CTR_DIAG_ACR_AD	Diagnostic counter for the check of the adaptation value during continuously BOL adaptation. Usually the value is set to 10 (correspond with 1 s error debounce time).
C_CTR_DIAG_END_ACR_AD	End of diagnosis counter during continuously adaptation of the lower mechanical stop. A recommend value is 20.
C_OPG_SP_MAX_AD_ACR C_OPG_SP_AD_LGRD	The position controller must drive the actuator valve in the upper mechanical stop before the adaptation can be started. C_OPG_SP_MAX_AD_ACR is used to reach the upper stop into a short time and C_OPG_SP_AD_LGRD is used to protect the actuator valve, that means the valve reach the upper mechanical stop with speed limitation. • C_OPG_SP_MAX_AD_ACR = 80 % • C_OPG_SP_AD_LGRD = 0.5 %



C_CRLC_ACR_AD	The correlation constant is used for the calculation of the moving mean values used for determination of the sensor adaptation values. The filter rate of the filter algorithm is increased with falling correlation constant. The constant must be calibrated with consideration of the adaptation interval C_CTR_ACR_AD, e.g. C_CRLC_ACR_AD = 0.02!
C_V_ACR_AD_CHK_CTL	The calibration threshold is used for the check of the actuator function. If the actuator valve is adjusted in the upper mechanical stop then the described calibration threshold must be exceed otherwise an adaptation error occurs. The value must lie outside the diagnostic windows of the lower and upper mechanical stop, • C_V_ACR_AD_CHK_CTL = 2.5 V.
C_V_ACR_AD_BOL_MIN C_V_ACR_AD_BOL_MAX	The diagnostic thresholds for the lower mechanical stop are defined by the data sheet of the actuator and tolerance calculation made for the ECU and the measuring system, possible range: 0.4 V – 0.7 V!
C_V_ACR_AD_TOL_MIN C_V_ACR_AD_TOL_MAX	The diagnostic thresholds for the upper mechanical stop are defined by the data sheet of the actuator and tolerance calculation made for the ECU and the measuring system, possible range: 4.4 V - 4.7 V!
LC_FAC_ACR_SLOP_AD_ENA	The logical constant allows to activate the adaptation of upper mechanical stop and the corresponding sensor signal slope during the "first key on" and the continuously adaptation. The value must be calibrate to one !

17.21 Calibration hints for actuator diagnosis

17.21.1 Calibration interfering functions

No special requirements for the calibration with interfering functions are known.

17.21.2 Calibration flowchart

The sequence is described by the order in calibrated methods.

17.21.3 Calibration method

Actuator position sensor diagnosis

C_V_MIN_SENS_ACR C_V_MAX_SENS_ACR	The electrical diagnostic thresholds define the admissible range of the actuator valve sensor signal. Both values are the result of the worst-case tolerance calculation made for the measuring system and the correct choice of the upper position setpoint limitation C_OPG_SP_MAX_ACR. Usually the admissible range should be defined from 0.15 V to 4.85 V. When the admissible diagnostic range for the upper signal range check is to small then a reduction of C_OPG_SP_MAX_ACR can solve the problem.
C_ABC_INC_SENS_ACR C_ABC_MAX_SENS_ACR	Calibrate the debounce counter increment with 2 and the maximum value with 40 for the electrical diagnosis, in this case the corresponding debounce time is 200 ms.

Actuator power stage diagnosis based on SF mode

C_VB_MIN_ACR_DR_DIAG	The power stage diagnosis is inhibited below the battery voltage threshold to get only valid diagnostic information from the power stage status flag.
C_ABC_INC_ACR_DR C_ABC_MAX_ACR_DR	Calibrate the debounce counter increment with 2 and the maximum value with 40 for the power stage diagnosis, in this case the corresponding debounce time is 200 ms.

Actuator diagnosis

C_ABC_INC_ACR_CTL C_ABC_DEC_ACR_CTL C_ABC_MAX_ACR_CTL	Calibrate the anti-bounce counter decrement larger than the increment value. In case of actuator malfunctions the diagnostic function has to indicate errors with a maximum admissible debounce time of 500 ms. The anti-bounce counter maximum value can be calculate as follow : C_ABC_MAX_ACR_CTL = 500 ms / 10 ms *... C_ABC_INC_ACR_CTL Recommend values are: • C_ABC_INC_ACR_CTL = 1 • C_ABC_DEC_ACR_CTL = 5 • C_ABC_MAX_ACR_CTL = 50

C_CRLC_OPG_SP_ACR	The calculation of the actuator position setpoint gradient is performed by a low pass filter of first order. The described correlation constant defines the corresponding filter time constant. The constant can be calculated by the definition of the continuously time constant: $\mathrm{C\_CRLC\_OPG\_SP\_ACR} = 1 - e^{-\frac{0.01\mathrm{s}}{T_{\mathrm{request}}}}$ A time constant smaller than 100 ms should be defined, e.g. C_CRLC_OPG_SP_ACR = 0.181 (50ms) !
IP_OPG_DIF_ACR_ERR_MAX	The characteristic map defines the admissible control deviation error dependent on the actuator position setpoint gradient. The admissible error has to be increase with rising actuator setpoint gradient.
C_PWM_ACR_MAX_DIAG	If the moving-mean value of the position controller output duty-cycle exceeds the defined threshold than an error will be indicated. The diagnosis threshold can be set to 95% with respect to correlation constant of the corresponding low pass filter, see the next label.
C_CRLC_PWM_ACR	The calculation of the moving mean value is done by a first order low pass filter. The correlation factor describes the time constant of the low pass filter and can be calculated as follows: $\mathrm{C\_CRLC\_PWM\_ETC} = 1 - e^{-\frac{0.01\mathrm{s}}{T_{\mathrm{request}}}}$ With a chosen time constant of 1 s the correlation constant can be calculated with 0.01, that means after a PWM step from zero to 100% PWM duty-cycle the above mentioned diagnostic threshold of 95% will be reached after 3*T_{request} (3 s) according to the low pass filter characteristic.
C_VB_MIN_PWM_ACR_DIAG	If the battery voltage falls below the described threshold than the component protection is deactivated. A value of approximately 10V is recommended.

17.22 Customer adaptation module, ACRC

17.22.1 BMW software to ACRC adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
OPG_SP_REQ_ACR	V/O	0...FFFH	0...99.97558	0.0244141	[%]
Actuator valve position setpoint request in normal operation					
EGR_RATIO_SP	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
EGR-Ratio Setpoint					

Input data:

Agrpos_soll	Agr_soll	LV_ACR_CTL_ENA	

FUNCTION DESCRIPTION:

This function is used for the adaptation to BMW customer software. *Remark:* the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes. The variables has to be executed before the ACRC functionality is calculated.

Application conditions:

Initialization:	at RST all to zero
Recurrence:	10 ms
Activation:	at all engine operating states
Deactivation:	no deactivation

Formula section:

EGR_RATIO_SP = Agr_soll

```
if      LV_ACR_CTL_ENA == 1
then    OPG_SP_REQ_ACR = Agrpos_soll
else    OPG_SP_REQ_ACR = 0
endif
```

17.22.2 ACRC to BMW software adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Agrpos	V/O	0...FFFFH	0...99.9985	0.001526	[%]
Current EGR valve position delivered to BWM software					
B_agr_ktrl	V/O	0...1H	0...1	1	[-]
EGR control active					

Input data:

OPG_ACR	LV_ACR_CTL_ENA		

FUNCTION DESCRIPTION:

This function is used for the adaptation to BMW customer software. *Remark:* the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes. The variable has to be executed after the ACRC functionality is calculated.

Application conditions:

Initialization: at RST set all to

Recurrences: 10 ms

Activation: at all engine operating states

Deactivation: no deactivation

Formula section:

B_agr_ktrl = LV_ACR_CTL_ENA
Agrpos = OPG_ACR

18 Exhaust gas treatment system

Table of Contents

Index

18.1 AGGR EGTR adaptation

18.1.1 Outputs for SV aggregates, SV internally

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CAT_DIAG_MOD_6[NC_CBK_EX_NR]	O/V/S	0...7FH	0...1.98437	0.015625	[-]
Cat. Diag. value for OBD Scantool Mod 6 output					
CAT_MAX_DIAG_MOD_6[NC_CBK_EX_NR]	O/V/S	0...7FH	0...1.98437	0.015625	[-]
Actually threshold for Scantool Mod 6 Output					
DELTA_LAMB_SP_O2L_MDL[NC_CBK_EX_NR]	O/V	F800...800H	-0.125...0.125	0.061e-3	[-]
lambda set point shift of catalyst O2 loading condition function					
LV_LAM_ADJ_RNG_VLD_CAT_DIAG[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Boolean flag indicating Trim Ctr P-Share is inside a calibrateable window					
LV_NT_REQ_RGN	O/V	0...1H	0...1	1	[-]
Logical value for regeneration phase request					

Input data:

EFF_CAT_DIAG_OBD[NC_CBK_EX_NR]	EFF_CAT_MAX_DIAG_OBD[NC_CBK_EX_NR]	LAMB_DELTA_P_LAM_ADJ[NC_CBK_EX_NR]	LV_RGN_NT_REQ
NC_CBK_EX_NR			

FUNCTION DESCRIPTION:

Adaptation to aggregate environment.

Application conditions:

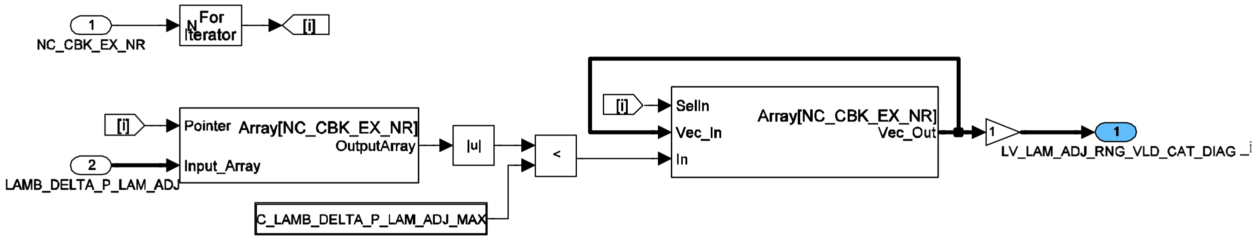
Initialisation:	0
Recurrence :	same recurrence as corresponding input data
Activation:	every engine operating state

Formula section:

LV_NT_REQ_RGN = LV_RGN_NT_REQ

CAT_DIAG_MOD_6[i] = EFF_CAT_DIAG_OBD[i]

CAT_MAX_DIAG_MOD_6[i] = EFF_CAT_MAX_DIAG_OBD[i]



Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_DELTA_P_LAM_ADJ_MAX	1	F800...7FFH	-0.125...0.12493	0.061e-3	[-]
max. P share threshold of trim controller for catalyst diagnosis activation					

18.2 Coordination of forced lambda stimulation

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
LAMB_PLS[NC_CBK_EX_NR]	O/V	F800...7FFH	-0.125 ... 0.12493896	6.10352E-5	-
Lambda deviation for the Lambda Controller					
LV_LAMB_PLS_AFL_COR_ACT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that corrective action for lean half wave of lambda stimulation is active					
LV_LAMB_PLS_AFR_COR_ACT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that corrective action for rich half wave of lambda stimulation is active					

Input data:

STATE_LAMB_PLS_DET_VALUE	LAMB_PLS_EXT[NC_CBK_EX_NR]	FAC_DIF_LAM_IN_MMV[NC_CBK_EX_NR]	NC_CBK_EX_NR
LV_ST_END	LAMB_PLS_O2L_OSC[NC_CBK_EX_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_DIF_THD_AFL_PLS_OFF	1	0...7FFFH	0...0.99996948	3.05176E-5	-
lambda control difference threshold for deactivating lean stimulation half wave					
C_FAC_DIF_THD_AFL_PLS_ON	1	8000...7FFFH	-1...0.99996948	3.05176E-5	-
lambda control difference threshold for re-activating lean stimulation half wave					
C_FAC_DIF_THD_AFR_PLS_OFF	1	8000...0H	-1...0	3.05176E-5	-
lambda control difference threshold for deactivating rich stimulation half wave					
C_FAC_DIF_THD_AFR_PLS_ON	1	8000...7FFFH	-1...0.99996948	3.05176E-5	-
lambda control difference threshold for re-activating rich stimulation half wave					
LC_LAMB_PLS_COR_ACT_IS	1	0...1H	0...1	1	-
calibration flag indicating that corrective action for lambda stimulation is allowed in idle speed					
LC_LAMB_PLS_EXT	1	0...1H	0...1	1	-
flag indicating that external LAMP_PLS is used					

18.2.1 EGTR_DEFSPFSTIC0

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements, the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

Application Condition

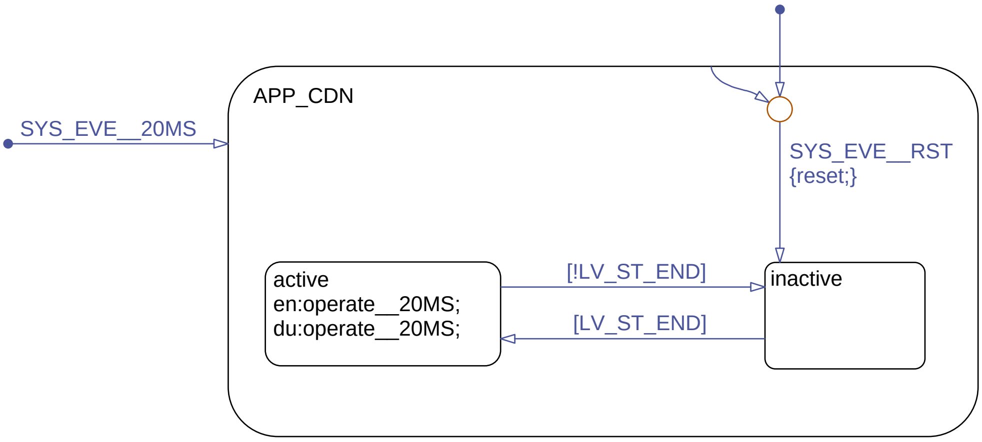
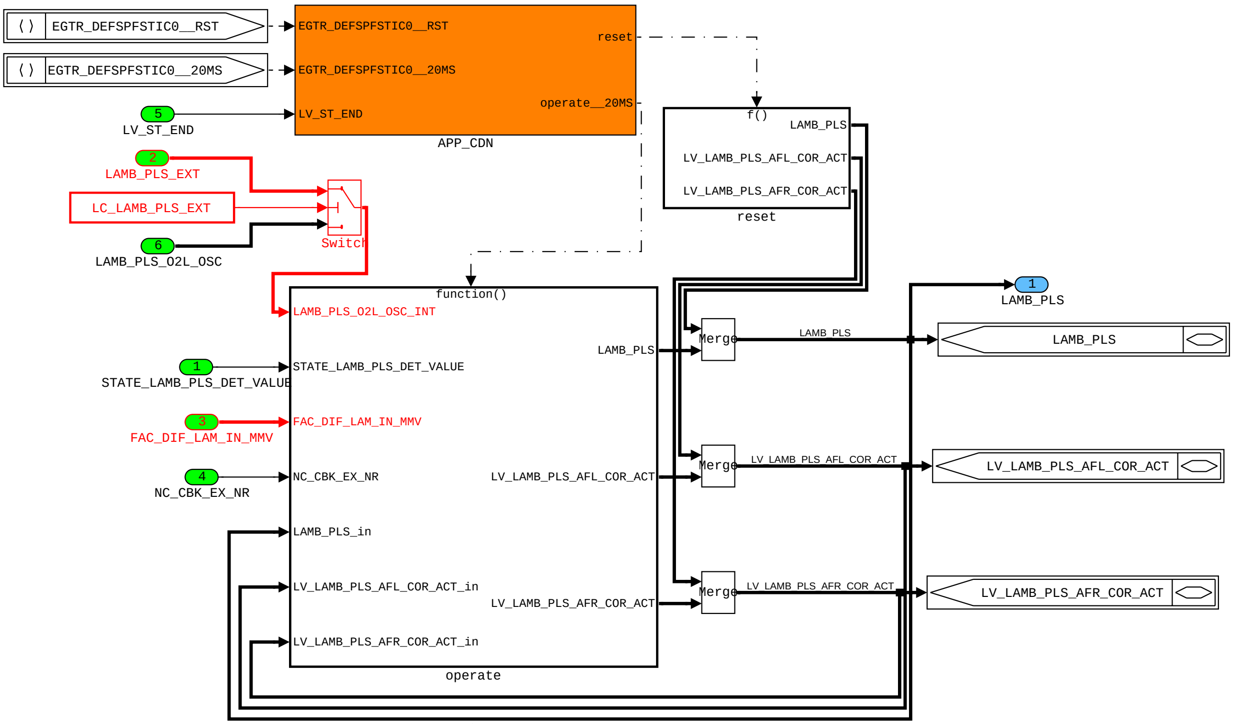


Figure 1 EGTR_DEFSPFSTIC0/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 25−Jan−2005

Figure 2 EGTR_DEFSPFSTIC0

18.2.1.1 SUBFUNCTION: operate

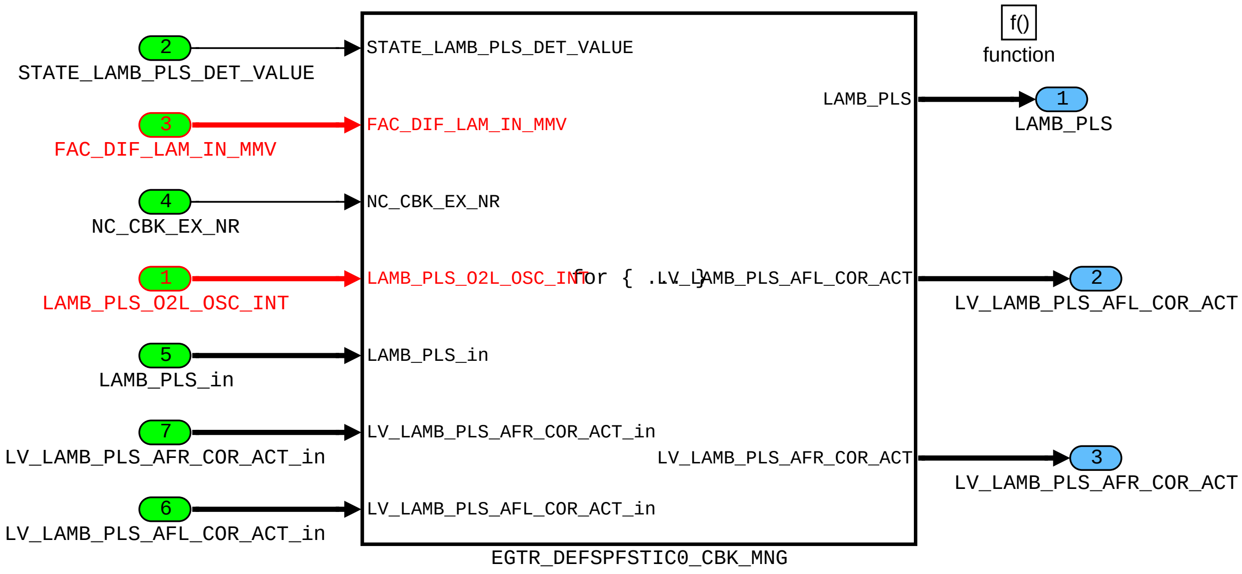


Figure 3 EGTR_DEFSPFSTIC0/ operate

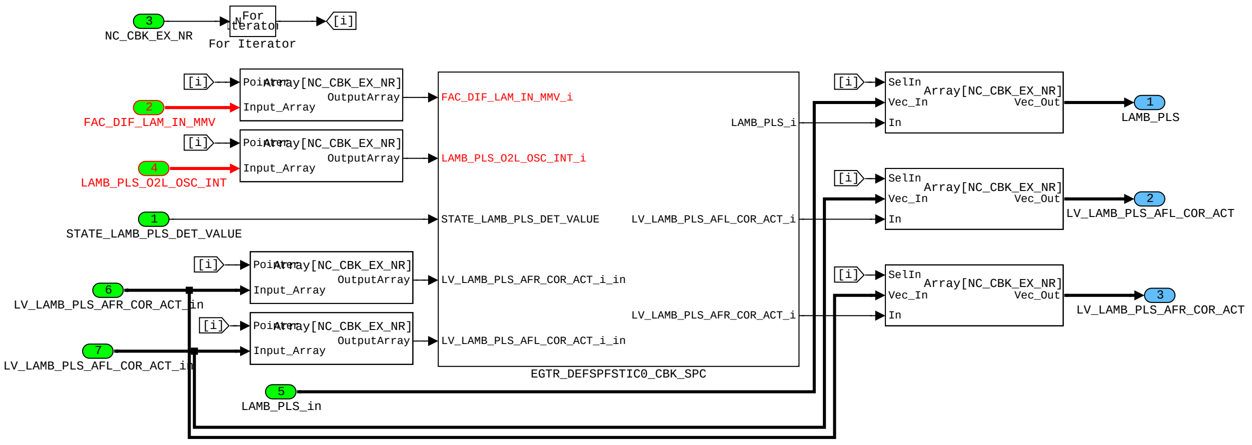


Figure 4 EGTR_DEFSPFSTIC0/ operate/ EGTR_DEFSPFSTIC0_CBK_MNG

Corrective action depending on lambda cotnroller difference

Depending on the lambda control difference FAC_LAM_DIF_IN[i] a corrective intervention on the lambda pulse can be applied. In case of too lean mixture the lean deviation can be suppressed and in case of too rich mixture the rich deviation can be suppressed. Activation and deactivation of the corrective intervention is realised with different thresholds in order to apply a hysteresis.

With LC_LAMB_PLS_COR_ACT_IS this corective action can also be applied in idle speed.

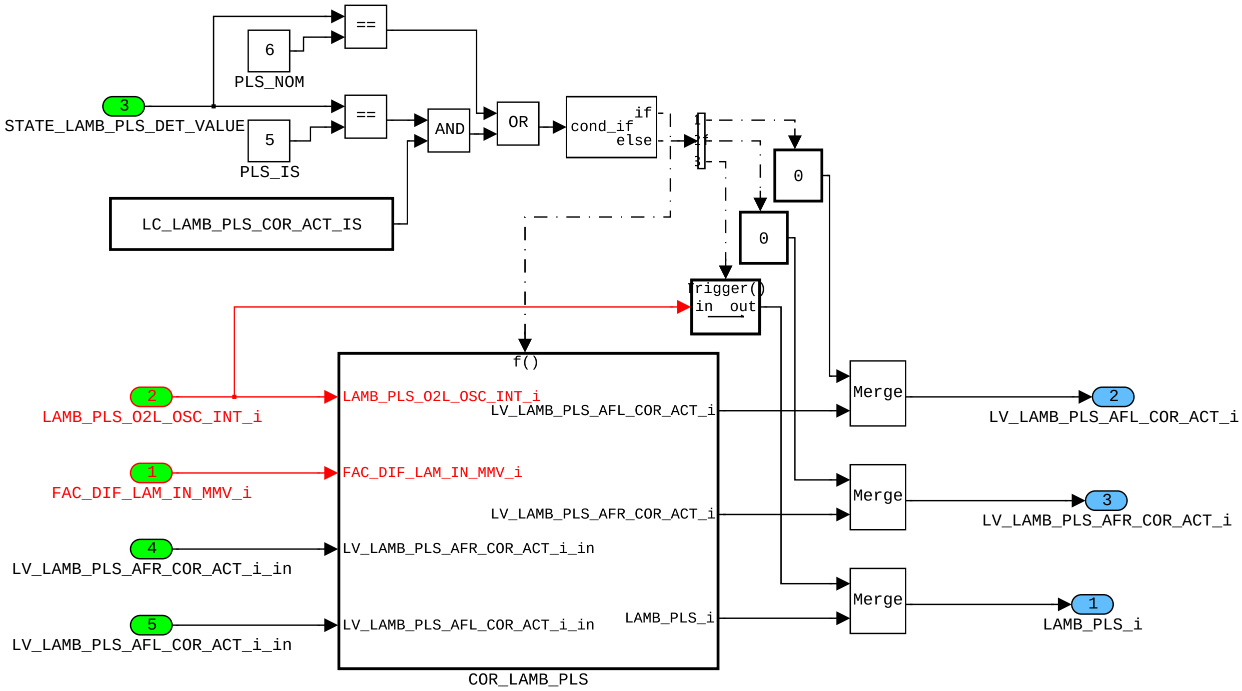


Figure 5 EGTR_DEFSPFSTIC0/ operate/ EGTR_DEFSPFSTIC0_CBK_MNG/ EGTR_DEFSPFSTIC0_CBK_SPC

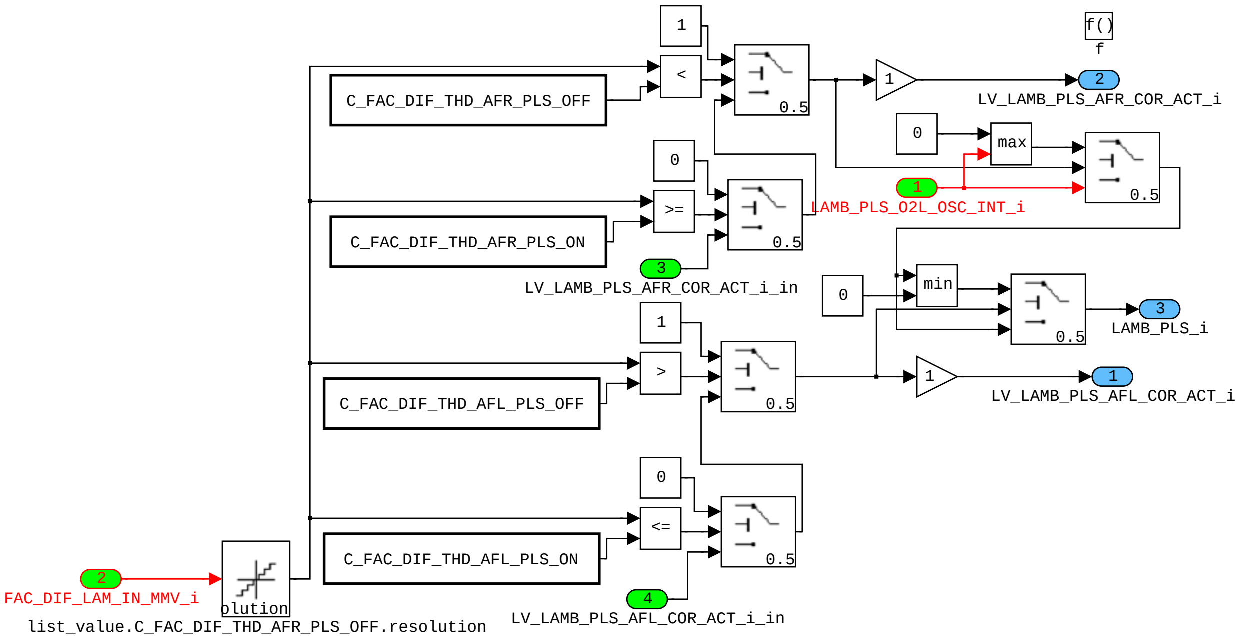


Figure 6 EGTR_DEFSPFSTIC0/ operate/ EGTR_DEFSPFSTIC0_CBK_MNG/ EGTR_DEFSPFSTIC0_CBK_SPC/ COR_LAMB_PLS

18.2.1.2 SUBFUNCTION: reset

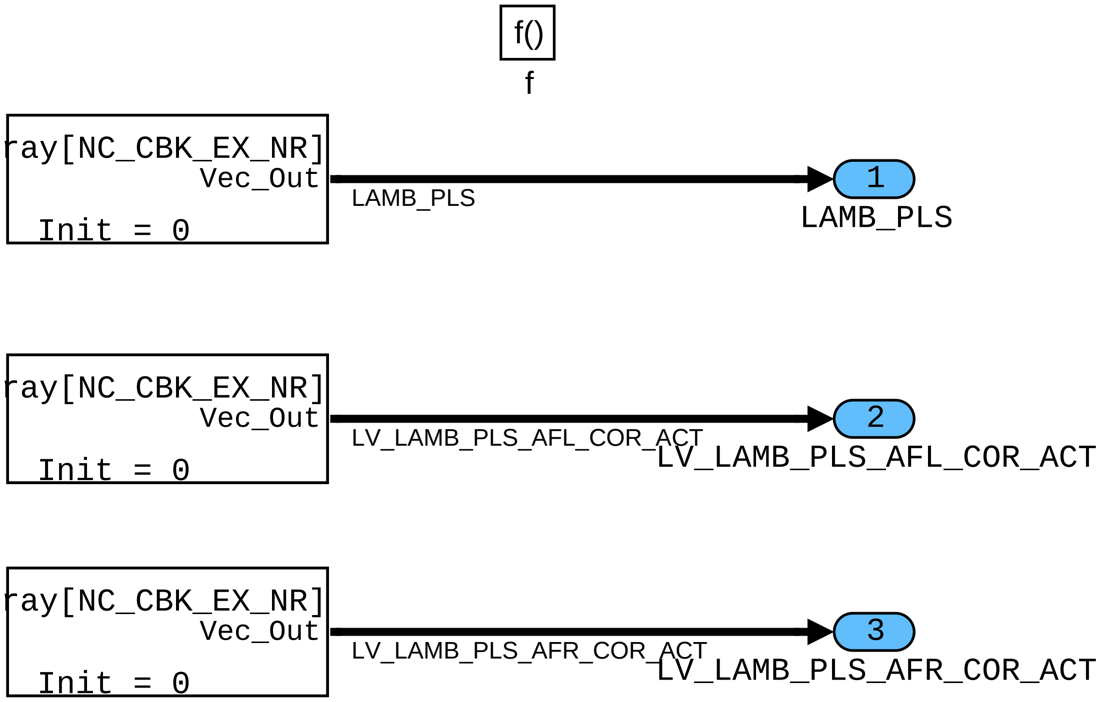


Figure 7 EGTR_DEFSPFSTIC0/ reset

18.3 Forced lambda stimulation for linear lambda control

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
FAC_O2L_LAMB_PLS [NC_CBK_EX_NR]	V	8000... 7FFFH	-0.25... 0.2499924	7.6294e-6	[-]
temporary factor for O2 loading calculation considering the lambda set point deviation					
LAMB_PLS_O2L_OSC [NC_CBK_EX_NR]	O/V	F800... 7FFH	-0.125... 0.124939	61.352e-6	[-]
lambda set point deviation based on forced stimulation O2 balancing					
LV_LAMB_PLS_MST_CBK [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
auxiliary flag indicating that this bank is master bank for forced stimulation					
LV_LAMB_PLS_O2L_OSC_POS [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
auxiliary flag indicating positive lambda stimulation half wave					
LV_LAMB_PLS_SYN_CBK	O/V	0... 1H	0... 1	1	[-]
flag used for exhaust gas bank synchonization					
O2L_ADD_LAMB_PLS [NC_CBK_EX_NR]	V	80000000... 7FFFFFFFH	-669.9236... 669.9236	311.96e-9	[g]
O2 load to be added to the O2 load integral each sample step					
O2L_LAMB_PLS [NC_CBK_EX_NR]	V	80000000... 7FFFFFFFH	-669.9236... 669.9236	311.96e-9	[g]
current O2 loading or release for forced lambda stimulation					
O2L_SP_HALF_LAMB_PLS [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.616849	39.306e-6	[g]
auxiliary value; half O2 loading set point of forced lambda stimulation					
O2L_SP_NEW_LAMB_PLS [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.616849	39.306e-6	[g]
O2 loading set point that is realized by forced lambda stimulation					
STATE_LAMB_PLS [NC_CBK_EX_NR]	O/V	0 1 2	OFF AFL AFR	-	[-]
state of forced lambda stimulation					

Input Data:

LAMB_DELTA_MAX_PLS	LAMB_SP_DE_PLS	LV_LAMB_PLS_ACT [NC_CBK_EX_NR]	LV_ST_END
MAF_CYL	NC_CBK_EX_NR	O2L_SP_PLS	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_STATE_LAMB_PLS_O2L_OSC_COR	1	0... 2H	0... 2	1	[-]
defines operation mode of forced stimulation: 0 - amplitude symmetric, 1 - timesync COR_AFL, 2 - timesync COR_AFR					
LC_O2L_PLS_CBK_SYN	1	0... 1H	0... 1	1	[-]
0: asynchronous forced stimulation, 1: synchronous forced stimulation					

General Information

The forced lambda stimulation generates an output signal that causes lean and rich deviations of the air fuel ratio from the lambda controller set point. By means of the stimulation parameters (set point deviation, period time and gradient) a well defined O2 load for the catalyst can be calibrated. So the catalyst efficiency can be increased.

Furthermore the forced lambda stimulation is used by diagnosis functions for the up stream lambda sensor and by the catalyst efficiency diagnosis.
The recurrence time is defined by T_SAMPLE in 20 ms.
NC_CBK_EX_NR defines the number of exhaust banks.
For vector elements, the variable extension _i is used in the model instead of [i] as found in the textual description.

Application Conditions

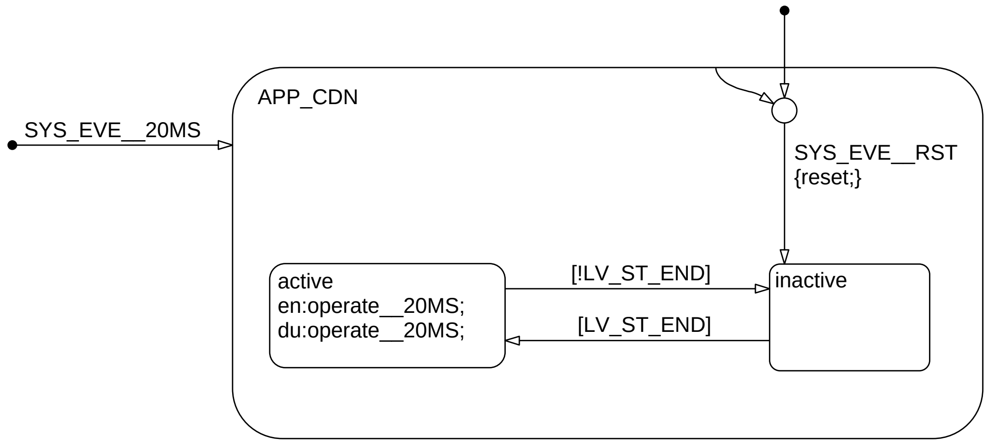


Figure 8:
Path: EGTR_DEFSPFSTI0/APP_CDN/Chart

Function description

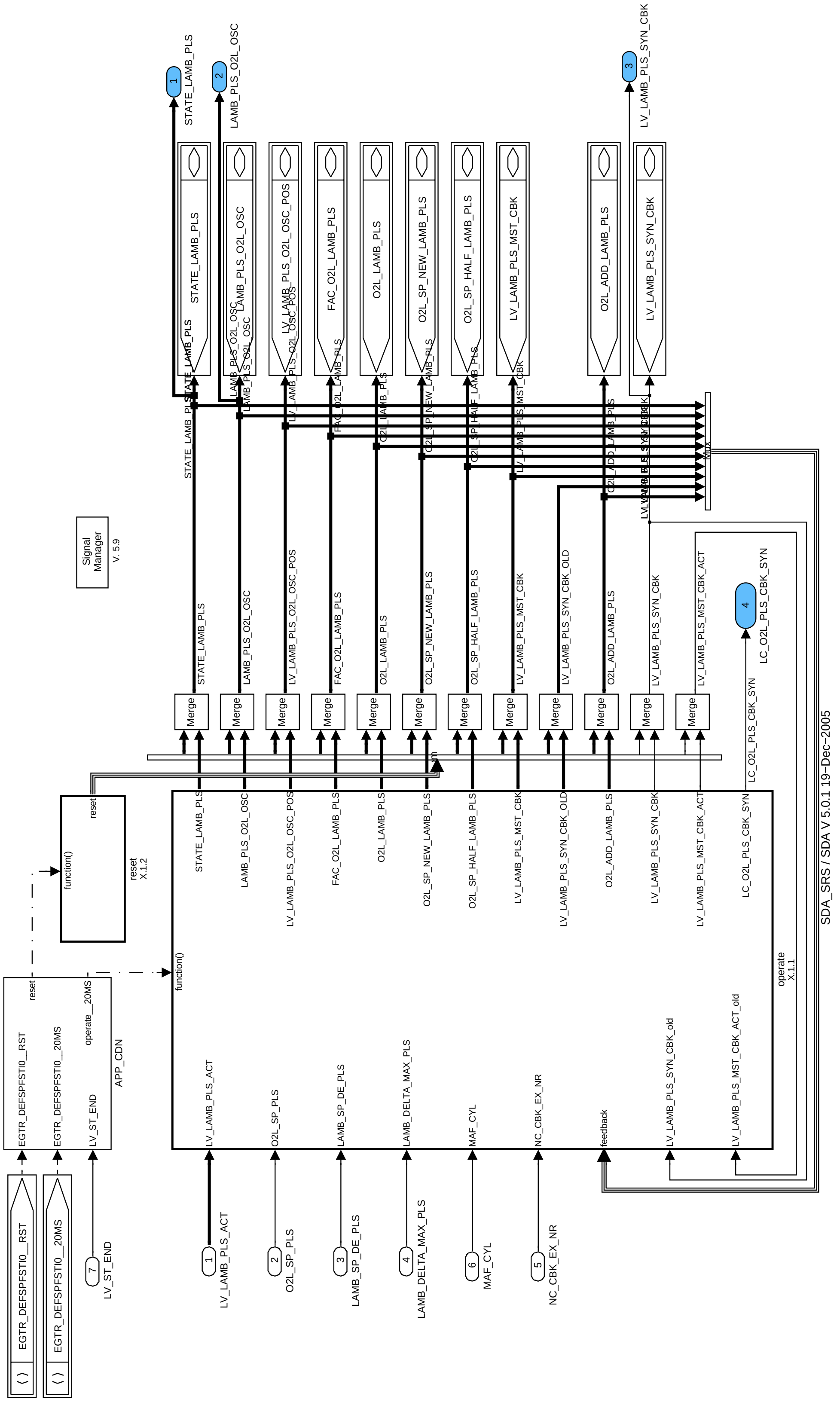


Figure 9:
Path: EGTR_DEFSPFSTI0

18.3.1 Operate level 20ms:

This level is calculated in 20ms tasks.

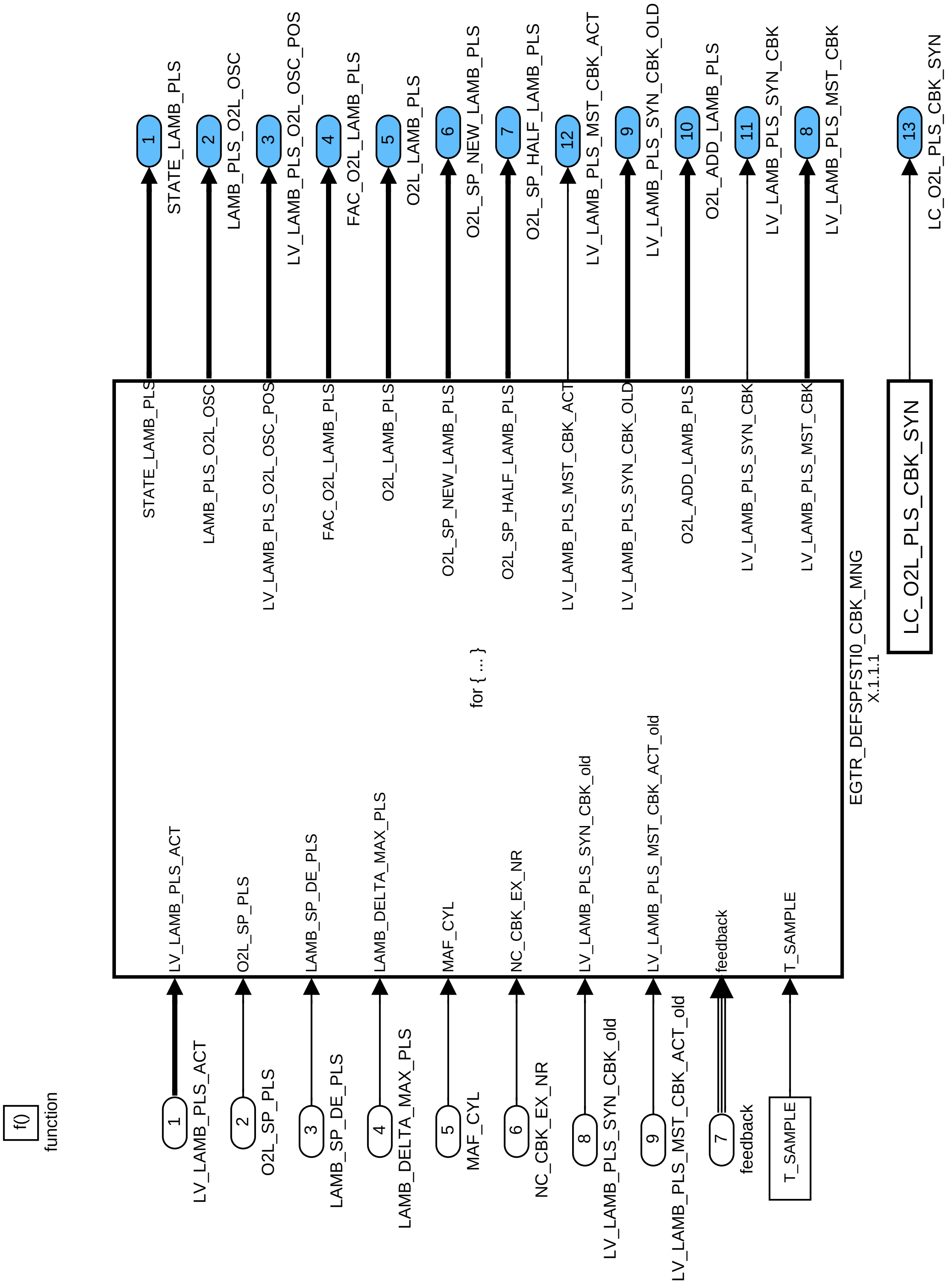


Figure 10:
Path: EGTR_DEFSPFSTI0/operate

18.3.1.1 Vectorization level for for-loop:

Here the vectors are split for the for-loop-calculation.

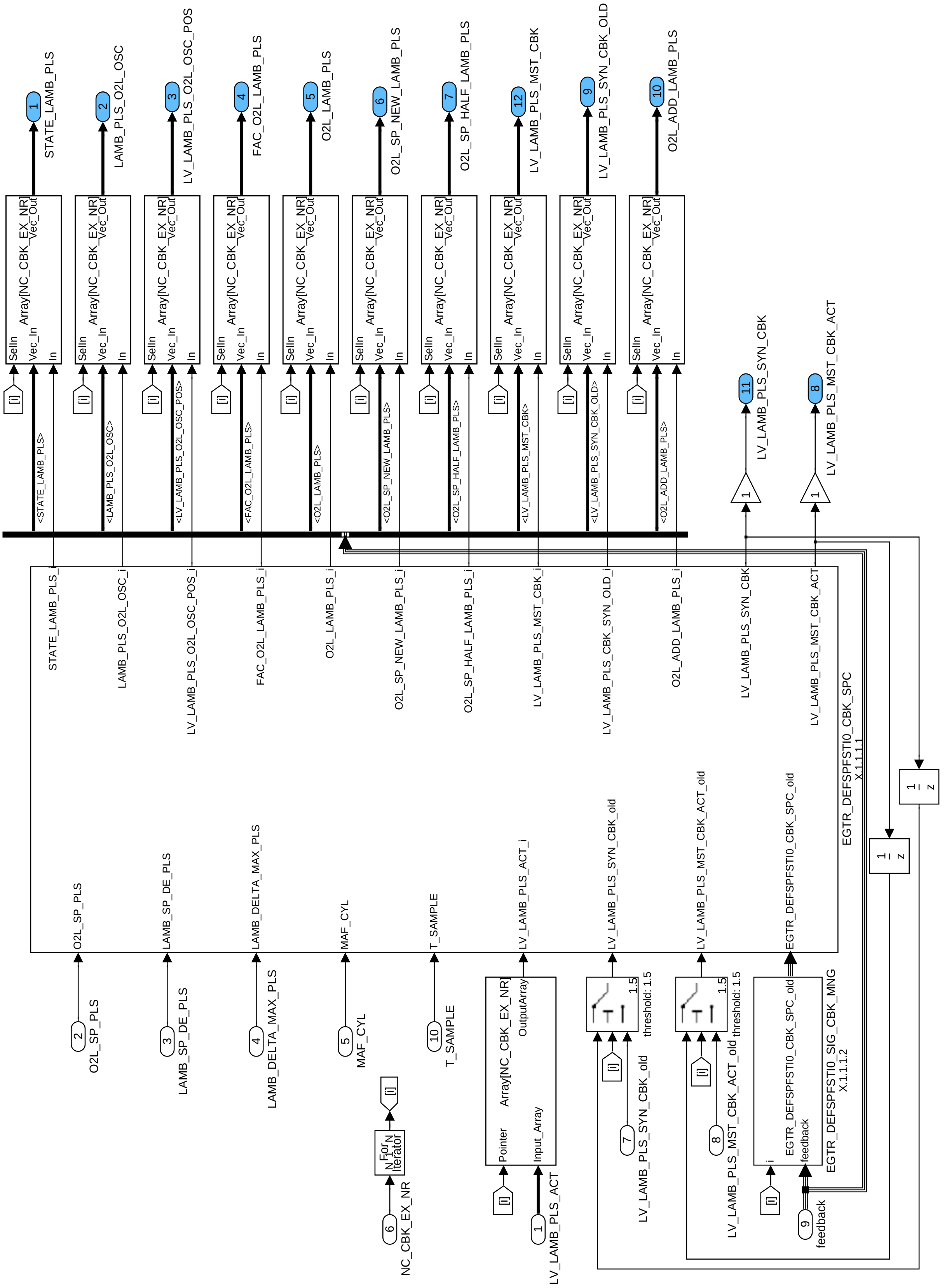


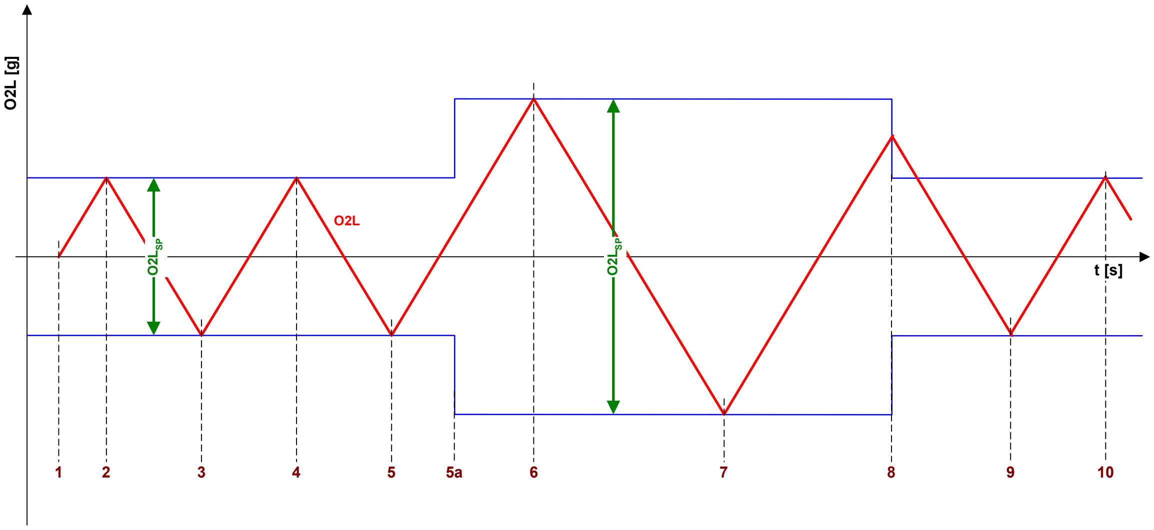
Figure 11:
Path: EGTR_DEFSPFSTI0/operate/EGTR_DEFSPFSTI0_CBK_MNG

18.3.1.1.1 Execution of state machine:

In state OFF the output of the forced lambda stimulation is 0. This state is active if LV_LAMB_PLS_ACT[i] is not set.
In case of LV_LAMB_PLS_ACT[i] = 1 either state AFL or state AFR is active. In state AFL the set point deviation is increased stepwise to the calibrated set point deviation. In state AFR the set point deviation is decreased. The transition between both states depend on the O2 loading calculated in both states (CLC_O2L_LAMB_PLS). When the current O2 loading reaches the current set point (O2L_SP_NEW_LAMB_PLS[i]) then the transition to the opposite state takes place. The O2 loading is calculated with the formula:

$$m_{O_2} = 23\% \cdot \int \left(1 - \tfrac{1}{1+\Delta\lambda_{SP}}\right) \cdot \dot{m}_{air} dt$$

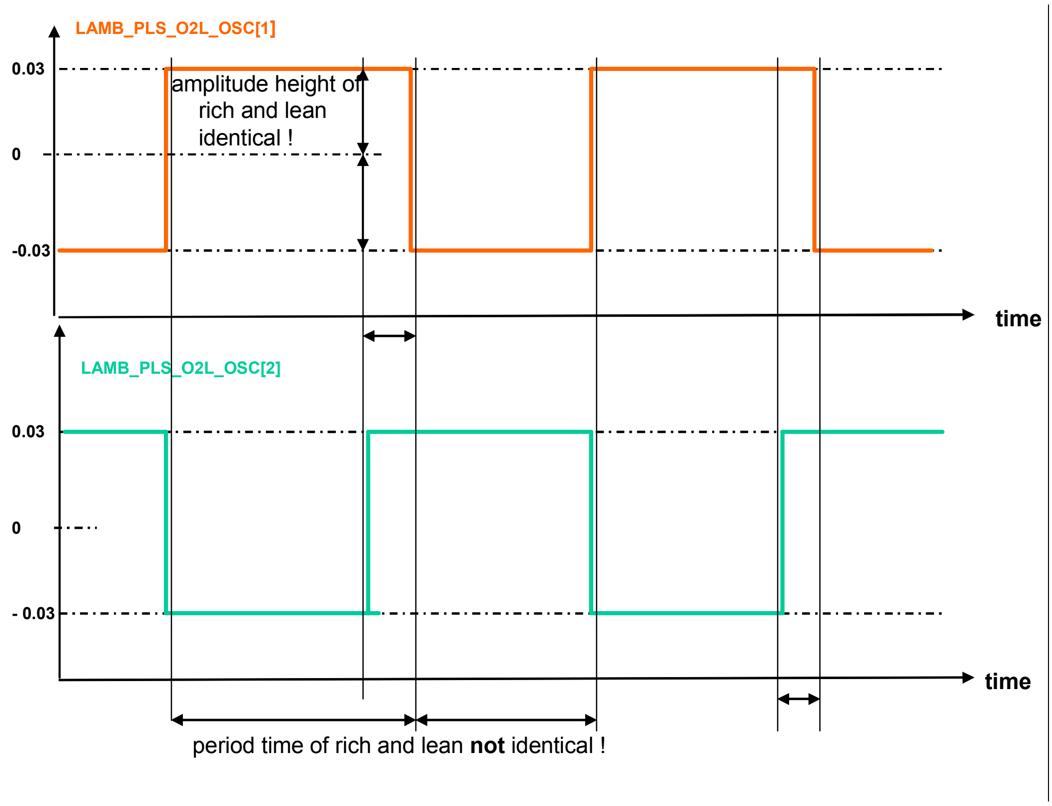
In order to realise synchronisation of both banks two flags are defined.
LV_LAMB_PLS_MST_CBK_ACT is a common flag that defines whether forced stimulation is already active for one bank. Master bank is the bank that has set the flag LV_LAMB_PLS_MST_CBK_ACT; this is also indicated by the bank selective flag LV_LAMB_PLS_MST_CBK[i] which must be set to 1 for the master bank and to 0 for the slave bank.
The second flag to synchronise both banks is LV_LAMB_PLS_SYN_CBK. This flag is set by the master when the transition from state AFR to AFL occurs in order to inform the slave to execute the desired transition.



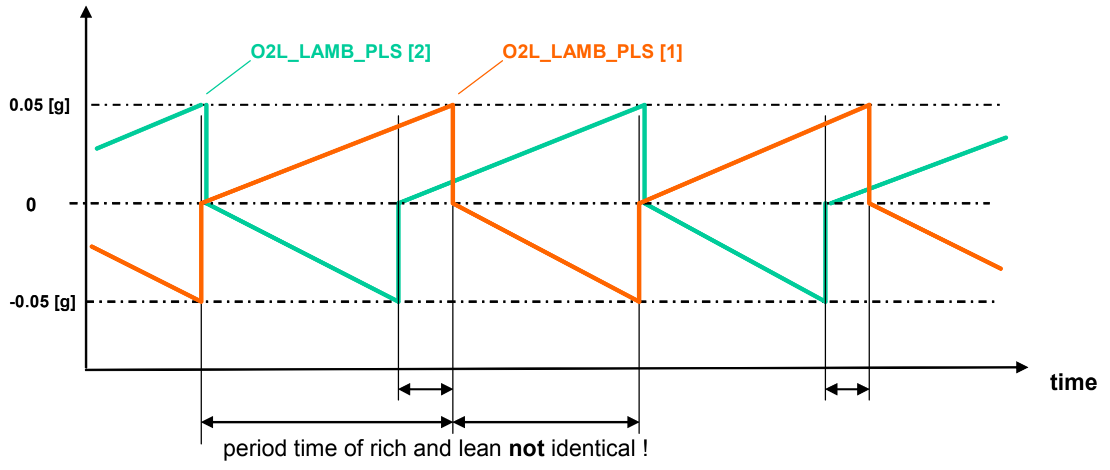
If the set point from Determination of forced lambda stimuation parameters (O2L_SP_PLS) changes the O2 loading set point that is adjusted switches for the next half wave to an intermedieate value and the next but one half wave will be the target value.
Asynchronous or synchronous operation:
The master bank changes from lean to rich or from rich to lean, if the O2 threshold O2L_SP_NEW_PLS is reached. The synchronization pulse LV_LAMB_PLS_SYN_CBK is set at the end of the rich phase.
In case of asynchronous forced stimulation (LC_O2L_PLS_CBK_SYN=0) the slave changes from lean to rich at synchronization pulse and from rich to lean, if the O2 threshold is reached.
In case of synchronous forced stimulation (LC_O2L_PLS_CBK_SYN=1) the slave changes from rich to lean at synchronization pulse and from lean to rich, if the O2 threshold is reached.
Timesynchronous operation:

general specification

Due to the formula for O2 load calculation the time duration of the lean part of the forced stimulation period is slightly longer than the rich part.
During the bankwise asynchronous operation mode (LC_O2L_PLS_CBK_SYN=0), this fact leads to small overlapping phases, where both banks are in the same STATE_LAMB_PLS = RICH or LEAN.

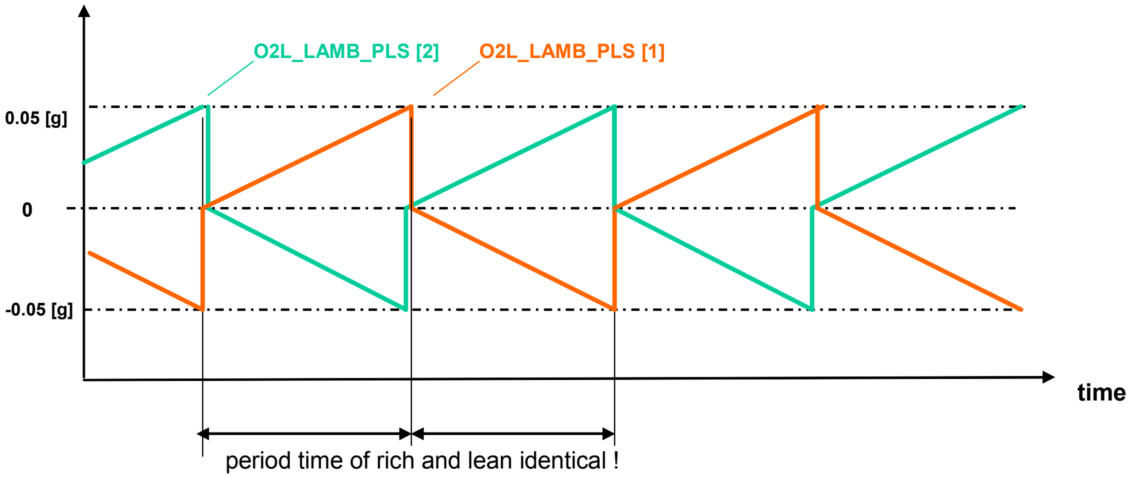


O2 load calculation



To prevent this behaviour and achieve a duration of the rich and lean phase of exact the same length, the amplitude of the forced stimulation LAMB_PLS_O2L_OSC can be corrected during the lean or during the rich phase (STATE_LAMB_PLS = AFL or AFR).

O2 load calculation



COR_AFL:
To switch on the timesynchronous operation with correction of the amplitude in STATE_LAMB_PLS=AFL set C_STATE_LAMB_PLS_O2L_OSC_COR = 1 (COR_AFL) .
As a consequence the amplitude LAMB_PLS_O2L_OSC is no more symmetric on the lean and rich side.
LAMB_PLS_O2L_OSC is limited to the calibrated value on the rich side and will be corrected to a slightly higher value on the lean side.
The correction factor on the lean side is calculated as follows:

$$m_{O_2}\_lean = -m_{O_2}\_rich$$

$$\frac{LAMB\_PLS\_O2L\_OSC(lean)}{1+LAMB\_PLS\_O2L\_OSC(lean)} = -\frac{LAMB\_PLS\_O2L\_OSC(rich)}{1+LAMB\_PLS\_O2L\_OSC(rich)}$$

$$LAMB\_PLS\_O2L\_OSC(lean) = -\frac{LAMB\_PLS\_O2L\_OSC(rich)}{1+2*LAMB\_PLS\_O2L\_OSC(rich)}$$

COR_AFR :
To switch on the timesynchronous operation with correction of the amplitude in STATE_LAMB_PLS=AFR set C_STATE_LAMB_PLS_O2L_OSC_COR = 2 (COR_AFR).
LAMB_PLS_O2L_OSC is then limited to the calibrated value on the lean side and will be corrected to a slighly lower value on the rich side.
The correction factor is on the rich side calculated as follows:

$$LAMB\_PLS\_O2L\_OSC(rich) = -\frac{LAMB\_PLS\_O2L\_OSC(lean)}{1+2*LAMB\_PLS\_O2L\_OSC(lean)}$$

COR_OFF:
If timesynchronous operation is not requested, it is possible to enable normal operation mode with symmetric amplitude hight by setting C_STATE_LAMB_PLS_O2L_OSC_COR = 0 (COR_OFF).

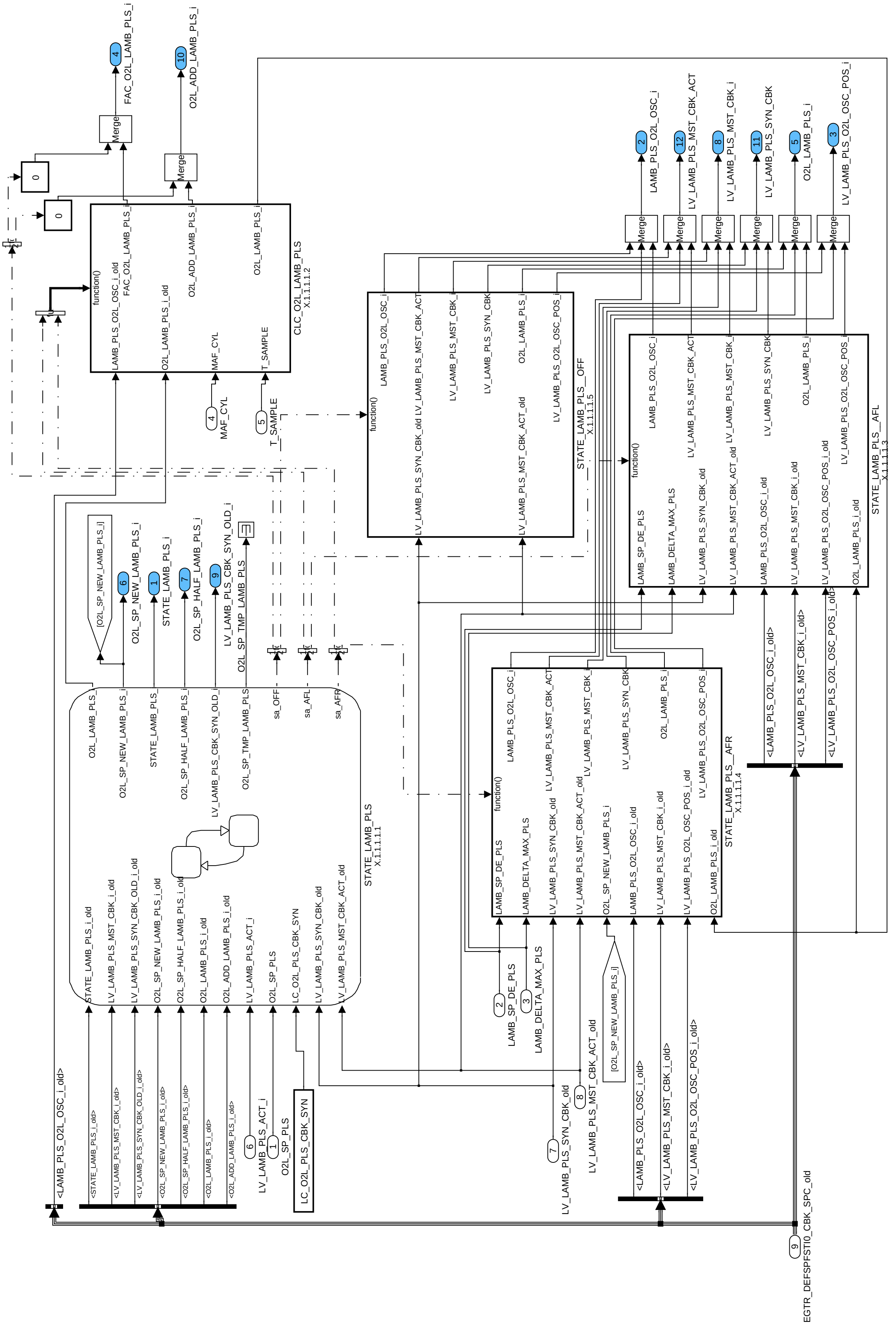


Figure 12:
Path: EGTR_DEFSPFSTI0/operate/EGTR_DEFSPFSTI0_CBK_MNG/EGTR_DEFSPFSTI0_CBK_SPC

18.3.1.1.1.1 No title given

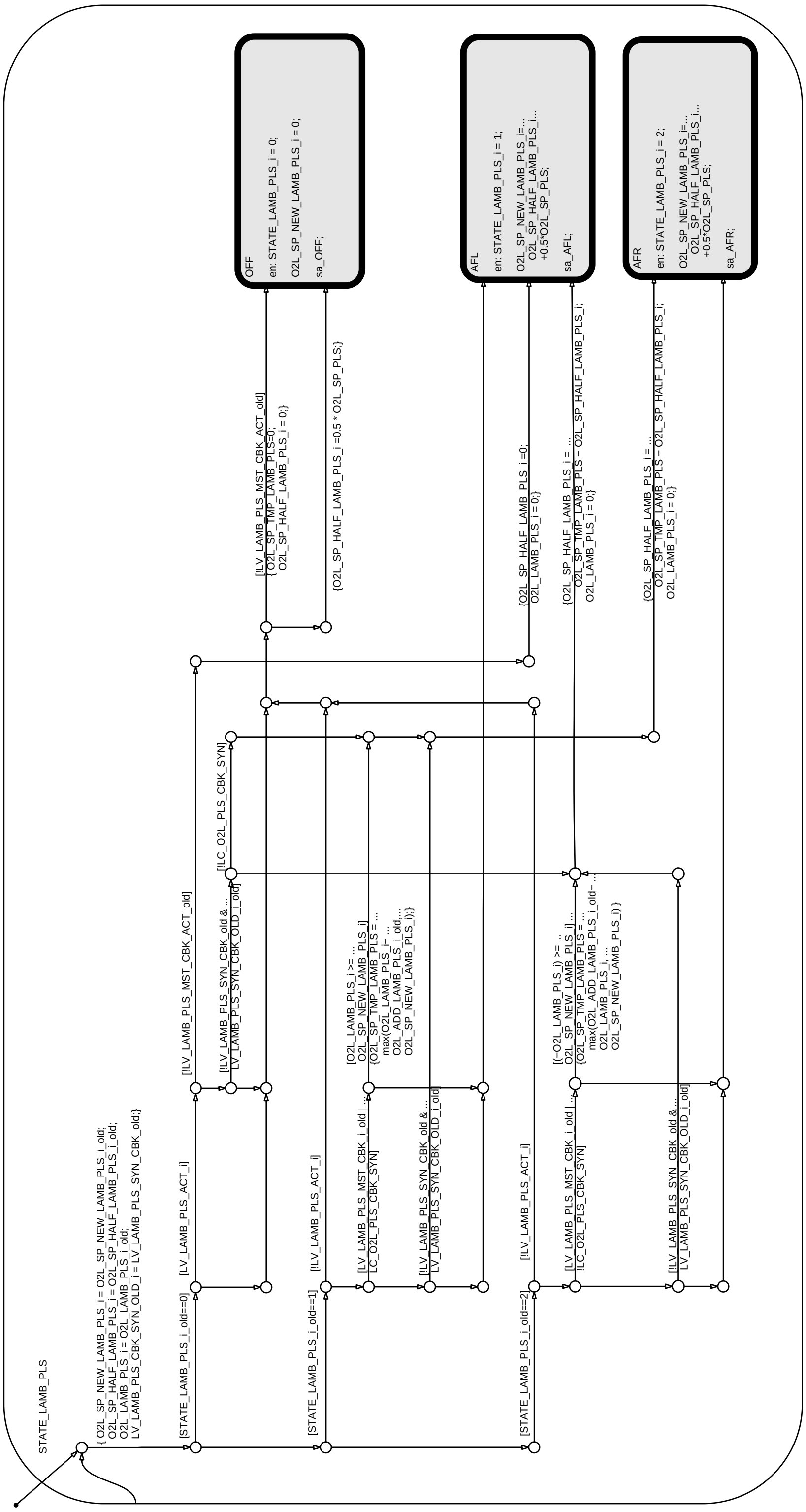


Figure 13:
Path: EGTR_DEFSPFSTI0/operate/EGTR_DEFSPFSTI0_CBK_MNG/EGTR_DEFSPFSTI0_CBK_SPC/STATE_LAMB_PLS

18.3.1.1.1.2 Calculation of O2 loading:

The O2 loading is calulated with the already mentioned formula. If the O2 loading that is to added each sample step is smaller than its resolution then the O2 loading is increased or decreased (depending on the sign) by the resolution value.

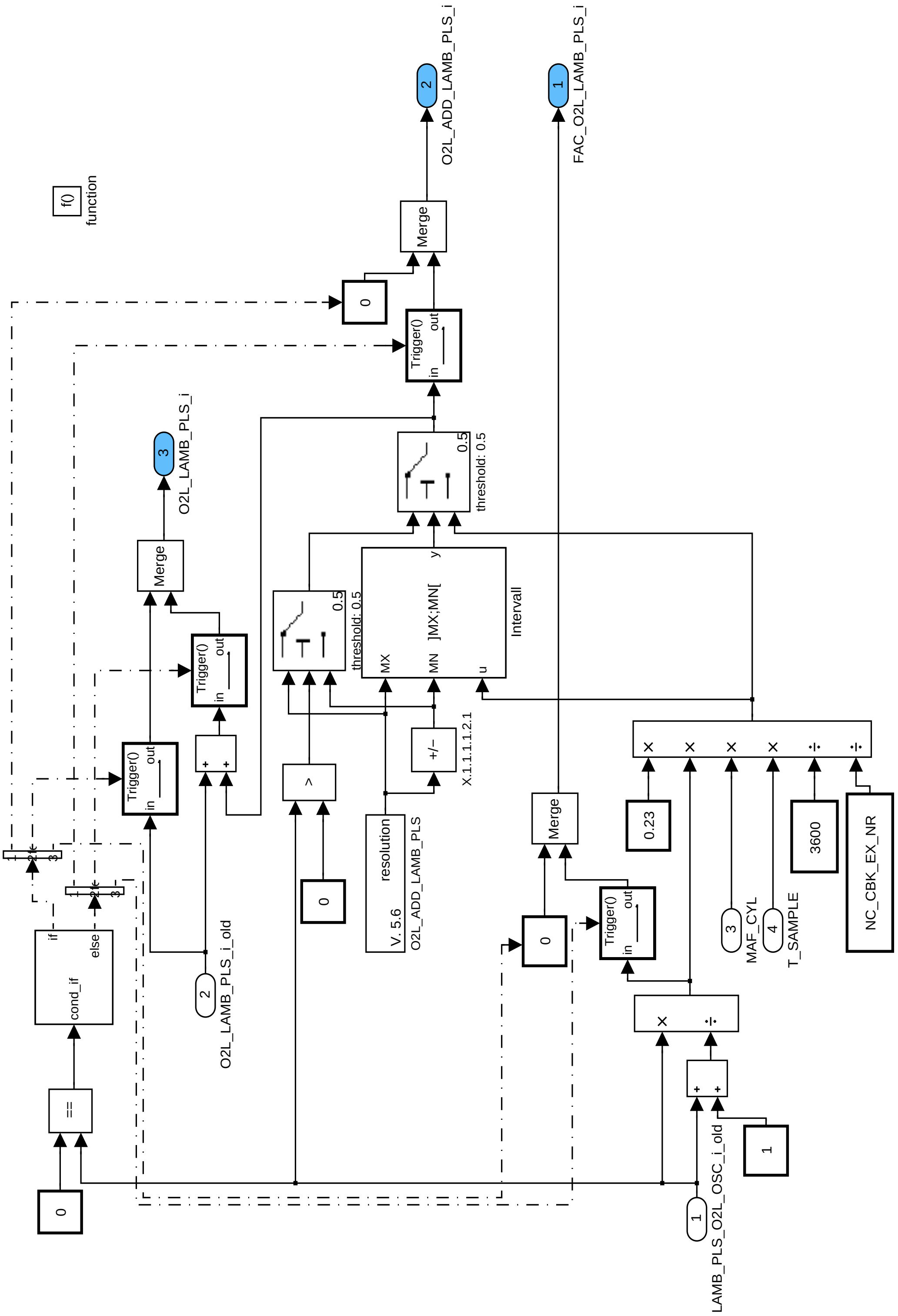


Figure 14:
Path: EGTR_DEFSPFSTI0/operate/EGTR_DEFSPFSTI0_CBK_MNG/EGTR_DEFSPFSTI0_CBK_SPC/CLC_O2L_LAMB_PLS

18.3.1.1.1.3 STATE "AFL":

In that state the lambda set point deviation LAMB_PLS_O2L_OSC[i] is increased step wise by LAMB_DELTA_MAX_PLS and limited to LAMB_SP_DE_PLS.
If timesynchronous operation mode is activated with C_STATE_LAMB_PLS_O2L_OSC_COR = 1, the lambda set point deviation is not symmetric. The rich half wave is limited to LAMB_SP_DE_PLS, but the lean half wave is corrected to a slightly higher value.
If the LAMB_PLS_O2L_OSC[i] becomes positive (indicated by rising edge of LV_LAMB_PLS_O2L_POS[i]) the sign of the current O2 loading changes in order adjust the right O2 loading (given by the set point).

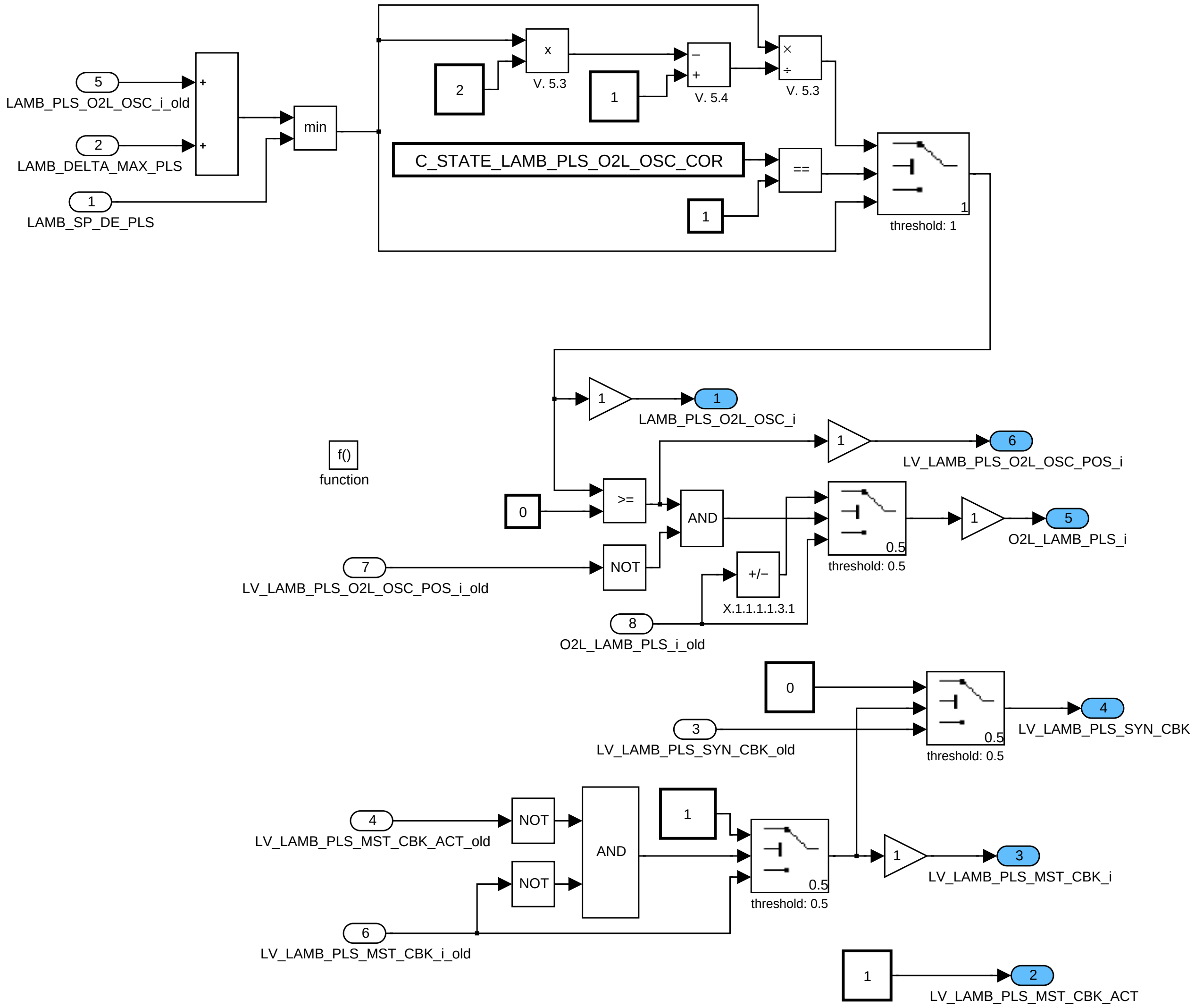


Figure 15:
Path: EGTR_DEFSPFSTI0/operate/EGTR_DEFSPFSTI0_CBK_MNG/EGTR_DEFSPFSTI0_CBK_SPC/STATE_LAMB_PLS__AFL

18.3.1.1.1.4 STATE "AFR":

In that state the lambda set point deviation LAMB_PLS_O2L_OSC[i] is decreased step wise by LAMB_DELTA_MAX_PLS and limited to minus LAMB_SP_DE_PLS.
If timesynchronous operation mode is activated with C_STATE_LAMB_PLS_O2L_OSC_COR = 2, the lambda set point deviation is not symmetric. The lean half wave is limited to LAMB_SP_DE_PLS, but the rich half wave is corrected to a slightly lower value.
If the LAMB_PLS_O2L_OSC[i] becomes negative (indicated by falling edge of LV_LAMB_PLS_O2L_POS[i]) the sign of the current O2 loading changes as in state AFL .
When the O2 loading set point is reached the master bank sets the synchronization flag.

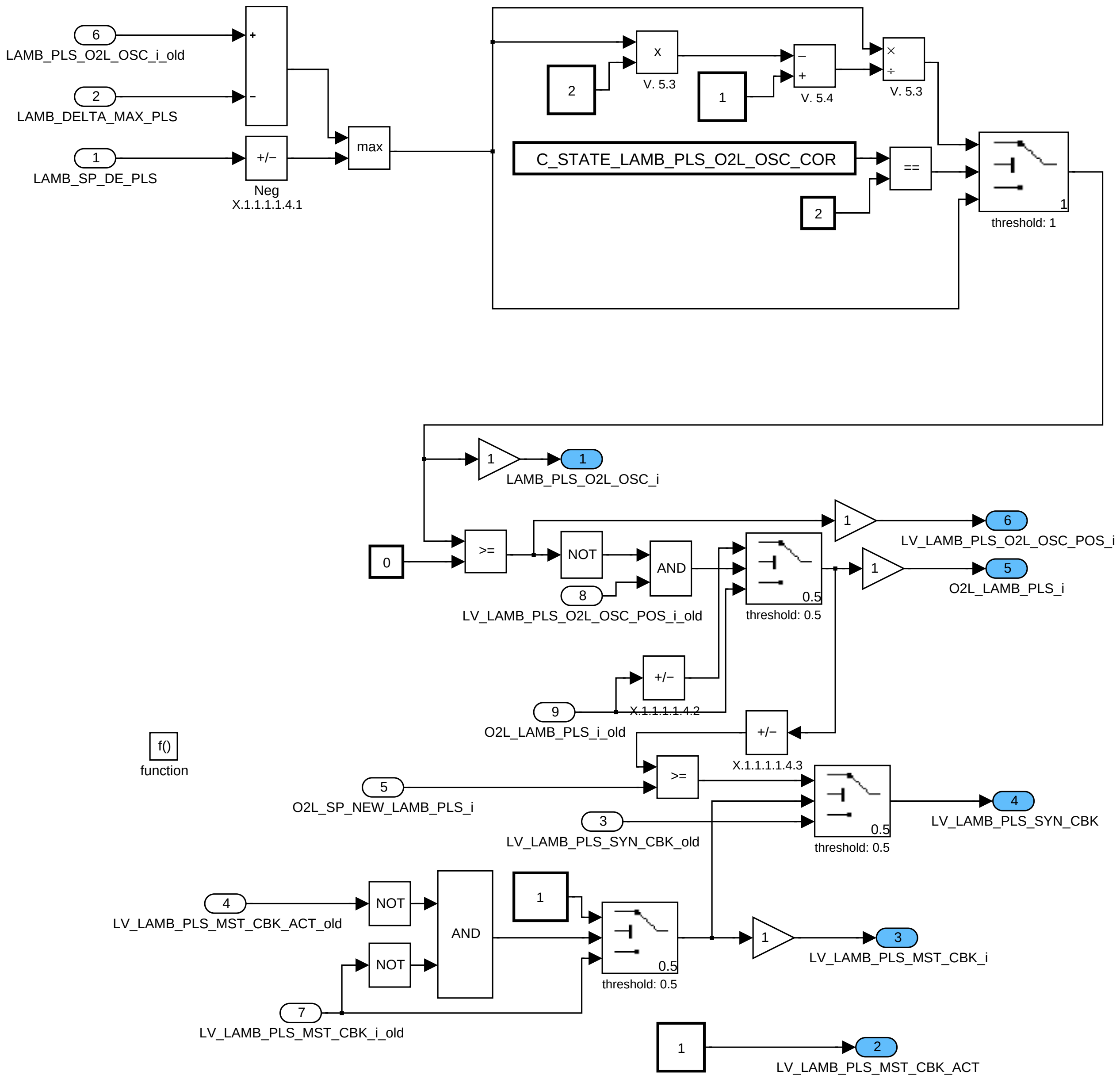


Figure 16:
Path: EGTR_DEFSPFSTI0/operate/EGTR_DEFSPFSTI0_CBK_MNG/EGTR_DEFSPFSTI0_CBK_SPC/STATE_LAMB_PLS__AFR

18.3.1.1.1.5 STATE "OFF":

In that state all flags and the current O2 loading are set to 0.

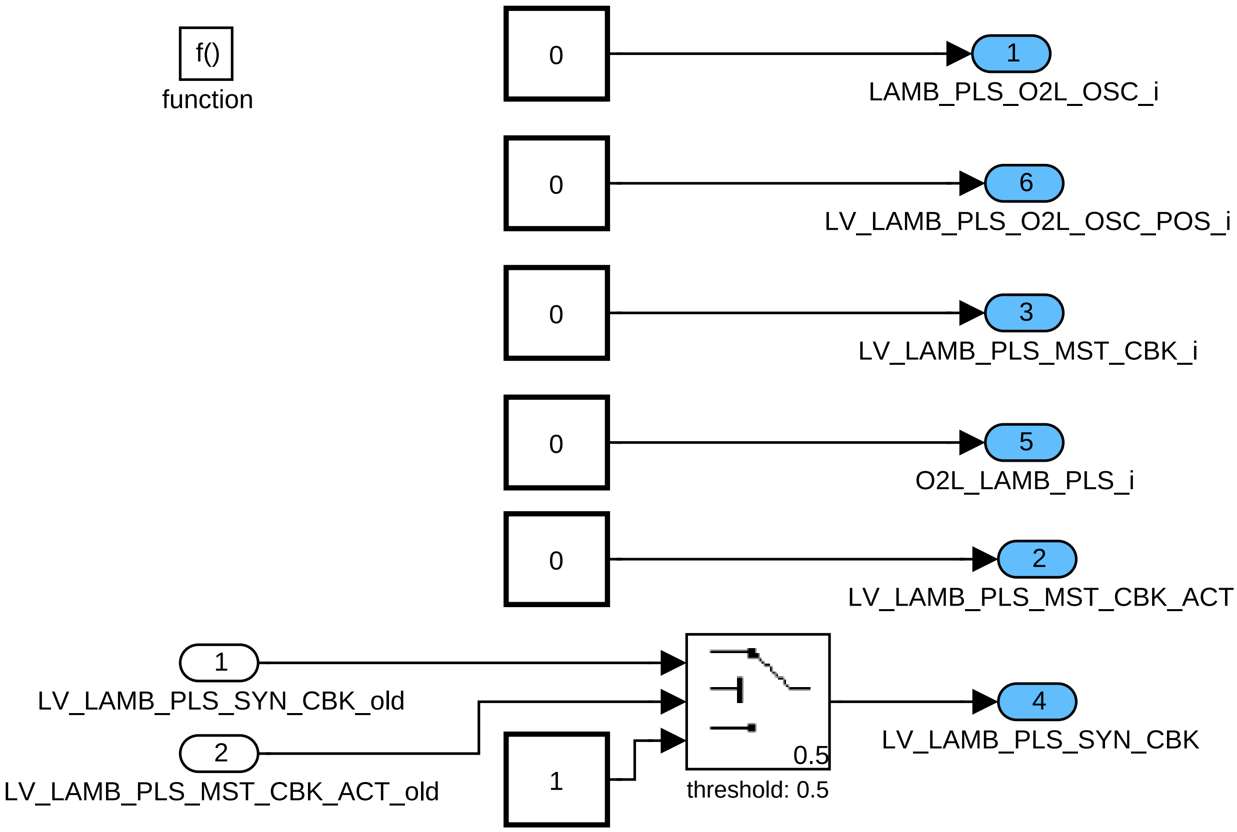


Figure 17:
Path: EGTR_DEFSPFSTI0/operate/EGTR_DEFSPFSTI0_CBK_MNG/EGTR_DEFSPFSTI0_CBK_SPC/STATE_LAMB_PLS__OFF

18.3.1.1.2 For-loop signal manager:

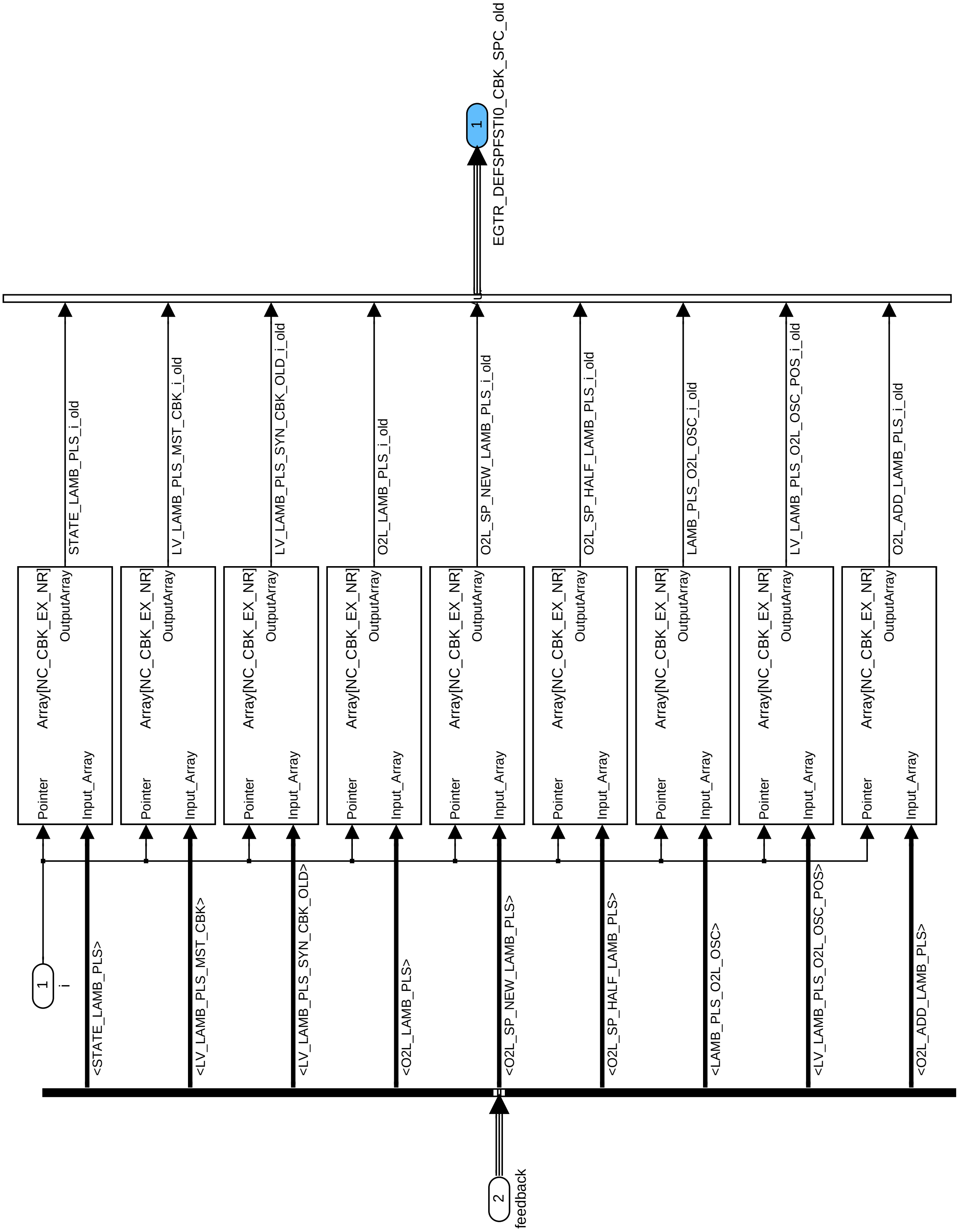


Figure 18:
Path: EGTR_DEFSPFSTI0/operate/EGTR_DEFSPFSTI0_CBK_MNG/EGTR_DEFSPFSTI0_SIG_CBK_MNG

18.3.2 Initialization:

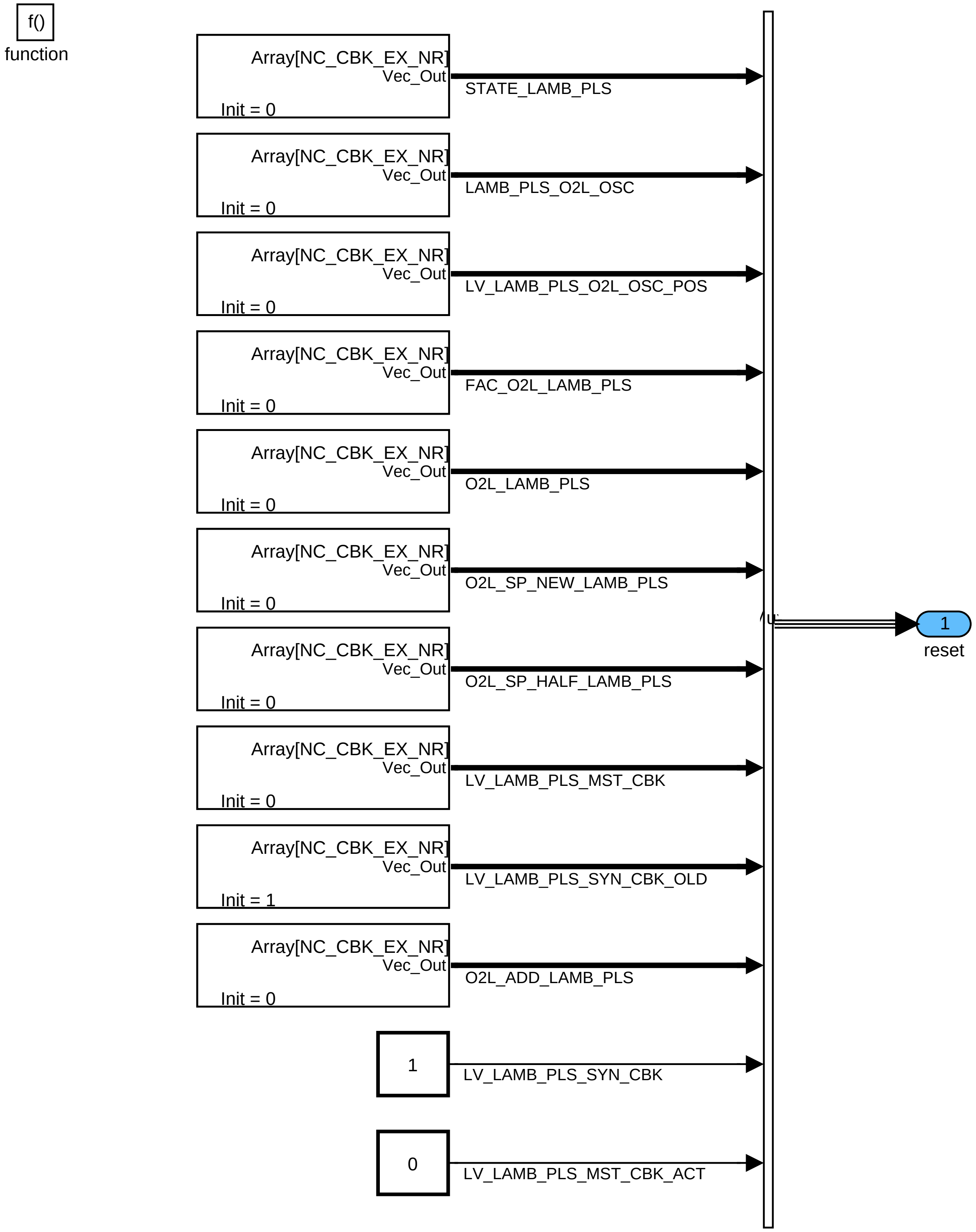


Figure 19:
Path: EGTR_DEFSPFSTI0/reset

18.4 Application Incidences for forced lambda stimulation

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LAMB_PLS_EXT [NC_CBK_EX_NR]	O/V	F800... 7FFH	-0.125... 0.12493896484	61.035e-6	[-]
lambda set point deviation from external source					
LV_INH_LAMB_PLS [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
inhibition flag for forced lambda stimulation					

Input Data:

LAMB_PLS_EXT_CUS [NC_CBK_EX_NR]	LV_LAMB_STI_OFF_LSL_OFS_ADJ [NC_CBK_EX_NR]	LV_ST_END	NC_CBK_EX_NR

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
LC_LAMB_PLS_INH [NC_CBK_EX_NR]	1	0... 1H	0... 1	1	[-]
calibration flag to force lambda stimulation to be inhibited					
LC_LAMB_STI_OFF_LSL_OFS_ADJ	1	0... 1H	0... 1	1	[-]
calibration flag to allow inhibition of lambda stimulation during offset adjustment					

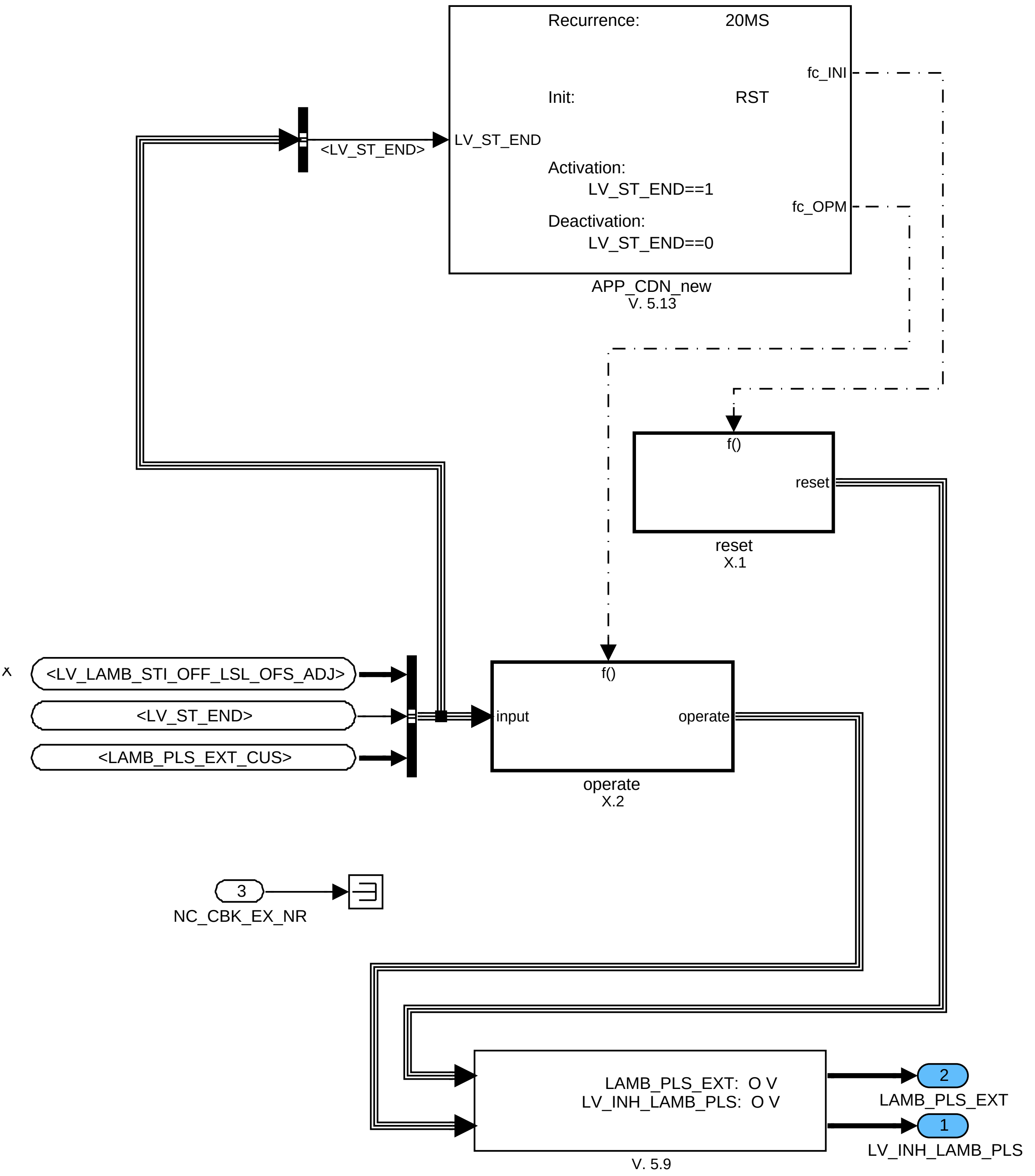
General Information

NC_CBK_EX_NR defines the number of exhaust banks.
For vector elements, the variable extension _i is used in the model instead of [i] as found in the textual description.

Application Conditions

Initialization: RST
Recurrence: 20MS
Activation: LV_ST_END==1
Deactivation: LV_ST_END==0

Function description



SDA_SRS / SDA V 5.0.2.2 25−Jul−2006

Figure 20:

18.4.1 Initialization

18.4.1.1 Calculation of Initialization

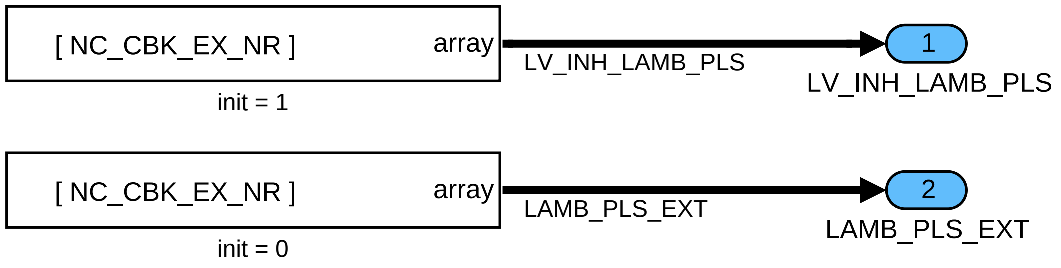


Figure 21:

18.4.2 Operate

During WRAF sensor controller offset adjustment the forced lambda stimulation is switched off if the respective calibration flag allows this.
Furthermore the general calibration flag LC_LAMB_PLS_INH[i] inhibits the function if set.

18.4.2.1 Calculation of Operate

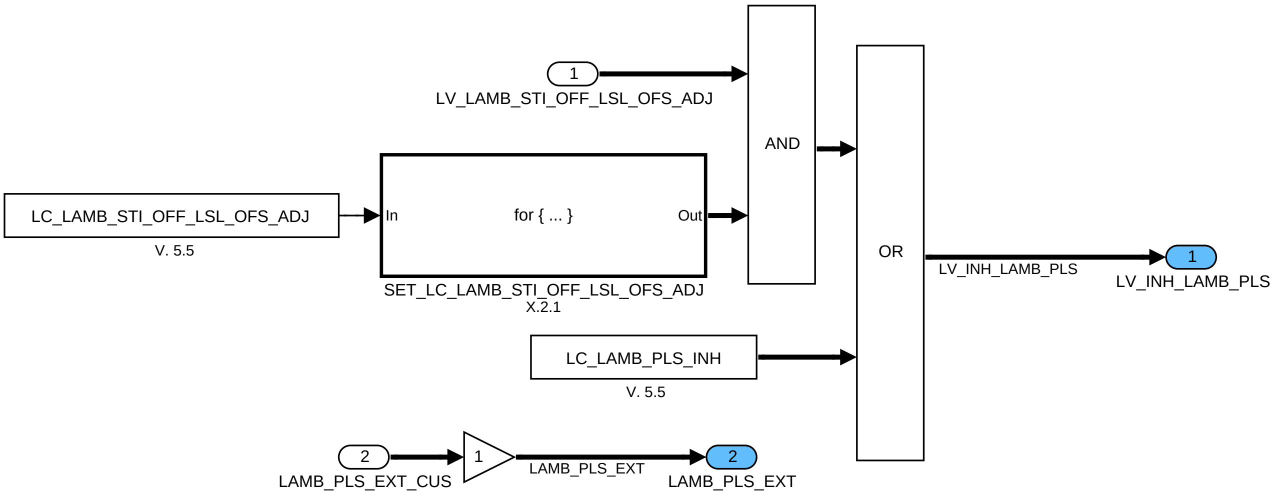


Figure 22:

18.4.2.1.1 No title given

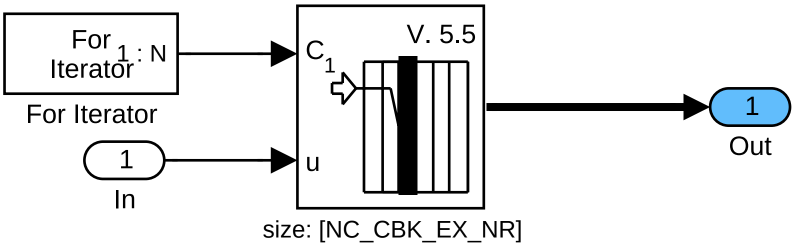


Figure 23:

18.5 Oxygen loading determination for catalyst enrichment function

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CAT_PURGE_ST[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating that cat purge can be started (set for one sample step)					
O2L_RLS_SP_PCAT_PURGE[NC_CBK_EX_NR]	O/V	0...FFFFH	0...2.61684896	3.99306E-5	g
oxygen loading to be purged (released) during pre cat purge					
FAC_ADD_MAIN_CAT_PURGE[NC_CBK_EX_NR]	O/V	0...FFH	0...1.9921875	0.0078125	-
additional weighting factor for pre and main cat differences					
LV_CAT_PURGE_POST_PUC_TRIG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
auxiliary flag indicating that cat purge after PUC was triggered					
STATE_CAT_PURGE[NC_CBK_EX_NR]	V	0H 1H 2H 3H 4H 5H	CAT_PURGE_OFF CAT_PURGE_PUC CAT_PURGE_SA CAT_PURGE_HST CAT_PURGE_CH CAT_PURGE_GS_SCC	1	-
state indicating the activation reason of cat purge					
LV_LAMB_SP_CH_ACT_CAT_PURGE[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicates that LAMB_SP_CH was under threshold					
LV_GS_SCC_OLD	-	0...1H	0...1	1	-
old value for cylinder cut of during gear shift					
LV_PUC_OLD[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
auxiliary flag for edge detection of LV_PUC					
LV_SAWUP_OLD[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
auxiliary flag for edge detection of LV_SAWUP					
LV_ST_END_OLD[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
auxiliary flag for edge detection of LV_ST_END					

Input data:

LV_INH_CAT_PURGE[NC_CBK_EX_NR]	LV_CAT_PURGE_ACT[NC_CBK_EX_NR]	O2L_RLS_PCAT_PURGE[NC_CBK_EX_NR]	EFF_CAT_DIAG[NC_CBK_EX_NR]
NC_CBK_EX_NR	LV_FCUT_IND	TCO_ST	LV_PUC
TEMP_CAT_DYN_MDL[NC_CBK_EX_NR]	MAF_INT_PUC_ACT	LV_SAWUP	LV_ST_END
LAMB_SP_CH[NC_CBK_EX_NR]	MAF_INT_GS_SCC	LV_GS_SCC	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_ADD_MAIN_CAT_PURGE_CH[NC_CBK_EX_NR]	1	0...FFH	0...1.9921875	0.0078125	-
additional weighting factor to define oxygen loading for main cat purge after catalyst heating phase					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_ADD_MAIN_CAT_PURGE_GS_SCC[NC_CBK_EX_NR]	1	0...FFH	0...1.9921875	0.0078125	-
additional weighting factor for pre and main cat differences in case of gear shift with cylinder cut off					
C_FAC_ADD_MAIN_CAT_PURGE_HST[NC_CBK_EX_NR]	1	0...FFH	0...1.9921875	0.0078125	-
additional weighting factor to define oxygen loading for main cat purge after hot start					
C_FAC_ADD_MAIN_CAT_PURGE_SA[NC_CBK_EX_NR]	1	0...FFH	0...1.9921875	0.0078125	-
additional weighting factor to define oxygen loading for main cat purge after secondary air injection					
C_LAMB_SP_CH_THD_CAT_PURGE	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Threshold for lambda setpoint out of catalyst heating to activate catalyst purge					
C_MAF_INT_GS_SCC_MIN_CAT_PURGE	1	0...FFFFH	0...2.91267E+3	0.04444444	g
minimum air mass flow integral during cylinder cut-off at gear shift to activate cat purge function					
C_MAF_INT_PUC_ACT_MIN_CAT_PURGE	1	0...FFFFH	0...2.91267E+3	0.04444444	g
minimum air mass flow integral during PUC to activate cat purge function					
C_O2L_RLS_MAX_PCAT_PURGE[NC_CBK_EX_NR]	1	0...FFFFH	0...2.61684896	3.99306E-5	g
upper limit of oxygen loading set point for pre cat purge					
C_O2L_RLS_SP_PCAT_PURGE_CH[NC_CBK_EX_NR]	1	0...FFFFH	0...2.61684896	3.99306E-5	g
oxygen loading to be purged after catalyst heating lean phase					
C_O2L_RLS_SP_PCAT_PURGE_HST[NC_CBK_EX_NR]	1	0...FFFFH	0...2.61684896	3.99306E-5	g
oxygen loading to be purged after hot start					
C_O2L_RLS_SP_PCAT_PURGE_SA[NC_CBK_EX_NR]	1	0...FFFFH	0...2.61684896	3.99306E-5	g
oxygen loading to be purged after secondary air injection					
C_TCO_THD_CAT_PURGE_HST	1	0...FEH	-48...142.5	0.75	°C
temperature threshold for hot start detection of cat purge function					
IP_O2L_RLS_SP_PCAT_PURGE_PUC[NC_CBK_EX_NR]	6	0...FFFFH	0...2.61684896	3.99306E-5	g
LDP_MAF_INT_PUC_ACT_IP_O2L_RLS	6	0...FFFFH	0...2.91267E+3	0.04444444	g
oxygen loading to be purged after pull cut off phase					
IP_FAC_O2L_RLS_SP_PCAT_PURGE[NC_CBK_EX_NR]	6x6	0...FFH	0...0.99609375	0.00390625	-
LDP_TEMP_CAT_IP_FAC_PCAT_PURGE	6	0...7FFFH	-273.15 ... 1.77479E+3	0.0625	°C
LDPM_EFF_CAT_DIAG_1_EGTR	6	0...FFH	0...1.9921875	0.0078125	-
weighting factor for oxygen loading to be released during pre cat purge					

18.5.1 EGTR_DEFSPCENRD0

This function determines the oxygen loading for different catalyst enrichment (purge) cases that is to be relased during pre cat purge. The oxygen loading to be released during main cat purge is defined in the main function based on the oxygen loading for the pre catayst. Furthermore a priorization of the different cases is included.

The different enrichment cases after lean air fuel mixture condition, that are described within this function, are catalyst purge

- after trailing throttle fuel cut-off phase (PUC),
- after secondary air injection (SA),
- after hot start (HST) and
- after catalyst heating (CH) with lean air fuel mixture condition.
- after cylinder cut-off at gear shifting (GS_SCC).

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements, the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

Application Condition

`

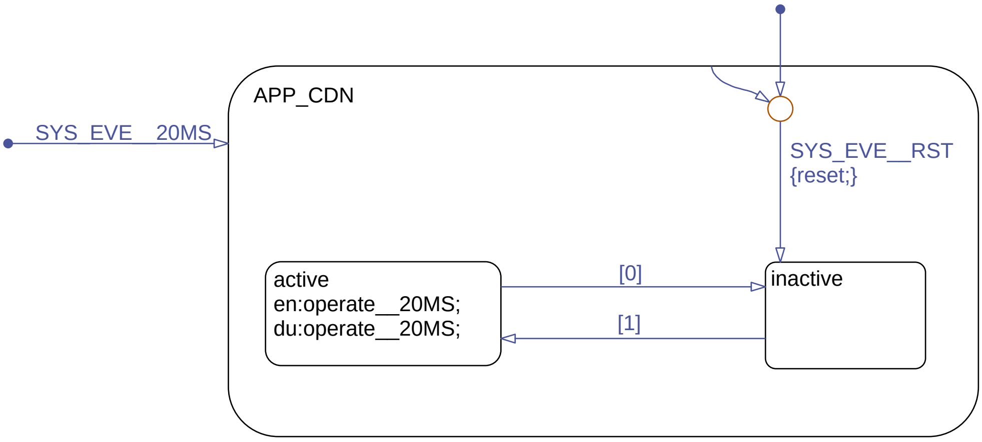
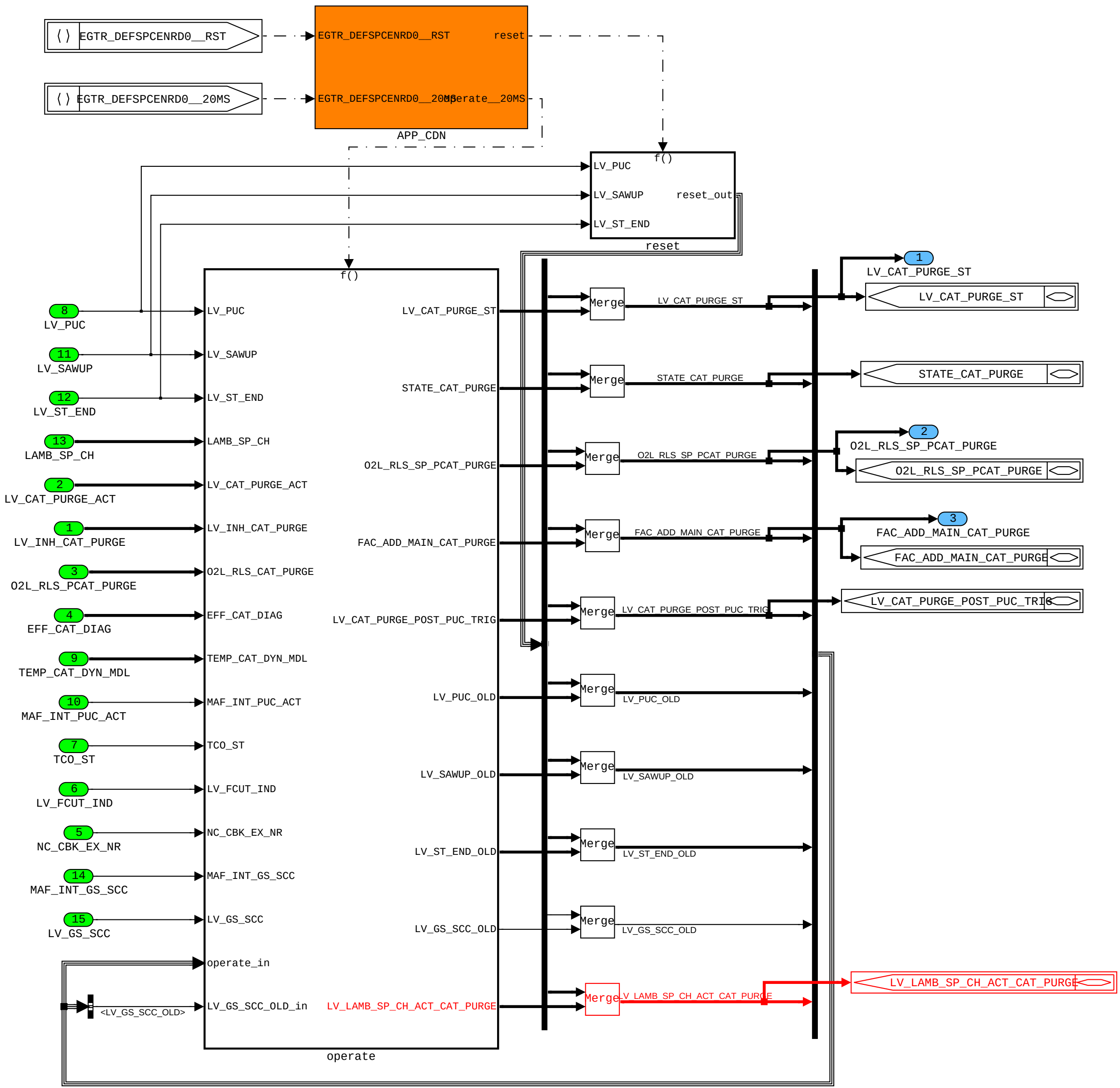


Figure 24 EGTR_DEFSPCENRD0/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 15−Dec−2005

Figure 25 EGTR_DEFSPCENRD0

18.5.1.1 SUBFUNCTION: operate

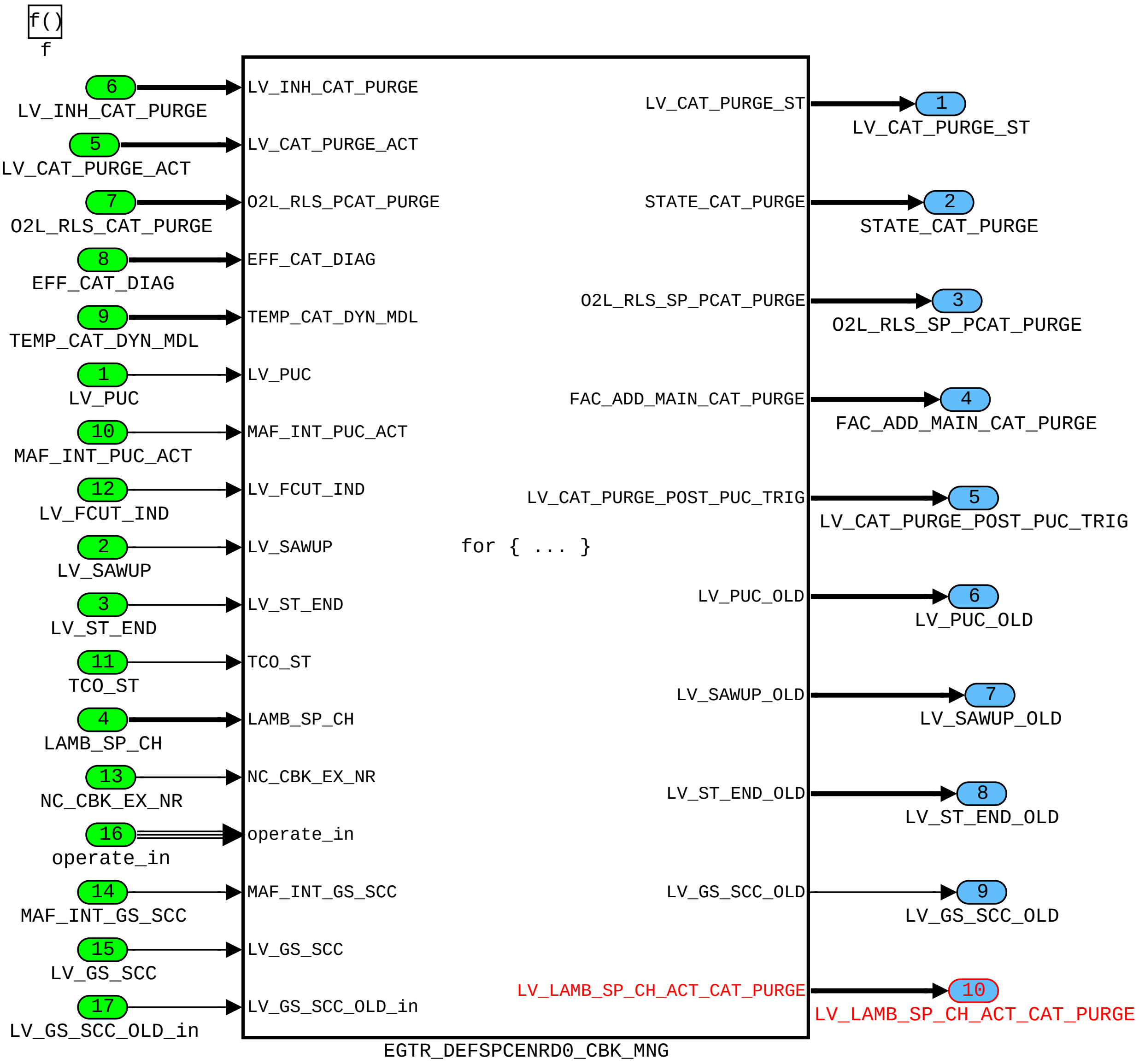


Figure 26 EGTR_DEFSPCENRD0/ operate

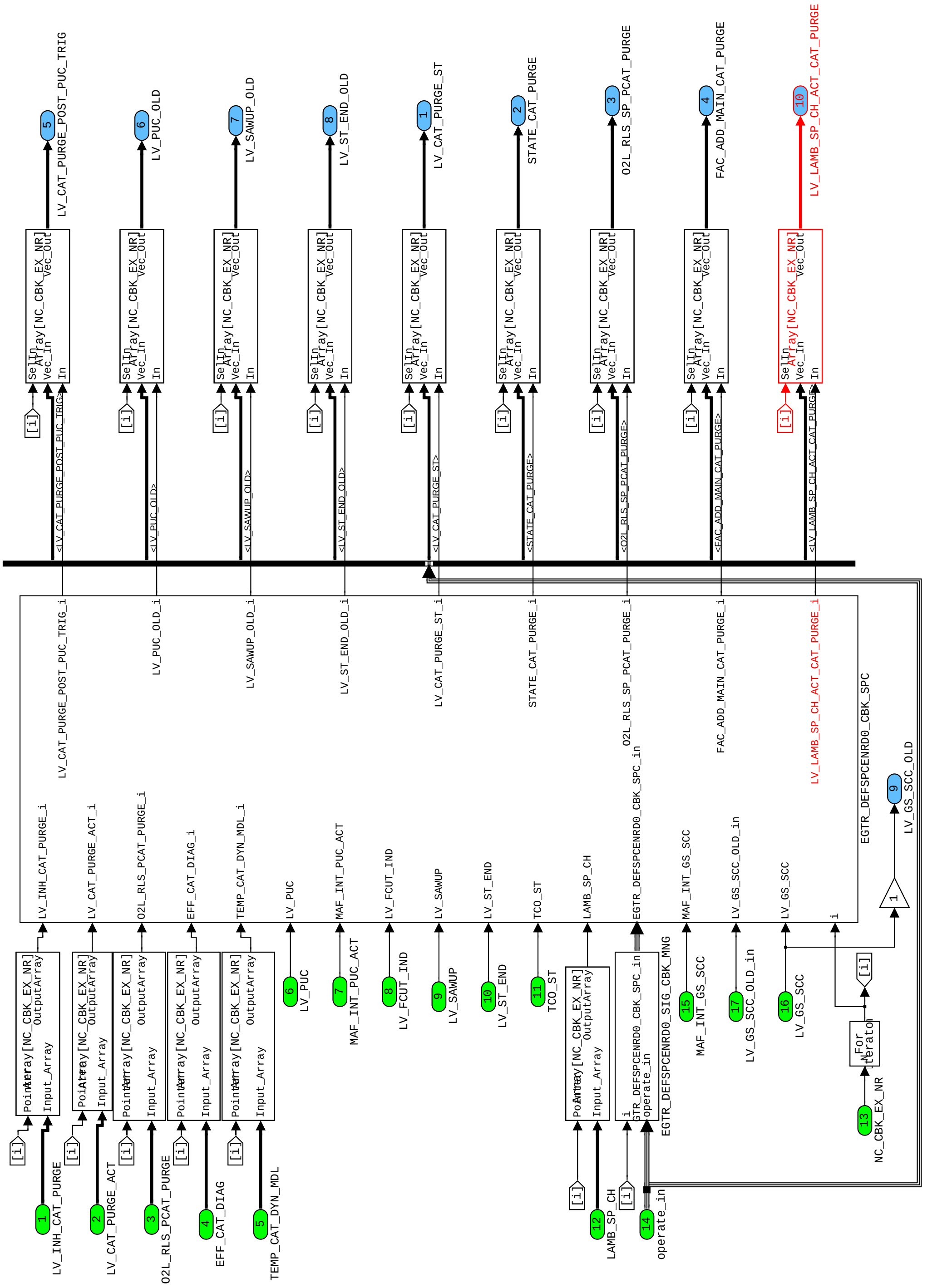


Figure 27 EGTR_DEFSPCENRD0/ operate/ EGTR_DEFSPCENRD0_CBK_MNG

Different catalyst purge cases

If LV_CAT_PURGE_ACT[i] is or gets 0 and STATE_CAT_PURGE[i] is not changing (compared to one sample step before) then the state is set to “CAT_PURGE_OFF”.

When one of the different cat purge cases gets active the flag LV_CAT_PURGE_ST[i] is set for one sample step to inform the main function.

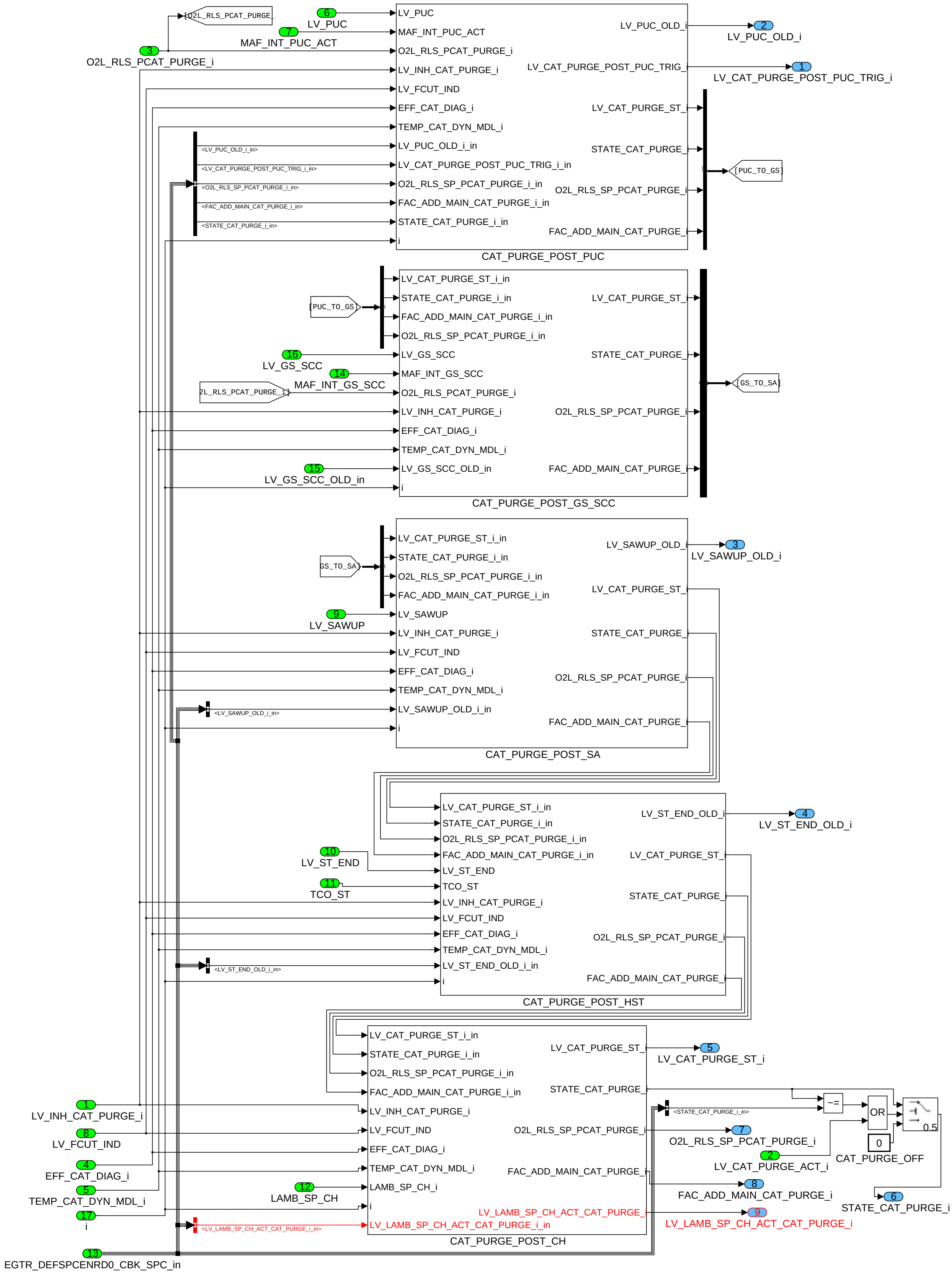


Figure 28 EGTR_DEFSPCENRD0/ operate/ EGTR_DEFSPCENRD0_CBK_MNG/ EGTR_DEFSPCENRD0_CBK_SPC

Catalyst purge after catalyst heating with lean air fuel mixture condition

Cat purge after catayst heating is activated at falling edge of LV_LAMB_CH. Cat purge after trailing throttle fuel cut-off phase must not be active, no cylinder must be shut off and the inhibition flag must not be set.

The cat purge state is set to “CAT_PURGE_CH”.

The oxygen loading to be relased during pre catalyst purge is defined by C_O2L_RLS_SP_PCAT_PURGE_CH multiplied by a catayst temperature and efficiency depending factor. An additional factor to calculate the main catayst loading is set by C_FAC_ADD_MAIN_CAT_PURGE_CH.

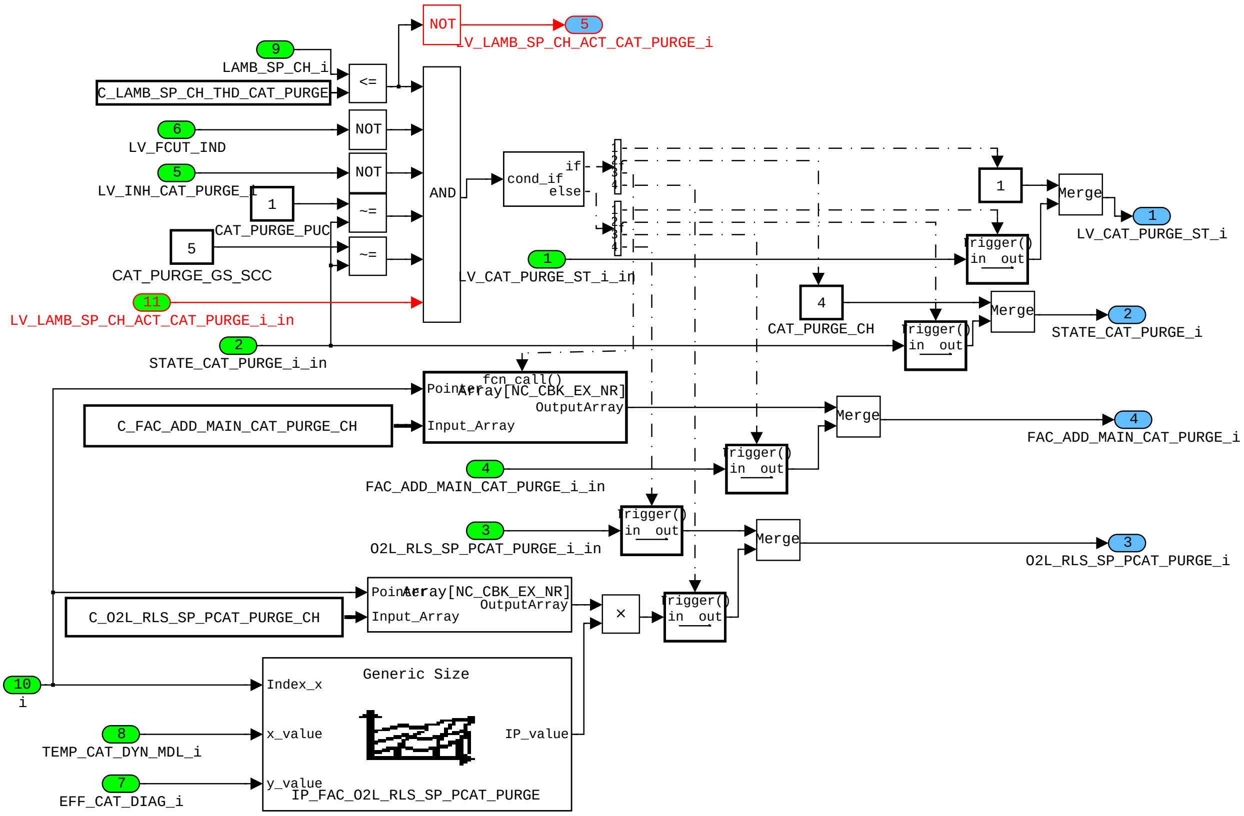


Figure 29 EGTR_DEFSPCENRD0/ operate/ EGTR_DEFSPCENRD0_CBK_MNG/ EGTR_DEFSPCENRD0_CBK_SPC/ CAT_PURGE_POST_CH

Catalyst purge after cylinder cut-off at gear shifting

Cat purge after cylinder cut-off at gear shifting is activated by a falling edge of LV_GS_SCC. It is similar to purge after PUC and has the same priority.

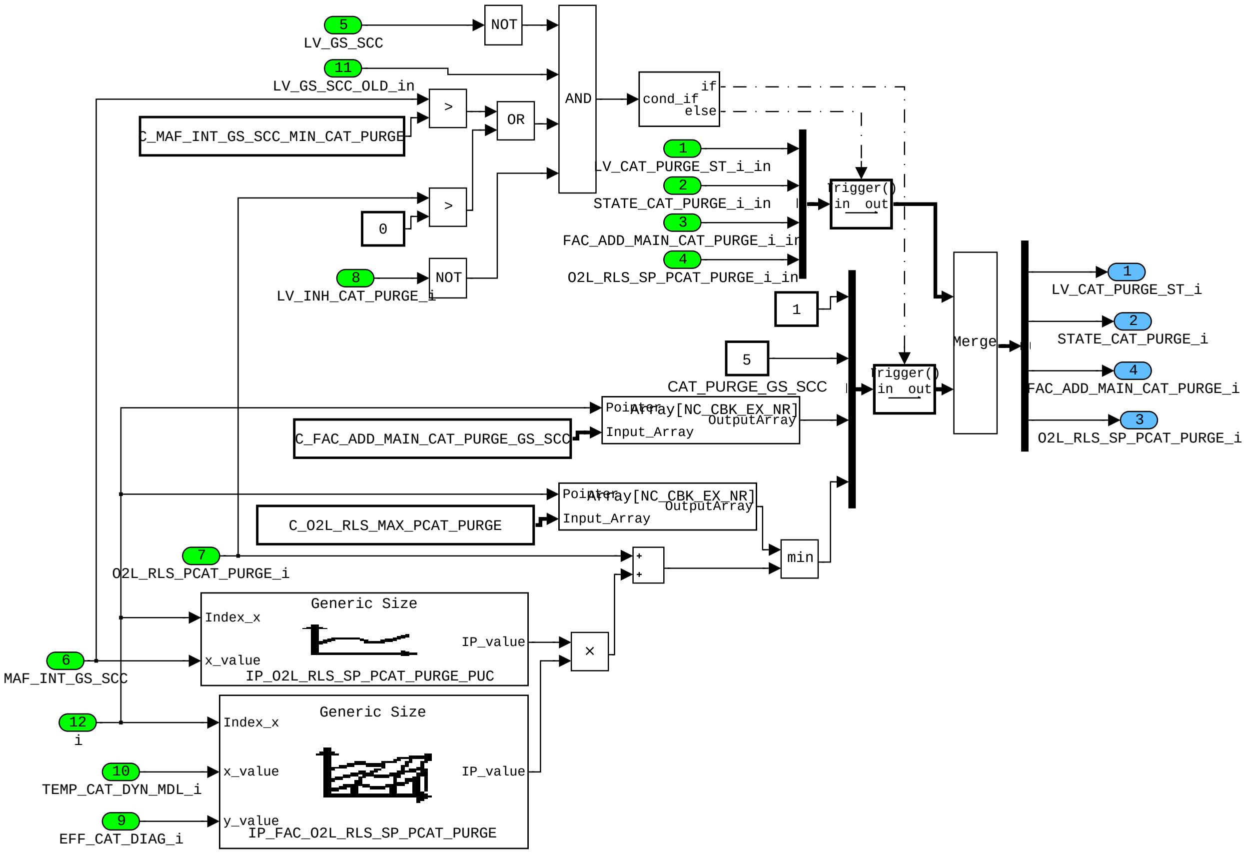


Figure 30 EGTR_DEFSPCENRD0/ operate/ EGTR_DEFSPCENRD0_CBK_MNG/ EGTR_DEFSPCENRD0_CBK_SPC/ CAT_PURGE_POST_GS_SCC

Catalyst purge after hot start

Cat purge after hot start is activated at rising edge of LV_ST_END when the cooling temperature at start (TCO_ST) is above a calibration threshold. No other cat purge case must be active, no cylinder must be shut off and the inhibition flag must not be set.

The cat purge state is set to “CAT_PURGE_HST”.

The oxygen loading to be relased during pre catalyst purge is defined by C_O2L_RLS_SP_PCAT_PURGE_HST multiplied by a catayst temperature and efficiency depending factor. An additional factor to calculate the main catayst loading is set by C_FAC_ADD_MAIN_CAT_PURGE_HST.

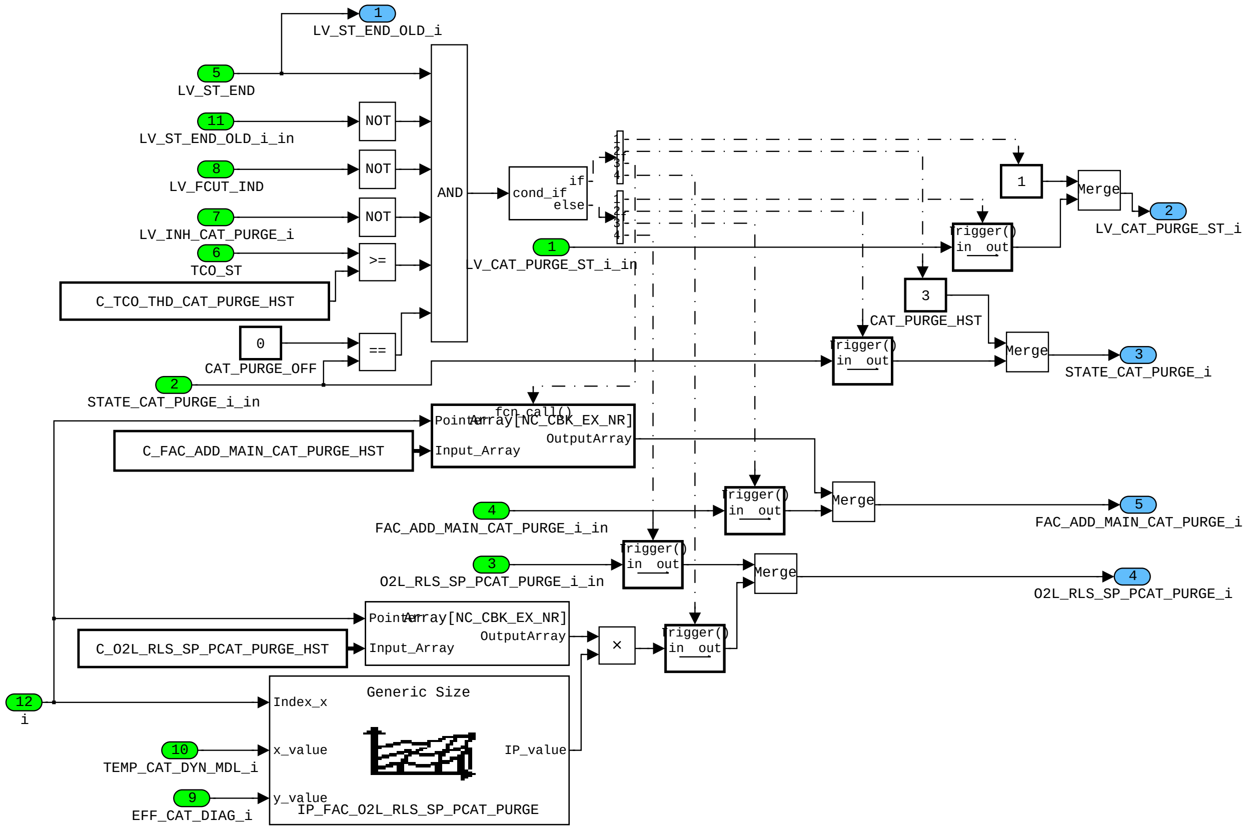


Figure 31 EGTR_DEFSPCENRD0/ operate/ EGTR_DEFSPCENRD0_CBK_MNG/ EGTR_DEFSPCENRD0_CBK_SPC/ CAT_PURGE_POST_HST

Catalyst purge after trailing throttle fuel cut-off phase

Cat purge after trailing throttle fuel cut-off (PUC) is activated at falling edge of LV_PUC. The inhibition flag must not be set and the air mass that was integrated during PUC must be above a minimum threshold or during last cat purge the pre cat purge was not finished (O2L_RLS_PCAT_PURGE[i] > 0).

If at least one cylinder is shut off (LV_FCUT_IND = 1) when the conditions above are fulfilled then the start of cat purge is delayed till no cylinder is shut off any more. This is realized by means of the auxiliary flag LV_CAT_PURGE_POST_PUC_TRIG[i].

The cat purge state is set to “CAT_PURGE_PUC”.

The oxygen loading to be relased during pre catalyst purge is defined by IP_O2L_RLS_SP_PCAT_PURGE_PUC multiplied by a catayst temperature and efficiency depending factor. The result is limited to the maximum oxygen loading of the pre catalyst defined by C_O2L_RLS_MAX_PCAT_PURGE. The additional factor to calculate the main catayst loading is set to 1.

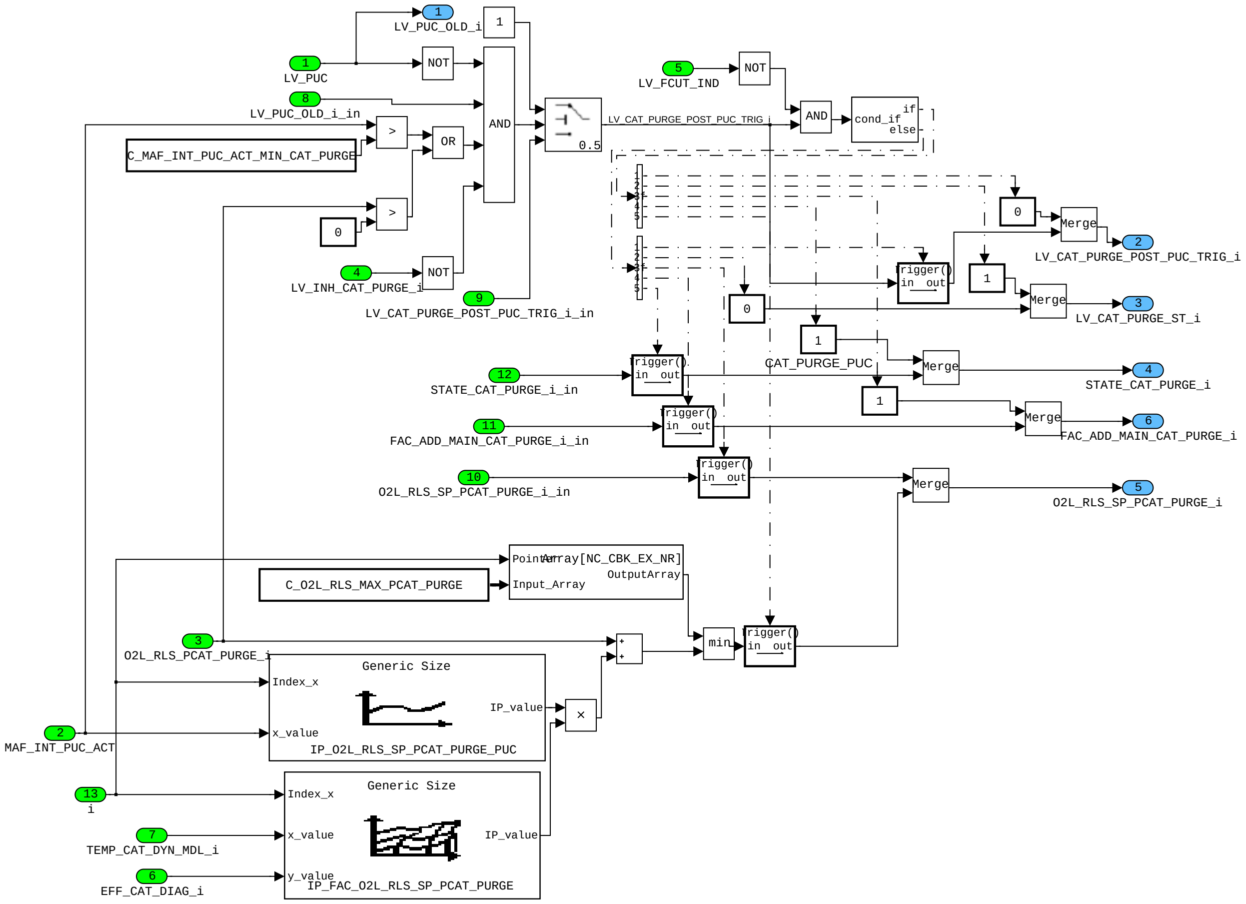


Figure 32 EGTR_DEFSPCENRD0/ operate/ EGTR_DEFSPCENRD0_CBK_MNG/ EGTR_DEFSPCENRD0_CBK_SPC/ CAT_PURGE_POST_PUC

Catalyst purge after secondary air injection

Cat purge after secondary air is activated at falling edge of LV_SAWUP. Cat purge after trailing throttle fuel cut-off phase must not be active, no cylinder must be shut off and the inhibition flag must not be set.

The cat purge state is set to “CAT_PURGE_SA”.

The oxygen loading to be relased during pre catalyst purge is defined by C_O2L_RLS_SP_PCAT_PURGE_SA multiplied by a catayst temperature and efficiency depending factor. An additional factor to calculate the main catayst loading is set by C_FAC_ADD_MAIN_CAT_PURGE_SA.

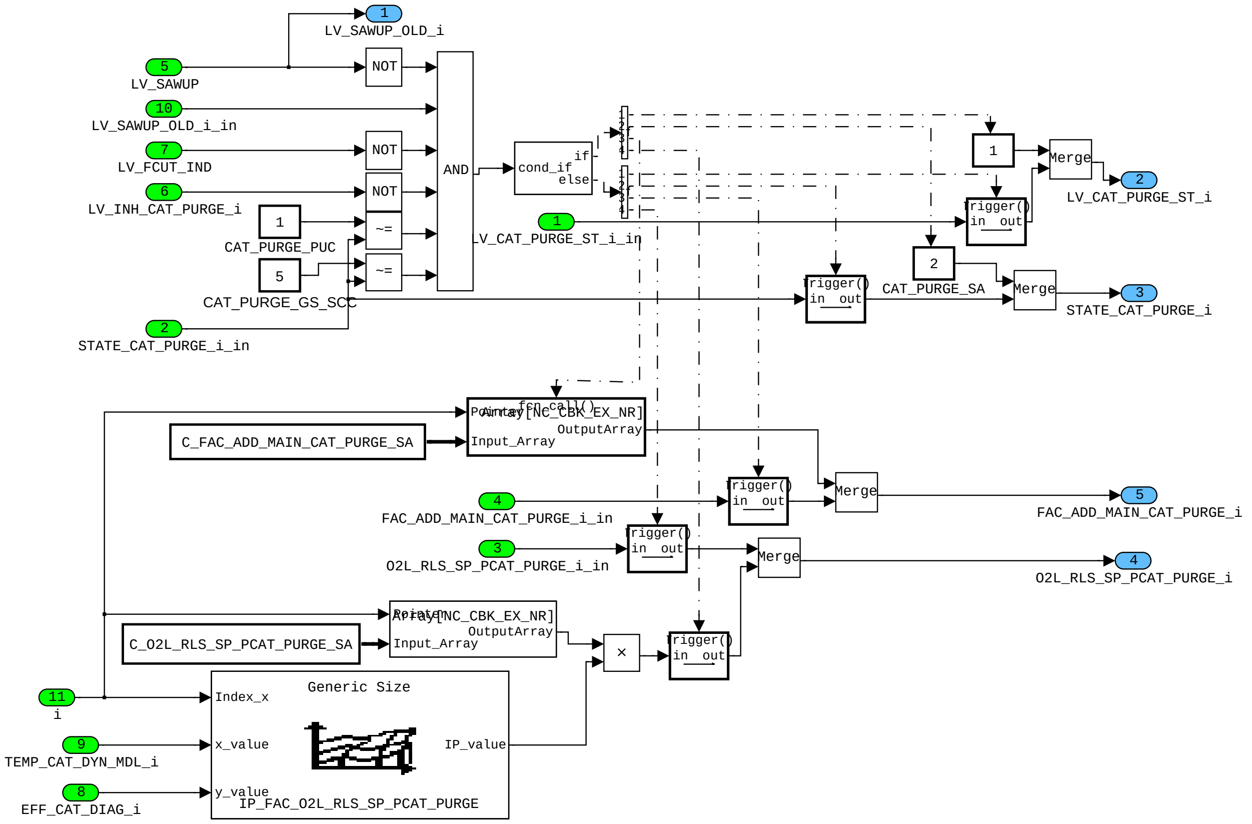


Figure 33 EGTR_DEFSPCENRD0/ operate/ EGTR_DEFSPCENRD0_CBK_MNG/ EGTR_DEFSPCENRD0_CBK_SPC/ CAT_PURGE_POST_SA

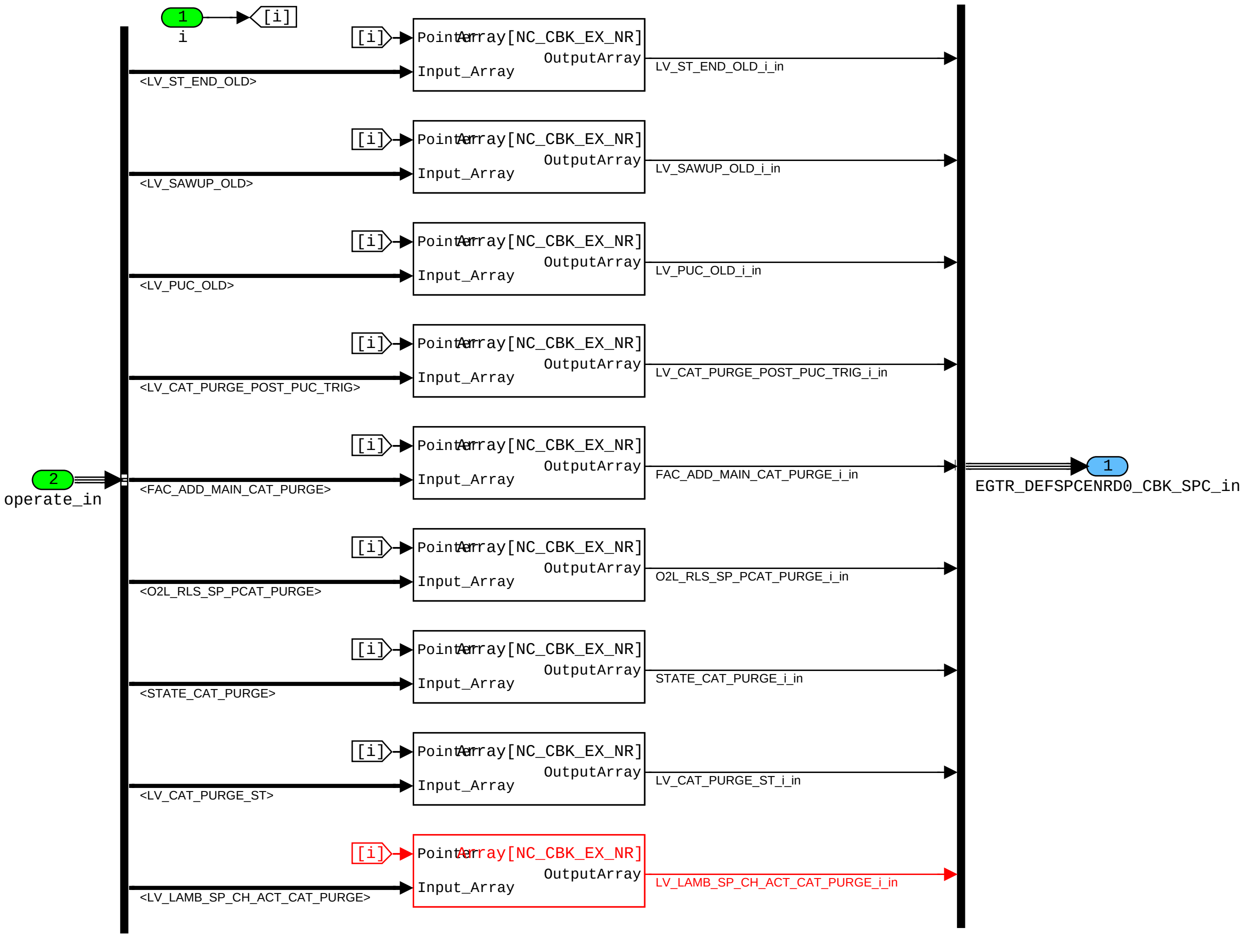


Figure 34 EGTR_DEFSPCENRD0/ operate/ EGTR_DEFSPCENRD0_CBK_MNG/ EGTR_DEFSPCENRD0_SIG_CBK_MNG

18.5.1.2 SUBFUNCTION: reset

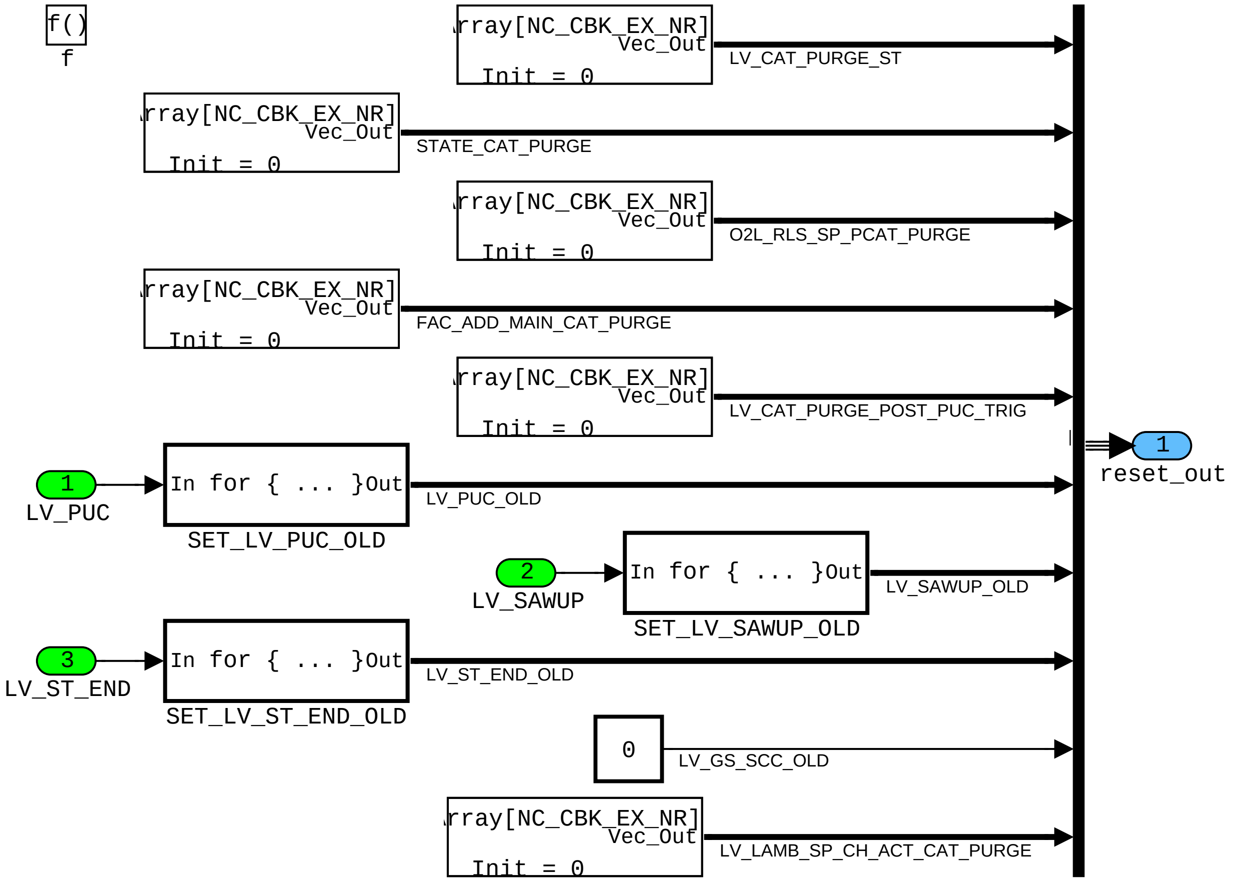


Figure 35 EGTR_DEFSPCENRD0/ reset

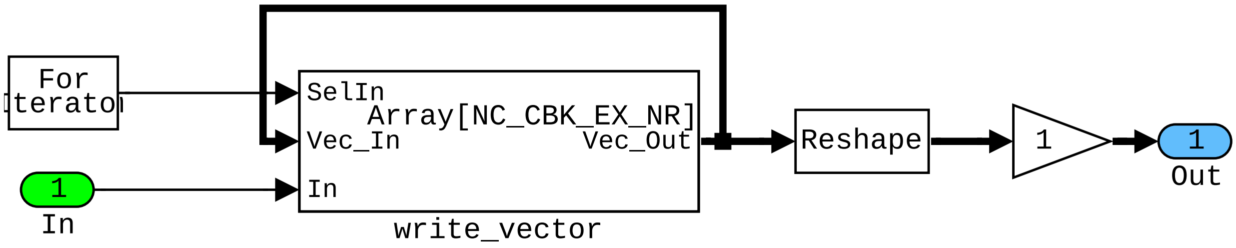


Figure 36 EGTR_DEFSPCENRD0/ reset/ SET_LV_PUC_OLD

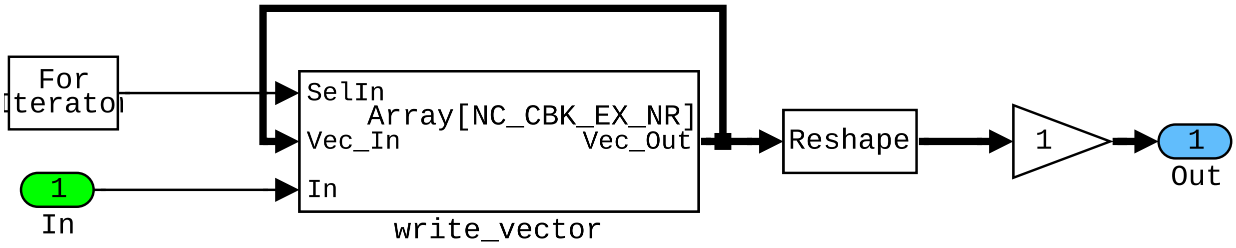


Figure 37 EGTR_DEFSPCENRD0/ reset/ SET_LV_SAWUP_OLD

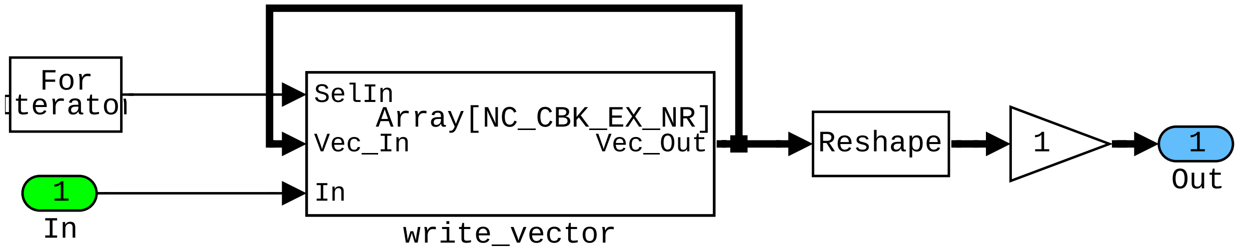


Figure 38 EGTR_DEFSPCENRD0/ reset/ SET_LV_ST_END_OLD

18.6 Catalyst enrichment function

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CAT_PURGE_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Status of catalyst purge activation					
LAMB_SP_DELTA_IT_CAT_PURGE[NC_CBK_EX_NR]	O/V	0...FFFH	0...0.249939	6.10352E-5	-
lambda set point shift of cat purge function (internal value)					
O2L_RLS_PCAT_PURGE[NC_CBK_EX_NR]	O/V	80000000...7FFFFFFFH	-669.923556 ... 669.923555	3.11957E-7	g
oxygen loading to be released for pre cat purge					
LV_VLS_DOWN_ACT_CAT_PURGE[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that downstream signal is/was below calibration threshold					
RATIO_O2L_RLS_CAT_PURGE[NC_CBK_EX_NR]	V	0...FFH	0...0.99609375	0.00390625	-
quotient used by cat purge: current rest oxygen loading / total oxygen loading					
FAC_O2L_RLS_CAT_PURGE[NC_CBK_EX_NR]	V	8000...7FFFH	-0.25 ... 0.24999237	7.62939E-6	-
temporary factor for O2 loading release calculation considering the lambda set point deviation					
LV_VLS_MAIN_CAT_PURGE_ACT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that sensor signal after main cat is/was below calibration threshold					
LV_VLS_MAIN_CAT_PRE_PURGE_ACT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that sensor signal after main cat is/was below calibration threshold for pre cat purge					
O2L_RLS_TOT_CAT_PURGE[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-669.923556 ... 669.923555	3.11957E-7	g
total oxygen loading to be released for pre and main cat purge					
O2L_RLS_REST_CAT_PURGE[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-669.923556 ... 669.923555	3.11957E-7	g
estimated rest oxygen loading to be released for pre and main cat purge					
O2L_RLS_ADD_CAT_PURGE[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-669.923556 ... 669.923555	3.11957E-7	g
oxygen loading to be added each sample step for cat purge air mass flow integral					
O2L_RLS_MAIN_CAT_PURGE[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-669.923556 ... 669.923555	3.11957E-7	g
oxygen loading to be released for main cat purge					
LV_CAT_PURGE_MAIN_CAT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that cat purge for main cat is active					
LV_CAT_PURGE_PCAT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that cat purge for pre cat is active					

Input data:

LV_INH_CAT_PURGE[NC_CBK_EX_NR]	TEMP_MAIN_CAT_PURGE[NC_CBK_EX_NR]	LV_CAT_PURGE_ST[NC_CBK_EX_NR]	O2L_RLS_SP_PCAT_PURGE[NC_CBK_EX_NR]
FAC_ADD_MAIN_CAT_PURGE[NC_CBK_EX_NR]	EFF_CAT_DIAG[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]	LV_ERR_LS_DOWN[NC_CBK_EX_NR]
STATE_LSH_DOWN[NC_CBK_EX_NR]	LV_PUC	MAF_CYL	NC_CBK_EX_NR
VLS_MAIN_CAT_PURGE[NC_CBK_EX_NR]	LV_VLS_MAIN_CAT_PURGE_READY[NC_CBK_EX_NR]	LV_ERR_VLS_MAIN_CAT_PURGE[NC_CBK_EX_NR]	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VLS_DOWN_THD_CAT_PURGE[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511719	0.00488281	V
downstream voltage signal threshold for end of pre cat purge detection					
C_VLS_MAIN_CAT_PURGE_THD	1	0...3FFH	0...4.99511719	0.00488281	V
voltage signal threshold from sensor after main cat for end of main cat purge detection					
LC_VLS_MAIN_CAT_PURGE[NC_CBK_EX_NR]	1	0...1H	0...1	1	-
calibration flag to use sensor signal after main cat to abort cat purge					
IP_FAC_MAIN_CAT_PURGE	6x6	0...FFH	0...1.9921875	0.0078125	-
LDP_TEMP_MAIN_CAT_PURGE_IP_FAC	6	0...7FFFH	-273.15 ... 1.77479E+3	0.0625	°C
LDPM_EFF_CAT_DIAG_1_EGTR	6	0...FFH	0...1.9921875	0.0078125	-
weighting factor considering volume ratio (pre / main cat) and cat temperature (for main cat purge)					
IP_LAMB_SP_DELTA_CAT_PURGE	6x8	0...FFFH	0...0.249939	6.10352E-5	-
LDP_MAF_CYL_IP_LAMB_SP	6	0...FFFFH	0...2.04797E+3	0.03125	kg/h
LDP_RATIO_O2L_RLS_IP_LAMB_SP	8	0...FFH	0...0.99609375	0.00390625	-
lambda set point shift of catalyst purge function					

18.6.1 EGTR_DEFSPCENR0

The catalyst purge function realizes fast adjustment of optimized conditions for a high catalyst efficiency after a transition from trailing throttle fuel cut-off phase or other lean air fuel mixture conditions to stoichiometric conditions. That requires a fuel enrichment in order to reduce the oxygen content inside the catalyst. The degree of the fuel enrichment depends on the oxygen and the NOx content in the catalyst.

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements, the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

Application Condition

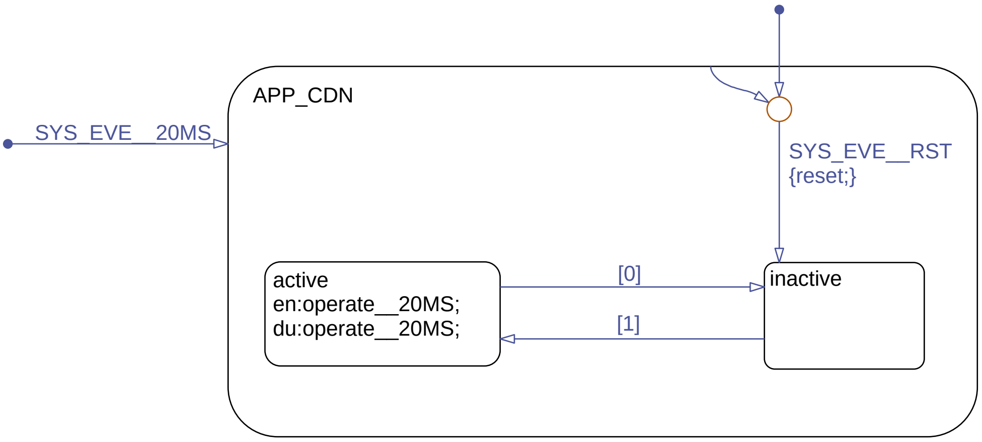


Figure 39 EGTR_DEFSPCENR0/ APP_CDN/ Chart

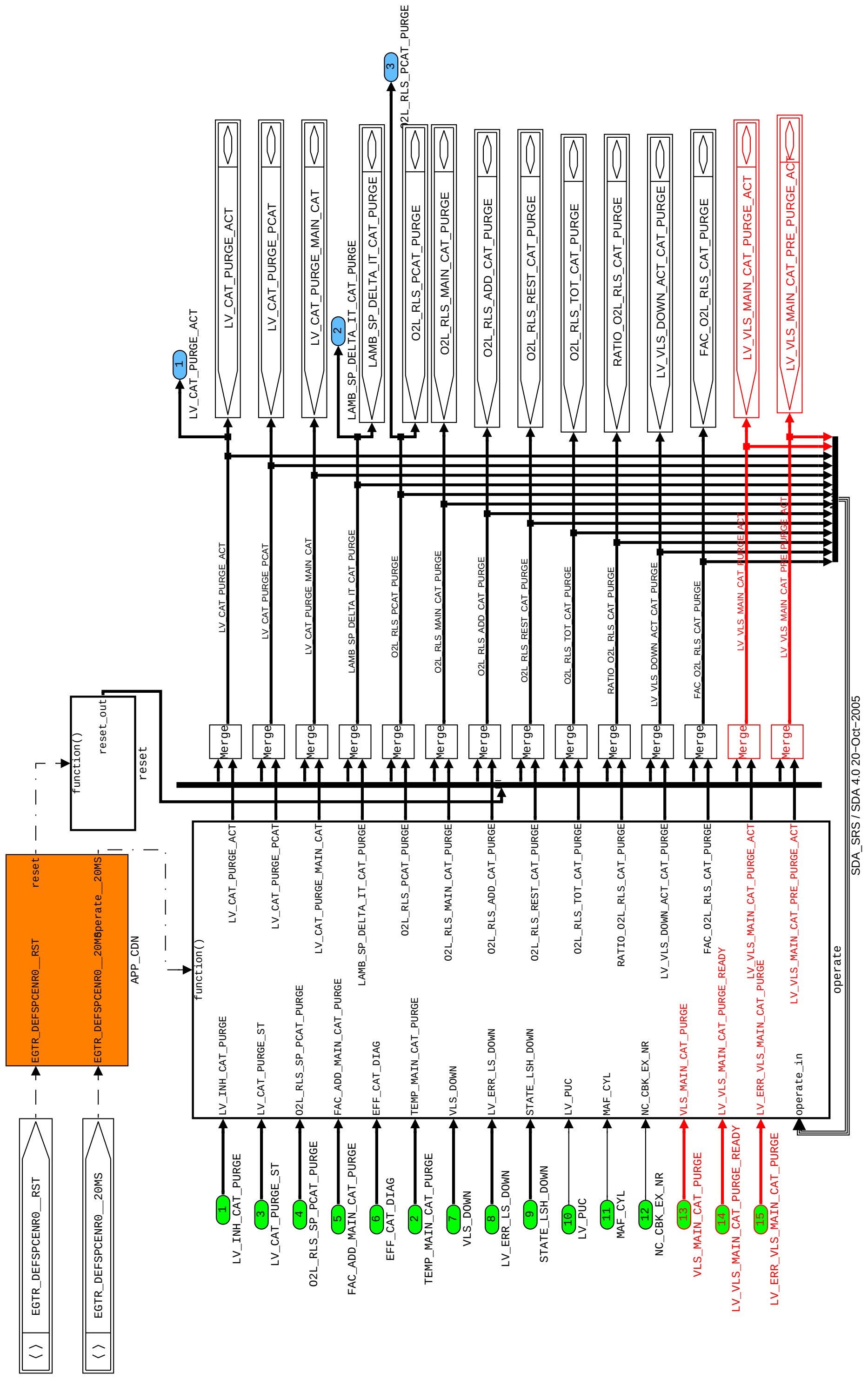


Figure 40 EGTR_DEFSPCENR0

18.6.1.1 SUBFUNCTION: operate

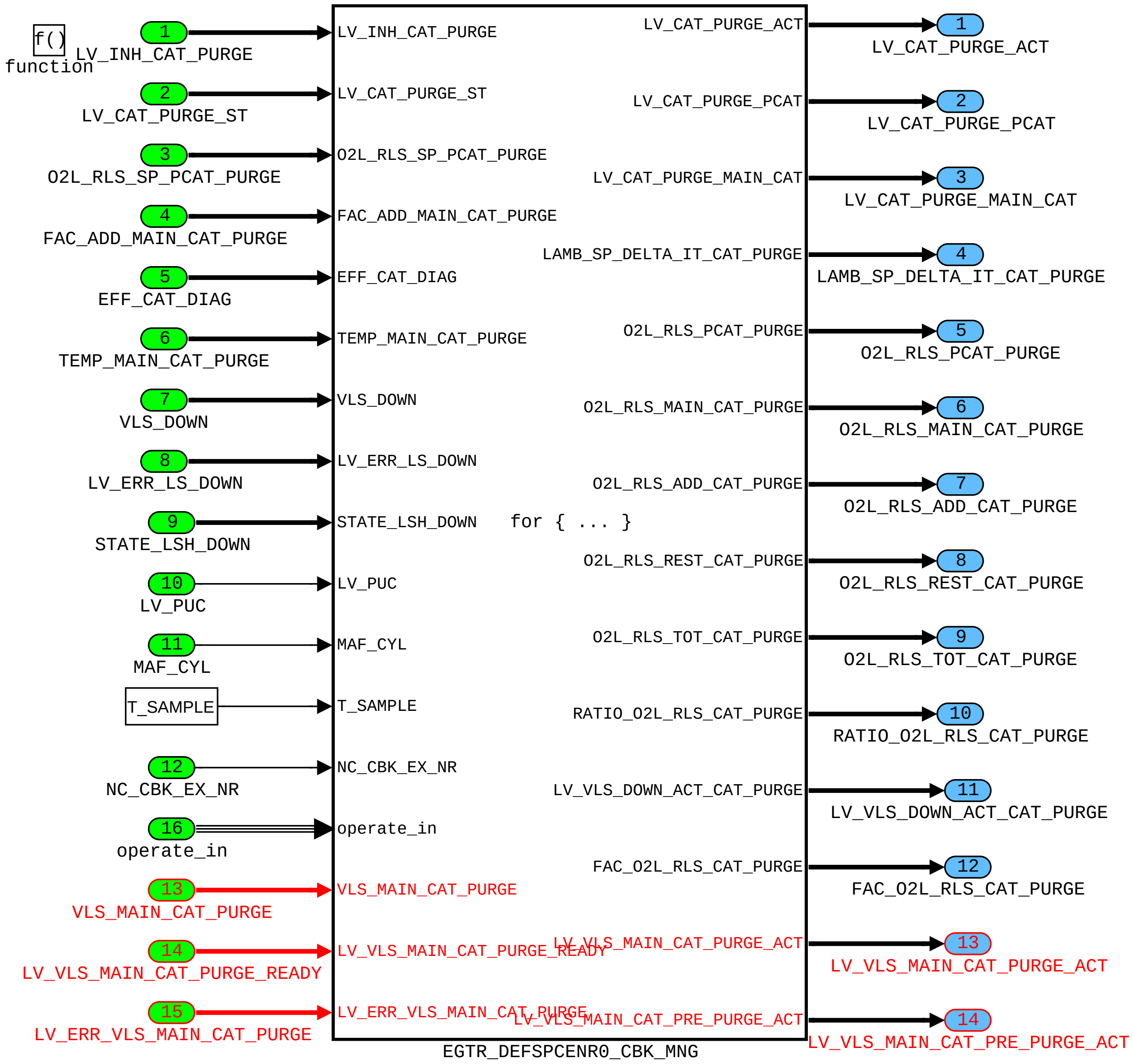


Figure 41 EGTR_DEFSPCENR0/ operate

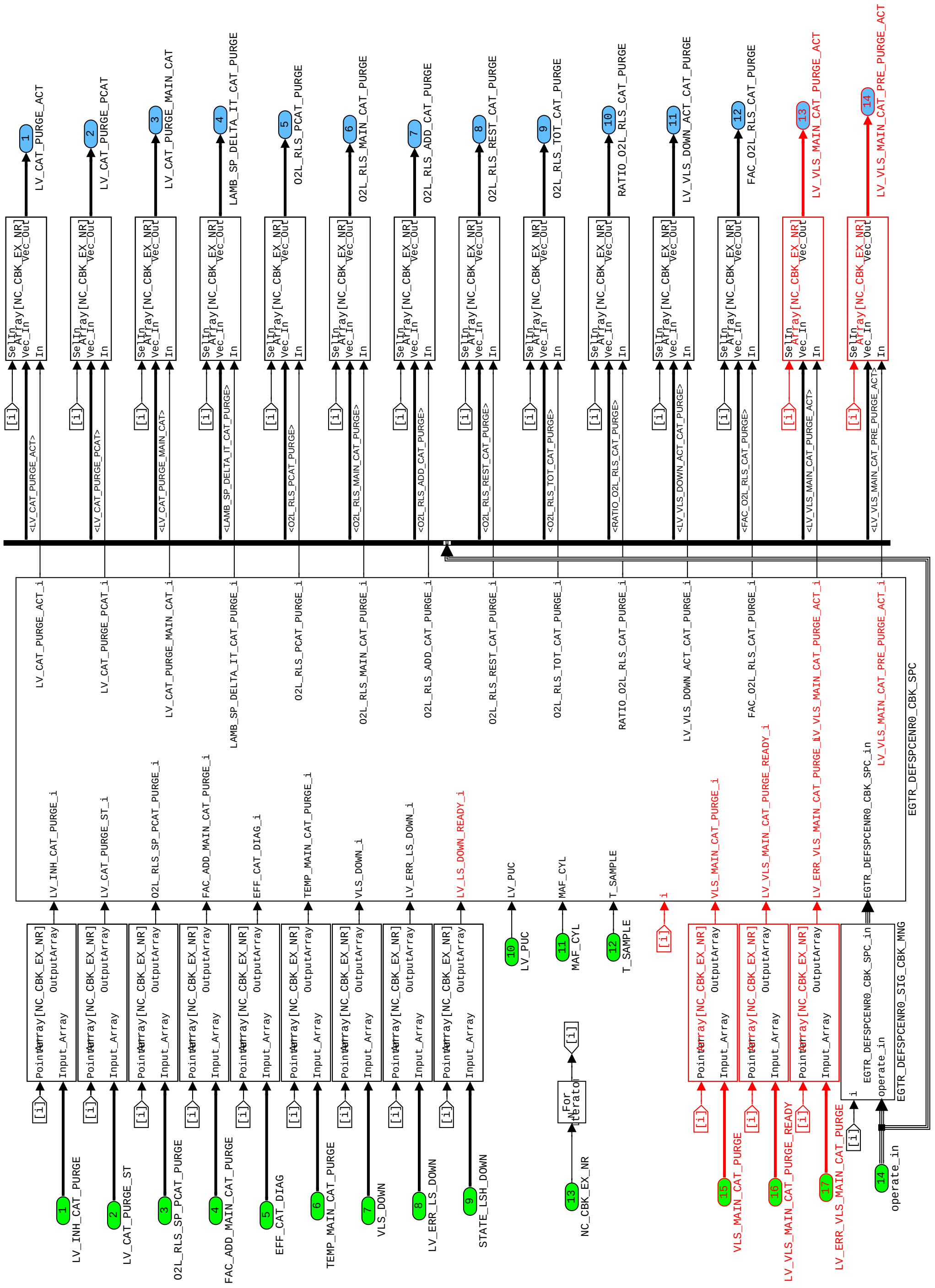


Figure 42 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG

<u>Management of catalyst purge initialization and pre and main catayst purge</u>

If LV_CAT_PRUGE_ST[i] is set by the function “Oxygen loading determination for catalyst enrichment function” the initialization sub-system (”INI_CAT_PURGE”) is called. The flag is set only for one sample step.

Depending on LV_CAT_PURGE_PCAT[i] and LV_CAT_PURGE_MAIN_CAT[i] one of the sub-systems “PRE_CAT_PURGE” or “MAIN_CAT_PURGE” is called.

If LV_PUC or the inhibition flag is set no cat purge is executed or an active cat purge is interrupted (sub-system “NOT_CAT_PURGE is called).

The cat purge activity flag LV_CAT_PURGE[i] is set to 1 if pre or main cat purge is active.

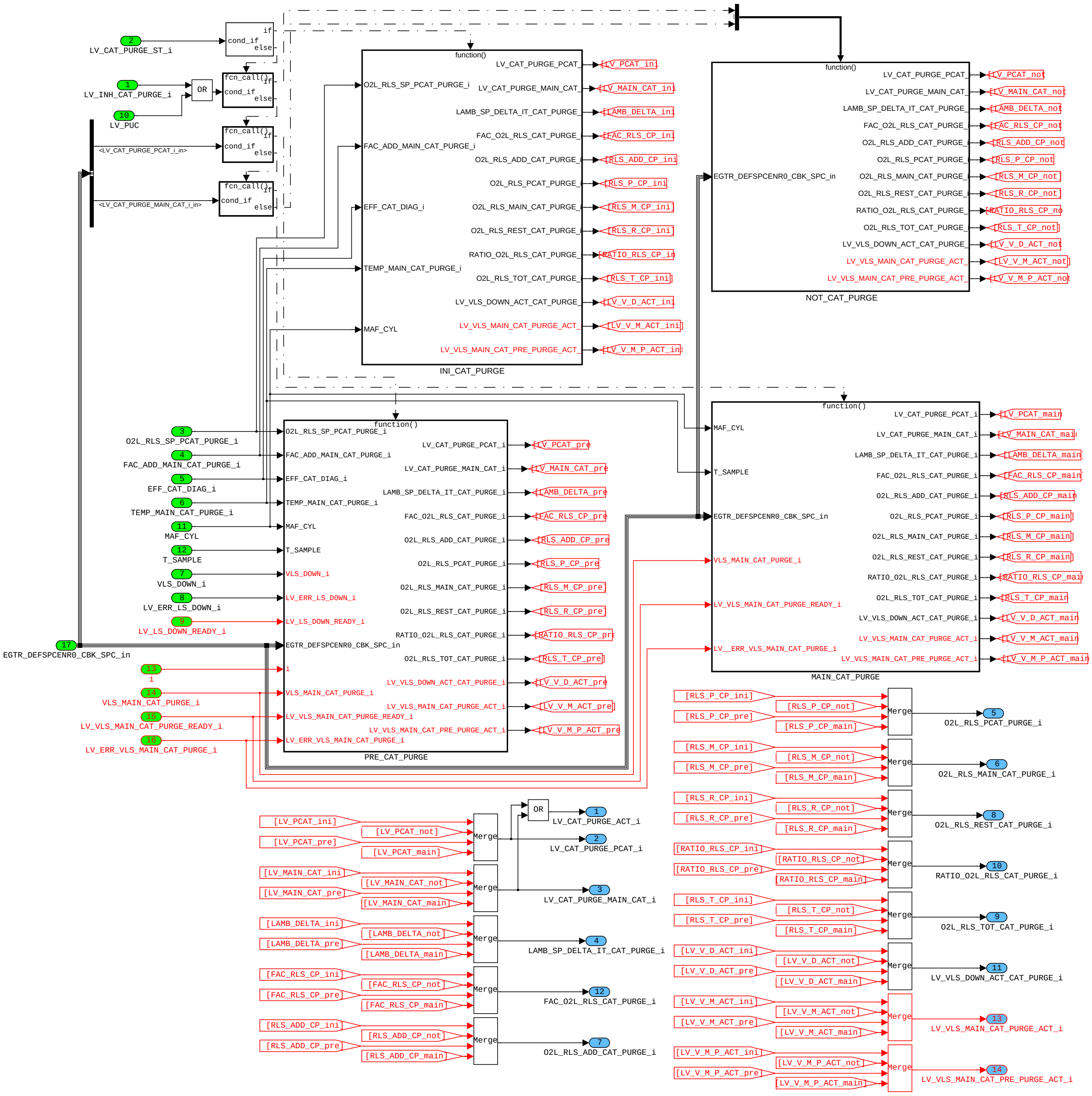


Figure 43 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC

Initialization of catalyst purge

Pre cat purge is activated by setting the respective flag. Main cat purge is deactivated.

The oxygen loading release integral for pre cat purge O2L_RLS_PCAT_PURGE[i] is initialised by O2L_RLS_SP_PCAT_PURGE[i]. The oxygen loading release integral for main cat purge is calulated by multiplying the one for pre cat purge with a factor map and with the additional factor defined in the function “Oxygen loading determination for catalyst enrichment function”. The rest oxygen loading release and the total oxygen loading release is the sum of the one for pre cat purge and main cat purge.

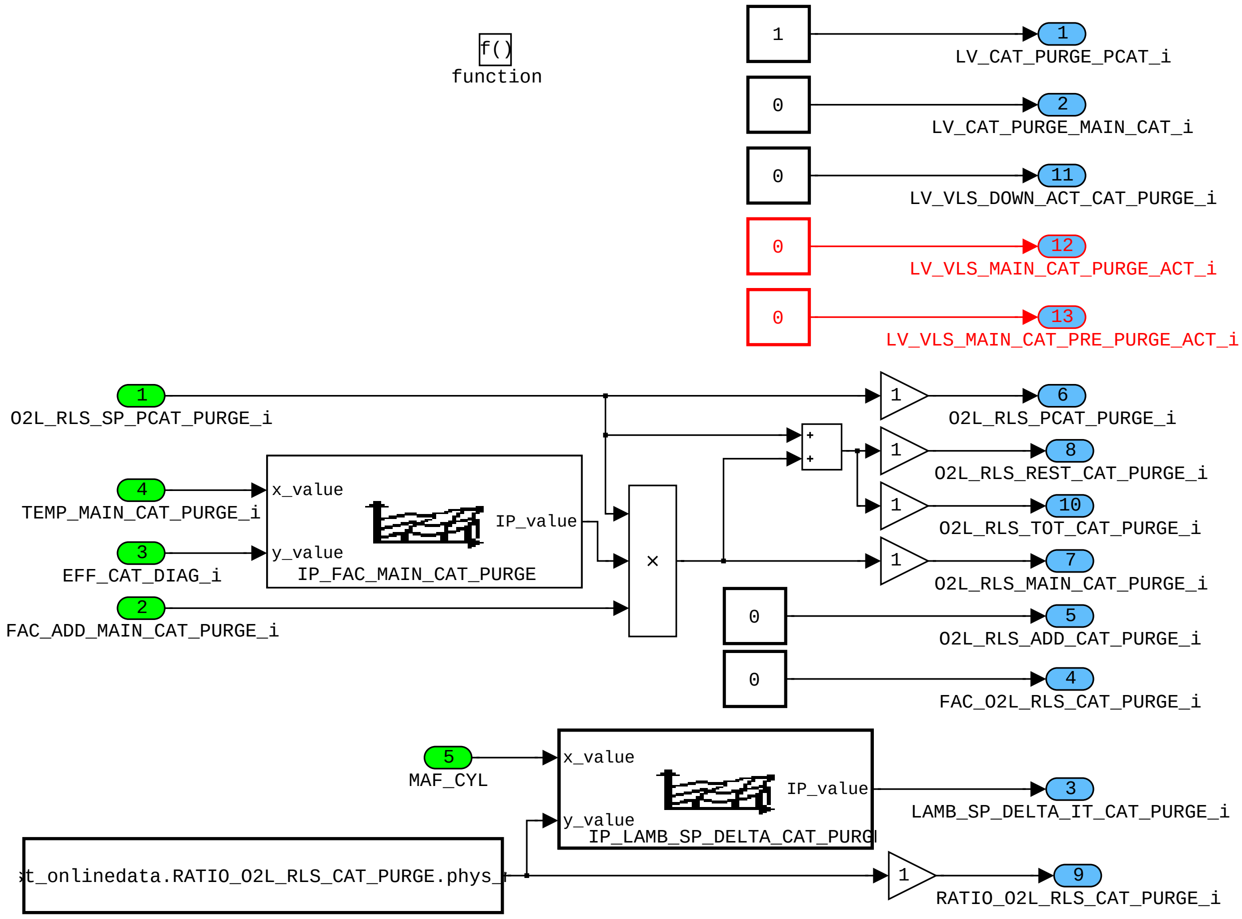


Figure 44 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ INI_CAT_PURGE

Main catalyst purge

If the oxygen loading for the main catalyst is released (O2L_RLS_MAIN_CAT_PURGE[i] <= 0) the end of main cat purge is detected and the respective sub-system “MAIN_CAT_PURGE_END” is triggered. Otherwise the sub-system ”MAIN_CAT_PURGE_CLC” is called.

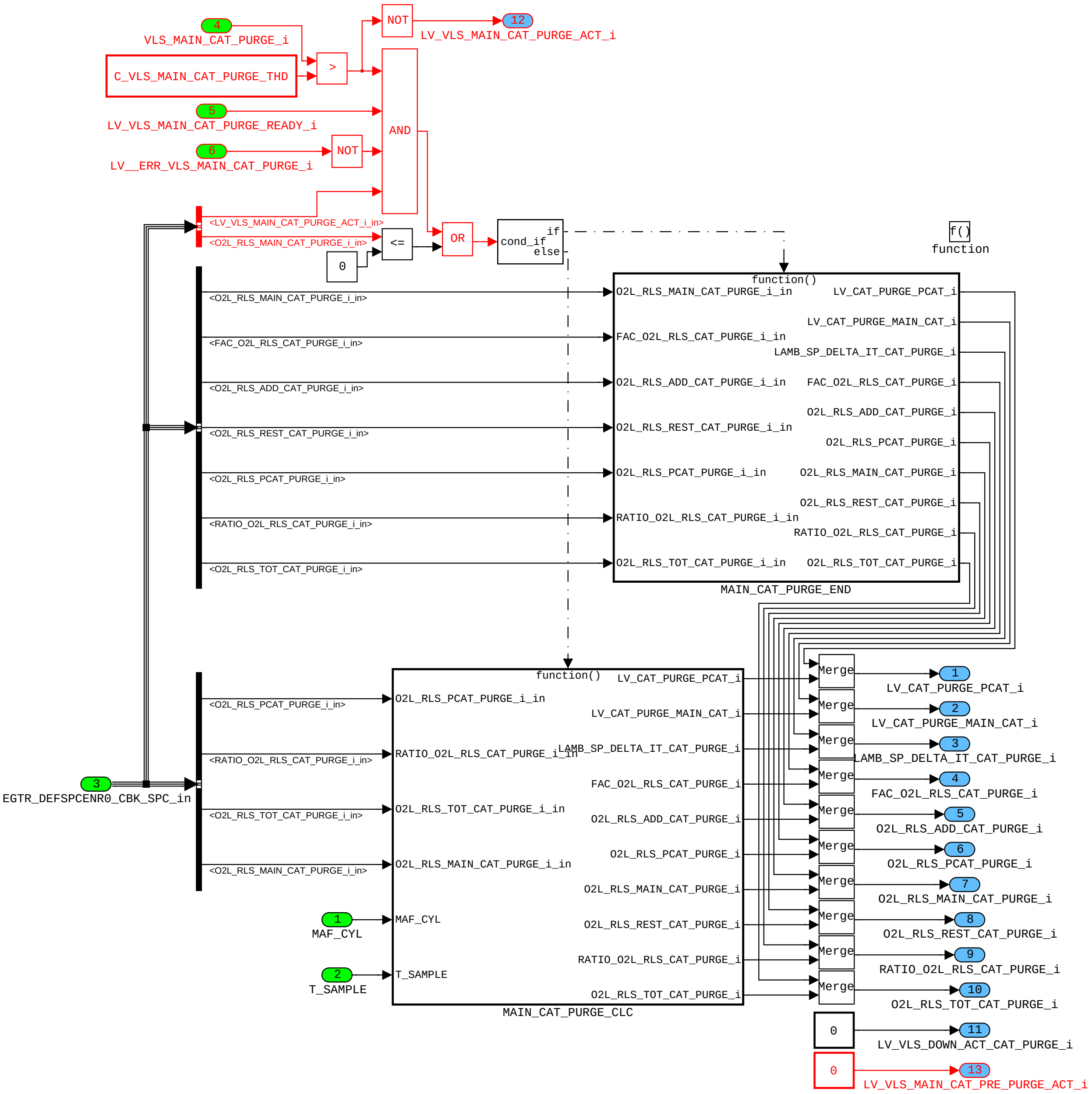


Figure 45 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ MAIN_CAT_PURGE

Calculation of main catalyst purge oxygen loading release

See textual description for “Calculation of pre catalyst purge oxygen loading release”.

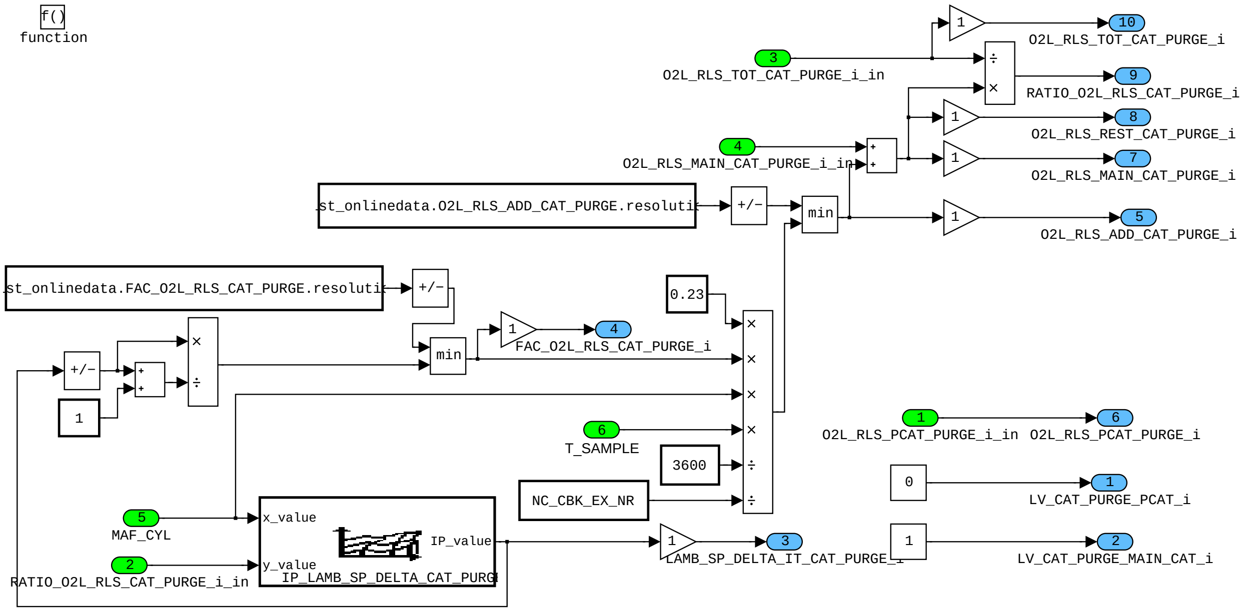


Figure 46 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ MAIN_CAT_PURGE/ MAIN_CAT_PURGE_CLC

End of main catalyst purge

Pre cat purge and main cat pruge is deactivated by setting the respective flags. The rich shift of the lambda set point is set to 0

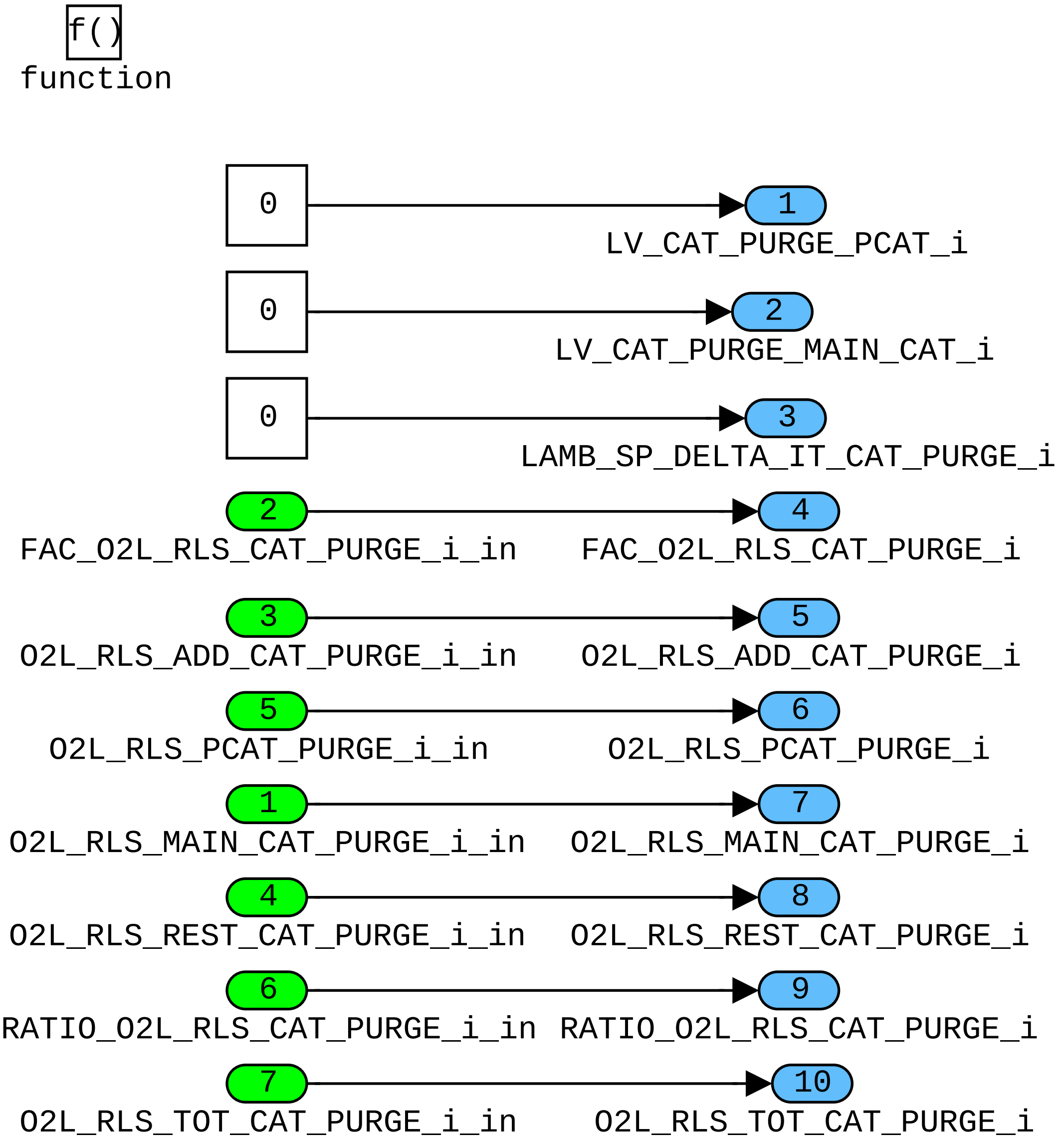


Figure 47 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ MAIN_CAT_PURGE/ MAIN_CAT_PURGE_END

Catalyst purge not active

Pre cat purge and main cat purge is deactivated by setting the respective flags to 0. The rich shift of the lambda set point is also set to 0.

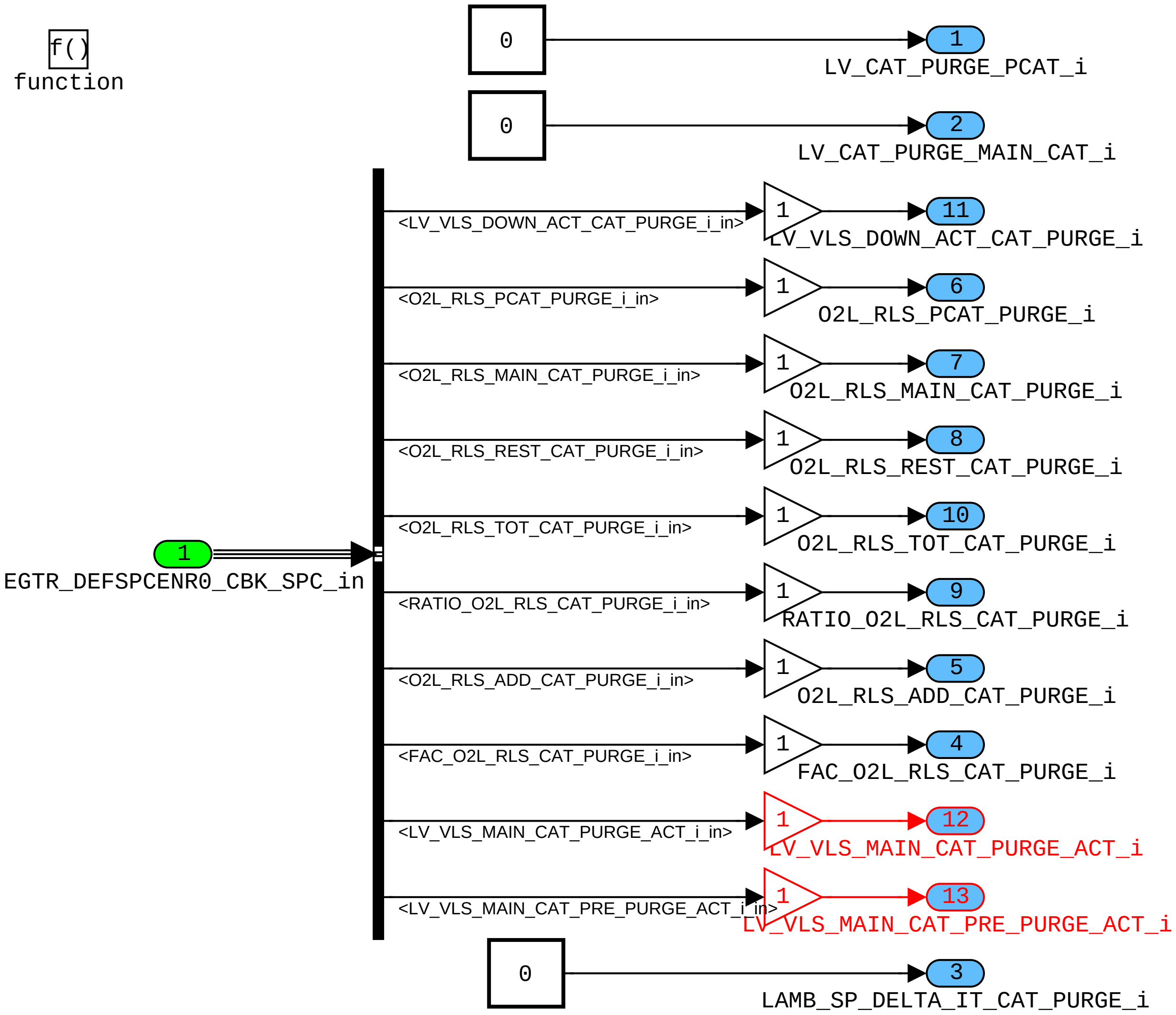


Figure 48 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ NOT_CAT_PURGE

Pre catalyst purge

If the oxygen loading for the pre catalyst is released (O2L_RLS_PCAT_PURGE[i] <= 0) or the downstream sensor signal exceeds a calibration threshold the end of pre cat purge is detected and the respective sub-system is triggered. The downstream signal is only evaluated if no error at the downstream sensor is present and the heater state is in “LSH_POW_CTL”. Furthermore the downstream signal must have been below the calibration threshold.

If the above mentioned conditions are not fulfilled the sub-system “PRE_CAT_PRUGE_CLC” is called.

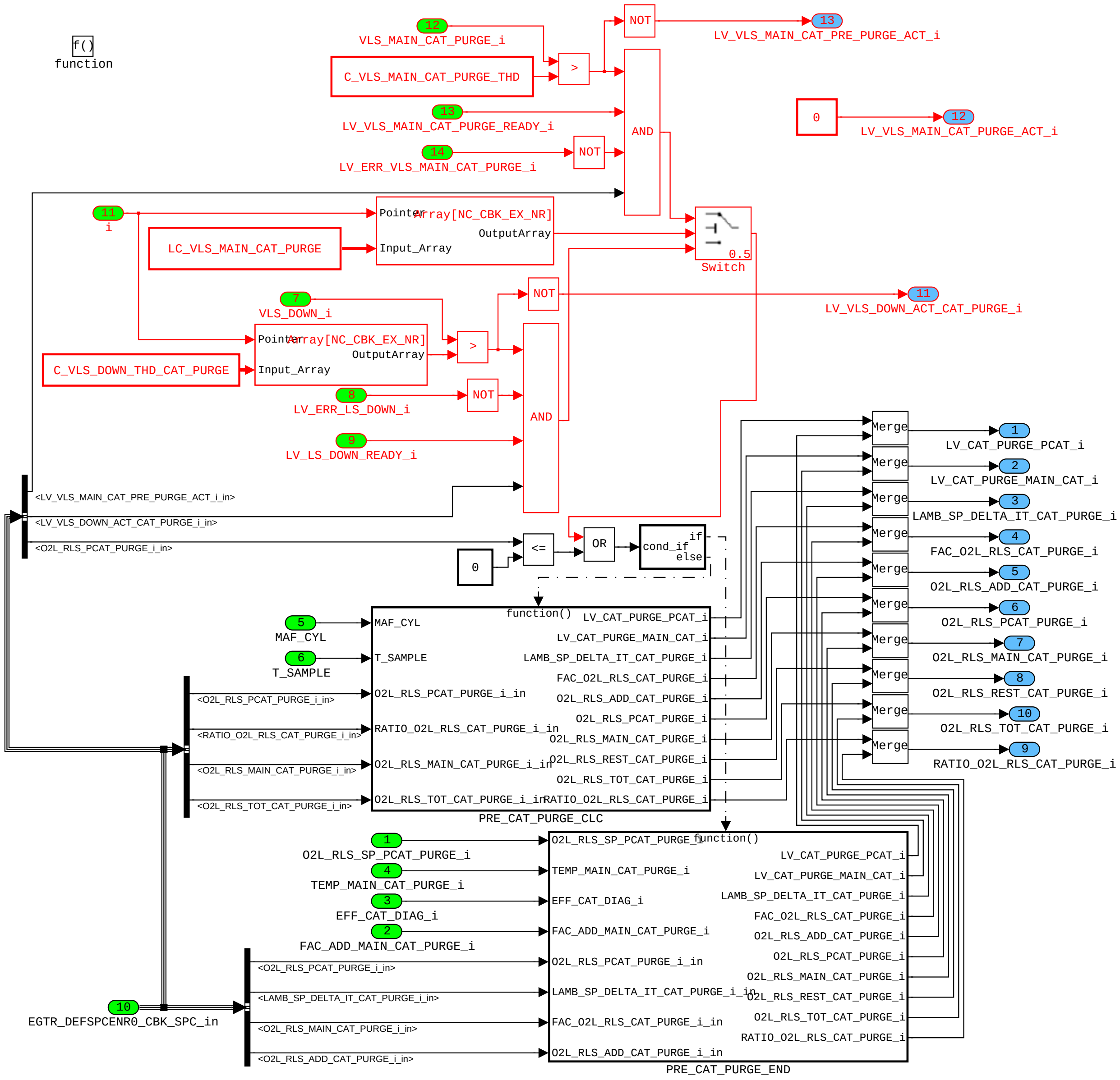


Figure 49 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ PRE_CAT_PURGE

Calculation of pre catalyst purge oxygen loading release

The rich shift of the lambda set point is taken from the respective map. The oxygen loading release is calulated by the following formula:

$$m_{O_2} = 0.23 \cdot \int \frac{-\Delta\lambda_{SP}}{1-\Delta\lambda_{SP}} \cdot \dot{m}_{air} dt$$

The remaining oxygen loading release (O2L_RLS_REST_CAT_PURGE[i]) is renewed and the quotient (RATIO_O2L_RLS_CAT_PURGE[i]): rest / total oxygen loading release is calculated.

For the calculation of the oxygen loading the unit conversion:

[g] = [kg/h] * [ms] * [(g*h) / (kg*ms)] / 3600

has to be considered.

The fraction of the formula above is minimized to the resolution of its variable (FAC_O2L_RLS_CAT_PURGE[i]). The oxygen loading that is added each sample step is also minimized to the resolution of its variable (O2L_RLS_ADD_CAT_PURGE[i]).

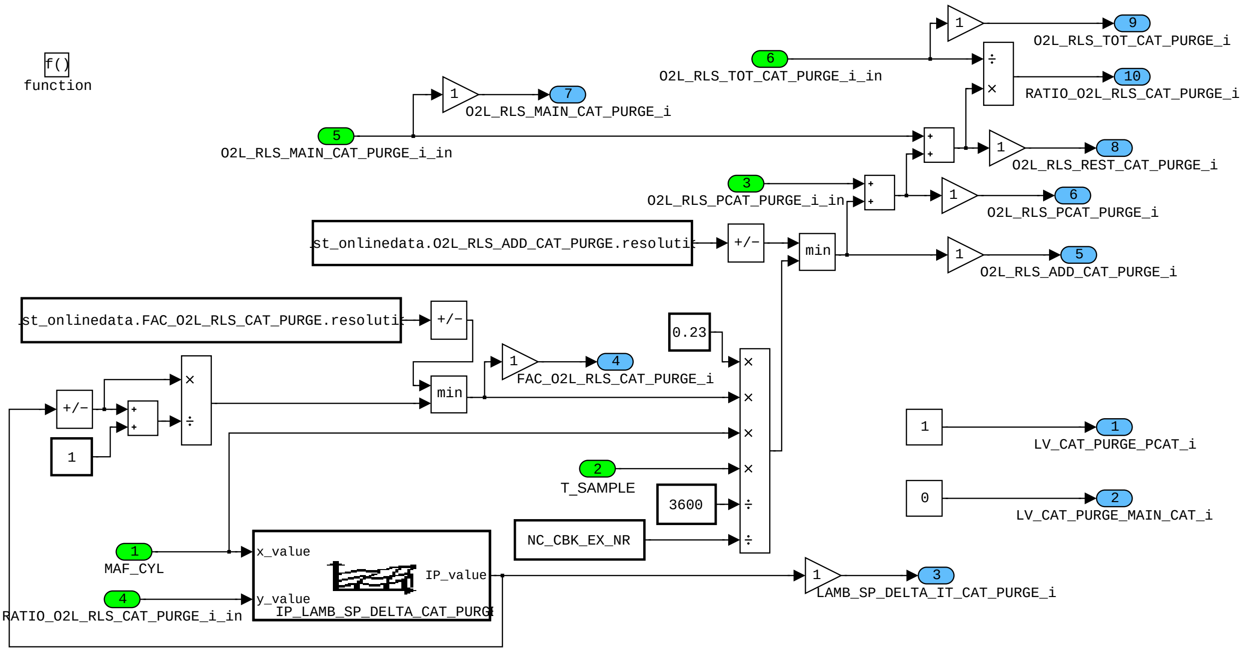


Figure 50 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ PRE_CAT_PURGE/ PRE_CAT_PURGE_CLC

End of pre catalyst purge

Main cat purge is activated by setting the respective flag. Pre cat purge is deactivated.

By newly evaluating the map IP_FAC_MAIN_CAT_PURGE the oxygen loading to be released during main cat purge is updated. The rest and total oxygen loading release is renewed and the quotient: rest / total oxygen loading release is also updated.

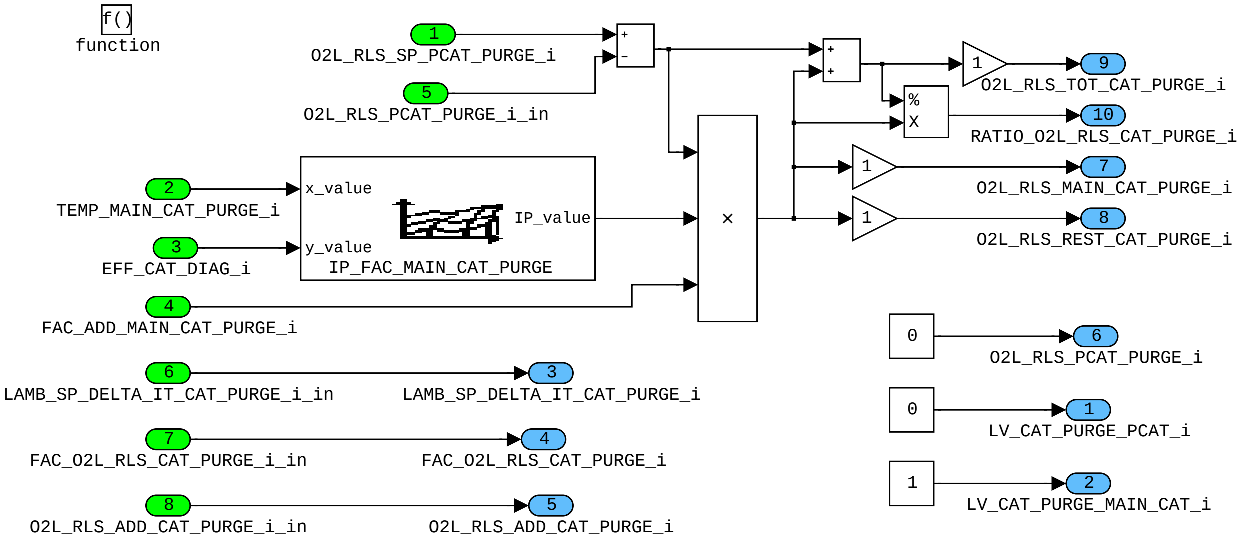
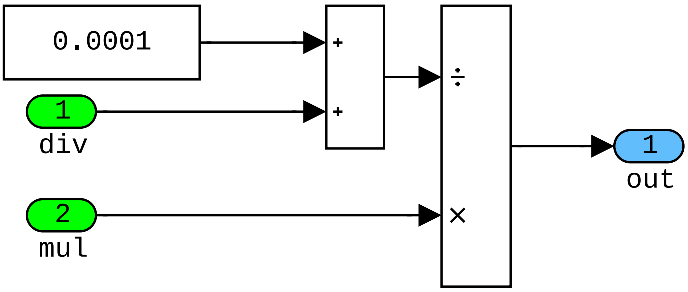


Figure 51 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ PRE_CAT_PURGE/ PRE_CAT_PURGE_END



Normal division delivers two warning ’division by zero’ during initialization, although subsystem is never triggered.

Figure 52 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_CBK_SPC/ PRE_CAT_PURGE/ PRE_CAT_PURGE_END/ SubSystem

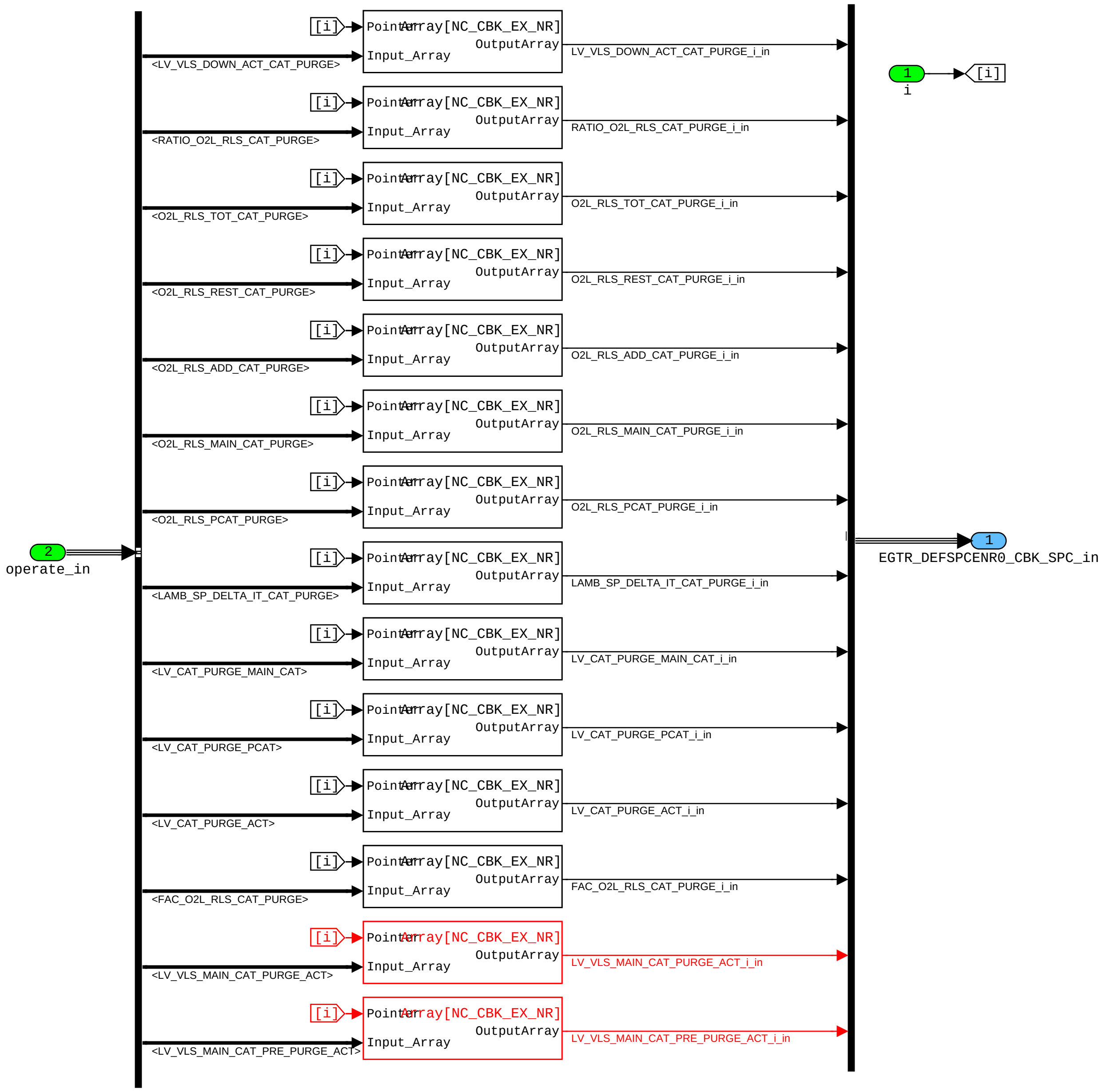


Figure 53 EGTR_DEFSPCENR0/ operate/ EGTR_DEFSPCENR0_CBK_MNG/ EGTR_DEFSPCENR0_SIG_CBK_MNG

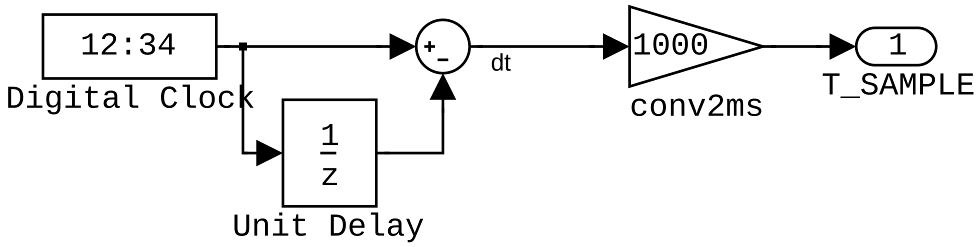


Figure 54 EGTR_DEFSPCENR0/ operate/ T_SAMPLE

18.6.1.2 SUBFUNCTION: reset

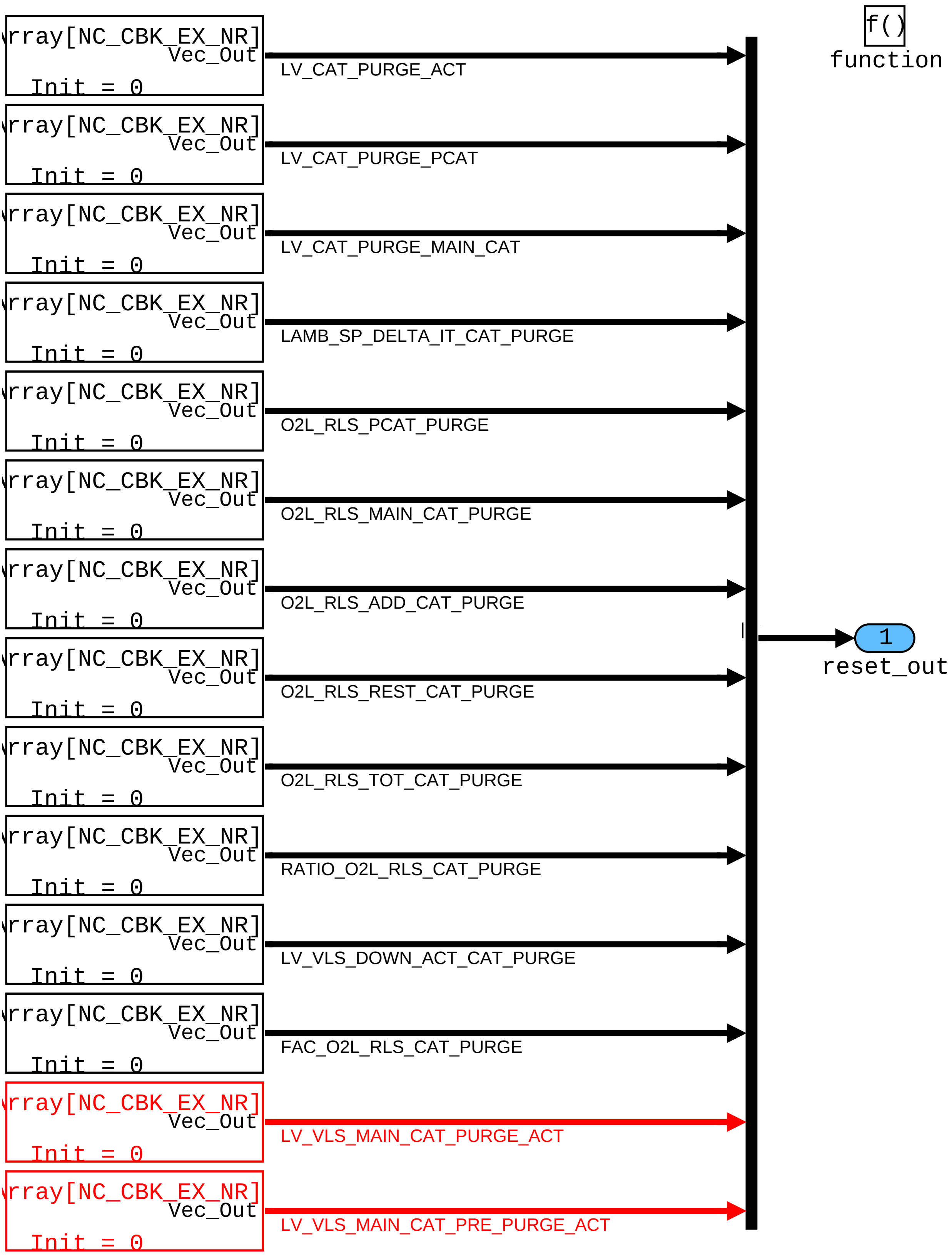


Figure 55 EGTR_DEFSPCENR0/ reset

18.7 Coordination of catalyst enrichment function

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_SP_DELTA_CAT_PURGE[NC_CBK_EX_NR]	O/V	0...FFFH	0...0.249939	6.10352E-5	-
lambda set point shift of cat pruge function					

Input data:

LAMB_SP_DELTA_IT_CAT_PURGE[NC_CBK_EX_NR]	NC_CBK_EX_NR	LAMB_SP_DELTA_ADD_CAT_PURGE[NC_CBK_EX_NR]	

18.7.1 EGTR_DEFSPCENRC0

This function coordinates the lambda set point rich shift of different functions. Up to now only the ouput of the catalyst enrichment function is considered.

NC_CBK_EX_NR defines the number of exhaust banks.

Application Condition

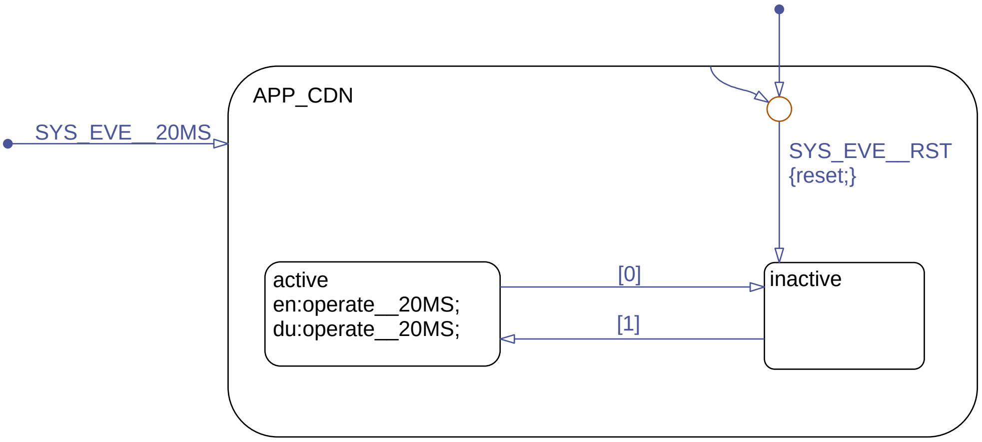


Figure 56 EGTR_DEFSPCENRC0/ APP_CDN/ Chart

Function Description

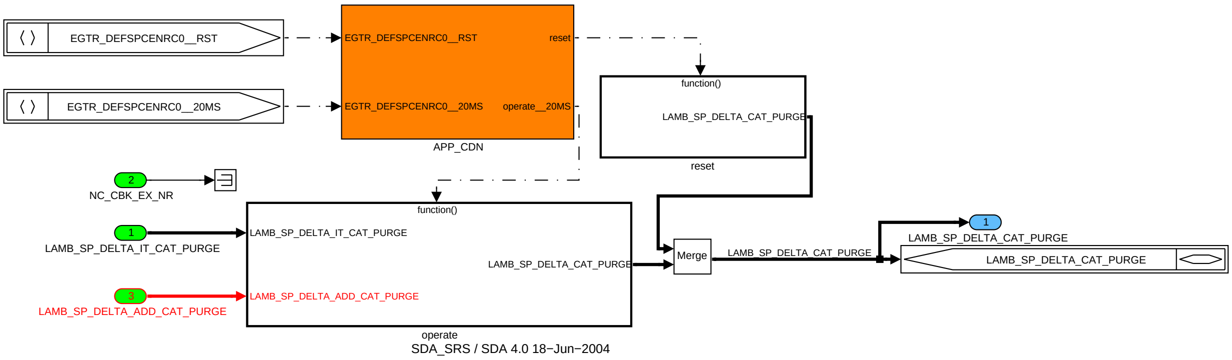


Figure 57 EGTR_DEFSPCENRC0

18.7.1.1 operate

The assignment of LAMB_SP_DELTA_CAT_PURGE[i] is a vector operation.

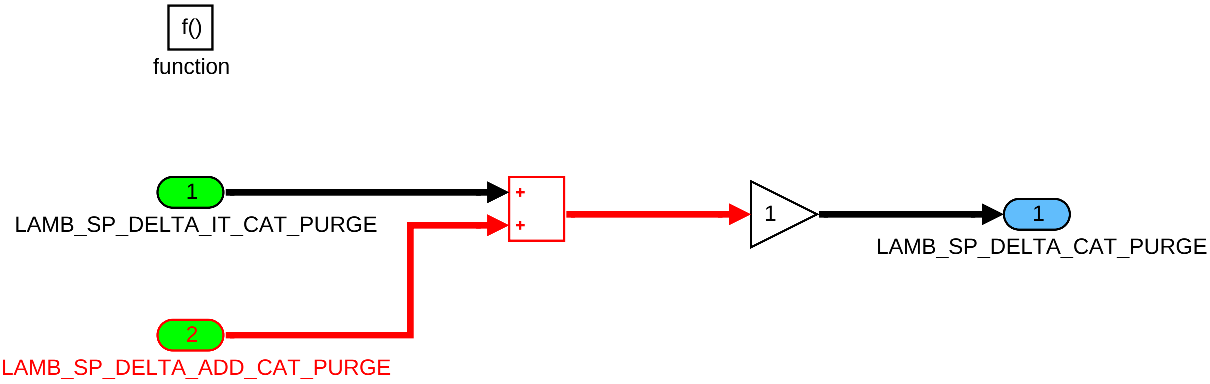


Figure 58 EGTR_DEFSPCENRC0/ operate

18.7.1.2 SUBFUNCTION: reset

Figure 59 EGTR_DEFSPCENRC0/ reset

18.8 Application incidences for catalyst enrichment function

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LAMB_SP_DELTA_ADD_CAT_PURGE [NC_CBK_EX_NR]	O/V	0... FFFH	0... 0.249939	61.0352e-6	[-]
additional lambda set point shift for cat purge output					
LV_ERR_VLS_MAIN_CAT_PURGE [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Final diagnostic of the sensor after main cat					
LV_GS_SCC	O/V	0... 1H	0... 1	1	[-]
cylinder cut of during gear shift					
LV_INH_CAT_PURGE [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
inhibition flag for cat purge function					
LV_VLS_MAIN_CAT_PURGE_READY [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Logical variable for operative readiness of sensor(s) after main cat					
MAF_INT_GS_SCC	O/V	0... FFFFH	0... 2912.66666667	0.0444444	[g]
air mass flow integral at cylinder cut of during gear shift phase					
MAF_INT_PUC_ACT	O/V	0... FFFFH	0... 2912.66666667	0.0444444	[g]
air mass flow integral during pull cut off phase					
MAF_INT_PUC_NOT_ACT	O/V	0... FFFFH	0... 2912.66666667	0.0444444	[g]
air mass flow integral out of pull cut off phase					
RATIO_NTL_DEC_INT [NC_CBK_EX_NR]	V	0... FFFFH	0... 7.99987792969	122.07e-6	[-]
Ratio of NTL_DEC_INT of both NOx traps					
TEMP_MAIN_CAT_PURGE [NC_CBK_EX_NR]	O/V	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
catalyst temperature of main cat for cat purge function					
VLS_MAIN_CAT_PURGE [NC_CBK_EX_NR]	O/V	0... 3FFH	0... 4.9951171875	4.88281e-3	[V]
Sensor voltage measured after Main Catalyst (eg NOx sensor after NOx trap)					

Input Data:

MAF_CYL	NC_CBK_EX_NR	N_32	TCO
TEMP_CAT_DYN_MDL [NC_CBK_EX_NR]	VLS_NS [NC_NOX_SENS_CONF]	LV_CAT_PURGE_REQ_POST_AFL [NC_NT_NR]	NC_NT_NR
NC_NOX_SENS_CONF	LV_PUC	LV_GS	EFF_SCC_AV
LV_VLS_NS_AUTH [NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_BOFF	LV_ERR_NS_CAN_MSG_LOST [NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_1_HTP [NC_NOX_SENS_CONF]
LV_ERR_NS_OBD_1_VLS [NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_2 [NC_NOX_SENS_CONF]	NTL_DEC_INT [NC_NT_NR]	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_MAF_CYL_MAX_CAT_PURGE	1	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
maximum air mass flow threshold for cat purge activation					
C_N_MAX_CAT_PURGE	1	0... FFH	0... 8160	32	[rpm]
maximum engine speed for cat purge					

C_TCO_MIN_CAT_PURGE	1	0... FEH	-48... 142.5	0.75	[°C]
minimum TCO threshold for cat purge activation					
C_TEMP_DIF_PCAT_MAIN_CAT_PURGE [NC_CBK_EX_NR]	1	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
temperature difference between pre and main cat					
IP_FAC_PURGE_RATIO_NTL	8	0... FFH	0... 7.96875	0.03125	[-]
LDP_RATIO_NTL_DEC_INT_IP_FAC	8	0... FFFFH	0... 7.99987792969	122.07e-6	[-]
Correction of different regenerated NOx traps depending on ratio of NTL_DEC_INT					
IP_LAMB_ADD_PURGE	6*8	0... FFFH	0... 0.249939	61.0352e-6	[-]
LDP_MAF_CYL_IP_LAMB_ADD_PURGE	6	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
LDP_VLS_NOX_SENS_IP_LAMB_ADD	8	0... 578H	-200... 1200	1	[mV]
delta lambda map for cat purge after lean mode					
LC_CAT_PURGE_INH [NC_CBK_EX_NR]	1	0... 1H	0... 1	1	[-]
calibration flag to force cat purge function to be inhibited					
LC_NS_2_DEAC	1	0... 1H	0... 1	1	[-]
force usage of first NOx sensor for cat purge of both banks					

General Information

This function defines the general inhibition of catalyst purge and the temperature of the main catalyst by means of a temperature gradient.
NC_CBK_EX_NR defines the number of exhaust banks.

Additionally, MAFintegrals from INSY (MAF_INT_PUC_ACT and MAF_INT_PUC_NOT_ACT) and an MAFintegral for gear shifting with cylinder cutoff (MAF_INT_GS_SCC) are included.
An additionally interface to NOXM for cat purge after lean operation is included.

Application Conditions

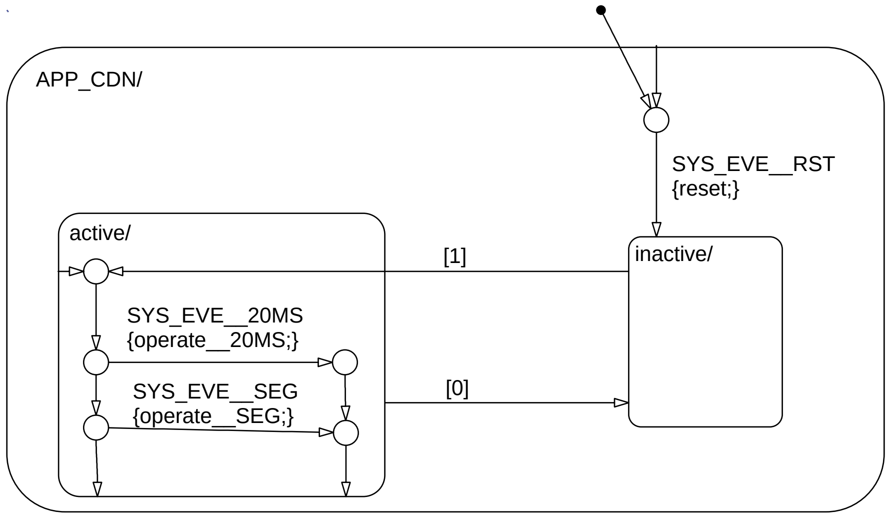
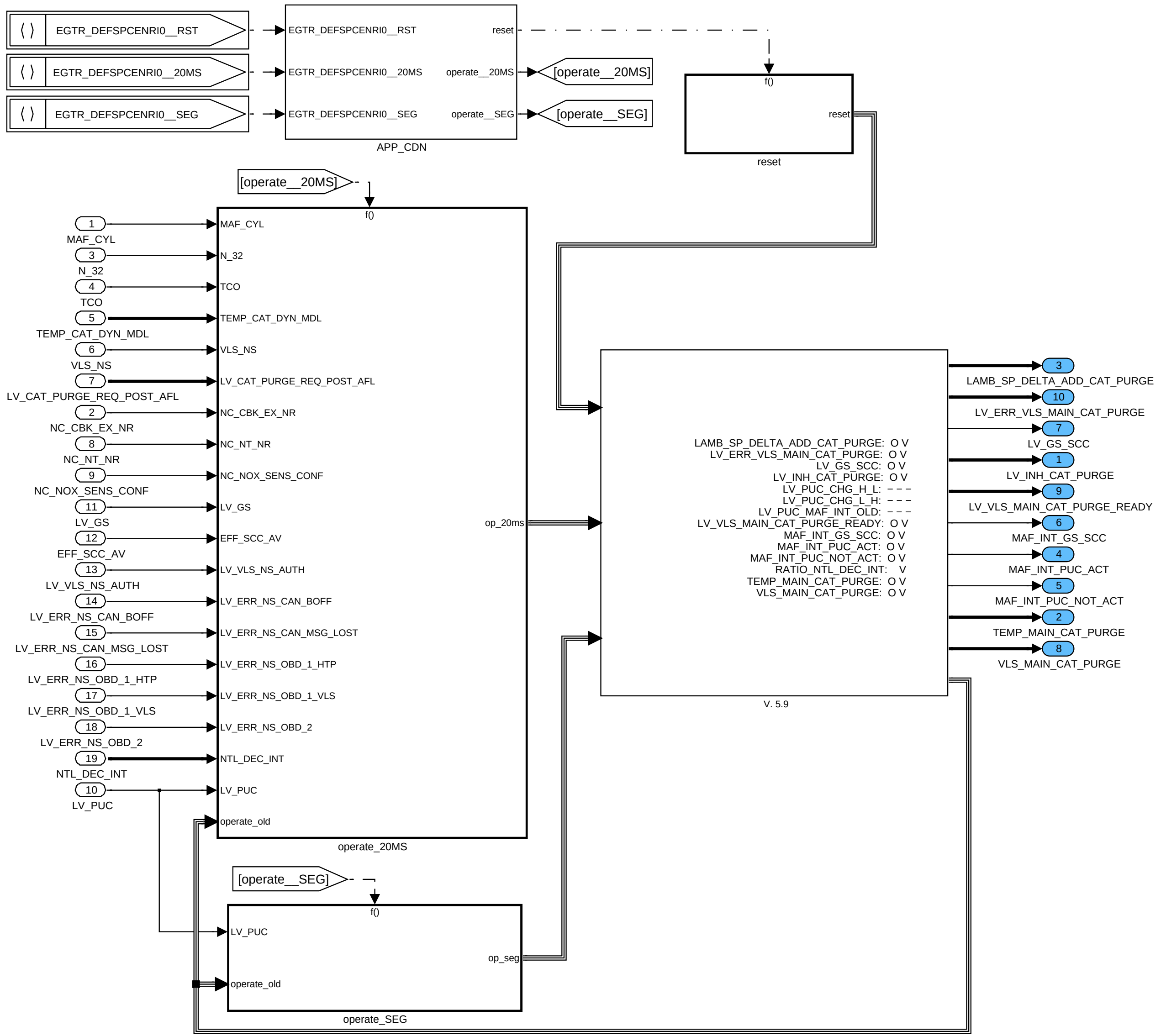


Figure 60:
Path: EGTR_DEFSPCENRI0/APP_CDN/Chart



SDA_SRS / SDA V 5.0.4 16−May−2006

Figure 61:
Path: EGTR_DEFSPCENRI0

18.8.1 SUBFUNCTION: operate_20MS

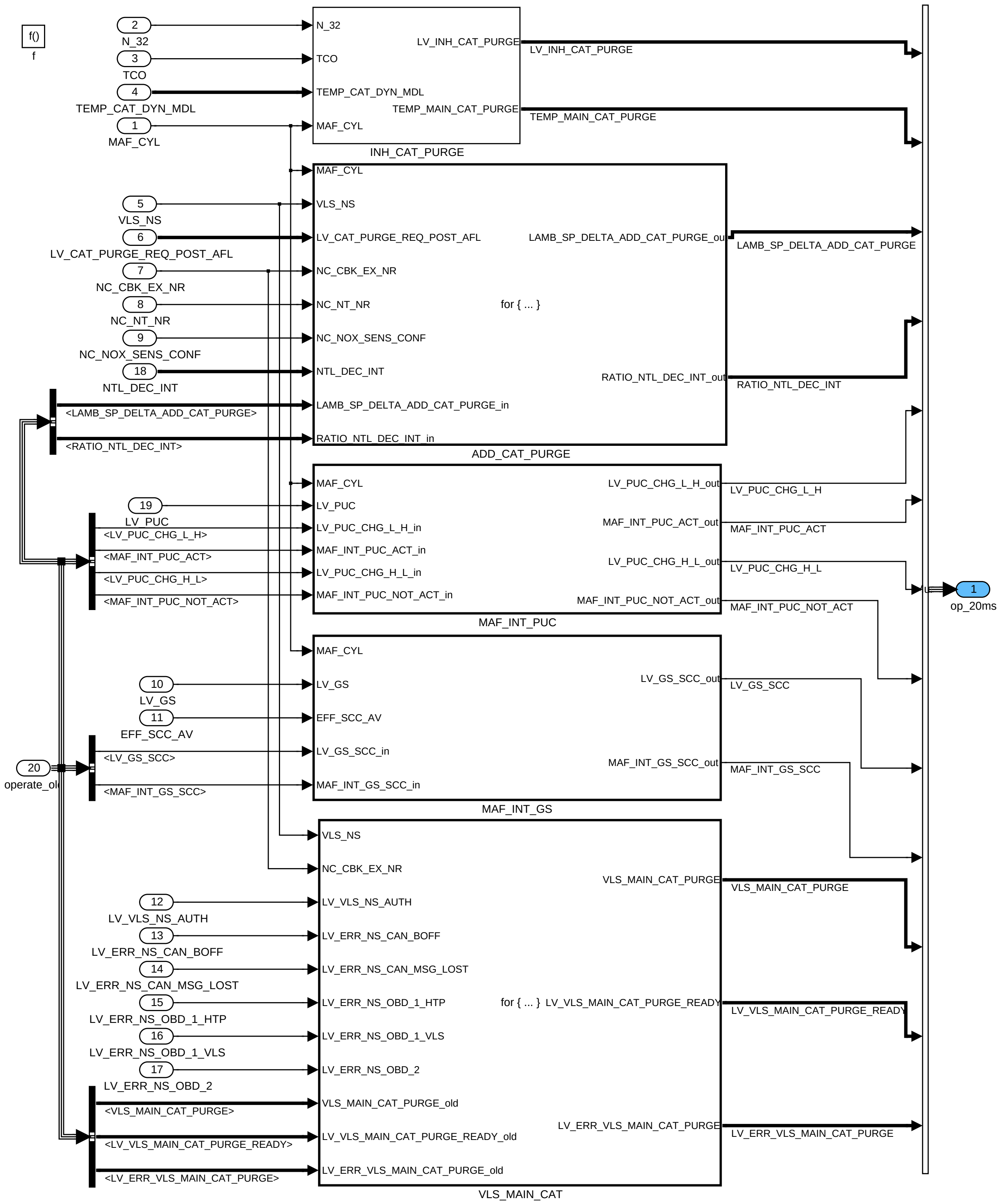


Figure 62:
Path: EGTR_DEFSPCENRI0/operate_20MS

18.8.1.1 LV_INH_CAT_PURGE and TEMP_MAIN_CAT_PURGE

LC_CAT_PURGE_INH[i] can inhibit each exhaust bank separately, while the threshold conditions of MAF_CYL, N_32 and TCO is valid for both banks in the same way. A vector operation must be applied.

The temperatur gradient calculation for the main catalyst is a vector operation too.

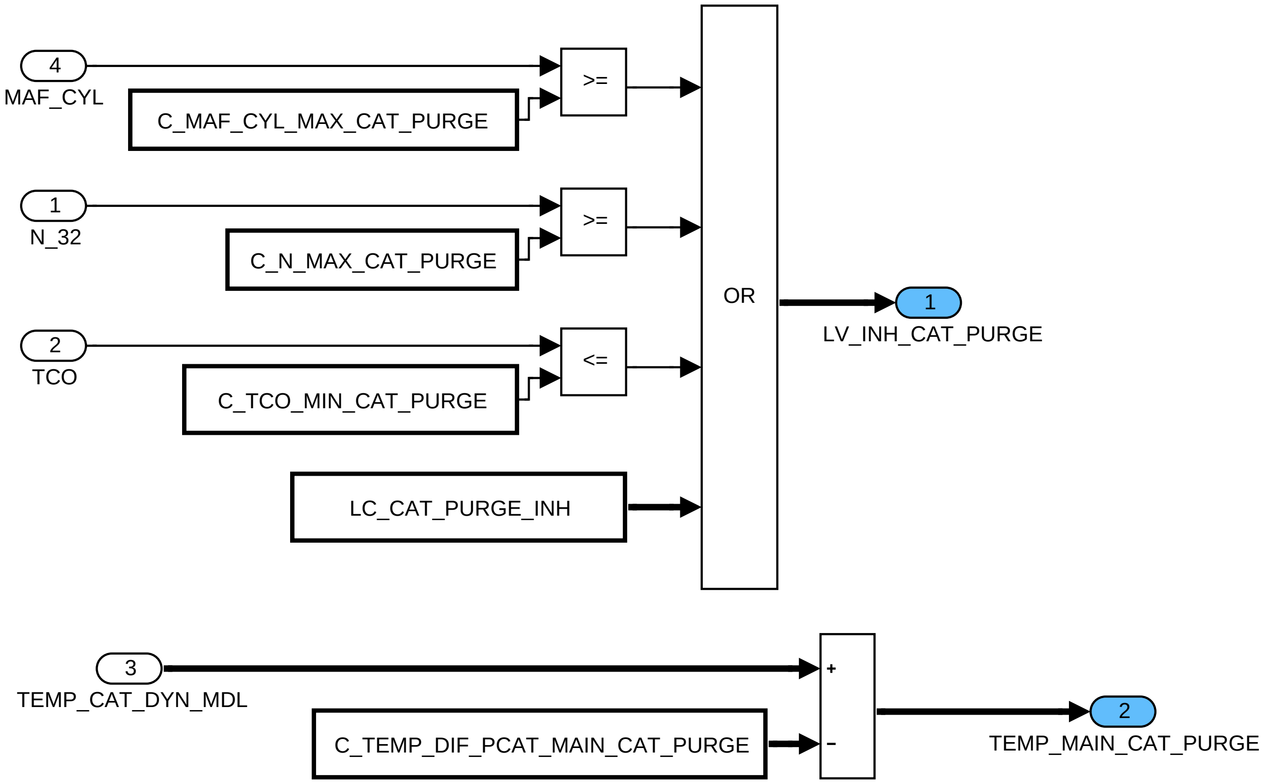


Figure 63:
Path: EGTR_DEFSPCENRI0/operate_20MS/INH_CAT_PURGE

18.8.1.2 LAMB_SP_DELTA_ADD_CAT_PURGE

This cat purge interface to NOXM is used after lean operation.
Vector size of input signals depend from NC_NT_NR and NC_NOX_SENS_CONF. Vector size of output signal depends from NC_CBK_EX_NR. Therefore the input signals have to be used dependend on different exhaust gas configurations. Especially, if NC_CBK_EX_NR is greater than the number of NOx sensors or NOx traps only the signal of the first bank have to be used for calculation of both banks.

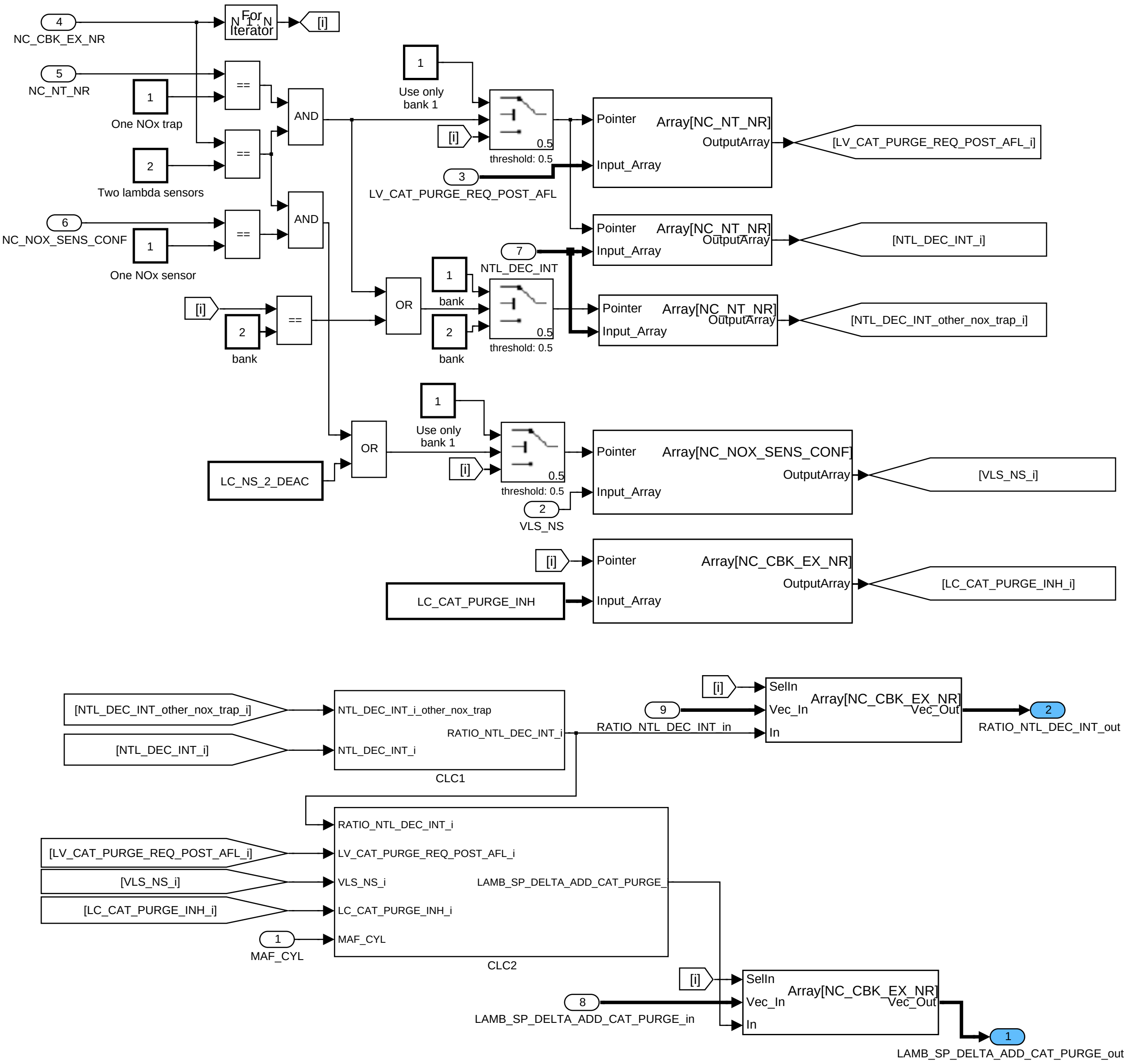


Figure 64:
Path: EGTR_DEFSPCENRI0/operate_20MS/ADD_CAT_PURGE

18.8.1.2.1 Ratio of NTL_DEC_INT of both NOx traps

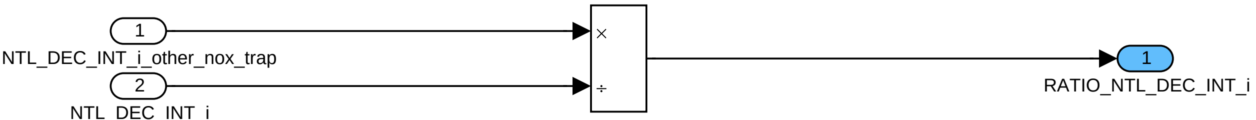


Figure 65:
Path: EGTR_DEFSPCENRI0/operate_20MS/ADD_CAT_PURGE/CLC1

18.8.1.2.2 Calculation of lambda set-point for purge

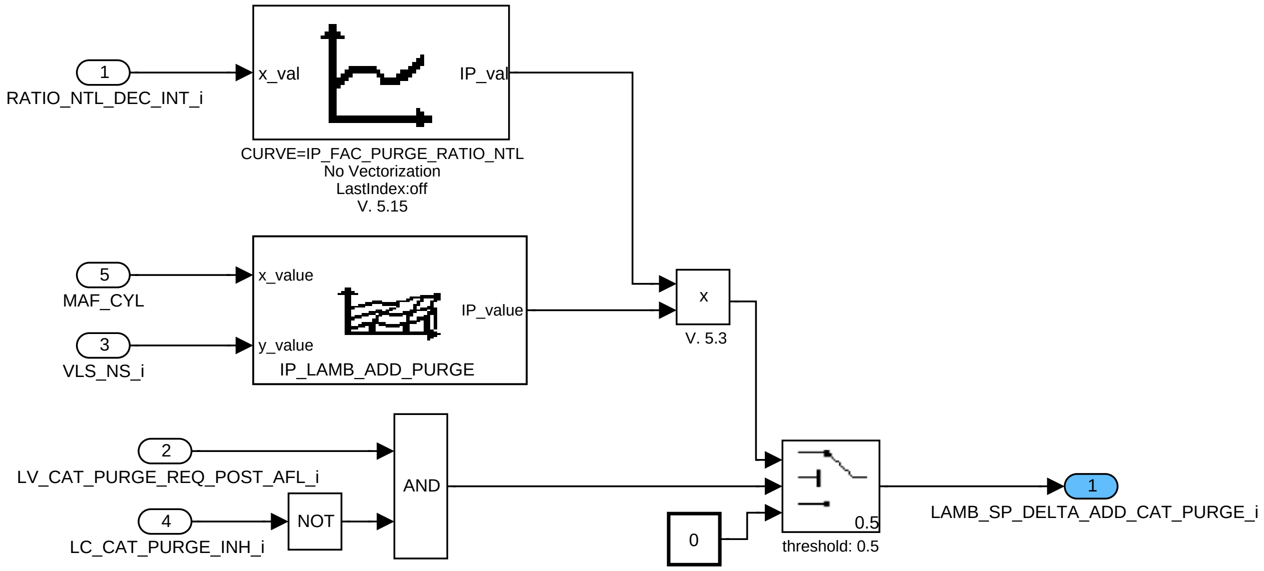


Figure 66:
Path: EGTR_DEFSPCENRI0/operate_20MS/ADD_CAT_PURGE/CLC2

18.8.1.3 MAF_INT_PUC_ACT and MAF_INT_PUC_NOT_ACT

Calculation of MAF integrals is done depending on LV_PUC, the reset of the integrals is done depending on edges of LV_PUC.

These edges will be detected in the synchronous segment loop and stored in LV_PUC_CHG_L_H or _H_L, to avoid loosing edges at high speed. After reset of an integral the edge memory flag will be cleared.

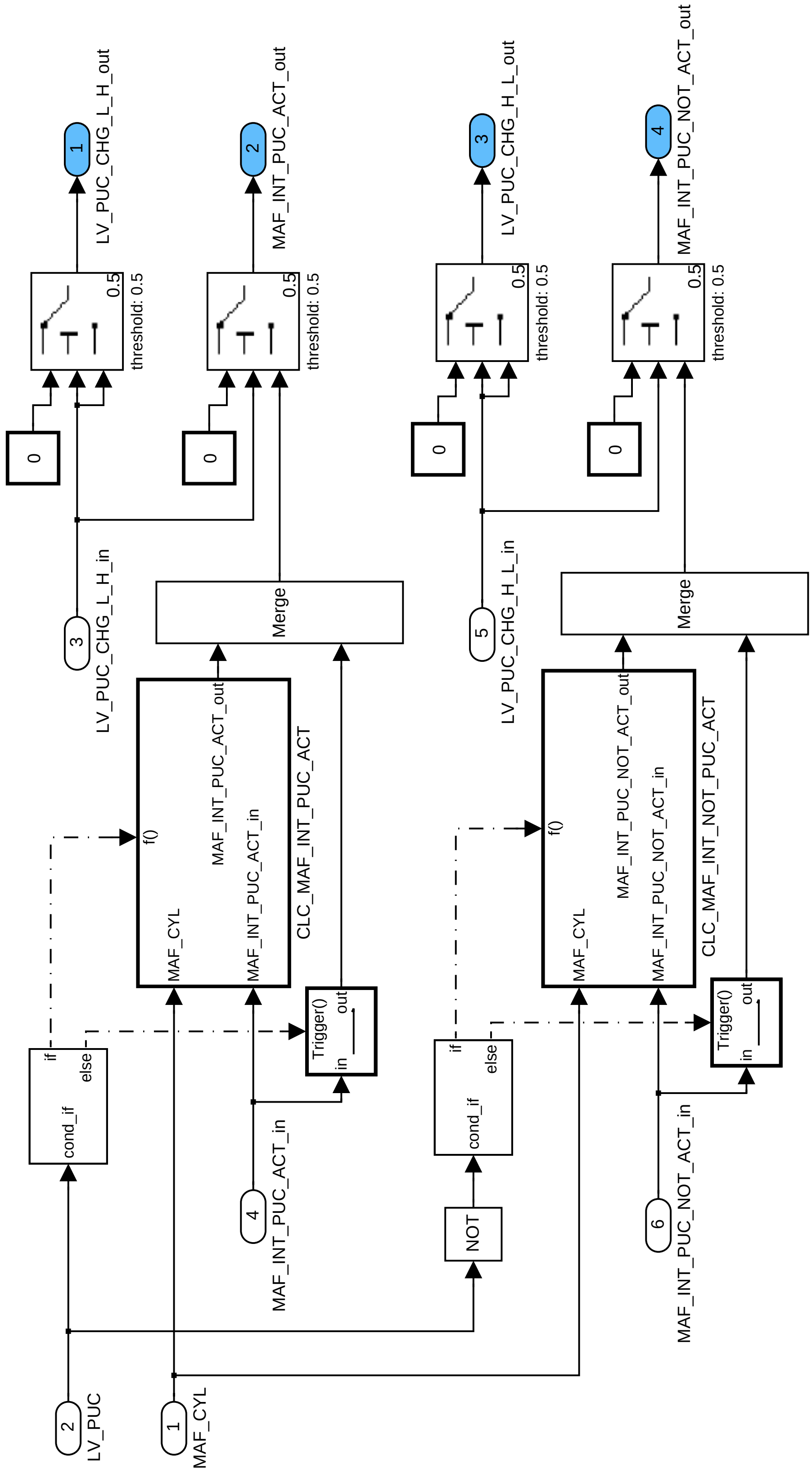


Figure 67:
Path: EGTR_DEFSPCENRI0/operate_20MS/MAF_INT_PUC

18.8.1.3.1 Calculation of MAF_INT_PUC_NOT_ACT

remark: MAF_INT_PUC_ACT have to be calculated internally on 32 bit

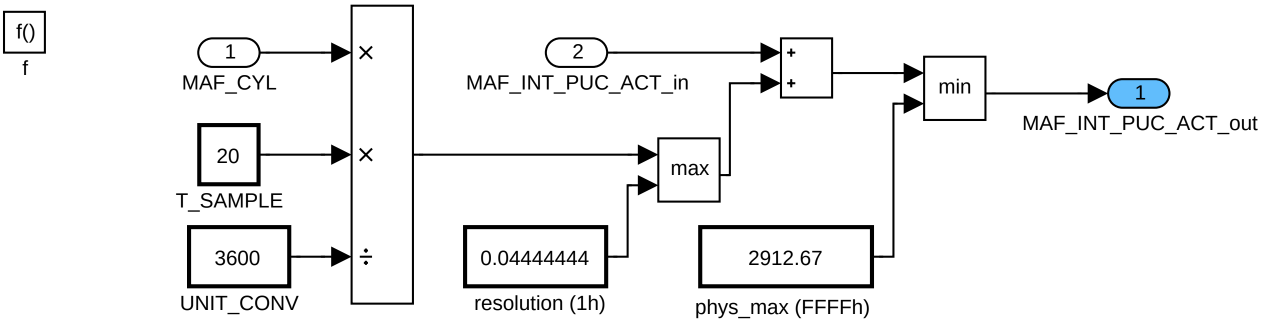


Figure 68:
Path: EGTR_DEFSPCENRI0/operate_20MS/MAF_INT_PUC/CLC_MAF_INT_PUC_ACT

18.8.1.3.2 Calculation of MAF_INT_PUC_ACT

remark: MAF_INT_NOT_PUC_ACT have to be calculated internally on 32 bit

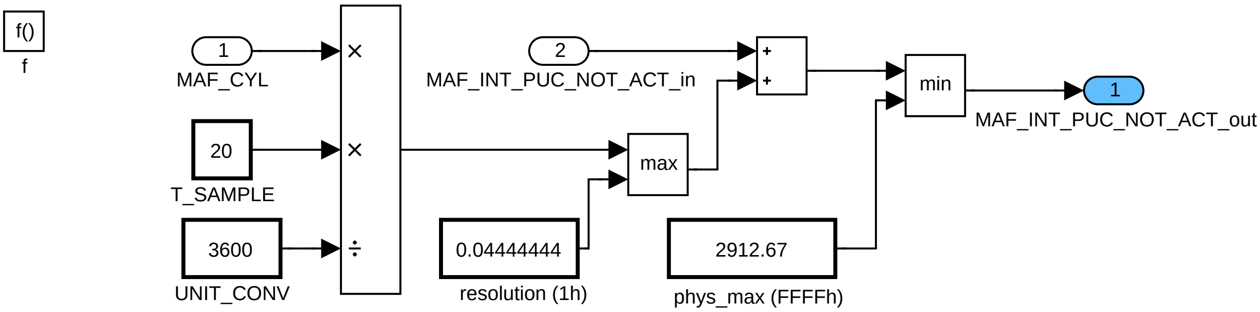


Figure 69:
Path: EGTR_DEFSPCENRI0/operate_20MS/MAF_INT_PUC/CLC_MAF_INT_NOT_PUC_ACT

18.8.1.4 LV_GS_SCC and MAF_INT_GS_SCC

A MAF integral is calculated as long as cylinder cutoff at gear shifting is active.

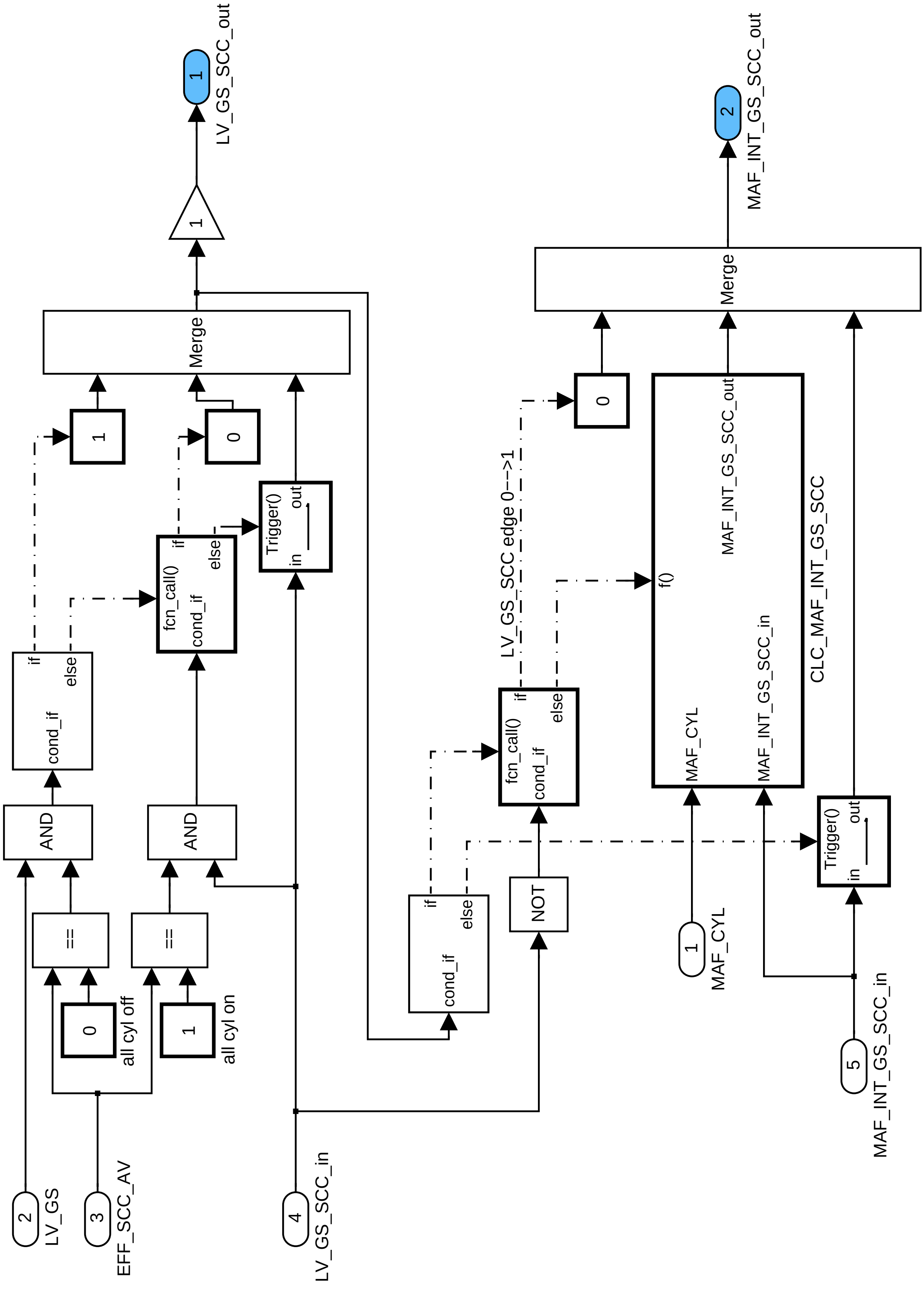


Figure 70:
Path: EGTR_DEFSPCENRI0/operate_20MS/MAF_INT_GS

18.8.1.4.1 Calculation of MAF_INT_GS_SCC

remark: MAF_INT_GS_SCC have to be calculated internally on 32 bit

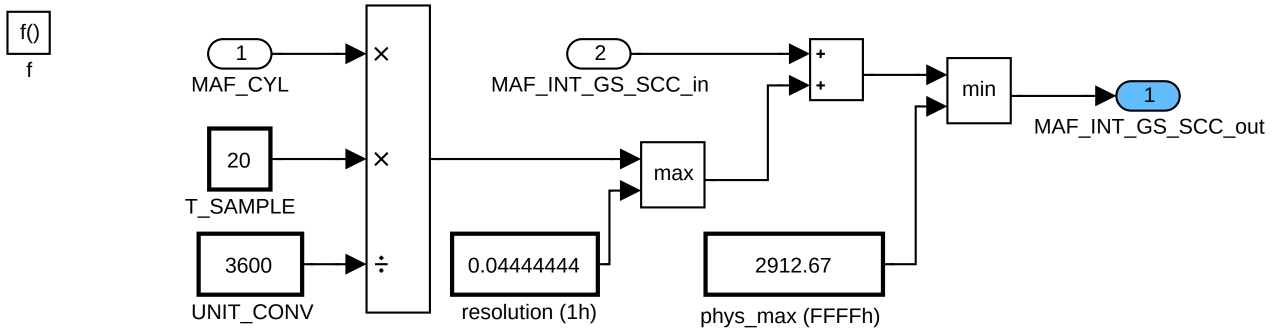


Figure 71:
Path: EGTR_DEFSPCENRI0/operate_20MS/MAF_INT_GS/CLC_MAF_INT_GS_SCC

18.8.1.5 Detection of cancelation of main catalyst purge

The binary lambda signal of the single NOx sensor will be used to cancel the purging of both main catalysts.

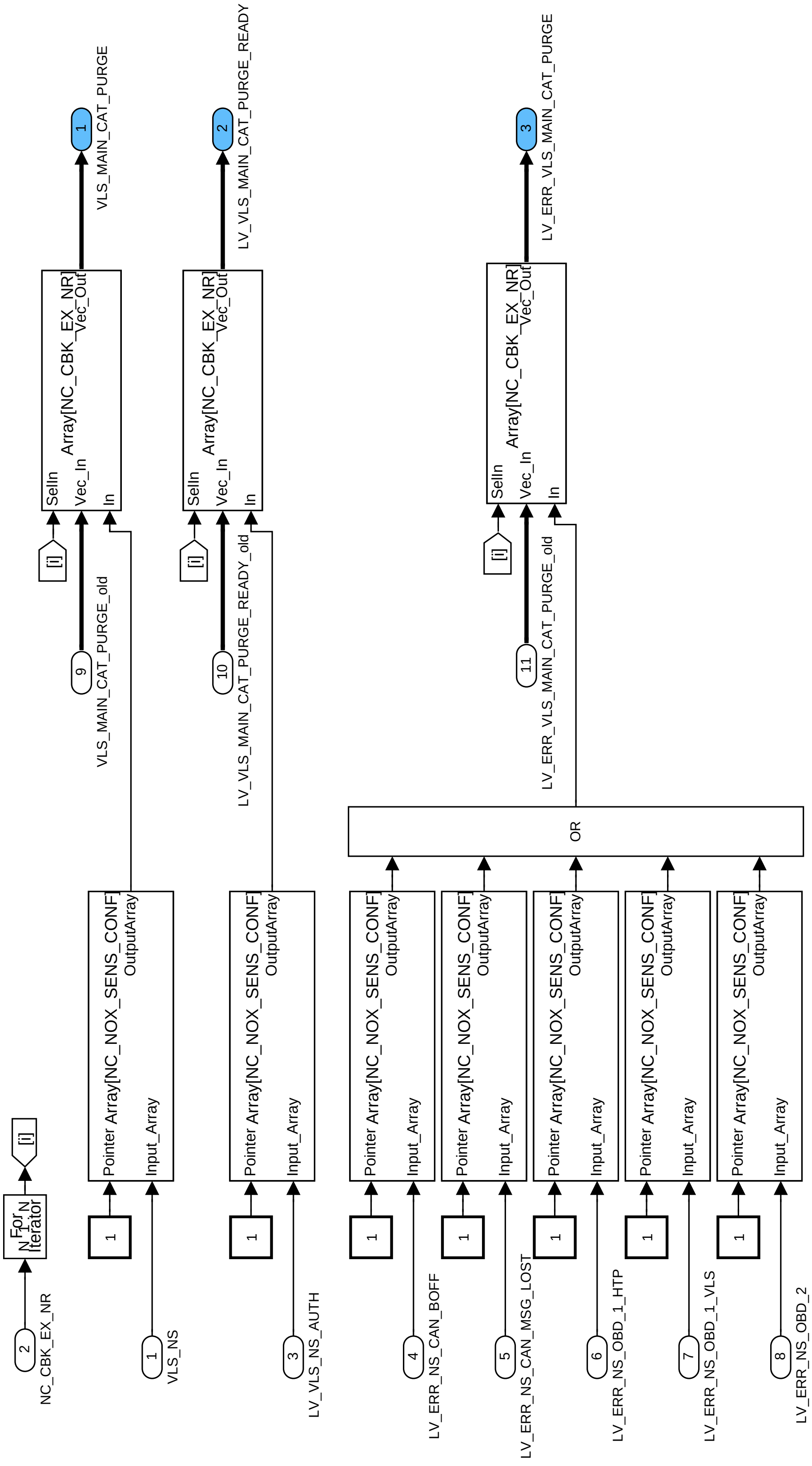


Figure 72:
Path: EGTR_DEFSPCENRI0/operate_20MS/VLS_MAIN_CAT

18.8.2 SUBFUNCTION: operate_SEG

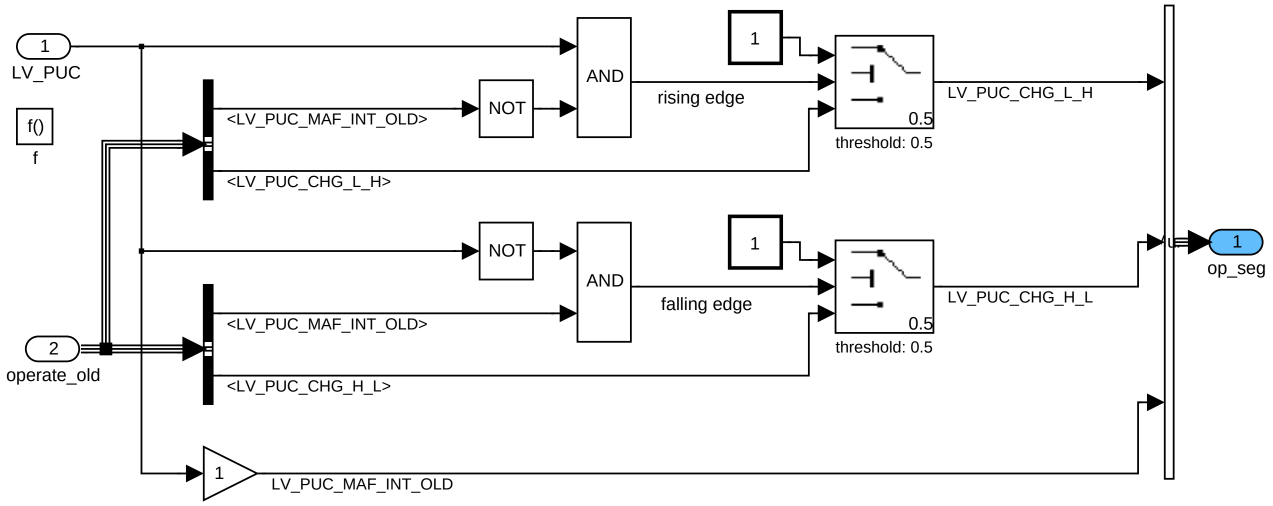


Figure 73:
Path: EGTR_DEFSPCENRI0/operate_SEG

18.8.3 SUBFUNCTION: reset

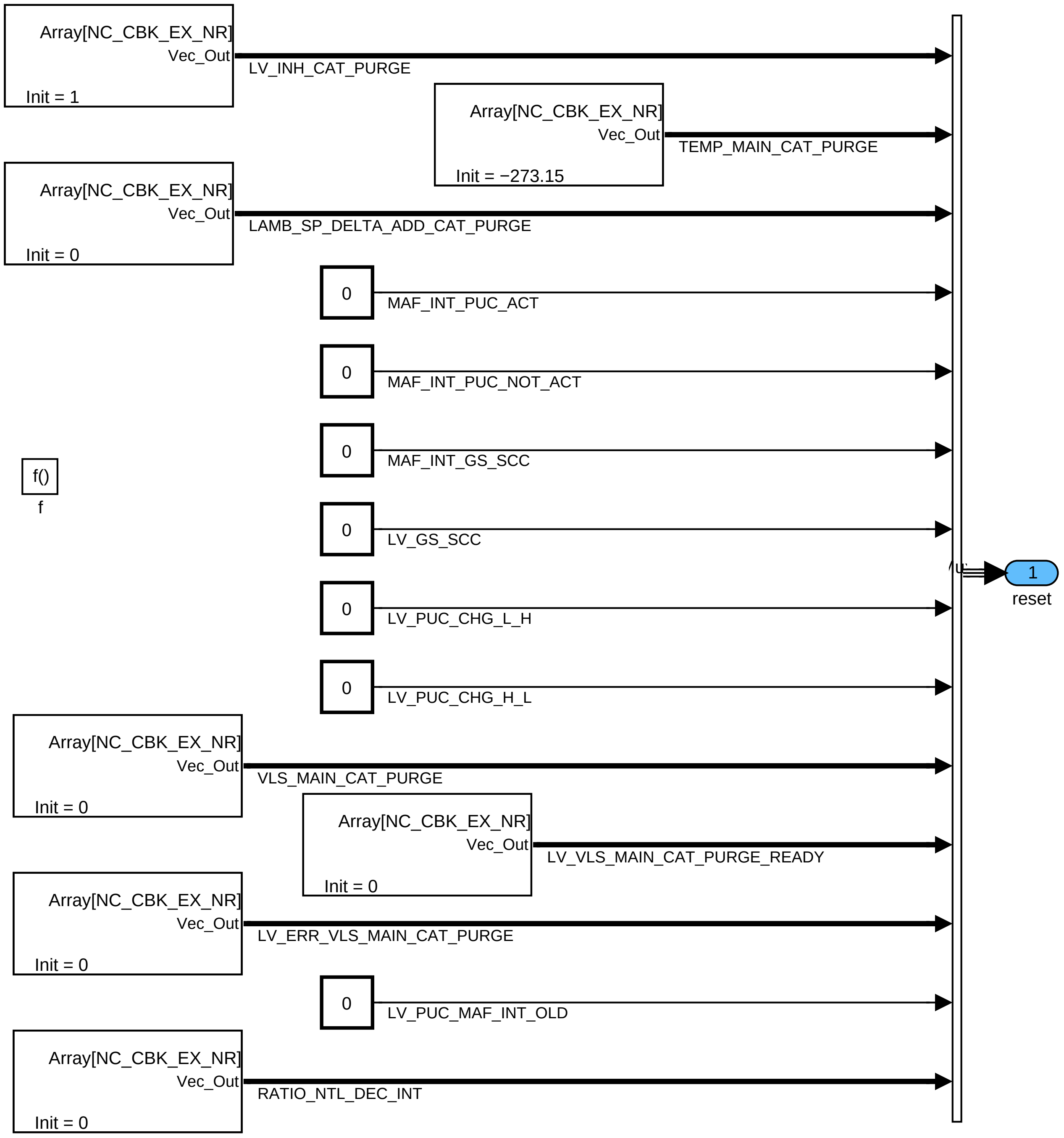


Figure 74:
Path: EGTR_DEFSPCENRI0/reset

18.9 Activation of forced lambda stimulation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAMB_PLS_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating that forced lambda stimulation is active					
LV_LAMB_PLS_DIAG_LSH_UP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating that requested exaggerated forced lambda stimulation is active					
LV_LAMB_PLS_ACT_CAT_PURGE[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag that indicates that the forced stimulation can be activated due to cat purge					
MAF_INT_LAMB_PLS_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...1.45633E+3	0.022222	g
air mass flow integral for forced lambda stimulation activation					

Input data:

LV_INH_LAMB_PLS[NC_CBK_EX_NR]	LAMB_SP_DELTA_CAT_PURGE[NC_CBK_EX_NR]	LV_ST_END	LAMB_SP_HOM[NC_CBK_EX_NR]
LV_LAM_STOP[NC_CBK_EX_NR]	LV_LAM_LSCL[NC_CBK_EX_NR]	TEMP_CAT_DYN_MDL[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_NOT_STAT_CDN[NC_CBK_EX_NR]
LV_LAMB_PLS_REQ_DIAG_LSH_UP[NC_CBK_EX_NR]	LV_PUC	MAF_CYL	NC_CBK_EX_NR

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_SP_DE_THD_LAMB_PLS_OFF	1	0...FFFH	0...0.249939	6.1035E-5	-
upper threshold of cat purge deviation for de-activating the forced stimulation					
C_LAMB_SP_DE_THD_LAMB_PLS_ON	1	0...FFFH	0...0.249939	6.1035E-5	-
lower threshold of cat purge deviation for activating the forced stimulation					
C_LAMB_SP_MAX_LAMB_PLS_ACT	1	0...7FFFH	0...1.999939	6.1035E-5	-
maximum lambda set point for forced lambda stimulation activation					
C_LAMB_SP_MIN_LAMB_PLS_ACT	1	0...7FFFH	0...1.999939	6.1035E-5	
minimum lambda set point for forced lambda stimulation activation					
C_MAF_INT_MIN_LAMB_PLS_ACT	1	0...FFFFH	0...1.45633E+3	0.022222	g
threshold for the air mass flow integral for the activation of the forced stimulation					
C_TEMP_CAT_MIN_LAMB_PLS_ACT	1	0...7FFFH	-273.15 ... 1.77479E+3	0.0625	°C
minimum value of the catalyst temperature for the activation of the forced stimulation					

18.9.1 EGTR_DEFSPFSTIA0

The reccurence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements, the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

Function Description

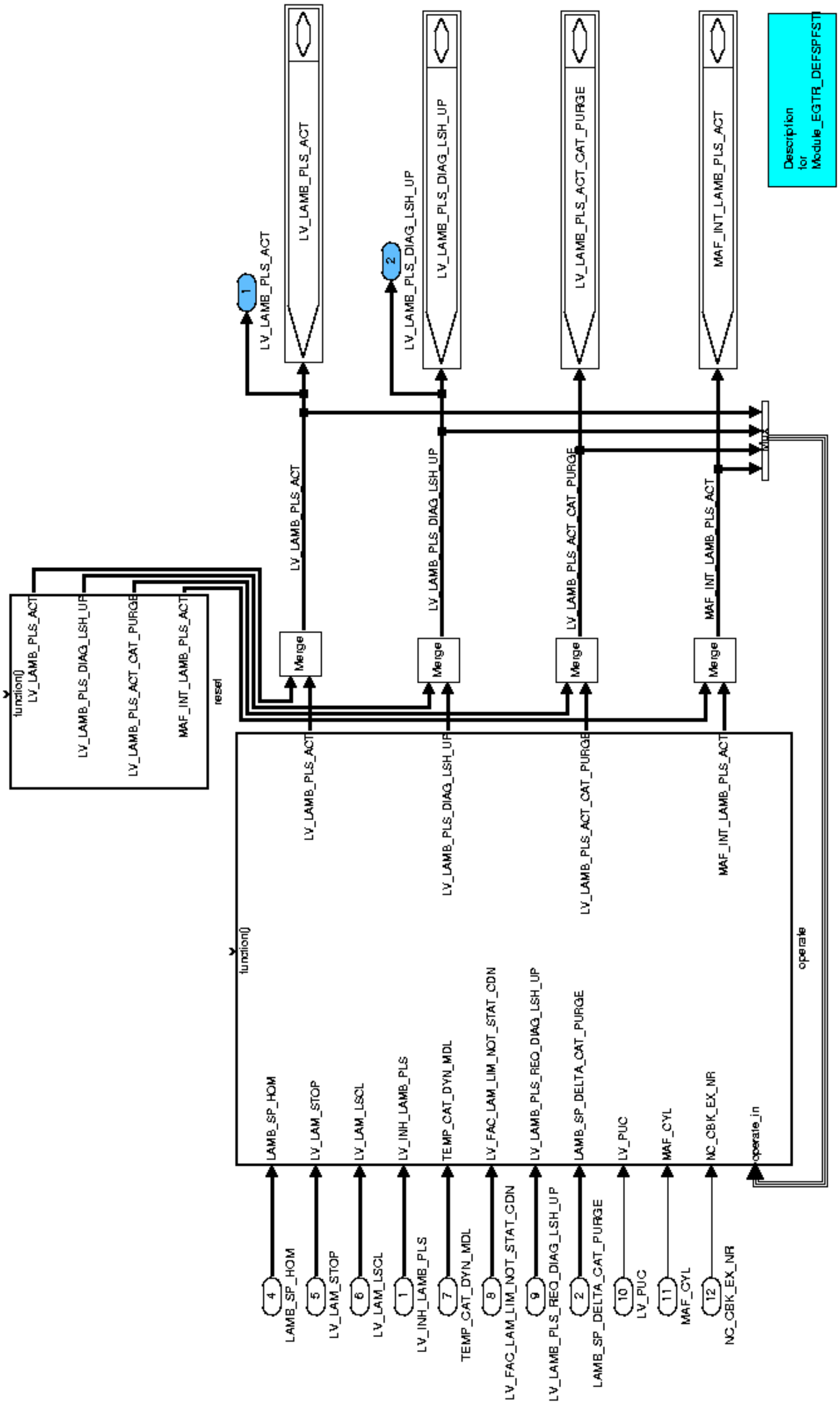


Figure 75 EGTR_DEFSPFSTIA0

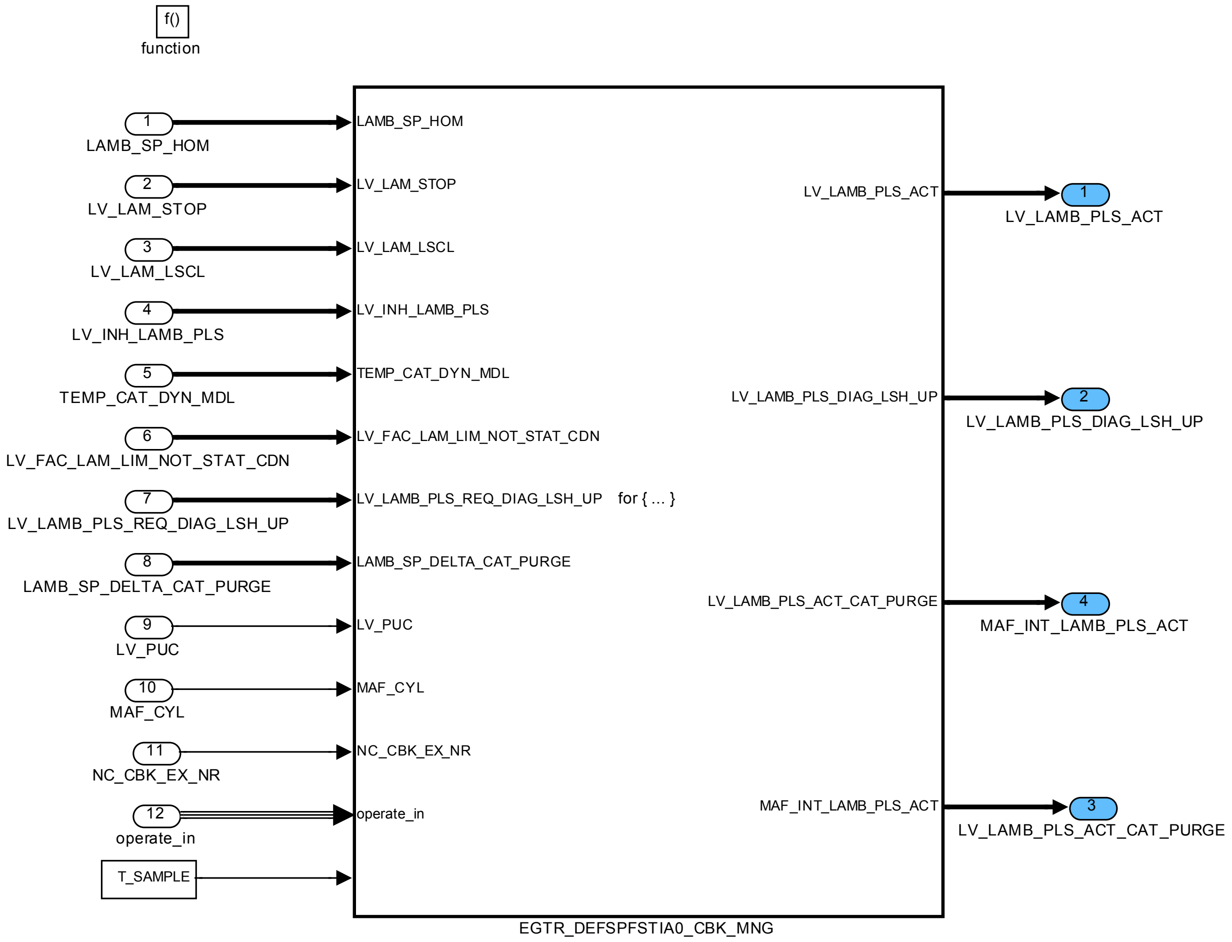


Figure 76 EGTR_DEFSPFSTIA0/ operate

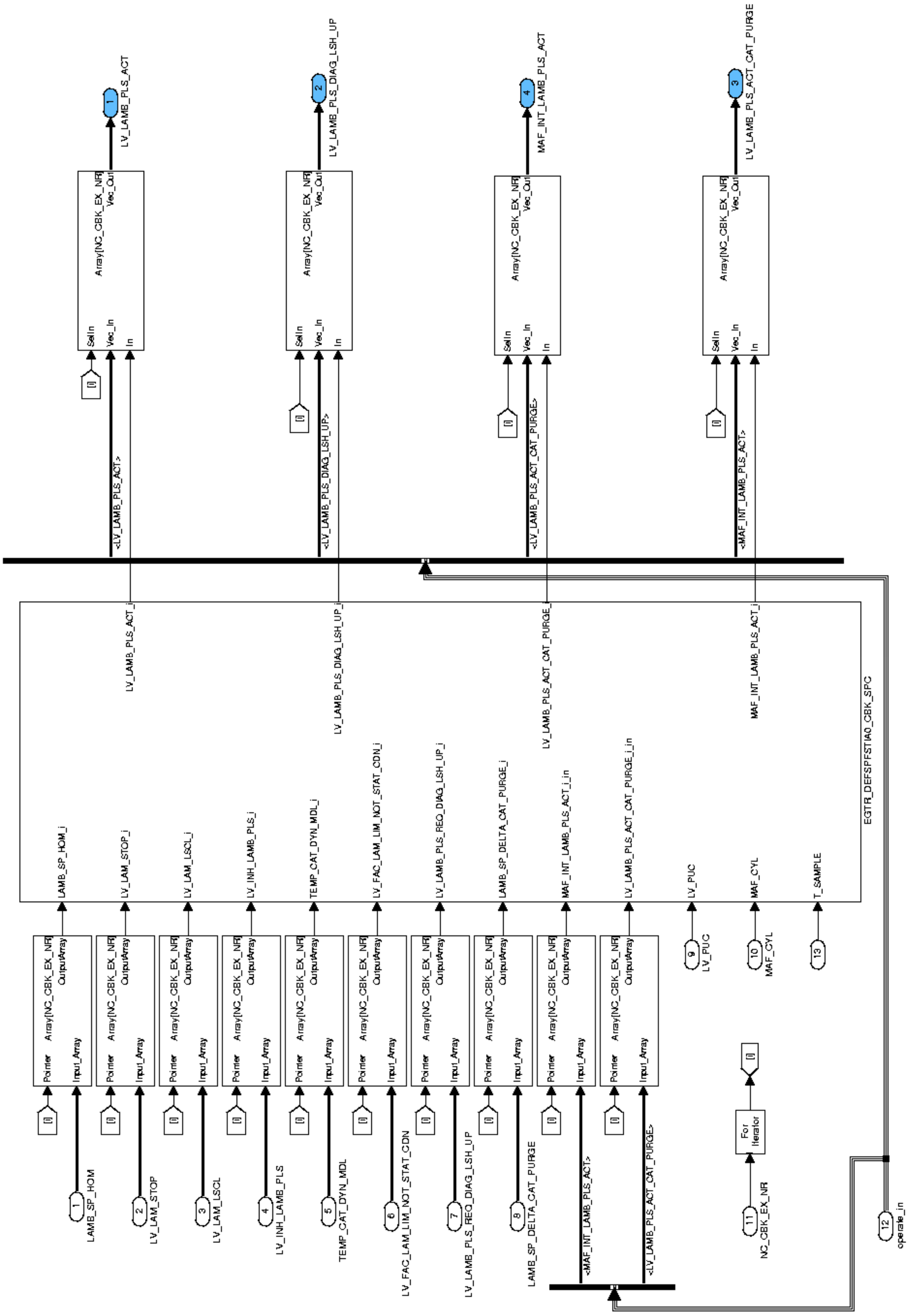


Figure 77 EGTR_DEFSPFSTIA0/ operate/ EGTR_DEFSPFSTIA0_CBK_MNG

Activation / deactivation

The forced stimulation is activated in requested mode if the following conditions are fulfilled:

- engine state pull fuel cut off is inactive

- no inhibition from application incidences
- the request has been recognised (LV_LAMB_PLS_REQ_DIAG_LSH_UP[i] = 1)

The activation in the requested mode has higher priority than the common mode.

The common mode is only activated if the activation condition for the requested mode are not fulfilled. For the activation in the common mode the following conditions must be fulfilled:

- engine state pull fuel cut off is inactive
- no inhibition from application incidences
- lambda controller is activated and not in stop mode
- no lambda controller limitation under non stationary operating conditions
- lambda set point deviation from catalyst purge is below calibration limit including hysteresis consideration (LV_LAMB_PLS_ACT_CAT_PURGE[i]); this flag is evaluated each time when the activation conditions are checked
- the lambda set point is in a valid range
- the catalyst temperature value exceeds a calibration threshold

If these conditions above are all fulfilled a mass air flow integral is calculated. The forced stimulation in the common mode will be finally activated when the integral exceeds a calibration limit.

For the calculation of the air mass flow integral the unit conversion:

[g] = [kg/h] * [ms] * [(g*h) / (kg*ms)] / 3600

has to be considered.

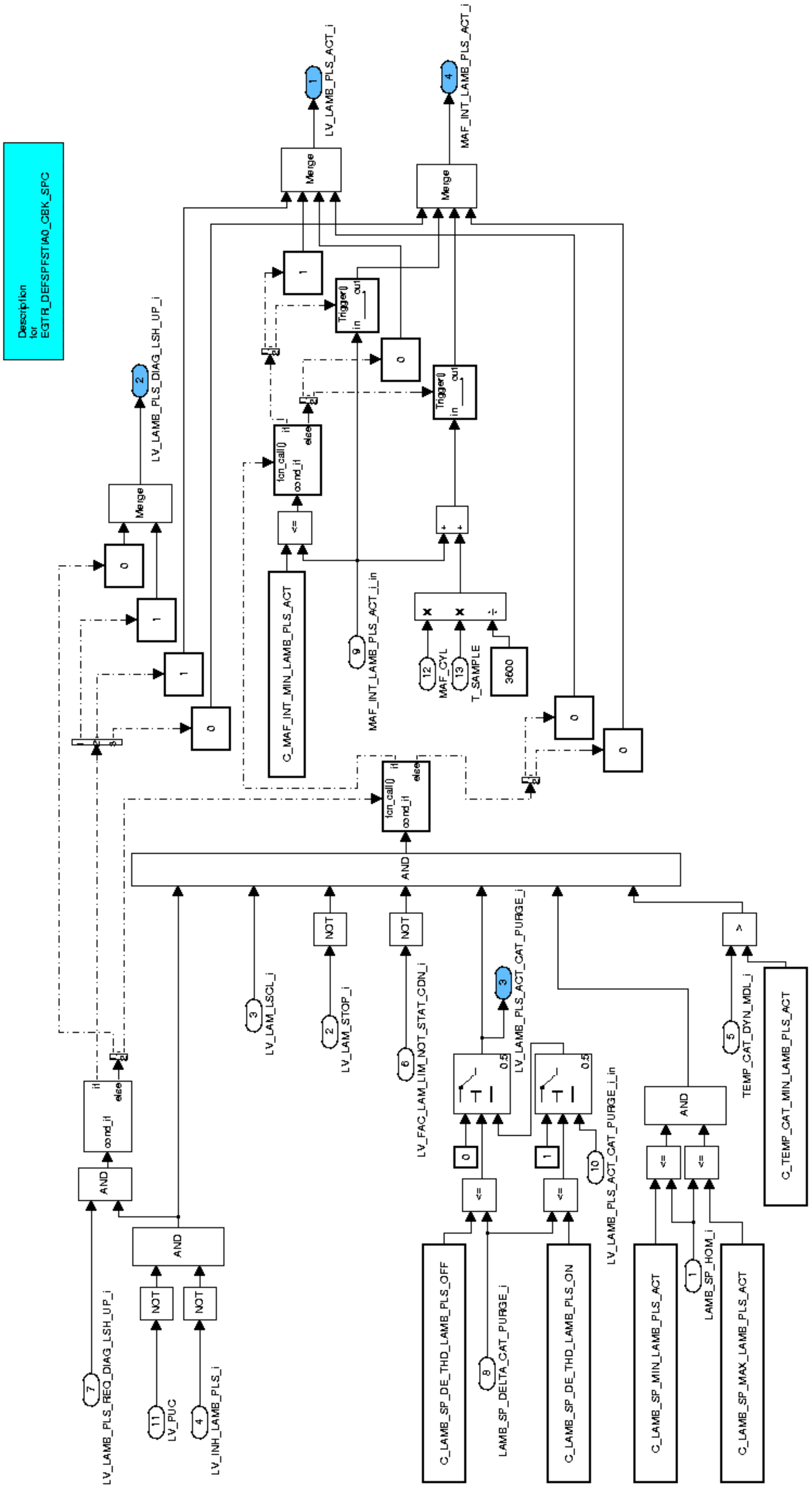


Figure 78 EGTR_DEFSPFSTIA0/ operate/ EGTR_DEFSPFSTIA0_CBK_MNG/ EGTR_DEFSPFSTIA0_CBK_SPC

18.10 Determination of forced lambda stimuation parameters

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_STEP_PLS_NOM_CAT_DIAG	O/V	0... 7H	0... 7	1	[-]
step counter to switch between nominal and cat diagnosis forced stimulation					
LAMB_DELTA_MAX_PLS	O/V	0... FFFH	0... 0.249939	61.0352e-6	[-]
lambda set point deviation step height per sample step					
LAMB_SP_DE_PLS	O/V	0... 7FFH	0... 0.12493896484	61.0352e-6	[-]
Amplitude of the forced lambda stimulation					
LV_STATE_LAMB_PLS_CBK_EQU	-	0... 1H	0... 1	1	[-]
auxiliary flag indicating that STATE_LAMB_PLS[i] is equal at both banks					
O2L_SP_PLS	O/V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
oxygen loading set point for forced lambda stimulation					
STATE_LAMB_PLS_DET_VALUE	O/V	0 1 2 3 4 5 6	PLS_OFF PLS_LSH_DIAG PLS_CAT_DIAG PLS_CAT_DIAG_TRAN PLS_DYN_DIAG PLS_IS PLS_NOM	-	[-]
State variable of forced lambda stimulation parameter determination					

Input Data:

LV_LAMB_PLS_ACT [NC_CBK_EX_NR]	LV_LAMB_PLS_DIAG_LSH_UP [NC_CBK_EX_NR]	LV_LAMB_PLS_SYN_CBK	STATE_LAMB_PLS [NC_CBK_EX_NR]
LV_LAMB_PLS_REQ_CAT_DIAG [NC_CBK_EX_NR]	EFF_CAT_DIAG [NC_CBK_EX_NR]	LV_IS	LV_LAMB_PLS_REQ_DYN_LSL_UP [NC_CBK_EX_NR]
TEMP_CAT_DYN_MDL [NC_CBK_EX_NR]	T_AST	N_32	MAF_HB
NC_CBK_EX_NR	LV_ST_END	LAMB_PLS_O2L_OSC [NC_CBK_EX_NR]	LC_O2L_PLS_CBK_SYN
TNT_MDL_H			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_LAMB_DELTA_MAX_IS_PLS	1	0... FFFH	0... 0.249939	61.0352e-6	[-]
lambda set point deviation step height per sample step for forced stimulation in case of idle speed					
C_LAMB_DELTA_MAX_LSH_DIAG_PLS	1	0... FFFH	0... 0.249939	61.0352e-6	[-]
lambda set point deviation step height per sample step for forced stimulation requested by sensor heater diagnosis					
C_LAMB_SP_DE_DYN_DIAG_PLS	1	0... 7FFH	0... 0.12493896484	61.0352e-6	[-]
amplitude of the forced lambda stimulation in case of active WRAF sensor dynamic diagnosis					
C_LAMB_SP_DE_IS_PLS	1	0... 7FFH	0... 0.12493896484	61.0352e-6	[-]
amplitude of forced stimulation in case of idle speed					
C_LAMB_SP_DE_LSH_DIAG_PLS	1	0... 7FFH	0... 0.12493896484	61.0352e-6	[-]
amplitude of forced stimulation requested by sensor heater diagnosis					
C_NR_STEP_PLS_NOM_CAT_DIAG	1	0... 7H	0... 7	1	[-]
Number of lean - rich steps to switch between nominal and cat diagnosis forced stimulation					

C_O2L_SP_IS_PLS	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
oxygen loading set point for forced lambda stimulation in case of idle speed					
C_O2L_SP_LSH_DIAG_PLS	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
oxygen loading set point for forced lambda stimulation requested by sensor heater diagnosis					
IP_FAC_1_LAMB_SP_DE_PLS	4	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_EFF_CAT_DIAG_IP_FAC_1_PLS	4	0... FFH	0... 1.9921875	7.8125e-3	[-]
first factor for forced stimulation amplitude (depending on maximal value of EFF_CAT_DIAG[i])					
IP_FAC_2_LAMB_SP_DE_PLS	4*4	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_T_AST_IP_FAC_2_PLS	4	0... FFFFH	0... 6553.5	0.1	[s]
LDP_TEMP_CAT_IP_FAC_2_PLS	4	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
second factor for forced stimulation amplitude(depending on time after start and cat temperature)					
IP_LAMB_DELTA_MAX_CAT_DIAG_PLS	8*8	0... FFFH	0... 0.249939	61.0352e-6	[-]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1389	5.4470588	[mg/stk]
lambda set point deviation step height per sample step for forced stimulation in case of active catalyst diagnosis					
IP_LAMB_DELTA_MAX_NOM_PLS	8*8	0... FFFH	0... 0.249939	61.0352e-6	[-]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1389	5.4470588	[mg/stk]
lambda set point deviation step height per sample step for nominal forced stimulation					
IP_LAMB_SP_DE_CAT_DIAG_PLS	8*8	0... 7FFH	0... 0.12493896484	61.0352e-6	[-]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1389	5.4470588	[mg/stk]
amplitude of forced stimulation in case of active catalyst diagnosis					
IP_LAMB_SP_DE_NOM_PLS	8*8	0... 7FFH	0... 0.12493896484	61.0352e-6	[-]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1389	5.4470588	[mg/stk]
amplitude of nominal forced stimulation					
IP_O2L_SP_CAT_DIAG_PLS	8*8	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1389	5.4470588	[mg/stk]
oxygen loading set point for forced lambda stimulation in case of active catalyst diagnosis					
IP_O2L_SP_DYN_DIAG_PLS	8*8	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1389	5.4470588	[mg/stk]
oxygen loading set point for forced lambda stimulation in case of active dynamic WRAF sensor diagnosis					
IP_O2L_SP_NOM_PLS	8*8	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1389	5.4470588	[mg/stk]
oxygen loading set point for nominal forced lambda stimulation					
IP_FAC_O2L_SP_PLS_TNT_MDL_H	8	0...FFH	0...1.99218	0.0078125	[-]
LDPM_TNT_MDL_H	8	0...FFFFH	0...1023.98437	0.015625	[°C]
oxygen loading set point for forced lambda stimulation dependent on NOX trap temperature					

General Information

NC_CBK_EX_NR defines the number of exhaust banks.

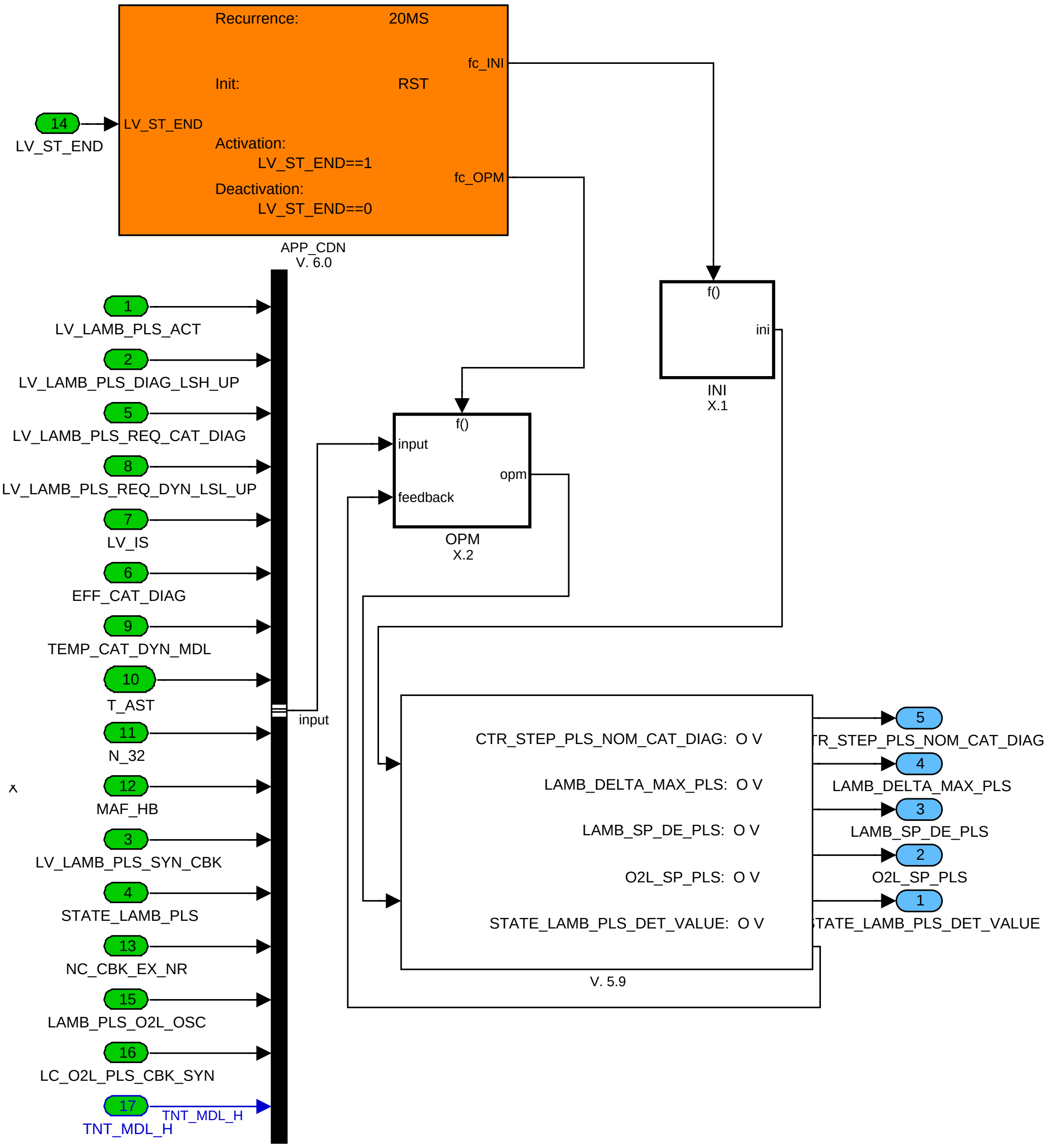
general specification

For vector elements, the variable extension _i is used in the model instead of [i] as found in the textual description.

Application Conditions

Initialization: RST
Recurrence: 20MS
Activation: LV_ST_END ==1
Deactivation: LV_ST_END ==0

Function description



SDA_SRS / SDA V 5.2 02−May−2007

Figure 79:
Path: EGTR_DEFSPFSTID0

18.10.1 Initialisation

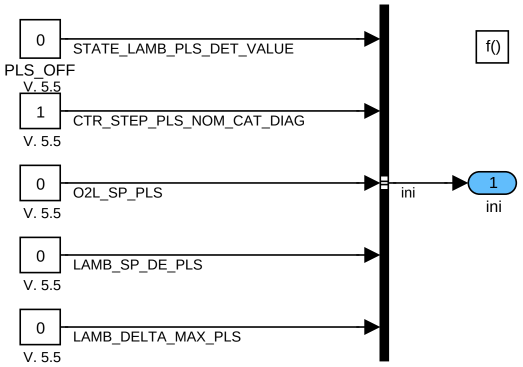


Figure 80:
Path: EGTR_DEFSPFSTID0/INI

18.10.2 Formula section

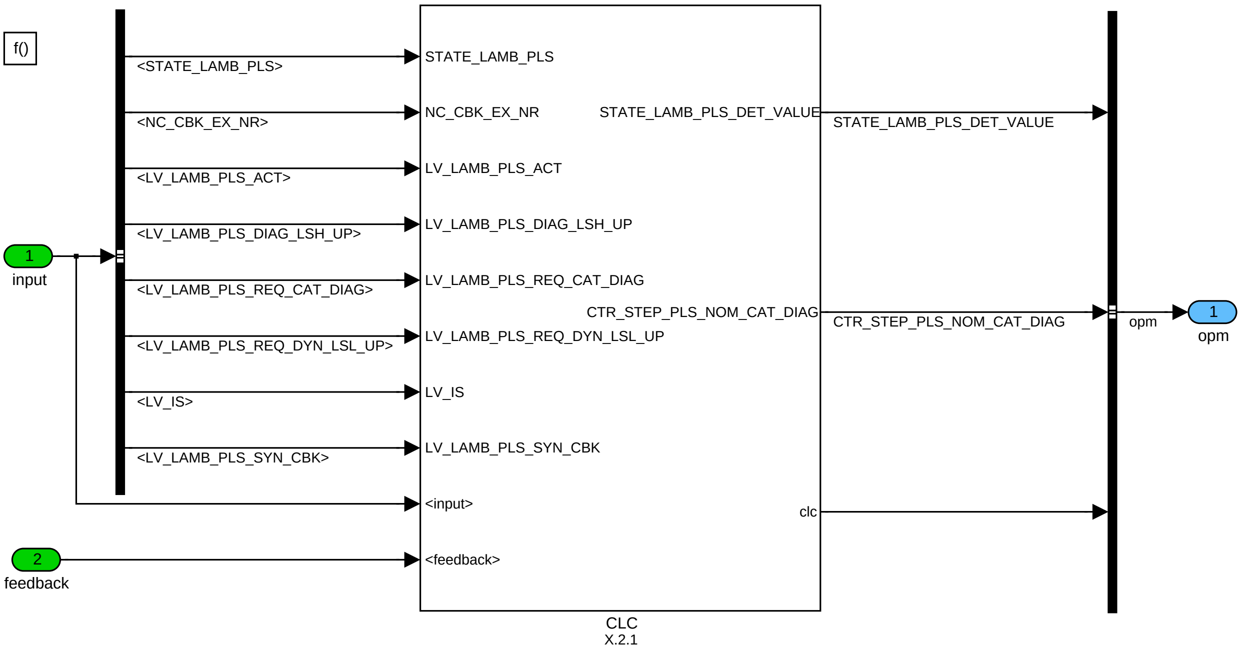


Figure 81:
Path: EGTR_DEFSPFSTID0/OPM

18.10.2.1 Calculations

The forced lambda stimulation distinguishes between 5 different parameters which are listed hereafter with its priority:

a) Priority 1: forced stimulation parameters for requested mode
b) Priority 2: forced stimulation parameters for catalyst efficiency diagnosis
c) Priority 3: forced stimulation parameters for WRAF sensor dynamic diagnosis
d) Priority 4: forced stimulation parameters for idle speed
f) Priority 5: forced stimulation parameters for normal operation

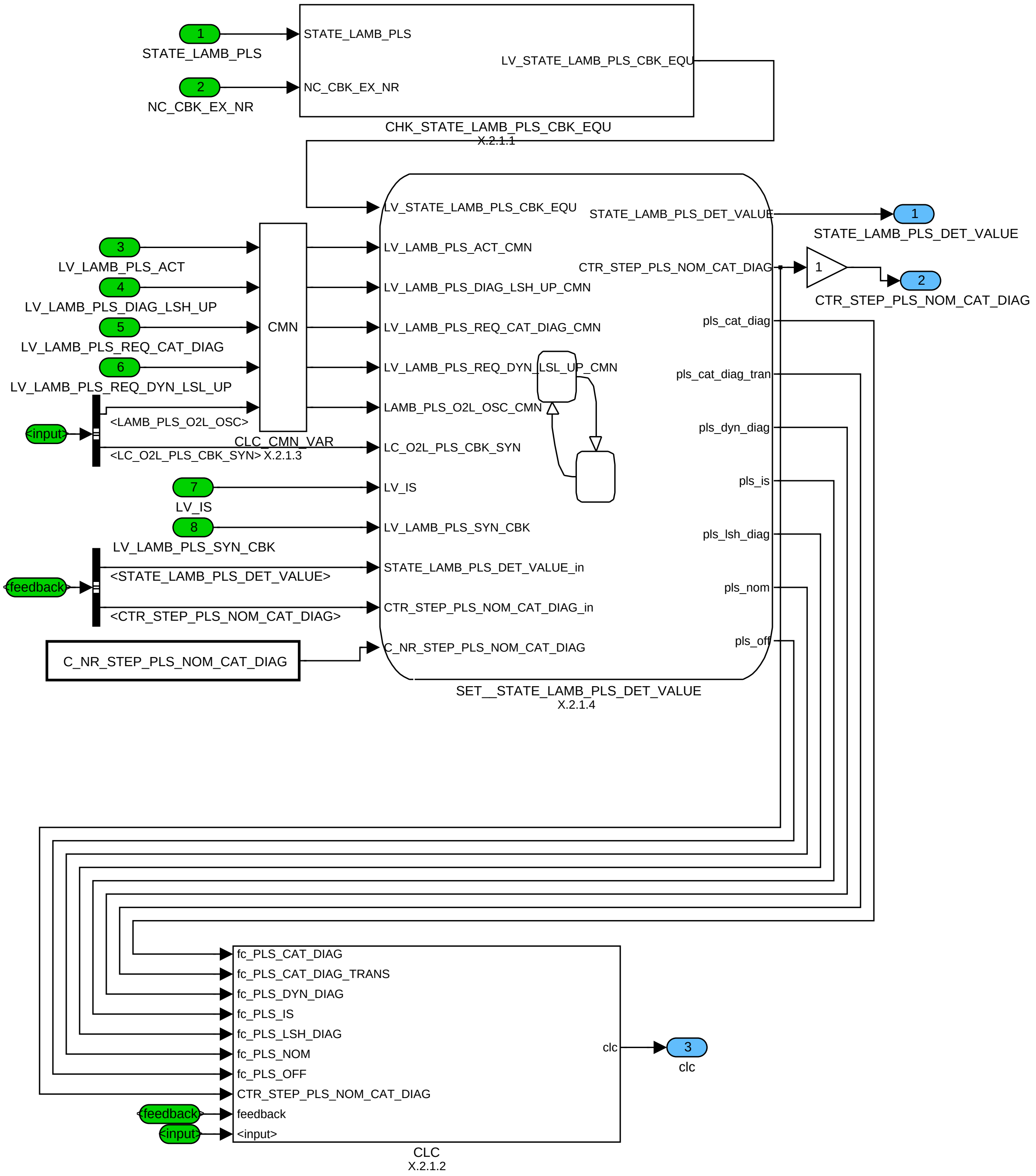


Figure 82:
Path: EGTR_DEFSPFSTID0/OPM/CLC

18.10.2.1.1 Check STATE_LAMB_PLS[i] is equal for both banks

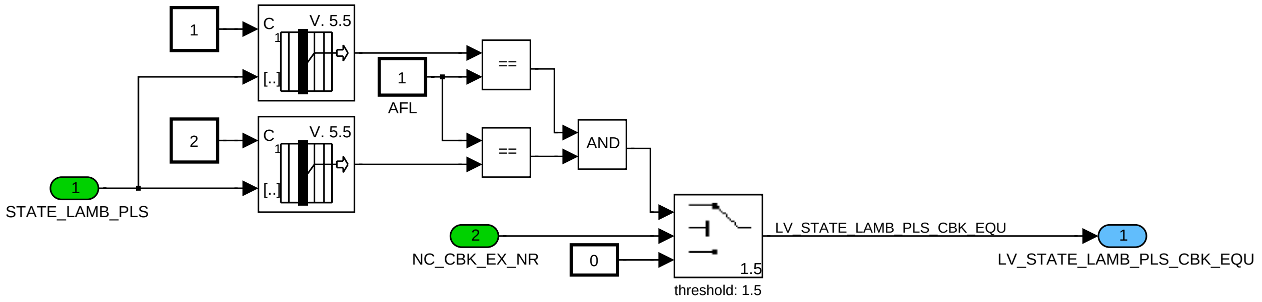


Figure 83:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CHK_STATE_LAMB_PLS_CBK_EQU

18.10.2.1.2 Calculation of state contents

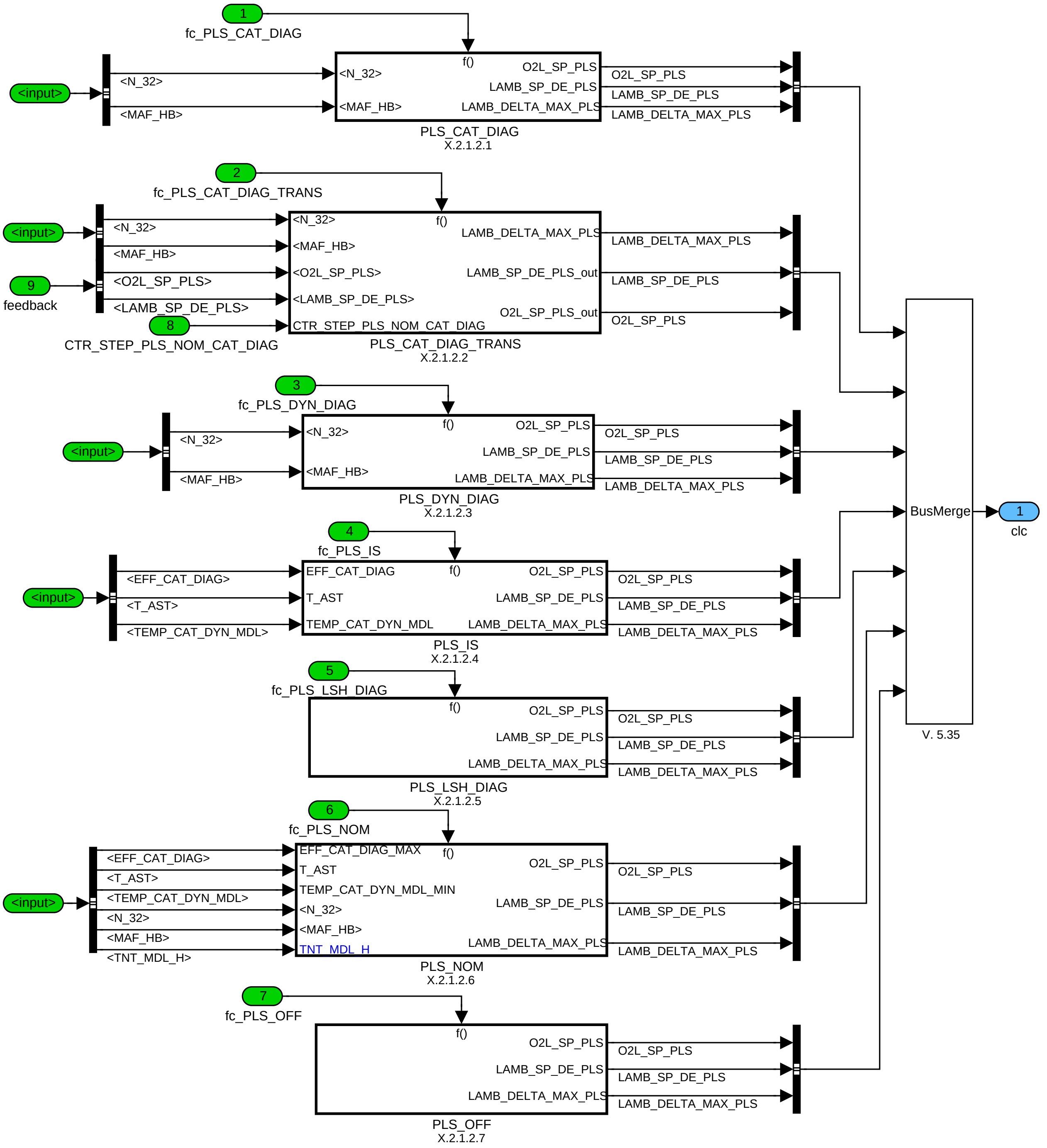


Figure 84:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC

18.10.2.1.2.1 Parameters for catalyst efficiency diagnosis

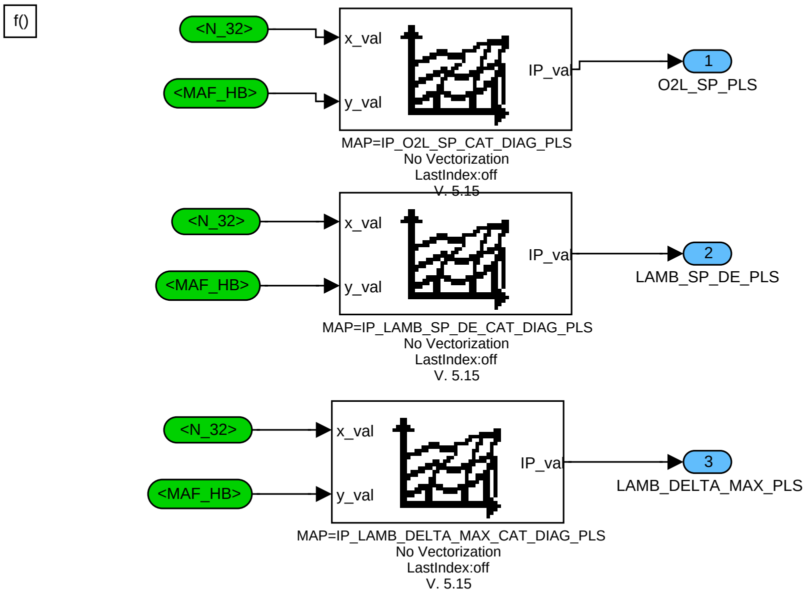


Figure 85:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC/PLS_CAT_DIAG

18.10.2.1.2.2 Parameters for transition phase to catalyst efficiency diagnosis

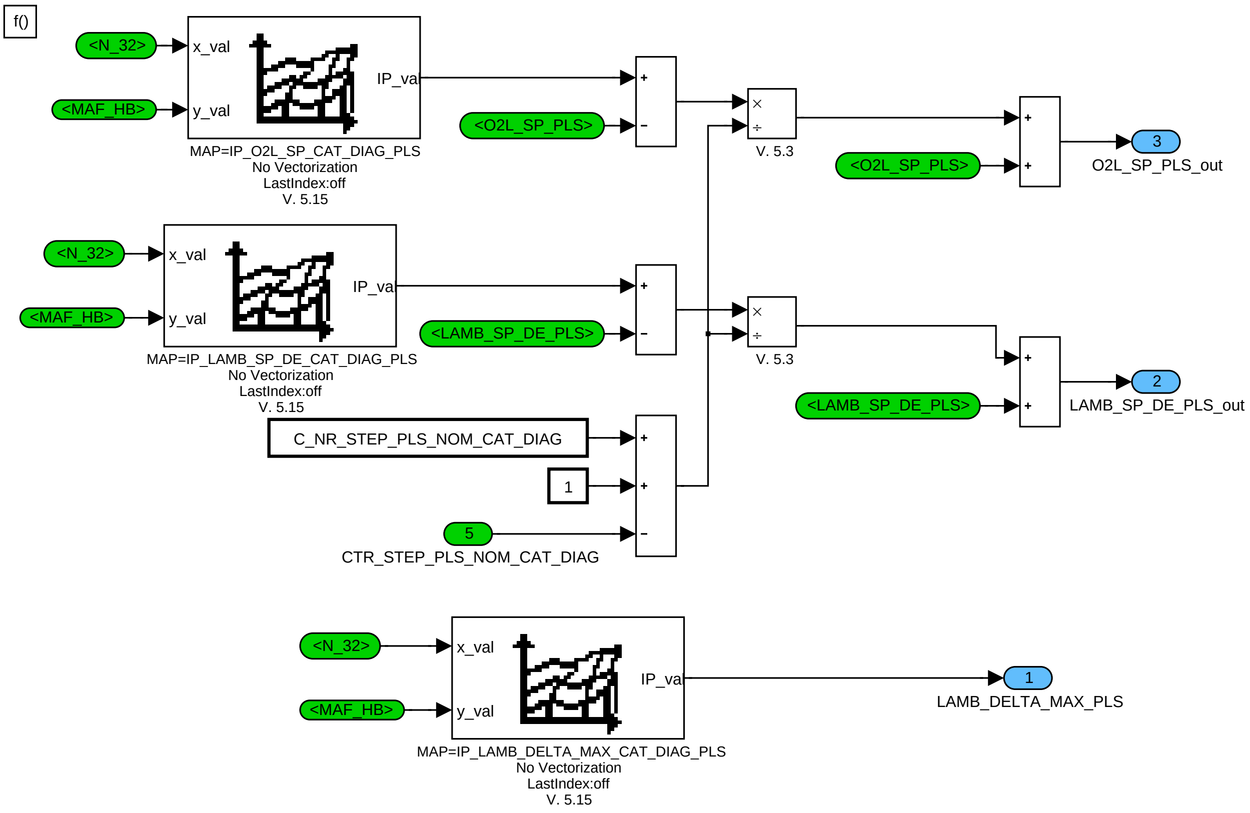


Figure 86:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC/PLS_CAT_DIAG_TRANS

18.10.2.1.2.3 Parameters for WRAF sensor dynamic diagnosis

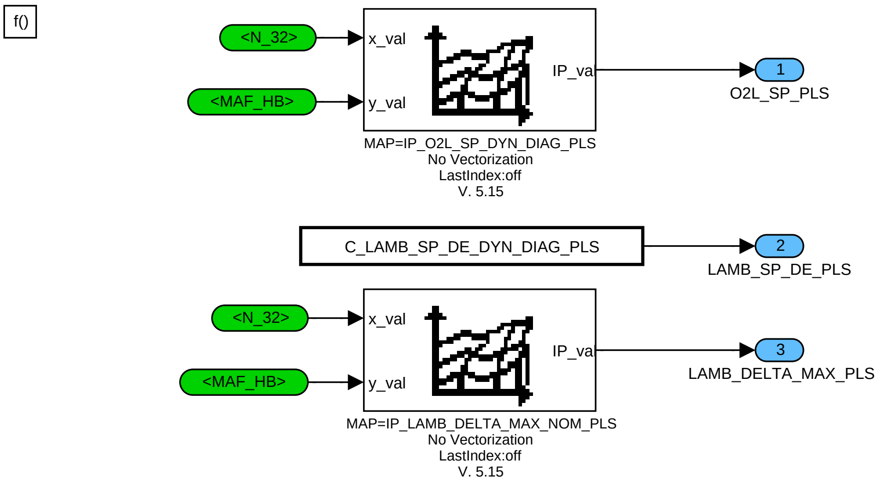


Figure 87:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC/PLS_DYN_DIAG

18.10.2.1.2.4 Parameters for IDLE speed

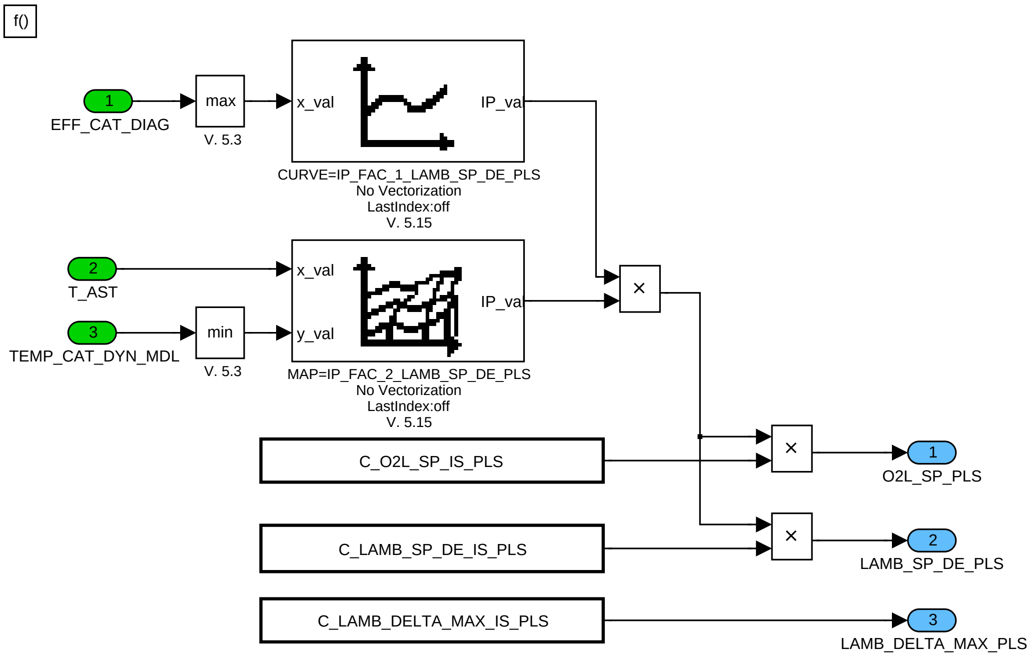


Figure 88:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC/PLS_IS

18.10.2.1.2.5 Parameters for requested mode

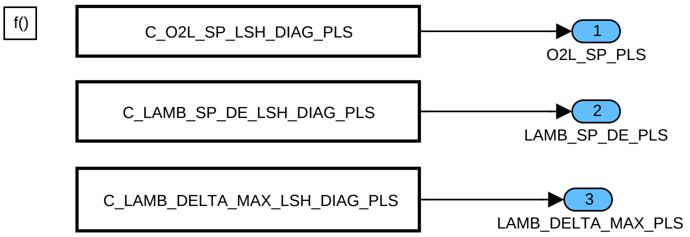


Figure 89:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC/PLS_LSH_DIAG

18.10.2.1.2.6 Parameters for nominal stimulation

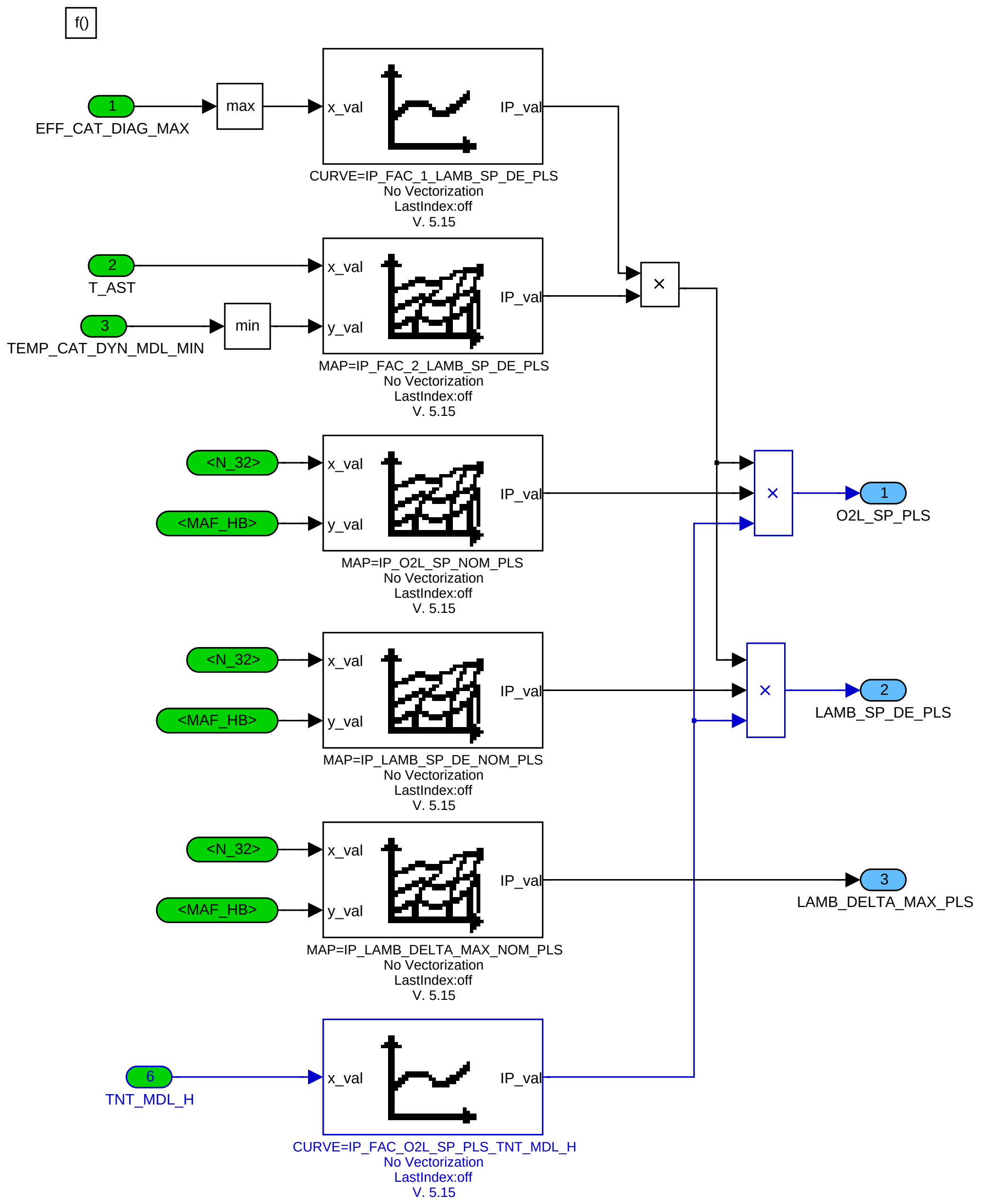


Figure 90:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC/PLS_NOM

18.10.2.1.2.7 No parameters

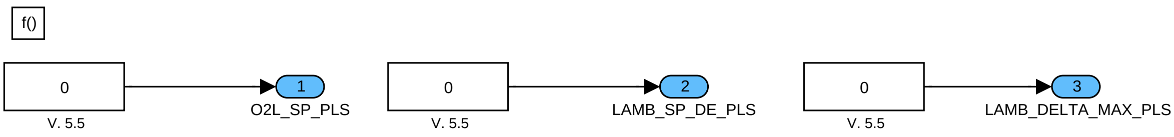


Figure 91:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC/PLS_OFF

18.10.2.1.3 Calculation bank crosses variables (CMN variables)

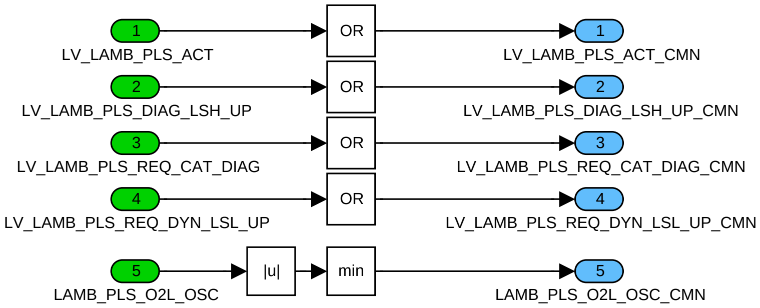


Figure 92:
Path: EGTR_DEFSPFSTID0/OPM/CLC/CLC_CMN_VAR

18.10.2.1.4 SET__STATE_LAMB_PLS_DET_VALUE

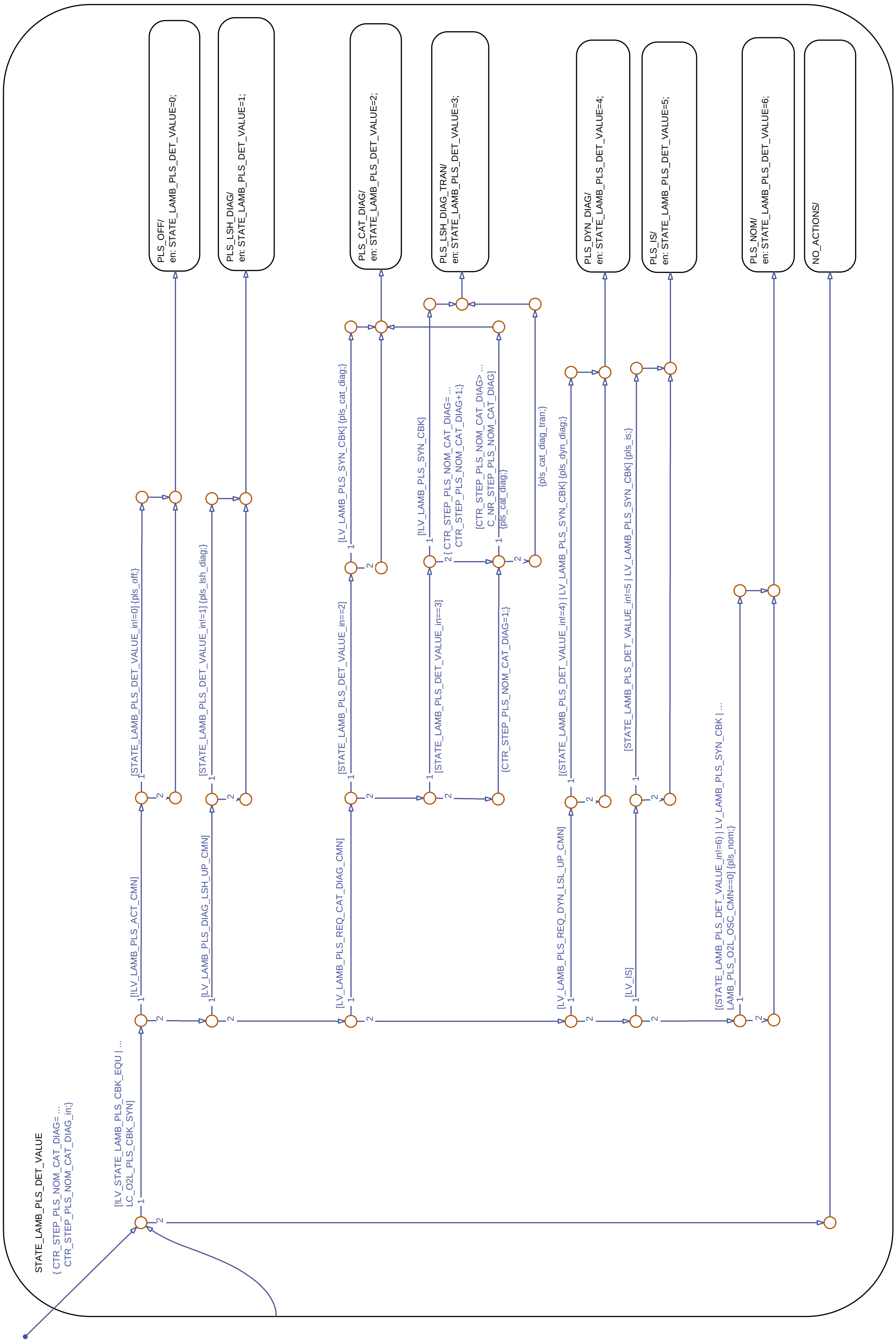


Figure 93:
Path: EGTR_DEFSPFSTID0/OPM/CLC/SET__STATE_LAMB_PLS_DET_VALUE

18.11 Limited dynamics for catalyst efficiency diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CAT_LDC[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
status of limited dynamics for catalyst efficiency diagnosis					
FAC_LAM_MV_DELTA_CAT_LDC[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
Difference between filtered and not filtered FAC_LAM_MV[i] for lambda limited dynamic					
MAF_MMV_CAT_LDC	V	0...FFFFH	0...1.389E+3	0.0211948	mg/stk
floating mean value for limited MAF					
N_MMV_CAT_LDC	V	0...1FE0H	0...8.16E+3	1	rpm
floating mean value for limited engine speed					
LOAD_GRD_CAT_LDC	V	200...1FFH	-1.25E+3 ... 1.24756E+3	2.44140625	%/s
Load gradient for limted dynamics of catalyst diagnosis					
LOAD_GRD_SUM_CAT_LDC	V	0...FFFFH	0...1.59998E+5	2.44140625	%/s
sum value of LOAD gradient fro limited dynamics for catalyst diagnosis					
LV_FAC_LAM_CAT_LDC[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Status limited dynamic FAC_LAM_MV[NC_CBK_EX_NR]					
LV_LOAD_GRD_CAT_LDC	V	0...1H	0...1	1	-
Status limited dynamic LOAD_GRD_CAT_LDC					
LV_MAF_CAT_LDC	V	0...1H	0...1	1	-
Status limited dynamic MAF					
LV_N_CAT_LDC	V	0...1H	0...1	1	-
Status limited dynamic N					
MAF_INT_CAT_LDC[NC_CBK_EX_NR]	V	0...FFFFH	0...1.45633E+3	0.022222	g
integral of mass air flow since limited dynamic conditions fulfilled					

Input data:

LOAD_CAT_LDC	FAC_LAM_MV[NC_CBK_EX_NR]	FAC_LAM_MV_MMV_LDC_DIAG[NC_CBK_EX_NR]	MAF
MAF_CYL	N	LV_DC	NC_CBK_EX_NR

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_MAF_CAT_LDC	1	0...FFFFH	0...0.999985	1.52588E-5	-
correlation constant for floating mean value calculation of air mass flow					
C_CRLC_N_CAT_LDC	1	0...FFFFH	0...0.999985	1.52588E-5	-
correlation constant for floating mean value calculation of engine speed					
C_FAC_LAM_MV_DYW_CAT	1	0...FFFFH	0...99.9984741	0.00152588	%
threshold limited dynamic lambda					
C_LOAD_GRD_DEC	1	0...3FFH	0...2.49756E+3	2.44140625	%/s
decrement for calculation LOAD_CAT_LDC gradient					
C_LOAD_GRD_MIN_SUM_CAT	1	0...1FFH	0...1.24756E+3	2.44140625	%/s
min. threshold for LOAD_GRD CAT_LDC summation					
C_LOAD_GRD_SUM_MAX	1	0...FFFFH	0...1.59998E+5	2.44140625	%/s
max threshold limited dynamic LOAD_CAT_LDC gradient					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LOAD_GRD_SUM_MIN	1	0...FFFFH	0...1.59998E+5	2.44140625	%/s
min threshold limited dynamic LOAD_CAT_LDC gradient					
C_MAF_DYW_CAT_LDC	1	0...FFFFH	0...1.389E+3	0.0211948	mg/stk
threshold limited dynamic engine MAF					
C_MAF_INT_MIN_CAT_LDC	1	0...FFFFH	0...1.45633E+3	0.022222	g
MAF integral after setting limited dynamic conditions LV_CAT_LDC[i] i before starting the monitoring cycle					
C_N_DYW_CAT_LDC	1	0...1FE0H	0...8.16E+3	1	rpm
threshold limited dynamic engine speed					
C_T_LOAD_GRD_DEC_SUM	1	0...FFH	0...5.1	0.02	s
time periods for calculation limited dynamic LOAD_CAT_LDC gradient					
LC_FAC_LAM_RST_CAT_LDC	1	0...1H	0...1	1	-
Switch for active reset of filtered FAC_LAM to not filtered after lim. Dyn. over threshold					
LC_MAF_RST_CAT_LDC	1	0...1H	0...1	1	-
Switch for active reset of filtered MAF to not filtered after lim. Dyn. over threshold					
LC_N_RST_CAT_LDC	1	0...1H	0...1	1	-
Switch for active reset of filtered N to not filtered after lim. Dyn. over threshold					

18.11.1 EGTR_FCTDGCEFLD0

The limited dynamic conditions LV_CAT_LDC[i] for catalyst efficiency diagnosis is set if the limited dynamic conditions for engine speed, mass air flow and lambda controller output exist and a calibration threshold for the integral of the air flow is exceeded.

Moreover the summation of the throttle position gradient LOAD_GRD_SUM_CAT_LDC[i] must not exceed a calibration threshold.

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements, the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

Function Description

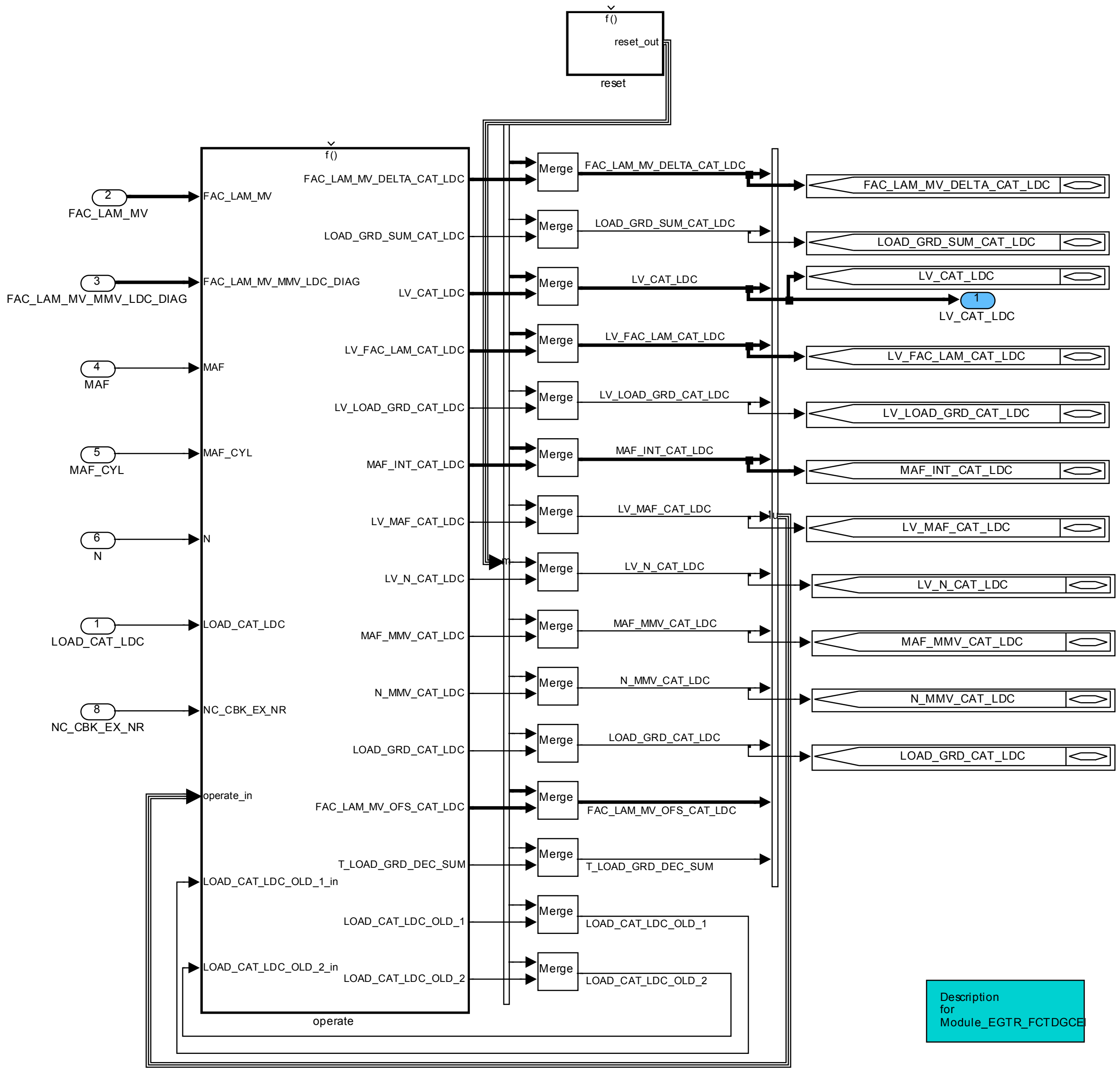


Figure 94 EGTR_FCTDGCEFLD0

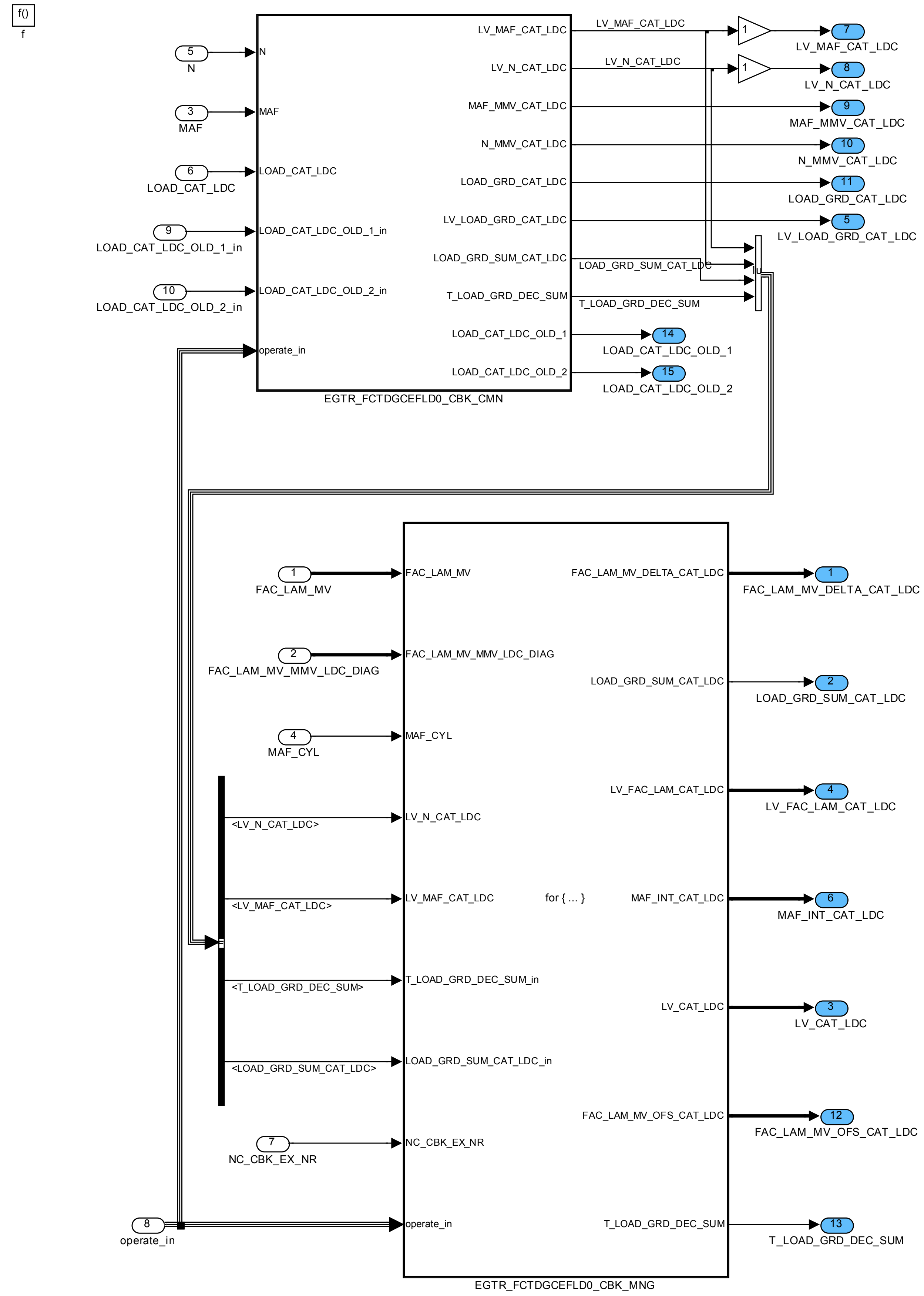

f()
f
5
N
3
MAF
6
LOAD_CAT_LDC
9
LOAD_CAT_LDC_OLD_1_in
10
LOAD_CAT_LDC_OLD_2_in
N
MAF
LOAD_CAT_LDC
LOAD_CAT_LDC_OLD_1_in
LOAD_CAT_LDC_OLD_2_in
operate_in
LV_MAF_CAT_LDC
LV_N_CAT_LDC
MAF_MMV_CAT_LDC
N_MMV_CAT_LDC
LOAD_GRD_CAT_LDC
LV_LOAD_GRD_CAT_LDC
LOAD_GRD_SUM_CAT_LDC
T_LOAD_GRD_DEC_SUM
LOAD_CAT_LDC_OLD_1
LOAD_CAT_LDC_OLD_2
EGTR_FCTDGCEFLD0_CBK_CMN
LV_MAF_CAT_LDC
LV_N_CAT_LDC
1
1
7
LV_MAF_CAT_LDC
8
LV_N_CAT_LDC
9
MAF_MMV_CAT_LDC
10
N_MMV_CAT_LDC
11
LOAD_GRD_CAT_LDC
5
LV_LOAD_GRD_CAT_LDC
LOAD_GRD_SUM_CAT_LDC
T_LOAD_GRD_DEC_SUM
14
LOAD_CAT_LDC_OLD_1
15
LOAD_CAT_LDC_OLD_2
1
FAC_LAM_MV
2
FAC_LAM_MV_MMV_LDC_DIAG
4
MAF_CYL
<LV_N_CAT_LDC>
<LV_MAF_CAT_LDC>
<T_LOAD_GRD_DEC_SUM>
<LOAD_GRD_SUM_CAT_LDC>
7
NC_CBK_EX_NR
8
operate_in
FAC_LAM_MV
FAC_LAM_MV_MMV_LDC_DIAG
MAF_CYL
LV_N_CAT_LDC
LV_MAF_CAT_LDC
T_LOAD_GRD_DEC_SUM_in
LOAD_GRD_SUM_CAT_LDC_in
NC_CBK_EX_NR
operate_in
for { ... }
FAC_LAM_MV_DELTA_CAT_LDC
LOAD_GRD_SUM_CAT_LDC
LV_FAC_LAM_CAT_LDC
MAF_INT_CAT_LDC
LV_CAT_LDC
FAC_LAM_MV_OFS_CAT_LDC
T_LOAD_GRD_DEC_SUM
EGTR_FCTDGCEFLD0_CBK_MNG
1
FAC_LAM_MV_DELTA_CAT_LDC
2
LOAD_GRD_SUM_CAT_LDC
4
LV_FAC_LAM_CAT_LDC
6
MAF_INT_CAT_LDC
3
LV_CAT_LDC
12
FAC_LAM_MV_OFS_CAT_LDC
13
T_LOAD_GRD_DEC_SUM

Figure 95 EGTR_FCTDGCEFLD0/ operate

Cylinder bank independent calculation

Calculation of LOAD_CAT_LDC gradient:

The LOAD_CAT_LDC gradient is calculated with the reccurence T_SAMPLE in [ms].

To this effect, the LOAD_CAT_LDC values are written into the ring buffer. Thus, the last two LOAD_CAT_LDC values are available in addition to the current LOAD_CAT_LDC value. The LOAD_CAT_LDC gradient is based on a LOAD_CAT_LDC difference of twice T_SAMPLE in [ms].

The LOAD_GRD_CAT_LDC reflects the gradient of the actual driver's demand prior to the corrections. If C_LOAD_GRD_SUM_MIN is larger than C_LOAD_GRD_SUM_MAX the LOAD_CAT_LDC gradient summation LOAD_GRD_SUM_CAT_LDC will be no condition to determine the limited dynamic bit LV_CAT_LDC[i].

Limited engine speed (N) dynamics:

The floating mean value N_MMV_CAT_LDC is computed using the filtering constant C_N_CRLC_CAT_LDC. The limited dynamics condition only exist while the engine speed N lies within the dynamics window around N_MMV_CAT_LDC.

If the above limited dynamics condition is violated, the floating mean value N_MMV_CAT_LDC is set to the current engine speed in order to reach the limited dynamics condition faster.

Limited mass air flow (MAF) dynamics:

The floating mean value MAF_MMV_CAT_LDC is computed using the averaging constant C_MAF_CRLC_CAT. The limited dynamics condition only exist while the air mass MAF stays within the dynamics window around MAF_MMV_CAT_LDC.

If the above limited dynamics condition is violated, the floating mean value MAF_MMV_CAT_LDC is set to the current air-mass value in order to reach the limited dynamics condition faster.

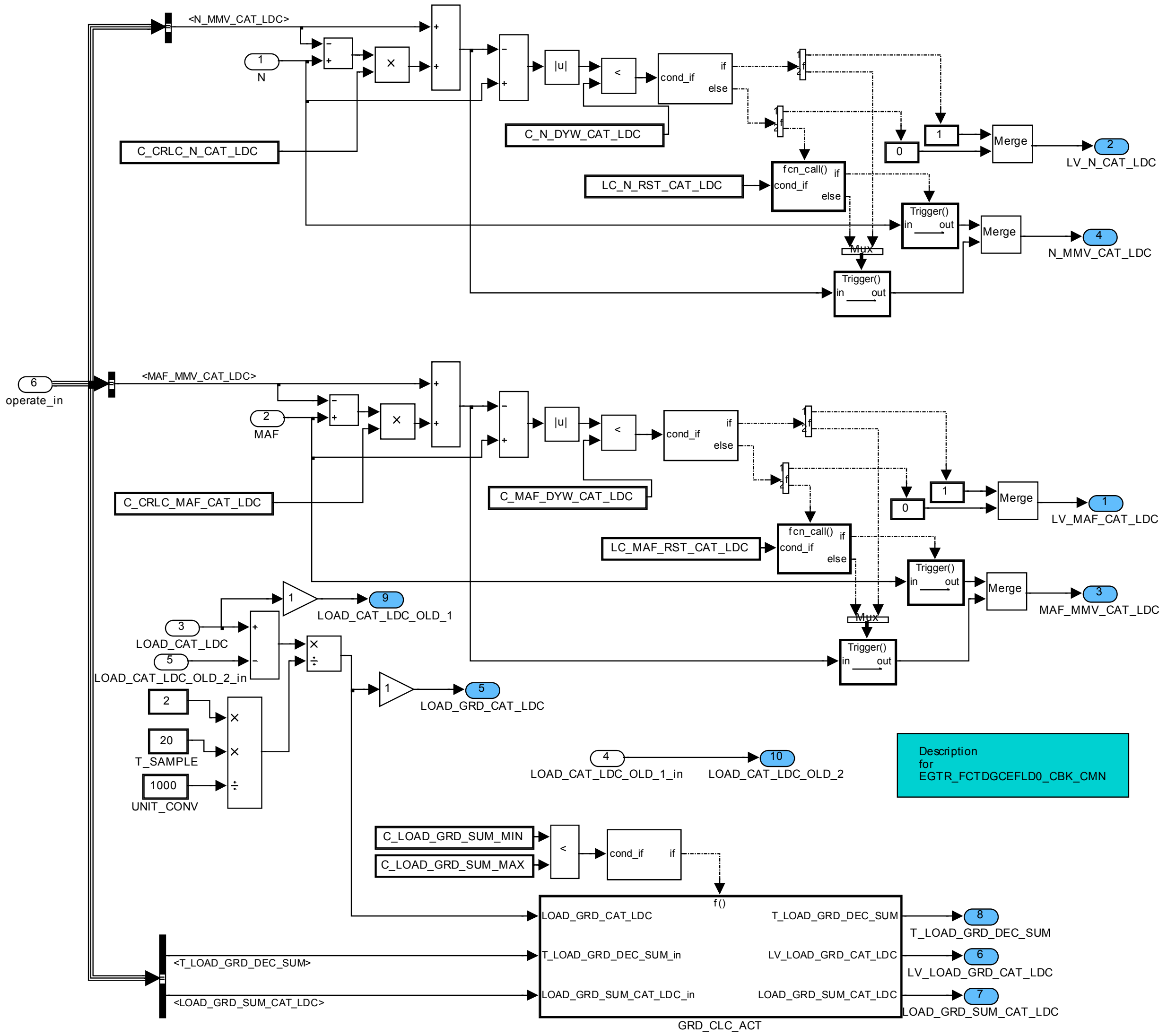


Figure 96 EGTR_FCTDGCEFLD0/ operate/ EGTR_FCTDGCEFLD0_CBK_CMN

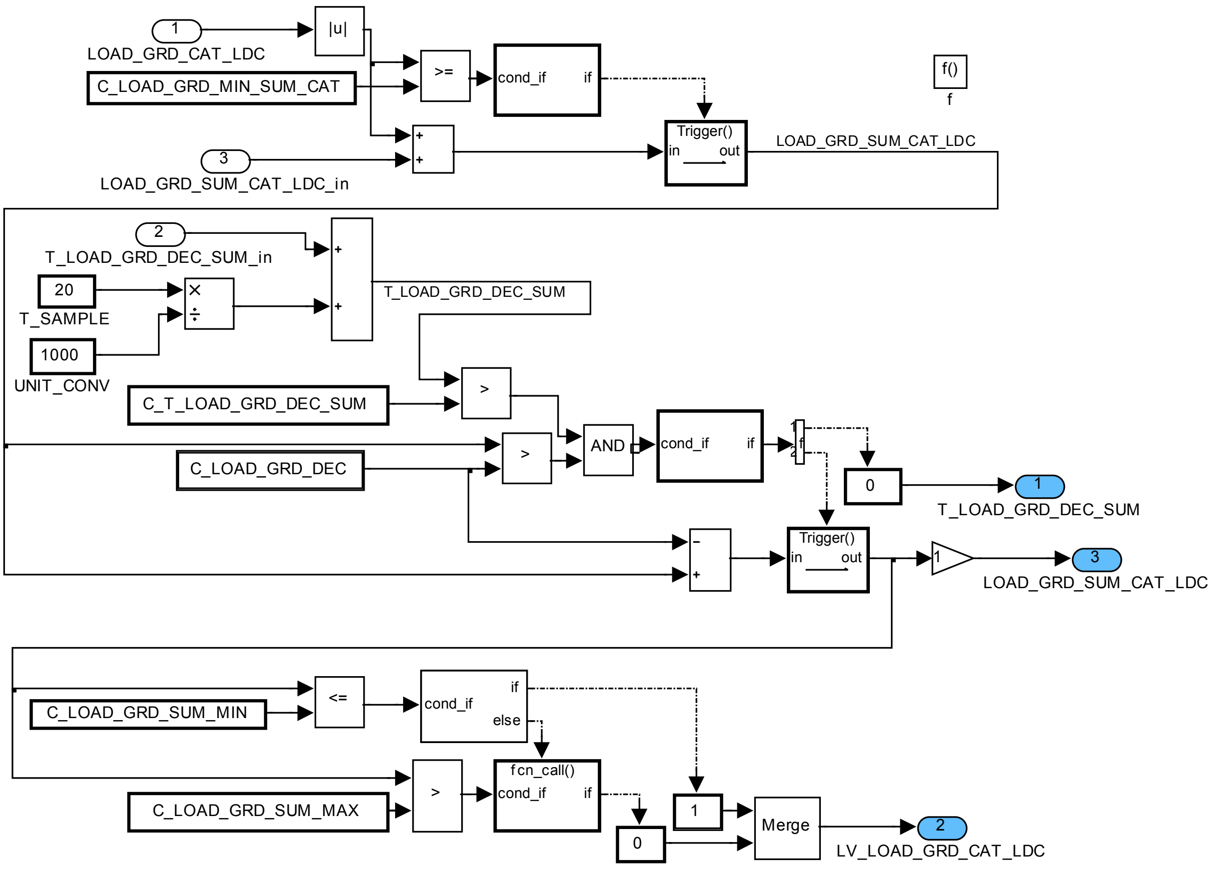


Figure 97 EGTR_FCTDGCEFLD0/ operate/ EGTR_FCTDGCEFLD0_CBK_CMN/ GRD_CLC_ACT

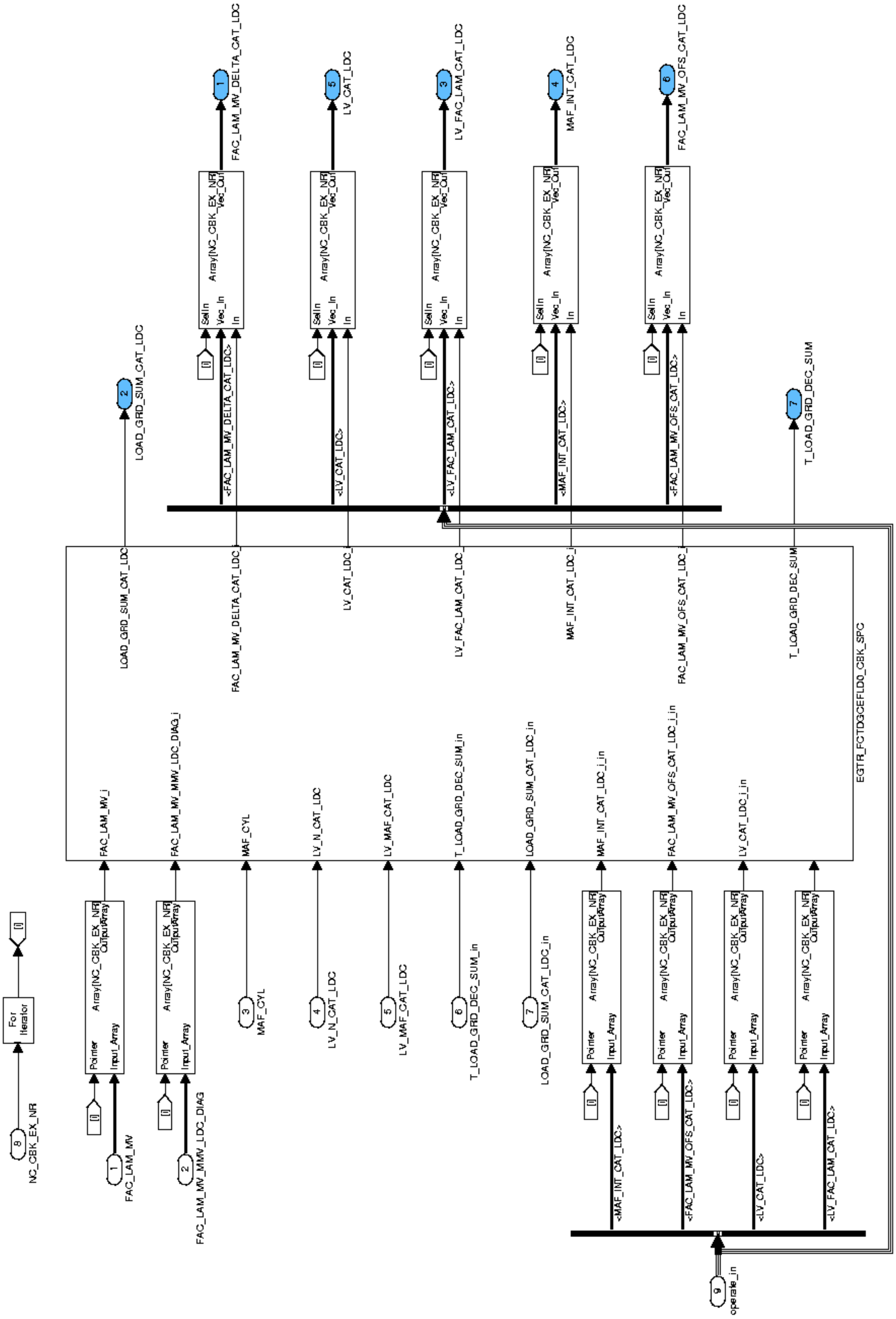


Figure 98 EGTR_FCTDGCEFLD0/ operate/ EGTR_FCTDGCEFLD0_CBK_MNG

<u>Cylinder bank dependent calculation</u>

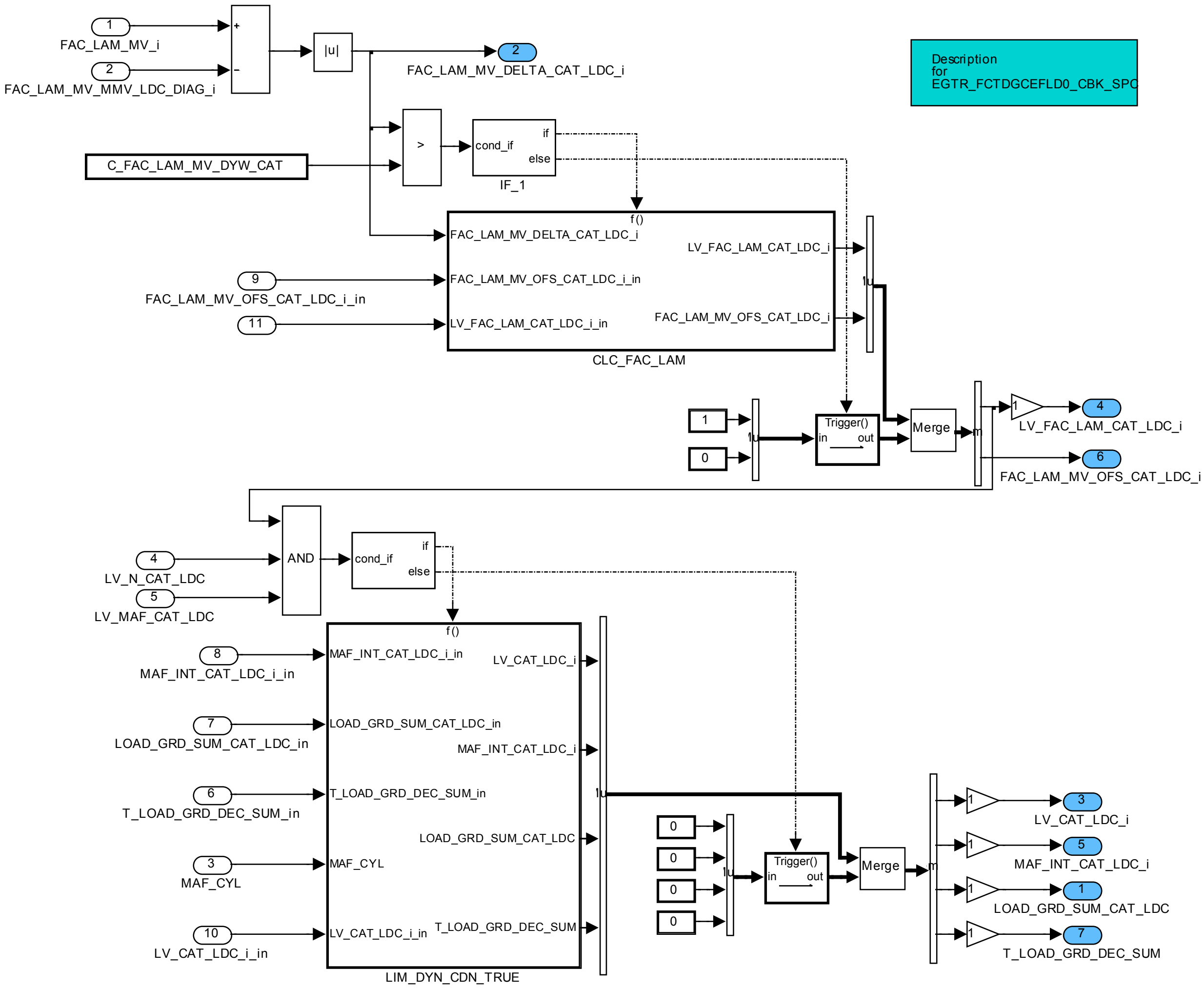


Figure 99 EGTR_FCTDGCEFLD0/ operate/ EGTR_FCTDGCEFLD0_CBK_MNG/ EGTR_FCTDGCEFLD0_CBK_SPC

Limited mean oxygen value (FAC_LAM_MV[i]) dynamics

The floating mean value FAC_LAM_MV_MMV_CAT_LDC[i] is computed inside the AGGR LACO.

The limited dynamics condition continues to exist while the mean oxygen value FAC_LAM_MV[i] stays within the dynamics window around FAC_LAM_MV_DELTA_CAT_LDC[i].

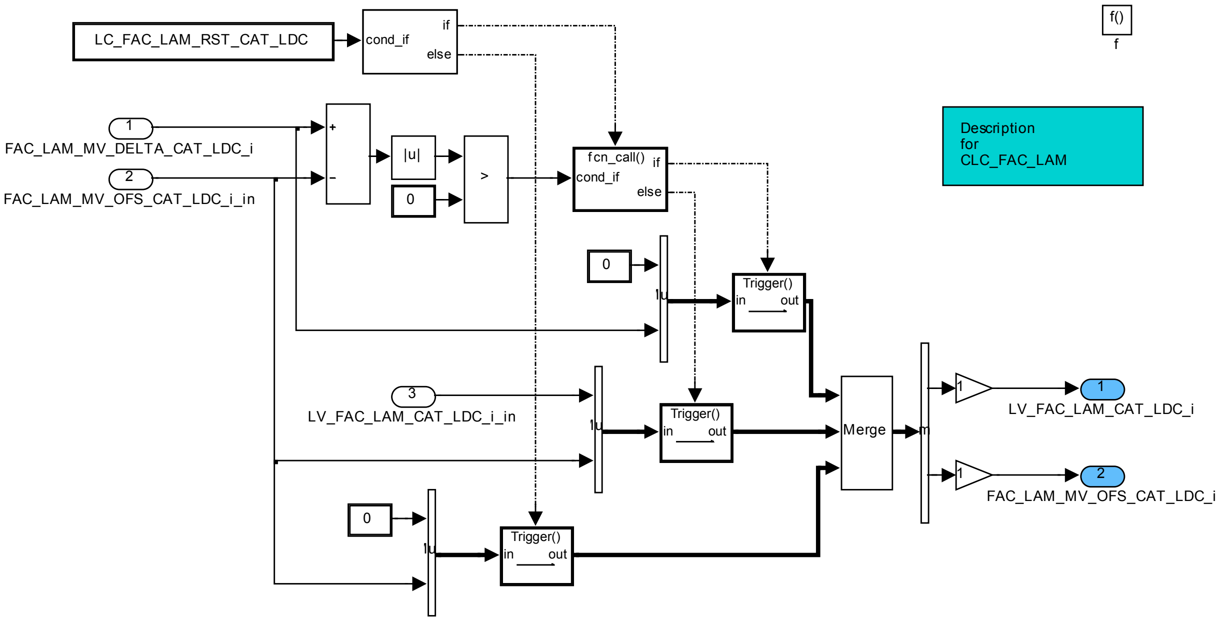


Figure 100 EGTR_FCTDGCEFLD0/ operate/ EGTR_FCTDGCEFLD0_CBK_MNG/ EGTR_FCTDGCEFLD0_CBK_SPC/ CLC_FAC_LAM

Setting of LV_CAT_LDC[i]

When the air mass flow integral exceeds a calibration limit the flag LV_CAT_LDC[i] is set to 1 indicating that the limited dynamics are fulfilled.

The reccurence time is defined by T_SAMPLE in [ms].

For the calculation of the air mass flow integral the unit conversion:

[g] = [kg/h] * [ms] * [(g*h) / (kg*ms)] / 3600

has to be considered.

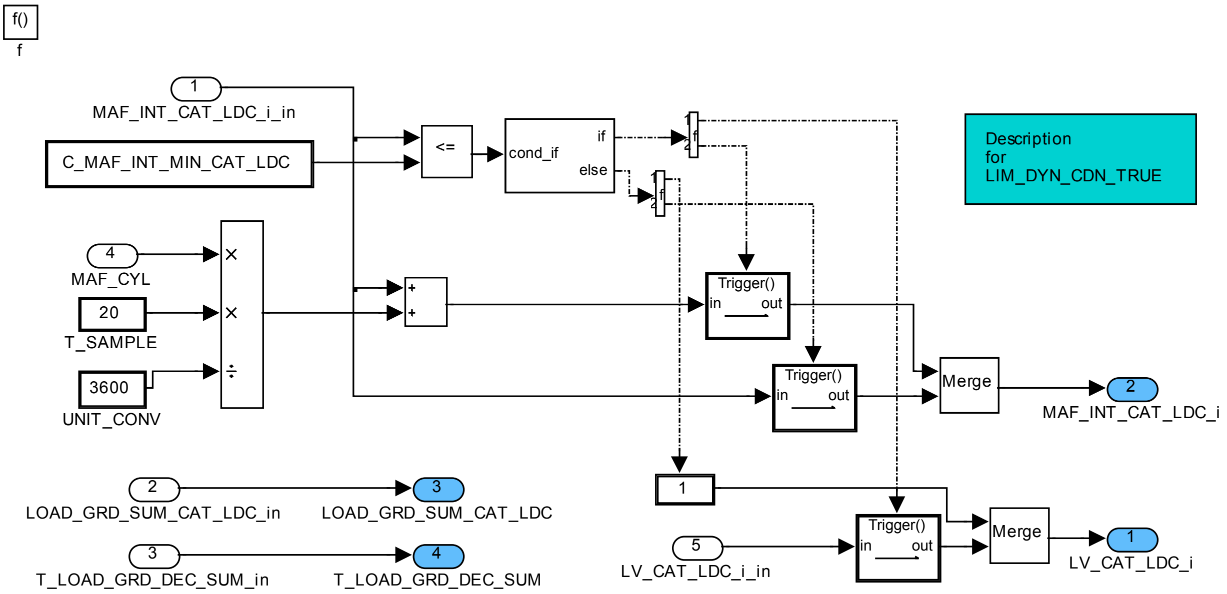


Figure 101 EGTR_FCTDGCEFLD0/ operate/ EGTR_FCTDGCEFLD0_CBK_MNG/ EGTR_FCTDGCEFLD0_CBK_SPC/ LIM_DYN_CDN_TRUE

18.12 Activation conditions for catalyst efficiency diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CAT_DIAG_CDN_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Activation conditions for O2 sensor and catalyst efficiency diagnosis (OSC-method)					
MAF_INT_DLY_CAT_DIAG[NC_CBK_EX_NR]	V	0...FFFFH	0...1.45633E+3	0.02222222	g
Air mass flow integral after conditions are fulfilled for catalyst diagnosis activation					

Input data:

LV_CAT_LDC[NC_CBK_EX_NR]	LV_LAMB_PLS_ACT[NC_CBK_EX_NR]	LV_CAT_PURGE_ACT[NC_CBK_EX_NR]	LV_LAM_ADJ_RNG_VLD_CAT_DIAG[NC_CBK_EX_NR]
AMP_AD	CL_MMV	LV_DC	LV_LAM_LSCL[NC_CBK_EX_NR]
LV_LAM_STOP[NC_CBK_EX_NR]	LV_LS_DOWN_READY[NC_CBK_EX_NR]	LV_SAWUP	MAF
MAF_CYL	NC_CBK_EX_NR	N_32	STATE_CP
STATE_LSH_DOWN[NC_CBK_EX_NR]	TCO	TEMP_CAT_DYN_MDL[NC_CBK_EX_NR]	VS
LV_CL_CLC_AVL	LV_CAT_DIAG_REQ_EOL		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_AMP_MIN_CAT	1	0...FFFFH	0...5.434E+3	0.08291752	hPa
Min. ambient pressure threshold					
C_CL_MAX_CAT	1	0...FFFFH	0...1.99996948	3.05176E-5	-
Maximum canister load to allow catalyst and O2 sensor diagnosis					
C_MAF_HYS_CAT	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
hysterese after exceeding MAF_LAM_CAT threshold					
C_MAF_INT_DLY_CAT_DIAG	1	0...FFFFH	0...1.45633E+3	0.02222222	g
Air mass flow threshold for cat diag activation after PUC, CL max purge, forced stimulation on, downstream sensor ready					
C_MAF_INT_SW_CAT_DIAG	1	0...FFFFH	0...1.45633E+3	0.02222222	g
Air mass flow initial for cat diag activation after MAX to NO_PURGE and back					
C_TCO_MIN_CAT	1	0...FEH	-48...142.5	0.75	°C
Min TCO threshold					
C_TEMP_HYS_CAT	1	0...FFH	-273.15 ... 1.76685E+3	8	°C
hysteresis after exeeding catayst temperature threshold for catalyst diagnosis activation					
C_TEMP_MAX_CAT	1	0...FFH	-273.15 ... 1.76685E+3	8	°C
max. catalyst temperature threshold for catalyst diagnosis activation					
C_TEMP_MIN_CAT	1	0...FFH	-273.15 ... 1.76685E+3	8	°C
min. catalyst temperature threshold for catalyst diagnosis activation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEMP_MIN_CAT_EOL	1	0...FFH	-273.15 ... 1.76685E+3	8	°C
min. catalyst temperature threshold for catalyst diagnosis activation in case of EOL test					
C_VS_MAX_CAT	1	0...FFH	0...255	1	km/h
max. VS threshold					
C_VS_MIN_CAT	1	0...FFH	0...255	1	km/h
min. VS threshold					
LC_CAT_DIAG_CDN_ACT_MAN	1	0...1H	0...1	1	-
flag to set catalyst diagnosis condition manually					
IP_MAF_MAX_CAT	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDPM_N_32_3_EGTR	6	0...FFH	0...8.16E+3	32	rpm
max. MAF threshold for catalyst diagnosis activation					
IP_MAF_MIN_CAT	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDPM_N_32_3_EGTR	6	0...FFH	0...8.16E+3	32	rpm
min. MAF threshold for catalyst diagnosis activation					

18.12.1 EGTR_FCTDGCEFAC0

This function determines the activation flag LV_CAT_DIAG_CDN_ACT[i] for catalyst efficiency diagnosis (OSC method). This flag can be also be set manually by LC_CAT_DIAG_CDN_ACT_MAN.

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements, the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

For the calculation of the air mass flow integral the unit conversion:

[g] = [kg/h] * [ms] * [(g*h) / (kg*ms)] / 3600

has to be considered.

Application Condition

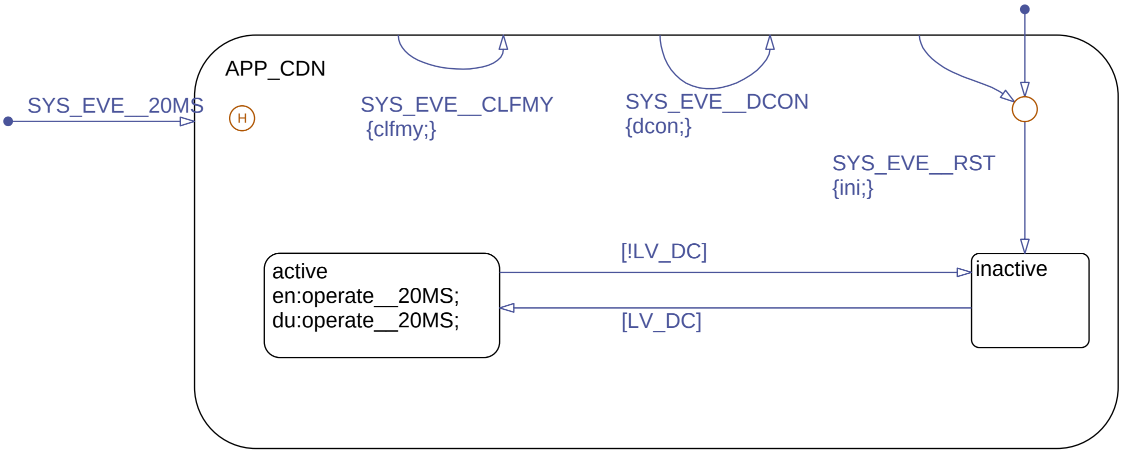


Figure 102 EGTR_FCTDGCEFAC0/ APP_CDN/ APP_CDN

Function Description

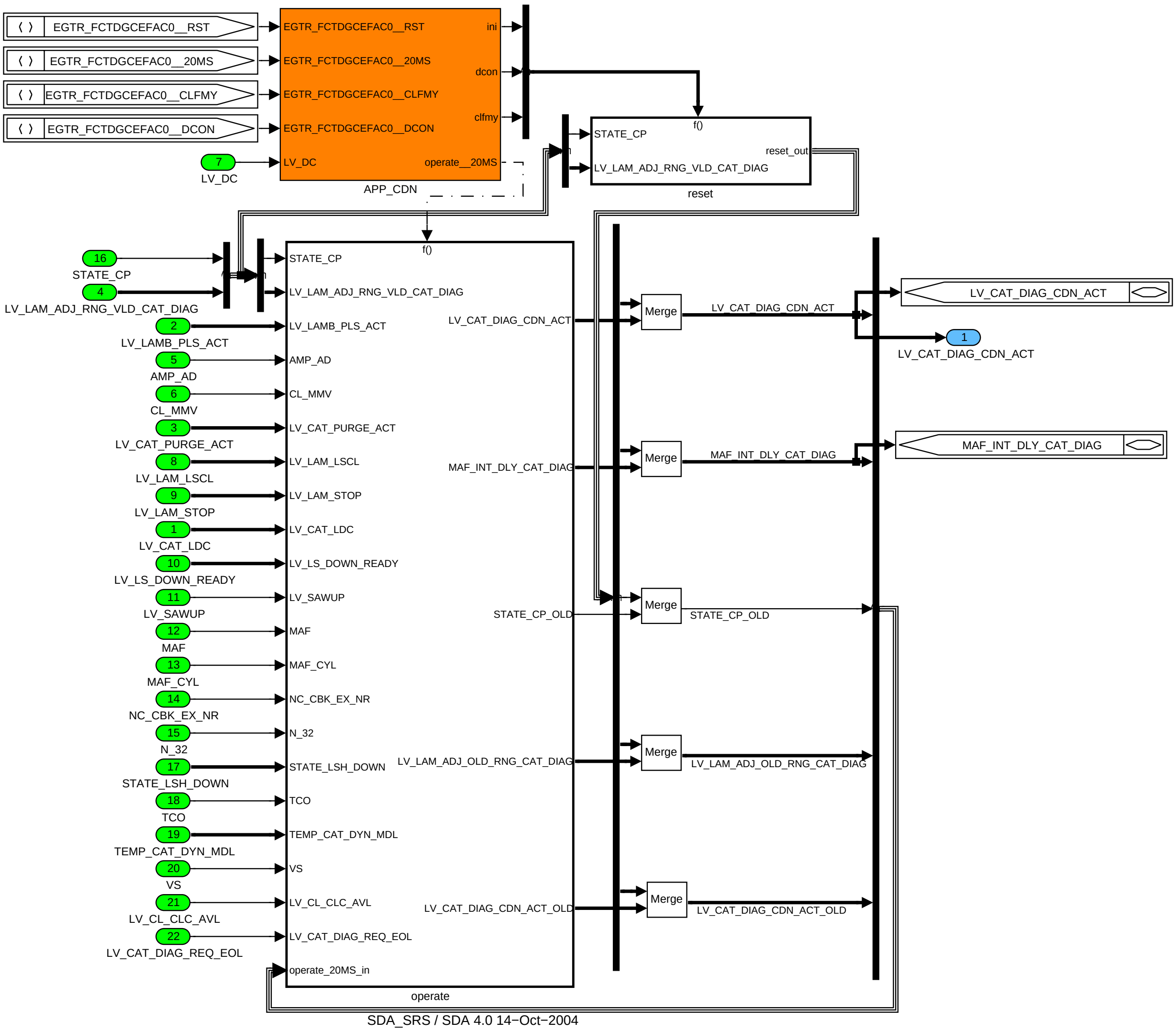


Figure 103 EGTR_FCTDGCEFAC0

18.12.1.1 SUBFUNCTION: operate

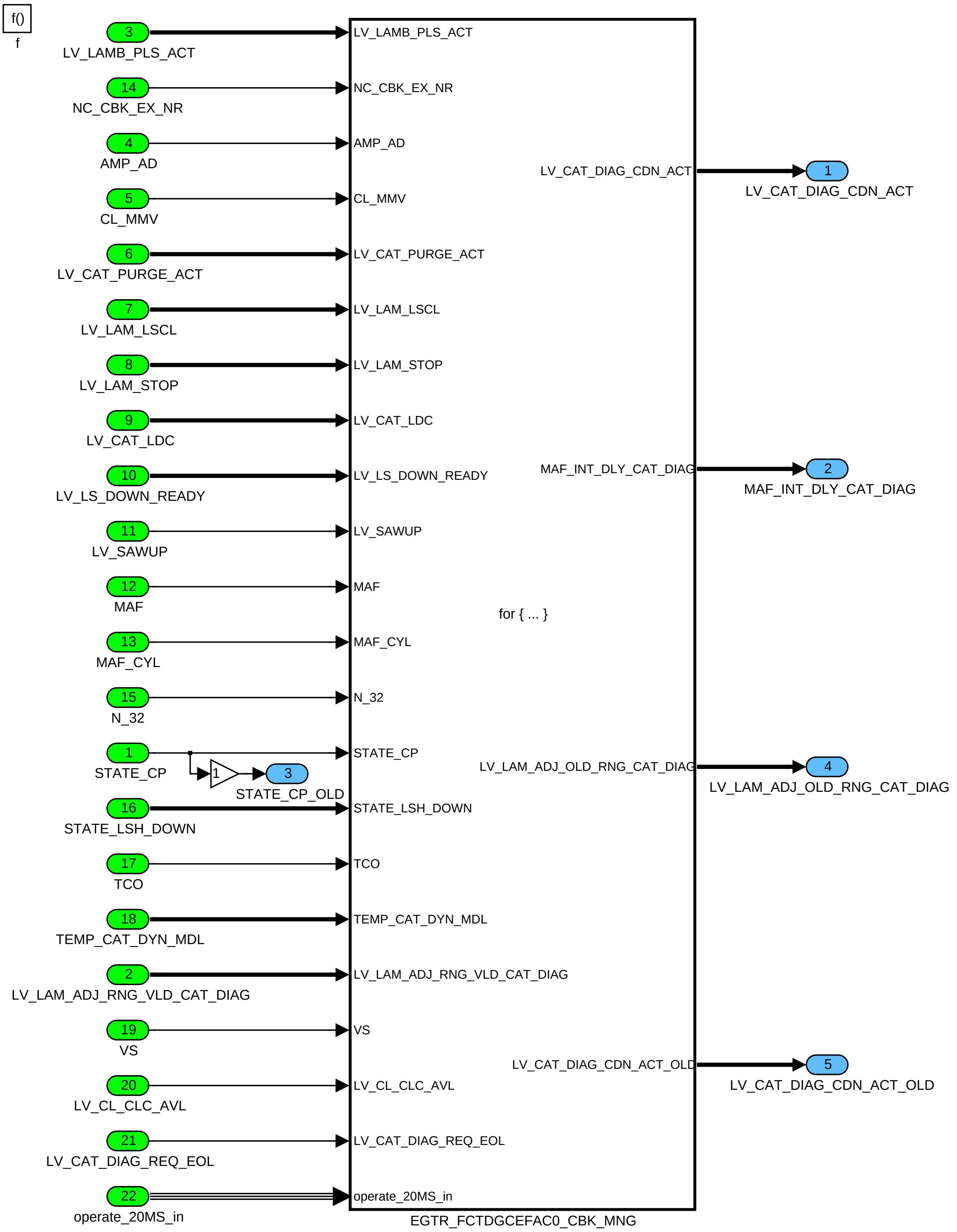


Figure 104 EGTR_FCTDGCEFAC0/ operate

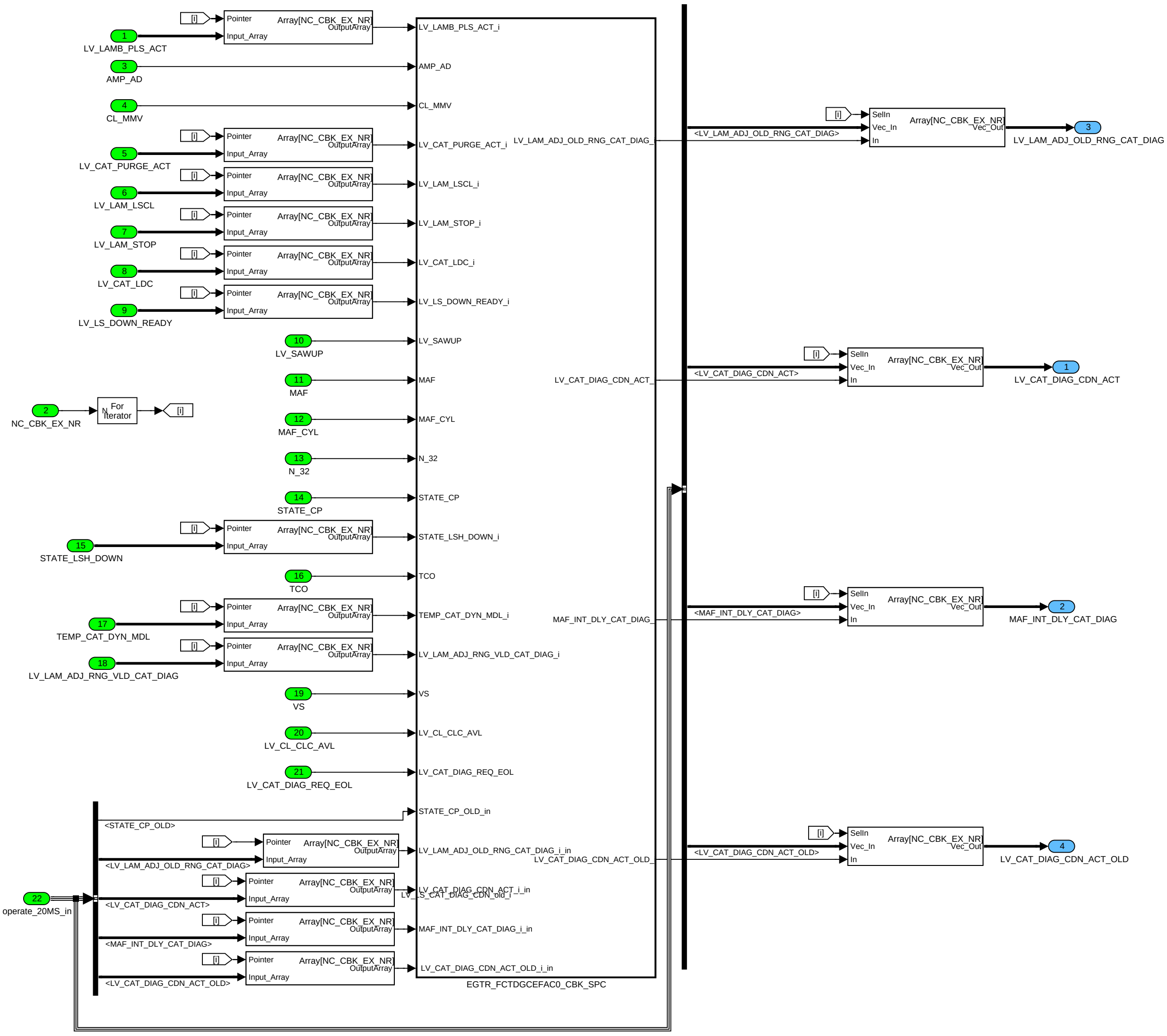


Figure 105 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG

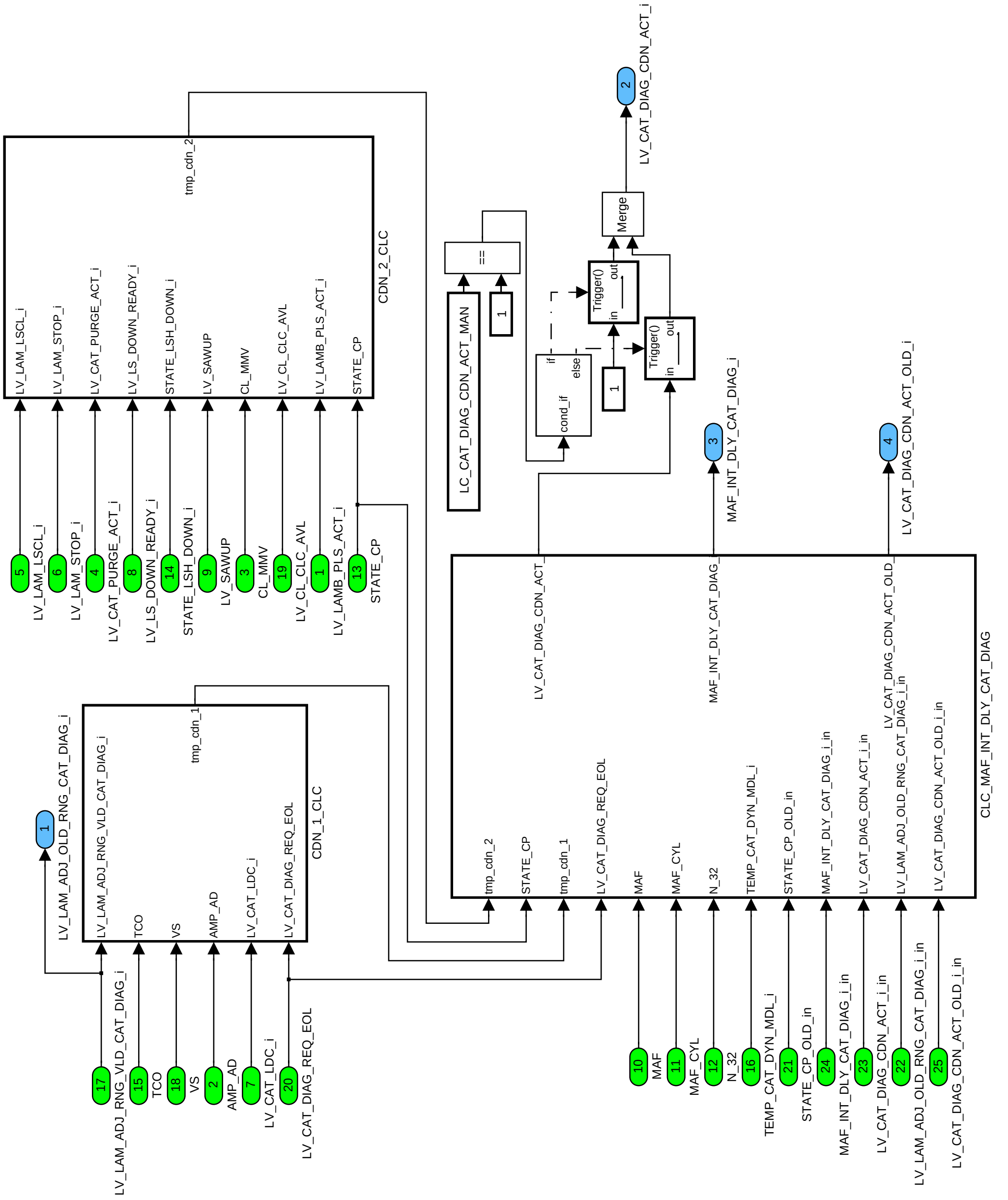


Figure 106 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC

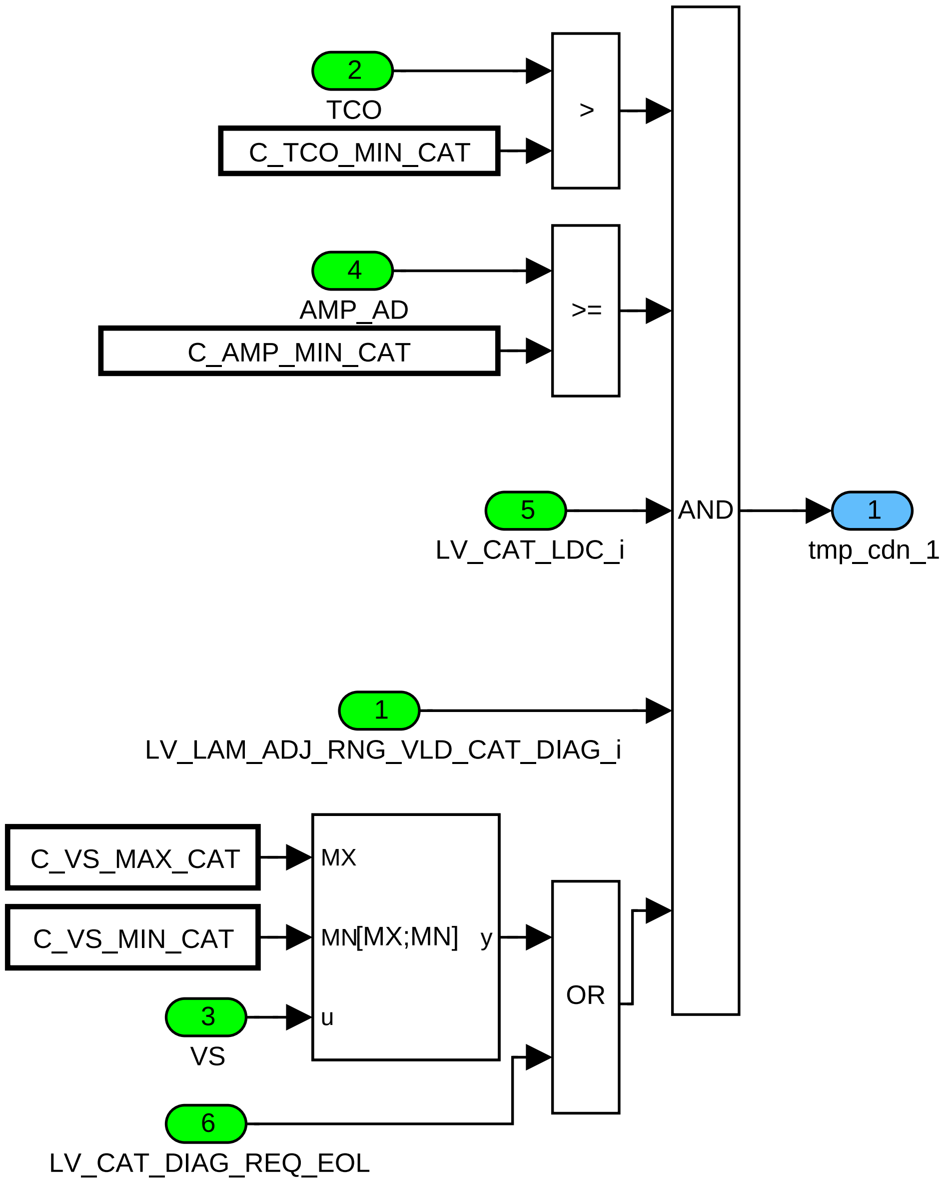


Figure 107 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CDN_1_CLC

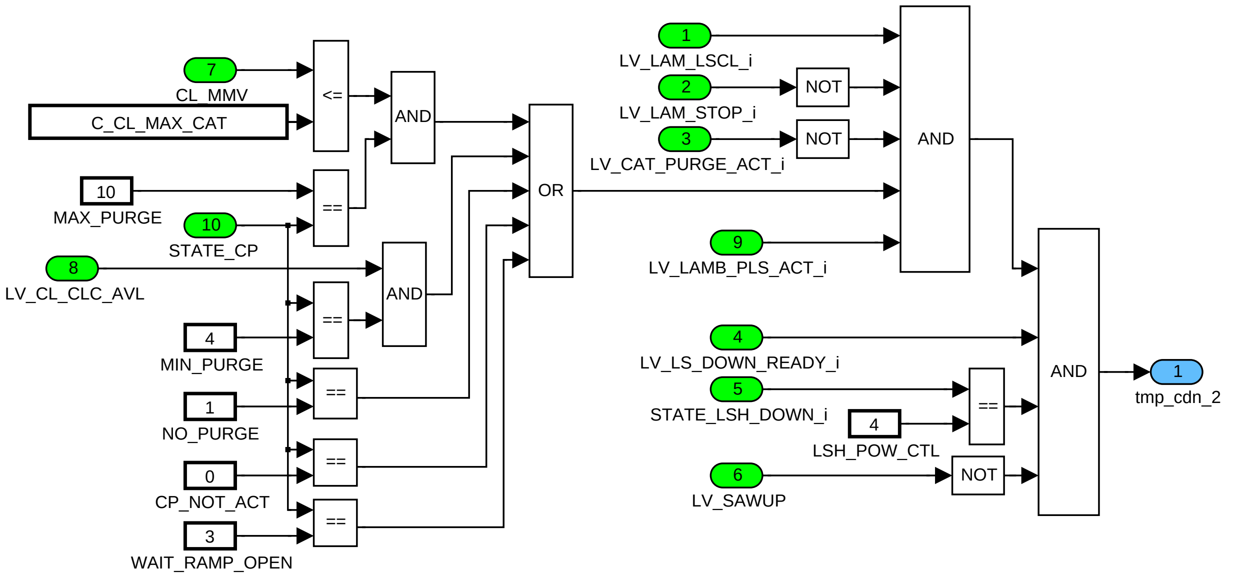


Figure 108 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CDN_2_CLC

general specification

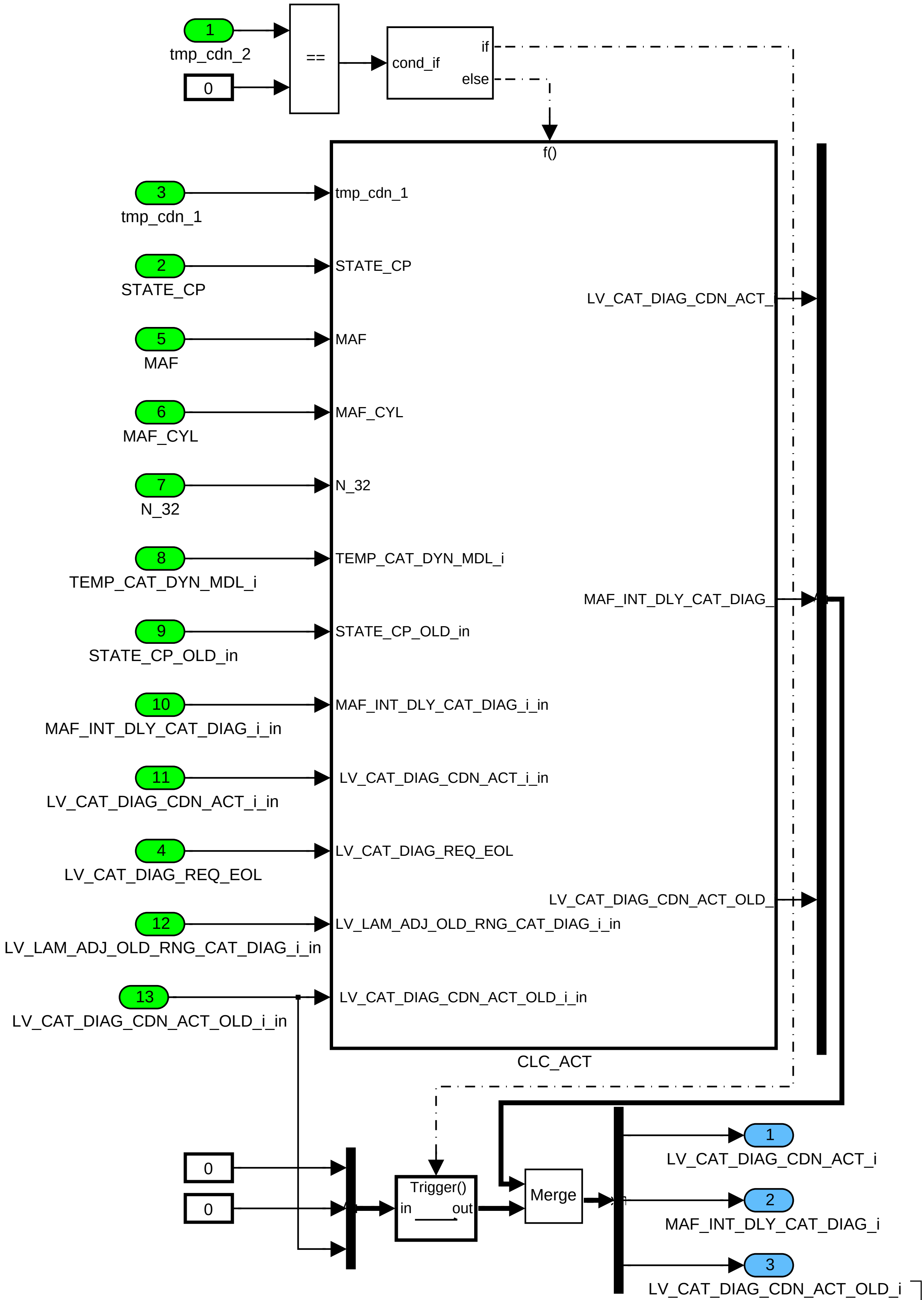


Figure 109 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CLC_MAF_INT_DLY_CAT_DIAG

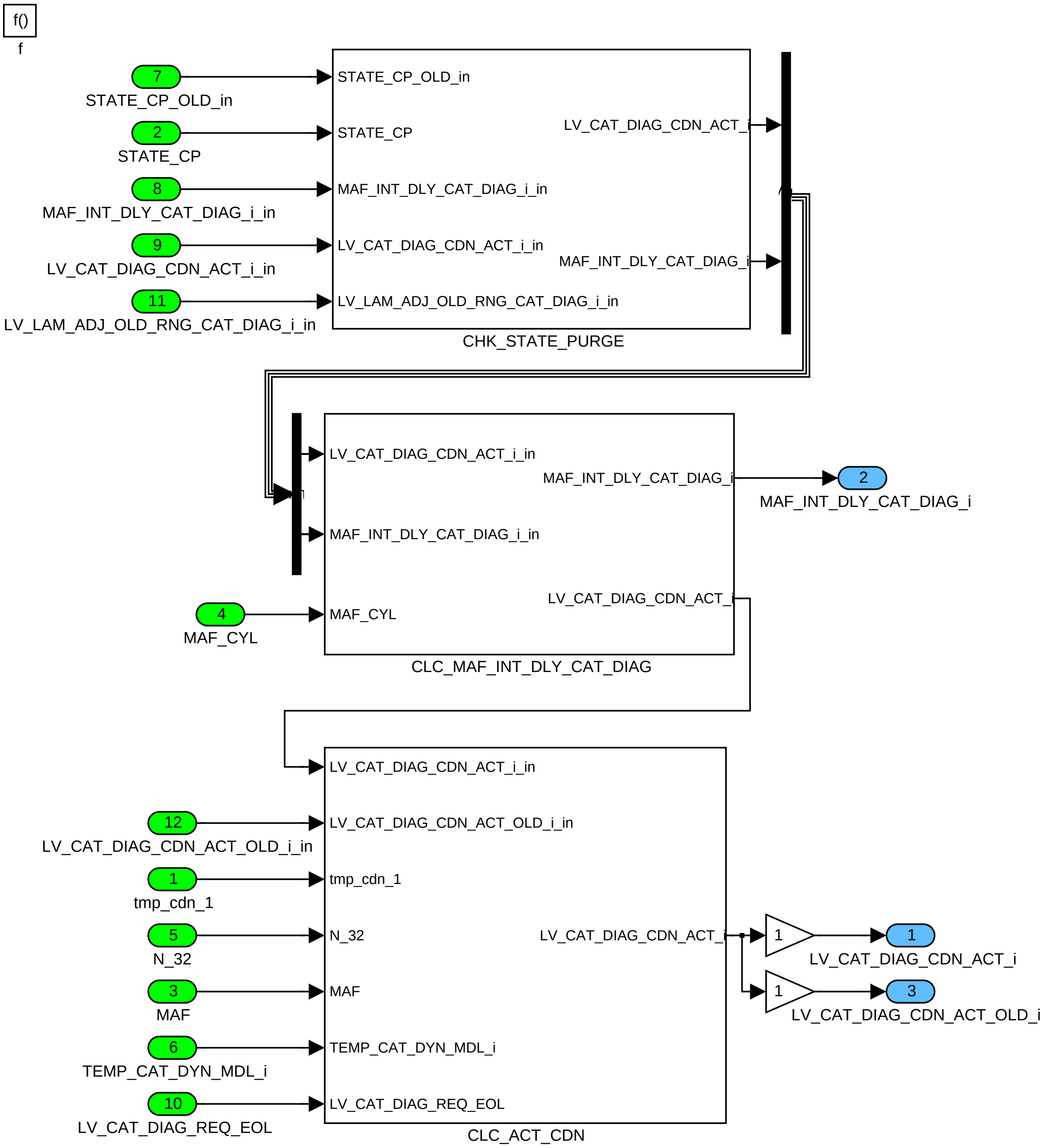


Figure 110 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CLC_MAF_INT_DLY_CAT_DIAG/ CLC_ACT

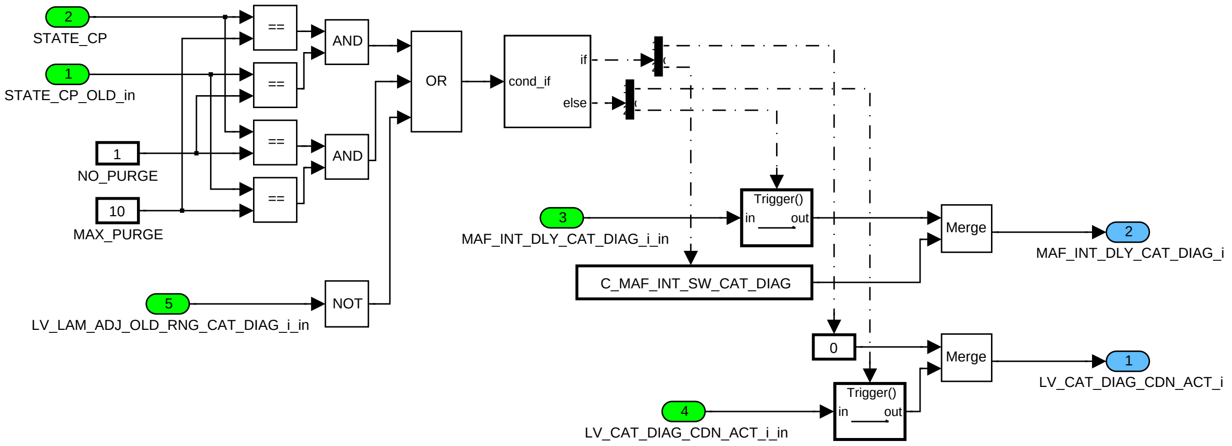


Figure 111 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CLC_MAF_INT_DLY_CAT_DIAG/ CLC_ACT/ CHK_STATE_PURGE

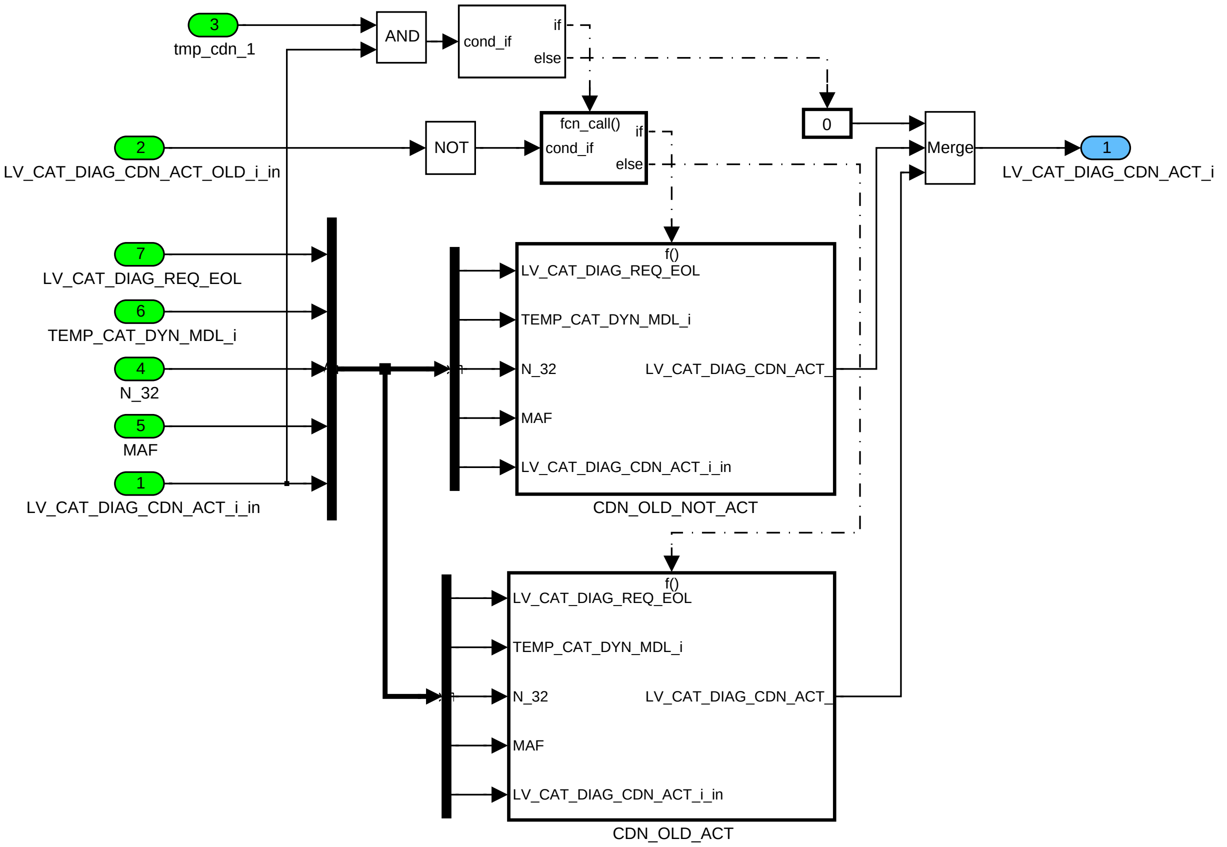


Figure 112 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CLC_MAF_INT_DLY_CAT_DIAG/ CLC_ACT/ CLC_ACT_CDN

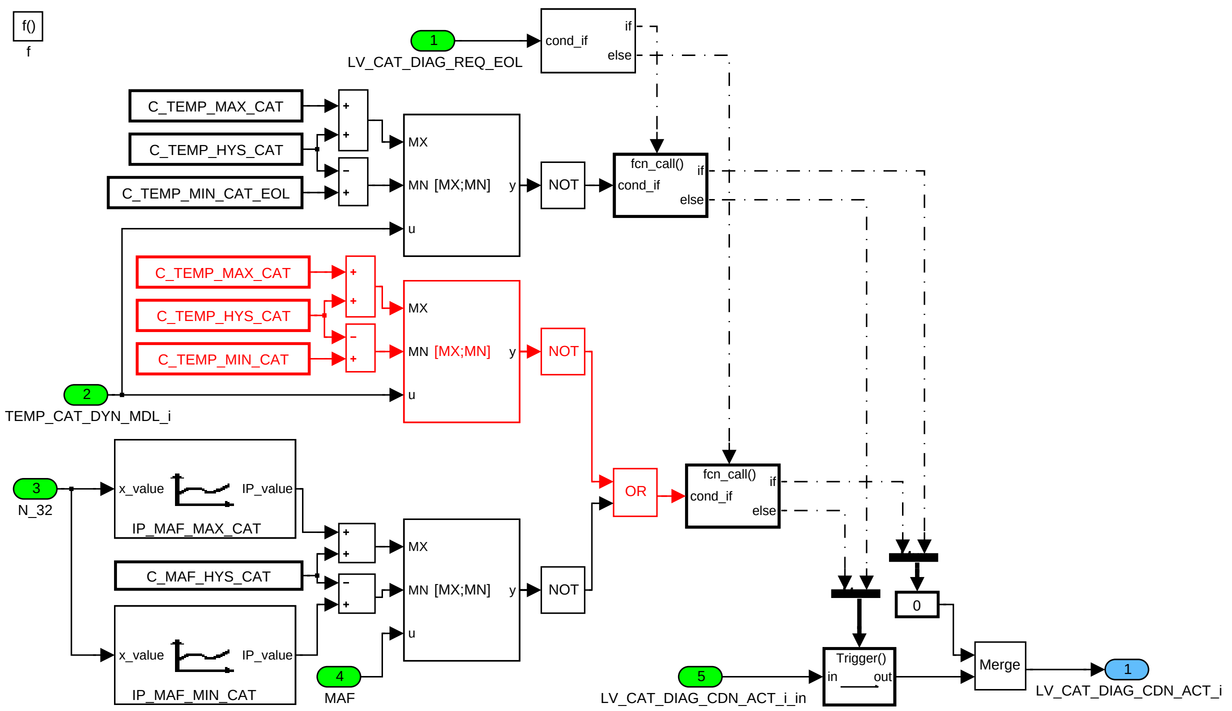


Figure 113 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CLC_MAF_INT_DLY_CAT_DIAG/ CLC_ACT/ CLC_ACT_CDN/ CDN_OLD_ACT

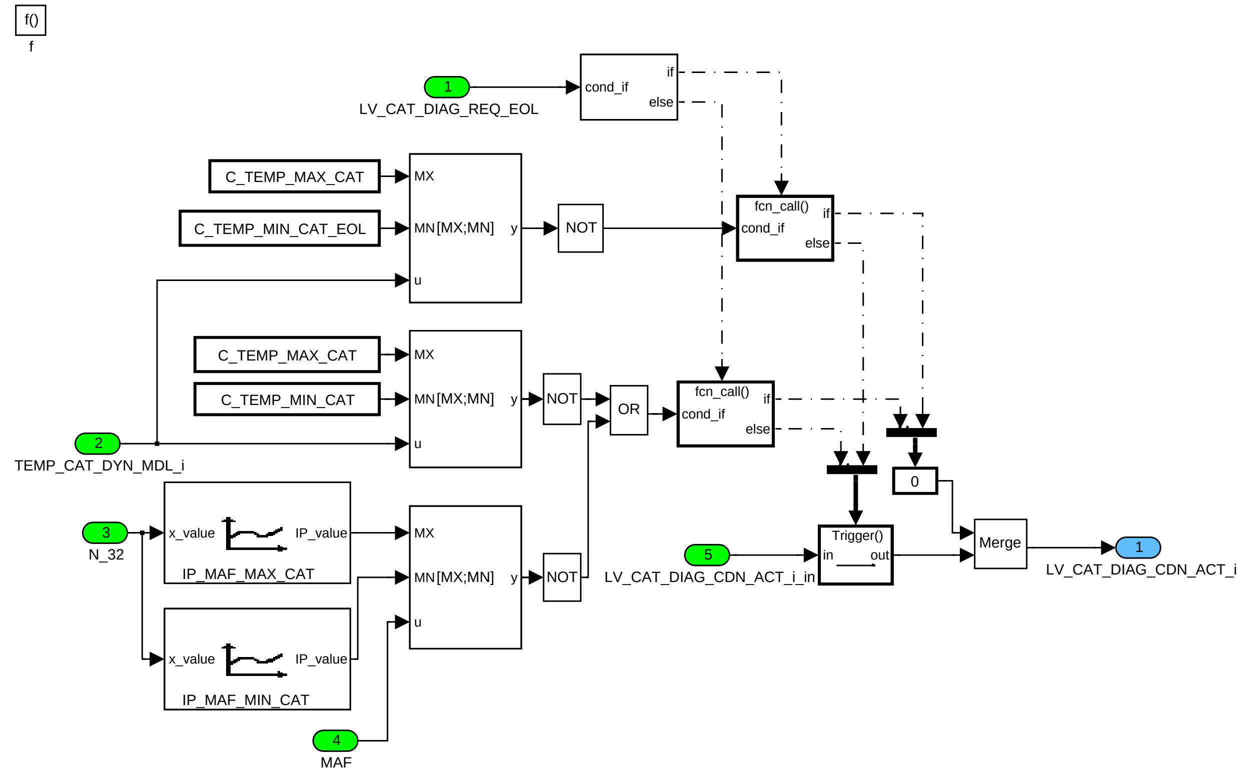


Figure 114 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CLC_MAF_INT_DLY_CAT_DIAG/ CLC_ACT/ CLC_ACT_CDN/ CDN_OLD_NOT_ACT

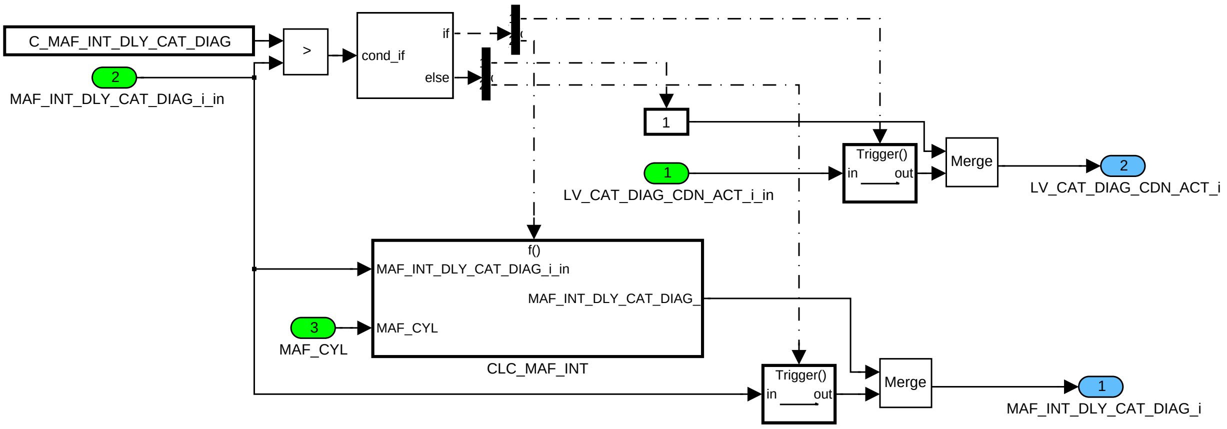


Figure 115 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CLC_MAF_INT_DLY_CAT_DIAG/ CLC_ACT/ CLC_MAF_INT_DLY_CAT_DIAG

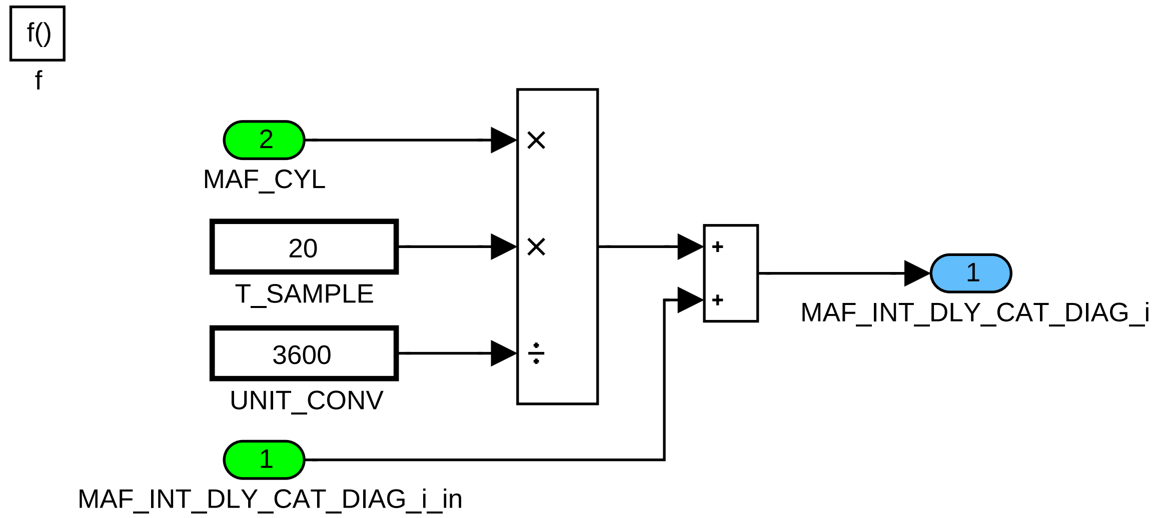


Figure 116 EGTR_FCTDGCEFAC0/ operate/ EGTR_FCTDGCEFAC0_CBK_MNG/ EGTR_FCTDGCEFAC0_CBK_SPC/ CLC_MAF_INT_DLY_CAT_DIAG/ CLC_ACT/ CLC_MAF_INT_DLY_CAT_DIAG/ CLC_MAF_INT

18.12.1.2 SUBFUNCTION: reset

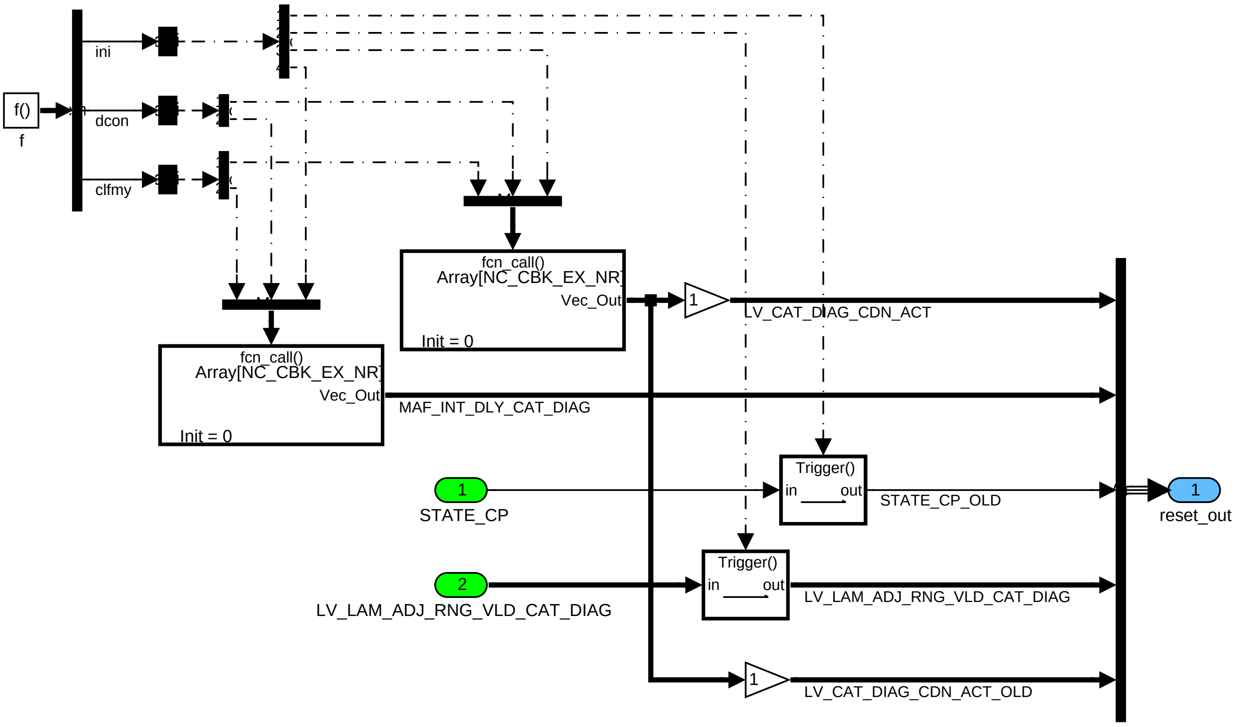


Figure 117 EGTR_FCTDGCEFAC0/ reset

18.13 Catalyst Efficiency Diagnosis (OSC Method) for linear lambda control

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_CAT_DIAG [NC_CBK_EX_NR]	V	0... 1FFH	0... 511	1	[-]
monitoring cycle counter					
CTR_LAMB_CAT_DOWN_SUM [NC_CBK_EX_NR]	V	0... FFFFH	0... 65535	1	[-]
Counter for the summation of LAMB_CAT_DOWN_SUM for each period					
CTR_LAMB_OUT_CAT_DIAG [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Counter of non valid cycle due to LAMB_CAT_DOWN_MV[i] is not in right range, for switch off Cat Diag					
CTR_TOT_CAT_DIAG [NC_CBK_EX_NR]	V/S	0... FFH	0... 255	1	[-]
Total sum of all completed diagnosis cycles since ECU programming					
EFF_CAT_DIAG_HOM [NC_CBK_EX_NR]	O/V/S	0... FFH	0... 1.992188	7.8125e-3	[-]
Catalyst diagnosis value in OSC range (from lambda=1 cat diag)					
FLOW_O2_CAT_DIF [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.0444e-3	31.196e-9	[g/ms]
downstream O2 flow difference to mean value					
LAMB_CAT_DOWN [NC_CBK_EX_NR]	V	0... FFFFH	0... 1.999969	30.176e-6	[-]
Lambda downstream of catalyst; linearization of downstream LS signal					
LAMB_CAT_DOWN_MV [NC_CBK_EX_NR]	V	0... FFFFH	0... 1.999969	30.176e-6	[-]
downstream lambda mean value for complete diagnosis period					
LAMB_CAT_DOWN_ST_CYC [NC_CBK_EX_NR]	V	0... FFFFH	0... 1.999969	30.176e-6	[-]
LAMB_CAT_DOWN[i] value at start of monitoring cycle					
LAMB_CAT_DOWN_SUM [NC_CBK_EX_NR]	V	0... FFFFFFFFH	0... 131072	30.176e-6	[-]
integral of downstream Lambda through 1diagnosis period for calculation of mean value					
LV_LAM_ADJ_CAT_DIAG [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag to switch trim control parameters for OSC/NOX Cat. Diagnosis (only in case of NC_USE_EGTR = 0)					
LV_LAMB_PLS_REQ_CAT_DIAG [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag to switch forced stimulation parameters for OSC/NOX Cat. Diagnosis					
MASS_O2_CAT_DIF_CYC [NC_CBK_EX_NR]	V	0... FFFFH	0... 0.0817765	1.2478e-6	[g]
Mass O2 downsteam catalyst to mean value through 1 diagnosis period					
RATIO_MASS_O2_CAT_DIF [NC_CBK_EX_NR]	V	0... FFFFH	0... 1.999969	30.176e-6	[-]
Weighted MASS_O2_CAT_DIF_CYC[i] value related to O2-load upstream catalyst from lambda controller					
RATIO_MASS_O2_CAT_DIF_SUM [NC_CBK_EX_NR]	V	0... FFFFFFFFH	0... 131072	30.176e-6	[-]
Summation of MASS_O2_CAT_DIF_RATIO[i] over all valid monitoring cycles					
STATE_CAT_DIAG [NC_CBK_EX_NR]	O/V	0 1 2 3 4 5	PASSIVE WAIT CYC_LEAN CYC_RICH END RAMP_UP	-	[-]
State of diagnosis					
STATE_CAT_DIAG_CYC_VLD [NC_CBK_EX_NR]	V	0... 3FH	0... 63	1	[-]
Summary validity bits, current monitoring cycle valid and involved in calculation of diagnosis value; 1H = 1CROSS; 2H = MVDEL; 4H = 2CROSS; 8H = DIFMV; 10H = RANGE; 20H = TEMP					
TEMP_CAT_DIF_DYN_STAT [NC_CBK_EX_NR]	V	0... FFH	-256... 254	2	[°C]
Temperature difference between dynamic and stationary catalyst temperature (from cat. temperature model)					

Input Data:

CTR_STEP_PLS_NOM_CAT_DIAG	C_NR_STEP_PLS_NOM_CAT_DIAG	LV_AFL [NC_CBK_EX_NR]	LV_AT
LV_CAT_DIAG_CDN_ACT [NC_CBK_EX_NR]	LV_CAT_DIAG_REQ_EOL	LV_DC	LV_INH_DIAG_CAT_DIAG [NC_CBK_EX_NR]
MAF_CYL	MAF_HB	NC_CBK_EX_NR	N_32
STATE_LAMB_PLS_DET_VALUE	TEMP_CAT_DYN_MDL [NC_CBK_EX_NR]	TEMP_CAT_STAT_MDL [NC_CBK_EX_NR]	VLS_DELTA_LAM_ADJ_CAT_DIAG [NC_CBK_EX_NR]
VLS_DOWN [NC_CBK_EX_NR]	VLS_SP_LAM_ADJ [NC_CBK_EX_NR]		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CTR_LAMB_OUT_CAT_DIAG	1	0... FFH	0... 255	1	[-]
Threshold to switch off Cat Diag due to to LAMB_CAT_DOWN_MV is not in the right range					
C_CTR_MIN_CAT_DIAG	1	0... 1FFH	0... 511	1	[-]
min. number of required monitoring cycles for catalyst diagnosis					
C_CTR_MIN_CAT_DIAG_EOL	1	0... 1FFH	0... 511	1	[-]
min. threshold of required monitoring cycles for catalyst diagnosis at EOL test					
C_LAMB_CAT_DELTA_MAX	1	0... FFFFH	0... 1.999969	30.176e-6	[-]
max. deviation between LAMB_CAT_MV and setpoint for a valid monitoring cycle					
C_LAMB_CAT_DELTA_MAX_EOL	1	0... FFFFH	0... 1.999969	30.176e-6	[-]
max. deviation between LAMB_CAT_MV and setpoint for a valid monitoring cycle at end-of-line test					
C_LAMB_CAT_MV_DELTA_MAX	1	0... FFFFH	0... 1.999969	30.176e-6	[-]
max. allowed fluctuation between LAMB_CAT_DOWN[i] and LAMB_CAT_DOWN_MV[i] fin the same monitoring cycle					
C_LAMB_CAT_MV_DIF_MAX	1	0... FFFFH	0... 1.999969	30.176e-6	[-]
max. allowed LAMB_CAT_DOWN_MV[i] fluctuation between 2 subsequent monitoring cycles					
C_N_32_CAT_DIAG_DEAC	1	0... FFH	0... 8160	32	[rpm]
Engine speed threshold to deactivate catalyst diagnosis to save ECU performance					
C_O2L_CAT_AFL	1	0... FFFFH	0... 2.616849	39.306e-6	[g]
Adjust parameter for RATIO_MASS_O2L_CAT_DIF calculation					
C_TEMP_CAT_DIF_DYN_MAX	1	0... 7FH	0... 254	2	[°C]
Max. Temperature difference for catalyst diagnosis cycle validity					
C_TEMP_CAT_DIF_DYN_ST	1	0... 7FH	0... 254	2	[°C]
Max. Temperature difference for catalyst diagnosis begin					
IP_FAC_MASS_O2_CAT_DIAG_AT	8x8	0... FFFFH	0... 1.999969	30.176e-6	[-]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1.389e+3	5.447059	[mg/stk]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8.16e+3	32	[rpm]
weight factor of downstream sensor signal integral for automatic transmission					
IP_FAC_MASS_O2_CAT_DIAG_MT	8x8	0... FFFFH	0... 1.999969	30.176e-6	[-]
LDPM_MAF_HB_1_EGTR	8	0... FFH	0... 1.389e+3	5.447059	[mg/stk]
LDPM_N_32_1_EGTR	8	0... FFH	0... 8.16e+3	32	[rpm]
weight factor of downstream sensor signal integral for manual transmission					
IP_FAC_TEMP_CAT_DIF_STAT	8x8	0... 3FFH	0... 4.995117	4.8828e-3	[V]
LDP_TEMP_CAT_DIF_DYN_STAT	8	0... FFH	-256... 254	2	[°C]
LDP_TEMP_CAT_DYN_MDL_IP_FAC_CAT	8	0... FFH	-273.15... 1.767e+3	8	[°C]
Factor to correct cat.diag. value at catalyst temperature different to stationary temperature in this working point					
IP_LAMB_CAT	12	0... FFFFH	0... 1.999969	30.176e-6	[-]
LDP_VLS_TEMP_IP_LAMB_CAT	12	0... 3FFH	0... 4.995117	4.8828e-3	[V]
lambda linearization from downstream LS signal					

General Information

That catalyst diagnosis function is suitable for single branch (only one exhaust bank) and twin branch exhaust lines (two exhaust banks).

For a twin branch exhaust line the function has the same structure for both exhaust lines. The conversion properties of the two catalysts are monitored separately.

NC_CBK_EX_NR labels the number of exhaust banks.

For vector elements, the variable extension _i is used in the model instead of [i] as found in the textual description.

The signals from the control sensor upstream and the associated monitoring sensor downstream from the catalyst are used for catalyst diagnosis.

If a catalyst has good conversion properties, the oscillating lambda signal upstream from the catalyst generated by the lambda controller (which shall ensure defined O2-load of catalyst) downstream is smoothed by the Oxygen Storage Capacity of the catalyst.

If the conversion provided by the catalyst is low due to aging, poisoning by leaded fuel or misfire, then the oscillating lambda signal upstream of the catalyst also exists downstream of the catalyst.

The lambda oscillation downstream of the catalyst is detected by an oxygen sensor (monitoring sensor).

Every monitored catalyst or combination of catalysts shall be indicated when the HC emission exceeds the applicable emission test threshold.

Description:

The diagnosis algorithm is realized with a state machine in order to improve specification understanding and coherence between specification and SW.

Monitoring cycle:(STATE_CAT_DIAG[i]: CYC_LEAN and CYC_RICH)

The first monitoring cycle is started following the rich-lean change (LV_AVL 0->1). A monitoring cycle lasts one complete lambda controller period (2 subsequent lambda richness changes).

Several monitoring cycles are needed to finish the diagnosis.

At activation of Catalyst diagnosis the Forced stimulation is switched to higher Cat diag forced stimulation (soft switch in NC_NOM_CAT_DIAG_STEP cycles) and Trim control is switched to Cat diag parameters (P-share and I-share with own calibration, additionally shift of VLS setpoint depends from EFF_CAT_DIAG_HOM value). VLS_DOWN signal from monitoring Lambda sensor is converted with inverse sensor characteristic LAMB_CAT_DOWN_i.

In the wait state before the first cycle after every reactivation (rich-lean change LV_AFL) only LAMB_CAT_DOWN_MV_i is calculated. LAMB_CAT_DOWN_SUM_i is obtained by the summation of the LAMB_CAT_DOWN_i over a complete monitoring cycle. If LAMB_CAT_DOWN_MV_i agree with VLS_DOWN setpoint, catalyst diagnosis could start. WAIT state turns to RAMP_UP state and the O2load (realized by forced stimulation) ramps up from nominal value to catalyst diagnosis value. After ramp up of O2load finished, forced stimulation informs the catalyst diagnosis by forced stimulation state, really evaluated catalyst diagnosis cycles could start (CYC_LEAN/RICH).

LAMB_CAT_DOWN_MV_i is the mean value of the LAMB_CAT_DOWN_i - O2_load downstream the Catalyst calculated as inverse characteristic of binary lambda sensor from VLS_DOWN_i signal for one monitoring cycle. The calculation of MASS_O2_CAT_DIF_CYC_i is based on LAMB_CAT_DOWN_MV_i of the previous monitoring cycle. MASS_O2_CAT_DIF_CYC_i is the integrated difference between actual LAMB_CAT_DOWN_i signal and LAMB_CAT_DOWN_MV_i of the previous monitoring cycle.

Illustration monitoring cycle

Validity check: (STATE_CAT_DIAG_i: Transition CYC_RICH -> CYC_LEAN)
The obtained MASS_O2_CAT_DIF_CYC[i] of a monitoring cycle is only used for the calculation of the diagnosis value if the validity check was passed (STATE_VLD_CAT_DIAG_i reach the minimum value for validity)
To pass the validity check it is necessary that:
LAMB_CAT_DOWN[i] must cross at least once LAMB_CAT_DOWN_MV[i] of the previous monitoring: 0.bit of STATE_VLD_CAT_DIAG_i is set (bit '1CROSS').
OR
The deviation between LAMB_CAT_DOWN[i] and LAMB_CAT_DOWN_MV[i] of the last monitoring cycle is smaller than C_LAMB_CAT_MV_DELTA_MAX: 1.bit of STATE_VLD_CAT_DIAG_i is set (bit 'MVDEL').
OR
LAMB_CAT_DOWN[i] cross two times LAMB_CAT_DOWN_MV[i] of the previous monitoring (periodically braking down of VLS_DOWN occurs): 2.bit of STATE_VLD_CAT_DIAG_i is set (bit '2CROSS').
AND
Difference between actual and the LAMB_CAT_DOWN_MV[i] of the last monitoring cycle is not too high (if LAMB_CAT_DOWN[i] cross two times LAMB_CAT_DOWN_MV[i], higher difference allowed): 3.bit of STATE_VLD_CAT_DIAG_i is set (bit 'DIVMV').
AND
LAMB_CAT_DOWN_MV[i] is in the right range 4.bit of STATE_VLD_CAT_DIAG_i is set (bit 'RANGE').
AND
Catalyst temperature dynamic is limited: 5.bit of STATE_VLD_CAT_DIAG_i is set (bit 'TEMP').
Illustration validity check

Calculation of diagnosis value: (STATE_CAT_DIAG[i]: Transition CYC_RICH -> CYC_LEAN)
Monitoring cycle counter CTR_CAT_DIAG[i] is incremented for each valid monitoring cycle.
MASS_O2_CAT_DIF_CYC[i] is weighted by engine speed and load (IP_FAC_MASS_O2_CAT_DIAG_XX). From that calculation MASS_O2_CAT_DIF_RATIO[i] results. MASS_O2_CAT_DIF_RATIO[i] is summed up for all monitoring cycles. That summation give MASS_O2_CAT_DIF_RATIO_SUM[i]. Finally, the average of MASS_O2_CAT_DIF_RATIO_SUM[i] over all valid monitoring cycles give the diagnosis value EFF_CAT_DIAG_HOM[i]
End of diagnosis: (STATE_CAT_DIAG[i]: Transition CYC_LEAN -> END)
If the necessary number of valid monitoring cycles CTR_CAT_DIAG[i] reaches C_CTR_MIN_CAT_DIAG the diagnosis is ended for that driving cycle and the catalyst diagnosis value EFF_CAT_DIAG_HOM[i] is calculated and stored in the EEPROM.
The monitoring counter CTR_CAT_DIAG[i] is used to set the readiness code. For this reason C_CTR_MIN_CAT_DIAG must be reached in the emission test cycle.
State diagram: STATE_CAT_DIAG[i]

Note: The priorities of the conditions to change between states are defined by order described in the STATEFLOW diagram below. The condition 8 as transition to PASSIVE is twice available and contain the same conditions each.
The recurence time is defined by T_SAMPLE in [ms].
Application Assistance
For OBD II O2 sensor diagnosis: C_CTR_MIN_CAT_DIAG has to be applied higher then the thresholds C_SUM_CYC_MIN_DIAG and C_SUM_CYCNR_MIN_DIAG (LV_LS_FRQ_DIAG_READY -> 1 and LV_LS_SWT_DIAG_READY -> 1).
They have to be reached before STATE_CAT_DIAG[i] is set to END.
Hint:
An active catalyst diagnosis can disable the module Downstream Fuel Trim Regulation also a additional lambda offset can be imposed on the basic lambda setpoint in the module Basic Lambda setpoint . When the catalyst diagnosis is completed the above functions return to their normal operation.
Overview: Chronological order of diagnosis algorithm:
Triggering of catalyst diagnosis calculation by LV_AFL[i] 0->1...
%setpoint of the O2-load with CAT DIAG for the lambda controller (output, not used here)
1) LAMB_CAT_DOWN[i] = IP_LAMB_CAT(VLS_DOWN[i]) %every recurrence
2) LAMB_CAT_DOWN_SUM[i] n+1 = LAMB_CAT_DOWN_SUM[i] n+ LAMB_CAT_DOWN[i] %every recurrence
3) LAMB_CAT_DOWN_MV[i] = LAMB_CAT_DOWN_SUM[i] / CTR_LAMB_CAT_DOWN_SUM[i]
(after end of current diagnosis period)
4) FLOW_O2_CAT_DIF[i] = 0.23 * (1/3600)[g/ms] * I1/LAMB_CAT_DOWN_MV[i] - 1/LAMB_CAT_DOWN[i] I * MAF_CYL[kg/h]
5) MASS_O2_CAT_DIF_CYC[i]n+1 = MASS_O2_CAT_DIF_CYC[i] n + FLOW_O2_CAT_DIF[i]*T_SAMPLE[ms] %every recurrence
6) RATIO_MASS_O2_CAT_DIF [i] = MASS_O2_CAT_DIF_CYC[i] * IP_FAC_TEMP_CAT_DIF_STAT / (C_O2L_CAT_AFL * IP_FAC_MASS_O2_CAT_DIAG_XX) (with XX for AT or MT (after end of current diagnosis period)
7) RATIO_MASS_O2_CAT_DIF_SUM[i] n+1 = RATIO_MASS_O2_CAT_DIF_SUM[i] n + MASS_O2_CAT_DIF_RATIO[i]
(after end of current diagnosis period and passed validity check)
8) EFF_CAT_DIAG_HOM[i] = MASS_O2_CAT_DIF_RATIO_SUM[i] / CTR_CAT_DIAG[i]
(Catalyst diagnosis value after diagnosis finished)
Scheduling of formulas:
1+2: states WAIT, RAMP_UP, CYC_LEAN, CYC_RICH
3: state WAIT, transitions RAMP_UP or CYC_RICH to CYC_LEAN
4+5: states CYC_LEAN, CYC_RICH
6+7: transition CYC_RICH to CYC_LEAN
8: transition CYC_LEAN to END
Several system events will trigger a function response:
EGTR_FCTDGCEF0__RST (reset): reset
EGTR_FCTDGCEF0__20MS (regular trigger every 20 msec): operate
EGTR_FCTDGCEF0__CLFMY (clear failure memory): reset
EGTR_FCTDGCEF0__STB (standby; ECU programming or fault): reset, init_nvmy
EGTR_FCTDGCEF0__UNVMY (update non volatile memory): save_nvmy

EGTR_FCTDGCEF0__RNVMY (read non volatile memory): read_nvmy

Application Conditions

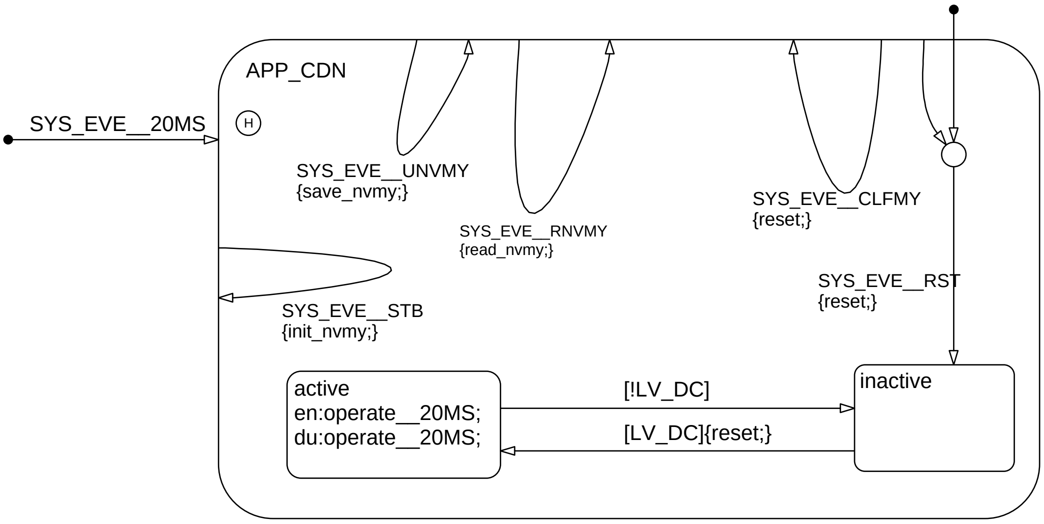
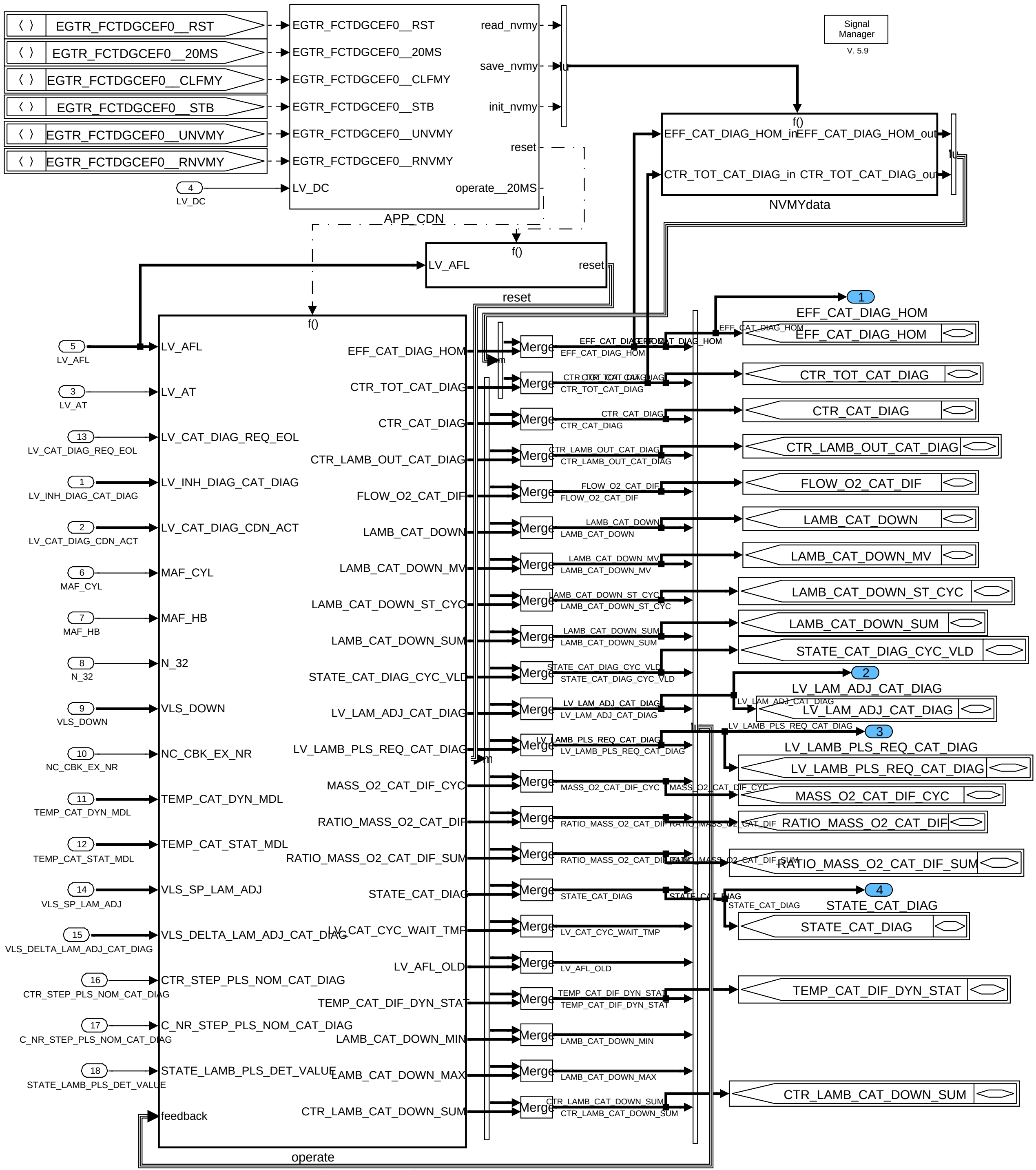


Figure 118: Path: EGTR_FCTDGCEF0/APP_CDN/Chart

Function description



SDA_SRS / SDA V 5.0 Pre Release 07−Oct−2005

Figure 119: Path: EGTR_FCTDGCEF0

18.13.1 SUBFUNCTION: operate

18.13.1.1 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG

18.13.1.1.1 Checking state:

Checking the transition conditions with aid of the state machine. Then, in case of changing the actual state, performing the respective transition actions and state actions for the new state.

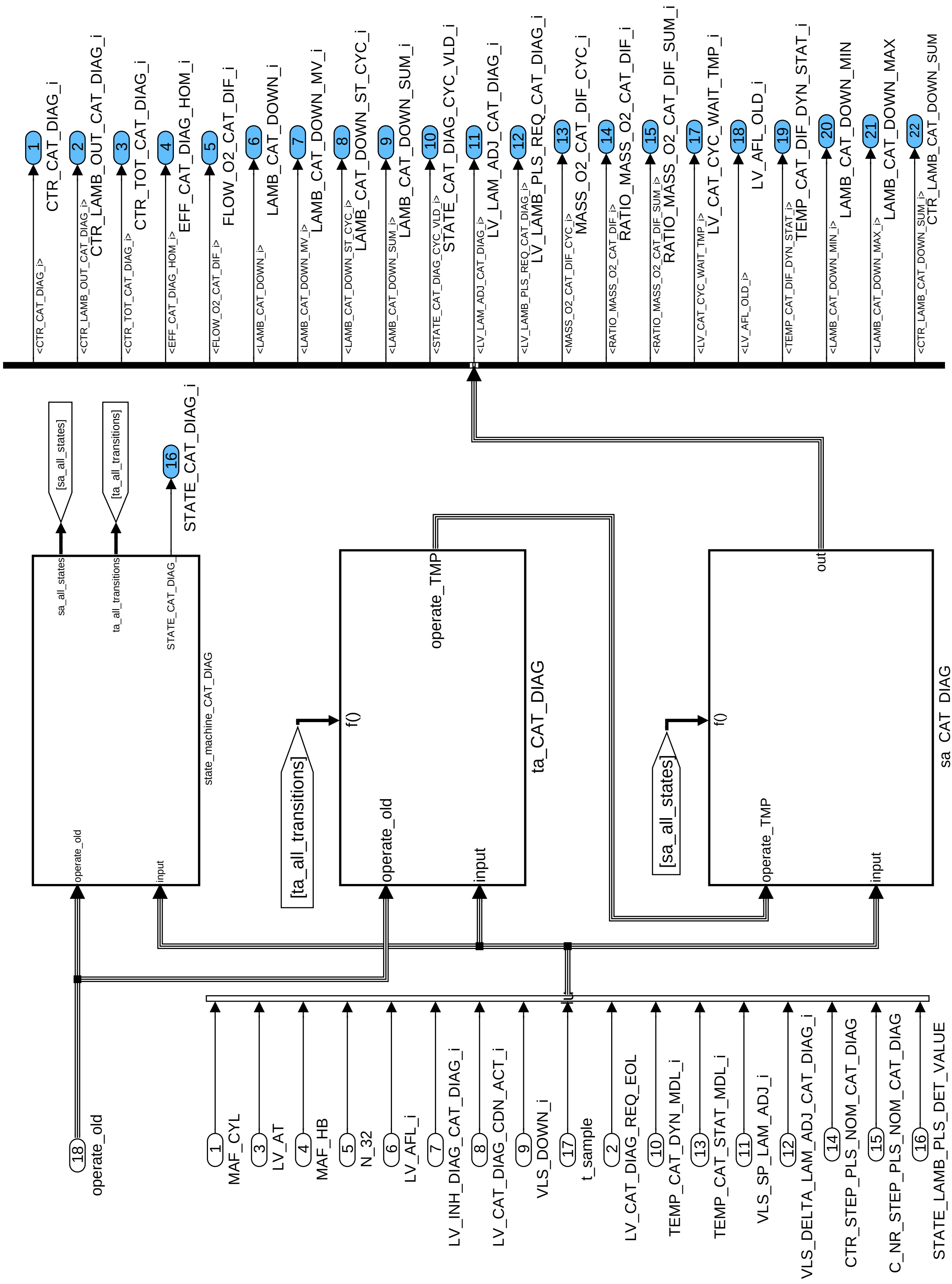


Figure 120: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC

18.13.1.1.1.1 State machine:

Checking the respective transition conditiones in dependence on current state and evaluation of the new state in compliance with exclusion thresholds, i.e. with the engine speed, the counter for the minimum required monitoring cycles to complete the catalyst diagnosis and the counter numerating subsequently following invalid monitoring cycles.
Afterwards issue of the trigger vectors for transition actions and state actions.

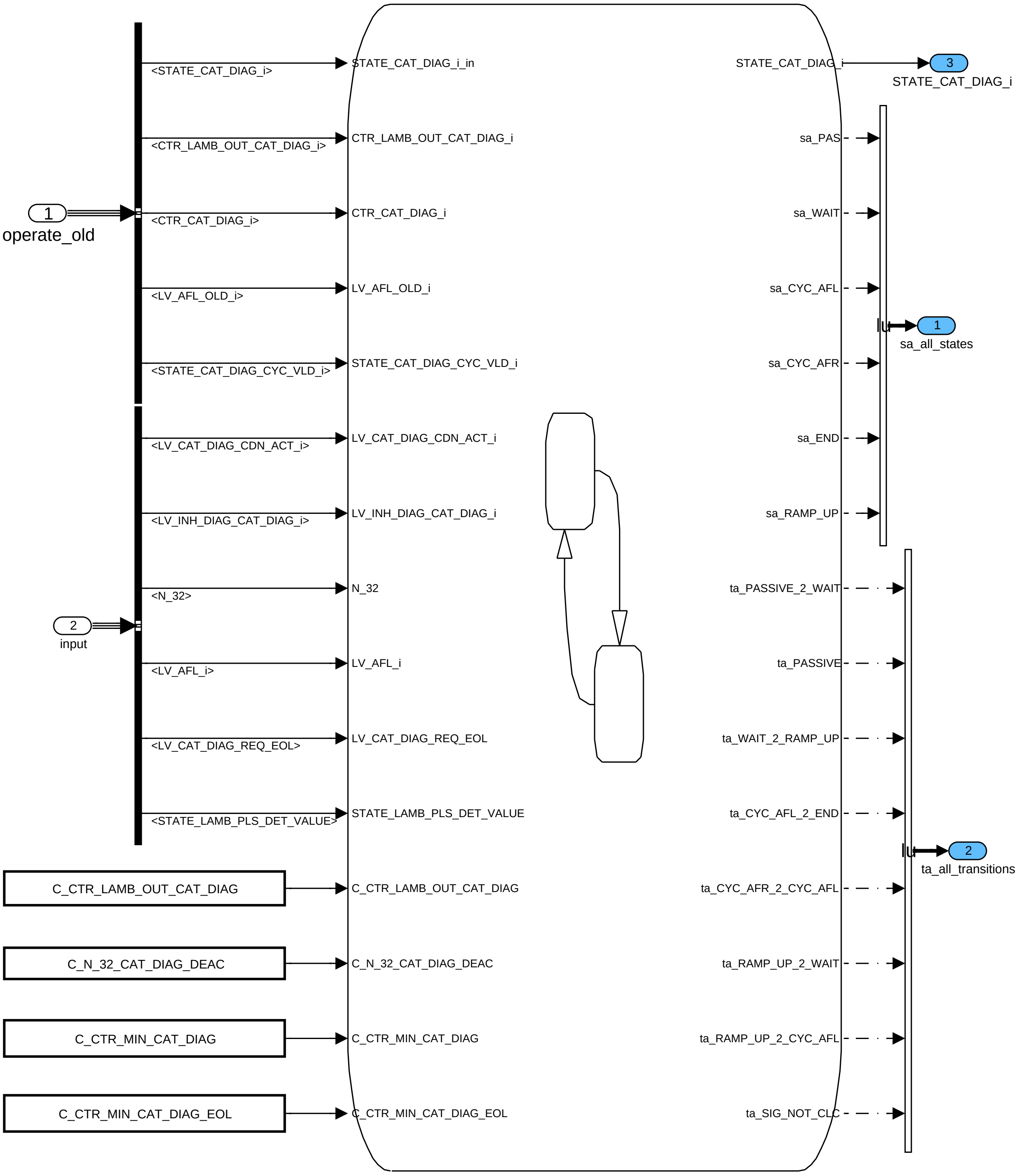


Figure 121: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/state_machine_CAT_DIAG

18.13.1.1.1.1.1 No title given

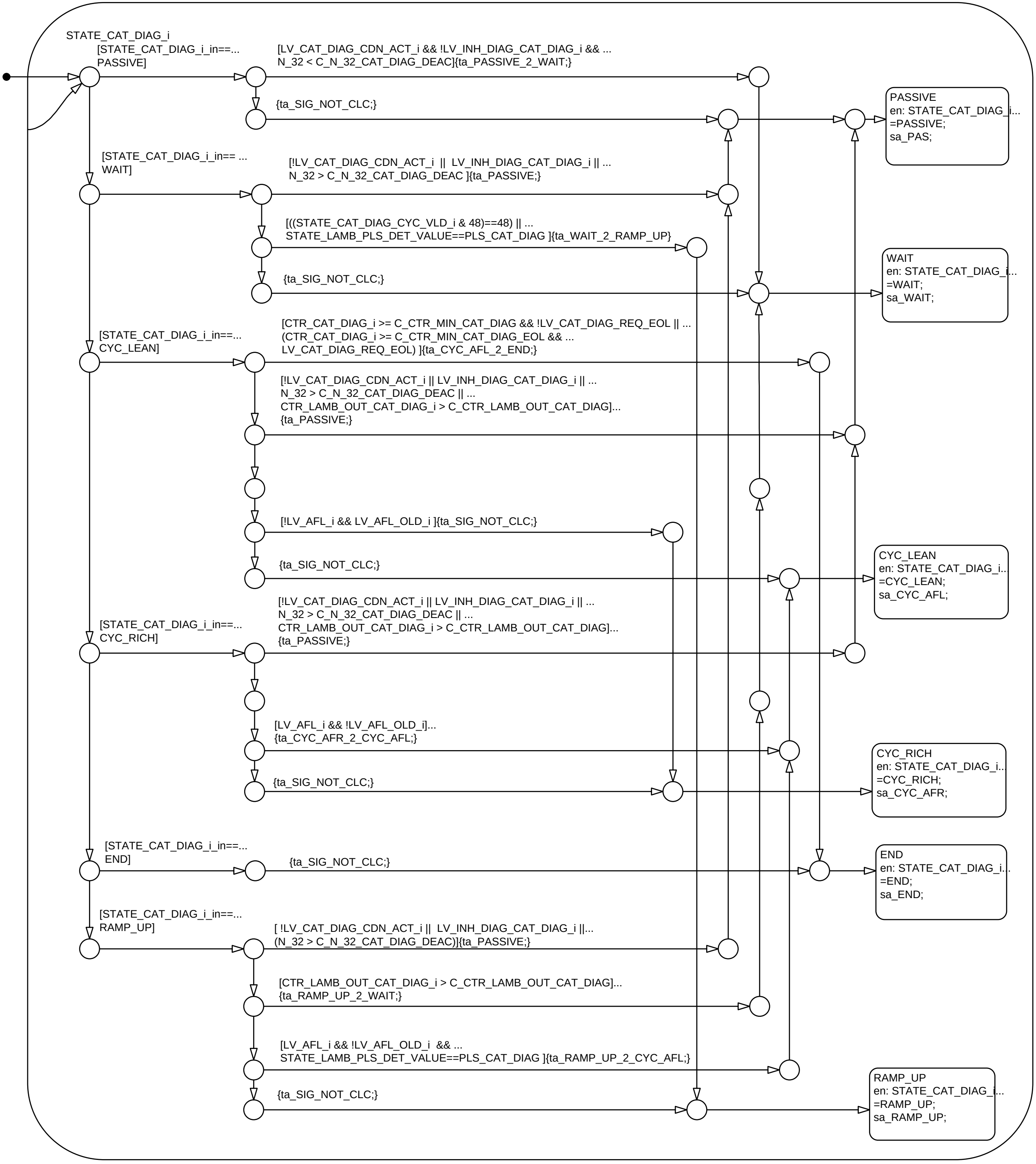


Figure 122: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/state_machine_CAT_DIAG/Chart

18.13.1.1.1.2 Transition actions:

Performing the concerned transition action induced by the issued trigger vector.

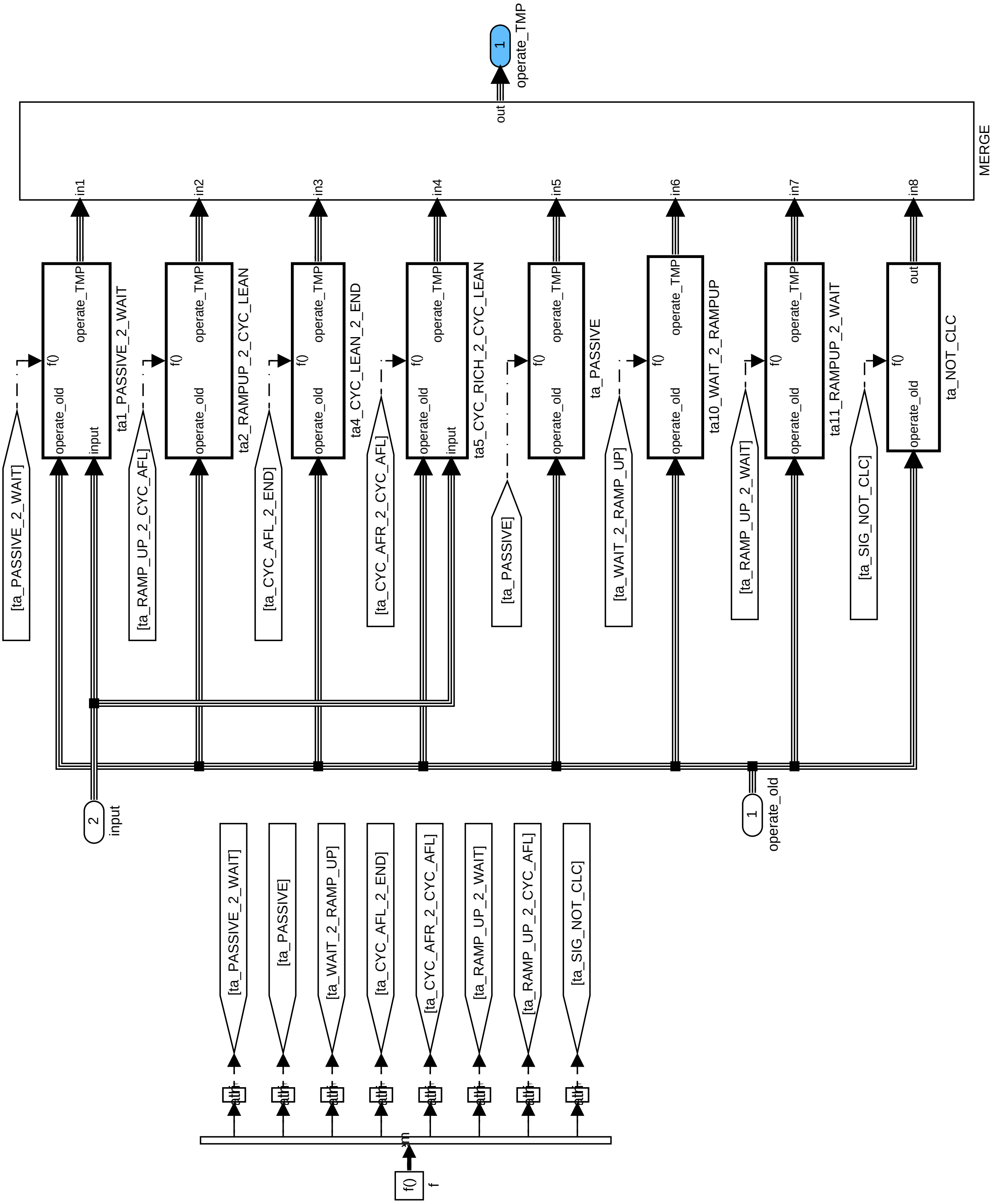


Figure 123: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG

18.13.1.1.1.2.1 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/TA_CAT_DIAG/TA1_PASSIVE_2_WAIT

18.13.1.1.1.2.1.1 Transition action PASSIVE to WAIT:

Setting the temporary WAIT state flag to zero to indicate the downstream lambda sum of this monitoring cycle not to be complete (no valid mean value for the next cycle).
Initializing the downstream lambda sum and its counter to zero.
Storing the old value of LV_AFL for edge detection.

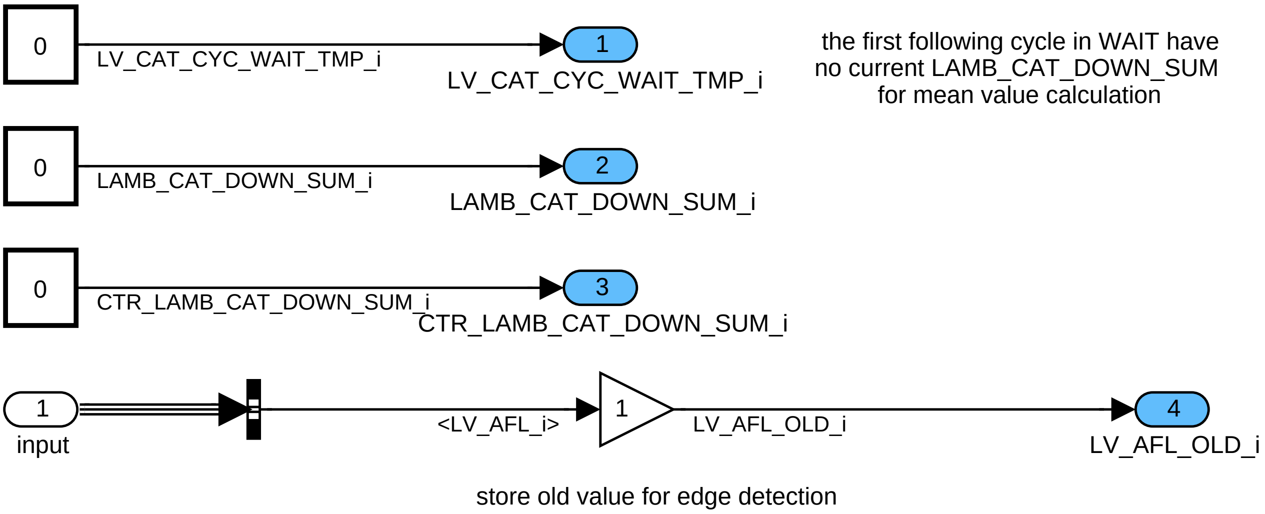


Figure 124: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta1_PASSIVE_2_WAIT/CLC

18.13.1.1.1.2.2 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/TA_CAT_DIAG/TA2_RAMPUP_2_CYC_LEAN

18.13.1.1.1.2.2.1 Transition action RAMP_UP to CYC_LEAN:

Calculation of the downstream lambda mean value of last cycle. Initializing the downstream lambda sum, its counter and the related oxygen mass to zero.
Storing the downstream lambda signal at start of the monitoring cycle and initializing the concerning min/max values to the same value.
Resetting the validity bits with regard to crossing events, setting the validity bit with regard to the permitted difference between the actual downstream lambda signal and the mean value of last cycle.

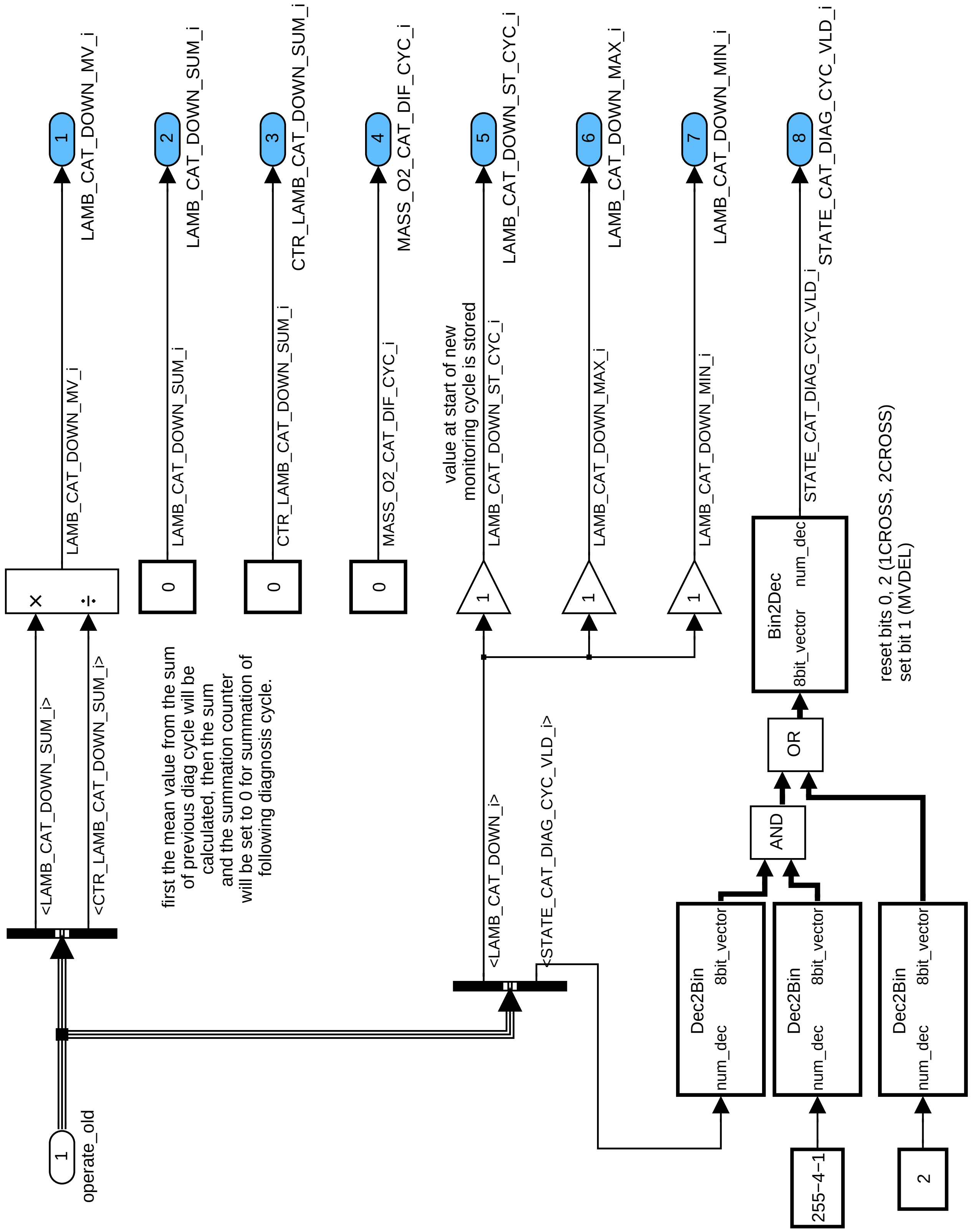


Figure 125: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta2_RAMPUP_2_CYC_LEAN/CLC

18.13.1.1.1.2.3 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/TA_CAT_DIAG/TA4_CYC_LEAN_2_END

18.13.1.1.1.2.3.1 Transition action CYC_LEAN to END:

Calculation of the new diagnosis value and incrementing the total number of completed diagnosis cycles since ECU programming by one.
Resetting forced stimulation and trim control parameters to nominal values and all validity bits to zero.

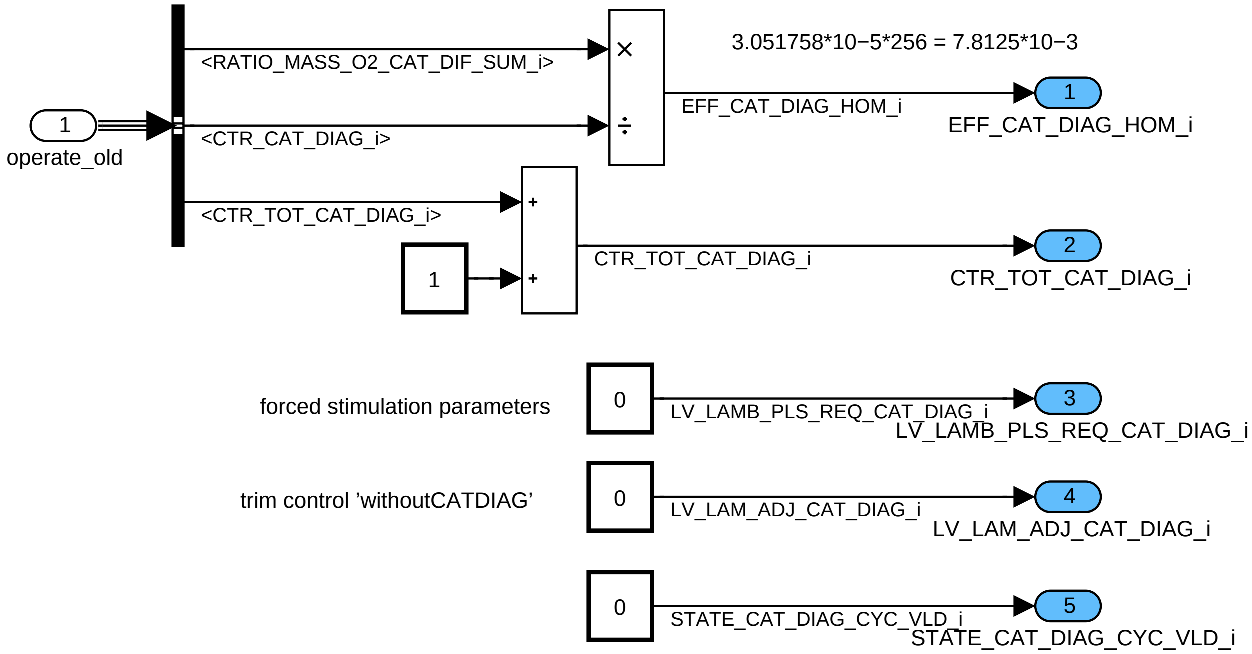


Figure 126: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta4_CYC_LEAN_2_END/CLC

18.13.1.1.1.2.4 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/TA_CAT_DIAG/TA5_CYC_RICH_2_CYC_LEAN

18.13.1.1.1.2.4.1 Transition action CYC_RICH to CYC_LEAN:

Start of a new monitoring cycle.
Calculation of the downstream lambda mean value of last cycle.
Evaluation of the diagnosis value merely related to last monitoring cycle. Computation of the actual temperature difference between dynamic and stationary catalyst temperature.
Incrementing/resetting the number of cycles showing lambda to be out of range and setting the validity bits. In case of validity sum up the diagnosis value and increment the counter of valid cycles by one.
Initializing the single cycle bound sums and counters for next cycle.

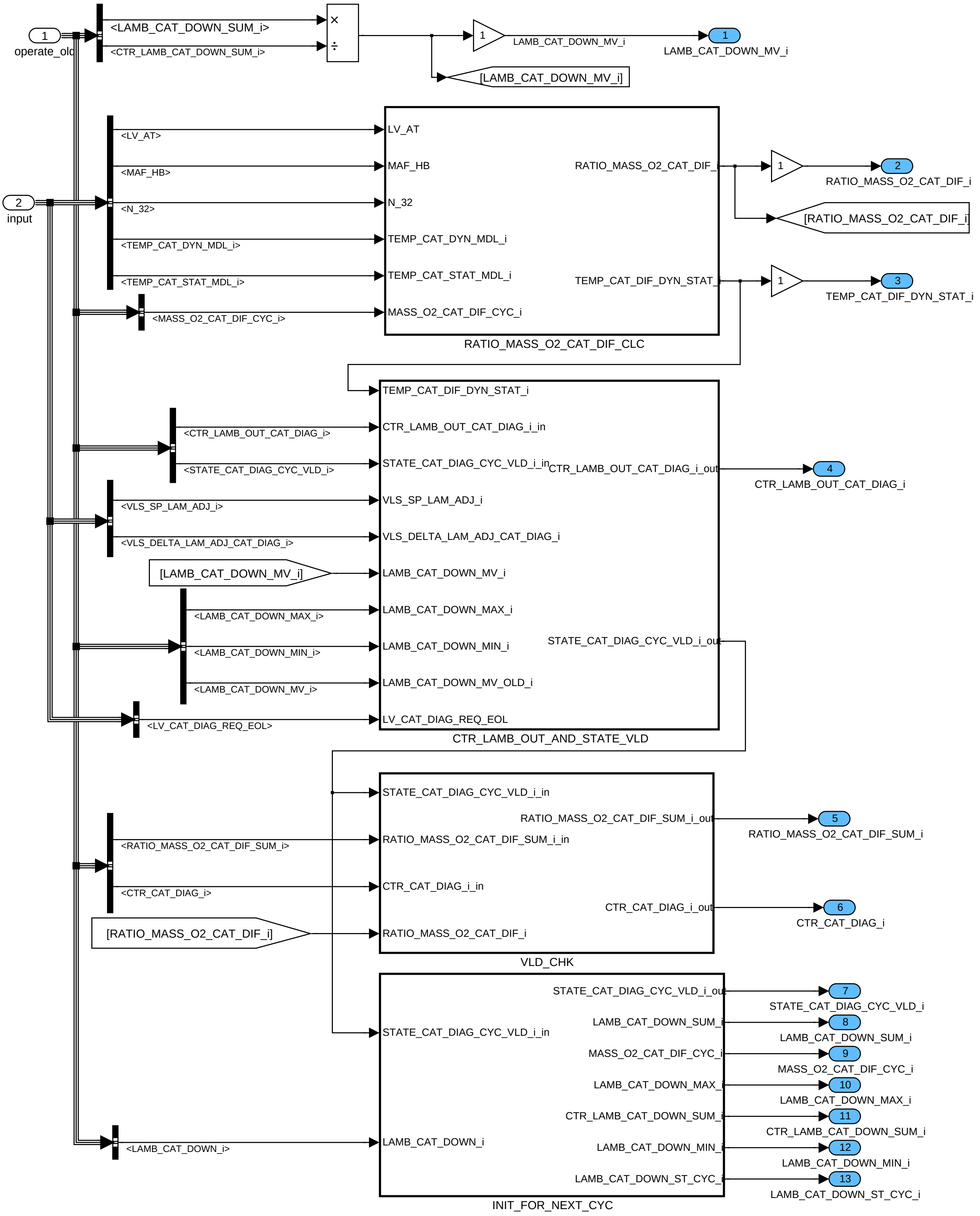
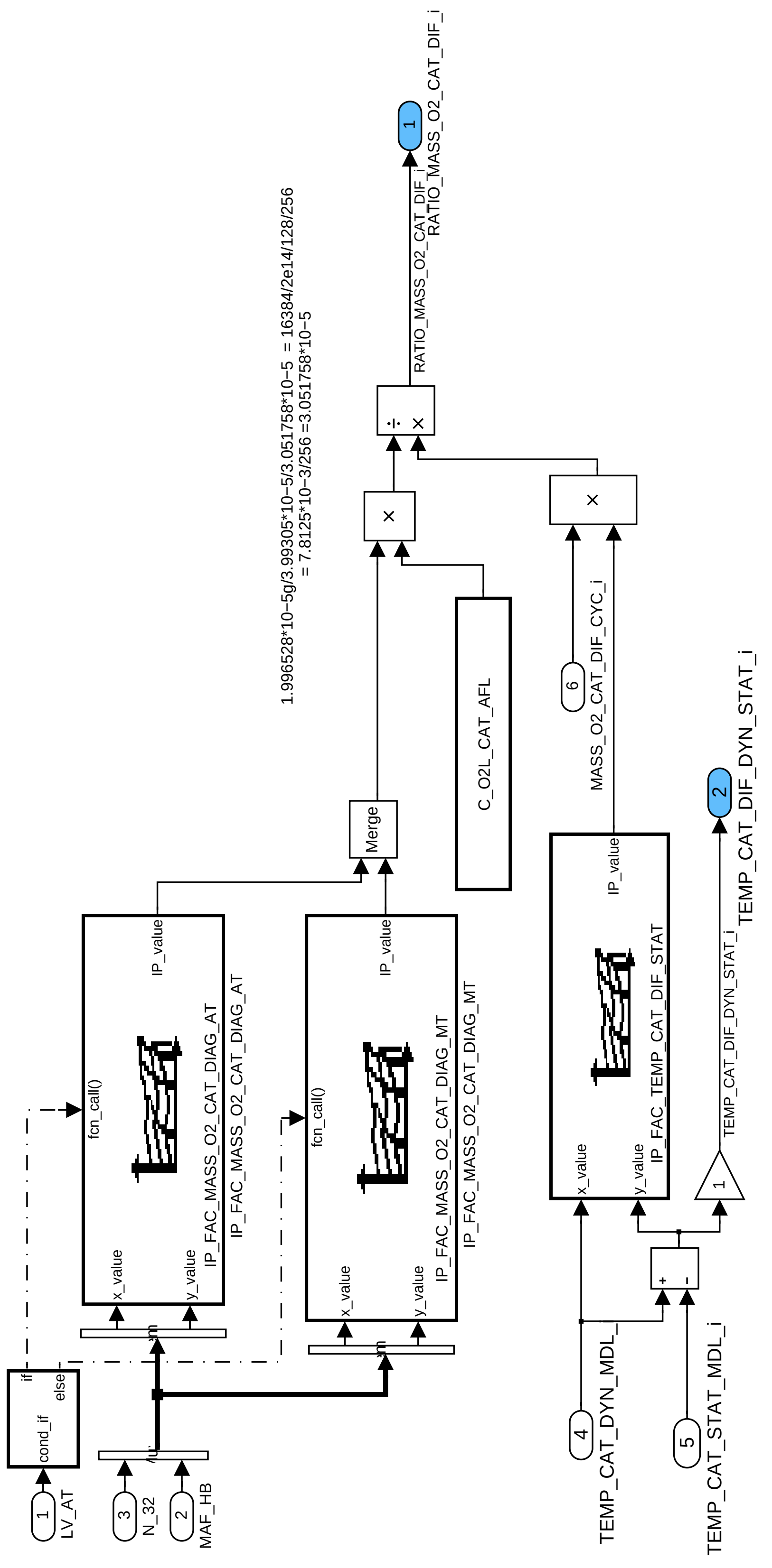


Figure 127: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta5_CYC_RICH_2_CYC_LEAN/CLC

18.13.1.1.1.2.4.1.1 Weighting the diagnosis value:

Weighting the diagnosis value related to last monitoring cycle, taking into account the concerning working point and temperature dynamics.



Chapter	Baseline	Include File
Exhaust gas treatment system	4DC3940S	30B00A02.00E

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Figure 128: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta5_CYC_RICH_2_CYC_LEAN/CLC/RATIO_MASS_O2_CAT_DIF_CLC

18.13.1.1.1.2.4.1.2 Monitoring cycle validity:

Computing the absolute difference between downstream lambda mean value and the corrected lambda setpoint and comparing it to the stored maximally admissible difference. Setting the RANGE bit in dependence on the result. Incrementing/resetting the counter of monitoring cycles showing lambda to be out of range.
Comparing the stat/dyn temperature difference to the stored maximally allowed value and setting the TEMPERATURE bit accordingly.
Computing the difference between downstream lambda min/max and dividing by eight. Adding this to the stored maximum allowed if the temperature range is o.k. Comparing the result to the absolute difference between actual and old mean value and setting the cycle validity bit in correspondence with.

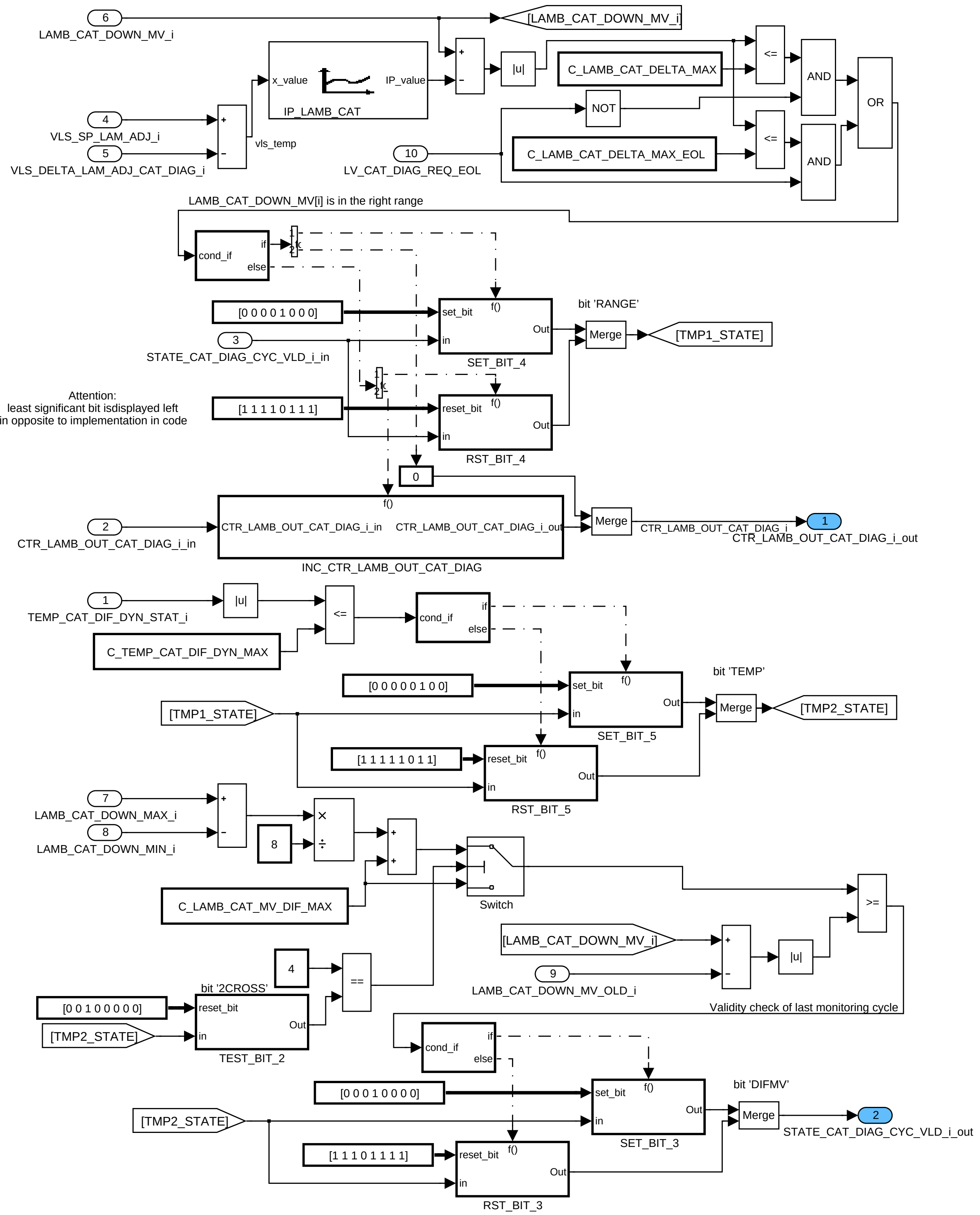


Figure 129: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta5_CYC_RICH_2_CYC_LEAN/CLC/CTR_LAMB_OUT_AND_STATE_VLD

18.13.1.1.1.2.4.1.2.1 Incrementing/resetting counter:

Incrementing/resetting the number of cycles showing lambda to be out of range.

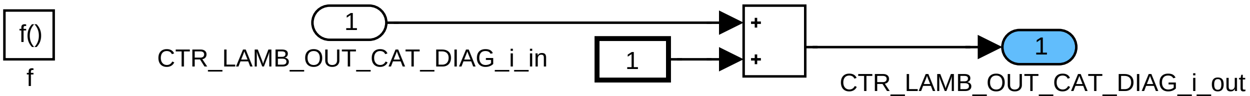


Figure 130: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta5_CYC_RICH_2_CYC_LEAN/CLC/CTR_LAMB_OUT_AND_STATE_VLD/INC_CTR_LAMB_OUT_CAT_DIAG

18.13.1.1.1.2.4.1.3 Validity Check:

Checking last cycle for validity. If the last cycle is valid summing up the diagnosis value and adding one to the number of valid monitoring cycles.

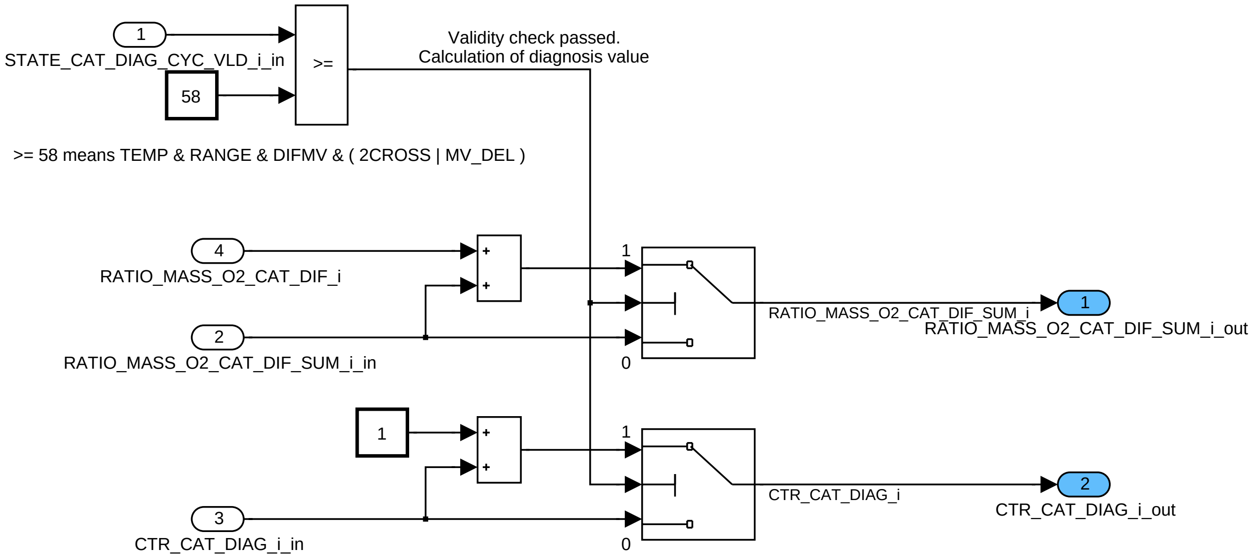


Figure 131: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta5_CYC_RICH_2_CYC_LEAN/CLC/VLD_CHK

18.13.1.1.1.2.4.1.4 Initialization for next cycle:

Resetting the validity bits with regard to crossing events, setting the validity bit with regard to the permitted difference between the actual downstream lambda signal and the mean value of last cycle.
Initializing the downstream lambda sum, its counter and the related oxygen mass to zero.
Storing the downstream lambda value at start of the monitoring cycle and initializing the concerning min/max values to the same value to detect crossing events.

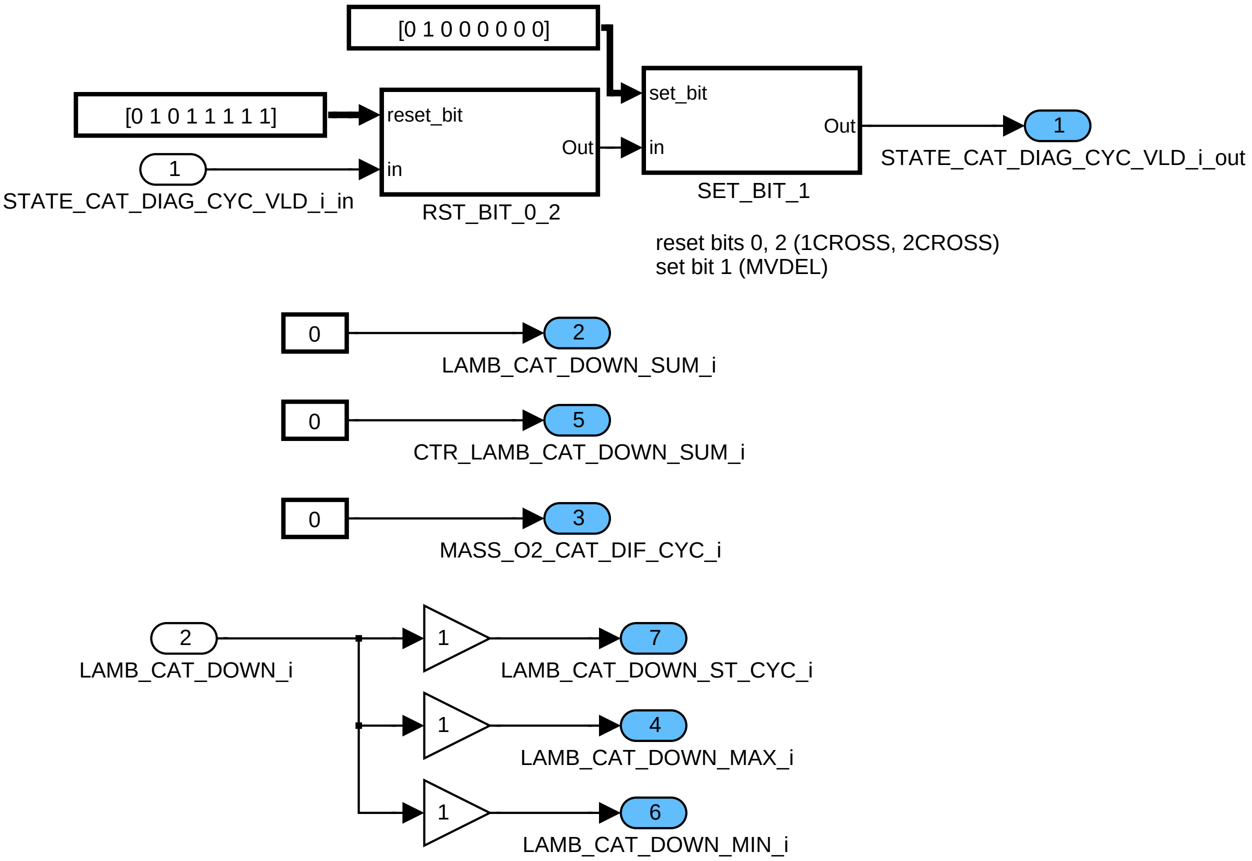


Figure 132: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta5_CYC_RICH_2_CYC_LEAN/CLC/INIT_FOR_NEXT_CYC

18.13.1.1.1.2.5 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/TA_CAT_DIAG/TA_PASSIVE

18.13.1.1.1.2.5.1 Transition action X to PASSIVE:

Resetting forced stimulation and trim control to nominal mode and all validity bits to zero.

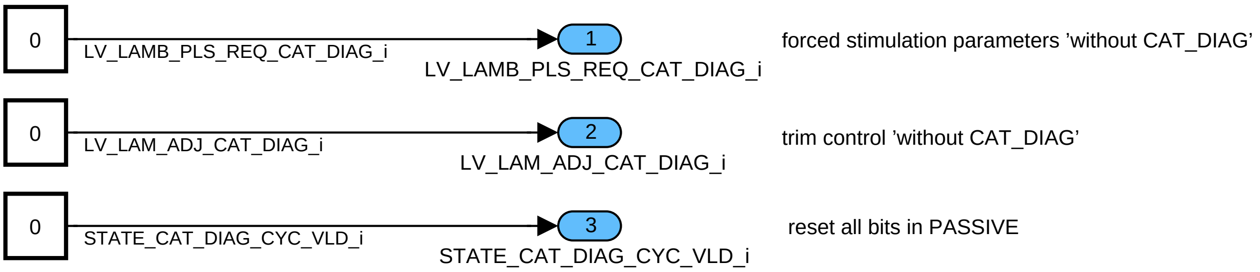


Figure 133: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta_PASSIVE/CLC

18.13.1.1.1.2.6 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/TA_CAT_DIAG/TA10_WAIT_2_RAMPUP

18.13.1.1.1.2.6.1 Transition action WAIT to RAMP_UP:

Setting forced stimulation and trim control to catalyst diagnosis mode and the number of cycles showing lambda to be out of range to zero.

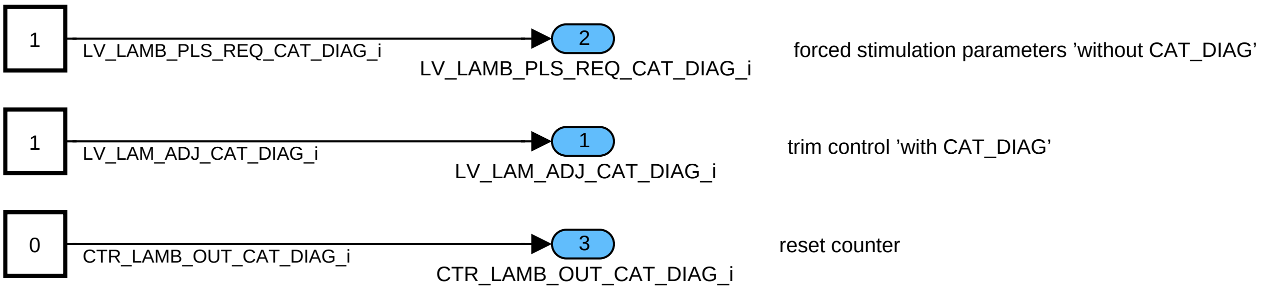


Figure 134: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta10_WAIT_2_RAMPUP/CLC

18.13.1.1.1.2.7 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/TA_CAT_DIAG/TA11_RAMPUP_2_WAIT

18.13.1.1.1.2.7.1 Transition action RAMP_UP to WAIT:

Resetting forced stimulation and trim control to nominal mode because of lambda being out of range. Resetting all validity bits to zero.

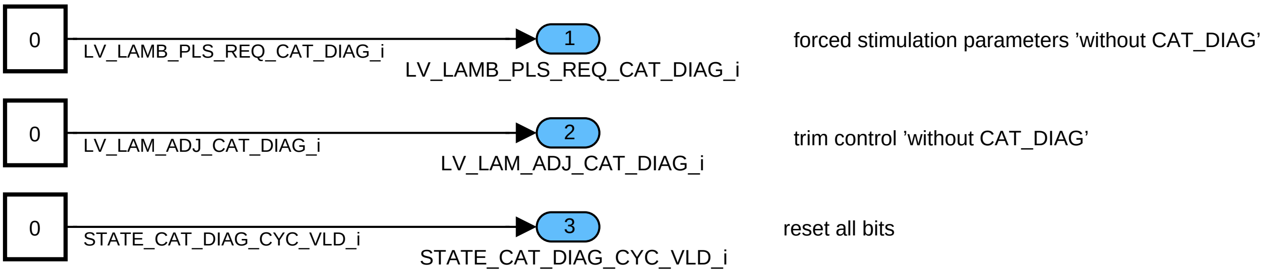


Figure 135: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ta_CAT_DIAG/ta11_RAMPUP_2_WAIT/CLC

18.13.1.1.1.3 State actions:

Performing the concerned state action induced by the issued trigger vector.

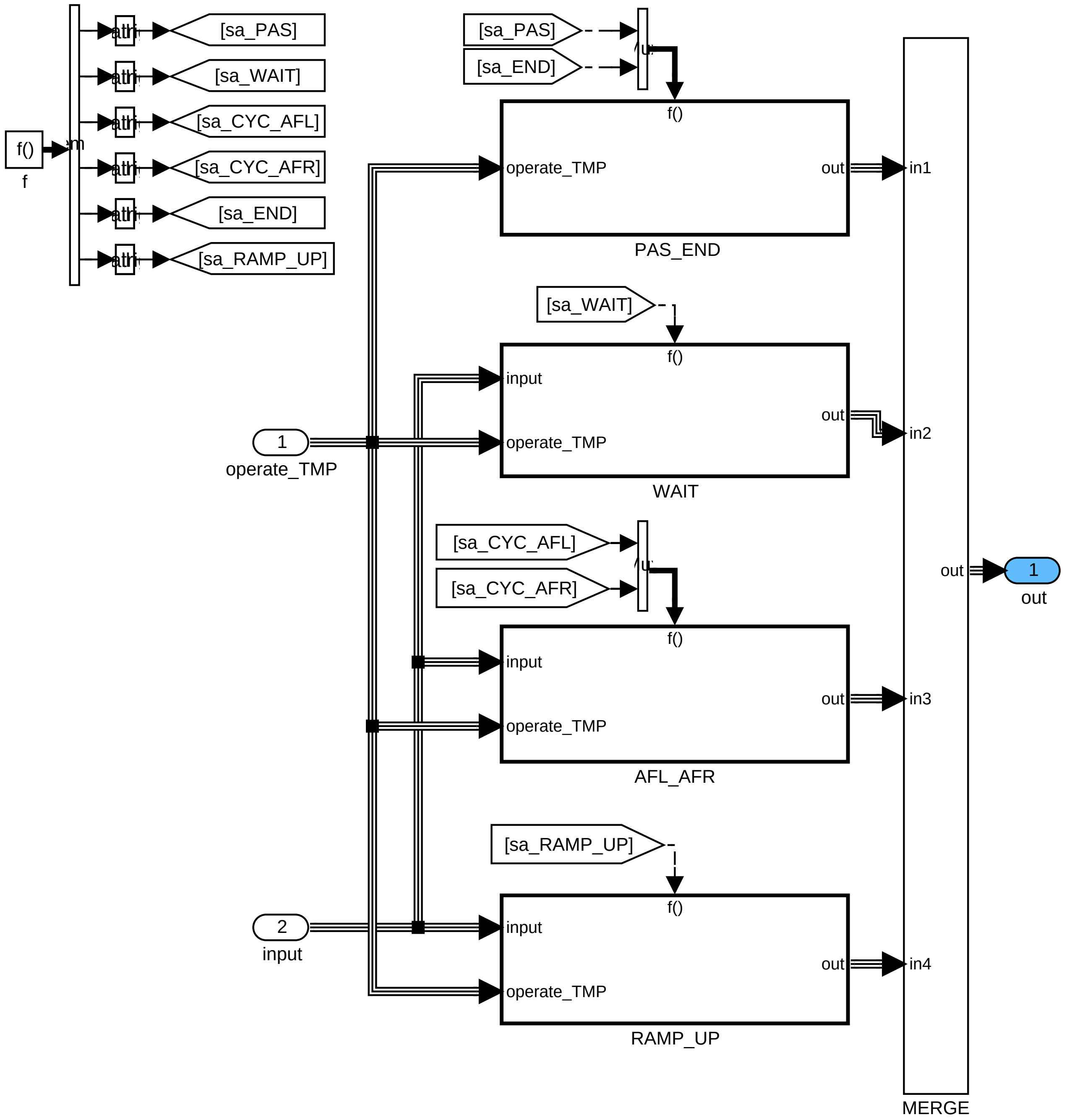


Figure 136: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG

18.13.1.1.1.3.1 State action PASSIVE/END:

No performance of any action in states PASSIVE and END.

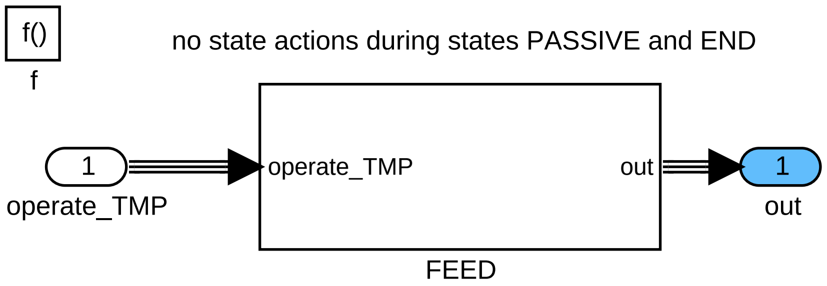


Figure 137: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG/PAS_END

18.13.1.1.1.3.2 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/SA_CAT_DIAG/WAIT

18.13.1.1.1.3.2.1 State action WAIT:

Converting the downstream sensor voltage to the related downstream lambda value.
Checking the flag for lean state and comparing with the allied old value. This action is checking for the start of a new monitoring cycle.
In case of a transition zero to one and validity of the cycle calculation of the downstream lambda mean value and resetting the downstream lambda sum and its counter afterwards. Calculating the new trim control setpoint the new trim control setpoint, the stat/dyn temperature difference and setting the validity bits.
In case of a transition zero to one and non validity of the cycle resetting the downstream lambda sum and its counter and marking the cycle as non valid.
Summing up downstream lambda and incrementing the counter in all other cases.

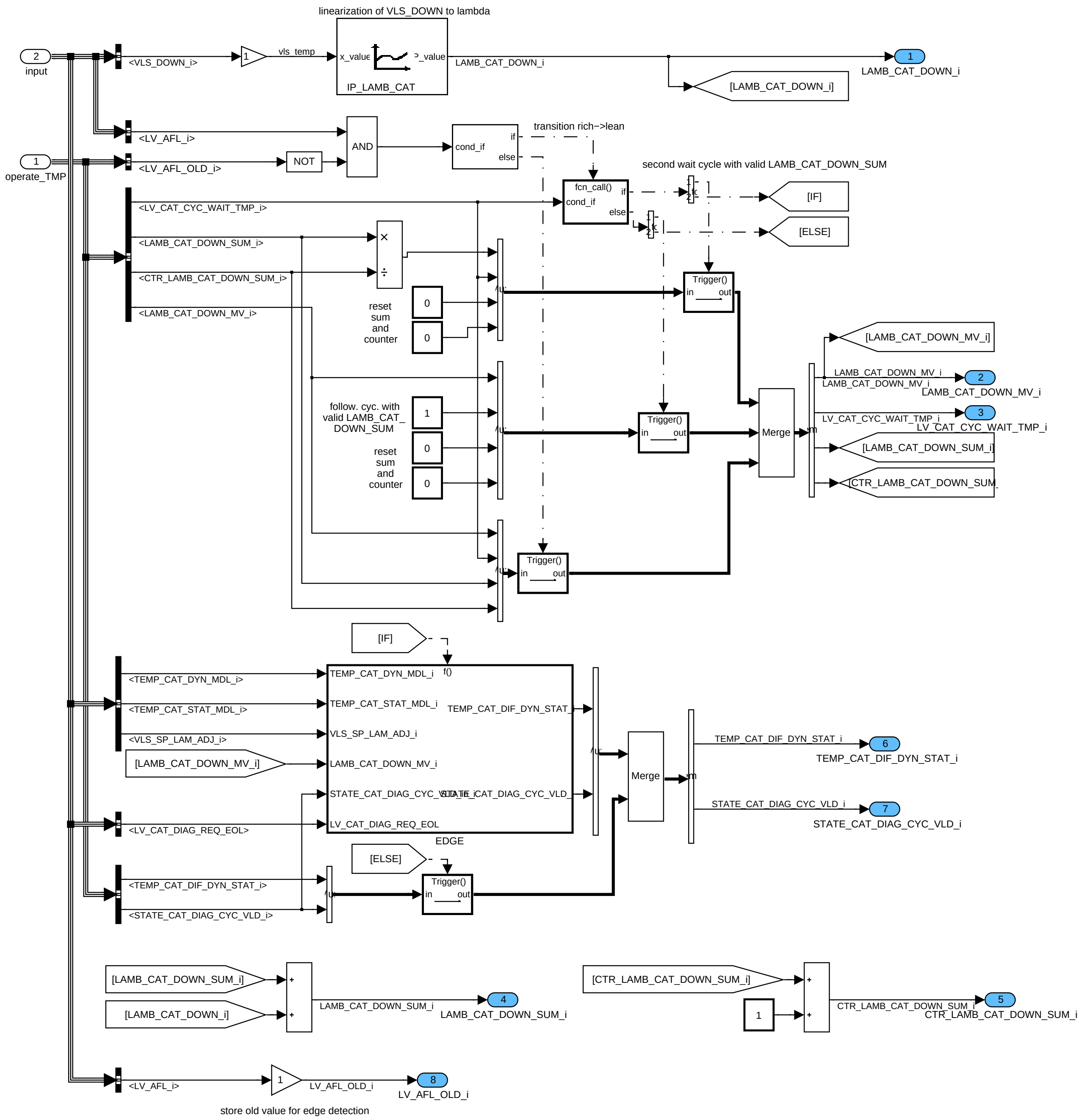


Figure 138: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG/WAIT/CLC

18.13.1.1.1.3.2.1.1 State action WAIT/edge action:

Computing the absolute difference between downstream lambda mean value and lambda setpoint and comparing it to the stored maximally admissible difference. Setting the RANGE bit in dependence on the result.
Comparing the stat/dyn temperature difference to the stored maximally allowed value and setting the TEMPERATURE bit accordingly.

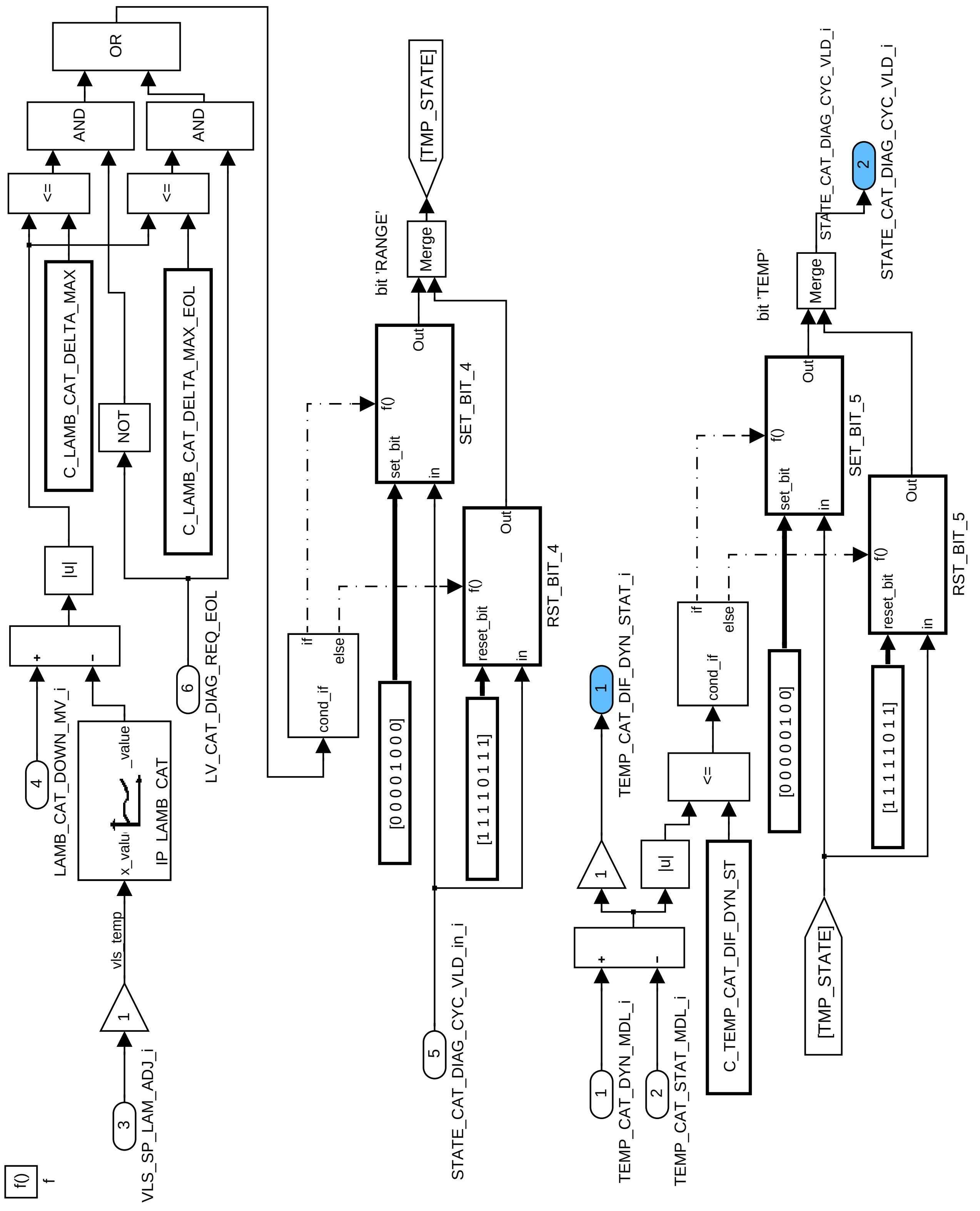


Figure 139: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG/WAIT/CLC/EDGE

18.13.1.1.1.3.3 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/SA_CAT_DIAG/AFL_AFR

18.13.1.1.1.3.3.1 State action AFL/AFR:

Converting the downstream sensor voltage to the related downstream lambda value.
Comparing the actual downstream lambda value with stored min/max values and overwriting in case of exceeding.
Summing up the actual downstream lambda value and incrementing the counter.
Calculation of the actual downstream oxygen mass flow and integrating the passed part of the cycle.
Setting of the summary validity bits.
Storing the old lean state flag for edge detection.

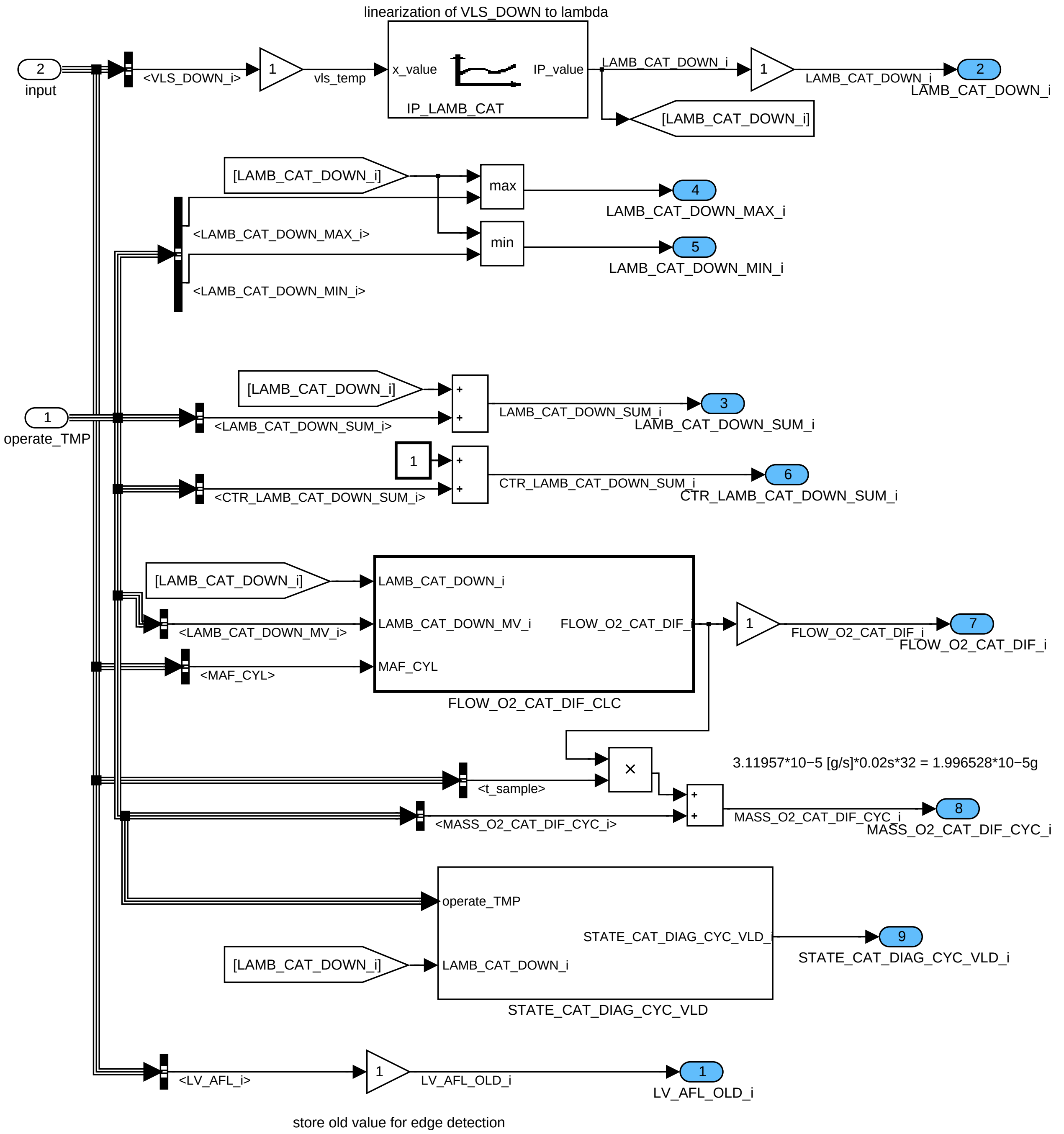
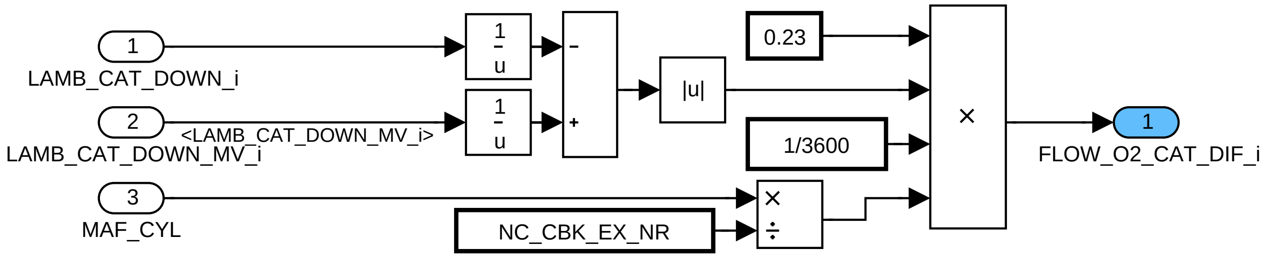


Figure 140: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG/AFL_AFR/CLC

18.13.1.1.1.3.3.1.1 Downstream oxygen mass flow:

Calculation of the actual downstream oxygen mass flow.



calculation 0.23*2/65536*0.03125*1000/3600*512=3.11957*10−5 [g/s]

Figure 141: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG/AFL_AFR/CLC/FLOW_O2_CAT_DIF_CLC

18.13.1.1.1.3.3.1.2 Check for cycle validity:

Resetting the 1CROSS bit. Checking for 1CROSS AND 2CROSS. Comparing the absolute difference between downstream lambda and lambda mean value to the related stored maximum and setting the MVDEL bit accordingly.

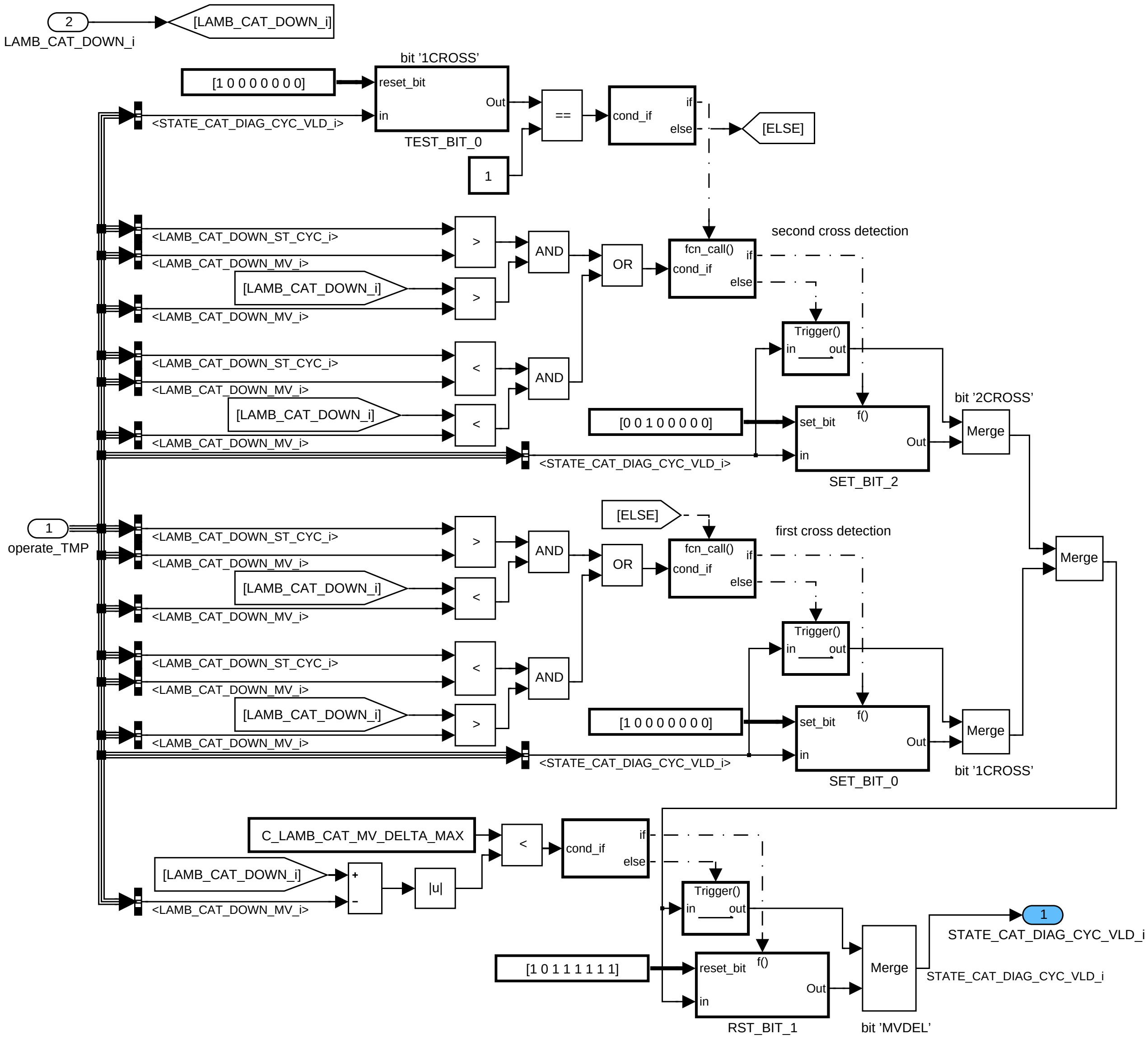


Figure 142: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG/AFL_AFR/CLC/STATE_CAT_DIAG_CYC_VLD

18.13.1.1.1.3.4 EGTR_FCTDGCEF0/OPERATE/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/SA_CAT_DIAG/RAMP_UP

18.13.1.1.1.3.4.1 State action RAMP_UP:

Converting the downstream sensor voltage to the related downstream lambda value.
Checking the flag for lean state and comparing with the allied old value (checking for the start of a new monitoring cycle).
In case of a transition zero to one calculation of the downstream lambda mean value, incrementing the counter of monitoring cycles showing lambda out of range, setting of the summary validity bits and summing up downstream lambda and its counter.

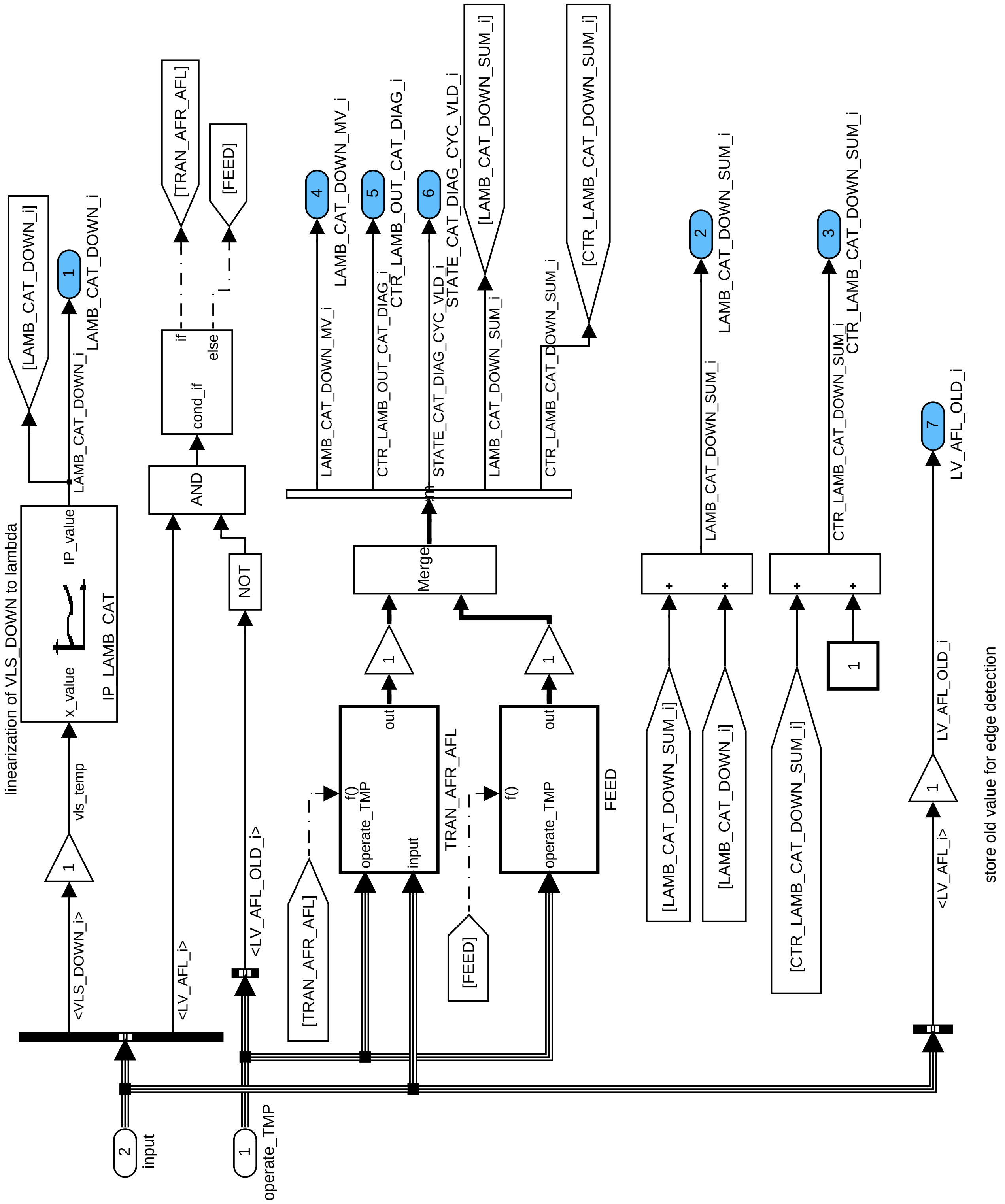


Figure 143: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG/RAMP_UP/CLC

18.13.1.1.1.3.4.1.1 TRAN_AFR_AFL:

Calculation of the downstream lambda mean value of last cycle and correcting the influence of asymmetric sensor characteristic.
Checking the actual sensor signal to be in the allowed range and setting the respective validity bit in accordance with as well as the counter of non valid monitoring cycles and the summary validity bits.
Resetting the downstream lambda sum and its counter to zero.

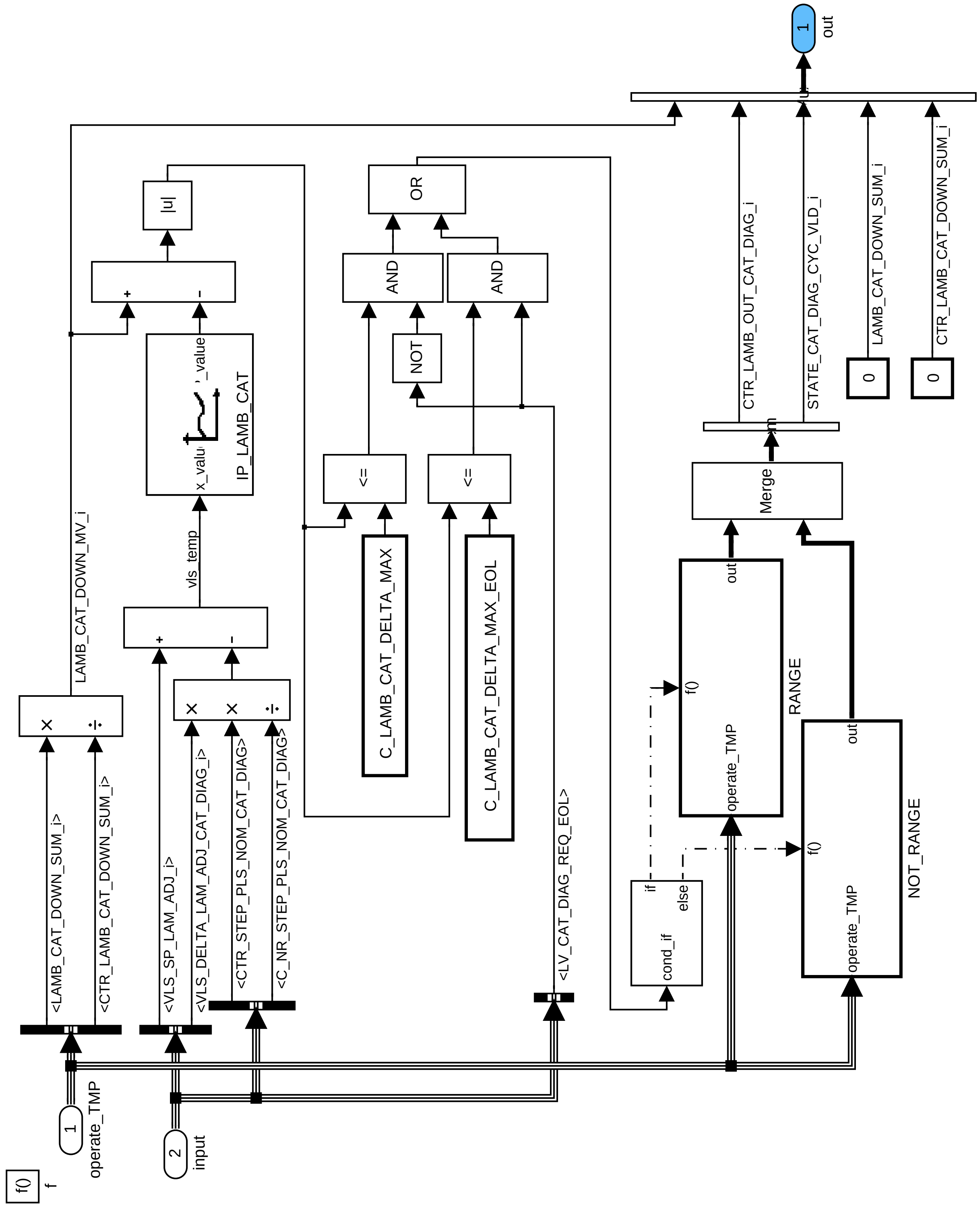


Figure 144: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/sa_CAT_DIAG/RAMP_UP/CLC/TRAN_AFR_AFL

18.13.1.1.1.3.4.1.1.1 TRAN_AFR_AFL/RANGE

Setting the counter of monitoring cycles showing lambda to be out of range to zero and the bit indicating that the diagnosis mean value is in the right range to one.

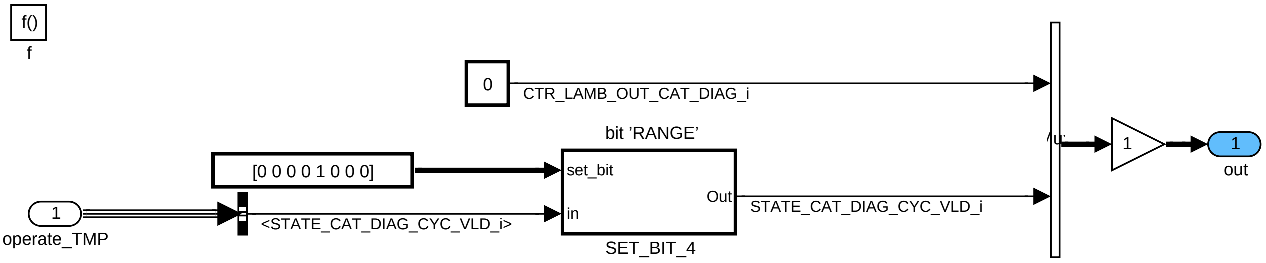


Figure 145: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ sa_CAT_DIAG/RAMP_UP/CLC/TRAN_AFR_AFL/RANGE

18.13.1.1.1.3.4.1.1.2 TRAN_AFR_AFL/NOT_RANGE

Adding one to the number of monitoring cycles showing lambda to be out of range and resetting the validity bit indicating that the diagnosis mean value is in the right range.

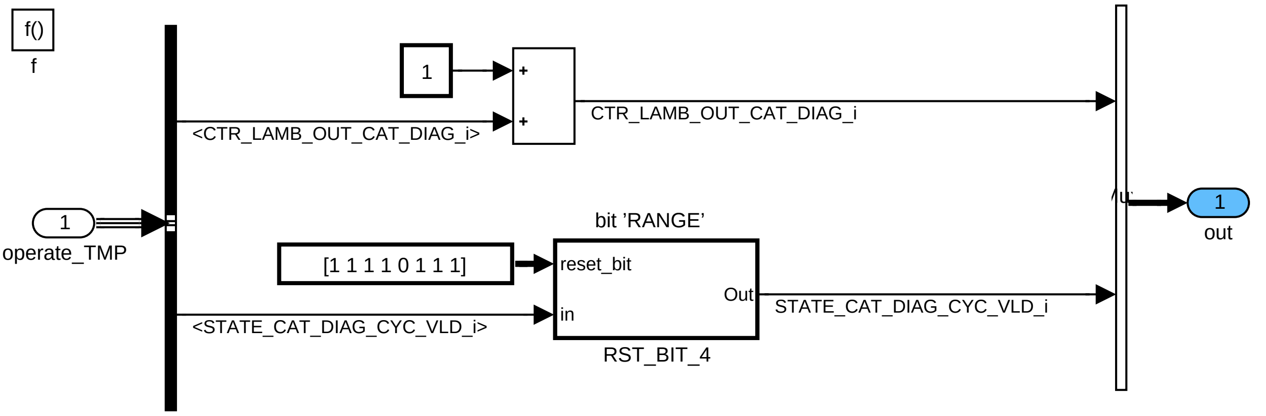


Figure 146: Path: EGTR_FCTDGCEF0/operate/EGTR_FCTDGCEF0_CBK_MNG/EGTR_FCTDGCEF0_CBK_SPC/ sa_CAT_DIAG/RAMP_UP/CLC/TRAN_AFR_AFL/NOT_RANGE

18.13.2 Handling data at non volatile memory:

Triggering of one of the following actions:
RNVMY (read non volatile memory): Reading the diagnosis value of last completed diagnosis cycle and the number of totally completed diagnosis cycles since ECU programming from non volatile memory.
UNVMY (update non volatile memory): Saving the diagnosis value of last completed diagnosis cycle and the number of totally completed diagnosis cycles since ECU programming to non volatile memory.
STB (standby): Initializing the diagnosis value and the number of totally completed diagnosis cycles since ECU programming to zero and saving to non volatile memory.

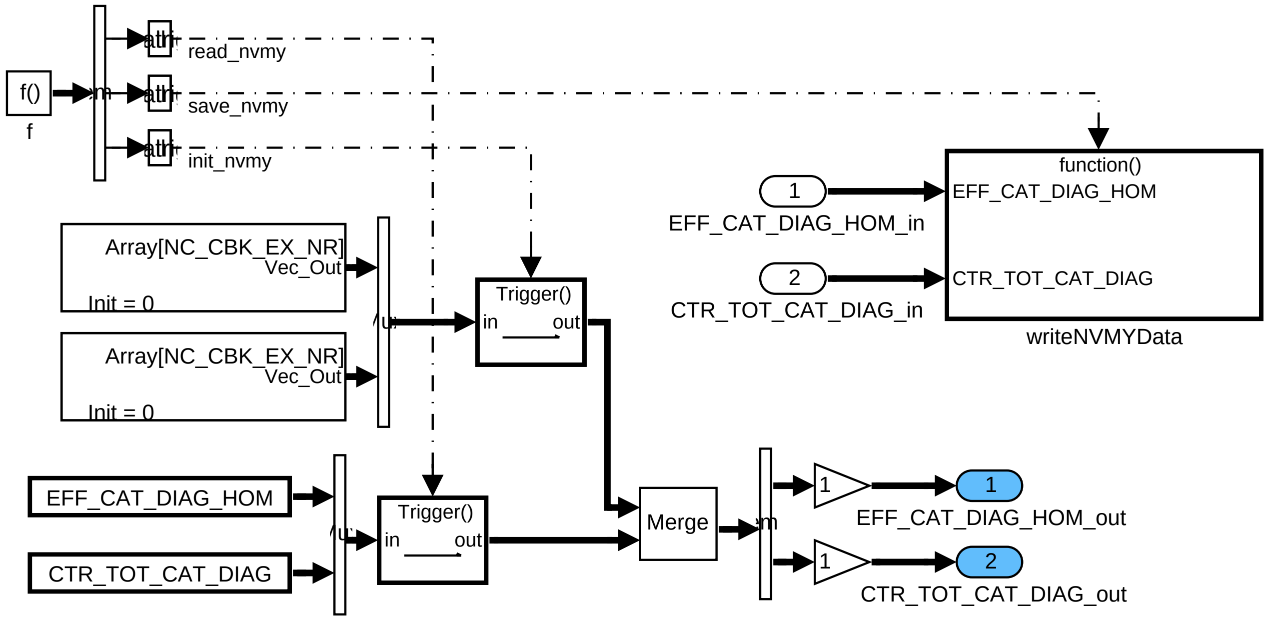


Figure 147: Path: EGTR_FCTDGCEF0/NVMYdata

18.13.2.1 Writing to non volatile memory (UNVMY):

Storing the final diagnosis value to non volatile memory and incrementing there the total number of completed diagnosis cycles since ECU programming by one.

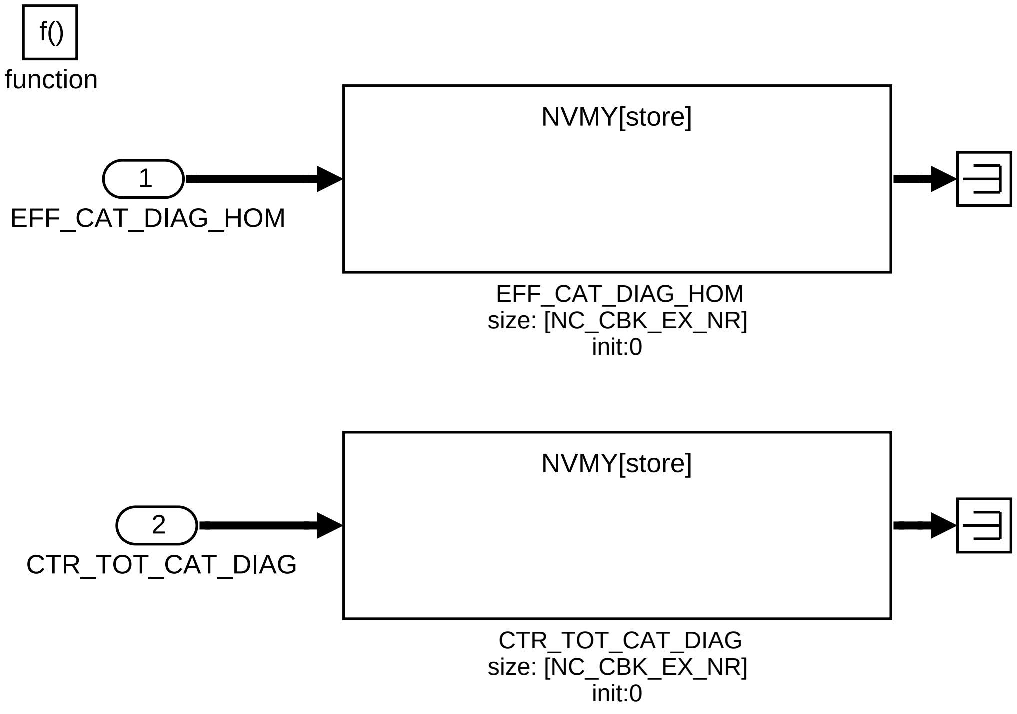


Figure 148: Path: EGTR_FCTDGCEF0/NVMYdata/writeNVMYData

18.13.3 RESET:

Initializing all listed variables to zero.

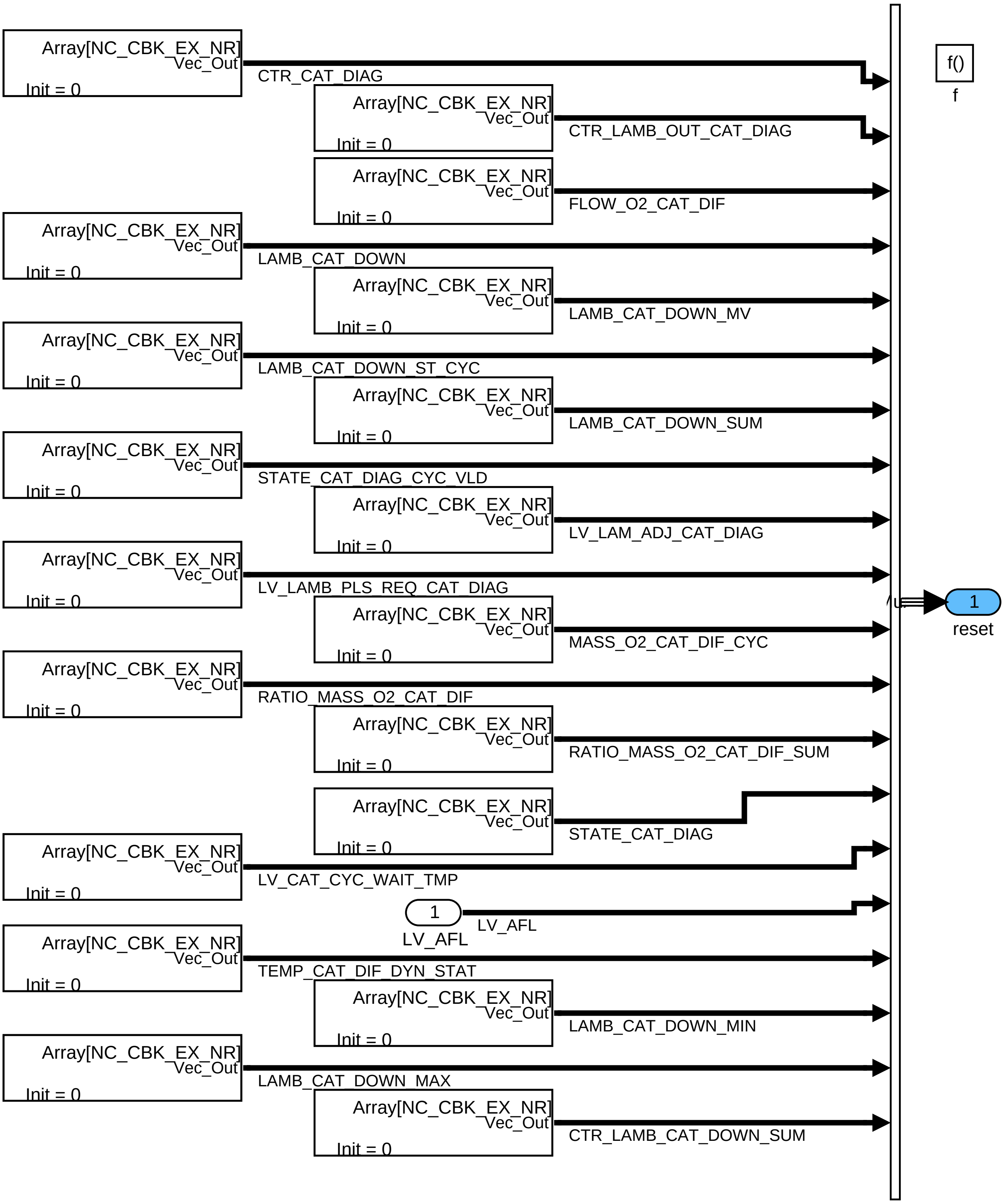


Figure 149: Path: EGTR_FCTDGCEF0/reset

18.14 Coordination of catalyst diagnosis results

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
EFF_CAT_DIAG [NC_CBK_EX_NR]	O/V/S	0... FFH	0... 1.9921875	7.8125e-3	[-]
At ended diagnosis final value for catalyst conversion capability					
ERR_SYM_CAT_DIAG_SUM [NC_CBK_EX_NR]	V	0 1 2 4 8	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	-	[-]
Detected error symptom for TWC summary of HOM and AFL diagnosis					
LV_CDN_DIAG_CAT_DIAG_SUM [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Diagnostic condition to start symptom detection (seto to one when condition is fulfilled)					
LV_END_DIAG_CAT_DIAG_SUM [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
TWC summary of HOM and AFL diagnosis performed at last one time					
LV_ERR_CAT_DIAG_SUM [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Present failure after filtering of the summary of HOM and AFL TWC diagnosis					

Input Data:

STATE_CAT_DIAG [NC_CBK_EX_NR]	LV_END_DIAG_CAT_DIAG_AFL [NC_CBK_EX_NR]	EFF_CAT_DIAG_HOM [NC_CBK_EX_NR]	O2L_CAT_DIAG_AFL [NC_CBK_EX_NR]
NC_CBK_EX_NR	ERR_SYM_CAT_DIAG [NC_CBK_EX_NR]	ERR_SYM_CAT_DIAG_AFL [NC_CBK_EX_NR]	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ABC_INC_CAT_DIAG_SUM	1	0... FFH	0... 255	1	[-]
Antibounce counter increment					
C_ABC_MAX_CAT_DIAG_SUM	1	1... FFH	1... 255	1	[-]
Maximum value for antibounce counter					
IP_EFF_CAT_DIAG_HOM	6	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_O2L_CAT_DIAG_AFL_IP_EFF_CAT	6	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Conversion of lean cat diag result O2L_CAT_DIAG_AFL to lambda=1 cat diag result EFF_CAT_DIAG					
LC_CAT_DIAG_AFL_ACT_SUM	1	0... 1H	0... 1	1	[-]
Activation of the AFL diagnosis for error calculation					
LC_CAT_DIAG_HOM_ACT_SUM	1	0... 1H	0... 1	1	[-]
Activation of the HOM diagnosis for error calculation					
LC_EFF_CAT_DIAG_HOM_ENA	1	0... 1H	0... 1	1	[-]
Use EFF_CAT_DIAG_HOM from linear cat diagnosis as result EFF_CAT_DIAG					
LC_O2L_CAT_DIAG_AFL_ENA	1	0... 1H	0... 1	1	[-]
Use O2L_CAT_DIAG_AFL from lean cat diagnosis as result EFF_CAT_DIAG					

General Information

This modul coordinates the results of both catalyst diagnosis methods
- lambda=1 catalyst diagnosis (HOM): EFF_CAT_DIAG_HOM[NC_CBK_EX_NR]
- lean catalyst diagnosis (AFL): O2L_CAT_DIAG_AFL[NC_CBK_EX_NR]

The common result is EFF_CAT_DIAG[NC_CBK_EX_NR]

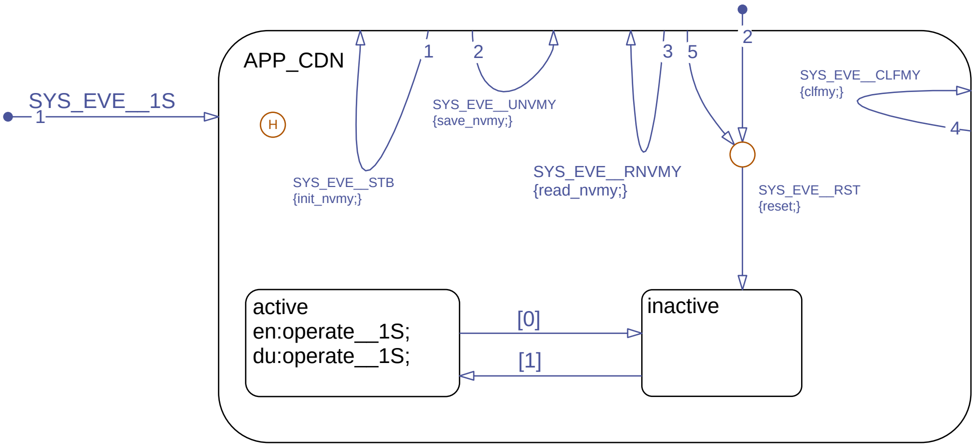
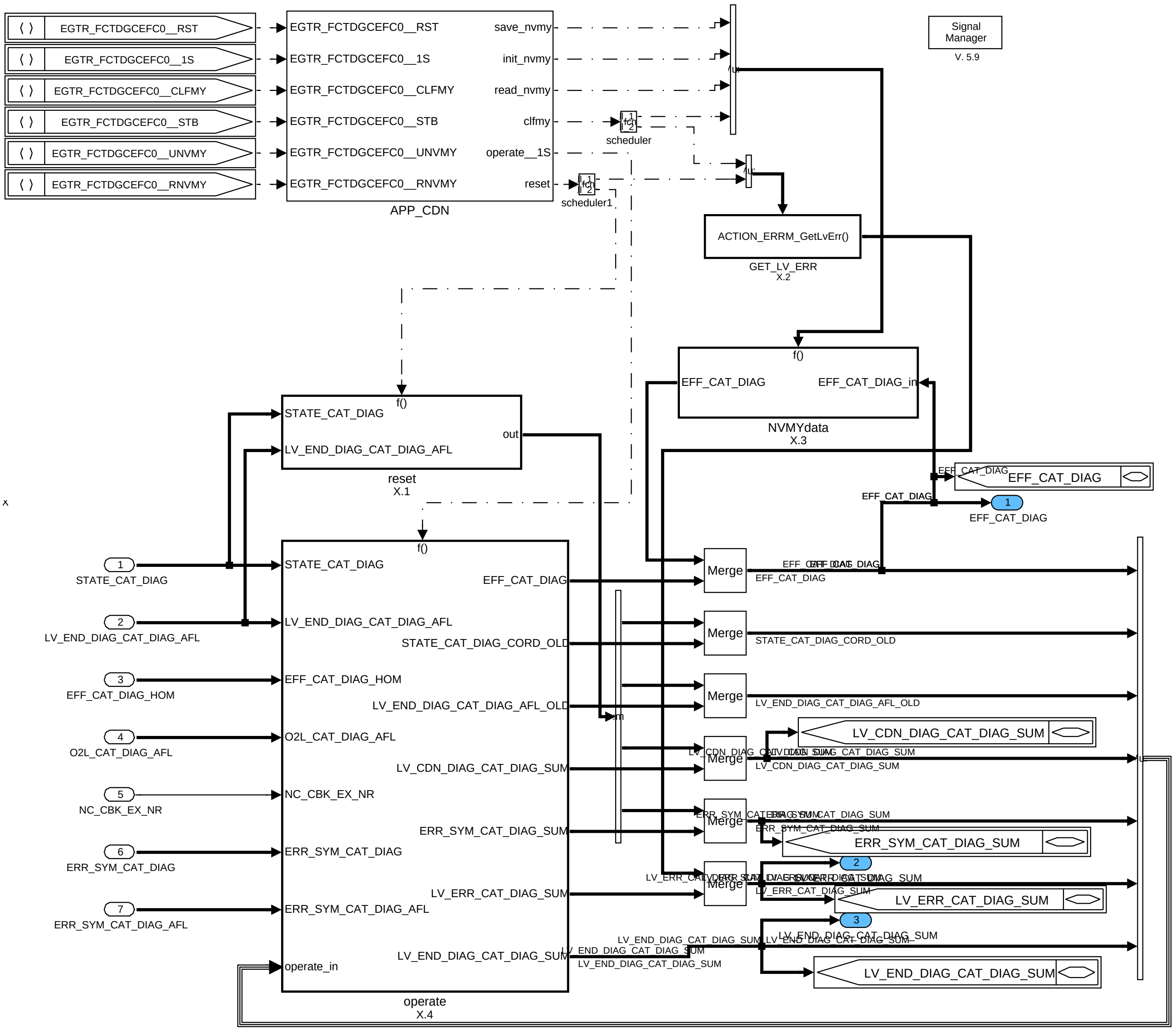


Figure 150:
Path: EGTR_FCTDGCEFC0/APP_CDN/Chart



SDA_SRS / SDA V 5.1 06−Apr−2006

Figure 151:
Path: EGTR_FCTDGCEFC0

18.14.1 Reset

The old values will be initialized with the regarding inputs, due to avoid an edge detection after reset.

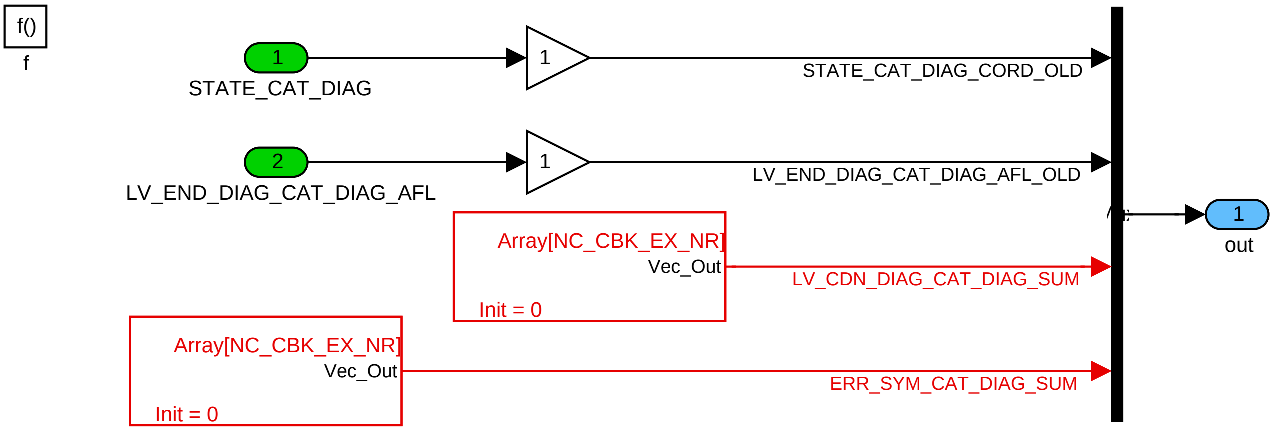


Figure 152:
Path: EGTR_FCTDGCEFC0/reset

18.14.2 Synchronization of the LV_ERR_CAT_DIAG_SUM

Synchronisation between LV_ERR_CAT_DIAG_SUM and its result from ERRM is done using action ACTION_ERRM_GetLvErr() at reset and clrfmy.

18.14.3 EFF_CAT_DIAG[NC_CBK_EX_NR] will be stored in the non-volatile memory

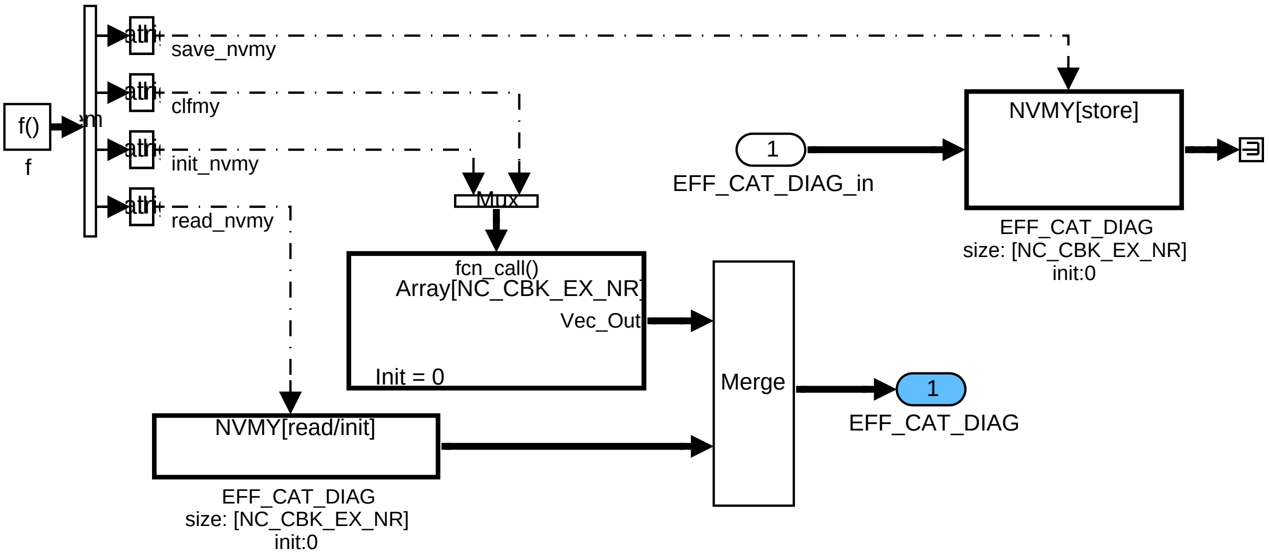


Figure 153:
Path: EGTR_FCTDGCEFC0/NVMYdata

18.14.4 FORMULA SECTION

18.14.4.1 Calculation of the summary diagnosis results

18.14.4.1.1 Calculation of the summary diagnosis "end" and "error symptom" outputs

The summary diagnosis is a combination of the TWC diagnosis of homogeneous and lean combustion modes. Last finished diagnosis saves the summary diagnosis result. When the ABC counter is calibrated to e.g. 2, then both diagnoses can influence the diagnosis result depending on which one is executed.

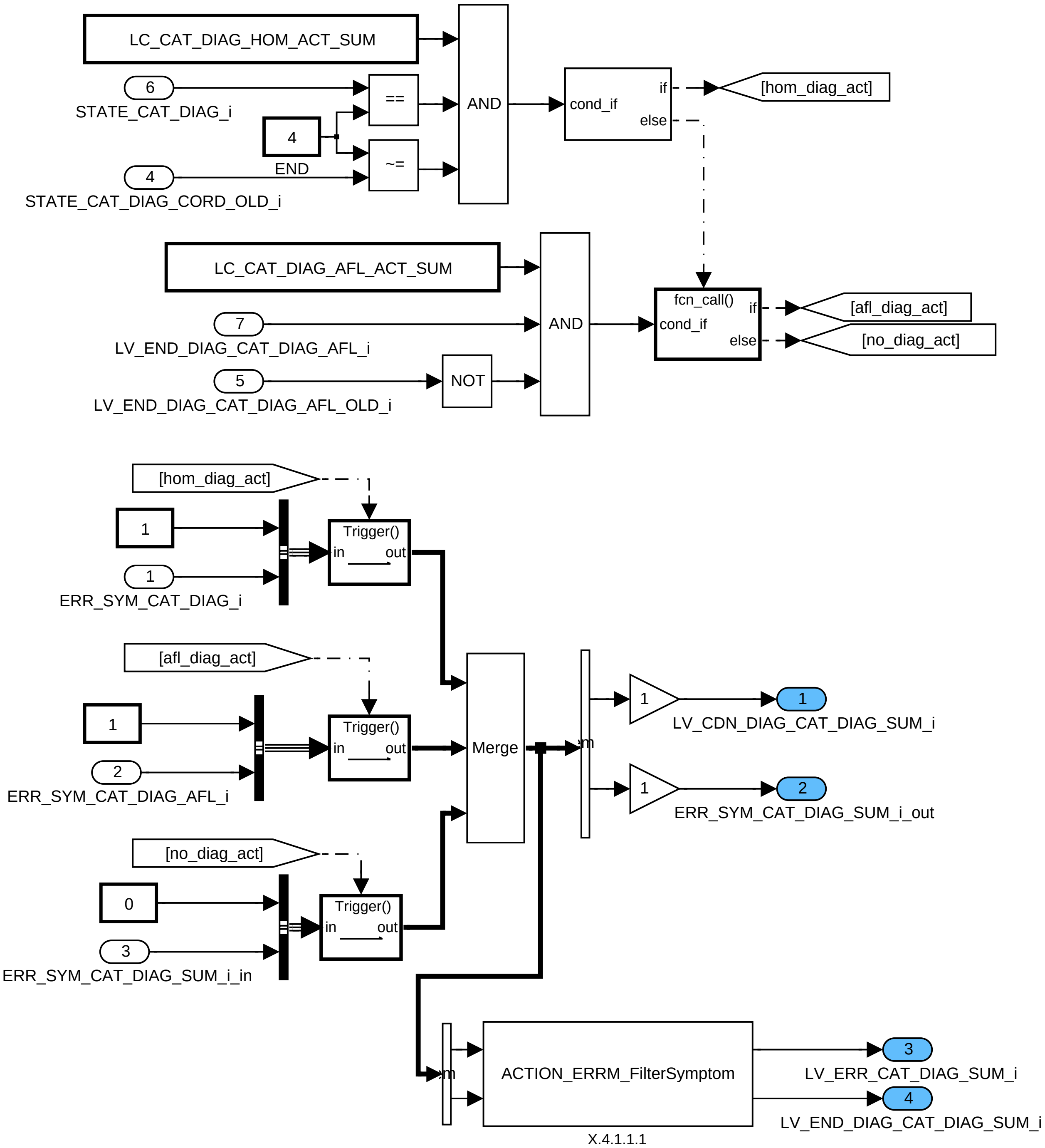


Figure 154:
Path: EGTR_FCTDGCEFC0/operate/CBK_MNG/CLC0

18.14.4.1.1.1 Call of the ERRM

Import actions:

ACTION_ERRM_FilterSymptom(IN<XX>, IN<lv_cdn_diag_XX>, IN<err_sym_XX >, IN<C_ABC_INC_XX >, IN<C_ABC_DEC_XX >, IN<C_ABC_MAX_XX >, OUT<LV_ERR_XX>)
The action computes the elementary anti-bounce filter for one failure treatment and returns filter results

Configuration for diagnostic symptoms:

Diagnostic CAT_DIAG_SUM[i]	Symptom description	Symptom	Filter type
Catalyst sum diagnosis	´CAT_EFFIC_LOW´	SYM_0	STD
	´CAT_DAMAGED’	SYM_1	
	Not used	SYM_2	
	Not used	SYM_3	

18.14.4.1.2 Calculation of the TWC efficiency

NC_CBK_EX_NR = 1 (one-bank system): calculation for index i = 1
NC_CBK_EX_NR = 2 (two-bank system): calculation for index i = 1 and 2

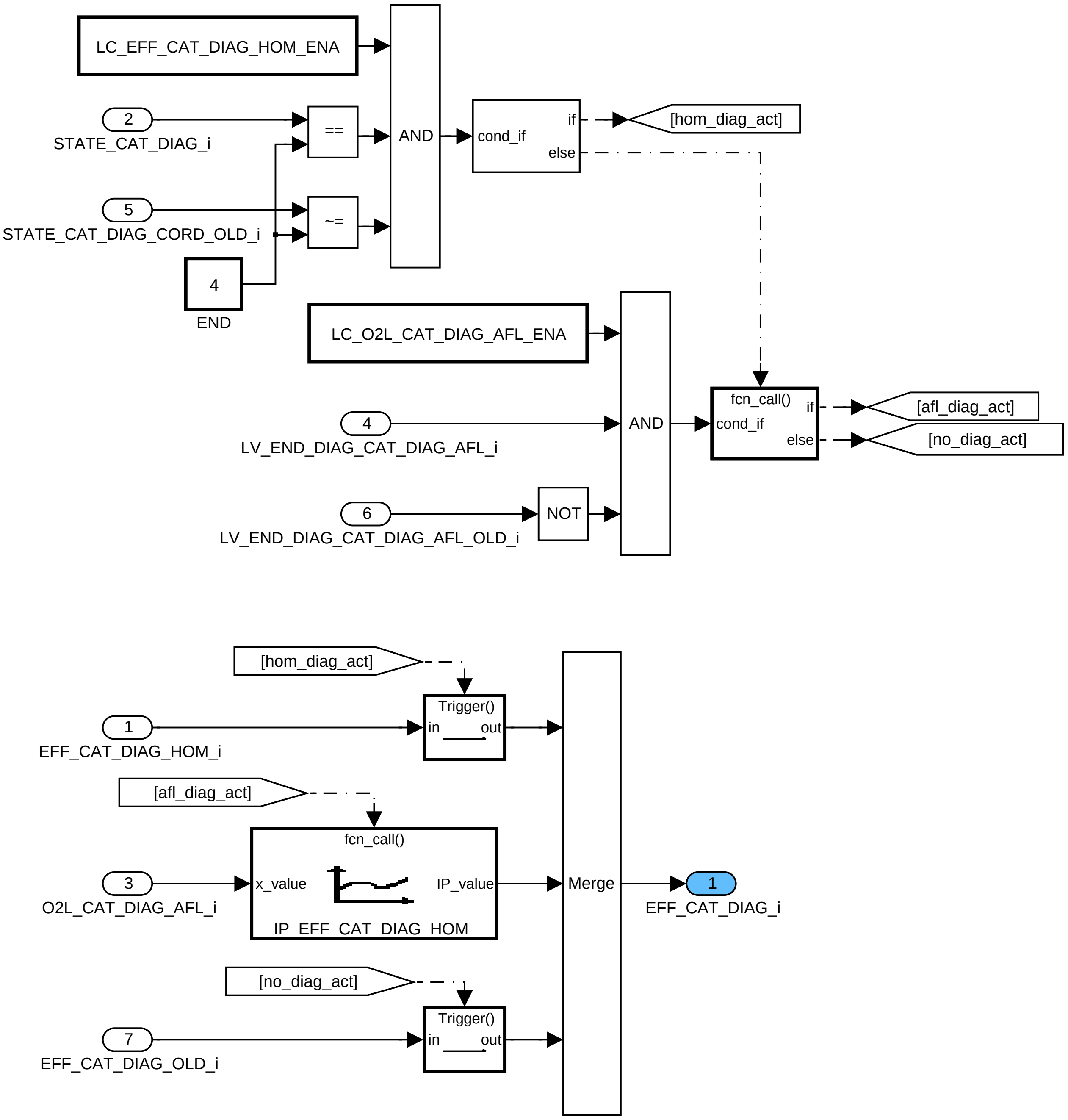


Figure 155:
Path: EGTR_FCTDGCEFC0/operate/CBK_MNG/CLC1

18.14.4.1.3 Calculation of the old values

The old values of the input signals will be stored for each bank.

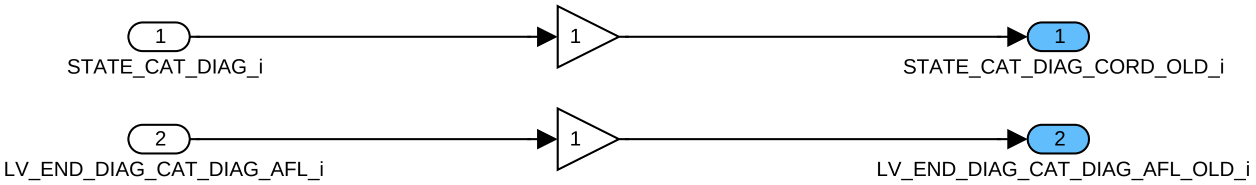


Figure 156:
Path: EGTR_FCTDGCEFC0/operate/CBK_MNG/CLC2

18.15 Catalyst efficiency diagnosis - OSC Method (Applic. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_CAT_DIAG[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag to inhibit catalyst diagnosis function					
LV_INH_DIAG_RBM_CAT_DIAG[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag to inhibit catalyst diagnosis function if there is a OBD failure					
STATE_RBM_CAT[NC_CBK_EX_NR]	O/V	0...FH	0...15	1	[-]
Interface of EFF_CAT_DIAG[i] monitor with the Rate-Based Monitoring statistics					
LOAD_CAT_LDC	O/V	0...3FFH	0...99.90234	0.0976563	[%]
Catalyst diagnosis specific load for limited dynamics calculation					

Input data:

LV_DC	LV_MTC_CUR_OFF	LV_ERR_MAF	FAC_TQ_REQ
LV_ERR_TPS	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_IVVT	LV_ERR_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_CPS	LV_ERR_DIAGCPS	LV_ERR_CRK	LV_ERR_TCO
LV_ERR_FSD[NC_CBK_EX_NR]	LV_MIS_STATE_A	LV_MIS_STATE_B	LV_ERR_LS_UP[NC_CBK_EX_NR]
LV_ERR_AMP	CTR_ERR_DYN_NR	LV_END_DIAG_CAT_DIAG[NC_CBK_EX_NR]	LV_ERR_MIS[NC_CYL_NR]
LV_ERR_AMP_PLAUS	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]	LV_CDN_DIAG_CAT_DIAG[NC_CBK_EX_NR]	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]
LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS
LV_ERR_VCV	LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]
LV_ERR_CAM_CUS	LV_ERR_SLV_IVVT_IN[NC_NR_CBK_IVVT]	LV_ERR_MAP_TPS_PLAUS	

18.15.1 Inhibition of diagnosis

FUNCTION DESCRIPTION:

General information:

The inhibition of the catalyst diagnosis can be deactivated by :

LC_INH_CAT_DIAG_MAN_DEAC = 1.

For LC_INH_CAT_DIAG_MAN_DEAC = 1 the catalyst diagosis function is not inhibited by a component failure.

Description:

If one of the following components fail the catalyst efficiency diagnosis function is stopped by LV_INH_DIAG_RBM_CAT_DIAG[i] = 1. The inhibition is calculated bank-wise.

Application conditions:

Initialization:	at reset all Variables shall be set to 0
Recurrence:	0.02 sec
Activation:	LV_DC transition 0 -> 1 (start driving cycle)
Deactivation:	LV_DC transition 1 -> 0 (driving cycle terminated)

Formula section:

Inhibition due to OBDI error

```
If        LV_ERR_MAF = 1
    or    LV_ERR_AMP = 1
    or    LV_ERR_AMP_PLAUS = 1
    or    LV_ERR_TPS = 1
    or    LV_ERR_LOAD_TPS_PLAUS = 1
    or    LV_ERR_IVVT = 1
    or    LV_ERR_LS_UP[i] = 1              // Global error flag for upstream sensor
    or    LV_ERR_LS_DOWN[i] = 1            // Global error flag for downstream sensor
    or    LV_ERR_DELTA_I_LAM[i] = 1
    or    LV_ERR_VLS_DOWN_DIF[i] = 1
    or    LV_ERR_CPS = 1
    or    LV_ERR_CRK = 1
    or    LV_ERR_TCO = 1
    or    LV_ERR_FSD[i] = 1
    or    LV_MIS_STATE_A = 1
    or    LV_MIS_STATE_B = 1
    or    LV_ERR_DIAGCPS = 1
```

```
    or      LV_ERR_FSD_LAM_LIM_x = 1
    or      LV_ERR_FUP = 1
    or      LV_ERR_FUP_MFP_PLAUS = 1
    or      LV_ERR_H_PRS_SYS = 1
    or      LV_ERR_VCV = 1
    or      LV_ERR_FUP_ORNG = 1
    or      LV_ERR_FUP_ST = 1
    or      LV_ERR_TTIP_MES_LSH_UP_x = 1
    or      LV_ERR_CAM_CUS = 1
    or      LV_ERR_SLV_IVVT_IN = 1
    or      LV_ERR_MAP_TPS_PLAUS = 1

Then        LV_INH_DIAG_RBM_CAT_DIAG[i] = 1
                    (Catalyst efficiency diagnosis inhibited due to OBD error )
Else        LV_INH_DIAG_RBM_CAT_DIAG[i] = 0
                    (Catalyst efficiency diagnosis not inhibited)
Endif
```

<u>Inhibition due to environmental condition / calibration:</u>

```
If          [ LV_MTC_CUR_OFF = 1
    Or      LV_INH_DIAG_RBM_CAT_DIAG[i] = 1]
    and     LC_INH_CAT_DIAG_MAN_DEAC = 0
Then        LV_INH_DIAG_CAT_DIAG[i] = 1
                    (Catalyst efficiency diagnosis inhibited)
Else        LV_INH_DIAG_CAT_DIAG[i] = 0
                    (Catalyst efficiency diagnosis not inhibited)
Endif
```

<u>Catalyst diagnosis specific load for limited dynamics calculation</u>

LOAD_CAT_LDC = FAC_TQ_REQ

The definition of both variables are different, therefore a linear interpolation between the physical limits (min and max) of FAC_TQ_REQ shall be applied, i.e LOAD_CAT_LDC = 0% if FAC_TQ_REQ = 0; LOAD_CAT_LDC = 99,9023438% if FAC_TQ_REQ = 1.999969.

Calibration data:

Name	Dim	Hex. limits	Phys. Limit	Resol.	Unit
LC_INH_CAT_DIAG_MAN_DEAC	1	0...1H	0...1	1	[-]
Manual deactivation of CAT diagnosis inhibition conditions					

18.15.2 Interface for Rate – Based - Monitoring

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the CAT_DIAG_i monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_CAT[i] data.

Within STATE_RBM_CAT[i], three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for catalyst diagnosis)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_CAT[i] = 0

on failure memory reset :

bit 1 of STATE_RBM_CAT[i] = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_FSD[i]	LV_ERR_AMP	LV_ERR_DIAGCPS	
LV_ERR_MAF			LV_ERR_AMP_PLAUS
LV_ERR_MIS[x]			
LV_ERR_DELTA_I_LAM[i]	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_SLV_IVVT_IN[i]	LV_ERR_SLV_IVVT_EX[i]
LV_ERR_MEC_IVVT_IN[i]	LV_ERR_MEC_IVVT_EX[i]	LV_ERR_LSH_DOWN[i]	LV_ERR_SCG_LS_DOWN[i]
LV_ERR_SCP_LS_DOWN[i]		LV_ERR_OC_LS_DOWN[i]	LV_ERR_CHK_LS_DOWN[i]
LV_ERR_DYN_VLD_LS_UP[i]	LV_ERR_FL_LS_DOWN[]i	LV_ERR_PUC_LS_DOWN[i]	LV_ERR_CPS
LV_ERR_REF_CRK_CAM_EX[i]	LV_ERR_REF_CRK_CAM_IN[i]	LV_ERR_PER_CAM_EX[i]	LV_ERR_PER_CAM_IN[i]
LV_ERR_SYN_CAM_EX[i]	LV_ERR_SYN_CAM_IN[i]	LV_ERR_SYN_CRK_CAM_IN[i]	LV_ERR_SYN_CRK_CAM_EX[i]
LV_ERR_PLAUS_CAM_EX[i]	LV_ERR_PLAUS_CAM_IN[i]	LV_ERR_CRK_PLAUS	LV_ERR_CRK_TOOTH
LV_ERR_CRK_TOOTH_PER	LV_ERR_CRK_SYN		LV_ERR_TCO_EL
LV_ERR_TCO_GRD	LV_ERR_TCO_STUCK	LV_ERR_TCO_PLAUS	LV_ERR_TCO_STUCK_RNG
LV_ERR_OBD_LSH_DOWN[i]	LV_ERR_SWT_LS_DOWN[i]	LV_ERR_PUE_LS_DOWN[i]	LV_ERR_VLS_DOWN_DIF[i]
LV_ERR_TPS_1	LV_ERR_TPS_2	LV_ERR_TPS_RATIO	LV_ERR_TPS_AD
LV_ERR_TPS_AD_BOL	LV_ERR_TPS_MAF_1	LV_ERR_TPS_MAF_2	LV_ERR_TPS_ST_CHK_2
LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3	LV_ERR_MTC_DR
LV_ERR_FSD_LAM_LIM_x	LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS
LV_ERR_VCV	LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP_x
LV_ERR_TOOTH_OFF_EX[i]	LV_ERR_TOOTH_OFF_IN[i]	LV_ERR_SLV_IVVT_IN	LV_ERR_MAP_TPS_PLAUS

```
If(1)       { CPU optimization at LV_DC 0 → 1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While  bit 1 of STATE_RBM_CAT[i] = 0 do
                   with each XX failure of the above list :
                   ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                   SYNCHRONIZATION<CALL>)
                   If(2)          XX has a pending status
                   Then(2)
                                  bit 1 of STATE_RBM_CAT[i] = 1
                   Endif(2)
            Endwhile
```

```
  Else(1)
              { the dynamic failure memory is empty }
              No action
  Endif(1)
Every 1 s :
If    bit 0 of STATE_RBM_XX = 0
Then
              If      LV_END_DIAG_CAT_DIAG[i] = 1
              Then    bit 0 of STATE_RBM_CAT[i] = 1
              Endif
Endif

If    bit 1 of STATE_RBM_CAT[i] = 0
Then
              If      LV_INH_DIAG_RBM_CAT_DIAG[i] = 1
              Then    bit 1 of STATE_RBM_CAT[i] = 1
              Endif
Endif

bit 2 of STATE_RBM_XX = 1
```

18.16 Catalyst efficiency diagnosis error interface for a 2 bank system

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
EFF_CAT_DIAG_OBD [NC_CBK_EX_NR]	O/V/S	0... FFH	0... 1.9921875	7.8125e-3	[-]
Cat. Diag. value for OBD Scantool Mod 6 output					
EFF_CAT_DIAG_SUM	V	0... FFH	0... 1.9921875	7.8125e-3	[-]
Common value for catalyst conversion capability of both cylinder banks (OBD emission limit)					
EFF_CAT_DIAG_TMP [NC_CBK_EX_NR]	-	0... FFH	0... 1.9921875	7.8125e-3	[-]
temporary Cat. Diag. value for OBD Scantool Mod 6 output					
EFF_CAT_MAX_DIAG_OBD [NC_CBK_EX_NR]	O/V/S	0... FFH	0... 1.9921875	7.8125e-3	[-]
Actually threshold for Scantool Mod 6 Output					
EFF_CAT_MAX_DIAG_TMP [NC_CBK_EX_NR]	-	0... FFH	0... 1.9921875	7.8125e-3	[-]
temporary actually threshold for Scantool Mod 6 Output					
ERR_SYM_CAT_DIAG [NC_CBK_EX_NR]	V	0 1 2 4 8	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	-	[-]
Detected failure of each symptom: failure without filtering of diagnosis value catalyst diagnose					
LV_CDN_DIAG_CAT_DIAG [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag condition for catalyst diagnosis active					
LV_END_DIAG_CAT_DIAG [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag catalyst diagnosis finished for that driving cycle					
LV_ERR_CAT_DIAG [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Present failure catalyst efficiency					
STATE_CAT_DIAG_OLD [NC_CBK_EX_NR]	-	0 1 2 3 4 5	PASSIVE WAIT CYC_LEAN CYC_RICH END RAMP_UP	-	[-]
old value of State of diagnosis for edge detection					

Input Data:

STATE_CAT_DIAG [NC_CBK_EX_NR]	EFF_CAT_DIAG_HOM [NC_CBK_EX_NR]	LV_DC	LV_END_DIAG_LAM_ADJ [NC_CBK_EX_NR]
LV_ERR_LAM_ADJ [NC_CBK_EX_NR]	LV_ERR_LS_DOWN [NC_CBK_EX_NR]	LV_ERR_LS_UP [NC_CBK_EX_NR]	LV_LS_DOWN_DIAG_END [NC_CBK_EX_NR]
LV_LS_UP_DIAG_END [NC_CBK_EX_NR]	LV_CAT_DIAG_REQ_EOL	NC_CBK_EX_NR	LV_END_DIAG_CAT_DIAG_AFL [NC_CBK_EX_NR]
ERR_SYM_CAT_DIAG_AFL [NC_CBK_EX_NR]	O2L_CAT_DIAG_AFL [NC_CBK_EX_NR]		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_EFF_CAT_MAX_50_DIAG [NC_CBK_EX_NR]	1	0... FFH	0... 1.9921875	7.8125e-3	[-]
max. threshold for EFF_CAT_DIAG to detect a separate catalyst with efficiency lower then 50%					
C_EFF_CAT_MAX_DIAG [NC_CBK_EX_NR]	1	0... FFH	0... 1.9921875	7.8125e-3	[-]
max. threshold for EFF_CAT_DIAG to detect a separate catalyst with OBD limit emissions					
C_EFF_CAT_MAX_DIAG_EOL [NC_CBK_EX_NR]	1	0... FFH	0... 1.9921875	7.8125e-3	[-]
max. threshold for EFF_CAT_DIAG[i] to detect a separate catalyst with OBD limit emissions at EOL Test					

C_EFF_CAT_MAX_SUM_DIAG	1	0... FFH	0... 1.9921875	7.8125e-3	[-]
max. threshold for EFF_CAT_DIAG to common detect of catalysts with vehicle OBD limit emissions					
C_O2L_CAT_DIAG_AFL_THD_ERR_CORD [NC_CBK_EX_NR]	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Error interface module local O2L threshold indicating bad catalyst from lean TWC diagnosis					
IP_EFF_CAT_DIAG_CLC_DIAG_AFL	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_O2L_IP_EFF_CLC_DIAG_AFL	8	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Correlation factor between calculated O2 loading and catalyst efficiency value					
IP_EFF_CAT_DIAG_SUM	6*6	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_EFF_CAT_DIAG_1_IP_EFF_CAT	6	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_EFF_CAT_DIAG_2_IP_EFF_CAT	6	0... FFH	0... 1.9921875	7.8125e-3	[-]
Map of common EFF_CAT_DIAG_SUM value depending from separate bank values for common OBD emiss. limit					
LC_CAT_DIAG_AFL_CHK	1	0... 1H	0... 1	1	[-]
Maual activation AFL diagnosis for set of readiness flag					
LC_CAT_DIAG_DIS	1	0... 1H	0... 1	1	[-]
For report value to Scan tool: separate EFF_CAT_DIAG for each catalyst or common, if over OBD emiss. limit					
LC_CAT_DIAG_RESU	1	0... 1H	0... 1	1	[-]
Switch over the failure memory setting: separate for each catalyst or common for OBD emiss. limit					

General Information

For 2 exhaust gas bank system 1 bank (catalyst) could be aged over OBD limit, but if other catalyst is O.K, emissions of whole vehicle need not be over OBD limit, so MIL needn t go on. A separate diagnosis of every cylinder bank(catalyst), or common diagnosis is switchable. A common EFF_CAT_DIAG_SUM is calculated from separate EFF_CAT_DIAG[i] values of separate banks and error memory entered if common value is over the threshold. For 50% catalyst conversion a higher threshold will be observed.

Configuration for diagnostic symptoms:

Diagnostic CAT_DIAG[i]	Symptom description	Symptom	Filter type
Catalyst diagnosis	´CAT_EFFIC_LOW´	SYM_0	NO
	´CAT_DAMAGED’	SYM_1	NO

NC_CBK_EX_NR defines the number of exhaust banks.
For vector elements, the variable extension _i is used in the model instead of [i] as found in the textual description.

Application Conditions

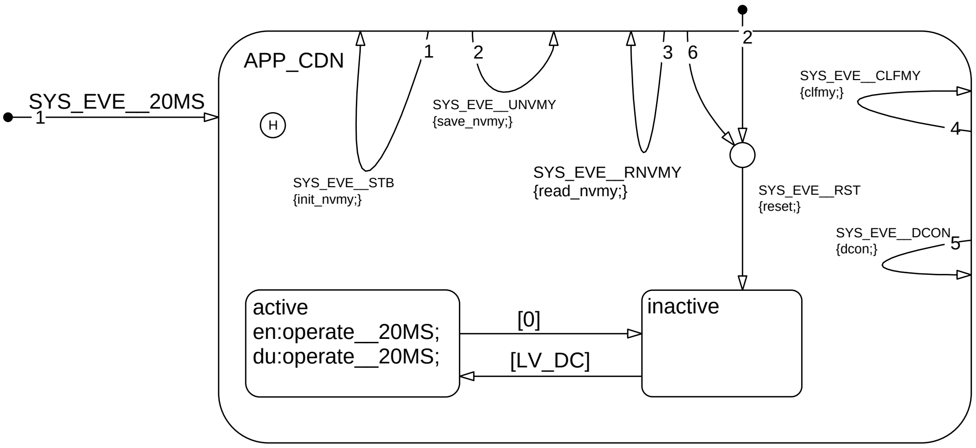


Figure 157:

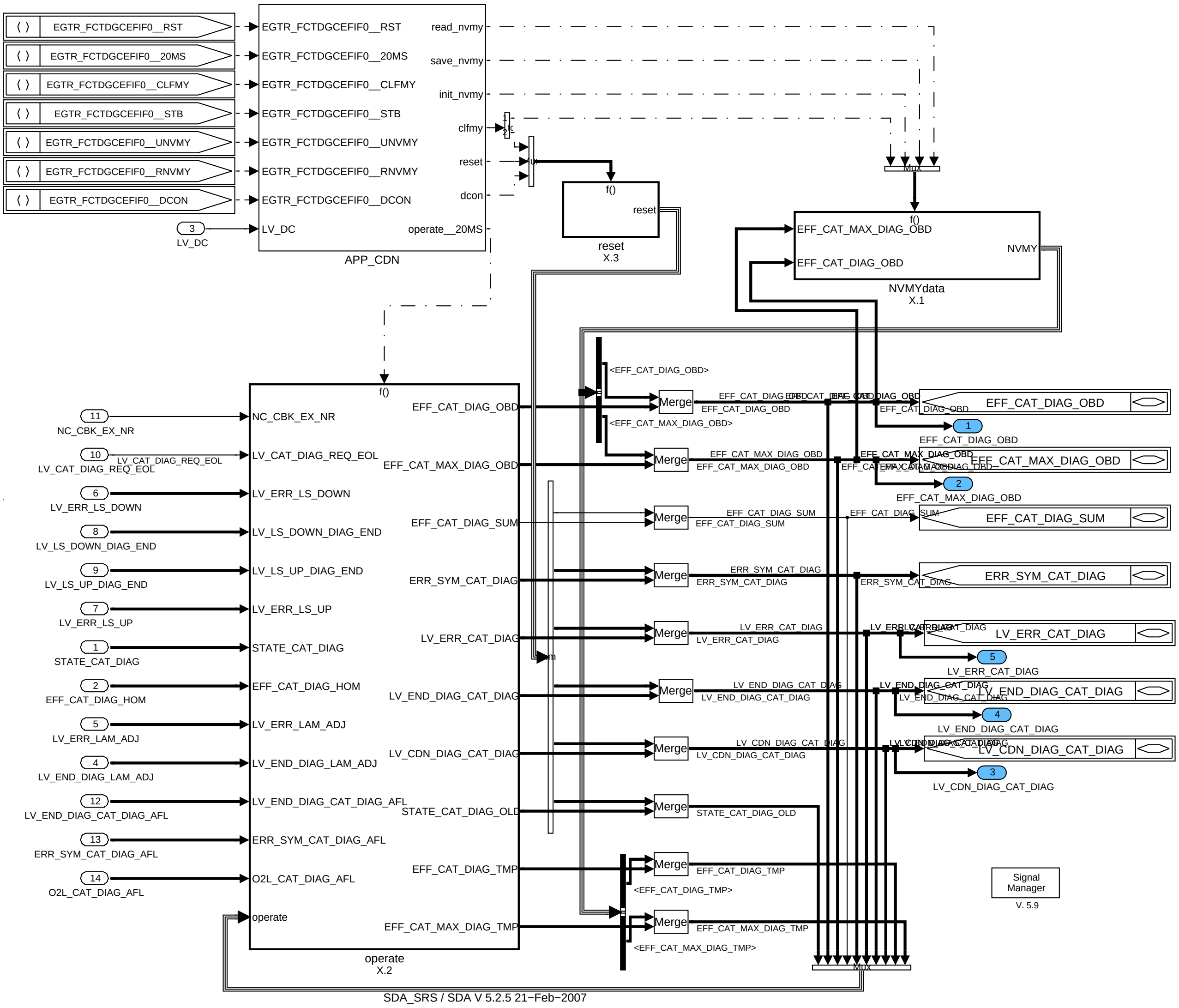

EGTR_FCTDGCEFIF0__RST
EGTR_FCTDGCEFIF0__20MS
EGTR_FCTDGCEFIF0__CLFMY
EGTR_FCTDGCEFIF0__STB
EGTR_FCTDGCEFIF0__UNVMY
EGTR_FCTDGCEFIF0__RNVMY
EGTR_FCTDGCEFIF0__DCON
3
LV_DC
read_nvmy
save_nvmy
init_nvmy
clfmy
reset
dcon
operate__20MS
APP_CDN
f()
reset
reset
X.3
Mux
f()
EFF_CAT_MAX_DIAG_OBD
EFF_CAT_DIAG_OBD
NVMY
NVMYdata
X.1
<EFF_CAT_DIAG_OBD>
<EFF_CAT_MAX_DIAG_OBD>
f()
11
NC_CBK_EX_NR
10
LV_CAT_DIAG_REQ_EOL
6
LV_ERR_LS_DOWN
8
LV_LS_DOWN_DIAG_END
9
LV_LS_UP_DIAG_END
7
LV_ERR_LS_UP
1
STATE_CAT_DIAG
2
EFF_CAT_DIAG_HOM
5
LV_ERR_LAM_ADJ
4
LV_END_DIAG_LAM_ADJ
12
LV_END_DIAG_CAT_DIAG_AFL
13
ERR_SYM_CAT_DIAG_AFL
14
O2L_CAT_DIAG_AFL
operate
EFF_CAT_DIAG_OBD
EFF_CAT_MAX_DIAG_OBD
EFF_CAT_DIAG_SUM
ERR_SYM_CAT_DIAG
LV_ERR_CAT_DIAG
LV_END_DIAG_CAT_DIAG
LV_CDN_DIAG_CAT_DIAG
STATE_CAT_DIAG_OLD
EFF_CAT_DIAG_TMP
EFF_CAT_MAX_DIAG_TMP
operate
X.2
Merge
<EFF_CAT_DIAG_TMP>
<EFF_CAT_MAX_DIAG_TMP>
1
2
3
4
5
Signal Manager
V. 5.9
SDA_SRS / SDA V 5.2.5 21−Feb−2007

Figure 158:

18.16.1 No title given

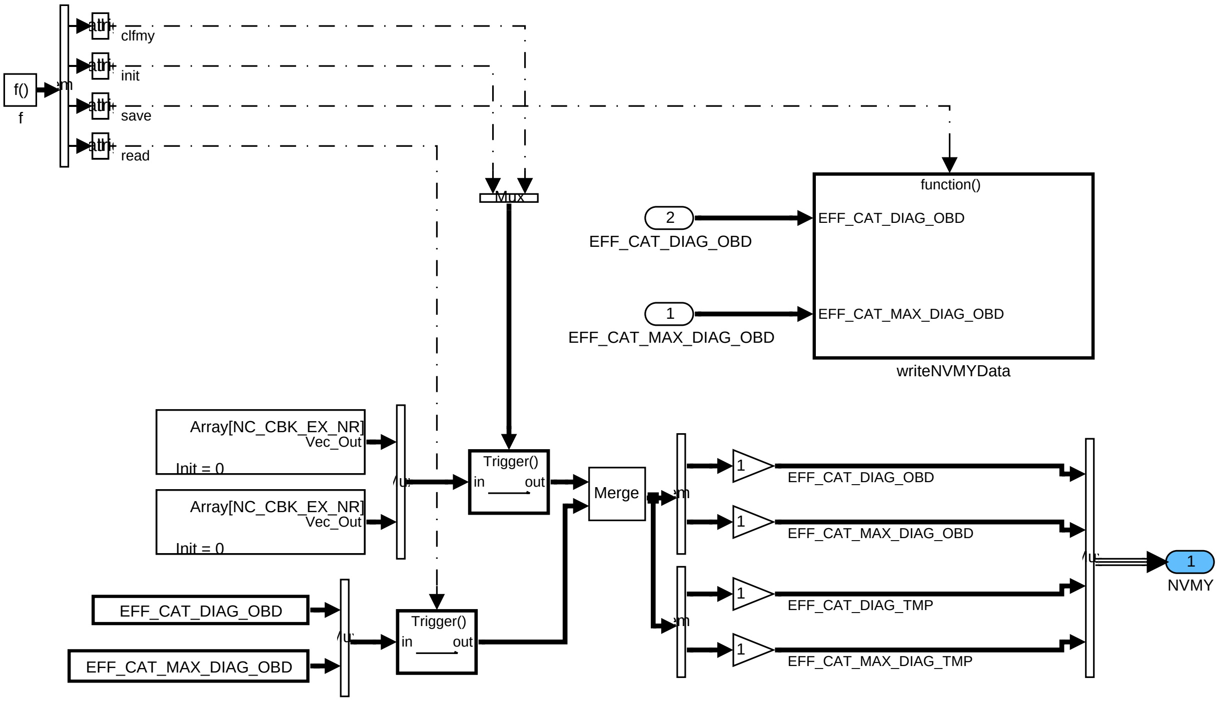


Figure 159:

18.16.2 Formula section

18.16.2.1 Introduction multiplebank system

18.16.2.1.1 Check end conditions of hom and lean catalyst diagnosis

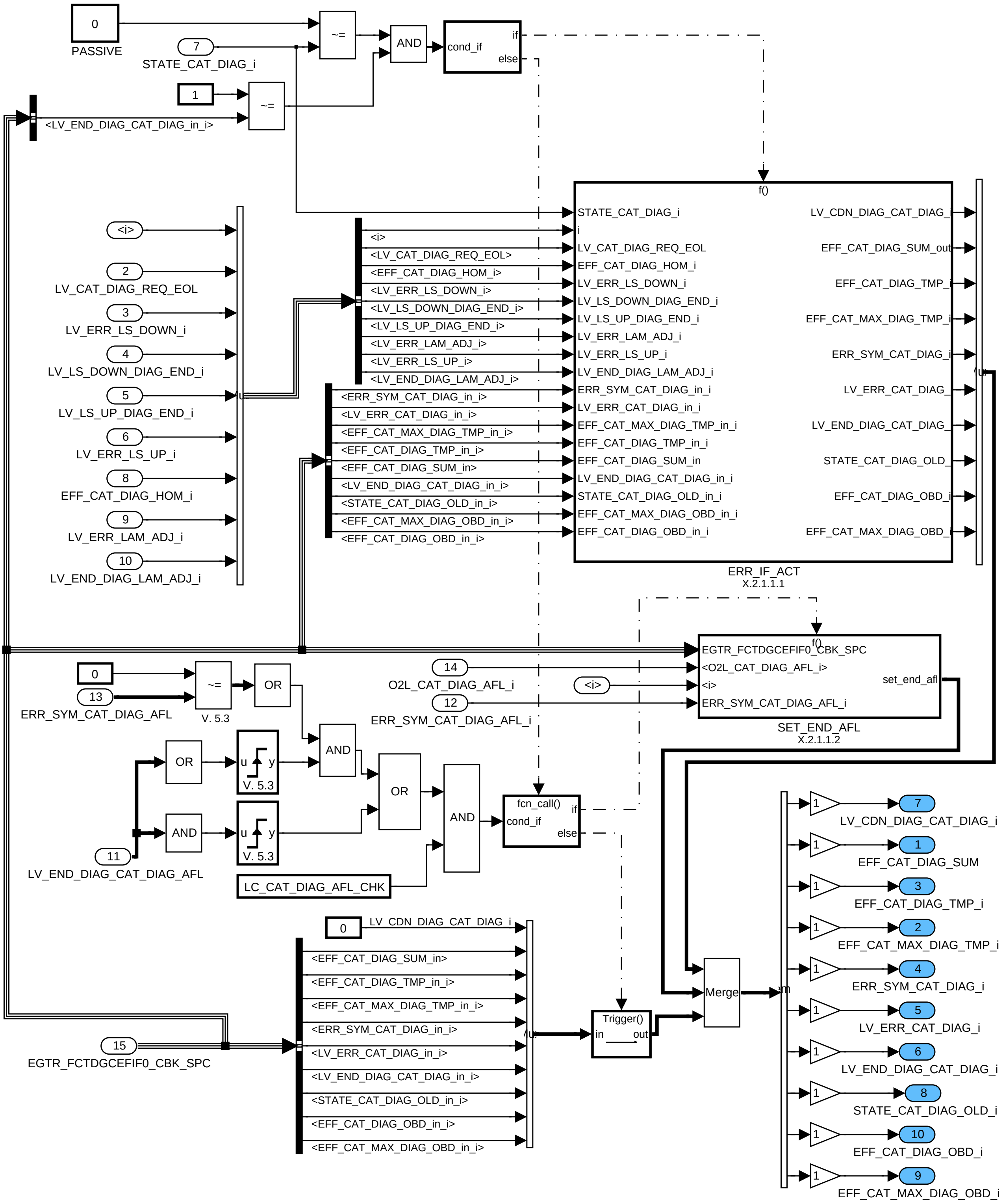


Figure 160:

18.16.2.1.1.1 Check end condition of the homogeneus catalys diagnosis

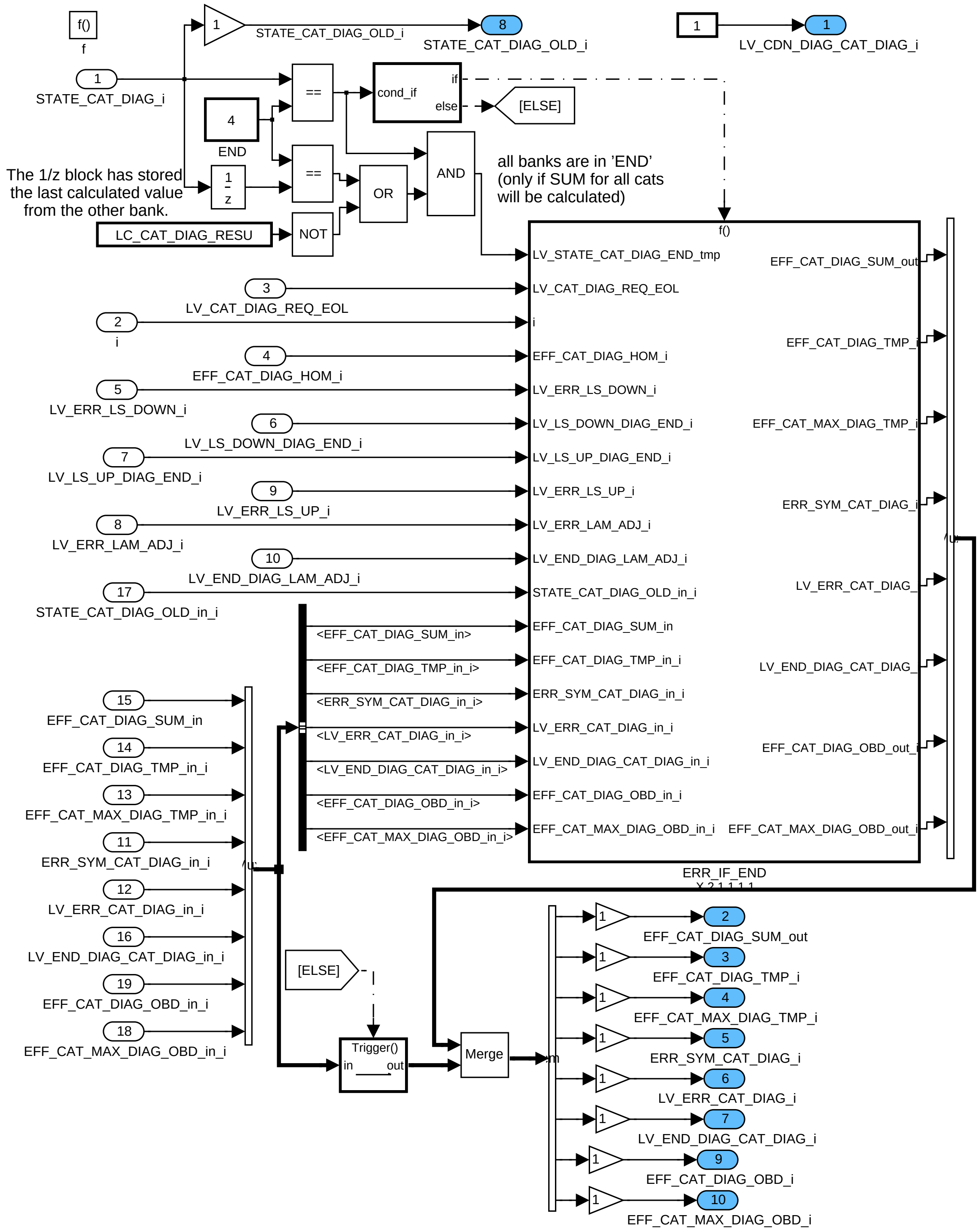


Figure 161:

18.16.2.1.1.1.1 Import actions:

ACTION_ERRM_StorePrevFrf (IN <XX>)

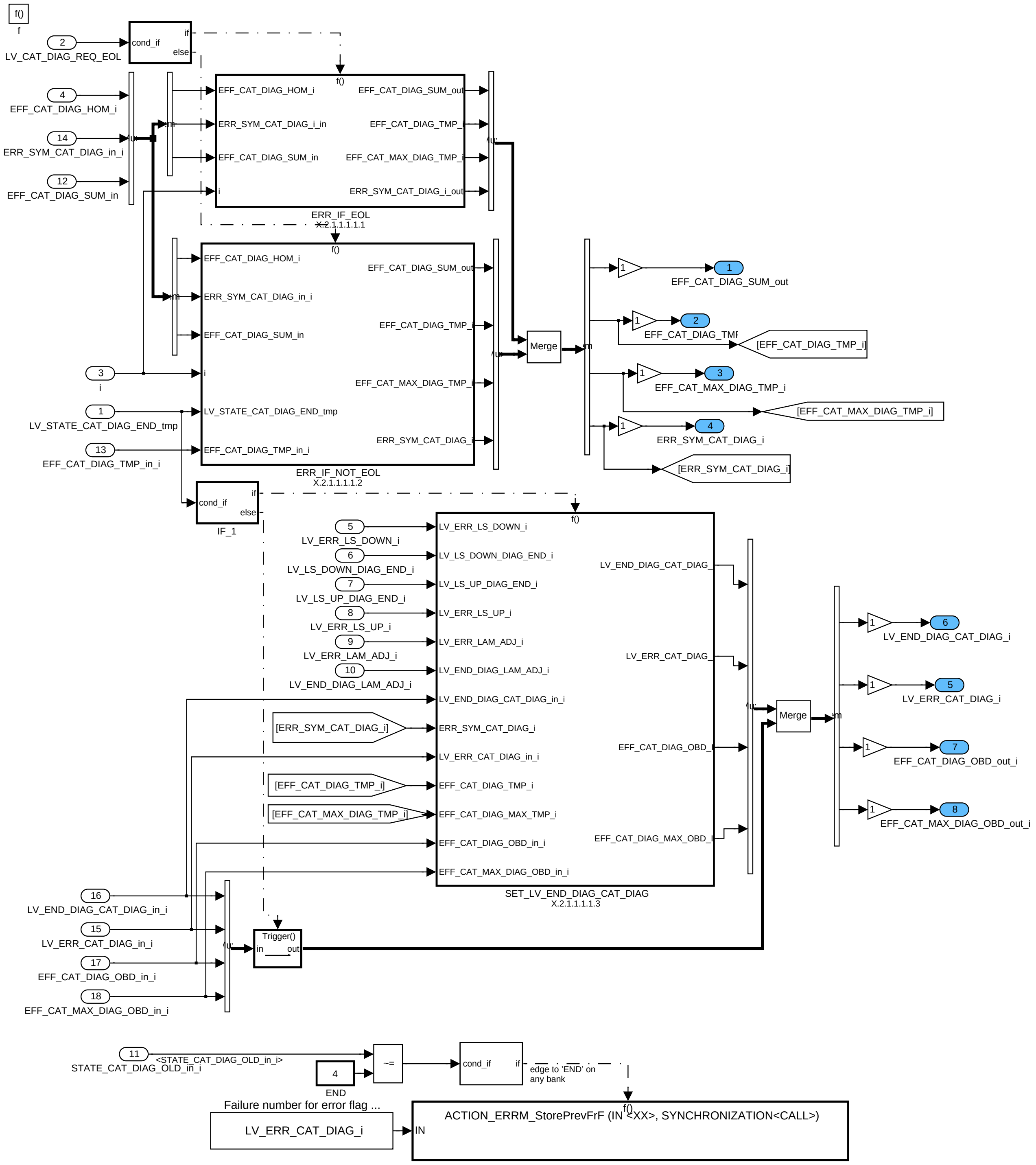


Figure 162:

18.16.2.1.1.1.1.1 No title given

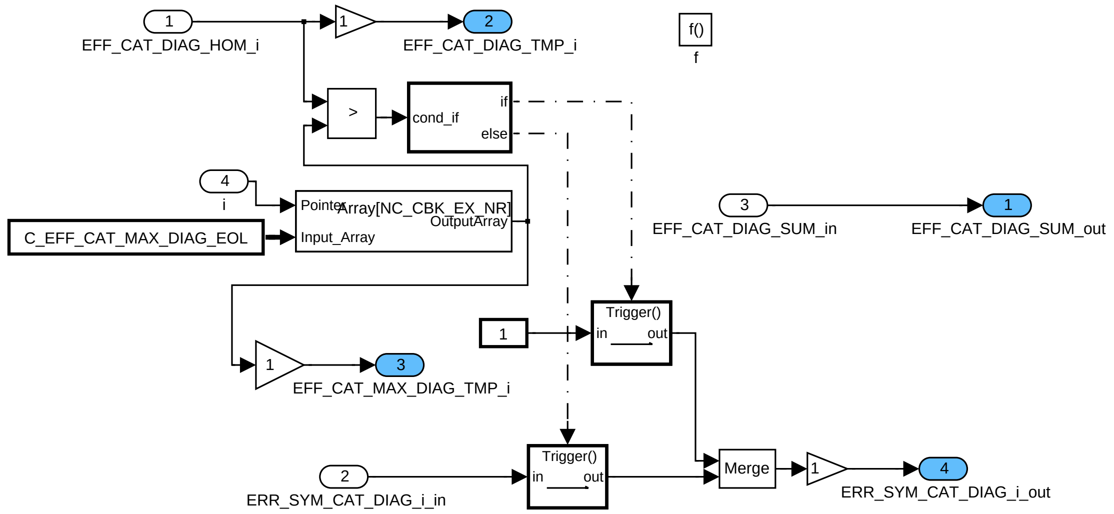


Figure 163:

18.16.2.1.1.1.1.2 Switch by LC_CAT_DIAG_RESU

This function is switchable with LC_CAT_DIAG_RESU: separate comparison of EFF_CAT_DIAG[i] with threshold and separate failure for every cylinder bank is stored, or from separate EFF_CAT_DIAG[i] is a common EFF_CAT_DIAG_SUM calculated and failure is stored only if common treshold C_EFF_CAT_MAX_DIAG_SUM is exceeded, or separate EFF_CAT_DIAG[i] exceeds a threshold C_EFF_CAT_MAX_50_DIAG. (Catalyst efficiency lower, then 50%). This modul is to call immediatelly after modul Catalyst efficiency diagnosis (OSC Method) is finished. For crossways question of symptoms i = the actually calculated bank and y = the other bank as actually calculated.

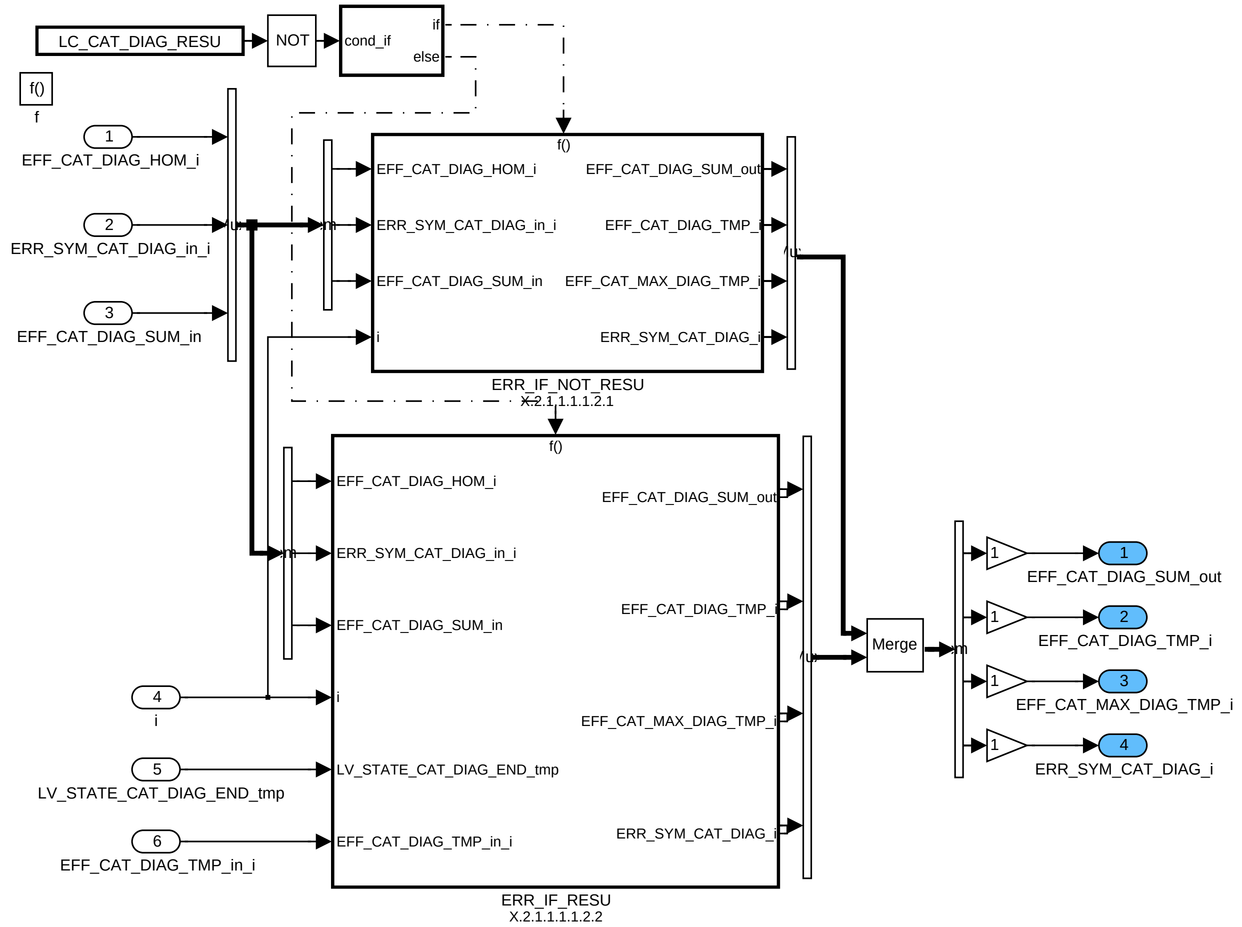


Figure 164:

18.16.2.1.1.1.1.2.1 No title given

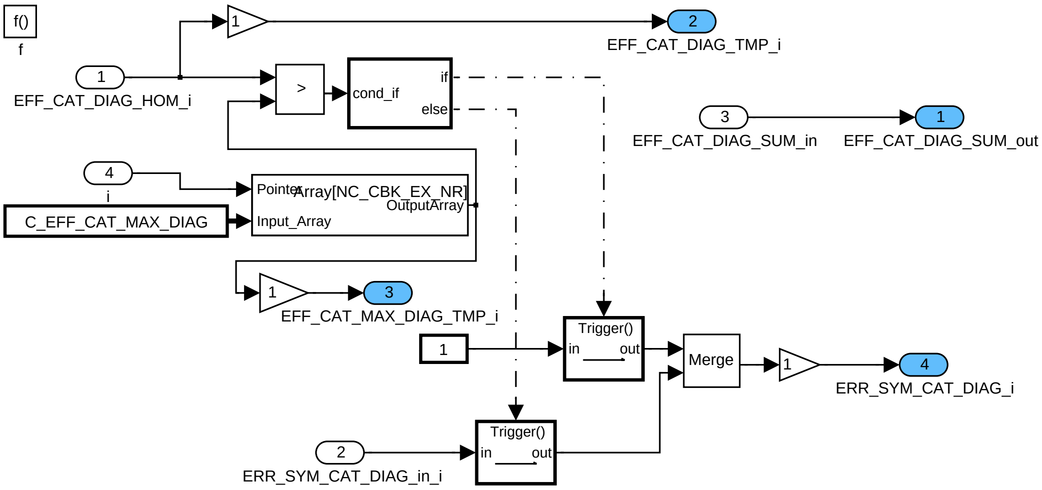


Figure 165:

18.16.2.1.1.1.1.2.2 No title given

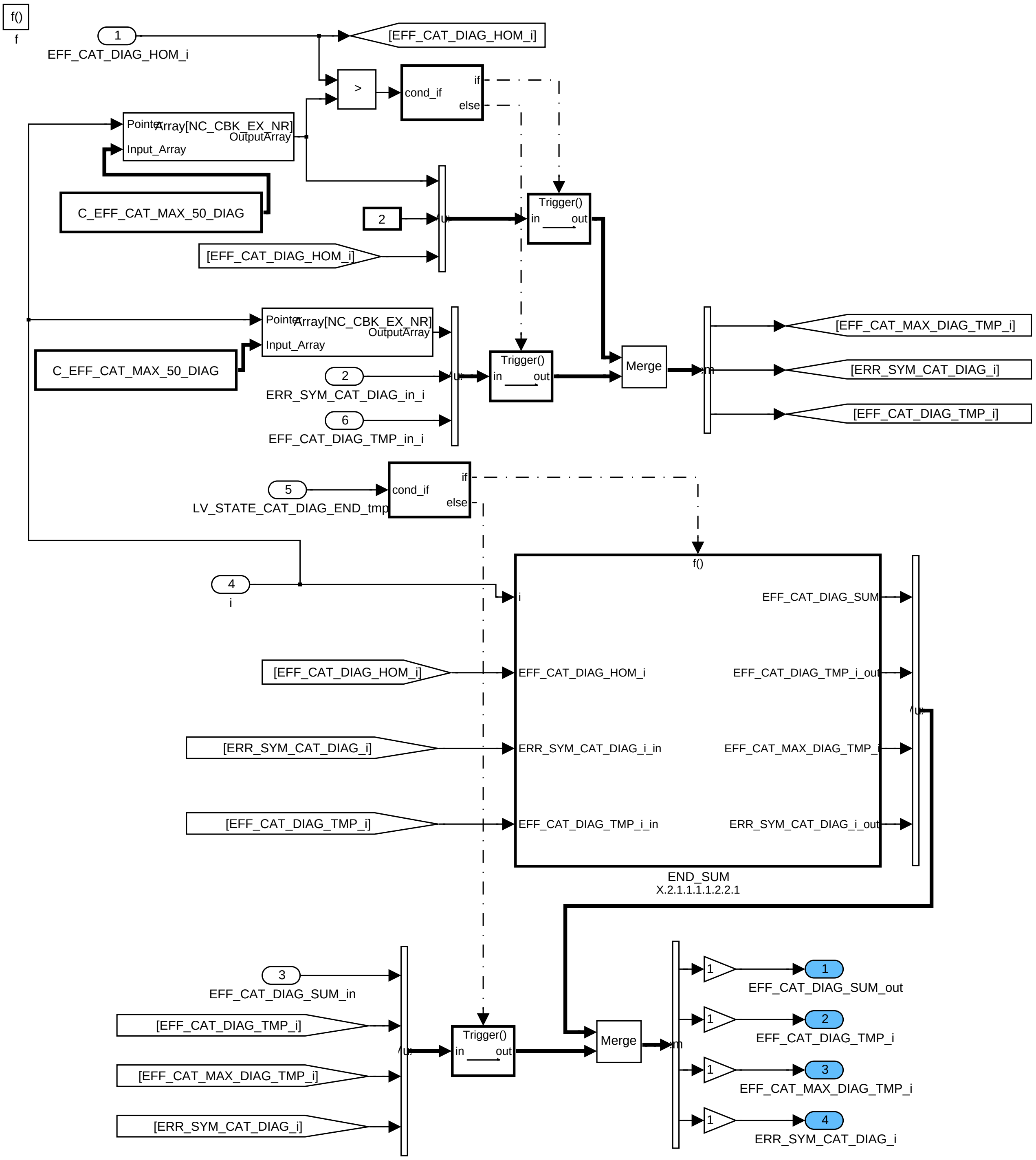


Figure 166:

18.16.2.1.1.1.1.2.2.1 No title given

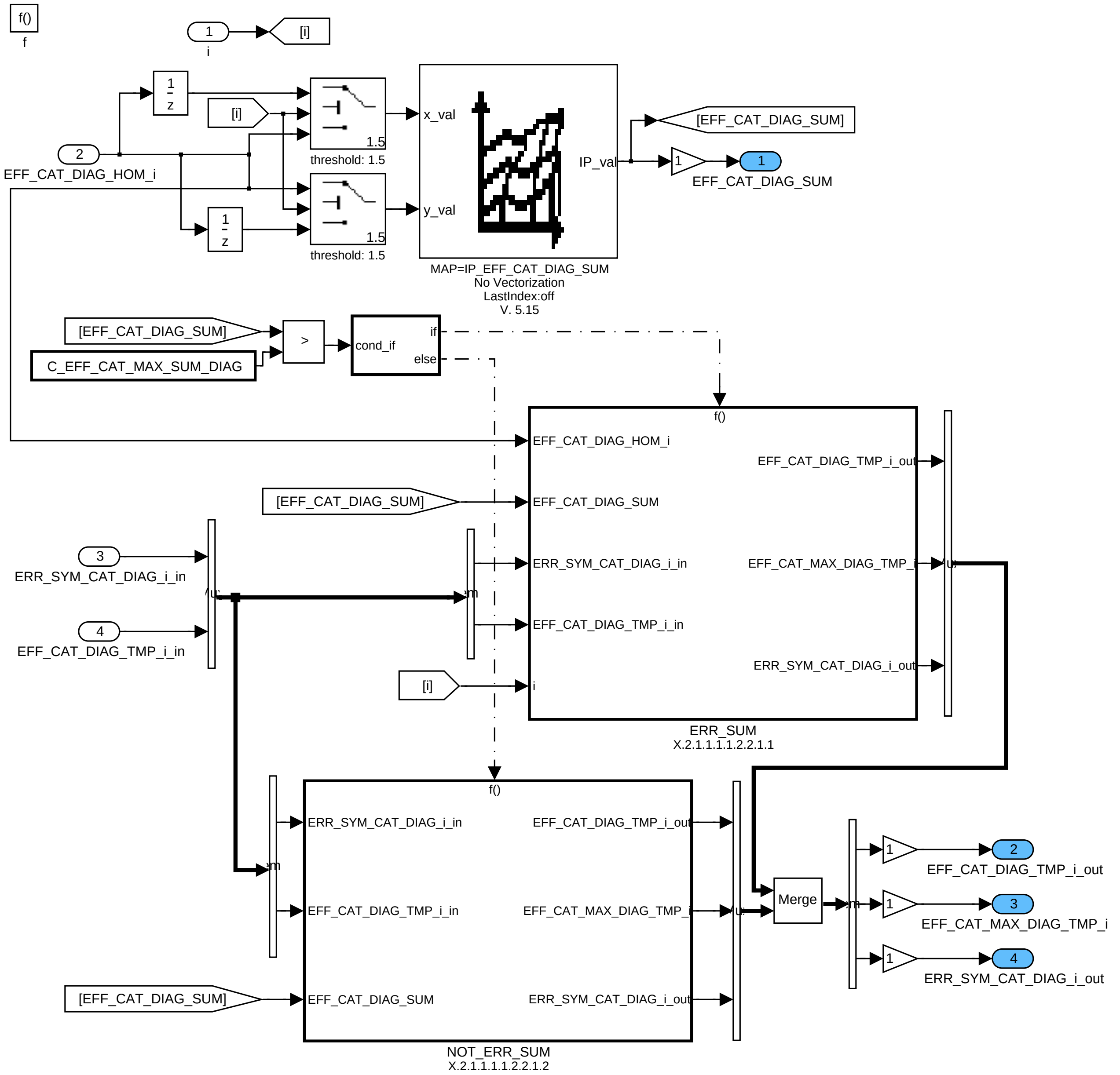


Figure 167:

Transmittal, reproduction, dissemination and/or editing of this document as well as utilization of its contents and communication there of to others without express authorization are prohibited. Offenders will be held liable for payment of damages. All rights created by patent grant or registration of a utility model or design patent are reserved.

No title given

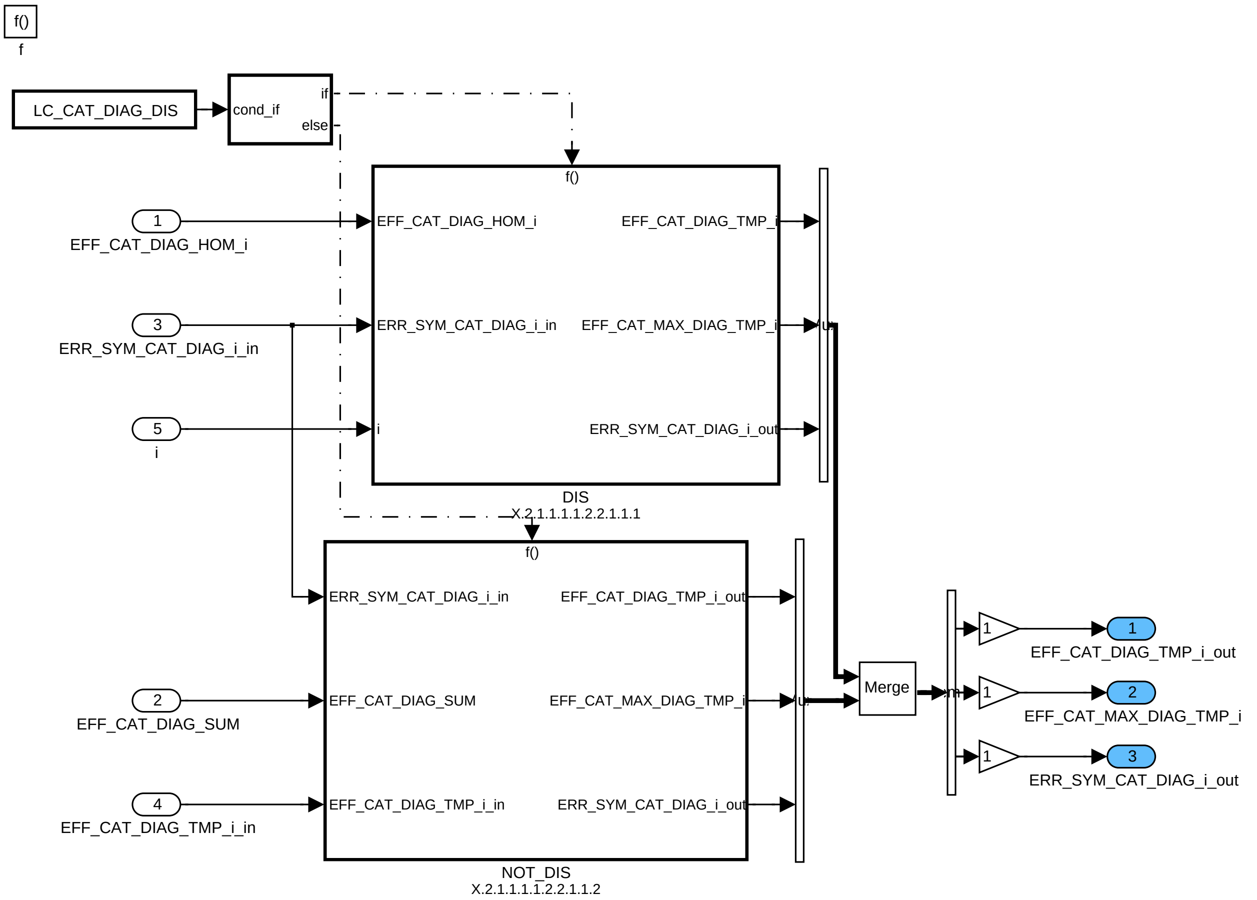


Figure 168:

No title given

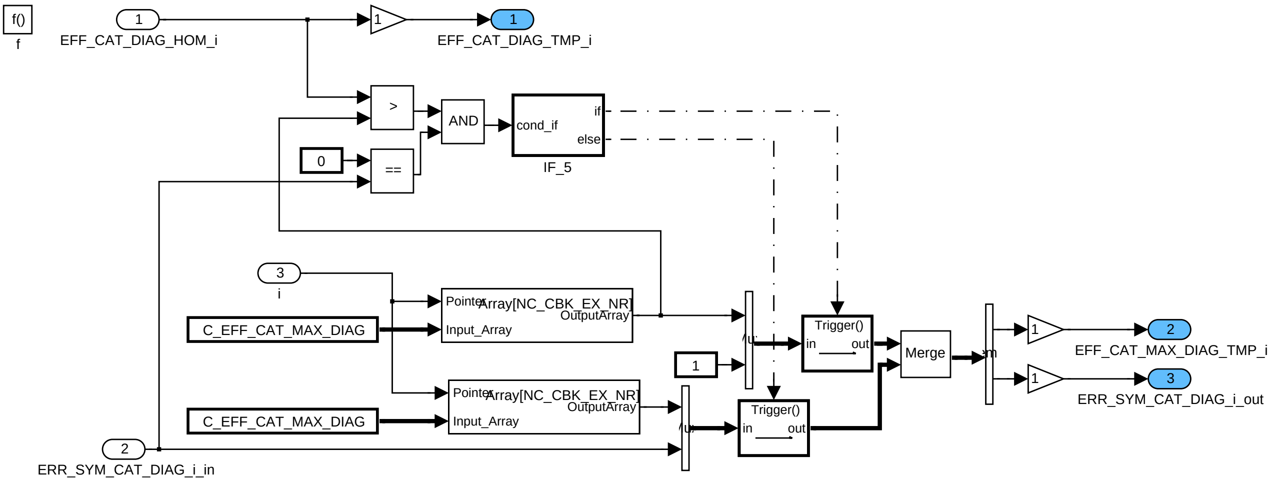


Figure 169:

No title given

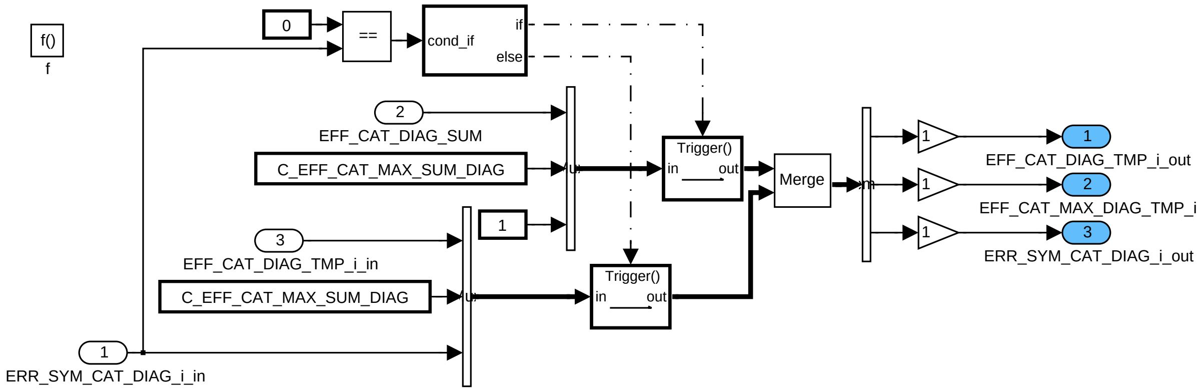


Figure 170:

No title given

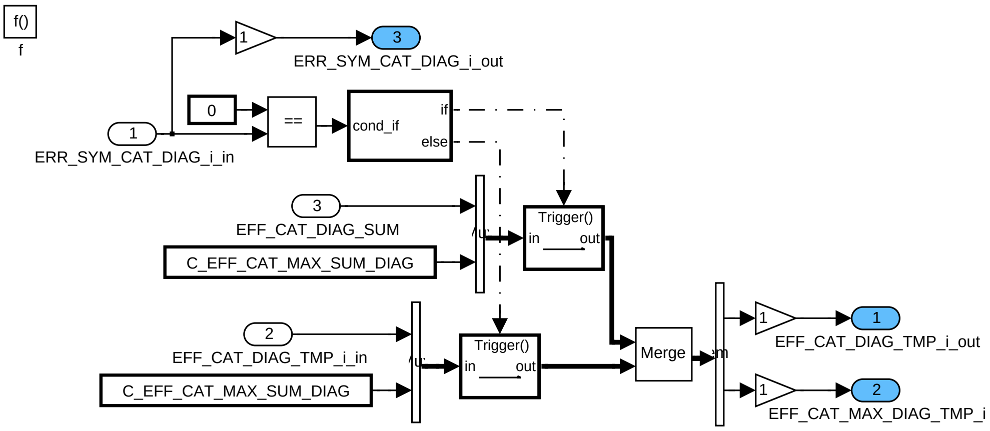


Figure 171:

18.16.2.1.1.1.1.3 Set LV_END_DIAG_CAT_DIAG[i]

When LV_END_DIAG_CAT_DIAG[i] is set to 1 the current failure status must be stored in the Error Management (by means of a service of ERRM).

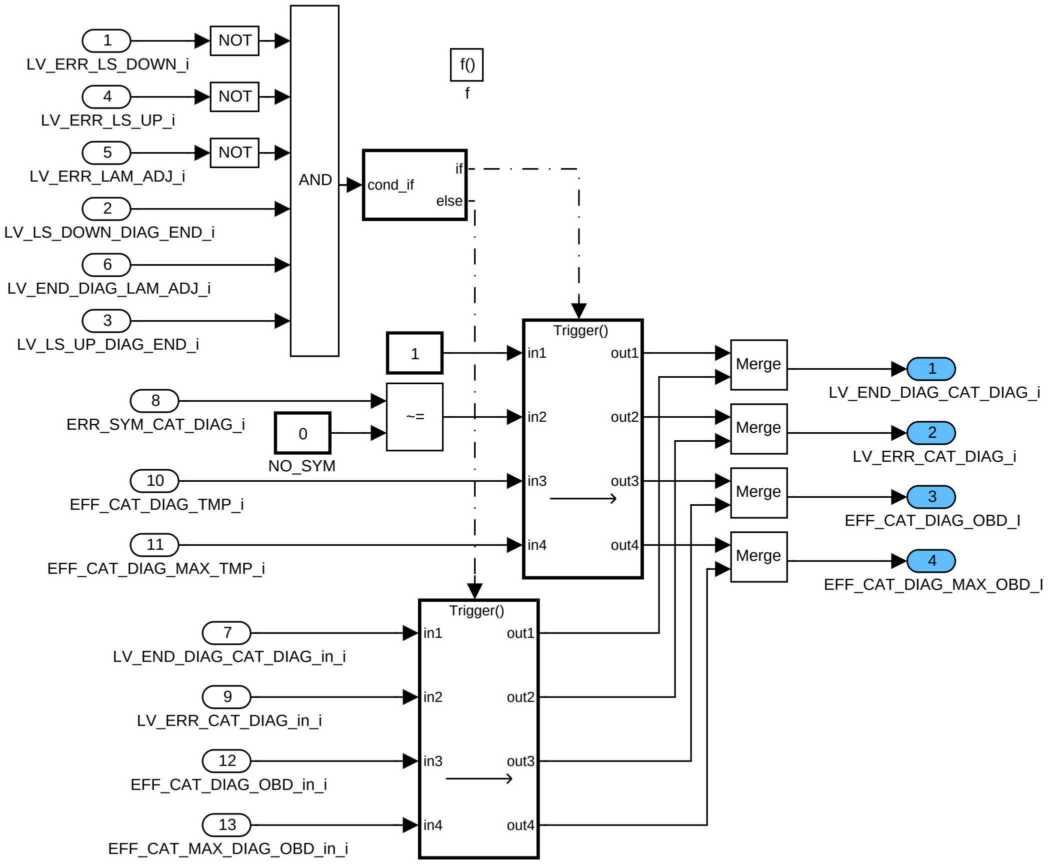


Figure 172:

18.16.2.1.1.2 Set readiness flag for the catalyst diagnosis through diagnosis from lean conditions

In case of lean combustion condition, a lean catalyst diagnosis is executed, but the homogeneuse diagnosis is not executed. In this case, readiness flag will be set by lean catalst diagnosis.

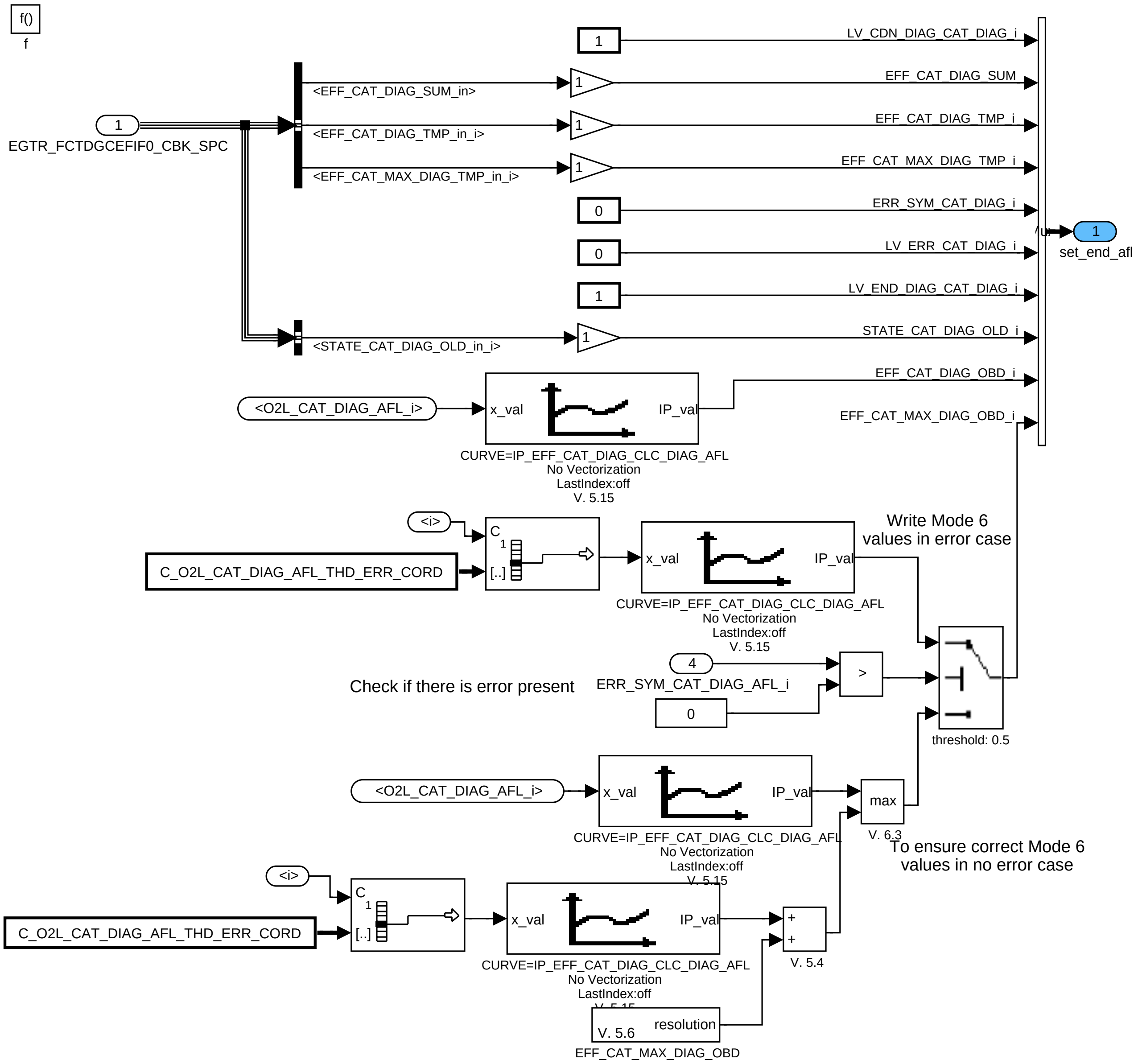


Figure 173:

18.16.3 Initialization at RESET

SDA needs initialisation of LV_END_DIAG_CAT_DIAG to run simulation. That is why LV_END_DIAG_CAT_DIAG is initialised to zero at Reset block. In the software this variable should be initialized by Error Management system.

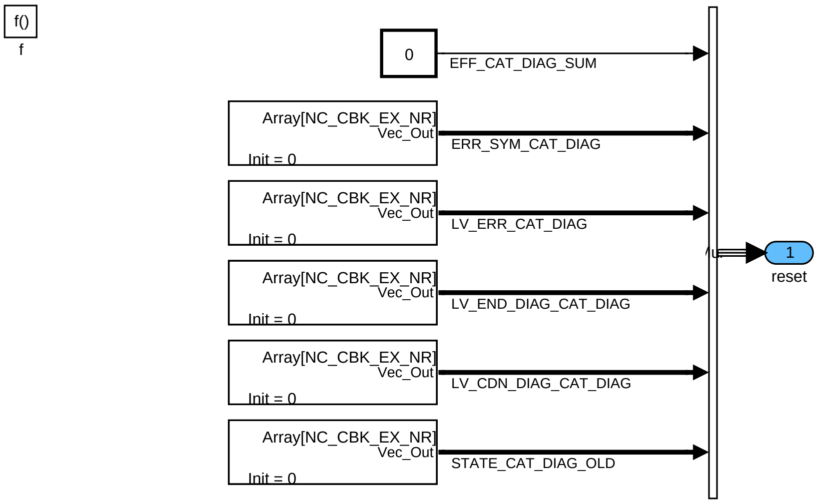

f()
f
0
EFF_CAT_DIAG_SUM
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
ERR_SYM_CAT_DIAG
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_ERR_CAT_DIAG
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_END_DIAG_CAT_DIAG
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_CDN_DIAG_CAT_DIAG
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
STATE_CAT_DIAG_OLD
1
reset

Figure 174:

18.17 OSC based TWC diagnosis for DI-engines at lean operation - activation conditions

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_O2L_CLC_ACT_CDN [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
General activation flag for calculation of oxygen capacity					
LV_O2L_CLC_ACT_CMB_CDN	V	0... 1H	0... 1	1	[-]
Activation for diagnosis regarding combustion conditions					
LV_O2L_CLC_ACT_LDC_TEMP_TMP [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Activation for diagnosis regarding TWC temperature limited dynamic					
LV_O2L_CLC_ACT_MAF_INT [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Activation for diagnosis regarding MAF integral after high load phase					
LV_O2L_CLC_ACT_OPP_TMP	V	0... 1H	0... 1	1	[-]
Activation of the diagnosis regarding operating point conditions					
LV_O2L_CLC_ACT_TCO_TMP	V	0... 1H	0... 1	1	[-]
Activation of the diagnosis regarding coolant temperature					
LV_O2L_CLC_ACT_TEMP_TMP [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Activation for diagnosis regarding TWC temperature					
MAF_INT_CAT_DIAG_AFL [NC_CBK_EX_NR]	V	0... FFFFH	0... 1820.41666667	0.0277778	[g]
Maf integral after high load phase					
TEMP_LDC_CAT_DIAG_AFL [NC_CBK_EX_NR]	V	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
Filtered modelled catalyst dynamic temperature					
TQI_LDC_CAT_DIAG_AFL	V	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Filtered TQI_AV					

Input Data:

LV_ST_END	LV_NT_ACT	MAF_CYL	N_32
TEMP_CAT_DYN_MDL [NC_CBK_EX_NR]	TCO	NC_CBK_EX_NR	LV_INH_DIAG_CAT_DIAG_AFL [NC_CBK_EX_NR]
LV_END_DIAG_CAT_DIAG_AFL [NC_CBK_EX_NR]	LV_END_DIAG_AFL_TMP [NC_CBK_EX_NR]	VS	STATE_NOX
EFF_CAT_DIAG [NC_CBK_EX_NR]	TNT_MDL_MV_SNG [NC_NT_NR]	TQI_AV	NC_NT_NR

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CRLC_TEMP_CAT_DIAG_AFL	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation constant for calculation of filtered TWC temperature					
C_CRLC_TQI_CAT_DIAG_AFL	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation constant for calculation of filtered TQI_AV					
C_MAF_INT_MAX_CAT_DIAG_AFL	1	0... FFFFH	0... 1820.41666667	0.0277778	[g]
MAF_CYL integral for delay of the diagnosis activation after high load phase					
C_N_32_DOWN_LIM_CAT_DIAG_AFL	1	0... FFH	0... 8160	32	[rpm]
Lower limit of N_32 to allow of OSC calculation					

C_N_32_UP_LIM_CAT_DIAG_AFL	1	0... FFH	0... 8160	32	[rpm]
Upper limit of N_32 to allow of OSC calculation					
C_TCO_DOWN_LIM_CAT_DIAG_AFL	1	0... FEH	-48... 142.5	0.75	[°C]
Lower limit of TCO to allow of OSC calculation					
C_TCO_UP_LIM_CAT_DIAG_AFL	1	0... FEH	-48... 142.5	0.75	[°C]
Upper limit of TCO to allow of OSC calculation					
C_TEMP_DYN_H_THD_CAT_DIAG_AFL	1	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
Maximal dynamic TWC temperature for activation of the diagnosis					
C_TEMP_LDC_CAT_DIAG_AFL	1	8000... 7FFFH	-2048... 2047.9375	0.0625	[°C]
Threshold for deactivation of OSC calculation regarding TWC temp. limited dynamic					
C_TEMP_UP_LIM_CAT_DIAG_AFL	1	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
Upper limit of TWC temperature to allow of OSC calculation					
C_TNT_DOWN_LIM_CAT_DIAG_AFL	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Lower limit of NT temperature to allow of OSC calculation					
C_TNT_UP_LIM_CAT_DIAG_AFL	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Upper limit of NT temperature to allow of OSC calculation					
C_TQI_LDC_CAT_DIAG_AFL	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Threshold for deactivation of OSC calculation regarding TQI_AV limited dynamic					
C_VS_DOWN_LIM_CAT_DIAG_AFL	1	0... FFH	0... 255	1	[km/h]
Minimum vehicle speed for activation of the diagnosis					
C_VS_UP_LIM_CAT_DIAG_AFL	1	0... FFH	0... 255	1	[km/h]
Maximum vehicle speed for activation of the diagnosis					
IP_MAF_DOWN_LIM_CAT_DIAG_AFL	6	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
LDPM_N_32_IP_MAF	6	0... FFH	0... 8160	32	[rpm]
Lower limit of MAF for activation of the diagnosis					
IP_MAF_UP_LIM_CAT_DIAG_AFL	6	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
LDPM_N_32_IP_MAF	6	0... FFH	0... 8160	32	[rpm]
Upper limit of MAF for activation of the diagnosis					
IP_TEMP_DOWN_LIM_CAT_DIAG_AFL	4	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
LDP_EFF_CAT_DIAG_AFL_IP_TEMP	4	0... FFH	0... 1.9921875	7.8125e-3	[-]
Lower limit of the dynamic catalyst temperature depending on catalyst aging state					

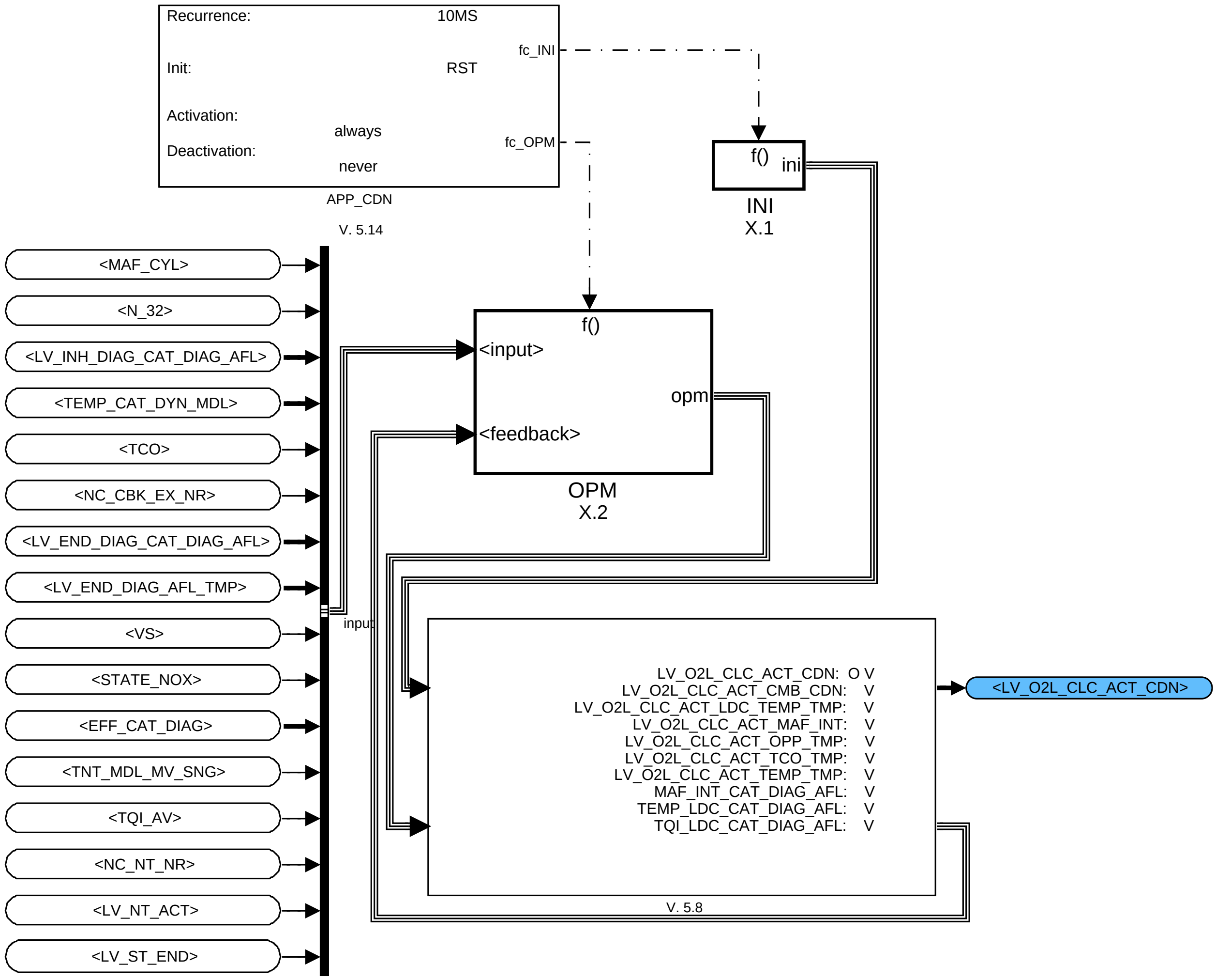
General Information

Activation condition for calculation of OSC values in module "TWC diagnosis at lean operation - OSC calculation".
If two separate exhaust bank systems are concerned, then:
  i = 1, for cylinder bank 1
  i = 2, for cylinder bank 2
Otherwise:
  i = 1

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.1 06−Apr−2006

Figure 175: Overview level

18.17.1 INITIALIZATION

All variables are initialized with "0"

18.17.2 FORMULA SECTION

The bank specific and bank not specific calculations are done separate

18.17.2.1 Function active - bank not specific calculations

In this case are checked:
Operating point
Coolant temperature
Combustion conditions

18.17.2.1.1 Check of the operating point

Deactivation of the diagnosis, when the mass air flow and rotational speed exceed defined range. Additional, dynamic of the torque request are checken

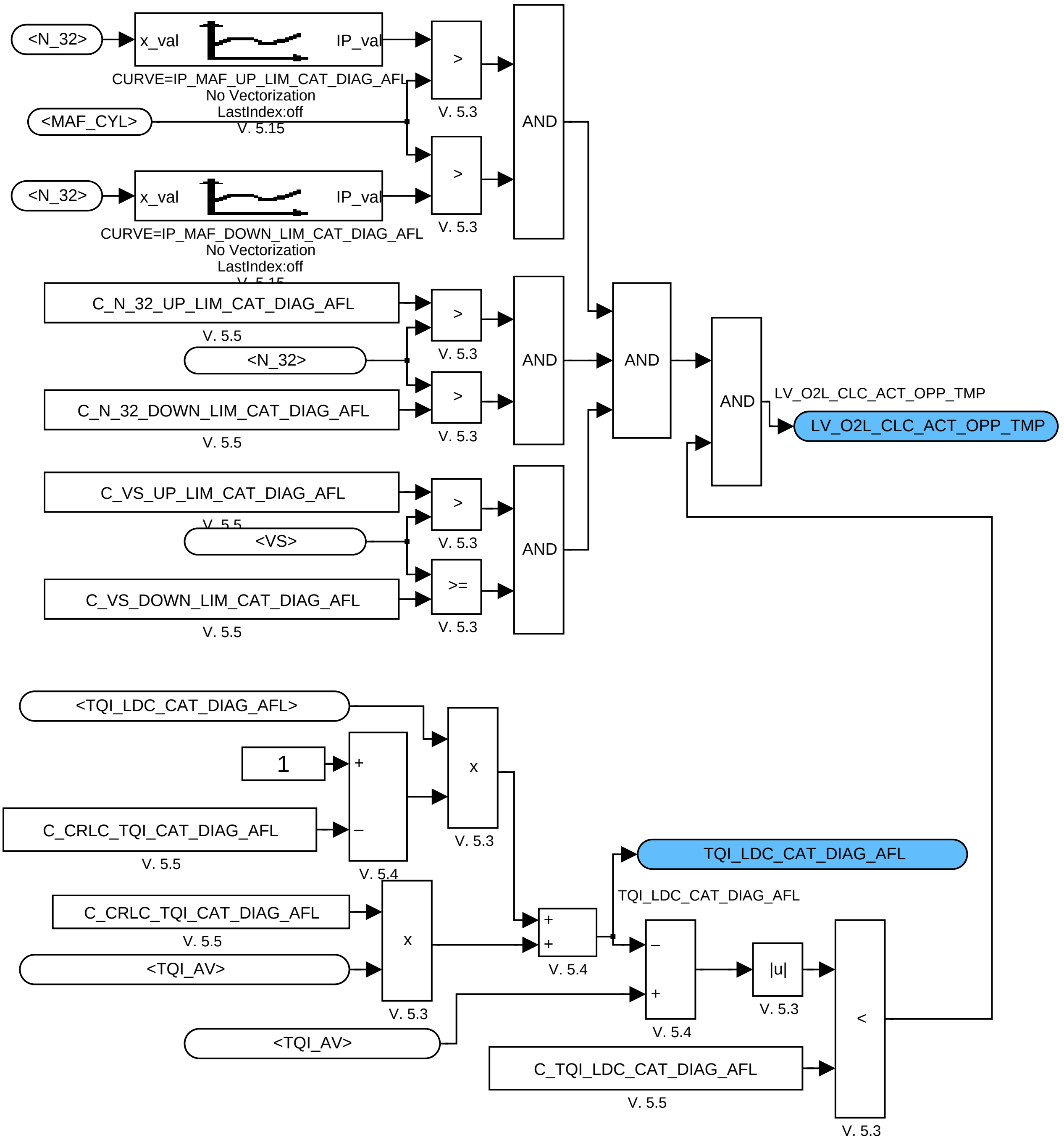


Figure 176:

18.17.2.1.2 Check of coolant temperature

Deactivation of the diagnosis, when the coolant temperature exceeds defined range

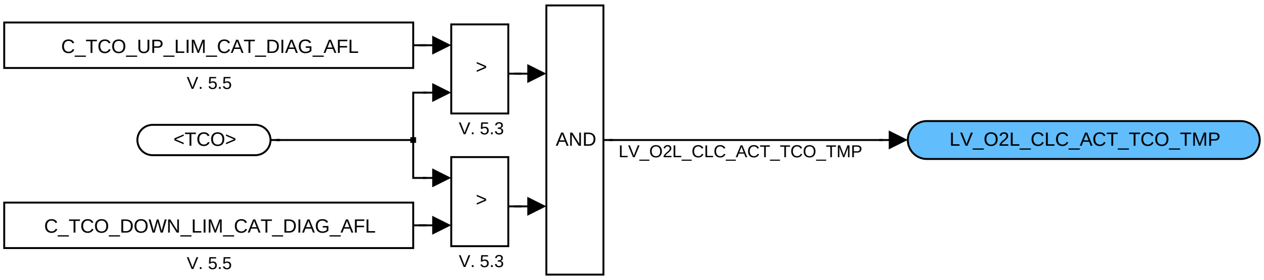


Figure 177:

18.17.2.1.3 Check of the combustion conditions

Deactivation when the NOx after treatment equipment is not active

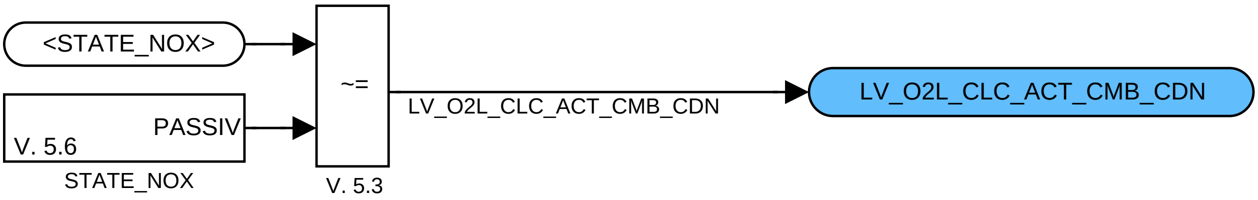


Figure 178:

18.17.2.2 Function active - bank specific calculations

18.17.2.2.1 Check of bank specific activation conditions

In this case:
The static temperature of TWC monolith is checked
The limited dynamic of TWC monolith temperature is checked
The MAF integral is calculated after high load phase
The final activation condition flag is calculated

18.17.2.2.1.1 Check of the TWC and NT monolith temperatures

Deactivation of the diagnosis, when the monolith temperatures of TWC and NT (NOx Trap) exceed defined range. For the NT, a mean value of the both monolith is used. In case of down limit of the TWC temperature, it can be different for catalysts with different aging state

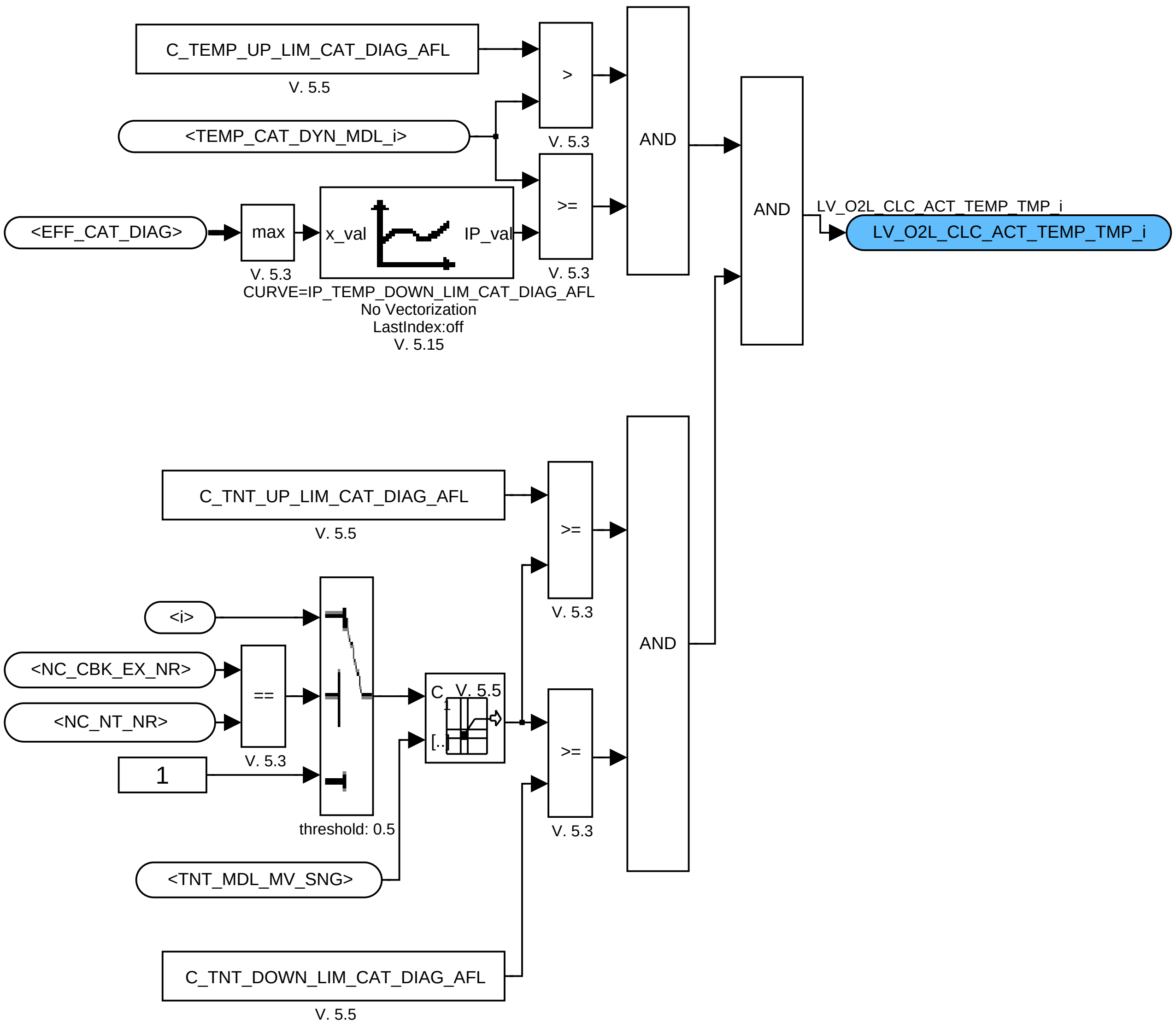


Figure 179:

18.17.2.2.1.2 Check of the limited dynamic of the TWC monolith temperature

The difference between filtered and not filtered TWC monolith temperature is shall not exceed defined threshold

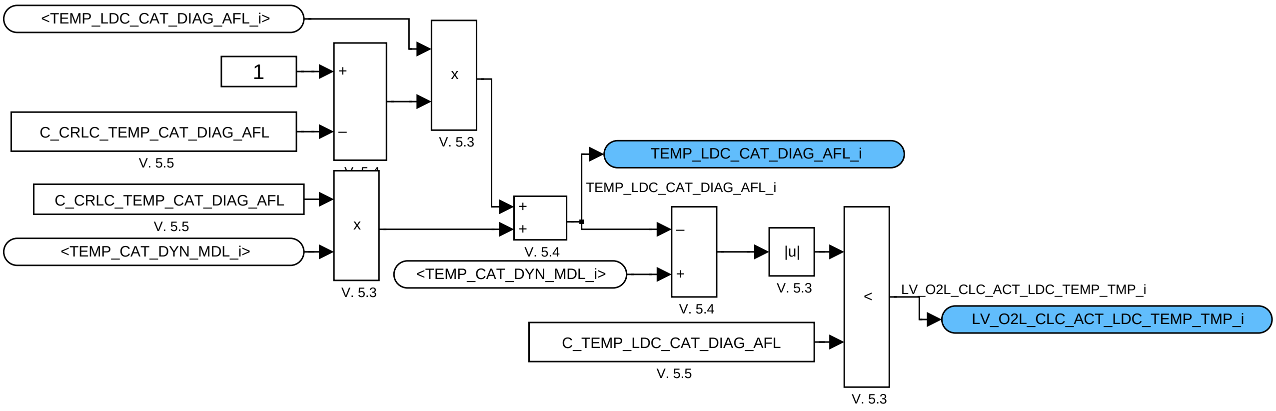


Figure 180:

18.17.2.2.1.3 Check of the MAF integral

Deactivation of the diagnosis after high load phase till defined MAF integral is not released

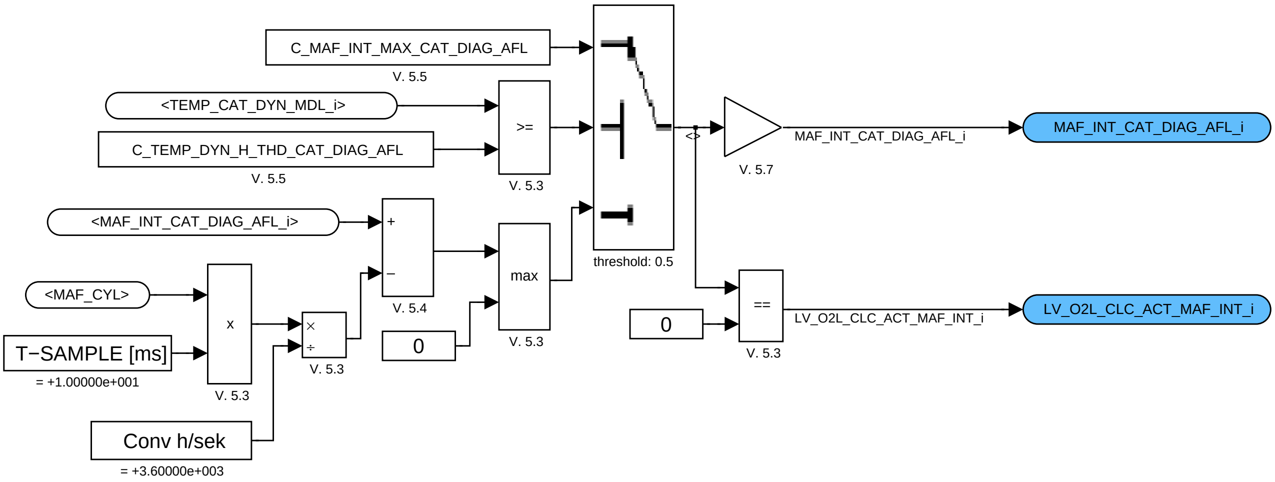


Figure 181:

18.17.2.2.1.4 Calculation of the final activation flag

The end activation flag results from the condition checked in this module, the inhibition condition and the diagnosis end condition. The diagnosis is deactivated when the calculation of it results is finished

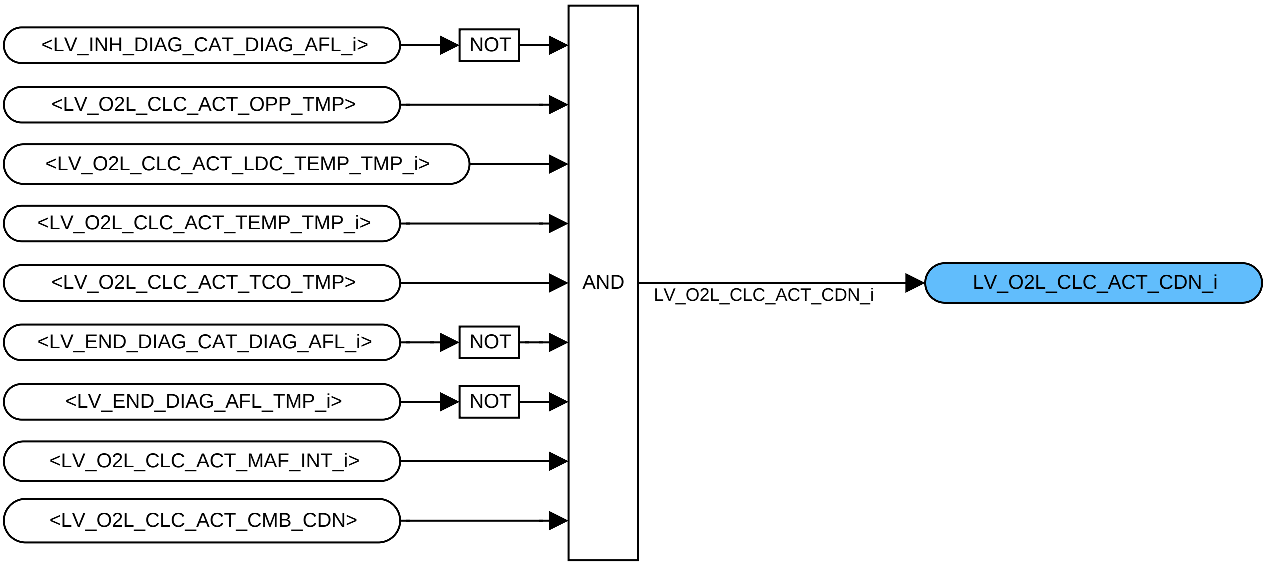


Figure 182:

18.17.2.3 Check activation conditions

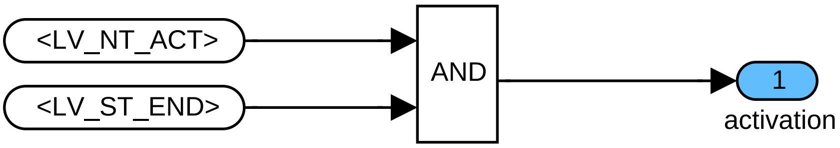


Figure 183:

18.17.2.4 Function not active

All variables are set to "0"

18.18 OSC based TWC diagnosis for DI-engines at lean operation - Calculation of the Lambda Setpoint

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LAMB_SP_DIAG_AFL [NC_CBK_EX_NR]	O/V	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda setpoint for precatalyst lean diagnosis					
LV_DIAG_AFL_REQ	O/V	0... 1H	0... 1	1	[-]
Request for precatalyst lean diagnosis					
LV_LAMB_SP_DIAG_AFL_PARK [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Indicator of lambda value position at park state					
LV_NT_RGN_REQ_PREV	-	0... 1H	0... 1	1	[-]
Value of LV_RGN_NT_REQ from previous calculation cycle					
MAF_INT_LAMB_SP_CAT_DIAG_AFL	V	0... FFFFH	0... 1820.42	0.0277778	[g]
MAF integral for keeping of rich lambda setpoint before switch to diagnosis					
STATE_RGN_CAT_DIAG_AFL [NC_CBK_EX_NR]	O/V	0... 3H	0... 3	1	[-]
State for calculation of lean TWC diagnosis due state of system regeneration					

Input Data:

LV_ST_END	LV_NT_ACT	LV_RGN_NT_REQ	STATE_RGN [NC_CBK_EX_NR]
LV_LAMB_SP_SWI	NC_CBK_EX_NR	N_32	MAF
STATE_CAT_DIAG_AFL [NC_CBK_EX_NR]	MAF_CYL	LV_O2L_CLC_ACT_CDN [NC_CBK_EX_NR]	

Calibration Data:

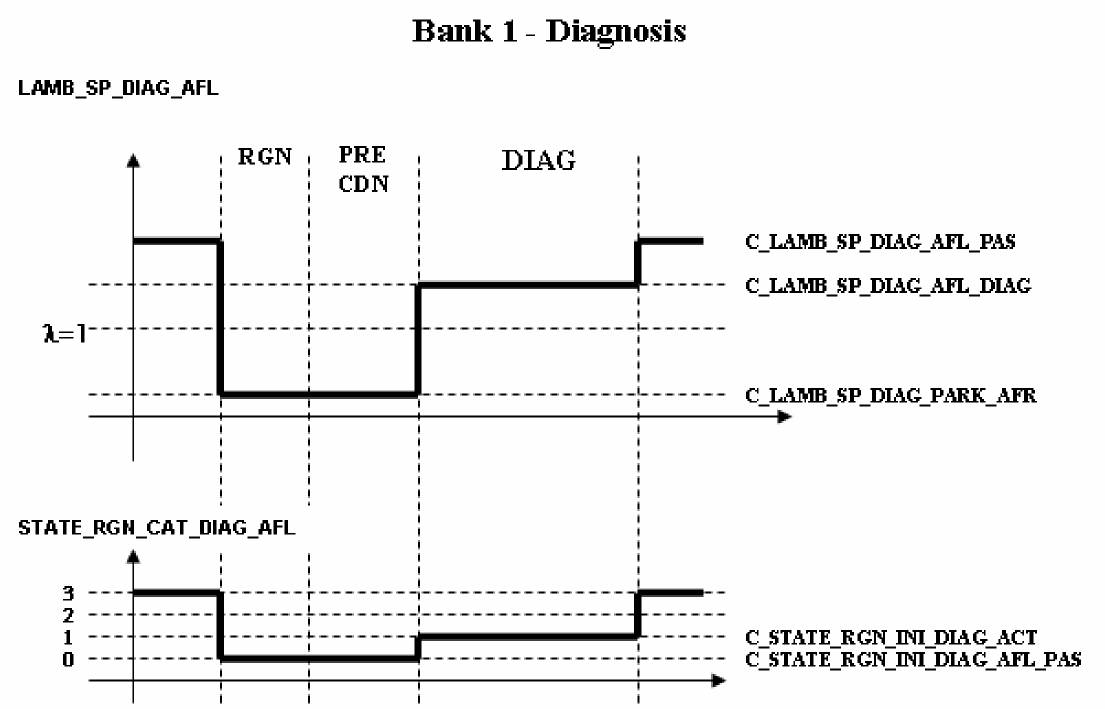
Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_LAMB_SP_DIAG_AFL_DIAG	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda setpoint for lean part of the lean precatalyst diagnosis					
C_LAMB_SP_DIAG_AFL_PAS	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda setpoint for passive part of the lean precatalyst diagnosis					
C_LAMB_SP_DIAG_PARK_AFR	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda setpoint for preconditioning phase before switching to diagnosis conditions					
C_LAMB_SP_DIAG_PARK_BAS	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Basis lambda setpoint of lean precatalyst diagnosis at park phase					
C_LAMB_SP_DIAG_PARK_SHIFT	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda shift from basis lambda at park phase of the precatalyst lean diagnosis					
C_MAF_INT_LAMB_SP_AFR	1	0... FFFFH	0... 1820.42	0.0277778	[g]
Threshold for MAF integral for keeping of the rich lambda setpoint					
C_STATE_RGN_DIAG_AFL_ACT	1	0... 3H	0... 3	1	[-]
Value of STATE_RGN_CAT_DIAG_AFL for activation of diagnosis					
C_STATE_RGN_INI_DIAG_AFL [NC_CBK_EX_NR]	1	0... 3H	0... 3	1	[-]
Initialization for the lambda setpoint for precatalyst lean diagnosis at active state					

C_STATE_RGN_INI_DIAG_AFL_PAS [NC_CBK_EX_NR]	1	0... 3H	0... 3	1	[-]
Initialization for the lambda setpoint for precatalyst lean diagnosis at passive state					
IP_LAMB_SP_DIAG_PARK_I	8*4	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDP_N_32_IP_LAMB_SP_DIAG_PARK_I	4	0... FFH	0... 8160	32	[rpm]
LDP_MAF_IP_LAMB_SP_DIAG_PARK_I	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Integration step for lambda setpoint for lean precatalyst diagnosis at park state					
LC_DIAG_AFL_CBK_SYN	1	0... 1H	0... 1	1	[-]
If true, reset of lambda setpoint flag after all banks finished, otherwise after first bank finished					
LC_STATE_RGN_DIAG_AFL	1	0... 1H	0... 1	1	[-]
Switch between regeneration states for calculation of O2L values					

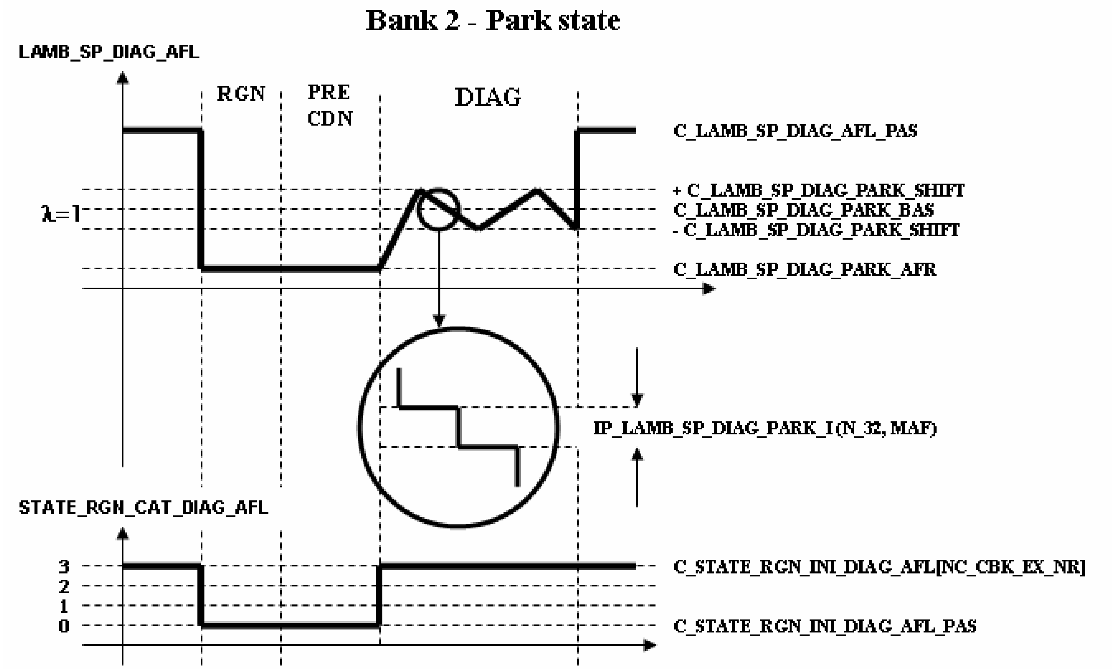
General Information

For the configuration without binary sensor after one bank, phases with rich and lean lambda value are required. When this conditions are not given by NOx Catalyst Regeneration, then a separately lambda setpoint directly after regeneration is generated. Directly before switching to the diagnosis, rich precondition of the catalysts is possible. Moreover, for each bank is used different lambda setpoint strategy. While at first bank (Picture 1, diagnosis) are the O2L values of the precatalyst and of the whole exhaust line (precatalyst + NOx Trap) calculated, the second bank (Picture 2, park state) is parked to minimize it influence to the diagnosed bank. The strategy is successively switched between both banks.

Picture 1: Lambda setpoint strategy for the diagnosed bank:



Picture 2: Lambda setpoint strategy for the parked bank:

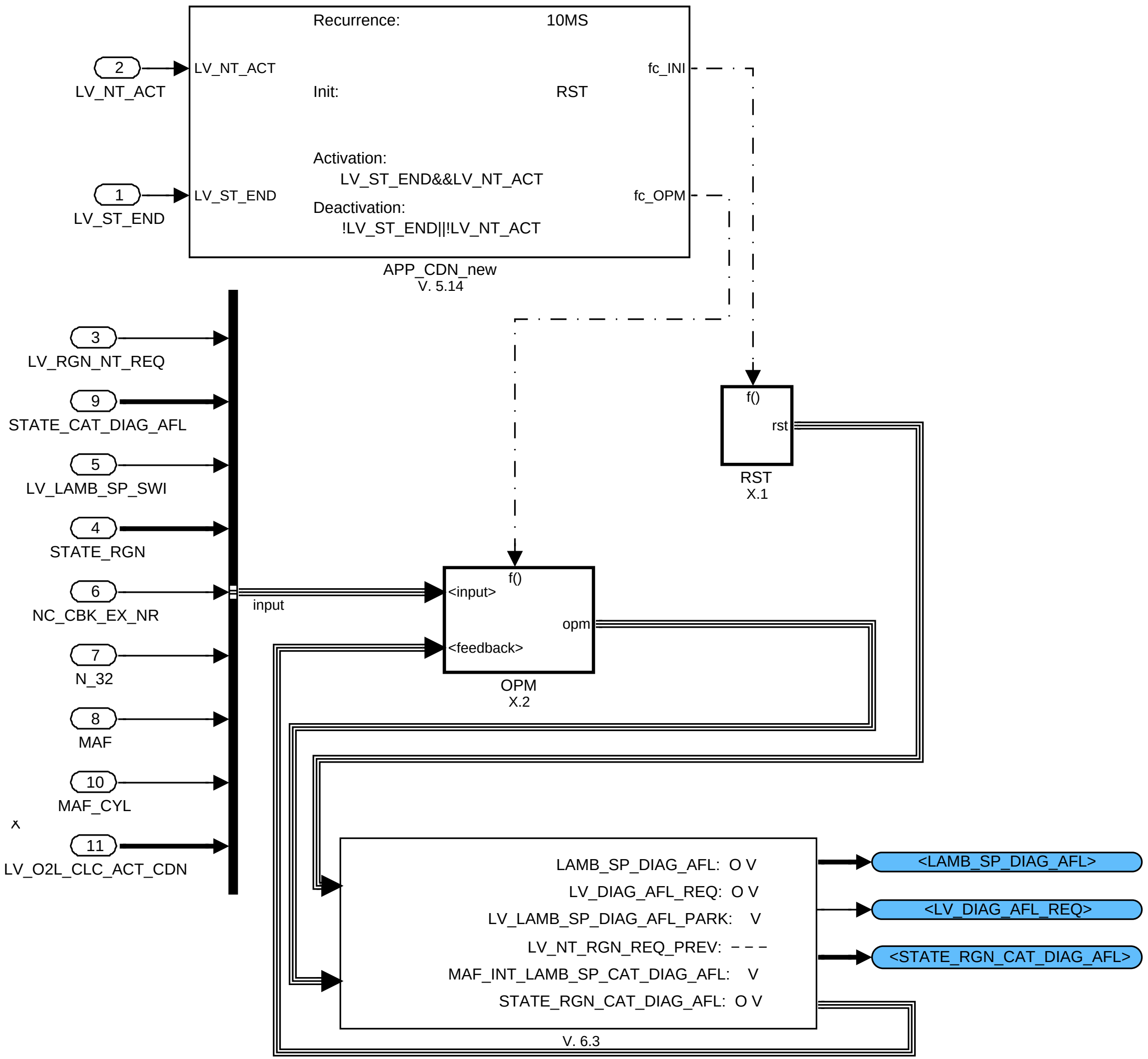


Bank 2 - Park state

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: LV_ST_END && LV_NT_ACT
Deactivation: !LV_ST_END || !LV_NT_ACT

Function description



SDA_SRS / SDA V 5.2.3 22−Jan−2007

Figure 184:

18.18.1 INITIALIZATION AT RESET

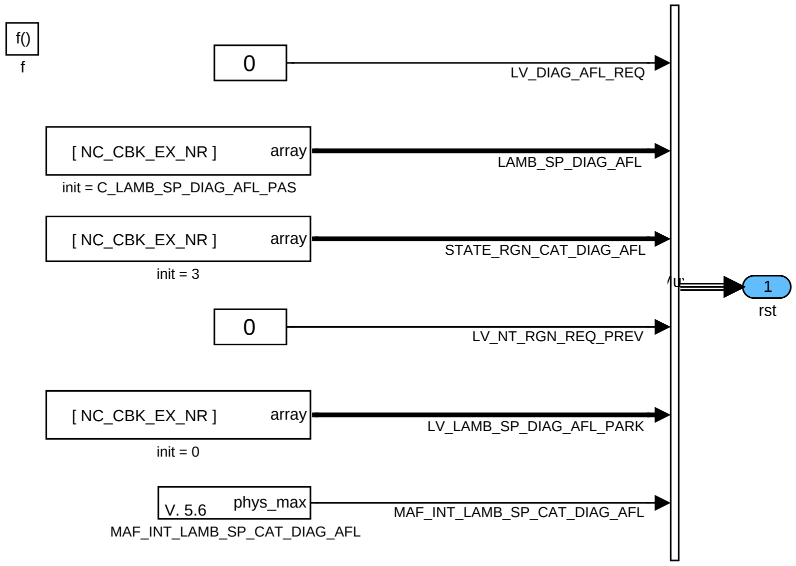


Figure 185:

18.18.2 FORMULA SECTION

The calculation of the STATE_RGN_CAT_DIAG is only active when the lambda behaviour during regeneration doesn't suit the requirements for the diagnosis. Otherwise the STATE_RGN is taken for the other modules of this diagnosis.
Then the activation flag for calculation of O2L values is checked. If any bank is not active, all variables are initialized.
The function contains of two main sections:
1.2.1: Calculation of the lambda setpoint and the regeneration state for the diagnosis (LAMB_SP_CLC)
1.2.2: Passive state (LAMB_SP_NOT_CLC)
1.2.3: Initialization when the activation conditions are not fulfilled

18.18.2.1 Initialization when the activation flag for O2L calculation at any bank is false

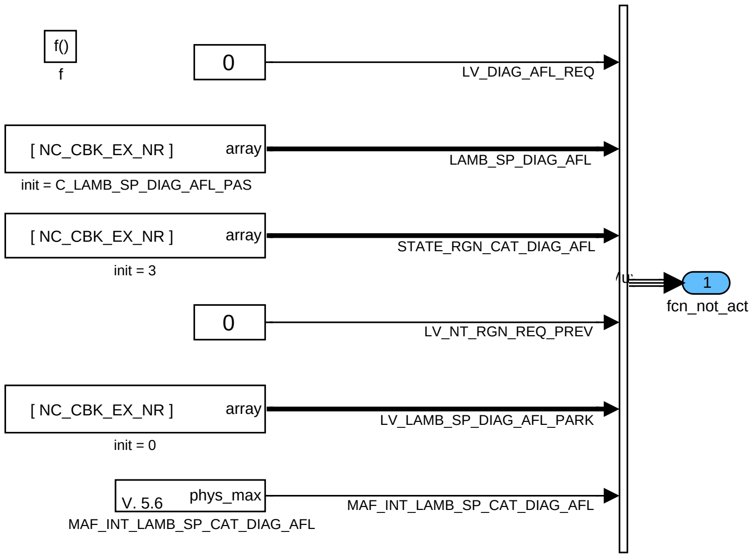


Figure 186:

18.18.2.2 CALCULATION OF THE LAMBDA SETPOINT AND THE REGENERATION STATE FOR THE DIAGNOSIS (LAMB_SP_CLC)

The calculation is split up into following sections:
- Request for the diagnosis
- Calculation of the regeneration state for the diagnosis
- Calculation of the lambda setpoint
- Calculation of the old variables

18.18.2.2.1 Request for the diagnosis

As long as the request LV_DIAG_AFL_REQ is set, the lambda set point for the system is defined by this function. This request is reset, when either all, or just any bank (depending on initialization STATE_RGN_CAT_DIAG_AFL of that bank, which will not be diagnosed in current cycle) switches to STATE_CAT_DIAG_AFL=PASSIVE

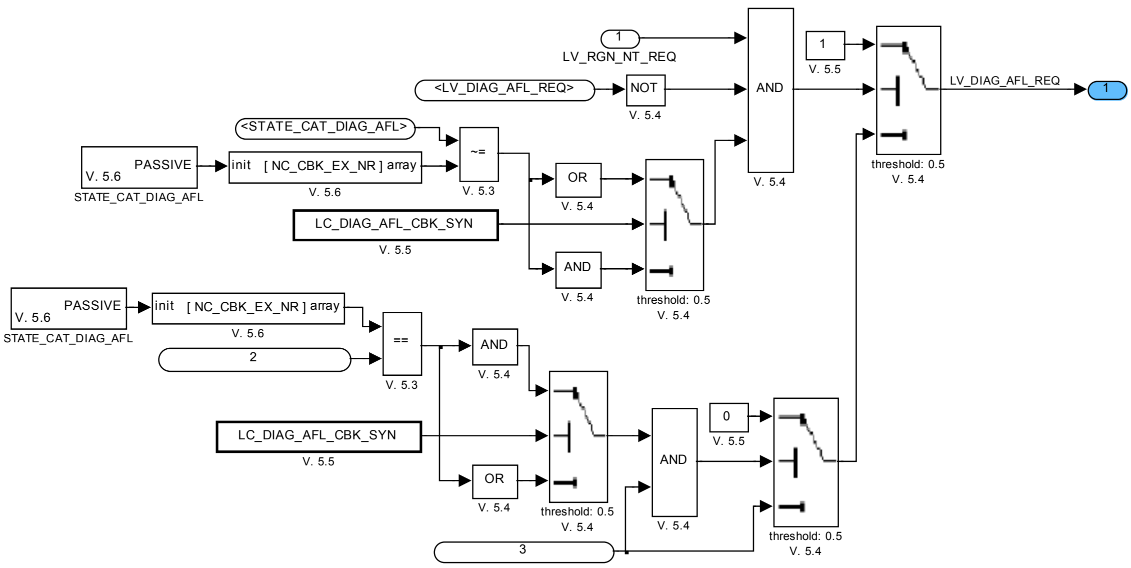


Figure 187:

18.18.2.2.2 Calculation of the regeneration state for the diagnosis

The own regeneration state for the diagnosis is initialized at the beginning of the NOx Trap regeneration. When the Nox Trap regeneration is finished, then depending on calibration a precondition phase is started. When the precondition is finished, then the own regeneration state is initialized accordingly for activation of the diagnosis at relevant bank and neutral park of the other

18.18.2.2.2.1 Check conditions

Check of the NOx Trap regeneration state, precondition state and bank alternate bit. The calculations follow in next picture

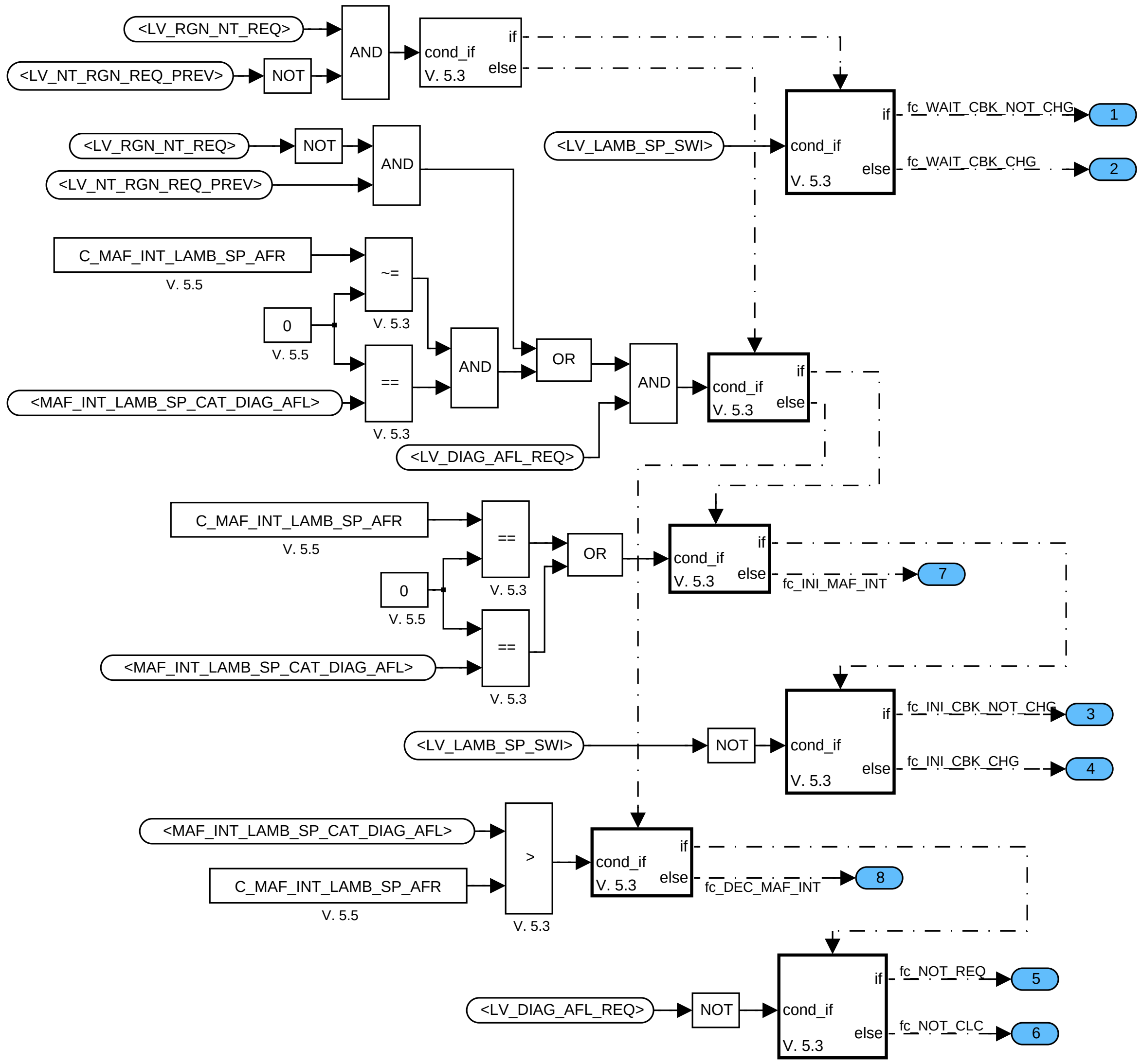


Figure 188:

18.18.2.2.2.2 Calculations

Calculation of the own regeneration state, lambda set point and MAF integral is done in following cases:

a) Waiting for the end of the Nox Trap regeneration and precondition, banks not interchanging
b) Waiting for the end of the Nox Trap regeneration and precondition, banks interchanging accordingly to C_STATE_RGN_INI_DIAG_AFL_PAS
c) Initialization for diagnosis state without interchanging of the banks
d) Initialization for diagnosis state with interchanging of the banks accordingly to C_STATE_RGN_INI_DIAG_AFL
e) Request for the diagnosis d sn't exist
f) Request for the diagnosis exists, but initializations don't take place
g) Initialization of the MAF integral for precondition phase
h) Decrementing of the MAF integral during precondition phase

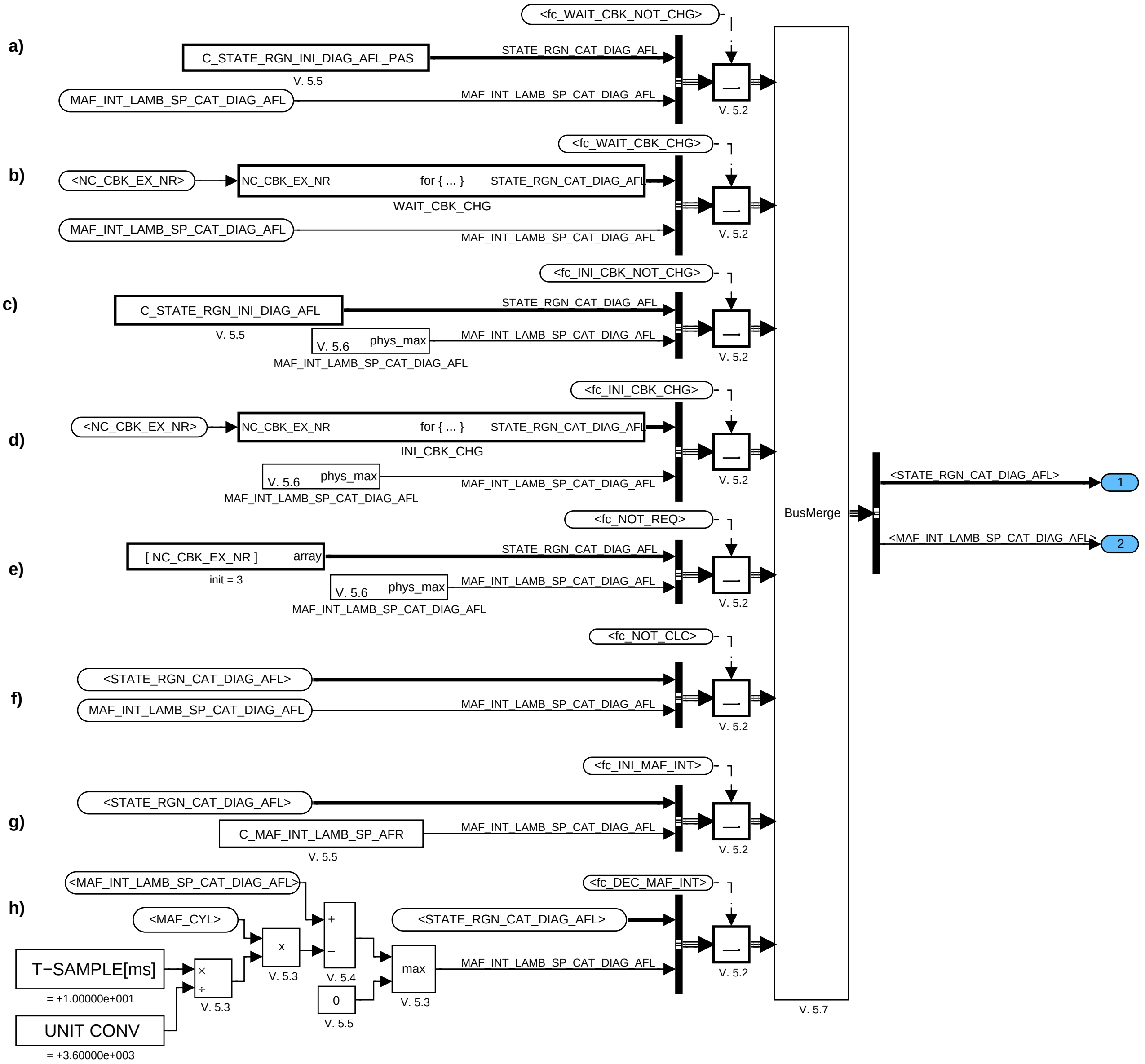


Figure 189:

18.18.2.2.3 Multiple bank system

18.18.2.2.3.1 Bank specific calculations - Lambda set point

The lambda set point for the precondition phase is active directly after Nox Trap regeneration. During this regeneration "WAIT" is active. After precondition the lambda set point for the diagnosed bank is calculated ("DIAG"), for the parked bank ("PARK") the calculations are done inside CLC_LAMB_PARK. When the request for the diagnosis isn't active, PASSIVE is called

18.18.2.2.3.1.1 Check of the regeneration and diagnosis requests

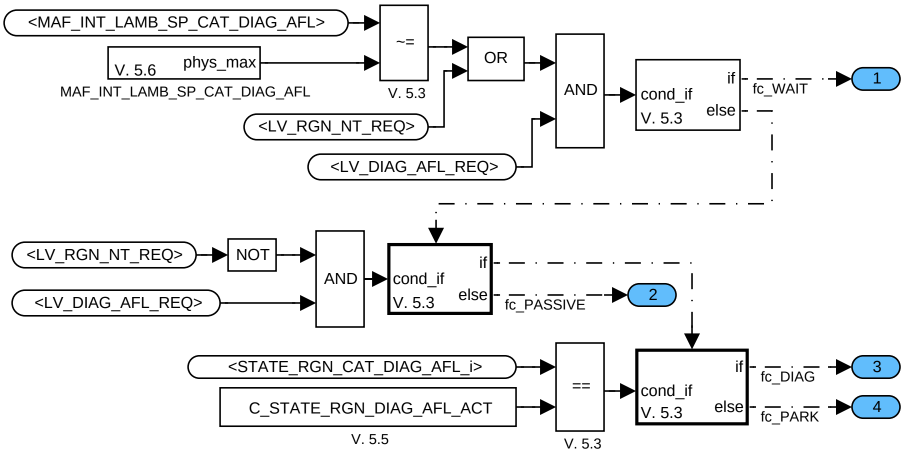


Figure 190:

18.18.2.2.3.1.2 Calculation of the lambda set point

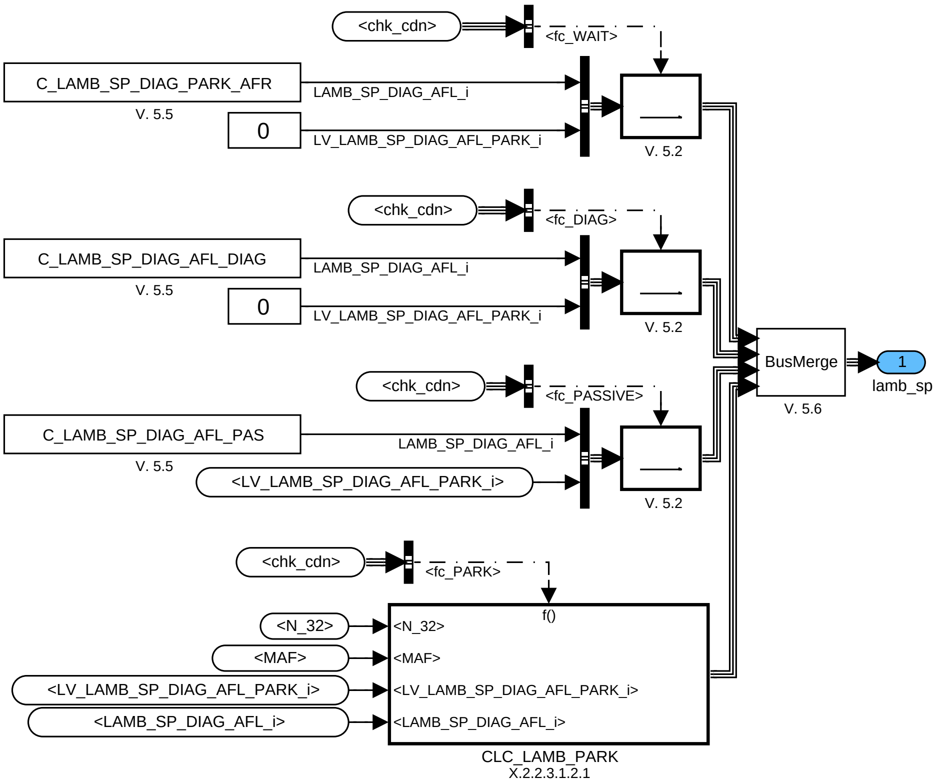


Figure 191:

18.18.2.2.3.1.2.1 Calculation of the lambda set point for the park state (CLC_LAMB_PARK)

18.18.2.2.3.1.2.1.1 Checking of the actual lambda value

When the lambda is bigger as the minimum then the incrementing is called, when the lambda value is smaller as the maximum then the decrementing is called

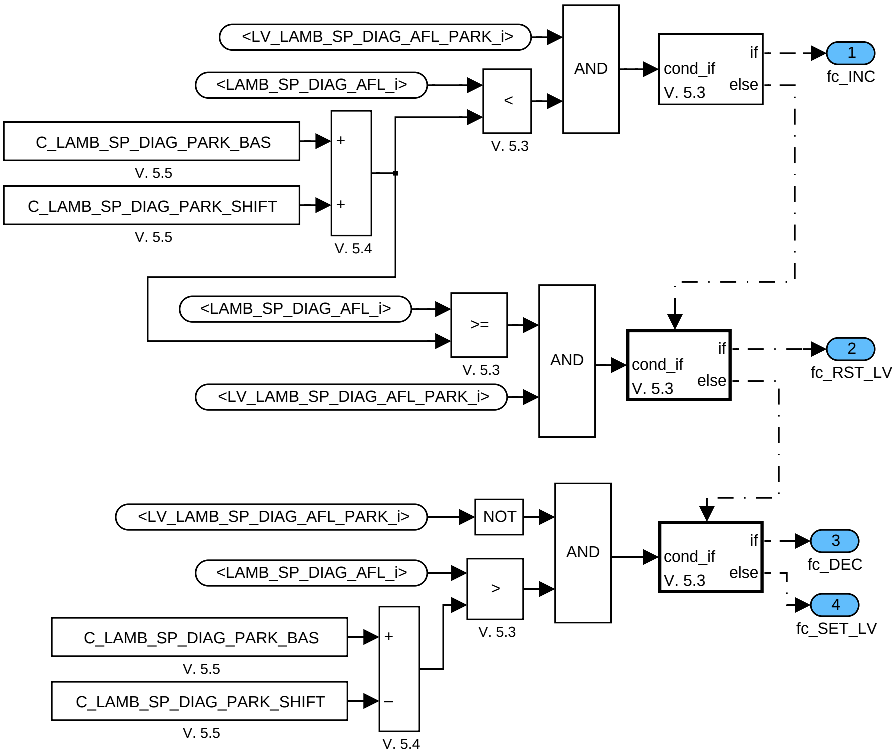


Figure 192:

18.18.2.2.3.1.2.1.2 INC: Incrementing of the lambda set point value

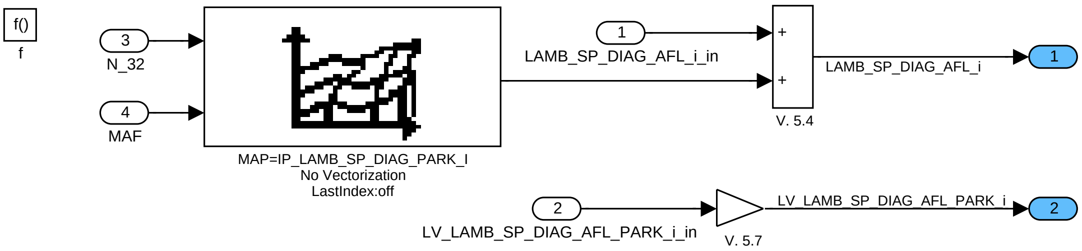


Figure 193:

18.18.2.2.3.1.2.1.3 RST_LV: Reset of the indicator flag for the lambda set point at maximum

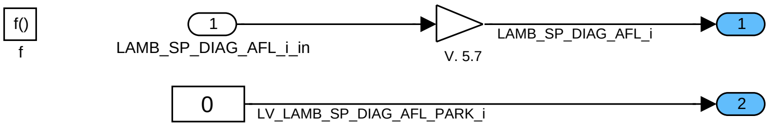


Figure 194:

18.18.2.2.3.1.2.1.4 DEC: Decrementing of the lambda set point value

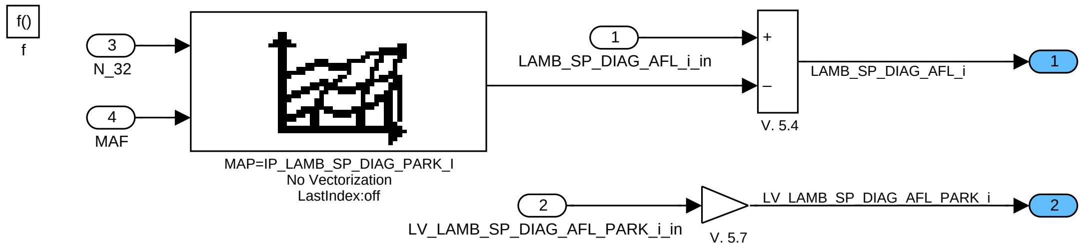


Figure 195:

18.18.2.2.3.1.2.1.5 SET_LV: Set of the indicator flag for the lambda set point at minimum

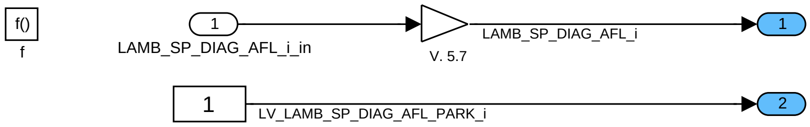


Figure 196:

18.18.2.2.4 Calculation of the old variables

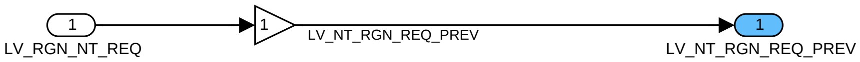


Figure 197:

18.18.2.3 PASSIVE STATE (LAMB_SP_NOT_CLC)

18.18.2.3.1 Calculation of the regeneration state for the diagnosis

In case the extended lambda setpoint is not required for the diagnosis the regeneration state is also not calculated, but only copied from STATE_RGN

Figure 198:

18.19 Module OSC based TWC diagnosis for DI-engines at lean operation - OSC values calculation

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_O2L_CLC_LOAD [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Flag for calculation of O2L during load phase					
LV_O2L_CLC_PARK [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Flag for calculation of O2L during park phase					
LV_O2L_CLC_RGN_H_AMPL [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Flag for calculation of O2L during high regeneration phase					
LV_O2L_CLC_RGN_L_AMPL [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Flag for calculation of O2L during low regeneration phase					
O2L_INT_LOAD_PHA [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L integral in the load phase					
O2L_INT_PARK_PHA [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L integral in the park phase					
O2L_INT_RGN_H_AMPL_PHA [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L integral in the high amplitude regeneration phase					
O2L_INT_RGN_L_AMPL_PHA [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L integral in the low amplitude regeneration phase					
O2L_INT_THD_TMP_DIAG_AFL [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Corrected threshold for quit of the O2L integral calculation					
O2L_LOAD_PHA [NC_CBK_EX_NR]	O/V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L calculated during load phase					
O2L_PARK_PHA [NC_CBK_EX_NR]	O/V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L calculated during park phase					
O2L_RGN_H_AMPL_PHA [NC_CBK_EX_NR]	O/V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L calculated during high amplitude regeneration phase					
O2L_RGN_L_AMPL_PHA [NC_CBK_EX_NR]	O/V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L calculated during low amplitude regeneration phase					
STATE_CAT_DIAG_AFL [NC_CBK_EX_NR]	O/V	0 1 2 3	PASSIVE RICH_CDN LEAN_CDN WAIT	-	[-]
OSC calculation state for TWC diagnosis at lean operation					
STATE_O2L_CLC_END [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
State for O2L calculation - calculation is finished					
STATE_O2L_CLC_VLD [NC_CBK_EX_NR]	O/V	0... FFH	0... 255	1	[-]
Validation state for O2L integral calculation					
STATE_RGN_CAT_DIAG_AFL_OLD [NC_CBK_EX_NR]	-	0... 3H	0... 3	1	[-]
Old value of state for calculation of lean TWC diagnosis due state of system regeneration					
T_AFL_CDN_OUT [NC_CBK_EX_NR]	V	0... FFFFH	0... 655.35	0.01	[s]
Timer for exit from state LEAN_CDN					

Input Data:

LV_O2L_CLC_ACT_CDN [NC_CBK_EX_NR]	STATE_RGN_CAT_DIAG_AFL [NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_ST_END
LV_NT_ACT	VLS_DOWN [NC_CBK_EX_NR]	LAMB_LS_UP [NC_CBK_EX_NR]	MAF_FG_CYL
N_32	LV_S_ACT	LV_HOM_AFL_ACT	TEMP_CAT_STAT_MDL [NC_CBK_EX_NR]
TEMP_CAT_DYN_MDL [NC_CBK_EX_NR]	MAF_CYL	VLS_NOX_SENS [NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
LV_T_AFL_MIN	T_AFL	TNT_MDL_MV_SNG [NC_NT_NR]	NC_NT_NR

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_LAMB_O2L_INT_CLC	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Threshold of linear upstream lambda signal for calculation of O2L integral					
C_O2L_INT_THD_TMP_DIAG_AFL	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Threshold to quit of the O2L integral calculation					
C_STATE_EX_CONF	1	1... 4H	1... 4	1	[-]
Exhaust gas system configuration					
C_STATE_O2L_CLC_ENA	1	0... FFH	0... 255	1	[-]
State to allow of calculation of O2L integrals					
C_STATE_RGN_CAT_DIAG_AFL	1	0... 8H	0... 8	1	[-]
Regenerations state for calculation PARK and LOAD at STATE_EX_CONF_2					
C_STATE_RGN_CAT_DIAG_AFR	1	0... 8H	0... 8	1	[-]
Additional regenerations state for calculation H_AMPL phase					
C_STATE_SENS_DOWN_MISS_CBK	1	1... 4H	1... 4	1	[-]
Indicator for bank without downstream lambda sensor					
C_T_AFL_CDN_OUT	1	0... FFFFH	0... 655.35	0.01	[s]
Maximal time to exit from LEAN_CDN state					
C_VLS_NOX_O2L_INT_CLC	1	0... 578H	-200... 1200	1	[mV]
Threshold of binary NOx-Sensor lambda signal for calculation of O2L integral					
C_VLS_O2L_INT_CLC	1	0... 3FFH	0... 4.9951171875	4.88281e-3	[V]
Threshold of binary downstream lambda signal for calculation of O2L integral					
IP_FAC_LAMB_DIAG_AFL	8	0... FFH	0... 3.984375	0.015625	[-]
LDP_LAMB_LS_UP_IP_FAC_DIAG_AFL	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
O2L correction factor depending on lambda					
IP_FAC_OPP_DIAG_AFL	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDP_N_32_IP_FAC_DIAG_AFL	8	0... FFH	0... 8160	32	[rpm]
LDP_MAF_CYL_IP_FAC_DIAG_AFL	8	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
O2L correction factor depending on operating point					
IP_FAC_OPP_DIAG_AFL_SUM	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDP_N_32_IP_FAC_DIAG_AFL	8	0... FFH	0... 8160	32	[rpm]
LDP_MAF_CYL_IP_FAC_DIAG_AFL	8	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Correction factor of the O2L calculated in PARK phase depending on operating point conditions					
IP_FAC_T_AFL_DIAG_AFL	8	0... FFH	0... 15.9375	0.0625	[-]
LDP_T_AFL_IP_FAC_DIAG_AFL	8	0... FFFFH	0... 1310.7	0.02	[s]
Correction factor for calculation of O2L value depending on vehicle speed					
IP_FAC_TEMP_DIAG_AFL	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDP_TEMP_STAT_IP_FAC_DIAG_AFL	8	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
LDP_TEMP_DIF_IP_FAC_DIAG_AFL	8	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
O2L correction factor depending on TWC temperature					

IP_FAC_TEMP_DIAG_AFL_SUM	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDP_TEMP_STAT_IP_FAC_DIAG_AFL	8	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
LDP_TNT_MDL_IP_FAC_DIAG_AFL	8	0... FFFFH	0... 1023.984375	0.015625	[°C]
O2L correction factor depending on TWC and NT temperatures					

Export Actions:

ACTION_EGTR_ClcValidCheck(IN <LV_O2L_CLC_OK_i_tmp>, **IN** <i>, **IN** <O2L_old_i>, **IN** <LV_PHA_i>, **OUT** <O2L_i>, **OUT** <STATE_O2L_CLC_VLD_Bx_i>, **INOUT** <O2L_INT_i>)
Local Action: Check of plausibility of O2L integral calculation and calcualtion of O2L end value
ACTION_EGTR_O2LIntegralCalc(IN <i>, **IN** <O2L_INT_old_i>, **OUT** <O2L_INT_i>)
Local Action: Calculation of O2L Integral

Description for Actions

ACTION_EGTR_ClcValidCheck(IN <LV_O2L_CLC_OK_i_tmp>, **IN** <i>, **IN** <O2L_old_i>, **IN** <LV_PHA_i>, **OUT** <O2L_i>, **OUT** <STATE_O2L_CLC_VLD_Bx_i>, **INOUT** <O2L_INT_i>)

Local Action: Check of plausibility of O2L integral calculation and calcualtion of O2L end value

Parameter	Type	Hex.Limits	Phys.Limits	Resol.	Unit
LV_O2L_CLC_OK_i_tmp	IN	0... 1H	0... 1	1	[-]
Calculation of O2L integral regarding upstream and downstream lambda passed					
i	IN	1... 4H	1... 4	1	[-]
Bank specific index					
O2L_old_i	IN	0... FFFFH	0... 2.61684896	39.9306e-6	[g]
Oxygen loading old value					
LV_PHA_i	IN	0... 1H	0... 1	1	[-]
Indicator of the park phase					
O2L_i	OUT	0... FFFFH	0... 2.61684896	39.9306e-6	[g]
Calculated oxygen loading of TWC					
STATE_O2L_CLC_VLD_Bx_i	OUT	0... 1H	0... 1	1	[-]
Read bit of STATE_O2L_CLC_VLD					
O2L_INT_i	INOUT	0... FFFFH	0... 2.61684896	39.9306e-6	[g]
Integral of O2L					

ACTION_EGTR_O2LIntegralCalc(IN <i>, **IN** <O2L_INT_old_i>, **OUT** <O2L_INT_i>)

Local Action: Calculation of O2L Integral

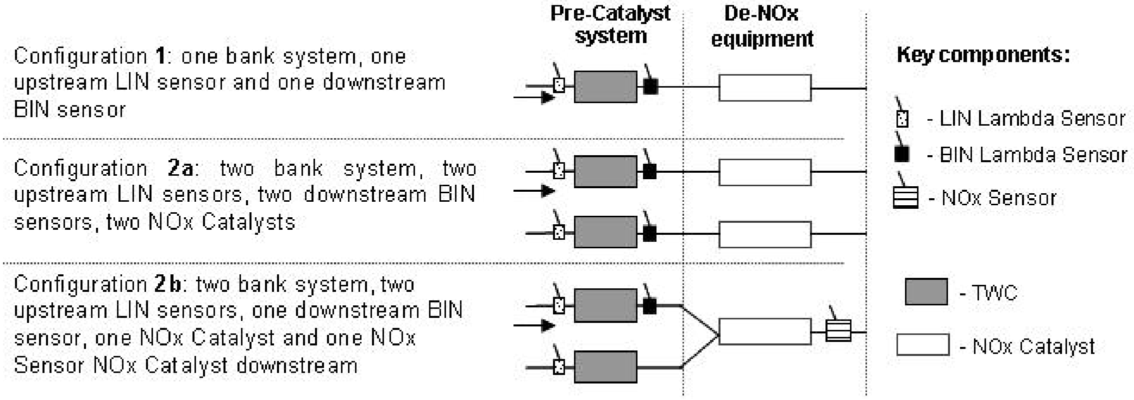
Parameter	Type	Hex.Limits	Phys.Limits	Resol.	Unit
i	IN	1... 4H	1... 4	1	[-]
Bank specific index					
O2L_INT_old_i	IN	0... FFFFH	0... 2.61684896	39.9306e-6	[g]
Integral of O2L - old value					
O2L_INT_i	OUT	0... FFFFH	0... 2.61684896	39.9306e-6	[g]
Integral of O2L					

General Information

For calculation of O2L the fresh gas per cylinder, linear lambda signal (TWC upstream), binary lambda signal (TWC downstream or NOx-Trap downstream) are used. The O2L removal is calculated when the linear lambda signal upstream shows the rich mixture (less then 1) and the binary lambda signal downstream shows the lean mixture (less then 500 [mV]). The O2L addition is calculated when the linear lambda signal upstream shows lean mixture (greater then 1) and the binary lambda signal downstream shows the rich mixture (greater then 500 [mV])

If two separate exhaust bank system are concerned, the diagnosis is relevant for two exhaust gas configurations switched with C_STATE_EX_CONF:
CONF_1: Exhaust gas configuration equipped with DeNOx technology, with two LIN-Sensors upstream TWC, two BIN-Sensors downstream TWC, two NOx Traps (NT) and one NOx-Sensor downstream NT (Picture 1, configuration 2a)
CONF_2: Exhaust gas configuration equipped with DeNOx technology, with two LIN-Sensors upstream TWC, one BIN-Sensor downstream TWC, one NOx Traps (NT) and one NOx-Sensor downstream NT (Picture 1, configuration 2b)

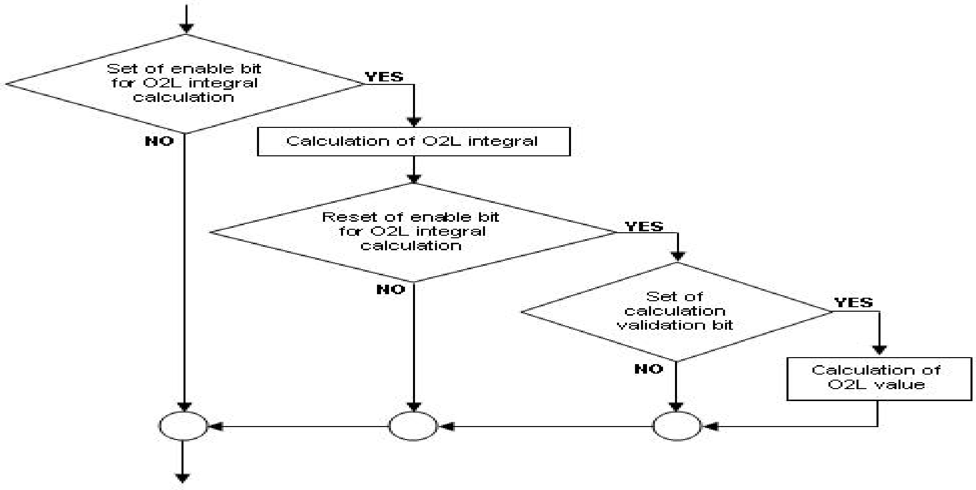
Picture 1: Exhaust gas configurations



The O2L-values are calculated in four cases:
Case 1: Regeneration with high amplitude
Case 2: Regeneration with low amplitude
Case 3: Park phase (for CONF_1 calculated O2L concerns TWC, for CONF_2 calculated O2L concerns sum of TWC and NT)
Case 4: Load phase

For each case for the calculation of O2L is one validation bit in STATE_O2L_CLC_VLD[NC_CBK_EX_NR] defined:
Bit 0 (STATE_O2L_CLC_VLD_B0): Case 1
Bit 1 (STATE_O2L_CLC_VLD_B1): Case 2
Bit 2 (STATE_O2L_CLC_VLD_B2): Case 3
Bit 3 (STATE_O2L_CLC_VLD_B3): Case 4

Picture 2: Signal flow for calculation of O2L values:



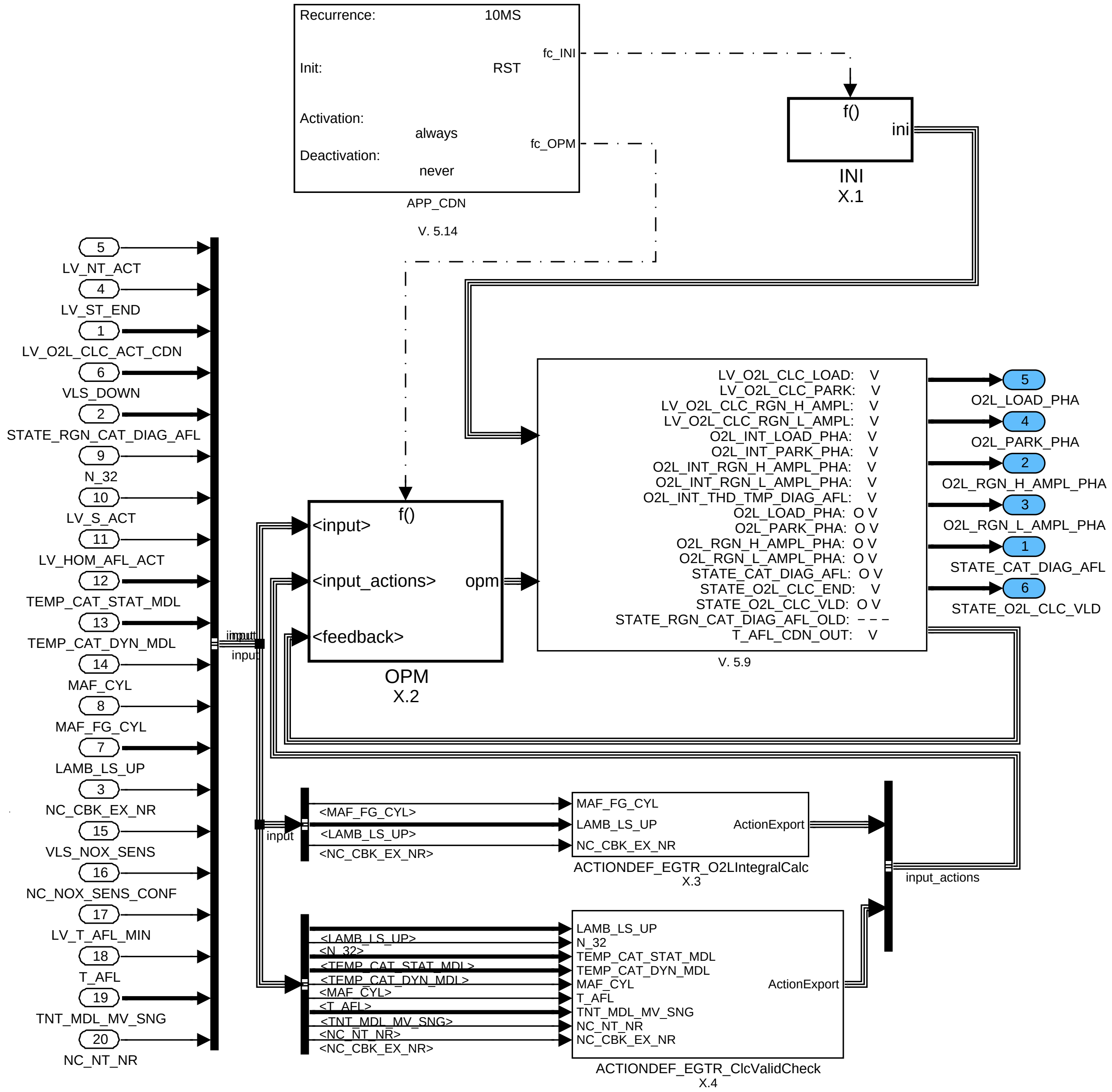
If two separate exhaust bank systems are concerned, then:
i = 1, for cylinder bank 1
i = 2, for cylinder bank 2
Otherwise:
i = 1

NC_NOX_SENS_CONF is the number of NOx-Sensors. Therefore
NC_NOX_SENS_CONF = 0, for no NOx-Sensor
NC_NOX_SENS_CONF = 1 (n = 1), for one NOx-Sensor
NC_NOX_SENS_CONF = 2 (n = i), for two NOx-Sensors

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.2.5 26−Feb−2007

Figure 199:

18.19.1 Initialization

O2L_INT_THD_TMP_DIAG_AFL=C_O2L_INT_THD_TMP_DIAG_AFL
All other variables are initialized with "0"

18.19.2 Formula section

All calculations are bank specific
Additional information:
"i" is an indicator for current bank within of for loop depending on NC_CBK_EX_NR.
"n" is an indicator for signals of NOx-Sensor. Therefore:
Case 1: If NC_CBK_EX_NR=1 then n=1 (in system the number of NOx-Sensors can not be greater then number of exhaust bank).
Case 2a: If NC_CBK_EX_NR=2 and NC_NOX_SENS_CONF=1 then n=1

Case 2b: If NC_CBK_EX_NR=2 and NC_NOX_SENS_CONF=2 then n=i

18.19.2.1 Check of the activation condition

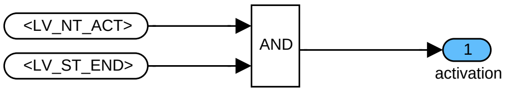


Figure 200:

18.19.2.2 Activation condition fulfilled

18.19.2.2.1 Overview of the function structure

At first, checking of transition condition and generation of function call for the active state of state machine "STATE_CAT_DIAG_AFL" are done. When the transition conditions are not fulfilled, the same state as in previous cycle is executed. In one calculation cycle only one state can be executed. The old value of STATE_RGN (for each bank separate) is also calculated after calculating of STATE_CAT_DIAG_AFL

18.19.2.2.1.1 STATE FLOW

Calculation of STATE_CAT_DIAG_AFL and generation of the function call for execution of one of it states

18.19.2.2.1.1.1 STATE_CAT_DIAG_AFL

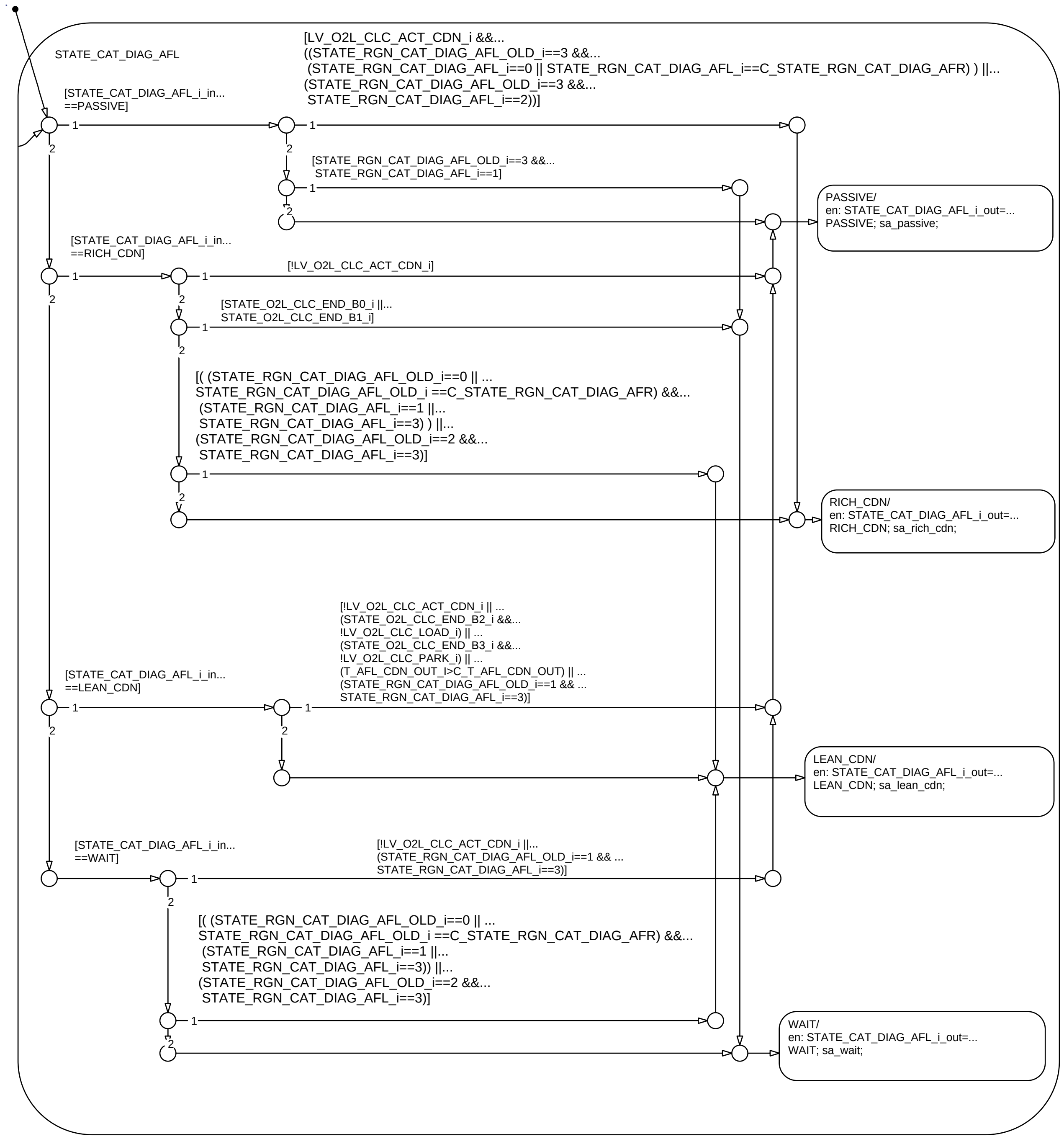


Figure 201:

18.19.2.2.1.2 EXECUTION OF THE STATES OF "STATE_CAT_DIAG_AFL"

STATE_CAT_DIAG_AFL = PASSIVE: Calculation of OSC is not active
STATE_CAT_DIAG_AFL = RICH_CDN: Calculation of OSC is done under rich operation conditions, O2L removal is calculated
STATE_CAT_DIAG_AFL = LEAN_CDN: Calculation of OSC is done under lean operation conditions, O2L addition is calculated
STATE_CAT_DIAG_AFL = WAIT: OSC calculation in RICH conditions is done, the system is waiting for change from rich to lean conditions

18.19.2.2.1.2.1 STATE ACTION: STATE_CAT_DIAG_AFL="PASSIVE"

18.19.2.2.1.2.1.1 Calculations

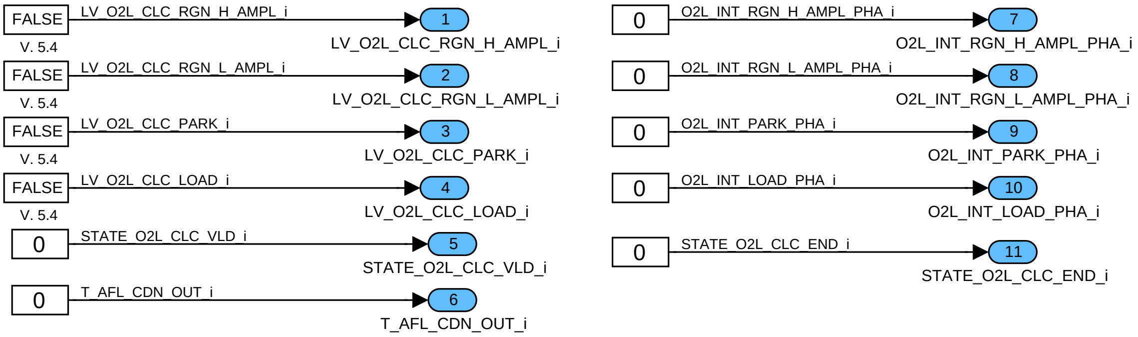


Figure 202:

18.19.2.2.1.2.2 STATE ACTION: STATE_CAT_DIAG_AFL="RICH_CDN"

In this states the O2L is calculated in one of two cases: either during high regeneration phase or during low regeneration phase.
For every of these phases are calculated separately:
a) Activation flag
b) O2L removal

For both phases:
a) Validation bits of O2L is calculated (BIT 0 for high amplitude phase and BIT 1 for low amplitude phase, BIT 2 and 3 are not calculated)
b) Counter for exit from LEAN-CDN is not changed
c) Outputs of not active state are calculated (LEAN_CDN)

18.19.2.2.1.2.2.1 OSC calculation in high amplitude regeneration - activation flag

Switching between exhaust gas configurations for calculation of the activation flag is done using C_STATE_EX_CONF

18.19.2.2.1.2.2.1.1 STATE_EX_CONF_1

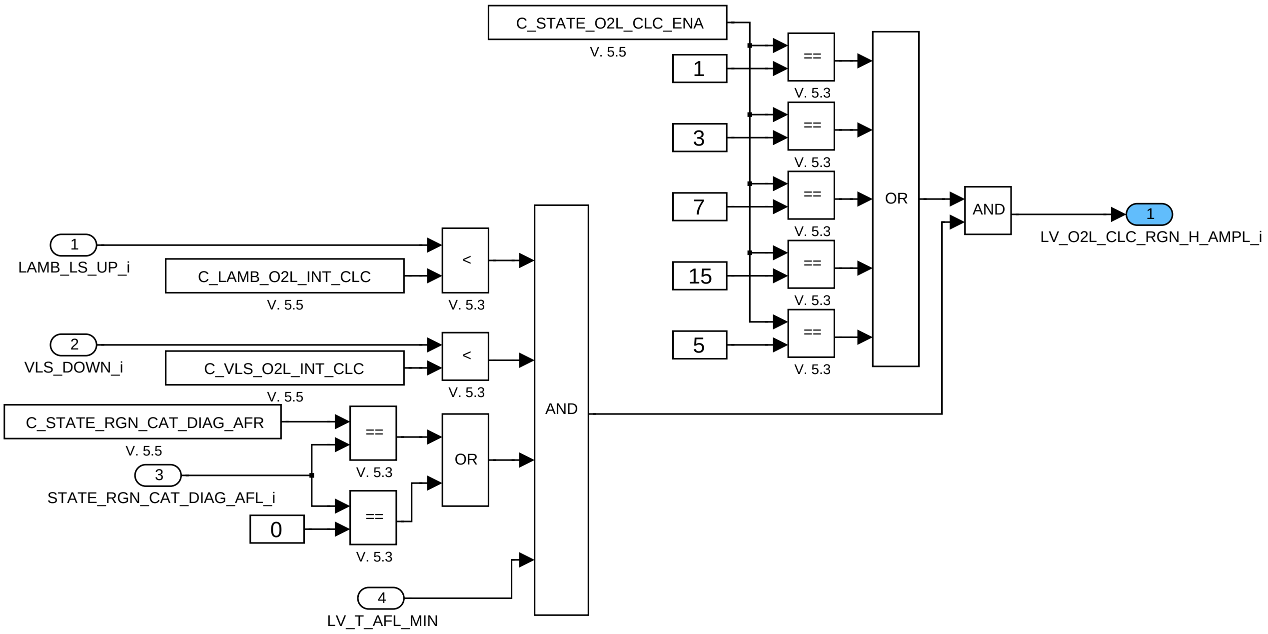


Figure 203:

18.19.2.2.1.2.2.1.2 STATE_EX_CONF_2

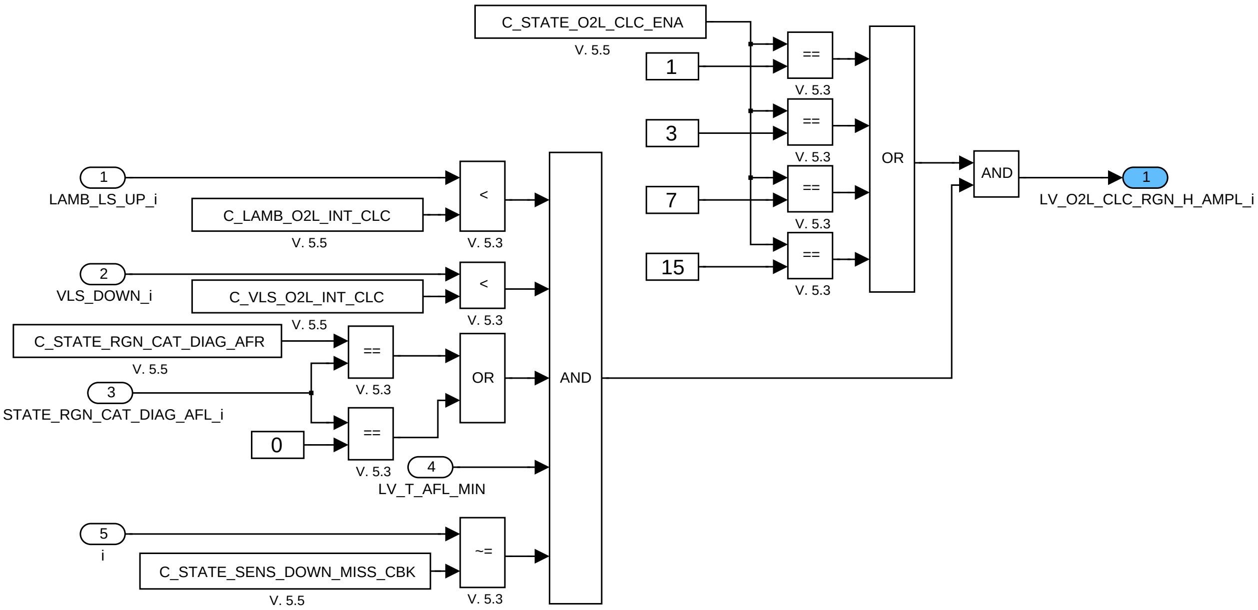


Figure 204:

18.19.2.2.1.2.2.2 OSC calculation - regeneration with high amplitude

The calculation of O2L is done in two steps:
Step 1: Calculation of O2L integral
Step 2: Calculation of O2L after calculating of O2L integral

18.19.2.2.1.2.2.2.1 Calculation of O2L-integral at regeneration with high amplitude

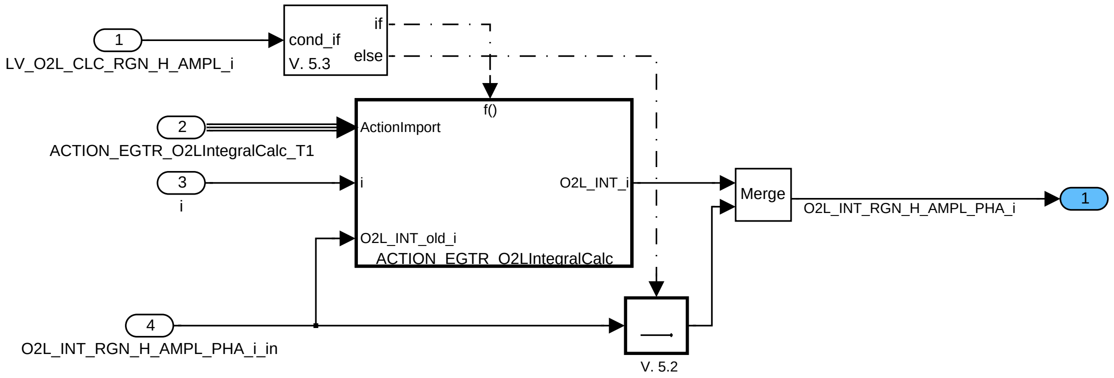


Figure 205:

18.19.2.2.1.2.2.2.2 Calculation of O2L at regeneration with high amplitude (after calculating of O2L integral)

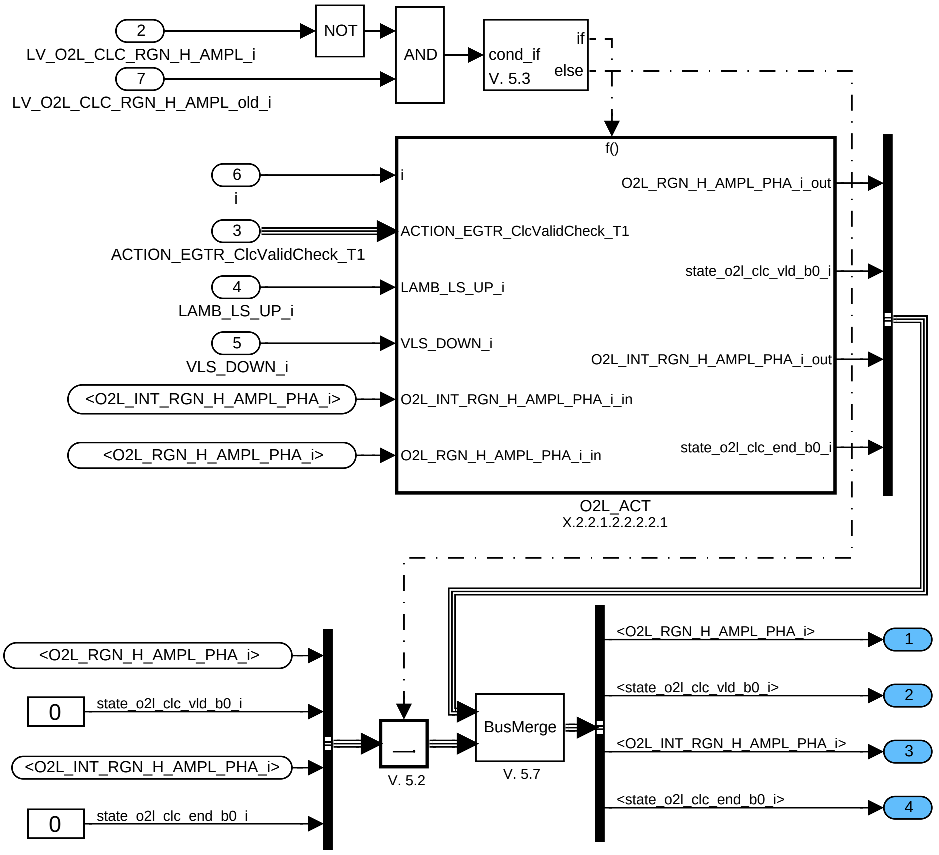


Figure 206:

18.19.2.2.1.2.2.2.2.1 Falling edge of the activation flag of integral calculation active

Check of the up and down lambda signal

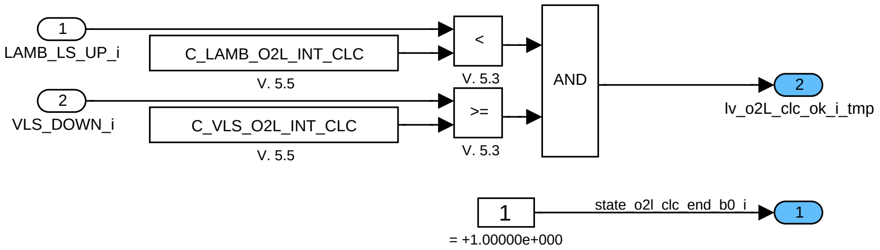


Figure 207:

Calculation of the O2L of the H_AMPL phase

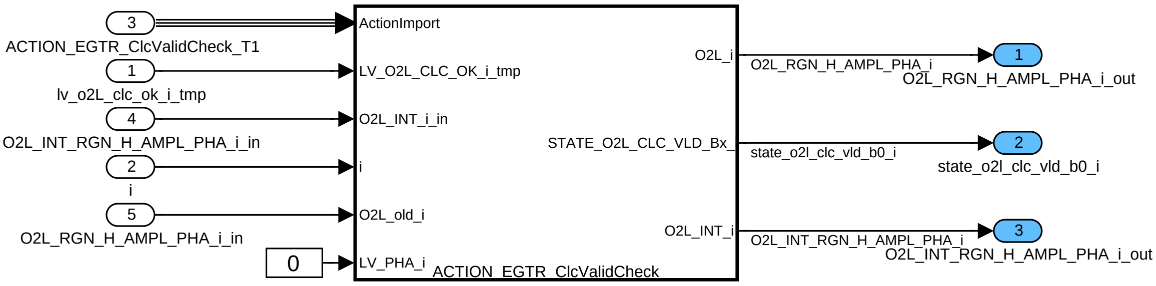


Figure 208:

18.19.2.2.1.2.2.3 OSC calculation in low amplitude regeneration - activation flag

Switching between exhaust gas configurations for calculation of the activation flag is done using C_STATE_EX_CONF

18.19.2.2.1.2.2.3.1 STATE_EX_CONF_1

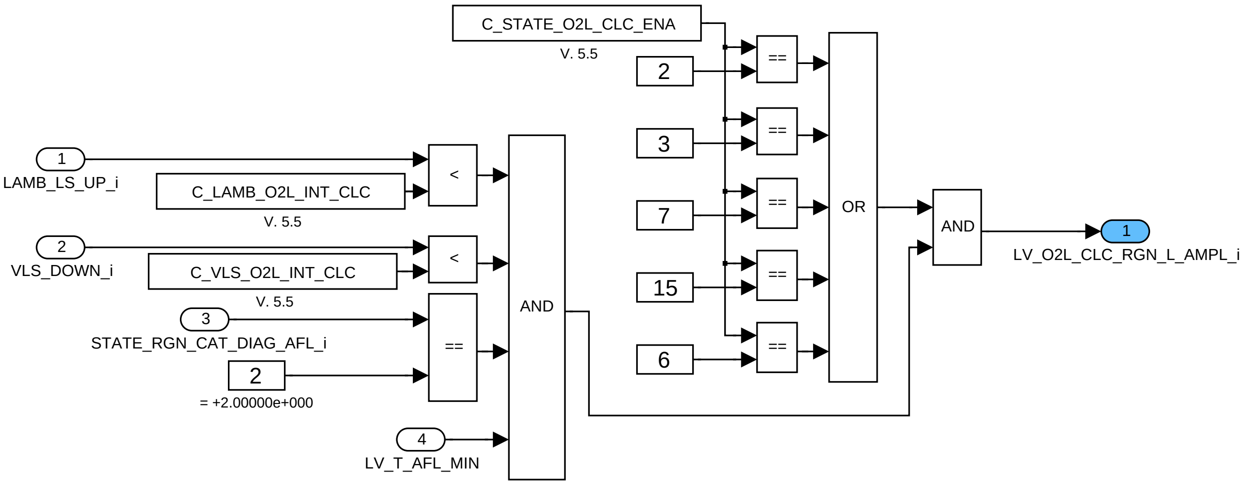


Figure 209:

18.19.2.2.1.2.2.3.2 STATE_EX_CONF_2

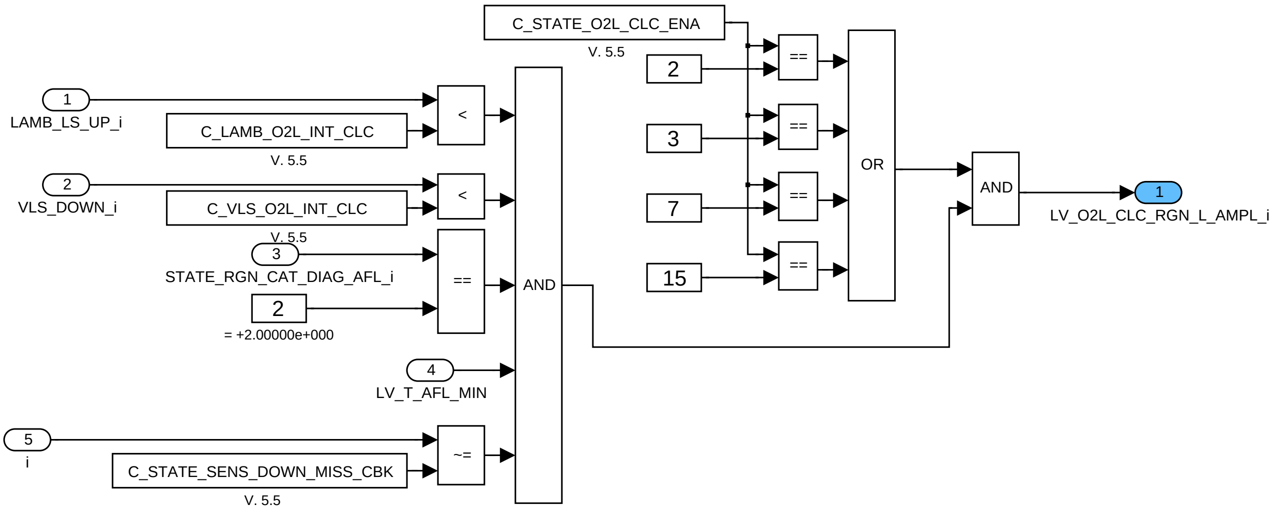


Figure 210:

18.19.2.2.1.2.2.4 OSC calculation - regeneration with low amplitude

The calculation of O2L is done in two steps:
Step 1: Calculation of O2L integral
Step 2: Calculation of O2L after calculation of O2L integral

18.19.2.2.1.2.2.4.1 Calculation of O2L-integral at regeneration with low amplitude

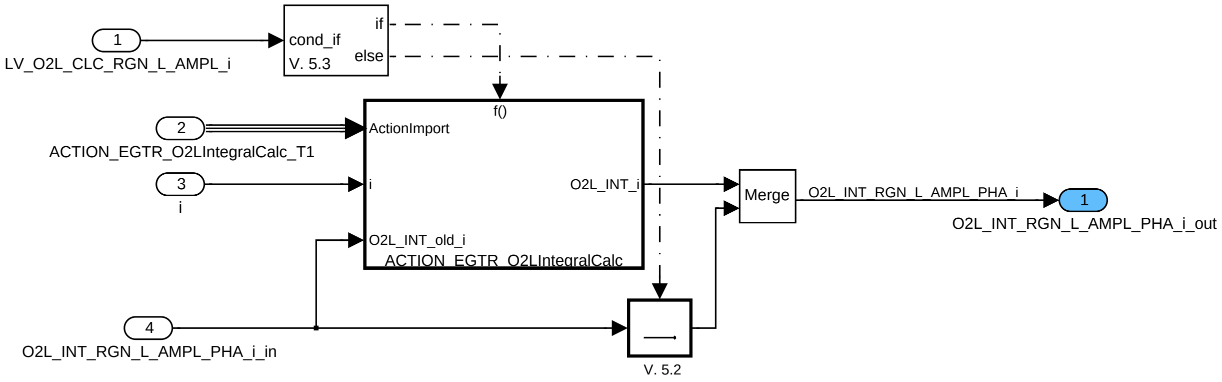


Figure 211:

18.19.2.2.1.2.2.4.2 Calculation of O2L at regeneration with low amplitude (after calculating of O2L integral)

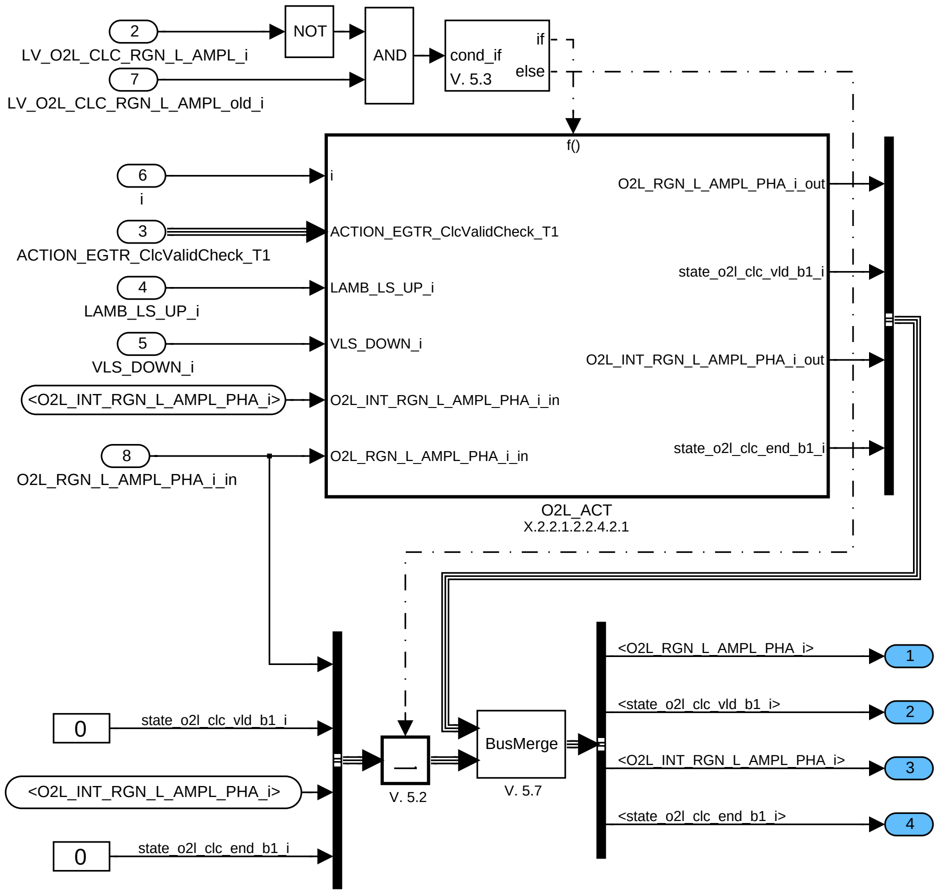


Figure 212:

18.19.2.2.1.2.2.4.2.1 Falling edge of the activation flag of integral calculation active

Check of the up and down lambda signal

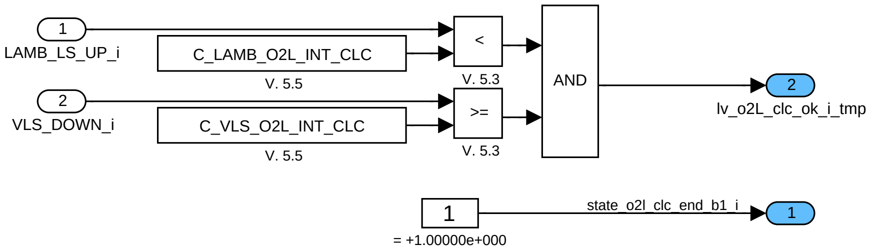


Figure 213:

Calculation of the O2L of the L_AMPL phase

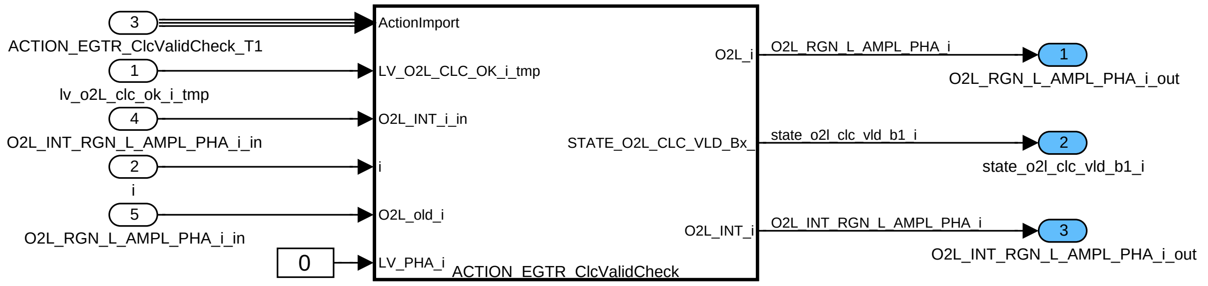


Figure 214:

18.19.2.2.1.2.2.5 Calculation of outputs of not active state

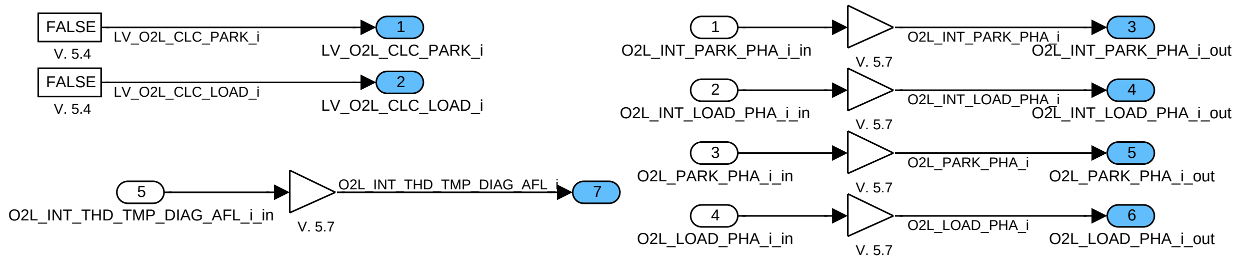


Figure 215:

18.19.2.2.1.2.3 STATE ACTION: STATE_CAT_DIAG_AFL="LEAN_CDN"

In this states the O2L is calculated in one of two cases: either during park phase or during load phase.
For every of these phases are calculated separately:
a) Activation flag
b) O2L addition

For both phases:
a) Validation bits of O2L (BIT 2 for park phase and BIT 3 for load phase, BIT 0 and BIT 1 are not calculated)
b) Counter for exit from LEAN-CDN phase
c) Outputs of not active state (RICH_CDN)

18.19.2.2.1.2.3.1 OSC calculation in park phase - activation flag

Switching between exhaust gas configurations for calculating of activation flag

18.19.2.2.1.2.3.1.1 STATE_EX_CONF_1

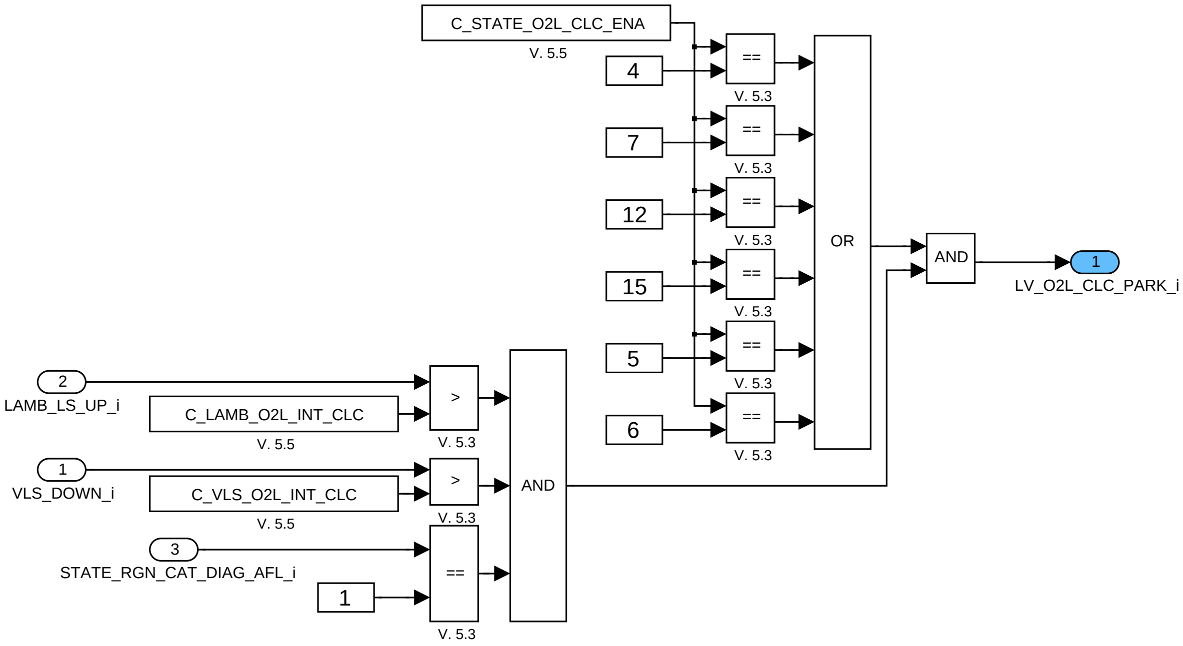


Figure 216:

18.19.2.2.1.2.3.1.2 STATE_EX_CONF_2

Apart from common conditions for calculation of the O2L integral in PARK phase, the bit 2 (saying that the calculation is finished), is checked. It shall forbid a recalculation of the O2L integral in PARK phase. Additional, to optimize the diagnosis duration, the calculation can be stopped when the O2L integral reaches defined threshold

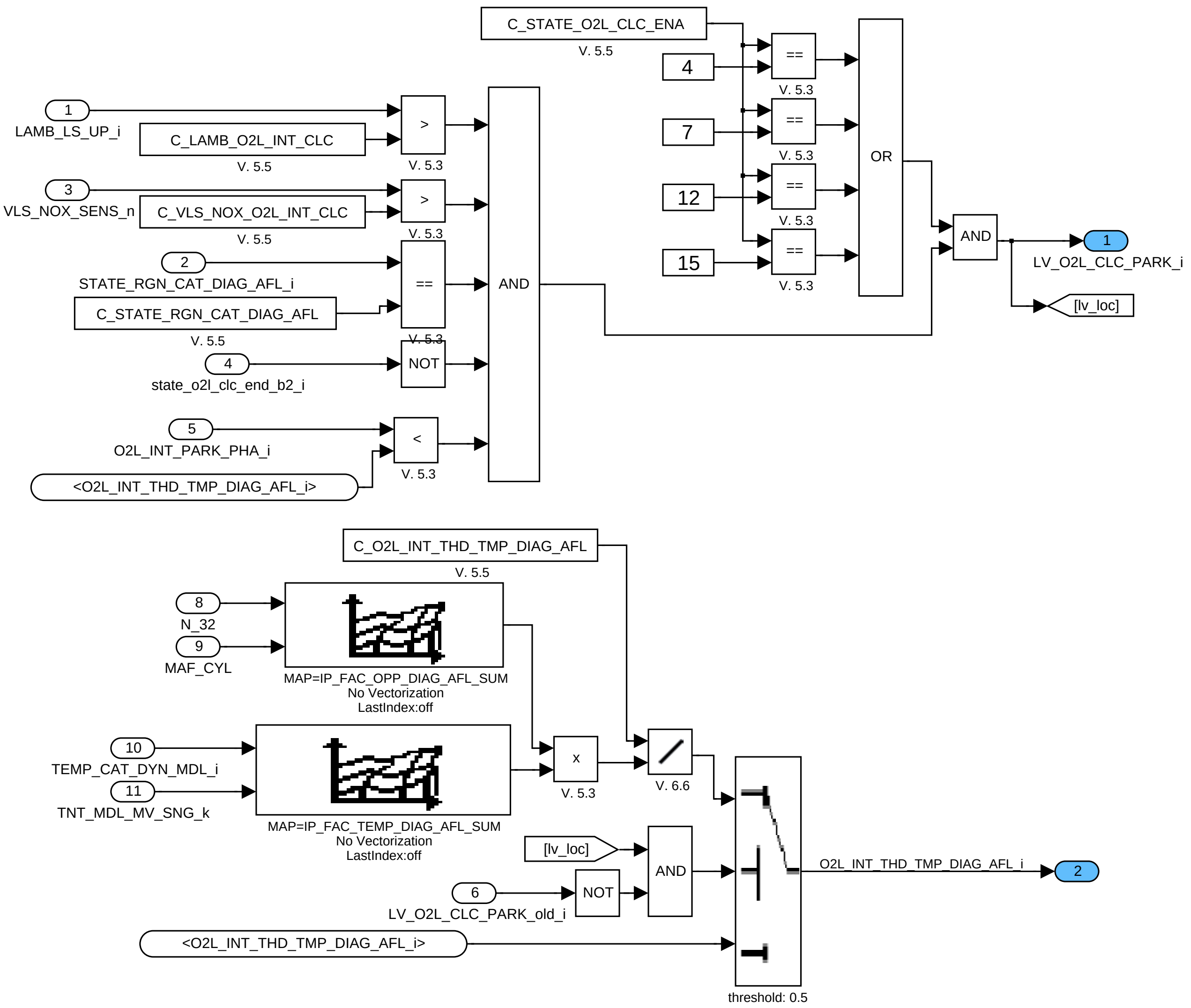


Figure 217:

18.19.2.2.1.2.3.2 OSC calculation - park phase

The calculation of O2L is done in two steps:
Step 1: Calculation of O2L integral
Step 2: Calculation of O2L after calculating of O2L integral

18.19.2.2.1.2.3.2.1 Calculation of O2L-integral at park phase

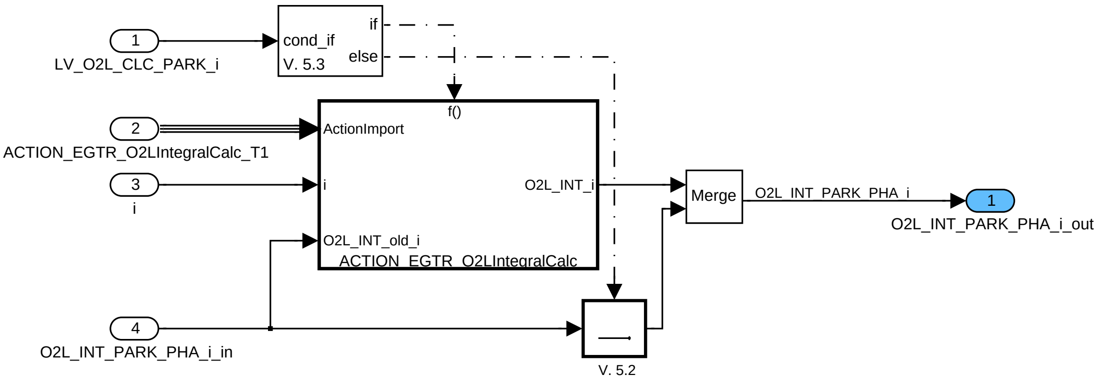


Figure 218:

18.19.2.2.1.2.3.2.2 Calculation of O2L at park phase (after calculating of O2L integral)

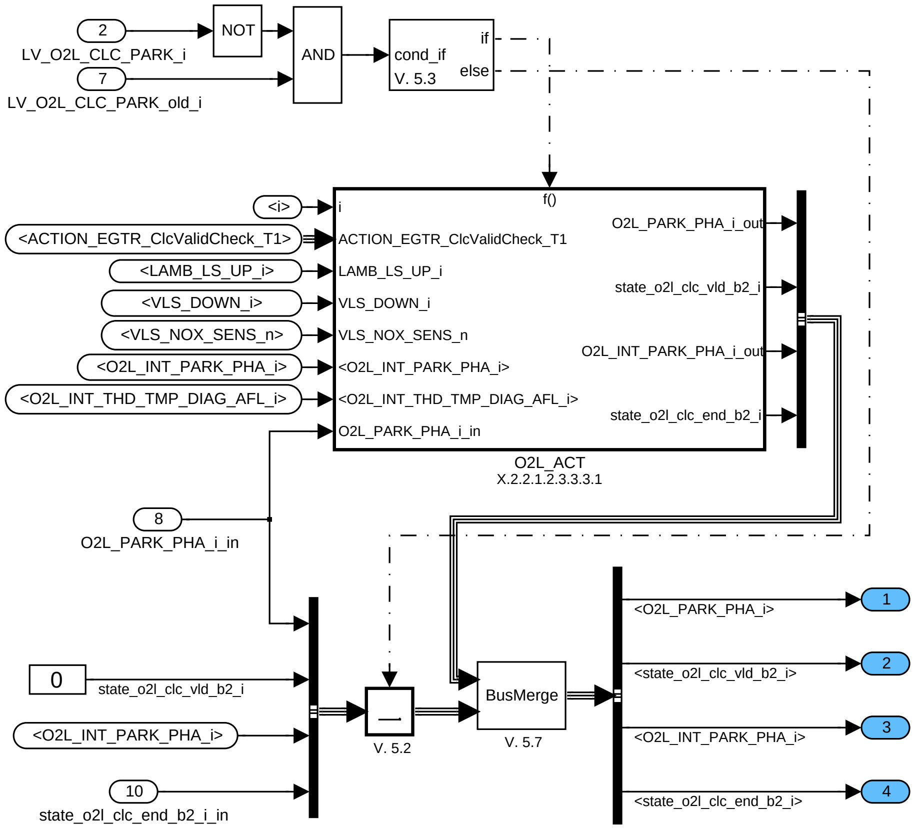


Figure 219:

18.19.2.2.1.2.3.2.2.1 Falling edge of the activation flag of integral calculation active

Check of the up and down lambda signal

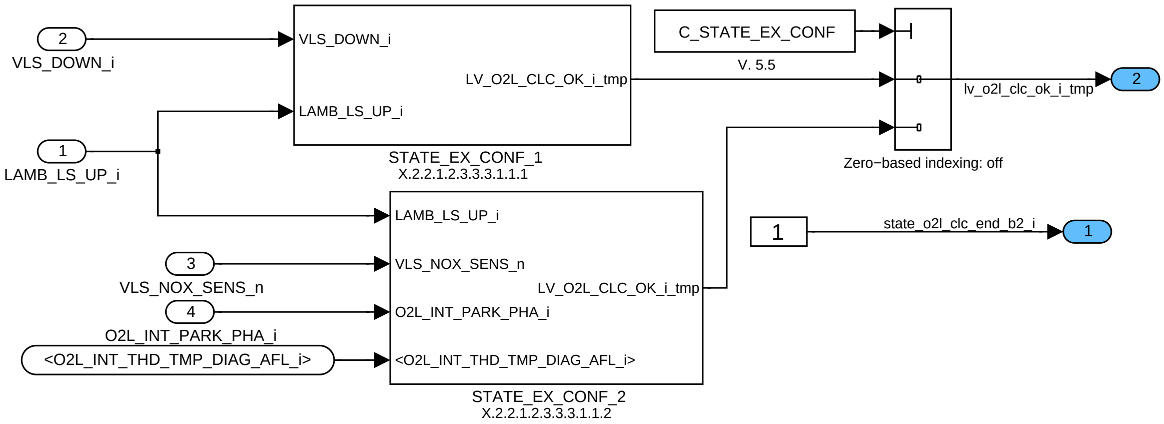


Figure 220:

STATE_EX_CONF_1

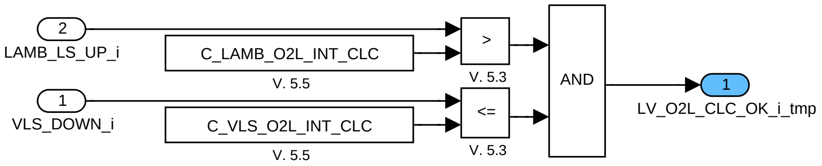


Figure 221:

STATE_EX_CONF_2

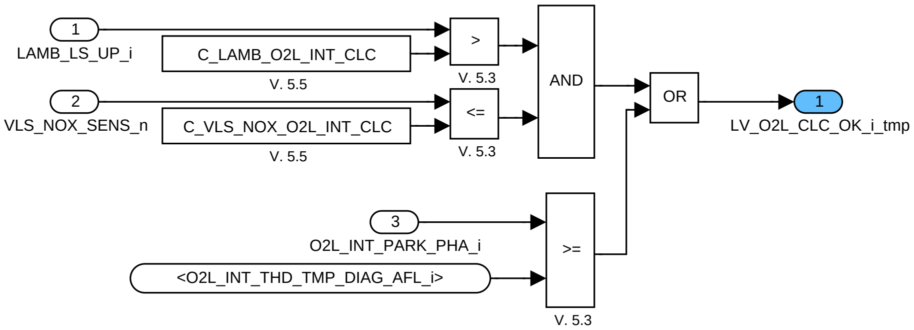


Figure 222:

Calculation of the O2L of the PARK phase

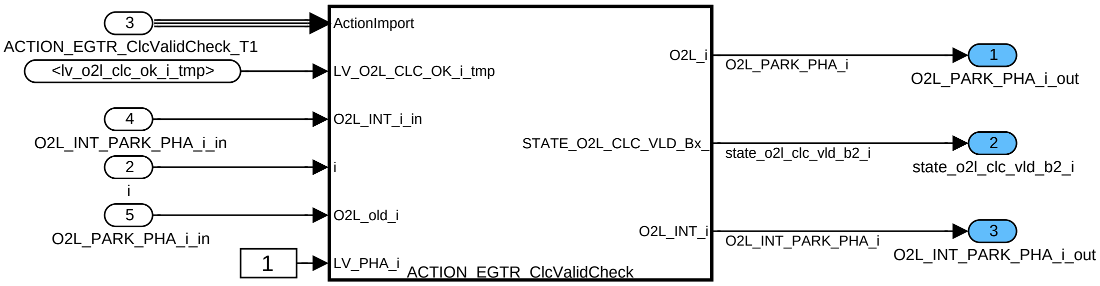


Figure 223:

18.19.2.2.1.2.3.3 OSC calculation in load phase - activation flag

Switching between exhaust gas configurations for calculation of the activation flag is done using C_STATE_EX_CONF

18.19.2.2.1.2.3.3.1 STATE_EX_CONF_1

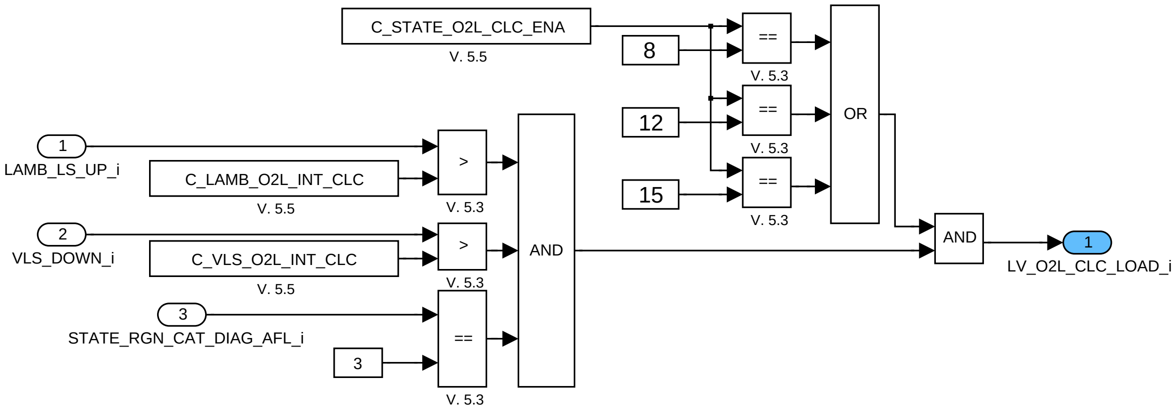


Figure 224:

18.19.2.2.1.2.3.3.2 STATE_EX_CONF_2

Apart from common conditions for calculation of the O2L integral in LOAD phase, the bit 3 (saying that the calculation is finished), is checked. It shall forbid a recalculation of the O2L integral in LOAD phase.

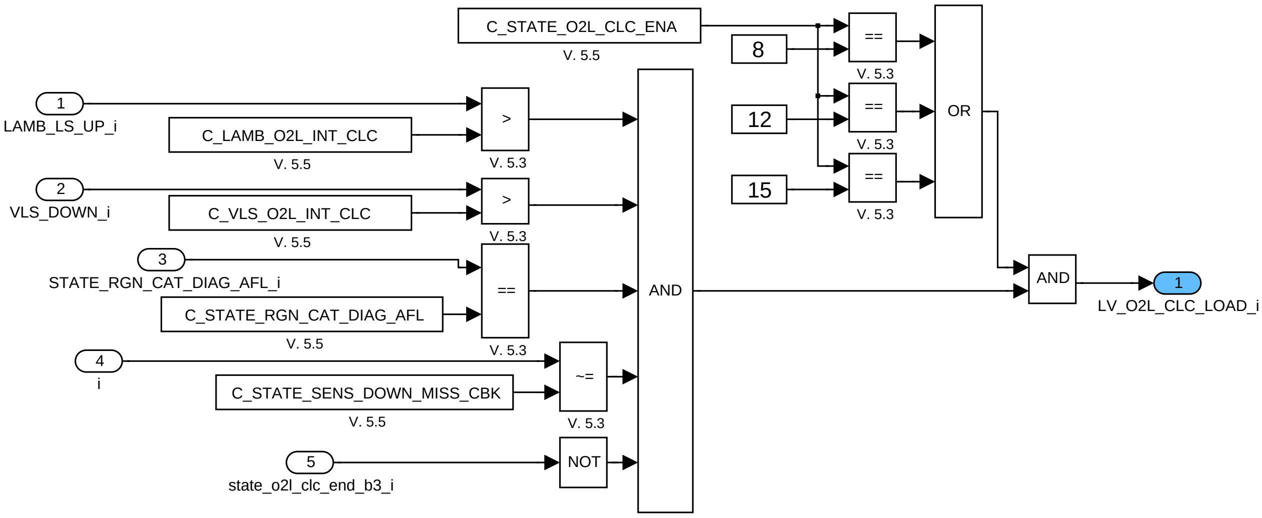


Figure 225:

18.19.2.2.1.2.3.4 OSC calculation - load phase

The calculation of O2L is done in two steps:
Step 1: Calculation of O2L integral
Step 2: Calculation of O2L after calculating of O2L integral

18.19.2.2.1.2.3.4.1 Calculation of O2L-integral at load phase

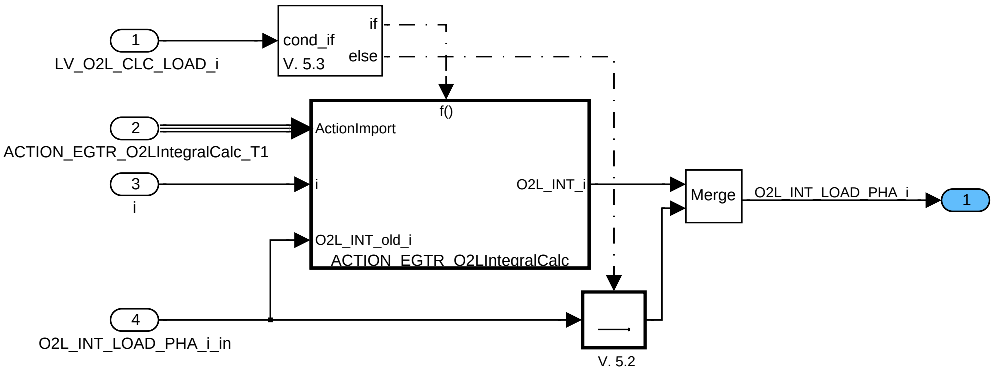


Figure 226:

18.19.2.2.1.2.3.4.2 Calculation of O2L at load phase (after calculating of O2L integral)

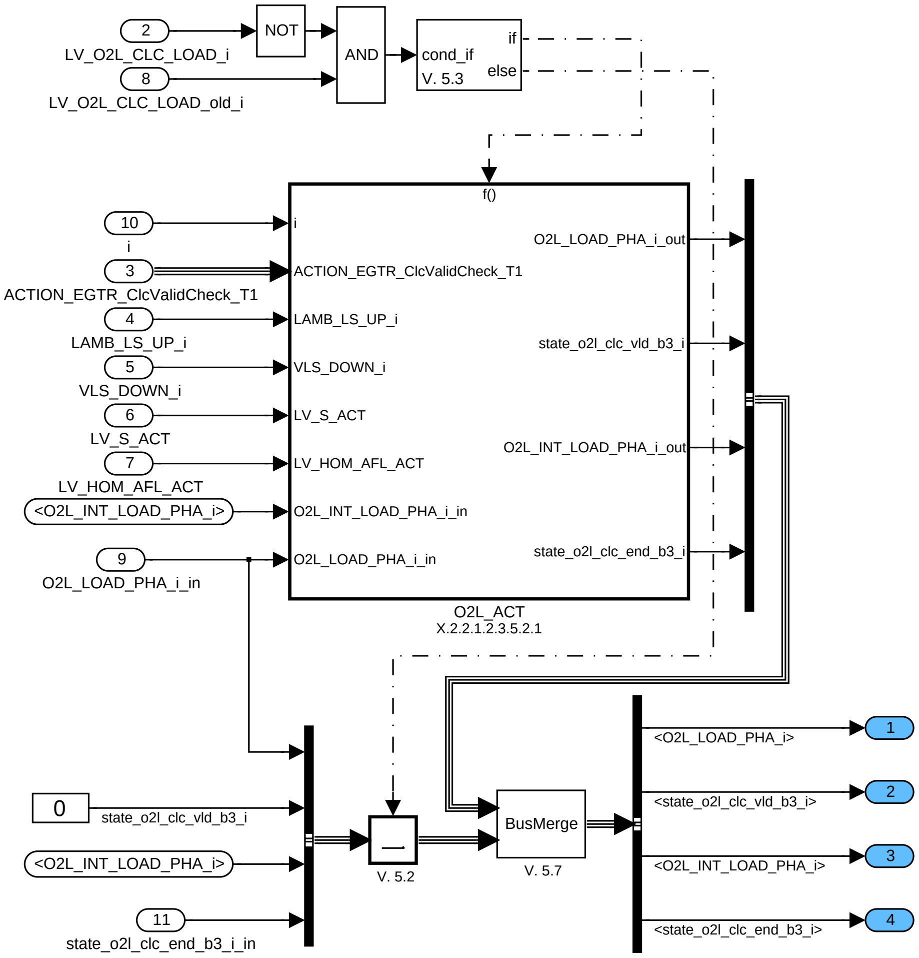


Figure 227:

18.19.2.2.1.2.3.4.2.1 Falling edge of the activation flag of integral calculation active

Check of the up and down lambda signal

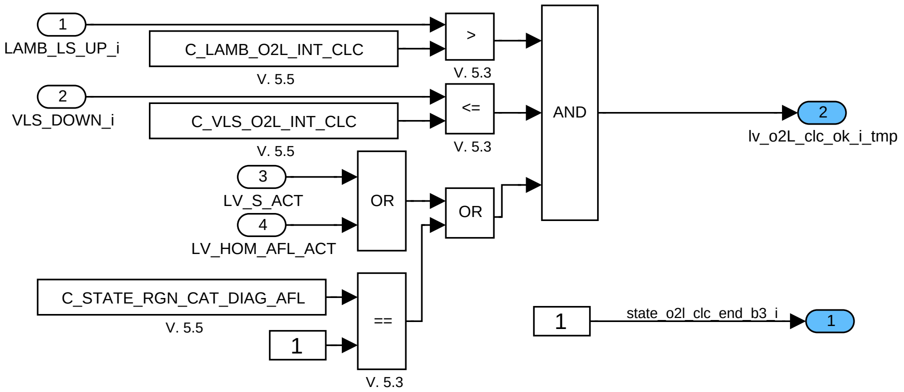


Figure 228:

Calculation of the O2L of the LOAD phase

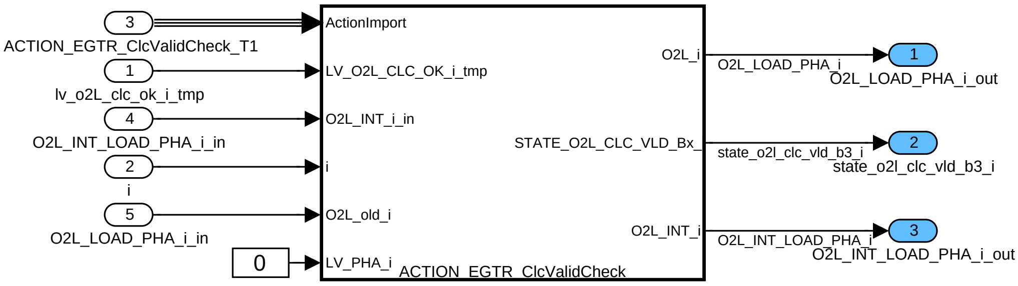


Figure 229:

18.19.2.2.1.2.3.5 Calculation of the timer to out from LEAN_CDN

The timer is calculated to make possible out from LEAN_CDN to PASSIVE if the calculation of OSC failed

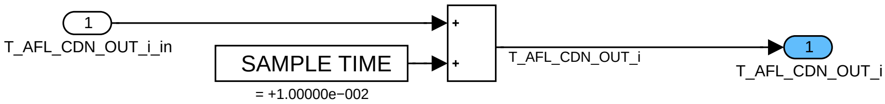


Figure 230:

18.19.2.2.1.2.3.6 Calculation of outputs of not active state

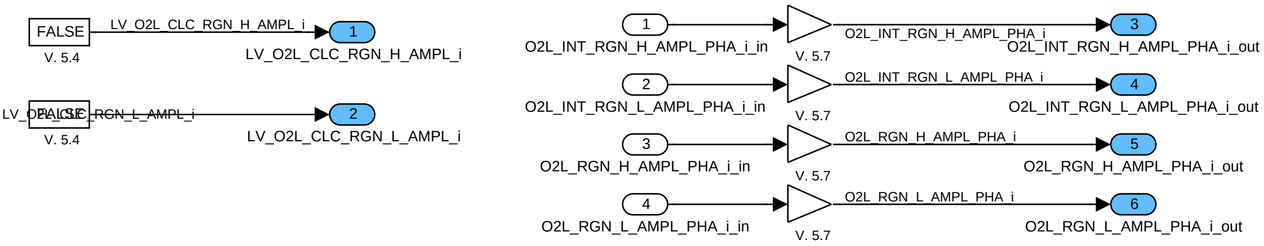


Figure 231:

18.19.2.2.1.2.4 STATE ACTION: STATE_CAT_DIAG_AFL="WAIT"

18.19.2.2.1.2.4.1 Calculations

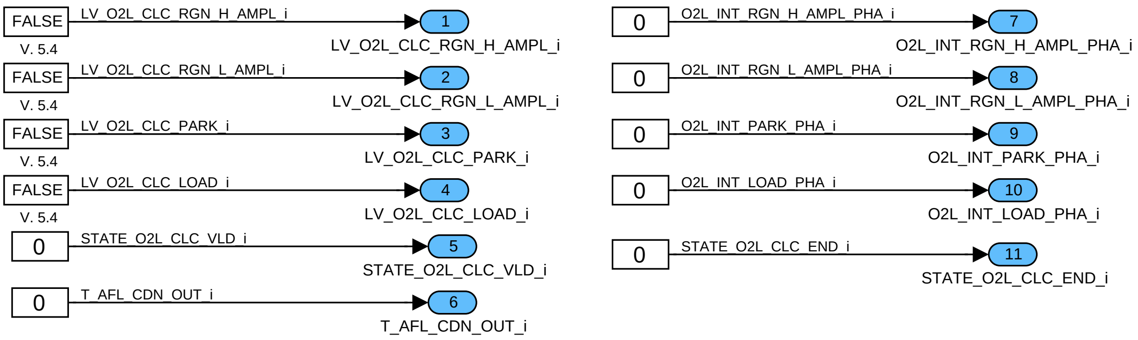


Figure 232:

18.19.2.3 Activation condition not fulfilled

O2L_INT_THD_TMP_DIAG_AFL=C_O2L_INT_THD_TMP_DIAG_AFL
All other variables are initialized with "0"

18.19.3 Detailed description for Action: ACTION_EGTR_ClcValidCheck

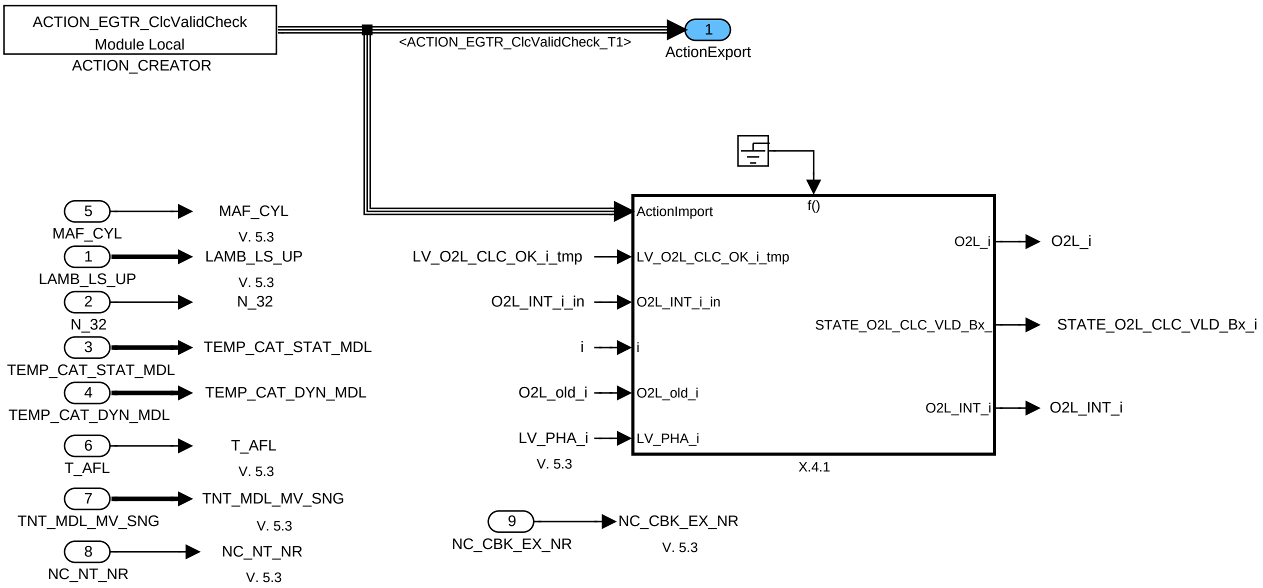


Figure 233:

18.19.3.1 Calculation of the O2L integral finished - check of the up and down lambda signal

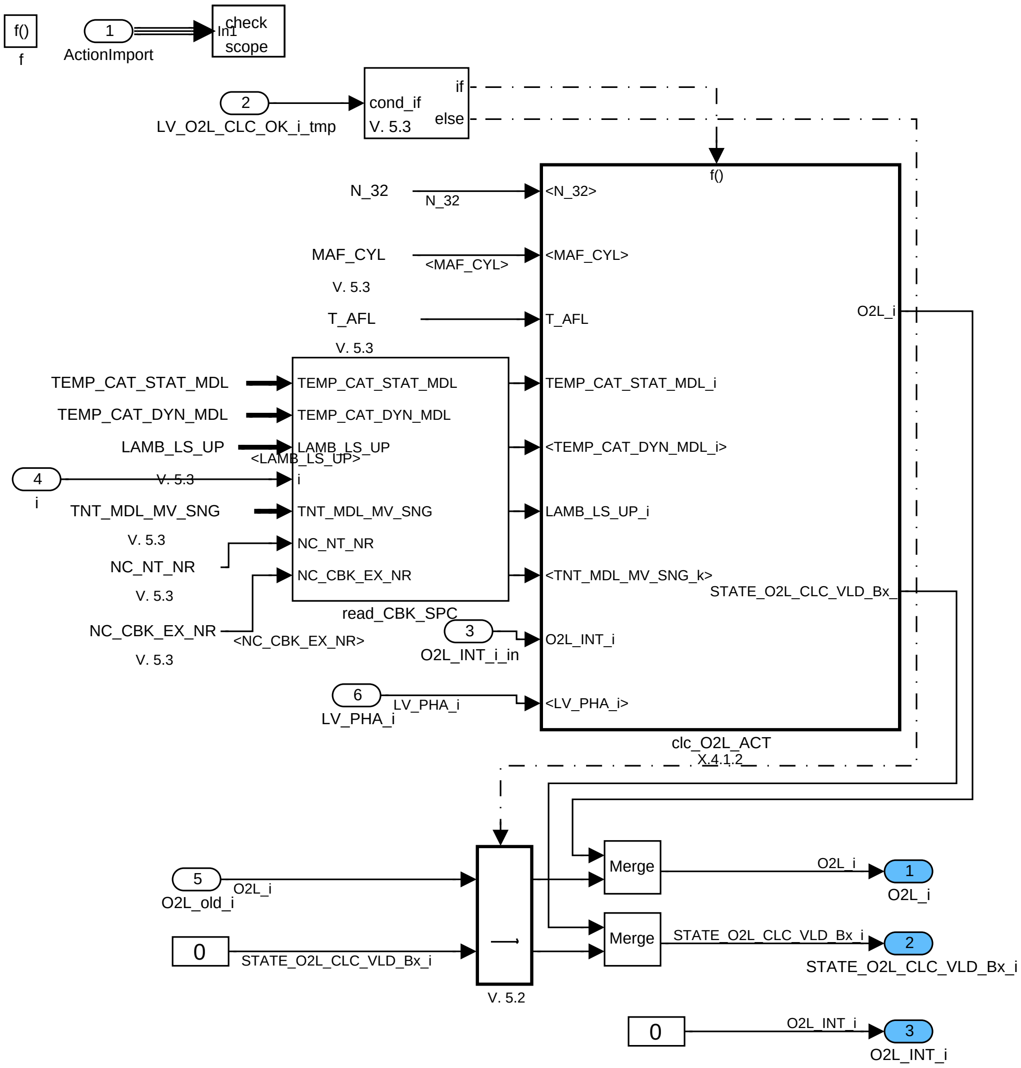


Figure 234:

18.19.3.1.1 Calculation and correction of the O2L value

Depending on the strategy for calculation of the O2L integral (configuration 1 and 2a or 2b), corresponding correction factors are used. For the configuration 2b, temperatures of TWC and NT are taken into account

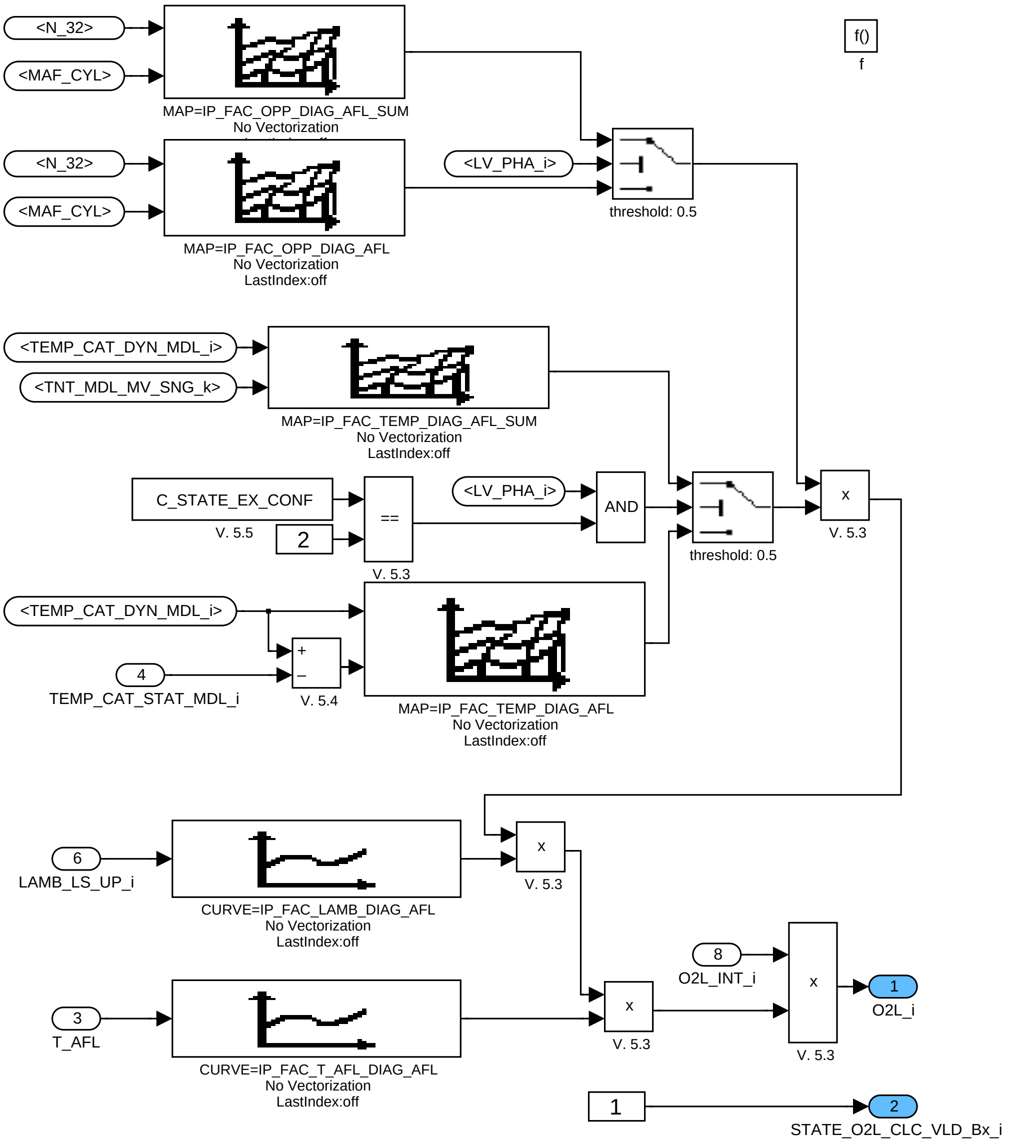


Figure 235:

18.19.4 Detailed description for Action: ACTION_EGTR_O2LIntegralCalc

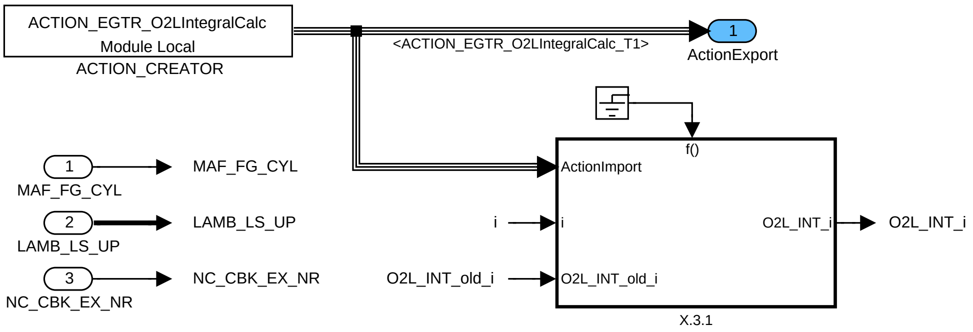


Figure 236:

18.19.4.1 Calculation of the O2L integral

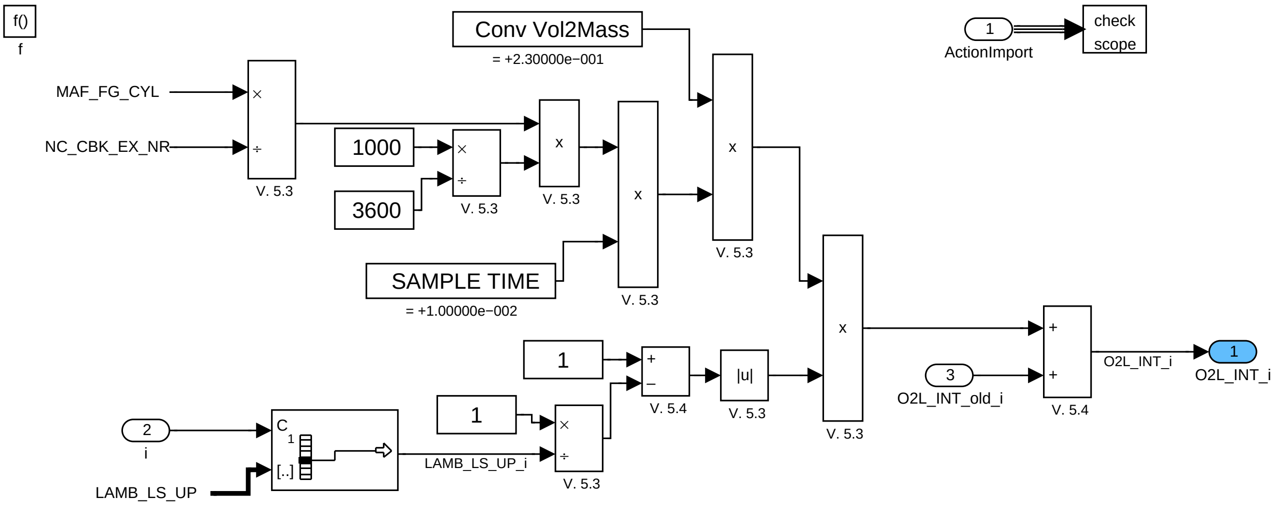


Figure 237:

18.20 OSC based TWC diagnosis for DI engines at lean operation - diagnosis value determination

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_CYC_CAT_DIAG_AFL [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Counter of diagnosis cycles for calculation of error indicator					
CTR_O2L_SUM_H_AMPL [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Counter for calculation of mean value of O2L calculated in high regeneration phases					
CTR_O2L_SUM_L_AMPL [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Counter for calculation of mean value of O2L calculated in low regeneration phases					
CTR_O2L_SUM_LOAD [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Counter for calculation of mean value of O2L calculated in load phases					
CTR_O2L_SUM_PARK [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Counter for calculation of mean value of O2L calculated in park phases					
ERR_SYM_CAT_DIAG_AFL [NC_CBK_EX_NR]	O/V	0 1 2 4 8	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	-	[-]
Detected error symptom for TWC lean diagnosis					
FAC_SUM_O2L_MMV_TMP [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Denominator for calculation of O2L_MMV_CAT_DIAG_AFL					
LV_CDN_DIAG_CAT_DIAG_AFL [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Diagnostic condition to start symptom detection (set to one when condition is fulfilled)					
LV_END_DIAG_AFL_TMP [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Temporary flag for end of the Diagnostic cycle without filtering					
LV_END_DIAG_CAT_DIAG_AFL [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Diagnostic performed at last one time					
LV_ERR_CAT_DIAG_AFL [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Present failure after filtering of diagnostic CAT_DIAG_AFL					
LV_ERR_CAT_DIAG_AFL_TMP [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Temporary present failure of diagnostic CAT_DIAG_AFL					
LV_O2L_AFL_CDN_MAT [NC_CBK_EX_NR]	-	0... 1H	0... 1	1	[-]
Calculation of O2L_AFL_CDN_MAT was successful					
LV_O2L_CAT_DIAG_AFL [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Calculation of general O2L value was successful					
LV_O2L_CAT_DIAG_AFL_SUM	V	0... 1H	0... 1	1	[-]
Calculation of O2L_CAT_DIAG_AFL_SUM was successful					
LV_O2L_CLC_READY [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Calculation of general diagnosis value is finished					
LV_O2L_MV_CLC_END [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Calculation of mean values during diagnosis cycle has been done					
LV_RGN_NT_REQ_AFL_PREV	-	0... 1H	0... 1	1	[-]
The previous value of LV_RGN_NT_REQ					
O2L_AFL_CDN_MAT [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Calculated O2L of PreCat in Lean Condition					
O2L_AFR_CDN_COR [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Corrected O2L value with mathematic method					
O2L_AFR_CDN_COR_MAT [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]

Corrected O2L value with empirical method					
O2L_CAT_DIAG_AFL [NC_CBK_EX_NR]	O/V/S	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Finally O2L value for calculation of Error Symptom					
O2L_CAT_DIAG_AFL_SUM	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Diagonal calculated O2L of both cylinder branches					
O2L_DELTA_AFR_CDN [NC_CBK_EX_NR]	V	0... FFFFH	-1.30844444444 ...1.30840451389	39.9306e-6	[g]
Delta between O2L values calculated during H_AMPL and L_AMPL regeneration phases					
O2L_H_AMPL_MV [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Mean value of O2L values calculated in regenerations with high amplitude phase					
O2L_H_AMPL_SUM [NC_CBK_EX_NR]	V	0... FFFFH	0... 20.93479	319.444e-6	[g]
Sum of O2L values calculated during regenerations with high amplitude phase					
O2L_L_AMPL_MV [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Mean value of O2L values calculated in regenerations with low amplitude phase					
O2L_L_AMPL_SUM [NC_CBK_EX_NR]	V	0... FFFFH	0... 20.93479	319.444e-6	[g]
Sum of O2L values calculated in regenerations with low amplitude phase					
O2L_LOAD_MV [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Mean value of O2L values calculated during load phases					
O2L_LOAD_SUM [NC_CBK_EX_NR]	V	0... FFFFH	0... 20.93479	319.444e-6	[g]
Sum of O2L values calculated during load phases					
O2L_MMV_CAT_DIAG_AFL [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Mean value of O2L values calculated in all phases					
O2L_NT_MAT	V	0... FFFFH	0... 20.93479	319.444e-6	[g]
Calculated O2L value of NOx catalyst					
O2L_PARK_MV [NC_CBK_EX_NR]	V	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Mean value of O2L values calculated in park phases					
O2L_PARK_SUM [NC_CBK_EX_NR]	V	0... FFFFH	0... 20.93479	319.444e-6	[g]
Sum of O2L values calculated in park phases					
O2L_SUM_CAT_DIAG_AFL [NC_CBK_EX_NR]	V	0... FFFFH	0... 20.93479	319.444e-6	[g]
Sum of O2L values of all phases for calculation of O2L_MMV_CAT_DIAG_AFL					
STATE_CAT_DIAG_AFL_OLD [NC_CBK_EX_NR]	-	0 1 2 3	PASSIVE RICH_CDN LEAN_CDN WAIT	-	[-]
Old value for STATE_CAT_DIAG_AFL					
STATE_O2L_MV_CLC_VLD [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Collected bits due to calculation of mean values at each phase					

Input Data:

C_STATE_EX_CONF	C_STATE_SENS_DOWN_MISS_CBK	LV_DC	LV_DIAG_END_SUM_DIAG_AFL [NC_CBK_EX_NR]
LV_DIAG_ERR_SUM_DIAG_AFL [NC_CBK_EX_NR]	LV_O2L_CLC_ACT_CDN [NC_CBK_EX_NR]	LV_RGN_NT_REQ	NC_CBK_EX_NR
O2L_LOAD_PHA [NC_CBK_EX_NR]	O2L_PARK_PHA [NC_CBK_EX_NR]	O2L_RGN_H_AMPL_PHA [NC_CBK_EX_NR]	O2L_RGN_L_AMPL_PHA [NC_CBK_EX_NR]
STATE_CAT_DIAG_AFL [NC_CBK_EX_NR]	STATE_O2L_CLC_VLD [NC_CBK_EX_NR]		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CTR_CYC_CAT_DIAG_AFL	1	1... FFH	1... 255	1	[-]
Number of diagnosis cycles for calculation of error indicator					
C_CTR_O2L_CYC_DIAG_END_AFL [NC_CBK_EX_NR]	1	1... FFH	1... 255	1	[-]
Number of valid cycles of O2L calculation in lean phase for finishing of the whole one diagnosis cycle					
C_CTR_O2L_CYC_DIAG_END_AFR [NC_CBK_EX_NR]	1	1... FFH	1... 255	1	[-]
Number of valid cycles of O2L calculation in rich phase for finishing of the whole one diagnosis cycle					
C_CTR_O2L_MV_CLC	1	1... FFH	1... 255	1	[-]
Number of necessary O2L calculations for calculation of mean value of each phase					
C_FAC_AFR_CDN_MAT	1	0... FFH	0... 15.9375	0.0625	[-]
Factor for calculation of corrected diagnosis					
C_O2L_CAT_DIAG_AFL_MAX	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Maximal O2L value used for initialization of diagnosis value					
C_O2L_THD_ERR_CAT_DIAG_AFL [NC_CBK_EX_NR]	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
OBD threshold of diagnosis value to set the diagnosis error					
C_O2L_THD_ERR_CAT_DIAG_AFL_SUM	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
OBD threshold of diagnosis value to set of diagnosis error depending on diagonally O2L calculation					
C_O2L_THD_NOT_ERR_CAT_DIAG_AFL [NC_CBK_EX_NR]	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
O2L threshold saying that the storage capacity of TWC is enough big to pre-end the diagnosis					
C_STATE_DIAG_CYC_CLC_ENA [NC_CBK_EX_NR]	1	0... FFH	0... 255	1	[-]
Choice of phases for finishing diagnosis cycle					
C_STATE_O2L_ERR_CLC	1	0... FH	0... 15	1	[-]
State for choice of O2L calculation strategy for calculation of diagnosis error					
C_STATE_O2L_MMV_CLC_ENA	1	0... FFH	0... 255	1	[-]
State to allow the calculation of the mean value of the O2L integrals					
IP_FAC_O2L_AFR_CDN_COR	8	0... FFH	0... 15.9375	0.0625	[-]
LDP_O2L_DELTA_IP_FAC_COR	8	0... FFFFH	-1.30844444444 ...1.30840451389	39.9306e-6	[g]
Correction factor for calculation of O2L value					
IP_O2L_CAT_DIAG_AFL_SUM	8*8	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
LDP_O2L_CAT_DIAG_1_IP_O2L_SUM	8	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
LDP_O2L_CAT_DIAG_2_IP_O2L_SUM	8	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Diagonally calculation of O2L due to both cylinder branches					
LC_O2L_CAT_DIAG_AFL_SUM	1	0... 1H	0... 1	1	[-]
Manually set calculation of diagonally O2L to valid					

Export Actions:

ACTION_EGTR_ClcO2L_SUMandMV(**IN** <LV_O2L_MV_CLC_ACT_TMP_i>, **IN** <STATE_O2L_CLC_VLD_Bx_i>, **IN** <O2L_i>, **OUT** <LV_O2L_SUM_CLC_i>, **INOUT** <CTR_SUM_i>, **INOUT** <O2L_SUM_i>, **INOUT** <O2L_MV_i>)
Calculation of mean value fo O2L

Description for Actions

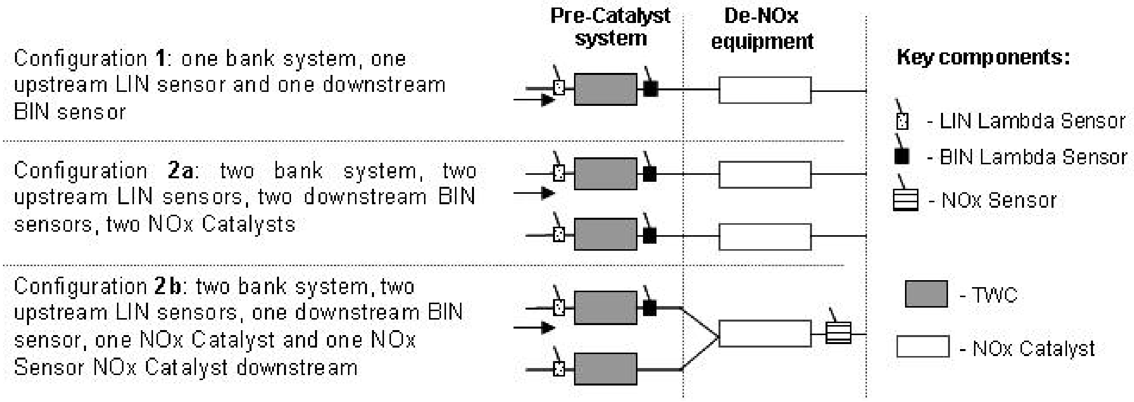
ACTION_EGTR_ClcO2L_SUMandMV(**IN** <LV_O2L_MV_CLC_ACT_TMP_i>, **IN** <STATE_O2L_CLC_VLD_Bx_i>, **IN** <O2L_i>, **OUT** <LV_O2L_SUM_CLC_i>, **INOUT** <CTR_SUM_i>, **INOUT** <O2L_SUM_i>, **INOUT** <O2L_MV_i>)
Calculation of mean value fo O2L

Parameter	Type	Hex.Limits	Phys.Limits	Resol.	Unit
LV_O2L_MV_CLC_ACT_TMP_i	IN	0... 1H	0... 1	1	[-]
Calculation of mean value allowed					
STATE_O2L_CLC_VLD_Bx_i	IN	0... 1H	0... 1	1	[-]
Read bit of STATE_O2L_CLC_VLD					
O2L_i	IN	0... FFFFH	0... 2.61684896	39.9306e-6	[g]
O2L value					
LV_O2L_SUM_CLC_i	OUT	0... 1H	0... 1	1	[-]
Sum calculation valid					
CTR_SUM_i	INOUT	0... FFH	0... 255	1	[-]
Counter for calculation of mean value of O2L					
O2L_SUM_i	INOUT	0... FFFFH	0... 20.93479	319.444e-6	[g]
Sum of O2L values					
O2L_MV_i	INOUT	0... FFFFH	0... 2.61684896	39.9306e-6	[g]
Mean value of O2L values					

General Information

In this module the diagnosis values are calculated and the link to the error management is implemented.
The calculation of diagnosis value is possible in several ways. Only one method is taken for calculation of finally diagnosis value. Choice of method is partly depending on exhaust gas configuration and number of regeneration cycles during diagnosis cycle. The finally bank specific diagnosis value is after that taken for calculation of diagonally diagnosis value. At the end the error indicator and the error symptom are calculated using bank specific and bank diagonally diagnosis value and the error management is called with condition "true".
Short description of all methods, numbering is equal to value in C_STATE_O2L_ERR_CLC and bit in STATE_O2L_MV_CLC_VLD:
Method 0: Mean value of O2L calculated during regeneration with high amplitude phase (excepted configuration 2b)
Method 1: Mean value of O2L calculated during regeneration with low amplitude phase (excepted configuration 2b)
Method 2: Mean value of O2L calculated during park phase (excepted configuration 2b)
Method 3: Mean value of O2L calculated during load phase (excepted configuration 2b)
Method 4: Mean value of O2L mean values calculated in all occurred phases (excepted configuration 2b)
Method 5: Corrected O2L value with empirical method to compensate of lambda sensor dynamic using phases with high and low regeneration amplitude (except configuration 2b)
Method 6: Corrected O2L value with mathematical method to compensate of lambda sensor dynamic using phases with high and low regeneration amplitude (except configuration 2b)
Method 7: O2L calculated using TWC- and NOx-Catalyst oxygen storage capacity for calculating of O2L value for bank without downstream Lambda Sensor (only configuration 2b)

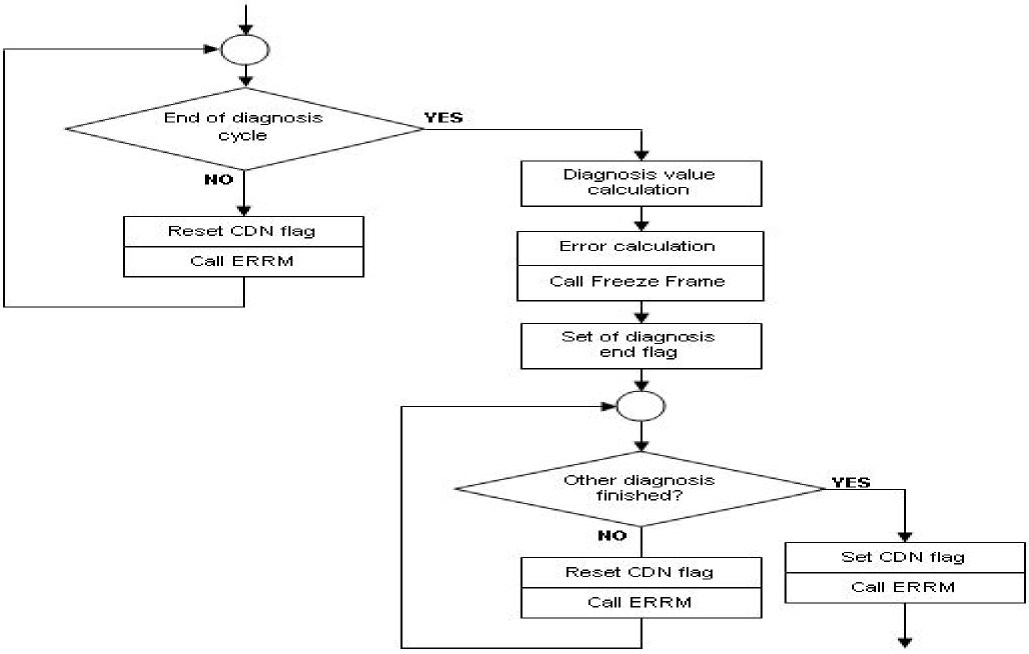
Picture 1: The diagnosis is able for following exhaust gas configurations:


Configuration **1**: one bank system, one upstream LIN sensor and one downstream BIN sensor

Configuration **2a**: two bank system, two upstream LIN sensors, two downstream BIN sensors, two NOx Catalysts

Configuration **2b**: two bank system, two upstream LIN sensors, one downstream BIN sensor, one NOx Catalyst and one NOx Sensor NOx Catalyst downstream

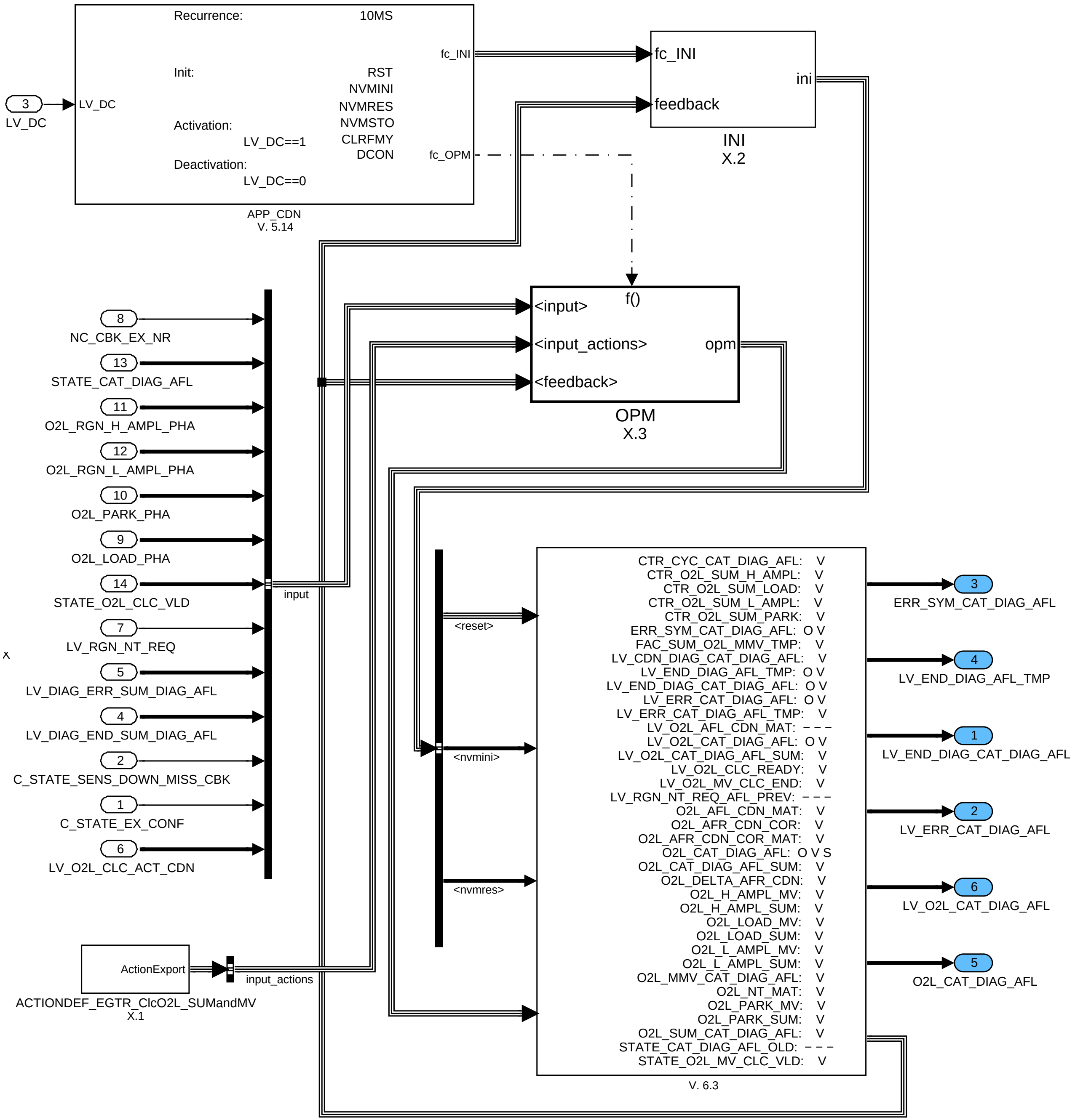
Picture 2: Signal flow for calculation of Error Indicator



If two separate exhaust bank systems are concerned, then:
i = 1, for cylinder bank 1
i = 2, for cylinder bank 2
Otherwise:
i = 1

Application Conditions

Initialization: RST, NVMINI, NVMRES, NVMSTO, CLRFMY
Recurrence: 10MS
Activation: LV_DC==1
Deactivation: LV_DC==0



SDA_SRS / SDA V 5.2.3 07−Feb−2007

Figure 238:

18.20.1 INITIALIZATION

18.20.1.1 Initialization of NVMY data

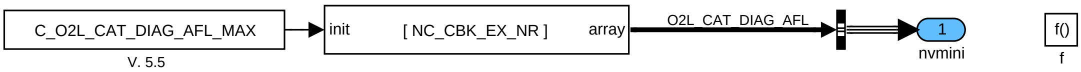


Figure 239:

18.20.1.2 Initialization at reset and function activation (start driving cycle)

At reset and function activation (DCON) all variables except NVMY data and diagnostic result are initialized with "0".
The diagnostic result is initialized according the filtering configuration.

18.20.2 FORMULA SECTION

The formula section consists of two main sections:
Regeneration cycles counter, bank diagonally calculations inclusive calculation of method 7 and the final diagnosis result
Methods 1 to 6 and ERRM interface

18.20.2.1 Regeneration cycles counter, bank diagonally calculations inclusive method 7 and final diagnosis value

18.20.2.1.1 Counting of the regeneration cycles

18.20.2.1.1.1 Bank specific calculation of the regeneration cycles counter

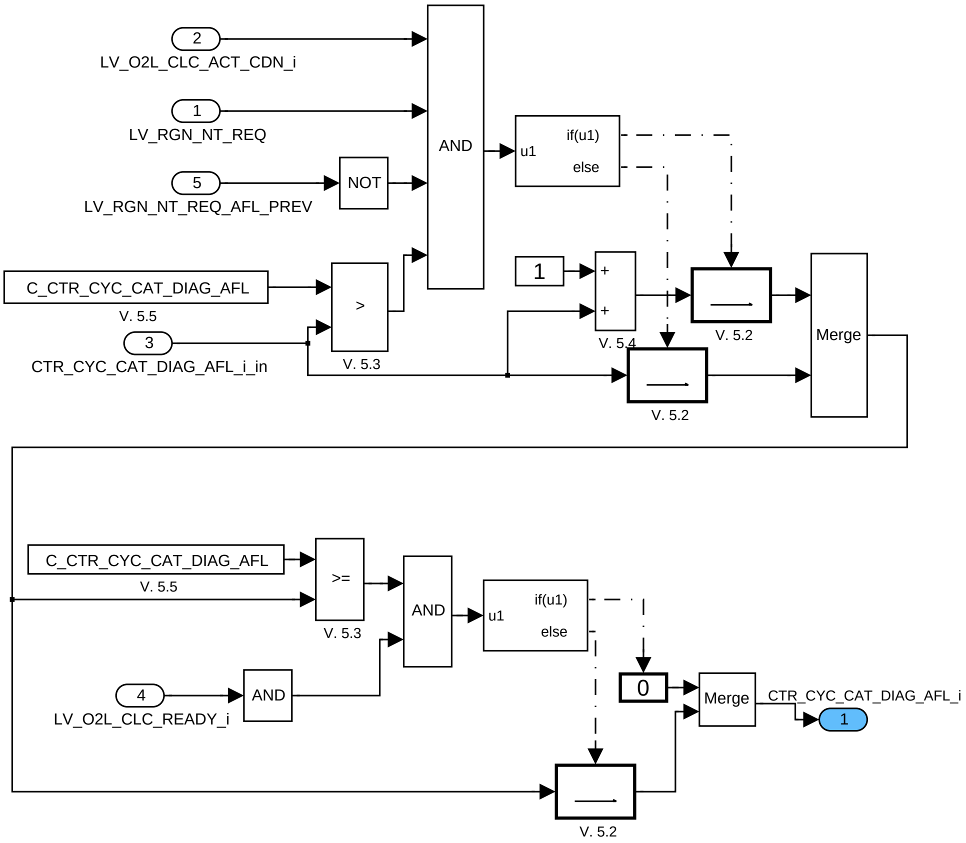


Figure 240:

18.20.2.1.2 Bank diagonally calculations inclusive method 7 and final diagnosis value

This section consists of:
Calculation of flag indicating, that diagnosis on any bank is finished
Calculation of O2L value of NOx-Catalyst in case of configuration 2b
Calculation of temporary O2L value for bank without binary downstream sensor in case of exhaust gas configuration 2b
Calculation of finally O2L value for both banks and validation bit in case of exhaust gas configuration 2b
Choice of method for calculation of finally diagnosis value
Calculation of sum diagnosis value over all banks

18.20.2.1.2.1 Calculation of flag indicating, that diagnosis on any bank is finished

The calculation of all banks is synchronized. This flag says that the calculation of all banks is finished. Then first the diagonal calculation and after that the calculation of diagnosis results is done.
This flag is active only for one cycle.

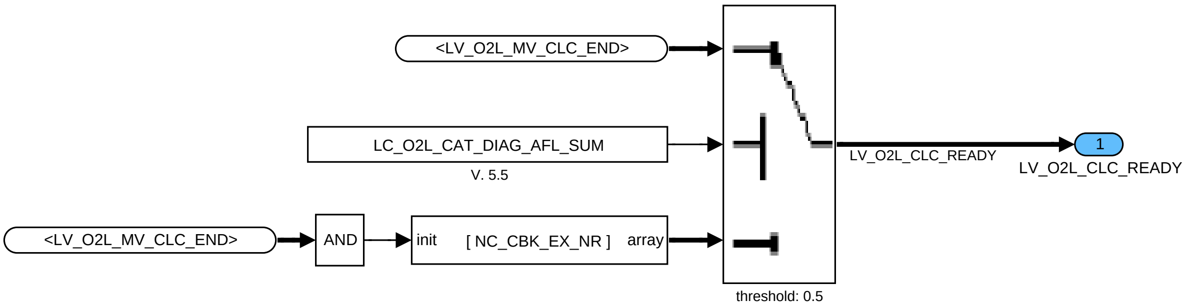


Figure 241:

18.20.2.1.2.2 Calculation of O2L value of NOx-Catalyst in case of configuration 2

The O2L value of NOx catalyst is a difference between O2L value of the whole line and O2L value of precatalyst calculated at bank with downstream binary sensor. The O2L value calculated at bank with downstream binary sensor is corrected according to the dynamic of this sensor. This is given using the IP_FAC_O2L_AFR_CDN_COR (see also Method 5)

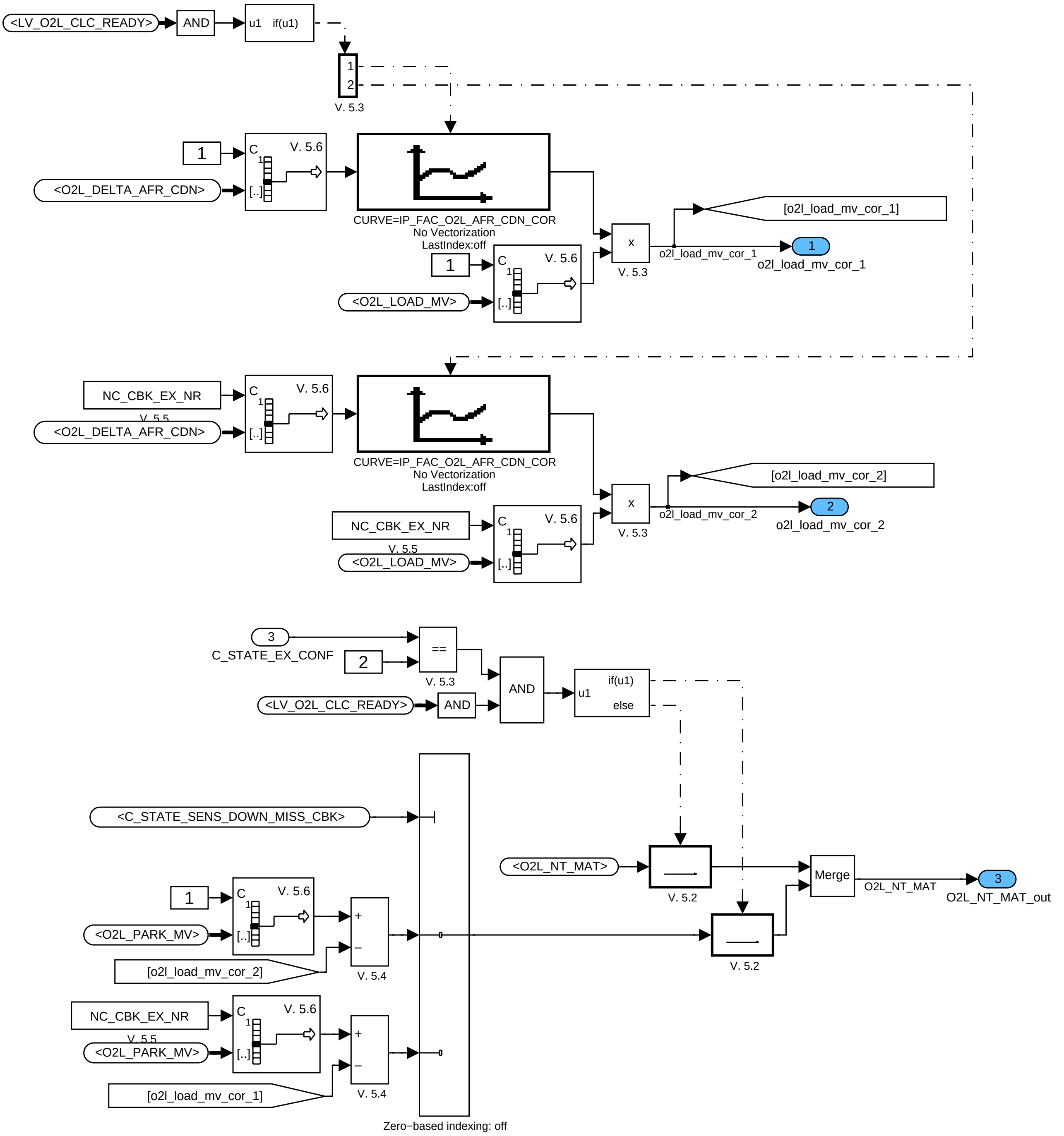


Figure 242:

18.20.2.1.2.3 Calculation of temporary O2L value for bank without binary downstream sensor in case of exhaust gas configuration 2

The O2L value of precatalyst is a difference between O2L value of the whole line and O2L value of NOx catalyst calculated at bank without downstream binary sensor

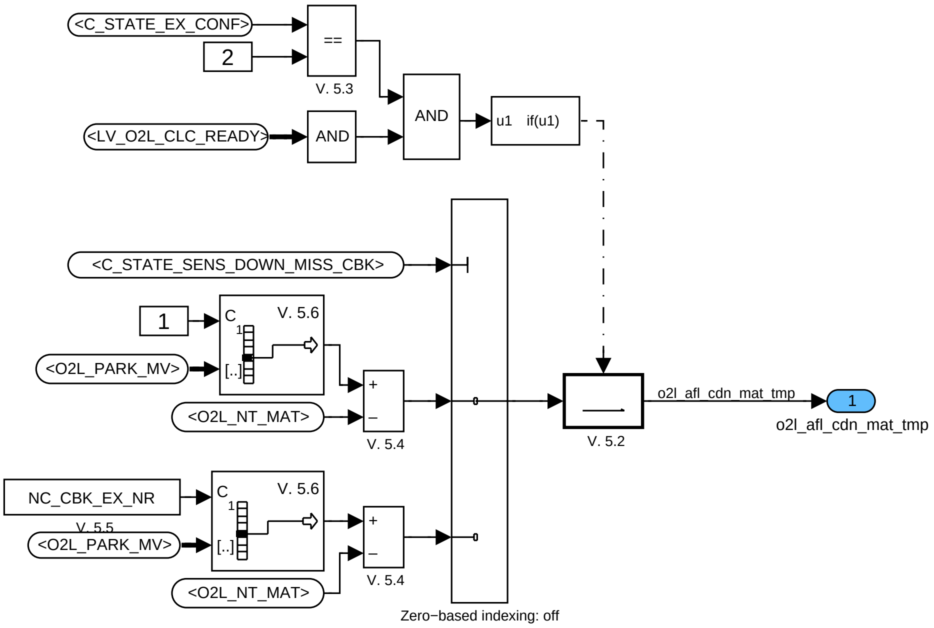


Figure 243:

18.20.2.1.2.4 Calculation of finally O2L value for both banks and validation bit in case of exhaust gas configuration 2 (Method 7)

After all diagonal calculations the result is checked, the diagnosis value of Method 7 is assigned and the valid flag is set for one recurrence.

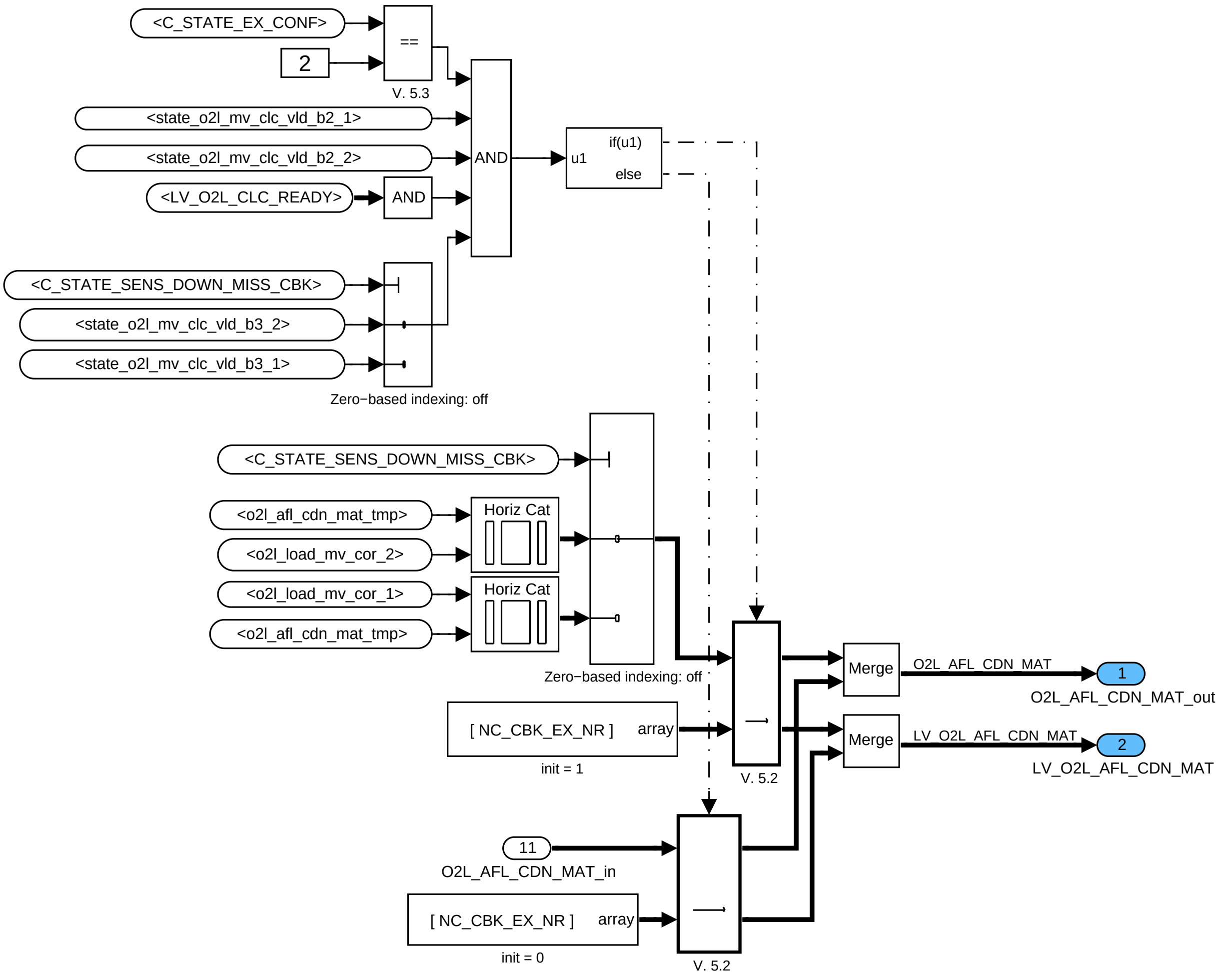


Figure 244:

18.20.2.1.2.5 Choice of method for calculation of finally diagnosis value

The O2L_CAT_DIAG_AFL used for calculation of diagnosis results is represented by O2L_xxx calculated by chosen method. The LV says that the calculation was plausible. This flag is stored in STATE_O2L_MV_CLC_VLD (when this diagnosis was finished before others) and the STATE is reset when the diagnosis condition was true.

18.20.2.1.2.5.1 Choice of the method and calculation temporary diagnosis value

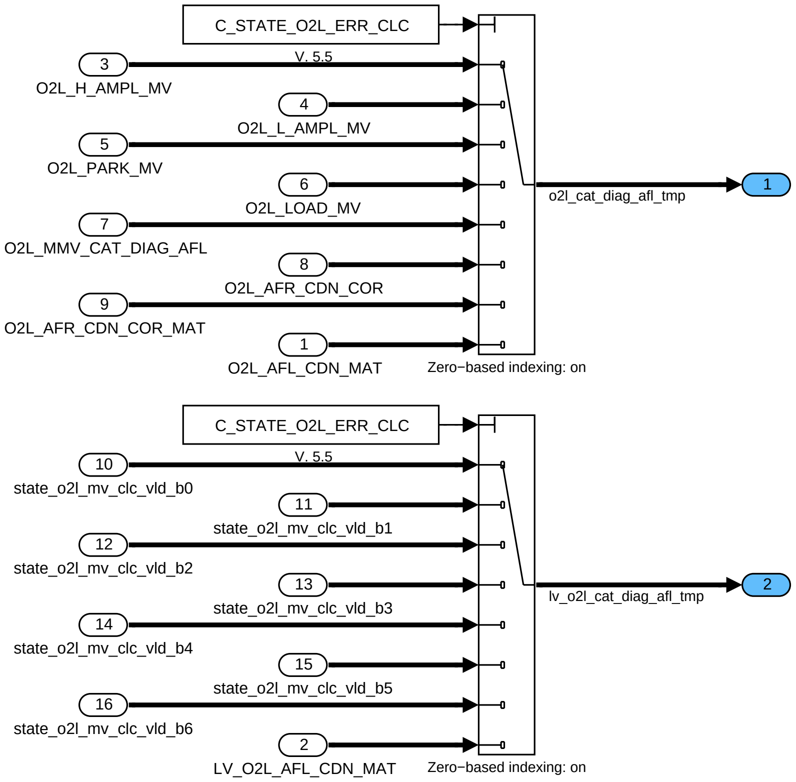


Figure 245:

18.20.2.1.2.5.2 Final diagnosis value and plausibility indicator

18.20.2.1.2.5.2.1 Calculation

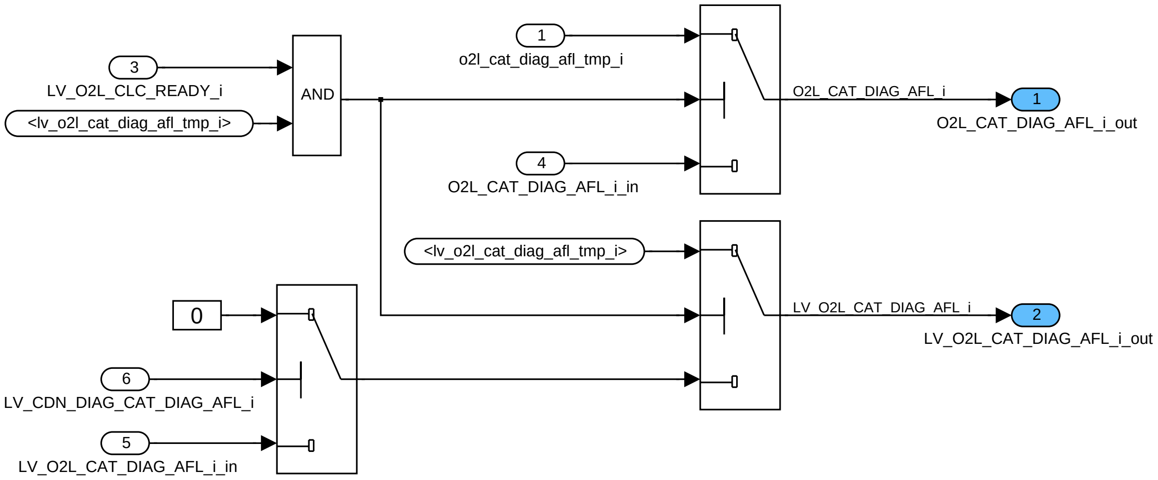


Figure 246:

18.20.2.1.2.6 Calculation of sum diagnosis value over all banks

Indexes "1" and "2" are used regarding cylinder bank 1 and 2. Sum diagnosis value considers all banks

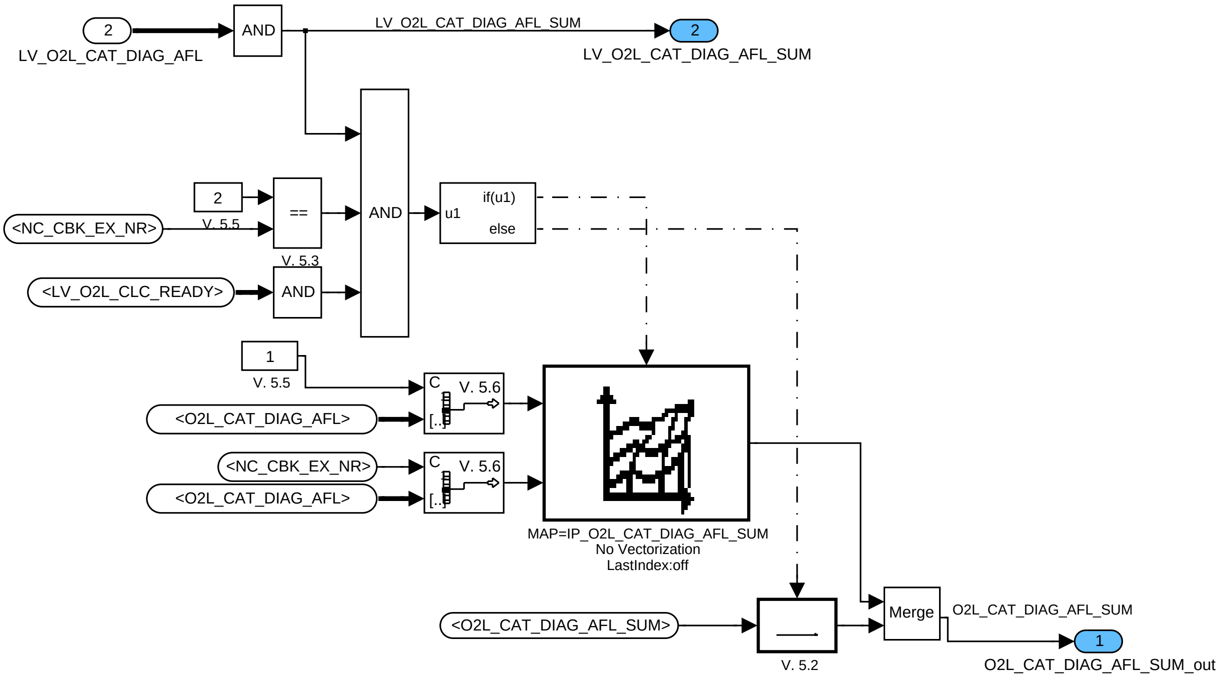


Figure 247:

18.20.2.2 Multiple bank system calculation

18.20.2.2.1 Methods 1 to 6 and ERRM interface

This section consists of three further sections:
Calculation of O2L sum and mean values
Calculation of corrected O2L values and set up of state including plausibility bits
Calculation of error symptom, condition flag and ERRM interface

18.20.2.2.1.1 Calculation of O2L sum and mean values

This section consists of:
Calculation of flag triggering calculation of mean values
Calculation of phase specific sum and mean values
Calculation of mean value of phase specific mean values (Method 4)

18.20.2.2.1.1.1 Calculation of flag triggering calculation of mean values

At the end of defined diagnosis cycles the flag triggering calculation of mean values (only one recurrence) is set for every bank separate (LV_O2L_MV_CLC_ACT_TMP). The flag LV_O2L_MV_CLC_END saves the LV_O2L_MV_CLC_ACT_TMP and is used in next cycle for calculation of summary flag of all banks (LV_O2L_CLC_READY). After that the diagonal values and diagnosis results are calculated.
The flag triggering calculation of the mean values can be set by checking counter of regeneration cycles (independent of plausibility) or counter of valid O2L calculation cycles

18.20.2.2.1.1.1.1 Check of regeneration cycles counter

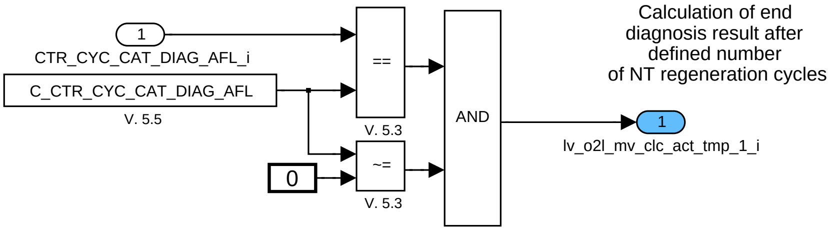


Figure 248:

18.20.2.2.1.1.1.2 Check of counter of valid O2L calculation cycles

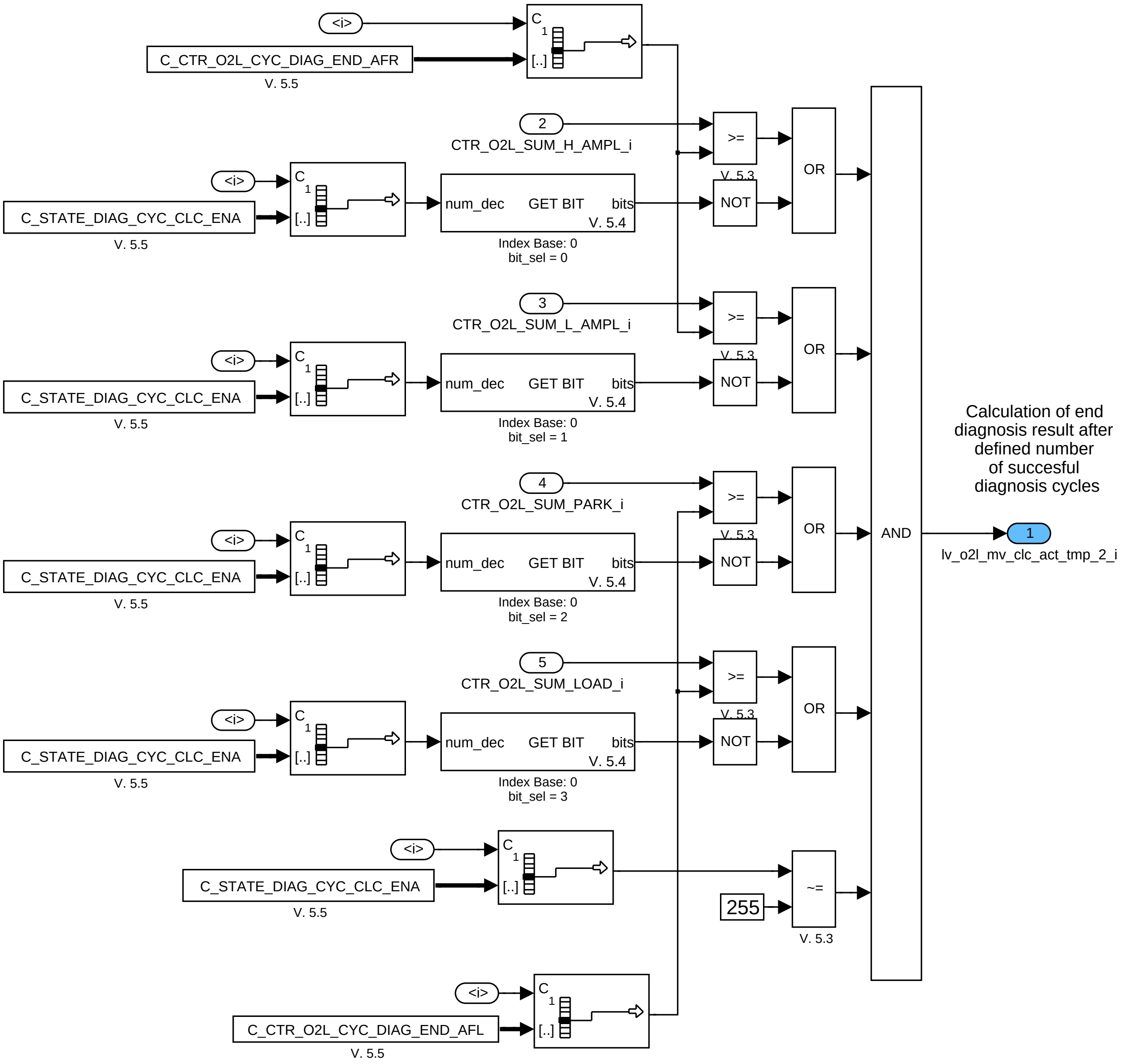


Figure 249:

18.20.2.2.1.1.1.3 Check of storage capacity of TWC for calculation of pre-end diagnosis result (no error present)

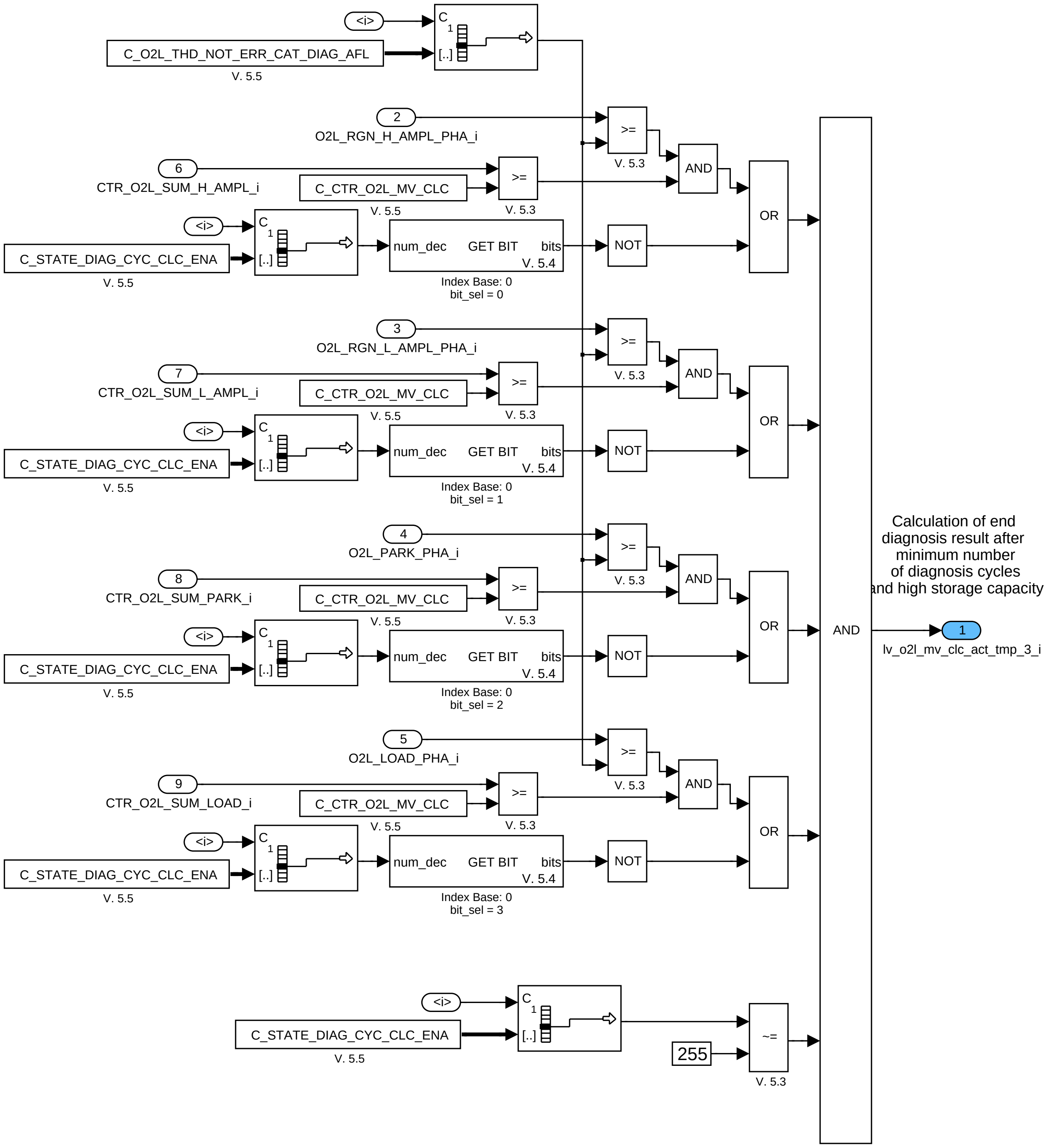


Figure 250:

18.20.2.2.1.1.1.4 Set final triggering flag for calculation of mean values

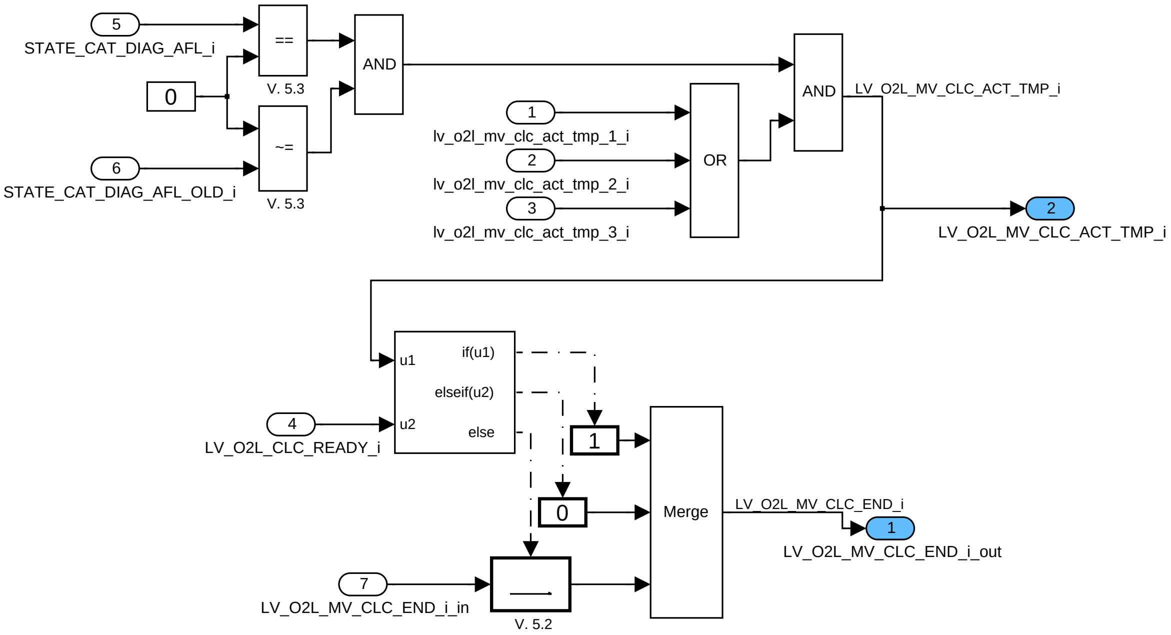


Figure 251:

18.20.2.2.1.1.2 Calculation of phase specific sum and mean values

The sum and mean values are calculated using local ACTION ClcO2L_SUMandMV

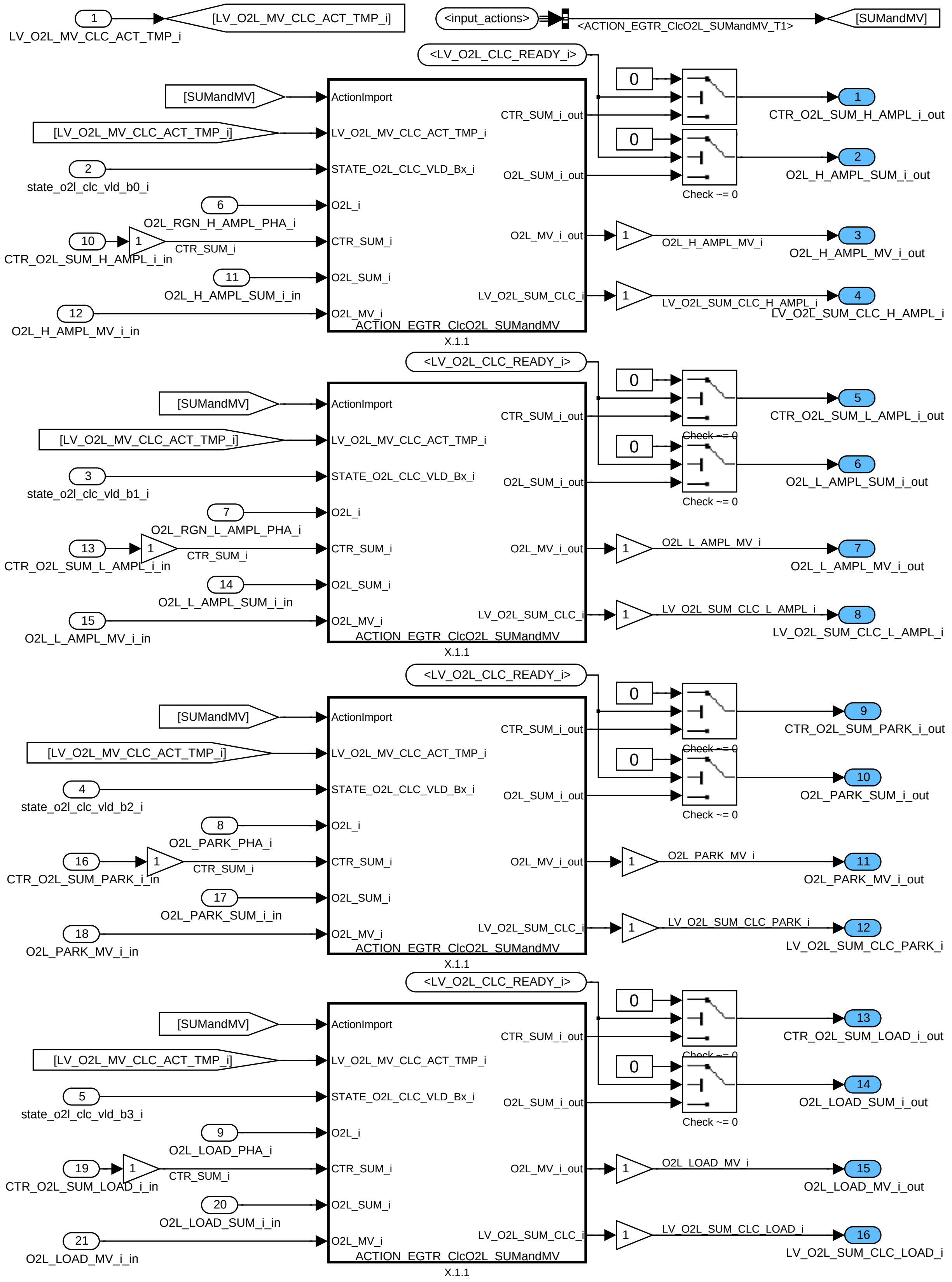

1
LV_O2L_MV_CLC_ACT_TMP_i
[LV_O2L_MV_CLC_ACT_TMP_i]
<input_actions>
<ACTION_EGTR_ClcO2L_SUMandMV_T1>
[SUMandMV]
<LV_O2L_CLC_READY_i>
0
[SUMandMV]
ActionImport
CTR_SUM_i_out
[LV_O2L_MV_CLC_ACT_TMP_i]
LV_O2L_MV_CLC_ACT_TMP_i
2
state_o2l_clc_vld_b0_i
STATE_O2L_CLC_VLD_Bx_i
O2L_SUM_i_out
6
O2L_i
O2L_RGN_H_AMPL_PHA_i
10
CTR_SUM_i
CTR_O2L_SUM_H_AMPL_i_in
CTR_SUM_i
O2L_MV_i_out
11
O2L_SUM_i
O2L_H_AMPL_SUM_i_in
LV_O2L_SUM_CLC_i
12
O2L_MV_i
O2L_H_AMPL_MV_i_in
ACTION_EGTR_ClcO2L_SUMandMV
X.1.1
Check ~= 0
1
CTR_O2L_SUM_H_AMPL_i_out
2
O2L_H_AMPL_SUM_i_out
O2L_H_AMPL_MV_i
3
O2L_H_AMPL_MV_i_out
LV_O2L_SUM_CLC_H_AMPL_i
4
LV_O2L_SUM_CLC_H_AMPL_i
3
state_o2l_clc_vld_b1_i
7
O2L_RGN_L_AMPL_PHA_i
13
CTR_O2L_SUM_L_AMPL_i_in
14
O2L_L_AMPL_SUM_i_in
15
O2L_L_AMPL_MV_i_in
5
CTR_O2L_SUM_L_AMPL_i_out
6
O2L_L_AMPL_SUM_i_out
O2L_L_AMPL_MV_i
7
O2L_L_AMPL_MV_i_out
LV_O2L_SUM_CLC_L_AMPL_i
8
LV_O2L_SUM_CLC_L_AMPL_i
4
state_o2l_clc_vld_b2_i
8
O2L_PARK_PHA_i
16
CTR_O2L_SUM_PARK_i_in
17
O2L_PARK_SUM_i_in
18
O2L_PARK_MV_i_in
9
CTR_O2L_SUM_PARK_i_out
10
O2L_PARK_SUM_i_out
O2L_PARK_MV_i
11
O2L_PARK_MV_i_out
LV_O2L_SUM_CLC_PARK_i
12
LV_O2L_SUM_CLC_PARK_i
5
state_o2l_clc_vld_b3_i
9
O2L_LOAD_PHA_i
19
CTR_O2L_SUM_LOAD_i_in
20
O2L_LOAD_SUM_i_in
21
O2L_LOAD_MV_i_in
13
CTR_O2L_SUM_LOAD_i_out
14
O2L_LOAD_SUM_i_out
O2L_LOAD_MV_i
15
O2L_LOAD_MV_i_out
LV_O2L_SUM_CLC_LOAD_i
16
LV_O2L_SUM_CLC_LOAD_i

Figure 252:

18.20.2.2.1.1.3 Mean value of phase specific mean values (Method 4)

This is calculation of the mean value of O2L values calculated in different phases (e.g. H_AMPL and L_AMPL). The phase, which is taken into the calculation, is chosen through C_STATE_O2L_MMV_CLC_ENA

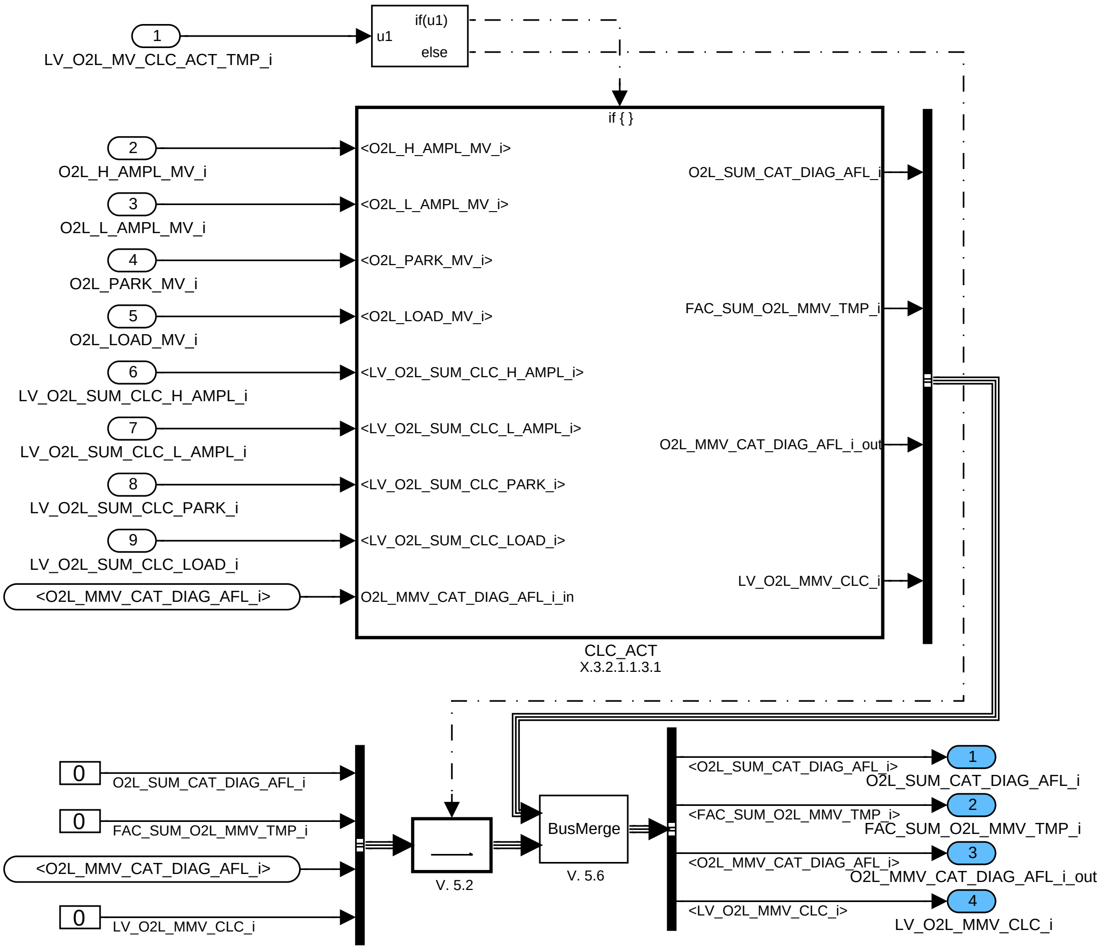


Figure 253:

18.20.2.2.1.1.3.1 Calculations

18.20.2.2.1.1.3.1.1 Calculation of sum

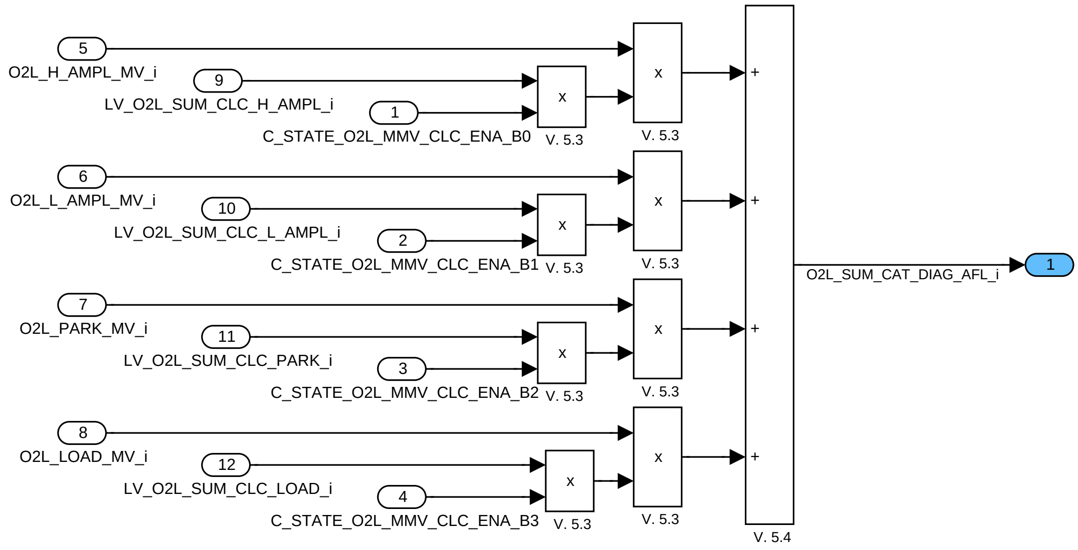


Figure 254:

18.20.2.2.1.1.3.1.2 Calculation of denominator

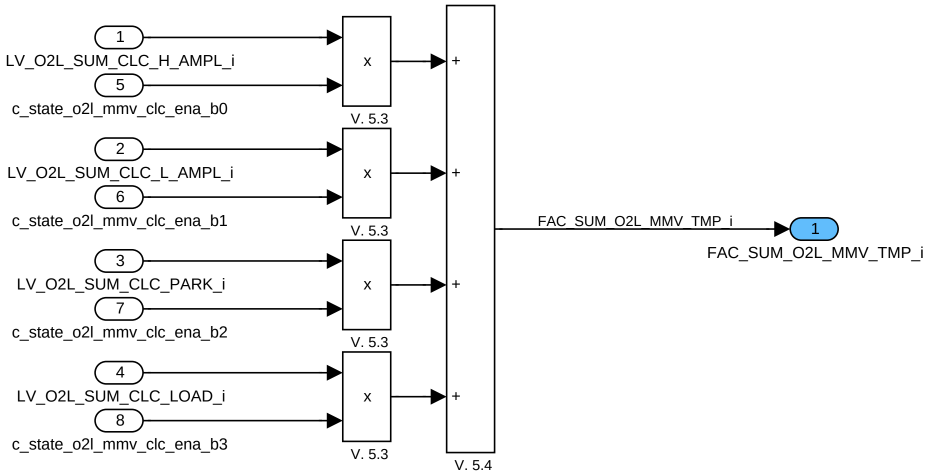


Figure 255:

18.20.2.2.1.1.3.1.3 Check denominator for calculation of MMV

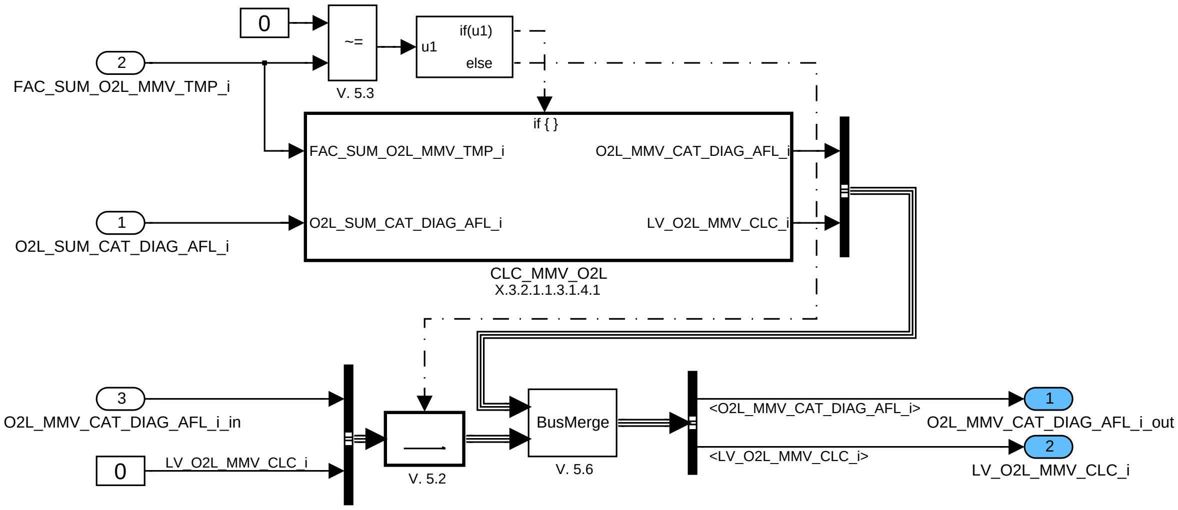


Figure 256:

18.20.2.2.1.1.3.1.3.1 Calculation of MMV

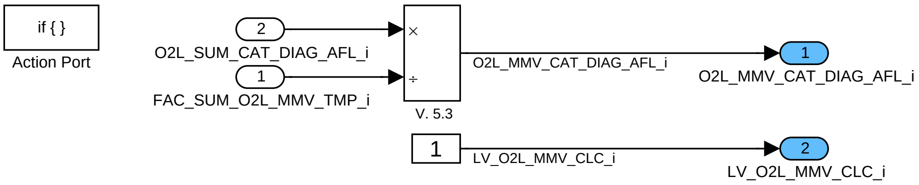


Figure 257:

18.20.2.2.1.2 Calculation of corrected O2L values and set up of state including plausibility bits

This section consists of:
Calculation of corrected O2L values
Set up of state including plausibility bits

18.20.2.2.1.2.1 Calculation of corrected O2L values (Methods 5, 6)

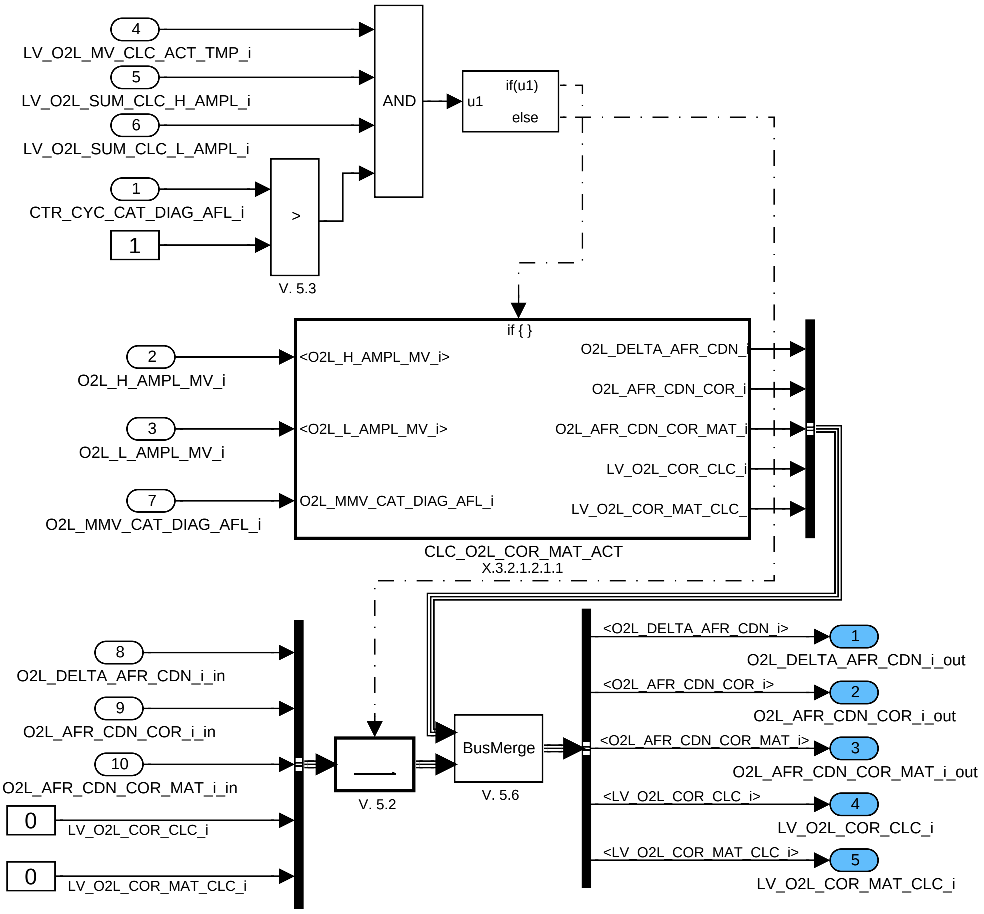


Figure 258:

18.20.2.2.1.2.1.1 Calculations

18.20.2.2.1.2.1.1.1 Calculation of delta O2L between H_AMPL and L_AMPL

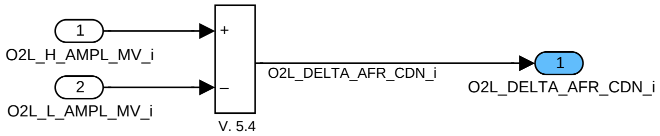


Figure 259:

18.20.2.2.1.2.1.1.2 Calculation of corrected O2L value with method 5

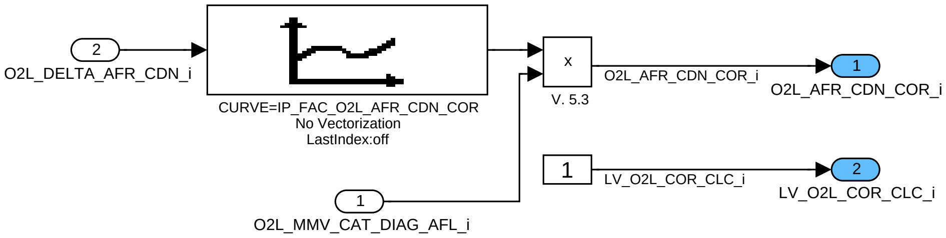


Figure 260:

18.20.2.2.1.2.1.1.3 Calculation of corrected O2L value with method 6

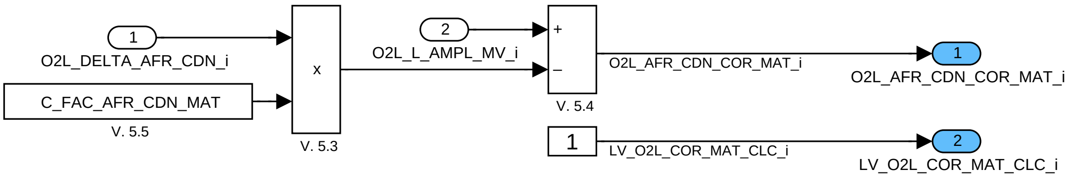


Figure 261:

18.20.2.2.1.2.2 Collecting of validation bit of all O2L calculation methods into a state

For each method for calculation of final diagnosis value exists one validation bit:
Bit 0: Using of O2L mean value from H_AMPL phase
Bit 1: Using of O2L mean value from L_AMPL phase
Bit 2: Using of O2L mean value from PARK phase
Bit 3: Using of O2L mean value from LOAD phase
Bit 4: Using of O2L mean value mean values of all phases
Bit 5: Using of O2L value corrected with empirical method
Bit 6: Using of O2L value corrected with mathematical method
Bit 7: Using of O2L calculated in lean condition in case of configuration 2b

18.20.2.2.1.2.2.1 Set up of the state

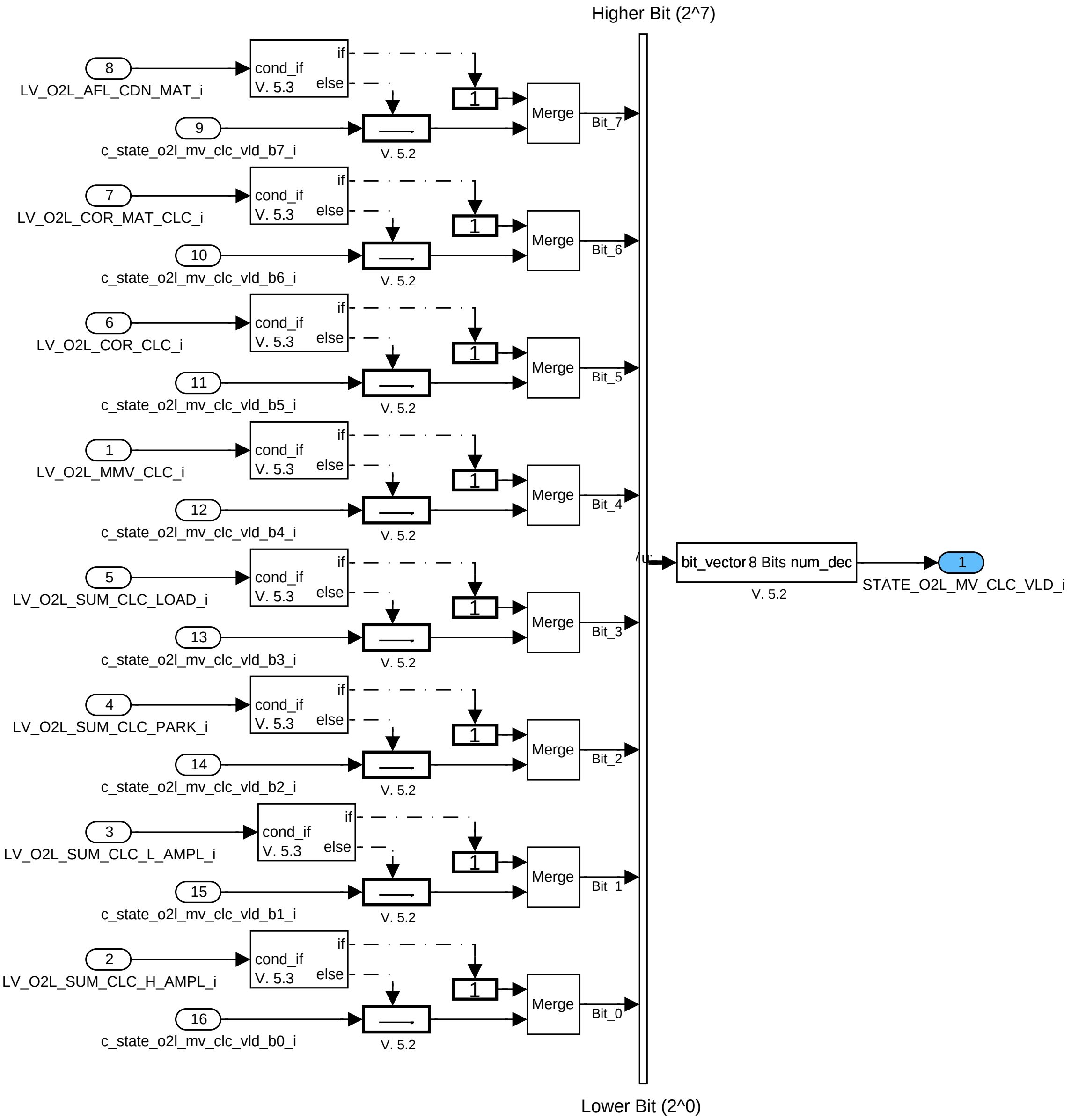


Figure 262:

18.20.2.2.1.2.2.2 Reset of the state

The STATE_O2L_MV_CLC_VLD is reset when the diagnosis condition bit was true in previous cycle

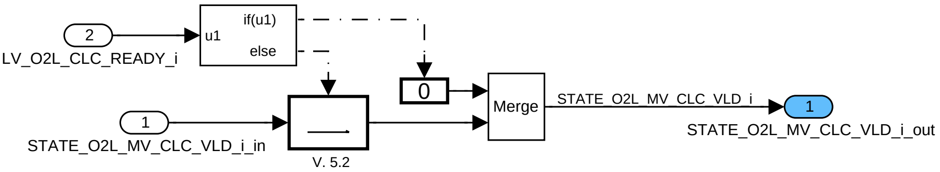


Figure 263:

18.20.2.2.1.3 Calculation of error symptom, condition flag and ERRM interface

This section consists of:
Calculation of temporary diagnosis error after finishing of diagnosis cycle

Calculation of temporary diagnosis end bit
Calculation of condition bit for calculation of error symptom using temporary error indicator after finishing of this and other relevant diagnosis
Interface to ERRM

18.20.2.2.1.3.1 Calculation of temporary diagnosis error after finishing of diagnosis cycle

The temporary diagnosis error is calculated at the end of diagnosis cycle. At this time, pre-store of environment data is executed

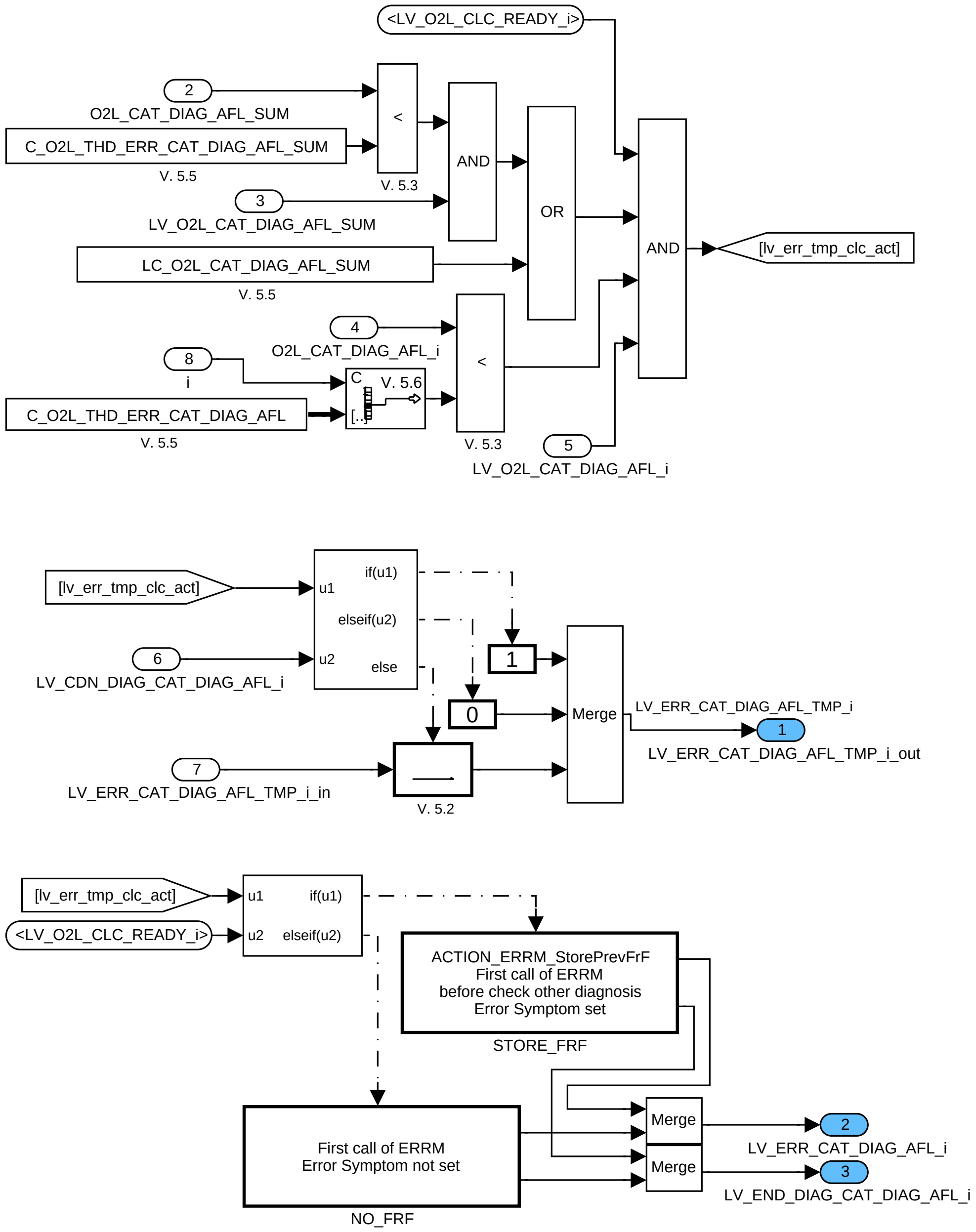


Figure 264:

18.20.2.2.1.3.2 Calculation of temporary diagnosis end bit

The temporary "end" bit is set when diagnosis cycle is finished and calculated O2L value is plausible.

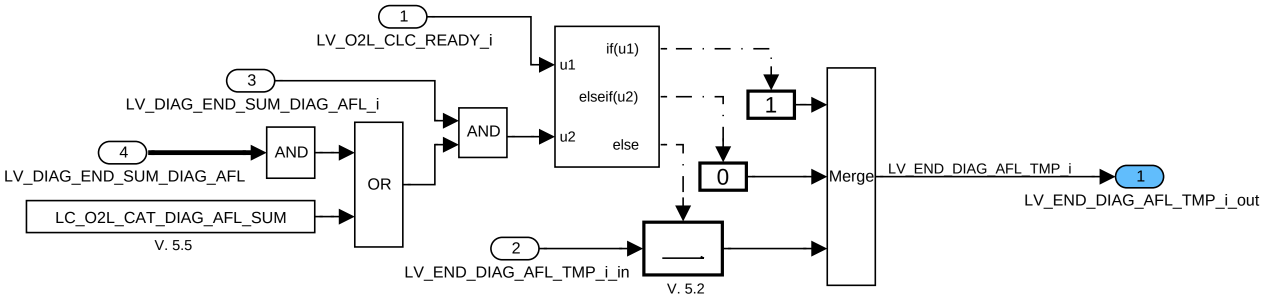


Figure 265:

18.20.2.2.1.3.3 Calculation of diagnosis condition bit and diagnosis error symptom

The diagnosis condition bit can be set when all other relevant diagnosis are finished and the result of this diagnosis is plausible. The calculation of error symptom is done after set of condition bit. The diagnosis condition bit is set only for one recurrence.

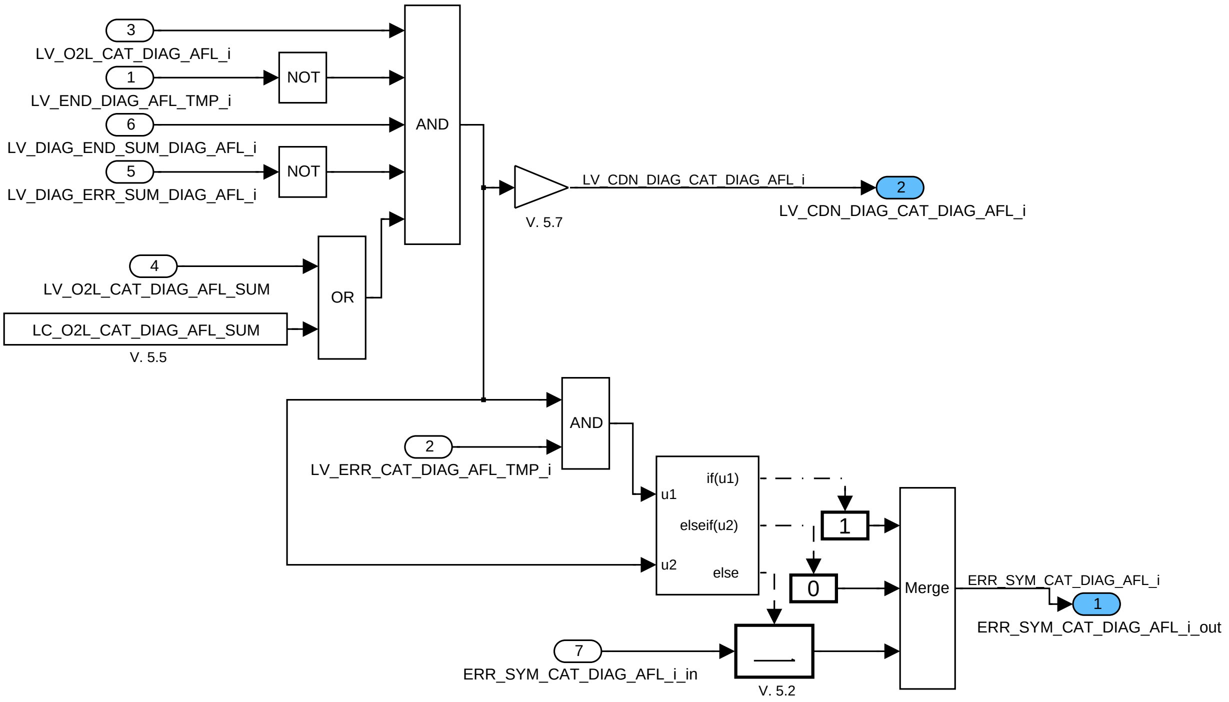


Figure 266:

18.20.2.2.1.3.4 Interface to error management

The diagnosis values are valid when the diagnosis condition flag is true. In this case the error management will take the diagnosis values into account.

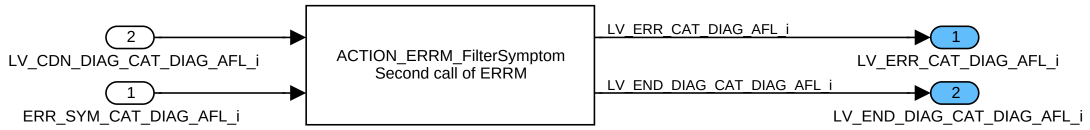


Figure 267:

18.20.3 Detailed description for Action: ACTION_EGTR_ClcO2L_SUMandMV

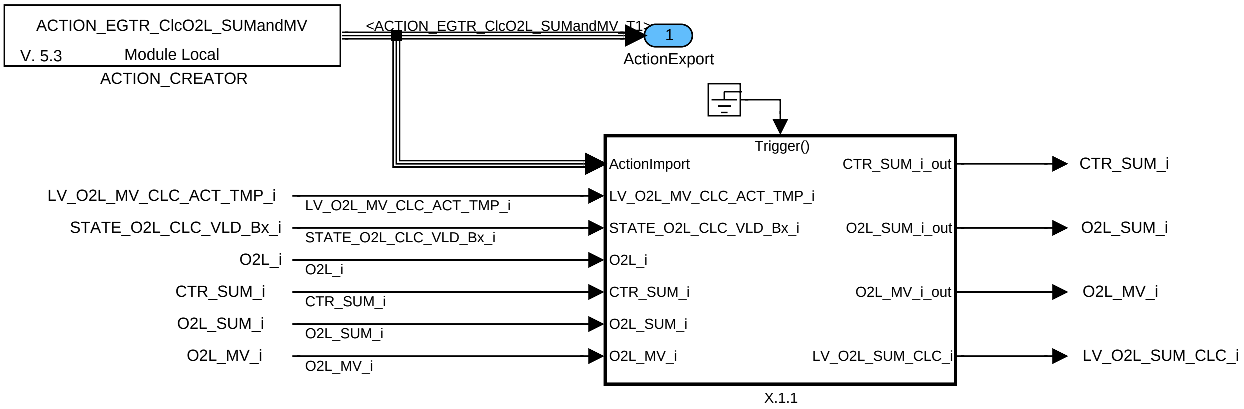


Figure 268:

18.20.3.1 Check of temporary trigger for calculation either sum, or mean o2l value

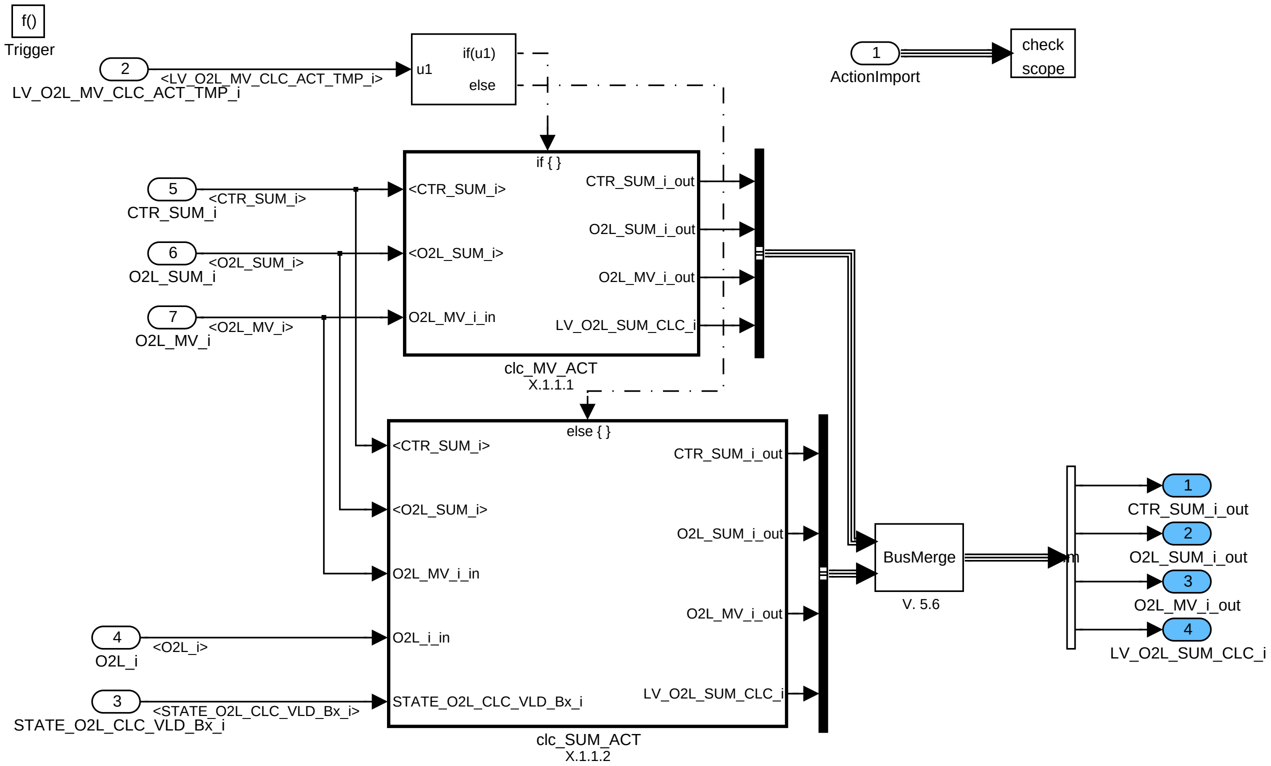


Figure 269:

18.20.3.1.1 Check number of calculated phases for calculation of o2l mean value

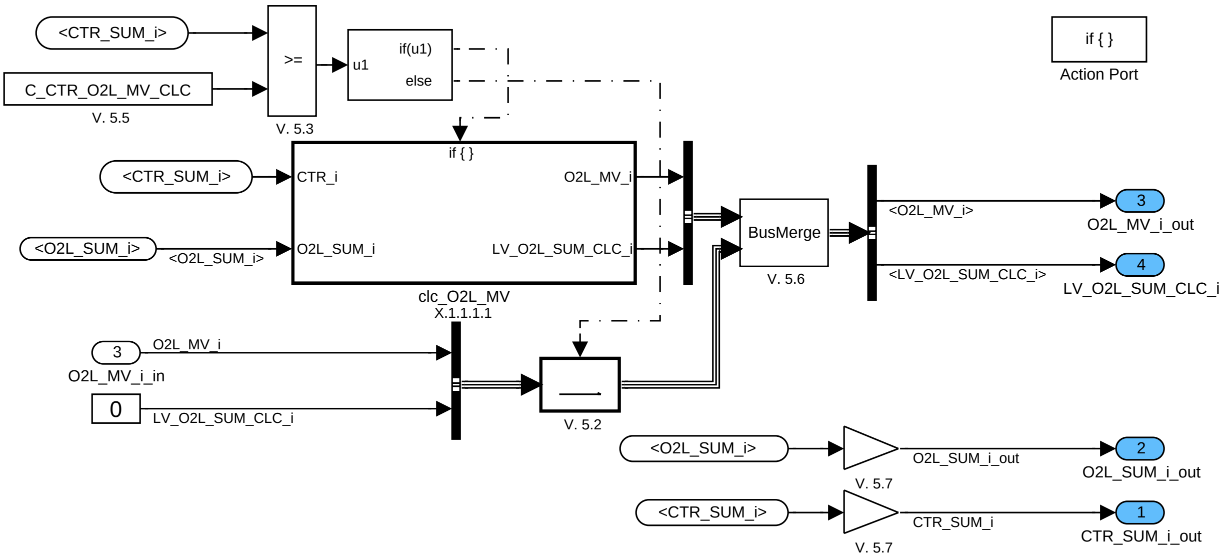


Figure 270:

18.20.3.1.1.1 Calculation of mean o2l value

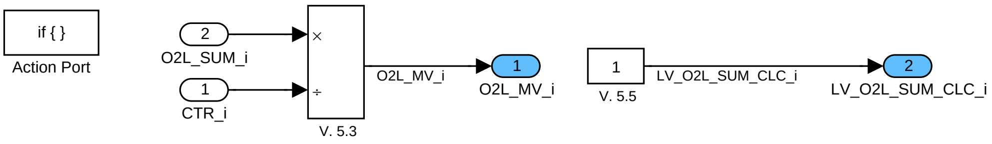


Figure 271:

18.20.3.1.2 Calculation o2l sum values

The sum value is just sum of o2l values calculated in several phases

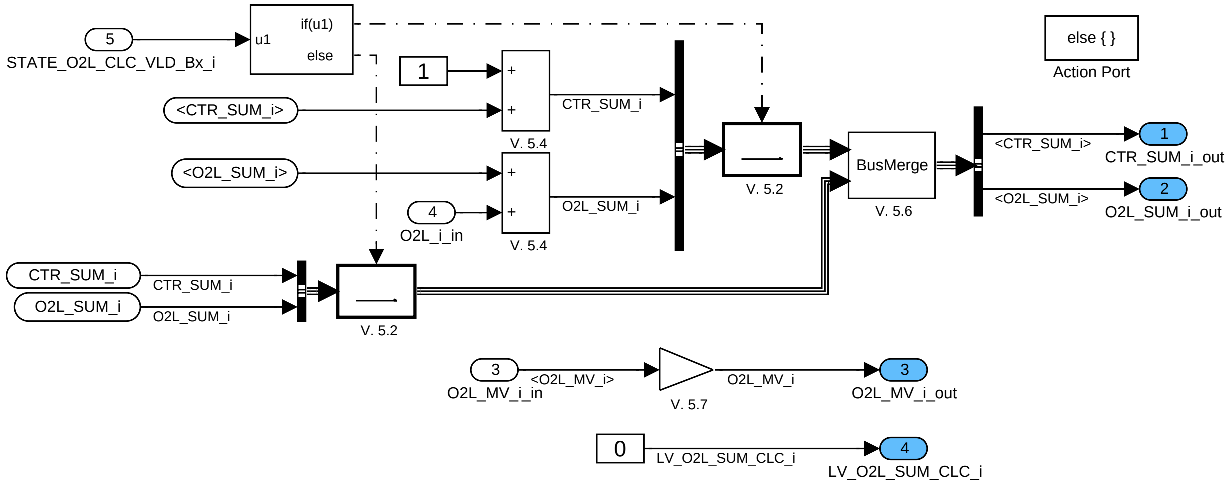


Figure 272:

18.21 TWC diagnosis at lean operation application incidences

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_CAT_DIAG_AFL[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Inhibition condition for the TWC lean diagnosis					
LV_DIAG_ERR_SUM_DIAG_AFL[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Collected flag or errors of oher relevant diagnosis					
LV_DIAG_END_SUM_DIAG_AFL[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Collected flag that oher relevant diagnosis are finished					

Input data:

LV_LS_DOWN_DIAG_END[NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_IGK	LV_LS_UP_DIAG_END[NC_CBK_EX_NR]
LV_ERR_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LS_UP[NC_CBK_EX_NR]	STATE_ERR_NS_OBD_2[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
STATE_END_NS_OBD_2[NC_NOX_SENS_CONF]	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MEC_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_MEC_IVVT_IN[NC_NR_CBK_IVVT]
LV_ERR_SLV_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_SLV_IVVT_IN[NC_NR_CBK_IVVT]	LV_ERR_FL_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_PUC_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_SCG_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_OC_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_EL_CPS
LV_ERR_DIAGCPS	LV_ERR_FSD[NC_CBK_EX_NR]	LV_MIS_STATE_A	LV_MIS_STATE_B
LV_ERR_CRK_PLAUS	LV_ERR_CRK_SYN	LV_ERR_CRK_TOOTH_PER	LV_ERR_CRK_TOOTH
LV_ERR_PLAUS_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_PLAUS_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_PER_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_PER_CAM_IN[NC_NR_CAM_CBK]
LV_ERR_REF_CRK_CAM_IN[NC_NR_CBK_IVVT]	LV_ERR_REF_CRK_CAM_EX[NC_NR_CBK_IVVT]	LV_ERR_SYN_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_SYN_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_TCO_EL	LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS	LV_ERR_TCO_STUCK
LV_ERR_AMP	LV_ERR_AMP_PLAUS	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]
LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS
LV_ERR_VCV	LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_TCO_STUCK_RNG
LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	NC_NR_CAM_CBK	NC_CYL_NR	NC_NR_CBK_IVVT
LV_ERR_MAP_TPS_PLAUS			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_CAT_DIAG_AFL	1	0...FFH	0...255	1	-
Decrement value for antibounce counter					
C_ABC_INC_CAT_DIAG_AFL	1	0...FFH	0...255	1	-
Antibounce counter increment					
C_ABC_MAX_CAT_DIAG_AFL	1	1...FFH	1...255	1	-
Maximum value for antibounce counter					
C_STATE_NS_OBD_2_DIAG_AFL	1	0...FFFFH	0...6.5535E+4	1	-
Choice of NS single diagnosis for calculation of summary bits					

general specification

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_DIAG_CAT_DIAG_AFL_DEAC[NC_CBK_EX_NR]	1	0...1H	0...1	1	-
Deactivation flag for the TWC lean diagnosis					
LC_DIAG_SUM_DIAG_AFL	1	0...1H	0...1	1	-
Manually set of end and error sum flags of other relevant diagnosis to passed					

18.21.1 General information

The flag LV_INH_DIAG_CAT_DIAG_AFL permit to deactivate the corresponding diagnosis.

If two separate exhaust bank systems are concerned, then:

i = 1, for cylinder bank 1

i = 2, for cylinder bank 2

Otherwise:

i = 1

Application Condition

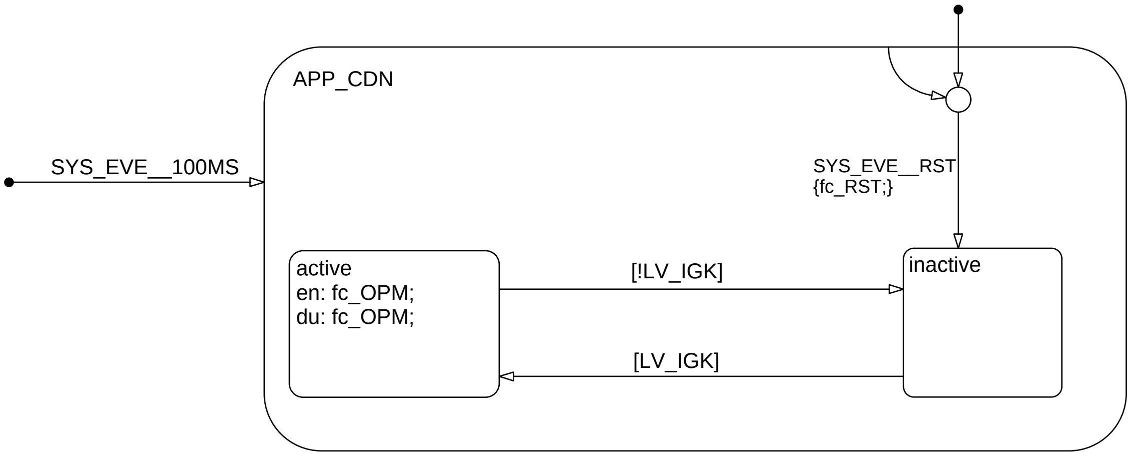
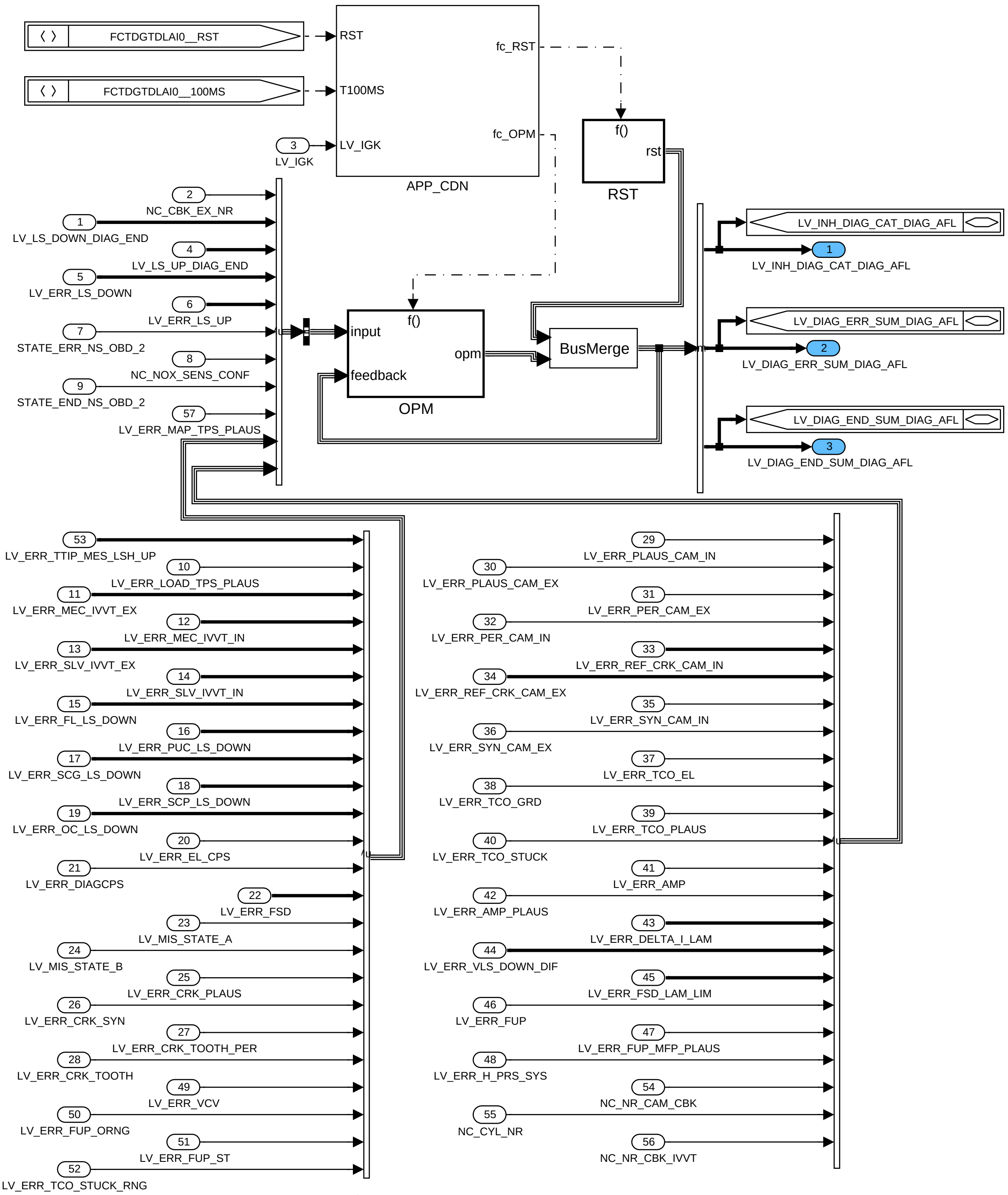


Figure 273 EGTR_FCTDGTDLAI0/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 19−Dec−2005

Figure 274 EGTR_FCTDGTDLAI0

18.21.1.1 SUBFUNCTION: Merge

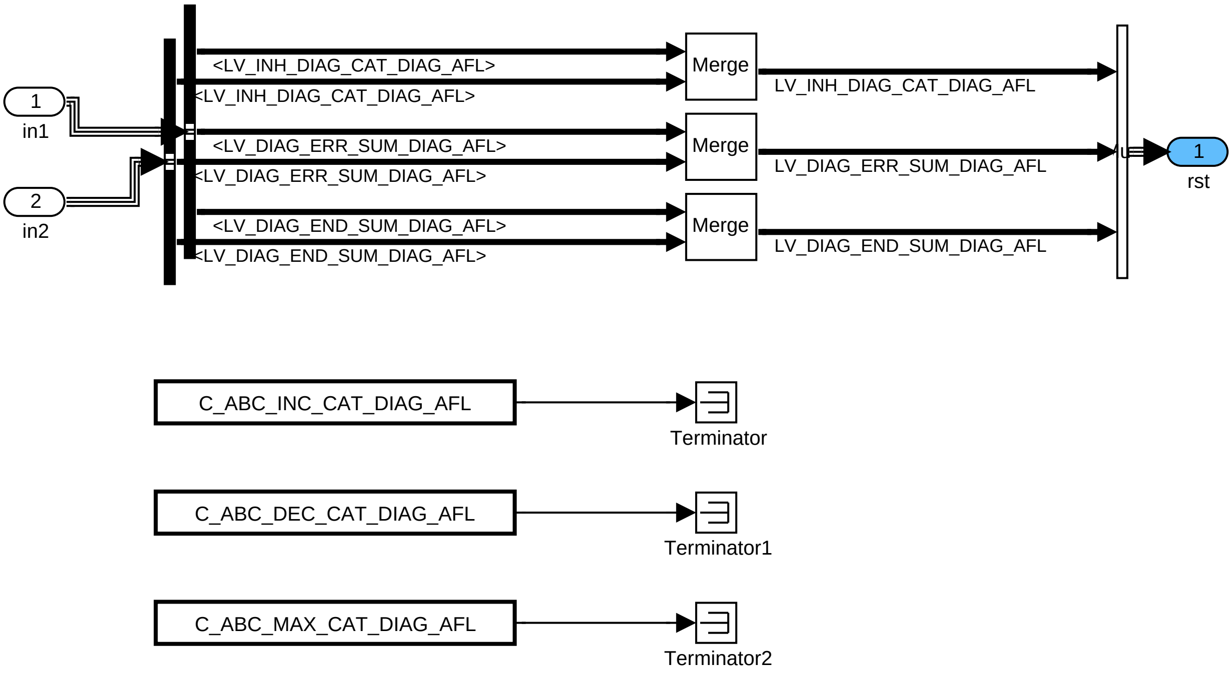


Figure 275 EGTR_FCTDGTDLAI0/ Merge

18.21.1.2 FORMULA SECTION

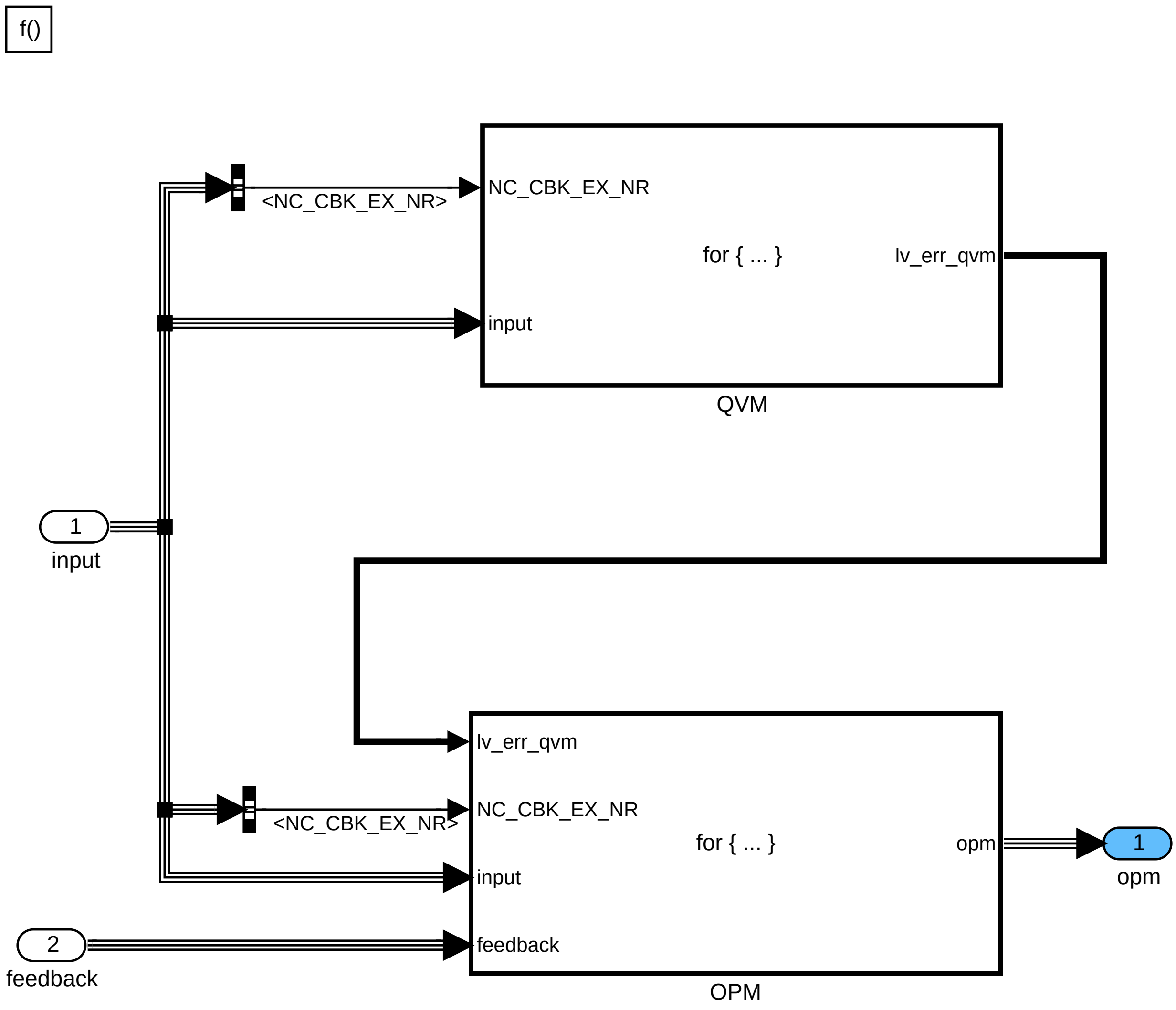


Figure 276 EGTR_FCTDGTDLAI0/ OPM

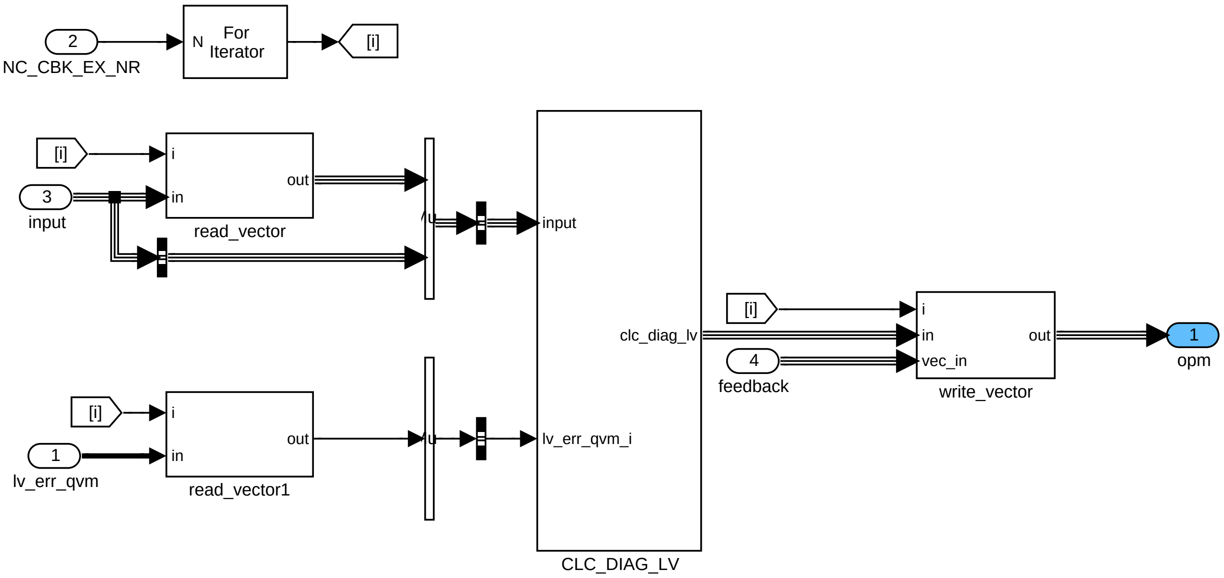


Figure 277 EGTR_FCTDGTDLAI0/ OPM/ OPM

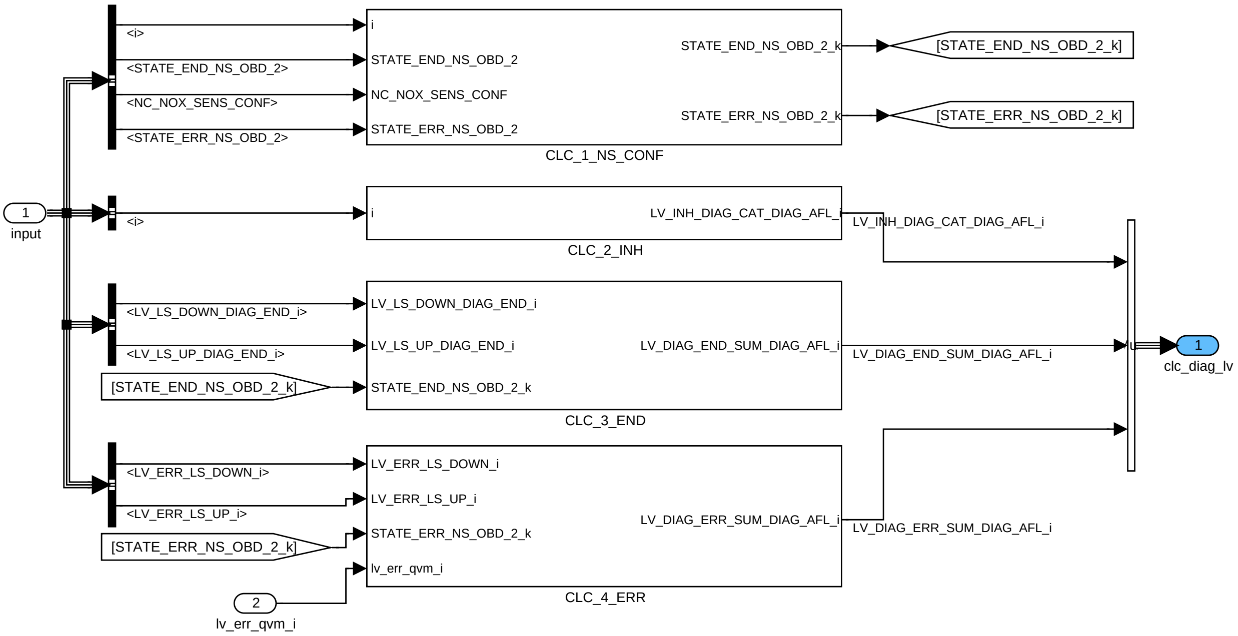


Figure 278 EGTR_FCTDGTDLAI0/ OPM/ OPM/ CLC_DIAG_LV

Calculation of the NOx Sensor specific variables

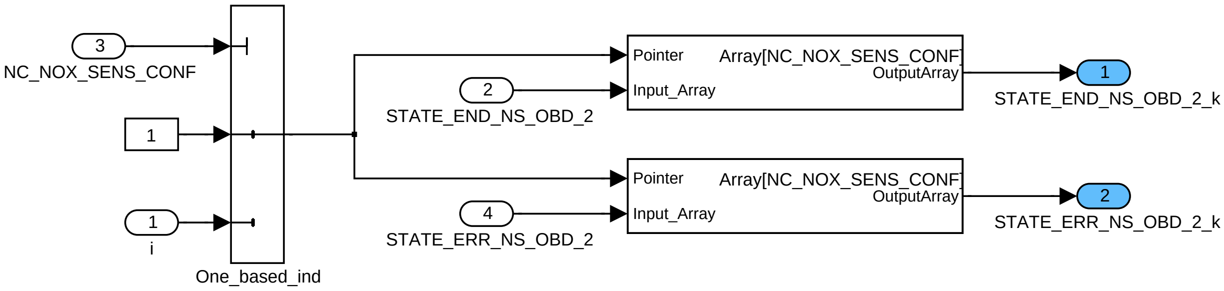


Figure 279 EGTR_FCTDGTDLAI0/ OPM/ OPM/ CLC_DIAG_LV/ CLC_1_NS_CONF

Calculation of the inhibition flag

Figure 280 EGTR_FCTDGTDLAI0/ OPM/ OPM/ CLC_DIAG_LV/ CLC_2_INH

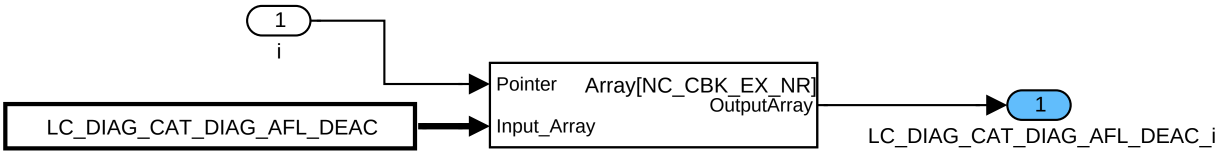


Figure 281 EGTR_FCTDGTDLAI0/ OPM/ OPM/ CLC_DIAG_LV/ CLC_2_INH/ CLC_2_END

Check of the end bits of other relevant diagnosis

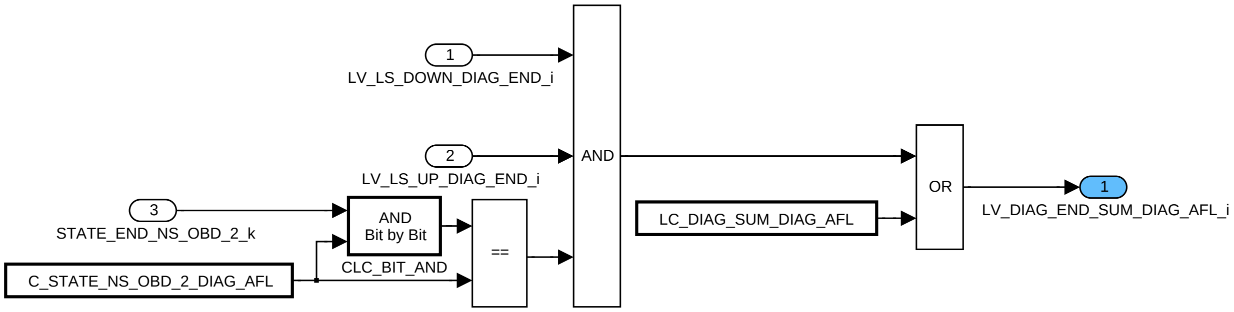


Figure 282 EGTR_FCTDGTDLAI0/ OPM/ OPM/ CLC_DIAG_LV/ CLC_3_END

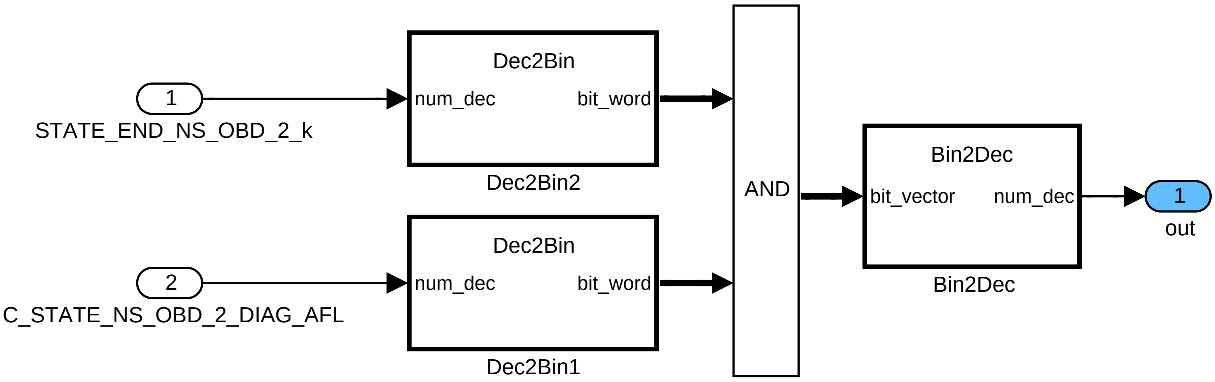


Figure 283 EGTR_FCTDGTDLAI0/ OPM/ OPM/ CLC_DIAG_LV/ CLC_3_END/ CLC_BIT_AND

Check of the error bits of other relevant diagnosis

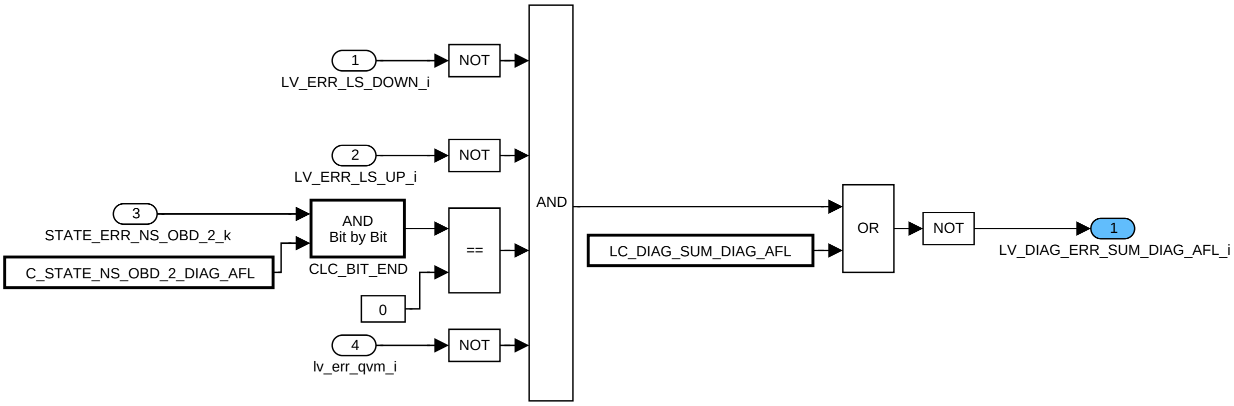


Figure 284 EGTR_FCTDGTDLAI0/ OPM/ OPM/ CLC_DIAG_LV/ CLC_4_ERR

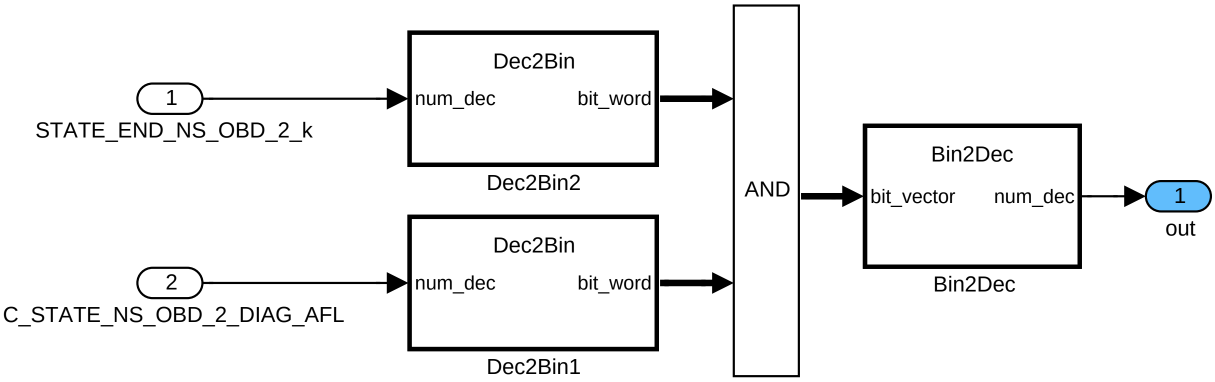


Figure 285 EGTR_FCTDGTDLAI0/ OPM/ OPM/ CLC_DIAG_LV/ CLC_4_ERR/ CLC_BIT_END

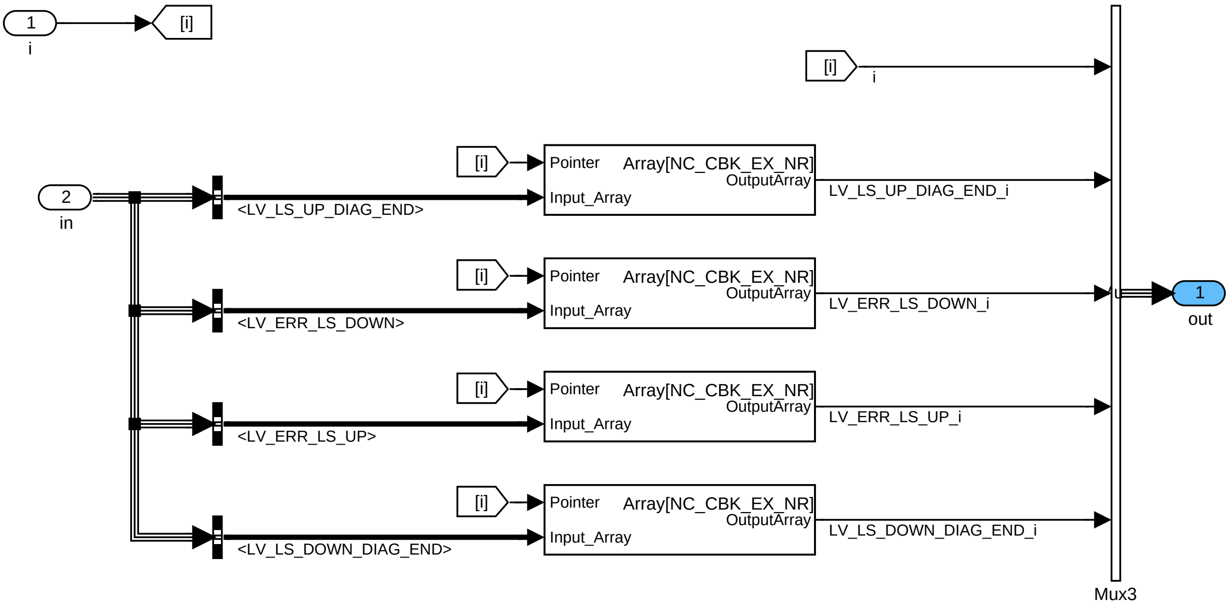


Figure 286 EGTR_FCTDGTDLAI0/ OPM/ OPM/ read_vector

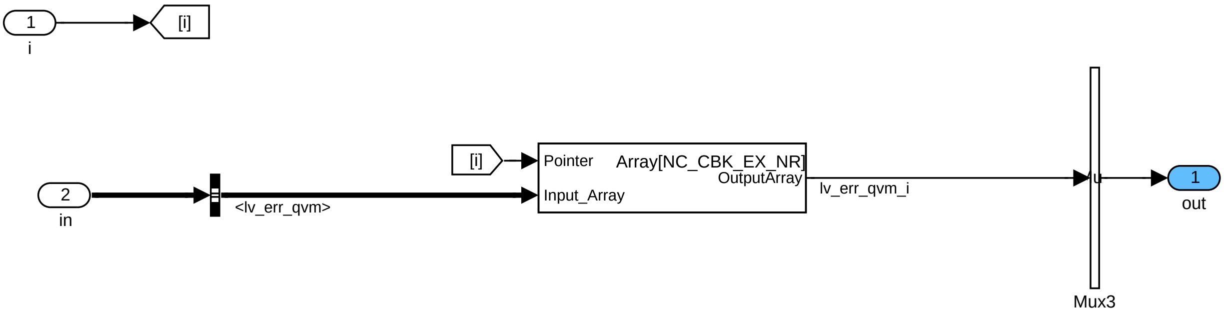


Figure 287 EGTR_FCTDGTDLAI0/ OPM/ OPM/ read_vector1

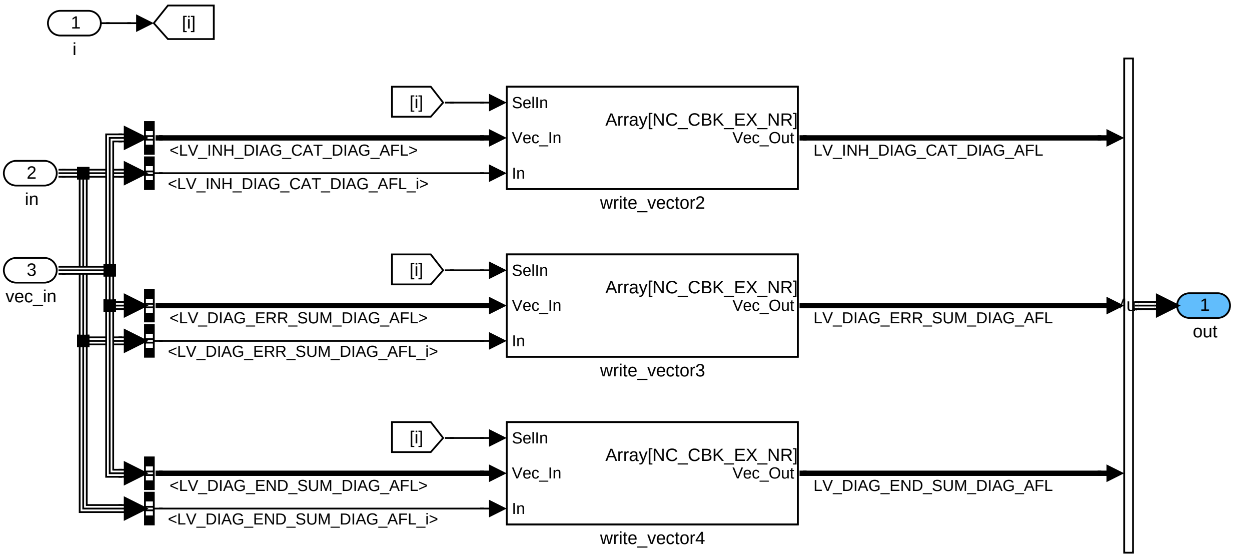


Figure 288 EGTR_FCTDGTDLAI0/ OPM/ OPM/ write_vector

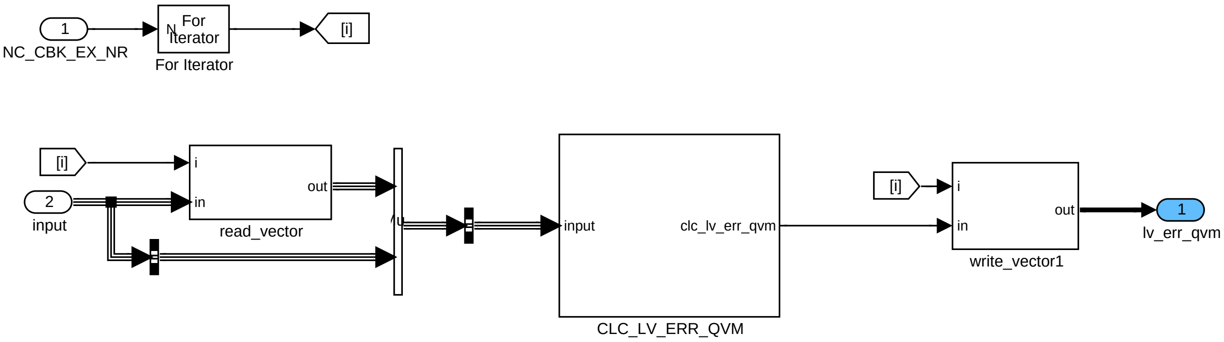


Figure 289 EGTR_FCTDGTDLAI0/ OPM/ QVM

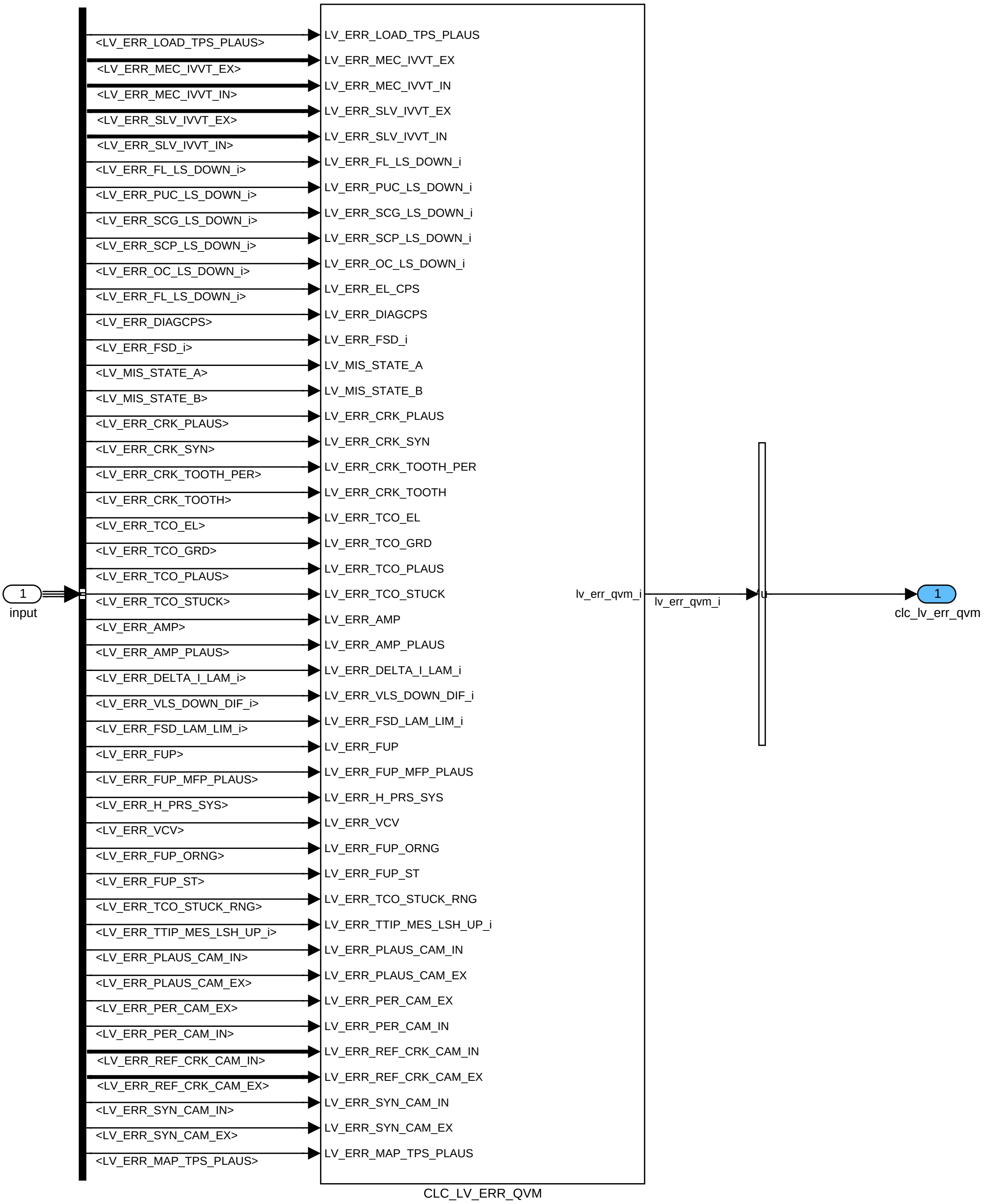


Figure 290 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM

Cross Interferences Matrix

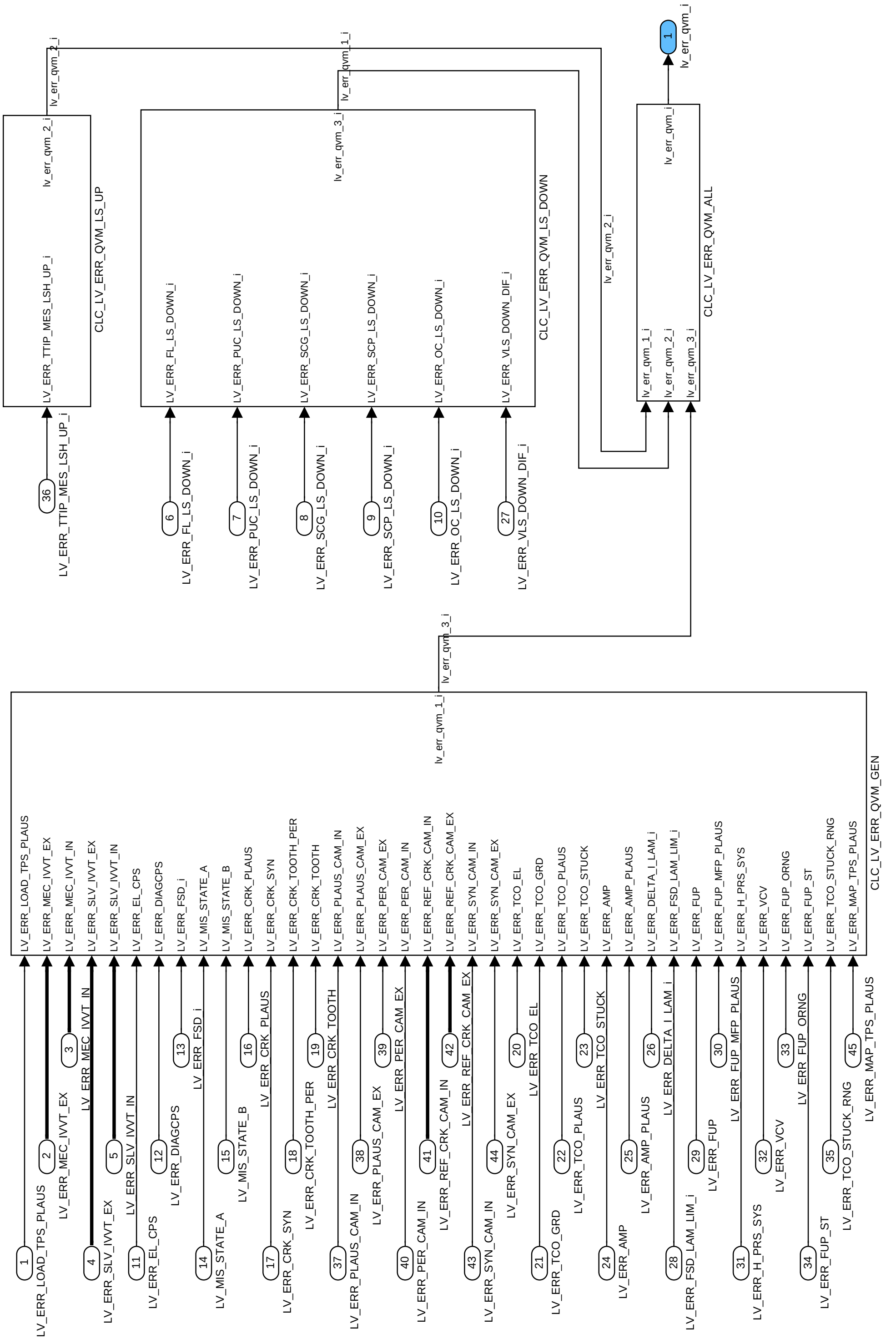


Figure 291 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM

Summary flag of Cross Interferences Matrix

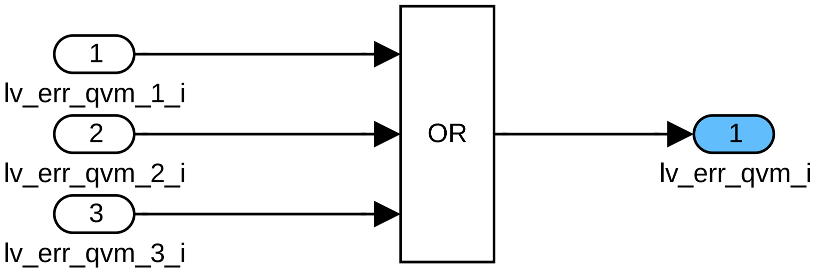


Figure 292 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_ALL

Cross Interferences Matrix – General Flags

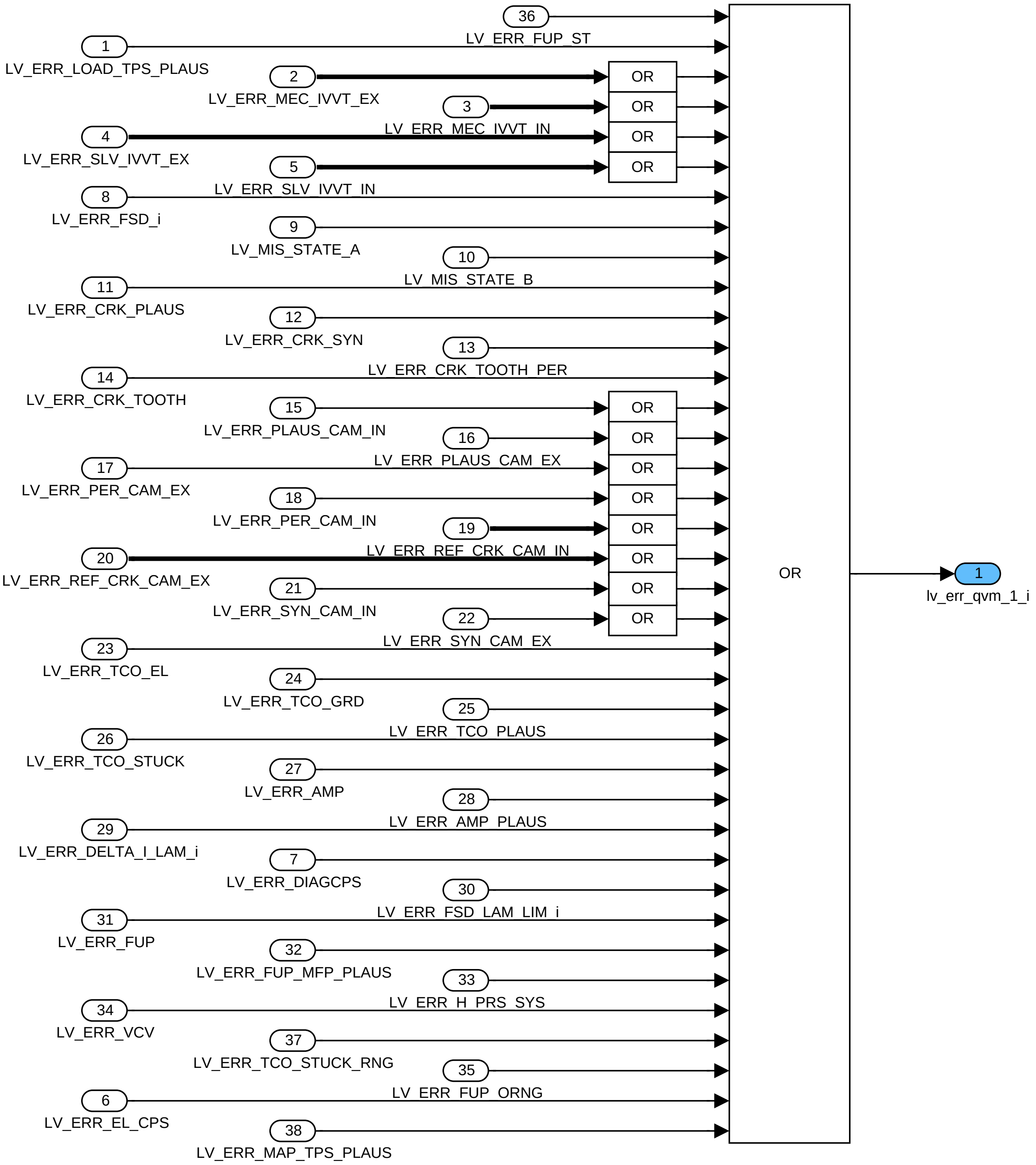


Figure 293 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN

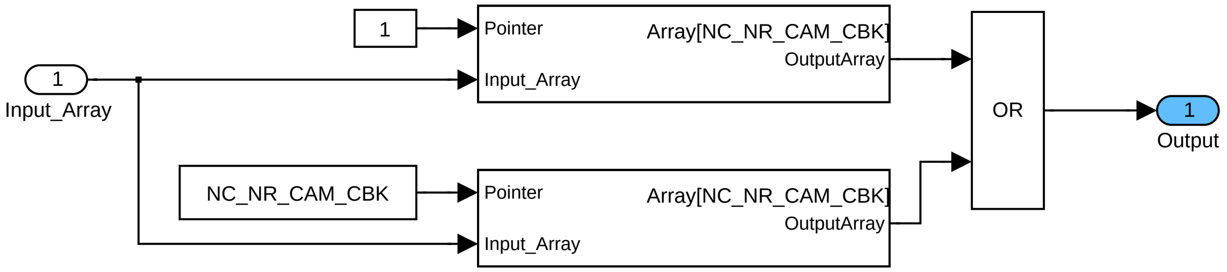


Figure 294 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR

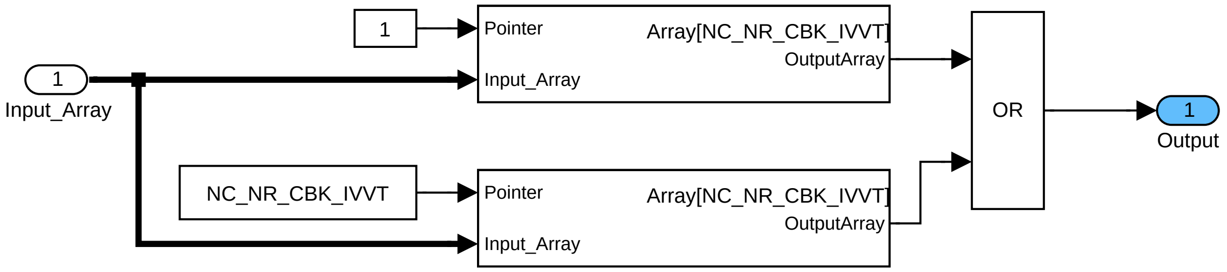


Figure 295 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR1

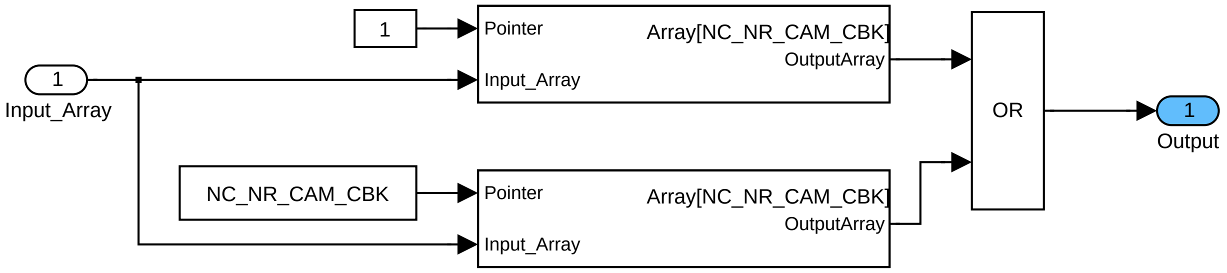


Figure 296 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR10

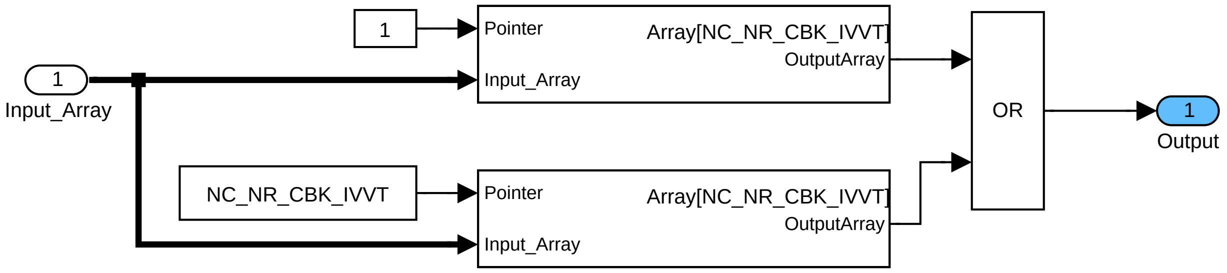


Figure 297 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR11

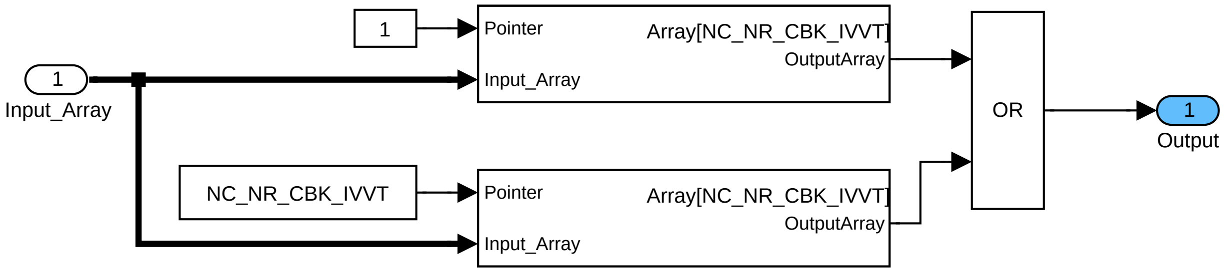


Figure 298 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR2

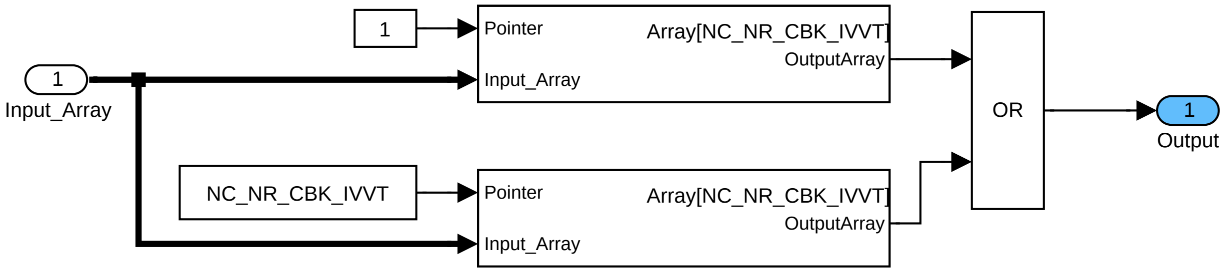


Figure 299 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR3

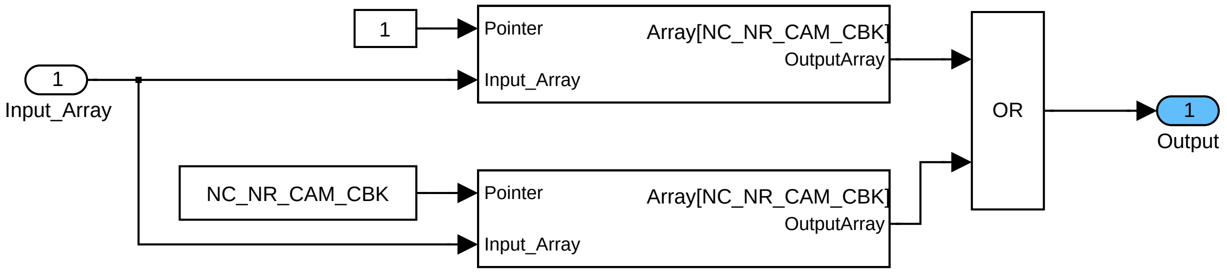


Figure 300 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR4

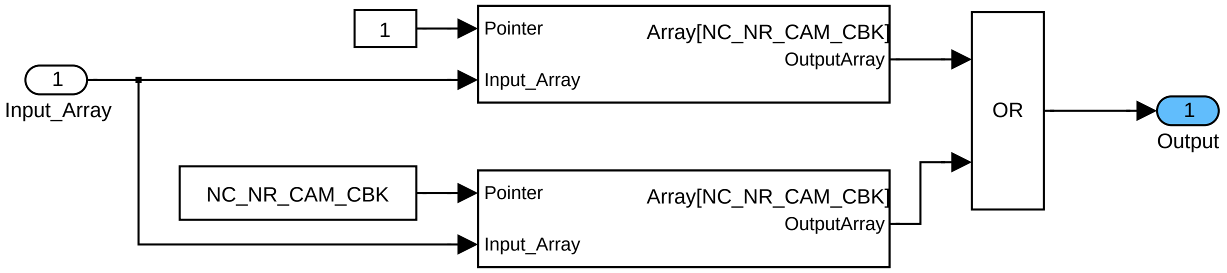


Figure 301 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR5

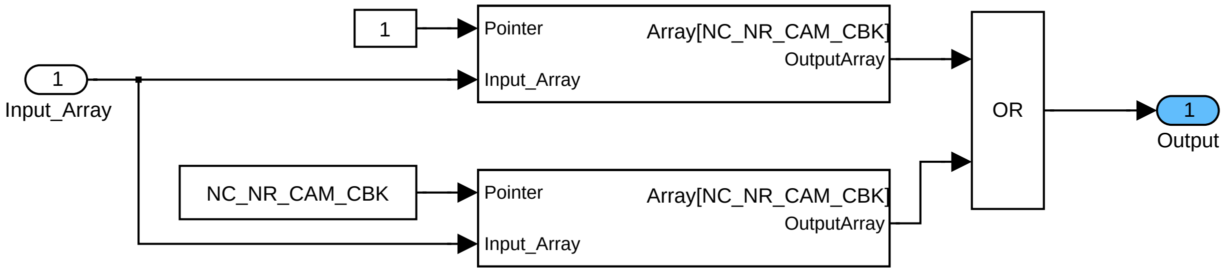


Figure 302 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR6

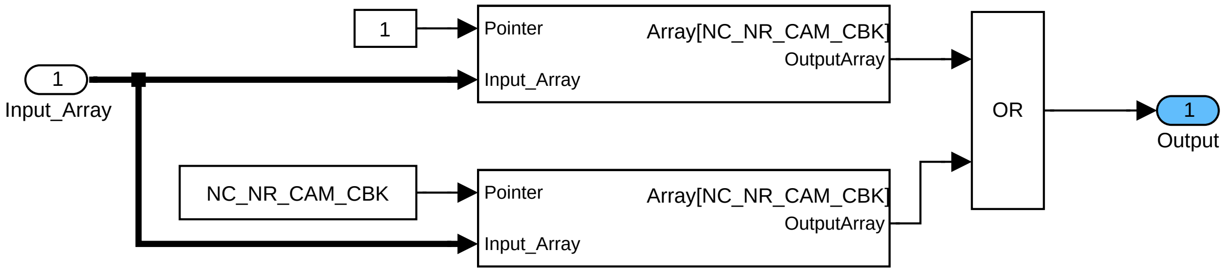


Figure 303 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR7

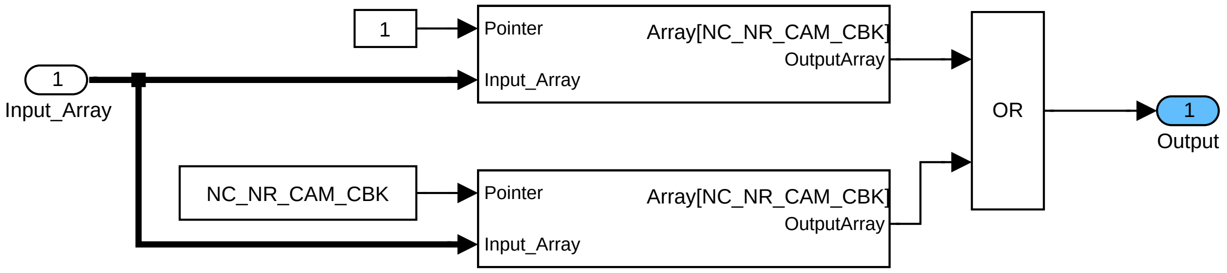


Figure 304 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR8

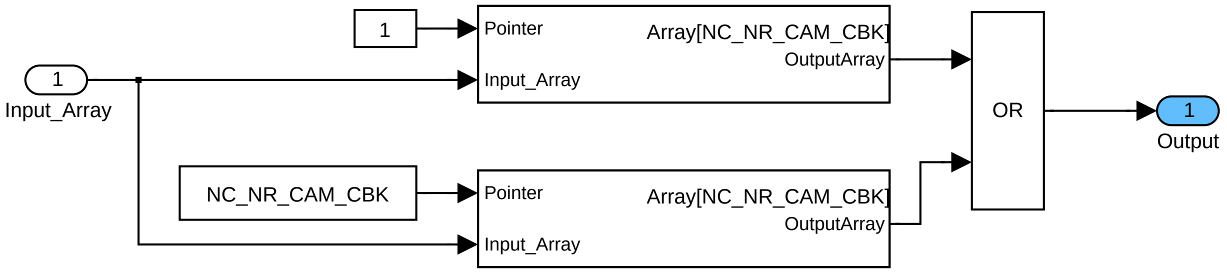


Figure 305 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_GEN/ OR9

Cross Interferences Matrix – Flags concerning downstream linear Sensor, not included in LV_ERR_LS_DOWN

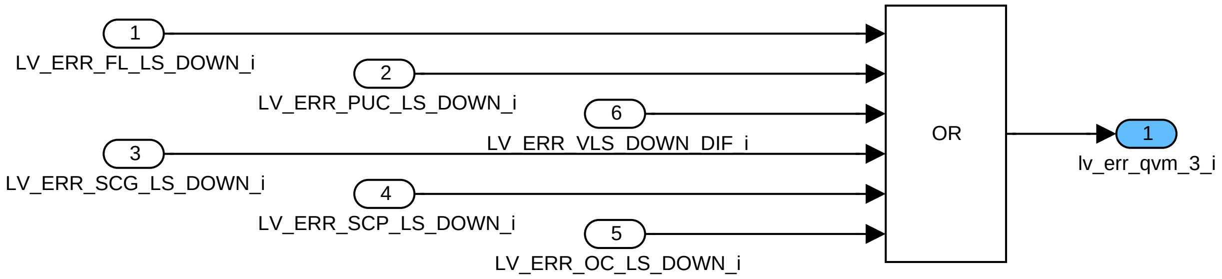


Figure 306 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_LS_DOWN

Cross Interferences Matrix – Flags concerning upstream linear Sensor, not included in LV_ERR_LS_UP

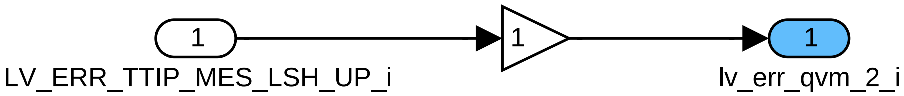


Figure 307 EGTR_FCTDGTDLAI0/ OPM/ QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM/ CLC_LV_ERR_QVM_LS_UP

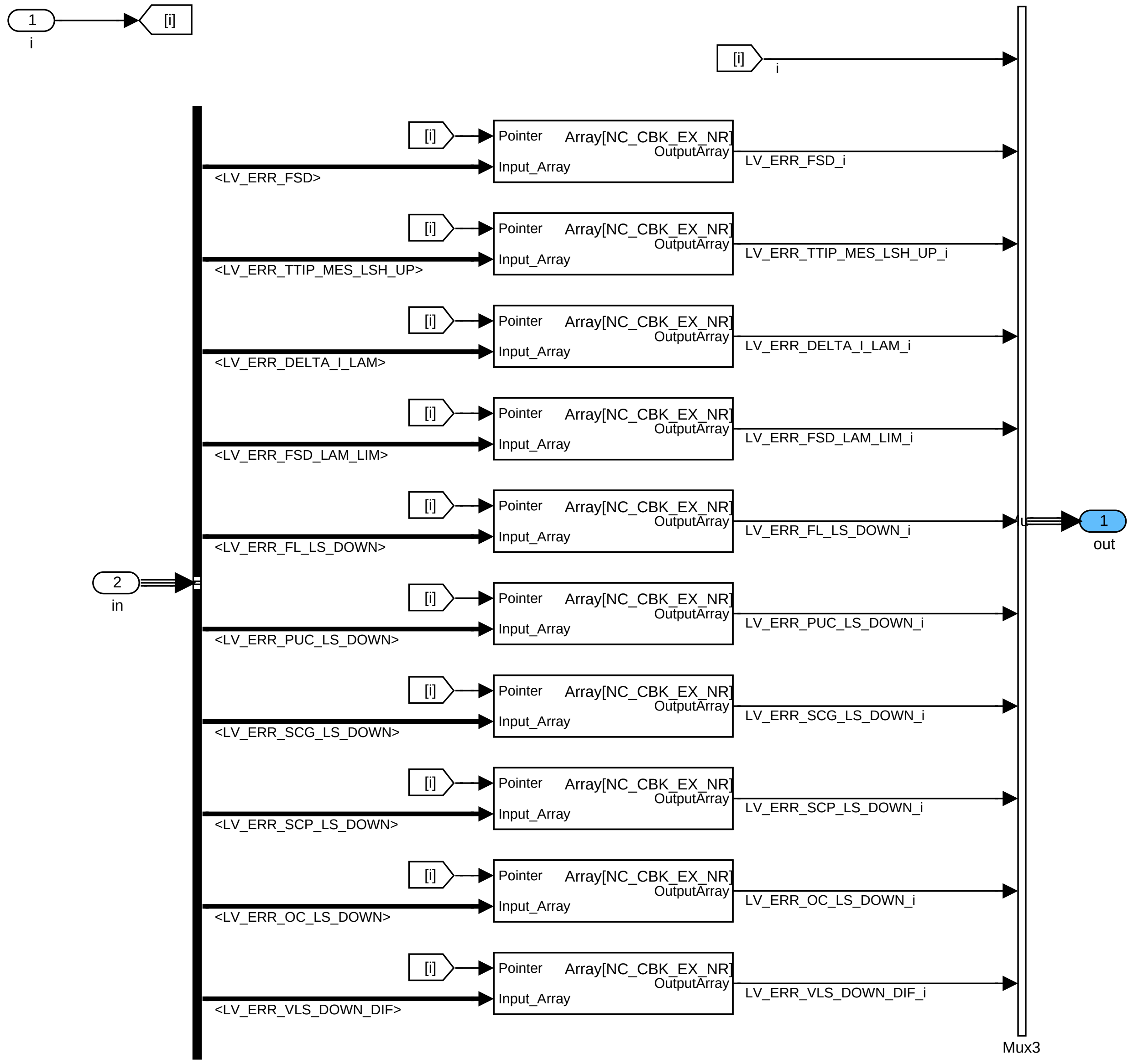


Figure 308 EGTR_FCTDGTDLAI0/ OPM/ QVM/ read_vector

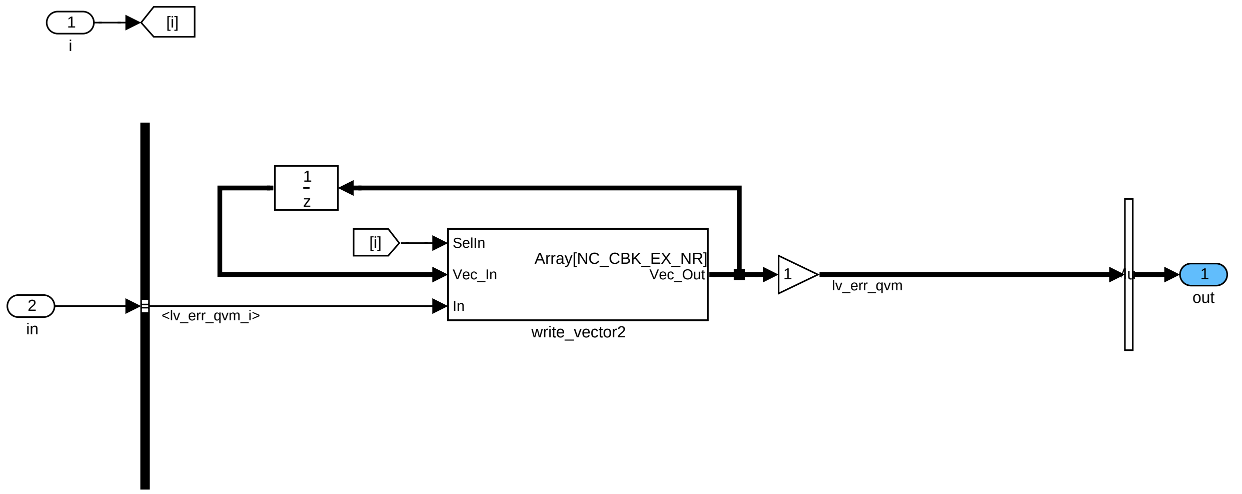


Figure 309 EGTR_FCTDGTDLAI0/ OPM/ QVM/ write_vector1

18.21.1.3 INITIALIZATION AT RESET

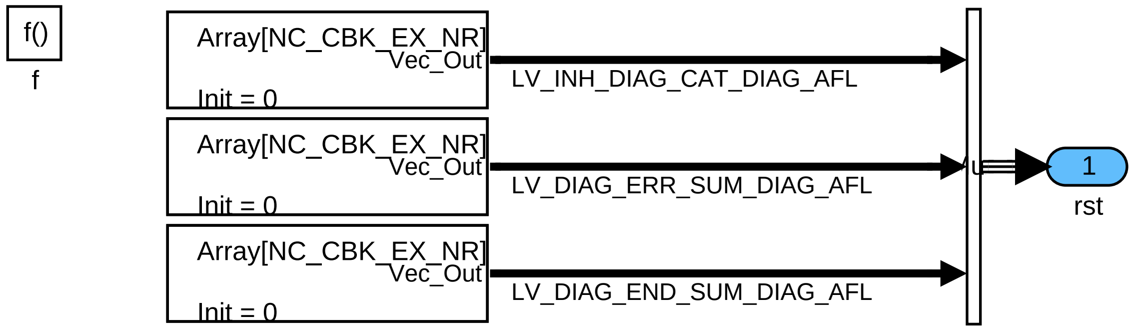


Figure 310 EGTR_FCTDGTDLAI0/ RST

18.22 Cus adap module: EGTR

18.22.1 Outputs for BMW functions which are not defined as EGTR exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_desu_fett	O	0...1H	0...1	1	[-]
Request of a fast desulfation (instead of an alternating desulfation					
B_desu_puls	O/V	0...1H	0...1	1	[-]
Flag to be passed to the lambda stimulation coordination					
B_kathom	O/V	0...1H	0...1	1	[-]
Request of lambda = 1 operation mode					
B_rgnkat	O/V	0...1H	0...1	1	[-]
Logical value for regeneration phase request					
B_rgnkat_hla1	O/V	0...1H	0...1	1	[-]
Request regeneration lambda = 1					
D_la_sa1	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
lambda controller set point shift					
D_la_sa2	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
lambda controller set point shift					
La_desu_kh1	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda setpoint during catalyst heating for desulfation bank 1					
La_desu_kh2	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda setpoint during catalyst heating for desulfation bank 2					
La_desu1	O	0...FFFFH	0...15.99975	0.2441e-3	[-]
Absolut lambda setpoint for lambda pulsation, desulfation bank1					
La_desu2	O	0...FFFFH	0...15.99975	0.2441e-3	[-]
Absolut lambda setpoint for lambda pulsation, desulfation bank2					
La_es	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda setpoint for desulfation					
La_puls1	O/V	80...7FH	-0.125...0.12402	0.9766e-3	[-]
Lambda setpoint for forced stimulation					
La_puls2	O/V	80...7FH	-0.125...0.12402	0.9766e-3	[-]
Lambda setpoint for forced stimulation					
La_rgn1	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
lambda setpoint for catalyst regeneration					
La_rgn2	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
lambda setpoint for catalyst regeneration					
LAMB_PLS_EXT_CUS[NC_CBK_EX_NR]	O/V	F800...7FFH	-0.125...0.12493	0.061e-3	[-]
lambda set point deviation from external source					
LV_PUC_LOCK_TNT	O/V	0...1H	0...1	1	[-]
LV suppression of PUC due to high Nox trap temperature					

Input data:

B_vb_sa_nk	Dla_soll_puls[NC_CBK_EX_NR]	ECU_STATE	La_soll_stat1
La_soll_stat2	La_soll1	La_soll2	LAMB_PLS_O2L_OSC[NC_CBK_EX_NR]
LAMB_PULS_SO2P[NC_CBK_EX_NR]	LAMB_RGN[NC_CBK_EX_NR]	LAMB_SO2P[NC_CBK_EX_NR]	LAMB_SP_CH_SO2P[NC_CBK_EX_NR]
LAMB_SP_DELTA_CAT_PURGE[NC_CBK_EX_NR]	LAMB_SP_DIAG_AFL[NC_CBK_EX_NR]	LV_DIAG_AFL_REQ	LV_NT_AFS_REQ
LV_RGN_REQ_NTLD_AFS	LV_SO2P_FAST_REQ	LV_SO2P_LAMB_PULS	LV_SO2P_REQ
LV_SO2P_REQ	NC_CBK_EX_NR	TQ_ADD_SO2P	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit power latch phase :

0.0 :	La_puls1 / 2, D_la_sa1/2, La_desu1/2
0:	B_kathom, B_rgnkat_hla1, B_desu_fett, B_desu_puls, B_rgnkat
1.0:	La_rgn1/2, La_desu_kh1/2, La_es

Recurrence :	100 ms	La_es
	10 ms	B_rgnkat, La_rgn1, La_rgn2, B_kathom, B_rgnkat_hla1
	20ms	D_la_sa1, D_la_sa2, B_desu_fett, B_desu_puls
	seg	La_puls1 / 2, La_desu1/2, La_desu_kh1/2

Activation: every engine state, except at power latch phase

Deactivation: at power latch phase

Values at deactivation: 0, except B_rgnkat, B_kathom,B_rgnkat_hla1=1

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
B_rgnkat          =    LV_RGN_NT_REQ OR LV_DIAG_AFL_REQ

IF (LV_RGN_NT_REQ == 0 AND LV_DIAG_AFL_REQ ==1)
THEN %lambda setpoint for TWC lean diagnosis after regeneration
    La_rgn1 = LAMB_SP_DIAG_AFL[1]
    La_rgn2 = LAMB_SP_DIAG_AFL[2]
ELSE %lambda setpoint for regeneration
    La_rgn1 = LAMB_RGN[1]
    La_rgn2 = LAMB_RGN[2]
ENDIF

La_es             =    LAMB_SO2P[1]
D_la_sa1          =    LAMB_SP_DELTA_CAT_PURGE[1]
D_la_sa2          =    LAMB_SP_DELTA_CAT_PURGE[2]
La_puls1          =    LAMB_PLS_O2L_OSC[1]
La_puls2          =    LAMB_PLS_O2L_OSC[2]
B_kathom          =    LV_NT_AFS_REQ
B_desu_fett       =    LV_SO2P_FAST_REQ       AND       LV_SO2P_REQ
La_desu1          =    1 + LAMB_PULS_SO2P[1]
```

La_desu2 = 1 + LAMB_PULS_SO2P[2]
B_desu_puls = LV_SO2P_LAMB_PULS
La_desu_kh1 = LAMB_SP_CH_SO2P[1]
La_desu_kh2 = LAMB_SP_CH_SO2P[2]
B_rgnkat_hla1 = LV_RGN_REQ_NTLD_AFS

18.22.2 Outputs for SV functions which are defined as BMW exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: LAMB_PLS_EXT_CUS[NC_CBK_EX_NR] = 0

LV_PUC_LOCK_TNT = 0

Recurrence : **200ms:** LV_PUC_LOCK_TNT

Segment synchronous: LAMB_PLS_EXT[NC_CBK_EX_NR] **(synchronised with the update of Dla_soll_puls):**

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

LV_PUC_LOCK_TNT = B_vb_sa_nk

LAMB_PLS_EXT_CUS[1] = Dla_soll_puls[1] **(synchronised with the update of Dla_soll_puls)**

LAMB_PLS_EXT_CUS[2] = Dla_soll_puls[2] **(synchronised with the update of Dla_soll_puls)**

19 Engine lubrication

Table of Contents

Index

19.1 Acquisition of oil pressure

19.1.1 Acquisition of oil pressure switch

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_POIL_SWI	O/V	0...1H	0...1	1	[-]
logical variable oil pressure switch (1= oil pressure o.k.)					
T_POIL_SWI	V	0...FFH	0...25.5	0.1	[s]
timer for delayed oil switch					
LV_BIOS_POIL_SWI	O/V	0...1H	0...1	1	[-]
Logical variable for non debounced oil pressure switch signal					
LV_POIL_SWI_SUB	-	0...1H	0...1	1	[-]
Logical variable for oil pressure switch or oil pressure sensor					
POIL	O/V	B1E0...4E10H	-20...20	1.0004e-3	[bar]
Oil pressure (Relative Sensor)					

Input data:

LV_IGK	T_PWL	C_T_MIN_PWL	LV_POIL_SWI_CUS
AC_VEH_LGT_TCS	AC_VEH_TRV_TCS	LV_ERR_SENS_POIL	C_V_POIL_MIN_DIAG
C_V_POIL_MAX_DIAG	V_POIL	ECU_STATE	

Import action:

ACTION_INFR_GetPoilSwi(OUT <lv_bios_poil_swi>**)**
This action reads the input pin of the "POIL Switch"

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_POIL_CTL_ENA	1	0...1H	0...1	1	[-]
Boolean to enable oil pressure control					
C_T_POIL_SWI_BAS	1	0...FFH	0...25.5	0.1	[s]
Basis time delay for oil pressure switch					
C_T_POIL_SWI_LGT	1	0...FFH	0...25.5	0.1	[s]
time delay for oil pressure switch due to LGT					
C_T_POIL_SWI_TRV	1	0...FFH	0...25.5	0.1	[s]
time delay for oil pressure switch due to TRV					
C_POIL_AC_VEH_LGT_TCS	1	8010...7FF0H	-51.175...51.175	0.0015625	[m/s²]
threshold for longitudinal acceleration for oil pressure switch					
C_POIL_AC_VEH_TRV_TCS	1	8010...7FF0H	-51.175...51.175	0.0015625	[m/s²]
threshold for transversal acceleration for oil pressure switch					
C_POIL_ERR_SUB	1	0...4E20H	0...20	0.001	[bar]
Substitute value of oil pressure in case of an sensor error					
IP_POIL_V_POIL_REL	8	31E0...CE20H	-20...20	0.001	[bar]
LDP_V_POIL_IP_POIL_V_POIL_REL	8	0...3FFH	0...4.99511	4.8828e-3	[V]
Oil pressure sensor linearisation table for relative sensor					

FUNCTION DESCRIPTION:

General information:

In case of oil pressure is below a minimum threshold the oil pressure switch switches from ground to open circuit (warning lamp on).

To avoid malfunction at high longitudinal or transversal accelerations the delay timer T_POIL_SWI is added.

The oil pressure switch (LV_POIL_SWI) has to be monitored also during power-latch-phase (due to oil pressure switch diagnosis).

Application conditions:

Initialisation: *at reset* *LV_POIL_SWI = 0*
T_POIL_SWI = C_T_POIL_SWI_BAS

Recurrence: 100 ms

Activation: at LV_IGK = 1

Deactivation: -

Formula section:

```
If      LC_POIL_CTL_ENA = 0
then    ACTION_INFR_GetPoilSwi(OUT <LV_BIOS_POIL_SWI>)
        LV_POIL_SWI_SUB = LV_BIOS_POIL_SWI
else    LV_POIL_SWI_SUB = LV_POIL_SWI_CUS
endif

If   LV_IGK = 1
Then    If      LV_POIL_SWI_SUB = 1
        Then    LV_POIL_SWI = 1
                T_POIL_SWI = C_T_POIL_SWI_BAS
        Else    If      AC_VEH_LGT_TCS < C_POIL_AC_VEH_LGT_TCS
                Then    T_POIL_SWI = C_T_POIL_SWI_LGT
                Else    If      abs(AC_VEH_TRV_TCS)
                                            > abs(C_POIL_AC_VEH_TRV_TCS)
                        Then    T_POIL_SWI = C_T_POIL_SWI_TRV
                        Else    If      T_POIL_SWI = 0
                                Then    LV_POIL_SWI = 0
                                Else    decrement T_POIL_SWI
                                Endif
                        Endif
                Endif
        Endif
```

```
Else    If      T_PWL < C_T_MIN_PWL
        Then    LV_POIL_SWI = LV_POIL_SWI_SUB
        Endif
Endif
```

19.1.2 Acquisition of oil pressure sensor

FUNCTION DESCRIPTION:

General information:

The sensor voltage signal V_POIL is converted via a linearisation table into the oil pressure POIL. In case of an oil sensor error a subsitude value can be used.

Application conditions:

Initialisation: at transition LV_IGK 0 –> 1 **or** reset

```
If      V_POIL < C_V_POIL_MIN_DIAG    or
        V_POIL > C_V_POIL_MAX_DIAG
then    POIL = C_POIL_ERR_SUB
else    POIL = IP_POIL_V_POIL_REL
endif
```

Recurrence: 10 ms

Activation: ECU_STATE <> PWL **and** LC_POIL_CTL_ENA = 1

Deactivation:

```
if      ECU_STATE = PWL   or   LC_POIL_CTL_ENA = 0
then    POIL = POIL(n-1)
endif
```

Formula section:

```
If      LV_ERR_SENS_POIL = 0
then    POIL = IP_POIL_V_POIL_REL
else    POIL = C_POIL_ERR_SUB
endif
```

19.2 ENLU - Requirements to Infrastructure interface

Export actions:

ACTION_INFR_GetVPoilSens(OUT <v_poil>**)**
This action gets the Oil pressure sensor voltage
ACTION_INFR_GetPoilSwi(OUT <lv_bios_poil_swi>**)**
This action reads the input pin of the "POIL Switch"
ACTION_INFR_GetElDiagPoil_dr(OUT <Cdn_diag_Poil_dr>, OUT <Err_diag_Poil_dr>**)**
This action reads the failure and condition information for a symptom of the oil pressure actuator power stage.
ACTION_INFR_SetFrqPwmPoilDr(IN <frq_poil_pwm**>)**
This action sets the frequency of the oil pressure actuator PWM
ACTION_INFR_SetPwmPoilDr(IN <poil_pwm**>)**
This action sets the duty cycle of the oil pressure actuator PWM

Description for actions:

ACTION_INFR_GetVPoilSens(OUT <v_poil>**)**					
This action gets the Oil pressure sensor voltage					
Name	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
v_poil	OUT	0...3FFH	0...4.9951118757	0.0048828125	V
Oil pressure sensor raw acquisition					

ACTION_INFR_GetPoilSwi(OUT <lv_bios_poil_swi>**)**					
This action reads the input pin of the "POIL Switch"					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
lv_bios_poil_swi	OUT	0...1H	0…1	1	-
Input pin of the "POIL Switch" (0= oil pressure failed; 1= oil pressure ok)					

ACTION_INFR_GetElDiagPoil_dr(OUT <Cdn_diag_Poil_dr>, OUT <Err_diag_Poil_dr>)					
This action reads the failure and condition information for a symptom of oil pressure actuator power stage. The readout of the power stage is performed autonomous and the information is gathered. When calling this action the information inside the infrastructure is reset after returning the OUT parameters.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Cdn_diag_poil_dr	OUT	0...7H	0...7	1	-
Diagnosis condition for symptoms of POIL_DR: bit 0: Set, if diagnosis condition for symptom SCP (SYM_0) is fulfilled bit 1: Set, if diagnosis condition for symptom SCG (SYM_1) is fulfilled bit 2: Set, if diagnosis condition for symptom OC (SYM_2) is fulfilled (all combinations are possible)					
Err_diag_poil_dr	OUT	0...7H	0...7	1	-
Raw value of error symptom: bit 0: Set, if raw value of error symptom SCP (SYM_0) is set bit 1: Set, if raw value of error symptom SCG (SYM_1) is set bit 2: Set, if raw value of error symptom OC (SYM_2) is set (all combinations are possible)					
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

ACTION_INFR_SetFrqPwmPoilDr(IN <frq_poil_pwm**>)**					
This action sets the frequency of the oil pressure actuator PWM					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

<table>
<tr><td>frq_poil_pwm</td><td>IN</td><td>0A...FFH</td><td>10...255</td><td>1</td><td>[Hz]</td></tr>
<tr><td colspan="6">Frequency of POIL_PWM</td></tr>
</table>

<table>
<tr><td colspan="6">**ACTION_INFR_SetPwmPoilDr(IN <**poil_pwm**>)**</td></tr>
<tr><td colspan="6">This action sets the duty cycle of the oil pressure actuator PWM</td></tr>
<tr><td>Name</td><td>Type</td><td>Hex. Limits</td><td>Phys. Limits</td><td>Resol.</td><td>Unit</td></tr>
<tr><td>poil_pwm</td><td>IN</td><td>0...FFFFH</td><td>0...99.99847</td><td>1.5259e-3</td><td>[%]</td></tr>
<tr><td colspan="6">Duty cycle of POIL_PWM</td></tr>
</table>

FUNCTION DESCRIPTION:

General information:

ACTION_INFR_GetVPoilSens()

Oil pressure sensor raw acquisition.

ACTION_INFR_GetPoilSwi()

This action reads the input pin of the "POIL Switch". Logical variable oil pressure switch (1= oil pressure o.k.).

ACTION_INFR_GetElDiagPoil_dr()

Returns the result of the electrical diagnosis of the oil pressure actuator power stage.

The device readout is performed autonomous by the Infrastructure each 10ms.
The error informations of every symptom is gathered in the infrastructure (or-ed symptom) until the application reads out the information by calling ACTION_INFR_GetElDiagPoil_dr().
After having read out the information by calling ACTION_INFR_GetElDiagPoil_dr(), the data inside the infrastructure is reset. Resetting of Cdn_diag_poil_dr avoids ambigious results in case of too short calling reccurrence of ACTION_INFR_GetElDiagPoil_dr(): Reset Cdn_diag_poil_dr indicates that the gathering of the information is not completely finished.

ACTION_INFR_SetFrqPwmPoilDr()
The oil pressure actuator is controlled by a pulse width modulation control signal. To counter a rough-running actuator in case of fouling, the frequency of the PWM can be changed by the action SetFrqPwmPoilDr. The frequency information is transmitted to the BSW by the variable *frq_poil_pwm*.

ACTION_INFR_SetPwmPoilDr()
The oil pressure actuator is controlled by a PWM control signal. The extent of control is done by the duty cycle via the variable *poil_pwm*.

Requirements for ACTION_INFR_GetVPoilSens():

<table>
<tr><td colspan="6">**Data acquisition**</td></tr>
<tr><td>**Parameter**</td><td>**Absolute precision**</td><td>**Relative precision**</td><td>**Resolution**</td><td>**Coincidence with parameter**</td><td>**Comment**</td></tr>
</table>

v_poil	Not relevant	0,1 %	0.0048828125 V	Not relevant	Not relevant

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR__GetPoilSwi:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
lv_bios_poil_swi	Not relevant	Not relevant	1	Not relevant	Not relevant

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_GetElDiagPoil_dr:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cdn_diag_poil_dr	-	-	<bit coded>	Err_diag_poil_dr	Diagnosis condition for each symptom **bit 0:** diagnosis condition for symptom SCP (SYM_0) **bit 1:** diagnosis condition for symptom SCG (SYM_1) **bit 2:** diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
Err_diag_poil_dr	-	-	<bit coded>	Cdn_diag_poil_dr	Bitcoded result of each symptom **bit 0:** raw value of error symptom SCP (SYM_0) **bit 1:** raw value of error symptom SCG (SYM_1) **bit 2:** raw value of error symptom OC (SYM_2) The relevant bit is set, if the raw error has been detected.

Diagnosis: ACTION_INFR_GetElDiagPoil_dr returns the electric diagnosis for the oil pressure actuator power stage.

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_SetFrqPwmPoilDr:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
frq_poil_pwm	Not relevant	Not relevant	1 Hz	Not relevant	244 Hz at initialization

Diagnosis: no electrical diagnosis done here

Coincidence requirements:

Requirements for ACTION_INFR_SetPwmPoilDr:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
poil_pwm	Not relevant	Not relevant	1.5259e-3 %	Not relevant	0% at initialization

Diagnosis: no electrical diagnosis done here

Coincidence requirements:

19.3 Oil overtemperature detection

19.3.1 Oil overtemperature detection – Detection algorithm

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TOIL_THD_WARN_1	O/V	0...1H	0...1	1	-
TOIL > warning threshold 1					
LV_TOIL_THD_WARN_2	O/V	0...1H	0...1	1	-
TOIL > warning threshold 2					

Input data:

TOIL	LV_IGK		

FUNCTION DESCRIPTION:

General information:

There are two thresholds for detecting a TOIL overtemperature. In case TOIL exceeds the first threshold a statusflag is set and a checkcontrol-message is sent, indicating that the TOIL is within the “yellow area”. If TOIL exceeds the second threshold a statusflag is set and a checkcontrol-message is set, indicating that the TOIL is within the “red area”.
The statusflags are reset if TOIL < threshold.

Application conditions:

Initialisation:	LV_TOIL_THD_WARN_i	= 0	at reset
Recurrence:	1000ms		
Activation:	LV_IGK = 1		

Formula section:

Determination of LV_TOIL_THD_WARN_1:

```
IF      (TOIL                  >    C_TOIL_CHK_CTL_1)
THEN    LV_TOIL_THD_WARN_1     =    1
ELSE    LV_TOIL_THD_WARN_1     =    0
ENDIF
```

Determination of LV_TOIL_THD_WARN_2:

```
IF      (TOIL                  >    C_TOIL_CHK_CTL_2)
THEN    LV_TOIL_THD_WARN_2     =    1
ELSE    LV_TOIL_THD_WARN_2     =    0
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TOIL_CHK_CTL_1	1	0..FFH	-40...215	1	°C
warning threshold 1 for TOIL overtemperature (appl. suggestion: 155°C)					
C_TOIL_CHK_CTL_2	1	0...FFH	-40...215	1	°C
warning threshold 2 for TOIL overtemperature (appl. suggestion: 158°C)					

19.3.2 Oil overtemperature detection - Environmental data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TOIL_MAX_WARN	O/V/S	0...C8H	-40...160	1	°C
max. TOIL at last warning threshold 1					
VS_TOIL_MAX	O/V/S	0...FFH	0...255	1	km/h
VS at last warning threshold 1					
N_TOIL_MAX	O/V/S	0...1FE0H	0...8160	1	rpm
N at last warning threshold 1					
GEAR_TOIL_MAX	O/V/S	0...FFH	0..255	1	-
GEAR at last warning threshold 1					
TQI_AV_TOIL_MAX	O/V/S	8000...7FFFH	-1024...1023.97	0.03125	Nm
indicated engine torque at last warning threshold 1					
TAM_TOIL_MAX	O/V/S	0...FEH	-48...142.5	0.75	°C
ambient temperature at last warning threshold 1					
TCO_TOIL_MAX	O/V/S	0...FEH	-48...142.5	0.75	°C
coolant temperature at last warning threshold 1					
CTR_TOIL_MAX	O/V/S	0...FFH	0...255	1	-
event frequency counter					
PV_AV_TOIL_MAX	O/V/S	0...FFH	0...99.6	0.39	%
PV_AV at last warning threshold 1					
DIST_TOIL_MAX	O/V/S	0...FFFFH	0...524280	8	km
mileage at last warning threshold 1					
LV_CS_TOIL_MAX	O/V/S	0...1H	0...1	1	-
clutch switch detection at last warning threshold 1					
TOIL_THD_TOIL_MAX	O/V/S	0...C8H	-40...160	1	°C
warning threshold 1					

Input data:

GEAR	DIST_KWP	N	PV_AV
TAM	TCO	TOIL	TQI_AV
VS	LV_CS	LV_IGK	C_TOIL_CHK_CTL_1
LV_TOIL_THD_WARN_1			

General information:

This function contains the environmental data, which should be stored in the nv-memory in case of detected oil overtemperature.

Application conditions:

Initialisation: GEAR_TMP = 0 **at reset**

all xxx_TOIL_MAX – data restored out of NVMY

Recurrence: 1000 ms

Activation: LV_IGK = 1

Formula section:

Determination of LV_TOIL_THD_1:

```
IF        LV_TOIL_THD_WARN_1   =    1
THEN      CTR_TOIL_MAX     =    CTR_TOIL_MAX + 1        /* only once per trigger */
    IF        TOIL                      >    TOIL_MAX_WARN
    THEN      TOIL_MAX_WARN             =    TOIL
              VS_TOIL_MAX               =    VS
              N_TOIL_MAX                =    N
              GEAR_TOIL_MAX             =    GEAR_TMP
              TQI_AV_TOIL_MAX           =    TQI_AV
              TAM_TOIL_MAX              =    TAM
              TCO_TOIL_MAX              =    TCO
              DIST_TOIL_MAX             =    DIST_KWP
              PV_AV_TOIL_MAX            =    PV_AV
              LV_CS_TOIL_MAX            =    LV_CS
              TOIL_THD_TOIL_MAX         =    C_TOIL_CHK_CTL_1
    ENDIF
ENDIF
```

Determination of last used gear:

```
IF        (GEAR      >    0)          /* in case of HS-Gearbox, GEAR = 0 if CS is active */
THEN      GEAR_TMP   =    GEAR
ENDIF
```

19.4 Oil pressure sensor diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SENS_POIL	V/O	0...1H	0...1	1	[-]
Oil pressure sensor error present					
LV_CDN_DIAG_SENS_POIL	V/O	0...1H	0...1	1	[-]
Diagnosis condition of oil pressure sensor					
ERR_SYM_SENS_POIL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom of oil pressure sensor					
LV_END_DIAG_SENS_POIL	V/O	0...1H	0...1	1	[-]
End of diagnosis of oil pressure sensor					

Input data:

V_POIL	LV_IGK	LC_POIL_CTL_ENA	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_POIL_MAX_DIAG	1	0...3FFH	0...4.99511	4.883E-03	[V]
Maximum diagnostic value for the oil pressure sensor signal					
C_V_POIL_MIN_DIAG	1	0...3FFH	0...4.99511	4.883E-03	[V]
Minimum diagnostic value for the oil pressure sensor signal					
C_ABC_INC_POIL	1	0...FFH	0...255	1	[-]
Anti - bounce counter increment					
C_ABC_MAX_POIL	1	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter					

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis shall be to detect electrical faults as defined by OBD I requirements.
The signal of the oil pressure sensor on the A/D-input of the microcontroller is checked.

Description:

Error-symtoms are defined to this diagnosis function as following :

```
|_|_|_|_|
 | | | └─> Signal line short to battery voltage or 5 V power supply  (= SYM_0)
 | | └───> Signal line short to ground or open load                 (= SYM_1)
 | └─────> -                                                         (= SYM_2)
 └───────> -                                                         (= SYM_3)
```

Application conditions:

Initialization: all 0 at LV_IGK 0 -->1 **or** reset

Recurrence: 100ms

```
Activation:     If      LV_IGK = 1   and   LC_POIL_CTL_ENA = 1
                Then    LV_CDN_DIAG_SENS_POIL = 1
                Else    LV_CDN_DIAG_SENS_POIL = 0
                Endif
Deactivation:   LV_IGK = 0   or   LC_POIL_CTL_ENA = 0
```

Formula section:

Short circuit to VBatt or 5 V power supply:

```
If      V_POIL  >  C_V_POIL_MAX_DIAG
Then    ERR_SYM_SENS_POIL = SYM_0
```

Short circuit to ground or open load:

```
Else    If      V_POIL  <  C_V_POIL_MIN_DIAG
        Then    ERR_SYM_SENS_POIL = SYM_1
        Else    ERR_SYM_SENS_POIL = NO_SYM
        Endif
Endif
```

Calculation of present error and end of diagnosis:

LV_ERR_SENS_POIL and LV_END_DIAG_SENS_POIL is calculated by the error management.

19.5 Electrical diagnosis of oil pressure actuator

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_POIL_DR	O/V	0...1H	0...1	1	[-]
Engine-oil pressure valve powerstage error					
CDN_DIAG_POIL_DR	V	0...7H	0...7	1	-
Diagnosis condition for each symptom of POIL_DR bit 0: diagnosis condition for symptom SYM_0 bit 1: diagnosis condition for symptom SYM_1 bit 2: diagnosis condition for symptom SYM_2 bit 3: diagnosis condition for symptom SYM_3					
ERR_DIAG_POIL_DR	-	0...7H	0...7	1	-
Raw value of error symptom for POIL_DR (only parameter) bit 0: Set, if raw value of error symptom SCP (SYM_0) is set bit 1: Set, if raw value of error symptom SCG (SYM_1) is set bit 2: Set, if raw value of error symptom OC (SYM_2) is set (all combinations are possible)					
LV_ERR_POIL_DR_SCG	O/V	0...1H	0...1	1	[-]
Engine-oil pressure valve powerstage error SCG					
LV_CDN_DIAG_POIL_DR_SCG	V	0...1H	0...1	1	[-]
Boolean for engine-oil pressure valve powerstage error SCG diagnosis conditions					
LV_END_DIAG_POIL_DR_SCG	V	0...1H	0...1	1	[-]
Boolean for end of engine-oil pressure valve powerstage error SCG diagnosis					
ERR_SYM_POIL_DR_SCG	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for engine-oil pressure valve powerstage error SCG					
T_POIL_SWI_PWL	V/O	0...3EH	0...6.2	0.1	[s]
Timer for oil pressure inclination after engine stop					
LV_ERR_POIL_SWI	V/O/S	0...1H	0...1	1	[-]
Error oil pressure switch					
LV_CDN_DIAG_POIL_SWI	V/O	0...1H	0...1	1	[-]
Status of diagnosis oil pressure switch					
LV_END_DIAG_POIL_SWI	V/O	0...1H	0...1	1	[-]
End of diagnosis oil pressure switch					
ERR_SYM_POIL_SWI	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptoms oil pressure switch					

Note :

- The data ERR_SYM_POIL_DR, LV_END_DIAG_POIL_DR and LV_CDN_DIAG_POIL_DR are not present in the output data table, because they are not used by other functions. Despite they are provided by the error management and are always visible.

Input data:

ECU_STATE	LC_POIL_CTL_ENA	LC_POIL_CTL_ENA	LV_CDN_VB_OBD1
LV_ERR_SPI_MPS	LV_IGK	LV_IGK	LV_POIL_SWI
POIL_PWM	TOIL		

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>, OUT<LV_ERR_XX>)
ACTION_INFR_GetElDiagPoil_dr(OUT< Cdn_diag_Poil_dr>, OUT< Err_diag_Poil_dr >)

Note :

ACTION_INFR_GetElDia*g*Poil_dr() is defined in the IRS (Infrastructure requirement specification)

FUNCTION DESCRIPTION:

General information:

The oil pressure actuator is driven by the ECU via an output driver. The failure detection is done by ECU Hardware.
The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements.

Description:

After activation conditions are met, the diagnosis activation is delayed for NC_PSD_DLY_POIL_DR executions, to avoid the usage of wrong infrastructure information.

Error-symptoms and conditions: are defined to this diagnosis function as following

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to battery (SCP)        (= SYM_0)
 │ │ └───> Short circuit to ground (SCG)         (= SYM_1)
 │ └─────> Open circuit (OC)                     (= SYM_2)
 └───────> -                                     (= SYM_3)
```

Application conditions:

Initialisation: at transition LV_IGK 0 --> 1:

LV_ERR_POIL_DR,

ERR_SYM_POIL_DR,

LV_END_DIAG_POIL_DR

(according filter-type)

Set delay counter for NC_PSD_DLY_POIL_DR at reset:

CDN_DIAG_POIL_DR = 0,

LV_ERR_POIL_DR = 0 and

ERR_SYM_POIL_DR = 0,

LV_CDN_DIAG_POIL_DR = 0,

LV_END_DIAG_POIL_DR = 0,

(according filter-type)

Set delay counter for NC_PSD_DLY_POIL_DR

Recurrence: 100 ms

Activation: LV_IGK = 1 **and** LC_POIL_CTL_ENA = 1 **and** LV_ERR_SPI_MPS = 0

Deactivation: LV_IGK = 0 **or** LC_POIL_CTL_ENA = 0 **or** LV_ERR_SPI_MPS = 1

Formula section:

```
If      LV_IGK = 1                  and
        LV_CDN_VB_OBD1 = 1
Then
```

Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_POIL_DR and basic diagnosis conditions CDN_DIAG_POIL_DR) received from the infrastructure:

ACTION_INFR_GetElDiagPoil_dr (OUT<Cdn_diag_poil_dr>, OUT<Err_diag_poil_dr>)

Basic diagnosis conditions are set according infrastructure information: CDN_DIAG_POIL_DR
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_POIL_DR

```
        If   Activation conditions are met for the NC_PSD_DLY_POIL_DR recurrence of
             the diagnosis
        Then { Additional diagnosis conditions }
             If    C_PWM_POIL_DR_MIN_DIAG_SCP > POIL_PWM
             then  bit 0 of CDN_DIAG_POIL_DR = 0 {Diagnosis of SCP not possible }
             Endif
             If    POIL_PWM > C_PWM_POIL_DR_MAX_DIAG_SCG
             then  bit 1 of CDN_DIAG_POIL_DR  = 0 {Diagnosis of SCG not possible }
             Endif
             If    POIL_PWM < C_PWM_POIL_DR_MIN_DIAG_OC
             Or    POIL_PWM > C_PWM_POIL_DR_MAX_DIAG_SCG
             then  bit 2 of CDN_DIAG_POIL_DR  = 0 {Diagnosis of OC not possible }
             Endif
        Else CDN_DIAG_POIL_DR = 0
        Endif
Else
        CDN_DIAG_POIL_DR = 0
Endif
```

Failure filtering and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_POIL_DR and ERR_DIAG_POIL_DR.

ACTION_ERRM_FilterMulticondition (IN<POIL_DR>, IN<CDN_DIAG_POIL_DR>, IN<ERR_DIAG_POIL_DR>, IN<C_ABC_INC_POIL_DR>, IN<C_ABC_MAX_POIL_DR>, OUT<LV_ERR_POIL_DR>)
This algorithm determines:

ERR_DIAG_POIL_DR	(detected error symptom for POIL_DR diagnosis)
LV_ERR_POIL_DR	(Error flag for debounced error of POIL_DR)
LV_CDN_DIAG_POIL_DR	(Diagnosis condition information)
LV_END_DIAG_POIL_DR	(End of diagnosis information)

Configuration for diagnostic symptoms :

Diagnostic **POIL_DR**	**Symptom description**	**Symptom**	**Filter type**
Driver diagnosis of oil pressure actuator	SCP	*SYM_0*	*MPL_STD_INI*
	SCG	*SYM_1*	
	OC	*SYM_2*	

19.6 Electrical diagnosis of oil pressure actuator for shortcut to ground (SCG)

Import actions:

ACTION_ERRM_GetLvErr(IN<IDX_DIAG>, OUT<LV_ERR>)
Action that returns the status of the debounced failure
ACTION_ERRM_FilterSymptom(IN<IDX_DIAG>, IN<LV_CDN_DIAG>, IN<ERR_SYM>, IN<ABC_INC>, IN<ABC_DEC>, IN<ABC_MAX>, OUT<LV_ERR>)
This action computes the elementary anti-bounce filter

FUNCTION DESCRIPTION:

General information:

In case of an electrical failure caused by a shortcut to ground (SCG) a special failure - reaction and a different warning - lamp (EML) - handling is necessary (caused by low oil - pressure the engine speed will be limited at 3000 rpm). Due to the need of a separate C_ERR_CLAS for this symptom of LV_ERR_POIL_DR it will be copied on LV_ERR_POIL_DR_SCG.

Error-symptoms and conditions: are defined to this diagnosis function as following

```
|_|_|_|_|
 | | | └─> -                                        (= SYM_0)
 | | └───> Short circuit to ground (SCG)          (= SYM_1)
 | └─────> -                                        (= SYM_2)
 └───────> -                                        (= SYM_3)
```

Application conditions:

Initialisation: all output data according ABC configuration "**MEM**"

// LV_IGK 0->1 ***or*** *reset* ***or*** *at clearing error memory*

This action initialized the diagnostic result according the filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_ POIL_DR_SCG >, OUT<LV_ERR_POIL_DR_SCG>)

Recurrence: 100 ms

Activation: LV_IGK = 1 **and** LC_POIL_CTL_ENA = 1 **and** LV_ERR_SPI_MPS = 0

Deactivation: LV_IGK = 0 **or** LC_POIL_CTL_ENA = 0 **or** LV_ERR_SPI_MPS = 1

Formula section:

LV_CDN_DIAG_POIL_DR_SCG = LV_CDN_DIAG_POIL_DR

```
If      ERR_SYM_POIL_DR = SYM_1
Then    ERR_SYM_POIL_DR_SCG = SYM_1
Else    ERR_SYM_POIL_DR_SCG = NO_SYM
Endif
```

For failure and error management treatment the anti-bounce mechanism is called with the parameters LV_CDN_DIAG_POIL_DR_SCG and ERR_SYM_POIL_DR_SCG.

ACTION_ERRM_FilterSymptom (IN< POIL_DR_SCG >, IN< LV_CDN_DIAG_POIL_DR_SCG >,
IN< ERR_SYM_POIL_DR_SCG >, IN< C_ABC_INC_POIL_DR_SCG >, IN< C_ABC_MAX_POIL_DR_SCG >,
IN< 1 >, OUT< LV_ERR_POIL_DR_SCG >)

This algorithm determines LV_ERR_POIL_DR_SCG and LV_END_DIAG_POIL_DR_SCG and delivers the result to Error Management.

Configuration for diagnostic symptoms :

Diagnostic POIL_DR_SCG	Symptom description	Symptom	Filter type
Driver diagnosis of oil pressure actuator only for shortcut to plus	-	SYM_0	MEM
	SCG	SYM_1	
	-	SYM_2	
	-	SYM_3	

19.7 Oil pressure switch diagnosis

FUNCTION DESCRIPTION:

General information:

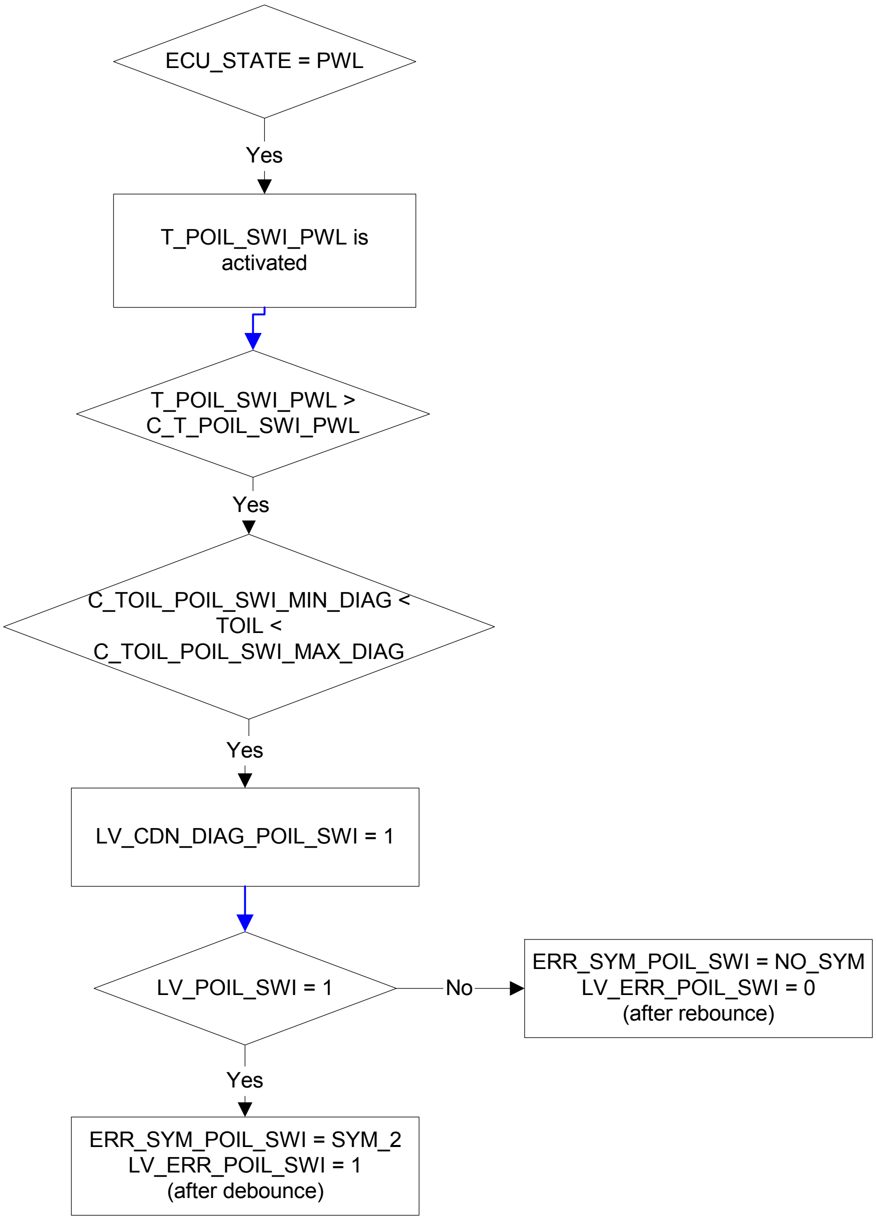
Due to the ECU recognizing the oil pressure as “ok” during “switch open”, a diagnosis of the oil pressure switch is required in order to detect a fallen off plug (open load).

Description:

Error symptoms are defined to this diagnosis function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─>                                  (= SYM_0)
 │ │ └───>                                  (= SYM_1)
 │ └─────> Open Load                        (= SYM_2)
 └───────> -                                (= SYM_3)
```

Signal flow diagram:



Application conditions:

Initialisation: according ERRM Type "**STD**" *(restored out from NVMY)*

at LV_IGK 0 → 1 **or** at reset: T_POIL_SWI_PWL = 0

Recurrence: 100 ms

Activation: LV_END_DIAG_POIL_SWI = 0 **and** ECU_STATE = PWL **and** LC_POIL_CTL_ENA = 0

Deactivation: LV_END_DIAG_POIL_SWI = 1 **or**
end of power-latch phase **or** LC_POIL_CTL_ENA = 1

Formula section:

The diagnosis shall be performed during the power-latch-phase until LV_END_DIAG_POIL_SWI = 1 or the end of power latch phase is reached.

```
IF (1)       ECU_STATE = PWL
THEN (1)     Timer T_POIL_SWI_PWL starts
             IF (2)       T_POIL_SWI_PWL > C_T_POIL_SWI_PWL
             THEN (2)     IF (3)       C_TOIL_POIL_SWI_MIN_DIAG <
                                              TOIL < C_TOIL_POIL_SWI_MAX_DIAG
                          THEN (3)     LV_CDN_DIAG_POIL_SWI = 1
                                       IF (4)       LV_POIL_SWI = 1
                                       THEN (4)     ERR_SYM_POIL_SWI = SYM_2
                                                    LV_ERR_POIL_SWI = 1 (after debounce)
                                       ELSE (4)     ERR_SYM_POIL_SWI = NO_SYM
                                                    LV_ERR_POIL_SWI = 0 (after rebounce)
                                       ENDIF (4)
                          ENDIF (3)
             ENDIF (2)
ENDIF (1)
```

End of diagnosis calculation LV_END_DIAG_POIL_SWI:

→ see chapter “Anti-bounce Algorithm: Calculation of the end of diagnosis”

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_POIL_SWI_PWL	1	0...3EH	0...6.2	0.1	[s]
Time for oil pressure inclination					
C_TOIL_POIL_SWI_MAX_DIAG	1	0...C8H	-40...160	1	[°C]
Maximum diagnosis threshold TOIL					
C_TOIL_POIL_SWI_MIN_DIAG	1	0...C8H	-40...160	1	[°C]
Minimum diagnosis threshold TOIL					
C_ABC_MAX_POIL_SWI	1	1...FFH	1...255	1	[-]
Anti-bounce max. value, POIL_SWI					
C_ABC_INC_POIL_SWI	1	0...FFH	0...255	1	[-]
Anti bounce increment value, POIL_SWI					
C_ABC_INC_POIL_DR_SCG	1	0...FFH	0...255	1	[-]
Antibounce counter increment engine-oil pressure valve powerstage error SCG					
C_ABC_MAX_POIL_DR_SCG	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter engine-oil pressure valve powerstage error SCG					
C_PWM_POIL_DR_MIN_DIAG_SCP	1	0..FFH	0..99.609375	0.391	%
Minimum threshold for SCP diagnosis window					
C_PWM_POIL_DR_MAX_DIAG_SCG	1	0..FFH	0..99.609375	0.391	%
Maximum threshold for SCG diagnosis window					
C_PWM_POIL_DR_MIN_DIAG_OC	1	0..FFH	0..99.609375	0.391	%
Minimum threshold for OC diagnosis window					
C_ABC_INC_POIL_DR	1	0..FFH	0..255	1	
Antibounce counter increment					
C_ABC_MAX_POIL_DR	1	1..FFH	1..255	1	
Maximum value for antibounce counter					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_POIL_DR	1	1 ... FH	1 ... 15	1	-
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (= 2)					

19.8 Cus adap module: ENLU

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Toel_getr	O/V	8000...7FFFH	-327.68...327.67	0.01	[°C]
gear box oil temperature					
Tget	O/V	8000...7FFFH	-327.68...327.67	0.01	[°C]
gear box oil temperature					
Toel_mdl	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
oil temperature model					
Oz_nivr	O/V	0...FFH	0...255	1	[-]
oil sensor raw value of oil level					
Oz_permr	O/V	0...FFFFH	0...65535	1	[-]
ÖZS: oil sensor raw value of oil permitivity					
Oz_tempr	O/V	0...FFH	0...255	1	[-]
ÖZS: oil sensor raw value of oil temperature					
Oz_status	O/V	0...FFH	0...255	1	[-]
ÖZS: status information of oil sensor					
B_td_ogok	O/V	0...1H	0...1	1	[-]
condition state machine once valid calculated					
Zrbosrt	O/V	0...FFH	0...255	1	[-]
Servicezählerwert vom Tester					
Id_bosrtt	O/V	0...FFH	0...255	1	[-]
ID f BOS_Reset vom Tester					
Bosbtvfbkt	O/V	0...FFH	0...255	1	[-]
Verfügbarkeit vom Tester					
Bosrlsmt	O/V	8000...7FFFH	-327680... 327670	10	[km]
Restlaufstrecke Motoröl vom Tester					
Bosrw2t	O/V	0...FFH	0...255000	1000	[km]
Prognose Intervall Weg vom Tester					
Bosmzielt	O/V	0...FFH	0...255	1	[-]
Zieltermin_Monat vom Tester					
Bosjzielt	O/V	0...FFH	0...255	1	[-]
Zieltermin Jahr vom Tester					
Bosprog2t	O/V	0...FFH	0...255	1	[-]
Prognose Intervall Zeit vom Tester					
Td_og_icnt	O/V	0...FFFFH	0...655.35	0.01	[s]
Time between flanks					
P_oel_ist	O/V	8000...7FFFH	-32768...32767	1	[hPa]
Ist Öldruck					
P_oelsol_tst	O/V	0...FFH	0...8160	32	[hPa]
Sollwert Öldruck über Tester					
B_poelsoltst	O/V	0...1H	0...1	1	[-]
Bedingung Sollwert Öldruck über Tester					
Oz_lf1c	O/V	0...FFH	0...2.55	0.01	[-]
Länderfaktor 1 codiert					
Oz_lf2c	O/V	0...FFH	0...2.55	0.01	[-]
Länderfaktor 2 codiert					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_poel	O/V	0...1H	0...1	1	[-]
Bedingung Öldrucksschalter					
Td_og_cnt	O/V	0...64H	0...100	1	[-]
Zähler für fallende Flanken im Töns Signal					
Td_og_tk	O/V	0...FFFFH	0...6553.5	0.1	[s]
Abkühlzeit vom TÖNS					
Td_og_tr	O/V	0...FFFFH	0...6553.5	0.1	[s]
Referntzeit					
Td_og_tv	O/V	0...FFFFH	0...6553.5	0.1	[s]
Abkühlzeit Vom TÖNS					
Td_og_uplcnt	O/V	0...64H	0...100	1	[-]
Zähler für unplausible Signale					
B_toel	O/V	0...1H	0...1	1	[-]
Bedingung tr und tv gültig					
B_td_og	O/V	0...1H	0...1	1	[-]
Bedingung Abkühlzeit gültig					
POIL_SP	O/V	0...4E20H	0...20	0.001	[bar]
setpoint of oil pressure					
POIL_PWM	O/V	0...FFFFH	0...99.99847	1.5259e-3	[%]
Tastverhältnis Öldruck-Regelventil					
FRQ_POIL_PWM	V	0A...FFH	10...255	1	[Hz]
Frequency of the oil pressure actuator PWM					
LV_POIL_SWI_CUS	O/V	0...1H	0...1	1	[-]
Condition oil pressure display (from customer)					
TOIL_MES	O/V	0...C8H	-40...160	1	[°C]
oil temperature					
TOIL	O/V	0...C8H	-40...160	1	[°C]
Oil temperature					
TOIL_STOP	O/V	0...C8H	-40...160	1	[°C]
Oil temperature at transition to ES					
DIST_RESI_OIL	O/V	8000...7FFFH	-327680... 327670	10	[km]
Ausgabewert Restlaufstrecke Motoröl					
DIST_RESI_OIL_KM	O/V	0...FFH	0...255	1	[-]
Einheit der BOS-Info (km)					
STATE_OIL_AVL	O/V	0...FFH	0...255	1	[-]
Rest-Verfügbarkeit Motoröl in Prozent					
zrbosmld	O/V/S	0...FFH	0...255	1	[-]
anzuzeigender Zähler für durchgeführten Ölservice					
idbosmsg	O	0...FFH	0...255	1	[-]
Kennung für BOS-Meldung vom Kombi					
STATE_OIL_REQ	O/V	0...FFH	0...255	1	[-]
Recommended refill volume engine-oil					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_LOIL	O/V	0...FFH	0...255	1	[-]
fluid-level engine-oil					
QOIL_DS_CAN_2_1	O/V	0...FFH	0...255	1	[-]
ID2 BOS message 2					
QOIL_DS_CAN_2_2	O/V	0...FFFFH	0...65535	1	[-]
ID function BOS message 2					
QOIL_DS_CAN_2_3	O/V	0...FFH	0...255000	1000	[km]
Forecast intermittent way BOS message 2					
QOIL_DS_CAN_2_4	O/V	0...FFH	0...255	1	[-]
target-date year BOS message 2					
QOIL_DS_CAN_2_5	O/V	0...FFH	0...255	1	[-]
target-date month BOS message 2					
QOIL_DS_CAN_2_6	O/V	0...FFH	0...255	1	[-]
Forecast intermittent time BOS message 2					
LV_SOIL_VLD_LOIL_CUS	O/V	0...1H	0...1	1	[-]
Oil temperature valid from customer					
LV_T_COOL_VLD_LOIL_CUS	O/V	0...1H	0...1	1	[-]
Cool off time valid from customer					
LV_LOIL_VLD_WR	O/V	0...1H	0...1	1	[-]
Flag to set LV_SOIL_VLD_LOIL and LV_T_COOL_VLD_LOIL					

Input data:

ozstatus	oznivr	oztempr	ozpermrw
QOIL_DS_RST_KWP_1	QOIL_DS_RST_KWP_2	QOIL_DS_RST_KWP_3	
QOIL_DS_RST_KWP_5	QOIL_DS_RST_KWP_6	QOIL_DS_RST_KWP_7	QOIL_DS_RST_KWP_8
TEMP_GB	FAC_OIL_EXT_REQ_1	FAC_OIL_EXT_REQ_2	LV_ST_END
POIL	POIL_SP_EXT_ADJ	LV_POIL_SP_EXT_ADJ	QOIL_DS_RST_KWP_9
LV_BIOS_POIL_SWI	T_EDGE_FALL_LOIL	CTR_EDGE_FALL_LOIL	T_COOL_LOIL
T_REF_SIG_LEN_LOIL	T_TEMP_SIG_LEN_LOIL	CTR_SIG_NOT_VLD_LOIL	LV_SOIL_VLD_LOIL
LV_T_COOL_VLD_LOIL	LV_T_ES_NOT_PLAUS	LV_STATE_RUN_LOIL	NC_CYL_NR
Id_bosrtak			
Toel	Bosrlsm	Bosstate	Oz_kvbsm_ul
Bosbtvfbk	B_poel_rot	Oz_lp	Oz_lv
Bosjziel	Id_bosmsg	Bosmziel	Bosprog2
Bosid2	Bosfxid2	Bosrw2	P_oel_soll
	LV_POIL_PWM_EXT_ADJ	POIL_PWM_EXT_ADJ	N
LC_POIL_CTL_ENA			LV_ERR_POIL_DR
Poel_fpwm	Tv_poel		

Import actions:

ACTION_INFR_SetFrqPwmPoilDr(IN <frq_poil_pwm**>)**
This action sets the frequency of the oil pressure actuator PWM
ACTION_INFR_SetPwmPoilDr(IN <poil_pwm**>)**
This action sets the duty cycle of the oil pressure actuator PWM

19.8.1 Outputs for BMW functions which are defined as ENLU exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: 0 at reset, *Td_og_icnt = 0s*
first valid value: Tget
Oz_lf1c = FAC_OIL_EXT_REQ_1
Oz_lf2c = FAC_OIL_EXT_REQ_2

Recurrence : 1000ms: Toel_mdl, Oz_permr, Oz_status, Oz_tempr, Oz_nivr,
Zrbosrt, Id_bosrtt, Bosbtvfbkt, Bosrlsmt,
Bosrw2t, Bosmzielt, Bosjzielt, Bosprog2t
100 ms: Toel_getr, B_td_ogok, B_poel, Td_og_icnt, Td_og_cnt,
Td_og_tk, Td_og_tr, Td_og_tv, Td_og_uplcnt, B_toel,
B_td_og
10 ms: Tget, P_oelsol_tst, P_oel_ist, B_poelsoltst
Activation: every engine state

During power latch:
Tget, P_oel_ist: last valid value
P_oelsol_tst, B_poelsoltst, Bosrlsmt = 0

Formula section:

Remark: all formulas are valid in a **physical** meaning

Toel_getr = Tget = TEMP_GB
Toel_mdl = C_TOIL_SUB_DIAG
Oz_nivr = oznivr
Oz_tempr = oztempr
Oz_permr = ozpermr_w
Oz_status = ozstatus

Id_bosrtt = QOIL_DS_RST_KWP_1
Bosbtvfbkt = QOIL_DS_RST_KWP_2
Zrbosrt = QOIL_DS_RST_KWP_3

```
IF      QOIL_DS_RST_KWP_9 = 1
THEN    Bosrlsmt    =    QOIL_DS_RST_KWP_4
ELSE    Bosrlsmt    =    8000h  (unplausibel)
```

Bosrw2t = QOIL_DS_RST_KWP_5
Bosmzielt = QOIL_DS_RST_KWP_6
Bosjzielt = QOIL_DS_RST_KWP_7
Bosprog2t = QOIL_DS_RST_KWP_8
Oz_lf1c = FAC_OIL_EXT_REQ_1
Oz_lf2c = FAC_OIL_EXT_REQ_2

```
P_oel_ist     = POIL
B_poelsoltst = LV_POIL_SP_EXT_ADJ
B_poel =  LV_BIOS_POIL_SWI


if LV_ST_END = 1
then P_oelsol_tst = POIL_SP_EXT_ADJ
else  P_oelsol_tst = 0
endif
if    NC_CYL_NR = 6
then
      Td_og_cnt = 0
      Td_og_icnt = 0
      Td_og_tk = 0
      Td_og_tr = 0
      Td_0g_tv = 0
      Td_og_uplcnt = 0
      B_toel = 0
      B_td_og = 0
      B_td_ogok = !LV_T_ES_NOT_PLAUS
else Td_og_icnt = T_EDGE_FALL_LOIL
      Td_og_cnt = CTR_EDGE_FALL_LOIL
      Td_og_tk = T_COOL_LOIL
      Td_og_tr =  T_REF_SIG_LEN_LOIL
      Td_og_tv =   T_TEMP_SIG_LEN_LOIL
      Td_og_uplcnt = CTR_SIG_NOT_VLD_LOIL
      B_toel = LV_SOIL_VLD_LOIL                    // before ozgg_100ms()
      B_td_og = LV_T_COOL_VLD_LOIL                 // before ozgg_100ms()
      B_td_ogok = LV_STATE_RUN_LOIL
endif
```

19.8.2 Outputs for SV aggregates

FUNCTION DESCRIPTION:

Adaptation to BMW environment

Application conditions:

Initialisation: at reset: DIST_RESI_OIL = Bosrlsm
STATE_OIL_AVL = Bosbtvfbk
FRQ_POIL_PWM = 244 Hz
ACTION_INFR_SetFrqPwmPoilDr(IN <FRQ_POIL_PWM>).

after BMW-ini2 (calcualtion like in formula section):
idbosmsg, STATE_OIL_AVL, OIL_DS_CAN_2_1,
QOIL_DS_CAN_2_2, QOIL_DS_CAN_2_3,
OIL_DS_CAN_2_4, QOIL_DS_CAN_2_5, QOIL_DS_CAN_2_6,
DIST_RESI_OIL, zrbosmld

at LV_ES 0->1 : TOIL_STOP

Recurrence: 10ms: POIL_SP, POIL_PWM, FRQ_POIL_PWM, LV_POIL_SWI_CUS
100 ms: LV_SOIL_VLD_LOIL_CUS, LV_T_COOL_VLD_LOIL_CUS,
LV_LOIL_VLD_WR

1000 ms: TOIL_MES, TOIL, DIST_RESI_OIL,
DIST_RESI_OIL_KM, STATE_OIL_AVL,
idbosmsg,QOIL_DS_CAN_2_1,QOIL_DS_CAN_2_2,
QOIL_DS_CAN_2_3, QOIL_DS_CAN_2_4, QOIL_DS_CAN_2_5,
QOIL_DS_CAN_2_6, zrbosmld

Activation: every engine state

Formula section:

TOIL_MES = Toel
TOIL = Toel
DIST_RESI_OIL = Bosrlsm
DIST_RESI_OIL_KM = Bosstate
STATE_OIL_AVL = Bosbtvfbk
zrbosmld = Zrbosmld
idbosmsg = Id_bosmsg
STATE_OIL_REQ = Oz_lv
STATE_LOIL = Oz_lp
QOIL_DS_CAN_2_1 = Bosid2
QOIL_DS_CAN_2_2 = Bosfxid2
QOIL_DS_CAN_2_3 = Bosrw2
QOIL_DS_CAN_2_4 = Bosjziel
QOIL_DS_CAN_2_5 = Bosmziel
QOIL_DS_CAN_2_6 = Bosprog2
LV_POIL_SWI_CUS = B_poel_rot
POIL_SP = P_oel_soll

```
FRQ_POIL_PWM          = Poel_fpwm

If        LC_POIL_CTL_ENA = 1
then      ACTION_INFR_SetFrqPwmPoilDr(IN <FRQ_POIL_PWM>)
          if      N = 0 and LV_POIL_PWM_EXT_ADJ = 1
          then    POIL_PWM = POIL_PWM_EXT_ADJ
          else    if      LV_ERR_POIL_DR = 0
                  then    POIL_PWM = Tv_poel
                  else    POIL_PWM = C_POIL_PWM_ERR_SUB
                  endif
          endif
          ACTION_INFR_SetPwmPoilDr(IN <POIL_PWM>)
          // Due to the used HW module CLK6 the specified resolution of POIL_PWM
          // (1.5259e-3) can not be realised. The CLK6 is limited by following technical
          // data: 4.47 Hz (0.2237ms) ... 292941.45 Hz (3.413µs)

else
          "no connection to I/O Software"
endif
LV_SOIL_VLD_LOIL_CUS = B_toel            // after ozgg_100ms()
LV_T_COOL_VLD_LOIL_CUS = B_td_og         // after ozgg_100ms()
If LV_LOIL_VLD_WR = 0
     then    LV_LOIL_VLD_WR = 1          // after ozgg_100ms()
     else    LV_LOIL_VLD_WR = 0          // after ozgg_100ms()
endif
```

19.8.2.1 Oil Temperature at Engine Stopping TOIL_STOP

TOIL_STOP is determined at each transition to ES. TOIL_STOP will be stored in the non-volatile memory during the ECU's self holding phase.
By occuring of EEPROM error TOIL_STOP should be initialized with –40 °C (0x0h).

Initialisation: with stored value

Formula section:

TOIL_STOP = TOIL

Calibration data:

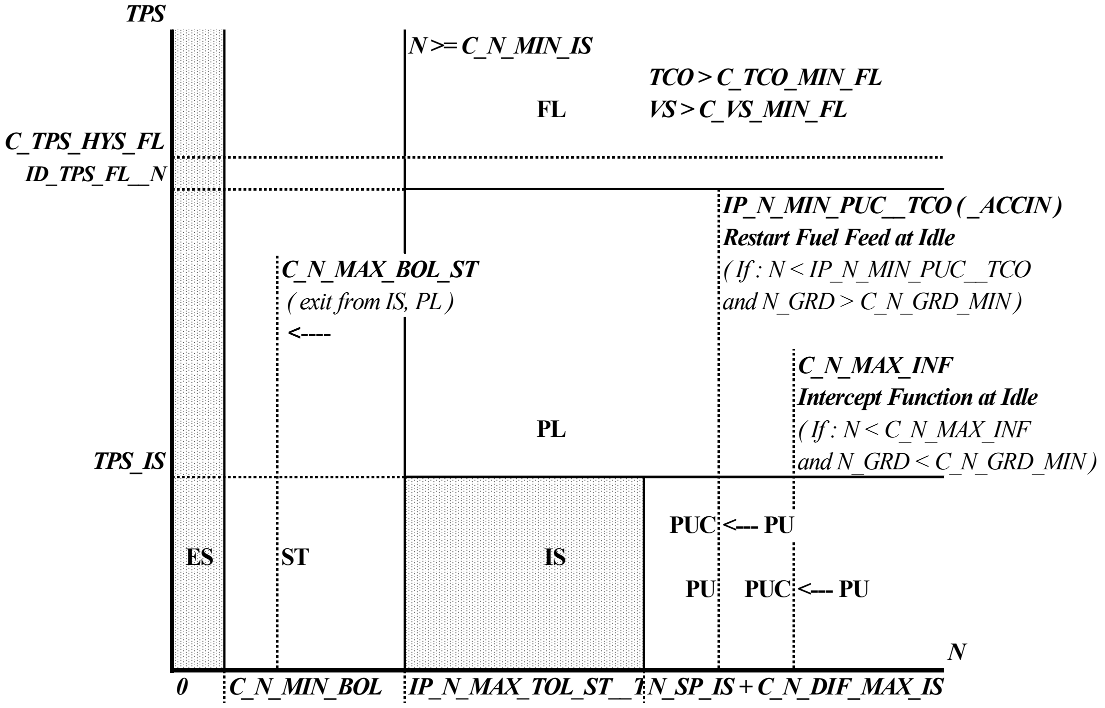
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TOIL_SUB_DIAG	1	0...C8H	-40...160	1	[°C]
substitute oil temperature					
C_T_TIA_TCO_DIAG	1	1...FFFFH	1...65535	1	[s]
time condiotion for toil-, tco-, tecu- and tia diagnosis					
C_POIL_PWM_ERR_SUB	1	0...FFH	0...99.60937	0.390625	[%]
Substitute value of POIL_PWM in case of detected failure on oil pressure actuator					

20 Engine operating states

Table of Contents

Index

20.1 General Drawing



LV_ES : Engine Stopped

LV_ST : Start

LV_IS : Idle Speed

LV_PL : Part Load

LV_PU : Trailing Throttle

LV_PUC : Trailing Throttle Fuel Cut Off

20.2 Basic Operating States : LV_ES, LV_ST, LV_IS, LV_PL, LV_PU & LV_PUC

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CYCNR_HYS_PUC_ST	-	0... FFFFH	0... 65535	1	[-]
Cycle counter for increased engine speed hysteresis for detection of LV_PUC after engine start					
LV_CS_PUC	V	0... 1H	0... 1	1	[-]
Clutch switch information including AMT					
LV_ES	O/V	0... 1H	0... 1	1	[-]
Engine operating state "Engine stopped"					
LV_IGA_MIN_PUC	O/V	0... 1H	0... 1	1	[-]
conditions for IGA_MIN_PUC					
LV_IS	O/V	0... 1H	0... 1	1	[-]
Engine operating state "Idle Speed"					
LV_PL	O/V	0... 1H	0... 1	1	[-]
Engine operating state "Part Load"					
LV_PU	O/V	0... 1H	0... 1	1	[-]
Engine operating state "Trailing throttle"					
LV_PUC	O/V	0... 1H	0... 1	1	[-]
Engine operating state "trailing throttle fuel cut off "					
LV_PUC_REQ	O/V	0... 1H	0... 1	1	[-]
Logical variable trailing throttle fuel cut off request					
LV_ST	O/V	0... 1H	0... 1	1	[-]
Engine operating state "Start"					
LV_ST_END	O/V	0... 1H	0... 1	1	[-]
Engine operating state "Start" not active					
N_DIF_MAX_HYS_1	-	0... 1FE0H	0... 8160	1	[rpm]
Visible value for testing hysteresis in transition IS to PU					
N_DIF_MAX_HYS_2	-	0... 1FE0H	0... 8160	1	[rpm]
Visible value for testing hysteresis in transition PU to PUC					
N_HYS_ACT_PUC	V	0... FFH	0... 8160	32	[rpm]
Actual effective engine speed hysteresis for LV_PUC					
N_HYS_MIN_PUC	V	0... FFH	0... 8160	32	[rpm]
Minimum engine speed hysteresis for LV_PUC					
N_HYS_PUC	-	0... FFH	0... 8160	32	[rpm]
Engine speed hysteresis for the entry into PUC					
N_MIN_PUC	V	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for the activation of LV_PUC.					
STATE_ENG	V	0 1 2 3 4 5	ES ST IS PL PU PUC	-	[-]
engine operating state					
T_IS	O/V	0... FFFFH	0... 655.35	0.01	[s]
Time since the activation of the state idle speed.					
T_PU	O/V	0... FFFFH	0... 655.35	0.01	[s]
Time elapsed into LV_PU					
T_PU_MSR	-	0... FFFFH	0... 655.35	0.01	[-]
Time after active MSR function					

Input Data:

CYC_CAST	GR_AT	GR_MT	LV_ACCOUT_RLY
LV_AT	LV_CT	LV_DCC_INC_ACT	LV_DCC_PUC_INH
LV_DT	LV_GS_INC_ACT	LV_INF	LV_LDM_PUC_INH
LV_MSR_ACT	LV_PUC_LOCK_TNT	LV_PUC_SA_INH	LV_RGN_NT_REQ
LV_RNG_L_REQ	LV_S_ACT	LV_TQI_BOL_SET	LV_TQI_BOL_SET_S
N	NC_N_MIN	N_32	N_DIF_MMV
N_GRD	N_MAX_TOL_ST	N_SP_IS	TCO
TQ_LOSS_ARS_SP_CAN	TQ_LOSS_PSTE_2_SP_CAN	T_AST	VS
LV_CS	OPM_AV	C_N_DIF_FAC	C_N_GRD_MIN
IP_N_MAX_INF	ID_IDX_IGA_MIN	LV_PUC_INH_TEMP_CAT	LV_INH_PUC_CUS
TQ_LOSS_ADD	LV_VAR_TCT	STATE_ETCU_CLU	STATE_TCT_INTV

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CLC_STATE_ETCU_CLU	1	0... FFH	0... 255	1	[-]
Constant to define closed clutch depending on shift-process TCT via CAN					
C_CYCNR_HYS_PUC	1	0... FFFFH	0... 65535	1	[°CRK]
Cycle counter to reset the LV_PUC engine speed hysteresis					
C_CYCNR_HYS_PUC_ST	1	0... FFFFH	0... 65535	1	[°CRK]
Cycle counter to reset the LV_PUC engine speed hysteresis in cold condition					
C_N_DIF_AST	1	0... 1FE0H	0... 8160	1	[rpm]
Additional engine speed hysteresis to avoid "LV_PU" direct after start					
C_N_DIF_MAX_HYS_IS	1	0... 1FE0H	0... 8160	1	[rpm]
Engine speed hys. for transition to IS with stopped car					
C_N_DIF_MAX_HYS_PUC	1	0... 1FE0H	0... 8160	1	[rpm]
Engine speed hys. for transition PU to PUC					
C_N_MAX_BOL_ST	1	0... FFH	0... 8160	32	[rpm]
Engine speed threshold to detect LV_ST or LV_PL from LV_IS					
C_T_MAX_PU	1	0... FFFFH	0... 655.35	0.01	[s]
Maximum delay time of PUC activation after entry of PU					
C_T_MIN_PU_MSR	1	0... FFFFH	0... 655.35	0.01	[s]
Minimum delay time of PUC activation after MSR					
C_T_MIN_PU_RNG_L	1	0... FFFFH	0... 655.35	0.01	[s]
Minimum delay time of PUC activation after entry of PU for LV_RNG_L_REQ = 1					
C_T_MIN_PU_VS	1	0... FFFFH	0... 655.35	0.01	[s]
Minimum delay time of PUC activation after entry of PU					
C_T_MIN_PU_VS_CS	1	0... FFFFH	0... 655.35	0.01	[s]
Minimum delay time for enry PUC due to clutch switch for MT					
C_T_N_DIF_AST	1	0... FFFFH	0... 6553.5	0.1	[s]
Time after start where C_N_DIF_AST is additional calculated to engine speed hysteresis					
C_TCO_MIN_PUC_ST	1	0... FEH	-48... 142.5	0.75	[°C]
Coolant temperature to select the LV_PUC engine speed hysteresis					
C_VS_MIN_PU	1	0... FFH	0... 255	1	[km/h]
minimum vehicle speed of IGA_Min of PU					
C_VS_PU_CS_MAX	1	0... FFH	0... 255	1	[km/h]
Maximum vehicle speed for delayed PUC due to clutch switch for MT					
C_VS_PU_CS_MIN	1	0... FFH	0... 255	1	[km/h]
Minimum vehicle speed for delayed PUC due to clutch switch for MT					
ID_N_HYS_MAX_PUC__GR_AT	8	0... FFH	0... 8160	32	[rpm]
LDPM_GR_AT_3	8	0... FFH	0... 255	1	[-]
Engine speed hysteresis set entering LV_PUC					
ID_N_HYS_MAX_PUC__GR_MT	8	0... FFH	0... 8160	32	[rpm]
LDPM_GR_MT_3	8	0... FFH	0... 255	1	[-]
Engine speed hysteresis set entering LV_PUC for MT transmissions					
ID_N_HYS_MIN_PUC__GR_AT	8*4	0... FFH	0... 8160	32	[rpm]

LDPM_GR_AT_3	8	0... FFH	0... 255	1	[-]
LDPM_VS	4	0... FFH	0... 255	1	[km/h]
Engine speed hysteresis reset when C_CYCNR_HYS_PUC_xx achieved					
ID_N_HYS_MIN_PUC__GR_MT	8*4	0... FFH	0... 8160	32	[rpm]
LDPM_GR_MT_3	8	0... FFH	0... 255	1	[-]
LDPM_VS	4	0... FFH	0... 255	1	[km/h]
Engine speed hysteresis reset when C_CYCNR_HYS_PUC_xx achieved					
ID_N_HYS_PUC_ST__GR_AT	6	0... FFH	0... 8160	32	[rpm]
LDPM_GR_AT_1	6	0... FFH	0... 255	1	[-]
Engine speed hysteresis set entering LV_PUC in cold condition for AT transmission					
ID_N_HYS_PUC_ST__GR_MT	8	0... FFH	0... 8160	32	[rpm]
LDPM_GR_MT_3	8	0... FFH	0... 255	1	[-]
Engine speed hysteresis set entering LV_PUC in cold condition for MT transmission.					
IP_CYC_CAST__TCO	9	0... FFFFH	0... 65535	1	[-]
LDPM_TCO_9	9	0... FEH	-48... 142.5	0.75	[°C]
Cycle counter to define the duration of the start phase charcterised by LV_ST					
IP_N_ACCOUT_MIN_PUC__TCO__GR_AT	8*6	0... FFH	0... 8160	32	[rpm]
LDPM_TCO_1_6	6	0... FEH	-48... 142.5	0.75	[°C]
LDPM_GR_AT_3	8	0... FFH	0... 255	1	[-]
LV_PUC engine speed threshold with air conditioned compressor active					
IP_N_ACCOUT_MIN_PUC__TCO__GR_MT	8*6	0... FFH	0... 8160	32	[rpm]
LDPM_TCO_1_6	6	0... FEH	-48... 142.5	0.75	[°C]
LDPM_GR_MT_3	8	0... FFH	0... 255	1	[-]
LV_PUC engine speed threshold with air conditioning compressor active					
IP_N_DIF_MAX_IS	6	0... 1FE0H	0... 8160	1	[rpm]
LDP_N_GRD_N_DIF_MAX_IS	6	0... FFH	-4096 ...4064	32	[rpm/s]
Engine speed hysteresis before "Idle Speed" (LV_IS)					
IP_N_MIN_PUC__TCO__GR_AT	8*6	0... FFH	0... 8160	32	[rpm]
LDPM_TCO_1_6	6	0... FEH	-48... 142.5	0.75	[°C]
LDPM_GR_AT_3	8	0... FFH	0... 255	1	[-]
LV_PUC engine speed threshold					
IP_N_MIN_PUC__TCO__GR_MT	8*6	0... FFH	0... 8160	32	[rpm]
LDPM_TCO_1_6	6	0... FEH	-48... 142.5	0.75	[°C]
LDPM_GR_MT_3	8	0... FFH	0... 255	1	[-]
LV_PUC engine speed threshold					
IP_N_MIN_PUC_ADD	6	0... FFH	0... 8160	32	[rpm]
LDP_TQ_LOSS_ADD__PUC	6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Additive term for reactivation engine speed					
IP_N_MIN_PUC_ADD_ARS	6	0... FFH	0... 8160	32	[rpm]
LDP_TQ_LOSS_ARS_SP_CAN__PUC	6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Additive term for reactivation engine speed					
IP_N_MIN_PUC_ADD_PSTE_2	6	0... FFH	0... 8160	32	[rpm]
LDP_TQ_LOSS_PSTE_2_SP_CAN__PUC	6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Additive term for reactivation engine speed					
IP_T_MIN_PU__N__TCO	8*6	0... FFFFH	0... 655.35	0.01	[s]
LDP_N__T_MIN_PU	8	0... 1FE0H	0... 8160	1	[rpm]
LDP_TCO__T_MIN_PU	6	0... FEH	-48... 142.5	0.75	[°C]
Minimum delay time of PUC activation					
IP_T_MIN_PU_HOM	3	0... FFFFH	0... 655.35	0.01	[s]
LDPM_N_T_MIN_PU_OPM	3	0... 1FE0H	0... 8160	1	[rpm]
Minimum delay time of PUC activation after entry of PU in homogenius mode					
IP_T_MIN_PU_RNG_L__N__TCO	8*6	0... FFFFH	0... 655.35	0.01	[s]
LDP_N__T_MIN_PU	8	0... 1FE0H	0... 8160	1	[rpm]
LDP_TCO__T_MIN_PU	6	0... FEH	-48... 142.5	0.75	[°C]
Minimum delay time of PUC activation					

IP_T_MIN_PU_S	3	0... FFFFH	0... 655.35	0.01	[s]
LDPM_N_T_MIN_PU_OPM	3	0... 1FE0H	0... 8160	1	[rpm]
Minimum delay time of PUC activation after entry of PU in s mode					
LC_INH_DT_PU	1	0... 1H	0... 1	1	[-]
Manual inhibition bit for deactivation drivetrain impact for controlling transition to PUC					
LC_INH_PUC_NT_REQ	1	0... 1H	0... 1	1	[-]
Manual switch to inhibit PUC due to regeneration request of NOx management					
LC_N_HYS_CYL_CUT_OFF_AT_CONF	1	0... 1H	0... 1	1	[-]
Logical constant for premature reset to minimum engine speed hysteresis, vehicle with automatic transmission					
LC_N_HYS_CYL_CUT_OFF_MT_CONF	1	0... 1H	0... 1	1	[-]
Logical constant for premature reset to minimum engine speed hysteresis, vehicle with manual transmission					

General Information

From the 6 different basic operating states only 1 can be active at the same time. So this function can be realised as a state machine. This specification describes the transition conditions.

Application Conditions

Initialization: RST, ERU2ES, ES2ERU
Activation: 10MS: always
SEG: always
Deactivation: never

Function description

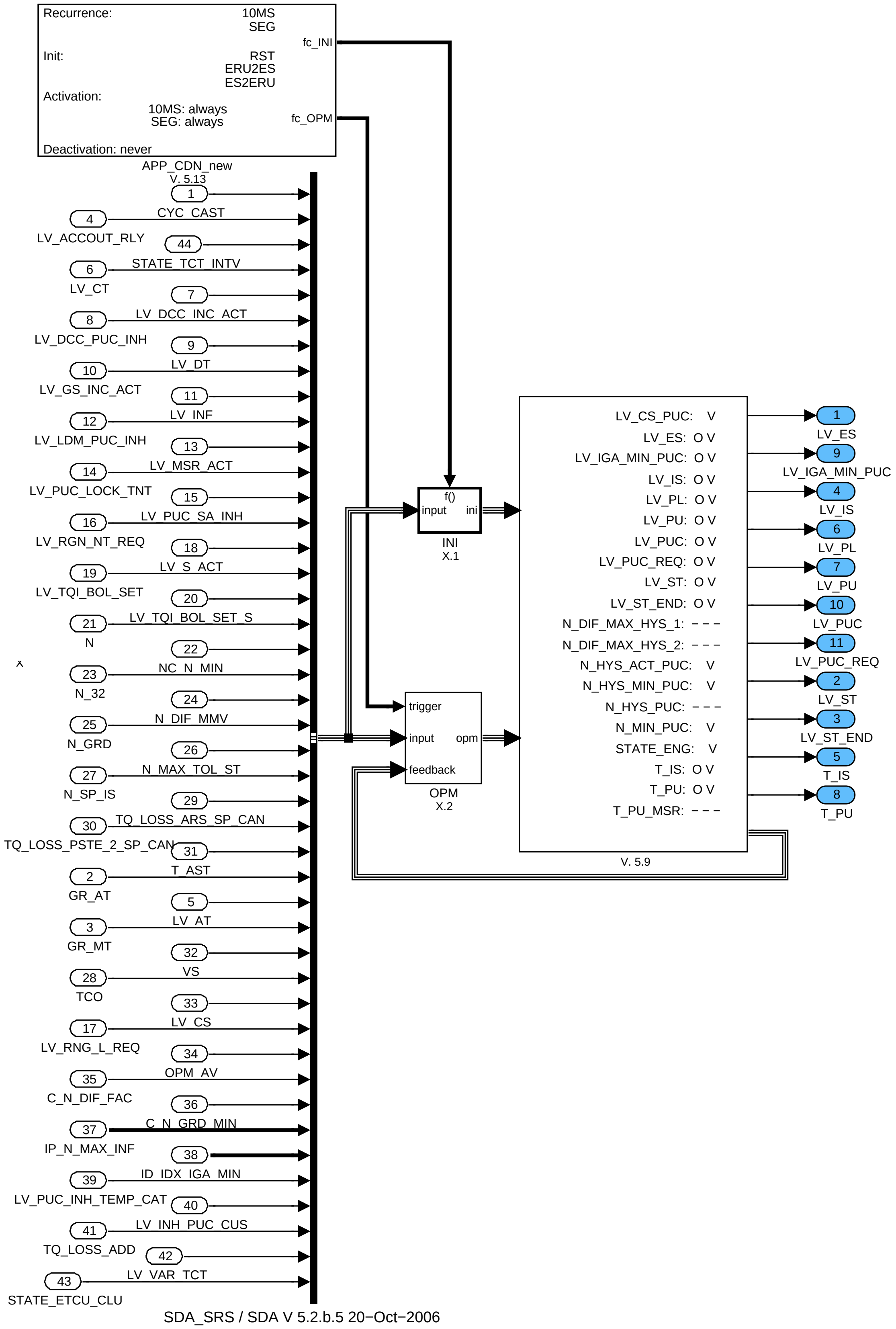


Figure 1:

Chapter	Baseline	Include File
Engine operating states	4DC3940S	43500202.00C

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

20.2.1 Engine operating state "engine stopped"

Initialisation of Variables at Reset,ES2ERU and ERU2ES
LV_ES is the operating state which is set during ECU initialisation.
The engine operating state Engine Stopped (LV_ES) is characterized by N_32 < NC_N_MIN. In this engine operating state, the functions are assigned to initialisation values.
The entire scope of diagnosis and actuator control is accessible.

20.2.1.1 Initilisation of Variables at Reset

LV_ES is the operating state which is set during ECU initialisation

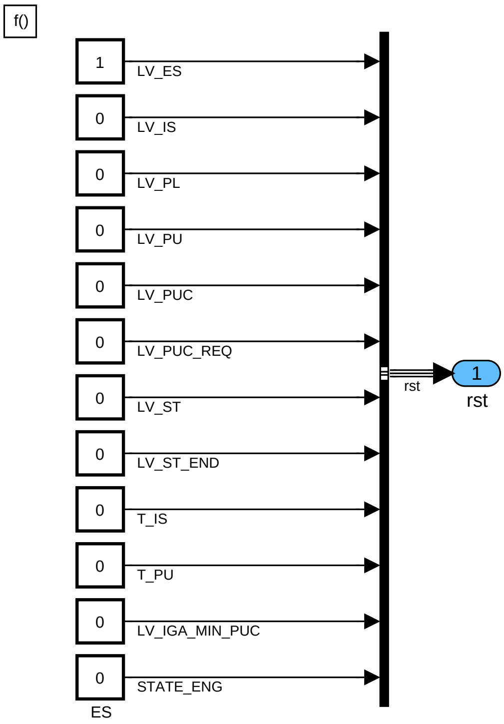


Figure 2:

20.2.1.2 Initilisation of Variables at ERU2ES Event

The initialization of variables are done at ERU2ES event

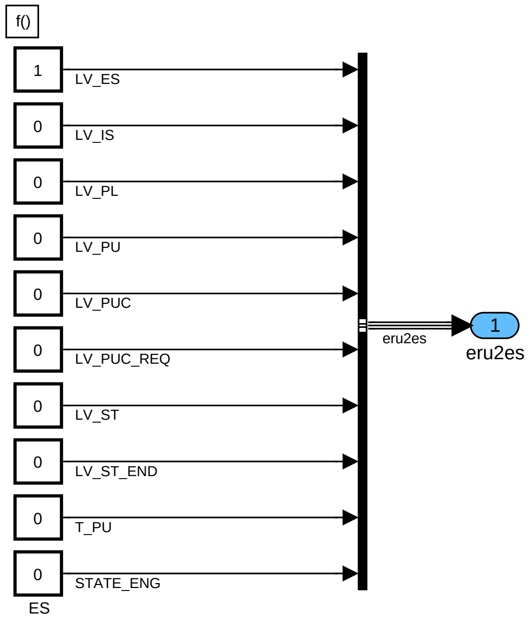


Figure 3:

20.2.1.3 Initialization of Variables at ES2ERU

The initialization of variables are done at ES2ERU event

20.2.1.3.1 Initialization of N_HYS_PUC

The trigger for the calculation of N_HYS_PUC

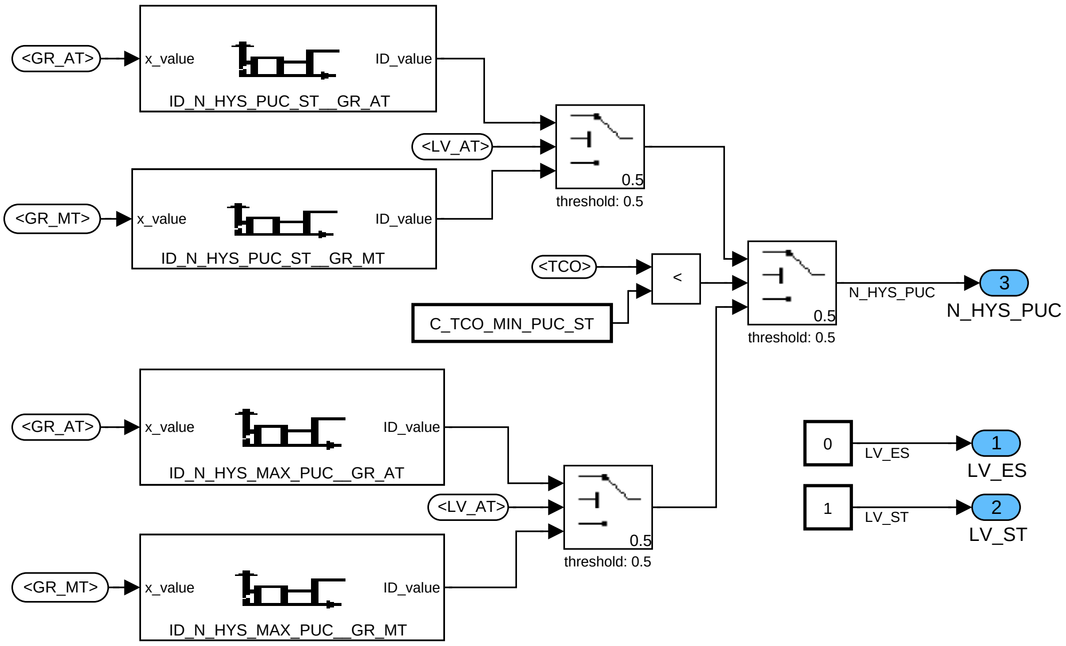


Figure 4:

20.2.2 Formula Section

All the variables are calculated under 10 MS and SEG recurrence

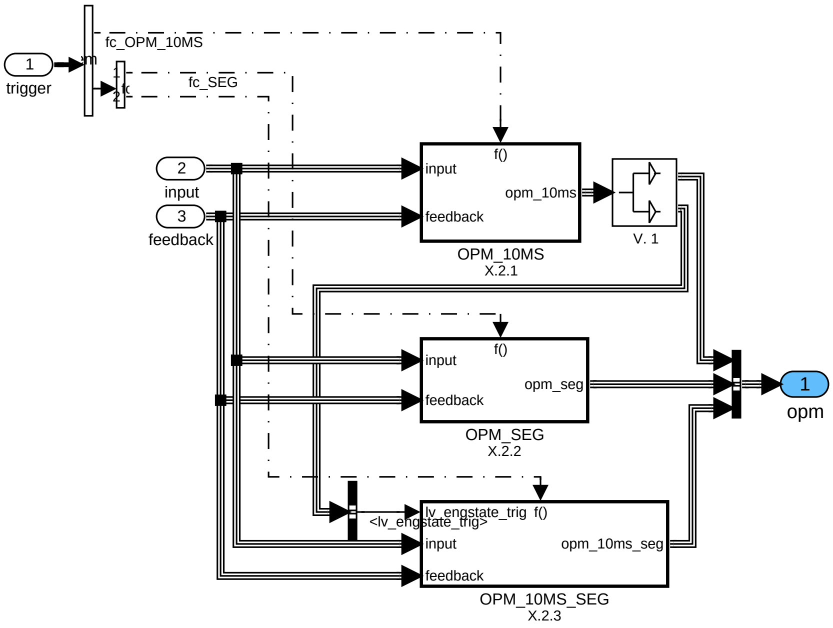


Figure 5:

20.2.2.1 Formula Section for 10 ms Recurrence

The timer values are calculated in this block at 10 ms recurrence.

20.2.2.1.1 Calculation of Timer T_IS

The timer is calculated in this subchapter

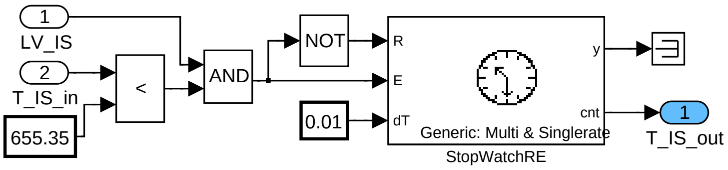


Figure 6:

20.2.2.1.2 Activation of the timer T_PU:

Timer T_PU is calculated in this block

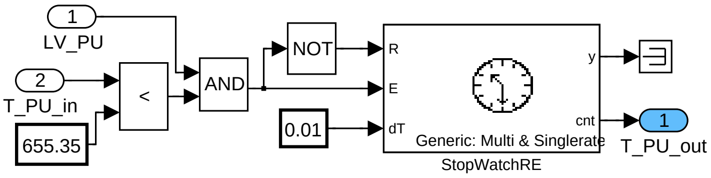


Figure 7:

20.2.2.1.3 Calculation of T_PU_MSR

This block computes T_PU_MSR

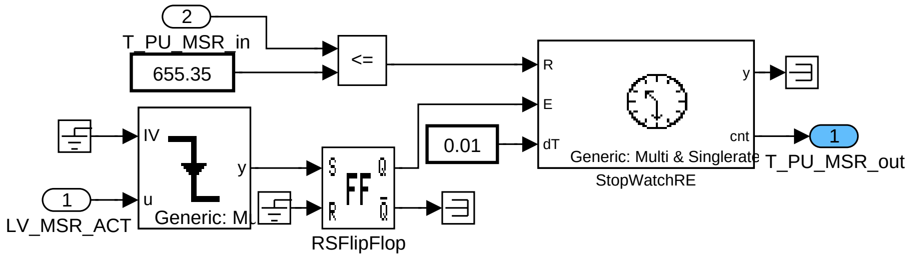


Figure 8:

20.2.2.1.4 Set engine state trigger flag

lv_engstate_trig is set to 1

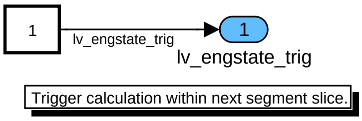


Figure 9:

20.2.2.2 Calculation of speed for segment recurrence

N_HYS_PUC is calculated.

20.2.2.2.1 Calculation of Flag

N_HYS_PUC is calculated by checking the values of LV_PUC flag.

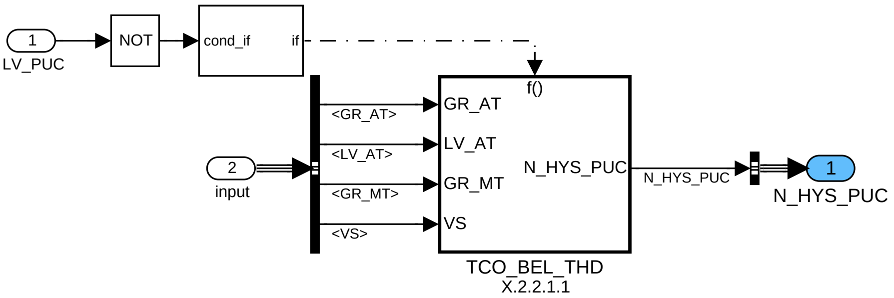


Figure 10:

20.2.2.2.1.1 Calculation of N_HYS_PUC

N_HYS_PUC is calculated in this block.

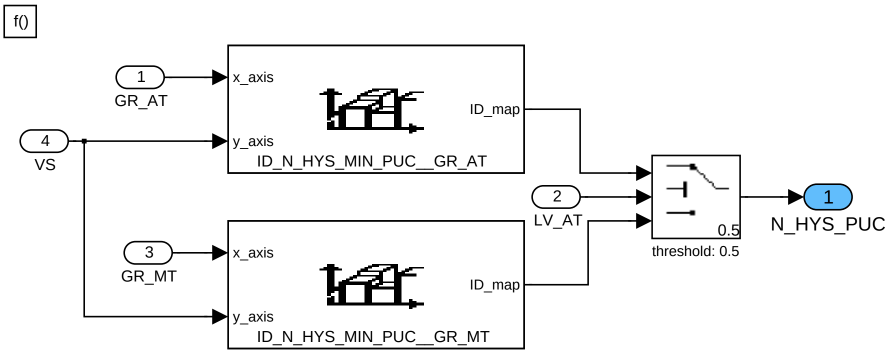


Figure 11:

20.2.2.3 Calculation of engine state every segment synchronized with 10ms slice

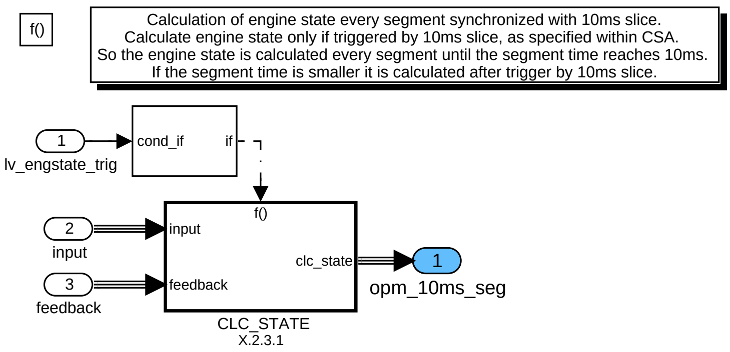


Figure 12:

20.2.2.3.1 Calculation at 10 MS and SEGMENT recurrence

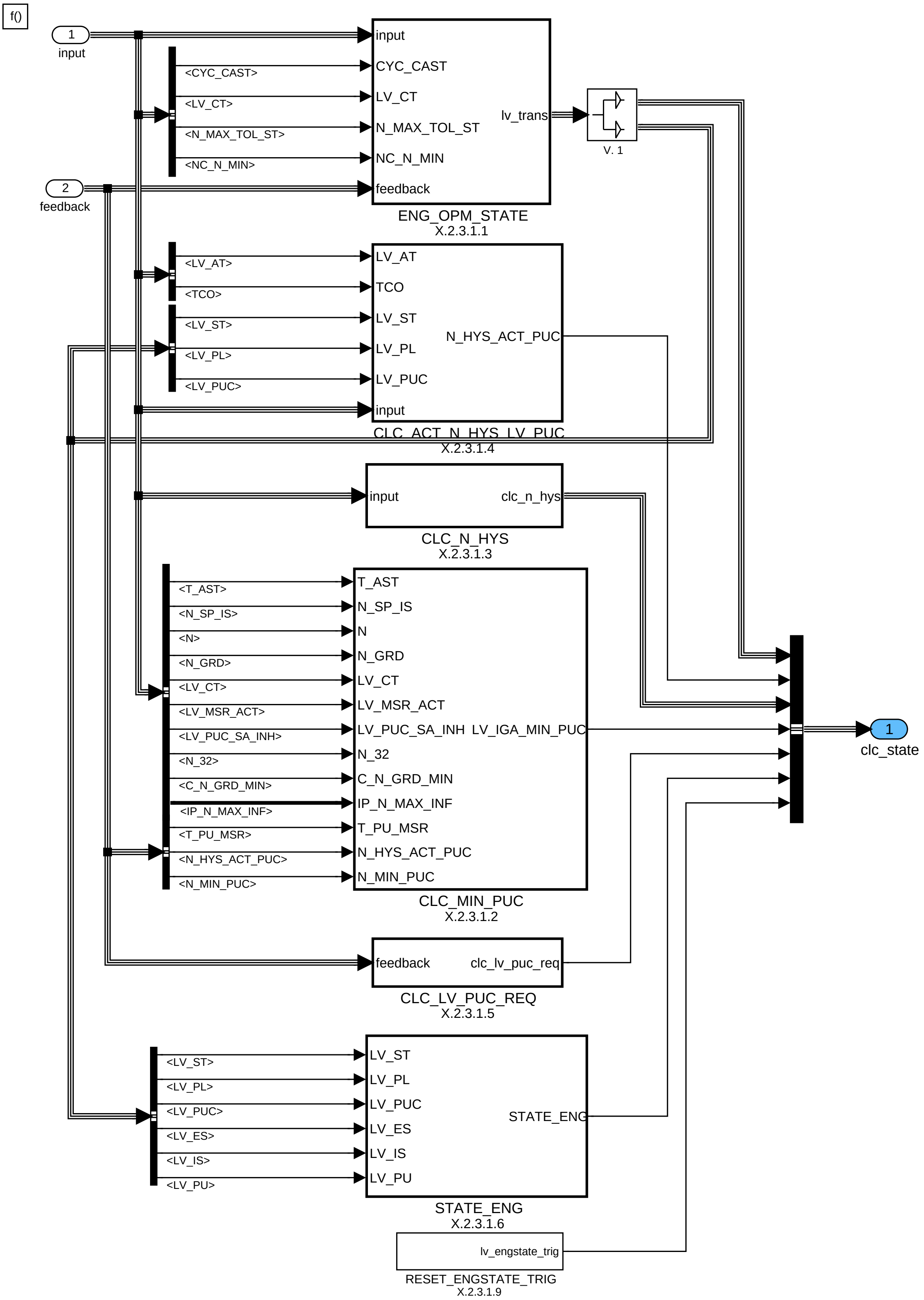


Figure 13:

20.2.2.3.1.1 Calculation of Engine States

The State variables are calculated

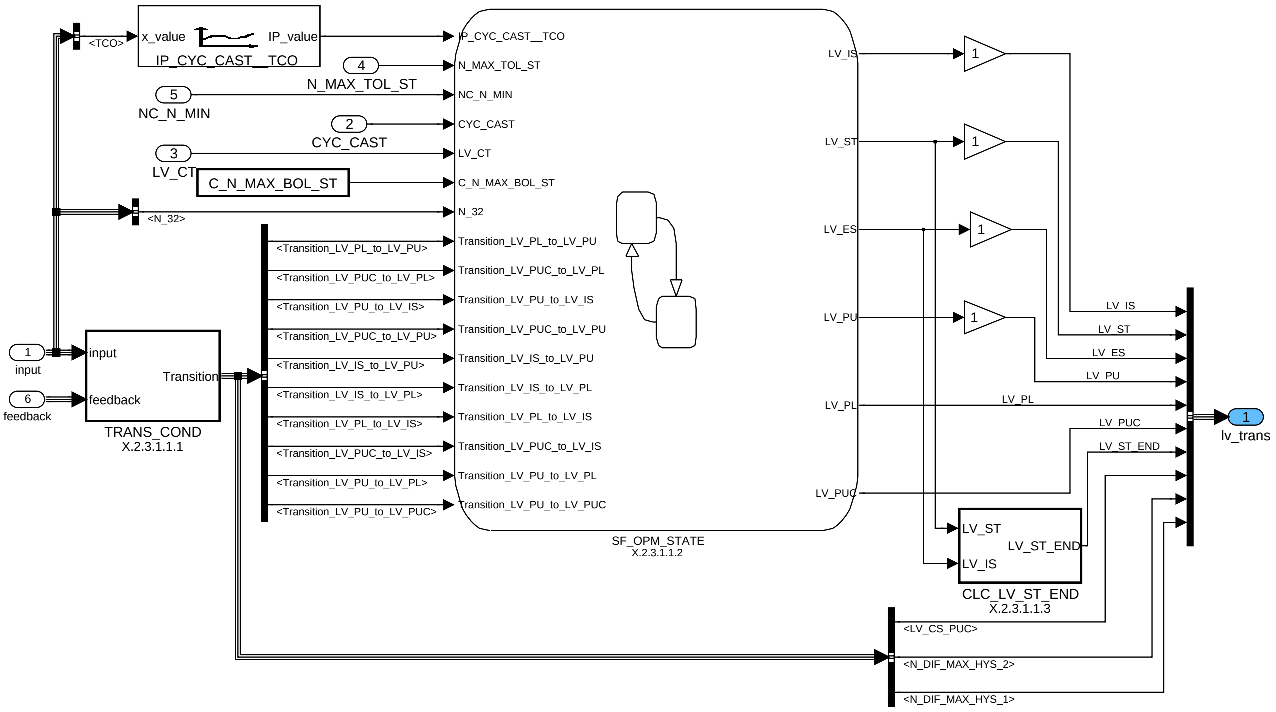


Figure 14:

20.2.2.3.1.1.1 Boolean condition calculations for Stateflow

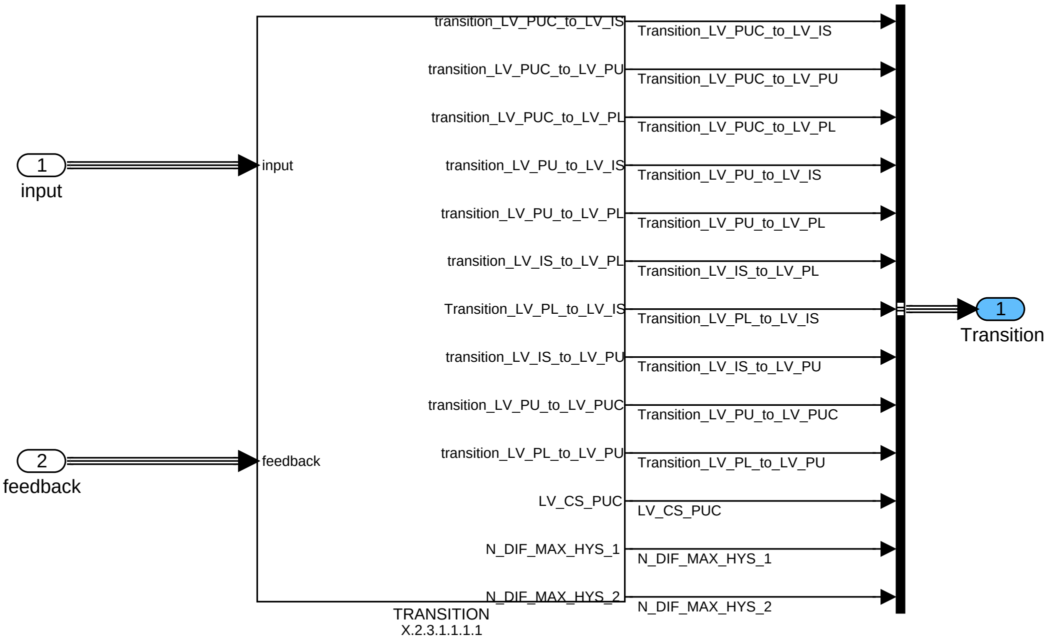


Figure 15:

20.2.2.3.1.1.1.1 Calculation of Engine Transition Conditions

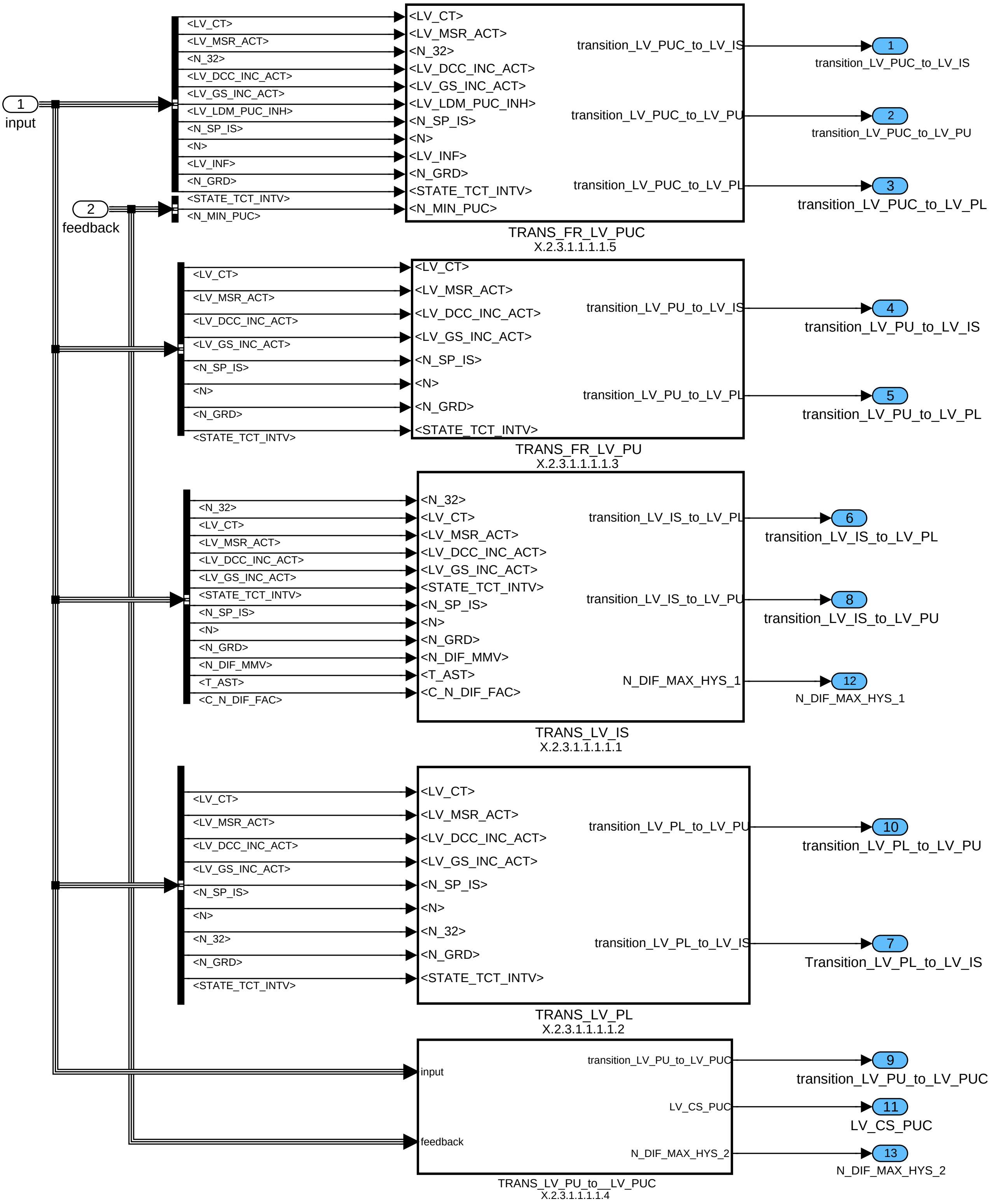


Figure 16:

20.2.2.3.1.1.1.1.1 Calculation of Transition Condition

Engine operating state : Idle Speed (LV_IS)The Transition condition for various To facilitate this documentation, N_SP_IS and N_DIF_MMV are described in the chapter System Variables . The following states are calculated in this part
Exit to LV_PL : Part Load, Exit to LV_PU : Trailing Throttle
states are calculated below.

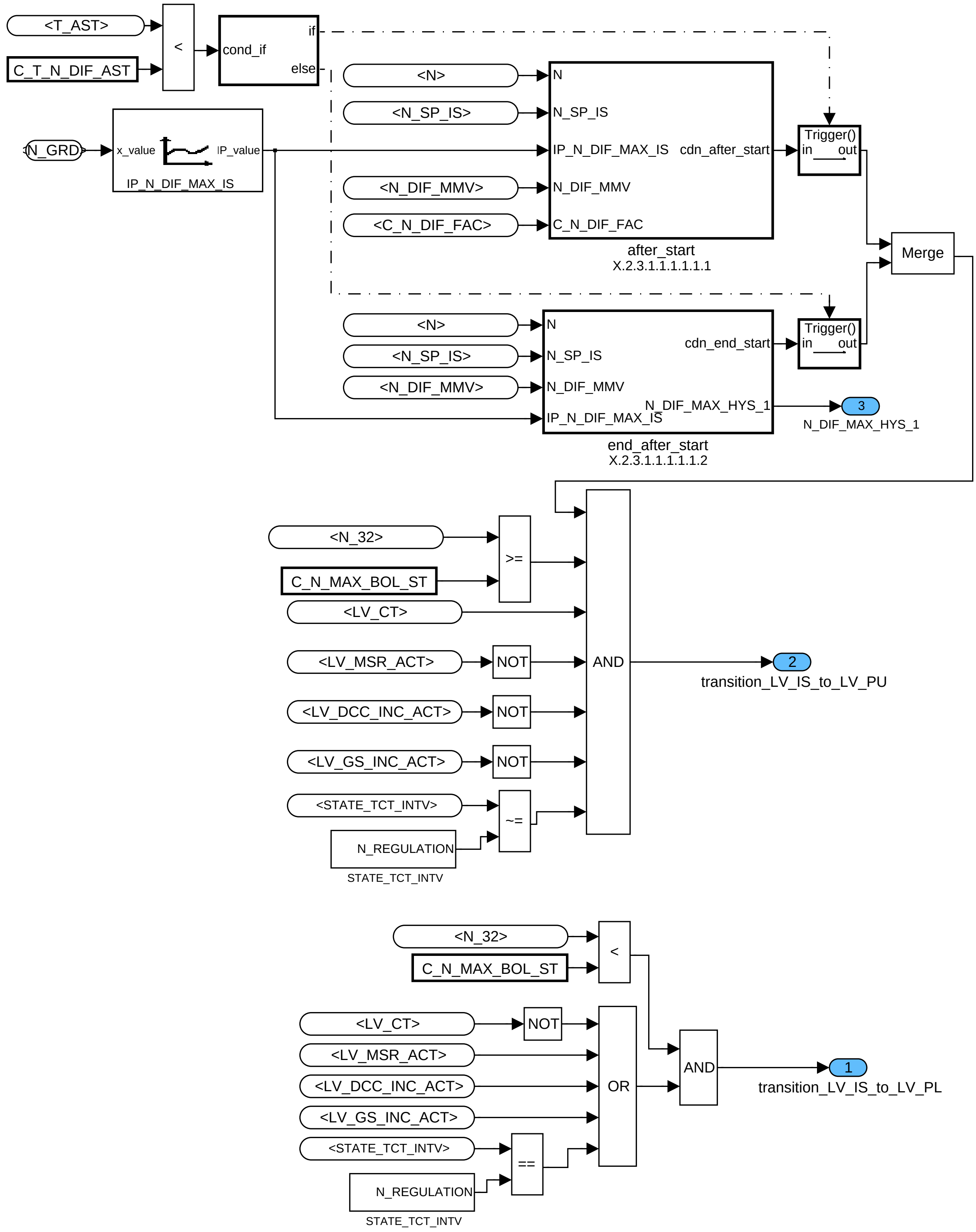


Figure 17:

20.2.2.3.1.1.1.1.1.1 Condition 1

This condition represents AFTER START

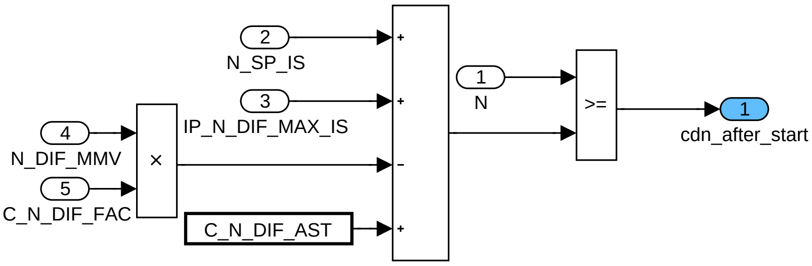


Figure 18:

20.2.2.3.1.1.1.1.2 Condition 2

This condition represents END AFTER START

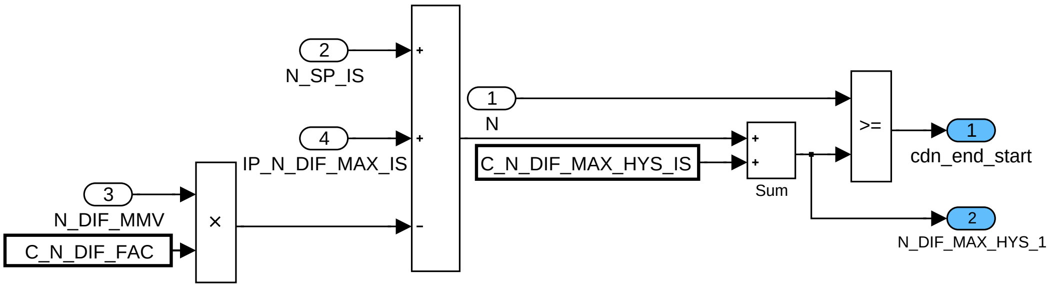


Figure 19:

20.2.2.3.1.1.1.1.2 Engine operating state : "Part Load" (LV_PL)

To facilitate this documentation, N_SP_IS is described in the chapter System Variables . The following states are calculated in this subchapter Exit to LV_IS : Idle Speed, Exit to LV_PU : Trailing Throttle.

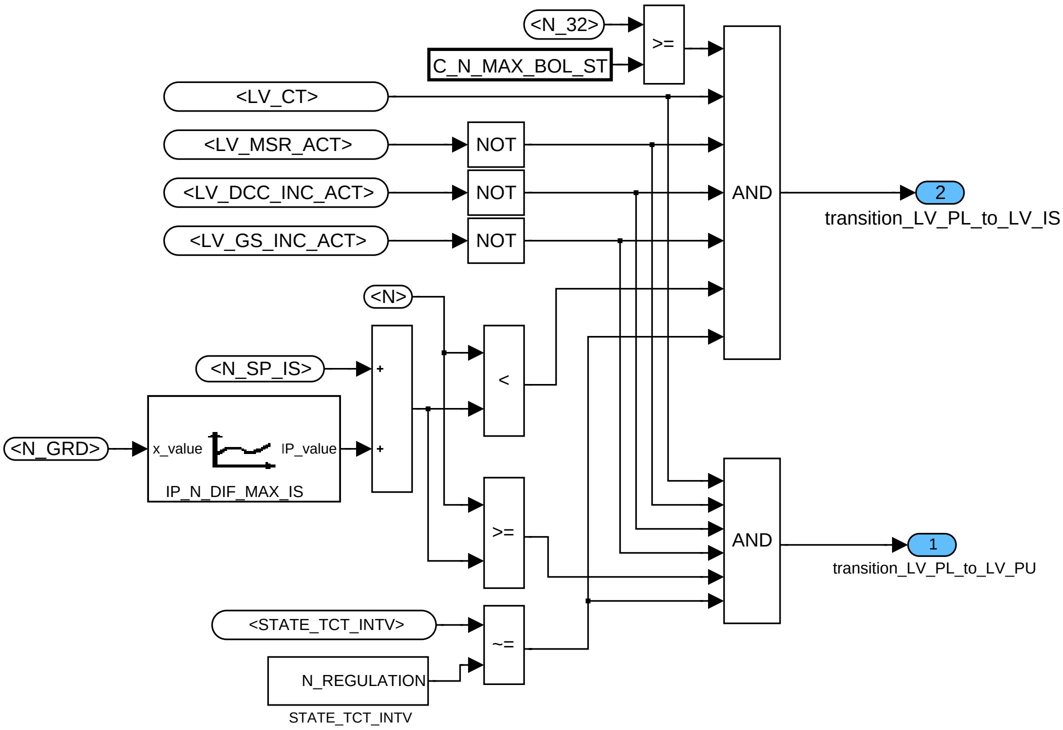


Figure 20:

20.2.2.3.1.1.1.1.3 Engine operating state : "Trailing Throttle" (LV_PU)

To facilitate this documentation, N_GRD and N_SP_IS are described in the chapter System Variables . The following state transitions are calculated Exit to LV_PL : Part Load, Exit to LV_IS : Idle Speed

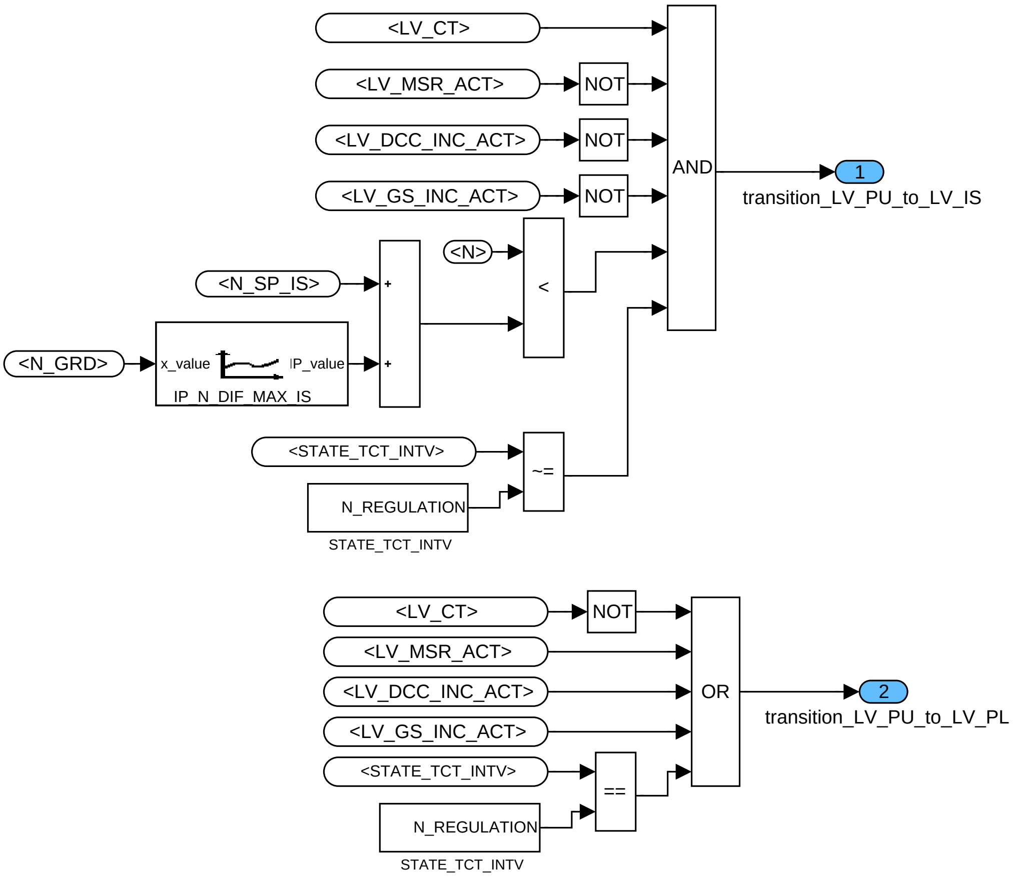
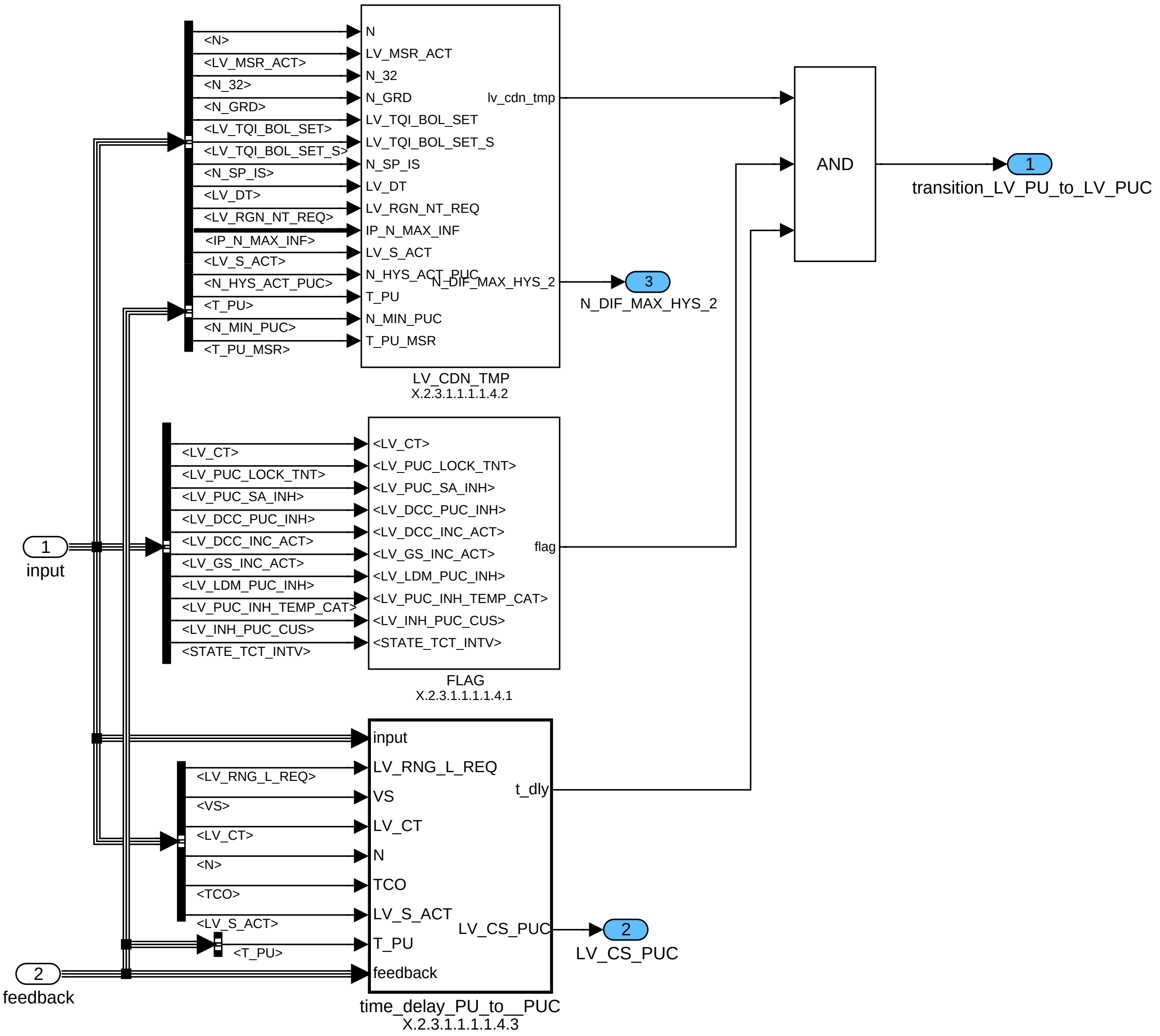


Figure 21:

20.2.2.3.1.1.1.1.4 Exit to LV_PUC : Trailing Throttle Fuel Cut Off

After active MSR function a delay time for entrance into PUC is started to suppress a jittering of MSR function.
T_PU_MSR started at LV_MSR_ACT 1 to 0

In order to avoid a sudden entry in PUC especially for TCT variant the flag LV_CS_PUC in dependence of STATE_ETCU_CLU is defined. The driveability in stop and go conditions is improved by using a delayed clutch switch for entry in PUC.



1
input
2
feedback
<N>
<LV_MSR_ACT>
<N_32>
<N_GRD>
<LV_TQI_BOL_SET>
<LV_TQI_BOL_SET_S>
<N_SP_IS>
<LV_DT>
<LV_RGN_NT_REQ>
<IP_N_MAX_INF>
<LV_S_ACT>
<N_HYS_ACT_PUC>
<T_PU>
<N_MIN_PUC>
<T_PU_MSR>
N
LV_MSR_ACT
N_32
N_GRD
LV_TQI_BOL_SET
LV_TQI_BOL_SET_S
N_SP_IS
LV_DT
LV_RGN_NT_REQ
IP_N_MAX_INF
LV_S_ACT
N_HYS_ACT_PUC
T_PU
N_MIN_PUC
T_PU_MSR
lv_cdn_tmp
N_DIF_MAX_HYS_2
LV_CDN_TMP
X.2.3.1.1.1.1.4.2
3
N_DIF_MAX_HYS_2
AND
1
transition_LV_PU_to_LV_PUC
<LV_CT>
<LV_PUC_LOCK_TNT>
<LV_PUC_SA_INH>
<LV_DCC_PUC_INH>
<LV_DCC_INC_ACT>
<LV_GS_INC_ACT>
<LV_LDM_PUC_INH>
<LV_PUC_INH_TEMP_CAT>
<LV_INH_PUC_CUS>
<STATE_TCT_INTV>
flag
FLAG
X.2.3.1.1.1.1.4.1
<LV_RNG_L_REQ>
<VS>
<LV_CT>
<N>
<TCO>
<LV_S_ACT>
<T_PU>
input
LV_RNG_L_REQ
VS
LV_CT
N
TCO
LV_S_ACT
T_PU
feedback
t_dly
LV_CS_PUC
2
LV_CS_PUC
time_delay_PU_to__PUC
X.2.3.1.1.1.1.4.3

Figure 22:

20.2.2.3.1.1.1.1.4.1 Computation of Flag

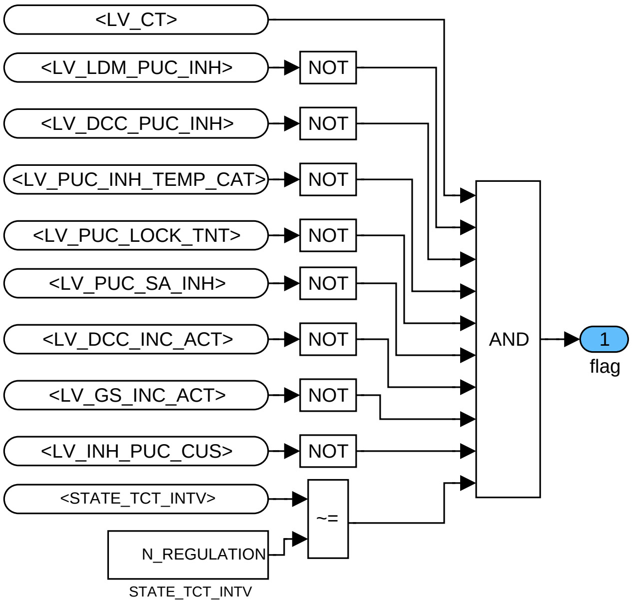


Figure 23:

20.2.2.3.1.1.1.1.4.2 Calculation of flag

In order to avoid a sudden entry in PUC especially for SMG variant the flag LV_CS_PUC in dependence of STATE_CLU_AMT is defined. The driveability in stop and go conditions is improved by using a delayed clutch switch for entry in PUC.

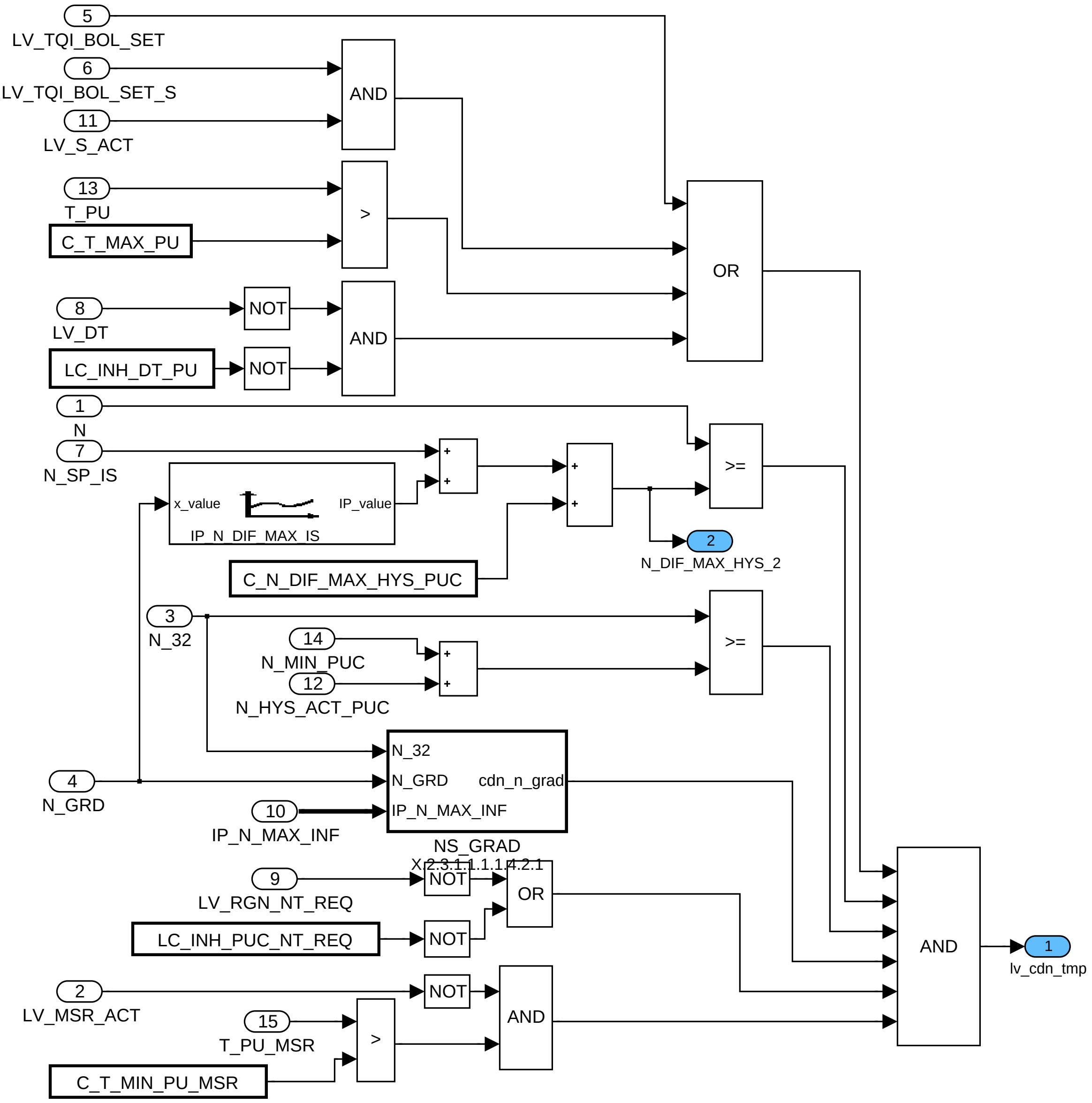


Figure 24:

Calculation of N_GRD

The following condition is satisfied in this block
N_GRD (C_N_GRD_MIN   (only if N_32 < IP_N_MAX_INF)

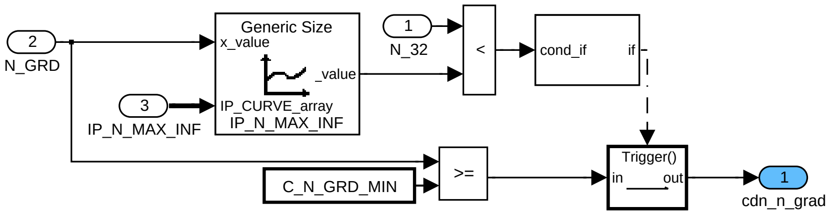


Figure 25:

20.2.2.3.1.1.1.1.4.3 The calculation below shows the transition from LV_PU to LV_PUC

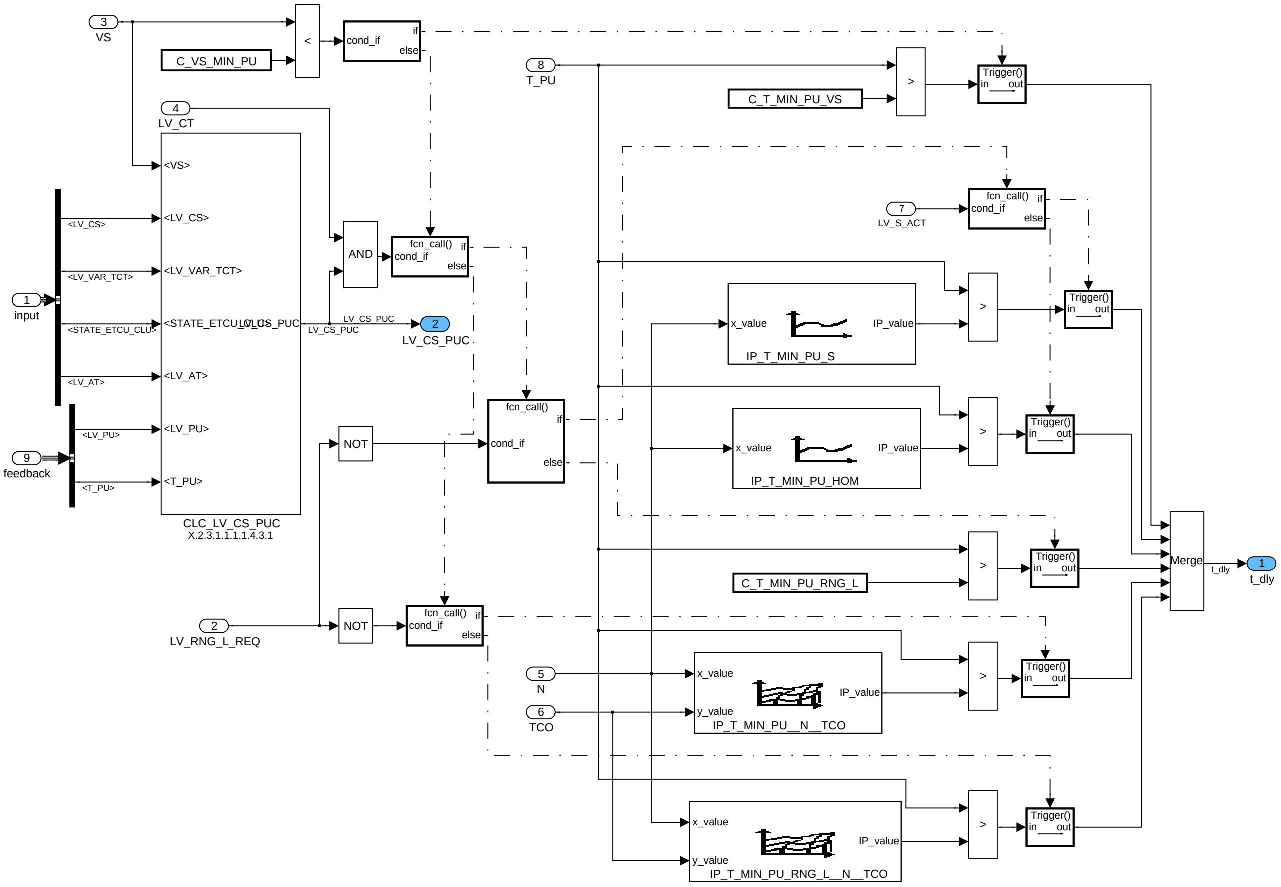


Figure 26:

Calculation of LV_CS_PUC

LV_CS_PUC is calculated based on Vehicle speed.

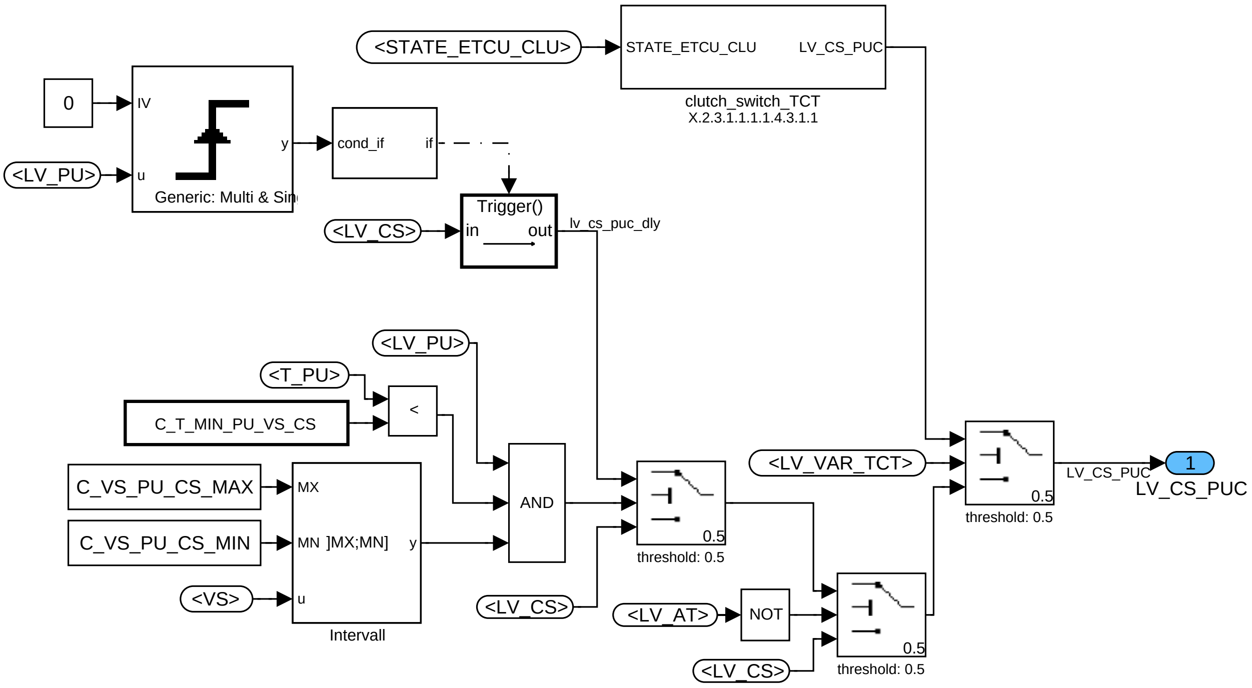


Figure 27:

Bit Wise Operator

LV_CS_PUC_RAW is calculated in this block for the gearbox variant twin clutch transmission

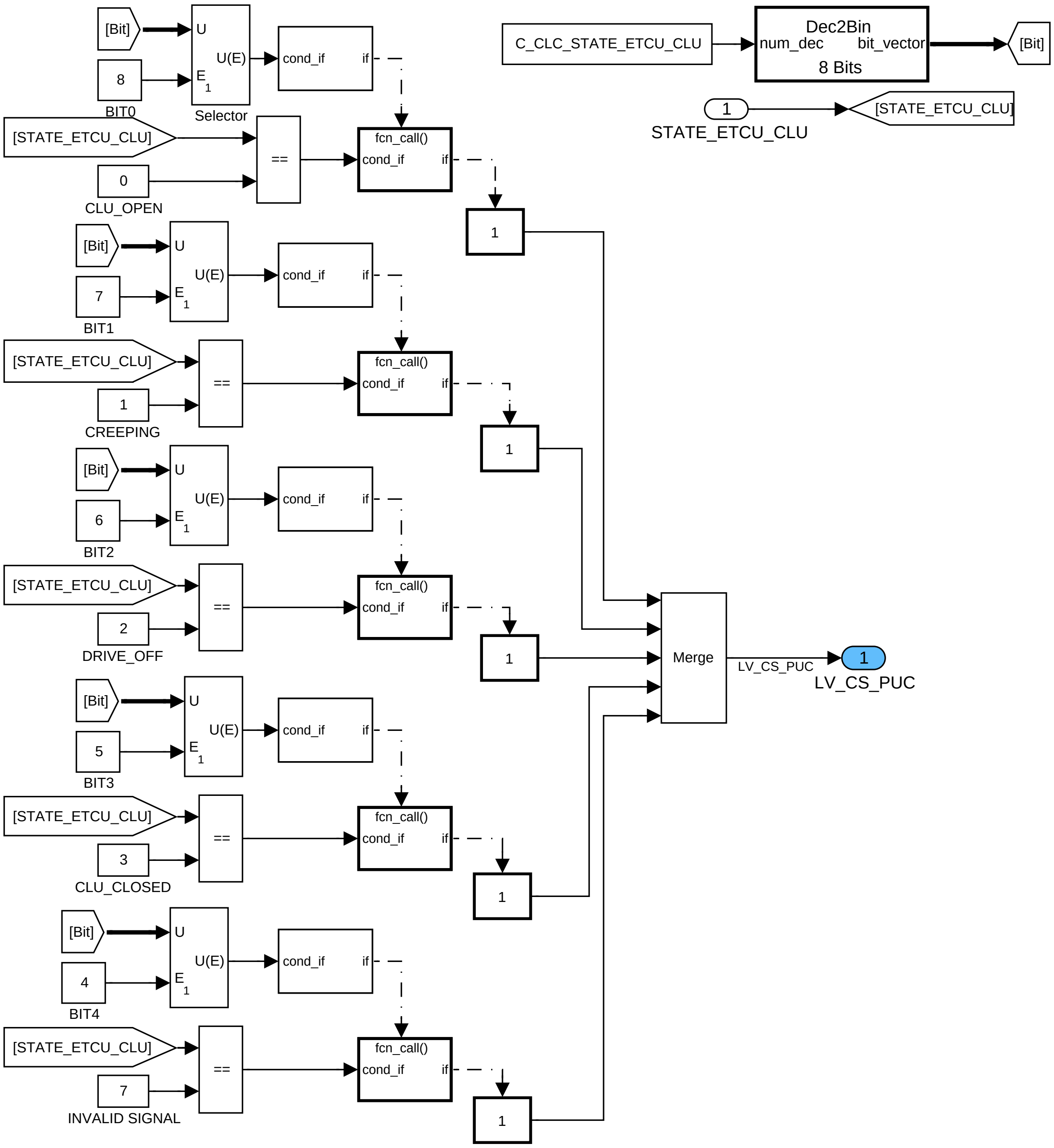


Figure 28:

20.2.2.3.1.1.1.1.5 Calculation of Transition from LV_PUC

The following transitions are involved in LV_PUC state Exit to LV_IS : Idle Speed, Exit to LV_PU : Trailing Throttle, Exit to LV_PL: Part Load

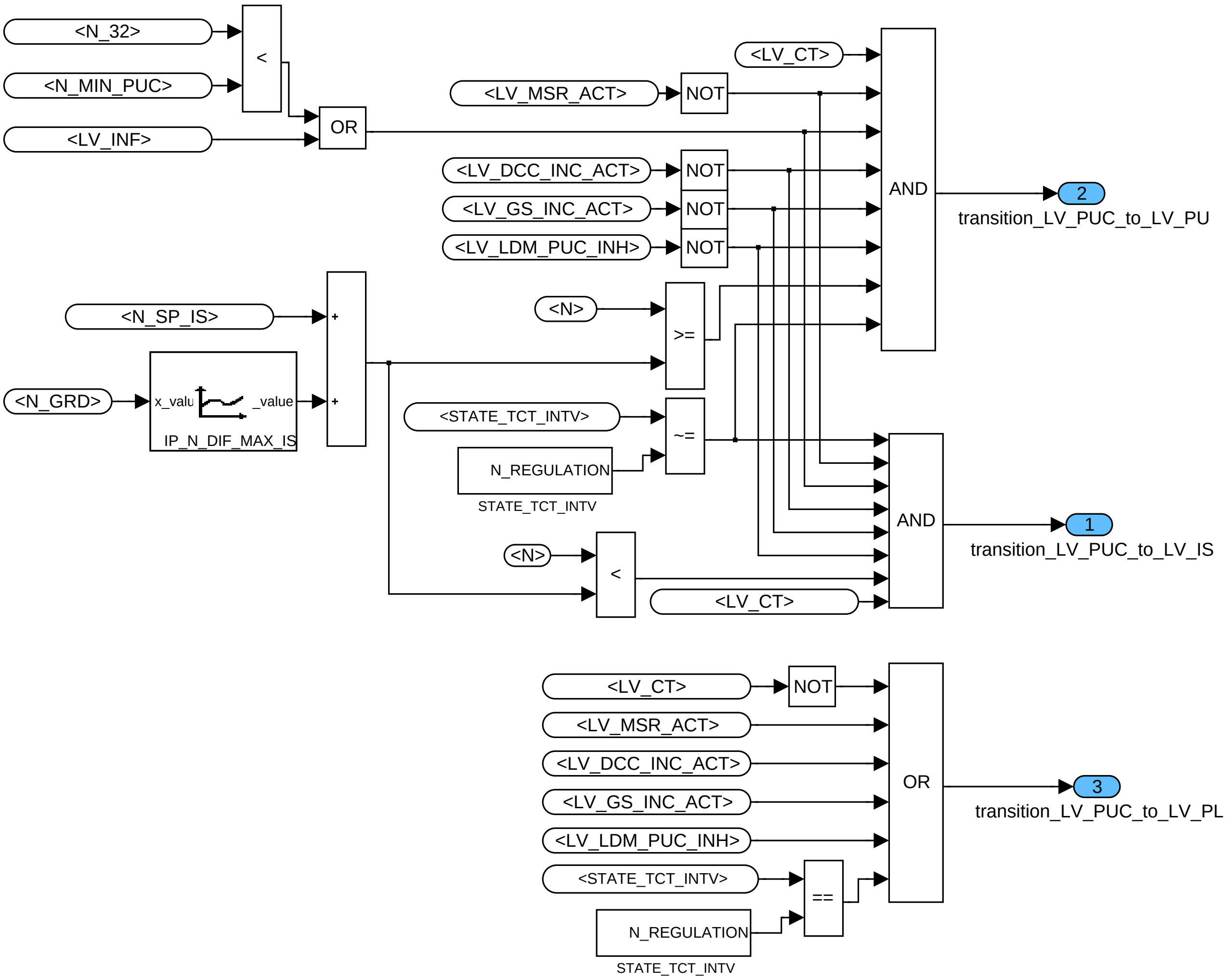

<N_32>
<N_MIN_PUC>
<LV_INF>
<
OR
<LV_CT>
<LV_MSR_ACT>
NOT
<LV_DCC_INC_ACT>
NOT
<LV_GS_INC_ACT>
NOT
<LV_LDM_PUC_INH>
NOT
AND
2
transition_LV_PUC_to_LV_PU
<N_SP_IS>
+
<N_GRD>
x_valu
_value
IP_N_DIF_MAX_IS
+
<N>
>=
<STATE_TCT_INTV>
~=
N_REGULATION
STATE_TCT_INTV
AND
1
transition_LV_PUC_to_LV_IS
<N>
<
<LV_CT>
<LV_CT>
NOT
<LV_MSR_ACT>
<LV_DCC_INC_ACT>
<LV_GS_INC_ACT>
<LV_LDM_PUC_INH>
<STATE_TCT_INTV>
==
N_REGULATION
STATE_TCT_INTV
OR
3
transition_LV_PUC_to_LV_PL

Figure 29:

20.2.2.3.1.1.2 SF_OPM_STATE

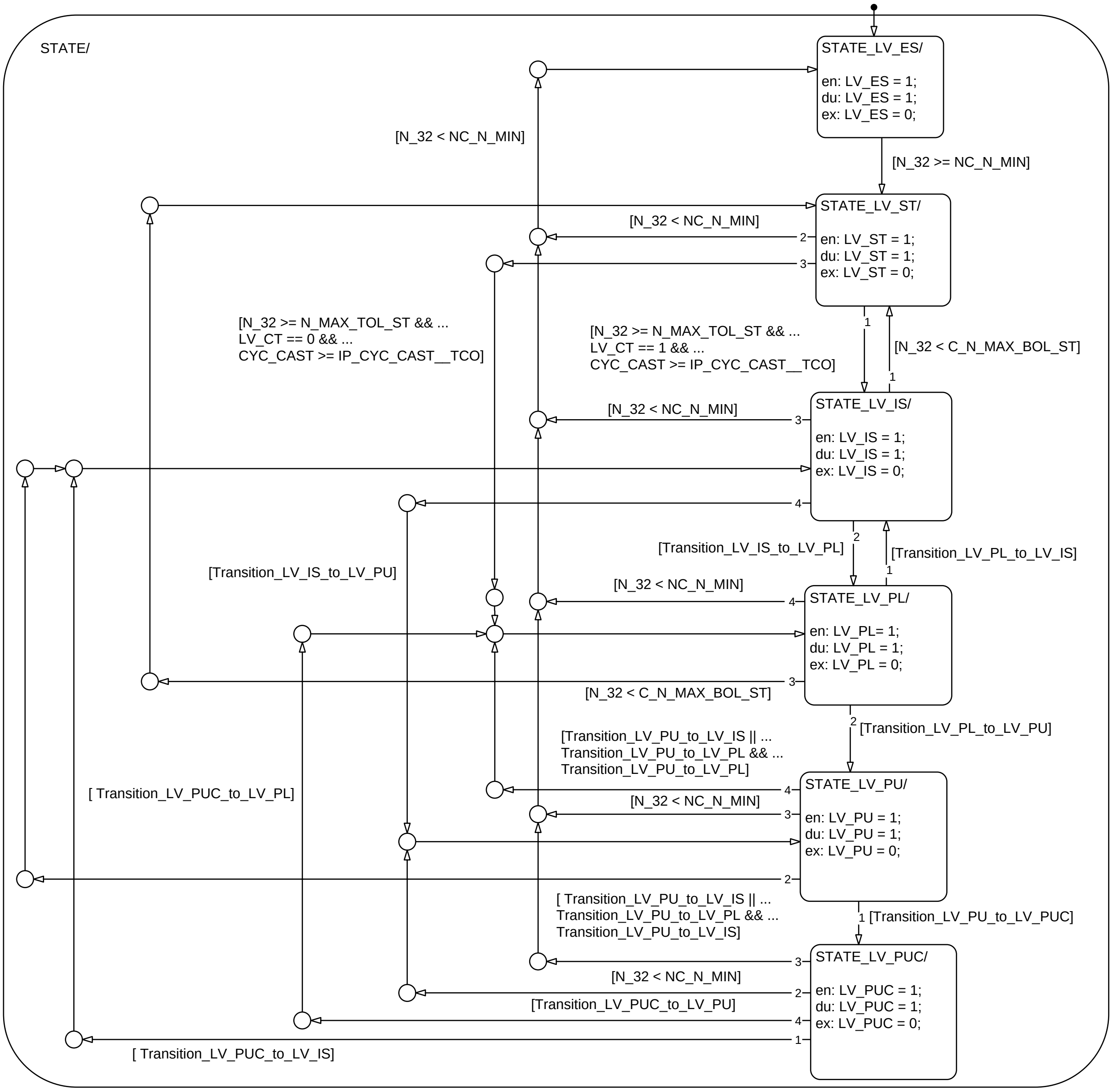


Figure 30:

20.2.2.3.1.1.3 Engine operating state : "Start" (LV_ST)

Calculation of LV_ST_END
The engine operating state Start (LV_ST) (cold start and hot start) is detected from all engine operating states via means of engine speed. The boolean LV_ST_END displays the transition from start to running engine .

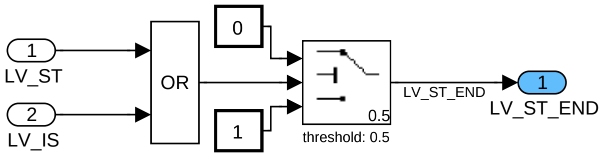


Figure 31:

20.2.2.3.1.2 Conditions for minimum ignition angle PUC:

LV_IGA_MIN_PUC is calculated in this sub block

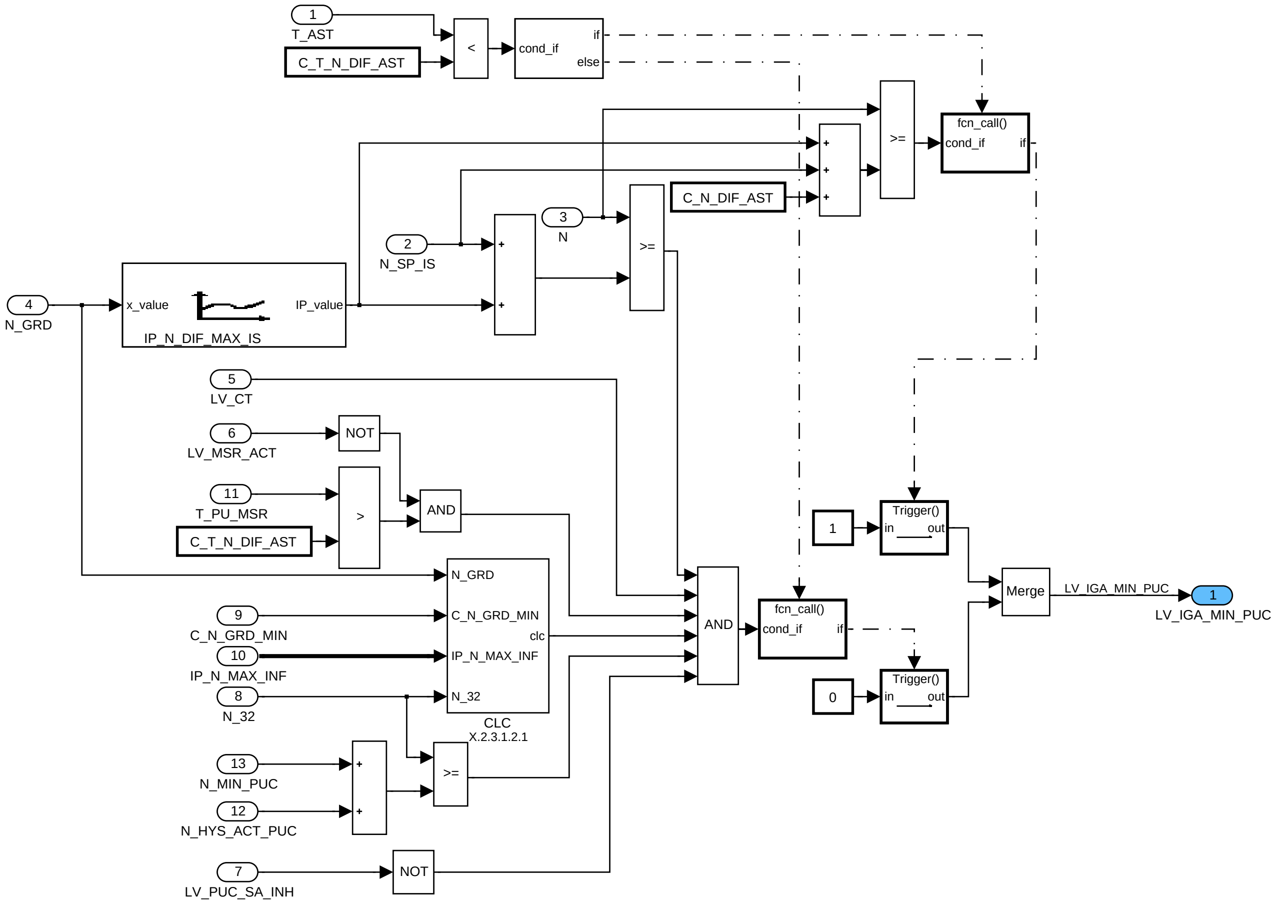


Figure 32:

20.2.2.3.1.2.1 N_GRD and C_N_GRD_MIN Check

The flag is set if and only if N_32 < IP_N_MAX_INF

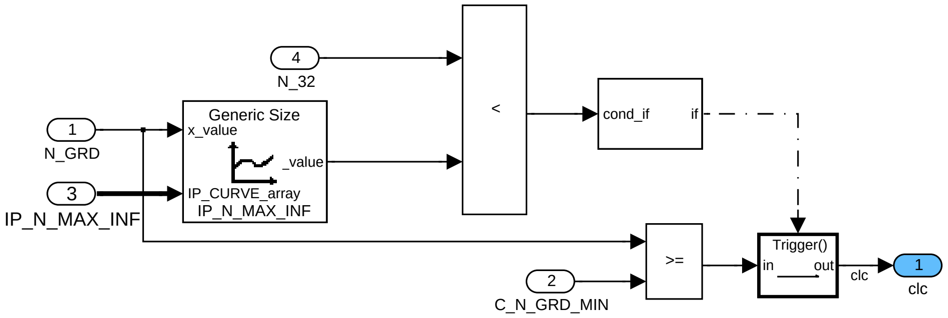


Figure 33:

20.2.2.3.1.3 Calculation of N_HYS variables

To facilitate this documentation, N_SP_IS is described in the chapter System Variables .

To facilitate this documentation, N_SP_IS is described in the chapter System Variables .

The engine speed threshold for termination of PUC (reactivation of injection at decreasing engine speed) is the reactivation engine speed N_MIN_PUC. It is dependant from TCO, gear ratio, vehicle type (transmission) and TQ_LOSS_ADD (additional torque losses). ARS and PSTE torque request maps remain preserved to keep the opportunity of separate treatment.

Function 'LV_PUC Hysteresis':
To avoid toggling PUC <> PU or PUC <> IS there is a hysteresis for the engine speed threshold which determines this transition. Transition from PUC when N < threshold (N_MIN_PUC); transition to PUC when N > threshold + hysteresis (N_MIN_PUC + N_HYS_ACT_PUC).
When entering Trailing Throttle Fuel Cut Off (LV_PUC), the engine speed hysteresis is set to a larger value, ID_N_HYS_MAX_PUC__GR_xx (xx = MT, AT), and a cycle counter is loaded with an adjustable number, C_CYCNR_HYS_PUC. The cycle counter is decremented once per program cycle in any engine operating state except 'Trailing Throttle Fuel Cut Off' (LV_PUC).
When the counter has reached its lower limit 0, the engine speed hysteresis is reset to the value ID_N_HYS_MIN_PUC__GR_xx. This value depends on gear GR_xx and vehicle speed VS and is the minimum engine speed hysteresis for LV_PUC. The hysteresis is reset to this minimum value also, if during counter decrementation the engine operating state changes to partload and this reset is enabled by calibration switch LC_N_HYS_CYL_CUT_OFF_xx_CONF.
If the coolant temperature is below the threshold C_TCO_MIN_PUC_ST when the engine is started, then the engine speed hysteresis for the detection of LV_PUC is increased to an adjustable value ID_N_HYS_PUC_ST__GR_xx, for an adjustable number of cycles C_CYCNR_HYS_PUC_ST. This results in faster catalyst heatup during trailing throttle operation.
For the customer request LV_INH_PUC_CUS = 1, the engine operating state LV_PUC is inhibited.
Description:

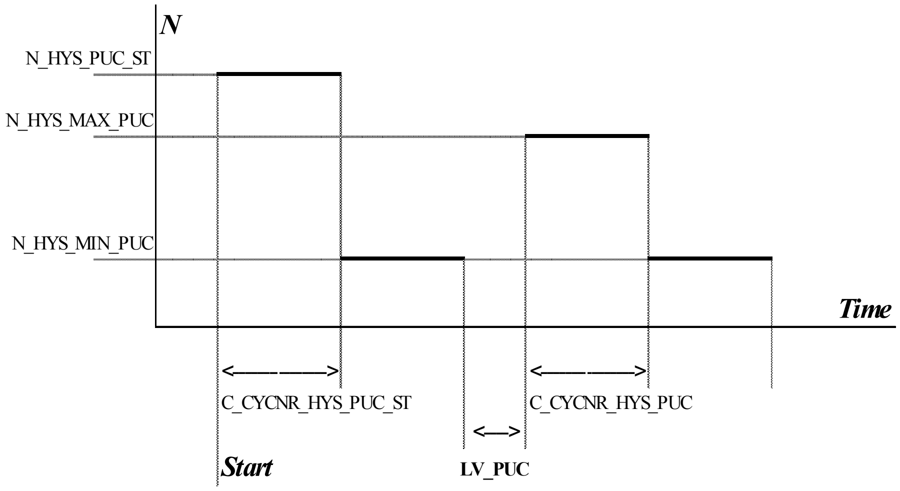


Figure 1: overview N hysteresis for PUC versus time

20.2.2.3.1.3.1 Engine operating state : "Trailing Throttle Fuel Cut Off" (LV_PUC)

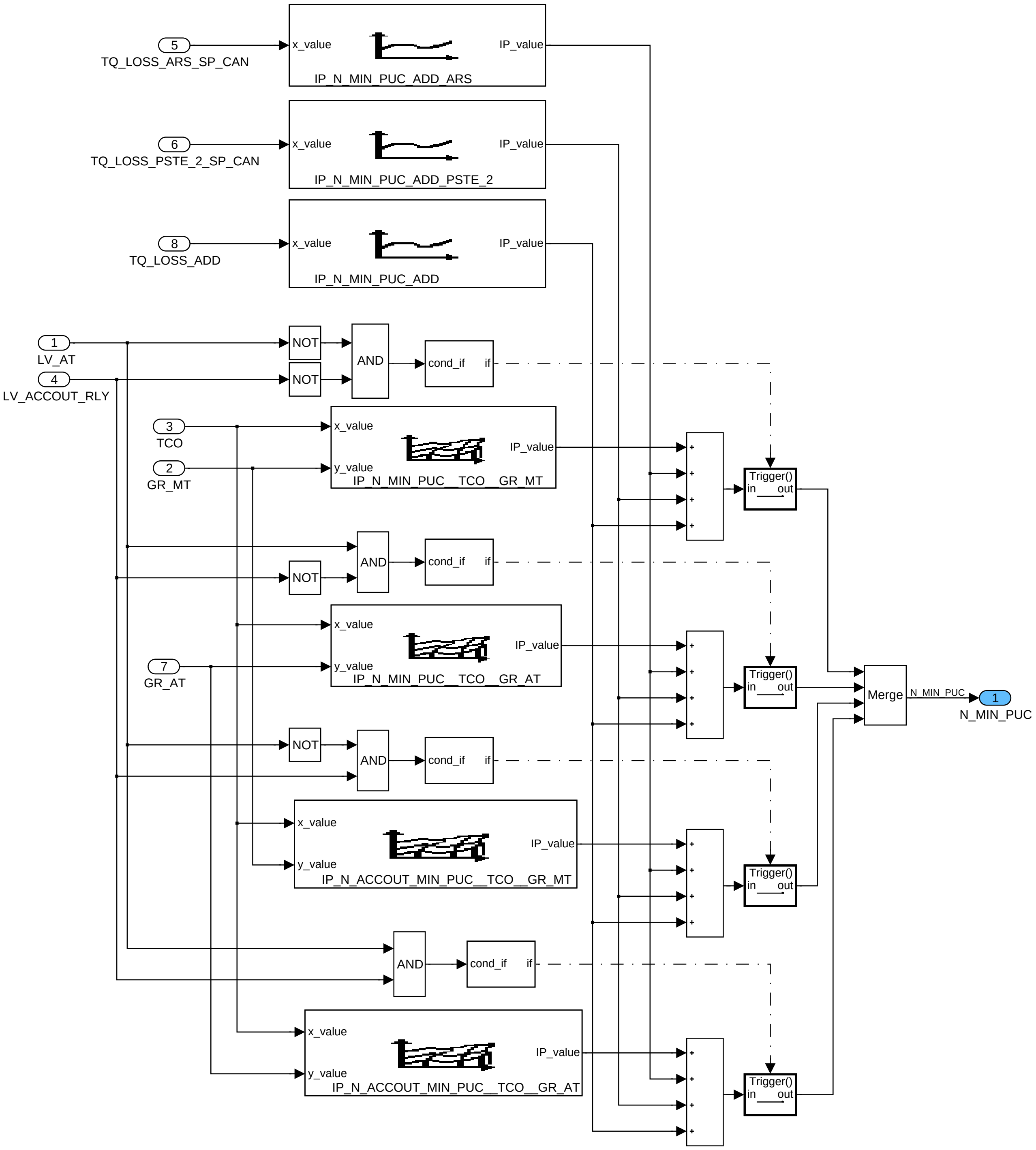


Figure 34:

20.2.2.3.1.3.2 Calculation of minimum engine speed hysteresis for LV_PUC:

Calculation of N_HYS_MIN_PUC

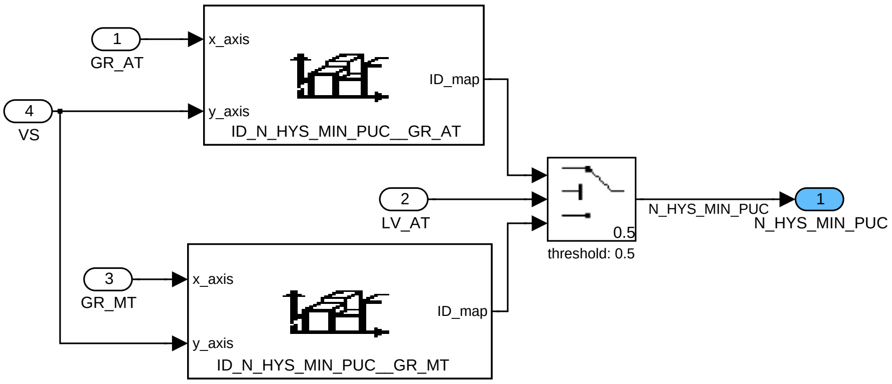


Figure 35:

20.2.2.3.1.4 Calculation of actual effective engine speed hysteresis for LV_PUC

N_HYS_ACT_PUC is calculated in this block.

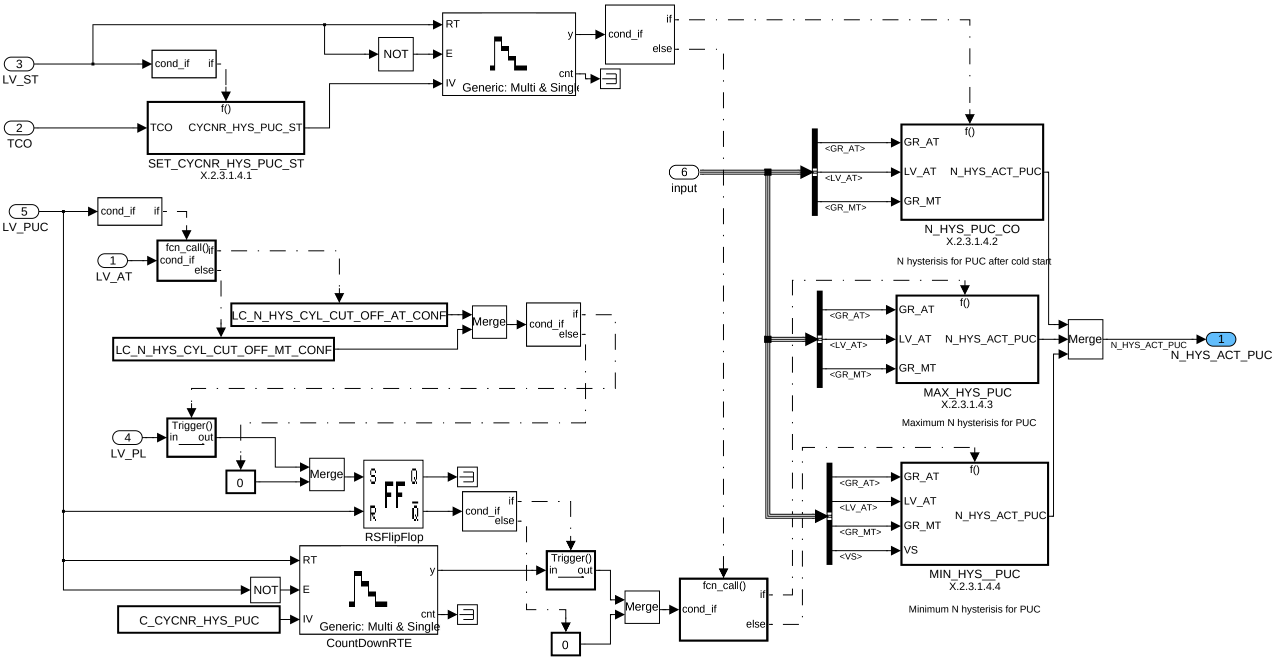


Figure 36:

20.2.2.3.1.4.1 SET_CYCNR_HYS_PUC_ST

CYCNR_HYS_PUC_ST is set.

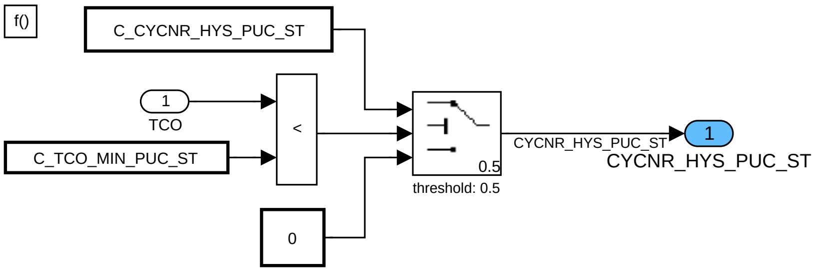


Figure 37:

20.2.2.3.1.4.2 Calculation at Coldstart

N_HYS_ACT_PUC is calculated in this block

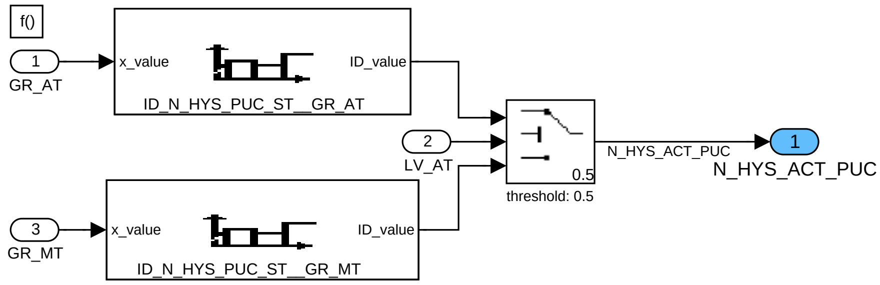


Figure 38:

20.2.2.3.1.4.3 Maximum Speed Hysteresis for PUC

N_HYS_ACT_PUC is calculated in this block.

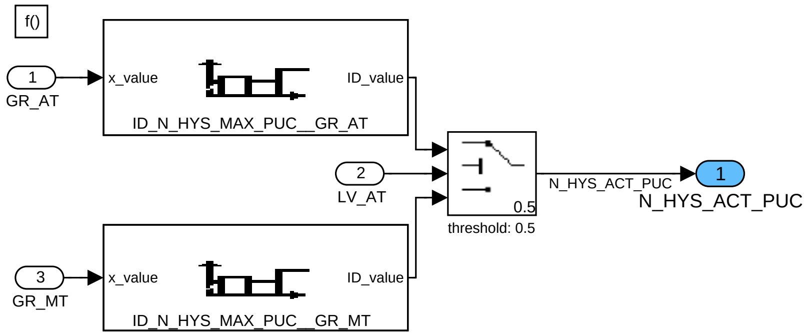


Figure 39:

20.2.2.3.1.4.4 Minimum Speed Hysteresis for PUC

N_HYS_ACT_PUC is calculated in this block.

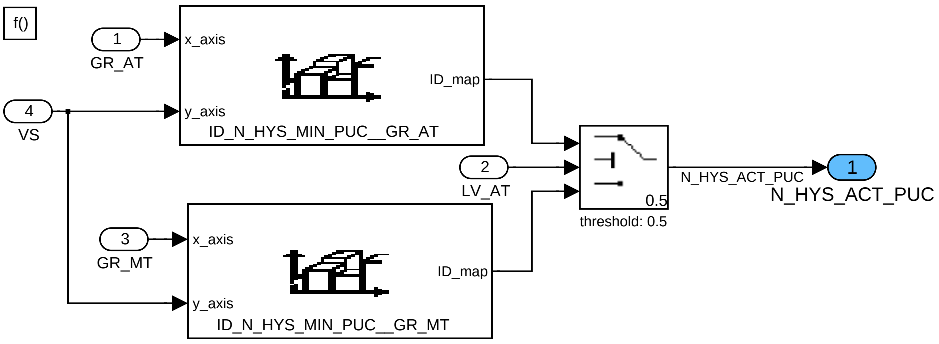


Figure 40:

20.2.2.3.1.5 Logical variable trailing throttle fuel cutoff request LV_PUC_REQ

LV_PUC_REQ is set at trailing throttle conditions. It is evaluated in the module Minimum torque at clutch .

20.2.2.3.1.5.1 No title given

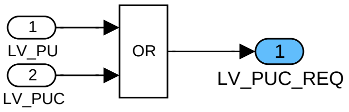


Figure 41:

20.2.2.3.1.6 Engine operating state : "Status byte" (STATE_ENG)

The statusbyte STATE_ENG gives a quick overview of the engine operating state for e.g. calibration.

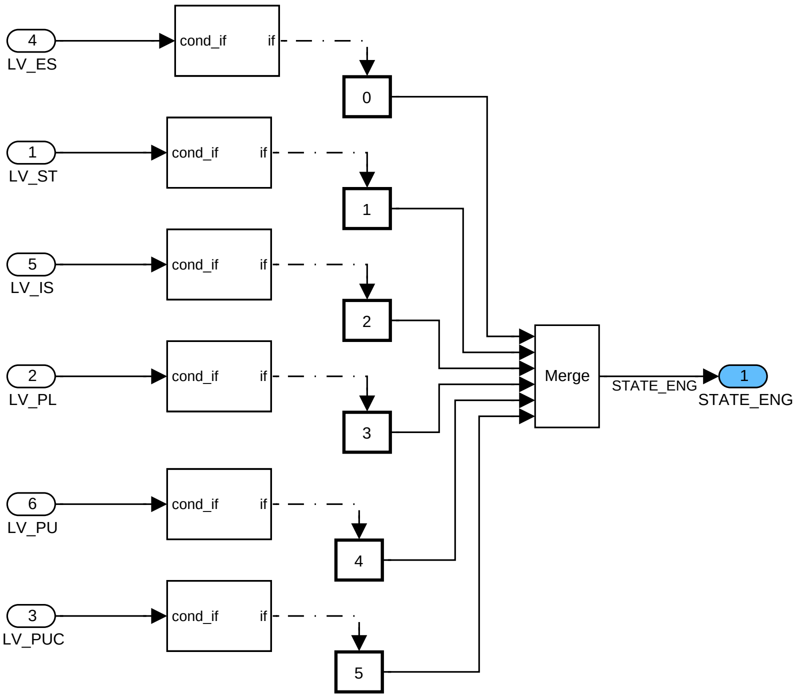


Figure 42:

20.2.2.3.1.7 Reset engine state trigger flag

lv_engstate_trig is reset to 0.

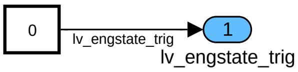


Figure 43:

20.3 Operation mode request from tester

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
OPM_EXT_REQ	O/V	0...FFFFH	0...6.5535E+4	1	-
Operation mode request from tester					
OPM_EXT_REQ_TMP	-	0...FFFFH	0...6.5535E+4	1	-
Operation mode request from tester without EOL request					

Input data:

LV_ACT_VLS_EOL_EXT_ADJ	LV_IGK	STATE_HOM_AFS_REQ_EXT_ADJ	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_OPM_EXT_REQ_1	1	0...FFFFH	0...6.5535E+4	1	-
Constant 1 for operation mode request from tester					
C_OPM_EXT_REQ_2	1	0...FFFFH	0...6.5535E+4	1	-
Constant 2 for operation mode request from tester					
C_OPM_EXT_REQ_3	1	0...FFFFH	0...6.5535E+4	1	-
Constant 3 for operation mode request from tester					
C_OPM_EXT_REQ_4	1	0...FFFFH	0...6.5535E+4	1	-
Constant 4 for operation mode request from tester					
C_OPM_EXT_REQ_5	1	0...FFFFH	0...6.5535E+4	1	-
Constant 5 for operation mode request from tester					
C_OPM_EXT_REQ_6	1	0...FFFFH	0...6.5535E+4	1	-
Constant 6 for operation mode request from tester					
C_OPM_EXT_REQ_7	1	0...FFFFH	0...6.5535E+4	1	-
Constant 7 for operation mode request from tester					
C_OPM_EXT_REQ_8	1	0...FFFFH	0...6.5535E+4	1	-
Constant 8 for operation mode request from tester					
C_OPM_EXT_REQ_VLS_EOL	1	0...FFFFH	0...6.5535E+4	1	-
Constant for operation mode request from tester (end of line test)					

20.3.1 FUNCTION DESCRIPTION:

General Information:

Via tester an operation mode (e.g. homogenous) can be requested. This is necessary e.g. for end of line tests. This information is used in this module, to switch between several calibratable operation modes. The generated information word OPM_EXT_REQ is send to the customer- software via layer.

Application Condition

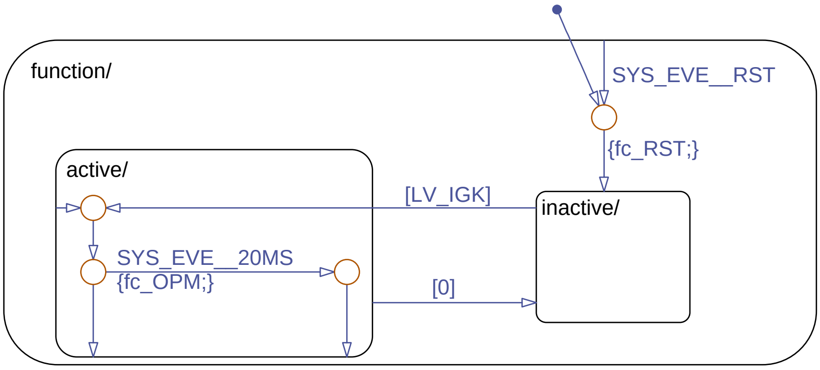
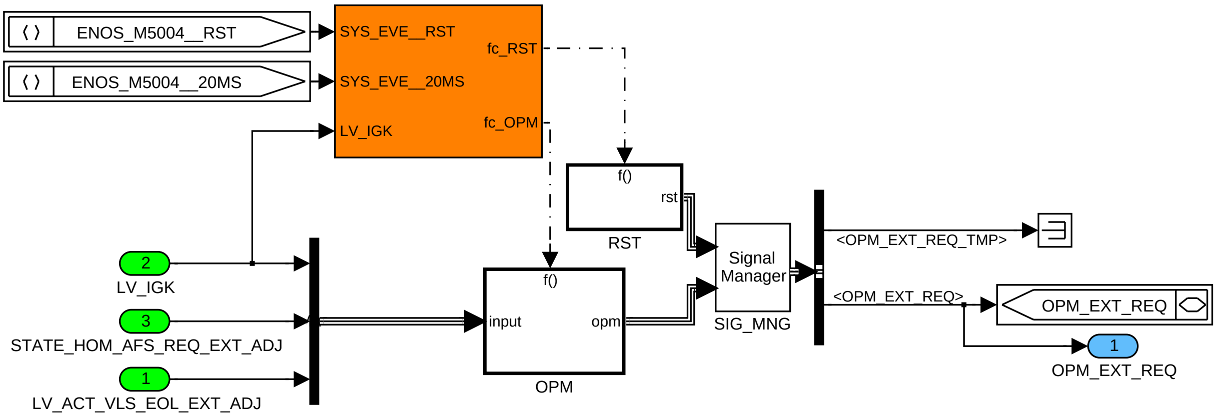


Figure 44 ENOS_M5004/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 21−Sep−2005

Figure 45 ENOS_M5004

20.3.1.1 Initialization

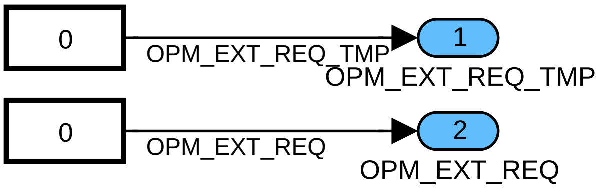


Figure 46 ENOS_M5004/ RST/ INI

20.3.1.2 Formula section

Calculation

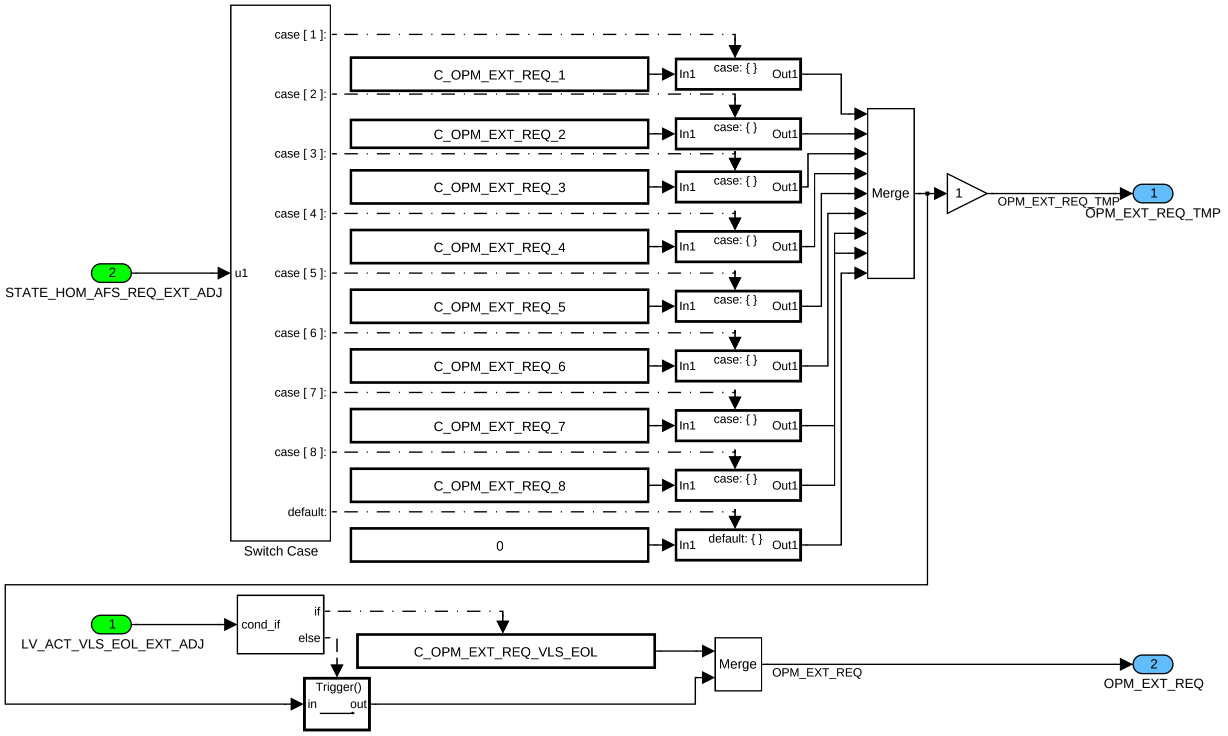


Figure 47 ENOS_M5004/ OPM/ CLC_OPM_EXT_REQ

20.4 Customer adaptation module : ENOS

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_sa	-	0...1H	0...1	1	[-]
1: pull fuel cut off					
B_sae	V/O	0...1H	0...1	1	[-]
1: trailing throttle					
B_start	V/O	0...1H	0...1	1	[-]
1: start condition					
B_startende	V/O	0...1H	0...1	1	[-]
1: start left to running engine					
Nwe	V/O	0...7FFFH	0...32767	1	[rpm]
reactivation engine speed					
LV_INH_PUC_CUS	V/O	0...1H	0...1	1	[-]
Flag indicating "Bedingung Schubabschaltung verboten"					

Input data:

LV_PUC	LV_ST_END	LV_PU	B_ums_vbsa
N_MIN_PUC			

20.4.1 Outputs for BMW which are defined as ENOS exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: 0 , **except**

B_start = 1

B_startende = 0 at reset and at transition LV_ES 0 -> 1

Recurrences: 10 ms: B_startende, B_start

20 ms: B_sa, B_sae

Activation: every engine state

Formula section:

B_sa = LV_PUC

B_sae = LV_PU

B_start = **!** LV_ST_END

B_startende = LV_ST_END

20.4.2 Outputs for BMW which are not defined as ENOS exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: 0

Recurrences: 20 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
if    LV_IGK = 1
  then     Nwe  =  N_MIN_PUC
  else     Nwe  =  0
endif
```

20.4.3 Outputs for SV aggregates

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: at reset : 0
at power latch phase : 0

Recurrence: 100 ms

Activation: at every engine state

Formula section:

LV_INH_PUC_CUS = B_ums_vbsa

21 Engine roughness determination

Chapter	Baseline		
Engine roughness determination	4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Pages
3563 of 9643

Table of Contents

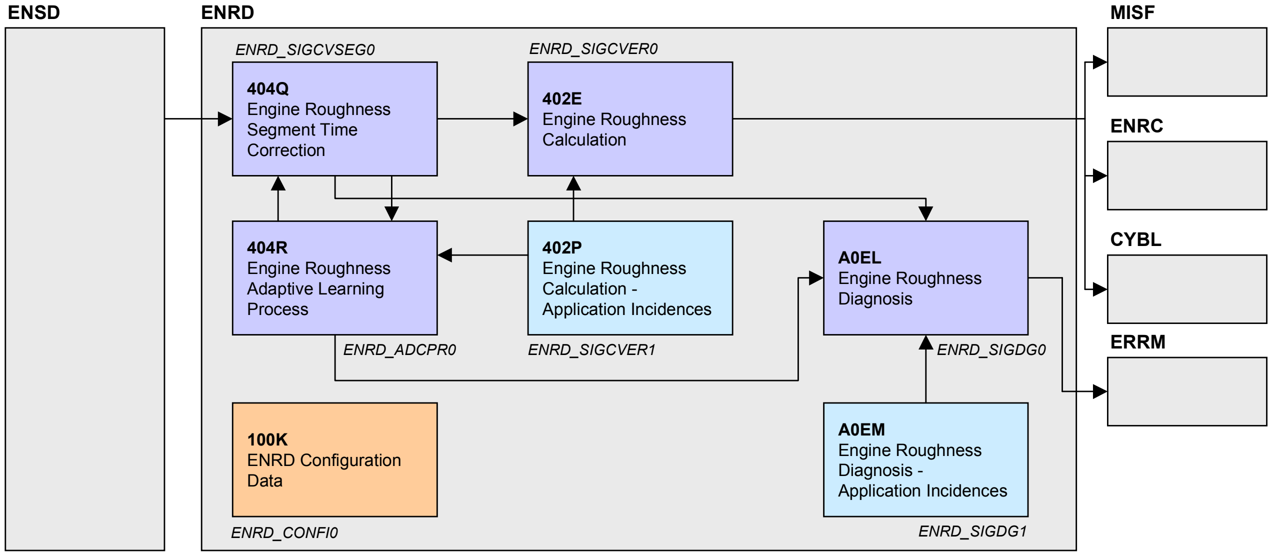
Index

21.1 ENRD General

21.1.1 General description

The goal of the ENRD is to provide an engine roughness information to functions linked to combustion process like misfire monitoring, engine roughness control and cylinder balancing.

21.1.2 Architecture Overview



21.1.3 Description of the containing functions

21.1.3.1.1 404Q Engine roughness segment time correction

Segment times coming from ENSD are managed and corrected according adaptive learning values obtained in 404R module.

21.1.3.1.2 402P Engine roughness calculation - Application Incidences

Application specific requirements, fade-out requests are handled to act on the generic function core. (Template module, modified by the project according its integration environment)

21.1.3.1.3 402E Engine roughness calculation

Engine roughness indexes calculation is grounded on corrected segment time samples provided by 404Q module.

21.1.3.1.4 404R Engine roughness adaptive learning process

Engine roughness adaptive learning process provides adaptive values to 404Q module, to minimise noises produced by flywheel mechanical tolerances.

This module also consumes fade-out requests coming from 402P module.

21.1.3.1.5 A0EM Engine roughness diagnosis - Application Incidences

Application specific requirements, fade-out requests are handled to act on the generic engine roughness diagnosis function. (Template module, modified by the project according its integration environment)

21.1.3.1.6 A0EL Engine roughness diagnosis

This modules provides two diagnosis :

A diagnosis for the ER segment acquisition (thin synchronisation default criterion).

A check range diagnosis for the ER adaptive values obtained by the 404R module.

21.1.3.1.7 100K ENRD Configuration data

This module defines ENRD configuration data used during software integration & compilation. These configuration options defines mainly data and buffer size according the number of cylinders. This module is not attached to a software task.

21.2 ENRD Configuration Data

Input data:

NC_CYL_NR	NC_SEG_DLY_ER_MES	NLC_CONF_GAIN_ADD_ER	NC_SIZE_SEG_T_COR_BUF

General information :

The following describes the general rules for determination of the configuration data

21.2.1 Local configuration data

Here are listed the configuration data, which are used only in the ENRD aggregate.

Data	**Value**
NC_ENRD_VERS	**1**
NC_SIZE_SEG_T_COR_BUF	**7**
NC_SEG_DLY_ER_MES	**1**
NLC_CONF_GAIN_ADD_ER	**1**

Default configurations:

NC_CYL_NR	NC_SIZE_SEG_T_COR_BUF
3	7
4	9
5	11
6	7
8	9

Configuration data:

Name	**Dim**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
NC_ENRD_VERS	1	0...FFH	0...255	1	[-]
ENRD aggregate version					

21.3 Engine roughness segment time correction

21.3.1 Phasing reference for the acquisition of engine roughness segment time

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SEG_NR_MES	V/O	0...7H	0...7	1	[-]
Phasing reference of ER segment acquisition					

Input data:

SEG_NR	NC_CYL_NR	LV_SYN_ENG	

FUNCTION DESCRIPTION:

The ER segments provided by the aggregate ENSD, are used to observe in-cylinder combustion effects on crankshaft. The cylinder x combustion is located between cylinder X tdc and cylinder X+1 tdc. Consequently, at X+1 tdc, the T_SEG_ER obtained is relative to cylinder X combustion, a delay of 1 tdc between SEG_NR value at cylinder observed *(NC_SEG_DLY_ER_MES=1)*.

Exceptionally, if the segment is located after ER segment, this delay must be cancelled *(NC_SEG_DLY_ER_MES=0)*, because in such case the phasing reference SEG_NR is updated after T_SEG_ER acquisition.

SEG_NR_MES, the ER segments phasing reference, will be used to identify which cylinder is observed with the current engine roughness segment, it's particularly useful for ER segment adaptive values identification.

Application conditions:

Initialisation: *at ECU reset, Engine Stop* ***Or*** *Deactivation event*

SEG_NR_MES = 0

Recurrence: Segment

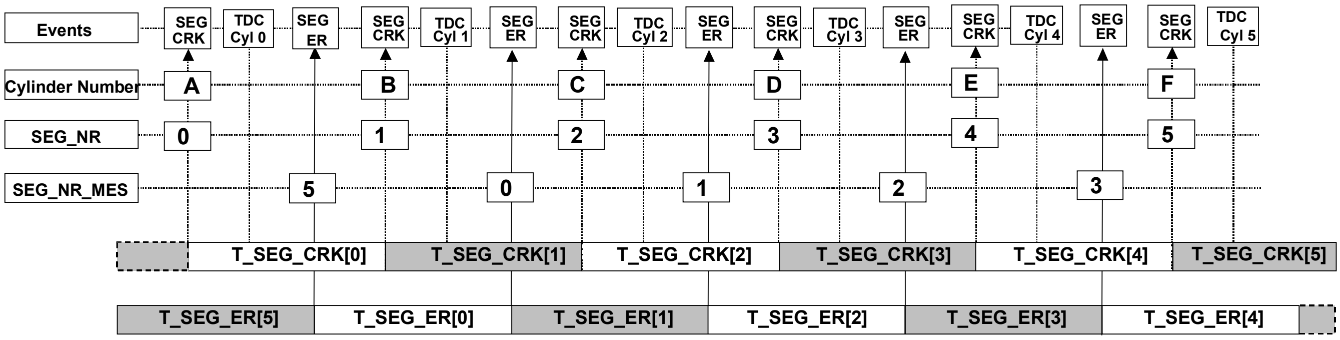
Activation: LV_SYN_ENG = 1

Deactivation: LV_SYN_ENG = 0

Formula section:

SEG_NR_MES = (SEG_NR – NC_SEG_DLY_ER_MES) % NC_CYL_NR

Implant scheme:



6 Cylinder Engine Example : ***NC_SEG_DLY_ER_MES = 1***

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_SEG_DLY_ER_MES	-	0...2H	0...2	1	[-]
Delay between ER segment and Cylinder measured segment					

21.3.2 Engine roughness segment time acquisition and control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SEG_T_MES	V	0...FFFFH	0...65535	1	[µs]
Engine roughness raw segment time relative to former TDC					
SEG_T_MES_CYL[NC_CYL_NR]	V/O	0...FFFFH	0...65535	1	[µs]
Buffer of cylinder engine roughness raw segment times					
LV_ENA_SEG_T_MES	V/O	0...1H	0...1	1	[-]
Engine roughness segment time validity flag					

Input data:

T_SEG_ER	LV_SYN_ENG	SEG_NR_MES	LV_INH_MIS_CRK
NC_CYL_NR	LV_ERR_CRK	LV_LIH_ERR_CRK	

General information:

Here's described the segment time buffers management & control

ER current raw segment time:

T_SEG_ER *(input from ENSD)*

Coming directly from the ENSD aggregate and update at each ER segment task according ER segment counter *(SEG_NR_MES)*.

Actual ER segment time sample is considered **unvalid** if :

- Engine synchronisation is disable (LV_SYN_ENG = 0)
- When on one crankshaft revolution at least one tooth less or more has been detected (LV_INH_MIS_CRK = 1 before antibounce counter),
- When the crahshaft error or limp home is activated (LV_ERR_CRK = 1 or LV_LIH_ERR_CRK = 1)
- SEG_T_MES obtained exceeds a defined valid range (C_SEG_T_MES_MIN & C_SEG_T_MES_MAX)

Cylinder ER raw segment times:

SEG_T_MES & SEG_T_MES_CYL[NC_CYL_NR] *(outputs)*

This is a buffer where T_SEG_ER values *(after data casting)* are stored according SEG_NR_MES counter. It allows having a memory of raw ER segments on a complete engine cycle. These values will be used by the ER adaptive learning process.

Application conditions:

Initialisation: *at ECU reset, Engine Stop* ***Or*** *Deactivation event*

```
SEG_T_MES = FFFFH
For x = 0 ... NC_CYL_NR-1
        SEG_T_MES_CYL[x] = FFFFH
EndFor
LV_ENA_SEG_T_MES = 0
```

Recurrence: Segment

Activation: LV_SYN_ENG = 1

Deactivation: LV_SYN_ENG = 0

Formula section:

```
SEG_T_MES = T_SEG_ER                    // physical equivalence conversion
If    LV_SYN_ENG = 0
      Or  SEG_T_MES > C_SEG_T_MES_MAX
      Or  SEG_T_MES < C_SEG_T_MES_MIN
      Or  LV_INH_MIS_CRK = 1
      Or  LV_ERR_CRK = 1
      Or  LV_LIH_ERR_CRK = 1
Then      LV_ENA_SEG_T_MES = 0
          SEG_T_MES = FFFFH
Else      LV_ENA_SEG_T_MES = 1
EndIf
SEG_T_MES_CYL[SEG_NR_MES] = SEG_T_MES
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_SEG_T_MES_MAX	1	0...FFFFH	0...65535	1	[µs]
ER maximum valid segment time					
C_SEG_T_MES_MIN	1	0...FFFFH	0...65535	1	[µs]
ER minimum valid segment time					

21.3.2.1 Correction of the engine roughness segment adaptive values

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SEG_AD_COR_ER	V	8000H...7FFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
ER segment adaptive values correction versus engine speed (for each segment)					

Input data:

SEG_NR_MES	N_32	NC_CYL_NR	LV_ENA_SEG_T_MES
LV_AT			

FUNCTION DESCRIPTION:

General information:

Due to crankshaft flexion appearance at high engine speed that decreases the misfire detection efficiency, an open-loop adaptive correction value depending on engine speed for each cylinder can be introduced per type of transmission.

A correction can be applied only for each singular segment of the flywheel.

Application conditions:

Initialisation: *at ECU reset, Engine Stop* ***Or*** *Deactivation event*

SEG_AD_COR_ER = 0

Recurrence: Segment

Activation: LV_ENA_SEG_T_MES = 1

Deactivation: LV_ENA_SEG_T_MES = 0

Formula section:

SEG_AD_COR_ER is updated according the following table *(function of NC_CYL_NR and the current value of SEG_NR_MES)*

SEG_NR_MES	NC_CYL_NR = 3	NC_CYL_NR = 4	NC_CYL_NR = 5	NC_CYL_NR = 6	NC_CYL_NR = 8
0	SEG_AD_COR_ER = 0				
1	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_1(N_32)				
2	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_2(N_32)	SEG_AD_COR_ER = 0	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_2(N_32)	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_2(N_32)	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_2(N_32)
3		SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_1(N_32)	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_3(N_32)	SEG_AD_COR_ER = 0	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_3(N_32)
4			SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_4(N_32)	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_1(N_32)	SEG_AD_COR_ER = 0
5				SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_2(N_32)	SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_1(N_32)
6					SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_2(N_32)
7					SEG_AD_COR_ER = IP_SEG_AD_COR_ER_xT_3(N_32)

_xT stands for _AT if LV_AT = 1 else stands for _MT

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
IP_SEG_AD_COR_ER_MT_1	6	0...FFFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
LDPM_N_32_1_ENRD	6	0...FFH	0...8160	32	[rpm]
Adaptive value correction versus engine speed in MT for segment 1					
IP_SEG_AD_COR_ER_AT_1	6	0...FFFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
LDPM_N_32_6_ENRD	6	0...FFH	0...8160	32	[rpm]
Adaptive value correction versus engine speed in AT for segment 1					
#IF NC_CYL_NR=3, 5, 6 Or 8					
IP_SEG_AD_COR_ER_MT_2	6	0...FFFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
LDPM_N_32_1_ENRD	6	0...FFH	0...8160	32	[rpm]
Adaptive value correction versus engine speed in MT for segment 2					
IP_SEG_AD_COR_ER_AT_2	6	0...FFFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
LDPM_N_32_6_ENRD	6	0...FFH	0...8160	32	[rpm]
Adaptive value correction versus engine speed in AT for segment 2					
#ENDIF					
#IF NC_CYL_NR=5 Or 8					
IP_SEG_AD_COR_ER_MT_3	6	0...FFFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
LDPM_N_32_1_ENRD	6	0...FFH	0...8160	32	[rpm]
Adaptive value correction versus engine speed in MT for segment 3					
IP_SEG_AD_COR_ER_AT_3	6	0...FFFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
LDPM_N_32_6_ENRD	6	0...FFH	0...8160	32	[rpm]
Adaptive value correction versus engine speed in AT for segment 3					
#ENDIF					
#IF NC_CYL_NR=5					
IP_SEG_AD_COR_ER_MT_4	6	0...FFFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
LDPM_N_32_1_ENRD	6	0...FFH	0...8160	32	[rpm]
Adaptive value correction versus engine speed in MT for segment 4					
IP_SEG_AD_COR_ER_AT_4	6	0...FFFFH	-7.8125...7.81226	0.2384e-3	[°/oo]
LDPM_N_32_6_ENRD	6	0...FFH	0...8160	32	[rpm]
Adaptive value correction versus engine speed in AT for segment 4					
#ENDIF					

21.3.3 Engine roughness segment time correction

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SEG_T_COR_BUF[NC_SIZE_SEG_T_COR_BUF]	O	0...FFFFH	0...65535	1	[µs]
ER corrected segment times buffer (after adaptation)					
SEG_T_COR	V/O	0...FFFFH	0...65535	1	[µs]
Current ER corrected segment time					

Input data:

SEG_T_MES	SEG_AD_COR_ER	SEG_NR_MES	SEG_AD_MMV_ER[NC_CYL_NR]
LV_ENA_SEG_T_MES	LV_SYN_ENG		

FUNCTION DESCRIPTION:

General information:

The prerequisite for a reliable engine roughness evaluation is an accurate ER segment time measurement.

As the crankshaft flywheel teeth are subject to manufacturing tolerances, geometrical run-out and off-center installation which result in angles modulation, consequently the measured raw segment periods aren't similar and have systematic inaccuracies regarding other segments.

As these inaccuracies are systematic, they can be "learned" by an adaptive process during fuel cut-off periods and used for ER segment time correction. This adaptive process is described in the module : Engine roughness adaptive learning process.

ER segment adaptive values: SEG_AD_MMV_ER[NC_CYL_NR] *(inputs)*

ER adaptive values are stored according SEG_NR_MES counter. These adaptive values are obtained and updated by the ER segment adaptive values learning process. There's one ER segment adaptive value per engine cylinder.

Remark: The SEG_AD_MMV_ER_x are stored in ECU non-volatile memory.

ER current corrected segment time: SEG_T_COR *(output)*

This value is the current result of the ER segment correction process obtained with T_SEG_ER as input.

ER corrected segment times buffer: SEG_T_COR_BUF[NC_SIZE_SEG_T_COR_BUF] *(outputs)*

This is a buffer where are stored SEG_T_COR values (ER corrected segment time current value) who are needed for ER calculation.

SEG_T_COR is stored according a FIFO management (first in, first out).

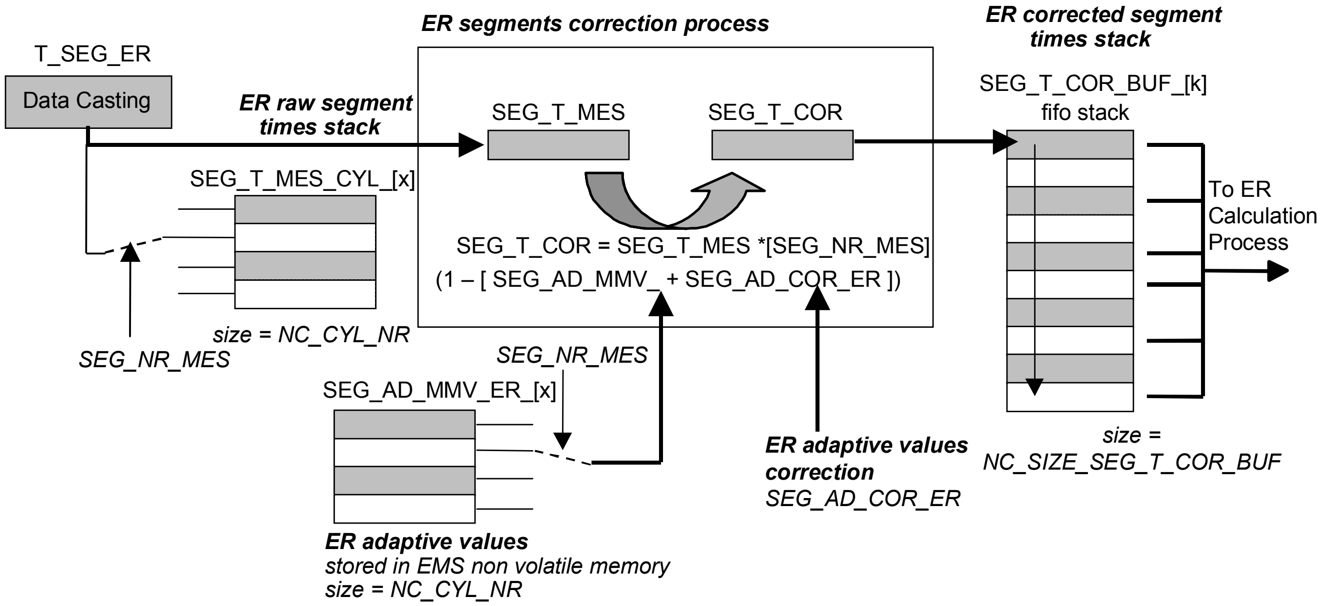
SEG_T_COR_BUF buffer size depends on engine type:

NC_SIZE_SEG_T_COR_BUF = *11 segment time samples for a 5 cylinder engine*

NC_SIZE_SEG_T_COR_BUF = *9 segment time samples for 4 & 8 cylinder engine*

NC_SIZE_SEG_T_COR_BUF = *7 segment time samples for 3 & 6 cylinder engine*

This size NC_SIZE_SEG_T_COR_BUF *is determined by the ER sample number required by ER calculation according engine type.*



Application conditions:

Initialisation: *at ECU reset, Engine Stop* ***Or*** *Deactivation event*

```
SEG_T_COR = FFFFH
For k = 0 to NC_SIZE_SEG_T_COR_BUF-1
    SEG_T_COR_BUF[k] = FFFFH
EndFor
```

Recurrence: Segment

Activation: LV_SYN_ENG = 1

Deactivation: LV_SYN_ENG = 0

Formula section:

```
If(1) LV_ENA_SEG_T_MES = 0
Then(1) SEG_T_COR = FFFFH
    For(2) k = 0 to NC_SIZE_SEG_T_COR_BUF-1
        SEG_T_COR_BUF[k] = FFFFH
    EndFor(2)
Else(1) SEG_T_COR =
        SEG_T_MES * (1 – [SEG_AD_MMV_ER[SEG_NR_MES] + SEG_AD_COR_ER] )
    For(2) k : 1 to NC_SIZE_SEG_T_COR_BUF-1
        SEG_T_COR_BUF[k] is managed as a FIFO stack (SEG_T_COR_BUF last value is lost)
```

```
        EndFor(2)
        SEG_T_COR_BUF[0] = SEG_T_COR
EndIf(1)
```

Where:

SEG_T_MES : *Current uncorrected ER segment time after data casting*

SEG_T_COR : *Current ER corrected segment time*

SEG_AD_MMV_ER[SEG_NR_MES] : *ER segment adaptive value corresponding to current T_SEG_ER phasing*

SEG_AD_COR_ER : *Adaptive value correction versus engine speed corresponding to current T_SEG_ER phasing*

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_SIZE_SEG_T_COR_BUF	-	0...FFH	0...255	1	[-]
Size of the array of engine roughness corrected segment times					

21.3.4 Engine roughness calculation

21.3.5 General information

The engine roughness index (ER) is a system variable relative to crankshaft segments behaviour. This index is mainly used by the misfire monitoring & algorithms dedicated to combustion process control.

Integration version for 3,4,5,6 & 8 cylinder engines.

The engine roughness system variable is calculated each ER window end occurrence, every time the engine is running & synchronised.

21.3.6 Engine roughness segment reference

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SEG_NR_ER	V/O	0...7H	0...7	1	[-]
ER segment counter for segment identification (0 ... NC_CYL_NR - 1)					

Input data:

SEG_NR	NC_CYL_NR	LV_SYN_ENG	

Due to engine roughness definition, there is a delay of C_SEG_DLY_ER segments between ER actual value and the segment that was observed by this ER value.

SEG_NR_ER will be used to identify which cylinder is observed with the engine roughness system variable, it's particularly useful for misfire cylinder identification.

The segment counter SEG_NR_ER with the limit 0...(NC_CYL_NR - 1) is incremented in the same way as the common segment counter SEG_NR.

Application conditions:

Initialisation: *at ECU reset, Engine Stop* ***Or*** *Deactivation condition event*

SEG_NR_ER = 0

Recurrence: Segment

Activation: LV_SYN_ENG = 1

Deactivation: LV_SYN_ENG = 0

Formula section:

SEG_NR_ER = (SEG_NR + 2 * NC_CYL_NR – C_SEG_DLY_ER) % NC_CYL_NR

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_SEG_DLY_ER	1	0...CH	0...12	1	[-]
Delay of ER segment reference					

21.3.7 Engine roughness segments control process

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ENA_ER	V/O	0...1H	0...1	1	[-]
Engine roughness valid calculation control flag					

Input data:

SEG_T_COR_BUF[NC_SIZE_SEG_T_COR_BUF]	LV_ENA_SEG_T_MES	N	

Description:

All ER components calculations are based on corrected segment time acquisitions buffer values SEG_T_COR_BUF.

As soon as an invalid segment has been detected during acquisition process (LV_ENA_SEG_T_MES = 0) or segment adaptive calculation, ER calculation & misfire detection have to be disabled (LV_ENA_ER = 0).

Application conditions:

Initialisation: *at ECU reset, Engine Stop* ***Or*** *Deactivation condition event*

LV_ENA_ER = 0

Recurrence: Segment

Activation:

Engine roughness calculation is executed only if:

NC_SIZE_SEG_T_COR_BUF consecutive valid ER segment times have been acquired & stored in SEG_T_COR_BUF buffer.

"**Valid**" stands for:

LV_ENA_SEG_T_MES = 1 on NC_SIZE_SEG_T_COR_BUF consecutive ER segments

And no ER components calculation overflow on current segment *(DRV0, DRV1, DRV2 and ER components)*

And N <= C_N_MAX_ER

And N >= C_N_MIN_ER

In this case:

ER components are calculated as described below.

LV_ENA_ER = 1 *(if no overflow occurs during ER components calculation)*

Deactivation:

Engine roughness calculation is stopped & initialised when at least one of the NC_SIZE_SEG_T_COR_BUF consecutive segment times have been acquired & stored in SEG_T_COR_BUF buffer have been detected unvalid.

"**Unvalid**" stands for:

LV_ENA_SEG_T_MES = 0 at least one time on NC_SIZE_SEG_T_COR_BUF consecutive ER segments

Or at least one calculation overflow occurs one of ER components *(DRV0, DRV1, DRV2 or ER component)* on current segment.

Or N > C_N_MAX_ER

Or N < C_N_MIN_ER

In that case :

LV_ENA_ER = 0

ER components are initialised to 0

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MIN_ER	1	0...1FE0H	0...8160	1	[rpm]
minimum engine speed for engine roughness calculation					
C_N_MAX_ER	1	0...1FE0H	0...8160	1	[rpm]
maximum engine speed for engine roughness calculation					

21.3.8 Cylinder specific engine roughness components correction

#IF NLC_CONF_GAIN_ADD_ER = 1

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_GAIN_ER[NC_CYL_NR]	V	0...1FFH	0...1.99609	3.9063e-3	[-]
ER multiplicative correction components					
FAC_ADD_ER[NC_CYL_NR]	V	8000...7FFFH	-32768...32767	1	[µs]
ER additive correction components					

Input data:

LV_AT	LOAD_MIS	N_32	NC_CYL_NR
LV_ENA_ER	LV_VAR_TCT		

FUNCTION DESCRIPTION:

General information:

The misfire detection based on ER index can be complicated by the torsional/flexion vibrations especially with multiple cylinder engines with a long crankshaft. This affects on misfire detection efficiency, especially at high engine speed.

To balance this phenomenon, after segment timing correction with ER adaptive values *(see ER segment correction process)*, an irregular operation is determined from the measured cylinder segment times, cylinder specific additive and multiplicative corrections are applied on the raw engine roughness value *(ER_RAW)*.

FAC_ADD_ER[NC_CYL_NR] and FAC_GAIN_ER[NC_CYL_NR] correction coefficients are based on calibration tables specific to each cylinder and to transmission type (MT/AT).

Application conditions:

Initialisation: *at ECU reset, at Engine Stop* ***Or*** *on Deactivation condition event*

```
For x = 0 to NC_CYL_NR-1
    FAC_GAIN_ER[x] = 1
    FAC_ADD_ER[x] = 0
EndFor
```

Recurrence: *Segment*

Activation: LV_ENA_ER = 1

Deactivation: LV_ENA_ER = 0

Formula section:

```
If(1) LV_AT = 1
Then(1)
        XX= AT
ElseIf(2)   LV_VAR_TCT = 1
    Then(2)
        XX= TCT
    Else(2)
        XX= MT
    Endif(2)
Endif(1)

FAC_GAIN_ER[SEG_NR_ER] = IP_FAC_GAIN_ER_xx[SEG_NR_ER](N_32)
FAC_ADD_ER[SEG_NR_ER] =  IP_FAC_ADD_ER_xx[SEG_NR_ER](N_32,LOAD_MIS)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_GAIN_ER_MT[NC_CYL_NR]	6	0...1FFH	0...1.99609	3.9063e-3	[-]
LDPM_N_32_2_ENRD[NC_CYL_NR]	6	0...FFH	0...8160	32	[rpm]
ER cylinder x multiplicative segment-dependent factor for torsion correction with MT					
IP_FAC_ADD_ER_MT[NC_CYL_NR]	6*6	0...FFFFH	-32768...32767	1	[µs]
LDPM_N_32_3_ENRD[NC_CYL_NR]	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_IP_FAC_ADD_ER_MT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER cylinder x additive segment-dependent factor for torsion correction with MT					
IP_FAC_GAIN_ER_AT[NC_CYL_NR]	6	0...1FFH	0...1.99609	3.9063e-3	[-]
LDPM_N_32_4_ENRD[NC_CYL_NR]	6	0...FFH	0...8160	32	[rpm]
ER cylinder x multiplicative segment-dependent factor for torsion correction with AT					
IP_FAC_ADD_ER_AT[NC_CYL_NR]	6*6	0...FFFFH	-32768...32767	1	[µs]
LDPM_N_32_5_ENRD[NC_CYL_NR]	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_IP_FAC_ADD_ER_AT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER cylinder x additive segment-dependent factor for torsion correction with AT					
IP_FAC_GAIN_ER_TCT[NC_CYL_NR]	6	0...1FFH	0...1.99609	3.9063e-3	[-]
LDP_N_32_7_ENRD[NC_CYL_NR]	6	0...FFH	0...8160	32	[rpm]
ER cylinder x multiplicative segment-dependent factor for torsion correction with TCT					
IP_FAC_ADD_ER_TCT[NC_CYL_NR]	6*6	0...FFFFH	-32768...32767	1	[µs]
LDP_N_32_8_ENRD[NC_CYL_NR]	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_IP_FAC_ADD_ER_TCT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER cylinder x additive segment-dependent factor for torsion correction with TCT					

#ENDIF

Configuration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
NLC_CONF_GAIN_ADD_ER	-	0...1H	0...1	1	[-]
ER multiplicative and additive correction enable					

21.3.9 Engine roughness DRV2 components filter & gain determination

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CRLC_DRV2_ER	V	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation factor for engine roughness curvature component calculation					
FAC_DRV2_ER	V	0...3FCH	0...4	3.9216e-3	[-]
Weighting factor for engine roughness curvative component calculation (correction amount)					

Input data:

N_32	LV_AT	LOAD_MIS	CTR_T_ZDLY_MIS
TCO	LV_IS	LV_VAR_TCT	

FUNCTION DESCRIPTION:

This function allows to determinate the coefficients relative to DRV2 components of the engine roughness, according different configurations:

If used *(see project/customer legal requirements concerning misfire monitoring after engine start)*, during zero delay misfire monitoring activation period, specific gain & filter coefficient can be apply to DRV2 component to be able to detect misfire just after engine start *(dynamic correction modified).*

Else, FAC_DRV2_ER & CRLC_DRV2_ER are defined according transmission type *(AT/MT)*.

Application conditions:

Initialisation: *at ECU reset, at Engine Stop* ***Or*** *on Deactivation condition event*

FAC_DRV2_ER = IP_FAC_DRV2_ZDLY_ER(TCO)

CRLC_DRV2_ER = C_CRLC_DRV2_ZDLY_ER

Recurrence: Segment

Activation: LV_ENA_ER = 1

Deactivation: LV_ENA_ER = 0

Formula section:

```
If(1) LV_AT = 1
Then(1)
        XX= AT
ElseIf(2)  LV_VAR_TCT = 1
    Then(2)
        XX= TCT
    Else(2)
        XX= MT
    Endif(2)
Endif(1)
```

```
If(1) CTR_T_ZDLY_MIS > C_T_DRV2_ZDLY_ER
Then(1)    // Zero delay misfire monitoring activation period
    FAC_DRV2_ER = IP_FAC_DRV2_ZDLY_ER(TCO)
    CRLC_DRV2_ER = C_CRLC_DRV2_ZDLY_ER
ElseIf(2)
        If(3)    LV_IS = 1   And    N_32 < C_N_32_MAX_DRV2_ER_IS
        Then(3)        FAC_DRV2_ER = IP_FAC_DRV2_ER_IS_xx(N, LOAD_MIS)
        Else(3)        FAC_DRV2_ER = IP_FAC_DRV2_ER_ xx (N_32, LOAD_MIS)
        EndIf(3)
        CRLC_DRV2_ER = IP_CRLC_DRV2_ER_ xx (N_32)
  EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_DRV2_ZDLY_ER	1	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation factor for engine roughness curvature component calculation after start					
C_N_32_MAX_DRV2_ER_IS	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed to apply DRV2 specific weighting factor in idle speed					
C_T_DRV2_ZDLY_ER	1	0...FFFFH	0...655.35	0.01	[s]
Delay for specific curvative component calculation during after start					
IP_FAC_DRV2_ER_AT	6*6	0...3FCH	0...4	3.9216e-3	[-]
LDP_N_32_IP_FAC_N_DRV2_ER_AT	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_MIS_IP_FAC_DRV2_ER_AT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for DRV2 component calculation - AT vehicle					
IP_FAC_DRV2_ER_IS_AT	4*4	0...3FCH	0...4	3.9216e-3	[-]
LDP_N_IP_FAC_N_DRV2_ER_IS_AT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_FAC_DRV2_ER_IS_AT	4	0...7FFFH	0...99.99694	3.05E-03	[%]
Weighting factor for DRV2 component calculation - AT vehicle in idle speed					
IP_CRLC_DRV2_ER_AT	9	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_N_32_IP_CRLC_DRV2_ER_AT	9	0...FFH	0...8160	32	[rpm]
Correlation factor for engine roughness curvature component calculation (correction duration) - AT vehicle					
IP_FAC_DRV2_ER_MT	6*6	0...3FCH	0...4	3.9216e-3	[-]
LDP_N_32_IP_FAC_N_DRV2_ER_MT	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_MIS_IP_FAC_DRV2_ER_MT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for DRV2 component calculation - MT vehicle					
IP_FAC_DRV2_ER_IS_MT	4*4	0...3FCH	0...4	3.9216e-3	[-]
LDP_N_IP_FAC_N_DRV2_ER_IS_MT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_FAC_DRV2_ER_IS_MT	4	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for DRV2 component calculation - MT vehicle in idle speed					
IP_CRLC_DRV2_ER_MT	9	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_N_32_IP_CRLC_DRV2_ER_MT	9	0...FFH	0...8160	32	[rpm]
Correlation factor for engine roughness curvature component calculation (correction duration) - MT vehicle					
IP_FAC_DRV2_ZDLY_ER	6	0...3FCH	0...4	3.9216e-3	[-]
LDP_TCO_IP_FAC_DRV2_ZDLY_ER	6	0...FEH	-48...142.5	0.75	[°C]
Weighting factor for engine roughness curvative component calculation (correction amount) after start					
IP_FAC_DRV2_ER_IS_TCT	4*4	0...3FCH	0...4	3.9216e-3	[-]
LDP_N_IP_FAC_N_DRV2_ER_IS_TCT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_FAC_DRV2_ER_IS_TCT	4	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for DRV2 component calculation - TCT vehicle in idle speed					
IP_FAC_DRV2_ER_TCT	6*6	0...3FCH	0...4	3.9216e-3	[-]
LDP_N_32_IP_FAC_N_DRV2_ER_TCT	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_MIS_IP_FAC_DRV2_ER_TCT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for DRV2 component calculation - TCT vehicle					
IP_CRLC_DRV2_ER_TCT	9	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_N_32_IP_CRLC_DRV2_ER_TCT	9	0...FFH	0...8160	32	[rpm]
Correlation factor for engine roughness curvature component calculation (correction duration) - TCT vehicle					

21.3.10 Engine roughness components calculation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
DRV0_ER	V/O	80000000... 7FFFFFFFH	-2147483648... 2147483647	1	[µs]
Engine roughness static component					
DRV1_ER	V/O	8000...7FFFH	-32768...32767	1	[µs]
Engine roughness dynamic component					
DRV2_ER	V	0...FFFFH	0...65535	1	[µs]
Engine roughness curvature component					
DRV2_MMV_ER	V	0...FFFFH	0...65535	1	[µs]
Averaged engine roughness curvature component					
ER_RAW	V/O	8000...7FFFH	-32768...32767	1	[µs]

Engine roughness without curvature component

Input data:

SEG_T_COR	SEG_T_COR_BUF[NC_SIZE_SEG_T_COR_BUF]	NC_CYL_NR	

General information:

The engine roughness calculation needs three individual components: a static component (DRV0_ER), a dynamic component (DRV1_ER) and a curvature component (DRV2_ER). These component definitions are specific to the engine cylinder number (see following related subchapters).

DRV2_MMV_ER is the filtered curvature component correction (DRV2_ER) who allows adding an artificial positive offset on ER value in case of dynamic curvature effects. This feature had a safety effect against wrong detection in case of segment oscillations.

ER_RAW is the engine roughness value without curvature component (DRV2_MMV_ER), used by cylinder balancing and engine warm up monitoring modules.

Application conditions:

Initialisation: *at ECU reset, at Engine Stop* ***Or*** *on Deactivation condition event*

DRV0_ER = 0

DRV1_ER = 0

DRV2_ER = 0

DRV2_MMV_ER = 0

ER_RAW = 0

Recurrence: Segment

Activation: LV_ENA_ER = 1

Deactivation: LV_ENA_ER = 0

Formula section:

Engine roughness static component:

DRV0_ER(n) = NC_CYL_NR * [SEG_T_COR(n) - SEG_T_COR(n+1)]

Engine roughness dynamic and curvature components:

DRV1_ER and DRV2_ER components are engine type specific.

3 Cylinder Engine

```
#If   NC_CYL_NR = 3
      DRV1_ER(n) = SEG_T_COR(n-2) - SEG_T_COR(n+1)
      DRV2_ER(n) =  FAC_DRV2_ER *
             max(0,[SEG_T_COR(n-3) - SEG_T_COR(n)] - [SEG_T_COR(n) - SEG_T_COR(n+3)])
#EndIf
```

4 or 8 Cylinder Engine

```
#If   NC_CYL_NR = 4 or 8
      DRV1_ER (n) = SEG_T_COR(n-2) - SEG_T_COR(n+2)
      DRV2_ER (n) =  FAC_DRV2_ER *
              max(0,[SEG_T_COR(n-4)- SEG_T_COR(n)] - [SEG_T_COR(n) -SEG_T_COR(n+4)])
#EndIf
```

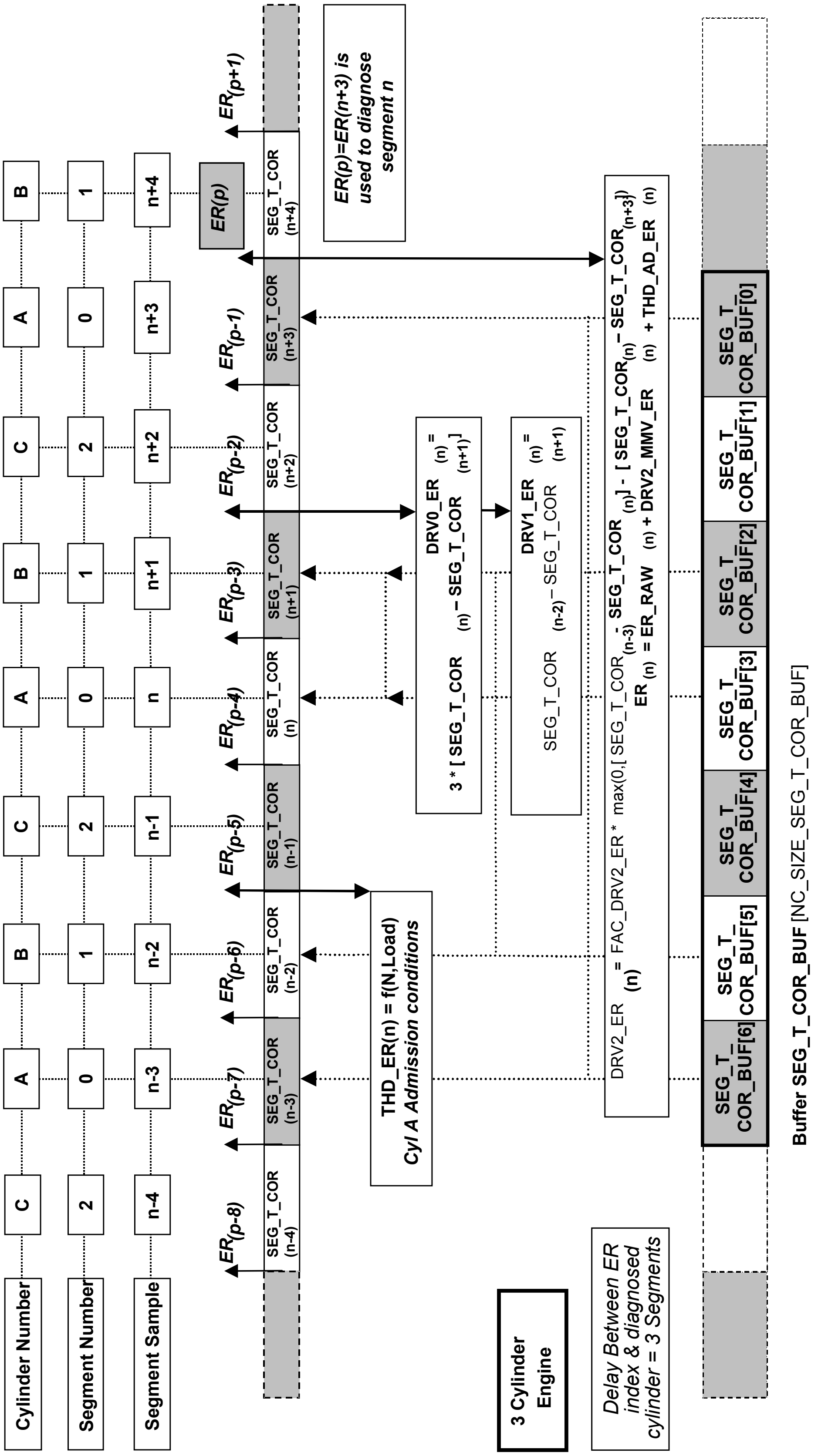
5 Cylinder Engine

```
#If   NC_CYL_NR = 5
      DRV1_ER (n) = SEG_T_COR(n-2) - SEG_T_COR(n+3)
      DRV2_ER (n) =  FAC_DRV2_ER *
               max(0,[SEG_T_COR(n-5)- SEG_T_COR(n)] - [SEG_T_COR(n)-SEG_T_COR(n+5)])
#EndIf
```

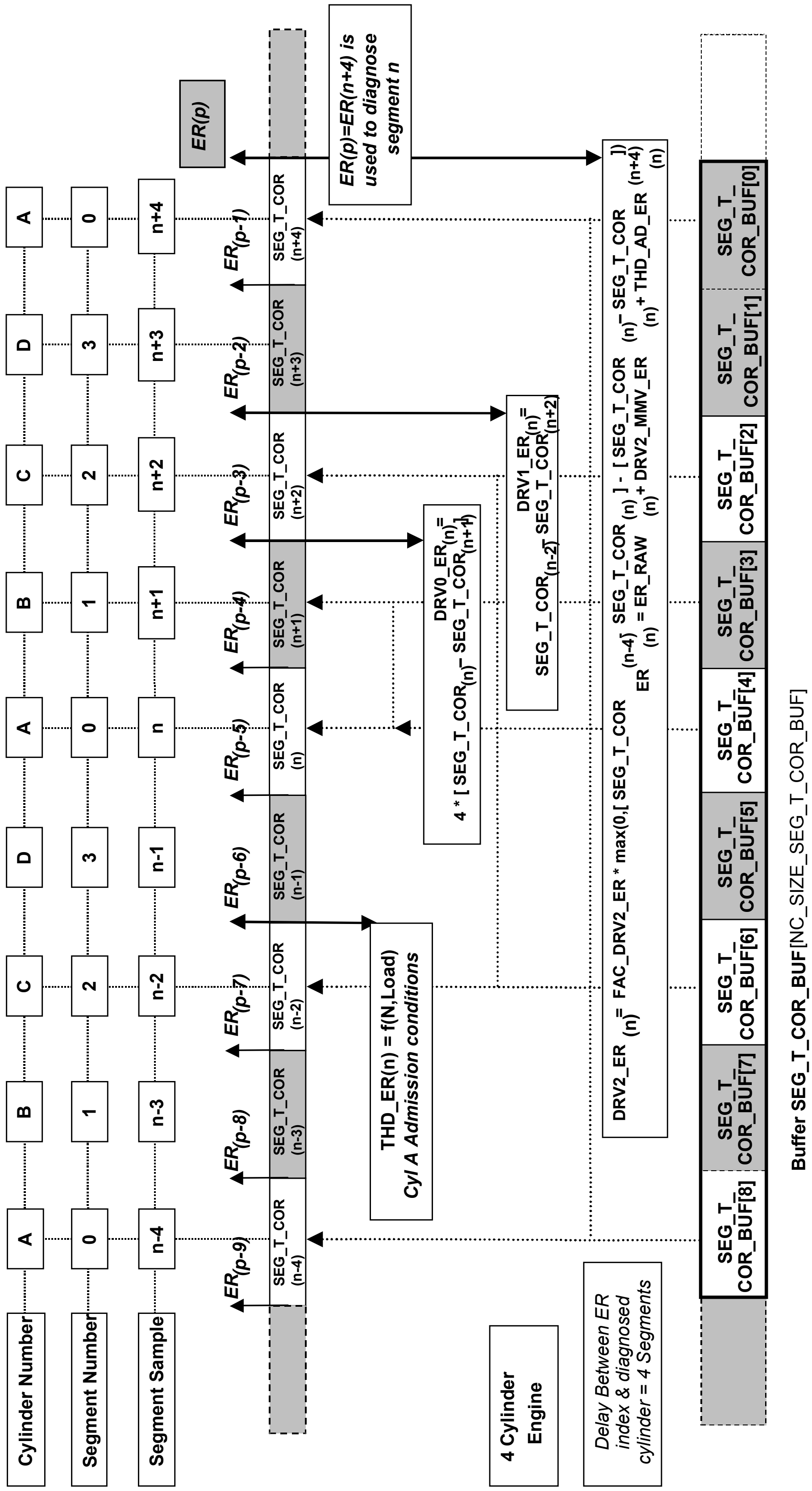
6 Cylinder Engine

```
#If   NC_CYL_NR = 6
      DRV1_ER (n) = SEG_T_COR(n-3) - SEG_T_COR(n+3)
      DRV2_ER (n) =  FAC_DRV2_ER *
             max(0,[SEG_T_COR(n-3)- SEG_T_COR(n)] - [SEG_T_COR(n)-SEG_T_COR(n+3)])
#EndIf
```

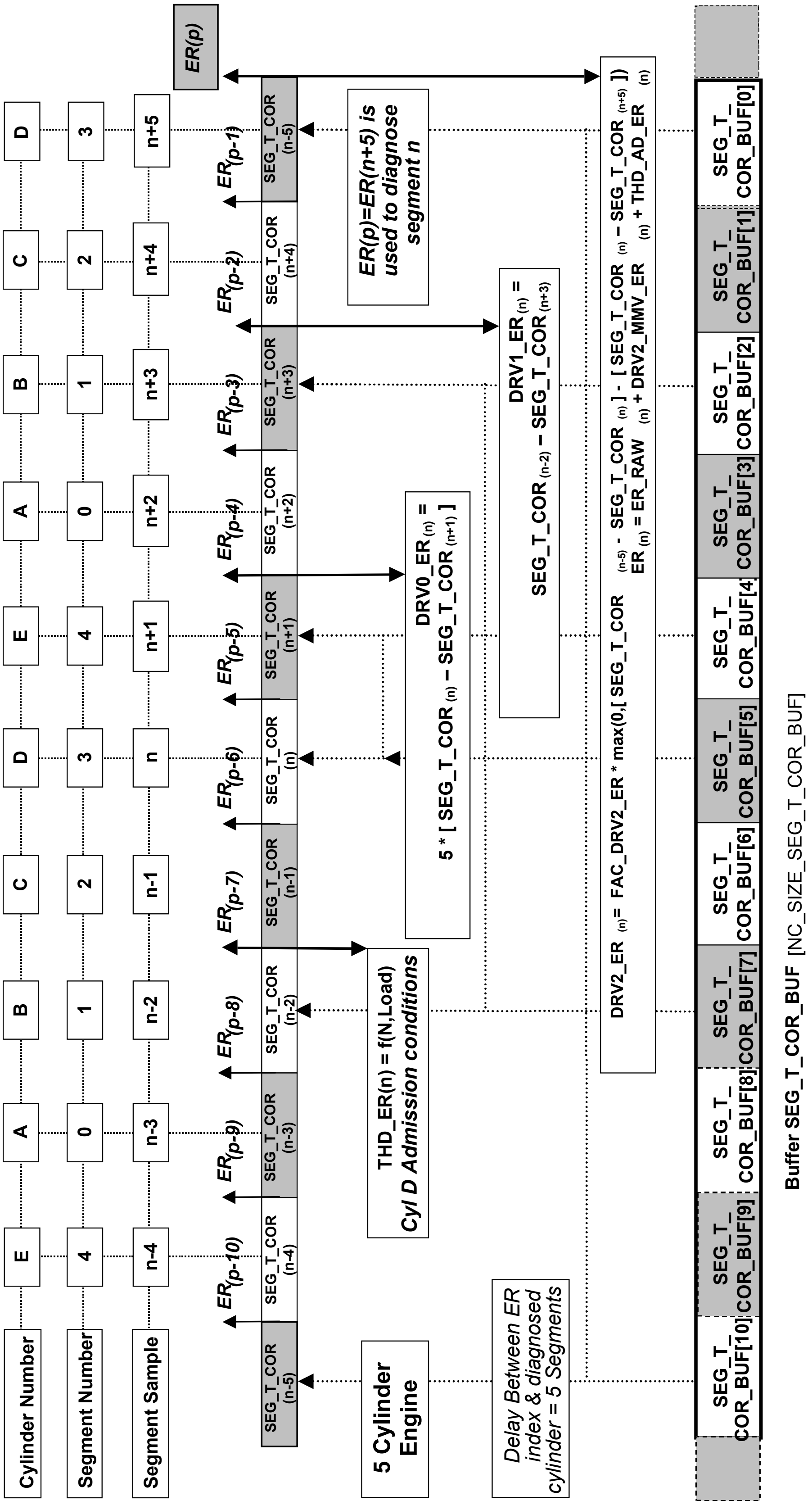
Implant scheme for 3 cylinder engine:



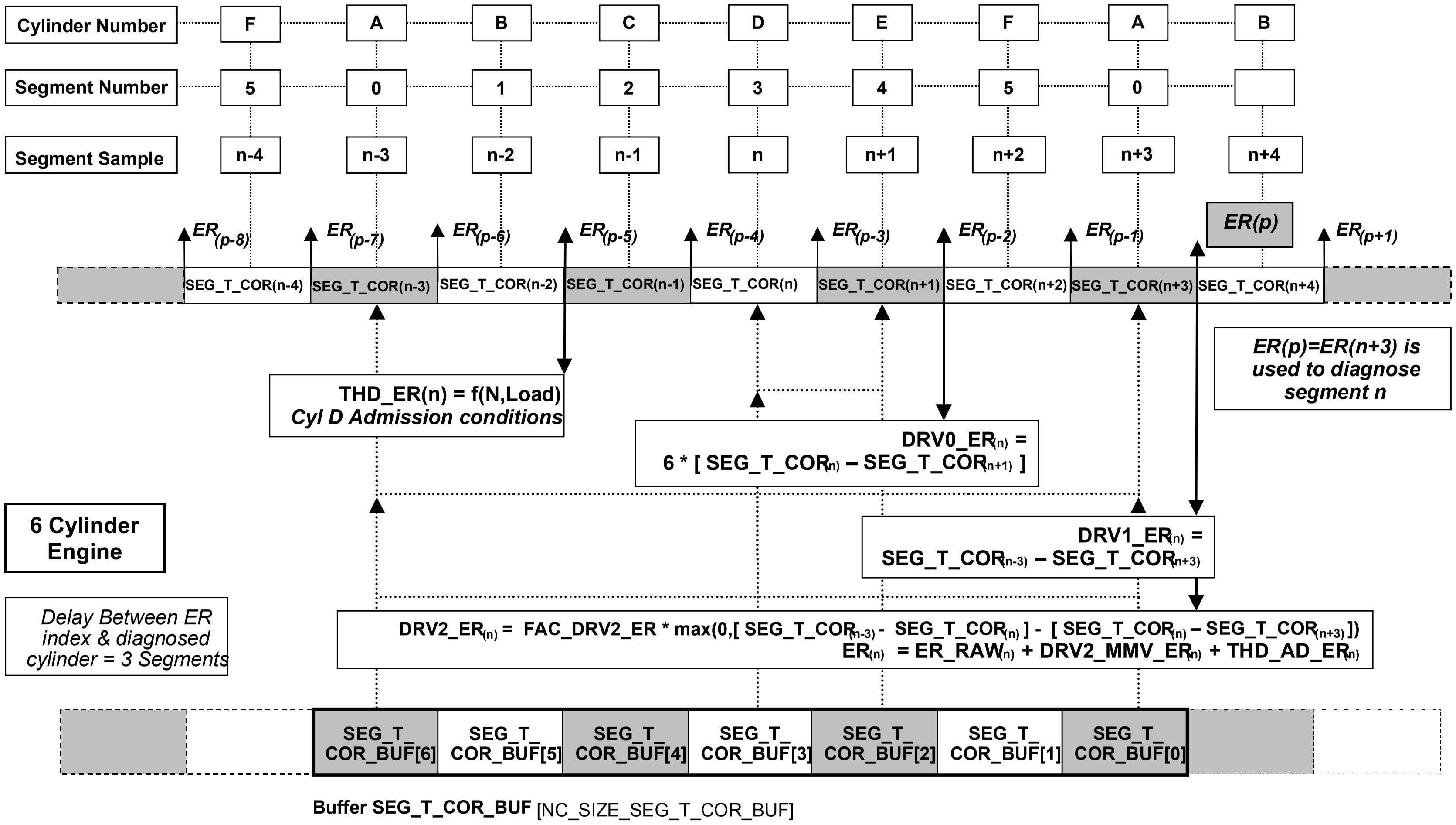
Implant scheme for 4 cylinder engine:



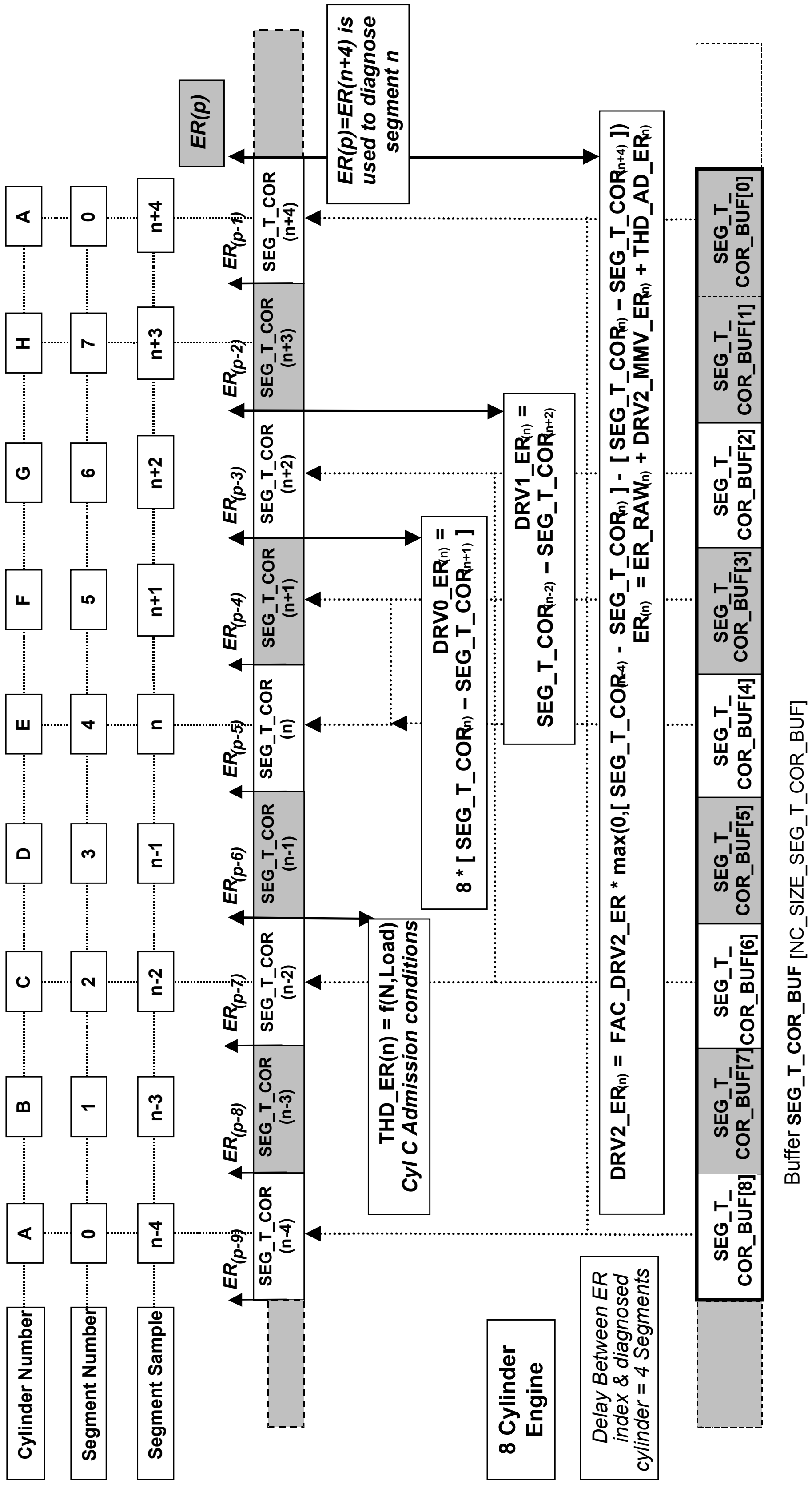
Implant scheme for 5 cylinder engine:



Implant scheme for 6 cylinder engine:



Implant scheme for 8 cylinder engine:



Engine roughness filtered curvature component:

$$DRV2\_MMV\_ER_{(n)} = DRV2\_MMV\_ER_{(n-1)} + CRLC\_DRV2\_ER * (DRV2\_ER_{(n)} - DRV2\_MMV\_ER_{(n-1)})$$

Engine roughness value without curvature component:

```
#If  NLC_CONF_GAIN_ADD_ER = 1
    ER_RAW = FAC_GAIN_ER[SEG_NR_ER] *
             (DRV0_ER - DRV1_ER - FAC_ADD_ER[SEG_NR_ER])
#Else
    ER_RAW = DRV0_ER - DRV1_ER
#EndIf
```

21.3.11 Engine roughness values determination

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ER	V/O	8000...7FFFH	-32768...32767	1	[µs]
Engine roughness					
ER_CYL[NC_CYL_NR]	V	8000...7FFFH	-32768...32767	1	[µs]
Cylinder normalised engine roughness values					
ER_STND	V/O	8000...7FFFH	-32768...32767	1	[-]
Normalised engine roughness					
ER_STND_CYL[NC_CYL_NR]	V	8000...7FFFH	-32768...32767	1	[-]
Cylinder normalised engine roughness values					

Input data:

THD_AD_ER	SEG_T_COR_BUF[NC_SIZE_SEG_T_COR_BUF]	N	C_SEG_DLY_ER

Description:

ER is the complete engine roughness value including all ER components (DRV0_ER, DRV1_ER & DRV2_MMV_ER), specified & used by misfire detection module.

ER_STND is the normalised engine roughness value versus engine speed that could be amplified by a gain C_FAC_SCA_ER_STND. This is the nominal engine roughness value divided by the corresponding cubic corrected ER segment time (SEG_T_COR_BUF[C_SEG_DLY_ER])^3 . The cubic segment time is calculated by using a mantisse and an exponent.

ER & ER_STND calculation formula are generic to all engine type.

A positive offset (THD_AD_ER) is added to ER components as long as the ER adaptive learning process hasn't been executed. This, in a way to avoid wrong misfire detection in engine operation areas where crankshaft decelerations due to a misfire are very close to ER noise due to the flywheel mechanical tolerances.

This offset decreases as the ER adaptive learning progress. At the end of the learning process, this offset is null *(see next chapter ER adaptive values learning process for more informations)*.

Once the ER fast adaptive learning stage has been executed, these flywheel mechanical tolerances are sufficiently reduced to able to detect misfire in full legal range without adding a positive offset.

ER_CYL_x and ER_STND_CYL_x are the engine roughness values obtained for corresponding cylinder x, these values are switched according SEG_NR_ER segment reference.

Application conditions:

Initialisation: *at ECU reset, at Engine Stop* ***Or*** *on Deactivation condition event*

```
  ER = 0
  ER_STND = 0
  For x = 0 to NC_CYL_NR-1
     ER_CYL[x] = 0
     ER_STND_CYL[x] = 0
  EndFor
```

Recurrence: Segment

Activation: LV_ENA_ER = 1

Deactivation: LV_ENA_ER = 0

Formula section:

Engine roughness:

```
  ER = ER_RAW + DRV2_MMV_ER + THD_AD_ER
  ER_CYL[SEG_NR_ER] = ER
```

Normalised Engine roughness versus engine speed:

```
If      N < C_N_MAX_ER_STND
Then
      $ER\_STND = ER / (SEG\_T\_COR\_BUF_{[C\_SEG\_DLY\_ER]})^3 * 2^{(16+C\_FAC\_SCA\_ER\_STND)}$
      ER_STND_CYL[SEG_NR_ER] = ER_STND
Else
      ER_STND = 0
      ER_STND_CYL[SEG_NR_ER] = 0
EndIf
```

Note: if C_SEG_DLY_ER ≥ NC_SIZE_SEG_T_COR_BUF, SEG_T_COR_BUF[C_SEG_DLY_ER] data is limited to the last cell of the buffer.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_SCA_ER_STND	1	0...FFH	0...255	1	[-]
Scaling factor used for normalised engine roughness calculation					
C_N_MAX_ER_STND	1	0...1FE0H	0...8160	1	[rpm]
Maximum engine speed to compute normalised engine roughness					

21.4 Engine roughness calculation (Appl. inc.)

21.4.1 Inhibition of engine roughness adaptive process - Application specific

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_APP_ER_AD	V/O	0...1H	0...1	1	[-]
ER segment adaptive process inhibition flag relative application specific conditions					
LV_REQ_SEG_AD_MMV_ER_APP	V/O	0...1H	0...1	1	[-]
Flag indicating request to reload stored segment adaptive values					
SEG_AD_MMV_ER_APP[NC_CYL_NR]	V/O	8000H...7FFFH	-7.8125...7.8123	0.000238	‰
ER filtered adaptive values used for reload (for each segment) if segment adaptive process not valid					

Input data:

LV_ENA_SEG_T_MES	LV_AT	GR_MT	GR_AT

FUNCTION DESCRIPTION:

Engine roughness segments adaptive process is inhibited when one of the following system conditions occurs. See application specific accessories that could have an impact on the transmission behaviour (CVT, Automatic gear shifting, specific gear ratios...).

Application conditions:

Initialisation: at ECU reset and engine Stop:

LV_INH_APP_ER_AD = 0

at ECU reset:

LV_REQ_SEG_AD_MMV_ER_APP = 0

SEG_AD_MMV_ER_APP[NC_CYL_NR] = 0

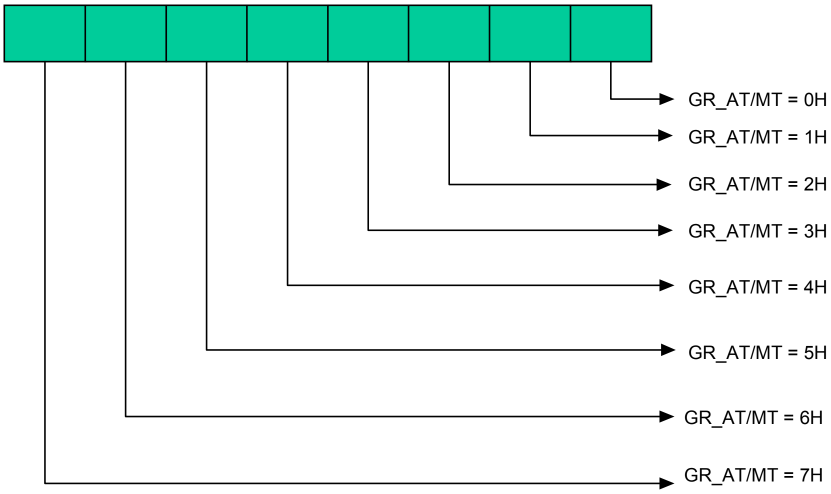
Update rate: Engine roughness segment

Activation: LV_ENA_SEG_T_MES = 1

Deactivation: LV_ENA_SEG_T_MES = 0

Diagram:

Bit-position within C_GR_INH_ER_AD_MT/AT



Example: Inhibition should be in 1st gear and 2st gear: => C_GR_INH_ER_AD_MT/AT = 6H

Formula section:

```
If      (LV_AT = 1 And C_GR_INH_ER_AD_AT[GR_AT] = 1)
   Or   ( LV_AT = 0 And C_GR_INH_ER_AD_MT[GR_MT] = 1)
Then    LV_INH_APP_ER_AD = 1
Else    LV_INH_APP_ER_AD = 0
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_GR_INH_ER_AD_AT	1	0...FFH	0...255	1	[-]
Bitfield used to inhibit engine roughness adaptive according current AT gear ratio (Gear corresponding bit = 1, ER_AD inhibited)					
C_GR_INH_ER_AD_MT	1	0...FFH	0...255	1	[-]
Bitfield used to inhabit engine roughness adaptive according current MT gear ratio (Gear corresponding bit = 1, ER_AD inhibited)					

21.4.2 ER segment adaptive process inhibition related to OBDI diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_OBD_ER_AD	V/O	0...1H	0...1	1	[-]
ER segment adaptive process inhibition flag.					

Input data:

LV_ERR_CRK	LV_ERR_CPS	LV_ERR_DIAGCPS	STATE_ERR_IV
LV_SYN_ENG			

General information:

ER segment adaptive process is inhibited when one of the mentioned OBD I errors occurs.

Application conditions:

Initialization: 0 at reset

Update rate: Engine roughness segment

Activation: LV_SYN_ENG = 1

Deactivation: LV_SYN_ENG = 0

Formula section:

```
If      STATE_ERR_IV <> 0        or
        LV_ERR_CPS = 1           or
        LV_ERR_DIAGCPS = 1       or
        LV_ERR_CRK = 1

Then    LV_INH_OBD_ER_AD = 1
Else    LV_INH_OBD_ER_AD = 0
EndIf
```

21.4.3 End of line & After sale service request to reset ER segment adaptive values

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SEG_AD_RST_ER_EOL	V/O	0...1H	0...1	1	[-]
EOL specific request to reset ER segment adaptive values					

Input data:

LV_EOL_OBD	LV_ENA_SEG_T_MES	LV_REQ_SEG_AD_RST_EOL	

General information:

During end of line plant and/or for After sale services, a request is generated via communication protocol, to reset ER segment adaptive values & learning process *(see project communication protocol).*

Application conditions:

Initialization: 0 at reset

Update rate: Engine roughness segment

Activation: LV_ENA_SEG_T_MES = 1

Deactivation: LV_ENA_SEG_T_MES = 0

Formula section:

```
If    LV_EOL_OBD = 1                        // OBD end of line process request
      And   LC_SEG_AD_RST_ENA_EOL = 1
      And   LV_REQ_SEG_AD_RST_EOL = 1
            // EOL request to reset segment adaptive values
Then        LV_SEG_AD_RST_ER_EOL = 1
Else        LV_SEG_AD_RST_ER_EOL = 0
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_SEG_AD_RST_ENA_EOL	1	0...1H	0...1	1	[-]
Specific EOL request enable flag to reset ER segment adaptive values & learning process					

21.4.4 Normalised engine roughness quotient for calibration

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ER_AV_QUO	V	0...FFH	0...2.55	0.01	[-]
Quotient of ER/THD_ER (to simplify calibration and testing)					

Input data:

ER	THD_ER	LV_REQ_INH_MIS	LV_ENA_ER
LV_ES	LV_ST		

FUNCTION DESCRIPTION:

General information:

To simplify the calibration, testing and controlling of the misfire function the variable ER_AV_QUO is defined.

Application conditions:

Initialisation: ER_AV_QUO = 0 at reset

Recurrence: updated every segment

Activation: LV_ENA_ER = 1

Deactivation: LV_ENA_ER = 0

Formula section:

```
If      LV_ES           =  1   or
        LV_ST           =  1   or
        LV_REQ_INH_MIS  =  1   or
        ER > 0
Then    ER_AV_QUO       =  0
Else    ER_AV_QUO       =  ER / THD_ER
Endif
```

21.5 Engine roughness adaptive learning process

General information:

To reduce systematic noises generated by target wheel mechanical tolerances, an adaptive learning process is required to improve the signal noise ratio on engine roughness variable and to avoid wrong misfire detection at high engine speed / low load.

ER segment adaptive values learning process is based on ER segments timing observation according a geometrical reference (ER segment time 0, SEG_NR_MES = 0).

During an engine deceleration in fuel cut-off mode on all cylinders (LV_INJ_CUT, no engine combustion influences), due to crankshaft inertia, engine speed can be considered in a short horizon as constant, thus on one engine cycle (720°Crk) timing dissimilarities between segment reference and others will be observed. For longer horizon observation, a deceleration is applied according current segment position in the engine cycle.

When all adaptation conditions fulfilled (smooth deceleration in fuel cut-off conditions), an adaptive values learning process is performed every **720° crank angle**, starting with **segment 0** *(adaptive process reference).*

If there is no cylinder specific mechanical damage or crankshaft torsion/flexion, the adaptation values for cylinders with the same mechanical segment will be approximately identical:

3 cylinder engine	4 cylinder engine	5 cylinder engine	6 cylinder engine	8 cylinder engine
SEG_AD_ER_0 *(geometrically singular)*	SEG_AD_ER_0 ≅ SEG_AD_ER_2	SEG_AD_ER_0 *(geometrically singular)*	SEG_AD_ER_0 ≅ SEG_AD_ER_3	SEG_AD_ER_0 ≅ SEG_AD_ER_4
SEG_AD_ER_1 *(geometrically singular)*	SEG_AD_ER_1 ≅ SEG_AD_ER_3	SEG_AD_ER_1 *(geometrically singular)*	SEG_AD_ER_1 ≅ SEG_AD_ER_4	SEG_AD_ER_1 ≅ SEG_AD_ER_5
SEG_AD_ER_2 *(geometrically singular)*		SEG_AD_ER_2 *(geometrically singular)*	SEG_AD_ER_2 ≅ SEG_AD_ER_5	SEG_AD_ER_2 ≅ SEG_AD_ER_6
		SEG_AD_ER_3 *(geometrically singular)*		SEG_AD_ER_3 ≅ SEG_AD_ER_7
		SEG_AD_ER_4 *(geometrically singular)*		
Segments length : *240°Crk*	*Segments length :* *180°Crk*	*Segments length :* *144°Crk*	*Segments length :* *120°Crk*	*Segments length :* *90°Crk*

21.5.1 Fade out conditions of engine roughness segment adaptive process

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
SEG_AD_FDOUT_ER_CDN	V	0...FFH	0...255	1	[-]
Segment adaptation fade out conditions carrier byte (before masking)					
SEG_AD_FDOUT_ER_CDN_NR	V	0...8H	0...8	1	[-]
Segment adaptation fade out conditions number (after masking)					
LV_SEG_AD_FDOUT_ER	V	0...1H	0...1	1	[-]
Boolean for segment adaptation fade out condition					
LV_INH_MAF_DIF_ER_AD	V	0...1H	0...1	1	[-]
ER adaptive process fade out request flag due to MAF transient condition					
LV_INH_LOAD_GRD_MAX_ER_AD	V	0...1H	0...1	1	[-]
ER adaptive process fade out request flag due to engine load transient condition					
LV_INH_TPS_GRD_ER_AD	V	0...1H	0...1	1	[-]
ER adaptive process fade out request flag due to TPS transient condition					
LV_INH_ACC_ER_AD	V	0...1H	0...1	1	[-]
ER adaptive process fade out request flag due to Air conditioner activation					

Input data:

LV_ACCOUT_RLY	MAF	LOAD_MIS	LV_ENA_SEG_T_MES
TPS_GRD	TPS_AV	LV_STATE_RR	LOAD_GRD_MIS
MAF_DIF	LV_SEG_AD_DIF_MAX_ER	LV_SEG_AD_LIM_ER	LV_INH_OBD_ER_AD
LV_INH_APP_ER_AD			

FUNCTION DESCRIPTION:

Description:

LV_SEG_AD_FDOUT_ER combines all system conditions to generate a fade out condition as input for the generic segment adaptation process.

Update rate: segment occurrence for conditions & data process

10ms for free running decounters

Application conditions:

Activation: *LV_ENA_SEG_T_MES = 1*

Deactivation: *LV_ENA_SEG_T_MES = 0*

Action on deactivation event ***Or*** *ECU reset:*

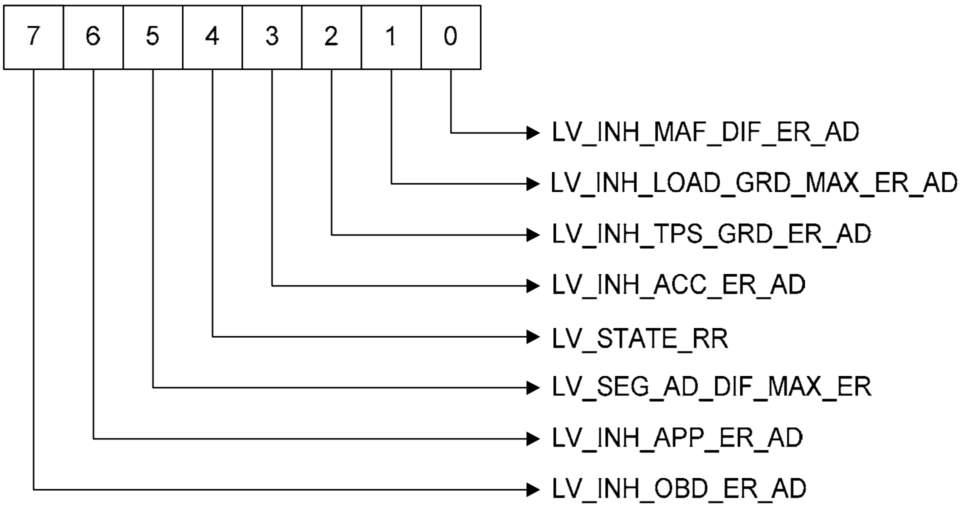
LV_SEG_AD_FDOUT_ER = 1

SEG_AD_FDOUT_ER_CDN = 0xFF

SEG_AD_FDOUT_ER_CDN_NR = 8

Definition of segment adaptation fade out conditions carrier:

SEG_AD_FDOUT_ER_CDN : *Carrier used for fade-out conditions merge*



21.5.1.1 Configurable Fade-out Management

Application conditions:

The Fade out management of the segment adaptive process can be configured with a bitfield mask calibration that allow to take in account or not some conditions set in the carrier.

If the corresponding bit in the SEG_AD_FDOUT_ER_CDN carrier structure is set to 0 in C_MASK_SEG_AD_FDOUT_ER calibration, then the corresponding condition will not fade-out the segment adaptive process via LV_SEG_AD_FDOUT_ER.

The number of fade-out conditions not masked by this calibration is set in SEG_AD_FDOUT_ER_CDN_NR.

Formula section:

SEG_AD_FDOUT_ER_CDN_NR =

sum (SEG_AD_FDOUT_ER_CDN **&** C_MASK_SEG_AD_FDOUT_ER)

! (bitfield operation) !

```
If    SEG_AD_FDOUT_ER_CDN_NR ≠ 0
Then      LV_SEG_AD_FDOUT_ER = 1
Else      LV_SEG_AD_FDOUT_ER = 0
EndIf
```

21.5.1.2 Maximum air-mass gradient

Due to trailing throttle / acceleration transition problems it is necessary to disable ER segment learning process for a short period when the air-mass gradient per segment exceeds a calibration value.

Application conditions:

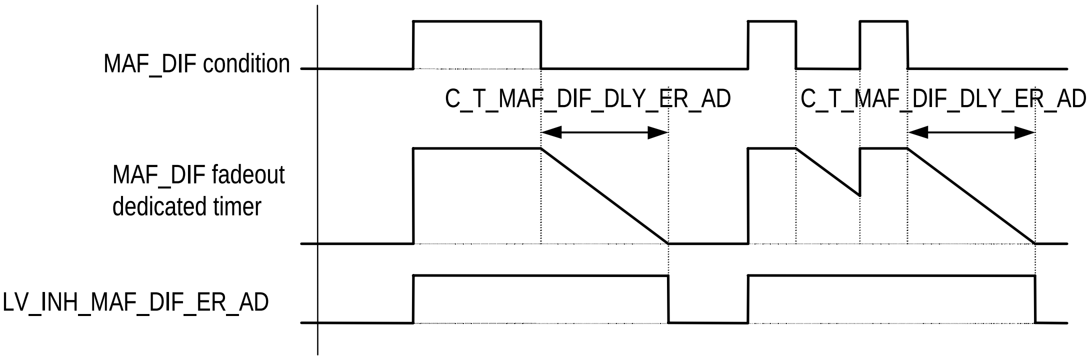
ER segment adaptation is disabled while the absolute of the air-mass gradient exceeds the value mentioned below:

After the last triggering of this segment adaptation is suppressed for a period of C_T_MAF_DIF_DLY_ER_AD.

Formula section:

```
If    | MAF_DIF | > IP_MAF_DIF_MAX_ER_AD(MAF)
Then
    After condition rising edge triggering, LV_INH_MAF_DIF_ER_AD flag is set to 1 as long as condition is true.
    After condition falling edge triggering, LV_INH_MAF_DIF_ER_AD flag is hold to 1 for a period of C_T_MAF_DIF_DLY_ER_AD.
EndIf
```

Fade out behaviour summary:



21.5.1.3 Maximum actual load gradient

Due to general intervention of the torque model it's necessary to disable ER segment adaptation for a short period when the actual torque gradient exceeds a calibration value.

Application conditions:

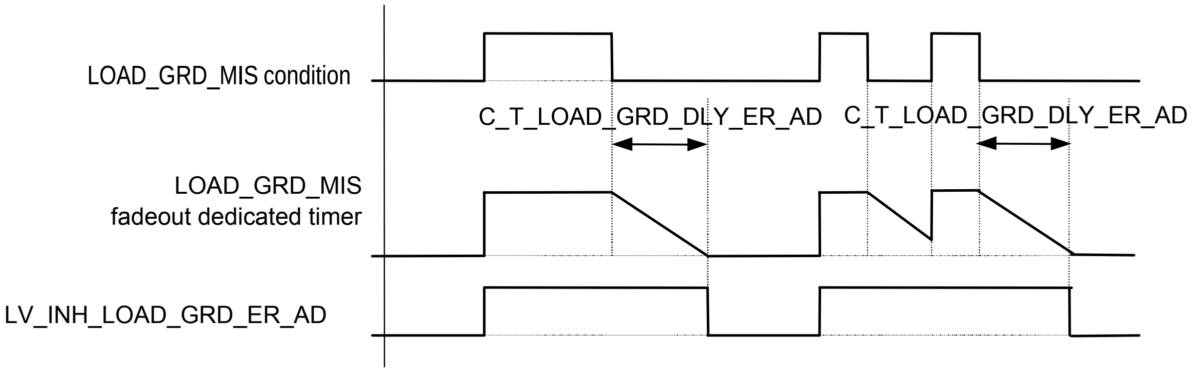
ER segment adaptation is disabled while the absolute of the torque gradient – here LOAD_GRD_MIS the reference for Misfire detection process – exceeds the value mentioned below.

After the last triggering of this segment adaptation is suppressed for a period of C_T_LOAD_GRD_DLY_ER_AD.

Formula section:

```
If    LOAD_GRD_MIS > IP_LOAD_GRD_ER_AD(LOAD_MIS)
Then
    After condition rising edge triggering, LV_INH_LOAD_GRD_MAX_ER_AD flag is set to 1 as long as condition is true.
    After condition falling edge triggering, LV_INH_LOAD_GRD_MAX_ER_AD flag is hold to 1 for a period of C_T_LOAD_GRD_DLY_ER_AD.
EndIf
```

Fade out behaviour summary:



21.5.1.4 Maximum throttle gradient

Due to trailing throttle / acceleration transient problems at low load it is necessary to disable ER segment adaptation for a short period when the throttle gradient exceeds a calibration value.

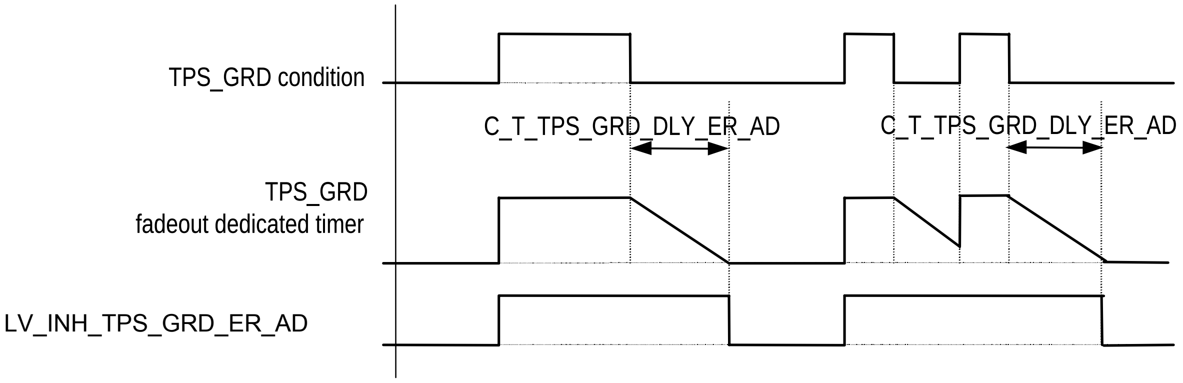
Application conditions:

ER segment adaptation is disabled while the throttle position gradient exceeds the value mentioned below.

After the last triggering of this segment adaptation is suppressed for a period of C_T_TPS_GRD_DLY_ER_AD.

Formula section:

```
If   TPS_GRD > IP_TPS_GRD_MAX_ER_AD(TP_AV)
Then
     After condition rising edge triggering, LV_INH_TPS_GRD_ER_AD flag is set to 1 as long as condition is true.
     After condition falling edge triggering, LV_INH_TPS_GRD_ER_AD flag is hold to 1 for a period of C_T_TPS_GRD_DLY_ER_AD.
EndIf
```



21.5.1.5 Air - conditioning compressor activation

When the air - conditioning compressor is switched on, an additional load is briefly applied to the engine.

This load jump can cause an ER segment period jump and crankshaft vibration.

Application conditions:

Engine roughness segment adaptation can be suppressed for the applicable constant period C_T_ACCOUT_DLY_ER_AD, starting if the air - conditioning compressor is switched on or off (LV_ACCOUT_RLY changes from 0 to 1 or from 1 to 0), during this period LV_INH_ACC_ER_AD is set to 1.

Formula section:

```
If    LV_ACCOUT_RLY = 0 → 1
      Or     LV_ACCOUT_RLY = 1 → 0
Then         LV_INH_ACC_ER_AD is set to 1 during C_T_ACCOUT_DLY_ER_AD
EndIf
```

21.5.1.6 Rough road condition active

Due to possible feedback from the driven wheels to the crankshaft it is necessary to fade out. ER segment adaptation when rough road condition is active (LV_STATE_RR = 1).

21.5.1.7 Engine roughness adaptive values on the same physical segment out of range

Due to possible feedback from Dual Mass Flywheel oscillations *(see transmission/crankshaft design)* in adaptive process engine speed range, fluctuations on ER adaptive values of identical physical segments can induce dissimilarity in ER segment correction. LV_SEG_AD_DIF_MAX_ER indicates that the engine roughness adaptive values difference on the same physical segment is out of range.

This flag is produced in the following chapter called: Engine roughness adaptive values difference on the same physical segment out of range.

21.5.1.8 Application specific inhibition request for engine roughness adaptive process

Due to possible accessories triggering specific to the application (CVT shifts...), an inhibition of the engine roughness adaptive process can be applied.

The inhibition is applied when LV_INH_APP_ER_AD = 1.

21.5.1.9 Inhibition request for engine roughness adaptive process linked to OBD errors

Due to possible troubles induced by OBD I sensors errors (CAM, CRK...), the engine roughness adaptive process can be inhibited by the flag LV_INH_OBD_ER_AD.

The inhibition is applied when LV_INH_OBD_ER_AD = 1.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_MAF_DIF_MAX_ER_AD	6	0...FFFFH	0...1389	2.12E-02	[mg/stk]
LDP_MAF_IP_MAF_DIF_MAX_ER_AD	6	0...FFFFH	0...1389	2.12E-02	[mg/stk]
Maximum air-mass gradient per SEG for fade out condition.					
IP_LOAD_GRD_ER_AD	6	0...7FFFH	0...99.99694	3.05E-03	[%]
LDP_LOAD_MIS_IP_LOAD_GRD_ER_AD	6	0...7FFFH	0...99.99694	3.05E-03	[%]
Maximum actual torque gradient fade out condition.					
IP_TPS_GRD_MAX_ER_AD	6	0...FFH	0...2987.5	11.7156863	[°TPS/s]
LDP_TPS_AV_IP_TPS_GRD_MAX_ER	6	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Maximum throttle gradient threshold for fade out condition.					
C_T_MAF_DIF_DLY_ER_AD	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when maximum air-mass gradient has been detected.					
C_T_LOAD_GRD_DLY_ER_AD	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when maximum actual torque gradient has been detected.					
C_T_TPS_GRD_DLY_ER_AD	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when maximum throttle gradient has been detected.					
C_T_ACCOUT_DLY_ER_AD	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when air-conditioning compressor has been switched on or off					
C_MASK_SEG_AD_FDOUT_ER	1	0...FFH	0...255	1	[-]
Configuration mask for SEG_AD_FDOUT_ER_CDN fade out carrier structure					

21.5.2 End of line specific request for ER segment adaptive values learning process

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
LV_SEG_AD_FDOUT_ER_EOL	V	0...1H	0...1	1	[-]
EOL specific request for ER segment adaptive learning process					

Input data:

LV_EOL_OBD	N_32	DRV1_ER	SEG_NR_MES
LV_INJ_CUT	LV_ENA_SEG_T_MES	LV_AT	LV_VAR_TCT

General information:

During end of line plant, specific conditions to activate ER adaptive process can be used. If OBD end of line process occurs, this request has priority on nominal conditions.

Update rate: ER segment occurrence

Application conditions:

Activation: *LV_ENA_SEG_T_MES = 1*

Deactivation: *LV_ENA_SEG_T_MES = 0*

Action on deactivation event ***Or*** *ECU reset:*

LV_SEG_AD_FDOUT_ER_EOL = 1

Description:

If LV_EOL_OBD = 1 *// OBD end of line process request*

And LV_INJ_CUT = 1 *(all cylinders shut-off)* during last C_NR_SEG_AD_ACT_MIN_ER_EOL segments since last SEG_NR_MES = 0

```
    And  {  [ LV_AT = 1  And  N_32 < C_N_MAX_SEG_AD_ER_EOL_AT
                         And  N_32 > C_N_MIN_SEG_AD_ER_EOL_AT ]
            Or  [ LV_AT = 0  And  N_32 < C_N_MAX_SEG_AD_ER_EOL_MT
                And  LV_VAR_TCT = 0  And  N_32 > C_N_MIN_SEG_AD_ER_EOL_MT ]
            Or  [ LV_AT = 0  And  N_32 < C_N_MAX_SEG_AD_ER_EOL_TCT
                And  LV_VAR_TCT = 1  And  N_32 > C_N_MIN_SEG_AD_ER_EOL_TCT] }

    And  | DRV1_ER | < C_DRV1_MAX_SEG_AD_ER_EOL during
                        C_NR_SEG_DRV1_MIN_ER_EOL segments
    And  LC_SEG_AD_ER_REQ_EOL = 1
Then     // Request in specific EOL range for segment adaptive process
         LV_SEG_AD_FDOUT_ER_EOL = 0
Else     // No specific request
         LV_SEG_AD_FDOUT_ER_EOL = 1
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_N_MAX_SEG_AD_ER_EOL_MT	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed threshold for crankshaft target wheel learning for EOL request mode in MT					
C_N_MIN_SEG_AD_ER_EOL_MT	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed threshold for crankshaft target wheel learning for EOL request mode in MT					
C_N_MAX_SEG_AD_ER_EOL_AT	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed threshold for crankshaft target wheel learning for EOL request mode in AT					
C_N_MIN_SEG_AD_ER_EOL_AT	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed threshold for crankshaft target wheel learning for EOL request mode in AT					
C_DRV1_MAX_SEG_AD_ER_EOL	1	0...7FFFH	0...32767	1	[µs]
Maximum value of DRV1_ER (dynamic part of engine roughness) for crankshaft target wheel learning for EOL request mode					
C_NR_SEG_AD_ACT_MIN_ER_EOL	1	0...FFH	0...255	1	[-]
Minimum segments number in fuel cut-off mode before triggering ER segment adaptive learning process for EOL request mode					
C_NR_SEG_DRV1_MIN_ER_EOL	1	0...FFH	0...255	1	[-]
Minimum number of segments for engine roughness dynamic part for EOL request mode					
LC_SEG_AD_ER_REQ_EOL	1	0...1H	0...1	1	[-]
Specific EOL request enable flag for ER segment adaptive value learning process					
C_N_MAX_SEG_AD_ER_EOL_TCT	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed threshold for crankshaft target wheel learning for EOL request mode in TCT					
C_N_MIN_SEG_AD_ER_EOL_TCT	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed threshold for crankshaft target wheel learning for EOL request mode in TCT					

21.5.3 Engine roughness adaptive learning process management

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
THD_AD_ER	V/O	0...FFFFH	0...65535	1	[µs]
Engine roughness threshold correction factor during segment adaptation learning process.					
DELTA_CRK_DIF_MAX_ER	V/S	0...FFFFH	0...999.98474	1.53E-02	[°/oo]
Limitation of adaptation range (mech. Tolerance)					
CRLC_SEG_AD_ER	V	0...FFFFH	0...0.99998	1.53E-05	[-]
ER segment adaptive values filtering coefficient					
CTR_SEG_AD_ER	V/S	0...FFFFH	0...65535	1	[-]
Counter of ER segment adaptation learning process					
LV_SEG_AD_AVL_ER	V/O/S	0...1H	0...1	1	[-]
Segment adaptation process state : no adaptive process executed (=0) fast adaptive process achieved (=1)					
LV_SEG_AD_RAW_ER	V	0...1H	0...1	1	[-]
ER segment adaptation process monitoring enable					
LV_SEG_AD_ER	V/O	0...1H	0...1	1	[-]
ER segment adaptation process enable					

Input data:

LV_INH_OBD_ER_AD	LV_SEG_AD_FDOUT_ER	N_32	DRV1_ER
LV_INJ_CUT	LV_ENA_SEG_T_MES	LV_SEG_AD_FDOUT_ER_EOL	LV_SEG_AD_RST_ER_EOL
SEG_NR_MES	NC_CYL_NR	SEG_T_COR	LV_ENA_ER
LV_AT	LV_VAR_TCT		

Description:

The adaptive process is managed in two phases:

- A fast adaptive process for end of line procedures (EOL), who allow a fast convergence to ER flywheel adaptive values.
- At the end of the fast adaptive process, a slow adaptive process who will monitor ER adaptive values during the engine lifetime.

An end of line specific filtering mode, to be used on EOL or special service request.

At the beginning of the segment adaptation learning process, it is necessary to upper the engine roughness value with an additional offset (THD_AD_ER), depending on the target wheel tolerances to be excepted.

For instance, a flywheel with a ± 0.3°Crk mechanical tolerance on the ER segment corresponds to following segment drift:

Engine Type	ER Segment length	Induced Drift with ± 0.3°Crk tolerance on ER segment
3 cyl engine	240°Crk	2.5‰
4 cyl engine	180°Crk	3.33‰
5 cyl engine	144°Crk	4.17‰
6 cyl engine	120°Crk	5‰
8 cyl engine	90°Crk	6.68‰

The value by which the ER is increased at the beginning is gradually reduced to zero in a linear way until the 95% correction is obtained.

This number of steps is controlled by an internal adaptation counter CTR_SEG_AD_ER.

21.5.3.1 Engine roughness adaptive learning process

First time initialisation :

The ER segment adaptive values learning process is initialised when :

- ECU is initialised for the first time *(flash memory formatting)*
- A loss of adaptation values has been detected *(flash memory corrupted)*.
- A specific external tool request occurs *(use of **LV_SEG_AD_RST_ER_EOL**)*

// ER segment adaptive process control variables reseted

```
CTR_SEG_AD_ER = 0
DELTA_CRK_DIF_MAX_ER = NC_CYL_NR * 4 * C_CRK_DIF_MAX_ER
LV_SEG_AD_AVL_ER = 0
```

Update rate: ER segment task, every engine cycle when SEG_NR_MES = 0 ; SEG_T_MES_0 segment acquired.

Application conditions:

Activation/Deactivation: (based on LV_SEG_AD_RAW_ER evaluation)

// Raw adaptive value monitoring activation

```
If    LV_INJ_CUT = 1        // All cylinders shut-off
      And   N_32 < C_N_SEG_T_AD_RAW_MAX_ER
      And   N_32 > C_N_SEG_T_AD_RAW_MIN_ER
      And   LV_ENA_SEG_T_MES = 1
Then        LV_SEG_AD_RAW_ER = 1
Else        LV_SEG_AD_RAW_ER = 0
            LV_SEG_AD_ER = 0
EndIf
```

Formula section:

// Adaptive value learning process activation

```
If(1) LV_SEG_AD_FDOUT_ER = 0
      And [  ( LV_INJ_CUT = 1 (all cylinders shut-off) during last
                  C_NR_SEG_AD_ACT_MIN_ER segments since last SEG_NR_MES = 0 )
              And  {   [  LV_AT = 1      And   N_32 < C_N_MAX_SEG_AD_ER_AT
                                         And   N_32 > C_N_MIN_SEG_AD_ER_AT  ]
                       Or
                       [  LV_AT = 0      And   N_32 < C_N_MAX_SEG_AD_ER_TCT
                 And LV_VAR_TCT = 1      And   N_32 > C_N_MIN_SEG_AD_ER_ TCT  ]   }
```

```
        Or
                [ LV_AT = 0        And   N_32 < C_N_MAX_SEG_AD_ER_MT
            And LV_VAR_TCT = 0     And   N_32 > C_N_MIN_SEG_AD_ER_ MT ] }

        And        | DRV1_ER | < C_DRV1_MAX_SEG_AD_ER during
                           C_NR_SEG_DRV1_MIN_ER segments )
        Or  LV_SEG_AD_FDOUT_ER_EOL = 0    ]
Then(1)  LV_SEG_AD_ER = 1                    // ER adaptive learning process enable
    If(2)   LV_SEG_AD_AVL_ER = 0
    Then (2)       // First adaptive process phase
        If(3)    LV_SEG_AD_FDOUT_ER_EOL = 0
        Then(3)        CRLC_SEG_AD_ER = C_CRLC_SEG_AD_ER_EOL
        Else(3)
            If(4)    LV_VAR_TCT = 0
            Then(4)
                     CRLC_SEG_AD_ER = C_CRLC_FAST_SEG_AD_ER
            Else(4)
                     CRLC_SEG_AD_ER = C_CRLC_FAST_SEG_AD_ER_TCT
            EndIf(4)
        EndIf(3)
        DELTA_CRK_DIF_MAX_ER(n) = DELTA_CRK_DIF_MAX_ER(n-1)
            -  NC_CYL_NR * 4 * C_CRK_DIF_MAX_ER * CRLC_SEG_AD_ER
               ------------------------------------------------
                                      3
        CTR_SEG_AD_ER = CTR_SEG_AD_ER + 1      (with saturation to max value)
        If(3)    CTR_SEG_AD_ER ≥ C_NR_SEG_AD_AVL_ER
        Then(3)        LV_SEG_AD_AVL_ER = 1   // First adaptive process phase achieved
        EndIf(3)
    Else(2)        // Engine lifetime adaptive process phase
        CTR_SEG_AD_ER = CTR_SEG_AD_ER + 1      (with saturation to max value)
        If(2)    LV_SEG_AD_FDOUT_ER_EOL = 0
        Then(2)        CRLC_SEG_AD_ER = C_CRLC_SEG_AD_ER_EOL
        Else(2)
            If(4)    LV_VAR_TCT = 0
            Then(4)
                     CRLC_SEG_AD_ER = C_CRLC_SEG_AD_ER
            Else(4)
```

```
                    CRLC_SEG_AD_ER = C_CRLC_SEG_AD_ER_TCT
                EndIf(4)
            EndIf(2)


        EndIf(1)
Else(1)     LV_SEG_AD_ER = 0                    // ER adaptive learning process disable
EndIf(1)
```

21.5.3.2 Engine roughness offset during first adaptive process phase

Update rate: ER segment task

Application conditions:

Activation: LV_ENA_ER = 1

Deactivation: LV_ENA_ER = 0

Action on deactivation condition ***Or*** *ECU reset :* THD_AD_ER = 0

Formula section:

```
If    LV_SEG_AD_AVL_ER = 0
Then        // The ER offset factor THD_AD_ER is decreased via DELTA_CRK_DIF_MAX_ER value
            THD_AD_ER = DELTA_CRK_DIF_MAX_ER * SEG_T_COR
Else        THD_AD_ER = 0
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_N_MAX_SEG_AD_ER_MT	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed threshold for crankshaft target wheel learning with MT					
C_N_MIN_SEG_AD_ER_MT	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed threshold for crankshaft target wheel learning with MT					
C_N_MAX_SEG_AD_ER_AT	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed threshold for crankshaft target wheel learning with AT					
C_N_MIN_SEG_AD_ER_AT	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed threshold for crankshaft target wheel learning with AT					
C_N_SEG_T_AD_RAW_MAX_ER	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed threshold for crankshaft target wheel monitoring					
C_N_SEG_T_AD_RAW_MIN_ER	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed threshold for crankshaft target wheel monitoring					
C_CRK_DIF_MAX_ER	1	0...1FFH	0...7.79724	0.0152588	[°/oo]
Limitation of ER adaptation range (mechanical tolerance)					
C_DRV1_MAX_SEG_AD_ER	1	0...7FFFH	0...32767	1	[µs]
Maximum value of DRV1_ER (dynamic part of engine roughness) for crankshaft target wheel learning.					
C_NR_SEG_AD_ACT_MIN_ER	1	0...FFH	0...255	1	[-]
Minimum segments number in fuel cut-off mode before triggering ER segment adaptive learning process					
C_NR_SEG_AD_AVL_ER	1	0...FFH	0...255	1	[-]
Segments number to achieve the first adaptive process phase					
C_NR_SEG_DRV1_MIN_ER	1	0...FFH	0...255	1	[-]
Minimum number of segments for engine roughness dynamic part					
C_CRLC_SEG_AD_ER	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Nominal ER segment adaptive values filtering coefficient					
C_CRLC_FAST_SEG_AD_ER	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Fast ER segment adaptive values filtering coefficient (fast learning phase)					
C_CRLC_SEG_AD_ER_EOL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
EOL specific filtering coefficient for ER segment adaptive values learning process					
C_N_MAX_SEG_AD_ER_TCT	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed threshold for crankshaft target wheel learning with TCT					
C_N_MIN_SEG_AD_ER_TCT	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed threshold for crankshaft target wheel learning with TCT					
C_CRLC_SEG_AD_ER_TCT	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Nominal ER segment adaptive values filtering coefficient (TCT vehicule)					
C_CRLC_FAST_SEG_AD_ER_TCT	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Fast ER segment adaptive values filtering coefficient (fast learning phase - TCT vehicule)					

21.5.4 Engine roughness adaptive values calculation & filtering

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
SEG_AD_RAW_MMV_ER[NC_CYL_NR]	V	8000...7FFFH	-7.8125...7.81226	2.38E-04	[°/oo]
ER filtered adaptive values used for learning process monitoring (for each segment)					
SEG_AD_MMV_ER[NC_CYL_NR]	V/O/S	8000...7FFFH	-7.8125...7.81226	2.38E-04	[°/oo]
ER filtered adaptive values used for learning process (for each segment)					
LV_SEG_AD_LIM_ER	V/O/S	0...1H	0...1	1	[-]
ER filtered adaptive values out of range					

Input data:

LV_SEG_AD_FDOUT_ER_EOL	LV_SEG_AD_AVL_ER	CRLC_SEG_AD_ER	LV_SEG_AD_RST_ER_EOL
LV_SEG_AD_ER	LV_SEG_AD_RAW_ER	SEG_T_MES_CYL[NC_CYL_NR]	SEG_NR_MES
LV_REQ_SEG_AD_MMV_ER_APP	SEG_AD_MMV_ER_APP[NC_CYL_NR]	LV_SEG_AD_ER	CTR_SEG_AD_ER

FUNCTION DESCRIPTION:

This function allows to determinate the coefficients for ER segment adaptive values filters used in the learning process. Three different stages can be differentiate :

- Nominal ER segment adaptive values filtering mode, used to monitor adaptive values during engine lifetime.
- Fast ER segment adaptive values filtering mode, used to quickly obtain valid adaptive values when the EMS is new.
- EOL specific filtering mode, to be used on EOL or special service request.

LV_SEG_AD_LIM_ER is set if at least one SEG_AD_MMV_ER_x value exceeds C_SEG_AD_MAX_ER.

First time initialisation :

The ER segment adaptive values learning process is initialised when :

- ECU is initialised for the first time *(flash memory formatting)*
- A loss of adaptation values has been detected *(flash memory corrupted)*.
- A specific external tool request occurs *(use of LV_SEG_AD_RST_ER_EOL defined in Application Incidences file)*

// ER segment adaptive values reseted

```
For x = 0 ... [NC_CYL_NR-1]
    SEG_AD_MMV_ER[x] = 0
EndFor
LV_SEG_AD_LIM_ER = 0
```

Initialisation at ECU reset:

```
For x = 0...(NC_CYL_NR - 1)
```

```
        SEG_AD_RAW_MMV_ER[x] = SEG_AD_MMV_ER[x]
    EndFor
```

Update rate: ER segment occurrence

Application conditions:

This process is executed every engine cycle when SEG_T_MES_0 (reference segment) is acquired and when the monitoring of the adaptive learning process is active

Activation: LV_SEG_AD_RAW_ER = 1

Process on activation transient:

```
    If      LC_SEG_AD_LIM_REAC_ER = 1    // reactivation of the adaptive values range check
    Then    LV_SEG_AD_LIM_ER = 0
    Else    No Operation
    EndIf
```

Deactivation: LV_SEG_AD_RAW_ER = 0

Formula section: *Temporary data are in italic*

```
For(1) x = 0 ... to ... [NC_CYL_NR-1]
    Deceleration correction factor (temporary data):
    FAC_DECE_SEG_AD_ER[x] =
            (x / NC_CYL_NR) * (SEG_T_MES_CYL[0] - SEG_T_MES_CYL[0]_prev)
    Raw adaptive values:
    SEG_AD_ER[x] =
      [ (SEG_T_MES_CYL[x] - SEG_T_MES_CYL[0]_prev) - FAC_DECE_SEG_AD_ER[x] ] /
          SEG_T_MES_CYL[0]_prev
    Engine roughness adaptive values filtering for monitoring and learning process:
    SEG_AD_RAW_MMV_ER[x]$_{i}$ = SEG_AD_RAW_MMV_ER[x]$_{i-1}$ +
          C_CRLC_SEG_AD_RAW_ER * (SEG_AD_ER[x]$_{i}$ - SEG_AD_RAW_MMV_ER[x]$_{i-1}$)
        If(2)     LV_SEG_AD_ER = 1
        Then(2)       If(3)       CTR_SEG_AD_ER = 1
                      Then(3)     SEG_AD_MMV_ER[x]$_{i}$ = SEG_AD_ER[x]$_{i}$
                      Else(3)     SEG_AD_MMV_ER[x]$_{i}$ = SEG_AD_MMV_ER[x]$_{i-1}$ +
                    CRLC_SEG_AD_ER * [SEG_AD_ER[x]$_{i}$ - SEG_AD_MMV_ER[x]$_{i-1}$]
                  EndIf(3)
                  Engine roughness adaptive values out of range check:
                  If(3)     | SEG_AD_MMV_ER[x] | >= C_SEG_AD_MAX_ER
                  Then(3)       LV_SEG_AD_LIM_ER = 1
                  EndIf(3)
    EndIf(2)
```

```
        If(2)   LV_SEG_AD_ER = 1 → 0
                And     LV_SEG_AD_AVL_ER = 1
                                //no risk because THD_AD_ER = 0, CTR_SEG_AD_ER no more used
                And     LV_REQ_SEG_AD_MMV_ER_APP = 1
        Then(2)         SEG_AD_MMV_ER[x] = SEG_AD_MMV_ER_APP[x]
                        LV_SEG_AD_LIM_ER = 0
        EndIf(2)
EndFor(1)
```

<u>where:</u>

SEG_AD_ER[x] : *raw adaptive value for logical segment x, <u>temporary data</u>*

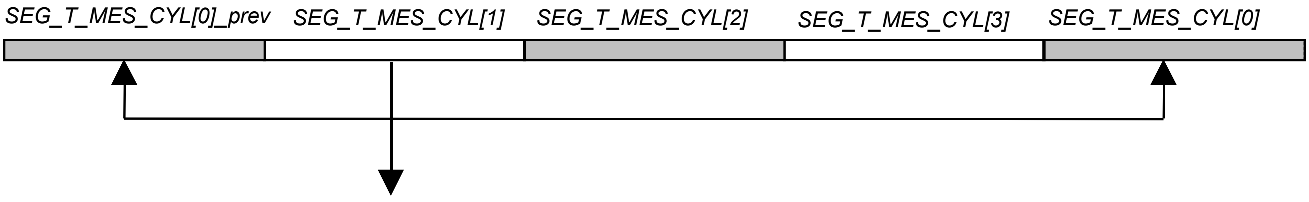
FAC_DECE_SEG_AD_ER[x] : *deceleration correction factor, <u>temporary data</u>*

SEG_AD_RAW_MMV_ER[x] : *fast filtered adaptive value for logical segment x*

SEG_AD_MMV_ER[x] : *filtered adaptive value for logical segment x*

SEG_T_MES_CYL[0]_prev : *raw segment period of reference segment (SEG_NR_MES=0), on previous engine cycle, <u>temporary data,</u> oldest value*

SEG_T_MES_CYL[0] : *raw segment period of reference segment (SEG_NR_MES=0) after NC_CYL_NR segments (720° crank angle), newest value*



FAC_DECE_SEG_AD_ER[1] =
*1/NC_CYL_NR * [SEG_T_MES_CYL[0] - SEG_T_MES_CYL[0]_prev]*

<u>Calibration data:</u>

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit

LC_SEG_AD_LIM_REAC_ER	1	0...1H	0...1	1	[-]
Reactivation of the ER segment adaptation range check at each learning phase					
C_CRLC_SEG_AD_RAW_ER	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Filter coefficient for ER segment adaptive values monitoring					
C_SEG_AD_MAX_ER	1	0...7FFFH	0...7.81226	2.38E-04	[°/oo]
Limitation of ER segment adaptation range (mechanical tolerance).					

21.5.5 Engine roughness adaptive values difference on the same physical segment out of range

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SEG_AD_DIF_MAX_ER	V/O	0...1H	0...1	1	[-]
Maximum ER adaptive values difference on same physical segment					
#IF NC_CYL_NR = 4, 6 Or 8					
SEG_AD_DIF_ER_0	V	0...7FFFH	0...7.81226	2.38E-04	[°/oo]
ER adaptive values difference on same physical segment (Segment 0 & Identical)					
SEG_AD_DIF_ER_1	V	0...7FFFH	0...7.81226	2.38E-04	[°/oo]
ER adaptive values difference on same physical segment (Segment 1 & Identical)					
#ENDIF					
#IF NC_CYL_NR = 6 Or 8					
SEG_AD_DIF_ER_2	V	0...7FFFH	0...7.81226	2.38E-04	[°/oo]
ER adaptive values difference on same physical segment (Segment 2 & Identical)					
#ENDIF					
#IF NC_CYL_NR = 8					
SEG_AD_DIF_ER_3	V	0...7FFFH	0...7.81226	2.38E-04	[°/oo]
ER adaptive values difference on same physical segment (Segment 3 & Identical)					
#ENDIF					

Input data:

SEG_AD_RAW_MMV_ER[NC_CYL_NR]	NC_CYL_NR	SEG_NR_MES	LV_SEG_AD_RAW_ER

General information:

Due to possible feedback from Dual Mass Flywheel oscillations *(see transmission/crankshaft design)* in adaptive process engine speed range, fluctuations on ER adaptive values of identical physical segments can induce dissimilarity in ER segment correction.

If difference between identical physical segments ER fast filtered adaptive values exceeds a calibratable gap, a fade out is applied.

ER identical physical segments adaptive value differences are evaluated each 720°Crk, for that this process is executed every engine cycle when SEG_T_MES_0 (*SEG_NR_MES = 0)* is acquired and when the adaptive monitoring process is active.

Update rate: *ER segment occurrence,*

Application conditions:

Activation: SEG_NR_MES = 0

And LV_SEG_AD_RAW_ER = 1

Deactivation: SEG_NR_MES != 0

Or LV_SEG_AD_RAW_ER = 0

Initialisation : *on ECU reset:*

```
    LV_SEG_AD_DIF_MAX_ER = 0
    #If      NC_CYL_NR = 4, 6 or 8
      SEG_AD_DIF_ER_0 = 0
      SEG_AD_DIF_ER_1 = 0
    #EndIf
    #If      NC_CYL_NR = 6 or 8
      SEG_AD_DIF_ER_2 = 0
    #EndIf
    #If      NC_CYL_NR = 8
      SEG_AD_DIF_ER_3 = 0
    #EndIf
```

Formula section:

```
#If  NC_CYL_NR = 3 Or NC_CYL_NR = 5
     LV_SEG_AD_DIF_MAX_ER = 0
#EndIf
#If  NC_CYL_NR = 4
     SEG_AD_DIF_ER_0 = | SEG_AD_RAW_MMV_ER_0 - SEG_AD_RAW_MMV_ER_2 |
     SEG_AD_DIF_ER_1 = | SEG_AD_RAW_MMV_ER_1 - SEG_AD_RAW_MMV_ER_3 |
     If     SEG_AD_DIF_ER_0 >= C_SEG_AD_DIF_MAX_ER
            Or     SEG_AD_DIF_ER_1 >= C_SEG_AD_DIF_MAX_ER
     Then   LV_SEG_AD_DIF_MAX_ER = 1
     Else   LV_SEG_AD_DIF_MAX_ER = 0
     EndIf
#EndIf
#If  NC_CYL_NR = 6
     SEG_AD_DIF_ER_0 = | SEG_AD_RAW_MMV_ER_0 - SEG_AD_RAW_MMV_ER_3 |
     SEG_AD_DIF_ER_1 = | SEG_AD_RAW_MMV_ER_1 - SEG_AD_RAW_MMV_ER_4 |
     SEG_AD_DIF_ER_2 = | SEG_AD_RAW_MMV_ER_2 - SEG_AD_RAW_MMV_ER_5 |
     If     SEG_AD_DIF_ER_0 >= C_SEG_AD_DIF_MAX_ER
            Or     SEG_AD_DIF_ER_1 >= C_SEG_AD_DIF_MAX_ER
                   Or     SEG_AD_DIF_ER_2 >= C_SEG_AD_DIF_MAX_ER
     Then   LV_SEG_AD_DIF_MAX_ER = 1
     Else   LV_SEG_AD_DIF_MAX_ER = 0
     EndIf
#EndIf
#If  NC_CYL_NR = 8
```

```
SEG_AD_DIF_ER_0 = | SEG_AD_RAW_MMV_ER_0 - SEG_AD_RAW_MMV_ER_4 |
SEG_AD_DIF_ER_1 = | SEG_AD_RAW_MMV_ER_1 - SEG_AD_RAW_MMV_ER_5 |
SEG_AD_DIF_ER_2 = | SEG_AD_RAW_MMV_ER_2 - SEG_AD_RAW_MMV_ER_6 |
SEG_AD_DIF_ER_3 = | SEG_AD_RAW_MMV_ER_3 - SEG_AD_RAW_MMV_ER_7 |
If      SEG_AD_DIF_ER_0 >= C_SEG_AD_DIF_MAX_ER
        Or      SEG_AD_DIF_ER_1 >= C_SEG_AD_DIF_MAX_ER
                Or      SEG_AD_DIF_ER_2 >= C_SEG_AD_DIF_MAX_ER
        Or      SEG_AD_DIF_ER_3 >= C_SEG_AD_DIF_MAX_ER
Then    LV_SEG_AD_DIF_MAX_ER = 1
Else    LV_SEG_AD_DIF_MAX_ER = 0
EndIf
#EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_SEG_AD_DIF_MAX_ER	1	0...7FFFH	0...7.81226	2.38E-04	[°/oo]
Maximum range for difference between same physical segment adaptive values					

21.6 Engine roughness diagnosis

21.6.1 Engine roughness segment time acquisition error

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CDN_DIAG_T_SEG_ER	V	0...1H	0...1	1	[-]
Diagnosis condition of segment time adaptive values					
ERR_SYM_T_SEG_ER	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of segment time adaptive values					
LV_ERR_T_SEG_ER	V/O	0...1H	0...1	1	[-]
Present failure for segment time adaptive values					
LV_END_DIAG_T_SEG_ER	V	0...1H	0...1	1	[-]
Diagnosis end of segment time adaptive values					

Input data:

LV_DC	LV_INH_MIS_CRK	LV_INH_DIAG_T_SEG_ER	

FUNCTION DESCRIPTION:

The purpose of this error is to diagnose synchronisation errors on engine roughness segments measurement.
An error symptom "*Missing/Adding 1 tooth or more on engine roughness segment acquisition*" is detected when a crankshaft synchronisation error caused by 1 or 2 missing/additional tooth occurs on one crankshaft revolution.

Application conditions:

Initialisation: *at ECU reset or KEY OFF/KEY ON event*

LV_CDN_DIAG_T_SEG_ER = 0

ERR_SYM_T_SEG_ER = *Refer to filtering configuration for the initialisation value*

LV_ERR_T_SEG_ER = *Refer to filtering configuration for the initialisation value*

LV_END_DIAG_T_SEG_ER = *Refer to filtering configuration for the initialisation value*

Recurrence: each 360°Crk at segment task

Activation/Deactivation:

```
If    LV_DC = 1
      And   LV_INH_DIAG_T_SEG_ER = 0
Then        LV_CDN_DIAG_T_SEG_ER = 1        diagnosis is active
Else        LV_CDN_DIAG_T_SEG_ER = 0        diagnosis is passive
EndIf
```

Formula section:

Error detection

```
If    LV_INH_MIS_CRK = 1
Then        ERR_SYM_T_SEG_ER = SYM_1
            (symptom "Missing/Adding 1 or 2 teeth on engine roughness segment
            acquisition" is active, ABC counter starts to increment)
            LV_ERR_T_SEG_ER = 1                  (after anti-bounce)
Else        ERR_SYM_T_SEG_ER = NO_SYM
            (no symptom is declared, ABC counter starts to decrement)
            LV_ERR_T_SEG_ER = 0                  (after anti-bounce)
EndIf
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_T_SEG_ER	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_T_SEG_ER	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					

Configuration for diagnostic symptoms

Diagnosis T_SEG_ER	Symptom	Nr	ABC type
Engine roughness segment time acquisition	-	*0*	STD_INI
	Missing/Adding 1 tooth or more on engine roughness segment acquisition	*1*	
		2	
T_SEG_ER		*3*	

21.6.2 Engine roughness segment adaptive values out of range error

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CDN_DIAG_SEG_AD_ER	V	0...1H	0...1	1	[-]
Diagnosis condition of ER segment time adaptive values					
ERR_SYM_SEG_AD_ER	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Symptoms of ER segment time adaptive values					
LV_ERR_SEG_AD_ER	V/O	0...1H	0...1	1	[-]
Present failure for ER segment time adaptive values					
LV_END_DIAG_SEG_AD_ER	V	0...1H	0...1	1	[-]
Diagnosis end of ER segment time adaptive values					

Input data:

LV_SEG_AD_ER	LV_SEG_AD_LIM_ER	LV_INH_DIAG_SEG_AD_ER	LV_SEG_AD_AVL_ER

Description:

The purpose is to diagnose the ER segment adaptive values.

If during or at the end of the segment adaptive values learning process *(LV_SEG_AD_ER = 1 or LV_SEG_AD_ER = 1 -> 0)*, at least one of the engine roughness segment adaptation values is at the limit (LV_SEG_AD_LIM_ER = 1), the error symptom SEG_AD_ER “engine roughness segment adaptation values at the limit” is detected.

Application conditions:

Initialisation: *at ECU reset or KEY OFF/KEY ON event*

LV_CDN_DIAG_SEG_AD_ER = 0

ERR_SYM_SEG_AD_ER = *Refer to filtering configuration for the initialisation value*

LV_ERR_SEG_AD_ER = *Refer to filtering configuration for the initialisation value*

LV_END_DIAG_SEG_AD_ER = *Refer to filtering configuration for the initialisation value*

Recurrence: segment task

Activation/Deactivaton:

```
If    [ ( LV_SEG_AD_ER = 1 -> 0 And LC_CDN_DIAG_SEG_AD_ER = 0 )
      Or    ( LV_SEG_AD_ER = 1 And LC_CDN_DIAG_SEG_AD_ER = 1 ) ]
      And   LV_SEG_AD_AVL_ER = 1              // adaptive values available
      And   LV_INH_DIAG_SEG_AD_ER = 0         // no application fade-out
Then        LV_CDN_DIAG_SEG_AD_ER = 1         // diagnosis is active
Else        LV_CDN_DIAG_SEG_AD_ER = 0         // diagnosis is passive
EndIf
```

Formula section:

Error detection

```
If    LV_SEG_AD_LIM_ER = 1                    // adaptive values at the limit
Then        ERR_SYM_SEG_AD_ER = SYM_1
            (symptom "ER segment time adaptive values at the limit" is active, ABC counter
            starts to increment)
            LV_ERR_SEG_AD_ER = 1              (after anti-bounce)
Else        ERR_SYM_SEG_AD_ER = NO_SYM
            (no symptom is declared, ABC counter starts to decrement)
            LV_ERR_SEG_AD_ER = 0              (after anti-bounce)
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SEG_AD_ER	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_SEG_AD_ER	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					
LC_CDN_DIAG_SEG_AD_ER	1	0...1H	0...1	1	[-]
Diagnosis condition mode of ER segment time adaptive values					

Configuration for diagnostic symptoms:

Diagnosis SEG_AD_ER	Symptom	Nr	ABC type
ER segment adaptation values at the limit	-	*0*	STD_INI
	ER segment adaptation values at the limit	*1*	
		2	
SEG_AD_ER		*3*	

21.7 Engine roughness diagnosis - Application Incidences

Output data:

Name	Mode	Hex. Limit	Phys. Limit	Resol.	Unit
LV_INH_DIAG_T_SEG_ER	V/O	0...1H	0...1	1	[-]
Specific conditions to inhibit Engine roughness segment time acquisition diagnosis					
LV_INH_DIAG_SEG_AD_ER	V/O	0...1H	0...1	1	[-]
Specific conditions to inhabit Engine roughness segment adaptive values diagnosis					

Input data:

LV_DC	LV_IGK	LV_ERR_MAF
LV_ERR_IVVT	LV_ERR_CAM_CUS	LV_ERR_CRK

General information:

In this module the project specific inhibition conditions for the engine roughness diagnosis are defined.

Application conditions:

Initialization: 0 at reset

Recurrence: Every segment

Activation: LV_DC = 1

Formula section:

Inhibition of SEG_AD_ER diagnosis:

LV_INH_DIAG_SEG_AD_ER = 0

Inhibition of T_SEG_ER diagnosis:

```
If      LV_IGK = 0
   Or   LV_ERR_MAF = 1
   Or   LV_ERR_IVVT = 1
   Or   LV_ERR_CAM_CUS = 1
   Or   LV_ERR_CRK = 1
Then    LV_INH_DIAG_T_SEG_ER = 1
Else    LV_INH_DIAG_T_SEG_ER = 0
Endif
```

21.8 "Engine roughness" idle diagnosis"

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ER_MMV_IS_DIAG[NC_CYL_NR]	O/V	8000...7FFFH	-8...7.99975586	2.44141E-4	-
Filtered engine roughness value cylinder individual, x = 0 ... 5					

Input data:

ER_CYL[NC_CYL_NR]	LV_IS	MAF_HB	N
TCO	NC_CYL_NR	SEG_NR_ER	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_IS_ER_DIAG	1	0...FFH	0...0.99609375	0.00390625	-
Correlation constant for calculating the ER diagnostic values					
IP_FAC_THD_IS_ER_DIAG__TCO	8	0...FFH	0...1.9921875	0.0078125	-
LDP_TCO__FAC_THD_IS_ER_DIAG	8	0...FEH	-48...142.5	0.75	°C
TCO-dependent reference value correction					
IP_THD_IS_ER_DIAG__N__MAF	4x4	1...7FFFH	-1...-3.2767E+4	-1	µs
LDP_N__THD_IS_ER_DIAG	4	0...1FE0H	0...8.16E+3	1	rpm
LDP_MAF__THD_IS_ER_DIAG	4	0...FFH	0...1.389E+3	5.44705882	mg/stk
ER reference value for ER diagnosis					

21.8.1 FUNCTION DESCRIPTION:

Conclusions on the combustion quality of a cylinder can be drawn by monitoring the crankshaft acceleration. This can be used in the workshop for diagnostic purposes.

The cylinder individual engine roughness values ER_CYL[NC_CYL_NR], which are determined for misfire diagnosis, are taken, related to a TCO- and operating point-dependent threshold and then smoothed by moving mean value calculation.

The 6 diagnostic values are transferred to the workshop tester via K-line. If an ER diagnostic value exceeds a determined threshold, it can be assumed that the corresponding cylinder has a malfunction.

Application Condition

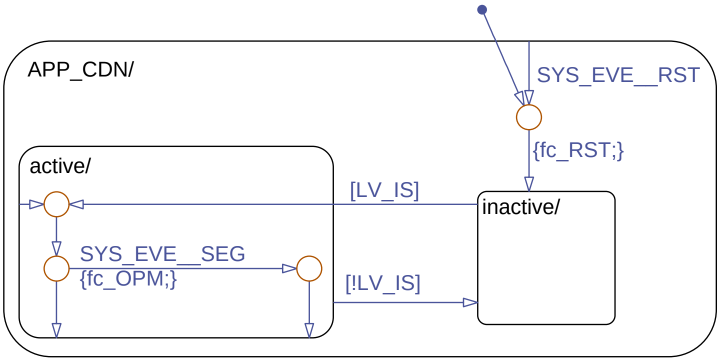
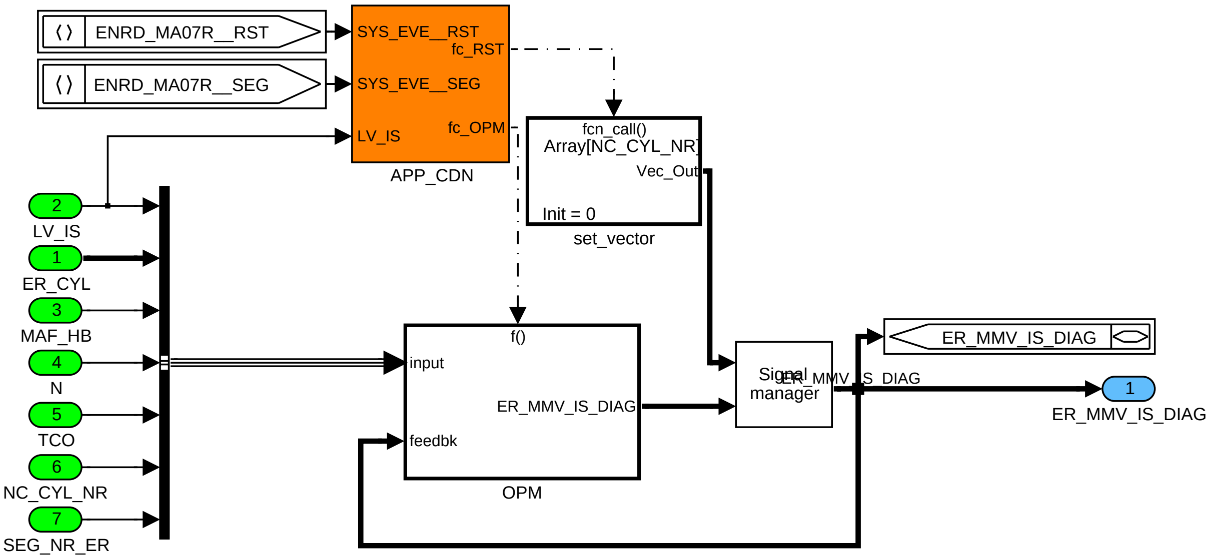


Figure 1 ENRD_MA07R/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 14−Dec−2005

Figure 2 ENRD_MA07R

21.8.1.1 Formula section

Calculation of filtered engine roughness value

ER_MMV_IS_DIAG is calculated here.

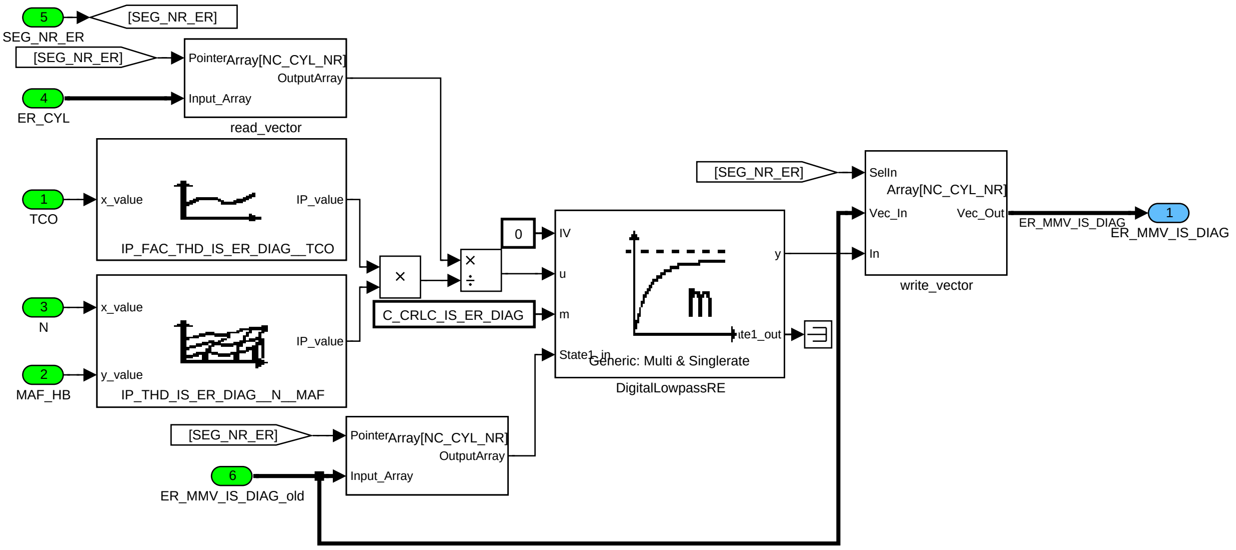


Figure 3 ENRD_MA07R/ OPM/ OPM

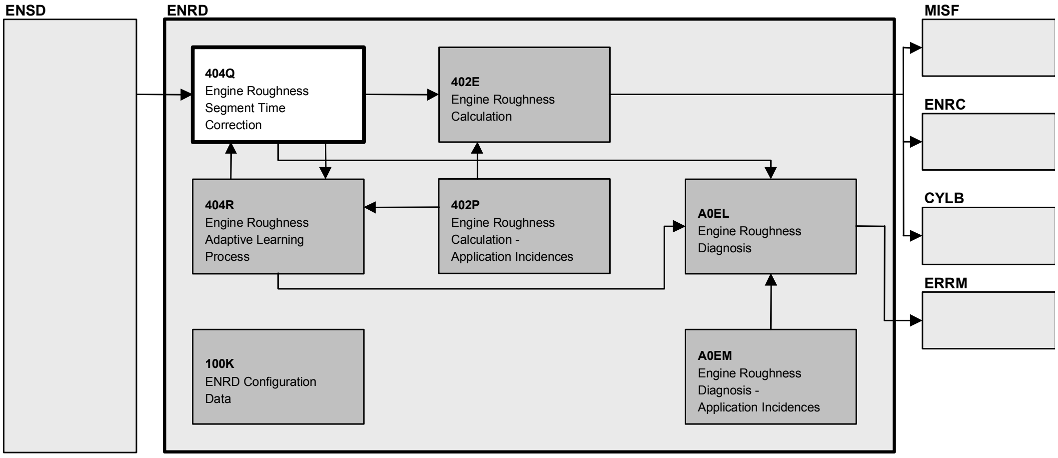
21.9 Engine roughness calculation & adaptation (gen)

21.9.1 Requirements

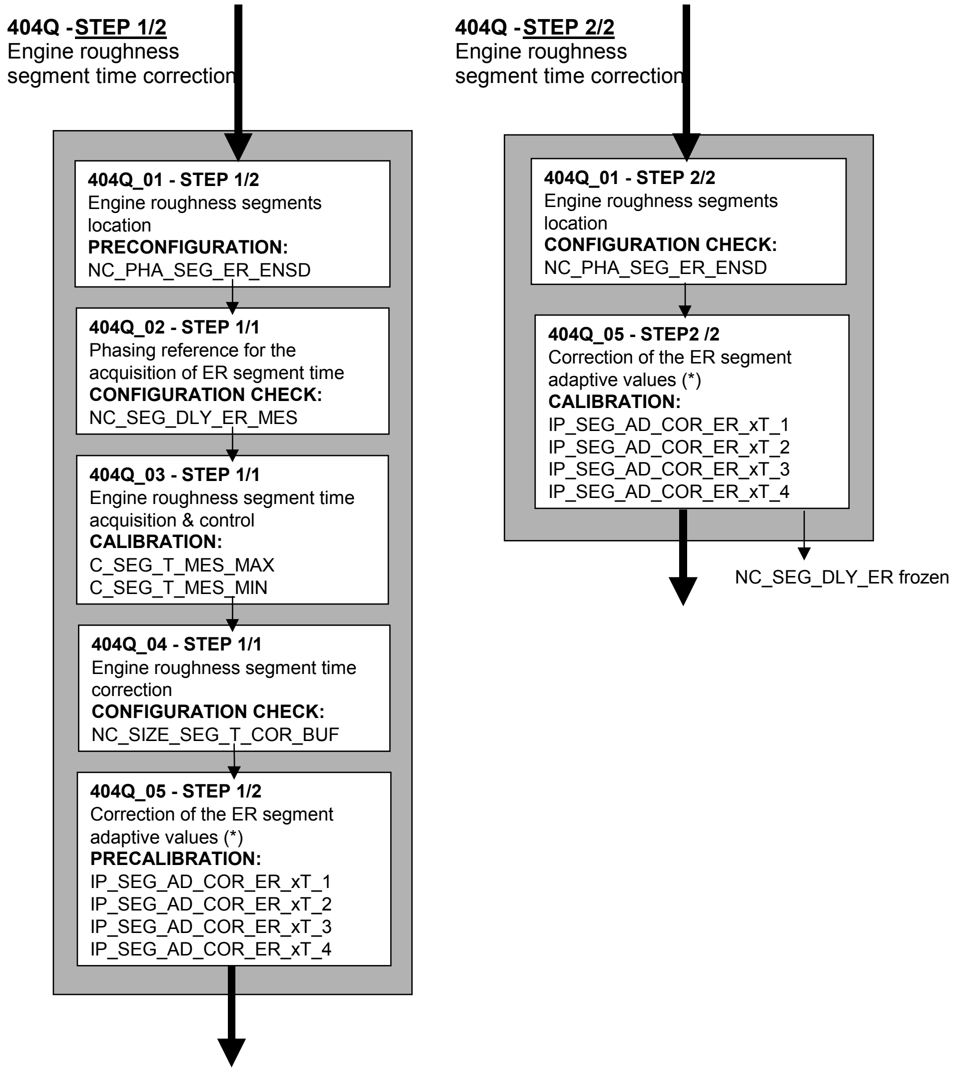
ENSD aggregate active and calibrated, no synchronisation errors/troubles

IGRE, IGSP & INJR precalibrated

21.9.2 Tuning hints for Engine roughness segment time correction (Module 404Q)



21.9.2.1 Module calibration flowchart



21.9.2.2 404Q_01: Engine roughness segments location

21.9.2.2.1 General information

To determinate the engine roughness indexes, crankshaft segment time samples are required. These segment time samples are acquired each (720 / NC_CYL_NR) °crankshaft angle. The ENSD aggregate (engine position and speed determination) provides one T_SEG_ER sample acquisition at each crankshaft segment task.

This segment location is different to T_SEG_CRK *(commonly used in the ECU by other functions)*, as the ER focuses on engine combustion behaviour, and is located regarding to the TDC's, and not to the long tooth position like for T_SEG_CRK.

The engine roughness segment starts NC_PHA_SEG_ER_ENSD°Crk after the TDC. Beginning and end of ER segments are successive and located at the same angle.

This value must be a multiple of flywheel tooth length (e.g. 6°Crk for 60-2 teeth flywheel or 10°Crk for 36-1 teeth flywheel).

Even if the ER segment phasing is defined in crankshaft angle, the ER segment acquisition task has to be imperatively located on the nearest tooth event corresponding to NC_PHA_SEG_ER _ENSD phasing angle.

21.9.2.2.2 Calibration Step 404Q_01 - Step 1/2

Task type: preconfiguration

This calibration step is a configuration choice according the number of cylinder of your application (NC_CYL_NR). These default values have been observed on most of the SV applications.

Please contact basic software person in charge of ENSD implementation on your project to see if there's no incompatibility between the ENSD configuration and these precalibrations.

Important Note

This calibration step is linked to ENSD, the configuration data NC_PHA_SEG_ER_ENSD must be defined in the ENSD configuration data module of the application (module 100D)

Example of configuration: *(default values)*

Engine type NC_CYL_NR	*ER segment length*	*NC_PHA_SEG_ER_ENSD typical value*
3 cylinder	240°Crk	42°Crk
4 cylinder	180°Crk	42°Crk
5 cylinder	144°Crk	36°Crk
6 cylinder	120°Crk	42°Crk
8 cylinder	90°Crk	42°Crk

21.9.2.2.3 Calibration Step 404Q_01 – Step 2 / 2

Task type: configuration check

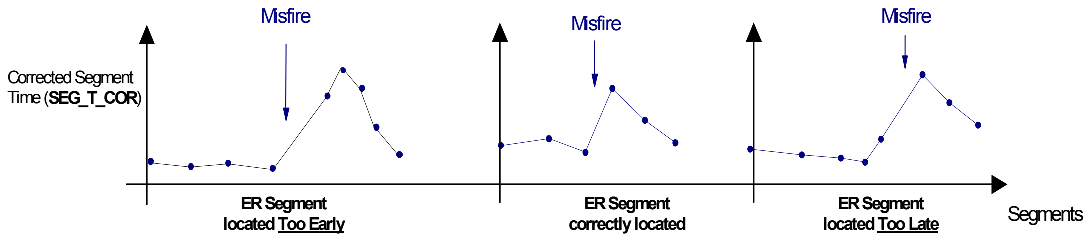
As this calibration/validation step uses the corrected segment time SEG_T_COR, previously all precalibrations/calibrations linked to the modules 404Q & 402E have to be already set.

Validation of Engine Roughness phasing (NC_PHA_SEG_ER_ENSD)

A check of the angular window used to measure the ER segment time should be performed to be sure that the whole engine speed drop due to a misfire is located on the segment chosen.

Remark: pre-defined values fit generally for most application.

Record at several engine speeds and loads (3000 rpm - 50% of full load) with single misfiring.



The acquisition of the segment time is performed every 720/NC_CYL_NR°Crk. We must be sure during the calibration of NC_PHA_SEG_ER_ENSD that the misfiring slope is taken into account between two contiguous acquisitions. This enables to localise the misfiring (number of cylinder for instance).

If the validation shows that the value of NC_PHA_SEG_ER_ENSD is not correct (window too early or too late), a change of NC_PHA_SEG_ER_ENSD is required or the switch to another phasing

Note: If the better NC_PHA_SEG_ER_ENSD phasing you've found exceeds more than 2 teeth of the default value, please inform the domain to check with an external tool (ex: B&S) the basic software results according the teeth number chosen.

21.9.2.2.4 404Q_02: Phasing reference for the acquisition of engine roughness segment time

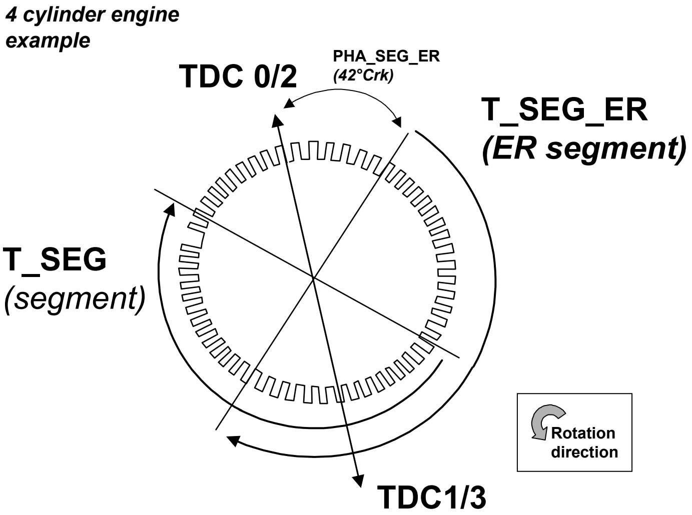
21.9.2.2.4.1 General Information

The ER segments are used to observe in-cylinder combustion effects on crankshaft. The cylinder x combustion is located between cylinder x tdc and cylinder x+1 tdc. Consequently, at x+1 tdc, the T_SEG_ER segment obtained is relative to cylinder X combustion, a delay of 1 tdc between SEG_NR value at cylinder observed (NC_SEG_DLY_ER_MES=1).

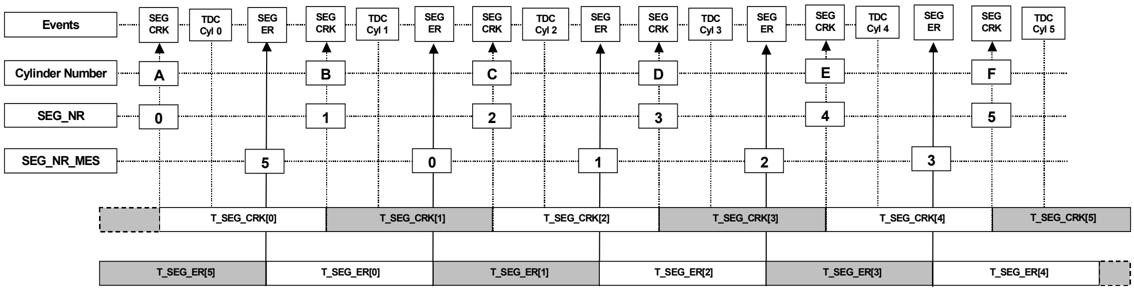
Identification of the segment is evaluated regarding SEG_NR value

Exceptionally, if the segment is located after ER segment, this delay must be cancelled (NC_SEG_DLY_ER_MES=0), because in such case the phasing reference SEG_NR is updated after T_SEG_ER acquisition.

SEG_NR_MES, the ER segments phasing reference, will be used to identify which cylinder is observed with the current engine roughness segment, it's particularly useful for ER segment adaptive values identification.



Implant scheme:



6 Cylinder Engine Example : *NC_SEG_DLY_ER_MES = 1*

21.9.2.2.4.2 Calibration Step 404Q_02 – Step 1 / 1

Task type: Configuration check

Please check, with your project ENSD configuration, the position of the SEG_NR toggle event versus the ER segment location.

21.9.2.2.5 404Q_03: Engine roughness segment time acquisition and control

21.9.2.2.5.1 General information

The rule of this module is to provide to ENRD:

- SEG_T_MES, the ER raw current segment time with correct resolution and format according ENRD calculation core (16 bits / 1µs for ENRD 2.0.0, 32 bits and identical to T_SEG_ER in the future).
- SEG_T_MES_CYL[NC_CYL_NR], the SEG_T_MES buffer, order by cylinder according SEG_NR_MES phasing. This array is used for ER segment adaptive learning process (Module 404R).
- LV_ENA_SEG_T_MES, validity information to share with all ENRD modules.

21.9.2.2.5.2 Calibration Step 404Q_03 – Step 1 / 1

Task type: calibration

This calibration step focuses on validity flag LV_ENA_SEG_T_MES definition.

This flag can be reset to 0, when a fine synchronisation trouble occurs (dedicated information LV_INH_MIS_CRK coming from ENSD), no calibration linked.

This flag can also be set if SEG_T_MES obtained via ENSD exceeds a defined valid range (C_SEG_T_MES_MIN & C_SEG_T_MES_MAX)

To define C_SEG_T_MES_MAX, see NC_N_MIN defined in ENSD (global NC_ data), typical NC_N_MIN = 30 rpm.

To defined C_SEG_T_MES_MIN, see NC_N_MAX defined in ENSD (global NC_ data), typical NC_N_MAX = 8160 rpm

21.9.2.2.6 404Q_04 Engine roughness segment time correction

21.9.2.2.6.1 General information

The prerequisite for a reliable engine roughness evaluation is an accurate ER segment time measurement.

As the crankshaft flywheel teeth are subject to manufacturing tolerances, geometrical run-out and off-centre installation which result in angles modulation, consequently the measured raw segment periods aren't similar and have systematic inaccuracies regarding other segments.

As these inaccuracies are systematic, they can be *"learned"* by an adaptive process during fuel cut-off periods and used for ER segment time correction. This adaptive process is described in the module: Engine roughness adaptive learning process.

The ER segment adaptive values SEG_AD_MMV_ER[NC_CYL_NR] are used to correct the ER raw segment time (obtained via ENSD) and provide to other ENRD modules the ER current corrected segment time: SEG_T_COR

21.9.2.2.6.2 Calibration Step 404Q_04 – Step 1 / 1

Task type: configuration check

SEG_T_COR is stored according a FIFO management (first in, first out). Just check the SEG_T_COR_BUF buffer size regarding engine type:

NC_SIZE_SEG_T_COR_BUF = 11 segment time samples for a 5 cylinder engine

NC_SIZE_SEG_T_COR_BUF = 9 segment time samples for 4 & 8 cylinder engine

NC_SIZE_SEG_T_COR_BUF = 7 segment time samples for 3 & 6 cylinder engine

This size NC_SIZE_SEG_T_COR_BUF is determined by the ER sample number required by ER calculation according engine type.

21.9.2.2.7 404Q_05: Correction of the engine roughness segment adaptive values

21.9.2.2.7.1 General information

On some applications, due to crankshaft flexion/torsion effects, the adaptive values are unsteady versus the engine speed, especially at high engine speed. As a result, these drifts produce additional noise on crankshaft segment times, who will decrease the signal / noise ratio for misfire detection.

Very often we can observe a general trend of these drifts versus the engine speed. To correct these effects, we apply an open loop correction on these adaptive values versus the engine speed in a way to cancel these drifts. A correction can be applied only for each singular segment of the flywheel.

As these corrections are in open loop, they can minimise the additional noises on segments <u>only</u> if the crankshaft segment drifts have identical trend on the overall car type population.

21.9.2.2.7.2 Calibration Step 404Q_05 – Step 1 / 2

<u>Task type: configuration check:</u>

Check that the number of calibration data appears as described in calibration data section (see above) according NC_CYL_NR

<u>Precalibration:</u>

<u>As the</u> Engine roughness adaptive learning process is not already calibrated, and adaptive values drifts (if exists) are not evaluated. Calibrations are initialised to 0.

IP_SEG_AD_COR_ER_xT_1=IP_SEG_AD_COR_ER_xT_2 = 0

IP_SEG_AD_COR_ER_xT_3 = IP_SEG_AD_COR_ER_xT_4 = 0

Note: _xT stands for _MT and _AT

21.9.2.2.7.3 Calibration Step 404Q_05 – Step 2 / 2

<u>Task type: calibration</u>

<u>Preliminary checks before calibration:</u>

To determinate the segment adaptive corrections IP_SEG_AD_COR_ER_xT_y needed for a car family, tests must be executed on all different transmission type, sensor, flywheel and crankshaft designs.

The amplitude of these corrections is the main aim of calibration definition, but as these corrections are open loop corrections, please take care to define average calibrations that can be suitable for the all cars population, respect to the transmission type:

- If we apply too important corrections, engines with small segment drifts will have reduced misfire detection efficiency than without correction.
- If we apply weak corrections, engines with important segment drifts will not reach expected misfire detection efficiency to fulfill OBDII/EOBD legal requirement at high engine speed.

We have to find the calibration data set that allows balancing in the best way between these two objectives to compensate the most important part of production vehicle population.

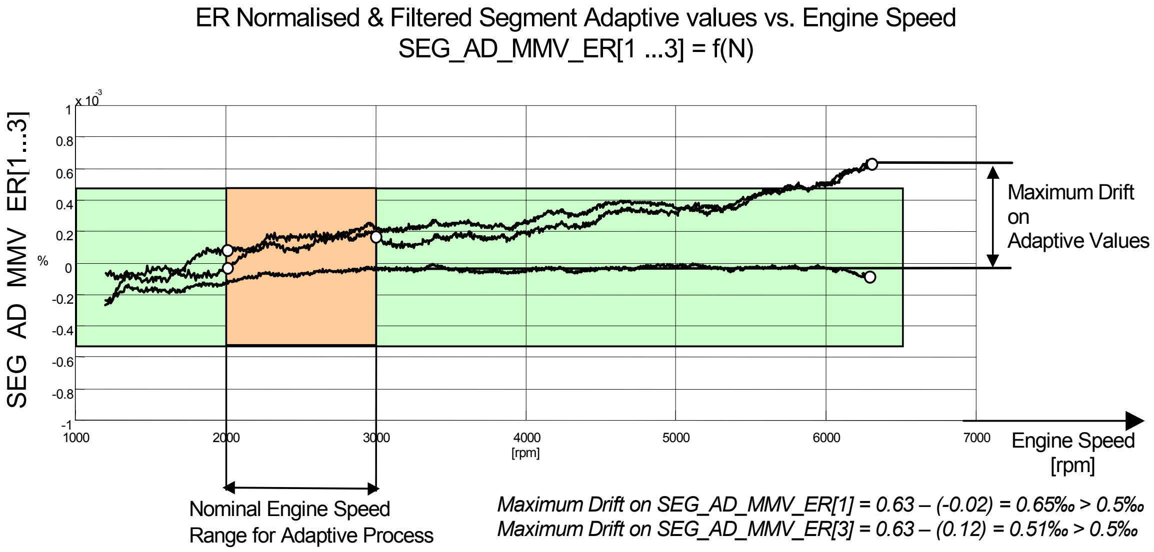
Before to start the calibration of segment adaptive correction, please check calibration data set relative to module 404R_03 (Engine roughness adaptive learning process management), especially C_N_MIN_SEG_AD_ER & C_CRLC_SEG_AD_ER.

Because the drift of segment adaptive values can be amplified (or created) by:

- A too light filtering (lot of noise on the SEG_AD_MMV_ER[NC_CYL_NR] values during the adaptive process) -> a noise capture at the end of the adaptive process could produce a drift.
- An engine speed range for learning where adaptive values are unsteady. Please, focus on the minimum engine speed, because very often the adaptive values are frozen when we go out of the adaptive conditions, and the main condition to exit of the learning process is the minimum engine speed C_N_MIN_SEG_AD_ER. So the last adaptive values learned (close to the minimum engine speed) are next apply to the segment time correction at each engine operating points

Worst operating case: highest engine speed with the adaptive value learned at the minimum engine speed

<u>Example</u> of a 4-cylinder engine with important segment drift at high engine speed.



ER Normalised & Filtered Segment Adaptive values vs. Engine Speed
SEG_AD_MMV_ER[1 ...3] = f(N)

Nominal Engine Speed
Range for Adaptive Process

Maximum Drift on SEG_AD_MMV_ER[1] = 0.63 – (-0.02) = 0.65‰ > 0.5‰
Maximum Drift on SEG_AD_MMV_ER[3] = 0.63 – (0.12) = 0.51‰ > 0.5‰

<u>Calibration Process:</u>

For each car available, execute coast down on chassis rolls in road mode, at least 4 times per car to check the repeatability of the crankshaft behaviour *(recommended: 1 coast down per acquisition file, easiest to process and post-process, add LV_PUC in acquisition file for cut-off information).*

Define calibrations to activate the ER adaptive process on engine speed range 1500 rpm to Engine Redline, extended fuel cut-off activation to this range, with no adaptive process fade-out and light filtering to be able to observe crankshaft behaviour (drift trends).

<u>For example:</u>

C_N_MAX_SEG_AD_ER = C_N_SEG_T_AD_RAW_MAX_ER = 6500 rpm

C_N_MIN_SEG_AD_ER = C_N_SEG_T_AD_RAW_MIN_ER = 1500 rpm

C_CRK_DIF_MAX_ER = C_DRV1_MAX_SEG_AD_ER = max values

C_NR_SEG_AD_ACT_MIN_ER = C_NR_SEG_DRV1_MIN_ER = 0

C_CRLC_SEG_AD_ER = C_CRLC_FAST_SEG_AD_ER = 0.1

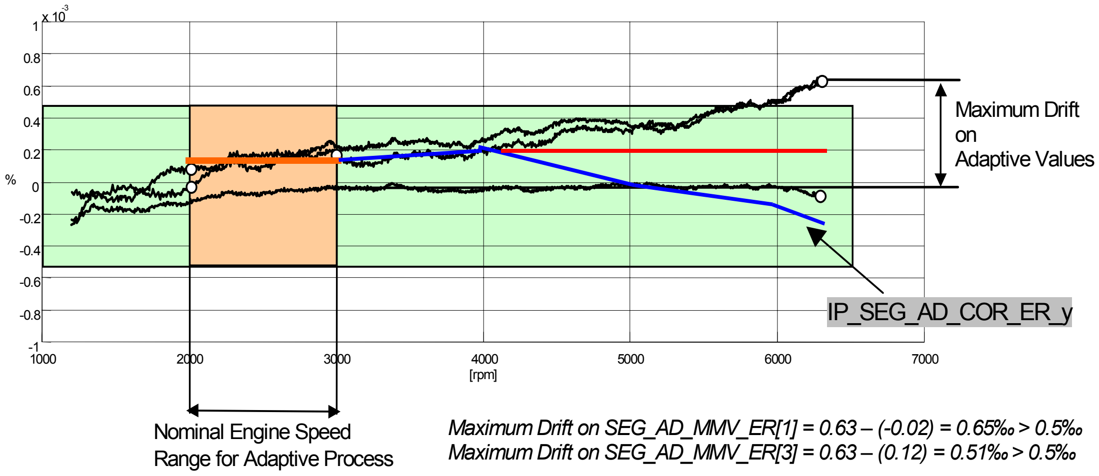
Check calibrations of 404R_01 module to obtain LV_SEG_AD_FDOUT_ER = 0 in the required engine speed range.

For each acquisition file obtained, observe the drift (trend and amplitude) for each segment between

1. Mean value of segment adaptive values on nominal engine speed range for adaptive process. In this example mean(SEG_AD_MMV_ER[1] & SEG_AD_MMV_ER[3]) = 0.17‰
2. Segment adaptive values at highest engine speed. In this example @ 6350 rpm SEG_AD_MMV_ER[1] = SEG_AD_MMV_ER[3] = 0.63‰

Compose the relative trend of segment adaptive trend as successive line sections to be able to approximate with 1D interpolation table. As IP_SEG_AD_COR_ER_xT_y corrections use multiple engine speed breakpoints LDPM_N_32_1_ENRD, set same origin for each line sections. To compensate the drifts, trend must be applied with opposite sign than the adaptive values observed.

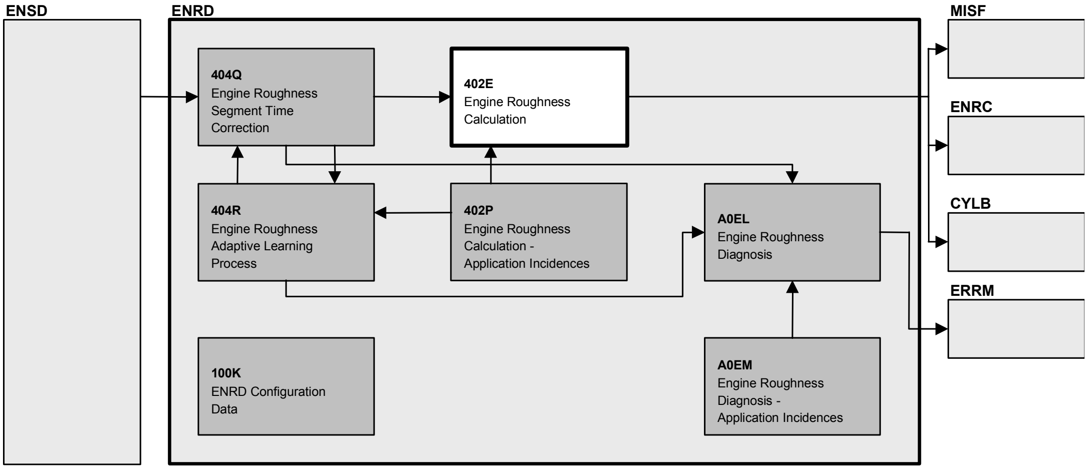
ER Segment Adaptive values vs. Engine Speed & Segment drift trends
SEG_AD_MMV_ER[1 ...3] = f(N)



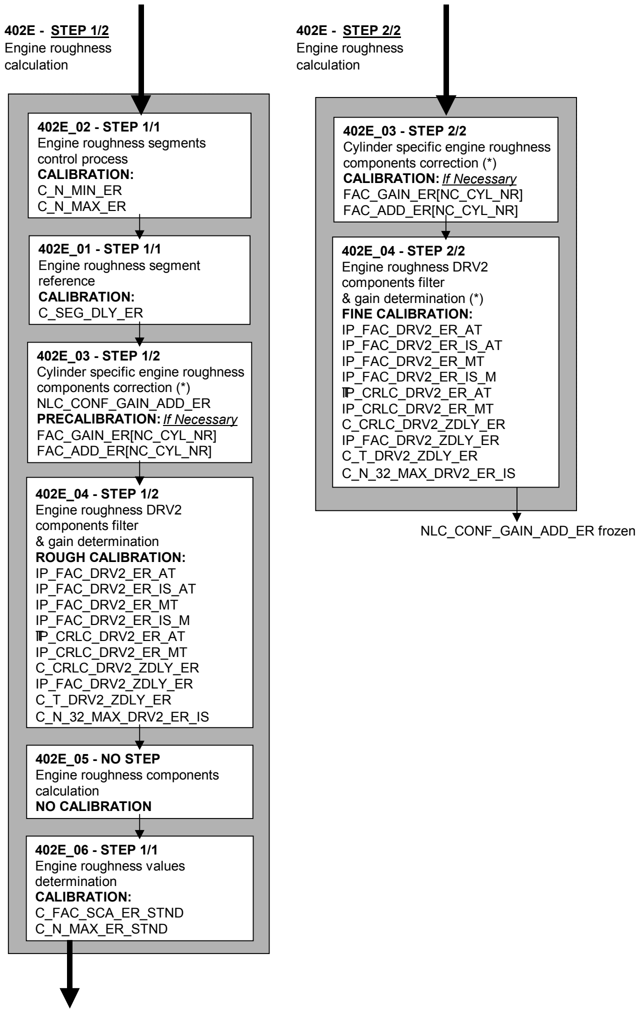
Maximum Drift on SEG_AD_MMV_ER[1] = 0.63 – (-0.02) = 0.65‰ > 0.5‰
Maximum Drift on SEG_AD_MMV_ER[3] = 0.63 – (0.12) = 0.51‰ > 0.5‰

Once all cars available are tested and evaluated, define the mean value per segment and per transmission type that could correct in the best way the most important sample of your car population.

21.9.3 Tuning hints for Engine roughness calculation (Module 402E)



21.9.3.1 Module calibration flowchart



21.9.3.2 Module calibration prerequisites

If NLC_CONF_GAIN_ADD_ER = 1, LOAD_MIS must be already calibrated for breakpoints calibration, see associated MISF calibration process.

21.9.3.3 402E_01: Engine roughness segment reference

21.9.3.3.1 General Information

Due to engine roughness definition, there is a delay of C_SEG_DLY_ER segments between ER actual value and the segment that was observed by this ER value.

SEG_NR_ER will be used to identify which cylinder is observed with the engine roughness system variable, it's particularly useful for misfire cylinder identification.

The segment counter SEG_NR_ER with the limit 0...(NC_CYL_NR - 1) is incremented in the same way than the common EMS segment counter SEG_NR.

21.9.3.3.2 Calibration Step 402E_01 – Step 1 / 1

Task type: calibration

To check correct cylinder identification via SEG_NR_ER segment counter, you must observe ER_RAW or ER regarding to SEG_NR_ER values.

Make an acquisition of ER_RAW, SEG_NR & SEG_NR_ER values, create a cylinder misfire sequence with C_TI_AS[0...NC_CYL_NR] = 1 then 0 then 1.

For a proper identification, in your acquisition file when you've produced C_TI_AS_0 = 0, the ER values must decrease as a misfire each NC_CYL_NR and on these TDC's, SEG_NR_ER must be equal to 0.

For C_TI_AS_1 = 0, the ER values must decrease as a misfire each NC_CYL_NR TDC's and on these TDC's, SEG_NR_ER must be equal to 1. Check in the same way all the other cylinders.

If C_TI_AS_[NC_CYL_NR] are not integrated on your project, unplug the correct cylinder with breaking box. Beware of equivalence between software and physical cylinder numbers.

If the segment counter SEG_NR_ER identify another cylinder, then correct the C_SEG_DLY_ER calibration data according the SEG_NR value you observe when you create a misfire.

Default calibration

NC_CYL_NR	C_SEG_DLY_ER
3	4
4	5
5	6
6	4
8	5

21.9.3.4 402E_02: Engine roughness segments control process

21.9.3.4.1 General Information

All ER components calculations are based on corrected segment time acquisitions buffer values SEG_T_COR_BUF.

As soon as an invalid segment has been detected during acquisition process (LV_ENA_SEG_T_MES = 0) or segment adaptive calculation, ER calculation & misfire detection have to be disabled (LV_ENA_ER = 0).

Engine roughness validity is also set according a valid engine speed range defined between C_N_MIN_ER & C_N_MAX_ER

21.9.3.4.2 Calibration Step 402E_02 – Step 1 / 1

The minimum engine speed for engine roughness calibration has to be lower than the requested minimum engine speed for misfire monitoring (N_IS_SP – 150 rpm for US applications). The Idle speed setpoint for hot engine minus 200 rpm is recommended (e.g. idle speed = 700 rpm; set C_N_MIN_ER = 500 rpm).

The maximum engine speed for engine roughness calibration has to be higher than the requested maximum engine speed for misfire monitoring (Fuel cut-off engine speed for US application, 4500 rpm for Europe). E.g. for C_N_MAX_ER = 4600 rpm for an EC application.

21.9.3.5 402E_04: Engine roughness DRV2 components filter & gain determination

21.9.3.5.1 General Information

This function allows to determinate the calibration coefficients relative to DRV2 components of the engine roughness, according different configurations:

If used *(see project/customer legal requirements concerning misfire monitoring after engine start)*, during zero delay misfire monitoring activation period, specific gain & filter coefficients can be apply to DRV2 component to be able to detect misfire just after engine start *(dynamic correction modified). This specific correction can be applied during a calibration time* C_T_ZDLY_MIS - C_T_DRV2_ZDLY_ER.

Else, FAC_DRV2_ER & CRLC_DRV2_ER are defined according transmission type *(AT/MT)*.

The misfire detection criteria ER must oscillate around 0 when there is no misfiring and have a negative value, representing the negative torque applied on the crankshaft, when there is misfiring.

21.9.3.5.2 Calibration Step 402E_04 – Step 1 / 2 – Rough Calibration

Example of calibration:

FAC_DRV2_ER = IP_FAC_DRV2_ER_xT = IP_FAC_DRV2_ER_IS_xT = 1.2 on the complete maps

IP_CRLC_DRV2_ER_xT = 0.4 on the complete map

C_CRLC_DRV2_ZDLY_ER = 0.95

IP_FAC_DRV2_ZDLY_ER = 0.4 on the complete map

C_T_DRV2_ZDLY_ER = 3.5s (with C_T_ZDLY_MIS= 5s, DRV2_ZDLY is 1.5s effective after engine start)

C_N_32_MAX_DRV2_ER_IS = 1700 rpm

Note: _xT stands for MT if LV_AT = 0 and stands for AT if LV_AT = 1

Calibration of FAC_DRV2_ER and CRLC_DRV2_ER in part load & idle speed mode

Target of the calibration:

Optimise IP_FAC_DRV2_ER_xT, IP_FAC_DRV2_ER_IS_xT, IP_FAC_DRV2_ZDLY_ER and IP_CRLC_DRV2_ER_xT in order that the oscillations generated after a misfiring or naturally by the engine are minimized or cancelled.

For that, it is aimed to oppose DRV2_MMV_ER to ER_RAW (DRV0_ER - DRV1_ER) when this later is negative to get finally ER near to 0.

The values IP_FAC_DRV2_ER_xT, IP_FAC_DRV2_ER_IS_xT are mainly responsible for the amplitude correction and it is nearly proportional to the amplitude of the crankshaft speed fluctuation. The biggest the value of FAC_DRV2_ER is, the biggest the correction value will be.

CRLC_DRV2_ER is mainly responsible for the correction fitting to the frequency of the speed fluctuation. The biggest the value is, the fastest but also the shortest the correction!

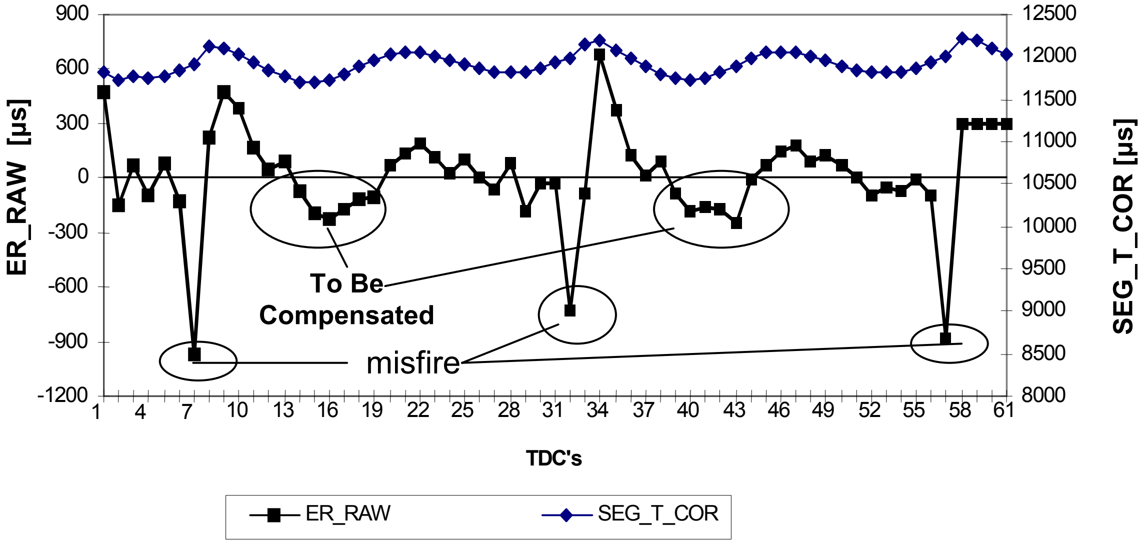
The term DRV2_ER enables the correction of ER_RAW; but the correction is generally performed too early. That explains the presence of the filtering value (CRLC_DRV2_ER) in order to adjust the correction position to fit with the real need. But the action of the filtering value reduces also the amplitude of the correction that is why the factor (FAC_DRV2_ER) is used. Therefore, the factor is generally greater than 1.

Adjust the filtering values IP_CRLC_DRV2_ER_xT in order that DRV2_MMV_ER is symmetrical to ER_RAW during oscillations after single misfiring by making the filter to “trail“ more or less. For that IP_FAC_DRV2_ER_xT & IP_FAC_DRV2_ER_IS_xT must be set to 1 or 1, 2.

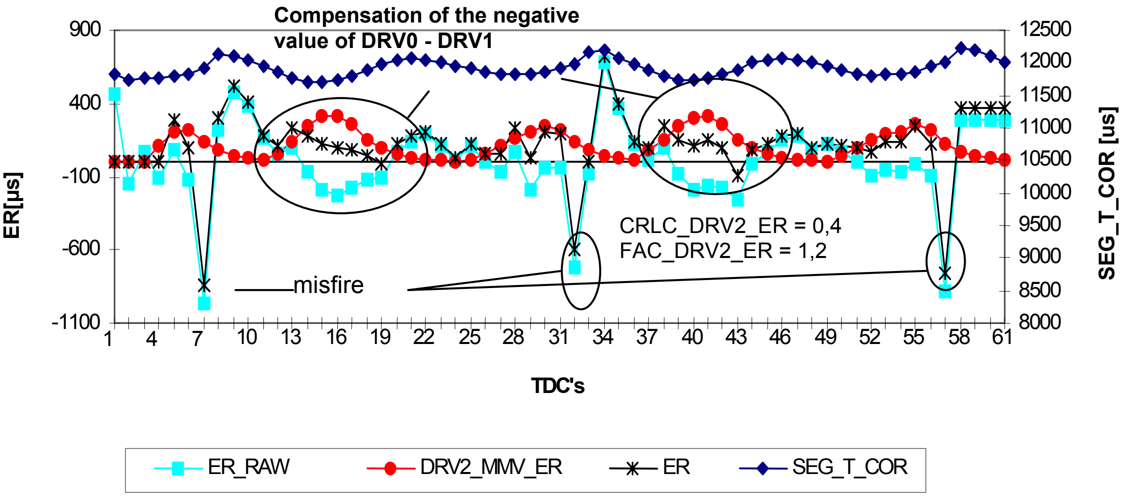
Adjust then the factor FAC_DRV2_ER in order that ER is always positive or null during oscillations. But take care to not reduce too much the value of ER at misfiring event by a too big value of DRV2_MMV_ER at this moment.

The following pictures show some examples of different calibration values (single misfire events with crankshaft oscillations afterwards).

When ER_RAW is observed during oscillations after one misfire event, we can see that ER_RAW had negative values. These negative values (due to oscillations) have to be compensated by the term DRV2.



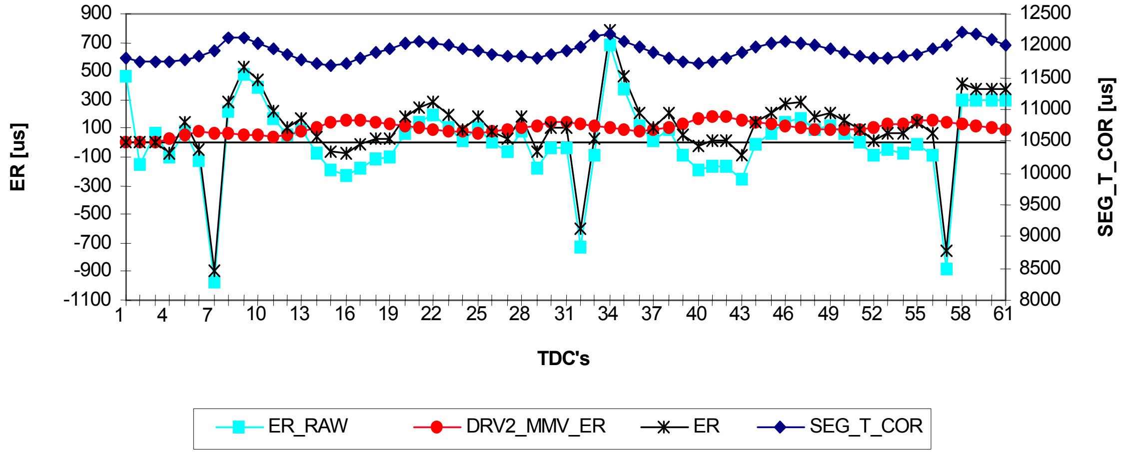
When CRLC_DRV2_ER and FAC_DRV2_ER are correctly calibrated, ER_RAW *(DRV0_ER - DRV1_ER)* should be compensated by DRV2_MMV_ER and ER remains positive or near 0:



Influence of CRLC_DRV2_ER, FAC_DRV2_ER is set at 1,2.

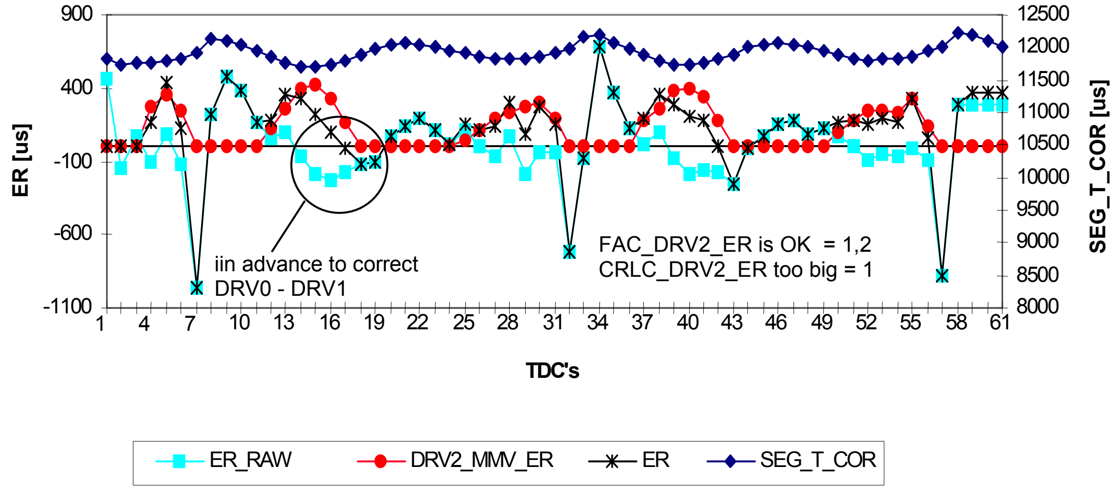
When CRLC_DRV2_ER is too small, DRV2_MMV_ER is delayed regarding ER_RAW and it compensates ER badly or decreases the final value of ER at time where misfire occurs:

⇒ Too slow response time



When CRLC_DRV2_ER is too big, DRV2_MMV_ER is in advance regarding ER_RAW and it compensates badly ER values:

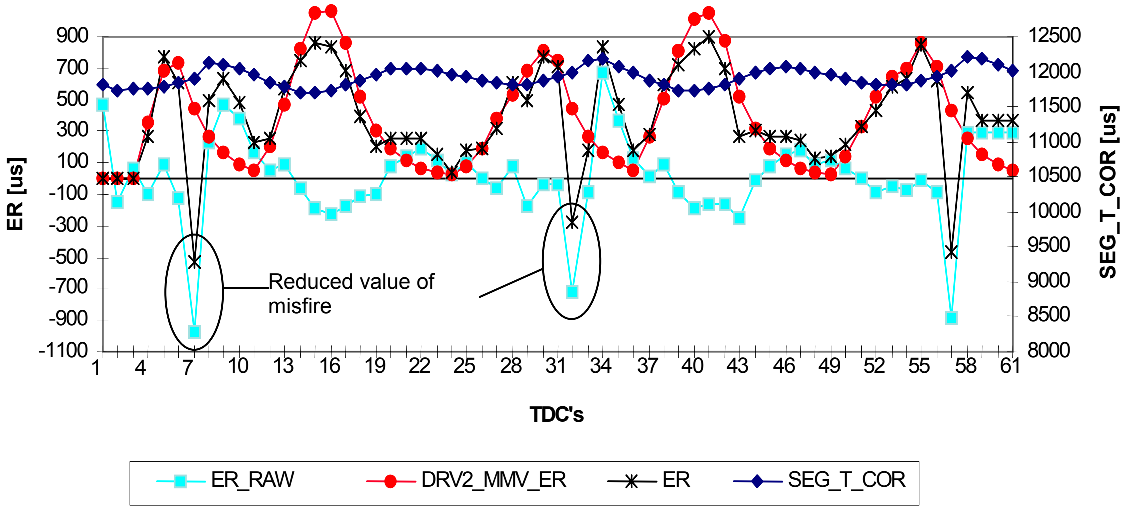
⇒Too fast response time



Influence of FAC_DRV2_ER, CRLC_DRV2_ER is set to 0,4.

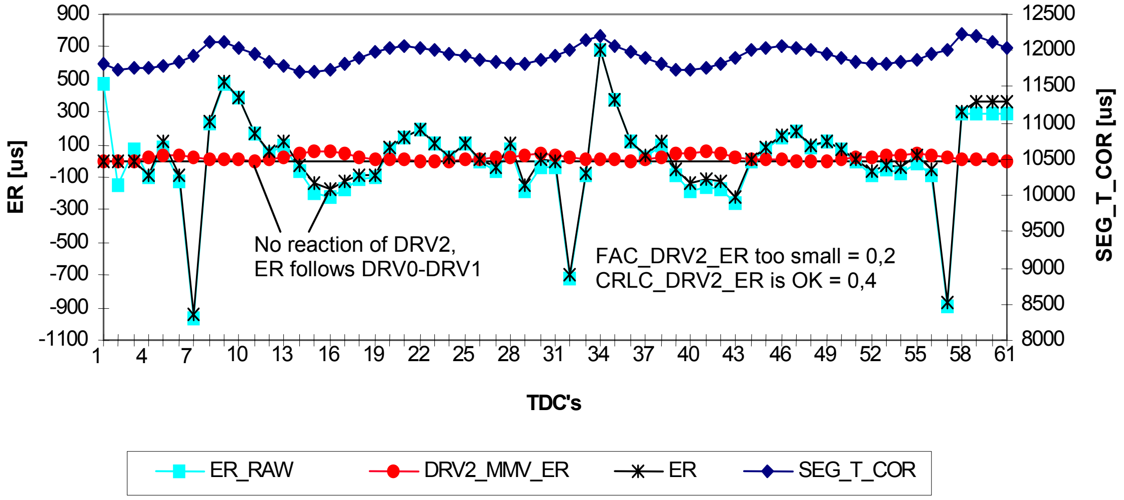
When FAC_DRV2_ER is too big, ER goes further from 0 and the value of ER is reduced at the time where a misfire event occurs, that can cause misdetection.

⇒ Too big correction ⇒ no misfire detection possible



When FAC_DRV2_ER is too small, ER can become negative during an oscillation and induce over-detection:

⇒ Too small correction ⇒ over-detection of misfire possible



The basic calibration should be performed on the chassis-dyno, and then a check on the street (several types of streets) is necessary. Make recordings at several gear ratios, several engine loads, several engine speeds and several misfiring rates

Record the values of SEG_T_COR, DRV0_ER, DRV1_ER, DRV2_MMV_ER, ER_RAW & ER.

N = breakpoints of the table

Load: limit of the positive torque and step of 100 mg/stk. to full load.

Gear ratios: 1st, 3rd, and 5th

Misfiring rate: 50%, 25%, 3% (for NC_CYL_NR = 4)

Then look for the best compromise between FAC_DRV2_ER and CRLC_DRV2_ER for each engine speed, engine load and transmission type.

If a specific DRV2_MMV_ER behaviour is necessary to compensate important oscillations due to idle speed controller at low engine speed, IP_FAC_DRV2_ER_IS_xT calibration maps are dedicated to such corrections.

C_N_32_MAX_DRV2_ZDLY_ER is the maximum authorised engine speed to apply these idle calibrations, to be set according the highest idle engine speed you require for this area.

21.9.3.6 Calibration Step 402E_04 – Step 2 / 2 – Fine Calibration

This calibration step occurs during (or after) a first misfire detection calibration phase (phase dedicated to misfire detection thresholds determination).

Calibration of FAC_DRV2_ER and CRLC_DRV2_ER in nominal mode

Check that no wrong misfire detections or non-misfire detections are induced due to inappropriate nominal DRV2 calibrations. For that, check ER index behaviour during misfire thresholds determination phase, check the difference between ER_RAW and ER in case of misfire, if too important, it means that you decrease too much the signal noise ratio in such case.

Calibration of FAC_DRV2_ER, CRLC_DRV2_ER & C_T_DRV2_ZDLY_ER in engine start mode (ZDLY)

For US applications it’s now mandatory to monitor misfire one engine cycle after engine start (ZDLY period) (legal definition of engine start = when engine speed reaches hot idle speed setpoint – 150 rpm). If we apply the nominal DRV2 components at engine start, this monitoring will be impossible due to a strong correction of the ER values during the engine cranking (high possible ER values).

Calibration target of DRV2 components in ZDLY:

The Goal of this calibration step is to centre ER values around at engine start and be able to detect misfire during this period.

For this, light coefficients are required for DRV2_ZDLY amount & filtering, see precalibrations.

Execute several engine crankings with and without misfire generation *(misfiring rate: 50%, 25%, 3% for NC_CYL_NR = 4)* and observe ER trend during first TDC's. Adjust FAC_DRV2_ER amount with IP_FAC_DRV2_ZDLY_ER calibration to have ER values centred on 0.

If a too important positive overshoot appears on ER (DRV2 correction too important), the misfire detection will not be possible.

If a too important negative overshoot appears on ER (DRV2 correction too little), there's a risk of wrong misfire detection during engine cranking.

Observe the mean time for engine stabilisation after cranking to switch to nominal engine running condition, this mean time will determinate the time to switch from ZDLY to nominal DRV2 calibrations.

Perform different engine crankings with different engine coolant temperatures (TCO) to be able to compensate the ER rise-up in any case during the zero delay phases, especially with engine cold start.

Precalibrations:

C_T_DRV2_ZDLY_ER = 3s, specific DRV2 correction applied only on first 2 seconds

C_CRLC_DRV2_ZDLY_ER = 0.94

IP_FAC_DRV2_ZDLY_ER = 0.4 *all map cells*

21.9.3.7 402E_03: Cylinder specific engine roughness components correction

21.9.3.7.1 General information:

The misfire detection based on ER index can be complicated by the torsion/flexion vibrations especially with multiple cylinder engines with a long crankshaft. This affects on misfire detection efficiency, especially at high engine speed.

To balance this phenomenon, after segment timing correction with ER adaptive values *(see ER segment correction process)*, an irregular operation is determined from the measured cylinder segment times, cylinder specific additive and multiplicative corrections are applied on the raw engine roughness value *(ER_RAW)*.

FAC_ADD_ER[NC_CYL_NR] and FAC_GAIN_ER[NC_CYL_NR] correction coefficients are based on calibration tables specific to each cylinder and to transmission type (MT/AT).

The measurement effect obtained by evaluating the crankshaft speed for the misfire detection is especially low in high engine speeds. In addition interferences can occur in this area such as mechanical tolerances of the target wheel and torsion vibrations of the crankshaft. This can eventually lead to, that in higher engine speeds; the conventional proceedings of the detection security can no longer be guaranteed for all cylinders.

After a correction of the mechanical tooth error an irregular operation is determined from the measured cylinder segment times. The interference caused by torsion vibrations is suppressed by adding a cylinder selective interference value, which is dependent on the load and engine speed areas of the operating point. The added interference variables are then determined on the engine test bench and put into cylinder individual maps.

The torsion depends on the drive train concept especially crankshaft and flywheel. If a change should occur (i.e. changed to a dual mass flywheel) then the calibration must be checked and if necessary, carried out again.

Function available only if NLC_CONF_GAIN_ADD_ER = 1 in your ENRD configuration date file.

21.9.3.7.2 Calibration Step 402E_03 – Step 1 / 2 – Precalibration

All values in the maps IP_FAC_ADD_ER_xT[NC_CYL_NR] are filled with zeros, all values for IP_FAC_GAIN_ER_xT[NC_CYL_NR] are filled with ones. Therefore no torsion correction is effective. This adjustment is valid during the whole measurement recording of the IP_FAC_ADD_ER_xT & IP_FAC_GAIN_ER_xT calibration.

Note: IP_FAC_ADD_ER_xT stands for IP_FAC_ADD_ER_AT for automatic transmissions & for IP_FAC_ADD_ER_MT for manual transmissions

21.9.3.7.3 Calibration Step 402E_03 – Step 2 / 2 – Calibration

Principle:

Measurement of the cylinder engine roughness values with the stationary operating points, engine speed and load over at least 500 working cycles.

No appearance of misfire.

Calculation of the ER-mean value for every cylinder.

Position of the engine speed: 3 operating points at low, middle and high loads (i.e. 33%, 55%, 85%) with a fine resolution in engine speeds (from 5000 rpm in 100 rpm intervals) beginning at 3000 rpm.

For the maps IP_FAC_GAIN_ER_xT [NC_CYL_NR] and IP_FAC_ADD_ER_xT [NC_CYL_NR], each cylinder can adjust its associated LDP table to fix the most judicious engine speeds.

Example:

To make explanations the clearest possible, LDP tables are the same for all the cylinders. However, once your LDP table frozen for one cylinder, you can completely change the position of the engine speed for another cylinder.

N [rpm]	2900	3008	3488	4000	4704	5696	6016	6240

Values in breakpoints: The value 0 is entered for masking all maps IP_FAC_ADD_ER_xT[NC_CYL_NR], for all loads in the lowest engine speed breakpoint and for the remaining breakpoints, measurements under stationary conditions and calculations of the cylinder individual ER-mean values.

Example:

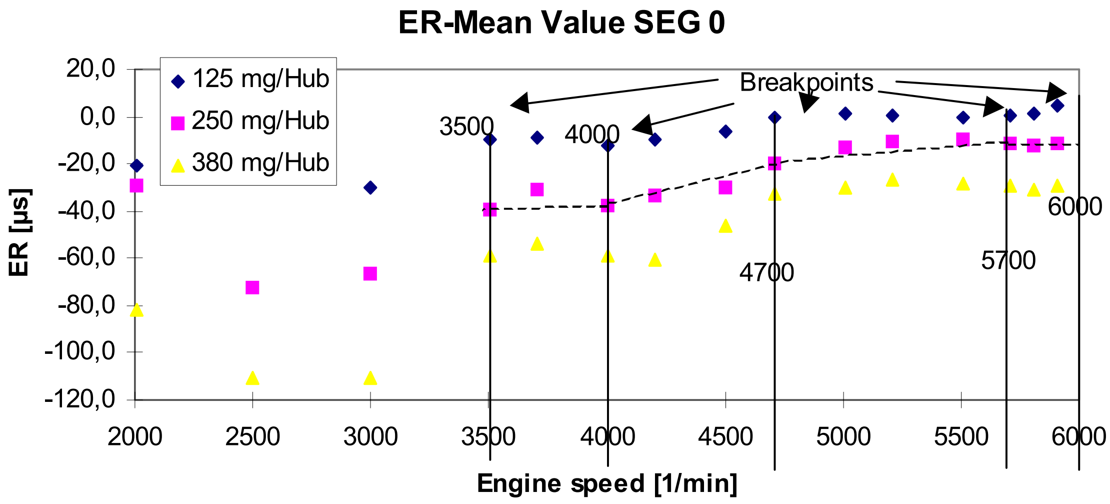
Measurements from over 500 firing cycles resulted in the following ER-mean values in µs for the single cylinder at 6000 rpm and 200 mg/stk:

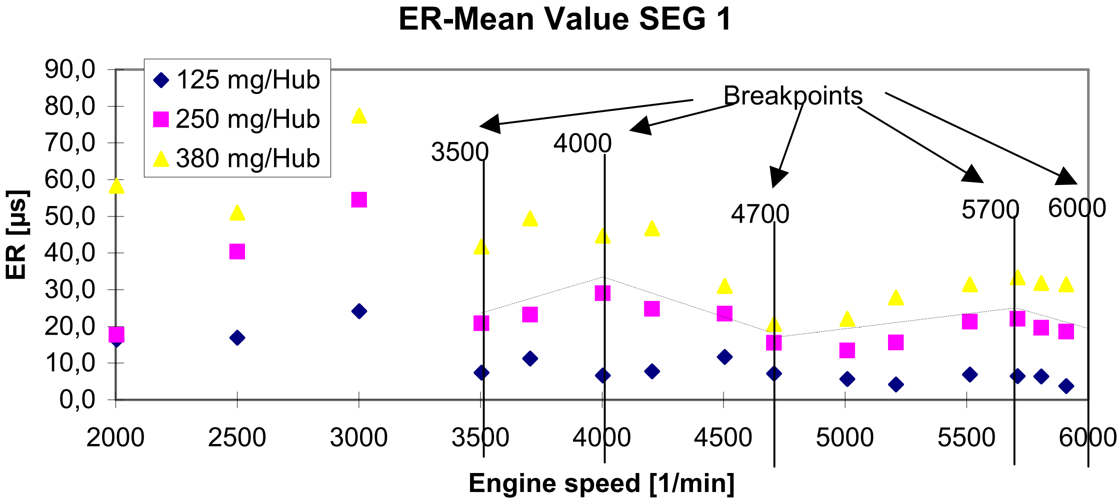
Cylinder 0: -1.3 Cylinder 1: -16.1 Cylinder 2: -8.6

Cylinder 3: 6.3 Cylinder 4: 2.6 Cylinder 5: 17.2.

The following values were entered on the 6000 rpm and 200 mg/stk breakpoints in the IP_FAC_ADD_ER_xT[NC_CYL_NR] maps:

IP_FAC_ADD_ER_xT[0] : -1.3 IP_FAC_ADD_ER_xT[1] : -16.1
IP_FAC_ADD_ER_xT[2] : -8.6 IP_FAC_ADD_ER_xT[3] : 6.3
IP_FAC_ADD_ER_xT[4] : 2.6 IP_FAC_ADD_ER_xT[5] : 17.2.





Graph: Position of the breakpoints with engine speed at [3500, 4000, 4700, 5700, 6000]. (Only two segments observed)

Calibration of the gain factor for torsion correction IP_FAC_GAIN_ER_xT[NC_CYL_NR]

In case of misfire, the factors in the cylinder individual maps IP_FAC_GAIN_ER_xT[0] ... IP_FAC_GAIN_ER_xT[5] should level out the collapse of the ER-values for the various cylinders. The relevant map is selected by means of segment number for the ER-value calculation SEG of the corresponding cylinders just as with the IP_FAC_ADD_ER_xT[NC_CYL_NR] maps. All IP_FAC_GAIN_ER_xT[NC_CYL_NR] maps have common engine speed breakpoints.

Dead application by filling with ones.
The factor value range should lie between 1 and approx. 1.8.

Precalibration
Maps IP_FAC_ADD_ER_xT[NC_CYL_NR] are completely calibrated, all values IP_FAC_GAIN_ER_xT[NC_CYL_NR] are filled with one.
This adjustment is valid during the whole measurement recording of the IP_FAC_GAIN_ER_xT[NC_CYL_NR] calibration.

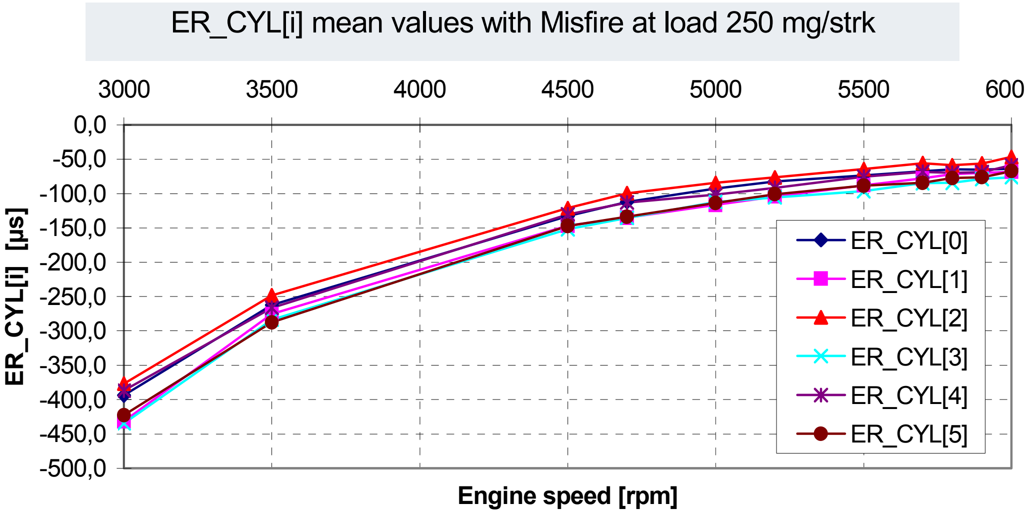
Calibration process
Principle: Measuring of the ER cylinder individual with steady operating points engine speed and load with generation of continuous misfire of cylinders. Various cylinders show different undershoots. Verification of individual misfiring of the same cylinder of which through sufficient intervals (e.g. every 4 firing cycles) influence each other.

1. Execute measurement file like describe above with at least 500 firing cycles for every misfiring cylinder *(NC_CYL_NR measurement files per operating point)*. Engine speed starting at 3000 rpm with a 500 rpm interval, starting at 5000 rpm with a 100 rpm interval *(focus on high engine speed)*.
2. Compute the ER mean values, ER_CYL[0] ... ER_CYL[5] only based on tdc's with cylinders in misfire for each operating point. *If Software misfire generator used, you can observe easily the misfire generated with flag LV_MIS_GEN.*
3. Choose the cylinder x with the highest absolute ER_CYL[x] value for overall operating points (here for example: cyl 3). Develop the ER_CYL[3] / ER_CYL[x] ratio for x=0...5 on every operating point. They are in accordance to the factors for the IP_FAC_GAIN_ER_xT[NC_CYL_NR] maps.
4. Check if the ratio ER_CYL[3] / ER_CYL[x] differs from the various load points. Set the engine speed breakpoints so that the linear regression produces a good approximation. Use of lowest breakpoint (suggestion: 3000 rpm) with one for masking.
5. Assignment of the cylinder x to the corresponding IP_FAC_GAIN_ER_xT[x] map

Example: (6cyl engine)

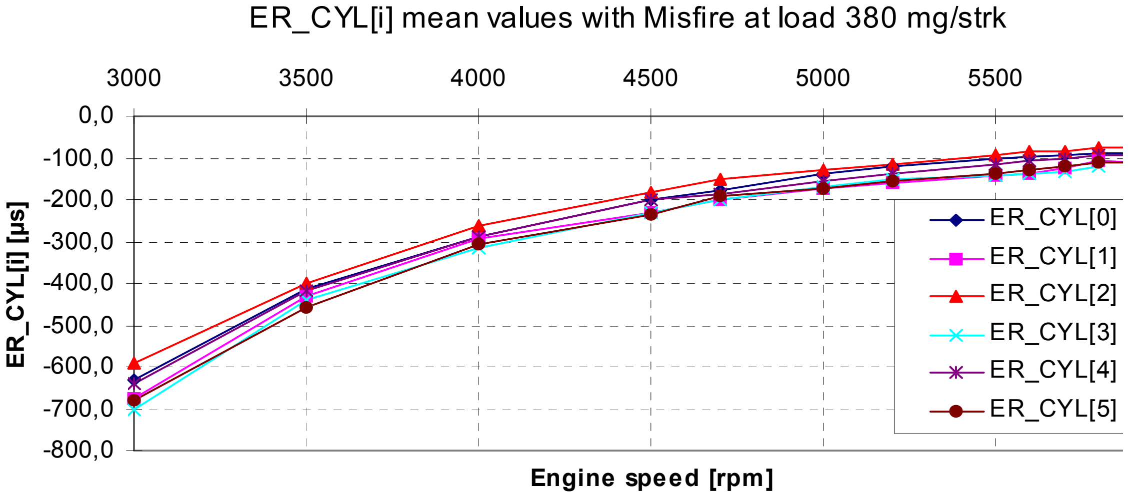
1. The measurements at 250 mg/stk produced the following mean values ER_CYL[x]

	ER_CYL[0]	**ER_CYL[1]**	**ER_CYL[2]**	**ER_CYL[3]**	**ER_CYL[4]**	**ER_CYL[5]**
3000	*-393.8*	*-431.3*	*-377.1*	*-434.7*	*-386.9*	*-422.8*
3500	*-262.4*	*-275.6*	*-248.1*	*-283.8*	*-266.3*	*-287.5*
4500	*-133.2*	*-146.7*	*-121.4*	*-152.2*	*-130.6*	*-147.1*
4700	*-112.2*	*-135.2*	*-99.8*	*-135.6*	*-113.1*	*-133.9*
5000	*-92.6*	*-116.6*	*-84.4*	*-112.0*	*-101.7*	*-114.1*
5200	*-82.2*	*-104.5*	*-76.8*	*-105.8*	*-91.8*	*-101.0*
5500	*-73.7*	*-88.1*	*-64.7*	*-96.8*	*-75.4*	*-88.9*
5700	*-67.8*	*-77.9*	*-55.9*	*-84.8*	*-68.3*	*-84.4*
5800	*-64.9*	*-72.0*	*-58.8*	*-84.5*	*-69.9*	*-77.4*
5900	*-65.2*	*-70.4*	*-56.1*	*-79.0*	*-68.7*	*-76.5*
6000	*-64.0*	*-68.6*	*-46.8*	*-76.0*	*-59.1*	*-67.4*



2. The measurement at 380 mg/stk produced the following mean values ER_CYL[x]

	ER_CYL[0]	ER_CYL[1]	ER_CYL[2]	ER_CYL[3]	ER_CYL[4]	ER_CYL[5]
3000	*-629.6*	*-677.6*	*-592.7*	*-700.1*	*-638.0*	*-680.4*
3500	*-413.1*	*-431.6*	*-398.6*	*-442.0*	*-417.4*	*-457.9*
4000	*-290.9*	*-293.5*	*-264.4*	*-313.6*	*-286.9*	*-305.9*
4500	*-198.2*	*-230.2*	*-180.7*	*-231.2*	*-200.5*	*-234.2*
4700	*-176.2*	*-200.7*	*-152.8*	*-198.5*	*-185.9*	*-191.7*
5000	*-140.0*	*-175.4*	*-130.9*	*-168.7*	*-154.0*	*-172.0*
5200	*-119.7*	*-162.2*	*-114.6*	*-152.5*	*-138.0*	*-155.2*
5500	*-103.1*	*-140.8*	*-92.5*	*-140.7*	*-115.0*	*-137.7*
5600	*-96.8*	*-137.9*	*-84.9*	*-137.0*	*-105.5*	*-129.3*
5700	*-94.9*	*-125.1*	*-83.4*	*-132.0*	*-103.0*	*-121.5*
5800	*-89.5*	*-108.6*	*-76.0*	*-120.1*	*-94.1*	*-112.6*
5900	*-88.3*	*-110.6*	*-77.4*	*-116.5*	*-94.4*	*-109.6*
6000	*-90.3*	*-99.8*	*-73.8*	*-115.6*	*-96.5*	*-106.4*



3. Cylinder 3 shows the absolute largest undershoot in ER above all observed measurement points

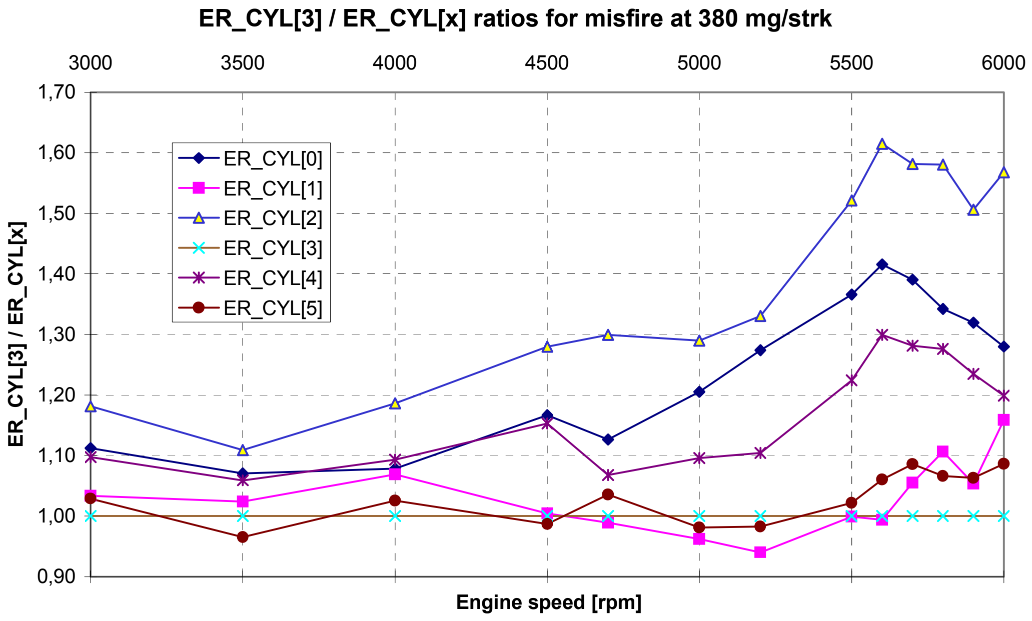
4. Normalisation ER_CYL[3] / ER_CYL[x] for operating point 5500 rpm. 380 mg/stk results

Cylinder 0 : 140.7 / 103.1 = 1.365   Cylinder 1 : 140.7 / 140.8 = 1

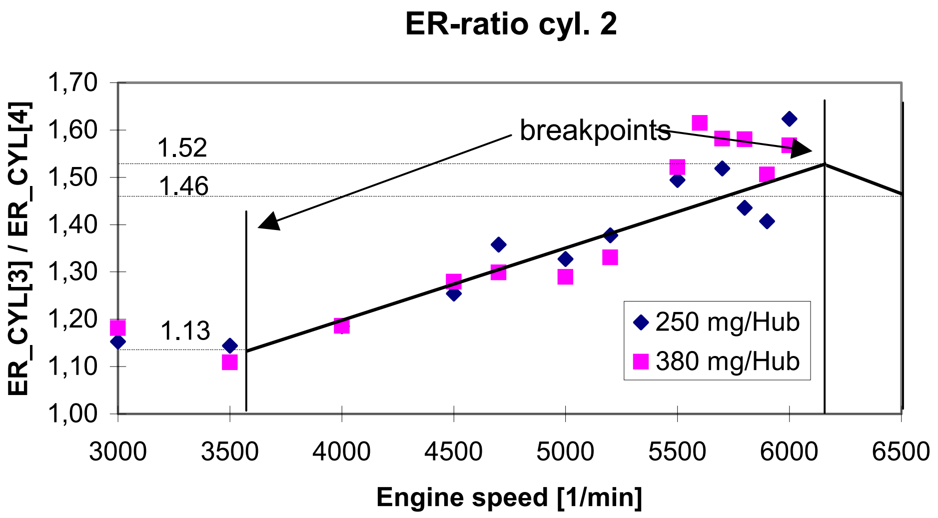
Cylinder 2 : 140.7 / 92.5 = 1.13   Cylinder 3 : 140.7 / 140.7 = 1

Cylinder 4 : 140.7 / 115 = 1.22   Cylinder 5 : 140.7 / 137.7 = 1.02

5. The total results for measurement series at 380 mg/stk :



6. Here only cylinder 2 is observed:



7. Cylinder 2 *(software order, see configuration file for equivalence between software & physical number)* corresponds to the table IP_FAC_GAIN_ER_xT[2]

N [rpm]	3000	3500	5700	6000
IP_FAC_GAIN_ER_*xT*[2]	1	1.13	1.52	1.46

8. Validation comment :

    Calibration on one vehicle per transmission type.

    Validation on at least two cars per transmission type, check that corrections obtained is valid for other cars with the same transmission type.

    Revalidation must be executed if any last minute crankshaft/DMF/transmission redesign occurs.

21.9.3.8 402E_05: Engine roughness components calculation

21.9.3.8.1 General information:

The engine roughness calculation needs three individual components: a static component (DRV0_ER), a dynamic component (DRV1_ER) and a curvature component (DRV2_ER). These component definitions are specific to the engine cylinder number (see following related subchapters).

DRV2_MMV_ER is the filtered curvature component correction (DRV2_ER) who allows adding an artificial positive offset on ER value in case of dynamic curvature effects. This feature had a safety effect against wrong detection in case of segment oscillations.

ER_RAW is the engine roughness value without curvature component (DRV2_MMV_ER) used by cylinder balancing and engine warm up monitoring modules.

21.9.3.8.2 Calibration Step 402E_05 – Step 0 – No Calibration

ER components calculation is only managed regarding NC_CYL_NR configuration data, no calibration data involved.

21.9.3.9 402E_06: Engine roughness values determination

21.9.3.9.1 General information:

ER is the complete engine roughness value including all ER components (DRV0_ER, DRV1_ER & DRV2_MMV_ER). Specified & used by misfire detection module.

ER_STND is the normalised engine roughness value versus engine speed that could be amplified by a gain C_FAC_SCA_ER_STND. This is the nominal engine roughness value divided by the corresponding cubic corrected ER segment time $(SEG\_T\_COR\_BUF[NC\_SEG\_DLY\_ER])^3$. The cubic segment time is calculated by using a mantisse and an exponent.

ER index is used by MISF aggregate only if NLC_USE_ER_STND_MIS = 0. In other case MISF uses the ER_STND index like for ENRC and CYLB aggregates.

21.9.3.9.2 Calibration Step 402E_06 – Step 1/1 –Calibration

Maximum engine speed to activate ER_STND calculation: C_N_MAX_ER_STND

This maximum engine speed is determined in function of the aggregates that uses this data.

If NLC_USE_ER_STND_MISF = 0, ENRC & CYLB not integrated. No modules will use ER_STND. As ER_STND use triple multiplication and huge data formatting for normalisation who consume CPU resources at TDC update rate. It could be interesting to stop this calculation process especially at high engine speed. In such case recommended calibration C_N_MAX_ER_STND = 0 rpm -> calculation cut-off.

For other configurations, it depends of the maximum engine speed needed for Aggregates using ER_STND data. *See table below for configuration sum-up.*

No ENRC & No CYLB integrated	ENRC & No CYLB integrated	CYLB & No ENRC integrated	CYLB & ENRC integrated
C_N_MAX_ER_STND = 0 rpm	C_N_MAX_ER_STND = Max(Eng Speed of ENRC)	C_N_MAX_ER_STND = Max(Eng Speed of CYLB)	C_N_MAX_ER_STND = Max(Max Eng Speed of CYLB, Max Eng Speed of ENRC)
C_N_MAX_ER_STND = 4500 + 100 rpm	C_N_MAX_ER_STND = Max(4500 + 100 rpm, Max Eng. Speed of ENRC)	C_N_MAX_ER_STND = Max(4500 + 100 rpm, Max Eng. Speed of CYLB)	C_N_MAX_ER_STND = Max(4500 + 100 rpm, Max Eng. Speed of CYLB, Max Eng. Speed of ENRC)
C_N_MAX_ER_STND = redline + 100 rpm	C_N_MAX_ER_STND = redline + 100 rpm	C_N_MAX_ER_STND = redline + 100 rpm	C_N_MAX_ER_STND = redline + 100 rpm

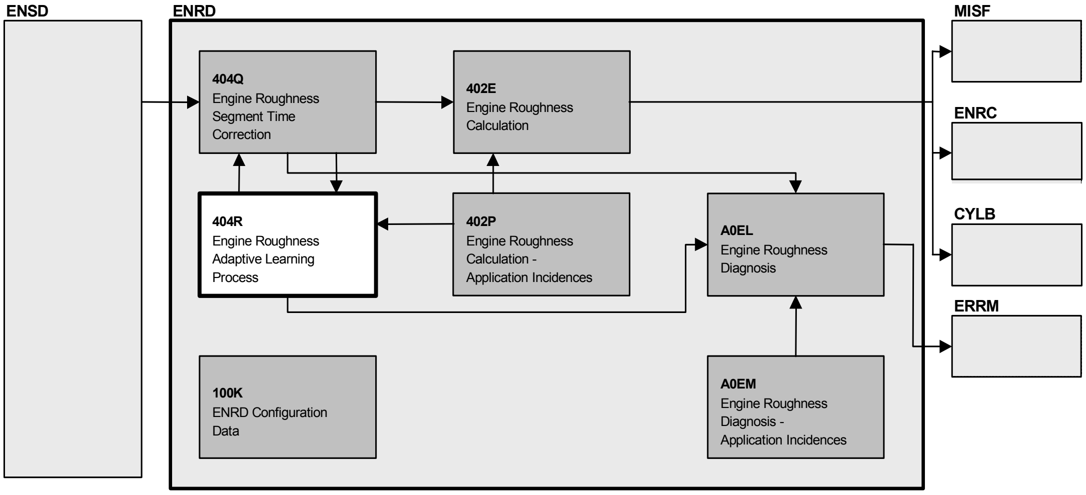
Scaling factor for normalised engine roughness index ER_STND:
C_FAC_SCA_ER_STND

This scaling factor have sense only when NLC_USE_ER_STND_MIS = 1 or ENRC integrated or CYLB integrated.

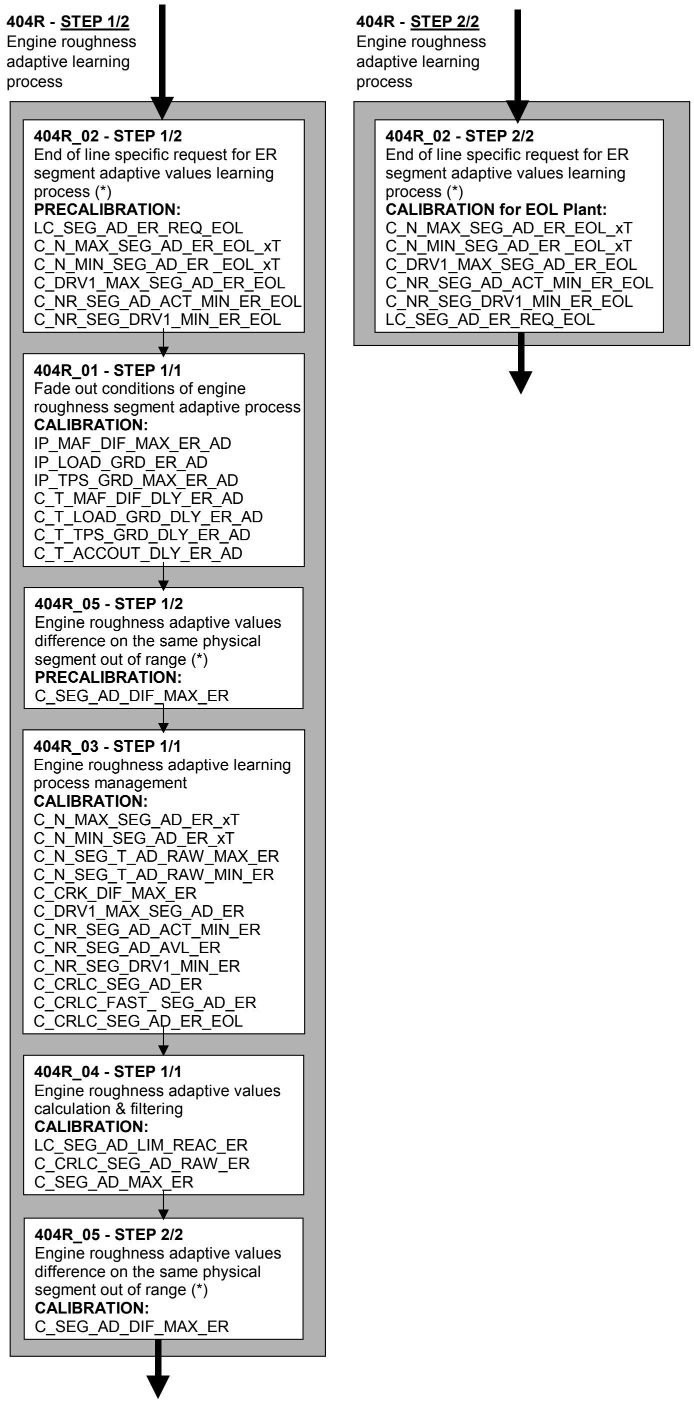
To determinate the best gain on ER_STND, please check that:

- No overflow occurs on ER_STND at low engine speed / high engine load with misfires (biggest engine speed drops possible).
- ER_STND quantum *(least resolution value)* must be smaller than at least the tenth of the smallest misfire drop at highest engine speed of legal target (4500rpm for EC or Redline for US) and lowest legal engine load for monitoring (see zero load line definition in MISF aggregate methodology).

21.9.4 Module 404R: Engine roughness adaptive learning process



21.9.4.1 Module calibration flowchart



21.9.4.2 404R_03: Engine roughness adaptive learning process management

21.9.4.2.1 General Information

The ER adaptive learning process is composed with two adaptive processes:

- A Fast adaptive process used for crankshaft/flywheel behaviour monitoring SEG_AD_RAW_MMV_ER[NC_CYL_NR]: The 'raw' adaptive values learned are not saved, there are only used to stop the slow adaptive process in case of important difference between SEG_AD_RAW_MMV_ER[NC_CYL_NR] values (fade-out condition).
- A Slow adaptive process used for flywheel mechanical noise learning SEG_AD_MMV_ER[NC_CYL_NR]. The adaptive values learned are used to correct basic segment time samples (SEG_T_MES).

The slow adaptive process is managed in two phases:

- A fast adaptive process for end of line procedures (EOL) who allow a fast convergence to ER flywheel adaptive values.
- At the end of the fast adaptive process, a slow adaptive process who will monitor ER adaptive values during the engine lifetime.

At the beginning of the fast segment adaptation learning process, it is necessary to upper the engine roughness value with an additional offset (THD_AD_ER), depending on the worst target wheel tolerances excepted to avoid wrong misfire detections due to flywheel mechanical tolerances. For instance, a flywheel with a ± 0.3°Crk mechanical tolerance on the ER segment corresponds to following segment drift:

Engine Type	ER Segment length	Induced Drift with ± 0.3°Crk tolerance on ER segment
3 cyl engine	240°Crk	2.5‰
4 cyl engine	180°Crk	3.33‰
5 cyl engine	144°Crk	4.17‰
6 cyl engine	120°Crk	5‰
8 cyl engine	90°Crk	6.68‰

The offset value (THD_AD_ER) by which the ER is increased at the beginning is gradually reduced to zero in a linear way until the 95% correction is obtained.

An internal adaptation counter CTR_SEG_AD_ER controls this number of engine cycle steps.

21.9.4.2.2 Calibration Step 404R_03 – Step 1/1 – Calibration

The calibration should be performed at the chassis-dyno with a max. tolerance target wheel but it is quite difficult to have one at its disposal, therefore any target wheel type can be used.

Example of calibration:

C_N_MAX_SEG_AD_ER_xT= 3500 rpm

C_N_MIN_SEG_AD_ER_xT = 2000 rpm

C_N_SEG_T_AD_RAW_MAX_ER = 3700 rpm

C_N_SEG_T_AD_RAW_MIN_ER = 1800 rpm

C_DRV1_MAX_SEG_AD_ER = 200

C_NR_SEG_DRV1_MIN_ER = 1 or 2 times NC_CYL_NR

C_NR_SEG_AD_ACT_MIN_ER = 1 or 2 times NC_CYL_NR

C_CRLC_SEG_AD_ER = 0.05

C_CRLC_FAST_SEG_AD_ER = 0.1

C_CRLC_SEG_AD_ER_EOL = 0.75

C_CRK_DIF_MAX_ER = *Check with Maximum Flywheel Tolerances provided by manufacturer for serial production, not for prototypes.*

C_NR_SEG_AD_AVL_ER = when SEG_AD_MMV_ER[NC_CYL_NR] reaches 95% of their final values = -ln(0.05) / C_CRLC_SEG_AD_ER_EOL = 3 / C_CRLC_SEG_AD_ER_EOL

Calibration of C_N_SEG_T_AD_RAW_MAX_ER, C_N_MAX_SEG_AD_ER_xT, C_N_SEG_T_AD_RAW_MIN_ER & C_N_MIN_SEG_AD_ER_xT:

Crankshaft oscillations (due to DMF, CVT ...) in some speed areas may disturb the adaptive process and may cause misdetections of misfire afterwards. To check this, the engine must be run at different speeds.

Remark: Check this item very carefully for engines with twin mass flywheels (DMF).

Set C_CRLC_SEG_AD_ER at 0.2 and the engine speed max and min thresholds to: C_N_SEG_T_AD_RAW_MIN_ER = C_N_MIN_SEG_AD_ER_xT = 900 rpm and C_N_SEG_T_AD_RAW_MAX_ER = C_N_MAX_SEG_AD_ER_xT = 6000 rpm (or redline) to enable the fast and slow target wheel learning processes in the complete engine speed range.

Run the complete engine speed range from 6000 to 1000 rpm very slowly in fuel cut off at several gear ratio and several temperatures. Control that values of SEG_AD_MMV_ER[NC_CYL_NR] do not vary more than 0.5‰ (stability of the adaptation value) for the calibration of the engine speed threshold.

Find the engine speed range where SEG_AD_MMV_ER[NC_CYL_NR] adaptive values are the more steady.

If some problems are observed *(curves, resonance)*, the adaptation has to be forbidden on such an area. Very often below 2000 rpm, due to drive train oscillations, the adaptive values start to diverge and above 3500 rpm adaptive values start to drift on some applications, to be checked.

Another point is to be sure that the adaptive processes are correctly executed and achieved during the End of Line process on chassis rolls in a way to have a full legal compliant car after EOL process. This point is a critical one, especially with AT and CVT transmission vehicles.

An adaptive range from 2000 rpm to 3500 rpm is often obtained and preferable. Control that the fuel cut-off phase is active inside the selected engine speed range.

Calibration of C_CRLC_SEG_AD_ER:

These calibrations impact the stability of the slow adaptation values SEG_AD_MMV_ER[NC_CYL_NR]

Select afterwards an engine speed between both previous engine speed thresholds (ex: 2500 rpm) at fuel cut off.

Make the filtering value C_CRLC_SEG_AD_ER vary for ex from 0.1 / 0.075 / 0.05 / 0.025 to 0.01.

Choose the value for which the adaptive value SEG_AD_MMV_ER[NC_CYL_NR] is nearly constant (variation lower than 0.5‰).

Robustness against disturbances

Check on the street, in fuel cut-off mode (PUC engine state), with several gear ratios by increasing and decreasing of the engine speed that the value of C_CRLC_SEG_AD_ER enables stable values of SEG_AD_MMV_ER[NC_CYL_NR]. Try several values of C_CRLC_SEG_AD_ER.

Remark: The segment stability should be checked during hot trip with warm engine. Hot conditions weaken drive train stiffness.

Result

The chosen value of C_CRLC_SEG_AD_ER should be the best compromise between adaptation stability at steady state conditions, fast learning and robustness against disturbances.

A value of 0.05 has shown in several cases to be the best compromise between adaptation stability, fast learning and robustness against disturbances. Notice that a value below this one could slow down too much the target wheel learning process.

Calibration of C_CRLC_SEG_AD_ER_EOL & C_NR_SEG_AD_AVL_ER

C_CRLC_SEG_AD_ER_EOL has to be chosen according two targets:

- Number of TDC in fuel cut-off mode during the EOL process *(normally not a critical point. often at least 100 tdc's in fuel cut-off)*
- Avoid learning too noisy adaptive values due to a too light filtering, even if corrected by slow adaptive process during engine lifetime.

C_NR_SEG_AD_AVL_ER is the needed number of engine cycles in EOL adaptive process to reach 95% of the final SEG_AD_MMV_ER[NC_CYL_NR] adaptive values obtained with C_CRLC_SEG_AD_ER_EOL *(1^{st} order filter)*

C_NR_SEG_AD_AVL_ER = -ln(0.05) / C_CRLC_SEG_AD_ER_EOL

≈ 3 / C_CRLC_SEG_AD_ER_EOL

Calibration of C_DRV1_MAX_SEG_AD_ER, C_NR_SEG_DRV1_MIN_ER & C_NR_SEG_AD_ACT_MIN_ER:

DRV1_ER is representative of the deceleration or acceleration of the engine in fuel cut-off.

Therefore for the robustness of the adaptation against disturbances, check on the street in deceleration phases and some brake events, with C_DRV1_MAX_SEG_AD_ER=200, if the adaptive values *(SEG_AD_MMV_ER[NC_CYL_NR])* are disturbed in a great extent *(It should stay nearly constant at the end of the adaptive phase)*. If the adaptive values are disturbed then measured the value of DRV1_ER in parallel to the adaptive value and reduces the value of C_DRV1_MAX_SEG_AD_ER if necessary.

C_NR_SEG_DVR1_MIN_ER = 2 times NC_CYL_NR is recommended.

C_NR_SEG_AD_ACT_MIN_ER is the delay in number of segment needed before starting the adaptive process after a fuel cut-off.

C_NR_SEG_AD_ACT_MIN_ER = 2 times NC_CYL_NR is recommended to avoid any phenomenon like extra combustions due to wallfilm or crankshaft oscillations after fuel cut-off.

21.9.4.3 404R_02: EOL specific request for ER segment adaptive learning process

21.9.4.3.1 General Information

During EOL process (LV_EOL_OBD = 1), it's possible to bypass nominal conditions for slow adaptive learning process with specific EOL calibrations. These calibrations must be adapted to customer EOL plant.

21.9.4.3.2 Calibration Step 404R_02 – Step 1/2 – Precalibration

In this calibration phase, check that LV_EOL_OBD isn't not active to not bypass nominal adaptive process conditions or else set LC_SEG_AD_ER_REQ_EOL = 1 (if LV_EOL_OBD is set to 0, LV_SEG_AD_FDOUT_ER_EOL will be forced to 1).

21.9.4.3.3 Calibration Step 404R_02 – Step 2/2 – Calibration

Example of calibration : *(identical to nominal adaptive conditions)*

C_N_MAX_SEG_AD_ER_EOL_xT = 3500 rpm

C_N_MIN_SEG_AD_ER _EOL_xT = 2000 rpm

C_N_MIN_SEG_AD_ER _EOL_MT = 2000 rpm

C_DRV1_MAX_SEG_AD_ER_EOL = 200 µs

C_NR_SEG_AD_ACT_MIN_ER_EOL = 1 or 2 times NC_CYL_NR

C_NR_SEG_DRV1_MIN_ER_EOL = 1 or 2 times NC_CYL_NR

LC_SEG_AD_ER_REQ_EOL = 1

Calibration

Adapt to the customer EOL plan:

- The engine speed range for adaptive process activation with C_N_MAX_SEG_AD_ER_EOL & C_N_MIN_SEG_AD_ER_EOL calibrations.
- The Maximum DRV1 value C_DRV1_MAX_SEG_AD_ER_EOL & associated delay C_NR_SEG_DRV1_MIN_ER_EOL to disable process activation if DRV1 exceeds.
- The delay C_NR_SEG_AD_ACT_MIN_ER_EOL to disable the adaptive process after beginning of the adaptive process.

Note : if LC_SEG_AD_ER_REQ_EOL is set to 0, EOL bypass for adaptive process will be always disable

21.9.4.4 404R_01: Fade out conditions of engine roughness segment adaptive process

21.9.4.4.1 General Information

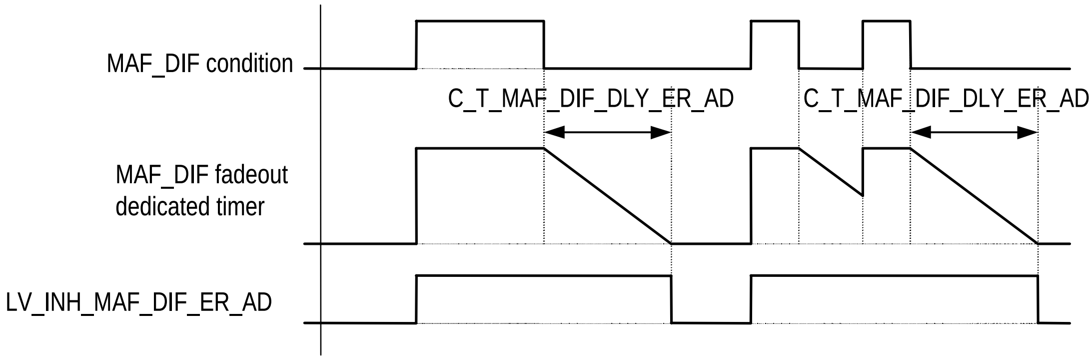
LV_SEG_AD_FDOUT_ER combines all system conditions to generate a fade out condition as input for the generic segment slow adaptation process, who provide SEG_AD_MMV_ER[NC_CYL_NR] values.

21.9.4.4.2 Calibration Step 404R_01 – Step 1/1 – Calibration

Calibration of IP_MAF_DIF_MAX_ER_AD & C_T_MAF_DIF_DLY_ER_AD ***: maximum air-mass gradient for adaptive process***

Due to trailing throttle / acceleration transition problems it is necessary to disable ER segment learning process for a short period when the air-mass gradient per segment exceeds a calibratable value *IP_MAF_DIF_MAX_ER_AD(MAF)*

Fade out behaviour summary:



Especially for engines with dual mass flywheels (DMF), critical behaviours at change from positive to negative torque on the power train could produce oscillations on crankshaft that could disturb slow adaptive process.

Check at the transitions from fuel cut off to part load and vice versa as well as all transition from low load to high load, if some speed fluctuations are critical to adaptive process.

Measure in parallel the load and look at which load difference the speed fluctuation is forced.

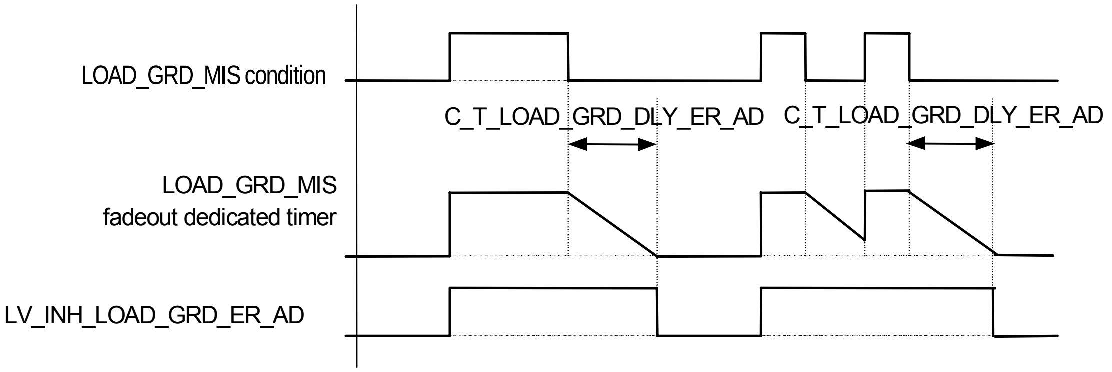
- Example: *IP_MAF_DIF_MAX_ER_AD(MAF). C_T_MAF_DIF_DLY_ER_AD* = 0.6s

MAF	IP_MAF_DIF_MAX_ER
87	2.2
103	5.0
169	20.0
218	40.0
261	65.4
272	201.5

Calibration of IP_LOAD_GRD_ER_AD & C_T_LOAD_GRD_DLY_ER_AD *: maximum engine load gradient for adaptive process*

Due to general intervention of the torque model it's necessary to disable ER segment adaptation for a short period when the actual torque gradient exceeds a calibratable value, *IP_LOAD_GRD_ER_AD(LOAD_MIS)*

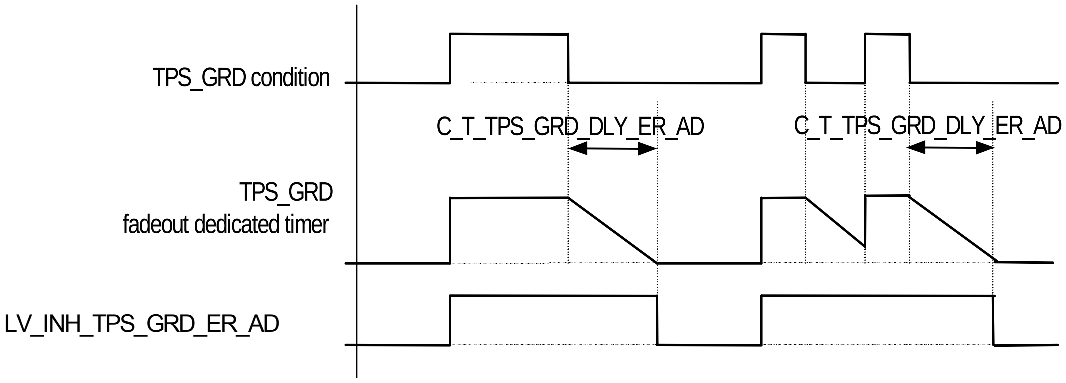
Fade out behaviour summary:



Calibration of IP_TPS_GRD_MAX_ER_AD & C_T_TPS_GRD_DLY_ER_AD*: maximum throttle gradient for adaptive process*

Due to trailing throttle / acceleration transient problems at low engine load it is necessary to disable ER segment adaptation for a short period when the throttle gradient exceeds a calibratable value, *IP_TPS_GRD_MAX_ER_AD*

Fade out behaviour summary:



The fade out through throttle gradient normally works at the transition from part load to fuel-cut-off. The air mass gradient is in this case too slow.

Example: IP_TPS_GRD_MAX_ER_AD. *C_T_TPS_GRD_DLY_ER_AD* = 0.8s

TPS_GRD	IP_TPS_GRD_MAX_ER_AD
2.8	23.4
12.2	70.3
20.2	140.6
23.4	187.5
27.2	398.4
50.1	996.0

Calibration of C_T_ACCOUT_DLY_ER_*AD Air - conditioning compressor activation*

When the air - conditioning compressor is switched on, an additional load is briefly applied to the engine.

This load jump could cause an ER segment period jump and crankshaft vibration.

To define C_T_ACCOUT_DLY_ER_AD, observe on SEG_AD_MMV_ER[NC_CYL_ER] the duration of the disturbance that could occurs on the crankshaft when ACC is active.

Example : C_T_ACCOUT_DLY_ER_AD = 0.75s

21.9.4.5 404R_04: Engine roughness adaptive values calculation & filtering

21.9.4.5.1 General Information

This function allows to determinate the coefficients for ER segment adaptive values filters used in the learning process. Three different stages can be differentiated:

- Nominal ER segment adaptive values filtering mode, used to monitor adaptive values during engine lifetime.
- Fast ER segment adaptive values filtering mode, used to quickly obtain valid adaptive values when the EMS is new.
- EOL specific filtering mode, to be used on EOL or special service request.

LV_SEG_AD_LIM_ER is set if at least one SEG_AD_MMV_ER[NC_CYL_NR] value exceeds C_SEG_AD_MAX_ER.

21.9.4.5.2 Calibration Step 404R_04 – Step 1/1 – Calibration

Calibration of C_CRLC_RAW_SEG_AD_ER

The aim of the fast adaptive process that provides SEG_AD_RAW_MMV_ER is to monitor the behaviours of the crankshaft that could produced wrong learning's on slow adaptive process, like oscillations produced by Dual Mass Flywheel (DMF) on a defined engine speed range.

Filtering induced by C_CRLC_FAST_SEG_AD_ER coefficient has to be enough light to observe such phenomenon, but not too much to avoid over fade-out due to noises on the slow adaptive learning process. Must be lighter than C_CRLC_SEG_AD_ER.

Example: C_CRLC_SEG_AD_ER = 0.05 ; C_CRLC_FAST_SEG_AD_ER = 0.1

LC_SEG_AD_LIM_REAC_ER allows to reactivate or not the absolute SEG_AD_MMV_ER[NC_CYL_NR] diagnosis. This absolute diagnosis is set as soon as at least one of the segment adaptive values exceeds the C_SEG_AD_MAX_ER calibration.

LC_SEG_AD_LIM_REAC_ER = 1, the diagnosis result is reinitialised each new activation of the segment adaptive process.

Generally C_SEG_AD_MAX_ER is set close to C_CRK_DIF_MAX_ER, the m*aximum Flywheel Tolerances provided by manufacturer for serial production, not for prototypes.*

21.9.4.6 404R_05: Engine roughness adaptive values difference on the same physical segment out of range

21.9.4.6.1 General information

Due to possible feedback from Dual Mass Flywheel oscillations *(see transmission/crankshaft design)* in adaptive process engine speed range, fluctuations on ER adaptive values of identical physical segments can induce dissimilarity in ER segment correction.

If difference between identical physical segments ER fast filtered adaptive values exceeds a calibratable gap *C_SEG_AD_DIF_MAX_ER,* a fade out of segment adaptive process is applied.

21.9.4.6.2 Calibration Step 404R_05 – Step 1/2 – Precalibration

In a first step, to avoid fade-out of segment adaptive process due to LV_SEG_AD_DIF_MAX_ER = 1, set C_SEG_AD_DIF_MAX_ER equals to C_CRK_DIF_MAX_ER

21.9.4.6.3 Calibration Step 404R_05 – Step 2/2 – Calibration

If on your application, crankshaft oscillations phenomenon occurs during fuel cut-off due for instance to DMF or CVT shifting, observe on SEG_AD_RAW_MMV_ER[NC_CYL_NR] values which amount of ER adaptive values difference on same physical segment could induce on SEG_AD_MMV_ER[NC_CYL_NR] drifts exceeding 0.5‰.

SEG_AD_RAW_MMV_ER[NC_CYL_NR] must change enough quickly *(see calibration of filter coefficient C_CRLC_SEG_AD_RAW_ER)* to allow to detect this drift amount before having too much impact on slow filtered adaptive values SEG_AD_MMV_ER[NC_CYL_NR].

The slow adaptive process will be retriggered only when all SEG_AD_DIF_ER_x differences will be less than C_SEG_AD_DIF_MAX_ER.

Check than the drifts of SEG_AD_RAW_MMV_ER[NC_CYL_NR] values don't come from a too wide engine speed range for fast adaptive process (drifts can occurs at low and high engine speed), *see C_N_SEG_T_AD_RAW_MAX_ER & C_N_SEG_T_AD_RAW_MIN_ER calibrations values.*

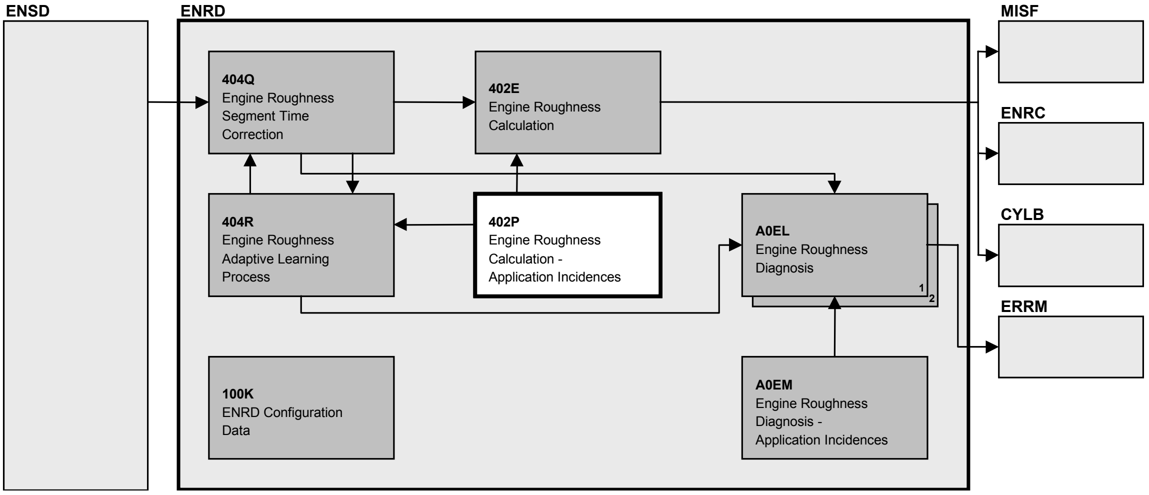
21.10 Engine roughness calculation & adaptation (cus)

21.10.1 Requirements

ENSD aggregate active and calibrated, no synchronisation errors/troubles

IGRE, IGSP & INJR precalibrated

21.10.2 Tuning hints for Engine roughness calculation - Application Incidences (Module 402P)



21.10.2.1 Inhibition of engine roughness adaptive process - Application specific

21.10.2.1.1 General Information

Engine roughness segments adaptive process can be inhibited when some system conditions that could disturb the adaptive process occur.

21.10.2.1.2 Calibration Steps

For this section, please see your application incidences file and associated calibration recommendations from your SV SI team.

LV_INH_APP_ER_AD must be set to 0, for nominal adaptive process conditions.

21.10.2.2 Engine roughness adaptive process inhibition related to OBDI diagnosis

21.10.2.2.1 General Information

Engine roughness segments adaptive process can be inhibited when some OBDI error that could disturb the adaptive process occurs (CRK, CAM, MAF, VVT ... sensors).

21.10.2.2.2 Calibration Steps

For this section, please see your application incidences file and associated calibration recommendations from your SV SI team (if using calibratable incidences matrix or QVM).

LV_INH_OBD_ER_AD must be set to 0, for nominal adaptive process conditions.

21.10.2.3 End of line & After sale service request to reset ER segment adaptive values

21.10.2.3.1 General Information

During end of line plant and/or for After sale services, a request is generated via communication protocol, to reset engine roughness segment adaptive values & learning process *(see project communication protocol).*

21.10.2.3.2 Calibration Steps

For this section, please see your application incidences file and associated calibration recommendations from your SV SI team (if using calibratable incidences matrix or QVM).

LV_SEG_AD_RST_ER_EOL must be set to 0, for nominal adaptive process conditions.

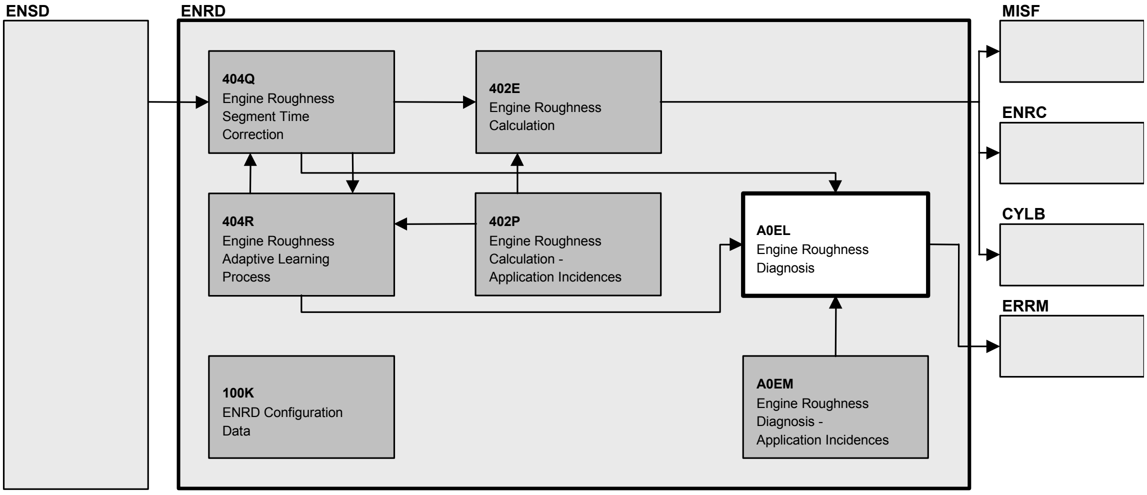
21.11 Engine roughness OBDII (gen)

21.11.1 Requirements

ENSD aggregate active and calibrated, no synchronisation errors/troubles

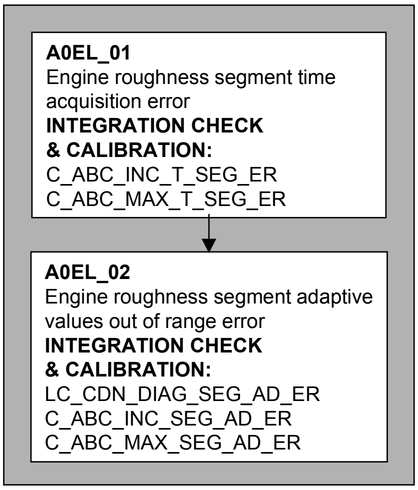
IGRE, IGSP & INJR precalibrated

21.11.2 Tuning hints for Engine roughness diagnosis (Module A0EL)



21.11.2.1 Module calibration flowchart

A0EL - STEP 1/1
Engine roughness
diagnosis



A0EL_01
Engine roughness segment time acquisition error
INTEGRATION CHECK
& CALIBRATION:
C_ABC_INC_T_SEG_ER
C_ABC_MAX_T_SEG_ER

A0EL_02
Engine roughness segment adaptive values out of range error
INTEGRATION CHECK
& CALIBRATION:
LC_CDN_DIAG_SEG_AD_ER
C_ABC_INC_SEG_AD_ER
C_ABC_MAX_SEG_AD_ER

21.11.2.2 Module calibration prerequisites

ENSD robustness calibration executed

ECU C sample & crankshaft, sensor, flywheel design frozen

No diagnosis inhibition coming from Module A0EM: Engine roughness diagnosis - Application Incidences *(see project specific document based on D7 template 30Q01R01.00A).* The project specific tuning hint dedicated to this module is called: ER OBDII (cus).

21.11.2.3 A0EL_01: Engine roughness segment time acquisition error

Configuration data

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
ER segment time acquisition	*Missing/Adding 1 tooth or more on engine roughness segment acquisition*	1	P0339 *(example)*	P0339 *(example)*	MIL_KEY / CARB_M
		2			
		3			
T_SEG_ER					

21.11.2.3.1 General Information

The purpose of this error is to diagnose synchronisation errors on engine roughness segments measurement.

An error symptom “*Missing/Adding 1 tooth or more on engine roughness segment acquisition*” is detected when a crankshaft synchronisation error caused by 1 or 2 missing/additional tooth occurs on one crankshaft revolution.

Note: In ENSD, most of the cases, the criterion to set LV_INH_MIS_CRK is smaller in term of missing/adding teeth number than the criterion to declare an engine desynchronisation. In consequence, LV_INH_MIS_CRK can be set without engine desynchronisation.

21.11.2.3.2 Calibration Step A0EL_01 – Step 1/1

Evaluate in mission, with a large C_ABC_MAX_T_SEG_ER maximum value and C_ABC_INC_T_SEG_ER = 1, the frequency of LV_INH_MIS_CRK occurrences.

A too important number of occurrences on one trip could bring to light troubles in ENSD robustness tuning, sensor and flywheel troubles.

There’s a handshake between LV_INH_MIS_CRK (low-level validity information) and ENRD application validity informations (LV_ENA_SEG_T_MES & LV_ENA_ER). As a final effect, it will fade-out monitoring in MISF & control in ENRC & CYLB.

If occurrences of LV_INH_MIS_CRK are too important, finally it could reduce the misfire monitoring coverage on cycle (act as a fade-out in the function).

Finally, to define C_ABC_MAX_T_SEG_ER value, you need to evaluate a compromise between error over detection and impact on monitoring ratio.

Note: there's no direct risk between misfire wrong detections and the bit LV_INH_MIS_CRK, as this bit will automatically shut-off the ER index calculation and inhibit the misfire monitoring as well. This diagnosis is mainly used to inform a too important occurrence of this error, which could have an impact on the diagnosis ratio for misfire monitoring. Most of the case, an ENSD error will be set before LV_ERR_T_SEG_ER, except if each time LV_INH_MIS_CRK is set without ENSD desynchronisation.

For associated Pcode, see if customer specific code already defined.

Subclass A	
Name	**Description**
MIL_KEY	Failure with impact on MIL / key cycle ***(Recommended)***

21.11.2.4 A0EL_02: Engine roughness segment adaptive values out of range error

Configuration data

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
Er segment adaptation values at the limit	*Er segment adaptation values at the limit*	1	P0336 *(example)*	P0336 *(example)*	MIL_KEY / CARB_M
		2			
		3			
SEG_AD_ER					

21.11.2.4.1 General Information

The purpose is to diagnose the absolute range of ER segment adaptive values.

If during **or** at the end of the segment adaptive values learning process *(LV_SEG_AD_ER = 1 or LV_SEG_AD_ER = 1 -> 0),* at least one of the engine roughness segment adaptation values is at the limit (LV_SEG_AD_LIM_ER = 1), the error symptom SEG_AD_ER “engine roughness segment adaptation values at the limit” is detected.

21.11.2.4.2 Calibration Step A0EL_02 – Step 1/1

If LC_CDN_DIAG_SEG_AD_ER = 0, the adaptive values out of range information (LV_SEG_AD_LIM_ER) will be taken in account only at the end of the adaptive process. For this solution, an important maximum ABC counter calibration isn’t required (for instance C_ABC_MAX_SEG_AD_ER = 10).

Otherwise, if LC_CDN_DIAG_SEG_AD_ER = 1, the diagnosis will be executed continuously. In this case the maximum ABC counter calibration must be important if you would to be able to set the error based on more than only one adaptive process.

For associated Pcode, see if customer specific code already defined.

Subclass B	
Name	**Description**
CARB_M	CARB failure Middle priority *(Recommended)*

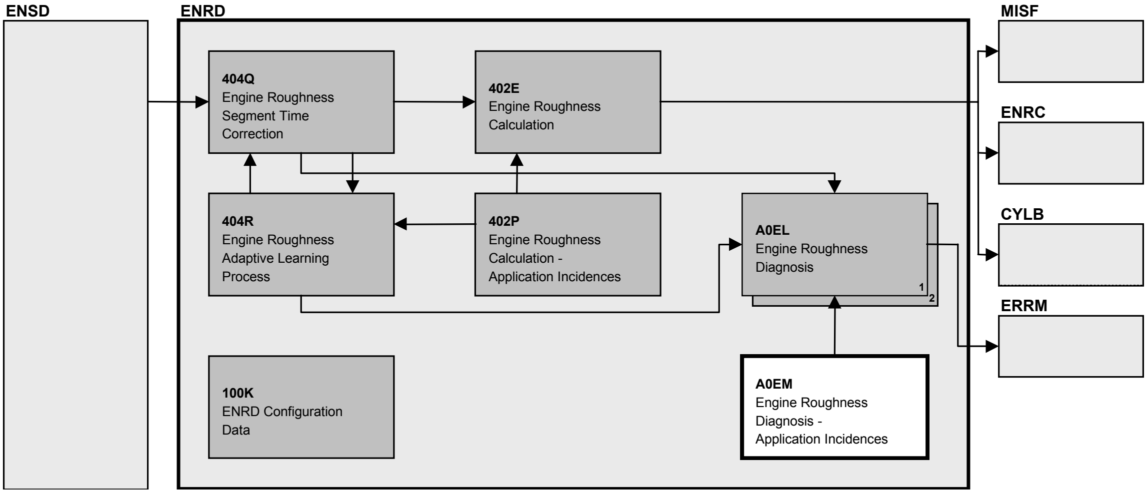
21.12 Engine roughness OBDII (cus)

21.12.1 Tuning requirements

ENSD aggregate active and calibrated, no synchronisation errors/troubles

IGRE, IGSP & INJR precalibrated

21.12.2 Tuning hints for Engine roughness diagnosis - Application Incidences



21.12.2.1 General information

For integration test and calibration, please contact your SI team for recommendations according what has been integrated for project specific inhibition of LV_ERR_T_SEG_ER & LV_ERR_SEG_AD_ER diagnosis *(application incidences file)*.

For a proper calibration of generic diagnosis, LV_INH_DIAG_T_SEG_ER & LV_INH_DIAG_SEG_AD_ER must be set to 0.

21.13 Customer adaptation module: AGGR ENRD

21.13.1 Outputs for BMW functions which are defined as ENRD exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Fak_er_schw	O/V	0...FFH	0...2.55	0.01	[-]
Quotient aus Engine Roughness und dem Schwellenwert der Aussetzererkennung					
Lur[NC_CYL_NR]	O/V	8000...7FFFH	-32768...32767	1	[1/s²]
engine roughness cylinder selektiv					
Lur_aus_quo	O/V	0...FFH	0...1.99218	0.0078125	[-]
Quotient Laufunruhewert / Aussetzerschwelle					

Input data:

ER_AV_QUO	ER_CYL[NC_CYL_NR]	NC_CYL_NR	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation at reset or at exit PWL:	0
Recurrence :	every segment
Activation:	every engine state
Special value at PWL:	Lur_aus_quo = 0 Fak_er_schw = 0

Formula section:

Remark: all formulas are valid in a **physical** meaning

Lur[NC_CYL_NR] = ER_CYL[NC_CYL_NR]
Lur_aus_quo = ER_AV_QUO
Fak_er_schw = ER_AV_QUO

22 Engine speed control

Table of Contents

Index

22.1 Driving off assistance via torque request

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ISC_OFF_DROF	O/V	0...1H	0...1	1	-
Boolean to enable/disable idle speed controller during drive off; 1=disabled					
LV_TQ_DROF_IS_AD_INH	O/V	0...1H	0...1	1	-
flag for inhibition of IS-control adaptations					
TQ_ADD_PL_DROF	O/V	0...7FFFH	0...1.02397E+3	0.03125004	Nm
TQ reserve for drive off support					
TQ_DROF_FAST	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
TQ request for drive off assistance					
TQ_DROF_SLOW	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
TQ request for drive off assistance					
N_DIF_DROF	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Engine speed deviation for DROF					
N_DIF_PRED_PL_DROF	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted idle speed control variable at drive-off-support					

Input data:

FAC_TQ_REQ	LV_AT	LV_BRAKE_DET	LV_CS
LV_ERR_CRK	LV_ERR_PVS	LV_ERR_VS	LV_N_SP_IS_CS
LV_ST_END	N	N_DIF	N_GRD
N_SP_IS	OPM_AV	TIA	TOIL
TQ_AV	VS_FIL	LV_VAR_TCT	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_DIF_DROF_MAX	1	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Maximum N_DIF_DROF for function TQ_DROF					
C_N_DIF_DROF_MIN	1	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Minimum N_DIF_DROF for function TQ_DROF					
C_TQ_ADD_PL_DROF_LGRD	1	0...FFH	0...7.96875	0.03125	Nm
Gradient limitation for drive off support TQ reserve					
C_TQ_DROF_AD	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
threshold TQ_DROF for inhibition of IS-control adaptations					
C_T_TQ_DROF_DLY_AD	1	0...FFH	0...2.55	0.01	s
timer TQ_DROF for inhibition of IS-control adaptations					
LC_TQ_ADD_PL_DROF_ENA_BRAKE	1	0...1H	0...1	1	-
Switch to select deactivation of drive off support TQ reserve by brake pedal; 0: deactivation when brake, 1: no deactivation when brake					
LC_TQ_DROF_ISC_OFF	1	0...1H	0...1	1	-
Boolean to enable/disable idle speed controller during drive off; 1=disabled					
ID_IDX_OPM_DROF	8	0...1H	0...1	1	-
LDPM_OPM_AV	8	0...8H	0...8	1	[-]
Switch map to choose DROF calculation path (OPM_1 or OPM_2) depending on OPM_AV					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_N_GRD_TQ_AV	4	0...FFFFH	0...1.999969	3.05176E-5	-
LDP_TQ_AV_IP_FAC_N_GRD	4	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	[Nm]
Weighing of additive part of N_DIF_DROF dependant from actual torque					
IP_FAC_TQ_DROF_FAC_TQ_REQ	6	0...FFFFH	0...1.999969	3.05176E-5	-
LDP_FAC_TQ_REQ_IP_FAC_TQ_DROF	6	0...FFFFH	0...1.99996	3.05E-5	[-]
Weighing of IP_TQ_DROF dependant from FAC_TQ_REQ					
IP_FAC_VS_TQ_ADD_PL_DROF	4	0...FFFFH	0...1.999969	3.05176E-5	-
LDP_VS_FIL_IP_FAC_VS_TQ_ADD_PL	4	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed correction for IP_TQ_ADD_PL_DROF					
IP_TQ_DROF_GRD_DEAC_OPM_1	6	0...7FFFH	0...1.02397E+3	0.03125	Nm
LDP_TQ_DROF_GRD_DEAC_1	6	0...7FFFH	0...1.02397E+3	0.03125	[Nm]
Decrement for the termination ramp of TQ_DROF_SLOW/_FAST in operation mode 1					
IP_TQ_DROF_GRD_DEAC_OPM_2	6	0...7FFFH	0...1.02397E+3	0.03125	Nm
LDP_TQ_DROF_GRD_DEAC_2	6	0...7FFFH	0...1.02397E+3	0.03125	[Nm]
Decrement for the termination ramp of TQ_DROF_SLOW/_FAST in operation mode 2					
IP_T_N_DIF_OFS_PRED_DROF	8	0...FFH	0...7.96875	0.03125	s
LDPM_N_T_N_DIF_OFS_PRED	8	0...1FE0H	0...8.16E+3	1	[rpm]
Time constant for calculation of predicted engine speed deviation at drive-off-support (PL)					
ID_TQ_DROF_ENA	3x3	0...1H	0...1	1	-
LDP_N_DIF_DROF__ID_TQ_DROF	3	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	[rpm]
LDP_N_GRD__ID_TQ_DROF	3	0...FFH	-4.096E+3 ... 4.064E+3	32	[rpm/s]
enable conditions					
IP_FAC_TEMP_TQ_ADD_DROF_OPM_1	6x6	0...FFFFH	0...1.99996948	3.05176E-5	-
LDP_TIA_IP_FAC_TEMP_TQ_ADD_1	6	0...FEH	-48...142.5	0.75	[°C]
LDP_TOIL_IP_FAC_TEMP_TQ_ADD_1	6	0...C8H	-40...160	1	[°C]
Temperature correction for IP_TQ_ADD_PL_DROF in operation mode 1					
IP_FAC_TEMP_TQ_ADD_DROF_OPM_2	6x6	0...FFFFH	0...1.99996948	3.05176E-5	-
LDP_TIA_IP_FAC_TEMP_TQ_ADD_1	6	0...FEH	-48...142.5	0.75	[°C]
LDP_TOIL_IP_FAC_TEMP_TQ_ADD_1	6	0...C8H	-40...160	1	[°C]
Temperature correction for IP_TQ_ADD_PL_DROF in operation mode 2					
IP_FAC_TQ_DROF_VS_OPM_1	6x6	0...FFFFH	0...1.99996948	3.05176E-5	-
LDP_VS_FIL_IP_FAC_TQ_DROF_VS_1	6	0...FFFFH	0...511.99218	0.0078125	[km/h]
LDP_N_IP_FAC_TQ_DROF_VS_1	6	0...1FE0H	0...8.16E+3	1	[rpm]
Weighing of IP_TQ_DROF dependant from engine- and vehicle speed in operation mode 1					
IP_FAC_TQ_DROF_VS_OPM_2	6x6	0...FFFFH	0...1.99996948	3.05176E-5	-
LDP_VS_FIL_IP_FAC_TQ_DROF_VS_1	6	0...FFFFH	0...511.99218	0.0078125	[km/h]
LDP_N_IP_FAC_TQ_DROF_VS_1	6	0...1FE0H	0...8.16E+3	1	[rpm]
Weighing of IP_TQ_DROF dependant from engine- and vehicle speed in operation mode 2					
IP_N_DIF_ADD_DROF_OPM_1	4x4	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDP_N_GRD_IP_N_DIF_ADD_DROF_1	4	0...FFH	-4.096E+3 ... 4.064E+3	32	[rpm/s]
LDP_N_SP_IS_IP_N_DIF_ADD_DROF_1	4	0...1FE0H	0...8.16E+3	1	[rpm]
Additive part of N_DIF_DROF dependant from engine speed gradient in operation mode 1					
IP_N_DIF_ADD_DROF_OPM_2	4x4	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDP_N_GRD_IP_N_DIF_ADD_DROF_1	4	0...FFH	-4.096E+3 ... 4.064E+3	32	[rpm/s]
LDP_N_SP_IS_IP_N_DIF_ADD_DROF_1	4	0...1FE0H	0...8.16E+3	1	[rpm]
Additive part of N_DIF_DROF dependant from engine speed gradient in operation mode 2					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQ_ADD_PL_DROF_OPM_1	6x6	0...7FFFH	0...1.02397E+3	0.03125	Nm
LDP_N_DIF_PRED_FAC_TQ_ADD_IS_1	6	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	[rpm]
LDP_FAC_TQ_REQ_DROF_1	6	0...FFFFH	0...1.99996	3.05E-5	[-]
Torque reserve at part load for drive-off-support in operation mode 1					
IP_TQ_ADD_PL_DROF_OPM_2	6x6	0...7FFFH	0...1.02397E+3	0.03125	Nm
LDP_N_DIF_PRED_FAC_TQ_ADD_IS_1	6	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	[rpm]
LDP_FAC_TQ_REQ_DROF_1	6	0...FFFFH	0...1.99996	3.05E-5	[-]
Torque reserve at part load for drive-off-support in operation mode 2					
IP_TQ_DROF_FAST_OPM_1	11x10	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_DROF_IP_TQ_DROF_1	11	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	[rpm]
LDPM_N_GRD_IP_TQ_DROF_1	10	0...FFH	-4.096E+3 ... 4.064E+3	32	[rpm/s]
Torque for driving off assistance, basic value value in operation mode 1					
IP_TQ_DROF_FAST_OPM_2	11x10	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_DROF_IP_TQ_DROF_1	11	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	[rpm]
LDPM_N_GRD_IP_TQ_DROF_1	10	0...FFH	-4.096E+3 ... 4.064E+3	32	[rpm/s]
Torque for driving off assistance, basic value value in operation mode 2					
IP_TQ_DROF_SLOW_OPM_1	11x10	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_DROF_IP_TQ_DROF_1	11	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	[rpm]
LDPM_N_GRD_IP_TQ_DROF_1	10	0...FFH	-4.096E+3 ... 4.064E+3	32	[rpm/s]
Torque for driving off assistance, basic value in operation mode 1					
IP_TQ_DROF_SLOW_OPM_2	11x10	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_DROF_IP_TQ_DROF_1	11	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	[rpm]
LDPM_N_GRD_IP_TQ_DROF_1	10	0...FFH	-4.096E+3 ... 4.064E+3	32	[rpm/s]
Torque for driving off assistance, basic value in operation mode 2					

22.1.1 FUNCTION DESCRIPTION:

The function “Driving off assistance” shall prevent the engine from stalling at driving off. Therefore an extra torque reserve is put at disposal if the necessity is given. This is shown by LV_N_SP_IS_CS = 1.defined in Engine speed setpoint calculation

In addition, an extra torque request is calculated under individual conditions. While this extra torque is effective, the idle speed controller can be suppressed or not via LC_TQ_DROF_ISC_OFF.

Application Condition

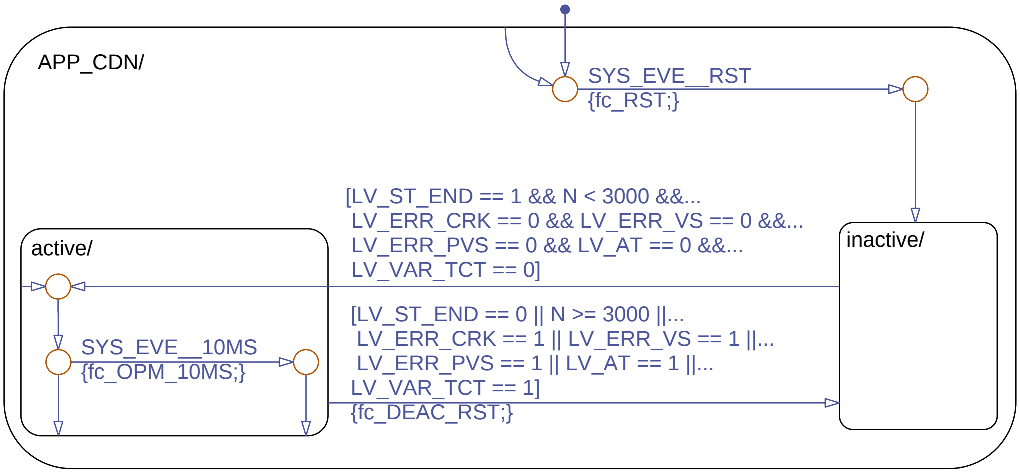
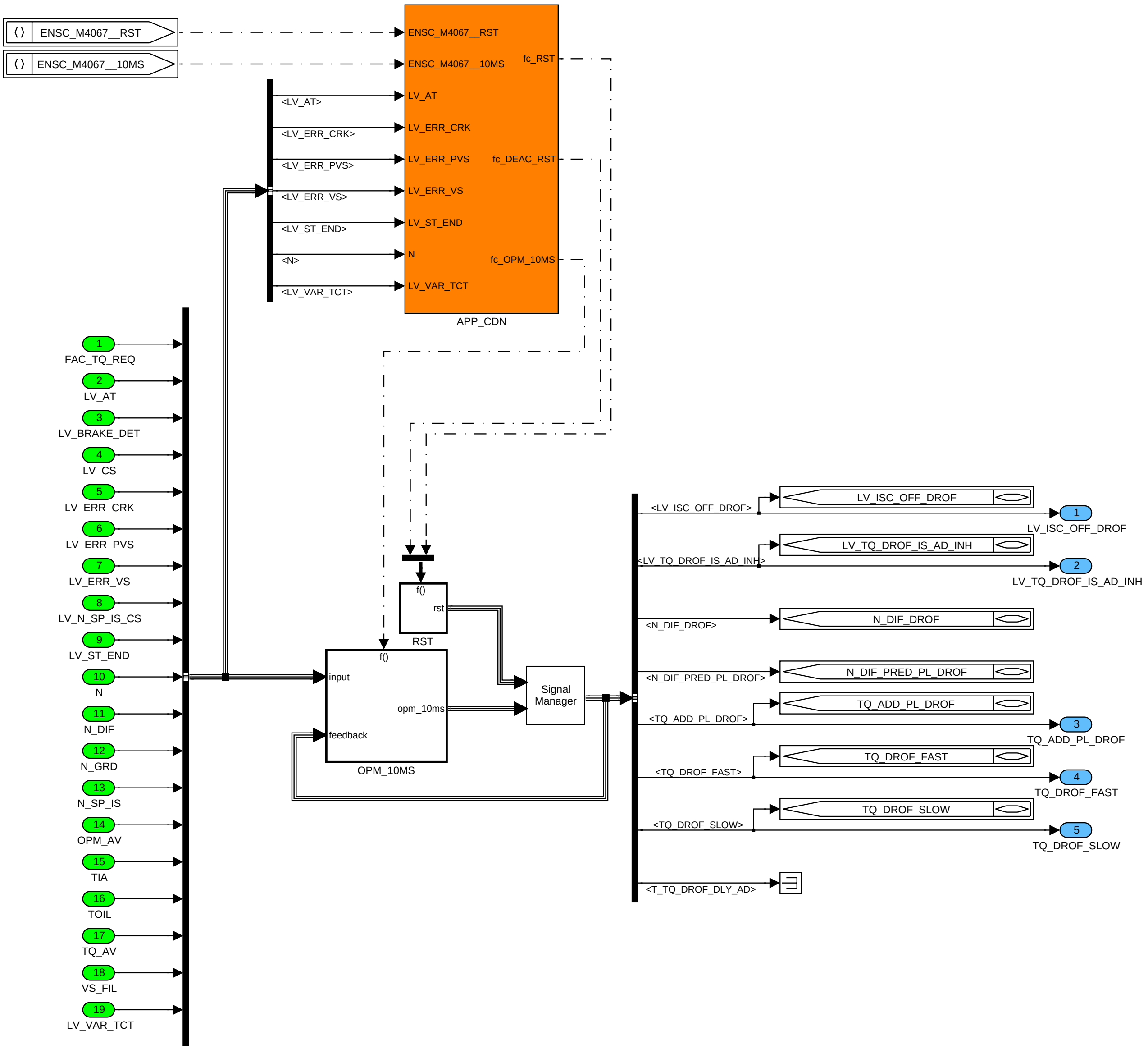


Figure 1 ENSC_M4067/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 18−Oct−2006

Figure 2 ENSC_M4067

22.1.1.1 Initialisation:

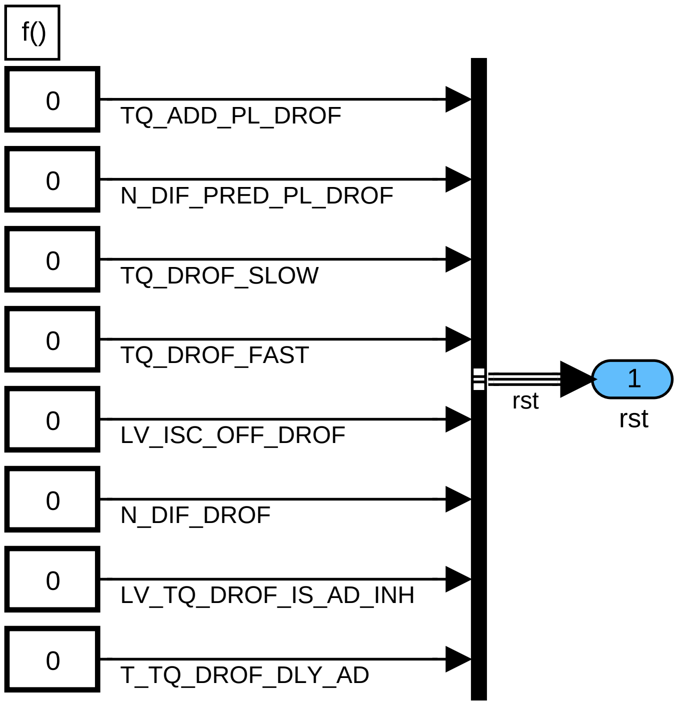


Figure 3 ENSC_M4067/ RST

22.1.1.2 Function Overview:

Calculation of additive & predicted drive off:

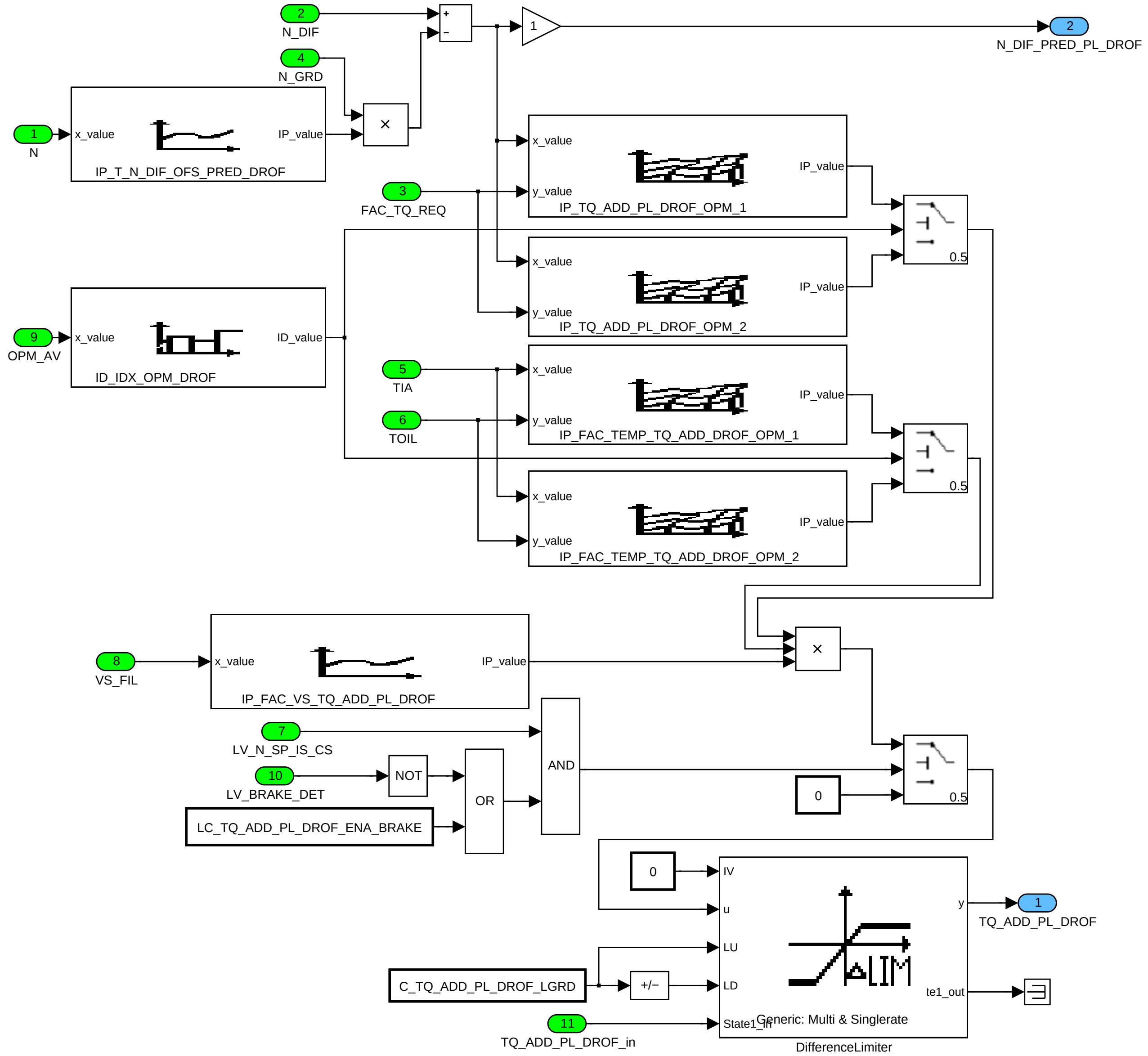


Figure 4 ENSC_M4067/ OPM_10MS/ CLC_TQ_ADD_PL_DROF

Determination of TQ_DROF_FAST, TQ_DROF_SLOW and LV_ISC_OFF_DROF:

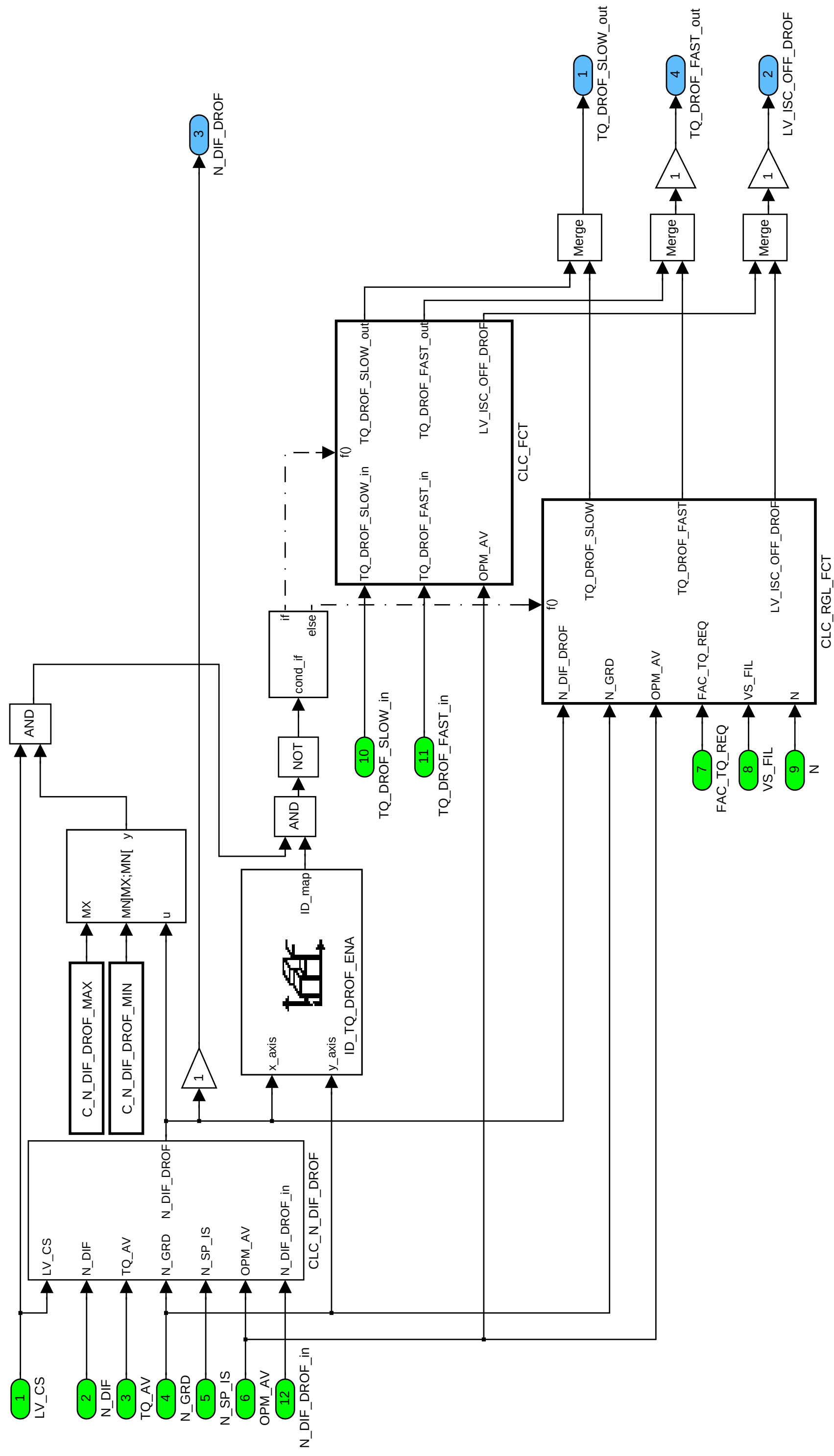


Figure 5 ENSC_M4067/ OPM_10MS/ CLC_INH_FAST_SLOW_DROF

Calculation of N_DIF_DROF:

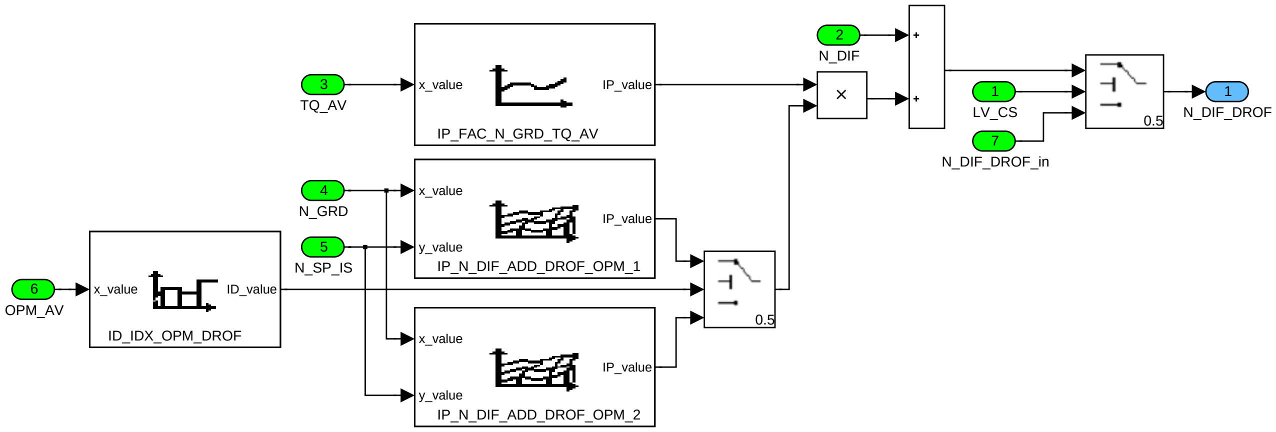


Figure 6 ENSC_M4067/ OPM_10MS/ CLC_INH_FAST_SLOW_DROF/ CLC_N_DIF_DROF

Function termination:

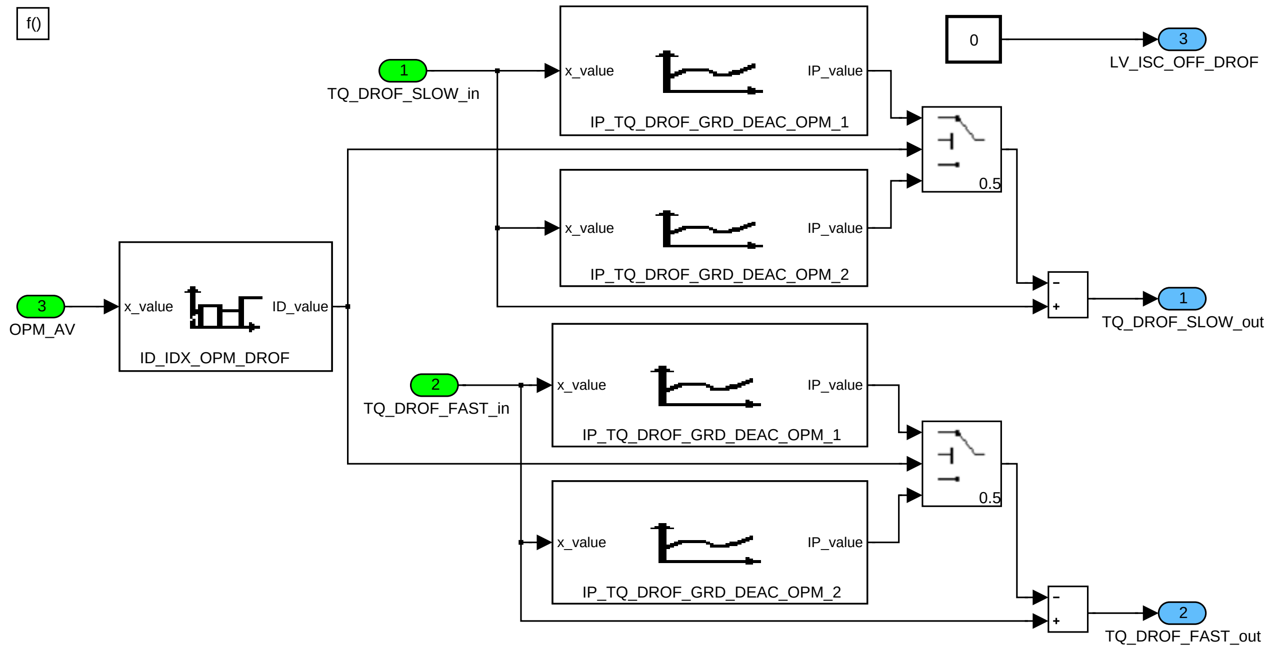


Figure 7 ENSC_M4067/ OPM_10MS/ CLC_INH_FAST_SLOW_DROF/ CLC_FCT

general specification

Regular function:

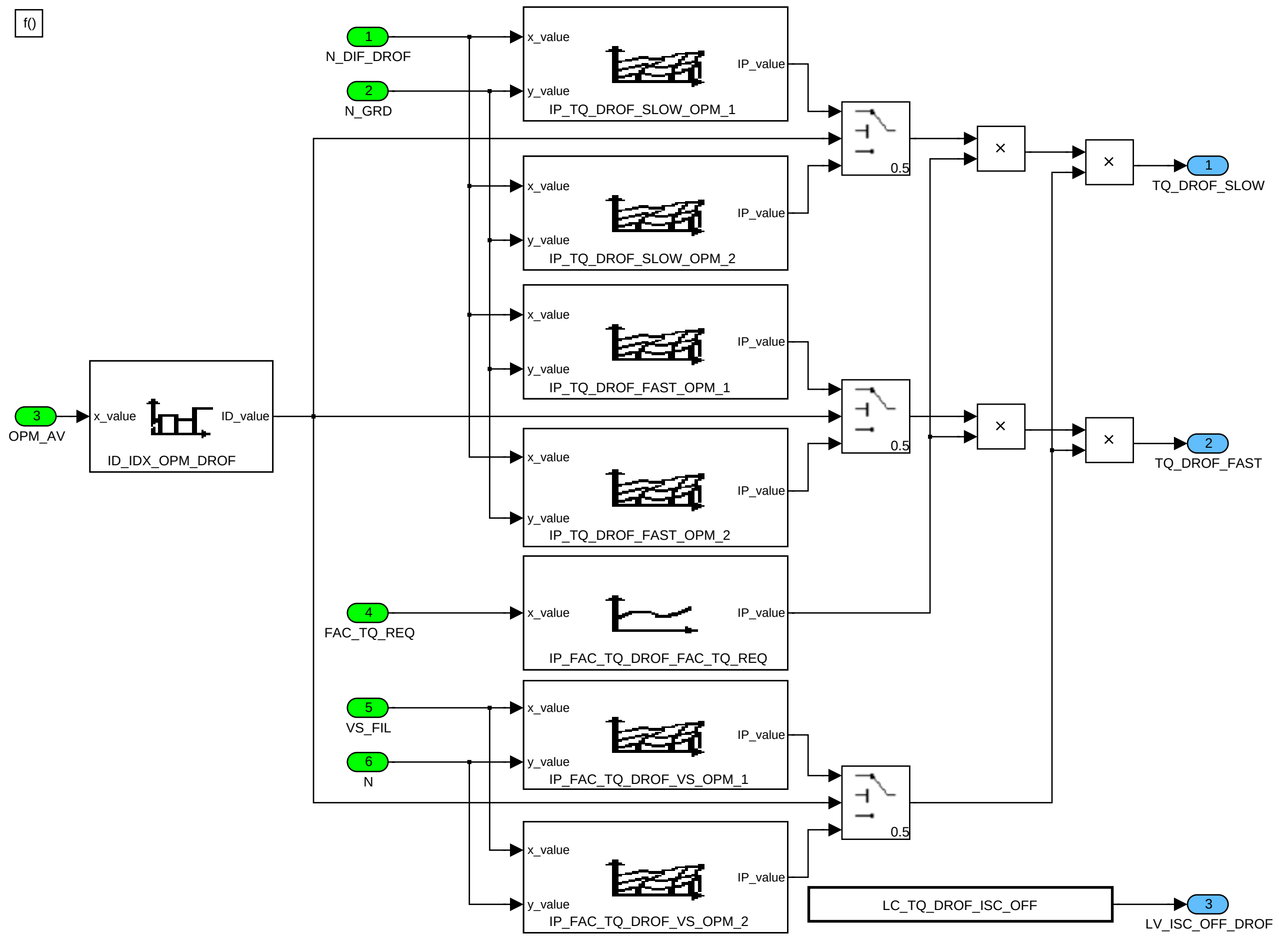


Figure 8 ENSC_M4067/ OPM_10MS/ CLC_INH_FAST_SLOW_DROF/ CLC_RGL_FCT

Condition for initialisation of IS control adaptations:

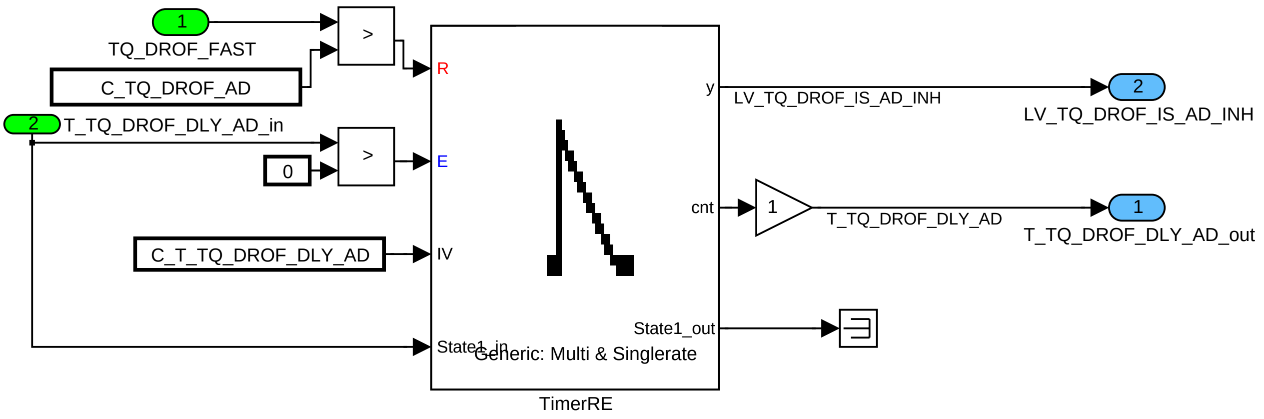


Figure 9 ENSC_M4067/ OPM_10MS/ CLC_IS_AD

22.2 Engine speed setpoint calculation

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_N_SP_IS_CS	O/V	0... 1H	0... 1	1	[-]
1: conditions for increased idle speed / torque reserve for drive-off-support fulfilled					
LV_N_SP_IS_PSTE	O/V	0... 1H	0... 1	1	[-]
1: conditions for increased idle speed / torque reserve for power steering					
LV_N_SP_IS_PSTE_2	O/V	0... 1H	0... 1	1	[-]
1: conditions for increased idle speed / torque reserve for power steering for second VS threshold					
LV_REQ_HEAT_N_SP_IS	V	0... 1H	0... 1	1	[-]
1: idle Speed setpoint required from heating control unit					
N_DIF	O/V	8000... 7FFFH	-32768... 32767	1	[rpm]
Engine speed deviation N_SP_IS-N					
N_DIF_COR	O/V	8000... 7FFFH	-32768... 32767	1	[rpm]
Idle speed control variable					
N_DIF_MMV	O/V	8000... 7FFFH	-32768... 32767	1	[rpm]
N_DIF moving mean value					
N_DIF_ST	O/V	8000... 7FFFH	-32768... 32767	1	[rpm]
Engine speed deviation N_MAX_TOL_ST - N					
N_MAX_TOL_ST	O/V	0... FFH	0... 8160	32	[rpm]
Top level end of start engine speed					
N_SP_IS	O/V	0... 1FE0H	0... 8160	1	[rpm]
Idle Speed Setpoint					
N_SP_IS_2	O/V	0... 1FE0H	0... 8160	1	[rpm]
basis idle speed setpoint					
N_SP_IS_3	V	0... 1FE0H	0... 8160	1	[rpm]
greatest of all idle speed setpoints					
N_SP_IS_CS	O/V	0... 1FE0H	0... 8160	1	[rpm]
Idle-speed setpoint given by DROF					
N_SP_IS_POW	V	0... 1FE0H	0... 8160	1	[rpm]
active nominal idle speed setpoint by power management active					
N_SP_IS_PWR_STAB	V	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint for power supply stability					
N_SP_IS_RATIO	O/V	0... FFFFH	0... 7.999878	122.07e-6	[-]
Ratio between actual engine speed and engine speed setpoint					
N_SP_IS_TCT	V	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoints for TCT					

Input Data:

ANG_PSTE	CTR_KM_CAN	C_N_MAX_KWP	C_VS_MAX_KWP
GEAR_INFO	ID_IDX_OPM_DROF	IGA_MV_ADJ_KNK_CUS	LV_ACIN
LV_ACT_N_SP_IS_BAS_EXT_ADJ	LV_ACT_N_SP_IS_EXT_ADJ	LV_ACT_SA_EOL	LV_AT
LV_CH_N_SP_IS	LV_CS	LV_DLY_N_SP_IS	LV_DRI
LV_IS	LV_NT_SO2P_EXT_ADJ_ACT	LV_N_SP_IS_LIH_ACT	LV_N_SP_IS_POW_ACT
LV_N_SP_IS_PWR_STAB	LV_PL	LV_PSTE_2_DISABLE	LV_PSTE_DISABLE
LV_REQ_HEAT	LV_VAR_ACIN	LV_VAR_ARS	LV_VAR_PSTE_2
LV_VAR_TCT	N	N_FAST	N_SP_IS_BRAKE
N_SP_IS_CH	N_SP_IS_EXT_ADJ	N_SP_IS_POIL_CTL	N_SP_IS_POW_1
N_SP_IS_TCT_CAN	N_SP_OFS_KWP	TAM	TCO
TOIL	T_AST	VEL_ANG_PSTE	VS
TQ_LOSS	OPM_AV	LV_ST_END	LV_VAR_PSTE_3

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ANG_EPS_ISC	1	0... 7FFFH	0... 1439.946	0.043945	[°STW]
Minimum ANG_PSTE for N_SP_EPS					
C_ANG_PSTE_ISC	1	0... 7FFFH	0... 1439.946	0.043945	[°STW]
Minimum ANG_PSTE for N_SP_PSTE					
C_DRI_VS_MIN_VB	1	0... FFH	0... 255	1	[km/h]
Minimum vehicle speed for activating battery charging in case LV_DRI = 1					
C_IGA_ADJ_HYS_N_SP_IS	1	0... 80H	-48... 0	0.375	[°CRK]
Spark retard hysteresis for idle speed setpoint N_SP_IS_KNK					
C_IGA_ADJ_THD_N_SP_IS	1	0... 80H	-48... 0	0.375	[°CRK]
Spark retard threshold for activating idle speed setpoint N_SP_IS_KNK					
C_N_DIF_CRLC	1	0... FFH	0... 0.9960938	3.9063e-3	[-]
Correlation factor for N_DIF_MMV calculation					
C_N_DIF_FAC	1	0... FFH	0... 0.9960938	3.9063e-3	[-]
Multiplicative factor for N_DIF_COR calculation					
C_N_DIF_MIN_MMV	1	F010... 0H	-4080... 0	1	[rpm]
Minimum filtering constant					
C_N_SP_DEC_CS	1	1... FFH	0.0625... 15.9375	0.0625	[rpm]
Decrement engine speed setpoint for drive-off-support					
C_N_SP_DEC_IS_CH_DRI	1	1... FFH	0.0625... 15.9375	0.0625	[rpm/10 ms]
Idle speed change limitation during catalyst heating and LV_DRI = 1					
C_N_SP_DEC_PL	1	1... FFH	0.0625... 15.9375	0.0625	[rpm]
Decrement idle speed setpoint for part load					
C_N_SP_DEC_PSTE	1	1... FFH	0.0625... 15.9375	0.0625	[rpm]
C_N_SP_DEC_PSTE					
C_N_SP_HEAT_AT	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed at heater request, automatic transmission vehicle					
C_N_SP_HEAT_MT	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed at heater request, manual transmission vehicle					
C_N_SP_INC_CS	1	1... FFH	0.0625... 15.9375	0.0625	[rpm]
Increment engine speed setpoint for drive-off-support					
C_N_SP_INC_PL	1	1... FFH	0.0625... 15.9375	0.0625	[rpm]
Increment idle speed setpoint for part load					
C_N_SP_INC_PSTE	1	1... FFH	0.0625... 15.9375	0.0625	[rpm]
Increment engine speed setpoint for power steering					
C_N_SP_IS_ARS_AT	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint ARS, automatic transmission vehicle					
C_N_SP_IS_ARS_MT	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint ARS, manual transmission vehicle					
C_N_SP_IS_PSTE_2_AT	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint PSTE_2, automatic transmission vehicle					
C_N_SP_IS_PSTE_2_MT	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint PSTE_2, manual transmission vehicle					
C_N_SP_IS_PWR_STAB	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint for power supply stability					
C_N_SP_IS_PWR_STAB_DRI	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint for power supply stability and driving position engaged					
C_N_SP_IS_SA_EOL	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint during the "secondary air function end of line test"					
C_N_SP_LGRD_IS	1	1... FFH	0.0625... 15.9375	0.0625	[rpm/10 ms]
Nominal idle speed change limitation					

C_N_SP_LGRD_IS_LIH	1	1... FFH	0.25... 63.75	0.25	[rpm/10 ms]
Minimum filtering constant					
C_N_SP_OFS_KWP_MAX	1	0... 7FFFH	0... 32767	1	[rpm]
Upper threshold for tester offset					
C_N_SP_OFS_KWP_MIN	1	8000... FFFFH	-32768... -1	1	[rpm]
C_N_SP_OFS_KWP_MIN					
C_NT_SO2P_EXT_ADJ_ACT	1	0... 1FE0H	0... 8160	1	[rpm]
idle speed setpoints for SO2P in service					
C_T_ANG_DLY_N_SP_DEC_PSTE	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for decrement engine speed setpoint for power steering angle and velocity					
C_T_DLY_GRD_LIM	1	0... 7FFH	0... 20.47	0.01	[s]
Delay time of N_SP_IS gradient limitation					
C_T_VS_DLY_N_SP_DEC_PSTE	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for decrement engine speed setpoint for power steering vehicle speed					
C_TAM_HYS	1	0... 40H	0... 48	0.75	[°C]
Hysteresis for ambient temperature threshold					
C_TAM_N_SP_HEAT	1	0... FEH	-48... 142.5	0.75	[°C]
Ambient temperature threshold idle speed at heater request					
C_TAM_N_SP_IS_KNK	1	0... FEH	-48... 142.5	0.75	[°C]
Ambient temperature for N_SP_IS_KNK calculation					
C_TOIL_HYS_N_SP_IS	1	0... C8H	-40... 160	1	[°C]
TOIL hysteresis for idle speed setpoint					
C_TOIL_THD_N_SP_IS	1	0... C8H	-40... 160	1	[°C]
C_TOIL_THD_N_SP_IS					
C_VEL_ANG_EPS_ISC	1	0... 7FFFH	0... 1439.946	0.043945	[°STW/s]
Minimum VEL_ANG_PSTE for N_SP_EPS					
C_VEL_ANG_PSTE_ISC	1	0... 7FFFH	0... 1439.946	0.043945	[°STW/s]
Minimum VEL_ANG_PSTE for N_SP_PSTE					
C_VS_HYS_N_SP_CS	1	0... FFH	0... 255	1	[km/h]
Vehicle speed hysteresis for VS threshold drive-off-support					
C_VS_MAX_N_SP_CS	1	0... FFH	0... 255	1	[km/h]
Maximum vehicle speed for drive-off-support					
C_VS_MAX_N_SP_IS_ARS	1	0... FFH	0... 255	1	[km/h]
Maximum vehicle speed for N_SP_IS_ARS					
C_VS_MIN_N_SP_EPS	1	0... FFH	0... 255	1	[km/h]
Minimum vehicle speed threshould for N_SP_EPS					
C_VS_MIN_N_SP_IS_ARS	1	0... FFH	0... 255	1	[km/h]
Minimum vehicle speed for N_SP_IS_ARS					
C_VS_MIN_N_SP_PSTE	1	0... FFH	0... 255	1	[km/h]
Minimum vehicle speed threshould for N_SP_PSTE					
C_VS_MIN_N_SP_PSTE_2	1	0... FFH	0... 255	1	[km/h]
Minimum vehicle speed threshould for N_SP_PSTE_2					
ID_N_SP_IS_KNK_AT	4	0... 1FE0H	0... 8160	1	[rpm]
LDPM_IGA_N_SP_IS_KNK [4]	4	0... 80H	-48... 0	0.375	[°CRK]
Minimum idle speed for spark retard due to knocking with AT					
ID_N_SP_IS_KNK_MT	4	0... 1FE0H	0... 8160	1	[rpm]
LDPM_IGA_N_SP_IS_KNK [4]	4	0... 80H	-48... 0	0.375	[°CRK]
Minimum idle speed for spark retard due to knocking with MT					
IP_ACIN_N_SP_IS	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TCO_N_SP_1 [8]	8	0... FEH	-48... 142.5	0.75	[°C]
Nominal idle speed with air conditioner switched on					
IP_DRI_ACIN_N_SP_IS	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TCO_N_SP_1 [8]	8	0... FEH	-48... 142.5	0.75	[°C]
Nominal idle speed with DRIVE (A/T) engaged and air conditioner switched on					
IP_DRI_N_SP_IS	8	0... 1FE0H	0... 8160	1	[rpm]

LDPM_TCO_N_SP_1 [8]	8	0... FEH	-48... 142.5	0.75	[°C]
Nominal idle speed with DRIVE (A/T) engaged					
IP_DRI_N_SP_IS_LIH	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TCO_N_SP_1 [8]	8	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint in case of limp home and engaged drivetrain					
IP_FAC_N_SP_OFS_KWP	8	0... FFH	0... 0.9960938	3.9063e-3	[-]
LDPM_TCO_N_SP_1 [8]	8	0... FEH	-48... 142.5	0.75	[°C]
TCO factor for N_SP_OFS_KWP					
IP_N_MAX_TOL_ST	9	0... FFH	0... 8160	32	[rpm]
LDPM_TCO_9 [9]	9	0... FEH	-48... 142.5	0.75	[°C]
Top level end of start engine speed					
IP_N_SP_IS	6	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TCO_1_ENSC [6]	6	0... FEH	-48... 142.5	0.75	[°C]
Nominal idle speed without additional load on the engine					
IP_N_SP_IS_CS_OPM_1	8	0... 1FE0H	0... 8160	1	[rpm]
LDP_TOIL_IP_N_SP_IS_CS_OPM_1 [8]	8	0... C8H	-40... 160	1	[°C]
Idle speed setpoint for drive-off-support with manual transmission in operation mode 1					
IP_N_SP_IS_CS_OPM_2	8	0... 1FE0H	0... 8160	1	[rpm]
LDP_TOIL_IP_N_SP_IS_CS_OPM_1 [8]	8	0... C8H	-40... 160	1	[°C]
Idle speed setpoint for drive-off-support with manual transmission in operation mode 2					
IP_N_SP_IS_EPS	6	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TCO_N_SP_IS_PSTE [6]	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for power steering variant EPS					
IP_N_SP_IS_LIH	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TCO_N_SP_1 [8]	8	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint in case of limp home					
IP_N_SP_IS_PL_OPM_1	6x6	0... 1FE0H	0... 8160	1	[rpm]
LDP_N_N_SP_IS_PL_OPM_1 [6]	6	0... 1FE0H	0... 8160	1	[rpm]
LDP_TQ_LOSS_N_SP_IS_PL_OPM_1 [6]	6	0... FFFFH	-1024... 1023.969	0.03125	[Nm]
Nominal idle speed at PL in operation mode 1					
IP_N_SP_IS_PL_OPM_2	6x6	0... 1FE0H	0... 8160	1	[rpm]
LDP_N_N_SP_IS_PL_OPM_1 [6]	6	0... 1FE0H	0... 8160	1	[rpm]
LDP_TQ_LOSS_N_SP_IS_PL_OPM_1 [6]	6	0... FFFFH	-1024... 1023.969	0.03125	[Nm]
Nominal idle speed at PL in operation mode 2					
IP_N_SP_IS_PL_TCT_OPM_1	6x6	0... 1FE0H	0... 8160	1	[rpm]
LDP_N_N_SP_IS_PL_TCT_OPM_1 [6]	6	0... 1FE0H	0... 8160	1	[rpm]
LDP_TQ_LOSS_N_SP_IS_PL_TCT_OPM1 [6]	6	0... FFFFH	-1024... 1023.969	0.03125	[Nm]
Nominal idle speed at PL variant TCT in operation mode 1					
IP_N_SP_IS_PL_TCT_OPM_2	6x6	0... 1FE0H	0... 8160	1	[rpm]
LDP_N_N_SP_IS_PL_TCT_OPM_1 [6]	6	0... 1FE0H	0... 8160	1	[rpm]
LDP_TQ_LOSS_N_SP_IS_PL_TCT_OPM1 [6]	6	0... FFFFH	-1024... 1023.969	0.03125	[Nm]
Nominal idle speed at PL variant TCT in operation mode 2					
IP_N_SP_IS_PSTE	6	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TCO_N_SP_IS_PSTE [6]	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for power steering					
IP_N_SP_IS_PSTE_2	6	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TCO_N_SP_IS_PSTE [6]	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for power steering for VS >= C_VS_MIN_N_SP_PSTE_2					
IP_N_SP_IS_TOIL_AT	4	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TOIL_N_SP_IS_TOIL [4]	4	0... C8H	-40... 160	1	[°C]
Minimum idle speed for TOIL with AT					
IP_N_SP_IS_TOIL_MT	4	0... 1FE0H	0... 8160	1	[rpm]
LDPM_TOIL_N_SP_IS_TOIL [4]	4	0... C8H	-40... 160	1	[°C]
Minimum idle speed for TOIL with MT					
IP_T_AST_MIN_N_SP_IS_CS	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_N_SP_1 [8]	8	0... FEH	-48... 142.5	0.75	[°C]
Minimum delay time after start for drive-off-support idle speed					

LC_IS_N_SP_CS_DIS	1	0... 1H	0... 1	1	[-]
LC to switch on/off LV_IS condition for drive-off-support					

Configuration Data:

Name	Mode	Hex.Limits	Phys.Limits	Resol.	Unit
NC_N_DIF_MIN_CRLC	1	F010... 0H	-4080... 0	1	[rpm]
Non calibrated correlation factor for N_DIF_MMV calculation					

General Information

Application Conditions

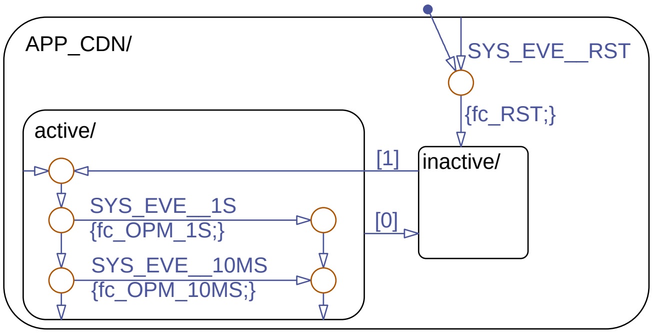
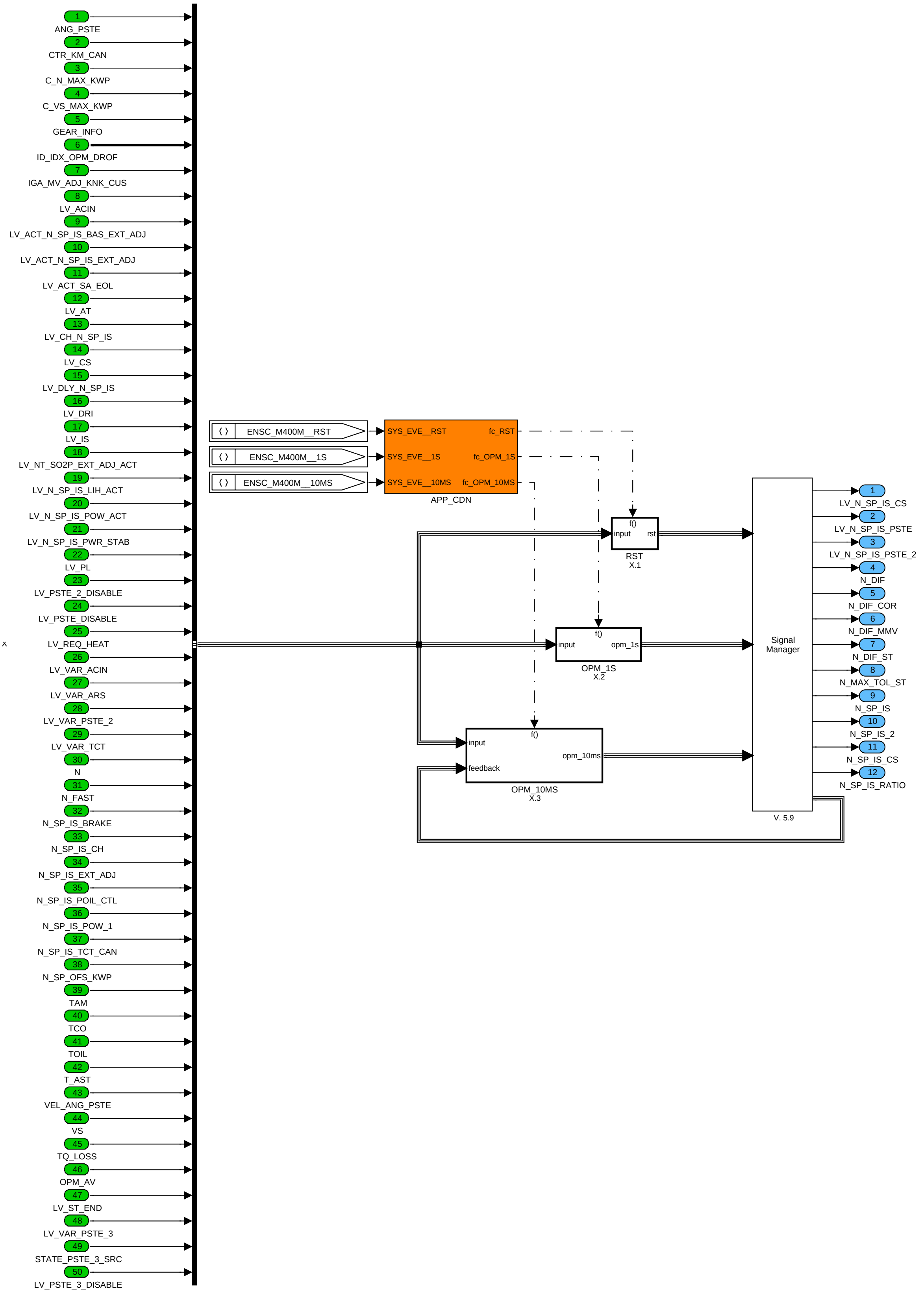


Figure 10:
Path: ENSC_M400M/APP_CDN/Chart



SDA_SRS / SDA V 5.0.2 19−Oct−2006

Figure 11:
Path: ENSC_M400M

Chapter Engine speed control	Baseline 4DC3940S		Include File 43400M01.00Q
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Pages
3699 of 9643

22.2.1 Initialisation at system event reset

All output variable are initialised at system event reset.

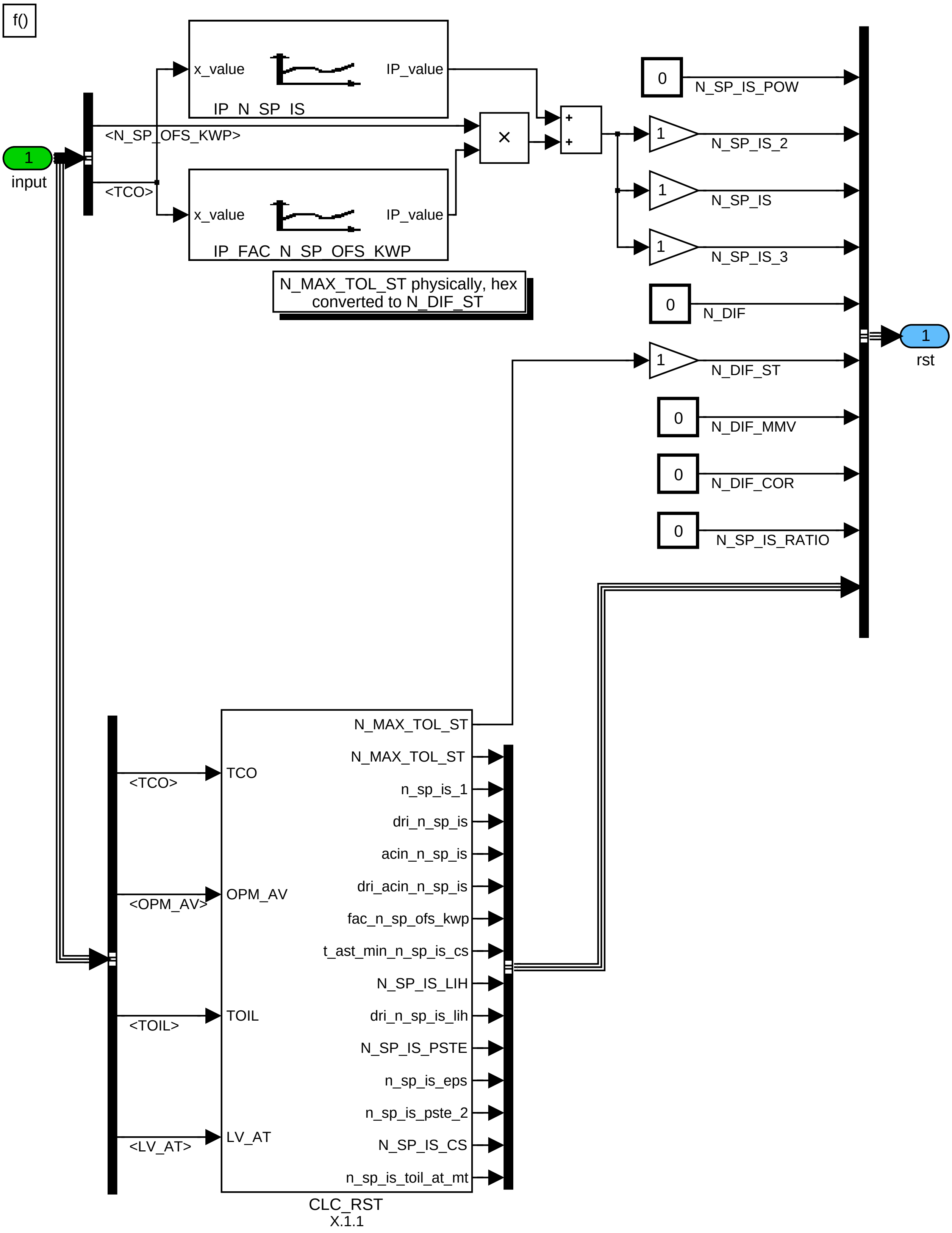


Figure 12:
Path: ENSC_M400M/RST

22.2.1.1 Initialisation of temperature dependent variables.

All the temperature dependent variables are initialised here from corresponding IP maps and curves.

general specification

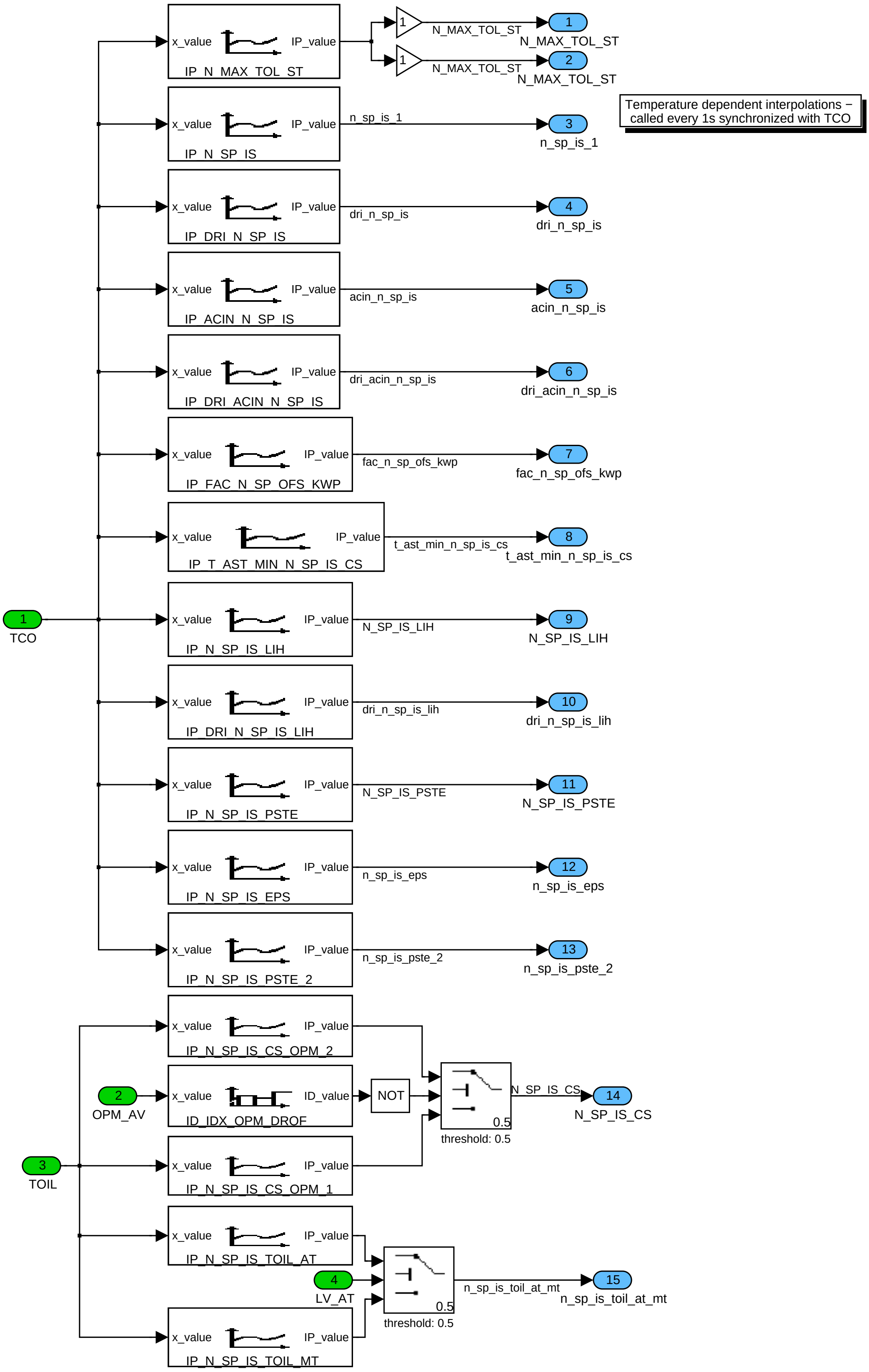


Figure 13:
Path: ENSC_M400M/RST/CLC_RST

Chapter	Baseline	Include File
Engine speed control	4DC3940S	43400M01.00Q

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Pages
3702 of 9643

22.2.2 Calculation at 1 second recurrence

22.2.2.1 Calculation of temperature dependent variables.

All the temperature dependent variables are calculated here from corresponding IP maps and curves.

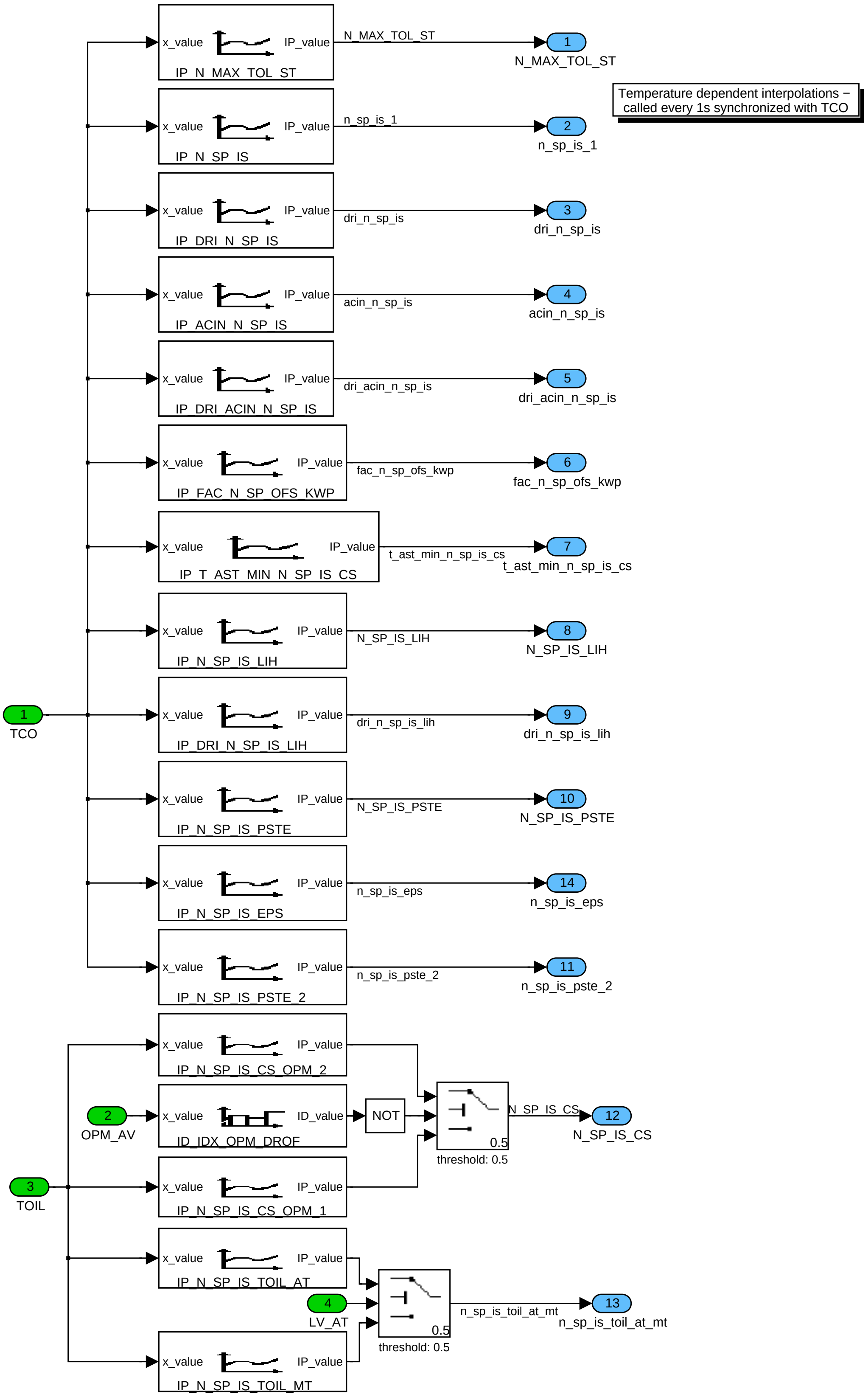


Figure 14:
Path: ENSC_M400M/OPM_1S/CLC_1S

Chapter	Baseline	Include File
Engine speed control	4DC3940S	43400M01.00Q

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Pages
3704 of 9643

22.2.3 Formula section.

22.2.3.1 Idle speed setpoint N_SP_IS

FUNCTION DESCRIPTION:
General information:
The input value LV_DLY_N_SP_IS (output from module Converter Torque) is derived from LV_DRI including a tuneable delay time. If drive is engaged the converter torque is build up after a short delay time (about 400 ms). If the change in engine speed setpoint for drive engaged also is delayed the new setpoint can be adjusted more stable.
In general the transition from one idle speed setpoint to another is performed with adjustable change limitations. The only exception is: no change limitation before engine start has finished (LV_ST_END = 0). This is to ensure the desired engine speed setpoint immediately after start.

22.2.3.1.1 Nominal idle speed with tester offset(Calculation of N_SP_IS_2)

The nominal idle speed setpoint calculation takes into consideration air conditioning compressor ACC on/off, drivetrain engaged (LV_DLY_N_SP_IS) and offset by tester via KWP.
Table of nominal idle speed N_SP_IS_1, dependant from ACC and drivetrain engaged:

LV_ACIN	LV_DLY_N_SP_IS	N_SP_IS_1
0	0	IP_N_SP_IS
0	1	IP_DRI_N_SP_IS
1	0	IP_ACIN_N_SP_IS
1	1	IP_DRI_ACIN_N_SP_IS

The tester offset is multiplied by a factor IP_FAC_N_SP_OFS_KWP which is TCO dependant and limited by upper and lower thresholds.
N_SP_IS_2 is calculated here depending upon logical variable LV_ACIN and LV_DLY_N_SP_IS. Using these two variables the control input is generated using the lookup table. If both variable are zero then input from port 1 of multiport switch is passed, similarly the inputs at other ports are passed depending upon these two logical variable.

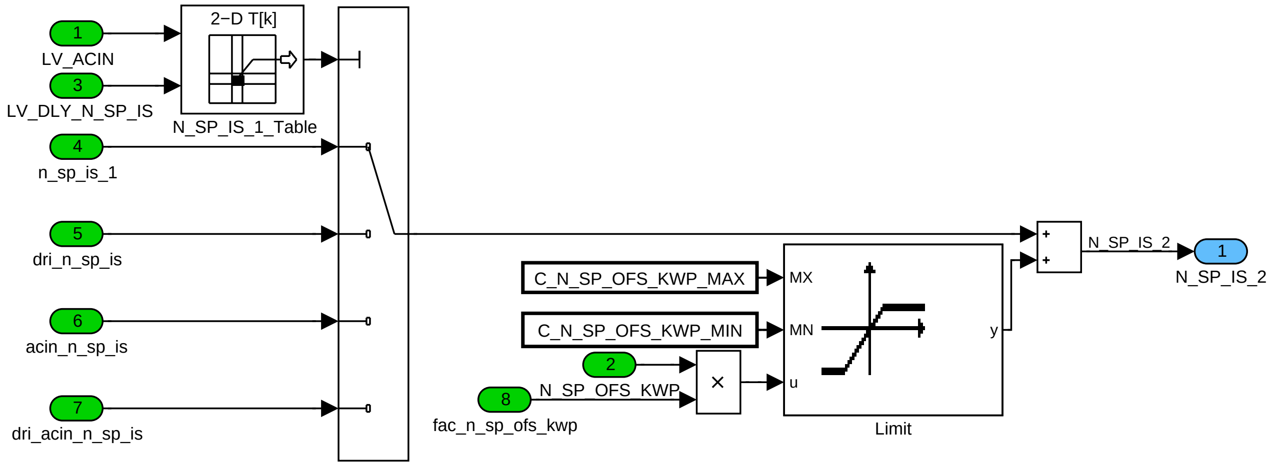


Figure 15:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_2

22.2.3.1.2 Pre calculation and calculation of N_SP_IS

22.2.3.1.2.1 Extraordinary engine speed setpoints(Calculation of first part of pre calculation.)

22.2.3.1.2.1.1 Idle speed setpoint due to Power Management

The engine speed setpoint N_SP_IS_POW_1 from Power Management is used if LV_N_SP_IS_POW_ACT = 1 and if it is the greatest of the speed setpoints entering the MAXselection

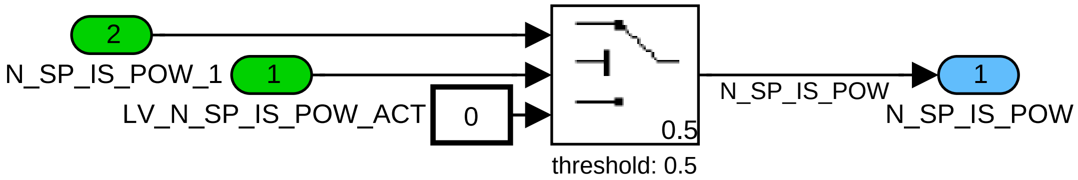


Figure 16:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_1/ISSP_DUE_TO_POW_MNG

22.2.3.1.2.1.2 Idle Speed setpoint required from heating control unit

The A/Cunit sets a requirement switch LV_REQ_HEAT to raise the idle speed in order to increase the heating power. If the vehicle is not equipped with A/C, LV_REQ_HEAT is made up with ambient temperature TAM. The idle speed setpoint C_N_SP_HEAT_MT/AT will be executed if LV_REQ_HEAT = 1 and if it is the greatest of the speed setpoints entering the MAXselection

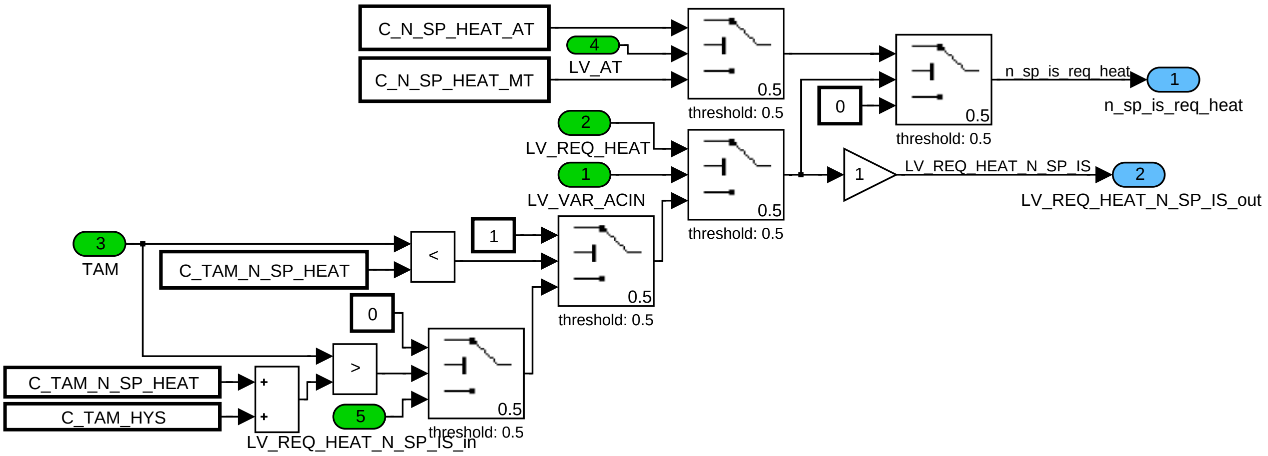


Figure 17:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_1/ISSP_DUE_TO_HEATING

22.2.3.1.2.1.3 Idle speed setpoint due to variant of power steering 2

The engine speed setpoint N_SP_IS_VAR_PSTE_2 depends on learnt variant LV_VAR_PSTE_2.

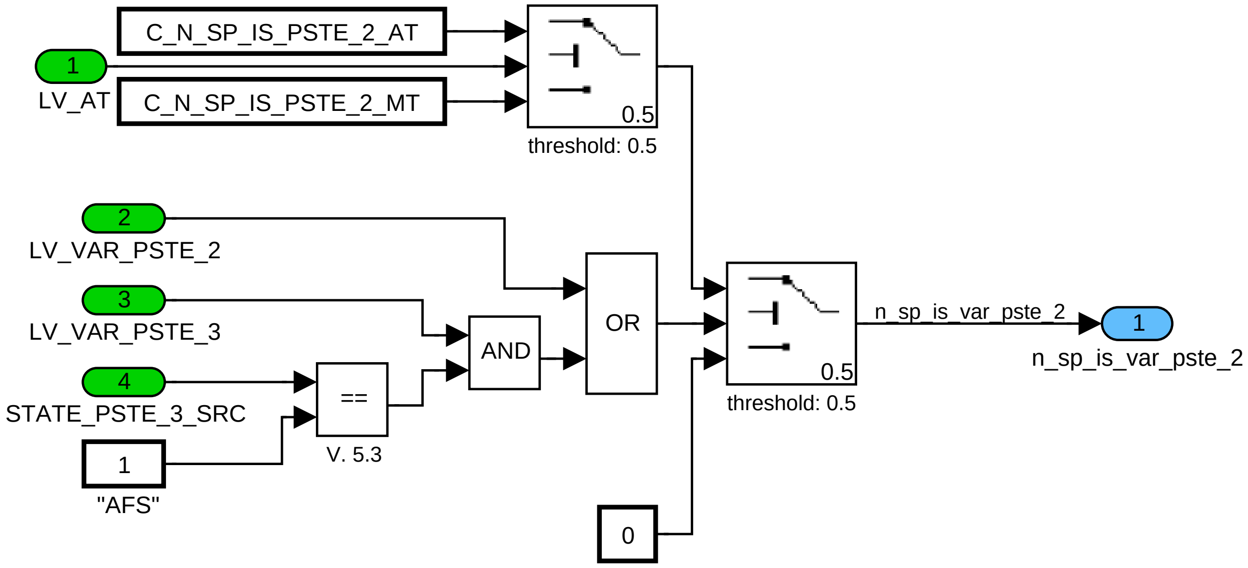


Figure 18:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_1/ISSP_DUE_TO_VAR_PSTE2

22.2.3.1.2.1.4 Idle speed setpoint due to variant of ARS

The engine speed setpoint N_SP_IS_ARS depends on learnt variant LV_VAR_ARS and vehicle speed.

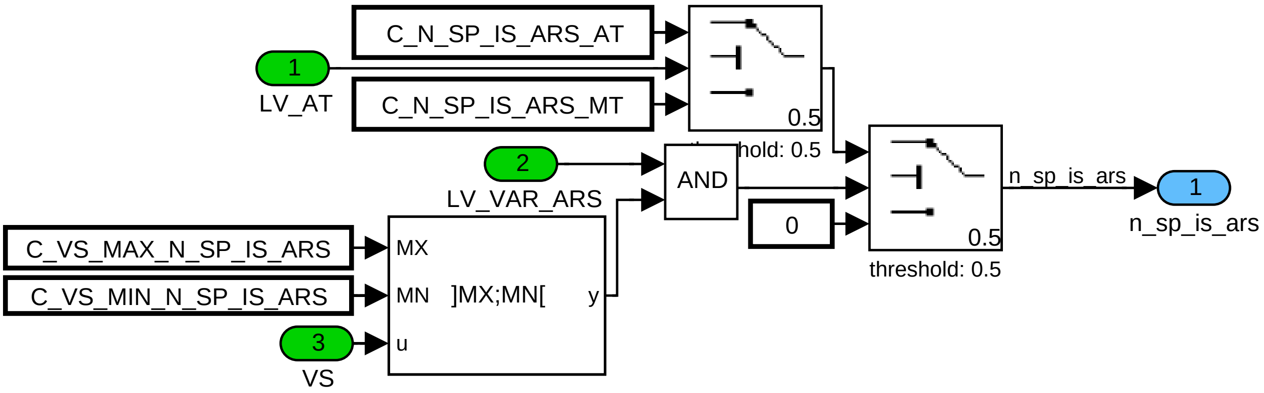


Figure 19:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_1/ISSP_DUE_TO_VAR_ARS

22.2.3.1.2.1.5 Idle Speed setpoint TOIL dependant

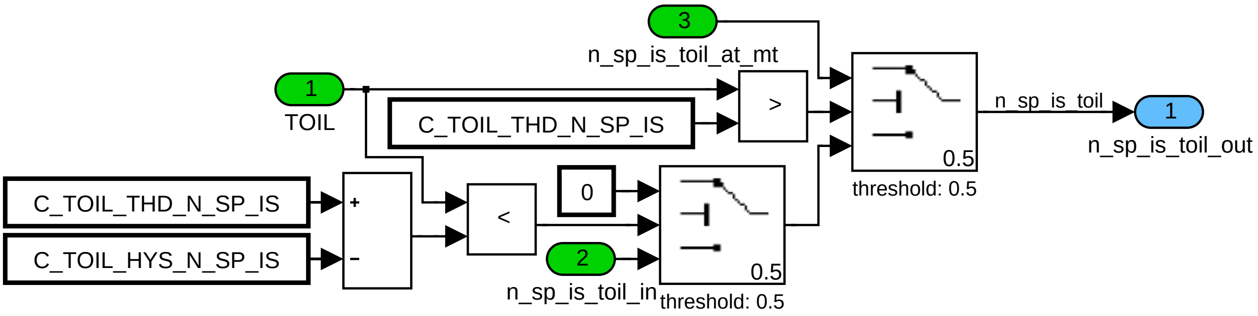


Figure 20:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_1/ISSP_TOIL

22.2.3.1.2.1.6 Idle Speed setpoint depending on average spark retard due to knocking

N_SP_IS takes care of the ignition spark retard due to knocking. Exceeding a calibrateable threshold an increased setpoint is active.

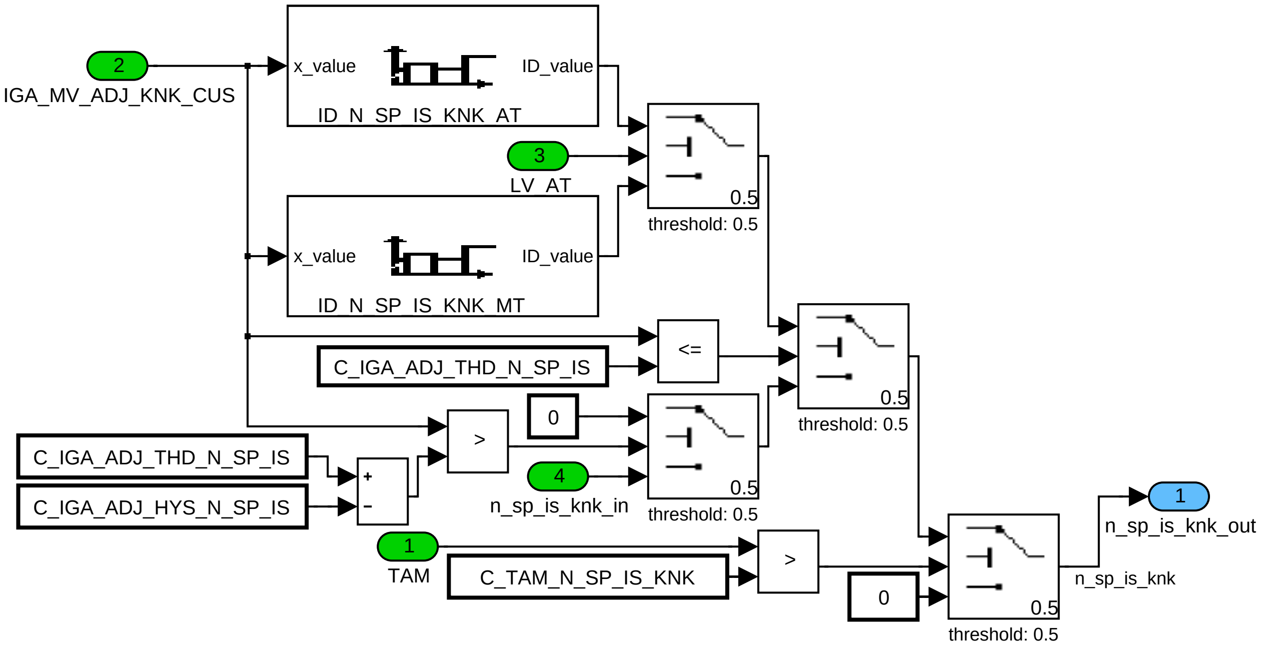


Figure 21:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_1/ISSP_AVG_SPARK_KNCK

22.2.3.1.2.1.7 Idle Speed setpoint at driveoffsupport

Absolute engine speed setpoint dependant from TOIL and transmission variant, used under certain conditions where vehicle driveoff is expected and if it is the greatest of the speed setpoints entering the MAXselection, see below. Individual gradient limitation C_N_SP_INC/DEC_CS.

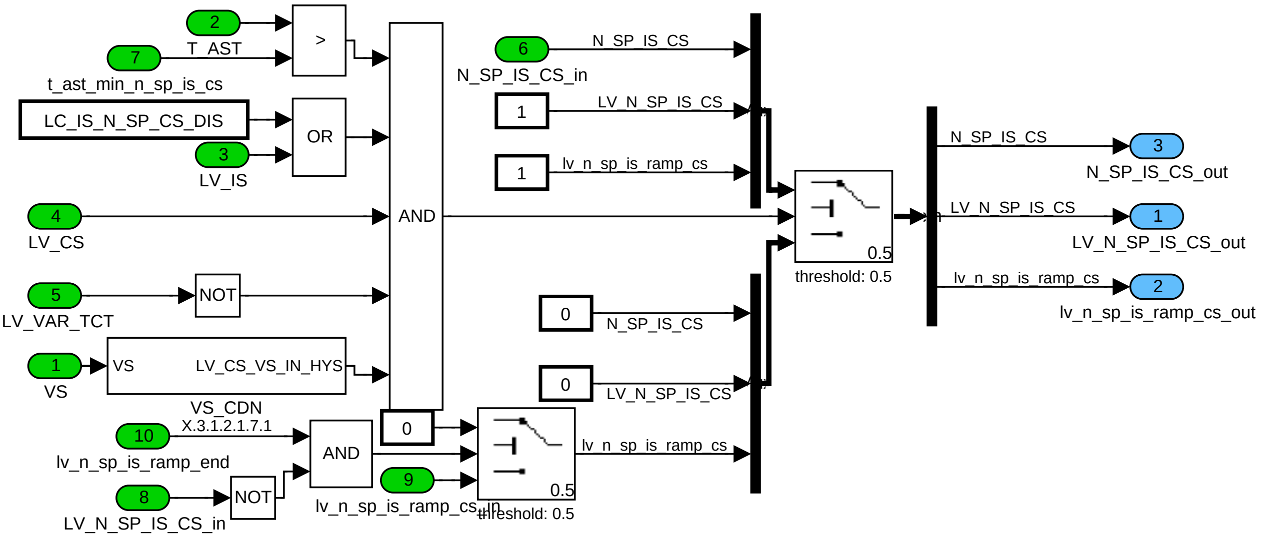


Figure 22:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_1/ISSP_DRIVE_SUPPORT

22.2.3.1.2.1.7.1 Calculation of logical variable

If VS is less than difference of maximum vehicle speed for drive of support (C_VS_MAX_N_SP_CS) and Vehicle speed hysteresis for VS threshold driveoffsupport

(C_VS_HYS_N_SP_CS) then generated LV_CS_VS_In_HYS_tmp is equal to 1 else if VS is greater than C_VS_MAX_N_SP_CS then . LV_CS_VS_In_HYS_tmp is set to 1 else it remains at old value.

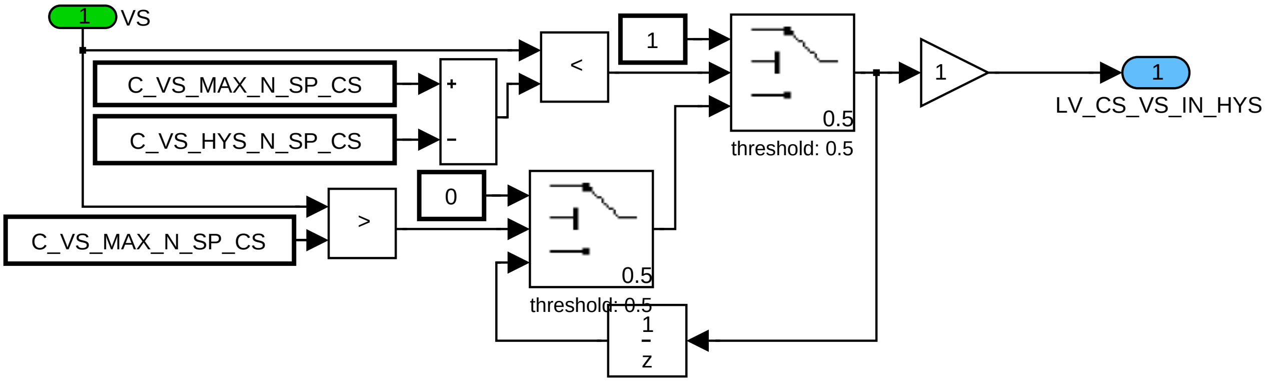


Figure 23:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_1/ISSP_DRIVE_SUPPORT/VS_CDN

22.2.3.1.2.2 Extraordinary engine speed setpoints(Calculation of second part of pre calculation.)

22.2.3.1.2.2.1 Idle Speed setpoint for power steering (EPS / PSTE / PSTE_2)

Absolute engine speed setpoint dependant from steering angle and steering angle velocity, used under certain conditions. If it is the greatest of the speed setpoints entering the MAXselection, see below.
For power steering variant EPS and AFS (ident: PSTE).

22.2.3.1.2.2.1.1 Idle speed setpoint for powersteering 1

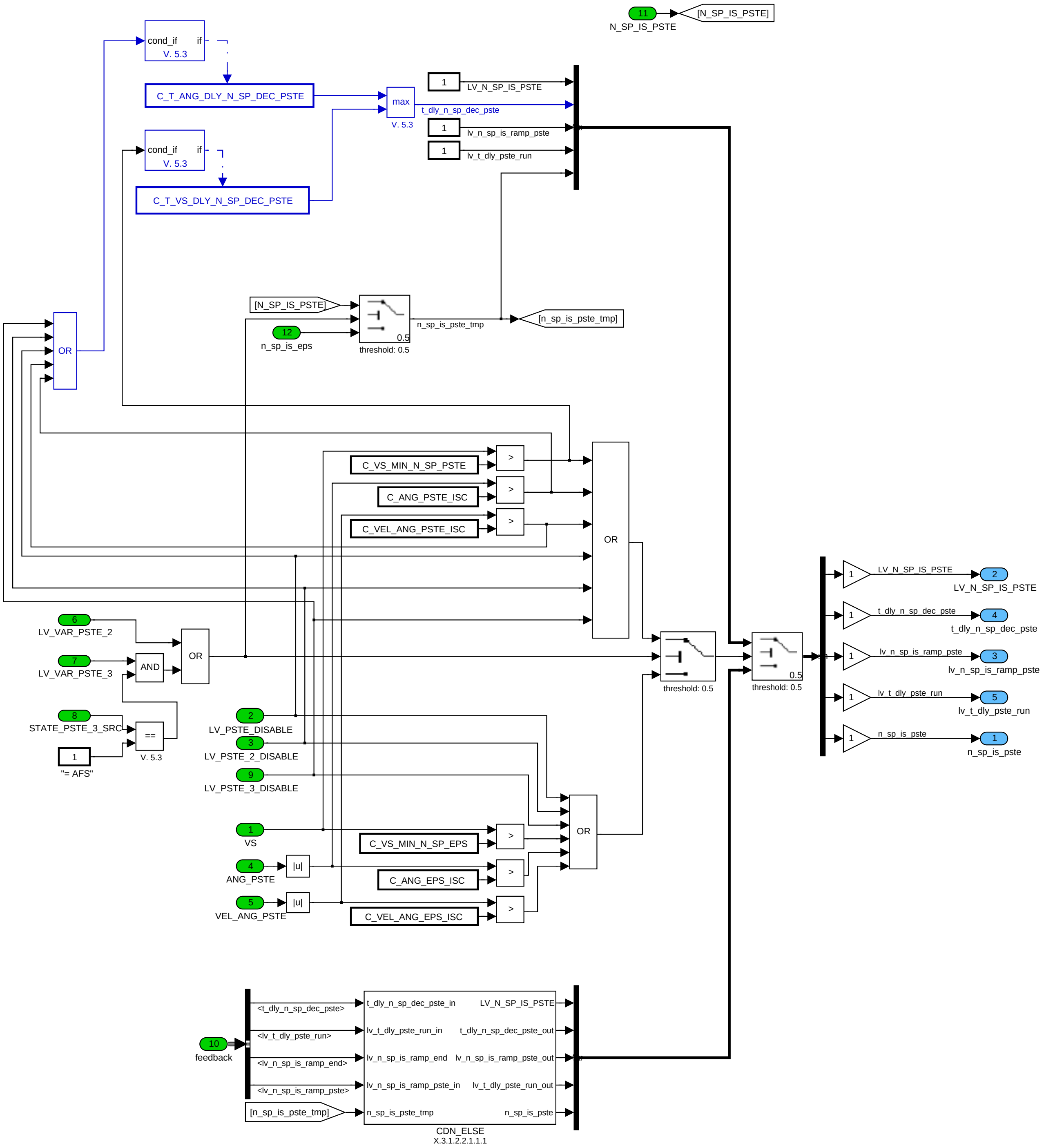


Figure 24:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_2/ISSP_FOR_PSTE/CDN_PSTE

22.2.3.1.2.2.1.1.1 Calculations when condition check fails.

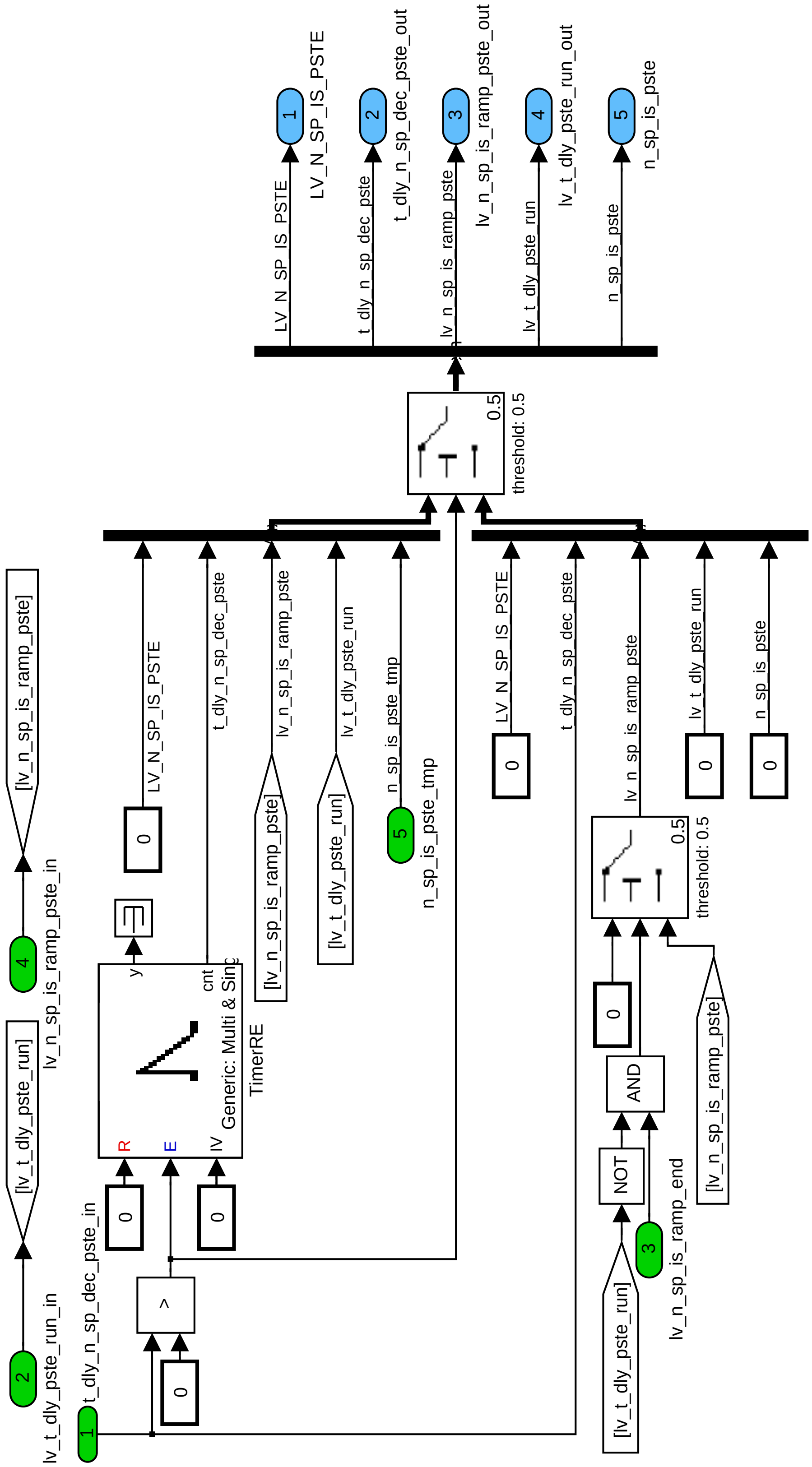


Figure 25:

Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_2/ISSP_FOR_PSTE/CDN_PSTE/CDN_ELSE

22.2.3.1.2.2.1.2 Idle speed setpoint for powersteering 2

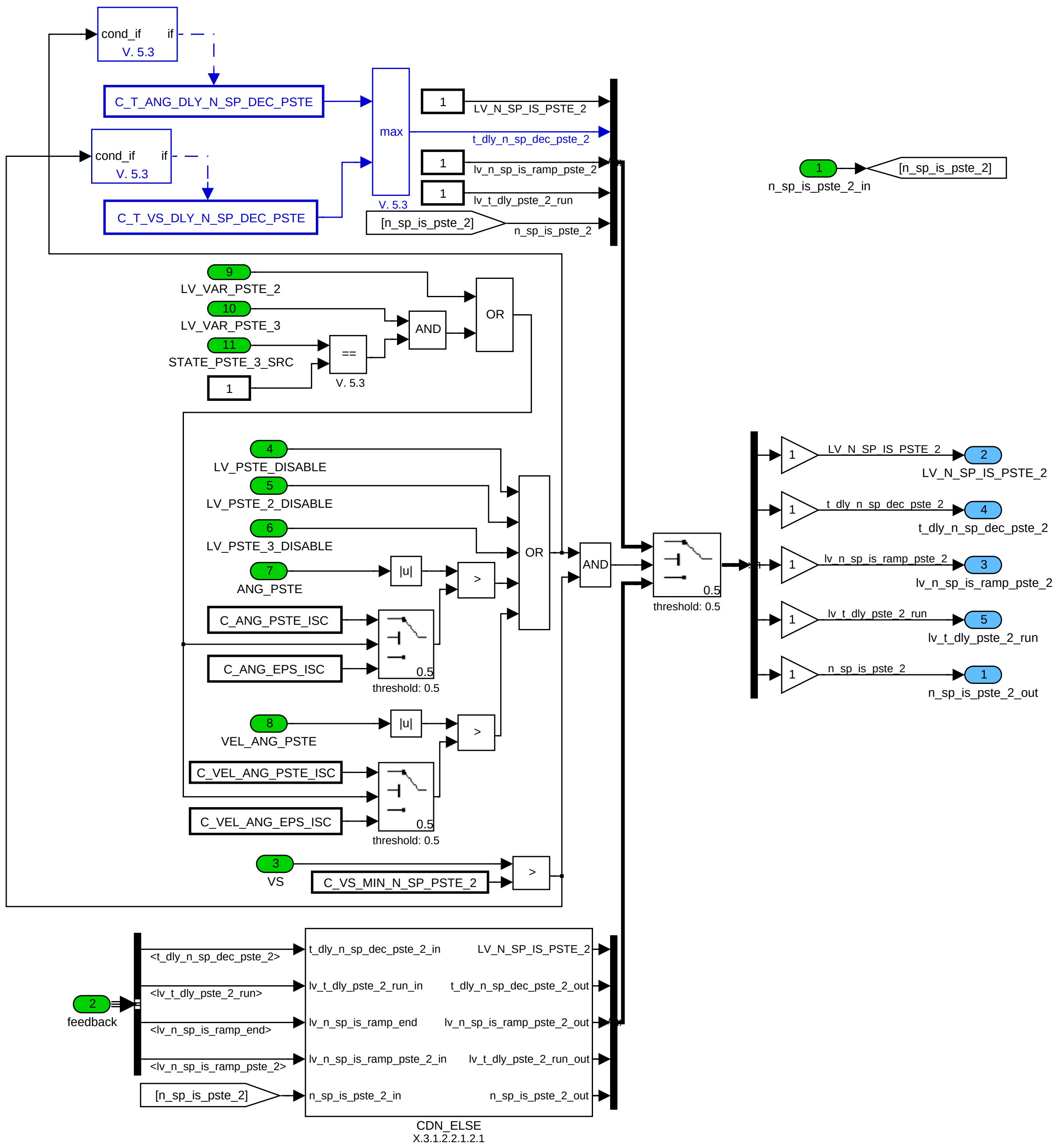


Figure 26:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_2/ISSP_FOR_PSTE/CDN_PSTE_2

22.2.3.1.2.2.1.2.1 Calculations when condition check fails.

Calculations when condition check fails.

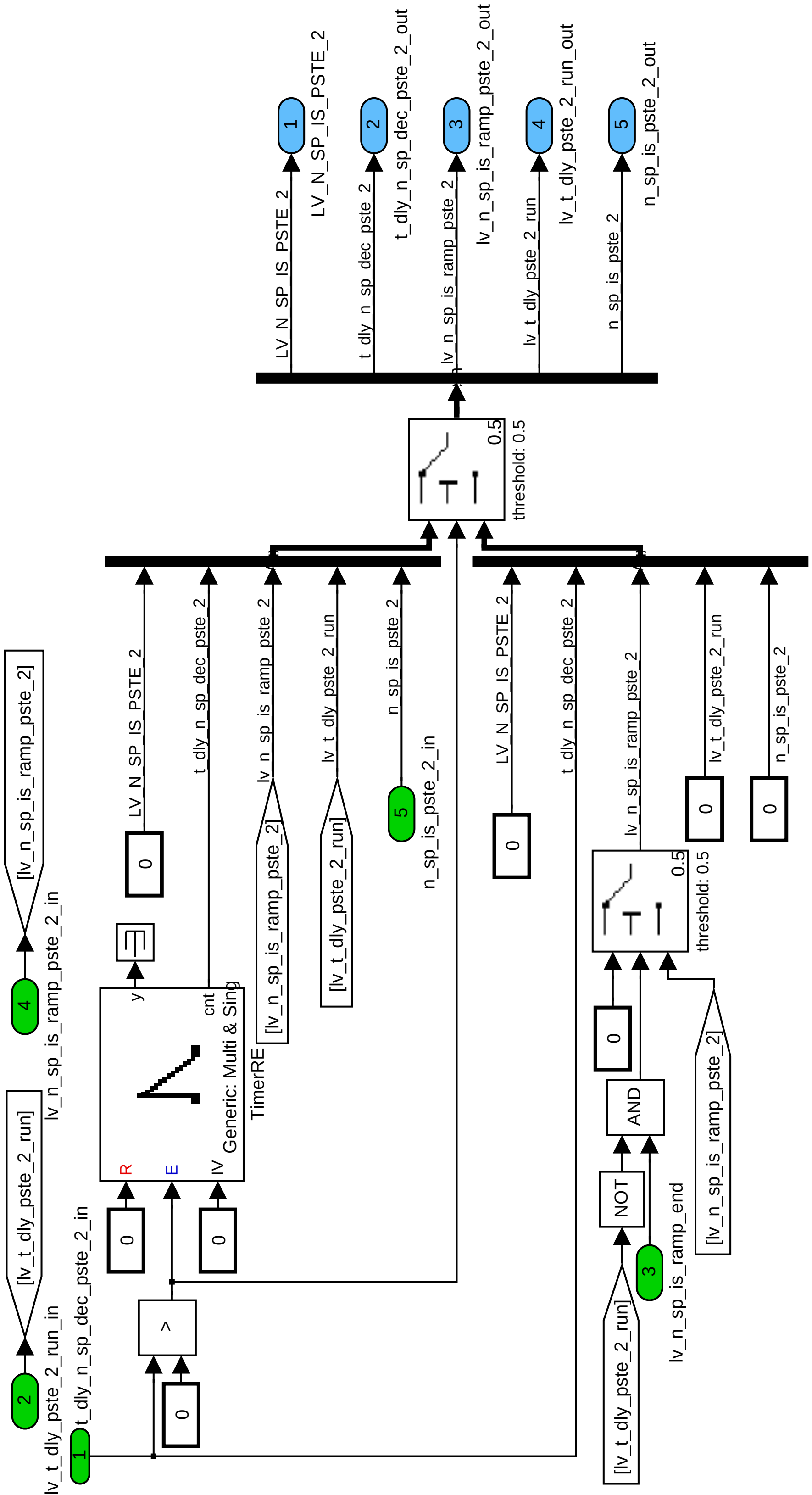


Figure 27:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_2/ISSP_FOR_PSTE/CDN_PSTE_2/CDN_ELSE

22.2.3.1.2.2.2 Idle Speed setpoint at part load (PL)

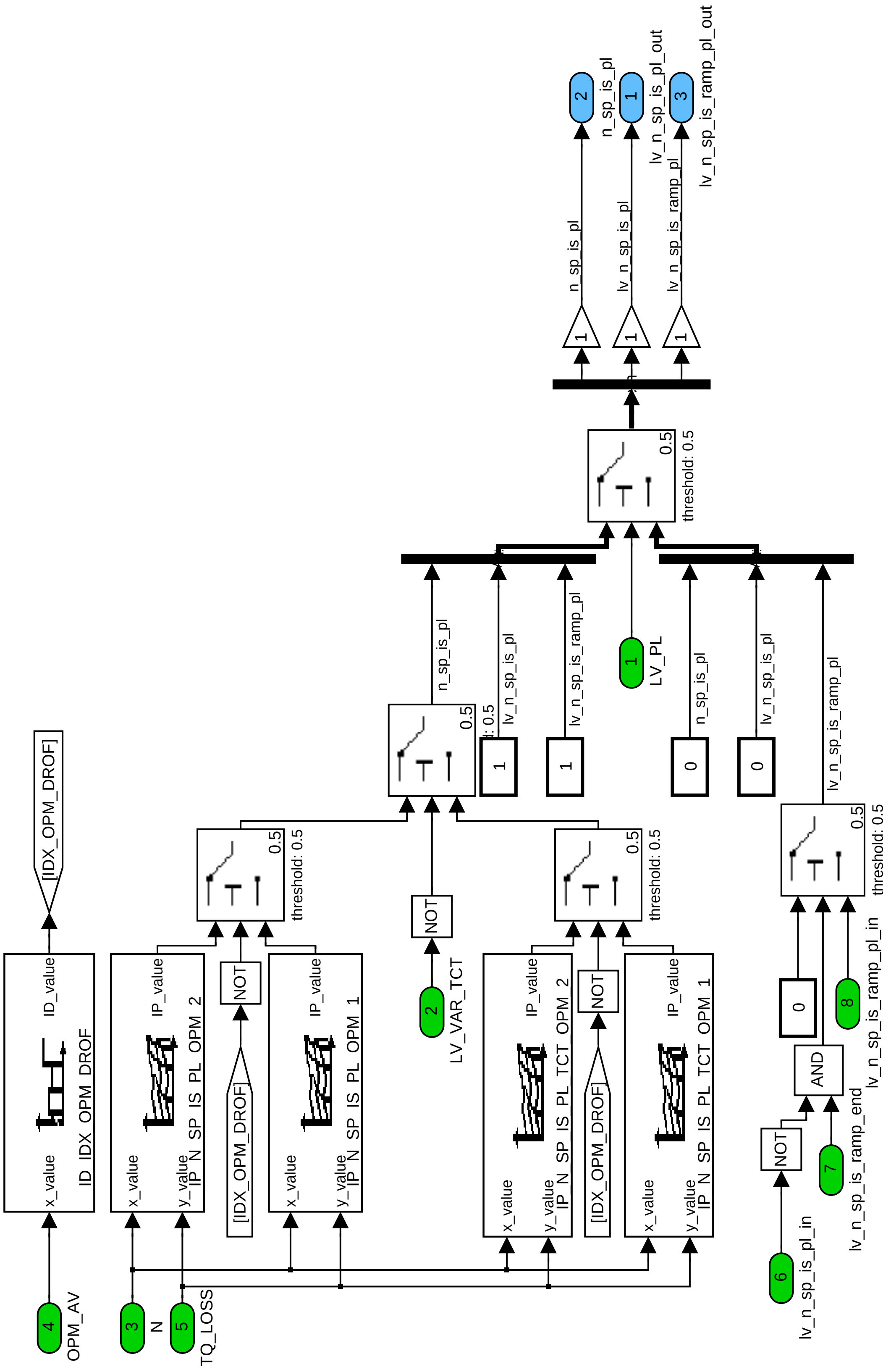


Figure 28:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_2/ISSP_AT_PL

22.2.3.1.2.2.3 Idle Speed setpoint due to power management request for power supply stability

The engine speed setpoint N_SP_IS_PWR_STAB can be activated by the power supply stability request from the power management. The height of this setpoint depends on whether the driving position is engaged or not.

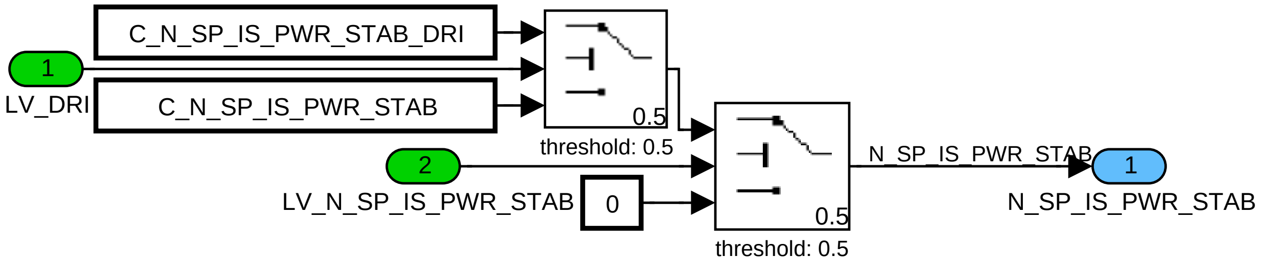


Figure 29:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_2/ISSP_PM_POW_SUPPLY

22.2.3.1.2.2.4 Idle speed set point due to TCT(twin clutch transmission)

The engine speed setpoint N_SP_IS_TCT_CAN from TCT is transmitted via CAN and used as N_SP_IS_TCT if LV_VAR_TCT = 1 and if it is the greatest of the speed setpoints entering the MAXselection.

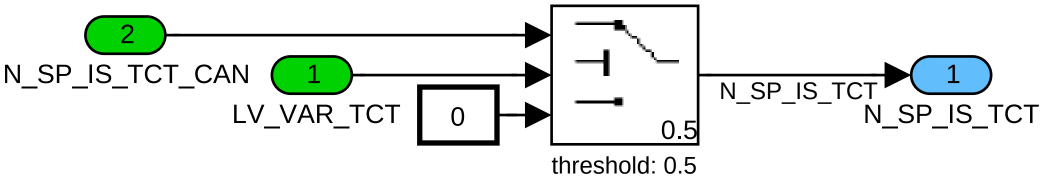


Figure 30:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_2/ISSP_DUE_TO_TCT

22.2.3.1.2.3 Exceptional engine speed setpoint

There are 4 other cases which are not mentioned in above formula section. First is a function which allows to bypass the MAXselection if a correspondingly command (LV_ACT_N_SP_IS_BAS_EXT_ADJ) was received via Kline. In this case, N_SP_IS_3 is set to the basis idle speed setpoint N_SP_IS_2.
Second is a function which allows to bypass the MAXselection if another correspondingly command (LV_ACT_N_SP_IS_EXT_ADJ) was received via Kline. In this case, N_SP_IS_3 is set to the requested idle speed setpoint N_SP_IS_EXT_ADJ.
Third is an extra idle speed setpoint C_N_SP_IS_SA_EOL during the secondary air function end of line test . It is valid during LV_ACT_SA_EOL = 1.
Fourth, a switch for activation of desulfatisation, LV_NT_SO2P_EXT_ADJ_ACT was introduced to increase N_SP_IS to 3000...4000 rpm.
4th case is limp home (highest priority). If PVS error and PVS limp home is requested (LV_N_SP_IS_LIH_ACT = 1) a different idle speed setpoint can be calibrated. So a higher vehicle dynamic as at nominal idle speed is possible.

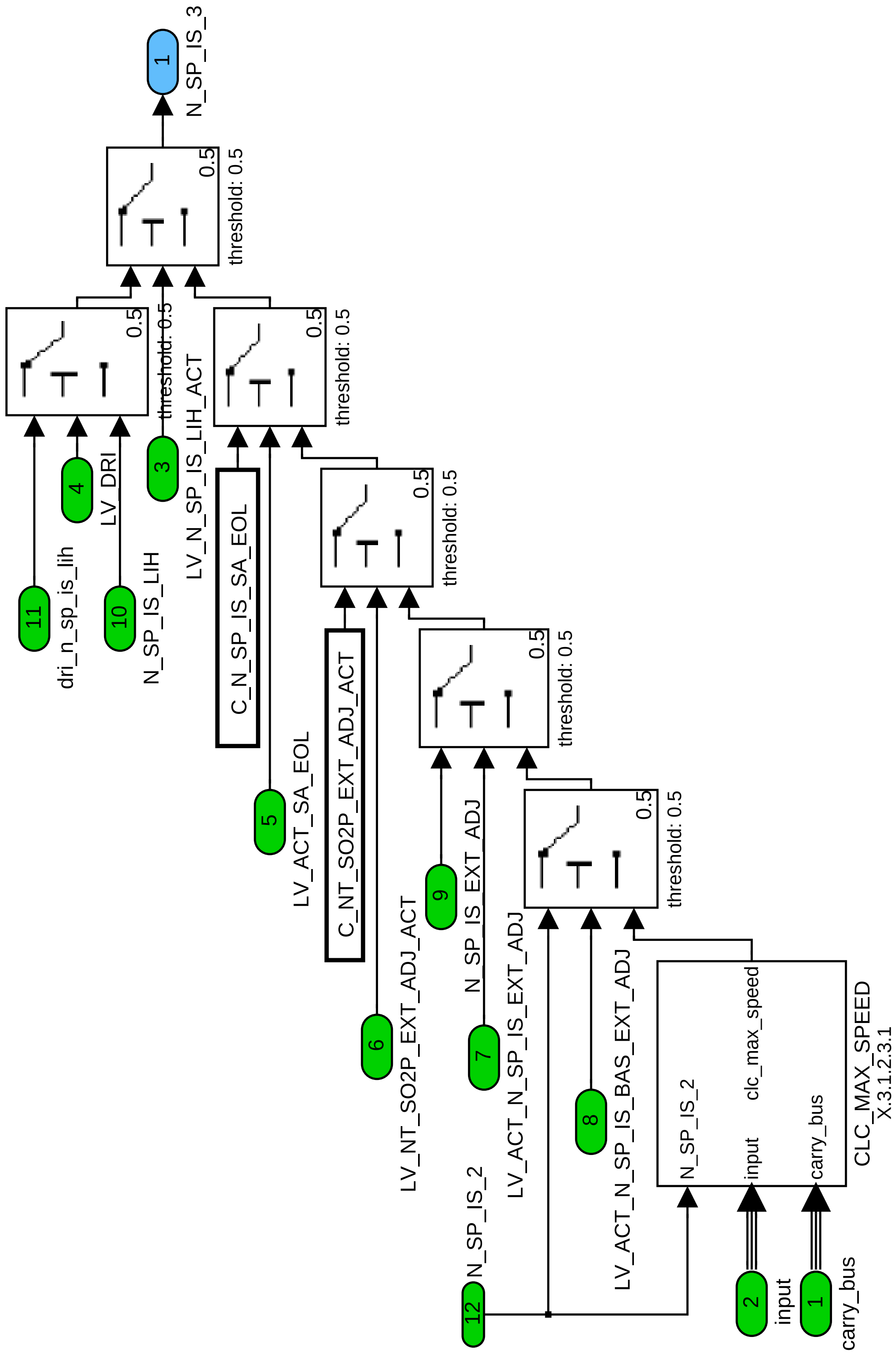


Figure 31:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_3

22.2.3.1.2.3.1 Calculation of maximum speed from set of idle speed. calculated in pre calculation subsystem.

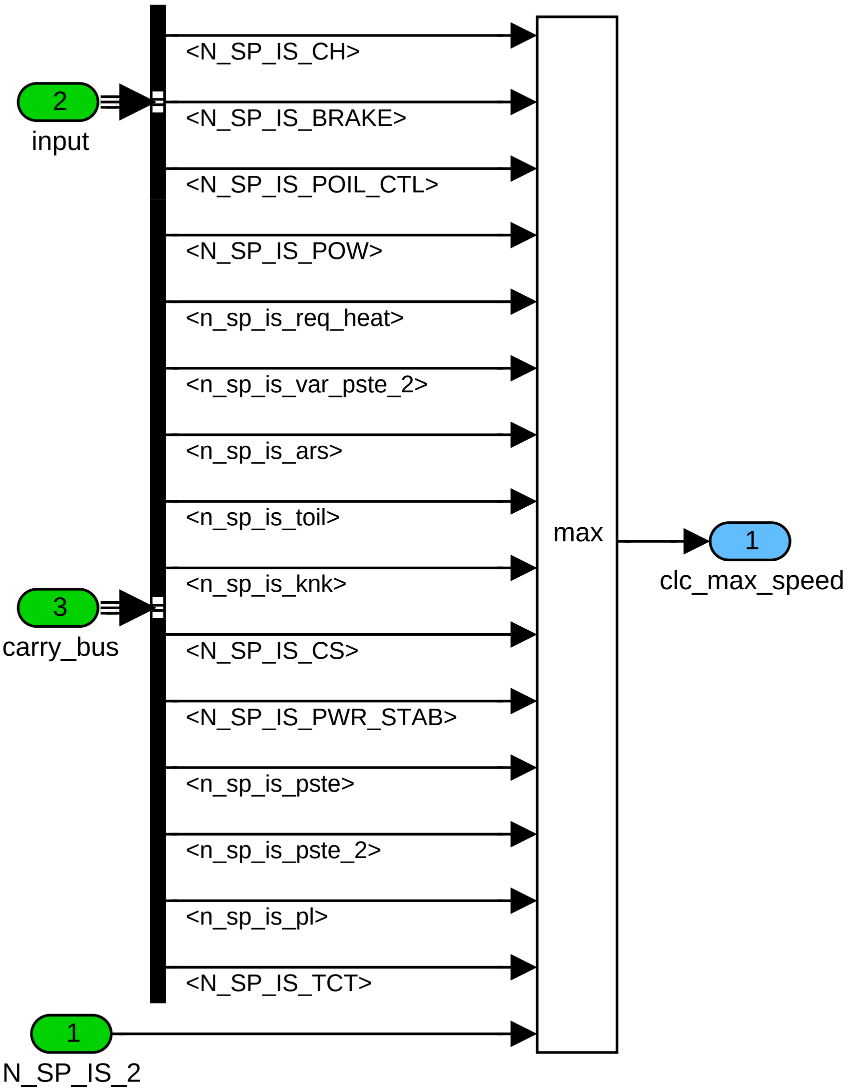


Figure 32:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_3/CLC_OUT_3/CLC_MAX_SPEED

22.2.3.1.3 Gradient limitation

The gradient limitation for N_SP_IS by C_N_SP_LGRD_IS, C_N_SP_DEC_IS_CH_DRI, C_N_SP_DEC/INC_CS or C_N_SP_DEC/INC_PSTE is finished as soon as the goal N_SP_IS_3 is reached (N_SP_IS = N_SP_IS_3)
During active limp home a special gradient limitation C_N_SP_LGRD_IS_LIH is used.

An increase of engine speed setpoint is not allowed in case of automatic transmission vehicle with drive engaged, vehicle standing or very low speed and engine operating state idle active. This is to prevent runaway of the car.

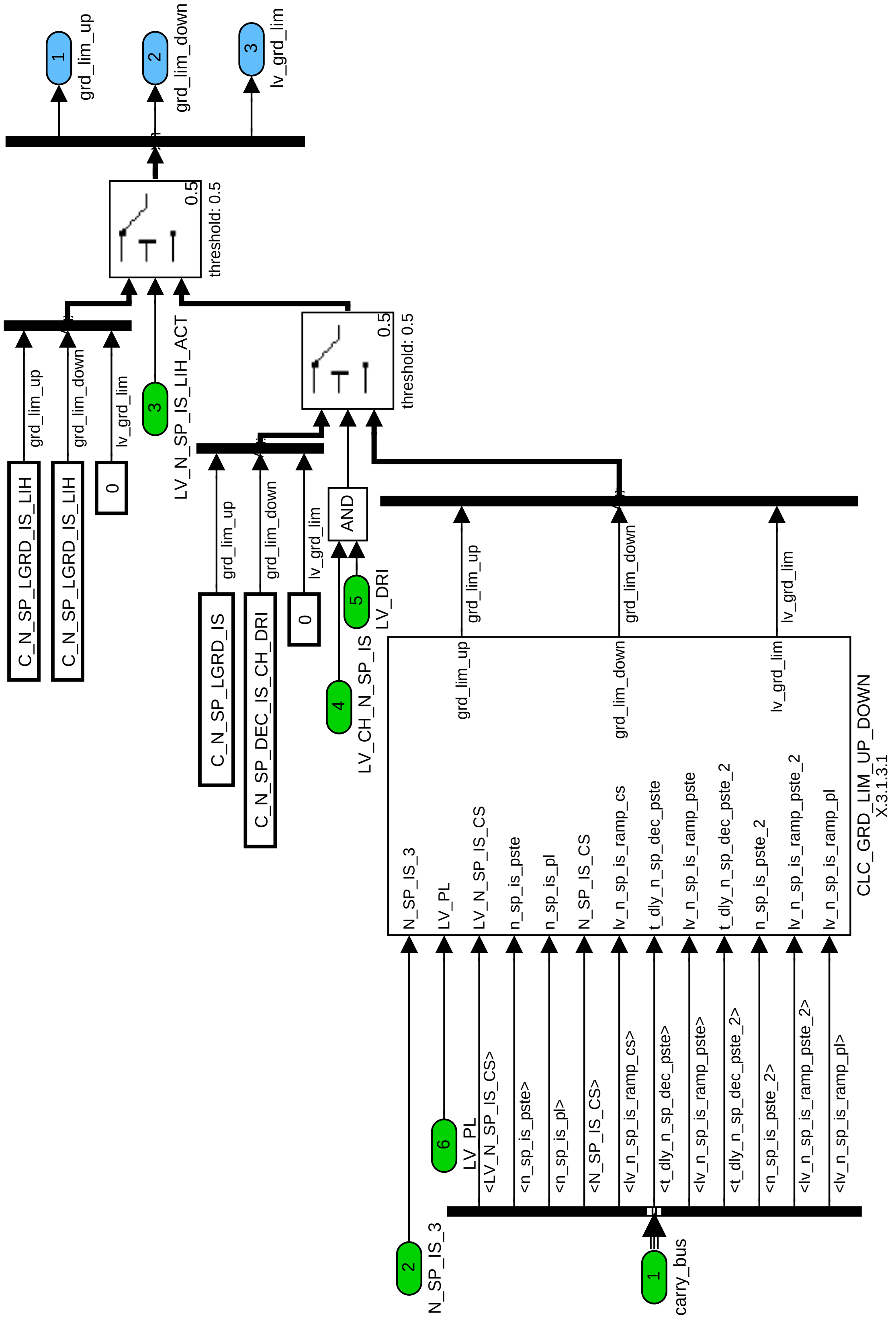


Figure 33:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_GRD_LIM

22.2.3.1.3.1 Calculation of Gradient limitation continued...

If LV_N_SP_IS_LIH_ACT is 0 and LV_DRI and LV_SH_N_SP_IS are 1 then this part is calculated.

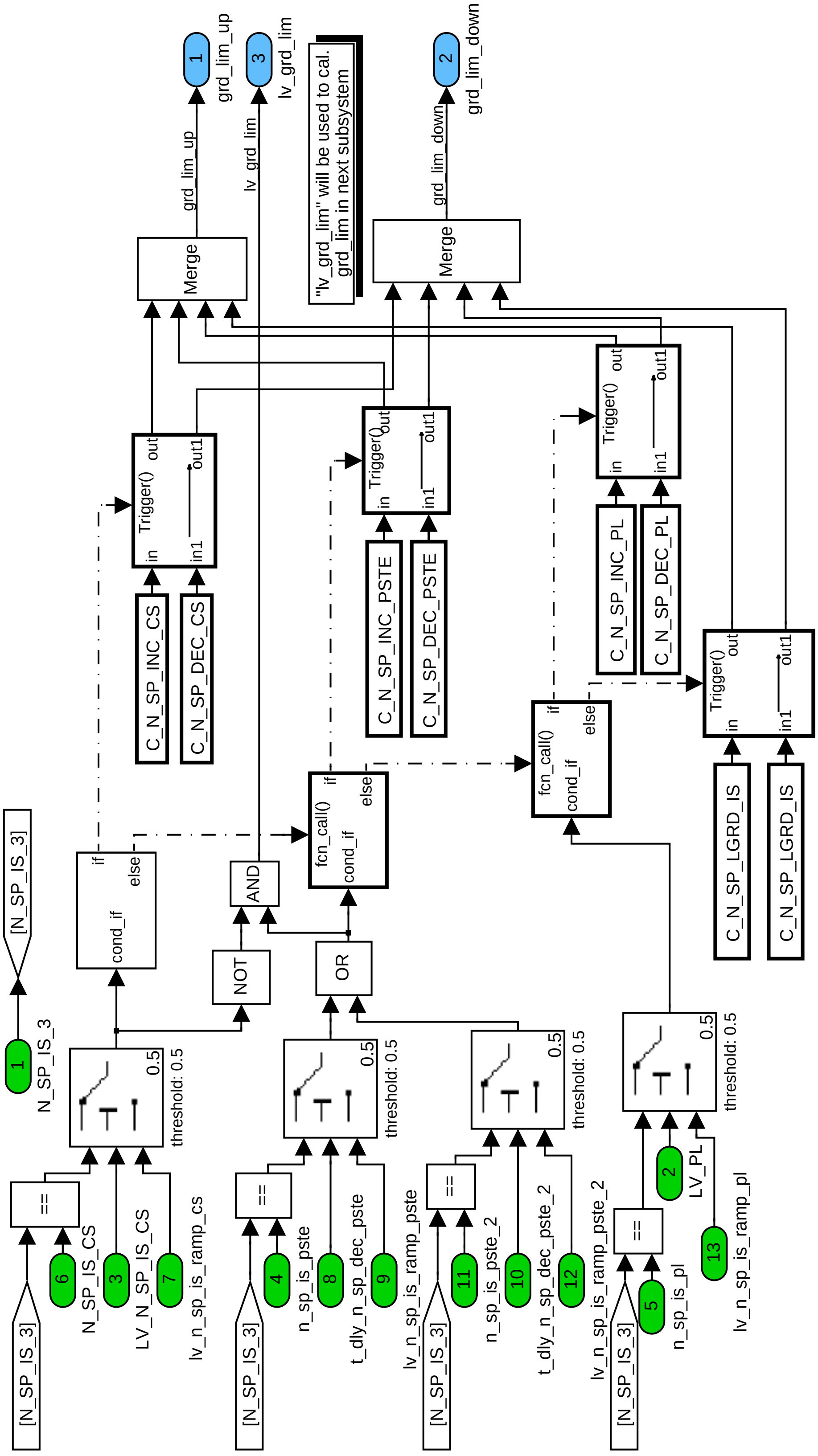


Figure 34:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_GRD_LIM/CLC_GRD_LIM_UP_DOWN

22.2.3.1.4 Determination of N_SP_IS

During start and after end of start gradient limitation of N_SP_IS can be disabled by delay time C_T_DLY_GRD_LIM.

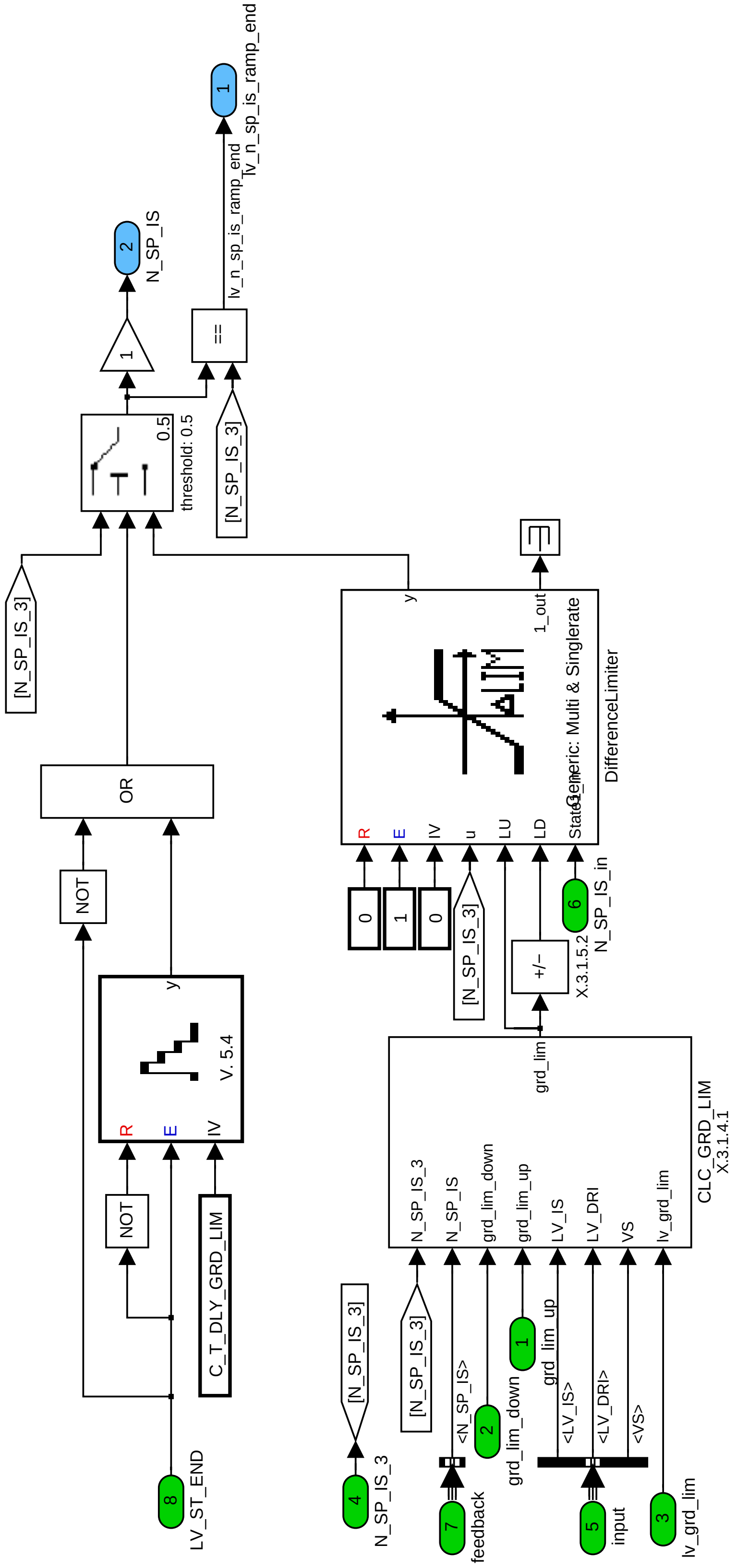


Figure 35:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS

22.2.3.1.4.1 Calculation of optimal value of gradient limitation.

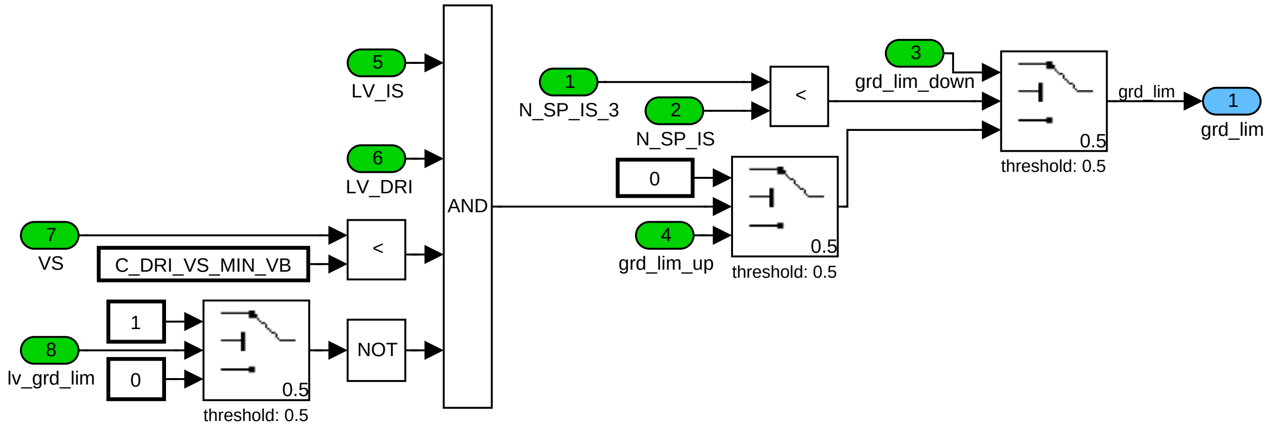


Figure 36:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS/CLC_GRD_LIM

22.2.3.1.5 Calculated engine speed setpoint ratio

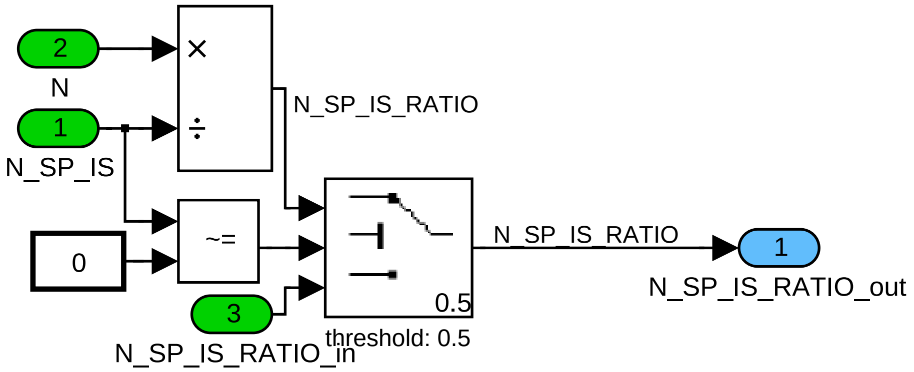
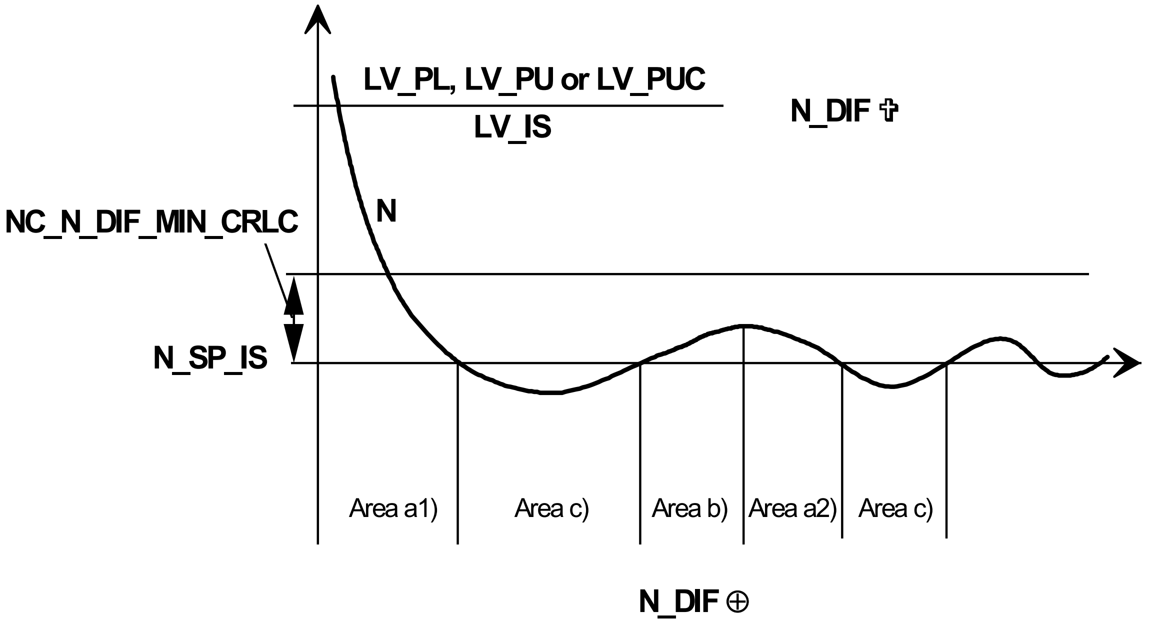


Figure 37:
Path: ENSC_M400M/OPM_10MS/CLC_N_SP_IS/CLC_N_SP_IS_RATIO

22.2.3.2 Calculation of N_DIF and N_DIF_ST

22.2.3.2.1 Engine speed deviations N_DIF, N_DIF_MMV, N_DIF_COR



N_DIF_MMV calculation:
After entry in Idle (LV_IS=1), N_DIF_MMV is set to N_DIF taking into account the limits defined below. Then the moving mean value is determined:

N_DIF_MMV (n) = N_DIF_MMV (n-1) + (N_DIF - N_DIF_MMV(n-1)) * N_DIF_CRLC

The correlation constant C_N_DIF_CRLC may assume 3 discrete values:

a1) Decreasing engine speed above nominal idle speed after entry in Idle (LV_IS=1) and conditions b.) and c.) not previously met (or N_DIF < NC_N_DIF_MIN_CRLC):

(N_DIF (n-3) < N_DIF (n) and N_DIF (n) < 0)

N_DIF_CRLC = C_N_DIF_CRLC (per adjustment)

a2) Decreasing engine speed above nominal idle speed

and N_DIF > NC_N_DIF_MIN_CRLC

and condition b.) previously fulfilled

N_DIF_CRLC = 1

b) Increasing engine speed above nominal idle speed

((N_DIF (n-3) >= N_DIF (n) and N_DIF (n) < 0)

N_DIF_CRLC = 1

c) Engine speed below nominal idle speed after entry at idle and is still below idle speed

(N_DIF (n) >= 0)

N_DIF_CRLC = 0

Limits of N_DIF_MMV:

C_N_DIF_MIN_MMV (N_DIF_MMV (0
N_DIF_COR calculation:

N_DIF_COR is relevant for the idle controller (idle-charge actuator and ignition timing):

N_DIF_COR = N_DIF - N_DIF_MMV * C_N_DIF_FAC

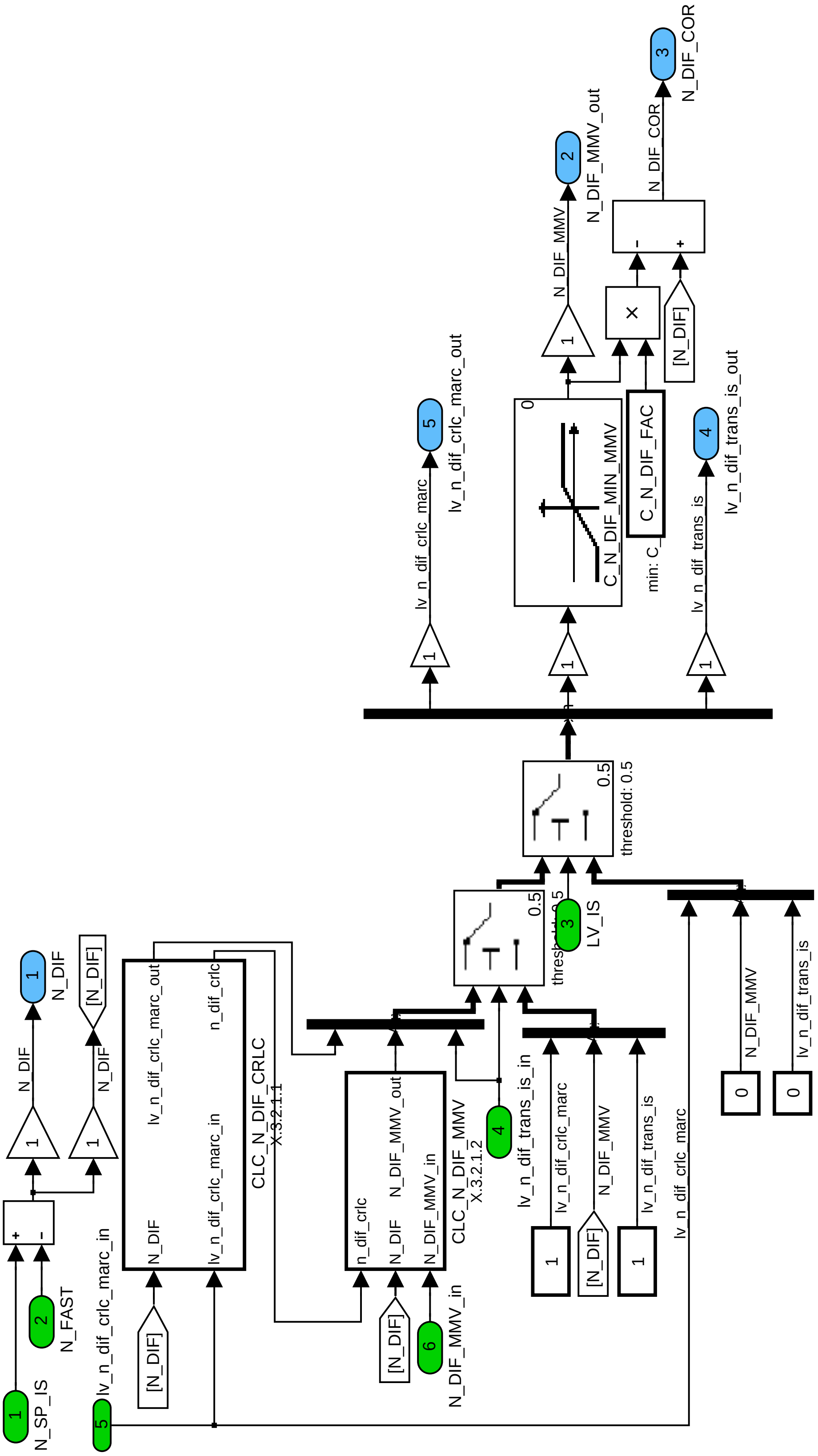


Figure 38:

Path: ENSC_M400M/OPM_10MS/CLC_N_DIF_N_DIF_ST/CLC_N_DIF

22.2.3.2.1.1 Determination of correlation factor

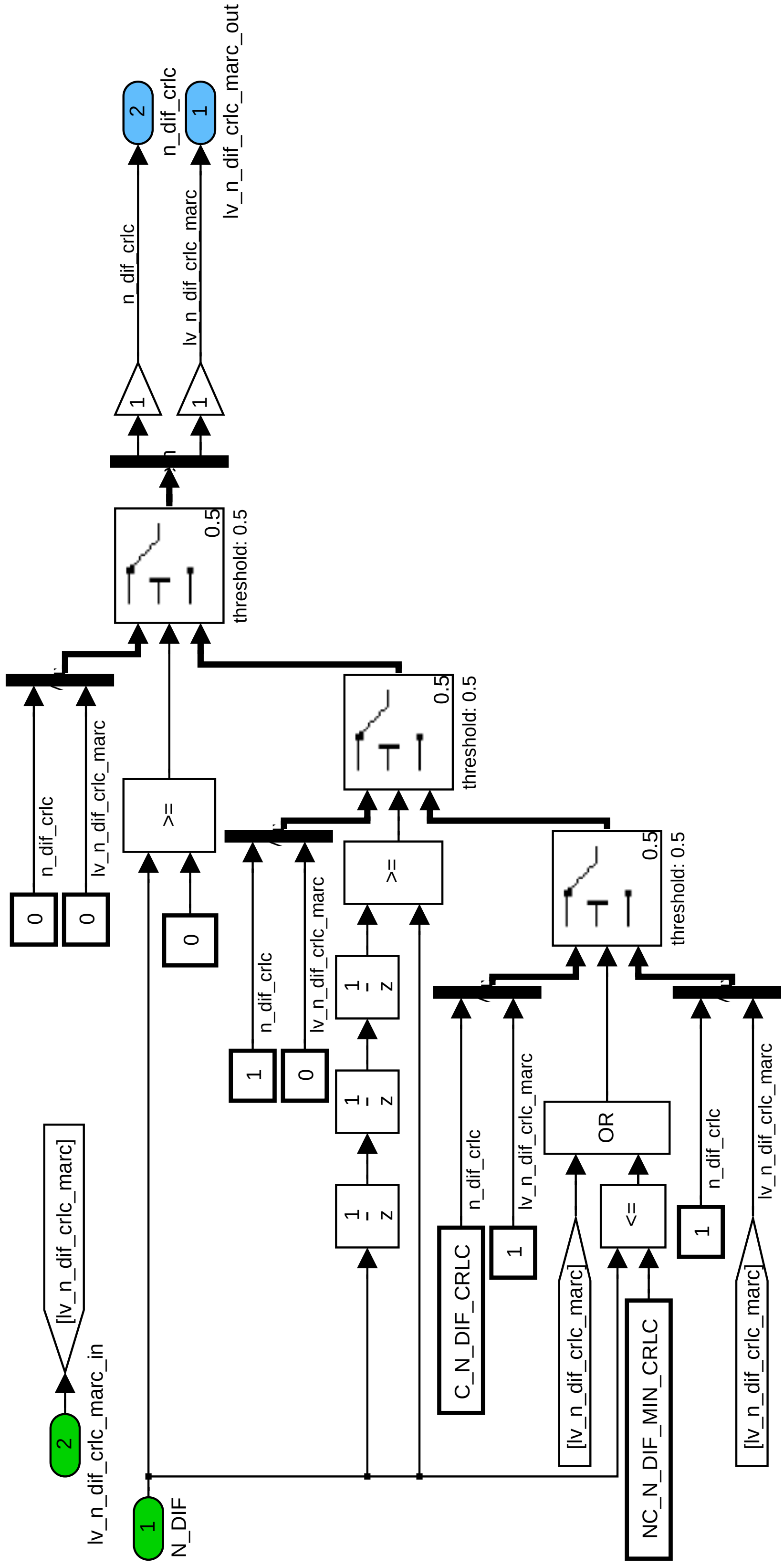


Figure 39:
Path: ENSC_M400M/OPM_10MS/CLC_N_DIF_N_DIF_ST/CLC_N_DIF/CLC_N_DIF_CRLC

22.2.3.2.1.2 Calculation of N_DIF_MMV

N_DIF_MMV is determined here using low pass filter.

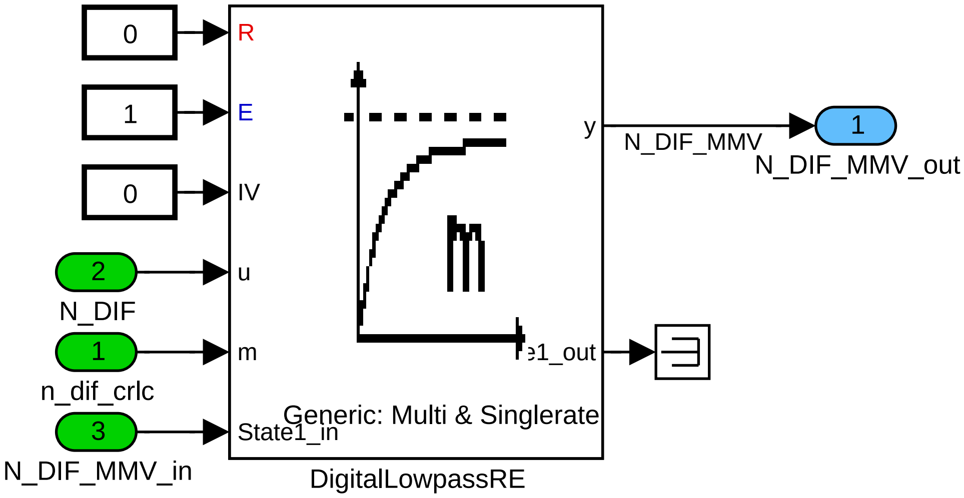


Figure 40:
Path: ENSC_M400M/OPM_10MS/CLC_N_DIF_N_DIF_ST/CLC_N_DIF/CLC_N_DIF_MMV

22.2.3.2.2 Engine speed deviation N_DIF_ST

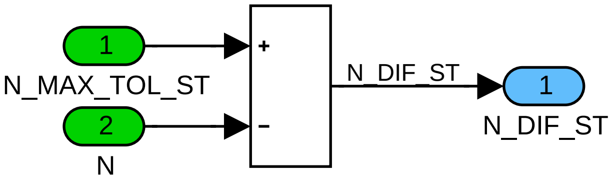
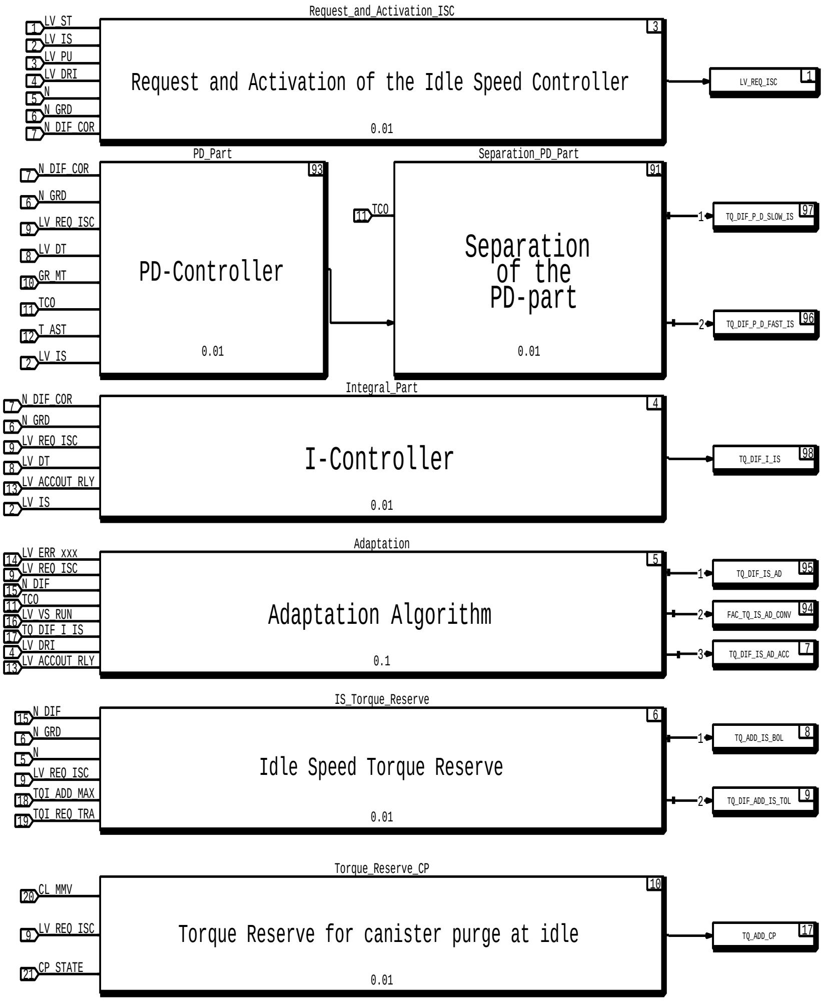


Figure 41:
Path: ENSC_M400M/OPM_10MS/CLC_N_DIF_N_DIF_ST/CLC_N_DIF_ST

22.3 Idle Speed Control General

22-FEB-99

Discrete SuperBlock	Sample Period	Sample Skew	Inputs	Outputs	Enable Signal
Idle_Speed_Control_General	0.01	0.	21	10	Parent



22.4 Idle Speed Controller

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_PAS_RAMP_ACT_P_D_IS	O/V	0...1H	0...1	1	-
Logical value for PD-part passive ramp active					
TQ_DIF_P_D_FAST_IS	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part for fast-path of torque coordination					
TQ_DIF_P_D_SLOW_IS	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part for slow-path of torque coordination					
TQ_DIF_I_IS	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I-part					
LV_PAS_RAMP_ACT_I_IS	O/V	0...1H	0...1	1	-
Logical value for I-part passive ramp active					
LV_TQ_ISC_I_TQ_PSTE	O/V	0...1H	0...1	1	-
Logical variable for limitation due to PSTE					
STATE_PAS_RAMP_ACT_P_D_IS	O/V	0...FFH	0...255	1	-
Circular buffer for the logical value for PD-part passive ramp active					
STATE_PAS_RAMP_ACT_I_IS	O/V	0...FFH	0...255	1	-
Circular buffer for the logical value for I-part passive ramp active					
LV_TQ_DIF_I_IS_INI_ACT	O/V	0...1H	0...1	1	-
Flag for activating the re-initialisation of I-Part of idle speed controller (f.e. by LV_DT=1->0)					
TQ_ADD_I_IS	V	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
Incrememt of limited integrator					
TQ_DIF_P_D_IS_OPM_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part before separation into fast/slow path					
TQ_DIF_P_D_IS_OPM_2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part before separation into fast/slow path					
TQ_DIF_I_IS_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I-part at normal out of calculation					
N_GRD_P_D_IS	V	80...7FH	-4.096E+3 ... 4.064E+3	32	rpm/s
Filtered gradient; gradient input of the PD-part maps					
TQ_DIF_P_D_DT_IS_SLOW_OPM_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part at drive train into slow path (mode1)					
TQ_DIF_P_D_DT_IS_SEL	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Resulting PD-part at drive train into fast path					
LV_TQ_P_D_ACT_SLOW	V	0...1H	0...1	1	-
Logical variable for PD-part air path active (STATE_P_D_ISC = P_D_NORMAL)					
TQ_DIF_P_D_DT_IS_SLOW_OPM_2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part at drive train into slow path (mode2)					
TQ_DIF_P_D_DT_IS_1_SLOW	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part at drive train into slow path					
TQ_DIF_P_D_IS_SLOW_OPM_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part before separation into fast/slow path					
TQ_DIF_P_D_IS_SLOW_SEL	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Resulting PD-part before separation into fast/slow path					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_DIF_P_D_IS_SLOW_OPM_2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part before separation into fast/slow path					
FAC_N_GRD_IS_SLOW	V	0...FFH	0...1.9921875	0.0078125	-
Factor in dependance of TCO and TOIL for weighting the speed gradient influence on the PD-part (for LV_DT=0)					
TQ_DIF_P_D_DT_IS_OPM_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part at drive train into fast path (mode1)					
TQ_DIF_P_D_IS_FAST	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part before separation into fast/slow path					
TQ_DIF_P_D_DT_IS_OPM_2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part at drive train into fast path (mode2)					
TQ_DIF_P_D_DT_IS_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part at drive train into fast path					
TQ_ADD_P_D_RAMP_FAST	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Increment of the PD-part ramps ignition path					
TQ_DIF_P_D_IS_SEL	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Resulting PD-part before separation into fast/slow path					
FAC_N_GRD_IS	V	0...FFH	0...1.9921875	0.0078125	-
Factor in dependance of TCO and TOIL for weighting the speed gradient influence on the PD-part (for LV_DT=0)					
TQ_PAS_I_IS_SEL	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Resulting I-component for idle speed controller at transition out of Passive with					
TQ_PAS_I_IS_OPM_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I-component for idle speed controller at transition out of Passive with (mode1)					
N_GRD_FAC_P_D	V	80...7FH	-4.096E+3 ... 4.064E+3	32	rpm/s
output of N_GRD_P_D_IS * FAC_N_GRD_IS; gradient input of the PD-part maps					
TQ_PAS_I_ACC_IS_SEL	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Resulting I-component for idle speed controller at transition out of Passive with air condition					
TQ_PAS_I_ACC_IS_OPM_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I-component for idle speed controller at transition out of Passive with air condition (mode1)					
TQ_PAS_I_ACC_IS_OPM_2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I-component for idle speed controller at transition out of Passive with air condition (mode2)					
TQ_I_IS_OPM_1	V	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
I-part at normal out of calculation (mode1)					
TQ_ADD_P_D_RAMP_SLOW	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Increment of the PD-part ramps air path					
TQ_I_IS_SEL	V	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
Resulting I-part at normal out of calculation					
TQ_I_IS_OPM_2	V	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
I-part at normal out of calculation (mode2)					
TQ_I_DT_IS_SEL	V	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
Resulting I-part at closed drive train out of calculation					
TQ_I_DT_IS_OPM_1	V	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
I-part at closed drive train out of calculation (mode1)					
TQ_I_DT_IS_OPM_2	V	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
I-part at closed drive train out of calculation (mode2)					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_DIF_P_D_IS_SLOW	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
PD-part before separation into fast/slow path					
TQ_DIF_P_D_DT_IS_SLOW_SEL	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Resulting PD-part at drive train into slow path					
N_GRD_I_IS	V	80...7FH	-4.096E+3 ... 4.064E+3	32	rpm/s
Filtered gradient; gradient input of the I-part maps					
TQ_ADD_I_RAMP	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Increment of the I-part ramps					
TQ_PAS_I_IS_OPM_2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I-component for idle speed controller at transition out of Passive with (mode2)					
TQ_DIF_I_IS_2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I-part at normal but ramp operation					
N_GRD_FAC_P_D_SLOW	V	80...7FH	-4.096E+3 ... 4.064E+3	32	rpm/s
output of N_GRD_P_D_IS * FAC_N_GRD_IS_SLOW; gradient input of the PD-slow-part maps					
LV_TQ_P_D_ACT_FAST_TEMP	V	0...1H	0...1	1	-
Flag to activate LV_TQ_P_D_ACT_FAST in case of manual or automatic transmission					
TQ_LIM_I_IS	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Actual torque limitation value for integral part					
STATE_P_D_ISC	V	0H 1H 2H	P_D_PASSIVE P_D_NORMAL RAMP_TO_ZERO	1	-
State of the PD-part: P_D_PASSIVE, P_D_NORMAL, RAMP_TO_ZERO					
LV_TQ_P_D_ACT_FAST	V	0...1H	0...1	1	-
Logical variable for PD-part fast active (STATE_P_D_ISC = P_D_NORMAL)					
STATE_I_ISC	V	0H 1H 2H 3H	I_PASSIVE I_NORMAL RAMP_TO_PAS RAMP_TO_LIM	1	-
State of the integral part: I_PASSIVE, I_NORMAL, RAMP_TO_PAS, RAMP_TO_LIM					
IP_FAC_P_D_SLOW_TCO_IS	-	0...FFFFH	0...1.999969	3.05176E-5	-
TCO depending factor for slow-path of torque coordination					

Input data:

LV_ES	LV_REQ_ISC	LV_DT	TCO
LV_ISC_OFF_DROF	LV_ISC_INH_EXT_ADJ	GR_MT	T_AST
N_DIF_COR	N_GRD	LV_IGK	LV_ACCOUT_RLY
TQ_ADD_PSTE	TQ_LOSS_PSTE	LV_CS	OPM_AV
OPM_REQ	FAC_TQ_ADD_IS_OPM_SEL	TQ_ADD_CH	LV_VAR_TCT
ID_IDX_OPM_DROF	ID_TQ_ADD_IS_OPM_REQ	ID_TQ_ADD_IS_OPM_AV	N
N_SP_IS	LV_AT	GR_AT	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_N_GRD	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Filter correlation constant					
C_NR_BUF_N_GRD_FIL	1	0...6H	0...6	1	-
number of relevant buffer cells for N_GRD_FIL_1_IS					
C_N_DIF_I_IS_INI	1	E020...1FE0H	-8.16E+3 ... 8.16E+3	1	rpm
engine speed difference to setpoint for re-initialisation of I-Part of idle speed controller					
C_THD_TQ_ADD_PSTE_I_ISC	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
TQ_ADD_PSTE threshold for calc of I limitation threshold					
C_THD_TQ_ADD_PSTE_I_ISC_HYS	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Hysteresis for TQ_ADD_PSTE threshold to leave I limitation					
C_THD_TQ_LOSS_PSTE_I_ISC	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
TQ_LOSS_PSTE threshold for calc of I limitation threshold					
C_THD_TQ_LOSS_PSTE_I_ISC_HYS	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Hysteresis for TQ_LOSS_PSTE threshold to leave I limitation					
C_TQ_BOL_I_DT_AT_IS	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Bottom limit for I-controller output and drive train closed-Automatic Transmission					
C_TQ_BOL_I_DT_IS	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Bottom limit for I-controller output and drive train closed					
C_TQ_BOL_I_IS	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Bottom limit for I-controller output					
C_TQ_DIF_I_IS_DT_THD	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Threshold value for reset of TQ_DIF_I_IS in case of LV_DT = 0 --> 1					
C_TQ_DIF_I_IS_INI	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Initialization value of TQ_DIF_I_IS in case of LV_DT = 0 --> 1					
C_TQ_ISC_I_MAX_TQ_DROF	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I limitation at drive off					
C_TQ_ISC_I_MAX_TQ_PSTE	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I limitation max value at PSTE intervention					
C_TQ_ISC_I_MIN_TQ_DROF	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I limitation at drive off					
C_TQ_ISC_I_MIN_TQ_PSTE	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
I limitation min value at PSTE intervention					
C_TQ_MAX_ISC_TQ_DROF_FAST	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
P_D Limitation at Drive off					
C_TQ_MAX_ISC_TQ_DROF_SLOW	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
P_D limitation at drive off					
C_TQ_MIN_ISC_TQ_DROF_FAST	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
P_D Limitation at Drive off					
C_TQ_MIN_ISC_TQ_DROF_SLOW	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
P_D limitation at drive off					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TQ_TOL_I_DT_IS_OPM_1	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Top limit for I-controller output and drive train closed in operation mode 1					
C_TQ_TOL_I_DT_IS_OPM_2	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Top limit for I-controller output and drive train closed in operation mode 2					
C_TQ_TOL_I_IS	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Top limit for I-controller output					
C_T_RAMP_LIM_I_DT_IS	1	0...FFFFH	0...1.3107E+3	0.02	s
Time constant for ramp limit of I-controller output in case of idle speed controller passive and drive train engagedResolution is directly coupled with update rate					
C_T_RAMP_LIM_I_IS	1	0...FFFFH	0...1.3107E+3	0.02	s
Time constant for ramp limit of I-controller output in case of idle speed controller passiveResolution is directly coupled with update rate					
C_T_RAMP_LIM_P_D_DT_IS	1	0...FFFFH	0...655.35	0.01	s
Time constant for ramp limit of P-, D-controller output in case of idle speed controller passive and LV_DT=1					
C_T_RAMP_LIM_P_D_IS	1	0...FFFFH	0...655.35	0.01	s
Time constant for ramp limit of P-, D-controller output in case of idle speed controller passive					
C_T_TQ_ISC_I_TQ_PSTE_LIM	1	0...FFFFH	0...1.3107E+3	0.02	s
Delay time to reset the idle speed controller limitation due to PSTE					
LC_ISC_INH	1	0...1H	0...1	1	-
Bit for inhibiting the PD- and integral part - default value = 0					
LC_N_GRD_FIL_I_SWI	1	0...1H	0...1	1	-
Logical variable for changing between N_GRD_FIL_1_IS and N_GRD_FIL_IS					
LC_N_GRD_FIL_P_D_SWI	1	0...1H	0...1	1	-
Logical variable for switching between N_GRD_FIL_1_IS and N_GRD_FIL_IS					
LC_P_D_ACT_FAST_AT	1	0...1H	0...1	1	-
Logical variable to activate LV_TQ_P_D_FAST_out for automatic transmission path					
LC_P_D_ACT_FAST_MT	1	0...1H	0...1	1	-
Logical variable to activate LV_TQ_P_D_FAST_out for manual transmission path					
LC_P_D_SLOW_FAST_SEL_ENA	1	0...1H	0...1	1	-
Enable separation of slow and fast path of TQ_DIF_P_D_IS - default value at initialisation = 0					
LC_TQ_ADD_I_INH_AS	1	0...1H	0...1	1	-
Logical variable for ISC I-part map output deactivation - default value at initialisation = 0					
LC_TQ_DIF_I_INH_AS	1	0...1H	0...1	1	-
Logical variable for ISC I-part deactivation (completely) - default value at initialisation = 0					
LC_TQ_DIF_I_IS_DT_INI	1	0...1H	0...1	1	-
Logical variable to activate the re-initialisation of I-Part of idle speed controller at LV_DT=1->0					
LC_TQ_DIF_P_D_INH_AS	1	0...1H	0...1	1	-
Bit for inhibiting the PD-part - default value = 0					
LC_TQ_P_D_FAST_INH_AS	1	0...1H	0...1	1	-
Logical variable for deactivation of TQ_DIF_P_D_IS - default value at initialisation = 0					
LC_TQ_P_D_SLOW_INH_AS	1	0...1H	0...1	1	-
Logical variable for deactivation of TQ_DIF_P_D_IS - default value at initialisation = 0					
ID_FAC_AT_P_D_DT_IS	8	0...FFFFH	0...1.99996948	3.05176E-5	-
LDPM_GR_AT_3	8	0...FFH	0...255	1	-
Weighting factor for TQ_DIF_P_D_DT_IS_SEL for AT					
ID_FAC_AT_P_D_DT_IS_SLOW	8	0...FFFFH	0...1.99996948	3.05176E-5	-
LDPM_GR_AT_3	8	0...FFH	0...255	1	-
Weighting factor for TQ_DIF_P_D_DT_IS_SLOW_SEL for AT					
ID_FAC_MT_P_D_DT_IS	8	0...FFFFH	0...1.99996948	3.05176E-5	-
LDPM_GR_MT_2	8	0...FFH	0...255	1	-
Weighting factor for TQ_DIF_P_D_DT_IS_SEL for MT					
ID_FAC_MT_P_D_DT_IS_SLOW	8	0...FFFFH	0...1.99996948	3.05176E-5	-
LDPM_GR_MT_2	8	0...FFH	0...255	1	-
Weighting factor for TQ_DIF_P_D_DT_IS_SLOW_SEL for MT					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_TCO_PAS_I_IS	4	0...FFH	0...0.99609	0.0039063	-
LDP_TCO__FAC_TCO_PAS_I_IS	4	0...FEH	-48...142.5	0.75	°C
TCO correction at entry in I - share calculation					
IP_TQ_I_DT_IS_OPM_1	12	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
LDPM_N_DIF_COR_TQ_I_DT_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
I-component for idle speed controller at closed drive train (mode1)					
IP_TQ_I_DT_IS_OPM_2	12	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
LDPM_N_DIF_COR_TQ_I_DT_IS_OPM_2	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
I-component for idle speed controller at closed drive train (mode2)					
IP_FAC_N_GRD_IS	8x8	0...FFH	0...1.9921875	0.0078125	-
LDP_TCO_IP_FAC_N_GRD_IS	8	0...FEH	-48...142.5	0.75	°C
LDP_T_AST_IP_FAC_N_GRD_IS	8	0...FFFFH	0...6.5535E+3	0.1	s
Weighting factor for N_GRD input for IP_P_D_IS					
IP_FAC_N_GRD_IS_SLOW	8x8	0...FFH	0...1.9921875	0.0078125	-
LDPM_TCO__FAC_N_GRD_IS	8	0...FEH	-48...142.5	0.75	°C
LDPM_T_AST__FAC_N_GRD_IS	8	0...FFFFH	0...6.5535E+3	0.1	s
Weighting factor for N_GRD input for IP_P_D_IS					
IP_FAC_P_D_FAST_IS	8x6	0...FFFFH	0...1.99996948	3.05176E-5	-
LDPM_TCO_IS_1	8	0...FEH	-48...142.5	0.75	°C
LDPM_TQ_ADD_CH_IS_1	6	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
TCO and TQ_ADD_CH depending factor for fast-path of torque coordination					
IP_FAC_P_D_SLOW_IS	8x6	0...FFFFH	0...1.99996948	3.05176E-5	-
LDPM_TCO_IS_1	8	0...FEH	-48...142.5	0.75	°C
LDPM_TQ_ADD_CH_IS_1	6	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
TCO and TQ_ADD_CH depending factor for slow-path of torque coordination					
IP_FAC_T_AST_I_IS_OPM_1	4x8	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_TCO__FAC_N_GRD_IS_2	4	0...FEH	-48...142.5	0.75	°C
LDPM_T_AST__FAC_N_GRD_IS	8	0...FFFFH	0...6.5535E+3	0.1	s
TQ_ADD_I_IS correction after start in operation mode 1					
IP_FAC_T_AST_I_IS_OPM_2	4x8	0...FFFFH	0...0.99998	1.53E-5	-
LDPM_TCO__FAC_N_GRD_IS_2	4	0...FEH	-48...142.5	0.75	°C
LDPM_T_AST__FAC_N_GRD_IS	8	0...FFFFH	0...6.5535E+3	0.1	s
TQ_ADD_I_IS correction after start in operation mode 2					
IP_P_D_DT_IS_OPM_1	12x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_P_D_DT_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_P_D_DT_IS_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
P- and D-component for idle speed controller at closed drive train (mode1)					
IP_P_D_DT_IS_OPM_2	12x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_P_D_DT_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_P_D_DT_IS_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
P- and D-component for idle speed controller at closed drive train (mode2)					
IP_P_D_DT_IS_SLOW_OPM_1	12x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_P_D_DT_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_P_D_DT_IS_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
P- and D-component for idle speed controller at closed drive train (mode1)					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_P_D_DT_IS_SLOW_OPM_2	12x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_P_D_DT_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_P_D_DT_IS_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
P- and D-component for idle speed controller at closed drive train (mode2)					
IP_P_D_IS_OPM_1	12x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_P_D_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_P_D_IS_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
P- and D-component for idle speed controller at open drive train (mode1)					
IP_P_D_IS_OPM_2	12x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_P_D_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_P_D_IS_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
P- and D-component for idle speed controller at open drive train (mode2)					
IP_P_D_IS_SLOW_OPM_1	12x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_P_D_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_P_D_IS_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
P- and D-component for idle speed controller at open drive train (mode1)					
IP_P_D_IS_SLOW_OPM_2	12x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_P_D_IS_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_P_D_IS_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
P- and D-component for idle speed controller at open drive train (mode2)					
IP_TQ_I_IS_OPM_1	12x8	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
LDPM_N_DIF_COR_TQ_I_IS_1_OPM_1	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_TQ_I_IS_1_OPM_1	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
I-component for idle speed controller at open drive train (mode1)					
IP_TQ_I_IS_OPM_2	12x8	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
LDPM_N_DIF_COR_TQ_I_IS_1_OPM_2	12	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_TQ_I_IS_1_OPM_2	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
I-component for idle speed controller at open drive train (mode2)					
IP_TQ_PAS_I_ACC_IS_OPM_1	4x4	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_TQ_I_IS_OPM_1	4	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_TQ_I_IS_OPM_1	4	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
I-component for idle speed controller at transition out of Passive with air condition (mode1)					
IP_TQ_PAS_I_ACC_IS_OPM_2	4x4	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_TQ_I_IS_OPM_2	4	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_TQ_I_IS_OPM_2	4	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
I-component for idle speed controller at transition out of Passive with air condition (mode2)					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQ_PAS_I_IS_OPM_1	4x4	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_TQ_I_IS_OPM_1	4	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_TQ_I_IS_OPM_1	4	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
I-component for idle speed controller at transition out of Passive with (mode1)					
IP_TQ_PAS_I_IS_OPM_2	4x4	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_COR_TQ_I_IS_OPM_2	4	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_N_GRD_TQ_I_IS_OPM_2	4	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
I-component for idle speed controller at transition out of Passive with (mode2)					

Local actions:

ACTION_ENSC_OPERATION_MANAGER(IN <IN1>, IN <IN2>, OUT <OUT>, OUT <TQ_DIF_OPM_1>, OUT <TQ_DIF_OPM_2>)
Operation manager

22.4.1 Overview

The controller consists of three parts:

PD-part

PD-part separation into slow and fast path

I-part

An operation mode manager splits up each controller part (applied for 2 different operation modes) in accordance with the received interpolation factor FAC_TQ_ADD_IS_OPM_SEL.

The operation manager for idle speed control operates with the operation mode parameters used in the module *Idle speed torque reserve*.

Application Condition

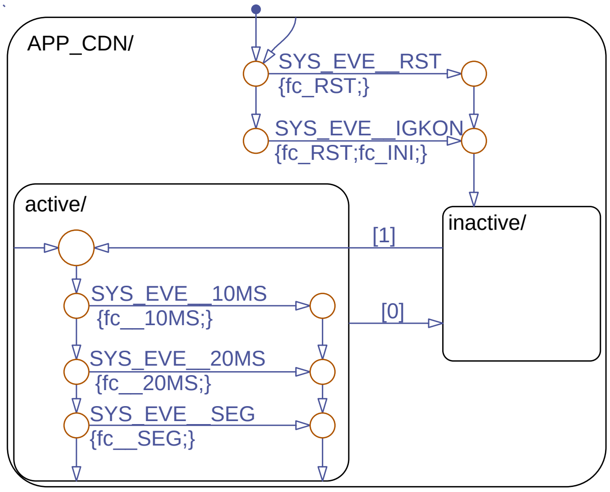
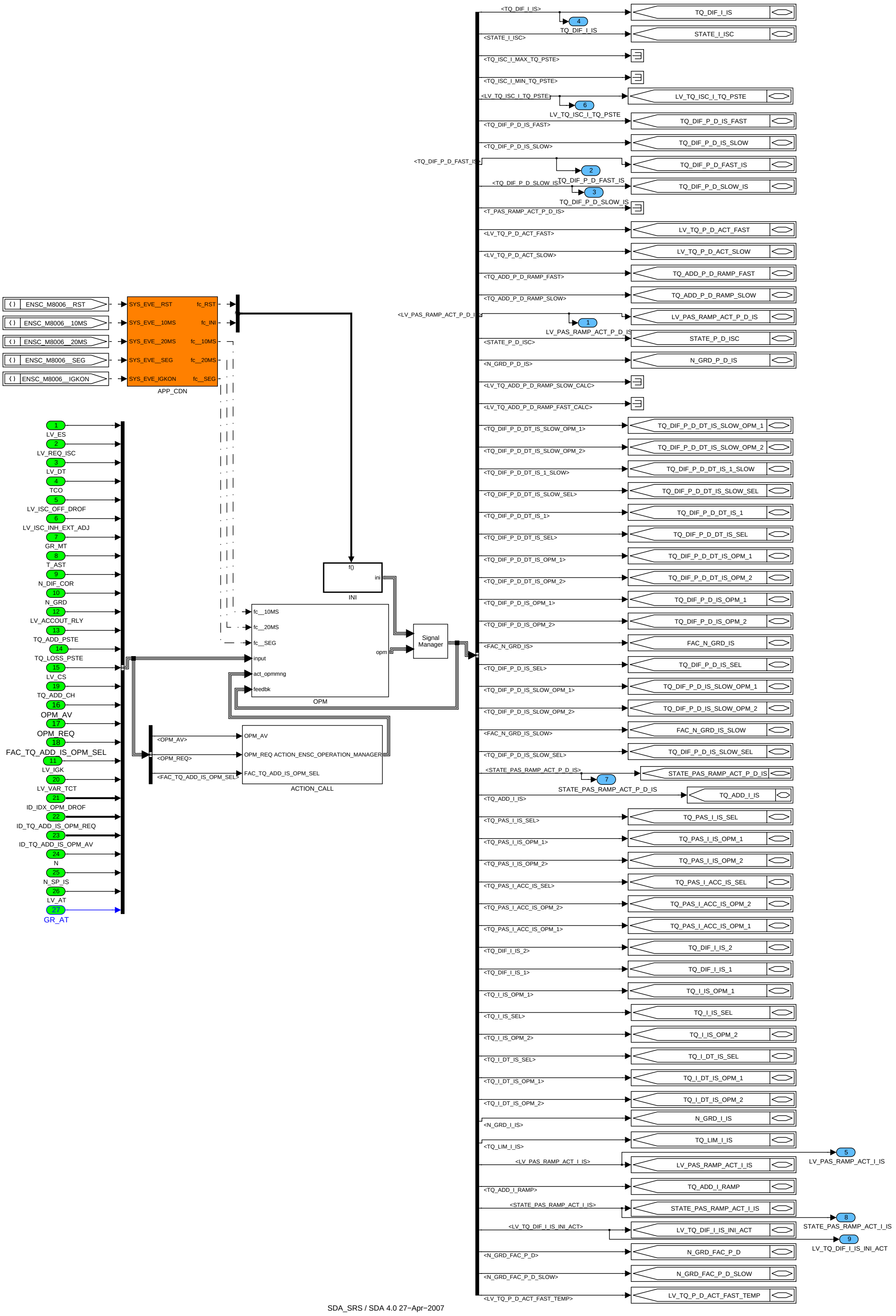


Figure 42 ENSC_M8006/ APP_CDN/ APPCND



SDA_SRS / SDA 4.0 27−Apr−2007

Figure 43 ENSC_M8006

22.4.1.1 Operation mode manager

There are two input values IN_1, IN_2. Each input value belongs to a special operation mode OPM_1, OPM_2. Moreover the value OPM_SEL_AV indicates the active and OPM_SEL_REQ indicates the requested operation mode. The aim of the operation manager is to create an output value depending on the active operation mode. If there is a switch from one mode to another a smooth changeover from one input value to the other is possible by using the interpolation factor FAC_OPM_SEL.

A special logic takes care that the needed part is calculated only. That means if OPM_1 is active and there is no switch requested then the output value of the operation manager = IN_1 and some part of the algorithm for IN_2 are not calculated anymore. In this case the bit LV_OPM_SEL_1=1 indicates that only the calculation of the IN_1-path is done.

The information of the actual and the requested operation mode can be changed by interpretations maps ID_...OPM_SEL. The effect of this is, that by calibration the operation mode 1 can be wired to the input 2. That means the belonging for a operation mode to an input value is flexible.

The following tables defines the local variables described above:

Common variables:

OPM_AV (from Layer)	OPM_AV	Input	Actual operation mode
OPM_REQ (from Layer)	OPM_REQ	Input	Requested operation mode
-	OPM_SEL_AV	Internal value	Actual active operation mode after interpretation
-	OPM_SEL_REQ	Internal value	Requested operation mode after interpretation
FAC_TQ_ADD_IS_OPM_SEL (from Layer)	FAC_OPM_SEL	Input	Interpolation factor
LV_TQ_DIF_IS_OPM_SEL_1	LV_OPM_SEL_1	Output	Operation mode 1 active
LV_TQ_DIF_IS_OPM_SEL_2	LV_OPM_SEL_2	Output	Operation mode 2 active
LV_TQ_DIF_IS_OPM_RLS	LV_OPM_RLS	Output	Operation mode within valid range
ID_TQ_ADD_IS_OPM_AV	ID_OPM_AV	cal	Mapping of active operation mode
ID_TQ_ADD_IS_OPM_REQ	ID_OPM_REQ	cal	Mapping of requested operation mode

Controller parameter:

P_D_SLOW:

TQ_DIF_P_D_IS_SLOW_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_DIF_P_D_IS_SLOW_OPM_2	IN_2	Input	Input value for operation mode 2
TQ_DIF_P_D_IS_SLOW_SEL	OUT	Output	Result of interpolation

P_D_FAST:

TQ_DIF_P_D_IS_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_DIF_P_D_IS_OPM_2	IN_2	Input	Input value for operation mode 2
TQ_DIF_P_D_IS_SEL	OUT	Output	Result of interpolation

P_D_DT_SLOW:

TQ_DIF_P_D_DT_IS_SLOW_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_DIF_P_D_DT_IS_SLOW_OPM_2	IN_2	Input	Input value for operation mode 2
TQ_DIF_P_D_DT_IS_SLOW_SEL	OUT	Output	Result of interpolation

P_D_DT_FAST:

TQ_DIF_P_D_DT_IS_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_DIF_P_D_DT_IS_OPM_2	IN_2	Input	Input value for operation mode 2
TQ_DIF_P_D_DT_IS_SEL	OUT	Output	Result of interpolation

I:

TQ_I_IS_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_I_IS_OPM_2	IN_2	Input	Input value for operation mode 2
TQ_I_IS_SEL	OUT	Output	Result of interpolation

I_DT:

TQ_I_DT_IS_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_I_DT_IS_OPM_2	IN_2	Input	Input value for operation mode 2
TQ_I_DT_IS_SEL	OUT	Output	Result of interpolation

I_PAS:

TQ_PAS_I_IS_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_PAS_I_IS_OPM_2	IN_2	Input	Input value for operation mode 2
TQ_PAS_I_IS_SEL	OUT	Output	Result of interpolation

I_PAS_ACC:

TQ_PAS_I_ACC_IS_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_PAS_I_ACC_IS_OPM_2	IN_2	Input	Input value for operation mode 2
TQ_PAS_I_ACC_IS_SEL	OUT	Output	Result of interpolation

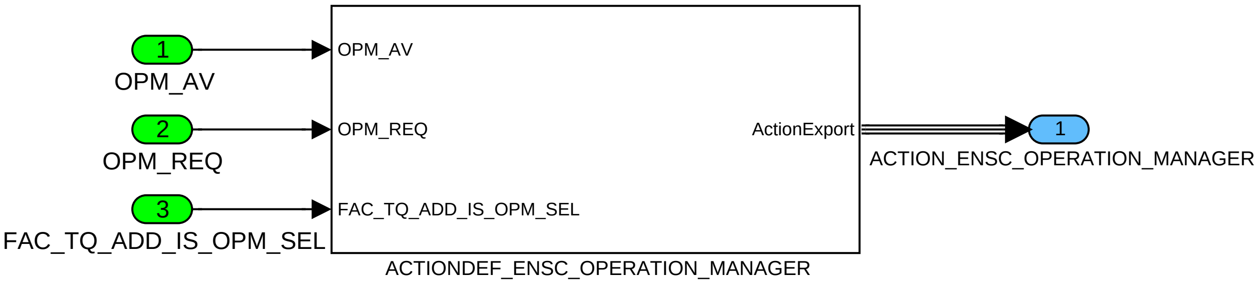


Figure 44 ENSC_M8006/ ACTION_CALL

22.4.1.2 Operation Manager Overview

Description for ACTION_ENSC_OPERATION_MANAGER

ACTION_ENSC_OPERATION_MANAGER(IN <IN1>, IN <IN2>, OUT <OUT>, OUT <TQ_DIF_OPM_1>, OUT <TQ_DIF_OPM_2>)

Operation manager for idle speed controller

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit
IN1	IN	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Input to action manager					
IN2	IN	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Input to action manager					
OUT	OUT	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Output of operation manager					
TQ_DIF_OPM_1	OUT	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Output #1 of operation manager					
TQ_DIF_OPM_2	OUT	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Output #2 of operation manager					

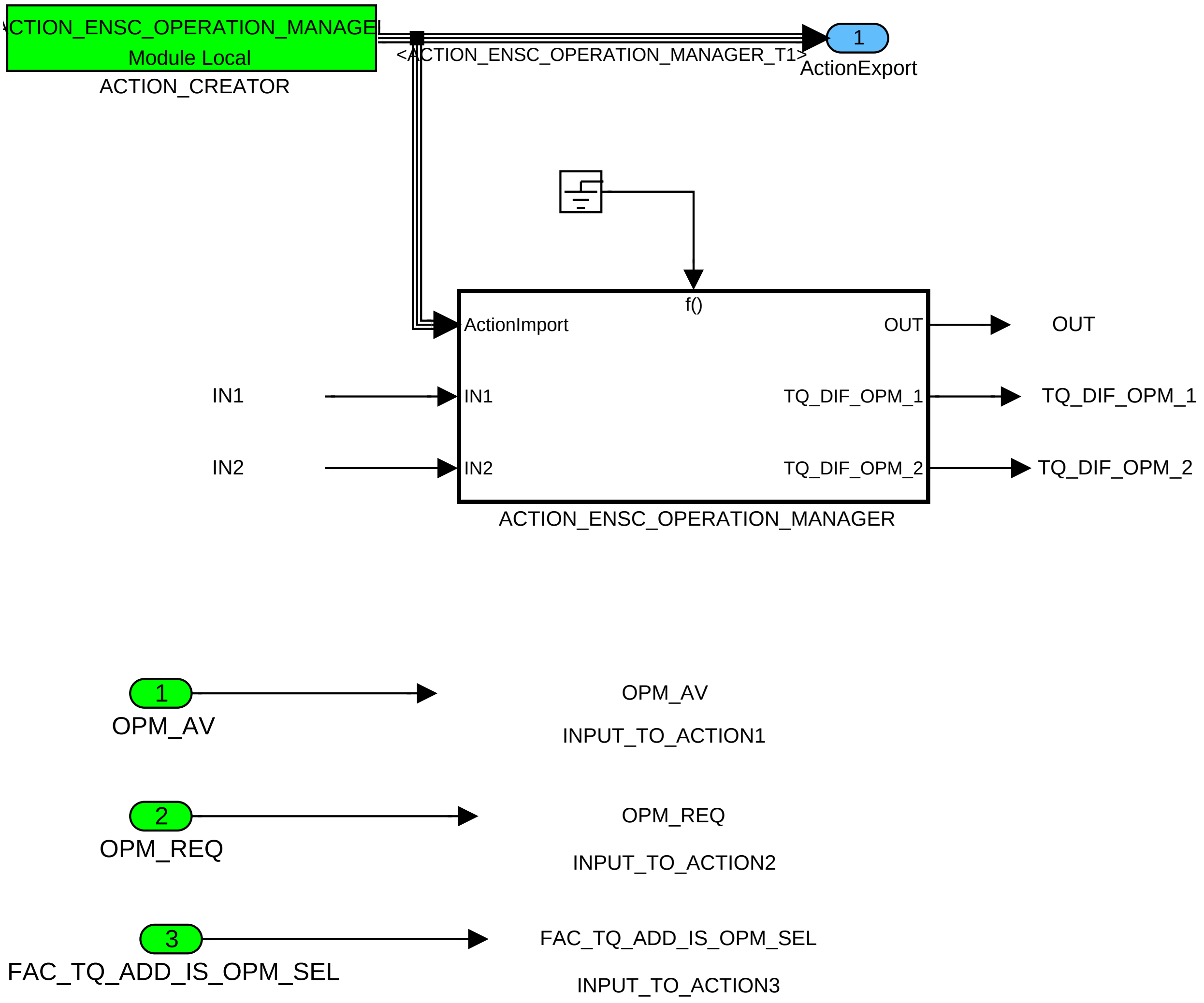


Figure 45 ENSC_M8006/ ACTION_CALL/ ACTIONDEF_ENSC_OPERATION_MANAGER

Selection of inputs to operation manager switch

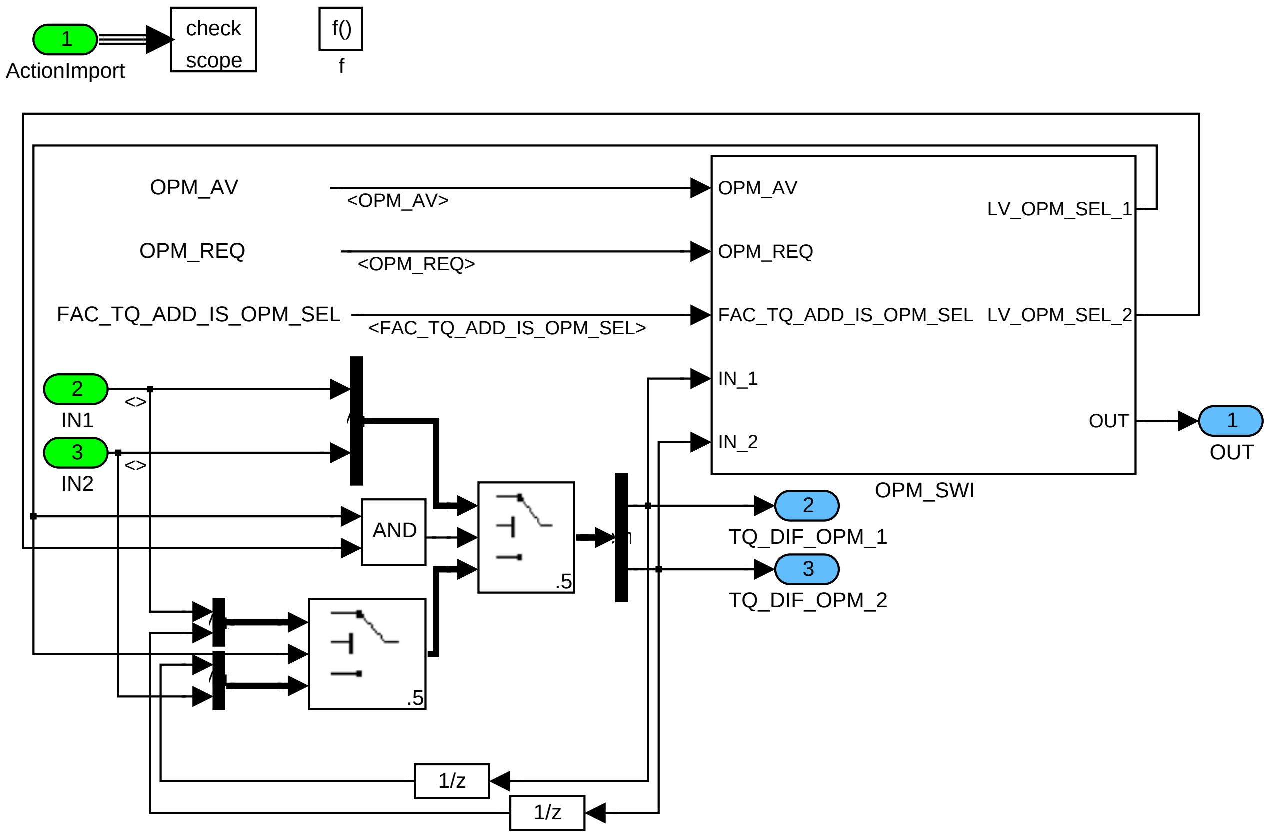


Figure 46 ENSC_M8006/ ACTION_CALL/ ACTIONDEF_ENSC_OPERATION_MANAGER/ ACTION_ENSC_OPERATION_MANAGER

Operation manager switch

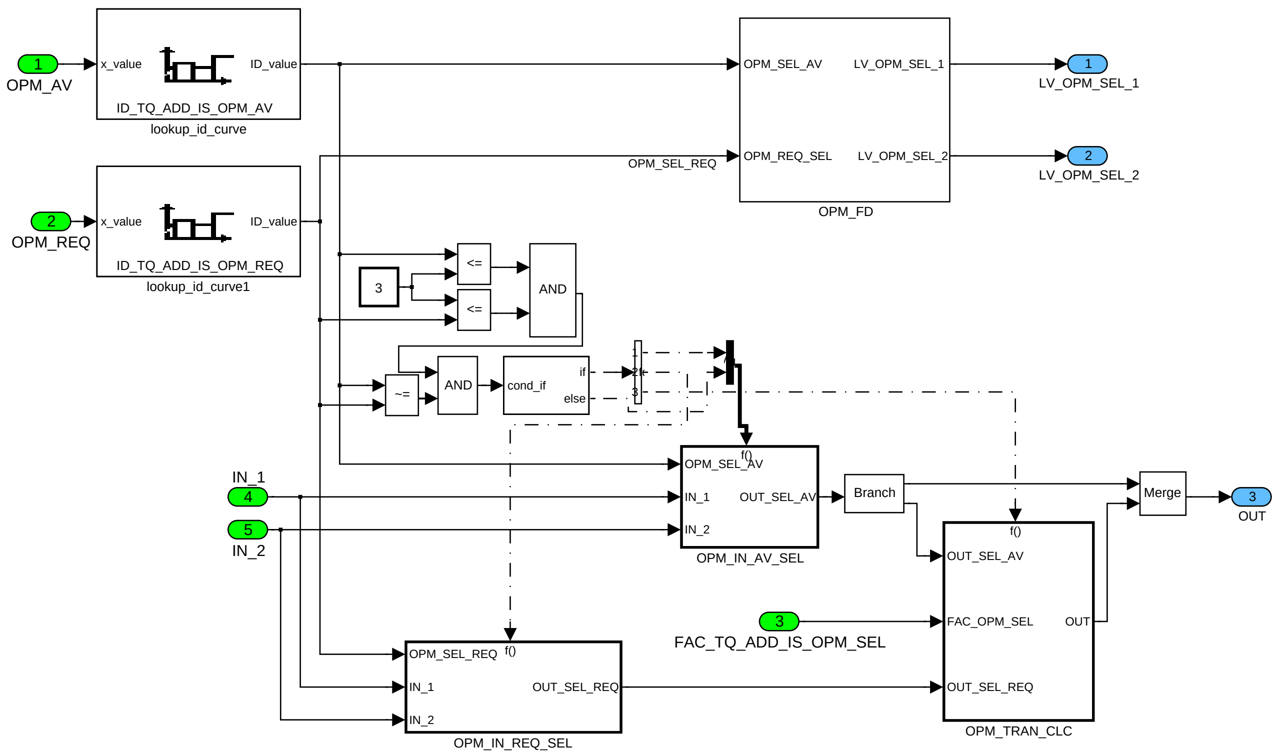


Figure 47 ENSC_M8006/ ACTION_CALL/ ACTIONDEF_ENSC_OPERATION_MANAGER/ ACTION_ENSC_OPERATION_MANAGER/ OPM_SWI

Operation manager feedback

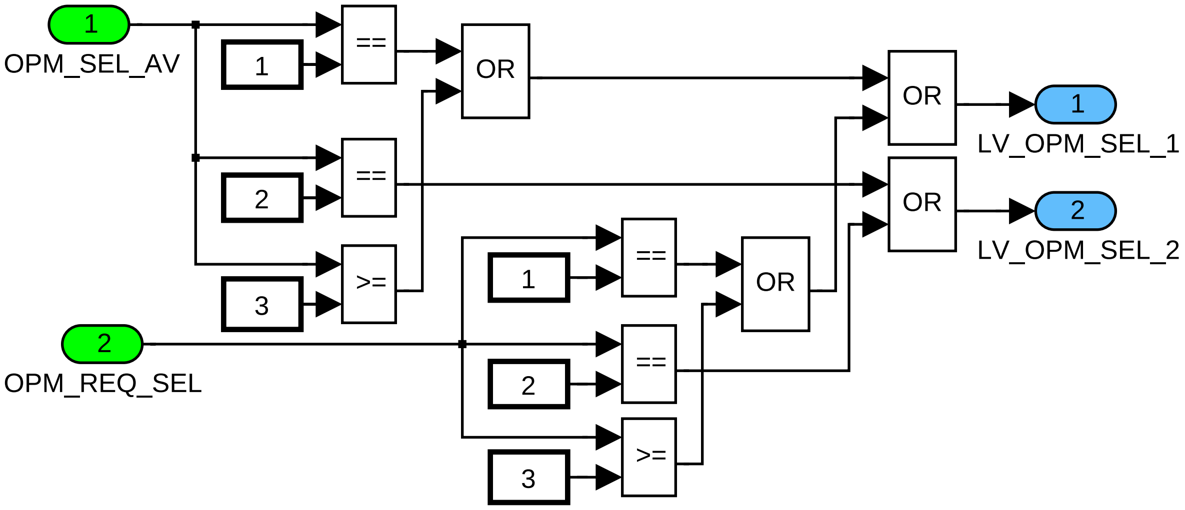


Figure 48 ENSC_M8006/ ACTION_CALL/ ACTIONDEF_ENSC_OPERATION_MANAGER/ ACTION_ENSC_OPERATION_MANAGER/ OPM_SWI/ OPM_FD

Selection of active operation mode

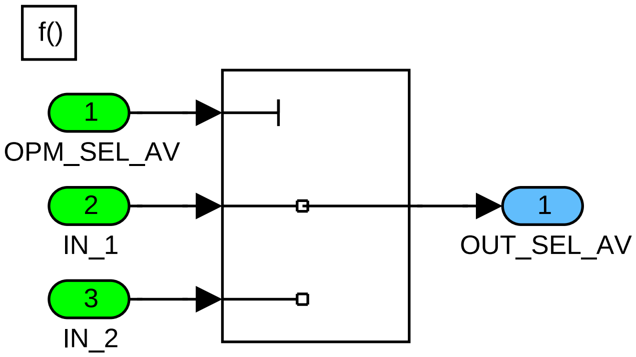


Figure 49 ENSC_M8006/ ACTION_CALL/ ACTIONDEF_ENSC_OPERATION_MANAGER/ ACTION_ENSC_OPERATION_MANAGER/ OPM_SWI/ OPM_IN_AV_SEL

Selection of requested operation mode

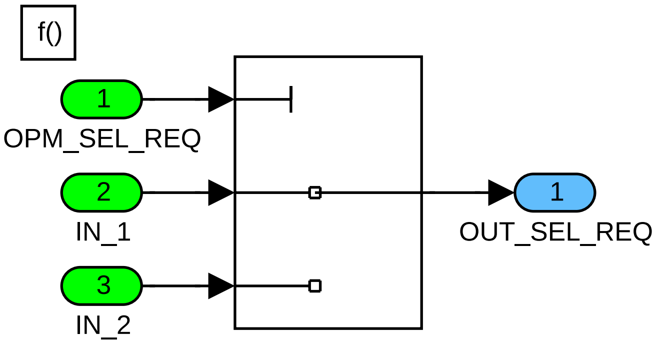


Figure 50 ENSC_M8006/ ACTION_CALL/ ACTIONDEF_ENSC_OPERATION_MANAGER/ ACTION_ENSC_OPERATION_MANAGER/ OPM_SWI/ OPM_IN_REQ_SEL

Output from the operation manager

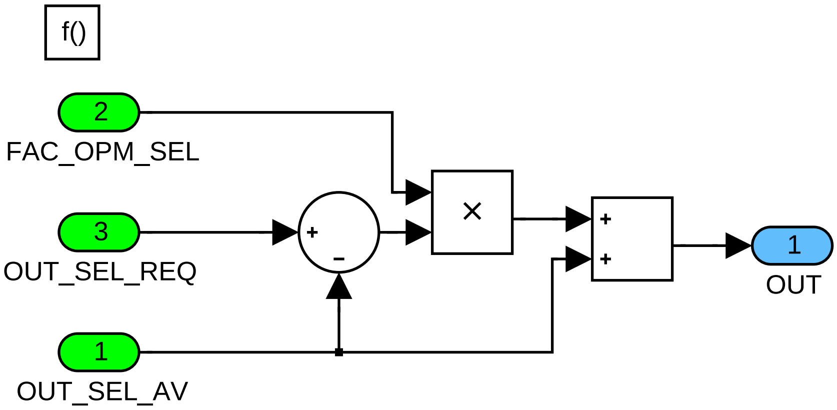


Figure 51 ENSC_M8006/ ACTION_CALL/ ACTIONDEF_ENSC_OPERATION_MANAGER/ ACTION_ENSC_OPERATION_MANAGER/ OPM_SWI/ OPM_TRAN_CLC

22.4.1.3 Initialisation at system event reset and igkon

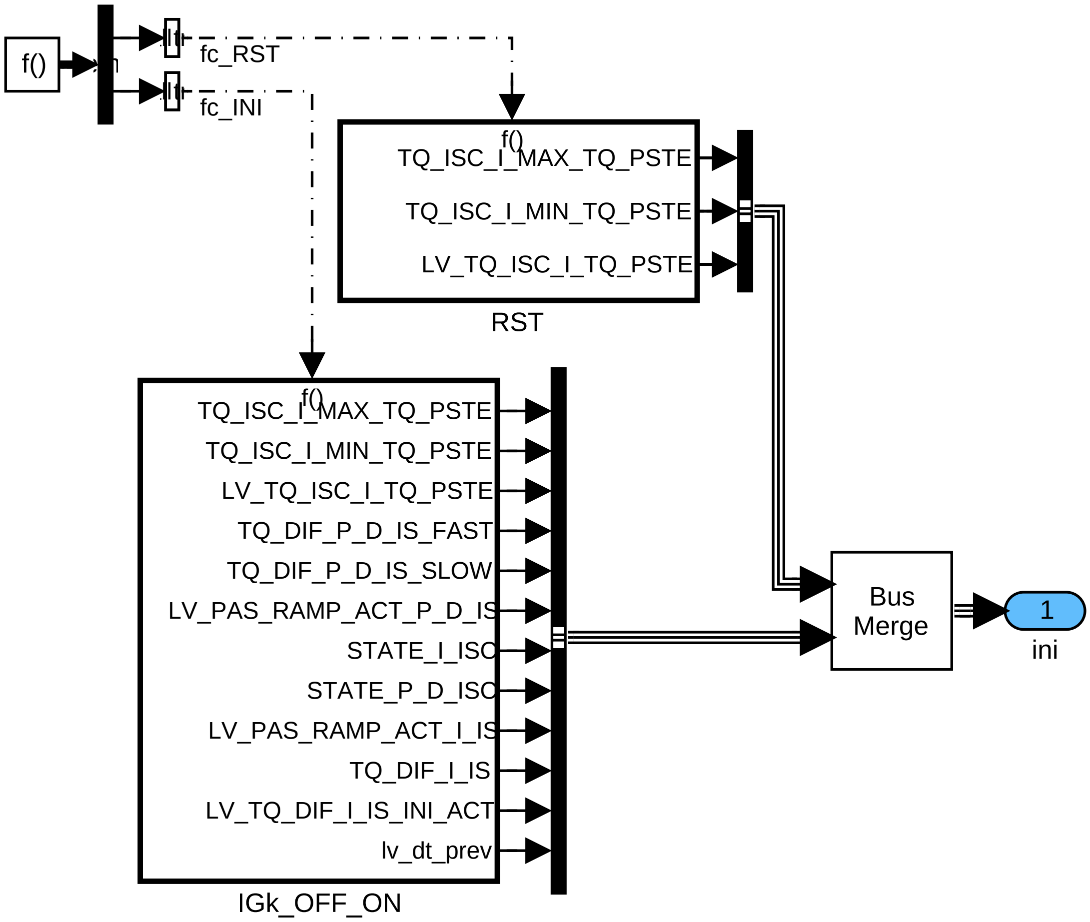


Figure 52 ENSC_M8006/ INI

Initialisation at igkon

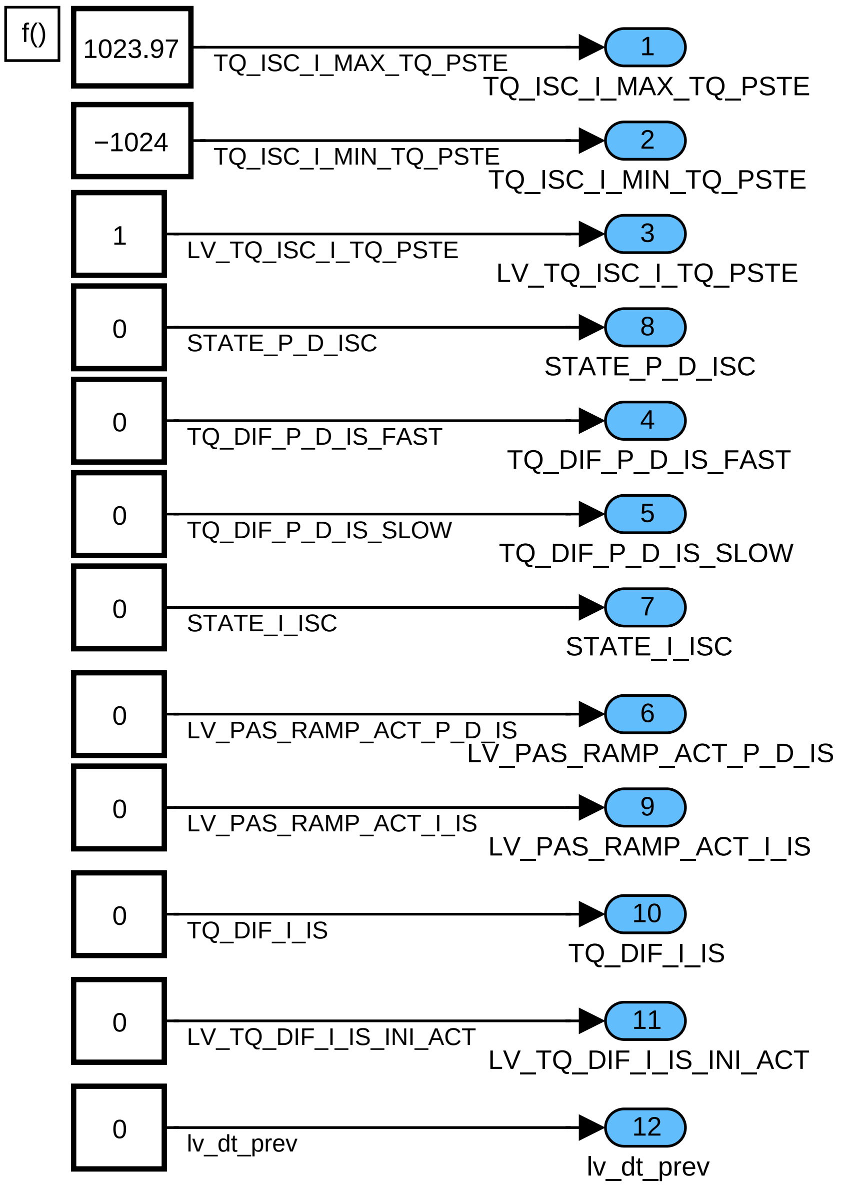


Figure 53 ENSC_M8006/ INI/ IGk_OFF_ON

Initialisation at system event reset

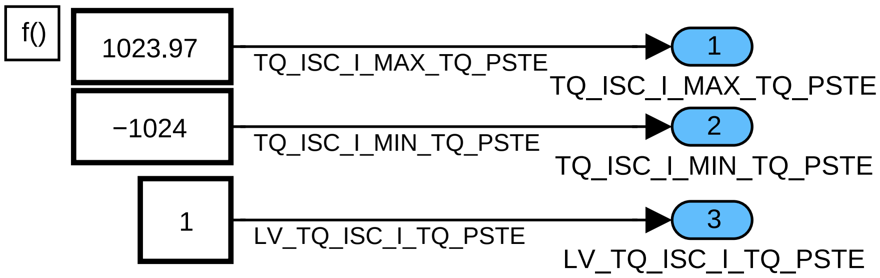


Figure 54 ENSC_M8006/ INI/ RST

22.4.1.4 Formula section

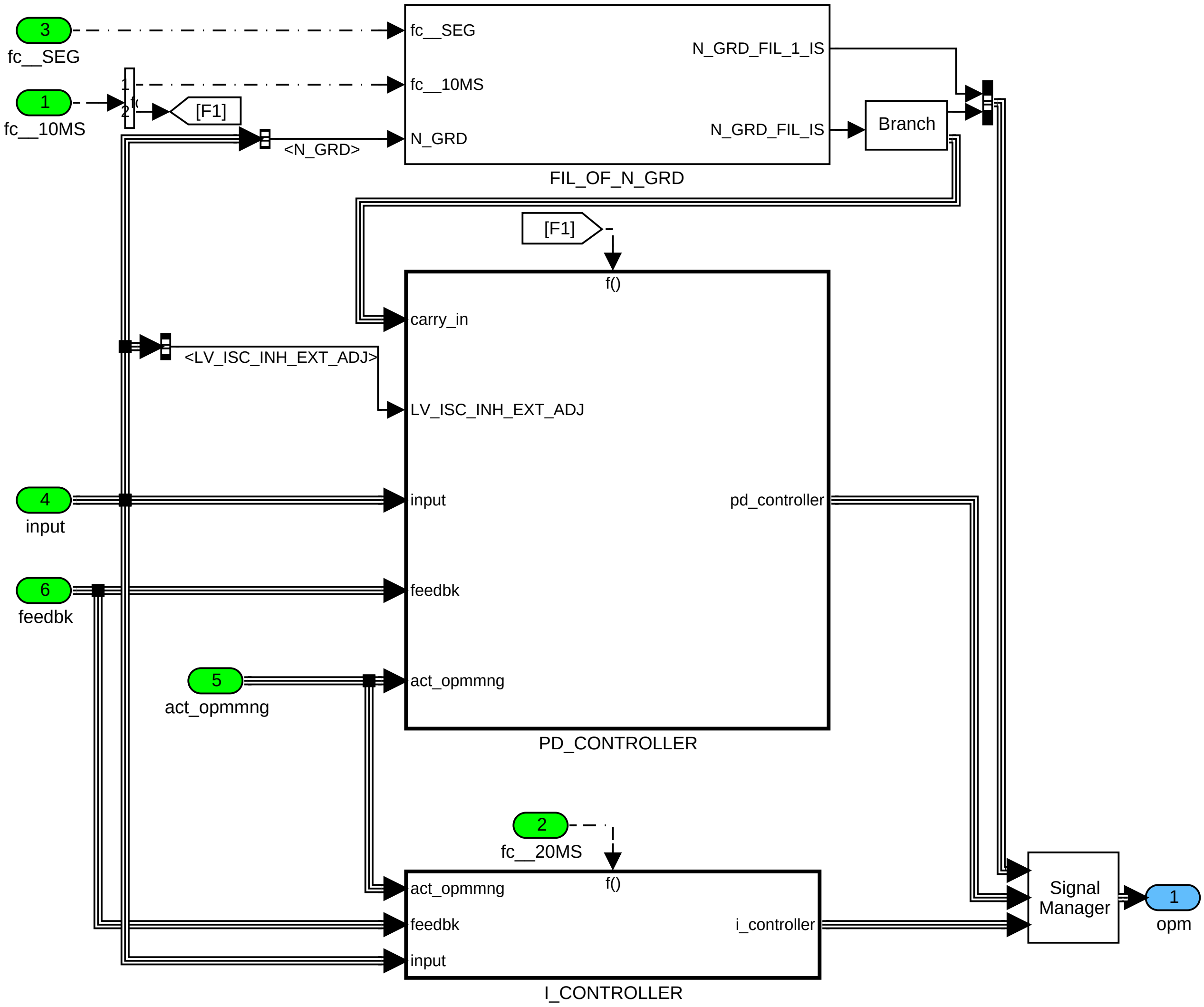


Figure 55 ENSC_M8006/ OPM

<u>Filtering of N_GRD</u>

General information:

In order to be able to smooth the gradient for the idle speed control, there are 2 options for filtering the raw engine speed gradient N_GRD:

1) with ring buffer with 1 ... 6 buffers (calibrateable)

2) with 1st order low-pass filter.

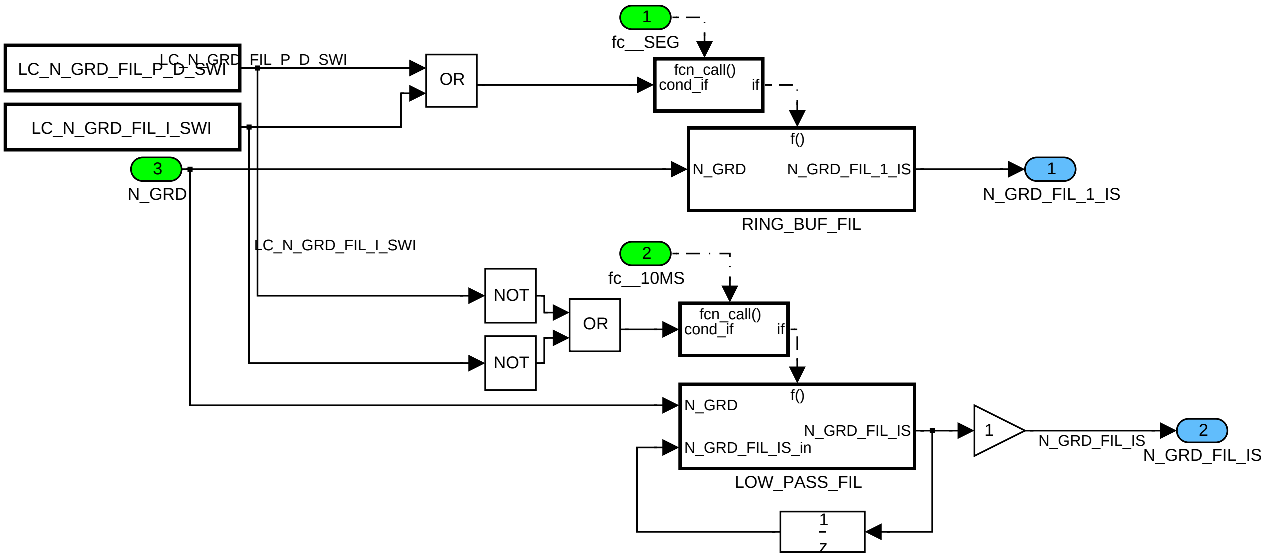


Figure 56 ENSC_M8006/ OPM/ FIL_OF_N_GRD

Low pass filtering of N_GRD at 10ms recurrence.

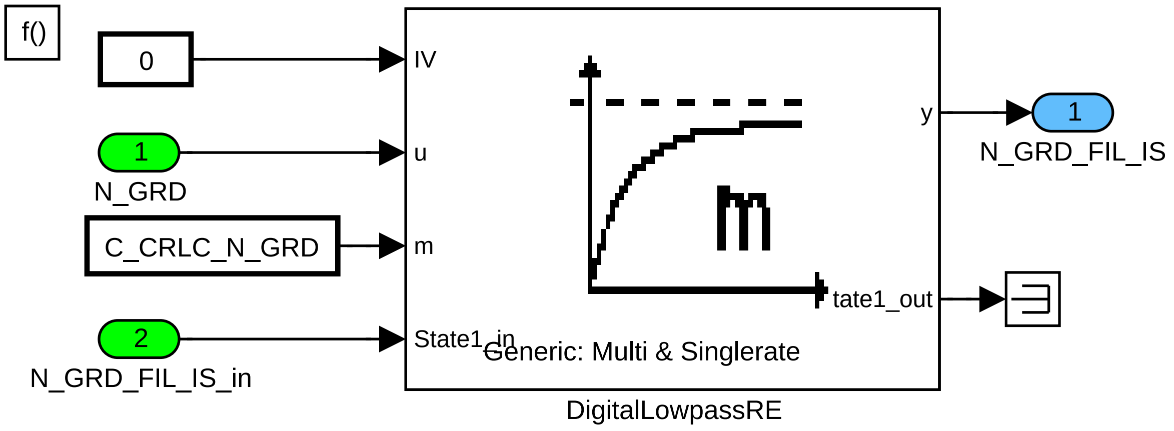


Figure 57 ENSC_M8006/ OPM/ FIL_OF_N_GRD/ LOW_PASS_FIL

Ring buffer filter at segment synchronous system recurrence

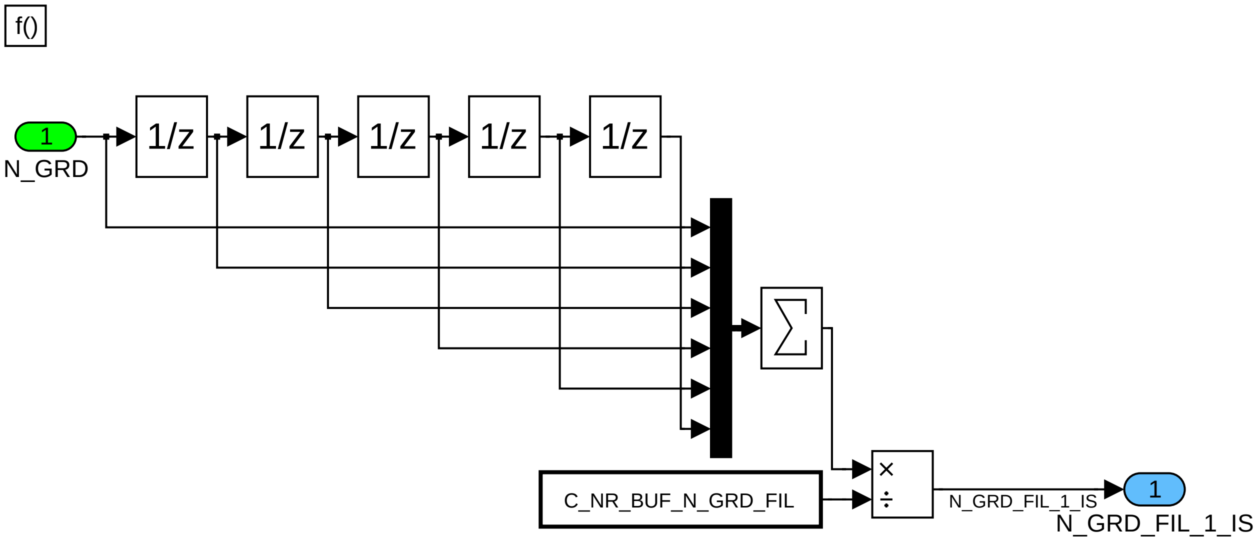


Figure 58 ENSC_M8006/ OPM/ FIL_OF_N_GRD/ RING_BUF_FIL

I-Controller

General information:

Signal flow diagram:

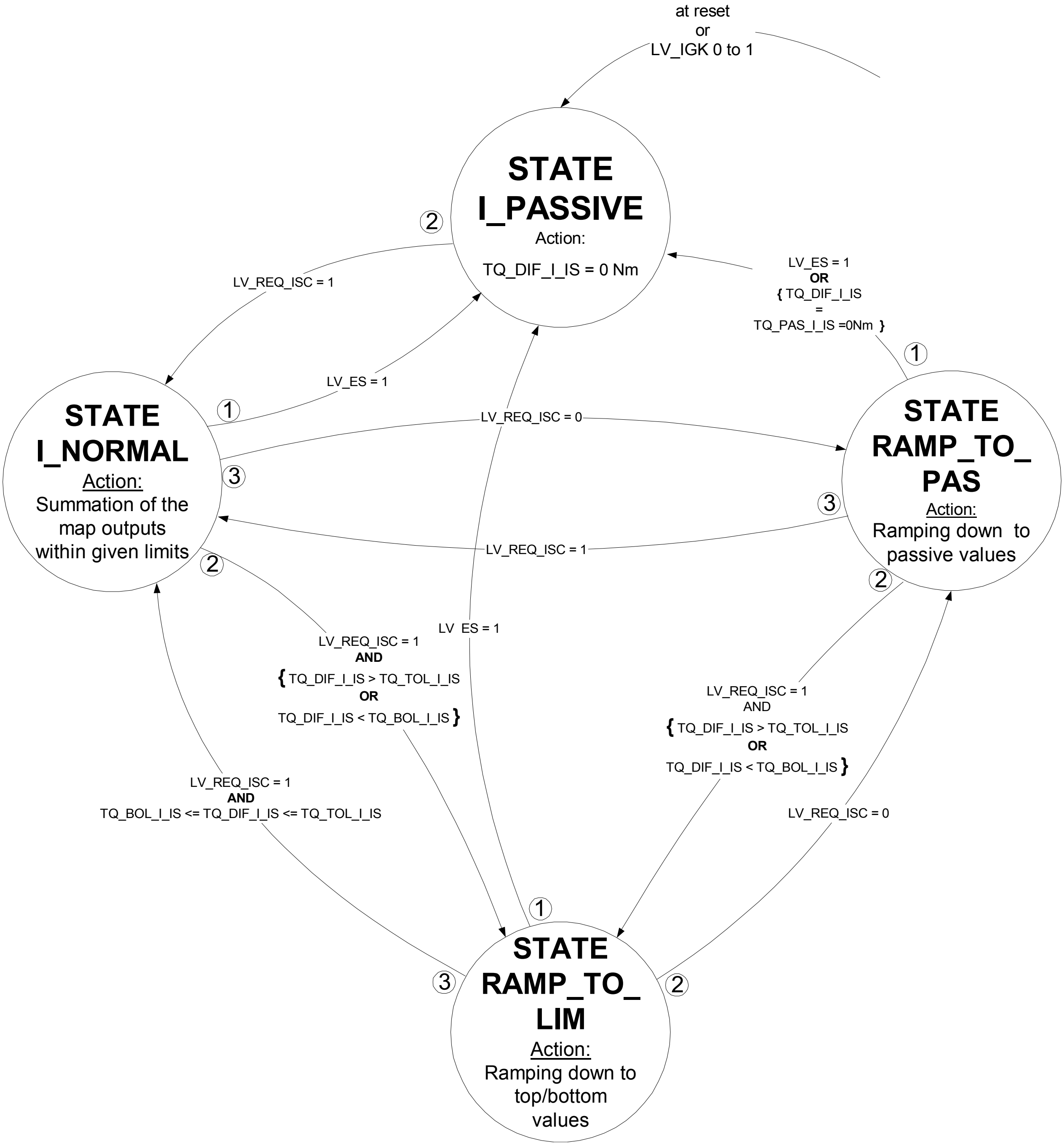


Figure 5: *State diagram of the integral part*

The integral part is devided into four states:

STATE_I_ISC = I_PASSIVE:

Initialization (at reset or when LV_IGK = 0 → 1; from all other states in case of LV_ES = 1)

STATE_I_ISC = I_NORMAL:

In this state the evaluation of the controller maps and the "normal" integration takes place.

STATE_I_ISC = RAMP_TO_LIM:

Ramp to limits; in this state the integral part is ramped down to the calibratible top/bottom limit values.

STATE_I_ISC = RAMP_TO_PAS:

Ramp to passive value 0 Nm; in this state the integral part is ramped down to zero in case LV_REQ_ISC = 0.

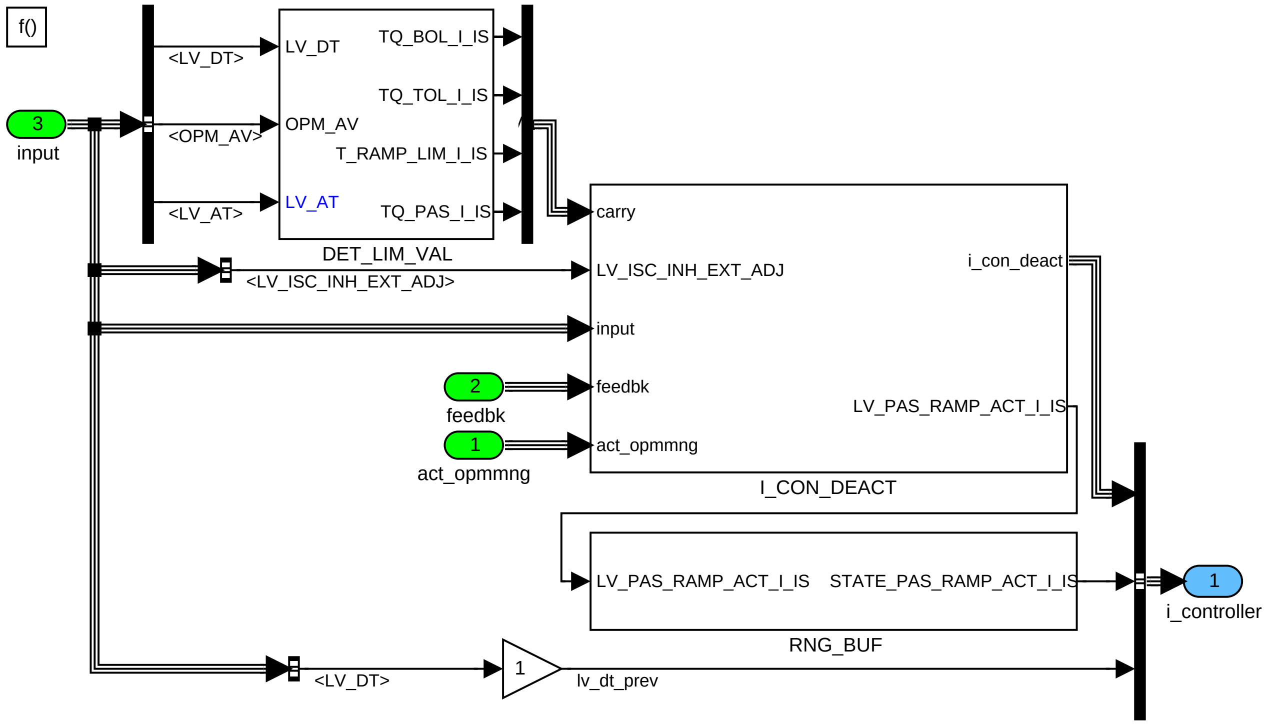


Figure 59 ENSC_M8006/ OPM/ I_CONTROLLER

Determination of the top/bottom limit values:

The passive value **(target value)** of the ramp operation is 0 Nm.

TQ_PAS_I_IS = 0Nm

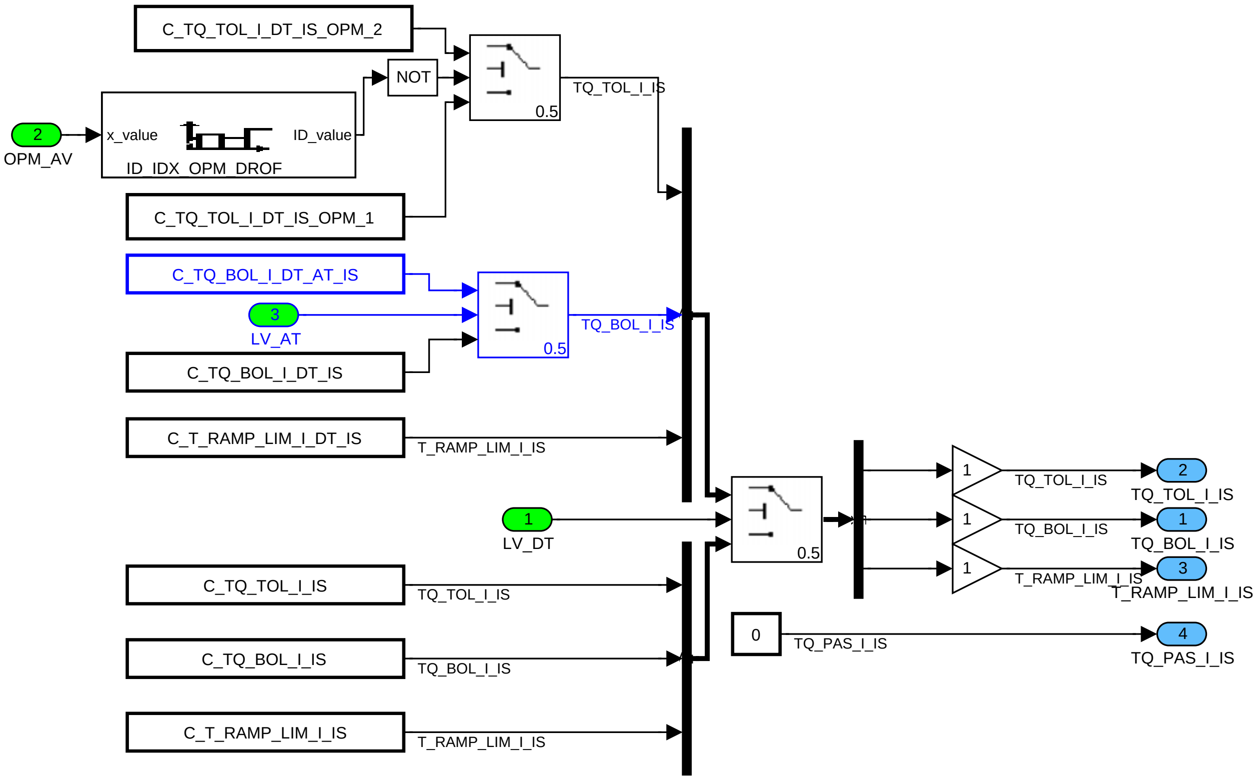


Figure 60 ENSC_M8006/ OPM/ I_CONTROLLER/ DET_LIM_VAL

Activation condition for I controller

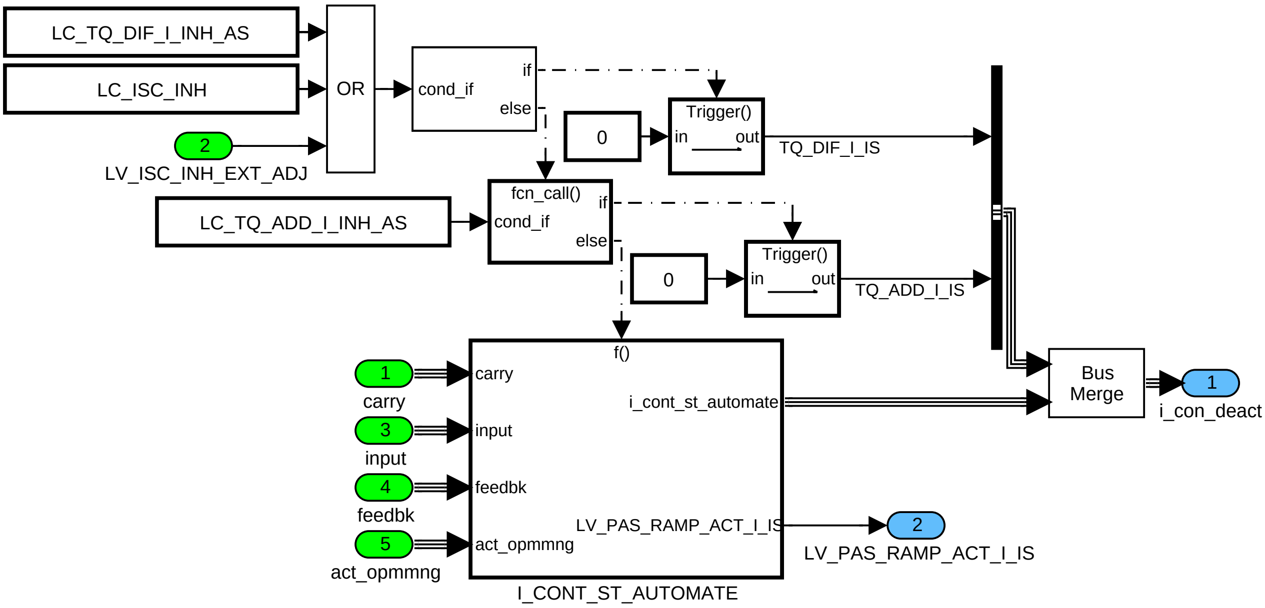


Figure 61 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT

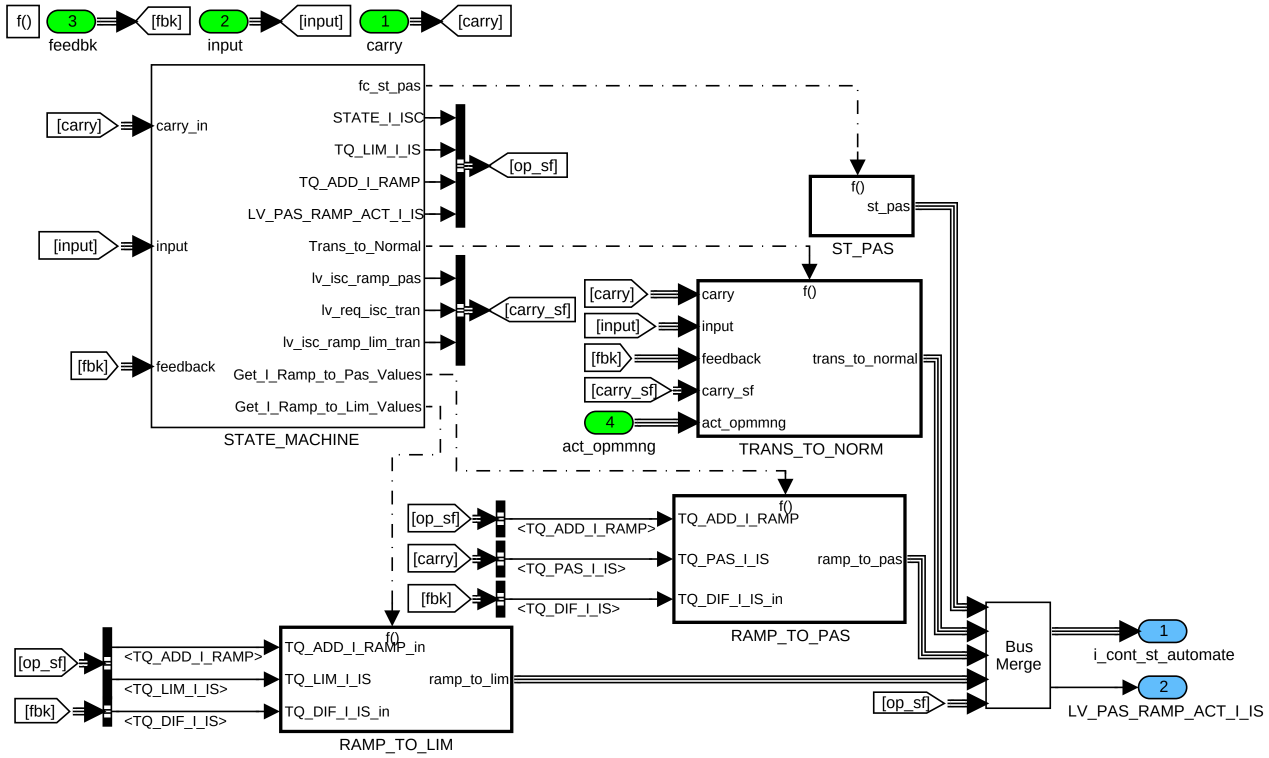


Figure 62 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE

State machine of I controller

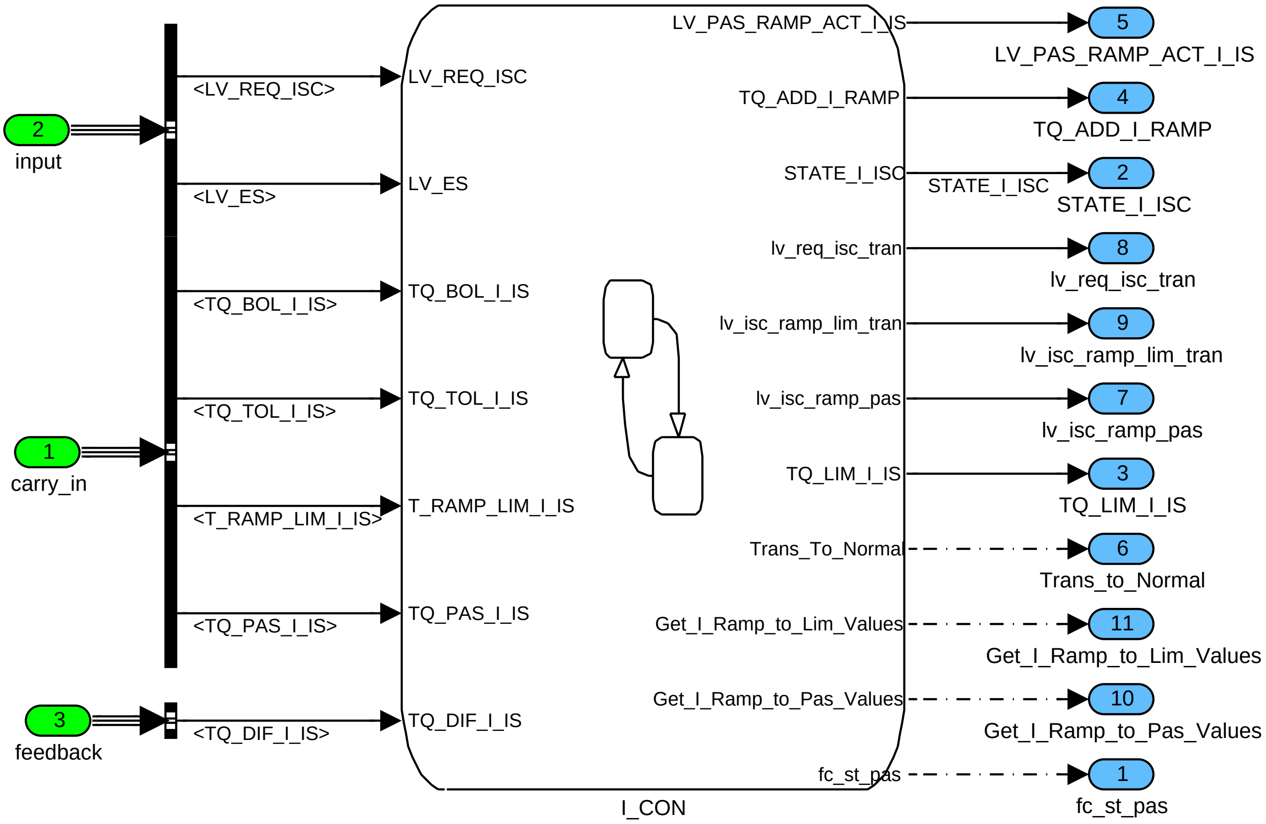


Figure 63 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ STATE_MACHINE

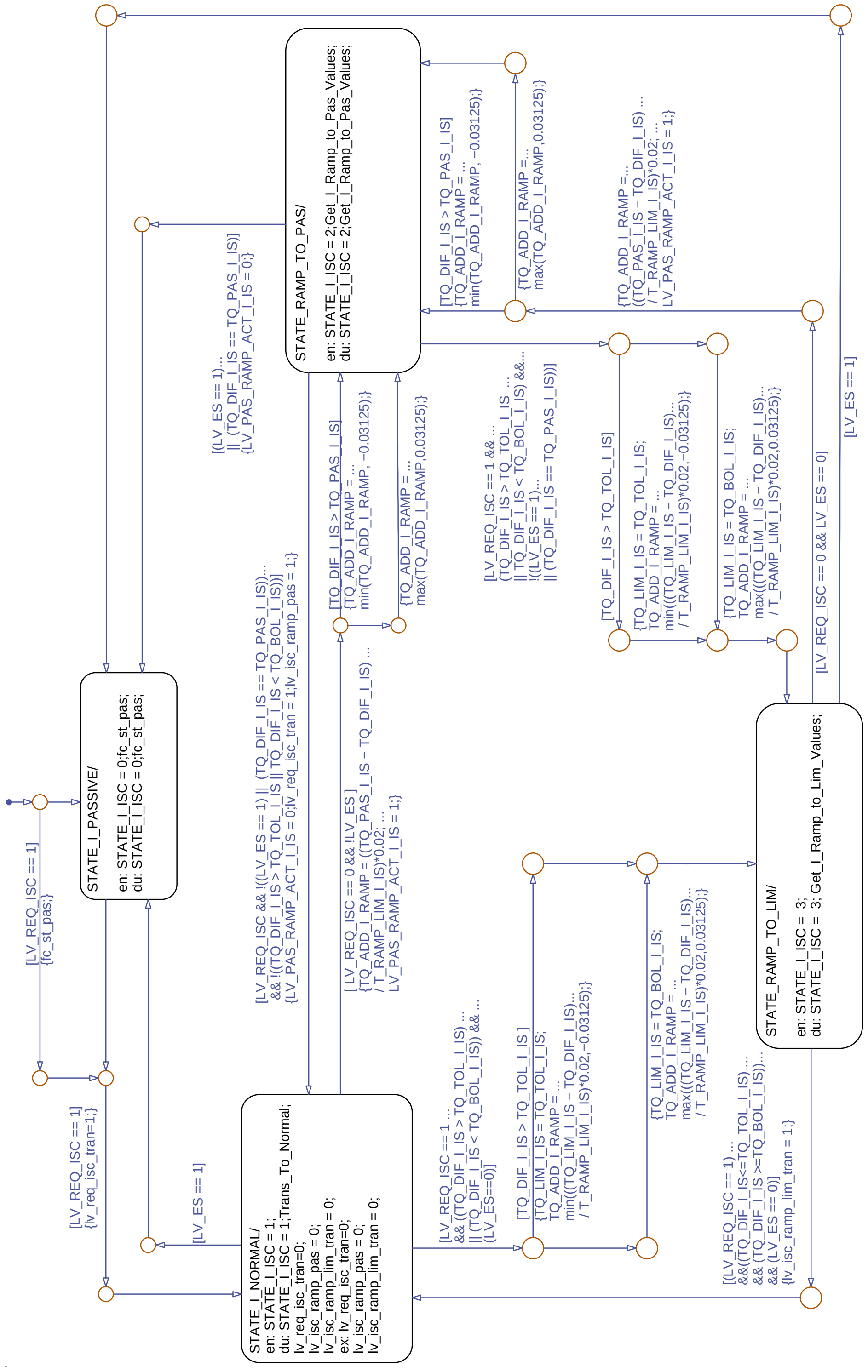


Figure 64 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ STATE_MACHINE/ I_CON

STATE_RAMP_TO_LIM

If this state is active and so the integrator output is outside the allowed boundaries, its value is controlled back to the valid boundaries on a ramp.

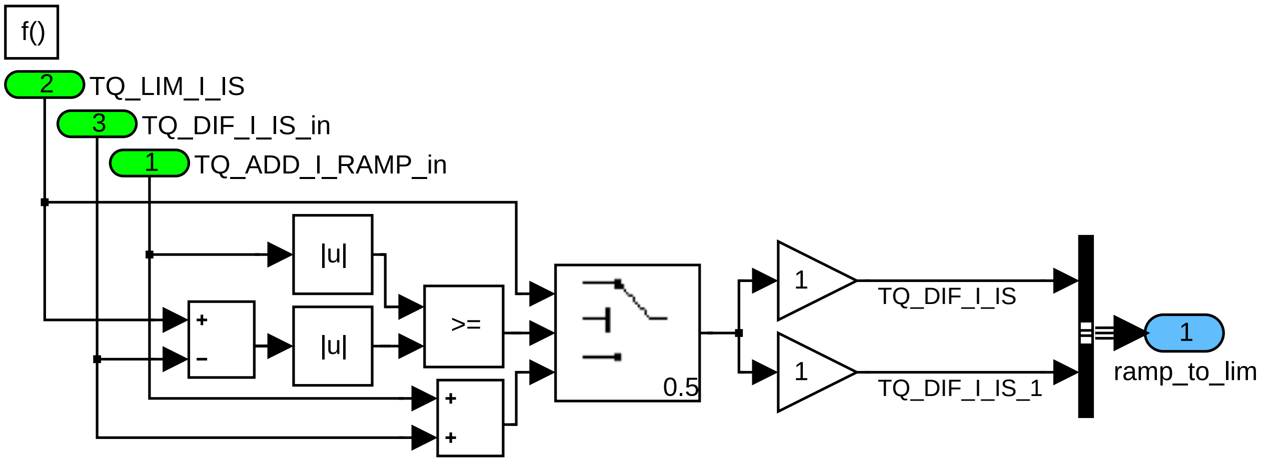


Figure 65 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ RAMP_TO_LIM

Calculation at state passive

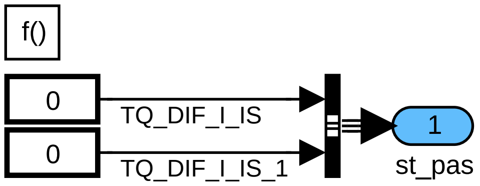
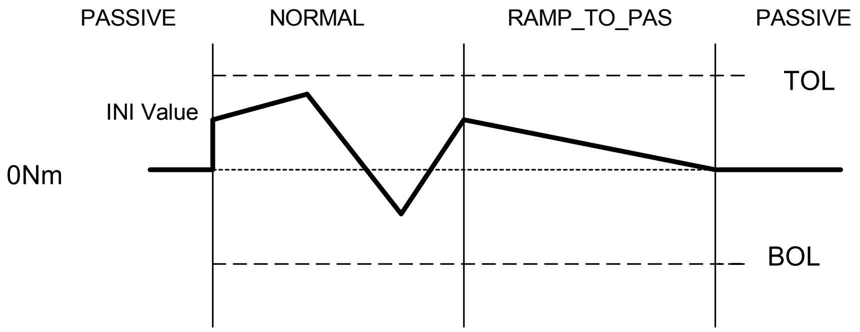


Figure 66 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ ST_PAS

Overview of calculations at normal state in state machine

If idle speed control is reacitvated during the ramp limit operation then the integrator starts with "first setting of TQ_DIF_I_IS value (see. calculation of I_NORMAL).

Excample of I share calculation without limitation



Excample of I share calculation with limitation

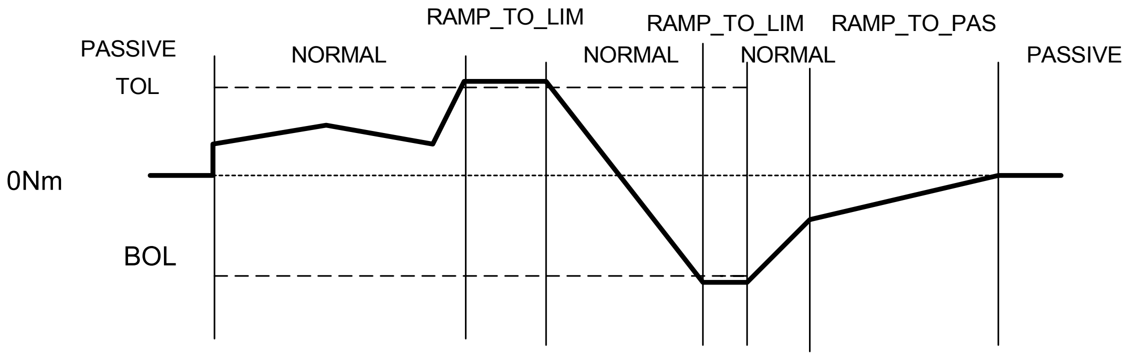


Figure 8: Principle operation of I share

Selection of N_GRD_I_IS

Calculation N_GRD_I_IS.

N_GRD_I_IS is calculated from N_GRD_FIL_IS and N_GRD_FIL_1_IS.

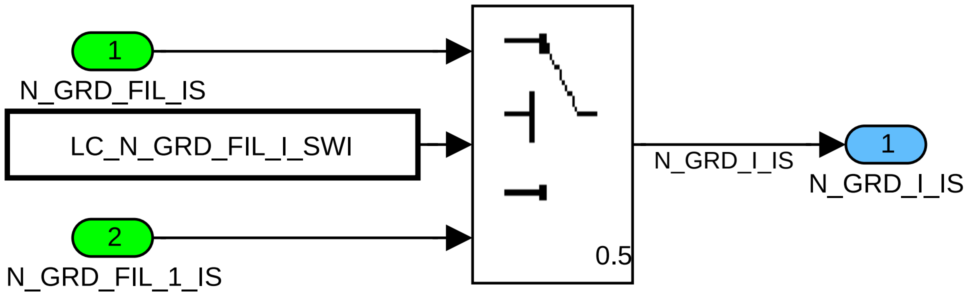


Figure 67 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_N_GRD_I_IS/ CLC

Check for transition from ramp to pas

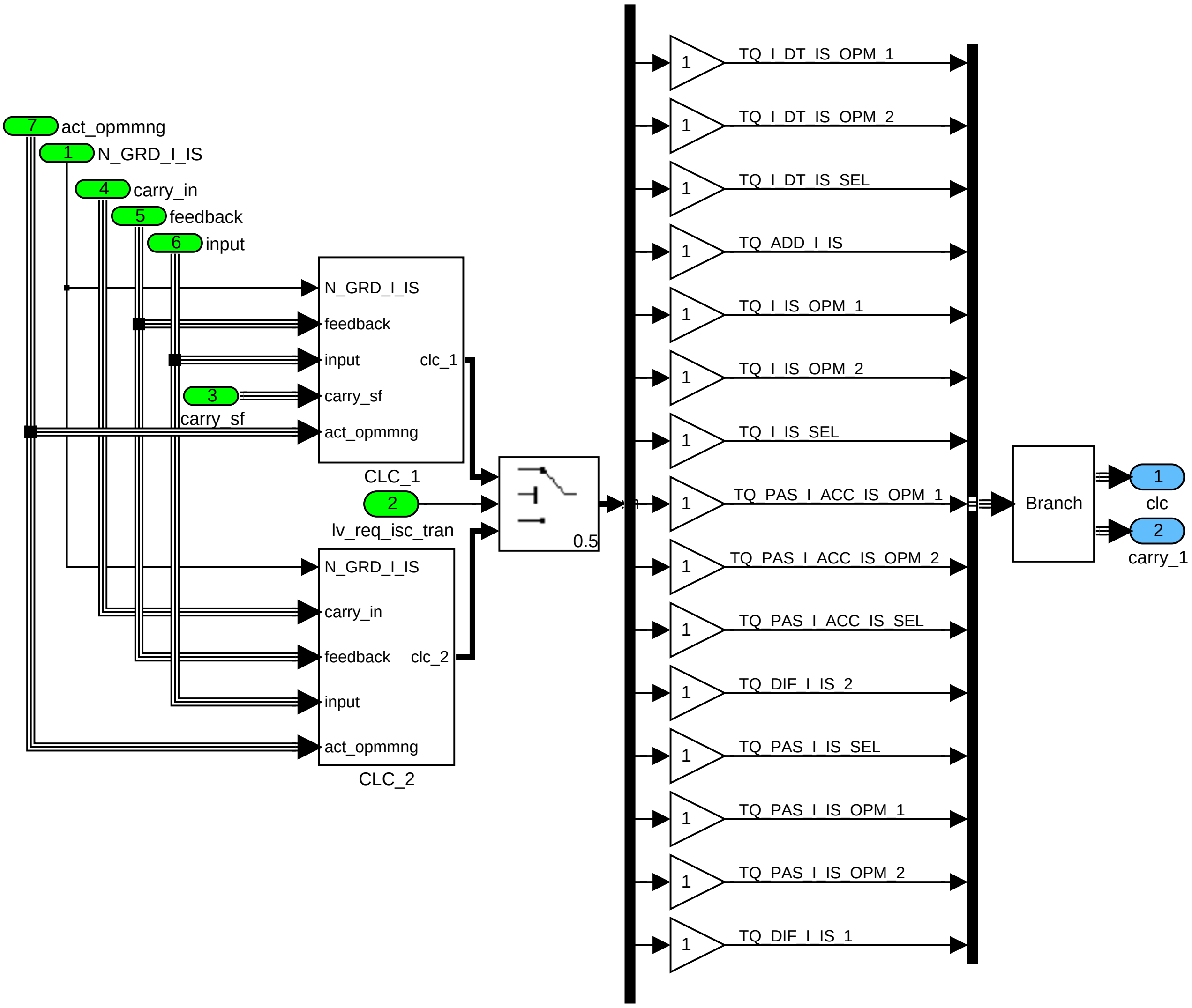


Figure 68 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC

Initialisaton of I share

Feed through

Old values are passed as it is.

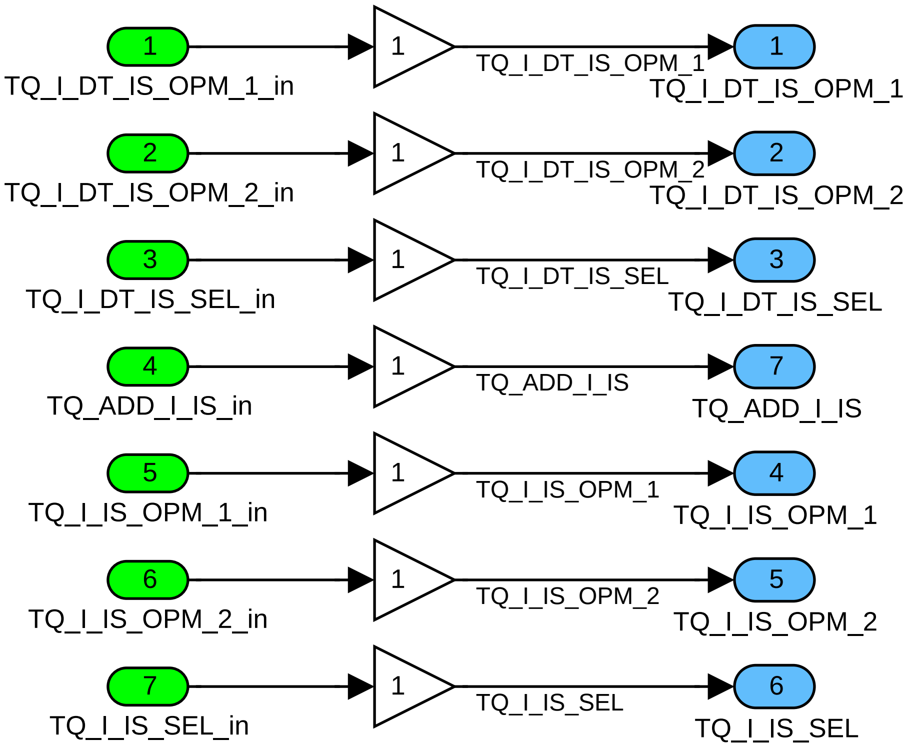


Figure 69 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_1/ DEFAULT

Condition check on LV_ACCOUT_RLY variable.

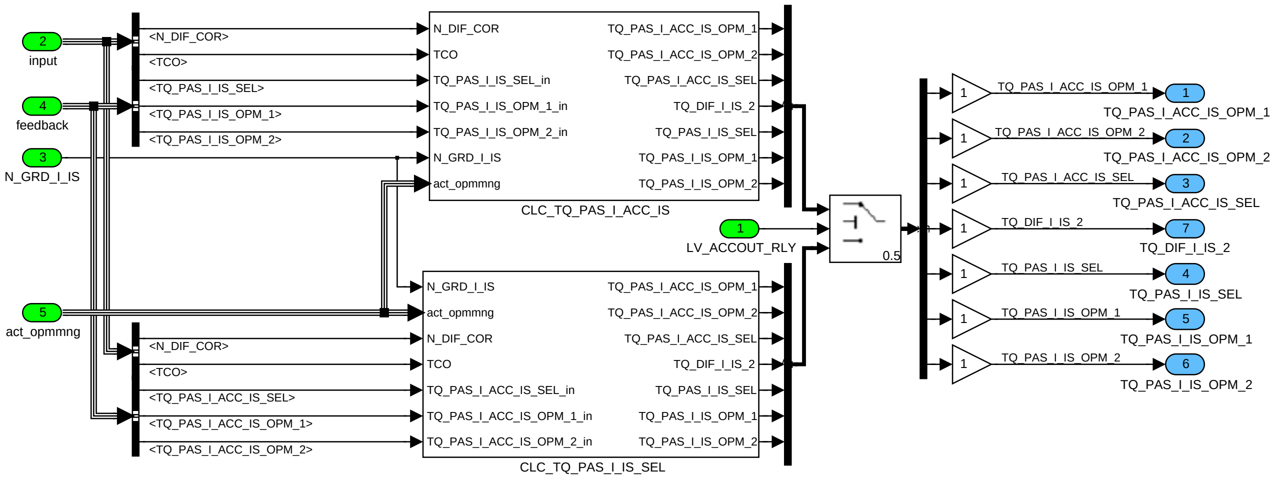


Figure 70 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_1/ CLC

Calculation of TQ_PAS_I_IS_SEL and TQ_DIF_I_IS_2 when condition check LV_ACCOUT_RLY is false.

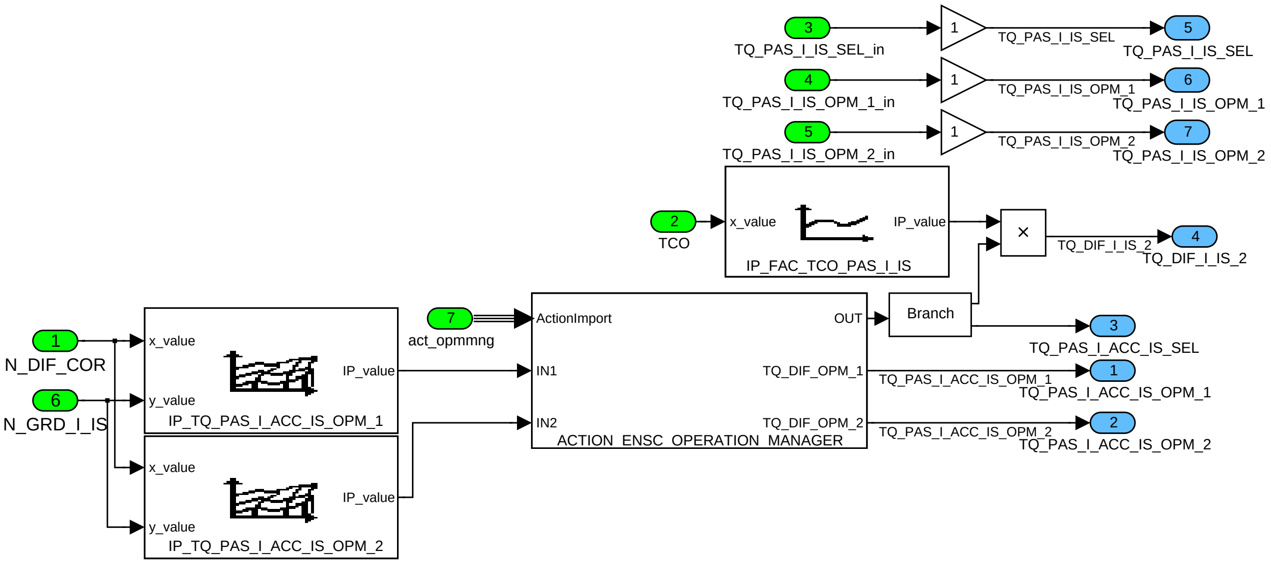


Figure 71 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_1/ CLC/ CLC_TQ_PAS_I_ACC_IS

Calculation of TQ_PAS_I_ACC_IS_SEL and TQ_DIF_I_IS_2 when condition check LV_ACCOUT_RLY is true.

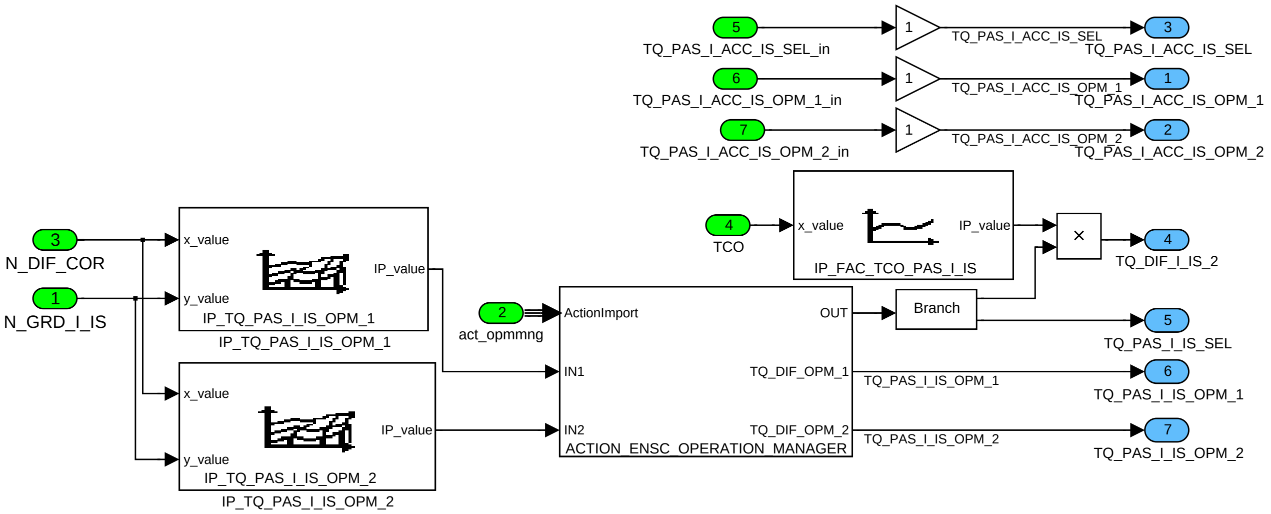


Figure 72 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_1/ CLC/ CLC_TQ_PAS_I_IS_SEL

Calculation of TQ_DIF_I_IS_1

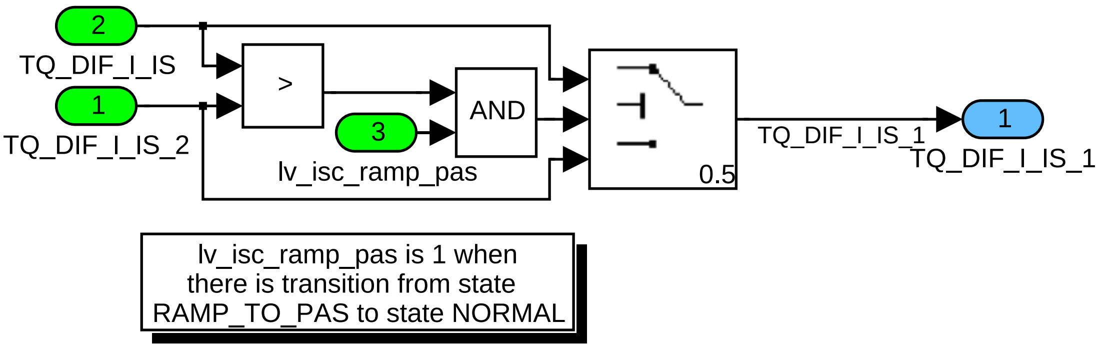


Figure 73 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_1/ CLC_TQ_DIF_I_IS_1

Calculation when there is no transition from passive or "ramp to pas" to normal state.

Condition check on LV_DT and determination of TQ_ADD_I_IS

In STATE I_NORMAL, see figure 5, the "normal" integration takes place. A torque difference TQ_ADD_I_IS is determined in dependence of the speed deviation N_DIF_COR and the speed gradient N_GRD_I_IS (for LV_DT = 0) and only in dependance of the speed deviation N_DIF_COR for LV_DT = 1 and in addition depending on time after start T_AST, figure 6.

Considering the operation mode manager there are included separated paths in the functionality for the calculation of the idle speed controller I-share. This enables a proper engine running in throttled and unthrottled mode. Also in after start an additional factor is used for each mode.

To reach two important targets, TQ_ADD_I_IS should be calibrated as follows:

compensation of a speed offset

automatic precalibration of the integral part: if there is a load active after the driver released the gas pedal (ACC, power steering, etc), then speed falls with a higher gradient into idle range than in comparision to the situation without load. This means, that the integral part should carry higher (positive) values to intercept speed. That means, that with the help of the gradient dependancy in the integral part an automatic precalibration of the TQ_ADD_I_IS (so TQ_DIF_I_IS) can be reached, which is load dependant. It must be said, that this interception function must be synchronized with the PD-part calibration. With the correct calibration, the passive values are not needed any more.

The integration takes place within calibratible boundaries, figure 7.

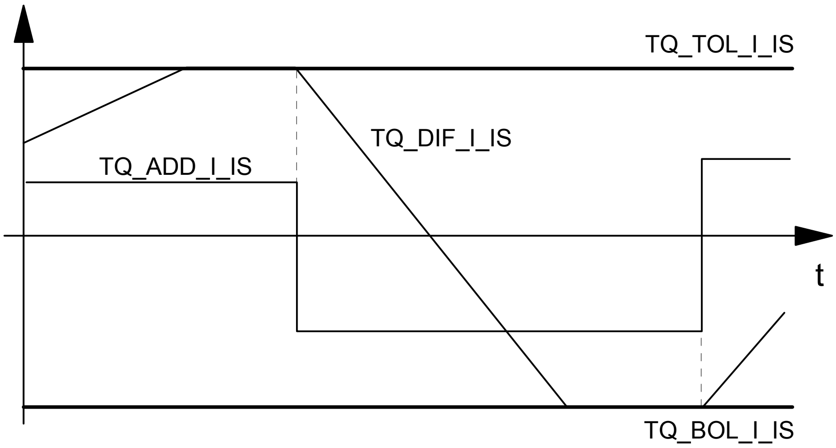


Figure 7: *Limited integration*

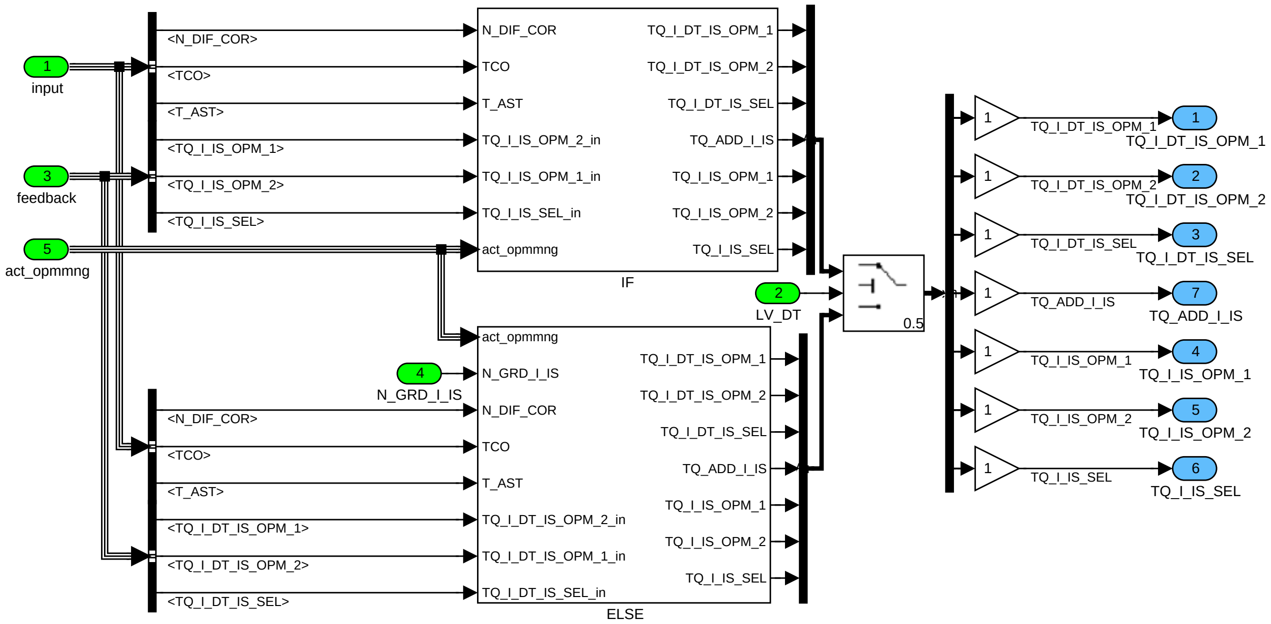


Figure 74 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_2/ CLC_1

Calculation of TQ_ADD_I_IS, TQ_I_DT_IS_SEL, TQ_I_DT_IS_OPM_1 and TQ_I_DT_IS_OPM_2

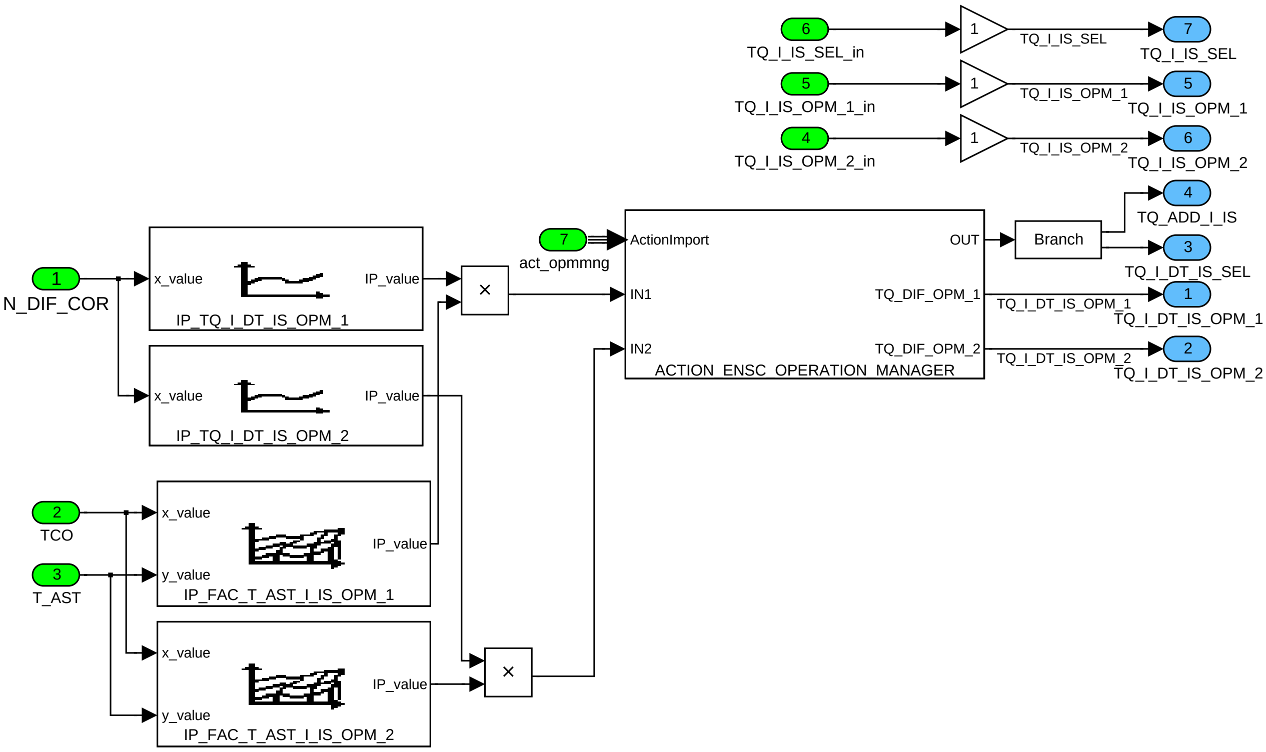


Figure 75 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_2/ CLC_1/ IF

Calculation of TQ_ADD_I_IS, TQ_I_IS_SEL, TQ_I_IS_OPM_1 and TQ_I_IS_OPM_2

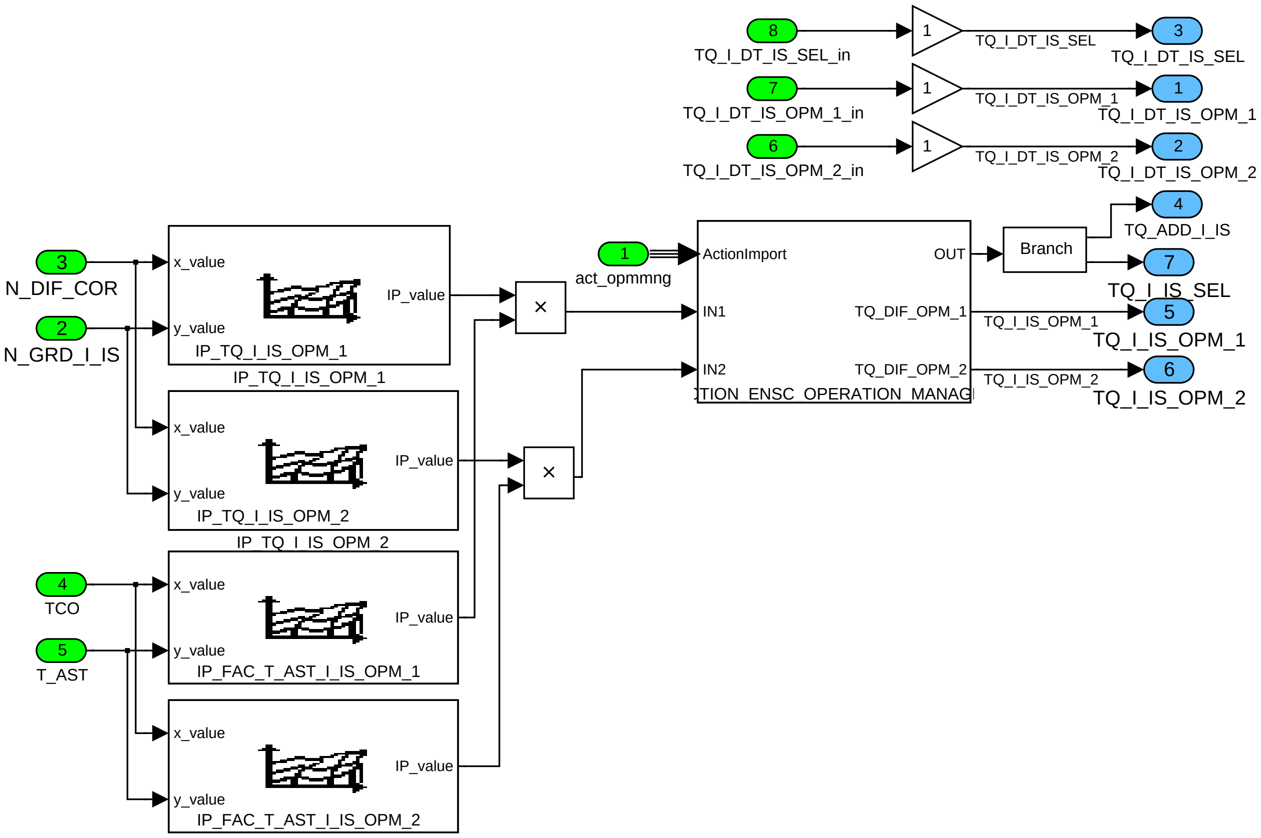


Figure 76 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_2/ CLC_1/ ELSE

Calculation of TQ_DIF_I_IS_1 as an ouput of integrator

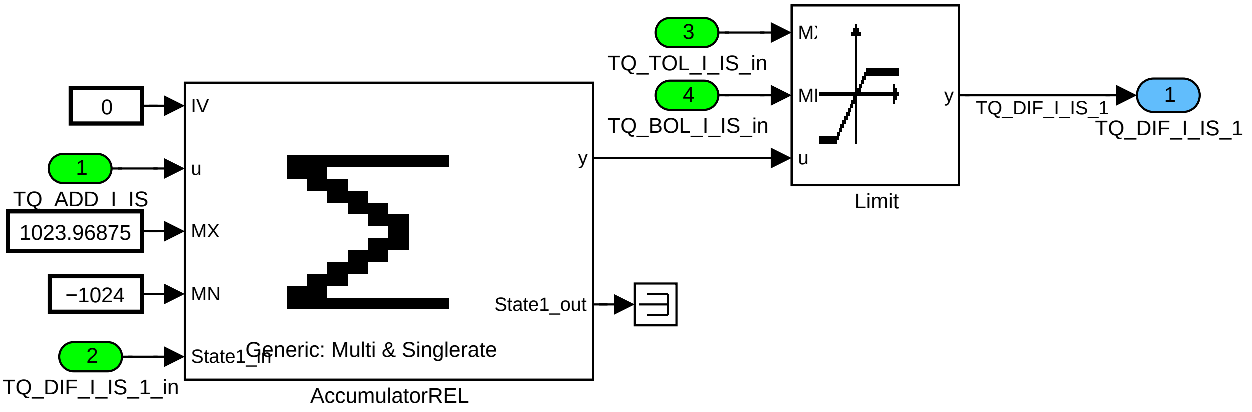


Figure 77 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_2/ CLC_2

Old value assignment

No calculations.

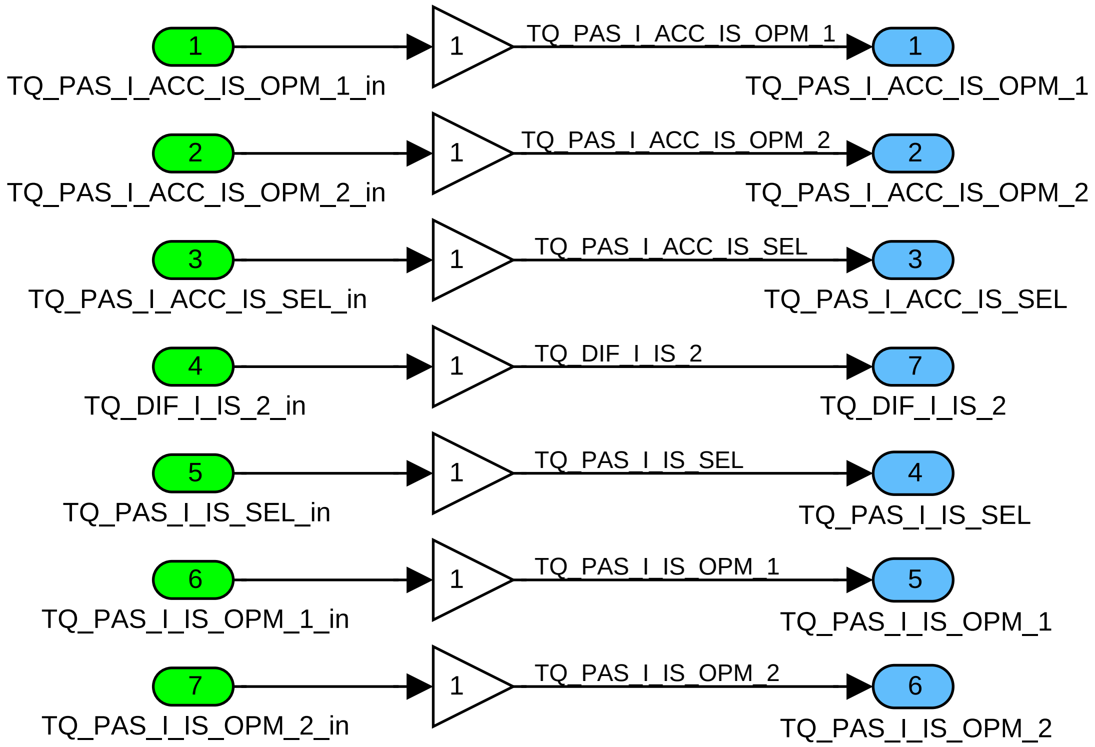


Figure 78 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC/ CLC_2/ DEFAULT

Calculation of PSTE limitation

Calculation of T_TQ_ISC_I_TQ_PSTE_LIM

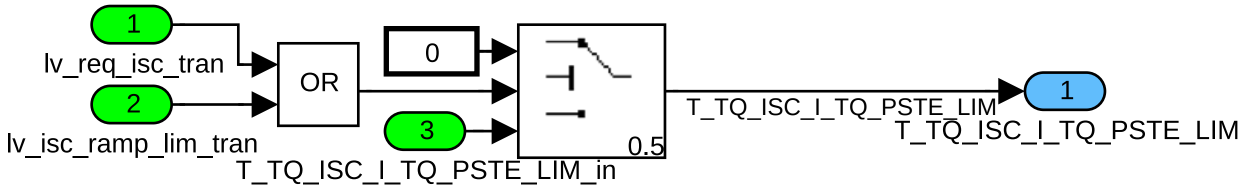


Figure 79 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_PSTE_LIM/ CLC

Condition check for Calculation of TQ_ISC_I_MAX/MIN_TQ_PSTE and Lock time for pste limitation

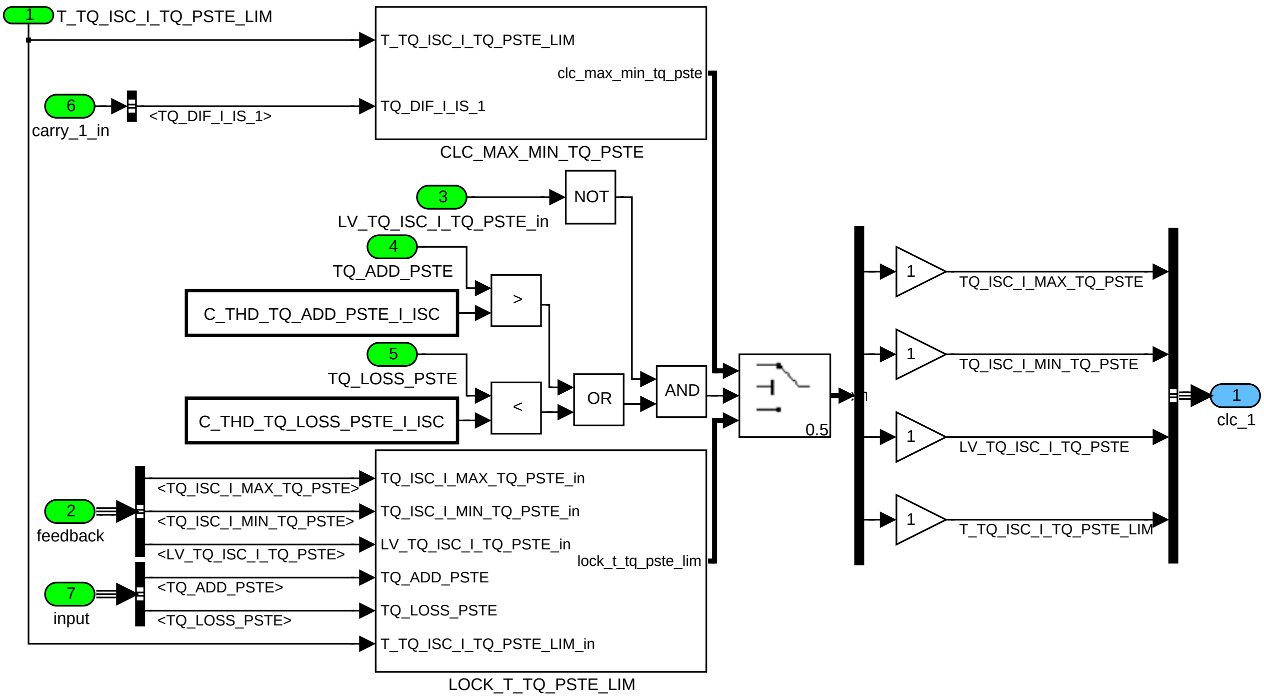


Figure 80 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_PSTE_LIM/ CLC_1

Calculation of TQ_ISC_I_MAX/MIN_TQ_PSTE

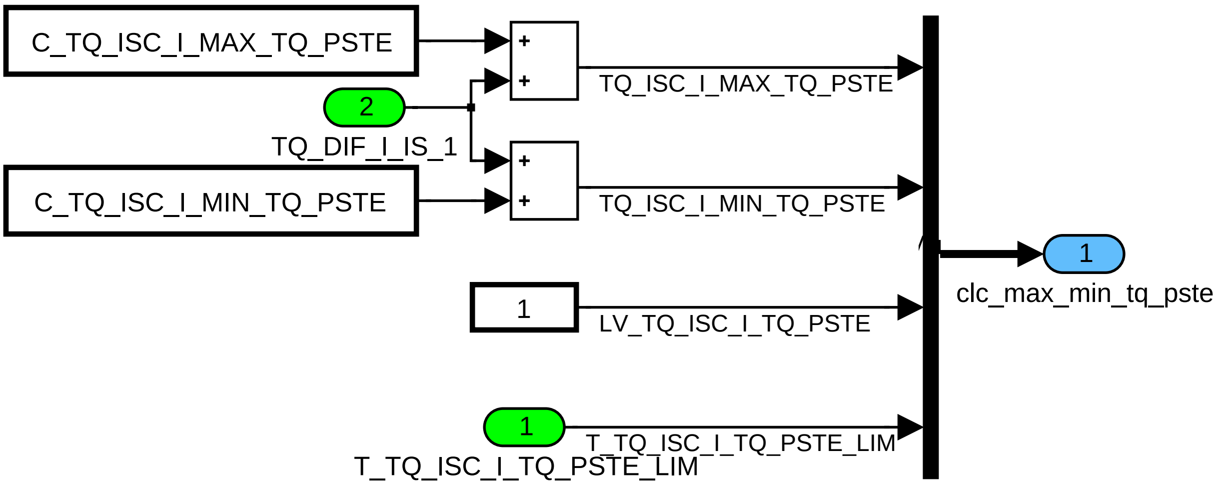


Figure 81 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_PSTE_LIM/ CLC_1/ CLC_MAX_MIN_TQ_PSTE

Lock time for pste limitation

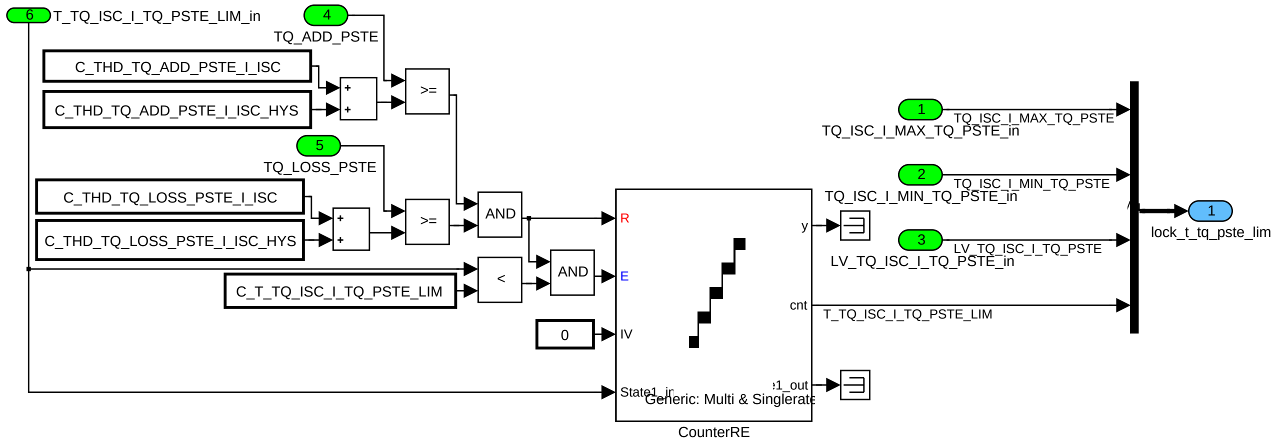


Figure 82 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_PSTE_LIM/ CLC_1/ LOCK_T_TQ_PSTE_LIM

Calculation for PSTE and DROF limitation

Condition check for calculation at PSTE and DROF limitation

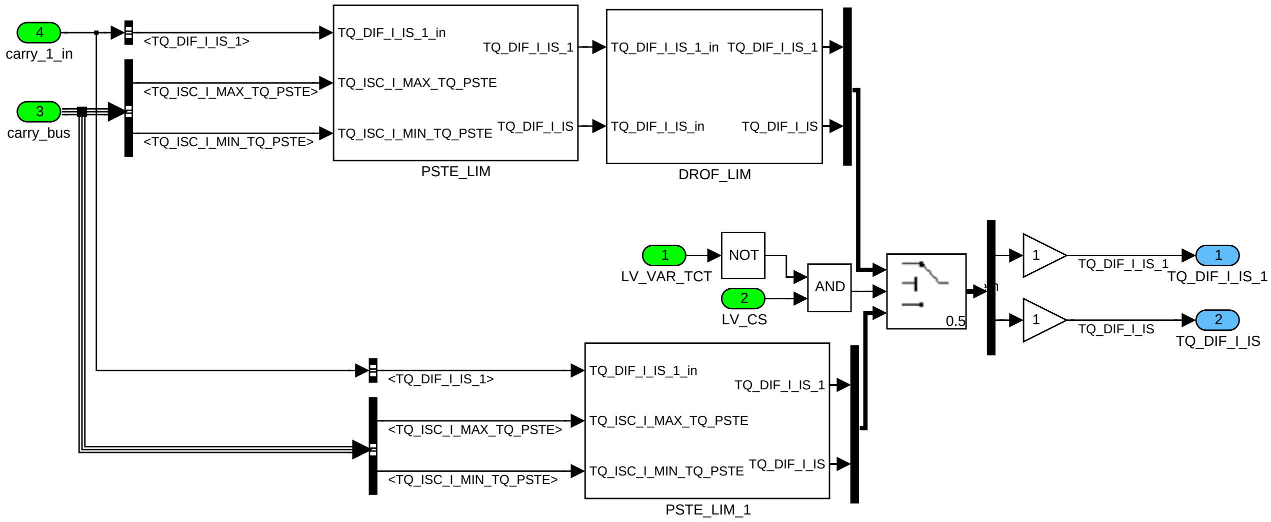


Figure 83 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_PSTE_DROF_LIM/ CLC

Condition true---->PSTE part of PSTE-DROF limitation is applied

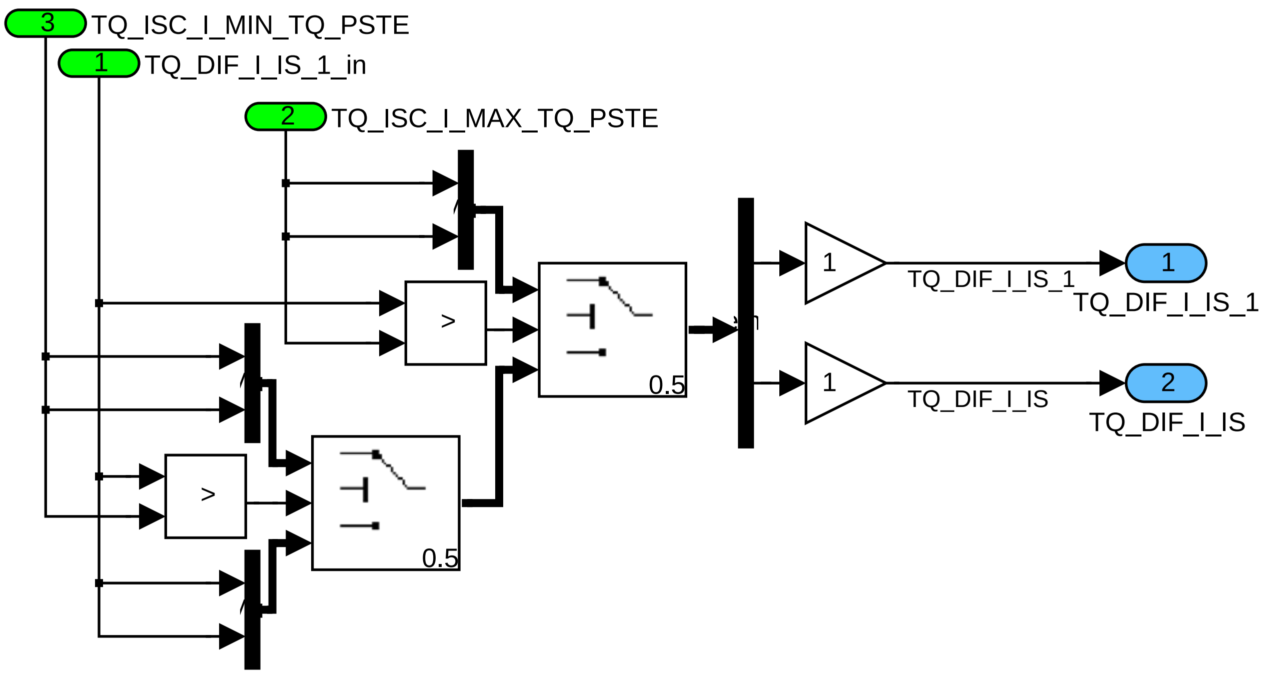


Figure 84 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_PSTE_DROF_LIM/ CLC/ PSTE_LIM

Condition true---->DROF part of PSTE-DROF limitation is applied

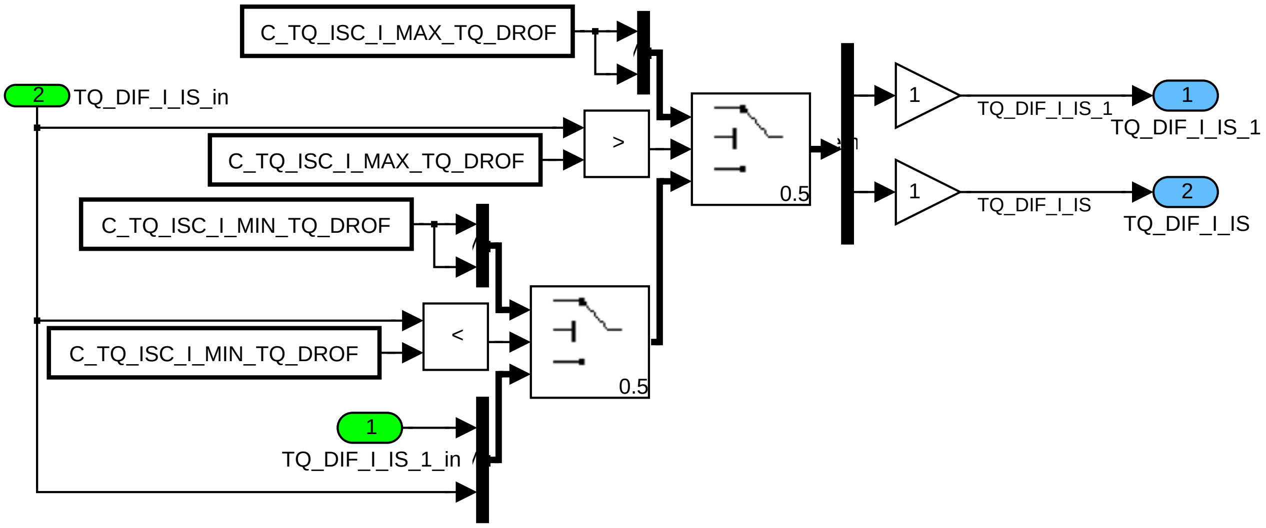


Figure 85 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_PSTE_DROF_LIM/ CLC/ DROF_LIM

Condition false---> Only PSTE limitation applied

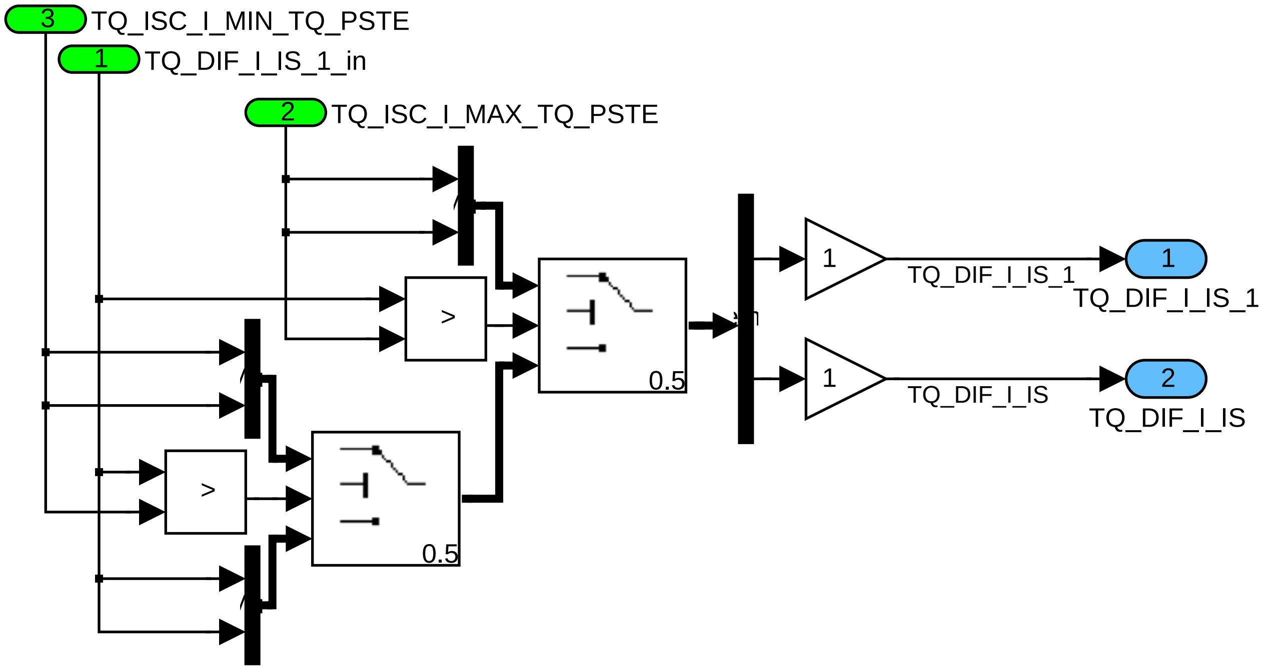


Figure 86 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ CLC_PSTE_DROF_LIM/ CLC/ PSTE_LIM_1

Re-initialization of TQ_DIF_I_IS

Calculation of TQ_DIF_I_IS

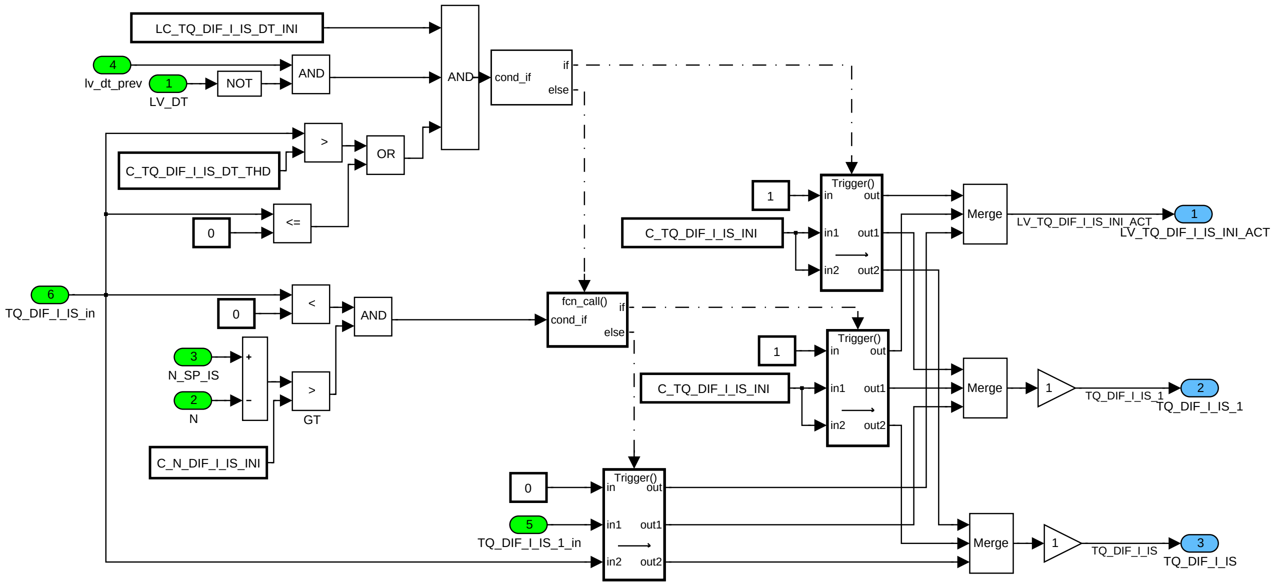


Figure 87 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ TRANS_TO_NORM/ RE_INI_TQ_DIF_I_IS/ CLC_TQ_DIF_I_IS

STATE_RAMP_TO_PAS

At deactivation of ISC the integrator output is controlled to a passive-value by ramp limit activation. The determination of the ramp takes place in the same way as described before for the top/bottom limit ramps.

Actions:

The ramp limit operation starts at deactivation of the ISC. The necessary increment (**TQ_ADD_I_RAMP**) to reach the required ramp-slope **is only calculated once while transitioning into this state**. This value is valid until deactivation of the ramp limit operation.

While transition into this state, so while the ramp is active, there could be a change in TQ_PAS_I_IS. If in this case TQ_ADD_I_RAMP is negative for example and the passive value is higher than the actual value, then the new requested passive value can never be reached, because TQ_ADD_I_RAMP is only calculated once. Therefore TQ_PAS_I_IS is fixed while transition into this state to prevent a wrong comparison in the following condition:

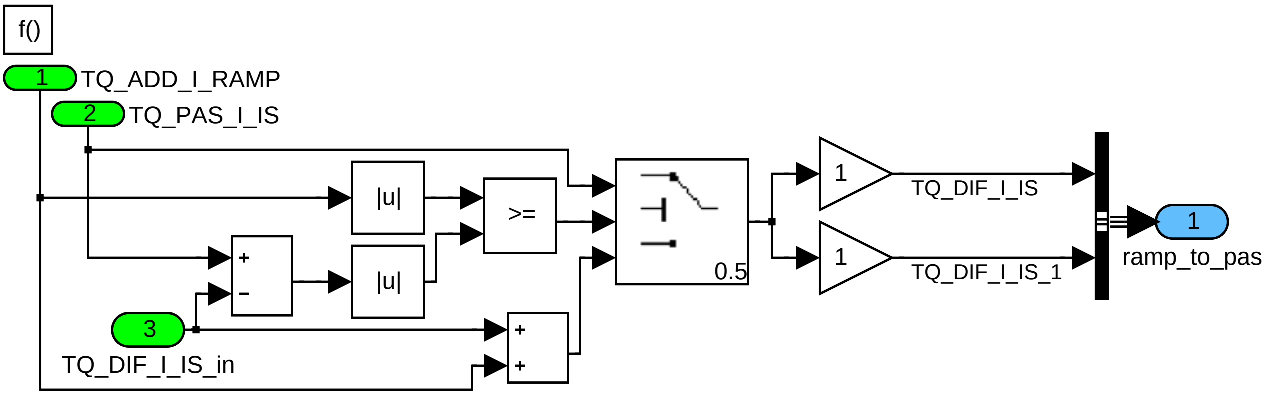
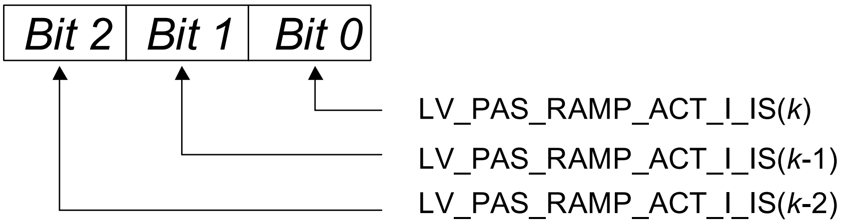


Figure 88 ENSC_M8006/ OPM/ I_CONTROLLER/ I_CON_DEACT/ I_CONT_ST_AUTOMATE/ RAMP_TO_PAS

Circular buffer for LV_PAS_RAMP_ACT_I_IS

In the following circular buffer STATE_PAS_RAMP_ACT_I_IS, the last three values of LV_PAS_RAMP_ACT_I_IS are stored:



This information is needed for the monitoring of the idle speed controller. It is updated every 20ms; the values are shifted from Bit j to Bit j+1, j=0,1.

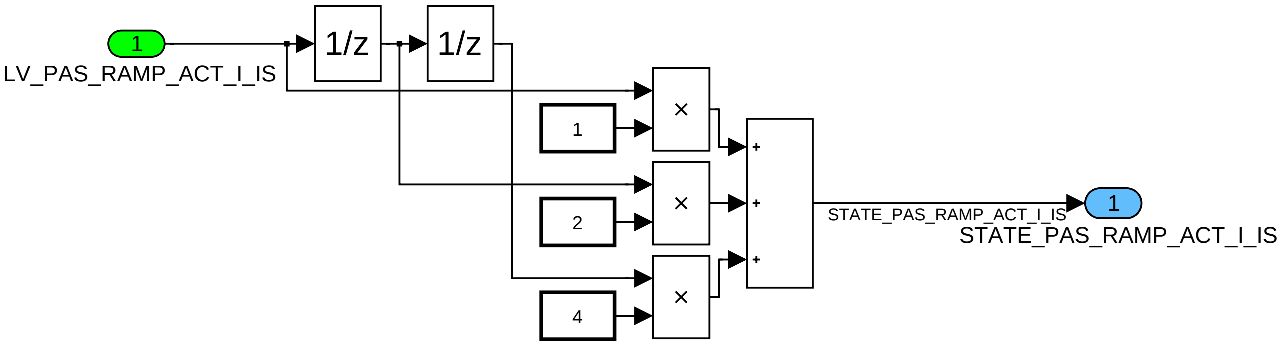


Figure 89 ENSC_M8006/ OPM/ I_CONTROLLER/ RNG_BUF

PD- Controller

General information:

State Machine

FUNCTION DESCRIPTION:

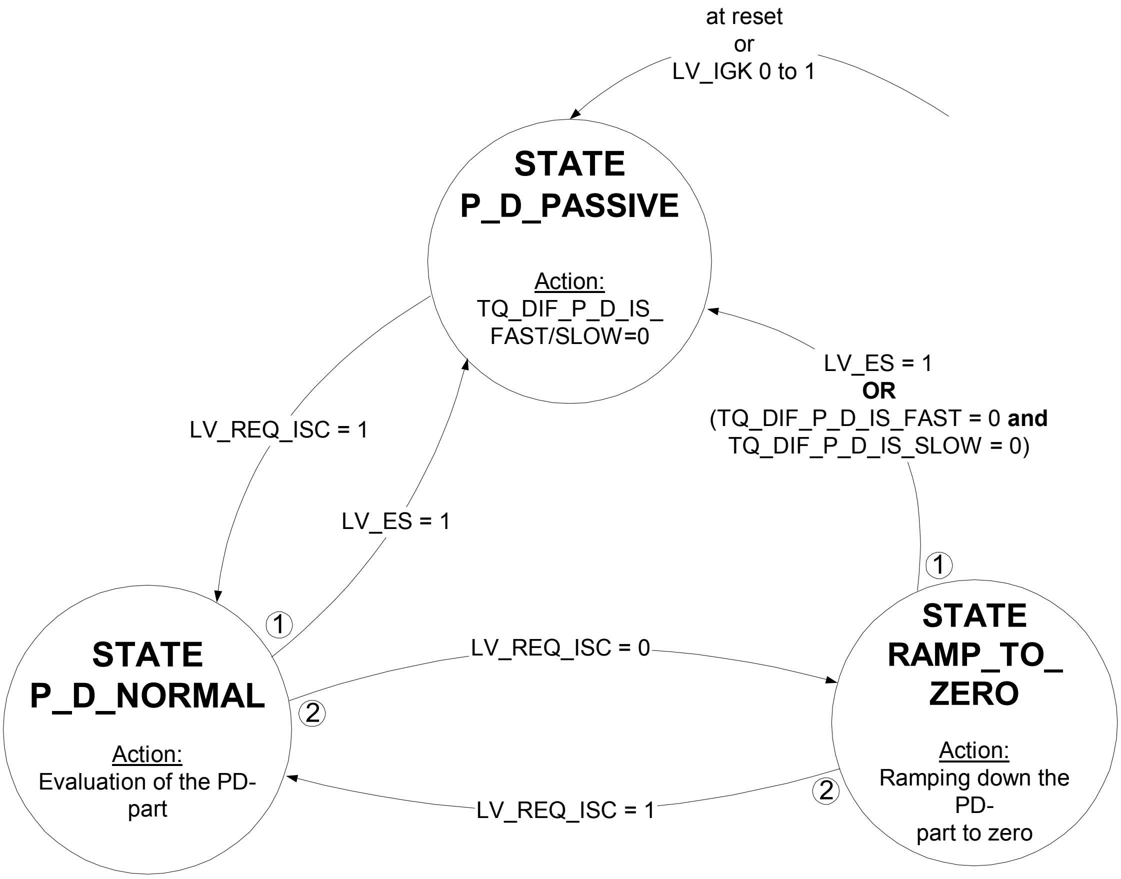


Figure 2: *State of the PD-part*

The PD-part is devided into three states:

STATE_P_D_ISC = P_D_PASSIVE:

Initialization (at reset or from LV_IGK = 0 → 1; from all other states in case of LV_ES = 1)

STATE_P_D_ISC = P_D_NORMAL:

In this state the evaluation of the controller maps takes place.

STATE_P_D_ISC = RAMP_TO_ZERO:

Ramp to passive value, which is always zero for the PD-part; in this state the PD-part is ramped down to zero in case of deactivation of the PD-part

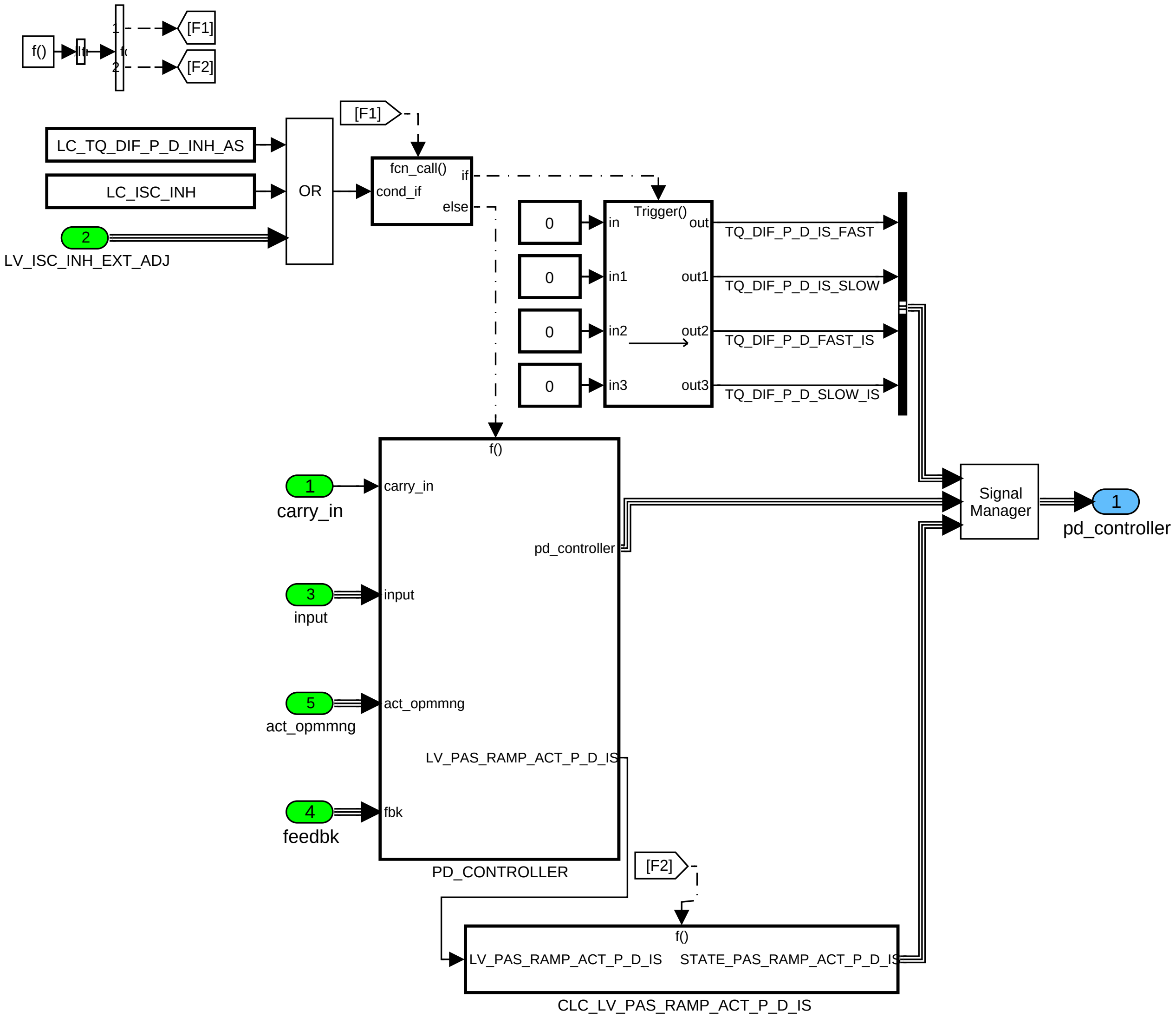


Figure 90 ENSC_M8006/ OPM/ PD_CONTROLLER

PD controller state machine and state calculation overview

For special function test, it´s possible to set TQ_DIF_P_D_IS_FAST / SLOW_OPM_x = 0 Nm by application system or diagnostic tool (LV_ISC_INH_EXT_ADJ).

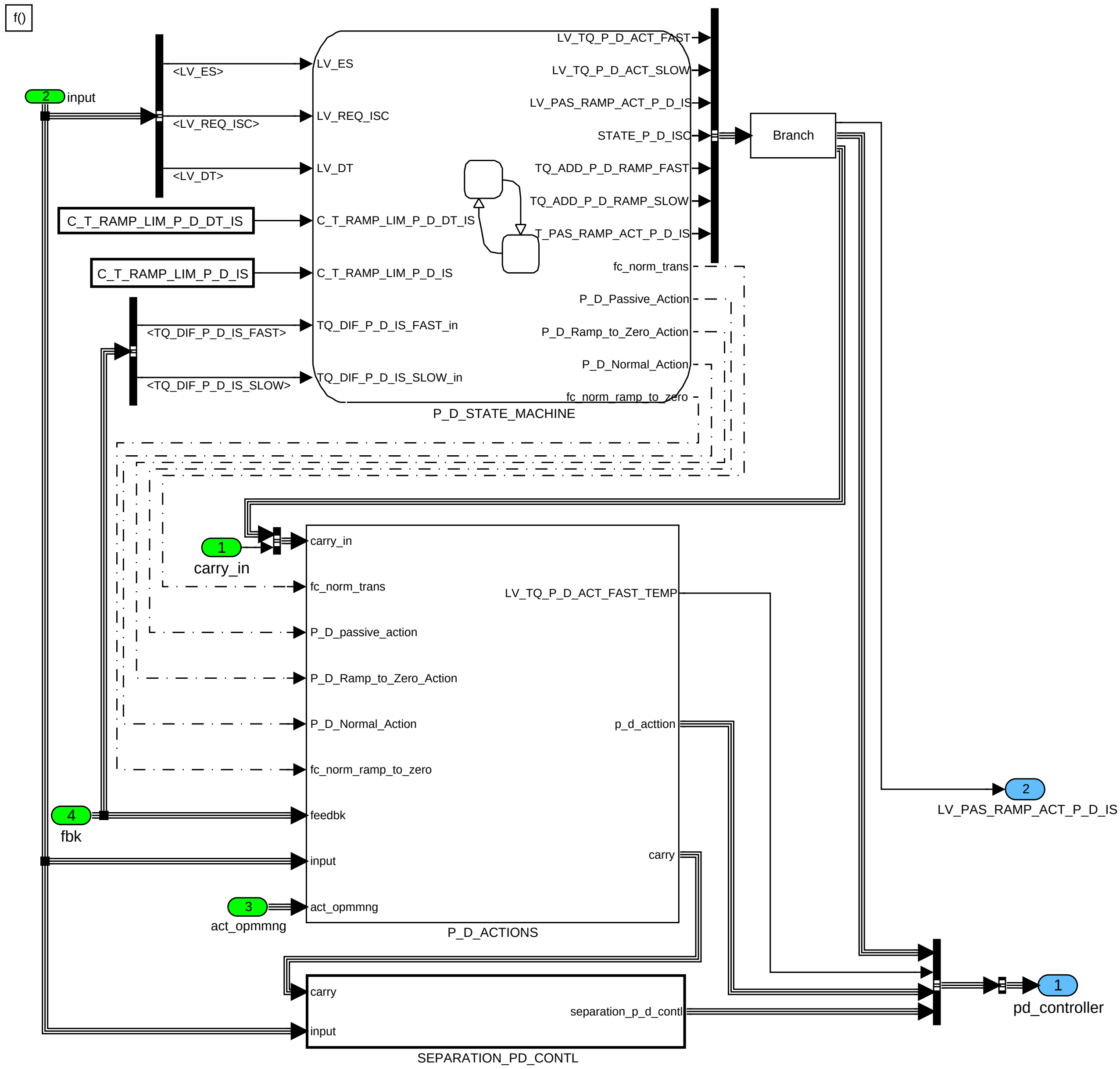


Figure 91 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER

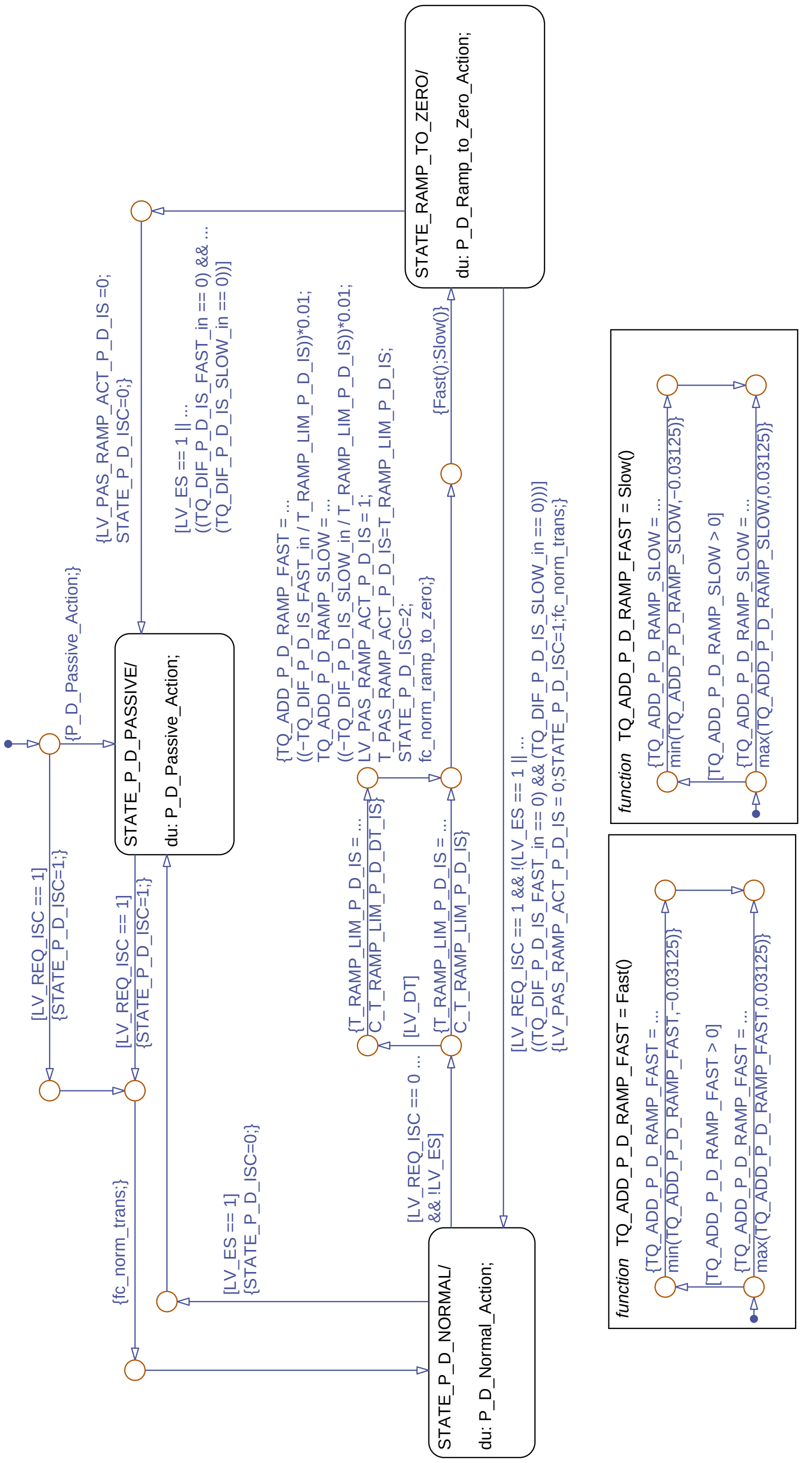


Figure 92 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_STATE_MACHINE

Calculation of N_GRD_P_D_IS and calculations at gear engaged and disengaged.

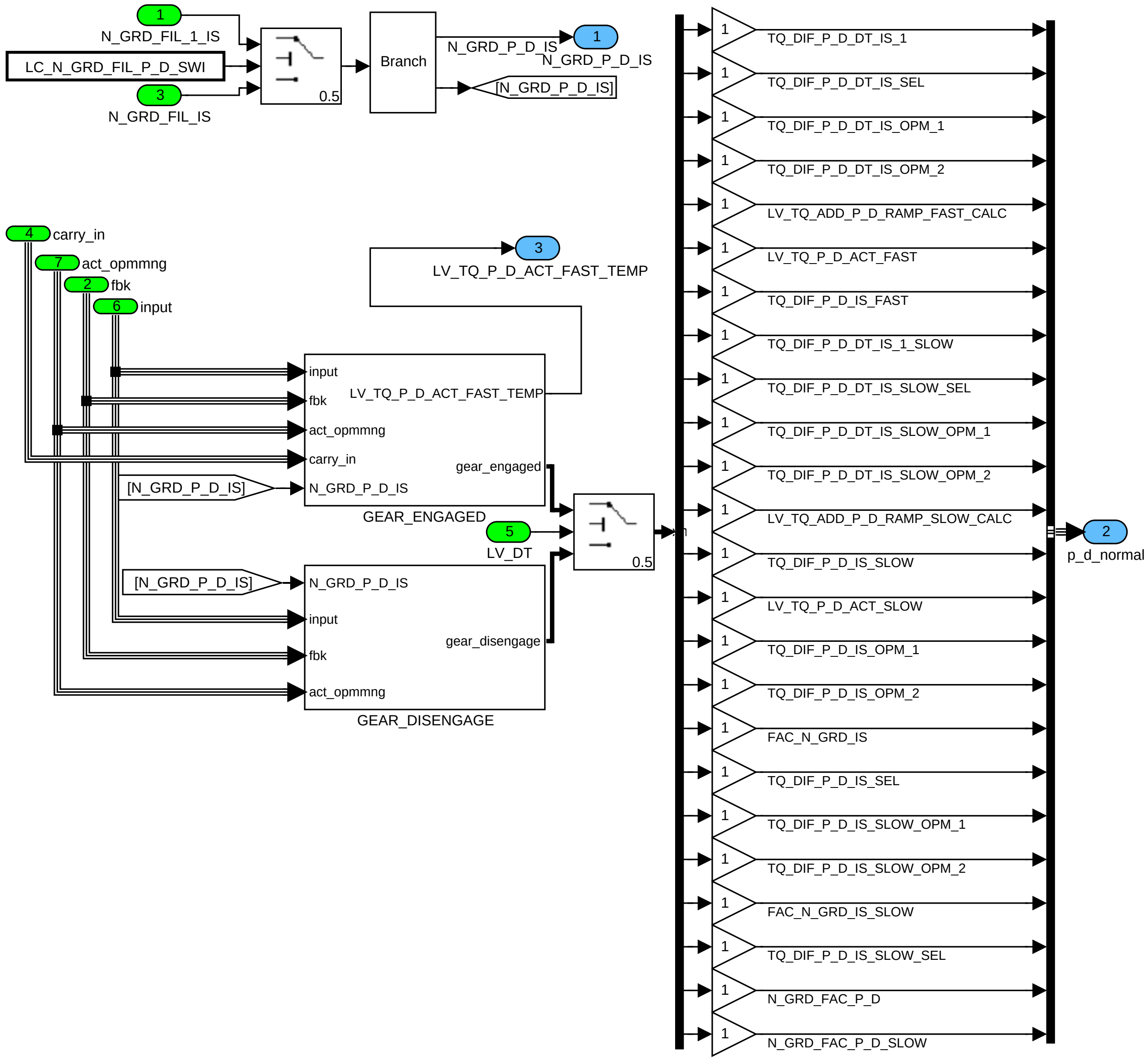


Figure 93 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL

Calculations at gear engaged

Gear engaged is calculated when LV_DT is 1

Calculations at PD part fast

Output from operation manager in PD part fast

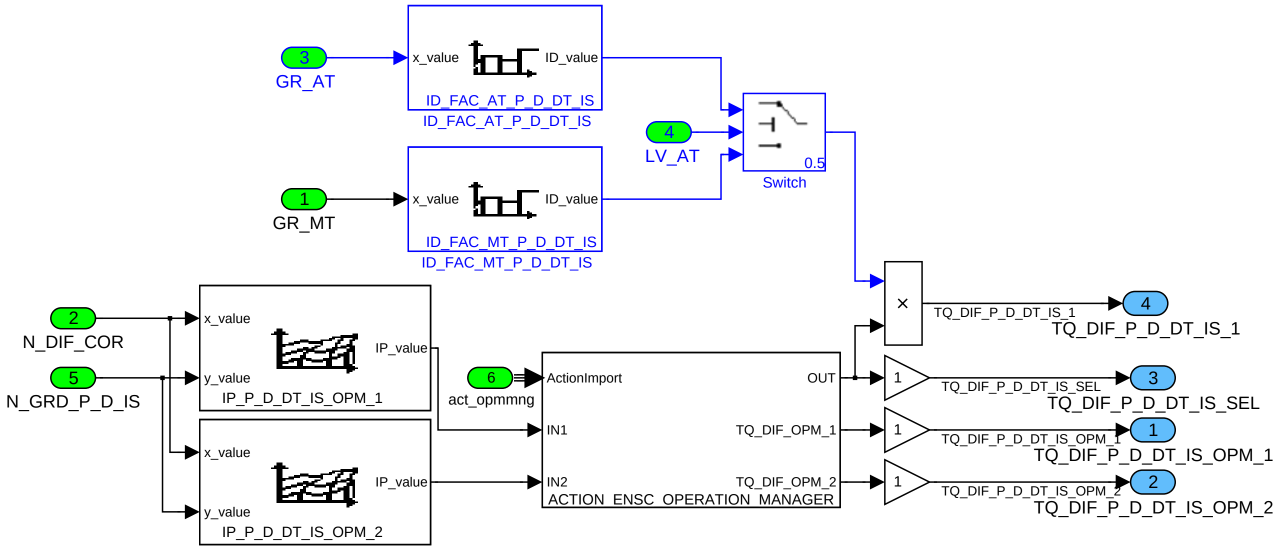


Figure 94 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_FAST/ CLC_P_D_DT_IS

Calculation of TQ_DIF_P_D_IS_FAST and LV_TQ_P_D_ACT_FAST

Calculation of LV_TQ_P_D_ACT_FAST

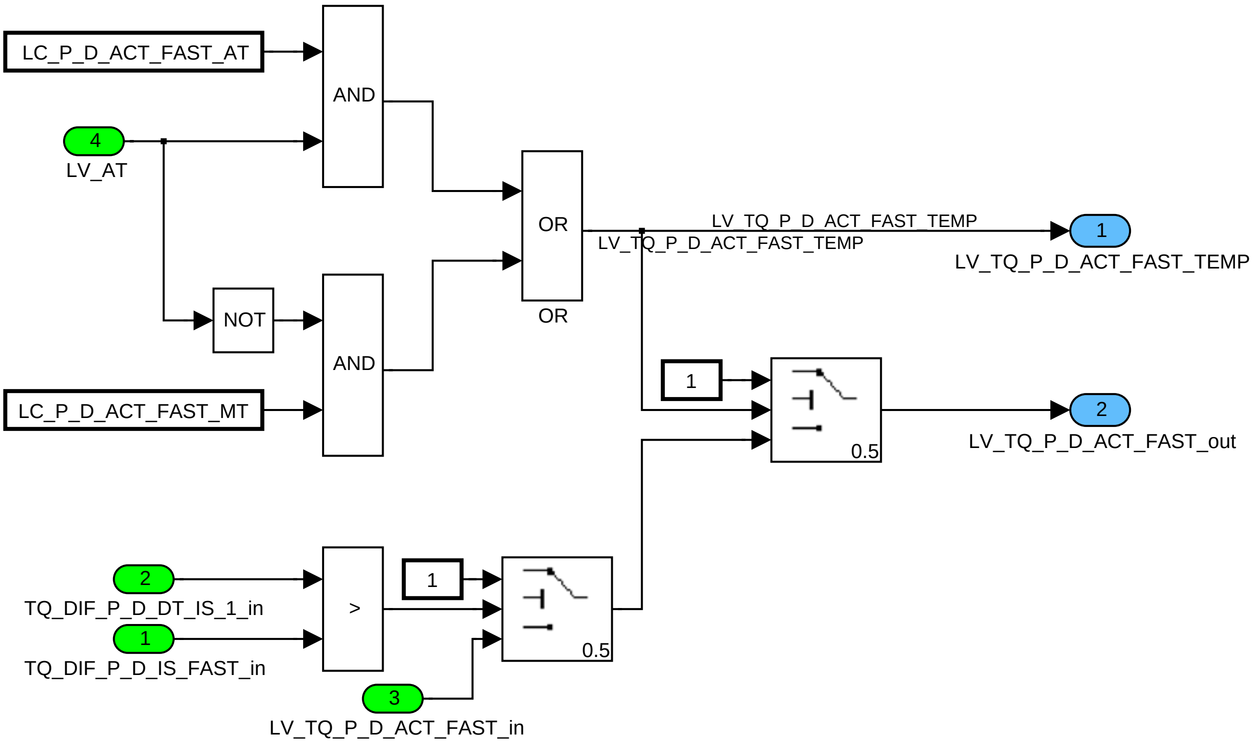


Figure 95 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_FAST/ CLC_P_D_IS/ CLC_LV_TQ_P_D_ACT_FAST

Calculation of LV_TQ_ADD_P_D_RAMP_FAST_CALC and TQ_DIF_P_D_IS_FAST

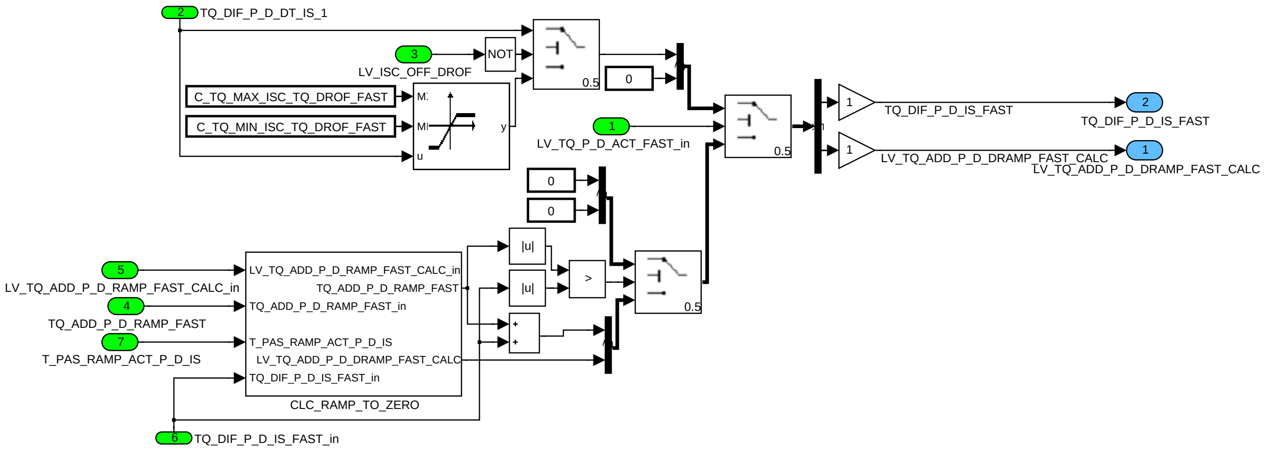


Figure 96 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_FAST/ CLC_P_D_IS/ CLC_P_D_IS

Calculation of LV_TQ_ADD_P_D_RAMP_FAST_CALC and TQ_DIF_P_D_IS_FAST at ramp_to_zero

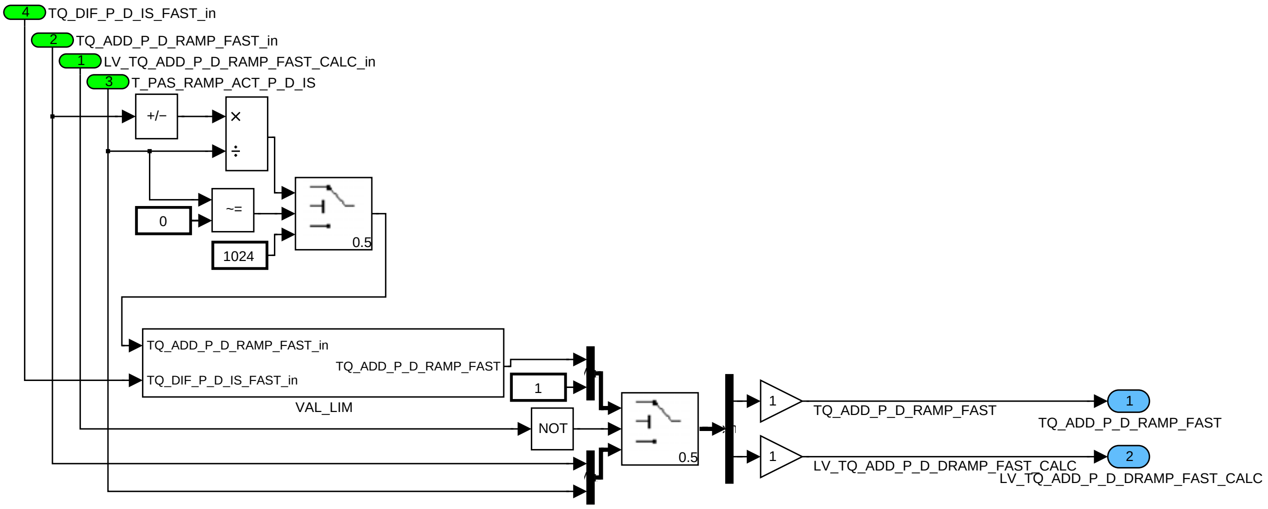


Figure 97 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_FAST/ CLC_P_D_IS/ CLC_P_D_IS/ CLC_RAMP_TO_ZERO

Limiting the value of TQ_DIF_P_D_IS_FAST

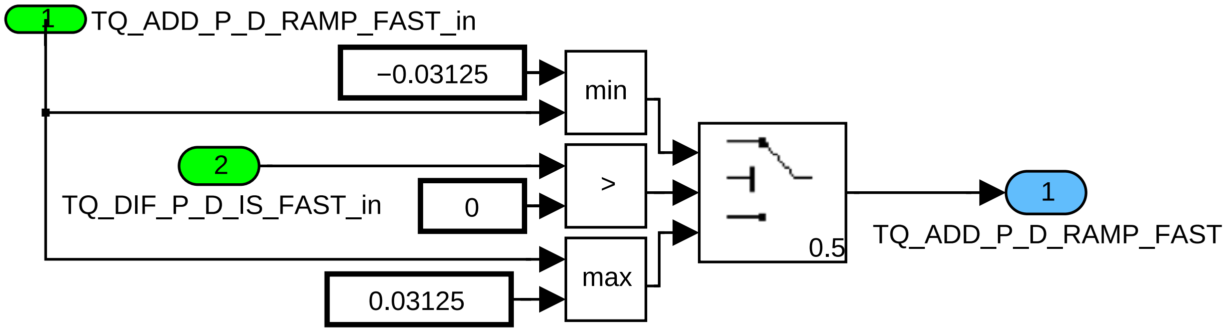


Figure 98 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_FAST/ CLC_P_D_IS/ CLC_P_D_IS/ CLC_RAMP_TO_ZERO/ VAL_LIM

Calculation at PD_slow

Output from operation manager in PD slow

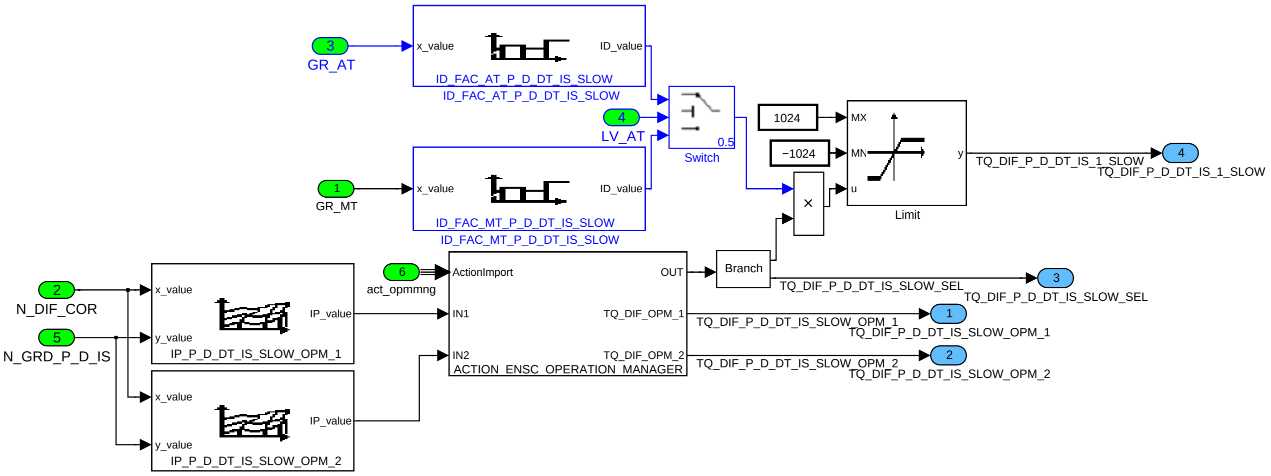


Figure 99 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_SLOW/ CLC_P_D_DT_IS

Calculation of LV_TQ_PD_ACT_SLOW and TQ_DIF_P_D_IS_SLOW

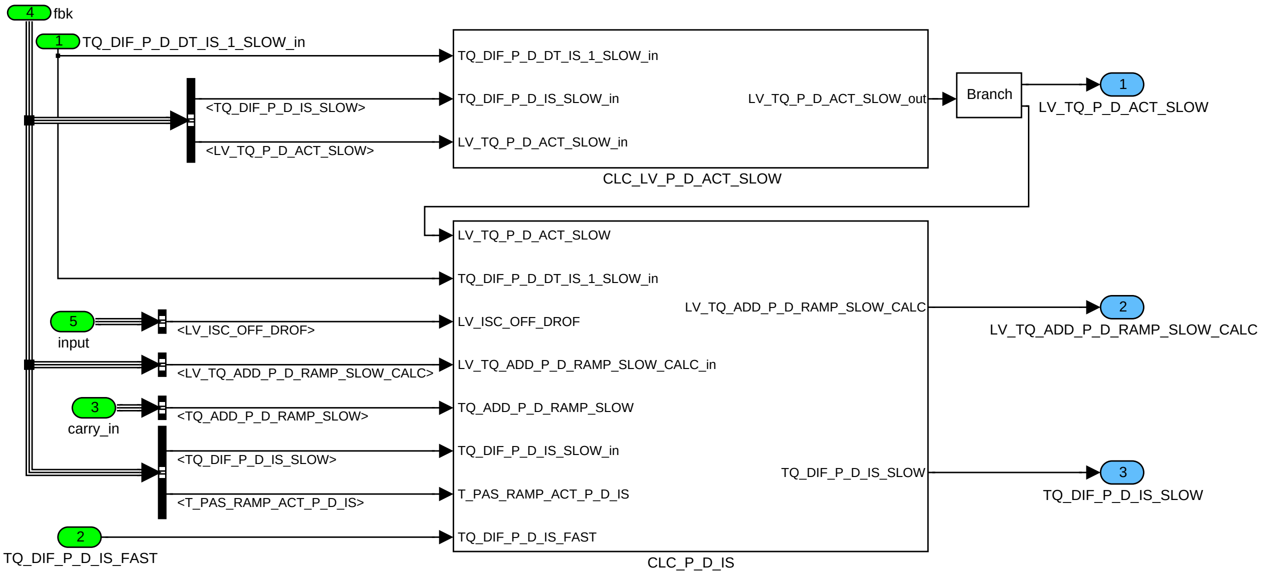


Figure 100 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_SLOW/ CLC_P_D_IS

Calculation of LV_ACT_P_D_IS_SLOW

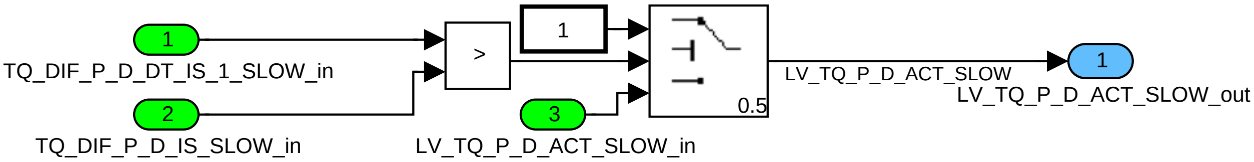


Figure 101 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_SLOW/ CLC_P_D_IS/ CLC_LV_P_D_ACT_SLOW

Calculation of TQ_DIF_P_D_IS_SLOW and LV_TQ_ADD_P_D_RAMP_SLOW_CALC

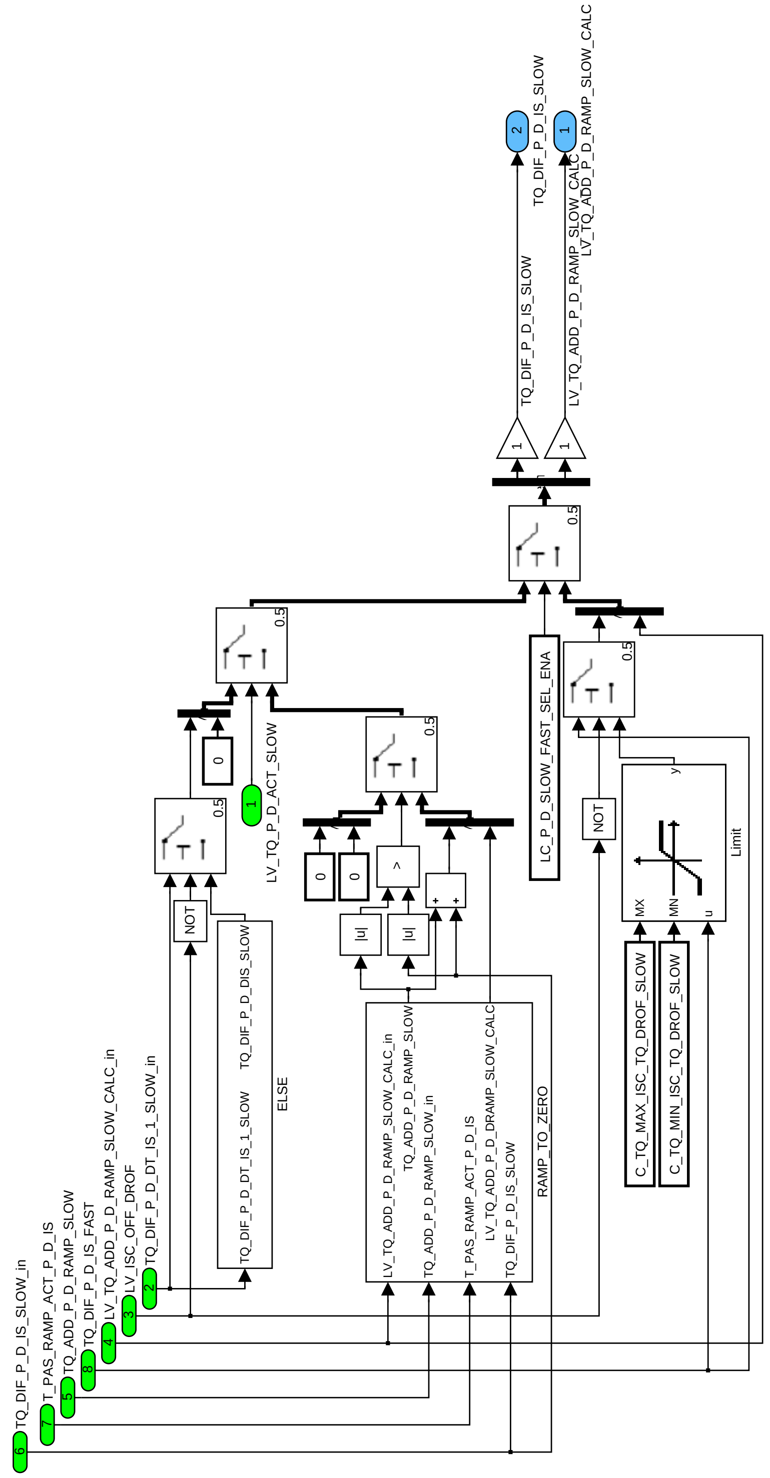


Figure 102 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_SLOW/ CLC_P_D_IS/ CLC_P_D_IS

Limiting the value of TQ_DIF_P_D_IS_SLOW

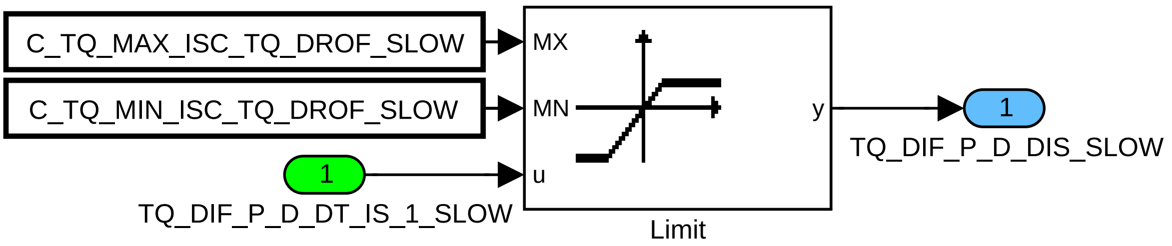


Figure 103 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_SLOW/ CLC_P_D_IS/ CLC_P_D_IS/ ELSE

Calculation at ramp_to_zero

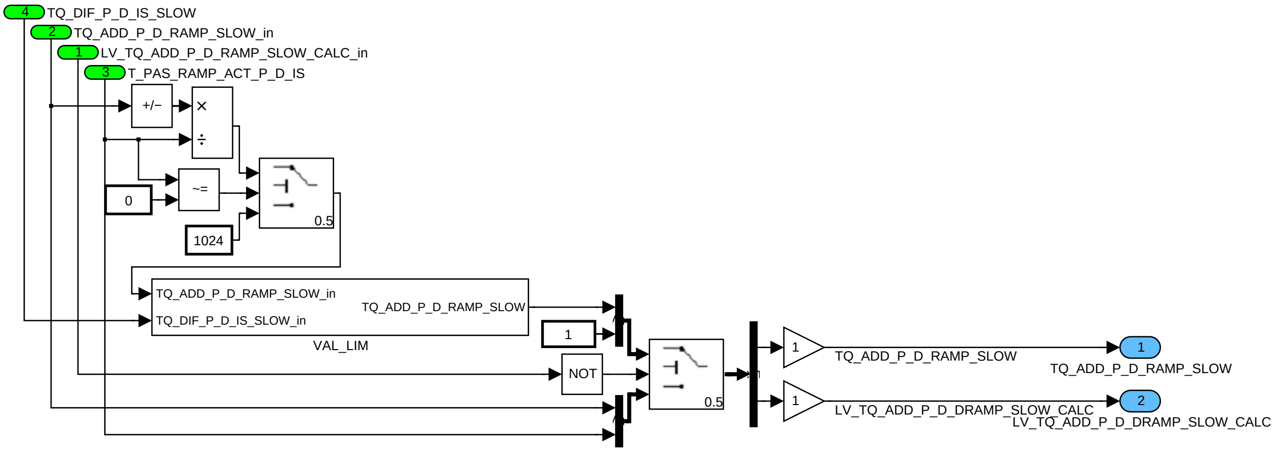


Figure 104 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_SLOW/ CLC_P_D_IS/ CLC_P_D_IS/ RAMP_TO_ZERO

Limiting the value of TQ_DIF_P_IS_SLOW

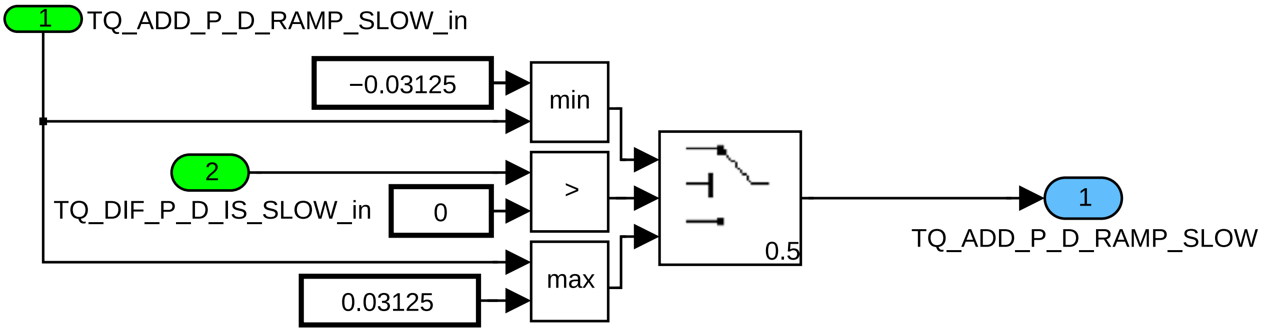


Figure 105 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ PD_PART_SLOW/ CLC_P_D_IS/ CLC_P_D_IS/ RAMP_TO_ZERO/ VAL_LIM

Passing the of old value

NO calculation

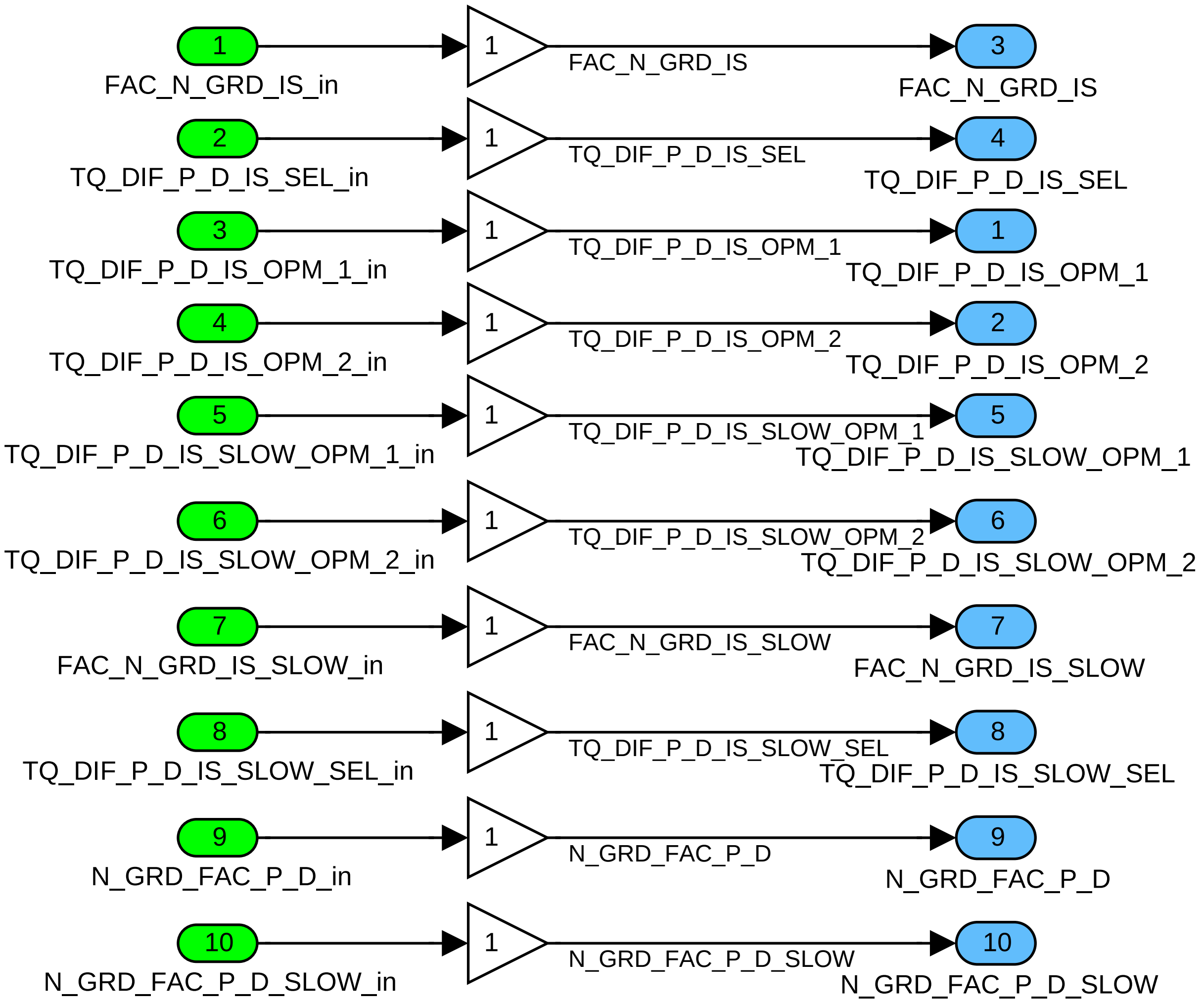


Figure 106 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_ENGAGED/ OLD_VALUES

Calculation at gear disengaged.

Calculations at gear disengaged occurs when LV_DT = 0;

Calculation gear disengaged---> PD part fast

Output from operation manager in PD part fast

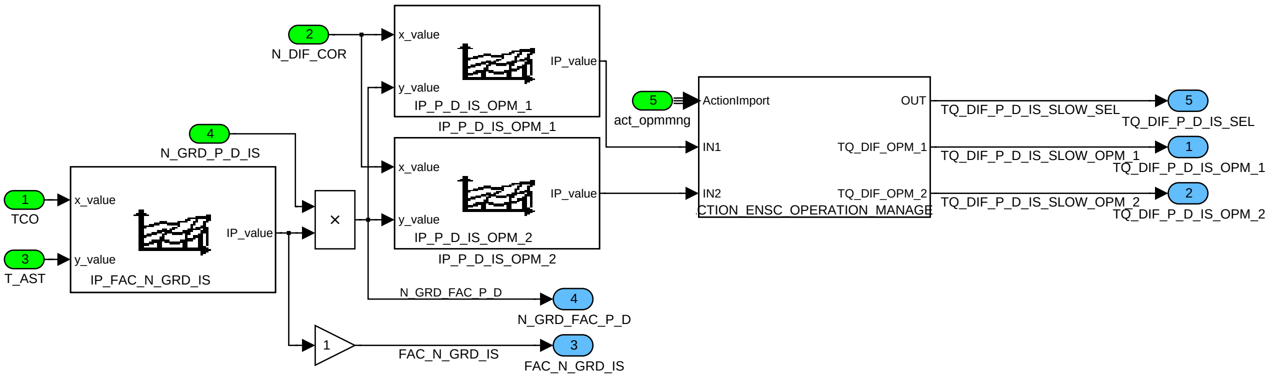


Figure 107 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_DISENGAGE/ PD_PART_FAST/ CLC_P_D_DT_IS

Calculation of LV_TQ_ADD_P_D_RAMP_FAST_CALC and TQ_DIF_P_D_IS_FAST.

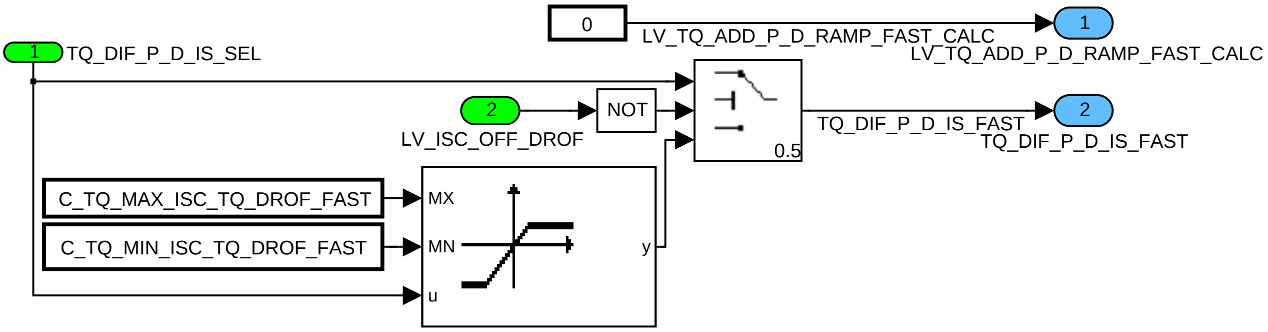


Figure 108 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_DISENGAGE/ PD_PART_FAST/ CLC_P_D_IS

Calculation at PD part slow

Condition check for calculation in PD part slow

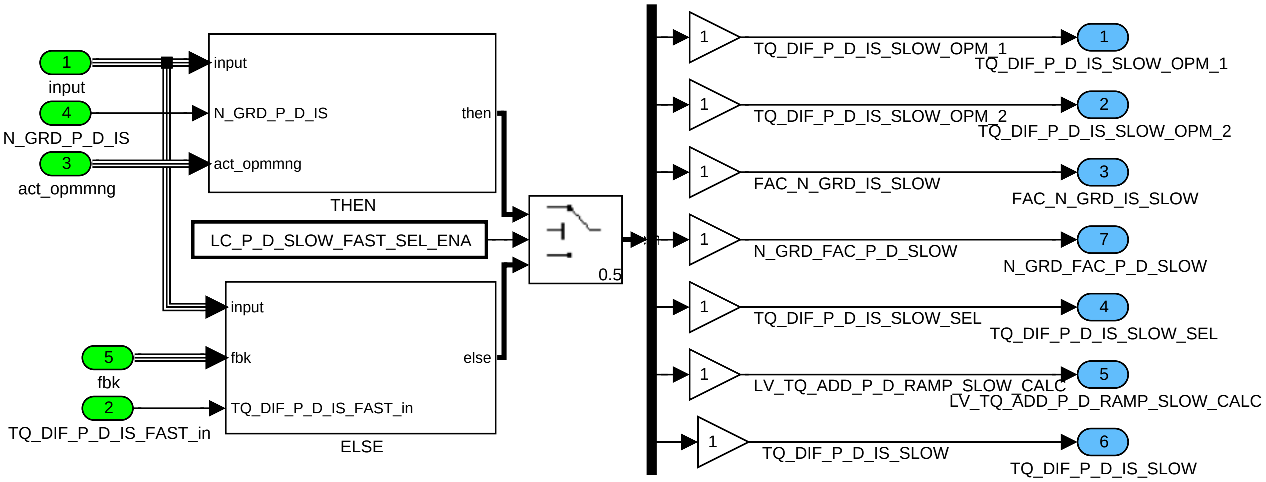


Figure 109 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_DISENGAGE/ PD_PART_SLOW/ CLC_P_D_IS

Calculation at pd part slow when condition is true.

Output from operation manager

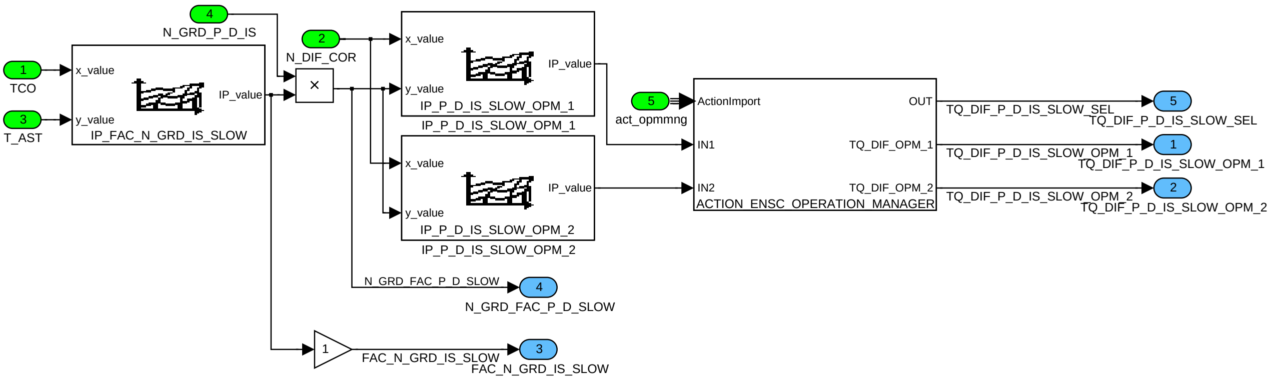


Figure 110 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_DISENGAGE/ PD_PART_SLOW/ CLC_P_D_IS/ THEN/ CLC_P_D_DT_IS

Calculation of TQ_DIF_P_D_IS_SLOW

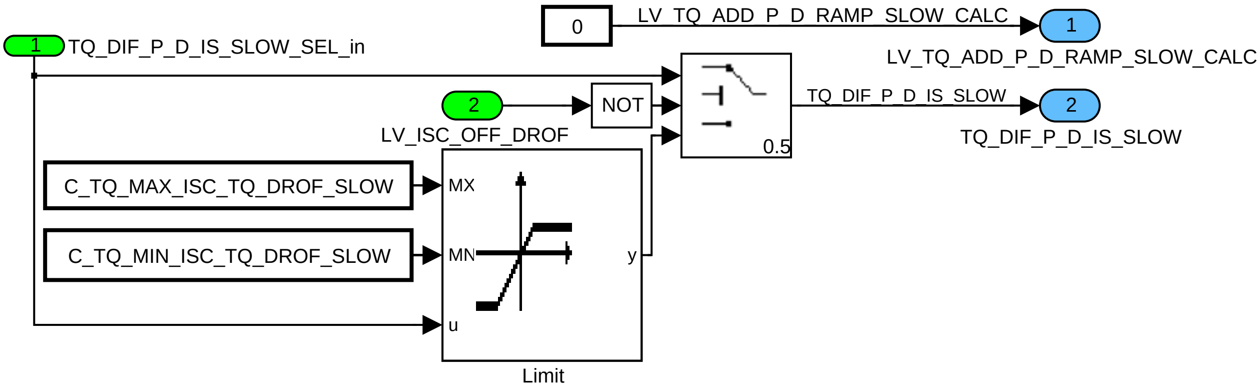


Figure 111 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_DISENGAGE/ PD_PART_SLOW/ CLC_P_D_IS/ THEN/ THEN_TQ_DIF_P_D_IS_SLOW

Calculation at pd part slow when condition is false.

Calculation of TQ_DIF_P_D_IS_SLOW and LV_TQ_ADD_P_D_RAMP_SLOW_CALC

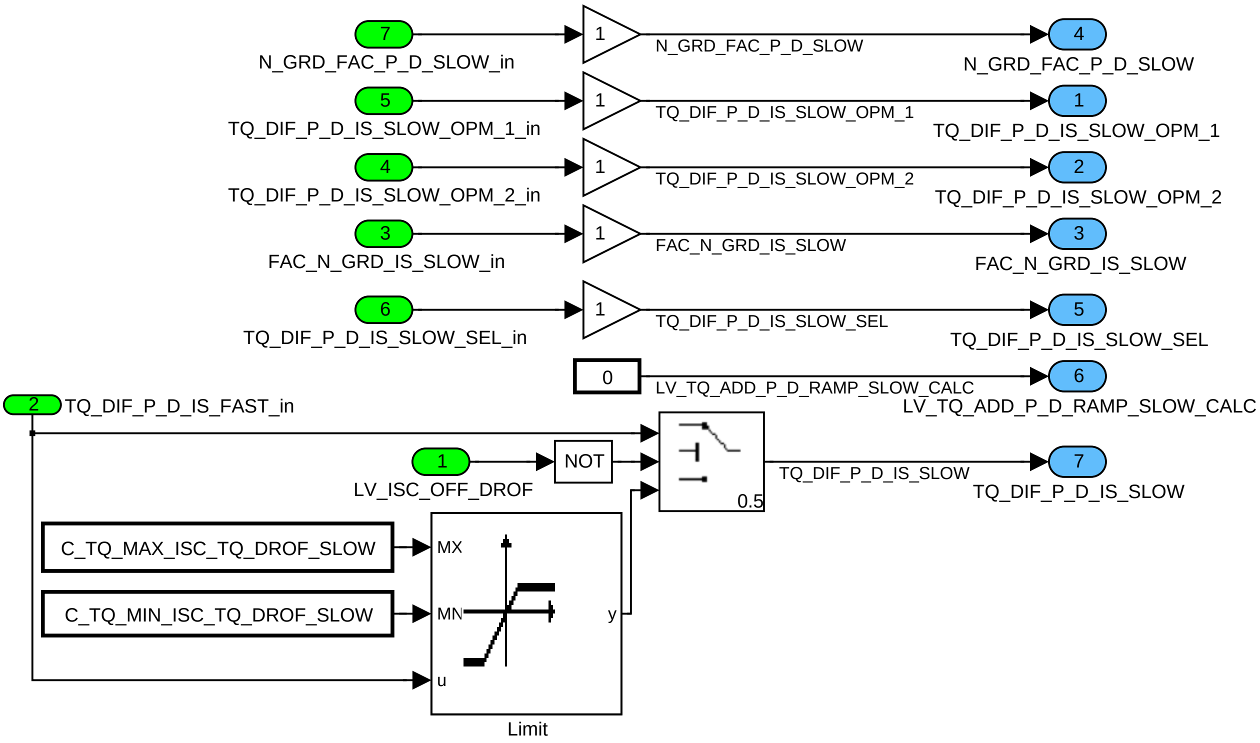


Figure 112 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_DISENGAGE/ PD_PART_SLOW/ CLC_P_D_IS/ ELSE/ CLC

Feed through

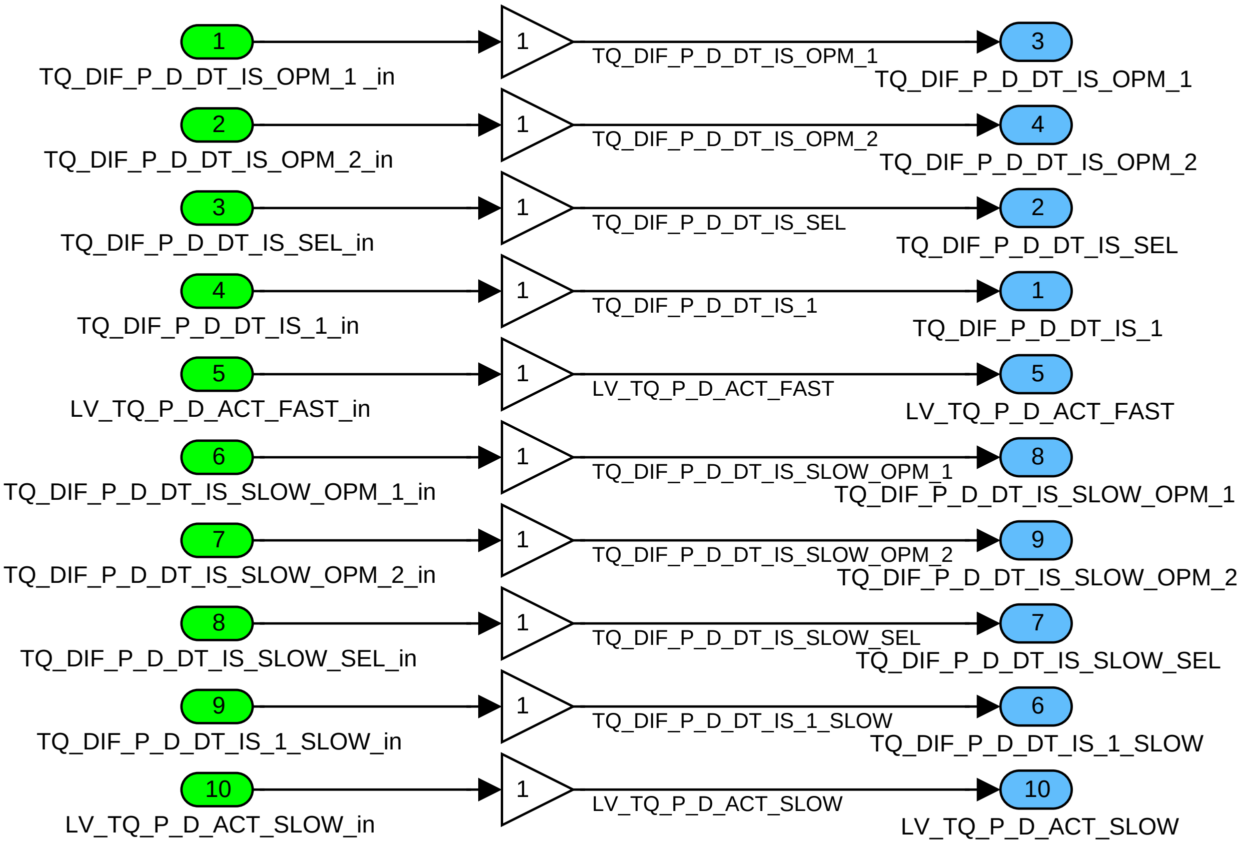


Figure 113 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ P_D_NORMAL/ P_D_NORMAL/ GEAR_DISENGAGE/ OLD_VALUES

Calculation at state ramp_to_zero

To prevent a torque jump if the PD part is on a certain value and the driver hits the pedal (transition to part load), it´s necessary to reduce the PD-part’s value by a deactivation ramp. The **target value** of the ramp operation is always TQ_DIF_P_D_IS_FAST / SLOW = 0 [Nm].

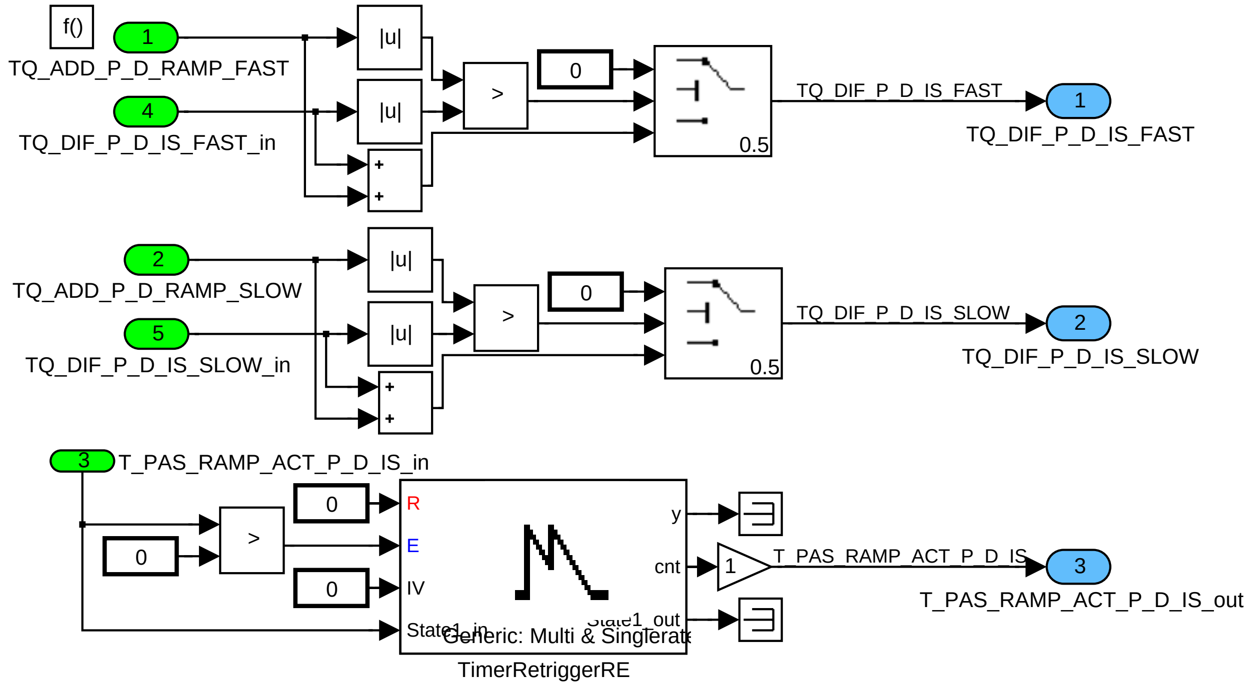


Figure 114 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ P_D_ACTIONS/ RAMP_TO_ZERO

Calculation of TQ_DIF_P_D_SLOW_IS

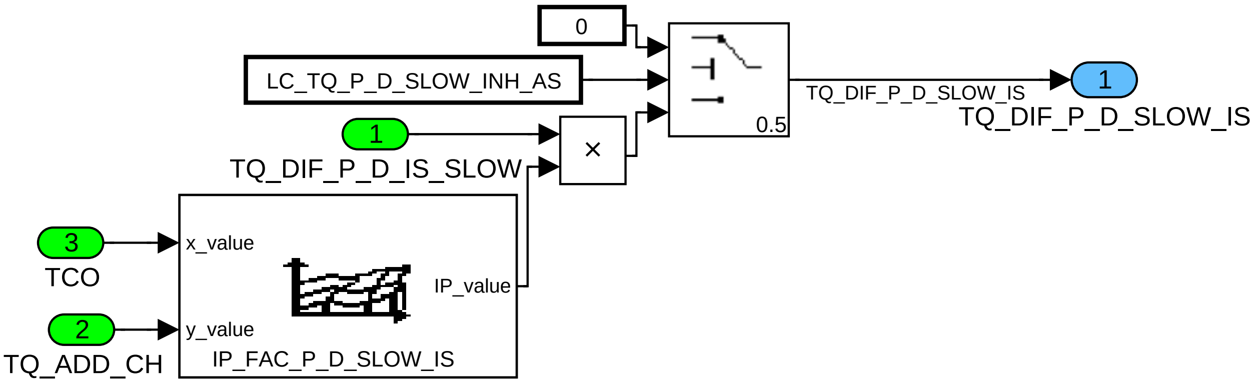


Figure 115 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ SEPARATION_PD_CONTL/ DET_TQ_DIF_P_D_SLOW_IS

Calculation of TQ_DIF_P_D_SLOW_IS

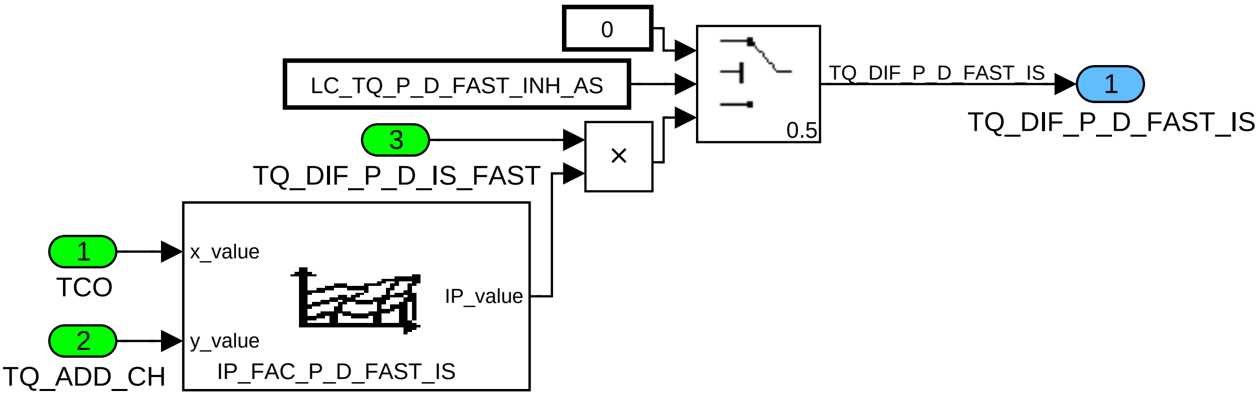
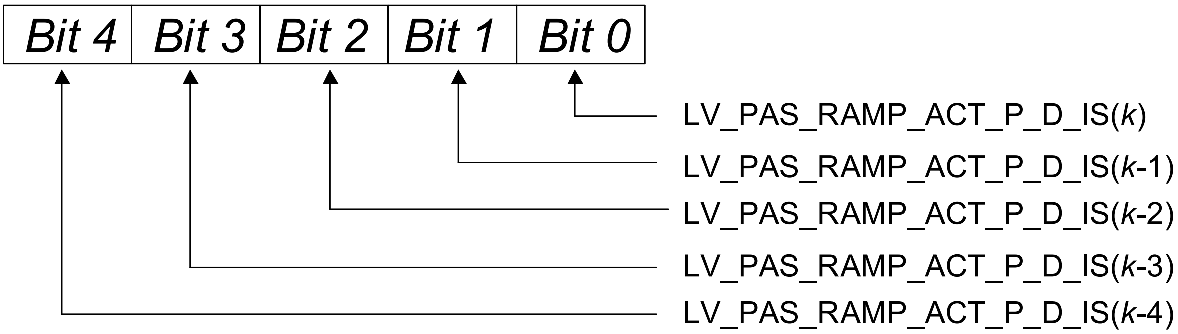


Figure 116 ENSC_M8006/ OPM/ PD_CONTROLLER/ PD_CONTROLLER/ SEPARATION_PD_CONTL/ DET_TQ_DIF_P_D_FAST_IS

Circular buffer for LV_PAS_RAMP_ACT_P_D_IS

In the following circular buffer STATE_PAS_RAMP_ACT_P_D_IS, the last five values of LV_PAS_RAMP_ACT_P_D_IS are stored. This information is needed for the Level 2 which has a recurrency of 40 ms and would not detected a high-low-high flank change inside 40 ms:



This information is needed for the monitoring of the idle speed controller. It is updated every 10ms; the values are shifted from Bit j to Bit j+1, j=0,1,2,3.

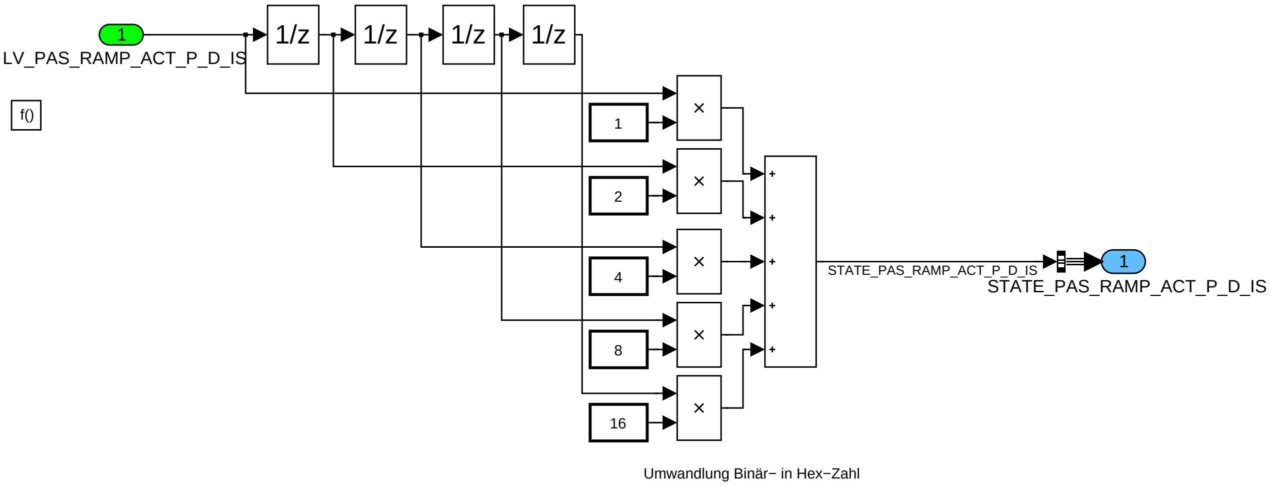


Figure 117 ENSC_M8006/ OPM/ PD_CONTROLLER/ CLC_LV_PAS_RAMP_ACT_P_D_IS

22.5 Application Incidences for Idle Speed Control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_REQ_ISC	V/O	0...1H	0...1	1	[-]
Logical variable for idle speed controller activatIon request					
N_GRD_FIL_ISC_ACT_AST	V	80...7FH	-4096...4064	32	[rpm/s]
Filtered gradient for ISC after start activation					
LV_NEG_N_GRD_FIL_MEM	V	0...1H	0...1	1	[-]
Memory bit for falling gradient after start					
STATE_REQ_ISC	V	0H 1H 2H 3H 4H	ISC_PASSIVE AFTER_START NORMAL TRAILING_THR PART_LOAD	1	[-]
Actual state for ISC request					
T_CTR_PU_ISC_ACT	V/O	0...FFFFH	0...655.35	0.01	[s]
Time counter for ISC activation in PU					

Input data:

LV_ST	LV_IS	N	N_GRD
LV_PU	N_SP_IS	VS	LV_DT
LV_AMT_CRAWL_ON	LV_PL	LV_N_SP_IS_CS	N_GB
N_GRD	N_DIF_COR	LV_AT	N_SP_IS_RATIO

Application conditions:

Activation:

```
IF   LC_ISC_ACT_CDN_SWI == 1
 THEN
          Use these application incidences
 ELSE IF LV_AMT_CRAWL_ON = 1
            THEN        LV_REQ_ISC  =  0
            ELSE        LV_REQ_ISC  =  LV_IS
          ENDIF
ENDIF
```

At reset: STATE_REQ_ISC = ISC_PASSIVE
T_CTR_PU_ISC_ACT = C_T_DLY_ISC_ACT_PU

Update rate: 10 ms

At deactivation: T_CTR_PU_ISC_ACT = C_T_DLY_ISC_ACT_PU

22.5.1 General

The activation of the idle speed controller depends on different driving situations. Five states are distinguished. The actual status of the idle speed activation is stored in STATE_REQ_ISC.

State 0: Idle speed controller is passive **(STATE_REQ_ISC = ISC_PASSIVE)**.

State 1: Activation of the ISC after start **(STATE_REQ_ISC = AFTER_START)**.

State 2: "Normal activation": Activiation of the ISC in case of LV_IS = 1
(STATE_REQ_ISC = NORMAL)

State 3: Activation of the ISC if speed remains stationary at a certain value in trailing throttle state **(STATE_REQ_ISC = TRAILING_THR)**.

State 4: Activation of the ISC in part load, in case of underspeed.
(STATE_REQ_ISC = PART_LOAD).

Formula section:

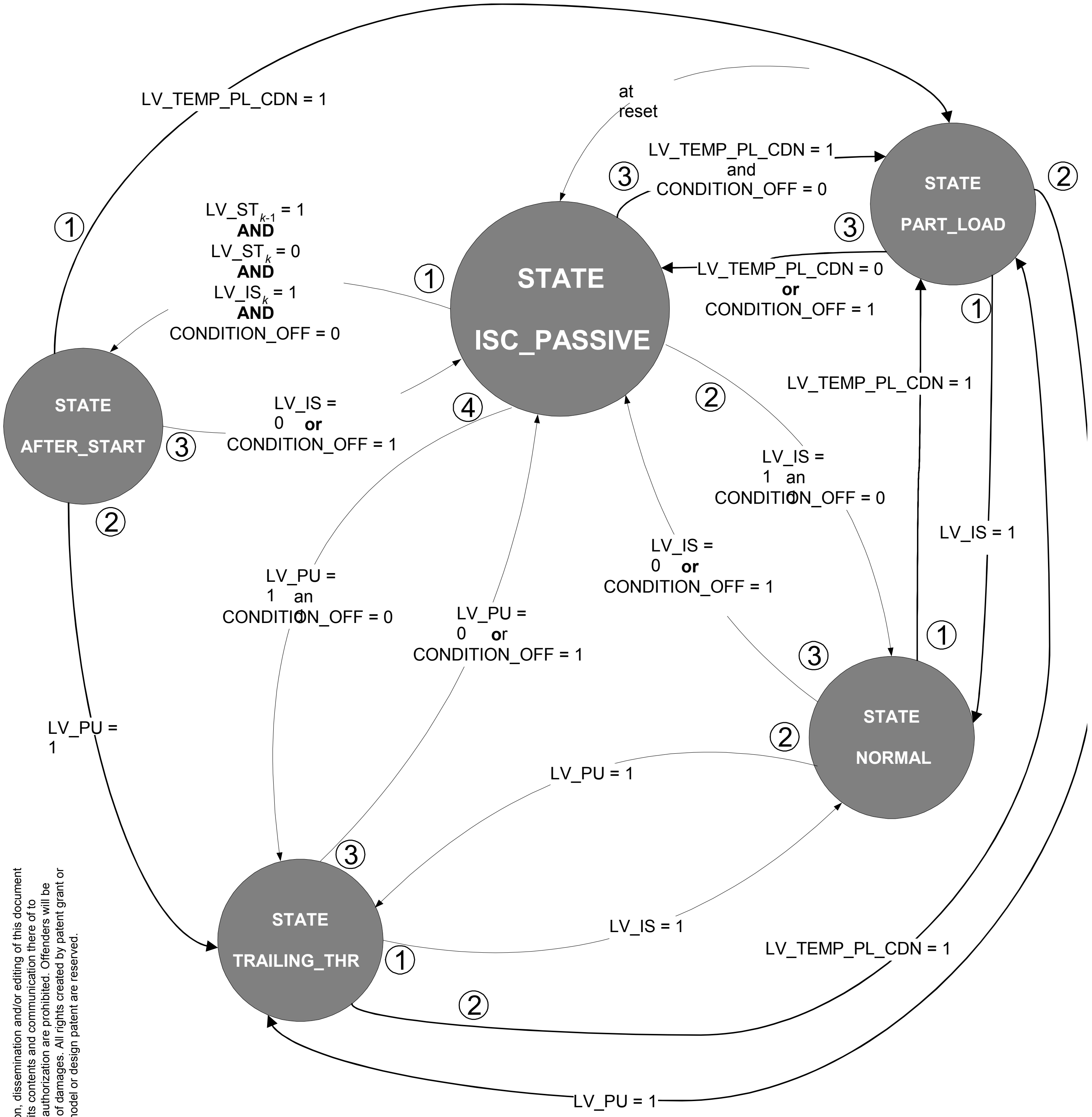
Action:

FlipFlop condition (LV_FF):

```
IF  N > (N_SP_IS + C_N_DIF_ISC_ACT_PL_UP)  THEN  LV_FF = 0
ELSE  IF  N < (N_SP_IS – C_N_DIF_ISC_ACT_PL_DOWN)  THEN  LV_FF = 1
```

```
IF      { LV_PL == 1 }  AND

        { LV_FF == 1 }  AND

        N_SP_IS < C_N_SP_IS_MAX_PL_ACT  AND

        { LC_ISC_PL_ACT_INH == 0 }

        OR

        { LV_N_SP_IS_CS  AND  { NOT LV_IS }   AND  { NOT LV_ST } }
THEN
        LV_TEMP_PL_CDN = 1

ELSE
        LV_TEMP_PL_CDN = 0
```

The following state diagram illustrates the connection between these five cases, figure 1.



Formula section:

Figure 1: *State diagram of the ISC activation.*
The numbers at different transitions of one state to another represent the priority
in which the transitions have to be checked.

22.5.2 ISC Passive (STATE ISC_PASSIVE)

FUNCTION DESCRIPTION:

This state is the idle speed controller passive state. The controller is deactivated if there is a transition to this state.

Formula section:

Action: LV_REQ_ISC = 0

Transition: see figure 1

Action during transition to AFTER_START: LV_NEG_N_GRD_FIL_MEM = 0
N_GRD_FIL_ISC_ACT_AST = 0

22.5.3 Checking ISC activation conditions after start (Entry into IS after ST, STATE AFTER_START)

FUNCTION DESCRIPTION:

After transition from engine state ST to IS, the activation of the ISC is requested if speed reaches a calibratible offset from N_SP_IS with a negative gradient N_GRD, figure 2. If speed does not reach this offset after decreasing and increasing again, the controller is also activated.

Transition: see figure 1

Action:

```
IF    LC_ISC_ACT_AST_CDN_SWI == 1
THEN
    LV_REQ_ISC = 1
ELSE
{
    % Calculation of a filtered gradient for after start activation

    $N\_GRD\_FIL\_ISC\_ACT\_AST_k = N\_GRD\_FIL\_ISC\_ACT\_AST_{k-1} +$
    $+ C\_CRLC\_N\_GRD\_FIL\_ISC\_ACT\_AST \cdot \{ N\_GRD_k - N\_GRD\_FIL\_ISC\_ACT\_AST_{k-1} \}$


    % If N reaches the calibratible offset N_SP_IS + C_N_DIF_ISC_ACT_AST
    % with a negative gradient, then the controller is activated. If
    % speed increases again before reaching N_SP_IS + C_N_DIF_ISC_ACT_AST,
    % then the controller is activated, too. The falling gradient
    % is layed up in LV_NEG_N_GRD_FIL_MEM.

    IF      N_GRD_FIL_ISC_ACT_AST < 0
    THEN
            LV_NEG_N_GRD_FIL_MEM = 1

            IF      { N  <  N_SP_IS + C_N_DIF_ISC_ACT_AST }
            THEN
                    LV_REQ_ISC =1
            ENDIF

    ELSEIF  { N_GRD_FIL_ISC_ACT_AST >= 0 }  AND
            { LV_NEG_N_GRD_FIL_MEM == 1}

    THEN
            LV_REQ_ISC =1

    ENDIF
}
```

After setting LV_NEG_N_GRD_FIL_MEM, its value will not be changed until the after start is left to STATE 0.

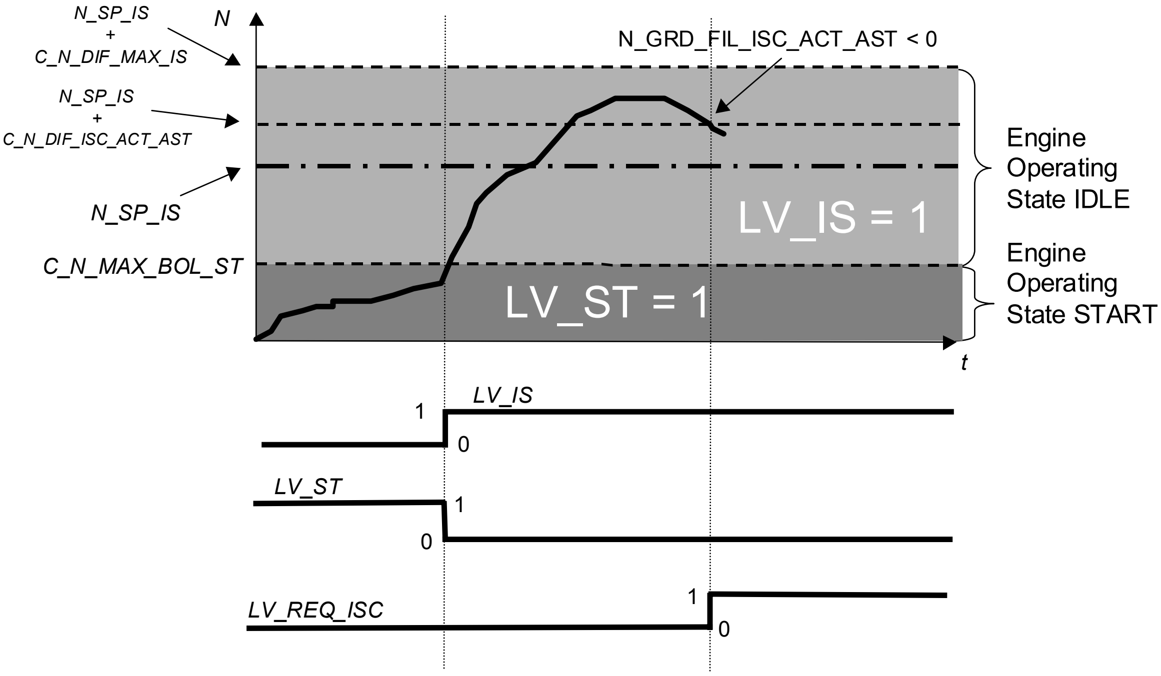


Figure 2: *ISC activation after start*

22.5.4 Checking ISC activation conditions for "normal activation", STATE NORMAL)

The idle speed controller is activated on principle in case of LV_IS = 1 (except STATE 1 is active).

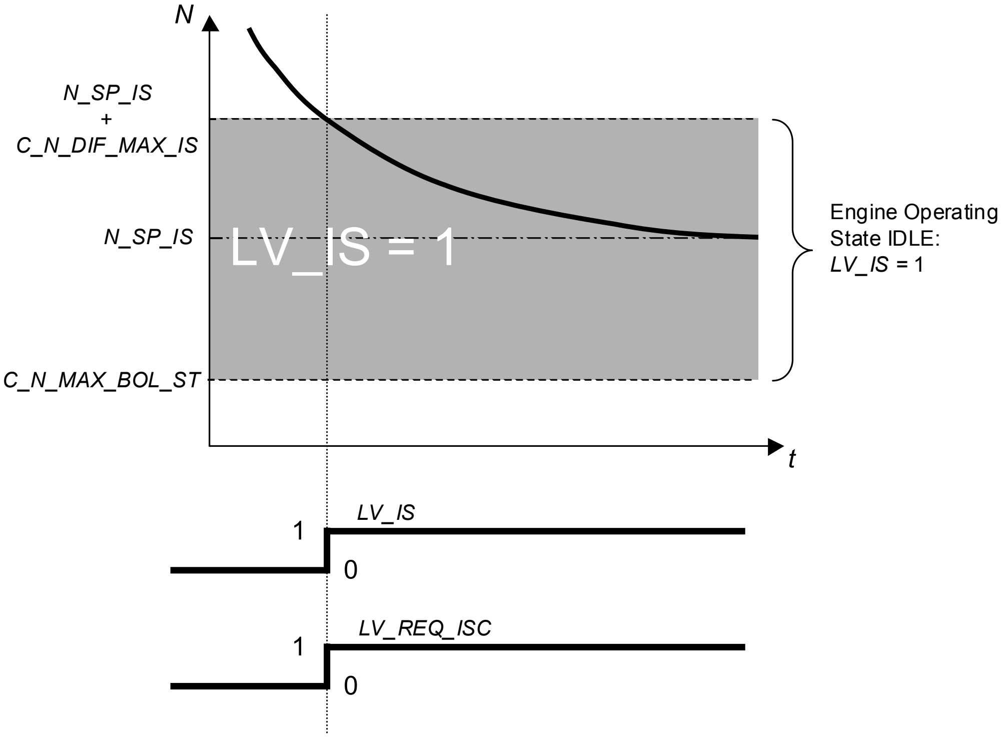


Figure 3: *Activation of the ISC in case of LV_IS = 1*

Formular section:

Transition: see figure 1

Action:

```
IF(LV_AT = 1)
THEN IF (LV_DT = 0)
        THEN LV_REQ_ISC = 1
        ELSE IF (N_GB < N - C_N_GB_DIF_N OR N_GRD < IP_N_GRD_N_SP_IS_RATIO OR N_DIF_COR >0 )
            THEN LV_REQ_ISC = 1
```

ELSE LV_REQ_ISC = 1;

22.5.5 ISC activation in case of low engine speed decrease in trailing throttle state (LV_PU = 1, STATE TRAILING_THR)

If speed decrease in trailing throttle state is very low or speed does not decrease, the idle speed controller is activated after a calibratible time *C_T_DLY_ISC_ACT_PU*. The condition for that is a standing vehicle (VS=0). The counter is started when entering PU ($LV\_PU_{k-1} = 0 \rightarrow LV\_PU_k = 1$). The counter is reinitialized with C_T_DLY_ISC_ACT_PU when leaving PU. See figure 4.

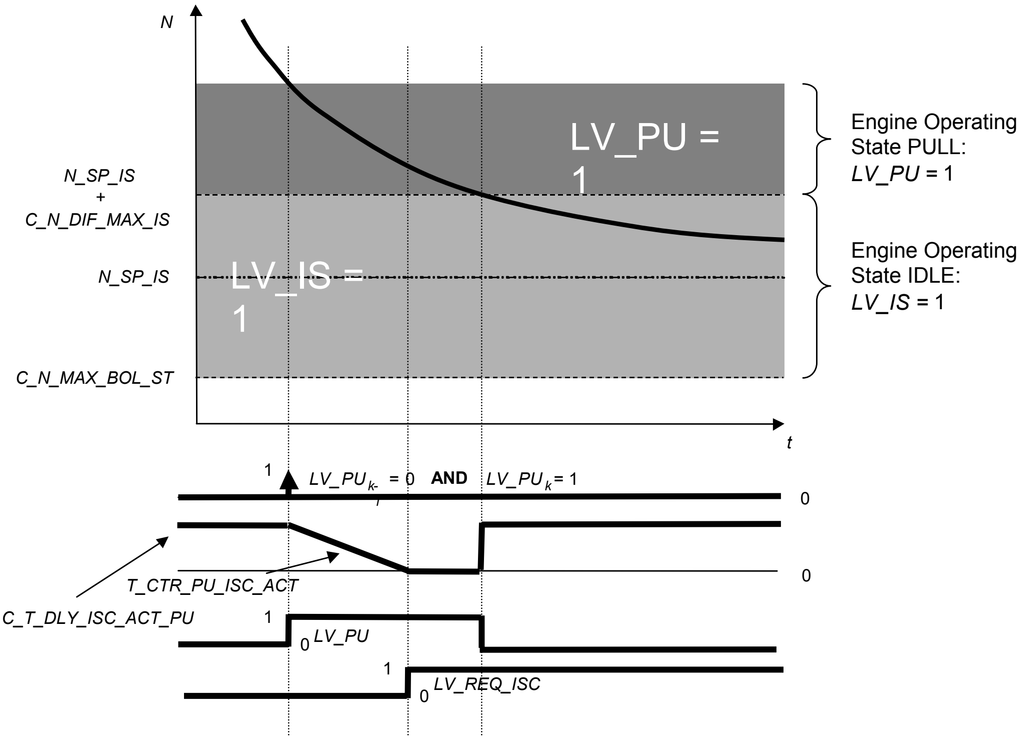


Figure 4: *Activation of the ISC in PU in case the speed decrease is too low and* ***the vehicle does not move (VS = 0)****. The counter starts from a tunable value C_T_DLY_ISC_ACT_PU and stops at zero.*

Formular section:

Transition: see figure 1

Action: start to ramp down *T_CTR_PU_ISC_ACT* when entering PU (the initialization of *T_CTR_PU_ISC_ACT* was already done before)

```
IF      { VS == 0  OR  LV_DT == 0 }   AND   { T_CTR_PU_ISC_ACT == 0 }
THEN
        LV_REQ_ISC = 1
ELSE
        LV_REQ_ISC = 0
ENDIF
```

Reinitialize *T_CTR_PU_ISC_ACT* with C_T_DLY_ISC_ACT_PU when leaving state TRAILING_THR

22.5.6 Conditions for ISC part load activation (STATE PART_LOAD)

In part load, the driver or cruise control torque request can be so small, that the idle speed setpoint N_SP_IS cannot be held and speed is decreasing. In this case, the idle speed controller is used to guaranty N ≥ N_SP_IS. The conditions for activating the ISC in this case depend on the speed and a calibratible hysteresis, figure 5.

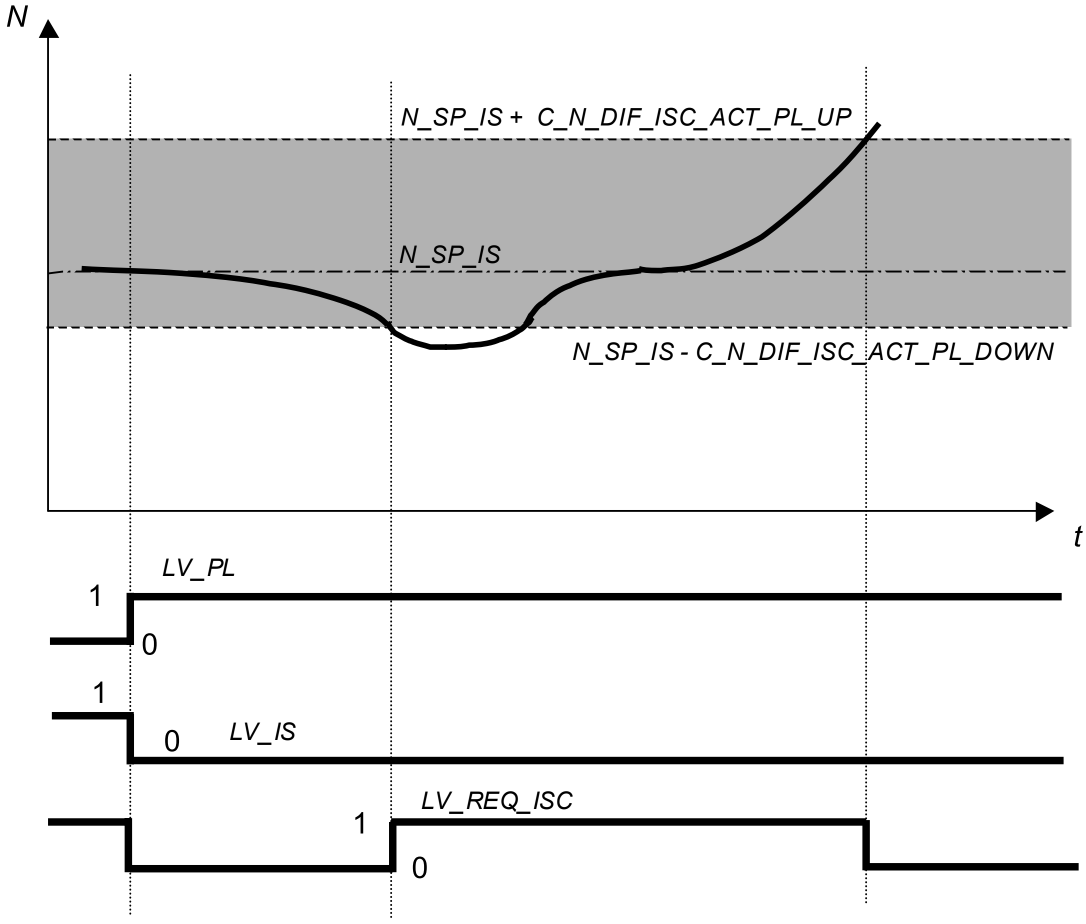


Figure 5: *Activating the ISC in part load*

Formula section:

Transition: see figure 1

Action: LV_REQ_ISC = 1

22.5.7 Checking ISC inhibit interfaces

During AMT shift intervention Phase 1 (LV_AMT_CRAWL_ON = 1) an additional Converter torque is calculated; during this time the ISC is looked State "PASSIVE".

All inhibit interfaces lead to CONDITION_OFF which is used in the state diagram (figure 1).

```
IF    LV_AMT_CRAWL_ON = 1
THEN        CONDITION_OFF  =  1
  ELSE      CONDITION_OFF  =  0
ENDIF
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
LC_ISC_ACT_CDN_SWI	1	0...1H	0...1	1	[-]
Calibratible logical bit for switching between different activation conditions					
LC_ISC_ACT_AST_CDN_SWI	1	0...1H	0...1	1	[-]
Switch for using different after start ISC activation conditions					
LC_ISC_PL_ACT_INH	1	0...1H	0...1	1	[-]
Calibratible bit to inhibit ISC activation in part load					
C_N_DIF_ISC_ACT_AST	1	8000...7FFFH	-32768...32767	1	[rpm]
Additive term to define the low limit for ISC activation after start					
C_CRLC_N_GRD_FIL_ISC_ACT_AST	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Filter correlation factor for speed gradient filtering in connection with after start activation of ISC					
C_T_DLY_ISC_ACT_PU	1	0...FFFFH	0...655.35	0.01	[s]
Time limit for ISC activation in PU					
C_N_DIF_ISC_ACT_PL_DOWN	1	8000...7FFFH	-32768...32767	1	[rpm]
Additive term to define the low limit for ISC activation in PL					
C_N_DIF_ISC_ACT_PL_UP	1	8000...7FFFH	-32768...32767	1	[rpm]
Additive term to define the upper limit for ISC deactivation in PL					
C_N_SP_IS_MAX_PL_ACT	1	0...1FE0H	0...8160	1	[rpm]
Speed setpoint threshold for part load activation					
C_N_GB_DIF_N	1	0...FFFEH	0...8191.75	0.125	[rpm]
Difference in input shaft speed turbine/gear box and engine speed					
IP_N_GRD_N_SP_IS_RATIO	8	0...FFH	-4096...4064	32	[rpm/s]
LDP_N_SP_IS_RATIO	8	0...FFFFH	0...7.99987	0.1221e-3	[-]
IP for Engine Speed Gradient					

22.6 Idle Speed Torque Reserve

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_ADD_IS_BOL	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Filtered calculated torque reserve bottom limit at idle					
TQ_DIF_ADD_IS_TOL	O/V	0...7FFFH	0...1.02397E+3	0.03125	Nm
Idle speed torque reserve limitiation for stabilizing the ISC at high torque reserves					
N_DIF_OFS_PRED_OPM_1	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted engine speed difference for torque reserve calculation (mode 1)					
N_DIF_PRED_CS_OPM_1	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted idle speed control variable at drive-off-support (mode 1)					
N_DIF_OFS_PRED_OPM_2	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted engine speed difference for torque reserve calculation (mode 2)					
FAC_TQ_ADD_IS_BOL_OPM_2	V	0...FFH	0...7.96875	0.03125	-
Resulting factor for TQ_ADD_IS_BOL_BAS (mode 2)					
FAC_TQ_ADD_IS_BOL_OPM_1	V	0...FFH	0...7.96875	0.03125	-
Resulting factor for TQ_ADD_IS_BOL_BAS (mode 1)					
TQ_ADD_IS_BOL_SEL	V	0...7FFFH	0...1.02397E+3	0.03125004	Nm
Calculated torque reserve botton limit at idle					
N_DIF_OFS_PRED_CS_OPM_1	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted engine speed difference for torque reserve calculation at drive-off-support (mode 1)					
T_N_DIF_OFS_PRED_OPM_2	V	0...FFH	0...7.96875	0.03125	s
Prediction time for engine speed difference (mode 2)					
N_DIF_OFS_PRED_CS_OPM_2	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted engine speed difference for torque reserve calculation at drive-off-support (mode 2)					
N_DIF_PRED_OPM_2	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted idle speed control variable (mode 2)					
TQ_ADD_IS_BOL_OPM_2	V	0...7FFFH	0...1.02397E+3	0.03125004	Nm
Torque reserve at idle (mode 2)					
N_DIF_PRED_OPM_1	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted idle speed control variable (mode 1)					
N_DIF_PRED_CS_OPM_2	V	8000...7FFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
Predicted idle speed control variable at drive-off-support (mode 2)					
TQ_ADD_IS_BOL_OPM_1	V	0...7FFFH	0...1.02397E+3	0.03125004	Nm
Torque reserve at idle (mode 1)					
T_N_DIF_OFS_PRED_CS_OPM_1	V	0...FFH	0...7.96875	0.03125	s
Prediction time for engine speed difference at drive-off-support (mode 1)					
T_N_DIF_OFS_PRED_OPM_1	V	0...FFH	0...7.96875	0.03125	s
Prediction time for engine speed difference (mode 1)					
T_N_DIF_OFS_PRED_CS_OPM_2	V	0...FFH	0...7.96875	0.03125	s
Prediction time for engine speed difference at drive-off-support (mode 2)					

Input data:

LV_IS	LV_ES	VS_FIL	TQI_ADD_MAX_TOL

FAC_TQ_ADD_IS_OPM_SEL	LV_AT	N_GRD	TQI_REQ_TRA
MAF	LV_DRI	VLFT_MIN	N
OPM_AV	TCO	LV_N_SP_IS_CS	MAP_DIP_SP_MMV
OPM_REQ	N_SP_IS	ID_IDX_OPM_DROF	N_DIF
TIA	TOIL	LV_VAR_TCT	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TQ_ADD_IS_BOL_CRLC	1	0...FFH	0...0.99609307	0.00390625	-
Filter constant for torque reserve					
C_TQ_DIF_ADD_IS_TOL_OPM_1	1	0...7FFFH	0...1.02397E+3	0.03125	Nm
Idle speed torque reserve limitiation for stabilizing the ISC at high torque reserves in OPM_1					
C_TQ_DIF_ADD_IS_TOL_OPM_2	1	0...7FFFH	0...1.02397E+3	0.03125	Nm
Idle speed torque reserve limitiation for stabilizing the ISC at high torque reserves in OPM_2					
ID_TQ_ADD_IS_OPM_AV	8	1...8H	1...8	1	-
LDPM_OPM_AV	8	0...8H	0...8	1	[-]
Operation mode matix (actual)					
ID_TQ_ADD_IS_OPM_REQ	8	1...8H	1...8	1	-
LDPM_OPM_REQ	8	0...8H	0...8	1	[-]
Operation mode matix (requested)					
IP_FAC_VS_TQ_ADD_IS_BOL_OPM_1	4	0...FFH	0...1.9921875	0.0078125	-
LDPM_VS_FIL_IP_FAC_VS_TQ_ADD_IS	4	0...FFFFH	0...511.992188	0.0078125	km/h
Factor for IP_FAC_TQ_ADD_IS_BOL_CS_OPM_1 (mode 1)					
IP_FAC_VS_TQ_ADD_IS_BOL_OPM_2	4	0...FFH	0...1.9921875	0.0078125	-
LDPM_VS_FIL_IP_FAC_VS_TQ_ADD_IS	4	0...FFFFH	0...511.992188	0.0078125	km/h
Factor for IP_FAC_TQ_ADD_IS_BOL_CS_OPM2 (mode 2)					
IP_T_N_DIF_OFS_PRED_CS_OPM_1	8	0...FFH	0...7.96875	0.03125	s
LDPM_N_T_N_DIF_OFS_PRED	8	0...1FE0H	0...8.16E+3	1	[rpm]
Time constant for calculation of predicted engine speed deviation at drive-off-support (mode1)					
IP_T_N_DIF_OFS_PRED_CS_OPM_2	8	0...FFH	0...7.96875	0.03125	s
LDPM_N_T_N_DIF_OFS_PRED	8	0...1FE0H	0...8.16E+3	1	[rpm]
Time constant for calculation of predicted engine speed deviation at drive-off-support (mode2)					
IP_T_N_DIF_OFS_PRED_OPM_1	8	0...FFH	0...7.96875	0.03125	s
LDPM_N_T_N_DIF_OFS_PRED	8	0...1FE0H	0...8.16E+3	1	[rpm]
Time constant for calculation of predicted engine speed deviation (mode 1)					
IP_T_N_DIF_OFS_PRED_OPM_2	8	0...FFH	0...7.96875	0.03125	s
LDPM_N_T_N_DIF_OFS_PRED	8	0...1FE0H	0...8.16E+3	1	[rpm]
Time constant for calculation of predicted engine speed deviation (mode 2)					
IP_FAC_TEMP_TQ_ADD_IS_BAS_OPM_1	6x6	0...FFH	0...1.9921875	0.0078125	-
LDPM_TIA_IP_FAC_TEMP_TQ_ADD_IS	6	0...FEH	-48...142.5	0.75	[°C]
LDPM_TOIL_IP_FAC_TEMP_TQ_ADD_IS	6	0...C8H	-40...160	1	[°C]
Temperature correction for IP_TQ_ADD_IS_BOL_BAS (mode1)					
IP_FAC_TEMP_TQ_ADD_IS_BAS_OPM_2	6x6	0...FFH	0...1.9921875	0.0078125	-
LDPM_TIA_IP_FAC_TEMP_TQ_ADD_IS	6	0...FEH	-48...142.5	0.75	[°C]
LDPM_TOIL_IP_FAC_TEMP_TQ_ADD_IS	6	0...C8H	-40...160	1	[°C]
Temperature correction for IP_TQ_ADD_IS_BOL_BAS (mode2)					
IP_FAC_TQ_ADD_IS_BOL_CS_OPM_1	8x6	0...FFH	0...7.96875	0.03125	-
LDP_N_DIF_PRED_CS_OPM_1	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_MAF_FAC_TQ_ADD_IS	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Factor for basis torque reserve at idle due to N_DIF_PRED and load for drive-off-support (mode1)					
IP_FAC_TQ_ADD_IS_BOL_CS_OPM_2	8x6	0...FFH	0...7.96875	0.03125	-
LDP_N_DIF_PRED_CS_OPM_2	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_MAF_FAC_TQ_ADD_IS	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Factor for basis torque reserve at idle due to N_DIF_PRED and load for drive-off-support (mode2)					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_TQ_ADD_IS_BOL_DRI_OPM_1	8x6	0...FFH	0...7.96875	0.03125	-
LDPM_N_DIF_PRED_OPM_1	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_MAF_FAC_TQ_ADD_IS	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Factor for basis torque reserve at idle due to N_DIF_PRED and load with drive (mode 1)					
IP_FAC_TQ_ADD_IS_BOL_DRI_OPM_2	8x6	0...FFH	0...7.96875	0.03125	-
LDPM_N_DIF_PRED_OPM_2	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_MAF_FAC_TQ_ADD_IS	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Factor for basis torque reserve at idle due to N_DIF_PRED and load with drive (mode 2)					
IP_FAC_TQ_ADD_IS_BOL_OPM_1	8x6	0...FFH	0...7.96875	0.03125	-
LDPM_N_DIF_PRED_OPM_1	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_MAF_FAC_TQ_ADD_IS	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Factor for basis torque reserve at idle due to N_DIF_PRED and load without drive (mode1)					
IP_FAC_TQ_ADD_IS_BOL_OPM_2	8x6	0...FFH	0...7.96875	0.03125	-
LDPM_N_DIF_PRED_OPM_2	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_MAF_FAC_TQ_ADD_IS	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Factor for basis torque reserve at idle due to N_DIF_PRED and load without drive (mode2)					
IP_LGRD_TQ_ADD_IS_BOL	6x6	0...FFH	0...7.96875	0.03125	Nm
LDPM_N_LGRD_TQ_ADD_IS_BOL	6	0...1FE0H	0...8.16E+3	1	[rpm]
LDPM_MAF_LGRD_TQ_ADD_IS_BOL	6	0...FFFFH	0...1.389E+3	0.0211948	[mg/stk]
Gradient for decrementation of the torque reserve at idle					
IP_LGRD_TQ_ADD_IS_BOL_OPM_2	6x6	0...FFH	0...7.96875	0.03125	Nm
LDPM_N_LGRD_TQ_ADD_IS_BOL	6	0...1FE0H	0...8.16E+3	1	[rpm]
LDPM_MAF_LGRD_TQ_ADD_IS_BOL	6	0...FFFFH	0...1.389E+3	0.0211948	[mg/stk]
Gradient for decrementation of the torque reserve at idle for OPM 2					
IP_TQ_ADD_IS_BOL_BAS_OPM_1	6x6	0...7FFFH	0...1.02397E+3	0.03125	Nm
LDPM_N_SP_IS_TQ_ADD_IS_BOL_BAS	6	0...1FE0H	0...8.16E+3	1	rpm
LDPM_TCO_TQ_ADD_IS_BOL_BAS	6	0...FEH	-48...142.5	0.75	°C
Basis torque reserve at idle (mode1)					
IP_TQ_ADD_IS_BOL_BAS_OPM_2	6x6	0...7FFFH	0...1.02397E+3	0.03125	Nm
LDPM_N_SP_IS_TQ_ADD_IS_BOL_BAS	6	0...1FE0H	0...8.16E+3	1	rpm
LDPM_TCO_TQ_ADD_IS_BOL_BAS	6	0...FEH	-48...142.5	0.75	°C
Basis torque reserve at idle (mode2)					
IP_TQ_ADD_IS_COR_OPM_1	6x6	0...7FFFH	0...1.02397E+3	0.03125	Nm
LDP_VLFT_MIN_IP_TQ_ADD_IS_COR	6	0...FFFFH	0...65.535	0.001	[mm]
LDP_MAP_DIP_SP_MMV_IP_TQ_ADD_IS	6	0...FFFFH	-1.28E+3 ... 1.27996E+3	0.0390625	[hPa]
Additive torque reserve at idle at VVT-mode (mode1)					

22.6.1 Function description

General Information:

The module „Idle Speed Torque Reserve“ consists of two parts:

1. Idle speed torque reserve due to low engine speed
2. Idle speed torque reserve limitiation for stabilizing the ISC at high torque reserves

1. Idle speed torque reserve due to low engine speed

Depending on the predicted engine speed difference N_DIF_PRED a torque reserve by increasing the load is calculated. If actual engine speed falls it is possible to increase load before the actual engine speed is below the engine speed setpoint for idle speed. At the same time a spark retard compensates the supposed torque increase. So a quick torque incraease by spark advance can be realized to stabilize engine speed at its setpoint.

For negative gradients of TQ_ADD_IS_BOL_SEL a PT1-filter for TQ_ADD_IS_BOL calculation is applied to avoid torque jumps caused by fast changes of ignition angle. For positive gradients of the input TQ_ADD_IS_BOL_SEL the output value TQ_ADD_IS_BOL is identical to the input.

2. Idle speed torque reserve limitiation for stabilizing the ISC at high torque reserves

At high torque reserve requests (ex.: catalyst heating) the ignition angle can be retarded down to the minimum ignition angle IGA_MIN. This may cause problems with idle speed stability. In order to prevent speed from oscillating in case the actual ignition angle is in the near of IGA_MIN caused by high torque reserves, the maximum availible torque reserve must be limited. In order to reach this, a tunable calibration constant C_TQ_DIF_ADD_IS_TOL_OPM_1/2 is introduced. C_TQ_DIF_ADD_IS_TOL_OPM_1/2 has the same influence than TQ_ADD_IS_BOL: while the last defines a minimum torque reserve, C_TQ_DIF_ADD_IS_TOL_OPM_1/2 takes influence on the maximum torque reserve in idle. The availible torque reserve range is shown in figure 1.

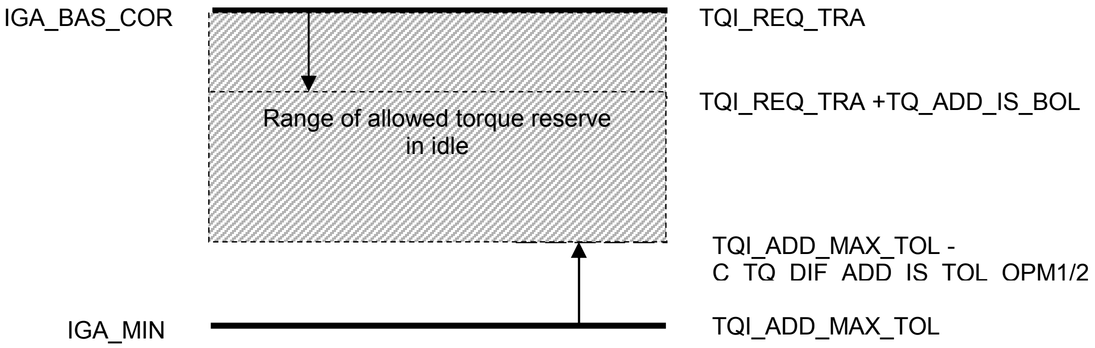


Figure 1: Availible torque reserve range after limiting:

TQI_ADD_MAX_TOL - C_TQ_DIF_ADD_IS_TOL_OPM_1/2 defines the max. idle torque reserve

Since C_TQ_DIF_ADD_IS_TOL_OPM_1/2 is a calibratable constant, this may cause a conflict situation as shown in figure 2.

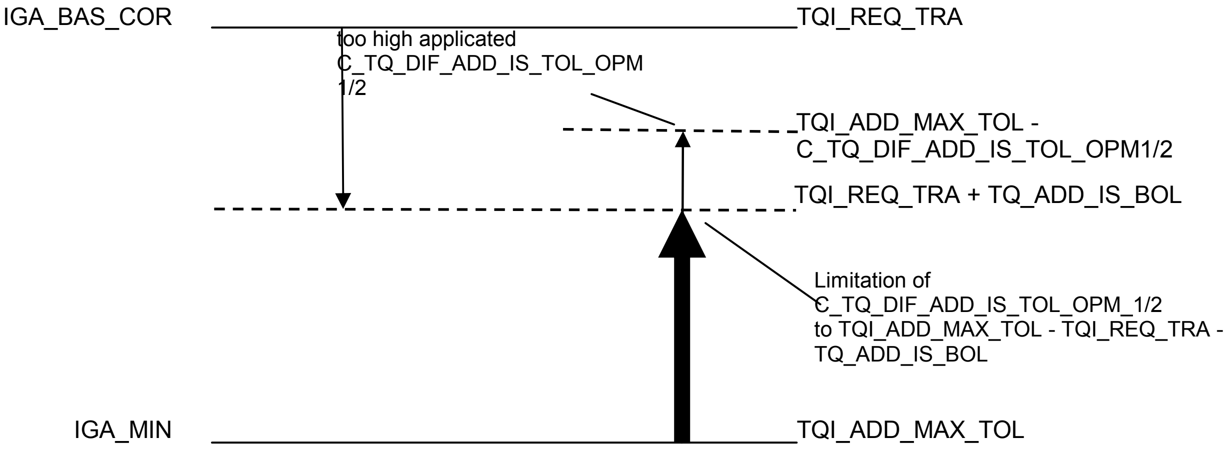


Figure 2: Limiting C_TQ_DIF_ADD_IS_TOL_OPM_1/2 in case of a conflict

In this case TQ_ADD_IS_BOL has a higher priority than C_TQ_DIF_ADD_IS_TOL_OPM_1/2, since the possibility of moving the ignition angle in direction spark advance by building up a torque reserve is more essential than keeping it at a certain point from IGA_MIN, which „only“ causes irregular running. The condition for a non-critical situation, figure 1, is

TQI_REQ_TRA+TQ_ADD_IS_BOL<= TQI_ADD_MAX_TOL - C_TQ_DIF_ADD_IS_TOL_OPM_1/2 (1)

Solving this relation leads to

C_TQ_DIF_ADD_IS_TOL_OPM_1/2<=TQI_ADD_MAX_TOL - TQI_REQ_TRA - TQ_ADD_IS_BOL (2)

If relation (2) is not fulfilled, C_TQ_DIF_ADD_IS_TOL_OPM_1/2 is limited, figure 3.

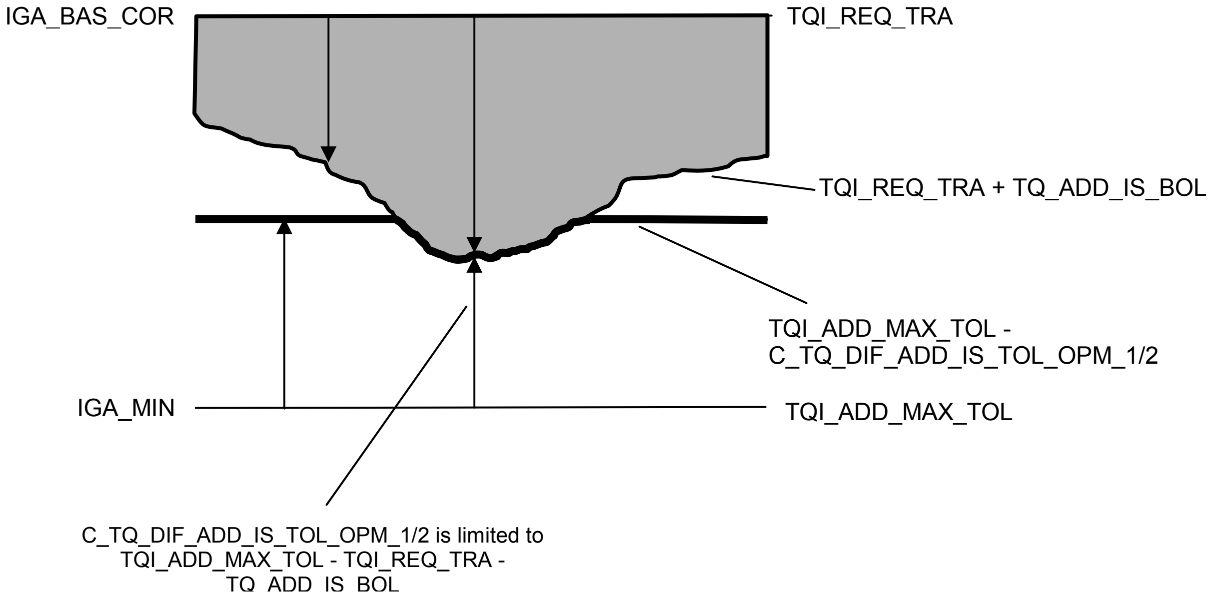


Figure 3: Limiting C_TQ_DIF_ADD_IS_TOL_OPM_1/2 in case TQI_REQ_TRA + TQ_ADD_IS_BOL is greater than TQI_ADD_MAX_TOL - C_TQ_DIF_ADD_IS_TOL_OPM_1/2

Application Condition

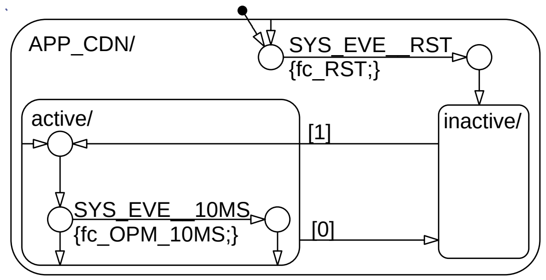
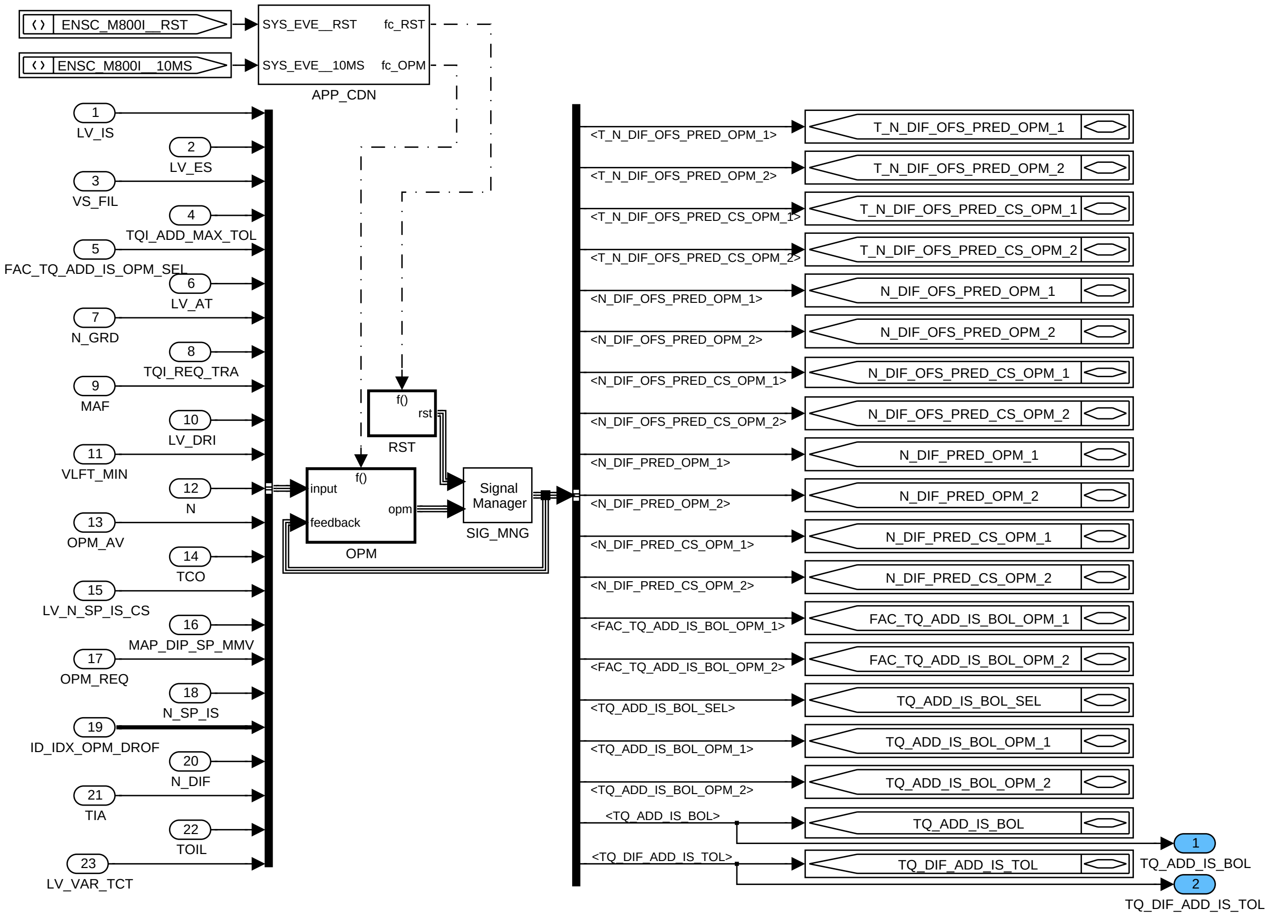


Figure 118 ENSC_M800I/ APP_CDN/ APP_CDN

Function Description



SDA_SRS / SDA 4.0 20−Jun−2007

Figure 119 ENSC_M800I

22.6.1.1 Initialization at Reset event

The following variables are initialized to zero.

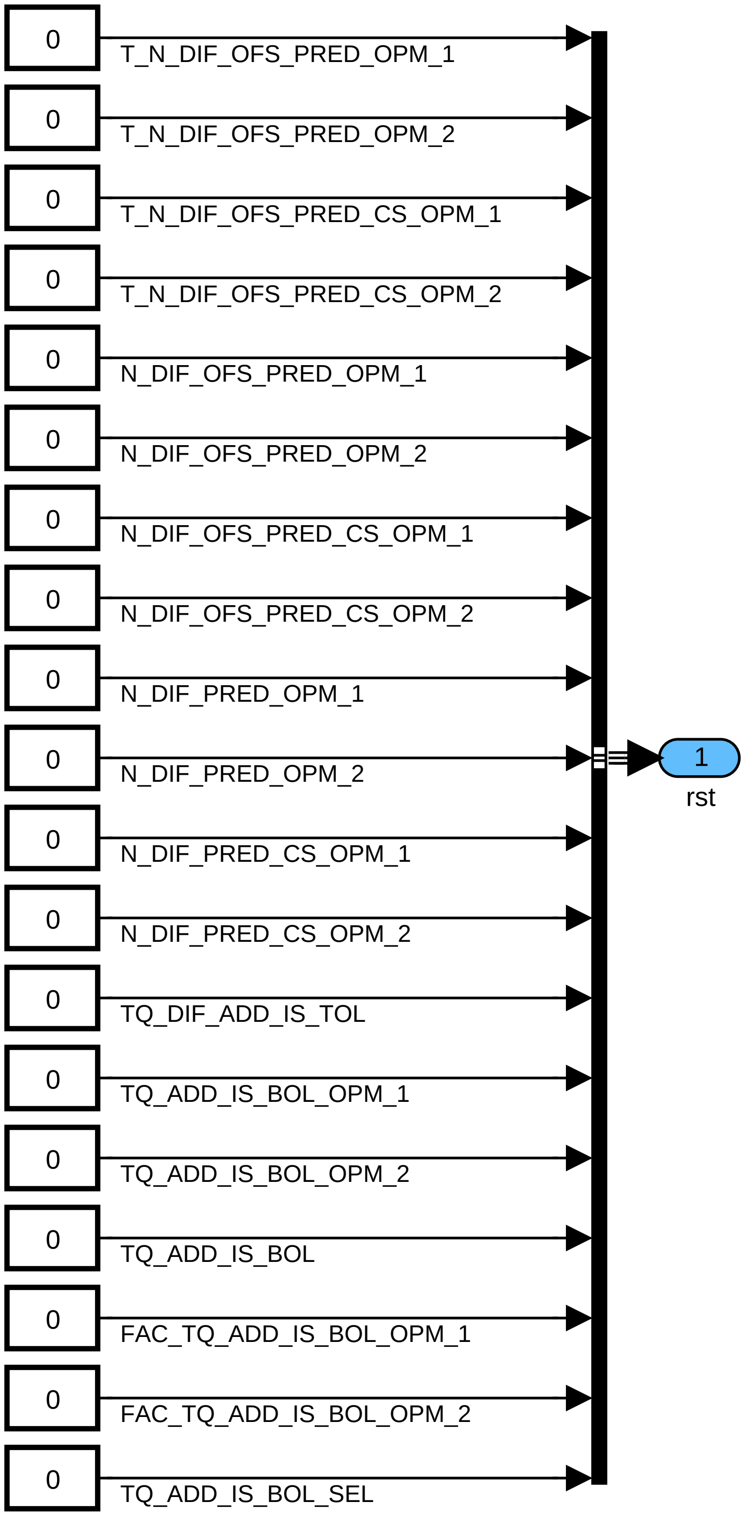


Figure 120 ENSC_M800I/ RST/ INI

22.6.1.2 Formula section

Check for activation

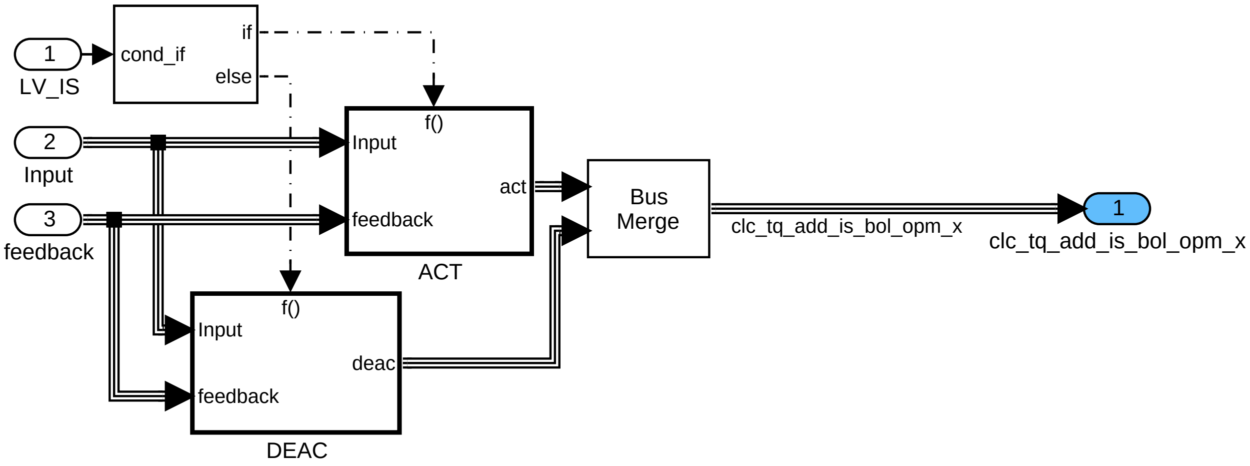


Figure 121 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x

Activation

Operation mode manager

There are two input values IN_1, IN_2. Each input value belongs to a special operation mode OPM_1, OPM_2. Moreover the value OPM_SEL_AV indicates the active and OPM_SEL_REQ indicates the requested operation mode. The aim of the operation manager is to create an output value depending on the active operation mode. If there is a switch from one mode to another a smooth changeover from one input value to the other is possible by using the interpolation factor FAC_OPM_SEL.

A special logic takes care that the needed part is calculated only. That means if OPM_1 is active and there is no switch requested then the output value of the operation manager = IN_1 and some part of the algorithm for IN_2 are not calculated anymore. In this case the bit LV_OPM_SEL_1=1 indicates that only the calculation of the IN_1-path is done.

The information of the actual and the requested operation mode can be changed by interpretations maps ID_...OPM_SEL. The effect of this is, that by calibration the operation mode 1 can be wired to the input 2. That means the belonging for a operation mode to an input value is flexible.

general specification

The following table defines the local variable described above:

TQ_ADD_IS_BOL_OPM_1	IN_1	Input	Input value for operation mode 1
TQ_ADD_IS_BOL_OPM_2	IN_2	Input	Input value for operation mode 2
OPM_AV (from Layer)	OPM_AV	Input	Actual operation mode
OPM_REQ (from Layer)	OPM_REQ	Input	Requested operation mode
-	OPM_SEL_AV	Internal value	Actual active operation mode after interpretation
-	OPM_SEL_REQ	Internal value	Requested operation mode after interpretation
FAC_TQ_ADD_IS_OPM_SEL (from Layer)	FAC_OPM_SEL	Input	Interpolation factor
LV_TQ_ADD_IS_OPM_SEL_1	LV_OPM_SEL_1	Output	Operation mode 1 active
LV_TQ_ADD_IS_OPM_SEL_2	LV_OPM_SEL_2	Output	Operation mode 2 active
TQ_ADD_IS_BOL_SEL_1	OUT	Output	Result of interpolation
LV_TQ_ADD_IS_OPM_RLS	LV_OPM_RLS	Output	Operation mode within valid range
ID_TQ_ADD_IS_OPM_AV	ID_OPM_AV	cal	Mapping of active operation mode
ID_TQ_ADD_IS_OPM_REQ	ID_OPM_REQ	cal	Mapping of requested operation mode

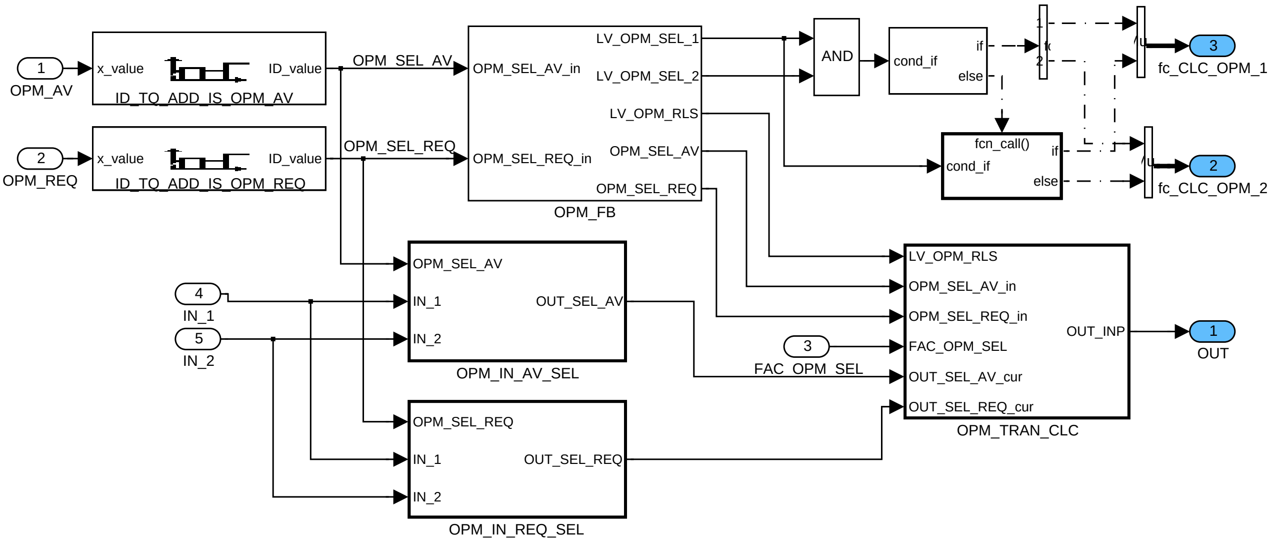


Figure 122 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ ACT/ OPM_SWI

Check for activation of operation mode 1 and 2

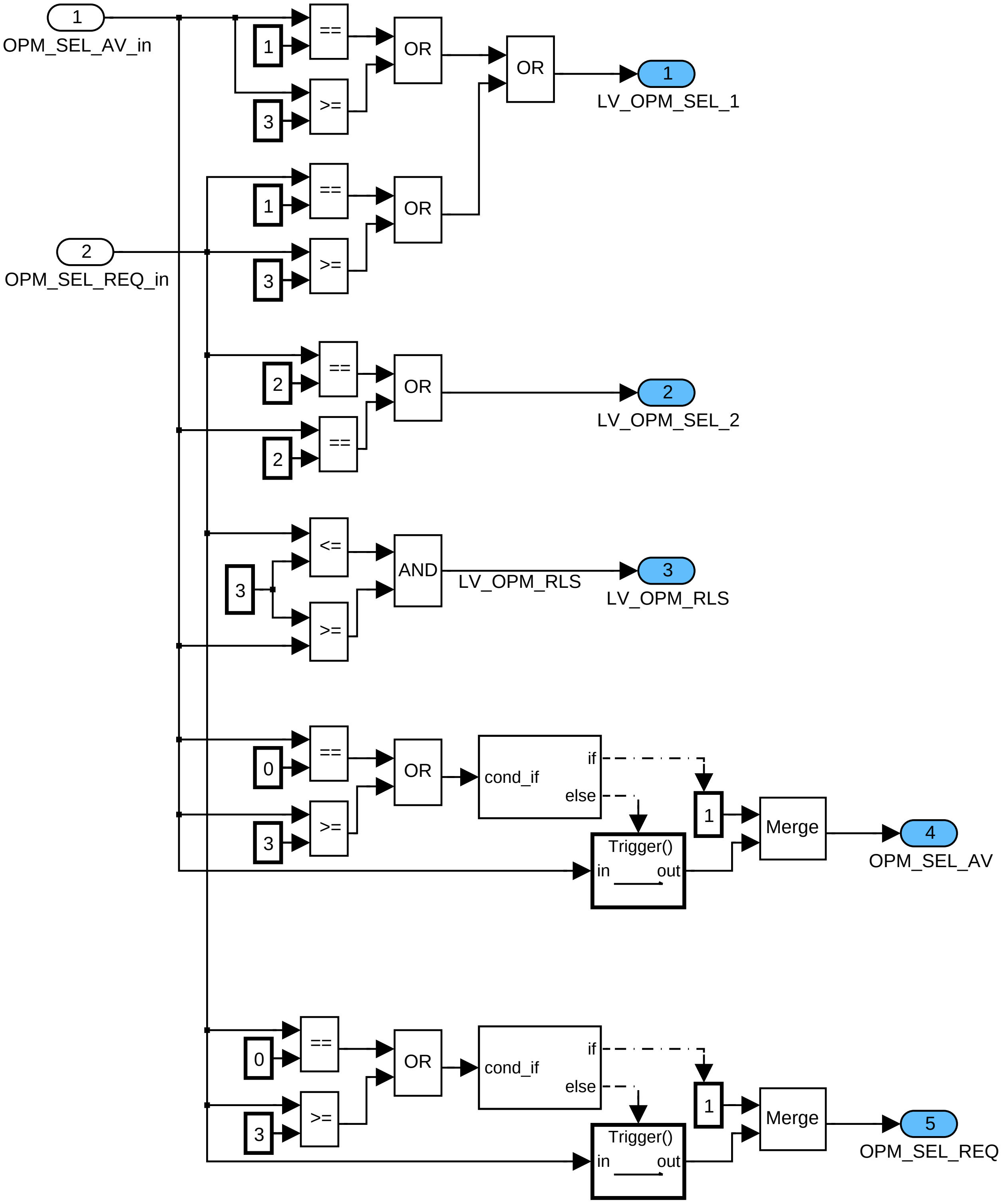


Figure 123 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ ACT/ OPM_SWI/ OPM_FB

Actual active operation mode after interpretation

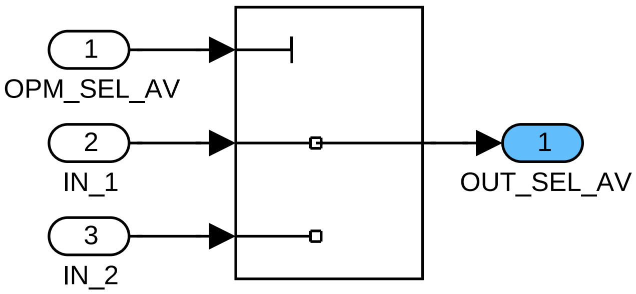


Figure 124 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ ACT/ OPM_SWI/ OPM_IN_AV_SEL

Requested operation mode after interpretation

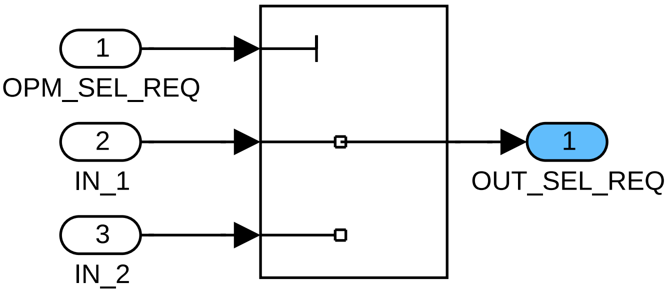


Figure 125 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ ACT/ OPM_SWI/ OPM_IN_REQ_SEL

Calculation of TQ_ADD_IS_BOL_SEL_1

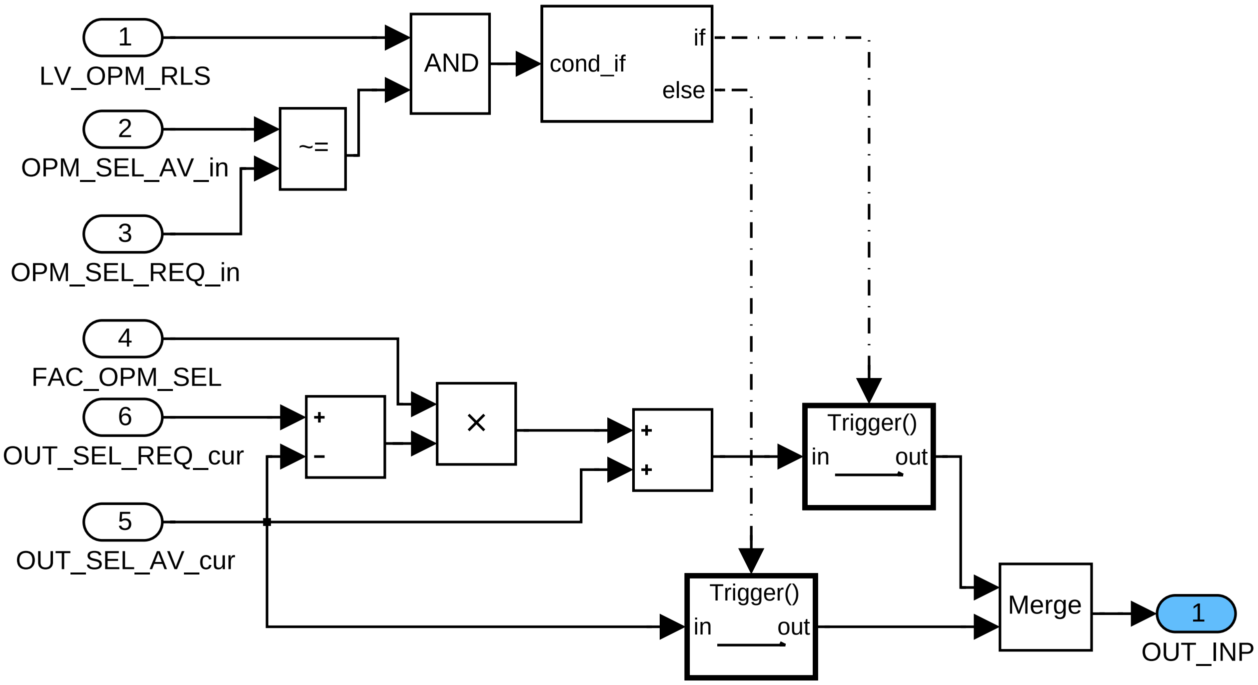


Figure 126 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ ACT/ OPM_SWI/ OPM_TRAN_CLC

TQ_ADD_IS_BOL_OPM_1 calculation

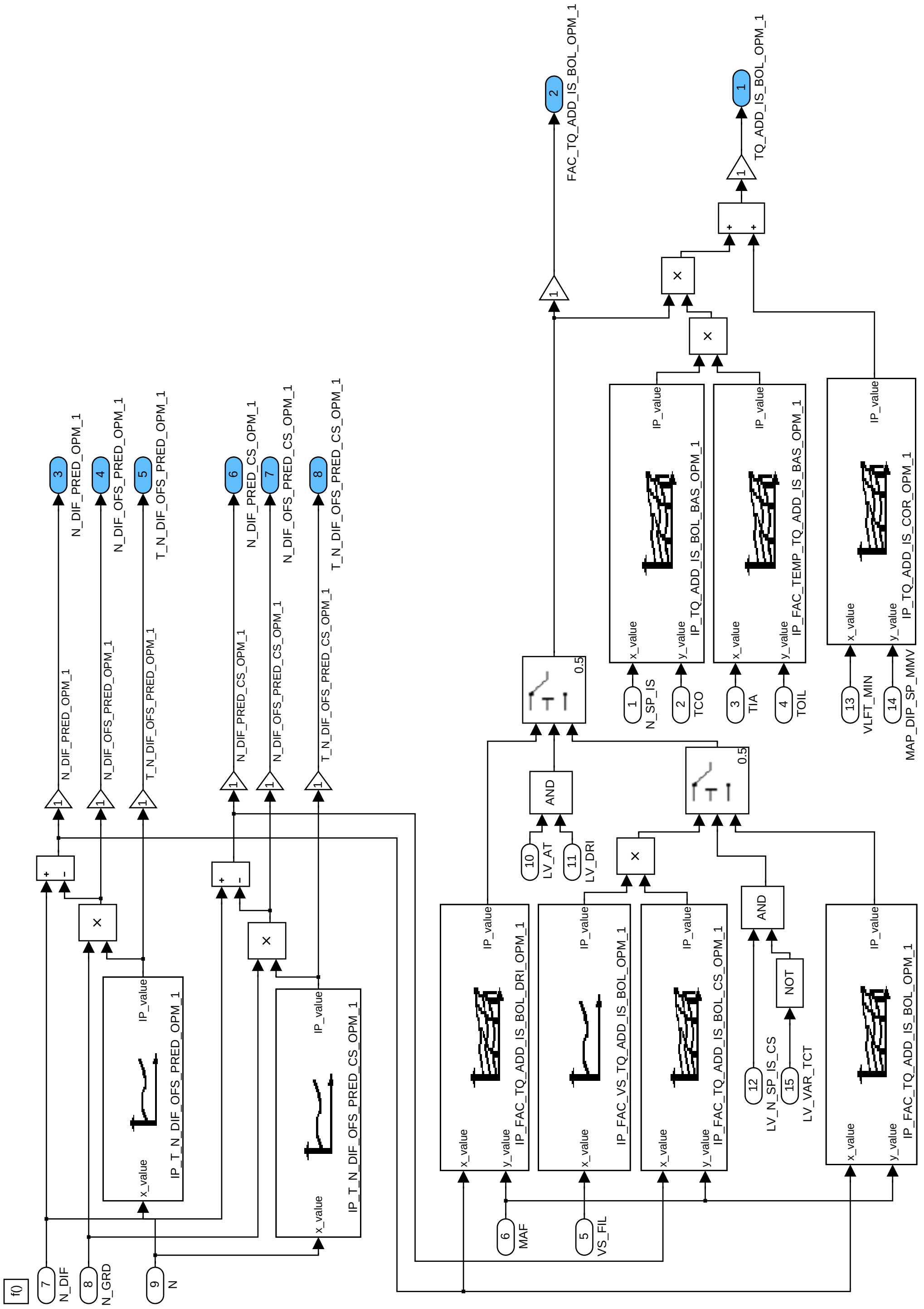


Figure 127 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ ACT/ CLC_OPM_1

TQ_ADD_IS_BOL_OPM_2 calculation

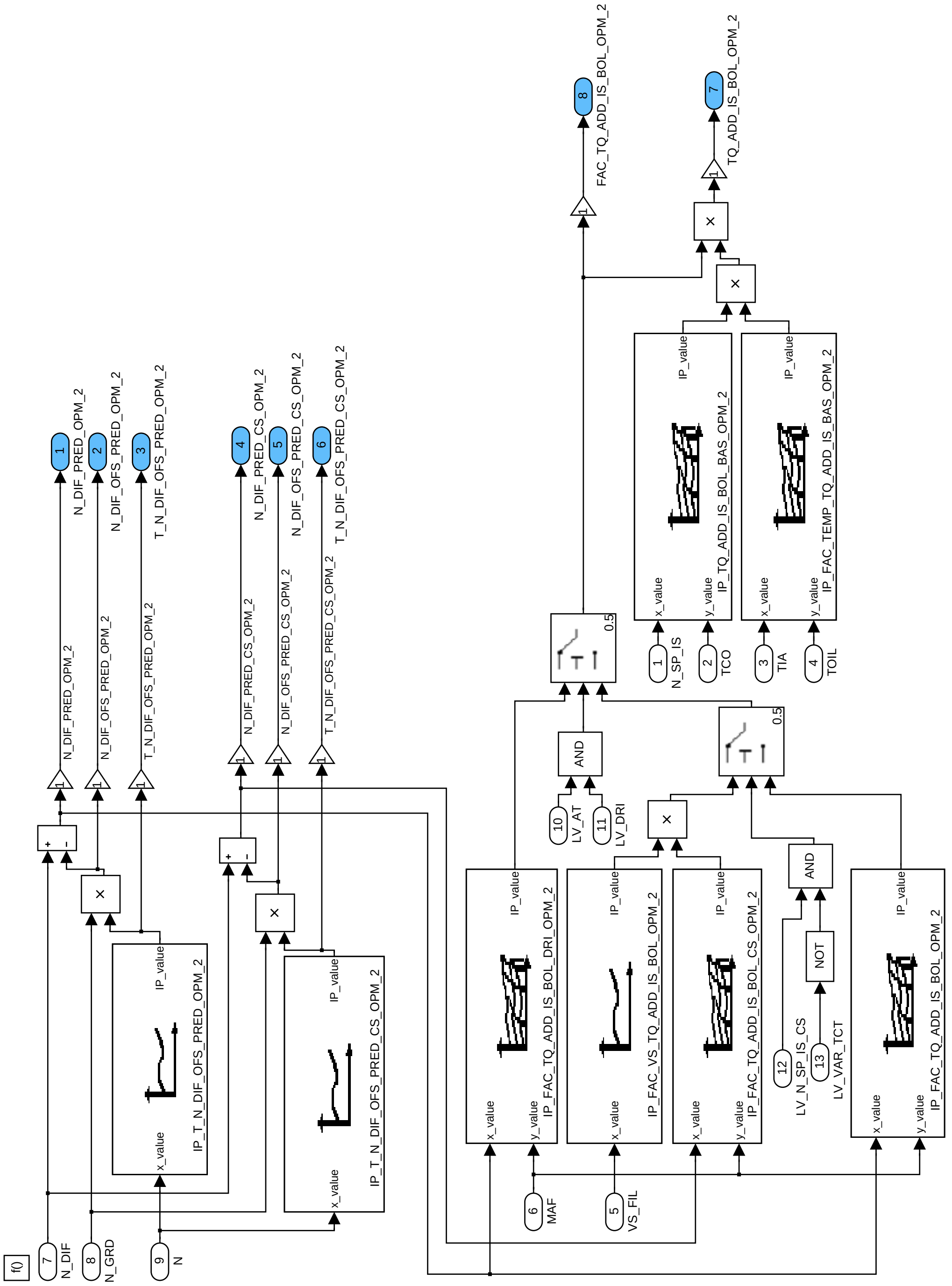


Figure 128 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ ACT/ CLC_OPM_2

Deactivation

Check for Deactivation

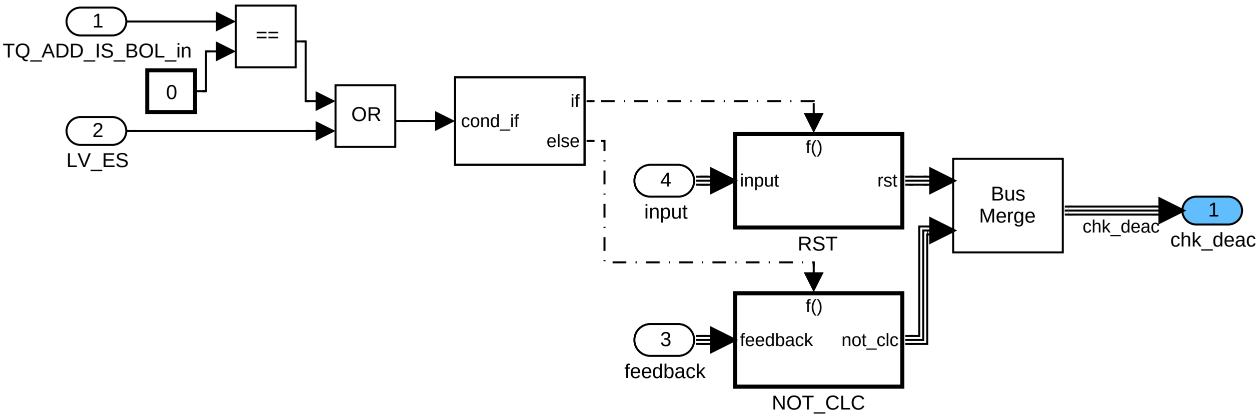


Figure 129 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ DEAC/ CHK_DEAC

Reset variables during Deactivation

The following variables are initialized during event deactivation

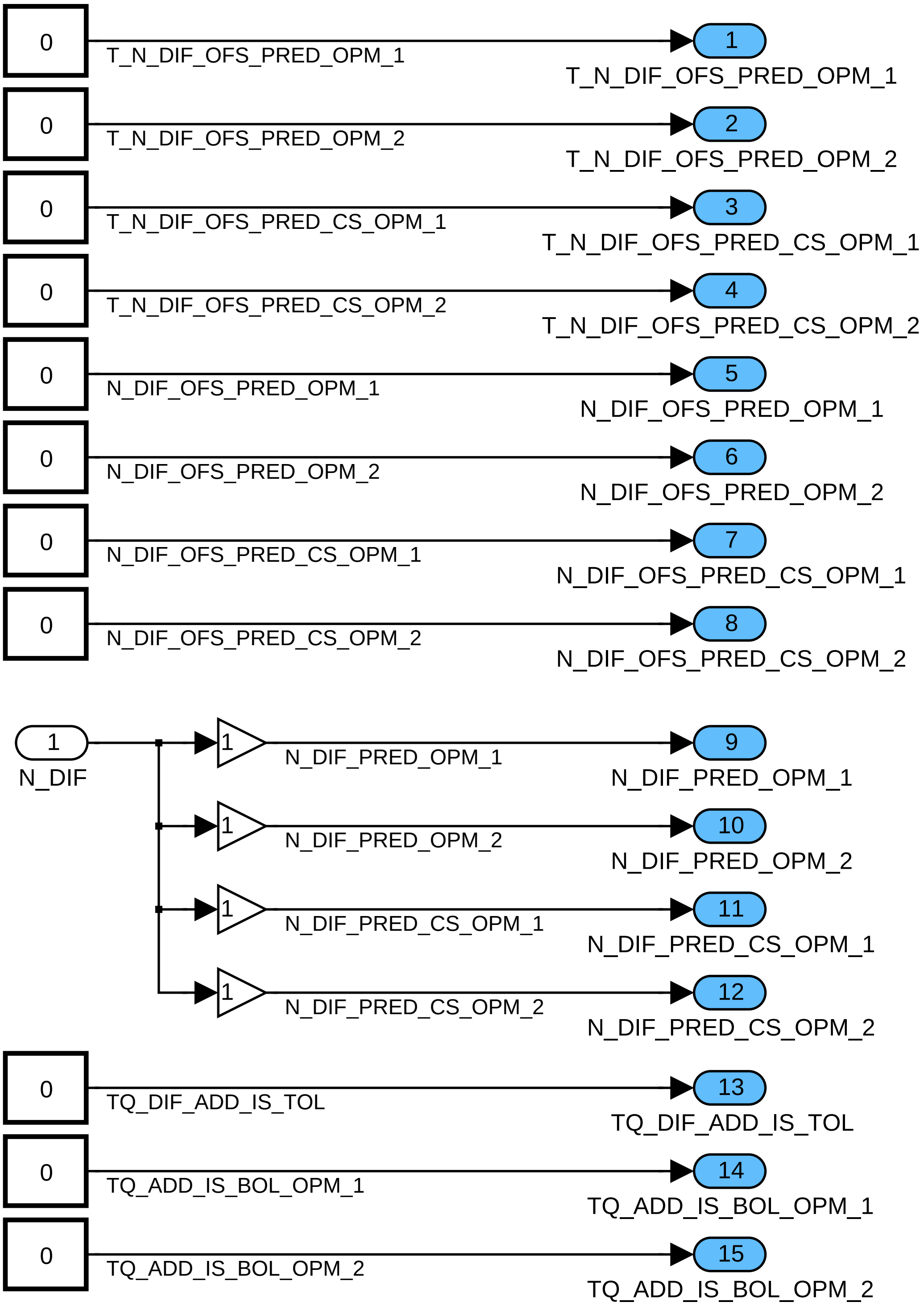


Figure 130 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ DEAC/ CHK_DEAC/ RST/ CLC

<u>NOT_CLC</u>

No calculations

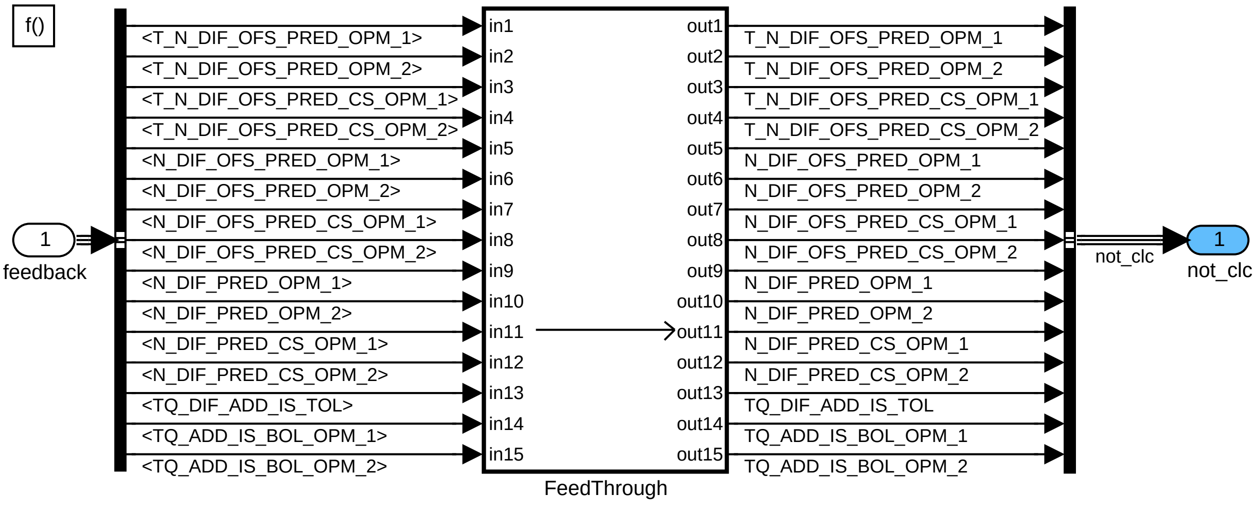


Figure 131 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ DEAC/ CHK_DEAC/ NOT_CLC

Ramp down TQ_ADD_IS_BOL_SEL

Calculations for TQ_ADD_IS_BOL_SEL

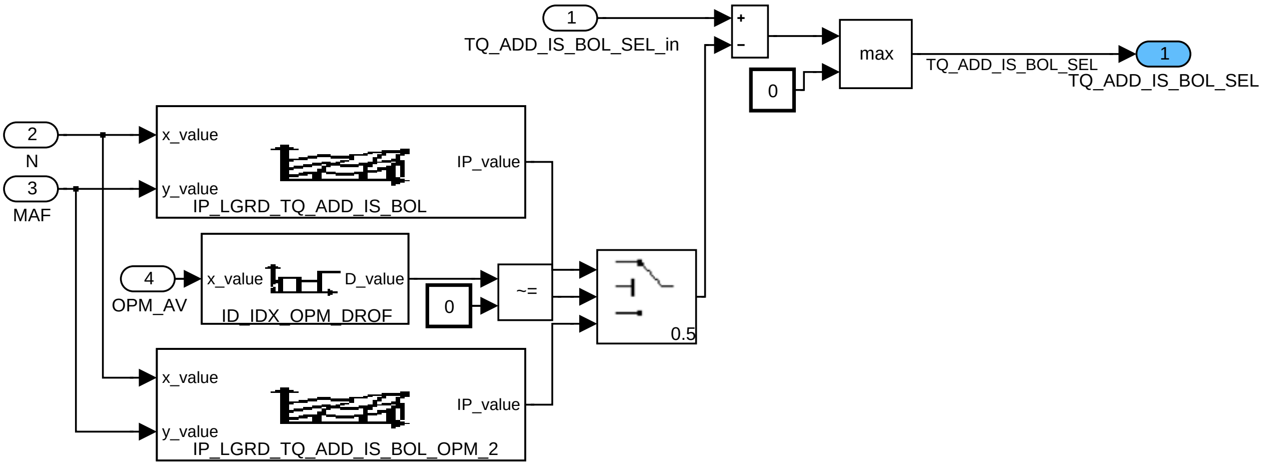


Figure 132 ENSC_M800I/ OPM/ CLC_TQ_ADD_IS_BOL_OPM_x/ DEAC/ RAMP_DOWN_TQ_ADD_IS_BOL_SEL/ CLC

PT1 Filter to limit negative gradient

For negative gradients of TQ_ADD_IS_BOL_SEL a PT1-filter for TQ_ADD_IS_BOL calculation is applied to avoid torque jumps caused by fast changes of ignition angle. For positive gradients of the input TQ_ADD_IS_BOL_SEL the output value TQ_ADD_IS_BOL is identical to the input.

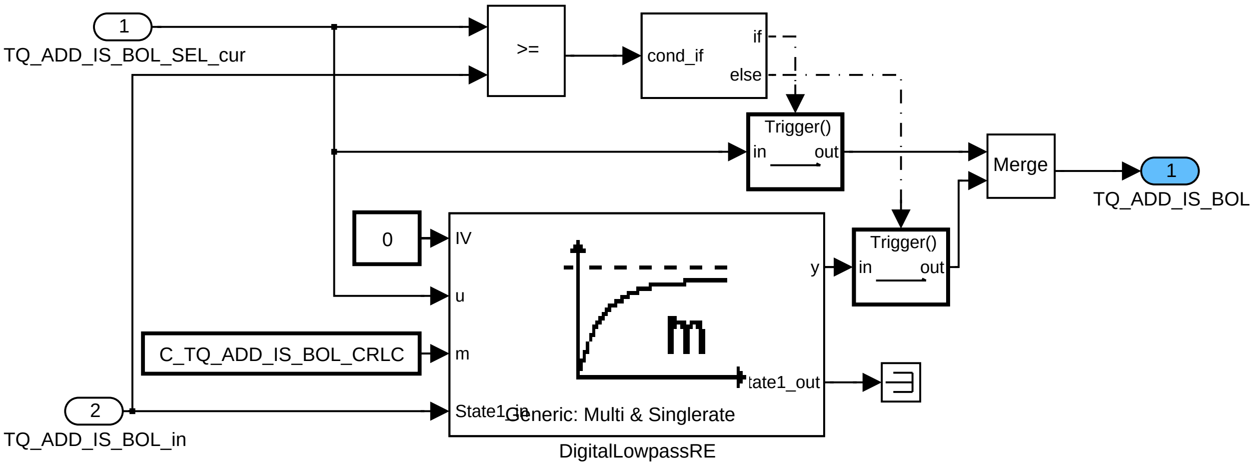


Figure 133 ENSC_M800I/ OPM/ PT1_FIL/ CLC

Limitation of idle speed torque reserve

Idle speed torque reserve for stabilizing the ISC at high torque reserves

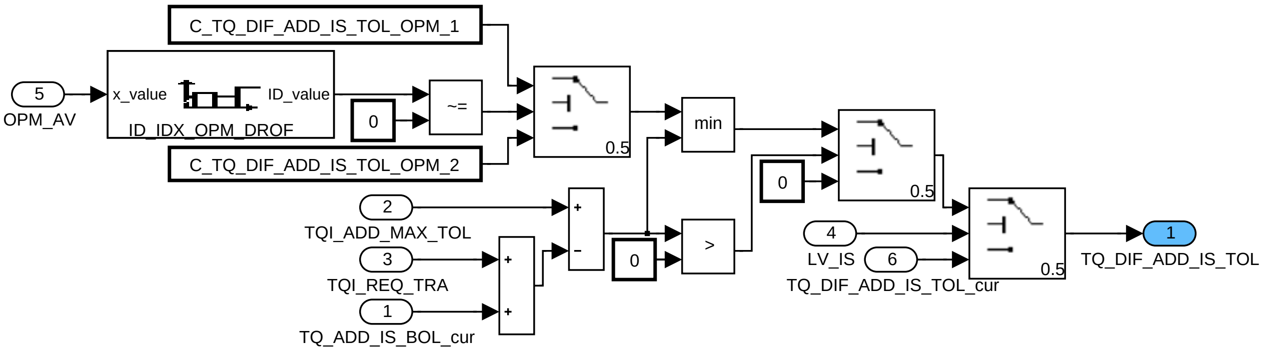


Figure 134 ENSC_M800I/ OPM/ LIM_IS_TQ/ CLC

22.7 Idle Speed Control Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_ISC	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected error sympton idle speed control diagnosis					
ERR_SYM_ISC_CST	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected error sympton idle speed control diagnosis at coldstart					
LV_CDN_DIAG_ISC	O/V	0...1H	0...1	1	-
Diagnosis conditions fulfilled					
LV_CDN_DIAG_ISC_CST	O/V	0...1H	0...1	1	-
Diagnosis conditions fulfilled idle speed at coldstart					
LV_ERR_ISC	O/V	0...1H	0...1	1	-
Boolean for fault memory Idle Control System: Unplausible Speed					
LV_ERR_ISC_CST	O/V	0...1H	0...1	1	-
Boolean for fault memory Idle Control System: Unplausible speed at coldstart					
N_DIF_MAX_DIAG_ISC_CST	V	0...1FE0H	0...8.16E+3	1	rpm
Upper engine speed deviation of idle speed control diagnosis at coldstart					
N_DIF_MIN_DIAG_ISC_CST	V	0...1FE0H	0...8.16E+3	1	rpm
Lower engine speed deviation of idle speed control diagnosis at coldstart					
T_INH_ISC_DIAG_AST	V	0...FFH	0...25.5	0.1	s
Time delay for diagnosis activation after start					
T_ISC_DIAG	V	0...FFFFH	0...6.5535E+3	0.1	s
Time delay for diagnosis activation					
LV_CDN_ISC_CST_CLU	V	0...1H	0...1	1	-
Diagnosis conditions for valid TQ_AV for diagnosis					
LV_END_DIAG_ISC	V	0...1H	0...1	1	-
Diagnosis done completely at least one time					
LV_END_DIAG_ISC_CST	V	0...1H	0...1	1	-
Diagnosis at coldstart done completely at least one time					

Input data:

FLOW_CPS	LV_ACCOUT_RLY	LV_AST_END	LV_AT
LV_CDN_VB_OBD2	LV_CH_N_SP_IS	LV_DRI	LV_IGK
LV_INH_DIAG_ISC	LV_IS	LV_MTC_CUR_OFF	LV_REQ_ISC
LV_ST_END	MAF_HB	N	N_SP_IS
TCO	TCO_ST	TQ_AV	T_AST
VS	NC_IDX_DIAG_ISC_CST	NC_IDX_DIAG_ISC	LV_VAR_TCT

0.03125 Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_ISC	1	0...FFH	0...255	1	-
Anti-bounce counter decrement for idle speed Plausibility check					
C_ABC_DEC_ISC_CST	1	0...FFH	0...255	1	-
Anti-bounce counter decrement for idle speed Plausibility check at coldstart					
C_ABC_INC_ISC	1	0...FFH	0...255	1	-
Anti-bounce counter increment for idle speed Plausibility check					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_ISC_CST	1	0...FFH	0...255	1	-
Anti-bounce counter increment for idle speed Plausibility check at coldstart					
C_ABC_MAX_ISC	1	1...FFH	1...255	1	-
Maximum value of the anti-bounce counter for idle speed Plausibility check					
C_ABC_MAX_ISC_CST	1	1...FFH	1...255	1	-
Maximum value of the anti-bounce counter for idle speed Plausibility check at coldstart					
C_FLOW_CPS_MAX_DIAG_ISC	1	0...FFFFH	0...7.99987793	1.2207E-4	kg/h
Maximum canister purge fuel flow from CPS to cylinder to perform idle speed plausibility check					
C_MAF_MAX_DIAG_ISC	1	0...FFH	0...1.389E+3	5.44705882	mg/stk
Maximum mass air flow to perform idle speed Plausibility check					
C_N_DIF_MAX_DIAG_ISC	1	0...1FE0H	0...8.16E+3	1	rpm
Maximum deviation above nominal idle speed					
C_N_DIF_MIN_DIAG_ISC	1	0...1FE0H	0...8.16E+3	1	rpm
Maximum deviation below nominal idle speed					
C_TCO_MIN_ISC	1	0...FEH	-48...142.5	0.75	°C
Min TCO threshold for idle speed diagnosis					
C_TCO_MIN_ISC_CST	1	0...FEH	-48...142.5	0.75	°C
Min TCO threshold for idle speed diagnosis at coldstart					
C_TQ_AV_MAX_DIAG_ISC_CST_AT	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Actual engine torque max level for ISC_CST diagnosis for automatic transmission					
C_TQ_AV_MAX_DIAG_ISC_CST_MT	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Actual engine torque max level for ISC_CST diagnosis for manual transmission					
C_TQ_AV_MAX_DIAG_ISC_CST_TCT	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Actual engine torque max level for ISC_CST diagnosis for automatic transmission					
C_T_DLY_ISC_DIAG	1	0...FFFFH	0...6.5535E+3	0.1	s
Time delay for diagnosis activation					
LC_INH_DIAG_ISC_MAN	1	0...1H	0...1	1	-
Flag for manual inhibitation of ISC diagnosis					
LC_T_INH_ISC_DIAG_AST	1	0...1H	0...1	1	-
Switch for inhibitation of activation timer after start of ISC diagnosis					
IP_FAC_DIF_DIAG_ISC_CST	6	0...FFH	0...1.9921875	0.0078125	-
LDP_T_AST_IP_FAC_DIAG_ISC_CST	6	0...FFFFH	0...6.5535E+3	0.1	[s]
Influence of time after start for limits of idle speed controller diagnosis at coldstart					
IP_T_INH_ISC_DIAG_AST	6	0...FFH	0...25.5	0.1	s
LDP_TCO_ST_IP_T_INH_ISC_DIAG	6	0...FEH	-48...142.5	0.75	[°C]
Initialization for activation timer of idle speed diagnosis after start					
IP_N_DIF_MAX_DIAG_ISC_CST	6x6	0...1FE0H	0...8.16E+3	1	rpm
LDP_TCO_IP_DIAG_ISC_CST	6	0...FEH	-48...142.5	0.75	[°C]
LDP_TCO_ST_IP_DIAG_ISC_CST	6	0...FEH	-48...142.5	0.75	[°C]
Upper engine speed deviation for limits idle speed control diagnosis at coldstart					
IP_N_DIF_MIN_DIAG_ISC_CST	6x6	0...1FE0H	0...8.16E+3	1	rpm
LDP_TCO_IP_DIAG_ISC_CST	6	0...FEH	-48...142.5	0.75	[°C]
LDP_TCO_ST_IP_DIAG_ISC_CST	6	0...FEH	-48...142.5	0.75	[°C]
Lower engine speed deviation for limits idle speed control diagnosis at coldstart					

Import actions:

ACTION_ERRM_GetErrSym(IN <IDX_DIAG>, OUT <ERR_SYM>)
Action that returns the symptom of the failure
ACTION_ERRM_GetLvCdnDiag(IN <IDX_DIAG>, OUT <LV_CDN_DIAG>)
Action that returns the diagnostic condition
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter

22.7.1 Function description

Application Condition

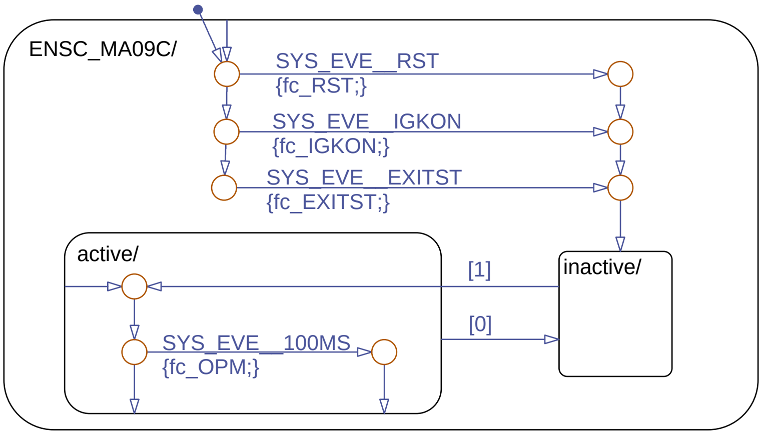
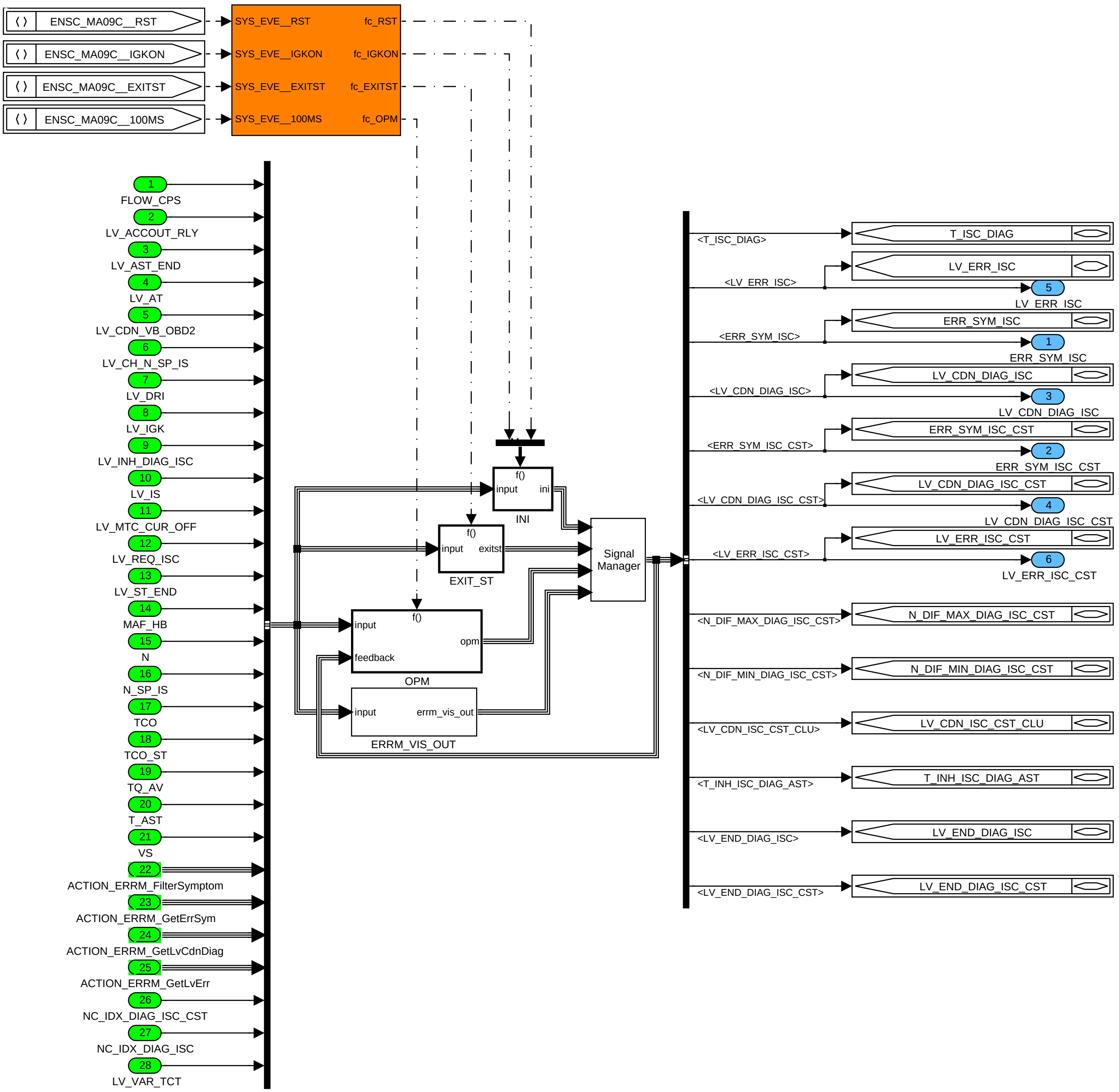


Figure 135 ENSC_MA09C/ APP_CDN/ Chart



SDA_SRS / SDA 4.0 26−Oct−2006

Figure 136 ENSC_MA09C

22.7.1.1 Initialisation at system event reset.

Initialisation V/O mode ERRM variable at system reset

according ABC type **"STD_INI"** + **"DEC_CAL"**

(0 at LV_IGK 0->1 or reset, calibratable decrement)

All the V/O mode ERRM variable are initialised according to the ERRM template.

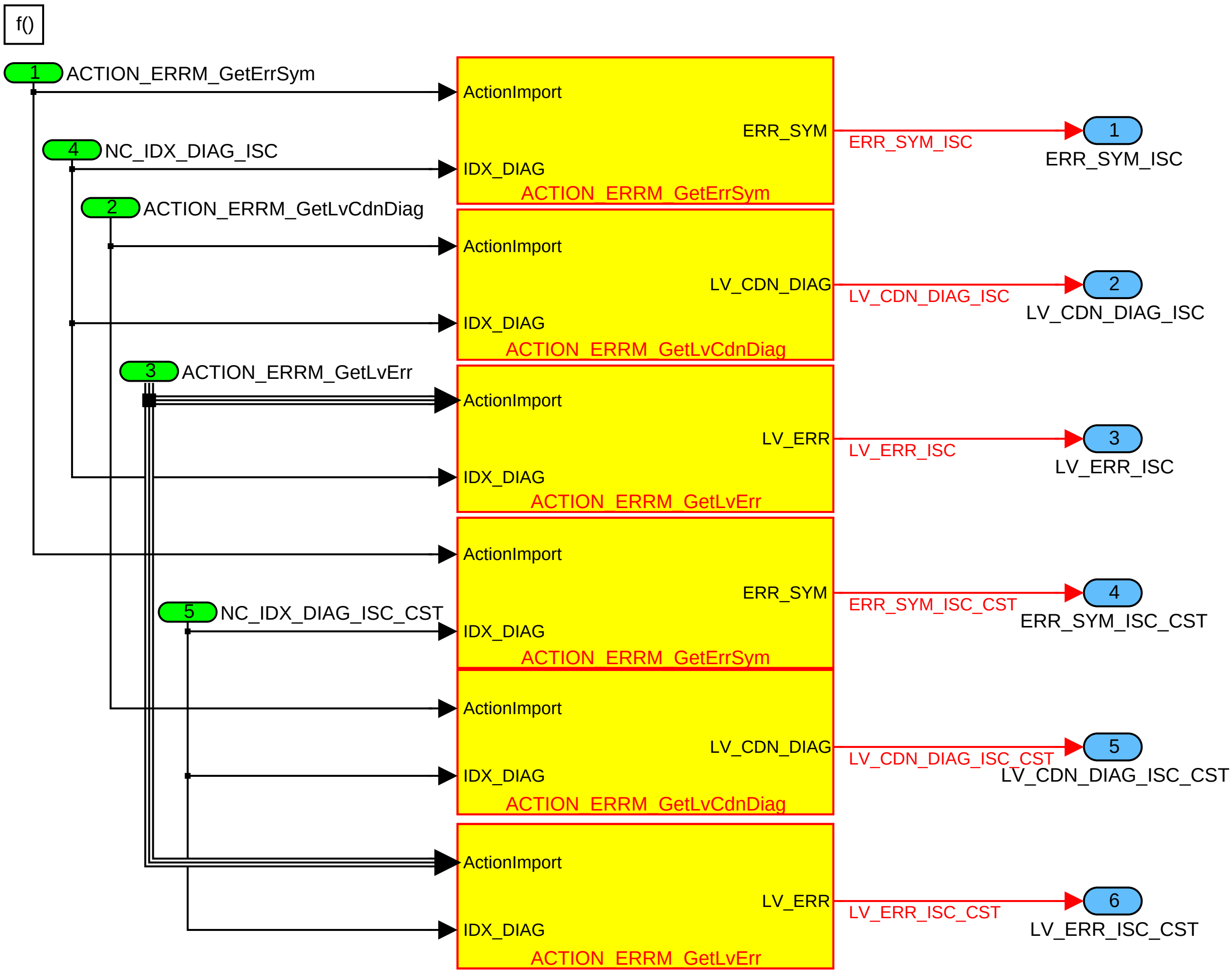


Figure 137 ENSC_MA09C/ INI/ INI_1

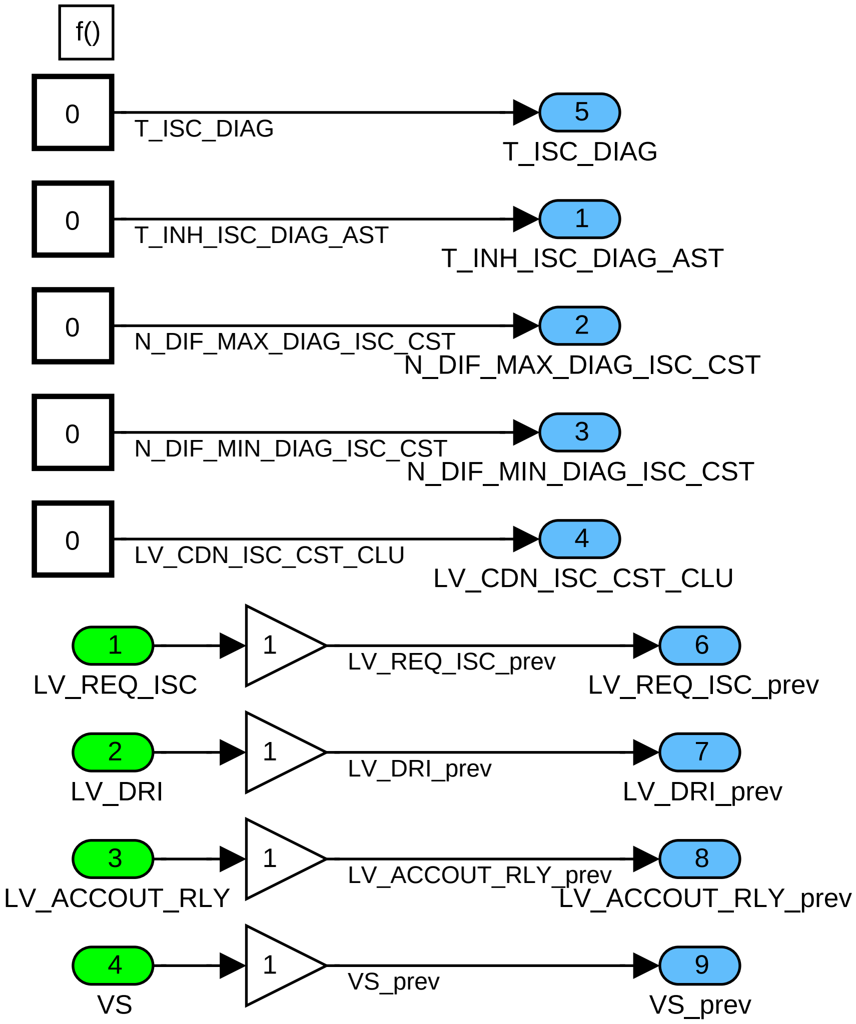


Figure 138 ENSC_MA09C/ INI/ INI_2

22.7.1.2 Condition check at EXIT_ST system event

Condition check when the transition of LV_ST_END 0 --> 1 and calculation of variable T_INH_ISC_DIAG_AST

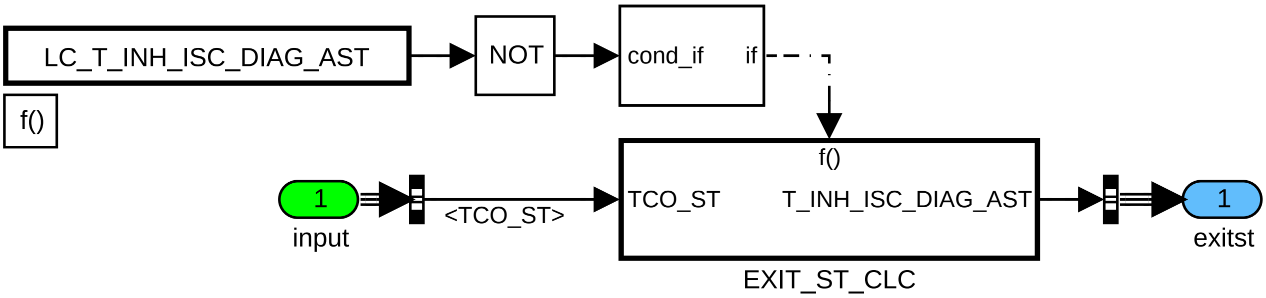


Figure 139 ENSC_MA09C/ EXIT_ST

<u>Calculation of T_INH_ISC_DIAG_AST</u>

T_INH_ISC_DIAG_AST is calculated using IP Curve IP_T_INH_ISC_DIAG_AST with TCO_ST as input.

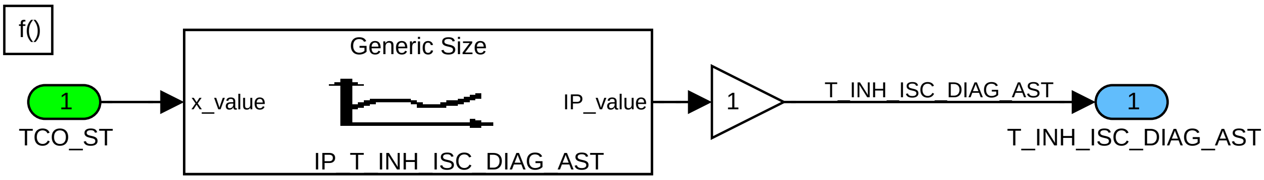


Figure 140 ENSC_MA09C/ EXIT_ST/ EXIT_ST_CLC

22.7.1.3 Plausibility check

FUNCTION DESCRIPTION:

General information:

Engine speed deviation from the nominal engine speed is monitored either the vehicle is stopped and the throttle is settled after closing or after engine start. Due to different engine operating characteristics the coldstart-monitoring is performed by using limits depending on the temperature of cooling water and time after start.

If the difference to the nominal idle speed keeps too high or too low in spite of an activated idle speed controller, an idle speed control plausibility error is detected.

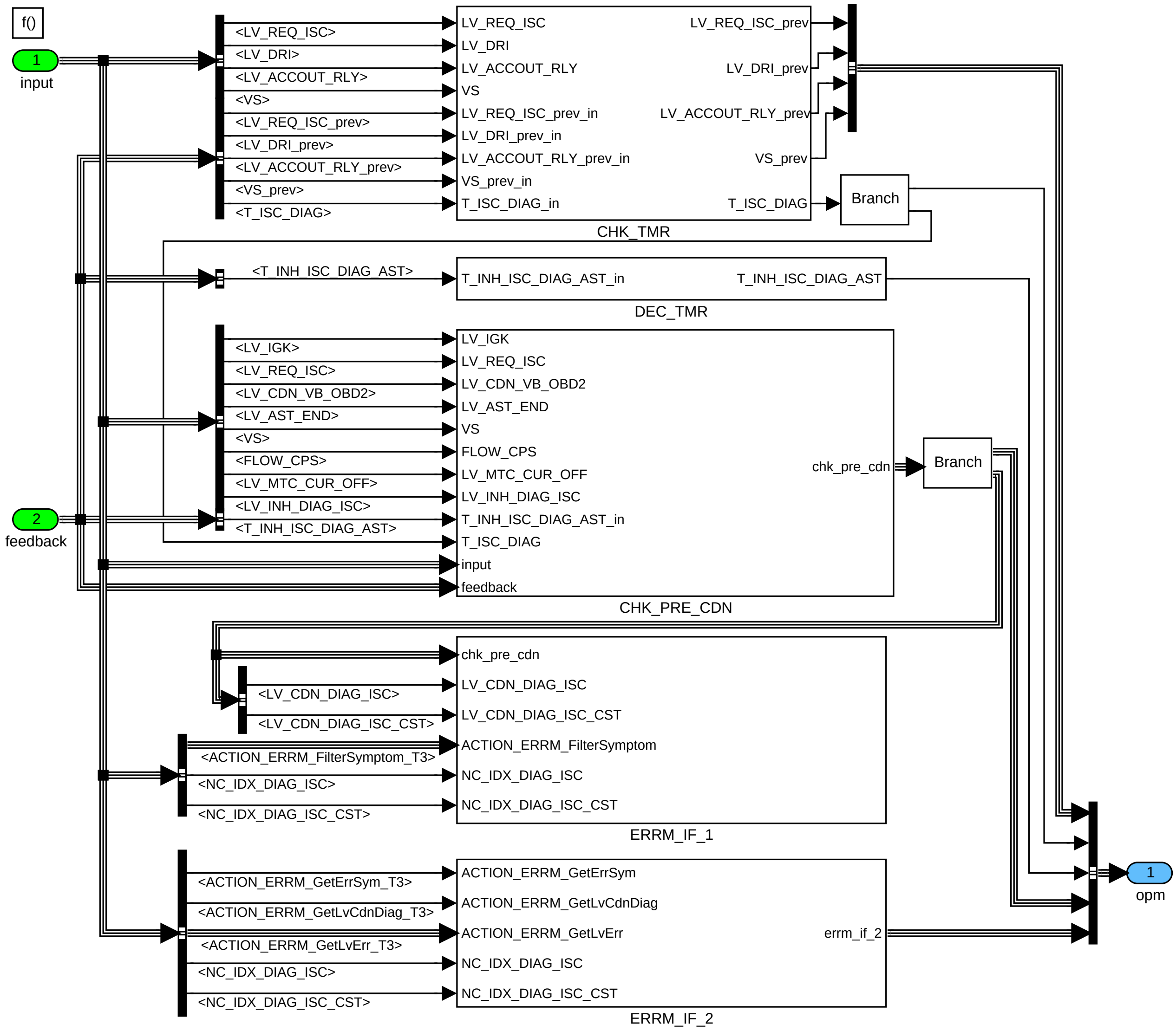


Figure 141 ENSC_MA09C/ OPM

Calculation of timer T_ISC_DIAG

The following timer is used for delaying the activation of the diagnosis algorithm when the idle speed controller is activated or speed is possibly excited due to strong disturbances such as ACC or drive train engaging (in order to prevent a diagnosis at high, normal speed deviations):

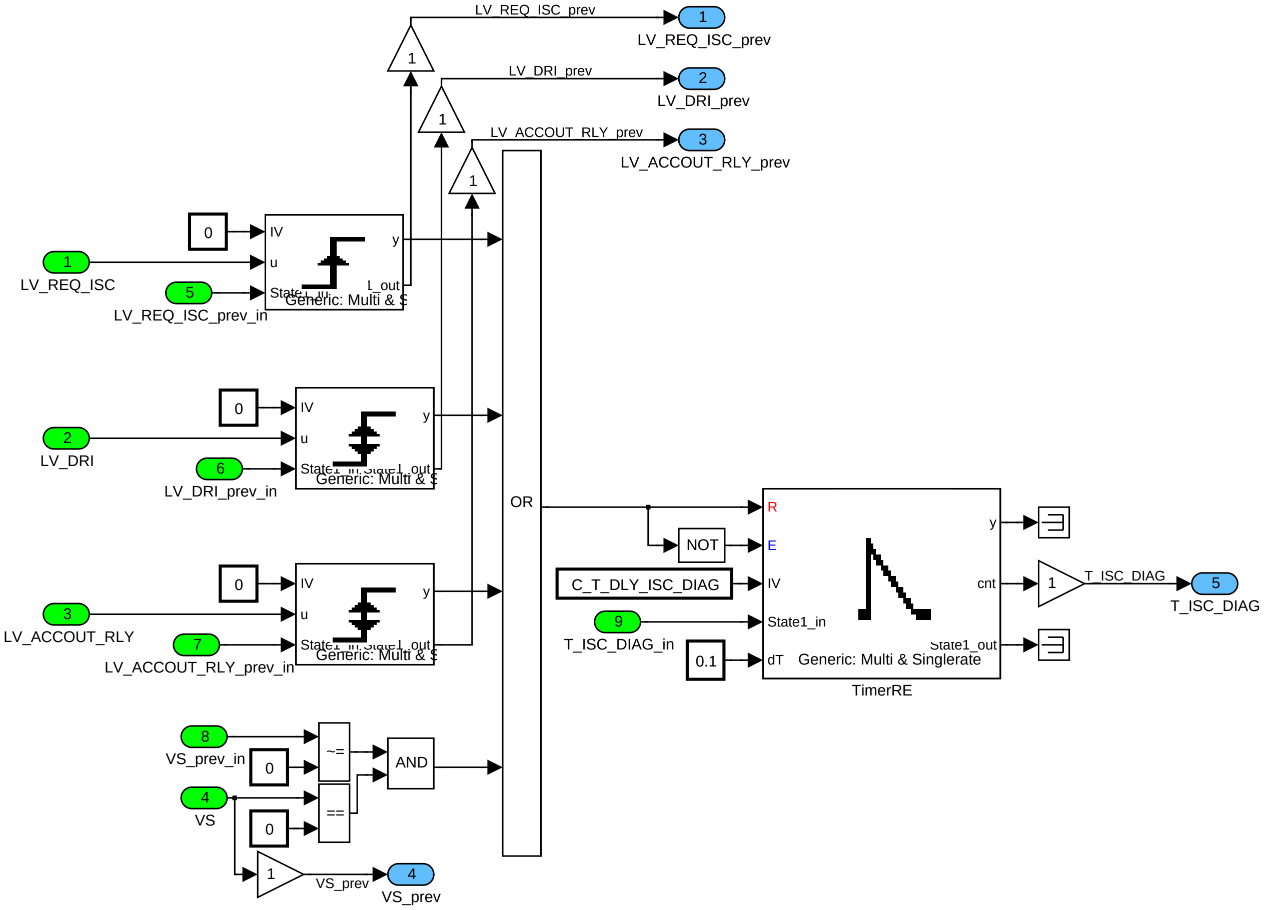


Figure 142 ENSC_MA09C/ OPM/ CHK_TMR

Calculation of timer T_INH_ISC_DIAG_AST

Timer is already initialised at system event EXITST. Here the timer starts decrementing.

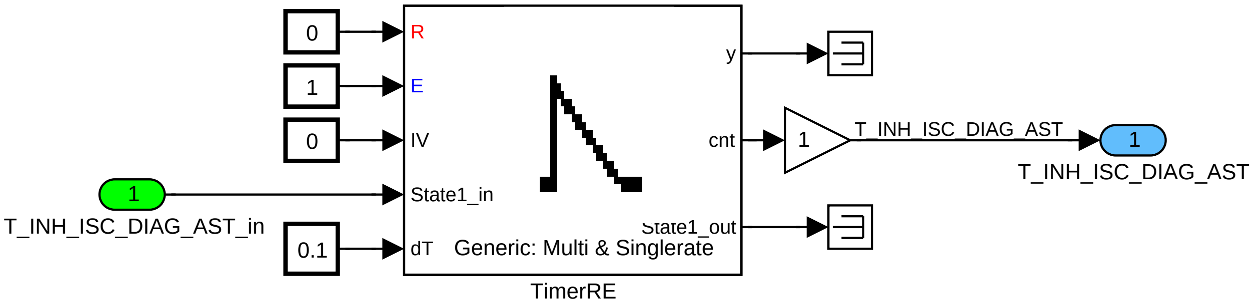


Figure 143 ENSC_MA09C/ OPM/ DEC_TMR

Checking of pre-condition

The plausibility diagnosis is performed only if the pre conditions are fulfilled

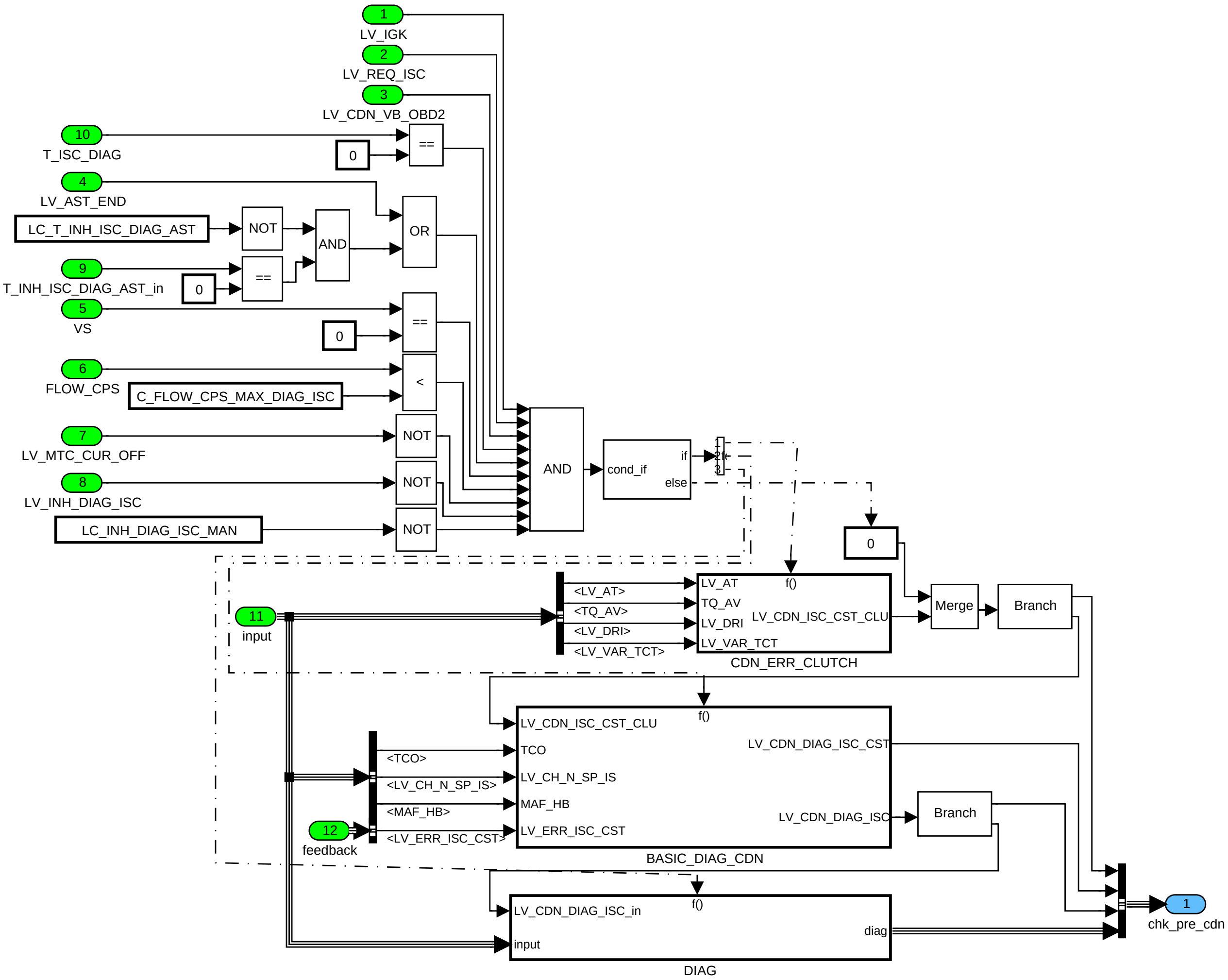


Figure 144 ENSC_MA09C/ OPM/ CHK_PRE_CDN

Condition to check clutch error

Condition to prevent error in case of clutch use

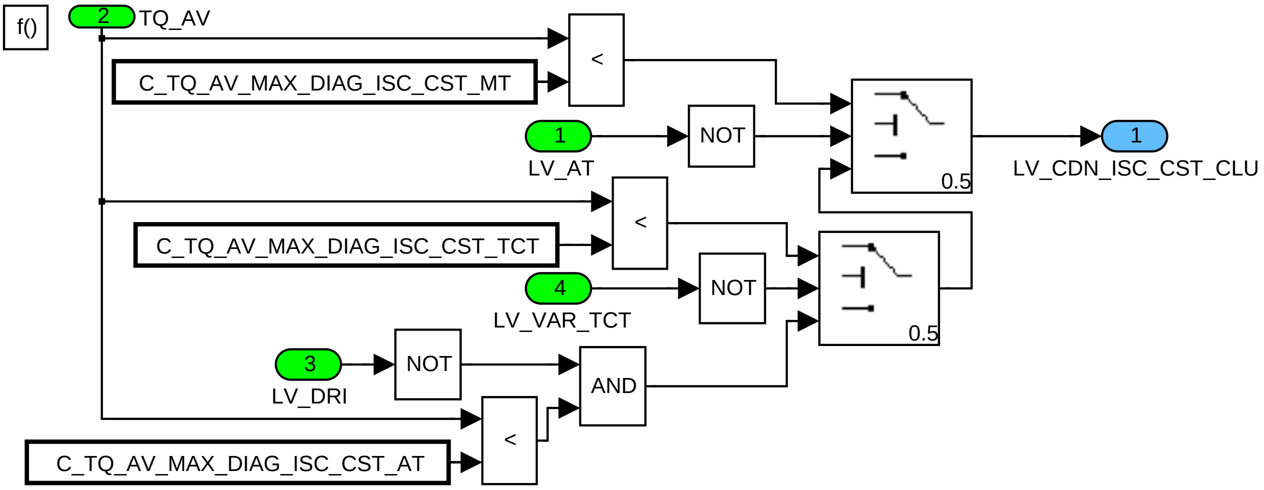


Figure 145 ENSC_MA09C/ OPM/ CHK_PRE_CDN/ CDN_ERR_CLUTCH

Calculation of ERRM condition check

Calculation LV_CDN_DIAG_ISC and LV_CDN_DIAG_ISC_CST is done here for diagnosis calculation part. of ERRM.

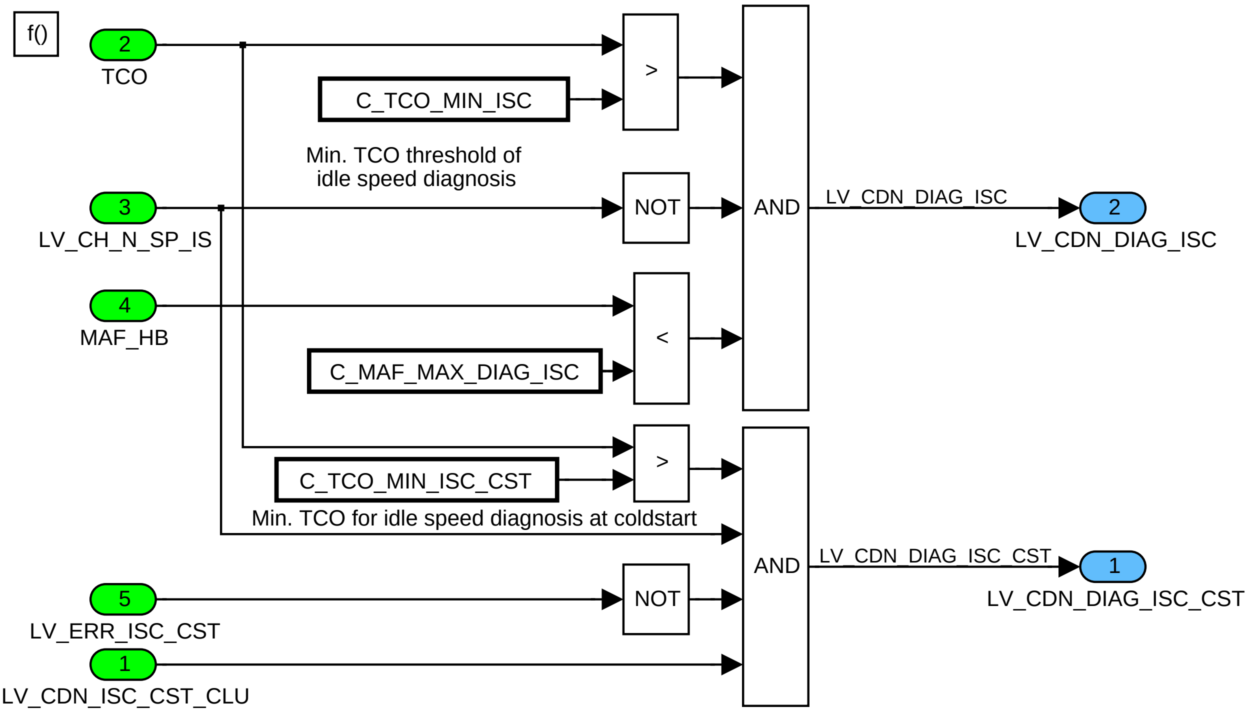


Figure 146 ENSC_MA09C/ OPM/ CHK_PRE_CDN/ BASIC_DIAG_CDN

ERRM diagnosis

Diagnosis at warmstart

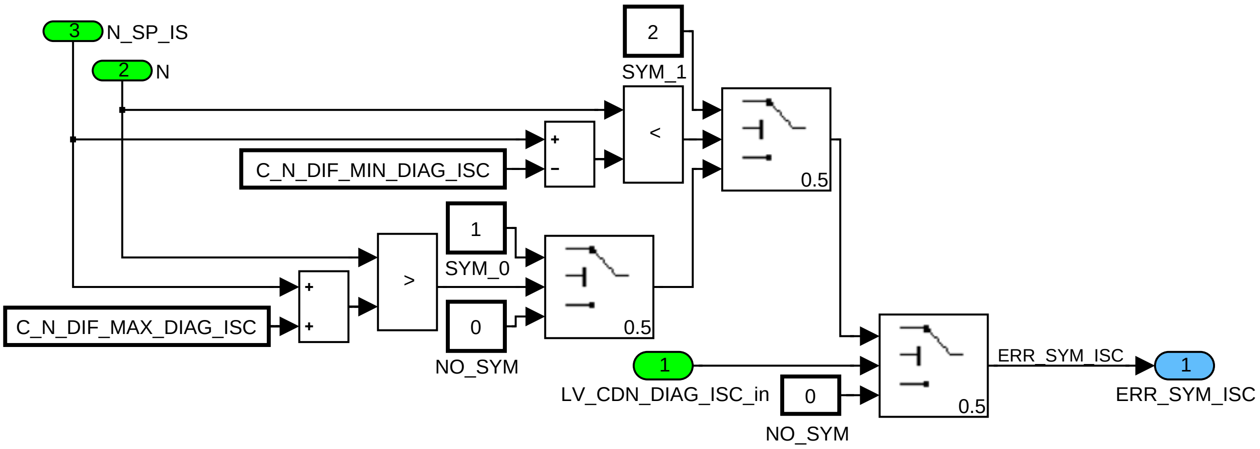


Figure 147 ENSC_MA09C/ OPM/ CHK_PRE_CDN/ DIAG/ WARMSTART

Diagnosis at coldstart

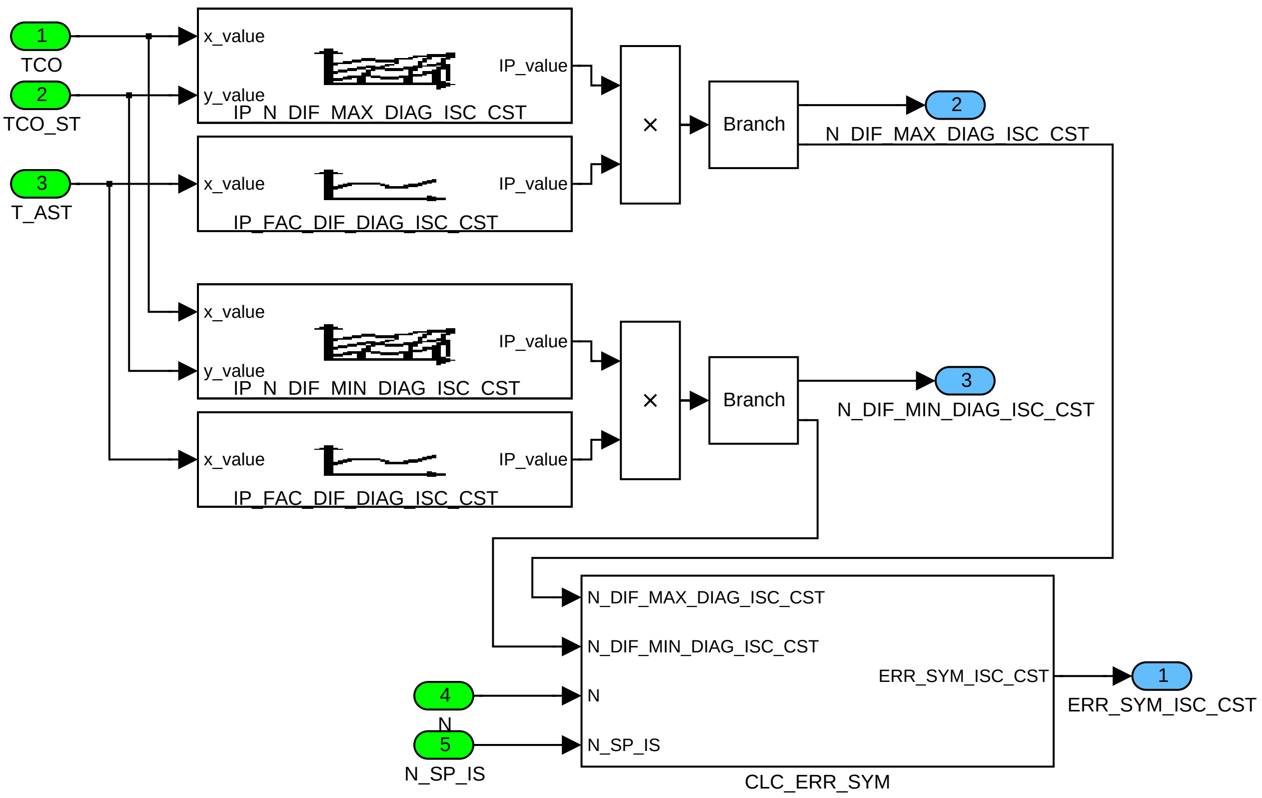


Figure 148 ENSC_MA09C/ OPM/ CHK_PRE_CDN/ DIAG/ COLDSTART

Calculation of ERR_SYM_ISC_CST

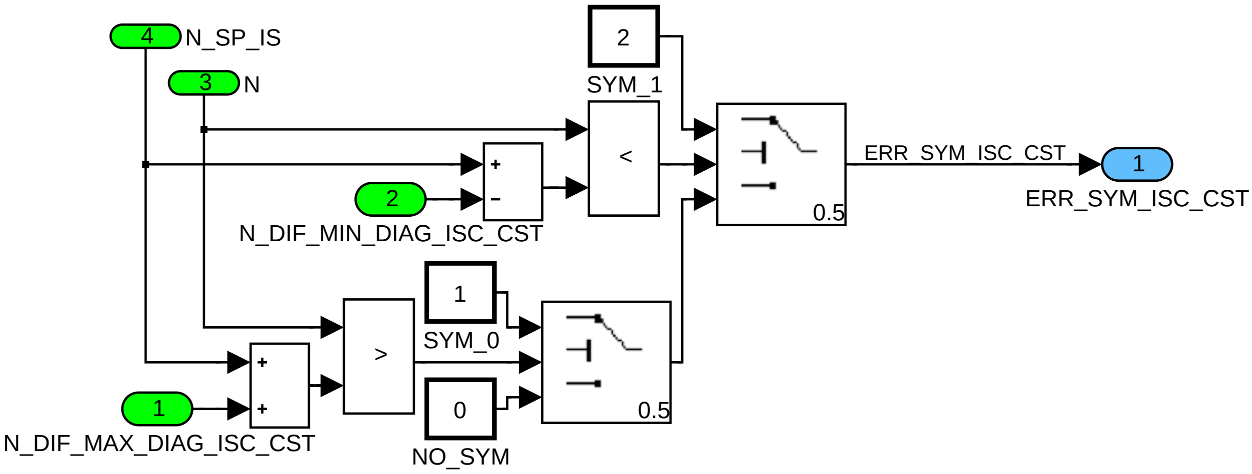


Figure 149 ENSC_MA09C/ OPM/ CHK_PRE_CDN/ DIAG/ COLDSTART/ CLC_ERR_SYM

ERRM interface 1

LV_ERR_ISC and LV_ERR_ISC_CST are terminated since they are in V/O mode. They will be calculated using the ERRM standard template for V/O mode ERRM variable.

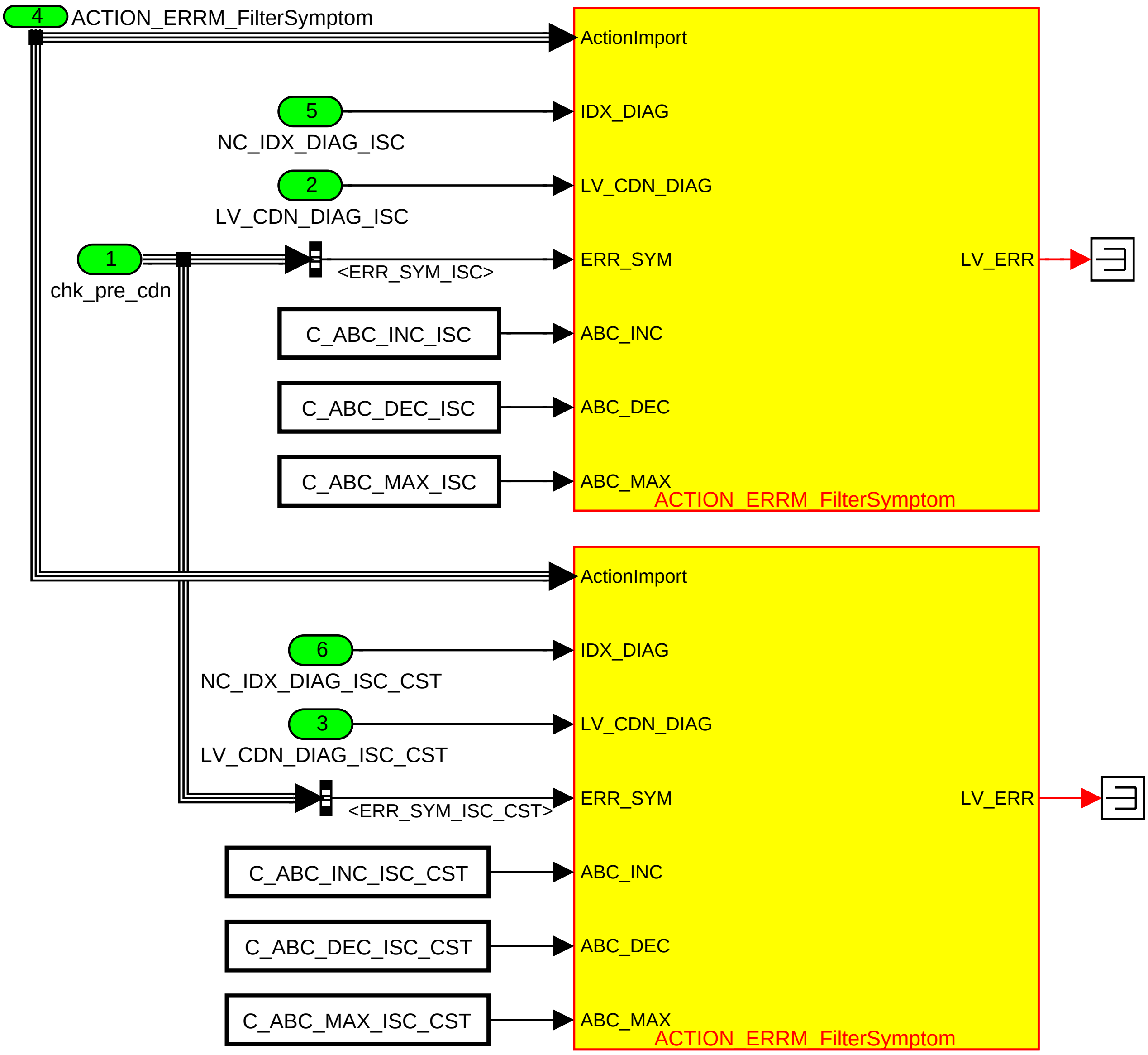


Figure 150 ENSC_MA09C/ OPM/ ERRM_IF_1

ERRM interface to calculate V/O mode ERRM variables.

V/O mode ERRM variables are calculated here.

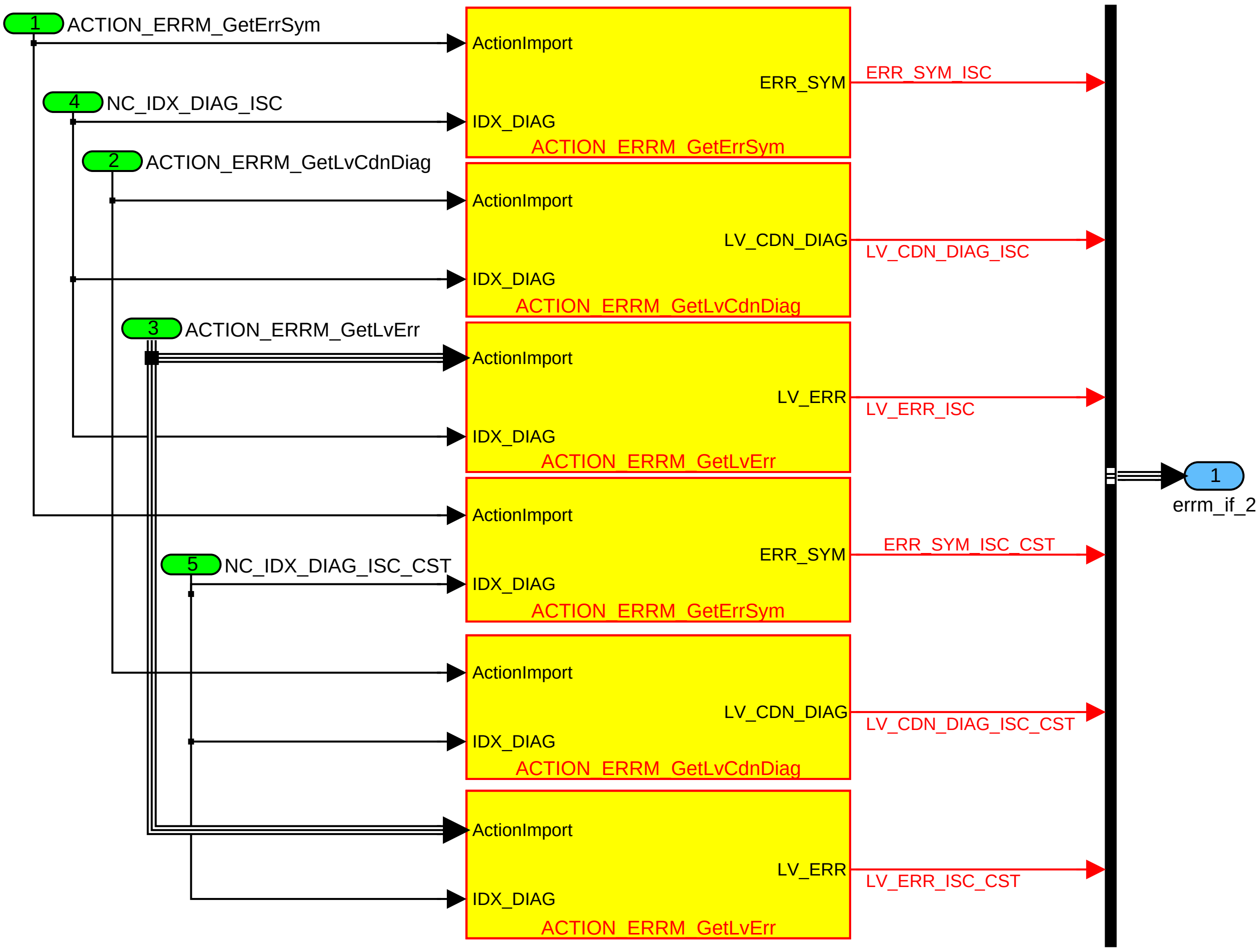


Figure 151 ENSC_MA09C/ OPM/ ERRM_IF_2

22.7.1.4 Visual mode output

Visual mode output is calculated here.

Calculation of LV_END_DIAG_ISC and LV_END_DIAG_ISC_CST using ERRM visual mode template

Calculation of LV_END_DIAG_ISC and LV_END_DIAG_ISC_CST is done here.

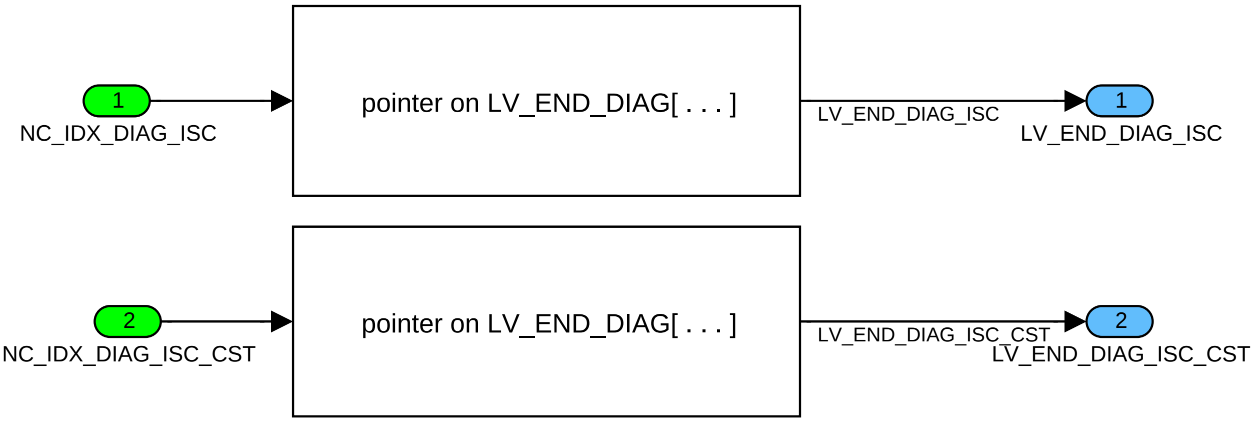
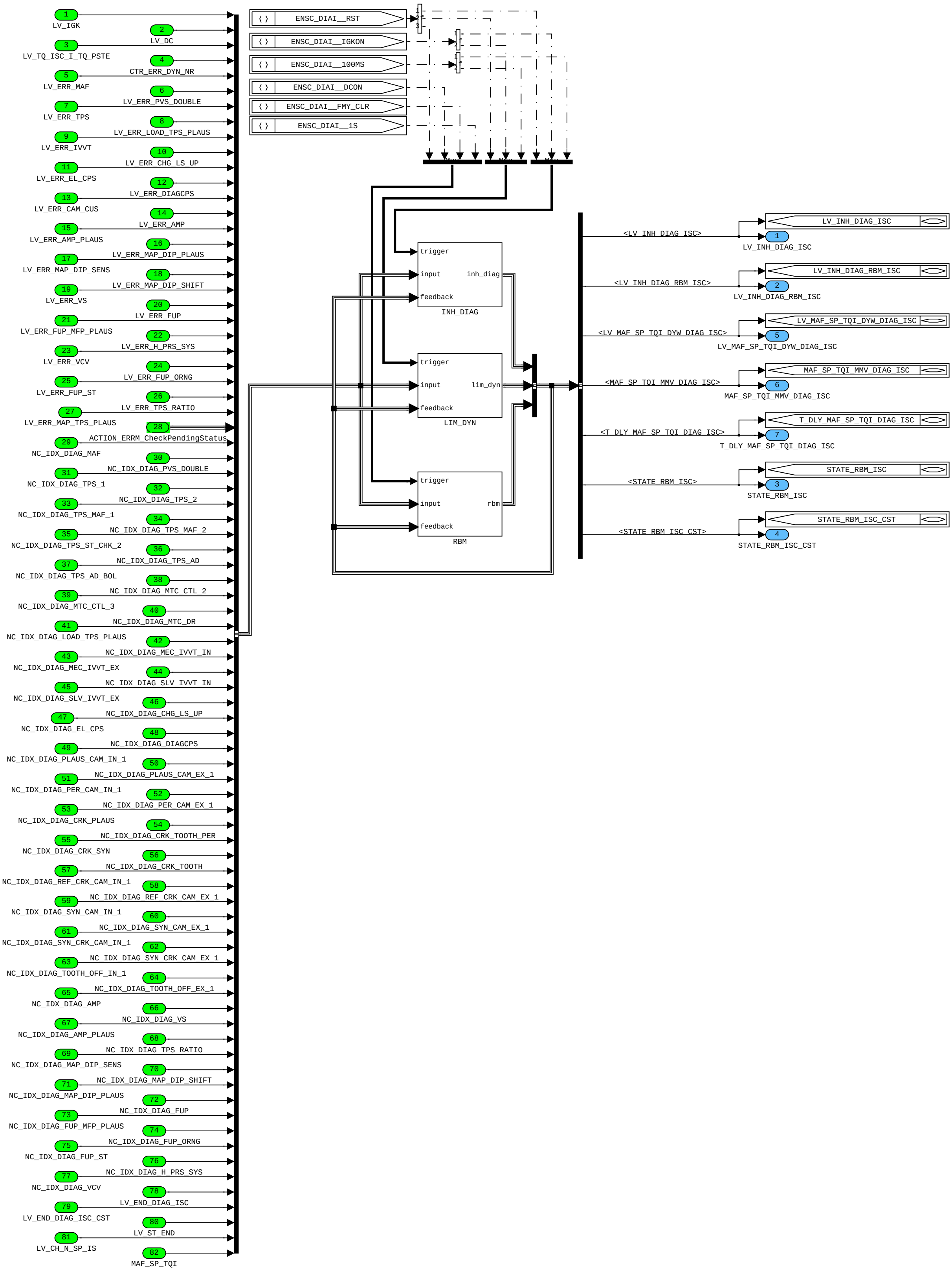


Figure 152 ENSC_MA09C/ ERRM_VIS_OUT/ CLC_VIS_OUT

22.8 Inhibition of idle speed control diagnosis (Appl. Inc.)

Overview



SDA_SRS / SDA 4.0 01−Feb−2006

Figure 153 ENSC_MA09D

Chapter	Baseline	Include File
Engine speed control	4DC3940S	43A09D01.00F

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Pages
3836 of 9643

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_ISC	O/V	0...1H	0...1	1	-
Flag for inhibitation of the Idle Speed Control Diagnosis					
LV_INH_DIAG_RBM_ISC	O/V	0...1H	0...1	1	-
Flag for inhibitation of the Idle Speed Control Diagnosis if there is a OBD failure					
STATE_RBM_ISC	O/V	0...7H	0...7	1	-
Interface of ISC monitor with the Rate-Based Monitoring statisticsBit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1)Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1)Bit 2: individual RBM conditio					
STATE_RBM_ISC_CST	O/V	0...7H	0...7	1	-
Interface of ISC_CST monitor with Rate-Based Monitoring statisticsBit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1)Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1)Bit 2: individual RBM conditio					
LV_MAF_SP_TQI_DYW_DIAG_ISC	O/V	0...1H	0...1	1	-
Indication for limited dynamic condition - functional check ISC					
MAF_SP_TQI_MMV_DIAG_ISC	O/V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Moving mean value of MAF_SP_TQI (RAM cell) - functional check ISC					
T_DLY_MAF_SP_TQI_DIAG_ISC	O/V	0...FFH	0...25.5	0.1	s
Actual deactivation time after drop of limited dynamic condition					

Input data:

LV_IGK	LV_DC	LV_TQ_ISC_I_TQ_PSTE	CTR_ERR_DYN_NR
LV_ERR_MAF	LV_ERR_PVS_DOUBLE	LV_ERR_TPS	LV_ERR_LOAD_TPS_PLAUS
LV_ERR_IVVT	LV_ERR_CHG_LS_UP	LV_ERR_EL_CPS	LV_ERR_DIAGCPS
LV_ERR_CAM_CUS	LV_ERR_AMP	LV_ERR_AMP_PLAUS	LV_ERR_MAP_DIP_PLAUS
LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_SHIFT	LV_ERR_VS	LV_ERR_FUP
LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV	LV_ERR_FUP_ORNG
LV_ERR_FUP_ST	LV_ERR_TPS_RATIO	LV_ERR_MAP_TPS_PLAUS	NC_IDX_DIAG_MAF
NC_IDX_DIAG_PVS_DOUBLE	NC_IDX_DIAG_TPS_1	NC_IDX_DIAG_TPS_2	NC_IDX_DIAG_TPS_MAF_1
NC_IDX_DIAG_TPS_MAF_2	NC_IDX_DIAG_TPS_ST_CHK_2	NC_IDX_DIAG_TPS_AD	NC_IDX_DIAG_TPS_AD_BOL
NC_IDX_DIAG_MTC_CTL_2	NC_IDX_DIAG_MTC_CTL_3	NC_IDX_DIAG_MTC_DR	NC_IDX_DIAG_LOAD_TPS_PLAUS
NC_IDX_DIAG_MEC_IVVT_IN[NC_NR_CBK_IVVT]	NC_IDX_DIAG_MEC_IVVT_EX[NC_NR_CBK_IVVT]	NC_IDX_DIAG_SLV_IVVT_IN[NC_NR_CBK_IVVT]	NC_IDX_DIAG_SLV_IVVT_EX[NC_NR_CBK_IVVT]
NC_IDX_DIAG_CHG_LS_UP	NC_IDX_DIAG_EL_CPS	NC_IDX_DIAG_DIAGCPS	NC_IDX_DIAG_PLAUS_CAM_IN_1
NC_IDX_DIAG_PLAUS_CAM_EX_1	NC_IDX_DIAG_PER_CAM_IN_1	NC_IDX_DIAG_PER_CAM_EX_1	NC_IDX_DIAG_CRK_PLAUS
NC_IDX_DIAG_CRK_TOOTH_PER	NC_IDX_DIAG_CRK_SYN	NC_IDX_DIAG_CRK_TOOTH	NC_IDX_DIAG_REF_CRK_CAM_IN_1
NC_IDX_DIAG_REF_CRK_CAM_EX_1	NC_IDX_DIAG_SYN_CAM_IN_1	NC_IDX_DIAG_SYN_CAM_EX_1	NC_IDX_DIAG_SYN_CRK_CAM_IN_1
NC_IDX_DIAG_SYN_CRK_CAM_EX_1	NC_IDX_DIAG_TOOTH_OFF_IN_1	NC_IDX_DIAG_TOOTH_OFF_EX_1	NC_IDX_DIAG_AMP
NC_IDX_DIAG_VS	NC_IDX_DIAG_AMP_PLAUS	NC_IDX_DIAG_TPS_RATIO	NC_IDX_DIAG_MAP_DIP_SENS
NC_IDX_DIAG_MAP_DIP_SHIFT	NC_IDX_DIAG_MAP_DIP_PLAUS	NC_IDX_DIAG_FUP	NC_IDX_DIAG_FUP_MFP_PLAUS
NC_IDX_DIAG_FUP_ORNG	NC_IDX_DIAG_FUP_ST	NC_IDX_DIAG_H_PRS_SYS	NC_IDX_DIAG_VCV
LV_END_DIAG_ISC	LV_END_DIAG_ISC_CST	LV_ST_END	LV_CH_N_SP_IS
MAF_SP_TQI			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_MAF_SP_TQI_DIAG_ISC	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Correlation constant for calculation of MAF_SP_TQI_MMV_DIAG_ISC					
C_CRLC_MAF_SP_TQI_DIAG_ISC_CST	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Correlation constant of MAF_SP_TQI_MMV_DIAG_ISC during coldstart monitoring					
C_MAF_SP_TQI_DYW_DIAG_ISC	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Dynamic window around MAF_SP_TQI_MMV_DIAG_ISC					
C_MAF_SP_TQI_DYW_DIAG_ISC_CST	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Dynamic window around MAF_SP_TQI_MMV_DIAG_ISC at coldstart					
C_T_DLY_MAF_SP_TQI_DIAG_ISC	1	0...FFH	0...25.5	0.1	s
Deactivation time after drop of limited dynamic					

Import actions:

ACTION_ERRM_CheckPendingStatus(IN <PRM_IDX_DIAG>, OUT <PRM_LV_ERR_PND>)
This API shall be used to verify if a failure stored in the dynamic memory has the pending status or not. When it impossible to determine if the fault is pending or not (failure not store because the dynamic memory is full), this failure should be considered as pending anyway.

22.8.1 Inhibition of diagnoses

Description:

If one of the following components fail, the ISC diagnoses function is stopped by LV_INH_DIAG_ISC = 1. The RBM of ISC diagnoses is stopped by

LV_INH_DIAG_RBM_ISC = 1.

Application Condition

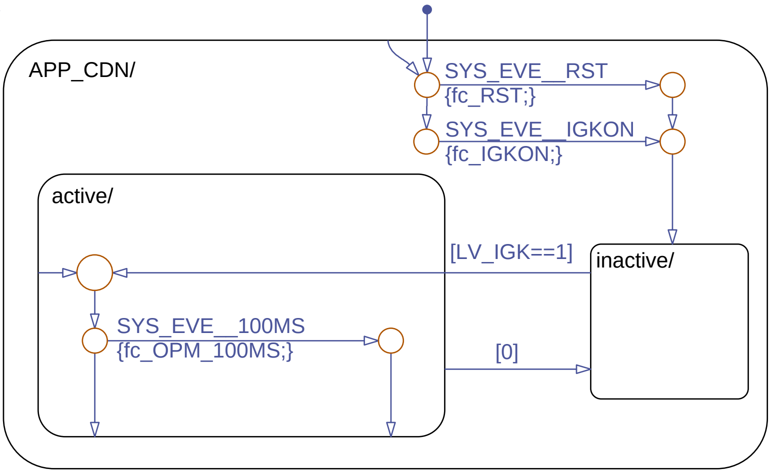


Figure 154 ENSC_MA09D/ INH_DIAG/ APP_CDN/ APPCND

Function Description

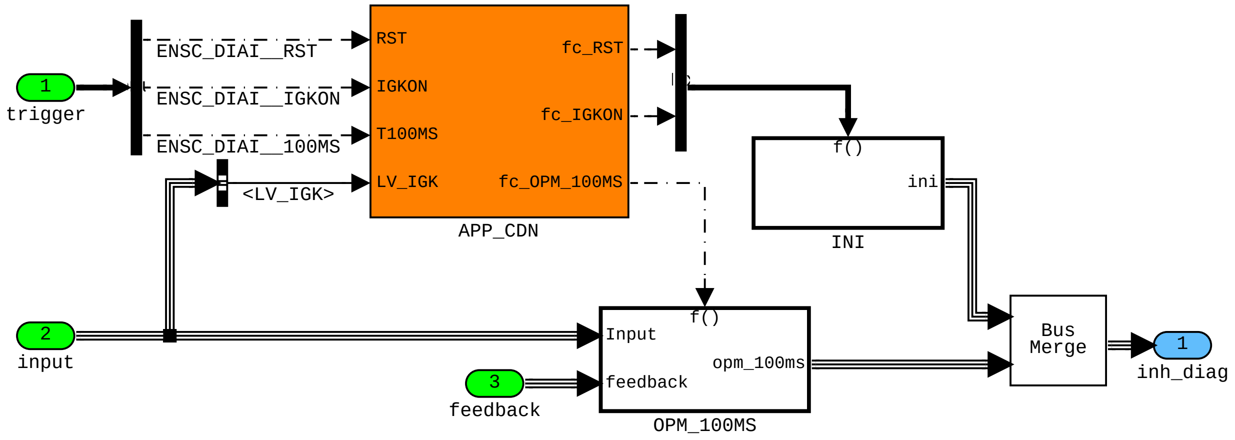


Figure 155 ENSC_MA09D/ INH_DIAG

22.8.1.1 Initialization at Reset and Ignition Key off to on

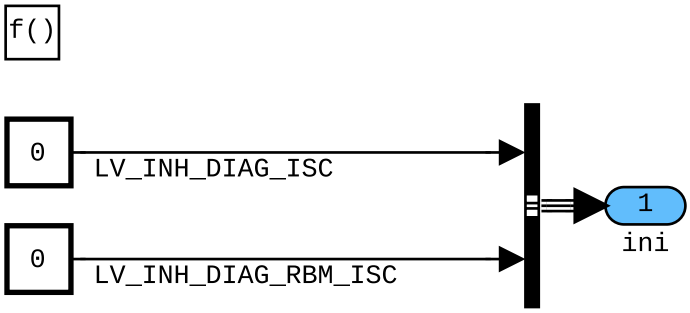


Figure 156 ENSC_MA09D/ INH_DIAG/ INI

22.8.1.2 Calculation at 100ms recurrence

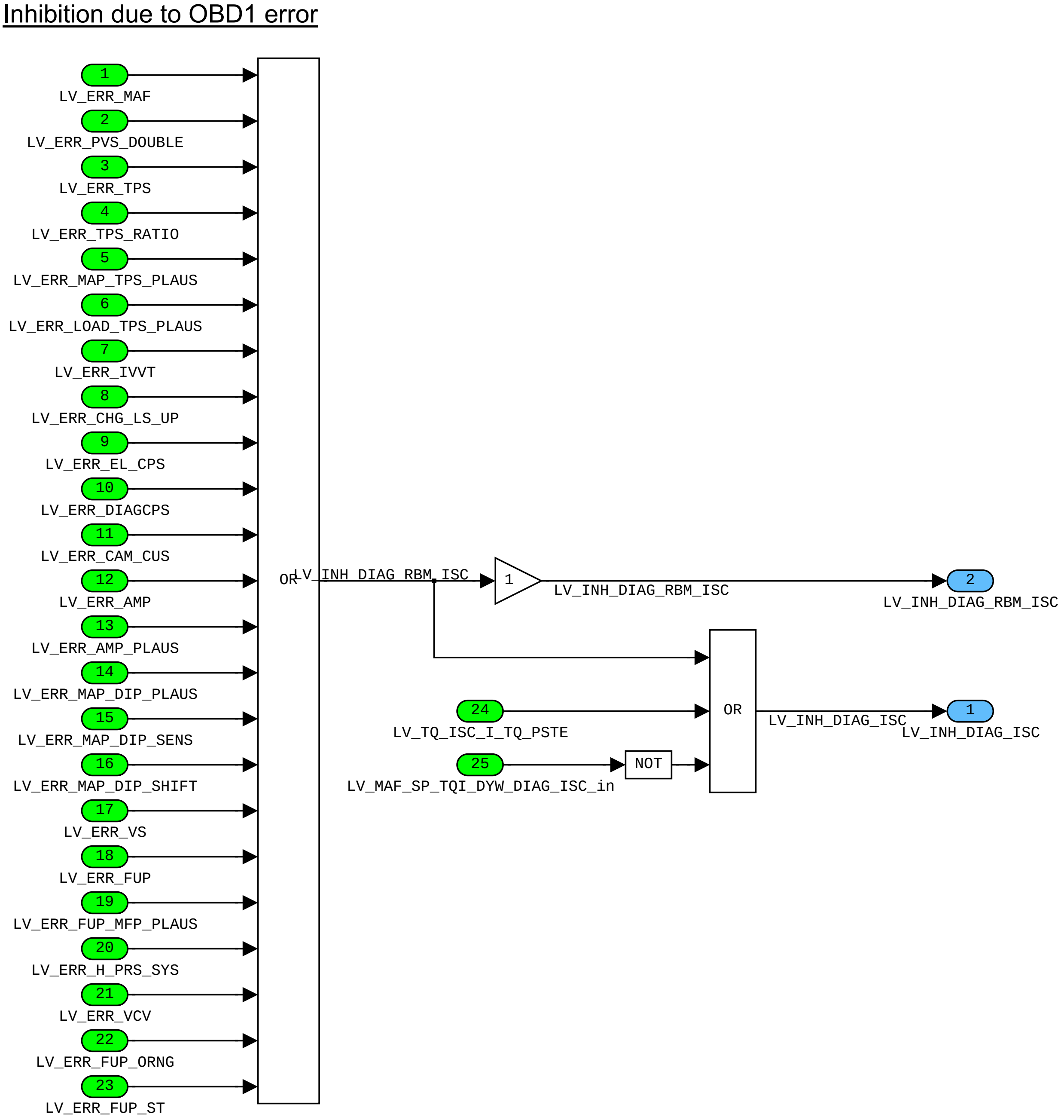


Figure 157 ENSC_MA09D/ INH_DIAG/ OPM_100MS/ CLC

22.8.2 Limited dynamic condition LV_MAF_SP_TQI_DYW_DIAG_ISC

Application Condition

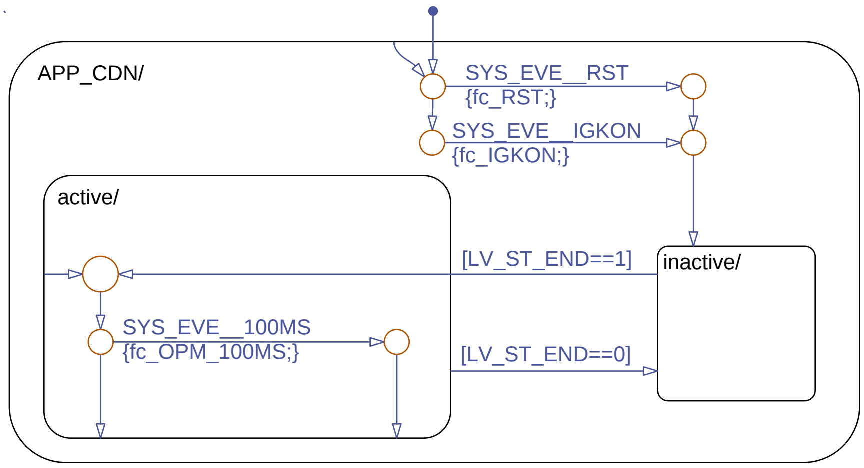


Figure 158 ENSC_MA09D/ LIM_DYN/ APP_CDN/ APPCND

Function Description

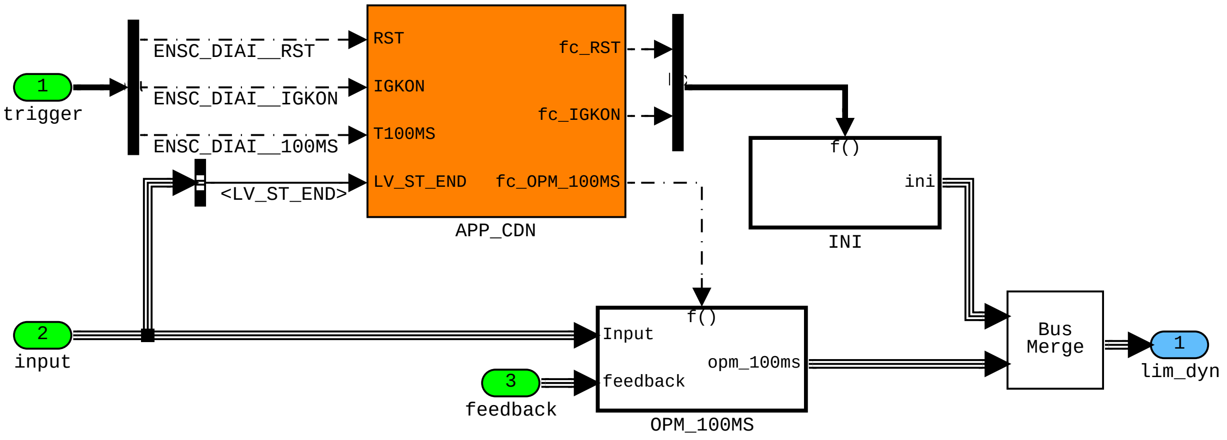


Figure 159 ENSC_MA09D/ LIM_DYN

22.8.2.1 Initialization at Reset and Ignition Key off to on

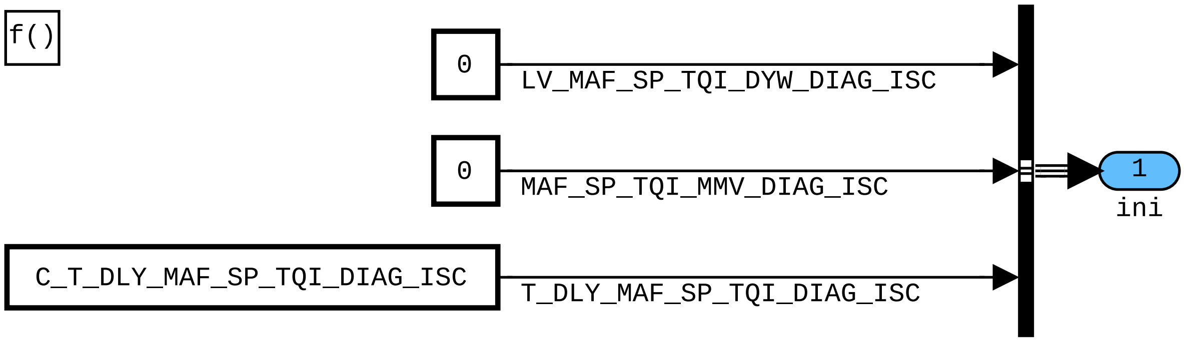


Figure 160 ENSC_MA09D/ LIM_DYN/ INI

22.8.2.2 Calculation at 100ms recurrence

Calculation of MAF_SP_TQI_MMV_DIAG_ISC:

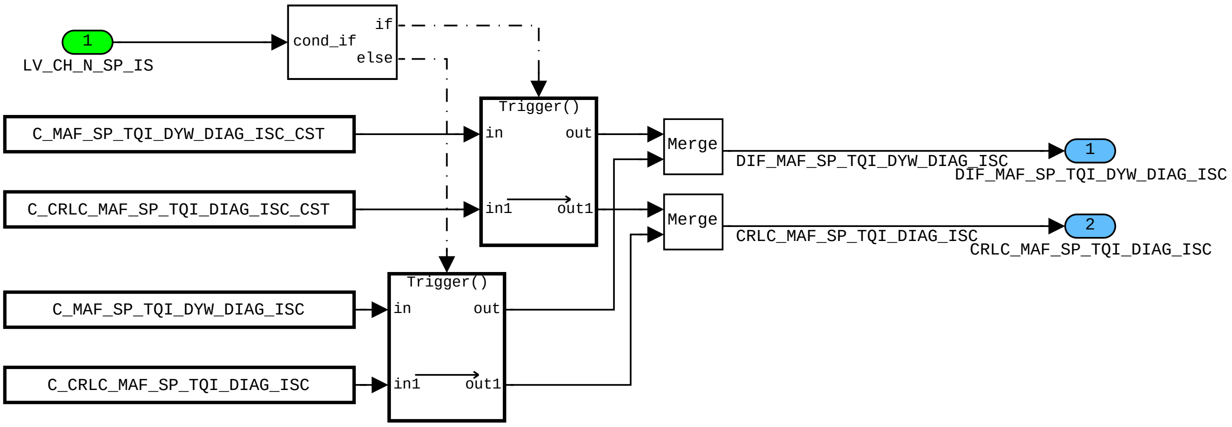


Figure 161 ENSC_MA09D/ LIM_DYN/ OPM_100MS/ CLC_1

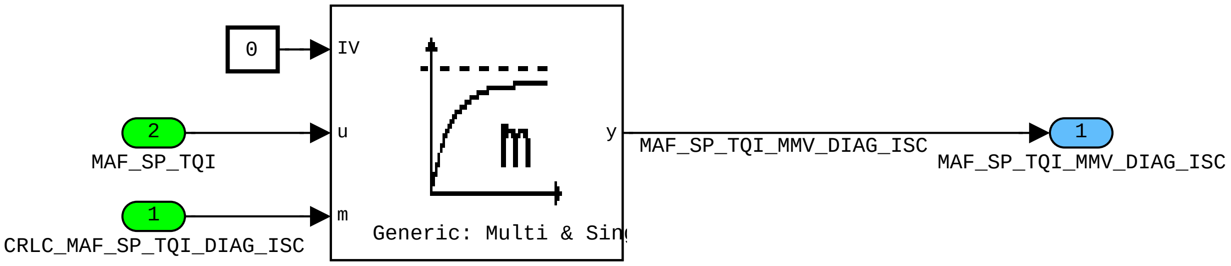


Figure 162 ENSC_MA09D/ LIM_DYN/ OPM_100MS/ CLC_2

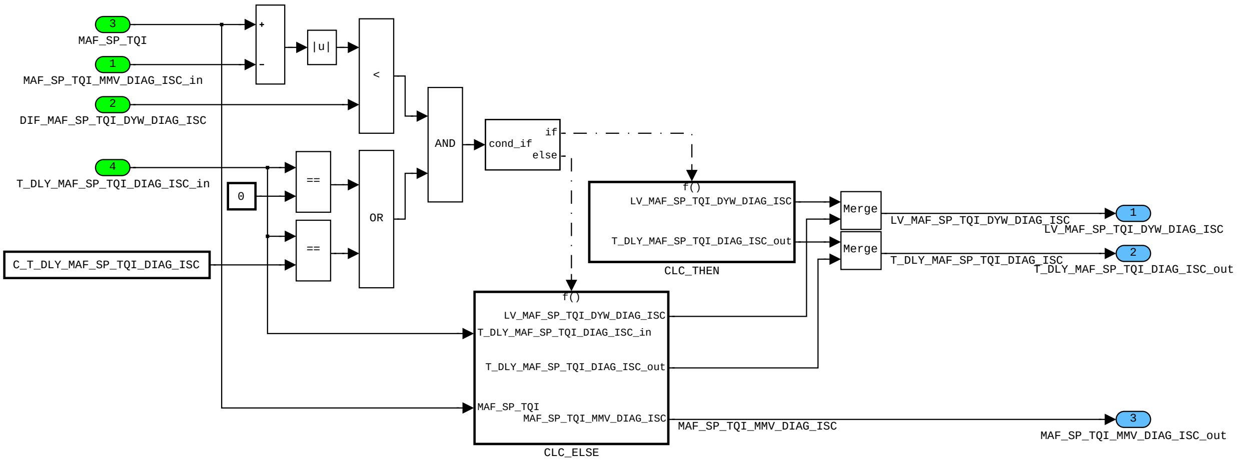


Figure 163 ENSC_MA09D/ LIM_DYN/ OPM_100MS/ CLC_3

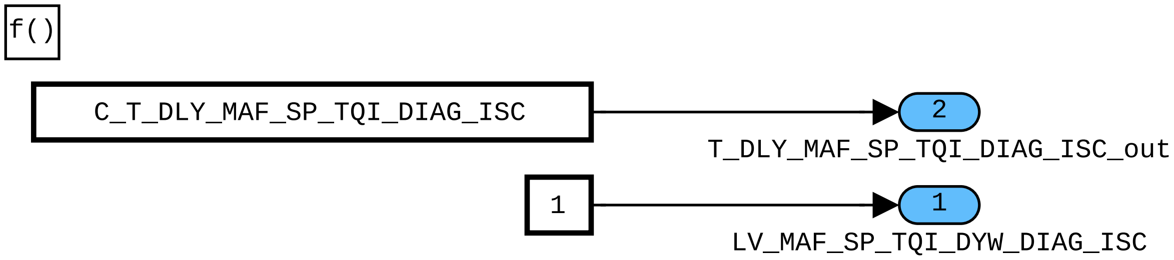


Figure 164 ENSC_MA09D/ LIM_DYN/ OPM_100MS/ CLC_3/ CLC_THEN

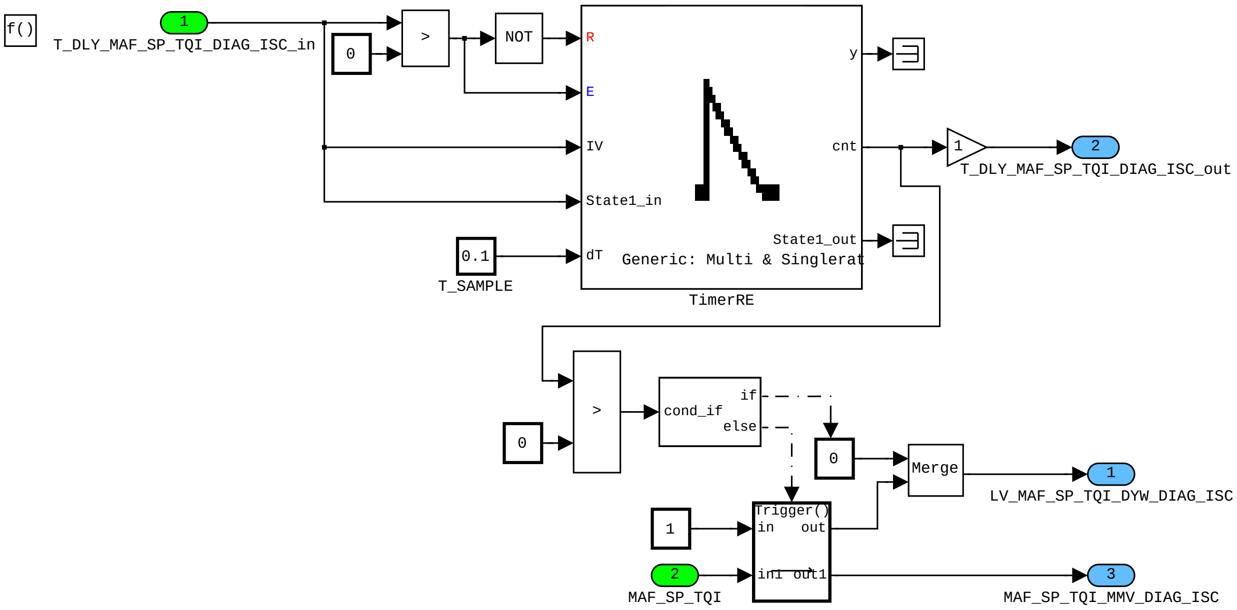


Figure 165 ENSC_MA09D/ LIM_DYN/ OPM_100MS/ CLC_3/ CLC_ELSE

22.8.3 Interface for Rate – Based Monitoring

General Information:

With this module the interface between the ISC / ISC_CST monitor and the rate based monitoring statistics is defined with STATE_RBM_ISC / STATE_RBM_ISC_CST data. Within STATE_RBM_ISC / ISC_CST, three different information are defined:

Conditions for monitoring are met long enough to detect manipulation (bit 0) (no intrusive operation, no short trip)

Monitor disabled because of system malfunction (bit 1) (depending on failure status: pending)

Monitor individual RBM conditions encountered within this DC(bit 2) (not valid for ISC daignosis)

Application Condition

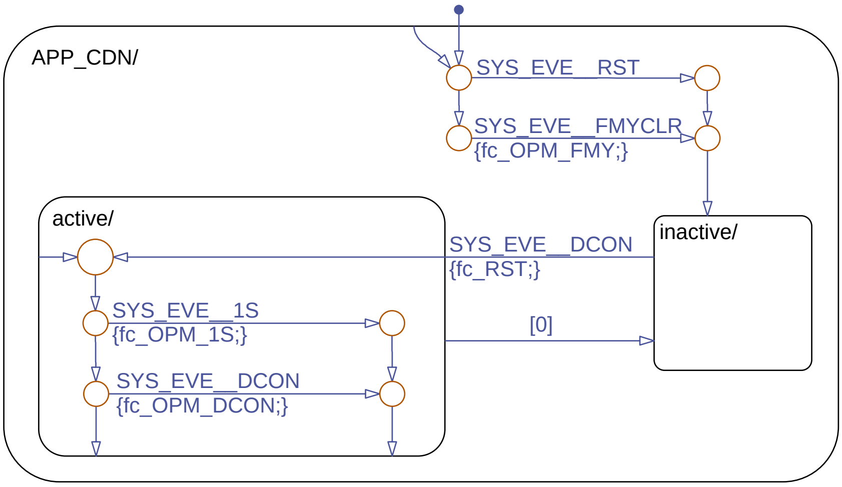


Figure 166 ENSC_MA09D/ RBM/ APP_CDN/ APPCND

Function Description

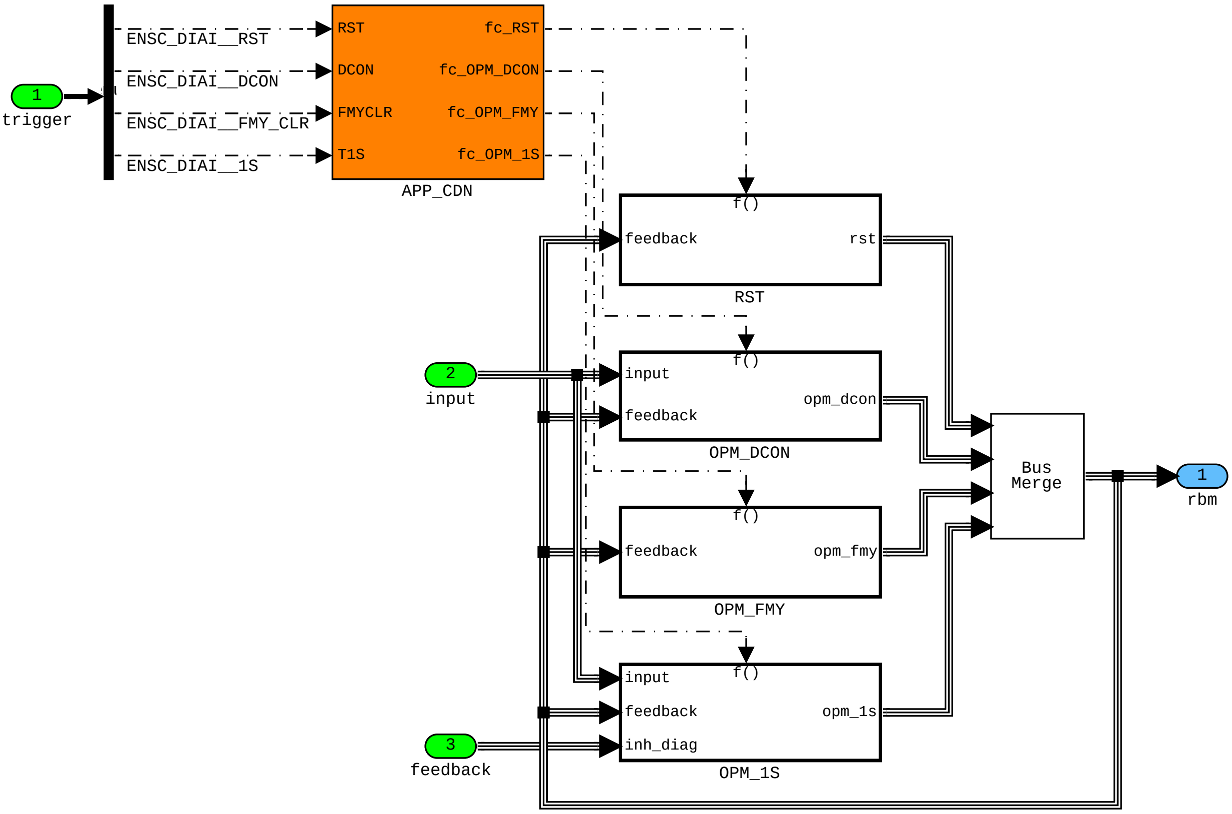


Figure 167 ENSC_MA09D/ RBM

22.8.3.1 Initialization at Reset

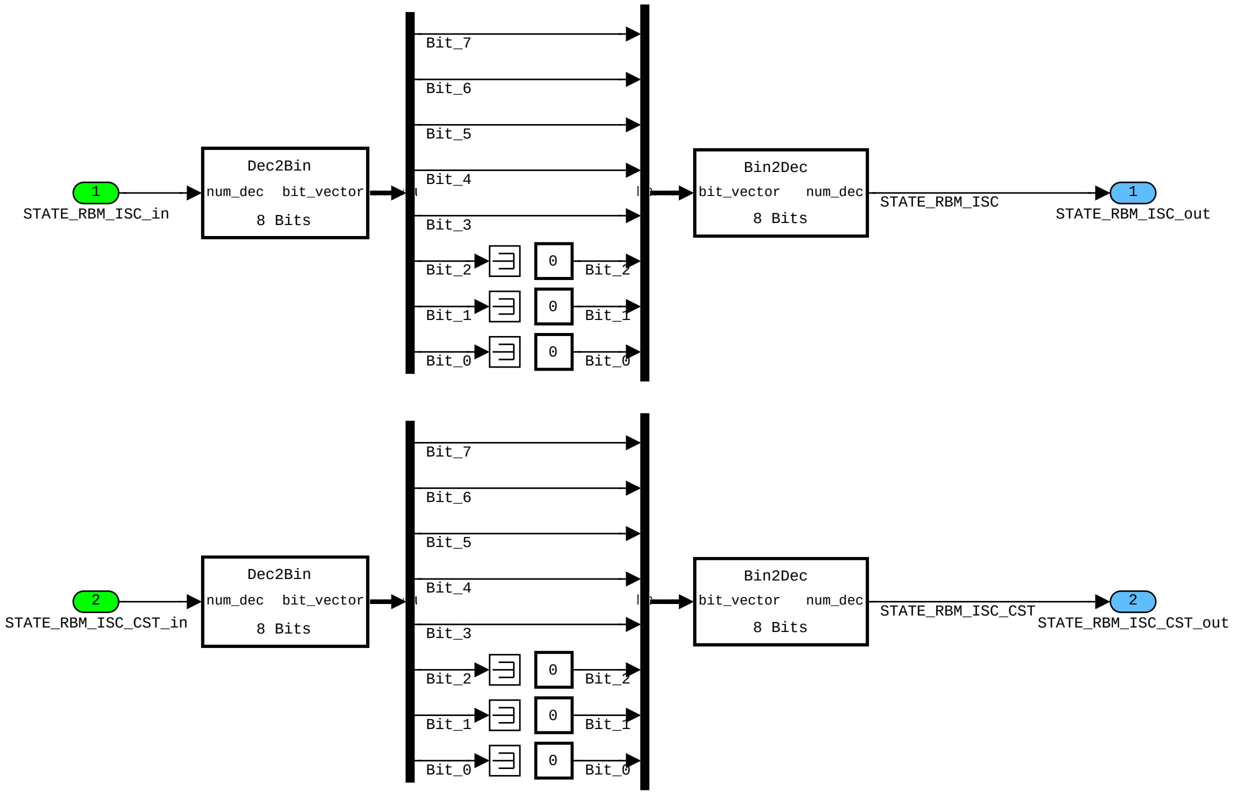


Figure 168 ENSC_MA09D/ RBM/ RST/ RESET_BIT_0_1_2

22.8.3.2 At LV_DC 0 to 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_MAF	LV_ERR_PVS_DOUBLE	LV_ERR_TPS_1	LV_ERR_TPS_2
LV_ERR_TPS_MAF_1	LV_ERR_TPS_MAF_2	LV_ERR_TPS_ST_CHK_2	LV_ERR_TPS_AD
LV_ERR_TPS_AD_BOL	LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3	LV_ERR_MTC_DR
LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MEC_IVVT_IN	LV_ERR_MEC_IVVT_EX	LV_ERR_SLV_IVVT_IN
LV_ERR_SLV_IVVT_EX	LV_ERR_CHG_LS_UP	LV_ERR_EL_CPS	LV_ERR_DIAGCPS
LV_ERR_PLAUS_CAM_IN_1	LV_ERR_PLAUS_CAM_EX_1	LV_ERR_PER_CAM_IN_1	LV_ERR_PER_CAM_EX_1
LV_ERR_CRK_PLAUS	LV_ERR_CRK_TOOTH_PER	LV_ERR_CRK_SYN	LV_ERR_CRK_TOOTH
LV_ERR_REF_CRK_CAM_IN_1	LV_ERR_REF_CRK_CAM_EX_1	LV_ERR_SYN_CAM_IN_1	LV_ERR_SYN_CAM_EX_1
LV_ERR_SYN_CRK_CAM_IN_1	LV_ERR_SYN_CRK_CAM_EX_1	LV_ERR_TOOTH_OFF_IN_1	LV_ERR_TOOTH_OFF_EX_1
LV_ERR_AMP	LV_ERR_VS	LV_ERR_AMP_PLAUS	LV_ERR_TPS_RATIO
LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_SHIFT	LV_ERR_MAP_DIP_PLAUS	LV_ERR_FUP
LV_ERR_FUP_MFP_PLAUS	LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_H_PRS_SYS
LV_ERR_VCV	LV_ERR_MAP_TPS_PLAUS		

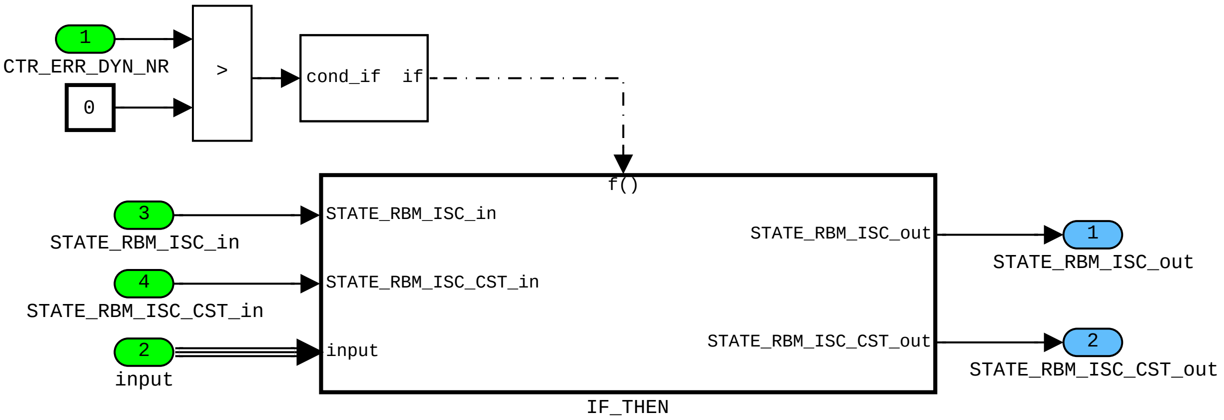


Figure 169 ENSC_MA09D/ RBM/ OPM_DCON/ CLC

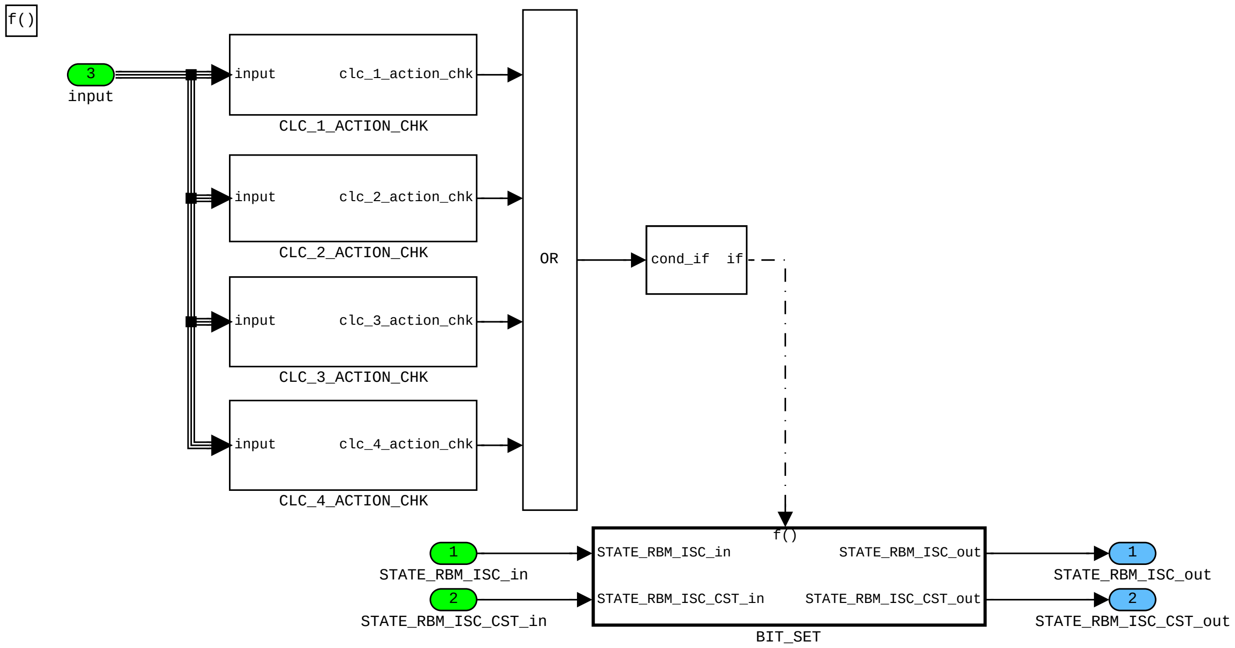


Figure 170 ENSC_MA09D/ RBM/ OPM_DCON/ CLC/ IF_THEN

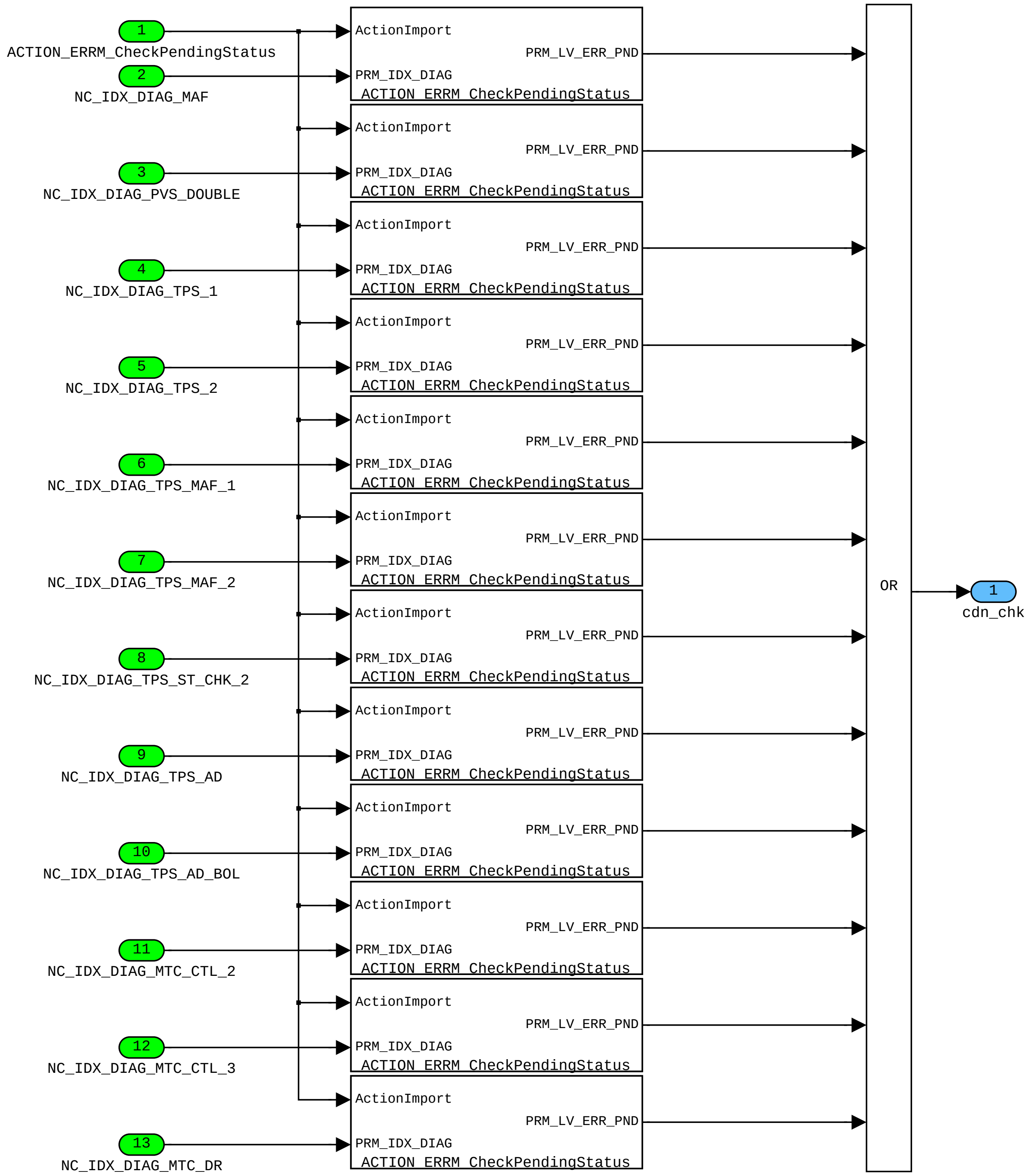


Figure 171 ENSC_MA09D/ RBM/ OPM_DCON/ CLC/ IF_THEN/ CLC_1_ACTION_CHK/ CDN_CHK

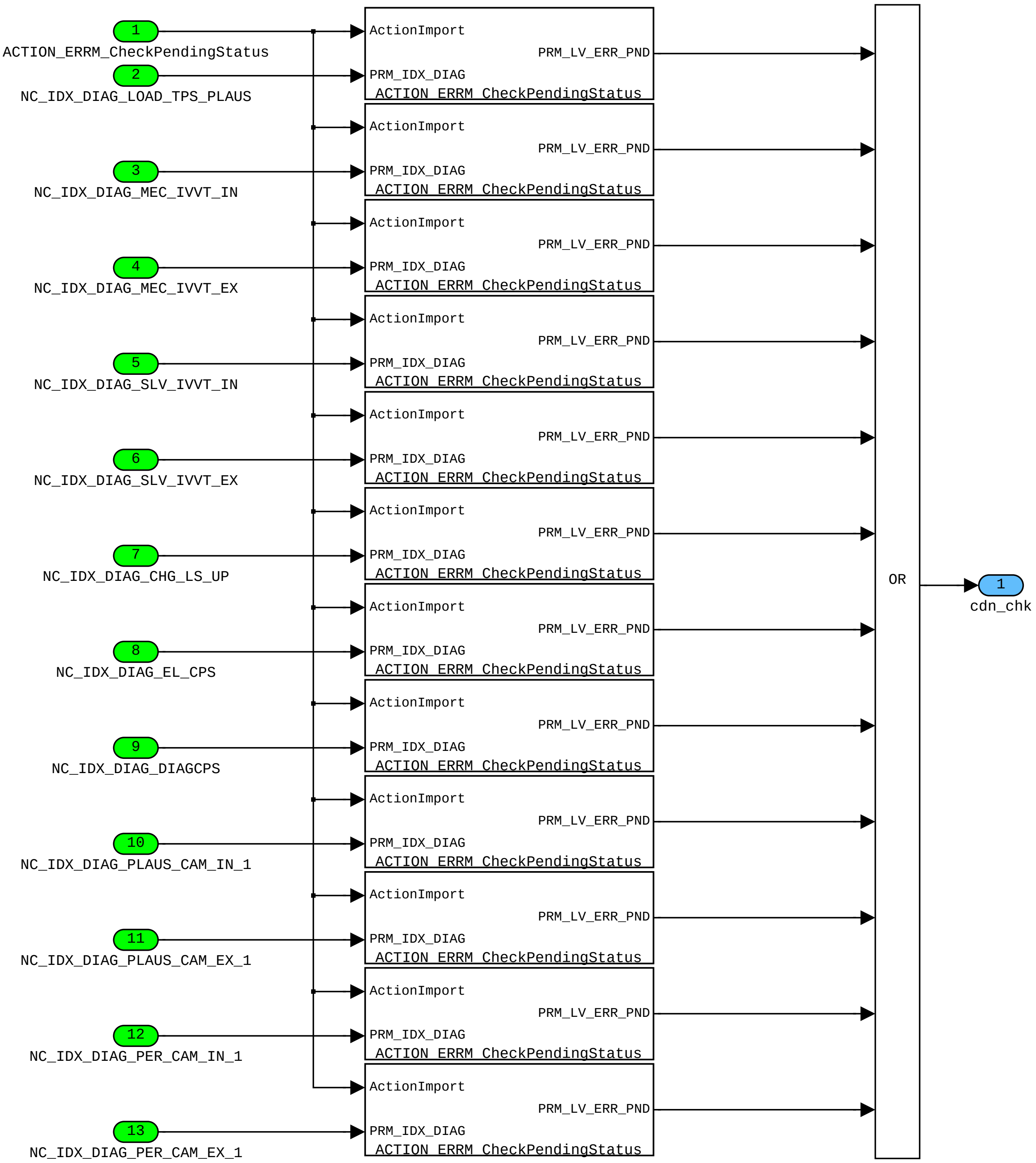


Figure 172 ENSC_MA09D/ RBM/ OPM_DCON/ CLC/ IF_THEN/ CLC_2_ACTION_CHK/ CDN_CHK

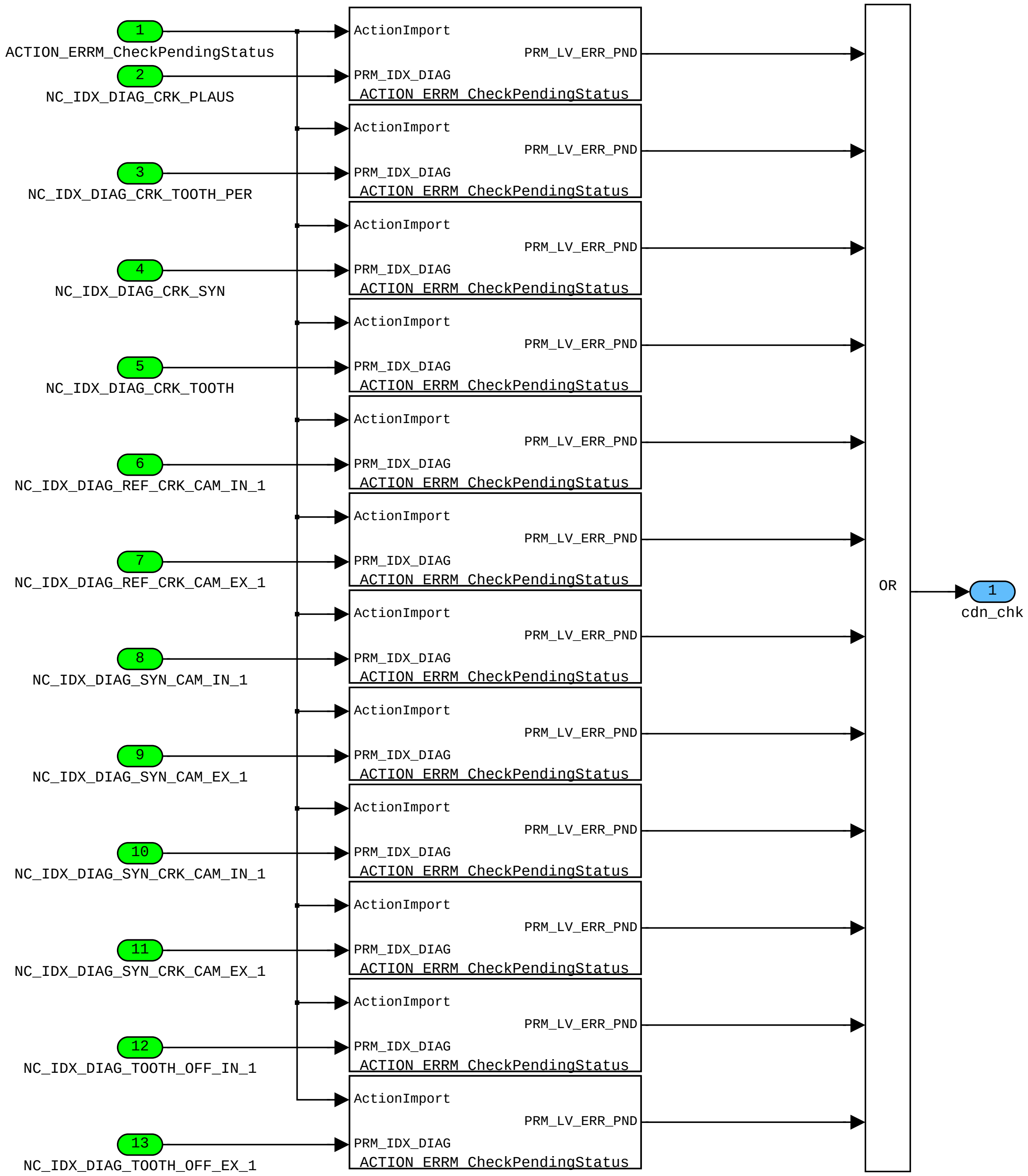


Figure 173 ENSC_MA09D/ RBM/ OPM_DCON/ CLC/ IF_THEN/ CLC_3_ACTION_CHK/ CDN_CHK

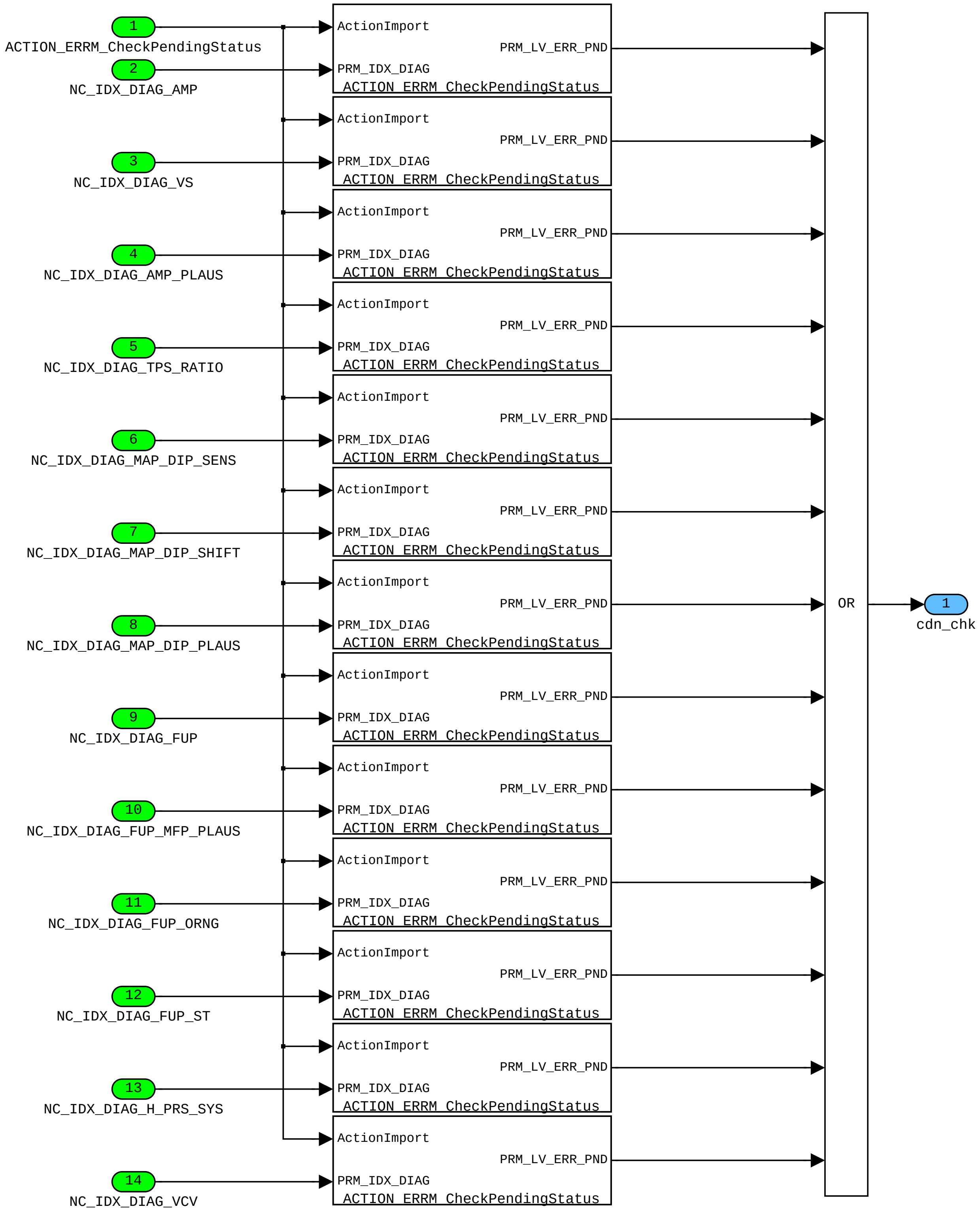


Figure 174 ENSC_MA09D/ RBM/ OPM_DCON/ CLC/ IF_THEN/ CLC_4_ACTION_CHK/ CDN_CHK

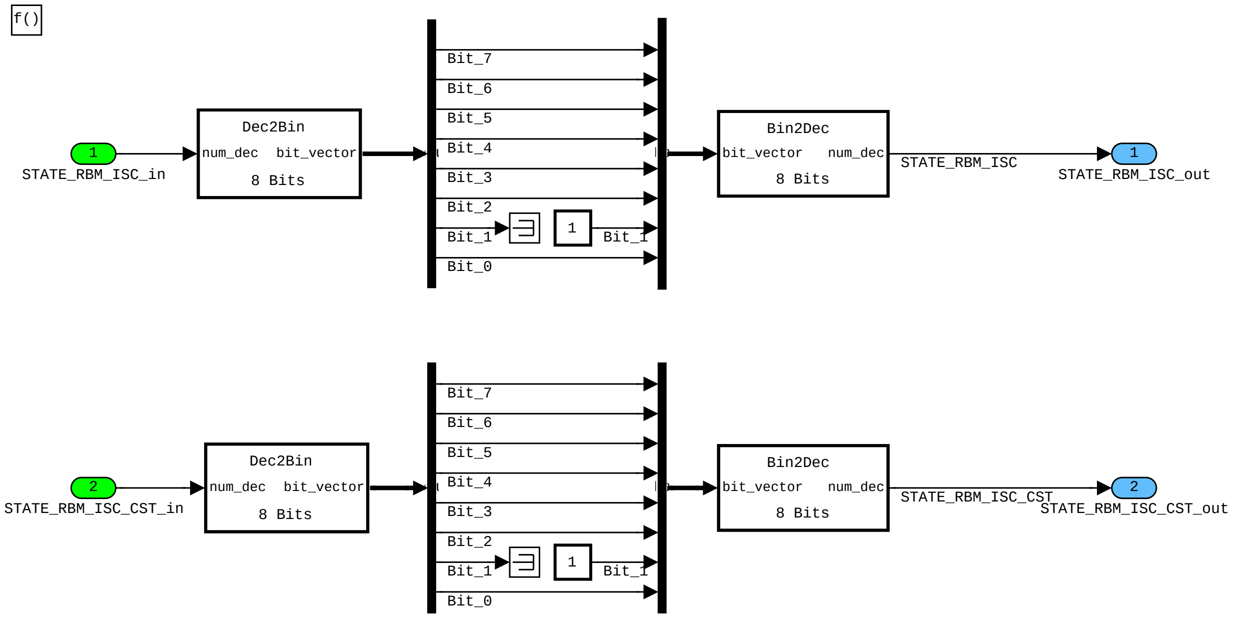


Figure 175 ENSC_MA09D/ RBM/ OPM_DCON/ CLC/ IF_THEN/ BIT_SET

22.8.3.3 Initialization at LV_DC 0 to 1 transition

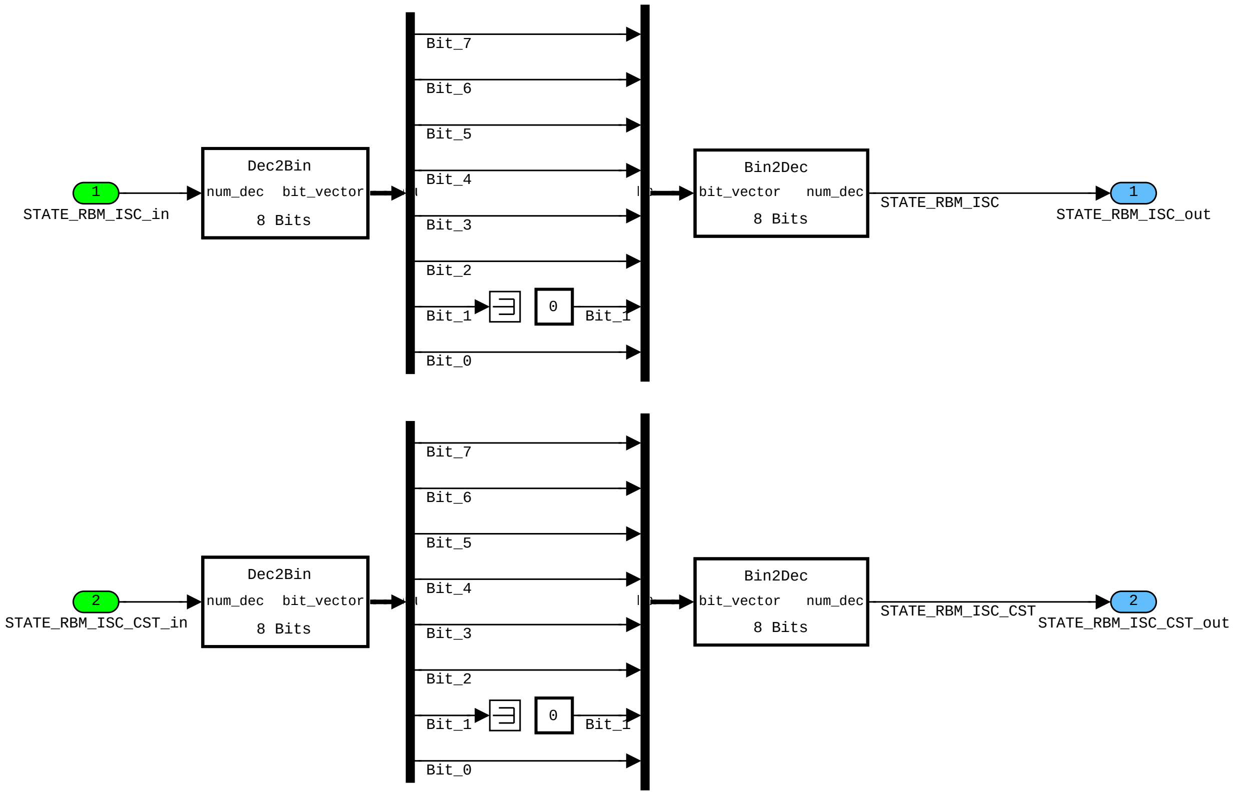


Figure 176 ENSC_MA09D/ RBM/ OPM_FMY/ RESET_BIT_1

22.8.3.4 Calculation at 1s recurrence

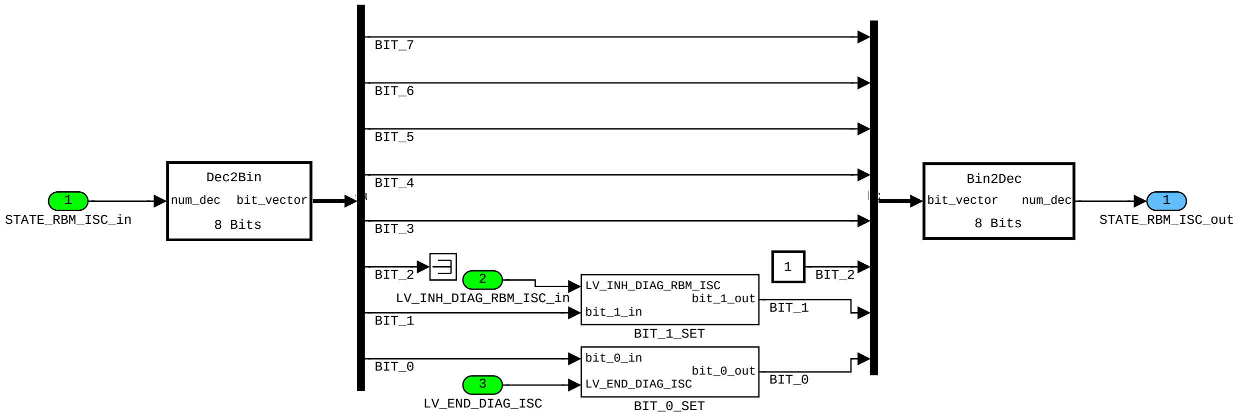


Figure 177 ENSC_MA09D/ RBM/ OPM_1S/ CLC_1

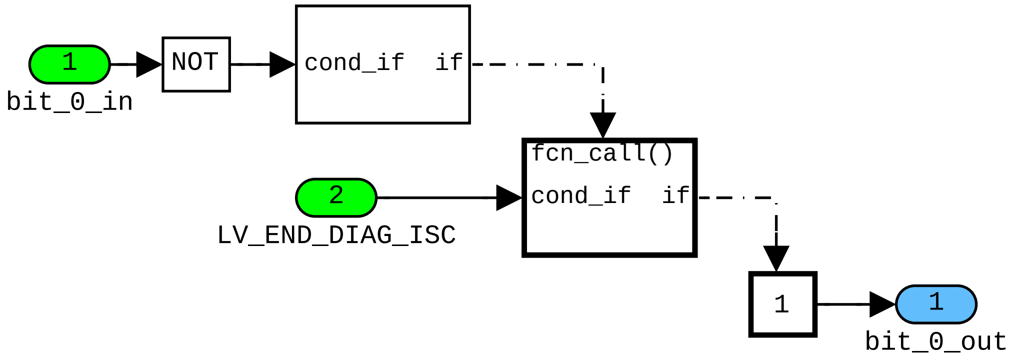


Figure 178 ENSC_MA09D/ RBM/ OPM_1S/ CLC_1/ BIT_0_SET

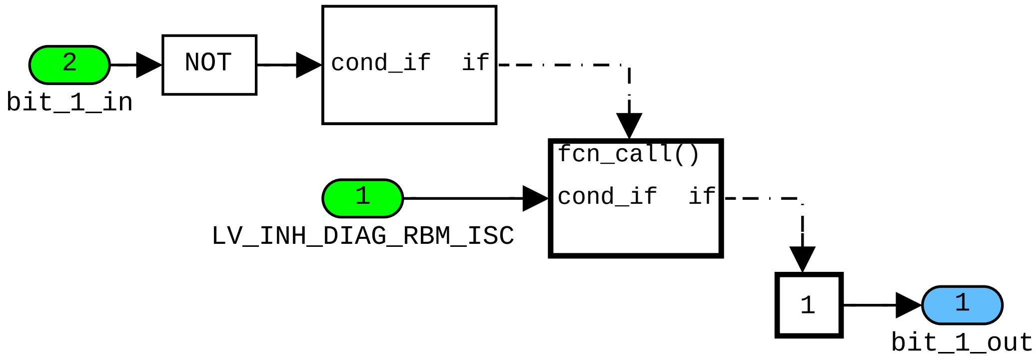


Figure 179 ENSC_MA09D/ RBM/ OPM_1S/ CLC_1/ BIT_1_SET

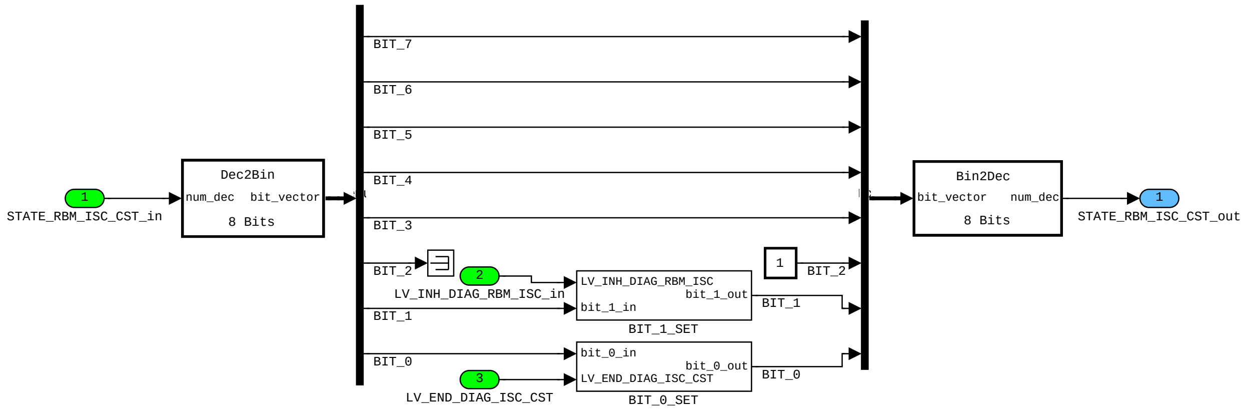


Figure 180 ENSC_MA09D/ RBM/ OPM_1S/ CLC_2

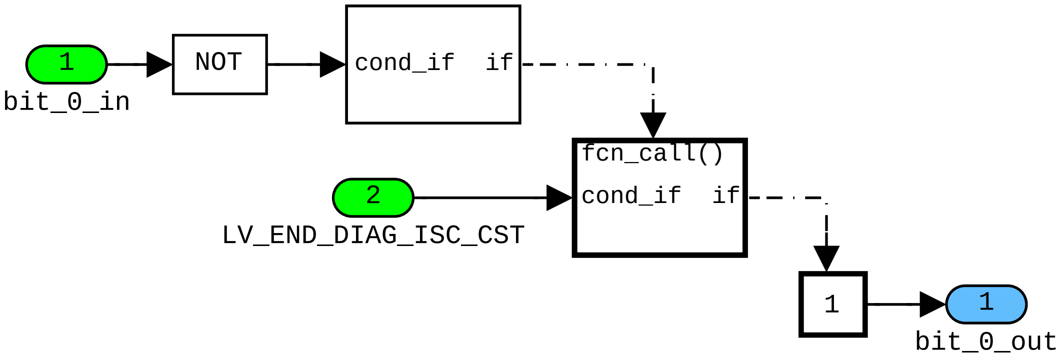


Figure 181 ENSC_MA09D/ RBM/ OPM_1S/ CLC_2/ BIT_0_SET

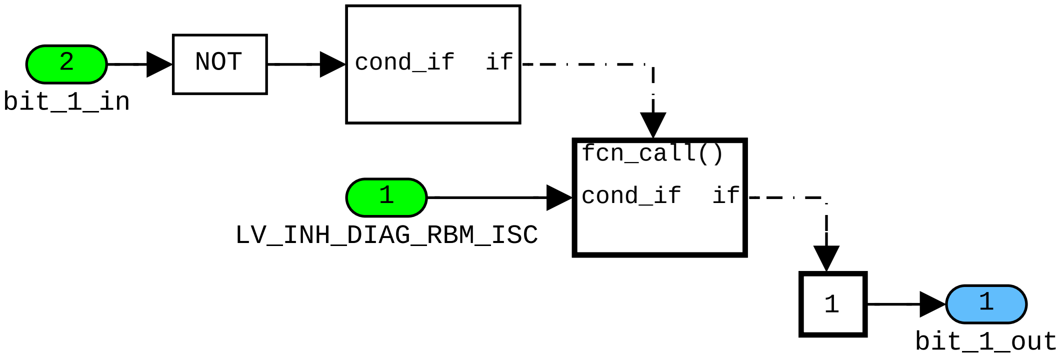


Figure 182 ENSC_MA09D/ RBM/ OPM_1S/ CLC_2/ BIT_1_SET

22.9 Customer adaptation module: AGGR ENSC

22.9.1 Outputs for BMW which are defined as ENSC exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Mdi_llri	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
idle speed controller, I-part					
Mdi_llrad	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
Idle speed controller, adaptation					
Mdi_llrp	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
idle speed controller, P/D-part slow					
Mdi_llrzw	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
idle speed controller, P/D-part fast					
Mdi_res_llmn	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
torque reserve at idle, lower limit					
Mdi_res_llmx	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
torque reserve at idle, limitation for stabilizing the ISC at high torque reserves					
Nstat	V/O	0...7FFFH	0...32767	1	[rpm]
Idle speed setpoint without guidance					
B_llr_on	V/O	0...1H	0...1	1	[-]
Logical variable for idle speed controller activatIon request					
Md_anfhi	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Moment Anfahrhilfe					
Md_res_anfhi	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Moment Anfahrhilfe					

Input data:

LV_REQ_ISC	N_SP_IS	TQ_DIF_P_D_FAST_IS	TQ_ADD_IS_BOL
TQ_DIF_ADD_IS_TOL	TQ_DIF_I_IS	ECU_STATE	TQ_DIF_P_D_SLOW_IS
TQ_DROF_FAST	TQ_DROF_SLOW	TQ_ADD_PL_DROF	TQ_DIF_IS_AD

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: The really possible physical limits of the outputs are different to the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit power latch phase: 0

Recurrences: Mdi_llri, Nstat : 20 ms

all others : 10 ms

Activation: every engine state

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
B_llr_on            =  LV_REQ_ISC

If ECU_STATE = "PWL"
Then
    Mdi_llrad      = 0
    Mdi_llri       = 0
    Mdi_llrp       = 0
    Mdi_llrzw      = 0
    Mdi_res_llmn   = 0
    Mdi_res_llmx   = 0
    Nstat          = 0
    Md_anfhi       = 0
    Md_res_anfhi   = 0
Elseif
    Mdi_llrad      = TQ_DIF_IS_AD
    Mdi_llri       =  TQ_DIF_I_IS
    Mdi_llrp       =  TQ_DIF_P_D_SLOW_IS
    Mdi_llrzw      =  TQ_DIF_P_D_FAST_IS
    Mdi_res_llmn   =  TQ_ADD_IS_BOL
    Mdi_res_llmx   =  TQ_DIF_ADD_IS_TOL
    Nstat          =  N_SP_IS
    Md_anfhi       = TQ_DROF_FAST
    Md_res_anfhi   = (TQ_DROF_SLOW - TQ_DROF_FAST) + TQ_ADD_PL_DROF
Endif
```

22.9.2 Outputs for BMW which are not defined as ENSC exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Nkw_soll	V/O	0...7FFFH	0...32767	1	[rpm]
Solldrehzahl					
Nsllm	V/O	0...7FFFH	0...32767	1	[rpm]
Idle speed setpoint without drivetrain engaged (LV_DRI = 0)					

Input data:

N_SP_IS_2	N_SP_IS		ECU_STATE

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation at reset or at exit power latch phase: 0

Recurrence: 10 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
Nsllm          =  N_SP_IS_2
If    ECU_STATE = "PWL"
then
      Nkw_soll = 0
else
      Nkw_soll   = N_SP_IS
endif
```

22.9.3 Outputs for SV Aggregates

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_SP_IS_BRAKE	V/O	0...7FFFH	0...32767	1	[rpm]
idle speed correction due to accumulator charge					
N_SP_IS_POIL_CTL	V/O	0...1FE0H	0...8160	1	[rpm]
idle speed correction due to oil pressure regulator					
N_SP_IS_POW_1	V/O	0...7FFFH	0...32767	1	[rpm]
idle speed correction due to accumulator charge					
FAC_TQ_ADD_IS_OPM_SEL	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Torque reserve for idle speed interpolation factor for operation switch manager					

Input data:

Nsl_koor	Nslb	F_llr_ba	Nkw_poel_soll

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: all outputs = 0 at Reset

Recurrence: N_SP_IS_POW_1 is updated every 1000 ms
N_SP_IS_BRAKE is updated every 100 ms
FAC_TQ_ADD_IS_OPM_SEL, N_SP_IS_POIL_CTL is updated every 10 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
N_SP_IS_BRAKE              =  Nsl_koor
```

N_SP_IS_POW_1 = Nslb
FAC_TQ_ADD_IS_OPM_SEL = F_llr_ba
N_SP_IS_POIL_CTL = Nkw_poel_soll

23 Engine position and speed determination

Chapter	Baseline		
Engine position and speed determination	4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

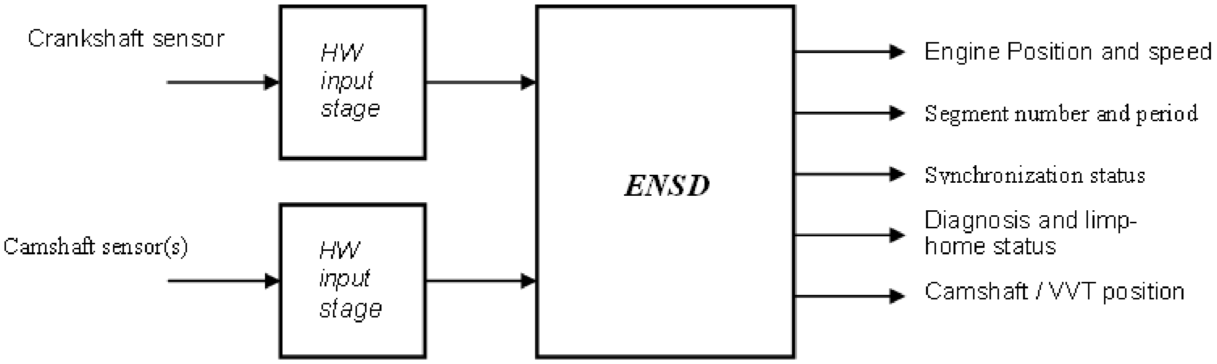
Pages
3859 of 9643

Table of Contents

Index

23.1 ENSD General

23.1.1 General Description



Up to 4 camshaft sensors are supported.

The ENSD aggregate calculates all output data which depend on crankshaft and camshaft sensor information. This includes:

- engine position
- engine speed
- engine speed gradient
- segment number and period
- synchronization status
- diagnosis and limp-home flags
- camshaft / VVT position feedback
- min./max. engine position for pre-injection

At engine start, three synchronization modes are activated:

1) Crankshaft (self-)synchronization:

   The purpose of this synchronization mode is to identify the crankshaft position (0...360° CRK). The crankshaft position is clearly identified at the reference gap of the crankshaft target wheel.

2) Camshaft (self-)synchronization:

   The purpose of this synchronization mode is to identify the camshaft position (0..720° CRK). The camshaft position is clearly identified as soon as an unambigous camshaft edge pattern is found.

3) Camshaft/crankshaft synchronization:

   The purpose of this synchronization mode is to identify the engine position (0..720° CRK). The engine position is clearly identified as soon as an unambigous crankshaft to camshaft position is found.

Injection / Ignition can be enabled as soon as

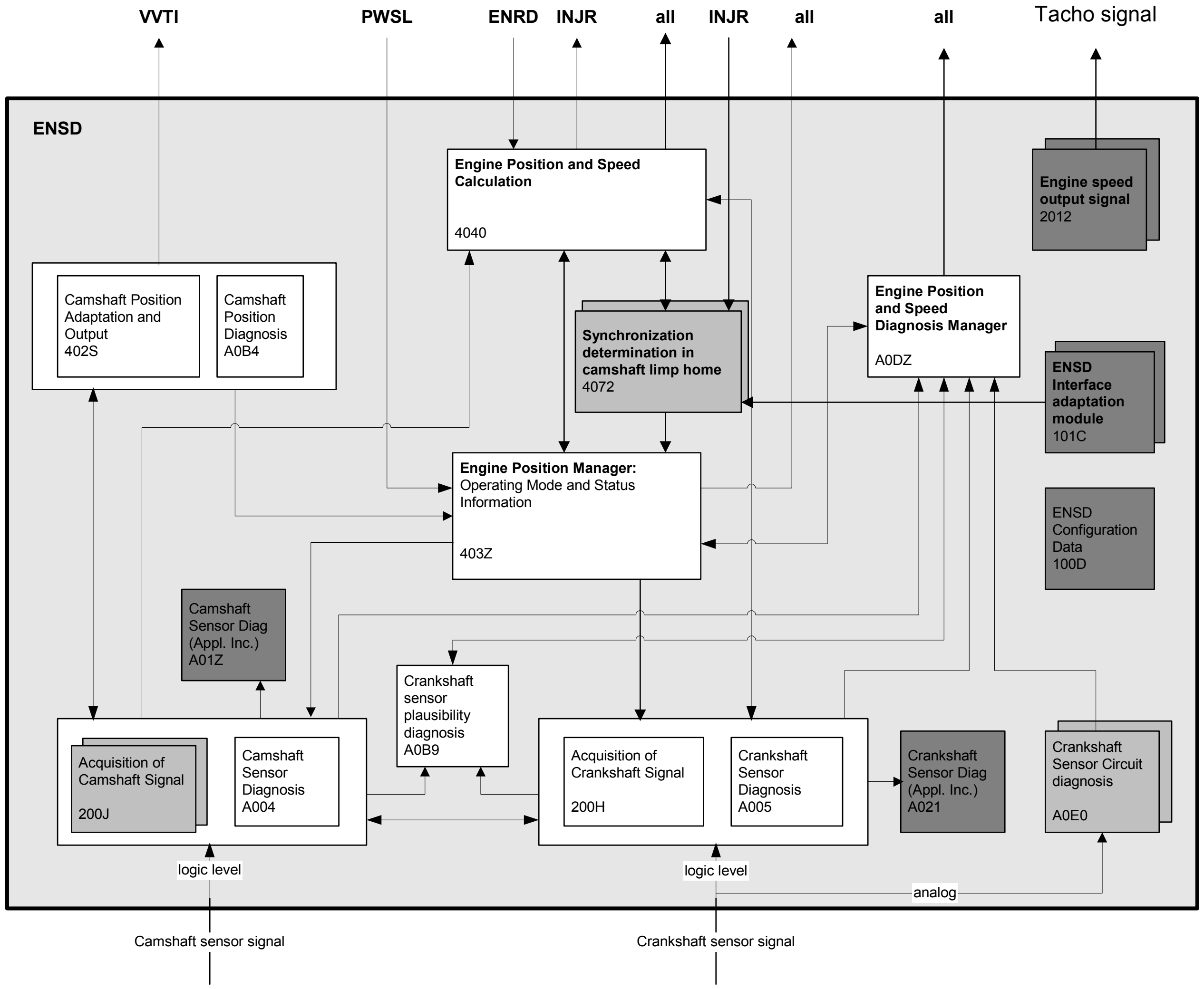
- engine position is identified (MPI engines)

- engine position is validated (DI / DS engines)

A corresponding status information (LV_SYN_ENG / LV_SYN_VLD) is produced and sent to the other aggregates.

23.1.2 Overview

The following figure shows the functional breakdown of the ENSD aggregate:



The light shaded blocs represent modules which are choosen in function of the actual system configuration (Hook modules).

The dark shaded blocs represent modules which have to be modified by the project („templates“ – Hook modules)

23.1.3 Short description of the containing functions

200H: Acquisition of crankshaft signal

Crankshaft synchronization, reference gap detection and crankshaft tooth validation. Measurements of tooth periods and segment periods

200J: Acquisition of camshaft signal

cam/crk synchronization (engine position offset determination), camshaft self-synchronization and engine position determination before engine synchronization (for pre-injection).

403Z: Engine position manager

Selection of operating modes (normal mode, camshaft or crankshaft limp-home mode) and generation of synchronization status information.

4040: Engine position and speed calculation

Calculation of engine position, speed, speed gradient, fast engine speed, segment period and number.

4072: Synchronization determination in camshaft limp-home

Engine synchronization determination in camshaft camshaft limp-home by specific injection pattern management

402S: Camshaft adaptation and position output

Camshaft edge position adaptation and camshaft position feedback for VVT-controller.

A0DZ: Engine position and speed diagnosis manager

Generation of diagnosis information and diagnosis inhibition flags.

A004: Camshaft sensor diagnosis

Camshaft segment period diagnosis and camshaft ratio check.

A0B4: Camshaft position diagnosis

Diagnosis of camshaft to crankshaft reference position.

A005: Crankshaft sensor diagnosis

Crankshaft tooth number and tooth period diagnosis.

A0B9: Crankshaft sensor plausibility diagnosis

Failure detection if crankshaft synchronization cannot be achieved, if camshaft signal is missing and if camshaft signal not valid for engine synchronization.

A021: Crankshaft sensor circuit diagnosis

Detection of open circuit and short circuit for magnetic crankshaft position sensor.

23.2 ENSD Configuration Data

Input data:

NC_CYL_NR	NC_NR_EDGE_CAM_IN	NC_NR_EDGE_CAM_EX	NC_N_MAX
NC_N_MIN	NLC_CAM_IN	NLC_CAM_EX	NC_NR_CAM_CBK
NC_OFS_TDC0_REF_CRK	NC_PSN_SEG_TDC_REF	NC_PRI_LIH_CAM_IN	NC_PRI_LIH_CAM_CBK
NC_T_SEG_MAX_CAM_IN	NC_T_SEG_MAX_CAM_EX	NC_T_SEG_MIN_CAM_IN	NC_T_SEG_MIN_CAM_EX
NLC_LIH_CAM_IN	NLC_LIH_CAM_EX	NC_PRI_SYN_CAM_IN	NC_PRI_SYN_CAM_CBK
NC_PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	NC_PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	NC_NR_TOOTH_TOL_MISS	NC_NR_TOOTH_TOL_ADD
NC_ACT_CRK_EDGE	NC_NR_TOOTH	NC_NR_TOOTH_GAP	NC_NR_VLD_TOOTH
NC_PHA_SEG_ER_ENSD	NC_NR_TOOTH_STALL	NC_CRK_WIN_SEG_LEN	NC_NR_GAP
NC_ACT_CAM_EDGE_SYN	NC_ACT_CAM_EDGE_LIH	NC_N_SEG_HALF_END	NC_NR_TOOTH_FIRST_GAP_MIN
NC_N_CRK_WIN_ENA			

General information :

The following describes the general rules for determination of the configuration data

23.2.1 Global configuration data

Here are listed the configuration data, which can be used in other aggregates :

Data	Value
NC_CYL_NR	6
NC_N_MIN	22
NC_N_MAX	8160
NLC_CAM_IN	1: intake camshaft sensor(s) present in system
NLC_CAM_EX	1: exhaust camshaft sensor(s) present in system
NC_NR_CAM_CBK	1: camshaft sensor present on cylinder bank 1
NC_NR_EDGE_CAM_IN	6
NC_NR_EDGE_CAM_EX	6

23.2.2 Local configuration data

Here are listed the configuration data, which are used only in the ENSD aggregate.

Data	Value
NLC_LIH_CAM_IN	1: intake camshaft sensor available for crankshaft limp-home
NLC_LIH_CAM_EX	1: exhaust camshaft sensor available for crankshaft limp-home
NC_PRI_LIH_CAM_IN	Camshaft selection priority for crankshaft limp-home: 1: intake camshaft sensor(s)

Data	Value
NC_PRI_LIH_CAM_CBK	Camshaft selection priority for crankshaft limp-home: 1: sensor(s) on cylinder bank 1
NC_PRI_SYN_CAM_IN	Camshaft selection priority for engine synchronization: 1: intake camshaft sensor(s)
NC_PRI_SYN_CAM_CBK	Camshaft selection priority for engine synchronization: 1: sensor(s) on cylinder bank 1
NC_T_SEG_MIN_CAM_IN	Intake camshaft sensor minimum signal duration: 0,000694
NC_T_SEG_MIN_CAM_EX	Exhaust camshaft sensor minimum signal duration: 0,000694
NC_T_SEG_MAX_CAM_IN	Intake camshaft sensor maximum signal duration: 1,409090
NC_T_SEG_MAX_CAM_EX	Exhaust camshaft sensor maximum signal duration: 1,409090
NC_CRK_SENS_TYP	Crankshaft sensor type: 0: ACPS
NC_CAM_SENS_TYP	Camshaft sensor type: 2: ACAM TPO
NC_CAM_WHEEL_TYP	Camshaft target wheel type: 1: multi-teeth, 1 gap on crk
NC_NR_CONF_CAM	Number of camshaft target wheel configurations: 1: only 1 configuration
NC_ACT_CRK_EDGE	Active edge of crankshaft signal: 0: falling edges
NC_ACT_CAM_EDGE_SYN	Active edge of camshaft signal for cam/crk synchronization: 3: falling and rising edges
NC_ACT_CAM_EDGE_LIH	Active edge of camshaft signal for crankshaft limp-home: 3: falling and rising edges
NC_NR_TOOTH	60
NC_NR_GAP	1
NC_NR_TOOTH_GAP	2
NC_NR_TOOTH_TOL_ADD	2
NC_NR_TOOTH_TOL_MISS	2
NC_NR_TOOTH_STALL	3
NC_NR_VLD_TOOTH	6
NC_OFS_TDC0_REF_CRK	60°
NC_PSN_SEG_TDC_REF	54°

Data	Value
NC_PHA_SEG_ER_ENSD	42°
NC_CRK_WIN_SEG_LEN	30°
NLC_ESS	1: engine speed (tacho) output controlled by basic SW
NC_CAM_LIH_SWI	0: No specific method for synchronization determination in camshaft limp home
NC_N_SEG_HALF_END	22
NC_NR_TOOTH_FIRST_GAP_MIN	2: minimum number of teeth to be simulated to detect the first reference gap
NC_N_CRK_WIN_ENA	0: minimum engine speed for enabling acceptance windows for crankshaft acquisition

Application conditions:

Recurrence: once, at ignition key on (ECU reset)

Formula Section:

Edge index z	1	2	3	4	5	6
NC_PSN_EDGE_z_CAM_IN_1	80	210	390	430	570	30
NC_PSN_EDGE_z_CAM_EX_1	80	210	390	430	570	30

Note:

- Per definition edge #1 is the first electrical **FALLING** edge after TDC0
- Camshaft edge positions are given in °CRK relative to TDC0
- Camshaft edge positions are given for VVT passive and are only valid for BMW NG6 engines with a 3 teeth camshaft wheel

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CRK_SENS_TYP	1	0H 1H	ACPS MCPS	1	[-]
Crankshaft sensor technology					
NC_CAM_WHEEL_TYP	1	0H 1H 2H	SINGLE_TOOTH MULTI_TEETH_1_GAP MULTI_TEETH_2_GAPS	1	[-]
Camshaft target wheel type					
NC_CAM_SENS_TYP	1	0H 1H 2H	MCAM ACAM ACAM_TPO	1	[-]
Camshaft sensor technology					
NC_NR_CONF_CAM	1	1...2H	1...2	1	[-]
Number of camshaft target wheel configurations					
NLC_ESS	1	0H 1H	NOT_PRESENT PRESENT	1	[-]
Engine speed (Tacho) hardware output controlled by basic SW present					
NC_CAM_LIH_SWI	1	0H 1H 2H 3H	NONE INJ FUP MAF	1	[-]
Method for synchronization determination in camshaft limp home					

23.3 ENSD interface adaptation module

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TCO_CMN	V/O	0...FFH	-50...205	1	[°C]
Coolant temperature (ENSD internal)					
VB_CMN	V/O	0...FFH	0...28.8	1.13E-01	[V]
Battery voltage (ENSD internal)					

Input data:

TCO	VB	NC_CAM_LIH_SWI	

FUNCTION DESCRIPTION:

General information:

The aim is to get a common definition for coolant temperature and battery voltage inside ENSD aggregate whatever their definition is in the rest of software.

Description:

TCO_CMN and VB_CMN are stubbed

Application conditions:

Initialisation: none

Recurrence: at reset

Activation: *NC_CAM_LIH_SWI ≠ "INJ"*

Deactivation: NC_CAM_LIH_SWI = "INJ"

Formula section:

TCO_CMN = 0
VB_CMN = 0

Remark: As this module is only a stub, TCO and VB are not used.

23.4 Acquisition of Crankshaft Signal

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ST_CRK_SYN	-	0...7H	0...7	1	[-]
Crankshaft synchronisation state					
LV_CRK_STOP	O	0...1H	0...1	1	[-]
Flag for Stop Engine request					
LV_CRK_RUN	O	0...1H	0...1	1	[-]
Flag for Running engine					
LV_ORNG_TOOTH_PER_CRK	V/O	0...1H	0...1	1	[-]
Invalid crankshaft tooth period					
LV_ORNG_NR_TOOTH_CRK	V/O	0...1H	0...1	1	[-]
Crankshaft tooth count incorrect at reference gap					
LV_LOST_SYN_CRK	V/O	0...1H	0...1	1	[-]
Crankshaft synchronization lost					
CRK_CTR	-	0...FFH	0...255	1	[-]
Crankshaft counter					
CRK_ADD_CTR	-	0...FFH	0...255	1	[-]
Crankshaft additional teeth counter					
CRK_MISS_CTR	-	0...FFH	0...255	1	[-]
Crankshaft missing teeth counter					
LV_CRK_FIRST_VLD_TOOTH	O	0...1H	0...1	1	[-]
Flag for first valid crankshaft tooth detected ready to synchronize					
LV_CRK_SYN	V/O	0...1H	0...1	1	[-]
Flag for crankshaft acquisition synchronized					
T_TOOTH	O	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Crankshaft tooth period					
T_SEG_ENSD	O	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Fine resolution segment period					
T_SEG_HALF_ENSD	O	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Fine resolution half-segment period					
T_SEG_ER	V/O	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Crankshaft segment period for misfire detection (ER algorithm)					
T_CRK_WIN_ENSD	V/O	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Crankshaft tooth window at end of segment					
LV_FIRST_REF_GAP	-	0...1H	0...1	1	[-]
First Reference gap is found					
LV_REF_GAP	-	0...1H	0...1	1	[-]
Second Reference gap is found					
LV_CRK_MISS_TOOTH	V/O	0...1H	0...1	1	[-]
At least one missing tooth has been simulated					

Input data:

N_32	LV_ACT_CRK	NC_CYL_NR	NC_NR_GAP

FUNCTION DESCRIPTION:

23.4.1 Crankshaft Synchronization

General information:

The crankshaft signal is generated by the crankshaft sensor in conjunction with the crankshaft target wheel. The target wheel has a theoretical number of NC_NR_TOOTH (typically 60) teeth, and NC_NR_TOOTH_GAP missing teeth (typically 2) forming the reference gap (“60-2”). The crankshaft sensor converts the mechanical shape into an electrical signal.

Magnetic Crankshaft Position Sensors (MCPS) deliver an AC signal with amplitude depending primarily on crankshaft speed, and sensor air gap. The signal is shaped into a rectangular signal by the ECU input circuitry. Each edge of that signal represents a tooth or a gap of the target wheel. Only signal edges generated from teeth will be processed by software (active signal edge definition). These may be falling or rising edges, depending on sensor polarity.

Active Crankshaft Position Sensors (ACPS) switch an open collector output on and off with every tooth or gap passing. Connecting the sensor output to a pull-up resistor inside the ECU generates a signal with rectangular shape. Each edge of that signal represents a tooth or a gap of the target wheel. Only falling signal edges will be processed by software (active signal edge definition), because they are faster. Design and installation of the sensor must ensure that falling signal edges are generated from teeth.

The corresponding configuration data is NC_ACT_CRK_EDGE.

The choice is taken in a way not to use the unprecise signal edge generated in the crankshaft reference gap.

It may be 0 (falling) or 1 (rising) with MCPS, depending on sensor polarity.

With ACPS it should always be 0, because the falling edge is faster due to the sensor’s open collector interface. The unprecise signal edge in the reference gap is avoided by choosing the correct installation of the sensor relativ to the target wheel motion direction.

Application conditions:

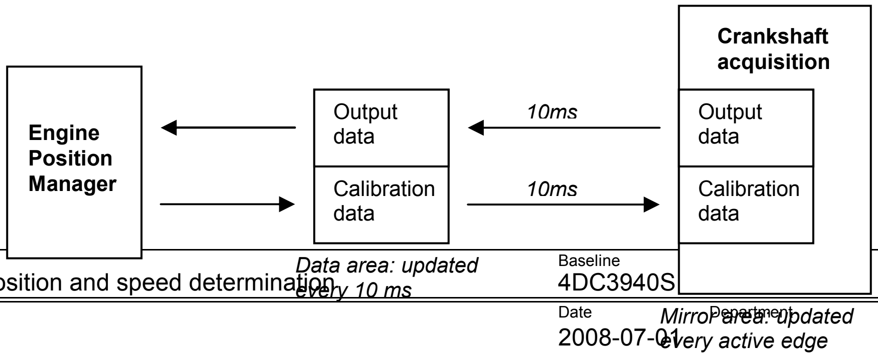
Recurrence:

For set of the RAM as flags, a mirror area is defined which contains a copy of the output and calibration data in the basic software area (see figure below).

The mirror area of the output data is updated by the lower layer at every active signal edge.

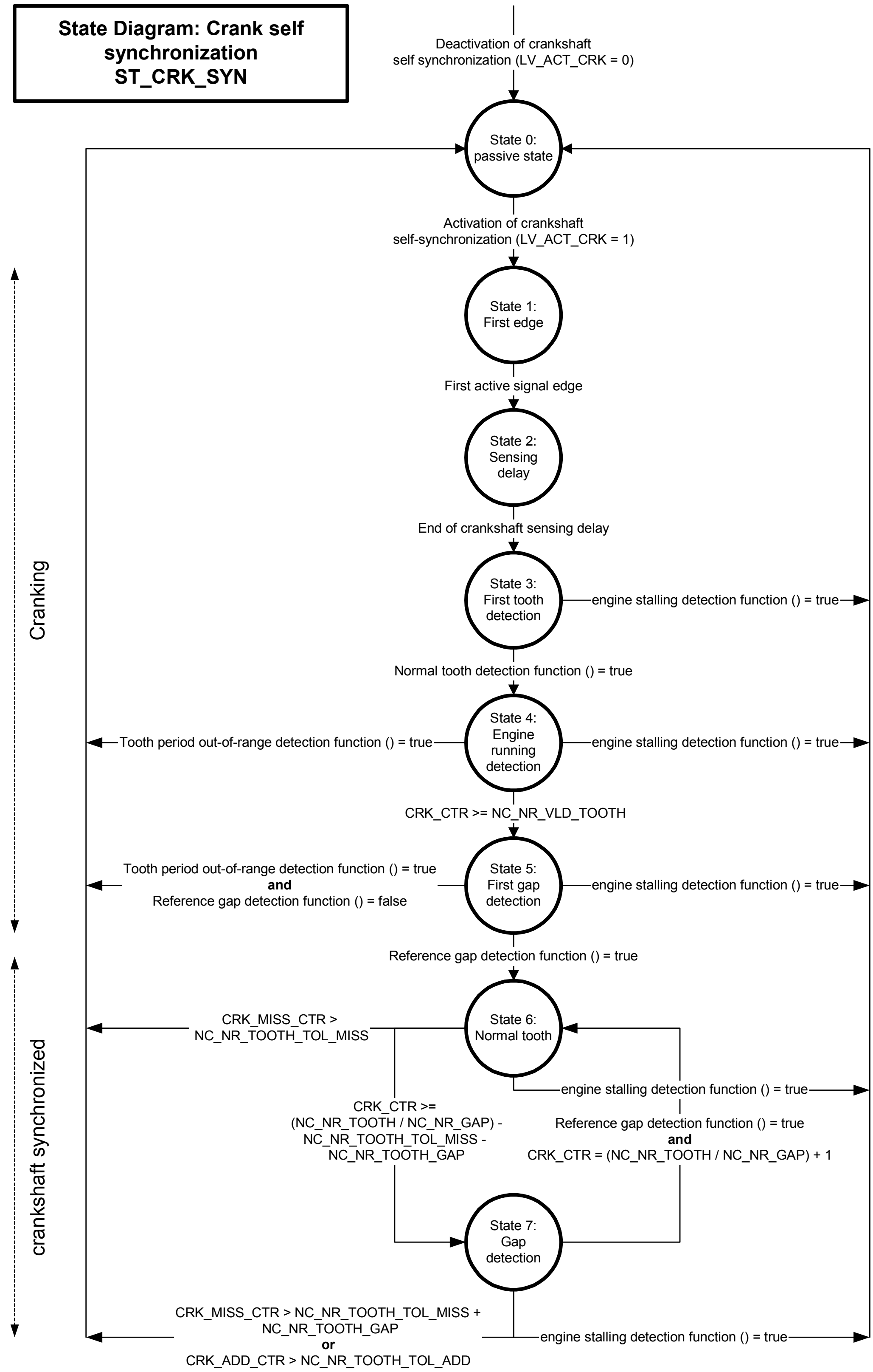
The mirror area of the calibration data is updated by the upper layer each 10ms.

The output data is copied from the mirror area every 10 ms

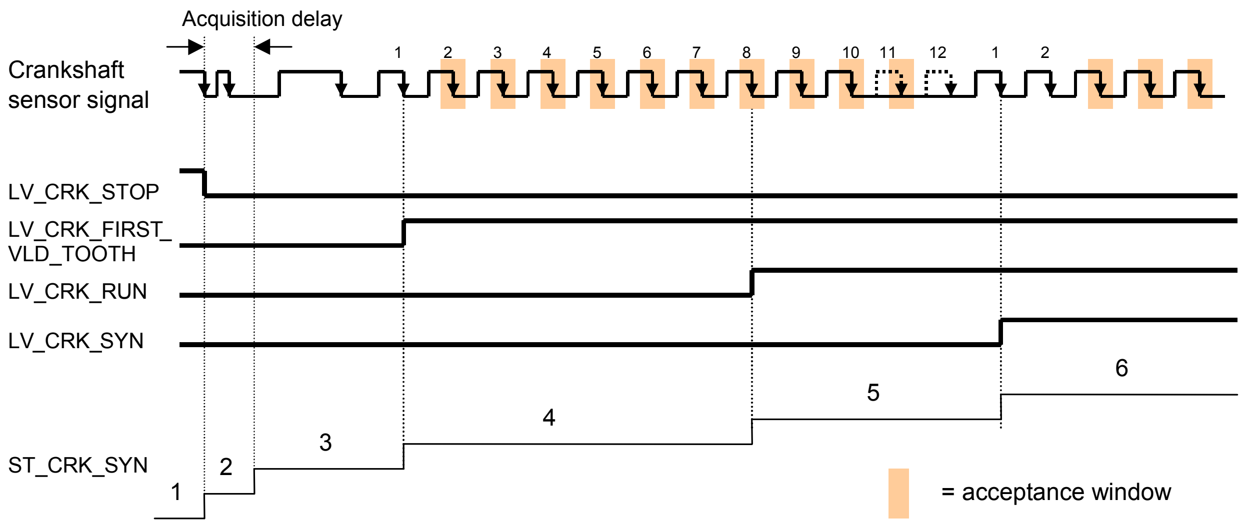


State Flow Diagram:

State Diagram: Crank self synchronization ST_CRK_SYN



Timing Diagram (example):



Formula section:

State 0: Crankshaft passive state

Input condition:

From state 4, 5:	Tooth period out-of-range detection function () = true
From state 6:	CRK_MISS_CTR > NC_NR_TOOTH_TOL_MISS
From state 7:	CRK_ADD_CTR > NC_NR_TOOTH_TOL_ADD Or CRK_MISS_CTR > NC_NR_TOOTH_TOL_MISS + NC_NR_TOOTH_GAP
From state 3,4,5,6,7:	Engine stalling detection function () = true
From EPM:	LV_ACT_CRK = 0

Output condition:

To State 1: Activation of crk self-synchronization (LV_ACT_CRK = 1)

Action in the state:

LV_CRK_FIRST_VLD_TOOTH = 0

LV_CRK_SYN = 0

CRK_CTR = 0

LV_CRK_STOP = 1

LV_CRK_RUN = 0

CRK_MISS_CTR = 0

CRK_ADD_CTR = 0

LV_FIRST_REF_GAP = 0

LV_REF_GAP = 0

Transient action:

None

State 1: First signal edge

Input condition:

From state 0: Activation of crk self-synchronization (LV_ACT_CRK = 1)

Output condition:

To state 2: First active signal edge

Action in the state:

Wait for first active signal edge

LV_CRK_STOP = 0

Transient action:

No actions

State 2: Crankshaft sensing delay

Input condition:

From state 1: First active signal edge detected

Output condition:

To state 3: End of crankshaft acquisition delay time

Action in the state:

Crankshaft signal acquisition inactive during a time interval C_T_CRK_DLY.

Transient action:

To state 3:	LV_ORNG_TOOTH_PER_CRK = 0
	LV_ORNG_NR_TOOTH_CRK = 0
	LV_LOST_SYN_CRK =0

State 3: First tooth detection

Input condition:

From state 2:	End of crankshaft acquistion delay time

Output condition:

State 4:	Normal tooth detection function () = true
State 0:	Engine stalling detection function () = true

Action in the state

Wait two next active signal edges in order to compute the first period

Activation of Normal tooth detection function ()

T_TOOTH n-1 = T_TOOTH n

Activation of Tooth period out-of-range detection function ()

Activation of Engine stalling detection function ()

Transient action:

To state 4:	LV_CRK_FIRST_VLD_TOOTH = 1
	Increment CRK_CTR

State 4: Engine running detection

Input condition:

From state 3:	Normal tooth detection function () = true

Output condition:

State 5:	CRK_CTR >= NC_NR_VLD_TOOTH

State 0: Engine stalling detection function () = true
Or
Tooth period out-of-range detection function () = true

Action in the state:

```
If      Normal tooth detection function () = true
    /* tested only when no acceptance window can be applied */
        Or
        Tooth period out-of-range detection function () = false
    /* tested only when acceptance window can be applied */
Then    Increment CRK_CTR
Endif

If      Tooth period out-of-range detection function () = true
Then    LV_ORNG_TOOTH_PER_CRK = 1
Endif
```

Action in transient:

To state 5: LV_CRK_RUN = 1 (Engine speed can be computed)

State 5: First gap detection

Input condition:

From state 4: CRK_CTR >= NC_NR_VLD_TOOTH

Output conditions:

To state 6: Reference gap detection function () = true
To State 0: Engine stalling detection function () = true
Or
(Tooth period out-of-range detection function () = true
And
Reference gap detection function () = false)

Action in the state:

Activation of Reference gap detection function ()

Activation of Missing tooth simulation in gap position function ()

```
If      Tooth period out-of-range detection function () = false
Then    Increment CRK_CTR
Endif

If      Tooth period out-of-range detection function () = true
        And
        Reference gap detection function () = false
Then    LV_ORNG_TOOTH_PER_CRK = 1
Endif
```

Transient action:

To state 6:
LV_CRK_SYN =1
CRK_CTR = 1
LV_FIRST_REF_GAP = 1

State 6: Normal tooth

Input conditions:

From State 5: Reference gap detection function () = true

From State 7: Reference gap detection function () = true
And
CRK_CTR = (NC_NR_TOOTH / NC_NR_GAP) + 1

Output conditions:

To State 7: CRK_CTR >= (NC_NR_TOOTH / NC_NR_GAP) –
NC_NR_TOOTH_GAP - NC_NR_TOOTH_TOL_MISS

To State 0: CRK_MISS_CTR > NC_NR_TOOTH_TOL_MISS
Or
Engine stalling detection function () = True

Action in the state:

```
Deactivation of Missing tooth simulation in gap position function ()
Activation of Missing tooth simulation function ()
If      Normal tooth detection function () = true
    /* tested only when no acceptance window can be applied */
        Or
        Tooth period out-of-range detection function () = false
    /* tested only when acceptance window can be applied */
Then    Increment CRK_CTR
Endif

If      Tooth period out-of-range detection function () = true
Then    LV_ORNG_TOOTH_PER_CRK = 1
Endif
```

Transient action:

To State 0:

```
If      CRK_MISS_CTR > NC_NR_TOOTH_TOL_MISS
Then    LV_LOST_SYN_CRK = 1
Endif
```

State 7: Gap detection

Input condition:

From state 6:
CRK_CTR >= (NC_NR_TOOTH / NC_NR_GAP) –
NC_NR_TOOTH_GAP - NC_NR_TOOTH_TOL_MISS

Output condition:

To State 6:
Reference gap detection function () = true
And
CRK_CTR = (NC_NR_TOOTH / NC_NR_GAP) + 1

To State 0:
CRK_MISS_CTR > NC_NR_TOOTH_TOL_MISS +
NC_NR_TOOTH_GAP
Or
CRK_ADD_CTR > NC_NR_TOOTH_TOL_ADD

```
Or
Engine stalling detection function () = True
```

Action in the state:

```
Deactivation of Missing tooth simulation function ()
Activation of Missing tooth simulation in gap position function ()
If    Tooth period out-of-range detection function () = false
Then  Increment CRK_CTR
      If    CRK_CTR > (NC_NR_TOOTH / NC_NR_GAP) –
                  NC_NR_TOOTH_GAP
            Then  Increment CRK_ADD_CTR
            Endif
Endif

If    Tooth period out-of-range detection function () = true
            And
            Reference gap detection function () = false
Then  LV_ORNG_TOOTH_PER_CRK = 1
Endif
```

Transient action:

To state 6:

```
If    CRK_MISS_CTR > 0
      Or
      CRK_ADD_CTR > 0
Then  LV_ORNG_NR_TOOTH_CRK = 1
Endif
CRK_CTR = 1
CRK_MISS_CTR = 0
CRK_ADD_CTR = 0
LV_REF_GAP =1
```

To State 0:

```
If    CRK_MISS_CTR > NC_NR_TOOTH_TOL_MISS +
                              NC_NR_TOOTH_GAP
      Or
```

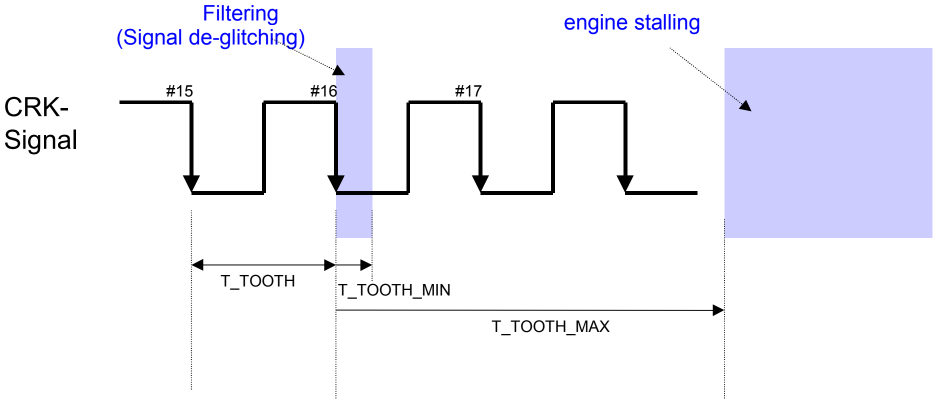
```
        CRK_ADD_CTR > NC_NR_TOOTH_TOL_ADD
Then    LV_LOST_SYN_CRK = 1
Endif
```

23.4.2 Definition of the sub function task

23.4.2.1 Normal tooth detection function ()

General information:

The purpose of this function is to validate the tooth period of a normal tooth when no acceptance window can be applied (first valid tooth, first tooth after a reference gap ...). The function returns false when the tooth period is too small (Signal edges detected before a delay time of T_TOOTH_MIN are not valid).



Formula section:

$$T\_TOOTH\_MIN = \frac{60}{NC\_NR\_TOOTH * NC\_N\_MAX}$$

```
If      T_TOOTH => T_TOOTH_MIN
Then    return (true)
Else    return (false)
Endif
```

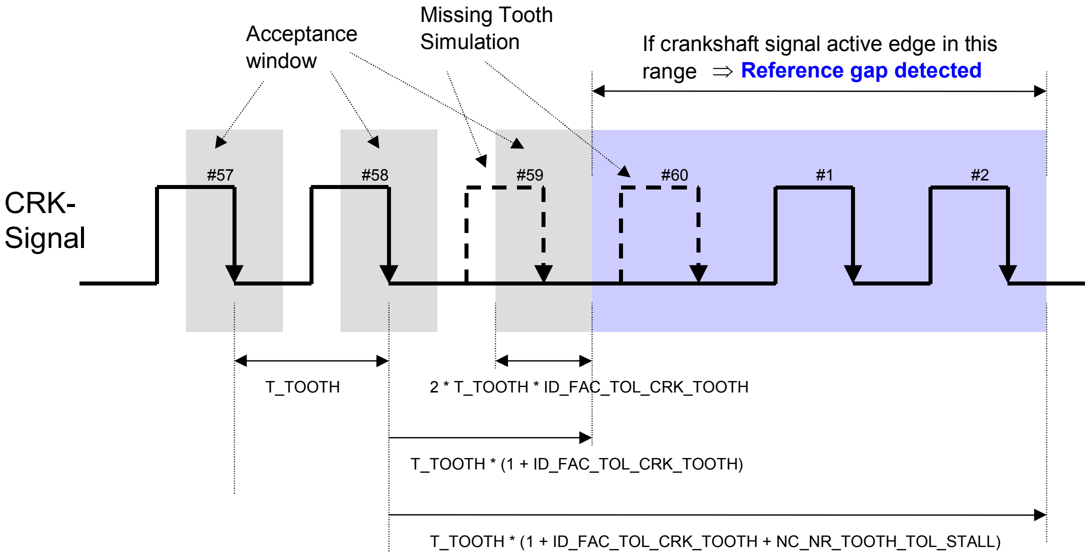
23.4.2.2 Reference gap detection function ()

General information:

The purpose of this function is to detect a reference gap.

A reference gap is detected if no signal edge was detected inside the crankshaft acceptance window and at least 1 (after synchronization) or NC_NR_TOOTH_FIRST_GAP_MIN (first gap detection) teeth have been completely simulated.

The follow figure shows the reference gap detection after synchronization:



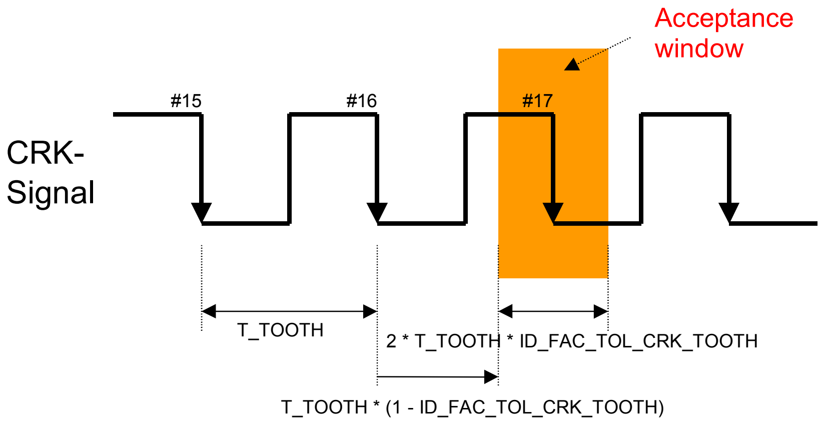
Formula section:

```
If      LV_CRK_SYN = 1
Then    If      Number of simulated missing teeth in gap position >=1
        Then    If      (NC_NR_TOOTH / NC_NR_GAP) – CRK_MISS_CTR <=
                            CRK_CTR <=
                            (NC_NR_TOOTH / NC_NR_GAP) + CRK_ADD_CTR
                Then    (correct tooth counter)
                        CRK_CTR = CRK_CTR + CRK_MISS_CTR – CRK_ADD_CTR
                        return (true)
                Else    return (false)
        Endif
Else    If      Number of simulated missing teeth in gap position >=
                        NC_NR_TOOTH_FIRST_GAP_MIN
        Then    return (true)
        Endif
Endif
```

23.4.2.3 Tooth period out-of-range detection function ()

General information:

The purpose of this function is to check if the tooth period is within a defined window area. The function returns true if the tooth period is outside the acceptance window.



Formula section:

Before first reference gap:

If T_TOOTH n-1 * (1 – ID_FAC_TOL_CRK_TOOTH) ≤ T_TOOTH n ≤ T_TOOTH n-1 * (1 + ID_FAC_TOL_CRK_TOOTH)

Then return (false)

Else return (true)

After first reference gap:

If T_TOOTH n-1 * (1 – ID_FAC_TOL_CRK_TOOTH) ≤ T_TOOTH n

Then return (false)

Else return (true)

Remark: If T_TOOTH > T_TOOTH n-1 * (1 + ID_FAC_TOL_CRK_TOOTH) after first reference gap, then a tooth is simulated (see below “Missing Tooth Simulation Function”).

The acceptance window is disabled

- on the first and second crankshaft signal edge after a reference gap detection
- on the first and second crankshaft signal edge after simulation of a missing tooth
- after engine synchronizarion, it could also be disabled below an engine speed, for example for engines having strong accelerations/decelerations at start:

  If N < NC_N_CRK_WIN_ENA

  And CRK_CTR > NC_NR_TOOTH_TOL_ADD

  And CRK_CTR < (NC_NR_TOOTH / NC_NR_GAP) - NC_NR_TOOTH_GAP - NC_NR_TOOTH_TOL_MIS – 2 (for latency time)

  This effectively means that the acceptance window is only applied for GAP detection below the configured engine speed threshold.

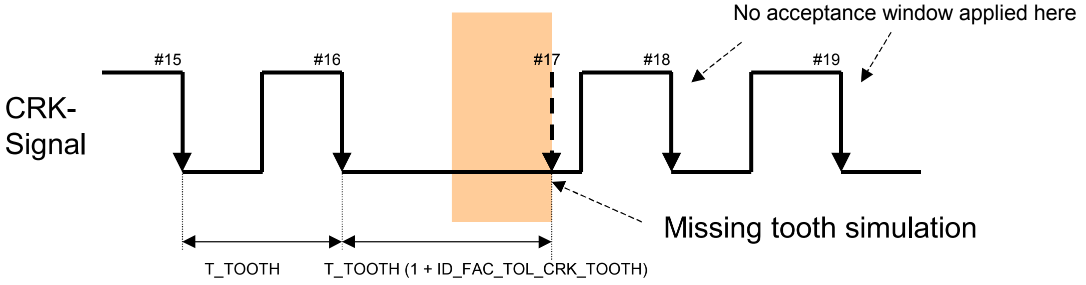
23.4.2.4 Missing Tooth Simulation Function ()

General information:

Simulate a tooth with tooth period T_TOOTH n-1 (last measured tooth period).

All measurements shall be based on the simulated tooth not on real edges.

Simulation starts at the last (real or simulated) signal edge if no signal edge is detected inside the acceptance window defined by ID_FAC_TOL_CRK_TOOTH.



Formula section:

Increment CRK_CTR.

Increment CRK_MISS_CTR.

LV_CRK_MISS_TOOTH = 1

23.4.2.5 Missing Tooth Simulation in Gap Position Function ()

General information:

Simulate a tooth with tooth period T_TOOTH n-1 (last measured tooth period).

Simulation starts at the last (real or simulated) signal edge if no signal edge is detected inside the acceptance window defined by ID_FAC_TOL_CRK_TOOTH.

Simulation stops when a edge is detected after the first simulated missing tooth.

All measurements shall be based on the simulated tooth not on real edges.

Formula section:

Increment CRK_CTR.

If number of simulated teeth < NC_NR_TOOTH_GAP

And

Missing tooth simulation stopped

Then Increment CRK_MISS_CTR

High acceleration during reference gap: only one tooth could be simulated

Endif

```
If      number of simulated teeth > NC_NR_TOOTH_GAP
Then    Increment CRK_ADD_CTR
        High deceleration during reference gap: more than NC_NR_TOOTH_GAP teeth
        simulated
Endif
```

23.4.2.6 Engine stalling detection function ()

General information:

Engine stalling is detected when the number of consecutive simulated teeth exceeds NC_NR_TOOTH_STALL or when the tooth period is greater than T_TOOTH_MAX.

Formula section:

$$\text{T\_TOOTH\_MAX} = \frac{60}{NC\_NR\_TOOTH * NC\_N\_MIN}$$

```
If      more than NC_NR_TOOTH_ STALL consecutive teeth are simulated
        Or
        T_TOOTH (1 + ID_FAC_TOL_CRK_TOOTH) >= T_TOOTH_MAX
Then    return (true)
Else    return (false)
Endif
```

23.4.3 Requirements to infrastructure

23.4.3.1 Crankshaft Tooth Time Measurement (T_TOOTH)

General information:

System request accuracy <= 4 us

The crankshaft tooth period T_TOOTH is the measured time between two consecutive valid active crankshaft signal edges (falling or rising according to configuration).

If a missing tooth is simulated, T_TOOTH represents the time period between the last two real valid signal edges. T_TOOTH keeps the latest value until a new measured value is available.

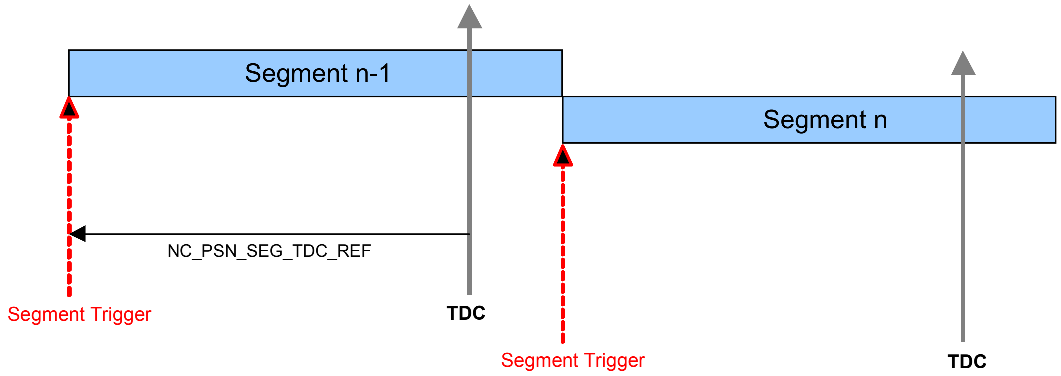
23.4.3.2 Segment Period Measurement (T_SEG_ENSD) and Trigger Generation

General information:

System request accuracy <= 1 us

At Segment event, a trigger is defined and a control signal is generated to trigger execution of segment synchronous tasks.

The corresponding time intervals T_SEG_ENSD are measured between tooth events located NC_PSN_SEG_TDC_REF degrees before each TDC.



Application conditions:

Initialisation:	T_SEG_ENSD is set to the maximum value
Recurrence :	before synchronization: once (when LV_CRK_RUN is set) after synchronization: NC_PSN_SEG_TDC_REF degrees before each TDC (segment trigger)
Activation :	LV_CRK_RUN = 1
Deactivation :	LV_CRK_STOP = 1

Formula section:

Before synchronization, one segment time is output at the moment when crankshaft rotation is validated. The first segment period is calculated from the most recent tooth period:

T_SEG_ENSD = (2 * T_TOOTH * NC_NR_TOOTH) / NC_CYL_NR

The same calculation applies to the first segment after synchronization.

After synchronization:

T_SEG_ENSD = time interval between tooth events located NC_PSN_SEG_TDC_REF degrees before each TDC

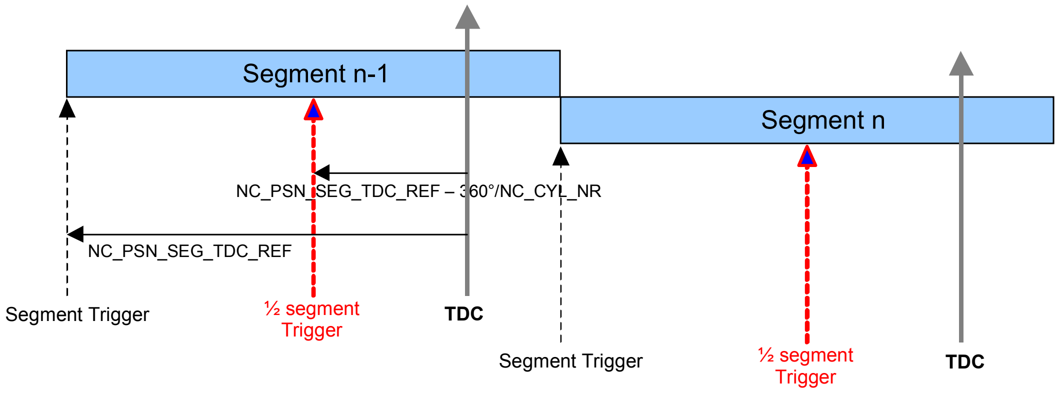
23.4.3.3 Half-Segment Period Measurement (T_SEG_HALF_ENSD) and Trigger Generation

General information:

System request accuracy <= 1 us

At the half of a segment (in the middle of two segment events) a half-segment event is generated and an half-segment trigger is defined and send to the ASW if the engine speed is lower than a threshold defined by NC_N_SEG_HALF_END.

This function provides a output value for the half-segment period. This measured time is send to the ASW.



Application conditions:

Initialisation:	T_SEG_HALF_ENSD is set to the maximum value
Recurrence:	before synchronization: once (when LV_CRK_RUN is set) after synchronization: NC_PSN_SEG_TDC_REF degrees before each TDC (segment trigger) **and** NC_PSN_SEG_TDC_REF – 360° CRK / NC_CYL_NR degrees before each TDC (half-segment trigger) **if** T_SEG_HALF_ENSD >= 60 / (NC_CYL_NR * NC_N_SEG_HALF_END)
Activation:	LV_CRK_RUN = 1
Deactivation:	LV_CRK_STOP = 1

Formula section:

The first half segment period is calculated from the most recent tooth period:

T_SEG_HALF_ENSD = (T_TOOTH * NC_NR_TOOTH) / NC_CYL_NR

After synchronization:

T_SEG_HALF_ENSD = time elapsed between

- a segment event and a half-segment event or
- a half-segment event and a segment event

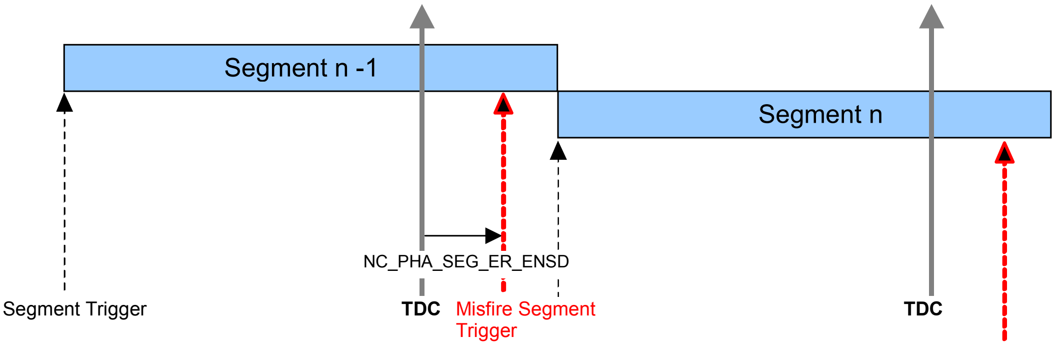
23.4.3.4 Misfire Segment Period Measurement (T_SEG_ER)

General information:

System request accuracy <= 1 us

The function provides output values for the misfire segment period.

The misfire segment period T_SEG_ER corresponds to the TDC period, measured at the tooth events, which are closest to NC_PHA_SEG_ER_ENSD after each TDC.



Application conditions:

Initialisation: T_SEG_ER is set to the maximum value

Recurrence: every misfire segment

Formula section:

T_SEG_ER = time interval between misfire segment tooth events.

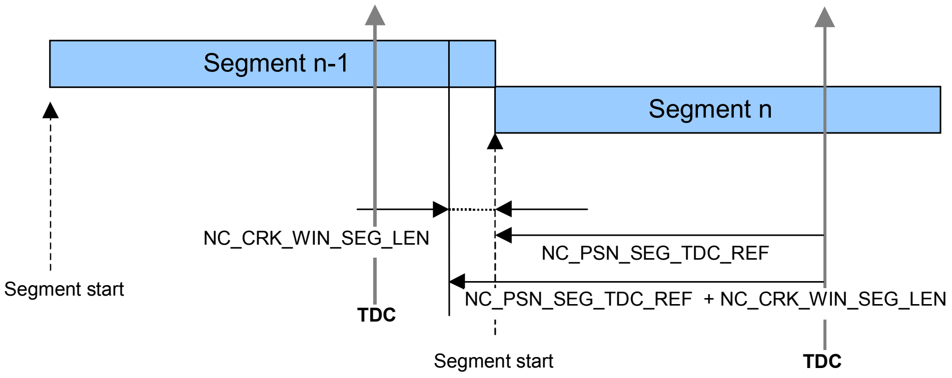
23.4.3.5 Crankshaft tooth window measurement (T_CRK_WIN_ENSD)

General information:

System request accuracy <= 1 us

The time interval elapsed between tooth events located between NC_CRK_WIN_SEG_LEN degrees before the end of each segment and the segment trigger is measured and send to the ASW segment synchronous

It provides and measures the time of the last NC_CRK_WIN_SEG_LEN teeth period before segment event.



Application conditions:

Initialisation: T_CRK_WIN_ENSD is set to the maximum value

Recurrence: every segment

Activation: LV_CRK_SYN = 1

Deactivation: LV_CRK_STOP = 1

Formula section:

```
If      LV_CRK_SYN is set to 1 between NC_CRK_WIN_SEG_LEN degrees before the
        end of the segment and the segment trigger event
Then    /* Recalculation from last available tooth period */
        T_CRK_WIN = (T_TOOTH * NC_NR_TOOTH) / (360°CRK /
        NC_CRK_WIN_SEG_LEN)
Else    /* Normal calculation */
        T_CRK_WIN_ENSD = time interval between tooth events located between
                NC_CRK_WIN_SEG_LEN degrees before the end of each segment and the
                segment triger
Endif
```

23.4.3.6 ID_FAC_TOL_CRK_TOOTH calculation

General information:

ID_FAC_TOL_CRK_TOOTH[0rpm] is used until crankshaft synchronization (i.e. until first gap is detected, LV_CRK_SYN is set).

Then after crankshaft synchronization, ID_FAC_TOL_CRK_TOOTH is calculated depending on engine speed value (N_32).

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_T_CRK_DLY	1	0...29083H	0...40000	0.2380003	[µs]
Sensing delay time after first active crankshaft signal edge					
ID_FAC_TOL_CRK_TOOTH	6	0...FH	0...0.9375	0.0625	[-]
LDP_N_32_ID_FAC_TOL_CRK_TOOTH	6	0...FFH	0...8160	32	[rpm]
Factor for calculation of expected tooth period					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_N_SEG_HALF_END	-	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for half-segment trigger					
NC_CRK_WIN_SEG_LEN	-	0...FFH	0...95.625	0.375	[°CRK]
Width of crankshaft tooth window in °CRK at end of segment					
NC_PSN_SEG_TDC_REF	-	0...780H	0...720	0.375	[°CRK]
Segment start in °CRK before TDC					
NC_OFS_TDC0_REF_CRK	-	0...780H	0...720	0.375	[°CRK]
Reference gap position in °CRK before TDC0					
NC_PHA_SEG_ER_ENSD	-	0...FFH	0...95.625	0.375	[°CRK]
Misfire segment start in °CRK after TDC					
NC_NR_TOOTH_TOL_ADD	-	1...2H	1...2	1	[-]
Number of additional teeth tolerated between two reference gap occurrences					
NC_NR_TOOTH_TOL_MISS	-	1...2H	1...2	1	[-]
Number of missing teeth tolerated between two reference gap occurrences					
NC_NR_VLD_TOOTH	-	0...FH	0...15	1	[-]
Number of valid teeth necessary for validation of crankshaft rotation					
NC_NR_TOOTH	-	1...FFH	1...255	1	[-]
Theoretical number of teeth per crankshaft revolution					
NC_NR_TOOTH_GAP	-	1...2H	1...2	1	[-]
Number of missing teeth forming the reference gap					
NC_NR_TOOTH_STALL	-	2...7H	2...7	1	[-]
Number of consecutive missing teeth for engine stalling detection					
NC_N_MIN	-	0...1FE0H	0...8160	1	[rpm]
Minimum engine speed					
NC_N_MAX	-	0...3FC0H	0...16320	1	[rpm]
Maximum engine speed					
NC_ACT_CRK_EDGE	-	0H 1H	FALLING RISING	1	[-]
Active edge of crankshaft signal					
NC_NR_TOOTH_FIRST_GAP_MIN	1	1...2H	1...2	1	[-]
Minimum number of teeth to be simulated to detect the first reference gap					
NC_N_CRK_WIN_ENA	1	0...1FE0H	0...8160	1	[rpm]
Minimum engine speed for enabling acceptance windows for crankshaft acquisition					

23.4.4 System requirements to sensor signal

23.4.4.1 General

With both sensor technologies, there is an angular offset between the mechanical reference (usually the trailing edge of a tooth) and the timing reference of the sensor (zero crossing of the MCPS signal, output switching of the ACPS) for an active edge. This offset will further on be referred to as phase angle. The phase angle depends strongly on sensor tolerances, installation tolerances (air gap), and operating conditions (speed, temperature).

The tolerances given in the following paragraphs refer to the angular position of the crankshaft target wheel when an active signal edge is recognized by the input port of the µP, or to the rotation angle between such events, respectively.

23.4.4.2 Relative phase angle tolerances

The usual synchronization algorithms can accept ±3% tolerance of the target wheel rotation angle between consecutive active signal edges.

e.g. for a “60-2” target wheel, this is 6° (18°) ±0.2°

The sensor/target wheel combination has to fulfill this requirement at all operating conditions.

The permissible accumulation of tooth angle tolerances over a TDC segment is ±0.25% of the angle for one TDC segment (720°/ NC_CYL_NR) at segment reference conditions.

e.g. for four-cylinder engines	180°	±0.45°
e.g. for six-cylinder engines	120°	±0.3°

Segment reference conditions are: 2000rpm, 80°C.

Please note that for engines with more than 8 cylinders, this requirement is more stringent than the requirement for consecutive signal edges.

23.4.4.3 Relative phase angle stability

The angle for one TDC period measured under segment reference conditions must not have a drift > 0.045° versus speed variation from 2000 rpm...7000 rpm and temperature variation within the complete range. This requirement has to be fulfilled for all tolerances and all possible displacements of the target wheel relative to the sensor (air gap, radial run-out, crankshaft flexion, bearing clearance...).

In case of MCPS sensors, phase angle stability must be measured with the ECU hardware stage.

23.4.4.4 Repeatability

Two different definitions of repeatability are functionally equal. The choice depends on the sensor test bench and measurement equipment.

1. Phase angle repeatability

(mechanical edge to signal edge for the same tooth) ±0.025°

2. Interval repeatability

(signal edge to signal edge for the same pair of teeth) ±0.05°

In both cases the repeatability is determined from the maximum and minimum value out of a large number of measurements under constant conditions.

23.4.5 Hardware and HAL requirements

23.4.5.1 EMI

Any signal disturbance must be suppressed as far as possible. Supplemental trigger events must not be generated due to such disturbance, particularly not at the slow zero crossing in the reference gap.

23.4.5.2 Phase shift repeatability

At constant operating conditions, the phase shift introduced by the input circuitry must be constant within ±0.1 µs

23.5 Acquisition of Camshaft Signal for multi-teeth Target Wheel with one or two Reference Gaps per Engine Revolution

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_SEG_CAM_IN[NC_NR_CAM_CBK]	V/O	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Camshaft segment period					
T_SEG_CAM_EX[NC_NR_CAM_CBK]	V/O	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Camshaft segment period					
REL_ANG_CAM_REF_GAP	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Measured angular distance between reference gap and camshaft edge					
REL_ANG_CRK_CAM	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Measured angular distance between consecutive active camshaft edges					
PSN_TOOTH1_CAM	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Theoretical angular distance between crankshaft tooth #1 and camshaft edge					
PSN_CAM_CAM	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Theoretical angular distance between consecutive active camshaft edges					
PSN_CAM_CAM_1	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Theoretical angular distance between the active camshaft edge and the following active edge					
ST_CAM_SS	-	0...5H	0...5	1	[-]
Camshaft self synchronization state					
ST_CAM_CRK_S	-	0...4H	0...4	1	[-]
Cam crank synchronization state					
ST_CAM_PRE_INJ_S	-	0...3H	0...3	1	[-]
Cam crank synchronization state					
LV_SYN_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Intake camshaft synchronized					
LV_SYN_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Exhaust camshaft synchronized					
CTR_EDGE_CAM_IN[NC_NR_CAM_CBK]	V/O	0...FFH	0...255	1	[-]
Continuous camshaft signal edge counter					
CTR_EDGE_CAM_EX[NC_NR_CAM_CBK]	V/O	0...FFH	0...255	1	[-]
Continuous camshaft signal edge counter					
IDX_EDGE_CAM_IN[NC_NR_CAM_CBK]	V/O	0...FH	1...16	1	[-]
Index of the last camshaft signal edge					
IDX_EDGE_CAM_EX[NC_NR_CAM_CBK]	V/O	0...FH	1...16	1	[-]
Index of the last camshaft signal edge					
RATIO_PER_CAM_IN[NC_NR_CAM_CBK]	-	0...FFH	0.0625...16	0.0625	[-]
Camshaft period ratio for synchronization					
RATIO_PER_CAM_EX[NC_NR_CAM_CBK]	-	0...FFH	0.0625...16	0.0625	[-]
Camshaft period ratio for synchronization					
LV_ORNG_PER_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Intake camshaft segment period out of range					
LV_ORNG_PER_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Exhaust camshaft segment period out of range					
LV_ORNG_RATIO_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Intake camshaft segment ratio out of range					
LV_ORNG_RATIO_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Exhaust camshaft segment ratio out of range					
LV_VLD_PSN_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Intake camshaft position measurement valid					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VLD_PSN_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Exhaust camshaft position measurement valid					
LV_CAM_STOP_IN[NC_NR_CAM_CBK]	O	0...1H	0...1	1	[-]
Self synchronization on Intake camshaft i is stopped					
LV_CAM_STOP_EX[NC_NR_CAM_CBK]	O	0...1H	0...1	1	[-]
Self synchronization on Exhaust camshaft i is stopped					
PSN_ENG_CRK_OFS	O	0...2CFFH	0...719.9375	0.0625	[°CRK]
Engine position offset for initialization at crankshaft synchronization.					
LV_CAM_SYN_CRK	V/O	0...1H	0...1	1	[-]
Camshaft acquisition ready for crankshaft synchronization.					
LV_ORNG_CAM_SYN_CRK	V/O	0...1H	0...1	1	[-]
Camshaft signal for crankshaft synchronization out of range					
RATIO_PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	-	0...FFH	0.0625...16	0.0625	[-]
Theoretical intake camshaft i period ratio at edge z					
RATIO_PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	-	0...FFH	0.0625...16	0.0625	[-]
Theoretical exhaust camshaft i period ratio at edge z					
PSN_ENG_SYN_CAM_MIN	O	0...2CFFH	0...719.9375	0.0625	[°CRK]
Minimum engine position during crankshaft synchronization phase.					
PSN_ENG_SYN_CAM_MAX	O	0...2CFFH	0...719.9375	0.0625	[°CRK]
Maximum engine position during crankshaft synchronization phase.					
CAM_DYW_SYN_IN	-	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window for angle between intake camshaft signal edges in crankshaft synchronization mode					
CAM_DYW_SYN_EX	-	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window for angle between exhaust camshaft signal edges in crankshaft synchronization mode					
CAM_DYW_CRK_SYN_ADC_IN	-	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window advance for intake camshaft to crankshaft reference in crankshaft synchronization mode					
CAM_DYW_CRK_SYN_ADC_EX	-	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window advance for exhaust camshaft to crankshaft reference in crankshaft synchronization mode					
CAM_DYW_CRK_SYN_RTD_IN	-	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window retard for intake camshaft to crankshaft reference in crankshaft synchronization mode					
CAM_DYW_CRK_SYN_RTD_EX	-	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window retard for exhaust camshaft to crankshaft reference in crankshaft synchronization mode					

Input data:

LV_CRK_FIRST_VLD_TOOTH	LV_CRK_SYN	NC_NR_TOOTH	PSN_ENG_CRK
NC_OFS_TDC0_REF_CRK	N_32	LV_ACT_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_ACT_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]
LV_ACT_CAM_IN[NC_NR_CAM_CBK]	LV_ACT_CAM_EX[NC_NR_CAM_CBK]	LV_CAM_LOCK_IVVT_IN[NC_NR_CBK_IVVT]	LV_CAM_LOCK_IVVT_EX[NC_NR_CBK_IVVT]
NLC_IVVT_IN	NLC_IVVT_EX	LV_STOP_ENG	

General information:

This specification is dedicated to acquisition of the signal from sensors and target wheels on up to 4 camshafts (intake and exhaust, cylinder bank 1 and 2).

The operating mode of the signal acquisition is controled for each camshaft separately.

A diagnostic output is delivered to allow detection of camshaft signal failure.

The signal is used for:

1. Self synchronization of all camshafts: timing validation of the camshaft signal edges in correspondance with its theoretical position (used for VVT controler, for detection of crankshaft signal failure, and for limp-home in case of crankshaft signal failure).
2. Camshaft / crankshaft synchronization of the selected camshaft: determination of the offset for the engine position calculation.
3. Engine position interface for pre-injection (with the selected camshaft used for cam/crank synchronization)

These three functions are described with three states diagrams.

Application conditions:

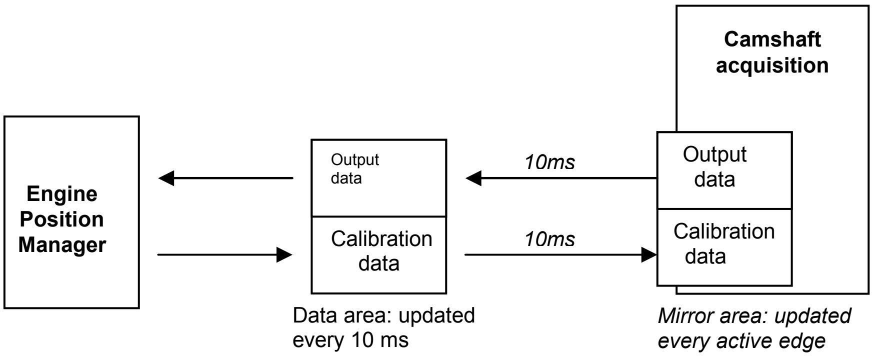
Recurrence:

For set of the RAM as flags, a mirror area is defined which contains a copy of the output and calibration data in the basic software area (see figure below).

The mirror area of the output data is updated by the lower layer at every signal edge.

The mirror area of the calibration data is updated by the upper layer each 10ms.

The output data is copied from the mirror area every 10 ms

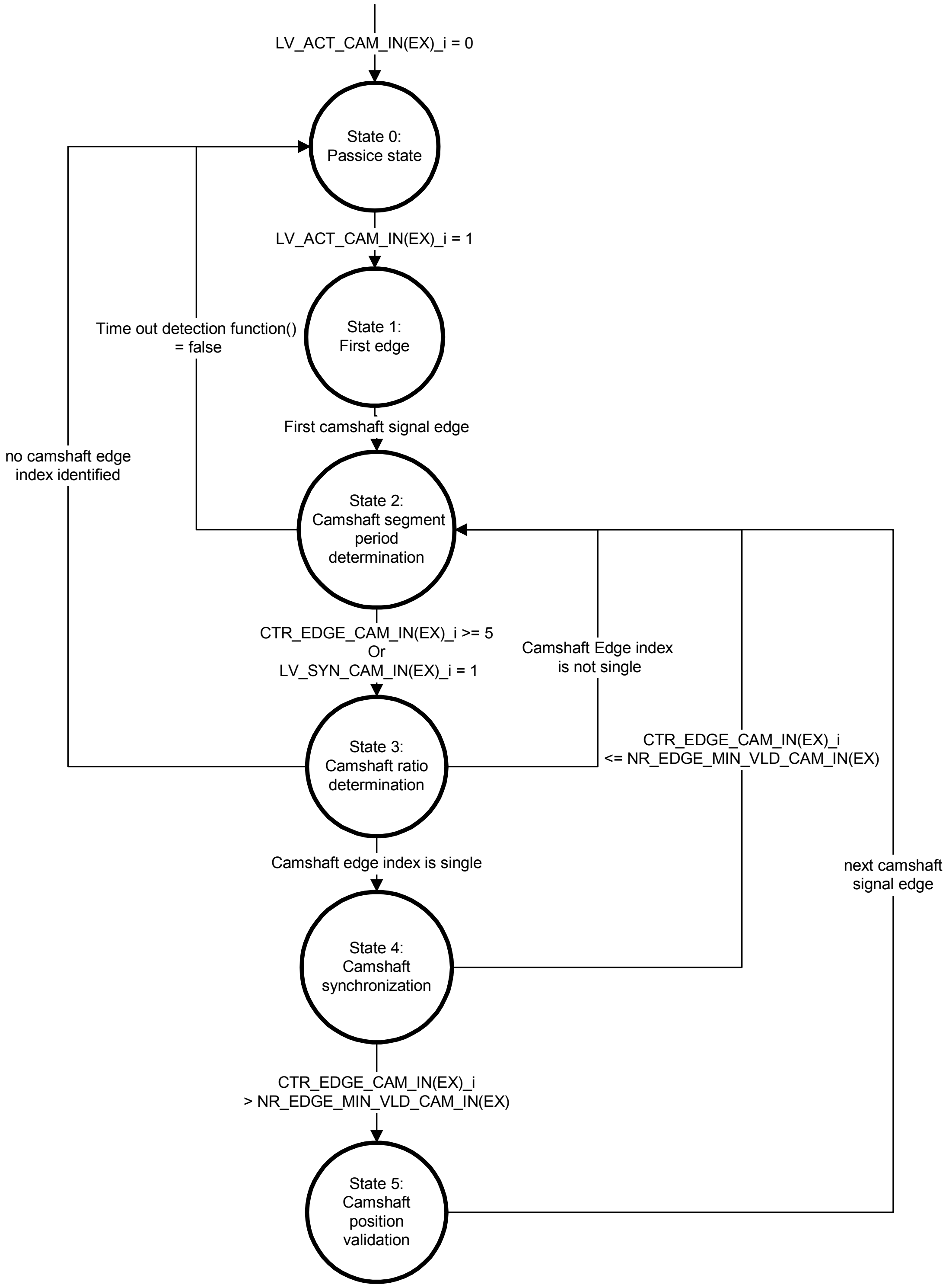


23.5.1 Camshaft Self-Synchronization

Application conditions:

Recurrence: every camshaft signal edge (falling and rising)

Signal flow diagram (ST_CAM_SS):



Formula section:

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN(EX)

State 0: Passive State

Input condition :

From EPM : LV_ACT_CAM_IN(EX)[i] = 0
From state 2 : Time out detection function () = false
From state 3 : No cam edge index identified

Output condition :

No output conditions

Action in the state :

LV_CAM_STOP_IN[i] = LV_CAM_STOP_EX[i] = 1
T_SEG_CAM_IN[i] = NC_T_SEG_MAX_CAM_IN
T_SEG_CAM_EX[i] = NC_T_SEG_MAX_CAM_EX
LV_SYN_CAM_IN[i] = LV_SYN_CAM_EX[i] = 0
LV_VLD_PSN_CAM_IN[i] = LV_VLD_PSN_CAM_EX[i] = 0
CTR_EDGE_CAM_IN[i] = CTR_EDGE_CAM_EX[i] = 0
IDX_EDGE_CAM_IN[i] = IDX_EDGE_CAM_EX[i] = 1
RATIO_PER_CAM_IN(EX)[i] = 0
RATIO_PSN_EDGE_CAM_IN(EX)[z][i]= 0

Action in transient:

No actions

State 1: First edge

Input condition:

From state 0: LV_ACT_CAM_IN(EX)[i] = 1

Output condition:

To state 2 First camshaft signal edge

Action in the state:

LV_CAM_STOP_IN(EX)[i] = 0
LV_ORNG_PER_CAM_IN(EX)[i] = 0
LV_ORNG_RATIO_CAM_IN(EX)[i] = 0

<u>Level detection:</u> The indexes of the camshaft edge leading to the camshaft signal level will be determined (example: if the signal level is high, all indexes correponding to a rising signal edges).

Wait for first camshaft signal edge

Action in transient:

No actions

<u>State 2: Camshaft segment period determination</u>

Input condition :

From state 1	First camshaft signal edge occured
From state 3:	Cam Edge index is not single
From state 4:	CTR_EDGE_CAM_IN(EX)[i] <= C_NR_EDGE_MIN_VLD_CAM_IN(EX) And next camshaft signal edge
From state 5:	next camshaft signal edge

Output condition :

To state 0 :	Time out detection function () = False
To state 3 :	CTR_EDGE_CAM_IN(EX)[i] >= 5 Or LV_SYN_CAM_IN(EX)[i] = 1

Action in the state:

```
If    Invalid segment period detection function() = true
Then  LV_ORNG_PER_CAM_IN(EX)[i] = 1
Else  Activation of Time out detection function ()
      Increment CTR_EDGE_CAM_IN(EX)[i]
      Indexes of the possible cam edges table are
      incremented by one modulo
      NC_NR_EDGE_CAM_IN(EX).
Endif
```

Action in transient:

No actions

State 3: Camshaft ratio determination

Input condition :

From state 2:	CTR_EDGE_CAM_IN(EX)[i] >= 5 Or LV_SYN_CAM_IN(EX)[i] = 1

Output condition :

To state 0:	No Cam Edge index identified
To state 2:	Cam Edge index is not single
To state 4:	Cam Edge index is single

Action in the state :

Activation of Camshaft segment ratio calculation function ()

Activation of Camshaft edge recognition function ()

Action in transient :

To state 0:	LV_ORNG_RATIO_CAM_IN(EX)[i] = 1
To state 4:	LV_SYN_CAM_IN(EX)[i] = 1

State 4: Camshaft synchronization

Input condition :

From state 3:	Cam Edge index is single

Output condition :

To state 2:	CTR_EDGE_CAM_IN(EX)[i] <= C_NR_EDGE_MIN_VLD_CAM_IN(EX)
To state 5:	CTR_EDGE_CAM_IN(EX)[i] > C_NR_EDGE_MIN_VLD_CAM_IN(EX)

Action in the state:	IDX_EDGE_CAM_IN(EX)[i] = camshaft edge index Wait next camshaft signal edge.

Action in transient:

To state 5:	LV_VLD_PSN_CAM_IN(EX)[i] = 1

State 5: Validation of camshaft position

Input condition:

From state 4: CTR_EDGE_CAM_IN(EX)[i] > C_NR_EDGE_MIN_VLD_CAM_IN(EX)

Output condition:

To state 2: Next camshaft signal edge

Action in the state: Wait next camshaft signal edge.

Action in transient:

None

23.5.2 Camshaft/Crankshaft Synchronization

General information:

Cam/crk synchronization is set as soon as the engine position is identified:

- Reference gap detected and
- Engine position offset determined

The engine position offset (PSN_ENG_CRK_OFF) can be

- 360° or 720° for NC_NR_GAP = 1
- 180°, 360°, 540°, 720° for NC_NR_GAP = 2

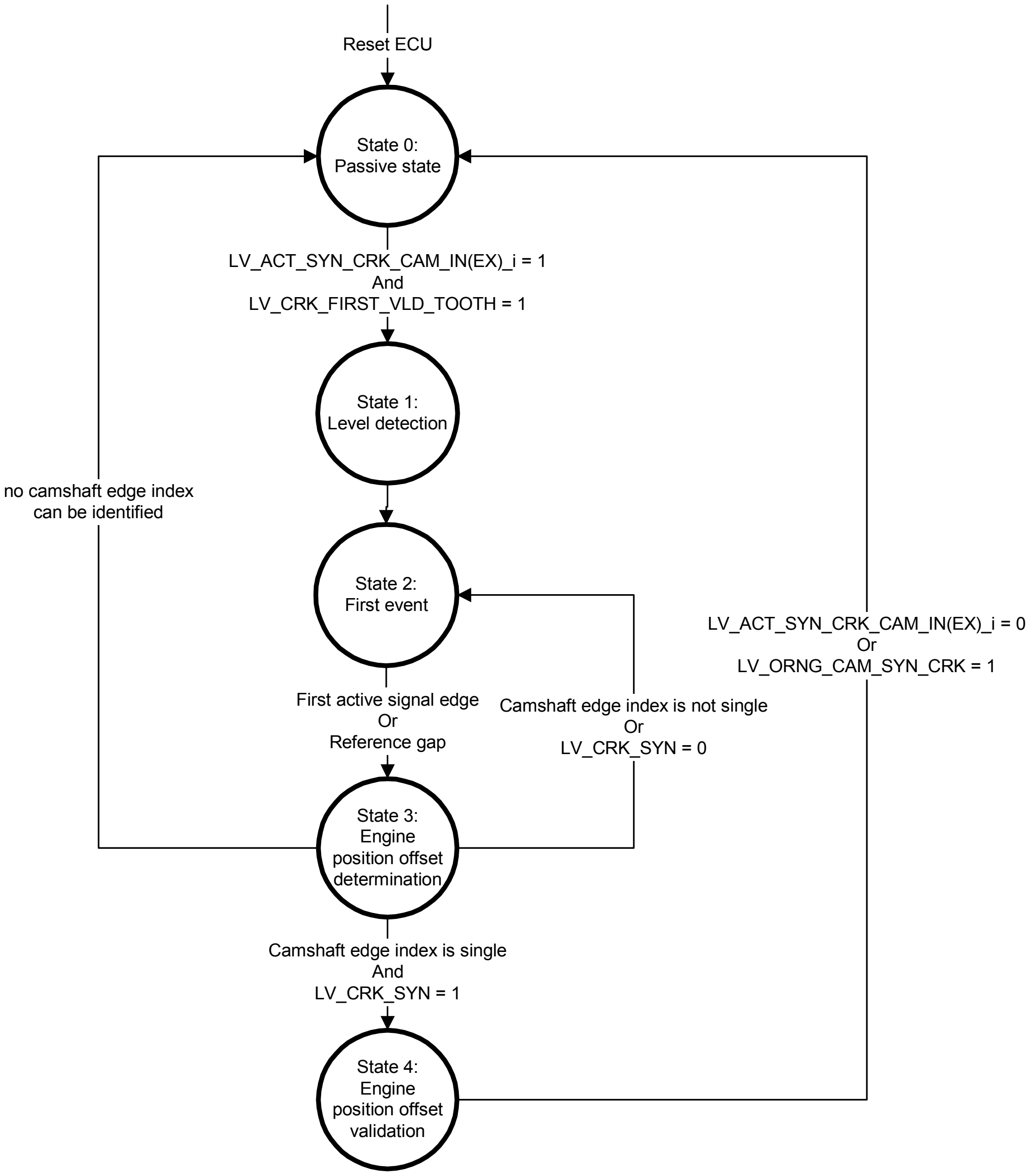
Application conditions:

Recurrence: every falling camshaft signal edge and every reference gap if NC_ACT_CAM_EDGE_SYN = 1

every rising camshaft signal edge and every reference gap if NC_ACT_CAM_EDGE_SYN = 2

every camshaft signal edge (rising and falling) and every reference gap if NC_ACT_CAM_EDGE_SYN = 3

Signal flow diagram ST_CAM_CRK_S:



Formula section:

State 0: Passive State

Input condition :

External event: Reset ECU

From State 4 : LV_ACT_SYN_CRK_CAM_IN(EX)[i] = 0

Or

LV_ORNG_CAM_SYN_CRK = 1

Output condition :

none

Action in the state:

PSN_ENG_CRK_OFS = 0° CRK

LV_CAM_SYN_CRK = 0

PSN_ENG_SYN_CAM_MIN = 0° CRK

PSN_ENG_SYN_CAM_MAX = 720° CRK

Activation of VVT lock check function ()

Action in transient :

none

State 1: Level Detection

Input condition:

From state 0 :

LV_ACT_SYN_CRK_CAM_IN(EX)[i] = 1

And

LV_CRK_FIRST_VLD_TOOTH = 1

Output condition :

To state 2:

At the end of the action in the state

Action in the state :

```
If      NC_ACT_CAM_EDGE_SYN = 1
Then    Index list = {1,3, ..., NC_NR_EDGE_CAM_IN(EX)-1}
        Keep only falling camshaft edges
        Keep initialization values for
        PSN_ENG_SYN_CAM_MIN(MAX)
Elseif  NC_ACT_CAM_EDGE_SYN = 2
Then    Index list = {2,4, ..., NC_NR_EDGE_CAM_IN(EX)}
        Keep only rising edges
```

Keep initialization values for PSN_ENG_SYN_CAM_MIN(MAX)

Else The indexes of the camshaft edge leading to the camshaft signal level will be determined (p.e. if the signal level is high, all indexes correponding to a rising signal edges).

Keep initialization values for PSN_ENG_SYN_CAM_MIN(MAX)

Endif

Action in transient :

To state 2: LV_ORNG_CAM_SYN_CRK = 0

State 2: First event

Input condition :

From state 1: At the end of the action in the state 1

From state 3: Camshaft edge index is not single

Or

LV_CRK_SYN = 0

Output condition :

To state 3: camshaft signal active edge for synchronization

Or

crankshaft reference gap

Action in the state :

Wait camshaft signal active edge or reference gap

Action in transient :

none

State 3: Engine Position Offset Determination

Input condition :

From state 3: camshaft signal active edge for synchronization

Or

crankshaft reference gap

Output condition :

To state 4 :	Camshaft signal edge index is single And LV_CRK_SYN = 1.
To state 2 :	Camshaft signal edge index is not single Or LV_CRK_SYN = 0
To state 0 :	If no cam edge index can be identified.

Action in the state :

Activation of camshaft edge index determination function ()

Action in transient :

To state 4 :	LV_CAM_SYN_CRK = 1
To state 0 :	LV_ORNG_CAM_SYN_CRK = 1

State 4: Engine Position Offset Validation

Input condition :

To state 4 :	Camshaft signal edge index is single And LV_CRK_SYN = 1.

Output condition :

To state 0 :	LV_ACT_SYN_CRK_CAM_IN(EX)[i] = 0 Or LV_ORNG_CAM_SYN_CRK = 1

Action in the state :

Activation of Cam/crk synchronization validation function()

```
If      Cam/crk synchronization validation function() = false
Then    LV_ORNG_CAM_SYN_CRK = 1
Endif
```

Action in transient :

None

23.5.3 Engine Position Interface for Pre-Injection

General information:

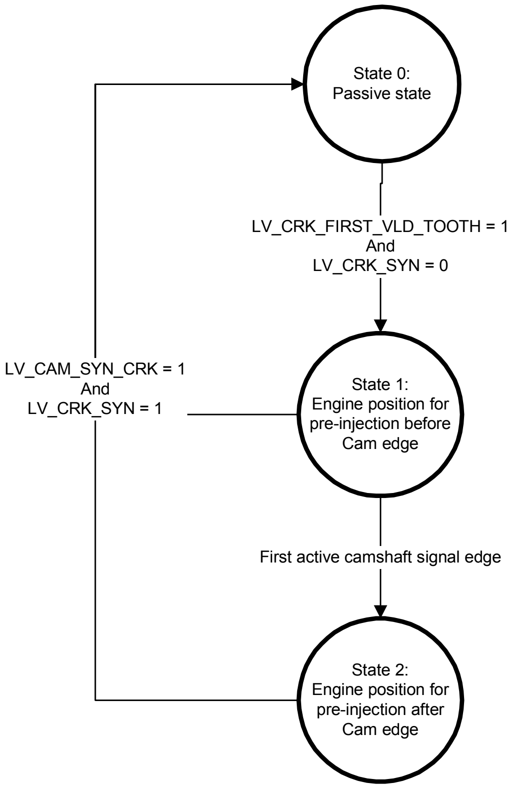
For rapid start of port-injection engines it is useful to start injection before synchronization on crankshaft signal is achieved. An approximative information about engine position is necessary for phasing the injection in a way to avoid emissions increase.

For this purpose, a minimum and the maximum engine position is determined according to the actual possible camshaft signal edge indexes found by camshaft signal acquisition for crankshaft synchronization. The minimum and the maximum position will converge gradually during the synchronization process.

Application conditions:

Recurrence:	every falling camshaft signal edge and every reference gap if NC_ACT_CAM_EDGE_SYN = 1
	every rising camshaft signal edge and every reference gap if NC_ACT_CAM_EDGE_SYN = 2
	every camshaft signal edge (rising and falling) and every reference gap if NC_ACT_CAM_EDGE_SYN = 3

Signal flow diagram (ST_CAM_PRE_INJ_S):



Formula section:

State 0: Passive State

Input condition :

From state 1,2 : LV_CRK_SYN = 1
And
LV_CAM_SYN_CRK = 1

Output condition :

To state 1: LV_CRK_FIRST_VLD_TOOTH = 1
And
LV_CRK_SYN = 0

Action in the state :

None

Action in transient :

None

State 1: Engine position for pre-injection before cam edge state

Input condition :

From state 1: End of the state 1

Output condition :

To state 2 : camshaft signal active edge for synchronization
To state 0 : LV_CRK_SYN = 1 and LV_CAM_SYN_CRK = 1

Action in the state :

Activation of Pre-injection interface computation before cam edge function ()

Action in transient :

To state 0 : PSN_ENG_SYN_CAM_MIN
= PSN_ENG_SYN_CAM_MAX
= PSN_ENG_CRK

State 2: Engine position for pre-injection after cam edge state

Input condition :

From state 1: camshaft signal active edge for synchronization

Output condition :

To state 0 : LV_CRK_SYN = 1 and LV_CAM_SYN_CRK = 1

Action in the state :

Activation of Pre-injection interface computation after Cam edge function ()

Action in transient :

To state 0 : PSN_ENG_SYN_CAM_MIN
= PSN_ENG_SYN_CAM_MAX
= PSN_ENG_CRK

23.5.4 Definition of the sub function tasks

23.5.4.1 Time-out detection function ()

General information:

Time-out is detected when the segment period exceeds NC_T_SEG_MAX_CAM_IN(EX). Time out detection is used for engine stalling detection during crankshaft limp-home.

Formula section:

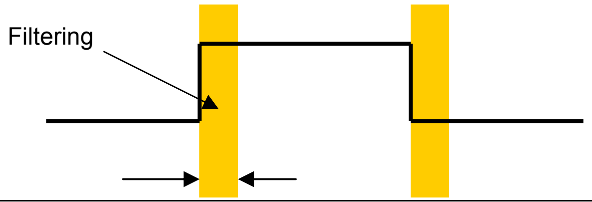
```
If      T_SEG_CAM_IN(EX)[i] <= NC_T_SEG_MAX_CAM_IN(EX)
Then    return (true)
Else    return(false)
Endif
```

23.5.4.2 Invalid segment period detection function ()

i = 1...NC_NR_CAM_CBK

General information:

Any signal edge detected before a delay time NC_T_SEG_MIN_CAM_IN(EX) following the previous signal edge will not produce any output.



Formula section:

```
If      T_SEG_CAM_IN(EX)[i] < NC_T_SEG_MIN_CAM_IN(EX)
Then    Invalid segment period
        return (true)
Else    return (false)
Endif
```

23.5.4.3 Camshaft segment ratio calculation function ()

General information:

The purpose of this function is to calculate a camshaft edge ratio from measured segment periods (measured ratio) and from designed edge positions (theoretical ratio).

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN(EX)

Application conditions:

Recurrence : every camshaft edge.

Formula section:

Measured ratio :

RATIO_PER_CAM_IN(EX)[i] =

$(T\_SEG\_CAM\_IN(EX)[i]_{n} + T\_SEG\_CAM\_IN(EX)[i]_{n-3})$ /

$(T\_SEG\_CAM\_IN(EX)[i]_{n-1} + T\_SEG\_CAM\_IN(EX)[i]_{n-2})$

Theoretical ratio :

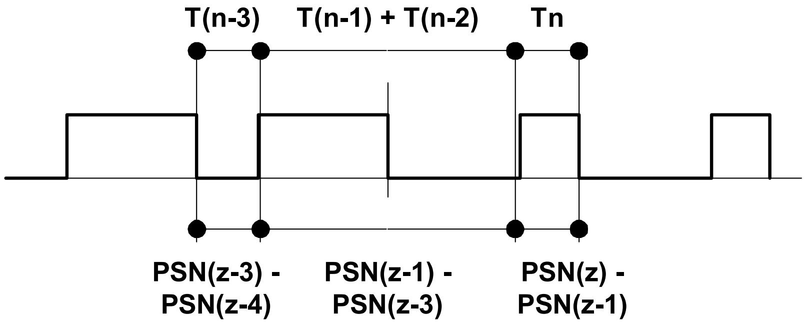
For all remaining indexes z:

RATIO_PSN_EDGE_CAM_IN(EX)[z][i] =

(NC_PSN_EDGE_CAM_IN(EX)[z][i] - NC_PSN_EDGE_CAM_IN(EX)[z-1][i]

+ NC_PSN_EDGE_CAM_IN(EX)[z-3][i] - NC_PSN_EDGE_CAM_IN(EX)[z-4][i]) /

(NC_PSN_EDGE_CAM_IN(EX)[z-1][i] - NC_PSN_EDGE_CAM_IN(EX)[z-3][i])



23.5.4.4 Camshaft edge recognition ()

General information:

The purpose of this function is to compare the measured camshaft edge ratio to the theoretical camshaft edge ratio. The camshaft edge is removed from the camshaft index index table if both ratios (theoretical and measured) are different more than ID_FAC_CAM_IN(EX).

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN(EX)

Application conditions:

Recurrence: every camshaft edge.

Formula section:

For all remaining indexes z:

```
If      RATIO_PSN_EDGE_CAM_IN(EX)[z][i] * ID_FAC_CAM_IN(EX) >
        RATIO_PER_CAM_IN(EX)[i] > RATIO_PSN_EDGE_CAM_IN(EX)[z][i] /
        ID_FAC_CAM_IN(EX)
Then    keep index z in the table
Else    eliminate index z from the table
Endif
```

23.5.4.5 Camshaft edge index determination function ()

General information:

This function determines the possible camshaft signal indexes and returns the engine position offset when the index is identified.

This function is used for

- determination of position offset for engine synchronization
- determination of min/max engine position for pre-injection

The actions and tests of this function shall be performed for all remaining indexes of the list of possible camshaft signal indexes.

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN(EX)

Application conditions:

Recurrence: every falling camshaft signal edge and every reference gap if NC_ACT_CAM_EDGE_SYN = 1

every rising camshaft signal edge and every reference gap if NC_ACT_CAM_EDGE_SYN = 2

every camshaft signal edge (rising and falling) and every reference gap if NC_ACT_CAM_EDGE_SYN = 3

Formula section:

```
If      active camshaft signal edge
Then    If      NC_ACT_CAM_EDGE_SYN = 3
        Then    Increment indexes in the list of possible camshaft signal index by 1
                modulo NC_NR_EDGE_CAM_IN(EX)
        Else    Increment by 2 indexes in the list of possible camshaft signal index by 2
                modulo NC_NR_EDGE_CAM_IN(EX)
Endif
```

Calculate theoretical angular distance between crankshaft tooth #1 and camshaft edge #z:

PSN_TOOTH1_CAM = NC_OFS_TDC0_REF_CRK +
NC_PSN_EDGE_CAM_IN(EX)[z][i]

Calculate theoretical angular distance between two successive active camshaft edges:

PSN_CAM_CAM is the distance between the current camshaft edge and the previous one, whereas PSN_CAM_CAM_1 is the distance between the current camshaft edge and the next one. PSN_CAM_CAM_1 is used in Test 10 in case of [CAM – CRK-Angle].

```
If      NC_ACT_CAM_EDGE_SYN = 3
Then    Falling and rising camshaft edges
        PSN_CAM_CAM = NC_PSN_EDGE_CAM_IN(EX)[z][i] –
                NC_PSN_EDGE_CAM_IN(EX)[z-1][i]
        PSN_CAM_CAM_1 = NC_PSN_EDGE_CAM_IN(EX)[z+1][i] –
                NC_PSN_EDGE_CAM_IN(EX)[z][i]

Else    Only Falling or only rising camshaft edges
        PSN_CAM_CAM = NC_PSN_EDGE_CAM_IN(EX)[z][i] –
                NC_PSN_EDGE_CAM_IN(EX)[z-2][i]
        PSN_CAM_CAM_1 = NC_PSN_EDGE_CAM_IN(EX)[z+2][i] –
                NC_PSN_EDGE_CAM_IN(EX)[z][i]

Endif
```

The number of possible indexes z will be reduced by checking the conditions described below.

The following table shows the actions to perform in function of the current event (n), previous event (n-1) and last event (n-2).

The event is a detection of a active camshaft edge (**CAM**) for synchronization or a detection of reference gap (**GAP**). If no event took place before (**none**), the tests are executed with the measured angular distance between engine start and current event.

To reduce the possible engine position range for pre-injection, the event could also be a measured crankshaft angle (**CRK-Angle**) from the last CAM event. The event should occur when :

REL_ANG_CRK_CAM = C_CRK_ANG_DLY_LST_CAM.

Note that two calibration datas are defined :

- C_CRK_ANG_DLY is the one used in case of [CRK-Angle – CAM] in Test 9 ;
- C_CRK_ANG_DLY_LST_CAM is the one used in case of [CAM - CRK-Angle] in Test 10.

This pre-injection improvement can be disabled by setting these calibration datas to 0°CRK.

Event (n)	CAM	CAM	GAP	GAP	CAM	CAM	GAP	CRK-Angle
Event (n-1)	None	CAM	None	CAM	GAP	GAP	GAP	CAM
Event (n-2)	-	-	-	-	None	CAM	-	-
Test	Test 1 Test 9	Test 2 Test 11	Test 3 Test 12	Test 4 Test 12	Test 1 Test 5 (1)	Test 2 Test 5 (1)	(2)	Test 10 (3)

(1) The first test shall be executed with event **n** and **n-2**. The second test shall be executed with event **n** and **n-1**

(2) No Camshaft index can be identified (All remaining indexes shall be removed)

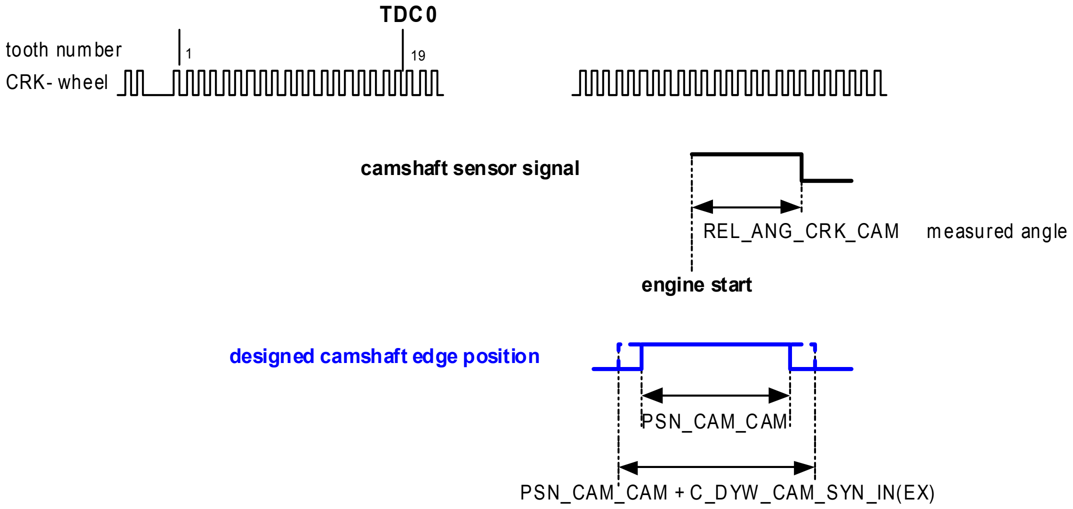
(3) CRK-Angle event will never be in position (n-1) as it will be replaced in the memory by the next event instead to be recorded in (n-1) position

NB1: The calculation of the CRK-Angle delay begins at each CAM edge, but is also reset in case of GAP event. No test should be performed in case of GAP –- CRK-Angle.

NB2: In all the following calculations, PSN_TOOTH1_CAM represents PSN_TOOTH1_CAM[z] (= PSN_TOOTH1_CAM for the camshaft edge z). If an other camshaft edge is used for computation, then PSN_TOOTH1_CAM[z] is mentioned.

Test 1:

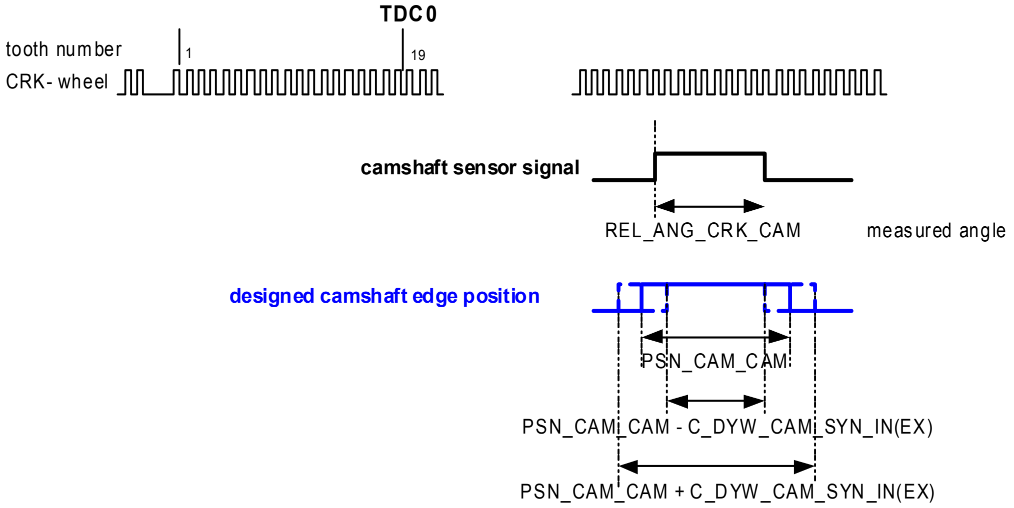
Active camshaft edge detected. No active camshaft edge occured before.



If	REL_ANG_CRK_CAM < PSN_CAM_CAM + CAM_DYW_SYN_IN(EX)
Then	(no operation)
Else	Remove index z from list of possible camshaft signal edge indexes
Endif	

Test 2:

Active camshaft edge detected. Last event was also an active camshaft edge.

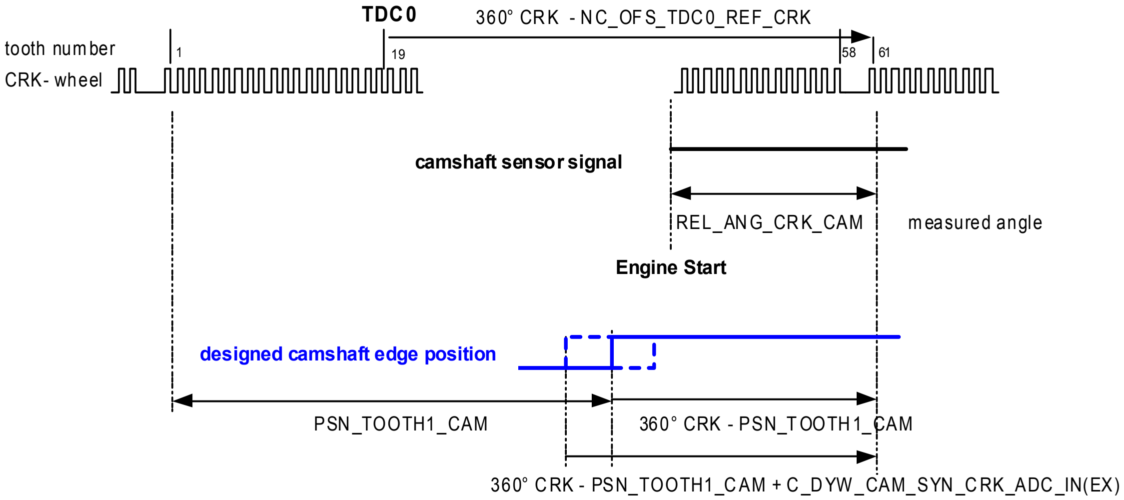


PSN_GAP = engine position offset for camshaft edge #z

```
If      PSN_CAM_CAM – CAM_DYW_SYN_IN(EX)
        < REL_ANG_CRK_CAM <
        PSN_CAM_CAM + CAM_DYW_SYN_IN(EX)
Then    PSN_ENG_CRK_OFS = PSN_GAP – 360° CRK
Else    Remove index z from list of possible camshaft signal edge indexes
Endif
```

Test 3:

First reference gap detected. No active camshaft edge occured before.



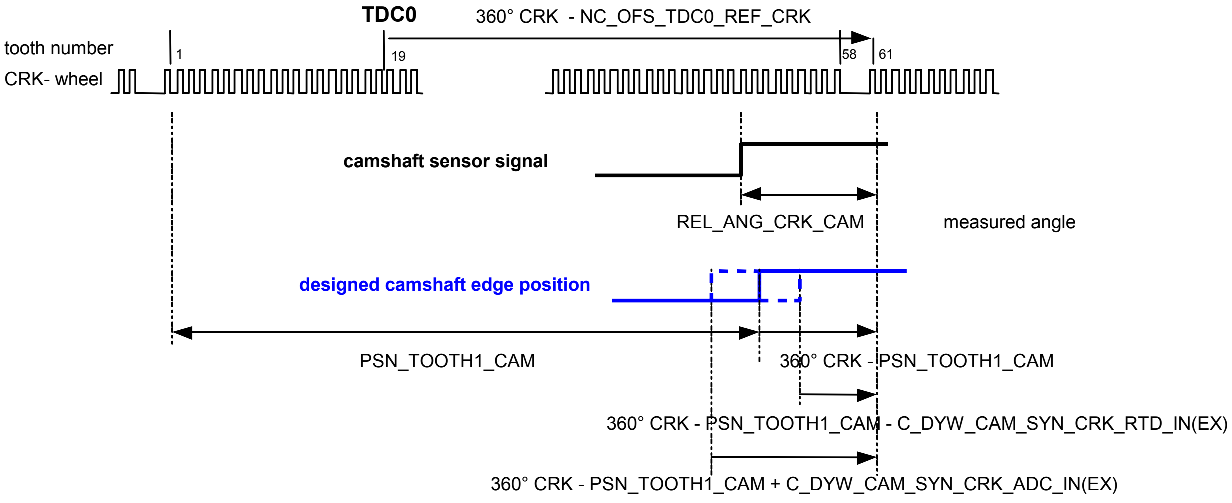
PSN_GAP = engine position offset for camshaft edge #z + 360° CRK / NC_NR_GAP

```
If      REL_ANG_CRK_CAM <
        PSN_GAP - PSN_TOOTH1_CAM +
        CAM_DYW_CRK_SYN_ADC_IN(EX)
        And
        REL_ANG_CRK_CAM < 360° CRK / NC_NR_GAP
Then      Engine position is PSN_GAP - NC_OFS_TDC0_REF_CRK
         PSN_ENG_CRK_OFS = PSN_GAP – 360° CRK
Else
        If      The camshaft edge can be found on both sides of
                the reference gap, it is not removed
                PSN_GAP - PSN_TOOTH1_CAM <
                CAM_DYW_CRK_SYN_RTD_IN(EX)
        Then
```

```
                Keep index z
            Else
                Remove index z from list of possible camshaft
                signal edge indexes
            Endif
    Endif
```

Test 4:

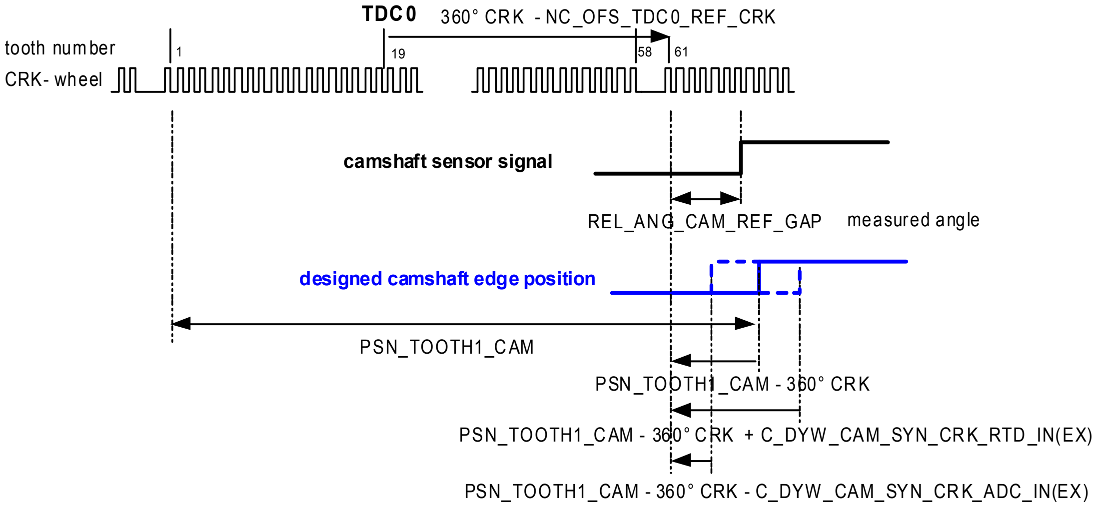
Reference gap detected. Last event was an active camshaft edge.



```
PSN_GAP = engine position offset for camshaft edge #z + 360° CRK / NC_NR_GAP
    If      PSN_GAP - PSN_TOOTH1_CAM –
            CAM_DYW_CRK_SYN_RTD_IN(EX)
            < REL_ANG_CRK_CAM <
            PSN_GAP - PSN_TOOTH1_CAM +
            CAM_DYW_CRK_SYN_ADC_IN(EX)
            And
            REL_ANG_CRK_CAM < 360° CRK / NC_NR_GAP
    Then
              Engine position is PSN_GAP - NC_OFS_TDC0_REF_CRK
             PSN_ENG_CRK_OFS = PSN_GAP – 360° CRK
    Else
            Remove index z from list of possible camshaft signal edge
            indexes
    Endif
```

Test 5:

Active camshaft edge detected. Last event was reference gap.



PSN_GAP = engine position offset for camshaft edge #z

If PSN_TOOTH1_CAM - PSN_GAP – CAM_DYW_CRK_SYN_ADC_IN(EX)
< REL_ANG_CAM_REF_GAP <
PSN_TOOTH1_CAM - PSN_GAP + CAM_DYW_CRK_SYN_RTD_IN(EX)

And

REL_ANG_CAM_REF_GAP < 360° CRK / NC_NR_GAP

Then *Engine position is PSN_GAP - NC_OFS_TDC0_REF_CRK*

PSN_ENG_CRK_OFS = PSN_GAP – 360° CRK

Else

Remove index z from list of possible camshaft signal edge indexes

Endif

Test 9:

Active camshaft edge detected. No active camshaft edge occured before.

This test is executed after test 1 only if no GAP occurred before CAM edge.

If an angular distance greater than the crank angle delay calibration has been measured without any new event, this test allows to eliminate the corresponding camshaft edges that should occur.

If NC_ACT_CAM_EDGE_SYN = 3 (both falling and rising edges used)

And C_CRK_ANG_DLY > 0

And REL_ANG_CRK_CAM >= C_CRK_ANG_DLY

```
                    And the cam edge z can not be seen on both sides of the reference gap
                    ((PSN_TOOTH1_CAM[z] - C_DYW_CAM_SYN_CRK_ADC_IN(EX)) modulo
                    (360/NC_NR_GAP)) < ((PSN_TOOTH1_CAM[z] +
                    C_DYW_CAM_SYN_CRK_RTD_IN(EX)) modulo (360/NC_NR_GAP))
                    And the cam edge z-1 can not be seen on both sides of the reference gap
                    ((PSN_TOOTH1_CAM[z-1] - C_DYW_CAM_SYN_CRK_ADC_IN(EX))
                    modulo (360/NC_NR_GAP)) < ((PSN_TOOTH1_CAM[z-1] +
                    C_DYW_CAM_SYN_CRK_RTD_IN(EX)) modulo (360/NC_NR_GAP))
        Then
                    If     REL_ANG_CRK_CAM <
                           PSN_TOOTH1_CAM modulo (360/NC_NR_GAP) +
                           CAM_DYW_CRK_SYN_RTD_IN(EX)
                    Then
                           Keep index z
                           Engine position range will be calculated in Pre-injection interface
                           computation after Cam edge function (Pre-inj action 2)
                    Else
                           Remove index z from list of possible camshaft signal edge indexes
                    Endif
        Else        (no operation)
        Endif
```

Test 10:

Crankshaft angle event detected. An active camshaft edge occurred before.

```
    If      C_CRK_ANG_DLY_LST_CAM > 0
    Then
        If          REL_ANG_CRK_CAM <
                    PSN_CAM_CAM_1 + CAM_DYW_SYN_IN(EX)

        Then        (No operation)
        Else        Remove index z from list of possible camshaft signal edge indexes
        Endif
```

On the remaining cam edges, after the first test has been performed on all edges:

PSN_GAP = engine position offset for camshaft edge #z

```
          If        The cam edge z can not be seen on both sides of the reference gap
                    ((PSN_TOOTH1_CAM[z] - C_DYW_CAM_SYN_CRK_ADC_IN(EX)) modulo
                    (360/NC_NR_GAP)) < ((PSN_TOOTH1_CAM[z] +
                    C_DYW_CAM_SYN_CRK_RTD_IN(EX)) modulo (360/NC_NR_GAP))
          Then
                    If      REL_ANG_CRK_CAM <
                            (720 - PSN_TOOTH1_CAM) modulo (360/NC_NR_GAP) +
                            CAM_DYW_CRK_SYN_ADC_IN(EX)
                    Then    Keep index z
                            Engine position range will be calculated in Pre-injection interface
                            computation after Cam edge function (Pre-inj action 5)
                            Engine position is PSN_GAP - NC_OFS_TDC0_REF_CRK
                            PSN_ENG_CRK_OFS = PSN_GAP – 360° CRK
                    Else
                            Remove index z from list of possible camshaft signal edge indexes
                    Endif
          Else      No operation
          Endif
Else          No operation
Endif
```

Test 11:

Active camshaft edge detected. Last event was also an active camshaft edge.

This test is done after test 2. So we know that any Gap is between the two camshaft edge events. We can remove all indexes where a Gap is located between those and the previous cam edge.

On the remaining cam edges, after the first test has been performed on all edges:

```
          If        NC_ACT_CAM_EDGE_SYN = 3 (both falling and rising edges used)
                    And
                    The cam edge z can not be seen on both sides of the reference gap
                    ((PSN_TOOTH1_CAM[z] - C_DYW_CAM_SYN_CRK_ADC_IN(EX)) modulo
                    (360/NC_NR_GAP)) < ((PSN_TOOTH1_CAM[z] +
                    C_DYW_CAM_SYN_CRK_RTD_IN(EX)) modulo (360/NC_NR_GAP))
                    And
                    The cam edge z-1 can not be seen on both sides of the reference gap
```

```
            ((PSN_TOOTH1_CAM[z-1] - C_DYW_CAM_SYN_CRK_ADC_IN(EX))
            modulo (360/NC_NR_GAP)) < ((PSN_TOOTH1_CAM[z-1] +
            C_DYW_CAM_SYN_CRK_RTD_IN(EX)) modulo (360/NC_NR_GAP))
    Then
            If          PSN_TOOTH1_CAM[z-1] > PSN_TOOTH1_CAM[z]
                        Or
                        PSN_TOOTH1_CAM[z-1] < 360/NC_NR_GAP <
                        PSN_TOOTH1_CAM[z]
                        Or
                        PSN_TOOTH1_CAM[z-1] < 360 < PSN_TOOTH1_CAM[z]
                        Or
                        PSN_TOOTH1_CAM[z-1] < 360*(2-1/NC_NR_GAP) <
                        PSN_TOOTH1_CAM[z]
            Then
                        Remove index z
            Else
                        Keep index z
            Endif
    Endif
```

Test 12:

This test is executed on Gap after the test 3 or 4.

On the remaining cam edges, has been performed on all edges:

```
            If          NC_ACT_CAM_EDGE_SYN = 3 (both falling and rising edges
                        used)
                        And
                        The cam edge z can not be seen on both sides of the reference
                        gap
                        ((PSN_TOOTH1_CAM[z] -
                        C_DYW_CAM_SYN_CRK_ADC_IN(EX)) modulo
                        (360/NC_NR_GAP)) < ((PSN_TOOTH1_CAM[z] +
                        C_DYW_CAM_SYN_CRK_RTD_IN(EX)) modulo
                        (360/NC_NR_GAP))
                        And
                        The cam edge z+1 can not be seen on both sides of the
                        reference gap
                        ((PSN_TOOTH1_CAM[z+1] -
                        C_DYW_CAM_SYN_CRK_ADC_IN(EX)) modulo
```

```
          (360/NC_NR_GAP)) < ((PSN_TOOTH1_CAM[z+1] +
          C_DYW_CAM_SYN_CRK_RTD_IN(EX)) modulo
          (360/NC_NR_GAP))
Then
          If        PSN_TOOTH1_CAM[z] >
                    PSN_TOOTH1_CAM[z+1]
                    Or
                    PSN_TOOTH1_CAM[z] < 360/NC_NR_GAP <
                    PSN_TOOTH1_CAM[z+1]
                    Or
                    PSN_TOOTH1_CAM[z] < 360 <
                    PSN_TOOTH1_CAM[z+1]
                    Or
                    PSN_TOOTH1_CAM[z] < 360*(2-
                    1/NC_NR_GAP) < PSN_TOOTH1_CAM[z+1]
          Then
                    Keep index z
          Else
                    Remove index z
          Endif
Endif
```

23.5.4.6 Cam/crk synchronization validation function ()

General information:

The purpose of this function is to check the angular distance between cam/crk events. The function returns false is the active camshaft signal edge z is outside the defined tolerance window.

i = 1...NC_NR_CAM_CBK   z = 0...NC_NR_EDGE_CAM_IN(EX)

Application conditions:

Recurrence:	every falling camshaft signal edge and every reference gap if NC_ACT_CAM_EDGE_SYN = 1
	every rising camshaft signal edge and every reference gap if NC_ACT_CAM_EDGE_SYN = 2
	every camshaft signal edge (rising and falling) and every reference gap if NC_ACT_CAM_EDGE_SYN = 3

Formula section:

```
If      active camshaft signal edge
Then    If      NC_ACT_CAM_EDGE_SYN = 3
        Then    Increment indexes in the list of possible camshaft signal index by 1
                modulo NC_NR_EDGE_CAM_IN(EX)
        Else    Increment by 2 indexes in the list of possible camshaft signal index by 2
                modulo NC_NR_EDGE_CAM_IN(EX)
Endif
```

Calculate theoretical angular distance between crankshaft tooth #1 and camshaft edge #z:

```
    PSN_TOOTH1_CAM = NC_OFS_TDC0_REF_CRK +
            NC_PSN_EDGE_CAM_IN(EX)[z][i]
```

Calculate theoretical angular distance between two successive active camshaft edges:

```
If      NC_ACT_CAM_EDGE_SYN = 3
Then    Falling and rising camshaft edges
        PSN_CAM_CAM = NC_PSN_EDGE_CAM_IN(EX)[z][i] –
                NC_PSN_EDGE_CAM_IN(EX)[z-1][i]
Else    Only Falling or only rising camshaft edges
        PSN_CAM_CAM = NC_PSN_EDGE_CAM_IN(EX)[z][i] –
                NC_PSN_EDGE_CAM_IN(EX)[z-2][i]
Endif
```

The following table shows the actions to perform in function of the current (n) and last event (n-1).

The event is a detection of a active camshaft signal edge for synchronization (CAM) or a detection of reference gap (GAP). Tests and edge elimination shall only be performed on active camshaft signal edges with current value of REL_ANG_CRK_CAM.

Event (n)	CAM	GAP	CAM	GAP
Event (n-1)	CAM	CAM	GAP	GAP
Test	Test 6	Test 7	Test 8	(*)

(*) No Camshaft index can be identified (All remaining indexes shall be removed)

Test 6:

Active camshaft edge detected. Last event was also a active camshaft edge.

```
If      PSN_CAM_CAM – CAM_DYW_SYN_IN(EX)
        < REL_ANG_CRK_CAM <
        PSN_CAM_CAM + CAM_DYW_SYN_IN(EX)
Then    camshaft signal edge is valid
        return (true)
Else    Invalid camshaft signal edge, synchronization validation failed
        return (false)
Endif
```

Test 7:

Reference gap detected. Last event was a active camshaft signal edge.

PSN_GAP = current engine position offset

```
If      PSN_GAP - PSN_TOOTH1_CAM – CAM_DYW_CRK_SYN_RTD_IN(EX)
        < REL_ANG_CRK_CAM <
        PSN_GAP - PSN_TOOTH1_CAM + CAM_DYW_CRK_SYN_ADC_IN(EX)
        And
        REL_ANG_CRK_CAM <= 360° CRK / NC_NR_GAP
        Then    camshaft signal edge is valid
                return (true)
        Else    Invalid camshaft signal edge
                return (false)
```

```
Endif
```

Test 8:

Active camshaft signal edge detected. Last event was a reference gap.

```
PSN_GAP = current engine position offset
If      PSN_TOOTH1_CAM - PSN_GAP – CAM_DYW_CRK_SYN_ADC_IN(EX)
        < REL_ANG_CAM_REF_GAP <
        PSN_TOOTH1_CAM - PSN_GAP + CAM_DYW_CRK_SYN_RTD_IN(EX)
        And
        REL_ANG_CAM_REF_GAP <= 360° CRK / NC_NR_GAP
        Then    camshaft signal edge is valid
                return (true)
        Else    Invalid camshaft signal edge
                return (false)
Endif
```

23.5.4.7 Pre-injection interface computation before Cam edge function ()

General information:

A truth table is used in order to define the possible engine position.

Application conditions:

Recurrence : every crankshaft edge (CRK) or every reference gap (GAP)

Formula section:

Event (n)	CRK	GAP
Pre-inj Action	1	2

Pre-inj Action 1

PSN_ENG_SYN_CAM_MIN n = **PSN_ENG_SYN_CAM_MIN** n-1 + 360°/NC_NR_TOOTH

PSN_ENG_SYN_CAM_MAX n = **PSN_ENG_SYN_CAM_MAX** n-1

Pre-inj Action 2

PSN_ENG_SYN_CAM_MIN n = **PSN_ENG_SYN_CAM_MIN** n-1 + (NC_NR_TOOTH_GAP+1)* 360°/NC_NR_TOOTH

PSN_ENG_SYN_CAM_MAX n = **PSN_ENG_SYN_CAM_MAX** n-1

23.5.4.8 Pre-injection interface computation after Cam edge function ()

General information:

A truth table is used in order to define the possible engine position.

Application conditions:

Recurrence:	every falling camshaft signal edge (CAM), every reference gap (GAP) or every active crankshaft edge (CRK) if NC_ACT_CAM_EDGE_SYN = 1
	every rising camshaft signal (CAM), every reference gap (GAP) or every active crankshaft edge(CRK) if NC_ACT_CAM_EDGE_SYN = 2
	every falling and rising camshaft signal edge (CAM), every reference gap (GAP) or every active crankshaft edge (CRK) if NC_ACT_CAM_EDGE_SYN = 3

Formula section:

Event (n)	CRK	CRK	CAM	GAP	GAP	CRK-Angle
Event (n-1)	CRK	CAM	CRK	CRK	CAM	CRK
Pre-inj Action	1	1	2	3	4	5

Pre-inj Action 1

PSN_ENG_SYN_CAM_MIN n = **PSN_ENG_SYN_CAM_MIN** n-1 + 360°/NC_NR_TOOTH

PSN_ENG_SYN_CAM_MAX n = **PSN_ENG_SYN_CAM_MAX** n-1 + 360°/NC_NR_TOOTH

Pre-inj Action 2

PSN_ENG_SYN_CAM_MIN = NC_PSN_EDGE_CAM_IN(EX)[z][i] – CAM_DYW_SYN_IN(EX)

for the lowest possible signal edge index z

PSN_ENG_SYN_CAM_MAX = NC_PSN_EDGE_CAM_IN(EX)[z][i] + CAM_DYW_SYN_IN(EX)

for the highest possible signal edge index z

Note: The possible signal edges z are determined by checking the conditions of “Camshaft Edge Index determination function()”. This function must be executed before action 2.

Pre-inj Action 3

PSN_ENG_SYN_CAM_MIN n = **PSN_ENG_SYN_CAM_MIN** n-1 + (NC_NR_TOOTH_GAP+1)* 360°/NC_NR_TOOTH

PSN_ENG_SYN_CAM_MAX n = **PSN_ENG_SYN_CAM_MAX** n-1 + (NC_NR_TOOTH_GAP+1)* 360°/NC_NR_TOOTH

Pre-inj Action 4

A camshaft edge has occured during the reference gap (when the camshaft edge is detected Pre-inj Action 2 is performed and when the first crankshaft edge after the reference is detected Pre-inj Action 4 is performed)

PSN_ENG_SYN_CAM_MIN n = **PSN_ENG_SYN_CAM_MIN** n-1 + **REL_ANG_CRK_CAM**

PSN_ENG_SYN_CAM_MAX n = **PSN_ENG_SYN_CAM_MAX** n-1 + **REL_ANG_CRK_CAM**

Pre-inj Action 5

A crank-angle event has occurred. A precise calculation of the engine position for pre-injection is possible.

PSN_ENG_SYN_CAM_MIN = NC_PSN_EDGE_CAM_IN(EX)[z][i] – CAM_DYW_SYN_IN(EX) + REL_ANG_CRK_CAM

for the lowest possible signal edge index z

PSN_ENG_SYN_CAM_MAX = NC_PSN_EDGE_CAM_IN(EX)[z][i] + CAM_DYW_SYN_IN(EX) + REL_ANG_CRK_CAM

for the highest possible signal edge index z

Note: The possible signal edge z is determined by checking the conditions of “Camshaft Edge Index determination function()”. This function must be executed before action 5.

23.5.4.9 VVT lock check function ()

General information:

The purpose of this function is to check the VVT lock position flag and to extend the cam/crk tolerance windows if the VVT is not in locked position at engine start.

Application conditions:

Recurrence: at transition of LV_STOP_ENG from 0 to 1

Or

Reset ECU

Formula section:

If LV_CAM_LOCK_IVVT_IN(EX)[i] = 0 for the camshaft sensor selected for cam/crk synchronization

And

```
        NLC_IVVT_IN(EX) = 1
Then    VVT not in locked position at engine start
        CAM_DYW_SYN_IN = C_DYW_CAM_SYN_IN + C_CAM_ADJ_VVT_SYN_IN
        CAM_DYW_SYN_EX = C_DYW_CAM_SYN_EX + C_CAM_ADJ_VVT_SYN_EX
        CAM_DYW_CRK_SYN_ADC_IN = C_DYW_CAM_CRK_SYN_ADC_IN + C_CAM_ADJ_VVT_SYN_CRK_ADC_IN
        CAM_DYW_CRK_SYN_ADC_EX = C_DYW_CAM_CRK_SYN_ADC_EX
        CAM_DYW_CRK_SYN_RTD_IN = C_DYW_CAM_CRK_SYN_RTD_IN
        CAM_DYW_CRK_SYN_RTD_EX = C_DYW_CAM_CRK_SYN_RTD_EX + C_CAM_ADJ_VVT_SYN_CRK_RTD_EX
Else
        VVT in locked position at engine start or no VVT at all
        CAM_DYW_SYN_IN = C_DYW_CAM_SYN_IN
        CAM_DYW_SYN_EX = C_DYW_CAM_SYN_EX
        CAM_DYW_CRK_SYN_ADC_IN = C_DYW_CAM_CRK_SYN_ADC_IN
        CAM_DYW_CRK_SYN_ADC_EX = C_DYW_CAM_CRK_SYN_ADC_EX
        CAM_DYW_CRK_SYN_RTD_IN = C_DYW_CAM_CRK_SYN_RTD_IN
        CAM_DYW_CRK_SYN_RTD_EX = C_DYW_CAM_CRK_SYN_RTD_EX
Endif
```

23.5.5 Requirements to Infrastructure

23.5.5.1 Camshaft segment period measurement

General information:

System request accuracy <= 4 us

The camshaft segment period T_SEG_CAM_IN(EX)[i] is the measured time between two consecutive valid camshaft signal edges.

23.5.5.2 Measurement of angular distance to active camshaft edge

General information:

The purpose of this function is to provide a measurement for

- the angular distance between the first valid crankshaft teeth and the first active camshaft edge
- the angular distance between two consecutive valid active camshaft signal edges

Only active camshaft signal edges shall be processed. The active camshaft edge for synchronization is defined by NC_ACT_CAM_EDGE_SYN.

Application conditions:

Recurrence: every active camshaft edge and every active crankshaft edge.

Formula section:

```
If      LV_CRK_FIRST_VLD_TOOTH = 0
Then    REL_ANG_CRK_CAM = 0
Else    If      active camshaft signal edge
        Then    REL_ANG_CRK_CAM = 0
        Else    If      active crankshaft signal edge or after missing tooth simulation
                Then    REL_ANG_CRK_CAM = REL_ANG_CRK_CAM +
                                  (360° CRK  / NC_NR_TOOTH)
                Else    REL_ANG_CRK_CAM is frozen
                Endif
        Endif
Endif
```

23.5.5.3 Measurement of angular distance from reference gap to active camshaft edge

General information:

The purpose of this function is to provide a measurement for

- the angular distance between a crankshaft reference gap and a active camshaft signal edge

Application conditions:

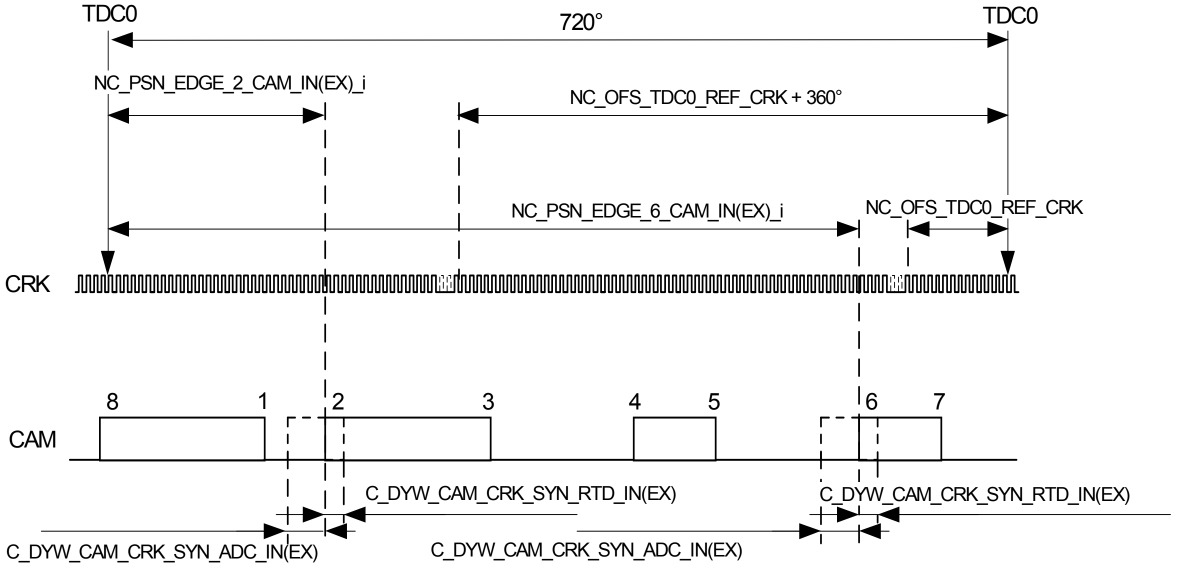
Recurrence: every active crankshaft signal edge

Formula section:

```
If      Reference Gap is detected
Then    REL_ANG_CAM_REF_GAP = 0
Else    If      crank active edge
        Then    REL_ANG_CAM_REF_GAP = REL_ANG_CAM_REF_GAP +
                                  (360° CRK / NC_NR_TOOTH)
        Else    REL_ANG_CAM_REF_GAP is frozen
        Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NR_EDGE_MIN_VLD_CAM_IN	1	0...FFH	0...255	1	[-]
Number of intake camshaft signal edges for valid position output					
C_NR_EDGE_MIN_VLD_CAM_EX	1	0...FFH	0...255	1	[-]
Number of exhaust camshaft signal edges for valid position output					
C_DYW_CAM_SYN_IN	1	0...1FFH	0...31.9375	0.0625	[°CRK]
Tolerance window for angle between intake camshaft signal edges in crankshaft synchronization mode					
C_DYW_CAM_SYN_EX	1	0...1FFH	0...31.9375	0.0625	[°CRK]
Tolerance window for angle between exhaust camshaft signal edges in crankshaft synchronization mode					
C_DYW_CAM_CRK_SYN_ADC_IN	1	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window advance for intake camshaft to crankshaft reference in crankshaft synchronization mode					
C_DYW_CAM_CRK_SYN_ADC_EX	1	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window advance for exhaust camshaft to crankshaft reference in crankshaft synchronization mode					
C_DYW_CAM_CRK_SYN_RTD_IN	1	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window retard for intake camshaft to crankshaft reference in crankshaft synchronization mode					
C_DYW_CAM_CRK_SYN_RTD_EX	1	0...7FFH	0...127.9375	0.0625	[°CRK]
Tolerance window retard for exhaust camshaft to crankshaft reference in crankshaft synchronization mode					
C_CAM_ADJ_VVT_SYN_IN	1	0...7FFH	0...127.9375	0.0625	[°CRK]
Extension of tolerance window CAM_DYW_SYN_IN when VVT is not in locked position at start					
C_CAM_ADJ_VVT_SYN_EX	1	0...7FFH	0...127.9375	0.0625	[°CRK]
Extension of tolerance window CAM_DYW_SYN_EX when VVT is not in locked position at start					
C_CAM_ADJ_VVT_SYN_CRK_ADC_IN	1	0...7FFH	0...127.9375	0.0625	[°CRK]
Extension of tolerance window CAM_DYW_CRK_SYN_ADC_IN when VVT is not in locked position at start					
C_CAM_ADJ_VVT_SYN_CRK_RTD_EX	1	0...7FFH	0...127.9375	0.0625	[°CRK]
Extension of tolerance window CAM_DYW_CRK_SYN_RTD_EX when VVT is not in locked position at start					
C_CRK_ANG_DLY_LST_CAM	1	0...2CFFH	0...719.9375	0.0625	[°CRK]
Crankshaft angle threshold after last cam edge to reduce the possible engine position range for pre-injection					
C_CRK_ANG_DLY	1	0...2CFFH	0...719.9375	0.0625	[°CRK]
Crankshaft angle threshold after start of cranking to reduce the possible engine position range for pre-injection					
ID_FAC_CAM_IN	16	10...FFH	1...15.9375	0.0625	[-]
LDPM_N_32_1_ENSD	16	0...FFH	0...8160	32	[rpm]
Period ratio tolerance factor for intake camshaft					
ID_FAC_CAM_EX	16	10...FFH	1...15.9375	0.0625	[-]
LDPM_N_32_1_ENSD	16	0...FFH	0...8160	32	[rpm]
Period ratio tolerance factor for exhaust camshaft					



Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_T_SEG_MIN_CAM_IN	-	0...FFFFFFH	0...3.99999	0.2384e-6	[s]
Minimum time between camshaft signal edges					
NC_T_SEG_MAX_CAM_IN	-	0...FFFFFFH	0...3.99999	0.2384e-6	[s]
Maximum time between camshaft signal edges					
NC_T_SEG_MIN_CAM_EX	-	0...FFFFFFH	0...3.99999	0.2384e-6	[s]
Minimum time between camshaft signal edges					
NC_T_SEG_MAX_CAM_EX	-	0...FFFFFFH	0...3.99999	0.2384e-6	[s]
Maximum time between camshaft signal edges					
NC_NR_EDGE_CAM_IN	-	1...FH	1...15	1	[-]
Number of signal edges per intake camshaft revolution					
NC_NR_EDGE_CAM_EX	-	1...FH	1...15	1	[-]
Number of signal edges per exhaust camshaft revolution					
NC_NR_GAP	-	1...2H	1...2	1	[-]
Number of reference gaps per engine revolution					
NC_ACT_CAM_EDGE_SYN	-	1...3H	1...3	1	[-]
Active edge of camshaft signal for cam/crk synchronization					
NC_ACT_CAM_EDGE_LIH	-	1...3H	1...3	1	[-]
Active edge of camshaft signal for crankshaft limp-home					
NC_PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Ideal engine position relative to TDC0 for intake camshaft i signal edge z					
NC_PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Ideal engine position relative to TDC0 for exhaust camshaft i signal edge z					

23.6 Engine speed signal output “TD”

FUNCTION DESCRIPTION:

The engine speed signal TD is a tooth-fixed event at the ECU Pin A_F_TD. The signal is set to ‘high’ at the teeth 1, 21, 41, 61, 81, 101 and reset to ‘low’ at the teeth 11, 31, 51, 71, 91, 111. This means: falling edge at every TDC, rising edge at every BDC (of any cylinder).
The artificially generated teeth are used in the CAM limp home. Since the cylinder group assignment is not decisive for the output of the TD signal, the CRK limp home has no influence on the output of the TD signal.

Initialisation :

The output signal is initialised with a low logic state (0).

23.7 Requirements to Infrastructure Interface for partial segment time acquisition

Export actions:

ACTION_INFR_EnableT5T()
This action activates segment time acquisition for 5 teeth- segments
ACTION_INFR_DisableT5T()
This action deactivates segment time acquisition for 5 teeth- segments
ACTION_INFR_GetT5T(IN <idx>, **OUT** <time>**)**
Acquisition of partial segment time over 5 teeth
ACTION_INFR_GetT5TDiag(OUT <diag**>)**
Diagnostic information for acquisition of 5-teeth- segments

Description for actions:

ACTION_INFR_EnableT5T()					
This action activates segment time acquisition for 5 teeth- segments					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

ACTION_INFR_DisableT5T()					
This action deactivates segment time acquisition for 5 teeth- segments					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit

ACTION_INFR_GetT5T(IN <idx>, **OUT** <time>**)**					
Signal acquisition for 5 teeth- segments					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
idx	IN	0…18H	0...23	1	[-]
Index of the array of measured partial segment time					
time	OUT	0…FFFFH	0...65,535	0,001	ms
Measured partial segment time over 5 teeth					

ACTION_INFR_GetT5T_Diag(diag**)**					
Diagnostic information for acquisition of 5-teeth- segments					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
diag	OUT	0…1H	0...1	1	[-]
Diagnostic for partial segment time acquisition, set to 1 when buffer content is valid					

General information:

The function shall sample a time stamp every 5 teeth and calculate the difference of the actual and the last time stamp. The result is stored in a buffer with 24 values. The return value of ACTION_INFR_GetT5T(idx,time) is the time from the indexed cell of the array.

Acquisition shall start at TDC cylinder 0. Values are written to the appropriate location of the array (e.g., the first partial segment time measured after TDC 0 is written to the first location of the array).
In case of overflow (below approx. 80rpm), the measured segment time is limited to the maximum value.
When enabling or disabling action is called, all buffer values are set to maximum value.
A diagnostic action delivers information about validity of the buffer contents. The return value is 0 after enabling as long as buffer content is not valid, and changes to one as soon as the buffer is ready to be read out. Buffer content is regarded as valid as soon as all 24 values in the buffer are filled.

Requirements for ACTION_INFR_EnableT5T:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

Coincidence requirements:

Requirements for ACTION_INFR_DisableT5T:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

Diagnosis:

Coincidence requirements:

Requirements for ACTION_INFR_GetT5T:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
index	1	1	1		
time		0,001ms	0,001ms		

Diagnosis:

Coincidence requirements: Acquisition shall start at the start of segment 0, and repeat every 5 teeth.

Requirements for ACTION_INFR_GetT5TDiag:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Diag	1		1		

Diagnosis: no diagnosis done here

Coincidence requirements: end of acquisition for the buffer (all 24 values filled)

23.8 AGGR ENSD adaptation: Engine Position and Speed Determination

23.8.1 Outputs for SV functions

23.8.1.1 Generation of N_TOOTH

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_TOOTH_CUS	O/V	0...1FE0H	0...8160	1	[rpm]
fast engine speed generated especially for engine start (independently of C_N_FAST_SWI)					

Input data:

LV_RUN_ENG	N_32	C_N_TOOTH_END	NC_NR_TOOTH
T_TOOTH			

FUNCTION DESCRIPTION:

Description:

As there is a strong request for a fast engine speed especially for engine start, and ENSD solely has the possibility to generate one out of four (N_FAST can either be similar to N_TOOTH or three differently calculated N_FASTs – see “engine position and speed calculation”), N_TOOTH has to be calculated additionally in this module.

Application conditions:

Initialisation: N_TOOTH_CUS = 0

Recurrence: 10ms

Activation: LV_RUN_ENG = 1 **and**

$N\_32 \le C\_N\_TOOTH\_END$

Deactivation: LV_RUN_ENG = 0 **or**

$N\_32 > C\_N\_TOOTH\_END$

Formula section:

N_TOOTH_CUS (phys) = 60 / (NC_NR_TOOTH * T_TOOTH)

23.8.1.2 Generation of LV_ERR_CAM_CUS

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CAM_CUS	V/O	0...1H	0...1	1	[-]
Customer specific CAM error					

Input data:

LV_ERR_CAM_TOT			

Description:

LV_ERR_CAM_CUS is for customer specific adaptations on the ENSD interface LV_ERR_CAM_TOT.

Application conditions:

Initialisation: LV_ERR_CAM_CUS = 0 at reset, LV_IGK 0->1 and clearance of failurememory

Recurrence: every Camshaft signal edge (just after calculation of all CAM diagnosis)

Activation: every engine operating state

Formula section:

LV_ERR_CAM_CUS = LV_ERR_CAM_TOT

23.9 Segment time acquisition for partial segments

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_T_SEG_5_TOOTH_VLD	V/O	0...1H	0...1	1	[-]
Flag: acquisition of partial segment times valid					
LV_T_SEG_5_TOOTH_ACT	-	0...1H	0...1	1	[-]
Flag: acquisition of partial segment times active					

Input data:

N_32			

Import actions:

ACTION_INFR_EnableT5T()
This action activates segment time acquisition for 5 teeth- segments
ACTION_INFR_DisableT5T()
This action deactivates segment time acquisition for 5 teeth- segments
ACTION_INFR_GetT5TDiag(OUT <diag**>)**
Diagnostic information for acquisition of 5-teeth- segments

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_32_MAX_SEG_5_TOOTH	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed for acquisition of partial segment time over 5 teet					

FUNCTION DESCRIPTION:

General information:

The function shall measure the period for segments of 5 teeth, and output the result as an array of 24 values for the whole engine cycle.

Acquisition shall start at regular segment event for cylinder 0 (SEG_NR = 0). Every 5 teeth a time stamp shall be sampled, the period calculated as the difference of the actual and the last time stamp, and written to the array of temporary values at the appropriate location (e.g., the first partial segment time measured after start of regular segment 0 is written to the first location of the array).

Application conditions:

Activation: at ENG_STOP to SYN_ENG_IGK_ON
or RUN_ENG to SYN_ENG_IGK_ON

Deactivation : at SYN_ENG_IGK_ON to RUN_ENG (if supported by ECOP)
or SYN_ENG_IGK_ON to ENG_STOP
or SYN_ENG_IGK_OFF to PWL :
ACTION_INFR_DisableT5T()
LV_T_SEG_5_TOOTH_VLD = 0

LV_T_SEG_5_TOOTH_ACT = 0

Initialization : at reset :

LV_T_SEG_5_TOOTH_VLD = 0

LV_T_SEG_5_TOOTH_ACT = 0

Recurrence:

segment

Formula section:

```
If(1)       N_32 <= C_N_32_MAX_SEG_5_TOOTH
Then(1)
            If(2)       ( LV_T_SEG_5_TOOTH_ACT =  0)
            Then(2)     ACTION_INFR_EnableT5T()
                        LV_T_SEG_5_TOOTH_ACT = 1
            Endif(2)
              get diagnostic information from infrastructure
            ACTION_INFR_GetT5TDiag(
                        OUT          LV_T_SEG_5_TOOTH_VLD)
Else (1)
            If(3) ( LV_T_SEG_5_TOOTH_ACT =  1)
            Then(3)    ACTION_INFR_DisableT5T()
                         LV_T_SEG_5_TOOTH_ACT = 0
                         LV_T_SEG_5_TOOTH_VLD = 0
            Endif(3)
Endif(1)
```

23.10 Engine Position Manager

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
ST_EPM	-	0...8H	0...8	1	[-]
Engine position manager state					
LV_STOP_ENG	O	0...1H	0...1	1	[-]
Boolean information for engine stop request					
LV_FIRST_VLD_TOOTH	V/O	0...1H	0...1	1	[-]
First tooth detected after engine stop					
LV_RUN_ENG	V/O	0...1H	0...1	1	[-]
Engine running					
LV_SYN_ENG	V/O	0...1H	0...1	1	[-]
Boolean information for engine synchronization completed.					
LV_SYN_VLD	V/O	0...1H	0...1	1	[-]
Engine synchronization validated					
PSN_ENG_CRK	O	0...2CFFH	0...719.9375	0.0625	[°CRK]
Engine position from crankshaft					
PSN_ENG_CAM_LIH_CRK	O	0...2CFFH	0...719.9375	0.0625	[°CRK]
Engine position in crankshaft limp-home mode					
LV_LIH_ERR_CRK	V/O	0...1H	0...1	1	[-]
Crankshaft limp-home mode (crankshaft sensor failure)					
LV_LIH_ERR_CAM	V	0...1H	0...1	1	[-]
Camshaft limp-home mode (failure on all camshaft sensors)					
LV_ACT_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Selfsynchronisation activated on intake camshaft i					
LV_ACT_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Selfsynchronisation activated on exhaust camshaft i					
LV_ACT_LIH_CRK_CAM_IN[NC_NR_CAM_CBK]	V	0...1H	0...1	1	[-]
Crankshaft limp-home operating mode activated on intake camshaft i					
LV_ACT_LIH_CRK_CAM_EX[NC_NR_CAM_CBK]	V	0...1H	0...1	1	[-]
Crankshaft limp-home operating mode activated on exhaust camshaft i					
LV_ACT_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Crankshaft synchronization operating mode activated on intake camshaft i					
LV_ACT_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Crankshaft synchronization operating mode activated on exhaust camshaft i					
LV_ACT_CRK	V/O	0...1H	0...1	1	[-]
Crankshaft acquisition active					
LV_ERR_CAM	V/O	0...1H	0...1	1	[-]
Failure on all camshaft sensor signals in normal mode					
LV_ERR_LIH_CRK_CAM	V	0...1H	0...1	1	[-]
No more camshaft sensor signals available for crankshaft limp-home mode					
CTR_VLD_CAM_SYN_CRK	V	0...FFH	0...255	1	[-]
Counter for validation of camshaft signal for crankshaft synchronization					
LV_ENG_BACK_DET	V/O	0...1H	0...1	1	[-]
Engine backwards rotation detected					
LV_ENG_BACK_CFM	V/O	0...1H	0...1	1	[-]
Engine backwards rotation confirmed					
LV_CRK_MISS_RUN_ENG	V/O	0...1H	0...1	1	[-]
Boolean information engine running after engine stop request					

Input data:

LV_SYN_CAM_IN[NC_NR_CAM_CBK]	LV_IGK	N	LV_CRK_FIRST_VLD_TOOTH
LV_SYN_CAM_EX[NC_NR_CAM_CBK]	LV_CRK_RUN	LV_ERR_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_CAM_EX[NC_NR_CAM_CBK]
LV_CAM_SYN_CRK	LV_CRK_SYN	LV_CRK_STOP	LV_CAM_STOP_IN[NC_NR_CAM_CBK]
LV_ORNG_CAM_SYN_CRK	PSN_ENG_SYN_CAM_MIN	PSN_ENG_CRK_OFS	LV_CAM_STOP_EX[NC_NR_CAM_CBK]
PSN_EDGE_AD_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	NC_OFS_TDC0_REF_CRK	NC_PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	NC_PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]
PSN_EDGE_AD_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	NC_NR_TOOTH	IDX_EDGE_CAM_IN[NC_NR_CAM_CBK]	IDX_EDGE_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_CRK	LV_SYN_VLD_CAM_LIH	LV_SEG_NR_UPD_REQ	LV_CAM_LIH_EXT_ENA
NC_ACT_CAM_EDGE_SYN	NC_ACT_CAM_EDGE_LIH	LV_CRK_MISS_TOOTH	PSN_ENG_ENSD
C_DYW_CAM_CRK_SYN_RTD_IN	C_DYW_CAM_CRK_SYN_RTD_EX	C_DYW_CAM_CRK_SYN_ADC_IN	C_DYW_CAM_CRK_SYN_ADC_EX
LV_ORNG_RATIO_CAM_IN[NC_NR_CAM_CBK]	LV_ORNG_RATIO_CAM_EX[NC_NR_CAM_CBK]	LV_LOST_SYN_CRK	C_N_NOT_REST
LC_ENG_BACK_INH	N_TOOTH		

FUNCTION DESCRIPTION:

General information:

The Engine Position Manager provides a summary of camshaft and crankshaft synchronization status information to the system. It sets the operating modes of camshaft and crankshaft acquisition depending on synchronization status, availability, and detected failures. The signal source for engine position will be selected based on the activated operating mode.

Camshaft and crankshaft signal acquisition will be set passive for a calibrated delay after key-off has been detected at low engine speed, and for another calibrated delay after a loss of synchronization at low engine speed. This avoids wrong injection, ignition and diagnosis with backward rotation after rapid key-off/on or stalling.

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Description:

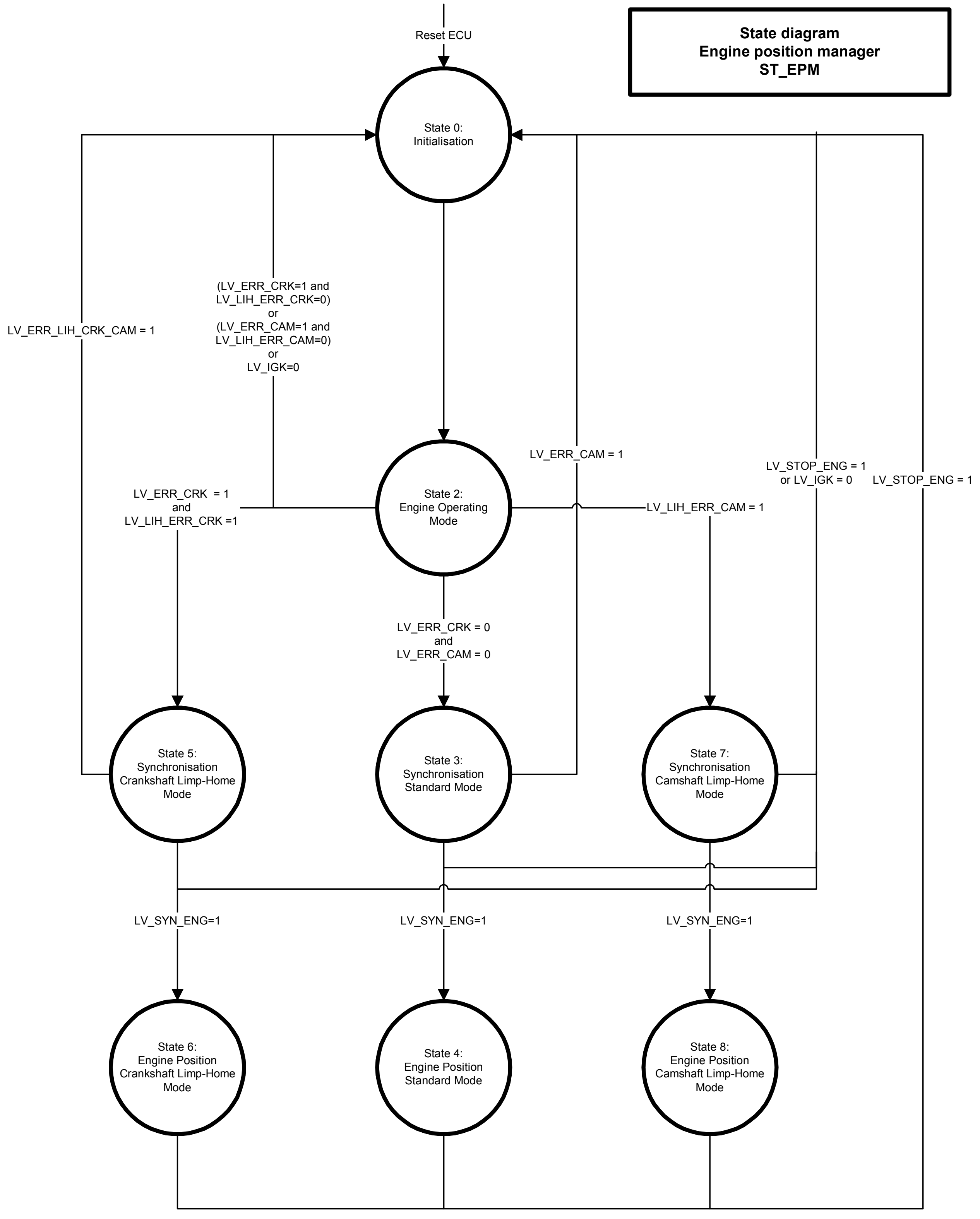
Engine synchronized (LV_SYN_ENG)	Yes (LV_SYN_ENG = 1)	No (LV_SYN_ENG = 0)
Crankshaft sensor available (LV_ERR_CRK = 0) At least one camshaft sensor available (LV_ERR_CAM = 0)	Standard Mode: Engine position and speed calculation with crankshaft sensor signal	Standard Mode Engine synchronisation with camshaft and crankshaft sensor information
Crankshaft sensor available (LV_ERR_CRK = 0) Failure on **all** Camshaft sensors (LV_ERR_CAM = 1)	Limp home Cam: Engine position and speed calculation with crankshaft sensor signal	Limp home Cam: Engine synchronisation without camshaft sensor information
Crankshaft sensor failure (LV_ERR_CRK = 1) At least one camshaft sensors available for crank limp-home (LV_ERR_LIH_CRK_CAM = 0)	Limp home Crank: Engine position and speed calculation without crankshaft sensor signal.	Limp home Crank: Engine synchronisation without crankshaft sensor information
Crankshaft sensor failure (LV_ERR_CRK = 1) Failure on **all** Camshaft sensors available for crank limp-home		Engine synchronisation fiasco: Engine cannot be synchronized

(LV_ERR_LIH_CRK_CAM = 1)		

Application conditions:

Recurrence: 10 ms

23.10.1 State flow diagram



Initialisation:

LV_ERR_CAM is reset at 0 to 1 transition of LV_IGK

LV_CRK_MISS_RUN_ENG is reset and timer C_T_DLY_STOP_ENG stopped at 1 to 0 transition of LV_IGK

Formula section:

State 0 : Initialisation

Initialization of output data and deactivation of all state flow diagrams.

Input conditions:

External event: ECU Reset
From state 2: (LV_ERR_CRK = 1 And LV_LIH_ERR_CRK = 0)
Or
(LV_ERR_CAM = 1 And LV_LIH_ERR_CAM = 0)
Or
LV_IGK = 0
From state 3: LV_ERR_CAM = 1
From state 5: LV_ERR_LIH_CRK_CAM = 1
From state 3,5,7: LV_STOP_ENG = 1
Or
LV_IGK = 0
From state 4,6,8: LV_STOP_ENG = 1

Output conditions:

To State 2 : End of initialisation

Action in the state:

LV_STOP_ENG = 1
LV_FIRST_VLD_TOOTH = 0
LV_RUN_ENG = 0
LV_SYN_ENG = 0
LV_SYN_VLD = 0
PSN_ENG_CRK = PSN_ENG_CAM_LIH_CRK = 0
CTR_VLD_CAM_SYN_CRK = 0
LV_LIH_ERR_CRK = 0
LV_LIH_ERR_CAM = 0
LV_ERR_LIH_CRK_CAM = 0

Deactivation of Crankshaft self-synchronisation:

LV_ACT_CRK = 0

Deactivation of Camshaft self-synchronisation for all camshaft signals:

LV_ACT_CAM_IN(EX)[i] = 0

Deactivation of Camshaft/Crankshaft synchronisation for all camshaft signals:

LV_ACT_SYN_CRK_CAM_IN(EX)[i] = 0

Deactivation of limp-home mode for all camshaft signals:

LV_ACT_LIH_CRK_CAM_IN(EX)[i] = 0

Deactivation of synchronization determination in camshaft limp home

LV_CAM_LIH_EXT_ENA = 0

LV_SYN_VLD_CAM_LIH = 0

LV_SEG_NR_UPD_REQ = 0

Deactivation of synchronization validation in camshaft limp home

Transition action:

None

State 2 : Engine Operating Mode

Select engine operating mode in function of diagnostic information and limp home status.

Input conditions:

From State 0: End of initialisation

Output conditions:

To state 0: (LV_ERR_CRK = 1 And

(LC_INH_LIH_CRK = 1 Or LV_ERR_LIH_CRK_CAM = 1))

Or

(LV_ERR_CAM = 1 And

LC_INH_LIH_CAM = 1 And LV_ERR_CRK = 0)

Or

LV_IGK = 0

To state 3: LV_ERR_CRK = 0 And LV_ERR_CAM = 0

To state 5: LV_LIH_ERR_CRK = 1 And LV_ERR_CRK = 1

To state 7: LV_LIH_ERR_CAM = 1

Action in the state:

```
If      LV_ERR_CRK = 1
Then    If      LC_INH_LIH_CRK = 0
        And     LV_ERR_LIH_CRK_CAM = 0
        Then    Select Crankshaft limp-home mode
                LV_LIH_ERR_CRK = 1
        Else    Crankshaft limp-home not possible
        Endif
Else    If      LV_ERR_CAM = 1
        And     LC_INH_LIH_CAM = 0
        Then    Select Camshaft limp-home mode
                LV_LIH_ERR_CAM = 1
        Else    Camshaft limp-home not possible
        Endif
Endif
```

Transition action:

To state 3,5,7: LV_STOP_ENG = 0

To state 3: Activation of Crankshaft self-synchronisation:

LV_ACT_CRK = 1

Activation of Camshaft self-synchronisation for all camshaft signals:

LV_ACT_CAM_IN(EX)[i] = 1

Activation of camshaft selection for synchronisation ()

Activation of Camshaft/Crankshaft synchronisation for the selected camshaft signal.

To state 5: Activation of Camshaft self-synchronisation for all the camshaft signals:

LV_ACT_CAM_IN(EX)[i] = 1

Activation of camshaft selection for limp-home ()

To state 7: Activation of Camshaft self-synchronisation for all the camshaft signals:

LV_ACT_CAM_IN(EX)[i] = 1

Activation of Crankshaft self-synchronisation:

LV_ACT_CRK = 1

State 3 : Synchronisation Standard Mode

Engine synchronisation with camshaft and crankshaft sensor information.

Input conditions:

From state 2: LV_ERR_CRK = 0 And LV_ERR_CAM = 0

Output conditions:

To state 0: LV_STOP_ENG = 1
Or
LV_IGK = 0
Or
LV_ERR_CAM = 1

To state 4: LV_SYN_ENG = 1

Action in the state:

```
LV_FIRST_VLD_TOOTH = LV_CRK_FIRST_VLD_TOOTH
LV_RUN_ENG = LV_CRK_RUN
Activation of calculation of PSN_ENG_CRK in standard mode()
If      LV_FIRST_VLD_TOOTH = 1
Then    Activation of synchronization validation ()
End if
Activation of engine position calculation standard mode ()
Activation of Camshaft self-synchronisation reactivation ()
Activation of engine stop detection ()

If      LV_RUN_ENG = 1
Then    Activation of engine speed calculation
Endif

On 0 to 1 transition of LV_CAM_SYN_CRK:
    If          LV_ERR_CAM_IN(EX)[i] = 0
                For all available camshaft sensors
    Then        Activation of fast synchronization ()
    Else        Activation of slow synchronization ()
    Endif
```

Transition action:

To state 4: LV_CRK_MISS_RUN_ENG = 0

Stop timer C_T_DLY_STOP_ENG

State 4 : Engine Position Standard Mode

Engine position and speed calculation with crankshaft sensor signal.

Input conditions:

From state 3: LV_SYN_ENG = 1

Output conditions:

To state 0: LV_STOP_ENG = 1

Action in the state:

```
Reactivation of Camshaft self-synchronisation ()
Activation of engine position calculation standard mode ()
Activation of segment period calculation ()
Activation of speed gradient calculation ()
Activation of crankshaft sensor phase angle correction ()
Activation of engine stop detection ()

If     LV_SYN_VLD = 1
Then   Activation of Camshaft adaptation and position output ()
Endif
Activation of Engine backwards rotation detection ()
```

Transition action:

To state 0:

```
If     C_T_DLY_STOP_ENG > 0
       And LV_IGK = 1
       And LV_CRK_MISS_RUN_ENG = 0
Then   LV_CRK_MISS_RUN_ENG = 1
              Start timer C_T_DLY_STOP_ENG
       Reset LV_CRK_MISS_RUN_ENG after timer has elapsed
Endif
```

State 5 : Synchronisation Crankshaft Limp-Home Mode

Engine synchronisation without crankshaft sensor signal.

Input conditions:

From state 2: LV_LIH_ERR_CRK = 1

And

LV_ERR_CRK = 1

Output conditions:

To state 0: LV_STOP_ENG = 1

Or

LV_IGK = 0

Or

LV_ERR_LIH_CRK_CAM = 1

To state 6: LV_SYN_ENG=1

Action in the state:

Activation of crankshaft limp-home synchronization() for the selected camshaft signal

Activation of calculation of PSN_ENG_CAM_LIH_CRK()

Activation of engine position calculation crankshaft limp-home mode

Reactivation of Camshaft self-synchronisation ()

Activation of engine stop detection ()

```
If    LV_RUN_ENG = 1
Then  Activation of engine speed calculation
      Endif
```

Transition action:

To state 6: LV_CRK_MISS_RUN_ENG = 0

Stop timer C_T_DLY_STOP_ENG

State 6 : Engine Position Crankshaft Limp-Home Mode

Engine position and speed calculation without crankshaft sensor signal.

Input conditions:

From state 5: LV_SYN_ENG = 1

Output conditions:

To state 0: LV_STOP_ENG = 1

Action in the state:

Activation of engine position calculation crankshaft limp-home mode

Activation of Camshaft self-synchronisation reactivation ()

Activation of segment period calculation

Activation of speed gradient calculation

Activation of engine stop detection ()

Transition action:

To state 0:

```
If      C_T_DLY_STOP_ENG > 0
        And LV_IGK = 1
        And LV_CRK_MISS_RUN_ENG = 0
Then    LV_CRK_MISS_RUN_ENG = 1
                Start timer C_T_DLY_STOP_ENG
        Reset LV_CRK_MISS_RUN_ENG after timer has elapsed
Endif
```

State 7 : Synchronisation Camshaft Limp-Home Mode

Engine synchronisation without camshaft sensor signal.

Input conditions:

From state 2: LV_LIH_ERR_CAM = 1

Output conditions:

To state 0: LV_STOP_ENG = 1
Or
LV_IGK = 0

To state 8: LV_SYN_ENG = 1

Action in the state:

Activation of camshaft limp-home synchronization()

LV_FIRST_VLD_TOOTH = LV_CRK_FIRST_VLD_TOOTH

LV_RUN_ENG = LV_CRK_RUN

Activation of calculation of PSN_ENG_CRK in camshaft limp-home mode()

Activation of engine position calculation standard mode

Activation of engine stop detection ()

If LV_RUN_ENG = 1

Then Activation of engine speed calculation

Endif

Transition action:

To state 8: LV_CRK_MISS_RUN_ENG = 0

Stop timer C_T_DLY_STOP_ENG

State 8 : Engine Position Camshaft Limp-Home Mode

Engine position and speed calculation without camshaft sensor signal.

Input conditions:

From state 7: LV_SYN_ENG = 1

Output conditions:

To state 0: LV_STOP_ENG = 1

Action in the state:

Activation of Synchronization determination in camshaft limp home()

Activation of Synchronization validation in camshaft limp home()

Reactivation of Camshaft self-synchronisation ()

Activation of PSN_ENG_CRK correction in camshaft limp home()

Activation of engine position calculation standard mode

Activation of segment period calculation

Activation of segment number correction in camshaft limp home()

Activation of speed gradient calculation

Activation of crankshaft sensor phase angle correction

Activation of engine stop detection ()

Activation of Engine backwards rotation detection ()

Transition action:

```
To state 0:    If     C_T_DLY_STOP_ENG > 0
                      And LV_IGK = 1
                      And LV_CRK_MISS_RUN_ENG = 0
               Then   LV_CRK_MISS_RUN_ENG = 1
                          Start timer C_T_DLY_STOP_ENG
                      Reset LV_CRK_MISS_RUN_ENG after timer has elapsed
               Endif
```

23.10.2 Definition of the sub-function tasks

23.10.2.1 Fast Engine synchronization

General information:

Fast engine synchronization is in charge to synchronize as soon as the crankshaft signal acquisition is synchronized and a camshaft signal information is available on the selected camshaft.

Application conditions:

Recurrence: on 0 to 1 transition of LV_CAM_SYN_CRK

Formula section:

```
If      LV_CAM_SYN_CRK = 1
Then    LV_SYN_ENG = 1
Endif
```

23.10.2.2 Slow Engine synchronization

General information:

Slow engine synchronization is activated when a failure was detected on at least one camshaft sensor. The engine is synchronized as soon as a number of calibratable camshaft edges has occured (engine synchronized when synchronization is validated). This avoids multiple injection and ignition events with wrong synchronization.

Application conditions:

Recurrence: at every active camshaft edge and on 0 to 1 transition of LV_CAM_SYN_CRK

The active camshaft edge for synchronization is defined by NC_ACT_CAM_EDGE_SYN.

Formula section:

The engine speed value used to calculate NR_VLD_CAM_SYN_CRK is the one determined at transition from 0 to 1 of LV_CAM_SYN_CRK (see § Synchronization validation)

```
If      LV_CAM_SYN_CRK = 1
```

```
Then    If      CTR_VLD_CAM_SYN_CRK > NR_VLD_CAM_SYN_CRK
        Then    LV_SYN_ENG = 1
        Endif
Endif
```

Remark: CTR_VLD_CAM_SYN_CRK is incremented in "synchronization validation ()" function

23.10.2.3 Crankshaft limp-home synchronization

General information:

Only the informations of the selected camshaft signal is used to perform the engine synchronization.

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Application conditions:

Recurrence: at every camshaft edge

Formula section:

```
If      LV_SYN_CAM_IN(EX)[i] = 1 for the selected camshaft signal
Then    LV_FIRST_VLD_TOOTH = 1
        LV_RUN_ENG = 1
        LV_SYN_ENG = 1
Endif
```

23.10.2.4 Camshaft limp-home synchronization

General information:

No camshaft signals are available for the Crank/Cam synchronization so the engine synchronization is performed as soon as the crankshaft is synchronized (the engine revolution can not be identified so a default value is used).

Application conditions:

Recurrence: at every reference gap

Formula section:

```
If      LV_CRK_SYN = 1
Then    LV_SYN_ENG = 1
Endif
```

23.10.2.5 Synchronisation validation

General information:

Synchronisation is validated as soon as a number of calibratable camshaft edges has occured. During validation phase, the angular distance between cam/crk events is checked. Engine synchronisation is reset if one of the tests fails.
i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Application conditions:

Recurrence: *at every active camshaft edge and on 0 to 1 transition of LV_CAM_SYN_CRK as long as LV_SYN_VLD = 0*

The active camshaft edge for synchronization is defined by NC_ACT_CAM_EDGE_SYN.

Formula section:

At transition from 0 to 1 of LV_CAM_SYN_CRK, NR_VLD_CAM_SYN_CRK = ID_NR_VLD_CAM_SYN_CRK.

Increment CTR_VLD_CAM_SYN_CRK

```
If      LV_CAM_SYN_CRK = 1
Then    If      CTR_VLD_CAM_SYN_CRK > NR_VLD_CAM_SYN_CRK
        Then    LV_SYN_VLD = 1
                Deactivation of Camshaft/Crankshaft synchronisation for the camshaft
                signal selected for this synchronization:
                LV_ACT_SYN_CRK_CAM_IN(EX)[i] = 0
        Endif
Endif
```

23.10.2.6 Synchronisation validation in camshaft limp home

General information:

Synchronization validation information is set as soon as it is requested by “Synchronization determination in camshaft limp home” module which is then deactivated.

Application conditions:

Initialisation: none

Recurrence: each segment event

Formula section:

```
If      LV_SYN_VLD_CAM_LIH = 1
Then    LV_SYN_VLD = 1
        Deactivation of Synchronization determination in camshaft limp home()
End if
```

23.10.2.7 Camshaft Selection for synchronisation

General information:

Target wheels on intake and exhaust camshafts may be different, and all may be useable for synchronization. A camshaft signal can only be selected if no error is detected on the respective camshaft (LV_ERR_CAM_IN(EX)[i] = 0). For port-injection engines activation of synchronization mode on intake camshafts should be preferred. Pre-injection needs to be synchronized with intake timing, and the engine position for pre-injection is calculated from the synchronization signal.

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Formula section:

The selection order is given by the following table:

	NC_PRI_SYN_CAM_CBK = 1	NC_PRI_SYN_CAM_CBK = 2
NC_PRI_SYN_CAM_IN = 0	CAM_EX_1 CAM_EX_2 CAM_IN_1 CAM_IN_2	CAM_EX_2 CAM_EX_1 CAM_IN_2 CAM_IN_1
NC_PRI_SYN_CAM_IN = 1	CAM_IN_1 CAM_IN_2 CAM_EX_1 CAM_EX_2	CAM_IN_2 CAM_IN_1 CAM_EX_2 CAM_EX_1

The first camshaft sensor in the above list which fulfills the following conditions is selected:

- physically present in the system (configuration data NLC_CAM_IN, NLC_CAM_EX and NC_NR_CAM_CBK)
- no failure reported on it (LV_ERR_CAM_IN(EX)[i] = 0)

The crankshaft synchronization mode is activated on one camshaft signal by setting LV_ACT_SYN_CRK_CAM_IN(EX)[i] for the selected camshaft.

LV_ERR_CAM is set if all available camshaft sensors report and error.

23.10.2.8 Camshaft Selection for limp-home

General information:

Target wheels on intake and exhaust camshafts may be different, and not all may be useable for limp-home. The availability of camshaft sensors for limp-home is given by NLC_LIH_CAM_IN(EX) and NC_NR_CAM_CBK. A camshaft signal can only be selected if no error is detected on the respective camshaft (LV_ERR_CAM_IN(EX)[i] = 0).

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Formula section:

The selection order is given by the following table:

	NC_PRI_LIH_CAM_CBK = 1	NC_PRI_LIH_CAM_CBK = 2
NC_PRI_LIH_CAM_IN = 0	CAM_EX_1 CAM_EX_2 CAM_IN_1 CAM_IN_2	CAM_EX_2 CAM_EX_1 CAM_IN_2 CAM_IN_1
NC_PRI_LIH_CAM_IN = 1	CAM_IN_1 CAM_IN_2 CAM_EX_1 CAM_EX_2	CAM_IN_2 CAM_IN_1 CAM_EX_2 CAM_EX_1

The first camshaft sensor in the above list which fulfills the following conditions is selected:

- physically present in the system (configuration data NLC_CAM_IN, NLC_CAM_EX_IN and NC_NR_CAM_CBK)
- no failure reported on it (LV_ERR_CAM_IN(EX)[i] = 0)
- available for crankshaft limp-home (NLC_LIH_CAM_IN = 1 or NLC_LIH_CAM_EX = 1)

The crankshaft limp-home mode is activated on one camshaft signal by setting LV_ACT_LIH_CRK_CAM_IN(EX)[i] for the selected camshaft.

LV_ERR_LIH_CRK_CAM is set if no more camshaft sensors is available.

23.10.2.9 Engine stop detection

Application conditions:

Recurrence: 10ms

Formula section:

LV_STOP_ENG is set if one of the conditions in the table below are met:

In Standard mode (States 3 and 4)	In Camshaft limp-home mode (States 7 and 8)	In Crankshaft limp-home mode (States 5 and 6)	In Engine position Standard and Camshaft limp-home mode (States 4 and 8)
LV_CRK_STOP = 1	LV_CRK_STOP = 1	LV_CAM_STOP_IN(	LV_ENG_BACK_CF

		EX)[i] = 1 for the selected camshaft signal	M switches from 0 to 1
LV_ORNG_CAM_SYN_CRK = 1 (*)			
LV_ERR_CRK = 1 And LC_INH_LIH_CRK = 0			

(*) This information is available as long as LV_SYN_VLD = 0.

23.10.2.10 Camshaft self-synchronisation reactivation

General information:

When the self-synchronization of a camshaft signal goes to the fiasco state (LV_CAM_STOP_IN(EX)[i] = 1), the self-synchronization on this camshaft signal will be reactivated as soon as possible. If this camshaft is selected for the synchronization in crankshaft limp-home mode an engine stop is detected (by *engine stop detection()* sub-function task) and a new synchronization is restarted.

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Application conditions:

Recurrence: 10ms

Formula section:

For all available camshaft signal sensors:

```
If      LV_CAM_STOP_IN(EX)[i] = 1
AndLV_ACT_LIH_CRK_CAM_IN(EX)[i] = 0
Then    Activation of Camshaft self-synchronisation for this camshaft signal
Endif
```

23.10.2.11 *Calculation of PSN_ENG_CRK in standard mode()*

General information:

Formula section:

Before synchronization (LV_SYN_ENG = 0):

Truth table					
LV_CRK_SYN n	0	1			
LV_CRK_SYN n-1	0	0		1	
LV_CAM_SYN_CRK n	0	0	1	0	1
LV_CAM_SYN_CRK n-1	0	0	0	0	0

Engine position action	**1**	**2**	**3**	**4**	**5**

Action 1:

PSN_ENG_CRK = PSN_ENG_SYN_CAM_MIN

Action 2:

PSN_ENG_CRK = PSN_ENG_SYN_CAM_MIN

PSN_ENG_CRK_GAP = 360° - NC_OFS_TDC0_REF_CRK

Action 3:

PSN_ENG_CRK = 360° - NC_OFS_TDC0_REF_CRK + PSN_ENG_CRK_OFS

Action 4:

PSN_ENG_CRK = PSN_ENG_SYN_CAM_MIN

PSN_ENG_CRK_GAP n = PSN_ENG_CRK_GAP n-1 + (360°/NC_NR_TOOTH)

Action 5:

PSN_ENG_CRK = PSN_ENG_CRK_GAP + PSN_ENG_CRK_OFS

After the synchronization (LV_SYN_ENG = 1):

PSN_ENG_CRK is incremented on the tooth signal and jitter from 0 to 720°.

23.10.2.12 *Calculation of PSN_ENG_CRK in camshaft limp-home mode()*

General information:

Formula section:

Before synchronization (LV_SYN_ENG = 0):

Truth table		
LV_CRK_SYN n	0	1
LV_CRK_SYN n-1	0	0
Engine position action	**1**	**2**

Action 1:

PSN_ENG_CRK is incremented on the tooth signal and jitter from 0 to 360°

Action 2:

PSN_ENG_CRK = 720° - NC_OFS_TDC0_REF_CRK

After the synchronization (LV_SYN_ENG = 1):

PSN_ENG_CRK is incremented on the tooth signal and jitter from 0 to 720°

23.10.2.13 *PSN_ENG_CRK correction in camshaft limp-home()*

General information:

This function updates PSN_ENG_CRK depending on synchronization determination test result in camshaft limp home.

Application conditions:

Initialisation: none

Recurrence: none

Activation: LV_SEG_NR_UPD_REQ = 1

Deactivation: LV_SEG_NR_UPD_REQ = 0

Formula section:

PSN_ENG_CRK = PSN_ENG_CRK + (360°CRK / NC_NR_GAP)

23.10.2.14 *Calculation of PSN_ENG_CAM_LIH_CRK()*

General information:

The purpose of this function is to provide engine position during crankshaft limp-home. In crankshaft limp-home, the engine position is calculated from the camshaft sensor selected by LV_ACT_LIH_CRK_CAM_IN(EX)[i].

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Application conditions:

Initialization: PSN_ENG_CAM_LIH_CRK = 0

Recurrence: every camshaft signal edge

Activation: LV_SYN_ENG = 1

Deactivation: LV_SYN_ENG = 0

Formula section:

For z = IDX_EDGE_CAM_IN(EX)[i] and the camshaft sensor selected for crankshaft limp home by LV_ACT_LIH_CRK_CAM_IN(EX)[i]:

```
If      z is a falling camshaft signal edge
Then    If      NC_ACT_CAM_EDGE_LIH = 1
                Or
                NC_ACT_CAM_EDGE_LIH = 3
        Then    PSN_ENG_CAM_LIH_CRK = NC_PSN_EDGE_CAM_IN(EX)[z][i] +
                    Average value of PSN_EDGE_AD_CAM_IN(EX)[z][i] for all falling
                    camshaft signal edges
        Else    no update of PSN_ENG_CAM_LIH_CRK
        Endif
Else    If      NC_ACT_CAM_EDGE_LIH = 2
                Or
                NC_ACT_CAM_EDGE_LIH = 3
        Then    PSN_ENG_CAM_LIH_CRK = NC_PSN_EDGE_CAM_IN(EX)[z][i] +
                    Average value of PSN_EDGE_AD_CAM_IN(EX)[z][i] for all rising
                    camshaft signal edges
        Else    no update of PSN_ENG_CAM_LIH_CRK
        Endif
End
```

23.10.2.15 Engine backwards rotation detection ()

General information:

The aim of this function is to detect a change in the direction of engine rotation which would lead to a false crankshaft teeth counter value and thus to a wrong timing for injection and/or ignition events. This could occur at low engine speed with hard compression phases.
This functionality gets a lower performance if camshaft target wheel is symmetrical as only the tests on crankshaft signal can be done (Action 1 and Test 2).
i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Application conditions:

Initialisation: at reset or engine stalling (LV_STOP_ENG transition from 0 to 1)
LV_ENG_BACK_DET = 0
LV_ENG_BACK_CFM = 0

Recurrence: 10ms and every camshaft edge (for all camshaft sensors) and every reference gap and at engine stalling (LV_STOP_ENG transition from 0 to 1)

Activation: LC_ENG_BACK_INH = 0

Deactivation: when application conditions are not fulfilled

Formula section:

```
If      ( N_TOOTH < C_N_NOT_REST And  LV_CRK_MISS_TOOTH = 1 )
        or LV_ENG_BACK_DET = 1
Then    LV_ENG_BACK_DET = 1
        Activation of Engine backwards rotation confirmation ()
Else    LV_ENG_BACK_DET = 0
        LV_ENG_BACK_CFM = 0
End
```

LV_CRK_MISS_TOOTH is reset after reading.

Engine backwards rotation confirmation ():

The following table shows the test or action to perform in function of the current event.The event is a detection of a camshaft edge (CAM) or a detection of reference gap (GAP) or an engine stall (Engine stall).

Event	GAP	Engine stall	CAM
Test / Action	Action 1	Test 2	Test 3

Action 1:

This event means that the reference gap has been detected at the expected position (the number of teeth between 2 gaps is correct) and thus that engine backwards rotation is not confirmed.

LV_ENG_BACK_DET = 0

Test 2:

This event means that the number of teeth between 2 gaps is not correct due to engine backwards rotation which is thus confirmed.

```
If      LV_LOST_SYN_CRK = 1
Then    LV_ENG_BACK_CFM = 1
```

Test 3:

This test is only done on the camshaft signal that generated the event. It checks wether camshaft signal position is correct versus crankshaft signal or versus previous camshaft signal. If not, engine backwards rotation is confirmed.

```
If      LV_ERR_CAM_IN(EX)[i] = 0
Then    if      {
                ( LV_SYN_CAM_IN(EX)[i] = 1          'z value, which represents camshaft
                edge number, is know as camshaft is self-synchronized
```

```
                And
                ( PSN_ENG_ENSD ≤ NC_PSN_EDGE_CAM_IN(EX)[z][i]  -
                C_DYW_CAM_CRK_SYN_ADC_IN(EX)
                Or PSN_ENG_ENSD ≥ NC_PSN_EDGE_CAM_IN(EX)[z][i] +
                C_DYW_CAM_CRK_SYN_RTD_IN(EX) ) )
                Or
                LV_ORNG_RATIO_CAM_IN(EX)[i] = 1 }
        Then    LV_ENG_BACK_CFM = 1
        Else    if      LV_SYN_CAM_IN(EX)[i] = 1
                Then    LV_ENG_BACK_DET = 0
                End
        End
End
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
LC_INH_LIH_CRK	1	0...1H	0...1	1	[-]
Inhibition of crankshaft limp-home mode					
LC_INH_LIH_CAM	1	0...1H	0...1	1	[-]
Inhibition of camshaft limp-home mode					
C_T_DLY_STOP_ENG	1	0...FFH	0...2550	10	[ms]
Delay time for engine running after engine stop request					
ID_NR_VLD_CAM_SYN_CRK	2	0...FFH	0...255	1	[-]
LDP_N_ID_NR_VLD_CAM_SYN_CRK	2	0...1FE0H	0...8160	1	[rpm]
Number of camshaft signal edges to validate cam/crk synchronization					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PRI_SYN_CAM_IN	1	0...1H	0...1	1	[-]
Priority for synchronization mode on intake camshaft sensor(s)					
NC_PRI_SYN_CAM_CBK	1	1...2H	1...2	1	[-]
Priority for synchronization on camshaft sensor(s) of cylinder bank 1 or 2					
NC_PRI_LIH_CAM_IN	1	0...1H	0...1	1	[-]
Priority for limp-home mode on intake camshaft sensor(s)					
NC_PRI_LIH_CAM_CBK	1	1...2H	1...2	1	[-]
Priority for limp-home mode on camshaft sensor(s) of cylinder bank 1 or 2					
NLC_CAM_IN	1	0...1H	0...1	1	[-]
Intake camshaft sensor(s) present in the system					
NLC_CAM_EX	1	0...1H	0...1	1	[-]
Exhaust camshaft sensor(s) present in the system					
NC_NR_CAM_CBK	1	1...2H	1...2	1	[-]
Camshaft sensor(s) present on one or two cylinder banks					
NLC_LIH_CAM_IN	1	0...1H	0...1	1	[-]
Intake camshaft sensor(s) available for crankshaft limp-home					
NLC_LIH_CAM_EX	1	0...1H	0...1	1	[-]
Exhaust camshaft sensor(s) available for crankshaft limp-home					

23.11 Engine Position and Speed Calculation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_32	V/O	0...FFH	0...8160	32	[rpm]
Engine speed -Resolution 32 rpm					
N	V/O	0...1FE0H	0...8160	1	[rpm]
Engine speed - resolution 1 rpm					
N_FAST	V/O	0...1FE0H	0...8160	1	[rpm]
Fast engine speed					
N_TOOTH	V/O	0...1FE0H	0...8160	1	[rpm]
Engine Speed calculated on last crankshaft tooth period (update rate 10ms)					
N_MMV	V/O	0...1FE0H	0...8160	1	[rpm]
Engine Speed ¿ moving mean value					
N_CRK_WIN	-	0...1FE0H	0...8160	1	[rpm]
Engine speed on crankshaft tooth window at end of segment					
N_GRD	V/O	80...7FH	-4096...4064	32	[rpm/s]
Engine speed gradient					
N_GRD_H_RES	V/O	F000...0FFFH	-4096...4095	1	[rpm/s]
High resolution engine speed gradient					
T_REV_AV	-	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Actual revolution time					
T_REV_PREV	-	0...7FFFFFH	0...1.9965	0.238e-6	[s]
Previous revolution time					
T_SEG_AV	V/O	0...FFFFH	0...0.26214	0.000004	[s]
Segment period (saturated at low engine speed).					
T_SEG_HALF_AV	V/O	0...FFFFH	0...0.26214	0.000004	[s]
Exported half-segment period					
SEG_CTR	V/O	0...FFFFH	0...65535	1	[-]
Continuous segment counter (saturated at maximum value)					
SEG_NR	V/O	0...7H	0...7	1	[-]
Actual segment number (from 0 to NC_CYL_NR-1)					
PSN_ENG_SYN_MIN	V/O	0...780H	0...720	0.375	[°CRK]
Minimum engine position during crankshaft synchronization phase.					
PSN_ENG_SYN_MAX	V/O	0...780H	0...720	0.375	[°CRK]
Maximum engine position during crankshaft synchronization phase.					
PSN_ENG_ENSD	O	0...2CFFH	0...719.9375	0.0625	[°CRK]
Actual engine position in fine resolution					
PSN_ENG	O	0...780H	0...720	0.375	[°CRK]
Exported engine position					

Input data:

LV_RUN_ENG	LV_CRK_SYN	NC_NR_TOOTH	PSN_ENG_CRK
PSN_ENG_SYN_CAM_MIN	PSN_ENG_SYN_CAM_MAX	PSN_ENG_CAM_LIH_CRK	T_TOOTH
T_SEG_HALF_ENSD	T_CRK_WIN_ENSD	LV_STOP_ENG	NC_NR_TOOTH
NC_PSN_SEG_TDC_REF	NC_CRK_WIN_SEG_LEN	T_SEG_ENSD	LV_SEG_NR_UPD_REQ

FUNCTION DESCRIPTION:

At engine stalling (LV_STOP_ENG 0 to 1), all output data are set to initialization value.

23.11.1 Engine Position Calculation Standard Mode

General information:

The function provides output values for engine position in standard operating mode.

Application conditions:

Initialisation: PSN_ENG = PSN_ENG_ENSD = 0

Recurrence: every update of the corresponding input values

Formula section:

PSN_ENG = PSN_ENG_ENSD (resolution conversion)

PSN_ENG_ENSD = PSN_ENG_CRK + ID_CRK_PHA_COR

PSN_ENG_SYN_MIN = PSN_ENG_SYN_CAM_MIN

PSN_ENG_SYN_MAX = PSN_ENG_SYN_CAM_MAX

The resolution of variables that are exported outside the aggregate is different from the resolution used internally!

23.11.2 Engine Position Calculation Crankshaft Limp-Home Mode

General information:

The function provides output values for engine position in crankshaft limp-home mode.

Application conditions:

Initialisation: PSN_ENG = PSN_ENG_ENSD = 0

Recurrence: every update of the corresponding input values

Formula section:

PSN_ENG = PSN_ENG_ENSD (resolution conversion)

PSN_ENG_ENSD = PSN_ENG_CAM_LIH_CRK

PSN_ENG_SYN_MIN = PSN_ENG_SYN_CAM_MIN

PSN_ENG_SYN_MAX = PSN_ENG_SYN_CAM_MAX

The resolution of variables that are exported outside the aggregate is different from the resolution used internally!

23.11.3 Segment Period Calculation

General information:

The function provides output values for the segment period.

The engine cycle is divided into a number of TDC periods corresponding to the number of cylinders NC_CYL_NR. The corresponding time intervals T_SEG_AV are measured between tooth events located NC_PSN_SEG_TDC_REF degrees before each TDC.

The segment number SEG_NR indicates the number of the cylinder that will be next in the firing order. A control signal is generated to trigger execution of segment synchronous tasks.

Application conditions:

Initialisation: T_SEG_AV is set to the maximum value

SEG_CTR = 0

SEG_NR = 0

Activation: LV_RUN_ENG = 1

Recurrence : when LV_RUN_ENG is set
And when LV_CRK_SYN is set
And every segment trigger

Formula section:

T_SEG_AV = T_SEG_ENSD saturated to maximum value at low speed.

At engine synchronization, SEG_NR is initialized with current segment number.

SEG_CTR is incremented at every segment.

23.11.4 Segment number correction in camshaft limp home

General information:

This function updates SEG_NR value in camshaft limp home mode depending on synchronization determination test result.

Application conditions:

Initialisation: none

Recurrence: none

Activation: LV_SEG_NR_UPD_REQ = 1

Deactivation: LV_SEG_NR_UPD_REQ = 0

Formula section:

SEG_NR = SEG_NR + NC_CYL_NR / (2 * NC_NR_GAP)

23.11.5 Engine speed N, N_32, N_MMV

General information:

The engine speed is determined in each segment from the segment period T_SEG_ENSD.

Application conditions:

Activation: LV_RUN_ENG = 1

Recurrence: when LV_RUN_ENG is set
And when LV_CRK_SYN is set
And every segment trigger

Formula section:

N = $\frac{2*60}{NC\_CYL\_NR*T\_SEG\_ENSD}$ for T_SEG_ENSD in s

N_32 = N in resolution 32 rpm

Before synchronization and at synchronization, one engine speed is output at the moment when crankshaft rotation is validated. The output value will be calculated from the most recent tooth period (see “Segment Period Calculation”)

N_MMV is the moving mean value calculated on the last 720° CRK:

N_MMV = $\frac{N_n + N_{n-1} + \ldots + N_{n-NC\_CYL\_NR}}{NC\_CYL\_NR}$

Before engine synchronisation N_MMV = N. After engine synchronisation and before the first engine cycle is completed, N_MMV is calculated as the mean value of the most recent segments of this cycle.

Set N_FAST to N if no fast engine speed calculation is required.

```
If      C_N_FAST_SWI = 0
Then    N_FAST = N
Endif
```

23.11.6 Fast Engine speed based on last tooth period

General information:

Provide engine speed calculated every 10ms on the last crankshaft tooth period during cranking, synchronisation and start phase. Calculation is done until speed threshold is reached.
If, during running, Engine speed becomes lower than this threshold, this calculation is started again
In the formula after, T_TOOTH in s is the time for the last crankshaft tooth period, acquired in BSW just before the 10ms trigger event used for update rate of N_TOOTH here.

Application conditions:

Initialisation: N_TOOTH = 0

Recurrence : 10ms

Activation: LV_RUN_ENG = 1

Deactivation: LV_RUN_ENG = 0

Formula section:

```
If      N_32 <= C_N_TOOTH_END
Then    N_TOOTH = 60 / (NC_NR_TOOTH * T_TOOTH)
```

```
Endif
If      C_N_FAST_SWI = 1
Then    N_FAST = N_TOOTH
Endif
```

23.11.7 Fast Engine Speed based on Half-Segment Period Calculation

General information:

This function provides fast engine speed based on half-segment period.

Application conditions:

Initialisation: T_SEG_HALF_AV is set to the maximum value

Recurrence: before synchronization: once at transition from LV_RUN_ENG from 0 to 1
after synchronization: every ½ segment trigger **and** segment trigger

Activation: LV_RUN_ENG = 1

Formula section:

```
If      C_N_FAST_SWI = 2
Then    N_FAST = $\frac{60}{NC\_CYL\_NR * T\_SEG\_HALF\_ENSD}$
Endif
```

T_SEG_HALF_AV = T_SEG_HALF_ENSD saturated to maximum value at low speed.

23.11.8 Fast Engine Speed based on Crankshaft Tooth Window

General information:

This function provides fast engine speed output based on a crankshaft window at the end of each segment. Updated every segment

Application conditions:

Initialisation: N_FAST = 0

Recurrence : every segment trigger

Activation: LV_RUN_ENG = 1

And

(C_N_FAST_SWI = 3 **Or** C_N_FAST_SWI = 4)

Deactivation: C_N_FAST_SWI <> 3 **And** C_N_FAST_SWI <> 4

Formula section:

$$N\_CRK\_WIN = \frac{60}{T\_CRK\_WIN\_ENSD} \quad \frac{NC\_CRK\_WIN\_SEG\_LEN}{360° \; CRK}$$

```
If      C_N_FAST_SWI = 3
Then    N_FAST = N + C_FAC_N_FAST * (N_CRK_WIN n – N_CRK_WIN n-1)
Endif
If      C_N_FAST_SWI = 4
Then    N_FAST = N + C_FAC_N_FAST * (N n – N_CRK_WIN n-1)
Endif
```

23.11.9 Engine speed gradient N_GRD, N_GRD_H_RES

General information:

The engine speed gradient is the acceleration of the crankshaft in rpm/s. For engins with an even number of cylinders (2...8), the gradient is calculated from the duration of the most recent engine revolution, and the duration of the previous engine revolution. With this calculation, the gradient is free from errors due to mechanical tolerances of the crankshaft target wheel.

For three-cylinder engins, the gradient is calculated from the most recent segment time, and the segment time before. Errors from mechanical tolerances of the crankshaft target wheel are not corrected, but are supposed to have a minor effect due to the greater angle of the segments.

Application recurrence: every segment trigger

Formula section:

For even cylinder numbers, the duration of the most recent engine revolution is the sum of the most recent NC_CYL_NR/2 segment times. For e.g. a four-cylinder engine, this is:

$$T\_REV\_AV = T\_SEG\_ENSD_{n} + T\_SEG\_ENSD_{n-1}$$

The duration of the previous engine revolution is for the example of a four-cylinder engine:

$$T\_REV\_PREV = T\_SEG\_ENSD_{n-2} + T\_SEG\_ENSD_{n-3}$$

N_GRD calculation:

N_GRD_H_RES =

$$\frac{60*(T\_REV\_PREV - T\_REV\_AV)}{T\_REV\_PREV * T\_REV\_AV * 0.5*(T\_REV\_PREV + T\_REV\_AV)}$$

for T_REV_AV and T_REV_PREV in s

N_GRD = N_GRD_H_RES in resolution 32 rpm/s.

The resolution of the segment time introduces an error into the calculation of the engine speed gradient, depending on engine speed.

For a timer resolution of 0.25µs, a difference of 1 timer tick between T_REV_AV and T_REV_PREV corresponds to:

0.015 rpm/s at 600 rpm

and 15 rpm/s at 6000 rpm

For three-cylinder engins, the gradient is calculated as follows:

N_GRD_H_RES =

$$\frac{2*60*(T\_SEG\_ENSD_{n-1} - T\_SEG\_ENSD_{n})}{3*T\_SEG\_ENSD_{n-1}*T\_SEG\_ENSD_{n}*0.5*(T\_SEG\_ENSD_{n-1}+T\_SEG\_ENSD_{n})}$$

N_GRD = N_GRD_H_RES in resolution 32 rpm/s.

The resolution of the segment time introduces an error into the calculation of the engine speed gradient, depending on engine speed.

A difference of 1 timer tick between $T\_SEG\_ENSD_{n}$ and $T\_SEG\_ENSD_{n-1}$ corresponds to:

0.51 rpm/s at 600 rpm

and 51 rpm/s at 6000 rpm

23.11.10 Crankshaft Sensor Phase Angle Correction

General information:

The sensor phase angle is defined as the rotation angle of the target wheel between the mechanical reference and the electrical reference for a tooth of the target wheel.

The mechanical reference in the target wheel drawing usually is the trailing edge of a tooth in the defined rotation direction. The mechanical reference position is reached when the center of the sensor is aligned to that angular reference.

The electrical reference may be on the falling or on the rising edge of the sensor output signal, see active edge definition in the acquisition of crankshaft signal. The reference position is reached when the sensor output signal switches (for active sensors) respectively has its zero crossing (for VR sensors).

A retard of the electrical reference relative to the mechanical edge is defined as a positive phase angle. With most sensors, the signal edge is generated at the center of each tooth at low speed. At high speed the signal edge may be retarded.

The typical phase angle ID_CRK_PHA_COR (at typical air gap and operating temperature) of the used sensor/target wheel combination is calibrated in a table versus engine speed. The value corresponding to the actual engine speed is added to the actual engine position delivered from crankshaft acquisition, in order to obtain a corrected engine position (see position calculation in the first chapter of this specification).

Application conditions:

Recurrence: every segment trigger

Activation: LV_CRK_SYN = 1

Deactivation: LV_CRK_SYN = 0

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_TOOTH_END	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for N_TOOTH calculation at engine start					
C_N_FAST_SWI	1	0...4H	0...4	1	[-]
Switch case choice for N_FAST calculation mode					
C_FAC_N_FAST	1	0...FFH	0...0.99609	3.91E-03	[-]
Factor for fast engine speed prediction					
ID_CRK_PHA_COR	16	0...FFH	0...15.9375	0.0625	[°CRK]
LDP_N_32_ID_CRK_PHA_COR	16	0...FFH	0...8160	32	[rpm]
Table for crankshaft transmitter phase offset correction					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CYL_NR	1	1...8H	1...8	1	[-]
count of cylinder					

23.12 Synchronization determination in camshaft limp home

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CAM_LIH_EXT_ENA	V/O	0...1H	0...1	1	[-]
Camshaft limp home with external help enabled					
LV_SEG_NR_UPD_REQ	V/O	0...1H	0...1	1	[-]
SEG_NR update request					
LV_SYN_VLD_CAM_LIH	V/O	0...1H	0...1	1	[-]
Engine synchronization determined in camshaft limp home mode					

Input data:

N	N_GRD	SEG_CTR	LV_VS_RUN
INH_INJ	TCO_CMN	VB_CMN	SEG_NR
NC_CAM_LIH_SWI			

FUNCTION DESCRIPTION:

General information:

This functions aims at initializing the variables used by ENSD or others aggregates in case no camshaft limp home method has been selected.

Description:

Application conditions:

Initialisation: none

Recurrence: at reset

Activation: NC_CAM_LIH_SWI = “None”

Deactivation: NC_CAM_LIH_SWI ≠ “None”

Formula section:

LV_SYN_VLD_CAM_LIH = 0
LV_CAM_LIH_EXT_ENA = 0
LV_SEG_NR_UPD_REQ = 0

Remark: as this module is only a stub, N, N_GRD, SEG_CTR, LV_VS_RUN, INH_INJ, TCO_CMN, VB_CMN and SEG_NR are not used.

23.13 Camshaft adaptation and position output

Output data:

Name	Mode	Hex. Limit	Phys. Limit	Resol.	Unit
PSN_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1FFH	-96...95.625	0.375	[°CRK]
Actual intake camshaft position relative to adapted passive position					
PSN_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1FFH	-96...95.625	0.375	[°CRK]
Actual exhaust camshaft position relative to adapted passive position					
PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	O	0...FFFH	-128...127.9375	0.0625	[°CRK]
Actual intake camshaft position relative to adapted passive position at signal edge index z					
PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	O	0...FFFH	-128...127.9375	0.0625	[°CRK]
Actual exhaust camshaft position relative to adapted passive position at signal edge index z					
PSN_EDGE_AD_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	V/O/S	0...3FFH	-32...31.9375	0.0625	[°CRK]
Adapted intake camshaft i signal edge z position					
PSN_EDGE_AD_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	V/O/S	0...3FFH	-32...31.9375	0.0625	[°CRK]
Adapted exhaust camshaft i signal edge z position					
PSN_AD_CAM_IN[NC_NR_CAM_CBK]	V/O	0...FFH	-48...47.625	0.375	[°CRK]
Adapted intake camshaft position relative to designed passive position					
PSN_AD_CAM_EX[NC_NR_CAM_CBK]	V/O	0...FFH	-48...47.625	0.375	[°CRK]
Adapted exhaust camshaft position relative to designed passive position					
PSN_DIF_EDGE_CAM_IN[NC_NR_CAM_CBK]	V/O	0...780H	0...720	0.375	[°CRK]
Crankshaft angle between the previous and the current intake camshaft position determination					
PSN_DIF_EDGE_CAM_EX[NC_NR_CAM_CBK]	V/O	0...780H	0...720	0.375	[°CRK]
Crankshaft angle between the previous and the current exhaust camshaft i position determination					
T_DIF_EDGE_CAM_IN[NC_NR_CAM_CBK]	V/O	0...FFFFFFFFH	0...17179.86918	0.000004	[s]
Time between the previous and current camshaft signal edge					
T_DIF_EDGE_CAM_EX[NC_NR_CAM_CBK]	V/O	0...FFFFFFFFH	0...17179.86918	0.000004	[s]
Time between the previous and current camshaft signal edge					
CTR_REV_AD_REF_CAM_IN[NC_NR_CAM_CBK]	V	0...FFH	0...255	1	[-]
Intake camshaft reference position adaptation revolution counter					
CTR_REV_AD_REF_CAM_EX[NC_NR_CAM_CBK]	V	0...FFH	0...255	1	[-]
Exhaust camshaft reference position adaptation revolution counter					
LV_AD_END_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Intake camshaft reference position adaptation successful					
LV_AD_END_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Exhaust camshaft reference position adaptation successful					
PSN_ENG_CAM_IN[NC_NR_CAM_CBK]	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Engine position at the last intake camshaft i edge					
PSN_ENG_CAM_EX[NC_NR_CAM_CBK]	-	0...2CFFH	0...719.9375	0.0625	[°CRK]
Engine position at the last exhaust camshaft i edge					
CAM_PSN_LST_REF_AD_IN[NC_NR_CAM_CBK]	V/O	0...3FFH	-32...31.9375	0.0625	[°CRK]
Adapted intake camshaft i signal position after last correct reference adaptation					
CAM_PSN_LST_REF_AD_EX[NC_NR_CAM_CBK]	V/O	0...3FFH	-32...31.9375	0.0625	[°CRK]
Adapted exhaust camshaft i signal position after last correct reference adaptation					
LV_CAM_AD_SAVE_IN[NC_NR_CAM_CBK]	V/O/S	0...1H	0...1	1	[-]
Confirmation that adaptative values at intake i have already been stored					
LV_CAM_AD_SAVE_EX[NC_NR_CAM_CBK]	V/O/S	0...1H	0...1	1	[-]

Confirmation that adaptative values at exhaust i have already been stored					
LV_CAM_AD_PWL_NOT_SAVE_IN[NC_NR_CAM_CBK]	-	0...1H	0...1	1	[-]
Camshaft adaptive values storage in NVMY not allowed, for intake i					
LV_CAM_AD_PWL_NOT_SAVE_EX[NC_NR_CAM_CBK]	-	0...1H	0...1	1	[-]
Camshaft adaptive values storage in NVMY not allowed, for exhaust i					
LV_TOOTH_OFF_DET_ENA_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Confirmation that a reference adaptation occurred after the very first one, to allow the one_tooth_off detection (intake i)					
LV_TOOTH_OFF_DET_ENA_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Confirmation that a reference adaptation occurred after the very first one, to allow the one_tooth_off detection (exhaust i)					

Input data:

LV_DI_AD_REF_CAM_IVVT_IN	LV_DI_AD_REF_CAM_IVVT_EX	IDX_EDGE_CAM_IN[NC_NR_CAM_CBK]	IDX_EDGE_CAM_EX[NC_NR_CAM_CBK]
NC_NR_EDGE_CAM_IN	NC_NR_EDGE_CAM_EX	LV_VLD_PSN_CAM_IN[NC_NR_CAM_CBK]	LV_VLD_PSN_CAM_EX[NC_NR_CAM_CBK]
NC_PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	NC_PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	PSN_ENG_ENSD	LV_SYN_VLD
LC_NOT_ADJ_CAM_IVVT_IN[NC_NR_CBK_IVVT]	LC_NOT_ADJ_CAM_IVVT_EX[NC_NR_CBK_IVVT]	NLC_IVVT_IN	NLC_IVVT_EX
T_SEG_CAM_IN[NC_NR_CAM_CBK]	T_SEG_CAM_EX[NC_NR_CAM_CBK]	N_32	C_TOL_REF_CRK_CAM_IN
C_TOL_REF_CRK_CAM_EX			

FUNCTION DESCRIPTION:

General information:

The function provides a feedback about the actual camshaft position relative to the engine position measured with the crankshaft sensor. The signals of one, two or four camshaft sensors are treated (intake and/or exhaust camshaft on one or two cylinder banks).

The actual engine position is captured with every camshaft signal edge. The camshaft position is calculated as the difference between the captured engine position, and the engine position measured on the respective camshaft. The difference may be due to tolerances, or due to an intentional displacement by a camshaft phasing system (VVTI).

Two different adaptive learning algorithms eliminate the tolerances, in order to measure exactly the VVTI displacement.

The reference position adaptation is done for all signal edges. It is performed on all intake camshafts as long as LV_DI_AD_REF_CAM_IVVT_IN = 0. It is performed on all exhaust camshafts as long as LV_DI_AD_REF_CAM__IVVT_EX = 0. The camshafts are then situated in their respective VVTI passive positions. After the completion of the adaptation LV_AD_END_CAM_IN(EX)[i] is set and PSN_AD_CAM_IN(EX)[i] is calculated as the mean value of all adaptated signal edges. The reference edge has the index 1. It is the first electrical falling edge of the camshaft signal after TDC0.

The position of other signal edges is adapted continuously relative to the reference signal edge, in order to always deliver a coherent position information.

Both adaptation algorithms are performed only at a limited dynamic variation of the camshaft position measured at the reference edge.

The logical calibration bit LC_NOT_ADJ_CAM_IVVT_IN(EX)[i] is set if the VVTI system is inhibited on the corresponding camshafts. The respective camshafts will be treated like being constantly in the VVTI passive position. If an engine has more camshaft sensors then VVTI actuators, then NLC_IVVT_IN(EX) = 0 indicates that VVTI is not existing on intake or on exhaust camshafts.

23.13.1 Engine Position Determination from Camshaft signal

General information:

As soon as camshaft is self-synchronized, the actual engine position is calculated at every camshaft signal edge from theorical position plus adaptation value for the actual signal edge index.

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Description:

Recurrence : *every CAM active edge*

Formula section:

for z = IDX_EDGE_CAM_IN(EX)[i]

PSN_ENG_CAM_IN(EX)[i] (n) = NC_PSN_EDGE_CAM_IN(EX)[z][i] +
PSN_EDGE_AD_CAM_IN(EX)[z][i] (n-1) +
ID_CAM_PHA_COR

23.13.2 Camshaft Position Output

Application conditions:

Activation:

LV_VLD_PSN_CAM_IN(EX)[i] = 1 **And** LV_SYN_VLD = 1

Recurrence: *every update of IDX_EDGE_CAM_IN(EX)[i]*

Initialization:

PSN_CAM_IN(EX)[i] = 0
PSN_EDGE_CAM_IN(EX)[z][i] = 0

Formula section:

for z = IDX_EDGE_CAM_IN(EX)[i]

PSN_EDGE_CAM_IN(EX)[z][i] = PSN_ENG_ENSD - PSN_ENG_CAM_IN(EX)[i]

In the following calculations, the resolution has to be converted:

PSN_CAM_IN(EX)[i] = PSN_EDGE_CAM_IN(EX)[z][i]

PSN_DIF_EDGE_CAM_IN(EX)[i] = PSN_ENG_ENSD (n) - PSN_ENG_ENSD (n-1)

T_DIF_EDGE_CAM_IN(EX)[i] = T_SEG_CAM_IN(EX)[i]

23.13.3 Camshaft Position Adaptation

23.13.3.1 Reference Position Adaptation

General information:

The reference position adaptation is done whenever it is not disabled by the corresponding flag from VVTI, or if there is no VVTI on that camshaft, and if the condition for limited dynamic variation of the camshaft position is true. A flag is set to signal successful adaptation.

The flag is reset when the adaptation inhibit conditions become false.

If LV_DI_AD_REF_CAM_IVVT_IN(EX) = 1

Then LV_AD_END_CAM_IN(EX)[i] = 0

The camshaft position output value should be zero in VVTI passive position. At each ocurrence of the reference camshaft signal edge, a portion of the position deviation is added to the previous adaptation value, as long as adaptation conditions are true, this is done for all the camshaft signal edges.

If the conditions stay true for a number of revolutions, then the flag for successful end of adaptation is set (LV_AD_END_CAM_IN(EX)[i] = 1).

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Application conditions:

Activation:

LV_VLD_PSN_CAM_IN(EX)_I =1 **And** LV_SYN_VLD = 1

And

LV_DI_AD_REF_CAM_IVVT_IN(EX) = 0 **Or** LC_NOT_ADJ_CAM_IVVT_IN(EX)[i] = 1 **Or** NLC_IVVT_IN(EX) = 0

Recurrence: every reference camshaft signal edge (IDX_EDGE_CAM_IN(EX)[i] = 1)

Initialization:

LV_AD_END_CAM_IN(EX)[i] = 0

CTR_REV_AD_REF_CAM_IN(EX)[i] = 0

PSN_EDGE_AD_CAM_IN(EX)[z][i] from saved value in NVMY (if adaptation value has never been learned or if the stored value could not be read, PSN_EDGE_AD_CAM_IN(EX)[z][i] are initialized to 0)

PSN_AD_CAM_IN(EX)[i] = Sum **(** PSN_EDGE_AD_CAM_IN(EX)[z][i]

for z = 1…NC_NR_EDGE_CAM_IN(EX) **)** / NC_NR_EDGE_CAM_IN(EX)

(resolution conversion)

LV_CAM_AD_SAVE_IN(EX)[i] from saved value in NVMY (if adaptation value has never been learned or if the stored value could not be read, LV_CAM_AD_SAVE_IN(EX)[i] is initialized to 0)

PSN_EDGE_AD_CAM_IN(EX)[z][i] and LV_CAM_AD_SAVE_IN(EX)[i] shall be saved in NVMY only if: LV_CAM_AD_PWL_NOT_SAVE_IN(EX)[i] =0 (otherwise the last stored values, which have been stored after last driving cycle without error, shall not be changed)

At reset: CAM_PSN_LST_REF_AD_IN(EX)[i] = PSN_EDGE_AD_CAM_IN(EX)[1][i]

(All these adaptive values has to be reset in case of cam/crk mechanical change (Chain change, camshaft sensor change…))

Formula section:

```
If      |PSN_EDGE_CAM_IN(EX)[1][i] (n) – PSN_EDGE_CAM_IN(EX)[1][i] (n-1)| < C_GRD_AD_REF_MAX_CAM_IN(EX)
Then    increment CTR_REV_AD_REF_CAM_IN(EX)[i]
        The adaptation values are calculated for all indexes (for z = 1…NC_NR_EDGE_CAM_IN(EX))
        PSN_EDGE_AD_CAM_IN(EX)[z][i] (n) = PSN_EDGE_AD_CAM_IN(EX)[z][i] (n-1) + C_CRLC_AD_REF_CAM_IN(EX) * PSN_EDGE_CAM_IN(EX)[z][i]
Else    CTR_REV_AD_REF_CAM_IN(EX)[i] = 0
Endif

If      CTR_REV_AD_REF_CAM_IN(EX)[i] >= C_NR_REV_AD_REF_CAM_IN(EX)
Then    CTR_REV_AD_REF_CAM_IN(EX)[i] = 0
        PSN_AD_CAM_IN(EX)[i] = Sum ( PSN_EDGE_AD_CAM_IN(EX)[z][i]
        for z = 1…NC_NR_EDGE_CAM_IN(EX) ) / NC_NR_EDGE_CAM_IN(EX)
        (resolution conversion)
        If      LV_CAM_AD_SAVE_IN(EX)[i] = 1
        Then    If   |CAM_PSN_LST_REF_AD_IN(EX)[i] - PSN_EDGE_AD_CAM_IN(EX)[1][i]| < C_TOL_TOOTH_OFF_CAM_IN(EX)
                Then CAM_PSN_LST_REF_AD_IN(EX)[i] = PSN_EDGE_AD_CAM_IN(EX)[1][i]
                Else nothing
                Endif
                LV_TOOTH_OFF_DET_ENA_IN(EX)[i] = 1 (to allow the ONE_TOOTH_OFF diagnosis only if a reference adaptation has been done after the very first one)
        Else    If   C_TOL_TOOTH_OFF_CAM_IN(EX) <> 0
                Then      CAM_PSN_LST_REF_AD_IN(EX)[i] = PSN_EDGE_AD_CAM_IN(EX)[1][i]
                          LV_CAM_AD_SAVE_IN(EX)[i] = 1 (set at the end of the very first reference adaptation)
```

```
                Else    nothing
                Endif
        Endif
        LV_AD_END_CAM_IN(EX)[i] = 1
Else
        LV_TOOTH_OFF_DET_ENA_IN(EX)[i] = 0
Endif

If      LV_DI_AD_REF_CAM_IVVT_IN(EX) = 1
Then    CTR_REV_AD_REF_CAM_IN(EX)[i] = 0
Endif
```

23.13.3.2 Continuous Edge Position Adaptation

The continuous position adaptation is enabled at a limited dynamic variation of the camshaft position measured at the reference signal edge.

The difference of the actual camshaft position measured at the reference edge, and the actual camshaft position measured at any other signal edge, is calculated once per camshaft revolution. That difference is multiplied with a cofficient (<<1) and added to the previous adaptation value to form the new adaptation value, which is applied in the following camshaft revolution. This way the camshaft position output value calculated at any signal edges will approach the camshaft position output value calculated at the reference signal edge.

Application conditions:

Activation:

LV_VLD_PSN_CAM_IN[i] = 1 **And** LV_SYN_VLD = 1

And

LV_DI_AD_REF_CAM_IVVT_IN(EX) = 1 **And** LC_NOT_ADJ_CAM_IVVT_IN(EX)[i] = 0 **And** NLC_IVVT_IN(EX) = 1

Recurrence: *every reference camshaft signal edge (IDX_EDGE_CAM_IN(EX)[i] = 1)*

Initialization:

PSN_EDGE_AD_CAM_IN(EX)[z][i] from saved value in NVMY (if adaptation value has never been learned or if the stored value could not be read, PSN_EDGE_AD_CAM_IN(EX)[z][i] is initialized to 0)

PSN_EDGE_AD_CAM_IN(EX)[z][i] and LV_CAM_AD_SAVE_IN(EX)[i] shall be saved in NVMY only if: LV_CAM_AD_PWL_NOT_SAVE_IN(EX)[i] =0 (otherwise the last stored values, which have been stored after last driving cycle without error, shall not be changed)

Formula section:

For the following equations, the index z of the camshaft signal edges equals IDX_EDGE_CAM_IN(EX)[i] (z=1 for reference edge). The adaptation values are calculated for all indexes from 2 to NC_NR_EDGE_CAM_IN(EX).

```
If      |PSN_EDGE_CAM_IN(EX)[1][i] (n) - PSN_EDGE_CAM_IN(EX)[1][i] (n-1)| <
        C_GRD_AD_MAX_CAM_IN(EX)
Then    PSN_EDGE_AD_CAM_IN(EX)[z][i] (n)        =
        PSN_EDGE_AD_CAM_IN(EX)[z][i] (n-1) + C_CRLC_AD_CAM_IN(EX) *
        (PSN_EDGE_CAM_IN(EX)[z][i] - PSN_EDGE_CAM_IN(EX)[1][i])
Endif
```

23.13.3.3 Reference adaptive values save authorization at Powerlatch

Before saving the adaptive values in NVMY during PWL, it is important to check if they are correct. Even if the REF_CRK_CAM error is not set, the calculated adaptive values can be wrong (for example, the error flag could have been reset on a quick key 0 to 1 transition).

Application conditions:

Activation:

ECU_STATE = PWL

Recurrence: *once during PWL, for all camshaft edges of all present camshaft sensors, before saving the NVMY adaptive values*

Initialization:

LV_CAM_AD_PWL_NOT_SAVE_IN(EX)[i] =0

Formula section:

```
If      PSN_EDGE_AD_CAM_IN(EX)[z][i] < - C_TOL_REF_CRK_CAM_IN(EX)
        or      PSN_EDGE_AD_CAM_IN(EX)[z][i] > C_TOL_REF_CRK_CAM_IN(EX)
Then    LV_CAM_AD_PWL_NOT_SAVE_IN(EX)[i] = 1
Else
        If      LV_CAM_AD_SAVE_IN(EX)[i] = 1
                And
                CAM_PSN_LST_REF_AD_IN(EX)[i] <>
                PSN_EDGE_AD_CAM_IN(EX)[1][i]
                And
                C_TOL_TOOTH_OFF_CAM_IN(EX) <> 0
        Then    LV_CAM_AD_PWL_NOT_SAVE_IN(EX)[i] = 1
        Else    nothing, (if LV_CAM_AD_PWL_NOT_SAVE_IN(EX)[i] has been set to 1
                due to anyone of the camshaft edges, it must stay to 1 and  the adaptive
                will not be saved. Contrary, if all the camshaft edges are correct, adaptive
```

values can be saved, LV_CAM_AD_PWL_NOT_SAVE_IN(EX)[i] must stay at 0)

Endif

Endif

Calibration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
ID_CAM_PHA_COR	16	0...5FFH	-48...47.9375	0.0625	[°CRK]
LDP_N_32_ID_CAM_PHA_COR	16	0...FFH	0...8160	32	[rpm]
Table for camshaft transmitter phase offset correction					
C_GRD_AD_MAX_CAM_IN	1	0...FFH	0...15.9375	0.0625	[°CRK]
Max. permissible drift of reference edge position for continuous adaptation					
C_GRD_AD_MAX_CAM_EX	1	0...FFH	0...15.9375	0.0625	[°CRK]
Max. permissible drift of reference edge position for continuous adaptation					
C_GRD_AD_REF_MAX_CAM_IN	1	0...FFH	0...15.9375	0.0625	[°CRK]
Max. permissible drift of reference edge position for reference position adaptation					
C_GRD_AD_REF_MAX_CAM_EX	1	0...FFH	0...15.9375	0.0625	[°CRK]
Max. permissible drift of reference edge position for reference position adaptation					
C_CRLC_AD_CAM_IN	1	0...FFH	0...0.99609	3.9063e-3	[-]
Averaging constant for continuous adaptation					
C_CRLC_AD_CAM_EX	1	0...FFH	0...0.99609	3.9063e-3	[-]
Averaging constant for continuous adaptation					
C_CRLC_AD_REF_CAM_IN	1	0...FFH	0...0.99609	3.9063e-3	[-]
Averaging constant for reference position adaptation					
C_CRLC_AD_REF_CAM_EX	1	0...FFH	0...0.99609	3.9063e-3	[-]
Averaging constant for reference position adaptation					
C_NR_REV_AD_REF_CAM_IN	1	0...FFH	0...255	1	[-]
Number of camshaft revolutions for reference position adaptation					
C_NR_REV_AD_REF_CAM_EX	1	0...FFH	0...255	1	[-]
Number of camshaft revolutions for reference position adaptation					
C_TOL_TOOTH_OFF_CAM_IN	1	0...1FFH	0...31.9375	0.0625	[°CRK]
Reference position tolerance for tooth off detection for intake camshaft					
C_TOL_TOOTH_OFF_CAM_EX	1	0...1FFH	0...31.9375	0.0625	[°CRK]
Reference position tolerance for tooth off detection for exhaust camshaft					

23.14 Engine Position and Speed Diagnosis Manager

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CAM_TOT	V/O	0...1H	0...1	1	[-]
Failure of at least one camshaft sensor on a camshaft with IVVT actuator					
LV_ERR_CRK	V/O	0...1H	0...1	1	[-]
Crankshaft sensor failure					
LV_ERR_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Failure of intake camshaft sensor i					
LV_ERR_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Failure of exhaust camshaft sensor i					
LV_INH_DIAG_SYN_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of intake camshaft i synchronization diagnosis					
LV_INH_DIAG_SYN_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of exhaust camshaft i synchronization diagnosis					
LV_INH_DIAG_PER_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of intake camshaft i period diagnosis					
LV_INH_DIAG_PER_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of exhaust camshaft i period diagnosis					
LV_INH_DIAG_PLAUS_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of intake camshaft i plausibility diagnosis					
LV_INH_DIAG_PLAUS_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of exhaust camshaft i plausibility diagnosis					
LV_INH_DIAG_TOOTH_OFF_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of exhaust camshaft i one-tooth-off diagnosis					
LV_INH_DIAG_TOOTH_OFF_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of exhaust camshaft i one-tooth-off diagnosis					
LV_INH_DIAG_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of intake camshaft i position diagnosis					
LV_INH_DIAG_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of exhaust camshaft i position diagnosis					
LV_INH_DIAG_CRK_SYN	V/O	0...1H	0...1	1	[-]
Inhibition of Crankshaft sensor synchronization diagnosis					
LV_INH_DIAG_CRK_TOOTH	V/O	0...1H	0...1	1	[-]
Inhibition of Crankshaft sensor tooth number diagnosis					
LV_INH_DIAG_CRK_TOOTH_PER	V/O	0...1H	0...1	1	[-]
Inhibition of Crankshaft sensor tooth period diagnosis					
LV_INH_DIAG_CRK_PLAUS	V/O	0...1H	0...1	1	[-]
Inhibition of Crankshaft sensor plausibility diagnosis					
LV_INH_DIAG_CRK_OC	V/O	0...1H	0...1	1	[-]
Inhibition of Crankshaft sensor circuit diagnosis					
LV_INH_DIAG_ENSD	V/O	0...1H	0...1	1	[-]
Inhibition of cam/crk diagnosis during fail safe delay					
LV_IGN_INJ_LOCK_REQ	V/O	0...1H	0...1	1	[-]
Request to lock ignition and/or injection due to backwards rotation detection					
LV_INH_DIAG_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of intake camshaft sensor diagnosis for crankshaft synchronization					

LV_INH_DIAG_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	V/O	0...1H	0...1	1	[-]
Inhibition of exhaust camshaft sensor diagnosis for crankshaft synchronization					

Input data:

LV_ERR_CRK_SYN	LV_ERR_CRK_TOOTH	LV_ERR_CRK_PLAUS	LV_ERR_CRK_OC
NLC_IVVT_IN	NLC_IVVT_EX	LV_ERR_SYN_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_SYN_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_PER_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_PER_CAM_EX[NC_NR_CAM_CBK]	LC_NOT_ADJ_CAM_IVVT_IN[NC_NR_CBK_IVVT]	LC_NOT_ADJ_CAM_IVVT_EX[NC_NR_CBK_IVVT]
LV_ERR_PLAUS_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_PLAUS_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_REF_CRK_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_CRK_TOOTH_PER	LV_IGK
LV_STOP_ENG	N_TOOTH	LV_ORNG_CAM_SYN_CRK	LV_ENG_BACK_DET
LV_ENG_BACK_CFM	LV_SYN_VLD	LV_ORNG_TOOTH_PER_CRK	NC_NR_CAM_CBK
LV_SYN_ENG	LV_ERR_TOOTH_OFF_IN[NC_NR_CAM_CBK]	LV_ERR_TOOTH_OFF_EX[NC_NR_CAM_CBK]	

23.14.1 Calculation of Fail Safe Delay

General information:

Calculation of the time delay to inhibit cam/crk diagnosis when there is a risk of reverse engine rotation.

Application conditions:

Initialisation:	no initialization value.
Recurrence:	10ms **and** 1 to 0 transition of LV_IGK
Activation:	no condition
Deactivation:	no condition

Formula section:

```
If      LV_IGK = 0   And   LV_INH_DIAG_ENSD = 0
Then    check N_TOOTH every 10ms
        If      0 < N_TOOTH < C_N_NOT_REST
        Then    Inhibition of all diagnosis functions except sensor circuit diagnosis
                LV_INH_DIAG_ENSD = 1
                Wait for a time C_T_DLY_REST_IGK
                LV_INH_DIAG_ENSD = 0
        Endif
Endif
```

```
If      LV_STOP_ENG = 1   And LV_ENG_BACK_DET = 0
    /* to be checked before LV_ENG_BACK_DET reset on LV_STOP_ENG transition */
        And
        /* don't activate fail safe delay if cam/crk synchronization is out of range */
        LV_ORNG_CAM_SYN_CRK = 0
        And
        /* don't activate fail safe delay if tooth period is out of range */
        LV_ORNG_TOOTH_PER_CRK = 0
        And
        LV_INH_DIAG_ENSD = 0
Then    If      0 < N_TOOTH < C_N_NOT_REST
                        (N_TOOTH: most recently calculated engine speed)
        Then    Inhibition of all diagnosis functions except sensor circuit diagnosis
                LV_INH_DIAG_ENSD = 1
                Wait for a time C_T_DLY_REST_STALL
                Stop timer C_T_DLY_REST_STALL at 0 to 1 transition of LV_IGK
                LV_INH_DIAG_ENSD = 0
        Endif
Endif
```

23.14.2 Diagnosis general inhibition in case of engine backwards rotation detection

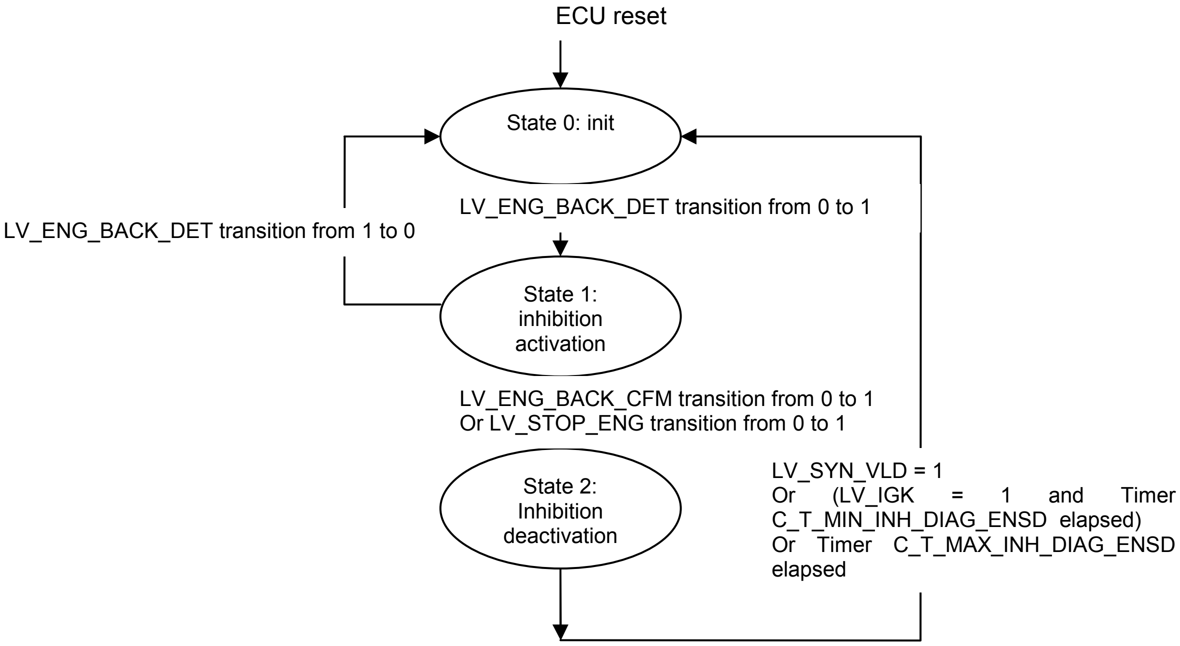
General information:

Inhibits cam/crk diagnosis when a reverse engine rotation has been detected until confirmation/nullification.

Application conditions:

Initialisation:	no initialization value.
Recurrence:	10ms **and** transition from 1 to 0 of LV_IGK **and** every update of LV_ENG_BACK_DET or LV_ENG_BACK_CFM **and** transition from 0 to 1 of LV_SYN_VLD **and** transition from 0 to 1 of LV_STOP_ENG
Activation:	LC_ENG_BACK_INH = 0
Deactivation:	when application conditions are not fulfilled

Formula section:



State 0: Initialisation

Input conditions:

External event: ECU reset

From state 1: LV_ENG_BACK_DET transition from 1 to 0

From state 2: LV_SYN_VLD = 1

Or (LV_IGK = 1 and timer C_T_MIN_INH_DIAG_ENSD elapsed)

Or timer C_T_MAX_INH_DIAG_ENSD elapsed

Output conditions:

To state 1: LV_ENG_BACK_DET transition from 0 to 1

And Timer C_T_DLY_REST_IGK stopped

Action in the state:

Set timer to C_T_MIN_INH_DIAG_ENSD

Set timer to C_T_MAX_INH_DIAG_ENSD

Transition action:

None

State 1: Inhibition activation

Input conditions:

From state 0: LV_ENG_BACK_DET transition from 0 to 1

Output conditions:

To state 0: ECU reset

Or LV_ENG_BACK_DET transition from 1 to 0

To state 2: LV_ENG_BACK_CFM transition from 0 to 1

Or LV_STOP_ENG transition from 0 to 1

Action in the state:

LV_INH_DIAG_ENSD = 1

LV_IGN_INJ_LOCK_REQ = 1

Stop timer C_T_MIN_INH_DIAG_ENSD if on-going

Stop timer C_T_MAX_INH_DIAG_ENSD if on-going

Transition action:

To state 0: LV_INH_DIAG_ENSD = 0

LV_IGN_INJ_LOCK_REQ = 0

To State 2: Start timer C_T_MAX_INH_DIAG_ENSD

Start timer C_T_MIN_INH_DIAG_ENSD if LV_IGK = 0

State 2: Inhibition deactivation

Input conditions:

From state 1: LV_ENG_BACK_CFM transition from 0 to 1

Or LV_STOP_ENG transition from 0 to 1

Output conditions:

To state 0: ECU reset

Or LV_SYN_VLD = 1

Or (LV_IGK = 1 and timer C_T_MIN_INH_DIAG_ENSD elapsed)

Or timer C_T_MAX_INH_DIAG_ENSD elapsed

Action in the state:

Start timer C_T_MIN_INH_DIAG_ENSD on transition of LV_IGK from 1 to 0

Transition action:

To state 0: LV_INH_DIAG_ENSD = 0

LV_IGN_INJ_LOCK_REQ = 0

Remark: if C_T_MIN_INH_DIAG_ENSD = 0, then timer is started at transition of LV_IGK from 1 to 0 and considered as elapsed immediately.

23.14.3 Diagnosis Manager

General information:

One purpose of the diagnosis manager is to provide a summary of diagnostic information, in order to facilitate taking the necessary actions in case of a failure, e.g. switch to limp-home mode. The corresponding outputs are crankshaft limp-home request a general camshaft error flag for IVVT, and a general camshaft error for synchronization.

The other purpose is to avoid getting multiple error codes and freeze frames for the same failure. The corresponding outputs are the inhibit flags for the various diagnosis functions.

i = 1...NC_NR_CAM_CBK

Application conditions:

Initialisation: no initialization value.

Recurrence: every change in the error flags of one of the diagnostic functions

Or

when error memory is cleared

Or

At key 0 to 1 transition

Activation:

Deactivation:

Formula section:

LV_ERR_CRK

LV_ERR_CRK = 1 **if** LV_ERR_CRK_SYN = 1 **or** LV_ERR_CRK_TOOTH_PER = 1 **or** LV_ERR_CRK_TOOTH = 1 **or** LV_ERR_CRK_PLAUS = 1 **or** LV_ERR_CRK_OC = 1

LV_ERR_CRK = 0 **if** LV_ERR_CRK_SYN = 0 **and** LV_ERR_CRK_TOOTH_PER = 0 **and** LV_ERR_CRK_TOOTH = 0 **and** LV_ERR_CRK_PLAUS = 0 **and** LV_ERR_CRK_OC = 0.

If LV_ERR_CRK = 1, all other diagnoses on the crankshaft signal will be inhibited via the interfaces LV_INH_DIAG_CRK_SYN, LV_INH_DIAG_CRK_TOOTH, LV_INH_DIAG_CRK_TOOTH_PER, LV_INH_DIAG_CRK_PLAUS and LV_INH_DIAG_CRK_OC, except the one that detected the failure.

LV_ERR_CAM_IN(EX)[i]

LV_ERR_CAM_IN(EX)[i] = 1 **if** LV_ERR_SYN_CAM_IN(EX)[i] = 1 **or** LV_ERR_PER_CAM_IN(EX)[i] = 1 **or** LV_ERR_PLAUS_CAM_IN(EX)[i] = 1 **or** LV_ERR_REF_CRK_CAM_IN(EX)[i] = 1 **or** LV_ERR_SYN_CRK_CAM_IN(EX)[i] = 1 **or** LV_ERR_TOOTH_OFF_IN(EX)[i] = 1.

LV_ERR_CAM_IN(EX)[i] = 0 **if** LV_ERR_SYN_CAM_IN(EX)[i] = 0 **and** LV_ERR_PER_CAM_IN(EX)[i] = 0 **and** LV_ERR_PLAUS_CAM_IN(EX)[i] = 0 **and** LV_ERR_REF_CRK_CAM_IN(EX)[i] = 0 **and** LV_ERR_SYN_CRK_CAM_IN(EX)[i] = 0 **and** LV_ERR_TOOTH_OFF_IN(EX)[i] = 0.

If LV_ERR_CAM_IN(EX)[i] = 1, other diagnoses on the same camshaft signal will be inhibited via the interfaces LV_INH_DIAG_SYN_CAM_IN(EX)[i], LV_INH_DIAG_PER_CAM_IN(EX)[i], LV_INH_DIAG_PLAUS_CAM_IN(EX)[i], LV_INH_DIAG_REF_CRK_CAM_IN(EX)[i] and LV_INH_DIAG_TOOTH_OFF_IN(EX)[i], except the one that detected the failure.

LV_ERR_CAM_TOT

VVT actuators existing in the system are given by a logical combination of the configuration data NLC_IVVT_IN(EX) and the calibration data LC_NOT_ADJ_CAM_IVVT_IN(EX)[i].

LV_ERR_CAM_TOT = 1 if LV_ERR_CAM_IN(EX)[i] = 1 for one of the sensors on camshafts with IVVT actuator.

LV_ERR_CAM_TOT = 0 if LV_ERR_CAM_IN(EX)[i] = 0 for all sensors on camshafts with IVVT actuator.

23.14.4 Inhibition of camshaft sensor diagnosis for crankshaft synchronization

General information:

Calculation of the inhibition flag of camshaft sensor diagnosis for crankshaft synchronization in case of possible reverse rotation.

I = 1...NC_NR_CAM_CBK

Application conditions:

Initialisation: LV_INH_DIAG_SYN_CRK_CAM_IN(EX)[i] = 0 at ECU reset **and** 0 to 1 transition of LV_IGK

Recurrence: 10ms **and** transition from 0 to 1 of LV_STOP_ENG **and** transition from 0 to 1 of LV_SYN_VLD

Activation: -

Deactivation: -

Formula section:

```
If     LV_SYN_ENG = 0
       And    LC_ENG_BACK_INH = 0
       And    LV_INH_DIAG_ENSD = 0
       And    0 < N_TOOTH < C_N_NOT_REST
       And    LV_ORNG_TOOTH_PER_CRK = 1
Then   LV_INH_DIAG_SYN_CRK_CAM_IN(EX)[i] = 1
Endif

If     LV_SYN_VLD = 1
Then   LV_INH_DIAG_SYN_CRK_CAM_IN(EX)[i] = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_NOT_REST	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for cam/crk diagnosis deactivation					
C_T_DLY_REST_IGK	1	0...FFH	0...2550	10	[ms]
Delay time for diagnosis reactivation after key-off at low engine speed					
C_T_DLY_REST_STALL	1	0...FFFFH	0...655350	10	[ms]
Delay time for diagnosis reactivation after engine stalling					
C_T_MIN_INH_DIAG_ENSD	1	0...FFFFH	0...655350	10	[ms]
Minimum delay before reactivating diagnosis in case of key off to on after engine backwards rotation confirmation					
C_T_MAX_INH_DIAG_ENSD	1	0...FFFFH	0...655350	10	[ms]
Maximum delay before reactivating diagnosis after engine backwards rotation confirmation					
LC_ENG_BACK_INH	1	0...1H	0...1	1	[-]
Inhibition of engine backwards rotation detection					

23.15 Camshaft Sensor Diagnosis

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CDN_DIAG_PER_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition					
LV_CDN_DIAG_PER_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition					
ERR_SYM_PER_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the failure camshaft period too short					
ERR_SYM_PER_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the failure camshaft period too short					
LV_ERR_PER_CAM_IN[NC_NR_CAM_CBK]	V/O/S	0H 1H	PASSIVE ACTIVE	1	[-]
Present failure camshaft period too short					
LV_ERR_PER_CAM_EX[NC_NR_CAM_CBK]	V/O/S	0H 1H	PASSIVE ACTIVE	1	[-]
Present failure camshaft period too short					
LV_END_DIAG_PER_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of the failure camshaft period too short					
LV_END_DIAG_PER_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of the failure camshaft period too short					
LV_CDN_DIAG_SYN_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition for intake camshaft i					
LV_CDN_DIAG_SYN_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition for exhaust camshaft i					
ERR_SYM_SYN_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the intake camshaft i sensor diagnosis					
ERR_SYM_SYN_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the exhaust camshaft i sensor diagnosis					
LV_ERR_SYN_CAM_IN[NC_NR_CAM_CBK]	V/O/S	0H 1H	PASSIVE ACTIVE	1	[-]
Present intake camshaft i sensor failure					
LV_ERR_SYN_CAM_EX[NC_NR_CAM_CBK]	V/O/S	0H 1H	PASSIVE ACTIVE	1	[-]
Present exhaust camshaft i failure sensor out of range					
LV_END_DIAG_SYN_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of intake camshaft i sensor diagnosis					

LV_END_DIAG_SYN_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of exhaust camshaft i sensor diagnosis					

Input data:

LV_ORNG_RATIO_CAM_IN[NC_NR_CAM_CBK]	LV_ORNG_PER_CAM_IN[NC_NR_CAM_CBK]	LV_ORNG_RATIO_CAM_EX[NC_NR_CAM_CBK]	LV_ORNG_PER_CAM_EX[NC_NR_CAM_CBK]
LV_ACT_CAM_IN[NC_NR_CAM_CBK]	LV_ACT_CAM_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_SYN_CAM_IN[NC_NR_CAM_CBK]	LV_INH_DIAG_PER_CAM_IN[NC_NR_CAM_CBK]
LV_INH_DIAG_SYN_CAM_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_PER_CAM_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_ENSD	LV_INH_DIAG_EXT_SYN_CAM_IN[NC_NR_CAM_CBK]
LV_INH_DIAG_EXT_SYN_CAM_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_EXT_PER_CAM_IN[NC_NR_CAM_CBK]	LV_INH_DIAG_EXT_PER_CAM_EX[NC_NR_CAM_CBK]	

23.15.1.1 Camshaft Segment Period Diagnosis

FUNCTION DESCRIPTION:

General information:

Camshaft signal acquisition validates the segment period by minimum limits derived from the maximum engine speed. The diagnostic state indicates when a segment period was too short, e.g. due to supplementary signal transitions. The camshaft signal acquisition will filter out such signal transitions as far as possible. If there are too many additional pulses however, the camshaft signal acquisition may not be able to synchronize any more, or the accuracy of camshaft position measurement may be degraded.

i = 1...NC_NR_CAM_CBK

Description:

Error-symptoms are defined for this diagnostic function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Camshaft segment period too short "CAM_PER"    (= SYM_0)
 │ │ └───> (not used)                                     (= SYM_1)
 │ └─────> (not used)                                     (= SYM_2)
 └───────> (not used)                                     (= SYM_3)
```

Application conditions:

Initialisation:
LV_CDN_DIAG_PER_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_PER_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
LV_ERR_PER_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
ERR_SYM_PER_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*

Recurrence: every camshaft signal edge

Activation: LV_ACT_CAM_IN(EX)[i] = 1
And LV_INH_DIAG_PER_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_EXT_PER_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_ENSD = 0

When the activation conditions are fulfilled then LV_CDN_DIAG_PER_CAM_IN(EX)[i] = 1

Deactivation:

When the activation conditions are not fulfilled then *LV_CDN_DIAG_PER_CAM_IN(EX)[i] = 0*

Formula section:

Symptoms calculation:

```
If    LV_ORNG_PER_CAM_IN(EX)[i] = 1
Then    ERR_SYM_PER_CAM_IN(EX)[i] = "SYM_0" {detection of symptom CAM_PER]
Else    ERR_SYM_PER_CAM_IN(EX)[i] = NO_SYM
Endif
```

LV_ORNG_PER_CAM_IN(EX)[i] is reset after reading.

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Filtering:

```
Apply filter on current symptoms
If filtering result available (after debounce)
Then
    LV_ERR_PER_CAM_IN(EX)[i] = filtering result
    LV_END_DIAG_PER_CAM_IN(EX)[i] = 1
    Deliver the result to Error Management
Endif
```

Configuration for diagnostic symptoms:

See Appl. Inc.

23.15.1.2 Camshaft Synchronization Diagnosis

FUNCTION DESCRIPTION:

General information:

The purpose of the function is to detect camshaft sensor failure. Each time the camshaft signal synchronization algorithm fails, it will set the diagnostic flag LV_ORNG_RATIO_CAM_IN(EX)[i] for the appropriate camshaft.

The camshaft signal acquisition will filter out supplementary signal transitions as far as possible. If there are too many additional pulses however, the camshaft signal acquisition will not be able to synchronize any more. Synchronization may also be impossible due to missing or inaccurate signal transitions of the sensor.

i = 1...NC_NR_CAM_CBK

Description:

Error-symptoms are defined for this diagnostic function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 |  |  |  └─> Camshaft synchronization failure "SYNC_LOSS"     (= SYM_0)
 |  |  └────> (not used)                                       (= SYM_1)
 |  └───────> (not used)                                       (= SYM_2)
 └──────────> (not used)                                       (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_SYN_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_SYN_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
LV_ERR_SYN_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
ERR_SYM_SYN_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*

Recurrence: *every camshaft signal edge*

Activation: LV_ACT_CAM_IN(EX)[i] = 1
And LV_INH_DIAG_SYN_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_EXT_SYN_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_ENSD = 0

When activation conditions are fulfilled then LV_CDN_DIAG_SYN_CAM_IN(EX)[i] = 1.

Deactivation:

When activation conditions are not fulfilled then LV_CDN_DIAG_SYN_CAM_IN(EX)[i] = 0.

Formula section:

Symptoms calculation:

If LV_ORNG_RATIO_CAM_IN(EX)[i] = 1

Then ERR_SYM_SYN_CAM_IN(EX)[i] = "SYM_0" {Detection of symptom SYNC_LOSS}

Else ERR_SYM_SYN_CAM_IN(EX)[i] = NO_SYM

```
Endif
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Filtering:

```
Apply filter on current symptoms
If filtering result available (after debounce)
Then
    LV_ERR_SYN_CAM_IN(EX)[i] = filtering result
    LV_END_DIAG_SYN_CAM_IN(EX)[i] = 1
    Deliver the result to Error Management
Endif
```

Configuration for diagnostic symptoms:

See Appl. Inc.

Calibration data

See Appl. Inc.

23.16 Camshaft Position Diagnosis

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CDN_DIAG_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition for intake camshaft i					
LV_CDN_DIAG_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition for exhaust camshaft i					
ERR_SYM_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the intake camshaft i position diagnosis					
ERR_SYM_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the exhaust camshaft i position diagnosis					
LV_ERR_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Present intake camshaft i failure reference violation					
LV_ERR_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Present exhaust camshaft i failure reference violation					
LV_END_DIAG_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of intake camshaft i position diagnosis					
LV_END_DIAG_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of exhaust camshaft i position diagnosis					
LV_CDN_DIAG_TOOTH_OFF_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
One-tooth-off diagnosis condition for intake camshaft i					
LV_CDN_DIAG_TOOTH_OFF_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
One-tooth-off diagnosis condition for exhaust camshaft i					
ERR_SYM_TOOTH_OFF_IN[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the intake i one-tooth-off diagnosis					
ERR_SYM_TOOTH_OFF_EX[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the exhaust i one-tooth-off diagnosis					
LV_ERR_TOOTH_OFF_IN[NC_NR_CAM_CBK]	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Present intake camshaft i one-tooth-off failure					
LV_ERR_TOOTH_OFF_EX[NC_NR_CAM_CBK]	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Present exhaust camshaft i one-tooth-off failure					
LV_END_DIAG_TOOTH_OFF_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of intake camshaft i one-tooth-off diagnosis					

LV_END_DIAG_TOOTH_OFF_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of exhaust camshaft i one-tooth-off diagnosis					

Input data:

PSN_EDGE_AD_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	PSN_EDGE_AD_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	LV_VLD_PSN_CAM_IN[NC_NR_CAM_CBK]	LV_VLD_PSN_CAM_EX[NC_NR_CAM_CBK]
LV_INH_DIAG_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_INH_DIAG_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_ENSD	LV_INH_DIAG_EXT_REF_CRK_CAM_IN[NC_NR_CAM_CBK]
LV_INH_DIAG_EXT_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	CAM_PSN_LST_REF_AD_IN[NC_NR_CAM_CBK]	CAM_PSN_LST_REF_AD_EX[NC_NR_CAM_CBK]	LV_TOOTH_OFF_DET_ENA_IN[NC_NR_CAM_CBK]
LV_TOOTH_OFF_DET_ENA_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_TOOTH_OFF_IN[NC_NR_CAM_CBK]	LV_INH_DIAG_TOOTH_OFF_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_EXT_TOOTH_OFF_IN[NC_NR_CAM_CBK]
LV_INH_DIAG_EXT_TOOTH_OFF_EX[NC_NR_CAM_CBK]			

23.16.1 Camshaft Position diagnosis

FUNCTION DESCRIPTION:

General information:

The purpose of the function is to detect when the camshaft reference position is outside of the designed range relative to the engine position from crankshaft. This allows to detect engine misbuilds (wrong assembly of the valve train), camshaft drive defects, or sensor failures that are not detected by the normal signal diagnosis.

i = 1...NC_NR_CAM_CBK   z = 0...NC_NR_EDGE_CAM_IN/EX

Description:

Error-symptoms are defined for this diagnostic function as following:

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │  │  │  └─> Camshaft to crankshaft reference violated "REF_CRK_CAM"  (= SYM_0)
  │  │  └────> (not used)                                              (= SYM_1)
  │  └───────> (not used)                                              (= SYM_2)
  └──────────> (not used)                                              (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_REF_CRK_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_REF_CRK_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
LV_ERR_REF_CRK_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
ERR_SYM_REF_CRK_CAM_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*

Recurrence: *every update of the camshaft edge position adaptation values PSN_EDGE_AD_CAM_IN(EX)[z][i]*

Activation: LV_VLD_PSN_CAM_IN(EX)[i] = 1
And LV_INH_DIAG_REF_CRK_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_EXT_REF_CRK_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_ENSD = 0

When the activation conditions are fulfilled then LV_CDN_DIAG_REF_CRK_CAM_IN(EX)[i] = 1.

Deactivation:

When the activation conditions are not fulfilled then LV_CDN_DIAG_REF_CRK_CAM_IN(EX)[i] = 0

Formula Section

Symptoms calculation:

```
If      PSN_EDGE_AD_CAM_IN(EX)[z][i] < - C_TOL_REF_CRK_CAM_IN(EX)
        or
        PSN_EDGE_AD_CAM_IN(EX)[z][i] > C_TOL_REF_CRK_CAM_IN(EX)
Then    ERR_SYM_REF_CRK_CAM_IN(EX)[i] = “SYM_0” {Detection of symptom
        REF_CRK_CAM}
Else    ERR_SYM_REF_CRK_CAM_IN(EX)[i] = NO_SYM
Endif
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Filtering :

```
apply filter on current symptoms
If filtering result available  (after debounce)
then
    LV_ERR_REF_CRK_CAM_IN(EX)[i] = filtering result
    LV_END_DIAG_REF_CRK_CAM_IN(EX)[i] = 1
    Deliver the result to Error Management
Endif
```

Configuration for diagnostic symptoms :

See Appl. Inc.

23.16.2 One-tooth-off diagnosis

FUNCTION DESCRIPTION:

General information:

The purpose of the function is to detect a sudden drift is the camshaft position.

i = 1...NC_NR_CAM_CBK z = 0...NC_NR_EDGE_CAM_IN/EX

Description:

Error-symptoms are defined for this diagnostic function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 |  |  |  └─> Sudden drift "ONE_TOOTH_OFF"        (= SYM_0)
 |  |  └────> (not used)                           (= SYM_1)
 |  └───────> (not used)                           (= SYM_2)
 └──────────> (not used)                           (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_TOOTH_OFF_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_TOOTH_OFF_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
LV_ERR_TOOTH_OFF_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*
ERR_SYM_TOOTH_OFF_IN(EX)[i] = *Refer to filtering configuration for the initialisation value*

Recurrence: at LV_TOOTH_OFF_DET_ENA_IN(EX)[i] 0 to 1 transition

Activation: And LV_INH_DIAG_TOOTH_OFF_IN(EX)[i] = 0
And LV_INH_DIAG_EXT_TOOTH_OFF_IN(EX)[i] = 0
And LV_INH_DIAG_ENSD = 0

When the activation conditions are fulfilled then LV_CDN_DIAG_TOOTH_OFF_IN(EX)[i] = 1.

Deactivation:

When the activation conditions are not fulfilled then
LV_CDN_DIAG_TOOTH_OFF_IN(EX)[i] = 0

Formula Section

Symptoms calculation:

```
If      CAM_PSN_LST_REF_AD_IN(EX)[i] = PSN_EDGE_AD_CAM_IN(EX)[1][i]
        Then    ERR_SYM_TOOTH_OFF_IN(EX)[i] = NO_SYM
Else    ERR_SYM_TOOTH_OFF_IN(EX)[i] = “SYM_0” {Detection of symptom
        ONE_TOOTH_OFF}
Endif
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Filtering :

```
apply filter on current symptoms
If filtering result available  (after debounce)
then
    LV_ERR_ TOOTH_OFF_IN(EX)[i] = filtering result
```

```
        LV_END_DIAG_ TOOTH_OFF_IN(EX)[i] = 1
        Deliver the result to Error Management
Endif
```

Configuration for diagnostic symptoms :

See Appl. Inc.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TOL_REF_CRK_CAM_IN	1	0...1FFH	0...31.9375	0.0625	[°CRK]
Reference position tolerance range for intake camshaft					
C_TOL_REF_CRK_CAM_EX	1	0...1FFH	0...31.9375	0.0625	[°CRK]
Reference position tolerance range for exhaust camshaft					

For filter-type-linked calibrations: see Appl. Inc.

23.17 Camshaft Sensor Diagnosis (Appl. Inc.)

23.17.1 External Inhibitions and filter type definitions

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_EXT_PER_CAM_IN[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft period diagnosis (intake)					
LV_INH_DIAG_EXT_PER_CAM_EX[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft period diagnosis (exhaust)					
LV_INH_DIAG_EXT_SYN_CAM_IN[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft self-synchronization diagnosis (intake)					
LV_INH_DIAG_EXT_SYN_CAM_EX[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft self-synchronization diagnosis (exhaust)					
LV_INH_DIAG_EXT_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft position diagnosis (intake)					
LV_INH_DIAG_EXT_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft position diagnosis (exhaust)					
LV_INH_DIAG_EXT_PLAUS_CAM_IN[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft sensor plausibility diagnosis (intake)					
LV_INH_DIAG_EXT_PLAUS_CAM_EX[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft sensor plausibility diagnosis (exhaust)					
LV_INH_DIAG_EXT_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft sensor diagnosis for crankshaft synchronization (intake)					
LV_INH_DIAG_EXT_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the camshaft sensor diagnosis for crankshaft synchronization (exhaust)					
LV_INH_DIAG_EXT_TOOTH_OFF_IN[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the one-tooth-off diagnosis (intake i)					
LV_INH_DIAG_EXT_TOOTH_OFF_EX[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Inhibition condition for the one-tooth-off diagnosis (exhaust i)					
LV_ERR_EXT_CAM_EX[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Failure on exhaust camshaft sensor due to external issue					
LV_ERR_EXT_CAM_IN[NC_NR_CAM_CBK]	O/V	0...1H	0...1	1	[-]
Failure on intake camshaft sensor due to external issue					

Input data:

LV_ERR_PER_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_PER_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_SYN_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_SYN_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_PLAUS_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_PLAUS_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_TOOTH_OFF_IN[NC_NR_CAM_CBK]	LV_ERR_TOOTH_OFF_EX[NC_NR_CAM_CBK]

FUNCTION DESCRIPTION:

General information:

The flags LV_INH_DIAG_EXT_... allow to deactivate the corresponding diagnostic.

Application conditions:

Initialisation: At reset : all = 0

Recurrence: once at reset

Activation: -/-

Deactivation: When the activation condition is not fulfilled

Formula section:

LV_INH_DIAG_EXT_PER_CAM_IN(EX)[i] = 0
LV_INH_DIAG_EXT_SYN_CAM_IN(EX)[i] = 0
LV_INH_DIAG_EXT_REF_CRK_CAM_IN(EX)[i] = 0
LV_INH_DIAG_EXT_PLAUS_CAM_IN(EX)[i] = 0
LV_INH_DIAG_EXT_SYN_CRK_CAM_IN(EX)[i] = 0
LV_INH_DIAG_EXT_TOOTH_OFF_IN(EX)[i] = 0
LV_ERR_EXT_CAM_IN(EX)[i] = 0

Configuration for diagnostic symptoms :

Diagnostic Camshaft sensor	Symptom description	Symptom	Filter type
Camshaft segment period diagnosis for intake	*PER_CAM_IN*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	
Camshaft segment period diagnosis for exhaust	*PER_CAM_EX*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	
Camshaft synchronization diagnosis for intake	*SYN_CAM_IN*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	
Camshaft synchronization diagnosis for exhaust	*SYN_CAM_EX*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	
Camshaft position diagnosis for intake	*REF_CRK_CAM_IN*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	
Camshaft position diagnosis for exhaust	*REF_CRK_CAM_EX*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	
One-tooth-off diagnosis for intake	*TOOTH_OFF*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	
One-tooth-off	*TOOTH_OFF*	*SYM_0*	*MEM*

diagnosis for exhaust		*SYM_1*	
		SYM_2	
		SYM_3	

Note:

For filter type definition of PLAUS_CAM_IN(EX) and SYN_CRK_CAM_IN(EX) see subchapters "Camshaft sensor plausibility diagnosis" and " Camshaft Sensor Diagnosis for Crankshaft Synchronization"

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_DIAG_REF_CRK_CAM_IN	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for intake camshaft position diagnosis					
C_ABC_INC_DIAG_REF_CRK_CAM_EX	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for exhaust camshaft position diagnosis					
C_ABC_MAX_DIAG_REF_CRK_CAM_IN	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for intake camshaft position diagnosis					
C_ABC_MAX_DIAG_REF_CRK_CAM_EX	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for exhaust camshaft position diagnosis					
C_ABC_INC_DIAG_PER_CAM_IN	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for intake camshaft segment period diagnosis					
C_ABC_INC_DIAG_PER_CAM_EX	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for exhaust camshaft segment period diagnosis					
C_ABC_MAX_DIAG_PER_CAM_IN	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for intake camshaft segment period diagnosis					
C_ABC_MAX_DIAG_PER_CAM_EX	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for exhaust camshaft segment period diagnosis					
C_ABC_INC_DIAG_SYN_CAM_IN	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for intake camshaft synchronization diagnosis					
C_ABC_INC_DIAG_SYN_CAM_EX	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for exhaust camshaft synchronization diagnosis					
C_ABC_MAX_DIAG_SYN_CAM_IN	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for intake camshaft synchronization diagnosis					
C_ABC_MAX_DIAG_SYN_CAM_EX	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for exhaust camshaft synchronization diagnosis					
C_ABC_INC_DIAG_TOOTH_OFF_IN	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for intake one-tooth-off diagnosis					
C_ABC_INC_DIAG_TOOTH_OFF_EX	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for exhaust one-tooth-off diagnosis					
C_ABC_MAX_DIAG_TOOTH_OFF_IN	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for intake one-tooth-off diagnosis					
C_ABC_MAX_DIAG_TOOTH_OFF_EX	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for exhaust one-tooth-off diagnosis					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_IDX_DIAG_PER_CAM_EX[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for PER_CAM_EX diagnosis					
NC_IDX_DIAG_PER_CAM_IN[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for PER_CAM_IN diagnosis					
NC_IDX_DIAG_PLAUS_CAM_EX[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for PLAUS_CAM_EX diagnosis					
NC_IDX_DIAG_PLAUS_CAM_IN[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for PLAUS_CAM_IN diagnosis					
NC_IDX_DIAG_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for REF_CRK_CAM_EX diagnosis					
NC_IDX_DIAG_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for REF_CRK_CAM_IN diagnosis					
NC_IDX_DIAG_SYN_CAM_EX[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]

Failure index for SYN_CAM_EX diagnosis					
NC_IDX_DIAG_SYN_CAM_IN[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for SYN_CAM_IN diagnosis					
NC_IDX_DIAG_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for SYN_CRK_CAM_EX diagnosis					
NC_IDX_DIAG_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for SYN_CRK_CAM_IN diagnosis					
NC_IDX_DIAG_TOOTH_OFF_EX[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for TOOTH_OFF_EX diagnosis					
NC_IDX_DIAG_TOOTH_OFF_IN[NC_NR_CAM_CBK]	-	0...FFFFH	0...65535	1	[-]
Failure index for TOOTH_OFF_IN diagnosis					

23.17.2 Interface of TOOTH_OFF_IN/EX error for Rate-based-monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_TOOTH_OFF_IN[NC_NR_CAM_CBK]	O/V	0...7H	0...7	1	[-]
Interface of TOOTH_OFF_IN monitor with the Rate-Based Monitoring statistics					
STATE_RBM_TOOTH_OFF_EX[NC_NR_CAM_CBK]	O/V	0...7H	0...7	1	[-]
Interface of TOOTH_OFF_EX monitor with the Rate-Based Monitoring statistics					

Input data:

LV_DC	CTR_ERR_DYN_NR	LV_END_DIAG_TOOTH_OFF_IN[NC_NR_CAM_CBK]	LV_END_DIAG_TOOTH_OFF_EX[NC_NR_CAM_CBK]
LV_INH_DIAG_TOOTH_OFF_IN[NC_NR_CAM_CBK]	LV_INH_DIAG_TOOTH_OFF_EX[NC_NR_CAM_CBK]		

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the TOOTH_OFF_IN/EX[i] monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TOOTH_OFF_IN/EX[i] data.

Within STATE_RBM_TOOTH_OFF_IN/EX[i] , three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for TOOTH_OFF_IN/EX[i] diagnosis)

***Remark**: Calculation must be made seperate for ..._EX[i] and ..._IN[i]*

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0 and bit 1 of STATE_RBM_TOOTH_OFF_IN/EX[i] = 0

bit 2 of STATE_RBM_TOOTH_OFF_IN/EX[i] = 1

on failure memory reset :

bit 1 of STATE_RBM_TOOTH_OFF_IN/EX[i] = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_SYN_CAM_IN/EX[i]
LV_ERR_PER_CAM_IN/EX[i]
LV_ERR_PLAUS_CAM_IN/EX[i]
LV_ERR_SYN_CRK_CAM_IN/EX[i]
LV_ERR_REF_CRK_CAM_IN/EX[i]

```
If(1)      { CPU optimization at LV_DC 0 → 1 transition }
           CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
           While  bit 1 of STATE_RBM_TOOTH_OFF_IN/EX[i]  = 0 do
                  with each XX failure of the above list :
```

```
                ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                SYNCHRONIZATION<CALL>)
                If(2)      XX has a pending status
                Then(2)    bit 1 of STATE_RBM_TOOTH_OFF_IN/EX[i] = 1
                Endif(2)
            Endwhile
    Else(1) { the dynamic failure memory is empty }
            No action
    Endif(1)
```

Every 1 s :

```
If   bit 0 of STATE_RBM_TOOTH_OFF_IN/EX[i] = 0
Then       If     LV_END_DIAG_TOOTH_OFF_IN/EX[i] = 1
           Then   bit 0 of STATE_RBM_TOOTH_OFF_IN/EX[i] = 1
           Endif
Endif

If   bit 1 of STATE_RBM_TOOTH_OFF_IN/EX[i] = 0
Then
           If     LV_INH_DIAG_TOOTH_OFF_IN/EX[i] = 1
           Then   bit 1 of STATE_RBM_TOOTH_OFF_IN/EX[i] = 1
           Endif
Endif
```

23.18 Crankshaft Sensor Diagnosis

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CDN_DIAG_CRK_TOOTH	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition					
ERR_SYM_CRK_TOOTH	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the crankshaft failure tooth number error					
LV_INH_MIS_CRK	V/O	0...1H	0...1	1	[-]
Inhibition of misfire detection with crankshaft tooth number error					
LV_ERR_CRK_TOOTH	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Present crankshaft failure tooth number error					
LV_END_DIAG_CRK_TOOTH	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of crankshaft tooth number error					
LV_CDN_DIAG_CRK_TOOTH_PER	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition for crankshaft diagnosis of implausible tooth period					
ERR_SYM_CRK_TOOTH_PER	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the crankshaft failure implausible tooth period					
LV_ERR_CRK_TOOTH_PER	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Present crankshaft failure tooth number error					
LV_END_DIAG_CRK_TOOTH_PER	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of crankshaft diagnosis implausible tooth period					
LV_CDN_DIAG_CRK_SYN	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition for crankshaft diagnosis with loss of synchronization					
ERR_SYM_CRK_SYN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the crankshaft failure with loss of synchronization					
LV_ERR_CRK_SYN	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Present crankshaft failure with loss of synchronization					
LV_END_DIAG_CRK_SYN	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of crankshaft diagnosis with loss of synchronization					

Input data:

LV_CRK_SYN	LV_RUN_ENG	LV_ORNG_TOOTH_PER_CRK	LV_ORNG_NR_TOOTH_CRK
LV_LOST_SYN_CRK	LV_INH_DIAG_CRK_SYN	LV_INH_DIAG_CRK_TOOTH	LV_INH_DIAG_CRK_TOOTH_PER
LV_LIH_ERR_CRK	LV_INH_DIAG_ENSD	LV_STOP_ENG	N_32

LV_INH_DIAG_EXT_CRK_TOOTH	LV_INH_DIAG_EXT_CRK_TOOTH_PER	LV_INH_DIAG_EXT_CRK_SYN	

General information:

The purpose of the function is to detect failures on the crankshaft signal while the system is synchronized.

The diagnosis is divided in three different parts: tooth number error, implausible tooth period, and loss of synchronization on crankshaft signal.

23.18.1.1 Crankshaft Sensor Diagnosis with Tooth Number Error

FUNCTION DESCRIPTION:

General information:

The number of teeth per crankshaft revolution is monitored by the crankshaft signal acquisition. The algorithm may tolerate a number of missing/additional teeth per revolution without loosing synchronization, depending on the used target wheel and on configuration data. The bit LV_ORNG_NR_TOOTH_CRK indicates wrong tooth number during the most recent crankshaft revolution.

If a tooth was missing or added during one revolution, then all variables based on teeth counting will be produced with an error. This concerns for example spark advance, segment time, misfire segments, camshaft position, etc. The purpose of the function is to provide information when the crankshaft signal is inaccurate, in order to take the necessary actions.

Description:

Error-symptoms are defined for this diagnostic function as following:

```
┌─┬─┬─┬─┐
│ │ │ │ │
└─┴─┴─┴─┘
 │ │ │ └─> Incorrect crankshaft tooth number "TOOTH_NUMBER"   (= SYM_0)
 │ │ └───> (not used)                                         (= SYM_1)
 │ └─────> (not used)                                         (= SYM_2)
 └───────> (not used)                                         (= SYM_3)
```

Application conditions

Initialisation:

LV_CDN_DIAG_CRK_TOOTH = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_CRK_TOOTH = *Refer to filtering configuration for the initialisation value*
LV_ERR_CRK_TOOTH = *Refer to filtering configuration for the initialisation value*
ERR_SYM_CRK_TOOTH = *Refer to filtering configuration for the initialisation value*

Recurrence: *once per crankshaft revolution, after detection of the reference gap, except in the first revolution after synchronization*

Activation: LV_CRK_SYN = 1
And LV_INH_DIAG_CRK_TOOTH = 0
And LV_INH_DIAG_EXT_CRK_TOOTH = 0
And LV_INH_DIAG_ENSD = 0
And N_32 > C_N_32_MAX_DIAG_CRK_TOOTH

When the activation conditions are fulfilled then LV_CDN_DIAG_CRK_TOOTH = 1.

Deactivation:

When the activation conditions are not fulfilled then LV_CDN_DIAG_CRK_TOOTH = 0.

Formula Section

Symptoms calculation:

```
If   LV_ORNG_NR_TOOTH_CRK = 0
Then     ERR_SYM_CRK_TOOTH = NO_SYM
Else     ERR_SYM_CRK_TOOTH = "SYM_0" {Detection of symptom TOOTH_NUMBER}
Endif
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Filtering:

```
Apply filter on current symptoms
If filtering result available (after debounce)
Then
    LV_ERR_CRK_TOOTH = filtering result
    LV_END_DIAG_CRK_TOOTH = 1
    Deliver the result to Error Management
Endif
```

Configuration for diagnostic symptoms :

See Appl. Inc.

Calculation of missing/added tooth information for misfire

Initialisation:

LV_INH_MIS_CRK = 0 at reset and at engine stalling (LV_STOP_ENG transition from 0 to 1).

Recurrence: *once per crankshaft revolution, after detection of the reference gap, except in the first revolution after synchronization*

Activation: LV_CRK_SYN = 1

Deactivation: *LV_CRK_SYN = 0*

Formula Section

Symptoms calculation:

```
If   LV_ORNG_NR_TOOTH_CRK = 0
Then     LV_INH_MIS_CRK = 0
Else     LV_INH_MIS_CRK = 1
Endif
```

LV_ORNG_NR_TOOTH_CRK is reset after reading at the reference gap

23.18.1.2 Crankshaft Sensor Diagnosis with implausible Tooth Period

FUNCTION DESCRIPTION:

General information:

Crankshaft signal acquisition validates the tooth period by limits derived from the current engine speed.

Before synchronization, LV_ORNG_TOOTH_PER_CRK indicates implausible gradient of the tooth period. This flag is also set if a reference gap occurs during valid teeth phase or when the first reference gap cannot be detected.

After synchronization, LV_ORNG_TOOTH_PER_CRK is set if spikes are detected outside the crankshaft window acceptance window.

Description:

Error-symptoms are defined for this diagnostic function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 |  |  |  └─> Implausible crankshaft tooth period "TOOTH_PERIOD"   (= SYM_0)
 |  |  └────> (not used)                                           (= SYM_1)
 |  └───────> (not used)                                           (= SYM_2)
 └──────────> (not used)                                           (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_CRK_TOOTH_PER = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_CRK_TOOTH_PER = *Refer to filtering configuration for the initialisation value*
LV_ERR_CRK_TOOTH_PER = *Refer to filtering configuration for the initialisation value*
ERR_SYM_CRK_TOOTH_PER = *Refer to filtering configuration for the initialisation value*

Recurrence: if LV_CRK_SYN = 1: every reference gap
If LV_CRK_SYN = 0: when LV_ORNG_TOOTH_PER_CRK is set

Activation: LV_RUN_ENG= 1
And LV_LIH_ERR_CRK = 0
And LV_INH_DIAG_CRK_TOOTH_PER = 0
And LV_INH_DIAG_EXT_CRK_TOOTH_PER = 0
And LV_INH_DIAG_ENSD = 0

When the activation conditions are fulfilled then LV_CDN_DIAG_CRK_TOOTH_PER = 1.

Deactivation:

When the activation conditions are not fulfilled then LV_CDN_DIAG_CRK_TOOTH_PER = 0.

Formula Section

Symptoms calculation:

Before crankshaft synchronization (LV_CRK_SYN = 0)

```
If      LV_ORNG_TOOTH_PER_CRK = 0
Then    ERR_SYM_CRK_TOOTH_PER = NO_SYM
Else    ERR_SYM_CRK_TOOTH_PER = “SYM_0" {Detection of symptom
        TOOTH_PERIOD}
Endif
```

After crankshaft synchronization (LV_CRK_SYN = 1)

```
If      LV_ORNG_TOOTH_PER_CRK = 1 set at a tooth event or end of tolerance window
Then    ERR_SYM_CRK_TOOTH_PER = “SYM_0" {Detection of symptom
        TOOTH_PERIOD}
Endif
LV_ORNG_TOOTH_PER_CRK is reset after reading
If      LV_ORNG_TOOTH_PER_CRK = 0 at the reference gap
Then    ERR_SYM_CRK_TOOTH_PER = NO_SYM
Endif
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Filtering:

```
Apply filter on current symptoms
If filtering result available (after debounce)
Then
    LV_ERR_CRK_TOOTH_PER = filtering result
    LV_END_DIAG_CRK_TOOTH_PER = 1
    Deliver the result to Error Management
Endif
```

Configuration for diagnostic symptoms :

See Appl. Inc.

23.18.1.3 Crankshaft Sensor Diagnosis with Loss of Synchronization

FUNCTION DESCRIPTION:

General information:

The purpose of the function is to detect crankshaft failure when the system looses synchronization on the crankshaft signal.

Synchronization will be lost if the reference gap is not detected at the correct position plus/minus a tolerance of missing/additional teeth, depending on the used target wheel and on configuration data.

If the signal fails completely (no more crankshaft signal edges are detected), then no error will be detected by this diagnosis, since the condition is similar to stopped engine. A failure then may be detected by the crankshaft sensor plausibility diagnosis (e.g. test against the camshaft signal).

Description:

Error-symptoms are defined for this diagnostic function as following:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 │  │  │  └─> Loss of synchronization "SYNC_LOSS"        (= SYM_0)
 │  │  └────> (not used)                                 (= SYM_1)
 │  └───────> (not used)                                 (= SYM_2)
 └──────────> (not used)                                 (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_CRK_SYN = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_CRK_SYN = *Refer to filtering configuration for the initialisation value*
LV_ERR_CRK_SYN = *Refer to filtering configuration for the initialisation value*
ERR_SYM_CRK_SYN = *Refer to filtering configuration for the initialisation value*

Recurrence: *once per crankshaft revolution, after the reference gap, except in the first revolution after synchronization, or when the bit LV_LOST_SYN_CRK is set.*

Activation: LV_CRK_SYN = 1
And LV_INH_DIAG_CRK_SYN = 0
And LV_INH_DIAG_EXT_CRK_SYN = 0
And LV_INH_DIAG_ENSD = 0

When activation conditions are fulfilled then LV_CDN_DIAG_CRK_SYN = 1.

Deactivation:

When activation conditions not are fulfilled then LV_CDN_DIAG_CRK_SYN = 0.

Formula Section

Symptoms calculation:

```
If    LV_LOST_SYN_CRK = 0
Then     ERR_SYM_CRK_SYN = NO_SYM
Else     ERR_SYM_CRK_SYN = “SYM_0" {Detection of symptom SYNC_LOSS}
Endif
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Filtering:

```
Apply filter on current symptoms
If filtering result available (after debounce)
Then
    LV_ERR_CRK_SYN = filtering result
    LV_END_DIAG_CRK_SYN = 1
    Deliver the result to Error Management
Endif
```

Configuration for diagnostic symptoms :

See Appl. Inc.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_32_MAX_DIAG_CRK_TOOTH	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed to enable tooth number diagnosis					

For filter-type-linked calibrations: see Appl. Inc.

23.19 Crankshaft Sensor Plausibility Diagnosis

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CDN_DIAG_CRK_PLAUS	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition					
ERR_SYM_CRK_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the crankshaft failure synchronization impossible					
LV_ERR_CRK_PLAUS	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Present crankshaft failure synchronization impossible					
LV_END_DIAG_CRK_PLAUS	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of crankshaft signal diagnosis synchronization impossible					
LV_CDN_DIAG_PLAUS_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition					
LV_CDN_DIAG_PLAUS_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition					
ERR_SYM_PLAUS_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the camshaft failure synchronization impossible					
ERR_SYM_PLAUS_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the camshaft failure synchronization impossible					
LV_ERR_PLAUS_CAM_IN[NC_NR_CAM_CBK]	V/O/S	0H 1H	PASSIVE ACTIVE	1	[-]
Present camshaft failure synchronization impossible					
LV_ERR_PLAUS_CAM_EX[NC_NR_CAM_CBK]	V/O/S	0H 1H	PASSIVE ACTIVE	1	[-]
Present camshaft failure synchronization impossible					
LV_END_DIAG_PLAUS_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of camshaft signal diagnosis synchronization impossible					
LV_END_DIAG_PLAUS_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of camshaft signal diagnosis synchronization impossible					
CTR_CYC_ENG_PLAUS_CAM_IN[NC_NR_CAM_CBK]	V	0...FFH	0...255	1	[-]
Engine cycle counter for camshaft signal plausibility diagnosis					
CTR_CYC_ENG_PLAUS_CAM_EX[NC_NR_CAM_CBK]	V	0...FFH	0...255	1	[-]
Engine cycle counter for camshaft signal plausibility diagnosis					
LV_CDN_DIAG_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition					
LV_CDN_DIAG_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]

Diagnosis condition					
ERR_SYM_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symtoms of the intake camshaft diagnosis for crankshaft synchronization					
ERR_SYM_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symtoms of the exhaust camshaft diagnosis for crankshaft synchronization					
LV_ERR_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	V/O/S	0H 1H	PASSIVE ACTIVE	1	[-]
Present intake camshaft failure for crankshaft synchronization					
LV_ERR_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	V/O/S	0H 1H	PASSIVE ACTIVE	1	[-]
Present exhaust camshaft failure for crankshaft synchronization					
LV_END_DIAG_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of intake camshaft diagnosis for crankshaft synchronization					
LV_END_DIAG_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	V	0H 1H	PASSIVE ACTIVE	1	[-]
Result available of exhaust camshaft diagnosis for crankshaft synchronization					
CTR_ORNG_CAM_SYN_CRK	V	0...FFH	0...255	1	[-]
Failure counter to detect camshaft error for crankshaft synchronization					

Input data:

CTR_EDGE_CAM_IN[NC_NR_CAM_CBK]	CTR_EDGE_CAM_EX[NC_NR_CAM_CBK]	LV_SYN_VLD	LV_ACT_CRK
LV_CRK_SYN	LV_SYN_CAM_IN[NC_NR_CAM_CBK]	LV_SYN_CAM_EX[NC_NR_CAM_CBK]	LV_IGK
LV_FIRST_VLD_TOOTH	LV_ORNG_CAM_SYN_CRK	LV_ACT_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_ACT_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]
LV_INH_DIAG_CRK_PLAUS	LV_INH_DIAG_PLAUS_CAM_IN[NC_NR_CAM_CBK]	LV_INH_DIAG_PLAUS_CAM_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_ENSD
LV_AD_END_CAM_IN[NC_NR_CAM_CBK]	LV_AD_END_CAM_EX[NC_NR_CAM_CBK]	LV_CRK_MISS_RUN_ENG	LV_INH_DIAG_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]
LV_INH_DIAG_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	LV_INH_DIAG_EXT_CRK_PLAUS	LV_INH_DIAG_EXT_PLAUS_CAM_IN[NC_NR_CAM_CBK]	LV_INH_DIAG_EXT_PLAUS_CAM_EX[NC_NR_CAM_CBK]
LV_INH_DIAG_EXT_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_INH_DIAG_EXT_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]	LV_STOP_ENG	

23.19.1.1 Diagnosis of crankshaft sensor

FUNCTION DESCRIPTION:

General information:

The purpose of the function is to detect a failure if synchronization on crankshaft signal cannot be achieved.

One of the available camshaft signals is selected for crankshaft synchronization by setting one of the flags LV_ACT_SYN_CRK_CAM_IN[i] in the engine position control function. The signal edge counter of that camshaft is observed while the crankshaft is not synchronized. If the count reaches a calibrated number, and the system is still not synchronized with the crankshaft signal, then crankshaft sensor plausibility error is set without further debouncing. If valid crankshaft teeth have already been detected, the symptom will be “implausible signal”, otherwise it will be “missing signal”.

i = 1...NC_NR_CAM_CBK

Description:

Error-symptoms are defined to this diagnostic function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 |  |  |  └─> Signal missing          (= SYM_0)
 |  |  └────> Signal implausible      (= SYM_1)
 |  └───────> (not used)              (= SYM_2)
 └──────────> (not used)              (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_CRK_PLAUS = 0 at reset
LV_END_DIAG_CRK_PLAUS = 0 at reset
ERR_SYM_CRK_PLAUS = NO_SYM at reset
LV_ERR_CRK_PLAUS = 0 at reset

Recurrence:	every update of CTR_EDGE_CAM_IN(EX)[i] And 0 to 1 transition of LV_STOP_ENG
Activation:	LV_ACT_CRK = 1 And LV_INH_DIAG_CRK_PLAUS = 0 And LV_INH_DIAG_EXT_CRK_PLAUS = 0 And LV_INH_DIAG_ENSD = 0

When the activation condition is fulfilled then LV_CDN_DIAG_CRK_PLAUS = 1.

Deactivation:

When the activation condition is not fulfilled then LV_CDN_DIAG_CRK_PLAUS = 0.

Formula Section

Symptoms calculation:

```
If      LV_STOP_ENG = 1
        And
        LV_CRK_MISS_RUN_ENG = 1
Then    ERR_SYM_CRK_PLAUS = "SYM_0"
        {Crankshaft signal missing but engine is still turning}
```

```
else
        If      LV_CRK_SYN = 0
        and     CTR_EDGE_CAM_IN(EX)[i] >
                C_NR_EDGE_CAM_IN(EX)_SYN_CRK_MAX
        Then    If      LV_FIRST_VLD_TOOTH = 1
                Then    ERR_SYM_CRK_PLAUS = “SYM_1"
                        {Detection of symptom SIGNAL IMPLAUSIBLE}
                Else    ERR_SYM_CRK_PLAUS = “SYM_0"
                        {Detection of symptom SIGNAL MISSING}
                Endif
                LV_ERR_CRK_PLAUS = 1
                LV_END_DIAG_CRK_PLAUS = 1
        Endif
Endif
```

LV_ERR_CRK_PLAUS and ERR_SYM_CRK_PLAUS are reset to initialization value at 0 to 1 transition of LV_IGK.

```
If      LV_CRK_SYN = 1
Then    LV_END_DIAG_CRK_PLAUS = 1
        LV_ERR_CRK_PLAUS = 0
        ERR_SYM_CRK_PLAUS = NO_SYM
        Deliver the result to Error Management
Endif
```

Configuration for diagnostic symptoms :

Diagnostic Crankshaft sensor	**Symptom description**	**Symptom**	**Filter type**
Crankshaft sensor	*SIGNAL MISSING*	*SYM_0*	*NO*
	SIGNAL IMPLAUSIBLE	*SYM_1*	
		SYM_2	
		SYM_3	

NB: The filter type for this diagnosis can not be configured, it behaves as a MEM filter.

23.19.1.2 Camshaft sensor plausibility diagnosis

FUNCTION DESCRIPTION:

General information:

The functional camshaft diagnosis cannot detect a failure when a camshaft sensor delivers no signal at all. Missing signal will be detected if the camshaft edge counter is not incremented at least once per engine cycle.

i = 1...NC_NR_CAM_CBK

Description:

Error-symptoms are defined to this diagnostic function as following:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 │  │  │  └─> Signal missing        (= SYM_0)
 │  │  └────> (not used)            (= SYM_1)
 │  └───────> (not used)            (= SYM_2)
 └──────────> (not used)            (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_PLAUS_CAM_IN(EX)[i] = 0 at reset
LV_END_DIAG_PLAUS_CAM_IN(EX)[i] = 0 at reset
ERR_SYM_PLAUS_CAM_IN(EX)[i] = NO_SYM at reset
LV_ERR_PLAUS_CAM_IN(EX)[i] = 0 at reset

CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i] = C_NR_CYC_ENG_PLAUS_CAM_MAX at reset

Recurrence: once per engine cycle (two crankshaft revolutions), at any phasing i.e. on crankshaft revolution which was on-going at engine synchronization

Activation:
LV_CRK_SYN = 1
And LV_INH_DIAG_PLAUS_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_EXT_PLAUS_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_ENSD = 0
And C_NR_CYC_ENG_PLAUS_CAM_MAX ≠ 0

When the activation condition is fulfilled then LV_CDN_DIAG_PLAUS_CAM_IN(EX)[i] =1.

Deactivation:

When the activation condition is not fulfilled then LV_CDN_DIAG_PLAUS_CAM_IN(EX)[i] = 0.

Formula Section

Symptoms calculation:

If $\mathrm{CTR\_EDGE\_CAM\_IN(EX)[i]}_n = \mathrm{CTR\_EDGE\_CAM\_IN(EX)[i]}_{n-1}$

```
Then    ERR_SYM_PLAUS_CAM_IN(EX)[i] = "SYM_0" {SIGNAL MISSING}
        increment CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i]
Else    ERR_SYM_PLAUS_CAM_IN(EX)[i] = NO_SYM
        decrement CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i]
Endif
```

CTR_EDGE_CAM_IN(EX)[i]$_{n-1}$ *will be initialized to zero at engine stop detection.*

```
If      CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i] ≥
        2*C_NR_CYC_ENG_PLAUS_CAM_MAX
Then    LV_ERR_PLAUS_CAM_IN(EX)[i] = 1
        LV_END_DIAG_PLAUS_CAM_IN(EX)[i] = 1
        reset CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i] (to
        C_NR_CYC_ENG_PLAUS_CAM_MAX)
        Deliver the result to Error Management
Else    If      CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i] =0
        Then    If      LC_CTR_DEC_PLAUS_CAM = 1
                Then    LV_ERR_PLAUS_CAM_IN(EX)[i] = 0
                Endif
                LV_END_DIAG_PLAUS_CAM_IN(EX)[i] = 1
                reset CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i] (to
                C_NR_CYC_ENG_PLAUS_CAM_MAX)
                Deliver the result to Error Management
        Endif
Endif
```

LV_ERR_PLAUS_CAM_IN(EX)[i] , ERR_SYM_PLAUS_CAM_IN(EX)[i] and CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i] are reset to initialization value at 0 to 1 transition of LV_IGK.

```
If      LV_SYN_CAM_IN(EX)[i] = 1
Then    If      LC_CTR_DEC_PLAUS_CAM = 1
        Then    LV_ERR_PLAUS_CAM_IN(EX)[i] = 0
        Endif
        LV_END_DIAG_PLAUS_CAM_IN(EX)[i] = 1
        ERR_SYM_PLAUS_CAM_IN(EX)[i] = NO_SYM
        reset CTR_CYC_ENG_PLAUS_CAM_IN(EX)[i] (to
        C_NR_CYC_ENG_PLAUS_CAM_MAX)
        Deliver the result to Error Management
```

Endif

Configuration for diagnostic symptoms :

Diagnostic Camshaft sensor plausability	Symptom description	Symptom	Filter type
Camshaft sensor plausability diagnosis for intake	*SIGNAL MISSING*	*SYM_0*	*NO*
		SYM_1	
		SYM_2	
		SYM_3	
Camshaft sensor plausability diagnosis for exhaust	*SIGNAL MISSING*	*SYM_0*	*NO*
		SYM_1	
		SYM_2	
		SYM_3	

NB: The filter type for this diagnosis can not be configured, but it behaves as a MEM if LC_CTR_DEC_PLAUS_CAM = 0, and as a STD_INI if LC_CTR_DEC_PLAUS_CAM = 1

23.19.1.3 Camshaft Sensor Diagnosis for Crankshaft Synchronization

FUNCTION DESCRIPTION:

General information:

The purpose of the function is to detect a failure of the camshaft information for crankshaft synchronization.

Up to four camshaft signals may be available in the system (intake and/or exhaust on one or two cylinder banks). The Engine Position Control selects one camshaft for synchronization by activating the corresponding operating mode. The camshaft signal acquisition will indicate “camshaft signal out of range for crankshaft synchronization” by setting LV_ORNG_CAM_SYN_CRK, which has to be reset by the diagnosis after treatment. A counter is incremented at every recurrence if LV_ORNG_CAM_SYN_CRK is set. The error is detected when the counter reaches a calibrated maximum.

i = 1...NC_NR_CAM_CBK

Description:

Error-symptoms are defined to this diagnostic function as following:

```
|_|_|_|_|
 | | | └-> Camshaft signal  not valid for crankshaft synchronization    (= SYM_0)
 | | └───> (not used)                                                  (= SYM_1)
 | └─────> (not used)                                                  (= SYM_2)
 └───────> (not used)                                                  (= SYM_3)
```

Application conditions:

Initialisation:

LV_CDN_DIAG_SYN_CRK_CAM_IN(EX)[i] = 0 at reset
LV_END_DIAG_SYN_CRK_CAM_IN(EX)[i] = 0 at reset
ERR_SYM_SYN_CRK_CAM_IN(EX)[i] = NO_SYM at reset

LV_ERR_SYN_CRK_CAM_IN(EX)[i] = 0 at reset

Recurrence: every crankshaft reference gap and every camshaft edge

Activation:
LV_ACT_CRK = 1
And LV_INH_DIAG_SYN_CRK_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_EXT_SYN_CRK_CAM_IN(EX)[i] = 0
And LV_INH_DIAG_ENSD = 0

When the activation condition is fulfilled then LV_CDN_DIAG_SYN_CRK_CAM_IN(EX)[i] = 1

Deactivation:

When the activation condition is not fulfilled then LV_CDN_DIAG_SYN_CRK_CAM_IN(EX)[i] = 0.

Formula Section

Symptoms calculation:

For LV_ACT_SYN_CRK_CAM_IN(EX)[i] = 1 (camshaft sensor currently selected for synchronization):

```
if      LV_ORNG_CAM_SYN_CRK = 0
then    ERR_SYM_SYN_CRK_CAM_IN(EX)[i] = NO_SYM
else    ERR_SYM_SYN_CRK_CAM_IN(EX)[i] = “SYM_0" {SIGNAL INVALID}
        increment CTR_ORNG_CAM_SYN_CRK
endif

if      CTR_ORNG_CAM_SYN_CRK > C_CTR_ORNG_CAM_SYN_CRK_MAX
then    LV_ERR_SYN_CRK_CAM_IN(EX)[i] = 1
        LV_END_DIAG_SYN_CRK_CAM_IN(EX)[i] = 1
        Deliver the result to Error Management
        reset CTR_ORNG_CAM_SYN_CRK
endif

if      LV_SYN_VLD = 1
then    LV_END_DIAG_SYN_CRK_CAM_IN(EX)[i] = 1
        ERR_SYM_SYN_CRK_CAM_IN(EX)[i] = NO_SYM
        LV_ERR_SYN_CRK_CAM_IN(EX)[i] = 0
        Deliver the result to Error Management
        reset CTR_ORNG_CAM_SYN_CRK
endif
```

For LV_ACT_SYN_CRK_CAM_IN(EX)[i] = 0 (camshaft sensor currently **not** selected for cam/crk synchronization):

```
if      LV_SYN_CAM_IN(EX)[i] = 1
        And
        LV_AD_END_CAM_IN(EX)[i] = 1
then    ERR_SYM_SYN_CRK_CAM_IN(EX)[i] = NO_SYM
        If          LC_CAM_ERR_ENA_VVT = 1
        Then    LV_ERR_SYN_CRK_CAM_IN(EX)[i] = 0
                LV_END_DIAG_SYN_CRK_CAM_IN(EX)[i] = 1
                Deliver the result to Error Management
        Endif
endif
```

LV_ERR_SYN_CRK_CAM_IN(EX)[i], ERR_SYM_SYN_CRK_CAM_IN(EX)[i] and CTR_ORNG_CAM_SYN_CRK are reset to initialization value at 0 to 1 transition of LV_IGK.

Configuration for diagnostic symptoms :

Diagnostic **Camshaft sensor diagnosis for crankshaft synchronization**	**Symptom description**	**Symptom**	**Filter type**
Camshaft sensor diagnosis for crankshaft synchronization on intake	*SIGNAL INVALID*	*SYM_0*	*NO*
		SYM_1	
		SYM_2	
		SYM_3	
Camshaft sensor diagnosis for crankshaft synchronization on exhaust	*SIGNAL INVALID*	*SYM_0*	*NO*
		SYM_1	
		SYM_2	
		SYM_3	

NB: The filter type for this diagnosis can not be configured, but it behaves as a MEM if LC_CAM_ERR_ENA_VVT = 0, and as a STD_INI if LC_CAM_ERR_ENA_VVT = 1

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_NR_EDGE_CAM_IN_SYN_CRK_MAX	1	1...FFH	1...255	1	[-]
Number of intake camshaft signal edges to detect crankshaft plausibility error before synchronization					
C_NR_EDGE_CAM_EX_SYN_CRK_MAX	1	1...FFH	1...255	1	[-]
Number of exhaust camshaft signal edges to detect crankshaft plausibility error before synchronization					
C_NR_CYC_ENG_PLAUS_CAM_MAX	1	0...7FH	0...127	1	[-]
Number of engine cycles to detect camshaft plausibility error					
C_CTR_ORNG_CAM_SYN_CRK_MAX	1	0...FFH	0...255	1	[-]
Failure count to detect camshaft error for crankshaft synchronization					
LC_CAM_ERR_ENA_VVT	1	0...1H	0...1	1	[-]
Switch to enable camshaft error debouncing for VVT reactivation					
LC_CTR_DEC_PLAUS_CAM	1	0...1H	0...1	1	[-]
Switch to enable camshaft plausibility debouncing					

23.19.2 Crankshaft Sensor Circuit Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CDN_DIAG_CRK_OC	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis condition					
ERR_SYM_CRK_OC	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom of the crankshaft sensor failure open or short circuit					
LV_ERR_CRK_OC	V/O	0H 1H	PASSIVE ACTIVE	1	[-]
Crankshaft sensor failure open or short circuit present					
LV_END_DIAG_CRK_OC	V	0H 1H	PASSIVE ACTIVE	1	[-]
Diagnosis result available					

Input data:

LV_IGK	LV_INH_DIAG_CRK_OC	LV_INH_DIAG_EXT_CRK_OC	N_32

FUNCTION DESCRIPTION:

General information:

This dummy file replaces the MCPS circuit diagnosis for systems with ACPS. A circuit diagnosis for ACPS is not considered neceessary, since with any of the three lines open or short-circuited, the signal will be constantly high or low. The failure will then be detected by the functional diagnosis.

Application conditions:

Activation: LV_IGK = 1
And LV_INH_DIAG_CRK_OC = 0
And LV_INH_DIAG_EXT_CRK_OC = 0

When the activation condition is fulfilled then LV_CDN_DIAG_CRK_OC = 1.

Desactivation:

When the activation condition is not fulfilled then LV_CDN_DIAG_CRK_OC = 0.

Formula section:

ERR_SYM_CRK_OC = 0

LV_ERR_CRK_OC = 0

LV_END_DIAG_CRK_OC = 1

23.19.3 Crankshaft Sensor Diagnosis (Appl. Inc.)

23.19.3.1 Crankshaft Sensor Diagnosis with Tooth Number Error

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_EXT_CRK_TOOTH	V/O	0...1H	0...1	1	[-]
Inhibition condition for the crankshaft sensor diagnosis with tooth number error					

Input data:

LV_ERR_CRK_TOOTH			

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_EXT_CRK_TOOTH allows to deactivate the corresponding diagnostic.

Application conditions:

Initialisation:
At reset event : LV_INH_DIAG_EXT_CRK_TOOTH= 0

Recurrence:
Application incidence recurrence = once at reset

Activation:
-/-

Deactivation:
When the activation condition is not fulfilled

Formula section:

LV_INH_DIAG_EXT_CRK_TOOTH = 0

Configuration for diagnostic symptoms :

Diagnostic Crankshaft sensor	Symptom description	Symptom	Filter type
Crankshaft sensor diagnosis with tooth number error	*TOOTH_NUMBER*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	

Filter types compatibility:

Filter Type	Compatibility	Remark
STD	NO	
STD_INI	YES	
MEM	YES	Prefered solution
MEM_INI	NO	
DEC_CAL	NO	
STC	YES	
NO_FIL	NO	

Diagnosis / Failure Flag	Symptom		DTC
Crankshaft sensor tooth number diagnosis LV_ERR_CRK_TOOTH	Bit 0	Additional or missing teeth per crankshaft revolution	P0370
	Bit 1	unused	
	Bit 2	unused	
	Bit 3	unused	

23.19.3.2 Crankshaft Sensor Diagnosis with implausible Tooth Period

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_EXT_CRK_TOOTH_PER	V/O	0...1H	0...1	1	[-]
Inhibition condition for the crankshaft sensor diagnosis with implausible tooth period					

Input data:

LV_ERR_CRK_TOOTH_PER			

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_EXT_CRK_TOOTH_PER allows to deactivate the corresponding diagnostic.

Application conditions:

Initialisation:
At reset event : LV_INH_DIAG_EXT_CRK_TOOTH_PER= 0

Recurrence:
Application incidence recurrence = once at reset

Activation:
-/-

Deactivation:
When the activation condition is not fulfilled

Formula section:

LV_INH_DIAG_EXT_CRK_TOOTH_PER = 0

Configuration for diagnostic symptoms :

Diagnostic Crankshaft sensor	Symptom description	Symptom	Filter type
Crankshaft sensor diagnosis with implausible tooth period error	*TOOTH_PERIOD*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	

Filter types compatibility:

Filter Type	Compatibility	Remark
STD	NO	
STD_INI	YES	
MEM	YES	Prefered solution
MEM_INI	NO	
DEC_CAL	NO	
STC	YES	
NO_FIL	NO	

Diagnosis / Failure Flag	Symptom		DTC
Crankshaft sensor tooth period diagnosis	Bit 0	Crankshaft tooth period implausible	P0370
	Bit 1	unused	
	Bit 2	unused	
LV_ERR_CRK_TOOTH_PER	Bit 3	unused	

23.19.3.3 Crankshaft Sensor Diagnosis with Loss of Synchronization

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_EXT_CRK_SYN	V/O	0...1H	0...1	1	[-]
Inhibition condition for the crankshaft sensor diagnosis with loss of synchronization					

Input data:

LV_ERR_CRK_SYN			

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_EXT_CRK_SYN allows to deactivate the corresponding diagnostic.

Application conditions:

Initialisation:
At reset event : LV_INH_DIAG_EXT_CRK_SYN = 0

Recurrence:
Application incidence recurrence = once at reset

Activation:
-/-

Deactivation:
When the activation condition is not fulfilled

Formula section:

LV_INH_DIAG_EXT_CRK_SYN = 0

Configuration for diagnostic symptoms :

Diagnostic **Crankshaft sensor**	**Symptom description**	**Symptom**	**Filter type**
Crankshaft sensor diagnosis with loss of synchronization	*SYNC_LOSS*	*SYM_0*	*MEM*
		SYM_1	
		SYM_2	
		SYM_3	

Filter types compatibility:

Filter Type	Compatibility	Remark
STD	NO	
STD_INI	NO	
MEM	YES	Prefered solution
MEM_INI	NO	
DEC_CAL	NO	
STC	NO	
NO_FIL	NO	

Diagnosis / Failure Flag	**Symptom**		**DTC**
Crankshaft sensor synchronization diagnosis	Bit 0	Loss of crankshaft synchronization	P0370
	Bit 1	unused	
	Bit 2	unused	
LV_ERR_CRK_SYN	Bit 3	unused	

23.19.4 Crankshaft Sensor Circuit Diagnosis (Appl. Inc.)

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_EXT_CRK_OC	V/O	0...1H	0...1	1	[-]
Inhibition condition for the crankshaft sensor circuit diagnosis					

Input data:

LV_ERR_CRK_OC			

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_EXT_CRK_OC allows to deactivate the corresponding diagnostic.

Application conditions:

Initialisation:
At reset event : LV_INH_DIAG_EXT_CRK_OC = 0

Recurrence:
Application incidence recurrence = once at reset

Activation:
-/-

Deactivation:
When the activation condition is not fulfilled

Formula section:

LV_INH_DIAG_EXT_CRK_OC = 0

Configuration for diagnostic symptoms :

Diagnostic Crankshaft sensor circuit	Symptom description	Symptom	Filter type
Magnetic crankshaft position sensor	*OPEN / SHORT CIRCUIT*	*SYM_0*	*STD_INI*
		SYM_1	
		SYM_2	
		SYM_3	

Filter types compatibility:

Filter Type	Compatibility	Remark
STD	NO	
STD_INI	YES	Prefered solution

MEM	YES	
MEM_INI	NO	
DEC_CAL	NO	
STC	YES	
NO_FIL	NO	

Diagnosis / Failure Flag	Symptom		DTC
Crankshaft sensor Circuit diagnosis LV_ERR_CRK_OC	Bit 0	open/short circuit	P0335
	Bit 1	unused	
	Bit 2	unused	
	Bit 3	unused	

23.19.5 Crankshaft Sensor Plausibility Diagnosis (Appl. Inc.)

23.19.5.1 Diagnosis of crankshaft sensor

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_EXT_CRK_PLAUS	V/O	0...1H	0...1	1	[-]
Inhibition condition for the diagnosis of crankshaft sensor					

Input data:

LV_ERR_CRK_PLAUS			

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_EXT_CRK_PLAUS allows to deactivate the corresponding diagnostic.

Application conditions:

Initialisation:
At reset event : LV_INH_DIAG_EXT_CRK_PLAUS = 0

Recurrence:
Application incidence recurrence = once at reset

Activation:
-/-

Deactivation:
When the activation condition is not fulfilled

Formula section:

LV_INH_DIAG_EXT_CRK_PLAUS = 0

Diagnosis / Failure Flag	Symptom		DTC
Crankshaft sensor plausibility diagnosis LV_ERR_CRK_PLAUS	Bit 0	Signal Missing	P0374
	Bit 1	Signal Implausible	P0373
	Bit 2	unused	
	Bit 3	unused	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_DIAG_CRK_OC	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for crankshaft sensor circuit diagnosis					
C_ABC_MAX_DIAG_CRK_OC	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for crankshaft sensor circuit diagnosis					
C_ABC_INC_DIAG_CRK_TOOTH	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for crankshaft tooth number diagnosis					
C_ABC_INC_DIAG_CRK_TOOTH_PER	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for crankshaft tooth period diagnosis					
C_ABC_INC_DIAG_CRK_SYN	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment for crankshaft diagnosis with loss of synchronization					
C_ABC_MAX_DIAG_CRK_TOOTH	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for crankshaft tooth number diagnosis					
C_ABC_MAX_DIAG_CRK_TOOTH_PER	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for crankshaft tooth period diagnosis					
C_ABC_MAX_DIAG_CRK_SYN	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum for crankshaft diagnosis with loss of synchronization					

Calibration interfering functions:

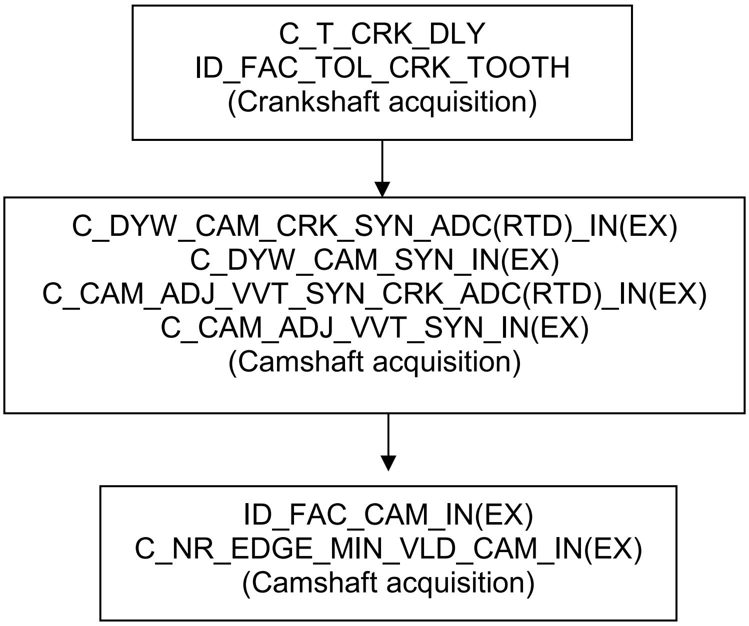
INJR, VVTI, IGSP and IGRE must be pre-calibrated

Calibration of C_DYW_CAM_SYN_IN(EX) and C_CAM_ADJ_VVT_SYN_IN(EX) influences pre-injection function (INJR-aggregate).

Calibration of C_NR_EDGE_MIN_VLD_CAM_IN(EX) influences VVT-function (VVTI-aggregate)

Note that engine will not start if one of the following calibration data is wrong (too low or too high).

Calibration flowchart:



Calibration process:

Calibration data ID_FAC_TOL_CRK_TOOTH

Calibration data ID_FAC_TOL_CRK_TOOTH defines the width of the crankshaft acceptance window for:

1) tooth validation
2) reference gap detection

ID_FAC_TOL_CRK_TOOTH [0rpm] is used for detection of the first reference gap and tooth validation before crankshaft synchronization.

Calibration of ID_FAC_TOL_CRK_TOOTH depends on engine accelerations / decelerations and crankshaft sensor/target wheel accuracy.

- ID_FAC_TOL_CRK_TOOTH [0rpm] should be calibrated taking into account engine accelerations before crankshaft synchronization (engine turning with starter):

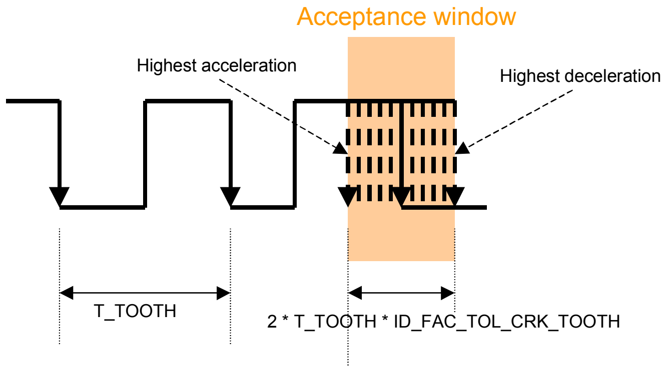
If NC_NR_TOOTH_FIRST_GAP_MIN = 2 (two teeth need to be simulated during first reference GAP):

- Low value should be preferred (between 0.2 and 0.3) in order to synchronize as soon as first reference gap occurs
- If calibration value is too high, engine synchronization will be late (after occurrence of first gap) or engine will not start at all.
- If value is too low (< 0.2), crankshaft tooth period error (LV_ERR_CRK_TOOTH_PER) could occur.

If strong accelerations are observed during reference GAP, it implies NC_NR_TOOTH_FIRST_GAP_MIN = 1 (only one tooth need to be simulated during first reference GAP) and then:

- medium value should be preferred (between 0.4 and 0.6) in order to distinguish correctly the GAP from the normal teeth and thus synchronize as soon as first reference gap occurs
- If calibration value is too high, engine synchronization will be late (after occurrence of first gap) or engine will not start at all.
- If value is too low, a normal tooth with strong deceleration could be considered as a GAP.

- ID_FAC_TOL_CRK_TOOTH [> 0rpm] should be calibrated taking into account engine accelerations/decelerations after engine synchronization (especially first combustion after synchronization):
  - Low value should be preferred (between 0.25 and 0.4) in order to detect invalid teeth and to avoid loss of synchronization after first combustion (especially if high acceleration occurs during reference gap, e.g. gap just after TDC).
  - Large window is not safer as margin of robustness for gap detection decreases.
- ID_FAC_TOL_CRK_TOOTH is also used for crankshaft tooth validation:



- If a active crankshaft edge occurs before the acceptance window defined by ID_FAC_TOL_CRK_TOOTH, the signal edge is invalid and the tooth counter is not incremented.
- If no active crankshaft edge occurs inside the acceptance window defined by ID_FAC_TOL_CRK_TOOTH, a crankshaft tooth is simulated and the tooth counter is incremented.

In both cases the, the tooth counter and, consequently, the engine position is corrected immediately. Otherwise, when the acceptance window is too large (high value of

ID_FAC_TOL_CRK_TOOTH), the engine position is corrected only at reference gap (no immediate correction possible) and timing events (injection/ignition) occur too late or too early. Therefore, for the speed range where engine acceleration is small, low values of ID_FAC_TOL_CRK_TOOTH should be prefered

- If less than NC_NR_GAP teeth are simulated during the reference gap (in case of high engine acceleration / speed variations), the crankshaft tooth counter is corrected and error symptom “ERR_SYM_CRK_TOOTH” is set. In order to avoid false crank error detection, ID_FAC_TOL_CRK_TOOTH should be small.

- Crankshaft acceptance window is deactivated for the defined speed range when ID_FAC_TOL_CRK_TOOTH is set to zero. Consequently, reference gap detection is also deactivated.

- After synchronization acceptance window could be disabled below NC_N_CRK_WIN_ENA, for example for engines having strong accelerations/decelerations at start: In this case the acceptance window is only applied for GAP detection and some teeth around the GAP below the configured engine speed threshold and should be adapted to accelerations/decelerations occuring on teeth where the GAP is searched.

Calibration data C_T_CRK_DLY

After the first active crankshaft edge, the crankshaft acquisition is deactivated during a sensing delay. The duration of the sensing delay is defined by the the calibration data C_T_CRK_DLY.

The purpose is to avoid acquisition of signal spikes:

- at very low engine speed (< 30rpm): In this case, the output signal level of a magnetic crankshaft position sensor (MCPS) is too low in order to produce plausible signal edges at the µC input pin.

- during start of the starter motor: a high current is drawn from the battery until the starter motor reaches its nominal speed. The current and voltage gradient can produce electromagnetic disturbances on the sensor lines and the ECU.

Calibration data C_DYW_CAM_CRK_SYN_ADC(RTD)_IN(EX)

Engine synchronization in ENSD is based on camshaft level detection at engine start and evaluation of the angular distance between crankshaft and camshaft signal events.

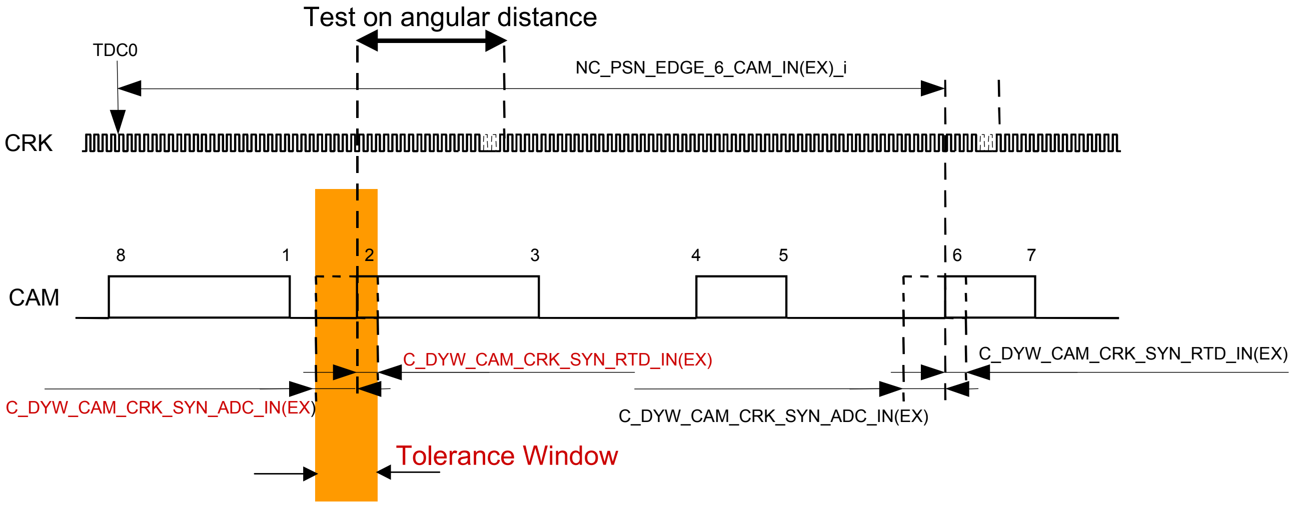
Possible camshaft to crankshaft signal events are:

1. start position to reference gap
2. camshaft signal edge to reference gap (see figure below)
3. reference gap to camshaft signal edge

If the measured angular distance between one of the above events is inside the designed distance including a tolerance window, engine synchronizes.

The advance range of the tolerance window is defined by calibration data C_DYW_CAM_CRK_SYN_ADC_IN(EX).

The retard range of the tolerance window is defined by calibration data C_DYW_CAM_CRK_SYN_RTD_IN(EX).



C_DYW_CAM_CRK_SYN_ADC(RTD)_IN(EX) should be calibrated taking into account engine-to-engine variations of camshaft to crankshaft edge positions (including angular shift of camshaft to crankshaft):

- The maximum spread from designed camshaft edge position to advanced position should be included in C_DYW_CAM_CRK_SYN_ADC_IN(EX)

- The maximum spread from designed camshaft edge position to retarded position should be included in C_DYW_CAM_CRK_SYN_RTD_IN(EX)

Please note that TDC0 is the reference position for all camshaft and crankshaft positions.

All mechanical and electrical tolerances of the sensors, the target wheels and the mounting should be included:

- mounting tolerances of camshaft and crankshaft sensor
- mounting tolerances of camshaft and crankshaft target wheel
- camshaft and crankshaft sensor accuracy
- machining of camshaft and crankshaft target wheel
- angular shift of crankshaft to crankshaft target wheel position

Sum up all these tolerance values for the intake and exhaust camshaft and set calibration values C_DYW_CAM_CRK_SYN_ADC(RTD)_IN(EX) accordingly.

Consider also that mechanical tolerances and angular shift of crank to cam can vary during engine lifetime (for example: increased belt/chain length). This variation occurs generally in retard direction.

If the camshaft sensor features a pre-calibration mode in order to adapt the air-gap between sensor and target wheel, sensor accuracy for pre-calibration mode shall be considered (low accuracy at cranking, auto-adaptation generally terminated several engine revolutions after engine synchronization).

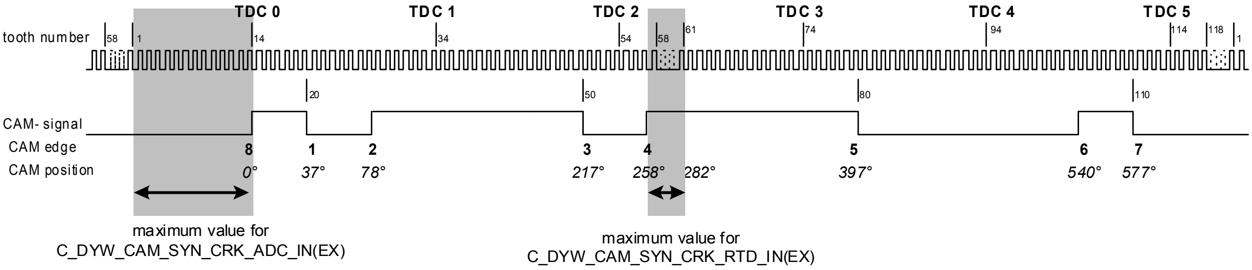
Comments:

1) The **maximum value for the tolerance window** is:

C_DYW_CAM_CRK_SYN_RTD_IN(EX) = minimal angular distance over one engine cycle between designed camshaft edge position and reference gap for all camshaft edges **before** reference gap

C_DYW_CAM_CRK_SYN_ADC_IN(EX) = minimal angular distance over one engine cycle between designed camshaft edge position and reference gap for all camshaft edges **after** reference gap

Example multi-teeth target wheel:



If a value higher than this maximum value is calibrated, engine position cannot be unambiguously identified at reference gap and the cam/crk synchronization out-of-range flag LV_ORNG_CAM_SYN_CRK is set.

Remember that the position of the reference gap is defined as the position in °CRK of the first active crankshaft edge after the gap.

2) Calibration data C_DYW_CAM_CRK_SYN_ADC(RTD)_IN(EX) applies for **VVT actuator locked at engine start** (normal condition).

Calibration data C_CAM_ADJ_VVT_SYN_CRK_ADC(RTD)_IN(EX)

If the VVT actuator is not locked at engine start (flag LV_CAM_LOCK_IVVT_IN(EX)_i set to 0), the cam/crk synchronization tolerance windows defined by C_DYW_CAM_CRK_SYN_ADC(RTD)_IN(EX) will be extended by calibration data C_CAM_ADJ_VVT_SYN_CRK_ADC(RTD)_IN(EX):

VVT on intake camshaft (adjustement in advance direction):

-> VVT actuator range in C_CAM_ADJ_VVT_SYN_CRK_ADC_IN

VVT on exhaust camshaft (adjustement in retard direction):

-> VVT actuator range included in C_CAM_ADJ_VVT_SYN_CRK_RTD_EX

C_CAM_ADJ_VVT_SYN_CRK_ADC(RTD)_IN(EX) should be lower than or equal to VVT actuator range. Actual value depends on time for VVT actuator to return to passive position and cam/crk configuration. Cam/crk synchronization should not be out-of-range (CTR_ORNG_CAM_SYN_CRK should remain zero).

Calibration data C_DYW_CAM_SYN_IN(EX)

Calibration data C_DYW_CAM_SYN_IN(EX) is used to test the angular distance between two camshaft edges (crankshaft events are not considered).

C_DYW_CAM_SYN_IN(EX) does not depend on absolute crankshaft or camshaft edge positions but is based exclusively on relative angular distances. The angular distance between camshaft edges is measured via the number of crankshaft signal edges which occurs between two camshaft events.

If the measured angular distance between two successive camshaft edges is inside the designed distance including a tolerance window, then

- the list of possible camshaft signal indexes is updated, and, consequently, the minimum and maximun engine position converges gradually (before engine synchronization: for pre-injection)
- the camshaft signal edge is valid for synchronization

C_DYW_CAM_SYN_IN(EX) should be calibrated taking into account engine-to-engine variations of camshaft edge positions:

- camshaft and crankshaft sensor accuracy
- machining of camshaft and crankshaft target wheel

Sum up all these tolerance values for the intake and exhaust camshaft and set calibration values C_DYW_CAM_SYN_IN(EX) accordingly.

If the camshaft sensor features a pre-calibration mode in order to adapt the air-gap between sensor and target wheel, sensor accuracy for pre-calibration mode shall be considered (as for C_DYW_CAM_SYN_CRK_ADC(RTD)_IN(EX)).

Comments:

1) C_DYW_CAM_SYN_IN(EX) must be lower than half the minimum angular distance between designed camshaft edge positions:

   C_DYW_CAM_SYN_IN(EX) < 0.5 * min(NC_PSN_EDGE_z_CAM_IN(EX)_i - NC_PSN_EDGE_z-1_CAM_IN(EX)_i)

2) Calibration data C_DYW_CAM_SYN_IN(EX) applies for **VVT actuator locked at engine start** (normal condition).

Calibration data C_CAM_ADJ_VVT_SYN_ IN(EX)

If the VVT actuator is not locked at engine start (flag LV_CAM_LOCK_IVVT_IN(EX)_i set to 0), the cam/crk tolerance windows defined by C_DYW_CAM_SYN_IN(EX) will be extended by calibration data C_CAM_ADJ_VVT_SYN_IN(EX):

VVT on intake camshaft (adjustement in advance direction):

-> VVT actuator range in C_CAM_ADJ_VVT_SYN_IN

VVT on exhaust camshaft (adjustement in retard direction):

-> VVT actuator range included in C_CAM_ADJ_VVT_SYN_EX

C_CAM_ADJ_VVT_SYN_IN(EX) should be lower than VVT actuator range. Actual value depends on time for VVT actuator to return to passive position and cam/crk configuration. Engine start could be longer (impact on pre-injection), but cam/crk synchronization should not be out-of-range (CTR_ORNG_CAM_SYN_CRK should remain zero).

Calibration data ID_FAC_CAM_IN(EX)

Calibration data ID_FAC_CAM_IN(EX) defines the tolerance factor for camshaft edge ratio calculation used for:

1) camshaft self-synchronization

2) camshaft sensor diagnosis ratio check

The camshaft edge ratio is defined as:

$$\frac{T\_SEG\_CAM\_IN(EX)\_i_n + T\_SEG\_CAM\_IN(EX)\_i_{n-3}}{T\_SEG\_CAM\_IN(EX)\_i_{n-1} + T\_SEG\_CAM\_IN(EX)\_i_{n-2}}$$

This measured ratio is compared to the designed (theoretical) ratio multiplied/divided by tolerance factor ID_FAC_CAM_IN(EX).

Calibration of ID_FAC_CAM_IN(EX) depends on engine accelerations/decelerations and engine-to-engine variations of camshaft edge positions.

If calibration data ID_FAC_CAM_IN(EX) is too low, camshaft out-of-range ratio flag (LV_ORNG_RATIO_CAM_IN(EX)_i) occurs and camshaft self-synchronization is reset.

The maximum value depends on the camshaft target wheel configuration: The upper limit is defined by theoretical ratio for falling and rising edges. If calibration data is higher than upper limit, the camshaft edge position cannot be identified and the camshaft will not self-synchronize.

If ID_FAC_CAM_IN(EX) is too low, camshaft self-synchronization is lost during high engine accelerations/decelerations resulting in de-activation of VVT.

Calibration data C_NR_EDGE_MIN_VLD_CAM_IN(EX)

Calibration data C_NR_EDGE_MIN_VLD_CAM_IN(EX) defines the number of camshaft edges to start camshaft reference position adaptation and to generate a camshaft position output for VVT control.

Generally, camshaft edges are less precise at engine start. Some camshaft sensors feature a pre-calibration mode in order to adapt the airgap between sensor and target wheel. In this case, camshaft edge positions are shifting during engine start until the nominal position (best accuracy) is reached. Camshaft reference position adaptation can only be started when camshaft edge position don’t shift anymore (otherwise wrong adaptation values are stored).

Camshaft sensor pre-calibration mode ends after a number of defined camshaft edges are detected. Calibration value C_NR_EDGE_MIN_VLD_CAM_IN(EX) should be greater or equal to this value specified in the sensor specification / data sheet.

If VVT actuator is not in passive position at engine start, calibration data C_NR_EDGE_MIN_VLD_CAM_IN(EX) should be greater than the number of camshaft edges necessary to bring the VVT actuator back to passive locked position (oscillations must be finished).

Calibration data LC_CAM_SWI_CONF

It is possible within ENSD to use the same SW for 2 different camshaft configurations (with same number of camshaft edges). When this feature is used, NC_NR_CONF_CAM shall be set to 2 and LC_CAM_SWI_CONF indicates which camshaft configuration will be used.

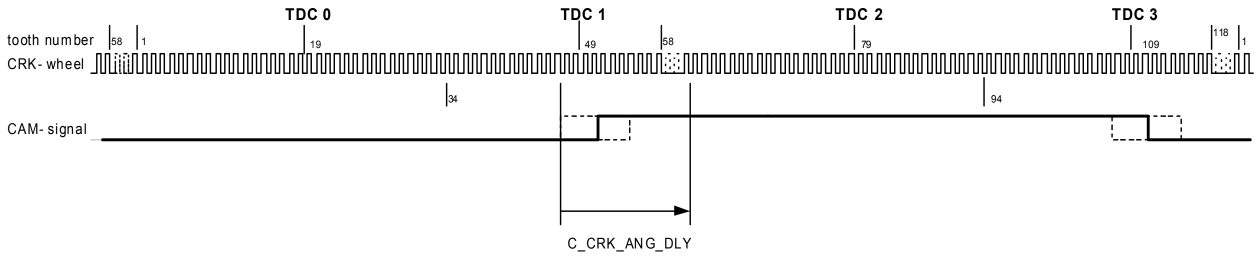
Calibration data C_CRK_ANG_DLY and C_CRK_ANG_DLY_LST_CAM

The maximum and minimum engine position are updated as soon as a crankshaft angle C_CRK_ANG_DLY has occured since first valid tooth and no camshaft signal edge was detected or as soon as a crankshaft angle C_CRK_ANG_DLY_LST_CAM has occured since last CAM and no camshaft signal edge was detected.

It has to be calibrated according to CAM/CRK configuration and timing.

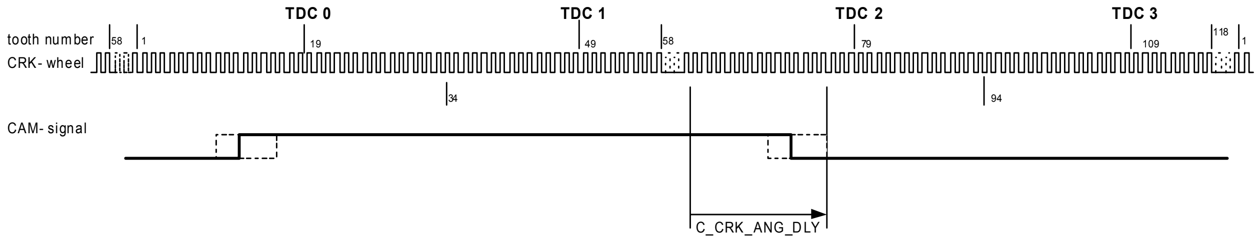
Two cases must be distinguished:

1) Camshaft edge before reference gap:



C_CRK_ANG_DLY = minimum angular distance between designed camshaft edge and reference gap + C_DYW_CAM_CRK_SYN_ADC_IN(EX)

2) Camshaft edge after reference gap:



C_CRK_ANG_DLY = minimum angular distance between designed camshaft edge and reference gap + C_DYW_CAM_CRK_SYN_RTD_IN(EX)

C_CRK_ANG_DLY_LST_CAM = min(angular distances CAM-CAM including tolerance value C_DYW_CAM_SYN_IN(EX), angular distances CAM-GAP including tolerance value C_DYW_CAM_CRK_SYN_ADC_IN(EX))

C_CRK_ANG_DLY = min(angular distances CAM-CAM including tolerance value C_DYW_CAM_SYN_IN(EX), angular distances GAP-CAM including tolerance value C_DYW_CAM_CRK_SYN_RTD_IN(EX))

C_CRK_ANG_DLY = 720°CRK and C_CRK_ANG_DLY_LST_CAM = 720°CRK disables this functionality.

23.21 Camshaft Position Determination

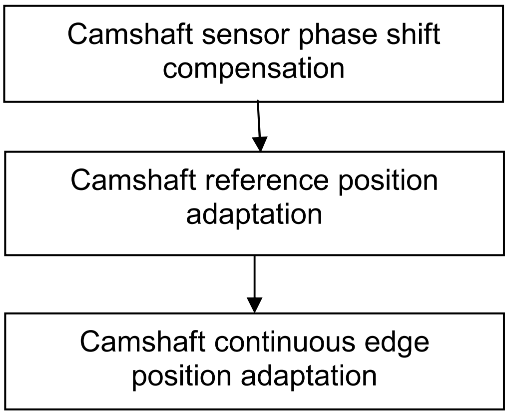
Calibration interfering functions:

INJR, VVTI, IGSP and IGRE pre-calibrated

Calibration data of this function are used for

- Camshaft position feedback for VVTI – function
- Crankshaft limp-home (Engine position and speed determination function)
- Camshaft position diagnosis (Engine position and speed determination function OBDI)

Calibration flowchart:



For system without VVT, only camshaft reference position adaptation has to be calibrated.

Calibration process:

Camshaft sensor phase shift compensation

The camshaft sensor phase shift vs. engine speed can be compensated by calibrating ID_CAM_PHA_COR.

Camshaft reference position adaptation

Camshaft reference adapation is performed when the VVT actuator is in passive position or when no VVT actuator is present in the system. The camshaft adaptation values represent the angular displacement of the camshaft signal edges relative to the designed position (NC_PSN_EDGE_z_CAM_IN(EX)_i). This angular displacement is caused by mechanical and sensor tolerances.

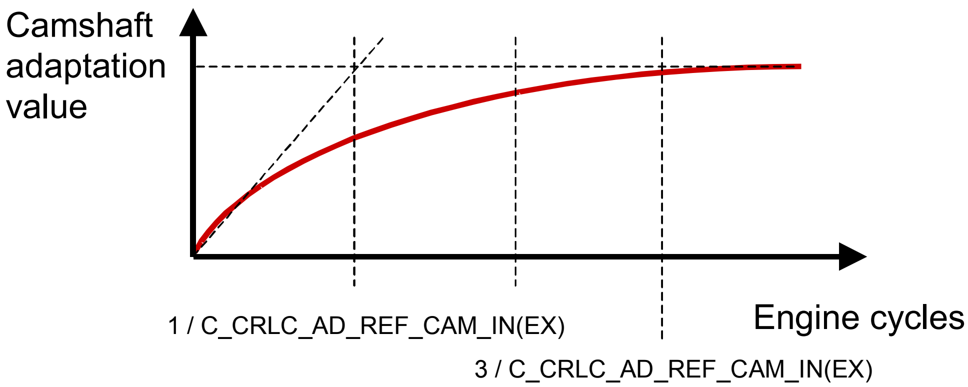
The camshaft reference position adaptation values are used for

- VVT position feedback
- Crankshaft limp-home

Calibration data C_NR_REV_AD_REF_CAM_IN(EX) defines the duration of the camshaft reference position adapation in number of engine cycles.

A filter with parameter C_CRLC_AD_REF_CAM_IN(EX) is applied to the camshaft edge adaptation values during the defined number of cycles.

Filter recurrency is the number of engine cycles. Steady state value is reached in 3/C_CRLC_AD_REF_CAM_IN(EX) engine cycles.



C_CRLC_AD_REF_CAM_IN(EX) = 1.0 (upper limit) means that steady state value is reached in 3 engine cycles (fastest value)

C_CRLC_AD_REF_CAM_IN(EX) = 0.0039 (lower limit) means that steady state value is reached in 3*256=768 engine cycles.

When the variation of camshaft edge positions during one camshaft revolution (VVT passive) is greater then the maximum gradient C_GRD_AD_REF_MAX_CAM_IN(EX), no reference position adaptation will be performed on the camshaft edge.

Note that adaptation values are applied immediately. If adaptation is calibrated very fast, the resulting jumps in the position output value that may be larger than C_GRD_AD_REF_MAX_CAM_IN(EX). In that case, adaptation is only performed every second engine cycle.

C_NR_REV_AD_REF_CAM_IN(EX) should be calibrated according to VVT function requirements. This calibration value determines the maximum dead time for VVT position feedback. In fact, continuous edge position adaptation cannot be started and, consequently, VVT position feedback is not available until the reference position adaptation is finished.

The filter on adaptation values has to converge before the end of reference adaptation.

Consequently, C_CRLC_AD_REF_CAM_IN(EX) should be calibrated as:

C_CRLC_AD_REF_CAM_IN(EX) > 3 / C_NR_REV_AD_REF_CAM_IN(EX

C_GRD_AD_REF_MAX_CAM_IN(EX) should be calibrated with respect to the max. allowed camshaft edge position variation with the VVT actuator in passive position. This value is generally very small.

Continuous edge position adaptation

Continuous edge position adapation is only performed when a VVT actuator is present in the system. It is activated when camshaft position is valid and reference position adaptation is not active.

Camshaft edges are adapted with reference to signal edge #1 (first falling edge after TDC0) on which only reference adaptation is done. Every angular displacement of edge #1 is considered as a camshaft displacement. The continous adaptation compensates the (unintentional) angular displacement of signal edges relative to the reference edge position due to temperature drift of the sensor etc.

A filter with time constant parameter C_CRLC_AD_CAM_IN(EX) is applied to the camshaft edge adaptation values during continuous edge position adaptation.

Filter recurrency is the number of engine cycles. Adaptation steady state value is reached in 3/C_CRLC_AD_CAM_IN(EX) engine cycles.

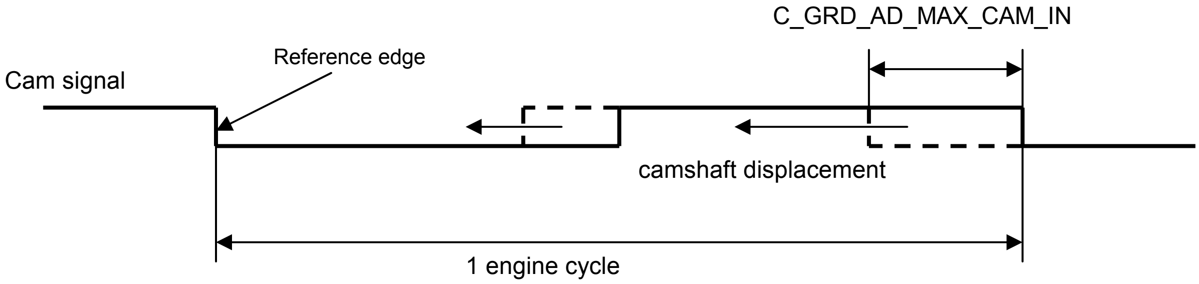
C_CRLC_AD_CAM_IN(EX) = 1.0 (upper limit) means that steady state value is reached in 3 engine cycles (fastest value)

C_CRLC_AD_CAM_IN(EX) = 0.0039 (lower limit) means that steady state value is reached in 3*256=768 engine cycles.

When the variation of camshaft edge positions between two consecutive camshaft reference edges (one engine cycle) is greater then the maximum gradient C_GRD_AD_MAX_CAM_IN(EX), no continuous edge position adaptation is performed.

The camshaft adaptation values should not be updated due to intentional VVT displacement during one engine cycle. Therefore C_GRD_AD_MAX_CAM_IN(EX) should be calibrated with respect to max. allowed camshaft edge variations with the VVT actuator in a fixed active position.

Example (half-moon target wheel, intake camshaft):

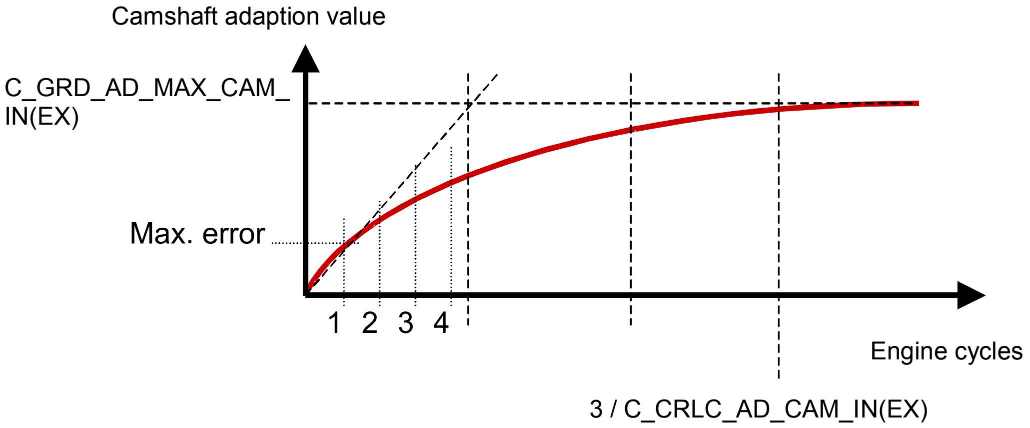


Calibration value C_CRLC_AD_CAM_IN(EX) should be calibrated according to the maximum tolerated error on VVT position during one engine cycle (VVT requirement):

C_CRLC_AD_CAM_IN(EX) = max. position error / C_GRD_AD_MAX_CAM_IN(EX)

(see figure below)

C_GRD_AD_MAX_CAM_IN(EX) is defined with respect to the maximum VVT actuator speed as above.

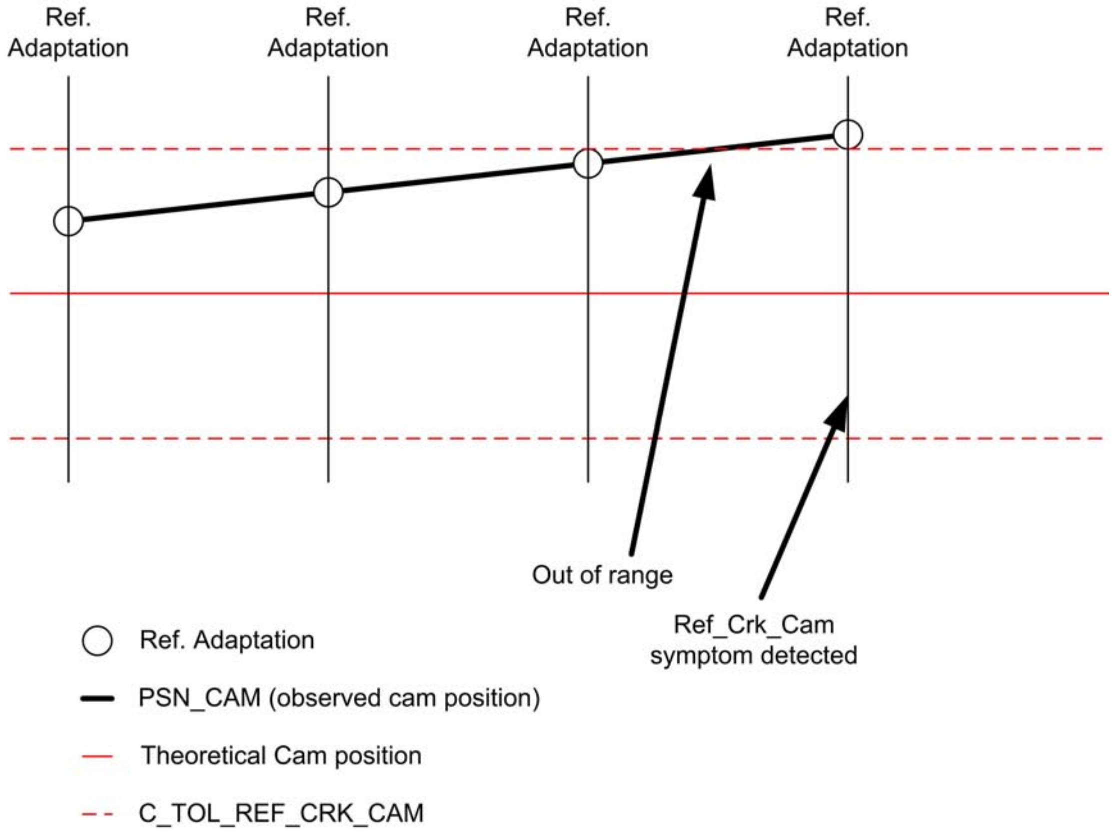


Generally, continuous position adapation should be much slower than reference position adaptation (C_CRLC_AD_CAM_IN(EX) << C_CRLC_AD_REF_CAM_IN(EX)).

Camshaft position diagnosis

REF_CRK_CAM symptom:

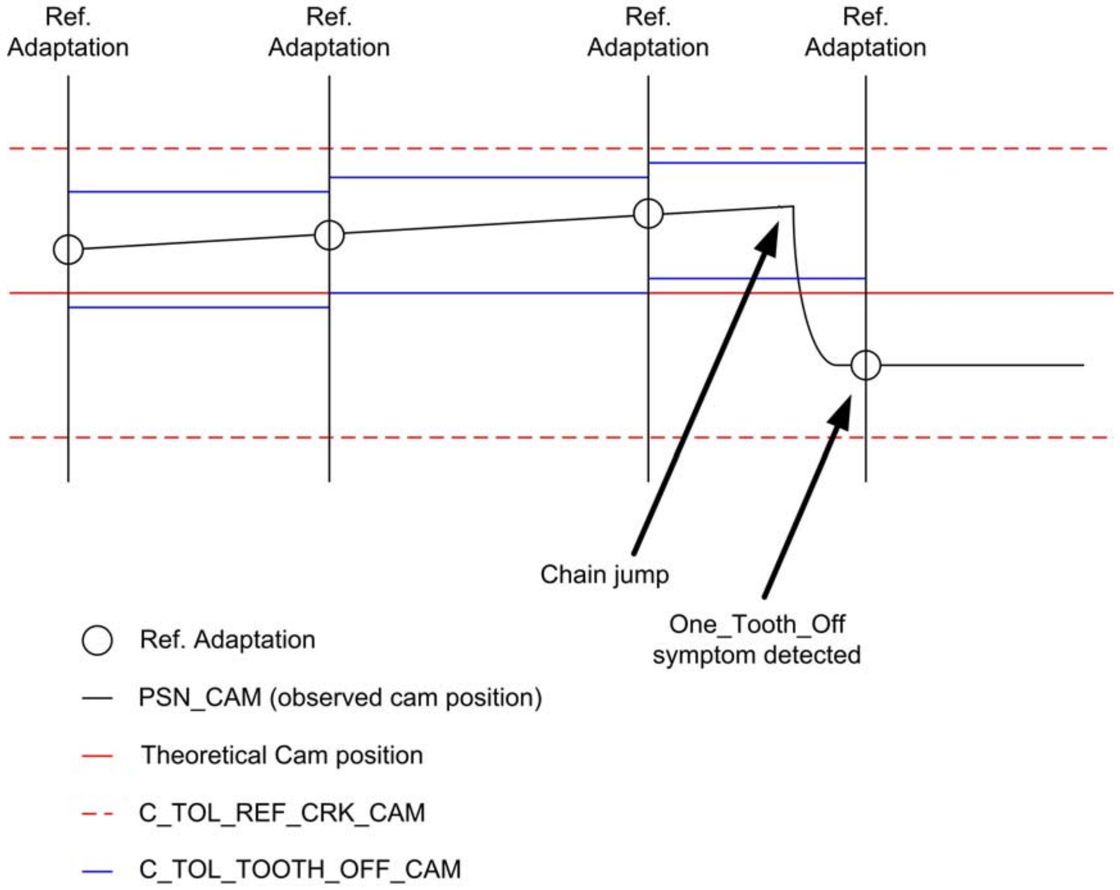
C_TOL_REF_CRK_CAM_IN and C_TOL_REF_CRK_CAM_EX represent the maximum tolerance between the adaptive camshaft edge position and the theoretical position. This difference can be due to mounting and electrical tolerances, and to slow deviation. If the observed camshaft position differs from the theory of more than C_TOL_REF_CRK_CAM_IN(EX), the symptom REF_CRK_CAM_IN(EX) is detected.



C_TOL_TOOTH_OFF_CAM_IN(EX) are not represented on this figure, because REF_CRK_CAM symptom is prioritary to ONE_TOOTH_OFF.

ONE_TOOTH_OFF symptom

C_TOL_TOOTH_OFF_CAM_IN and C_TOL_TOOTH_OFF_CAM_EX are used for ONE_TOOTH_OFF symptom detection. It represents the maximal sudden crk angle drift (between two consecutive adaptations) defined by OBDII to respect the emission standards. It can detect a quick jump of the camshaft edges positions which can be due to a chain jump for example.



To inhibit this diagnosis, C_TOL_TOOTH_OFF_CAM_IN(EX) have to be calibrated to 0.

23.22 Camshaft Sensor OBDI

Calibration interfering functions:

INJR, VVTI, IGSP and IGRE pre-calibrated

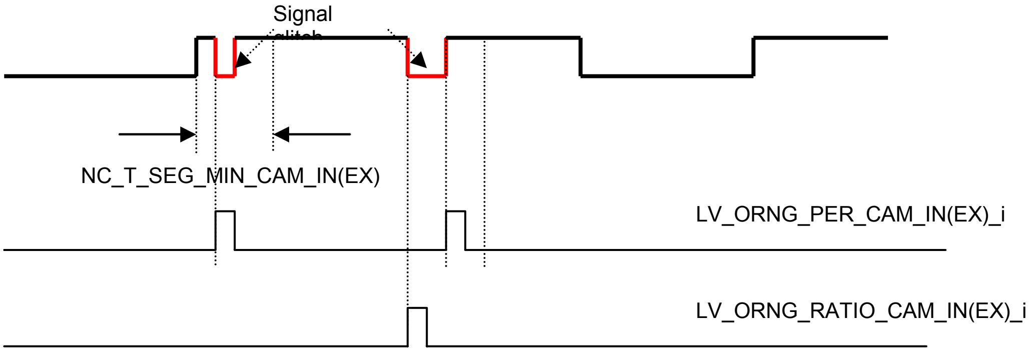
ACQ of CRK/CAM signal calibrated.

Calibration process:

The purpose of this diagnosis is to detect signal glitches on the camshaft sensor signal.

The camshaft sensor diagnosis is splitted in two parts:

- Camshaft segment period diagnosis based on segment time measurement
- Camshaft synchronization diagnosis based on camshaft ratio calculation



LV_ORNG_PER_CAM_IN(EX)_i is set if a spike occurs shortly after a valid camshaft signal edge.

LV_ORNG_RATIO_CAM_IN(EX)_i is set if a spike occurs between two valid camshaft signal edges. Note that the subsequent ratio calculations will also be wrong (ratio calculation done on last 2 (single tooth) or 4 (multi-teeth) camshaft segments).

The filter type can now be selected in Appl. Inc. file. In the case of a STD_INI (preferred solution), anti-bounce counter inc/max for camshaft segment period diagnosis should be set as:

C_ABC_INC_DIAG_PER_CAM_IN(EX) = NC_NR_EDGE_CAM_IN(EX)

C_ABC_MAX_DIAG_PER_CAM_IN(EX) = (NC_NR_EDGE_CAM_IN(EX) * max_symptom)

With this calibration, LV_ERR_PER_CAM_IN(EX)_i will be set

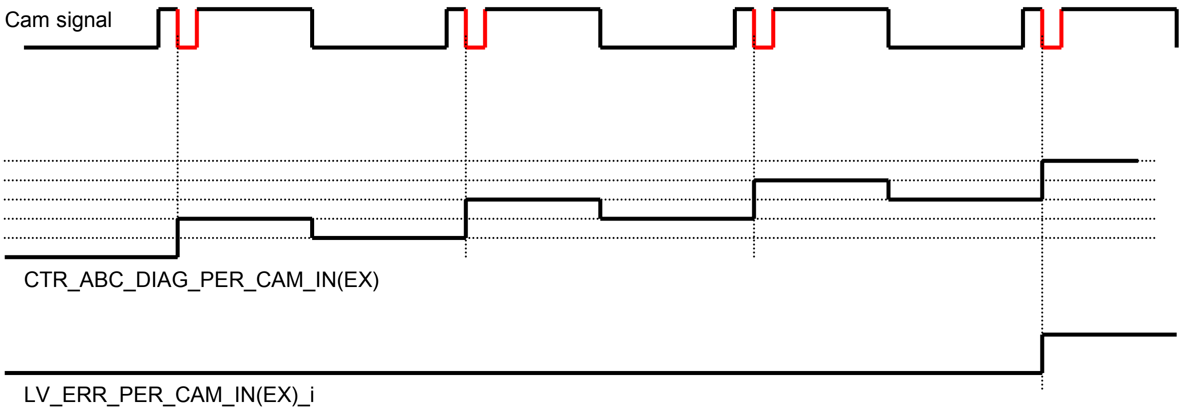
- as soon as a number of max_symptom succesive symptoms has occured on each camshaft edge
- as soon as a number of (C_ABC_MAX_DIAG_PER_CAM_IN(EX) – max_symptom) +1 succesive symptoms has occured during successive engine cycles on the same camshaft edge

For example:

half-moon camshaft target wheel

signal glitch on every rising signal edge max_symptom = 3

C_ABC_INC = 2, C_ABC_MAX = 6



If a camshaft ratio calculation is wrong, the camshaft self-synchronization is restarted. During the next 3 (single-tooth) or 5 (multi-teeth) camshaft edges no camshaft ratio will be calculated and the anti-bounce counter will be decremented by 3 or 5 (ENSD 4.0.0). Thus, the anti-bounce increment should be higher than 3 (single-tooth) or 5 (multi-teeth).

The filter type can now be selected in Appl. Inc. file. In the case of a STD_INI (preferred solution), anti-bounce counter inc/max for camshaft segment period diagnosis should be set as:

C_ABC_INC_DIAG_SYN_CAM_IN(EX) = NC_NR_EDGE_CAM_IN(EX) ... 2 * NC_NR_EDGE_CAM_IN(EX)

C_ABC_MAX_DIAG_SYN_CAM_IN(EX) = 2 * NC_NR_EDGE_CAM_IN(EX) ... 4 * NC_NR_EDGE_CAM_IN(EX)

Whatever the filter type is, the calibration should ensure that a periodic camshaft ratio out-of-range calculation produces a camshaft synchronization error.

The best way is to calibrate on car by generating electromagnetic disturbances on the signal lines (for example with a Schaffner on battery as used for EMI tests). No error should be set when generated disturbances are within the EMI levels which apply for the project.

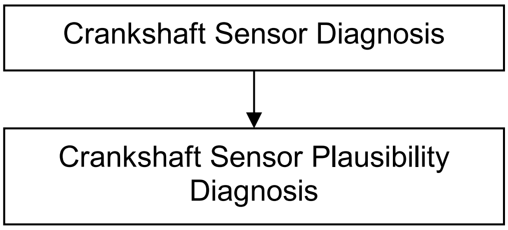
23.23 Crankshaft Sensor OBDI (ACPS)

Calibration interfering functions:

INJR, VVTI, IGSP and IGRE pre-calibrated.

ACQ of CRK/CAM signal calibrated.

Calibration flowchart:



Calibration process:

Crankshaft sensor diagnosis

The purpose of this diagnosis is to detect disturbances on the crankshaft signal and target wheel failures.

Open-circuits or short-circuit to battery or ground are detected with the crankshaft sensor circuit diagnosis.

Crankshaft sensor diagnosis depends on crankshaft synchronization status:

1) before synchronization:

A tooth period out-of-range flag is set (LV_ORNG_CRK_TOOTH_PER) and the corresponding anti-bounce counter incremented if an invalid crankshaft signal edge is detected (spike or missing tooth) after engine running detection. The signal edge is invalid when outside the crankshaft acceptance window defined by ID_FAC_TOL_CRK_TOOTH.

Tooth number diagnosis (LV_ERR_CRK_TOOTH) and loss of synchronization diagnosis (LV_ERR_CRK_SYN) are not activated.

2) after synchronization:

A tooth number out-of-range flag is set (LV_ORNG_CRK_TOOTH) and the corresponding anti-bounce counter incremented if one or more additional or missing teeth are detected during one engine revolution (information also to misfire-aggregate).

This diagnosis is inhibited below an engine speed threshold (engine start or rough engine stall) defined by C_N_32_MAX_DIAG_CRK_TOOTH.

However, even in this case, the information LV_INH_MIS_CRK is still sent to misfire-aggregate.

The value C_N_32_MAX_DIAG_CRK_TOOTH has to be calibrated lower than idle engine speed, but higher than starter speed.

If the number of tolerated additional or missing teeth is exceed during one engine cycle, a loss-of-synchronization flag is set (LV_LOST_SYN_CRK) and, for filter types with anti-bounce counters, the corresponding anti-bounce counter incremented.

If signal spikes are detected outside the crankshaft acceptance window defined by ID_FAC_TOL_CRK_TOOTH, a tooth period out-of-range flag is set and, for filter types with anti-bounce counters, the corresponding anti-bounce counter incremented at reference gap.

If the diagnoses filter types are set to MEM (preferred solution), or to filter types with anti-bounce counters:

Anti-bounce increment should be higher than 1 in order to detect sporadic crankshaft signal failures.

False detection of LV_ERR_CRK_TOOTH_PER may occur with interrupted start, or with vehicle abuse, e.g. high gear at very low speed. To suppress this, apply a very slow anti-bounce counter calibration. There is virtually no impact on ignition/injection timing, since signal disturbances will be corrected by the acquisition algorithm.

The number of re-synchronizations after loss of synchronization should be restricted to a very small number (max. 2 or 3) by calibrating C_ABC_MAX_DIAG_CRK_SYN to 1.5 to 3 times C_ABC_INC_DIAG_CRK_SYN. Engine could be damaged when timing error on injection/ignition is too high and, consequently, the engine should be definitely stopped (LV_ERR_CRK).

C_ABC_MAX_DIAG_CRK_TOOTH_PER should be calibrated higher than C_ABC_MAX_DIAG_CRK_TOOTH as the anti-bounce counter CTR_ABC_CRK_TOOTH_PER is incremented once at engine start when engine stopped close before reference gap (gap during valid teeth phase).

Crankshaft sensor plausibility diagnosis

The purpose of this diagnosis is to check if the crankshaft or camshaft signal is missing or implausible.

The anti-bounce counter CTR_ORNG_CAM_SYN_CRK is incremented by 1 if the camshaft signal edge is outside the tolerance window defined by C_DYW_CAM_CRK_SYN_ADC(RTD)_IN(EX) or if no camshaft signal was detected during one engine revolution (multi-teeth target wheel) as long as synchronization is not yet validated.

If more than C_NR_EDGE_CAM_IN(EX)_SYN_CRK_MAX camshaft edges have occurred and the engine is still not synchronized, LV_ERR_CRK_PLAUS is set.

The counter CTR_CYC_ENG_PLAUS_CAM_IN(EX)_i is incremented by 1 if no camshaft signal was detected during one engine cycle. It is decremented by 1 if at least one camshaft signal edge is detected during one engine cycle.

C_CTR_ORNG_CAM_SYN_CRK_MAX determines the time to detect camshaft error (for example missing signal) and to switch to camshaft limp-home mode or to another camshaft sensor. Consequently, it determines the starting time in case of camshaft sensor failure. Every time CTR_ORNG_CAM_SYN_CRK is incremented, the cam/crk synchronization is reset. The number of re-synchronizations should be limited to 1...5.

A camshaft error is set when CTR_ORNG_CAM_SYN_CRK exceeds C_CTR_ORNG_CAM_SYN_CRK_MAX. For single-tooth camshaft configurations, C_CTR_ORNG_CAM_SYN_CRK_MAX should be higher than 1 in order to avoid false cam error detection in case of early ignition key release during start.

C_NR_EDGE_CAM_IN(EX)_SYN_CRK_MAX determines the time to detect crankshaft error (for example missing signal) and to switch to crankshaft limp-home mode. Consequently, it determines the engine starting time in case of crankshaft sensor failure. C_NR_EDGE_CAM_IN(EX)_SYN_CRK_MAX must be higher than the number of camshaft edges per engine cycle.

C_NR_EDGE_CAM_IN(EX)_SYN_CRK_MAX >= NC_NR_EDGE_CAM_IN(EX)

If crankshaft failure occurs, low value of C_NR_EDGE_CAM_IN(EX)_SYN_CRK_MAX prevents engine stalling.

C_NR_CYC_ENG_PLAUS_CAM_MAX determines time to detect camshaft error when engine is running (engine will remain synchronized). During this time, camshaft position feedback will be wrong and VVT cannot work properly. Once camshaft error is detected, the VVT actuator switches to limp-home.

A camshaft error is set when C_NR_CYC_ENG_PLAUS_CAM_MAX engine cycles have occured without camshaft signal.

The diagnosis is inhibited with C_NR_CYC_ENG_PLAUS_CAM_MAX = 0.

Calibration data LC_CAM_ERR_ENA_VVT

If a camshaft error is detected at engine start and failure disappears during driving cycle:
- LV_ERR_SYN_CRK_CAM will be debounced and cam/crk diagnosis information updated if LC_CAM_ERR_ENA_VVT = 1
- LV_ERR_SYN_CRK_CAM remains 1 until next ignition key on transition if LC_CAM_ERR_ENA_VVT = 0

If IVVT - actuator can be reactivated during driving cycle after camshaft error, LC_CAM_ERR_ENA_VVT should be set to 1. If no reactivation is possible, LC_CAM_ERR_ENA_VVT should be set to 0.

Calibration data LC_CTR_DEC_PLAUS_CAM

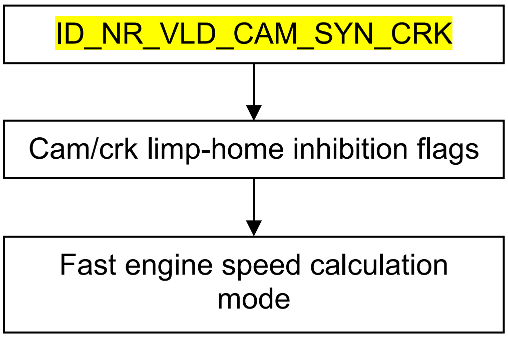
If a camshaft error is detected at engine start and failure disappears during driving cycle:
- LV_ERR_PLAUS_CAM_IN will be debounced and cam/crk diagnosis information updated if LC_CTR_DEC_PLAUS_CAM = 1 (STD_INI filter type behavior, preconized for IVVT-aggregate use)
- LV_ERR_PLAUS_CAM_IN remains 1 until next ignition key on transition if LC_CTR_DEC_PLAUS_CAM = 0 (MEM filter type behavior)

23.24 Engine Position and Speed Determination

Calibration interfering functions:

INJR, VVTI, IGSP and IGRE pre-calibrated

Calibration flowchart:



Calibration process:

Camshaft/Crankshaft Limp-Home

The following calibrations define if limp-home mode can be activated or not.

- LC_INH_LIH_CRK (crankshaft limp-home: crankshaft sensor failure, engine position determination with camshaft sensor)
- LC_INH_LIH_CAM (Camshaft limp-home: failure on all camshaft sensors, engine position determination with crankshaft sensor)

Calibration data ID_NR_VLD_CAM_SYN_CRK

Engine synchronization is validated as soon as a number of ID_NR_VLD_CAM_SYN_CRK valid camshaft signal edges has occurred after engine start .

During synchronization validation phase, the angular distance between cam/crk events are checked and engine synchronization is reset (LV_SYN_ENG) if one of the tests fails.

ID_NR_VLD_CAM_SYN_CRK should be set to the number of camshaft edges which should be checked. This value depends on camshaft/crankshaft configuration and project needs.

For DS- and DI-engines, injection is enabled as soon as LV_SYN_VLD is set to 1 (i.e. after a certain number of camshaft edges has been checked after engine start or loss of synchronization). The target is quick and safe engine start (no risk of wrong injection/ignition) at low engine speed but mainly safe engine resynchronization at high engine speed.

Generally, it is sufficient to check one camshaft edge (ID_NR_VLD_CAM_SYN_CRK = 1) at low engine speed but more than one at high engine speed (ID_NR_VLD_CAM_SYN_CRK = 4).

The risk of wrong injection/ignition is further reduced with ID_NR_VLD_CAM_SYN_CRK >1 but engine starting time increases.

- For engine start positions where no camshaft edge occurs before reference gap, engine position could be wrong when LV_SYN_ENG is set (e.g. camshaft sensor signal is missing). Engine position is validated as soon as LV_SYN_VLD is set.
- It is preferable to set ID_NR_VLD_CAM_SYN_CRK > 0 in order to reset engine synchronization (and disable injection/ignition) in case of wrong engine position at reference gap. Generally, for DS and DI engines, injection should be enabled on LV_SYN_VLD.

Fast Engine Speed Computation

A fast engine speed (N_FAST) can be calculated depending on the state of switch C_N_FAST_SWI:

C_N_FAST_SWI = 0:

Fast engine speed calculation de-activated. No additional CPU load.

C_N_FAST_SWI = 1:

Fast engine speed calculation based on last crankshaft tooth period T_TOOTH. Calculation is done until speed threshold C_N_TOOTH_END is reached. N_FAST based on last tooth period is used to calculate more precisely injection/ignition events during engine start (C_N_TOOTH_END = 1200 ... 1800rpm).

C_N_FAST_SWI = 2:

Fast engine speed calculation based on half-segment period calculation.

C_N_FAST_SWI = 3:

Fast engine speed calculation based on crankshaft tooth window. A filter with time constant C_FAC_N_FAST is applied:

N_FAST = N + C_FAC_N_FAST + (N_CRK_WIN n – N_CRK_WIN n-1)

C_N_FAST_SWI = 4 is used only for development purposes.

The output data N_TOOTH is calculated independently of calibration switch C_N_FAST_SWI until speed threshold C_N_TOOTH_END. This data is also used to determine the activation condition of the reverse engine rotation detection and the fail safe delay. A necessary condition to start the fail safe delay is that N_TOOTH is lower then C_N_NOT_REST. Therefore, N_TOOTH shall be calculated at least until C_N_NOT_REST:

C_N_TOOTH_END + 32rpm > C_N_NOT_REST

(+32 rpm for resolution conversion)

Calibration data ID_CRK_PHA_COR

Calibration data ID_CRK_PHA_COR represents the angular shift of the crankshaft active edges in function of engine speed. This angular shift is generally very small.

This functionality is not activated and, therefore, ID_CRK_PHA_COR should be calibrated to 0 (zero).

Calibration data ID_CAM_PHA_COR

Calibration data ID_CAM_PHA_COR represents the angular shift (in °CRK) of the camshaft signal edges in function of engine speed. For most camshaft sensors, this phase shift is small and ID_CAM_PHA_COR can be calibrated to zero.

Calibration of engine backwards rotation detection:

This functionality uses calibration already used for cam/crk synchronisation (C_DYW_CAM_CRK_SYN_ADC_IN(EX)_i and C_DYW_CAM_CRK_SYN_RTD_IN(EX)_i) and crankshaft acquisition (ID_FAC_TOL_CRK_TOOTH which sets LV_CRK_MISS_TOOTH). Check that detection and confirmation are correct with the values already calibrated.

Backward rotation detection is based on missing tooth simulation below an engine speed threshold. A change of rotation direction can only be detected if ID_FAC_TOL_CRK_TOOTH is not too high for engine speed breakpoints lower than C_N_NOT_REST.

Calibration data C_T_DLY_STOP_ENG:

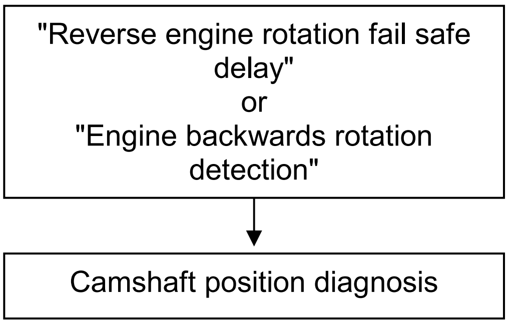
A crankshaft error plausibility symptom rises up ("signal missing") if LV_CRK_MISS_RUN_ENG is set after an engine stop request indicating that engine is still turning but engine speed is not available (f.ex. crk sensor failure). This information can be used for other functions/ECUs to prevent switching to passive state. LV_CRK_MISS_RUN_ENG is reset after a delay time C_T_DLY_STOP. C_T_DLY_STOP_ENG should be calibrated as the time interval between an engine stop request (crankshaft sensor failure) and engine resynchronization in crankshaft limp-home mode (or crankshaft failure setting when crankshaft limp-home is inhibited). This feature can be inhibited with C_T_DLY_STOP_ENG = 0 when not used.

Calibration interfering functions:

INJR, VVTI, IGSP and IGRE pre-calibrated

C_N_TOOTH_END has to be calibrated before calibration of reverse engine rotation fail safe delay.

Calibration flowchart:



Calibration process:

Reverse engine rotation fail safe delay

Crankshaft and camshaft diagnosis are inhibited during the reverse engine rotation fail safe delay.

The reverse engine fail safe delay is activated:

- on engine stop if the last calculated engine speed is lower than C_N_NOT_REST and no reverse engine rotation has been detected. Duration is C_T_DLY_REST_STALL
- If reverse engine detection is inhibited on ignition key off as soon as the engine speed drops below C_N_NOT_REST. Duration is C_T_DLY_REST_IGK

The purpose of the fail safe delay is to prevent false error detection in case of reverse engine rotation.

Set calibration data for reverse engine fail safe delay to a high value (for example: C_N_NOT_REST = 1000rpm, C_T_DLY_REST_IGK(STALL) = 5s).

Display all camshaft and crankshaft diagnosis symptoms with SAM2000 (lowest possible sampling, for example: 5ms), N_TOOTH and LV_STOP_ENG.

At least, two vehicle tests should be done:

a) low rpm clutch engagement (3^{rd} gear)

- stall the engine with 3^{rd} gear engaged, engine turning with idle speed
- start engine with 3^{rd} gear engaged, vehicle stopped

From SAM2000 measurement:

- determine engine speed (N_TOOTH) when engine stop is set (LV_STOP_ENG) or when synchronization is reset. This value should be calibrated for C_N_NOT_REST (repeat several times and apply a security margin)

- determine the time duration between engine stop and the instant when LV_FIRST_VLD_TOOTH is definitely zero. This value should be calibrated for C_T_DLY_REST_STALL (repeat several times and apply a security margin).

b) stop engine shortly after engine synchronization (rapid key on – key off).

As before: determine time duration between engine stop and the instant when LV_FIRST_VLD_TOOTH is definitely zero. Calibrate this value for C_T_DLY_REST_IGK.

Note that N_TOOTH is used to determine the activation condition of the reverse engine rotation fail safe delay. A necessary condition to start the fail safe delay is that N_TOOTH is lower then C_N_NOT_REST. Therefore, N_TOOTH shall be calculated at least until C_N_NOT_REST (C_N_TOOTH_END has to be calibrated accordingly).

Engine backwards rotation detection

Engine backwards rotation detection is activated if LC_ENG_BACK_INH is set to 0.

Crankshaft and camshaft diagnosis are inhibited when a reverse engine rotation has been detected until its confirmation or nullification.

Inhibition is stopped and cam/crk diagnosis reactivated as soon as:

- delay time C_T_MAX_INH_DIAG_ENSD has elapsed without ignition key transition **or**
- engine resynchronizes and synchronization is validated **or**
- delay time C_T_MIN_INH_DIAG_ENSD has elapsed after a key off transition **or**
- ignition key off and delay time C_T_MIN_INH_DIAG_ENSD elapsed after engine stop **or** backward rotation confirmation

Tests to be done for calibration of engine backwards rotation

a) low rpm clutch engagement (3rd gear)

- stall the engine with 3rd gear engaged, engine turning with idle speed
- start engine with 3rd gear engaged, vehicle stopped
- start engine with reverse gear engaged, vehicle running forward

b) stop engine shortly after engine synchronization (rapid key on – key off).

Determine engine speed when backwards rotation occurs: calibrate this value for C_N_NOT_REST with a margin.

C_T_MIN_INH_DIAG_ENSD should be set to the time necessary to stall engine (at low engine speed) after a key on to off transition. This calibration value is similar to C_T_DLY_REST_IGK.

C_T_MAX_INH_DIAG_ENSD is the maximum delay before reactivating diagnosis after engine backwards rotation confirmation. It should be set to the maximum duration of reverse rotation if diagnosis inhibition until LV_SYN_VLD is not acceptable, otherwise it should be set to its maximum value.

Camshaft position diagnosis

The purpose of this diagnosis is to detect a abnormal angular shift of the camshaft sensor signal edges relative to the crankshaft. A error symptom is set when one of the camshaft adaptation values is outside the defined tolerance range.

The tolerance window C_TOL_REF_CRK_CAM_IN(EX) shall include all mechanical and electrical tolerances of the sensors, the target wheels and the mounting:

- mounting tolerances of camshaft and crankshaft sensor
- mounting tolerances of camshaft and crankshaft target wheel
- camshaft and crankshaft sensor accuracy
- machining of camshaft and crankshaft target wheel
- angular shift of crankshaft to crankshaft target wheel position

Sum up all these tolerance values for the intake and exhaust camshaft and set calibration values C_TOL_REF_CRK_CAM_IN(EX) accordingly.

Consider also that mechanical tolerances and angular shift of crank to cam can vary during engine lifetime (for example: increased belt/chain length).

In contrast to tolerance window C_DYW_CAM_CRK_SYN_ADC(RTD)_IN(EX), the following features shall **not** be considered for in C_TOL_REF_CRK_CAM_IN(EX):

- camshaft sensor pre-calibration mode (sensor self-adaptation)
- VVT actuator range

This is the reason why C_TOL_REF_CRK_CAM_IN(EX) is generally lower than C_DYW_CAM_CRK_SYN_ADC(RTD)_IN(EX).

Anti-bounce counter inc/max for camshaft position diagnosis should be set as

C_ABC_INC_DIAG_REF_CRK_CAM_IN(EX) = NC_NR_EDGE_CAM_IN(EX)

C_ABC_MAX_DIAG_REF_CRK_CAM_IN(EX) = (NC_NR_EDGE_CAM_IN(EX) * max_symptom)

With this calibration, LV_ERR_REF_CRK_CAM_IN(EX)_i will be set

- as soon as a number of max_symptom succesive symptoms has occured on each camshaft edge
- as soon as a number of (C_ABC_MAX_DIAG_REF_CRK_CAM_IN(EX) – max_symptom) +1 succesive symptoms has occured during successive engine cycles on the same camshaft edge

23.26 Customer adaptation module: ENSD (Engine Position and Speed Determination)

23.26.1 Outputs for BMW functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_kwnotl	V/O	0...1H	0...1	1	[-]
error detected at crank engine position determination system					
B_kwdreht	V/O	0...1H	0...1	1	[-]
engine speed > 0 rpm					
Kw_pos	V/O	0...FFFFH	0...1535.97656	0.0234375	[°CRK]
Engine position, absolute					
Nkw	V/O	8000...7FFFH	-32768...32767	1	[rpm]
engine speed					
Segakt	V/O	0...FFH	0...255	1	[-]
segment number					
Seg5z[24]	V/O	0...FFFFH	0...65.535	0.001	[ms]
Segment time for 5 tooth					
Nkw_grad	V/O	80...7FH	-4096...4064	32	[rpm/s]
engine speed gradient					
Vse_adp_ext	V/O	0...FFFFH	0...6553.5	0.1	[°CRK]
Theoretical engine position for intake camshaft i edge 6					
Vsa_adp_ext	V/O	0...FFFFH	0...6553.5	0.1	[°CRK]
Theoretical engine position for exhaust camshaft i edge 6					
Nkw_zahn	V/O	8000...7FFFH	-32768...32767	1	[rpm]
Engine speed calculated from T_TOOTH					

Input data:

LV_FIRST_VLD_TOOTH	N	SEG_NR	LV_ERR_CRK
N_GRD	LV_ES	PSN_ENG	LV_SYN_VLD
NC_NR_CAM_CBK	NC_PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	NC_PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	N_TOOTH_CUS
NC_CYL_NR	LV_ST_END	LV_T_SEG_5_TOOTH_VLD	

Import actions:

ACTION_INFR_GetT5T(IN <idx>, **OUT** <time>**)**
Acquisition of partial segment time over 5 teeth

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation **at reset:**

```
If      NC_CYL_NR = 6
        Vse_adp_ext  =    NC_PSN_EDGE_CAM_IN[6][1]
        Vsa_adp_ext  =    NC_PSN_EDGE_CAM_EX[6][1]
Else
        Vse_adp_ext  =    NC_PSN_EDGE_CAM_IN[1][1]
        Vsa_adp_ext  =    NC_PSN_EDGE_CAM_EX[1][1]
end
```

0: B_kwnotl, B_kwdreht, Nkw, Segakt, Nkw_grad, Nkw_zahn, Seg5z[24]

720°CRK: Kw_pos

at LV_ES 0-> 1: B_kwdreht = 0, Nkw = 0

at PWL : 0: B_kwnotl, Segakt, Nkw_grad

Recurrence :

B_kwnotl, Segakt:	segment
Kw_pos, Nkw_grad, Nkw_zahn, B_kwdreht:	10ms
Vse_adp_ext, Vse_adp_ext:	once at reset
Seg5z[24]:	Segment, after "Segment time acquisition for partial segments"

Nkw:

```
If LV_ST_END = 0
Then   recurrence is 10ms
Else   recurrence is segment
Endif
```

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
B_kwnotl   =   LV_ERR_CRK
B_kwdreht  =   LV_FIRST_VLD_TOOTH
Nkw        =   N
Segakt     =   SEG_NR
If(1)      LV_T_SEG_5_TOOTH_VLD = 1
then(1)    transfer of measured times from BSW to ASW
           Seg5z(idx) = ACTION_INFR_GetT5T(
                                  IN      idx
                                  OUT     time)
           For idx derived from SEG_NR as depicted in following table:
```

SEG_NR	0	1	2	3	4	5
idx	16...19	20...23	0...3	4...7	8...11	12...15

```
  else(1)  For idx = 0...23
           Seg5z(idx) =  FFFFH
endif(1)


If   LV_SYN_VLD = 1
     Kw_pos        =      PSN_ENG
Else
     Kw_pos        = 720 °CRK
end
Nkw_grad           =      N_GRD
Nkw_zahn           =      N_TOOTH_CUS
```

24 Engine speed limitation

Table of Contents

Index

24.1 Engine Speed Limit Coordination

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_MAX_MT	-	0...1FE0H	0...8160	1	[rpm]
Engine speed limit for stopped MT-vehicle					
N_MAX_AT	-	0...1FE0H	0...8160	1	[rpm]
Engine speed limit for stopped AT-vehicle or at high vehicle speed					
LV_N_MAX_H	V	0...1H	0...1	1	[-]
Boolean for increased engine speed limitation active					
N_MAX_LIH_ETC	-	0...1FE0H	0...8160	1	[rpm]
Engine speed limit in case of throttle limp home required by throttle monitoring or ETC safety concept					
N_MAX_LIH_MTC	-	0...1FE0H	0...8160	1	[rpm]
Engine speed limit in case of MTC error					
N_MAX_THD	O/V	0...1FE0H	0...8160	1	[rpm]
Actual engine speed limit					
N_MAX_VS_MAX	V	0...1FE0H	0...8160	1	[rpm]
Engine speed limit in case of vehicle speed limitation					
LV_VS_MAX	O/V	0...1H	0...1	1	[-]
Boolean for vehicle speed limitation active					
VS_RATIO	V	0...FFFFH	0...31.99951	0.4883e-3	[-]
Ratio between vehicle speed limit and actual vehicle speed					
N_MAX_THD_1	O/V	0...1FE0H	0...8160	1	[rpm]
Actual engine speed limit depending on TOIL-orange area (BN2000)					
N_MAX_THD_2	O/V	0...1FE0H	0...8160	1	[rpm]
Actual engine speed limit depending on TOIL-red area (BN2000)					
LV_N_MAX_ETC_LIH	O/V	0...1H	0...1	1	[-]
engine speed limitation (MTC or ETC_LIH) active					
T_GEAR_VS_MAX	V	0...FFH	0...25.5	0.1	[s]
Time since the last gear shift.					
N_MAX_EGY	-	0...1FE0H	0...8160	1	[rpm]
Engine speed limit in case of activated energy spare mode					
N_MAX_DROF	-	0...1FE0H	0...8160	1	[rpm]
Engine speed limit in case of jump start					
N_MAX_LIH_PRS_OIL	V	0...1FE0H	0...8160	1	[rpm]
Engine speed limit for required oil pressure limp home mode					
STATE_N_MAX_THD_LIH	O/V	0...3H	0...3	1	[-]
Active engine speed reduction due to LIH (0..no reduce, 1..minor reduce, 2..major reduce)					
N_MAX_THD_LIH	V	0...1FE0H	0...8160	1	[rpm]
Actual engine speed limit due to LIH					
N_MAX_LIH_EGS	V	0...1FE0H	0...8160	1	[rpm]
Engine speed limit for AT in case of CAN error to EGS					

Input data:

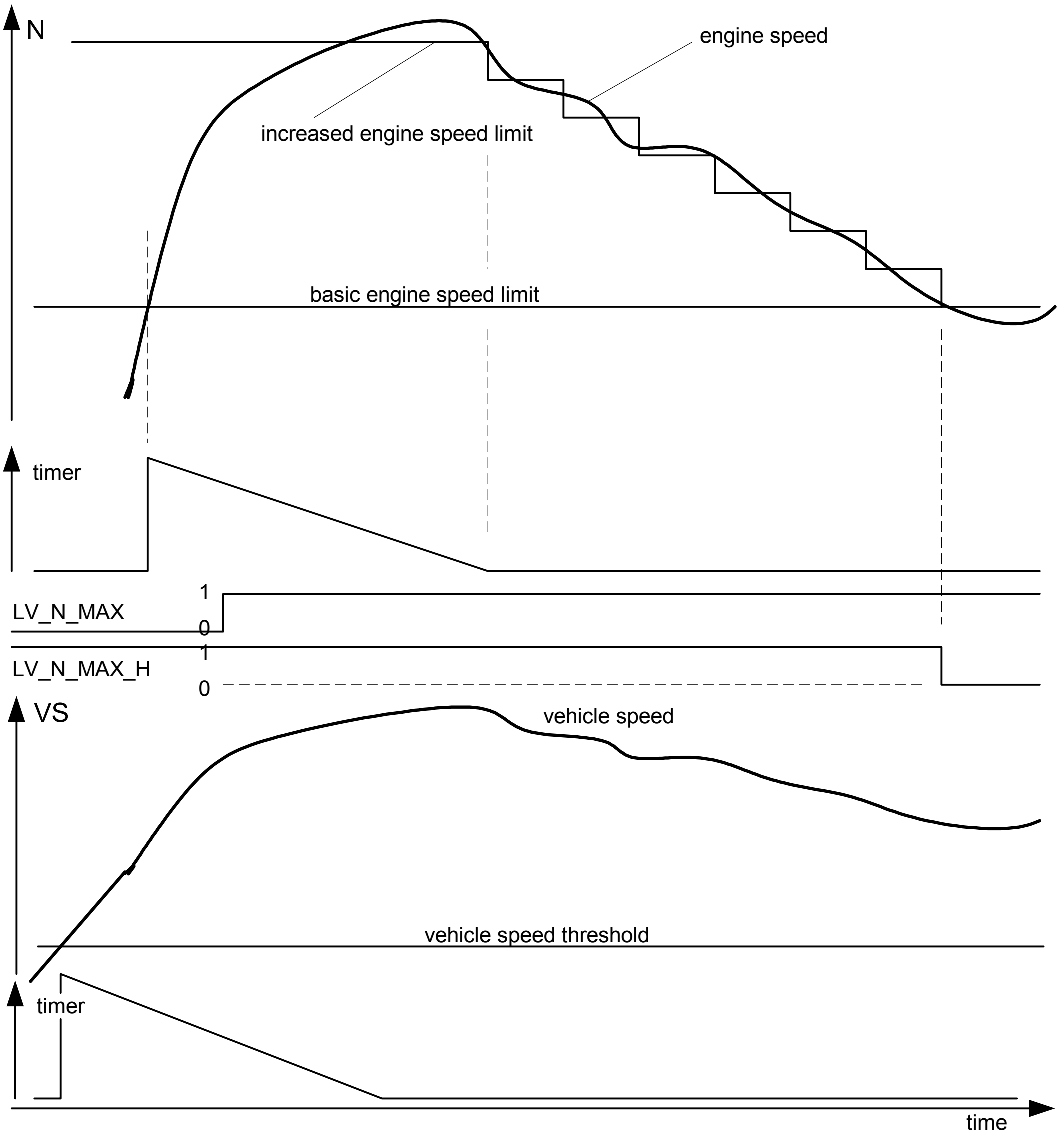
VS	GEAR		LV_AT
LV_N_MAX	LV_TPS_MTC_N_LIM	LV_MTC_LIH_ACT	N
PV_AV	LV_MTC_LIH_CUR_OFF	LV_N_LIM_REQ_MON	LV_MAF_BLS_DIAG
	LV_GP	LV_GS	STATE_GEAR_REV_AT_AMT
		TCO	LV_CS
LV_VAR_BN	LV_VAR_TCT	LV_ERR_ECU_CKS	LV_ERR_ECU_RAM
VS_MAX_SEL_EXT_REQ	N_MAX_LIH_STATE_CMB_S	STATE_EGY_MIN_KWP	LV_INH_GP_SUP
N_LIH_PRS_OIL	LV_N_LIM_REQ_RST_CHK	STATE_ETCU_PROG_INFO	LV_ETCU_DISABLE_CAN
STATE_ST_TQ_LIM_GS	C_VS_MAX_0_DFT	C_VS_MAX_1	C_VS_MAX_2

C_VS_MAX_3		CTR_KM_CAN	LV_ST_END
T_AST			

FUNCTION DESCRIPTION:

General information:

Several conditions can activate an engine speed limit. The actual engine speed limit is calculated as minimal selection of all requests.



Signal flow diagram

If the vehicle speed is below a threshold or exceeded this threshold not more than a calibrateable time before, an increased speed limit is valid for a short time (this is **not** true, if the transmission variant is AT and the reverse gear is engaged, i.e. if LV_AT and STATE_GEAR_REV_AT_AMT = 1). So if the engine speed exceeds the basic engine speed limit while the time after VS exceeded C_VS_THD_N_MAX_H_xx is at the most C_T_MAX_N_MAX_H_xx, the increased engine speed limit is valid for a calibrateable time.

After this time the valid limit is changed towards the basic engine speed limit via calibrateable steps. If C_T_MAX_N_MAX_H_xx is terminated without limitation being active (LV_N_MAX = 0), N_MAX_xx is set from increased speed limit to basic engine speed in one step.

A jump back to the increased engine speed limit is possible not before the engine speed limitation is finished (LV_N_MAX = 0). Furthermore the VS condition must be fulfilled (still or again). All the 4 parameters

- basic engine speed limit
- increased engine speed limit
- time for increased engine speed limit and
- stepwidth to basic engine speed limit

are seperately calibrateable for the 3 vehicle variants

- automatic or twin clutch transmission
- manual transmission
- automatic manual transmission

- automatic ot twin clutch transmission in manual mode 'M'

and dependant from the actual gear.

C_VS_THD_N_MAX_H_xx and C_T_MAX_N_MAX_H_xx are seperately calibrateable for the 3 vehicle variants (xx = MT/ AT).

Above a specific engine speed, the calculation of the stratified combustion mode is switched off. To enable a stratified limp home mode even when the stratified combustion mode calculation is switched off, the engine speed limitation N_MAX_LIH_STATE_CMB_S is introduced to reactivate stratified mode calculation.

Application conditions:

Initialisation:
N_MAX_THD = NC_N_REF_MAX (= 8160 rpm)
N_MAX_THD_1 = 0
N_MAX_THD_2 = 0
LV_VS_MAX = 0
N_MAX_THD_LIH = NC_N_REF_MAX (= 8160 rpm)
N_MAX_LIH_EGS = NC_N_REF_MAX (= 8160 rpm)
STATE_N_MAX_THD_LIH = 0

Recurrence: 100 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

a1) Basic engine speed limit :

Manual transmission vehicle

```
If      (LV_AT = 0   and LV_VAR_TCT = 0)
  Then  N_MAX_MT   =  ID_N_MAX_MT
  Else  N_MAX_MT   =  NC_N_REF_MAX
Endif
```

Automatic transmission or twin clutch transmission vehicle

```
If      (LV_AT = 1 or LV_VAR_TCT = 1)
Then
        If      STATE_GEAR_REV_AT_AMT = 0
        then    if    [ (( STATE_ETCU_PROG_INFO=1 and LV_VAR_BN=0 ) or
                           ( STATE_ETCU_PROG_INFO=2 and LV_VAR_BN=1 )     )  and
                          LC_VAR_ETCU_SPT_SWI = 0 ]   or

                      [ LV_INH_GP_SUP = 1  and  LC_VAR_ETCU_SPT_SWI = 1 ]

                Then        N_MAX_AT   =   ID_N_MAX_AT_MAN

                else        N_MAX_AT   =   ID_N_MAX_AT

                endif

        Else    N_MAX_AT  =   C_N_MAX_DRI_RVL
        Endif
Else    N_MAX_AT           =   NC_N_REF_MAX
Endif
```

Remark: a change of the vehicle-(transmission-)type is done only during VS = 0 km/h.

a2) Parameters for increased engine speed limit :

Parameter	Vehicle type *)	Value
VS-threshold	1)	C_VS_THD_N_MAX_H_MT
	3)	C_VS_THD_N_MAX_H_AT
	4)	C_VS_THD_N_MAX_H_AT_MAN
increased engine speed limit	1)	ID_N_MAX_H_MT
	3)	ID_N_MAX_H_AT
	4)	ID_N_MAX_H_AT_MAN
time before decrementation begins	1)	ID_T_N_MAX_H_MT
	3)	ID_T_N_MAX_H_AT
	4)	ID_T_N_MAX_H_AT_MAN
decrease stepwidth	1)	ID_N_DEC_N_MAX_H_MT
	3)	ID_N_DEC_N_MAX_H_AT
	4)	ID_N_DEC_N_MAX_H_AT_MAN

*) Vehicle types :		
	1)	Manual transmission vehicle
	2)	Automatic manual transmission vehicle
	3)	Automatic/Twin clutch transmission vehicle
	4)	Automatic/Twin clutch transmission vehicle in 'M' (manual) mode

Functionality (N_MAX_MT/ AT = ...) : see signal flow diagram above.

Status bit for increased engine speed limit:

```
If   ID_N_MAX_MT < N_MAX_MT < NC_N_REF_MAX
     OR
     [ ID_N_MAX_AT < N_MAX_AT < NC_N_REF_MAX          and
```

```
      STATE_GEAR_REV_AT_AMT = 0                              and
      { (((STATE_ETCU_PROG_INFO != 1  and  LV_VAR_BN = 0) or
          (STATE_ETCU_PROG_INFO != 2  and  LV_VAR_BN = 1)  ) and
         LC_VAR_ETCU_SPT_SWI = 0 )     or
        ( LV_INH_GP_SUP = 0  and  LC_VAR_ETCU_SPT_SWI = 1 )  }  ]

    OR

    [ ID_N_MAX_AT_MAN < N_MAX_AT < NC_N_REF_MAX   and
      STATE_GEAR_REV_AT_AMT = 0                      and
      { (((STATE_ETCU_PROG_INFO = 1  and  LV_VAR_BN = 0) or
          (STATE_ETCU_PROG_INFO = 2  and  LV_VAR_BN = 1)   ) and
         LC_VAR_ETCU_SPT_SWI = 0 )     or
      ( LV_INH_GP_SUP = 1  and  LC_VAR_ETCU_SPT_SWI = 1 )  }  ]

  Then   LV_N_MAX_H  =  1

  Else   LV_N_MAX_H  =  0

Endif
```

a4) Engine speed limit if limp home mode required by throttle monitoring

In case of limp home request from main throttle controller a lower engine speed limit should be active.

```
If          LV_N_LIM_REQ_MON            ==1        or
            LV_TPS_MTC_N_LIM            ==1        or
            LV_MTC_LIH_ACT              ==1        or
            LV_N_LIM_REQ_RST_CHK        ==1        or
            LV_MAF_BLS_DIAG             ==1        or
            LV_ERR_ECU_CKS              ==1        or
            LV_ERR_ECU_RAM              ==1

  Then      N_MAX_LIH_ETC       =    C_N_MAX_MTC_LIH
  Else      N_MAX_LIH_ETC       =    NC_N_REF_MAX
Endif
```

a5) Engine speed limit if limp home mode only due to MTC error

```
If          LV_MTC_LIH_CUR_OFF   ==  1

Then        N_MAX_LIH_MTC          =   IP_N_MAX_MTC_CUR_OFF( PV_AV )

Else        N_MAX_LIH_MTC          =   NC_N_REF_MAX

Endif


If          N_MAX_LIH_MTC or N_MAX_LIH_ETC not  NC_N_REF_MAX

  Then      LV_N_MAX_ETC_LIH = 1

  Else      LV_N_MAX_ETC_LIH = 0

Endif
```

a6) Engine speed limit for vehicle speed limitation

In case of vehicle speed limitation the engine speed limitation will be activated.

```
If(1)                  LV_GP   =  0
  Then(1)
    If(2)              VS_MAX_SEL_EXT_REQ  =  1
      Then(2)          VS_MAX  =  C_VS_MAX_1
      Else(2)
        If(3)          VS_MAX_SEL_EXT_REQ  =  2
          Then(3)      VS_MAX = C_VS_MAX_2
          Else(3)
            If(4)      VS_MAX_SEL_EXT_REQ  =  3
              Then(4)  VS_MAX  =  C_VS_MAX_3
              Else(4)  VS_MAX  =  C_VS_MAX_0_DFT
            Endif(4)
        Endif(3)
    Endif(2)
  Else(1)              VS_MAX  =  255
Endif(1)

VS_RATIO        =       VS_MAX / VS

If(1)                  (VS_MAX < 255)  AND  LC_N_MAX_VS_MAX_ACT = 1
  Then(1)
    If(2)              (VS_RATIO <= 1.0) ("Hysteresis")
      Then(2)          LV_VS_MAX = 1
      Else(2)
        If(3)          VS <= VS_MAX – C_VS_MAX_HYS
          Then(3)      LV_VS_MAX = 0
        Endif(3)
        If(4)          GEAR ≠ 0                                  and
                       GEAR ≠ GEAR_PREV_VS_MAX
          Then(4)      GEAR_PREV_VS_MAX = GEAR
                       LV_GEAR_SHIFT_VS_MAX = 1
            If(5)      LV_AT = 1 or LV_VAR_TCT = 1
              Then(5)  T_GEAR_VS_MAX = C_T_GEAR_VS_MAX_AT
              Else(5)  T_GEAR_VS_MAX = C_T_GEAR_VS_MAX_MT
```

```
            Endif(5)
          Else(4)      decrement T_GEAR_VS_MAX
                       LV_GEAR_SHIFT_VS_MAX = 0
        Endif(4)

        If(6)          VS > VS_MAX – C_VS_MAX_HYS
          Then(6)
            If(7)      [N_MAX_VS_MAX = NC_N_REF_MAX    or
                       LV_GEAR_SHIFT_VS_MAX = 1        or
                       T_GEAR_VS_MAX ≠ 0]              or
                       ((LV_AT = 1 or LV_VAR_TCT = 1)  and
                       STATE_CC ≠ "Closed")
              Then(7)  N_MAX_VS_MAX = N * VS_RATIO,
              Else(7)  N_MAX_VS_MAX remains
            Endif(7)
          Else(6)      N_MAX_VS_MAX = NC_N_REF_MAX
        Endif(6)
    Endif(2)
  Else(1)              {Engine speed limit for vehicle speed limitation deactivated
                       LV_VS_MAX        =  0
                       N_MAX_VS_MAX =  NC_N_REF_MAX }
Endif(1)
```

a7) Engine speed limit for energy spare mode (production, transportation or garage mode)

```
IF        STATE_EGY_MIN_KWP = 1
then      N_MAX_EGY = C_N_MAX_EGY_1
elseif    STATE_EGY_MIN_KWP = 2                AND
          CTR_KM_CAN <= C_CTR_KM_CAN_EGY_2_MAX
then      N_MAX_EGY = C_N_MAX_EGY_2
elseif    STATE_EGY_MIN_KWP = 3
then      N_MAX_EGY = C_N_MAX_EGY_3
else      N_MAX_EGY = 1FE0H
ENDIF
```

a8) Engine speed limit in order to avoid a maximum torque in powertrain due to jump start

```
IF        LV_CS = 1
then      N_MAX_DROF = IP_N_MAX_DROF( VS )
else      N_MAX_DROF = 1FE0H
```

ENDIF

a9) Engine speed limit in order to avoid overspeed in oil pressure limb home mode

N_MAX _LIH_PRS_OIL = N_LIH_PRS_OIL + C_N_MAX_PRS_OIL_ADD

a10) Engine speed limitation for automatic transmission in case of TCU - EGS limb home (CAN error)

```
IF        ( LV_ETCU_DISABLE_CAN == 1          AND
            LV_ST_END == 1                    AND
            T_AST > C_T_AST_ETCU_CAN_VLD )          OR
            STATE_ST_TQ_LIM_GS == 1
THEN        N_MAX_LIH_EGS = IP_N_MAX_LIH_EGS (VS)
ELSE        N_MAX_LIH_EGS = N_MAX_LIH_EGS + C_N_MAX_LIH_EGS_LGRD
endif
```

b) Selection of the currently active engine speed limit N_MAX_THD:

The lowest active engine speed limitation is selected as input for the controller:

```
N_MAX_THD      =      max(  C_N_MAX_THD_MIN,
                            min(  N_MAX_MT,
                                  N_MAX_AMT,
                                  N_MAX_AT,
                                  N_MAX_LIH_ETC,
                                  N_MAX_LIH_MTC,
                                  N_MAX_VS_MAX,
                                  N_MAX_LIH_STATE_CMB_S,
                                  N_MAX_EGY,
                                  N_MAX_DROF,
                                  N_MAX_LIH_PRS_OIL,
                                  N_MAX_LIH_EGS )     )
```

c1) Currently active engine speed limit (BN) depending on TCO N_MAX_THD_1(orange area):

N_MAX_THD_1 = IP_N_MAX_1_TCO

c2) Currently active engine speed limit (BN) depending on TCO N_MAX_THD_2 (red area):

N_MAX_THD_2 = IP_N_MAX_2_TCO

d) Selection of the currently active engine speed limit due to limp home N_MAX_THD_LIH :

The lowest active engine speed limitation caused by limp home is used for determination of the level of limitation (STATE_N_MAX_THD_LIH):

```
N_MAX_THD_LIH          = min( N_MAX_LIH_ETC,
                              N_MAX_LIH_MTC,
                              N_MAX_LIH_STATE_CMB_S,
                              N_MAX_EGY,
                              N_MAX_LIH_PRS_OIL)
```

```
If      N_MAX_THD_LIH < C_N_MAX_THD_LIH_L
Then    STATE_N_MAX_THD_LIH = 2              // major limitation of N due to LIH
Else    If      N_MAX_THD_LIH < C_N_MAX_THD_LIH_H
        Then    STATE_N_MAX_THD_LIH = 1      // minor limitation of N due to LIH
        Else    STATE_N_MAX_THD_LIH = 0      // no limitation of N due to LIH
        Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_DRI_RVL	1	0...1FE0H	0...8160	1	[rpm]
engine speed limit AT-vehicle in reverse gear					
C_N_MAX_MTC_LIH	1	0...1FE0H	0...8160	1	[rpm]
Engine speed limitation in case of fault detected by throttle monitoring or ETC safety concept					
C_N_MAX_THD_MIN	1	0...1FE0H	0...8160	1	[rpm]
Minimum allowed engine speed threshold					
C_T_GEAR_VS_MAX_AT	1	0...FFH	0...25.5	0.1	[s]
Time for calculating N_MAX_VS_MAX after a gear shift with AT.					
C_T_GEAR_VS_MAX_MT	1	0...FFH	0...25.5	0.1	[s]
Time for calculating N_MAX_VS_MAX after a gear shift with MT.					
C_T_MAX_N_MAX_H_AT	1	0...FFH	0...25.5	0.1	[s]
maximum time after exceeding the VS threshold, AT vehicle					
C_T_MAX_N_MAX_H_MT	1	0...FFH	0...25.5	0.1	[s]
maximum time after exceeding the VS threshold, MT vehicle					
C_VS_MAX_HYS	1	0...FFH	0...255	1	[km/h]
Hysteresis of vehicle speed limitation					
C_VS_THD_N_MAX_H_AT	1	0...FFH	0...255	1	[km/h]
VS threshold for increased engine speed limit, AT vehicle					
C_VS_THD_N_MAX_H_MT	1	0...FFH	0...255	1	[km/h]
VS threshold for increased engine speed limit, MT vehicle					
ID_N_DEC_N_MAX_H_AT	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
stepwidth to basic engine speed limit, AT vehicle					
ID_N_DEC_N_MAX_H_MT	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
stepwidth to basic engine speed limit, MT vehicle					
ID_N_MAX_AT	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
basic engine speed limit AT-vehicle					
ID_N_MAX_H_AT	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
increased engine speed limit AT-vehicle					
ID_N_MAX_H_MT	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
increased engine speed limit MT-vehicle					
ID_N_MAX_MT	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
basic engine speed limit MT-vehicle					
ID_T_N_MAX_H_AT	9	0...FFFFH	0...6553.5	0.1	[s]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
time for increased engine speed limit, AT vehicle					
ID_T_N_MAX_H_MT	9	0...FFFFH	0...6553.5	0.1	[s]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
time for increased engine speed limit, MT vehicle					
IP_N_MAX_1_TCO	4	0...1FE0H	0...8160	1	[rpm]
LDP_TCO_N_MAX_1	4	0...C8H	-40...160	1	[°C]
Engine speed limit (orange area) depending on TCO - BN2000					
IP_N_MAX_2_TCO	4	0...1FE0H	0...8160	1	[rpm]
LDP_TCO_N_MAX_2	4	0...C8H	-40...160	1	[°C]
Engine speed limit (orange area) depending on TCO - BN2000					
IP_N_MAX_MTC_CUR_OFF__PV_AV	4	0...1FE0H	0...8160	1	[rpm]
LDP_PV_AV__N_MAX_MTC_CUR_OFF	4	0...FFH	0...99.60937	0.390625	[%]
Engine speed limit for limp home modus due to MTC error					
C_N_MAX_EGY_1	1	0...1FE0H	0...8160	1	[rpm]
Engine speed limit in case of production energy spare mode					
C_N_MAX_EGY_2	1	0...1FE0H	0...8160	1	[rpm]
Engine speed limit in case of transportation energy spare mode					

general specification

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_EGY_3	1	0...1FE0H	0...8160	1	[rpm]
Engine speed limit in case of garage energy spare mode					
IP_N_MAX_DROF__VS	8	0...1FE0H	0...8160	1	[rpm]
LDP_VS_N_MAX_DROF	8	0...FFH	0...255	1	[km/h]
Engine speed limit to avoid max torque at jump start					
C_N_MAX_PRS_OIL_ADD	1	0...1FE0H	0...8160	1	[rpm]
offset for engine speed limit for oil pressure limb home mode					
ID_N_MAX_AT_MAN	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
engine speed limit AT vehicle in 'M' (manual) mode					
ID_N_MAX_H_AT_MAN	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
increased engine speed limit AT vehicle in 'M' (manual) mode					
ID_T_N_MAX_H_AT_MAN	9	0...FFFFH	0...6553.5	0.1	[s]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
time for increased engine speed limit, AT vehicle in 'M' (manual) mode					
ID_N_DEC_N_MAX_H_AT_MAN	9	0...1FE0H	0...8160	1	[rpm]
LDPM_GEAR	9	0...FFH	0...255	1	[-]
stepwidth to basic engine speed limit, AT vehicle in 'M' (manual) mode					
C_T_MAX_N_MAX_H_AT_MAN	1	0...FFH	0...25.5	0.1	[s]
maximum time after exceeding the VS threshold, AT vehicle in 'M' (manual) mode					
C_VS_THD_N_MAX_H_AT_MAN	1	0...FFH	0...255	1	[km/h]
VS threshold for increased engine speed limit, AT vehicle in 'M' (manual) mode					
LC_VAR_ETCU_SPT_SWI	1	0...1H	0...1	1	[-]
LC indicating version of AT or TCT transmission (0 = ETCU without sport switch, 1 = ETCU with sport switch)					
C_N_MAX_THD_LIH_L	1	0...1FE0H	0...8160	1	[rpm]
Lower engine speed limit threshold (classifies major N_MAX reduction due to LIH)					
C_N_MAX_THD_LIH_H	1	0...1FE0H	0...8160	1	[rpm]
Higher engine speed limit threshold (classifies minor N_MAX reduction due to LIH)					
C_N_MAX_LIH_EGS_LGRD	1	0...1FE0H	0...8160	1	[rpm]
Limitation gradient for n_max during AT gear shift in case of CAN error to EGS					
IP_N_MAX_LIH_EGS	8	0...1FE0H	0...8160	1	[rpm]
LDP_VS__N_MAX	8	0...FFH	0...255	1	[km/h]
Engine speed limit for AT in case of CAN error to EGS					
LC_N_MAX_VS_MAX_ACT	1	0...1H	0...1	1	[-]
Switch to activate vs_max limitation over N_MAX_VS_MAX					
C_CTR_KM_CAN_EGY_2_MAX	1	0...FFFFH	0...655350	10	[km]
Maximum vehicle mileage to activate torque/engine speed-limitation due to energy spare mode 2 = "transport modus"					
C_T_AST_ETCU_CAN_VLD	1	0...FFFFH	0...6553.5	0.1	[s]
Time after start limit, until ETCU_CAN is valid					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_N_REF_MAX	1	0...1FE0H	0...8160	1	[rpm]
Threshold for not active engine speed limitation, 8160rpm					

24.2 Engine Speed Limitation Controller

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_N_MAX	O/V	0...1H	0...1	1	-
Logical variable engine speed limitation active					
LV_N_MAX_REQ_FCUT	O/V	0...1H	0...1	1	-
Logical variable fuel cut-off of all cylinder requested for engine speed limitation					
TQI_N_MAX	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Indicated engine torque setpoint for N_MAX-limitation					
TQI_N_MAX_REF	V	0...7FFFH	0...1.02397E+3	0.03125	Nm
Reference indicated engine torque on which controller output is applied					
TQ_N_MAX_DIF	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Torque output of N_MAX-PI-Controller					
TQ_N_MAX_DIF_I	V	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Torque output of N_MAX-I-Controller					
TQ_N_MAX_DIF_P	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Torque output of VS_MAX-P-Controller					
TQ_N_MAX_INP_I	V	8000...7FFFH	-4...3.99987793	1.2207E-4	Nm
Torque input for N_MAX I-controller					
N_DIF_MAX_LIM	V	E020...1FE0H	-8.16E+3 ... 8.16E+3	1	rpm
Engine speed difference as an input for PI-controller					
N_GRD_FIL	V	80...7FH	-4.096E+3 ... 4.064E+3	32	rpm/s
Engine speed gradient pre-selection for low pass filter					
N_GRD_FIL_FIL	V	80...7FH	-4.096E+3 ... 4.064E+3	32	rpm/s
Low pass filtered engine speed gradient for predicted engine speed					
N_PRED_MAX_LIM	V	0...1FE0H	0...8.16E+3	1	rpm
Predicted engine speed					
TQI_N_MAX_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Intermidiate value 1 for indicated engine torque after considering PI-controller output					
TQI_N_MAX_2	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Intermidiate value 2 indicated engine torque after considering PI-controller output					

Input data:

GEAR	LV_AT	LV_VAR_BN	LV_VAR_TCT
N	N_32	N_GRD	N_MAX_THD
STATE_ETCU_PROG_INFO	TQI_AV	TQI_REQ_TRA	LV_INH_GP_SUP

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ADD_N_MAX_FCUT	1	0...1FE0H	0...8.16E+3	1	rpm
Additive threshold to maximum engine speed for activation fuel cut-off all cylinders					
C_CRLC_N_GRD_FIL_1	1	0...FFFFH	0...0.99998474	1.52588E-5	-
N_GRD_FIL low pass filter constant if LV_N_MAX = 0					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_N_GRD_FIL_2	1	0...FFFFH	0...0.99998474	1.52588E-5	-
N_GRD_FIL low pass filter constant if LV_N_MAX = 1					
C_N_GRD_HYS	1	80...7FH	-4.096E+3 ... 4.064E+3	32	rpm/s
Hysteresis for N_GRD Filtering					
C_TQI_MAX_PAS	1	0...7FFFH	0...1.02397E+3	0.03125	Nm
Passive value for TQI_N_MAX if engine-speed limitation is passive					
C_TQI_N_MAX_INC	1	0...7FFFH	0...1.02397E+3	0.03125	Nm
Maximum increment for the output when limiting increasing rate					
LC_VAR_ETCU_SPT_SWI	1	0...1H	0...1	1	-
LC indicating version of AT or TCT transmission (0 = ETCU without sport switch, 1 = ETCU with sport switch)					
ID_FAC_N_MAX_I_AT__GEAR	9	0...FFFFH	0...7.99987793	1.2207E-4	Nm/rpm
LDPM_GEAR	9	0...FFH	0...255	1	-
I-control parameter for N_MAX-controller automatic transmission and AMT					
ID_FAC_N_MAX_I_MT__GEAR	9	0...FFFFH	0...7.99987793	1.2207E-4	Nm/rpm
LDPM_GEAR	9	0...FFH	0...255	1	-
I-control parameter for N_MAX-controller manual transmission					
ID_FAC_N_MAX_P_AT__GEAR	9	0...FFH	0...7.96875	0.03125	Nm/rpm
LDPM_GEAR	9	0...FFH	0...255	1	-
P-control parameter for N_MAX-controller automatic transmission and AMT					
ID_FAC_N_MAX_P_MT__GEAR	9	0...FFH	0...7.96875	0.03125	Nm/rpm
LDPM_GEAR	9	0...FFH	0...255	1	-
P-control parameter for N_MAX-controller manual transmission					
ID_FAC_TQI_N_MAX_REF	9	0...FFH	0...0.99609375	0.00390625	-
LDPM_GEAR	9	0...FFH	0...255	1	-
Factor on TQI_N_MAX_REF value to init. I-controller					
ID_T_N_MAX_PRED_AT_MAN	9	0...FFH	0...7.96875	0.03125	s
LDPM_GEAR	9	0...FFH	0...255	1	-
Prediction time for predicted engine speed for AT in manual mode ´M´					
ID_T_N_MAX_PRED_AT__GEAR	9	0...FFH	0...7.96875	0.03125	s
LDPM_GEAR	9	0...FFH	0...255	1	-
Prediction time for predicted engine speed for automatic transmission and AMT					
ID_T_N_MAX_PRED_MT__GEAR	9	0...FFH	0...7.96875	0.03125	s
LDPM_GEAR	9	0...FFH	0...255	1	-
Prediction time for predicted engine speed for manual transmission					
IP_N_GRD_THD__N_32	8	0...FFH	-4.096E+3 ... 4.064E+3	32	rpm/s
LDP_N_32_N_GRD_THD	6	0...FFH	0...8.16E+3	32	rpm
Engine speed gradient threshold for filtering of basic engine speed gradient					

24.2.1 Function description

General information:

The objective of this function is to protect the engine against overspeed and therefore to control the engine speed. A PI controller is used to hold the engine speed at the engine speed limit setpoint, which derives from the module 'Engine speed limit coordination' from chapter 4. Engine overspeed mainly is controlled by reduction of engine torque. The output value of this module is TQI_N_MAX, which is an input in the torque coordination module. Different tasks are designed in this module:

prediction of engine speed to avoid great engine speed overshoots, especially in the first phase of controlling

activation of speed limit controller due to speed deviation between predicted engine speed and engine speed limit

PI controller

activation of fuel cut off for exceeding the engine speed limit for more than a calibratable difference

rate limiter for the output in increasing direction to avoid uncomfortable jumps in the torque demand

Application Condition

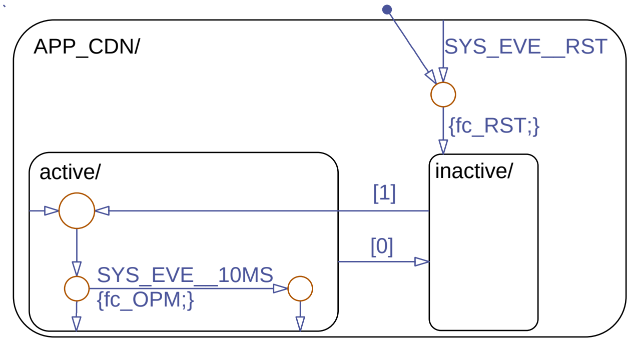


Figure 1 ENSL_M901E/ APP_CDN/ APPCND

Function Description

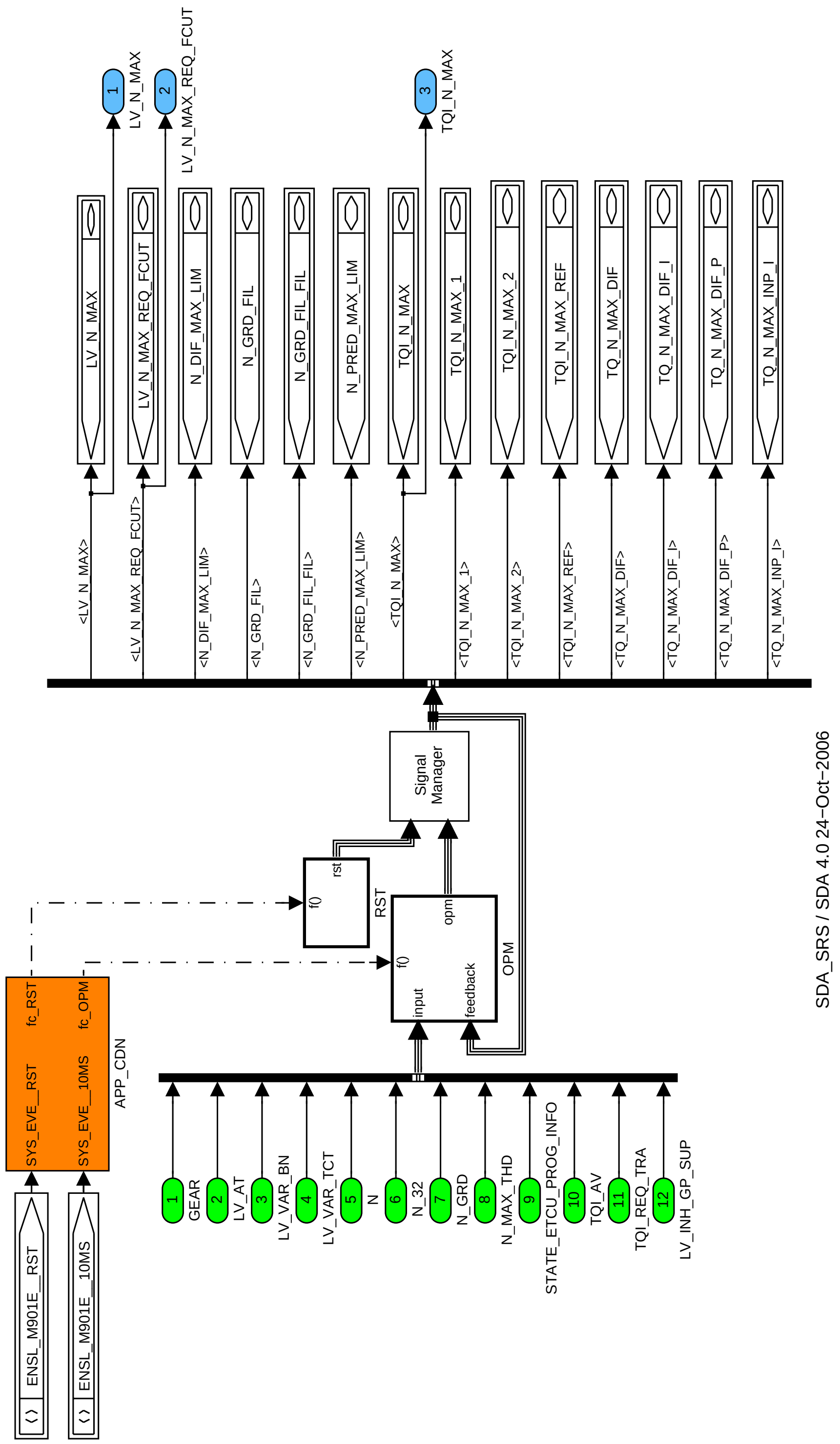


Figure 2 ENSL_M901E

24.2.1.1 Initialisation at system event reset

All output variables are initialised here.

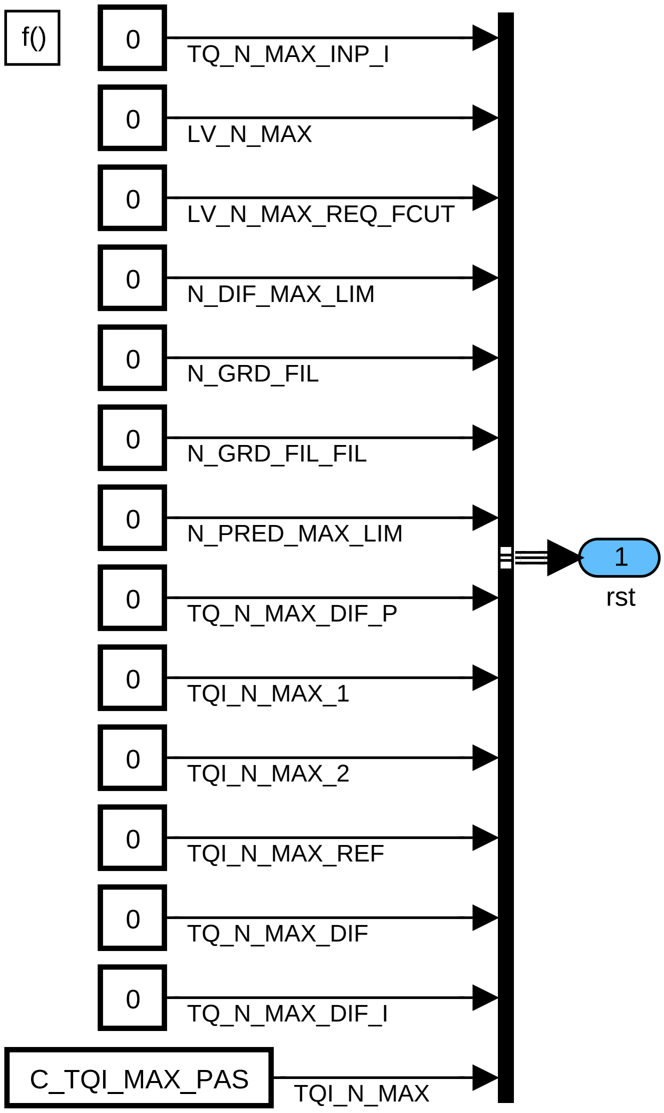


Figure 3 ENSL_M901E/ RST

24.2.1.2 Formula section

Calculation of predicted engine speed

The engine speed prediction shall activate the engine speed limitation controller before the real engine speed reaches the threshold N_MAX_THD. Therefore the complete controller has a PID behaviour at activation, to prevent high engine speed overshoot.

The prediction is deactivated, if the controller has stopped the engine speed increase. During the following period the PI controller can guarantee the engine speed limitation up to N_MAX_THD without predicting the engine speed.

Due to a bad resolution of N_GRD, the prediction is only calculated, if a clear high engine speed gradient is detected. This is possible by comparing the gradient N_GRD with an engine speed depending threshold IP_N_GRD_THD__N_32.

The prediction is stopped latest, if the real engine speed is no longer increasing (N_GRD =< 0).

general specification

To prevent sudden changes for the engine speed limit controller input at activation or deactivation, it is necessary to have a PT1 filter for the engine speed controller input. So a negative feed back of the controller output to the real engine speed is prevented.

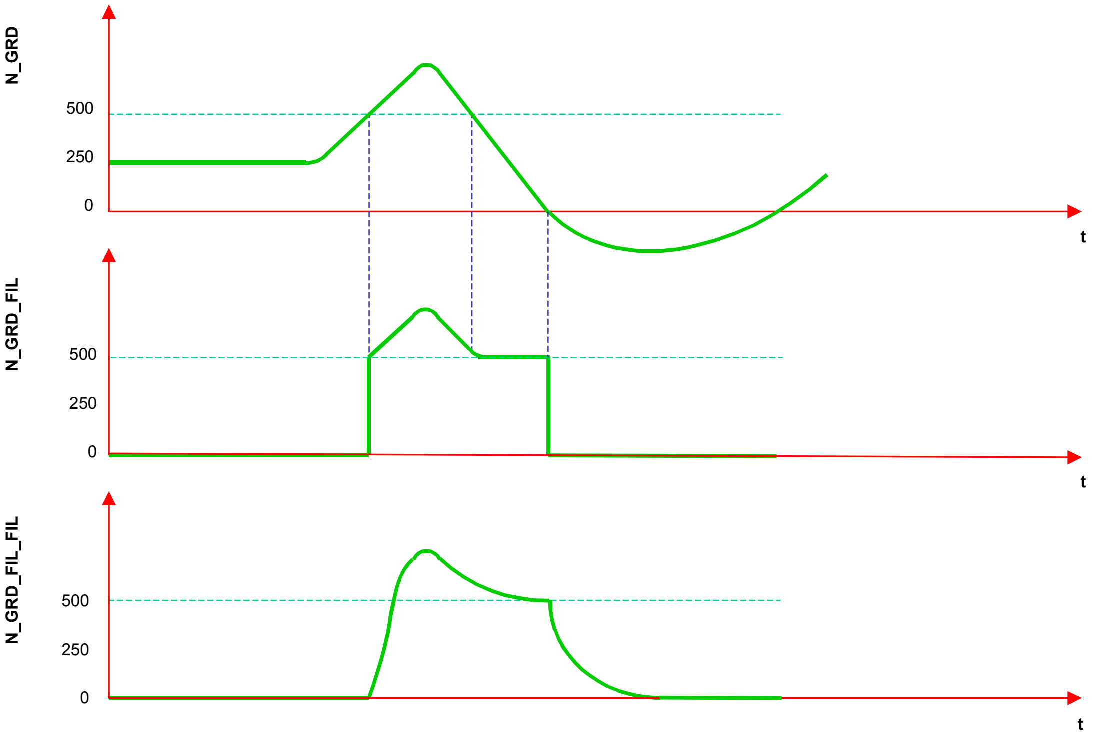


Figure 1: Filtering of engine speed gradient

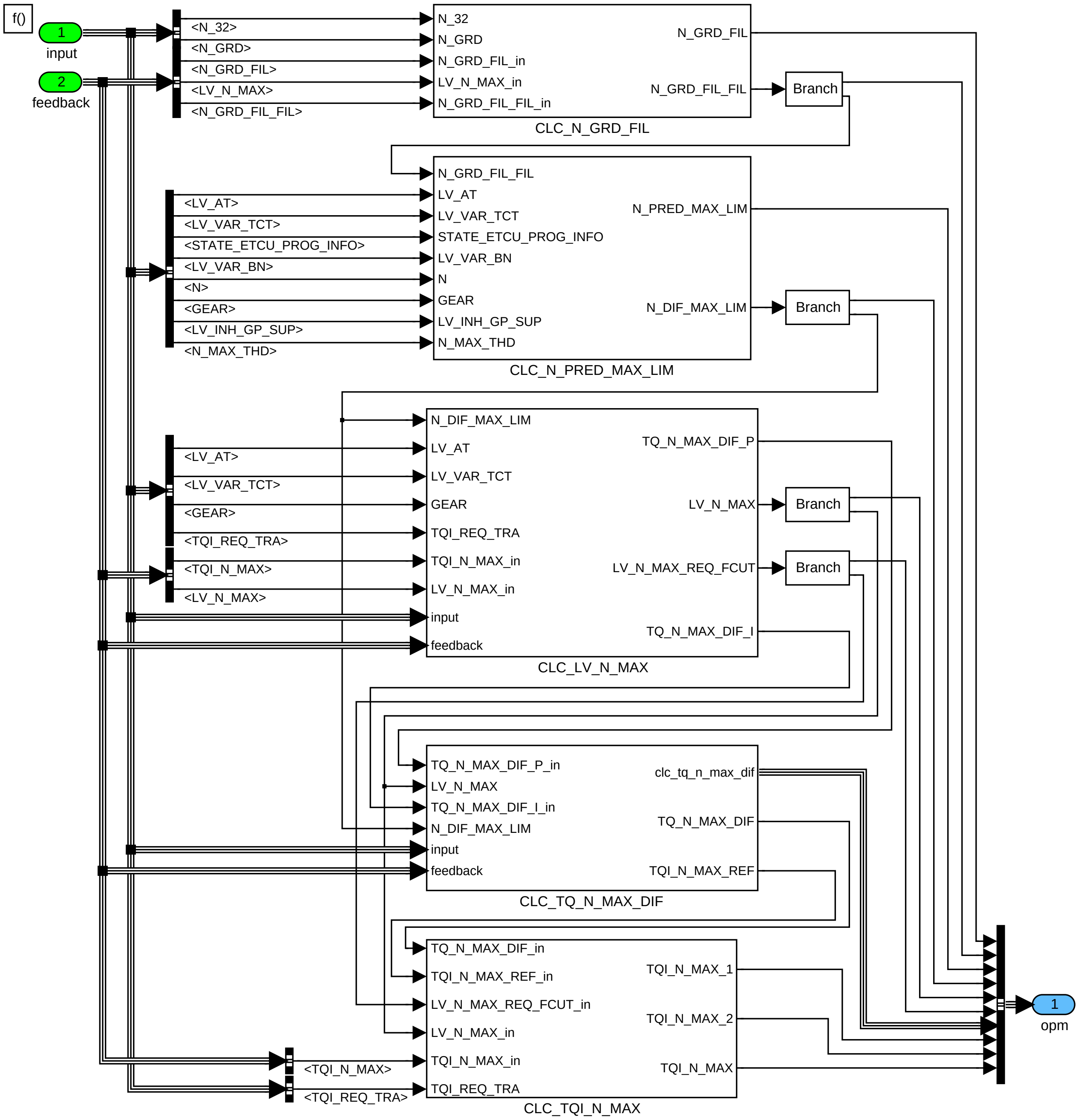


Figure 4 ENSL_M901E/ OPM

Calculation of engine speed gradient pre-selection and low pass filtered engine speed gradient for predicted engine speed

N_GRD_FIL and N_GRD_FIL_FIL are calculated.

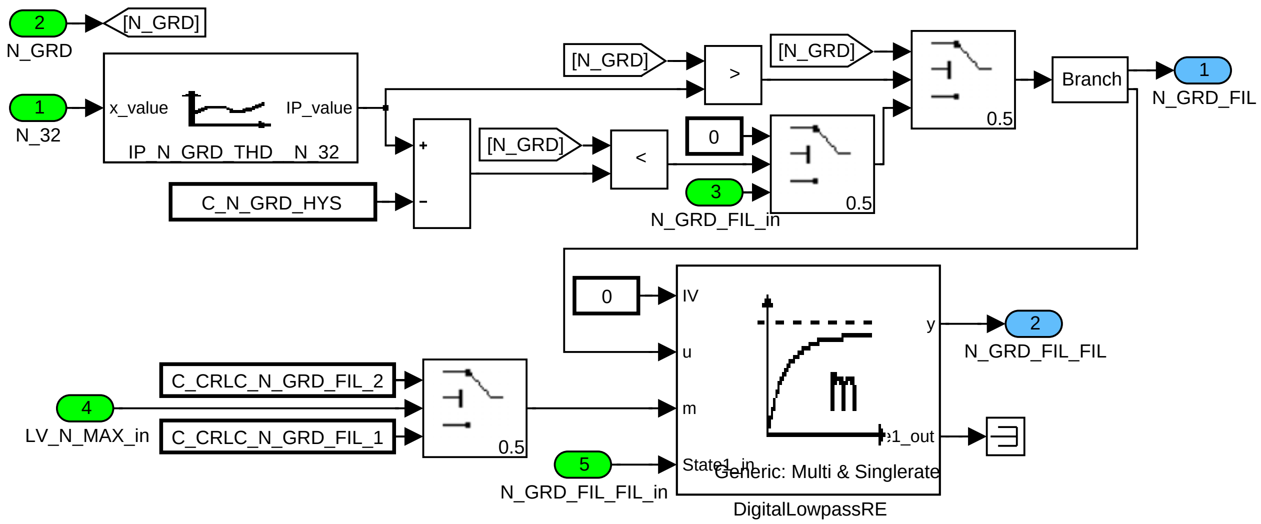


Figure 5 ENSL_M901E/ OPM/ CLC_N_GRD_FIL

Calculation of predicted engine speed and engine speed deviation from threshold

N_PRED_MAX_LIM and N_DIF_MAX_LIM are calculated

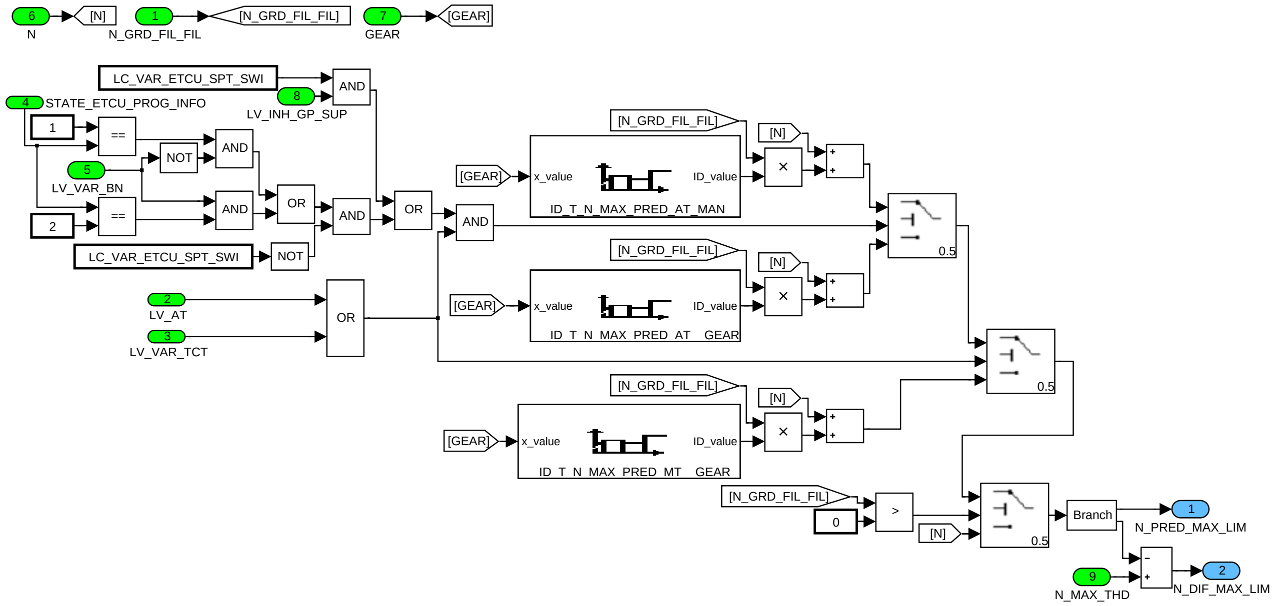


Figure 6 ENSL_M901E/ OPM/ CLC_N_PRED_MAX_LIM

Calculation of TQ_N_MAX_DIF_P, LV_N_MAX and LV_N_MAX_REQ_FCUT

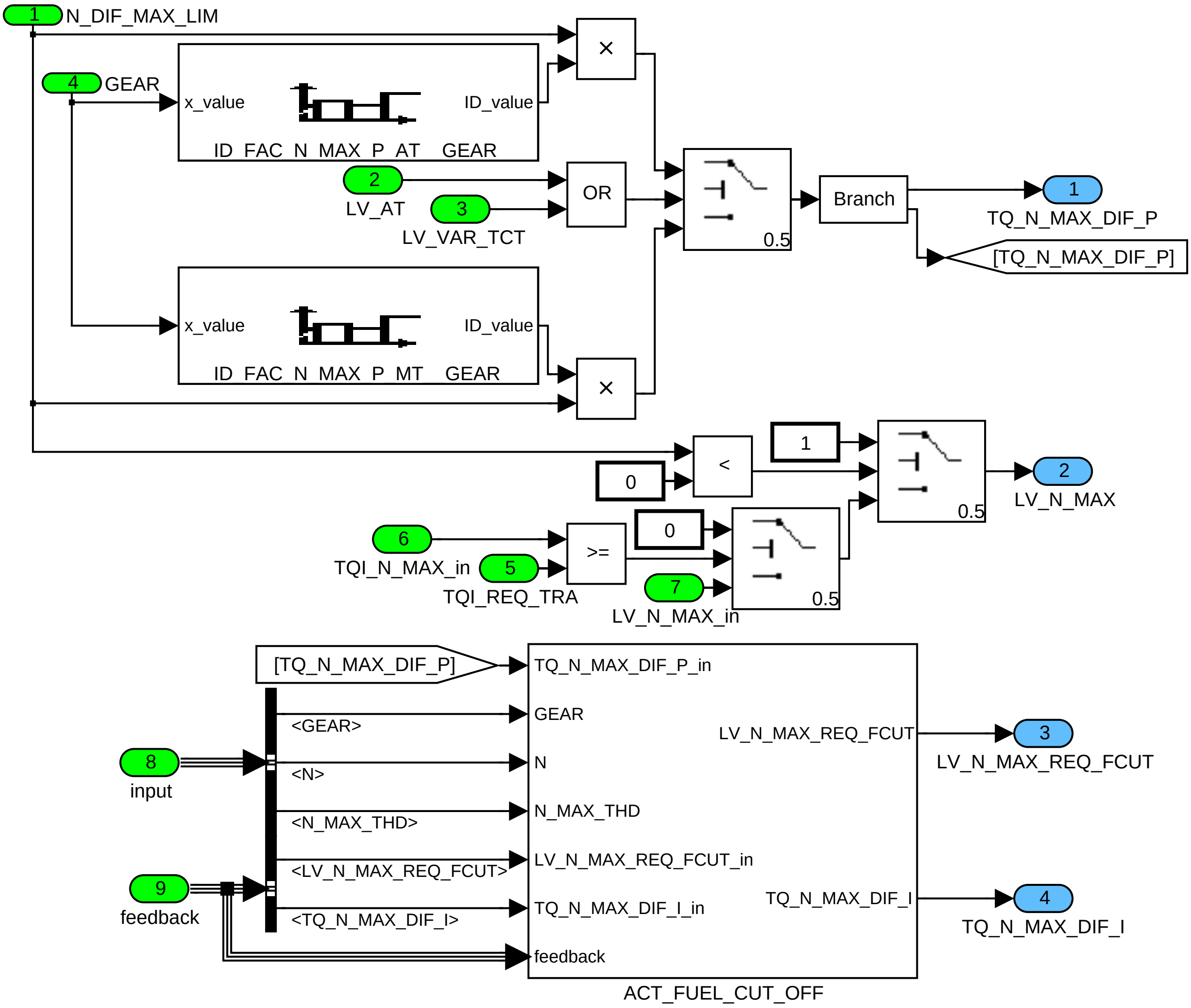


Figure 7 ENSL_M901E/ OPM/ CLC_LV_N_MAX

Calculation of logical variable for fuel cut off.

LV_N_MAX_REQ_FCUT is calculated.

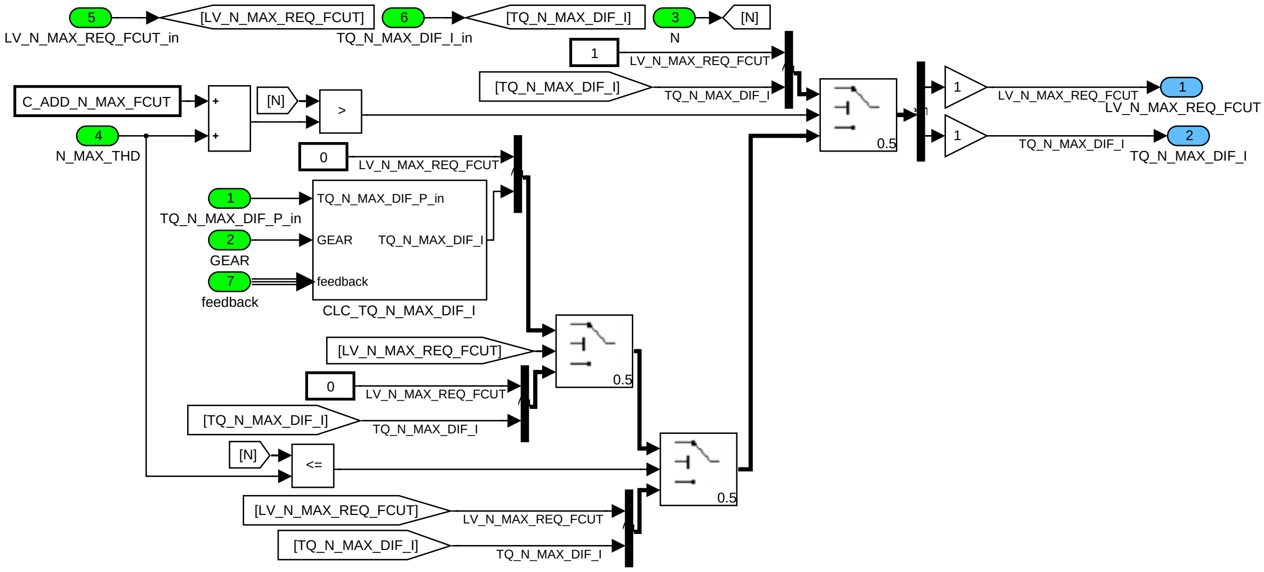


Figure 8 ENSL_M901E/ OPM/ CLC_LV_N_MAX/ ACT_FUEL_CUT_OFF

Initialisation of N MAX integrator.

Calculation of TQ_N_MAX_DIF_I

The integrator has to be initialized for the transition of LV_N_MAX from 0 to 1 and for the transition of LV_N_MAX_REQ_FCUT from 1 to 0. In the first case the integrator output is set to zero and in the second case the integrator output has to ensure that the TQI_N_MAX does not show a step (during LV_N_MAX_REQ_FCUT active the TQI_N_MAX is set to zero).

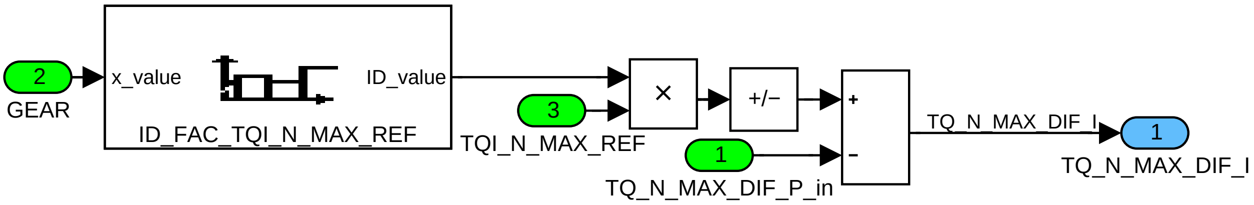


Figure 9 ENSL_M901E/ OPM/ CLC_LV_N_MAX/ ACT_FUEL_CUT_OFF/ CLC_TQ_N_MAX_DIF_I/ CLC_TQ_N_MAX_DIF_I

Determination n_max-limitation TQ_N_MAX_DIF using N_MAX_PI-controller

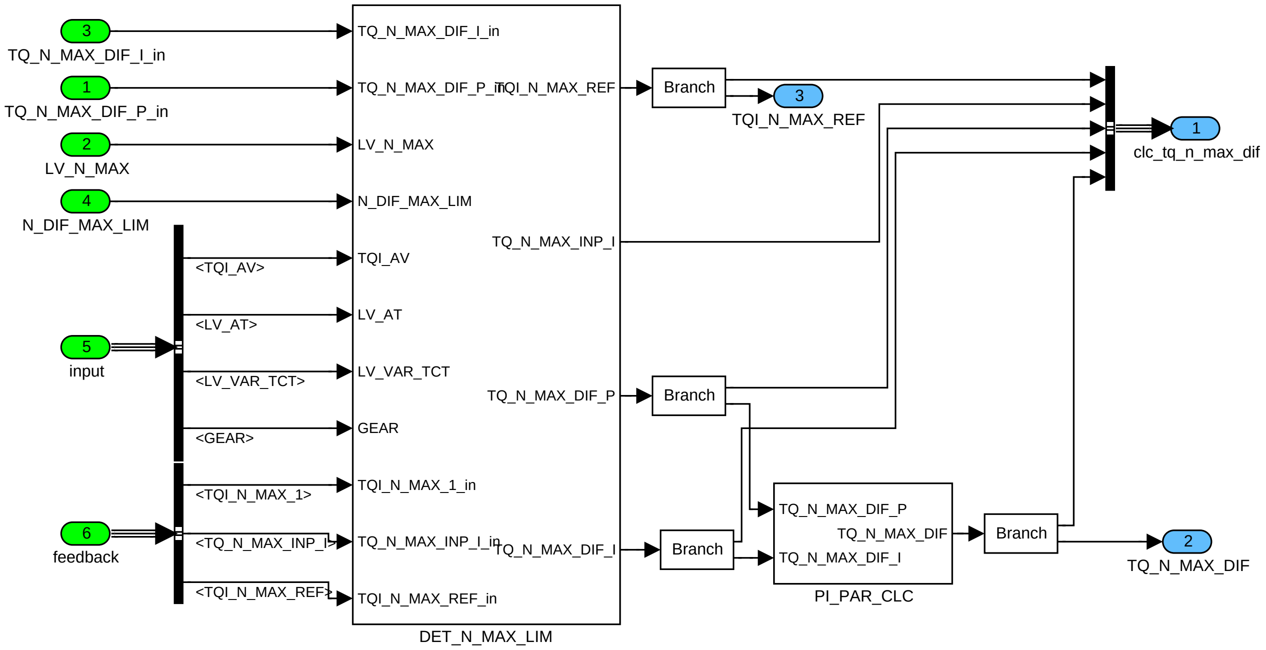


Figure 10 ENSL_M901E/ OPM/ CLC_TQ_N_MAX_DIF

Determination of TQ_N_MAX_REF, TQ_N_MAX_DIF_P, TQ_N_MAX_DIF_I and TQ_N_MAX_INP_I

The integrator input has to be set to zero, when the TQ_N_MAX_1 is smaller than zero and the calculated input TQ_N_MAX_INP_~~I~~_1 is smaller than or equal to zero. This avoids a steadily decrease of the integrator output which could cause problems when re entering the controller.

TQ_N_MAX_REF, TQ_N_MAX_DIF_P, TQ_N_MAX_DIF_I and TQ_N_MAX_INP_I are calculated here.

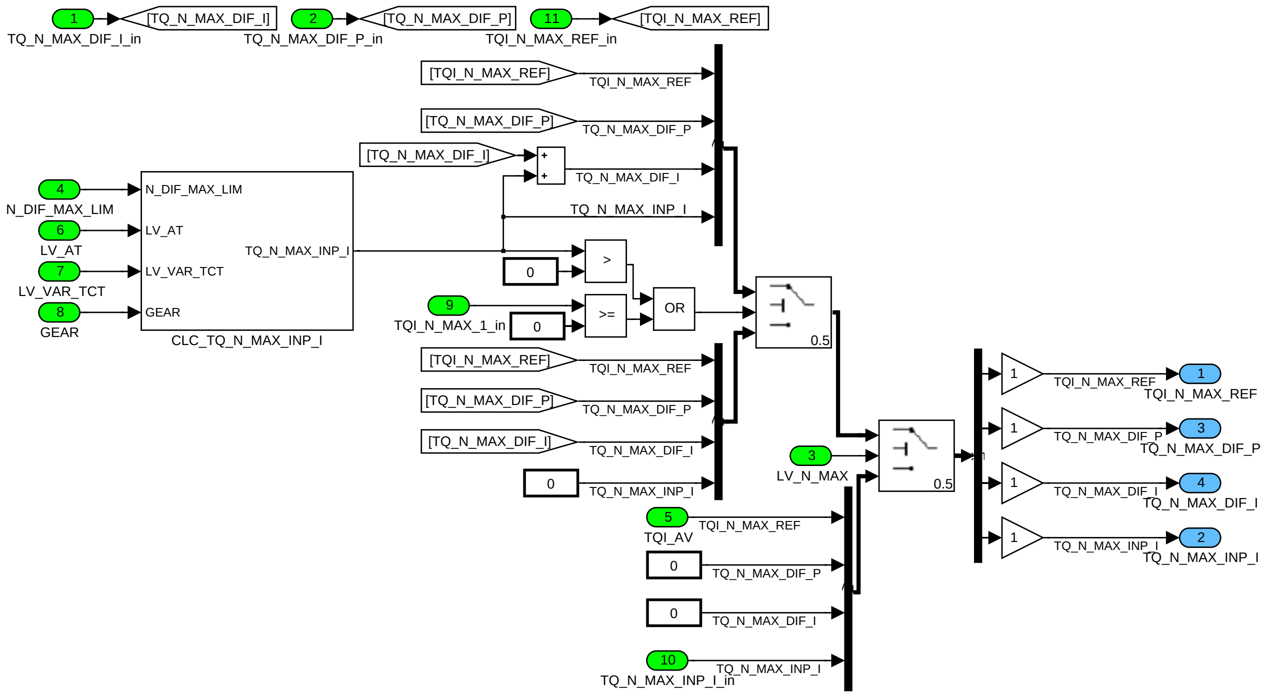


Figure 11 ENSL_M901E/ OPM/ CLC_TQ_N_MAX_DIF/ DET_N_MAX_LIM

Calculation of TQ_N_MAX_INP_I

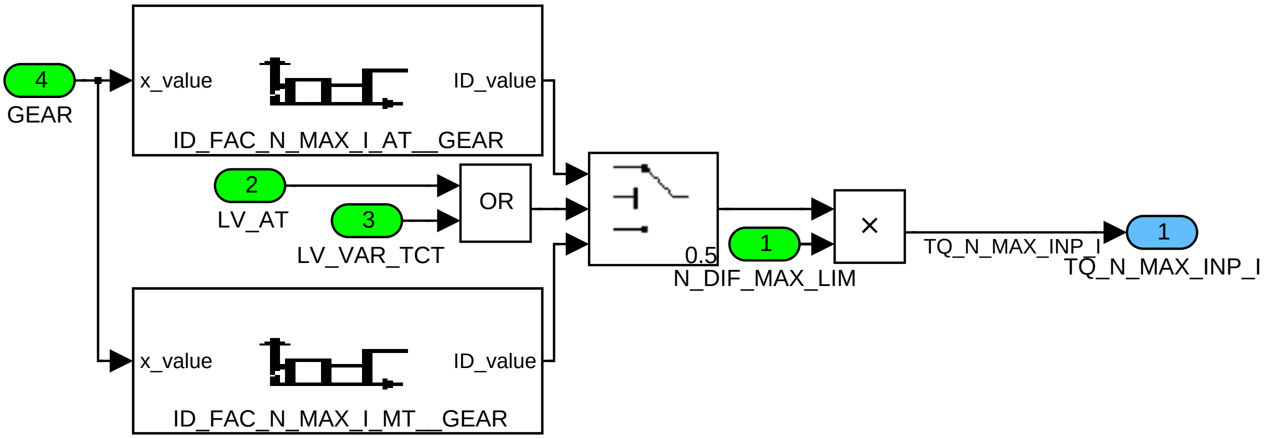


Figure 12 ENSL_M901E/ OPM/ CLC_TQ_N_MAX_DIF/ DET_N_MAX_LIM/ CLC_TQ_N_MAX_INP_I

Calculation of TQ_N_MAX_DIF

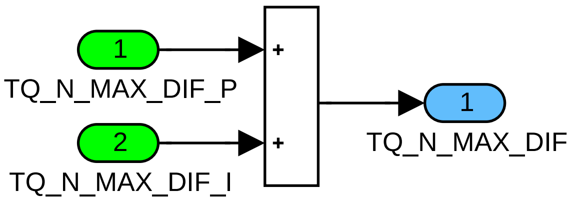


Figure 13 ENSL_M901E/ OPM/ CLC_TQ_N_MAX_DIF/ PI_PAR_CLC

Determination of indicated engine torque setpoint for N_MAX limitation

TQ_N_MAX_1, TQ_N_MAX_2 and TQI_N_MAX are calculated here.

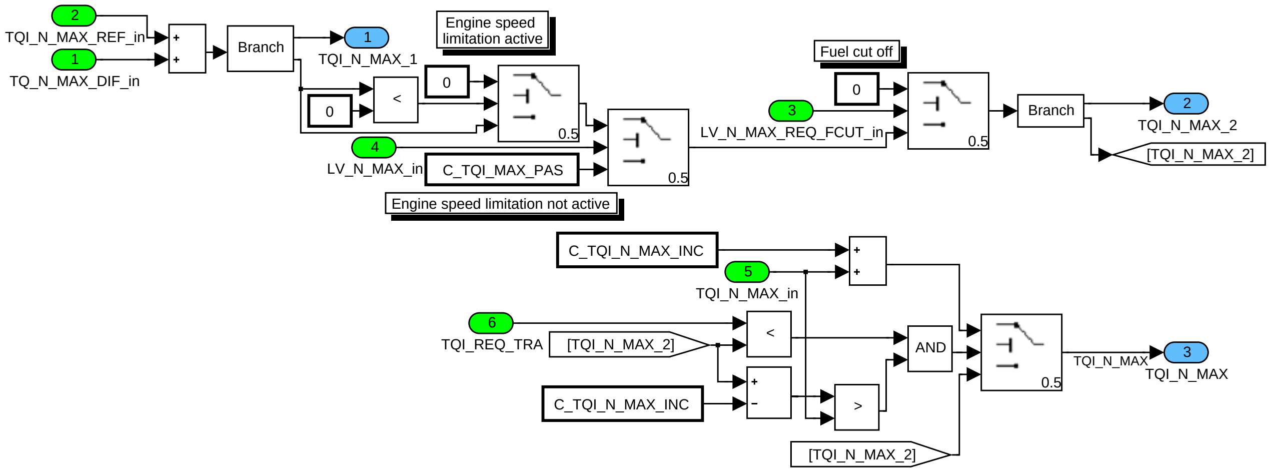


Figure 14 ENSL_M901E/ OPM/ CLC_TQI_N_MAX

24.3 Engine overspeed detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_MAX	V/O/S	1 ... 1FE0H	1 ... 8160	1	rpm
highest engine speed reached					
TRT_N_MAX	V/O/S	0...FFFFFFFFH	0...119304.6471	2,7778e-5	h
total running time in case of maximum engine speed reached					
CTR_N_MAX	V/O/S	0...FFH	0...255	1	-
event frequency counter					
CTR_KM_N_MAX	V/O/S	0....FFFFH	0....655350	10	km
km counter at highest event					
N_GRD_N_MAX	V/O/S	80...7FH	-4096...4064	32	rpm
Engine speed gradient at highest event					
VS_N_MAX	V/O/S	0...FFH	0...255	1	km/h
Vehicle speed at highest event					
PV_AV_N_MAX	V/O/S	0..FFH	0...99,6	0.39	%
Pedal value at highest event					
GEAR_EF_N_MAX	V/O/S	0...FFH	0...255	1	-
Gear ratio at highest event					
LV_CS_N_MAX	V/O/S	0...1H	0...1	1	-
Clutch switch detection at highest event					
T_SUM_N_MAX	V/O/S	0...FFFFH	0...6553,5	0.1	s
Time sum of all events maximum engine speed reached					
T_N_MAX	V/O/S	0...FFFFH	0...6553,5	0.1	s
Time sum at highest event					
NR_PAT_SCC_N_MAX	V/O/S	0...FFH	0...255	1	-
Selected index of fuel cut off pattern at highest event					

Input data:

N	LV_N_MAX	TRT	LV_VS_MAX
CTR_KM_CAN	N_GRD	VS	PV_AV
GEAR_EF	LV_CS	NR_PAT_SCC	

General information:

This function is for detection of an overreving event and with non-volatile storage of environmental datas if triggered.

Description:

An overreving event starts at exceeding the engine speed C_N_LIM_MIN and active engine speed limitation (LV_N_MAX = 1).

At each overreving event, the event frequency counter CTR_N_MAX is incremented by 1. The timer T_SUM_N_MAX is always incremented at every active event and remains at the maximum value if reached. The timer T_N_TMP recorded the duration of each event and is used for the T_N_MAX calculation (= the duration with the highest event)

The remaining environmental data are stored at the first event but will be only overritten if the new event has a higher engine speed then the old one. At the end of a driving cylce the datas are stored in NVMY.

Application conditions:

Initialisation: all data restored out of NVMY

Recurrence: segment synchronous

Activation: **at** **[** LV_N_MAX = 1 **and** LV_VS_MAX = 0 **]**

Deactivation: **at** LV_N_MAX = 0

Formula section:

determination of event-frequency-counter :

```
if      N > C_N_LIM_MIN
then    CTR_N_MAX = CTR_N_MAX + 1        ( only once per trigger )
else    CTR_N_MAX = CTR_N_MAX
endif
```

determination of total time :

```
if      N > C_N_LIM_MIN
then    starts timer T_SUM_N_MAX (always incremented, remains at FFFFH if reached)
else    stop timer T_SUM_N_MAX
endif
```

determination of individual time per event :

```
if      N > C_N_LIM_MIN
then    T_N_TMP ++
        N_MAX_TMP = MAX [ N ]
elseif          N_MAX_TMP > N_MAX        //stored in NVMY or stored at last event
    then        T_N_MAX = T_N_TMP        //update
    else        T_N_MAX = T_N_MAX        //unchanged
    endif
    T_N_TMP = 0          //reset
    N_MAX_TMP = 0        //reset
endif
```

determination of environmental data at first event :

```
if      N > C_N_LIM_MIN
then    N_MAX           = N
        TRT_N_MAX       = TRT
        CTR_KM_N_MAX    = CTR_KM_CAN
        N_GRD_N_MAX     = N_GRD
        VS_N_MAX        = VS
        PV_AV_N_MAX     = PV_AV
```

```
            GEAR_EF_N_MAX           = GEAR_EF
            LV_CS_N_MAX             = LV_CS
            NR_PAT_SCC_N_MAX        = NR_PAT_SCC
endif
```

determination of environmental data at the next events:

```
if          N > C_N_LIM_MIN    and
            N > N_MAX                          (stored in NVMY or stored at last event)
then        update of all environmental data:
            N_MAX                   = N
            TRT_N_MAX               = TRT
            CTR_KM_N_MAX            = CTR_KM_CAN
            N_GRD_N_MAX             = N_GRD
            VS_N_MAX                = VS
            PV_AV_N_MAX             = PV_AV
            GEAR_EF_N_MAX           = GEAR_EF
            LV_CS_N_MAX             = LV_CS
            NR_PAT_SCC_N_MAX        = NR_PAT_SCC
else        old values remains unchanged:
            N_MAX                   = N_MAX
            TRT_N_MAX               = TRT_N_MAX
            CTR_KM_N_MAX            = CTR_KM_N_MAX
            N_GRD_N_MAX             = N_GRD_N_MAX
            VS_N_MAX                = VS_N_MAX
            PV_AV_N_MAX             = PV_AV_N_MAX
            GEAR_EF_N_MAX           = GEAR_EF_N_MAX
            LV_CS_N_MAX             = LV_CS_N_MAX
            NR_PAT_SCC_N_MAX        = NR_PAT_SCC_N_MAX
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_LIM_MIN	1	0 ... 1FE0H	0 ... 8160	1	rpm
engine speed threshold for overreving event					

24.4 Customer adaptation module: ENSL

24.4.1 Outputs for BMW functions which are not defined as ENSL exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Mdi_nmax	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Maximum torque at engine speed limitation					
Nmax_var	V/O	8000...7FFFH	-32768...32767	1	[rpm]
Actual engine speed limit depending on TOIL - red area (BN2000)					
B_nlf	V/O	0...1H	0...1	1	[-]
Notluftfahren aktiv					
B_ska	V/O	0...1H	0...1	1	[-]
SKA aktiv, ausgelöst über Ebene 1, 2 oder 3					

Input data:

TQI_N_MAX	N_MAX_THD_2	LV_MTC_LIH_CUR_OFF	LV_TPS_MTC_N_LIM
LV_MTC_LIH_ACT	LV_MAF_BLS_DIAG	LV_ERR_ECU_CKS	LV_ERR_ECU_RAM
LV_N_LIM_REQ_MON			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: Mdi_nmax = 1000 Nm

B_nlf, B_ska, Nmax_var = 8000 rpm

Value at PWL: last calculated value: B_nlf, B_ska

Recurrence : 10 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

Mdi_nmax = TQI_N_MAX

Nmax_var = N_MAX_THD_2

```
IF      LV_N_LIM_REQ_MON == 1 OR  // Request from safty
        LV_TPS_MTC_N_LIM == 1  OR  // Request from ETC diagnosis
```

```
        LV_MTC_LIH_ACT== 1      OR  // Request from PVS diagnosis
        LV_MAF_BLS_DIAG==1      OR  // MAF error and brake active
        LV_ERR_ECU_CKS ==1      OR  // Request from Checksum calculation
        LV_ERR_ECU_RAM == 1         // Request from RAM check

THEN    B_ska = 1
ELSE    B_ska = 0
END

B_nlf = LV_MTC_LIH_CUR_OFF
```

24.4.2 Outputs for SV functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_MAX_LIH_STATE_CMB_S	V/O	0...1FE0H	0...8160	1	rpm
Engine speed limit for required stratified limp home mode					
N_LIH_PRS_OIL	V/O	0...1FE0H	0...8160	1	[rpm]
Required engine speed limit by customer for oil pressure limp home mode					

Input data:

Nmax_ba	Nkw_poel_notl		

FUNCTION DESCRIPTION:

General information:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Description:

Above a specific engine speed, the calculation of the stratified combustion mode is switched off. To enable a stratified limp home mode even when the stratified combustion mode calculation is switched off, the engine speed limitation N_MAX_LIH_STATE_CMB_S is introduced to reactivate stratified mode calculation. The BMW-variable Nmax_ba from the Layer is copied to the SV-variable N_MAX_LIH_STATE_CMB_S

Application conditions:

Initialisation: N_MAX_LIH_STATE_CMB_S = Nmax_ba
N_LIH_PRS_OIL = Nkw_poel_notl

Recurrence: N_MAX_LIH_STATE_CMB_S: 100 ms

N_LIH_PRS_OIL: 10ms

Activation: at every engine operating state

Deactivation: -

Formula section:

Remark: all formulas are valid in a **physical** meaning

N_MAX_LIH_STATE_CMB_S = Nmax_ba

N_LIH_PRS_OIL = Nkw_poel_notl

25 Engine start and stop

Table of Contents

Index

25.1 Engine off duration

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_ES	O/V	0...FFFFH	0...65535	1	[min]
Engine off duration time					
T_ES_AST	V	0...FFFFH	0...65535	1	[min]
Engine off duration time for plausibilsiaiton AST					
T_REL_CAN_ES	V/S	0...FFFFFFFFH	0...4294967295	1	[min]
Time counter value at the start of engine off phase					
T_REL_CAN_ST	V	0...FFFFFFFFH	0...4294967295	1	[min]
Time counter value at the end of engine off phase					
T_REL_CAN_AST	V	0...FFFFFFFFH	0...4294967295	1	[min]
Time counter value at the after start					
LV_T_ES_NOT_PLAUS	O/V	0...1H	0...1	1	[-]
T_ES not plausible					
LV_T_ES_NOT_PLAUS_AST	V	0...1H	0...1	1	[-]
T_ES not plausible AST					
LV_T_ES_PLAUS_AST_END	O/V	0...1H	0...1	1	[-]
Second plausi of T_ES after start finished (for T_ES-diagnosis)					
T_ES_CUS	O/V	0...FFFFH	0...65535	1	[min]
Engine off duration time for customer - value holds while engine is running					
LV_END_DIAG_T_ES_WIN	O/V	0...1H	0...1	1	[-]
End flag for T_ES Diagnosis					
T_ES_2	O/V	0...FFFFH	0...65535	1	[s]
Engine off duration time in second					
T_REL_CAN_ES_2	V/S	0...FFFFFFFFH	0...4294967295	1	[s]
Time counter value at the end of start of engine off phase					
T_REL_CAN_ST_2	V	0...FFFFFFFFH	0...4294967295	1	[s]
Time counter value at the end of engine off phase					

Input data:

T_REL_CAN	LV_ES	LV_IGK	LV_T_REL_CAN_REG
LV_VAR_BN	LV_ST_END	T_AST	TCO
TCO_STOP	T_REL_CAN_2		

FUNCTION DESCRIPTION:

The time T_ES describes the engine off duration time. For this the relative time counter of CAN Message INSTR2 is evaluated. This counter is incremented every minute by 1 Hex and started at zero again if FFFFH is reached. A destinction between CAN11h and BN2000 is necessarry because of different Hexlimits of T_REL_CAN_... .

At the transition into the Engine State LV_ES the last valid Value of T_REL_CAN is hand over to the non-volatile memory as T_REL_CAN_ES.

As soon as the condition LV_ES and LV_IGK = 1 is recognised the Value of T_REL_CAN is hand over to T_REL_CAN_ST and will be updated every minute until the State LV_ES is leaving.

As soon as T_REL_CAN_ST is present the Time T_ES is calculated and will be updated every minute until the State LV_ES is leaving too.

The time T_ES is compared with the difference of TCO_STOP and TCO. This is to check if time T_ES is possible at this temperature range.

The time T_ES_2 calculated in every second. This is needed for the calculation of restart factor of the mass fuel flow (fac_st_rest). Simular to T_ES, as soon as T_REL_CAN_ST_2 is present the Time T_ES_2 is calculated and will be updated every second until the State LV_ES is leaving too.

General information:

The engine off duration time will be used mainly in temperature critical functions like injection and dew point recognition

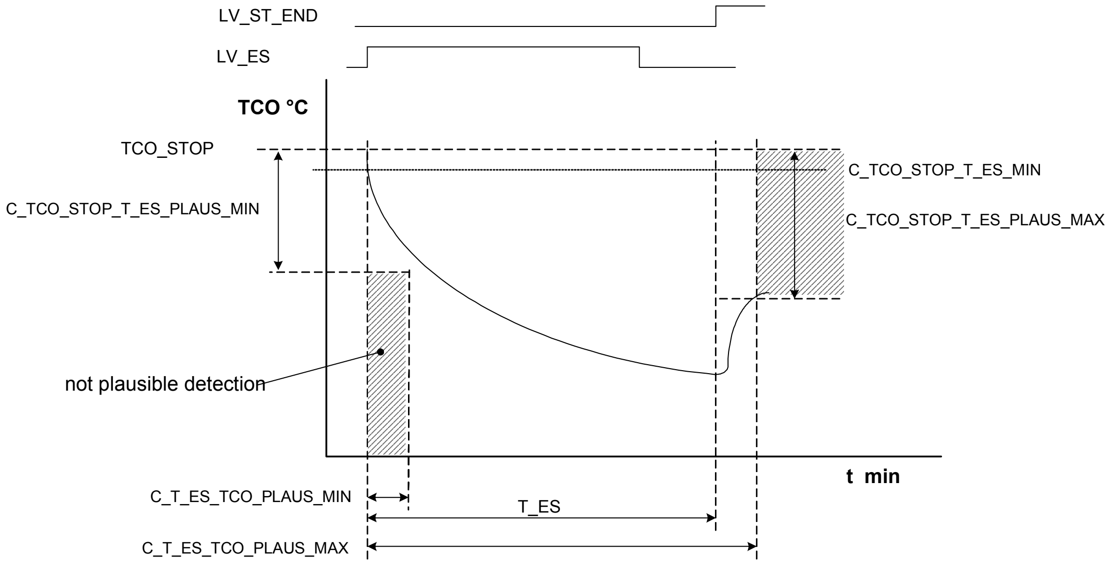
Application conditions:

Initialisation at reset: T_ES = 0
T_ES_2 = 0
T_REL_CAN_ES_2= 0 (out of NVMY)
T_REL_CAN_ST_2= 0
T_ES_AST = 0
T_REL_CAN_ES = 0 (out of NVMY)
T_REL_CAN_ST = 0
T_REL_CAN_AST = 0
LV_ST_END_T_ES = 0
T_ES_CUS = 0
LV_T_ES_PLAUS_AST_END = 0
LV_END_DIAG_T_ES_WIN = 0

Recurrence: 100ms

Activation: LV_ES =1

Signal flow diagram:



Formula section:

```
If      LV_T_REL_CAN_REG = 1                                   and
        LV_ST_END_T_ES = 1                                     and
        $LV\_ES_n = 1$ and $LV\_ES_{n-1} = 0$      (Transition to engine stop)
Then    T_REL_CAN_ES = T_REL_CAN                   (non volatile)
        LV_ST_END_T_ES = 0
        If LV_VAR_BN = 1
        Then T_REL_CAN_ES_2 = T_REL_CAN_2 (non volatile)
        Else T_REL_CAN_ES_2 = T_REL_CAN * 60 (non volatile)
Endif
```

//Remark: T_ES_CUS must be calculated after T_ES is calculated!

```
If      LV_ST_END = 0
Then    T_ES_CUS = T_ES
Else    if      T_AST > C_T_AST_T_ES
        then    if      LV_T_ES_NOT_PLAUS = 1 and
                        LV_T_ES_NOT_PLAUS_AST = 0
                Then    T_ES_CUS = T_ES_AST
                        LV_T_ES_NOT_PLAUS = 0
                Endif
                LV_T_ES_PLAUS_AST_END = 1
        Else    $T\_ES\_CUS_n = T\_ES\_CUS_{n-1}$         //T_ES_CUS does not change
                LV_T_ES_PLAUS_AST_END = 0
        Endif
Endif
```

Calculation of T_ES:

```
If(1)    LV_ST_END = 0
Then (1)
    If (2)    (LV_IGK = 1  or LV_VAR_BN = 1)  and
              LV_T_REL_CAN_REG = 1
    Then (2)      T_REL_CAN_ST = T_REL_CAN
        If (3)    ( T_REL_CAN_ST ≥ T_REL_CAN_ES )     or
                  LV_VAR_BN= 1
        Then (3)  T_ES =    T_REL_CAN_ST - T_REL_CAN_ES
        Else (3)  T_ES =    (T_REL_CAN_ST - T_REL_CAN_ES)  +  FFFFH
        Endif (3)
        If (4)    (T_ES < C_T_ES_TCO_PLAUS_MIN                          and
                  TCO_STOP - TCO > C_TCO_STOP_T_ES_PLAUS_MIN)
                                                     or
                  (T_ES > C_T_ES_TCO_PLAUS_MAX                          and
                  C_TCO_STOP_T_ES_MIN < TCO_STOP                        and
                  TCO_STOP - TCO < C_TCO_STOP_T_ES_PLAUS_MAX)
        Then(4)   LV_T_ES_NOT_PLAUS = 1
                  LV_END_DIAG_T_ES_WIN = 1
                  T_ES = C_T_ES_SUB
        Else (4)
            If(5)       T_ES >= C_T_ES_TCO_PLAUS_MIN                    and
                        TCO < C_TCO_STOP_T_ES_MIN –
                                       C_TCO_STOP_T_ES_PLAUS_MAX
          Then(5)       LV_T_ES_NOT_PLAUS = 0
                        LV_END_DIAG_T_ES_WIN = 1
          Endif(5)
        Endif (4)
     Endif (2)
Else(1)           T_ES = 0
          LV_ST_END_T_ES = 1
Endif (1)
```

Calculation of T_ES_2:

```
If(1)    LV_ST_END = 0
Then (1)
    If (2)    LV_VAR_BN = 1 and  LV_T_REL_CAN_REG = 1
    Then (2)        T_REL_CAN_ST_2 = T_REL_CAN_2
        If (3)      (T_REL_CAN_ST_2 ≥ T_REL_CAN_ES_2)
        Then (3)    T_ES_2 = T_REL_CAN_ST_2 - T_REL_CAN_ES_2
        Else (3)    T_ES_2= C_T_ES_SUB * 60
        Endif (3)
        If(4)LV_T_ES_NOT_PLAUS = 1
        Then(4)     T_ES_2 = C_T_ES_SUB * 60
        Endif (4)
      Else(2) T_ES_2 = T_ES *60
      Endif (2)
Else(1)             T_ES_2= 0
Endif (1)
```

Calculation of plausibility of T_ES for diagnosis:

```
If(1)    T_AST = C_T_AST_T_ES
Then (1)
    If (2a)   (LV_IGK = 1   or LV_VAR_BN = 1)  and
              LV_T_REL_CAN_REG = 1
    Then (2a)       T_REL_CAN_AST = T_REL_CAN
        If (3)      ( T_REL_CAN_AST ≥ T_REL_CAN_ES )    or
                    LV_VAR_BN = 1
        Then (3)    T_ES_AST =     T_REL_CAN_AST - T_REL_CAN_ES
        Else (3)    T_ES_AST =  (T_REL_CAN_AST - T_REL_CAN_ES)  +  FFFFH
        Endif (3)
        If (4)      (T_ES_AST < C_T_ES_TCO_PLAUS_MIN
                    and
                    TCO_STOP - TCO > C_TCO_STOP_T_ES_PLAUS_MIN)
                                                             or
                    (T_ES_AST > C_T_ES_TCO_PLAUS_MAX
                    and
                    C_TCO_STOP_T_ES_MIN < TCO_STOP                     and
```

```
                 TCO_STOP - TCO < C_TCO_STOP_T_ES_PLAUS_MAX)
        Then(4)  LV_T_ES_NOT_PLAUS_AST = 1
                 T_ES_AST = C_T_ES_SUB
        Else (4)
          If(5)  T_ES >= C_T_ES_TCO_PLAUS_MIN                     and
                 TCO < ( C_TCO_STOP_T_ES_MIN –
                                  C_TCO_STOP_T_ES_PLAUS_MAX)
          Then(5)      LV_T_ES_NOT_PLAUS_AST = 0
          Endif(5)
        Endif (4)
      Endif (2a)
  Endif (1)
```

Please note:

Please note the influence the definition of the 'engine off duration' has on the 'engine off duration diagnosis':

In chapter 'Engine off duration diagnosis' ERR_SYM_T_ES = SYM_2 (no signal from CAN) is not used any longer as it is already covered by the error-entries for CAN-Timeouts. Otherwise there would be a double-error-entry.

Therefore it is important that SYM_2 "no signal from CAN" will never occur. This is currently ensured by definition of LV_END_DIAG_T_ES_WIN, defined in chapter 'Engine off duration': LV_END_DIAG_T_ES_WIN is condition for the start of the diagnosis, but LV_END_DIAG_T_ES_WIN is only set if there is a signal from CAN.

As a consequence the definition of LV_END_DIAG_T_ES_WIN **must not** be changed. This solution should be kept this way in the running projects in series.

If the definition of LV_END_DIAG_T_ES_WIN will be changed then also ERR_SYM_T_ES = SYM_2 has to be removed from diagnosis.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_ES_TCO_PLAUS_MAX	1	0...FFFFH	0...65535	1	[min]
Threshold for maximum T_ES time					
C_T_ES_TCO_PLAUS_MIN	1	0...FFFFH	0...65535	1	[min]
Threshold for minimum T_ES time					
C_TCO_STOP_T_ES_PLAUS_MIN	1	0...FEH	0...190.5	0.75	[°C]
Threshold for minimum differences of cooling temperature					
C_TCO_STOP_T_ES_PLAUS_MAX	1	0...FEH	0...190.5	0.75	[°C]
Threshold for maximum differences of cooling temperature					
C_T_ES_SUB	1	0...FFFFH	0...65535	1	[min]
substitute value of T_ES					
C_T_AST_T_ES	1	0...FFFFH	0...6553.5	0.1	[s]
2nd plausibilisation of T_ES after start					
C_TCO_STOP_T_ES_MIN	1	0...FEH	-48...142.5	0.75	[°C]
Threshold for minimum cooling temperature at engine stop					

25.2 Auxiliary start functions

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>CYC_CAST</td><td>O/V</td><td>0...FFFFH</td><td>0...65535</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Cycle counter at start and after start</td></tr>
<tr><td>CTR_N_WIN_STALL</td><td>V</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Number of events inside the STALL rpm range</td></tr>
<tr><td>LV_AST</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Auxiliary function "After-start"</td></tr>
<tr><td>LV_REST</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Auxiliary function "Re-start"</td></tr>
<tr><td>LV_ST_ES</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Detection of engine start break off (passive / active)</td></tr>
<tr><td>LV_STALL</td><td>O/V/S</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Auxiliary function "Engine Stall detected"</td></tr>
<tr><td>LV_WUP_CAN</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">0 = catalyst hating/ warm up - 1 = engine at operating temperature</td></tr>
<tr><td>T_AST</td><td>O/V</td><td>0...FFFFH</td><td>0...6553.5</td><td>0.1</td><td>[s]</td></tr>
<tr><td colspan="6">Time after start</td></tr>
<tr><td>T_AST_DIAG</td><td>O/V</td><td>0...FFFFH</td><td>0...32767.5</td><td>0.5</td><td>[s]</td></tr>
<tr><td colspan="6">After start time for common sensor diagnosis</td></tr>
<tr><td>T_AST_REST</td><td>O/V/S</td><td>0...FFFFH</td><td>0...6553.5</td><td>0.1</td><td>[s]</td></tr>
<tr><td colspan="6">After start time for re-start</td></tr>
<tr><td>T_ST_ES</td><td>V</td><td>1...FFFFH</td><td>0.1...6553.5</td><td>0.1</td><td>[s]</td></tr>
<tr><td colspan="6">Time after transition from LV_ST to LV_ES</td></tr>
<tr><td>TCO_REST</td><td>V/S</td><td>0...FEH</td><td>-48...142.5</td><td>0.75</td><td>[°C]</td></tr>
<tr><td colspan="6">Coolant temperature, stored to detect a re-start</td></tr>
<tr><td>T_AST_SAE</td><td>O/V</td><td>0...FFFFH</td><td>0...65535</td><td>1</td><td>[s]</td></tr>
<tr><td colspan="6">PID1F Cumulated time since engine start</td></tr>
</table>

Input data:

LV_ES	LV_IGK	LV_IS	LV_PL
LV_ST	LV_ST_END	N_TOOTH_CUS	N_32
TCO	TI_CAST	LV_AST_END	LV_DC

25.2.1 Cycle counter CYC_CAST

FUNCTION DESCRIPTION:

General information:

- CYC_CAST shows the number of 120°crk cycles that have elapsed during the actual start phase (LV_ST=1). The start value is either zero or limited by the constant C_CYC_CAST_MAX_ST_ES (if start break off was detected).
- During actual after start phase (LV_AST from 0 to 1), the counter is reseted and restarts counting the number of cycles after start. When LV_AST is finished CYC_CAST remains at its last value.

Application conditions:

Activation: LV_ST = 1 or LV_AST = 1

Deactivation: LV_ST = 0 and LV_AST = 0

Initialization: CYC_CAST = 0 at ECU-reset

Recurrence: Every TDC

Formula section:

```
If      LV_AST  0 => 1
    then    CYC_CAST = 0
            else if     LV_ST or LV_AST is 1 and CYC_CAST < 65535
            then        $CYC\_CAST_n = CYC\_CAST_{n-1} + 1$
            else    $CYC\_CAST_n = CYC\_CAST_{n-1}$
    if      LV_ST_ES = 1
            then    if      CYC_CAST > C_CYC_CAST_MAX_ST_ES
            then            CYC_CAST = C_CYC_CAST_MAX_ST_ES
    endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CYC_CAST_MAX_ST_ES	1	0...FFFFH	0...65535	1	[-]
Max value of CYC_CAST for the beginning of incrementation if LV_ST_ES is set					

25.2.2 Start Break off LV_ST_ES

Description:

Activation:

For *engine stall during start phase*, i.e. transition from LV_ST to LV_ES before N_32 reaching IP_N_MAX_TOL_ST__TCO, the flag **LV_ST_ES** is set to 1, if LV_ST was active for a minimum number of cycles C_CYC_CAST_MIN_ST_ES. (Hint: At the same time the counter CYC_CAST will be frozen to its actual value).

Deactivation and application:

Case 1 (start after C_T_MIN_ECU_REST elapsed or powerlatch ended):

A timer T_ST_ES is started at the transition from LV_ST to LV_ES. If the timer elapses before a new start takes place (T_ST_ES ≥ C_T_MIN_ECU_REST), LV_ST_ES is reset and CYC_CAST is set to zero in the next start.

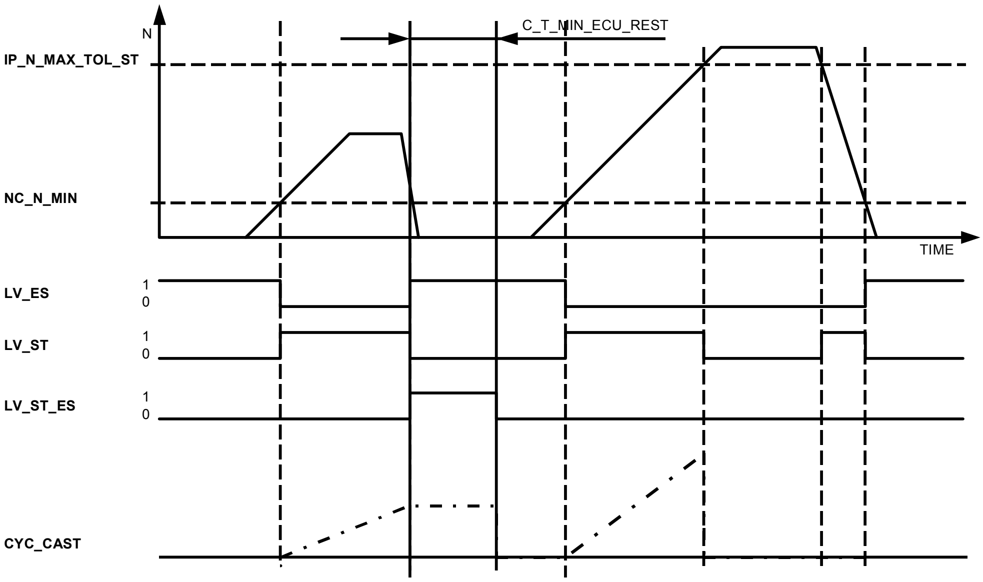
Application conditions:

Initialisation: T_ST_ES = 0 at ECU reset

Recurrence: 100 ms

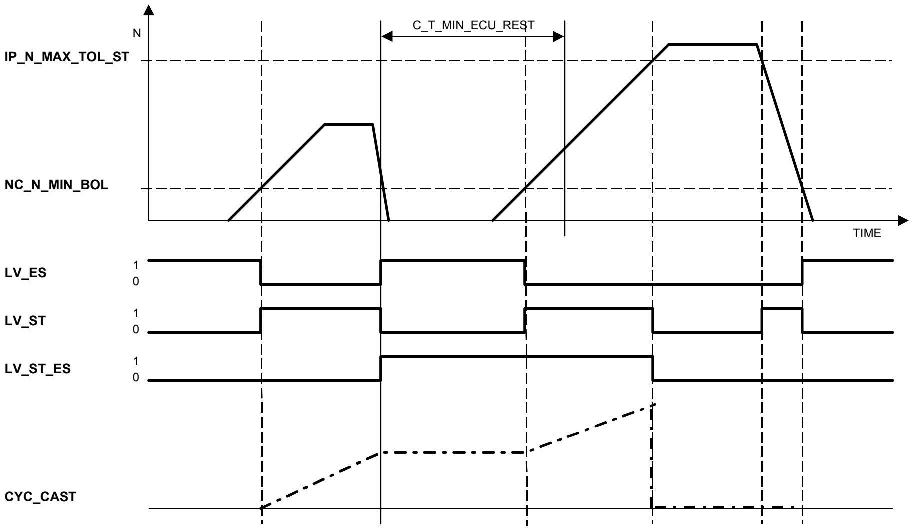
Formula section:

```
If      transition LV_ST → LV_ES
and     LV_ST_END = 0                       ; eng.op.state Start not left
and     CYC_CAST ≥ C_CYC_CAST_MIN_ST_ES
then    LV_ST_ES = 1
endif
If      LV_AST  0 => 1
and     LV_ST  1 => 0
or      T_ST_ES ≥ C_T_MIN_ECU_REST
then    LV_ST_ES = 0
endif

If      LV_ST_ES = 1        and     LV_ES=1
then    T_ST_ESn = T_ES_STn-1 + 1
    If      T_ST_ES >= C_T_MIN_ECU_REST
    then    LV_ST_ES = 0
            CYC_CAST = 0
            endif
else    T_ST_ES = 0
endif
```



Case 2 (start within C_T_MIN_ECU_REST):

If a new start takes place before the timer has elapsed (T_ST_ES < C_T_MIN_ECU_REST), LV_ST_ES remains set until the engine operation state start is exited to part load or idle, In this case the counter CYC_CAST starts with the previous frozen CYC_CAST-value *limited to a adjustable maximum value (see definition of CYC_CAST).*



Remark :

If C_T_MIN_ECU_REST > C_T_MAX_ECU (power latch time) and powerlatch-phase ended the value of CYC_CAST is reseted to zero for the next engine start after “start break off”-event.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MIN_ECU_REST	1	1...FFFFH	0.1...6553.5	0.1	[s]
Time threshold to detect new start after start-break off.					
C_CYC_CAST_MIN_ST_ES	1	0...FFFFH	0...65535	1	[-]
Min value of CYC_CAST for the detection of LV_ST_ES					

25.2.3 Restart function detection (LV_REST)

FUNCTION DESCRIPTION:

General information:

The re-start function is requested as an injection time correction during start.

- If the CAN clock timing is available LV_T_REL_CAN_REG=1 the re-start factor is calculated by TCO_STOP, T_AST_REST and T_ES (see “Fuel mass setpoint and pre-injection time calculation”).
- If the CAN clock timing is not available LV_T_REL_CAN_REG=0 the re-start factor is calculated as a substitue from IP_TI_ST_REST.

Application conditions:

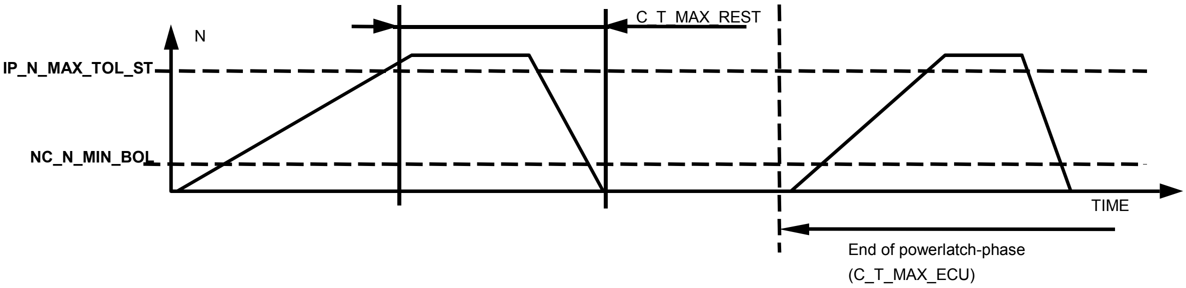
Activation:

After transition from engine start (LV_ST = 1) to engine stop (LV_ES = 1) LV_REST is checked to the end of the powerlatch-phase. As the engine was operated in part load or idle before engine stop, the next start is considered as a restart if following conditions are fullfilled:

If TCO_REST < FFH (no flash-error)
and TCO ≥ TCO_REST - C_TCO _DIF_MAX_REST

then LV_REST = 1
else LV_REST = 0

Therefore the restart-flag LV_REST is set to **1** before the next start, if the start happens before the end of the powerlatch-phase and the coolant temperature did not drop too much. The calculation of start injection time deactivation factor will start with CYC_CAST = 0.



Deactivation:

A counter T_AST is started at transition LV_ST → (LV_IS or LV_PL) .

If the engine operating state engine stopped (LV_ES) is detected before the counter has reached its maximum value C_T_MAX_REST, the related coolant temperature is stored :

(TCO → TCO_REST) in a non volatile memory.

Recurrence: segment

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MAX_REST	1	1...FFH	1...255	1	[s]
Time delay after switching on from LV_ST to detect LV_REST					
C_TCO_DIF_MAX_REST	1	0...50H	0...60	0.75	[°C]
Coolant temperature threshold to detect LV_REST					

25.2.4 Engine stall detection LV_STALL

General information:

Engine stall detection is used for injection and CRK diagnosis.

If engine is stopped with ignition-key-on there are two different possibilities:

- Engine is stalling
- CRK error is active

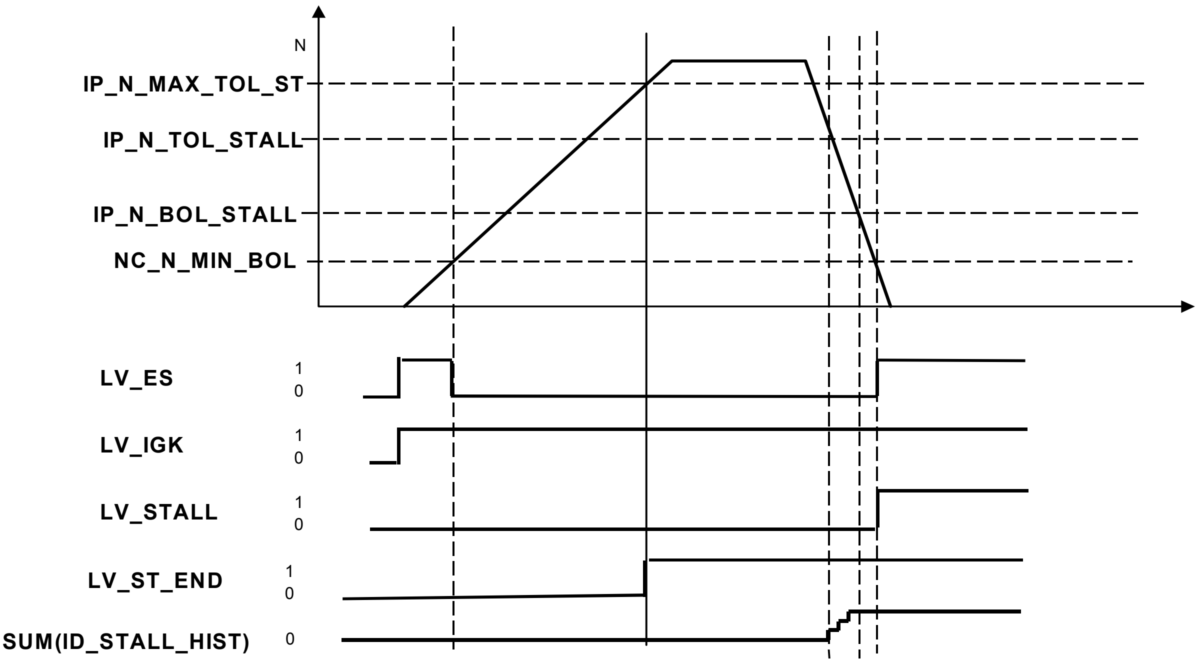
In order to detect a real engine stalling the engine speed value N_TOOTH_CUS is monitored. The reason for monitoring N_TOOTH_CUS instead of N is that N_TOOTH_CUS is calculated with a recurrency of 10ms, whereas N is calculated segment-syncronous. With engine-speed-values of < 1500 rpm the segment-time increases, the resulting update-rate of N is not precise enough for the different cases of engine-stall-detection.
If a CRK error occurs N_TOOTH_CUS is set to 0. If the engine is stalling N_TOOTH_CUS is passing a defined stalling-window.

Description:

ID_STALL_HIS is a buffer for the 32 last update rates (= 320ms History).

If N_TOOTH_CUS is between IP_N_TOL/BOL_STALL then the value in the buffer is set to 1 (=true-event).
After transition from engine running (LV_ST_END = 1) to engine stop (LV_ES = 1) with **turned on ignition** (LV_IGK = 1), it is checked if the sum of true-events in the buffer is bigger then a stalling - threshold.

Signal flow diagram:



Application conditions:

Initialisation: LV_STALL is restored out of NVMY

Recurrence: 10ms

Activation: N_TOOTH_CUS < C_N_MAX_CDN_STALL **and** LV_ST_END = 1

Formula section:

Reset of event-counter and ring-buffer after activation:

ID_STALL_HIS = 0

CTR_N_WIN_STALL = 0

Caclulation of LV_STALL:

```
if      IP_N_BOL_STALL < N_TOOTH_CUS < IP_N_TOL_STALL
then    ID_STALL_HIS = 1
        CTR_N_WIN_STALL ++          (stop ++ if FFH is reached)
else    ID_STALL_HIS = 0
        CTR_N_WIN_STALL = 0
```

```
endif

If      LV_ES  0 => 1
    and LV_ST_END = 1
    and LV_IGK = 1
    and SUM( ID_STALL_HIS ) > C_CTR_N_WIN_STALL
then    LV_STALL = 1
endif
```

Reset LV_STALL

```
if      LV_ST_END  0 => 1
then    LV_STALL = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_CDN_STALL	1	0...1FE0H	0...8160	1	[rpm]
Engine-speed threshold for deactivation of stalling detection					
C_CTR_N_WIN_STALL	1	0...FFH	0...255	1	[-]
Counter of N true events					
IP_N_BOL_STALL	9	0...1FE0H	0...8160	1	[rpm]
LPDM_TCO__N_STALL	9	0...FEH	-48...142.5	0.75	[°C]
Minimum engine speed for engine stalling detection					
IP_N_TOL_STALL	9	0...1FE0H	0...8160	1	[rpm]
LPDM_TCO__N_STALL	9	0...FEH	-48...142.5	0.75	[°C]
Maximum engine speed for engine stalling detection					

25.2.5 After-Start function : (LV_AST)

FUNCTION DESCRIPTION:

General information:

The after-start function serves to supply the extra injection quantity required after start by the cold walls of the combustion chamber.
The after-start function (**LV_AST**) is requested whenever the engine exits the operating state start (LV_ST) and is executed simultaneously with other engine operating states, except engine stopped (LV_ES) and start (LV_ST).
The value of T_AST when engine stopped (LV_ES=1) - after LV_ST_END = 1 has reached once - detected is transfered in T_AST_REST and safed in the non volatile memory at the end of the powerlatch - phase.

The variable T_AST indicates the time after leaving start-state. After reaching it's maximum value, T_AST is frozen. Reset occurs with every new start (Transition from LV_ST = 0 to 1).

T_AST_DIAG is calculated from T_AST to satisfy OBD requirements: recurrency must be 500ms for input signal aquisition/diagnosis.

Application conditions:

Activation:

The state **LV_AST** is activated as soon as the engine operating state start (LV_ST) is exited to go to the engine operating states idle speed (LV_IS) or part load (LV_PL).

Deactivation:

TI_CAST = 0 (refer to Chapter : ” Injection ”).

Recurrence:

Segment (LV_AST), 100ms (T_AST/T_AST_REST), 500ms (T_AST_DIAG)

Formula section:

```
If          LV_ES = 0  AND  LV_ST = 0
then
    if      T_AST < max. Value
    then    increment T_AST
    endif
endif
T_AST_DIAG = T_AST    (physical meaning)

If    LV_ES = 1 or LV_ST = 1 or LV_AST_END = 1
Then        LV_AST = 0
Else        LV_AST = 1
Endif

If    $LV\_ST\_END_{n} = 0$     and    $LV\_ST\_END_{n-1} = 1$          and
      $LV\_ES_{n} = 1$          and    $LV\_ES_{n-1} = 0$
Then        T_AST_REST = T_AST
Endif
```

25.2.6 State warm up

FUNCTION DESCRIPTION:

General information:

Calculation of state warm up. When end of warm up phase and catalyst heating is detected, LV_WUP_CAN is set to 1.

Application conditions:

Initialisation: 0

Recurrence: 100 ms

Formula section:

IF T_AST > IP_T_AST_SWI_WUP__CTR_KM_CAN **or**
TCO > C_TCO_SWI_WUP
THEN LV_WUP_CAN = 1 (catalyst heating ended indication, engine at operating temperature)
ELSE LV_WUP_CAN = 0

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_SWI_WUP	1	0...FEH	-48...142.5	0.75	[°C]
TCO condition for setting LV_WUP_CAN					
IP_T_AST_SWI_WUP__CTR_KM_CAN	4	0...FFFFH	0...6553.5	0.1	[s]
LPDM_CTR_KM_CAN	4	0...FFFEH	0...655340	10	[km]
Time condition for setting LV_WUP_CAN					

25.2.7 Cumulated time since engine start for OBD scan tool (PID1F)

FUNCTION DESCRIPTION:

General information:

Time after start is needed for OBD scan tool (PID1F)

Application conditions:

Initialisation: at reset or transition LV_DC = 1->0 set T_AST_SAE = 0 s

Recurrence: 1s

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

$T\_AST\_SAE_n = T\_AST\_SAE_{n-1} + 1s$

25.3 Starter Relay Control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_RLY_ST	O/V	0...1H	0...1	1	-
Starter relay output					

Input data:

LV_ACT_RLY_ST_EXT_ADJ	N_32	LV_RLY_ST_EXT_ADJ	TCO

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_N_MAX_TOL_STR	9	0...FFH	0...8.16E+3	32	rpm
LDPM_TCO_9	9	0...FEH	-48...142.5	0.75	°C
Top level end of starter relay control					

25.3.1 General information:

The function is defined to switch the starter relay output, dependent on the engine speed. An interface for external adjustment is offered. Through the action call is the ECU pin level "Starter Relay " set.

Application Condition

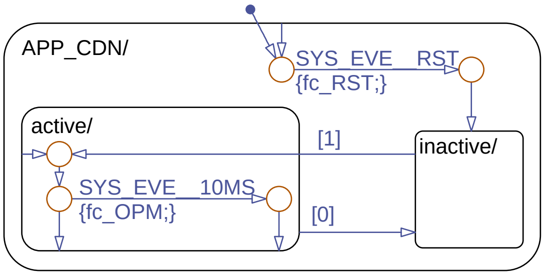


Figure 1 ENSS_M9015/ APP_CDN/ Chart

Function Description

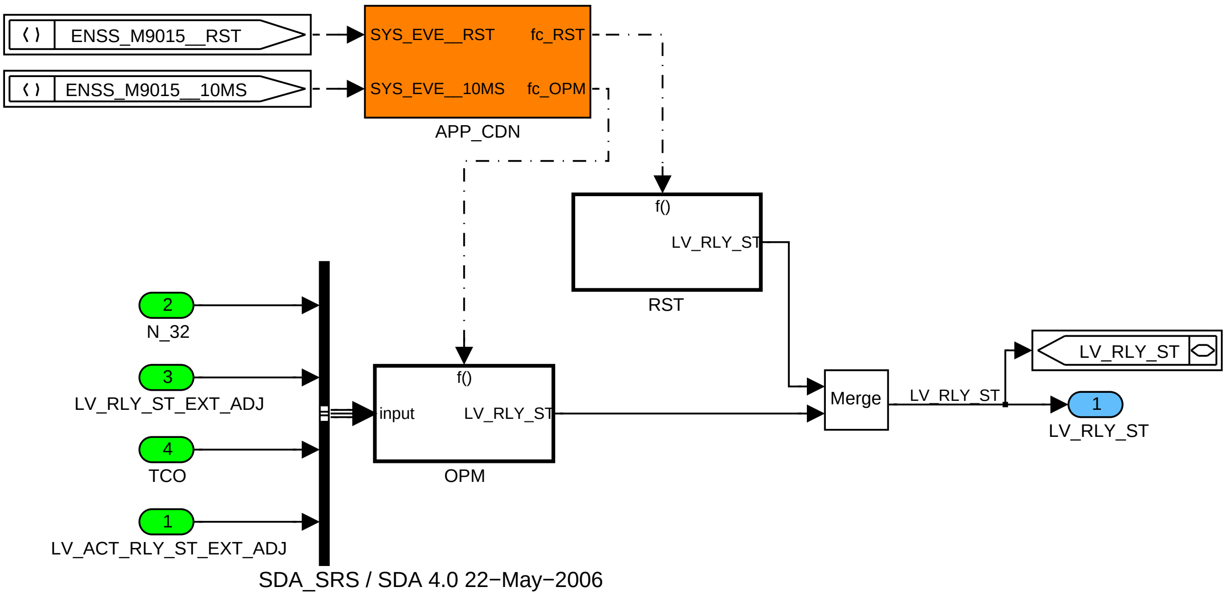


Figure 2 ENSS_M9015

25.3.1.1 Initialization at Reset

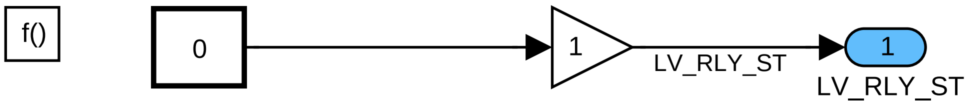


Figure 3 ENSS_M9015/ RST

25.3.1.2 Formula section

Activation of Starter Relay

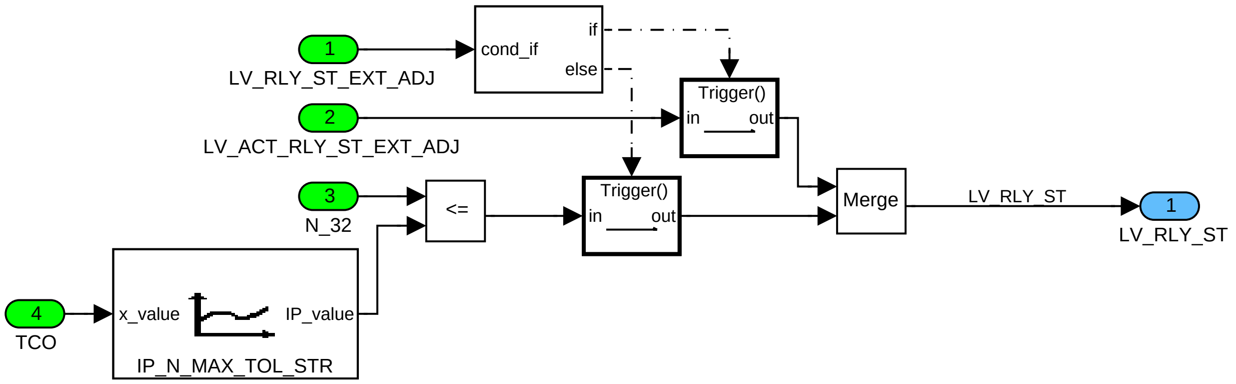


Figure 4 ENSS_M9015/ OPM/ CLC_LV_RLY_ST

25.4 Engine Off duration diagnosis

25.4.1 Diagnosis algorithm

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_T_ES	V/O	0...1H	0...1	1	[-]
Engine off duration error					
ERR_SYM_T_ES	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom of engine off duration error					
LV_CDN_DIAG_T_ES	V/O	0...1H	0...1	1	[-]
Diagnosis condition of engine off duration error					
LV_END_DIAG_T_ES	V/O	0...1H	0...1	1	[-]
End of diagnosis of engine off duration error					

Input data:

LV_IGK	LV_T_REL_CAN_REG	LV_T_ES_NOT_PLAUS	LV_CDN_VB_OBD2
LV_ERR_TCO	LV_END_DIAG_TCO_EL	LV_END_DIAG_TCO_GRD	LV_END_DIAG_TCO_STUCK
TCO	LV_ERR_ECU_NVMY	LV_T_ES_PLAUS_AST_END	LV_END_DIAG_T_ES_WIN

FUNCTION DESCRIPTION:

General information:

The engine off duration should be monitored concerning legislation requirements. Two different error symptoms are detectable.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                          (= SYM_0)
 │ │ └───> -                          (= SYM_1)
 │ └─────> no signal from CAN         (= SYM_2)
 └───────> T_ES not plausible to TCO  (= SYM_3)
```

Application conditions:

Initialisation: according ERRM initialization type "**STD_INI**"

Recurrence: 100 ms

Activation: LV_IGK = 1 **and** LV_END_DIAG_T_ES = 0

Formula section:

```
If      LV_ST_END = 1                       and
        LV_CDN_VB_OBD2 = 1                  and
        LV_ERR_TCO = 0                      and
        LV_END_DIAG_TCO_EL = 1              and
        LV_END_DIAG_TCO_GRD = 1             and
        (LV_END_DIAG_TCO_STUCK = 1 or
        TCO > C_TCO_STUCK_T_ES_DIAG)        and
        LV_T_ES_PLAUS_AST_END = 1           and
        LV_END_DIAG_T_ES_WIN = 1            and
        LV_ERR_ECU_NVMY = 0

Then    LV_CDN_DIAG_T_ES = 1
    If          LV_T_REL_CAN_REG = 0
    Then        ERR_SYM_T_ES = SYM_2                //no signal from CAN
    Else        If      LV_T_ES_NOT_PLAUS = 1
                Then    ERR_SYM_T_ES = SYM_3        //T_ES not plausible to TCO
                Else    ERR_SYM_T_ES = NO_SYM
                Endif
    Endif
Else    LV_CDN_DIAG_T_ES = 0
Endif
```

Remark: LV_END_DIAG_T_ES and LV_ERR_T_ES is calculated in anti bounce.

Please note:

Please note the influence the definition of the 'engine off duration' has on the 'engine off duration diagnosis':

In chapter 'Engine off duration diagnosis' ERR_SYM_T_ES = SYM_2 (no signal from CAN) is not used any longer as it is already covered by the error-entries for CAN-Timeouts. Otherwise there would be a double-error-entry.

Therefore it is important that SYM_2 "no signal from CAN" will never occur. This is currently ensured by definition of LV_END_DIAG_T_ES_WIN, defined in chapter 'Engine off duration': LV_END_DIAG_T_ES_WIN is condition for the start of the diagnosis, but LV_END_DIAG_T_ES_WIN is only set if there is a signal from CAN.

As a consequence the definition of LV_END_DIAG_T_ES_WIN **must not** be changed. This solution should be kept this way in the running projects in series.

If the definition of LV_END_DIAG_T_ES_WIN will be changed then also ERR_SYM_T_ES = SYM_2 has to be removed from diagnosis.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_T_ES	1	0...FFH	0...255	1	[-]
Debounce counter increment - T_ES diagnosis					
C_ABC_MAX_T_ES	1	1...FFH	1...255	1	[-]
Debounce counter maximum value -T_ES diagnosis					
C_TCO_STUCK_T_ES_DIAG	1	0...FEH	-48...142.5	0.75	[°C]
TCO-threshold for enabling T_ES-diagnosis independent of TCO_STUCK-diagnosis					

// Calibration hint:

the threshold C_TCO_STUCK_T_ES_DIAG is used to enable the T_ES-diagnosis in case of TCO stuck at high temperature (and no readiness of TCO_STUCK-diagnosis set); therefore the calibration of the constant should be done according to the TCO-threshold resulting from IP_TCO_SUB_DIF_DIAG_TCO_STUCK of the TCO_STUCK-diagnosis

25.5 Engine Off duration diagnosis (Applic. Inc.)

25.5.1 Interface for Rate-based-monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_T_ES	V/O	0...7H	0...7	1	[-]
Interface of T_ES monitor with the Rate-Based Monitoring statistics					
LV_CDN_RBM_DIAG_T_ES	V	0...1H	0...1	1	[-]
Individual diagnosis condition for T_ES diagnosis					

Input data:

LV_END_DIAG_T_ES	LV_END_DIAG_T_ES	T_ES_CUS	C_T_ES_TCO_PLAUS_MAX
C_T_ES_TCO_PLAUS_MIN	CTR_ERR_DYN_NR	LV_ERR_TCO	LV_ERR_ECU_NVMY
LV_ERR_TCO_STUCK_RNG			

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the T_ES monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_T_ES data.

Within STATE_RBM_T_ES, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor indiv. RBM conditions encountered within this DC (bit 2 not valid for T_ES)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_T_ES = 0

LV_CDN_RBM_DIAG_T_ES = 0

on failure memory reset :

bit 1 of STATE_RBM_T_ES = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_TCO_EL	LV_ERR_TCO_GRD	LV_ERR_TCO_STUCK	LV_ERR_TCO_PLAUS
LV_ERR_ECU_NVMY	LV_ERR_TCO_STUCK_RNG		

```
If(1)      { CPU optimization at LV_DC 0 → 1 transition }
           CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
           While bit 1 of STATE_RBM_T_ES = 0 do
                 with each XX failure of the above list :
                 ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                 SYNCHRONIZATION<CALL>)
                 If(2)        XX has a pending status
                 Then(2)      bit 1 of STATE_RBM_T_ES = 1
                 Endif(2)
           Endwhile
Else(1)    { the dynamic failure memory is empty }
           No action
Endif(1)
```

Every 1 s :

```
If         T_ES_CUS > C_T_ES_TCO_PLAUS_MAX                  or
           T_ES_CUS < C_T_ES_TCO_PLAUS_MIN
Then       LV_CDN_RBM_DIAG_T_ES = 1
Endif
If    bit 0 of STATE_RBM_T_ES = 0
Then       If    LV_END_DIAG_T_ES = 1    and
                 LV_CDN_RBM_DIAG_T_ES = 1
           Then  bit 0 of STATE_RBM_T_ES = 1
           Endif
Endif
If    bit 1 of STATE_RBM_T_ES = 0
Then       If    LV_ERR_TCO = 1    or
                 LV_ERR_ECU_NVMY = 1
           Then  bit 1 of STATE_RBM_T_ES = 1
           Endif
Endif
```

bit 2 of STATE_RBM_T_ES = 1   (no individual RBM conditions)

25.6 Cus adap module: ENSS

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_abgewuergt	O/V	0...1H	0...1	1	[-]
Motor abgewürgt					
B_dc_new	O/V	0...1H	0...1	1	[-]
Neuer Fahrzyklus mit Klemmenwechsel					
B_motlaeuft	O/V	0...1H	0...1	1	[-]
Running engine without starter, only through combustion					
B_msacanerr	O/V	0...1H	0...1	1	[-]
CAN error of MSA relevant signals					
B_msadeakt	O/V	0...1H	0...1	1	[-]
Bedingung MSADeaktivierung					
B_msaklimaav	O/V	0...1H	0...1	1	[-]
Bedingung MSA-Abschaltverhinderer von KLIMA					
B_msaklimaea	O/V	0...1H	0...1	1	[-]
Bedingung MSA-Einschaltaufforderer von KLIMA					
B_msastopt	O/V	0...1H	0...1	1	[-]
Motor stop because MSAt					
B_msavadapt	O/V	0...1H	0...1	1	[-]
MSA inhibition due to adaption					
B_msaverh	-	0...1H	0...1	1	[-]
Bedingung MSAVerhinderung					
B_schlok	O/V	0...1H	0...1	1	[-]
Boolean for key detected					
LV_ERR_STST	O/V	0...1H	0...1	1	[-]
MSA error					
LV_LSH_DOWN_OFF_REQ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
request switch off heating of downstream lambda sonde, bank selektiv					
LV_LSH_NS_OFF_REQ	O/V	0...1H	0...1	1	[-]
request switch off heating of NOx-sensor					
LV_LSH_UP_OFF_REQ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
request switch off heating of upstream lambda sonde, bank selektiv					
LV_STST_ACT	O/V	0...1H	0...1	1	[-]
MSA active					
LV_STST_ACT_READY	O/V	0...1H	0...1	1	[-]
MSA active and ready					
LV_STST_PRE_STOP_REQ	O/V	0...1H	0...1	1	[-]
Preliminary MSA engine stop request					
LV_STST_ST_REQ_CUS	O/V	0...1H	0...1	1	[-]
MSA engine start request					
LV_STST_ST_RLS	O/V	0...1H	0...1	1	[-]
MSA engine start release					
LV_STST_STOP_REQ_CUS	O/V	0...1H	0...1	1	[-]
MSA engine stop request					
T_CTR_ES	V	0...FFH	0...2550	10	[ms]
time until detection of delayed engine stop					
Tn_abstell	O	0...FFFFH	0...65535	1	[s]
engine off duration time (resolution s)					
Tn_abstellm	O/S	0...FFFFH	0...65535	1	[min]
Engine off duration time (resolution min)					
Tn_start_10	O/V	0...FFFFH	0...655350	10	[ms]
time after engine start					
Tn_start_1s	O/V	0...FFFFH	0...65535	1	[s]
Zeit nach Startende in 1s aufgelöst					
Zr_seg_nstart	O	0...FFFFH	0...65535	1	[-]
Cycle counter at start and after start (reset at exit start to after start)					

Input data:

B_hlsh1_off	B_hlsh2_off	B_hlsu1_off	B_hlsu2_off
B_hnox_off	B_msaakt	B_msaaktber	B_msaerr
B_msaprestopp	B_msastart	B_msastartf	B_msastopp
C_NR_CBK_EX	CYC_CAST	ECU_STATE	LV_DC
LV_ENG_RUN_CMB	LV_ERR_CAN_STST	LV_IGK	LV_KEY_VLD
LV_ST_END	LV_STALL	LV_STST_DEAC_ACT	LV_STST_ES
LV_STST_INH_ACT	LV_STST_INH_CDN_AD	NC_CBK_EX_NR	STATE_STST_REQ_CAN
T_AST	T_ES_CUS		

25.6.1 Outputs for BMW functions which are defined as ENSS exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: 0

Recurrence : event triggered / 10ms: B_abgewuergt, T_CTR_ES
10 ms: Zr_seg_nstart,Tn_start_10, B_motlaeuft, B_msadeakt,
B_msaverh, B_schlok

Remark: the resolution of T_AST is 100 ms

1000ms: Tn_start_1s

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
if    ECU_STAT = "PWL"
  then      B_motlaeuft   = 0
            Tn_start_10   = 0
            Tn_start_1s   = 0
  else      Tn_start_10   = T_AST
            Tn_start_1s   = T_AST
            B_motlaeuft   = LV_ENG_RUN_CMB
            B_msadeakt    = LV_STST_DEAC_ACT
            B_msaverh     = LV_STST_INH_ACT
            B_schlok      = LV_KEY_VLD
endif

Zr_seg_nstart = CYC_CAST

/* at LV_ES 0-> 1 */
T_CTR_ES = C_T_CTR_ES
```

```
/* every 10ms */
if(1)       LV_ES = 1
then(1)     decrement T_CTR_ES
            If(2)   transition T_CTR_ES->0 and LV_IGK = 1
            Then(2)
                    If(3)      LV_STST_ES = 0
                    Then(3)    B_abgewuergt = 1
                               call abwuerg_ini()
                    Else(3)    call msastopp_ini()
                    Endif(3)
            Endif(2)
Endif(1)

/* at LV_ST_END 0 -> 1*/
B_abgewuergt = 0
```

25.6.2 Outputs for BMW functions which are not defined as ENSS exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: 0

Recurrence : 10 ms: Tn_abstell Remark: the input data recurrence is 1000ms

B_msacanerr, B_msavadapt, B_msastopt, B_dc_new

100ms: B_msaklimaav, B_msaklimaea

1000 ms: Tn_abstellm

Activation: every ECU state, including WAKE_UP

Except B_msacanerr, B_msavadapt

Value in power latch phase: last valid value

Except: B_dc_new = 0

Formula section:

Remark: all formulas are valid in a **physical** meaning

B_dc_new = LV_DC

Tn_abstell = T_ES_CUS * 60

Hint:

T_ES_CUS only has a resolution of 1 minute, but Tn_abstell must have a resolution of 1 second ! Thus Tn_abstell jumps in 60 sec steps.

Tn_abstellm = T_ES_CUS

B_msacanerr = LV_ERR_CAN_STST

B_msavadapt = LV_STST_INH_CDN_AD

```
If     ECU_STATE = "PWL"
Then   B_msastopt = 0
Else   B_msastopt =  LV_STST_ES
       If      STATE_STST_REQ_CAN = " NO_REQ"
          or STATE_STST_REQ_CAN =  " INVALID_SIGNAL"
       Then  B_msaklimaav= 0,B_msaklimaea= 0
       Endif
       If      STATE_STST_REQ_CAN =  "START_REQ"
       Then  B_msaklimaav= 0,B_msaklimaea= 1
       Endif
       If      STATE_STST_REQ_CAN =  "STOP_REQ"
       Then  B_msaklimaav= 0,B_msaklimaea= 0
       Endif
       If      STATE_STST_REQ_CAN =  "START_INH"
       Then  B_msaklimaav= 0,B_msaklimaea= 0
       Endif
       If      STATE_STST_REQ_CAN =  "STOP_INH"
       Then  B_msaklimaav= 1,B_msaklimaea= 0
       Endif
Endif
```

25.6.3 Outputs for SV aggregates, customer → SV

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation:	0
Recurrence :	10 ms / 100ms
Activation:	at every engine state
Deactivation:	-

Formula section:

/* recurrence 10ms */

LV_STST_ACT = B_msaakt

LV_STST_ACT_READY = B_msaaktber

LV_ERR_STST = B_msaerr

LV_STST_ST_RLS = B_msastartf

```
LV_STST_ST_REQ_CUS              = B_msastart
LV_STST_STOP_REQ_CUS      = B_msastopp
LV_STST_PRE_STOP_REQ      = B_msaprestopp

/* recurrence 100ms */
LV_LSH_NS_OFF_REQ = B_hnox_off
If    C_NR_CBK_EX = 2    //OL configuration
then
    LV_LSH_DOWN_OFF_REQ[1] = B_hlsh1_off
    LV_LSH_DOWN_OFF_REQ[2] = B_hlsh2_off
    LV_LSH_UP_OFF_REQ[1] = B_hlsu1_off
    LV_LSH_UP_OFF_REQ[2] = B_hlsu2_off
Else     // C_NR_CBK_EX = 1 , UL configuration
    LV_LSH_DOWN_OFF_REQ[1] = B_hlsh1_off
    LV_LSH_DOWN_OFF_REQ[2] = B_hlsh1_off
    LV_LSH_UP_OFF_REQ[1] = B_hlsu1_off
    LV_LSH_UP_OFF_REQ[2] = B_hlsu1_off
EndIf
```

Calibration data:

Name	Dimension	Hex. limits	Phys. limits	Resol.	Unit
C_T_CTR_ES	1	0...FFH	0...2550	10	[ms]
delay time for detection of engine stop					

26 Engine temperature

Table of Contents

Index

26.1 ENTE Configuration data for MSD80 project

Input data:

NLC_TCO_2_CONF	NC_ECTPWM_MAX	NC_ECTPWM_MIN	NC_ECTPWM_SUB_DIAG

FUNCTION DESCRIPTION:

General information:

The logical constants NC(NLC)_xxx_CONF are necessary to adapt the Aggregate to a given system environment. The values describe either the variant of a used system component, or a vehicle component is available or not.

With use of the logical constant **NLC_ECT_CONF** the availability of an electronically controlled thermostat within the system is determined. In case of a vehicle configuration with an ECT, the component control and diagnosis is included within the aggregate version. (refer: ENTE A.I.D.)

With use of the logical constant **NLC_TCO_2_CONF** the availability of a coolant temperature sensor at radiator outlet is determined. In case of a vehicle configuration with a TCO_2 sensor, the sensor signal acquisition and diagnosis is included within the aggregate version. (refer: ENTE A.I.D.)

The configuration constant **NC_ECF_CONF** allows to determine the wanted fan control strategy. In principal it is possible to choose between a RLY- or a PWM-control architecture. A RLY/PWM fan control strategy in parallel is also existing. In this case the aggregate architecture allows switching either the control of RLY-fan(s) or the control of PWM–fan(s) during ECU runtime depending on the setting of a corresponding configuration bit. The control of both variants (RLY- and PWM-fan(s)) at the same time is not allowed and not supported within the aggregate.

Independent on the chosen fan control strategy (PWM/RLY), the configuration constant **NC_ECF_NR** describes the number of available cooling fans (hardware components) at the vehicle. In case of a PWM-fan configuration, the number of available PWM cooling fans at the vehicle is always equal to the number of PWM fan output stages provided by the ECU hardware.

Because one or more RLY-switch(es) may control only one cooling fan (hardware component) in case of a RLY-fan configuration, the number of RLY-switch(es) per cooling fan can be different. With use of the configuration constant **NC_ECF_RLY_NR**, the number of RLY output stages per cooling fan are determined.

With use of the constant **NC_ECTPWM_xxx** it is possible to adjust the PWM output range of the electronic controlled thermostat signal during normal operation conditions or when a thermostat error occurs. In case of a thermostat failure the PWM output signal is set to a defined value for safety issues.

The following describes the general rules for determination of the configuration data.

26.1.1 Local configuration data

Here are listed the configuration data, which are used only in the ENTE aggregate.

Data	**Value**
NLC_ECT_CONF	[1] Electronically controlled thermostat (ECT) is available
NLC_TCO_2_CONF	[1] Coolant temperature sensor at radiator outlet is available
NC_ECF_CONF	[1] Control of PWM-fan(s) requested (only PWM fan(s))
NC_ECF_NR	[1] Number of available cooling fans (hardware components) at the vehicle
NC_ECF_RLY_NR	[0] Number of RLY output stages per cooling fan (hardware components) (NC_ECF_RLY_NR=0 have to be adjusted in case of NC_ECF_CONF=1)
NC_ECTPWM_MAX	99,6 % Maximum limitation of ECTPWM signal
NC_ECTPWM_MIN	0,4 % Minimum limitation of ECTPWM signal
NC_ECTPWM_SUB_DIAG	[4,31%] – (B08**H**) Adjusted ECTPWM signal in case of failures
NC_NR_TCO_SENS	2

Configuration data:

Name	**Dim**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
NLC_ECT_CONF	1	0...1H	0...1	1	[-]
System configuration flag (compiler) if an electronic controlled thermostat is available (=1) or not (=0)					
NC_ECF_CONF	1	0...3H	0...3	1	[-]
System configuration switch for cooling fan control strategy (RLY || PWM || RLY and PWM)					
NC_ECF_NR	1	1...FFH	1...255	1	[-]
Number of available cooling fans (hardware components) at the vehicle					
NC_ECF_RLY_NR	1	0...FFH	0...255	1	[-]
Number of RLY output stages per cooling fan (hardware components)					
NC_NR_TCO_SENS	1	1...FFH	1...255	1	[-]
Number of available coolant temperature sensors (hardware components) at the vehicle					

26.2 Acquisition of electronically controlled thermostat signal

26.2.1 Pulse width modulated outputs

Input data:

ECTPWM			

FUNCTION DESCRIPTION:

General information:

Not all vehicles are equipped with an electronic controlled thermostat for the regulation of the coolant liquid circulation. Depending on the hardware configuration a pulse width modulated value (ECTPWM) for the electronic controlled thermostat is calculated or not. Only in case of a vehicle configuration with an electronic controlled thermostat a pulse width modulated signal is generated at the ECU output.

PWM output name	Frequency	Updating of duty cycle	Range of duty cycle
ECTPWM	hardware dependent	1000 ms	0 -100%

Application conditions:

Initialisation at reset: ECTPWM = 0%

Activation: at every engine operating state

Deactivation: -

26.3 Acquisition of electronically controlled cooling fan signal

Input data:

ECFPWM[NC_ECF_NR]	NC_ECF_NR		

FUNCTION DESCRIPTION:

General information:

Depending on the hardware configuration a pulse width modulated value is calculated (ECFPWM[NC_ECF_NR]) to control the requested cooling fan(s). In case of a vehicle configuration with a PWM controlled cooling fan(s), a signal at the ECU output is generated.

PWM output name	Frequency	Updating of duty cycle	Range of duty cycle
ECFPWM[NC_ECF_NR] (ECFPWM_ECF)	100 Hz (10 Hz during power latch)	200 ms	0 -100%

Application conditions:

Activation: at every engine operating state

Deactivation: -

(see: "ENTE scheduler")

26.4 ENTE – Requirements to infrastructure interface

Input data:

NC_NR_TCO_SENS			

Export actions:

ACTION_INFR_GetVpTco (OUT <>)
This action provides the digitized voltage value of the available coolant temperature sensors in an array.

Description for actions:

ACTION_INFR_GetVpTco (OUT <>)					
This action provides the digitized voltage value of the available coolant temperature sensors in an array delivered from the infrastructure. The AD conversion is performed autonomously by the infrastructure. When the action is called the gathered information will be provided to the application software level. The dimension of the parameter *Vp_tco_sens* is given by configuration input NC_NR_TCO_SENS. The index range is 0 … NC_NR_TCO_SENS-1.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Vp_tco_sens [NC_NR_TCO_SENS]	OUT	0...7FFFH	0...4.999847412	1.53E-04	V
Coolant temperature sensor voltage value					

FUNCTION DESCRIPTION:

General information:

The following action is used for the acquisition of the digitized voltage value of the adressed coolant temperature sensor in an array. The action delivers the coolant temperature sensor voltage value to the application software level from the standard AD converter queue.

- NC_NR_TCO_SENS is valid from 1...255.

  The voltage belonging to the first sensor is returned in Vp_tco_sens[0].

  The voltage belonging to the *n* sensor is returned in Vp_tco_sens[n-1].
- The AD conversion is performed autonomously by the infrastructure, the returned value is not older than 10 ms.
- The voltage value is gathered in the infrastructure until the application reads out the information by calling the action, old values are replaced by new values.

Requirements for ACTION_INFR_GetVpTco:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Vp_tco_sens [NC_NR_TCO_SENS]			10 Bit	The following coinicidence requirement has to be fulfilled in case of NC_NR_TCO_SENS > 1: Vp_tco_sens[k] is digitized at point in time t_k. For the returned array the maximum allowed time periode $\mid t_k - t_i \mid$ must less than 10ms.	No comment

Diagnosis: no electrical diagnosis done here

Coincidence requirements: When calling this action, all returned voltages must be within the last 100 ms.

26.5 Engine temperature management input/output

Input data:

LV_RAS_OUT			

Export actions:

ACTION_INFR_SetRas (IN <Lv_ras_out>)
This action gives the control of the RAS signal
ACTION_INFR_GetElDiagRas (OUT<Cdn_diag_ras_out_el>, OUT<Err_diag_ras_out_el>)
This action reads the primary failure and condition information for a symptom of the radiator shutter

Description for actions:

ACTION_INFR_SetRas(IN <Lv_ras_out>)					
This action gives the control of the RAS signal					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
lv_ras_out	IN	0...1H	0...1	1	-
lv_ras_out = 1 --> PKKS open. Magnet not feeded lv_ras_out = 0 --> PKKS closed. Magnet feeded					

ACTION_INFR_GetElDiagRas (OUT<Cdn_diag_ras_out_el>, OUT<Err_diag_ras_out_el>)					
This action reads the primary failure and condition information for a symptom of the RAS.					
Name	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cdn_diag_ras_out_el	OUT	0...7H	0...7	1	-
Primary diagnosis condition for each symptom of RAS Bit 0: diagnosis condition for symptom SYM_0 Bit 1: diagnosis condition for symptom SYM_1 Bit 2: diagnosis condition for symptom SYM_2					
Err_diag_ras_out_el	OUT	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Raw value of error symptom for RAS (only parameter)					

FUNCTION DESCRIPTION:

1.1 Set bit signal for RAS control

General information:

This action is used to set the RAS signal.

Requirements for ACTION_INFR_SetRas:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Lv_ras_out	-	-	1 Bit		The Ras_Out signal is a single bit signal lv_ras_out = 1 PKKS open – Magnet not feeded lv_ras_out = 0 PKKS closed – Magnet feeded

Diagnosis: no electrical diagnosis done here

Coincidence requirements: An update of the signal passed to the Infrastructure has to be performed within the next five periods.

1.2 Get RAS electrical diagnosis result

General information:

This action returns the result of the electrical diagnosis of the RAS.

- The device readout is performed autonomous by the Infrastructure.
- The error information are gathered in the Infrastructure until the Application reads out the information by calling ACTION_INFR_GetElDiagRas.
- After having read out the information by calling ACTION_INFR_GetElDiagRas, the data inside the Infrastructure are reset.

Requirements for ACTION_INFR_GetElDiagRas:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cdn_diag_ras_out_el	-	-	<bit coded>	Err_diag_ras_out_el	Diagnosis condition for each symptom of RAS **Bit 0:** diagnosis condition for symptom SYM_0 **Bit 1:** diagnosis condition for symptom SYM_1 **Bit 2:** diagnosis condition for symptom SYM_2 The relevant bit is set, if the condition for a valid diagnosis is fulfilled.
Err_diag_ras_out_el	-	-	<bit coded>	Cdn_diag_ras_out_el	Bit coded result of each symptom **0H** = NO_SYM **1H** = SYM_0 – SCP **2H** = SYM_1 – SCG **4H** = SYM_2 – OC **8H =** SYM_3 – not used The relevant bit is set, if the error has been detected.

Diagnosis: ACTION_INFR_GetElDiagRas returns the electric diagnosis of the RAS.

Coincidence requirements: no coincidence requirements to other events

26.6 AGGR ENTE adaptation: Engine temperature

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TCO_EX	V/O	0...FEH	-48...142.5	0.75	[°C]
TCO_2 adaptation					
TCO_EX_MES	V/O	0...FEH	-48...142.5	0.75	[°C]
TCO_2_MES adaptation					
LV_ERR_TCO_EX	V/O	0...1H	0...1	1	[-]
Adaptation of TCO_2 diagnosis to ERRM					
ECFPWM_ECF	V/O	0...FFH	0...99.60937	0.390625	[%]
Electric fan control pulse width modulation					
LV_ERR_ECT_MEC	V/O	0...1H	0...1	1	[-]
Boolean for electronic controlled thermostat mechanical error					

Input data:

TCO_2	LV_ERR_TCO_2		TCO_2_MES
ECFPWM[NC_ECF_NR]	LV_ES		LV_ERR_TH

FUNCTION DESCRIPTION:

Adaptation of ENTE variables to MSD70 environment.

General information:

At the MSD70 platform only one electronic controlled cooling fan is available. The number of cooling fans is configured with “1” therefore. In this case the following condition is permitted:

ECFPWM[NC_ECF_NR] = ECFPWM[1]

Application conditions:

Initialisation: all 0 at reset
except:

LV_ERR_ECT_MEC = LV_ERR_TH

Recurrence: 1000 ms

Activation: every engine state

Formula section:

TCO_EX = TCO_2

TCO_EX_MES = TCO_2_MES

LV_ERR_TCO_EX = LV_ERR_TCO_2

ECFPWM_ECF = ECFPWM[1]

LV_ERR_ECT_MEC = LV_ERR_TH

26.7 Coolant temperature

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TCO	V/O	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature					
TCO_MES	V/O	0...FEH	-48...142.5	0.75	[°C]
Measured coolant temperature					
TCO_ST	V/O	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature at start					
TCO_STOP	V/O/S	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature at transition to engine stop (ES)					
LV_CDN_INI_TCO	V	0...1H	0...1	1	[-]
Boolean for coolant temperature initialisation conditions					
VP_TCO[NC_NR_TCO_SENS]	V/O	0...7FFFH	0...4.99984	0.1526e-3	[V]
Sensor raw acquisition of all available coolant temperature sensors					
TCO_ST_DC	V/O	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature at first start of driving cycle					

Input data:

TCO_SUB	LV_ERR_TCO	LV_ERR_TCO_PREL	TIA_IM
LV_ERR_TIA_IM	LV_ES	NC_NR_TCO_SENS	VP_TCO_COPL[NC_NR_TCO_SENS]
LV_ECU_SLA			

Import actions:

ACTION_INFR_GetVpTco (OUT <>)

FUNCTION DESCRIPTION:

General information:

The coolant temperature raw sensor voltage is measured with use of an A/D converter by the ECU hardware. The software then is generating a modified sensor signal (VP_TCO[0]), which is requested by the application software for further execution.

The modified sensor signal (VP_TCO[0]) is converted into a measured temperature value (TCO_MES) with use of a one-dimension interpolation table (IP_TCO_MES).

Application conditions:

See separate chapters:

26.7.1 Coolant temperature gradient monitoring

FUNCTION DESCRIPTION:

General information:

The measured coolant temperature (TCO_MES) is monitored for not plausible gradients. It is checked if the temperature difference between the old and the new measured coolant temperature (TCO_MES) exceeds the permissible gradient C_TCO_MES_GRD_MAX. In this case the old measured value remains unchanged. If the temperature difference between the measured coolant temperature values is exceeding the permissible gradient again during the next measurement, the new measured value is transferred into the working memory. A measured value can be inhibited only if the gradient exceeded for one time (recurrence).

The monitoring of the coolant temperature gradient does not lead to failure entries. The only purpose is to extract implausible measured values (noise).

Application conditions:

Initialisation at Reset:

```
FOR k = 0 to NC_NR_TCO_SENS -1
    IF  LV_ECU_SLA =1
        THEN VP_TCO[k] = VP_TCO_COPL[k]
        ELSE
        ACTION_INFR_GetVpTco(VP_TCO[k])

    ENDIF
END FOR
// all sensor values are imported and provided as output
$TCO\_MES_{(n)} = TCO\_MES_{(n-1)} = IP\_TCO\_MES$
// acquisition of the main coolant temperature measurement signal
```

Recurrence: 100 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

```
// all sensor values are imported and provided as output
FOR k = 0 to NC_NR_TCO_SENS - 1
    IF  LV_ECU_SLA =1
        THEN VP_TCO[k] = VP_TCO_COPL[k]
        ELSE
        ACTION_INFR_GetVpTco(VP_TCO[k])
    ENDIF
END FOR
// VP_TCO[0] is used as input signal for the interpolation table IP_TCO_MES
TCO_MES = IP_TCO_MES
// acquisition of the main coolant temperature measurement signal
If      $| TCO\_MES_{(n-1)} - TCO\_MES_{(n)} | > C\_TCO\_MES\_GRD\_MAX$
then    $TCO\_MES_{(n)} = TCO\_MES_{(n-1)}$          (only one time!)
```

```
else        TCO_MES(n) = TCO_MES(n)
endif
```

26.7.2 Coolant temperature acquisition

FUNCTION DESCRIPTION:

General information:

The actualisation of the coolant temperature value (TCO) is only executed, if no coolant temperature error / preliminary error is detected. In case of a detected error, the TCO value will be set to the calculated coolant temperature substitute value (TCO_SUB).

The initialisation of the coolant temperature value (TCO) is always performed as long as the engine operating state “Engine stop” is active.

Application conditions:

Initialisation:

at Reset TCO = TCO_MES
at Reset **or** Engine stop: LV_CDN_INI_TCO = 1

at Engine stop to Engine run (ES_to_ERU):
LV_CDN_INI_TCO = 0

Activation: at every engine operating state

Deactivation: -

26.7.2.1 Coolant temperature acquisition at engine stop

Application conditions:

Recurrence: 10 ms

Activation: LV_CDN_INI_TCO = 1

Deactivation: LV_CDN_INI_TCO = 0

Formula section:

```
if (1)          LV_ERR_TCO = 0          and
                LV_ERR_TCO_PREL = 0
then (1)        TCO = TCO_MES
else (1)
        if (2)      LV_ERR_TIA_IM = 0
        then (2)    TCO = IP_TCO_INI_TIA_IM
        else (2)    TCO = C_TCO_INI_TIA_IM
```

```
        endif (2)
endif (1)
```

26.7.2.2 Coolant temperature acquisition at engine run

Recurrence: 100 ms

Activation: LV_CDN_INI_TCO = 0

Deactivation: LV_CDN_INI_TCO = 1

Formula section:

```
if (1)          LV_ERR_TCO = 0
then (1)
        if (2)      LV_ERR_TCO_PREL = 0
        then (2)    TCO = TCO_MES
        else (2)    $TCO_{(n)} = TCO_{(n-1)}$
        endif (2)
else (1)        TCO = TCO_SUB
endif (1)
```

26.7.3 Coolant temperature at engine start

FUNCTION DESCRIPTION:

General information:

The coolant temperature at engine start (TCO_ST) is determined as long as the engine operating state “Engine stop” is detected. During this time, TCO_ST is set with the actual coolant temperature value TCO.

Application conditions:

Initialisation at Reset:

TCO_ST = TCO_MES

TCO_ST_DC = TCO_MES

Recurrence: 10 ms

Activation: LV_ES = 1

Deactivation: LV_ES = 0

Formula section:

TCO_ST = TCO
TCO_ST_DC = TCO

26.7.4 Coolant Temperature at engine stop

FUNCTION DESCRIPTION:

General information:

The coolant temperature at engine stop (TCO_STOP) is determined at each transition to the engine operating state "Engine stop". TCO_STOP will be stored in the non-volatile memory during the ECU's self-holding phase.

Application conditions:

Initialisation at Reset:

TCO_STOP = TCO_STOP

(stored in the non-volatile memory)

Initialization at Engine run to Engine stop (ERU_to_ES):

TCO_STOP = TCO

Activation: at every engine operating state

Deactivation: -

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TCO_MES	16	0...FEH	-48...142.5	0.75	[°C]
LDP_VP_TCO_IP_TCO_MES	16	0...7FFFH	0...4.99984	0.1526e-3	[V]
Linearization of the coolant temperature sensor voltage for temperature acquisition					
IP_TCO_INI_TIA_IM	6	0...FEH	-48...142.5	0.75	[°C]
LDP_TIA_IM_IP_TCO_INI_TIA_IM	6	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature initialization value in case of a coolant temperature failure					
C_TCO_INI_TIA_IM	1	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature initialization value in case of a coolant- and intake air- temperature failure					
C_TCO_MES_GRD_MAX	1	0...FEH	0...190.5	0.75	[°C]
Maximum permissible temperature gradient for coolant temperature gradient monitoring					

26.8 Coolant temperature (radiator outlet)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TCO_2	V/O	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature (radiator outlet)					
TCO_2_MES	V/O	0...FEH	-48...142.5	0.75	[°C]
Measured coolant temperature (radiator outlet)					
TCO_2_ST	V/O	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature (radiator outlet) at start					
TCO_2_MES_ST	V/O	0...FEH	-48...142.5	0.75	[°C]
Measured coolant temperature (radiator outlet) at start					
TCO_1_2	V/O	80...7FH	-96...95.25	0.75	[°C]
Coolant temperature difference between the TCO- and the TCO_2 sensor					
LV_CDN_INI_TCO_2	V	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) initialisation conditions					

Input data:

LV_ERR_TCO_2	TCO	LV_ERR_TCO_2_PREL	LV_ES
VP_TCO[NC_NR_TCO_SENS]	NC_NR_TCO_SENS	LV_VAR_TCO_2	

FUNCTION DESCRIPTION:

General information:

The coolant temperature raw sensor voltage at radiator outlet is measured with use of an A/D converter by the ECU hardware. The software then is generating a modified sensor signal (VP_TCO[1]), which is requested by the application software for further execution.

The modified sensor signal (VP_TCO[1]) is converted into a measured temperature value at radiator outlet (TCO_2_MES) with use of a one-dimension interpolation table (IP_TCO_2_MES).

If the sensor is not available (LV_VAR_TCO_2 = 0) then TCO_2 is set to an substitute value.

Application conditions:

See separate chapters:

26.8.1 Coolant temperature (radiator outlet) gradient monitoring

FUNCTION DESCRIPTION:

General information:

The measured coolant temperature at radiator outlet (TCO_2_MES) is monitored for not plausible gradients. It is checked if the temperature difference between the old and the new measured coolant temperature (TCO_2_MES) exceeds the permissible gradient C_TCO_2_MES_GRD_MAX. In this case the old measured value remains unchanged. If the temperature difference between the measured coolant temperature values at radiator outlet is exceeding the permissible gradient again during the next measurement, the new measured value is transferred into the working memory. A measured value can be inhibited only if the gradient exceeded for one time (recurrence).

The monitoring of the coolant temperature (radiator outlet) gradient does not lead to failure entries. The only purpose is to extract implausible measured values (noise).

Application conditions:

Initialisation at Reset:

$TCO\_2\_MES_{(n)} = TCO\_2\_MES_{(n-1)} = IP\_TCO\_2\_MES$

// acquisition of the coolant temperature (radiator outlet) measurement signal

Recurrence: 100 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

```
If    $LV\_VAR\_TCO\_2 = 1$
then
      $TCO\_2\_MES = IP\_TCO\_2\_MES$
      // VP_TCO[1] is used as input signal for the interpolation table IP_TCO_2_MES
      If      $| TCO\_2\_MES_{(n-1)} - TCO\_2\_MES_{(n)} | > C\_TCO\_2\_MES\_GRD\_MAX$
      then    $TCO\_2\_MES_{(n)} = TCO\_2\_MES_{(n-1)}$          (only one time!)
      else    $TCO\_2\_MES_{(n)} = TCO\_2\_MES_{(n)}$
      endif
else
      If      $LC\_SWI\_TCO\_2\_SUB = 0$
      then    $TCO\_2\_MES = TCO - C\_TCO\_2\_DIF\_TCO$
      else    $TCO\_2\_MES = C\_TCO\_2\_SUB$
      endif
endif
```

26.8.2 Coolant temperature (radiator outlet) acquisition

FUNCTION DESCRIPTION:

General information:

The actualisation of the coolant temperature value at radiator outlet (TCO_2) is only executed, if no coolant temperature (radiator outlet) error / preliminary error is detected. In case of a detected error, the TCO_2 value will be set to its substitute value, which is calculated by the coolant temperature (TCO) in combination with an offset value.

The initialisation of the coolant temperature value at radiator outlet (TCO_2) is only performed once per engine run as long as the engine operating state "Engine stop" is active. If the engine stalls and in this case the engine operating state is changing to "Engine stop" again, a new initialization of the coolant temperature at radiator outlet will not be done. Power latch must always be finished before a new initialization process is performed as long as no ECU reset occurs. An ECU reset is handled like the beginning of a new engine run and an initialization of the TCO_2 value is also done in this case for at least one recurrence.

Application conditions:

Initialisation at Reset

LV_CDN_INI_TCO_2 = 1

TCO_2 = TCO_2_MES

Initialisation at Engine stop to Engine run (ES_to_ERU):

LV_CDN_INI_TCO_2 = 0

Activation: at every engine operating state

Deactivation: -

26.8.2.1 Coolant temperature (radiator outlet) acquisition (at ES after RESET)

Application conditions:

Recurrence: 10 ms

Activation: LV_CDN_INI_TCO_2 = 1

Deactivation: LV_CDN_INI_TCO_2 = 0

Formula section:

if (1) LV_ERR_TCO_2 = 0 **and**
LV_ERR_TCO_2_PREL = 0
then (1) TCO_2 = TCO_2_MES
else (1) TCO_2 = TCO - C_TCO_2_DIF_TCO

```
endif (1)
```

26.8.2.2 Coolant temperature (radiator outlet) acquisition (at ERU / at ES after ERU_to_ES)

Application conditions:

Recurrence: 100 ms

Activation: LV_CDN_INI_TCO_2 = 0

Deactivation: LV_CDN_INI_TCO_2 = 1

Formula section:

```
if (1)          LV_ERR_TCO_2 = 0
then (1)
        if (2)      LV_ERR_TCO_2_PREL = 0
        then (2)    TCO_2 = TCO_2_MES
        else (2)    $TCO\_2_{(n)} = TCO\_2_{(n-1)}$
        endif (2)
else (1)        TCO_2 = TCO - C_TCO_2_DIF_TCO
endif (1)
```

26.8.3 Coolant temperature (radiator outlet) at engine start

FUNCTION DESCRIPTION:

General information:

The coolant temperature (radiator outlet) and the measured coolant temperature at engine start (TCO_2_ST, TCO_2_MES_ST) is determined as long as the engine operating state “Engine stop” is detected. During this time, TCO_2_ST is updated with the actual coolant temperature value at radiator outlet (TCO_2) and TCO_2_MES_ST is updated with the measured coolant temperature value at radiator outlet (TCO_2_MES).

Application conditions:

Initialisation at Reset:

TCO_2_ST = TCO_2_MES

TCO_2_MES_ST = TCO_2_MES

Recurrence: 10 ms

Activation: LV_ES = 1

Deactivation: LV_ES = 0

Formula section:

TCO_2_ST = TCO_2
TCO_2_MES_ST = TCO_2_MES

26.8.4 Temperature difference between the coolant temperature sensors

FUNCTION DESCRIPTION:

General information:

The main coolant temperature sensor (TCO) is located next to the engine, while the second coolant temperature sensor (TCO_2) is fixed at radiator outlet. The temperature difference between the coolant temperature values (TCO_1_2) is determined and used for further calibration issues. Only positive values for the coolant temperature difference are plausible at normal driving conditions.

Application conditions:

Initialisation at Reset:

TCO_1_2 = 95.25°C

Recurrence: 100 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

TCO_1_2 = TCO – TCO_2

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TCO_2_MES	16	0...FEH	-48...142.5	0.75	[°C]
LDP_VP_TCO_2_IP_TCO_2_MES	16	0...7FFFH	0...4.999847412	1.53E-04	[V]
Linearization of the coolant temperature (radiator outlet) sensor voltage (VP_TCO[1]) for temperature acquisition					
C_TCO_2_MES_GRD_MAX	1	0...FEH	0...190.5	0.75	[°C]
Maximum permissible temperature gradient for coolant temperature (radiator outlet) gradient monitoring					
C_TCO_2_DIF_TCO	1	FFFFFF80...7FH	-96...95.25	0.75	[°C]
Temperature difference to the TCO-value in case of a coolant temperature (radiator outlet) failure					
C_TCO_2_SUB	1	0...FEH	-48...142.5	0.75	[°C]
Substitute value for TCO_2 in case of no sensor is available					
LC_SWI_TCO_2_SUB	1	0...1H	0...1	1	[-]
Minimum engine cooling temperature to activate the diagnosis condition					

26.9 Coolant temperature substitute model

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TCO_SUB	O/V	0...FE00H	-48...142.5	2.9297e-3	[°C]
Coolant temperature substitute value					
TCO_SUB_INC	V	0...FE00H	-48...142.5	2.9297e-3	[°C]
Coolant temperature substitute increment value					
FAC_TCO_SUB_COR	V	0...FFH	0...1.99218	0.0078125	[-]
Coolant temperature substitute value correction factor					
FAC_TCO_SUB_COR_N_REL_CWP	V	0...FFFFH	0...255.99609	3.9063e-3	[-]
Coolant temperature substitute value correction factor for CWP functionality					
FAC_TCO_SUB_COR_STATE_COC	V	0...FFH	0...51	0.2	[-]
Coolant temperature substitute value correction factor after warm-up					
FAC_TCO_SUB_VS	V	0...FFFFH	-8...7.99975	0.2441e-3	[-]
Coolant temperature substitute value vehicle speed correction factor					
T_DLY_TCO_SUB	V	0...FFFFH	0...65535	1	[s]
Delay time counter for coolant temperature substitute value calculation during pull fuel cutoff or idle speed					
T_DLY_TCO_SUB_COR_STATE_COC	V	0...FFFFH	0...65535	1	[s]
Delay time counter for coolant temperature substitute value calculation after warm-up					
LV_CDN_ENA_TCO_SUB	V	0...1H	0...1	1	[-]
Boolean for coolant temperature substitute model initialisation condition					
T_DLY_THD_TCO_SUB_ST	V	0...FFFFH	0...6553.5	0.1	[s]
Delay time threshold for coolant temperature substitute value depending on CWP speed					
N_REL_CWP_GRD	V	8000...7FFFH	-32768...32767	1	[-]
Gradient of N _ REL _ CWP					
CTR_N_REL_CWP_H	V	0...FFH	0...255	1	[-]
Counter to activate TCO-SUB correction if Heater is active					
T_N_REL_CWP_GRD_H	V	0...FFH	0...255	1	[s]
Timer to block coolant temperature model correction					

Input data:

LV_ST	LV_IS	LV_PL	LV_PUC
LV_PU	TCO	TCO_ST	LV_ERR_TCO
TAM	MAF_KGH	N_REL_CWP	T_AST
VS	OPM_AV		

FUNCTION DESCRIPTION:

General information:

The coolant temperature substitute model (TCO_SUB) is initialized with the coolant temperature at engine start (TCO_ST) as soon as the system event “Engine stop” to “Engine run” is detected. The substitute model increment values are dependent on the actual engine performance (equivalent to mass air flow MAF_KGH), ambient temperature and vehicle speed. To take the smaller incrementation values at higher coolant temperatures into consideration, the increment values are weighted.

During “Idle speed” and “Pull fuel cutoff” phases the calculation of the coolant temperature substitute value remains activated.

A delay time after engine start to activate the incrementation of the temperature substitute value is incremented, to simulate the warm up of the real coolant temperature very efficient.

In case of a coolant temperature acquisition failure (LV_ERR_TCO = 1), the coolant temperature substitute value is reinitialized with the latest valid coolant temperature value. In this case the model incrementation starts from the initialization value on.

Application conditions:

Initialisation at Engine stop to Engine run (ES_to_ERU):

T_DLY_TCO_SUB = 0

T_DLY_TCO_SUB_COR_STATE_COC = 0

$TCO\_SUB_{(n)} = TCO\_SUB_{(n-1)} = TCO\_ST$

FAC_TCO_SUB_COR_STATE_COC = 1

Initialisation at Exit start (EXIT_ST):

LV_CDN_ENA_TCO_SUB = 1

T_N_REL_CWP_GRD_H = C_T_N_REL_CWP_GRD_H

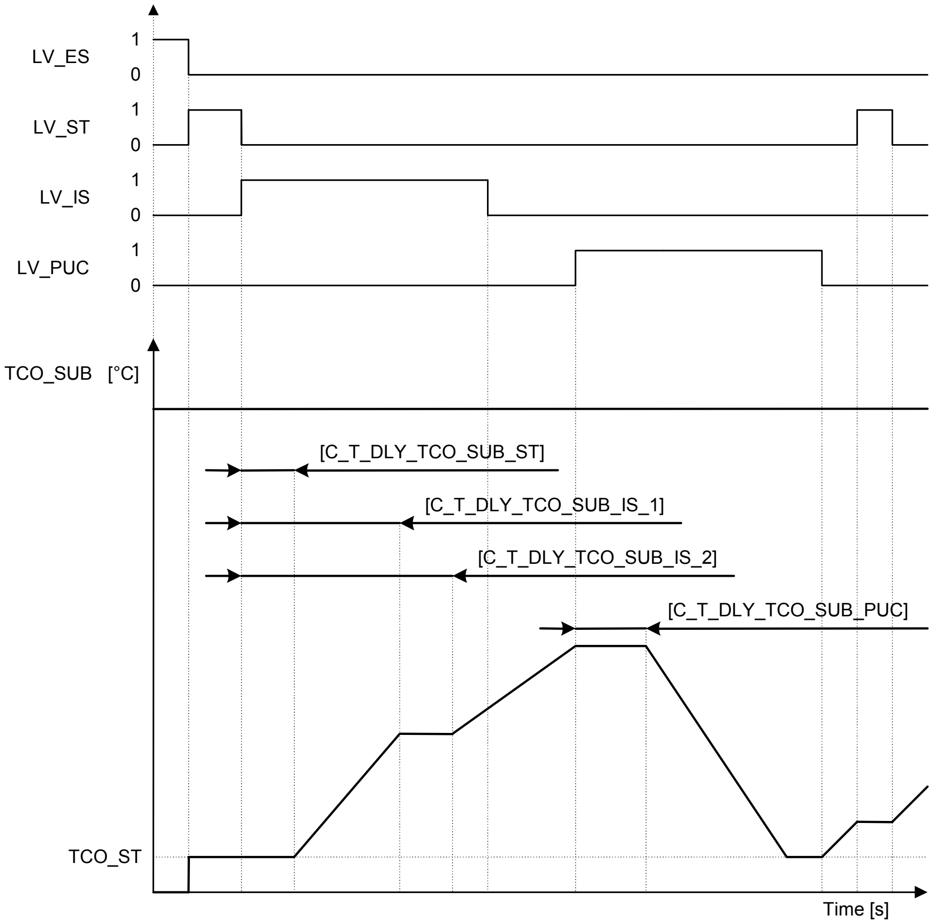
CTR_N_REL_CWP_H = C_CTR_N_REL_CWP_H

Initialisation at Engine run to Engine stop (ERU_to_ES):

LV_CDN_ENA_TCO_SUB = 0

Recurrence:	1000 ms
Activation:	LV_CDN_ENA_TCO_SUB = 1
Deactivation:	LV_CDN_ENA_TCO_SUB = 0

Signal flow diagram:



Formula section:

Calculation of the factor FAC_TCO_SUB_COR_STATE_COC for CWP functionality:

```
If          N_REL_CWP  <  C_N_REL_CWP_THD_COC
then        T_DLY_TCO_SUB_COR_STATE_COC  =  0
            T_DLY_THD_TCO_SUB_ST  =  C_T_DLY_TCO_SUB_ST_CWP_OFF
            FAC_TCO_SUB_COR_STATE_COC = 1
else        T_DLY_THD_TCO_SUB_ST  =  C_T_DLY_TCO_SUB_ST_CWP_ON
    if      T_DLY_TCO_SUB_COR_STATE_COC
                < C_T_DLY_TCO_SUB_COR_STATE_COC
    then    FAC_TCO_SUB_COR_STATE_COC  =
            C_FAC_TCO_SUB_COR_STATE_COC
    else    FAC_TCO_SUB_COR_STATE_COC  =  1
    endif
            T_DLY_TCO_SUB_COR_STATE_COC  =
            T_DLY_TCO_SUB_COR_STATE_COC + 1 s
endif
```

Calculation of factor FAC_TCO_SUB_COR_N_REL_CWP for CWP functionality:

```
            FAC_TCO_SUB_COR_N_REL_CWP  =
            IP_FAC_TCO_SUB_COR_N_REL_CWP
                * FAC_TCO_SUB_COR_STATE_COC
```

Calculation of the substitute model increment value during “Engine start”:

```
If          LV_ST = 1
then        TCO_SUB_INC  =  0
endif
```

Calculation of factor FAC_TCO_SUB_VS for vehicle speed functionality:

```
If  (TCO_SUB >= C_TCO_SUB_MIN_FAC_VS)
then
   FAC_TCO_SUB_VS = IP_FAC_TCO_SUB_VS
else
  FAC_TCO_SUB_VS = 1
endif
```

Calculation of the substitute model increment value during “Part load” or “Pull”:

```
If (1)          LV_PL = 1                                  or
```

```
                    LV_PU = 1
then (1)
    if (2)   OPM_AV= S   (at operation mode stratified)
    then (2)
             TCO_SUB_INC  =
                      IP_TCO_SUB_INC_S * IP_FAC_TCO_SUB_INC_S *
                      FAC_TCO_SUB_COR* FAC_TCO_SUB_COR_N_REL_CWP *
                      FAC_TCO_SUB_VS
    else (2)
             TCO_SUB_INC  =
                      IP_TCO_SUB_INC * IP_FAC_TCO_SUB_INC * FAC_TCO_SUB_COR
                      * FAC_TCO_SUB_COR_N_REL_CWP * FAC_TCO_SUB_VS
    endif (2)
endif (1)
```

Calculation of the substitute model increment value during “Pull fuel cutoff”:

```
If   Transition:   LV_PUC = 0 -> LV_PUC = 1
then               T_DLY_TCO_SUB = 0
endif
if (1)             LV_PUC = 1
then (1)
    if (2)         T_DLY_TCO_SUB < C_T_DLY_TCO_SUB_PUC
    then (2)       TCO_SUB_INC  =  0
    else (2)       TCO_SUB_INC  = -( IP_TCO_SUB_DEC )
    endif (2)
                   T_DLY_TCO_SUB = T_DLY_TCO_SUB + 1 s
endif (1)
```

Calculation of the substitute model increment value during “Idle speed”:

```
If   Transition:   LV_IS = 0 -> LV_IS = 1
then               T_DLY_TCO_SUB = 0
endif
if (1)             LV_IS = 1
then (1)
    if (2)         T_DLY_TCO_SUB < C_T_DLY_TCO_SUB_IS_1  or
                   T_DLY_TCO_SUB > C_T_DLY_TCO_SUB_IS_2
    if (3)   OPM_AV= S          (at operation mode stratified)
             then (3)
```

```
                    TCO_SUB_INC  =
                            IP_TCO_SUB_INC_S * IP_FAC_TCO_SUB_INC_S*
                            FAC_TCO_SUB_COR* FAC_TCO_SUB_COR_N_REL_CWP
            else (3)
                    TCO_SUB_INC  =
                    IP_TCO_SUB_INC * IP_FAC_TCO_SUB_INC * FAC_TCO_SUB_COR
                    * FAC_TCO_SUB_COR_N_REL_CWP
            endif (3)
        else (2)    TCO_SUB_INC  = IP_TCO_SUB_INC_IS
        endif (2)
                    T_DLY_TCO_SUB = T_DLY_TCO_SUB + 1 s
endif (1)
```

Correction of the coolant temperature substitute model value during N_REL_CWP increasing:

Gradient of N_REL_CWP:

```
N_REL_CWP_GRD = N_REL_CWP - N_REL_CWP_OLD
N_REL_CWP_OLD = N_REL_CWP
```

TCO_SUB correction if N_REL_CWP is active

```
If          N_REL_CWP_GRD > C_N_REL_CWP_THD_GRD_H_MIN
    AND N_REL_CWP > C_N_REL_CWP_THD_H_MIN
    AND CTR_N_REL_CWP_H > 0
    AND T_N_REL_CWP_GRD_H = 0               (timer is finished)

 then
      T_N_REL_CWP_GRD_H = C_T_N_REL_CWP_GRD_H     (initialize timer)
      CTR_N_REL_CWP_H = CTR_N_REL_CWP_H -1            (reduce counter)
      TCO_SUB = TCO_SUB + C_TCO_SUB_COR_H_ON;       (reduce TCO_SUB)

 endif


if      T_N_REL_CWP_GRD_H > 0
 then
```

```
        T_N_REL_CWP_GRD_H = T_N_REL_CWP_GRD_H -1 (timer)
endif
```

Calculation of the coolant temperature substitute model value:

```
If (1)          LV_ERR_TCO = 0
then (1)        FAC_TCO_SUB_COR = 1
else (1)        FAC_TCO_SUB_COR = C_FAC_TCO_SUB_COR
endif (1)

If (1)          T_AST < T_DLY_THD_TCO_SUB_ST
then (1)        $TCO\_SUB_{(n)} = TCO\_SUB_{(n-1)}$
else (1)
    if (2)      Transition: LV_ERR_TCO = 0 -> LV_ERR_TCO =1
    then (2)    $TCO\_SUB_{(n)} = TCO_{(n-1)}$
                (re-initialization of TCO_SUB with the last plausible TCO value)
    else (2)
        if(3)       $TCO\_SUB_{(n-1)}$ <= IP_TCO_SUB_MAX_TAM        or
                    TCO_SUB_INC < 0
        then(3)     $TCO\_SUB_{(n)} = TCO\_SUB_{(n-1)} + TCO\_SUB\_INC$
        else(3)     $TCO\_SUB_{(n)} = TCO\_SUB_{(n-1)}$
        endif(3)
    endif (2)
endif (1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TCO_SUB_INC	6*6	0...FE00H	0...190.5	2.9297e-3	[°C]
LDPM_MAF_KGH_IP_TCO_SUB	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDPM_TAM_IP_TCO_SUB	6	0...FEH	-48...142.5	0.75	[°C]
Increment value for the coolant temperature substitute value calculation					
IP_FAC_TCO_SUB_INC	6*6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_MAF_KGH_IP_TCO_SUB	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDPM_TCO_SUB_IP_TCO_SUB	6	0...FE00H	-48...142.5	2.9297e-3	[°C]
Increment factor for the coolant temperature substitute value calculation					
IP_TCO_SUB_DEC	6*6	0...FE00H	0...190.5	2.9297e-3	[°C]
LDPM_TCO_SUB_IP_TCO_SUB	6	0...FE00H	-48...142.5	2.9297e-3	[°C]
LDPM_TAM_IP_TCO_SUB	6	0...FEH	-48...142.5	0.75	[°C]
Decrement value for the coolant temperature substitute value calculation					
IP_FAC_TCO_SUB_COR_N_REL_CWP	4	0...FFH	0...4.98046	0.0195313	[-]
LDP_N_REL_CWP_IP_FAC_TCO_SUB	4	0...FFH	0...255	1	[-]
Coolant temperature substitute value correction factor dependend on CWP speed					
IP_FAC_TCO_SUB_VS	6*6	0...FFFFH	-8...7.99975	0.2441e-3	[-]
LDP_VS_IP_FAC_TCO_SUB_VS	6	0...FFH	0...255	1	[km/h]
LDPM_MAF_KGH_IP_TCO_SUB	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Coolant temperature substitute value correction factor dependend on vehicle speed and engine load					
IP_TCO_SUB_MAX_TAM	6	0...FEH	-48...142.5	0.75	[°C]
LDPM_TAM_IP_TCO_SUB	6	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature substitute maximum value dependend on ambient air temperature					
C_T_DLY_TCO_SUB_PUC	1	0...FFFFH	0...65535	1	[s]
Delay time after pull fuel cutoff to activate the coolant temperature substitute value calculation					
C_T_DLY_TCO_SUB_IS_1	1	0...FFFFH	0...65535	1	[s]
Delay time after idle speed to stop the coolant temperature substitute value calculation					
C_T_DLY_TCO_SUB_IS_2	1	0...FFFFH	0...65535	1	[s]
Delay time after idle speed to activate the coolant temperature substitute value calculation					
C_FAC_TCO_SUB_COR	1	0...FFH	0...1.99218	0.0078125	[-]
Multiple Coolant temperature substitute value correction factor					
C_T_DLY_TCO_SUB_COR_STATE_COC	1	0...FFFFH	0...65535	1	[s]
Delay time after warm-up for coolant temperature substitute value correction factor					
C_FAC_TCO_SUB_COR_STATE_COC	1	0...FFH	0...51	0.2	[-]
Coolant temperature substitute value correction factor after warm-up					
C_T_DLY_TCO_SUB_ST_CWP_ON	1	0...FFFFH	0...6553.5	0.1	[s]
Delay time after start to activate the coolant temperature substitute value calculation for running CWP					
C_T_DLY_TCO_SUB_ST_CWP_OFF	1	0...FFFFH	0...6553.5	0.1	[s]
Delay time after start to activate the coolant temperature substitute value calculation for stopped CWP					
C_N_REL_CWP_THD_COC	1	0...FFH	0...255	1	[-]
CWP speed threshold for detecting running or stopped CWP					
C_TCO_SUB_MIN_FAC_VS	1	0...FEH	-48...142.5	0.75	[°C]
Minimum model temperature to activate the vehicle speed correction factor					
IP_TCO_SUB_INC_IS	6*6	0...FE00H	-48...142.5	2.9297e-3	[°C]
LDP_TCO_SUB_IP_TCO_SUB_INC	6	0...FE00H	-48...142.5	2.9297e-3	[°C]
LDP_TAM_IP_TCO_SUB_INC	6	0...FEH	-48...142.5	0.75	[°C]
Decrement value for the coolant temperature substitute value calculation					
C_N_REL_CWP_THD_GRD_H_MIN	1	8000...7FFFH	-32768...32767	1	[-]
Threshold to el. cooling water pump to correct TCO _ SUB model					
C_N_REL_CWP_THD_H_MIN	1	0...FFH	0...255	1	[-]
Minimum cooling water pump speed for correcting coolant thermostat diagnosis					
C_T_N_REL_CWP_GRD_H	1	0...FFH	0...255	1	[s]
Threshold to initialize timer to block TCO-SUB-Model correction					
C_TCO_SUB_COR_H_ON	1	0...FEH	-48...142.5	0.75	[°C]
Correction TCO-SUB-model if el. waterpump begins to run					
C_CTR_N_REL_CWP_H	1	0...FFH	0...255	1	[-]
counter to correct TCO-SUB model via colling water pump gradient					
IP_TCO_SUB_INC_S	4*4	0...FE00H	0...190.5	2.9297e-3	[°C]

LDPM_MAF_KGH_IP_TCO_SUB_S	4	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDP_TAM_IP_TCO_SUB_S	4	0...FEH	-48...142.5	0.75	[°C]
Increment value for the coolant temperature substitute value calculation at operation mode stratified					
IP_FAC_TCO_SUB_INC_S	4*4	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_MAF_KGH_IP_TCO_SUB_S	4	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDP_TCO_SUB_IP_TCO_SUB_S	4	0...FE00H	-48...142.5	2.9297e-3	[°C]
Increment factor for the coolant temperature substitute value calculation at operation mode stratified					

26.10 Control of the electric fan

26.10.1 Control of ECFPWM[NC_ECF_NR] with ignition key on

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ECFPWM[NC_ECF_NR]	V/O	0...FFH	0...99.60937	0.390625	[%]
Pulse width modulated output signal for the electronic controlled cooling fan(s)					
T_N_PERC_ECF_ACT	V	0...FFH	0...51	0.2	[s]
Time condition for electric fan during self diagnostic					

Input data:

N_PERC_ECF	LV_IGK	LV_ECFPWM_ECF_EXT_ADJ	ECFPWM_ECF_EXT_ADJ
LV_ERR_ECF[NC_ECF_NR]	NC_ECF_NR		

FUNCTION DESCRIPTION:

General information:

The electric fan is controlled with a 100 Hz PWM signal to regulate its rotational speed. The calculated pulse width modulation depends on various components which are handled by the BMW “Wärmemanagementkoordinator”.

Application conditions:

Initialisation: all = 0 at reset

T_N_PERC_ECF_ACT = C_T_N_PERC_ECF_ACT

Recurrence: 200ms

Activation: LV_IGK = 1

(see: “*ENTE scheduler*”)

26.10.1.1 Calculation of PWM-signal for the electric cooling fan

Description:

All calculations which are necessary for the computation of the overall cooling fan speed are done by the BMW “Wärmemanagementkoordinator”. Via layer BMW hands over the final cooling fan speed N_PERC_ECF which is already standardized to percentage.

Self-diagnosis runs each time the electric fan is switched on. The electric fan should be operated with constant, low speed stage for this.

```
If      N_PERC_ECF  >  C_N_PERC_ACT_ECF  and
        T_N_PERC_ECF_ACT > 0 sec.
Then    T_N_PERC_ECF_ACT is decremented
        ECFPWM[0]  =  C_N_PERC_ACT_ECF
Else    ECFPWM[0]  =  N_PERC_ECF
Endif
```

Formula section:

```
If (1)      LV_ERR_ECF[0] = 1
then (1)    ECFPWM[0] = NC_ECFPWM_ECF_SUB_DIAG
else (1)
    if (2)      LV_ECFPWM_ECF_EXT_ADJ = 1
    then (2)    ECFPWM[0] = ECFPWM_ECF_EXT_ADJ
    else (2)
        if (3)      LC_ECFPWM_ECF_MAN = 1
        then (3)    ECFPWM[0] = C_ECFPWM_ECF_MAN
        endif (3)
    endif (2)
endif (1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_N_PERC_ECF_ACT	1	0...FF	0...51	0.2	[s]
Time duration for electric fan during self diagnosis					
C_N_PERC_ACT_ECF	1	0...FFH	0...99.60937	0.390625	[%]
Set value for N_PERC_ECF					
C_ECFPWM_ACT_ECF	1	0...FFH	0...99.60937	0.390625	[%]
Min limitation ECFPWM_ECF from speed stage C_N_PERC_ACT_ECF					
LC_ECFPWM_ECF_MAN	1	0...1H	0...1	1	[-]
Constant to switch to manual ECFPWM_ECF activation					
C_ECFPWM_ECF_MAN	1	0...FFH	0...99.60937	0.390625	[%]
Setpoint of ECFPWM_ECF; manual control					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_ECFPWM_ECF_SUB_DIAG	1	0...FFH	0...99.60937	0.390625	[%]
Configuration constant for ECFPWM in case of active failure					

Default-value for NC_ECFPWM_ECF_SUB_DIAG = 2.37%

26.10.2 Control with ignition key off up to detection of ES and in the self-holding phase

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ECFPWM_IGK_OFF	V/O	0...FFH	0...99.60937	0.390625	[%]
Electric fan pulse width modulation in the self-holding phase					

Input data:

N_PERC_ECF	ECU_STATE	T_PWL	

General information:

With ignition key off until ES and in the self-holding phase of the control unit, the control frequency is switched over to 10Hz and a pulse width modulation ECFPWM_IGK_OFF is set. The power latch time is defined in Chapter Basic Software Inputs and Outputs.

Application conditions:

Recurrence: 100 ms

Activation: ECU_STATE = PWL

```
If      T_PWL < C_T_ECFPWM_PWL
then    ECFPWM_IGK_OFF = N_PERC_ECF
else    ECFPWM_IGK_OFF = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_ECFPWM_PWL	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum time ECFPWM is sent at PWL					

26.11 Radiator Shutter (RAS) controlled by a magnetic clamp

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_RAS	V	0...1H	0...1	1	[-]
Bit used to determine if the radiator shutter should be open or closed (1 = shutter open, power off for the magnetic clamp; 0 = shutter closed, power on for the magnetic clamp)					
LV_RAS_OUT	O/V	0...1H	0...1	1	[-]
Bit that is sent to the PIN that controls the magnetic clamp for the radiator shutter					

Input data:

LV_RAS_EXT_ADJ	LV_ACT_RAS_EXT_ADJ	LV_ECRAS_UP	

Import actions:

ACTION_INFR_SetRas (IN <Lv_ras_out>)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_RAS_ACT	1	0...1H	0...1	1	[-]
Manual switch to activate or deactivate the functionality for the Radiator Shutter (RAS)					

FUNCTION DESCRIPTION:

Description:

This module contains the functionality for the control of the radiator shutter via a magnetic clamp. The output LV_RAS_OUT is sent to the PIN that controls the magnetic clamp for the radiator shutter.

Application conditions:

Initialisation: at reset all 0

Recurrence: 100 ms

Activation: LC_RAS_ACT = 1

Deactivation: LC_RAS_ACT = 0

Formula section:

```
if      LV_ECRAS_UP = 1
then    LV_RAS  = 1
else    LV_RAS = 0
endif

if      LV_RAS_EXT_ADJ = 1
then    LV_RAS_OUT = LV_ACT_RAS_EXT_ADJ
else    LV_RAS_OUT = LV_RAS
endif


ACTION_INFR_SetRas(LV_RAS_OUT)
```

26.12 Electronically controlled thermostat

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ECTPWM	V/O	0...FFFFH	0...99.99847	1.53E-03	[%]
Pulse width modulated output signal for the electronic controlled thermostat					
T_ECTPWM_AST	-	0...FFH	0...255	1	[s]
time for ECTPWM-calculation after start					

Input data:

LV_ECTPWM_EXT_ADJ	ECTPWM_EXT_ADJ	LV_ERR_ECT_EL	ECTPWM_CLC
LV_ES	LV_ST_END	TCO	LV_ERR_ECT_EL_OC
LV_ERR_ECT_EL_SCP			

26.12.1 Adjustment of the pulse width modulated output signal (ECTPWM)

FUNCTION DESCRIPTION:

General information:

For calibration issues and service tool operations the calculated pulse width modulated signal (ECTPWM_CLC) has to be adjusted manually before it is sent to the power stage of the electronic controlled thermostat (ECTPWM). Two possibilities for the manual adjustment are existing. With the logical constant LC_ECTPWM_MAN it is possible to adjust the calibratable value C_ECTPWM_MAN to the PWM output with use of the calibration system. The service tool allows the adjustment of the PWM signal as well. In this case the logical variable LV_ECTPWM_EXT_ADJ is set from logical low (0) to logical high (1) and the adjustable value ECTPWM_EXT_ADJ is used as output signal (ECTPWM).

As soon as the thermostat failure algorithm detects a electrical thermostat failure, the PWM output signal is set to a calibratable value (NC_ECTPWM_SUB_DIAG) at once and the ECT control state “LIH” is activated.

Application conditions:

Initialisation: all 0 at reset,

Except: T_ECTPWM_AST :
at reset or at transition LV_ES 0 --> 1: T_ECTPWM_AST = 0

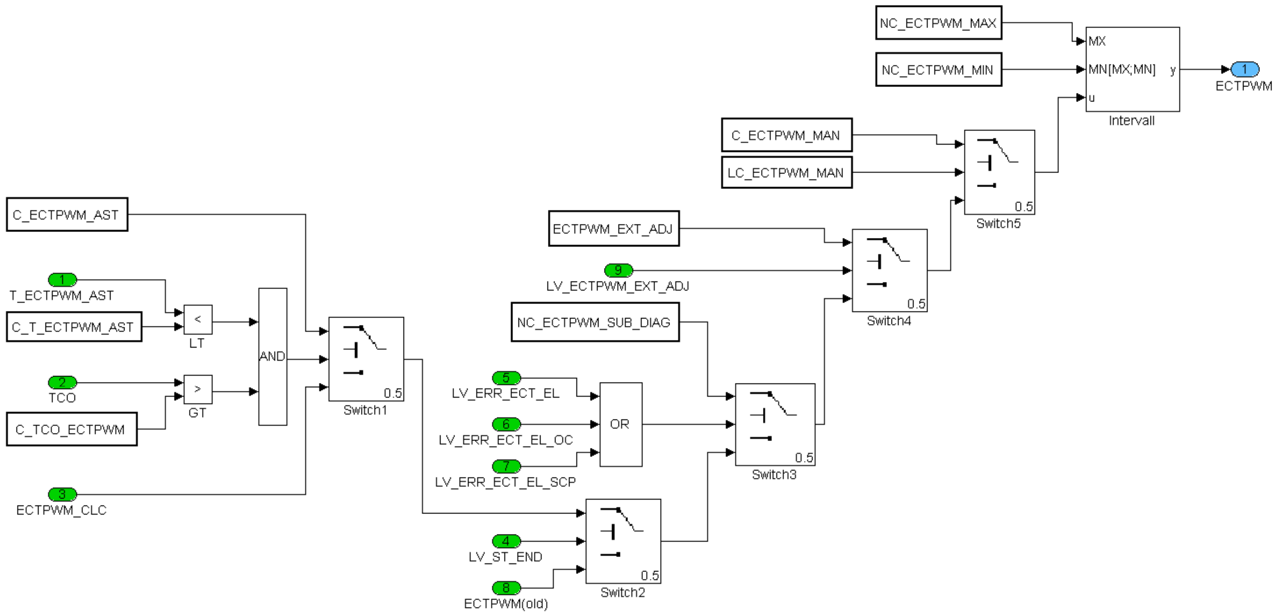
// the time condition C_T_ECTPWM_AST is reset each time
the engine state “engine stop” is reached or ecu is reset

Recurrence: 1000 ms

Activation: at every engine operating state

Deactivation: -

Signal flow diagram:



Formula section:

If LV_ST_END = 1 **and** T_ECTPWM_AST < C_T_ECTPWM_AST

Then T_ECTPWM_AST = ++

Endif

```
If (1)      LC_ECTPWM_MAN = 1
Then (1)    ECTPWM = C_ECTPWM_MAN
Else (1)    If (2)   LV_ECTPWM_EXT_ADJ = 1
            Then(2)ECTPWM = ECTPWM_EXT_ADJ
            Else (2)If(3)   LV_ERR_ECT_EL = 1          OR
                            LV_ERR_ECT_EL_OC = 1       OR
                            LV_ERR_ECT_EL_SCP = 1
                  Then(3)ECTPWM = NC_ECTPWM_SUB_DIAG
                  Else(3)If(4)          LV_ST_END = 1
                        Then(4)
                              If(5)     0 <= T_ECTPWM_AST < C_T_ECTPWM_AST
                                        // time C_T_ECTPWM_AST has not elapsed
                                        and
                                        TCO < C_TCO_ECTPWM
                              Then(5)         ECTPWM = C_ECTPWM_AST
                                              // for   time C_T_ECTPWM_AST
                              Else (5)  ECTPWM = ECTPWM_CLC
                              Endif (5)
                        Endif(4)
                  Endif(3)
            Endif (2)
Endif (1)
```

26.12.2 Limitation of the pulse width modulated output signal (ECTPWM)

FUNCTION DESCRIPTION:

General information:

The limits of the pulse width modulated output signal for the electronic controlled thermostat are fixed to NC_ECTPWM_MIN for the lower and NC_ECTPWM_MAX for the upper value. The controller output signal must be limited as soon as the amount of the controller is reaching the limits.

The power stage output signal for the electronic controlled thermostat is limited to a minimum and a maximum value to guarantee a continuous working electrical diagnosis of the power stage.

Application conditions:

Initialisation: all 0 at reset

Recurrence: 1000 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

$NC\_ECTPWM\_MIN \leq ECTPWM \leq NC\_ECTPWM\_MAX$

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ECTPWM_MAN	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Manual adjustable PWM output signal for the electronic controlled thermostat					
LC_ECTPWM_MAN	1	0...1H	0...1	1	[-]
Configuration bit for activation of manual PWM output signal adjustment					
C_T_ECTPWM_AST	1	0...FFH	0...255	1	[s]
time for ECTPWM-calculation after start					
C_ECTPWM_AST	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
calibratable ECTPWM after start					
C_TCO_ECTPWM	1	0...FEH	-48...142.5	0.75	[°C]
calibratable TCO-threshold for ECTPWM after start					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_ECTPWM_MAX	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Maximum PWM output signal for the electronic controlled thermostat					
NC_ECTPWM_MIN	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Minimum PWM output signal for the electronic controlled thermostat					
NC_ECTPWM_SUB_DIAG	1	0...FFFFH	0...99.99847	1.53E-03	[%]
PWM output signal for the electronic controlled thermostat in case of a failure					

26.13 ENTE Manager 01 – project specific tasks

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
POW_OUT_ECF	O/V	0…FFH	0…99.61	0.39	%
Cooling fan power output of all available fan(s) at the vehicle					
LV_ERR_ECF[NC_ECF_NR]	O/V	0...1H	0...1	1	-
Boolean for electronic controlled cooling fan error					

Input data:

LV_ERR_ECF_EL[NC_ECF_NR]	ECFPWM[NC_ECF_NR]	NC_ECF_NR	NC_ECF_CONF
ECFPWM_IGK_OFF			

FUNCTION DESCRIPTION:

General information:

This manager specifies the sequencing of project specific ENTE tasks.

Description:

Within the aggregate ENTE, the cooling fan functions (acquisition, control, diagnosis, torque loss) are handled project specific with use of HOOK modules. Because different projects use different update rates for the functions, it is necessary to handle the function calls specific within this module.

Application conditions:

Recurrence: see formula section below

Activation: at every engine operating state

Deactivation: -

Formula section:

#IF NC_ECF_CONF = 0 **(** control of cooling fan(s) only by Relay(s) **)**

Tasks
No action

#ENDIF

#IF NC_ECF_CONF = 1 **(** control of cooling fan(s) only by PWM(s) **)**

RESET - Task
POW_OUT_ECF = 0% LV_ERR_ECF[NC_ECF_NR] = 0
10ms - Task
ENTE_ACCTLECT1 (Electronically controlled Thermostat (Appl. Inc.) – “*Conversion from ECTPWM to LV_ECTPWM_STAT*” chapter 1.1.1)
100ms - Task
if (LV_IGK = 0) **then** ENTE_ACCTLECF1 (Cooling and condenser fan control – “*Control with ignition key off up to detection of ES and in the self-holding phase*” chapter 1.2)
if (LV_IGK = 1 *AND* LV_ES = 0) **then** ENTE_REQGNTQECF1 (Torque Loss - Electronic cooling fan PWM)

200ms - Task
if (LV_IGK = 1) **then** ENTE_ACCTLECF1 (Cooling and condenser fan control – *“Control of ECFPWM[NC_ECF_NR] with ignition key on*” chapter 1.1)
ENTE_IFINFECF1 (Acquisition of electronically controlled cooling fan signal) Note: When using ISR-action, this spec does not need to appear in the calculation order!

1000ms - Task
if (LV_ES = 0) **then** ENTE_ACCTLECF1 (Cooling and condenser fan control – *“Calculation of TEMP_MMV_SUB_CAT_ECF*” chapter 1.1.2)
if (LV_IGK = 1) **then** POW_OUT_ECF = ECFPWM[1] **if** (LV_IGK = 0) **then** POW_OUT_ECF = ECFPWM_IGK_OFF
ENTE_OUTDGECF1 (Cooling fan diagnosis)
LV_ERR_ECF[NC_ECF_NR] = LV_ERR_ECF_EL[NC_ECF_NR]

#ENDIF

#IF NC_ECF_CONF = 2 **(** control of cooling fan(s) by Relay(s) **or** PWM(s) **)**

Tasks
No action

#ENDIF

26.14 Electronically controlled radiator shutter (ECRAS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ECRASPWM	V/O	0...FFH	0...99.60937	0.390625	[%]
PWM for ectrical radiator shutter					
STATE_ECRAS_SYS	V/O	0...FFFFH	0...65535	1	[-]
State of ECRAS system					
LV_ECRAS_UP_1	V/O	0...1H	0...1	1	[-]
Bit used to determine ECRASPWM (input for ID)					
LV_ECRAS_DOWN_1	V/O	0...1H	0...1	1	[-]
Bit used to determine ECRASPWM (input for ID)					
T_ECRAS_UP_DET	V	0...FFH	0...25.5	0.1	[-]
Timer indicating time until learning stopped (if no variant recognised)					
LV_ECRAS_EOL	V	0...1H	0...1	1	[-]
Bit for determining ECRASPWM during ECRAS system test (EOL)					
LV_ECRAS_EOL_END	V	0...1H	0...1	1	[-]
Bit indicating finished ECRAS system test (EOL)					
T_WAIT_ECRAS_EOL	V	0...FFH	0...25.5	0.1	[s]
Timer for switch ECRAS control output					
CTR_ECRAS_EOL	V	0...FFH	0...255	1	[-]
Counter for stopping ECRAS system test (EOL)					
LV_ECRAS_EOL_INH	V/O	0...1H	0...1	1	[-]
Bit indicating inhibition or ECRAS EOL (if = 1 EOL stopped)					
LV_ERR_ECRAS_EOL	V/O	0...1H	0...1	1	[-]
Bit indicating ECRAS error during EOL ECRAS system test					

Input data:

LV_ECRAS_UP	LV_ECRAS_DOWN	LV_ECRAS_UP_EXT_ADJ	LV_ECRAS_DOWN_EXT_ADJ
LV_ACT_ECRAS_UP_EXT_ADJ	LV_ACT_ECRAS_DOWN_EXT_ADJ	LV_IGK	LV_CDN_VB_MIN_DIAG
LV_ECRAS_DOWN_DET_ACT	LV_DIAG_ECRAS	LV_VAR_ECRAS_UP	LV_VAR_ECRAS_DOWN
LV_DIAG_ECRAS_END	LV_END_DIAG_ECRAS_UP_FB	LV_END_DIAG_ECRAS_DOWN_FB	LV_END_DIAG_ECRAS_EL
LV_ERR_ECRAS_UP_FB	LV_ERR_ECRAS_DOWN_FB	LV_ERR_ECRAS_EL	LV_ERR_ECRAS
LC_AD_CLR_VAR	LV_ACT_ECRAS_EOL_EXT_ADJ	LV_DIAG_ECRAS_ERR_END	LV_ECRAS_DOWN_DET_SET

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ECRASPWM_10	1	0...FFH	0...99.60937	0.390625	[%]
ECRASPWM for diagnosis mode of ECRAS control unit (10% PWM)					
C_ECRASPWM_90	1	0...FFH	0...99.60937	0.390625	[%]
ECRASPWM for learning of ECRAS_DOWN (90% PWM)					
C_ECRASPWM_MAN	1	0...FFH	0...99.60937	0.390625	[%]
Manually given ECRASPWM					
LC_ECRASPWM_MAN	1	0...1H	0...1	1	[-]
Switch for manual setting of ECRASPWM					
ID_ECRASPWM	2*2	0...FFH	0...99.60937	0.390625	[%]
LDP_LV_ECRAS_UP_1	2	0...1H	0...1	1	[-]
LDP_LV_ECRAS_DOWN_1	2	0...1H	0...1	1	[-]
Map for determining ECRASPWM					

C_ECRASPWM_DIAG	1	0...FFH	0...99.60937	0.390625	[%]
PWM for radiator shutter for diagnosis readyness					
C_T_ECRAS_UP_DET	1	0...FFH	0...25.5	0.1	[s]
Timer threshold for learning ECRAS_UP					
C_CTR_ECRAS_EOL	1	0...FFH	0...255	1	[-]
Counter threshold for stopping ECRAS system test (EOL)					
C_T_WAIT_ECRAS_EOL	1	0...FFH	0...25.5	0.1	[s]
Timer threshold for switch ECRAS control output					

FUNCTION DESCRIPTION:

Description:

This modul consists of three parts.

- End of line Sytem test functionality
- Calculation of ECRASPWM
- Setting of STATE_ECRAS_SYS

EOL-System test: the system test is activated by LV_ACT_ECRAS_EOL_EXT_ADJ which is set by tester (see "Services for EOL tests" chapter). During this EOL test several ECRASPWM will be set for a certain time. This allows that every possible failure can be detected by the ECRAS feedback diagnosis.
Calculation of ECRASPWM: The ECRAS PWM signal will be claculated. ECRASPWM requests will be prioritised.
Setting of STATE_ECRAS_SYS: A State STATE_ECRAS_SYS is created which displays specific information about the whole ECRAS system (eg. diagnosis, EOL etc.).

Application conditions:

Initialisation: reset **or** LV_IGK = 0 -> 1 all 0, except ECRASPWM = C_ECRASPWM_DIAG

Recurrence: 100 ms

Activation: always

Deactivation: none

Formula section:

EOL System test

```
if    LV_ACT_ECRAS_EOL_EXT_ADJ = 1
then          // if activation by tester request recognised reset values
              if      LV_ACT_ECRAS_EOL_EXT_ADJ  0->1
              then    LV_ECRAS_EOL_END = 0
                      T_WAIT_ECRAS_EOL = 0
                      CTR_ECRAS_EOL = 0
                      LV_ECRAS_EOL = 0
                      LV_ERR_ECRAS_EOL = 0
              endif

              // increment timer
              if      LV_DIAG_ECRAS_END = 1                   and
```

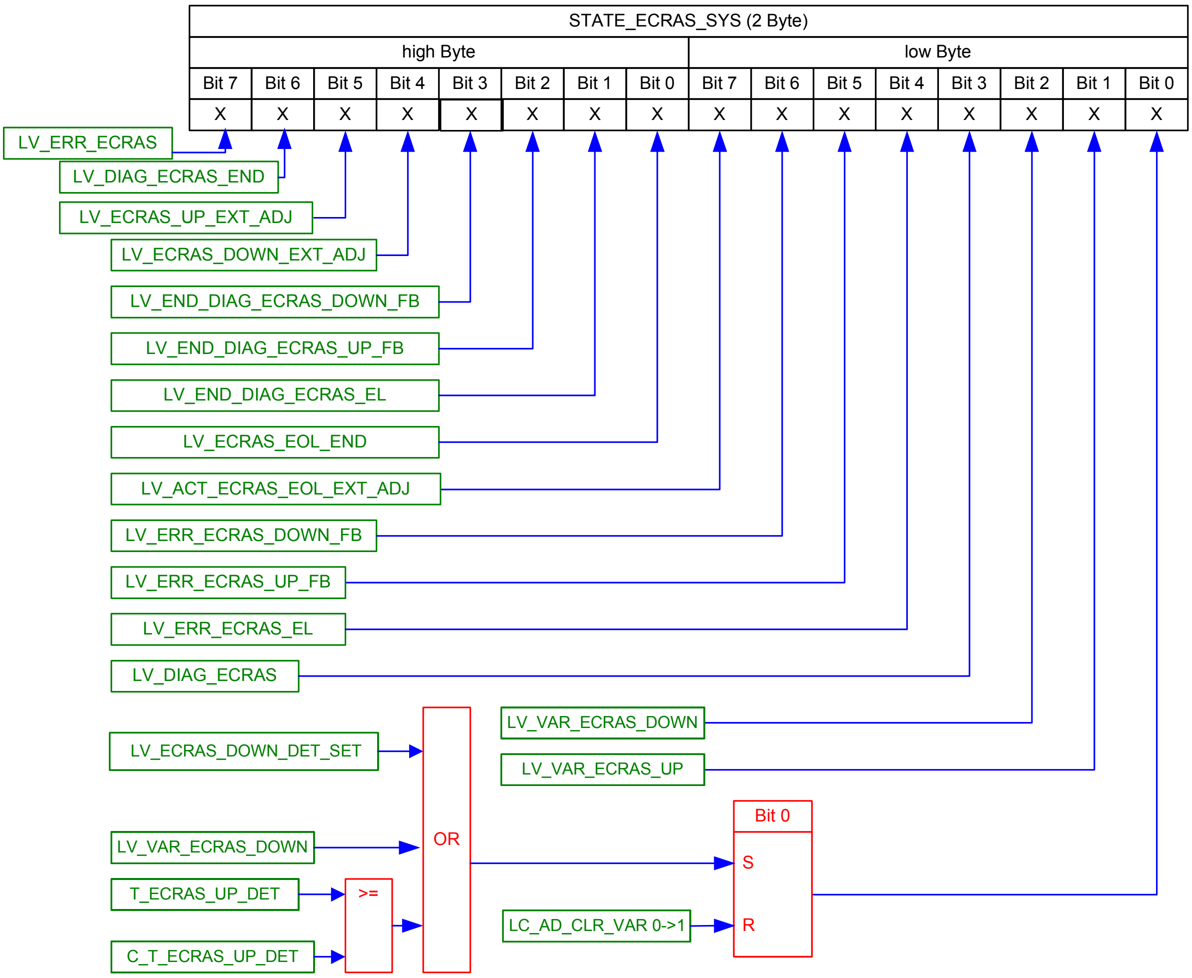
```
                    LV_ECRAS_DOWN_DET_ACT = 0
            then    T_WAIT_ECRAS_EOL = T_WAIT_ECRAS_EOL + 100ms
            endif

            // toggle control of ECRAS flaps (see also calculation of ECRASPWM to enable
            diagnosis
            if      T_WAIT_ECRAS_EOL >= C_T_WAIT_ECRAS_EOL
            then    if      LV_ECRAS_EOL = 0
                    then    LV_ECRAS_EOL = 1
                    else    LV_ECRAS_EOL = 0
                    endif
                    T_WAIT_ECRAS_EOL = 0
                    CTR_ECRAS_EOL = CTR_ECRAS_EOL + 1
            endif

            // set end of EOL system test
            if      CTR_ECRAS_EOL>=C_CTR_ECRAS_EOL          or
                    LV_DIAG_ECRAS_ERR_END = 1
            then    LV_ECRAS_EOL_INH = 1
                    LV_ECRAS_EOL_END = 1
            endif

            // set LV_ERR_ECRAS_EOL if during EOL error is occurred or present
            if      LV_ERR_ECRAS = 1
            then    LV_ERR_ECRAS_EOL = 1
            endif
else
            LV_ECRAS_EOL_INH = 0
endif
```

Calculation of ECRASPWM

```
if          LV_VAR_ECRAS_UP = 1
then        // priorisation of ECRASPWM requests
            if      LC_ECRASPWM_MAN = 1
            then    ECRASPWM = C_ECRASPWM_MAN
            elseif  LV_ECRAS_DOWN_DET_ACT = 1
            then    ECRASPWM = C_ECRASPWM_90
            elseif  LV_DIAG_ECRAS = 1
            then    ECRASPWM = C_ECRASPWM_10
            else    if      LV_ACT_ECRAS_EOL_EXT_ADJ = 1
                    then    LV_ECRAS_UP_1 = LV_ECRAS_EOL
                            LV_ECRAS_DOWN_1 = LV_ECRAS_EOL
                    else    if      LV_ECRAS_UP_EXT_ADJ = 1
                            then    LV_ECRAS_UP_1 = LV_ACT_ECRAS_UP_EXT_ADJ
                            else    LV_ECRAS_UP_1 = LV_ECRAS_UP
                            endif
                            if      LV_ECRAS_DOWN_EXT_ADJ = 1
                            then    LV_ECRAS_DOWN_1=
                                                    LV_ACT_ECRAS_DOWN_EXT_ADJ
                            else    if      LV_VAR_ECRAS_DOWN = 1
```

```
                                        then   LV_ECRAS_DOWN_1 = LV_ECRAS_DOWN
                                        else   LV_ECRAS_DOWN_1 = 0
                                        endif
                              endif
                    endif
                    ECRASPWM = ID_ECRASPWM
          endif
else
          if        LV_CDN_VB_MIN_DIAG = 1
          then      T_ECRAS_UP_DET = T_ECRAS_UP_DET + 100ms
          else      freeze timer
          endif
          if        T_ECRAS_UP_DET >= C_T_ECRAS_UP_DET
          then      ECRASPWM = 0
          else      // waiting, learning of LV_VAR_ECRAS_UP still possible
                    ECRASPWM = C_ECRASPWM_DIAG
          endif
endif
```

Setting of STATE_ECRAS_SYS


STATE_ECRAS_SYS (2 Byte)
high Byte
low Byte
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit 0
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit 0
X
LV_ERR_ECRAS
LV_DIAG_ECRAS_END
LV_ECRAS_UP_EXT_ADJ
LV_ECRAS_DOWN_EXT_ADJ
LV_END_DIAG_ECRAS_DOWN_FB
LV_END_DIAG_ECRAS_UP_FB
LV_END_DIAG_ECRAS_EL
LV_ECRAS_EOL_END
LV_ACT_ECRAS_EOL_EXT_ADJ
LV_ERR_ECRAS_DOWN_FB
LV_ERR_ECRAS_UP_FB
LV_ERR_ECRAS_EL
LV_DIAG_ECRAS
LV_VAR_ECRAS_DOWN
LV_VAR_ECRAS_UP
LV_ECRAS_DOWN_DET_SET
OR
LV_VAR_ECRAS_DOWN
T_ECRAS_UP_DET
>=
C_T_ECRAS_UP_DET
Bit 0
S
R
LC_AD_CLR_VAR 0->1

26.15 Coolant temperature diagnosis interface

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_DIAG_AST	O/V	0...FFFFH	0...65535	1	s
Time after engine start for diagnosis functions					
T_IS_AST	V	0...FFFFH	0...65535	1	s
“Idle speed” activation time since engine start					
RATIO_T_IS_AST	O/V	0...FFH	0...99.609	0.3906	%
“Idle speed” activation time in percent since engine start					
T_PUC_AST	V	0...FFFFH	0...65535	1	s
"Trailing throttle fuel cut-off" activation time since engine start					
RATIO_T_PUC_AST	O/V	0...FFH	0...99.609	0.3906	%
"Trailing throttle fuel cut-off" activation time in percent since engine start					
T_LOAD_MIN_AST	V	0...FFFFH	0...65535	1	s
"Minimum load" time since engine start					
RATIO_T_LOAD_MIN_AST	O/V	0...FFH	0...99.609	0.3906	%
"Minimum load" time in percent since engine start					
T_VS_MIN_AST	V	0...FFFFH	0...65535	1	s
"Minimum vehicle speed " time since engine start					
RATIO_T_VS_MIN_AST	O/V	0...FFH	0...99.609	0.3906	%
"Minimum vehicle speed " time in percent since engine start					
T_VS_MAX_AST	V	0...FFFFH	0...65535	1	s
"Maximum vehicle speed " time since engine start					
RATIO_T_VS_MAX_AST	O/V	0...FFH	0...99.609	0.3906	%
"Maximum vehicle speed " time in percent since engine start					
LV_CDN_ENA_RATIO_T_AST	V	0...1H	0...1	1	-
Boolean for ratio time after engine start enable condition					

Input data:

LV_IS	LV_PUC	MAF_KGH	VS
TAM			

FUNCTION DESCRIPTION:

General information:

The coolant temperature increase after engine start is most dependent on the ambient- and the engine operating conditions. To avoid monitoring failures within the diagnosis functions related to the coolant temperature behaviour, the influence of critical engine driving conditions must be taken into consideration. To allow a comparison to the time elapsed since engine start, the activation time of the critical driving conditions are calculated in percent of the elapsed time. The percentage values are used in several diagnosis functions to cancel monitoring in order to avoid failures.

Application conditions:

Initialisation at Exit start (EXIT_ST):

LV_CDN_ENA_RATIO_T_AST = 1

all other output data are set to “0” [phys]

Initialisation at Engine run to Engine stop (ERU_to_ES):

LV_CDN_ENA_RATIO_T_AST = 0

T_DIAG_AST = 0 s

Recurrence: 1000 ms

Activation: LV_CDN_ENA_RATIO_T_AST = 1

Deactivation: LV_CDN_ENA_RATIO_T_AST = 0

Formula section:

Calculation of the after start time for diagnosis functions:

```
If      $\text{T\_DIAG\_AST} < 65535\ \text{s}$
then    $\text{T\_DIAG\_AST}_{(n)} = \text{T\_DIAG\_AST}_{(n-1)} + 1\ \text{s}$
else    $\text{T\_DIAG\_AST}_{(n)} = \text{T\_DIAG\_AST}_{(n-1)}$
endif
```

Calculation of the "idle speed" period after start:

```
If      $\text{LV\_IS} = 1$
then    $\text{T\_IS\_AST}_{(n)} = \text{T\_IS\_AST}_{(n-1)} + 1\ \text{s}$
else    $\text{T\_IS\_AST}_{(n)} = \text{T\_IS\_AST}_{(n-1)}$
endif
$\text{RATIO\_T\_IS\_AST} = (\text{T\_IS\_AST} / \text{T\_DIAG\_AST}) * 100\ \%$
```

Calculation of the " trailing throttle fuel cut-off " period after start:

```
If      $\text{LV\_PUC} = 1$
then    $\text{T\_PUC\_AST}_{(n)} = \text{T\_PUC\_AST}_{(n-1)} + 1\ \text{s}$
else    $\text{T\_PUC\_AST}_{(n)} = \text{T\_PUC\_AST}_{(n-1)}$
endif
$\text{RATIO\_T\_PUC\_AST} = (\text{T\_PUC\_AST} / \text{T\_DIAG\_AST}) * 100\ \%$
```

Calculation of the "low load" period after start:

```
If      $\text{MAF\_KGH} <= \text{IP\_MAF\_KGH\_LOAD\_MIN\_AST}$
then    $\text{T\_LOAD\_MIN\_AST}_{(n)} = \text{T\_LOAD\_MIN\_AST}_{(n-1)} + 1\ \text{s}$
else    $\text{T\_LOAD\_MIN\_AST}_{(n)} = \text{T\_LOAD\_MIN\_AST}_{(n-1)}$
```

```
endif
RATIO_T_LOAD_MIN_AST = (T_LOAD_MIN_AST / T_DIAG_AST) * 100 %
```

Calculation of the "low vehicle speed" period after start:

```
If      VS <= C_VS_THD_VS_MIN_AST
then    $T\_VS\_MIN\_AST_{(n)} = T\_VS\_MIN\_AST_{(n-1)} + 1\ s$
else    $T\_VS\_MIN\_AST_{(n)} = T\_VS\_MIN\_AST_{(n-1)}$
endif
RATIO_T_VS_MIN_AST = (T_VS_MIN_AST / T_DIAG_AST) * 100 %
```

Calculation of the "high vehicle speed" period after start:

```
If      VS > C_VS_THD_VS_MAX_AST
then    $T\_VS\_MAX\_AST_{(n)} = T\_VS\_MAX\_AST_{(n-1)} + 1\ s$
else    $T\_VS\_MAX\_AST_{(n)} = T\_VS\_MAX\_AST_{(n-1)}$
endif
RATIO_T_VS_MAX_AST = (T_VS_MAX_AST / T_DIAG_AST) * 100 %
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_THD_VS_MIN_AST	1	0...FFH	0...255	1	km/h
Minimum vehicle speed threshold for “low vehicle speed” period calculation after start					
C_VS_THD_VS_MAX_AST	1	0...FFH	0...255	1	km/h
Maximum vehicle speed threshold for “high vehicle speed” period calculation after start					
IP_MAF_KGH_LOAD_MIN_AST	4	0...FFFFH	0...2047.96875	0.03125	kg/h
LDP_TAM_IP_MAF_KGH_LOAD_MIN_AST	4	0...FEH	-48...142.5	0.75	°C
Ambient temperature dependent minimum engine load threshold for “low load” period calculation after start					

26.16 Coolant temperature sensor diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TCO	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on coolant temperature acquisition					
LV_ERR_TCO_PREL	V/O	0...1H	0...1	1	[-]
Boolean for preliminary error currently present on coolant temperature acquisition					
LV_ERR_TCO_PREL_DET	V/S	0...1H	0...1	1	[-]
Boolean for preliminary error present on coolant temperature acquisition during driving cycle					
LV_CDN_ENA_TCO_PREL	V	0...1H	0...1	1	[-]
Boolean for preliminary coolant temperature error enable conditions					
LV_ERR_TCO_EL	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature signal range error					
ERR_SYM_TCO_EL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for coolant temperature signal range error					
LV_CDN_DIAG_TCO_EL	V	0...1H	0...1	1	[-]
Boolean for coolant temperature signal range diagnosis conditions					
LV_END_DIAG_TCO_EL	V/O	0...1H	0...1	1	[-]
Boolean for end of coolant temperature signal range diagnosis					
LV_ERR_TCO_GRD	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature signal gradient error					
ERR_SYM_TCO_GRD	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for coolant temperature signal gradient error					
LV_CDN_DIAG_TCO_GRD	V	0...1H	0...1	1	[-]
Boolean for coolant temperature signal gradient diagnosis conditions					
LV_END_DIAG_TCO_GRD	V/O	0...1H	0...1	1	[-]
Boolean for end of coolant temperature signal gradient diagnosis					

Input data:

TCO_MES	LV_ERR_TCO_STUCK	LV_ERR_TCO_STUCK_RNG	LV_ERR_TCO_PLAUS
TIA_THR	T_DIAG_AST	LV_IGK	LV_INH_DIAG_TCO_EL
LV_CDN_VB_MIN_DIAG	CTR_ABC_TCO_EL	CTR_ABC_TCO_GRD	LV_INH_DIAG_TCO_GRD
VP_TCO[NC_NR_TCO_SENS]	NC_NR_TCO_SENS		

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis function is to detect electrical failures as defined by OBD I and OBD II requirements. The coolant temperature raw sensor voltage (VP_TCO[0]) is measured with use of an A/D converter by the ECU hardware. The modified sensor signal is converted then into a measured temperature value (TCO_MES). Each by the ECU hardware measured raw sensor value is checked as well as the converted temperature signal.

Application conditions:

See separate chapters:

26.16.1 Global coolant temperature failure

FUNCTION DESCRIPTION:

General information:

A global coolant temperature error is set without debounce as soon as an electrical-, or signal gradient-, or stuck signal-, or plausibility- error is detected. All separate diagnosis functions are performed all the time if the diagnosis conditions are fulfilled. The deactivation of diagnosis functions with use of configuration switches is not possible.

In addition a preliminary coolant temperature error is set as soon as debouncing of an electrical-, or signal gradient- symptom occurs. The preliminary coolant temperature error will be stored in the non-volatile memory during the ECU’s self-holding phase to be available at the beginning of the next driving cycle.

Application conditions:

Initialisation at Reset:

LV_ERR_TCO = 0
LV_CDN_ENA_TCO_PREL = 1
LV_ERR_TCO_PREL = LV_ERR_TCO_PREL_DET *(NVMY)*
LV_ERR_TCO_PREL_DET = LV_ERR_TCO_PREL_DET *(NVMY)*
(NVMY: stored in the non-volatile memory)

Initialisation at Engine stop to Engine run (ES_to_ERU):

LV_ERR_TCO_PREL_DET = 0
LV_CDN_ENA_TCO_PREL = 0

Initialisation at Engine stop

LV_CDN_ENA_TCO_PREL = 1

Initialisation at FMY clear:

LV_ERR_TCO = 0
LV_ERR_TCO_PREL = 0
LV_ERR_TCO_PREL_DET = 0

Recurrence: 100 ms
Activation: at every engine operating state

Formula section:

If LV_ERR_TCO_EL = 1 **or**
LV_ERR_TCO_GRD = 1 **or**
LV_ERR_TCO_STUCK = 1 **or**
LV_ERR_TCO_STUCK_RNG = 1 **or**
LV_ERR_TCO_PLAUS = 1 **or**
then LV_ERR_TCO = 1 (without debounce)

```
else        LV_ERR_TCO = 0
endif
```

26.16.1.1 Preliminary coolant temperature failure at engine stop

Application conditions:

Recurrence: 100 ms

Activation: LV_CDN_ENA_TCO_PREL = 1

Deactivation: LV_CDN_ENA_TCO_PREL = 0

Formula section:

```
If          CTR_ABC_TCO_EL > 0            or
            CTR_ABC_TCO_GRD > 0           or
            LV_ERR_TCO_PREL_DET = 1
then        LV_ERR_TCO_PREL = 1           (without debounce)
else        LV_ERR_TCO_PREL = 0
endif
```

26.16.1.2 Preliminary coolant temperature failure at engine run

Application conditions:

Recurrence: 100 ms

Activation: LV_CDN_ENA_TCO_PREL = 0

Deactivation: LV_CDN_ENA_TCO_PREL = 1

Formula section:

```
If          CTR_ABC_TCO_EL         > C_CTR_ABC_THD_TCO_PREL   or
            CTR_ABC_TCO_GRD        > C_CTR_ABC_THD_TCO_PREL
then        LV_ERR_TCO_PREL_DET   = 1      (irrevocable for current driving cycle)
endif

If          CTR_ABC_TCO_EL         > 0            or
            CTR_ABC_TCO_GRD        > 0            or
then        LV_ERR_TCO_PREL        = 1            (without debounce)
else        LV_ERR_TCO_PREL        = 0
endif
```

26.16.2 Coolant temperature signal range diagnosis (TCO_EL)

FUNCTION DESCRIPTION:

General information:

The purpose of the signal range- or electrical diagnosis is to detect electrical errors in the input circuit of the coolant temperature sensor. The analog sensor input signal is converted into a digital value with use of an A/D converter. All symptoms of the error code are handled by anti-bouncing. Only one symptom can be active at a time.

Description:

Error-symptoms are defined to this diagnosis function as following:

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │ │ │ └─> Short circuit to  Vbatt or Open Circuit "SCP_OC"     (= SYM_0)
  │ │ └───> Short circuit to Ground "SCG"                        (= SYM_1)
  │ └─────> (not used)                                           (= SYM_2)
  └───────> (not used)                                           (= SYM_3)
```

Application conditions:

Initialisation:

LV_ERR_TCO_EL = *Refer to filtering configuration for the initialisation*
LV_END_DIAG_TCO_EL = *Refer to filtering configuration for the initialisation*
LV_CDN_DIAG_TCO_EL = *Refer to filtering configuration for the initialisation*
ERR_SYM_TCO_EL = *Refer to filtering configuration for the initialisation*

Recurrence: 100 ms

Activation: LV_IGK = 1 **and**
LV_CDN_VB_MIN_DIAG = 1 **and**
LV_INH_DIAG_TCO_EL = 0

Deactivation: Activation condition not true
At Deactivation: LV_CDN_DIAG_TCO_EL = 0

Formula section:

Error detection:

```
if (1)          VP_TCO[0] < C_VP_TCO_MIN_DIAG_TCO_EL
                // coolant temperature sensor voltage signal (VP_TCO[0])
then (1)        (error detection "Short circuit to GND")
                LV_CDN_DIAG_TCO_EL = 1
                ERR_SYM_TCO_EL = "SYM_1"
                LV_ERR_TCO_EL = 1                          (after debounce)
else (1)
    if (2)      VP_TCO[0] > C_VP_TCO_MAX_DIAG_TCO_EL
                // coolant temperature sensor voltage signal (VP_TCO[0])
```

```
        then (2)     (error detection "Short circuit to Vbatt or open load" possible)
            if (3)   TIA_THR >= C_TIA_THR_MIN_DIAG_TCO_EL
                     or
                     TIA_THR < C_TIA_THR_MIN_DIAG_TCO_EL              and
                     T_DIAG_AST > C_T_AST_MIN_DIAG_TCO_EL
            then (3) LV_CDN_DIAG_TCO_EL = 1
                     ERR_SYM_TCO_EL = "SYM_0"
                     LV_ERR_TCO_EL = 1                          (after debounce)
            else (3) LV_CDN_DIAG_TCO_EL = 0
            endif (3)
        else (2)     LV_CDN_DIAG_TCO_EL = 1
                     ERR_SYM_TCO_EL = "NO_SYM"
                     LV_ERR_TCO_EL = 0                          (after rebounce)
        endif (2)
    endif (1)
```

End of diagnosis:

For the calculation of LV_END_DIAG_TCO_EL see "Anti-bounce Algorithm, calculation of the end of diagnosis".

26.16.3 Coolant temperature signal gradient diagnosis (TCO_GRD)

FUNCTION DESCRIPTION:

General information:

The purpose of the signal gradient diagnosis is to detect a not plausible gradient on the measured coolant temperature sensor signal. The monitoring is based on a comparison between the actual measured and the last measure coolant temperature value. The symptom of the error code is handled by anti-bouncing.

Description:

Error-symptoms are defined to this diagnosis function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> -                                          (= SYM_0)
 │  │  └───> -                                          (= SYM_1)
 │  └─────> -                                          (= SYM_2)
 └───────> Coolant temperature signal gradient error         (= SYM_3)
```

Application conditions:

Initialisation:

LV_ERR_TCO_GRD = *Refer to filtering configuration for the initialisation*

LV_END_DIAG_TCO_GRD = *Refer to filtering configuration for the initialisation*

LV_CDN_DIAG_TCO_GRD = *Refer to filtering configuration for the initialisation*

ERR_SYM_TCO_GRD = *Refer to filtering configuration for the initialisation*

Recurrence: 100 ms

Activation:

```
If      LV_IGK = 1                     and
        LV_CDN_VB_MIN_DIAG = 1         and
        LV_INH_DIAG_TCO_GRD = 0
then    LV_CDN_DIAG_TCO_GRD = 1
else    LV_CDN_DIAG_TCO_GRD = 0
endif
```

Formula section:

Error detection:

```
If      $| TCO\_MES_{(n-1)} - TCO\_MES_{(n)} | > C\_TCO\_MES\_GRD\_MAX\_DIAG\_TCO\_GRD$
then    ERR_SYM_TCO_GRD = “SYM_3”
        LV_ERR_TCO_GRD = 1              (after debounce)
else    ERR_SYM_TCO_GRD = “NO_SYM”
        LV_ERR_TCO_GRD = 0              (after rebounce)
endif
```

End of diagnosis:

For the calculation of LV_END_DIAG_TCO_GRD see “Anti-bounce Algorithm, calculation of the end of diagnosis”.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VP_TCO_MIN_DIAG_TCO_EL	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Minimum raw sensor signal voltage for the coolant temperature electrical diagnosis					
C_VP_TCO_MAX_DIAG_TCO_EL	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Maximum raw sensor signal voltage for the coolant temperature electrical diagnosis					
C_TIA_THR_MIN_DIAG_TCO_EL	1	0...FEH	-48...142.5	0.75	[°C]
Minimum intake air temperature value for the coolant temperature electrical diagnosis					
C_T_AST_MIN_DIAG_TCO_EL	1	0...FFFFH	0...65535	1	[s]
Minimum time delay after start for the coolant temperature electrical diagnosis					
C_TCO_MES_GRD_MAX_DIAG_TCO_GRD	1	0...FEH	0...190.5	0.75	[°C]
Maximum coolant temperature gradient for the coolant temperature gradient diagnosis					
C_CTR_ABC_THD_TCO_PREL	1	0...FFH	0...255	1	[-]
Threshold for anti-bounce counter value for preliminary coolant temperature error detection					

26.17 Coolant temperature sensor diagnosis (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_TCO_EL	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature signal range diagnosis inhibit					
LV_INH_DIAG_TCO_GRD	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature signal gradient diagnosis inhibit					
LV_INH_DIAG_TCO_STUCK	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature stuck signal diagnosis inhibit					
LV_INH_DIAG_RBM_TCO_STUCK	V	0...1H	0...1	1	[-]
Boolean for temperature stuck signal diagnosis inhibit due to OBD error					

Input data:

LV_IGK	LV_ERR_TCO_EL	LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS
LV_ST_END	LV_ERR_TCO_STUCK_RNG		

FUNCTION DESCRIPTION:

General information:

Task of the coolant temperature sensor diagnosis (Appl. Inc.) is to allow the projects to adapt their specific requirements to generic coolant temperature sensor diagnosis. The setting of the interface output variables have to be provided by the projects.

Application conditions:

Initialisation: see separate chapter

Recurrence: see separate chapter

Activation: at every engine operating state

Deactivation: -

26.17.1 Coolant temperature signal range diagnosis - interface parameter

FUNCTION DESCRIPTION:

General information:

Depending on project specific requirements, the electrical diagnosis of the main coolant temperature sensor can be inhibited by setting of LV_INH_DIAG_TCO_EL.

Application conditions:

Initialisation:

```
If      Reset or Transition: LV_IGK = 0 -> LV_IGK = 1
then    LV_INH_DIAG_TCO_EL = 0
endif
```

Recurrence: 100 ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

LV_INH_DIAG_TCO_EL = 0

Error treatment:

Error debounce:

Debounce counter increment:	C_ABC_INC_TCO_EL
Debounce counter maximum value:	C_ABC_MAX_TCO_EL

Diagnosis Configuration

ERRM configuration for LV_ERR_TCO_EL is "**STD_INI**" //*all 0 at LV_IGK 0->1 or reset*

26.17.2 Coolant temperature signal gradient diagnosis - interface parameter

FUNCTION DESCRIPTION:

General information:

Depending on project specific requirements, the signal gradient diagnosis of the main coolant temperature sensor can be inhibited by setting of LV_INH_DIAG_TCO_GRD.

Application conditions:

Initialisation:

```
If      Reset or Transition: LV_IGK = 0 -> LV_IGK = 1
then    LV_INH_DIAG_TCO_GRD = 0
endif
```

Recurrence: 100 ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

```
If      LV_ERR_TCO_EL = 1                  or
        LV_ERR_TCO_STUCK = 1               or
        LV_ERR_TCO_PLAUS = 1
then    LV_INH_DIAG_TCO_GRD = 1
else    LV_INH_DIAG_TCO_GRD = 0
endif
```

Error treatment:

Error debounce:

Debounce counter increment: C_ABC_INC_TCO_GRD
Debounce counter maximum value: C_ABC_MAX_TCO_GRD

Diagnosis Configuration

ERRM configuration for LV_ERR_TCO_GRD is "**MEM**" //*error is set irreversible*

26.17.3 Coolant temperature signal stuck diagnosis - interface parameter

FUNCTION DESCRIPTION:

General information:

Depending on project specific requirements, the signal stuck diagnosis of the main coolant temperature sensor can be inhibited by setting of LV_INH_DIAG_TCO_STUCK.

Application conditions:

Initialisation: at RESET **or** LV_IGK = 0 -> 1 **or** FMY clear:
LV_INH_DIAG_TCO_STUCK = 0
LV_INH_DIAG_RBM_TCO_STUCK = 0

Recurrence: 1000 ms

Activation: LV_ST_END = 1
(Calculation of the TCO stuck diagnosis interface parameter enabled)

Deactivation: LV_ST_END = 0

(Calculation of the TCO stuck diagnosis interface parameter disabled)

Formula section:

```
If      LV_ERR_TCO_EL = 1                   or
        LV_ERR_TCO_GRD = 1                  or
        LV_ERR_TCO_STUCK_RNG = 1            or
        LV_ERR_TCO_PLAUS = 1
then    LV_INH_DIAG_RBM_TCO_STUCK = 1
endif

If      LV_INH_DIAG_RBM_TCO_STUCK = 1       or
        LV_IGK = 0                          or
        LV_DC = 0
then    LV_INH_DIAG_TCO_STUCK = 1
endif
```

Error treatment:

Error debounce:

no debounce

Diagnosis Configuration

Initialization LV_ERR_TCO_STUCK is managed by function, thus configuration is "**NO**"

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TCO_EL	1	0...FFH	0...255	1	[-]
Debounce counter increment value for the coolant temperature sensor electrical diagnosis					
C_ABC_MAX_TCO_EL	1	1...FFH	1...255	1	[-]
Debounce counter maximum value for the coolant temperature sensor electrical diagnosis					
C_ABC_INC_TCO_GRD	1	0...FFH	0...255	1	[-]
Debounce counter increment value for the coolant temperature sensor gradient diagnosis					
C_ABC_MAX_TCO_GRD	1	1...FFH	1...255	1	[-]
Debounce counter maximum value for the coolant temperature sensor gradient diagnosis					

26.17.4 Interface for Rate – Based - Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_TCO_STUCK	V/O	0...7H	0...7	1	[-]
Interface of TCO_STUCK monitor with the Rate-Based Monitoring statistics					

Input data:

LV_END_DIAG_TCO_STUCK	CTR_ERR_DYN_NR	LV_DC	LV_CDN_CST_RBM

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the TCO_STUCK monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TCO_STUCK data.

Within STATE_RBM_TCO_STUCK, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

Application conditions:

Initialisation :

at ECU reset :

bit 0, bit 1 and bit 2 of STATE_RBM_TCO_STUCK = 0

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_TCO_STUCK = 0

on failure memory reset :

bit 1 of STATE_RBM_TCO_STUCK = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_TCO_EL	LV_ERR_TCO_PLAUS	LV_ERR_TCO_GRD	LV_ERR_TCO_STUCK_RNG

```
    If(1)       { CPU optimization at LV_DC 0 → 1 transition }
                CTR_ERR_DYN_NR <> 0   { the dynamic failure memory isn't empty }
    Then(1)
                While  bit 1 of STATE_RBM_TCO_STUCK = 0 do
                       with each XX failure of the above list :
                       ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                       SYNCHRONIZATION<CALL>)
                       If(2)        XX has a pending status
                       Then(2)      bit 1 of STATE_RBM_TCO_STUCK = 1
                       Endif(2)
                Endwhile
    Else(1)
                { the dynamic failure memory is empty }
                No action
    Endif(1)
```

Every 1 s :

```
If    bit 0 of STATE_RBM_TCO_STUCK = 0
Then
            If     LV_END_DIAG_TCO_STUCK = 1
            Then   bit 0 of STATE_RBM_TCO_STUCK = 1
            Endif
Endif
If    bit 1 of STATE_RBM_TCO_STUCK = 0
Then
            If     LV_INH_DIAG_RBM_TCO_STUCK = 1
            Then   bit 1 of STATE_RBM_TCO_STUCK = 1
            Endif
Endif
If    bit 2 of STATE_RBM_TCO_STUCK = 0
Then
            If     LV_CDN_CST_RBM = 1
            Then   bit 2 of STATE_RBM_TCO_STUCK = 1
            Endif
Endif
```

26.18 Coolant temperature sensor (radiator outlet) diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TCO_2	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on coolant temperature (radiator outlet) acquisition					
LV_ERR_TCO_2_PREL	V/O	0...1H	0...1	1	[-]
Boolean for preliminary error currently present on coolant temperature (radiator outlet) acquisition					
LV_ERR_TCO_2_PREL_DET	V/S	0...1H	0...1	1	[-]
Boolean for preliminary error present on coolant temperature (radiator outlet) acquisition during driving cycle					
LV_CDN_ENA_TCO_2_PREL	V	0...1H	0...1	1	[-]
Boolean for preliminary coolant temperature (radiator outlet) error enable conditions					
LV_ERR_TCO_2_EL	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal range error					
ERR_SYM_TCO_2_EL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for coolant temperature (radiator outlet) signal range error					
LV_CDN_DIAG_TCO_2_EL	V	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal range diagnosis conditions					
LV_END_DIAG_TCO_2_EL	V	0...1H	0...1	1	[-]
Boolean for end of coolant temperature (radiator outlet) signal range diagnosis					
LV_ERR_TCO_2_GRD	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal gradient error					
ERR_SYM_TCO_2_GRD	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for coolant temperature (radiator outlet) signal gradient error					
LV_CDN_DIAG_TCO_2_GRD	V	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal gradient diagnosis conditions					
LV_END_DIAG_TCO_2_GRD	V	0...1H	0...1	1	[-]
Boolean for end of coolant temperature (radiator outlet) signal gradient diagnosis					

Input data:

VP_TCO [NC_NR_TCO_SENS]	TIA_THR	LV_CDN_VB_MIN_DIAG	LV_INH_DIAG_TCO_2_EL
LV_IGK	T_DIAG_AST	CTR_ABC_TCO_2_EL	LV_INH_DIAG_TCO_2_GRD
CTR_ABC_TCO_2_GRD	TCO_2_MES	NC_NR_TCO_SENS	LV_ERR_TCO_2_PLAUS

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis function is to detect electrical failures as defined by OBD I requirements. The coolant temperature (radiator outlet) raw sensor voltage (VP_TCO [1]) is measured with use of an A/D converter by the ECU hardware. The modified sensor signal is converted then into a measured temperature value at radiator outlet. Each by the ECU hardware measured raw sensor value is checked as well as the converted temperature signal.

A global coolant temperature (radiator outlet) error is set without debounce as soon as an electrical- or signal gradient- or coolant temperature (radiator outlet) plausibility error is detected. All separate diagnosis functions are performed all the time if the diagnosis conditions are fulfilled. The deactivation of diagnosis functions with use of configuration switches is not possible.

In addition a preliminary coolant temperature (radiator outlet) error is set as soon as debouncing of an electrical-, or signal gradient- symptom occurs. The preliminary coolant temperature (radiator outlet) error will be stored in the non-volatile memory during the ECU's self-holding phase to be available at the beginning of the next driving cycle.

Application conditions:

Initialisation at Reset:

LV_ERR_TCO_2 = 0
LV_CDN_ENA_TCO_2_PREL = 1
LV_ERR_TCO_2_PREL = LV_ERR_TCO_2_PREL_DET *(NVMY)*
LV_ERR_TCO_2_PREL_DET = LV_ERR_TCO_2_PREL_DET *(NVMY)*
(NVMY: stored in the non-volatile memory)

Initialisation at Engine stop to Engine run (ES_to_ERU):

LV_ERR_TCO_2_PREL_DET = 0
LV_CDN_ENA_TCO_2_PREL = 0

Initialisation at FMY clear:

LV_ERR_TCO_2 = 0
LV_ERR_TCO_2_PREL = 0
LV_ERR_TCO_2_PREL_DET = 0

Recurrence: 100 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

```
If      LV_ERR_TCO_2_EL = 1        or
        LV_ERR_TCO_2_GRD = 1       or
        LV_ERR_TCO_2_PLAUS = 1
then    LV_ERR_TCO_2 = 1           (without debounce)
else    LV_ERR_TCO_2 = 0
endif
```

26.18.1.1 Preliminary coolant temperature (radiator outlet) failure (at ES after RESET)

Application conditions:

Recurrence: 100 ms

Activation: LV_CDN_ENA_TCO_2_PREL = 1

Deactivation: LV_CDN_ENA_TCO_2_PREL = 0

Formula section:

```
If      CTR_ABC_TCO_2_EL > 0          or
        CTR_ABC_TCO_2_GRD > 0         or
        LV_ERR_TCO_2_PREL_DET = 1
then    LV_ERR_TCO_2_PREL = 1         (without debounce)
else    LV_ERR_TCO_2_PREL = 0
endif
```

26.18.1.2 Preliminary coolant temperature failure (at ERU / at ES after ERU_to_ES)

Application conditions:

Recurrence: 100 ms

Activation: LV_CDN_ENA_TCO_2_PREL = 0

Deactivation: LV_CDN_ENA_TCO_2_PREL = 1

Formula section:

```
If      CTR_ABC_TCO_2_EL > C_CTR_ABC_THD_TCO_2_PREL     or
        CTR_ABC_TCO_2_GRD > C_CTR_ABC_THD_TCO_2_PREL    or
        LV_ERR_TCO_2_PLAUS = 1
then    LV_ERR_TCO_2_PREL_DET = 1     (irrevocable for current driving cycle)
endif

If      CTR_ABC_TCO_2_EL > 0          or
        CTR_ABC_TCO_2_GRD > 0         or
        LV_ERR_TCO_2_PLAUS = 1        or
then    LV_ERR_TCO_2_PREL = 1         (without debounce)
else    LV_ERR_TCO_2_PREL = 0
endif
```

26.18.2 Coolant temperature (radiator outlet) signal range diagnosis

FUNCTION DESCRIPTION:

General information:

The purpose of the signal range- or electrical diagnosis is to detect electrical errors in the input circuit of the coolant temperature (radiator outlet) sensor. The analog sensor input signal is converted into a digital value with use of an A/D converter. All symptoms of the error code are handled by anti-bouncing. Only one symptom can be active at a time.

Description:

Error-symptoms are defined to this diagnosis function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to  Vbatt or Open Circuit "SCP_OC"   (= SYM_0)
 │ │ └───> Short circuit to Ground "SCG"                      (= SYM_1)
 │ └─────> (not used)                                         (= SYM_2)
 └───────> (not used)                                         (= SYM_3)
```

Application conditions:

Initialisation:

LV_ERR_TCO_2_EL = *Refer to filtering configuration for the initialisation*

LV_END_DIAG_TCO_2_EL = *Refer to filtering configuration for the initialisation*

LV_CDN_DIAG_TCO_2_EL = *Refer to filtering configuration for the initialisation*

ERR_SYM_TCO_2_EL = *Refer to filtering configuration for the initialisation*

Recurrence: 100 ms

Activation:

LV_IGK = 1 **and**

LV_CDN_VB_MIN_DIAG = 1 **and**

LV_INH_DIAG_TCO_2_EL = 0

Deactivation: Activation condition not true

At deactivation: LV_CDN_DIAG_TCO_2_EL = 0

Formula section:

Error detection:

if (1) VP_TCO[1] < C_VP_TCO_2_MIN_DIAG_TCO_2_EL

```
                    // coolant temperature sensor (radiator outlet) voltage signal (VP_TCO[1])
then (1)            (error detection “Short circuit to GND”)
                    LV_CDN_DIAG_TCO_2_EL = 1
                    ERR_SYM_TCO_2_EL = “SYM_1”
                    LV_ERR_TCO_2_EL = 1                        (after debounce)
else (1)
    if (2)          VP_TCO[1] > C_VP_TCO_2_MAX_DIAG_TCO_2_EL
                    // coolant temperature sensor (radiator outlet) voltage signal (VP_TCO[1])
    then (2)        (error detection “Short circuit to Vbatt or open load” possible)
        if (3)      TIA_THR >= C_TIA_THR_MIN_DIAG_TCO_2_EL
                    or
                    TIA_THR < C_TIA_THR_MIN_DIAG_TCO_2_EL  and
                    T_DIAG_AST > C_T_AST_MIN_DIAG_TCO_2_EL
        then (3)    LV_CDN_DIAG_TCO_2_EL = 1
                    ERR_SYM_TCO_2_EL = “SYM_0”
                    LV_ERR_TCO_2_EL = 1                        (after debounce)
        else (3)    LV_CDN_DIAG_TCO_2_EL = 0
        endif (3)
    else (2)        LV_CDN_DIAG_TCO_2_EL = 1
                    ERR_SYM_TCO_2_EL = “NO_SYM”
                    LV_ERR_TCO2_EL = 0                         (after rebounce)
    endif (2)
endif (1)
```

End of diagnosis:

For the calculation of LV_END_DIAG_TCO_2_EL see “Anti-bounce Algorithm, calculation of the end of diagnosis”.

26.18.3 Coolant temperature (radiator outlet) signal gradient diagnosis

FUNCTION DESCRIPTION:

General information:

The purpose of the signal gradient diagnosis is to detect a not plausible gradient on the measured coolant temperature (radiator outlet) sensor signal. The monitoring is based on a comparison between the actual measured and the last measured coolant temperature value at radiator outlet. The symptom of the error code is handled by anti-bouncing.

Description:

Error-symptoms are defined to this diagnosis function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                        (= SYM_0)
 │ │ └───> -                                        (= SYM_1)
 │ └─────> -                                        (= SYM_2)
 └───────> Coolant temperature signal gradient error       (= SYM_3)
```

Application conditions:

Initialisation:

LV_ERR_TCO_2_GRD = *Refer to filtering configuration for the initialisation*

LV_END_DIAG_TCO_2_GRD = *Refer to filtering configuration for the initialisation*

LV_CDN_DIAG_TCO_2_GRD = *Refer to filtering configuration for the initialisation*

ERR_SYM_TCO_2_GRD = *Refer to filtering configuration for the initialisation*

Recurrence: 100 ms

Activation:

```
If      LV_IGK = 1                          and
        LV_CDN_VB_MIN_DIAG = 1              and
        LV_INH_DIAG_TCO_2_GRD = 0
then    LV_CDN_DIAG_TCO_2_GRD = 1
else    LV_CDN_DIAG_TCO_2_GRD = 0
endif
```

Deactivation: Activation condition not true

Formula section:

Error detection:

```
If      $| TCO\_2\_MES_{(n-1)} – TCO\_2\_MES_{(n)} | >$
            C_TCO_2_MES_GRD_MAX_DIAG_TCO_2
then    ERR_SYM_TCO_2_GRD = “SYM_3”
        LV_ERR_TCO_2_GRD = 1                (after debounce)
else    ERR_SYM_TCO_2_GRD = “NO_SYM”
        LV_ERR_TCO_2_GRD = 0                (after rebounce)
endif
```

End of diagnosis:

For the calculation of LV_END_DIAG_TCO_2_GRD see “Anti-bounce Algorithm, calculation of the end of diagnosis”.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VP_TCO_2_MIN_DIAG_TCO_2_EL	1	0...7FFFH	0...4.999847412	1.53E-04	[V]
Minimum raw sensor signal voltage for the coolant temperature (radiator outlet) electrical diagnosis					
C_VP_TCO_2_MAX_DIAG_TCO_2_EL	1	0...7FFFH	0...4.999847412	1.53E-04	[V]
Maximum raw sensor signal voltage for the coolant temperature (radiator outlet) electrical diagnosis					
C_TIA_THR_MIN_DIAG_TCO_2_EL	1	0...FEH	-48...142.5	0.75	[°C]
Minimum intake air temperature value for the coolant temperature (radiator outlet) electrical diagnosis					
C_T_AST_MIN_DIAG_TCO_2_EL	1	0...FFFFH	0...65535	1	[s]
Minimum time delay after start for the coolant temperature (radiator outlet) electrical diagnosis					
C_TCO_2_MES_GRD_MAX_DIAG_TCO_2	1	0...FEH	0...190.5	0.75	[°C]
Maximum coolant temperature gradient for the coolant temperature (radiator outlet) gradient diagnosis					
C_CTR_ABC_THD_TCO_2_PREL	1	0...FFH	0...255	1	[-]
Threshold for anti-bounce counter value for preliminary coolant temperature (radiator outlet) error detection					

26.19 Coolant temperature sensor (radiator outlet) diagnosis (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_TCO_2_EL	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal range diagnosis inhibit					
LV_INH_DIAG_TCO_2_GRD	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal gradient diagnosis inhibit					

Input data:

LV_IGK	LV_ERR_TCO_2_EL		

FUNCTION DESCRIPTION:

General information:

Task of the coolant temperature (radiator outlet) sensor diagnosis (Appl. Inc.) is to allow the projects to adapt their specific requirements to generic coolant temperature (radiator outlet) sensor diagnosis. The setting of the interface output variables have to be provided by the projects.

Application conditions:

Initialisation: see separate chapter

Recurrence: 1000 ms

Activation: at every engine operating state

Deactivation: -

26.19.1 Coolant temperature (radiator outlet) signal range diagnosis - interface parameter

FUNCTION DESCRIPTION:

General information:

Depending on project specific requirements, the electrical- or signal range diagnosis of the coolant temperature sensor at radiator outlet can be inhibited by setting of the logical variable LV_INH_DIAG_TCO_2_EL.

Application conditions:

Initialisation:

If Reset **or** Transition: LV_IGK = 0 -> LV_IGK = 1

```
    then        LV_INH_DIAG_TCO_2_EL = 0
    endif
```

Recurrence: 1000 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

LV_INH_DIAG_TCO_2_EL = 0

Error treatment:

Error debounce:

Debounce counter increment:	C_ABC_INC_TCO_2_EL
Debounce counter maximum value:	C_ABC_MAX_TCO_2_EL

Diagnosis Configuration

ERRM configuration for LV_ERR_TCO_2_EL is "**STD_INI**" *//all 0 at LV_IGK 0->1 or reset*

26.19.2 Coolant temperature (radiator outlet) signal gradient diagnosis - interface parameter

FUNCTION DESCRIPTION:

General information:

Depending on project specific requirements, the signal gradient diagnosis of the coolant temperature sensor at radiator outlet can be inhibited by setting of the logical value LV_INH_DIAG_TCO_2_GRD.

Application conditions:

Initialisation:

```
If          Reset or Transition: LV_IGK = 0 -> LV_IGK = 1
then        LV_INH_DIAG_TCO_2_GRD = 0
endif
```

Recurrence: 1000 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

```
If      LV_ERR_TCO_2_EL = 1
Then    LV_INH_DIAG_TCO_2_GRD = 1
Else    LV_INH_DIAG_TCO_2_GRD = 0
Endif
```

Error treatment:

Error debounce:

Debounce counter increment:	C_ABC_INC_TCO_2_GRD
Debounce counter maximum value:	C_ABC_MAX_TCO_2_GRD

Diagnosis Configuration:

ERRM configuration for LV_ERR_TCO_2_GRD is "**MEM**" *//error is set irreversible*

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TCO_2_EL	1	0...FFH	0...255	1	[-]
Debounce counter increment value for the coolant temperature (radiator outlet) sensor electrical diagnosis					
C_ABC_MAX_TCO_2_EL	1	1...FFH	1...255	1	[-]
Debounce counter maximum value for the coolant temperature (radiator outlet) sensor electrical diagnosis					
C_ABC_INC_TCO_2_GRD	1	0...FFH	0...255	1	[-]
Debounce counter increment value for the coolant temperature (radiator outlet) sensor gradient diagnosis					
C_ABC_MAX_TCO_2_GRD	1	1...FFH	1...255	1	[-]
Debounce counter maximum value for the coolant temperature (radiator outlet) sensor gradient diagnosis					

26.20 Diagnosis electrical controlled water pump CWP

26.20.1 Electrical controlled water pump speed monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CWP_PLAUS	V/O	0...1H	0...1	1	[-]
Plausibility error electrical waterpump					
ERR_SYM_CWP_PLAUS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error-symptoms electrical waterpump					
LV_END_DIAG_CWP_PLAUS	V/O	0...1H	0...1	1	[-]
End diag flag of Plausibility error electrical waterpump					
LV_CWP_NOT_PLAUS	V/O	0...1H	0...1	1	[-]
Waterpump speed stages out of range					
N_REL_CWP_DIF	V/O	0...FFH	0...255	1	[-]
Difference of actual waterpump speed range to setpiont					
LV_CDN_DIAG_CWP_PLAUS	V/O	0...1H	0...1	1	[-]
Condition diagnosis waterpump plausibility error					
T_CWP_DIF	V	0...FFH	0...25.5	0.1	[s]
Timer for CWP speed monitoring					
T_CWP_DIF_END_DIAG	V	0...FFH	0...25.5	0.1	[s]
Timer for CWP speed monitoring End Diag detection					

Input data:

TCO	N_REL_CWP	N_REL_CWP_SP	LV_IGK
LV_CWP_PRE_LOCK	LV_ERR_BSD	V_CWP	LV_CWP_LOCK
TEMP_EL_CWP	LV_CWP_VCC_PLAUS	LV_CWP_TEMP_HIGH	C_TEMP_EL_CWP_MAX_THD_1
LV_N_MON_CWP_DEAC	LV_INH_CWP_DIAG		

FUNCTION DESCRIPTION:

Description:

This diagnosis function is the inteface between BSD driver and waterpump controller.

Error-symptoms are defined to this diagnosis function as following :

```
 ┌─┬─┬─┬─┐
 │ │ │ └─> pumping speed out of range        (= SYM_0)
 │ │ └───> -                                  (= SYM_1)
 │ └─────> -                                  (= SYM_2)
 └───────> -                                  (= SYM_3)
```

Application conditions:

Initialization:	all: 0 at LV_IGK 0->1 or reset or at clearing error memory (filter type NO_FIL)
Recurrence:	100ms
Activation:	LV_IGK = 1 **and** LV_ERR_BSD = 0 **and** LV_INH_CWP_DIAG = 0
Deactivation:	LV_IGK = 0 **or** LV_ERR_BSD = 1 **or** LV_INH_CWP_DIAG = 1

Formula section:

N_REL_CWP_DIF = ABS (N_REL_CWP - N_REL_CWP_SP)

Ratio detection:

```
If   TCO > C_TCO_CWP_REL_MIN_DIAG                    and
     TEMP_EL_CWP < C_TEMP_EL_CWP_MAX_THD_1           and
     V_CWP > C_V_CWP_MIN_DIAG_MON                    and
     LV_CWP_VCC_PLAUS = 0                            and
     LV_CWP_PRE_LOCK = 0                             and   // LV_CWP_PRE_LOCK can
     LV_CWP_TEMP_HIGH = 0
 Then  LV_CDN_DIAG_CWP_PLAUS = 1
       If   LV_CWP_NOT_PLAUS = 0
       Then If   N_REL_CWP_DIF > C_N_REL_CWP_DIF_MAX
            Then increment T_CWP_DIF every 100ms (limit to FFH)
                 If    T_CWP_DIF ≥ C_T_CWP_DIF_MAX
```

```
                  Then LV_CWP_NOT_PLAUS = 1
                  Endif
            Else  T_CWP_DIF = 0 secs
            Endif
        Else If   N_REL_CWP_DIF ≤ C_N_REL_CWP_DIF_MAX - C_N_REL_CWP_HYS
            Then  LV_CWP_NOT_PLAUS = 0 Keine zeitliche Entprellung im GUT Fall!!
                  T_CWP_DIF = 0s
            Endif
        Endif
Else    LV_CDN_DIAG_CWP_PLAUS = 0
Endif
```

Error management:

```
If      LV_CWP_NOT_PLAUS =1
Then    ERR_SYM_CWP_PLAUS = SYM_0
        LV_ERR_CWP_PLAUS = 1
        LV_END_DIAG_CWP_PLAUS = 1
Else    ERR_SYM_CWP_PLAUS = NO_SYM
        LV_ERR_CWP_PLAUS = 0
Endif
```

End of diagnosis:

```
If      LV_CDN_DIAG_CWP_PLAUS = 1
Then    increment T_CWP_DIF_END_DIAG every 100ms (limit to FFH)
        If      T_CWP_DIF_END_DIAG ≥ C_T_CWP_DIF_MAX
        Then    LV_END_DIAG_CWP_PLAUS = 1
        Endif
Else    T_CWP_DIF_END_DIAG = 0 secs
Endif
```

If one of the error symtoms is active then the error memory interface LV_ERR_CWP_PLAUS is active and reset at "NO_SYM".

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_REL_CWP_DIF_MAX	1	0...FFH	0...255	1	[-]
Max- control deviation threshold for ratio speed stages diagnosis					
C_T_CWP_DIF_MAX	1	0...FFH	0...25.5	0.1	[s]
Timer Threshold CWP speed monitoring (Timer: T_CWP)					
C_N_REL_CWP_HYS	1	0...FFH	0...255	1	[-]
Ratio speed stages hysteresis					
C_TCO_CWP_REL_MIN_DIAG	1	0...FEH	-48...142.5	0.75	[°C]
TCO min threshold to enable CWP speed monitoring					
C_V_CWP_MIN_DIAG_MON	1	0...FFH	0...25.5	0.1	[V]
Min threshold voltage CWP to enable CWP speed monitoring					

26.20.2 Diagnosis of hardware input line for Limp Home Operation of CWP

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CWP_COM_PLAUS	V/O	0...1H	0...1	1	[-]
Plausibility error electrical waterpump communication					
ERR_SYM_CWP_COM_PLAUS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error-symptoms electrical waterpump communication					
LV_CDN_DIAG_CWP_COM_PLAUS	V/O	0...1H	0...1	1	[-]
Diagnosis condition electrical waterpump communication					
LV_END_DIAG_CWP_COM_PLAUS	V/O	0...1H	0...1	1	[-]
End diag flag of Plausibility error electrical waterpump communication					

Input data:

LV_ERR_BSD	LV_INH_CWP_DIAG	LV_IGK	LV_CDN_VB_MIN_DIAG
LV_CWP_HW_LIH_IN_CHK			

FUNCTION DESCRIPTION:

General information:

This function detects an error in the hardware line for Limp Home Operation of the CWP.

A hardware signal is connected to the CWP to activate the limp home mode of the CWP in case of a BSD communication error. To detect the correct functionality of the hardware signal line, the CWP state is submitted to the ECU (EWM_ERR_3). The ECU compares the state of the LV_IGK or LV_CDN_VB_MIN_DIAG (dependend on a switch LC_SWI_CWP_INP) with the state of EWM_ERR3 register (LV_CWP_HW_LIH_IN_CHK). For no compliance a plausibility failure is set.

Description:

Error-symptoms are defined to this diagnosis function as following :

```
|_|_|_|_|
 | | | └─> -                                          (= SYM_0)
 | | └───> -                                          (= SYM_1)
 | └─────> -                                          (= SYM_2)
 └───────> Hardware input line not plausible          (= SYM_3)
```

Application conditions:

Initialisation: according ABC filter Type **STD_INI**; all other output data are set to 0 at LV_IGK 0→1 **or** reset **or** at clearing error memory

Recurrence: 100 ms

Activation: LV_ERR_BSD = 0 **and** LV_INH_CWP_DIAG = 0 **and** LV_IGK = 1

Deactivation: LV_ERR_BSD = 1 **or** LV_INH_CWP_DIAG = 1 **or** LV_IGK = 0

Formula section:

```
/* Counter for LV_CWP_HW_LIH_IN_CHK signal */
If          LC_SWI_CWP_INP = 1
Then        LV_IGK_CDN_VB_MIN_DIAG = LV_IGK
Else        LV_IGK_CDN_VB_MIN_DIAG = LV_CDN_VB_MIN_DIAG
Endif
If           LV_IGK_CDN_VB_MIN_DIAG =  0  --> 1
        or  LV_IGK_CDN_VB_MIN_DIAG =  1  --> 0
Then        reset debounce counter
Else        do nothing
Endif


If          LV_IGK_CDN_VB_MIN_DIAG = 1
Then
    If          LV_CWP_HW_LIH_IN_CHK = 0
    Then        ERR_SYM_CWP_COM_PLAUS = SYM_3    /* debounce error */
                LV_ERR_CWP_COM_PLAUS = 1
                LV_END_DIAG_CWP_COM_PLAUS = 1
    Else        ERR_SYM_CWP_COM_PLAUS = NO_SYM
                LV_ERR_CWP_COM_PLAUS = 0
    Endif
Else
    If          LV_CWP_HW_LIH_IN_CHK = 1
    Then        ERR_SYM_CWP_COM_PLAUS = SYM_3    /* debounce error */
                LV_ERR_CWP_COM_PLAUS = 1
                LV_END_DIAG_CWP_COM_PLAUS = 1
```

```
        Else     ERR_SYM_CWP_COM_PLAUS = NO_SYM
                 LV_ERR_CWP_COM_PLAUS = 0
        Endif
    Endif
```

Error management:

```
If      LV_ERR_BSD = 0  and  LV_INH_CWP_DIAG = 0
Then    LV_CDN_DIAG_CWP_COM_PLAUS = 1
Else    LV_CDN_DIAG_CWP_COM_PLAUS = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_SWI_CWP_INP	1	0...1H	0...1	1	[-]
switch between LV_IGK and LV_CDN_VB_MIN_DIAG					
C_ABC_MAX_CWP_CHK_INP	1	1...FFH	1...255	1	[-]
number of timeouts for antibounce of LV_ERR_CWP_COM_PLAUS					
C_ABC_INC_CWP_CHK_INP	1	0...FFH	0...255	1	[-]
Anti-bounce increment for LV_ERR_CWP_COM_PLAUS					

26.20.3 Waterpump internal shut off errors

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CWP_INT_OFF	V/O	0...1H	0...1	1	[-]
CWP shut down error triggered by internal CWP errors					
ERR_SYM_CWP_INT_OFF	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symtom CWP shut down error triggered by internal CWP errors					
LV_CDN_DIAG_CWP_INT_OFF	V/O	0...1H	0...1	1	[-]
Condition diagnosis CWP internal error					
LV_END_DIAG_CWP_INT_OFF	V/O	0...1H	0...1	1	[-]
End diag flag for diagnosis CWP internal error					
LV_CWP_TMP_OFF	V/O	0...1H	0...1	1	[-]
Temporary shut down of the CWP triggered by internal CWP errors					
LV_CWP_OFF	V/O	0...1H	0...1	1	[-]
Shut down of the CWP triggered by internal CWP errors, set after delay time					
T_CWP_LOCK	V	0...FFH	0...25.5	0.1	[s]
Time delay before the error management is triggered by the internal error "Current consumption too high"					
T_CWP_TEMP_HIGH	V	0...FFH	0...25.5	0.1	[s]
Time delay before the error management is triggered by the internal error "Electronic temp. too high"					
T_CWP_VCC_PLAUS	V	0...FFH	0...25.5	0.1	[s]
Time delay before the error management is triggered by the internal error "Supply voltage too high"					
T_CWP_CDN_DIAG_INT_OFF	V	0...FFH	0...25.5	0.1	[s]
Time delay after the CDN Diag Flag can be set if no error is present --> End Diag Flag set by error Manag.					

Input data:

LV_CWP_TEMP_HIGH	LV_CWP_VCC_PLAUS	LV_CWP_LOCK	LV_ERR_BSD
LV_IGK	LV_N_MON_CWP_DEAC	LV_INH_CWP_DIAG	

FUNCTION DESCRIPTION:

Description:

Error-symptoms are defined to this diagnosis function as following :

```
□□□□
| | | └─> Electronic temperature too high     (= SYM_0)
| | └───> Supply voltage too high            (= SYM_1)
| └─────> Current consumption too high       (= SYM_2)
└───────> -                                  (= SYM_3)
```

Application conditions:

Initialisation: according ABC filter Type **STD_INI**; all other output data are set to 0 at LV_IGK 0→1 **or** reset **or** at clearing error memory

Recurrence: 100ms

Activation: LV_IGK = 1 **and** LV_ERR_BSD = 0 **and**
LV_INH_CWP_DIAG = 0

Deactivation: LV_IGK = 0 **or** LV_ERR_BSD = 1 **or**
LV_INH_CWP_DIAG = 1

Formula section:

```
If      LV_CWP_LOCK = 1                              or
        LV_CWP_TEMP_HIGH = 1                         or
        LV_CWP_VCC_PLAUS = 1
Then    LV_CWP_TMP_OFF = 1
            If      LV_CWP_LOCK = 1
            // Current consumption too high, CWP blocked //
            Then    increment T_CWP_LOCK every 100ms (limit to FFH)
                    If      T_CWP_LOCK ≥ C_T_CWP_LOCK_MAX
                    Then    LV_CWP_OFF  = 1
                    Endif
            Else    T_CWP_LOCK = 0
            Endif
            If      LV_CWP_TEMP_HIGH = 1
            // Electronics CWP temperature too high //
            Then    increment T_CWP_TEMP_HIGH every 100ms (limit to FFH)
                    If      T_CWP_TEMP_HIGH ≥ C_T_CWP_TEMP_HIGH_MAX
                    Then    LV_CWP_OFF  = 1
                    Endif
            Else    T_CWP_TEMP_HIGH = 0
            Endif
            If      LV_CWP_VCC_PLAUS = 1
            // Supply voltage CWP too high //
            Then    increment T_CWP_VCC_PLAUS every 100ms (limit to FFH)
                    If      T_CWP_VCC_PLAUS ≥ C_T_CWP_VCC_PLAUS_MAX
                    Then    LV_CWP_OFF  = 1
                    Endif
            Else    T_CWP_VCC_PLAUS = 0
            Endif
Else    LV_CWP_TMP_OFF = 0
        LV_CWP_OFF  = 0
        T_CWP_LOCK = 0
```

```
            T_CWP_TEMP_HIGH = 0
            T_CWP_VCC_PLAUS = 0
        // Set of LV_CDN_DIAG_CWP_INT_OFF in case of no error present //
        If      LV_N_MON_CWP_DEAC = 0    // LV_CWP_LOCK can only be set at controlled operation
                                            no separate timer for the other two symptoms!        //
        Then  increment T_CWP_CDN_DIAG_INT_OFF every 100ms (limit to FFH)
                If      T_CWP_CDN_DIAG_INT_OFF ≥ C_T_CWP_LOCK_MAX          and
                        T_CWP_CDN_DIAG_INT_OFF ≥ C_T_CWP_TEMP_HIGH_MAX and
                        T_CWP_CDN_DIAG_INT_OFF ≥ C_T_CWP_VCC_PLAUS_MAX
                Then    LV_CDN_DIAG_CWP_INT_OFF  = 1
                Endif
        Else   T_CWP_CDN_DIAG_INT_OFF = 0 secs
        Endif
    Endif
```

Error management:

```
    If        LV_CWP_TEMP_HIGH = 1                          and
              LV_CWP_OFF = 1
    Then         ERR_SYM_CWP_INT_OFF = SYM_0
                 LV_CDN_DIAG_CWP_INT_OFF = 1
    Else     If          LV_CWP_VCC_PLAUS = 1                     and
                         LV_CWP_OFF = 1
             Then      ERR_SYM_CWP_INT_OFF = SYM_1
                         LV_CDN_DIAG_CWP_INT_OFF = 1
             Else    If      LV_CWP_LOCK =1                            and
                             LV_CWP_OFF = 1
                     Then       ERR_SYM_CWP_INT_OFF = SYM_2
                                LV_CDN_DIAG_CWP_INT_OFF = 1
                     Else      ERR_SYM_CWP_INT_OFF = NO_SYM
                     Endif
             Endif
    Endif
```

If one of the error symtoms is active then the error memory interface LV_ERR_CWP_INT_OFF is debounced by the standard debounce algorithm and LV_END_DIAG_CWP_INT_OFF is set by the error memory.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_CWP_INT_OFF	1	0...FFH	0...255	1	[-]
CWP internal Diagnosis increment					
C_ABC_MAX_CWP_INT_OFF	1	1...FFH	1...255	1	[-]
Debounce threshold for internal shut off errors CWP					
C_T_CWP_LOCK_MAX	1	0...FFH	0...25.5	0.1	[s]
Timer threshold for internal CWP error "current consumption too high"					
C_T_CWP_TEMP_HIGH_MAX	1	0...FFH	0...25.5	0.1	[s]
Timer threshold for internal CWP error "Electronic temperature too high"					
C_T_CWP_VCC_PLAUS_MAX	1	0...FFH	0...25.5	0.1	[s]
Timer threshold for internal CWP error "Supply voltage too high"					

26.20.4 Electrical controlled water pump Power Reduction Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
REL_CWP_PWR	V/O	0...FFH	0...99.60937	0.390625	[%]
relative CWP power (reduced power)					
LV_ERR_CWP_PWR	V/O	0...1H	0...1	1	[-]
Logical value for CWP PWR error					
LV_CDN_DIAG_CWP_PWR	V/O	0...1H	0...1	1	[-]
Condition diagnosis elektrical waterpump internal error					
ERR_SYM_CWP_PWR	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error-sym. electrical waterpump power					
LV_END_DIAG_CWP_PWR	V/O	0...1H	0...1	1	[-]
End of CWP PWR diagnosis					
T_CWP_CDN_DIAG_PWR	V	0...FFH	0...25.5	0.1	[s]
Time delay after the CDN Diag Flag can be set if no error is present --> End Diag Flag set by error Manag.					

Input data:

LV_INH_CWP_DIAG	LV_IGK	LV_ERR_BSD	

FUNCTION DESCRIPTION:

This function monitors the power reduction modes of the coolant water pump.

General information:

If the operating conditions of the electrical coolant water pump deviates from the normal conditions the output power of the pump will be reduced by the power management. Depending on the actual state a calibratible power setpoint is given. The state is determined separately by a state machine.

Description:

This diagnosis function is the inteface between BSD driver and waterpump controller.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
│ │ │ │ │
└─┴─┴─┴─┘
 |  |  |  └─> Pump is running dry                                (= SYM_0)
 |  |  └───> Voltage supply not sufficient                       (= SYM_1)
 |  └─────> CWP electronics exceeds temp. threshold level1       (= SYM_2)
 └───────> CWP electronics exceeds temp. threshold level2        (= SYM_3)
```

Application conditions:

Initialization: according ABC filter Type **STD_INI**; all other output data are set to 0 at LV_IGK 0→1 **or** reset **or** at clearing error memory

Recurrence: 100ms

Activation: LV_IGK = 1 **and** LV_ERR_BSD = 0 **and**

LV_INH_CWP_DIAG = 0

Deactivation: LV_IGK = 0 **or** LV_ERR_BSD = 1 **or**
LV_INH_CWP_DIAG = 1

Formula section:

REL_CWP_PWR = ID_REL_CWP_PWR

Error management:

```
If (1)      STATE_CWP_INT = 6
Then (1)    ERR_SYM_CWP_PWR = SYM_0
            LV_CDN_DIAG_CWP_PWR = 1
Else (1)  If (2)      STATE_CWP_INT = 3
          Then (2)    ERR_SYM_CWP_PWR = SYM_1
                      LV_CDN_DIAG_CWP_PWR = 1
          Else (2)  If (3)      STATE_CWP_INT = 4
                    Then (3)    ERR_SYM_CWP_PWR = SYM_2
                                LV_CDN_DIAG_CWP_PWR = 1
                    Else (3)  If (4)      STATE_CWP_INT = 5
                              Then (4)    ERR_SYM_CWP_PWR= SYM_3
                                          LV_CDN_DIAG_CWP_PWR = 1
                              Else (4)    ERR_SYM_CWP_PWR = NO_SYM
                                          // Set of LV_CDN_DIAG_CWP_PWR in case of no error //
                                          If (5)      STATE_CWP_INT = 2
                                          Then (5)    increment T_CWP_CDN_DIAG_PWR
                                                      every 100ms (limit to FFH)
                                              If (6)      T_CWP_ CDN_DIAG_PWR ≥
                                                          C_T_TEMP_EL_CWP_THD_1          and
                                                          T_CWP_ CDN_DIAG_PWR ≥
                                                          C_T_TEMP_EL_CWP_THD_2
                                              Then (6)    LV_CDN_DIAG_CWP_PWR  = 1
                                              Else (6)    LV_CDN_DIAG_CWP_PWR  = 0
                                              Endif (6)
                                          Else (5)    T_CWP_CDN_DIAG_PWR = 0 secs
                                          Endif (5)
                              Endif (4)
                    Endif (3)
          Endif (2)
Endif (1)
```

If one of the error symtoms is active then the error memory interface LV_ERR_CWP_PWR is debounced by the standard debounce allgorithm and LV_END_DIAG_CWP_PWR is set by the error memory.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_REL_CWP_PWR	11	0...FFH	0...99.60937	0.390625	[%]
LDP_STATE_CWP_INT	11	0...AH	0...10	1	[-]
power reducuction of the CWP					
C_ABC_INC_CWP_PWR	1	0...FFH	0...255	1	[-]
CWP internal Diagnosis increment					
C_ABC_MAX_CWP_PWR	1	1...FFH	1...255	1	[-]
Diagnosis increment					

26.20.5 Waterpump internal state

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CWP_INT	V/O	0...AH	0...10	1	[-]
internal operating states of the CWP					
T_CWP_4	V	0...FFH	0...25.5	0.1	[s]
Time delay before the transition to power reduced operation is performed due to voltage supply too low					
T_CWP_5	V	0...FFH	0...25.5	0.1	[s]
Time delay before the transition to power reduced operation is performed due to electronics temp. too high					

Input data:

LV_CWP_BLOCK_DEAC	TEMP_EL_CWP	LV_CWP_PRE_LOCK	LV_ALTER_ERR_TEMP
V_CWP	LV_N_MON_CWP_DEAC	N_REL_CWP_SP	LV_ES
LV_ST	LV_IGK	LV_ERR_BSD	LV_ALTER_ERR_MEC
LV_INH_CWP_DIAG	LV_ALTER_ERR_EL		

FUNCTION DESCRIPTION:

This state machine determines the operation mode of the coolant water pump. By distinguishing several operation modes a specific diagnosis of the CWP at different states is possible.

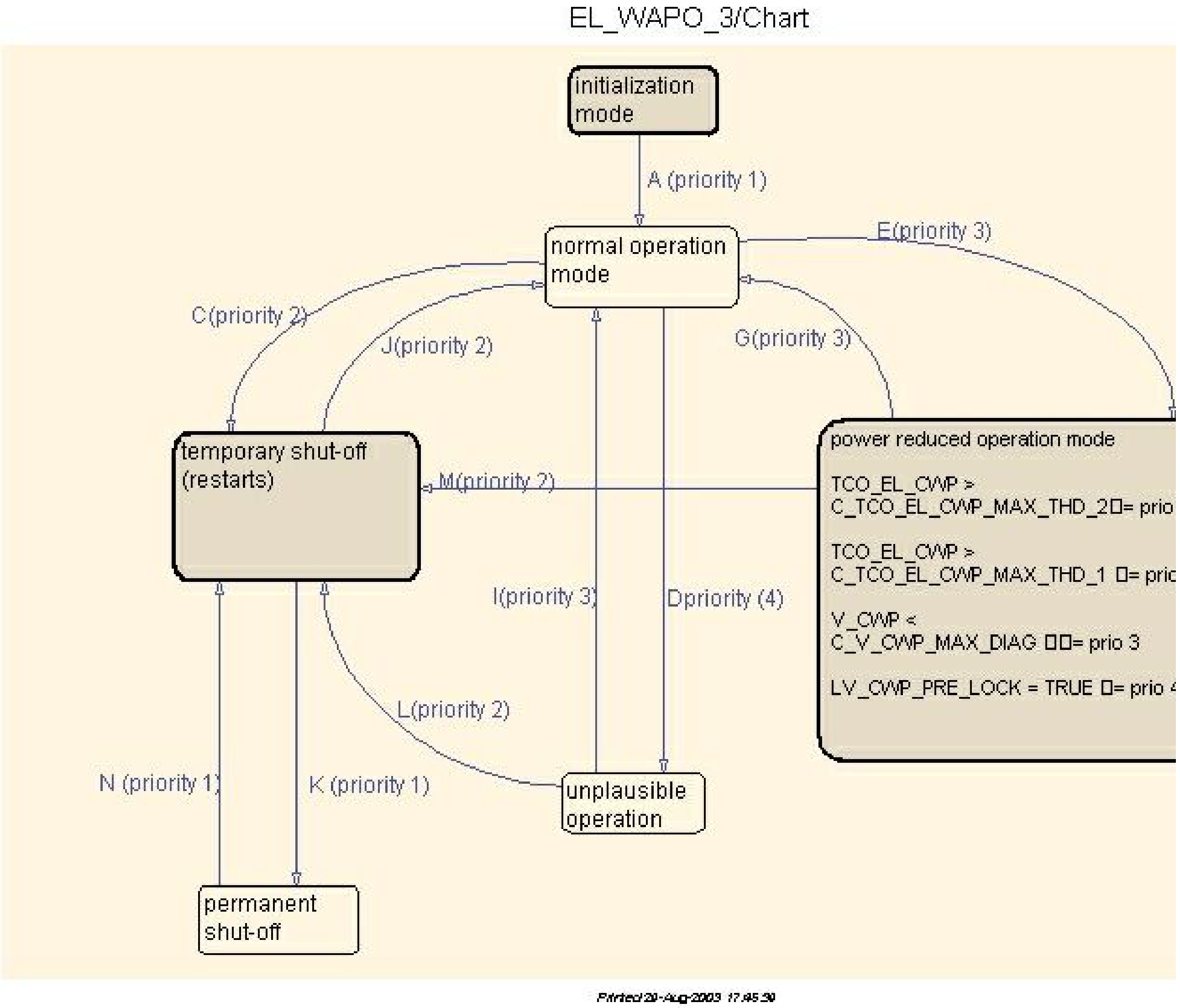
General information:

After initialization some transition conditions must be fulfilled to change from one state to the other.

Operating State of the CWP STATE CWP INT (controlled internally in the CWP electronics):

STATE_CWP_INT	Operating state of the CWP	Addition explanations concerning the operating State
0	initialization	Pump is initialized, not ready to operate
1	normal operation mode open loop control	No speed monitoring active (speed setpoint= actual speed) LV_N_MON_CWP_DEAC = 1
2	normal operation mode closed loop control	Speed Monitoring active, closed loop controlled operation LV_N_MON_CWP_DEAC = 0
3	power reduced mode	Power reduced operation due to low voltage supply Voltage suppy potetially not sufficient to achieve the required speed setpoint
4	power reduced mode	Power reduced operation due to high CWP electronics temperature (level 1) Pump is not able the achieve the required speed setpoint
5	power reduced mode	Power reduced operation due to high CWP electronics temperature (level 2) Pump is not able the achieve the required speed setpoint
6	pump running dry	Pump is running dry and limits threfore the max speed
7	temporary shut-off	High current consumption detected; automatic de-blocking disabled; max. 3 restarts are executed before permanent shut off
8	temporary shut-off	High current consumption detected; automatic de-blocking enabled; pump tries to de-block itself
9	permanent shut-off	Pump deactivates itself; possible reasons: - Voltage supply too high - Current consumption too high - CWP electronics temperature too high
10	unplausible operation	Actual pump speed deviates to desired pump speed; diagnosis active above opening temp. of the thermostat and voltage supply sufficient only

Signal flow diagram:



Application conditions:

Initialization:	LV_IGK 0->1 or reset or at clearing error memory then STATE_CWP_INT = 0
Recurrence:	100ms
Activation:	see Transition A
Deactivation:	-

Formula section:

General transition conditions:

Transition condition A (prio 1):

If LV_IGK = 1

And LV_ERR_BSD = 0

AndLV_INH_CWP_DIAG = 0

Transition condition C (prio 2):

LV_CWP_TMP_OFF = true

Transition condition D (prio 4):

LV_CWP_NOT_PLAUS = true

Transition condition E (prio 3):

```
If          TEMP_EL_CWP > C_TEMP_EL_CWP_MAX_THD_1
Then        increment T_CWP_5 every 100ms (limit to FFH)
            reset  T_CWP_4 = 0 secs
    If      T_CWP_5 ≥ C_T_TEMP_EL_CWP_THD_1
    Then    change mode and reset T_CWP_5 = 0 secs
    Endif
Else        reset T_CWP_5 = 0 secs
    If      V_CWP < C_V_CWP_MIN_DIAG_IT                  and
            N_REL_CWP_SP  >= C_N_REL_CWP_SP_THD_MIN      and
            LV_ST = 0  and  LV_ES = 0                    and
            LV_ALTER_ERR_EL = 0                          and
            LV_ALTER_ERR_MEC = 0                         and
            LV_ALTER_ERR_TEMP = 0
    Then    increment T_CWP_4 every 100ms (limit to FFH)
       If   T_CWP_4 ≥  C_T_TEMP_EL_CWP_THD_2
            Then   change mode and reset T_CWP_4 = 0 secs
       Endif
    Else    reset  T_CWP_4 = 0 secs
       If   LV_CWP_PRE_LOCK = true
       Then change mode
       Endif
    Endif
Endif
```

Transition condition G (prio 3):

```
If      TEMP_EL_CWP < C_TEMP_EL_CWP_MAX_THD_1 - C_TEMP_EL_CWP_HYS  and
        V_CWP > C_V_CWP_MIN_DIAG_IT + C_V_CWP_HYS                   and
        LV_CWP_PRE_LOCK = false
Then    increment T_CWP_5 every 100ms (limit to FFH)
    If      T_CWP_5 ≥ C_T_V_CWP_MAX_DIAG
    Then    change mode and reset T_CWP_5 = 0 secs
    Endif
```

```
Else    reset T_CWP_5 = 0 secs
Endif
```

Transition condition I (prio 3):

LV_CWP_NOT_PLAUS = false

Transition condition J (prio 2):

LV_CWP_TMP_OFF = false

Transition condition K (prio 1):

LV_CWP_OFF = true

Transition condition L (prio 2):

LV_CWP_TMP_OFF = true

Transition condition M (prio 2):

LV_CWP_TMP_OFF = true

Transition condition N (prio 1):

LV_CWP_OFF = false

<u>Additional conditions to determine the correct state:</u>

State	**additional condition**	**STATE_CWP_INT**
initialization	non	0
normal operation mode	LV_N_MON_CWP_DEAC = true LV_N_MON_CWP_DEAC = false	1 2
power reduced mode	TEMP_EL_CWP > C_TEMP_EL_CWP_MAX_THD_2 TEMP_EL_CWP > C_TEMP_EL_CWP_MAX_THD_1 V_CWP < C_V_CWP_MIN_DIAG_IT LV_CWP_PRE_LOCK = true	5 = prio 1 4 = prio 2 3 = prio 3 6 = prio 4

Internal 'Change Modus' conditions:

```
If       TEMP_EL_CWP > C_TEMP_EL_CWP_MAX_THD_2
Then     STATE_CWP_INT = 5
Else If  TEMP_EL_CWP > C_TEMP_EL_CWP_MAX_THD_1
    Then STATE_CWP_INT = 4
    Else If   V_CWP < C_V_CWP_MIN_DIAG_IT
        Then STATE_CWP_INT = 3
        Else If   LV_CWP_PRE_LOCK = 1
            Then STATE_CWP_INT = 6
            Else If   T_CWP_5 < C_T_V_CWP_MAX_DIAG
```

```
                              Then keep actual state STATE_CWP_INT
                              Endif
                          Endif
                      Endif
                  Endif
              Endif
```

unplausible operation	Non Reset timer T_CWP_4 and T_CWP_5	10
temporary shut-off	Reset timer T_CWP_4 and T_CWP_5 (LV_CWP_TMP_OFF = true **and** LV_CWP_BLOCK_DEAC = false) (LV_CWP_TMP_OFF = true **and** LV_CWP_BLOCK_DEAC = true)	 7 8
permanent shut-off	non	9

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEMP_EL_CWP_MAX_THD_1	1	0...FFH	-50...205	1	[°C]
first power reduction threshold temperature CWP electronics					
C_TEMP_EL_CWP_MAX_THD_2	1	0...FFH	-50...205	1	[°C]
second power reduction threshold temperature CWP electronics					
C_TEMP_EL_CWP_HYS	1	0...FFH	-50...205	1	[°C]
Electronics CWP temperature hysteresis					
C_V_CWP_HYS	1	0...FFH	0...25.5	0.1	[V]
voltage hysteresis					
C_T_V_CWP_MAX_DIAG	1	0...FFH	0...25.5	0.1	[s]
timer threshold (for transition condition "G")					
C_T_TEMP_EL_CWP_THD_1	1	0...FFH	0...25.5	0.1	[s]
timer threshold (for first transition condition "E")					
C_T_TEMP_EL_CWP_THD_2	1	0...FFH	0...25.5	0.1	[s]
timer threshold (for second transition condition "E")					
C_N_REL_CWP_SP_THD_MIN	1	0...FFH	0...255	1	[-]
Minimum CWP speed above the voltage monitoring is activated					
C_V_CWP_MIN_DIAG_IT	1	0...FFH	0...25.5	0.1	[V]
Min threshold voltage CWP to enable CWP internal state					

26.21 Electrical cooling fan diagnosis

26.21.1 Diagnostic information from infrastructure

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CDN_DIAG_ECFPWM_BAS[NC_ECF_NR]	V/O	0...7H	0...7	1	[-]
Primary diagnosis condition read from the infrastructure for each symptom of ECFPWM bit 0: diagnosis condition for symptom SYM_0 (SCP) bit 1: diagnosis condition for symptom SYM_1 (SCG) bit 2: diagnosis condition for symptom SYM_2 (OC)					
ERR_DIAG_ECFPWM[NC_ECF_NR]	V/O	0...7H	0...7	1	[-]
Raw value of error symptom for ECFPWM detected on interface ECU and ECF control unit bit 0: diagnosis condition for symptom SYM_0 (SCP) bit 1: diagnosis condition for symptom SYM_1 (SCG) bit 2: diagnosis condition for symptom SYM_2 (OC)					

Input data:

NC_ECF_NR	LV_IGK		

FUNCTION DESCRIPTION:

Description:

The ECF control unit is controlled by a PWM signal line driven by the ATIC 39. The driver can distinguish between three symptoms: ‘Short circuit to Vbatt’, ‘Short circuit to GND’ and ‘Open Load’.
This function reads the failure information (ERR_DIAG_ECFPWM[x]) from the infrastructure. The [x] is the placeholder for [NC_ECF_NR].

Application conditions:

Initialisation: reset or transition LV_IGK off --> on or reset of failure memory
CDN_DIAG_ECFPWM_BAS[x] = 0
ERR_DIAG_ECFPWM[x] = 0

Recurrence: 100ms
Activation: every ECU state

Formula section:

Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_ECFPWM[x] and diagnosis conditions CDN_DIAG_ECFPWM_BAS[x]) received from the infrastructure:

Diagnosis conditions are set according infrastructure info.: CDN_DIAG_ECFPWM_BAS[x]
Failure symptoms are set according infrastructure info.: ERR_DIAG_ECFPWM[x]

26.21.2 ECF control unit: Failure feedback recognition

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_ECFPWM_FB[NC_ECF_NR]	V/O	0...FFH	0...255	1	[-]
The failure status of the failure feedback coming from the ECF control unit 00H: no failure feedback 01H: failure 1 detected by ECF control unit FFH: init value / feedback possibleThe values 01H are only valid, if LV_ECFPWM_FB_VLD[x] = 1, else the f					
T_ECFPWM_FB[NC_ECF_NR]	V	0...FFH	0...25.5	0.1	[s]
Timer counting tlow for failure feedback recognition					
LV_ECFPWM_FB_VLD[NC_ECF_NR]	V/O	0...1H	0...1	1	[-]
If LV_ECFPWM_FB_VLD[x] = 1 the failure number ERR_ECFPWM_FB[x] (01H) is valid					

Input data:

LV_ERR_SPI_MPS	ERR_DIAG_ECFPWM[NC_ECF_NR]	LV_CDN_VB_MIN_DIAG	LV_IGK

FUNCTION DESCRIPTION:

If a failure is detected from the ECF control unit, a failure feedback is send to the ECU by pulling the signal line (interface ECU to the ECF control unit) to ground in a definited pattern. The time t_{low} is a identification for the failure.

Description:

If a failure is detected from the ECF control unit, a failure feedback is send to the ECU by pulling the signal line (interface ECU to the ECF control unit) to ground in a definited pattern. The time t_{low} is a identification for the failure.

For the recognition of the failure feedback the information, if a failure “short circuit to ground” is detected on the signal line between ECU and ECF control unit is used.

Application conditions:

Initialisation: reset or transition LV_IGK off --> on or reset of failure memory

ERR_ECFPWM_FB[x] = FFH

T_ECFPWM_FB[x] = 0

LV_ECFPWM_FB_VLD[x] = 0

Recurrence: 100ms

Activation: LV_CDN_VB_MIN_DIAG = 1 **and**

LV_ERR_SPI_MPS= 0

Deactivation: at deactivation function is initialized one time

Formula section:

```
LV_ECFPWM_FB_VLD[x] = 0
If   bit 1 of ERR_DIAG_ECFPWM[x] = 1 (SCG detected by HW)
Then      T_ECFPWM_FB[x] = T_ECFPWM_FB[x] + 100ms
          ERR_ECFPWM_FB[x] = FFH         (failure feedback possible)
          If     C_T_ECFPWM_FB_MAX[x]  < T_ECFPWM_FB[x]
                 (T_ECFPWM_FB[x] out of range for FB)
          Then   ERR_ECFPWM_FB[x]= 00H
          Endif
Else      If     C_T_ECFPWM_FB_MIN[x]   ≤   T_ECFPWM_FB[x]   ≤   C_T_ECFPWM_FB_MAX[x]
          Then   failure feedback detected
                 ERR_ECFPWM_FB[x] = 01H
                 LV_ECFPWM_FB_VLD[x] = 1
          Else   no failure feedback possible
                 ERR_ECFPWM_FB[x] = 00H
          Endif
          T_ECFPWM_FB[x] = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_ECFPWM_FB_MIN[NC_ECF_NR]	1	0...FFH	0...25.5	0.1	[s]
Time range necessary for identification of failure feedback minimal value					
C_T_ECFPWM_FB_MAX[NC_ECF_NR]	1	0...FFH	0...25.5	0.1	[s]
Time range necessary for identification of failure feedback maximal value					

26.21.3 Electrical diagnosis of ECF

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ECF_EL[NC_ECF_NR]	V/O	0...1H	0...1	1	[-]
Boolean for electronic controlled cooling fan signal range error					
LV_CDN_DIAG_ECF_EL[NC_ECF_NR]	V	0...1H	0...1	1	[-]
Boolean for electronic controlled cooling fan signal range diagnosis conditions					
ERR_SYM_ECF_EL[NC_ECF_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for electronic controlled cooling fan signal range error					
LV_END_DIAG_ECF_EL[NC_ECF_NR]	V	0...1H	0...1	1	[-]
Boolean for end of electronic controlled cooling fan signal range diagnosis					
CDN_DIAG_ECF_EL	V	0...FFH	0...255	1	[-]
Diagnosis condition for each symptom					

Input data:

LV_CDN_VB_MIN_DIAG	LV_ERR_SPI_MPS	ECFPWM[NC_ECF_NR]	CDN_DIAG_ECFPWM_BAS[NC_ECF_NR]
ERR_DIAG_ECFPWM[NC_ECF_NR]	ERR_ECFPWM_FB[NC_ECF_NR]		

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>,OUT<LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the signal from the driver, which controls the electrical cooling fan.

The input signal is a modulated control pulse (PWM).

Description:

The driver can distinguish between three symptoms: ‘Short circuit to Vbatt’, ‘Short circuit to GND’ and ‘Open Load’. The diagnosis of all these symptoms is possible at the same time in the duty cycle range C_ECFPWM_DIAG_MIN[[x] ... C_ECFPWM_DIAG_MAX[x].

The error detection algorithm in this function for “short circuit to GND” is respecting the status of the failure feedback recognition (ERR_ECFPWM_FB[x]). The failure, detected by the HW, is delivered from the infrastructure ERR_DIAG_ECFPWM[x].

Error-symptoms: are defined to this diagnosis function as following

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Vbatt          (= SYM_0)
 │ │ └───> Short circuit to GND            (= SYM_1)
 │ └─────> Open load                       (= SYM_2)
 └───────> -                               (= SYM_3)
```

Application conditions:

Initialization: according filter-type: MPL_STD_INI

Recurrence: 100ms

Activation: activation in all engine states

Formula section:

```
If      LV_CDN_VB_MIN_DIAG = 1
        and LV_ERR_SPI_MPS= 0
Then    If      Activation conditions are met for the NC_PSD_DLY_ECF_EL recurrence
        and     C_ECFPWM_DIAG_MIN[x] ≤ ECFPWM[NC_ECF_NR] ≤ C_ECFPWM_DIAG_MAX[x]
        and     ERR_ECFPWM_FB[x] = 00H (no failure feedback possible)
        Then    CDN_DIAG_ECF_EL[x] = CDN_DIAG_ECFPWM_BAS[x]
                ERR_DIAG_ECFPWM[x] is used for debounce mechanism
        Else    CDN_DIAG_ECF_EL[x] = 0 (diagnosis not possible)
        Endif
Else    CDN_DIAG_ECF_EL[x] = 0
Endif
```

Failure filtering and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_ECF_EL[x] and ERR_DIAG_ECFPWM[x].

```
ACTION_ERRM_FilterMulticondition    (IN<ECF_EL[x]>,
                                     IN<CDN_DIAG_ECF_EL[x] >,
                                     IN<ERR_DIAG_ECFPWM[x] >,
                                     IN<C_ABC_INC_ECF_EL[x] >,
                                     IN<C_ABC_INC_ECF_EL[x]>,
                                     OUT< LV_ERR_ECF_EL[X] >,
                                     SYNCRONIZATION<CALL>)
```

This algorithm determines:

ERR_SYM_ECF_EL[x]	(detected error symptom for ECF diagnosis)
LV_ERR_ECF_EL[x]	(Error flag for debounced error of ECF)
LV_CDN_DIAG_ECF_EL[x]	(Diagnosis condition information)
LV_END_DIAG_ECF_EL[x]	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ABC_INC_ECF_EL	1	0...FFH	0...255	1	[-]
Debounce counter increment ECF diagnosis					
C_ABC_MAX_ECF_EL	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - ECF diagnosis					
C_ECFPWM_DIAG_MIN[NC_ECF_NR]	1	0...FFH	0...99.60937	0.390625	[%]
Minimum duty cycle threshold of ECF_EL for electrical ECF diagnosis					
C_ECFPWM_DIAG_MAX[NC_ECF_NR]	1	0...FFH	0...99.60937	0.390625	[%]
Maximum duty cycle threshold of ECF_EL for electrical ECF diagnosis					

Configuration for diagnostic symptoms :

Diagnostic ECF_EL[NC_ECF_NR]	Symptom description	Symptom	Filter type
ECF _EL[NC_ECF_NR] Diagnostic	*SCP*	*SYM_0*	*MPL_STD_INI*
	SCG	*SYM_1*	
	OC	*SYM_2*	

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_ECF_EL	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (typical 2)					

26.21.4 ECF control unit: Failure feedback diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ECFPWM_FB[NC_ECF_NR]	V/O	0...1H	0...1	1	[-]
Error currently present, indicated by failure feedback from ECF control unit.					
LV_CDN_DIAG_ECFPWM_FB[NC_ECF_NR]	V	0...1H	0...1	1	[-]
Diagnosis condition ECF failure feedback diagnosis.					
ERR_SYM_ECFPWM_FB[NC_ECF_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom ECF failure feedback diagnosis.					
LV_END_DIAG_ECFPWM_FB[NC_ECF_NR]	V	0...1H	0...1	1	[-]
End of diagnosis ECF failure feedback diagnosis.					
ABC_ECFPWM_FB[NC_ECF_NR]	V	0...FFH	0...255	1	[-]
Antibounce counter for ECF failure feedback diagnosis.					
T_ACT_ECFPWM_FB[NC_ECF_NR]	V	0...FFFFH	0...6553.5	0.1	[s]
Time with ECFPWM[NC_ECF_NR] > C_EFPPWM_MIN_FB[NC_ECF_NR] necessary to have failure feedback from ECF control unit					

Input data:

LV_IGK	LV_CDN_VB_MIN_DIAG
ECFPWM[NC_ECF_NR]	ERR_ECFPWM_FB[NC_ECF_NR]
LV_ECFPWM_FB_VLD[NC_ECF_NR]	LV_ERR_SPI_MPS
TAM	

FUNCTION DESCRIPTION:

General information:

If a failure is detected by the ECF control unit, a failure feedback is send to the ECU.

Description:

According the possible failure feedback from the ECF control unit (defined by ERR_ECFPWM_FB[x] and LV_ECFPWM_FB_VLD[x]), the diagnosis is managing the following failure: LV_ERR_ECFPWM_FB[x].

Error-symptoms of failure LV_ERR_ECFPWM_FB[x]:

```
☐☐☐☐
 | | | └─> not used                          (= SYM_0)
 | | └───> not used                          (= SYM_1)
 | └─────> not used                          (= SYM_2)
 └───────> problem on ECF control unit detected   (= SYM_3)
```

Application conditions:

Initialization: reset *or transition LV_IGK off --> on* or reset of failure memory
all output = 0

Recurrence: 100ms

Activation: activation in all engine states

Formula section:

```
If   LV_CDN_VB_MIN_DIAG =1          and
     LV_ERR_SPI_MPS= 0
Then      LV_CDN_DIAG_ECFPWM_FB[x] = 1
          If    ECFPWM[x] > C_ECFPWM_MIN_FB[x]
          Then  T_ACT_ECFPWM_FB[x] = T_ACT_ECFPWM_FB[x] +100ms
                T_ACT_ECFPWM_FB[x] is limited to 6553.5s
          Endif
          If    LV_ERR_ECFPWM_FB[x] = 0
          Then  If    LV_ECFPWM_FB_VLD[x] = 1
                      (valid failure feedback information available)        and
                      TAM > C_TAM_MIN_ECFPWM_FB_DIAG_TH
                Then  ERR_SYM_ECFPWM_FB[x] = SYM_3
                      ABC_ECFPWM_FB[x] = ABC_ECFPWM_FB[x] + 1
                      If    ABC_ECFPWM_FB[x] ≥ C_ABC_ECFPWM_FB[x]
                      Then  LV_ERR_ECFPWM_FB[x] = 1
                            LV_END_DIAG_ECFPWM_FB[x] = 1
                      Endif
                Else  If    ERR_ECFPWM_FB[x] = 0    (no failure feedback detected)
                      Then  ERR_SYM_ECFPWM_FB[x] = 0
                      //please note: the LV_CDN_DIAG_ECFPWM_FB [x] should not be reset
                            If    T_ACT_ECFPWM_FB[x] ≥
                                  C_T_ACT_ECFPWM_FB[x]
                            Then  LV_END_DIAG_ECFPWM_FB[x] = 1
                            Endif
                      Endif
                Endif
          Else  LV_CDN_DIAG_ECFPWM_FB[x] = 0
          Endif
Else      LV_CDN_DIAG_ECFPWM_FB[x] = 0
```

Endif

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ABC_ECFPWM_FB[NC_ECF_NR]	1	1...FFH	1...255	1	[-]
If C_ABC_ECFPWM_FB[x] times sequentially a failure feedback for the same failure is detected, the failure LV_ERR_ECFPWM_FB[x] is set.					
C_ECFPWM_MIN_FB[NC_ECF_NR]	1	0...FFH	0...99.60937	0.390625	[%]
Minimum duty cycle of ECFPWM to have a failure feedback from ECF control unit					
C_T_ACT_ECFPWM_FB[NC_ECF_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time necessary to have failure feedback from ECF control unit					
C_TAM_MIN_ECFPWM_FB_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Minimum TAM for enabling ECFPWM_FB-diagnosis					

Configuration for diagnostic symptoms :

Diagnostic **ECFPWM_ERR_FB[NC_ECF_NR]**	**Symptom description**	**Symptom**	**Filter type**
failure feedback diagnosis of ECF control unit	see description	*SYM_0*	*NO*
		SYM_1	
		SYM_2	
		SYM_3	

26.22 Electronically controlled thermostat diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CDN_DIAG_ECT_EL	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of ECT bit 0: diagnosis condition for symptom SYM_0 bit 1: diagnosis condition for symptom SYM_1 bit 2: diagnosis condition for symptom SYM_2					
ERR_DIAG_ECT_EL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for ECT (only parameter)					
ERR_SYM_ECT_EL_OC	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for electrical thermostat open circuit error.					
ERR_SYM_ECT_EL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for ECT (only parameter) - SCG					
ERR_SYM_ECT_EL_SCP	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for electrical thermostat short circuit to battery error.					
LV_CDN_DIAG_ECT_EL_OC	V	0...1H	0...1	1	[-]
Boolean for electrical thermostat open circuit diagnosis conditions.					
LV_CDN_DIAG_ECT_EL	V	0...1H	0...1	1	[-]
Boolean for electrical thermostat short circuit to ground diagnosis conditions.					
LV_CDN_DIAG_ECT_EL_SCP	V	0...1H	0...1	1	[-]
Boolean for electrical thermostat short circuit to battery diagnosis conditions.					
LV_END_DIAG_ECT_EL_OC	V	0...1H	0...1	1	[-]
Boolean for end electrical thermostat open circuit diagnosis.					
LV_END_DIAG_ECT_EL	V	0...1H	0...1	1	[-]
Boolean for end of electrical thermostat diagnosis - SCG					
LV_END_DIAG_ECT_EL_SCP	V	0...1H	0...1	1	[-]
Boolean for end of electrical thermostat short circuit to battery diagnosis.					
LV_ERR_ECT	O/V	0...1H	0...1	1	[-]
Boolean for electronic controlled thermostat error					
LV_ERR_ECT_EL_OC	O/V	0...1H	0...1	1	[-]
Boolean electrical thermostat open circuit error.					
LV_ERR_ECT_EL	O/V	0...1H	0...1	1	[-]
ECT Electrical error detected - SCG					
LV_ERR_ECT_EL_SCP	O/V	0...1H	0...1	1	[-]
Boolean electrical thermostat short circuit to battery error.					

Input data:

ECTPWM	LV_CDN_VB_OBD1	LV_ERR_ECT_MEC	LV_IGK
LV_INH_DIAG_ECT_EL			

Import actions:

ACTION_ERRM_FilterSymptom(IN< *XX* >, IN< CDN_DIAG_XX>, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX >, IN<C_ABC_MAX_XX>, OUT< LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result

FUNCTION DESCRIPTION:

General information:

A global electronically controlled thermostat error (LV_ERR_ECT) is set without debounce as soon as SCG (LV_ERR_ECT_EL) or a mechanical (LV_ERR_ECT_MEC) thermostat failure is detected. The activation of the mechanical thermostat diagnosis (ECT rationality check) is not configurable with a configuration switch. Both, the electrical- and mechanical thermostat diagnosis function are performed if the vehicle is equipped with an electrically controlled thermostat.

Application conditions:

Initialisation at Reset: LV_ERR_ECT = 0

Recurrence: 500 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

```
If      LV_ERR_ECT_EL = 1          or
        LV_ERR_ECT_MEC = 1
then    LV_ERR_ECT = 1             (without debounce)
else    LV_ERR_ECT = 0
endif
```

26.22.1 ECT diagnosis (Electrical diagnostic with PWM output -- Multicondition filter)

FUNCTION DESCRIPTION:

General information:

The ECT is driven by the ECU via an output driver. The failure detection is done by ECU Hardware.

general specification

The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements.

Description:

The purpose is to perform the electrical diagnosis of the ECT actuator. 3 symptoms are distinguished, each having its own error location:

Error-symptoms and conditions: are defined to this diagnosis function as following

```
☐☐☐☐
 │ │ │ └─> Short circuit to battery “SCP”                    (= SYM_0)
 │ │ └───> (not used)                                        (= SYM_1)
 │ └─────> (not used)                                        (= SYM_2)
 └───────> (not used)                                        (= SYM_3)
```

```
☐☐☐☐
 │ │ │ └─> (not used)                                        (= SYM_0)
 │ │ └───> Short circuit to Ground “SCG”                     (= SYM_1)
 │ └─────> (not used)                                        (= SYM_2)
 └───────> (not used)                                        (= SYM_3)
```

```
☐☐☐☐
 │ │ │ └─> (not used)                                        (= SYM_0)
 │ │ └───> (not used)                                        (= SYM_1)
 │ └─────> Open Circuit “OC”                                 (= SYM_2)
 └───────> (not used)                                        (= SYM_3)
```

Application conditions:

Initialisation:	all ERRM data are initialized according to Filter-type. At Reset: CDN_DIAG_ECT_EL = 0
Recurrence:	500 ms
Activation:	every engine operating state
Deactivation:	-

Formula section:

Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_ECT_EL and diagnosis conditions CDN_DIAG_ECT_EL) received from the infrastructure:

Diagnosis conditions are set according infrastructure information: CDN_DIAG_ECT_EL
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_ECT_EL

If LV_IGK = 1 (Activation conditions)
and LV_CDN_VB_OBD1 = 1
and LV_INH_DIAG_ECT_EL = 0

Then

```
          If      Activation conditions are met for the NC_PSD_DLY_ECT_EL recurrence
          Then    Additional diagnosis conditions:
                  If      bit 0 of CDN_DIAG_ECT_EL = 1      (info from infrastructure)
                          and     C_ECT_EL_PWM_DIAG_MIN_SCP <= ECTPWM
                  then
                          bit 0 of CDN_DIAG_ECT_EL  = 1     (Diagnosis of SCP possible)
                  Endif
                  If      bit 1 of CDN_DIAG_ECT_EL = 1      (info from infrastructure)
                          and     ECTPWM <= C_ECT_EL_PWM_DIAG_MAX_SCG
                  then
                          bit 1 of CDN_DIAG_ECT_EL = 1      (Diagnosis of SCG possible)
                  Endif
                  If      bit 2 of CDN_DIAG_ECT_EL = 1      (info from infrastructure)
                          and     C_ECT_EL_PWM_DIAG_MIN_OC <= ECTPWM <=
                                  C_ECT_EL_PWM_DIAG_MAX_SCG
                  then
                          bit 2 of CDN_DIAG_ECT_EL  = 1     (Diagnosis of OC possible)
                  Endif
          Else
                  CDN_DIAG_ECT_EL = 0
          Endif
Else
          CDN_DIAG_ECT_EL = 0
Endif


If        bit 0 of CDN_DIAG_ECT_EL = 1      //SCP
Then      LV_CDN_DIAG_ECT_EL_SCP = 1
Else      LV_CDN_DIAG_ECT_EL_SCP = 0
Endif
If        bit 1 of CDN_DIAG_ECT_EL = 1      //SCG
Then      LV_CDN_DIAG_ECT_EL = 1
Else      LV_CDN_DIAG_ECT_EL = 0
Endif
If        bit 2 of CDN_DIAG_ECT_EL = 1      //OC
Then      LV_CDN_DIAG_ECT_EL_OC = 1
Else      LV_CDN_DIAG_ECT_EL_OC = 0
Endif
If        bit 0 of ERR_DIAG_ECT_EL = 1      //SCP
Then      ERR_SYM_ECT_EL_SCP = SYM_0
Else      ERR_SYM_ECT_EL_SCP = NO_SYM
Endif
If        bit 1 of ERR_DIAG_ECT_EL = 1      //SCG
```

```
Then    ERR_SYM_ECT_EL = SYM_1
Else    ERR_SYM_ECT_EL = NO_SYM
Endif
If      bit 2 of ERR_DIAG_ECT_EL = 1     //OC
Then    ERR_SYM_ECT_EL_OC = SYM_2
Else    ERR_SYM_ECT_EL_OC = NO_SYM
Endif
```

LV_ERR_ECT_EL(_XX) and LV_END_DIAG_ECT_EL(_XX) are calculated by error management.

Failure filtering and error management treatment:

For failure and error management treatment the anti-bounce mechanisms are called with the parameters ERR_SYM_ECT_EL(_XX) and LV_CDN_DIAG_ECT_EL(_XX).

ACTION_ERRM_FilterSymptom (IN<ECT_EL_XX>, IN<LV_CDN_DIAG_ECT_EL(_XX)>, IN<LV_ERR_ECT_EL(_XX)>, IN<C_ABC_INC_ECT_EL(_XX)>, IN<C_ABC_MAX_ECT_EL(_XX)>, OUT< LV_ERR_ECT_EL(_XX)>, SYNCRONIZATION<CALL>)

This algorithm determines:

ERR_SYM_ECT_EL(_XX)	(detected error symptom for ECT diagnosis)
LV_ERR_ECT_EL(_XX)	(Error flag for debounced error of ECT)
LV_CDN_DIAG_ECT_EL(_XX)	(Diagnosis condition information)
LV_END_DIAG_ECT_EL(_XX)	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_ECT_EL_OC	1	0...FFH	0...255	1	[-]
Antibounce counter increment, electronical controlled thermostat diagnosis					
C_ABC_INC_ECT_EL	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_INC_ECT_EL_SCP	1	0...FFH	0...255	1	[-]
Antibounce counter increment, electronical controlled thermostat diagnosis					
C_ABC_MAX_ECT_EL_OC	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter, electronical controlled thermostat diagnosis					
C_ABC_MAX_ECT_EL	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					
C_ABC_MAX_ECT_EL_SCP	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter, electronical controlled thermostat diagnosis					
C_ECT_EL_PWM_DIAG_MAX_SCG	1	0...FFH	0...99.60937	0.390625	[%]
Maximum threshold for SCG diagnosis window					
C_ECT_EL_PWM_DIAG_MIN_OC	1	0...FFH	0...99.60937	0.390625	[%]
Minimum threshold for OC diagnosis window					
C_ECT_EL_PWM_DIAG_MIN_SCP	1	0...FFH	0...99.60937	0.390625	[%]
Minimum threshold for SCP diagnosis window					

Error treatment

Diagnostic Instance		**Symptoms**		**ABC type**	**CARB class**
Description	**Name (XX)**	**Description**	**Nr**		
Short to plus of electronical controlled thermostat	*ECT_EL_SCP*	*SCP*	0	STD_INI	CC
		-	1		
		-	2		
		-	3		

Diagnostic Instance		**Symptoms**		**ABC type**	**CARB class**
Description	**Name (XX)**	**Description**	**Nr**		
Short to ground of electronical controlled thermostat	*ECT_EL*	-	0	STD_INI	CC
		SCG	1		
		-	2		
		-	3		

Diagnostic Instance		**Symptoms**		**ABC type**	**CARB class**
Description	**Name (XX)**	**Description**	**Nr**		
Open circuit of electronical controlled thermostat	*ECT_EL_OC*	-	0	STD_INI	CC
		-	1		
		OC	2		
		-	3		

Fields information (For more information refers to Error Management file “Table of Failure”) :

- ABC Type : STD_INI, STD, MEM, MEM_INI, DEC_CAL, STC, NO, MPL_STD_INI
- CARB Class : MIS, FSD, CC, CAT, HC, EVAP, SA, AC, LS, LSH, EGR, OTHER, NO.

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_ECT_EL	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (typical 2)					

26.23 Electronically controlled thermostat diagnosis (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_ECT_EL	V/O	0...1H	0...1	1	[-]
Boolean for electronically controlled thermostat signal range diagnosis inhibit					

Input data:

LV_IGK	LV_ERR_SPI_MPS		

FUNCTION DESCRIPTION:

General information:

Depending on project specific requirements, the electrical diagnosis of the electronically controlled thermostat can be inhibited by setting of LV_INH_DIAG_ECT_EL.

Application conditions:

Initialisation: 0 at LV_IGK 0->1 or Reset

Recurrence: 500 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

```
IF      LV_ERR_SPI_MPS = 0
THEN    LV_INH_DIAG_ECT_EL = 0
ELSE    LV_INH_DIAG_ECT_EL = 1
ENDIF
```

26.24 Electronically Controlled Radiator Shutter (ECRAS) Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ECRAS	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on ECRAS command signal.					
CDN_DIAG_ECRASPWM_BAS	O	0...7H	0...7	1	[-]
Primary diagnosis condition read from the infrastructure for each symptom of ECRASPWM bit 0: diagnosis condition for symptom SYM_0 (SCP) bit 1: diagnosis condition for symptom SYM_1 (SCG) bit 2: diagnosis condition for symptom SYM_2 (OC)					
ERR_DIAG_ECRASPWM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for ECRASPWM detected on interface ECU and ECRAS control unit bit 0: diagnosis condition for symptom SYM_0 (SCP) bit 1: diagnosis condition for symptom SYM_1 (SCG) bit 2: diagnosis condition for symptom SYM_2 (OC)					
ERR_ECRASPWM_FB	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
The failure status of the failure feedback coming from the ECRAS control unit 00H: no failure feedback 01H: failure 1 detected by ECRAS control unit FFH: init value / feedback p					
T_ECRASPWM_FB	V	0...FFH	0...25.5	0.1	[s]
Timer counting tlow for failure feedback recognition					
LV_ECRAS_FB_DET	V	0...1H	0...1	1	[-]
Change from short circuit to ground to normal PWM signal detected (bit 1 ERR_DIAG_ECRASPWM 1->0)					
STATE_ECRAS_FB	V	0H 1H 2H 3H 4H 5H 6H	NO_FB DIAG_ABILITY EL_ERR_UP MEC_ERR_UP EL_ERR_DOWN EL_ERR_UP_AND_DOWN MEC_ERR_UP_AND_EL_ERR_DOWN	1	[-]
State of the received feedback from ECRAS control unit					
LV_ERR_ECRAS_UP_FB	V/O	0...1H	0...1	1	[-]
ECRAS_UP error currently present, indicated by failure feedback from ECRAS control unit.					
LV_CDN_DIAG_ECRAS_UP_FB	V	0...1H	0...1	1	[-]
Diagnosis condition ECRAS_UP failure feedback diagnosis.					
ERR_SYM_ECRAS_UP_FB	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom ECRAS_UP failure feedback diagnosis.					
LV_END_DIAG_ECRAS_UP_FB	V/O	0...1H	0...1	1	[-]
End of diagnosis ECRAS_UP failure feedback diagnosis.					
LV_ERR_ECRAS_DOWN_FB	V/O	0...1H	0...1	1	[-]
ECRAS_DOWN error currently present, indicated by failure feedback from ECRAS control unit.					
LV_CDN_DIAG_ECRAS_DOWN_FB	V	0...1H	0...1	1	[-]
Diagnosis condition ECRAS_DOWN failure feedback diagnosis.					

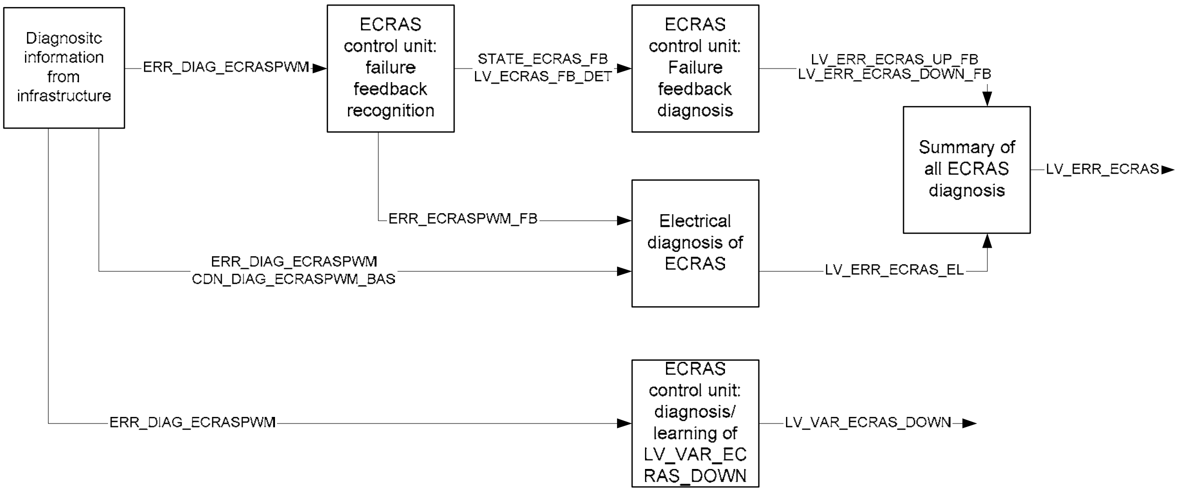
ERR_SYM_ECRAS_DOWN_FB	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom ECRAS_DOWN failure feedback diagnosis.					
LV_END_DIAG_ECRAS_DOWN_FB	V/O	0...1H	0...1	1	[-]
End of diagnosis ECRAS_DOWN failure feedback diagnosis.					
LV_DIAG_ECRAS	V/O	0...1H	0...1	1	[-]
Diagnosis mode of ECRAS control unit activated (10% PWM)					
LV_DIAG_ECRAS_END	V/O	0...1H	0...1	1	[-]
Diagnosis mode feedback of ECRAS control unit received (diagnosis ability of ECRAS detected)					
LV_EDGE_ECRAS_UP	V	0...1H	0...1	1	[-]
Detected change in LV_ECRAS_UP_1					
LV_EDGE_ECRAS_DOWN	V	0...1H	0...1	1	[-]
Detected change in LV_ECRAS_DOWN_1					
T_DIAG_ECRAS	V	0...FFH	0...25.5	0.1	[s]
Timer for waiting of answer of ECRAS control unit					
LV_ECRAS_FB_DIAG	V	0...1H	0...1	1	[-]
Feedback of ECRAS control unit or timeout detected					
LV_ECRAS_DOWN_DET_RST	V	0...1H	0...1	1	[-]
Reset learning of ECRAS_DOWN					
LV_DIAG_ECRAS_ERR_END	V/O	0...1H	0...1	1	[-]
Diagnosis mode feedback of ECRAS control unit not received (diagnosis ability of ECRAS not detected)					
LV_VAR_ECRAS_DOWN	V/O/S	0...1H	0...1	1	[-]
ECRAS_DOWN flap of ECRAS system detected (2 flap system)					
LV_ECRAS_DOWN_DET_SET	V/O/S	0...1H	0...1	1	[-]
Diagnosis / learning of ECRAS_DOWN run through one-time					
T_ECRAS_DOWN_DET	V	0...FFH	0...25.5	0.1	[s]
Timer for learning of ECRAS_DOWN					
LV_ECRAS_DOWN_DET_ACT	V/O	0...1H	0...1	1	[-]
Bit indicating learning of ECRAS_DOWN					
CTR_ECRAS_DOWN_DET	V	0...FFH	0...255	1	[-]
Counter for detecting ECRAS_DOWN					
T_ECRAS_DOWN_DET_NEG	V	0...FFH	0...25.5	0.1	[s]
Timer indicating time since start of short circuit to ground phase of ECRASPWM					
T_ECRAS_DOWN_DET_POS	V	0...FFH	0...25.5	0.1	[s]
Timer indicating time since start of normal ECRASPWM signal (no SCG)					
LV_ERR_ECRAS_EL	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on ECRAS command signal					
LV_CDN_DIAG_ECRAS_EL	V	0...1H	0...1	1	[-]
Diagnosis condition ECRAS diagnosis					
ERR_SYM_ECRAS_EL	V	0...4H	0...4	1	[-]
Error symptom ECRAS diagnosis					
LV_END_DIAG_ECRAS_EL	V/O	0...1H	0...1	1	[-]
End of diagnosis ECRAS diagnosis					
CDN_DIAG_ECRAS_EL	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of ECRAS_EL bit 0: diagnosis condition for symptom SYM_0 (SCP) bit 1: diagnosis condition for symptom SYM_1 (SCG) bit 2: diagnosis condition for symptom SYM_2 (OC)					
LV_INTR_DIAG_ECRAS_EL	V/O	0...1H	0...1	1	[-]
Boolean for irreverible interrupt of ECRAS_EL diagnosis					

Input data:

LV_VAR_ECRAS_UP	LV_IGK	LV_ERR_SPI_MPS	LV_CDN_VB_MIN_DIAG
ECRASPWM	VS	LV_ACT_ECRAS_EOL_EXT_ADJ	LV_ST_END
LV_ECRAS_UP_1	LV_ECRAS_DOWN_1		

General information:

This modul consists of 6 parts. The parts and their correlation is described in the figure below. The diagnostic information from the infrastructure is used for detection of feedback from the ECRAS control unit and for the electrical diagnosis. The information from the infrastructure is also used for the diagnosis / learning of the ECRAS variant with the second flap (LV_VAR_ECRAS_DOWN). The recognised feedbacks from the ECRAS control unit are processed by the failure feedback diagnosis.



26.24.1 Summary of all ECRAS diagnosis

FUNCTION DESCRIPTION:

Description:

This function shall summarize all ECRAS failures in one flag only.

Application conditions:

Initialisation: reset **or** reset of failure memory **or** LV_IGK = 0 -> 1

all = 0

Recurrence: 100ms

Activation: LV_VAR_ECRAS_UP = 1

Formula section:

If LV_ERR_ECRAS_EL = 1 **or**
LV_ERR_ECRAS_UP_FB = 1 **or**
LV_ERR_ECRAS_DOWN_FB = 1

Then LV_ERR_ECRAS = 1

Else LV_ERR_ECRAS = 0

Endif

26.24.2 Diagnostic information from infrastructure

FUNCTION DESCRIPTION:

Description:

The ECRAS control unit is controlled by a PWM signal line driven by the ATIC 39. This driver is able to distinguish between three symptoms: 'Short circuit to Vbatt', 'Short circuit to GND' and 'Open Load'.

This function shall reads the failure information (ERR_DIAG_ECRASPWM) delivered from the infrastructure (basic software).

Application conditions:

Initialisation: reset **or** reset of failure memory **or** LV_IGK = 0 -> 1

all = 0

Recurrence: 100ms

Activation: LV_VAR_ECRAS_UP = 1

Formula section:

Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_ECRASPWM and diagnosis conditions CDN_DIAG_ECRASPWM_BAS) received from the infrastructure:

Diagnosis conditions are set according infrastructure info.: CDN_DIAG_ECRASPWM_BAS
Failure symptoms are set according infrastructure info.: ERR_DIAG_ECRASPWM

26.24.3 ECRAS control unit: Failure feedback recognition

FUNCTION DESCRIPTION:

Description:

If a failure is detected from the ECRAS control unit, a failure feedback is send to the ECU by pulling the signal line (interface ECU to the ECRAS control unit) to ground in a definited pattern. The time t_{low} is a identification for the failure. Depending on t_{low} a specific failure is identified and set in STATE_ECRAS_FB.

For the recognition of the failure feedback the information, if a failure "short circuit to ground" is detected on the signal line between ECU and ECRAS control unit is used.

Application conditions:

Initialisation: reset **or** reset of failure memory **or** LV_IGK = 0 -> 1

ERR_ECRASPWM_FB = FFH, all others = 0

Recurrence: 100ms

Activation: LV_CDN_VB_MIN_DIAG = 1 **and**
LV_ERR_SPI_MPS= 0 **and**
LV_VAR_ECRAS_UP = 1 **and**
LV_IGK = 1

Deactivation: at deactivation function is initialized one time

Formula section:

```
STATE_ECRAS_FB      = 0H
LV_ECRAS_FB_DET     = 0

If    bit 1 of ERR_DIAG_ECRASPWM  = 1 (SCG detected by HW)
Then        T_ECRASPWM_FB  = T_ECRASPWM_FB  + 100ms
            ERR_ECRASPWM_FB  = FFH       (ECRAS ECU failure possible)
            If      T_ECRASPWM_FB > C_T_ECRASPWM_FB_MAX
                    (T_ECRASPWM_FB  out of range for FB, C_T_ECRASPWM_FB_MAX should be
                    > than C_ T_ECRASPWM_FB_MAX_6, DIAG_EL will be enabled)
            Then    ERR_ECRASPWM_FB = 00H
            Endif
Else        If      T_ECRASPWM_FB > 0     //change from SCG to normal detected
            then    LV_ECRAS_FB_DET = 1
            endif
            If      C_T_ECRASPWM_FB_MIN_1   ≤ T_ECRASPWM_FB  ≤
                                              C_T_ECRASPWM_FB_MAX_1
            Then    ECRAS feedback detected: diagnosis abilityof ECRAS
                    ERR_ECRASPWM_FB  = 01H
                    STATE_ECRAS_FB = 01H
            Elseif  C_T_ECRASPWM_FB_MIN_2   ≤ T_ECRASPWM_FB  ≤
                                              C_T_ECRASPWM_FB_MAX_2
            Then    ECRAS feedback detected: elctrical failure ECRAS_UP
                    ERR_ECRASPWM_FB  = 01H
                    STATE_ECRAS_FB = 02H
            Elseif  C_T_ECRASPWM_FB_MIN_3   ≤ T_ECRASPWM_FB  ≤
                                              C_T_ECRASPWM_FB_MAX_3
            Then    ECRAS feedback detected: mechanical failure ECRAS_UP
                    ERR_ECRASPWM_FB  = 01H
                    STATE_ECRAS_FB = 03H
            Elseif  C_T_ECRASPWM_FB_MIN_4   ≤ T_ECRASPWM_FB  ≤
                                              C_T_ECRASPWM_FB_MAX_4
            Then    ECRAS feedback detected: electrical failure ECRAS_DOWN
                    ERR_ECRASPWM_FB  = 01H
                    STATE_ECRAS_FB = 04H
            Elseif  C_T_ECRASPWM_FB_MIN_5   ≤ T_ECRASPWM_FB  ≤
                                              C_T_ECRASPWM_FB_MAX_5
            Then    ECRAS feedback detected: elctrical failure ECRAS_UP/ _DOWN
```

```
            ERR_ECRASPWM_FB  = 01H
            STATE_ECRAS_FB = 05H
    Elseif  C_T_ECRASPWM_FB_MIN_6  ≤ T_ECRASPWM_FB  ≤
                                          C_T_ECRASPWM_FB_MAX_6
    Then    ECRAS feedb. detected: mech. failure ECRAS_UP, elctr. ECRAS_DOWN
            ERR_ECRASPWM_FB  = 01H
            STATE_ECRAS_FB = 06H
    Else    no ECRAS ECU failure possible
            ERR_ECRASPWM_FB  = 00H
    Endif
    T_ECRASPWM_FB  = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_ECRASPWM_FB_MAX	1	0...FFH	0...25.5	0.1	[s]
C_T_ECRASPWM_FB_MAX					
C_T_ECRASPWM_FB_MIN_1	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'diagnosis ability', min value					
C_T_ECRASPWM_FB_MAX_1	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'diagnosis ability', max value					
C_T_ECRASPWM_FB_MIN_2	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'elctrical failure ECRAS_UP', min value					
C_T_ECRASPWM_FB_MAX_2	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'elctrical failure ECRAS_UP', max value					
C_T_ECRASPWM_FB_MIN_3	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'mechanical failure ECRAS_UP', min value					
C_T_ECRASPWM_FB_MAX_3	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'mechanical failure ECRAS_UP', max value					
C_T_ECRASPWM_FB_MIN_4	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'electrical failure ECRAS_DOWN', min value					
C_T_ECRASPWM_FB_MAX_4	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'electrical failure ECRAS_DOWN', max value					
C_T_ECRASPWM_FB_MIN_5	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'electrical failure ECRAS_UP/_DOWN', min value					
C_T_ECRASPWM_FB_MAX_5	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'electrical failure ECRAS_UP/_DOWN', max value					
C_T_ECRASPWM_FB_MIN_6	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'mech. failure ECRAS_UP, el. ECRAS_DOWN', min value					
C_T_ECRASPWM_FB_MAX_6	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback 'mech. failure ECRAS_UP, el. ECRAS_DOWN', max value					

26.24.4 Electrical diagnosis of ECRAS

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>,OUT<LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the signal from the driver, which controls the electrical cooling fan.

The input signal is a modulated control pulse (PWM).

Description:

The driver can distinguish between three symptoms: ‘Short circuit to Vbatt’, ‘Short circuit to GND’ and ‘Open Load’.
The error detection algorithm in this function for “short circuit to GND” is respecting the status of the failure feedback recognition (ERR_ECRASPWM_FB). The failure, detected by the HW, is delivered from the infrastructure ERR_DIAG_ECRASPWM .

Error-symptoms: are defined to this diagnosis function as following

```
□□□□
 | | | └─> Short circuit to Vbatt        (= SYM_0)
 | | └───> Short circuit to GND          (= SYM_1)
 | └─────> Open load                     (= SYM_2)
 └───────> -                             (= SYM_3)
```

Application conditions:

Initialization: according filter-type: **MPL_STD_INI**

LV_INTR_DIAG_ECRAS_EL = 0 at LV_IGK 0->1 **or** reset **or** clear FMY

Recurrence: 100ms

Activation: LV_VAR_ECRAS_UP = 1

Deactivation: if LV_INTR_DIAG_ECRAS_EL = 1, set CDN_DIAG_ECRAS_EL = 0
if LV_VAR_ECRAS_UP = 0, set LV_END_DIAG_ECRAS_EL = 1

Formula section:

```
If      VS >= C_VS_MAX_ECRAS_DIAG
Then    LV_INTR_DIAG_ECRAS_EL = 1        //remains active until next initialization
Endif
```

```
If        LV_CDN_VB_MIN_DIAG = 1
    and   LV_ERR_SPI_MPS= 0
    and   LV_IGK = 1
Then
    Note: The failure symptoms ERR_SYM_ECRAS_EL and the diagnosis conditions
    CDN_DIAG_ECRAS_EL (raw values) shall be set according infrastructure information
    ERR_DIAG_ECRASPWM and CDN_DIAG_ECRASPWM_BAS.

    If    Activation conditions are met for the NC_PSD_DLY_ECRAS_EL recurrence
    And   ERR_ECRASPWM_FB = 00h
    Then

        { Additional diagnosis conditions }

        If  ECRASPWM < C_ECRASPWM_DIAG_MIN_SCP
                { condition that disables the SYM_0 detection }
        Then
            bit 0 of CDN_DIAG_ECRAS_EL  = 0   { Diagnosis of SCP is not possible }
        Endif
        If  ECRASPWM > C_ECRASPWM_DIAG_MAX_SCG
                { condition that disables the SYM_1 detection }
        Then
            bit 1 of CDN_DIAG_ECRAS_EL = 0    { Diagnosis of SCG is not possible }
        Endif
        If  ECRASPWM < C_ECRASPWM_DIAG_MIN_OC
            or ECRASPWM > C_ECRASPWM_DIAG_MAX_SCG
                { conditions that disable the SYM_2 detection }
        Then
            bit 2 of CDN_DIAG_ECRAS_EL  = 0   { Diagnosis of OC is not possible }
        Endif
    Else
        CDN_DIAG_ECRAS_EL = 0
    Endif
Else
        CDN_DIAG_ECRAS_EL = 0
Endif
```

Failure filtering and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_ECRAS_EL and ERR_DIAG_ECRASPWM .

```
ACTION_ERRM_FilterMulticondition     (IN<ECRAS_EL >,
                                      IN<CDN_DIAG_ECRAS_EL  >,
                                      IN<ERR_DIAG_ECRASPWM  >,
                                      IN<C_ABC_INC_ECRAS_EL  >,
                                      IN<C_ABC_MAX_ECRAS_EL >,
                                      OUT< LV_ERR_ECRAS_EL  >,
                                      SYNCRONIZATION<CALL>)
```

This algorithm determines:

ERR_SYM_ECRAS_EL	(detected error symptom for ECRAS diagnosis)

LV_ERR_ECRAS_EL	(Error flag for debounced error of ECRAS)
LV_CDN_DIAG_ECRAS_EL	(Diagnosis condition information)
LV_END_DIAG_ECRAS_EL	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_ECRAS_EL	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_ECRAS_EL	1	1...FFH	1...255	1	[-]
C_ABC_MAX_ECRAS_EL					
C_ECRASPWM_DIAG_MIN_SCP	1	0...FFH	0...99.60937	0.390625	[%]
C_ECRASPWM_DIAG_MIN_SCP					
C_ECRASPWM_DIAG_MAX_SCG	1	0...FFH	0...99.60937	0.390625	[%]
C_ECRASPWM_DIAG_MAX_SCG					
C_ECRASPWM_DIAG_MIN_OC	1	0...FFH	0...99.60937	0.390625	[%]
C_ECRASPWM_DIAG_MIN_OC					
C_VS_MAX_ECRAS_DIAG	1	0...FFH	0...255	1	[km/h]
Maximum VS treshold for deactivation for electrical ECRAS diagnosis					

Configuration for diagnostic symptoms :

Diagnostic ECRAS_EL	**Symptom description**	**Symptom**	**Filter type**
ECRAS _EL Diagnostic	*SCP*	*SYM_0*	*MPL_STD_INI*
	SCG	*SYM_1*	
	OC	*SYM_2*	

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_ECRAS_EL	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (typical 2)					

26.24.5 ECRAS control unit: Failure feedback diagnosis

FUNCTION DESCRIPTION:

General information:

If a failure is detected by the ECRAS control unit (ECRAS_UP), a failure feedback is send to the ECU and recognized.

Description:

According the possible failure feedback from the ECRAS control unit (defined by ERR_ECRASPWM_FB and STATE_ECRAS_FB), the diagnosis is managing the following failure: LV_ERR_ECRAS_UP_FB, LV_ERR_ECRAS_DOWN_FB. Additional the diagnosis ability of the ECRAS control unit will be diagnosed (LV_DIAG_ECRAS / _END).

Error-symptoms of failure LV_ERR_ECRAS_UP_FB :


```
 │ │ │ └─> electrical failure                    (= SYM_0)
 │ │ └───> mechanical failure                    (= SYM_1)
 │ └─────> not used                              (= SYM_2)
 └───────> no answer of ECRAS                    (= SYM_3)
```

Error-symptoms of failure LV_ERR_ECRAS_DOWN_FB :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> electrical failure                    (= SYM_0)
 │ │ └───> not used                              (= SYM_1)
 │ └─────> not used                              (= SYM_2)
 └───────> not used                              (= SYM_3)
```

Application conditions:

Initialization:	according filter type **STD_INI**, all others = 0 at reset **or** reset of failure memory **or** LV_IGK = 0 -> 1
Recurrence:	100ms
Activation:	LV_VAR_ECRAS_UP = 1 **and** (ERR_SYM_ECRAS_UP_FB != SYM_3 **or** LV_ERR_ECRAS_UP_FB =0)
Deactivation:	if LV_VAR_ECRAS_UP=0, set LV_END_DIAG_ECRAS_UP_FB = 1 and LV_END_DIAG_ECRAS_DOWN_FB = 1 if (LV_ERR_ECRAS_UP_FB = 1 and ERR_SYM_ECRAS_UP_FB = Sym_3), set LV_CDN_DIAG_ECRAS_UP_FB = 0 and LV_CDN_DIAG_ECRAS_DOWN_FB = 0

Formula section:

```
IF   LV_CDN_VB_MIN_DIAG = 1         and
     LV_ERR_SPI_MPS= 0              and
     LV_ERR_ECRAS_EL = 0            and
     LV_IGK = 1
THEN
     if      LV_ECRAS_UP_1  0 -> 1  or  1 -> 0
     then    LV_EDGE_ECRAS_UP = 1
     endif

     if      LV_ECRAS_DOWN_1  0 -> 1  or  1 -> 0
     then    LV_EDGE_ECRAS_DOWN = 1
     endif

     // activated diagnosis-mode of ECRAS_UP (PWM = 10%)
     if       (LV_ACT_ECRAS_EOL_EXT_ADJ = 1 or LV_ST_END = 1)
              and LV_DIAG_ECRAS_END = 0 and LV_DIAG_ECRAS_ERR_END = 0
             and LV_EDGE_ECRAS_UP = 0   and LV_EDGE_ECRAS_DOWN = 0
     then    LV_DIAG_ECRAS = 1
     endif
```

```
// calculating timer
if      LV_DIAG_ECRAS = 1  or  LV_EDGE_ECRAS_UP = 1  or
                                        LV_EDGE_ECRAS_DOWN = 1
then    T_DIAG_ECRAS = T_DIAG_ECRAS + 100ms
endif

// setting feedback diagnosis enabled -> required to set LV_CDN_DIAG_XXX
if      LV_ECRAS_FB_DET = 1  or  T_DIAG_ECRAS >= C_T_DIAG_ECRAS
then    LV_ECRAS_FB_DIAG = 1
else    LV_ECRAS_FB_DIAG = 0
endif

// setting of the diagnosis conditions
if      LV_ECRAS_FB_DIAG = 1  and  (LV_EDGE_ECRAS_UP = 1  or
                                        LV_DIAG_ECRAS = 1)
then    LV_CDN_DIAG_ECRAS_UP_FB = 1
else    LV_CDN_DIAG_ECRAS_UP_FB = 0
endif
if      LV_ECRAS_FB_DIAG = 1  and  LV_EDGE_ECRAS_DOWN = 1
then    LV_CDN_DIAG_ECRAS_DOWN_FB = 1
else    LV_CDN_DIAG_ECRAS_DOWN_FB = 0
endif

//setting LV_DIAG_ECRAS_END if diagnosis mode of ECRAS_UP successful and
setting of LV_DIAG_ECRAS_ERR_END if diagnosis mode was not successful
if      LV_DIAG_ECRAS = 1 and  LV_ECRAS_FB_DIAG = 1
then    if      STATE_ECRAS_FB = 01H
        then    LV_DIAG_ECRAS_END = 1
        else    LV_DIAG_ECRAS_ERR_END = 1
        endif
endif

// diagnosis LV_ERR_ECRAS_UP_FB
if      LV_CDN_DIAG_ECRAS_UP_FB = 1
then    if      LV_DIAG_ECRAS = 0  and (STATE_ECRAS_FB = 02H  or
                STATE_ECRAS_FB = 05H)
        then    ERR_SYM_ECRAS_UP_FB = SYM_0        (debouncing by ERRM)
        elseif  LV_DIAG_ECRAS = 0  and (STATE_ECRAS_FB = 03H  or
                STATE_ECRAS_FB = 06H)
        then    ERR_SYM_ECRAS_UP_FB = SYM_1        (debouncing by ERRM)
        elseif  LV_DIAG_ECRAS = 1  and  T_DIAG_ECRAS >= C_T_DIAG_ECRAS
        then    if      C_ABC_INC_ECRAS_UP_FB > 0
                then    ERR_SYM_ECRAS_UP_FB = SYM_3
                        CTR_ABC_ECRAS_UP_FB= C_ABC_MAX_ECRAS_UP_FB
                        (-> thus LV_ERR_ECRAS_UP_FB will be set by ERRM)
                else    ERR_SYM_ECRAS_UP_FB = SYM_3  (debouncing by ERRM)
                endif
        else    ERR_SYM_ECRAS_UP_FB = NO_SYM     (debouncing by ERRM)
        endif
endif
// LV_ERR_ECRAS_UP_FB and LV_END_DIAG_ECRAS_UP_FB set by ERRM
```

```
// diagnosis LV_ERR_ECRAS_DOWN_FB
if      LV_CDN_DIAG_ECRAS_DOWN_FB = 1
then    if      STATE_ECRAS_FB = 04H   or
                STATE_ECRAS_FB = 05H   or
                STATE_ECRAS_FB = 06H
        then    if      LV_VAR_ECRAS_DOWN = 1
                then    ERR_SYM_ECRAS_DOWN_FB = SYM_0 (debouncing by ERRM)
                else    ERR_SYM_ECRAS_DOWN_FB = NO_SYM (debounc. by ERRM)
                endif
                LV_ECRAS_DOWN_DET_RST = 0
        else    ERR_SYM_ECRAS_DOWN_FB = NO_SYM  (debouncing by ERRM)
                if      LV_VAR_ECRAS_DOWN = 0
                then    LV_ECRAS_DOWN_DET_RST = 1
                else    LV_ECRAS_DOWN_DET_RST = 0
                endif
        endif
else    LV_ECRAS_DOWN_DET_RST = 0
endif
// LV_ERR_ECRAS_DOWN_FB and LV_END_DIAG_ECRAS_DOWN_FB set by ERRM

// reseting of values in case of feedback diagnosis -> calculated at last in this modul !
if      LV_ECRAS_FB_DIAG = 1
then    LV_EDGE_ECRAS_UP       = 0
        LV_EDGE_ECRAS_DOWN  = 0
        LV_DIAG_ECRAS          = 0
        T_DIAG_ECRAS           = 0
endif
ENDIF
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ABC_INC_ECRAS_UP_FB	1	0...FFH	0...255	1	[-]
anti bounce counter increment for ECRAS_UP diagnosis					
C_ABC_MAX_ECRAS_UP_FB	1	1...FFH	1...255	1	[-]
maximal value of anti bounce counter for ECRAS_UP diagnosis					
C_ABC_INC_ECRAS_DOWN_FB	1	0...FFH	0...255	1	[-]
anti bounce counter increment for ECRAS_DOWN diagnosis					
C_ABC_MAX_ECRAS_DOWN_FB	1	1...FFH	1...255	1	[-]
maximal value of anti bounce counter for ECRAS_DOWN diagnosis					
C_T_DIAG_ECRAS	1	0...FFH	0...25.5	0.1	[s]
Maximum waiting time for answer of ECRAS control unit					

26.24.6 ECRAS control unit: diagnosis/detecting of LV_VAR_ECRAS_DOWN

FUNCTION DESCRIPTION:

Description:

If an ECRAS control unit is detected (LV_VAR_ECRAS_UP) a learn algorithm / diagnosis for the variant LV_VAR_ECRAS_DOWN will be start once. LV_VAR_ECRAS_DOWN will be recognised by a pattern of short circiuit to ground and normal phases of the ECRASPWM signal set by the ECRAS control unit. For this mode the ECRAS control unit has to be driven by an ECRASPWM of 90% (learnig mode – see Calculation of ECRASPWM).

Application conditions:

Initialisation: all = 0 at reset **or** LV_IGK 0->1, except LV_ECRAS_DOWN_DET_SET and LV_VAR_ECRAS_DOWN from NVMY
LV_ECRAS_DOWN_DET_SET = 0 and LV_VAR_ECRAS_DOWN = 0 at LC_AD_CLR_VAR 0→1

Recurrence: 100 ms

Activation: LV_CDN_VB_MIN_DIAG = 1 and
LV_IGK = 1

Formula section:

```
IF   LV_VAR_ECRAS_UP = 1
     //setting and resetting of LV_ECRAS_DOWN_DET_SET
     if      T_ECRAS_DOWN_DET >= C_T_ECRAS_DOWN_DET
     then    LV_ECRAS_DOWN_DET_SET = 1
     elseif  LV_ECRAS_DOWN_DET_RST = 1
     then    LV_ECRAS_DOWN_DET_SET = 0
     endif

     // enabling of learning of LV_VAR_ECRAS_DOWN
     if      LV_ECRAS_DOWN_DET_SET = 0          and
             LV_VAR_ECRAS_DOWN = 0
     then    LV_ECRAS_DOWN_DET_ACT = 1
             T_ECRAS_DOWN_DET = T_ECRAS_DOWN_DET + 100 ms
     else    LV_ECRAS_DOWN_DET_ACT = 0
             T_ECRAS_DOWN_DET = 0
             CTR_ECRAS_DOWN_DET = 0
     endif

     if      LV_ECRAS_DOWN_DET_ACT = 1
     then    // detect change SCG <--> normal PWM and set counter if correct pattern
             if      (bit 1 of ERR_DIAG_ECRASPWM  0 -> 1   and
                     C_T_MIN_ECRAS_DOWN_DET < T_ECRAS_DOWN_DET_POS
                                                < C_T_MAX_ECRAS_DOWN_DET)
                     or
                     (bit 1 of ERR_DIAG_ECRASPWM  1 -> 0   and
                     C_T_MIN_ECRAS_DOWN_DET < T_ECRAS_DOWN_DET_NEG
                                               < C_T_MAX_ECRAS_DOWN_DET)
             then    CTR_ECRAS_DOWN_DET = CTR_ECRAS_DOWN_DET + 1
             else    CTR_ECRAS_DOWN_DET unchanged
             endif

             // Variant LV_VAR_ECRAS_DOWN recognised
             if      CTR_ECRAS_DOWN_DET >= C_CTR_ECRAS_DOWN_DET
             then    LV_VAR_ECRAS_DOWN = 1
             else    LV_VAR_ECRAS_DOWN unchanged
```

```
            endif

            // calculate time since start of SCG phase (T_ECRAS_DOWN_DET_NEG)
            if      bit 1 of ERR_DIAG_ECRASPWM  = 1          and
                    [(T_ECRAS_DOWN_DET_NEG = 0               and   //cdn for first entry
                     CTR_ECRAS_DOWN_DET = 0)                 or
                     T_ECRAS_DOWN_DET_NEG > 0  or
                     C_T_MIN_ECRAS_DOWN_DET < T_ECRAS_DOWN_DET_POS
                                             < C_T_MAX_ECRAS_DOWN_DET]
            then    T_ECRAS_DOWN_DET_NEG = T_ECRAS_DOWN_DET_NEG +
                    100 ms
                    T_ECRAS_DOWN_DET_POS = 0 s
            endif

            // calculate time since start of normal PWM (T_ECRAS_DOWN_DET_POS)
            if      bit 1 of ERR_DIAG_ECRASPWM  = 0          and
                    (T_ECRAS_DOWN_DET_POS > 0  or
                    C_T_MIN_ECRAS_DOWN_DET < T_ECRAS_DOWN_DET_NEG
                                             < C_T_MAX_ECRAS_DOWN_DET)
            then    T_ECRAS_DOWN_DET_POS = T_ECRAS_DOWN_DET_POS +
                    100 ms
                    T_ECRAS_DOWN_DET_NEG = 0 s
            endif
        endif
    ELSE
        LV_VAR_ECRAS_DOWN = 0
        LV_ECRAS_DOWN_DET_SET = 0
        T_ECRAS_DOWN_DET_NEG = 0 s
        T_ECRAS_DOWN_DET_POS = 0 s
        CTR_ECRAS_DOWN_DET = 0
        T_ECRAS_DOWN_DET = 0
    ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_ECRAS_DOWN_DET	1	0...FFH	0...25.5	0.1	[s]
Time window for diagnosis / learning of ECRAS_DOWN					
C_CTR_ECRAS_DOWN_DET	1	0...FFH	0...255	1	[-]
Counter threshold for detection of ECRAS_DOWN					
C_T_MIN_ECRAS_DOWN_DET	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback SCG or normal PWM, min value					
C_T_MAX_ECRAS_DOWN_DET	1	0...FFH	0...25.5	0.1	[s]
Time range for identification of the ECRAS feedback SCG or normal PWM, max value					

26.25 Radiator Shutter Electrical Diagnosis (RAS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_RAS	O/V	0...1H	0...1	1	[-]
Boolean for error currently present on Radiator Shutter (= Sound Flap) command signal					
LV_CDN_DIAG_RAS	V	0...1H	0...1	1	[-]
Diagnosis condition RAS power stage diagnosis					
ERR_SYM_RAS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom Radiator Shutter (= Sound Flap) diagnosis					
LV_END_DIAG_RAS	V	0...1H	0...1	1	[-]
End of Radiator Shutter (= Sound Flap) diagnosis					
CDN_DIAG_RAS	O/V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom					
ERR_DIAG_RAS	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for RAS (only parameter)					

Input data:

LV_IGK	LV_VB_CDN_OBD_1	LV_ERR_SPI_MPS	LV_RAS_OUT
LC_RAS_ACT			

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN<CDN_DIAG_XX >, IN<ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>, OUT< LV_ERR_XX>)
ACTION_INFR_GetElDiagRas (OUT< CDN_DIAG_XX>, OUT< ERR_DIAG_XX >)

FUNCTION DESCRIPTION:

General information:

The RAS is driven by the ECU via an output driver. The failure detection is done by ECU Hardware. The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements. The RAS can be switched on or off (no PWM-signal).

Description:

The purpose is to perform the electrical diagnosis of the RAS actuator. 3 symptoms are distinguished:

Error treatment

Diagnostic Instance		Symptoms		ABC type	CARB class
Description	**Name (XX)**	**Description**	**Nr**		
Radiator shutter	RAS	Short circuit to Vbatt SCP	0	MPL_STD_INI	CC
		Short circuit to Ground SCG	1		
		Open Circuit OC	2		
		-	3		

Possible configuration for ABC type and CARB class (see also “Table of Failure”)

ABC Type: STD_INI, STD, MEM, MEM_INI, DEC_CAL, STC, NO, MPL_STD_INI

CARB Class: MIS, FSD, CC, CAT, HC, EVAP, SA, AC, LS, LSH, EGR, OTHER, NO

Application conditions:

Initialisation: ERRM variables are initialized according Filter-type.
At Reset: All 0
Set delay counter for NC_PSD_DLY_RAS

Recurrence: 100ms

Activation: LV_IGK = 1 **and** LC_RAS_ACT= 1

Deactivation: LV_IGK = 0 **or** LC_RAS_ACT= 0

Formula section:

```
If  LV_ERR_SPI_MPS = 0
    and  LV_VB_CDN_OBD_1 = 1
Then

    Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_RAS and
    diagnosis conditions CDN_DIAG_RAS) received from the infrastructure:
    ACTION_INFR_GetElDiagRas (OUT<CDN_DIAG_RAS>, OUT<ERR_DIAG_RAS>)

    Basic diagnosis conditions are set according to infrastructure information:
    CDN_DIAG_RAS
    Failure symptoms (raw value) are set according to infrastructure information:
    ERR_DIAG_ RAS

    If    Activation conditions are met for the NC_PSD_DLY_RAS recurrence
    Then
          { No additional diagnosis conditions are necessary }
    Else
          CDN_DIAG_RAS = 0
    Endif
Else
          CDN_DIAG_RAS = 0
```

Endif

Failure filtering and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_RAS and ERR_DIAG_RAS.

ACTION_ERRM_FilterMulticondition (IN<RAS_OUT>, IN<CDN_DIAG_RAS>,
IN<ERR_DIAG_RAS>, IN<C_ABC_INC_RAS>, IN<C_ABC_MAX_RAS>,
OUT< LV_ERR_RAS>)

This algorithm determines:

ERR_SYM_RAS	(detected error symptom for RAS diagnosis)
LV_ERR_RAS	(Error flag for debounced error of RAS)
LV_CDN_DIAG_RAS	(Diagnosis condition information)
LV_END_DIAG_RAS	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_RAS	1	0...FFH	0...255	1	[-]
Debounce counter increment Radiator Shutter power stage diagnosis					
C_ABC_MAX_RAS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value Radiator Shutter power stage diagnosis					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_RAS	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (= 2)					

26.26 Coolant temperature plausibility diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TCO_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature plausibility error					
ERR_SYM_TCO_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for coolant temperature plausibility error					
LV_CDN_DIAG_TCO_PLAUS	V	0...1H	0...1	1	[-]
Boolean for coolant temperature plausibility diagnosis conditions					
LV_END_DIAG_TCO_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for end of coolant temperature plausibility diagnosis					
LV_END_DIAG_RBM_TCO_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for end of coolant temperature plausibility diagnosis for RBM					
T_MIN_DIAG_TCO_PLAUS	V	0...FFFFH	0...65535	1	[s]
Minimum time to activate the coolant temperature plausibility diagnosis					
T_AST_DIAG_TCO_PLAUS	V	0...FFFFH	0...65535	1	[s]
Time after start of the performed coolant temperature plausibility diagnosis					
T_AST_MAX_DIAG_TCO_PLAUS	V	0...FFFFH	0...65535	1	[s]
Maximum time after start to reach closed-loop enable temperature for coolant temperature plausibility diagnosis					
TCO_DIF_DIAG_TCO_PLAUS	V	80...7FH	-96...95.25	0.75	[°C]
Difference between closed-loop enable- and coolant temperature at start for coolant plausibility diagnosis					

Input data:

LV_PUC	TCO	TCO_ST	TCO_SUB
LV_ST_END	TIA_THR_ST	T_DIAG_AST	RATIO_T_IS_AST
LV_CDN_VB_OBD_2	LV_INH_DIAG_TCO_PLAUS		

FUNCTION DESCRIPTION:

General information:

To monitor the activation of the lambda control after start above a coolant temperature threshold (OBD II requirement), a plausibilization between the coolant temperature increase (TCO) and the calculated coolant temperature increase (TCO_SUB) is made.

The coolant temperature plausibility diagnosis is performed once per engine run.

If the engine stalls and the coolant temperature plausibility diagnosis has not run out, then at the next engine start the timer will be initialized new and the diagnosis starts again. When the diagnosis is finished, the Boolean for end of coolant temperature plausibility diagnosis is set.

Remark: *At the time when the “TCO-plausibility diagnosis” is performed, it is not possible to distinguish between a “TCO-plausibility error” and a “Thermostat error” (which can only be detected later). Therefore a “TCO-plausibility error” is entered, even if the problem may be related to the Thermostat (in order to activate the limp home in case there is really a problem with the coolant temperature sensor).*
That means, if additionally to the “TCO- plausibility error” a “Thermostat diagnosis error” is detected, the “TCO-plausibility error” can be ignored.

Description:

Error-symptoms are defined to this diagnosis function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> -                                        (= SYM_0)
 │  │  └────> -                                        (= SYM_1)
 │  └───────> -                                        (= SYM_2)
 └──────────> Coolant temperature plausibility error   (= SYM_3)
```

Application conditions:

Initialisation at EXIT_ST:

LV_ERR_TCO_PLAUS = 0

LV_CDN_DIAG_TCO_PLAUS = 0

LV_END_DIAG_RBM_TCO_PLAUS = 0

ERR_SYM_TCO_PLAUS = “NO_SYM”

T_MIN_DIAG_TCO_PLAUS = ID_T_MIN_DIAG_TCO_PLAUS

T_AST_DIAG_TCO_PLAUS = 0

TCO_DIF_DIAG_TCO_PLAUS =

ID_TCO_MIN_DIAG_TCO_PLAUS - TCO_ST

T_AST_MAX_DIAG_TCO_PLAUS = ID_T_MAX_DIAG_TCO_PLAUS

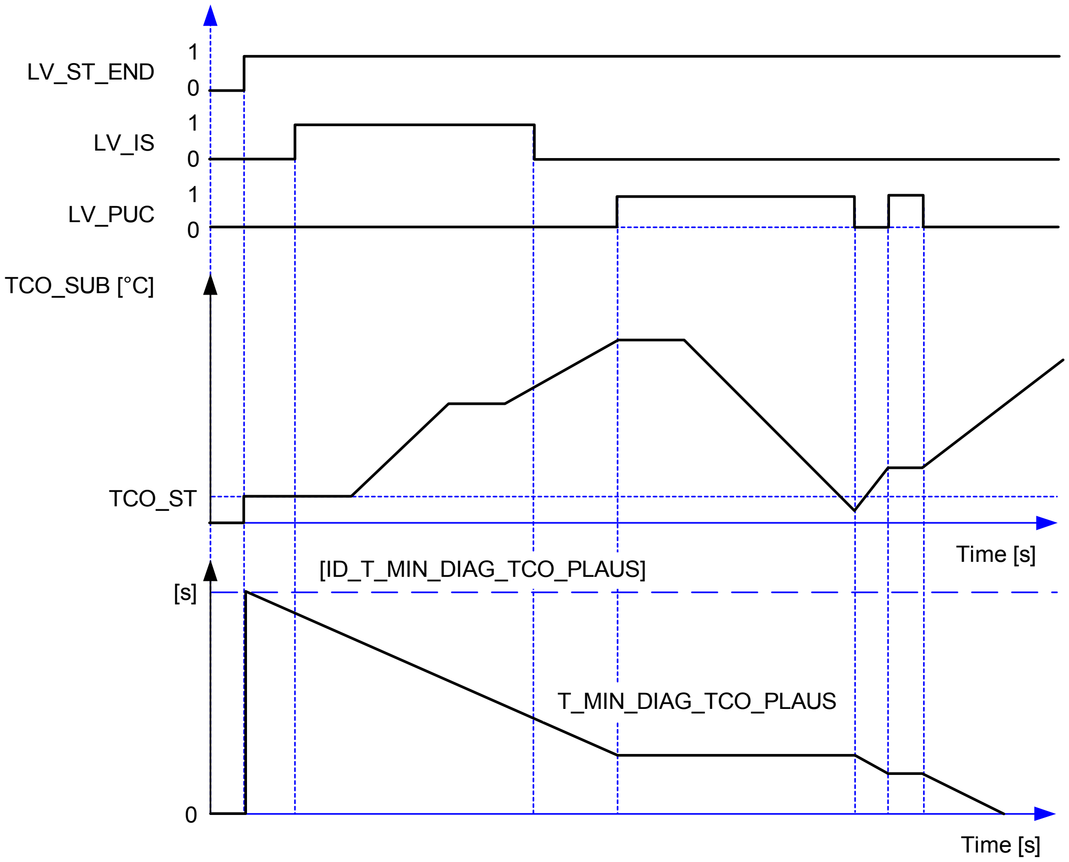
// LV_END_DIAG_TCO_PLAUS is directly initialized by ERRM

Recurrence: 1000 ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Signal flow diagram:



Formula section:

Calculation of plausibility diagnosis activation conditions:

```
If        LV_INH_DIAG_TCO_PLAUS = 0        and
          LV_END_DIAG_TCO_PLAUS = 0        and
          LV_CDN_VB_OBD_2 = 1
then      LV_CDN_DIAG_TCO_PLAUS = 1
else          LV_CDN_DIAG_TCO_PLAUS = 0
endif
```

Calculation of minimum diagnosis time after engine start and end of diagnosis for RBM:

```
If(1)         LV_END_DIAG_RBM_TCO_PLAUS = 0        and
              LV_INH_DIAG_TCO_PLAUS = 0            and
              LV_CDN_VB_OBD_2 = 1
then(1)
    if(2a)    LV_PUC = 1
    then(2a)  T_MIN_DIAG_TCO_PLAUS$_{(n)}$ = T_MIN_DIAG_TCO_PLAUS$_{(n-1)}$
```

```
                T_AST_MAX_DIAG_TCO_PLAUS$_{(n)}$ =
                T_AST_MAX_DIAG_TCO_PLAUS$_{(n-1)}$
    else(2a)    T_MIN_DIAG_TCO_PLAUS$_{(n)}$ = T_MIN_DIAG_TCO_PLAUS$_{(n-1)}$ - 1 s
                T_AST_MAX_DIAG_TCO_PLAUS$_{(n)}$ =
                T_AST_MAX_DIAG_TCO_PLAUS$_{(n-1)}$ -1s
                // counter minimum value limited to zero, no overflow possible
    endif(2a)
    if(2b)      LC_T_DLY_TCO_PLAUS = 1
    then(2b)    // only use of delay time based on absolute temperature at start
        if(2c)      T_MIN_DIAG_TCO_PLAUS = 0                          and
                    ( TCO_SUB > ID_TCO_MIN_DIAG_TCO_PLAUS  or
                    LC_TCO_PLAUS_TCO_SUB_DIAG_INH = 1 )
                    // default is LC_TCO_PLAUS_TCO_SUB_DIAG_INH = 1; this must also be kept
                    for US applications, exept lambda activation by TCO is circumvented
        then(2c)    LV_END_DIAG_RBM_TCO_PLAUS = 1
                    // end of diagnosis for RBM
        endif(2c)
    else(2b)    // only use of delay time based on relative temperature at start
        if(2d)      T_AST_MAX_DIAG_TCO_PLAUS = 0                      and
                    ( TCO_SUB > ID_TCO_MIN_DIAG_TCO_PLAUS  or
                    LC_TCO_PLAUS_TCO_SUB_DIAG_INH = 1 )
                            // default is LC_TCO_PLAUS_TCO_SUB_DIAG_INH = 1; this must also
                    be kept for US applications, exept lambda activation by TCO is circumvented
        then(2d)    LV_END_DIAG_RBM_TCO_PLAUS = 1
                    // end of diagnosis for RBM
        endif(2d)
    endif(2b)
endif(1)
```

Error detection:

```
If(1)           LV_CDN_DIAG_TCO_PLAUS = 1
then(1)
    If(2)       TCO > ID_TCO_MIN_DIAG_TCO_PLAUS
    then(2)     ERR_SYM_TCO_PLAUS = “NO_SYM”
                LV_ERR_TCO_PLAUS = 0                  // without debounce
                LV_END_DIAG_TCO_PLAUS = 1             // end of diagnosis
                T_AST_DIAG_TCO_PLAUS = T_DIAG_AST
```

```
        else(2)
        if(3)      LC_T_DLY_TCO_PLAUS = 1
        then(3)    // only use of delay time based on absolute temperature at start

            if(4a)     T_MIN_DIAG_TCO_PLAUS = 0                                   and
                       ( TCO_SUB > ID_TCO_MIN_DIAG_TCO_PLAUS or
                       LC_TCO_PLAUS_TCO_SUB_DIAG_INH = 1 )
                            // default is LC_TCO_PLAUS_TCO_SUB_DIAG_INH = 1; this must also
                       be kept for US applications, exept lambda activation by TCO is circumvented
            then(4a)   ERR_SYM_TCO_PLAUS = “SYM_3”
                       LV_ERR_TCO_PLAUS = 1                       // without debounce
                       LV_END_DIAG_TCO_PLAUS = 1                  // end of diagnosis
                       T_AST_DIAG_TCO_PLAUS = T_DIAG_AST
            endif(4a)
        else(3)    // only use of delay time based on relative temperature at start
            if(4b)     T_AST_MAX_DIAG_TCO_PLAUS = 0                               and
                       ( TCO_SUB > ID_TCO_MIN_DIAG_TCO_PLAUS or
                       LC_TCO_PLAUS_TCO_SUB_DIAG_INH = 1 )
                            // default is LC_TCO_PLAUS_TCO_SUB_DIAG_INH = 1; this must also
                       be kept for US applications, exept lambda activation by TCO is circumvented
            then(4b)   ERR_SYM_TCO_PLAUS = “SYM_3”
                       LV_ERR_TCO_PLAUS = 1                       // without debounce
                       LV_END_DIAG_TCO_PLAUS = 1                  // end of diagnosis
                       T_AST_DIAG_TCO_PLAUS = T_DIAG_AST
            endif(4b)
        endif(3)
        endif(2)
    endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_T_MIN_DIAG_TCO_PLAUS	6	1...FFFFH	1...65535	1	[s]
LDP_TIA_THR_ST_ID_T_TCO_PLAUS	6	0...FEH	-48...142.5	0.75	[°C]
Minimum time for coolant temperature plausibility diagnosis activation					
ID_TCO_MIN_DIAG_TCO_PLAUS	6*6	0...FEH	-48...142.5	0.75	[°C]
LDP_TCO_ST_ID_TCO_MIN_DIAG_TCO	6	0...FEH	-48...142.5	0.75	[°C]
LDP_RATIO_T_IS_AST_ID_TCO_DIAG	6	0...FFH	0...99.60937	0.390625	[%]
Minimum coolant temperature threshold for coolant temperature plausibility diagnosis					
ID_T_MAX_DIAG_TCO_PLAUS	4	1...FFFFH	1...65535	1	[s]
LDP_TCO_DIF_DIAG_TCO_PLAUS_ID_T	4	0...FFH	-96...95.25	0.75	[°C]
Time interval to reach closed-loop enable temperature for coolant temperature plausibility diagnosis					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TCO_PLAUS_TCO_SUB_DIAG_INH	-	0...1H	0...1	1	[-]
Logical calibration data to inhibit the use of TCO_SUB condition for error detection					
LC_T_DLY_TCO_PLAUS	-	0...1H	0...1	1	[-]
Logical calibration data to switch between delay timer based on absolute or relative temperature at start					

26.27 Coolant temperature sensor plausibility diagnosis (Appl. Inc.)

26.27.1 Diagnosis inhibition

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_TCO_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature plausibility diagnosis inhibit					
LV_INH_DIAG_RBM_TCO_PLAUS	V	0...1H	0...1	1	[-]
Boolean for coolant temperature plausibility diagnosis inhibit due to OBD error					

Input data:

LV_ES	LV_IGK	LV_CDN_DIAG_TCO_PLAUS	LV_ERR_TCO_EL
LV_ERR_TCO_GRD	LV_ERR_TCO_STUCK	LV_ERR_TIA_EL	LV_ERR_TIA_PLAUS
LV_ERR_TCO_STUCK_RNG			

FUNCTION DESCRIPTION:

General information:

The coolant temperature sensor plausibility diagnosis is performed at transition from the logical value for “Start” from “1” to “0”. If the engine stalls or one of the conditions for inhibiting the diagnosis is present, it will be stopped and starts again at the next engine start.

The diagnosis is finished as soon as the boolean for end of coolant temperature sensor plausibility diagnosis is set to “1”. In this case the boolean for coolant temperature plausibility error indicates either a not plausible (= 1) or a normal (= 0) coolant temperature sensor signal.

Depending on project specific requirements, the coolant temperature sensor plausibility diagnosis can be inhibited by setting of LV_INH_DIAG_TCO_PLAUS.

Application conditions:

Initialisation:

```
If      Reset or Transition: LV_IGK = 0 -> 1
then    LV_INH_DIAG_TCO_PLAUS= 0
endif
```

Recurrence: 1000 ms

Activation:

```
If          LV_CDN_DIAG_TCO_PLAUS = 1
then    (Calculation of the TCO plausi. diagnosis interface parameter enabled)
endif
```

Deactivation:

```
If      LV_CDN_DIAG_TCO_PLAUS = 0
then    (Calculation of the TCO plausi. diagnosis interface parameter disabled)
endif
```

Formula section:

```
If      LV_ERR_TCO_EL = 0                  and
        LV_ERR_TCO_GRD = 0                 and
        LV_ERR_TCO_STUCK = 0               and
        LV_ERR_TCO_STUCK_RNG = 0           and
        LV_ERR_TIA_EL = 0                  and
        LV_ERR_TIA_PLAUS = 0
then    LV_INH_DIAG_RBM_TCO_PLAUS = 0
else    LV_INH_DIAG_RBM_TCO_PLAUS = 1
endif

If      LV_ES = 0                          and
        LV_IGK = 1                         and
        LV_INH_DIAG_RBM_TCO_PLAUS = 0
then    LV_INH_DIAG_TCO_PLAUS = 0
else    LV_INH_DIAG_TCO_PLAUS = 1
endif
```

Error treatment:

Error debounce:

no debounce

26.27.2 Interface for Rate – Based - Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_TCO_PLAUS	V/O	0...7H	0...7	1	[-]
Interface of TCO_PLAUS monitor with the Rate-Based Monitoring statistics					

Input data:

LV_DC	CTR_ERR_DYN_NR	LV_END_DIAG_RBM_TCO_PLAUS	LV_CDN_CST_RBM

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the TCO_PLAUS monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TCO_PLAUS data.

Within STATE_RBM_TCO_PLAUS, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_TCO_PLAUS = 0

on failure memory reset :

bit 1 of STATE_RBM_TCO_PLAUS = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_TCO_EL	LV_ERR_TCO_STUCK	LV_ERR_TCO_GRD	LV_ERR_TIA_EL
LV_ERR_TIA_PLAUS	LV_ERR_TCO_STUCK_RNG		

```
If(1)       { CPU optimization at LV_DC 0 → 1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While  bit 1 of STATE_RBM_TCO_PLAUS = 0 do
                   with each XX failure of the above list :
                   ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                   SYNCHRONIZATION<CALL>)
                   If(2)          XX has a pending status
                   Then(2)
                                  bit 1 of STATE_RBM_TCO_PLAUS = 1
                   Endif(2)
            Endwhile
Else(1)
            { the dynamic failure memory is empty }
            No action
Endif(1)
```

Every 1 s :

```
If    bit 0 of STATE_RBM_TCO_PLAUS = 0
Then
            If     LV_END_DIAG_RBM_TCO_PLAUS = 1
            Then   bit 0 of STATE_RBM_TCO_PLAUS = 1
            Endif
Endif


If    bit 1 of STATE_RBM_TCO_PLAUS = 0
Then
            If     LV_INH_DIAG_RBM_TCO_PLAUS = 1
            Then   bit 1 of STATE_RBM_TCO_PLAUS = 1
            Endif
Endif


If    bit 2 of STATE_RBM_TCO_PLAUS = 0
Then
```

```
        If      LV_CDN_CST_RBM = 1
        Then    bit 2 of STATE_RBM_TCO_PLAUS = 1
        Endif
Endif
```

26.28 Coolant temperature low sided rationality check

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TCO_STUCK	V/O	0...1H	0...1	1	[-]
TCO Stuck signal detected					
ERR_SYM_TCO_STUCK	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for coolant temperature stuck signal error					
LV_CDN_DIAG_TCO_STUCK	V	0...1H	0...1	1	[-]
Boolean for coolant temperature stuck signal diagnosis conditions					
LV_END_DIAG_TCO_STUCK	V/O	0...1H	0...1	1	[-]
Boolean for end of coolant temperature stuck signal diagnosis					
TCO_MES_DIF_DIAG_TCO_STUCK	V	0...FEH	0...190.5	0.75	[°C]
Difference of measured coolant temperature values for coolant temperature stuck signal diagnosis					
TCO_MES_MIN_DIAG_TCO_STUCK	V	0...FEH	-48...142.5	0.75	[°C]
Measured coolant temperature minimum value for coolant temperature stuck signal diagnosis					
TCO_MES_MAX_DIAG_TCO_STUCK	V	0...FEH	-48...142.5	0.75	[°C]
Measured coolant temperature maximum value for coolant temperature stuck signal diagnosis					
TCO_SUB_DIF_DIAG_TCO_STUCK	V	0...FEH	0...190.5	0.75	[°C]
Difference of substitute coolant temperature values for coolant temperature stuck signal diagnosis					
TCO_SUB_MIN_DIAG_TCO_STUCK	V	0...FEH	-48...142.5	0.75	[°C]
Substitute coolant temperature minimum value for coolant temperature stuck signal diagnosis					
TCO_SUB_MAX_DIAG_TCO_STUCK	V	0...FEH	-48...142.5	0.75	[°C]
Substitute coolant temperature maximum value for coolant temperature stuck signal diagnosis					

Input data:

TCO_MES	TCO_ST	TCO_SUB	LV_ST_END
LV_CDN_VB_MIN_DIAG	LV_INH_DIAG_TCO_STUCK		

FUNCTION DESCRIPTION:

General information:

The purpose of the signal stuck diagnosis is to detect a stuck measured coolant temperature sensor signal. The diagnosis function checks if after a variation of the calculated coolant temperature substitute value (TCO_SUB) also a variation of the measured coolant temperature value (TCO_MES) is detected.

The diagnosis is performed only once per engine run. The symptom of the error code is not handled by anti-bouncing.

Description:

Error-symptoms are defined to this diagnosis function as following:

```
☐☐☐☐
 | | | └─> -                                          (= SYM_0)
 | | └───> -                                          (= SYM_1)
 | └─────> -                                          (= SYM_2)
 └───────> Coolant temperature signal stuck error     (= SYM_3)
```

Application conditions:

Initialisation at Exit start (EXIT_ST):

LV_ERR_TCO_STUCK = 0
LV_CDN_DIAG_TCO_STUCK = 0
LV_END_DIAG_TCO_STUCK = 0
ERR_SYM_TCO_STUCK = “NO_SYM”
TCO_MES_DIF_DIAG_TCO_STUCK = 0°C
TCO_SUB_DIF_DIAG_TCO_STUCK = 0°C
TCO_MES_MIN_DIAG_TCO_STUCK = 142.5°C
TCO_MES_MAX_DIAG_TCO_STUCK = -48°C
TCO_SUB_MIN_DIAG_TCO_STUCK = TCO_SUB
TCO_SUB_MAX_DIAG_TCO_STUCK = TCO_SUB

Recurrence: 1000 ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

```
If      LV_CDN_VB_MIN_DIAG = 1          and
        LV_INH_DIAG_TCO_STUCK = 0       and
        LV_END_DIAG_TCO_STUCK = 0
then    LV_CDN_DIAG_TCO_STUCK = 1
else    LV_CDN_DIAG_TCO_STUCK = 0
endif

IF(1)       LV_CDN_DIAG_TCO_STUCK = 1
THEN(1)
Calculation of the coolant temperature-boundary values:
If(2)       TCO_MES  <  TCO_MES_MIN_DIAG_TCO_STUCK
then(2)     TCO_MES_MIN_DIAG_TCO_STUCK = TCO_MES
else(2)     TCO_MES_MIN_DIAG_TCO_STUCK = TCO_MES_MIN_DIAG_TCO_STUCK
endif(2)
If(2)       TCO_MES  >  TCO_MES_MAX_DIAG_TCO_STUCK
then(2)     TCO_MES_MAX_DIAG_TCO_STUCK = TCO_MES
else(2)     TCO_MES_MAX_DIAG_TCO_STUCK = TCO_MES_MAX_DIAG_TCO_STUCK
endif(2)
If(2)       TCO_SUB  <  TCO_SUB_MIN_DIAG_TCO_STUCK
then(2)     TCO_SUB_MIN_DIAG_TCO_STUCK = TCO_SUB
```

```
else(2)     TCO_SUB_MIN_DIAG_TCO_STUCK = TCO_SUB_MIN_DIAG_TCO_STUCK
endif(2)
If(2)       TCO_SUB  >  TCO_SUB_MAX_DIAG_TCO_STUCK
then(2)     TCO_SUB_MAX_DIAG_TCO_STUCK = TCO_SUB
else(2)     TCO_SUB_MAX_DIAG_TCO_STUCK = TCO_SUB_MAX_DIAG_TCO_STUCK
endif(2)
```

Calculation of the coolant temperature-difference values:

```
TCO_MES_DIF_DIAG_TCO_STUCK =
    | TCO_MES_MAX_DIAG_TCO_STUCK – TCO_MES_MIN_DIAG_TCO_STUCK |
TCO_SUB_DIF_DIAG_TCO_STUCK =
    | TCO_SUB_MAX_DIAG_TCO_STUCK – TCO_SUB_MIN_DIAG_TCO_STUCK |
```

Error detection:

```
If (2)          TCO_SUB_DIF_DIAG_TCO_STUCK  >
                        IP_TCO_SUB_DIF_DIAG_TCO_STUCK
then (2)
    if (3)      TCO_MES_DIF_DIAG_TCO_STUCK  <
                        IP_TCO_MES_DIF_DIAG_TCO_STUCK
    then (3)    LV_ERR_TCO_STUCK = 1                    (without debounce)
                ERR_SYM_TCO_STUCK = “SYM_3”
                LV_END_DIAG_TCO_STUCK = 1
    else (3)    LV_ERR_TCO_STUCK = 0                    (without debounce)
                ERR_SYM_TCO_STUCK = “NO_SYM”
                LV_END_DIAG_TCO_STUCK = 1
    endif (3)
endif (2)
ENDIF(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TCO_SUB_DIF_DIAG_TCO_STUCK	6	0...FEH	0...190.5	0.75	[°C]
LDPM_TCO_ST_1_ENTE	6	0...FEH	-48...142.5	0.75	[°C]
Minimum increase of the TCO_SUB value for the coolant temperature stuck signal diagnosis					
IP_TCO_MES_DIF_DIAG_TCO_STUCK	6	0...FEH	0...190.5	0.75	[°C]
LDPM_TCO_ST_1_ENTE	6	0...FEH	-48...142.5	0.75	[°C]
Minimum increase of the TCO value for the coolant temperature stuck signal diagnosis					

26.29 TCO sensor stuck in range plausibility diagnosis with engine off time

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TCO_STUCK_RNG	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature signal stuck in range detected					
ERR_SYM_TCO_STUCK_RNG	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for coolant temperature signal stuck in range error					
LV_CDN_DIAG_TCO_STUCK_RNG	V	0...1H	0...1	1	[-]
Boolean for coolant temperature signal stuck in range diagnosis conditions					
LV_END_DIAG_TCO_STUCK_RNG	V/O	0...1H	0...1	1	[-]
Boolean for end of coolant temperature signal stuck in range diagnosis					
TCO_ST_STUCK_RNG	V	0...FEH	-48...142.5	0.75	[°C]
Measured coolant temperature at start					

Input data:

TCO_MES	TAM_ST	TIA_THR_ST
LV_IGK	LV_CDN_VB_OBD_2	LV_INH_DIAG_TCO_STUCK_RNG
T_ES_CUS	T_AST	LV_ST
LV_END_DIAG_TCO_EL	LV_END_DIAG_TCO_GRD	

FUNCTION DESCRIPTION:

General information:

The purpose of the signal stuck in range diagnosis is to detect a measured coolant temperature signal, which is stuck in range above the lowest maximum enable temperature. The diagnosis function checks, if the cooling behavior of the coolant temperature is plausible in comparison to the engine off time.

The diagnosis is performed only once per engine run.The symptom of the error code is not handled by an antibounce counter.

Description:

Error-symptoms are defined to this diagnosis function as following:

```
□□□□
 │ │ │ └─> -                                              (= SYM_0)
 │ │ └───> -                                              (= SYM_1)
 │ └─────> -                                              (= SYM_2)
 └───────> Coolant temperature signal stuck in range error  (= SYM_3)
```

Application conditions:

```
Initialisation:     at Exit start (EXIT_ST)
                    LV_ERR_TCO_STUCK_RNG = 0
                    ERR_SYM_TCO_STUCK_RNG = "NO_SYM"
                    LV_CDN_DIAG_TCO_STUCK_RNG = 0

                    at reset:
                    TCO_ST_STUCK_RNG = 0

                    // LV_END_DIAG_TCO_STUCK_RNG is directly initialized by ERRM

Recurrence:         100ms

Activation:         LV_IGK = 1

Deactivation:       LV_IGK = 0                                or
                    LV_END_DIAG_TCO_STUCK_RNG = 1
```

Formula section:

```
If          LV_ST = 1
Then        TCO_ST_STUCK_RNG = TCO_MES
Endif

If          T_AST > C_T_AST_MIN_TCO_STUCK_RNG  //delay-time to get a valid T_ES_CUS
Then
    If              LV_CDN_VB_OBD_2 = 1
            and     TIA_THR_ST < C_TIA_THR_ST_MAX_TCO_STUCK_RNG
            and     TIA_THR_ST > C_TIA_THR_ST_MIN_TCO_STUCK_RNG
            and     LV_INH_DIAG_TCO_STUCK_RNG = 0
            and     TAM_ST > C_TAM_ST_MIN_TCO_STUCK_RNG
            and     |TIA_THR_ST - TAM_ST| < C_TIA_TAM_DIF_MAX_TCO_STUCK_RNG
            and     T_ES_CUS > C_T_ES_MIN_TCO_STUCK_RNG
            and     LV_END_DIAG_TCO_EL = 1
            and     LV_END_DIAG_TCO_GRD = 1
    Then            LV_CDN_DIAG_TCO_STUCK_RNG = 1
    Else            LV_CDN_DIAG_TCO_STUCK_RNG = 0
```

```
Endif


If(1)       LV_CDN_DIAG_TCO_STUCK_RNG = 1
then(1)


    if(2)   (TCO_ST_STUCK_RNG - TIA_THR_ST) > IP_TCO_ST_DIF_TIA_THR_ST
                (path 1 - comparison of intake air and coolant temperature value)
            and

            TCO_ST_STUCK_RNG > IP_TCO_ST_TCO_STUCK_RNG
                (path 2 - comparison of engine off time to coolant temperature value)

    then(2)     LV_ERR_TCO_STUCK_RNG = 1
                ERR_SYM_TCO_STUCK_RNG = "SYM_3"
                LV_END_DIAG_TCO_STUCK_RNG = 1
    else(2)     LV_ERR_TCO_STUCK_RNG = 0
                ERR_SYM_TCO_STUCK_RNG = "NO_SYM"
                LV_END_DIAG_TCO_STUCK_RNG = 1
    endif(2)


endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TCO_ST_TCO_STUCK_RNG	4*6	0...FEH	-48...142.5	0.75	[°C]
LDPM_T_ES_CUS_TCO_STUCK_RNG	4	0...FFFFH	0...65535	1	[min]
LDP_TIA_THR_ST_IP_TCO_ST	6	0...FEH	-48...142.5	0.75	[°C]
Interpolation table for expected coolant temperature value depending on TIA_THR_ST and T_ES_CUS					
IP_TCO_ST_DIF_TIA_THR_ST	4	0...FEH	0...190.5	0.75	[°C]
LDPM_T_ES_CUS_TCO_STUCK_RNG	4	0...FFFFH	0...65535	1	[min]
Interpolation table for difference between coolant temperature and intake air temperature at start					
C_T_ES_MIN_TCO_STUCK_RNG	1	0...FFFFH	0...65535	1	[min]
Constant for minimum required engine off time for diagnosis					
C_TIA_THR_ST_MAX_TCO_STUCK_RNG	1	0...FEH	-48...142.5	0.75	[°C]
Constant for maximum intake air temperature at throttle at start					
C_TIA_THR_ST_MIN_TCO_STUCK_RNG	1	0...FEH	-48...142.5	0.75	[°C]
Constant for minimum intake air temperature at throttle at start					
C_TIA_TAM_DIF_MAX_TCO_STUCK_RNG	1	0...FEH	0...190.5	0.75	[°C]
Constant for difference between intake air temperature and ambient temperature at start					
C_TAM_ST_MIN_TCO_STUCK_RNG	1	0...FEH	-48...142.5	0.75	[°C]
Constant for minimum ambient temperature at start					
C_T_AST_MIN_TCO_STUCK_RNG	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum time after start to enable diagnosis					

26.30 TCO sensor stuck in range plausibility diagnosis with engine off time (Appl. Inc.)

26.30.1 Inhibition of diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_TCO_STUCK_RNG	V/O	0...1H	0...1	1	[-]
Boolean for inhibition of coolant temperature signal stuck in range diagnosis					

Input data:

LV_ERR_TCO_EL	LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS	LV_ERR_TIA_PLAUS
LV_ERR_TCO_STUCK	LV_ERR_TIA_EL	LV_ERR_TAM_PLAUS	LV_ERR_TAM_CAN
LV_ERR_BN_T_ICL	LV_ERR_CAN_BOFF	LV_IGK	LV_END_DIAG_TCO_STUCK_RNG
LV_ERR_T_ES	LV_ERR_ECU_NVMY	LV_ERR_TOUT_ICL_3	LV_ST_END

FUNCTION DESCRIPTION:

FUNCTION DESCRIPTION:

General information:

Depending on project specific requirements, the coolant temperature sensor stuck in range diagnosis can be inhibited by setting of the logical variable LV_INH_DIAG_TCO_STUCK_RNG.

Application conditions:

Initialisation at RESET **or** EXIT_ST **or** LV_IGK = 0 -> 1 **or** FMY clear:

LV_INH_DIAG_TCO_STUCK_RNG = 0

Recurrence: 100 ms

Activation: LV_IGK = 1

(Calculation of the TCO_STUCK_RNG diag. interface parameter enabled)

Deactivation: LV_IGK = 0 **or**

LV_END_DIAG_TCO_STUCK_RNG = 1

(Calculation of the TCO_STUCK_RNG diag. interface parameter disabled)

Formula section:

If LV_ERR_TCO_EL = 1 **or**

LV_ERR_TCO_GRD = 1 **or**

LV_ERR_TIA_EL = 1 **or**

```
          LV_ERR_TIA_PLAUS = 1                  or
          LV_ERR_TAM_CAN = 1                    or
          LV_ERR_TAM_PLAUS = 1                  or
          LV_ERR_BN_T_ICL = 1                   or
          LV_ERR_TOUT_ICL_3 = 1                 or
          LV_ERR_CAN_BOFF = 1                   or
          LV_ERR_T_ES = 1                       or
          LV_ERR_ECU_NVMY = 1                   or
          (( LV_ST_END = 1                      and
          (LV_ERR_TCO_PLAUS = 1                 or
          LV_ERR_TCO_STUCK = 1))                or
Then      LV_INH_DIAG_TCO_STUCK_RNG = 1
Endif
```

Configuration for diagnostic symptoms:

Diagnostic **TCO_STUCK_RNG**	**Symptom description**	**Symptom**	**Filter type**
TCO sensor stuck in range diagnosis			*NO*
	TCO sensor stuck in range error	*SYM_3*	

26.30.2 Interface to rate-based monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_TCO_STUCK_RNG	V/O	0...FFH	0...255	1	[-]
Interface of TCO_STUCK_RNG monitor with the Rate-Based Monitoring statistics					

Input data:

LV_DC	CTR_ERR_DYN_NR	LV_END_DIAG_TCO_STUCK_RNG	LV_CDN_CST_RBM

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the TCO_STUCK_RNG monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TCO_STUCK_RNG data.

Within STATE_RBM_TCO_STUCK_RNG, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

Application conditions:

Initialisation:

at ECU reset:

bit 0, bit 1 and bit 2 of STATE_RBM_TCO_STUCK_RNG = 0

at LV_DC 0 → 1 transition:

bit 0, bit 1 and bit 1 of STATE_RBM_TCO_STUCK_RNG = 0

on failure memory reset:

bit 1 of STATE_RBM_TCO_STUCK_RNG = 0

Recurrence: 100ms

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once:

LV_ERR_TCO_EL	LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS	LV_ERR_TIA_PLAUS
LV_ERR_TCO_STUCK	LV_ERR_TIA_EL	LV_ERR_TAM_PLAUS	LV_ERR_TAM_CAN
LV_ERR_BN_T_ICL	LV_ERR_CAN_BOFF	LV_ERR_ECU_NVMY	LV_ERR_T_ES
LV_ERR_TOUT_ICL_3			

```
If(1)     { CPU optimization at LV_DC 0 → 1 transition }
          CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
          While  bit 1 of STATE_RBM_TCO_STUCK_RNG = 0 do
                 with each XX failure of the above list :
                 ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                 SYNCHRONIZATION<CALL>)
                 If(2)        XX has a pending status
                 Then(2)
                              bit 1 of STATE_RBM_TCO_STUCK_RNG = 1
                 Endif(2)
          Endwhile
Else(1)
          { the dynamic failure memory is empty }
          No action
Endif(1)
```

Every 100ms:

```
If    bit 0 of STATE_RBM_TCO_STUCK_RNG = 0
Then
          If      LV_END_DIAG_TCO_STUCK_RNG = 1
          Then    bit 0 of STATE_RBM_TCO_STUCK_RNG = 1
          Endif
Endif
If    bit 1 of STATE_RBM_TCO_STUCK_RNG = 0
Then
          If      LV_INH_DIAG_TCO_STUCK_RNG = 1
          Then    bit 1 of STATE_RBM_TCO_STUCK_RNG = 1
          Endif
Endif
If    bit 2 of STATE_RBM_TCO_STUCK_RNG = 0
```

```
Then
        If      LV_CDN_CST_RBM = 1
        Then    bit 2 of STATE_RBM_TCO_STUCK_RNG = 1
        Endif
Endif
```

26.31 TCO sensor (radiator outlet) plausibility diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TCO_2_PLAUS	V/O	0...1H	0...1	1	-
Boolean for coolant temperature (radiator outlet) plausibility error					
LV_CDN_DIAG_TCO_2_PLAUS	V	0...1H	0...1	1	-
Boolean for coolant temperature (radiator outlet) plausibility diagnosis conditions					
ERR_SYM_TCO_2_PLAUS	V	0H 1H 8H	NO_SYM SYM_0 SYM_3	1	-
Symptom for coolant temperature (radiator outlet) plausibility error					
LV_END_DIAG_TCO_2_PLAUS	V	0...1H	0...1	1	-
Boolean for end of coolant temperature (radiator outlet) plausibility diagnosis					
LV_END_DIAG_RBM_TCO_2_PLAUS	V/O	0...1H	0...1	1	-
Boolean for end of coolant temperature (radiator outlet) plausibility diagnosis for RBM					
LV_TCO_2_PLAUS_ENA	V	0…1H	0…1	1	-
Coolant temperature (radiator outlet) plausibility (long) enabled					
TCO_2_DIF_TCO_2_PLAUS	V	0…FEH	0…190.5	0.75	°C
Maximum measured coolant temperature (radiator outlet) gap in diagnosis window					
CTR_DLY_TCO_2_PLAUS	V	0…FFFFH	0...65535	1	-
Delay counter before coolant temperature (radiator outlet) change is expected					
T_DLY_TCO_2_PLAUS	V	0…FFFFH	0...65535	1	s
Coolant temperature (radiator outlet) plausibility diagnosis diagnostic window time					
TCO_2_PLAUS_H	V	0…FEH	-48…142.5	0.75	°C
Maximum latched measured coolant temperature (radiator outlet) value					
TCO_2_PLAUS_L	V	0…FEH	-48…142.5	0.75	°C
Minimum latched measured coolant temperature (radiator outlet) value					

Input data:

TCO	TCO_ST	TCO_2_MES	TCO_2_MES_ST
N	LV_INH_DIAG_TCO_2_PLAUS	TIA_THR_ST	LV_PL
LV_ST_END	LV_TCO_2_PLAUS_ENA_EXT	LV_CDN_VB_OBD2	VS

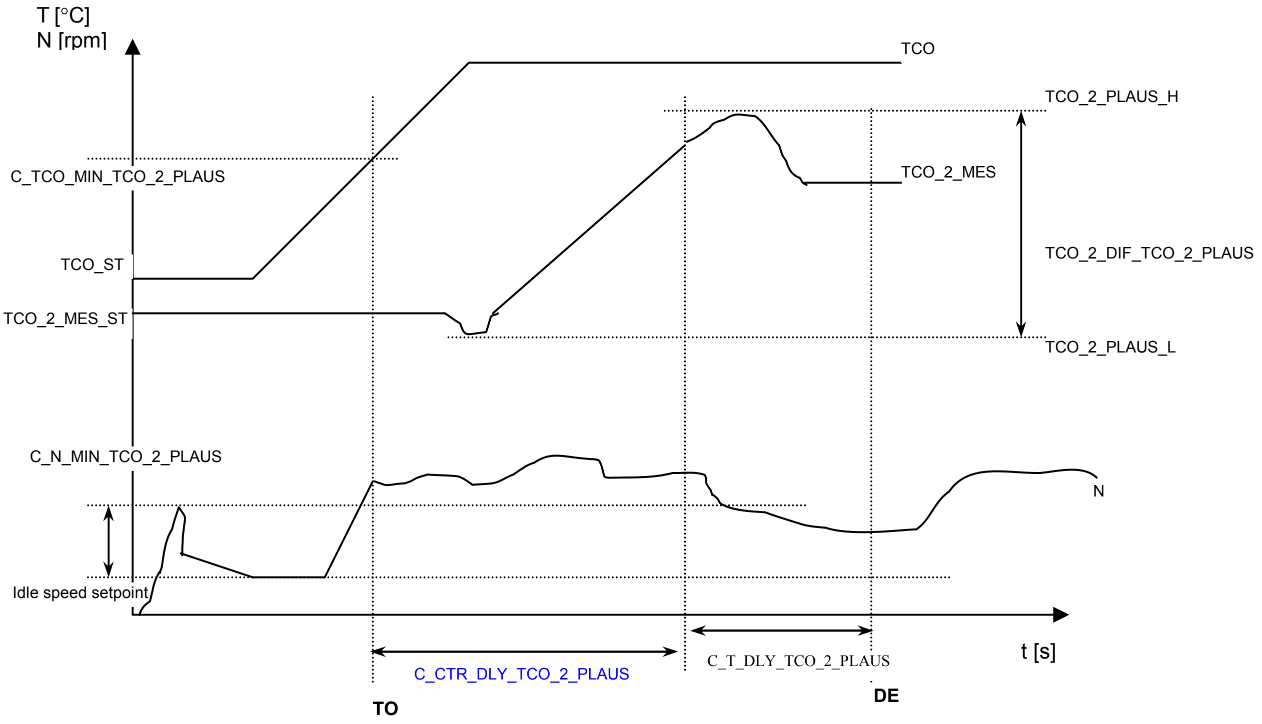
FUNCTION DESCRIPTION:

General information:

The diagnosis is based on monitoring the alteration of TCO_2_MES (positive and negative change) at:

- During warm – up phase (long term check, no error detection)
- After opening of thermostat (long term check with error detection)
- Just after start (short test with error detection)

26.31.1.1 Signal flow diagram:



Application conditions:

Initialisation at EXIT_ST:

LV_ERR_TCO_2_PLAUS = 0
LV_CDN_DIAG_TCO_2_PLAUS = 0
LV_END_DIAG_TCO_2_PLAUS = 0
ERR_SYM_TCO_2_PLAUS = “NO_SYM”
LV_END_DIAG_RBM_TCO_2_PLAUS = 0
LV_TCO_2_PLAUS_ENA = 0
TCO_2_DIF_TCO_2_PLAUS = 0 °C
TCO_2_PLAUS_H = -48°C
TCO_2_PLAUS_L = 142,5°C
CTR_DLY_TCO_2_PLAUS = C_CTR_DLY_TCO_2_PLAUS
T_DLY_TCO_2_PLAUS = C_T_DLY_TCO_2_PLAUS

Recurrence: 1000 ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Description:

- During warm – up phase (long term check, no error detection)

The TCO_2 rationality diagnostic function will only run if TCO_ST and TIA_THR_ST are within a tunable range after engine start. TCO_2_PLAUS_H and TCO_2_PLAUS_L are continuously updated from engine start, after first being initialized to TCO_2_ST at power up. The difference between these two values, TCO_2_DIF_TCO_2_PLAUS, is also calculated. When this value exceeds a minimum threshold C_TCO_2_DIF_MIN_TCO_2_PLAUS then the diagnosis is finished with a positive rationality check result.

- After opening of thermostat (long term check with error detection)

The error detection is based on a change of TCO_2_MES after the opening temperature of the thermostat is reached and following the vehicle/engine is driven under certain conditions (vehicle speed, part load, engine speed, opening of thermostat) for the delay time CTR_DLY_TCO_2_PLAUS. However, should one condition drop below the threshold or the engine exit part load state during this tunable period, then the timer is reinitialized/incremented.

Additional, at CTR_DLY_TCO_2_PLAUS = 0 a second delay timer is started (T_DLY_TCO_2_PLAUS) to wait for the change of TCO_2_MES. If the delay time is over and there’s still no change in TCO_2_MES then the error is set.

- Just after start (short test with error detection)

Apart from the long term checks, an extra rationality check is carried out at engine start; if TCO_ST is smaller than a threshold and TCO_2_MES_ST – TCO_ST is greater than a threshold then a fault is also detected.

Error-symptoms are defined to this diagnosis function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> TCO_2 to high after ST    (= SYM_0)
 │  │  └────> -                         (= SYM_1)
 │  └───────> -                         (= SYM_2)
 └──────────> TCO_2 not plausible       (= SYM_3)
```

Formula section:

Calculation of diagnosis condition

```
IF      LV_INH_DIAG_TCO_2_PLAUS = 0                      AND
        LV_END_DIAG_TCO_2_PLAUS = 0                      AND
        LV_CDN_VB_OBD2 = 1
THEN    LV_CDN_DIAG_TCO_2_PLAUS = 1
ELSE    LV_CDN_DIAG_TCO_2_PLAUS = 0
ENDIF
```

Calculation of Diagnosis warm-up phase:

```
IF(1)       TCO_ST > C_TCO_ST_MIN_TCO_2_PLAUS                    AND
            TCO_ST < C_TCO_ST_MAX_TCO_2_PLAUS                    AND
            TIA_THR_ST > C_TIA_THR_ST_MIN_TCO_2_PLAUS            AND
            TIA_THR_ST < C_TIA_THR_ST_MAX_TCO_2_PLAUS            AND
            LV_END_DIAG_RBM_TCO_2_PLAUS = 0
THEN(1)     (calculation of formula section A, B, C and D – "SYM_3" error check)
```

A Calculation of min. and max. TCO_2_MES and delta TCO_2_MES

```
    IF(2a)        TCO_2_MES > TCO_2_PLAUS_H
    THEN(2a)      TCO_2_PLAUS_H = TCO_2_MES
    ELSE(2a)
          IF(3a)        TCO_2_MES < TCO_2_PLAUS_L
          THEN(3a)      TCO_2_PLAUS_L = TCO_2_MES
          ENDIF(3a)
    ENDIF(2a)
    TCO_2_DIF_TCO_2_PLAUS = TCO_2_PLAUS_H - TCO_2_PLAUS_L
```

B Calculation of TCO_2_DIF_TCO_2_PLAUS to end the diagnosis

```
    IF(2b)        LV_CDN_DIAG_TCO_2_PLAUS = 1                      AND
                  TCO_2_DIF_TCO_2_PLAUS >= C_TCO_2_DIF_MIN_TCO_2_PLAUS
    THEN(2b)      LV_END_DIAG_TCO_2_PLAUS = 1          // end of diagnosis
    ENDIF(2b)
```

C Calculation of delay counter the reach conditions

```
    IF(2c)        [ TCO > C_TCO_MIN_TCO_2_PLAUS       OR
                  LV_TCO_2_PLAUS_ENA_EXT = 1 ]                     AND
                  LV_TCO_2_PLAUS_ENA = 0
    THEN(2c)      //thermostat opening temperature has been reached
          IF(3c)        N > C_N_MIN_TCO_2_PLAUS                    AND
                        LV_PL = 1                                  AND
                        VS >= C_VS_MIN_TCO_2_PLAUS
          THEN(3c)      CTR_DLY_TCO_2_PLAUS = CTR_DLY_TCO_2_PLAUS - 1
                        // conditions causing change in TCO_2_MES are satisfied, decrement counter
                IF(4c)        CTR_DLY_TCO_2_PLAUS = 0
                THEN(4c)      LV_TCO_2_PLAUS_ENA = 1
```

```
                ELSE(4c)    LV_TCO_2_PLAUS_ENA = 0
                ENDIF(4c)
            ELSE(3c)    CTR_DLY_TCO_2_PLAUS = CTR_DLY_TCO_2_PLAUS
                            + C_CTR_DLY_INC_TCO_2_PLAUS
                        // Conditions causing change in TCO_2_MES not fulfilled, increment counter
                IF(5c)      CTR_DLY_TCO_2_PLAUS >=
                                C_CTR_DLY_TCO_2_PLAUS
                THEN(5c)    CTR_DLY_TCO_2_PLAUS =
                                C_CTR_DLY_TCO_2_PLAUS
                ENDIF(5c)
            ENDIF(3c)
        ELSE(2c)    CTR_DLY_TCO_2_PLAUS = CTR_DLY_TCO_2_PLAUS          //freeze
        ENDIF(2c)
```

D Start diagnostic window / Diagnostic check:

```
        IF(2d)      LV_TCO_2_PLAUS_ENA = 1
        THEN(2d)    T_DLY_TCO_2_PLAUS - -
                    //decrement timer for delay-time
                    // lower decrement timer value limited to 0 s, no negative values possible
            IF(3d)      T_DLY_TCO_2_PLAUS = 0                          AND
                        LV_INH_DIAG_TCO_2_PLAUS = 0                    AND
                        LV_CDN_VB_OBD2 = 1
            THEN(3d)
                IF(4d)      LV_CDN_DIAG_TCO_2_PLAUS = 1
                THEN(4d)    ERR_SYM_TCO_2_PLAUS = "SYM_3"
                            LV_ERR_TCO_2_PLAUS = 1         // without debounce
                            LV_END_DIAG_TCO_2_PLAUS = 1        // end of diagnosis
                ENDIF(4d)
                LV_END_DIAG_RBM_TCO_2_PLAUS = 1            // end of diagnosis for RBM
            ENDIF(3d)
        ELSE(2d)    no start condition
        ENDIF(2d)
```

Calculation of Plausibility after start condition:

```
    ELSEIF(1) TCO_ST < C_TCO_ST_THD_TCO_2_PLAUS                        AND
              TIA_THR_ST <= C_TIA_THR_ST_MIN_TCO_2_PLAUS               AND
              LV_END_DIAG_RBM_TCO_2_PLAUS = 0
```

```
THEN(1)    (calculation of formula section E – “SYM_0” error check)

------------------------------------------------------------------------------------------

E   Calculation of Plausibility after start :

    IF(2e)        LV_CDN_DIAG_TCO_2_PLAUS = 1
    THEN(2e)
        IF(3e)        TCO_2_MES_ST – TCO_ST >
                      C_TCO_ST_DIF_MIN_TCO_2_PLAUS
        THEN(3e)      ERR_SYM_TCO_2_PLAUS = “SYM_0”
                      LV_ERR_TCO_2_PLAUS = 1                 // without debounce
        ENDIF(3e)
        LV_END_DIAG_TCO_2_PLAUS = 1                          // end of diagnosis
        LV_END_DIAG_RBM_TCO_2_PLAUS = 1                      // end of diagnosis for RBM
    ENDIF(2e)

------------------------------------------------------------------------------------------

ELSE(1)    (TCO_2 diagnosis finished or diagnosis conditions not met)
ENDIF(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_DLY_TCO_2_PLAUS	1	0...FFFFH	0...65535	1	[-]
Coolant system lag time before change in TCO_2_MES is expected					
C_T_DLY_TCO_2_PLAUS	1	0…FFFFH	65535	1	s
Diagnostic window					
C_TCO_2_DIF_MIN_TCO_2_PLAUS	1	0…FEH	0…190.5	0.75	°C
Min. delta TCO_2_MES for plausibility					
C_TCO_MIN_TCO_2_PLAUS	1	0…FEH	-48…142.5	0.75	°C
Opening temperature of coolant thermostat					
C_VS_MIN_TCO_2_PLAUS	1	0…FFH	0…255	1	km/h
Min. vehicle speed needed to open thermostat relief valve					
C_N_MIN_TCO_2_PLAUS	1	0…1FE0H	0…8160	1	rpm
Min. engine speed needed to open thermostat relief valve					
C_TCO_ST_MAX_TCO_2_PLAUS	1	0…FEH	-48…142.5	0.75	°C
Max. TCO_ST for TCO_2 plausibility check					
C_TCO_ST_MIN_TCO_2_PLAUS	1	0…FEH	-48…142.5	0.75	°C
Min. TCO_ST for TCO_2 plausibility check					
C_TIA_THR_ST_MAX_TCO_2_PLAUS	1	0…FEH	-48…142.5	0.75	°C
Max. TIA_THR_ST for TCO_2 plausibility check					
C_TIA_THR_ST_MIN_TCO_2_PLAUS	1	0…FEH	-48…142.5	0.75	°C
Min. TIA_THR_ST for TCO_2 plausibility check					
C_TCO_ST_THD_TCO_2_PLAUS	1	0…FEH	-48…142.5	0.75	°C
Max. TCO_ST for additional rationality check					
C_TCO_ST_DIF_MIN_TCO_2_PLAUS	1	0…FFH	-96…95.5	0.75	°C
Min. difference between TCO_ST and TCO_2_MES_ST to diagnose fault at engine start					
C_CTR_DLY_INC_TCO_2_PLAUS	1	0...FFFFH	0...65535	1	-
Increment for counter if drop out of diagnostic conditions					

26.32 TCO sensor (radiator outlet) plausibility diagnosis (Appl. Inc.)

26.32.1 Interface parameter

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_TCO_2_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal plausibility diagnosis inhibit					
LV_INH_DIAG_RBM_TCO_2_PLAUS	V	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal plausibility diagnosis inhibit due to OBD error					
LV_TCO_2_PLAUS_ENA_EXT	V/O	0...1H	0...1	1	[-]
Boolean for coolant temperature (radiator outlet) signal plausibility diagnosis external enable					

Input data:

LV_IGK	LV_ERR_TCO	LV_ERR_TCO_2_EL	LV_ERR_TCO_2_GRD
LV_ERR_TIA	LV_ERR_VS	LV_ERR_ECT	LV_ST_END
ECTPWM			

General information:

Depending on project specific requirements, the plausibility diagnosis of the coolant temperature sensor at radiator outlet can be inhibited by setting of the logical variable LV_INH_DIAG_TCO_2_PLAUS. As a default value LV_INH_DIAG_TCO_2_PLAUS is set as shown below.

Application conditions:

Initialisation at RESET **or** EXIT_ST **or** LV_IGK = 0 -> 1 **or** FMY clear:
LV_INH_DIAG_TCO_2_PLAUS = 0
LV_INH_DIAG_RBM_TCO_2_PLAUS = 0
LV_TCO_2_PLAUS_ENA_EXT = 0

Recurrence: 1000 ms

Activation: LV_ST_END = 1
(Calculation of the TCO_2 plausi. diagnosis interface parameter enabled)

Deactivation: LV_ST_END = 0
(Calculation of the TCO_2 plausi. diagnosis interface parameter disabled)

Formula section:

Calculation of diagnosis interface parameter:

If LV_ERR_TCO = 1 **or**
LV_ERR_TCO_2_EL = 1 **or**
LV_ERR_TCO_2_GRD = 1 **or**
LV_ERR_TIA = 1 **or**
LV_ERR_VS = 1 **or**
LV_ERR_ECT = 1

```
Then     LV_INH_DIAG_RBM_TCO_2_PLAUS = 1
Endif

If       LV_IGK = 0                                    or
         LV_INH_DIAG_RBM_TCO_2_PLAUS = 1
Then     LV_INH_DIAG_TCO_2_PLAUS = 1
Endif

If       ECTPWM > C_ECTPWM_MIN_TCO_2_PLAUS
Then     LV_TCO_2_PLAUS_ENA_EXT = 1
Else     LV_TCO_2_PLAUS_ENA_EXT = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ECTPWM_MIN_TCO_2_PLAUS	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Minimum ECTPWM to expect a change in TCO_2					

26.32.2 Interface for Rate-based-monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_TCO_2_PLAUS	V/O	0...7H	0...7	1	[-]
Interface of TCO_2 monitor with the Rate-Based Monitoring statistics					

Input data:

LV_DC	CTR_ERR_DYN_NR	LV_END_DIAG_RBM_TCO_2_PLAUS	LV_CDN_CST_RBM

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

General information:

With this module the interface between the TCO_2 plausibility monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TCO_2_PLAUS data.

Within STATE_RBM_TCO_2_PLAUS, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

Application conditions:

Initialisation :

at ECU reset :

bit 0, bit 1 and bit 2 of STATE_RBM_TCO_2_PLAUS = 0

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_TCO_2_PLAUS = 0

on failure memory reset :

bit 1 of STATE_RBM_TCO_2_PLAUS = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_TCO_EL	LV_ERR_TCO_PLAUS	LV_ERR_TCO_GRD	LV_ERR_TCO_STUCK
LV_ERR_TCO_2_EL	LV_ERR_TCO_2_GRD	LV_ERR_TIA_EL	LV_ERR_TIA_PLAUS
LV_ERR_VS	LV_ERR_TH	LV_ERR_ECT_EL	

```
If(1)     { CPU optimization at LV_DC 0 → 1 transition }
          CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
          While bit 1 of STATE_RBM_TCO_2_PLAUS = 0 do
                with each XX failure of the above list :
                ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                SYNCHRONIZATION<CALL>)
                If(2)        XX has a pending status
                Then(2)
                             bit 1 of STATE_RBM_TCO_2_PLAUS = 1
                Endif(2)
```

```
            Endwhile
    Else(1)
            { the dynamic failure memory is empty }
            No action
    Endif(1)

  Every 1 s :
  If    bit 0 of STATE_RBM_TCO_2_PLAUS = 0
  Then
            If      LV_END_DIAG_RBM_TCO_2_PLAUS = 1
            Then    bit 0 of STATE_RBM_TCO_2_PLAUS = 1
            Endif
  Endif

  If    bit 1 of STATE_RBM_TCO_2_PLAUS = 0
  Then
            If      LV_INH_DIAG_RBM_TCO_2_PLAUS = 1
            Then    bit 1 of STATE_RBM_TCO_2_PLAUS = 1
            Endif
  Endif

  If    bit 2 of STATE_RBM_TCO_2_PLAUS = 0
  Then
            If      LV_CDN_CST_RBM = 1
            Then    bit 2 of STATE_RBM_TCO_2_PLAUS = 1
            Endif
  Endif
```

26.33 Coolant thermostat monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_TH	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptom for coolant thermostat error					
LV_ACT_DIAG_RBM_TH	O/V	0...1H	0...1	1	[-]
Boolean for coolant thermostat diagnosis activated for RBM					
LV_ACT_DIAG_TH	O/V	0...1H	0...1	1	[-]
Boolean for coolant thermostat diagnosis activated					
LV_CDN_DIAG_TH	O/V	0...1H	0...1	1	[-]
Boolean for coolant thermostat diagnosis conditions					
LV_CNL_DIAG_TH	O/V	0...1H	0...1	1	[-]
Boolean for coolant thermostat diagnosis cancelled					
LV_END_DIAG_RBM_TH	O/V	0...1H	0...1	1	[-]
Boolean for end of coolant thermostat diagnosis for RBM					
LV_END_DIAG_RBM_TH_TMP	O/V	0...1H	0...1	1	[-]
Temporary bit for end of coolant thermostat diagnosis for RBM					
LV_END_DIAG_TH	O/V	0...1H	0...1	1	[-]
Boolean for end of coolant thermostat diagnosis					
LV_ERR_TH	O/V	0...1H	0...1	1	[-]
Boolean for coolant thermostat error					
LV_TCO_DE_MAX_DIAG_TH	O/V	0...1H	0...1	1	[-]
Boolean for high coolant temperature deviation for coolant thermostat diagnosis					
T_TCO_DE_DIAG_TH	O/V	0...FFFFH	0...65535	1	[s]
Timer for coolant temperature deviation for coolant thermostat diagnosis					
TCO_DE_DIAG_TH	O/V	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature deviation value for coolant thermostat diagnosis					
TCO_DE_INI_DIAG_TH	O/V	0...FEH	-48...142.5	0.75	[°C]
Initialization value for coolant temperature deviation for coolant thermostat diagnosis					
TCO_DIAG_TH	O/V	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature of the performed coolant thermostat diagnosis					
TIA_DE_DIAG_TH	O/V	0...FEH	-48...142.5	0.75	[°C]
Intake air temperature deviation value for coolant thermostat diagnosis					
TIA_MIN_DIAG_TH	O/V	0...FEH	-48...142.5	0.75	[°C]
Minimum intake air temperature value during coolant thermostat diagnosis					

Input data:

LV_CDN_VB_OBD2	LV_INH_DIAG_TH	LV_ST_END	N
N_REL_CWP	RATIO_T_LOAD_MIN_AST	RATIO_T_PUC_AST	RATIO_T_VS_MAX_AST
RATIO_T_VS_MIN_AST	TCO_MES	TCO_ST	TCO_SUB
TIA_THR[NC_SENS_NR_TIA]	TIA_THR_ST	TAM	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_DIAG_TH	1	0...1FE0H	0...8160	1	[rpm]
Maximum engine speed for high coolant temperature deviation detection for coolant thermostat diagnosis					
C_N_REL_CWP_MAX_DIAG_TH	1	0...FFH	0...255	1	[-]
Maximum cooling water pump speed for coolant thermostat diagnosis					
C_RATIO_T_VS_MAX_THD_DIAG_TH	1	0...FFH	0...99.60937	0.390625	[%]
"Vehicle speed maximum" ratio threshold in per cent for coolant thermostat diagnosis					

C_T_TCO_DE_DIAG_TH	1	0...FFFFH	0...65535	1	[s]
Time for high coolant temperature deviation detection for coolant thermostat diagnosis					
C_TCO_DE_MAX_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Maximum coolant temperature deviation for coolant thermostat diagnosis					
C_TCO_ST_MAX_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Maximum coolant temperature at start for coolant thermostat diagnosis					
C_TCO_ST_MIN_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant temperature at start for coolant thermostat diagnosis					
C_TCO_TH_OPEN_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Regulation temperature for coolant thermostat opening					
C_TCO_WUP_MIN_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant warm up temperature for coolant thermostat diagnosis					
C_TIA_DE_MAX_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Maximum intake air temperature deviation for coolant thermostat diagnosis					
C_TIA_ST_MIN_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Minimum intake air temperature at start for coolant thermostat diagnosis					
IP_RATIO_T_LOAD_MIN_THD_DIAG_TH	4	0...FFH	0...99.60937	0.390625	[%]
LDPM_TAM_IP_RATIO_DIAG_TH	4	0...FEH	-48...142.5	0.75	[°C]
'Engine load minimum' ratio threshold in per cent for coolant thermostat diagnosis					
IP_RATIO_T_PUC_THD_DIAG_TH	4	0...FFH	0...99.60937	0.390625	[%]
LDPM_TAM_IP_RATIO_DIAG_TH	4	0...FEH	-48...142.5	0.75	[°C]
'Pull fuel cut off' ratio threshold in per cent for coolant thermostat diagnosis					
IP_RATIO_T_VS_MIN_THD_DIAG_TH	4	0...FFH	0...99.60937	0.390625	[%]
LDPM_TAM_IP_RATIO_DIAG_TH	4	0...FEH	-48...142.5	0.75	[°C]
'Vehicle speed minimum' ratio threshold in per cent for coolant thermostat diagnosis					

FUNCTION DESCRIPTION:

General information:

Task of the coolant thermostat is to effect a quick engine warm up after start. The thermostat is closed after engine start to limit the cooling-liquid circulation until the thermostat regulating temperature is reached. The thermostat opens at this temperature and the limitation of the cooling-liquid circulation is finished. If an opened stuck thermostat occurs, the cooling-liquid circulation will not be limited after start. That means an increase of the engine warm up time and can cause emission increase as well.

To take this behaviour into consideration, an opened stuck thermostat must be detected if either other diagnosis functions are inhibited or the coolant temperature does not reach its operating temperature within 20°F. The conditions for inhibiting the diagnosis are described in the chapter “Application incidences for coolant thermostat monitoring”. The coolant temperature model (TCO_SUB) must simulate the worst engine warm up temperature behaviour for a normal working thermostat (heating on. A/C on. etc.) after start until the thermostat opening temperature is reached.

Description:

The error detection is done by a comparison between the raw value of the coolant temperature sensor and the value of the coolant temperature substitute model.

Error-symptoms are defined to this diagnosis function as following:

☐☐☐☐

```
 │ │ │ └─> -                       (= SYM_0)
 │ │ └───> -                       (= SYM_1)
 │ └─────> -                       (= SYM_2)
 └───────> TH open stuck           (= SYM_3)
```

26.33.1 Activation condition for thermostat monitoring

FUNCTION DESCRIPTION:

General information:

The initialization of the function is performed as soon as EXIT_ST is detected. In this case it is regardless if the actual thermostat diagnosis status is set to enabled or canceled.

Thermostat diagnosis is performed as soon as all conditions below are fulfilled at the same time. The function is enabled at once if LV_ST_END is set to “1”. A minimum and a maximum threshold for the coolant temperature at start are limiting the diagnosis activation range as well as a minimum threshold for the intake air temperature at start.

Thermostat monitoring is finished if either the logical value for “thermostat diagnosis finished” or the logical value for “thermostat diagnosis cancelled” equals to “1”. In this case the diagnosis activation bit is set to “0” (LV_ACT_DIAG_TH = 0).

Application conditions:

Initialisation at EXIT_ST:

LV_ERR_TH = 0

LV_CDN_DIAG_TH = 0

LV_END_DIAG_TH = 0

ERR_SYM_TH = “NO_SYM”

LV_END_DIAG_RBM_TH = 0

LV_END_DIAG_RBM_TH_TMP = 0

LV_ACT_DIAG_RBM_TH = 0

LV_CNL_DIAG_TH = 0

LV_ACT_DIAG_TH = 0

TCO_DIAG_TH = -48°C

LV_TCO_DE_MAX_DIAG_TH = 0

Recurrence: 1000 ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

Calculation of the functionality check (RBM) enable condition:

```
If        TCO_ST > C_TCO_ST_MIN_DIAG_TH        and
          TCO_ST < C_TCO_ST_MAX_DIAG_TH        and
          TIA_THR_ST > C_TIA_ST_MIN_DIAG_TH    and
          LV_CDN_VB_OBD2 = 1                   and
          LV_INH_DIAG_TH = 0                   and
          LV_CNL_DIAG_TH = 0
then      LV_ACT_DIAG_RBM_TH = 1
else      LV_ACT_DIAG_RBM_TH = 0
endif
```

Calculation of the functionality check enable condition:

```
If        LV_ACT_DIAG_RBM_TH = 1               and
          LV_END_DIAG_TH = 0
then      LV_ACT_DIAG_TH = 1
else      LV_ACT_DIAG_TH = 0
endif
```

Calculation of the intake air temperature deviation:

```
If (1)         TCO_SUB <= C_TCO_TH_OPEN_DIAG_TH     and
               LV_ACT_DIAG_RBM_TH = 1
then (1)
    if (2)     TIA_THR < TIA_MIN_DIAG_TH $_{(n-1)}$
    then (2)   TIA_MIN_DIAG_TH $_{(n)}$ = TIA_THR
    else (2)   TIA_MIN_DIAG_TH $_{(n)}$ = TIA_MIN_DIAG_TH $_{(n-1)}$
    endif (2)
else (1)       TIA_DE_DIAG_TH = TIA_MIN_DIAG_TH - TIA_THR_ST
endif (1)
```

Calculation of the coolant temperature deviation during high engine speed:

```
If (1)         TCO_SUB <= C_TCO_TH_OPEN_DIAG_TH     and
               LV_ACT_DIAG_RBM_TH = 1
then (1)
    if (2)     N >= C_N_MAX_DIAG_TH
               OR  N_REL_CWP > C_N_REL_CWP_MAX_DIAG_TH
    then (2)   (Trigger the coolant temperature deviation calculation)
```

```
                    1st. step: "Initialization" : No re-triggering until timer ends
                    TCO_DE_INI_DIAG_TH = TCO_MES
                    TCO_DE_DIAG_TH = TCO_MES - TCO_DE_INI_DIAG_TH
                    T_TCO_DE_DIAG_TH = C_T_TCO_DE_DIAG_TH
                    Timer T_TCO_DE_DIAG_TH starts

                    2nd. step: "Check for coolant temperature deviations"
                    TCO_DE_DIAG_TH = TCO_MES - TCO_DE_INI_DIAG_TH

          if (3)    TCO_DE_DIAG_TH < C_TCO_DE_MAX_DIAG_TH
          then  (3) LV_TCO_DE_MAX_DIAG_TH = 1
          endif (3)
     endif (2)
endif (1)
```

Calculation of the diagnosis result:

```
If (1)              LV_ACT_DIAG_TH = 1
then (1)            // Coolant thermostat functionality check enabled
                    // Calculation of chapter 1.2
endif (1)
```

Calculation of the diagnosis condition:

```
If (1)              LV_END_DIAG_TH = 1                    or
                    LV_INH_DIAG_TH = 1                    or
                    LV_CNL_DIAG_TH = 1
then (1)            LV_CDN_DIAG_TH = 0
    If(2)           Transition LV_END_DIAG_TH = 0 to LV_END_DIAG_TH = 1
    then(2)         TCO_DIAG_TH = TCO_MES
    endif (2)
endif (1)
```

Calculation of end of diagnosis for RBM:

```
If (1)              LV_ACT_DIAG_RBM_TH = 1                        and
                    TCO_SUB > C_TCO_TH_OPEN_DIAG_TH               and
```

```
                LV_END_DIAG_RBM_TH_TMP = 0                  and
                LV_CNL_DIAG_TH = 0
then (1)
                LV_END_DIAG_RBM_TH_TMP = 1
    if (2)      TCO_MES >= C_TCO_WUP_MIN_DIAG_TH
    then (2)
        if (3a)     RATIO_T_VS_MIN_AST <=
                        IP_RATIO_T_VS_MIN_THD_DIAG_TH       and
                    RATIO_T_LOAD_MIN_AST <=
                        IP_RATIO_T_LOAD_MIN_THD_DIAG_TH
        then (3a)   LV_END_DIAG_RBM_TH = 1
        endif (3a)
    else (2)
        if (3b)     RATIO_T_PUC_AST <=
                        IP_RATIO_T_PUC_THD_DIAG_TH              and
                    RATIO_T_LOAD_MIN_AST <=
                        IP_RATIO_T_LOAD_MIN_THD_DIAG_TH         and
                    RATIO_T_VS_MAX_AST <=
                        C_RATIO_T_VS_MAX_THD_DIAG_TH            and
                    TIA_DE_DIAG_TH >= C_TIA_DE_MAX_DIAG_TH
    and
                    LV_TCO_DE_MAX_DIAG_TH = 0
        then (3b)   LV_END_DIAG_RBM_TH = 1
        endif (3b)
    endif (2)
endif (1)
```

26.33.2 Coolant thermostat functionality check

<u>FUNCTION DESCRIPTION:</u>

<u>General information:</u>

Task of the functionality check is to detect a stuck open thermostat. Correct functionality is not guaranteed if the coolant temperature does not reach a specified warmed up temperature.

The thermostat functionality check is enabled as long as the boolean for “coolant thermostat diagnosis activated” is set to "1". As soon as EXIT_ST is detected, all timers and initialization values are set to neutral values.

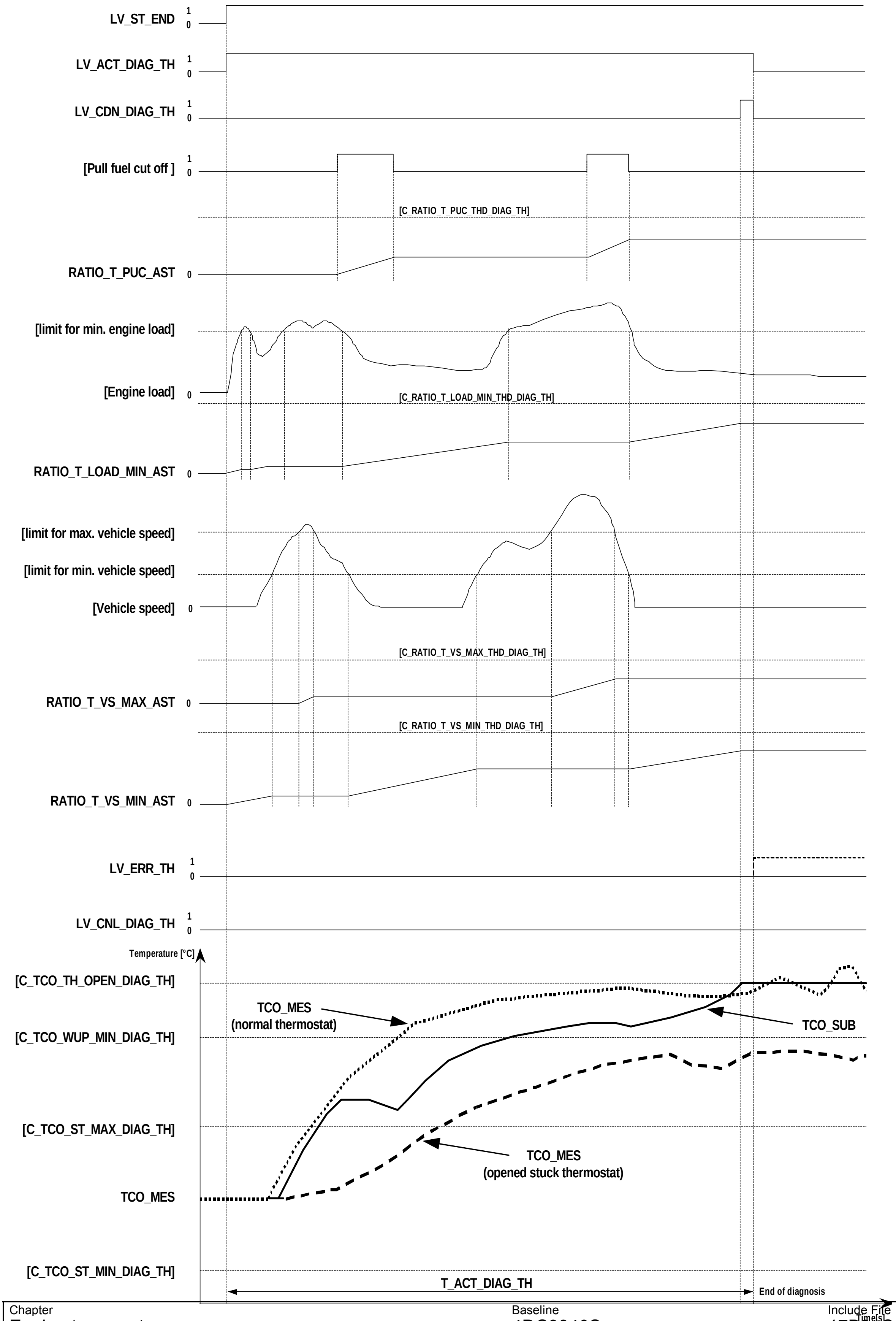
Application conditions:

Initialisation at EXIT_ST:

TIA_DE_DIAG_TH = 0°C
T_TCO_DE_DIAG_TH = 0s
TCO_DE_INI_DIAG_TH = 0°C
TCO_DE_DIAG_TH = 0°C
TIA_MIN_DIAG_TH = TIA_THR_ST

Recurrence:	1000 ms
Activation:	LV_ACT_DIAG_TH = 1 *(Coolant thermostat functionality check enabled)*
Deactivation:	LV_ACT_DIAG_TH = 0 *(Coolant thermostat functionality check disabled)*

Signal flow diagram:



Formula section:

Calculation of the diagnosis result:

```
If (1)              TCO_MES > C_TCO_TH_OPEN_DIAG_TH
then (1)
    if (2)          RATIO_T_VS_MIN_AST > IP_RATIO_T_VS_MIN_THD_DIAG_TH
                or  RATIO_T_LOAD_MIN_AST > IP_RATIO_T_LOAD_MIN_THD_DIAG_TH
    then (2)        LV_CDN_DIAG_TH = 0
                    LV_CNL_DIAG_TH = 1
    else (2)        LV_CDN_DIAG_TH = 1
                    ERR_SYM_TH = “NO_SYM”
                    LV_CNL_DIAG_TH = 0
                    LV_END_DIAG_TH = 1                    (calculated in ABC)
    endif (2)
else (1)
    if (2)          TCO_SUB > C_TCO_TH_OPEN_DIAG_TH
    then (2)
        if (3)      TCO_MES >= C_TCO_WUP_MIN_DIAG_TH
        then (3)
          if        RATIO_T_VS_MIN_AST > IP_RATIO_T_VS_MIN_THD_DIAG_TH
                or  RATIO_T_LOAD_MIN_AST > IP_RATIO_T_LOAD_MIN_THD_DIAG_TH
          then      LV_CDN_DIAG_TH = 0
                    LV_CNL_DIAG_TH = 1
          else      LV_CDN_DIAG_TH = 1
                    ERR_SYM_TH = “NO_SYM”
                    LV_END_DIAG_TH = 1                    (calculated in ABC)
                    LV_CNL_DIAG_TH = 0
          endif
        else (3)    (In case of "TCO_MES less than C_TCO_WUP_MIN_DIAG_TH ")
          if        ( RATIO_T_PUC_AST > IP_RATIO_T_PUC_THD_DIAG_TH
                or  RATIO_T_LOAD_MIN_AST > IP_RATIO_T_LOAD_MIN_THD_DIAG_TH
                or  RATIO_T_VS_MAX_AST > C_RATIO_T_VS_MAX_THD_DIAG_TH
                or  TIA_DE_DIAG_TH < C_TIA_DE_MAX_DIAG_TH
                or  LV_TCO_DE_MAX_DIAG_TH = 1 )
          then      LV_CDN_DIAG_TH = 0
                    LV_CNL_DIAG_TH = 1
```

```
                else    LV_CDN_DIAG_TH = 1
                        ERR_SYM_TH = “SYM_3”
                        LV_ERR_TH = 1                    (after debounce)
                        LV_END_DIAG_TH = 1               (calculated in ABC)
                        LV_CNL_DIAG_TH = 0
                endif
            endif (3)
        endif (2)
    endif (1)
```

26.34 Coolant thermostat monitoring (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_TH	V/O	0...1H	0...1	1	[-]
Boolean for coolant thermostat diagnosis inhibit					
LV_INH_DIAG_RBM_TH	V	0...1H	0...1	1	[-]
Boolean for coolant thermostat diagnosis inhibit due to OBD error					
LV_N_REL_CWP_INH_DIAG_TH	V	0...1H	0...1	1	[-]
Boolean for coolant thermostat diagnosis inhibit due high cooling water pump speed					
LV_TIA_DE_INH_DIAG_TH	V	0...1H	0...1	1	[-]
Boolean for coolant thermostat diagnosis inhibit due high intake air temperature deviation					
MASS_VOL_CWP_DIAG_TH	V	0...FFFFH	0...65535	1	[-]
Cooling water pump effected coolant mass volume for coolant thermostat diagnosis					
MASS_VOL_CWP_DIAG_TH_TMP	V	0...FFFFH	0...65535	1	[-]
Cooling water pump effected coolant mass volumetric-flow for coolant thermostat diagnosis					
TIA_MIN_INH_DIAG_TH	V	0...FEH	-48...142.5	0.75	[°C]
Minimum Intake air temperature value during coolant thermostat diagnosis for inhibition					
TIA_DE_INH_DIAG_TH	V	0...FEH	-48...142.5	0.75	[°C]
Intake air temperature deviation value for coolant thermostat diagnosis inhibition					
T_ECTPWM_INH_DIAG_TH	V	0...FFH	0...255	1	[s]
Max. time for ECTPWM exceeding threshold - for coolant thermostat diagnosis					
T_ECFPWM_INH_DIAG_TH	V	0...FFH	0...255	1	[s]
Max. time for ECFPWM exceeding threshold - for coolant thermostat diagnosis					
LV_ECTPWM_INH_DIAG_TH	V	0...1H	0...1	1	[-]
Condition for ECTPWM exceeding threshold - for coolant thermostat diagnosis					
LV_ECFPWM_INH_DIAG_TH	V	0...1H	0...1	1	[-]
condition for ECFPWM exceeding threshold - for coolant thermostat diagnosis					
T_N_REL_CWP_MAX_DIAG_TH	V	0...FFH	0...255	1	[s]
Max. time for CWP speed exceeding threshold before coolant thermostat diagnosis inhibition					
STATE_RBM_TH	V/O	0...7H	0...7	1	[-]
Interface of thermostat monitoring with the RBM statistics,Bit0: cond. for monitoring are met long enough to detect malfunction,Bit1: inhib. of the monitor due to system failures,Bit2: indiv. RBM cond. of the monitor were encount. within this DC					

Input data:

LV_IGK	LV_ST_END	LV_ERR_TCO	LV_ERR_MAF
LV_ERR_TIA	LV_ERR_VS	N_REL_CWP	TIA_THR
TIA_THR_ST	TAM	LV_ERR_TAM	LV_ERR_ECT_EL
ECTPWM	ECFPWM[NC_ECF_NR]	LV_DC	CTR_ERR_DYN_NR
LV_END_DIAG_RBM_TH	TCO_SUB		

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

The coolant thermostat diagnosis is performed at transition from the logical value for “engine stop” from “1” to “0”. If the engine stalls or one of the conditions for inhibiting the diagnosis is present, it will be stopped and starts again at the next engine start.

The diagnosis is finished as soon as the boolean for end of coolant thermostat diagnosis is set to “1”. In this case the boolean for coolant thermostat error indicates either an open stuck thermostat (= 1) or a normal working thermostat (= 0).

Depending on project specific requirements, the coolant thermostat diagnosis can be inhibited by setting of LV_INH_DIAG_TH.
Interface of thermostat monitoring with the RBM statistics:
Bit 0: cond. for monitoring are met long enough to detect malfunction (bit0=1)
Bit 1: inhib. of the monitor because of system failure(s) (bit1=1)
Bit 2: individual RBM cond. of the monitor were encountered within this DC (bit2=1)

Application conditions:

Initialisation: at RESET **or** EXIT_ST **or** LV_IGK = 0->1 **or** FMY clear:

LV_INH_DIAG_TH = 0
LV_INH_DIAG_RBM_TH = 0
LV_TIA_DE_INH_DIAG_TH = 0
LV_N_REL_CWP_INH_DIAG_TH = 0
LV_ECTPWM_INH_DIAG_TH = 0
LV_ECFPWM_INH_DIAG_TH = 0
T_ECTPWM_INH_DIAG_TH = C_T_ECTPWM_INH_DIAG_TH
T_ECFPWM_INH_DIAG_TH = C_T_ECFPWM_INH_DIAG_TH
T_N_REL_CWP_MAX_DIAG_TH = C_T_N_REL_CWP_MAX_DIAG_TH

at RESET or EXIT_ST:

MASS_VOL_CWP_DIAG_TH_TMP = 0l
MASS_VOL_CWP_DIAG_TH = 0l
TIA_DE_INH_DIAG_TH = 0°C
TIA_MIN_INH_DIAG_TH = TIA_THR_ST

Recurrence: 1000 ms

Activation: LV_ST_END = 1

(Coolant thermostat monitoring (Appl. Inc.)” function enabled)

Deactivation: LV_ST_END = 0

(Coolant thermostat monitoring (Appl. Inc.)” function disabled)

Formula section:

Calculation of Cooling water pump effected coolant mass volumetric flow:

```
MASS_VOL_CWP_DIAG_TH_TMP=(N_REL_CWP / 255)*C_MASS_VOL_CWP_MAX
//phs!
```

Calculation of the maximum CWP speed for TH monitoring:

```
If      N_REL_CWP > ID_N_REL_CWP_MAX_DIAG_TH
Then    If      T_N_REL_CWP_MAX_DIAG_TH > 0
        Then    T_N_REL_CWP_MAX_DIAG_TH - -              //count down
        Else    LV_N_REL_CWP_INH_DIAG_TH = 1             //set until next Ini
        Endif
Else    T_N_REL_CWP_MAX_DIAG_TH = C_T_N_REL_CWP_MAX_DIAG_TH
Endif
```

Calculation of the maximum intake air temperature deviation:

```
If      TIA_THR < TIA_MIN_INH_DIAG_TH $_{(n-1)}$
then    TIA_MIN_INH_DIAG_TH $_{(n)}$ = TIA_THR
else    TIA_MIN_INH_DIAG_TH $_{(n)}$ = TIA_MIN_INH_DIAG_TH $_{(n-1)}$
endif

TIA_DE_INH_DIAG_TH = TIA_MIN_INH_DIAG_TH - TIA_THR_ST

If      TIA_DE_INH_DIAG_TH < C_TIA_DE_MAX_INH_DIAG_TH
then    LV_TIA_DE_INH_DIAG_TH = 1                        //set until next Ini
endif
```

Calculation of the minimum CWP volume for TH monitoring:

```
If (1)      LV_N_REL_CWP_INH_DIAG_TH = 0                 and
            MASS_VOL_CWP_DIAG_TH(n) <=
                    C_MASS_VOL_CWP_MIN_DIAG_TH
then (1)
    If (2)          N_REL_CWP > C_N_REL_CWP_MIN_DIAG_TH
    then (2)        MASS_VOL_CWP_DIAG_TH$_{(n)}$ =       //Integration
                    MASS_VOL_CWP_DIAG_TH$_{(n-1)}$ + MASS_VOL_CWP_DIAG_TH_TMP$_{(n)}$
    else (2)        MASS_VOL_CWP_DIAG_TH$_{(n)}$ = 0
    endif (2)
else (1)            stop of integration, keep last value
```

```
endif (1)
```

Calculation of the ECTPWM condition for TH monitoring:

```
If      ECTPWM > C_ECTPWM_INH_DIAG_TH
then    if      T_ECTPWM_INH_DIAG_TH > 0
        then    T_ECTPWM_INH_DIAG_TH - -
        else    LV_ECTPWM_INH_DIAG_TH = 1
        endif
else    T_ECTPWM_INH_DIAG_TH = C_T_ECTPWM_INH_DIAG_TH
Endif
```

Calculation of the ECFPWM condition for TH monitoring:

```
If      ECFPWM > C_ECFPWM_INH_DIAG_TH
then    if      T_ECFPWM_INH_DIAG_TH > 0
        then    T_ECFPWM_INH_DIAG_TH - -
        else    LV_ECFPWM_INH_DIAG_TH = 1
        endif
else    T_ECFPWM_INH_DIAG_TH = C_T_ECFPWM_INH_DIAG_TH
endif
```

Calculation of diagnosis interface parameter:

```
If      LV_ERR_TCO = 1          or
        LV_ERR_MAF = 1          or
        LV_ERR_TIA = 1          or
        LV_ERR_VS = 1           or
        LV_ERR_TAM = 1          or
        LV_ERR_ECT_EL = 1
then    LV_INH_DIAG_RBM_TH = 1
endif
```

```
If      LV_INH_DIAG_RBM_TH = 1                                or
        LV_N_REL_CWP_INH_DIAG_TH = 1                      or
        LV_TIA_DE_INH_DIAG_TH = 1                         or
        TAM < C_TAM_MIN_DIAG_TH                           or
        MASS_VOL_CWP_DIAG_TH(n)  <  C_MASS_VOL_CWP_MIN_DIAG_TH or
        LV_ECTPWM_INH_DIAG_TH = 1   or
        LV_ECFPWM_INH_DIAG_TH = 1

then    LV_INH_DIAG_TH = 1
else    LV_INH_DIAG_TH = 0
endif
```

Error treatment:

Error debounce:

Debounce counter increment:	C_ABC_INC_TH
Debounce counter maximum value:	C_ABC_MAX_TH

Error type:

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Recurence	Failure class A/B
Coolant Thermostat Monitoring	*Not used*	*0*				STD_INI
	Not used	*1*				
	TH open stuck	*2*				
TH Monitoring	*Not used*	*3*				

26.34.2 Rate Base Monitoring interface for thermostat monitoring.

FUNCTION DESCRIPTION:

General information:

With this module the interface between the thermostat monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TH data.

Within STATE_RBM_TH, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for TH diagnosis)

Application conditions:

Initialisation :

at ECU reset :

bit 0, bit 1 and bit 2 of STATE_RBM_TH = 0

at LV_DC 0 → 1 transition :

bit 0 and bit 1of STATE_RBM_TH = 0

bit 2 of STATE_RBM_TH = 1

on failure memory reset :

bit 1 of STATE_RBM_TH = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_TCO_EL	LV_ERR_TCO_GRD	LV_ERR_TCO_STUCK	LV_ERR_TCO_PLAUS
LV_ERR_TIA_EL	LV_ERR_TIA_PLAUS	LV_ERR_MAF	LV_ERR_VS
LV_ERR_TAM_CAN	LV_ERR_TAM_PLAUS	LV_ERR_ECT_EL	

```
If(1)    { CPU optimization at LV_DC 0 → 1 transition }
         CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
         While  bit 1 of STATE_RBM_TH = 0 do
                with each XX failure of the above list :
```

```
                    ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                    SYNCHRONIZATION<CALL>)
                    If(2)        XX has a pending status
                    Then(2)
                                 bit 1 of STATE_RBM_TH = 1
                    Endif(2)
            Endwhile
  Else(1)
            { the dynamic failure memory is empty }
            No action
  Endif(1)
Every 1 s :
If    bit 0 of STATE_RBM_TH = 0
Then
            If      LV_END_DIAG_RBM_TH = 1
            Then    bit 0 of STATE_RBM_TH = 1
            Endif
Endif
If    bit 1 of STATE_RBM_TH = 0
Then
            If      LV_INH_DIAG_RBM_TH = 1
            Then    bit 1 of STATE_RBM_TH = 1
            Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TH	1	0...FFH	0...255	1	[-]
Debounce counter increment value for the coolant thermostat diagnosis					
C_ABC_MAX_TH	1	1...FFH	1...255	1	[-]
Debounce counter maximum value for the coolant thermostat diagnosis					
C_TIA_DE_MAX_INH_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Maximum intake air temperature deviation for coolant thermostat diagnosis inhibition					
C_N_REL_CWP_MIN_DIAG_TH	1	0...FFH	0...255	1	[-]
Minimum cooling water pump speed for coolant thermostat diagnosis					
C_TAM_MIN_DIAG_TH	1	0...FEH	-48...142.5	0.75	[°C]
Minimum ambient air temperature for thermostat diagnosis					
C_MASS_VOL_CWP_MIN_DIAG_TH	1	0...FFFFH	0...65535	1	[-]
Minimum cooling water pump effected coolant mass volume for coolant thermostat diagnosis					
C_MASS_VOL_CWP_MAX	1	0...FFFFH	0...65535	1	[-]
C_MASS_VOL_CWP_MAX					
C_T_ECTPWM_INH_DIAG_TH	1	0...FFH	0...255	1	[s]
Max. time for ECTPWM exceeding threshold - for coolant thermostat diagnosis					
C_T_ECFPWM_INH_DIAG_TH	1	0...FFH	0...255	1	[s]
Max. time for ECFPWM exceeding threshold - for coolant thermostat diagnosis					
C_ECTPWM_INH_DIAG_TH	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Max. ECTPWM to deactivate coolant thermostat diagnosis					
C_ECFPWM_INH_DIAG_TH[NC_ECF_NR]	1	0...FFH	0...99.60937	0.390625	[%]
Max. ECFPWM to deactivate coolant thermostat diagnosis					
ID_N_REL_CWP_MAX_DIAG_TH	6	0...FFH	0...255	1	[-]
LDP_TCO_SUB_ID_N_REL_CWP_MAX	6	0...FE00H	-48...142.5	2.9297e-3	[°C]
Maximum cooling water pump speed for coolant thermostat diagnosis					
C_T_N_REL_CWP_MAX_DIAG_TH	1	0...FFH	0...255	1	[s]
Max. time for CWP speed exceeding threshold before coolant thermostat diagnosis inhibition					

26.35 Torque Loss - Electric Cooling Fan

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_LOSS_ECF	V/O	8000...7FFFH	-1024...1023.97	0.03125	Nm
Torque loss of the electronic controlled cooling fan(s)					

Input data:

ECFPWM[NC_ECF_NR]	N	LV_IGK	LV_ES
STATE_ALTER	LV_POW_CORD_INH	NC_ECF_NR	

FUNCTION DESCRIPTION:

General information:

TQ_LOSS_ECF represents the engine torque loss due to power consumption by the electric cooling fan (ECF).

Application conditions:

Activation: LV_IGK = 1 and LV_ES = 0

Deactivation: -

Initialisation: TQ_LOSS_ECF = 0 Nm at reset

Recurrence: 100 ms

*(see: "*ENTE scheduler*")*

Formula section:

Annotation:

At the MSV70 project only one electronic controlled cooling fan is available. The number of cooling fans is configured with "1" therefore. In this case the following condition is permitted:

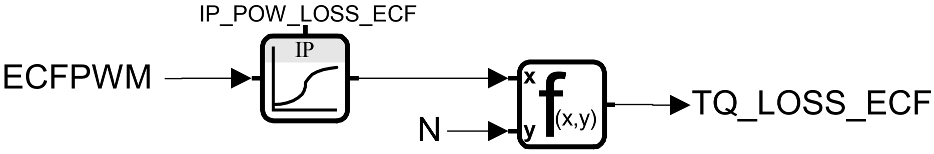
ECFPWM[NC_ECF_NR] = ECFPWM[1] = ECFPWM_1

Calculation of the torque loss:

```
If      ( LV_POW_CORD_INH = 0 and STATE_ALTER = 0 )     or

        N = 0

then    TQ_LOSS_ECF = 0 Nm

else    TQ_LOSS_ECF = ( IP_POW_LOSS_ ECF * 60 ) / ( 2 * 3.141828 * N )

endif
```

The change of TQ_LOSS_ECF is limited by limitation gradient C_LGRD_TQ_LOSS_ECF

Signal flow diagram:



Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_POW_LOSS_ECF	8	0...FF H	0...-5100	20	W
LDP_ECFPWM[1]__POW_LOSS_ECF	8	0...FF H	0...99.6	0.39	%
ECF power consumption					
C_LGRD_TQ_LOSS_ECF	1	0...7FFF H	0...1023.97	0.03125	Nm/100ms
ECF torque loss limitation gradient					

26.36 Customer adaptation module: ENTE

26.36.1 Outputs for BMW functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Tmot	O/V	EC78...7FFFH	-50...327.67	0.01	[°C]
Coolant temperature					
Tmot_start	O/V	EC78...7FFFH	-50...327.67	0.01	[°C]
Coolant temperature at engine start					
Tmot_start_dc	O/V	EC78...7FFFH	-50...327.67	0.01	[°C]
Coolant temperature at first start of driving cycle					
Tmot_abstell	O/V	EC78...7FFFH	-50...327.67	0.01	[°C]
Coolant temperature at engine stop					
Tka	O/V	EC78...7FFFH	-50...327.67	0.01	[°C]
Coolant temperature at radiator outlet					
Tsg	V/O	EC78...7FFFH	-50...327.67	0.01	[°C]
Temperatur Steuergerät					
L_pwmout	O/V	0...FFH	0...99.60937	0.390625	[%]
Electric fan control pulse width modulation					
Ba_ist_ewp	O/V	0...AH	0...10	1	[-]
internal operation state of the CWP					
Md_na_el	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Torque loss of the electronic controled cooling fan(s)					
Md_na_ewapu	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Torque losses electric water pump					
B_nkmtemp	O/V	0...1H	0...1	1	[-]
Boolean for low coolant temperature (from IHKA)					
B_elu600	O/V	0...1H	0...1	1	[-]
Learnt cooling fan variant					
B_lklps_ln	O/V	0...1H	0...1	1	[-]
Boolean for learnt variant ECRAS_UP					
B_lklps_ln1	O/V	0...1H	0...1	1	[-]
Boolean for learnt variant ECRAS_DOWN					
St_lklps	O/V	0...FFFFH	0...65535	1	[-]
Status ECRAS - System					

Input data:

TCO	TCO_ST	TCO_STOP	TECU
TCO_2		LV_FAN_VAR_AD	ECFPWM_ECF
TQ_LOSS_ECF	STATE_ENG	ECU_STATE	STATE_ECRAS_SYS
		TQ_LOSS_CWP_EL	TCO_ST_DC
STATE_CWP_INT	LV_REQ_TCO_L		
LV_VAR_ECRAS_UP	LV_VAR_ECRAS_DOWN		

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the output are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: all 0
except B_lklps_ln = true , Tmot_start=TCO; Tmot=TCO; Tmot_abstell =

TCO_STOP, Tsg = TECU
(this init has to be done after the init process of tco/tecu-acquisition)

Recurrence :	10ms:	Md_na_el, Md_na_ewapu
	100 ms:	Tmot, Tka, L_pwmout, B_elu600, Tsg, Tmot_start_dc
	200 ms:	B_lklps_ln, B_lklps_ln1, St_lklps Ba_ist_ewp, B_nkmtemp
	1000 ms:	*Tmot_start, Tmot_abstell*
Activation:	every engine state	

Formula section:

Remark: all formulas are valid in a **physical** meaning

Tmot = TCO

Tmot_start_dc = TCO_ST_DC

```
If      STATE_ENG  =  0
Then    Tmot_start  =  TCO
Else    Tmot_start  =  TCO_ST
Endif

If      ECU_STATE  =  4        "power latch"
Then    Tmot_abstell  =  TCO_STOP
Endif
```

Tka = TCO_2

L_pwmout = ECFPWM_ECF

Ba_ist_ewp = STATE_CWP_INT

Md_na_el = TQ_LOSS_ECF

Md_na_ewapu = TQ_LOSS_CWP_EL

B_nkmtemp = LV_REQ_TCO_L

B_elu600 = LV_FAN_VAR_AD

B_lklps_ln = LV_VAR_ECRAS_UP

B_lklps_ln1 = LV_VAR_ECRAS_DOWN

St_lklps = STATE_ECRAS_SYS

Tsg = TECU

26.36.2 Outputs for SV functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ECTPWM_CLC	V/O	0...FFFFH	0...99.996	0.001526	[%]
PWM for ECT from BMW, for MSx70 specific solution of ECT functionality					
N_PERC_ECF	V/O	0...FFH	0...99.60937	0.390625	[%]
percentage value of electronic fan speed setpoint					
N_REL_CWP_SP	V/O	0...FFH	0...255	1	[-]
Setpoint of EWAPU sent via bsd-component driver					
LV_LIH_COC_ON	V/O	0...1H	0...1	1	[-]
Limp home cooling circuit - engine ovrtemperature - request to transmission					
LV_TEMP_ENG_WARN_1	V/O	0...1H	0...1	1	[-]
1: activate yellow warning lamp for TCO overtemperature (Check-Control-Messages in BN vehicles)					
LV_TEMP_ENG_WARN_2	V/O	0...1H	0...1	1	[-]
1: activate red warning lamp for TCO overtemperature (Check-Control-Messages in BN vehicles)					
STATE_COC	O/V	0...FFH	0...255	1	[-]
State coolant circuit management					
LV_ECRAS	V/O	0...1H	0...1	1	[-]
Indication of air flaps being closed (1)/open (0)					
LV_ECRAS_UP	V/O	0...1H	0...1	1	[-]
SV label for B_LKLPS (from layer)					
LV_ECRAS_DOWN	O/V	0...1H	0...1	1	[-]
SV label for B_LKLPS1 (from layer)					
LV_TEMP_ENG_WARN_3	V/O	0...1H	0...1	1	[-]
1: activate gearshift up for TCO overtemperature (Check-Control-Messages in BN vehicles)					

Input data:

B_ntlkws	Tv_kft	Nelueft_wm	Newp_soll
Ba_wm_ist	B_tmmi_warn1	B_tmmi_warn2	B_lklps
LV_N_REL_CWP_SP_EXT_ADJ	N_REL_CWP_SP_EXT_ADJ	B_lklps1	B_tmmi_warn3

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the output are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: all 0 at reset

Recurrence : 100 ms: ECTPWM_CLC

200 ms: N_REL_CWP_SP, LV_LIH_COC_ON, LV_TEMP_ENG_WARN_1, LV_TEMP_ENG_WARN_2, LV_TEMP_ENG_WARN_3, STATE_COC, LV_ECRAS, N_PERC_ECF, LV_ECRAS_UP, LV_ECRAS_DOWN

Activation: every engine state

Formula section:

ECTPWM_CLC = Tv_kft
N_PERC_ECF = Nelueft_wm
LV_LIH_COC_ON = B_ntlkws

LV_TEMP_ENG_WARN_1 = B_tmmi_warn1
LV_TEMP_ENG_WARN_2 = B_tmmi_warn2
LV_TEMP_ENG_WARN_3 = B_tmmi_warn3
STATE_COC = Ba_wm_ist // up to now not used; planned to put on CAN

LV_ECRAS = 0

LV_ECRAS_UP = B_lklps

LV_ECRAS_DOWN = B_lklps1

```
IF      LV_N_REL_CWP_SP_EXT_ADJ = 1
THEN    N_REL_CWP_SP  = N_REL_CWP_SP_EXT_ADJ * 2,55 [1/%]
ELSE    N_REL_CWP_SP = Newp_soll
ENDIF
```

27 Error management

Table of Contents

Index

27.1 ERRM General

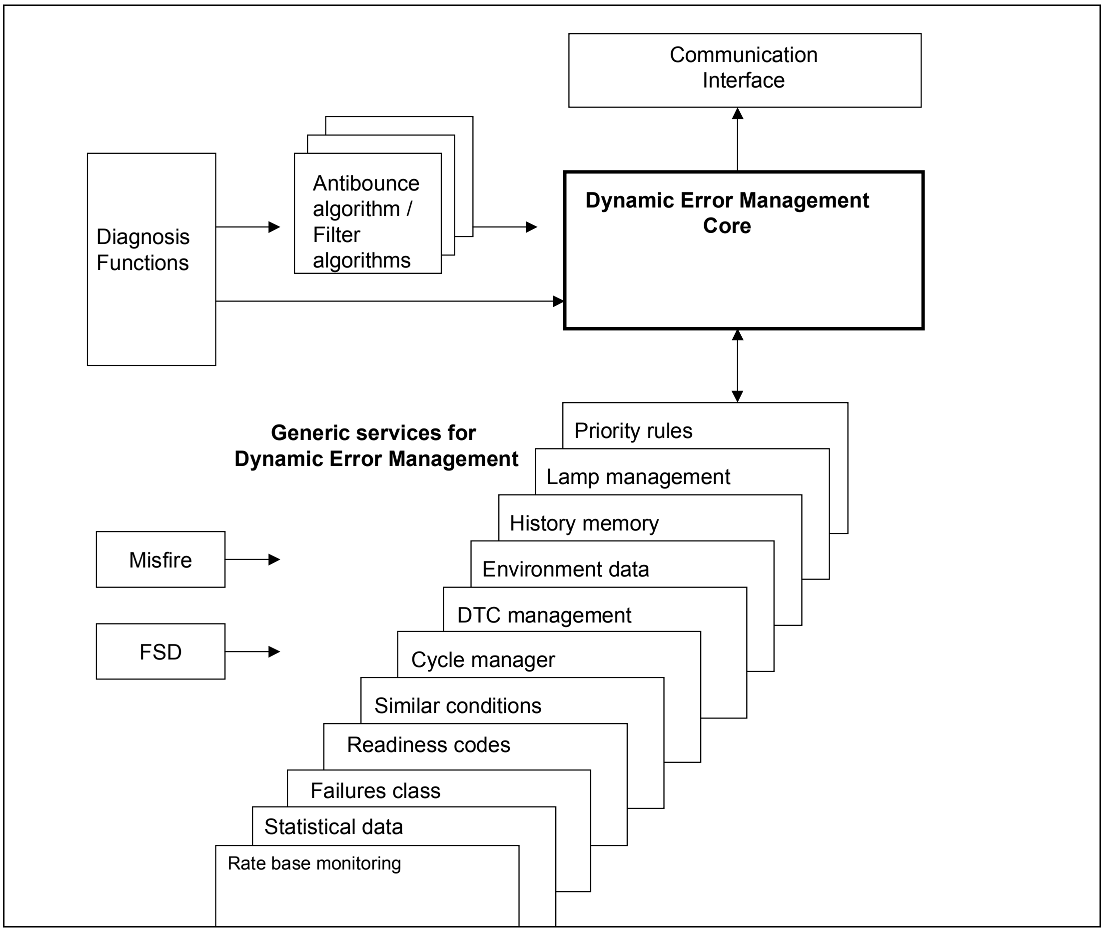
27.1.1 Target / Objectives of ERRM function

The purpose of the Error Management is :

- to collect, store and provide diagnostics information for repair and inspection programs
- to inform the driver via dashboard lamps from malfunctions which could affect emissions compliance with regulations (CARB for US market / EOBD for European market).

27.1.2 Architecture Overview

Principle of the dynamic error management is to receive diagnoses results (filtered or not by anti-bounce, muliti-condition, or statistical algorithms). These results are managed as diagnoses failures, according **CARB** and **EOBD** standards.



27.1.3 Description of the containing functions

- Dynamic Error Management core

The dynamic error management core manages different failures states thanks to the fault detection performed by the diagnoses : detected, present, temporary, pending, disappeared and confirmed. All failures are detected without any limitation on quantity. On the contrary, the number of present failures able to be stored into dynamic memory is limited. Management of this limitation is performed thanks to priority rules mecanism.

When a failure is stored into dynamic memory, the related environment data is stored, the lamp management may force lamp illumination (MIL, WAL,...), and DTCs management is done until failure is deleted. When deleted, failures are stored into history memory.

- History memory

History memory permits to trace historic of deleted failures. Some specific information are stored for each deleted failure. Only the more pertinent data linked to failure shall be stored.

- Debounce algorithm

The debounce algorithms is in charge of detected failures filtering according several available and predefined filters.

- Multi-conditions debounce algorithm

Debounce algorithm able to filter failures according a conditions per symptoms based algorithm.

- Cycle manager

Cycle manager computes driving cycle and warm-up cycle according to engine states and engine coolant temperature. Cycle manager permits to evaluate failure states.

- Failures class

Each failure can be configured to obtain a specific behaviour : emission relevant or not, can illuminate MIL or not ... This module contain to all the predefined behaviours each failure can be associated with. It permits to symplify greatly tuning of diagnosis by decreasing quantity of calibrations.

- DTC management

DTC management permits to catch some data when a failure becomes present and to generate a code identifier, called DTC. It permits to identify default of the function.

- Communication interface

This module allow to access failure memory data related to error management such as freeze frame, stored DTCs, readiness codes.

- Priority rules

This module manages dynamic memory size limitation. It defines criteria, called priority rules, to store or not a new failure into this dynamic memory.

- Environmental data

This module describes the structure of the freeze frame and its management : storage and delete.

- Similar conditions

Similar conditions are additional conditions based on engine and load status to erase a failure in memory (for Misfire and Fuel system failure only).

- Lamp management

Lamp management module can manage illumination of several warning lamps (MIL and other warning lamps) including pre-drive check.

- Readiness codes

Readiness codes allow to know if a diagnosis has been performed or not.

- Rate-based monitoring

Rate-base monitoring functionality : monitoring performances under real world conditions. Performs statistics calculation on diagnoses.

- Statistical data

Statistical data allows monitoring and storage of statistical data on error management data flow.

27.2 ERRM Configuration Data (MSD80 – 6 Cylinder)

Input data:

NC_ABC_CONF_FCT_DIAG_XX	NC_NR_DTC_FMT	NC_ERR_DTC_CONF
NC_NR_ENVD_CUS_SET_CMN	NC_MIL_CHK_TYP	NC_NR_ENVD_CUS_CMN
NC_NR_ENVD_PREV	NC_NR_ENVD_CUS_SET_SPC	NC_NR_ENVD_OBD
NC_NR_FRF_SET	NC_NR_ERR_DYN	NC_NR_ERR_HIS
NC_NR_WIN_SCDN	NC_NR_HIS	NC_NR_SYM_XX
NC_ENVD_CUS_CMN_UPD	NC_ERR_DTC_REQ_CUS	NC_ERR_DTC_REQ_OBD
NLC_ABC_INI_DC_END_DIAG	NC_NR_LAMP	NC_NR_DIAG_RBM
NC_ERR_PRI_H	NLC_INC_ERR_PRI	NLC_OLD_ERR_PRI
NLC_ERR_DET_UPD	NLC_ABC_INI_DC_END_DIAG	NLC_ENA_SCDN
NC_ERR_PRI_H	NLC_INC_ERR_PRI	NLC_OLD_ERR_PRI
NLC_OBD_FRF_PND	NLC_BENCH_MODE	NC_NR_ERR_INTM

General information:

The following chapters describe the general rules for configuration data determination.

27.2.1 Global configuration data

Data	Value	
	Typical values are listed below	
NLC_CAN_LAMP_ACT	**1**	Warning lamps interface type 0: warning lamps hardwired to ECU 1: warning lamps status sended to CAN inter-system
NLC_ERR_DET_UPD	**0**	Configuration for all diagnoses : 00h: no sporadic error detection 01h: sporadic error detection
NLC_ENA_SCDN	**1**	Similar conditions present or not : 0 : Similar conditions disable, acceptable for Europe applications. 1 : Similar conditions enable, for US applications
NLC_ENA_SCDN_NEW	**1**	Old or new Similar conditions functionality used : 0 : old SCDN functionality used 1 : new SCDN functionality used
NLC_OBD_RBM_ENA	**1**	Rate-based monitoring functionality present or not : 0 : Rate-based monitoring disabled 1 : Rate-based monitoring enabled

27.2.2 Local configuration data, project specific for MSD80

Here are listed the configuration data, which are specific for MSD80

Data	Value	
	Typical values are listed below	
NC_NR_DIAG_RBM	**52**	Number of diagnosis using the RBM statistics
NC_NR_WIN_SCDN	**10**	Quantity of diagnoses using similar conditions

27.2.3 Local configuration data

Here are listed the configuration data, which are used only in the ERRM aggregate.

Data	Value	
	Typical values are listed below	
NC_NR_ERR_DYN	**15**	Dynamic memory size (quantity of failures)
NLC_OLD_ERR_PRI	**0**	Failure chronological priority order **0: priority to new failures** 1: priority to old failures
NLC_INC_ERR_PRI	**0**	Set this bit to 1 to enable increased failure priority **Set this bit to 0 to disable increased failure priority**
NC_ERR_PRI_H	**6**	Failure priority level linked to NLC_INC_ERR_PRI
NC_NR_ENVD_OBD	**38**	Size of obd environmental data, Quantity of data byte
NC_NR_ENVD_CUS_CMN	**1**	Size of common environmental data, Quantity of data byte
NC_NR_ENVD_CUS_SET_CMN	**3**	Size of environmental data set, Quantity of data byte
NC_NR_ENVD_CUS_SET_SPC	**4**	Size of specific environmental data set Quantity of data byte
NC_NR_FRF_SET	**3**	Quantity of environmental data sets. Quantity of sets
NC_NR_ENVD_PREV	**1**	00h: Prestored freeze frame functionality is not used 01h...FFh: Quantity of XX failures instances using the prestored freeze frame
NC_ENVD_CUS_CMN_UPD	**1**	Way to update environmental data 0: no update at failure status change **1: update at failure change status (same as OBD)**
NC_NR_ERR_HIS	**10**	Size of history memory (quantity of failures)
NC_NR_HIS	**9**	Size of data stored for a single failure in history memory (quantity of data bytes)
NC_ERR_DTC_REQ_OBD	**0**	Type of OBD/CARB DTCs returned by API **0 : detailed for each symptom** 1 : limited to the global failure
NC_ERR_DTC_REQ_CUS	**1**	Type of customer DTCs returned by API 0 : detailed for each symptom **1 : limited to the global failure**
NC_ERR_DTC_CONF	**3**	Type of DTCs displayed through ERR_DTC_IDX 0 : J2012 DTC detailed for each symptom 1 : J2012 DTC limited to the global failure 2 : customer DTC detailed for each symptom **3 : customer DTC limited to the global failure**
NC_LDP_1_DTC_TABLE_SIZE	**1**	Table size definition for ID_ERR_DTC_XX. Must be set to NC_NR_DTC_FMT+1.
NC_LDP_2_DTC_TABLE_SIZE	**6**	Table size definition for ID_ERR_DTC_XX. Must be set to 6 - NC_NR_DTC_FMT.
NC_LDP_2_DTC_MIS_TABLE_SIZE	**8**	Table size definition for ID_ERR_DTC_MIS. Must be to NC_CYL_NR+2.
NC_NR_DTC_FMT	**0**	Defines 1 or 2 different DTC identifiers per symptom. Defintion with 1 DTC permits a DTC use in common for obd and customer. **0 : a single DTC per symptom (a single one for obd and cutomer)** 1 : 2 DTCs defined per symptom (one for obd, one for cutomer)

Data	Value	
	Typical values are listed below	
NC_MIL_CHK_TYP	**03h**	Pre-drive check strategy on stalling event : Bit0 = 0 : Disable pre-drive check after “stalling event/engine start” **Bit0 = 1 : Enable pre-drive check after “stalling event/engine start”** Display readiness status during pre-drive check (MIL blinking) : Bit1 = 0 : disable readiness status display during pre-drive check **Bit1 = 1 : enable readiness status display during pre-drive check**
NLC_OBD_FRF_PND	**0**	Defines the EOBD/CARB freeze frame strategy as follow : **0 : EOBD/CARB freeze frame data of mode 02h will be returned after confirmed failure detection** 1 : EOBD/CARB freeze frame data of mode 02h will be returned : - after a confirmed emission relevant failure, if confirmed in memory - after a pending emission relevant detection, if not confirmed in memory
NC_NR_LAMP	**0**	Number of warning lamps (not used because NLC_CAN_LAMP_ACT = CAN)

Data	Value	
	Typical values are listed below	
Filters		
NC_NR_SYM_XX See ATIC 39 diagnosis	**07h**	Configuration to define available symptom for each XX diagnosis : This constant is used in the multicondition filtering algorithm. 03h: SYM_0 & SYM_1 available **07h: SYM_0...SYM_2 available** *Please note that this configuration shall be mentioned in the “Table of failure” module, for each diagnosis*
NC_ABC_CONF_FCT_DIAG_XX	****)**	Configuration depends on each diagnosis XX behaviour : 00h: Standard configuration STD 03h: Standard configuration STD_INI with initialisation(only this config is available for multicondition) 0Bh: Memory configuration MEM 0Eh: Memory configuration MEM_INI 10h: Decrement calibration configuration DEC_CAL 13h: Statistical configuration STC *****) See initialization of each diagnosis***
NLC_ABC_INI_DC_END_DIAG	**01h**	Configuration for all diagnoses : 00h: Diagnosis inititialisation is managed at LV_IGK 0 → 1 transition **01h: End of diagnosis initialisation is managed at LV_DC 1 → 0 (only this configuration is available for multicondition filtering)**
NC_NR_ERR_INTM	**1**	Size of the sporadic error memory. When NLC_ERR_DET_UPD = 00h, then NC_NR_ERR_INTM must be set to 1.
NLC_BENCH_MODE	**1**	Bench – Mode is enabled

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_CAN_LAMP_ACT	1	0...1H	0...1	1	[-]
Selection of the strategy to drive warning lamps (0: wire, 1: CAN)					
NLC_ERR_DET_UPD	1	0...1H	0...1	1	[-]
Activation of sporadic errors storage and visualisation					
NLC_ENA_SCDN	1	0...1H	0...1	1	[-]
Boolean to indicate if similar conditions functionality is present (1) or not (0)					
NLC_ENA_SCDN_NEW	1	0...1H	0...1	1	[-]
Activation of old/new similar conditions functionality (0:old strategy, 1:new strategy)					
NLC_OBD_RBM_ENA	1	0...1H	0...1	1	[-]
Activation of rate-based monitoring functionality (0: disabled, 1: enabled)					

27.3 AGGR ERRM adaptation (MSD80)

27.3.1 General interface description

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_GEN	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on Generator					
LV_CDN_DIAG_GEN	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on generator diagnosis					
LV_END_DIAG_GEN	V	0...1H	0...1	1	[-]
End of diagnosis generator diagnosis					
ERR_SYM_GEN	V	0H	NO_SYM	1	[-]
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Detected error generator diagnosis					
LV_ERR_QOIL_SENS	V/O	0...1H	0...1	1	[-]
Error Flag for quality Oil Sensor					
LV_CDN_DIAG_QOIL_SENS	V	0...1H	0...1	1	[-]
oil quality Diagnosis Condition					
LV_END_DIAG_QOIL_SENS	V	0...1H	0...1	1	[-]
End of quality Oil Diagnosis					
ERR_SYM_QOIL_SENS	V	0H	NO_SYM	1	[-]
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Detected quality Oil Sensor Symptom					
LV_ERR_BAT_SENS	V/O	0...1H	0...1	1	[-]
communication error BAT_SENS					
LV_ERR_BAT_SENS_IT_EL	V/O	0...1H	0...1	1	[-]
internal error (electrical) BAT_SENS					
LV_ERR_BAT_SENS_IT	V/O	0...1H	0...1	1	[-]
internal error (component) BAT_SENS					
LV_CDN_DIAG_BAT_SENS	V	0...1H	0...1	1	[-]
condition for BAT_SENS-diagnosis fulfilled					
LV_CDN_DIAG_BAT_SENS_IT_EL	V	0...1H	0...1	1	[-]
condition for BAT_SENS-diagnosis (communication) fulfilled					
LV_CDN_DIAG_BAT_SENS_IT	V	0...1H	0...1	1	[-]
condition for BAT_SENS-diagnosis (electrical) fulfilled					
LV_END_DIAG_BAT_SENS	V	0...1H	0...1	1	[-]
End of Diagnosis BAT_SENS communication					
LV_END_DIAG_BAT_SENS_IT_EL	V	0...1H	0...1	1	[-]
End of Diagnosis BAT_SENS electrical					
LV_END_DIAG_BAT_SENS_IT	V	0...1H	0...1	1	[-]
End of Diagnosis BAT_SENS component					
ERR_SYM_BAT_SENS	V	0H	NO_SYM	1	[-]
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Detected symptom BAT_SENS (component) communication					
ERR_SYM_BAT_SENS_IT_EL	V	0H	NO_SYM	1	[-]
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Detected symptom BAT_SENS electrical					

ERR_SYM_BAT_SENS_IT	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BAT_SENS component					
LV_ERR_ALTER_BN	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on PM board network					
LV_ERR_ALTER_BN_BAT	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on PM battery					
LV_CDN_DIAG_ALTER_BN	V	0...1H	0...1	1	[-]
condition PM board network diagnosis					
LV_CDN_DIAG_ALTER_BN_BAT	V	0...1H	0...1	1	[-]
condition PM battery diagnosis					
LV_END_DIAG_ALTER_BN	V	0...1H	0...1	1	[-]
end of PM board network diagnosis					
LV_END_DIAG_ALTER_BN_BAT	V	0...1H	0...1	1	[-]
end of PM battery diagnosis					
ERR_SYM_ALTER_BN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error PM board network diagnosis					
ERR_SYM_ALTER_BN_BAT	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error PM battery diagnosis					
LV_CDN_DIAG_LOAD_TPS_PLAUS	V	0...1H	0...1	1	[-]
Diagnosis Condition LOAD_TPS_PLAUS error					
LV_ERR_LOAD_TPS_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean that indicates inconsistencies between actual load and throttle position					
ERR_SYM_LOAD_TPS_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom load/TPS plausibility check					
LV_END_DIAG_LOAD_TPS_PLAUS	V	0...1H	0...1	1	[-]
End of LOAD_TPS_PLAUS error sensor diagnosis					
LV_INH_DIAG_RBM_LOAD_TPS_PLAUS	V	0...1H	0...1	1	[-]
Flag set if LOAD_TPS_PLAUS diagnosis is inhibitied due to present OBD error					
LV_CDN_RBM_LOAD_TPS_PLAUS	V	0...1H	0...1	1	[-]
Flag set if LOAD_TPS_PLAUS diagnosis condition RBM is fulfilled					
LV_END_RBM_LOAD_TPS_PLAUS	V	0...1H	0...1	1	[-]
Flag set if LOAD_TPS_PLAUS diagnosis cycle RBM is finished					
LV_CDN_DIAG_VIM_PLAUS	V	0...1H	0...1	1	[-]
Diagnosis Condition VIM_PLAUS error					
LV_ERR_VIM_PLAUS	V/O	0...1H	0...1	1	[-]
Flag to indicate VIM failure global error					
ERR_SYM_VIM_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected VIM_PLAUS error symptom					
LV_END_DIAG_VIM_PLAUS	V	0...1H	0...1	1	[-]
End of VIM_PLAUS error sensor diagnosis					
LV_CDN_DIAG_MAP_DIP_PLAUS	V	0...1H	0...1	1	[-]
Diagnosis Condition MAP_DIP_PLAUS error					
LV_ERR_MAP_DIP_PLAUS	-	0...1H	0...1	1	[-]

Error Flag for MAP_DIP_PLAUS error					
ERR_SYM_MAP_DIP_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected MAP_DIP_PLAUS error symptom					
LV_END_DIAG_MAP_DIP_PLAUS	V	0...1H	0...1	1	[-]
End of MAP_DIP_PLAUS error sensor diagnosis					
LV_INH_DIAG_RBM_MAP_DIP_PLAUS	V	0...1H	0...1	1	[-]
Flag set if MAP_DIP_PLAUS diagnosis is inhibitied due to present OBD error					
LV_CDN_RBM_MAP_DIP_PLAUS	V	0...1H	0...1	1	[-]
Flag set if MAP_DIP_PLAUS diagnosis condition RBM is fulfilled					
LV_END_RBM_MAP_DIP_PLAUS	V	0...1H	0...1	1	[-]
Flag set if MAP_DIP_PLAUS diagnosis cycle RBM is finished					
LV_CDN_DIAG_ALTER_BN_RGN	V	0...1H	0...1	1	[-]
Diagnosis Condition ALTER_BN_RGN error					
LV_ERR_ALTER_BN_RGN	V/O	0...1H	0...1	1	[-]
Error Flag for ALTER_BN_RGN error					
ERR_SYM_ALTER_BN_RGN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected ALTER_BN_RGN error symptom					
LV_END_DIAG_ALTER_BN_RGN	V	0...1H	0...1	1	[-]
End of ALTER_BN_RGN error sensor diagnosis					
LV_ERR_TOIL_LEVEL	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on TOENS sensor					
LV_END_DIAG_TOIL_LEVEL	V	0...1H	0...1	1	[-]
End of diagnosis TOENS sensor					
LV_CDN_DIAG_TOIL_LEVEL	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on TOENS sensor					
ERR_SYM_TOIL_LEVEL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error TOENS sensor					
LV_ERR_AMP_PLAUS_CUS	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on AMP plausibility diagnosis					
LV_END_DIAG_AMP_PLAUS_CUS	V	0...1H	0...1	1	[-]
End of diagnosis AMP plausibility diagnosis					
LV_CDN_DIAG_AMP_PLAUS_CUS	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on AMP plausibility diagnosis					
ERR_SYM_AMP_PLAUS_CUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error AMP plausibility diagnosis					
LV_ERR_TCHA_PRS_LOW	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on charger pressure to low diagnosis					
LV_END_DIAG_TCHA_PRS_LOW	V	0...1H	0...1	1	[-]
End of diagnosis charger pressure to low diagnosis					
LV_CDN_DIAG_TCHA_PRS_LOW	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on charger pressure to low diagnosis					
ERR_SYM_TCHA_PRS_LOW	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error charger pressure to low diagnosis					

LV_ERR_TCHA_LEAK	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on charger leakage diagnosis					
LV_END_DIAG_TCHA_LEAK	V	0...1H	0...1	1	[-]
End of diagnosis charger leakage diagnosis					
LV_CDN_DIAG_TCHA_LEAK	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on charger leakage diagnosis					
ERR_SYM_TCHA_LEAK	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error charger leakage diagnosis					
LV_ERR_TCHA_PRS_DIF	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on charger pressure difference diagnosis					
LV_END_DIAG_TCHA_PRS_DIF	V	0...1H	0...1	1	[-]
End of diagnosis charger pressure difference diagnosis					
LV_CDN_DIAG_TCHA_PRS_DIF	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on charger pressure difference diagnosis					
ERR_SYM_TCHA_PRS_DIF	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error charger pressure difference diagnosis					
LV_ERR_TCHA_PRS_HIGH	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on charger pressure to high diagnosis					
LV_END_DIAG_TCHA_PRS_HIGH	V	0...1H	0...1	1	[-]
End of diagnosis charger pressure to high diagnosis					
LV_CDN_DIAG_TCHA_PRS_HIGH	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on charger pressure to high diagnosis					
ERR_SYM_TCHA_PRS_HIGH	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error charger pressure to high diagnosis					
LV_ERR_POIL_CTL_STAT	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on oil pressure contoller (static)					
LV_END_DIAG_POIL_CTL_STAT	V	0...1H	0...1	1	[-]
End of diagnosis oil pressure contoller (static)					
LV_CDN_DIAG_POIL_CTL_STAT	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on oil pressure contoller (static)					
ERR_SYM_POIL_CTL_STAT	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error oil pressure contoller (static)					
LV_ERR_POIL_CTL_DYN	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on oil pressure contoller (dynamic)					
LV_END_DIAG_POIL_CTL_DYN	V	0...1H	0...1	1	[-]
End of diagnosis oil pressure contoller (dynamic)					
LV_CDN_DIAG_POIL_CTL_DYN	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on oil pressure contoller (dynamic)					
ERR_SYM_POIL_CTL_DYN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error oil pressure contoller (static)					
LV_ERR_PUT_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on PUT plausibility diagnosis					
LV_END_DIAG_PUT_PLAUS	V	0...1H	0...1	1	[-]

End of diagnosis PUT plausibility diagnosis					
LV_CDN_DIAG_PUT_PLAUS	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on PUT plausibility diagnosis					
ERR_SYM_PUT_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error PUT plausibility diagnosis					
LV_ERR_MAP_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for detected error of Manifold pressure plausibility diagnosis					
LV_END_DIAG_MAP_PLAUS	V	0...1H	0...1	1	[-]
End of diagnosis flag for Manifold pressure plausibility diagnosis					
LV_CDN_DIAG_MAP_PLAUS	V	0...1H	0...1	1	[-]
Status of diagnosis flag for Manifold pressure plausibility					
ERR_SYM_MAP_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom Manifold pressure plausibility diagnosis					
LV_CDN_DIAG_TQ_CST	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on TQ_CST sensor					
LV_ERR_TQ_CST	V/O	0...1H	0...1	1	[-]
Error Flag for TQ_CST error					
ERR_SYM_TQ_CST	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected TQ_CST error symptom					
LV_END_DIAG_TQ_CST	V	0...1H	0...1	1	[-]
End of TQ_CST error diagnosis					
LV_INH_DIAG_RBM_TQ_CST	V	0...1H	0...1	1	[-]
Flag set if TQ_CST diagnosis is inhibitied due to present OBD error					
LV_END_RBM_TQ_CST	V	0...1H	0...1	1	[-]
Flag set if TQ_CST diagnosis cycle RBM is finished					
LV_CDN_RBM_TQ_CST	V	0...1H	0...1	1	[-]
Flag set if TQ_CST diagnosis condition RBM is fulfilled					
LV_ERR_POIL_CTL_MEC	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on Oilpressure valve diagnosis					
LV_END_DIAG_POIL_CTL_MEC	V	0...1H	0...1	1	[-]
End of diagnosis Oilpressure valve diagnosis					
LV_CDN_DIAG_POIL_CTL_MEC	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on Oilpressure valve diagnosis					
ERR_SYM_POIL_CTL_MEC	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error Oilpressure valve diagnosis					
LV_ERR_POIL_SENS_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on Oilpressure sensor diagnosis					
LV_END_DIAG_POIL_SENS_PLAUS	V	0...1H	0...1	1	[-]
End of diagnosis Oilpressure sensor diagnosis					
LV_CDN_DIAG_POIL_SENS_PLAUS	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on Oilpressure sensor diagnosis					
ERR_SYM_POIL_SENS_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error Oilpressure sensor diagnosis					

LV_ERR_NEUT_PSN_GB_LRN	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on Neutral gear learning diagnosis					
LV_END_DIAG_NEUT_PSN_GB_LRN	V	0...1H	0...1	1	[-]
End of diagnosis Neutral gear learning diagnosis					
LV_CDN_DIAG_NEUT_PSN_GB_LRN	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on Neutral gear learning diagnosis					
ERR_SYM_NEUT_PSN_GB_LRN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error Neutral gear learning diagnosis					
LV_ERR_POIL_PUMP	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on Oilpump diagnosis					
LV_END_DIAG_POIL_PUMP	V	0...1H	0...1	1	[-]
End of diagnosis Oilpump diagnosis					
LV_CDN_DIAG_POIL_PUMP	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on Oilpump diagnosis					
ERR_SYM_POIL_PUMP	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error Oilpump diagnosis					
LV_ERR_POIL_SYS	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on Oilsystem diagnosis					
LV_END_DIAG_POIL_SYS	V	0...1H	0...1	1	[-]
End of diagnosis Oilsystem diagnosis					
LV_CDN_DIAG_POIL_SYS	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on Oilsystem diagnosis					
ERR_SYM_POIL_SYS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error Oilsystem diagnosis					
LV_ERR_GEN_EL	V/O	0...1H	0...1	1	[-]
Flag for electrical generator error					
LV_CDN_DIAG_GEN_EL	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present for electrical generator diagnosis					
LV_END_DIAG_GEN_EL	V	0...1H	0...1	1	[-]
End of diagnosis flag for electrical generator diagnosis					
ERR_SYM_GEN_EL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom in electrical generator diagnosis					
LV_ERR_GEN_MEC	V/O	0...1H	0...1	1	[-]
Flag for mechanical generator error					
LV_CDN_DIAG_GEN_MEC	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present for mechanical generator diagnosis					
LV_END_DIAG_GEN_MEC	V	0...1H	0...1	1	[-]
End of diagnosis flag for mechanical generator diagnosis					
ERR_SYM_GEN_MEC	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom in mechanical generator diagnosis					
LV_ERR_GEN_H_TEMP	V/O	0...1H	0...1	1	[-]
Flag for generator high temperature error					
LV_CDN_DIAG_GEN_H_TEMP	V	0...1H	0...1	1	[-]

Boolean for diagnosis condition present for generator high temperature diagnosis					
LV_END_DIAG_GEN_H_TEMP	V	0...1H	0...1	1	[-]
End of diagnosis flag for generator high temperature diagnosis					
ERR_SYM_GEN_H_TEMP	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom in generator high temperature diagnosis					
LV_ERR_GEN_TYP_NOT_PLAUS	V/O	0...1H	0...1	1	[-]
Flag for generator type not plausible error					
LV_CDN_DIAG_GEN_TYP_NOT_PLAUS	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present for generator type plausibility diagnosis					
LV_END_DIAG_GEN_TYP_NOT_PLAUS	V	0...1H	0...1	1	[-]
End of diagnosis flag for generator type plausibility diagnosis					
ERR_SYM_GEN_TYP_NOT_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom in generator type plausibility diagnosis					
LV_CDN_DIAG_GEN_MSG_LOST	V	0...1H	0...1	1	[-]
Diagnosis condition GEN_MSG_LOST					
LV_END_DIAG_GEN_MSG_LOST	V	0...1H	0...1	1	[-]
End of diagnosis GEN_MSG_LOST					
ERR_SYM_GEN_MSG_LOST	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom GEN_MSG_LOST					
LV_ERR_GEN_MSG_LOST	V/O	0...1H	0...1	1	[-]
Present error GEN_MSG_LOST					
LV_ERR_GEN_CLC_V_NOT_PLAUS	V/O	0...1H	0...1	1	[-]
Flag for calculated generator voltage plausibility error					
LV_CDN_DIAG_GEN_CLC_V_NOT_PLAUS	V	0...1H	0...1	1	[-]
Diagnosis condition GEN_CLC_V_NOT_PLAUS					
LV_END_DIAG_GEN_CLC_V_NOT_PLAUS	V	0...1H	0...1	1	[-]
End of diagnosis GEN_CLC_V_NOT_PLAUS					
ERR_SYM_GEN_CLC_V_NOT_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom CLC_V_NOT_PLAUS					
LV_ERR_GEN_H_TEMP_CLC	V/O	0...1H	0...1	1	[-]
Flag for calculated generator high temperature error					
LV_CDN_DIAG_GEN_H_TEMP_CLC	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present for calculated generator high temperature diagnosis					
LV_END_DIAG_GEN_H_TEMP_CLC	V	0...1H	0...1	1	[-]
End of diagnosis flag for calculated generator high temperature diagnosis					
ERR_SYM_GEN_H_TEMP_CLC	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom in calculated generator high temperature diagnosis					
LV_ERR_GEN_CTL_NOT_PLAUS	V/O	0...1H	0...1	1	[-]
Flag for calculated generator controller plausibility error					
LV_CDN_DIAG_GEN_CTL_NOT_PLAUS	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present for generator controller plausibility diagnosis					
LV_END_DIAG_GEN_CTL_NOT_PLAUS	V	0...1H	0...1	1	[-]
End of diagnosis flag for generator controller plausibility diagnosis					

ERR_SYM_GEN_CTL_NOT_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom in generator controller plausibility diagnosis					
LV_ERR_EGRV_PSN_PLAUS	O/V	0...1H	0...1	1	[-]
Error flag for position of exhaust gas recirculation valve plausibility					
LV_CDN_DIAG_EGRV_PSN_PLAUS	V	0...1H	0...1	1	[-]
Diagnosis condtion for diagnosis instance of exhaust gas recirculation valve position plausibility					
LV_END_DIAG_EGRV_PSN_PLAUS	V	0...1H	0...1	1	[-]
Diagnosis result available for diagnosis instance exhaust gas recirculation valve position plausibility					
ERR_SYM_EGRV_PSN_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error sympthom for exhaust gas recirculation valve position plausibility					
LV_ERR_TCHA_PRS_CTL	O/V	0...1H	0...1	1	[-]
Boolean for error currently present on charger pressure controller diagnosis					
LV_CDN_DIAG_TCHA_PRS_CTL	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on charger pressure controller diagnosis					
LV_END_DIAG_TCHA_PRS_CTL	V	0...1H	0...1	1	[-]
End of diagnosis charger pressure controller diagnosis					
ERR_SYM_TCHA_PRS_CTL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error charger pressure controller diagnosis					
LV_ERR_MEC_IVVT_IN	O/V	0...1H	0...1	1	[-]
Error Flag for functional error on the inlet CAM adjustment					
LV_ERR_MEC_IVVT_EX	O/V	0...1H	0...1	1	[-]
Error Flag for functional error on the exhaust CAM adjustment					
LV_CDN_DIAG_MEC_IVVT_IN	V	0...1H	0...1	1	[-]
Diagnosis Condition inlet CAM adjustment					
LV_CDN_DIAG_MEC_IVVT_EX	V	0...1H	0...1	1	[-]
Diagnosis Condition exhaust CAM adjustment					
ERR_SYM_MEC_IVVT_IN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom on the inlet CAM adjustment					
ERR_SYM_MEC_IVVT_EX	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom on the exhaust CAM adjustment					
LV_END_DIAG_MEC_IVVT_IN	V	0...1H	0...1	1	[-]
End of inlet CAM adjustment Diagnosis					
LV_END_DIAG_MEC_IVVT_EX	V	0...1H	0...1	1	[-]
End of exhaust CAM adjustment Diagnosis					
LV_INH_DIAG_RBM_MEC_IVVT_IN	V	0...1H	0...1	1	[-]
Flag set if IVVT diagnosis is inhibitied due to present OBD error					
LV_INH_DIAG_RBM_MEC_IVVT_EX	V	0...1H	0...1	1	[-]
Flag set if IVVT diagnosis is inhibitied due to present OBD error					
LV_END_RBM_MEC_IVVT_IN	V	0...1H	0...1	1	[-]
Flag set if IVVT diagnosis cycle RBM is finished					
LV_END_RBM_MEC_IVVT_EX	V	0...1H	0...1	1	[-]
Flag set if IVVT diagnosis cycle RBM is finished					
LV_CDN_RBM_MEC_IVVT_IN	V	0...1H	0...1	1	[-]

Flag set if IVVT diagnosis condition RBM is fulfilled					
LV_CDN_RBM_MEC_IVVT_EX	V	0...1H	0...1	1	[-]
Flag set if IVVT diagnosis condition RBM is fulfilled					
LV_ERR_N_MAX_HOMS	O/V	0...1H	0...1	1	[-]
Boolean for error engine overspeed for changing the combustion mode					
LV_CDN_DIAG_N_MAX_HOMS	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on engine overspeed for changing the combustion mode diagnosis					
LV_END_DIAG_N_MAX_HOMS	V	0...1H	0...1	1	[-]
End of diagnosis engine overspeed for changing the combustion mode diagnosis					
ERR_SYM_N_MAX_HOMS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error engine overspeed for changing the combustion mode diagnosis					
LV_ERR_MAP_TPS_PLAUS	O/V	0...1H	0...1	1	[-]
Boolean for error MAP ratio plausibility diagnosis					
LV_CDN_DIAG_MAP_TPS_PLAUS	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on MAP ratio plausibility diagnosis					
LV_END_DIAG_MAP_TPS_PLAUS	V	0...1H	0...1	1	[-]
End of diagnosis MAP ratio plausibility diagnosis					
ERR_SYM_MAP_TPS_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error MAP ratio plausibility diagnosis					
LV_ERR_TCHA_SYS_1	O/V	0...1H	0...1	1	[-]
Boolean for error currently present on charger system diagnosis 1					
LV_END_DIAG_TCHA_SYS_1	V	0...1H	0...1	1	[-]
End of diagnosis charger system diagnosis 1					
LV_CDN_DIAG_TCHA_SYS_1	V	0...1H	0...1	1	[-]
Boolean for diagnosis condition present on charger system diagnosis 1					
ERR_SYM_TCHA_SYS_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error charger system diagnosis 1					
STATE_RBM_MAP_DIP_PLAUS	O/V	0...7H	0...7	1	[-]
Interface of MAP_DIP_PLAUS monitor with the Rate-Based Monitoring statistics					
STATE_RBM_LOAD_TPS_PLAUS	O/V	0...7H	0...7	1	[-]
Interface of LOAD_TPS_PLAUS monitor with the Rate-Based Monitoring statistics					
STATE_RBM_TQ_CST	O/V	0...7H	0...7	1	[-]
Interface of TQ_CST monitor with the Rate-Based Monitoring statistics					
STATE_RBM_MEC_IVVT_IN	O/V	0...7H	0...7	1	[-]
Interface of MEC_IVVT_IN monitor with the Rate-Based Monitoring statistics					
STATE_RBM_MEC_IVVT_EX	O/V	0...7H	0...7	1	[-]
Interface of MEC_IVVT_EX monitor with the Rate-Based Monitoring statistics					
LV_EOL_OBD_DC	O/V	0...1H	0...1	1	[-]
Flag for EOL activation in driving cycle					
LV_ERR_TOIL	O/V	0...1H	0...1	1	[-]
Flag for oil temperature error (set to "0" because TOIL_MDL is used if no value from sensor or CAN is available)					
LV_ERR_IVVT	O/V	0...1H	0...1	1	[-]
Boolean for error currently present IVVT system					
LV_ERR_MAF_LAMB_MAX	O/V	0...1H	0...1	1	[-]
Error flag for maximum mass air flow calculated from lambda					
LV_CDN_DIAG_MAF_LAMB_MAX	V	0...1H	0...1	1	[-]
Diagnosis condition for maximum mass air flow calculated from lambda					
LV_END_DIAG_MAF_LAMB_MAX	V	0...1H	0...1	1	[-]

Diagnosis result available for daignosis instance of maximum mass air flow calculated from lambda					
ERR_SYM_MAF_LAMB_MAX	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error sympthom for maximum mass air flow calculated from lambda					
LV_ERR_ER_STRAT	O/V	0...1H	0...1	1	[-]
Error flag for engine roughness improvement at stratified mode					
LV_CDN_DIAG_ER_STRAT	V	0...1H	0...1	1	[-]
Diagnosis condition for engine roughness improvement at stratified mode					
LV_END_DIAG_ER_STRAT	V	0...1H	0...1	1	[-]
Diagnosis result available for daignosis instance of engine roughness improvement at stratified mode					
ERR_SYM_ER_STRAT	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error sympthom for engine roughness improvement at stratified mode					
LV_ERR_ER_STRAT_WUP	O/V	0...1H	0...1	1	[-]
Error flag for engine roughness improvement at stratified mode at warmup					
LV_CDN_DIAG_ER_STRAT_WUP	V	0...1H	0...1	1	[-]
Diagnosis condition for engine roughness improvement at stratified mode at warmup					
LV_END_DIAG_ER_STRAT_WUP	V	0...1H	0...1	1	[-]
Diagnosis result available for daignosis instance of engine roughness improvement at stratified mode at warmup					
ERR_SYM_ER_STRAT_WUP	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error sympthom for engine roughness improvement at stratified mode at warmup					
LV_ERR_N_MAX_DRIV_REQ	O/V	0...1H	0...1	1	[-]
Error flag for too high engine rotation speed at vehicle stop					
LV_CDN_DIAG_N_MAX_DRIV_REQ	V	0...1H	0...1	1	[-]
Diagnosis condition for too high engine rotation speed at vehicle stop					
LV_END_DIAG_N_MAX_DRIV_REQ	V	0...1H	0...1	1	[-]
Diagnosis result available for daignosis instance of too high engine rotation speed at vehicle stop					
ERR_SYM_N_MAX_DRIV_REQ	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error sympthom for too high engine rotation speed at vehicle stop					

Input data:

LV_ERR_SENS_POIL	LV_ERR_TCO_2_EL	LV_ERR_TCO_EL	LV_ERR_TPS_1
LV_ERR_MAF_FRQ_RNG[NC_MAF_NR]	LV_ERR_AMP	LV_ERR_EL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_LSH_UP[NC_CBK_EX_NR]
LV_ERR_VS	LV_ERR_RLY_ACCOUT	LV_ERR_ACR_AD	LV_ERR_ECF_EL[NC_ECF_NR]
LV_ERR_TOUT_ICL_2	LV_ERR_BN_ICL	LV_ERR_TOUT_ETCU_1	LV_ERR_BN_ETCU
LV_ERR_TOUT_ASR_1	LV_ERR_BN_TCS	LV_ERR_POIL_DR	LV_ERR_SYN_CRK_CAM_EX[NC_NR_CAM_CBK]
LV_ERR_SENS_POIL	LV_ERR_BSD	LV_ERR_SLV_IVVT_IN[NC_NR_CBK_IVVT]	LV_ERR_SLV_IVVT_EX[NC_NR_CBK_IVVT]
LV_ERR_PLAUS_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_PLAUS_CAM_IN[NC_NR_CAM_CBK]
LV_ERR_PER_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_PER_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_CRK_SYN	LV_ERR_ALTER_COM
LV_ERR_MIS[NC_CYL_NR]	LV_ERR_ECT_EL	LV_ERR_TH	LV_ERR_CWP_PWR
LV_ERR_CWP_PLAUS	LV_ERR_CWP_INT_OFF	LV_ERR_MEM_XX	LV_READY_XX
CTR_ABC_XX	ERR_SYM_MEM[NC_NR_ERR_DYN]	LV_VAR_BN	LV_DC_MAX[NC_NR_ERR_DYN]
WAL_CONF_XX	LV_ERR_PND[NC_NR_ERR_DYN]	LV_ERR_TIA_PLAUS	LV_ERR_TCO_PLAUS
LV_ERR_TCO_GRD	LV_ERR_TCO_STUCK	LV_ERR_TPS_2	LV_ERR_MTC_CTL_2
LV_ERR_MTC_CTL_3	LV_ERR_TPS_MAF_1	LV_ERR_TPS_ST_CHK_2	LV_ERR_TPS_AD
LV_ERR_MTC_DR	LV_ERR_TAM_CAN	LV_ERR_TAM_PLAUS	LV_ERR_TPS_AD_BOL
LV_ERR_MAF	LV_ERR_PBSU_PLAUS	LV_ERR_CS_PLAUS	
LV_ERR_EL_CPS	LV_ERR_SYN_CRK_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_FSD[NC_CBK_EX_NR]	LV_ERR_VIM_1_EL
LV_ERR_VIM_2_EL	LV_ERR_TPS_MAF_2	LV_ERR_SENS_BAT_SMT_COM	LV_ERR_CWP_COM
LV_ERR_QOIL_COM	LV_ERR_ACR_CTL	LV_ERR_MAP_DIP_SENS	LV_ERR_SENS_ACR
LV_ERR_MAP_DIP_SHIFT	LV_ERR_TPS_RATIO	LV_ERR_CRK_PLAUS	LV_ERR_CRK_TOOTH_PER
LV_ERR_SYN_CAM_EX[NC_NR_CAM_CBK]	LV_ERR_SYN_CAM_IN[NC_NR_CAM_CBK]	LV_ERR_VEH_POW_VAR	LV_ERR_AMP_PLAUS
LV_ERR_ECFPWM_FB[NC_ECF_NR]	LV_ERR_DIAGCPS	LV_ERR_CRK_TOOTH	LV_ERR_VIMPWM_1_FB
LV_ERR_VIMPWM_2_FB	LV_ERR_NS_OBD_1_HTP[NC_NOX_SENS_CONF]	LV_ERR_PUT_EL	LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]
LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_OC_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAF_FRQ_GRD[NC_MAF_NR]	LV_ERR_MAF_FRQ_EL[NC_MAF_NR]
LV_ERR_TOOTH_OFF_IN[NC_NR_CAM_CBK]	LV_ERR_TOOTH_OFF_EX[NC_NR_CAM_CBK]	LV_ERR_TEG_PCAT_DOWN	LV_ERR_PBSU
LV_ERR_NEUT_PSN_GB	LV_ERR_NS_OBD_1_NOX[NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_1_LAMB[NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_1_VLS[NC_NOX_SENS_CONF]
LV_ERR_IV[NC_CYL_NR]	LV_ERR_ACK_IGK_OFF	LV_ERR_NS_PWR[NC_NOX_SENS_CONF]	LV_ERR_WG_1_DR
LV_ERR_NS_VLS_DYN[NC_NOX_SENS_CONF]	LV_ERR_TEG_PCAT_DOWN	LV_ERR_NS_STOP[NC_NOX_SENS_CONF]	LV_ERR_NS_ACT[NC_NOX_SENS_CONF]
LV_ERR_NS_OFS[NC_NOX_SENS_CONF]	LV_ERR_NS_PUC[NC_NOX_SENS_CONF]	LV_ERR_NT_AGI	LV_ERR_NS_VLS_DYN[NC_NOX_SENS_CONF]
LV_ERR_CAT_DIAG[NC_CBK_EX_NR]	LV_ERR_CAT_DIAG_AFL[NC_CBK_EX_NR]	LV_ERR_NS_AFR[NC_NOX_SENS_CONF]	LV_ERR_NS_AVL[NC_NOX_SENS_CONF]
LV_ERR_NS_HTP[NC_NOX_SENS_CONF]	LV_ERR_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]	LV_ERR_NS_OFS[NC_NOX_SENS_CONF]	LV_ERR_NS_PUC[NC_NOX_SENS_CONF]
LV_ERR_NS_STOP[NC_NOX_SENS_CONF]	LV_ERR_TCO_2_GRD	LV_ERR_TCO_2_PLAUS	LV_ERR_TCO_STUCK_RNG
LV_ERR_VLS_NS_PUC[NC_NOX_SENS_CONF]	LV_ERR_LAMB_NS_PUC[NC_NOX_SENS_CONF]	LV_ERR_NOX_NS_PUC[NC_NOX_SENS_CONF]	LV_ERR_WG_2_DR

LV_DC	LV_ERR_SLV_IVVT_IN[NC_NR_CBK_IVVT]	LV_ERR_SLV_IVVT_EX[NC_NR_CBK_IVVT]	LV_ERR_NEUT_PSN_GB_PLAUS

Export actions:

ACTION_ERRM_CLCed_report (IN <f_index>, IN <err_mask>, OUT <err_status>)
This action is called by customer OBJ File to inform SV error manager concerning actual failure situation in customer OBJs. Error Status is sent back to customer OBJs.
ACTION_ERRM_CLCrbm_report (IN <rbm_index>, IN <rbm_data>) This action is called by customer OBJ File to inform SV error manager concerning actual failure situation for RBM
ACTION_ERRM_CLCed_status (IN <f_index>, OUT <err_status>)
This action is called by customer in order to get informations concerning a specific failure index.

FUNCTION DESCRIPTION:

General information:

This three Action Calls are used to exchange failure memory informations through the interface module (Layer) between SV SW environment and Customer OBJs. The error handling is performed in the error mangement in the SV SW. Even the anti-bounce algorithm for Customer OBJ exported failures is handled by SV SW.

Description:

The transformation between the dynamicly created error index list (f_index) / (rbm_index) on the customer OBJ Files side and the as well dynamicly created errror index list on SV SW side is realized by the help of the following translation table.

-

Syntax : ACTION_ERRM_CLCed_report (IN <f_index>, IN <err_mask>, OUT <err_status>)

Parameter (in) :	f_index	Index of failure based on dynamic list of the customer OBJ
	err_mask	Includes ERR_SYM_XX and LV_CDN_DIAG_XX
Parameter (out) :	err_status	Includes several informations concerning the corresponding index of failure

Short description : *This action is called by customer OBJ File to inform SV error manager concerning actual error situation in customer OBJs. Error Status is sent back to customer OBJs.*

-

Syntax : ACTION_ERRM_CLCrbm_report (IN <rbm_index>, IN <rbm_data>)

Parameter (in) :	rbm_index	Index of RBM - failure based on list of the customer OBJ
	rbm_data	Includes LV_END_RBM_XX and LV_INH_RBM_XX and LV_CDN_RBM_XX
Parameter (out) :	- -	

Short description : *This action is called by customer OBJ File to inform SV error manager concerning actual error situation in customer OBJs for RBM statictics. There is no return value.*

-

Syntax : ACTION_ERRM_CLCed_status (IN <f_index>, OUT <err_status>)

Parameter (in) : f_index Index of failure based on dynamic list of the customer OBJ

Parameter (out) : err_status Includes several informations concerning the corresponding index of failure

Short description : *This action is called by customer in order to get informations concerning a specific failure index.*

Application conditions:

Initialisation: -

Recurrence: depending on the index of failure

Activation: at action request

Formula section:

f_index:

f_index BMW	I/O	Failure BMW / FI_	Description BMW	Failure Flag SV XX	Description SV
1	I	TANS	Sensorspannung Ansaugluftemperatur	LV_ERR_TIA_EL **or** LV_ERR_TIA_PLAUS	Intake air temperature sensor
2	I	TKA	Sensorspannung Temperatur Kühlerausgang	LV_ERR_TCO_2_EL **or** LV_ERR_TCO_2_GRD **or** LV_ERR_TCO_2_PLAUS	Coolant temperature sensor (radiator out)
3	I	TMOT	Sensorspannung Motortemperatur	LV_ERR_TCO_EL **or** LV_ERR_TCO_PLAUS **or** LV_ERR_TCO_GRD **or** LV_ERR_TCO_STUCK **or** LV_ERR_TCO_STUCK_RNG	Coolant temperature sensor
4	I	UB	Batteriespg. nach HR	**Not available !**	
5	I	DK	Fehler Drosselklappe	LV_ERR_TPS_1 **or** LV_ERR_TPS_2 **or** LV_ERR_TPS_MAF_1 **or** LV_ERR_TPS_MAF_2 **or** LV_ERR_MTC_CTL_2 **or** LV_ERR_MTC_CTL_3 **or** LV_ERR_TPS_ST_CHK_2 **or** LV_ERR_TPS_AD **or**	TPS - electrical diagnosis TPS – controller – diagnosis TPS – adaptation

				LV_ERR_TPS_AD_BOL **or** LV_ERR_MTC_DR **or** LV_ERR_TPS_RATIO	TPS – powerstage diagnosis TPS – Ratio error
6	I	HFM	Luftmassenmesser	LV_ERR_MAF_FRQ_RNG_0 **or** LV_ERR_MAF_FRQ_FRQ_0 **or** LV_ERR_MAF_FRQ_EL_0	Mass air flow sensor diagnosis (only for **6** cylinder)
6	I	HFM	Luftmassenmesser	LV_ERR_MAF	Mass air flow sensor diagnosis (only for **4** cylinder)
7	I	PUS	Umgebungsdrucksensor	LV_ERR_AMP **or** LV_ERR_AMP_PLAUS	Ambient pressure sensor signal diagnosis
8	I	DPSR	Drucksensor Saugrohr (Differenzdruck)	LV_ERR_MAP_DIP_SENS **or** LV_ERR_MAP_DIP_SHIFT	Manifold pressure electrical diagnosis (N53)
9	I	LSV	Lambda-Sonde vor Kat	LV_ERR_EL_LSL_UP_1 **or** LV_ERR_OC_LSL_UP_1	El. Diagnosis WRAF sensor
10	I	HSV	Lambdasonden-Heizung vor Katalysator	LV_ERR_LSH_UP_1	Lambda sensor upstream heater diagnosis
11	I	LSV2	Lambda-Sonde vor Kat Bank 2	LV_ERR_EL_LSL_UP_2 **or** LV_ERR_OC_LSL_UP_2	El. Diagnosis WRAF sensor OBD1
12	I	HSV2	Lambdasonden-Heizung vor Katalysator Bank 2	LV_ERR_LSH_UP_2	Lambda sensor upstream heater diagnosis
13	I	KOSE	Klima-Kompressorsteurung Endstufe	LV_ERR_RLY_ACCOUT	Air cond. Compr. Relais diagnosis
14	I	ELUEE	E-Luefter Endstufe	LV_ERR_ECF_EL[NC_ECF_NR] **or** LV_ERR_ECFPWM_FB[NC_ECF_NR]	El. Cooling fan diagnosis
15	I	INS	Fehler Can-Botschaften vom Instrumentenkombi	LV_ERR_TOUT_ICL_2 v (f(LV_VAR_BN=0/1)) LV_ERR_BN_ICL	Diagnosis CAN timout (INST2) v BN2000 diagnosis KOMBI (Status)
16	I	EGS	Fehler Can-Botschaft Automatikgetriebe	LV_ERR_TOUT_ETCU_1 v (f(LV_VAR_BN=0/1)) LV_ERR_BN_ETCU	Diagnosis CAN timout (EGS1) v BN2000 diagnosis AT Gearbox (Status)
17	I	ASC	Fehler Can-Botschaft ASC	LV_ERR_TOUT_ASR_1 v (f(LV_VAR_BN=0/1)) LV_ERR_BN_TCS	Diagnosis CAN timout (ASC1) v BN2000 diagnosis Traction control system (Status DSC)
18	I	TUMG	Fehler Umgebungstemperatur	LV_ERR_TAM_CAN **or** LV_ERR_TAM_PLAUS	Ambient temperature diagnosis (CAN / Plaus)
19	O	ENWS	Fehler Einlassnockenwellensteuerung	LV_ERR_MEC_IVVT_IN	Functional diagnosis of IVVT inlet CAM
20	O	ANWS	Fehler Auslassnockenwellensteuerung	LV_ERR_MEC_IVVT_EX	Functional diagnosis of IVVT exhaust CAM
21	O	GEN	Fehler Generator	LV_ERR_GEN	Alternator diagnosis
22	O	BSD	Fehler Kommunikation		**Not supported from**

					BWM
23	I	ENWSE	Fehler Endstufe Einlassnockenwellensteuerung	LV_ERR_SLV_IVVT_IN	Driver diagnosis of solenoid valve IVVT inlet CAM
24	I	ANWSE	Fehler Endstufe Auslassnockenwellensteuerung	LV_ERR_SLV_IVVT_EX	Driver diagnosis of solenoid valve IVVT exhaust CAM
25	I	BM	Fehler Bezugsmarke (KW-Geber)	LV_ERR_CRK_SYN **or** LV_ERR_CRK_PLAUS **or** LV_ERR_CRK_TOOTH_PER **or** LV_ERR_CRK_TOOTH	Crankshaft sensor diagnosis; teeth not plausible
26	I	PH0_PER	Fehler Periode CAM 0	LV_ERR_PER_CAM_IN_1 **or** LV_ERR_SYN_CAM_IN_1	Camshaft sensor diagnosis
27	I	PH0_PSN	Fehler Position CAM 0	LV_ERR_REF_CRK_CAM_IN_1 **or** LV_ERR_TOOTH_OFF_IN_1	Camshaft sensor diagnosis (position)
28	I	PH0_PLA	Fehler Plausibilisierung CAM 0	LV_ERR_PLAUS_CAM_IN_1 **or** LV_ERR_SYN_CRK_CAM_IN_1	Camshaft sensor diagnosis
29	I	PH1_PER	Fehler Periode CAM 1	LV_ERR_PER_CAM_EX_1 **or** LV_ERR_SYN_CAM_EX_1	Camshaft sensor diagnosis
30	I	PH1_PSN	Fehler Position CAM 1	LV_ERR_REF_CRK_CAM_EX_1 **or** LV_ERR_TOOTH_OFF_EX_1	Camshaft sensor diagnosis (position)
31	I	PH1_PLA	Fehler Plausibilisierung CAM 1	LV_ERR_PLAUS_CAM_EX_1 **or** LV_ERR_SYN_CRK_CAM_EX_1	Camshaft sensor diagnosis
32	O	QLT	Fehler QLT	LV_ERR_QOIL_SENS	Oil sensor diagnosis
33	O	IBS1	Fehler Kommunkationsverlust IBS	LV_ERR_BAT_SENS	Intelligent battery sensor diagnosis
34	O	IBS2	Fehler IBS unplausibel	LV_ERR_BAT_SENS_IT	Intelligent battery sensor diagnosis
35	O	IBS3	Fehler IBS allgemein	LV_ERR_BAT_SENS_IT_EL	Intelligent battery sensor diagnosis
36	O	PMBN	Fehler PM-Bordnetz	LV_ERR_ALTER_BN	Powermanagement diagnosis
37	O	PMBATT	Fehler PM-Batterie	LV_ERR_ALTER_BN_BAT	Powermanagement diagnosis
38	I	VFZ	Fehler Fahrzeuggeschwindigkeit	LV_ERR_VS	Vehicle speed signal diagnosis
39	I	MDZ0	Aussetzer Zuendung 0	LV_ERR_MIS_0	Cylinder specific misfire failure
40	I	MDZ1	Aussetzer Zuendung 1	LV_ERR_MIS_1	Cylinder specific misfire failure
41	I	MDZ2	Aussetzer Zuendung 2	LV_ERR_MIS_2	Cylinder specific misfire failure
42	I	MDZ3	Aussetzer Zuendung 3	LV_ERR_MIS_3	Cylinder specific misfire failure
43	I	MDZ4	Aussetzer Zuendung 4	LV_ERR_MIS_4	Cylinder specific misfire failure, not supported for 4-cylinder SW

44	I	MDZ5	Aussetzer Zuendung 5	LV_ERR_MIS_5	Cylinder specific misfire failure, not supported for 4-cylinder SW
45	I	MDZ6	Aussetzer Zuendung 6		**Not implemented in MSDxx**
46	I	MDZ7	Aussetzer Zuendung 7		**Not implemented in MSDxx**
47	I	KFT	Fehler Kennfeldthermostat	LV_ERR_ECT_EL **or** LV_ERR_TH	Electronically controlled thermostat diagnosis
48	I	EWAPU	Fehler elektrische Wasserpumpe	LV_ERR_CWP_PWR **or** LV_ERR_CWP_PLAUS **or** LV_ERR_CWP_INT_OFF	Cooling water pump diagnosis
49	O	WMKD	Fehler koordinierung Wärmemanagement		**Not supported from BWM**
50	I	PLD	Druck vor Drosselklappe	LV_ERR_PUT_EL	Pressure-up throttle sensor – electrical diagnosis (not supported for 4-cylinder SW !)
51	O	HFMPL	HFM plausibilisierung	LV_ERR_LOAD_TPS_PLAUS	Load – TPS ratio check
52	O	DISAPL	DISA plausibilisierung	LV_ERR_VIM_PLAUS	VIM plausibility diagnosis
53	O	DPSRPL	Saugrohrdrucksensor plausibilisierung	LV_ERR_MAP_DIP_PLAUS	MAP_DIP_SENS plausibility diagnosis
54	O	TEVPL	Tankent-lüftungsventil plausibilisierung		**Not supported from BWM**
55	O	SLPPL	Sekundärluftpumpe plausibilisierung		**Not supported from BWM**
56	I	DKALRN	Drosselklappe – Lernen untere Anschlag	LV_ERR_TPS_AD_BOL	TPS adaption lower position
57	I	TEVE	Endstufe Tankentlüftungsventil	LV_ERR_EL_CPS **or** LV_ERR_DIAGCPS	ATIC39 Powerstage-diagnoisis CPS / TEV check
58	I	SLPE	Endstufe Sekundärluftpumpe		**Not implemented in MSDxx**
59	I	KSD_1	Kraftstoffsystemdiagnose	LV_ERR_FSD_1 **or** LV_ERR_DELTA_I_LAM_1 **or** LV_ERR_FSD_LAM_LIM_1	Fuel-system diagnosis
60	I	KSD_2	Kraftstoffsystemdiagnose	LV_ERR_FSD_2 **or** LV_ERR_DELTA_I_LAM_2 **or** LV_ERR_FSD_LAM_LIM_2	Fuel-system diagnosis
61	I	DISAE_H	Endstufe DISA-H	LV_ERR_VIM_1_EL **or** LV_ERR_VIMPWM_1_FB	ATIC39 Powerstage-diagnoisis VIM_1
62	I	DISAE_L	Endstufe DISA-L	LV_ERR_VIM_2_EL **or** LV_ERR_VIMPWM_2_FB	ATIC39 Powerstage-diagnoisis VIM_2
63	I	BSD_0	Kommunikation BSD	LV_ERR_SENS_BAT_SMT_COM	BSD diagnosis
64	I	BSD_1	Kommunikation BSD		**Not implemented in MSDxx**
65	I	BSD_2	Kommunikation BSD		**Not implemented in MSDxx**
66	I	BSD_3	Kommunikation BSD	LV_ERR_CWP_COM	BSD diagnosis, CWP

					device
67	I	BSD_4	Kommunikation BSD	LV_ERR_QOIL_COM	BSD diagnosis, QOIL device
68	I	BSD_5	Kommunikation BSD		**Not implemented in MSDxx**
69	I	BSD_6	Kommunikation BSD	LV_ERR_ALTER_COM	BSD diagnosis, ALTER device
70	I	BSD_7	Kommunikation BSD		**Not implemented in MSDxx**
71	O	PMRUHVERL	Powermanagement Ruhestromverletzung	LV_ERR_ALTER_BN_RGN	Powermanagement diagnosis
72	I	BSDGLOB	Fehler BSD global	LV_ERR_BSD	BSD diagnosis
73	O	TOENS	Fehler Toens	LV_ERR_TOIL_LEVEL	TOENS fehler
74	I	PWRVAR	Fehler OL / UL	LV_ERR_VEH_POW_VAR	Vehicle power variant not plausible
75	O	PUSPL	Fehler Umgebungsdrucksensor	LV_ERR_AMP_PLAUS_CUS	Ambient pressure sensor not plausible
76	I	DKOBD	Fehler Drosselklappe OBD relevant	LV_ERR_TPS_1 **or** LV_ERR_TPS_2 **or** LV_ERR_TPS_MAF_1 **or** LV_ERR_TPS_MAF_2 **or** LV_ERR_MTC_CTL_2 **or** LV_ERR_MTC_CTL_3 **or** LV_ERR_TPS_ST_CHK_2 **or** LV_ERR_TPS_AD **or** LV_ERR_TPS_AD_BOL **or** LV_ERR_MTC_DR	Emission relevant throttle error.
77	I	SEGRADP	Segmentzeitadaptionsfehler	LV_ERR_SEG_AD_ER	Segment-time adaptation values out of range
78	I	AGR_AD	EGR -adaptionsfehler	LV_ERR_ACR_AD	Adaptation of EGR valve
79	I	AGR_SENS	EGR -Sensorfehler	LV_ERR_SENS_ACR	EGR sensor diagnosis
80	I	AGR_DR	EGR - Endstufenfehler	LV_ERR_ACR_DR	EGR powerstage diagnosis
81	I	AGR_CTL	Fehler EGR -lageregler	LV_ERR_ACR_CTL	EGR controller diagnosis
82	I	AGR_VCC	VCC Überwachung EGR ventil	LV_ERR_V_REF_1	Power supply diagnosis PVS / EGR....
83	O	NWEKW	Versatz Einlass NW zu KW		**Not supported in MSDxx**
84	O	NWAKW	Versatz Auslass NW zu KW		**Not supported in MSDxx**
85	O	ENWSAD	Einlassnockenwellensteuerung Adaption		**Not supported in MSDxx**
86	O	ANWSAD	Auslassnockenwellensteuerung Adaption		**Not supported in MSDxx**
87	O	ATLRMX	Lader Systemdiagnosis 1	LV_ERR_TCHA_SYS_1	
88	O	PLDMN	Lader Systemdiagnosis 3	LV_ERR_TCHA_PRS_LOW	Charger pressure to low
89	O	ATLLEKFAST	Lader Systemdiagnosis 5	LV_ERR_TCHA_LEAK	Charger leakage

90	O	ATLGLS	Lader Systemdiagnosis 7	LV_ERR_TCHA_PRS_DIF	Charger pressure difference
91	O	PLDMX	Lader Systemdiagnosis 4	LV_ERR_TCHA_PRS_HIGH	Charger pressure to high
92	I	POELSENS	Öldrucksensor	LV_ERR_SENS_POIL	Oilpressure sensor error
93	I	POELREGVE	Öhldruck Regenventil Endstufe	LV_ERR_POIL_DR	Oilpressure valve powerstage error
94	O	POELRSTAT	Öldruckregellung statisch	LV_ERR_POIL_CTL_STAT	Oilpressure static - control error
95	O	POELRDYN	Öldruckregellung dynamisch	LV_ERR_POIL_CTL_DYN	Oilpressure dynamic - control error
96	I	DKRAT1	?	LV_ERR_TPS_MAF_1	
97	I	DKRAT2	?	LV_ERR_TPS_MAF_2	
98	I	DKRATIO	?	LV_ERR_TPS_RATIO	
99	I	ABGAS_TSENS	Fehler Abgastemperatur	LV_ERR_TEG_PCAT_DOWN	Exhaust temperature sensor
100	I	NOX_SENS	Fehler NOx Sensor		**Not supported in MSDxx**
101	I	DK_EF	Fehler einfach elektrisch an DK	LV_ERR_TPS_1 **or** LV_ERR_TPS_2	Electrical throttle error
102	I	PVDKDS	Fehler elektrisch Druck vor DK	LV_ERR_PUT_EL	Pressure-up throttle sensor – electrical diagnosis (not supported for 4-cylinder SW !)
103	I	PS_ABS	Drucksensor Saugrohr (Absolutdruck)	LV_ERR_MAP_DIP_SENS	Manifold pressure electrical diagnosis (N54)
104	O	PVDKDS_PL	Fehler Plausibilität Druck vor Drosselklappe	LV_ERR_PUT_PLAUS	Pressure-up throttle sensor – plausibility diagnosis
105	O	PSABS_PL	Fehler Plausibiltät Absolutdrucksensor	LV_ERR_MAP_PLAUS	Ambient pressure sensor not plausible
106	I	DK_NL	Fehler DK Notlauf		**Not defined yet**
107	O	DMDKH		LV_ERR_TQ_CST	Cold start monitoring - Torque reserve catalyst heating
108	O	POILREGV	Fehler Öldruckregelventil mechanisch	LV_ERR_POIL_CTL_MEC	Oilpressure control valve mecanical error
109	O	POILSENS_PL	Fehler Öldrucksensor	LV_ERR_POIL_SENS_PLAUS	Oilpressure sensor not plausible
110	I	PBREMSU	Fehler Bremsunterdrucksensor	LV_ERR_PBSU	Brake pressure sensor error (only 4 cylinder)
111	I	NGANG	Fehler Nullgangsensor	LV_ERR_NEUT_PSN_GB	Neutral gear sensor (only 4 cylinder)
112	O	NGLERN	Fehler Nullgangsensorlernen	LV_ERR_NEUT_PSN_GB_LRN	Neutral gear sensor learning
113	O	POILPUMP	Fehler Oeldruckpumpe	LV_ERR_POIL_PUMP	Oilpressure pump
114	O	POILSYS	Fehler Oeldrucksystem allg.	LV_ERR_POIL_SYS	Oelpressure system
115	O	GENEL	Elektrischer Fehler Generator	LV_ERR_GEN_EL	electrical generator error
116	O	GENMECH	Mechanischer Fehler Generator	LV_ERR_GEN_MEC	mechanical generator error
117	O	GENHT	Hochtemperaturfehler Generator	LV_ERR_GEN_H_TEMP	generator high temperature error
118	O	GENUPL	Generatortyp unplausibel	LV_ERR_GEN_TYP_NOT_PLAUS	generator type not plausible
119	O	GENKOMM	Keine Generatorkommunikation	LV_ERR_GEN_MSG_LOST	generator message lost

120	O	GENELB	Generatorspannung ausBerechnung unplausibel	LV_ERR_GEN_CLC_V_NOT_PLAUS	generator voltage not plausible
121	O	GENHTB	Hochtemperaturfehler Generator aus Berechnung	LV_ERR_GEN_H_TEMP_CLC	calculated generator high temperature error
122	O	GENREGUPL	Generatorregler unplausibel	LV_ERR_GEN_CTL_NOT_PLAUS	calculated generator controller plausibility error
123	O	AGRPL	AGR - Ventil unplausibel	LV_ERR_EGRV_PSN_PLAUS	Error flag for position of exhaust gas recirculation valve plausibility
124	I	KAT	Fehler Katalysator	LV_ERR_CAT_DIAG[1] **or** LV_ERR_CAT_DIAG[2]	Present failure catalyst efficiency
125	I	WG	Fehler Wastegate Ansteuerung	LV_ERR_WG_1_DR **or** LV_ERR_WG_2_DR	Waste gate driver error cylinder bank 1 or 2 detected (only 6 cyclinder)
126	I	TVLD	Fehler Temperatursensor vor Verdichter	LV_ERR_TEG_PCAT_DOWN	TEG sensor diagnosis
127	O	ATL_CTL	Fehler Ladedruckregelung	LV_ERR_TCHA_PRS_CTL	error currently present on charger pressure controller diagnosis
128	I	KATH_DIAG	Fehlerort Katdiagnose	LV_ERR_CAT_DIAG_AFL[1] **or** LV_ERR_CAT_DIAG_AFL[2]	Present failure after filtering of diagnostic CAT_DIAG_AFL
129	I	NSOBD2	Wenn der NOx - Sensor defekt erkannt wird	LV_ERR_NS_ACT [1] **or** LV_ERR_NS_AFR[1] **or** LV_ERR_NS_AVL[1] **or** LV_ERR_NS_HTP[1] **or** LV_ERR_NS_LSL_UP_DOWN[1] **or** LV_ERR_NS_OFS[1] **or** LV_ERR_NS_PUC[1] **or** LV_ERR_NS_STOP[1] **or** LV_ERR_NS_VLS_DYN[1]	NOx sensor OBD 2 failure is present (only **6** cylinder)
130	I	NSOBD2	Wenn der NOx - Sensor defekt erkannt wird	LV_ERR_NS_ACT [1] **or** LV_ERR_NS_AFR[1] **or** LV_ERR_NS_AVL[1] **or** LV_ERR_NS_HTP[1] **or** LV_ERR_NS_LSL_UP_DOWN[1] **or** LV_ERR_NS_OFS[1] **or** LV_ERR_NS_PWR [1] **or** LV_ERR_NS_STOP[1] **or** LV_ERR_NS_VLS_DYN[1] **or** LV_ERR_VLS_NS_PUC[1] **or**	NOx sensor OBD 2 failure is present (only **4** cylinder)

				LV_ERR_LAMB_NS_PUC[1] **or** LV_ERR_NOX_NS_PUC[1]	
130	I	NTAGI	Wenn der NOx - Kat defekt erkannt wird	LV_ERR_NT_AGI	Error flag of NOx catalyst aging diagnosis
131	I	NSOBD1	NOX - Sensor	LV_ERR_NS_OBD_1_HTP[i] **or** LV_ERR_NS_OBD_1_NOX[i] **or** LV_ERR_NS_OBD_1_LAMB[i] **or** LV_ERR_NS_OBD_1_VLS[i]	NOx sensor 1 or 2 OBD I failure is present
132	I	EV_DIAG_0	Einspritzventil Diagnose	LV_ERR_IV[0]	at least one IV is detected as faulty
133	I	EV_DIAG_1	Einspritzventil Diagnose	LV_ERR_IV[1]	at least one IV is detected as faulty
134	I	EV_DIAG_2	Einspritzventil Diagnose	LV_ERR_IV[2]	at least one IV is detected as faulty
135	I	EV_DIAG_3	Einspritzventil Diagnose	LV_ERR_IV[3]	at least one IV is detected as faulty
136	I	EV_DIAG_4	Einspritzventil Diagnose	LV_ERR_IV[4]	at least one IV is detected as faulty (not supported in 4 cylinder sw)
137	I	EV_DIAG_5	Einspritzventil Diagnose	LV_ERR_IV[5]	at least one IV is detected as faulty (not supported in 4 cylinder sw)
138	O	NMAX_BA	Fehler Drehzahlbegrenzung durch Notlaufmanager	LV_ERR_N_MAX_HOMS	engine overspeed for changing the combustion mode
139	O	DKPSABS_PL	Fehler Saugrohrdruck-DK Verhältnis Plausibilisierung	LV_ERR_MAP_TPS _PLAUS	Boolean for error MAP ratio plausibility diagnosis
140	I	NG_PLAUS	Plauibilität Signal Nullganssensor	LV_ERR_NEUT_PSN_GB_PLAUS	Neutral gear sensor plausibility error
141	O	MSLAM_MX	Ueberschreitung des maximalen Luftmassenstromes	LV_ERR_MAF_LAMB_MAX	Error flag for maximum mass air flow calculated from lambda
142	I	PBU_PLAUS	Plausibilität Bremsunterdrucksignal	LV_ERR_PBSU_PLAUS	Plausibility error brake pressure sensor (only for **4** cylinder)
143	O	LVS_AKTIV	Eingriff über LVS aktiv	LV_ERR_ER_STRAT	Error flag for engine roughness improvement at stratified mode
144	O	LVSWL_AKT	Eingriff über LVS aktiv im Motorwarmlauf	LV_ERR_ER_STRAT_WUP	Error flag for engine roughness improvement at stratified mode at warmup
145	I	MSALTG	Fehler MSA Leitung	LV_ERR_ACK_IGK_OFF	Starter relay MSA power stage error (only **4** cylinder)
146	I	KUPP_PLAUS	Kupplungssignal nicht plausibel	LV_ERR_CS_PLAUS	Clutch switch plausibility error (only **4** cylinder)
147	O	PWGLL_NKW	Drehzahlreduzierung umd	LV_ERR_N_MAX_DRIV_REQ	Error flag for too high

		RED	EML Lampe durch Vollgas		engine rotation speed at vehicle stop

rbm_index:

rbm_index BMW	Failure BMW	Description BMW	RBM data SV XX	Description SV
0	FI_ENWS	Fehler Einlassnockenwellensteuerung	LV_END_RBM_MEC_IVVT_IN LV_INH_DIAG_RBM_MEC_IVVT_IN LV_CDN_RBM_MEC_IVVT_IN	Functional diagnosis of IVVT inlet CAM
1	FI_ANWS	Fehler Auslassnockenwellensteuerung	LV_END_RBM_MEC_IVVT_EX LV_INH_DIAG_RBM_MEC_IVVT_EX LV_CDN_RBM_MEC_IVVT_EX	Functional diagnosis of IVVT exhaust CAM
2	FI_HFMPL	HFM plausibilisierung	LV_END_RBM_LOAD_TPS_PLAUS LV_INH_DIAG_RBM_LOAD_TPS_PLAUS LV_CDN_RBM_LOAD_TPS_PLAUS	Load – TPS ratio check
3	FI_DPSRPL	Saugrohrdrucksensor plausibilisierung	LV_END_RBM_MAP_DIP_PLAUS LV_INH_DIAG_RBM_MAP_DIP_PLAUS LV_CDN_RBM_MAP_DIP_PLAUS	MAP_DIP_SENS plausibility diagnosis
4	FI_TEVPL	Tankent-lüftungsventil plausibilisierung	**Not supported !**	
5	FI_NWEKW	Fehler Versatz Einlass-NW zu KW	**Not supported !**	
6	FI_NWAKW	Fehler Versatz Auslass-NW zu KW	**Not supported !**	
7	FI_ENWSAD	Fehler Einlassnocken-wellensteuerung Anschlagsadaption	**Not supported !**	
8	FI_ANWSAD	Fehler Auslassnockenwellensteuerung Anschlagsadaption	**Not supported !**	
9	FI_DMDKH	No Text	LV_END_RBM_TQ_CST LV_INH_DIAG_RBM_TQ_CST LV_CDN_RBM_TQ_CST	Cold start monitoring – Torque reserve catalyst heating

err_mask:

ERR_SYM_XX	= bit 0-3	*//detected symptom*
LV_CDN_DIAG_XX	= bit 4	*//diagnosis condition*
LV_INH_DIAG_RBM_XX	= bit 5	*//RBM inhibition due to OBD error, only used for RBM diagnosis*

rbm_data:

LV_END_RBM_XX	= bit 0	*//diagnosis cycle for RBM finished*
LV_INH_DIAG_RBM_XX	= bit 1	*//RBM inhibition due to OBD error*
LV_CDN_RBM_XX	= bit 2	*//Special RBM diagnosis condition*
not used	= bit 3-7	

err_status:

Remark:
For some f_index BMW there are more then one LV_ERR_xx (Diagnosis instances) which fit, thus a algorithm to set the bits in the right way is defined:
e.g. f_index 1 = LV_ERR_TIA_EL **or** LV_ERR_TIA_PLAUS
=> Formula section below is described in the way LV_ERR_XX **or** LV_ERR_YY

Status of memorised symptom
```
bit 3-0   of error_status        =  ERR_SYM_MEM_IDX   with IDX = f(XX)
                                 (In case of more then one ERR_SYM_MEM_IDX,
                                 only the first is considered )
```

Status of MIL activation
```
bit 4     of error_status    if     LV_DC_MAX_IDX = 1
                                    with IDX = f(XX)
                                    &  bit 2 of WAL_CONF_XX  =  1          or
                                    LV_DC_MAX_IDX = 1
                                    with IDX = f(YY)
                                    &  bit 2 of WAL_CONF_YY  =  1
                             then   bit 4 of error_status = 1
                             else   bit 4 of error_status = 0
```

Status of present error and passive symptom
```
bit 5     of error_status    if     CTR_ABC_xx  ≠  0            or
                                    CTR_ABC_yy ≠  0             or
                                    LV_ERR_xx  =  1                    or
                                    LV_ERR_yy  =  1
                             then   bit 5 of error_status = 1
                             else   bit 5 of error_status = 0
```

Status of present error
```
bit 6     of error_status    if     LV_ERR_XX  =  1             or
                                    LV_ERR_YY  =  1
                             then   bit 6 of error_status = 1
                             else   bit 6 of error_status = 0
```

Status of end of diagnosis cycle
```
bit 7     of error_status    if     LV_END_DIAG_XX  =  1        and
                                    LV_END_DIAG_YY =  1
                             then   bit 7 of error_status = 0
                             else   bit 7 of error_status = 1
```

Status of OBD readiness
```
bit 8     of error_status    if     LV_READY_XX  =  0           and
                                    LV_READY_YY  =  0
                             then   bit 8 of error_status = 0
                             else   bit 8 of error_status = 1
```

Status if present error is stored in ERRM

bit 9 of error_status **if** LV_ERR_MEM_XX = 1 **or**
LV_ERR_MEM_YY = 1
then bit 9 of error_status = 1
else bit 9 of error_status = 0

Status of diagnosis condition

bit 10 of error_status **if** LV_CDN_DIAG_XX = 1 **or**
LV_CDN_DIAG_YY = 1
then bit 10 of error_status = 1
else bit 10 of error_status = 0

Status of detected symtom

bit 11 of error_status **if** ERR_SYM_XX != 0 **or**
ERR_SYM_YY != 0
then bit 11 of error_status = 1
else bit 11 of error_status = 0

Status if error is pending

bit 14 of error_status **if** LV_ERR_PND_IDX = 1 **with** IDX = f(XX) **or**
LV_ERR_PND_IDX = 1 **with** IDX = f(YY)
then bit 14 of error_status = 1
else bit 14 of error_status = 0

Status if error is calibrated as OBD relevat

bit 15 of error_status **if** bit 4 of WAL_CONF_XX = 1 **or**
bit 4 of WAL_CONF_YY = 1
then bit 15 of error_status = 1
else bit 15 of error_status = 0

27.3.2 Output – data of diagnosis handled by customer OBJ File

FUNCTION DESCRIPTION:

General information:

All failures are fed by the customer diagnosis algorithm which detects malfunctions.

The relevant informations for the error management are transferred with the help of the action call.

To give a reflection of the actual status of the failure memory management - which might as well be used for QVM related affairs (if desired) and/or for the information of system developers – the relevant SV variables remain defined.

Since the debounce algorithm is part of the SV failure memory management, all corresponding calibration variables (C_ABC_...) remain on SV side.

All diagnosis are using the STD_INI debounce algorithm (not valid for LOAD_TPS_PLAUS diagnosis).

Description:

The following diagnosis LV_ERR_*(DIAG_INST)* are defined:

- ENWS (DIAG_INST = **MEC_IVVT_IN**)
- ANWS (DIAG_INST = **MEC_IVVT_EX**)
- GEN (DIAG_INST = **GEN**)
- QLT (DIAG_INST = **QOIL_SENS**)
- IBS1 (DIAG_INST = **BAT_SENS**)
- IBS2 (DIAG_INST = **BAT_SENS_IT**)
- IBS3 (DIAG_INST = **BAT_SENS_IT_EL**)
- PMBN (DIAG_INST = **ALTER_BN**)
- PMBATT (DIAG_INST = **ALTER_BN_BAT**)
- HFMPL (DIAG_INST = **LOAD_TPS_PLAUS**)
- DISAPL (DIAG_INST = **VIM_PLAUS**)
- DPSRPL (DIAG_INST = **MAP_DIP_PLAUS**)
- PMRUHVERL (DIAG_INST = **ALTER_BN_RGN**)
- TOENS (DIAG_INST = **TOIL_LEVEL**)
- PUSPL (DIAG_INST = **AMP_PLAUS_CUS**)
- PLDMN (DIAG_INST = **TCHA_PRS_LOW**)
- ATLLEKFAST (DIAG_INST = **TCHA_LEAK**)
- ATLGLS (DIAG_INST = **TCHA_PRS_DIF**)
- PLDMX (DIAG_INST = **TCHA_PRS_HIGH**)
- POELRSTAT (DIAG_INST = **POIL_CTL_STAT**)
- POELRDYN (DIAG_INST = **POIL_CTL_DYN**)
- PVDKDS_PL (DIAG_INST = **PUT_PLAUS**)
- PSABS_PL (DIAG_INST = **MAP_PLAUS**)
- DMDKH (DIAG_INST = **TQ_CST**)
- POILREGV (DIAG_INST = **POIL_CTL_MEC**)
- POILSENS_PL (DIAG_INST = **POIL_SENS_PLAUS**)
- NGLERN (DIAG_INST = **NEUT_PSN_GB_LRN**)
- POILPUMP (DIAG_INST = **POIL_PUMP**)
- POILSYS (DIAG_INST = **POIL_SYS**)

- GENEL (DIAG_INST = **GEN_EL**)
- GENMECH (DIAG_INST = **GEN_MEC**)
- GENHT (DIAG_INST = **GEN_H_TEMP**)
- GENUPL (DIAG_INST = **GEN_TYP_NOT_PLAUS**)
- GENKOMM (DIAG_INST = **GEN_MSG_LOST**)
- GENELB (DIAG_INST = **GEN_CLC_V_NOT_PLAUS**)
- GENHTB (DIAG_INST = **GEN_H_TEMP_CLC**)
- GENREGUPL (DIAG_INST = **GEN_CTL_NOT_PLAUS**)
- AGRPL (DIAG_INST = **EGRV_PSN_PLAUS**)
- ATL_CTL (DIAG_INST = **TCHA_PRS_CTL**)
- NMAX_BA (DIAG_INST = **N_MAX_HOMS**)
- DKPSABS_PL (DIAG_INST = **MAP_TPS_PLAUS**)
- ATLRMX (DIAG_INST = **TCHA_SYS_1**)
- MSLAM_MX (DIAG_INST = **MAF_LAMB_MAX**)
- LVS_AKTIV (DIAG_INST = **ER_STRAT**)
- LVSWL_AKT (DIAG_INST = **ER_STRAT_WUP**)
- PWGLL_NKWRED (DIAG_INST = **N_MAX_DRIV_REQ**)

Application conditions:

Initialisation: according **STD_INI** configuration

(all 0 at LV_IGK 0->1 or reset)

only for DIAG_INST = LOAD_TPS_PLAUS

according **STD_INI** and **DEC_CAL** configuration

(all 0 at LV_IGK 0->1 or reset + cal. Decrement)

only for DIAG_INST = PUT_PLAUS

according **STD** configuration

all other Output data 0 at LV_IGK 0->1 or reset

Recurrence: see activation

Activation: at ACTION_ERRM_CLCed_report (IN <f_index>, IN <err_mask>, OUT <err_status>)

Formula section:

According the data delivered by the err_masks (includes ERR_SYM_*(DIAG_INST)* and LV_CDN_DIAG_*(DIAG_INST)* the anti-bounce algorithm is called:

If ERR_SYM_*(DIAG_INST)* is not NO_SYM

Then debounce of LV_ERR_*(DIAG_INST)* using general debounce algorithm

Else rebounce of LV_ERR_*(DIAG_INST)* using general debounce algorithm

Endif

End of diagnosis calculation LV_END_DIAG_*(DIAG_INST)* = 1*:*

--> see chapter "Anti-bounce Algorithm: Calculation of end of diagnosis"

27.3.3 Rate – base monitor Interface for OBJ-file diagnosis

FUNCTION DESCRIPTION:

General information:

With this module the interface between the Object-file diagnosis and the Rate-Based Monitoring statistics is defined with STATE_RBM_xx data.

Within STATE_RBM_xx, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0) (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1) (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

xx = applied for MEC_IVVT_IN / MEC_IVVT_EX / MAP_DIP_PLAUS / LOAD_TPS_PLAUS / TQ_CST

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_xx = 0

on failure memory reset :

bit 1 of STATE_RBM_xx = 0

Recurrence: see activation

Activation: at ACTION_ERRM_CLCrbm_report (IN <rbm_index>, IN <rbm_data>)

Deactivation: LV_DC = 0

Formula section:

Algorithm must be applied sperate for each defined xx diagnosis

```
If    bit 0 of STATE_RBM_xx = 0
Then
        If      LV_END_RBM_xx = 1
        Then    bit 0 of STATE_RBM_xx = 1
        Endif
Endif
If    bit 1 STATE_RBM_xx = 0
Then
        If      LV_INH_DIAG_RBM_xx = 1
        Then    bit 1 of STATE_RBM_xx = 1
```

```
        Endif
Endif
If      bit 2 STATE_RBM_xx = 0
Then    If      LV_CDN_RBM_xx = 1
        Then    bit 2 of STATE_RBM_xx = 1
        Endif
Endif
```

27.3.4 Outputs for SV aggregates, SV internally

FUNCTION DESCRIPTION:

General information:

Adaptation to aggregate-environment.

Application conditions:

Initialisation:	0 at reset
Recurrence:	same as corresponding input data
Activation:	every engine operating state

Formula section:

LV_EOL_OBD_DC = 0

Global error flag for IVVT system

```
If      LV_ERR_MEC_IVVT_IN     = 1
   Or   LV_ERR_MEC_IVVT_EX     = 1
   Or   LV_ERR_SLV_IVVT_IN     = 1
   Or   LV_ERR_SLV_IVVT_EX     = 1
Then    LV_ERR_IVVT            = 1
Else    LV_ERR_IVVT            = 0
Endif
```

Global error flag for OIL system

```
If      LV_ERR_TOIL_LEVEL      = 1
   Or   LV_ERR_QOIL_SENS       = 1
Then    LV_ERR_TOIL            = 1
Else    LV_ERR_TOIL            = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_GEN	1	0...FFH	0...255	1	[-]
Debounce counter increment - GEN diagnosis					
C_ABC_MAX_GEN	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - GEN diagnosis					
C_ABC_INC_QOIL_SENS	1	0...FFH	0...255	1	[-]
Debounce counter increment - QOIL_SENS diagnosis					
C_ABC_MAX_QOIL_SENS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - QOIL_SENS diagnosis					
C_ABC_INC_BAT_SENS	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ BAT_SENS diagnosis					
C_ABC_MAX_BAT_SENS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ BAT_SENS diagnosis					
C_ABC_INC_BAT_SENS_IT	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ BAT_SENS_IT diagnosis					
C_ABC_MAX_BAT_SENS_IT	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ BAT_SENS_IT diagnosis					
C_ABC_INC_ALTER_BN	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ ALTER_BN diagnosis					
C_ABC_MAX_ALTER_BN	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ ALTER_BN diagnosis					
C_ABC_INC_ALTER_BN_BAT	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ ALTER_BN_BAT diagnosis					
C_ABC_MAX_ALTER_BN_BAT	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ ALTER_BN_BAT diagnosis					
C_ABC_INC_LOAD_TPS_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ LOAD_TPS_PLAUS diagnosis					
C_ABC_MAX_LOAD_TPS_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ LOAD_TPS_PLAUS diagnosis					
C_ABC_DEC_LOAD_TPS_PLAUS	1	0...FFH	0...255	1	[-]
Decrement of the load/TPS plausibility check anti-bounce counter					
C_ABC_INC_VIM_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ VIM_PLAUS diagnosis					
C_ABC_MAX_VIM_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ VIM_PLAUS diagnosis					
C_ABC_INC_MAP_DIP_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ MAP_DIP_PLAUS diagnosis					
C_ABC_MAX_MAP_DIP_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ MAP_DIP_PLAUS diagnosis					
C_ABC_INC_ALTER_BN_RGN	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ ALTER_BN_RGN diagnosis					
C_ABC_MAX_ALTER_BN_RGN	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ ALTER_BN_RGN diagnosis					
C_ABC_INC_TOIL_LEVEL	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ TOIL_LEVEL diagnosis					
C_ABC_MAX_TOIL_LEVEL	1	1...FFH	1...255	1	[-]
Debounce counter maximum value ¿ TOIL_LEVEL diagnosis					
C_ABC_INC_AMP_PLAUS_CUS	1	0...FFH	0...255	1	[-]
Debounce counter increment AMP_PLAUS_CUS diagnosis					
C_ABC_MAX_AMP_PLAUS_CUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value AMP_PLAUS_CUS diagnosis					
C_ABC_INC_TCHA_PRS_LOW	1	0...FFH	0...255	1	[-]
Debounce counter increment charger pressure to low diagnosis					
C_ABC_MAX_TCHA_PRS_LOW	1	1...FFH	1...255	1	[-]
Debounce counter maximum charger pressure to low diagnosis					
C_ABC_INC_TCHA_LEAK	1	0...FFH	0...255	1	[-]
Debounce counter increment charger leakage diagnosis					

C_ABC_MAX_TCHA_LEAK	1	1...FFH	1...255	1	[-]
Debounce counter maximum charger leakage diagnosis					
C_ABC_INC_TCHA_PRS_DIF	1	0...FFH	0...255	1	[-]
Debounce counter increment charger pressure difference diagnosis					
C_ABC_MAX_TCHA_PRS_DIF	1	1...FFH	1...255	1	[-]
Debounce counter maximum charger pressure difference diagnosis					
C_ABC_INC_TCHA_PRS_HIGH	1	0...FFH	0...255	1	[-]
Debounce counter increment charger pressure to high diagnosis					
C_ABC_MAX_TCHA_PRS_HIGH	1	1...FFH	1...255	1	[-]
Debounce counter maximum charger pressure to high diagnosis					
C_ABC_INC_TCHA_SYS_1	1	0...FFH	0...255	1	[-]
Debounce counter increment charger system diagnosis 1					
C_ABC_MAX_TCHA_SYS_1	1	1...FFH	1...255	1	[-]
Debounce counter maximum charger lsystem diagnosis 1					
C_ABC_INC_POIL_CTL_STAT	1	0...FFH	0...255	1	[-]
Debounce counter increment oil pressure controller (static)					
C_ABC_MAX_POIL_CTL_STAT	1	1...FFH	1...255	1	[-]
Debounce counter maximum value pressure controller (static)					
C_ABC_INC_POIL_CTL_DYN	1	0...FFH	0...255	1	[-]
Debounce counter increment oil pressure controller (dynamic)					
C_ABC_MAX_POIL_CTL_DYN	1	1...FFH	1...255	1	[-]
Debounce counter maximum value pressure controller (dynamic)					
C_ABC_INC_PUT_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment PUT_PLAUS diagnosis					
C_ABC_MAX_PUT_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value PUT_PLAUS diagnosis					
C_ABC_INC_MAP_PLAUS	1	0...FFH	0...255	1	[-]
Increment of the MAP Plausibility anti-bounce counter					
C_ABC_MAX_MAP_PLAUS	1	1...FFH	1...255	1	[-]
Threshold to be reached, before permanently activating MAP Plausibility error					
C_ABC_INC_TQ_CST	1	0...FFH	0...255	1	[-]
Debounce counter increment TQ_CST diagnosis					
C_ABC_MAX_TQ_CST	1	1...FFH	1...255	1	[-]
Debounce counter maximum value TQ_CST diagnosis					
C_ABC_INC_POIL_CTL_MEC	1	0...FFH	0...255	1	[-]
Debounce counter increment Oilpressure valve diagnosis					
C_ABC_MAX_POIL_CTL_MEC	1	1...FFH	1...255	1	[-]
Debounce counter maximum value Oilpressure valve diagnosis					
C_ABC_INC_POIL_SENS_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment Oilpressure sensor diagnosis					
C_ABC_MAX_POIL_SENS_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value Oilpressure sensor diagnosis					
C_ABC_INC_NEUT_PSN_GB_LRN	1	0...FFH	0...255	1	[-]
Debounce counter increment Neutral gear learning diagnosis					
C_ABC_MAX_NEUT_PSN_GB_LRN	1	1...FFH	1...255	1	[-]
Debounce counter maximum value Neutral gear learning diagnosis					
C_ABC_INC_POIL_PUMP	1	0...FFH	0...255	1	[-]
Debounce counter increment Oelpump diagnosis					
C_ABC_MAX_POIL_PUMP	1	1...FFH	1...255	1	[-]
Debounce counter maximum value Oelpump diagnosis					
C_ABC_INC_POIL_SYS	1	0...FFH	0...255	1	[-]
Debounce counter increment Oelsystem diagnosis					
C_ABC_MAX_POIL_SYS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value Oelsystem diagnosis					
C_ABC_INC_GEN_EL	1	0...FFH	0...255	1	[-]
Debounce counter increment GEN_EL diagnosis					
C_ABC_MAX_GEN_EL	1	1...FFH	1...255	1	[-]
Debounce counter maximum value GEN_EL diagnosis					
C_ABC_INC_GEN_MEC	1	0...FFH	0...255	1	[-]
Debounce counter increment GEN_MEC diagnosis					

C_ABC_MAX_GEN_MEC	1	1...FFH	1...255	1	[-]
Debounce counter maximum value GEN_MEC diagnosis					
C_ABC_INC_GEN_H_TEMP	1	0...FFH	0...255	1	[-]
Debounce counter increment GEN_H_TEMP diagnosis					
C_ABC_MAX_GEN_H_TEMP	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - GEN_H_TEMP diagnosis					
C_ABC_INC_GEN_TYP_NOT_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment GEN_TYP_NOT_PLAUS diagnosis					
C_ABC_MAX_GEN_TYP_NOT_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value GEN_TYP_NOT_PLAUS diagnosis					
C_ABC_INC_GEN_MSG_LOST	1	0...FFH	0...255	1	[-]
Debounce counter increment GEN_MSG_LOST diagnosis					
C_ABC_MAX_GEN_MSG_LOST	1	1...FFH	1...255	1	[-]
Debounce counter maximum value GEN_MSG_LOST diagnosis					
C_ABC_INC_GEN_CLC_V_NOT_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment GEN_CLC_V_NOT_PLAUS diagnosis					
C_ABC_MAX_GEN_CLC_V_NOT_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value GEN_CLC_V_NOT_PLAUS diagnosis					
C_ABC_INC_GEN_H_TEMP_CLC	1	0...FFH	0...255	1	[-]
Debounce counter increment GEN_H_TEMP_CLC diagnosis					
C_ABC_MAX_GEN_H_TEMP_CLC	1	1...FFH	1...255	1	[-]
Debounce counter maximum value GEN_H_TEMP_CLC diagnosis					
C_ABC_INC_GEN_CTL_NOT_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment GEN_CTL_NOT_PLAUS diagnosis					
C_ABC_MAX_GEN_CTL_NOT_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value GEN_CTL_NOT_PLAUS diagnosis					
C_ABC_INC_EGRV_PSN_PLAUS	1	0...FFH	0...255	1	[-]
Antibounce counter increment for exhaust gas recirculation valve position plausibility					
C_ABC_MAX_EGRV_PSN_PLAUS	1	1...FFH	1...255	1	[-]
Maximal value of antibounce counter for exhaust gas recirculation valve position plasuibility					
C_ABC_INC_TCHA_PRS_CTL	1	0...FFH	0...255	1	[-]
Debounce counter increment charger pressure controller diagnosis					
C_ABC_MAX_TCHA_PRS_CTL	1	1...FFH	1...255	1	[-]
Debounce counter maximum charger pressure contoller diagnosis					
C_ABC_INC_MEC_IVVT_IN	1	0...FFH	0...255	1	[-]
Debounce counter increment - MEC_IVVT_IN diagnosis					
C_ABC_MAX_MEC_IVVT_IN	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - MEC_IVVT_IN diagnosis					
C_ABC_INC_MEC_IVVT_EX	1	0...FFH	0...255	1	[-]
Debounce counter increment - MEC_IVVT_EX diagnosis					
C_ABC_MAX_MEC_IVVT_EX	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - MEC_IVVT_EX diagnosis					
C_ABC_INC_N_MAX_HOMS	1	0...FFH	0...255	1	[-]
Debounce counter engine overspeed for changing the combustion mode diagnosis					
C_ABC_MAX_N_MAX_HOMS	1	1...FFH	1...255	1	[-]
Debounce counter maximum cengine overspeed for changing the combustion mode diagnosis					
C_ABC_INC_MAP_TPS_PLAUS	1	0...FFH	0...255	1	[-]
Debounce counter increment MAP ratio plausibility diagnosis					
C_ABC_MAX_MAP_TPS_PLAUS	1	1...FFH	1...255	1	[-]
Debounce counter maximum MAP ratio plausibility diagnosis					
C_ABC_INC_MAF_LAMB_MAX	1	0...FFH	0...255	1	[-]
Anti-bounce increment value for mass air flow calculated from lambda					
C_ABC_MAX_MAF_LAMB_MAX	1	1...FFH	1...255	1	[-]
Maximal value of antibounce counter for maximum of mass air flow calculated from lambda					
C_ABC_INC_ER_STRAT	1	0...FFH	0...255	1	[-]
Anti-bounce increment value for engine roughness improvement at stratified mode					
C_ABC_MAX_ER_STRAT	1	1...FFH	1...255	1	[-]
Maximal value of antibounce counter for engine roughness improvement at stratified mode					
C_ABC_INC_ER_STRAT_WUP	1	0...FFH	0...255	1	[-]
Anti-bounce increment value for engine roughness improvement at stratified mode at warmup					

C_ABC_MAX_ER_STRAT_WUP	1	1...FFH	1...255	1	[-]
Maximal value of antibounce counter for engine roughness improvement at stratified mode at warmup					
C_ABC_INC_N_MAX_DRIV_REQ	1	0...FFH	0...255	1	[-]
Anti-bounce increment value for too high engine rotation speed at vehicle stop					
C_ABC_MAX_N_MAX_DRIV_REQ	1	1...FFH	1...255	1	[-]
Maximal value of antibounce counter for too high engine rotation speed at vehicle stop					

27.4 Emmision related variable definition

27.4.1 Calculation of minimum / maximum downstream lambda sensor voltage for service $06

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_DOWN_MIN_DC[NC_CBK_EX_NR]	O/V	0...3FFH	0...4.99511	4.8828e-3	[V]
Min Voltage Value in a DC					
VLS_DOWN_MAX_DC[NC_CBK_EX_NR]	O/V	0...3FFH	0...4.99511	4.8828e-3	[V]
Max Voltage Value in a DC					

Input data:

VLS_DOWN[NC_CBK_EX_NR]	LV_DC	LV_END_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR]	LV_END_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

VLS_DOWN_MIN/MAX_DC[NC_CBK_EX_NR] are used for OBD communication (MODE 06h), indicating the maximum/minimum downstream voltage to check if sensors are not aged or damaged. Once the electical diagnosis (SCP and SCG) of the downstream lambda sensors has been completed in a driving cycle, VLS_DOWN_MIN/MAX_DC [NC_CBK_EX_NR] are updated respectively by VLS_DOWN[NC_CBK_EX_NR].

Application conditions:

Initialisation: VLS_DOWN_MIN_DC[NC_CBK_EX_NR = 0.425[V]
VLS_DOWN_MAX_DC[NC_CBK_EX_NR = 0.425[V]
at reset **or** clearing FMY **or** LV_DC 0->1

Recurrence: 100ms

Activation: LV_DC = 1

Formula section:

```
IF      LV_END_DIAG_SCG_LS_DOWN[NC_CBK_EX_NR] =1    AND
         VLS_DOWN[NC_CBK_EX_NR] < VLS_DOWN_MIN_DC[NC_CBK_EX_NR]
THEN    VLS_DOWN_MIN_DC[NC_CBK_EX_NR] = VLS_DOWN[NC_CBK_EX_NR]
ENDIF
IF      LV_END_DIAG_SCP_LS_DOWN[NC_CBK_EX_NR] =1 AND
         VLS_DOWN[NC_CBK_EX_NR]  > VLS_DOWN_MAX_DC[NC_CBK_EX_NR]
THEN    VLS_DOWN_MAX_DC[NC_CBK_EX_NR] = VLS_DOWN[NC_CBK_EX_NR]
ENDIF
```

27.5 Table of failures

FUNCTION DESCRIPTION:

General information:

This specification is automatic generated out from SW and Dataset and consits of two parts:

- **Part 1:** Configuration of Readiness class for each diagnosis instance.
- **Part 2:** Listing of all implemented diagnosis instances (LV_ERR_xx) including all calibration data (proposal) like error class (C_ERR_CLAS_xx), Environmental data (ID_ERR_ENVD_xx), Trouble codes (ID_ERR_DTC_xx).

  Remark:
  All diagnosis instances marked with grey colour do have a missing calibration !

27.5.1 Configuration part

Description:

For generation of the readiness-status of the OBD system in MODE 01h (only done if the diagnosis instance is calibrated as emission relevant / visible for SCAN – tool => LC_OBD_ERR = 1 / C_ERR_CLAS_xx , see “Failure classes”) all diagnosis must be clasified as:

- CARB_CC - Comprehensive component, continuous test
- CARB_MIS - Misfire monitoring, continuous test
- CARB_FSD - Fuel system monitoring, continuous test
- CARB_CAT - Catalyst monitoring, sequential test
- CARB_EVAP - Evaporative system monitoring, sequential test
- CARB_SA - Secondary air monitoring
- CARB_LS - Oxygen sensor monitoring
- CARB_LSH - Oxygen sensor heater monitoring
- CARB_EGR - EGR system monitoring, sequential test

Formula section:

All diagnosis instances are configured as **CARB_CC**,

ecxept the list below

Diagnostic Instance Name(xx)	CARB clasification for Readiness
MIS_0 MIS_1 MIS_2 MIS_3 MIS_4 MIS_5 MIS_FTL_L MIS_MPL	CARB_MIS
DELTA_I_LAM_1 DELTA_I_LAM_2 FSD_1 FSD_2 FSD_LAM_LIM_1 FSD_LAM_LIM_2	CARB_FSD
CAT_DIAG_SUM_1 CAT_DIAG_SUM_2 CAT_DIAG_AFL_1 CAT_DIAG_AFL_2 CAT_DIAG_1 CAT_DIAG_2	CARB_CAT
DIAGCPS DMTL_PUMP DMTLH DMTLM DMTLS SMALL_LEAK ROUGH_LEAK	CARB_EVAP
not supported	CARB_SA
AIR_LSL_UP_1 AIR_LSL_UP_2 CHK_LS_DOWN_1 CHK_LS_DOWN_2 DYN_VLD_LS_UP_1 DYN_VLD_LS_UP_2 PUC_VLD_LS_UP_1 PUC_VLD_LS_UP_2 PUE_LS_DOWN_1 PUE_LS_DOWN_2 SHIFT_AFL_LSL_UP_1 SHIFT_AFL_LSL_UP_2 SHIFT_AFR_LSL_UP_1 SHIFT_AFR_LSL_UP_2 SWT_LS_DOWN_1 SWT_LS_DOWN_2 VLS_DOWN_DIF_1 VLS_DOWN_DIF_2 CHG_LS_DOWN	CARB_LS
OBD_LSH_DOWN_1 OBD_LSH_DOWN_2 OBD_VLD_LSH_UP_1 OBD_VLD_LSH_UP_2	CARB_LSH
not supported	CARB_EGR

27.5.2 Calibration part (sorted by alphabetic order)

Made from dataset file 393IBGAB_4DC3930S.hex

Diagnostic Instance	Diagnostic Trouble Code		Failure class	Environmental Data	
Name(xx)	DTC	Nr.			Nr.
ACK_IGK_OFF		SYM0	0x11	0x11	0
		SYM1		0x0C	1
		SYM2		0x6A	2
		SYM3		0x23	3
		GLOBAL			
	11170	DTC			
ACR_AD	0x3045	SYM0	0x01	0xFC	0
	0x3046	SYM1		0xF7	1
	0x3048	SYM2		0x3C	2
	0x3047	SYM3		0xFD	3
		GLOBAL			
	10767	DTC			
ACR_CTL	0x3043	SYM0	0x01	0xFC	0
	0x3044	SYM1		0xF7	1
		SYM2		0x3C	2
		SYM3		0xFD	3
		GLOBAL			
	10766	DTC			
ACR_DR	0x3042	SYM0	0x01	0xFC	0
		SYM1		0xF7	1
		SYM2		0x3C	2
		SYM3		0xFD	3
		GLOBAL			
	10765	DTC			
AEB_0	0x0A16	SYM0	0x11	0x8B	0
	0x0A15	SYM1		0x0C	1
	0x0A14	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12203	DTC			
AEB_1	0x0AB8	SYM0	0x00	0x8B	0
	0x0AB7	SYM1		0x0C	1
	0x0AB6	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12204	DTC			
AIR_LSL_UP_1	0x2414	SYM0	0x56	0x8B	0
		SYM1		0x49	1
		SYM2		0x45	2
		SYM3		0x8C	3
		GLOBAL			
	11323	DTC			
AIR_LSL_UP_2	0x2415	SYM0	0x56	0x8B	0

		SYM1		0x4B	1
		SYM2		0x48	2
		SYM3		0x8F	3
		GLOBAL			
	11324	DTC			
ALTER_BN	0x160C	SYM0	0x11	0x11	0
	0x160D	SYM1		0x6A	1
	0x160E	SYM2		0x98	2
		SYM3		0x3C	3
		GLOBAL			
	11755	DTC			
ALTER_BN_BAT		SYM0	0x11	0x68	0
	0x160A	SYM1		0x69	1
		SYM2		0x6A	2
	0x160B	SYM3		0xA8	3
		GLOBAL			
	11756	DTC			
ALTER_BN_RGN		SYM0	0x11	0x6B	0
		SYM1		0x6C	1
		SYM2		0x6E	2
	0x160F	SYM3		0x3C	3
		GLOBAL			
	11757	DTC			
ALTER_COM		SYM0	0x11	0x8B	0
		SYM1		0x35	1
	0xD132	SYM2		0x42	2
		SYM3		0x15	3
		GLOBAL			
	11928	DTC			
AMP	0x2229	SYM0	0x56	0x21	0
	0x2228	SYM1		0x34	1
		SYM2		0x70	2
		SYM3		0x7C	3
		GLOBAL			
	12150	DTC			
AMP_PLAUS	0x321E	SYM0	0x56	0x34	0
	0x321F	SYM1		0x70	1
		SYM2		0x33	2
		SYM3		0x24	3
		GLOBAL			
	12151	DTC			
AMP_PLAUS_CUS		SYM0	0x11	0x34	0
		SYM1		0x0B	1
		SYM2		0xDD	2
		SYM3		0xDE	3
		GLOBAL			
	12153	DTC			
BAT_SENS	0x150A	SYM0	0x11	0x7C	0
		SYM1		0x24	1

	0x150B	SYM2		0x6A	2
	0x150C	SYM3		0x3C	3
		GLOBAL			
	11915	DTC			
BAT_SENS_IT	0x150D	SYM0	0x11	0x7C	0
		SYM1		0x24	1
	0x150E	SYM2		0x6A	2
	0x150F	SYM3		0x3C	3
		GLOBAL			
	11916	DTC			
BAT_SENS_IT_EL	0x151A	SYM0	0x11	0x7C	0
		SYM1		0x24	1
	0x151C	SYM2		0x6A	2
	0x151B	SYM3		0x3C	3
		GLOBAL			
	11917	DTC			
BN_ACC		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD12C	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52649	DTC			
BN_ANG_PSTE		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD120	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52641	DTC			
BN_ARS		SYM0	0x11	0x11	0
	0xD123	SYM1		0x21	1
	0xD124	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52644	DTC			
BN_CAS		SYM0	0x11	0x11	0
	0xD11D	SYM1		0x21	1
	0xD11E	SYM2		0x3C	2
	0xD11F	SYM3		0x7C	3
		GLOBAL			
	52640	DTC			
BN_CDN_DOOR		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD13A	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52657	DTC			
BN_DHL_CTL		SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2

		SYM3		0x7C	3
		GLOBAL			
	52670	DTC			
BN_EFP		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD128	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52646	DTC			
BN_EFP_CRASH		SYM0	0x11	0x0D	0
		SYM1		0x3C	1
	0xD12D	SYM2		0x11	2
		SYM3		0x32	3
		GLOBAL			
	52650	DTC			
BN_ETCU		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD11A	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52637	DTC			
BN_ETCU_2		SYM0	0x11	0x0D	0
		SYM1		0x3C	1
	0xD11B	SYM2		0x11	2
		SYM3		0x32	3
		GLOBAL			
	52638	DTC			
BN_ETCU_3		SYM0	0x11	0x11	0
	0xD161	SYM1		0x21	1
	0xD160	SYM2		0x3C	2
	0xD162	SYM3		0x7C	3
		GLOBAL			
	52660	DTC			
BN_ETCU_DISP		SYM0	0x11	0x0D	0
		SYM1		0x3C	1
	0xD13E	SYM2		0x11	2
		SYM3		0x32	3
		GLOBAL			
	52656	DTC			
BN_GEAR_REV		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD129	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52647	DTC			
BN_ICL		SYM0	0x56	0x11	0
	0xD12A	SYM1		0x21	1
	0xD12B	SYM2		0x3C	2
		SYM3		0x7C	3

		GLOBAL			
	52648	DTC			
BN_KM_ICL		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD11C	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52639	DTC			
BN_LDM		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD13D	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52653	DTC			
BN_LTG_HDLP_L		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD134	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52651	DTC			
BN_MSW		SYM0	0x11	0x11	0
	0xD102	SYM1		0x21	1
	0xD103	SYM2		0x3C	2
	0xD104	SYM3		0x7C	3
		GLOBAL			
	52629	DTC			
BN_PBR		SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52665	DTC			
BN_POW_GEN		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD122	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52643	DTC			
BN_POW_VB		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD121	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52642	DTC			
BN_REQ_PBR		SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			

	52666	DTC			
BN_STAT_TCT		SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52675	DTC			
BN_T_CLK		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD13C	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52654	DTC			
BN_T_ICL		SYM0	0x56	0x11	0
		SYM1		0x21	1
	0xD101	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52628	DTC			
BN_TCS		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD126	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52645	DTC			
BN_TQ_AMT		SYM0	0x00	0x11	0
	0xD111	SYM1		0x21	1
	0xD112	SYM2		0x3C	2
	0xD113	SYM3		0x7C	3
		GLOBAL			
	52634	DTC			
BN_TQ_DCC		SYM0	0x11	0x11	0
	0xD105	SYM1		0x21	1
	0xD106	SYM2		0x3C	2
	0xD107	SYM3		0x7C	3
		GLOBAL			
	52630	DTC			
BN_TQ_ETCU		SYM0	0x56	0x11	0
	0xD10E	SYM1		0x21	1
	0xD10F	SYM2		0x3C	2
	0xD110	SYM3		0x7C	3
		GLOBAL			
	52633	DTC			
BN_TQ_PBR		SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52674	DTC			

BN_TQ_PSTE_2		SYM0	0x11	0x11	0
	0xD108	SYM1		0x21	1
	0xD109	SYM2		0x3C	2
	0xD10A	SYM3		0x7C	3
		GLOBAL			
	52631	DTC			
BN_TQ_PSTE_3		SYM0	0x11	0x11	0
	0xD108	SYM1		0x21	1
	0xD109	SYM2		0x3C	2
	0xD10A	SYM3		0x7C	3
		GLOBAL			
	52659	DTC			
BN_TQ_TCS		SYM0	0x11	0x11	0
	0xD10B	SYM1		0x21	1
	0xD10C	SYM2		0x3C	2
	0xD10D	SYM3		0x7C	3
		GLOBAL			
	52632	DTC			
BN_TQ_TCT		SYM0	0x00	0x11	0
	0xD10E	SYM1		0x21	1
	0xD10F	SYM2		0x3C	2
	0xD110	SYM3		0x7C	3
		GLOBAL			
	52664	DTC			
BN_TRL		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xC137	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52655	DTC			
BN_VEH_MOD		SYM0	0x11	0x11	0
	0xD114	SYM1		0x21	1
	0xD115	SYM2		0x3C	2
	0xD116	SYM3		0x7C	3
		GLOBAL			
	52635	DTC			
BN_VS_TCS		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0xD118	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52636	DTC			
BSD		SYM0	0x11	0x11	0
		SYM1		0x3C	1
	0x16C6	SYM2		0x67	2
		SYM3		0x7C	3
		GLOBAL			
	11900	DTC			
CAN_BOFF		SYM0	0x56	0x11	0

		SYM1		0x32	1
	0x3202	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52615	DTC			
CAT_DIAG_1	0x0420	SYM0	0x56	0x11	0
	0x0420	SYM1		0x18	1
		SYM2		0x1F	2
		SYM3		0x1E	3
		GLOBAL			
	10740	DTC			
CAT_DIAG_2	0x0430	SYM0	0x56	0x11	0
	0x0430	SYM1		0x18	1
		SYM2		0x1F	2
		SYM3		0x1E	3
		GLOBAL			
	10741	DTC			
CAT_DIAG_AFL_1	0x140F	SYM0	0x01	0x0D	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10790	DTC			
CAT_DIAG_AFL_2	0x141F	SYM0	0x01	0x0D	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10791	DTC			
CAT_DIAG_SUM_1		SYM0	0x00	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10742	DTC			
CAT_DIAG_SUM_2		SYM0	0x00	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10743	DTC			
CHG_LS_DOWN		SYM0	0x56	0x1F	0
		SYM1		0x8B	1
		SYM2		0x49	2
	0x0041	SYM3		0x4B	3
		GLOBAL			
	11370	DTC			
CHG_LS_UP		SYM0	0x56	0x05	0
		SYM1		0x8B	1

		SYM2		0x45	2
	0x0040	SYM3		0x48	3
		GLOBAL			
	11300	DTC			
CHK_FUC	0x0455	SYM0	0x01	0x3B	0
		SYM1		0x59	1
		SYM2		0x5B	2
		SYM3		0x8D	3
		GLOBAL			
	10779	DTC			
CHK_LS_DOWN_1	0x2271	SYM0	0x56	0x45	0
	0x2270	SYM1		0x5C	1
		SYM2		0x11	2
		SYM3		0x49	3
		GLOBAL			
	11371	DTC			
CHK_LS_DOWN_2	0x2273	SYM0	0x56	0x48	0
	0x2272	SYM1		0x5D	1
		SYM2		0x11	2
		SYM3		0x4B	3
		GLOBAL			
	11372	DTC			
CONV_MON_1		SYM0	0x88	0xB8	0
		SYM1		0x47	1
		SYM2		0x54	2
	0x1618	SYM3		0x3C	3
		GLOBAL			
	11612	DTC			
CRK_PLAUS	0x0335	SYM0	0x58	0x11	0
	0x0335	SYM1		0x32	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10900	DTC			
CRK_SYN	0x0373	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10901	DTC			
CRK_TOOTH	0x0370	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10902	DTC			
CRK_TOOTH_PER	0x0370	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x7C	2

		SYM3		0x3C	3
		GLOBAL			
	10903	DTC			
CRU_INH_MON_1	0x16A6	SYM0	0x11	0xC4	0
	0x16B4	SYM1		0x0D	1
	0x166A	SYM2		0xB7	2
		SYM3		0x81	3
		GLOBAL			
	11600	DTC			
CS	0x0832	SYM0	0x11	0x11	0
	0x0831	SYM1		0x0D	1
		SYM2		0x32	2
		SYM3		0x18	3
		GLOBAL			
	12135	DTC			
CTL_LSL_UP_1	0x2627	SYM0	0x00	0x45	0
	0x2628	SYM1		0x71	1
	0x2246	SYM2		0x8C	2
		SYM3		0x01	3
		GLOBAL			
	11333	DTC			
CTL_LSL_UP_2	0x2630	SYM0	0x00	0x48	0
	0x2631	SYM1		0x73	1
	0x2250	SYM2		0x8F	2
		SYM3		0x01	3
		GLOBAL			
	11334	DTC			
CWP_COM		SYM0	0x11	0x11	0
		SYM1		0x05	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	11908	DTC			
CWP_COM_PLAUS		SYM0	0x11	0x11	0
		SYM1		0x05	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	11909	DTC			
CWP_INT_OFF		SYM0	0x21	0x05	0
		SYM1		0xE9	1
		SYM2		0xEC	2
		SYM3		0xED	3
		GLOBAL			
	11906	DTC			
CWP_PLAUS		SYM0	0x11	0x05	0
		SYM1		0xE9	1
		SYM2		0xEA	2
		SYM3		0xEB	3

		GLOBAL			
	11905	DTC			
CWP_PWR		SYM0	0x11	0x05	0
		SYM1		0xE9	1
		SYM2		0xEC	2
		SYM3		0xEE	3
		GLOBAL			
	11907	DTC			
CYL_BAL_ER_0	0x029B	SYM0	0x11	0x0D	0
	0x029A	SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12400	DTC			
CYL_BAL_ER_1	0x02AB	SYM0	0x11	0x0D	0
	0x02AA	SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12404	DTC			
CYL_BAL_ER_2	0x02A3	SYM0	0x11	0x0D	0
	0x02A2	SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12402	DTC			
CYL_BAL_ER_3	0x02AF	SYM0	0x11	0x0D	0
	0x02AE	SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12405	DTC			
CYL_BAL_ER_4	0x029F	SYM0	0x11	0x0D	0
	0x029E	SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12401	DTC			
CYL_BAL_ER_5	0x02A7	SYM0	0x11	0x0D	0
	0x02A6	SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12403	DTC			
CYL_BAL_LAM_0		SYM0	0x11	0x0D	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			

	12412	DTC			
CYL_BAL_LAM_1		SYM0	0x11	0x0D	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12416	DTC			
CYL_BAL_LAM_2		SYM0	0x11	0x0D	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12414	DTC			
CYL_BAL_LAM_3		SYM0	0x11	0x0D	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12417	DTC			
CYL_BAL_LAM_4		SYM0	0x11	0x0D	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12413	DTC			
CYL_BAL_LAM_5		SYM0	0x11	0x0D	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12415	DTC			
DCC		SYM0	0x11	0x0D	0
		SYM1		0x11	1
		SYM2		0x32	2
		SYM3		0x7C	3
		GLOBAL			
	11710	DTC			
DELTA_I_LAM_1	0x2097	SYM0	0x76	0x49	0
	0x2096	SYM1		0x45	1
		SYM2		0x78	2
		SYM3		0xF5	3
		GLOBAL			
	11313	DTC			
DELTA_I_LAM_2	0x2099	SYM0	0x76	0x4B	0
	0x2098	SYM1		0x48	1
		SYM2		0x79	2
		SYM3		0xF6	3
		GLOBAL			
	11314	DTC			

DIAGCPS		SYM0	0x56	0x1F	0
		SYM1		0x18	1
	0x0441	SYM2		0x11	2
	0x0440	SYM3		0x4D	3
		GLOBAL			
	10778	DTC			
DMTL_PUMP	0x2402	SYM0	0x56	0x34	0
	0x2401	SYM1		0x74	1
	0x2400	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10771	DTC			
DMTLH	0x240C	SYM0	0x56	0x34	0
	0x240B	SYM1		0x74	1
	0x240A	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10776	DTC			
DMTLM	0x1449	SYM0	0x56	0x3B	0
	0x1448	SYM1		0x59	1
	0x1434	SYM2		0x67	2
	0x1447	SYM3		0x24	3
		GLOBAL			
	10775	DTC			
DMTLS	0x2420	SYM0	0x56	0x34	0
	0x2419	SYM1		0x74	1
	0x2418	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10770	DTC			
DUR_IGC_MPL		SYM0	0x11	0x05	0
		SYM1		0x3C	1
		SYM2		0x11	2
	0x1383	SYM3		0x32	3
		GLOBAL			
	11895	DTC			
DYN_VLD_LS_UP_1	0x0133	SYM0	0x56	0x71	0
		SYM1		0x45	1
		SYM2		0x30	2
		SYM3		0x8C	3
		GLOBAL			
	11321	DTC			
DYN_VLD_LS_UP_2	0x0153	SYM0	0x56	0x73	0
		SYM1		0x48	1
		SYM2		0x31	2
		SYM3		0x8F	3
		GLOBAL			
	11322	DTC			
EBOX_CFA	0x3226	SYM0	0x11	0x11	0

	0x3227	SYM1		0x1E	1
	0x3228	SYM2		0x21	2
		SYM3		0x0D	3
		GLOBAL			
	12145	DTC			
ECF_EL_1	0x0692	SYM0	0x11	0x20	0
	0x0691	SYM1		0x7F	1
	0x0480	SYM2		0x32	2
		SYM3		0x3C	3
		GLOBAL			
	12030	DTC			
ECFPWM_FB_1		SYM0	0x11	0x24	0
		SYM1		0x7F	1
		SYM2		0x3C	2
	0x14C0	SYM3		0x20	3
		GLOBAL			
	12031	DTC			
ECRAS_DOWN_FB	0x14C6	SYM0	0x11	0x3C	0
		SYM1		0x24	1
		SYM2		0xD2	2
		SYM3		0xD3	3
		GLOBAL			
	12048	DTC			
ECRAS_EL	0x300A	SYM0	0x11	0x11	0
	0x300B	SYM1		0x0D	1
	0x300C	SYM2		0x3C	2
		SYM3		0x80	3
		GLOBAL			
	12045	DTC			
ECRAS_UP_FB	0x14C5	SYM0	0x11	0x3C	0
	0x14C4	SYM1		0x24	1
		SYM2		0xD2	2
	0xD130	SYM3		0xD3	3
		GLOBAL			
	12049	DTC			
ECT_EL		SYM0	0x56	0x1F	0
	0x0598	SYM1		0x20	1
		SYM2		0x32	2
		SYM3		0x3C	3
		GLOBAL			
	12021	DTC			
ECT_EL_OC		SYM0	0x11	0x1F	0
		SYM1		0x20	1
	0x0597	SYM2		0x32	2
		SYM3		0x3C	3
		GLOBAL			
	12023	DTC			
ECT_EL_SCP	0x0599	SYM0	0x11	0x1F	0
		SYM1		0x20	1

		SYM2		0x32	2
		SYM3		0x3C	3
		GLOBAL			
	12024	DTC			
ECU_CKS	0x16A0	SYM0	0x99	0x11	0
	0x16A1	SYM1		0x21	1
	0x16A2	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10931	DTC			
ECU_NVMY	0x16A3	SYM0	0x56	0x11	0
		SYM1		0x21	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10932	DTC			
ECU_RAM	0x0604	SYM0	0x56	0x11	0
	0x0604	SYM1		0x21	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10930	DTC			
EF	0x0478	SYM0	0x11	0x0D	0
	0x0477	SYM1		0x8B	1
	0x0475	SYM2		0xAD	2
		SYM3		0x3C	3
		GLOBAL			
	12140	DTC			
EFP	0x1214	SYM0	0x11	0x32	0
	0x1215	SYM1		0x3C	1
	0x1216	SYM2		0x7C	2
	0x1217	SYM3		0xAF	3
		GLOBAL			
	10926	DTC			
EFP_CRASH		SYM0	0x11	0x32	0
		SYM1		0x3C	1
	0x213F	SYM2		0x7C	2
		SYM3		0xAF	3
		GLOBAL			
	10925	DTC			
EFPPWM_PLAUS		SYM0	0x11	0xF3	0
		SYM1		0x11	1
		SYM2		0x3C	2
		SYM3		0xBA	3
		GLOBAL			
	10927	DTC			
EGRV_PSN_PLAUS		SYM0	0x01	0x0C	0
		SYM1		0xF7	1
		SYM2		0x0B	2

		SYM3		0xE2	3
		GLOBAL			
	10764	DTC			
EGY_MIN		SYM0	0x11	0x0D	0
		SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0x32	3
		GLOBAL			
	12230	DTC			
EGY_MIN_2		SYM0	0x11	0x11	0
	0x15A9	SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12231	DTC			
EL_CPS	0x0459	SYM0	0x56	0x11	0
	0x0458	SYM1		0x32	1
	0x0444	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10777	DTC			
EL_LSL_UP_1	0x0132	SYM0	0x56	0x37	0
	0x0131	SYM1		0x15	1
		SYM2		0x45	2
		SYM3		0x27	3
		GLOBAL			
	11327	DTC			
EL_LSL_UP_2	0x0152	SYM0	0x56	0x38	0
	0x0151	SYM1		0x15	1
		SYM2		0x48	2
		SYM3		0x28	3
		GLOBAL			
	11328	DTC			
ER_STRAT		SYM0	0x00	0x1F	0
		SYM1		0x11	1
		SYM2		0xD1	2
		SYM3		0x0D	3
		GLOBAL			
	12548	DTC			
ER_STRAT_WUP		SYM0	0x00	0x1F	0
		SYM1		0x11	1
		SYM2		0xD1	2
		SYM3		0x0D	3
		GLOBAL			
	12549	DTC			
FL_LS_DOWN_1		SYM0	0x00	0x96	0
		SYM1		0x49	1
		SYM2		0x16	2
	0x1204	SYM3		0x45	3

		GLOBAL			
	11375	DTC			
FL_LS_DOWN_2		SYM0	0x00	0x97	0
		SYM1		0x4B	1
		SYM2		0x16	2
	0x1205	SYM3		0x48	3
		GLOBAL			
	11376	DTC			
FSD_1		SYM0	0x76	0x0C	0
		SYM1		0x0B	1
	0x0171	SYM2		0x13	2
	0x0172	SYM3		0xE0	3
		GLOBAL			
	10720	DTC			
FSD_2		SYM0	0x76	0x0C	0
		SYM1		0x0B	1
	0x0174	SYM2		0x13	2
	0x0175	SYM3		0xE0	3
		GLOBAL			
	10721	DTC			
FSD_H_RNG_1	0x2192	SYM0	0x01	0x0C	0
	0x2191	SYM1		0x0B	1
		SYM2		0x13	2
		SYM3		0xE0	3
		GLOBAL			
	10725	DTC			
FSD_H_RNG_2	0x2194	SYM0	0x01	0x0C	0
	0x2193	SYM1		0x0B	1
		SYM2		0x13	2
		SYM3		0xE0	3
		GLOBAL			
	10726	DTC			
FSD_LAM_LIM_1	0x0171	SYM0	0x76	0x0C	0
	0x0172	SYM1		0x0B	1
		SYM2		0x13	2
		SYM3		0xE0	3
		GLOBAL			
	10795	DTC			
FSD_LAM_LIM_2	0x0174	SYM0	0x76	0x0C	0
	0x0175	SYM1		0x0B	1
		SYM2		0x13	2
		SYM3		0xE0	3
		GLOBAL			
	10796	DTC			
FTL_LE_CAN	0x0463	SYM0	0x33	0x0E	0
	0x0462	SYM1		0x16	1
		SYM2		0x39	2
	0x1407	SYM3		0x61	3
		GLOBAL			

	11748	DTC			
FTL_MIN	0x142E	SYM0	0x56	0x1F	0
	0x142F	SYM1		0x18	1
		SYM2		0x11	2
	0x140E	SYM3		0x3B	3
		GLOBAL			
	10716	DTC			
FTL_OBD		SYM0	0x33	0x0E	0
		SYM1		0x16	1
		SYM2		0x39	2
	0x144B	SYM3		0x61	3
		GLOBAL			
	10786	DTC			
FTL_RI_CAN	0x2068	SYM0	0x33	0x0E	0
	0x2067	SYM1		0x16	1
		SYM2		0x39	2
	0x1408	SYM3		0x61	3
		GLOBAL			
	11749	DTC			
FUP	0x0190	SYM0	0x58	0x11	0
	0x0192	SYM1		0xF0	1
		SYM2		0xF4	2
		SYM3		0xEF	3
		GLOBAL			
	10722	DTC			
FUP_EFP	0x2542	SYM0	0x11	0x11	0
	0x2541	SYM1		0xF4	1
		SYM2		0xF3	2
		SYM3		0x3B	3
		GLOBAL			
	10739	DTC			
FUP_EFP_NOT_PLAUS		SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10793	DTC			
FUP_MFP_PLAUS	0x3283	SYM0	0x58	0xF0	0
	0x3284	SYM1		0xF2	1
		SYM2		0x55	2
		SYM3		0x56	3
		GLOBAL			
	10737	DTC			
FUP_ORNG		SYM0	0x22	0xF0	0
		SYM1		0x11	1
		SYM2		0xF2	2
		SYM3		0xBA	3
		GLOBAL			
	12224	DTC			

FUP_ST		SYM0	0x11	0xF0	0
		SYM1		0xF2	1
		SYM2		0x3C	2
		SYM3		0xF3	3
		GLOBAL			
	12223	DTC			
FUP_STOP		SYM0	0x11	0xF0	0
		SYM1		0xF2	1
		SYM2		0x11	2
		SYM3		0x32	3
		GLOBAL			
	12222	DTC			
GEN	0x0A3B	SYM0	0x11	0x44	0
	0x324A	SYM1		0x87	1
	0x0620	SYM2		0x98	2
	0x3223	SYM3		0x15	3
		GLOBAL			
	11927	DTC			
GEN_CLC_V_NOT_PLAUS	0x325A	SYM0	0x11	0x57	0
		SYM1		0x87	1
		SYM2		0x98	2
		SYM3		0x15	3
		GLOBAL			
	11982	DTC			
GEN_CTL_NOT_PLAUS	0x324E	SYM0	0x11	0x72	0
		SYM1		0x35	1
		SYM2		0x42	2
		SYM3		0x15	3
		GLOBAL			
	11986	DTC			
GEN_DIAG	0x3255	SYM0	0x11	0x8B	0
		SYM1		0x32	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	11926	DTC			
GEN_EL	0x0620	SYM0	0x11	0x57	0
		SYM1		0x87	1
		SYM2		0x98	2
		SYM3		0x15	3
		GLOBAL			
	11981	DTC			
GEN_H_TEMP	0x0A3B	SYM0	0x11	0x44	0
		SYM1		0x87	1
		SYM2		0x57	2
		SYM3		0x13	3
		GLOBAL			
	11983	DTC			
GEN_H_TEMP_CLC	0x324C	SYM0	0x11	0x44	0

		SYM1		0x87	1
		SYM2		0x57	2
		SYM3		0x13	3
		GLOBAL			
	11984	DTC			
GEN_MEC	0x3223	SYM0	0x11	0x8B	0
		SYM1		0x87	1
		SYM2		0x98	2
		SYM3		0x15	3
		GLOBAL			
	11985	DTC			
GEN_MSG_LOST	0xD132	SYM0	0x11	0x8B	0
		SYM1		0x35	1
		SYM2		0x42	2
		SYM3		0x15	3
		GLOBAL			
	11980	DTC			
GEN_TYP_NOT_PLAUS	0x324A	SYM0	0x11	0x8B	0
		SYM1		0x35	1
		SYM2		0x42	2
		SYM3		0x15	3
		GLOBAL			
	11987	DTC			
GS		SYM0	0x01	0x32	0
		SYM1		0x81	1
		SYM2		0x7C	2
	0x0700	SYM3		0x3C	3
		GLOBAL			
	10960	DTC			
H_PRS_SYS	0x3003	SYM0	0x58	0xF0	0
		SYM1		0x11	1
	0x3090	SYM2		0xF2	2
		SYM3		0xBA	3
		GLOBAL			
	10738	DTC			
IGC_SCG_0		SYM0	0x11	0x05	0
	0x1301	SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB1	3
		GLOBAL			
	11800	DTC			
IGC_SCG_1		SYM0	0x11	0x05	0
	0x1305	SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB2	3
		GLOBAL			
	11804	DTC			
IGC_SCG_2		SYM0	0x11	0x05	0
	0x1303	SYM1		0x3C	1

		SYM2		0x11	2
		SYM3		0xB3	3
		GLOBAL			
	11802	DTC			
IGC_SCG_3		SYM0	0x11	0x05	0
	0x1306	SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB4	3
		GLOBAL			
	11805	DTC			
IGC_SCG_4		SYM0	0x11	0x05	0
	0x1302	SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB5	3
		GLOBAL			
	11801	DTC			
IGC_SCG_5		SYM0	0x11	0x05	0
	0x1304	SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB6	3
		GLOBAL			
	11803	DTC			
IGC_SCP_0	0x2301	SYM0	0x11	0x05	0
		SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB1	3
		GLOBAL			
	12448	DTC			
IGC_SCP_1	0x2313	SYM0	0x11	0x05	0
		SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB2	3
		GLOBAL			
	12452	DTC			
IGC_SCP_2	0x2307	SYM0	0x11	0x05	0
		SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB3	3
		GLOBAL			
	12450	DTC			
IGC_SCP_3	0x2316	SYM0	0x11	0x05	0
		SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB4	3
		GLOBAL			
	12453	DTC			
IGC_SCP_4	0x2304	SYM0	0x11	0x05	0
		SYM1		0x3C	1
		SYM2		0x11	2

		SYM3		0xB5	3
		GLOBAL			
	12449	DTC			
IGC_SCP_5	0x2310	SYM0	0x11	0x05	0
		SYM1		0x3C	1
		SYM2		0x11	2
		SYM3		0xB6	3
		GLOBAL			
	12451	DTC			
IM_BLS_PLAUS		SYM0	0x11	0xCE	0
		SYM1		0xB7	1
		SYM2		0x7C	2
	0x0571	SYM3		0x4A	3
		GLOBAL			
	12131	DTC			
IM_BTS_PLAUS		SYM0	0x11	0xCE	0
		SYM1		0xB7	1
		SYM2		0x7C	2
	0x0703	SYM3		0x4A	3
		GLOBAL			
	12132	DTC			
IMOB_0		SYM0	0x11	0x11	0
	0x1667	SYM1		0x3C	1
		SYM2		0x7C	2
		SYM3		0x21	3
		GLOBAL			
	12105	DTC			
IMOB_1	0x165A	SYM0	0x11	0x11	0
	0x1660	SYM1		0x3C	1
	0x1661	SYM2		0x7C	2
	0x165B	SYM3		0x21	3
		GLOBAL			
	12106	DTC			
IMOB_2	0x165C	SYM0	0x11	0x11	0
	0x165D	SYM1		0x3C	1
	0x1668	SYM2		0x7C	2
	0x165E	SYM3		0x21	3
		GLOBAL			
	12107	DTC			
IMOB_3		SYM0	0x11	0x11	0
	0xD166	SYM1		0x3C	1
	0xC167	SYM2		0x7C	2
		SYM3		0x21	3
		GLOBAL			
	12108	DTC			
ISC	0x0507	SYM0	0x56	0x11	0
	0x0506	SYM1		0x12	1
		SYM2		0x13	2
		SYM3		0x14	3

		GLOBAL			
	10975	DTC			
ISC_CST	0x1562	SYM0	0x56	0x11	0
	0x1561	SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10976	DTC			
IV_0	0x0201	SYM0	0x58	0x11	0
	0x310B	SYM1		0x1F	1
	0x310B	SYM2		0x32	2
	0x0201	SYM3		0x3C	3
		GLOBAL			
	11824	DTC			
IV_1	0x0205	SYM0	0x58	0x11	0
	0x312B	SYM1		0x1F	1
	0x312B	SYM2		0x32	2
	0x0205	SYM3		0x3C	3
		GLOBAL			
	11828	DTC			
IV_2	0x0203	SYM0	0x58	0x11	0
	0x311B	SYM1		0x1F	1
	0x311B	SYM2		0x32	2
	0x0203	SYM3		0x3C	3
		GLOBAL			
	11826	DTC			
IV_3	0x0206	SYM0	0x58	0x11	0
	0x312E	SYM1		0x1F	1
	0x312E	SYM2		0x32	2
	0x0206	SYM3		0x3C	3
		GLOBAL			
	11829	DTC			
IV_4	0x0202	SYM0	0x58	0x11	0
	0x310E	SYM1		0x1F	1
	0x310E	SYM2		0x32	2
	0x0202	SYM3		0x3C	3
		GLOBAL			
	11825	DTC			
IV_5	0x0204	SYM0	0x58	0x11	0
	0x311E	SYM1		0x1F	1
	0x311E	SYM2		0x32	2
	0x0204	SYM3		0x3C	3
		GLOBAL			
	11827	DTC			
IV_SC_0	0x3149	SYM0	0x11	0x11	0
	0x3102	SYM1		0x1F	1
	0x3150	SYM2		0x32	2
	0x3101	SYM3		0x3C	3
		GLOBAL			

	12460	DTC			
IV_SC_1	0x3161	SYM0	0x11	0x11	0
	0x3118	SYM1		0x1F	1
	0x3162	SYM2		0x32	2
	0x3117	SYM3		0x3C	3
		GLOBAL			
	12464	DTC			
IV_SC_2	0x3155	SYM0	0x11	0x11	0
	0x3110	SYM1		0x1F	1
	0x3156	SYM2		0x32	2
	0x3109	SYM3		0x3C	3
		GLOBAL			
	12462	DTC			
IV_SC_3	0x3164	SYM0	0x11	0x11	0
	0x3122	SYM1		0x1F	1
	0x3165	SYM2		0x32	2
	0x3121	SYM3		0x3C	3
		GLOBAL			
	12465	DTC			
IV_SC_4	0x3152	SYM0	0x11	0x11	0
	0x3106	SYM1		0x1F	1
	0x3153	SYM2		0x32	2
	0x3105	SYM3		0x3C	3
		GLOBAL			
	12461	DTC			
IV_SC_5	0x3158	SYM0	0x11	0x11	0
	0x3114	SYM1		0x1F	1
	0x3159	SYM2		0x32	2
	0x3113	SYM3		0x3C	3
		GLOBAL			
	12463	DTC			
KNK_PRE_0		SYM0	0x05	0x11	0
		SYM1		0x13	1
		SYM2		0x1E	2
		SYM3		0x1F	3
		GLOBAL			
	12525	DTC			
KNK_PRE_1		SYM0	0x05	0x11	0
		SYM1		0x13	1
		SYM2		0x1E	2
		SYM3		0x1F	3
		GLOBAL			
	12529	DTC			
KNK_PRE_2		SYM0	0x05	0x11	0
		SYM1		0x13	1
		SYM2		0x1E	2
		SYM3		0x1F	3
		GLOBAL			
	12527	DTC			

KNK_PRE_3		SYM0	0x05	0x11	0
		SYM1		0x13	1
		SYM2		0x1E	2
		SYM3		0x1F	3
		GLOBAL			
	12530	DTC			
KNK_PRE_4		SYM0	0x05	0x11	0
		SYM1		0x13	1
		SYM2		0x1E	2
		SYM3		0x1F	3
		GLOBAL			
	12526	DTC			
KNK_PRE_5		SYM0	0x05	0x11	0
		SYM1		0x13	1
		SYM2		0x1E	2
		SYM3		0x1F	3
		GLOBAL			
	12528	DTC			
KNKS_1	0x0328	SYM0	0x58	0x11	0
	0x0327	SYM1		0x12	1
		SYM2		0x83	2
	0x0326	SYM3		0x85	3
		GLOBAL			
	11880	DTC			
KNKS_2	0x1328	SYM0	0x58	0x11	0
	0x1327	SYM1		0x12	1
		SYM2		0x86	2
	0x135B	SYM3		0x88	3
		GLOBAL			
	11881	DTC			
L_PRS_SYS	0x008B	SYM0	0x11	0xF3	0
	0x3095	SYM1		0x11	1
	0x3096	SYM2		0x3C	2
		SYM3		0xBA	3
		GLOBAL			
	10797	DTC			
LAM_AD_CUS_1		SYM0	0x56	0x11	0
	0x119D	SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11892	DTC			
LAM_AD_CUS_2		SYM0	0x56	0x11	0
	0x119E	SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11893	DTC			
LDM	0x166B	SYM0	0x11	0x11	0

		SYM1		0x13	1
		SYM2		0x32	2
	0x166C	SYM3		0x91	3
		GLOBAL			
	11712	DTC			
LOAD_TPS_PLAUS	0x00BD	SYM0	0x01	0x0C	0
	0x00BC	SYM1		0x12	1
		SYM2		0x58	2
		SYM3		0x89	3
		GLOBAL			
	11526	DTC			
LOCAN_BOFF		SYM0	0x11	0x11	0
		SYM1		0x21	1
	0x3205	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	52619	DTC			
LSH_DOWN_1	0x0038	SYM0	0x56	0x96	0
	0x0037	SYM1		0x5C	1
	0x0036	SYM2		0x49	2
		SYM3		0x29	3
		GLOBAL			
	11422	DTC			
LSH_DOWN_2	0x0058	SYM0	0x56	0x97	0
	0x0057	SYM1		0x5D	1
	0x0056	SYM2		0x4B	2
		SYM3		0x2A	3
		GLOBAL			
	11423	DTC			
LSH_LSL_UP_1	0x2231	SYM0	0x00	0x15	0
		SYM1		0x45	1
		SYM2		0x3B	2
		SYM3		0x8C	3
		GLOBAL			
	11319	DTC			
LSH_LSL_UP_2	0x2234	SYM0	0x00	0x15	0
		SYM1		0x48	1
		SYM2		0x3B	2
		SYM3		0x8F	3
		GLOBAL			
	11320	DTC			
LSH_UP_1	0x0032	SYM0	0x56	0x8C	0
	0x0031	SYM1		0x8B	1
	0x0030	SYM2		0x15	2
		SYM3		0x27	3
		GLOBAL			
	11420	DTC			
LSH_UP_2	0x0052	SYM0	0x56	0x8F	0
	0x0051	SYM1		0x8B	1

	0x0050	SYM2		0x15	2
		SYM3		0x28	3
		GLOBAL			
	11421	DTC			
LSL_UP_IF_1	0x3024	SYM0	0x56	0x9B	0
	0x3022	SYM1		0x2C	1
		SYM2		0x45	2
		SYM3		0x15	3
		GLOBAL			
	11329	DTC			
LSL_UP_IF_2	0x3025	SYM0	0x56	0x9C	0
	0x3023	SYM1		0x2D	1
		SYM2		0x48	2
		SYM3		0x15	3
		GLOBAL			
	11330	DTC			
MAF	0x116C	SYM0	0x01	0xAE	0
		SYM1		0x11	1
	0x116E	SYM2		0x58	2
		SYM3		0x1E	3
		GLOBAL			
	11535	DTC			
MAF_AN	0x0103	SYM0	0x00	0x4F	0
	0x0102	SYM1		0x11	1
		SYM2		0x58	2
		SYM3		0x1E	3
		GLOBAL			
	11534	DTC			
MAF_FRQ_EL_0	0x0100	SYM0	0x01	0x12	0
	0x114F	SYM1		0x18	1
		SYM2		0x0C	2
		SYM3		0x0F	3
		GLOBAL			
	11542	DTC			
MAF_FRQ_GRD_0	0x114E	SYM0	0x00	0x12	0
		SYM1		0x18	1
		SYM2		0x0C	2
		SYM3		0x0F	3
		GLOBAL			
	11543	DTC			
MAF_FRQ_RNG_0	0x0101	SYM0	0x01	0x12	0
		SYM1		0x18	1
		SYM2		0x0C	2
		SYM3		0x0F	3
		GLOBAL			
	11541	DTC			
MAF_LAMB_MAX		SYM0	0x11	0x11	0
		SYM1		0xDD	1
		SYM2		0x06	2

		SYM3		0x07	3
		GLOBAL			
	11557	DTC			
MAP_DIP_PLAUS		SYM0	0x00	0x11	0
		SYM1		0x26	1
		SYM2		0x0B	2
		SYM3		0x13	3
		GLOBAL			
	11572	DTC			
MAP_DIP_SENS	0x119A	SYM0	0x56	0x0B	0
	0x119B	SYM1		0x11	1
		SYM2		0x1F	2
		SYM3		0x7C	3
		GLOBAL			
	11571	DTC			
MAP_DIP_SHIFT		SYM0	0x05	0x1E	0
		SYM1		0x1F	1
		SYM2		0x0B	2
	0x119C	SYM3		0x3C	3
		GLOBAL			
	11573	DTC			
MAP_PLAUS		SYM0	0x56	0x34	0
		SYM1		0x0B	1
		SYM2		0xDD	2
	0x129B	SYM3		0xDE	3
		GLOBAL			
	11563	DTC			
MAP_TPS_PLAUS	0x112F	SYM0	0x56	0x11	0
	0x112E	SYM1		0x26	1
		SYM2		0x0B	2
		SYM3		0x13	3
		GLOBAL			
	11566	DTC			
MEC_IVVT_EX		SYM0	0x58		
		SYM1			
		SYM2			
	0x0015	SYM3			
		GLOBAL			
	10887	DTC			
MEC_IVVT_IN		SYM0	0x58	0x11	0
		SYM1		0x1A	1
		SYM2		0x1B	2
	0x0012	SYM3		0x1F	3
		GLOBAL			
	10882	DTC			
MFF_MON_1		SYM0	0x88	0xC2	0
		SYM1		0x18	1
	0x3259	SYM2		0x0A	2
	0x323F	SYM3		0xA7	3

		GLOBAL			
	11616	DTC			
MIS_0	0x0301	SYM0	0x76	0x1F	0
	0x0301	SYM1		0xE5	1
	0x0301	SYM2		0x11	2
		SYM3		0x06	3
		GLOBAL			
	10701	DTC			
MIS_1	0x0305	SYM0	0x76	0x1F	0
	0x0305	SYM1		0xE5	1
	0x0305	SYM2		0x11	2
		SYM3		0x08	3
		GLOBAL			
	10705	DTC			
MIS_2	0x0303	SYM0	0x76	0x1F	0
	0x0303	SYM1		0xE5	1
	0x0303	SYM2		0x11	2
		SYM3		0x06	3
		GLOBAL			
	10703	DTC			
MIS_3	0x0306	SYM0	0x76	0x1F	0
	0x0306	SYM1		0xE5	1
	0x0306	SYM2		0x11	2
		SYM3		0x08	3
		GLOBAL			
	10706	DTC			
MIS_4	0x0302	SYM0	0x76	0x1F	0
	0x0302	SYM1		0xE5	1
	0x0302	SYM2		0x11	2
		SYM3		0x06	3
		GLOBAL			
	10702	DTC			
MIS_5	0x0304	SYM0	0x76	0x1F	0
	0x0304	SYM1		0xE5	1
	0x0304	SYM2		0x11	2
		SYM3		0x08	3
		GLOBAL			
	10704	DTC			
MIS_FTL_L		SYM0	0x21	0xF0	0
	0x0313	SYM1		0x6D	1
		SYM2		0x34	2
		SYM3		0x3B	3
		GLOBAL			
	10713	DTC			
MIS_MPL	0x0300	SYM0	0x56	0x24	0
	0x0300	SYM1		0xF0	1
	0x0300	SYM2		0x3C	2
	0x0300	SYM3		0x6D	3
		GLOBAL			

	10700	DTC			
MON_3	0x0605	SYM0	0x11	0xD9	0
	0x0604	SYM1		0xDA	1
	0x0606	SYM2		0xAA	2
	0x060C	SYM3		0xA9	3
		GLOBAL			
	11623	DTC			
MSW_2		SYM0	0x11	0x0D	0
		SYM1		0x7A	1
		SYM2		0x32	2
	0x1576	SYM3		0x7C	3
		GLOBAL			
	11701	DTC			
MSW_3		SYM0	0x11	0x0D	0
		SYM1		0x7A	1
		SYM2		0x32	2
	0x1563	SYM3		0x7C	3
		GLOBAL			
	11702	DTC			
MSW_TOG		SYM0	0x11	0x0D	0
		SYM1		0x7A	1
		SYM2		0x32	2
	0x155A	SYM3		0x7C	3
		GLOBAL			
	11703	DTC			
MTC_CTL_1	0x1638	SYM0	0x33	0x58	0
		SYM1		0x3F	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11500	DTC			
MTC_CTL_2	0x1639	SYM0	0x58	0x58	0
		SYM1		0x3F	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11501	DTC			
MTC_CTL_3	0x1637	SYM0	0x58	0x58	0
		SYM1		0x3F	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11502	DTC			
MTC_DR		SYM0	0x58	0x58	0
		SYM1		0x3F	1
	0x1636	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	11503	DTC			

N_32_MON_1		SYM0	0x88	0x11	0
		SYM1		0xB8	1
		SYM2		0xC0	2
	0x16A9	SYM3		0x32	3
		GLOBAL			
	11602	DTC			
N_MAX_DRIV_REQ		SYM0	0x22	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12177	DTC			
N_MAX_HOMS		SYM0	0x01	0x8B	0
		SYM1		0x0C	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11008	DTC			
NEUT_PSN_GB_LRN		SYM0	0x00	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12240	DTC			
NEUT_PSN_GB_PLAUS		SYM0	0x00	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12239	DTC			
NOX_SENS_1		SYM0	0x00	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11952	DTC			
NOX_SENS_2		SYM0	0x00	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11953	DTC			
NS_ACT_1	0x2201	SYM0	0x01	0xA2	0
		SYM1		0xA1	1
		SYM2		0xA3	2
		SYM3		0x10	3
		GLOBAL			
	12502	DTC			
NS_AFR_1	0x123C	SYM0	0x01	0xA3	0

	0x123D	SYM1		0xA1	1
	0x123E	SYM2		0x89	2
		SYM3		0xA2	3
		GLOBAL			
	12504	DTC			
NS_AVL_1	0x124C	SYM0	0x01	0xA4	0
	0x124D	SYM1		0x36	1
		SYM2		0xA2	2
		SYM3		0x10	3
		GLOBAL			
	12506	DTC			
NS_GAIN_1		SYM0	0x00	0xA4	0
		SYM1		0xA1	1
		SYM2		0xA6	2
		SYM3		0xA5	3
		GLOBAL			
	11006	DTC			
NS_HTP_1	0x125A	SYM0	0x01	0xA4	0
	0x125B	SYM1		0x8B	1
	0x125C	SYM2		0x0F	2
		SYM3		0x6A	3
		GLOBAL			
	12508	DTC			
NS_LSL_UP_DOWN_1	0x126A	SYM0	0x01	0xA2	0
		SYM1		0x89	1
		SYM2		0x8A	2
		SYM3		0xA3	3
		GLOBAL			
	12510	DTC			
NS_OBD_1_HTP_1		SYM0	0x01	0x10	0
		SYM1		0x32	1
	0x2205	SYM2		0x53	2
	0x121C	SYM3		0xA4	3
		GLOBAL			
	11013	DTC			
NS_OBD_1_LAMB_1		SYM0	0x01	0x10	0
		SYM1		0x32	1
	0x121E	SYM2		0x53	2
	0x121F	SYM3		0xA4	3
		GLOBAL			
	11014	DTC			
NS_OBD_1_NOX_1		SYM0	0x01	0x10	0
		SYM1		0x32	1
	0x2200	SYM2		0x53	2
	0x122C	SYM3		0xA4	3
		GLOBAL			
	11015	DTC			
NS_OBD_1_VLS_1		SYM0	0x01	0x10	0
		SYM1		0x32	1

	0x122E	SYM2		0x53	2
	0x122F	SYM3		0xA4	3
		GLOBAL			
	11003	DTC			
NS_OFS_1	0x126C	SYM0	0x01	0x10	0
		SYM1		0xA6	1
		SYM2		0x9E	2
		SYM3		0xA5	3
		GLOBAL			
	12512	DTC			
NS_PUC_1	0x126E	SYM0	0x01	0x49	0
	0x126F	SYM1		0xA1	1
	0x127B	SYM2		0xA6	2
	0x127A	SYM3		0xA3	3
		GLOBAL			
	12514	DTC			
NS_SHIFT_1		SYM0	0x00	0xE3	0
		SYM1		0xD1	1
		SYM2		0x49	2
		SYM3		0x4B	3
		GLOBAL			
	10988	DTC			
NS_STOP_1	0x128A	SYM0	0x01	0x49	0
	0x128B	SYM1		0xA3	1
		SYM2		0x4B	2
		SYM3		0x10	3
		GLOBAL			
	12516	DTC			
NS_VERS_1		SYM0	0x00	0x4A	0
		SYM1		0x32	1
		SYM2		0x1F	2
		SYM3		0x8B	3
		GLOBAL			
	10989	DTC			
NS_VLS_DYN_1	0x128E	SYM0	0x01	0xA2	0
		SYM1		0x4B	1
		SYM2		0xA1	2
		SYM3		0x10	3
		GLOBAL			
	12518	DTC			
NT_AGI	0x2000	SYM0	0x01	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12521	DTC			
NT_SO2P		SYM0	0x00	0xA5	0
		SYM1		0xA6	1
		SYM2		0x9E	2

		SYM3		0xA0	3
		GLOBAL			
	12522	DTC			
OBD_LSH_DOWN_1	0x0141	SYM0	0x56	0x96	0
		SYM1		0x5C	1
		SYM2		0x29	2
		SYM3		0x49	3
		GLOBAL			
	11432	DTC			
OBD_LSH_DOWN_2	0x0161	SYM0	0x56	0x97	0
		SYM1		0x5D	1
		SYM2		0x2A	2
		SYM3		0x4B	3
		GLOBAL			
	11433	DTC			
OBD_VLD_LSH_UP_1	0x0135	SYM0	0x56	0x94	0
	0x0135	SYM1		0x15	1
	0x165F	SYM2		0x27	2
		SYM3		0x8C	3
		GLOBAL			
	11430	DTC			
OBD_VLD_LSH_UP_2	0x0155	SYM0	0x56	0x95	0
	0x0155	SYM1		0x15	1
	0x166F	SYM2		0x28	2
		SYM3		0x8F	3
		GLOBAL			
	11431	DTC			
OC_LS_DOWN_1		SYM0	0x56	0x96	0
		SYM1		0x5C	1
	0x0140	SYM2		0x49	2
		SYM3		0x8B	3
		GLOBAL			
	11383	DTC			
OC_LS_DOWN_2		SYM0	0x56	0x97	0
		SYM1		0x5D	1
	0x0160	SYM2		0x4B	2
		SYM3		0x8B	3
		GLOBAL			
	11384	DTC			
OC_LSL_UP_1	0x2243	SYM0	0x56	0x71	0
	0x112C	SYM1		0x9B	1
	0x112C	SYM2		0x45	2
	0x2626	SYM3		0x8C	3
		GLOBAL			
	11325	DTC			
OC_LSL_UP_2	0x2247	SYM0	0x56	0x73	0
	0x112D	SYM1		0x9C	1
	0x112D	SYM2		0x48	2
	0x2629	SYM3		0x8F	3

		GLOBAL			
	11326	DTC			
OFS_LSL_UP_1		SYM0	0x00	0x9B	0
		SYM1		0x8B	1
		SYM2		0x21	2
		SYM3		0x15	3
		GLOBAL			
	11331	DTC			
OFS_LSL_UP_2		SYM0	0x00	0x9C	0
		SYM1		0x8B	1
		SYM2		0x21	2
		SYM3		0x15	3
		GLOBAL			
	11332	DTC			
PBK_IV_1	0x10A5	SYM0	0x58	0x11	0
	0x10A6	SYM1		0x1F	1
	0x10A7	SYM2		0x32	2
		SYM3		0x3C	3
		GLOBAL			
	12474	DTC			
PBK_IV_2	0x10A8	SYM0	0x58	0x11	0
	0x10A9	SYM1		0x1F	1
	0x10AA	SYM2		0x32	2
		SYM3		0x3C	3
		GLOBAL			
	12475	DTC			
PER_CAM_EX_1	0x130A	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10915	DTC			
PER_CAM_IN_1	0x1300	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10914	DTC			
PLAUS_CAM_EX_1	0x0365	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10913	DTC			
PLAUS_CAM_IN_1	0x0340	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			

	10912	DTC			
PLAUS_IGK_3		SYM0	0x00	0x00	0
		SYM1		0x00	1
		SYM2		0x00	2
		SYM3		0x00	3
		GLOBAL			
		DTC			
PLAUS_IGK_BN	0x15B0	SYM0	0x11	0x11	0
	0x15B1	SYM1		0x32	1
	0x15B2	SYM2		0x3C	2
	0x15B3	SYM3		0x7C	3
		GLOBAL			
	11715	DTC			
PLAUS_IV_CAL		SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12478	DTC			
POIL_CTL_DYN	0x159E	SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12480	DTC			
POIL_CTL_MEC	0x15A1	SYM0	0x11	0x11	0
	0x15A2	SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12484	DTC			
POIL_CTL_STAT	0x159F	SYM0	0x11	0x11	0
	0x15A0	SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12481	DTC			
POIL_DR	0x1582	SYM0	0x11	0x11	0
	0x1583	SYM1		0x22	1
	0x1584	SYM2		0x6F	2
		SYM3		0x3C	3
		GLOBAL			
	12482	DTC			
POIL_DR_SCG		SYM0	0x01	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12489	DTC			

POIL_PUMP	0x15A3	SYM0	0x11	0x11	0
	0x0524	SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12485	DTC			
POIL_SENS_PLAUS	0x15A6	SYM0	0x11	0x11	0
	0x15A7	SYM1		0x21	1
		SYM2		0x3C	2
	0x0521	SYM3		0x7C	3
		GLOBAL			
	12486	DTC			
POIL_SWI		SYM0	0x11	0x11	0
		SYM1		0x22	1
	0x0520	SYM2		0x1F	2
		SYM3		0x3C	3
		GLOBAL			
	12155	DTC			
POIL_SYS	0x15A5	SYM0	0x05	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12487	DTC			
PUC_LS_DOWN_1		SYM0	0x56	0x96	0
		SYM1		0x5C	1
		SYM2		0x49	2
	0x0139	SYM3		0x45	3
		GLOBAL			
	11387	DTC			
PUC_LS_DOWN_2		SYM0	0x56	0x97	0
		SYM1		0x5D	1
		SYM2		0x4B	2
	0x0159	SYM3		0x48	3
		GLOBAL			
	11388	DTC			
PUC_VLD_LS_UP_1	0x2297	SYM0	0x56	0x0B	0
		SYM1		0x45	1
		SYM2		0x7D	2
		SYM3		0x8C	3
		GLOBAL			
	11309	DTC			
PUC_VLD_LS_UP_2	0x2298	SYM0	0x56	0x0B	0
		SYM1		0x48	1
		SYM2		0x7E	2
		SYM3		0x8F	3
		GLOBAL			
	11310	DTC			
PUE_LS_DOWN_1		SYM0	0x01	0x96	0

		SYM1		0x5C	1
		SYM2		0x06	2
	0x1097	SYM3		0x49	3
		GLOBAL			
	11385	DTC			
PUE_LS_DOWN_2		SYM0	0x01	0x97	0
		SYM1		0x5D	1
		SYM2		0x08	2
	0x1098	SYM3		0x4B	3
		GLOBAL			
	11386	DTC			
PUT_EL	0x0238	SYM0	0x58	0x11	0
	0x0237	SYM1		0xDD	1
		SYM2		0x58	2
		SYM3		0x7C	3
		GLOBAL			
	10940	DTC			
PUT_PLAUS		SYM0	0x22	0x34	0
		SYM1		0x0B	1
		SYM2		0xDD	2
	0x0236	SYM3		0xDE	3
		GLOBAL			
	10941	DTC			
PVS_1	0x2123	SYM0	0x22	0x46	0
	0x2122	SYM1		0x47	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11547	DTC			
PVS_2	0x2128	SYM0	0x22	0x46	0
	0x2127	SYM1		0x47	1
		SYM2		0x54	2
		SYM3		0x3C	3
		GLOBAL			
	11548	DTC			
PVS_BLS_NOT_PLAUS		SYM0	0x11	0xB7	0
		SYM1		0x0D	1
		SYM2		0x14	2
	0x2299	SYM3		0xCE	3
		GLOBAL			
	12175	DTC			
PVS_DOUBLE		SYM0	0x88	0x43	0
		SYM1		0x54	1
		SYM2		0x46	2
	0x2120	SYM3		0x47	3
		GLOBAL			
	11551	DTC			
PVS_MON_1		SYM0	0x88	0xC4	0
		SYM1		0xB9	1

		SYM2		0xE7	2
	0x16B0	SYM3		0xE8	3
		GLOBAL			
	11605	DTC			
PVS_RATIO		SYM0	0x22	0x46	0
		SYM1		0x47	1
		SYM2		0x43	2
	0x2138	SYM3		0x14	3
		GLOBAL			
	11552	DTC			
QOIL_COM		SYM0	0x11	0x11	0
		SYM1		0x0D	1
	0x252A	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11937	DTC			
QOIL_SENS	0x1586	SYM0	0x11	0x11	0
	0x1587	SYM1		0x32	1
	0x1521	SYM2		0x7C	2
	0x1588	SYM3		0x3C	3
		GLOBAL			
	11935	DTC			
RAS		SYM0	0x11	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12126	DTC			
REF_CRK_CAM_EX_1	0x1553	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10905	DTC			
REF_CRK_CAM_IN_1	0x1554	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10904	DTC			
RFP_DR		SYM0	0x00	0x01	0
		SYM1		0x0B	1
		SYM2		0x03	2
		SYM3		0x07	3
		GLOBAL			
	12490	DTC			
RLY_ACCOUT	0x0647	SYM0	0x11	0x11	0
	0x0646	SYM1		0x0D	1
	0x0645	SYM2		0x1F	2

		SYM3		0x3C	3
		GLOBAL			
	12050	DTC			
RLY_CRCV_HEAT	0x053C	SYM0	0x11	0x24	0
	0x053B	SYM1		0x3A	1
	0x053A	SYM2		0x8B	2
		SYM3		0x7C	3
		GLOBAL			
	10980	DTC			
RLY_MAIN	0x0687	SYM0	0x21	0x8B	0
	0x0686	SYM1		0x4A	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10955	DTC			
RLY_MAIN_DLY	0x16C5	SYM0	0x11	0x43	0
		SYM1		0x4A	1
		SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10956	DTC			
RLY_MTC_HEAT	0x167B	SYM0	0x11	0x8B	0
	0x167C	SYM1		0x0C	1
	0x167D	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11531	DTC			
RLY_ST	0x0512	SYM0	0x11	0x8B	0
	0x0512	SYM1		0x4A	1
	0x0512	SYM2		0x53	2
		SYM3		0x3C	3
		GLOBAL			
	12120	DTC			
ROUGH_LEAK	0x0442	SYM0	0x56	0x3B	0
		SYM1		0x59	1
		SYM2		0x5A	2
		SYM3		0x8D	3
		GLOBAL			
	10773	DTC			
SCG_LS_DOWN_1		SYM0	0x56	0x96	0
	0x0137	SYM1		0x5C	1
		SYM2		0x49	2
		SYM3		0x8B	3
		GLOBAL			
	11381	DTC			
SCG_LS_DOWN_2		SYM0	0x56	0x97	0
	0x0157	SYM1		0x5D	1
		SYM2		0x4B	2
		SYM3		0x8B	3

		GLOBAL			
	11382	DTC			
SCP_LS_DOWN_1	0x0138	SYM0	0x56	0x96	0
		SYM1		0x5C	1
		SYM2		0x49	2
		SYM3		0x8B	3
		GLOBAL			
	11379	DTC			
SCP_LS_DOWN_2	0x0158	SYM0	0x56	0x97	0
		SYM1		0x5D	1
		SYM2		0x4B	2
		SYM3		0x8B	3
		GLOBAL			
	11380	DTC			
SEG_AD_ER		SYM0	0x56	0x11	0
	0x1396	SYM1		0x3C	1
		SYM2		0xF8	2
		SYM3		0xF9	3
		GLOBAL			
	10714	DTC			
SEG_EOC_INJ		SYM0	0x11	0x89	0
		SYM1		0x8A	1
		SYM2		0x11	2
		SYM3		0x14	3
		GLOBAL			
	11873	DTC			
SENS_ACR	0x040A	SYM0	0x01	0xFC	0
	0x040C	SYM1		0xF7	1
		SYM2		0xE4	2
		SYM3		0x7C	3
		GLOBAL			
	10768	DTC			
SENS_BAT_SMT_COM		SYM0	0x11	0x7C	0
		SYM1		0x24	1
	0x150B	SYM2		0x6A	2
		SYM3		0x3C	3
		GLOBAL			
	11918	DTC			
SENS_POIL	0x0523	SYM0	0x11	0x0D	0
	0x0522	SYM1		0x11	1
		SYM2		0x22	2
		SYM3		0x6F	3
		GLOBAL			
	12483	DTC			
SHIFT_AFL_LSL_UP_1	0x2195	SYM0	0x56	0x8C	0
		SYM1		0x49	1
		SYM2		0x71	2
		SYM3		0x45	3
		GLOBAL			

	11303	DTC			
SHIFT_AFL_LSL_UP_2	0x2197	SYM0	0x56	0x8F	0
		SYM1		0x4B	1
		SYM2		0x73	2
		SYM3		0x48	3
		GLOBAL			
	11304	DTC			
SHIFT_AFR_LSL_UP_1	0x2196	SYM0	0x56	0x8C	0
		SYM1		0x49	1
		SYM2		0x71	2
		SYM3		0x45	3
		GLOBAL			
	11307	DTC			
SHIFT_AFR_LSL_UP_2	0x2198	SYM0	0x56	0x8F	0
		SYM1		0x4B	1
		SYM2		0x73	2
		SYM3		0x48	3
		GLOBAL			
	11308	DTC			
SLV_IVVT_EX	0x2091	SYM0	0x58	0x11	0
	0x2090	SYM1		0x32	1
	0x0013	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10885	DTC			
SLV_IVVT_IN	0x2089	SYM0	0x58	0x11	0
	0x2088	SYM1		0x32	1
	0x0010	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10880	DTC			
SMALL_LEAK		SYM0	0x56	0x3B	0
	0x0456	SYM1		0x59	1
		SYM2		0x5B	2
		SYM3		0x8D	3
		GLOBAL			
	10774	DTC			
SOF	0x15AC	SYM0	0x00	0x11	0
	0x15AD	SYM1		0x0D	1
	0x15AE	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10945	DTC			
SOF_REQ	0x1540	SYM0	0x11	0x11	0
	0x1541	SYM1		0x0D	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10950	DTC			

SPI_KNK		SYM0	0x58	0x11	0
		SYM1		0x21	1
	0x16A4	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10933	DTC			
SPI_MPS		SYM0	0x56	0x11	0
		SYM1		0x21	1
	0x16A5	SYM2		0x7C	2
		SYM3		0x3C	3
		GLOBAL			
	10934	DTC			
SWI_AFS_MON		SYM0	0x11	0xFE	0
		SYM1		0x2E	1
		SYM2		0x2F	2
		SYM3		0x9D	3
		GLOBAL			
	11620	DTC			
SWT_LS_DOWN_1		SYM0	0x56	0x96	0
		SYM1		0x5C	1
		SYM2		0x10	2
	0x1130	SYM3		0x49	3
		GLOBAL			
	11373	DTC			
SWT_LS_DOWN_2		SYM0	0x56	0x97	0
		SYM1		0x5D	1
		SYM2		0x10	2
	0x1131	SYM3		0x4B	3
		GLOBAL			
	11374	DTC			
SYN_CAM_EX_1	0x0369	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10911	DTC			
SYN_CAM_IN_1	0x0344	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10910	DTC			
SYN_CRK_CAM_EX_1	0x0366	SYM0	0x22	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10907	DTC			
SYN_CRK_CAM_IN_1	0x0341	SYM0	0x58	0x11	0

		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10906	DTC			
T_ES		SYM0	0x56	0xA8	0
		SYM1		0x1F	1
	0x1551	SYM2		0x7C	2
	0x1515	SYM3		0x3C	3
		GLOBAL			
	12160	DTC			
T_SEG_ER		SYM0	0x11	0x11	0
	0x0370	SYM1		0x1F	1
		SYM2		0x18	2
		SYM3		0x3C	3
		GLOBAL			
	10715	DTC			
TAM_CAN	0x0073	SYM0	0x56	0x24	0
	0x0072	SYM1		0x33	1
	0x0070	SYM2		0x1E	2
		SYM3		0x7C	3
		GLOBAL			
	12186	DTC			
TAM_PLAUS		SYM0	0x56	0x24	0
		SYM1		0x33	1
		SYM2		0x82	2
	0x0071	SYM3		0x20	3
		GLOBAL			
	12185	DTC			
TCHA_LEAK		SYM0	0x22	0x34	0
		SYM1		0x24	1
		SYM2		0xDD	2
	0x15AA	SYM3		0x0C	3
		GLOBAL			
	12540	DTC			
TCHA_PRS_CTL		SYM0	0x22	0x34	0
		SYM1		0x24	1
		SYM2		0xDD	2
		SYM3		0x0C	3
		GLOBAL			
	12544	DTC			
TCHA_PRS_DIF		SYM0	0x00	0x34	0
		SYM1		0x24	1
		SYM2		0xDD	2
		SYM3		0x0C	3
		GLOBAL			
	12541	DTC			
TCHA_PRS_HIGH		SYM0	0x22	0x34	0
		SYM1		0x24	1

		SYM2		0xDD	2
	0x0234	SYM3		0x0C	3
		GLOBAL			
	12542	DTC			
TCHA_PRS_LOW		SYM0	0x22	0x34	0
		SYM1		0x24	1
		SYM2		0xDD	2
	0x0299	SYM3		0x0C	3
		GLOBAL			
	12543	DTC			
TCHA_SYS_1		SYM0	0x00	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12545	DTC			
TCO_2_EL	0x00B4	SYM0	0x00	0x52	0
	0x00B3	SYM1		0x20	1
		SYM2		0x24	2
		SYM3		0x1E	3
		GLOBAL			
	12010	DTC			
TCO_2_GRD		SYM0	0x00	0x20	0
		SYM1		0x1F	1
		SYM2		0x24	2
	0x3197	SYM3		0xEA	3
		GLOBAL			
	12011	DTC			
TCO_2_PLAUS	0x3196	SYM0	0x00	0x20	0
		SYM1		0x82	1
		SYM2		0x1F	2
	0x00B2	SYM3		0x32	3
		GLOBAL			
	12012	DTC			
TCO_EL	0x0118	SYM0	0x56	0x50	0
	0x0117	SYM1		0x1F	1
		SYM2		0x24	2
		SYM3		0x1E	3
		GLOBAL			
	12000	DTC			
TCO_GRD		SYM0	0x56	0x1F	0
		SYM1		0x20	1
		SYM2		0x24	2
	0x3198	SYM3		0x7F	3
		GLOBAL			
	12003	DTC			
TCO_PLAUS		SYM0	0x01	0x1F	0
		SYM1		0x20	1
		SYM2		0x3C	2

	0x0125	SYM3		0xEC	3
		GLOBAL			
	12001	DTC			
TCO_STUCK		SYM0	0x56	0x1F	0
		SYM1		0x20	1
		SYM2		0x24	2
	0x3199	SYM3		0x82	3
		GLOBAL			
	12002	DTC			
TCO_STUCK_RNG		SYM0	0x56	0x23	0
		SYM1		0x82	1
		SYM2		0x3A	2
	0x316A	SYM3		0x33	3
		GLOBAL			
	12006	DTC			
TECU	0x0669	SYM0	0x11	0x41	0
	0x0668	SYM1		0x21	1
		SYM2		0x24	2
		SYM3		0x3C	3
		GLOBAL			
	12165	DTC			
TEG_PCAT_DOWN	0x0546	SYM0	0x01	0x11	0
	0x0545	SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11399	DTC			
TH		SYM0	0x56	0x24	0
		SYM1		0x82	1
		SYM2		0x20	2
	0x0128	SYM3		0x11	3
		GLOBAL			
	12020	DTC			
TIA_EL	0x0113	SYM0	0x58	0x51	0
	0x0112	SYM1		0x1E	1
		SYM2		0x24	2
		SYM3		0x3C	3
		GLOBAL			
	12040	DTC			
TIA_GRD		SYM0	0x58	0x11	0
		SYM1		0x21	1
	0x11BB	SYM2		0x3C	2
	0x115E	SYM3		0x7C	3
		GLOBAL			
	12044	DTC			
TIA_PLAUS	0x111E	SYM0	0x58	0x1E	0
	0x111F	SYM1		0x3A	1
		SYM2		0x24	2
	0x0111	SYM3		0x1F	3

		GLOBAL			
	12041	DTC			
TIA_TCHA_UP	0x104E	SYM0	0x00	0x51	0
	0x104F	SYM1		0x1E	1
		SYM2		0x24	2
		SYM3		0xD5	3
		GLOBAL			
	12042	DTC			
TOIL_LEVEL		SYM0	0x11	0x11	0
	0x250F	SYM1		0x21	1
	0x250A	SYM2		0x3C	2
	0x250B	SYM3		0x7C	3
		GLOBAL			
	12190	DTC			
TOOTH_OFF_EX_1	0x0017	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10917	DTC			
TOOTH_OFF_IN_1	0x0016	SYM0	0x58	0x11	0
		SYM1		0x32	1
		SYM2		0x22	2
		SYM3		0x3C	3
		GLOBAL			
	10916	DTC			
TOUT_NOX_SENS_1		SYM0	0x00	0x11	0
		SYM1		0x21	1
	0xD144	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11950	DTC			
TOUT_NOX_SENS_2		SYM0	0x00	0x11	0
		SYM1		0x21	1
	0xD145	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11951	DTC			
TPS		SYM0	0x88	0x11	0
		SYM1		0x1E	1
	0x1417	SYM2		0x1F	2
		SYM3		0x7C	3
		GLOBAL			
	11529	DTC			
TPS_1	0x0123	SYM0	0x58	0x4E	0
	0x0122	SYM1		0x4C	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			

	11513	DTC			
TPS_2	0x0223	SYM0	0x58	0x4E	0
	0x0222	SYM1		0x4C	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11514	DTC			
TPS_AD	0x1632	SYM0	0x58	0x4E	0
	0x1633	SYM1		0x4C	1
	0x1694	SYM2		0xB0	2
	0x1644	SYM3		0x3C	3
		GLOBAL			
	11515	DTC			
TPS_AD_BOL	0x1635	SYM0	0x58	0x4E	0
		SYM1		0x4C	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11518	DTC			
TPS_JAM_DET	0x164E	SYM0	0x11	0x8B	0
	0x164F	SYM1		0x0C	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	11532	DTC			
TPS_MAF_1		SYM0	0x58	0xAB	0
		SYM1		0xE4	1
		SYM2		0x4C	2
	0x0121	SYM3		0x4E	3
		GLOBAL			
	11510	DTC			
TPS_MAF_2		SYM0	0x58	0xAC	0
		SYM1		0xE4	1
		SYM2		0x4C	2
	0x0221	SYM3		0x4E	3
		GLOBAL			
	11511	DTC			
TPS_MON_1		SYM0	0x88	0x58	0
		SYM1		0xB8	1
		SYM2		0x4E	2
	0x110D	SYM3		0x4C	3
		GLOBAL			
	11617	DTC			
TPS_RATIO		SYM0	0x01	0x4E	0
		SYM1		0x4C	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11527	DTC			

TPS_ST_CHK_1	0x1675	SYM0	0x11	0x4E	0
		SYM1		0x4C	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11517	DTC			
TPS_ST_CHK_2	0x1634	SYM0	0x58	0x4E	0
	0x169A	SYM1		0x4C	1
		SYM2		0x43	2
		SYM3		0x3C	3
		GLOBAL			
	11516	DTC			
TQ_CST	0x1559	SYM0	0x00	0x11	0
		SYM1		0x21	1
		SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10977	DTC			
TQ_DIF_ISC_MON_1		SYM0	0x88	0xC3	0
		SYM1		0xC7	1
	0x16B1	SYM2		0xC8	2
	0x16B2	SYM3		0xCA	3
		GLOBAL			
	11606	DTC			
TQ_EXT_MON_1	0x16B3	SYM0	0x11	0x81	0
		SYM1		0xBF	1
	0x16B5	SYM2		0x93	2
	0x16B6	SYM3		0x3C	3
		GLOBAL			
	11607	DTC			
TQ_REQ_CAN		SYM0	0x01	0x11	0
		SYM1		0x2B	1
		SYM2		0x3C	2
	0x1771	SYM3		0x7C	3
		GLOBAL			
	11717	DTC			
TQ_REQ_MON_1	0x16B7	SYM0	0x88	0xD4	0
	0x16B8	SYM1		0xD6	1
	0x16B9	SYM2		0xCD	2
		SYM3		0x32	3
		GLOBAL			
	11608	DTC			
TQI_AV_MON_1		SYM0	0x88	0xB8	0
		SYM1		0xCF	1
		SYM2		0xD0	2
	0x16C1	SYM3		0x75	3
		GLOBAL			
	11609	DTC			
TQI_LIM		SYM0	0x11	0x11	0

		SYM1		0x32	1
		SYM2		0xCF	2
		SYM3		0xD1	3
		GLOBAL			
	11610	DTC			
TQI_N_MAX_MON_1		SYM0	0x88	0xC5	0
		SYM1		0xC6	1
		SYM2		0xDB	2
	0x16C2	SYM3		0xDC	3
		GLOBAL			
	11603	DTC			
TTIP_MES_LSH_UP_1	0x3026	SYM0	0x56	0x94	0
	0x3026	SYM1		0x15	1
	0x165F	SYM2		0x27	2
		SYM3		0x45	3
		GLOBAL			
	11434	DTC			
TTIP_MES_LSH_UP_2	0x3027	SYM0	0x56	0x95	0
	0x3027	SYM1		0x15	1
	0x166F	SYM2		0x28	2
		SYM3		0x48	3
		GLOBAL			
	11435	DTC			
V_REF_1		SYM0	0x44	0x43	0
		SYM1		0x54	1
		SYM2		0x46	2
	0x164C	SYM3		0x3C	3
		GLOBAL			
	11549	DTC			
V_REF_2		SYM0	0x44	0x43	0
		SYM1		0x54	1
		SYM2		0x47	2
	0x1625	SYM3		0x3C	3
		GLOBAL			
	11550	DTC			
VAR_COD		SYM0	0x11	0x11	0
		SYM1		0x3C	1
	0x062F	SYM2		0x7C	2
		SYM3		0x8B	3
		GLOBAL			
	12195	DTC			
VCV	0x0092	SYM0	0x58	0xF2	0
	0x0091	SYM1		0xF0	1
	0x0090	SYM2		0xE4	2
		SYM3		0x7C	3
		GLOBAL			
	12220	DTC			
VCV_PLAUS		SYM0	0x11	0xF0	0
		SYM1		0x11	1

		SYM2		0xF2	2
		SYM3		0xBA	3
		GLOBAL			
	12227	DTC			
VEH_POW_VAR		SYM0	0x11	0x11	0
		SYM1		0x3C	1
		SYM2		0x7C	2
	0x3235	SYM3		0x8B	3
		GLOBAL			
	12196	DTC			
VIM_1_EL	0x1513	SYM0	0x11	0x11	0
	0x1512	SYM1		0x0D	1
	0x1511	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10920	DTC			
VIM_2_EL	0x1513	SYM0	0x11	0x11	0
	0x1512	SYM1		0x0D	1
	0x1511	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	10921	DTC			
VIM_PLAUS		SYM0	0x11	0x11	0
		SYM1		0x1F	1
		SYM2		0x32	2
		SYM3		0x7C	3
		GLOBAL			
	10922	DTC			
VIMPWM_1_FB		SYM0	0x11	0x3C	0
		SYM1		0x0C	1
		SYM2		0x18	2
	0x14C2	SYM3		0x24	3
		GLOBAL			
	10923	DTC			
VIMPWM_2_FB		SYM0	0x11	0x3C	0
		SYM1		0x0C	1
		SYM2		0x18	2
	0x14C3	SYM3		0x24	3
		GLOBAL			
	10924	DTC			
VLS_DOWN_DIF_1	0x114A	SYM0	0x56	0x49	0
	0x114B	SYM1		0x45	1
		SYM2		0x78	2
		SYM3		0xF5	3
		GLOBAL			
	11390	DTC			
VLS_DOWN_DIF_2	0x114C	SYM0	0x56	0x4B	0
	0x114D	SYM1		0x48	1
		SYM2		0x79	2

		SYM3		0xF6	3
		GLOBAL			
	11391	DTC			
VS		SYM0	0x56	0x11	0
		SYM1		0x32	1
	0x0500	SYM2		0x3C	2
		SYM3		0x81	3
		GLOBAL			
	12110	DTC			
VS_PLAUS		SYM0	0x56	0x11	0
		SYM1		0x32	1
		SYM2		0x7C	2
	0x0503	SYM3		0x3C	3
		GLOBAL			
	12111	DTC			
WARM_RST	0x0602	SYM0	0x05	0x67	0
		SYM1		0x3D	1
	0x1640	SYM2		0x3E	2
	0x1615	SYM3		0x40	3
		GLOBAL			
	11615	DTC			
WG_1_DR	0x0246	SYM0	0x22	0x11	0
	0x0245	SYM1		0x1F	1
	0x0243	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12495	DTC			
WG_2_DR	0x0250	SYM0	0x22	0x11	0
	0x0249	SYM1		0x1F	1
	0x0247	SYM2		0x3C	2
		SYM3		0x7C	3
		GLOBAL			
	12496	DTC			

27.6 Anti-bounce algorithms

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_PREV_XX	-	0...FH	0...15	1	[-]
Previous value of ERR_SYM_XX when the diagnosis conditions are fulfilled					
CTR_ABC_XX	V/O/S	0...FFH	0...255	1	[-]
anti bounce counter of diagnosis XX					
CTR_ABC_END_DIAG_XX	V	0...FFH	0...255	1	[-]
Counter for end of diagnosis XX generation					
LV_END_DIAG_XX	V	0...1H	0...1	1	[-]
Diagnostic done completely at least one time					
LV_CDN_DIAG_XX	V	0...1H	0...1	1	[-]
Diagnostic condition to start symptom detection (set to 1 when condition is fulfilled)					
ERR_SYM_XX	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
For each symptom : status of failure (set to 1 when failure symptom detected)					

Remark : The present failure and the anti-bounce counter may be saved or not (see 'Calculation of the Anti-bounce')

Input data:

LV_IGK	LV_DC	LV_ABC_INH	

Import actions:

#IF NLC_ERR_DET_UPD = 1
ACTION_ERRM_UpdErrDet (IN<XX>)
This action indicates each occurrence of the detected error
#ENDIF

Export actions:

Name
ACTION_ERRM_FilterSymptom(IN< XX >, IN< lv_cdn_diag_XX >, IN< err_sym_XX >, IN< C_ABC_INC_XX >, IN< C_ABC_DEC_XX >, IN< C_ABC_MAX_XX >, OUT< LV_ERR_XX >)
This action computes the elementary anti-bounce filter for one failure treatment and returns filter result
ACTION_ERRM_NoFilterSymptom(IN< XX >, IN< lv_cdn_diag_XX >, IN< err_sym_XX >, IN< lv_err_set_XX >, IN< lv_err_reset_XX >, IN< lv_end_diag_XX >, OUT< LV_ERR_XX >)
This action computes the elementary treatment case no filtering is used
ACTION_ERRM_AbcFilterSymptomEnd(IN< XX >, IN< C_ABC_MAX_XX>, OUT< LV_ERR_XX >)
This action erases diagnostic failure and sets the end of diagnostic in case of antibounce filter type
ACTION_ERRM_NoFilterSymptomEnd(IN< XX >, OUT< LV_ERR_XX >)
This action erases diagnostic failure and sets the end of diagnostic in case of no filter
ACTION_ERRM_AbcFilterReset(IN< XX >, OUT< LV_ERR_XX >)
This action resets data filter in case of antibounce filter type
ACTION_ERRM_NoFilterReset(IN< XX >, OUT< LV_ERR_XX >)
This action resets data filter in case of no filter
ACTION_ERRM_GetLvErr(IN< XX >, OUT< LV_ERR_XX >)
This action is used to get LV_ERR_XX value
ACTION_ERRM_GetLvEndDiag(IN< XX >, OUT< LV_END_DIAG_XX >)
This action is used to get LV_END_DIAG_XX value
ACTION_ERRM_GetLvCdnDiag(IN< XX >, OUT< LV_CDN_DIAG_XX >)
This action is used to get LV_CDN_DIAG_XX value
ACTION_ERRM_GetErrSym(IN< XX >, OUT< ERR_SYM_XX >)
This action is used to get ERR_SYM_XX value

ERR_SYM_XX is defined like output data for each diagnosis as following :

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
ERR_SYM_XX	O/V	0..0FH	All combination	1	-
		0H	NO_SYM		
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
		FH	BENCH_MODE		
Symptom : failure without filtering of diagnosis XX					

It is possible to combine several symptoms.

FUNCTION DESCRIPTION:

General information:

The filtering algorithm (anti-bounce or statistic) is used usually for the simple diagnosis (e.g. : usually OBD1 diagnosis). If there is another need, diagnosis manages itself the filtering and the end of diagnosis (e.g. : usually OBD2 diagnosis).

The filtering is used to filter the detected failure. The failure becomes present after filtering.

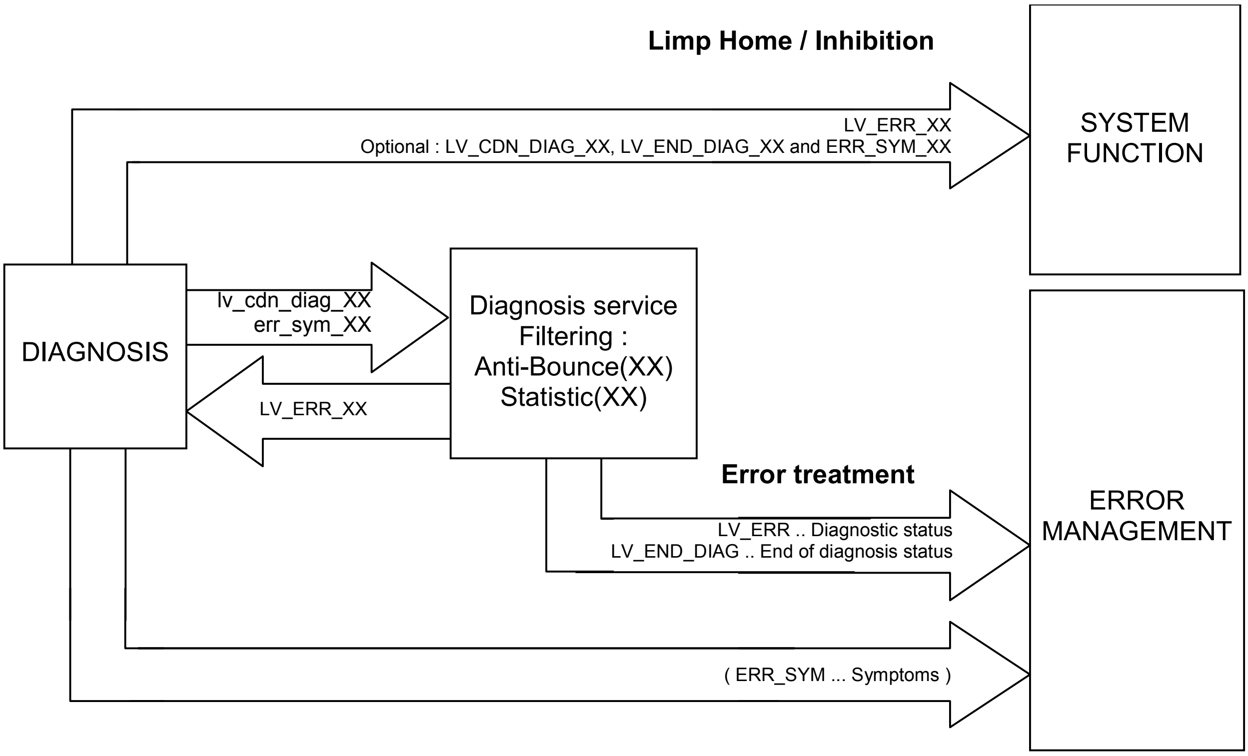
The end of diagnosis definition is realised after a time defined as the minimum time to detect a present failure.

The filtering algorithm (anti-bounce or statistic) uses one counter to check failure detection threshold and one counter to manage the end of diagnosis (Diagnostic result available).

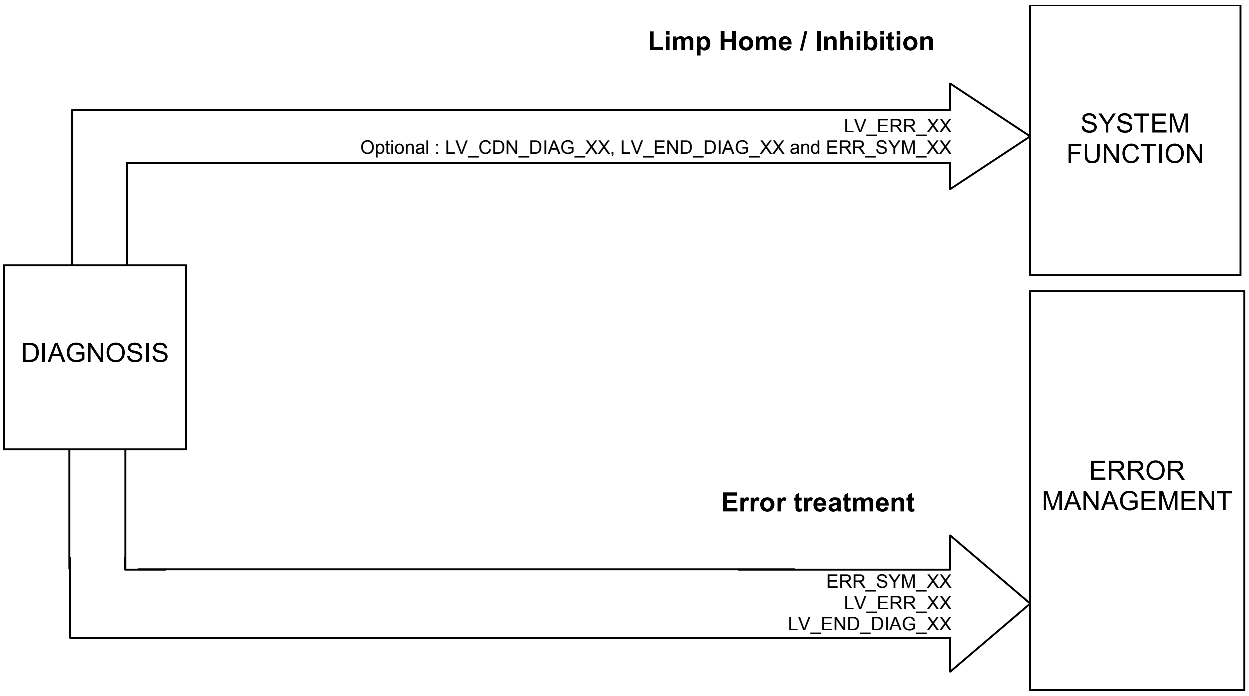
Filter choice selection : according to the diagnostic filtering configuration the filtering algorithm used is the anti-bounce filtering (NLC_ABC_STC_XX = 0) or the standard statistical filtering (NLC_ABC_STC_XX=1)

Signal flow diagram :

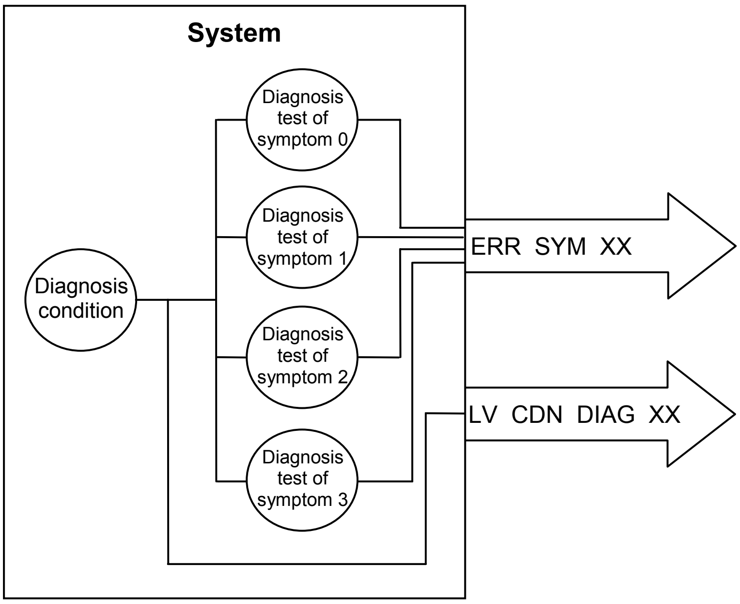
Diagnosis with an anti-bounce algorithm :



Diagnosis without anti-bounce algorithm :



The symptoms are evaluated when diagnosis conditions are fulfilled :



27.6.1 Initialisation of anti-bounce, statistical and multi-conditions filters

This paragraph is dedicated to the initialisation of main data, at the following events:

- ECU reset
- LV_IGK 0 → 1 transition before the end of the power-latch phase
- LV_DC 1 → 0 transition

27.6.1.1 At ECU reset

Management of NVMY

```
If          NVMY not corrupted and not first ECU power-up
Then
            CTR_ABC_ XX is restored from NVMY
            LV_ERR_XX is restored from NVMY
            ERR_SYM_XX is restored from NVMY
Else
            CTR_ABC_ XX = 0
            LV_ERR_XX = 0
            ERR_SYM_XX = 0
Endif

LV_CDN_DIAG_XX = 0
CTR_ABC_END_DIAG_XX = 0
LV_END_DIAG_XX = 0

If          NLC_ABC_NOT_SAVE_XX = 1
Then
            { case of STD_INI, MEM and STC configurations }
            CTR_ABC_XX = 0
            LV_ERR_XX = 0
            ERR_SYM_XX = 0
Endif
```

Configuration evaluation

```
If          NLC_ABC_INI_VALUE_XX = 1
Then
            { case of MEM_INI configuration }

            If      LV_ERR_XX = 1
            Then
                    CTR_ABC_XX = C_ABC_MAX_XX - C_ABC_INC_XX
            Else
                    CTR_ABC_XX = 0
            Endif
            LV_ERR_XX = 0
            ERR_SYM_XX = 0
Endif
```

27.6.1.2 At LV_IGK 0 → 1 transition, without ECU reset

```
If      NLC_ABC_NOT_SAVE_XX = 1 and NLC_ABC_INI_IGK_XX = 1
Then
        CTR_ABC_ XX = 0
        LV_ERR_XX = 0
        ERR_SYM_XX = 0
Endif

LV_CDN_DIAG_XX = 0

If      NLC_ABC_NOT_SAVE_XX = 0
        and NLC_ABC_INI_VALUE_XX = 1
        and NLC_ABC_INI_IGK_XX = 1
Then
        { case of MEM_INI configuration }
        If    LV_ERR_XX = 1
        Then
              CTR_ABC_XX = C_ABC_MAX_XX - C_ABC_INC_XX
        Else
              CTR_ABC_XX = 0
        Endif
        LV_ERR_XX = 0
        ERR_SYM_XX = 0
Endif

If      NLC_ABC_INI_DC_END_DIAG = 0
Then
        CTR_ABC_END_DIAG_XX = 0
        LV_END_DIAG_XX = 0
Endif
```

27.6.1.3 At LV_DC 1 → 0 transition

```
If      NLC_ABC_INI_DC_END_DIAG = 1
Then
        CTR_ABC_END_DIAG_XX = 0
        LV_END_DIAG_XX = 0
Endif

If      LV_ERR_XX = 0 and NLC_ABC_INI_VALUE_XX = 0
Then
        { all cases excepted MEM_INI configuration }
        CTR_ABC_XX = 0
        ERR_SYM_XX = 0
Endif
```

27.6.2 Recurrence

The recurrence is always managed by the diagnosis recurrence.

27.6.3 Filtering algorithm description

27.6.3.1 Description:

Syntax :	ACTION_ERRM_FilterSymptom(IN<XX>, IN<lv_cdn_diag_XX>, IN< err_sym_xx>, IN<C_ABC_INC_XX>, IN<C_ABC_DEC_XX>,IN<C_ABC_MAX_XX>, OUT<LV_ERR_XX>)	
Parameter(in) :	XX	Type: index to identify the Diagnostic
	lv_cdn_diag_XX	Type: Diagnostic condition
	err_sym_XX	Status of failure
	C_ABC_INC_XX	Anti-bounce counter increment
	C_ABC_DEC_XX	Anti-bounce counter decrement
	C_ABC_MAX_XX	Maximum value of anti-bounce counter
Parameter(out) :	LV_ERR_XX	Type: Boolean (=1 when failure)
Short Description :	This action returns the filter result on symptoms detected at each diagnostic recurrence	

27.6.3.2 Antibounce algorithm filtering type

27.6.3.2.1 Calculation of the anti-bounce counter

Description:

Anti-bounce operation:

If the system failure is detected and the diagnosis conditions are fulfilled (LV_CDN_DIAG_XX = 1) then the failure is detected (ERR_SYM_XX ≠ 0) and the anti-bounce counter is incremented. In this case an action is called to manage some functionality (eg : to visualise the sporadic error of all diagnosis using this anti-bounce algorithm).

If the system failure isn't detected and the diagnosis conditions are fulfilled (LV_CDN_DIAG_XX = 1) then the failure isn't detected (ERR_SYM_XX = 0) and the anti-bounce counter is decremented.

If the diagnosis conditions aren't fulfilled (LV_CDN_DIAG_XX = 0) then the detected failure is unchanged ($ERR\_SYM\_XX_{(N)} = ERR\_SYM\_XX_{(N-1)}$) and the anti-bounce counter is unchanged.

The present failure is set when the anti-bounce counter has reached its MAX (unequal to zero).

The configuration allows:

- to save or not the present failure and the anti-bounce counter
- to initialise or not the present failure and the anti-bounce counter at transition LV_IGK 0 → 1
- to manage the decrement of the anti-bounce counter or not (the anti-bounce counter remains at the maximum value)
- to calibrate or not the decrement of the anti-bounce counter.

Anti-bounce bench mode:

To make easier the development phase, it may be interesting to force the LV_ERR_XX flag (present failure) and the associated symptoms in order to free the diagnosis and to test only the error management and/or the application incidence (limp home or inhibition) associated to the present failure. This mode is active only if the LC_ABC_BENCH and NLC_BENCH_MODE flags are set.

To set LV_ERR_XX to one and ERR_SYM_XX = FH (all detected error symptoms) :

Set the C_ABC_INC_XX calibration to the maximum value (FFH).

To set LV_ERR_XX to zero and ERR_SYM_XX = 0 (no detected error symptom) :

Set the C_ABC_INC_XX calibration to zero.

Note:

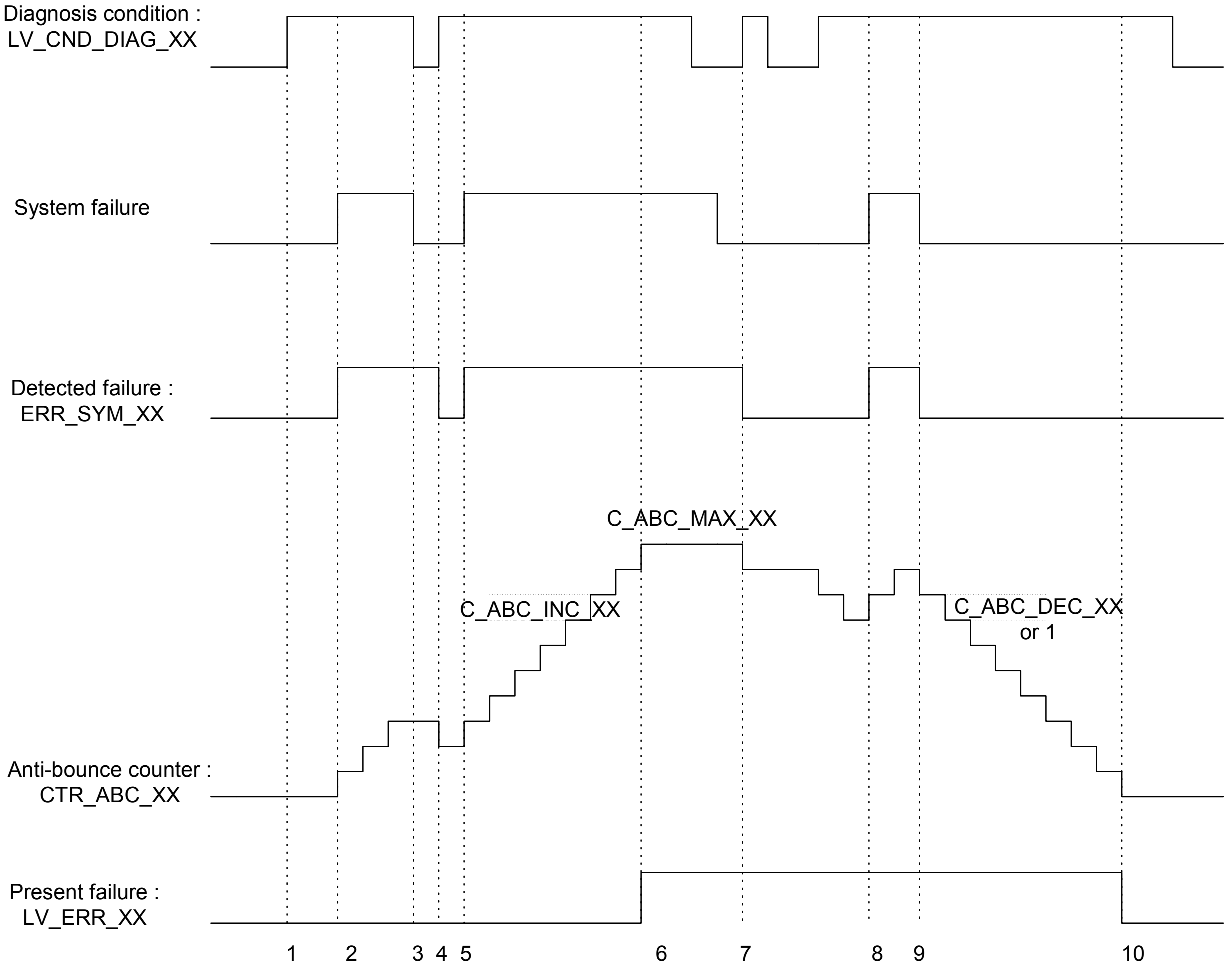
In this mode, the diagnosis condition (LV_CDN_DIAG_XX) and the end of diagnosis (LV_END_DIAG_XX) are set to 1 (LV_CDN_DIAG_XX = 1 means the ERR_SYM_XX is valid and LV_END_DIAG_XX = 1 means the LV_ERR_XX flag is valid).

Anti-bounce inhibition:

For specific needs, the anti-bounce filtering calculation (anti-bounce operation and anti-bounce bench mode) may be inhibited, with LV_ABC_INH inhibition flag.

Signal flow diagram:

1 -The anti-bounce counter may be decremented (NLC_ABC_NOT_DEC_XX = 0):



1: The diagnosis conditions are fulfilled
but the failure isn't detected ➔ CTR_ABC_XX remains at the value 0.
LV_CDN_DIAG_XX = 1, LV_ERR_XX = 0.
ERR_SYM_XX = 0.

2: The diagnosis conditions are fulfilled
and the failure is detected. ➔ CTR_ABC_XX = 0 + C_ABC_INC_XX
LV_CDN_DIAG_XX = 1, LV_ERR_XX = 0.
ERR_SYM_XX <> 0.

3: The diagnosis conditions aren’t fulfilled
and the failure remains detected. ➔ CTR_ABC_XX is frozen
LV_CDN_DIAG_XX = 0, LV_ERR_XX = 0.
ERR_SYM_XX <> 0.

.

4: The diagnosis conditions are fulfilled again
and the failure is no more detected. ➔ CTR_ABC_XX =
LV_CDN_DIAG_XX = 1, CTR_ABC_XX – decrement
ERR_SYM_XX = 0. LV_ERR_XX = 0.

5: The diagnosis conditions are still fulfilled
and the failure is detected again ➔ CTR_ABC_XX =
LV_CDN_DIAG_XX = 1, CTR_ABC_XX + C_ABC_INC_XX
ERR_SYM_XX <> 0. LV_ERR_XX = 0.

6: The diagnosis conditions are still fulfilled,
the failure is still detected
and the failure becomes present. ➔ CTR_ABC_XX = C_ABC_MAX_XX
LV_CDN_DIAG_XX = 1, LV_ERR_XX = 1.
ERR_SYM_XX <> 0.

7: The diagnosis conditions are fulfilled,
the failure isn’t detected
and the failure remains present. ➔ CTR_ABC_XX =
LV_CDN_DIAG_XX = 1, CTR_ABC_XX – decrement
ERR_SYM_XX = 0. LV_ERR_XX = 1.

8: The diagnosis conditions are still fulfilled,
the failure is detected again
and the failure remains present. ➔ CTR_ABC_XX =
LV_CDN_DIAG_XX = 1, CTR_ABC_XX + C_ABC_INC_XX
ERR_SYM_XX <> 0. LV_ERR_XX = 1.

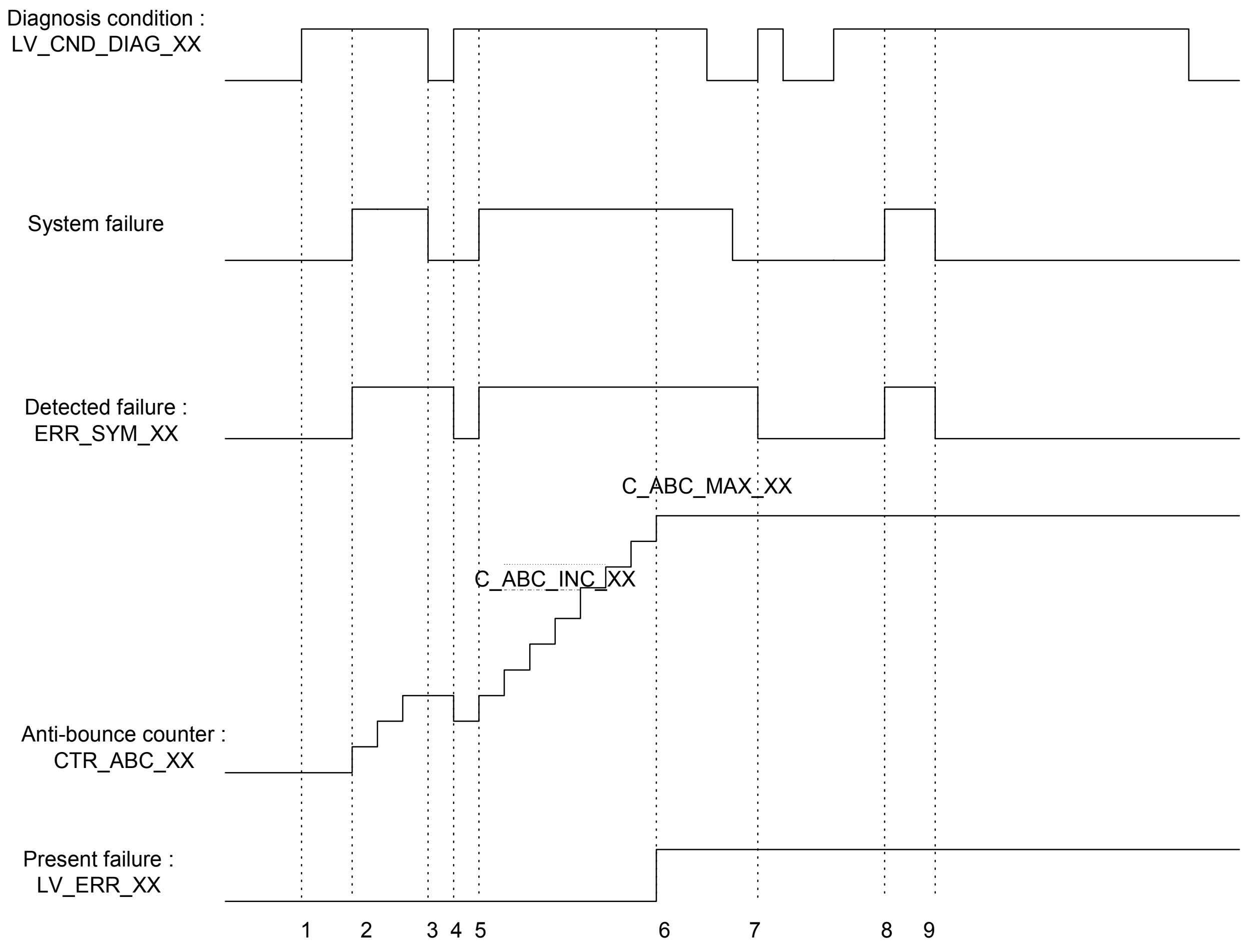
9: The diagnosis conditions are still fulfilled,
the failure isn't detected
and the failure remains present. ➔ CTR_ABC_XX =
LV_CDN_DIAG_XX = 1, CTR_ABC_XX – decrement
ERR_SYM_XX = 0. LV_ERR_XX = 1.

10: The diagnosis conditions are still fulfilled,
the failure isn't detected
and the failure is no more present. ➔ CTR_ABC_XX = 0
LV_CDN_DIAG_XX = 1, LV_ERR_XX = 0.
ERR_SYM_XX = 0.

Remark

- Decrement value = 1 if NLC_ABC_CAL_DEC_XX bit = 0
- Decrement value = C_ABC_DEC_XX if NLC_ABC_CAL_DEC_XX bit = 1.

2-The anti-bounce counter isn’t decremented (NLC_ABC_NOT_DEC_XX = 1):



1: The diagnosis conditions are fulfilled
but the failure isn’t detected ➔ CTR_ABC_XX remains at the value 0.
LV_CDN_DIAG_XX = 1, LV_ERR_XX = 0.
ERR_SYM_XX = 0.

2: The diagnosis conditions are fulfilled
and the failure is detected. ➔ CTR_ABC_XX = 0 + C_ABC_INC_XX
LV_CDN_DIAG_XX = 1, LV_ERR_XX = 0.
ERR_SYM_XX <> 0.

3: The diagnosis conditions aren’t fulfilled
and the failure remains detected. ➔ CTR_ABC_XX is frozen
LV_CDN_DIAG_XX = 0, LV_ERR_XX = 0.
ERR_SYM_XX <> 0.

4: The diagnosis conditions are fulfilled again
and the failure is no more detected. ➔ CTR_ABC_XX =
LV_CDN_DIAG_XX = 1, CTR_ABC_XX – decrement
ERR_SYM_XX = 0. LV_ERR_XX = 0.

5: The diagnosis conditions are still fulfilled
and the failure is detected again ➔ CTR_ABC_XX =
LV_CDN_DIAG_XX = 1, CTR_ABC_XX + C_ABC_INC_XX
ERR_SYM_XX <> 0. LV_ERR_XX = 0.

6: The diagnosis conditions are still fulfilled,
the failure is still detected
and the failure becomes present. ➔ CTR_ABC_XX = C_ABC_MAX_XX
LV_CDN_DIAG_XX = 1, LV_ERR_XX = 1.
ERR_SYM_XX <> 0.

7: The diagnosis conditions are fulfilled,
the failure isn’t detected
and the failure remains present. ➔ CTR_ABC_XX = C_ABC_MAX_XX
LV_CDN_DIAG_XX = 1, (counter not decrement)
ERR_SYM_XX = 0. LV_ERR_XX = 1.

.

8: The diagnosis conditions are still fulfilled,
the failure is detected again
and the failure remains present. ➔ CTR_ABC_XX = C_ABC_MAX_XX
LV_CDN_DIAG_XX = 1, LV_ERR_XX = 1.
ERR_SYM_XX <> 0.

.

.9: The diagnosis conditions are still fulfilled, the failure isn’t detected and the failure remains present.	➔ CTR_ABC_XX = C_ABC_MAX_XX
LV_CDN_DIAG_XX = 1,	(counter not decrement)
ERR_SYM_XX = 0.	LV_ERR_XX = 1.

Remark

- Decrement value = 1 if NLC_ABC_CAL_DEC_XX bit = 0
- Decrement value = C_ABC_DEC_XX if NLC_ABC_CAL_DEC_XX bit = 1.

Application conditions:

Initialization:	See "Initialisation of anti-bounce, statistical and multi-conditions filters" paragraph
Recurrence:	The recurrence is managed by the diagnosis.
Activation:	The activation is managed by the diagnosis for the anti-bounce operation or the anti-bounce bench mode.
Deactivation:	LV_ABC_INH = 1

Formula section:

Anti-bounce bench mode :

```
If(1)     (LC_ABC_BENCH = 1 and NLC_BENCH_MODE=1) and ((C_ABC_INC_XX = 0) or (C_ABC_INC_XX = FFH))
Then(1)
          If(2)     C_ABC_INC_XX = 0
          Then(2)
                    ERR_SYM_XX = 0
                    LV_ERR_XX = 0
                    CTR_ABC_XX = 0
          Else(2)   If(3)     C_ABC_INC_XX = FFH
                    Then(3)
                              ERR_SYM_XX = 0FH
                              LV_ERR_XX = 1
                              CTR_ABC_XX = C_ABC_MAX_XX
                    Endif(3)
          Endif(2)
Endif(1)
```

Anti-bounce operation (not in bench mode) :

```
If(1a)  (LV_CDN_DIAG_XX = 1) and not(     (LC_ABC_BENCH = 1 and
        NLC_BENCH_MODE=1) and ((C_ABC_INC_XX = 0) or (C_ABC_INC_XX =
        FFH)))
Then(1a)
   If(1b)  C_ABC_INC_XX = 0  { component XX not available }
   Then(1b)
        { the diagnosis result is forced to 'no error' }
        ERR_SYM_XX = 0
        LV_ERR_XX = 0
        CTR_ABC_XX = 0
   Else(1b)
      If(2)       ERR_SYM_XX <> 0
      Then(2)
                  { at each detected error, a treatment shall be done }
            If(3a)      NLC_ERR_DET_UPD = 1
                        and LC_ERR_DET_UPD=1
            Then(3a)
                        ACTION_ERRM_UpdErrDet( IN<XX>,
                        SYNCHRONIZATION<CALL> )
```

```
              Endif(3a)
              If(3b)  CTR_ABC_XX <> C_ABC_MAX_XX
              Then(3b)
                          CTR_ABC_XX = CTR_ABC_XX + C_ABC_INC_XX
                          If(4)   CTR_ABC_XX >= C_ABC_MAX_XX
                          Then(4)
                                  LV_ERR_XX = 1
                                  CTR_ABC_XX = C_ABC_MAX_XX
                          Endif(4)
              Endif(3b)
      Else(2)
              If(3)   (NLC_ABC_NOT_DEC_XX = 0 OR LV_ERR_XX = 0)
              Then(3)
                      If(4)   CTR_ABC_XX <> 0
                      Then(4) If(5)   NLC_ABC_CAL_DEC_XX = 0
                              Then(5)
                                      (without calibratable decrement)
                                      CTR_ABC_XX = CTR_ABC_XX – 1
                                      If(6)       CTR_ABC_XX = 0
                                      Then(6)
                                                  LV_ERR_XX = 0
                                      Endif(6)
                              Else(5)
                                      (with calibratable decrement)
                                      CTR_ABC_XX =      ( CTR_ABC_XX – C_ABC_DEC_XX )
                                      If(6)       CTR_ABC_XX <= 0
                                      Then(6)
                                                  LV_ERR_XX = 0
                                                  CTR_ABC_XX = 0
                                      Endif(6)
                              Endif(5)
                      Endif(4)
              Endif(3)
      Endif(2)
  Endif(1b)
Endif(1a)
```

27.6.3.2.2 Calculation of the end of diagnosis

Description:

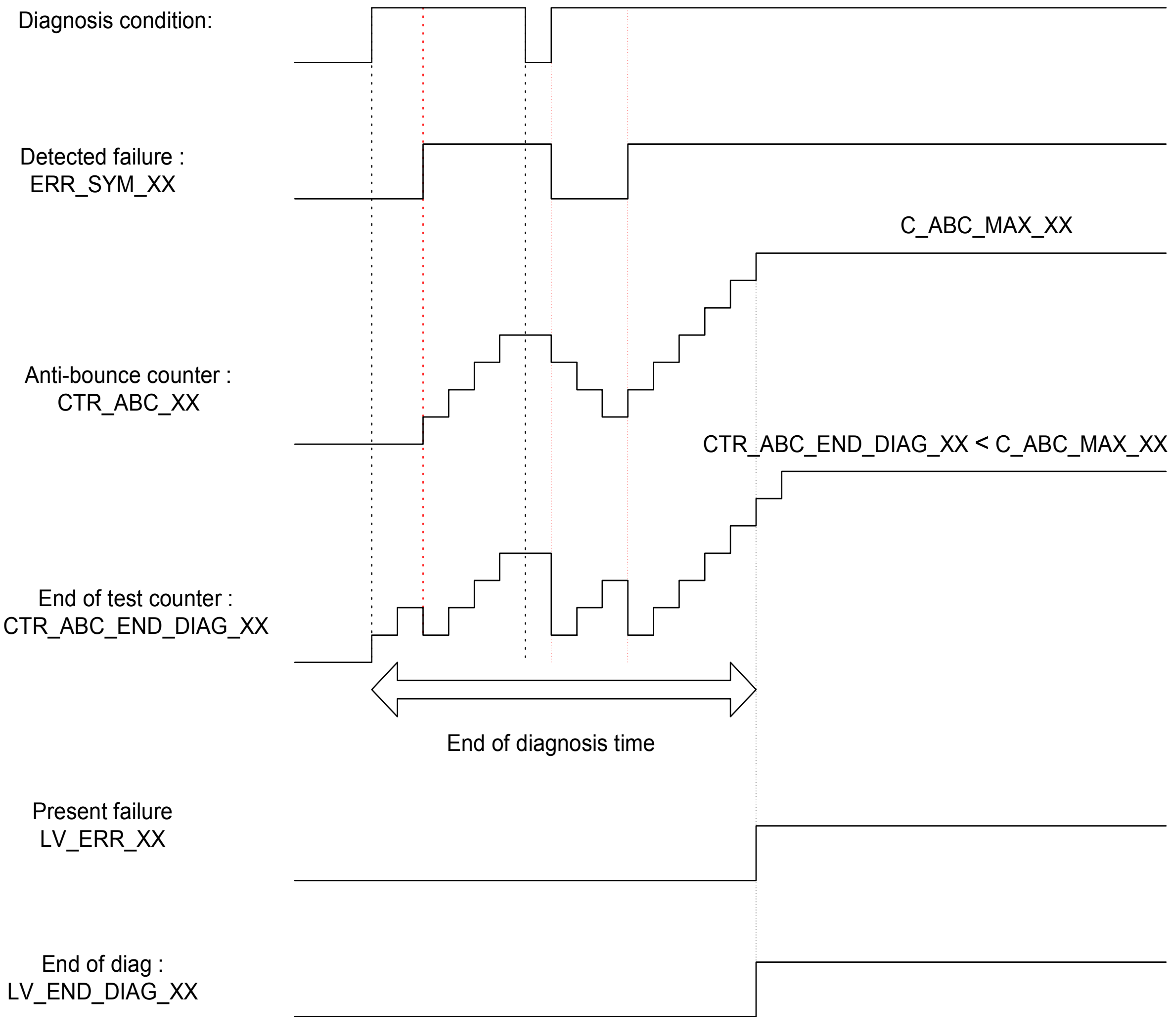
The end of diagnosis definition is set after a time defined like being the time necessary to evaluate the failure (present or not).

Signal flow diagram:

End of diagnosis algorithm illustration in standard case :

The end of diagnosis counter is reset at each transition of the detected failure.

When the failure becomes present : the end of diagnosis is set.

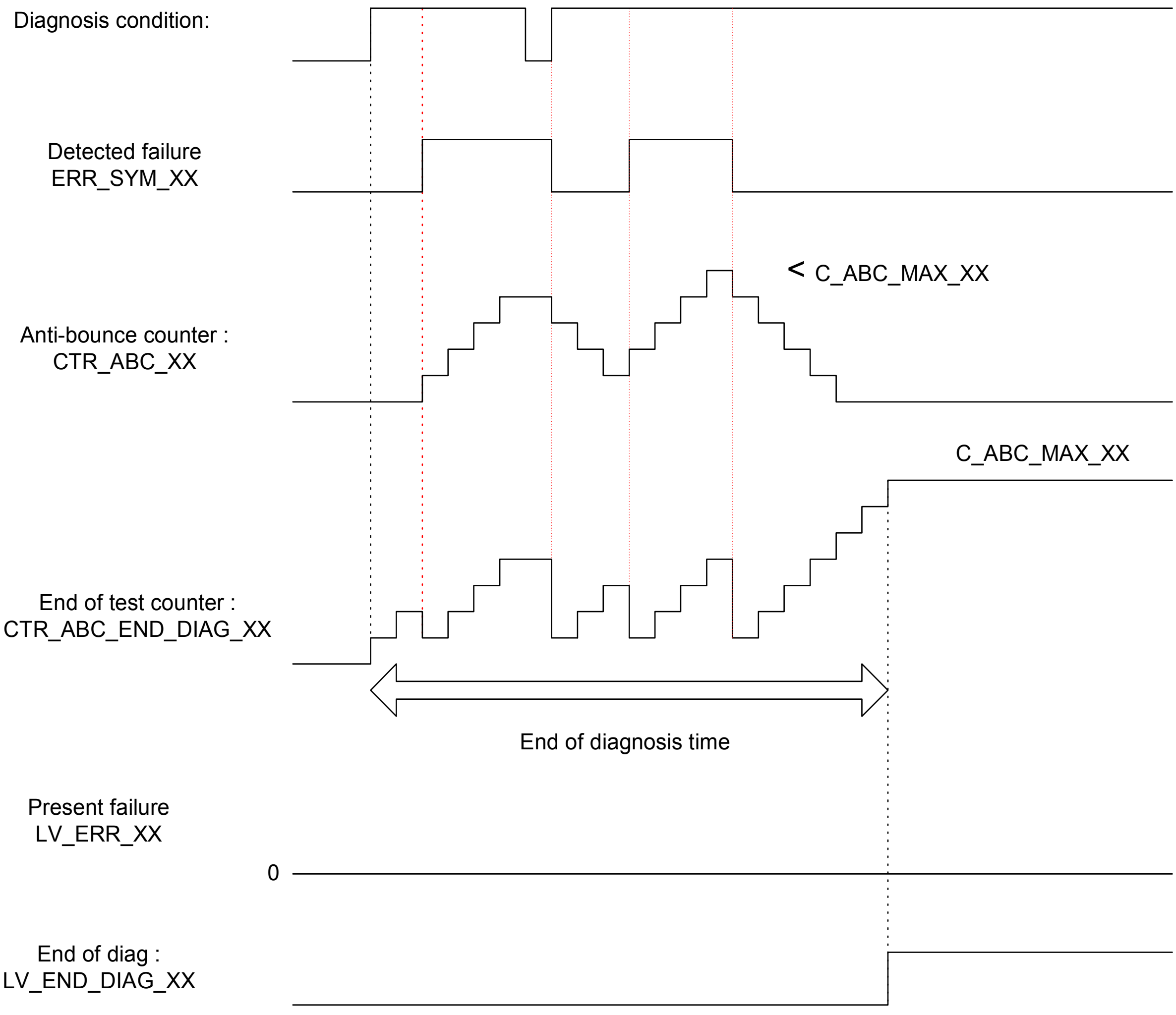


The detected failure will disappear and the anti-bounce counter won’t reach the maximum value then the failure won’t become present.

The end of diagnosis counter is reset at each transition of the detected failure.

When the detected failure is stable : the end of diagnosis counter is incremented until the maximum value while the diagnosis conditions are fulfilled.

When the end of diagnosis counter has reached the maximum value : the end of diagnosis is set.

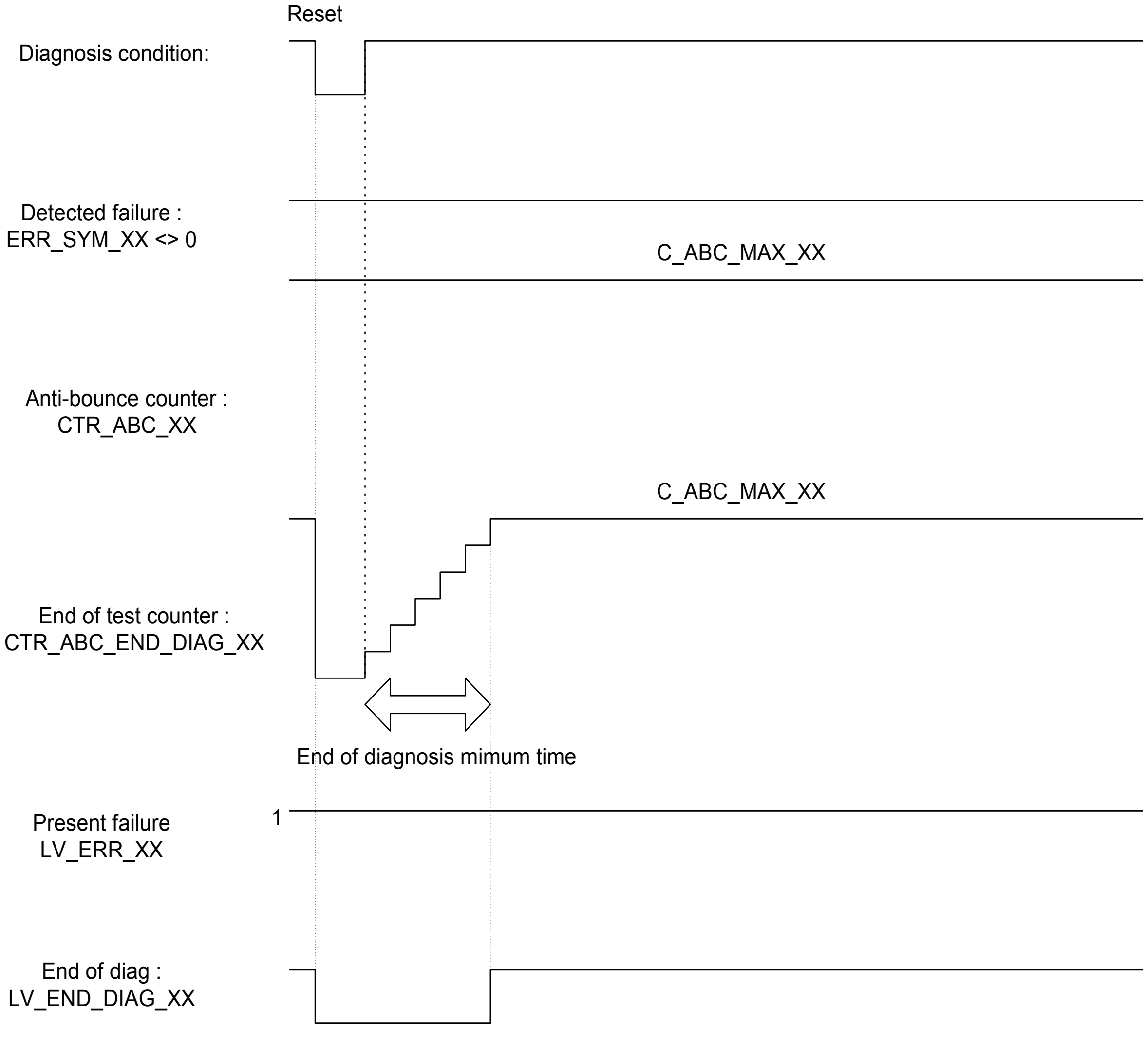


End of diagnosis algorithm illustration in case of reset event :

After the reset event, the detected failure remains present (stable) : the end of diagnosis counter is incremented until the maximum value while the diagnosis conditions are fulfilled.

When the end of diagnosis counter has reached the maximum value : the end of diagnosis is set.

The failure has been present and remains present after reset.



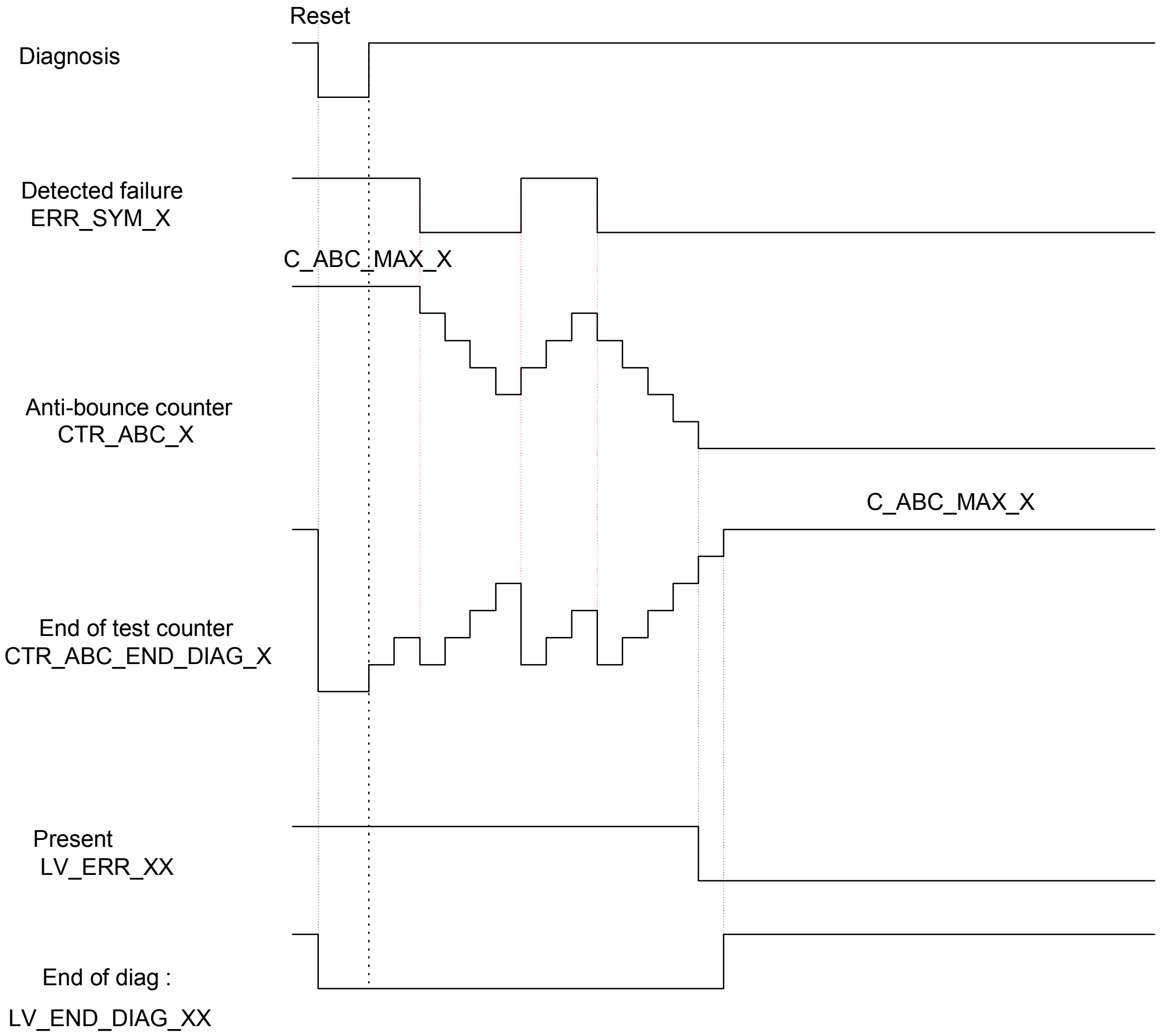
Before the reset, the failure was detected until the failure became present.

After the reset, the detected failure appears during a glitch so the end of diagnosis counter is reset at each transition of the detected failure,

When the detected failure disappears after all (stable) then

- the present failure disappears
- the end of diagnosis counter is incremented until the maximum value while the diagnosis conditions are fulfilled.

  When the end of diagnosis counter has reached the maximum value : the end of diagnosis is set.

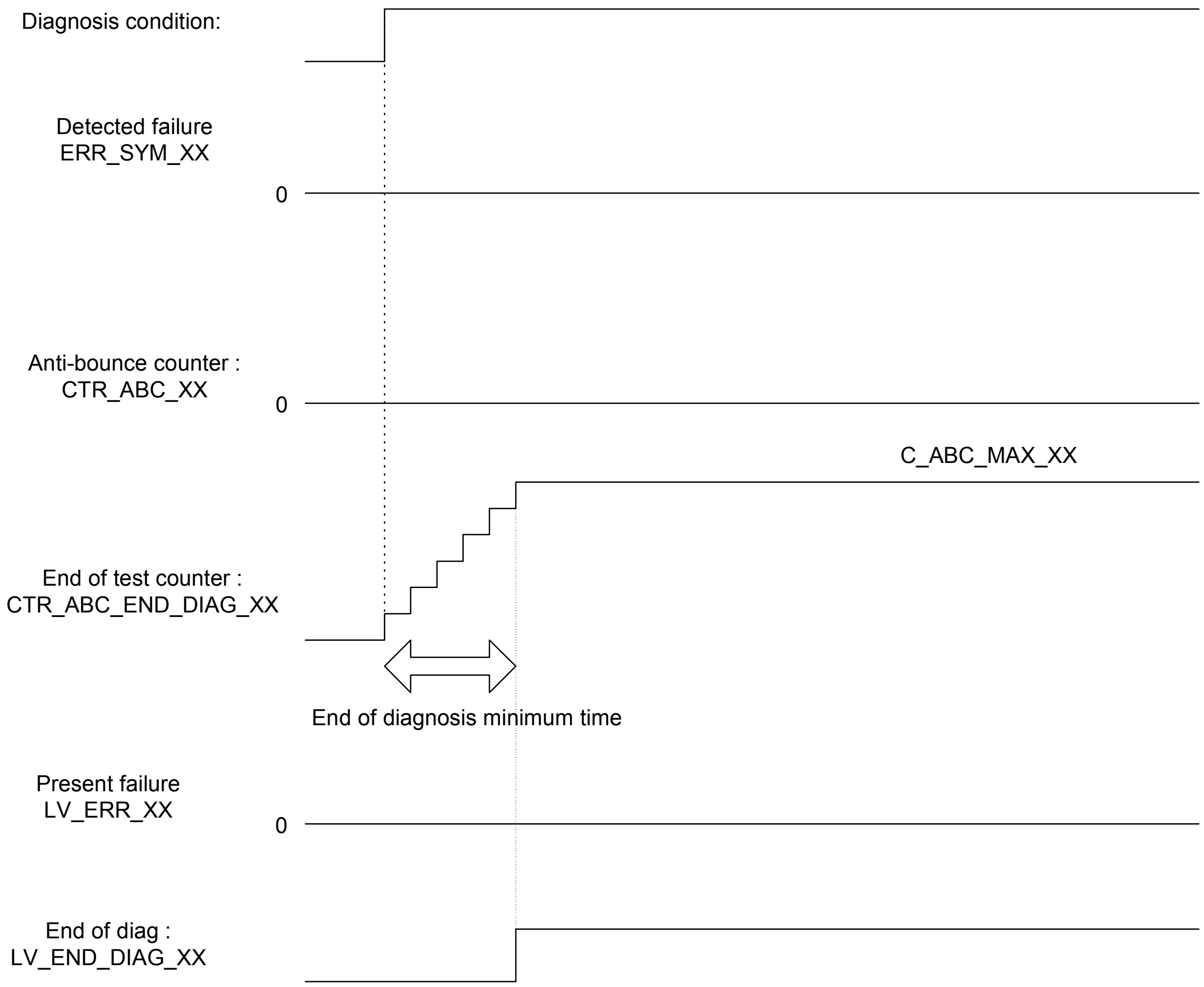


End of diagnosis algorithm illustration when the time to reach the end of diagnosis is minimum

The failure isn't detected (stable) : the end of diagnosis counter is incremented until the maximum value while the diagnosis conditions are fulfilled.

When the end of diagnosis counter has reached the maximum value : the end of diagnosis is set.

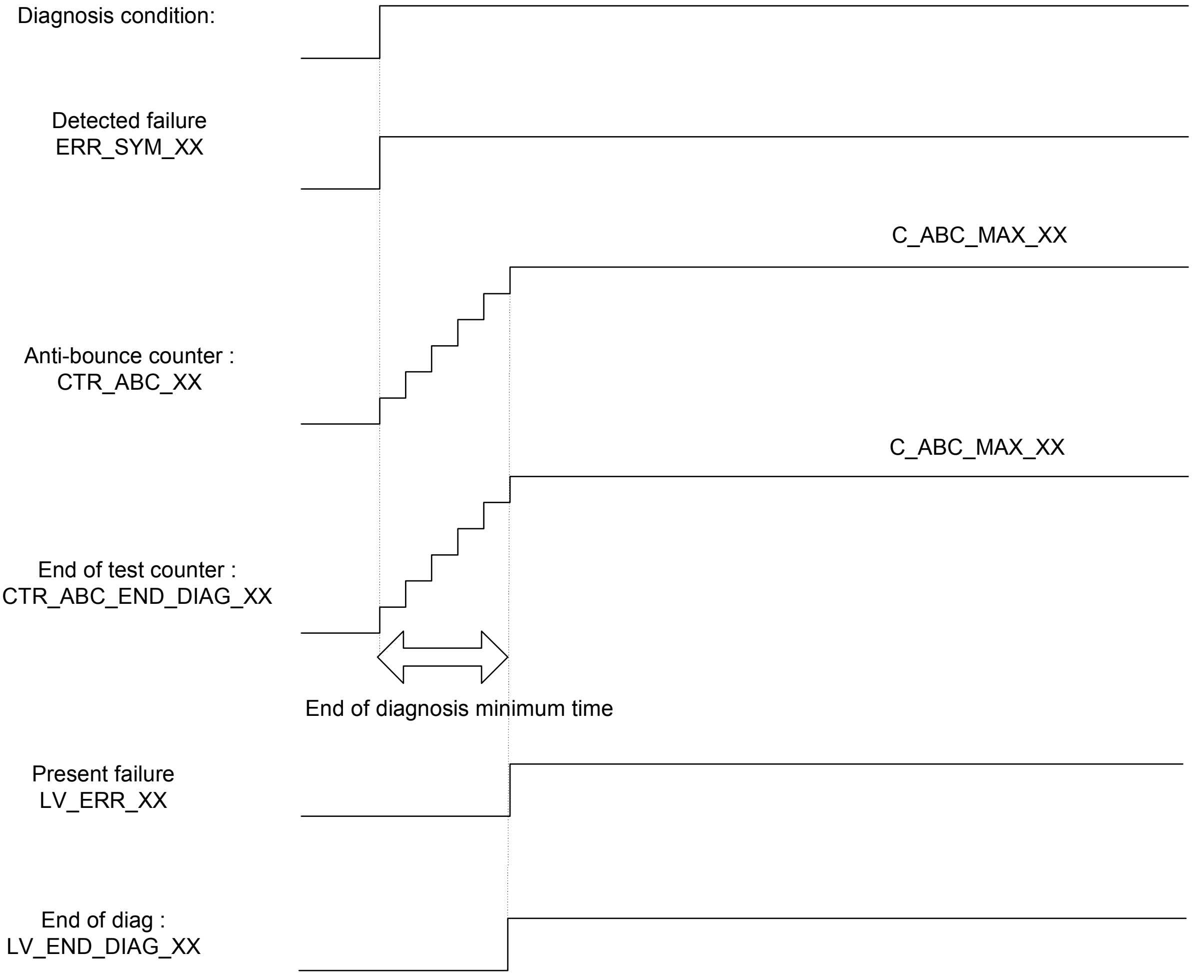
The end of diagnosis appears after the minimum time (the detected failure doesn't appear).



The failure is detected unchangingly (stable) : the end of diagnosis counter is incremented until the maximum value while the diagnosis conditions are fulfilled.

When the failure becomes present : the end of diagnosis is set.

The end of diagnosis and the failure appears after the minimum time (the detected failure is permanent).



Application conditions:

Initialization: See "Initialisation of anti-bounce, statistical and multi-conditions filters" paragraph

Recurrence: The recurrence is managed by the diagnosis.

Activation: The activation is managed by the diagnosis for the anti-bounce operation or the anti-bounce bench mode.

Deactivation: -

Formula section:

```
If(0) LV_ABC_INH = 0      {No inhibition of anti-bounce filtering}
Then(0)
    End of diagnosis in the bench mode:
    If(1)   (LC_ABC_BENCH = 1 and NLC_BENCH_MODE=1) and ((C_ABC_INC_XX = 0)
    or (C_ABC_INC_XX = FFH))
    Then(1)
            LV_CDN_DIAG_XX = 1
            LV_END_DIAG_XX = 1
            CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
    End of diagnosis in the normal mode:
    Else(1)
      If(2)       C_ABC_INC_XX = 0   { component XX not available }
      Then(2)
                  LV_END_DIAG_XX = 0
                  CTR_ABC_END_DIAG_XX = 0
      Else(2)
        If(3a)      (LV_CDN_DIAG_XX = 1) and (LV_END_DIAG_XX = 0)
        Then(3a)  If(4a)      ((ERR_SYM_PREV_XX ≠ 0) and (ERR_SYM_XX = 0))
                              or ((ERR_SYM__PREV_XX  = 0) and (ERR_SYM_XX ≠ 0))
                  Then(4a)
                              CTR_ABC_END_DIAG_XX = 0
                  Endif(4a)
                  If(4b)      (ERR_SYM_XX ≠ 0)
                  Then(4b)
                              CTR_ABC_END_DIAG_XX = (CTR_ABC_END_DIAG_XX+
                              C_ABC_INC_XX)
                  Else(4b)
                              If(5)     (NLC_ABC_CAL_DEC_XX = 0)
                              Then(5)
                                        CTR_ABC_END_DIAG_XX =
                                        (CTR_ABC_END_DIAG_XX + 1)
```

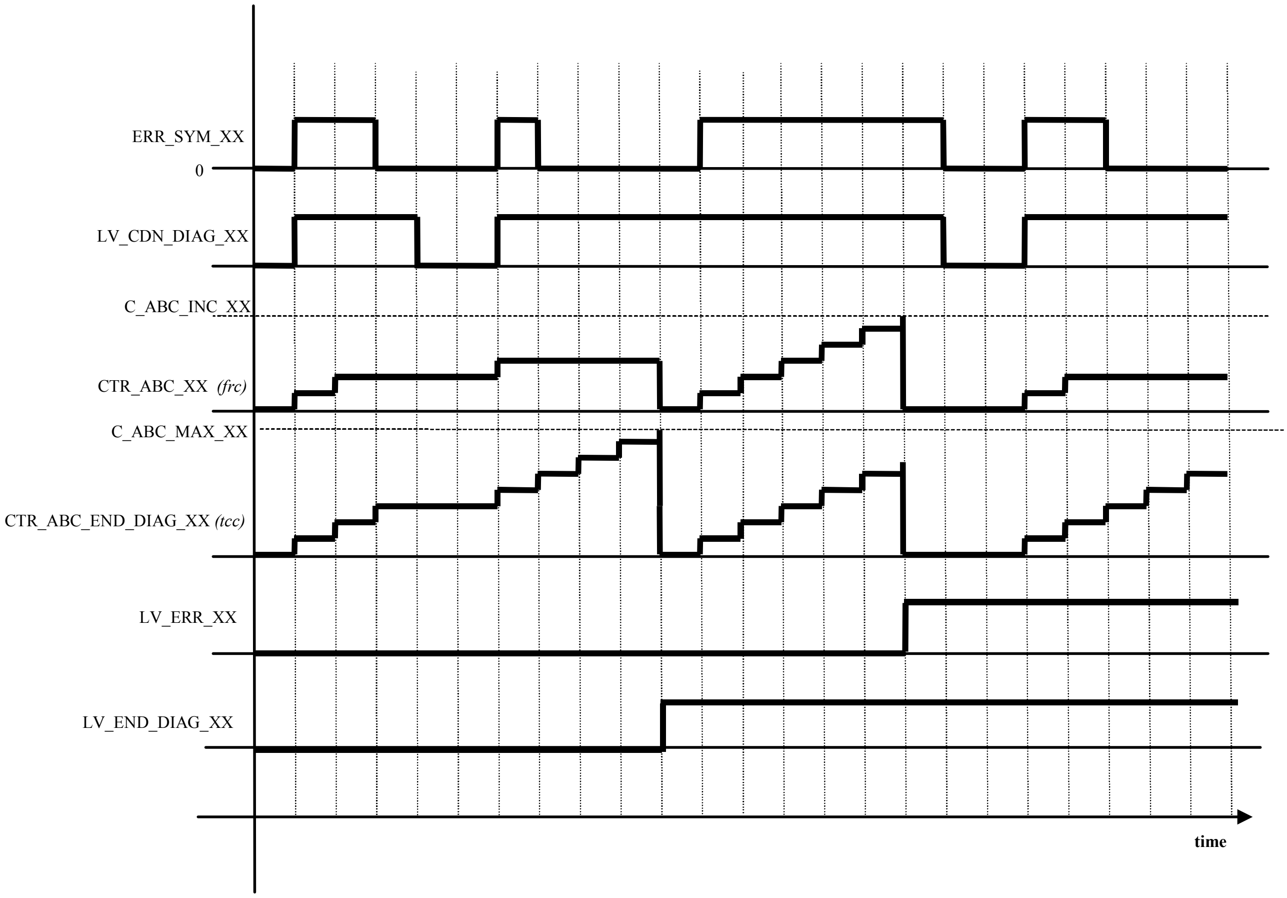
```
                              Else(5)
                                        CTR_ABC_END_DIAG_XX =
                                        (CTR_ABC_END_DIAG_XX + C_ABC_DEC_XX)
                              Endif(5)
                    Endif(4b)
                    If(4c)    CTR_ABC_END_DIAG_XX >= C_ABC_MAX_XX
                    Then(4c)  LV_END_DIAG_XX = 1
                              CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
                    Endif(4c)
                    ERR_SYM_PREV_XX = ERR_SYM_XX
          Endif(3a)
          If(3b)    LV_ERR_XX 0 → 1
          Then(3b)  LV_END_DIAG_XX = 1
          Endif(3b)
        Endif(2)
    Endif(1)
Else(0)
      {Inhibition of anti-bounce filtering}
      If(1)   (LV_CDN_DIAG_XX = 1
              or
              ((LC_ABC_BENCH = 1 and NLC_BENCH_MODE=1)
              and (C_ABC_INC_XX = 0) or (C_ABC_INC_XX = FFH))
      Then(1)
              {ERR_SYM_PREV_XX is updated at each diagnosis recurrence even
              during the inhibition, so that when the inhibition is removed
              ERR_SYM_PREV_XX is updated with the value calculated at the last
              diagnosis recurrence of the inhibition phase.}

              ERR_SYM_PREV_XX = ERR_SYM_XX
        Endif(1)
Endif(0)
```

27.6.3.3 Standard statistical filtering type

General information:

The purpose is to make an error detection by statistical evaluation.

An error is only detected, if a symptom is active for C_ABC_INC_XX times within the test period C_ABC_MAX XX, with test symptom condition fulfilled (LV_CDN_DIAG_XX = 1).



In this chapter,

- the data CTR_ABC_XX represents the frequency counter
- the calibration C_ABC_INC_XX represents the maximum for frequency counter of the processed symptom
- the data CTR_ABC_END_DIAG_XX represents the period counter increment
- the calibration C_ABC_MAX_XX represents the maximum for test counter of the processed symptom

The calibration C_ABC_MAX_XX must be greater than C_ABC_INC_XX.

Formula section:

Anti-bounce in the bench mode :

```
If      (LC_ABC_BENCH = 1 and NLC_BENCH_MODE=1)
        and
        ((C_ABC_INC_XX = 0) or (C_ABC_INC_XX = FFH))
Then
        If      C_ABC_INC_XX = 0
        Then
                ERR_SYM_XX = 0
                LV_ERR_XX = 0
                CTR_ABC_XX = 0
        Else
                If      C_ABC_INC_XX = FFH
                Then
                        ERR_SYM_XX = 0FH
                        LV_ERR_XX = 1
                        CTR_ABC_XX = C_ABC_INC_XX
                Endif
        Endif
Endif
```

Frequency counter increment:

```
If          LV_CDN_DIAG_XX = 1
            and ERR_SYM_XX ≠ 0
            and C_ABC_INC_XX ≠ 0
Then
            If      NLC_ERR_DET_UPD = 1
                    and LC_ERR_DET_UPD = 1
            Then
                    ACTION_ERRM_UpdErrDet( IN<XX>, SYNCHRONIZATION<CALL> )
            Endif
            CTR_ABC_XX = CTR_ABC_XX + 1
Endif
```

Period counter increment:

```
If          LV_CDN_DIAG_XX = 1
Then
            CTR_ABC_END_DIAG_XX = CTR_ABC_END_DIAG_XX + 1
Endif
```

Detection of the error state for the current symptom:

```
If          CTR_ABC_XX >= C_ABC_INC_XX
Then
            LV_ERR_XX = 1
            { The current symptom is entered to failure memory }
Endif
```

Detection of the error free symptom:

```
If          CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
            and CTR_ABC_XX < C_ABC_INC_XX
Then
            LV_ERR_XX = 0
Endif
```

End of the diagnostic period for statistic symptom:

```
If          CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
            or CTR_ABC_XX = C_ABC_INC_XX
Then
            CTR_ABC_XX = 0
            CTR_ABC_END_DIAG_XX = 0
            { End of diagnostic period for statistic symptom is reached }
            LV_END_DIAG_XX=1
Endif
```

27.6.4 Description for ACTION_ERRM_NoFilterSymptom :

Syntax :	ACTION_ERRM_NoFilterSymptom(IN<XX>, IN<lv_cdn_diag_XX>, IN<err_sym_XX>, IN<lv_err_set_XX>, IN<lv_err_reset_XX>, IN<lv_end_diag_XX>, OUT<LV_ERR_XX>)	
Parameter(in) :	XX	Type: index to identify the Diagnostic
	lv_cdn_diag_XX	Type: Diagnostic condition
	err_sym_XX	Status of failure
	lv_err_set_XX	Type: Boolean to set the failure
	lv_err_reset_XX	Type: Boolean to reset the failure
	lv_end_diag_XX	Type: Boolean (=1 when result available)
Parameter(out) :	LV_ERR_XX	Type: Boolean (=1 when failure)
Short Description :	This action returns the result on symptoms detected at each diagnostic recurrence, when no filter is used.	

27.6.5 System Description:

This action is used in case of diagnostics which don’t need any filtering will use the “NO type” filter.

LV_ERR_XX is managed as a **STD** anti-bounce filtering type (saved in NVMY at end of power latch, ...).

LV_END_DIAG_XX is managed by error management (reset at DC transition, ...).
ERR_SYM_XX is updated at each diagnostic recurrence, to fulfil the diagnostic result.

Formula section :

```
If LV_ABC_INH = 0      {No inhibition of anti-bounce filtering}
Then
    If lv_end_diag_XX = 1
    Then
            LV_END_DIAG_XX = 1
    Endif

    If lv_err_set_XX = 1
    Then
            LV_ERR_XX = 1
    Endif

    If lv_err_reset_XX = 1
    Then
            LV_ERR_XX = 0
    Endif
```

```
        If lv_cdn_diag_XX =1
        Then
                LV_CDN_DIAG_XX = 1
                ERR_SYM_XX = err_sym_xx
        Else
                LV_CDN_DIAG_XX = 0
                { ERR_SYM_XX is unchanged }
        Endif
Endif
```

27.6.6 Description for ACTION_ERRM_AbcFilterSymptomEnd:

Syntax:	ACTION_ERRM_AbcFilterSymptomEnd(	IN<XX>, IN< C_ABC_MAX_XX>, OUT< LV_ERR_XX >)
Parameter(in):	XX	Type: index to identify the Diagnostic
	C_ABC_MAX_XX	Maximum value of anti-bounce counter
Parameter(out):	LV_ERR_XX	Type: Boolean
Short Description:	This action erases diagnostic failure and sets immediately the end of diagnostic information in case of anti-bounce filter (STD, STD_INI, STC, MEM and MEM_INI)	

Formula section :

```
If   LV_ABC_INH = 0          {No inhibition of anti-bounce filtering}
Then
        LV_CDN_DIAG_XX = 1
        ERR_SYM_XX = NO_SYM
        LV_ERR_XX = 0
        CTR_ABC_XX = 0
        LV_END_DIAG_XX = 1
        CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
Endif
```

27.6.7 Description for ACTION_ERRM_NoFilterSymptomEnd:

Syntax:	ACTION_ERRM_NoFilterSymptomEnd(IN<XX>, OUT<LV_ERR_XX>)	
Parameter(in):	XX	Type: index to identify the Diagnostic
Parameter(out):	LV_ERR_XX	Type: Boolean
Short Description:	This action erases diagnostic failure and sets immediately the end of diagnostic information in case of no filter (NO)	

Formula section :

```
If   LV_ABC_INH = 0          {No inhibition of anti-bounce filtering}
Then
        LV_CDN_DIAG_XX = 1
```

```
        ERR_SYM_XX = NO_SYM
        LV_ERR_XX = 0
        LV_END_DIAG_XX = 1
    Endif
```

27.6.8 Description for ACTION_ERRM_AbcFilterReset:

Syntax:	ACTION_ERRM_AbcFilterReset(IN<XX>, OUT<LV_ERR_XX>)	
Parameter(in):	XX	Type: index to identify the Diagnostic
Parameter(out):	LV_ERR_XX	Type: Boolean
Short Description:	This action resets data filter in case of ABC filter type	

Formula section :

```
If LV_ABC_INH = 0      {No inhibition of anti-bounce filtering}
Then
    LV_CDN_DIAG_XX = 0
    ERR_SYM_XX = NO_SYM
    LV_ERR_XX = 0
    CTR_ABC_XX = 0
    LV_END_DIAG_XX is unchanged
    CTR_ABC_END_DIAG_XX is unchanged
Endif
```

27.6.9 Description for ACTION_ERRM_NoFilterReset:

Syntax:	ACTION_ERRM_NoFilterReset(IN<XX>, OUT<LV_ERR_XX>)	
Parameter(in):	XX	Type: index to identify the Diagnostic
Parameter(out):	LV_ERR_XX	Type: Boolean
Short Description:	This action resets data filter in case of no filter usage	

Formula section :

```
If LV_ABC_INH = 0      {No inhibition of anti-bounce filtering}
Then
    LV_CDN_DIAG_XX = 0
    ERR_SYM_XX = NO_SYM
    LV_ERR_XX = 0
    LV_END_DIAG_XX is unchanged
Endif
```

27.6.10 Description for ACTION_ERRM_GetLvErr:

Syntax:	ACTION_ERRM_GetLvErr(IN<XX>, OUT<LV_ERR_XX>)	
Parameter(in):	XX	Type: index to identify the Diagnostic
Parameter(out):	LV_ERR_XX	Type: Boolean
Short Description:	This action returns the failure status for the diagnostic XX.	

Formula section :

LV_ERR_XX = failure status for the diagnostic XX

27.6.11 Description for ACTION_ERRM_GetLvEndDiag:

Syntax:	ACTION_ERRM_GetLvEndDiag(IN<XX>, OUT<LV_END_DIAG_XX>)	
Parameter(in):	XX	Type: index to identify the Diagnostic
Parameter(out):	LV_END_DIAG_XX	Type: Boolean
Short Description:	This action returns the end of diagnostic flag for the diagnostic XX.	

Formula section :

LV_END_DIAG_XX = end of diagnostic flag for the diagnostic XX

27.6.12 Description for ACTION_ERRM_GetLvCdnDiag:

Syntax:	ACTION_ERRM_GetLvCdnDiag(IN<XX>, OUT<LV_CDN_DIAG_XX>)	
Parameter(in):	XX	Type: index to identify the Diagnostic
Parameter(out):	LV_CDN_DIAG_XX	Type: Boolean
Short Description:	This action returns the condition status for the diagnostic XX.	

Formula section :

LV_CDN_DIAG_XX = condition status for the diagnostic XX

27.6.13 Description for ACTION_ERRM_GetErrSym:

Syntax:	ACTION_ERRM_GetErrSym(IN<XX>, OUT<ERR_SYM_XX>)	
Parameter(in):	XX	Type: index to identify the Diagnostic
Parameter(out):	ERR_SYM_XX	Status of failure
Short Description:	This action returns the symptoms for the diagnostic XX.	

Formula section :

ERR_SYM_XX = symptoms for the diagnostic XX

27.6.14 CPU load optimisation mechanisms

General information:

This section describes the mechanism of optimisation. The principle is to suspend the filter algorithm execution when it is not necessary.

Formula section:

```
If      NLC_BENCH_MODE = 0
Then
        { the bench mode isn't available whatever LC_ABC_BENCH value }
        the CPU load is reduced
Else
        { the bench mode is available if LC_ABC_BENCH = 1 }
        less CPU load reduction than with NLC_BENCH_MODE = 0
Endif
```

Case 1 : the diagnostic uses an anti-bounce filtering algorithm

```
If      LV_ERR_XX = 0
        and LV_END_DIAG_XX = 1
        and ERR_SYM_XX = 0
        and CTR_ABC_XX = 0
Then
        { it is not necessary to call the Error Management (including the filtering) }
        LV_END_DIAG_XX = 1    { unchanged }
        LV_ERR_XX = 0         { unchanged }
Endif
```

Case 2 : the diagnostic doesn't use an anti-bounce filtering algorithm

```
If      ( LV_ERR_XX = 0 and LV_END_DIAG_XX = 1 and ERR_SYM_XX = 0 )
Then
        { it is not necessary to call the Error Management }
        LV_END_DIAG_XX = 1    { unchanged }
        LV_ERR_XX = 0         { unchanged }
Endif
```

27.6.15 Synchronisation between Anti-bounce algorithm and Error Management

General information:

This section describes the mechanism of synchronisation of the failure filtering with the Error Management core. The principle is to call the Error Management upon events.

LV_END_DIAG_XX 0 → 1 transition

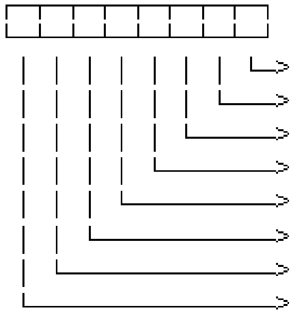
LV_ERR_XX 0 → 1 transition

LV_ERR_XX 1 → 0 transition

This mechanism is followed whatever the filter (type STD, STD_INI, DEC_CAL, MEM, MEM_INI, STC) but also for diagnoses which don't use any filter (type NO).

27.6.16 Configuration and calibration data

NC_ABC_CONF_FCT_DIAG_XX :



bit 0 : NLC_ABC_NOT_SAVE_XX
bit 1 : NLC_ABC_INI_IGK_XX
bit 2 : NLC_ABC_INI_VALUE_XX
bit 3 : NLC_ABC_NOT_DEC_XX
bit 4 : NLC_ABC_CAL_DEC_XX
bit 5 : NLC_ABC_STC_XX
bit 6 : Not used
bit 7 : Not used

Bits definition

- NLC_ABC_NOT_SAVE_XX : if this bit is set to 0 the anti-bounce context (CTR_ABC_XX and LV_ERR_XX) is saved at the end of power latch, only when the failure is in dynamic memory, and restored at the event Reset.
Limitation : the saved configuration is only available if the NVMY manager take into account the filter environment data (LV_ERR_XX, CTR_ABC_XX).

- NLC_ABC_INI_IGK_XX : LV_ERR_XX and CTR_ABC_XX are reset at the transition LV_IGK 0 -> 1 and at the event reset.

- NLC_ABC_INI_VALUE_XX : if this bit is set to 1 then the anti-bounce counter is initialised with C_ABC_MAX_XX – C_ABC_INC_XX if there was a failure between the last and this initialisation (NLC_ABC_NOT_SAVE_XX = 0 to test the failure before this initialisation) else the anti-bounce counter is initialised with 0 value.

- NLC_ABC_NOT_DEC_XX : if this bit is set to 0 then the anti-bounce counter can be decremented if the condition for decrement are fulfilled. If this bit is set to 1 then the anti-bounce counter can be decremented if the condition for decrement are fulfilled until the anti-bounce counter hasn't reached the maximum value; when the anti-bounce counter has reached the maximum value then the anti-bounce counter and the failure remains (the anti-bounce counter can be decremented anymore) until a transition LV_IGK 0 → 1 or reset following the NLC_ABC_INI_IGK_XX bit configuration (If the NLC_ABC_NOT_DEC_XX bit is set to 1 then NLC_ABC_NOT_SAVE_XX must be set to 1).

- NLC_ABC_CAL_DEC_XX : if this bit is set then the anti-bounce decrement is calibratable, the calibration is named C_ABC_DEC_XX else the anti-bounce decrement is equal to 1 and there isn't calibration.

- NLC_ABC_STC_XX : if this bit is set then the filter used is the statistical filter else the filter used is the anti-bounce filter (Standard configuration).

Predefined filter type :

	NLC_ABC_STC_XX	NLC_ABC_CAL_DEC_XX	NLC_ABC_NOT_DEC_XX	NLC_ABC_INI_VALUE_XX	NLC_ABC_INI_IGK_XX	NLC_ABC_NOT_SAVE_XX
STD_INI	0	0	0	0	1	1
STD	0	0	0	0	0	0
MEM	0	0	1	0	1	1
MEM_INI	0	0	1	1	1	0
DEC_CAL	0	1	0	0	0	0
STC	1	0	0	0	1	1
NO	0	0	0	0	0	0

The **standard configuration with initialisation (STD_INI)** is
NC_ABC_CONF_FCT_DIAG_XX = 3H (default configuration)

The data CTR_ABC_XX and LV_ERR_XX are reset at the transition LV_IGK 0 → 1 and reset (NLC_ABC_NOT_SAVE_XX = 1, NLC_ABC_INI_IGK_XX = 1) and the anti-bounce counter is initialised to 0 value (NLC_ABC_INI_VALUE_XX = 0).

The anti-bounce counter may be decremented (NLC_ABC_NOT_DEC_XX = 0).
The decrement value is equal to 1 (NLC_ABC_CAL_DEC_XX = 0).
The filter used is the Anti-bounce (NLC_ABC_STC_XX = 0).

The **statistical configuration (STC)** is NC_ABC_CONF_FCT_DIAG_XX = 23H

The data CTR_ABC_XX and LV_ERR_XX are reset at the transition LV_IGK 0 → 1 and reset (NLC_ABC_NOT_SAVE_XX = 1, NLC_ABC_INI_IGK_XX = 1) and the frequency counter (CTR_ABC_XX) is initialised to 0 value (NLC_ABC_INI_VALUE_XX = 0).

The filter used is the statistical (NLC_ABC_STC_XX = 1).

The **standard configuration (STD)** is NC_ABC_CONF_FCT_DIAG_XX = 0H

The data CTR_ABC_XX and LV_ERR_XX are saved (NLC_ABC_NOT_SAVE_XX = 0, NLC_ABC_INI_IGK_XX = 0, NLC_ABC_INI_VALUE_XX = 0).

The anti-bounce counter may be decrement (NLC_ABC_NOT_DEC_XX = 0).

The decrement value is equal to 1 (NLC_ABC_CAL_DEC_XX = 0).

The **memory configuration (MEM)** is NC_ABC_CONF_FCT_DIAG_XX = BH

The data CTR_ABC_XX and LV_ERR_XX are reset at the transition LV_IGK 0 → 1 and reset (NLC_ABC_NOT_SAVE_XX = 1, NLC_ABC_INI_IGK_XX = 1) and the anti-bounce counter is initialised to 0 value (NLC_ABC_INI_VALUE_XX = 0).

The anti-bounce counter can be decremented if the condition for decrement are fulfilled until the anti-bounce counter hasn't reached the maximum value; when the anti-bounce counter has reached the maximum value then the anti-bounce counter and the failure remains until a transition LV_IGK 0 → 1 or reset (NLC_ABC_NOT_DEC_XX = 1).

The decrement value is equal to 1 (NLC_ABC_CAL_DEC_XX = 0).

The **memory configuration (MEM_INI)** is NC_ABC_CONF_FCT_DIAG_XX = EH

The data CTR_ABC_XX and LV_ERR_XX are initialised at the transition LV_IGK 0 → 1 and reset (NLC_ABC_INI_IGK_XX = 1). The anti-bounce counter is initialised to the C_ABC_MAX_XX – C_ABC_INC_XX value (NLC_ABC_INI_VALUE_XX = 1) only if there was a failure between the last and this initialisation (NLC_ABC_NOT_SAVE_XX = 0 to test the failure before this initialisation) else the anti-bounce counter is initialised with 0 value. The data LV_ERR_XX is initialised with 0 value.

The anti-bounce counter can be decremented if the condition for decrement are fulfilled until the anti-bounce counter hasn't reached the maximum value; when the anti-bounce counter has reached the maximum value then the anti-bounce counter and the failure remains until a transition LV_IGK 0 → 1 or reset (NLC_ABC_NOT_DEC_XX = 1).

The decrement value is equal to 1 (NLC_ABC_CAL_DEC_XX = 0).

The **decrement calibratable configuration (DEC_CAL)** is set with NC_ABC_CONF_FCT_DIAG_XX = 10H

The data CTR_ABC_XX and LV_ERR_XX are saved (NLC_ABC_NOT_SAVE_XX = 0, NLC_ABC_INI_IGK_XX = 0, NLC_ABC_INI_VALUE_XX = 0).

The anti-bounce counter may be decrement (NLC_ABC_NOT_DEC_XX = 0).

The decrement value is calibratable (NLC_ABC_CAL_DEC_XX = 1).

The standard configuration is NLC_ABC_INI_DC_END_DIAG = 0 and NLC_ERR_DET_UPD = 0 (default configuration).

Some functionalities (e.g. : to visualise the sporadic error of all diagnosis using this anti-bounce algorithm) aren't used when the failure is detected.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_ABC_BENCH	1	0...1H	0...1	1	[-]
Logical constant to manage the anti-bounce bench mode					
LC_ERR_DET_UPD	1	0...1H	0...1	1	[-]
Logical constant to manage some functionality as the visualisation of the sporadic error					
C_ABC_DEC_XX	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement for XX diagnosis					
C_ABC_INC_XX	1	0...FFH	0...255	1	[-]
XX diagnosis anti-bounce counter increment (for anti-bounce filtering) or frequency counter maximum value (for statistical filtering)					
C_ABC_MAX_XX	1	1...FFH	1...255	1	[-]
XX diagnosis anti-bounce counter maximum value (for anti-bounce filtering) or threshold of period counter for failure detection (for statistical filtering)					

Remark :

The standard calibration is LC_ABC_BENCH = 0 **and** NLC_BENCH_MODE=0 (default calibration) then the bench mode isn't managed.

If the diagnosis doesn't manage calibratable decrement then the C_ABC_DEC_XX calibration isn't defined in calibration data (NLC_ABC_CAL_DEC_XX = 0).

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_ABC_CONF_FCT_DIAG_XX	1	0...FFH	0...255	1	[-]
Anti-bounce algorithm configuration of each XX diagnosis					
NLC_ABC_INI_DC_END_DIAG	1	0...1H	0...1	1	[-]
Initialisation of LV_END_DIAG_XX at LV_DC transition or not					
NLC_BENCH_MODE	1	0...1H	0...1	1	[-]
Bench mode configuration (0: the bench mode is removed, max CPU optimisation is reached)					

Configuration detailed description:

NLC_ABC_INI_DC_END_DIAG

1 : Initialisation of LV_END_DIAG_XX at LV_DC 1→0 transition

0 : Initialisation of LV_END_DIAG_XX at LV_IGK 0→1 transition or ECU reset

27.7 Anti-bounce algorithm (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_INTM_DIAG_INST[NC_NR_ERR_INTM]	V/S	0...FFFFH	0...65535	1	[-]
Array of diagnosis instances with intermittent failure					
ERR_INTM_CTR_FRC[NC_NR_ERR_INTM]	V/S	0...FFH	0...255	1	[-]
Frequency of reappearence of failure stored in ERR_INTM_DIAG_INST					
CTR_ERR_INTM_NR	V	0...FFH	0...255	1	[-]
Number of failures stored in intermittent failure array					
LV_ABC_INH	V/O	0...1H	0...1	1	[-]
Anti-bounce filterings inhibition					

Export actions:

ACTION_ERRM_InhibitFiltering (IN<InhibitFilter>)
This action shall be used to inhibit anti-bounce filtering calculations

FUNCTION DESCRIPTION:

General information:

Intermittent failures are neither visualized nor stored with the following configuration :
NLC_ERR_DET_UPD = 00h

Application conditions:

Initialisation: ERR_INTM_DIAG_INST[0...NC_NR_ERR_INTM –1] = 0
ERR_INTM_CTR_FRC[0...NC_NR_ERR_INTM –1] = 0
CTR_ERR_INTM_NR = 0

Recurrence: -

Activation: at ECU reset

Formula section:

No action to be performed with this configuration.

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_ERR_INTM	1	1...FFH	1...255	1	[-]
Maximum number of diagnosis instances to be listed in ERR_INTM_DIAG_INST					

27.7.1 Anti-bounce filtering inhibition

Description:

In addition, LV_ABC_INH flag must be securised to avoid any undesired inhibition.

To avoid failure recording during specific ECU phases (e.g. reprogramming,...), diagnoses are inhibited. To realize this inhibition, anti-bounce filtering calculations are frozen.

This inhibition is generally used during reprogramming phase.

Syntax:	ACTION_ERRM_InhibitFiltering (IN <InhibitFilter>)
Parameter(in):	InhibitFilter filter inhibition
Parameter(out:	No parameter
Short Description:	This API permits to activate or deactivate the anti-bounce calculation.

Application conditions:

Initialisation:	at ECU reset LV_ABC_INH = 0 (default value : Anti-bounce calculation is not inhibited)
Recurrence:	-
Activation:	at action request

Formula section:

LV_ABC_INH = 0

27.8 Multi-condition debounce algorithm

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
CTR_ABC_XX	V/O/S	0...FFH	0...255	1	[-]
anti bounce counter of diagnosis XX					
CTR_ABC_END_DIAG_XX	V/O	0...FFH	0...255	1	[-]
Counter for end of diagnosis XX generation					
CDN_DIAG_PREV_XX	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Diagnostic condition of last failure symptom					

Input data:

LC_ABC_BENCH	NC_ABC_CONF_FCT_DIAG_XX	NLC_ABC_INI_DC_END_DIAG	NLC_ERR_DET_UPD
LC_ERR_DET_UPD	LV_IGK	LV_ABC_INH	NLC_BENCH_MODE

Import actions:

ACTION_ERRM_UpdErrDet (IN<XX>)
This action indicates each occurrence of the detected error

Export actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>,OUT<LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result
ACTION_ERRM_MultiFilterSymEnd(IN< XX >, IN<C_ABC_MAX_XX>, OUT<LV_ERR_XX>)
This action erases diagnostic failure and sets the end of diagnostic in case of multi-conditions filter type
ACTION_ERRM_MultiFilterReset(IN< XX >,OUT<LV_ERR_XX>)
This action resets data filter in case of multi-conditions filter type

General information:

Description:

According the diagnosis condition information CDN_DIAG_XX and the diagnosis symptom (raw value) information the symptom calculation ERR_SYM_XX is performed. Then the failure symptom is debounced to set the present failure LV_ERR_XX.

A special readiness calculation is done to set the end of diagnosis information. LV_END_DIAG_XX is set only if there was enough time to make a diagnosis of all symptoms or a failure gets present.

Note:

To set the LV_END_DIAG_XX is necessary to have the conditions for all symptoms at the same time for a given time.

Restrictions:

In case of multiple symptoms at the same time, only one symptom will be taken into account according the following priority rule: SYM_0 ==> SYM_3 ; Priority max ==> Priority Min

Only the last symptom present will be taken into account in this algorithm, so in case of symptom change, only the last symptom will be used to increment and decrement the counters.

Application conditions:

Initialization: At ECU reset and LV_IGK 0 → 1 transition

CDN_DIAG_PREV_XX = 0

Other data initialization

ERR_SYM_XX = *Refer to “Antibounce algorithms” chapter for initialization*
CTR_ABC_XX = *Refer to “Antibounce algorithms” chapter for initialization*
LV_ERR_XX = *Refer to “Antibounce algorithms” chapter for initialization*
CTR_ABC_END_DIAG_XX = *Refer to “Antibounce algorithms” chapter for initialization*

27.8.1 Multi-condition anti-bounce filter algorithm description

Description for ACTION_ERRM_FilterMulticondition:

Syntax : ACTION_ERRM_FilterMulticondition (IN<XX>, IN<CDN_DIAG_XX>, IN<ERR_DIAG_XX>, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>, OUT<LV_ERR_XX>)

Parameter(in):	XX	Type: index to identify the Diagnostic
	CDN_DIAG_XX	Type: Byte carrier (Each symptom condition)
	ERR_DIAG_XX	Type: Byte carrier (Raw value symptom)
	C_ABC_INC_XX	Type: Calibration byte for the increment value
	C_ABC_MAX_XX	Type: Calibration byte for the maximum value
Parameter(out):	LV_ERR_XX	Type: Boolean (=1 when failure)

Short Description: This action returns information necessary for the diagnostic and the error management.

Formula section:

Anti-bounce bench mode :

```
If(0) LV_ABC_INH = 0
Then
If(1) (LC_ABC_BENCH = 1 and NLC_BENCH_MODE=1) and ((C_ABC_INC_XX = 0) or (C_ABC_INC_XX = FFH))
Then(1)
        If(2)   C_ABC_INC_XX = 0
        Then(2)
                {Case force no failure}
                LV_ERR_XX = 0
                ERR_SYM_XX =  0
                CTR_ABC_XX = 0
                LV_END_DIAG_XX = 1
                CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
        Else(2)
                If(3)   C_ABC_INC_XX = FFH
                Then(3)
                        {Forced failure}
                        LV_ERR_XX = 1
                        ERR_SYM_XX = 0Fh
                        CTR_ABC_XX = C_ABC_MAX_XX
                        LV_END_DIAG_XX = 1
                        CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
                Endif(3)
        Endif(2)
Endif(1)
```

Symptom calculation:

```
{Erase not available bit}

CDN_DIAG_XX = CDN_DIAG_XX and NC_NR_SYM_XX
ERR_DIAG_XX = ERR_DIAG_XX and NC_NR_SYM_XX

If(1)       CDN_DIAG_XX ≠ 0  {Diagnosis of at least one symptom is valid}
Symptom SYM_0 calculation:
Then(1)
            If(2)           bit 0 of CDN_DIAG_XX = 1
                            and
                            bit 0 of ERR_DIAG_XX = 1
            Then(2)
                            ERR_SYM_XX = 1H
                            {Memorization of last failure symptom number}
                            CDN_DIAG_PREV_XX = 1H

Symptom SYM_1 calculation:
            Else(2)
                  If(3)     bit 1 of CDN_DIAG_XX = 1
                            and
                            bit 1 of ERR_DIAG_XX = 1
                  Then(3)
                            ERR_SYM_XX = 2H
                            {Memorization of last failure symptom number}
                            CDN_DIAG_PREV_XX = 2H

Symptom SYM_2 calculation:
                  Else(3)
                            If(4)           bit 2 of CDN_DIAG_XX = 1
                                            and
                                            bit 2 of ERR_DIAG_XX = 1
                            Then(4)
                                            ERR_SYM_XX = 4H
                                            {Memorization of last failure symptom number}
                                            CDN_DIAG_PREV_XX = 4H

Symptom SYM_3 calculation:
                            Else(4)
                                  If(5)     bit 3 of CDN_DIAG_XX = 1
                                            and
                                            bit 3 of ERR_DIAG_XX = 1
                                  Then(5)
                                            ERR_SYM_XX = 8H
                                            {Memorization of last failure symptom number}
                                            CDN_DIAG_PREV_XX = 8H
```

Symptom calculation for ‘No symptom detected’:

```
                        Else(5)
                            If      bit 0 of CDN_DIAG_XX = 1
                            Then
                                    bit 0 of ERR_SYM_XX = 0
                            Endif
                            If      bit 1 of CDN_DIAG_XX = 1
                            Then
                                    bit 1 of ERR_SYM_XX = 0
                            Endif
                            If      bit 2 of CDN_DIAG_XX = 1
                            Then
                                    bit 2 of ERR_SYM_XX = 0
                            Endif
                            If      bit 3 of CDN_DIAG_XX = 1
                            Then
                                    bit 3 of ERR_SYM_XX = 0
                            Endif
                        Endif(5)
                    Endif(4)
                Endif(3)
            Endif(2)
Endif(1)
If(1)       (CDN_DIAG_PREV_XX = 0 and CDN_DIAG_XX ≠ 0)
            or      (bit 0 of CDN_DIAG_PREV_XX = 1 and bit 0 of CDN_DIAG_XX = 1)
            or      (bit 1 of CDN_DIAG_PREV_XX = 1 and bit 1 of CDN_DIAG_XX = 1)
            or      (bit 2 of CDN_DIAG_PREV_XX = 1 and bit 2 of CDN_DIAG_XX = 1)
            or      (bit 3 of CDN_DIAG_PREV_XX = 1 and bit 3 of CDN_DIAG_XX = 1)
Then(1)
            LV_CDN_DIAG_XX = 1
Else(1)
            {Anti-bounce value is frozen}
            LV_CDN_DIAG_XX = 0
Endif(1)
```

Anti-bounce of diagnosis instance XX:

```
If(1)       LV_CDN_DIAG_XX = 1
```

Incrementation of antibounce:

```
Then(1)
            If(2a)  ERR_SYM_XX ≠ 0
            Then(2a) If   C_ABC_INC_XX ≠ 0
                    Then  { at each detected error, a treatment may be done }
                          If      NLC_ERR_DET_UPD = 1 and LC_ERR_DET_UPD = 1
                          Then
                                  ACTION_ERRM_UpdErrDet(IN<XX>,
                                  SYNCHRONIZATION<CALL>)
                          Endif

                          If      CTR_ABC_XX ≠ C_ABC_MAX_XX
                          Then
                                  CTR_ABC_XX = CTR_ABC_XX + C_ABC_INC_XX
                                  If      CTR_ABC_XX ≥ C_ABC_MAX_XX
                                  Then
                                          LV_ERR_XX = 1
                                          CTR_ABC_XX = C_ABC_MAX_XX
                                  Endif
                          Endif
                    Endif
            Else (2a)
                    If    CTR_ABC_XX ≠ 0
                    Then
                          {Decrementation of antibounce}
                          CTR_ABC_XX = CTR_ABC_XX – 1
                          If      CTR_ABC_XX = 0           { minimum value }
                          Then
                              LV_ERR_XX = 0
                              {Reset of last failure symptom memorized}
                              CDN_DIAG_PREV_XX = 0H
                          Endif
                    Endif
            Endif(2a)
```

Readiness calculation for diagnosis instance XX:

```
            If(2b)        CTR_ABC_END_DIAG_XX < C_ABC_MAX_XX
```

End of diagnosis is set, if failure gets present:

```
        Then(2b) If   LV_ERR_XX = 1              { failure XX is present }
                 Then LV_END_DIAG_XX = 1
                      CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
```

End of diagnosis is set, if all symptoms can be detected after a specified time:

```
                 Else
                      {Check if all symptoms can be detected if all significant bit 0, bit 1,
                      bit 2 and bit 3 of CDN_DIAG_XX = 1 }
                      If      CDN_DIAG_XX = NC_NR_SYM_XX
                              and
                              ERR_SYM_XX = 0
                      Then
                              CTR_ABC_END_DIAG_XX = CTR_ABC_END_DIAG_XX
                                                              + C_ABC_INC_XX
                              If      CTR_ABC_END_DIAG_XX ≥ C_ABC_MAX_XX
                              Then
                                      CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX
                                      LV_END_DIAG_XX = 1
                              Endif
                      Else
                              {Check if a failure is detected}
                              If      ERR_SYM_XX ≠ 0
                              Then
                                      CTR_ABC_END_DIAG_XX = 0
                              Endif
                      Endif
                 Endif
        Endif(2b)
Endif(1)

Endif(0)
```

27.8.2 Description for ACTION_ERRM_MultiFilterSymEnd :

Syntax:	ACTION_ERRM_MultiFilterSymEnd(IN<XX>, IN<C_ABC_MAX_XX>, OUT<LV_ERR_XX>)	
Parameter(in):	XX C_ABC_MAX_XX	Type: index to identify the Diagnostic Type: Calibration byte for the maximum value
Parameter(out):	LV_ERR_XX	Type: Boolean
Short Description:	This action erases diagnostic failure and sets immediately the end of diagnostic information in case of multi-conditions filter (MPL_STD_INI)	

Formula section :

LV_CDN_DIAG_XX = 1
ERR_SYM_XX = NO_SYM
LV_ERR_XX = 0
CTR_ABC_XX = 0
LV_END_DIAG_XX = 1
CTR_ABC_END_DIAG_XX = C_ABC_MAX_XX

27.8.3 Description for ACTION_ERRM_MultiFilterReset:

Syntax:	ACTION_ERRM_MultiFilterReset(IN<XX>, OUT<LV_ERR_XX>)	
Parameter(in):	XX	Type: index to identify the Diagnostic
Parameter(out):	LV_ERR_XX	Type: Boolean
Short Description:	This action resets data filter in case of Multiconditions filter type	

Formula section :

LV_CDN_DIAG_XX = 0
ERR_SYM_XX = NO_SYM
LV_ERR_XX = 0
CTR_ABC_XX = 0
LV_END_DIAG_XX is unchanged
CTR_ABC_END_DIAG_XX is unchanged

27.8.4 Synchronization between multi-condition algorithm & Error Management

General information:

This section describes the mechanism of synchronisation of the failure filtering with the Error Management core. The principle is to call the Error Management upon events.

LV_END_DIAG_XX 0 → 1 transition
LV_ERR_XX 0 → 1 transition
LV_ERR_XX 1 → 0 transition

27.8.5 Mechanism for CPU load optimisation

General information:

This section describes the mechanism of optimisation. The principle is to suspend the filter algorithm execution when is not necessary. When the bench mode is used for a diagnostic XX the mechanism of CPU load optimisation is disabled for this instance.

Formula section:

```
If          LV_ERR_XX = 0
            and LV_END_DIAG_XX = 1
            and  ERR_SYM_XX = 0
            and CTR_ABC_XX = 0
Then
        It is not necessary to call error management (including filtering)
        LV_END_DIAG_XX = 1    {unchanged}
        LV_ERR_XX = 0         {unchanged}
Endif
```

27.8.6 Configuration data

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_SYM_XX	1	1...FH	1...15	1	[-]
Available symptom(s) for XX diagnostic					

Configuration data detailed description:

Symptoms 0 and 1 available:	03H
Symptoms 0, 1 and 2 available:	07H (typical value)
Symptoms 0, 1, 2 and 3 available:	0FH

Predefined filter type:

	NLC_ABC_STC_XX	NLC_ABC_CAL_DEC_XX	NLC_ABC_NOT_DEC_XX	NLC_ABC_INI_VALUE_XX	NLC_ABC_INI_IGK_XX	NLC_ABC_NOT_SAVE_XX
MPL_STD_INI	0	0	0	0	1	1

The **standard configuration with initialisation (MPL_STD_INI)** is NC_ABC_CONF_FCT_DIAG_XX = 3H (default configuration)

The data CTR_ABC_XX and LV_ERR_XX are Reset at the transition LV_IGK 0 → 1 and Reset (NLC_ABC_NOT_SAVE_XX = 1, NLC_ABC_INI_IGK_XX = 1) and the anti-bounce counter is initialised to 0 value (NLC_ABC_INI_VALUE_XX = 0).

The anti-bounce counter may be decremented (NLC_ABC_NOT_DEC_XX = 0).
The decrement value is equal to 1 (NLC_ABC_CAL_DEC_XX = 0).
The filter used is the Anti-bounce (NLC_ABC_STC_XX = 0).

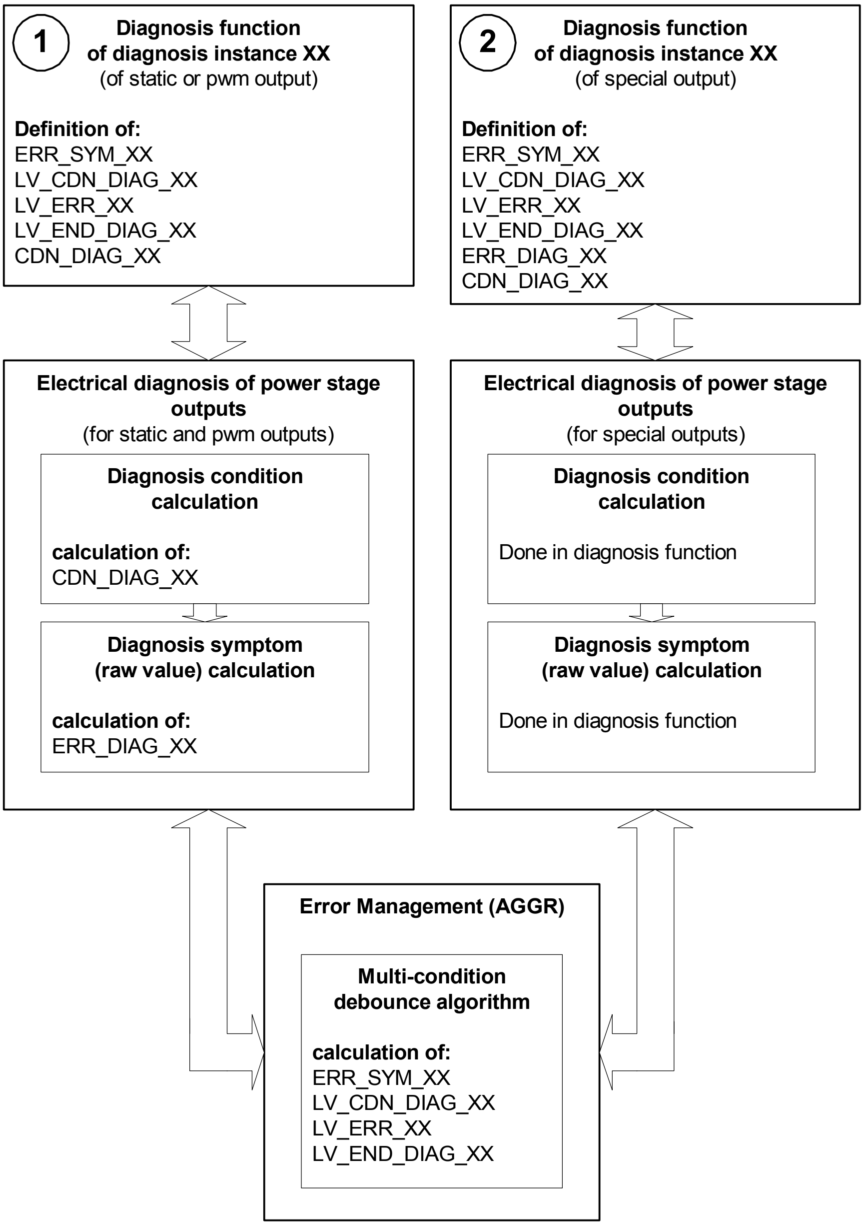
27.9 Electrical diagnosis of power stage outputs

FUNCTION DESCRIPTION:

General information:

In this function description the electrical diagnosis of power stage outputs is specified. Because of the possibility of different hardware configurations, the specification is splited into three parts plus one common part for the symptom debounce and error management treatment.

- Diagnosis of static outputs (e.g. relays)
- Diagnosis of pwm outputs (e.g. VVT actuator)
- Diagnosis of special outputs (other hardware configurations)



According these three different kind of driver outputs the management of the electrical diagnosis is different between static/pwm outputs and special outputs (see previous picture case 1 and 2).

For all types of these diagnoses three main calculations must be done:

- Diagnosis condition calculation
- Diagnosis symptom (raw value) calculation
- anti-bounce mechanism with error management treatments

In the following chapter these treatments are described.

27.9.1 Diagnosis of static outputs

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
ERR_DIAG_XX	-	0 ... 7H	0 ... 7	1	-
Diagnosis symptom (raw value) for each symptom for XX					
CDN_DIAG_XX	-	0 ... 7H	0 ... 7	1	-
Diagnosis condition for each symptom for XX					

Input data:

LV_IGK	LV_INH_DIAG_XX		

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX>, IN< ERR_DIAG_XX >, OUT< LV_END_DIAG_XX >, OUT< LV_ERR_XX >, OUT< LV_CDN_DIAG_XX >, OUT< ERR_SYM_XX >)

General information:

The purpose is to diagnose electrical errors detected by the hardware for static outputs. The signal is controlled by the ATIC 39.

The driver can distinguish between three symptoms: ‘Short to battery’ SCB, ‘Short to ground’ SCG and ‘Open circuit’ OC.

Description:

By the diagnosis condition calculation the information given by the BSW (‘Diagnosis of symptom ... is valid’) that shows, that a diagnosis of a symptom was possible, since the last recurrence of the diagnosis function, is displayed for each symptom within the value CDN_DIAG_XX.

The information of the failure symptom is also delivered by the BSW (‘Failure symptom ... detected by HW’). According this information the diagnosis symptom (raw value) calculation is performed to set ERR_DIAG_XX.

The calculation of LV_ERR_XX, LV_CDN_DIAG_XX, ERR_SYM_XX and LV_END_DIAG_XX is done in the Multi-condition debounce algorithm (part of Error Management AGGR).

Application conditions:

Initialization: -

Recurrence: -

Activation: every ECU state

reactivation in the NC_PSD_DLY recurrence after LV_INH_DIAG_XX 1->0 and LV_IGK 0->1 and reset and FMY reset

Deactivation: LV_INH_DIAG_XX = 1

at deactivation: CDN_DIAG_XX = 0

Formula section:

Diagnosis condition calculation:

CDN_DIAG_XX = 0

Diagnosis condition calculation for symptom 'Short to battery' SCB:

```
If      'Diagnosis of symptom SCB is valid'          (BSW information)
Then    bit 0 of CDN_DIAG_XX  = 1
Endif
```

Diagnosis condition calculation for symptom 'Short to ground' SCG:

```
If      'Diagnosis of symptom SCG is valid'          (BSW information)
Then    bit 1 of CDN_DIAG_XX  = 1
Endif
```

Diagnosis condition calculation for symptom 'Open circuit' OC:

```
If      'Diagnosis of symptom OC is valid'           (BSW information)
Then    bit 2 of CDN_DIAG_XX  = 1
Endif
```

Diagnosis symptom (raw value) calculation:

Symptom calculation for 'Short to battery' SCB:

```
If      'Failure symptom SCB detected by HW'      (BSW information)
Then    bit 0 of ERR_DIAG_XX = 1
Else    bit 0 of ERR_DIAG_XX = 0
Endif
```

Symptom calculation for 'Short to ground' SCG:

```
If      'Failure symptom SCG detected by HW'      (BSW information)
Then    bit 1 of ERR_DIAG_XX = 1
Else    bit 1 of ERR_DIAG_XX = 0
Endif
```

Symptom calculation for 'Open circuit' OC:

```
If      'Failure symptom OC detected by HW'       (BSW information)
Then    bit 2 of ERR_DIAG_XX = 1
Else    bit 2 of ERR_DIAG_XX = 0
Endif
```

ACTION_ERRM_FilterMulticondition (*XX*, CDN_DIAG_XX, ERR_DIAG_XX, LV_END_DIAG_XX, LV_ERR_XX, LV_CDN_DIAG_XX, ERR_SYM_XX)

Remark: For failure debouncing and error management treatment the algorithm Multi-condition debounce algorithm (part of Error Management AGGR) is used with the parameters CDN_DIAG_XX and ERR_DIAG_XX.

27.9.2 Diagnosis of PWM outputs

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
ERR_DIAG_XX	-	0 ... 7H	0 ... 7	1	-
Diagnosis symptom (raw value) for each symptom for XX					
CDN_DIAG_XX	-	0 ... 7H	0 ... 7	1	-
Diagnosis condition for each symptom for XX					

Input data:

LV_IGK	LV_INH_DIAG_XX	XXPWM	

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX>, IN< ERR_DIAG_XX >,
OUT< LV_END_DIAG_XX >, OUT< LV_ERR_XX >, OUT< LV_CDN_DIAG_XX >, OUT< ERR_SYM_XX >)

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose electrical errors detected by the hardware for PWM outputs. The signal is controlled by ATIC39.
The driver can distinguish between three symptoms: ‘Short to battery’ SCB, ‘Short to ground’ SCG and ‘Open circuit’ OC.

Because of the different filter times for SCB, SCG and OC detection, the diagnosis window, where a symptom can be detected, is defined for each symptom. A diagnosis window is defined by a minimum and a maximum duty cycle of the PWM signal controlled by the output of the driver.

SCB diagnosis window:

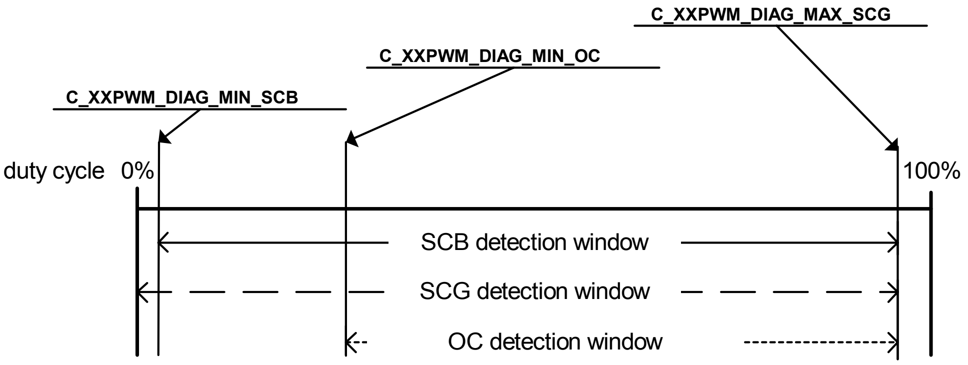
C_XXPWM_DIAG_MIN_SCB <-------> C_XXPWM_DIAG_MAX_SCG

SCG diagnosis window:

0 <-------> C_XXPWM_DIAG_MAX_SCG

OC diagnosis window:

C_XXPWM_DIAG_MIN_OC <-------> C_XXPWM_DIAG_MAX_SCG



For some diagnosis it's not necessary to define different diagnosis windows for each symptom. This is possible for all diagnosis, where a limitation of the duty cycle range is made within the function which controls the corresponding output. The limitation must be made in this way, that a diagnosis of all three symptoms can always (if LV_INH_DIAG_XX= 0) be made (CDN_DIAG_XX = 7H, LV_CDN_DIAG_XX = 1).

These two possibilities of a diagnosis condition calculation are described in the following chapter. It's distinguished between a "open range" (different diagnosis windows for each symptom necessary) and a "limited" duty cycle calculation.

27.9.2.1 Diagnosis condition calculation

27.9.2.1.1 "Open range" duty cycle

Description:

For an output with an "open range" duty cycle range, a diagnosis result of a symptom can be considered as valid, if the duty cycle of the PWM output was in the right window at least one time between two calculations of the diagnosis function and not only if diagnosis function is called. To have a better reflection of this behavior, the diagnosis condition is calculated independently of the diagnosis function.

If the recurrence of the 'diagnosis condition' calculation is the same like the 'symptom' calculation, the 'diagnosis condition' calculation must be treated after the 'symptom' calculation to have a new diagnosis condition CDN_DIAG_XX information of the following diagnosis window for the next symptom calculation.

Application conditions:

Initialization: -

Recurrence: -

Activation: every ECU state

reactivation in the NC_PSD_DLY recurrence after LV_INH_DIAG_XX 1->0 and LV_IGK 0->1 and reset and FMY reset

Deactivation: LV_INH_DIAG_XX = 1

at deactivation CDN_DIAG_XX = 0

Formula section:

Diagnosis window calculation:

```
If      0 <= XXPWM < C_XXPWM_DIAG_MIN_SCB
Then    bit 1 of CDN_DIAG_XX = 1
        Diagnosis of SCG is valid

Else    If      XXPWM < C_XXPWM_DIAG_MIN_OC
        Then    bit 0 and bit 1 of CDN_DIAG_XX = 1
                Diagnosis of SCB, SCG  is valid

        Else    If      XXPWM <= C_XXPWM_DIAG_MAX_SCG
                Then    bit 0, bit 1 and bit 2 of CDN_DIAG_XX = 1
                        Diagnosis of SCB, SCG  and OC is valid
                Endif
        Endif
Endif
```

27.9.2.1.2 "Limited" duty cycle

Description:

For an output with a "limited" duty cycle range, the diagnosis result of a symptom can always be considered as valid, because of the current duty cycle. It only can be invalid because special conditions managed by the BSW.

Application conditions:

Initialization: -

Recurrence: -

Activation: every ECU state

reactivation in the NC_PSD_DLY recurrence after LV_INH_DIAG_XX 1->0 and LV_IGK 0->1 and reset and FMY reset

Deactivation: LV_INH_DIAG_XX = 1

at deactivation: CDN_DIAG_XX = 0

Formula section:

bit 0, bit 1 and bit 2 of CDN_DIAG_XX = 1

Diagnosis of SCB, SCG and OC is valid

27.9.2.2 Diagnosis symptom (raw value) calculation

FUNCTION DESCRIPTION:

Application conditions:

Initialization: -

Recurrence: -

Activation: every ECU state

reactivation in the NC_PSD_DLY recurrence after LV_INH_DIAG_XX 1->0 and LV_IGK 0->1 and reset and FMY reset

Deactivation: LV_INH_DIAG_XX = 1

Formula section:

Symptom calculation for 'Short to battery' SCB:

```
If      'Failure symptom SCB detected by HW'      (BSW information)
Then    bit 0 of ERR_DIAG_XX = 1
Else    bit 0 of ERR_DIAG_XX = 0
Endif
```

Symptom calculation for 'Short to ground' SCG:

```
If      'Failure symptom SCG detected by HW'      (BSW information)
Then    bit 1 of ERR_DIAG_XX = 1
Else    bit 1 of ERR_DIAG_XX = 0
Endif
```

Symptom calculation for 'Open circuit' OC:

```
If      'Failure symptom OC detected by HW'       (BSW information)
Then    bit 2 of ERR_DIAG_XX = 1
Else    bit 2 of ERR_DIAG_XX = 0
Endif
```

Diagnosis condition from BSW verification:

Diagnosis condition calculation for symptom 'Short to battery' SCB:

```
If      'Diagnosis of symptom SCB is not valid'   (BSW information)
Then    bit 0 of CDN_DIAG_XX  = 0
Endif
```

Diagnosis condition calculation for symptom 'Short to ground' SCG:

```
If      'Diagnosis of symptom SCG is not valid'      (BSW information)
Then    bit 0 of CDN_DIAG_XX  = 0
Endif
```

Diagnosis condition calculation for symptom 'Open circuit' OC:

```
If      'Diagnosis of symptom OC is not valid'       (BSW information)
Then    bit 0 of CDN_DIAG_XX  = 0
Endif
```

ACTION_ERRM_FilterMulticondition (*XX*, CDN_DIAG_XX, ERR_DIAG_XX, LV_END_DIAG_XX, LV_ERR_XX, LV_CDN_DIAG_XX, ERR_SYM_XX)

Remark: For failure debouncing and error management treatment the algorithm Multi-condition debounce algorithm (part of Error Management AGGR) is used with the parameters CDN_DIAG_XX and ERR_DIAG_XX.

CDN_DIAG_XX = 0 (after failure debouncing and error management treatment)

27.9.3 Diagnosis of special outputs

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
ERR_DIAG_XX	-	0 ... 7H	0 ... 7	1	-
Diagnosis symptom (raw value) for each symptom for XX					
CDN_DIAG_XX	-	0 ... 7H	0 ... 7	1	-
Diagnosis condition for each symptom for XX					

Input data:

LV_IGK	ERR_DIAG_XX	CDN_DIAG_XX	

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX>, IN< ERR_DIAG_XX >, OUT< LV_END_DIAG_XX >, OUT< LV_ERR_XX >, OUT< LV_CDN_DIAG_XX >, OUT< ERR_SYM_XX >)

General information:

The purpose is to diagnose electrical errors detected by the hardware for special outputs. Special outputs are outputs, which cannot be treated like "normal" static or pwm outputs, like described in the previous chapters.

Description:

The diagnosis condition calculation for the driver output is done in the diagnosis function of the diagnosis instance XX. It's displayed for each symptom within the value CDN_DIAG_XX.

Also the diagnosis symptom (raw value) calculation (ERR_DIAG_XX) is performed in the diagnosis function for special outputs.

The calculation of LV_ERR_XX, LV_CDN_DIAG_XX, ERR_SYM_XX and LV_END_DIAG_XX is done in the Multi-condition debounce algorithm (part of Error Management AGGR).

Application conditions:

Initialization: -

Recurrence: -

Activation: every ECU state

reactivation in the NC_PSD_DLY recurrence after CDN_DIAG_XX > 0 and LV_IGK 0->1 and reset and FMY reset

Deactivation: -

Formula section:

Diagnosis condition / Diagnosis symptom (raw value) calculation:

CDN_DIAG_XX and ERR_DIAG_XX are calculated in the diagnosis function.

ACTION_ERRM_FilterMulticondition (*XX*, CDN_DIAG_XX, ERR_DIAG_XX, LV_END_DIAG_XX, LV_ERR_XX, LV_CDN_DIAG_XX, ERR_SYM_XX)

Remark: For failure debouncing and error management treatment the algorithm Multi-condition debounce algorithm (part of Error Management AGGR) is used with the parameters CDN_DIAG_XX and ERR_DIAG_XX.

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY	1	0 ... FFH	0 ... 255	1	-
The diagnosis is reactivated after (LV_INH_DIAG_XX 1->0 or CDN_DIAG_XX >0 (depending on case)) and LV_IGK 0->1 and reset and FMY reset after NC_PSD_DLY recurrences.					

27.10 Electrical diagnosis of powerstage outputs (Applic. Inc.)

General information:

This specification defines all output-data and calibration data for all ATIC39 powerstage diagnosis as well as specific functions like limp home which are not covered by the generic ATIC39 basic algorithm for static and PWM outputs

Remark:

For all static outputs a minimum switch-on time of 50ms for error symptom detection is necessary.

27.10.1 Diagnosis of static outputs

27.10.1.1 Crankcase ventilation heater relay diagnosis (LV_ERR_RLY_CRCV_HEAT)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_RLY_CRCV_HEAT	V/O	0...1H	0...1	1	[-]
Logical value for CRCV_HEAT error					
LV_CDN_DIAG_RLY_CRCV_HEAT	V/O	0...1H	0...1	1	[-]
Diagnosis condition CRCV_HEAT					
ERR_SYM_RLY_CRCV_HEAT	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CRCV_HEAT					
LV_END_DIAG_RLY_CRCV_HEAT	V/O	0...1H	0...1	1	[-]
End of CRCV_HEAT diagnosis					
LV_INH_DIAG_RLY_CRCV_HEAT	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition RLY_CRCV_HEAT diagnosis					
CDN_DIAG_RLY_CRCV_HEAT	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of RLY_CRCV_HEAT bit 0: diagnosis condition for symptom SCP (SYM_0)bit 1: diagnosis condition for symptom SCG (SYM_1)bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_RLY_CRCV_HEAT	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for RLY_CRCV_HEAT (only parameter)					

Input data:

LV_IGK	LV_CDN_VB_MIN_DIAG	LV_ERR_SPI_MPS	

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the signal from the driver which controls the crankcase ventilation heater relay.

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Vbatt        (= SYM_0)
 │ │ └───> Short circuit to GND          (= SYM_1)
 │ └─────> Open load                     (=SYM_2)
 └───────> -                             (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 1000ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_RLY_CRCV_HEAT = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_RLY_CRCV_HEAT. Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_RLY_CRCV_HEAT.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_RLY_CRCV_HEAT and ERR_DIAG_RLY_CRCV_HEAT.

This algorithm determines:

ERR_SYM_RLY_CRCV_HEAT	(Detected error symptom)
LV_ERR_RLY_CRCV_HEAT	(Error flag for debounced error)
LV_CDN_DIAG_RLY_CRCV_HEAT	(Diagnosis condition information)
LV_END_DIAG_RLY_CRCV_HEAT	(End of diagnosis information)

Diagnosis inhibition:

If LV_IGK = 1 **and**
LV_CDN_VB_MIN_DIAG = 1 **and**
LV_ERR_SPI_MPS = 0

Then LV_INH_DIAG_RLY_CRCV_HEAT = 0
Else LV_INH_DIAG_RLY_CRCV_HEAT = 1

Endif

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_RLY_CRCV_HEAT	1	0...FFH	0...255	1	[-]
Debounce counter increment CRCV_HEAT					
C_ABC_MAX_RLY_CRCV_HEAT	1	1...FFH	1...255	1	[-]
Debounce counter maximum value CRCV_HEAT					

27.10.1.2 Air condition compressor relay diagnosis (LV_ERR_RLY_ACCOUT)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_RLY_ACCOUT	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on RLY_ACCOUT command signal.					
LV_CDN_DIAG_RLY_ACCOUT	V/O	0...1H	0...1	1	[-]
Diagnosis condition RLY_ACCOUT diagnosis					
ERR_SYM_RLY_ACCOUT	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom RLY_ACCOUT diagnosis					
LV_END_DIAG_RLY_ACCOUT	V/O	0...1H	0...1	1	[-]
End of diagnosis RLY_ACCOUT diagnosis					
LV_INH_DIAG_RLY_ACCOUT	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition RLY_ACCOUT diagnosis					
CDN_DIAG_RLY_ACCOUT	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of RLY_ACCOUT bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_RLY_ACCOUT	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for RLY_ACCOUT (only parameter)					

Input data:

LV_IGK	LV_CDN_VB_MIN_DIAG	LV_VAR_ACIN	LV_ERR_SPI_MPS
LV_VAR_RLY_ACCOUT			

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the signal from the driver which controls the air condition relay signal.

The diagnosis is only active if “Vehicle with ACC” is learnt.

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> Short circuit to Vbatt              (= SYM_0)
 │  │  └────> Short circuit to GND                (= SYM_1)
 │  └───────> Open load                           (=SYM_2)
 └──────────> -                                   (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 1s

Activation: LV_VAR_RLY_ACCOUT = 1 **AND**

LV_VAR_ACIN = 1

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_RLY_ACCOUT = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_RLY_ACCOUT .
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_RLY_ACCOUT.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG RLY_ACCOUT and ERR_DIAG_RLY_ACCOUT.

This algorithm determines:

ERR_SYM_RLY_ACCOUT	(Detected error symptom)
LV_ERR_RLY_ACCOUT	(Error flag for debounced error)
LV_CDN_DIAG_RLY_ACCOUT	(Diagnosis condition information)
LV_END_DIAG_RLY_ACCOUT	(End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_IGK = 1                    AND
        LV_CDN_VB_MIN_DIAG = 1        AND
        LV_ERR_SPI_MPS = 0
THEN    LV_INH_DIAG_RLY_ACCOUT = 0
ELSE    LV_INH_DIAG_RLY_ACCOUT = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_RLY_ACCOUT	1	0...FFH	0...255	1	[-]
Debounce counter increment RLY_ACCOUT diagnosis					
C_ABC_MAX_RLY_ACCOUT	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - RLY_ACCOUT diagnosis					

27.10.1.3 DMTL change over valve diagnosis (LV_ERR_DMTLS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_DMTLS	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on DMTLS command signal.					
LV_CDN_DIAG_DMTLS	V/O	0...1H	0...1	1	[-]
Diagnosis condition DMTLS diagnosis					
ERR_SYM_DMTLS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom DMTLS diagnosis					
LV_END_DIAG_DMTLS	V/O	0...1H	0...1	1	[-]
End of diagnosis DMTLS diagnosis					
LV_INH_DIAG_DMTLS	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition DMTLS diagnosis					
CDN_DIAG_RLY_DMTLS	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of DMTLS bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_RLY_DMTLS	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for DMTLS (only parameter)					

Input data:

LV_IGK	LV_CDN_VB_OBD1	LV_CONF_DMTL	LV_ERR_SPI_MPS
ECU_STATE			

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the signal from the driver which controls the DMTL valve.

The diagnosis is only activated if DMTL is configurated.

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> Short circuit to Vbatt                  (= SYM_0)
 │  │  └────> Short circuit to GND                    (= SYM_1)
 │  └───────> Open load                               (=SYM_2)
 └──────────> -                                       (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 100ms

Activation: LV_CONF_DMTL = 1

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_DMTLS = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_DMTLS.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_DMTLS.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_DMTLS and ERR_DIAG_DMTLS.

This algorithm determines:

ERR_SYM_DMTLS	(detected error symptom)
LV_ERR_DMTLS	(Error flag for debounced error)
LV_CDN_DIAG_DMTLS	(Diagnosis condition information)
LV_END_DIAG_DMTLS	(End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_CDN_VB_OBD1 = 1      AND
        LV_ERR_SPI_MPS = 0      AND
        ECU_STATE =! WAKE_UP
THEN    LV_INH_DIAG_DMTLS = 0
ELSE    LV_INH_DIAG_DMTLS = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_DMTLS	1	0...FFH	0...255	1	[-]
Debounce counter increment DMTLS diagnosis					
C_ABC_MAX_DMTLS	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - DMTLS diagnosis					

27.10.1.4 DMTL pump diagnosis (LV_ERR_DMTL_PUMP)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_DMTL_PUMP	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on DMTL_PUMP command signal.					
LV_CDN_DIAG_DMTL_PUMP	V/O	0...1H	0...1	1	[-]
Diagnosis condition DMTL_PUMP diagnosis					
ERR_SYM_DMTL_PUMP	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom DMTL_PUMP diagnosis					
LV_END_DIAG_DMTL_PUMP	V/O	0...1H	0...1	1	[-]
End of diagnosis DMTL_PUMP diagnosis					
LV_INH_DIAG_DMTL_PUMP	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition DMTLH_PUMP diagnosis					
T_V_DMTL_MAX_DIAG	V	0...FFH	0...2.55	0.01	[s]
SCB detection timer					
CDN_DIAG_RLY_DMTL_PUMP	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of DMTL_PUMP bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_RLY_DMTL_PUMP	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for DMTL_PUMP (only parameter)					

Input data:

LV_IGK	LV_CDN_VB_OBD1	LV_CONF_DMTL	LV_ERR_SPI_MPS
V_DMTL	LV_DMTL_PUMP	ECU_STATE	

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the signal from the driver which controls the DMTL pump.

The diagnosis is only activated if DMTL is configurated.

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.

Remark:

Due to hardware conditions and overtemperature protection for shunt resistor (max. time for SCB is 50ms) the SCB detection is only possible with monitoring the voltage V_DMTL.

Error-symptoms are defined to this diagnosis function as following :

```
|_|_|_|_|
 | | | └─> Short circuit to Vbatt        (= SYM_0)
 | | └───> Short circuit to GND          (= SYM_1)
 | └─────> Open load                     (=SYM_2)
 └───────> -                             (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 100ms for SCG/OL detection

10ms for SCB detection

Activation: LV_CONF_DMTL = 1

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_DMTL_PUMP = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_DMTL_PUMP.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_DMTL_PUMP.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_DMTL_PUMP and ERR_DIAG_DMTL_PUMP.

This algorithm determines:

ERR_SYM_DMTL_PUMP	(Detected error symptom)
LV_ERR_DMTL_PUMP	(Error flag for debounced error)
LV_CDN_DIAG_DMTL_PUMP	(Diagnosis condition information)
LV_END_DIAG_DMTL_PUMP	(End of diagnosis information)

Diagnosis inhibition:

IF LV_CDN_VB_OBD1 = 0 **OR**

LV_ERR_SPI_MPS = 1 **OR**

```
              ECU_STATE = WAKE_UP   OR
              ERR_SYM_DMTL_PUMP = SYM_0
THEN          LV_INH_DIAG_DMTL_PUMP = 1
ELSE          LV_INH_DIAG_DMTL_PUMP = 0
ENDIF
```

Detection of SCB:

```
IF(1)         LV_DMTL_PUMP = 1          //Pump is switched on
THEN(1)IF(2)      V_DMTL > C_V_DMTL_MAX_DIAG
    THEN(2)IF(3)         T_V_DMTL_MAX_DIAG >= C_T_V_DMTL_MAX_DIAG
        THEN(3)          ERR_SYM_DMTL_PUMP = SYM_0
                         LV_ERR_DMTL_PUMP = 1    ( set with
                         CTR_ABC_DMTL_PUMP = C_ABC_MAX_DMTL_PUMP )
                         LV_ERR_DMTL_PUMP = 1
                         LV_END_DIAG_DMTL_DUMP = 1
                         DMTL_PUMP pin is switched off by hardware immediately,
                         diagnosis is inhibited until next initialization
        ELSE(3)          T_V_DMTL_MAX_DIAG ++
        ENDIF(3)
    ELSE(2)       T_V_DMTL_MAX_DIAG - -
    ENDIF(2)
ELSE(1)       normal diagnosis algorithm OL/SCG
ENDIF(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_DMTL_PUMP	1	0...FFH	0...255	1	[-]
Debounce counter increment DMTL_PUMP diagnosis					
C_ABC_MAX_DMTL_PUMP	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - DMTL_PUMP diagnosis					
C_V_DMTL_MAX_DIAG	1	0...3FFH	0...4.99511	4.88E-03	[V]
Threshold for SCB detection					
C_T_V_DMTL_MAX_DIAG	1	0...FFH	0...2.55	0.01	[s]
Time threshold for SCB detection					

27.10.1.5 DMTL heater diagnosis (LV_ERR_DMTLH)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_DMTLH	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on DMTLH command signal.					
LV_CDN_DIAG_DMTLH	V/O	0...1H	0...1	1	[-]
Diagnosis condition DMTLH diagnosis					
ERR_SYM_DMTLH	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom DMTLH diagnosis					
LV_END_DIAG_DMTLH	V/O	0...1H	0...1	1	[-]
End of diagnosis DMTLH diagnosis					
LV_INH_DIAG_DMTLH	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition DMTLH diagnosis					
CDN_DIAG_DMTLH	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of DMTLH bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_DMTLH	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for DMTLH (only parameter)					

Input data:

LV_CONF_DMTL	LV_CDN_VB_OBD1	LV_ERR_SPI_MPS	ECU_STATE

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the signal from the driver which controls the DMTL heater.

The diagnosis is only activated if DMTL is configurated.

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Vbatt          (= SYM_0)
 │ │ └───> Short circuit to GND            (= SYM_1)
 │ └─────> Open load                       (=SYM_2)
 └───────> -                               (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 1s

Activation: LV_CONF_DMTL = 1

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_DMTLH = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_DMTLH.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_DMTLH.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_DMTLH and ERR_DIAG_DMTLH.

This algorithm determines:

ERR_SYM_DMTLH	(detected error symptom)
LV_ERR_DMTLH	(Error flag for debounced error)
LV_CDN_DIAG_DMTLH	(Diagnosis condition information)
LV_END_DIAG_DMTLH	(End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_CDN_VB_OBD1 = 1     AND
        LV_ERR_SPI_MPS = 0     AND
        ECU_STATE =! WAKE_UP
THEN    LV_INH_DIAG_DMTLH = 0
ELSE    LV_INH_DIAG_DMTLH = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_DMTLH	1	0...FFH	0...255	1	[-]
Debounce counter increment DMTLH diagnosis					
C_ABC_MAX_DMTLH	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - DMTLH diagnosis					

27.10.1.6 E-box cooling fan diagnosis (LV_ERR_EBOX_CFA)

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>LV_ERR_EBOX_CFA</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Boolean for error currently present on EBOX_CFA command signal.</td></tr>
<tr><td>LV_CDN_DIAG_EBOX_CFA</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Diagnosis condition EBOX_CFA diagnosis</td></tr>
<tr><td>ERR_SYM_EBOX_CFA</td><td>V/O</td><td>0H
1H
2H
4H
8H</td><td>NO_SYM
SYM_0
SYM_1
SYM_2
SYM_3</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Error symptom EBOX_CFA diagnosis</td></tr>
<tr><td>LV_END_DIAG_EBOX_CFA</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">End of diagnosis EBOX_CFA diagnosis</td></tr>
<tr><td>LV_INH_DIAG_EBOX_CFA</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Diagnosis inhibition EBOX_CFA diagnosis</td></tr>
<tr><td>CDN_DIAG_EBOX_CFA</td><td>V</td><td>0...7H</td><td>0...7</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Diagnosis condition for each symptom of EBOX_CFA
bit 0: diagnosis condition for symptom SCP (SYM_0)
bit 1: diagnosis condition for symptom SCG (SYM_1)
bit 2: diagnosis condition for symptom OC (SYM_2)</td></tr>
<tr><td>ERR_DIAG_EBOX_CFA</td><td>-</td><td>0H
1H
2H
4H
8H</td><td>NO_SYM
SYM_0
SYM_1
SYM_2
SYM_3</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Raw value of error symptom for EBOX_CFA (only parameter)</td></tr>
</table>

Input data:

LV_IGK	LV_CDN_VB_MIN_DIAG	LV_VAR_EBOX_CFA	LV_ERR_SPI_MPS

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the signal from the driver which controls the EBOX cooling fan.

The input signal is modulated control pulse (PWM).

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Vbatt        (= SYM_0)
 │ │ └───> Short circuit to GND          (= SYM_1)
 │ └─────> Open load                     (=SYM_2)
 └───────> -                             (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 1s

Activation: LV_VAR_EBOX_CFA = 1

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_EBOX_CFA = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_EBOX_CFA.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_EBOX_CFA.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_EBOX_CFA and ERR_DIAG_EBOX_CFA.

This algorithm determines:

ERR_SYM_EBOX_CFA	(detected error symptom)
LV_ERR_EBOX_CFA	(Error flag for debounced error)
LV_CDN_DIAG_EBOX_CFA	(Diagnosis condition information)
LV_END_DIAG_EBOX_CFA	(End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_IGK = 1                  AND
        LV_CDN_VB_MIN_DIAG = 1      AND
        LV_ERR_SPI_MPS = 0
THEN    LV_INH_DIAG_EBOX_CFA = 0
ELSE    LV_INH_DIAG_EBOX_CFA = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_EBOX_CFA	1	0...FFH	0...255	1	[-]
Debounce counter increment EBOX_CFA diagnosis					
C_ABC_MAX_EBOX_CFA	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - EBOX_CFA diagnosis					

27.10.1.7 Exhaust flap diagnosis (LV_ERR_EF)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_EF	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on EF command signal.					
LV_CDN_DIAG_EF	V/O	0...1H	0...1	1	[-]
Diagnosis condition EF diagnosis					
ERR_SYM_EF	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom EFdiagnosis					
LV_END_DIAG_EF	V/O	0...1H	0...1	1	[-]
End of diagnosis EF diagnosis					
LV_INH_DIAG_EF	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition EF diagnosis					
CDN_DIAG_EF	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of EF bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_EF	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for EF (only parameter)					

Input data:

LV_IGK	LV_CDN_VB_MIN_DIAG	LV_VAR_EF	LV_ERR_SPI_MPS

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the exhaust flag signal from the driver which controls the exhaust flap.

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> Short circuit to Vbatt          (= SYM_0)
 │  │  └────> Short circuit to GND            (= SYM_1)
 │  └───────> Open load                       (=SYM_2)
 └──────────> -                               (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 1s

Activation: LV_VAR_EF = 1

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_EF = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_EF.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_EF.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_EF and ERR_DIAG_EF.

This algorithm determines:

ERR_SYM_EF	(Detected error symptom)
LV_ERR_EF	(Error flag for debounced error)
LV_CDN_DIAG_EF	(Diagnosis condition information)
LV_END_DIAG_EF	(End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_IGK = 1                      AND
        LV_CDN_VB_MIN_DIAG = 1          AND
        LV_ERR_SPI_MPS = 0
THEN    LV_INH_DIAG_EF = 0
ELSE    LV_INH_DIAG_EF = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_EF	1	0...FFH	0...255	1	[-]
Debounce counter increment - EF diagnosis					
C_ABC_MAX_EF	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - EF diagnosis					

27.10.1.8 Starter relay output diagnosis (LV_ERR_RLY_ST)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_RLY_ST	V/O	0...1H	0...1	1	[-]
Status of starter output diagnosis.					
LV_CDN_DIAG_RLY_ST	V/O	0...1H	0...1	1	[-]
Diagnosis condition RLY_ST diagnosis					
ERR_SYM_RLY_ST	V/O	0H	NO_SYM	1	[-]
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Detected error symptom RLY_ST diagnosis					
LV_END_DIAG_RLY_ST	V/O	0...1H	0...1	1	[-]
End of diagnosis RLY_ST diagnosis					
LV_INH_DIAG_RLY_ST	V/O	0...1H	0...1	1	[-]
Inhibition of diagnosis RLY_ST diagnosis					
CDN_DIAG_RLY_ST	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of RLY_ST bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_RLY_ST	-	0H	NO_SYM	1	[-]
		1H	SYM_0		
		2H	SYM_1		
		4H	SYM_2		
		8H	SYM_3		
Raw value of error symptom for RLY_ST (only parameter)					

Input data:

LV_IGK	LV_CDN_VB_OBD1	LV_VAR_RLY_ST	LV_ST_END
LV_ERR_SPI_MPS			

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the starter signal from the driver which switches the output signal for the CAS - ECU. This diagnosis detects electrical faults as defined by OBD I requirements. The diagnosis is only performed if the variant “BN2000” is enabled.

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”

Error-symptoms are defined to this diagnosis function as following :

```
□□□□
 | | | └-> Short circuit to Vbatt        (= SYM_0)
 | | └---> Short circuit to GND          (= SYM_1)
 | └-----> Open load                     (= SYM_2)
 └-------> -                             (= SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 50ms in case of LV_ST_END = 0

1000ms in case of LV_ST_END = 1

Activation: LV_VAR_RLY_ST = 1

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_RLY_ST = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_RLY_ST.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_RLY_ST.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_RLY_ST and ERR_DIAG_RLY_ST.

This algorithm determines:

ERR_SYM_RLY_ST	(detected error symptom)
LV_ERR_RLY_ST	(Error flag for debounced error)
LV_CDN_DIAG_RLY_ST	(Diagnosis condition information)
LV_END_DIAG_RLY_ST	(End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_IGK = 1              AND
        LV_CDN_VB_OBD1 = 1      AND
        LV_ERR_SPI_MPS = 0
  THEN  LV_INH_DIAG_RLY_ST = 0
  ELSE  LV_INH_DIAG_RLY_ST = 1
  ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_RLY_ST	1	0...FFH	0...255	1	[-]
Anti-bounce increment value					
C_ABC_MAX_RLY_ST	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value					

27.10.2 Diagnosis of PWM outputs

27.10.2.1 IVVT solenoid valve diagnosis (LV_ERR_SLV_IVVT_IN / EX)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SLV_IVVT_IN	V/O	0...1H	0...1	1	[-]
Error flag solenoid valve of inlet camshaft i					
LV_ERR_SLV_IVVT_EX	V/O	0...1H	0...1	1	[-]
Error flag solenoid valve of exhaust camshaft i					
LV_CDN_DIAG_SLV_IVVT_IN	V/O	0...1H	0...1	1	[-]
Diagnosis condition SLV_IVVT_IN_i					
ERR_SYM_SLV_IVVT_IN	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom SLV_IVVT_IN_i					
LV_END_DIAG_SLV_IVVT_IN	V/O	0...1H	0...1	1	[-]
End of SLV_IVVT_IN_i diagnosis					
LV_CDN_DIAG_SLV_IVVT_EX	V/O	0...1H	0...1	1	[-]
Diagnosis condition SLV_IVVT_EX_i					
ERR_SYM_SLV_IVVT_EX	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom SLV_IVVT_EX_i					
LV_END_DIAG_SLV_IVVT_EX	V/O	0...1H	0...1	1	[-]
End of SLV_IVVT_EX_i diagnosis					
LV_INH_DIAG_SLV_IVVT_IN	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition IVVT_IN					
LV_INH_DIAG_SLV_IVVT_EX	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition IVVT_EX					
CDN_DIAG_SLV_IVVT_IN	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of SLV_IVVT_IN bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
CDN_DIAG_SLV_IVVT_EX	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of SLV_IVVT_EX bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_SLV_IVVT_IN	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for SLV_IVVT_IN (only parameter)					
ERR_DIAG_SLV_IVVT_EX	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for SLV_IVVT_EX (only parameter)					

Input data:

LV_CDN_VB_OBD1	LV_IGK	IVVTPWM_0	IVVTPWM_1

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis is to detect electrical faults in the IVVT unit.

The input signal is a modulated control pulse (PWM).

Description:

For error detection algorithm see “Electrical diagnosis of powerstage outputs ATIC39”.
The diagnosis condition calculation is done according **“Open range”** duty cycle, thus a definition of the PWM diagnosis windows is neccessary (see calibration data).

Error-symptoms are defined to this diagnosis function as following :

```
☐☐☐☐
 | | | └─> Short circuit to VBatt          (= SYM_0)
 | | └───> Short circuit to GND            (= SYM_1)
 | └─────> Open load                       (= SYM_2)
 └───────> -                               (= SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**
(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Recurrence: 100ms

Activation: at every engine operating state

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_SLV_IVVT_IN/EX = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_SLV_IVVT_IN/EX.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_SLV_IVVT_IN/EX .

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_SLV_IVVT_IN/EX and ERR_DIAG_SLV_IVVT_IN/EX.

This algorithm determines:

ERR_SYM_SLV_IVVT_IN/EX	(detected error symptom)
LV_ERR_SLV_IVVT_IN/EX	(Error flag for debounced error)
LV_CDN_DIAG_SLV_IVVT_IN/EX	(Diagnosis condition information)
LV_END_DIAG_SLV_IVVT_IN/EX	(End of diagnosis information)

Diagnosis inhibition:

IF LV_IGK = 1 **AND**

```
        LV_CDN_VB_OBD1 = 1          AND
        LV_ERR_SPI_MPS = 0
THEN    LV_INH_DIAG_SLV_IVVT_IN/EX = 0
ELSE    LV_INH_DIAG_SLV_IVVT_IN/EX = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SLV_IVVT	1	0...FFH	0...255	1	[-]
Debounce counter increment - VANOS diagnosis magnetic valve					
C_ABC_MAX_SLV_IVVT	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - VANOS diagnosis magnetic valves					
C_SLV_IVVT_PWM_DIAG_MIN_SCB	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Minimum threshold for SCB diagnosis window					
C_SLV_IVVT_PWM_DIAG_MAX_SCG	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Maximum threshold for SCG diagnosis window					
C_SLV_IVVT_PWM_DIAG_MIN_OC	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Minimum threshold for OC diagnosis window					

27.11 Dynamic Error Management

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
CTR_ERR_DYN_NR	V/O/S	0...FFH	0...255	1	[-]
Number of failures stored in dynamic memory (2nd layer)					
LV_ERR_MEM_XX	V/O/S	0...1H	0...1	1	[-]
Failure is stored in dynamic memory (2nd layer)					
CTR_FRC[NC_NR_ERR_DYN]	V/O/S	0...FFH	0...255	1	[-]
Frequency counter : number of occurrences of present failure					
LV_ERR_TMP[NC_NR_ERR_DYN]	V/O/S	0...1H	0...1	1	[-]
Temporary failure status flag					
LV_ERR_PND[NC_NR_ERR_DYN]	V/O/S	0...1H	0...1	1	[-]
Pending failure status flag					
LV_ERR_MKD[NC_NR_ERR_DYN]	V/O/S	0...1H	0...1	1	[-]
Marked failure status flag					
DIAG_INST[NC_NR_ERR_DYN]	V	0...FFFFH	0...65535	1	[-]
Index of diagnosis instance XX with XX = F(IDX)					
LV_ERR_DC[NC_NR_ERR_DYN]	V/O	0...1H	0...1	1	[-]
Failure present status for this driving cycle					
LV_DC_MAX[NC_NR_ERR_DYN]	V/O/S	0...1H	0...1	1	[-]
Flag for the MIL state (on or off) related to failure IDX					
CTR_DC[NC_NR_ERR_DYN]	V/O/S	0...FFH	0...255	1	[-]
Counter for driving cycle					
LV_ERR_DISA[NC_NR_ERR_DYN]	V/O/S	0...1H	0...1	1	[-]
Disappeared failure status flag					
LV_ERR_CFM[NC_NR_ERR_DYN]	V/O/S	0...1H	0...1	1	[-]
Confirmed failure status flag					
CTR_WUP_CYC[NC_NR_ERR_DYN]	V/O/S	0...FFH	0...255	1	[-]
Warm-up cycles counter					

Input data:

DC_DEC_XX	DC_INC_XX	DC_MAX_XX	LV_DC
LV_END_DIAG_XX	LV_ERR_XX	LV_MKD_MOD	LV_WUP_CYC
LC_ENA_SCDN	NLC_ENA_SCDN		

Export actions:

ACTION_ERRM_EraseErr (IN <IDX>)
This action erases the failure IDX in the second layer memory
ACTION_ERRM_IncrementDCctrScdn (IN <XX>)
This action is used to increment the driving cycle counter for failures using similar conditions functionality
ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT <PendingStatus>)
This action shall be called to determine if a failure is pending or not

Import actions:

ACTION_ERRM_StoreDtc (IN <IDX>)
ACTION_ERRM_StoreDtcLst (IN <IDX>)
ACTION_ERRM_EraseDtc (IN <IDX>)
ACTION_ERRM_StoreFrf (IN <IDX>)
ACTION_ERRM_EraseFrf (IN <IDX>)
ACTION_ERRM_StoreHistory (IN <IDX>)
ACTION_ERRM_DecrementDCctrScdn (OUT <DCdec>, IN <XX>)
ACTION_ERRM_ConfirmErrScdn (OUT <DCconf>, IN <XX>)
ACTION_ERRM_EraseScdn (IN <XX>)
ACTION_ERRM_PrioRule (OUT <RESP> , OUT <IDX>)
ACTION_ERRM_TrigErrDyn (IN <XX>)

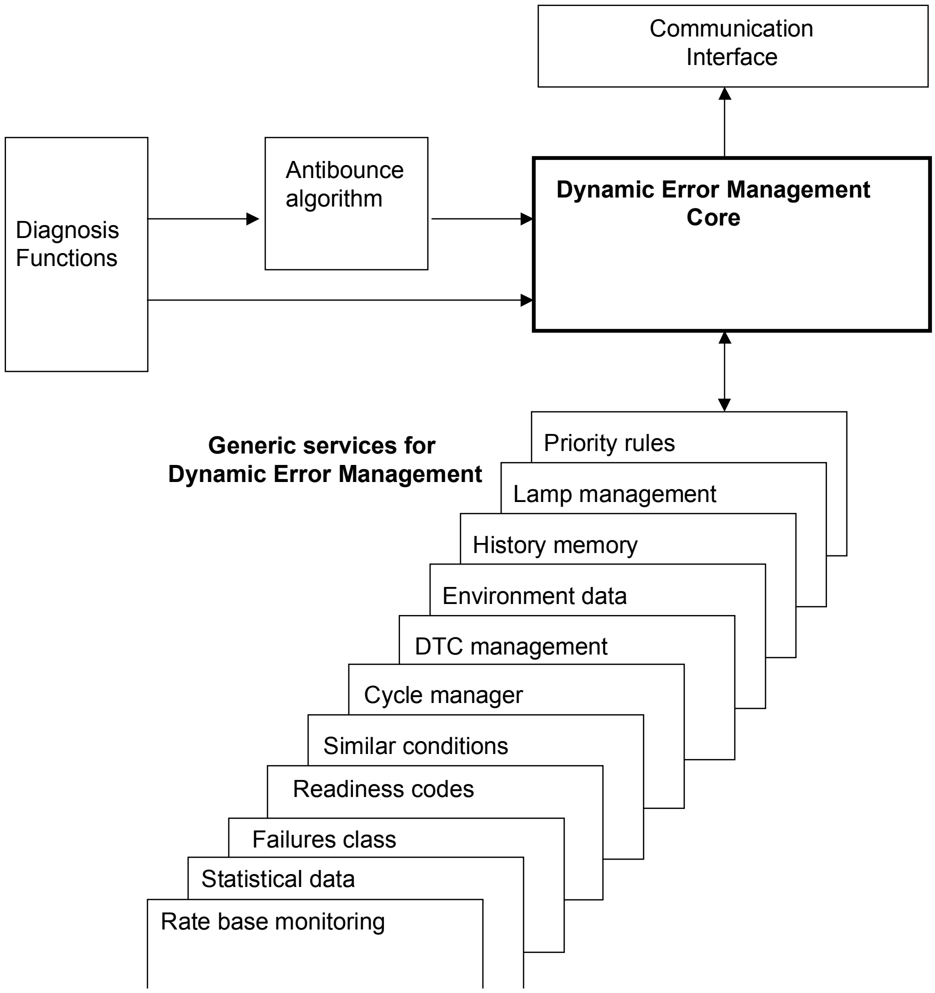
FUNCTION DESCRIPTION:

General information:

This module is the heart of the dynamic error management function. It defines statement of principles and mechanism of dynamic error management function.

Dynamic error management receives results from diagnosis function (may be filtered by antibounce algorithm). These results are, in this module, managed as diagnosis failure following **CARB** and **EOBD** standards.

In addition, this module uses generic services defined by others modules (priority rules for example).


Communication Interface
Diagnosis Functions
Antibounce algorithm
Dynamic Error Management Core
Generic services for Dynamic Error Management
Priority rules
Lamp management
History memory
Environment data
DTC management
Cycle manager
Similar conditions
Readiness codes
Failures class
Statistical data
Rate base monitoring

27.11.1 Diagnosis failure description

27.11.1.1 Failures definition

Description:

Dynamic error management manages failures, which have different states.

Detected failure

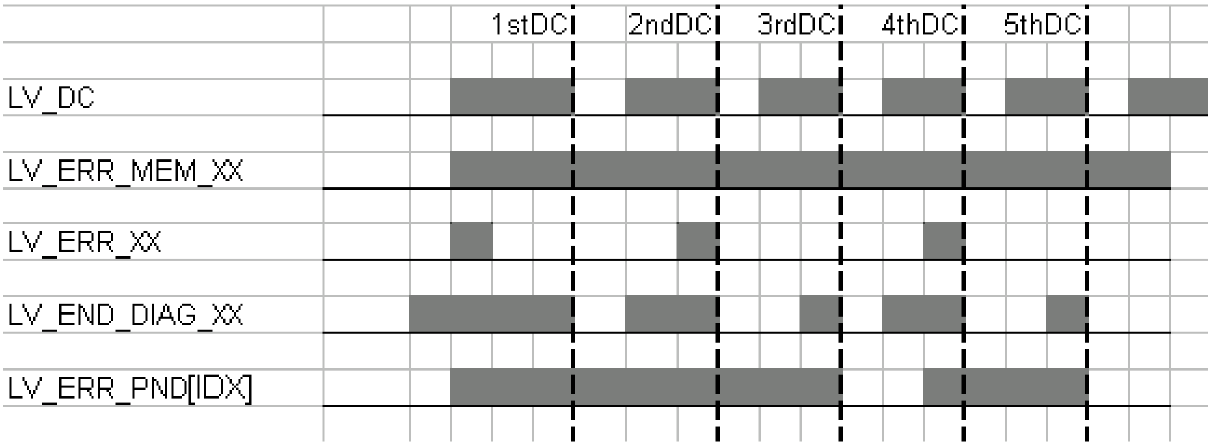
The detected failure is provided by a defective system (the error provided by this system is basic). It occurs when the diagnosis test and diagnosis condition of at least one symptom or more are true and disappears in other cases.

Present failure

The present failure is the detected failure after filtering. The real entry point of the error management is the present failure.

Pending failure

The pending failure is the detected failure after filtering. The failure shall stay pending until detection of a whole driving cycle without the failure present and the diagnosis done.



Temporary failure

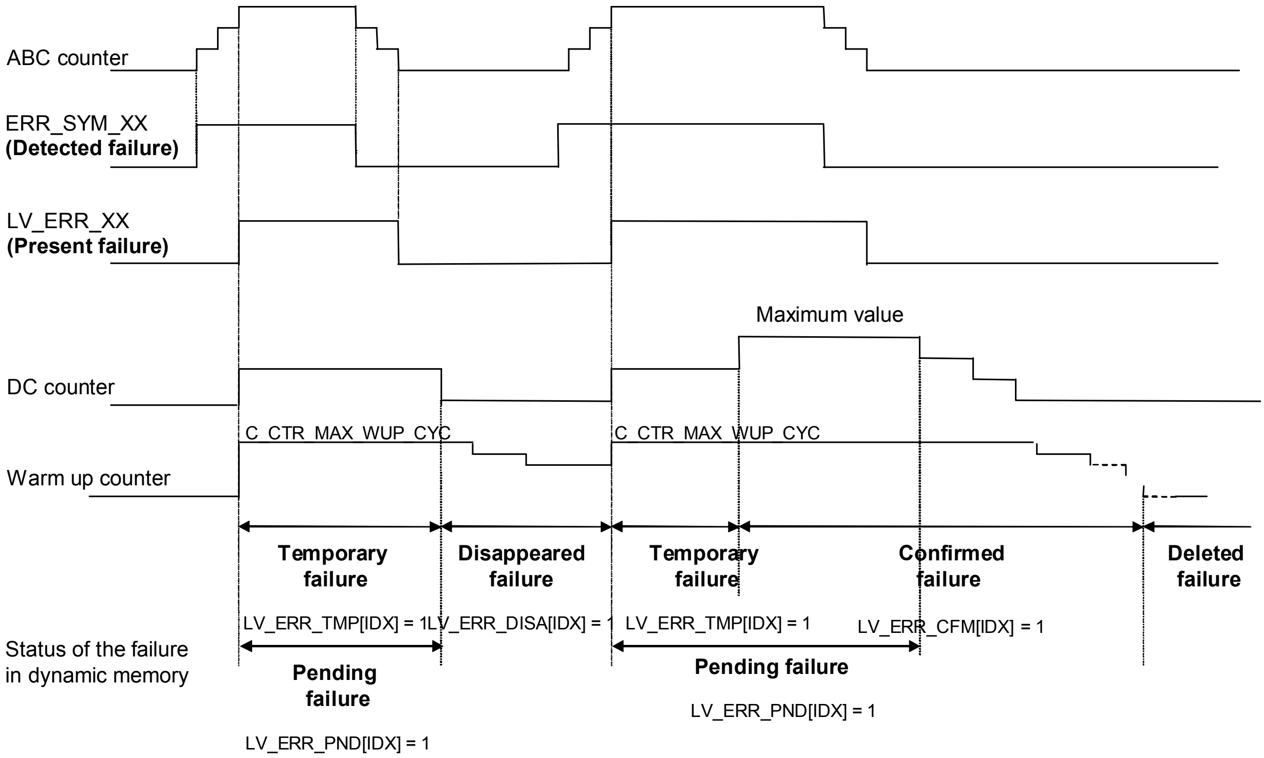
A failure is considered as temporary when this failure is present and leaves this state either when this failure is confirmed or disappeared (in this case, this failure isn't confirmed).

Disappeared failure

A failure is considered as disappeared, when no failure was detected during a complete driving cycle and the failure wasn't confirmed. It leaves this state either when the failure reoccurs or when the warm-up counter reaches 0.

Confirmed failure

A failure is considered as confirmed when the driving cycle counter reaches the maximum value and leaves this state when the warm-up counter reaches 0.



with XX = F(IDX)

27.11.1.2 Failures information

Description:

Each failure is constituted of informations, which are stored in the memory.

Some informations are allocated in the static memory. It means that these informations are available for each failure.

Static memory: XX represents any failure defined in the application (index of XX∈[1...65535])

- ABC counter	CTR_ABC_XX
- End of diagnosis counter	CTR_ABC_END_DIAG_XX
- End of diagnosis flag	LV_END_DIAG_XX
- Readiness flag	LV_READY_XX
- Diagnosis condition	LV_CDN_DIAG_XX
- Detected symptom	ERR_SYM_XX
- Present failure	LV_ERR_XX
- Failure is stored in dynamic memory	LV_ERR_MEM_XX

Some informations are allocated in the dynamic memory. It means that these informations are created in memory in case of a new dynamic failure entry (limited number of dynamic entries managed with priority rules).

Dynamic memory : IDX∈ [0...NC_NR_ERR_DYN-1]

- Frequency counter	CTR_FRC[IDX]
- Driving cycle counter	CTR_DC[IDX]
- Warm up counter	CTR_WUP_CYC[IDX]
- Memorized the first symptom (DTC)	ERR_SYM_MEM[IDX]
- Memorized the last symptom (DTC)	ERR_SYM_LST[IDX]
- Environmental data (Freeze frame)	ENVD_OBD[u][m] ENVD_CUS_CMN[v][m] ENVD_CUS_SET_CMN[W][x][m] ENVD_CUS_SET_SPC[y][z][m]
- Confirmed failure	LV_ERR_CFM[IDX]
- Disappeared failure	LV_ERR_DISA[IDX]
- Memorized MIL activation flag	LV_DC_MAX[IDX]
- Failure during DC	LV_ERR_DC[IDX]
- Temporary failure	LV_ERR_TMP[IDX]
- Pending failure	LV_ERR_PND[IDX]
- Marked failure	LV_ERR_MKD[IDX]

Remark :

To keep the link between the index of the failure in dynamic memory and the index of the failure in static memory, a transfer function is defined by : XX = F (IDX)

with m=IDX_FRF[IDX]

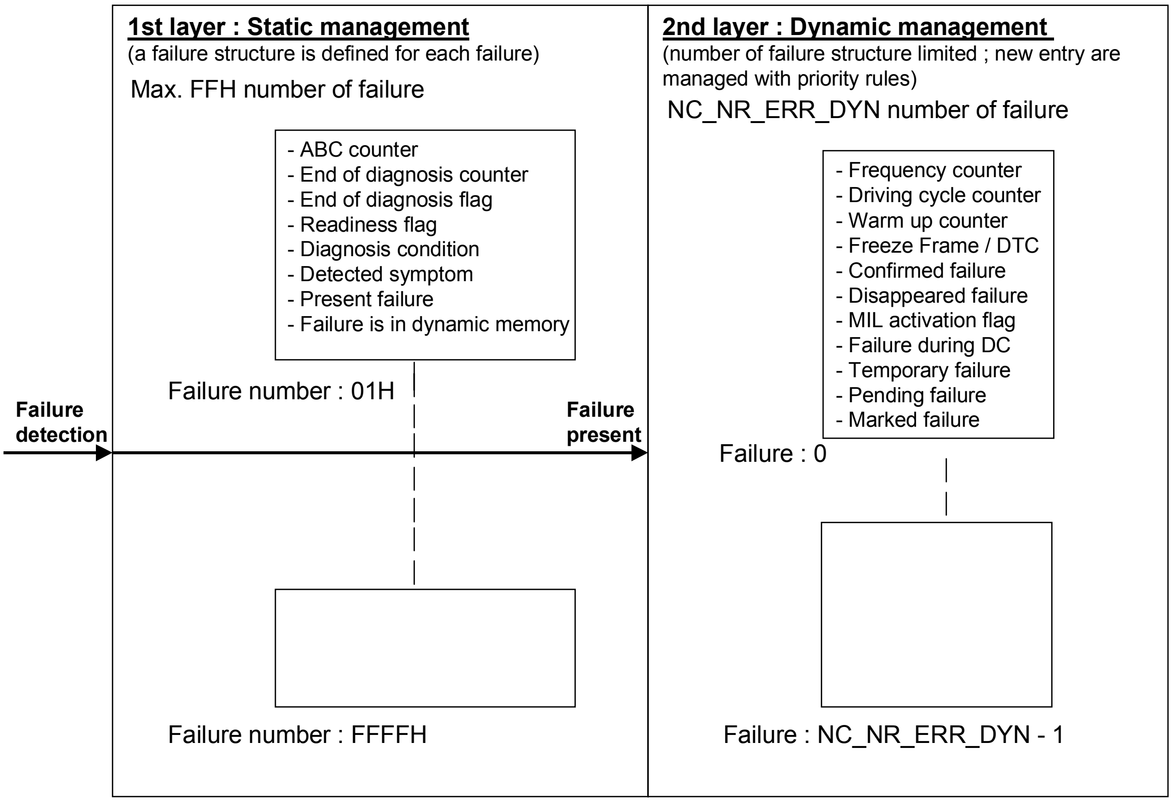
27.11.2 Memory management

Description:

As explained above, the memory is managed as two layers.

The first layer is managed as static memory. According to failure detection, ABC counter and end of diagnosis counter are incremented or decremented. Present failure information is the output of this layer. All failures are managed in the first layer.

When a present failure appears, it may provoke a dynamic entry in the second layer (dynamic) of the memory. In this second layer, only present failures are managed. The number of failure structure is limited to a maximum value defined by NC_NR_ERR_DYN.



27.11.2.1 Failures configuration

Description:

For every diagnosis two bytes exist to configure the failure. The following items can be influenced :

- control of dashboard lamp (MIL and/or Warning lamp 1 and/or Warning lamp 2)
- the emission relevance of a failure
- failure priority, used for failure storage

The definition of these bytes is done in Failure Class module.

27.11.2.2 Store / Erase a failure in 2nd layer memory

Description:

This module manages the “Entry” (storage) and “Exit” (deleting) of failures in the 2nd layer memory.

Storage :

Syntax : ACTION_ERRM_StoreErr (IN <XX>)

Parameter (in) : XX number of the failure to store in dynamic memory

Parameter (out) : -

Short description :

If other modules call the action to store the failure XX with ACTION_ERRM_StoreErr (IN<XX>) then the failure XX (with XX = F(IDX)) is taken into the 2nd layer memory and :

- frequency counter (number of occurrences of present failure) is updated ;
- the DTC is stored with ACTION_ERRM_StoreDtc (IN<IDX>) (if needed) ;
- the Freeze Frame is stored with ACTION_ERRM_StoreFrf (IN<IDX>) ;
- the index of diagnosis instance XX is stored in DIAG_INST[IDX] for visualisation

Deleting :

Syntax : ACTION_ERRM_EraseErr (IN <IDX>)

Parameter (in) : IDX index of failure in 2nd layer memory to erase

Parameter (out) : -

Short description :

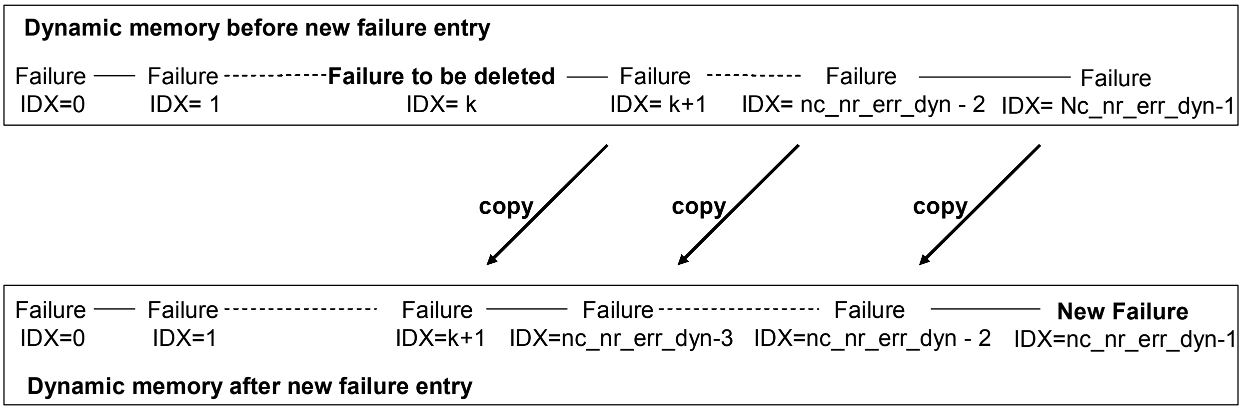
If other modules call the action to erase the failure XX with ACTION_ERRM_EraseErr (IN<IDX>) then the failure IDX is taken out of the 2nd layer memory and :

- the failure is stored in history memory with ACTION_ERRM_StoreHistory (IN<IDX>) ;
- the DTC of the failure is erased with ACTION_ERRM_EraseDtc (IN<IDX>) ;
- the Freeze Frame of the failure is erased with ACTION_ERRM_EraseFrf (IN<IDX>) ;
- similar conditions of the failure are erased with ACTION_ERRM_EraseScdn (IN<XX>) if existing

2nd layer memory description :

It is useful to sort the failures in the sequence in which they occurred ; first failure has index (IDX) number zero, second failure has index (IDX) number one, ... For example, when a new failure entry is accepted (memory is full), one failure is deleted, sorting algorithm is respected and the new failure entry is stored in last position.

In any case, sorting algorithm must be always respected.



Application conditions:

Initialization : On saved RAM lost

CTR_ERR_DYN_NR = 0 ; LV_ERR_MEM_XX = 0

CTR_FRC[IDX] = 0; LV_ERR_TMP[IDX] = 0; LV_ERR_MKD[IDX] = 0

DIAG_INST[IDX] = 0;

Recurrence: -

Activation: at Action request

Formula section:

Store the failure in 2nd layer memory :

```
ACTION_ERRM_StoreErr (IN <XX>) :
    If      LV_ERR_MEM_XX = 0
    Then    Failure XX is stored in 2nd layer memory in position IDX = CTR_ERR_DYN_NR
            LV_ERR_MEM_XX = 1
            LV_ERR_TMP[IDX] = 1
            DIAG_INST[IDX] = index of diagnosis instance XX (with XX=F(IDX))
            ACTION_ERRM_StoreDtc (IN<IDX>, SYNCHRONIZATION<CALL>)
            CTR_ERR_DYN_NR = CTR ERR_DYN_NR +1
            If      LV_MKD_MOD = 1
            Then    LV_ERR_MKD[IDX] = 1
            Endif
```

```
Else    ( IDX compliant with XX = F(IDX) )
Endif
CTR_FRC[IDX] = CTR_FRC[IDX] + 1
If      CTR_FRC[IDX]  > 255
Then    CTR_FRC[IDX] = 255
Endif
ACTION_ERRM_StoreFrf (IN<IDX>, SYNCHRONIZATION<CALL>)
(previous failure status (if disappeared) necessary for freeze frame calculation)
If      LV_ERR_DISA[IDX] = 1
Then    LV_ERR_DISA[IDX] = 0
        LV_ERR_TMP[IDX]  = 1
        ACTION_ERRM_StoreDtc (IN<IDX>, SYNCHRONIZATION<CALL>)
        (Store DTC corresponding of the first occurrence of the failure)
Endif
LV_ERR_PND[IDX] = 1
ACTION_ERRM_StoreDtcLst (IN<IDX>, SYNCHRONIZATION<CALL>)
(Store DTC corresponding of the last occurrence of the failure)
```

Clear the failure out of 2^{nd} layer memory

```
ACTION_ERRM_EraseErr (IN<IDX>) :
ACTION_ERRM_StoreHistory (IN<IDX>, SYNCHRONIZATION<CALL>)
ACTION_ERRM_EraseDtc (IN<IDX>, SYNCHRONIZATION<CALL>)
ACTION_ERRM_EraseFrf (IN<IDX>, SYNCHRONIZATION<CALL>)
If      failure is using similar conditions functionality (see Similar conditions Appli.Inc.)
        (LC_ENA_SCDN=1 and NLC_ENA_SCDN=1)
Then    ACTION_ERRM_EraseScdn (IN<XX>, SYNCHRONIZATION<CALL>)
        (with XX=F(IDX))
Endif
Failure XX (with XX=F(IDX)) is cleared (all data in 2nd layer memory related to failure
IDX are filled with 0 ; all data in static memory related to failure XX are unchanged)
LV_ERR_MEM_XX = 0 (with XX = F(IDX))
CTR_ERR_DYN_NR = CTR_ERR_DYN_NR –1
```

Temporary flag management :

For each failure in 2nd layer memory

```
    If      LV_ERR_CFM[IDX] = 1
            or LV_ERR_DISA[IDX] = 1
    Then    LV_ERR_TMP[IDX] = 0
    Endif
```

27.11.2.3 Clear failure memory

Description:

For development conveniences, it is often useful to reinit all failure in the memory. This is possible by setting the calibration bit LC_ERR_FMY_CLR to 1.

Application conditions:

Initialisation:	-
Recurrence:	-
Activation:	0 → 1 transition of LC_ERR_FMY_CLR

Formula section:

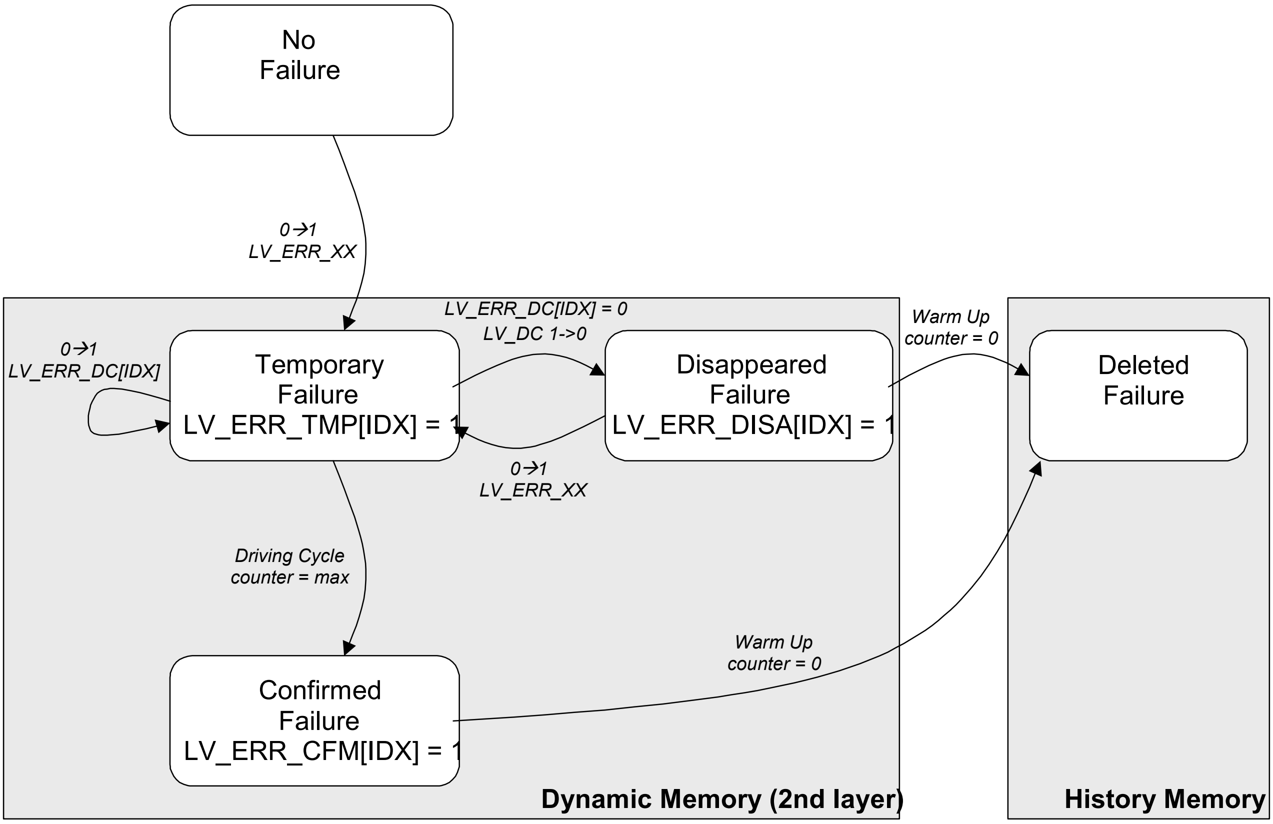
```
If      LC_ERR_FMY_CLR = 1
Then    Reset failure memory (all data in 2nd layer memory and all data in static memory)
        CTR_ERR_DYN_NR = 0
Endif
```

27.11.3 Failure management

27.11.3.1 State diagram

Description:

In the following diagram, the failure state progress is described.



There are also additional transitions possible due to :

- Overwriting/Erasement by priority rule or Tool Request :

Temporary Failure	→	Deleted Failure
Disappeared Failure	→	Deleted Failure
Confirmed Failure	→	Deleted Failure

	Driving Cycle counter value	Warm Up counter value
No Failure	0	0
Temporary Failure	[0, Max [without having reached Max	Max
Confirmed Failure	[0, Max] with having reached Max	] 0, Max]
Disappeared Failure	0	] 0, Max]
Deleted Failure	0	0

27.11.3.2 Present failure occurred

Description:

When a failure occurs (becomes present), this failure shall be stored in the 2nd layer memory (if possible). If the 2nd layer memory is full and marked mode is not set, a choice is made in the module "Priority rules" :

- RESP is set to OK by priority rules, if the failure can be stored in the 2nd layer memory ; in this case IDX gives the index of the failure to be replaced into dynamic memory;
- RESP is set to NOK by priority rules if the failure can't be stored in the 2nd layer memory.

To monitor the failure, ACTION_ERRM_TrigErrDyn shall be called when the failure occurs (the failure is evaluated or re-evaluated) or the failure is removed.

Application conditions:

Initialisation: -

Recurrence: -

Activation: at LV_END_DIAG_XX 0 → 1 transition

with (LV_ERR_XX = 1 **and** LV_ERR_MEM_XX = 1)

or

at LV_ERR_XX 1 → 0 transition

(due to the physical default disappears only: it means that a clear failure memory or diagnosis initialization aren't activation condition)

Formula section:

ACTION_ERRM_TrigErrDyn (IN<XX>, SYNCHRONIZATION<CALL>)
This action is called to indicate that a new failure has occurred, although failure is not stored.

Application conditions:

Initialisation: -

Recurrence: -

Activation: LV_ERR_XX 0 → 1 transition **with** LV_END_DIAG_XX = 1

or

LV_END_DIAG_XX 0 → 1 transition

with (LV_ERR_XX = 1 **and** LV_ERR_MEM_XX = 0)

Formula section:

If (1) CTR_ERR_DYN_NR < NC_NR_ERR_DYN

```
          or        LV_ERR_MEM_XX = 1
Then (1)  ACTION_ERRM_StoreErr (IN<XX>, SYNCHRONIZATION<CALL>)
Else (1)  If (2)    LV_MKD_MOD = 0
          Then (2)  ACTION_ERRM_PrioRule (OUT<RESP>, OUT<IDX>,
                    SYNCHRONIZATION<CALL>)
                    If (3)    RESP = OK
                    Then (3)  ACTION_ERRM_EraseErr (IN<IDX>,
                              SYNCHRONIZATION<CALL>)
                              ACTION_ERRM_StoreErr (IN<XX>,
                              SYNCHRONIZATION<CALL>)
                    Endif (3)
          Endif (2)
Endif (1)
```

ACTION_ERRM_TrigErrDyn (IN<XX>, SYNCHRONIZATION<CALL>)
This action is called to indicate that a new failure has occurred.

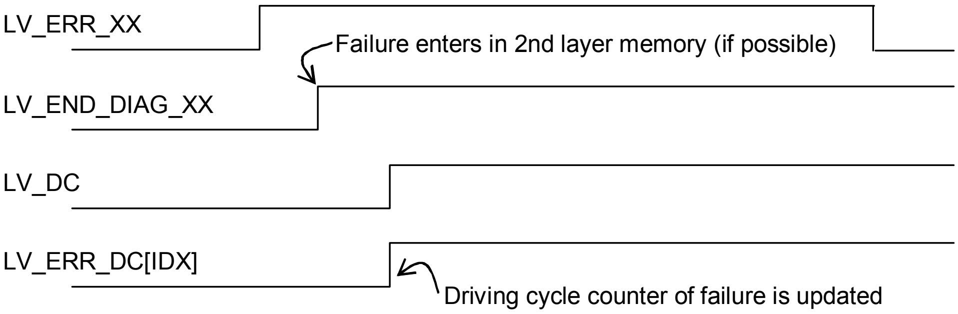
27.11.3.3 Error occurred during present driving cycle

Description:

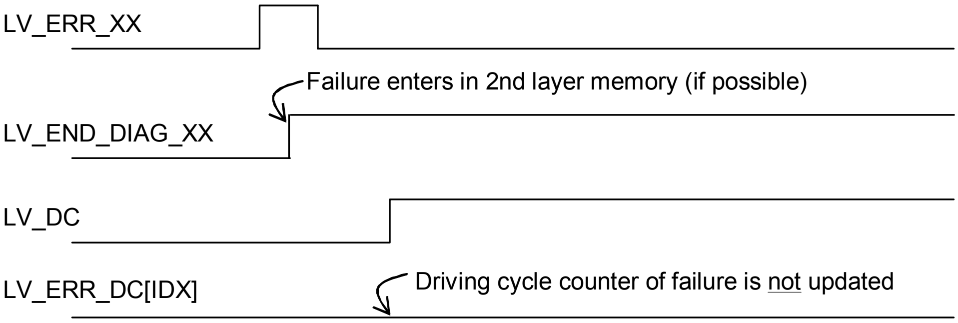
An active flag called LV_ERR_DC[IDX] indicates that an error XX (with XX = F(IDX)) has occurred during a driving cycle.

Even if the error disappears during the driving cycle, the flag remains active during the whole driving cycle.

Failure present, then driving cycle is recognized, then failure not present



Failure present, then failure not present, then driving cycle is recognized



Application conditions:

Initialisation: at ECU reset **or** transition LV_DC 1->0 after treatment done at this event

LV_ERR_DC[IDX] = 0

Recurrence: same as the diagnosis routine

all failures are updated at transition 0->1 of LV_DC

Activation: at every engine operating state

Formula section:

If LV_ERR_XX = 1 (failure is present)

and LV_END_DIAG_XX =1 (current diagnosis is available)

```
        and LV_DC = 1               (within driving cycle)
        and LV_ERR_MEM_XX =1        (failure is stored in dynamic memory)
    Then    LV_ERR_DC[IDX] = 1      (it remains 1 until the next initialization)
    Endif
```

27.11.3.4 Driving Cycle Counter management

Description:

The driving cycle counter CTR_DC[IDX] is associated to each failure which has a structure defined in second layer of memory.

This counter is incremented at each driving cycle in the following way:

- for all failures not using SCDN functionality (withLC_ENA_SCDN=0 or NLC_ENA_SCDN=0), when the associated failure gets present (LV_ERR_DC[IDX]).
- for failures using SCDN functionality (with LC_ENA_SCDN=1 and NLC_ENA_SCDN=1)
  - when the failure gets present the first time (and stored)
  - when the failure gets present again in a following driving cycle
  - when the failure gets present again in another driving cycle (not previous) (driving cycle counter of this failure is != 0/max before new occurrence), if the similar conditions functionality decides, that the incrementation is possible

The driving cycle counter is decremented at the end of a driving cycle in the following cases:

- for all failures not using SCDN functionality (or LC_ENA_SCDN=0 or NLC_ENA_SCDN=0), when the associated failure was not present the complete DC (LV_ERR_DC[IDX] = 0).
- for failures using SCDN functionality (with LC_ENA_SCDN=1 and NLC_ENA_SCDN=1)
  - when similar conditions are encountered without failure (for failure with present MIL illumination)
  - when the similar conditions functionality decides that the decrementation is possible (for temporary failure or confirmed failure with no present MIL illumination)

The driving cycle counter is useful for MIL management and warm-up counter management (see specific chapter).

With the calibration LC_ENA_SCDN it's possible to enable/disable the similar conditions functionality. If LC_ENA_SCDN = 0 the failures using SCDN functionality (see similar conditions Appl.Inc) are treated like normal failures in the driving cycle counter management.

Application conditions:

Initialization: On saved RAM lost

LV_DC_MAX[IDX] = 0
CTR_DC[IDX] = 0
LV_ERR_DISA[IDX] = 0
LV_ERR_CFM[IDX] = 0

Recurrence: -

Activation: -

Formula section:

Treatment for all failures not using SCDN functionality (with LC_ENA_SCDN=0 or NLC_ENA_SCDN=0) stored in 2^{nd} layer memory :

Conditions to increment the driving cycle counter :

```
At LV_ERR_DC[IDX] 0 → 1 transition
LV_ERR_DISA[IDX] = 0
If      LV_DC_MAX[IDX] =1                       (for confirmed failure with MIL ON)
Then    CTR_DC[IDX] = DC_MAX_XX                        (counter is forced to max.)

Else    CTR_DC[IDX] = CTR_DC[IDX] + DC_INC_XX
        If      CTR_DC[IDX] ≥ DC_MAX_XX
        Then    CTR_DC[IDX] = DC_MAX_XX
                LV_DC_MAX[IDX] = 1                     (failure becomes confirmed)
                LV_ERR_CFM[IDX] = 1
        Endif
Endif
```

Conditions to decrement the driving cycle counter :

```
At transition LV_DC 1 -> 0:
If      LV_ERR_DC[IDX] = 0
        and     LV_END_DIAG_XX  = 1
Then    LV_ERR_PND[IDX] = 0                            (Pending fault is erased)
        If      LV_ERR_CFM[IDX] = 1                    (for confirmed failure)
        Then    If LV_DC_MAX[IDX] =1
                Then    CTR_DC[IDX] = CTR_DC[IDX] - DC_DEC_XX
                        If      CTR_DC[IDX] ≤ 0
                        Then    LV_DC_MAX[IDX] = 0
                                CTR_DC[IDX] = 0
                        Endif
                Else    CTR_DC[IDX] = 0
                Endif
        Else                                           (for temporary failure)
                CTR_DC[IDX] = 0
                LV_ERR_DISA[IDX] = 1                   (failure disappeared)
        Endif
```

Endif

Treatment for failures using SCDN functionality (LC_ENA_SCDN=1 and NLC_ENA_SCDN=1) (running within DC) stored in 2nd layer memory.

Conditions to increment the driving cycle counter :

Done upon ACTION_ERRM_IncrementDCctrSCDN call:

(see also similar conditions specification)

Description:

Syntax :	ACTION_ERRM_IncrementDCctrScdn (IN <XX>)	
Parameter (in) :	XX	failure in static memory with XX=F(IDX)
Parameter (out) :		

Short description:

With the ACTION_ERRM_IncrementDCctrScdn (IN <XX>) the driving cycle counter for the failure XX stored in 2nd layer memory at position IDX, which is using similar conditions functionality, is incremented. This action is called at the end of a diagnosis window in the following cases :

- failure occurred and similar conditions are recorded
- failure already stored in a previous driving cycle. Now failure gets present again and was also present in last driving cycle.
- failure already stored in a previous driving cycle. Now failure gets present again with SCDN recognition for detected failure.
- failure already stored in a previous driving cycle and LV_DC_MAX[IDX] = 1 (MIL is illuminated, if LC_MIL_ON =1).

For details see similar conditions specification.

Application conditions:

Initialization:	-
Recurrence:	-
Activation:	at action request

Formula section:

ACTION_ERRM_IncrementDCctrScdn (IN<XX>) :

```
If      LV_ERR_MEM_XX = 1              (failure XX is stored in 2nd layer memory)
Then    LV_ERR_DISA[IDX] = 0
        If    LV_DC_MAX[IDX] =1        (for confirmed failure with MIL on)
        Then  CTR_DC[IDX] = DC_MAX_XX        (counter is forced to max.)
        Else  ACTION_ERRM_ConfirmErrScdn (OUT<DCconf>, IN<XX>,
              SYNCHRONIZATION<CALL>)
              (allows direct failure confirmation; see similar conditions Appl.Inc.)
              If      DCconf = YES
```

```
                    Then   CTR_DC[IDX] = DC_MAX_XX        (counter is forced to max.)
                    Else   CTR_DC[IDX] = CTR_DC[IDX] + DC_INC_XX
                    Endif
                    If     CTR_DC[IDX] ≥ DC_MAX_XX
                    Then   CTR_DC[IDX] = DC_MAX_XX
                           LV_DC_MAX[IDX] = 1             (failure confirmed)
                           LV_ERR_CFM[IDX] = 1
                           LV_ERR_TMP[IDX] = 0
                    Endif
            Endif
Endif
```

Conditions to decrement the driving cycle counter :

At transition LV_DC 1 -> 0:

```
If          LV_ERR_DC[IDX] = 0
            and    LV_END_DIAG_XX  = 1
Then        ACTION_ERRM_DecrementDCctrSCDN (OUT<DCdec>, IN<XX>,
            SYNCHRONIZATION<CALL>)        (with XX=F(IDX))
            The similar conditions functionality decides, if it's possible to decrement the
            driving cycle counter.
            If     DCdec = YES                            (decrementing is possible)
            Then   LV_ERR_PND[IDX] = 0                    (Pending fault is erased)
                   If     LV_ERR_CFM[IDX] = 1             (for confirmed failure)
                   Then   If     LV_DC_MAX[IDX] = 1
                          Then   CTR_DC[IDX] = CTR_DC[IDX] - DC_DEC_XX
                                 If     CTR_DC[IDX] ≤ 0
                                 Then   LV_DC_MAX[IDX] = 0
                                        CTR_DC[IDX] = 0
                                 Endif
                          Else
                                 CTR_DC[IDX] = 0
                          Endif
                   Else                                   (for temporary failure)
                      CTR_DC[IDX] = 0
                      LV_ERR_DISA[IDX] = 1                (failure disappeared)
                   Endif
            Endif
```

```
Endif
```

27.11.3.5 Warm-Up counter management

Description:

As soon as the failure gets present and stored in the memory, the warm-up cycle counter is initialized.

With every valid warm-up cycle with no error occurrence, the warm-up cycle counter is decremented by 1. If the warm-up cycle counter reaches zero the error is deleted from the 2nd layer memory.

Application conditions:

Initialization: on saved ram lost
CTR_WUP_CYC[IDX]= 0
Recurrence: -
Activation: only at transition LV_DC 1 -> 0

Formula section:

Treatment for all failures not using SCDN functionality (or LC_ENA_SCDN = 0) stored in 2nd layer memory :

Condition to decrement the warm up counter :

The following treatment is done for all failure stored in 2nd layer memory **and only at transition LV_DC 1 -> 0** (**before** treatment of driving cycle counter decrementing at end of driving cycle):

```
If      ( LV_WUP_CYC = 1  and    C_CTR_MAX_WUP_CYC != 255 )
        the warm-up cycle is recognised and failure healing is possible
Then    If      ( CTR_DC[IDX] = 0   and    LV_ERR_DC[IDX] = 0 )
                the MIL is extinguished and no failure in this driving cycle
                or
                ( CTR_DC[IDX] != 0 and LV_DC_MAX[IDX] = 0
                case before the MIL may be illuminated
                and LV_ERR_DC[IDX] = 0 and LV_END_DIAG_XX = 1 )
                the failure is no more pending
        Then    CTR_WUP_CYC[IDX] = CTR_WUP_CYC[IDX] - 1
                If      CTR_WUP_CYC[IDX] = 0
                        Then   ACTION_ERRM_EraseErr (IN<IDX>,
                        SYNCHRONIZATION<CALL>)
                Endif
        Endif
Endif
```

Treatment for failures using SCDN functionality (and LC_ENA_SCDN = 1) (running within DC) stored in 2nd layer memory.

Condition to decrement the warm up counter :

The following treatment is done for all failure stored in 2nd layer memory **and only at transition LV_DC 1 -> 0** (**before** treatment of driving cycle counter decrementation at end of driving cycle):

```
If      ( LV_WUP_CYC = 1  and   C_CTR_MAX_WUP_CYC != 255 )
        the warm-up cycle is recognized and failure healing is possible
Then    If      ( CTR_DC[IDX] != 0 and LV_DC_MAX[IDX] = 0
                case before the MIL may be illuminated
                and LV_ERR_DC[IDX] = 0 and LV_END_DIAG_XX = 1 )
        then    ACTION_ERRM_DecrementDCctrSCDN (OUT<DCdec>, IN<XX>,
                SYNCHRONIZATION<CALL>) (with XX=F(IDX))
        The similar conditions functionality decides, if it's possible to decrement the
        driving cycle counter.
                If      DCdec = YES                (the failure is no more pending)
                Then    CTR_WUP_CYC[IDX] = CTR_WUP_CYC[IDX] - 1
                        If      CTR_WUP_CYC[IDX] = 0
                                Then  ACTION_ERRM_EraseErr (IN<IDX>,
                                SYNCHRONIZATION<CALL>)
                        Endif
                Endif
        Else    If      ( CTR_DC[IDX] = 0   and   LV_ERR_DC[IDX] = 0 )
                the MIL is extinguished and no failure in this driving cycle
                Then    CTR_WUP_CYC[IDX] = CTR_WUP_CYC[IDX] - 1
                        If      CTR_WUP_CYC[IDX] = 0
                                Then  ACTION_ERRM_EraseErr (IN<IDX>,
                                SYNCHRONIZATION<CALL>)
                        Endif
                Endif
        Endif
Endif
```

Condition to reload the warm up counter :

The following treatment is done for failure XX when it is stored in 2nd layer memory **when ACTION_ERRM_StoreErr (IN<XX>) is called :**

CTR_WUP_CYC[IDX] = C_CTR_MAX_WUP_CYC

27.11.4 Miscellaneous services to get information

27.11.4.1 API for determination of pending status of failures

General information:

This API shall be called to determine if any XX failure is pending or not.

Description:

Syntax:	ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT <PendingStatus>)	
Parameter(in):	XX	Diagnostic instance of the failure
Parameter(out):	PendingStatus	State of the failure : - TRUE : the XX failure is pending (or it's impossible to determine its status) - FALSE : the XX failure is not pending
Short Description:	This API shall be used to verify if a failure stored in the dynamic memory has the pending status or not. When it impossible to determine if the fault is pending or not (failure not store because the dynamic memory is full), this failure should be considered as pending anyway.	

Application conditions:

Initialisation: -

Recurrence: -

Activation: at action call

Deactivation: -

Formula section:

```
If (1)          { the XX failure analysed is present in the dynamic memory }
                LV_ERR_MEM_XX = 1  ( with XX = F(IDX) )
Then (1)
                If (2)          { IDX is the failure index, in the dynamic memory }
                                LV_ERR_PND[IDX] = 1
                Then (2)
                                PendingStatus = TRUE
                Else (2)
                                PendingStatus = FALSE
                Endif (2)
Else (1)
                { failure not stored in the dynamic memory }
```

```
        If (2)      LV_ERR_XX = 1
        Then (2)
                    { the failure is stored in the static memory but it's not present in the
                    dynamic memory : impossible to determine its status }
                    { then the failure shall be considered as pending }
                    PendingStatus = TRUE
        Else (2)
                    { the failure is neither stored in the static memory nor in the
                    dynamic memory }
                    PendingStatus = FALSE
        Endif (2)
    Endif (1)
```

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_ERR_DYN	1	1...FFH	1...255	1	[-]
Maximum number of failure defined in dynamic structure (typical value : 10)					

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_MAX_WUP_CYC	1	0...FFH	0...255	1	[-]
Initialization value for the warm-up cycle of diagnostics					
LC_ERR_FMY_CLR	1	0...1H	0...1	1	[-]
Boolean to clear failure memory when set to 1					

27.12 Dynamic Error Management (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_WUP_CYC	V/O	0...1H	0...1	1	[-]
Warm-up cycles calculation inhibition					
TCO_DSL_CMN	V/O	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature (used only in ERRM for GS-DS diversity)					
TCO_ST_DSL_CMN	V/O	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature at start (used only in ERRM for GS-DS diversity)					

Input data:

TCO	TCO_ST		

Export actions:

ACTION_ERRM_TrigErrDyn (IN <XX>)
This action is called by the error management function when a new failure occurs

27.12.1 Warm-up cycle inhibition

FUNCTION DESCRIPTION:

Description:

Wurm-up cycle is not inhibited

Application conditions:

Initialisation: at ECU reset

LV_INH_WUP_CYC = 0

TCO_DSL_CMN = TCO

TCO_ST_DSL_CMN = TCO_ST

Recurrence: 100 ms

Activation: LV_IGK = 1

Deactivation: -

Formula section:

LV_INH_WUP_CYC = 0

TCO_DSL_CMN = TCO

TCO_ST_DSL_CMN = TCO_ST

27.12.2 API for Trigger Event

General information:

This file defines the Action "ACTION_ERRM_TrigErrDyn" called by the error management function when a failure occurs or disappears.
The contain of this action should be provided by the project team

Description:

Syntax: ACTION_ERRM_TrigErrDyn (IN <XX>)

Parameter (in): XX diagnostic instance of failure ≡ index of failure in static memory

Parameter (out): -

Short description: This action allows to inform the communication tools in line as soon as a failure occurs

Application conditions:

Initialisation: -

Recurrence: -

Activation: at action request

Formula section:

ACTION_ERRM_TrigErrDyn (XX): **additional function not defined yet**

27.13 Cycle manager

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DC	V/O	0...1H	0...1	1	[-]
driving cycle status flag					
LV_WUP_CYC	V/O	0...1H	0...1	1	[-]
Warm-up cycle status flag					
LV_STATE_WUP	V/O	0...1H	0...1	1	[-]
tco state for exceeding the warm-up cycle tco threshold					

Input data:

LV_IGK	LV_ST_END	LV_ES	TCO_DSL_CMN
TCO_ST_DSL_CMN	LV_INH_WUP_CYC		

27.13.1 Driving cycle

FUNCTION DESCRIPTION:

General information:

Description:

A driving cycle consists of a vehicle operation phase from engine startup to engine shutoff and includes the power-lach phase after the engine running phase (In order manage all OBD failures set only in PWL phase).

After key is on, the flag LV_DC is set when the engine exits from “engine start” state.

The flag LV_DC is reset at the end of the power-latch phase before the NVMY management. In case of a LV_IGK 0→1 transition during the powerlatch phase or at ECU reset LV_DC is also reseted.

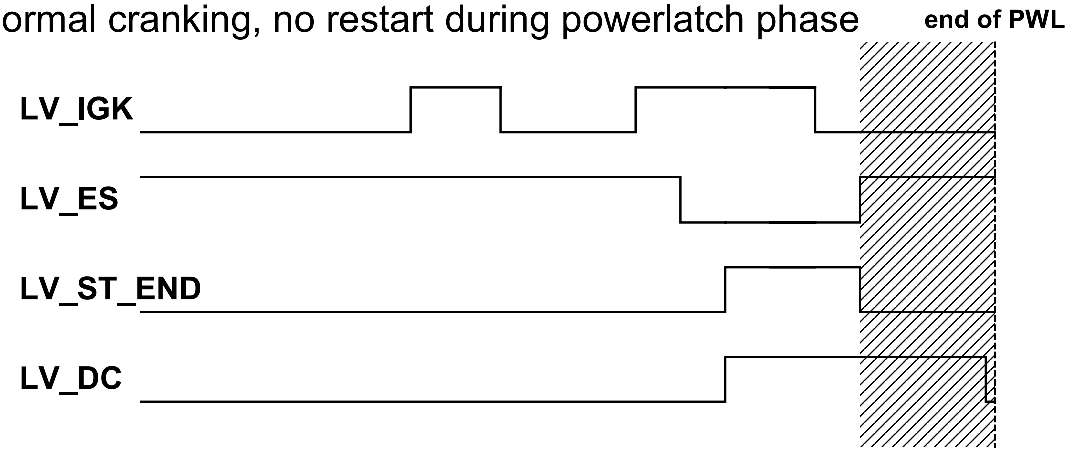
Remark :

For simplification and global coherence, the following rule is applied:

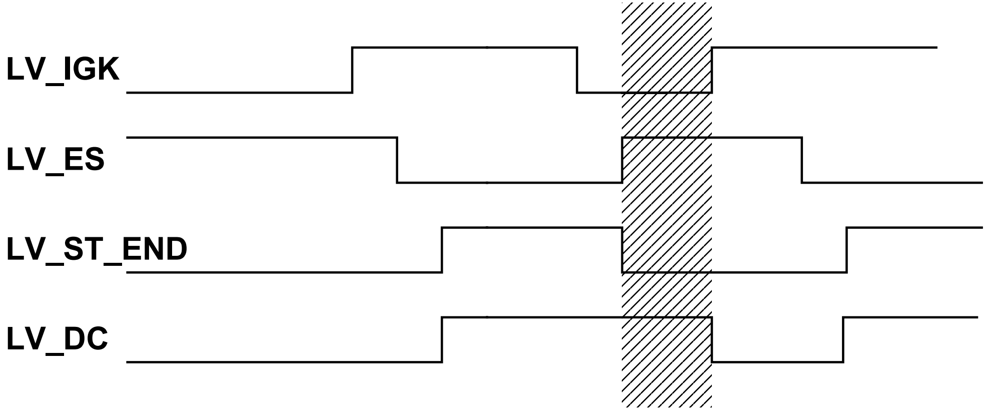
Engine stalling is ignored (if engine restarts after stalling, without key-off/key-on transition, the cycle is considered as the same driving cycle).

Signal flow diagram:

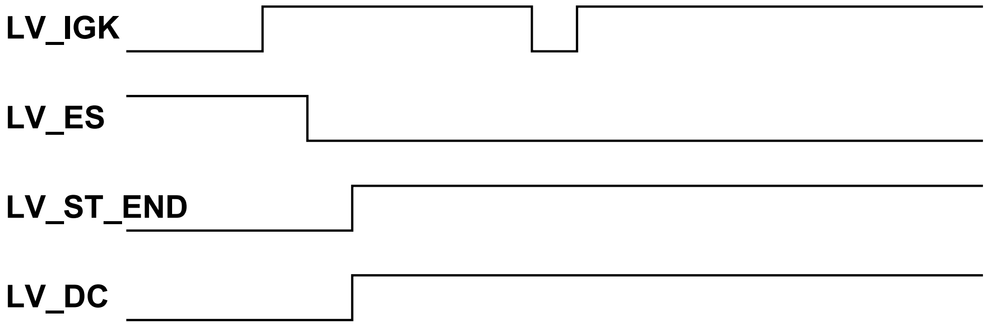
Normal cranking, no restart during powerlatch phase



Normal cranking, with restart during powerlatch phase



Short key off/on while engine is running



Application conditions:

Initialisation: At ECU reset

LV_DC = 0

Before storage of data of the error management within the NVMY the driving cycle is finished.

Recurrence: -

Activation: *see Formula section*

Formula section:

Activation:

```
If      LV_ST_END 0→1 transition
Then
        LV_DC = 1
Endif
```

Deactivation:

```
If      LV_IGK 0→1 transition
        and
        LV_ES =1
Then
        LV_DC = 0
Endif

If      End of power-latch phase is reached (before NVMY storing)
Then
        LV_DC = 0
Endif
```

27.13.2 Warm-Up Cycle and Warm-Up Status

FUNCTION DESCRIPTION:

General information:

The detection of the warm-up cycle is based on the coolant temperature. The warm-up cycle is detected as soon as the coolant temperature reaches C_TCO_WUP_THD and the change with regard to the starting temperature exceeds C_TCO_WUP_INC. These values are defined by law, so a change of these values is only for development validation commodity.

The application must guarantee warm-up cycle detection when a coolant temperature sensor error exists.

Therefore if a coolant temperature sensor error has been debounced, the threshold for warm-up cycle detection are referred to the backup coolant temperature value computed by the engine control (see chapter “Diagnosis and emergency operation“).

The warm-up status is needed for similar driving conditions test.

Application conditions:

Initialisation: *At ECU reset or LV_DC 1→0 transition (after warm-up cycle counter management)*

LV_WUP_CYC =0
LV_STATE_WUP = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

```
If    LV_INH_WUP_CYC = 0
Then
        If      TCO_DSL_CMN > C_TCO_WUP_THD
        Then
                LV_STATE_WUP = 1
                If      TCO_DSL_CMN > (TCO_ST_DSL_CMN + C_TCO_WUP_INC)
                Then
                        LV_WUP_CYC = 1
                Endif
        Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_WUP_THD	1	0...FEH	-48...142.5	0.75	[°C]
Minimum value for warm-up cycle detection					
C_TCO_WUP_INC	1	0...FEH	-48...142.5	0.75	[°C]
Minimum increase of the coolant temperature for warm-up cycle detection					

Calibration data detailed description:

C_TCO_WUP_THD =	at least 72 °C for gasoline engines	(OBD II requirement)
	at least 70 °C for gasoline engines	(EOBD requirement)
C_TCO_WUP_INC =	at least 23 °C for gasoline engines	(OBD II requirement)
	at least 22 °C for gasoline engines	(EOBD requirement)

In case of Diesel systems, recommended values are the same, except :

C_TCO_WUP_THD = at least 60 °C for gasoline engines (OBD II requirement)

27.14 Environmental data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ENVD_OBD[NC_NR_ENVD_OBD][NC_NR_ERR_DYN]	V/O/S	0...FFH	0...255	1	[-]
Freeze frame FRF_OBD : environmental data (fixed by law)					
ENVD_CUS_CMN[NC_NR_ENVD_CUS_CMN][NC_NR_ERR_DYN]	V/O/S	0...FFH	0...255	1	[-]
Freeze frame FRF_CUS_CMN : environmental data common to all failure (not fixed by law)					
ENVD_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN][NC_NR_FRF_SET][NC_NR_ERR_DYN]	V/O/S	0...FFH	0...255	1	[-]
Freeze frame FRF_CUS_SET_CMN : environmental data common to all failure (not fixed by law) stored in set					
ENVD_CUS_SET_SPC[NC_NR_ENVD_CUS_SET_SPC][NC_NR_FRF_SET][NC_NR_ERR_DYN]	V/O/S	0...FFH	0...255	1	[-]
Freeze frame FRF_CUS_SET_SPC : environmental data specific to each failure (not fixed by law) stored in set					
ENVD_PREV_OBD[NC_NR_ENVD_OBD][NC_NR_ENVD_PREV]	O/S	0...FFH	0...255	1	[-]
Prestored freeze frame FRF_PREV_OBD as content for FRF_OBD;					
ENVD_PREV_CUS_CMN[NC_NR_ENVD_CUS_CMN][NC_NR_ENVD_PREV]	O/S	0...FFH	0...255	1	[-]
Prestored freeze frame FRF_PREV_CUS_CMN as content for FRF_CUS_CMN;					
ENVD_PREV_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN][NC_NR_ENVD_PREV]	O/S	0...FFH	0...255	1	[-]
Prestored Freeze frame FRF_PREV_CUS_SET_CMN as content for FRF_CUS_SET_CMN					
ENVD_PREV_CUS_SET_SPC[NC_NR_ENVD_CUS_SET_SPC][NC_NR_ENVD_PREV]	O/S	0...FFH	0...255	1	[-]
Prestored freeze frame FRF_PREV_CUS_SET_SPC :as content for FRF_CUS_SET_SPC					
IDX_FRF[NC_NR_ERR_DYN]	V/O/S	0...FFH	0...255	1	[-]
Pointer to make the link between the failure and the freeze frame					

Note:

The link between the failure stored in the dynamic error mangement at position *IDX* is made with IDX_FRF[*IDX*]. E. g. a failure stored at the position *IDX* = 2 belongs to freeze frame ENVD_..._[m] with m = IDX_FRF[2]

Input data:

ENVD_CONF_OBD[NC_NR_ENVD_OBD]	ENVD_CONF_CUS_CMN[NC_NR_ENVD_CUS_CMN]	ENVD_CONF_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN]
CTR_FRC[NC_NR_ERR_DYN]	NC_NR_ERR_DYN	LV_ERR_DISA[NC_NR_ERR_DYN]

Export actions:

ACTION_ERRM_StoreFrf (IN <IDX>)
This action stores the freeze frame of the failure IDX
ACTION_ERRM_EraseFrf (IN <IDX>)
This action erases the freeze frame of the failure IDX
ACTION_ERRM_StorePrevFrf (IN <XX>)
This action stores the prestored freeze frame for the diagnosis instance XX

FUNCTION DESCRIPTION:

General information:

This module describes the structure of the freeze frame and manages the storage and deleting of the freeze frame.

A freeze frame is a group of vehicle environment data catched when a failure occurs.

Freeze frame is filled up with variables, which are not limp home values : the variables are set up before going to the limp home.

A freeze frame is made of many parts :

- FRF_OBD: Environmental data, which are common to all failures and stored on time. It’s only updated, if the failure has disappeared before and occurs again.

  The content is defined by law. It’s called usually CARB Freeze Frame and the content is according to SAE J1979.

  Size of this part is defined by configuration data NC_NR_ENVD_OBD affected value is in the Environmental data (Appl. Inc.) module.

  Contents of this part is fixed and defined in Environmental data (Appl. Inc.) module.

- FRF_CUS_CMN : Environmental data, which are common to all failure.

  It’s stored one time, if NC_ ENVD_CUS_CMN_UPD = 0.

  With NC_ ENVD_CUS_CMN_UPD = 1, FRF_CUS_CMN is updated, if the failure was disappeared before and occurs again.

  The content is not required by law.

  Size of this part is defined by configuration data NC_NR_ENVD_CUS_CMN; affected value is in the Environmental data (Appl. Inc.) file.

  Contents of this part is fixed and defined in the Environmental data (Appl. Inc.) module.

- FRF_CUS_SET_CMN : Environmental data, which are common to all failure and stored many times. The content is not required by law. For each failure occurrence several catches (called sets) can be done.

  Size of this part is defined by configuration data, affected value is in the Environmental data (Appl. Inc.) module :

  NC_NR_FRF_SET for number of sets per freeze frame

  NC_NR_ENVD_CUS_SET_CMN for number of common data (in bytes) in each set

  Contents of this part is fixed and defined in the Environmental data (Appl. Inc.) module.

- FRF_CUS_SET_SPC : Environmental data, which are specific to each failure and stored many times. The content is not required by law. For each failure occurrence several catches (called sets) can be done.

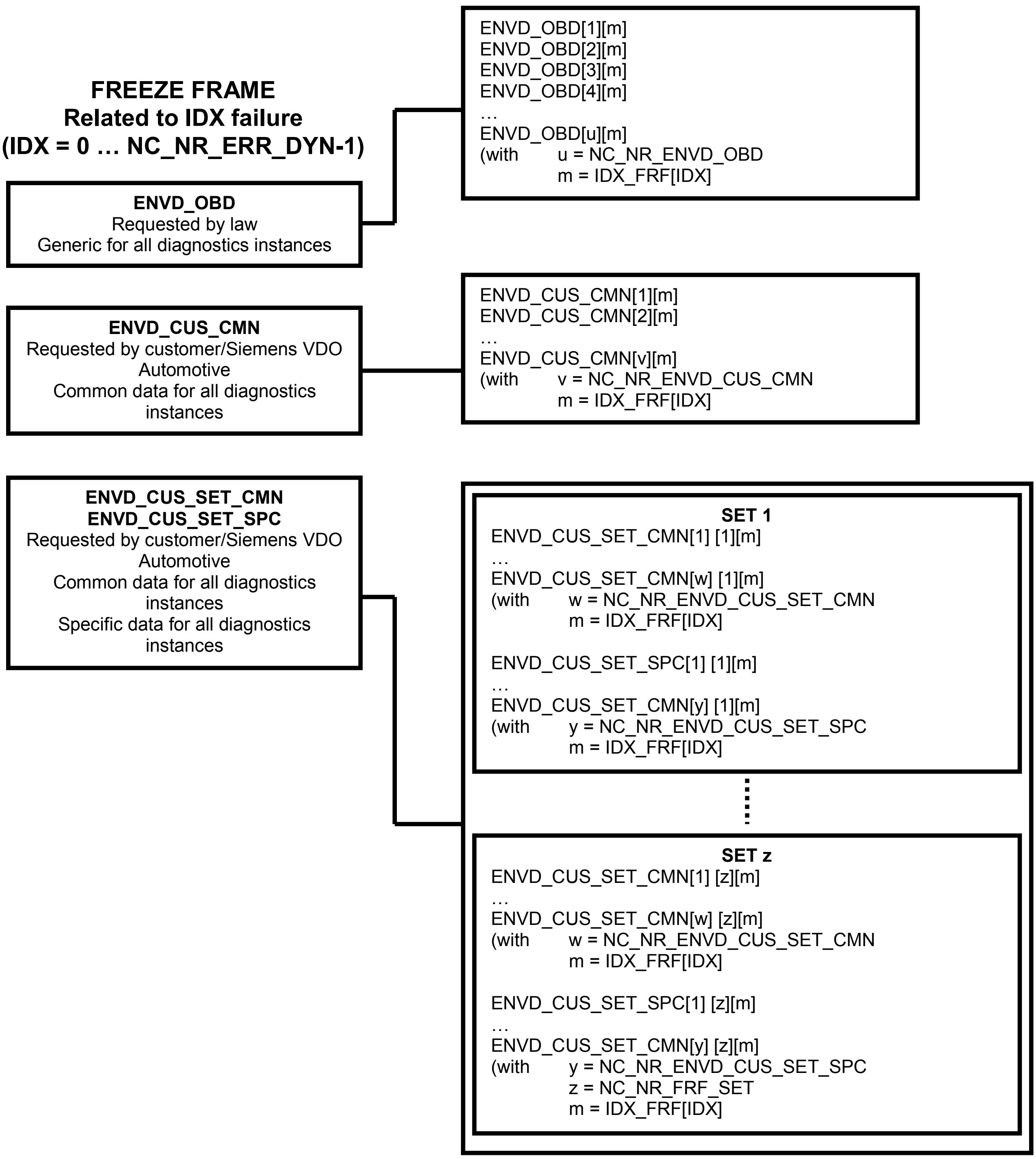
  Size of this part is defined by configuration data, affected value is in Environmental data (Appl. Inc.) module :

  NC_NR_FRF_SET for number of sets per freeze frame

  NC_NR_ENVD_CUS_SET_SPC for number of specific data (in bytes) in each set

  Contents of this part is defined by calibration, process is described in the last chapter of this module.

Signal flow diagram:



FREEZE FRAME
Related to IDX failure
(IDX = 0 … NC_NR_ERR_DYN-1)

ENVD_OBD
Requested by law
Generic for all diagnostics instances

ENVD_OBD[1][m]
ENVD_OBD[2][m]
ENVD_OBD[3][m]
ENVD_OBD[4][m]
…
ENVD_OBD[u][m]
(with u = NC_NR_ENVD_OBD
m = IDX_FRF[IDX]

ENVD_CUS_CMN
Requested by customer/Siemens VDO Automotive
Common data for all diagnostics instances

ENVD_CUS_CMN[1][m]
ENVD_CUS_CMN[2][m]
…
ENVD_CUS_CMN[v][m]
(with v = NC_NR_ENVD_CUS_CMN
m = IDX_FRF[IDX]

ENVD_CUS_SET_CMN
ENVD_CUS_SET_SPC
Requested by customer/Siemens VDO Automotive
Common data for all diagnostics instances
Specific data for all diagnostics instances

SET 1
ENVD_CUS_SET_CMN[1] [1][m]
…
ENVD_CUS_SET_CMN[w] [1][m]
(with w = NC_NR_ENVD_CUS_SET_CMN
m = IDX_FRF[IDX]

ENVD_CUS_SET_SPC[1] [1][m]
…
ENVD_CUS_SET_CMN[y] [1][m]
(with y = NC_NR_ENVD_CUS_SET_SPC
m = IDX_FRF[IDX]

SET z
ENVD_CUS_SET_CMN[1] [z][m]
…
ENVD_CUS_SET_CMN[w] [z][m]
(with w = NC_NR_ENVD_CUS_SET_CMN
m = IDX_FRF[IDX]

ENVD_CUS_SET_SPC[1] [z][m]
…
ENVD_CUS_SET_CMN[y] [z][m]
(with y = NC_NR_ENVD_CUS_SET_SPC
z = NC_NR_FRF_SET
m = IDX_FRF[IDX]

Remark: This signal flow diagram doesn't include the prestored freeze frame functionality described in chapter 1.4.

27.14.1 Freeze frame storage

Description:

Syntax : ACTION_ERRM_StoreFrf (IN <IDX>).

Parameter (in) : IDX Index of failure in 2nd layer memory to store the freeze frame

Parameter (out) : -

Short description :

When the failure IDX gets present, the storage of a freeze frame is done in the 2nd layer memory by calling the action ACTION_ERRM_StoreFrf (IDX).

If several failures occur (TCO sensor, injector 1, injector 4), a freeze frame is stored for each failure. For ram memory limitation, the number of freeze frame is limited to NC_NR_ERR_DYN.

If same failure occurs several times (occurrence of the failure is defined by the frequency counter CTR_FRC[IDX]), the following rules are applied :

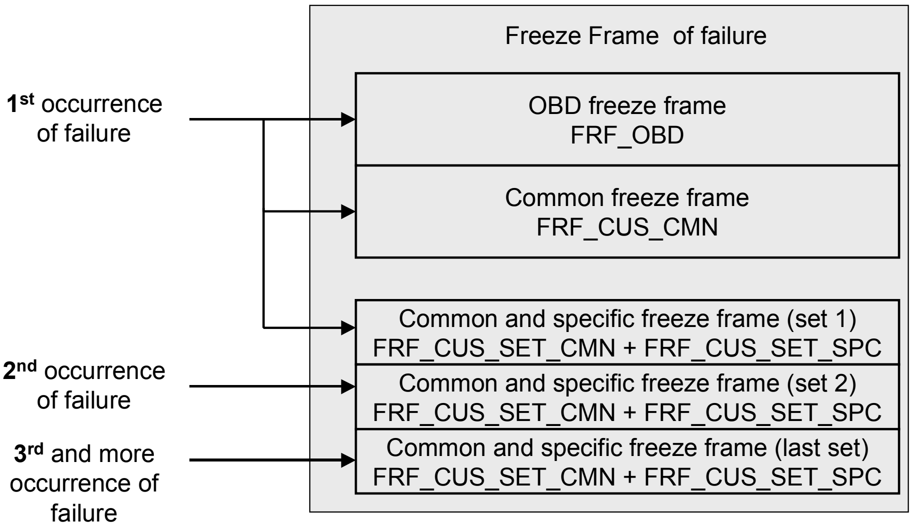
- 1st occurrence :

  ⇨ Store FRF_OBD +FRF_CUS_CMN +FRF_CUS_SET_CMN + FRF_CUS_SET_SPC
- occurrence ∈ [2 to NC_NR_FRF_SET]

  ⇨ Store FRF_CUS_SET_CMN + FRF_CUS_SET_SPC
- occurrence > NC_NR_FRF_SET

  ⇨ Replace last FRF_CUS_SET_CMN + FRF_CUS_SET_SPC

Example with NC_NR_FRF_SET set to 3



Freeze frame informations are available as long as the failure is not erased in 2nd layer memory.

Remark: Special behavior for FRF_OBD and FRF_CUS_CMN in case of failure status change from disappeared to temporary/confirmed.

Application conditions:

Initialization: after system initialization the freeze frame is restored from non volatile memory or in case of a damaged non volatile memory the freeze frame is initialized with zero

Recurrence: -

Activation: at Action request

Formula section:

```
ACTION_ERRM_StoreFrf (IDX) :
If      CTR_FRC[IDX] = 1
Then    ( 1st occurrence : storage
          With  u = 1 ... NC_NR_ENVD_OBD
                v = 1 ... NC_NR_ENVD_CUS_CMN
                w = 1 ... NC_NR_ENVD_CUS_SET_CMN
                y = 1 ... NC_NR_ENVD_CUS_SET_SPC )
        If  (NC_NR_ENVD_PREV != 0        and
            XX = diagnosis instance XX using prestored freeze frame functionality (see
            Environmental data (Appl. Inc.) module) with XX = F(IDX))
        Then  ENVD_OBD[u][m] = ENVD_PREV_OBD[u][t]
              ENVD_CUS_CMN[v][m] = ENVD_PREV_CUS_CMN[v][t]
              ENVD_CUS_SET_CMN[w][1][m] = ENVD_PREV_CUS_SET_CMN[w][t]
              ENVD_CUS_SET_SPC[y][1][m] = ENVD_PREV_CUS_SET_SPC[y][t]
              (with m=IDX_FRF[IDX]
               with t related to failure XX, see Environmental data Appli.Inc.)

        Else  ENVD_OBD[u][m] = ENVD_CONF_OBD[u]
              ENVD_CUS_CMN[v][m] = ENVD_CONF_CUS_CMN[v]
              ENVD_CUS_SET_CMN[w][1][m] = ENVD_CONF_CUS_SET_CMN[w]
              ENVD_CUS_SET_SPC[y][1][m] = ID_ERR_ENVD_XX[y]
              (with XX = F(IDX) )
        Endif
```

```
Else    (occurrence 2 to NC_NR_FRF_SET
        With    w = 1 .. NC_NR_ENVD_CUS_SET_CMN
        y = 1 ... NC_NR_ENVD_CUS_SET_SPC
        z = CTR_FRC[IDX], if CTR_FRC[IDX] ≤ NC_NR_FRF_SET : storage
        z = NC_NR_FRF_SET, if CTR_FRC[IDX] > NC_NR_FRF_SET : update
        ( with m=IDX_FRF[IDX] )
        If  (NC_NR_ENVD_PREV != 0        and
            XX = diagnosis instance XX using prestored freeze frame functionality
            (see Environmental data (Appl. Inc.) module) with XX = F(IDX))

        Then  If (LV_ERR_DISA[IDX] = 1)
              failure was in disappeared state before this occurrence
              Then ENVD_OBD[u][m] = ENVD_PREV_OBD[u][t]
                   FRF_OBD is updated
                   If (NC_ ENVD_CUS_CMN_UPD = 1)
                   Then ENVD_CUS_CMN[v][m] = ENVD_PREV_CUS_CMN[v][t]
                        FRF_CUS_CMN is updated
                   Endif
              Endif
              ENVD_CUS_SET_CMN[w][z][m] = ENVD_PREV_CUS_SET_CMN[w][t]
              ENVD_CUS_SET_SPC[y][z][m] = ENVD_PREV_CUS_SET_SPC[y][t]
             (with t related to failure XX, see Environmental data Appli.Inc.)
        Else  If     (LV_ERR_DISA[IDX] = 1)
                     Failure was in disappeared state before this occurrence
              Then   ENVD_OBD[u][m] = ENVD_CONF_OBD[u]
                     Freeze frame ENVD_OBD is updated
                     If      (NC_ ENVD_CUS_CMN_UPD = 1)
                     Then    ENVD_CUS_CMN[v][m]= ENVD_CONF_CUS_CMN[v]
                             FRF_CUS_CMN is updated
                     Endif
```

```
                **Endif**
                ENVD_CUS_SET_CMN[w][z][m] = ENVD_CONF_CUS_SET_CMN[w]
                ENVD_CUS_SET_SPC[y][z][m] = ID_ERR_ENVD_XX [ y ]
                (with XX = F(IDX))
            Endif
    Endif
```

27.14.2 Freeze frame deleting

Description:

Syntax : ACTION_ERRM_EraseFrf (IN <IDX>).

Parameter (in) : IDX Index of failure in 2^{nd} layer memory to store the freeze frame

Parameter (out) : -

Short description :

When the failure IDX is erased from 2^{nd} layer memory, the freeze frame of this failure is also erased by calling the action ACTION_ERRM_EraseFrf (IDX).

Application conditions:

Initialization: -

Recurrence: -

Activation: at Action request

Formula section:

```
ACTION_ERRM_EraseFrf (IDX) :
    ( Erase OBD freeze frame, common freeze frame and all specific freeze frame
            With    u = 1 ... NC_NR_ENVD_OBD
                    v = 1 ... NC_NR_ENVD_CUS_CMN
                    w = 1 ... NC_NR_ENVD_CUS_SET_CMN
                    y = 1 ... NC_NR_ENVD_CUS_SET_SPC
                    z = 1 ... NC_NR_FRF_SET )
    ENVD_OBD[u][m] = 0
    ENVD_CUS_CMN[v][m] = 0
    ENVD_CUS_SET_CMN[w][z][m] = 0
    ENVD_CUS_SET_SPC[y][z][m] = 0
```

27.14.3 Calibratable set of environmental data

Description:

The freeze frame part called FRF_CUS_SET_SPC contains environmental data, which are :

- specific to each failure, not required by law ;
- and calibratable.

This chapter describes “how to define by calibration the contents of FRF_CUS_SET_SPC”.

The calibratable system of environmental data is managed via tables defined as following:

ID_ENVD_FAC :	Configuration table, which describes all possible environmental data, which can be stored in a freeze frame (specific part) ; Affected value defined in Environmental data (Appl. Inc.) module.
ID_ERR_ENVD_XX :	Calibration table, which defines for each failure XX a maximum of NC_NR_ENVD_CUS_SET_SPC environmental data, which should be stored.

Example :

To store the environmental data	TIA and TCO_ST	in case of TCO failure ;
	MAP and AMP	in case of EGR failure ;

First, you should defined the possible environmental data in ID_ENVD_FAC table :

ID_ENVD_FAC[1] = &TIA

ID_ENVD_FAC[2] = &TCO_ST

ID_ENVD_FAC[3] = &MAP

ID_ENVD_FAC[4] = &AMP

Then, you should declare for each failure, the environmental data to stored :

ID_ERR_ENVD_TCO[1] = 1 ; ID_ERR_ENVD_TCO [2] = 2

ID_ERR_ENVD_EGR[1] = 3 ; ID_ERR_ENVD_EGR [2] = 4

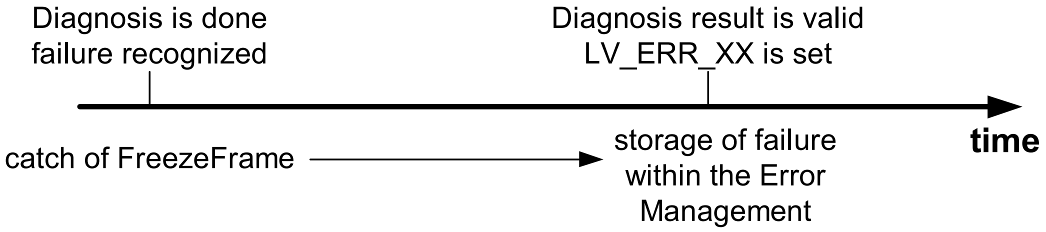
27.14.4 Prestored freeze frame

FUNCTION DESCRIPTION:

General information:

Usually the content for the freeze frame is catched, when the failure is stored within the failure memory. In special cases, when the entry of a failure within the failure memory is delayed, it’s necessary to catch the content for the freeze frame at the same time as the failure information is available.

Example:



Description:

Syntax : ACTION_ERRM_StorePrevFrf (IN <XX>).

Parameter (in) : XX diagnosis instance

Parameter (out) : -

Short description :

If ACTION_ERRM_StorePrevFrf (XX) is called by the diagnosis instance XX, a “prestored” freeze frame for diagnosis instance XX is stored. It’s used as content for the freeze frame, if the failure is entered in the failure memory later.

Application conditions:

Initialization: -

Recurrence: -

Activation: at Action request

Formula section:

ACTION_ERRM_StorePrevFrf (XX) :

with t related to failure XX, see Environmental data Appli.Inc.

ENVD_PREV_OBD[u][t] = ENVD_CONF_OBD[u]

ENVD_PREV_CUS_CMN[v][t] = ENVD_CONF_CUS_CMN[v]

ENVD_PREV_CUS_SET_CMN[w][t] = ENVD_CONF_CUS_SET_CMN[w]

ENVD_PREV_CUS_SET_SPC[y][t] = ID_ERR_ENVD_XX[y] (with XX = F(IDX))

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
ID_ERR_ENVD_XX	NC_NR_ENVD_CUS_SET_SPC	1...FFH	1...255	1	[-]
LDP_1_ID_ERR_ENVD_XX	NC_NR_ENVD_CUS_SET_SPC	0...FFH	0...255	1	[-]
Freeze frame FRF_CUS_SET_SPC : environmental data contents for failure XX					

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_NR_ENVD_OBD	-	1...FFH	1...255	1	[-]
Number of different environment data (in bytes) which are fixed by the law (stored one time)					
NC_NR_ENVD_CUS_CMN	-	1...FFH	1...255	1	[-]
Number of different environment data (in bytes) which are common for all diagnosis instance (stored one time)					
NC_NR_ENVD_CUS_SET_CMN	-	0...FFH	0...255	1	[-]
Number of different environment data (in bytes) which are common for all diagnosis instance (stored many time)					
NC_NR_ENVD_CUS_SET_SPC	-	1...FFH	1...255	1	[-]
number of different environment data (in bytes) which are specific to each diagnosis instance (stored many time)typical value is 4					
NC_NR_FRF_SET	-	1...FFH	1...255	1	[-]
Number of different environment data groups (set) which can be store for one freeze frametypical value is 3 ; the value 0 is forbidden					
NC_ENVD_CUS_CMN_UPD	-	0...1H	0...1	1	[-]
Selection of FRF_CUS_CMN update method					
NC_NR_ENVD_PREV	-	1...FFH	1...255	1	[-]
Number of different failure instances XX using prestored freeze frameIf NC_NR_ENVD_PREV = 0 , then no prestored freeze frame functionality required					
NC_ID_ENVD_FAC	-	1...FFH	1...255	1	[-]
Size of ID_ENVD_FAC table					
ID_ENVD_FAC[255]	-	1...FFH	1...255	1	[-]
Possible stored environmental data in FRF_CUS_SET_SPC					

Configuration data detailed description:

NC_NR_ENVD_CUS_SET_SPC : Typical value is 4.

NC_NR_FRF_SET :	Typical value is 3. The value 0 is forbidden.
NC_ ENVD_CUS_CMN_UPD	0: FRF_CUS_CMN is not updated when failure status changes from "disappeared" to "temporary/confirmed". 1: FRF_CUS_CMN is updated when failure status changes from "disappeared" to "temporary/confirmed".
NC_NR_ENVD_PREV	0: No pre-stored freeze frame functionality required

Size of ID_ENVD_FAC is defined according number of data in the table ; maximum size is 255.

27.15 Environmental data (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ENVD_CONF_OBD[NC_NR_ENVD_OBD]	O	0...FFH	0...255	1	[-]
Freeze frame FRF_OBD : environmental OBD data contents					
ENVD_CONF_CUS_CMN[NC_NR_ENVD_CUS_CMN]	O	0...FFH	0...255	1	[-]
Freeze frame FRF_CUS_CMN : environmental common data contents					
ENVD_CONF_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN]	O	0...FFH	0...255	1	[-]
Freeze frame FRF_CUS_SET_CMN : environmental calibrated data contents					
LOAD_CLC	V/O	0...FFH	0...99.60937	0.390625	[%]
Calculated load					
LOAD_ABSV	V/O	0...FFFFH	0...25700	0.3921569	[%]
Absolute load					
OBD_TPS_SP	V/O	0...FFH	0...99.60937	0.390625	[%]
Commanded throttle position					
OBD_TPS_1	V/O	0...FFH	0...99.60937	0.390625	[%]
Absolute throttle position (sensor 1)					
OBD_TPS_2	V/O	0...FFH	0...99.60937	0.390625	[%]
Absolute throttle position (sensor 2)					
OBD_TPS_REL	V/O	0...FFH	0...99.60937	0.390625	[%]
Relative throttle position					
OBD_FTL	V/O	0...FFH	0...99.60937	0.390625	[%]
Fuel level input					
OBD_PV_1	V/O	0...FFH	0...99.60937	0.390625	[%]
Accelerator pedal position (sensor 1)					
OBD_PV_2	V/O	0...FFH	0...99.60937	0.390625	[%]
Accelerator pedal position (sensor 2)					
OBD_EGR_DIF	V/O	0...FFH	0...99.60937	0.390625	[%]
EGR actual – EGR commanded					

Input data:

OBD_VB	STATE_LS[NC_CBK_EX_NR]	VS	STATE_OBD_SA
TPS_SP_MDL	V_TPS_1	NC_NR_ENVD_CUS_CMN	NC_NR_FRF_SET
NC_NR_ENVD_CUS_SET_CMN	ERR_SYM_XX	DIST_KWP	NC_NR_ENVD_CUS_SET_SPC
V_PVS_2	N_32	MAF	AMP
NC_ID_ENVD_FAC	NC_NR_ENVD_PREV	N_SAE_BYTE_KWP	C_TPS_SP_MAX
OBD_TCO	OBD_LAM_COR[NC_CBK_EX_NR]	OBD_LAM_AD[NC_CBK_EX_NR]	OBD_FUP
MAP_SAE	OBD_N	OBD_IGA_IGC	OBD_TIA
OBD_MAF	OBD_TAM	OBD_LAMB_SP	T_AST_SAE
OBD_VB	OBD_AMP	V_TPS_2	TPS_AV
C_TPS_MAX	FTL	V_PVS_1	OPG_DIF_ACR
NC_ENVD_CUS_CMN_UPD	NC_ID_ENVD_FAC	CPPWM_CPS	OBD_FUP_RNG_H
OPG_SP_ACR			

27.15.1 Definition of OBD freeze frame content and size

Description:

This file defines sizes and contents of Environmental data (freeze frame data) catched when a failure occurs.

Formula section:

Freeze frame part	Caracteristics	Value
FRF_OBD	Number of data (in bytes)	**NC_NR_ENVD_OBD = 38** *“= 38 OBD values, stored once, updated if failure was disappeared and occures again”*
	Contents	ENVD_CONF_OBD[1] = STATE_LS_1 ***(PID 03)*** *Fuel system status bank 1*
		ENVD_CONF_OBD[2] = STATE_LS_2 ***(PID 03)*** *Fuel system status bank 2*
		ENVD_CONF_OBD[3] = LOAD_CLC ***(PID 04)*** *Calculated load*
		ENVD_CONF_OBD[4] = OBD_TCO ***(PID 05)*** *Engine coolant temperature*
		ENVD_CONF_OBD[5] = OBD_LAM_COR_1 ***(PID 06)*** *Short term fuel trim bank 1*
		ENVD_CONF_OBD[6] = OBD_LAM_AD_1 ***(PID 07)*** *Long term fuel trim bank 1*
		ENVD_CONF_OBD[7] = OBD_LAM_COR_2 ***(PID 08)*** *Short term fuel trim bank 2*
		ENVD_CONF_OBD[8] = OBD_LAM_AD_2 ***(PID 09)*** *Long term fuel trim bank 2*
		ENVD_CONF_OBD[9] = OBD_FUP ***(PID 0A)*** *Fuel pressure*
		ENVD_CONF_OBD[10] = MAP_SAE ***(PID 0B)*** *Intake manifold absolute pressure*
		ENVD_CONF_OBD[11] = OBD_N ***(PID 0C)*** *Engine RPM (high byte)*
		ENVD_CONF_OBD[12] = OBD_N ***(PID 0C)*** *Engine RPM (low byte)*
		ENVD_CONF_OBD[13] = VS ***(PID 0D)*** *Vehicle speed*

		ENVD_CONF_OBD[14] = OBD_IGA_IGC ***(PID 0E)*** *Ignition timing advance cylinder 1*
		ENVD_CONF_OBD[15] = OBD_TIA ***(PID 0F)*** *Temperature intake air*
		ENVD_CONF_OBD[16] = OBD_MAF ***(PID 10)*** *Air flow rate mass air flow sensor (high byte)*
		ENVD_CONF_OBD[17] = OBD_MAF ***(PID 10)*** *Air flow rate mass air flow sensor (low byte)*
		ENVD_CONF_OBD[18] = OBD_TPS_1 ***(PID 11)*** *Absolute Throttle position (signal 1)*
		ENVD_CONF_OBD[19] = STATE_OBD_SA ***(PID 12)*** *Commanded secondary air status*
		ENVD_CONF_OBD[20] = T_AST_SAE ***(PID 1F)*** *Time since engine start (high byte)*
		ENVD_CONF_OBD[21] = T_AST_SAE ***(PID 1F****) Time since engine start (low byte)*
		ENVD_CONF_OBD[22] = OBD_FUP_RNG_H ***(PID 23)*** *Fuel Pressure High (high byte)*
		ENVD_CONF_OBD[23] = OBD_FUP_RNG_H ***(PID 23)*** *Fuel Pressure High (low byte)*
		ENVD_CONF_OBD[24] = CPPWM_CPS ***(PID 2E)*** *Commanded Evaporative Purge*
		ENVD_CONF_OBD[25] = OBD_FTL ***(PID 2F)*** *Fuel level input*
		ENVD_CONF_OBD[26] = OBD_AMP ***(PID 33)*** *Barometric pressure*
		ENVD_CONF_OBD[27] = OBD_VB ***(PID 42)*** *Control module voltage (high byte)*
		ENVD_CONF_OBD[28] = OBD_VB ***(PID 42)*** *Control module voltage (low byte)*
		ENVD_CONF_OBD[29] = LOAD_ABSV ***(PID 43)*** *Absolute load value (high byte)*
		ENVD_CONF_OBD[30] = LOAD_ABSV ***(PID 43)*** *Absolute load value (low byte)*

		ENVD_CONF_OBD[31] = OBD_LAMB_SP ***(PID 44)** Commanded equivalence ratio (high byte)*
		ENVD_CONF_OBD[32] = OBD_LAMP_SP ***(PID 44)** Commanded equivalence ratio (low byte)*
		ENVD_CONF_OBD[33] = OBD_TPS_REL ***(PID 45)** Relative Throttle position*
		ENVD_CONF_OBD[34] = OBD_TAM ***(PID 46)** Ambient air temperature*
		ENVD_CONF_OBD[35] = OBD_TPS_2 ***(PID 47)** Absolute throttle position (sensor 2)*
		ENVD_CONF_OBD[36] = OBD_PV_1 ***(PID 49)** Acceleration pedal position (sensor 1)*
		ENVD_CONF_OBD[37] = OBD_PV_2 ***(PID 4A)** Acceleration pedal position (sensor 2)*
		ENVD_CONF_OBD[38] = OBD_TPS_SP ***(PID 4C)** Commanded throttle position*

Freeze frame part	**Caracteristics**	**Value**
FRF_CUS_CMN	Number of data (in bytes)	**NC_NR_ENVD_CUS_CMN = 1** *"= One common value for each diagnosis, stored at same event like FRF_OBD"*
	Contents	ENVD_CONF_CUS_CMN[1] = N_SAE_BYTE_KWP *Engine Speed (SAE resolution)*
"Set of Freeze Frame"	Number of set	**NC_NR_FRF_SET = 3** *"= First, second, last occurrence"*
FRF_CUS_SET _CMN	Number of common data per set (in bytes)	**NC_NR_ENVD_CUS_SET_CMN = 3** *"= Three common values for all diagnosis, stored at every occurrence"*
	Contents	ENVD_CONF_CUS_SET_CMN[1] = ERR_SYM_XX *Detected symptom*
		ENVD_CONF_CUS_SET_CMN[2] = DIST_KWP *Calculated Km counter [8km] (high-byte)*
		ENVD_CONF_CUS_SET_CMN[3] = DIST_KWP *Calculated Km counter [8km] (low-byte)*

FRF_CUS_SET_SPC	Number of specific data per set (in bytes)	**NC_NR_ENVD_CUS_SET_SPC = 4** *"=4 calibratable environmental data per failure"*
	Contents	Define by calibration ID_ERR_ENVD_XX (see Environmental data module) Possible stored environmental data : NC_ID_ENVD_FAC [1] = see DDLI list NC_ID_ENVD_FAC [2] = see DDLI list ... NC_ID_ENVD_FAC [255] = see DDLI list

27.15.2 Configuration of update of FRF_CUS_CMN

FUNCTION DESCRIPTION:

Description:

This file defines the update of the FRF_CUS_CMN in case of failure status change from disappeared to temporary/confirmed. To follow specific customer requirements, this file is a template to be fulfilled by project team.

All data in gray should be completed by the project team.

Update of FRF_CUS_CMN in case of failure status change from disappeared to temporary/confirmed.	NC _ENVD_CUS_CMN_UPD = **1** 0: no update 1: update

27.15.3 Configuration for prestored freeze frame

FUNCTION DESCRIPTION:

Description:

This file defines the usage of the prestored freeze frame functionality. To follow specific customer requirements, this file is a template to be fulfilled by project team.

All data in gray should be completed by the project team.

Number of diagnosis instances XX using prestored freeze frame	NC_NR_ENVD_PREV = 1
Diagnosis instance XX	**prestored freeze frame number (t)**

ROUGH_LEAK	1

Remark: The usage of the prestored freeze frame functionality must be limited for special diagnosis functions, because of memory consumption.

27.15.4 Definition of OBD – Systemvariables for MODE 1 / 2

General information:

The variables are used for OBD powertrain diagnostic data (MODE1) and OBD Freeze-frame calculation (MODE 2).

Application conditions:

Initialisation: 0 at reset

Recurrence: 100ms

Activation: at every engine operating state

Formula section:

All calculation done in physical way

LOAD_CLC = IP_LOAD_CLC__N_32__MAF * IP_LOAD_CLC_AMP__AMP * 100%
LOAD_ABSV = (MAF / C_MAF_REF_OBD) * 100%
OBD_TPS_SP = (TPS_SP_MDL / C_TPS_SP_MAX) * 100%
OBD_TPS_1 = (V_TPS_1 / 5V) * 100%
OBD_TPS_2 = ((5V – V_TPS_2) / 5V) * 100%
OBD_TPS_REL = (TPS_AV / C_TPS_MAX) 100%
OBD_FTL = (FTL / C_FTL_REF_OBD) * 100%
OBD_PV_1 = (V_PVS_1 / 5V) * 100%
OBD_PV_2 = (V_PVS_2 / 2,5V) * 100%
OBD_EGR_DIF = I OPG_DIF_ACR / OPG_SP_ACR I * 100%

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_LOAD_CLC_AMP__AMP	6	0...FFH	0...1.99218	0.0078125	[-]
LDP_AMP_IP_LOAD_CLC_AMP	6	0...FFFFH	0...5434	8.29E-02	[hPa]
Atmospheric pressure ratio					
IP_LOAD_CLC__N_32__MAF	8*8	0...FFH	0...0.99609	3.91E-03	[-]
LDP_N_32_IP_LOAD_CLC	8	0...FFH	0...8160	32	[rpm]
LDP_MAF_IP_LOAD_CLC	8	0...FFFFH	0...1389	2.12E-02	[mg/stk]
Airflow ratio					
C_FTL_REF_OBD	1	0...7EH	0...126	1	[l]
Maximum poosible FTL in the vehicle					
C_MAF_REF_OBD	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Maximum air mass concerning cylinder displacement					

27.16 Statistical data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_DTC_CLR	V/O/S	0...FFFFH	0...65535	1	[min]
Time since diagnostic trouble codes cleared					
DIST_ACT_MIL	V/O/S	0...FFFFH	0...65535	1	[km]
Global distance traveled while the MIL is illuminated					
DIST_REL_ACT_MIL	- /S	0...FFFFFFFFH	0...4294967295	1	[km]
Relative distance to calculate distance traveled while the MIL is illuminated					
CTR_ACT_MIL	- /S	0...FFH	0...255	1	[-]
Counter for warm up cycle with MIL Off					
T_ACT_MIL	V/O/S	0...FFFFH	0...65535	1	[min]
Minutes run by the engine while MIL activated					
T_ACT_MIL_60	V/S	0...3CH	0...60	1	[s]
Seconds run by the engine while MIL activated for T_ACT_MIL calculation (0 ... 59)					
DIST_DTC_CLR	V/O/S	0...FFFFH	0...65535	1	[km]
Distance since diagnostic trouble codes cleared					
DIST_REL_DTC_CLR	V/S	0...FFFFFFFFH	0...4294967295	1	[km]
Distance at following event : diagnostic trouble codes cleared					
CTR_WUP_DTC_CLR	V/O/S	0...FFH	0...255	1	[-]
Number of warm-ups since diagnostic trouble codes cleared					
STATE_ENA_OBD	V/O	0...FFFFH	0...65535	1	[-]
Monitor enable status for the current driving cycle					
STATE_CMPL_OBD	V/O	0...FFFFH	0...65535	1	[-]
Monitor completion status for the current driving cycle					

Input data:

DIST	C_CTR_MAX_WUP_CYC	STATE_MIL	LV_DC
C_STATE_READY_OBD_1	C_STATE_READY_OBD_2	LV_END_DIAG_XX	WAL_CONF_XX
LV_WUP_CYC			

Import actions:

ACTION_ERRM_MonitorEnableStatus (INOUT < MonitorEnableStatus>)
This action allows to read the MonitorEnableStatus information

FUNCTION DESCRIPTION:

General information:

EOBD/CARB official requirements include a list of statistical data related to Error Management. These statistical data are counters and timers related to Error Management events. You can find below, in the following chapters, a description of all these requirements.

27.16.1 Time since diagnostic trouble codes cleared

Description:

Purpose of this strategy is to compute the time accumulated by the vehicle since diagnostic trouble codes cleared (referenced in ISO15031 as Modes 01h & 02h PID4Eh). It is simply an indication for I/M (Inspection/Maintenance), of the last time an external test equipment was used to clear DTCs.

Application conditions:

Initialization: on non-volatil memory reset or lost
or on failure erase service received
T_DTC_CLR = 0

Recurrence : 1 minute ±1 second, after event : on “failure erase” service received

Activation : -

Deactivation: -

Formula section :

```
{ T_DTC_CLR is updated at required recurrence }
{ T_DTC_CLR shall saturate to its maximal value 65535 }

If      T_DTC_CLR < 65535
Then
        T_DTC_CLR = T_DTC_CLR + 1
Endif
```

27.16.2 Distance counters since MIL activation and Minutes run by the engine with MIL activated

Description:

According to ISO/DIS 15031-5.8 (Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 5: Emissions-related diagnostic services), ECU must be able to indicate :

- the distance travelled while the MIL is activated
- the number of minutes run by the engine while MIL is activated

Distance Travelled While MIL is Activated

The purpose is to compute the covered distance while the MIL is illuminated (MIL blinking is considered as a MIL illumination). This mileage (DIST_ACT_MIL) is a global distance :

- which is not linked with any failure and is independent from the Pre-drive check
- which is dependent from the global inhibition MIL boolean
- which is dependent of the MIL request
- which is readable with Scan-Tool with kilometer as unit

Conditions for “Distance travelled” counter :

- reset to 0 when MIL state changes from deactivated to activated by this ECU
- accumulate counts in km if MIL is activated (ON)
- do not change value while MIL is not activated (OFF)
- reset to 0 if diagnostic information is cleared either by service 04h or 40 warm-up cycles without MIL activated
- do not wrap to 0 if value is FFFFh

Minutes run by the engine while MIL activated

The purpose is to compute the time accumulated while MIL is activated (Modes 01h & 02h PID4Dh). This duration shall be cleared when failure memory is erased and after 40 warm-up cycles.

Conditions for “Minutes run by the engine while MIL activated” counter :

- reset to 0 when MIL state changes from deactivated to activated by this ECU
- accumulate counts in minutes if MIL is activated (ON)
- do not change value while MIL is not activated (OFF)
- reset to 0 if diagnostic information is cleared either by service 04h or 40 warm-up cycles without MIL activated
- do not wrap to 0000h if value is FFFFh

Application conditions:

Initialization: On saved ram lost

or STATE_MIL transition OFF -> MIL_FLL or OFF -> ON

```
            or      failure erase service received
                    DIST_REL_ACT_MIL = DIST
                    DIST_ACT_MIL = 0
                    CTR_ACT_MIL = C_CTR_MAX_WUP_CYC
                    T_ACT_MIL = 0
                    T_ACT_MIL_60 = 0
```

Recurrence: upon LV_DC 1→0 transition and 1 s

Activation: all engine states

<u>Formula section:</u>

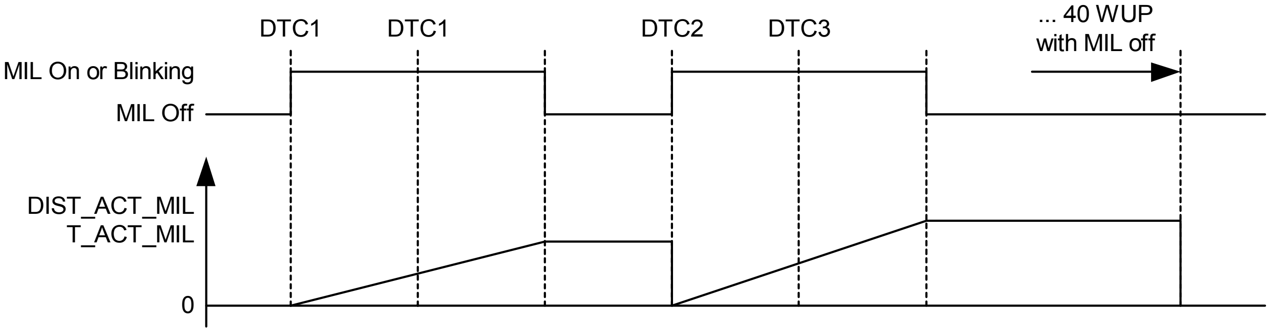
<u>Each 1s:</u>

```
If(1)   STATE_MIL = MIL_FLL
        or
        STATE_MIL = ON
Then(1)
        If(2)   ( DIST – DIST_REL_ACT_MIL ) < 65535
        Then(2)
                DIST_ACT_MIL = DIST – DIST_REL_ACT_MIL
        Else(2)
                DIST_ACT_MIL = 65535
        Endif(2)
        T_ACT_MIL_60 = T_ACT_MIL_60 + 1 s
        If(2)   (T_ACT_MIL_60 = 60)
        Then(2)
                T_ACT_MIL_60 = 0
                If(3)   T_ACT_MIL < 65535
                Then(3)
                        T_ACT_MIL = T_ACT_MIL + 1 min
                Else(3)
                        T_ACT_MIL = 65535
                Endif(3)
        Endif(2)
Endif(1)
```

<u>At LV_DC 1→0 transition :</u>

```
If(1)   LV_WUP_CYC = 1
```

```
        and
        STATE_MIL was OFF during all the warm-up cycle
        and
        CTR_ACT_MIL > 0
    Then(1)
        CTR_ACT_MIL = CTR_ACT_MIL –1
        If(2)   CTR_ACT_MIL = 0
        Then(2)
                DIST_REL_ACT_MIL = DIST
                DIST_ACT_MIL = 0
                T_ACT_MIL = 0
                T_ACT_MIL_60 = 0
        Endif(2)
    Endif(1)
```



27.16.3 Distance since diagnostic trouble codes cleared

Description :

Purpose of this strategy is to compute the distance accumulated by the vehicle since diagnostic trouble codes cleared (referenced in ISO15031 as mode01h&02hPID31h). It is simply an indication for Inspection/Maintenance, of the last time an external test equipment was used to clear DTCs.

Application conditions:

Initialization: on non-volatil memory reset or lost

or

on failure erase service received
DIST_DTC_CLR = 0
DIST_REL_DTC_CLR = DIST

Recurrence : 1 second

Activation : -

Deactivation: -

Formula section :

{ DIST_DTC_CLR is updated regularly regarding DIST }
{ DIST_DTC_CLR shall saturate to its maximal value 65535 km}

```
If      DIST - DIST_REL_DTC_CLR < 65535
Then
        DIST_DTC_CLR = DIST - DIST_REL_DTC_CLR
Else
        DIST_DTC_CLR = 65535
Endif
```

27.16.4 Number of warm-ups since diagnostic trouble codes cleared

Description:

Purpose of this strategy is to compute number of warm-ups since diagnostic trouble codes cleared (referenced in ISO15031 as Modes 01h & 02h PID30h). It’s just an indication for Inspection/Maintenance, to determine the last time an external test equipment was used to clear DTCs.

Application conditions:

Activation :	On LV_WUP_CYC 0 → 1 transition
Initialization:	On non-volatil memory reset or lost **or** on failure erase service received CTR_WUP_DTC_CLR=0
Recurrence :	none, executed single time on activation.

Formula section:

```
{ on detection of a warm-up cycle CTR_WUP_DTC_CLR is incremented }
{ CTR_WUP_DTC_CLR shall saturate to its maximal value 255 }

If      CTR_WUP_DTC_CLR < 255
Then
        CTR_WUP_DTC_CLR = CTR_WUP_DTC_CLR + 1
Endif
```

27.16.5 Monitor enable/completion status for the current driving cycle

General information:

Purpose of this strategy is to get the follwoing informations :
- if a diagnosis (a monitor) is disabled for rest of this monitoring cycle or not (referenced in ISO15031 as Modes 01h & 02h PID41h).
- if a diagnosis (a monitor) is completed for the current monitoring cycle or not (referenced in ISO15031 as Modes 01h & 02h PID41h).

Each concerned diagnosis is part of a generic list : Misfire, Fuel system, Comprehensive component, Catalyst, Heated catalyst, Evaporative system, Secondary air system, A/C system refrigerant, Oxygen sensor, Oxygen sensor heater, EGR system monitoring.

MonitorEnableStatus :
Each bit of STATE_ENA_OBD variable contains status for each predefined diagnosis as defined below :
Enable status of a diagnosis this monitoring cycle:
- NO (STATE_ENA_OBD[n]=0) means disabled for rest of this monitoring cycle or not supported in C_STATE_READY_OBD_1 & C_STATE_READY_OBD_2
- YES (STATE_ENA_OBD[n]=1) means enabled for this monitoring cycle.

MonitorCompletionStatus :
Each bit of STATE_CMPL_OBD variable contains status for each predefined diagnosis as defined below :
Completion status of a diagnosis this monitoring cycle:
- YES (STATE_CMPL_OBD[n]=0) means monitor complete this monitoring cycle, or not supported in C_STATE_READY_OBD_1 & C_STATE_READY_OBD_2
- NO (STATE_CMPL_OBD[n]=1) means monitor not complete this monitoring cycle.

Bit 0 to 7 of STATE_ENA_OBD[0..7] :

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ └─> bit0: Misfire monitoring
 │ │ │ │ │ │ └───> bit1: Fuel system monitoring
 │ │ │ │ │ └─────> bit2: Comprehensive component monitoring
 │ │ │ │ └───────> bit3: reserved
 │ │ │ └─────────> bit4: reserved
 │ │ └───────────> bit5: reserved
 │ └─────────────> bit6: reserved
 └───────────────> bit7: reserved
```

Bit 8 to 15 of STATE_ENA_OBD[8..15] :

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ └─> bit8: Catalyst monitoring
 │ │ │ │ │ │ └───> bit9: Heated catalyst monitoring
 │ │ │ │ │ └─────> bit10: Evaporative system monitoring
 │ │ │ │ └───────> bit11: Secondary air system monitoring
 │ │ │ └─────────> bit12: A/C system refrigerant monitoring
 │ │ └───────────> bit13: Oxygen sensor monitoring
 │ └─────────────> bit14: Oxygen sensor heater monitoring
 └───────────────> bit15: EGR system monitoring monitoring
```

Bit 0 to 7 of STATE_CMPL_OBD[0..7] :

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ └─> bit0: Misfire monitoring
 │ │ │ │ │ │ └───> bit1: Fuel system monitoring
 │ │ │ │ │ └─────> bit2: Comprehensive component monitoring
 │ │ │ │ └───────> bit3: reserved
 │ │ │ └─────────> bit4: reserved
 │ │ └───────────> bit5: reserved
 │ └─────────────> bit6: reserved
 └───────────────> bit7: reserved
```

Bit 8 to 15 of STATE_CMPL_OBD[8..15] :

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ └─> bit8: Catalyst monitoring
 │ │ │ │ │ │ └───> bit9: Heated catalyst monitoring
 │ │ │ │ │ └─────> bit10: Evaporative system monitoring
 │ │ │ │ └───────> bit11: Secondary air system monitoring
 │ │ │ └─────────> bit12: A/C system refrigerant monitoring
 │ │ └───────────> bit13: Oxygen sensor monitoring
 │ └─────────────> bit14: Oxygen sensor heater monitoring
 └───────────────> bit15: EGR system monitoring monitoring
```

Application conditions:

Initialization: on LV_DC 1 → 0 transition
STATE_ENA_OBD = 0
STATE_CMPL_OBD = 0

Recurrence : 10 seconds

Activation : -

Deactivation: -

Formula section :

```
{

{     Action call definition shall define strategy related to each supported diagnosis      }
{     each bit of STATE_ENA_OBD shall be computed according diagnosis strategy      }


{     STATE_ENA_OBD computation     }
{ STATE_ENA_OBD is the result returned by ACTION_ERRM_MonitorEnableStatus action      }
{     Please refer to ACTION_ERRM_MonitorEnableStatus definition in                 }
{     the Application Incidence file                                                }

ACTION_ERRM_MonitorEnableStatus(INOUT<STATE_ENA_OBD>,
SYNCHRONIZATION<CALL> )



{     STATE_CMPL_OBD computation            }
{     Each bit of STATE_CMPL_OBD shall be computed according end diagnosis flags       }

STATE_CMPL_OBD[0..7]        = C_STATE_READY_OBD_1
STATE_CMPL_OBD[8..15]       = C_STATE_READY_OBD_2

Misfire monitoring status bit :
    If      LV_END_DIAG_XX=1 for all diagnosis XX defined below
            Take into account all LV_END_DIAG_XX which fulfill the following condition:
                    this XX failure has the status CARB_MIS (see tables of failures) and
                    the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)

    Then    Bit 0 of STATE_CMPL_OBD = 0
    Endif
```

Fuel system monitoring status bit :

```
If      LV_END_DIAG_XX=1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_FSD (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 1 of STATE_CMPL_OBD = 0
Endif
```

Comprehensive component monitoring status bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_CC (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 2 of STATE_CMPL_OBD = 0
Endif
```

Catalyst monitoring bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_CAT (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 8 of STATE_CMPL_OBD = 0
Endif
```

Heated catalyst bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_HC (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 9 of STATE_CMPL_OBD = 0
Endif
```

Evaporative system monitoring bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_EVAP(see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 10 of STATE_CMPL_OBD = 0
Endif
```

Secondary air system monitoring bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_SA (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 11 of STATE_CMPL_OBD = 0
Endif
```

A/C system refrigerant monitoring bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_AC (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 12 of STATE_CMPL_OBD = 0
Endif
```

Oxygen sensor monitoring bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_LS (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 13 of STATE_CMPL_OBD = 0
Endif
```

Oxygen sensor heater monitoring bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_LSH (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 14 of STATE_CMPL_OBD = 0
Endif
```

EGR system monitoring bit :

```
If      LV_END_DIAG_XX = 1 for all diagnosis XX defined below
        Take into account all LV_END_DIAG_XX which fulfill the following condition:
                this XX failure has the status CARB_EGR (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)

Then    Bit 15 of STATE_CMPL_OBD = 0
Endif
```

27.17 Statistical data (App. Inc.)

Output data:

T_ACT_MIL_SAE_60	O/V/S	0...3CH	0...60	1	[s]
Seconds run by the engine while MIL activated for T_ACT_MIL_SAE calculation (0 ... 59)					
T_ACT_MIL_SAE	O/V/S	0...FFFFH	0...65535	1	[min]
PID 4D Time run by the engine while MIL activated					

Input data:

LV_ES	CTR_ACT_MIL	STATE_MIL	LV_DC
LV_WUP_CYC	C_STATE_READY_OBD_1	LV_INH_FSD_STOP_OIL	C_AMP_MIN_CAT
C_AMP_MIN_FSD	C_STATE_READY_OBD_2	AMP	C_AMP_MIN_MIS

27.17.1 Monitor enable status for the current driving cycle (PID41)

Export actions:

ACTION_ERRM_MonitorEnableStatus (OUT < MonitorEnableStatus>)
This action allows to read the MonitorEnableStatus information

FUNCTION DESCRIPTION:

General information:

Purpose of this strategy is to get the follwoing informations :

- if a diagnosis (a monitor) is disabled for rest of this monitoring cycle or not (referenced as Mode01h&02hPID41h ISO15031).

Each concerned diagnosis is part of a generic list : Misfire, Fuel system, Comprehensive component, Catalyst, Heated catalyst, Evaporative system, Secondary air system, A/C system refrigerant, Oxygen sensor, Oxygen sensor heater, EGR system monitoring.

MonitorEnableStatus :
Each bit of MonitorEnableStatus variable contains status for each predefined diagnosis as defined below :
Enable status of a diagnosis this monitoring cycle: NO (MonitorEnableStatus[n]=0) means disabled for rest of this monitoring cycle, YES (MonitorEnableStatus[n]=1) means enabled for this monitoring cycle.

Bit 0 to 7 of MonitorEnableStatus[0..7] :

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ └─> bit0: Misfire monitoring
 │ │ │ │ │ │ └───> bit1: Fuel system monitoring
 │ │ │ │ │ └─────> bit2: Comprehensive component monitoring
 │ │ │ │ └───────> bit3: reserved
 │ │ │ └─────────> bit4: reserved
 │ │ └───────────> bit5: reserved
 │ └─────────────> bit6: reserved
 └───────────────> bit7: reserved
```

Bit 8 to 15 of MonitorEnableStatus[8..15] :

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ └─> bit8: Catalyst monitoring
 │ │ │ │ │ │ └───> bit9: Heated catalyst monitoring
 │ │ │ │ │ └─────> bit10: Evaporative system monitoring
 │ │ │ │ └───────> bit11: Secondary air system monitoring
 │ │ │ └─────────> bit12: A/C system refrigerant monitoring
 │ │ └───────────> bit13: Oxygen sensor monitoring
 │ └─────────────> bit14: Oxygen sensor heater monitoring
 └───────────────> bit15: EGR system monitoring monitoring
```

Description :

Syntax:	ACTION_ERRM_MonitorEnableStatus (OUT < MonitorEnableStatus >)
Parameter (in):	No parameter
Parameter (out):	MonitorEnableStatus with the same format as MonitorEnableStatus
Short Description:	This API calculates and returns the monitor enable status result information in the software structure.

Application conditions:

Activation:	at ACTION request
Initialization:	all bits 0
Recurrence:	-

Formula section:

Misfire monitoring status bit :

```
if      AMP > C_AMP_MIN_MIS
then    Bit 0 of MonitorEnableStatus = 1
else    Bit 0 of MonitorEnableStatus = 0
endif
```

Fuel system monitoring status bit :

```
if      AMP > C_AMP_MIN_FSD                     and
```

```
          LV_INH_FSD_STOP_OIL = 0
then      Bit 1 of MonitorEnableStatus = 1
else      Bit 1 of MonitorEnableStatus = 0
endif
```

Comprehensive component monitoring status:

```
          Bit 2 of MonitorEnableStatus = 1
```

Catalyst monitoring bit :

```
if        AMP > C_AMP_MIN_CAT
then      Bit 8 of MonitorEnableStatus = 1
else      Bit 8 of MonitorEnableStatus = 0
endif
```

Heated catalyst monitoring status bit6 :

```
          Bit 9 of MonitorEnableStatus = 0
```

Evaporative system monitoring bit :

```
          Bit 10 of MonitorEnableStatus = 1
```

Secondary air system monitoring bit :

```
          Bit 11 of MonitorEnableStatus = 0
```

A/C system refrigerant monitoring bit :

```
          Bit 12 of MonitorEnableStatus = 0
```

Oxygen sensor monitoring bit :

```
          Bit 13 of MonitorEnableStatus = 1
```

Oxygen sensor heater monitoring bit :

```
          Bit 14 of MonitorEnableStatus = 1
```

EGR system monitoring bit :

```
          Bit 14 of MonitorEnableStatus = 0
```

```
MonitorEnableStatus [0..7] = MonitorEnableStatus [0..7] and C_STATE_READY_OBD_1
MonitorEnableStatus [8..15] = MonitorEnableStatus 8..15] and C_STATE_READY_OBD_2
```

27.17.2 Minutes run by the engine with MIL activated

Description:

According to ISO/DIS 15031-5.8 (Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 5: Emissions-related diagnostic services), ECU must be able to indicate :

- the number of minutes run by the engine while MIL is activated

Minutes run by the engine while MIL activated

The purpose is to compute the time accumulated while MIL is activated (Modes 01h & 02h PID4Dh). This duration shall be cleared when failure memory is erased and after 40 warm-up cycles.

Conditions for “Minutes run by the engine while MIL activated” counter :

- reset to 0 when MIL state changes from deactivated to activated by this ECU
- accumulate counts in minutes if MIL is activated (ON)
- do not change value while MIL is not activated (OFF)
- reset to 0 if diagnostic information is cleared either by service 04h or 40 warm-up cycles without MIL activated
- do not wrap to 0000h if value is FFFFh

Application conditions:

Initialization: On saved ram lost

or STATE_MIL transition OFF -> MIL_FLL or OFF -> ON

or failure erase service received

T_ACT_MIL_SAE = 0

T_ACT_MIL_SAE_60 = 0

Recurrence: upon LV_DC 1→0 transition and 1 s

Activation: LV_ES=0

Formula section:

Each 1s:

```
If(1)   STATE_MIL = MIL_FLL
        or
        STATE_MIL = ON
Then(1)
        T_ACT_MIL_SAE_60 = T_ACT_MIL_SAE_60 + 1 s
        If(2)   (T_ACT_MIL_SAE_60 = 60)
        Then(2)
                T_ACT_MIL_SAE_60 = 0
                If(3)   T_ACT_MIL_SAE < 65535
                Then(3)
                        T_ACT_MIL_SAE = T_ACT_MIL_SAE + 1 min
                Else(3)
                        T_ACT_MIL_SAE = 65535
                Endif(3)
        Endif(2)
Endif(1)
```

At LV_DC 1→0 transition : calculates after CTR_ACT_MIL calculation

```
If(2)   CTR_ACT_MIL = 0
Then(2)
    T_ACT_MIL_SAE = 0
    T_ACT_MIL_SAE_60 = 0
Endif(2)
```

27.18 Rate-Based Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_COMP_RBM[NC_NR_DIAG_RBM]	V/O/S	0...FFFFH	0...65535	1	[-]
Monitor individual numerator: Number of DC with monitor done since first power up					
CTR_CDN_RBM[NC_NR_DIAG_RBM]	V/O/S	0...FFFFH	0...65535	1	[-]
Monitor individual denominator: Number of DC with convenient vehicle operation for monitor since first power up					
STATE_CTR_RBM[NC_NR_DIAG_RBM]	V	0...3H	0...3	1	[-]
Individual numerator/denominator calculation status					
CTR_IGK_CYC_RBM	V/O/S	0...FFFFH	0...65535	1	[-]
Ignition cycle counter: Number of DC since first ECU power up					
LV_IGK_CYC_RBM	V	0...1H	0...1	1	[-]
Boolean to indicate, that incrementation of CTR_IGK_CYC_RBM done this DC					
CTR_CDN_OBD_RBM	V/O/S	0...FFFFH	0...65535	1	[-]
General denominator: Number of DC with valid standardized vehicle operations since first ECU power up					
LV_DC_RBM	V/O	0...1H	0...1	1	[-]
DC including standardized vehicle operations for Rate-Based Monitoring					
STATE_DC_RBM	V	0...FFH	0...255	1	[-]
Status information about all conditions necessary to set LV_DC_RBM					
T_AST_RBM	V	0...FFFFH	0...6553.5	0.1	[s]
Cumulated time since engine start with Rate-Based Monitoring conditions					
T_VS_RBM	V	0...FFFFH	0...6553.5	0.1	[s]
Cumulated vehicle operation with vehicle speed >= C_T_VS_RBM with Rate-Based Monitoring conditions					
T_IS_RBM	V	0...FFFFH	0...6553.5	0.1	[s]
Continuous vehicle operation in idle >= C_T_IS_RBM with Rate-Based Monitoring conditions					

Output data detailed description:

STATE_CTR_RBM[NC_NR_DIAG_RBM] : Information data for CTR_COMP_RBM[NC_NR_DIAG_RBM] and CTR_CDN_RBM[NC_NR_DIAG_RBM] calculations.

bit 0: incrementation of monitor individual numerator done in this DC
bit 1: incrementation of monitor individual denominator done in this DC

Input data:

LV_DC	T_AST	STATE_RBM[NC_NR_DIAG_RBM]	LV_INH_DC_RBM
NC_NR_DIAG_RBM	T_IS	TAM	AMP
LV_IS	LV_ES	VS	PV

FUNCTION DESCRIPTION:

General information:

To gain more information about the monitoring performance under real world conditions, CARB required to implement software algorithms to track and report in-use performance for the following monitors in a standardized format: Catalyst, Oxygen sensor, Evaporating system, EGR system, VVT system and Secondary air system.

For this kind of statistic different counters are necessary:

- Ignition cycle counter

  Counter that indicates the number of ignition cycles a vehicle has experienced
- General denominator

  Measures the number of times a vehicle is operated under standardized conditions
- Monitor individual numerator

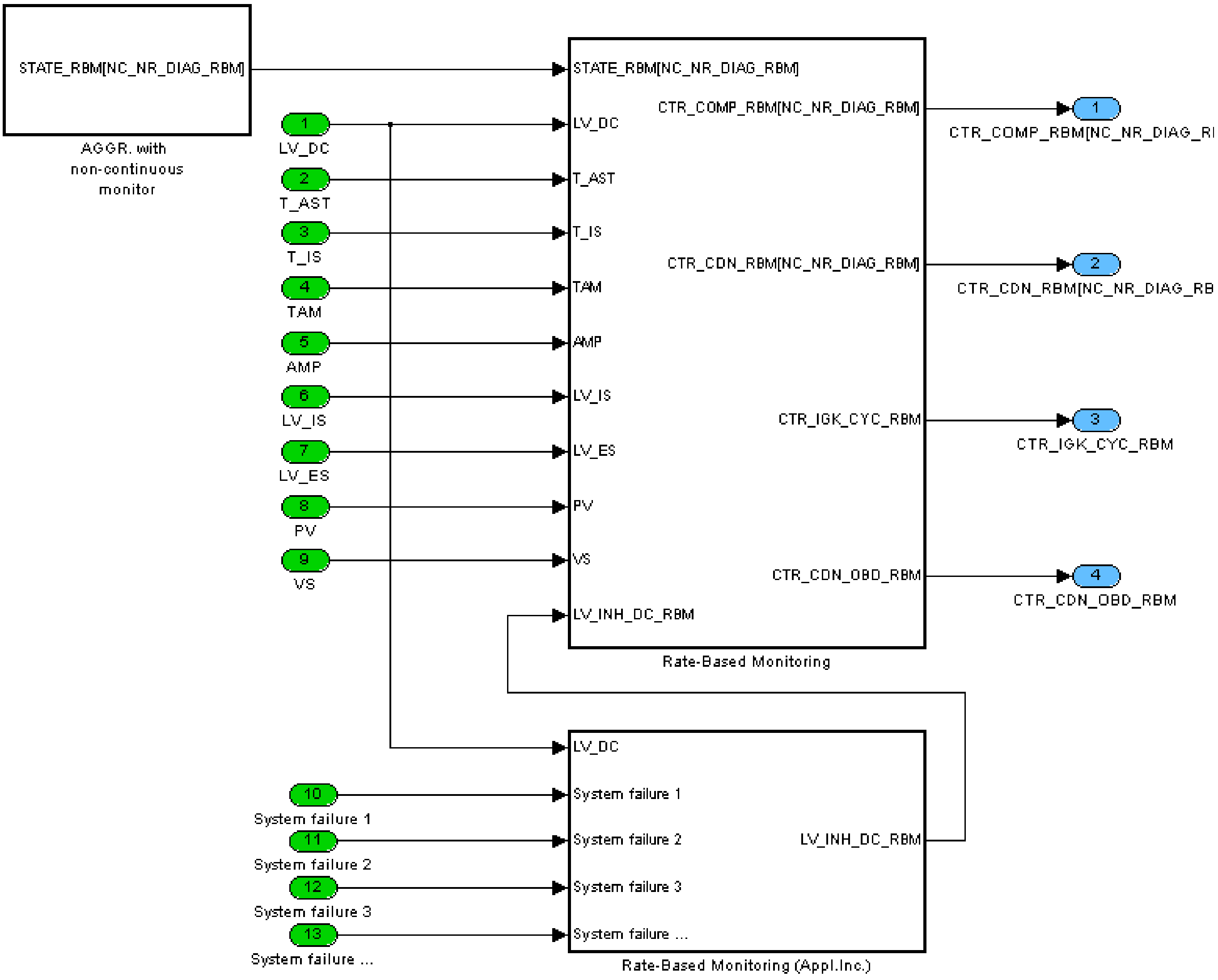
  Measures the number of driving cycles in which the monitor was done
- Monitor individual denominator

  Counter that indicates the number of times a vehicle is operated under monitor individual conditions.

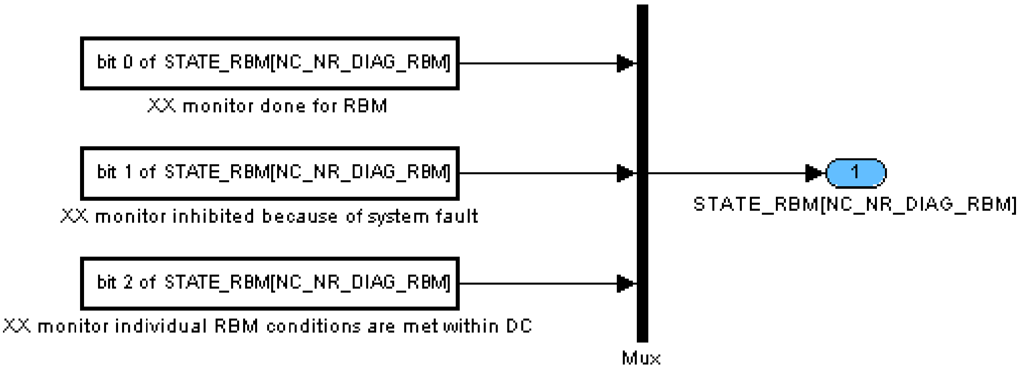
All these counters are set to zero only when a non-volatile memory reset occurs (e.g. reprogramming). They will not be initialized with zero on any other circumstances including when a scan-tool command to clear fault codes is received.

Signal flow diagram:

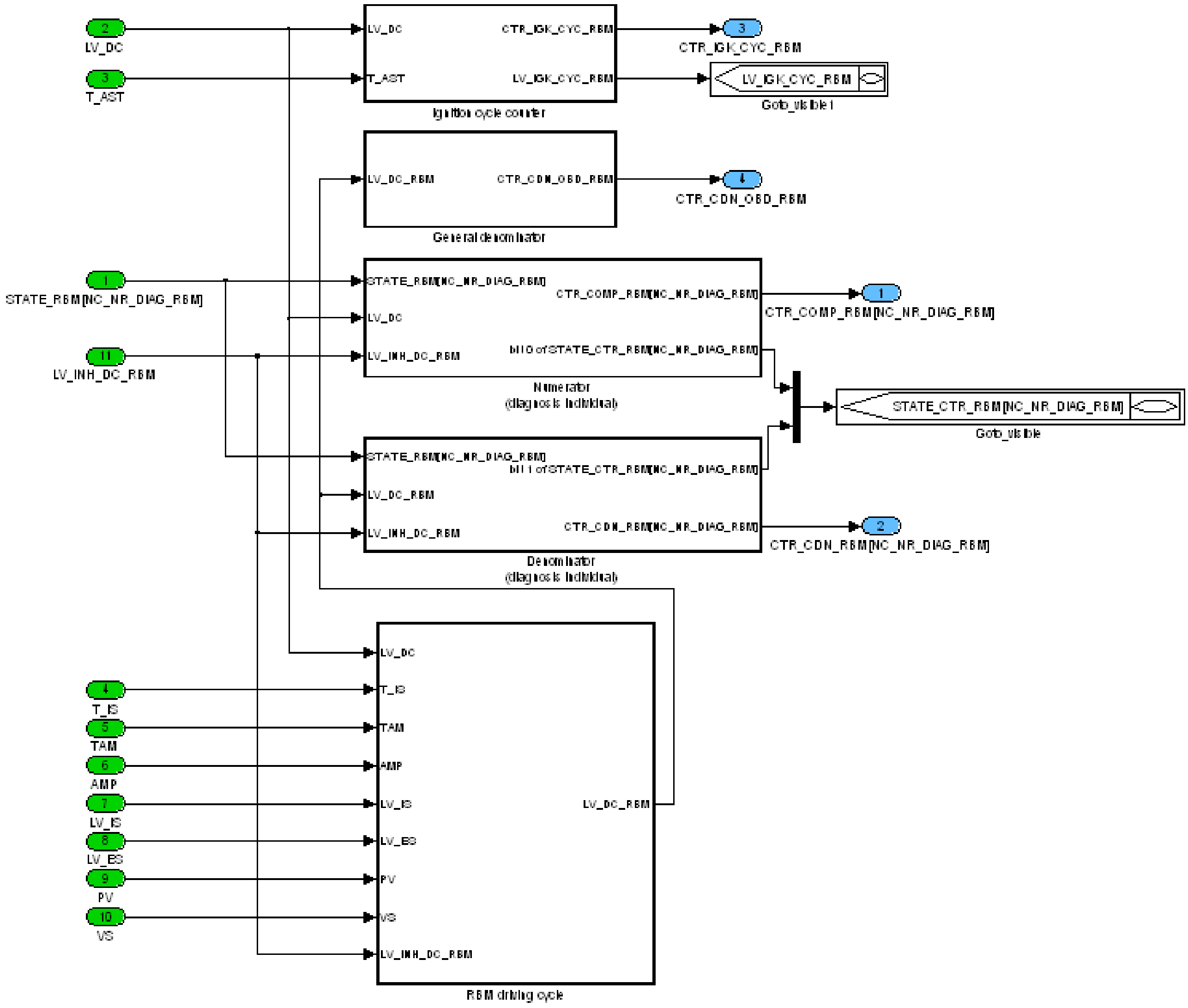
Overview :



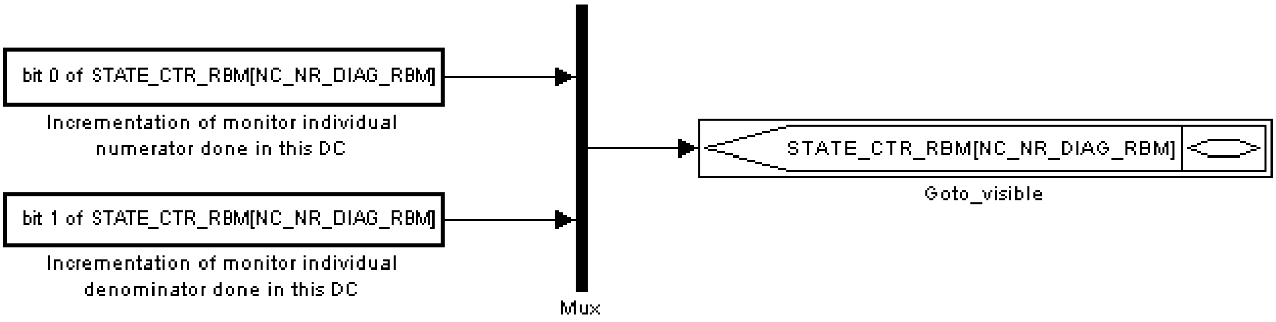
STATE_RBM[NC_NR_DIAG_RBM] definition :



Modules of Rate-Based Monitoring :



STATE_CTR_RBM[NC_NR_DIAG_RBM] definition:



27.18.1 Ignition cycle counter

Description:

The number of ignition cycles since the first ECU power up are counted within CTR_IGK_CYC_RBM. The counter is incremented each time a driving cycle with engine running for at least 2 seconds (±1s) (C_DLY_AST_RBM) is detected. This value is only initialized with 0 on saved RAM lost (or reprogramming). If CTR_IGK_CYC_RBM is incremented LV_IGK_CYC_RBM is set to 1 to indicate that counter is already incremented within this driving cycle.

If the counter reaches its maximum value, the value is set to 0 on the next calculation.

Application conditions:

Initialisation: at first ECU power up / on saved RAM lost (or reprogramming)

CTR_IGK_CYC_RBM = 0

at reset

CTR_IGK_CYC_RBM = restored from NVMY

at LV_DC 0 → 1 transition and ECU reset

LV_IGK_CYC_RBM = 0

Recurrence: 1 s

Activation: LV_DC = 1 **and** LV_IGK_CYC_RBM = 0

Formula section:

Quantity of driving cycles calculation:

```
If    T_AST >= C_DLY_AST_RBM        (time after start at least C_DLY_AST_RBM)
Then
        If     CTR_IGK_CYC_RBM = 65535 (maximum value of CTR_IGK_CYC_RBM)
        Then
               CTR_IGK_CYC_RBM = 0
        Else
               CTR_IGK_CYC_RBM = CTR_IGK_CYC_RBM + 1
        Endif
        LV_IGK_CYC_RBM = 1          (CTR_IGK_CYC_RBM incremented this DC)
Endif
```

27.18.2 General denominator calculation

Description:

The counter for the general denominator CTR_CDN_OBD_RBM is incremented each time a valid DC with standardized vehicle operation (LV_DC_RBM = 1) is recognized. The numbers of driving cycles are counted since the first ECU power up. This value is only initialized with 0 on saved RAM lost (or reprogramming).

If the counter reaches its maximum value, the value is set to 0 on the next calculation.

Application conditions:

Initialisation: at first ECU power up / on saved RAM lost (or reprogramming)

CTR_CDN_OBD_RBM = 0

at reset

CTR_CDN_OBD_RBM = restored from NVMY

Recurrence: -

Activation: LV_DC_RBM 0→1 transition

Formula section:

```
If      CTR_CDN_OBD_RBM = 65535 (maximum value of CTR_CDN_OBD_RBM)
Then
        CTR_CDN_OBD_RBM = 0
Else
        General denominator calculation:
        CTR_CDN_OBD_RBM     = CTR_CDN_OBD_RBM + 1
Endif
```

27.18.3 Monitor individual numerator calculation

Description:

For all monitors requiring Rate-Based Monitoring (m = 0 to (NC_NR_DIAG_RBM – 1)) a numerator must be calculated. This numerator CTR_COMP_RBM[m] is incremented, if the conditions for the monitor were present for a sufficient time (bit 0 of STATE_RBM[m] = 1), but only one time within the driving cycle. This value is only initialized with 0 on saved RAM lost (or reprogramming)

If the counter CTR_CDN_RBM[m] is halved, because of maximum value limitation, the value CTR_COMP_RBM[m] is also divided by 2 (positive rounded).

Application conditions:

Initialisation: at first ECU power up / on saved RAM lost (or reprogramming)

CTR_COMP_RBM[m] = 0

at reset

CTR_COMP_RBM[m] = restored from NVMY

at LV_DC 0 →1 transition or ECU reset

STATE_CTR_RBM[m] = 0

with m = 0 to (NC_NR_DIAG_RBM – 1)

Recurrence: 10 s **or** LV_DC 1 → 0 transition

Activation: LV_INH_DC_RBM = 0 **and** LV_DC = 1

Formula section:

```
For m = 0 to (NC_NR_DIAG_RBM – 1)
    all monitor requiring Rate-Based Monitoring
    { see specification “Rate-Based Monitoring (Appl. Inc.)” }
    If      bit 0 of STATE_CTR_RBM[m] = 0
            (Incrementation of CTR_COMP_RBM[m] not yet done in this DC)
    Then    If      bit 0 of STATE_RBM[m] = 1
                    (monitor is done for RBM)
            and     bit 1 of STATE_RBM[m] = 0
                    (monitor isn’t inhibited because of a system fault)
            Then    CTR_COMP_RBM[m] = CTR_COMP_RBM[m] + 1
                    bit 0 of STATE_CTR_RBM[m] = 1
                    (Incrementation of CTR_COMP_RBM[m] done in DC)
            Endif
    Endif
Endfor
```

27.18.4 Monitor individual denominator calculation

Description:

In comparison to the counter CTR_IGK_CYC_RBM, counting the numbers of driving cycles (longer C_DLY_AST_RBM) since the first ECU power-up, the monitor individual denominator is counting the number of DC in which special vehicle/driving and ambient conditions are met individually for each monitor.

The counter CTR_CDN_RBM[m] is managed individually for each monitor (m = 0 to (NC_NR_DIAG_RBM – 1)), for which the monitoring of the real in-use performance is made. To increment the monitor denominator, some common conditions must be fulfilled (LV_DC_RBM = 1) but also monitor individual conditions. These monitor individual conditions are managed within the monitor function (bit 2 of STATE_RBM[m]). The value is only initialized with 0 on saved RAM lost (or reprogramming).

If the counter CTR_COMP_RBM[m] is halved, because of maximum value limitation, the value CTR_CDN_RBM[m] is also divided by 2 (positive rounded).

Application conditions:

Initialisation: at first ECU power up / on saved RAM lost (or reprogramming)

CTR_CDN_RBM[m] = 0

at reset

CTR_CDN_RBM[m] = restored from NVMY

with m = 0 to (NC_NR_DIAG_RBM – 1)

Recurrence: 10 s **or** LV_DC 1 → 0 transition (before LV_DC_RBM initialisation performed at LV_DC 1 → 0)

Activation: LV_INH_DC_RBM = 0 **and** LV_DC_RBM = 1

Formula section:

```
For m = 0 to (NC_NR_DIAG_RBM – 1)
     all monitor requiring Rate-Based Monitoring
     { see specification “Rate-Based Monitoring (Appl. Inc.)” }
     If      bit 1 of STATE_CTR_RBM[m] = 0
             (incrementation is not done yet)

     Then    If      bit 1 of STATE_RBM[m] = 0
                     (monitor is not inhibit, because of a system fault)

             and     bit 2 of STATE_RBM[m] = 1
                     (monitor individual RBM conditions are met within this DC)

             Then    CTR_CDN_RBM[m] = CTR_CDN_RBM[m] +1

                             bit 1 of STATE_CTR_RBM[m] = 1
                     (incrementation is done within DC)

             Endif
     Endif
Endfor
```

27.18.5 Counter overflow treatment for monitor individual numerator and denominator

Description:

If either the numerator or denominator for a specific component reaches the maximum value (65535), both numbers shall be divided by two before either is incremented again to avoid overflow problems.

Application conditions:

Initialisation:	-
Recurrence:	-
Activation:	LV_DC 0 → 1 transition

Formula section:

```
For m = 0 to (NC_NR_DIAG_RBM – 1)
     all monitor requiring Rate-Based Monitoring
     { see specification “Rate-Based Monitoring (Appl. Inc.)” }
     If      CTR_CDN_RBM[m] = 65535 (maximum value of CTR_CDN_RBM[m])

             or

             CTR_COMP_RBM[m] = 65535 (maximum value of CTR_COMP_RBM[m])
     Then    CTR_CDN_RBM[m] = CTR_CDN_RBM[m] / 2

             CTR_COMP_RBM[m] = CTR_COMP_RBM[m] / 2

     Endif

Endfor
```

27.18.6 Standardized vehicle operation for Rate-Based Monitoring

FUNCTION DESCRIPTION:

General information:

For the calculation of the denominators for Rate-Based Monitoring, CARB standardized vehicle operations, which must be met to increment the denominator counter.

Description:

A special driving cycle (LV_DC_RBM) for Rate-Based Monitoring is defined, which is set, if the following standardized vehicle operations are fulfilled:

- Cumulated trip length (time since engine start) T_AST_RBM >= C_T_AST_RBM

  (CARB : 600 s)
- Cumulative vehicle operation >= C_VS_THD_RBM for minimum of C_T_VS_RBM

  (CARB : >= 40 km/h (25mph) for at least 300 s)
- Continuous vehicle operation in idle equal or longer than C_T_IS_RBM (accelerator pedal released and vehicle speed less or equal than 1 mph)

  (CARB : 30 s)

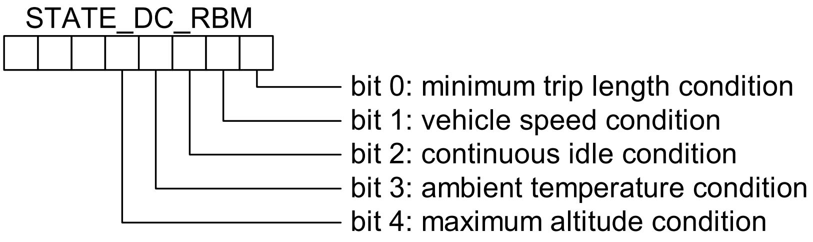
with

- altitude such as ambient pressure AMP > C_AMP_MIN_RBM

  (CARB : 8000 feet)
- ambient temperature (TAM) >= C_TAM_MIN_RBM

  (CARB : 20 °F)

Within STATE_DC_RBM the different conditions necessary to set LV_DC_RBM are visualized.



bit 0: minimum trip length condition
bit 1: vehicle speed condition
bit 2: continuous idle condition
bit 3: ambient temperature condition
bit 4: maximum altitude condition

bit value = 0: condition not met yet; bit value = 1: condition met

Application conditions:

Initialisation: at LV_DC 0 → 1 transition **or** reset

LV_DC_RBM = 0
STATE_DC_RBM = 0H
T_AST_RBM = 0 s
T_VS_RBM = 0 s
T_IS_RBM = 0 s

at LV_DC 1 → 0 transition

after individual numerator/denominator computations:

LV_DC_RBM = 0
STATE_DC_RBM = 0H

Recurrence: 100 ms

Activation: LV_DC = 1
and
LV_DC_RBM = 0
and
LV_INH_DC_RBM = 0

Formula section:

Ambient temperature

If (1)	TAM >= C_TAM_MIN_RBM		(ambient temperature condition)
Then (1)	bit 3 of STATE_DC_RBM	= 1	(ambient temperature condition reached)
Else (1)	bit 3 of STATE_DC_RBM	= 0	(ambient temperature condition not reached)
Endif (1)			

Maximum altitude

If (1)	AMP > C_AMP_MIN_RBM		(maximum altitude condition)
Then (1)	bit 4 of STATE_DC_RBM	= 1	(maximum altitude condition reached)
Else (1)	bit 4 of STATE_DC_RBM	= 0	(maximum altitude condition not reached)
Endif (1)			

```
If (1)      bit 3 and bit 4 of STATE_DC_RMB = 1
Then (1)    Minimum trip length
            If (2)      bit 0 of STATE_DC_RBM = 0    (minimum trip length not reached yet)
                        and
                        LV_ES = 0
            Then (2)    T_AST_RBM  = T_AST_RBM + 100 ms
                        If (3)      T_AST_RBM = C_T_AST_RBM
                        Then (3)    bit 0 of STATE_DC_RBM = 1
                        Endif (3)
            Endif (2)
            Vehicle speed
            If (2)      bit 1 of STATE_DC_RBM = 0 (vehicle speed condition not met yet)
                        and
                        VS >= C_VS_THD_RBM
            Then (2)    T_VS_RBM = T_VS_RBM + 100 ms
                        If (3)      T_VS_RBM = C_T_VS_RBM
                        Then (3)    bit 1 of STATE_DC_RBM = 1
                        Endif (3)
            Endif (2)
            Continuous idle
            If (2)      bit 2 of STATE_DC_RBM = 0    (continuous idle cond. not met yet)
            Then (2)    If (3)      LV_IS = 1               (engine state = 'idle')
                                    and
                                    PV <= C_PV_IS_RBM   (accelerator pedal released)
                                    and
                                    VS <= C_VS_THD_IS_CDN_RBM (CARB: 1 mile/h)
                        Then (3)    If (4)      T_IS < T_IS_RBM  (new idle phase detected)
                                    Then (4)    T_IS_RBM = 0
                                    Else (4)    T_IS_RBM = T_IS_RBM + 100 ms
                                                If (5)          T_IS_RBM = C_T_IS_RBM
                                                Then (5)    bit 2 of STATE_DC_RBM = 1
                                                Endif (5)
                                    Endif (4)
                        Else (3)    T_IS_RBM = 0
                        Endif (3)
            Endif (2)
```

```
Else (1)    T_IS_RBM = 0        (ambient temperature and maximum altitude conditions not
                                met during the whole continuous idle speed condition)
Endif (1)
```

Global information about DC including standardized vehicle operations

```
If (1)      STATE_DC_RBM = 1FH      (all conditions are met)
Then (1)    LV_DC_RBM = 1
Endif (1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_AST_RBM	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum trip length for Rate-Based Monitoring					
C_VS_THD_RBM	1	0...FFH	0...255	1	[km/h]
Minimum vehicle speed threshold to start timer T_VS_RBM					
C_T_VS_RBM	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time with VS_H_RES greater or equal than C_VS_THD_RBM for Rate-Based Monitoring					
C_TAM_MIN_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Minimum ambient temperature for Rate-Based Monitoring					
C_AMP_MIN_RBM	1	0...FFFFH	0...5434	0.0829175	[hPa]
Minimum ambient pressure for Rate-Based Monitoring (relation to altitude)					
C_DLY_AST_RBM	1	0...FFFFH	0...6553.5	0.1	[s]
Time after start, necessary to increment the ignition cycle counter					
C_T_IS_RBM	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time of continuous idle operation for Rate-Based Monitoring					
C_VS_THD_IS_CDN_RBM	1	0...FFH	0...25.5	0.1	[km/h]
Vehicle speed threshold for continuous idle speed operation, for Rate-Based Monitoring					
C_PV_IS_RBM	1	0...3FFH	0...99.90234375	0.09765625	[%]
Minimum threshold to detect accelerator pedal released					

27.19 Rate-Based Monitoring – MSD project (Appl. Inc.)

27.19.1 Definition of diagnosis using RBM

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
STATE_RBM[NC_NR_DIAG_RBM]	O	0...FH	0...15	1	[-]
Monitor interface for the Rate-Based Monitoring statistics					

Input data:

LV_DC	STATE_RBM_DYN_VLD_LS_UP[NC_CBK_EX_NR]	STATE_RBM_SHIFT_AFL_LSL_UP[NC_CBK_EX_NR]	STATE_RBM_SHIFT_AFR_LSL_UP[NC_CBK_EX_NR]
STATE_RBM_AIR_LSL_UP[NC_CBK_EX_NR]	STATE_RBM_DIAGCPS	STATE_RBM_SMALL_LEAK	STATE_RBM_ROUGH_LEAK
STATE_RBM_REF_CRK_CAM_EX[NC_NR_CAM_CBK]	STATE_RBM_REF_CRK_CAM_IN[NC_NR_CAM_CBK]	STATE_RBM_MEC_IVVT_IN	STATE_RBM_MEC_IVVT_EX
STATE_RBM_TOOTH_OFF_IN[NC_NR_CAM_CBK]	STATE_RBM_TOOTH_OFF_EX[NC_NR_CAM_CBK]	STATE_RBM_TH	STATE_RBM_TCO_PLAUS
STATE_RBM_TCO_STUCK	STATE_RBM_TCO_2_PLAUS	STATE_RBM_TAM_PLAUS	STATE_RBM_VS_PLAUS
STATE_RBM_FTL_OBD	STATE_RBM_OBD_LSH_DOWN[NC_CBK_EX_NR]	STATE_RBM_OBD_VLD_LSH_UP[NC_CBK_EX_NR]	STATE_RBM_CS
STATE_RBM_ISC	STATE_RBM_MAF	STATE_RBM_TIA_PLAUS	STATE_RBM_AMP_PLAUS
STATE_RBM_LOAD_TPS_PLAUS	STATE_RBM_MAP_DIP_PLAUS	STATE_RBM_VLS_DOWN_DIF[NC_CBK_EX_NR]	STATE_RBM_CHK_LS_DOWN[NC_CBK_EX_NR]
STATE_RBM_SWT_LS_DOWN[NC_CBK_EX_NR]	STATE_RBM_T_ES	STATE_RBM_TPS	STATE_RBM_ISC_CST
STATE_RBM_TQ_CST	STATE_RBM_TCO_STUCK_RNG	STATE_RBM_PUC_LS_DOWN[NC_CBK_EX_NR]	

FUNCTION DESCRIPTION:

Description:

Depending on the system configuration and the customer requirements, the diagnosis individual counters for Rate-Based Monitoring, the numerator and denominator, must be calculated for specific monitors.

Within this module the monitors using RBM statistics are defined.

Application conditions:

Initialisation : at ECU reset and LV_DC 0 → 1 transition :

```
For i = 0 to NC_NR_DIAG_RBM do
    STATE_RBM[i] = 0
Endfor
```

Recurrence: 10 s { always executed prior to Rate-Based Monitoring main module }

Activation: every engine operating state

Formula section:

	Interface byte	Group Name
0	STATE_RBM_CAT_DIAG_1	CAT_1
1	STATE_RBM_CAT_DIAG_2	CAT_2
2	STATE_RBM_DYN_VLD_LS_UP_1	LS_UP_1
3	STATE_RBM_DYN_VLD_LS_UP_2	LS_UP_2
4	STATE_RBM_SHIFT_AFL_LSL_UP_1	LS_UP_1
5	STATE_RBM_SHIFT_AFL_LSL_UP_2	LS_UP_2
6	STATE_RBM_SHIFT_AFR_LSL_UP_1	LS_UP_1
7	STATE_RBM_SHIFT_AFR_LSL_UP_2	LS_UP_2
8	STATE_RBM_AIR_LSL_UP_1	LS_UP_1
9	STATE_RBM_AIR_LSL_UP_2	LS_UP_2
10	STATE_RBM_SMALL_LEAK	EVAP
11	STATE_RBM_REF_CRK_CAM_IN_1	EGR_VVT
12	STATE_RBM_REF_CRK_CAM_EX_1	EGR_VVT
13	STATE_RBM_MEC_IVVT_IN	EGR_VVT
14	STATE_RBM_MEC_IVVT_EX	EGR_VVT
15	STATE_RBM_TOOTH_OFF_IN_1	EGR_VVT
16	STATE_RBM_TOOTH_OFF_EX_1	EGR_VVT
17	0 (not used)	SA
18	STATE_RBM_DIAGCPS	CUSTOMER
19	STATE_RBM_ROUGH_LEAK	CUSTOMER
20	STATE_RBM_DMTLM	CUSTOMER
21	STATE_RBM_TH	CUSTOMER
22	STATE_RBM_TCO_PLAUS	CUSTOMER
23	STATE_RBM_TCO_STUCK	CUSTOMER
24	STATE_RBM_TCO_2_PLAUS	CUSTOMER
25	STATE_RBM_TAM_PLAUS	CUSTOMER
26	STATE_RBM_VS_PLAUS	CUSTOMER
27	STATE_RBM_FTL_OBD	CUSTOMER
28	STATE_RBM_OBD_LSH_DOWN_1	CUSTOMER
29	STATE_RBM_OBD_LSH_DOWN_2	CUSTOMER
30	STATE_RBM_OBD_VLD_LSH_UP_1	CUSTOMER
31	STATE_RBM_OBD_VLD_LSH_UP_2	CUSTOMER
32	STATE_RBM_CS	CUSTOMER
33	STATE_RBM_ISC	CUSTOMER
34	STATE_RBM_MAF	CUSTOMER

35	STATE_RBM_TIA_PLAUS	CUSTOMER
36	STATE_RBM_AMP_PLAUS	CUSTOMER
37	STATE_RBM_LOAD_TPS_PLAUS	CUSTOMER
38	STATE_RBM_MAP_DIP_PLAUS	CUSTOMER
39	STATE_RBM_VLS_DOWN_DIF_1	CUSTOMER
40	STATE_RBM_VLS_DOWN_DIF_2	CUSTOMER
41	STATE_RBM_CHK_LS_DOWN_1	CUSTOMER
42	STATE_RBM_CHK_LS_DOWN_2	CUSTOMER
43	STATE_RBM_SWT_LS_DOWN_1	CUSTOMER
44	STATE_RBM_SWT_LS_DOWN_2	CUSTOMER
45	STATE_RBM_T_ES	CUSTOMER
46	STATE_RBM_TPS	CUSTOMER
47	STATE_RBM_ISC_CST	CUSTOMER
48	STATE_RBM_TQ_CST	CUSTOMER
49	STATE_RBM_TCO_STUCK_RNG	CUSTOMER
50	STATE_RBM_PUC_LS_DOWN_1	CUSTOMER
51	STATE_RBM_PUC_LS_DOWN_2	CUSTOMER
NC_NR_DIAG_RBM	**52**	

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_DIAG_RBM	1	0...FFH	0...255	1	[-]
Instance number of monitor using RBM statistics					

27.19.2 Clear rate base monitoring statistics

Import actions:

ACTION_ERRM_ClearRbmStatistics (OUT < ResultClrInfo>)

Description:

For development conveniences, it is often useful to reinit the rate base monitoring statistics. This is possible by setting the calibration bit LC_RBM_CLR to 1.

Application conditions:

Initialisation: -

Recurrence: -

Activation: 0 → 1 transition of LC_RBM_CLR

Formula section:

If LC_RBM_CLR = 1

Then ACTION_ERRM_ClearRbmStatistics (OUT < ResultClrInfo>)

Endif

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_RBM_CLR	1	0...1H	0...1	1	[-]
reinit the rate base monitoring statistics at LC_RBM_CLR 0->1					

27.19.3 Inhibition of standardized vehicle operation calculation

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_INH_DC_RBM	V/O	0...1H	0...1	1	[-]
Inhibition of standardized vehicle operation calculation, due to a system failure that influences the conditions necessary to set LV_DC_RBM					

Input data:

LV_ERR_AMP_PLAUS	LV_DC_RBM	LV_ERR_VS	CTR_ERR_DYN_NR
LV_ERR_VS	LV_ERR_TAM_CAN	LV_ERR_TAM_PLAUS	

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT <PendingStatus>)

FUNCTION DESCRIPTION:

General information:

The calculation of the standardized vehicle operation LV_DC_RBM, necessary for the calculation of the denominators for the Rate-Based Monitoring functionality, depends on different engine and ambient conditions.

Description:

With the value LV_INH_DC_RBM the inhibition of the standardized vehicle operation calculation is controlled. An inhibition is possible due to a system failure that influences the conditions necessary to set LV_DC_RBM.

Application conditions:

Initialisation: at ECU reset or LV_DC 0 → 1 transition : LV_INH_DC_RBM = 0

Recurrence: 1 s

Activation: LV_DC = 1

and LV_INH_DC_RBM = 0

Formula section:

At LV_DC 0 → 1 transition :

The pending status of the following failures has to be checked only once :

- LV_ERR_AMP
- LV_ERR_AMP_PLAUS
- LV_ERR_VS
- LV_ERR_TAM_PLAUS
- LV_ERR_TAM_CAN

```
If(1)       { CPU optimization at LV_DC 0 → 1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While LV_INH_DC_RBM = 0 do
                  with each XX failure of the above list :
                  ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                  SYNCHRONIZATION<CALL>)
                  If(2)         XX has a pending status
                  Then(2)
                                LV_INH_DC_RBM = 1
                  Endif(2)
            Endwhile
Else(1)
            { the dynamic failure memory is empty }
            No action
Endif(1)
```

Every 1 s :

```
If          LV_ERR_AMP = 1
     or     LV_ERR_AMP_PLAUS = 1
     or     LV_ERR_VS = 1
     or     LV_ERR_TAM_PLAUS = 1
     or     LV_ERR_TAM_CAN = 1
Then        LV_INH_DC_RBM = 1
Endif
```

27.19.4 Calculation of cold start denominator

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CDN_CST_RBM	V/O	0...1H	0...1	1	[-]
Cold start conditions for individual denominator calculation					
T_AST_CST_RBM	V	0...FFFFH	0...65535	1	[s]
Comulative time (depending on ambient temperature) since engine start					

Input data:

TCO_ST	TAM	TAM_ST	

FUNCTION DESCRIPTION:

General information:

The individual monitor condition for evaporative system denominator calculation LV_CDN_CST_RBM (see §1968.2 - 4.3.2, section D) can be applied for other diagnosis (Bit 2 calculation of STATE_RBM_xx) that require extentend monitoring evaluation e.g. coolant or intake air rationality.

Application conditions:

Initialisation: at LV_DC 0 → 1 transition **or** reset

T_AST_CST_RBM = 0s

LV_CDN_CST_RBM = 0

Recurrence: 1s

Activation: LV_DC = 1

and

LV_CDN_CST_RBM = 0

and

LV_INH_DC_RBM = 0

Formula section:

Comulative time (depending on ambient temperature) since engine start:

```
If      C_TAM_MIN_CST_RBM ≤ TAM  ≤ C_TAM_MAX_CST_RBM
        (ambient temperature greater than or equal to 40 degrees Fahrenheit but
        less than or equal to 95 degrees Fahrenheit)
Then    If     T_AST_CST_RBM ≥ C_T_AST_CST_RBM
        Then   T_AST_CST_RBM keeps old value
        Else   T_AST_CST_RBM = T_AST_CST_RBM + 1s
        Endif
Else    no incrementation of T_AST_CST_RBM
```

```
Endif

Calculation of cold start conditions for RBM encountered within this DC
If      LV_CDN_CST_RBM = 0
Then    If      T_AST_CST_RBM ≥ C_T_AST_CST_RBM
                (time since start is greater than or equal to 600 seconds)
            and C_TCO_ST_MIN_CST_RBM ≤ TCO_ST  ≤
                C_TCO_ST_MAX_CST_RBM
                (engine coolant temperature at engine start is greater than or equal to 40
                degrees Fahrenheit but less than or equal to 95 degrees Fahrenheit)
            and TCO_ST ≤  (TAM_ST + C_TAM_ST_HYS_CST_RBM )
                (less than or equal to 12 degrees Fahrenheit higher than ambient
                temperature at engine start
        Then LV_CDN_CST_RBM = 1
        Endif
 Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TAM_MIN_CST_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Minimum ambient temperature threshold for RBM cold start condition					
C_TAM_MAX_CST_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Maximum ambient temperature threshold for for RBM cold start condition					
C_T_AST_CST_RBM	1	0...FFFFH	0...65535	1	[s]
Minimum comulative time since engine start for RBM cold start condition					
C_TCO_ST_MIN_CST_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Minimum engine start temperature threshold for RBM cold start condition					
C_TCO_ST_MAX_CST_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Maximum engine start temperature threshold for RBM cold start condition					
C_TAM_ST_HYS_CST_RBM	1	0...FEH	0...190.5	0.75	[°C]
Ambient temperature hysteresis for RBM cold start condition					

27.20 Failure Classes

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
WAL_CONF_XX	O	0...FFH	0...255	1	[-]
Lamp configuration for failure XX					
PRI_CONF_XX	O	0...7H	0...7	1	[-]
Priority configuration for failure XX					
CTR_DC_INC_XX	O	0...FFH	0...255	1	[-]
Increment of the driving cycle counter for failure XX					
CTR_DC_DEC_XX	O	0...FFH	0...255	1	[-]
Decrement of the driving cycle counter for failure XX					
CTR_DC_MAX_XX	O	0...FFH	0...255	1	[-]
Maximum value of the driving cycle counter for failure XX					

FUNCTION DESCRIPTION:

General information:

Because the diagnosis system becomes more and more complex, to avoid multiplication of calibrations and to help the tuning team for simplifying the calibration process, each failure is defined via classes of failure.

Each class defines the type of this failure as emission relevant or not, as MIL handling or not, as using driving cycle for MIL illumination, and so on.

The configuration of a failure is defined by WAL_CONF_XX, PRI_CONF_XX, CTR_DC_INC_XX, CTR_DC_DEC_XX and CTR_DC_MAX_XX.

Format of WAL_CONF_XX		
Bit	Logical value	Description
0	LC_WAL_1_ON	Enable WAL_1 (0 : Off / 1 : On) See details in “Lamp management” module.
1	LC_WAL_2_ON	Enable WAL_2 (0 : Off / 1 : On) See details in “Lamp management” module.
2	LC_MIL_ON	Enable MIL (0 : Off / 1 : On) See details in “Lamp management” module.
3	LC_MIL_FLL	Enable flash mode for MIL (0 : Off / 1 : On) See details in “Lamp management” module.
4	LC_OBD_ERR	Defines failure as a “CARB/EOBD failure” (emission relevant). (failure visible via scantool) 0 : failure is <u>not</u> considered as a “CARB/EOBD failure” 1 : failure is considered as a “CARB/EOBD failure”

Format of PRI_CONF_XX		
Bit	**Logical value**	**Description**
0	LC_ERR_PRI_1	This option defines the priority of the XX diagnosis.
1	LC_ERR_PRI_2	It is used for storage of failure and freeze frame.
2	LC_ERR_PRI_3	000: lowest priority for errors which are not significant for emission 011: highest priority for errors, which are not significant for emission 100: lowest priority for errors, which are significant for emission 111: highest priority which are significant for emission See details in “Priority rules” module. For emission relevant (EOBD, CARB) failure : LC_ERR_PRI_3 = 1 For non emission relevant failure : LC_ERR_PRI_3 = 0

27.20.1 Failure assignation

For each failure XX, a failure class is allocated. The failure class number makes this affectation, which is a calibration named C_ERR_CLAS_XX.

This table is defined in the “General Diagnosis Information”.

Diagnosis instance	Failure Class number Class A / Class B
Failure XX	C_ERR_CLAS_XX
Failure YY	C_ERR_CLAS_YY
...	...

The failure class number is a byte divided in two subclasses named class B (defined by the most significant quartet) and class A (defined by the less significant quartet).

Failure class number for failure XX

C_ERR_CLAS_XX =

(	Class B		)	(	Class A		)

27.20.2 Failure class definition

A failure class defines for a failure, which belongs to this failure class :

Part sub-class A :

- if the failure have impact on MIL illumination or not ;
- if the failure have impact on MIL blinking or not ;
- if the failure have impact on Warning Lamp 1 or not ;
- if the failure have impact on Warning lamp 2 or not ;
- if the failure is considered as a CARB/EOBD failure (emission relevant)
  (failure visible with scantool, failure can set OBD freeze frame for mode 2h)

Part sub-class B :

- the increment value for driving cycle counter management ;
- the maximum value of driving cycle counter ;
- the decrement value for driving cycle counter management ;
- the priority of the failure.

Each subclass of a failure class is composed of 4 bits, so 16 different values are possible for each subclass. For each subclass, some values are predefined and some values are reserved for improvement/validation and are managed by calibration.

Sub Class A definition : ID_ERR_CLAS_A_FMT (1 byte x 16)

Subclass A Value (i) in hex / Name	**Description**	**ID_ERR_CLAS_A_FMT [i]**
0 / NO_LAMP	*- Predefined value –* Failure without impact on any lamp	%0000 0000 (in binary)
1 / WAL_1_ON	*- Predefined value -* Failure with impact on WAL1 lamp	%0000 0001 (in binary)
2 / WAL_2_ON	*- Predefined value –* Failure with impact on WAL2 lamp	%0000 0010 (in binary)
3 / MIL_ON	*- Predefined value -* Failure with impact on Mil (on)	%0001 0100 (in binary)
4 / MIL_ON_FLL	*- Predefined value –* Failure with impact on Mil (on & blinking)	%0001 1100 (in binary)
5 / LAMP_CUS_1	*Free for customer definition*	*Free for customer definition*
6 / LAMP_CUS_2	*Free for customer definition*	*Free for customer definition*
7 / LAMP_CUS_3	*Free for customer definition*	*Free for customer definition*
8 / LAMP_CUS_4	*Free for customer definition*	*Free for customer definition*
9 / LAMP_CUS_5	*Free for customer definition*	*Free for customer definition*
A / LAMP_CUS_6	*Free for customer definition*	*Free for customer definition*
B / LAMP_CUS_7	*Free for customer definition*	*Free for customer definition*
C / LAMP_CUS_8	*Free for customer definition*	*Free for customer definition*
D / LAMP_CUS_9	*Free for customer definition*	*Free for customer definition*
E / LAMP_CUS_10	*Free for customer definition*	*Free for customer definition*
F / LAMP_CUS_11	*Free for customer definition*	*Free for customer definition*

Formula section:

For each failure XX allocated to the failure class number C_ERR_CLAS_XX :

WAL_CONF_XX = ID_ERR_CLAS_A_FMT [4 Less Significant Bit of C_ERR_CLAS_XX]

Sub Class B definition : ID_ERR_CLAS_B_FMT (4 bytes x 16)

		ID_ERR_CLAS_B_FMT			
Subclass B Value (i) in hex/ Name	**Description**	**[i] [0] DC inc**	**[i] [1] DC max**	**[i] [2] DC dec**	**[i] [3] Priority**
0 / NO_LAW_L	*- Predefined value –* Non Carb/EOBD failure with low priority	1	1	1	0
1 / NO_LAW_M	*- Predefined value -* Non Carb/EOBD failure with medium priority	1	1	1	1
2 / NO_LAW_H	*- Predefined value –* Non Carb/EOBD failure with high priority	1	1	1	2
3 / LAW_CC_L	*- Predefined value -* Carb/EOBD failure with low priority Comprehensive component	3	3	1	4
4 / LAW_L	*- Predefined value -* Carb/EOBD failure with low priority	3 / 2 *)	6	2	4
5 / LAW_M	*- Predefined value -* Carb/EOBD failure with medium priority	3 / 2 *)	6	2	5
6 / LAW_H	*- Predefined value -* Carb/EOBD failure with high priority	3 / 2 *)	6	2	6
7 / VALID	For validation	1	1	1	4
8 / NO_ERASE	*- Predefined value -* failure with high priority, not erasable by warm-up cycles.	3	3	0	6
9 / DC_CUS_1	*Free for customer definition*	-	-	-	-
A / DC_CUS_2	*Free for customer definition*	-	-	-	-
B / DC_CUS_3	*Free for customer definition*	-	-	-	-
C / DC_CUS_4	*Free for customer definition*	-	-	-	-
D / DC_CUS_5	*Free for customer definition*	-	-	-	-
E / DC_CUS_6	*Free for customer definition*	-	-	-	-
F / DC_CUS_7	*Free for customer definition*	-	-	-	-

Formula section:

For each failure XX allocated to the failure class number C_ERR_CLAS_XX :

CTR_DC_INC_XX = ID_ERR_CLAS_B_FMT[4 Most Significant Bit of C_ERR_CLAS_XX][0]

CTR_DC_MAX_XX = ID_ERR_CLAS_B_FMT[4 Most Significant Bit of C_ERR_CLAS_XX][1]

CTR_DC_DEC_XX = ID_ERR_CLAS_B_FMT[4 Most Significant Bit of C_ERR_CLAS_XX][2]

Bit 0,1,2 of PRI_CONF_XX = ID_ERR_CLAS_B_FMT [4 Most Significant Bit of C_ERR_CLAS_XX][3]

Remark: The increment of the DC counter (*) for the failures defined by the failure classes LAW_L, LAW_M and LAW_H depends on the required regulation (Carb/EOBD).

27.20.3 Example

To define the failure MIS_A (Misfiring A) as :

- a failure with high priority
- with only impact on MIL (On and Blinking)

Set the failure MIS_A in class number 64 (in hex) with C_ERR_CLAS_MIS_A = 64H.

That means :

- The value of sub class B is 6H => failure with high priority ;
- The value of sub class A is 4H => Failure with impact on MIL (On and blinking) ;

= > Failure is a “CARB/EOBD failure”

DC_INC_MIS_A = 3

DC_MAX_MIS_A = 6

DC_DEC_MIS_A = 2

WAL_CONF_MIS_A = %0001 1100 (in binary)

PRI_CONF_MIS_A = %0000 0110 (in binary)

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_ERR_CLAS_XX	1	0...FFH	0...255	1	[-]
Failure class number for failure XX					
ID_ERR_CLAS_A_FMT	16	0...FFH	0...255	1	[-]
LDPM_CLAS_A_ID_CLAS_FMY	16	0...FFFFH	0...65535	1	[-]
Sub Class A definition					
ID_ERR_CLAS_B_FMT	4*16	0...FFH	0...255	1	[-]
LDP_CLAS_B_ID_CLAS_FMY	4	0...FFFFH	0...65535	1	[-]
LDPM_CLAS_A_ID_CLAS_FMY	16	0...FFFFH	0...65535	1	[-]
Sub Class B definition					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_ERR_PRI_H	1	0...7H	0...7	1	[-]
Failure with priority ≥ NC_ERR_PRI_H are considered as high priority failures					

27.21 Lamp management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_MIL	V/O/S	0...2H	0...2	1	[-]
State of MIL illumination commanded by error management (ON (1), OFF (0), MIL_FLL (2))					
STATE_WAL_1	V/O/S	0...1H	0...1	1	[-]
State of WAL_1 illumination commanded by error management (ON (1), OFF (0))					
STATE_WAL_2	V/O/S	0...1H	0...1	1	[-]
State of WAL_2 illumination commanded by error management (ON (1), OFF (0))					
LV_MIL	V/O	0...1H	0...1	1	[-]
Boolean indicating MIL physical output (0 : off / 1 : on)					
LV_WAL_1	V/O	0...1H	0...1	1	[-]
Boolean indicating WAL_1 physical output (0 : off / 1 : on)					
LV_WAL_2	V/O	0...1H	0...1	1	[-]
Boolean indicating WAL_2 physical output (0 : off / 1 : on)					
LV_WAL_ST	V/O	0...1H	0...1	1	[-]
Boolean indicating Pre-drive Check mode (0 : off / 1 : on)					
DLY_MIL_READY	V	0...FFFFH	0...32767.5	0.5	[s]
Delay time to unable MIL blinking during pre-drive check mode					
LV_MIL_FLL_READY	V/O	0...1H	0...1	1	[-]
Boolean indicating MIL shall blink during pre-drive check mode phase 2 (0 : not blink / 1 : blink)					
T_WAL_ST	V	0...FFH	0...127.5	0.5	[s]
Timer for pre-drive check duration					

Input data:

LV_ES	SYM_CYL_MIS_A	LV_WAL_1_ACT_REQ	LV_WAL_2_ACT_REQ
LV_MIS_STATE_A	LV_DC_MAX[NC_NR_ERR_DYN]	INH_IV_MIS	WAL_CONF_XX
LV_IGK	LV_MIL_ACT_REQ	LV_ERR_XX	STATE_READY_OBD_1
STATE_READY_OBD_2			

FUNCTION DESCRIPTION:

General information:

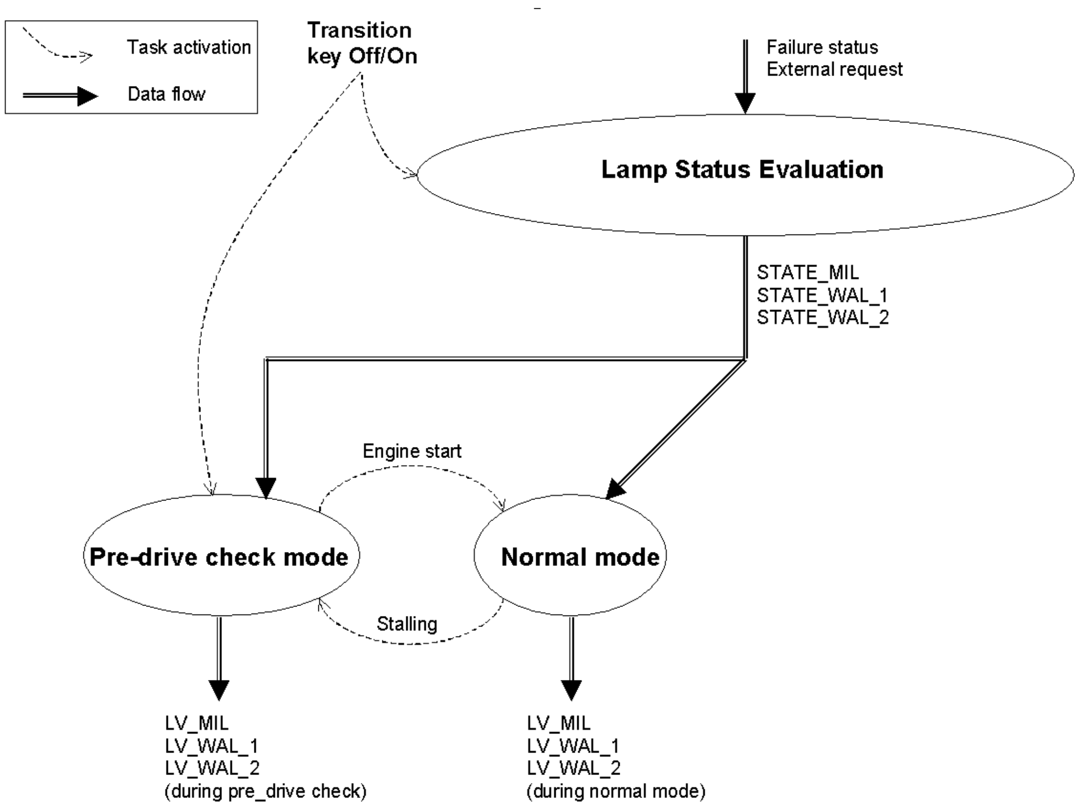
It is possible to manage up to three different lamps. They are called:

- MIL: Malfunction Indicator Lamp; this lamp is used to inform the vehicle driver in case of failures which can affect emissions.
- WAL_1: Warning Lamp 1 and WAL_2: Warning Lamp 2; these lamps are used to inform the vehicle driver in case of others failures.

For each lamp, a task evaluated the state of MIL, WAL_1 and WAL_2. These states STATE_MIL, STATE_WAL_1 and STATE_WAL_2 are respectively what error management commanded to the lamps regard to failure status.

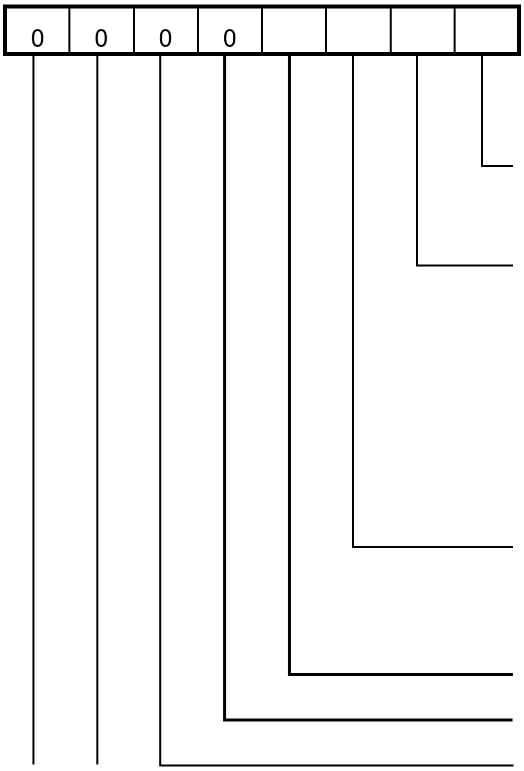
Then, the real physical state of each lamp (LV_MIL for MIL, LV_WAL_1 for WAL_1 and LV_WAL_2 for WAL_2) is build related to the mode in which they are. Lamps behavior is based on two modes:

- A predrive check mode between transition LV_IGK 0->1 and engine start; this mode is used to check the correct lamps working.
- A normal mode for the rest of the time.



A configuration byte is defined for each lamp: C_CONF_XXX (with XXX = MIL C_CONF_MIL, WAL_1 or WAL_2, C_CONF_WAL_1 C_CONF_WAL_2).

It is then possible to inhibit a lamp illumination, to allow or not external request for lamp illumination and to link lamp illumination each other's.



bit 0 of C_CONF_XXX : Global lamp XXX inhibition bit
0 : lamp XXX illumination is authorized
1 : lamp XXX is never illuminated except by an external request for lamp XXX

bit 1 of C_CONF_XXX : Lamp XXX request enable bit
0 : lamp XXX request input is never taken into account
1 : lamp XXX request input is taken into account

For MIL :
bit 2 of C_CONF_XXX: Special behavior for the Misfiring A failure in case of cylinder cut off.
0 : Without management of cylinder cut off
1 : With management of cylinder cut off
For WAL_1 and WAL_2 :
bit 2 of C_CONF_XXX : Illumination of MIL instead of lamp XXX
0 : no links between lamp XXX and MIL
1 : MIL is illuminated if illumination of lamp XXX is requested

bit 3 of C_CONF_XXX :
Inhibition of Predrive check for lamp XXX
0: Predrive check for lamp XXX enabled
1: Predrive check for lamp XXX disabled

Not used

Not used

Not used

Not used

Impact of each failure on lamp is defined in WAL_CONF_XX byte:

- bit 0 : if set, the present failure may caused a WAL_1 illumination ;
- bit 1 : if set, the present failure may caused a WAL_2 illumination ;
- bit 2 : if set, the failure may caused a MIL illumination when driving cycle counter reaches maximum ;
- bit 3: if set, the present failure may cause a MIL blinking.

27.21.1 Lamp Status Evaluation

Description:

Management of WAL_1 and WAL_2

Behavior of WAL_1 and WAL_2 are identical. WAL_1 and WAL_2 may be illuminated (state is ON) or not (state is OFF). When the global lamp inhibition bit is not set, the lamp is requested to illuminate:

- if at least one failure is present (LV_ERR_XX) with the corresponding bit of WAL_CONF_XX (bit 0 for WAL_1, bit 1 for WAL_2) in enable position and this failure is stored in 2nd layer memory ;
- or in case of allowed lamp request.

Management of MIL

MIL internal management is based on 3 states:

STATE_MIL = OFF	(MIL is requested continuously off)
STATE_MIL = ON	(MIL is requested continuously on)
STATE_MIL = MIL_FLL	(MIL is requested blinking at 1 Hz)

When the global MIL inhibition bit is not set, the MIL is requested to illuminate (ON):

- if at least one failure has reached its driving cycle maximum value and is unequal to zero (LV_DC_MAX[IDX]) with the corresponding bit of WAL_CONF_XX (bit 2) in enable position and this failure is stored in 2nd layer memory ;
- or in case of allowed lamp request ;
- or WAL_1 is requested to be illuminated and WAL_1 is link with MIL ;
- or WAL_2 is requested to be illuminated and WAL_2 is link with MIL ;

When the global MIL inhibition bit is not set, the MIL is requested to be blinking at 1hz (MIL_FLL):

- if at least one failure is present (LV_ERR_XX) with the corresponding bit of WAL_CONF_XX (bit 3) in enable position and this failure is stored in 2nd layer memory ;

A special calculation can be enabled for Misfire A failure with the management of cylinder shut off (bit 2 of C_CONF_MIL):

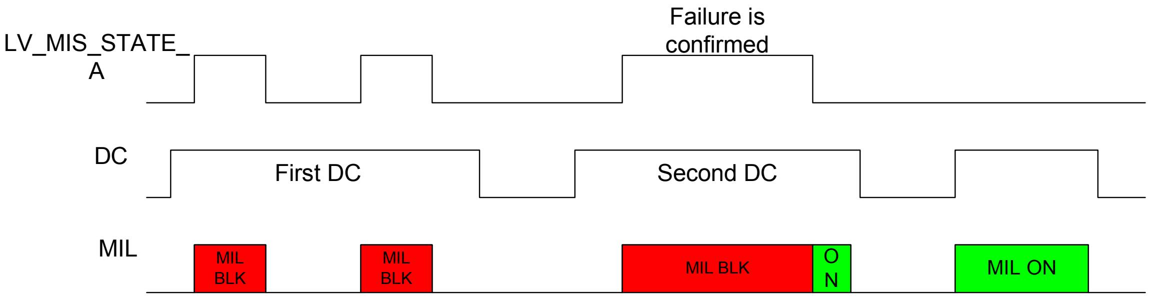
- bit 2 of C_CONF_MIL = 0: When a misfire A failure (LV_MIS_STATE_A) is present, the MIL is blinking.
- bit 2 of C_CONF_MIL = 1: If Misfire A failure (LV_MIS_STATE_A) is present with cylinder cut off, the MIL is blinking only for a calibratable time (C_DLY_INH_IV_MIS). After this time, the MIL is ON again.

Beware: The “blinking mode” is priority mode compared to “continuous on mode”.

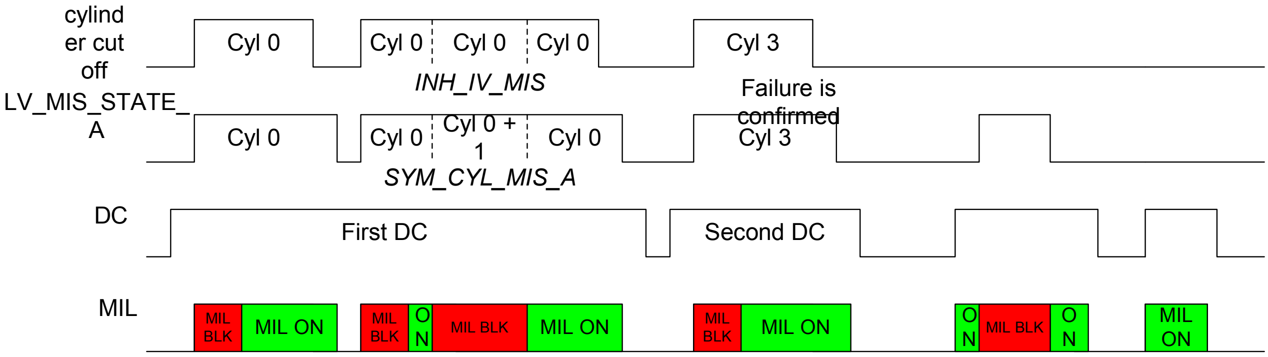
Signal flow diagram:

Special behavior for Misfire A failure with bit LC_MIL_INH_IV_MIS of C_CONF_MIL:

Behavior of Misfiring failure MIS_A without management of cylinder cut off (LC_MIL_INH_IV_MIS = 0)



Behavior of Misfiring A failure with management of cylinder cut off (LC_MIL_INH_IV_MIS = 1) :



Normal mode : MIL is ON (because the failure is confirmed)

Application conditions:

Initialization: -

Recurrence: 500 ms

Activation: -

Formula section:

```
If      bit 0 of C_CONF_WAL_1 = 1      (lamp WAL_1 illumination not authorized)
Then    STATE_WAL_1 = OFF      (lamp WAL_1 is off)
Else    If      at least one failure in 2nd layer memory has:
                        (LV_ERR_XX and bit 0 of WAL_CONF_XX) = 1
                                (lamp WAL_1 illumination by a failure)
        Then    STATE_WAL_1 = ON       (lamp WAL_1 is on)
        Else    STATE_WAL_1 = OFF      (lamp WAL_1 is off)
        Endif
Endif

If      bit 0 of C_CONF_WAL_2 = 1      (lamp WAL_2 illumination not authorized)
Then    STATE_WAL_2 = OFF      (lamp WAL_2 is off)
Else    If      at least one failure in 2nd layer memory has:
                        (LV_ERR_XX and bit 1 of WAL_CONF_XX) = 1
                                (lamp WAL_2 illumination by a failure)
        Then    STATE_WAL_2 = ON       (lamp WAL_2 is on)
        Else    STATE_WAL_2 = OFF      (lamp WAL_2 is off)
        Endif
Endif
```

```
If      bit 0 of C_CONF_MIL = 1      (MIL illumination not authorized)
Then    STATE_MIL = OFF    (MIL is requested continuously off)
Else    If      at least one failure in 2nd layer memory has:
                (LV_DC_MAX[IDX]  and bit 2 of WAL_CONF_XX ) = 1
                (at least one failure request to illuminate the MIL)
                or (STATE_WAL_1 = ON  and   bit 2 of C_CONF_WAL_1 = 1)
                (a failure illuminates WAL_1 and WAL1 is linked with MIL)
                or (STATE_WAL_2 = ON  and   bit 2 of C_CONF_WAL_2 = 1)
                (a failure illuminates WAL_2 and WAL_2 is linked with MIL)
        Then    STATE_MIL = ON      (MIL is requested continuously on)
        Else    STATE_MIL = OFF     (MIL is requested continuously off)
        Endif
        If      LV_MIS_STATE_A = 1
                (CARB A misfire failure criterion)
        then    if      bit 2 of C_CONF_MIL = 0
                        (Without management of cylinder shut off for the misfiring A)
                then    STATE_MIL = MIL_FLL
                        (MIL is requested blinking at 1Hz)
                else    (Management of cylinder shut off for the misfiring A)
                        If      (any cylinder bit within SYM_CYL_MIS_A 0 -> 1
                        and     corresponding bit within INH_IV_MIS is set to 1)
                        Then    STATE_MIL = MIL_FLL only during C_DLY_INH_IV_MIS
                                After C_DLY_INH_IV_MIS, STATE_MIL = ON
                                mode = 'blink mode during a delay + continuous ON'
                        Endif
                        if      any cylinder bit within SYM_CYL_MIS_A 0 -> 1
                                and corresponding bit within INH_IV_MIS is set to 0
                        then    mode = 'blink mode'
                        Endif
                        If      mode = 'blink mode'
                        Then    if      any cylinder bit  SYM_CYL_MIS_A is set
                                        And
                                        corresponding bit within INH_IV_MIS remains to 0
                                        (no transition 1-> 0 of the corresponding bit during
                                        the failure)
                                Then    mode = 'blink mode'
```

```
                        Else    mode ='blink mode during a delay + continuous ON'
                        Endif
                Endif
                If      mode = 'blink mode'
                Then    STATE_MIL = MIL_FLL
                        (MIL is requested blinking at 1Hz)
                else    if      C_DLY_INH_IV_MIS isn't achieved
                        Then    STATE_MIL = MIL_FLL
                                (MIL is requested blinking at 1Hz)
                        Else    STATE_MIL = ON
                                (MIL is requested continuously on)
                        Endif
                Endif
        Endif
Endif
If      at least one failure in 2nd layer memory has :
        (LV_ERR_XX and bit 3 of WAL_CONF_XX) = 1
        (at least one failure request to blink the MIL)
Then    STATE_MIL = MIL_FLL
        (MIL is requested blinking at 1Hz)
Endif
```

Endif

Remark:

For the misfiring A failure (treated as a particular case), the Bit 3 of WAL_CONF_XX mustn't be set to 1 by calibration (only bit 2 for continuous MIL illumination).

27.21.2 Lamp mode

27.21.2.1 Predrive Check mode

Description:

Predrive check mode is used to enable a visual check of the correct lamp working. Additionally it can be used to display readiness status as an option (MIL blinking).

Nevertheless a failure may also illuminate a lamp. Illumination of lamp during pre-drive check mode is performed only when :

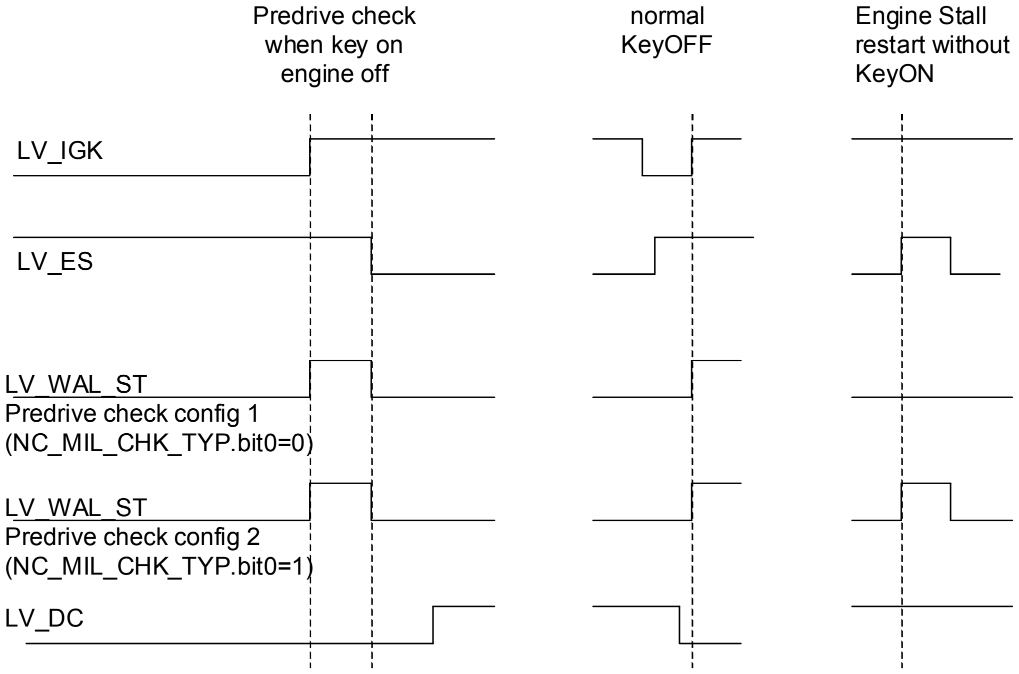
- the global lamp inhibition bit is not set (bit 0 of C_CONF_XXX = 1) and the lamp individual predrive check inhibition is not set (bit 3 of C_CONF_XXX = 1)
- or error management commands a lamp

A) Configuration of predrive check behaviour in case of stalling event:

with NC_MIL_CHK_TYP (bit 0)

Predrive check mode phase can be defined in 2 ways:
- between key on transition LV_IGK 0->1 and engine start (bit 0 of NC_MIL_CHK_TYP=0)

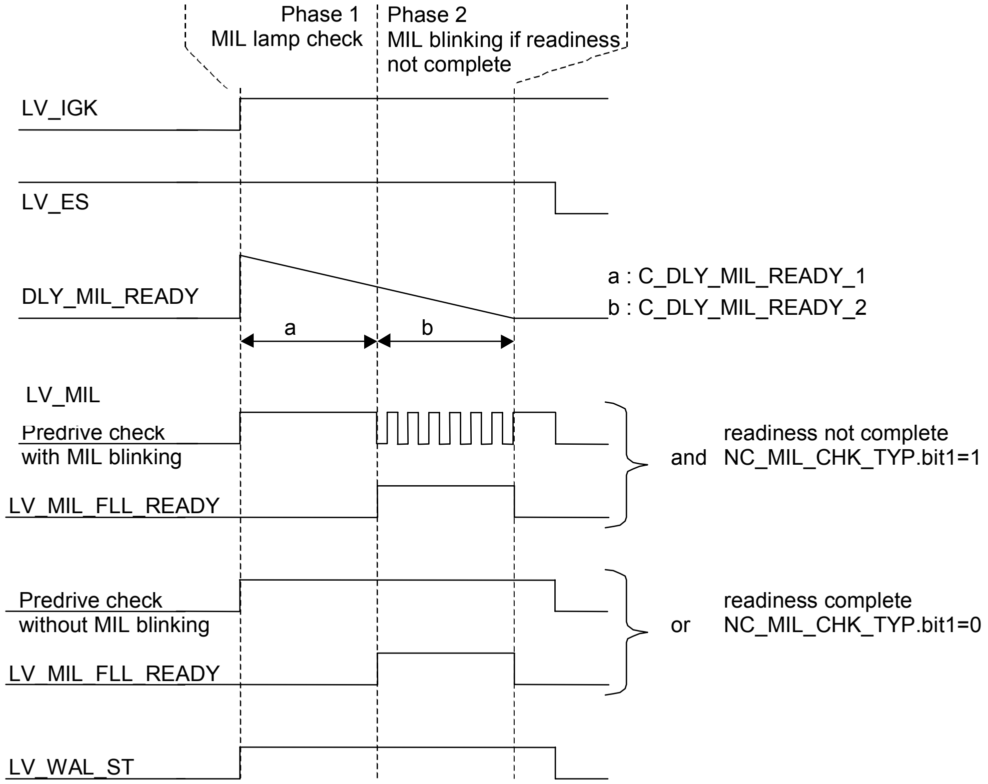
- or between key on transition LV_IGK 0->1 and engine start, plus between an engine stall event and engine start (bit 0 of NC_MIL_CHK_TYP=1)



B) Configuration of MIL blinking readiness status functionality:

with NC_MIL_CHK_TYP (bit 1)

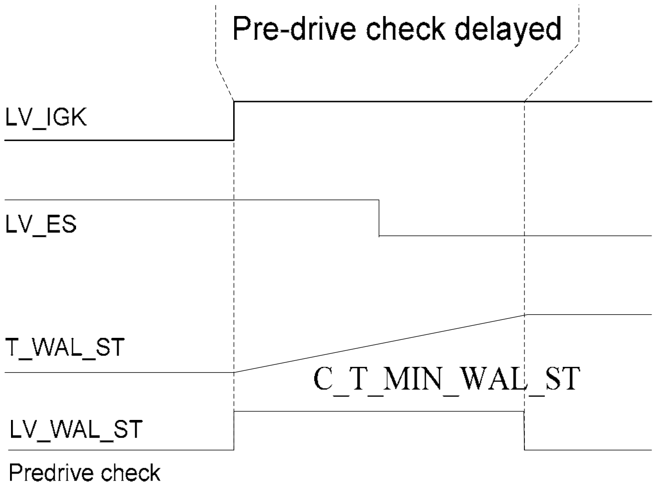
During predrive check mode, you can display the status of readiness codes through the MIL warning lamp : the MIL is blinking if readiness not complete.
This strategy is part of predrive check and does not affect the MIL status (STATE_MIL).



C) Management of minimum time of predrive check duration

-> only relevant after engine cranking (LV_ES 1->0)

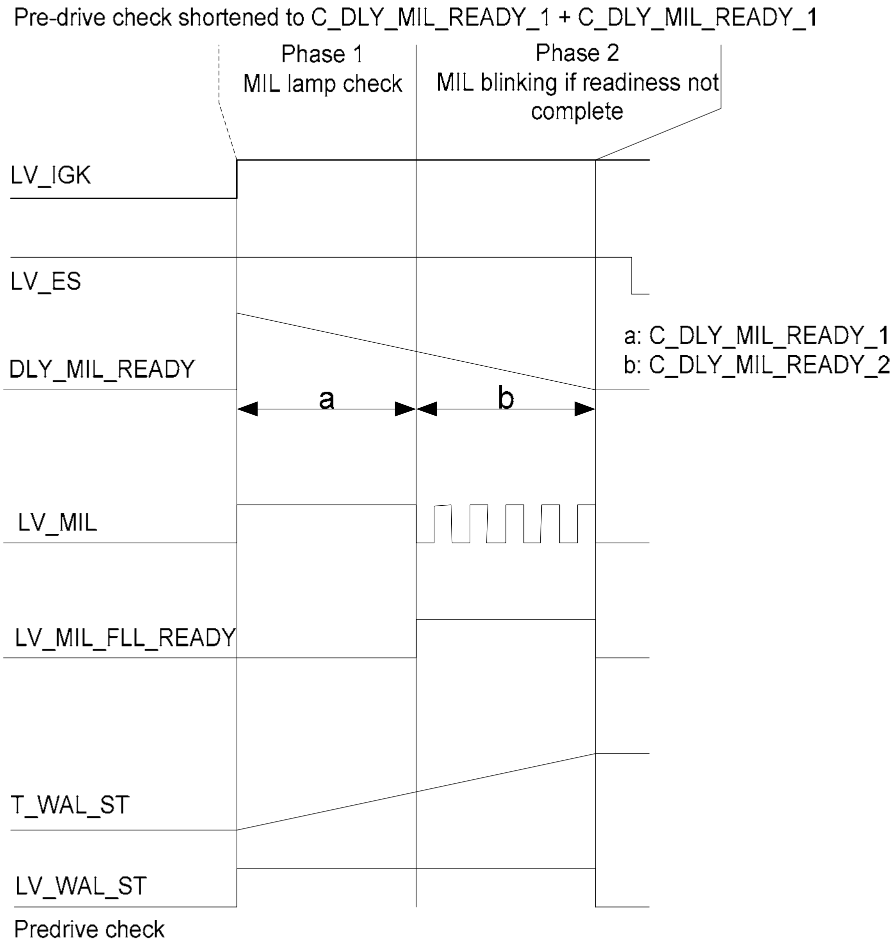
Wih the calibration C_T_MIN_WAL_ST it's possible to set a minimum time for the predrive check duration independant of engine cranking (LV_ES 1->0). This gives the possibility to illuminate the Mil for predrive check (e.g. in case of a quick start).



D) Management of maximum time of predrive check duration

-> only relevant before engine cranking (LV_ES = 0)

With the calibration LC_T_MAX_WAL_ST_ENA, it's possible to activated the maximum time of predrive check functionality. If this functionality is activated (LC_T_MAX_WAL_ST_ENA = 1) the maximum predrive check duration <u>before engine cranking</u> is C_DLY_MIL_READY_1 + C_DLY_MIL_READY_2. This gives the possibility to finish the predrive check before engine cranking e.g. in case of a long key on engine stop phase.



<u>Application conditions:</u>

Initialization: At LV_IGK transition 0 to 1
T_WAL_ST = 0 **and** LV_MIL_FLL_READY=0

At LV_ES transition 0 to 1
T_WAL_ST = 0

At LV_IGK transition 1 to 0 or if reset
LV_WAL_ST = 0 (Pre-drive check mode deactivated)

Recurrence: 500 ms and at LV_IGK 0 -> 1 transition

Activation: (pre-drive check enable between “stalling event/engine start”)

bit 0 of NC_MIL_CHK_TYP=1
and LV_IGK = 1 **and** LV_ES = 1

and (T_WAL_ST < C_DLY_READY_1 + C_DLY_READY_2

or LC_T_MAX_WAL_ST_ENA = 0)

and (bit 0 of C_CONF_WAL_1 = 0

or bit 0 of C_CONF_WAL_2 = 0

or bit 0 of C_CONF_MIL = 0)

or

(pre-drive check disable between “stalling event/engine start”)

bit 0 of NC_MIL_CHK_TYP=0
and transition LV_IGK 0 ->1 **and** LV_ES = 1

and (T_WAL_ST < C_DLY_READY_1 + C_DLY_READY_2

or LC_T_MAX_WAL_ST_ENA = 0)

and (bit 0 of C_CONF_WAL_1 = 0

or bit 0 of C_CONF_WAL_2 = 0

or bit 0 of C_CONF_MIL = 0)

Deactivation: **(**LV_ES = 0 (engine not stopped)

and T_WAL_ST >= C_T_ MIN_WAL_ST**)**

(minimum time of predrive check reached)

or

((LV_ES = 1 (engine stopped)

and T_WAL_ST >= C_DLY_READY_1 + C_DLY_READY_2

and LC_T_MAX_WAL_ST_ENA = 1**)**

(maximum time of predrive check reached)

or LV_IGK=0 (or key off)

Formula section:

LV_WAL_ST=1 during activation of this formula section.
LV_WAL_ST=0 and LV_MIL_FLL_READY=0 during deactivation of this formula section (end of predrive check).

T_WAL_ST = T_WAL_ST + 0.5 s

If bit 1 of NC_MIL_CHK_TYP = 1 (**if readiness display enable**)
Then

If transition of LV_WAL_ST from 0 to 1 (on beginning of predrive check)

```
    Then                                                            (initialize time delay)
            Initialize Delay time DLY_MIL_READY with
            C_DLY_MIL_READY_1+C_DLY_MIL_READY_2
    Endif
    DLY_MIL_READY = DLY_MIL_READY - 0.5 s

                                            (check if delay is within the timing window)

    If              DLY_MIL_READY < C_DLY_MIL_READY_2
            And     DLY_MIL_READY > 0
    Then
                                                                                        (ch

        If          (bit 4 to 7 of STATE_READY_OBD_1=0)
            And     (STATE_READY_OBD_2=0)

        Then
            LV_MIL_FLL_READY=0                      (if "complete" then MIL does not blink)
        Else
            LV_MIL_FLL_READY=1                  (if "not complete" then MIL lamp shall blink)
        Endif

    Else
            LV_MIL_FLL_READY = 0            (the readiness status phase is finished)
    Endif

Endif

If      bit 0 of C_CONF_MIL = 0             (MIL not inhibit)
and     bit 3 of C_CONF_MIL = 0             (predrive check for MIL enable)
and     LV_MIL_FLL_READY = 1                (readiness not complete)
Then
        LV_MIL shall blink at 1Hz
Endif
```

(“**predrive check**”: **switch on warning lamps**)

```
If    bit 0 of C_CONF_WAL_1 = 0           (WAL_1 not inhibit)
and        bit 3 of C_CONF_WAL_1 = 0      (predrive check for WAL_1 enabled)
Then       LV_WAL_1 = 1
Endif
If    bit 0 of C_CONF_WAL_2 = 0           (WAL_2 not inhibit)
and        bit 3 of C_CONF_WAL_2 = 0      (predrive check for WAL_2 enabled)
Then       LV_WAL_2 = 1
Endif
If    bit 0 of C_CONF_MIL = 0             (MIL not inhibit)
and        bit 3 of C_CONF_MIL = 0        (predrive check for MIL enabled)
and        LV_MIL_FLL_READY = 0           (MIL not blinking readiness status)
Then       LV_MIL = 1
Endif
```

27.21.2.2 Normal mode

The normal mode is defined as “not to be in pre-drive check mode”.

27.21.2.3 All modes

Description:

The following treatment is done in pre-drive check mode and in normal mode.

Application conditions:

Initialization: at transition LV_IGK 1->0 or if reset

LV_MIL = 0 (MIL is off)

LV_WAL_1 = 0 (WAL_1 is off)

LV_WAL_2 = 0 (WAL_2 is off)

Recurrence: 500 ms and at LV_IGK 0 -> 1 transition

Activation: LV_IGK=1

Formula section:

```
If      STATE_MIL = MIL_FLL
Then LV_MIL blink at 1Hz
Else  If     (STATE_MIL = ON or (LV_MIL_ACT_REQ = 1 and bit 1 of C_CONF_MIL = 1))
                                         (allowed MIL request)
             AND (LV_MIL_FLL_READY=0 or bit 3 of C_CONF_MIL = 1)
             (case MIL blinking during the pre-drive check: readiness not completed)
```

```
        Then LV_MIL = 1
        Else  If        we are in normal mode or bit 3 of C_CONF_MIL = 1
                        (predrive check not authorized for MIL)
              Then      LV_MIL = 0
              Endif
        Endif
Endif

If      STATE_WAL_1 = ON or (LV_WAL_1_ACT_REQ = 1 and bit 1 of C_CONF_WAL_1 =
1)
                                                    (allowed lamp WAL_1 request)

Then LV_WAL_1 = 1
Else  If        we are in normal mode or bit 3 of C_CONF_WAL_1 = 1
                (pre drivecheck not authorized for WAL_1)
      Then LV_WAL_1 = 0
      Endif
Endif

If      STATE_WAL_2 = ON or (LV_WAL_2_ACT_REQ = 1 and bit 1 of C_CONF_WAL_2 =
1)
                                                    (allowed lamp WAL_2 request)

Then LV_WAL_2 = 1
Else  If        we are in normal mode or bit 3 of C_CONF_WAL_2 = 1
                (pre drivecheck not authorized for WAL_2)

      Then LV_WAL_2 = 0
      Endif
Endif
```

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_MIL_CHK_TYP	-	0...3H	0...3	1	[-]
Bit 0 = 0 : disable pre-drive check between "stalling event/engine start"Bit 0 = 1 : enable pre-drive check between "stalling event/engine start"Bit 1 = 0 : disable readiness status display during pre-drive check (MIL blinking)Bit 1 = 1 : enable readiness					

Configuration data detailed description:

Bit 0 = 0 : disable pre-drive check between “stalling event/engine start”
Bit 0 = 1 : enable pre-drive check between “stalling event/engine start”
Bit 1 = 0 : disable readiness status display during pre-drive check (MIL blinking)
Bit 1 = 1 : enable readiness
Typical value bit 0 = 0, bit 1 = 0

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CONF_MIL	1	0...FFH	0...255	1	[-]
Configuration of MIL					
C_CONF_WAL_1	1	0...FFH	0...255	1	[-]
Configuration of warning lamp 1 (WAL_1)					
C_CONF_WAL_2	1	0...FFH	0...255	1	[-]
Configuration of warning lamp 2 (WAL_2)					
C_DLY_MIL_READY_1	1	0...FFH	0...127.5	0.5	[s]
Delay to start MIL blinking for readiness display during pre-drive checkShall be in between 15 and 20 sec. Typical value 20sec					
C_DLY_MIL_READY_2	1	0...FFH	0...127.5	0.5	[s]
MIL blinking duration for readiness display during pre-drive checkShall be in between 5 and 10 sec. Typical value 10sec					
C_DLY_INH_IV_MIS	1	0...FFH	0...127.5	0.5	[s]
Delay for MIL blinking in case of misfiring present failure with cylinder cut off					
C_T_MIN_WAL_ST	1	0...FFH	0...127.5	0.5	[s]
Minimum time of pre-drive check duration					
LC_T_MAX_WAL_ST_ENA	1	0...1H	0...1	1	[-]
Activation of maximum time of pre-drive check					

Calibration data detailed description:

C_DLY_MIL_READY_1: Delay to start MIL blinking for readiness display during pre-drive check. Shall be tuned between 15 s and 20 s. Typical value 20 s.

C_DLY_MIL_READY_2: MIL blinking duration for readiness display during pre-drive check. Shall be tuned between 5 s and 10 s. Typical value 10 s.

LC_T_MAX_WAL_ST_ENA: = 1: maximum time of predrive check with LV_ES = 1 is C_DLY_MIL_READY_1 + C_DLY_MIL_READY_2

= 0: maximum time of predrive check with LV_ES = 1 is unlimited

27.22 Lamp Management (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MIL_ACT_REQ	V/O	0...1H	0...1	1	[-]
Request of MIL activation by others					
LV_WAL_1_ACT_REQ	V/O	0...1H	0...1	1	[-]
Request of WAL 1 activation by others					
LV_WAL_2_ACT_REQ	V/O	0...1H	0...1	1	[-]
Request of WAL 2 activation by others					

Input data:

STATE_ETCU_OBD	LV_IGK	LV_MIL_REQ_ETCU	

27.22.1 External request management

FUNCTION DESCRIPTION:

General information:

The external request management of all lamps is done directly on the CAN output-signal, thus this function is now only used by the EGS - ECU.

Application conditions:

Activation:	LV_IGK = 1
Deactivation:	LV_IGK = 0 then set all outputs equal to 0
Initialization:	At ECU reset: LV_MIL_ACT_REQ = 0 LV_WAL_1_ACT_REQ = 0 LV_WAL_2_ACT_REQ = 0
Recurrence:	100 ms

Formula section:

```
If   LV_MIL_REQ_ETCU = 1             or
     [ LC_MIL_ACT_REQ_ETCU = 1       and
      (STATE_ETCU_OBD = 0100 b       or
       STATE_ETCU_OBD = 0110 b)      ]
Then       LV_MIL_ACT_REQ = 1
Else       LV_MIL_ACT_REQ = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_MIL_ACT_REQ_ETCU	1	0...1H	0...1	1	[-]
logical constant for MIL illumination by ETCU					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_LAMP	1	0...3H	0...3	1	[-]
Quantity of warning lamps					

27.23 Priority rules

Input data:

PRI_CONF_XX	WAL_CONF_XX	LV_ERR_DISA[NC_NR_ERR_DYN]	LV_DC_MAX[NC_NR_ERR_DYN]
LV_ERR_CFM[NC_NR_ERR_DYN]	NC_ERR_PRI_H		

Export actions:

ACTION_ERRM_PrioRule (OUT <RESP>, OUT <IDX>)
This action performs priority rules, when failure memory is full, to say if the new failure is accepted or not.

FUNCTION DESCRIPTION:

General information:

The dynamic error management module calls this module in case of present failure occurrence when the 2nd layer memory is full.

It defines criteria to store or not this failure in the 2nd layer memory (dynamic memory). Only one failure occurrence is stored.

Results of this module is either:

- "storage of new failure not possible"
- or ("storage of new failure possible" ; index of failure to delete in 2nd layer memory)

In case of new failure entry whereas the dynamic memory is full (number of stored failure = NC_NR_ERR_DYN), different criteria are applied to prioritize the failure:

- A new failure is more important than a disappeared failure
- Failure priority as defined in PRI_CONF_XX ;
    - Chronological failure order

      If NLC_OLD_ERR_PRI = 1: old failure has priority regards of new failure;

      (Failure i older than failure j means failure i has entered 2nd layer memory before failure j)

      If NLC_OLD_ERR_PRI = 0: new failure has priority regards of old failure;

      (Failure i newer than failure j means failure i has entered 2nd layer memory after failure j)

Additional features of this functionality:

- MIL can never be switched off
- Failure setting the OBD freeze frame (e.g. used for diagnostic communication with scantool / mode 2h) cannot be erased.
- The "increased priority" can be enabled with the configuration NLC_INC_ERR_PRI. This priority is a final rule, which is used, if it's not possible to enter a failure with any of the other possibilities. Then it's possible (only for failure with priority >= 4) to store the failure, if a confirmed failure not of the highest priority (priority < NC_ERR_PRI_H), which is not illumination the MIL anymore, can erased instead.

If the new failure entry is not accepted because of the priority, this failure will not be stored in the 2nd layer memory but limp home is activated, thus the freeze frame is not memorized and also no lamp management for this failure is possible.

Description:

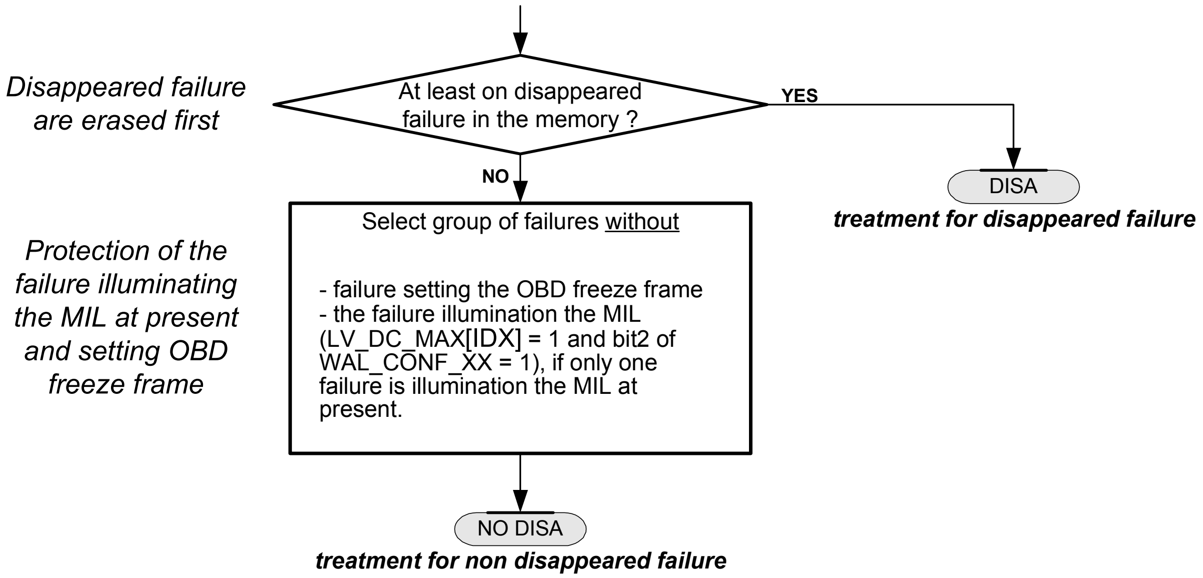
Syntax :	ACTION_ERRM_PrioRule (OUT <RESP>, OUT <IDX>)	
Parameter (in) :	-	
Parameter (out) :	RESP	OK if storage is possible; NOK if storage is not possible
	IDX	Index of failure in 2nd layer memory to delete
Short description :	This action performs priority rules, when failure memory is full, to say if the new failure is accepted or not.	

Application conditions:

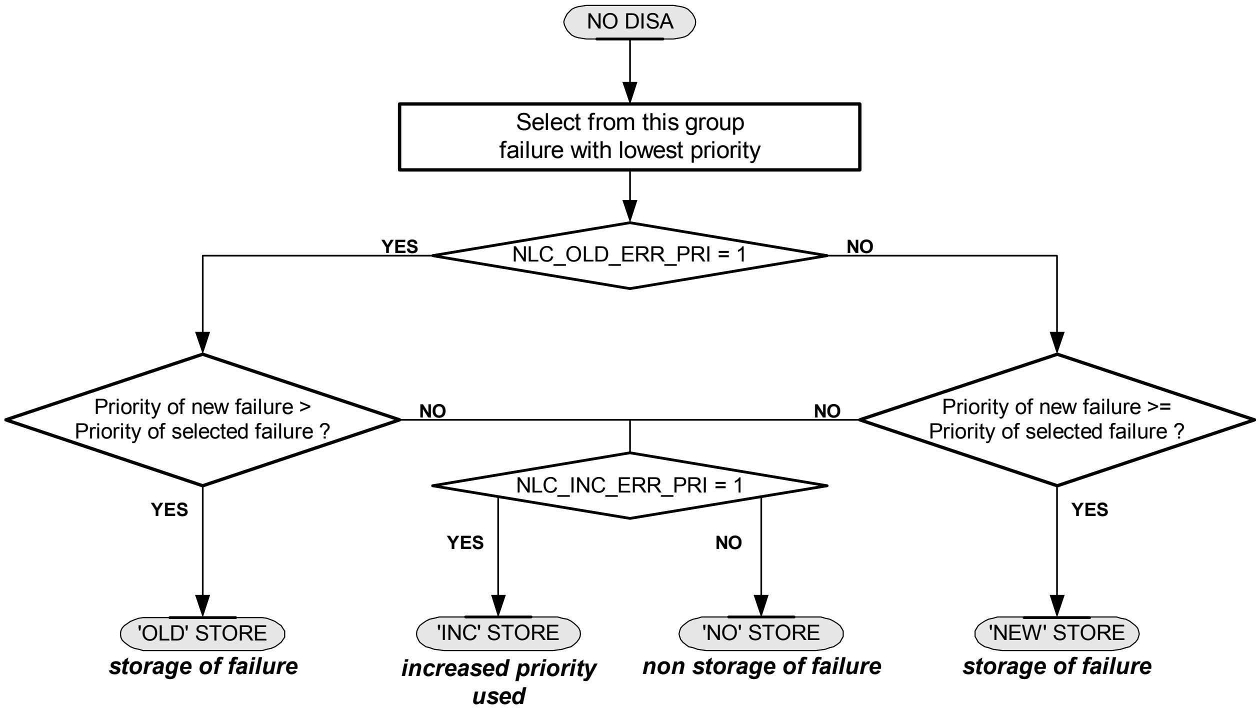
Activation:	-
Recurrence:	-
Deactivation:	at action request

Formula section:

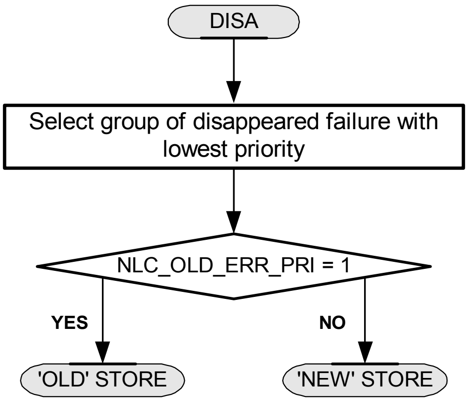
ACTION_ERRM_PrioRule (RESP, IDX) :



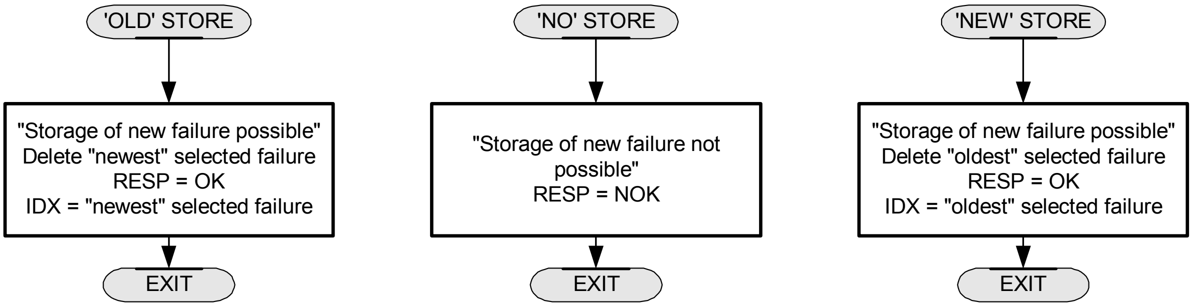
Treatment for non disappeared failure:



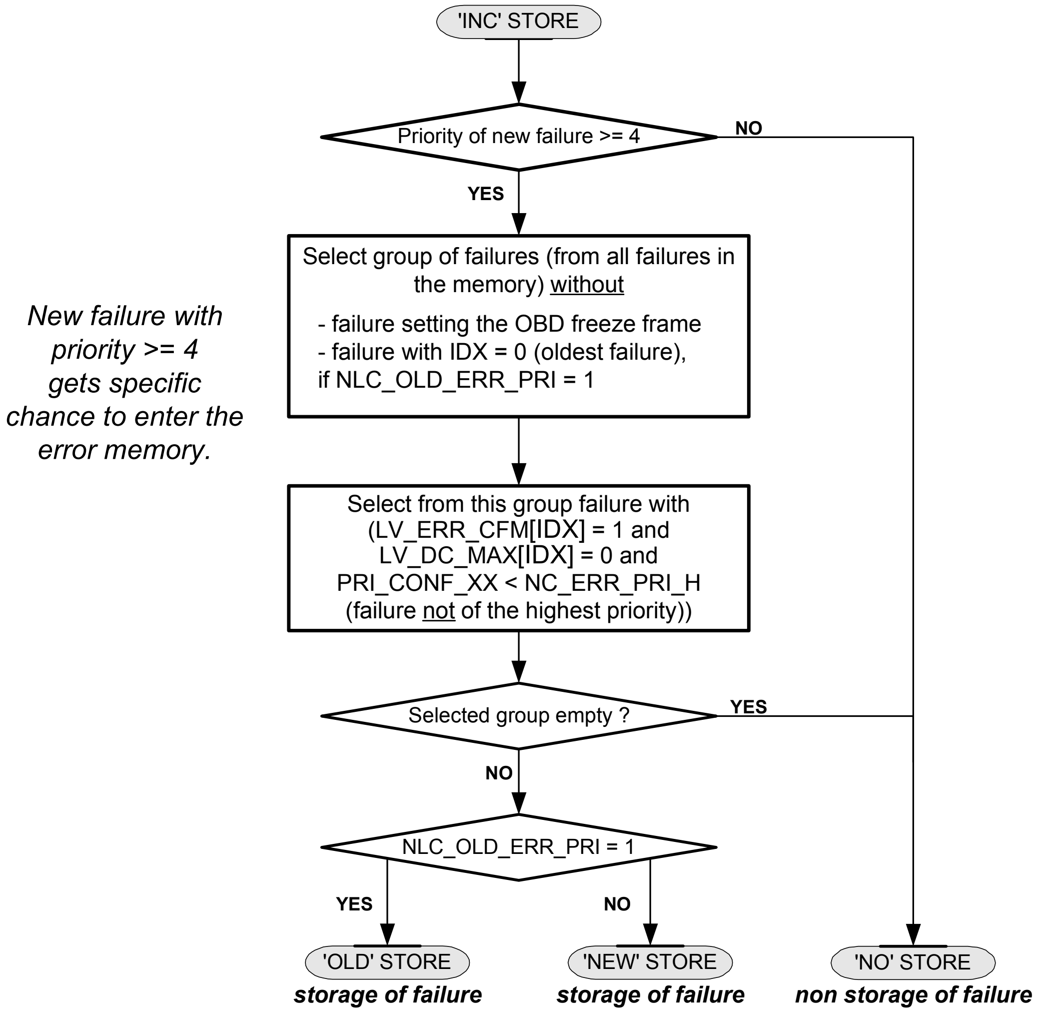
Treatment for disappeared failure:



Storage / non storage of failure:



Increased priority (with NLC_INC_ERR_PRI = 1):



Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_OLD_ERR_PRI	1	0...1H	0...1	1	[-]
Selection of old failure priority face to new failure					
NLC_INC_ERR_PRI	1	0...1H	0...1	1	[-]
Enable/disable increased failure priority					

Configuration data detailed description:

NLC_OLD_ERR_PRI : Set this bit to 1 to give priority to old failure regarding new failure.
Set this bit to 0 to give priority to new failure regarding old failure.

NLC_INC_ERR_PRI : Set this bit to 1 to enable increased failure priority.
Set this bit to 0 to disable increased failure priority.

27.24 Readiness codes

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_READY_XX	V/O/S	0...1H	0...1	1	[-]
Readiness flag related to diagnosis XX					

Input data:

LV_END_DIAG_XX	LV_ERR_DC[NC_NR_ERR_DYN]	LV_DC_MAX[NC_NR_ERR_DYN]	LV_DC
LV_ERR_MEM_XX	LV_EOL_OBD_DC	LV_ERR_XX	

Export actions:

ACTION_ERRM_InitReadiness ( )
This action initialise all the readiness flag

FUNCTION DESCRIPTION:

General information:

Readiness flag allows to know if a full diagnostic check (not in EOL phase) has been done or not (minimum number of checks necessary for MIL illumination if failure is present).

A readiness flag is defined for each diagnosis (LV_READY_XX). Based on these flags, readiness code status (2 bytes STATE_READY_OBD_1 and STATE_READY_OBD_2) are generated in the Communication Interface (API) file for Carb mode.

Readiness flags are mainly used by after market service and inspections maintenance. When after market achieves a repair, the system should be controlled before to give back the car to its owner.

Because some failure need a long time (and may be 2 DC) to be diagnosed, the repair operator need an as quick as possible information meaning the repairs is OK.

The readiness flag indicates when it is set to 0 that :

- the diagnosis related to failure XX is done since the last “clear DTC” service received.
- and if a failure XX is present, MIL would be illuminated.

For after market time optimisation, in case of no failure at all, the readiness flag is set to 0.

The readiness flag indicates when it is set to 1 that :

- the diagnosis related to failure XX is not done since the last “clear DTC” service received.
- or if a failure XX is present, MIL is not still illuminated (may takes many driving cycle).

After failure erase, all bits associated to supported functions are set to 1.

When the system has executed enough driving cycles for determining if MIL should be switch On or not, all the readiness flags are in 0 state. At this occurrence, failures which may have switch on the MIL could be considered as repaired if MIL is Off.

27.24.1 Readiness flag set to 0

Description:

The readiness flag LV_READY_XX is set to 0 when enough driving cycle has been done without failure XX.

Application conditions:

Initialization: On saved ram lost

LV_READY_XX = 1

Recurrence: at the recurrence of activation condition

Activation: at transition LV_DC 0->1
or at transition LV_END_DIAG_XX 0->1

Formula section:

```
If    LV_READY_XX = 1                          (diag XX not still ready)
      and LV_DC = 1                            (driving cycle valid)
            and LV_END_DIAG_XX = 1                       (diag done)
            and LV_EOL_OBD_DC = 0                        (not in EOL tests)
Then        If      LV_ERR_MEM_XX = 1          (failure in 2nd layer memory)
            Then    If      LV_ERR_DC[IDX] = 0 (no failure present this DC)
                    Then    LV_READY_XX = 0    (diag XX is ready)
                    Else    If      LV_DC_MAX[IDX] = 1
                            Then    LV_READY_XX = 0   (diag XX is ready)
                            Endif
                    Endif
            Else    If      LV_ERR_XX = 0
                    Then    LV_READY_XX = 0    (diag XX is ready)
                    Endif
            Endif
Endif
```

27.24.2 Readiness flag reinitialisation (set to 1)

Description:

Syntax : ACTION_ERRM_InitReadiness ( )

Parameter(in) : No parameter

Parameter(out) : No parameter

Short description :

The readiness flag LV_READY_XX is reinitialized (set to 1) upon communication tool request via the action ACTION_ERRM_InitReadiness ( ).

This reinitilisation is always done when all failure are erased from failure memory.

Application conditions:

Initialization: -

Recurrence: -

Activation: at action request

Formula section:

For every failure : LV_READY_XX = 1

27.24.3 Readiness code information update

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_READY_OBD_1	V/O/S	0...FFH	0...255	1	[-]
Readiness code completion status 1					
STATE_READY_OBD_2	V/O/S	0...FFH	0...255	1	[-]
Readiness code completion status 2					
STATE_READY_OBD_3	V/O/S	0...FFH	0...255	1	[-]
Readiness code completion status 3					

Input data:

LV_DC			

Import actions:

ACTION_ERRM_ReadReadinessCode(INOUT<ReadinessCode>, OUT<ResultReadinessCode>)
ACTION_ERRM_ReadReadinessCodCus(INOUT<ReadinessCode>, OUT<ResultReadinessCode>)

Description:

Purpose of this paragraph is to update readiness code completion status. This update shall be done :

- at end of driving cycle, for readiness status display,
- on action call, for customer tool and Scan Tool coherency with STATE_READY_OBD_X.

Important remark: each time ACTION_ERRM_ReadReadinessCode is called Readiness codes completion status (STATE_READY_OBD_1 and STATE_READY_OBD_2) are calculated.

Application conditions:

Activation: at LV_DC 1→ 0 transition

Initialization: on non-volatil memory reset/lost **or** on failure erase service received

STATE_READY_OBD_1 = FFh,

STATE_READY_OBD_2 = FFh
{FFh value indicates that readiness code is not yet computed}

Recurrence: none, executed single time on activation.

Formula section:

{ Action call to update STATE_READY_OBD_1 and STATE_READY_OBD_2 }

Call ACTION_ERRM_ReadReadinessCode(INOUT<STATE_READY_OBD_1, STATE_READY_OBD_2>, OUT<ResultReadinessCode>)

Application conditions:

Activation : at LV_DC 1→ 0 transition

Initialization: on non-volatil memory reset/lost

or

on failure erase service received
STATE_READY_OBD_3 = FFh
{ *FFh value permits to indicate that readiness is not yet computed* }

Recurrence : none, executed single time on activation.

Formula section:

{ Action call to update STATE_READY_OBD_3 }

Call ACTION_ERRM_ReadReadinessCodCus(INOUT<STATE_READY_OBD_3>, OUT<ResultReadinessCode>)

27.25 Similar conditions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_SCDN[NC_NR_WIN_SCDN]	V/S	0...3FH	0...63	1	[-]
Similar condition status					
CTR_SCDN_EQU_DC[NC_NR_WIN_SCDN]	V/S	0...FFH	0...255	1	[-]
Counter of non occurrence of similar condition					
N_MAX_SCDN_EQU[NC_NR_WIN_SCDN]	V/S	0...1FE0H	0...8160	1	[rpm]
maximal engine speed of similar conditions window					
N_MIN_SCDN_EQU[NC_NR_WIN_SCDN]	V/S	0...1FE0H	0...8160	1	[rpm]
minimal engine speed of similar conditions window					
LOAD_MAX_SCDN_EQU[NC_NR_WIN_SCDN]	V/S	0...FFH	0...99.60937	0.390625	[%]
maximal load of similar conditions window					
LOAD_MIN_SCDN_EQU[NC_NR_WIN_SCDN]	V/S	0...FFH	0...99.60937	0.390625	[%]
minimal load of similar conditions window					
LV_WUP_SCDN_EQU[NC_NR_WIN_SCDN]	V/S	0...1H	0...1	1	[-]
Warm-up status for similar conditions for OBDII continuous error					
CTR_SCDN_EQU[NC_NR_WIN_SCDN]	V	0...FFFFH	0...65535	1	[-]
Number of times N/LOAD is within SCDN window					
CTR_SCDN_SUM[NC_NR_WIN_SCDN]	V	0...FFFFH	0...65535	1	[-]
Number of times the diagnosis condition are set					

STATE_SCDN[k] : Similar conditions status for diagnostic instance k

bit 0:	0: SCDN will be recorded (updated) at next failure occurrence 1: SCDN are recorded
bit 1:	Recognition of similar conditions with failure detection this DC
bit 2:	Recognition of similar condition without failure/exceedance detection this DC It’s also set after C_CTR_SCDN_EQU_DC_MAX driving cycles without similar conditions without error
bit 3:	Failure XX was present last DC
bit 4:	Request to increment driving cycle counter of failure XX
bit 5:	" exceedance" of specific limit was reported for failure XX

Input data:

LV_DC	LV_STATE_WUP	N	LOAD_SCDN
LC_ENA_SCDN	NLC_ENA_SCDN	LV_ERR_DC[NC_NR_ERR_DYN]	CTR_DC[NC_NR_ERR_DYN]
LV_CDN_DIAG_XX			

Export actions:

ACTION_ERRM_DecrementDCctrScdn (OUT <DCdec>, IN <XX>)
This action calculates the possibility to decrement the driving cycle counter related to failure XX
ACTION_ERRM_EraseScdn(IN <XX>)
This action erase the similar conditions related to failure XX (if existing)
ACTION_ERRM_CdnDiagScdn (IN <XX>)
This action is used to calculate a ratio to recognize the similar condition without failure
ACTION_ERRM_EndWinScdn (IN <XX>, IN <EXC>)
This is used to store similar conditions and recognize similar condition with or without failure

Import actions:

ACTION_ERRM_IncrementDCctrScdn (IN <XX>)

FUNCTION DESCRIPTION:

General information:

The different states of a failure are managed regards their driving cycle and warm up cycle counters values (see Dynamic error management core for details).

For Misfire and Fuel system failures, some additional conditions should be fulfilled to increment and decrement their driving cycle counter. These additional conditions are related to engine conditions and are called similar conditions.

When similar conditions without failure are fulfilled for the failure XX already stored in the dynamic memory in a previous driving cycle, bit 2 of STATE_SCDN[k] is set to 1. That means, that the driving cycle counter of failure XX could be decremented at the end of the driving cycle if the failure XX will not get present till this treatment. If an exceedance is reported by the diagnostic function (bit 5 of STATE_SCDN[k] is set to 1), the driving cycle counter is not decremented at the end of the DC. Exception: Driving cycle counter decrementation linked with C_CTR_SCDN_EQU_DC_MAX DCs, see description below.

Also the incrementation of the driving cycle counter of a failure stored in dynamic memory is controlled by the similar conditions functionality. The incrementation done in the error management core is triggered by the ACTION_ERRM_IncrementDCctrScdn (XX) in the following cases:

- Similar conditions are recorded

  (bit 0 of STATE_SCDN[k] 0->1).
- A failure already stored in memory occurs again in the following driving cycle (bit 2 of STATE_SCDN[k] = 1).
- A temporary failure or a confirmed failure with MIL off (LV_DC_MAX[IDX] = 0) occurs again not in the following driving cycle but with similar conditions with failure (bit 1 of STATE_SCDN[k] = 1).
- A confirmed failure with MIL on (LV_DC_MAX[IDX] = 1) occurs again.

In all of these cases, the driving cycle counter is incremented (bit 4 of STATE_SCDN[k] 0→1).

For a temporary failure or confirmed failure with MIL off (LV_DC_MAX[IDX] = 0), even if similar conditions are not recognized during a C_CTR_CDN_EQU_DC_MAX number of driving cycles, the driving cycle counter of the failure can be decremented after C_CTR_CDN_EQU_DC_MAX driving cycles.

Similar conditions functionality is inhibited when LC_ENA_SCDN=0 or NLC_ENA_SCDN=0.

Remark:

To keep the link between the index of the failure using the similar condition and the index of the failure in static memory, a transfer function is defined: k = Scdn (XX). The link between the diagnosis instance and the similar conditions instance k is described in the application incidence of similar condition.

27.25.1 Ratio calculation to recognize the similar condition without failure

Description:

For the recognition of similar conditions the current engine speed / load conditions are compared with the recorded similar conditions window.

Based on two counters calculated with this algorithm the SCDN recognition at end of the diagnostic window is done. (see next chapter)

- CTR_SCDN_EQU[k]

  counting the number of times engine speed / load conditions are within the recorded similar conditions window (with LV_CDN_DIAG_XX = 1)

- CTR_SCDN_SUM[k]

  counting the number of times the diagnosis conditions are set (LV_CDN_DIAG_XX = 1)

The calculation of the counters is stopped, if an exceedance is reported from diagnostic function.

Syntax : ACTION_ERRM_CdnDiagScdn (IN <XX>)

Parameter (in) : XX failure XX using similar condition instance XX

Parameter (out) : -

Short description:

This action is called at each diagnosis condition. It is used to calculate a ratio to recognize the similar condition with or without failure.

Application conditions:

Initialization: at transition LV_DC 0 -> 1 or reset

bit 0, bit 3 of STATE_SCDN[k] = restored from NVMY
bit 1, bit 2, bit 4, bit 5 of STATE_SCDN[k] = 0

CTR_SCDN_EQU[k] = 0

```
CTR_SCDN_SUM[k]                              = 0
```

Recurrence:

Activation: At action request **and** LV_DC = 1

and LC_ENA_SCDN = 1 and NLC_ENA_SCDN = 1
(similar conditions enable)

Formula section:

```
k = Scdn (XX)
If  bit 0 of STATE_SCDN[k]    = 1   (similar conditions are recorded for failure XX)
and bit 5 of STATE_SCDN[k]    = 0   (no exceedance reported from diagnostic function)
Then    If     LV_CDN_DIAG_XX = 1
        Then   CTR_SCDN_SUM[k] ++
               (check if current N, LOAD_SCDN point is within SCDN window)
               If     N <= N_SCDN_MAX_EQU[k]
               and    N >= N_SCDN_MIN_EQU[k]
               and    LOAD_SCDN <= LOAD_SCDN_MAX_EQU[k]
               and    LOAD_SCDN >= LOAD_SCDN_MIN_EQU[k]
               Then   CTR_SCDN_EQU[k] ++
               Endif
Endif
```

27.25.2 Record and recognition of similar conditions at end of diagnosis window

Description:

The similar conditions are stored for a failure XX (or updated) in the following cases:

- The similar conditions are frozen within the window N/LOAD_MAX/MIN_SCDN_EQU[k], if the corresponding failure is stored in the memory (failure gets present the first time)
- The similar conditions are stored again (updated) within the window N/LOAD_MAX/MIN_SCDN_EQU[k], if the corresponding failure occurs again and the driving cycle counter of this failure was 0 before.

In both cases bit 0 of STATE_SCDN[k] is 0 before the storage.

The recognition of similar conditions for detected failure is done by comparing, if the actual N/LOAD point is within the stored SCDN window. If also the same warmup status is reached at this time, similar conditions with failure are recognized. This calculation is done at the action call depending of the monitoring function, only in the following case:

- Similar conditions are stored in a previous driving cycle (bit 0 of STATE_SCDN[k] = 1)
- and the same failure XX was not present the last driving cycle. In case of same failure present last driving cycle, the driving cycle counter is incremented without asking for similar conditions.

- and failure is not illuminating the MIL. In case failure is illuminating the MIL at present the driving cycle counter is set to the maximum without asking for similar conditions.
- and driving cycle counter not already incremented in this driving cycle

If no error is present at the end of the diagnosis window, the ratio (CTR_SCDN_EQU[k] / CTR_SCDN_SUM[k]) is compared with the threshold C_CTR_SCDN_NOT_ERR_THD[k]. If the threshold and the same warmup status are reached at this time, similar conditions without failure are recognized.

This calculation is done at the end of a diagnosis window, only if no failure XX is present in the following cases:

- Similar conditions are already recorded for failure XX (bit 0 of STATE_SCDN[k] = 1) in a previous driving cycle
- Similar conditions were not already encountered within this driving cycle with no failure
- no exceedance reported from diagnostic function (bit 5 of STATE_SCDN[k] = 0)
- Driving cycle counter was not incremented this driving cycle yet.

Syntax :	ACTION_ERRM_EndWinScdn (IN <XX>, <EXC>)	
Parameter (in) :	XX	failure XX using similar condition instance XX
	EXC	Exceedance reported from Diagnostic function
Parameter (out) :	-	

Short description:

This action is called at each end of diagnosis windows if there's no failure or when the failure gets present or when an exceedance is recognized by the diagnostic function. This usage of this function depends on the diagnostic function. It is used for recognition of similar condition with or without failure

Application conditions:

Initialization: -

Recurrence: -

Activation: At action request **and** LV_DC = 1
and LC_ENA_SCDN = 1 **and** NLC_ENA_SCDN = 1
(similar conditions enable)

Formula section:

```
k = Scdn(XX)
If      EXC = 1     (Exceedance reported from Diagnostic function)
Then    bit 5 of STATE_SCDN[k] = 1
Endif
If      bit 4 of STATE_SCDN[k] = 0          (request to increment driving cycle
                                             counter of failure XX not made)
and     LV_ERR_XX = 1                        (failure present)
```

```
Then  If      bit 0 of STATE_SCDN[k] = 1       (similar conditions recorded
                                               for failure XX)

      Then  If   bit 3 of STATE_SCDN[k] = 1
                      (failure present the last driving cycle)
            or        LV_DC_MAX[IDX] = 1
                           (failure is confirmed with MIL on)
            Then      ACTION_ERRM_IncrementDCctrScdn (XX)
                      (request to increment driving cycle counter for failure XX)
                           bit 4 of STATE_SCDN[k] = 1
                           (request to increment driving cycle counter of was made)

            Else      (recognition of similar conditions with failure)
                      (check if ratio to recognize SCDN is reached)
                      If   N <= N_MAX_SCDN_EQU[k]
                      and  N >= N_MIN_SCDN_EQU[k]
                      and  LOAD_SCDN <= LOAD_MAX_SCDN_EQU[k]
                      and  LOAD_SCDN >= LOAD_MIN_SCDN_EQU[k]
                      and  LV_WUP_SCDN_EQU[k] = LV_STATE_WUP
                      Then bit 1 of STATE_SCDN[k] = 1
                           (similar conditions encountered with failure window)
                           bit 2 of STATE_SCDN[k] = 0
                           (similar conditions not encountered with no failure)
                                ACTION_ERRM_IncrementDCctrScdn (XX)
                           (request to increment driving cycle counter for failure XX)
                                bit 4 of STATE_SCDN[k] = 1
                                     (request to increment driving cycle counter
of was made)
                      Endif
            Endif

      Else  (SCDN engine speed/load point frozen)
              N_SCDN_MAX_EQU[k] = N + C_N_SCDN_EQU
              N_SCDN_MIN_EQU[k] = N - C_N_SCDN_EQU
              LOAD_SCDN_MAX_EQU[k] = LOAD_SCDN + C_LOAD_SCDN_EQU
              LOAD_SCDN_MIN_EQU[k] = LOAD_SCDN - C_LOAD_SCDN_EQU
              LV_WUP_SCDN_EQU[k] = LV_STATE_WUP     (warm-up status stored)
              CTR_SCDN_EQU_DC[k] = C_CTR_SCDN_EQU_DC_MAX
              bit0 of STATE_SCDN[k] = 1                (similar conditions recorded
                                                       for failure XX)
              ACTION_ERRM_IncrementDCctrScdn (XX)
              (request to increment driving cycle counter for failure XX)
```

```
                bit 4 of STATE_SCDN[k] = 1
                (request to increment driving cycle counter was made)
        Endif
Else (recognition of similar conditions without failure)
    If  bit 0 of STATE_SCDN[k]      = 1   (similar conditions recorded
                                           for failure XX)
    and bit 4 of STATE_SCDN[k]      = 0   (request to increment driving cycle
                                           counter of failure XX was not made)
    and bit 2 of STATE_SCDN[k]      = 0   (similar conditions not encountered
                                           with no failure)
    and LV_ERR_XX                   = 0   (no failure present)
    and bit 5 of STATE_SCDN[k]      = 0   (no exceedance reported from
                                           Diagnostic function)

    Then If CTR_SCDN_EQU[k] / CTR_SCDN_SUM[k]>=
         C_CTR_SCDN_NOT_ERR_THD[k]
         and  LV_STATE_WUP = LV_WUP_SCDN_EQU[k] (warm-up status)
         Then bit 2 of STATE_SCDN[k] = 1 (similar conditions encountered
                                          with no failure)
         Endif
    Endif
Endif


If bit 0 of STATE_SCDN[k] = 1     (similar conditions recorded for failure XX)
Then CTR_SCDN_EQU[k] = 0
     CTR_SCDN_SUM[k] = 0
Endif
```

27.25.3 SCDN storage and recognition for detected failure performed last DC

Description:

According regulation the information, if a failure XX (using SCDN functionality) was present the last driving cycle, must be respected to increment the DC counter. In case of a new failure occurrence, if the same failure was present the last driving cycle, the driving cycle counter must be incremented (if not already incremented this driving cycle) independently of similar conditions.

Application conditions:

Initialization: -

Recurrence:

Activation: At LV_DC transition 1 -> 0 **and**

LC_ENA_SCDN = 1 **and** NLC_ENA_SCDN = 1
(similar conditions enable)

Formula section:

```
For k = 0 to NC_NR_WIN_SCDN - 1
    k = Scdn(XX); XX = F(IDX)
        If    LV_ERR_DC[IDX]     = 1      (failure present in this DC)
        Then         bit 3 of STATE_SCDN[k] = 1
            (failure present the last driving cycle)
            CTR_SCDN_EQU_DC[k] = C_CTR_SCDN_EQU_DC_MAX
            (re-initialization of similar conditions counters)
        Else         bit 3 of STATE_SCDN[k] = 0
            (failure not present the last driving cycle)
        Endif
EndFor
```

27.25.4 SCDN usage to decrement driving cycle counter

Description:

Syntax :	ACTION_ERRM_DecrementDCctrScdn (OUT <DCdec>, IN <XX>)	
Parameter (in) :	XX	number of failure to store in dynamic memory
Parameter (out) :	DCdec	YES if decrementation of DC counter is possible NO if decrementation of DC counter is not possible

Short description:

At the end of a driving cycle (LV_DC 1->0) the driving cycle counter for the failure XX can be decremented if similar conditions without failure are recognized and no exceedance reported from diagnostic function (bit 5 of STATE_SCDN[k] = 0) (used for misfire monitoring). For temporary failure or confirmed failure with MIL off, even if similar conditions are not recognized during a C_CTR_CDN_EQU_DC_MAX number of driving cycle, the driving cycle counter of the failure can be decremented after C_CTR_CDN_EQU_DC_MAX driving cycles.

Application conditions:

Initialization: -

Recurrence: -

Activation: at action request

Formula section:

```
k = Scdn(XX)
DCdec = NO        (decrementation of DC counter is not possible)
If   bit 2 of STATE_SCDN[k] = 1          (similar conditions encountered
                                           with no failure)
and  bit 5 of STATE_SCDN[k] = 0          (no exceedance reported)
Then      DCdec = YES                    (decrementation of DC counter is possible)
Else      If    LV_DC_MAX[IDX] = 0       (failure is temporary or confirmed with
                                          MIL off)
          Then  CTR_SCDN_EQU_DC[k] = CTR_SCDN_EQU_DC[k] – 1
                If    CTR_SCDN_EQU_DC[k] = 0
                      bit 2 of STATE_SCDN[k] = 1
                      DCdec = YES        (decrementation of DC counter is possible)
                Endif
          Endif
Endif


If    DCdec = YES
and   (LV_DC_MAX[IDX] = 0             (failure is temporary or confirmed with MIL off and
                                       DC counter of failure will be 0 after decrementation)
      or  (LV_DC_MAX[IDX] = 1 and CTR_DC[IDX] <= DC_DEC_XX))
                                      (failure is confirmed with MIL on and
                                       MIL will be switched off after decrementation)
Then  bit 0 of STATE_SCDN[k] = 0      (The SCDN window of failure XX will be updated
                                       With new engine conditions (N / LOAD (min/max))
                                       at next failure occurrence)
Endif
```

27.25.5 Similar conditions erase

Description:

Syntax : ACTION_ERRM_EraseScdn (IN <XX>)

Parameter (in) : XX number of failure to store in dynamic memory

Parameter (out) : -

Short description:

The similar conditions are erased by calling the action ACTION_ERRM_EraseScdn :
- when saved ram lost (at initialization)
- when corresponding failure is erased (see dynamic error management core module)

Application conditions:

Initialization: -

Recurrence: -

Activation: at action request

Formula section:

k = Scdn (XX)

STATE_SCDN[k] = 0

N_MAX_SCDN_ EQU [k] = 0 [rpm]

N_MIN_SCDN_EQU[k] = 8160 [rpm]

LOAD_MAX_SCDN_EQU[k] = 0 [%]

LOAD_MIN_SCDN_EQU[k] = 99.60937 [%]

CTR_SCDN_EQU_DC[k] = 0

LV_WUP_SCDN_EQU[k] = 0

CTR_SCDN_EQU[k] = 0

CTR_SCDN_SUM[k] = 0

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_SCDN_EQU	1	0...1FE0H	0...8160	1	[rpm]
engine speed threshold for similar conditions detection (375 rpm)					
C_LOAD_SCDN_EQU	1	0...FFH	0...99.60937	0.390625	[%]
LOAD value for similar conditions detection (typical 20%)					
C_CTR_SCDN_EQU_DC_MAX	1	0...FFH	0...255	1	[-]
maximum value of the driving cycle counter for similar conditions erase					
C_CTR_SCDN_NOT_ERR_THD[NC_NR_WIN_SCDN]	1	0...FFH	0...99.60937	0.390625	[%]
threshold for ratio CTR_SCDN_EQU[NC_NR_WIN_SCDN] / CTR_SCDN_SUM[NC_NR_WIN_SCDN] to recognize SCDN without failure					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_WIN_SCDN	1	1...FFH	1...255	1	[-]
Number of instances used for similar conditions calculation					

27.26 Similar conditions (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LOAD_SCDN	V/O	0...FFH	0...99.60937	0.390625	[%]
Engine load for SCDN calculation					

Input data:

NC_NR_WIN_SCDN	LV_CDN_DIAG_FSD[NC_CBK_EX_NR]	LV_END_DIAG_WIN_FSD[NC_CBK_EX_NR]	LOAD_CLC
NC_CBK_EX_NR	LV_END_DIAG_WIN_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_CDN_DIAG_FSD_LAM_LIM[NC_CBK_EX_NR]	

Import actions:

ACTION_ERRM_CdnDiagScdn (IN <XX>)
ACTION_ERRM_EndWinScdn (IN <XX> , IN <EXC>)

FUNCTION DESCRIPTION:

General information:

This file defines the usage of the similar conditions functionality.

27.26.1 Similar conditions window definition

27.26.1.1 Similar condition data definition

Description:

There is only some failure using the similar condition so this table allows to do the link between the index of the failure in static memory and the index of the failure using the similar condition:

Diagnosis instances XX using similar conditions functionality	NC_NR_WIN_SCDN = **10**

Diagnosis instance XX	**similar conditions instance (k)**
FSD_1	0
FSD_2	1
FSD_LAM_LIM_1	2
FSD_LAM_LIM_2	3
MIS_0	4
MIS_1	5
MIS_2	6
MIS_3	7
MIS_4	8
MIS_5	9
	(NC_NR_WIN_SCDN - 1)

One function is defined to do the link between the index of the failure in static memory and the index of the failure using the similar condition: k = Scdn(XX)

27.26.1.2 Load calculation for similar condition

Description:

The calculated load SAE1979 (LOAD_CLC) is used for similar condition.

Application conditions:

Initialisation:	At reset:	LOAD_SCDN = LOAD_CLC
Recurrence:	100 ms	
Activation:	-	

Formula section:

LOAD_SCDN = LOAD_CLC

27.26.1.3 Treatment activated by the fuel system diagnosis for the similar condition

Description:

This treatement allows to call the two generic ACTION_ERRM_CdnDiagScdn (IN <XX>), ACTION_ERRM_EndWinScdn (IN <XX>, IN <EXC>) actions using for the similar condition.

Application conditions:

Initialisation: -

Recurrence: fuel system diagnosis recurrency

Activation: -

Formula section:

```
For i = 0 to NC_CBK_EX_NR - 1
    Adaptation to calculate a ratio used to recognize the similar condition with or without failure:
    If LV_CDN_DIAG_FSD[i] = 1
    Then   ACTION_ERRM_CdnDiagScdn (FSD[i])
    Endif
    If LV_CDN_DIAG_FSD_LAM_LIM[i] = 1
    Then   ACTION_ERRM_CdnDiagScdn (FSD_LAM_LIM[i])
    Endif

    Adaptation for the record and recognition of similar conditions:

    If LV_END_DIAG_WIN_FSD[i] 0 ->1

    Then

            Record and recognition of similar conditions at the end of diagnosis or when the failure is present:

            ACTION_ERRM_EndWinScdn (FSD[i], 0)

    Endif

    If LV_END_DIAG_WIN_FSD_LAM_LIM[i] 0 ->1

    Then

            Record and recognition of similar conditions at the end of diagnosis or when the failure is present:

            ACTION_ERRM_EndWinScdn (FSD_LAM_LIM[i], 0)

    Endif

EndFor
```

27.26.2 SCDN usage for direct failure confirmation

Export actions:

ACTION_ERRM_ConfirmErrScdn (OUT <DCconf>, IN <XX>)
This action calculates the possibility to confirm a failure XX directly

Description:

Syntax :	ACTION_ERRM_ConfirmErrScdn (OUT <DCconf>, IN <XX>)	
Parameter (in) :	XX	number of failure to store in dynamic memory
Parameter (out) :	DCconf	1 if failure must be confirmed directly
		0 if failure should not be confirmed directly

Short description:

With the ACTION_ERRM_ConfirmErrScdn (OUT <DCconf>, IN <XX>) the value DCconf is transmitted to the driving cycle counter management to give the possibility to confirm a failure directly by incrementing the driving cycle counter for the failure XX to the maximum. This treatment must be done in the following situation:

A temporary failure XX caused by misfire B is in the memory and a failure XX caused by misfire A occures (not stored before). Then the misfire B depending failure can be erased and the misfire A failure can be confirmed directly.

Application conditions:

Initialization:	-
Recurrence:	-
Activation:	at action request

Formula section:

DCconf = 0 (direct confirmation of failure XX is not possible)

27.27 Error management communication interface

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_MKD_MOD	O	0...1H	0...1	1	[-]
Marked Mode					
ERR_SYM_DTC[NC_NR_ERR_DYN]	-	0...FH	0...15	1	[-]
Symtom calculation used by the API communication					
#IF NLC_TREAT_DIAG_MIS = 0					
SYM_CYL_DTC_MIS_A	-	0...FFFFH	0...65535	1	[-]
Cylinder calculation with misfire A used by the API communication					
SYM_CYL_DTC_MIS_B1	-	0...FFFFH	0...65535	1	[-]
Cylinder calculation with misfire B1 used by the API communication					
SYM_CYL_DTC_MIS_B4	-	0...FFFFH	0...65535	1	[-]
Cylinder calculation with misfire B4 used by the API communication					
#ENDIF					
IDX_TMP_RBM	V	0...FFH	0...255	1	[-]
Temporary current position, within the list					
RATIO_TMP_RBM	V	0...FFFFH	0...7.99987	0.1221e-3	[-]
Temporary variable used to store the lowest Numerator/Denominator ratio of the monitor located at IDX_TMP_RBM position					

Input data:

ERR_SYM_MEM[NC_NR_ERR_DYN]	SYM_CYL_MEM_MIS_A	SYM_CYL_MEM_MIS_B1	SYM_CYL_MEM_MIS_B4
ERR_SYM_LST[NC_NR_ERR_DYN]	SYM_CYL_LST_MIS_A	SYM_CYL_LST_MIS_B1	SYM_CYL_LST_MIS_B4
ID_ERR_DTC_XX	ID_ERR_DTC_MIS	NC_CYL_NR	NC_NR_DTC_FMT
LV_ERR_XX	LV_ERR_MEM_XX	LV_ERR_TMP[NC_NR_ERR_DYN]	WAL_CONF_XX
LV_ERR_CFM[NC_NR_ERR_DYN]	LV_ERR_DISA[NC_NR_ERR_DYN]	PRI_CONF_XX	ENVD_OBD[NC_NR_ENVD_OBD][NC_NR_ERR_DYN]
ENVD_CUS_CMN[NC_NR_ENVD_CUS_CMN][NC_NR_ERR_DYN]	ENVD_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN][NC_NR_FRF_SET][NC_NR_ERR_DYN]	ENVD_CUS_SET_SPC[NC_NR_ENVD_CUS_SET_SPC][NC_NR_FRF_SET][NC_NR_ERR_DYN]	C_ERR_CLAS_XX
CTR_ABC_XX	CTR_ABC_END_DIAG_XX	LV_END_DIAG_XX	LV_READY_XX
LV_CDN_DIAG_XX	ERR_SYM_XX	CTR_FRC[NC_NR_ERR_DYN]	CTR_DC[NC_NR_ERR_DYN]
CTR_WUP_CYC[NC_NR_ERR_DYN]	CTR_ERR_DYN_NR	LV_DC_MAX[NC_NR_ERR_DYN]	LV_ERR_DC[NC_NR_ERR_DYN]
NLC_TREAT_DIAG_MIS	DC_DEC_XX	DC_INC_XX	DC_MAX_XX
CTR_IGK_CYC_RBM	CTR_CDN_OBD_RBM	CTR_CDN_RBM[NC_NR_DIAG_RBM]	CTR_COMP_RBM[NC_NR_DIAG_RBM]
NC_NR_DIAG_RBM	NLC_OBD_RBM_ENA	STATE_READY_OBD_1	STATE_READY_OBD_2
STATE_READY_OBD_3	LV_DC		

Export actions:

ACTION_ERRM_ReadDtcByTypeOfDtc (
IN <TypeOfDtc>,
IN <LevelOfDtc>,
INOUT <ListOfDtc>,
OUT <ResultDtc>)

This action returns a list of DTC. All the DTC returns have the same type defined by the parameter TypeOfDtc

```
ACTION_ERRM_ReadDtcByDtc (
                        IN <DtcIdentifier>,
                        IN <LevelOfDtc>,
                        OUT <ResultDtc>)
```

This action returns a result to learn if a DTC number of a certain level is stored in memory.

```
ACTION_ERRM_ReadInfoByTypeOfDtc (
                        IN <TypeOfDtc>,
                        IN <InfoIdentifier>,
                        INOUT <ListOfDtcInfo>,
                        OUT <ResultDtc> )
```

This action returns a structure of data of DTCs. All the data returned are related to DTC, which have the same type defined by the parameter TypeOfDtc and which are stored in dynamic memory.

```
ACTION_ERRM_ReadInfoByDtc (
                        IN <DtcIdentifier>,
                        IN <LevelOfDtc>,
                        IN <InfoIdentifier>,
                        INOUT <ListOfDtcInfo>,
                        OUT <ResultDtc> )
```

This action returns a structure of data of a DTC, which is stored in dynamic memory.

```
ACTION_ERRM_ReadDtcLevelByDtcLevel (
                        IN <DtcIdentifier>,
                        IN <LevelOfDtc>,
                        INOUT <ListofDtc>,
                        OUT <ResultDtc> )
```

This action returns a Customer DTCLevel of Law DTCLevel which is stored in dynamic memory or returns a Law DTCLevel of Customer DTCLevel which is stored in dynamic memory.

```
ACTION_ERRM_ReadQuantityOfDtc (
                        IN <TypeOfDtc>,
                        IN <LevelOfDtc>,
                        INOUT <Quantity>,
                        OUT <ResultQuantity>)
```

This action returns the quantity of DTCs with a certain type, which are stored in memory.

```
ACTION_ERRM_ReadFrfByDtc (
                        IN <TypeOfFF>,
                        IN <FFIdentifier>,
                        IN <DtcIdentifier>,
                        IN <LevelOfDtc>,
                        INOUT <Frf>,
                        OUT <ResultFrf>)
```

This action returns a Freeze_Frame (with a particular type) related to the DTC given in parameter.

```
ACTION_ERRM_ClrInfoByTypeOfDtc (
                        IN <TypeOfDtc>,
                        OUT <ResultClrInfo>)
```

This action clear all the failure in dynamic memory associated to DTC with a certain type.

```
ACTION_ERRM_ClrInfoByDtc (
                        IN <DtcIdentifier>,
                        IN <LevelOfDtc>,
                        OUT <ResultClrInfo>)
```

This action clear the failure in dynamic memory associated to a DTC number.

```
ACTION_ERRM_ControlDtcSettings (
                        IN <MarkedMode>)
```

This action allows to activate or deactivate the marked mode. In this mode all new failure which appears are marked and then erased when we leave this mode.

```
ACTION_ERRM_ActivateMarkedMode (
                         IN <MarkedMode> )
```

This API shall be used to activate/deactivate the marked mode.

```
ACTION_ERRM_ReadReadinessCode (
                        INOUT <ReadinessCode>,
                        OUT <ResultReadinessCode>)
```

This action allows to read the readiness code information

```
ACTION_ERRM_ReadReadinessCodCus (
                        INOUT <ReadinessCode>,
                        OUT <ResultReadinessCode>)
```

This action allows to read the customer specific readiness code information CARB_OTHER/NO_CARB

ACTION_ERRM_SelectRbmData (INOUT <ListRbmData>, OUT < ResultRbmData >)
This action is used to request the selection of data to be transmitted by the Communication Mode $09
ACTION_ERRM_SelectRbmByGroup (IN < GroupName >, INOUT < ListOfRbmDataByGroup >, OUT < ResultRbmDataSelection >)
This action determines within one single group, the data with lowest ratios.
ACTION_ERRM_ClearRbmStatistics (OUT < ResultClrInfo>)
This action clears all the Rate-Based Monitoring statistics

Import actions:

ACTION_ERRM_EraseErr (IN <IDX>)
ACTION_ERRM_InitReadiness ( )

FUNCTION DESCRIPTION:

General information:

This chapter provides an open interface for other module which have link with error management functionality (communication module for instance). This interface is described with API (Application Programming Interface) to provide access to dynamic data flow in contrast to static data flow accessed directly through variable names.

By calling API, you can access in failure memory data related to Error Management such freeze frame and DTC stored, readiness code state and so on.

An API can have some input parameters.

Results returns by the API are defined as output parameters.

To returns functional results, a software structure (buffer) is used that allows to returns some data with not fixed size (a list of data for instance).

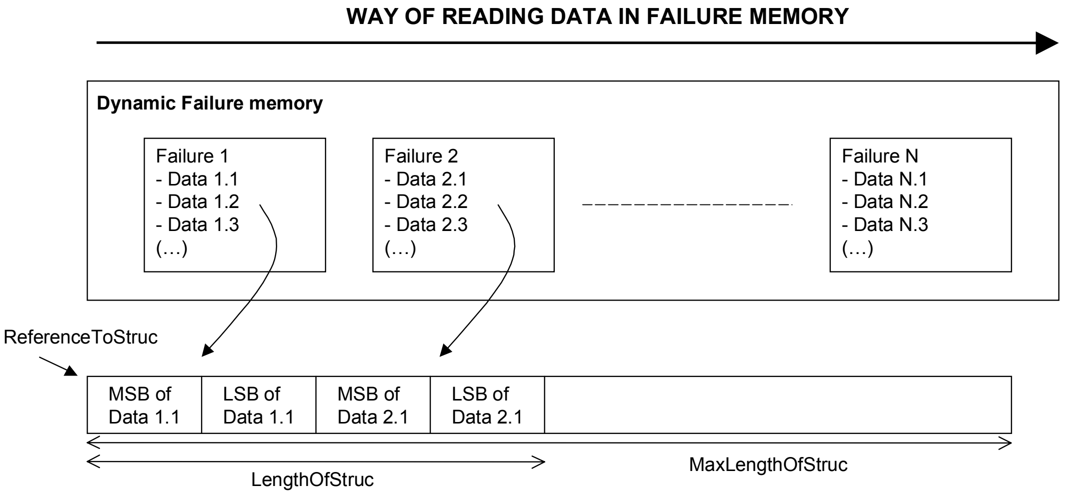
This structure is defined by the parameter in INOUT.

In the output parameter, a flag indicate if the software buffer is full or not.

Way of reading the failure memory :

API has to read through the failure memory to access some data related to Error Management.

In this case the failure memory is read in the order following the first failure entry, that means the oldest one. Thus the software structure is fill in first with data related to first failure entry (if need), then with data related to second failure entry (if need) and so on.


WAY OF READING DATA IN FAILURE MEMORY
Dynamic Failure memory
Failure 1
- Data 1.1
- Data 1.2
- Data 1.3
(…)
Failure 2
- Data 2.1
- Data 2.2
- Data 2.3
(…)
Failure N
- Data N.1
- Data N.2
- Data N.3
(…)
ReferenceToStruc
MSB of Data 1.1
LSB of Data 1.1
MSB of Data 2.1
LSB of Data 2.1
MaxLengthOfStruc
LengthOfStruc

Description of parameters IN, OUT, INOUT used by API:

The INPUT parameters used by API and their authorised value are defined below :

TypeOfDtc : Type of Dtc

Authorised values for TypeOfDtc are :

- ALL : all the DTC stored in Dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
- ALL FIRST: all the DTC stored at the first time in Dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
- ALL LAST: all the DTC stored at the last time in Dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
- PRESENT : all the DTC of present failure stored in dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
  and LV_ERR_XX = 1
- PENDING : all the DTC of pending failure stored in dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
  and LV_ERR_PND[IDX] = 1
- TEMPORARY : all the DTC of temporary failure stored in dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
  and LV_ERR_TMP[IDX] = 1
- CONFIRMED : all the DTC of confirmed failure stored in dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
  and LV_ERR_CFM[IDX] = 1
- DISAPPEARED : all the DTC of disappeared failure stored in dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
  and LV_ERR_DISA[IDX] = 1
- MARKED : all DTC of failure marked and stored in dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
  and LV_ERR_MKD[IDX] = 1
- NOT MARKED : all DTC of failure not marked and stored in dynamic memory.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
  and LV_ERR_MKD[IDX] = 0
- OBD : all DTC related to emission relevant failure.
  Failures concerned in error management : LV_ERR_MEM_XX = 1
  and a Carb/EOBD failure :
  (emission relevant)
  (definition see failure class specification)

- OBD FIRST: all the DTC stored at the first time in Dynamic memory related to emission relevant failure.
  Failures concerned in error management : LV_ERR_MEM_XX = 1

  and a Carb/EOBD failure :
  (emission relevant)
  (definition see failure class specification)
- OBD LAST: all the DTC stored at the last time in Dynamic memory.related to emission relevant failure.
  Failures concerned in error management : LV_ERR_MEM_XX = 1

  and a Carb/EOBD failure :
  (emission relevant)
  (definition see failure class specification)
- Logical AND combination between these values.
  For instance : PRESENT and CONFIRMED are failure in error management with flags : LV_ERR_MEM_XX = 1 and LV_ERR_XX = 1 and LV_ERR_CFM[IDX] = 1.

Calculation of ERR_SYM_DTC[IDX] and SYM_CYL_DTC_XX following the TypeOfDtc parameter:

For all failure except the misfiring failure treated by type :

```
case TypeOfDtc
        ALL : ERR_SYM_DTC[IDX] = ERR_SYM_LST[IDX] or ERR_SYM_MEM[IDX]
        ALL FIRST: ERR_SYM_DTC[IDX] = ERR_SYM_MEM[IDX]
        ALL LAST : ERR_SYM_DTC[IDX] = ERR_SYM_LST[IDX]
        PRESENT : ERR_SYM_DTC[IDX] = ERR_SYM_LST[IDX]
        PENDING : ERR_SYM_DTC[IDX] = ERR_SYM_LST[IDX]
        TEMPORARY : ERR_SYM_DTC[IDX] = ERR_SYM_MEM[IDX]
        CONFIRMED : ERR_SYM_DTC[IDX] = ERR_SYM_MEM[IDX]
        DISAPPEARED : ERR_SYM_DTC[IDX] = ERR_SYM_MEM[IDX]
        MARKED :      ERR_SYM_DTC[IDX] =
                      ERR_SYM_LST[IDX] or ERR_SYM_MEM[IDX]
        NOT MARKED :        ERR_SYM_DTC[IDX] =
                            ERR_SYM_LST[IDX] or ERR_SYM_MEM[IDX]
        OBD : ERR_SYM_DTC[IDX] = ERR_SYM_LST[IDX] or ERR_SYM_MEM[IDX]
        OBD FIRST : ERR_SYM_DTC[IDX] = ERR_SYM_MEM[IDX]
        OBD LAST : ERR_SYM_DTC[IDX] = ERR_SYM_LST[IDX]
End case
```

For the misfiring failure treated by type (XX = MIS_A, MIS_B1 or MIS_B4)

```
IF NLC_TREAT_DIAG_MIS = 0
case TypeOfDtc
        ALL : SYM_CYL_DTC_XX = SYM_CYL_LST_XX or SYM_CYL_MEM_XX
        ALL FIRST : SYM_CYL_DTC_XX = SYM_CYL_MEM_XX
        ALL LAST: SYM_CYL_DTC_XX = SYM_CYL_LST_XX
        PRESENT: SYM_CYL_DTC_XX = SYM_CYL_LST_XX
        PENDING : SYM_CYL_DTC_XX = SYM_CYL_LST_XX
        TEMPORARY : SYM_CYL_DTC_XX = SYM_CYL_MEM_XX
        CONFIRMED : SYM_CYL_DTC_XX = SYM_CYL_MEM_XX
        DISAPPEARED : SYM_CYL_DTC_XX = SYM_CYL_MEM_XX
        MARKED : SYM_CYL_DTC_XX = SYM_CYL_LST_XX or SYM_CYL_MEM_XX
        NOT MARKED : SYM_CYL_DTC_XX =
                        SYM_CYL_LST_XX or SYM_CYL_MEM_XX
        OBD : SYM_CYL_DTC_XX = SYM_CYL_LST_XX or SYM_CYL_MEM_XX
        OBD FIRST: SYM_CYL_DTC_XX = SYM_CYL_MEM_XX
        OBD LAST: SYM_CYL_DTC_XX = SYM_CYL_LST_XX
End case
ENDIF        { NLC_TREAT_DIAG_MIS = 0 }
```

For the logical AND combination between the different TypeOfDtc values, the ERR_SYM_DTC[IDX] or SYM_CYL_DTC_XX will be calculated with a binary logical AND from the associated carrier symptom.

For instance : PRESENT and CONFIRMED are failure in error management with flags : LV_ERR_MEM_XX = 1 and LV_ERR_XX = 1 and LV_ERR_CFM[IDX] = 1.
It means : ERR_SYM_DTC[IDX] = ERR_SYM_LST[IDX] **&** ERR_SYM_MEM[IDX],
SYM_CYL_DTC_XX = SYM_CYL_LST_XX **&** SYM_CYL_MEM_XX.

Calculation of ERR_SYM_DTC[IDX] and SYM_CYL_DTC_XX in case the API doesn't use the TypeOfDtc parameter :

ERR_SYM_DTC[IDX] = ERR_SYM_LST **or** ERR_SYM_MEM[IDX]

SYM_CYL_DTC_XX = SYM_CYL_LST_XX **or** SYM_CYL_MEM_XX

DtcIdentifier : Number of the DTC (identifier)

Authorised value for DtcIdentifier are : NO_DTC, ...<integer>

TypeOfFF : Type of Freeze Frame

Authorised values for TypeOfFF are :

- LAW : part of the freeze frame defined by the law (Carb)

- CUS_CMN : part of the freeze frame defined for the customer and not in the set
- CUS_SET : part of the freeze frame defined for the customer and in the set
- All combination of these values.

LevelOfDtc : Allowing to access to OBD error codes or customer error codes.

Authorised values for LevelOfDtc are :

- LAW : the DTC corresponds to a OBD DTC (Carb)
- CUS : the DTC corresponds to a specific customer DTC

FFIdentifier : Number of FF (identifier)
Authorised values for FFIdentifier are :

- FIRST : first freeze frame is requested
- SECOND : second freeze frame is request
- ...

MarkedMode : State of the marked mode

Authorised values for MarkedMode are :

- ON : marked mode activation : all failure stored are marked.
- OFF : marked mode deactivation (normal mode)

InfoIdentifier : Identifier for the diagnosis related information which is returned by the API (only words are returned)

Authorised information for InfoIdentifier are (only one information is returned by the API):

- CTR_ABC_XX
- CTR_ABC_END_DIAG_XX
- LV_END_DIAG_XX
- LV_READY_XX
- LV_CDN_DIAG_XX
- ERR_SYM_XX
- LV_ERR_XX
- LV_ERR_MEM_XX
- CTR_FRC[IDX]
- CTR_DC[IDX]
- CTR_WUP_CYC[IDX]
- ERR_SYM_MEM[IDX]
- ERR_SYM_LST[IDX]
- ERR_SYM_DTC[IDX]
- LV_ERR_CFM[IDX]
- LV_ERR_DISA[IDX]
- LV_ERR_TMP[IDX]

- LV_ERR_PND[IDX]
- LV_DC_MAX[IDX]
- LV_ERR_DC[IDX]
- LV_ERR_MKD[IDX]
- C_ERR_CLAS_XX
- WAL_CONF_XX
- PRI_CONF_XX
- DC_INC_XX
- DC_DEC_XX
- DC_MAX_XX
- ID_ERR_DTC_XX[0][0]
- ID_ERR_DTC_XX[0][1]
- ID_ERR_DTC_XX[0][2]
- ID_ERR_DTC_XX[0][3]
- ID_ERR_DTC_XX[0][4]

if NC_NR_DTC_FMT = 0

(Manage 6 DTC per failure)

- ID_ERR_DTC_XX[0][5]

else

(Manage 10 DTC per failure)

- ID_ERR_DTC_XX[1][0]
- ID_ERR_DTC_XX[1][1]
- ID_ERR_DTC_XX[1][2]
- ID_ERR_DTC_XX[1][3]
- ID_ERR_DTC_XX[1][4]

endif

Please note that some application incidences can provide additional project specific informations.

The OUTPUT parameters used by API and their authorised value are defined below :

ResultDtc : Result to say if there is some DTC in memory corresponding to the API request

Authorised values for ResultDtc are :

1st bit of ResultDtc :

- (0) NO_DTC_PRESENT : there is no DTC in memory corresponding to the API request
- (1) DTC_PRESENT : there is some DTC in memory corresponding to the API request

2nd bit of ResultDtc :

- (0) NO_BUFFER_FULL : The software buffer used to return data is not full
- (1) BUFFER_FULL : The software buffer used to return data is full

ResultFrf : Result to say if there is some Freeze Frame in memory corresponding to the API request

Authorised values for ResultFrf are :

1st bit of ResultFrf :

- (0) NO_FRF_PRESENT : there is no freeze frame in memory corresponding to the API request
- (1) FRF_PRESENT : there is some freeze frame in memory corresponding to the API request

2nd bit of ResultFrf :

- (0) NO_BUFFER_FULL : The software buffer used to return data is not full
- (1) BUFFER_FULL : The software buffer used to return data is full

ResultQuantity : Result to say if the result of the API is valid.

Authorised values for ResultQuantity are :

1st bit of ResultQuantity :

- (0) NO_DTC_PRESENT : there is no DTC in memory corresponding to the API request
- (1) DTC_PRESENT : there is some DTC in memory corresponding to the API request

ResultClrInfo : Result to say if the clear DTC made by the API is done or not.

Authorised values for ResultClrInfo are :

1st bit of ResultClrInfo :

- (0) INFO_CLEARED : DTC are cleared by the call of the API
- (1) INFO_NOT_CLEARED : no DTC is cleared by the call of the API

ResultReadinessCode : Result to say if the API is done or not.

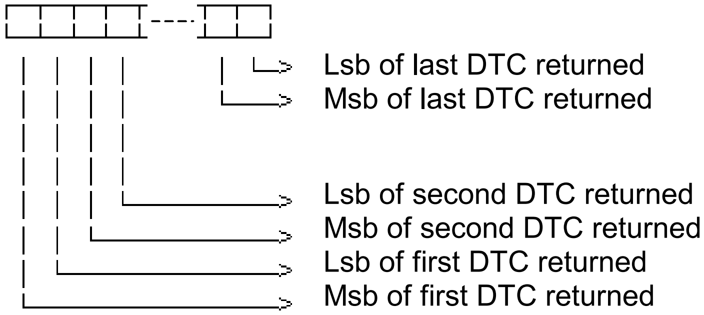
Authorised values for ResultReadinessCode are :

1st bit of ResultReadinessCode :

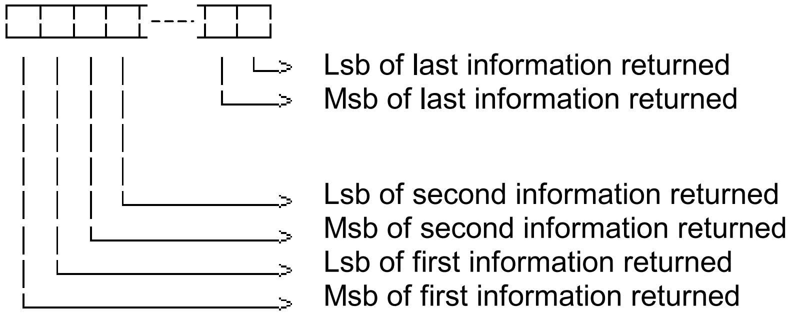
- (0) POSITIVE_RESPONSE : the call of the API is successful
- (1) NEGATIVE_RESPONSE : the call of the API is not successful

The INPUT/OUTPUT parameters used by API and their authorised value are defined below :

- ListOfDtc : Software structure filled up with a list of DTC

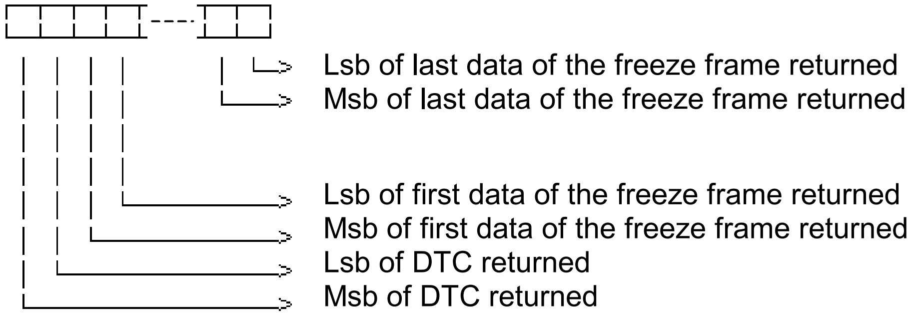


- ListOfDtcInfo : Software structure filled up with a list of DTCInfo



- Quantity : Software structure filled up with a quantity of DTC

- Frf : Software structure filled up with a freeze frame



- ReadinessCode : Software structure filled up with readiness code

27.27.1 API for reading Diagnostic Trouble Code (DTC)

27.27.1.1 Request a list of DTCs by type of DTC.

Description :

Syntax : ACTION_ERRM_ReadDtcByTypeOfDtc (
IN <TypeOfDtc>,
IN <LevelOfDtc>,
INOUT <ListOfDtc>,
OUT <ResultDtc>)

Parameter(in) :	TypeOfDtc	Type of DTC which is requested by the API.
	LevelOfDtc	OBD or customer DTC is requested by the API
Parameter(out) :	ResultDtc	Boolean to say if some DTC is in memory
Parameter(inout) :	ListOfDtc	Software structure fills up with a ListOfDtc

Short Description : This API returns a list of DTCs. All the DTC returns have the same type defined by the parameter TypeOfDtc and the same level defined in LevelOfDtc.

For each failure, if NC_NR_DTC_FMT = 0 then 6 DTC are defined : one DTC for each symptom (OBD or Customer), one OBD global DTC, one customer global DTC else 10 DTC are defined : one OBD DTC for each symptom, one OBD global DTC, one customer specific DTC number for each symptom and one customer global DTC.

When LevelOfDtc = LAW : OBD Dtc is request

If NC_ERR_DTC_REQ_OBD = SYMPTOM, the DTC is build up following the detected symptom. If more than one symptom is detected, a DTC per symptom detected is returned. There is a particular case for Misfiring failure. For this failure, the DTC number is built up following the cylinder where the misfiring is detected.

If NC_ERR_DTC_REQ_OBD = FAILURE, the OBD global DTC is returned.

When LevelOfDtc = CUS : Customer Dtc is request

If NC_ERR_DTC_REQ_CUS = SYMPTOM, the DTC is build up following the detected symptom. If more than one symptom is detected, a DTC per symptom detected is returned. There is a particular case for Misfiring failure. For this failure, the DTC number is built up following the cylinder where the misfiring is detected.

If NC_ERR_DTC_REQ_CUS = FAILURE, the customer global DTC is returned.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

```
1st bit of ResultDtc = NO_DTC_PRESENT
2nd bit of ResultDtc = NO_BUFFER_FULL

For each failure IDX stored in dynamic memory,
    If      the failure belongs to failures group defined by the parameter TypeOfDtc
    Then    1st bit of ResultDtc = DTC_PRESENT
            If  failure is not with status = CARB_MIS
            Then Condition for failure traitment is true
            Endif
            If  NLC_TREAT_DIAG_MIS = 1
            Then Condition for failure traitment is true
            Endif
            If  Condition for failure traitment is true
            Then If     LevelOfDtc = LAW                          (OBD DTC is request)
                 Then   If  NC_ERR_DTC_REQ_OBD=SYMPTOM (access by symptom)
                        Then  For each bit i which is set in ERR_SYM_DTC[IDX]
                                    software structure fill up with
                                    ID_ERR_DTC_XX[ 0 ] [ i ]
                              Endfor
                        Else  (access by failure; NC_ERR_DTC_REQ_OBD = FAILURE)
                              software structure fill up with ID_ERR_DTC_XX[0][4]
                        Endif
                 Else   (Customer DTC is request ; LevelOfDtc = CUS)
                        If  NC_ERR_DTC_REQ_CUS = SYMPTOM (access by symptom)
                        Then  For each bit i which is set in ERR_SYM_DTC[IDX]
                                    software structure fill up with
                                    ID_ERR_DTC_XX[ NC_NR_DTC_FMT ] [ i ]
                              Endfor
                        Else  (access by failure; NC_ERR_DTC_REQ_CUS = FAILURE)
                              software structure fill up with
                              ID_ERR_DTC_XX
                              [NC_NR_DTC_FMT][5-NC_NR_DTC_FMT]
                        Endif
                 Endif
```

```
        Else        (particular case for misfiring failure NLC_TREAT_DIAG_MIS = 0)
                    XX = F (IDX) with XX = MIS_A, MIS_B1 or MIS_B4
            If      LevelOfDtc = LAW                         (OBD DTC is request)
            Then    If      SYM_CYL_DTC_XX [12]=1  (Random)
                    Then    software structure fill up with
                            ID_ERR_DTC_MIS [ 0 ] [ NC_CYL_NR+1]
                    Endif
                    If      SYM_CYL_DTC_XX [13]=1  (Multiple cylinder)
                    Then    software structure fill up with
                            ID_ERR_DTC_MIS [ 0 ] [ NC_CYL_NR ]
                    Endif
                    For each cylinder bit i which is set in SYM_CYL_DTC_XX
                                software structure fill up with
                                ID_ERR_DTC_MIS[ 0 ] [ i ]
                    Endfor
            Else    (Customer DTC is request ; LevelOfDtc = CUS)
                    If      SYM_CYL_DTC_XX [12]=1  (Random)
                    Then    software structure fill up with
                            ID_ERR_DTC_MIS [ 1 ] [ NC_CYL_NR+1]
                    Endif
                    If      SYM_CYL_DTC_XX [13]=1  (Multiple cylinder)
                    Then    software structure fill up with
                            ID_ERR_DTC_MIS [ 1 ] [ NC_CYL_NR ]
                    Endif
                    For each cylinder bit i which is set in SYM_CYL_DTC_XX
                            software structure fill up with
                            ID_ERR_DTC_MIS[ 1 ] [ i ]
                    Endfor
            Endif
        Endif
    Endif
Endfor
```

During all the software structure filling-in, if the software buffer is full then 2nd bit of ResultDtc is set to BUFFER_FULL, software process is aborted.

If there is duplicated DTC then only the oldest is returned

27.27.1.2 Request a DTC by DTC number

Description :

Syntax :	ACTION_ERRM_ReadDtcByDtc (IN <DtcIdentifier>, IN <LevelOfDtc>, OUT <ResultDtc>)	
Parameter(in) :	DtcIdentifier	Number of the DTC (identifier) which is requested.
	LevelOfDtc	OBD or customer DTC is requested by the API
Parameter(out) :	ResultDtc	Boolean to say if there is the DTC in memory
Short Description :	This API returns a result to learn if a DTC number of a certain level is stored in memory.	

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

1st bit of ResultDtc = NO_DTC_PRESENT

2nd bit of ResultDTC = NO_BUFFER_FULL

```
For each failure IDX stored in dynamic memory,
    If   failure is not with status = CARB_MIS
    Then Condition for failure traitment is true
    Endif
    If   NLC_TREAT_DIAG_MIS = 1
    Then Condition for failure traitment is true
    Endif
    If   Condition for failure traitment is true
    Then If      LevelOfDtc = LAW                         (OBD DTC is request)
         Then If      NC_ERR_DTC_REQ_OBD = SYMPTOM   (access by symptom)
              Then For each bit i which is set in ERR_SYM_DTC[IDX]
                           If      DtcIdentifier = ID_ERR_DTC_XX [0][i]
                           Then    1st bit of ResultDtc = DTC_PRESENT
                           Endif
                   Endfor
              Else (access by failure ; NC_ERR_DTC_REQ_OBD = FAILURE)
                   If      DtcIdentifier = ID_ERR_DTC_XX [0][4]
                   Then    1st bit of ResultDtc = DTC_PRESENT
                   Endif
              Endif
         Else (Customer DTC is request ; LevelOfDtc = CUS)
              If      NC_ERR_DTC_REQ_CUS = SYMPTOM   (access by symptom)
              Then For each bit i which is set in ERR_SYM_DTC[IDX]
                   If      DtcIdentifier = ID_ERR_DTC_XX [NC_NR_DTC_FMT][i]
                   Then    1st bit of ResultDtc = DTC_PRESENT
                   Endif
                   Endfor
              Else (access by failure ; NC_ERR_DTC_REQ_CUS = FAILURE)
                   If      DtcIdentifier = ID_ERR_DTC_XX
                           [NC_NR_DTC_FMT][5-NC_NR_DTC_FMT]
                   Then    1st bit of ResultDtc = DTC_PRESENT
                   Endif
              Endif
         Endif
```

```
                Else      (particular case for misfiring failure NLC_TREAT_DIAG_MIS = 0)
                    XX = F (IDX) with XX = MIS_A, MIS_B1 or MIS_B4
                    If      LevelOfDtc = LAW                      (OBD DTC is request)
                    Then    If      SYM_CYL_DTC_XX [12]=1          (Random)
                            Then    If      DtcIdentifier=ID_ERR_DTC_MIS [0] [NC_CYL_NR+1]
                                    Then    1st bit of ResultDtc = DTC_PRESENT
                                    Endif
                            Endif
                            If      SYM_CYL_DTC_XX [13]=1          (Multiple cylinder)
                            Then    If      DtcIdentifier=ID_ERR_DTC_MIS [0] [NC_CYL_NR]
                                    Then    1st bit of ResultDtc = DTC_PRESENT
                                    Endif
                            Endif
                            For each cylinder bit i which is set in   SYM_CYL_DTC_XX
                                    If      DtcIdentifier = ID_ERR_DTC_MIS[0][i]
                                    Then    1st bit of ResultDtc = DTC_PRESENT
                                    Endif
                            Endfor
                    Else    (Customer DTC is request ; LevelOfDtc = CUS)
                            If      SYM_CYL_DTC_XX [12]=1          (Random)
                            Then    If      DtcIdentifier=ID_ERR_DTC_MIS [1] [NC_CYL_NR+1]
                                    Then    1st bit of ResultDtc = DTC_PRESENT
                                    Endif
                            Else
                            If      SYM_CYL_DTC_XX [13]=1          (Multiple cylinder)
                            Then    If      DtcIdentifier=ID_ERR_DTC_MIS [1] [NC_CYL_NR]
                                    Then    1st bit of ResultDtc = DTC_PRESENT
                                    Endif
                            Endif
                            For each cylinder bit i which is set in   SYM_CYL_DTC_XX
                                    If      DtcIdentifier = ID_ERR_DTC_MIS[1][i]
                                    Then    1st bit of ResultDtc = DTC_PRESENT
                                    Endif
                            Endfor
                    Endif
            Endif
        Endfor
```

27.27.2 API for reading the status of Diagnostic Trouble Code

27.27.2.1 Request a list of status of DTCs by type of DTC

Description :

Syntax :

```
ACTION_ERRM_ReadInfoByTypeOfDtc (
        IN <TypeOfDtc>,
        IN <InfoIdentifier>,
        INOUT <ListOfDtcInfo>,
        OUT < ResultDtc> )
```

Parameter(in) :	TypeOfDtc	Type of DTC which is requested by the API.
	InfoIdentifier	Information which is returned by the API
Parameter(out) :	ResultDtc	Boolean to say if there is the DTC in memory
Parameter(inout) :	ListOfDtcInfo	Software structure fill up with ListOfDtcInfo

Short Description : This API returns a diagnosis information. All the information returned are related to DTC which have the same type defined by the parameter TypeOfDtc and which are stored in memory.

Diagnosis information returned is defined by InfoIdentifier parameter.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

```
1st bit of ResultDtc = NO_DTC_PRESENT
2nd bit of ResultDtc = NO_BUFFER_FULL
For each failure stored in dynamic memory,
    If    the failure belongs to group defined by the parameters TypeOfDtc
    Then  1st bit of ResultDtc = DTC_PRESENT
          ListOfDtcInfo is fill up with data of this failure corresponding to InfoIdentifier.
    Endif
EndFor
```

During all the software structure filling-in, if the software buffer is full then 2nd bit of ResultDtc is set to BUFFER_FULL, software process is aborted.

27.27.2.2 Request a status of a DTC by DTC number

Description :

Syntax : ACTION_ERRM_ReadInfoByDtc (
IN <DtcIdentifier>,
IN <LevelOfDtc>,
IN <InfoIdentifier>,
INOUT <ListOfDtcInfo>,
OUT <ResultDtc>)

Parameter(in) :	DtcIdentifier	Number of the DTC (identifier) which is requested.
	LevelOfDtc	OBD or customer DTC is requested by the API
	InfoIdentifier	Information which is returned by the API
Parameter(out) :	ResultDtc	Boolean to say if there is the DTC in memory
Parameter(inout) :	ListOfDtcInfo	Software structure fill up with ListOfDtcInfo

Short Description : This API returns a structure of diagnosis information related to a DTC which is stored in dynamic memory.

Diagnosis information returned is defined by InfoIdentifier parameter.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

1st bit of ResultDtc = NO_DTC_PRESENT

2nd bit of ResultDtc = NO_BUFFER_FULL

```
For each failure IDX stored in dynamic memory,
    If   failure is not with status = CARB_MIS
    Then Condition for failure traitment is true
    Endif
    If   NLC_TREAT_DIAG_MIS = 1
    Then Condition for failure traitment is true
    Endif
    If   Condition for failure traitment is true
    Then If     LevelOfDtc = LAW                       (OBD DTC is request)
         Then If     NC_ERR_DTC_REQ_OBD = SYMPTOM   (access by symptom)
              Then For each bit i which is set in ERR_SYM_DTC[IDX]
                           If     DtcIdentifier = ID_ERR_DTC_XX [0][i]
                           Then   1st bit of ResultDtc = DTC_PRESENT
                                  ListOfDtcInfo is fill up with data of this failure
                                  corresponding to InfoIdentifier.
                           Endif
                   Endfor
              Else (access by failure ; NC_ERR_DTC_REQ_OBD = FAILURE)
                   If     DtcIdentifier = ID_ERR_DTC_XX [0][4]
                   Then   1st bit of ResultDtc = DTC_PRESENT
                          ListOfDtcInfo is fill up with data of this failure
                          corresponding to InfoIdentifier.
                   Endif
              Endif
         Else (Customer DTC is request ; LevelOfDtc = CUS)
              If     NC_ERR_DTC_REQ_CUS = SYMPTOM   (access by symptom)
              Then For each bit i which is set in ERR_SYM_DTC[IDX]
                   If     DtcIdentifier = ID_ERR_DTC_XX [NC_NR_DTC_FMT][i]
                   Then   1st bit of ResultDtc = DTC_PRESENT
                          ListOfDtcInfo is fill up with data of this failure
                          corresponding to InfoIdentifier.
                   Endif
                   Endfor
              Else (access by failure ; NC_ERR_DTC_REQ_CUS = FAILURE)
                   If     DtcIdentifier = ID_ERR_DTC_XX
                          [NC_NR_DTC_FMT] [5-NC_NR_DTC_FMT]
                   Then   1st bit of ResultDtc = DTC_PRESENT
                          ListOfDtcInfo is fill up with data of this failure
                          corresponding to InfoIdentifier.
                   Endif
              Endif
         Endif
```

```
Else        (particular case for misfiring failure NLC_TREAT_DIAG_MIS = 0)
    XX = F (IDX) with XX = MIS_A, MIS_B1 or MIS_B4
    If      LevelOfDtc = LAW                      (OBD DTC is request)
    Then    If      SYM_CYL_DTC_XX [13]=1         (Multiple cylinder)
            Then    If      DtcIdentifier=ID_ERR_DTC_MIS [0] [NC_CYL_NR]
                    Then    1st bit of ResultDtc = DTC_PRESENT
                            ListOfDtcInfo is fill up with data of this failure
                            corresponding to InfoIdentifier.
                    Endif
            Endif
            If      SYM_CYL_DTC_XX [12]=1         (Random)
            Then    If      DtcIdentifier=ID_ERR_DTC_MIS[0][NC_CYL_NR+1]
                    Then    1st bit of ResultDtc = DTC_PRESENT
                            ListOfDtcInfo is fill up with data of this failure
                            corresponding to InfoIdentifier.
                    Endif
            Endif
            For each cylinder bit i which is set in SYM_CYL_DTC_XX
                    If DtcIdentifier = ID_ERR_DTC_MIS[0][i]
                    Then    1st bit of ResultDtc = DTC_PRESENT
                            ListOfDtcInfo is fill up with data of this failure
                            corresponding to InfoIdentifier.
                    Endif
            Endfor
```

```
                Else    (Customer DTC is request ; LevelOfDtc = CUS)
                        If      SYM_CYL_DTC_XX [13]=1           (Multiple cylinder)
                        Then If         DtcIdentifier=ID_ERR_DTC_MIS [1] [NC_CYL_NR]
                             Then       1st bit of ResultDtc = DTC_PRESENT
                                        ListOfDtcInfo is fill up with data of this failure
                                        corresponding to InfoIdentifier.
                             Endif
                        Endif
                        If      SYM_CYL_DTC_XX [12]=1           (Random)
                        Then If         DtcIdentifier=ID_ERR_DTC_MIS[1][NC_CYL_NR+1]
                             Then       1st bit of ResultDtc = DTC_PRESENT
                                        ListOfDtcInfo is fill up with data of this failure
                                        corresponding to InfoIdentifier.
                             Endif
                        Endif
                        For each cylinder bit i which is set in SYM_CYL_DTC_XX
                             If DtcIdentifier = ID_ERR_DTC_MIS[1][i]
                             Then       1st bit of ResultDtc = DTC_PRESENT
                                        ListOfDtcInfo is fill up with data of this failure
                                        corresponding to InfoIdentifier.
                             Endif
                        Endfor
                Endif
        Endif
Endfor
```

During all the software structure filling-in, if the software buffer is full then 2nd bit of ResultDtc is set to BUFFER_FULL, software process is aborted.

When a failure is found in dynamic memory with the right DtcIdentifier, the treatment is stopped. That means that only the status of the first DTC found (oldest DTC) is returned.

27.27.2.3 Request a DTCLevel by an another DTCLevel

Description :

Syntax :

```
ACTION_ERRM_ReadDtcLevelByDtcLevel (
        IN <DtcIdentifier>,
        IN <LevelOfDtc>,
        INOUT <ListOfDtc>,
        OUT <ResultDtc> )
```

Parameter(in) :	DtcIdentifier	Number of the DTC (identifier) which is requested.
	LevelOfDtc	OBD or customer DTC is requested by the API
Parameter(out) :	ResultDtc	Boolean to say if there is the DTC in memory
Parameter(inout) :	ListOfDtcInfo	Software structure fill up with ListOfDtcInfo

Short Description : This API returns the customer DTC from the Law DTC or the DTC Law from the DTC customer following the 'levelofDTC' parameter.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

1^{st} bit of ResultDtc = NO_DTC_PRESENT

2^{nd} bit of ResultDtc = NO_BUFFER_FULL

```
For each failure IDX stored in dynamic memory,
    If failure is not with status = CARB_MIS
    Then Condition for failure traitment is true
    Endif
    If NLC_TREAT_DIAG_MIS = 1
    Then Condition for failure traitment is true
    Endif
    If Condition for failure traitment is true
    Then If LevelOfDtc = LAW                                (OBD DTC is request)
         Then If NC_ERR_DTC_REQ_OBD = SYMPTOM    (access by symptom)
              Then For each bit i which is set in ERR_SYM_DTC[IDX]
                        If   DtcIdentifier = ID_ERR_DTC_XX [0][i]
                        Then 1st bit of ResultDtc = DTC_PRESENT
                             ListOfDtcInfo is fill up with
                             If   NC_NR_DTC_FMT = 0
                             Then ID_ERR_DTC_XX[0][5]
                             Else if NC_ERR_DTC_REQ_CUS=SYMPTOM
```

```
                              Then  ID_ERR_DTC_XX[1][i]
                              Else  ID_ERR_DTC_XX[1][4]
                              Endif
                        Endif
                        Exit
                  Endif
            Endfor

      Else  (access by failure ; NC_ERR_DTC_REQ_OBD = FAILURE)

            If    DtcIdentifier = ID_ERR_DTC_XX [0][4]
            Then  1st bit of ResultDtc = DTC_PRESENT
                  ListOfDtc is fill up with
                  If    NC_ERR_DTC_REQ_CUS = SYMPTOM
                        (access by symptom for the customer code)
                  Then  If    only one bit is set in ERR_SYM_MEM[IDX]
                        Then  i = bit number which is set
                              in ERR_SYM_MEM[IDX]
                              ID_ERR_DTC_XX[NC_NR_DTC_FMT][i]
                        Else(many or no bit is set in ERR_SYM_MEM[IDX])
                            ID_ERR_DTC_XX
                            [NC_NR_DTC_FMT][5-NC_NR_DTC_FMT]
                        Endif
                  else  (access by failure for the customer code)
                        ID_ERR_DTC_XX
                        [NC_NR_DTC_FMT][5-NC_NR_DTC_FMT]
                  Endif
            Endif

      Endif

Else  (Customer DTC is request ; LevelOfDtc = CUS)

      If    NC_ERR_DTC_REQ_CUS = SYMPTOM    (access by symptom)

      Then  For each bit i which is set in ERR_SYM_DTC[IDX]
            If    DtcIdentifier = ID_ERR_DTC_XX [NC_NR_DTC_FMT][i]
            Then  1st bit of ResultDtc = DTC_PRESENT
                  ListOfDtcInfo is fill up with
                  If    NC_NR_DTC_FMT = 0
                  Then  ID_ERR_DTC_XX[0][4]
                  Else  if    NC_ERR_DTC_REQ_OBD = SYMPTOM
                        Then  ID_ERR_DTC_XX[0][i]
                        Else  ID_ERR_DTC_XX[0][4]
                        Endif
                  Endif
                  Exit
            Endif
            Endfor

      Else  (access by failure ; NC_ERR_DTC_REQ_CUS = FAILURE)

            If    DtcIdentifier = ID_ERR_DTC_XX
                  [NC_NR_DTC_FMT] [5-NC_NR_DTC_FMT]

            Then  1st bit of ResultDtc = DTC_PRESENT
                  ListOfDtc is fill up with
```

```
                If      NC_ERR_DTC_REQ_OBD = SYMPTOM
                        (access by symptom for the DTC law)
                        If      only one bit is set in ERR_SYM_MEM[IDX]
                        Then    i = bit number which is set
                                in ERR_SYM_MEM[IDX]
                                ID_ERR_DTC_XX[0][i]
                        Else    (many or no bit is set
                                in ERR_SYM_MEM[IDX])
                                ID_ERR_DTC_XX[0][4]
                        Endif
                else    (access by failure for the customer code)
                        ID_ERR_DTC_XX[0][4]
                Endif

            Endif

        Endif

    Endif


Else            (particular case for misfiring failure)

    XX = F (IDX) with XX = MIS_A, MIS_B1 or MIS_B4

    If      LevelOfDtc = LAW                        (OBD DTC is request)

    Then    If      SYM_CYL_DTC_XX [12]=1  (Random)
            Then    If      DtcIdentifier=ID_ERR_DTC_MIS [0] [NC_CYL_NR+1]
                    Then    1st bit of ResultDtc = DTC_PRESENT
                            ListOfDtc is fill up with ID_ERR_DTC_MIS[1]
                            [NC_CYL_NR+1]
                            Exit
                    Endif
            Endif
            If      SYM_CYL_DTC_XX [13]=1  (Multiple cylinder)
            Then    If      DtcIdentifier=ID_ERR_DTC_MIS [0] [NC_CYL_NR]
                    Then    1st bit of ResultDtc = DTC_PRESENT
                            ListOfDtc is fill up with ID_ERR_DTC_MIS[1]
                            [NC_CYL_NR]
                            Exit
                    Endif
            Endif

            For each cylinder bit i which is set in         SYM_CYL_DTC_XX

                    If DtcIdentifier = ID_ERR_DTC_MIS[0][i]

                    Then    1st bit of ResultDtc = DTC_PRESENT
                            ListOfDtc is fill up with ID_ERR_DTC_MIS[1][i]
                            Exit
                    Endif
            Endfor
    Else    (Customer DTC is request ; LevelOfDtc = CUS)

            If      SYM_CYL_DTC_XX [12]=1  (Random)
            Then    If      DtcIdentifier=ID_ERR_DTC_MIS [1] [NC_CYL_NR+1]
                    Then    1st bit of ResultDtc = DTC_PRESENT
```

```
                    ListOfDtc is fill up with ID_ERR_DTC_MIS[0]
                    [NC_CYL_NR+1]
                    Exit
                Endif
            Endif
            If      SYM_CYL_DTC_XX [13]=1  (Multiple cylinder)
            Then If     DtcIdentifier=ID_ERR_DTC_MIS [1] [NC_CYL_NR]
                 Then   1st bit of ResultDtc = DTC_PRESENT
                        ListOfDtc is fill up with ID_ERR_DTC_MIS[0]
                        [NC_CYL_NR]
                        Exit
                 Endif
            Endif
            For each cylinder bit i which is set in     SYM_CYL_DTC_XX
                 If DtcIdentifier = ID_ERR_DTC_MIS[1][i]
                 Then   1st bit of ResultDtc = DTC_PRESENT
                        ListOfDtc is fill up with ID_ERR_DTC_MIS[0][i]
                        Exit
                 Endif
            Endfor
        Endif
    Endif
Endfor
```

During all the software structure filling-in, if the software buffer is full then 2nd bit of ResultDtc is set to BUFFER_FULL, software process is aborted.

When a failure is found in dynamic memory with the right DtcIdentifier, the treatment is stopped. That means that only the status of the first DTC found (oldest DTC) is returned.

27.27.3 API for reading the number of Diagnostic Trouble Code

27.27.3.1 Request the quantity of DTC with a certain type stored in memory

Description :

```
Syntax :           ACTION_ERRM_ReadQuantityOfDtc (
                         IN <TypeOfDtc>,
                         IN <LevelOfDtc>,
                         INOUT <Quantity>,
                         OUT <ResultQuantity> )
```

Parameter(in) :	TypeOfDtc	Type of DTC which is requested by the API.
	LevelOfDtc	OBD or customer DTC is requested by the API
Parameter(out) :	ResultQuantity	Boolean to say if there is at least 1 DTC in memory
Parameter(inout) :	Quantity	Software structure fill up with quantity of Dtc

Short Description : This API returns the quantity of DTCs with a certain type, which are stored in memory. Quantity of DTCs is the number of failures in memory. If failures have same DTC values then they are counted once.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

Quantity = 0

ResultQuantity = NO_DTC_PRESENT

```
For each failure IDX stored in dynamic memory,
    If      the failure belongs to failures group defined by the parameter TypeOfDtc
    Then    ResultQuantity = DTC_PRESENT
            If  failure is not with status = CARB_MIS
            Then Condition for failure traitment is true
            Endif
            If  NLC_TREAT_DIAG_MIS = 1
            Then Condition for failure traitment is true
            Endif
            If  Condition for failure traitment is true
            Then  If     LevelOfDtc = LAW                        (OBD DTC is request)
                  Then   If     NC_REQ_OBD = SYMPTOM             (access by symptom)
                         Then   Quantity  = Quantity + j
                                with j = number of bit set in ERR_SYM_DTC_IDX
                         Else   (access by failure ; NC_REQ_OBD = FAILURE )
                                Quantity = Quantity +1
                         Endif
                  Else   (Customer DTC is request ; LevelOfDtc = CUS)
                         If     NC_REQ_CUS = SYMPTOM             (access by symptom)
                         Then   Quantity  = Quantity + j
                                with j = number of bit set in ERR_SYM_DTC_IDX
                         Else   (access by failure ; NC_REQ_OBD = FAILURE )
                                Quantity = Quantity +1
                         Endif
                  Endif

            Else  (particular case for misfiring failure)
                  XX = F (IDX) with XX = MIS_A, MIS_B1 or MIS_B4
                  If     LevelOfDtc = LAW                        (OBD DTC is request)
                  Then   If     SYM_CYL_DTC_XX [12]=1  (Random)
                         Then   Quantity = Quantity +1
                         Endif
                         If     SYM_CYL_DTC_XX [13]=1  (Multiple cylinder)
                         Then   Quantity = Quantity +1
                         Endif
                         If     (many cylinder bit is set in SYM_CYL_DTC_XX)
                                Quantity = Quantity + j
                                With j   = number of cylinder bit set in SYM_CYL_DTC_XX
                         Endif
```

```
                    Else   (Customer DTC is request ; LevelOfDtc = CUS)
                        If     SYM_CYL_DTC_XX [12]=1  (Random)
                        Then   Quantity = Quantity +1
                        Endif
                        If     SYM_CYL_DTC_XX [13]=1  (Multiple cylinder)
                        Then   Quantity = Quantity +1
                        Endif
                        If     (many cylinder bit is set in SYM_CYL_DTC_XX)
                               Quantity = Quantity + j
                               With j  = number of cylinder bit set in SYM_CYL_DTC_XX
                        Endif
                    Endif
                Endif
            Endif
        EndFor
```

For each duplicated DTC, the corresponding quantity will be subtracted. Duplicated DTC will be counted once.

27.27.4 API for reading the Freeze Frame

27.27.4.1 Request a freeze frame with a certain type by DTC

Description :

Syntax : ACTION_ERRM_ReadFrfByDtc(

```
ACTION_ERRM_ReadFrfByDtc(
        IN <TypeOfFF>,
        IN <FFIdentifier>,
        IN <DtcIdentifier>,
        IN < LevelOfDtc>
        INOUT <Frf>
        OUT <ResultFrf> )
```

Parameter(in) :	TypeOfFF	Type of freeze frame which are request by the API
	FFIdentifier	Identifier of freeze frame requested
	DtcIdentifier	Identifier of DTC of the freeze frame requested
	LevelOfDtc	OBD or customer DTC is requested by the API
Parameter(out) :	ResultFrf	Boolean to say if requested freeze frame is present
Parameter(inout) :	Frf	Software structure fill with Freeze frame

Short Description : This API returns a freeze frame or a part of freeze frame.

With the parameter TypeOfFF, some different part of a freeze frame (defined by the law or by the customer - including set or not) can be returned.

The parameter FFIdentifier is not used (reserved for future improvement).

The freeze frame can be requested :

- by DTC ; in this case the parameters DtcIdentifier and LevelOfDtc should be fulfilled.
- for the CARB/EOBD request (mode 02h) ; in this case, the parameter DtcIdentifier has to be set to NO_DTC, the parameter TypeOfFF has to be set to LAW, and the parameter FFIdentifier has to be set to FIRST.

  The freeze frame associated to the oldest Carb failure among highest priority Carb failure is returned.

See definitions of freeze frame in module “Environmental data” module in “Error management” aggregate.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

1st bit of ResultFrf = NO_FRF_PRESENT

2nd bit of ResultFrf = NO_BUFFER_FULL

```
If        (FFIdentifier = FIRST and TypeOfFF = LAW and DtcIdentifier = NO_DTC)
Then      (Law freeze frame is requested)
          If      NLC_OBD_FRF_PND = 0
          Then
                  Find oldest Carb/EOBD failure (emission relevant, LC_OBD_ERR = 1) failure stored in dynamic memory and confirmed (LV_ERR_CFM[IDX] = 1). If a misfire or fuel system failure is in the memory with LV_ERR_CFM[IDX] = 1, then the oldest of these failures must be chosen.
          Else
                  Find oldest Carb/EOBD failure (emission relevant, LC_OBD_ERR = 1) failure stored in dynamic memory and confirmed (LV_ERR_CFM[IDX] = 1). If a misfire or fuel system failure is in the memory with LV_ERR_CFM[IDX] = 1, then the oldest of these failures must be chosen.

                  If this failure isn't existing, find oldest Carb/EOBD failure (emission relevant, LC_OBD_ERR = 1) failure stored in dynamic memory and pending (LV_ERR_PND[IDX] =1). If a misfire or fuel system failure is in the memory with LV_ERR_PND[IDX] = 1, then the oldest of these failures must be chosen.
          Endif
          If  failure is not with status = CARB_MIS
          Then Condition for failure traitment is true
          Endif
          If  NLC_TREAT_DIAG_MIS = 1
          Then Condition for failure traitment is true
          Endif
          If  Condition for failure traitment is true
          Then  If      NC_ERR_DTC_REQ_OBD = SYMPTOM   (access by symptom)
                Then  If      only one bit is set in ERR_SYM_MEM[IDX]
                      Then    i = bit number which is set in ERR_SYM_MEM[IDX]
                              ReturnDtc = ID_ERR_DTC_XX [0][i]
                      Else    (many or no bit is set in ERR_SYM_MEM[IDX])
                              ReturnDtc = ID_ERR_DTC_XX [0][4]
                      Endif
                Else  (access by failure ; NC_ERR_DTC_REQ_OBD = FAILURE)
                      ReturnDtc = ID_ERR_DTC_XX [0][4]
                Endif
          Else  (particular case for misfiring failure)
                XX = F (IDX) with XX = MIS_A, MIS_B1 or MIS_B4
                If      SYM_CYL_MEM_XX [12]=1 (Random)
                Then    ReturnDtc = ID_ERR_DTC_MIS[0][NC_CYL_NR+1]
```

```
                    Else  if    SYM_CYL_MEM_XX [13]=1 (Multiple cylinder)
                                Or many cylinder bits are set in
                          Then  ReturnDtc = ID_ERR_DTC_MIS[0][NC_CYL_NR]
                          Else  (only one cylinder bit is set in SYM_CYL_MEM_XX)
                                i = bit number which is set in SYM_CYL_MEM_XX
                                ReturnDtc = ID_ERR_DTC_MIS[0][i]
                          Endif
                          Endif
              Endif
              Software structure is fill up with ReturnDtc + ENVD_OBD[u][IDX]
              1st bit of ResultFrf = FRF_PRESENT

  Else        (others freeze frame is requested)
      For each failure IDX stored in dynamic memory,
              If  failure is not with status = CARB_MIS
              Then Condition for failure traitment is true
              Endif
              If  NLC_TREAT_DIAG_MIS = 1
              Then Condition for failure traitment is true
              Endif
              If  Condition for failure traitment is true
              Then  If    LevelOfDtc = LAW                  (OBD DTC is request)
                    Then  If NC_ERR_DTC_REQ_OBD=SYMPTOM  (access by symptom)
                          Then  For each bit i which is set in ERR_SYM_DTC[IDX]
                                      If DtcIdentifier = ID_ERR_DTC_XX[0][i]
                                      Then  1st bit of ResultFrf = FRF_PRESENT
                                            Software structure is filled-up with
                                            DtcIdentifier plus freeze frame of failure
                                            according parameter TypeOfFF
                                            Exit
                                      Endif
                                Endfor
                          Else  (access by failure; NC_ERR_DTC_REQ_OBD = FAILURE)
                                If    DtcIdentifier = ID_ERR_DTC_XX [0][4]
                                Then  1st bit of ResultFrf = FRF_PRESENT
                                      Software structure is filled-up with DtcIdentifier plus
                                      freeze frame of failure according parameter
                                      TypeOfFF
                                Endif
                          Endif
                    Else  (Customer DTC is request ; LevelOfDtc = CUS)
```

```
                    If NC_ERR_DTC_REQ_CUS=SYMPTOM  (access by symptom)
                    Then  For each bit i which is set in ERR_SYM_DTC[IDX]
                              If DtcIdentifier = ID_ERR_DTC_XX[NC_NR_DTC_FMT][i]
                              Then  1st bit of ResultFrf = FRF_PRESENT
                                    Software structure is filled-up with DtcIdentifier  plus
                                    freeze frame of failure according parameter
                                    TypeOfFF
                                    Exit
                              Endif
                          Endfor
                    Else  (access by failure; NC_ERR_DTC_REQ_CUS = FAILURE)
                              If    DtcIdentifier = ID_ERR_DTC_XX
                                    [NC_NR_DTC_FMT][5-NC_NR_DTC_FMT]
                              Then  1st bit of ResultFrf = FRF_PRESENT
                                    Software structure is filled-up with DtcIdentifier  plus
                                    freeze frame of failure according parameter
                                    TypeOfFF
                              Endif

                    Endif

               Endif

       Else         (particular case for misfiring failure)
               XX = F (IDX) with XX = MIS_A, MIS_B1 or MIS_B4

               If     LevelOfDtc = LAW                    (OBD DTC is request)
               Then If      SYM_CYL_DTC_XX [12]=1  (Random)
                    Then If DtcIdentifier=ID_ERR_DTC_MIS[0][NC_CYL_NR+1]
                         Then  1st bit of ResultFrf = FRF_PRESENT
                               Software structure is filled-up with DtcIdentifier  plus
                               freeze frame of failure according parameter
                               TypeOfFF
                               Exit
                         Endif
                    Endif

                    If       SYM_CYL_DTC_XX [13]=1 (Multiple cylinder)
                    Then If    DtcIdentifier=ID_ERR_DTC_MIS[0][NC_CYL_NR]
                         Then  1st bit of ResultFrf = FRF_PRESENT
                               Software structure is filled-up with DtcIdentifier  plus
                               freeze frame of failure according parameter
                               TypeOfFF
                               Exit
                         Endif
                    Endif
                    For each cylinder bit i which is set in SYM_CYL_DTC_XX

                         If DtcIdentifier = ID_ERR_DTC_MIS[0][i]

                         Then  1st bit of ResultFrf = FRF_PRESENT
                               Software structure is filled-up with DtcIdentifier  plus
                               freeze frame of failure according parameter
                               TypeOfFF
```

```
                        Exit
                    Endif
                Endfor
            Else    (Customer DTC is request ; LevelOfDtc = CUS)
                If      SYM_CYL_DTC_XX [12]=1  (Random)
                Then    If DtcIdentifier=ID_ERR_DTC_MIS[1][NC_CYL_NR+1]
                        Then    1st bit of ResultFrf = FRF_PRESENT
                                Software structure is filled-up with DtcIdentifier  plus
                                freeze frame of failure according parameter
                                TypeOfFF
                                Exit
                        Endif
                Endif

                If       SYM_CYL_DTC_XX [13]=1 (Multiple cylinder)
                Then    If DtcIdentifier=ID_ERR_DTC_MIS[1][NC_CYL_NR]
                        Then    1st bit of ResultFrf = FRF_PRESENT
                                Software structure is filled-up with DtcIdentifier  plus
                                freeze frame of failure according parameter
                                TypeOfFF
                                Exit
                        Endif
                Endif
                For each cylinder bit i which is set in SYM_CYL_DTC_XX

                        If DtcIdentifier = ID_ERR_DTC_MIS[1][i]

                        Then    1st bit of ResultFrf = FRF_PRESENT
                                Software structure is filled-up with DtcIdentifier  plus
                                freeze frame of failure according parameter
                                TypeOfFF
                                Exit
                        Endif
                Endfor
            Endif
        Endif
    Endfor
Endif
```

“Software structure is filled-up with freeze frame of failure according parameter TypeOfFF” means :

(for failure IDX)

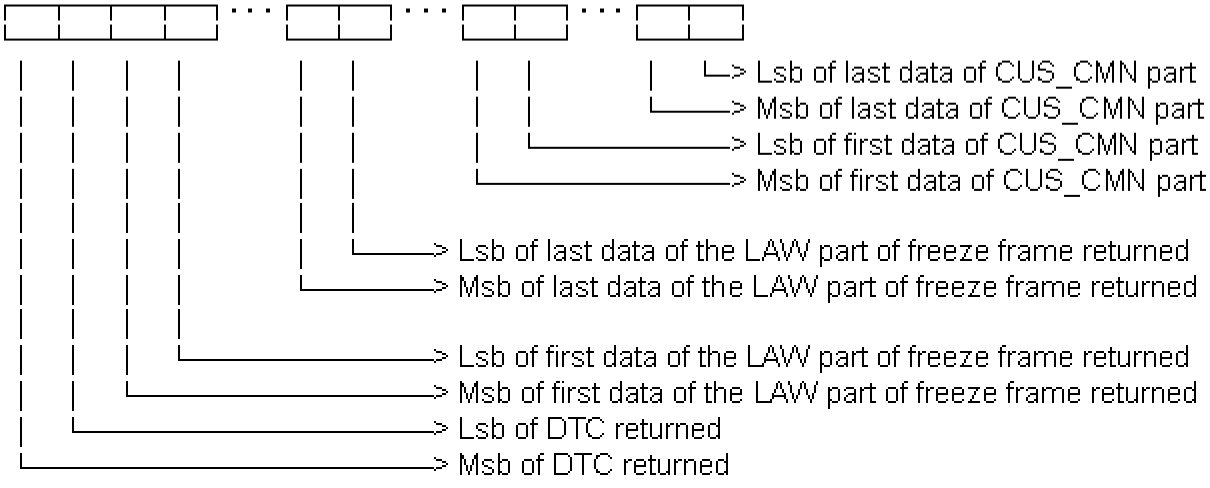
Software structure is filled-up with DTCIdentifier plus the softwate structure defined below :

```
If      TypeOfFF = LAW
Then    Software structure is fill up with ENVD_OBD[u][IDX]
Endif
If      TypeOfFF = CUS_CMN
Then    Software structure is fill up with ENVD_CUS_CMN[v][IDX]
Endif
If      TypeOfFF = CUS_SET
Then    Software structure is fill up with
                ENVD_CUS_SET_CMN[w][z][IDX] + ENVD_CUS_SET_SPC[y][z][IDX]
Endif
```

Combination of parameter Type OfFF is possible. For instance if TypeOfFF is equal to LAW and CUS_CMN, both part are returned. In case of multiple part of freeze frame to returned, , the following order is respected :

- Part LAW of the freeze frame is returned first ;
- Part CUS_CMN of the freeze frame is then returned ;
- Part CUS_SET of the freeze frame is then returned .

Example : In the case of a request with TypeOfFF = LAW and CUS_CMN then the following structure is returned :



When a failure is found in dynamic memory with the right DtcIdentifier, the treatment is stopped. That means that only the freeze frame of the first DTC is found (oldest DTC) and returned by the API.

During all the software structure filling-in, if the software buffer is full then 2nd bit of ResultFrf is set to BUFFER_FULL, software process is aborted.

27.27.5 API for erasing Diagnostic information

27.27.5.1 Clear the failure associated to DTC with a certain type.

Description :

Syntax : ACTION_ERRM_ClrInfoByTypeOfDtc (
IN <TypeOfDtc>,
OUT < ResultClrInfo>)

Parameter(in) : TypeOfDtc Type of DTC which is requested by the API

Parameter(out) : ResultClrInfo Indicate if some DTC are cleared or not

Short Description : This API clear all the failure in dynamic memory associated to DTC with a certain type. It permits also to reinitialised all readiness code.

Remark :

For reason of design, and time resource, it’s recommended to activate this API when engine stop. (Ref. SAE Mode $04).

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

```
ResultClrInfo = INFO_NOT_CLEARED
For each failure stored in dynamic memory,
    If      the failure belongs to failures group defined by the parameter TypeOfDtc
    Then    ACTION_ERRM_EraseErr ( IN<IDX>, SYNCHRONIZATION<CALL> )
            All data in static memory related to failure XX (with XX = F(IDX)) is also cleared
            ResultClrInfo = INFO_CLEARED
    Endif
Endfor
```

In the case TypeOfDTC = All, then, the whole static memory shall be cleared (same behavior as clear failure memory using LC_ERR_FMY_CLR calibration bit)

{ All readiness bit are reinitialised }

ACTION_ERRM_InitReadiness (SYNCHRONIZATION<CALL>)

Retransmit “failure erase service received”

27.27.5.2 Clear the failure associated to a DTC

Description :

Syntax : ACTION_ERRM_ClrInfoByDtc (
IN <DtcIdentifier>,
IN <LevelOfDtc>,
OUT < ResultClrInfo>)

Parameter(in) :	DtcIdentifier	Number of DTC which is requested (identifier)
	LevelOfDtc	OBD or customer DTC is requested by the API
Parameter(out) :	ResultClrInfo	to say if some DTC are cleared or not

Short Description : This API clear all the failure in dynamic memory associated to a DTC. The erasing is also applicable when there are identical DTC identifiers in the memory.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

```
ResultClrInfo = INFO_NOT_CLEARED
{ All readiness bit are reinitialised }
ACTION_ERRM_InitReadiness ( SYNCHRONIZATION<CALL> )
For each failure stored in dynamic memory do
    If  failure is not with status = CARB_MIS
    Then Condition for failure traitment is true
    Endif
    If  NLC_TREAT_DIAG_MIS = 1
    Then Condition for failure traitment is true
    Endif
    If  Condition for failure traitment is true
    Then  If     LevelOfDtc = LAW                          (OBD DTC )
          Then   If     NC_ERR_DTC_REQ_OBD = SYMPTOM   (access by symptom)
                 Then   For each bit i which is set in ERR_SYM_DTC[IDX]
                            If DtcIdentifier = ID_ERR_DTC_XX[0][i]
                            Then   ACTION_ERRM_EraseErr ( IN<IDX>,
                                   SYNCHRONIZATION<CALL> )
                                   All data in static memory related to failure XX (with
                                   XX = F(IDX)) is also cleared
                                   ResultClrInfo = INFO_CLEARED
                                   Exit
                            Endif
                        Endfor
```

```
          Else { access by failure : NC_ERR_DTC_REQ_OBD = FAILURE }
               If   DtcIdentifier = ID_ERR_DTC_XX[ 0 ][ 4 ] )
               Then ACTION_ERRM_EraseErr( IN<IDX>, SYNCHRONIZATION<CALL> )
                         All data in static memory related to failure XX (with XX = F(IDX)) is also cleared
                    ResultClrInfo = INFO_CLEARED
               Endif
          Endif
     Else ( Customer DTC : LevelOfDtc = CUSTOMER)
          If   NC_ERR_DTC_REQ_CUS = SYMPTOM    (access by symptom)
          Then For each bit i which is set in ERR_SYM_DTC[IDX]

                    If DtcIdentifier = ID_ERR_DTC_XX[NC_NR_DTC_FMT][i]

                    Then ACTION_ERRM_EraseErr ( IN<IDX>, SYNCHRONIZATION<CALL> )
                         All data in static memory related to failure XX (with XX = F(IDX)) is also cleared
                         ResultClrInfo = INFO_CLEARED
                         Exit
                    Endif
               Endfor
          Else (access by failure : NC_ERR_DTC_REQ_OBD = FAILURE)
               If   DtcIdentifier = ID_ERR_DTC_XX [NC_NR_DTC_FMT][5-NC_NR_DTC_FMT]
               Then ACTION_ERRM_EraseErr ( IN<IDX>, SYNCHRONIZATION<CALL> )
                    All data in static memory related to failure XX (with XX = F(IDX)) is also cleared
                    ResultClrInfo = INFO_CLEARED
               Endif
               Endif
     Endif
Else (particular case for misfiring failure)

     XX = F (IDX) with XX = MIS_A, MIS_B1 or MIS_B4

     If   LevelOfDtc = LAW                          (OBD DTC)

     Then If   SYM_CYL_DTC_XX [12]=1                (Random)
          Then If   DtcIdentifier=ID_ERR_DTC_MIS [0] [NC_CYL_NR+1]
               Then ACTION_ERRM_EraseErr ( IN<IDX>, SYNCHRONIZATION<CALL> )
                    All data in static memory related to failure XX (with XX = F(IDX)) is also cleared
                    ResultClrInfo = INFO_CLEARED
                    Exit
               Endif
          Endif

          If   SYM_CYL_DTC_XX [13]=1                (Multiple cylinder)
          Then If   DtcIdentifier=ID_ERR_DTC_MIS [0] [NC_CYL_NR]
               Then ACTION_ERRM_EraseErr ( IN<IDX>,
```

```
            SYNCHRONIZATION<CALL> )
            All data in static memory related to failure XX (with XX =
            F(IDX)) is also cleared
            ResultClrInfo = INFO_CLEARED
            Exit
        Endif
    Endif
    For each cylinder bit i which is set in         SYM_CYL_DTC_XX

        If DtcIdentifier = ID_ERR_DTC_MIS[0][i]

        Then ACTION_ERRM_EraseErr( IN<IDX>,
            SYNCHRONIZATION<CALL> )
            All data in static memory related to failure XX (with XX =
            F(IDX)) is also cleared
            ResultClrInfo = INFO_CLEARED
            Exit
        Endif
    Endfor
Else     (Customer DTC )

    If      SYM_CYL_DTC_XX [12]=1         (Random)
    Then    If    DtcIdentifier=ID_ERR_DTC_MIS [1] [NC_CYL_NR+1]
        Then ACTION_ERRM_EraseErr ( IN<IDX>,
            SYNCHRONIZATION<CALL> )
            All data in static memory related to failure XX (with XX =
            F(IDX)) is also cleared
            ResultClrInfo = INFO_CLEARED
            Exit
        Endif
    Endif

    If      SYM_CYL_DTC_XX [13]=1         (Multiple cylinder)
    Then    If      DtcIdentifier=ID_ERR_DTC_MIS [1] [NC_CYL_NR]
        Then ACTION_ERRM_EraseErr ( IN<IDX>,
            SYNCHRONIZATION<CALL> )
            All data in static memory related to failure XX (with XX =
            F(IDX)) is also cleared
            ResultClrInfo = INFO_CLEARED
            Exit
        Endif
    Endif
    For each cylinder bit i which is set in         SYM_CYL_DTC_XX

        If DtcIdentifier = ID_ERR_DTC_MIS[1][i]

        Then ACTION_ERRM_EraseErr ( IN<IDX>,
            SYNCHRONIZATION<CALL> )
            All data in static memory related to failure XX (with XX =
            F(IDX)) is also cleared
            ResultClrInfo = INFO_CLEARED
            Exit
        Endif
    Endfor
Endif
```

```
        Endif
    Endfor
```

When a failure is found in dynamic memory with the right DtcIdentifier, the treatment is not stopped. That means that all failure with the right DTC are erased.

27.27.6 API to manage the entry or exit of “Marked Mode”

27.27.6.1 Activate/deactivate the marked mode with clear failure

Description :

Syntax : ACTION_ERRM_ControlDtcSettings (IN <MarkedMode>)

Parameter(in) : MarkedMode State of marked mode

Short Description : This API permits to activate or deactivate the marked mode. In this mode all new failure which appears are marked and then erased when we leave this mode (useful in End Of Line phase). For errm inhibition, please refer to ACTION_ERRM_ActivateMarkedMode ().

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

```
If    MarkedMode = ON
Then  LV_MKD_MOD = 1
Else  LV_MKD_MOD = 0
      For each failure (IDX) stored in dynamic memory,
         If    LV_ERR_MKD[IDX] = 1
         Then  ACTION_ERRM_EraseErr ( IN<IDX>, SYNCHRONIZATION<CALL> )
               All data in static memory related to failure XX (with XX = F(IDX)) is also cleared
         Endif
      EndFor
      Reset all diagnoses :
         - For all diagnoses using filterings (anti-bounce, statistical and multi-condition filtering) clear all counters (anti-bounce,statistical and multicondition counters, end of diagnosis counters), LV_ERR_XX, ERR_SYM_XX, and LV_END_DIAG_XX.
         - For diagnoses without any filtering, reset the diagnoses, LV_ERR_XX and LV_END_DIAG_XX.
Endif
```

27.27.6.2 Activate/deactivate the marked mode without clearing failures

Description :

Syntax : ACTION_ERRM_ActivateMarkedMode (IN <MarkedMode>)

Parameter(in) : MarkedMode State of marked mode ON or OFF

Short Description : This API shall be used to activate/deactivate the marked mode. In this mode, each new occuring failure is marked.
This is useful for ERRM inhibition. This inhibition may be used during a reprogramming phase.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

```
If    MarkedMode = ON
Then     LV_MKD_MOD = 1
Else     LV_MKD_MOD = 0
Endif
```

27.27.6.3 Deactivation of marked mode within driving cycle

FUNCTION DESCRIPTION:

General information:

Purpose of this module is to add a security mechanism regarding marked mode. Because marked mode permits to inhibit error management, a strategy shall be introduced to avoid wrong inhibition.

This strategy is to deactivate marked mode when a driving cycle is detected. After driving cycle ends, reactivation of marked mode is the not supported.

Description:

At a low recurrency, a pulling is made to check if driving cycle conditions are met. If met (LV_MKD_MOD=ON **and** LV_DC=1), market mode is deactivated. An explicit request to activate marked mode shall be done.

Signal flow diagram:

Application conditions:

Initialisation: none

Recurrence: 1 second

Activation: LV_MKD_MOD=ON **and** LV_DC=1

Deactivation: none

Formula section:

```
MarkedMode = OFF
ACTION_ERRM_ActivateMarkedMode (   IN <MarkedMode>,
                                     SYNCHRONIZATION<CALL> )
```

27.27.7 API for reading readiness code

27.27.7.1 Read the readiness code information

Description :

Syntax : ACTION_ERRM_ReadReadinessCode (
INOUT <ReadinessCode>,
OUT <ResultReadinessCode>)

Parameter(inout) :	ReadinessCode	Software structure fill with readiness code

Short Description : This API calculate and returns the readiness code information in the software structure.

Readiness code information are build from readiness flags.

Please see “Readiness Code” module of the “Error management” aggregate for more details.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

ResultReadinessCode = NEGATIVE_RESPONSE

(Calculation of STATE_READY_OBD_1)

STATE_READY_OBD_1 :

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 |  |  |  |  |  |  |  └─>  bit0 : Misfire monitoring supported
 |  |  |  |  |  |  └───>  bit1 : Fuel system monitoring supported
 |  |  |  |  |  └─────>  bit2 : Comprehensive component monitoring supported
 |  |  |  |  └───────>  bit3 : Reserved
 |  |  |  └─────────>  bit4 : Misfire monitoring status
 |  |  └───────────>  bit5 : Fuel system monitoring status
 |  └─────────────>  bit6 : Comprehensive component monitoring status
 └───────────────>  bit7 : Reserved
```

4 less significant bit of STATE_READY_OBD_1 = 4 less significant bit of C_STATE_READY_OBD_1
4 most significant bit of STATE_READY_OBD_1 = 4 less significant bit of C_STATE_READY_OBD_1

STATE_READY_OBD_2:

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 |  |  |  |  |  |  |  └─>  bit 0 : Catalyst monitoring
 |  |  |  |  |  |  └───>  bit 1 : Heated catalyst
 |  |  |  |  |  └─────>  bit 2 : Evaporative system monitoring
 |  |  |  |  └───────>  bit 3 : Secondary air system monitoring
 |  |  |  └─────────>  bit 4 : A/C system refrigerant monitoring
 |  |  └───────────>  bit 5 : Oxygen sensor monitoring
 |  └─────────────>  bit 6 : Oxygen sensor heater monitoring
 └───────────────>  bit 7 : EGR system monitoring
```

STATE_READY_OBD_2 = C_STATE_READY_OBD_2

The readiness code calculation is performed in two steps:

1. setting of readiness of a specific group CARB_YY (e.g. CARB_SA, CARB_EVAP), depending on the dynamic Error Management. If at least one failure XX is confirmed with bit 4 of WAL_CONF_XX = 1 of the group CARB_YY, then the readiness of this group can be set to ready.
2. setting of readiness of a specific group CARB_YY, depending on the failure XX individual readiness information LV_READY_XX. E.g. in case of a failure free system only step 2 is used for the readiness code calculation.

Step 1:

```
If CTR_ERR_DYN_NR != 0        (at least one failure stored in Error Management)
Then  For IDX = 0 to CTR_ERR_DYN_NR – 1
          If      bit 4 of WAL_CONF_XX = 1 (failure is emission relevant)
                  with XX = DIAG_INST[IDX]
          and     LV_ERR_CFM[IDX] = 1              (failure is confirmed)
          Then    case CARB_YY for failure XX
                  CARB_MIS:   Bit 4 of STATE_READY_OBD_1 = 0
                  CARB_FSD:   Bit 5 of STATE_READY_OBD_1 = 0
                  CARB_CC:    Bit 6 of STATE_READY_OBD_1 = 0
                  CARB_CAT:   Bit 0 of STATE_READY_OBD_2 = 0
                  CARB_HC:    Bit 1 of STATE_READY_OBD_2 = 0
                  CARB_EVAP:  Bit 2  of STATE_READY_OBD_2 = 0
                  CARB_SA:    Bit 3 of STATE_READY_OBD_2 = 0
                  CARB_AC:    Bit 4 of STATE_READY_OBD_2 = 0
                  CARB_LS:    Bit 5 of STATE_READY_OBD_2 = 0
```

```
                CARB_LSH:   Bit 6 of STATE_READY_OBD_2 = 0
                CARB_EGR:   Bit 7 of STATE_READY_OBD_2 = 0
            Endif
        Endfor
    Endif
```

Step 2:

```
Misfire monitoring status bit4 :
    If      Bit 4 of C_STATE_READY_CMPL_OBD_1 = 0
    Then
        if  ∑ (LV_READY_XX) = 0
                Take into account all LV_READY_XX which fulfill the following condition:
                this XX failure has the status CARB_MIS (see tables of failures) AND
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
        then    Bit 4 of STATE_READY_OBD_1 = 0
        endif
    Else
            Readiness for Misfire monitoring status is set
            Bit 4 of STATE_READY_OBD_1 = 0
    EndIf

Fuel system monitoring status bit5 :
    If      Bit 5 of C_STATE_READY_CMPL_OBD_1 = 0
    Then
        if  ∑ (LV_READY_XX) = 0
                Take into account all LV_READY_XX which fulfill the following condition:
                this XX failure has the status CARB_FSD (see tables of failures) AND
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
        then    Bit 5 of STATE_READY_OBD_1 = 0
        endif
    Else
            Readiness for Fuel system monitoring status is set
            Bit 5 of STATE_READY_OBD_1 = 0
    EndIf

Comprehensive component monitoring status bit6 :
    If      Bit 6 of C_STATE_READY_CMPL_OBD_1 = 0
    Then
            if      ∑ (LV_READY_XX) = 0
                    Take into account all LV_READY_XX which fulfill the following condition:
                    this XX failure has the status CARB_CC (see tables of failures) AND
                    the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
            then    Bit 6 of STATE_READY_OBD_1 = 0
            endif
    Else
            Readiness for Comprehensive component status is set
            Bit 6 of STATE_READY_OBD_1 = 0
    EndIf
```

Bit 7 of STATE_READY_OBD_1 is unchanged

(Calculation of STATE_READY_OBD_2)

Catalyst monitoring bit 0 :

```
If      ∑ (LV_READY_XX) = 0
                Take into account all LV_READY_XX which fulfill the following condition:
                this XX failure has the status CARB_CAT (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 0 of STATE_READY_OBD_2 = 0
Endif
```

Heated catalyst bit 1 :

```
If      ∑ (LV_READY_XX) = 0
                Take into account all LV_READY_XX which fulfill the following condition:
                this XX failure has the status CARB_HC (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 1 of STATE_READY_OBD_2 = 0
Endif
```

Evaporative system monitoring bit2 :

```
If      ∑ (LV_READY_XX) = 0
                Take into account all LV_READY_XX which fulfill the following condition:
                this XX failure has the status CARB_EVAP(see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 2 of STATE_READY_OBD_2 = 0
Endif
```

Secondary air system monitoring bit3 :

```
If      ∑ (LV_READY_XX) = 0
                Take into account all LV_READY_XX which fulfill the following condition:
                this XX failure has the status CARB_SA (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 3 of STATE_READY_OBD_2 = 0
Endif
```

A/C system refrigerant monitoring bit4 :

```
If      ∑ (LV_READY_XX) = 0
                Take into account all LV_READY_XX which fulfill the following condition:
                this XX failure has the status CARB_AC (see tables of failures) and
                the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 4 of STATE_READY_OBD_2 = 0
Endif
```

Oxygen sensor monitoring bit5 :

```
If      ∑ (LV_READY_XX) = 0
                Take into account all LV_READY_XX which fulfill the following condition:
                this XX failure has the status CARB_LS (see tables of failures) and
```

```
                    the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
    Then   Bit 5 of STATE_READY_OBD_2 = 0
    Endif
```

Oxygen sensor heater monitoring bit6 :

```
    If     LV_READY_XX = 0
                    Take into account all LV_READY_XX which fulfill the following condition:
                    this XX failure has the status CARB_LSH (see tables of failures) and
                    the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
    Then   Bit 6 of STATE_READY_OBD_2 = 0
    Endif
```

EGR system monitoring bit7 :

```
    If     LV_READY_XX = 0
                    Take into account all LV_READY_XX which fulfill the following condition:
                    this XX failure has the status CARB_EGR (see tables of failures) and
                    the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
    Then   Bit 7 of STATE_READY_OBD_2 = 0
    Endif
```

ReadinessCode structure is filled up with : C_STATE_READY_OBD_2, STATE_READY_OBD_1 and STATE_READY_OBD_2.

ResultReadinessCode = POSITIVE_RESPONSE

Definition of Readiness Code:

In C_STATE_READY_OBD_1, a bit set to 1 means that corresponding test is supported according to the J1979 standard.

C_STATE_READY_OBD_1 :

bit0 : Misfire monitoring supported
bit1 : Fuel system monitoring supported
bit2 : Comprehensive component monitoring supported
bit3 = 0 (reserved)
bit4 = 0 (reserved)
bit5 = 0 (reserved)
bit6 = 0 (reserved)
bit7 = 0 (reserved)

With C_STATE_READY_OBD_CMPL_OBD_1, it’s possible to set directly Misfire, Fuel system, and comprehensive component readiness codes to "ready" status.

In C_STATE_READY_OBD_2, a bit set to 1 means that corresponding test is supported according to the J1979 standard.

C_STATE_READY_OBD_2:

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 |  |  |  |  |  |  |  └─>  bit 0 : Catalyst monitoring
 |  |  |  |  |  |  └───>  bit 1 : Heated catalyst
 |  |  |  |  |  └─────>  bit 2 : Evaporative system monitoring
 |  |  |  |  └───────>  bit 3 : Secondary air system monitoring
 |  |  |  └─────────>  bit 4 : A/C system refrigerant monitoring
 |  |  └───────────>  bit 5 : Oxygen sensor monitoring
 |  └─────────────>  bit 6 : Oxygen sensor heater monitoring
 └───────────────>  bit 7 : EGR system monitoring
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_READY_OBD_1	1	0...FFH	0...255	1	[-]
Status configuration for readiness code (continuous tests)					
C_STATE_READY_OBD_2	1	0...FFH	0...255	1	[-]
Status configuration for readiness code (non continuous tests)					
C_STATE_READY_CMPL_OBD_1	1	0...FFH	0...255	1	[-]
Readiness MIS, FSD, CC directly set to ready					

Calibration data detailled description for C_STATE_READY_CMPL_OBD_1 :

Readiness code calculation for CARB_MIS, CARB_FSD, CARB_CC diagnosis :

0: readiness of conserned group is calculated based on the readiness of each comprehensive component diagnostic

1: readiness of conserned group always indicates "ready"

C_STATE_READY_CMPL_OBD_1:

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 |  |  |  |  |  |  |  └─>  bit0 : 0
 |  |  |  |  |  |  └───>  bit1 : 0
 |  |  |  |  |  └─────>  bit2 : 0
 |  |  |  |  └───────>  bit3 : 0
 |  |  |  └─────────>  bit4 : set directly readiness for Misfire monitoring
 |  |  └───────────>  bit5 : set directly readiness Fuel system monitoring
 |  └─────────────>  bit6 : set directly readiness Comprehensive component monitoring
 └───────────────>  bit7 : 0
```

27.27.7.2 Read the readiness code information for customer specific purpose

Description :

Syntax : ACTION_ERRM_ReadReadinessCodCus(
INOUT <ReadinessCode>,
OUT <ResultReadinessCode>)

Parameter(inout):	ReadinessCode	Software structure filled-up with customer specific readiness code

Short Description : This API calculates and returns the readiness code information in the software structure.

Readiness code information are built thanks to readiness flags.

Please see “Readiness Code” module of the ERRM aggregate for more details.

Application conditions:

Deactivation : -

Initialization: -

Recurrence : -

Formula section :

ResultReadinessCode = NEGATIVE_RESPONSE

(Calculation of STATE_READY_OBD_3)

STATE_READY_OBD_3 :

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ └─> bit0 : for diagnosis classified as CARB_OTHER
 │ │ │ │ │ │ └───> bit1 : for diagnosis classified as NO_CARB
 │ │ │ │ │ └─────> bit2 : 0
 │ │ │ │ └───────> bit3 : 0
 │ │ │ └─────────> bit4 : 0
 │ │ └───────────> bit5 : 0
 │ └─────────────> bit6 : 0
 └───────────────> bit7 : 0
```

STATE_READY_OBD_3 = C_STATE_READY_OBD_3

Diagnosis classified as CARB_OTHER bit0:

```
If      ∑ (LV_READY_XX) = 0
            Take into account all LV_READY_XX which fulfill the following condition:
            this XX failure has the status CARB_OTHER (see tables of failures) and
            the bit 4 of WAL_CONF_XX = 1 (CARB/EOBD failure)
Then    Bit 0 of STATE_READY_OBD_3 = 0
Endif
```

Diagnosis classified as NO_CARB bit1

```
If      ∑ (LV_READY_XX) = 0
            Take into account all LV_READY_XX which fulfill the following condition:
            this XX failure has the status NO_CARB (see tables of failures) and
            the bit 4 of WAL_CONF_XX = 0 (CARB/EOBD failure)
Then    Bit 1 of STATE_READY_OBD_3 = 0
Endif
```

ReadinessCode structure is filled-up with : C_STATE_READY_OBD_3, STATE_READY_OBD_3.

ResultReadinessCode = POSITIVE_RESPONSE

Definition of Readiness Code:

This readiness code does not follow existing J1979 standard.

C_STATE_READY_OBD_3 :

```
┌───┬───┬───┬───┬───┬───┬───┬───┐
└───┴───┴───┴───┴───┴───┴───┴───┘
  │   │   │   │   │   │   │   └─>   bit0 : for diagnosis classified as CARB_OTHER
  │   │   │   │   │   │   └─────>   bit1 : for diagnosis classified as NO_CARB
  │   │   │   │   │   └─────────>   bit2 : 0
  │   │   │   │   └─────────────>   bit3 : 0
  │   │   │   └─────────────────>   bit4 : 0
  │   │   └─────────────────────>   bit5 : 0
  │   └─────────────────────────>   bit6 : 0
  └─────────────────────────────>   bit7 : 0
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_READY_OBD_3	1	0...FFH	0...255	1	[-]
Status configuration for customer specific readiness code					

27.27.8 Rate-Based Monitoring - Communication interface

FUNCTION DESCRIPTION:

Description:

The Rate-Based Monitoring statistics shall be reported to the Scan-Tool via the Mode $09.

The data to be transmitted are :

- Ignition cycle counter
- General denominator
- For each group of monitors, individual numerator and denominator of the monitor which has the lowest In-Use Performance ratios.

These statistics are required and transmitted within Mode $09 for the monitors of the following groups :

- Catalyst bank 1
- Catalyst bank 2
- Oxygen sensor upstream bank 1
- Oxygen sensor upstream bank 2
- EGR system and/or VVT system
- Secondary air system
- Evaporating system

The selection of data to be displayed on Scan-Tool shall respect some rules :

- For each group of monitors, the individual numerator and denominator of monitor which has the lowest numerical In-Use Performance ratio shall be reported.
- If two or more specific monitors of the same group have identical ratios, the corresponding numerator and denominator for the specific monitor that has the highest denominator shall be reported.
- Depending on system configurations, some components might not exist (e.g. O2 Sensor for bank 2 on vehicles with 1 bank engine). In this case the numerator and denominator shall be set to 0.

27.27.8.1 API to select Rate-Based Monitoring data to be transmitted via Mode $09

The following paragraph descibes the data selection algorithm to be executed only once (for CPU load saving reason), upon external tool request via Mode $09, by calling an API.

Syntax : ACTION_ERRM_SelectRbmData (INOUT < ListRbmData >,
OUT < ResultRbmData >)

Parameter (in) : No parameter

Parameter (out) : ResultRbmData

Parameter(inout) : ListRbmData

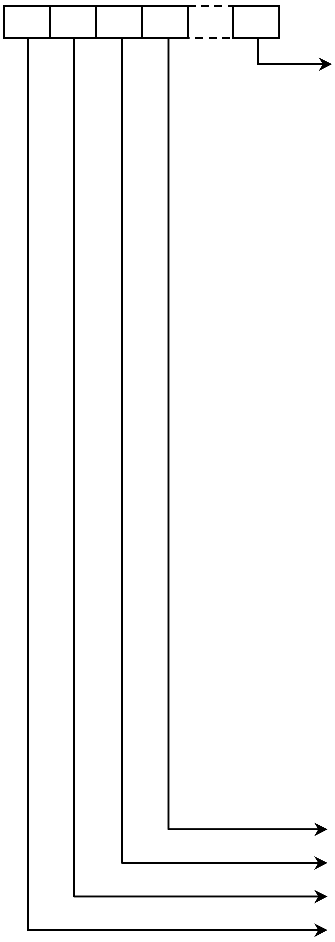
Short description : This API calculates and returns the Rate-Based Monitoring data to be transmitted to the Scan-Tool, when requested by the Mode 09$ (InfoType $08).

ResultRbmData : Indicates if the API has been executed or not.

Authorised values for ResultRbmDataSelection are :

- POSITIVE_RESPONSE : the call of the API is successful
- NEGATIVE_RESPONSE : the call of the API is unsuccessful

ListRbmData : Software structure filled-up with 16 counters of 2 bytes coming from Rate-Based Monitoring



EVAP Monitor Conditions Encountered Counts (denominator - LSB byte)
EVAP Monitor Conditions Encountered Counts (denominator - MSB byte)
EVAP Monitor Completion Condition Counts (numerator - LSB byte)
EVAP Monitor Completion Condition Counts (numerator - MSB byte)
AIR Monitor Conditions Encountered Counts (denominator - LSB byte)
AIR Monitor Conditions Encountered Counts (denominator - MSB byte)
AIR Monitor Completion Condition Counts (numerator - LSB byte)
AIR Monitor Completion Condition Counts (numerator - MSB byte)
EGR Monitor Conditions Encountered Counts (denominator - LSB byte)
EGR Monitor Conditions Encountered Counts (denominator – MSB byte)
EGR Monitor Completion Condition Counts (numerator - LSB byte)
EGR Monitor Completion Condition Counts (numerator - MSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 2 (denominator – LSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 2 (denominator - MSB byte)
O2 Sensor Monitor Completion Counts Bank 2 (numerator – LSB byte)
O2 Sensor Monitor Completion Counts Bank 2 (numerator - MSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 1 (denominator - LSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 1 (denominator - MSB byte)
O2 Sensor Monitor Completion Counts Bank 1 (numerator – LSB byte)
O2 Sensor Monitor Completion Counts Bank 1 (numerator – MSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 2 (denominator - LSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 2 (denominator - MSB byte)
Catalyst Monitor Completion Counts Bank 2 (numerator - LSB byte)
Catalyst Monitor Completion Counts Bank 2 (numerator - MSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 1 (denominator - LSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 1 (denominator - MSB byte)
Catalyst Monitor Completion Counts Bank 1 (numerator - LSB byte)
Catalyst Monitor Completion Counts Bank 1 (numerator - MSB byte)
Ignition Counter (LSB byte)
Ignition Counter (MSB byte)
OBD Monitoring Conditions Encountered Counts (general denominator - LSB byte)
OBD Monitoring Conditions Encountered Counts (general denominator - MSB byte)

Application conditions:

Initialization : Initialize the ListRbmData software structure with all counters set to 00h

ResultRbmData = NEGATIVE_RESPONSE

Recurrence : -

Activation : at Action call

Deactivation : -

Formula section:

```
If   NLC_OBD_RBM_ENA = 1
Then
     Fill-in ListRbmData software structure with Ignition counter = CTR_IGK_CYC_RBM and
     General denominator = CTR_CDN_OBD_RBM
     Call ACTION_ERRM_SelectRbmByGroup (IN<NC_RBM_CAT_1>,
     INOUT<ListOfRbmDataByGroup>, OUT<ResultRbmDataSelection>,
     SYNCHRONIZATION <CALL>)
     Fill-in ListOfRbmData with ListOfRbmDataByGroup
     Call ACTION_ERRM_SelectRbmByGroup (IN<NC_RBM_CAT_2>,
     INOUT<ListOfRbmDataByGroup>, OUT<ResultRbmDataSelection>,
     SYNCHRONIZATION <CALL>)
     Fill-in ListOfRbmData with ListOfRbmDataByGroup
     Call ACTION_ERRM_SelectRbmByGroup (IN< NC_RBM_LS_UP_1>,
     INOUT<ListOfRbmDataByGroup>, OUT<ResultRbmDataSelection>,
     SYNCHRONIZATION <CALL>)
     Fill-in ListOfRbmData with ListOfRbmDataByGroup
     Call ACTION_ERRM_SelectRbmByGroup (IN< NC_RBM_LS_UP_2>,
     INOUT<ListOfRbmDataByGroup>, OUT<ResultRbmDataSelection>,
     SYNCHRONIZATION <CALL>)
     Fill-in ListOfRbmData with ListOfRbmDataByGroup
     Call ACTION_ERRM_SelectRbmByGroup (IN< NC_RBM_EGR_VVT>,
     INOUT<ListOfRbmDataByGroup>, OUT<ResultRbmDataSelection>,
     SYNCHRONIZATION <CALL>)
     Fill-in ListOfRbmData with ListOfRbmDataByGroup
     Call ACTION_ERRM_SelectRbmByGroup (IN< NC_RBM_SA>,
     INOUT<ListOfRbmDataByGroup>, OUT<ResultRbmDataSelection>,
     SYNCHRONIZATION <CALL>)
     Fill-in ListOfRbmData with ListOfRbmDataByGroup
     Call ACTION_ERRM_SelectRbmByGroup (IN< NC_RBM_EVAP>,
     INOUT<ListOfRbmDataByGroup>, OUT<ResultRbmDataSelection>,
     SYNCHRONIZATION <CALL>)
     Fill-in ListRbmData with ListOfRbmDataByGroup
Else
     ListRbmData = 00000000000000000000000000000000h
Endif
```

ResultRbmData = POSITIVE_RESPONSE

27.27.8.2 API to select RBM statistic with lowest in-use performance ratio within one single group

General information:

This API shall be used to determine, within one single group (see specification “Rate-Based Monitoring (Appl. Inc.)”), the counters of the monitor which has the lowest in-use performance ratio.

Description:

Syntax : ACTION_ERRM_SelectRbmByGroup (
IN < GroupName >
INOUT < ListRbmDataByGroup >,
OUT < ResultRbmDataByGroup >)

Parameter (in) : GroupName

Parameter (out) : ResultRbmDataByGroup

Parameter(inout) : ListRbmDataByGroup

Short description : This API calculates and returns the Rate-Based Monitoring data with lowest ratios, within one single group.

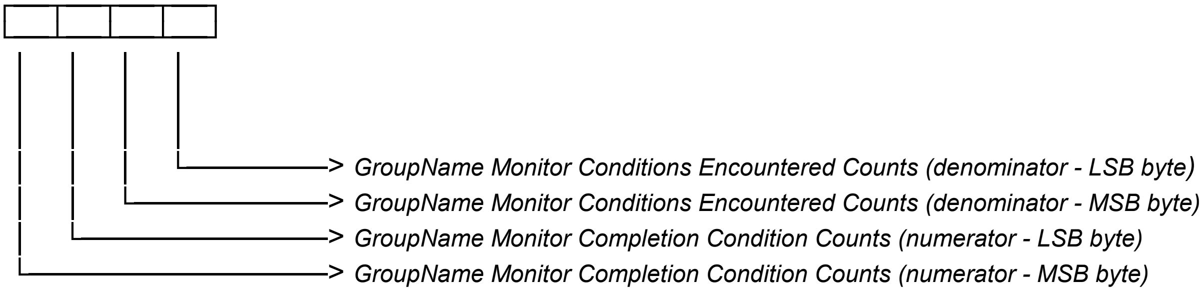
GroupName : Group name requested. See specification “Rate-Based Monitoring (Appl. Inc.)”

ResultRbmDataByGroup : Indicates if the API has been executed or not.

Authorised values for ResultRbmDataByGroup are :

- POSITIVE_RESPONSE : the call of the API is successful
- NEGATIVE_RESPONSE : the call of the API is unsuccessful

ListRbmDataByGroup : Software structure filled-up with the Rate-Based Monitoring counters of 2 bytes related to the requested group name.



Application conditions:

Initialization : -

Recurrence : -

Activation : at Action call

Deactivation : -

Formula section:

{ For CPU optimization reasons, this implementation assumes that Groups Names are ordered as indicated in the Rate-Based Monitoring (Appl. Inc.) file : Groups Names order must be respected and monitors belonging to a same group shall be joined }

{ If this requirement isn't respected, data sent to the Scan-Tool might be erroneous }

```
ResultRbmDataByGroup = NEGATIVE_RESPONSE
m = 0
While m ≤ ( NC_NR_DIAG_RBM – 1 ) do
    If(1) m monitor belongs to GroupName
    Then(1)

      If(2)  CTR_CDN_RBM[m] ≠ 0  { denominator is different from 0 }
      Then(2)
          RATIO_TMP_RBM = CTR_COMP_RBM[m] / CTR_CDN_RBM[m]
          Fill-in ListRbmDataByGroup with numerator = CTR_COMP_RBM[m] and denominator = CTR_CDN_RBM[m]
      Else(2)
          RATIO_TMP_RBM = FFFFh
          Fill-in ListRbmDataByGroup with numerator = CTR_COMP_RBM[m] and denominator = 0
      Endif(2)
      IDX_TMP_RBM = m

      For j = (m+1) to ( NC_NR_DIAG_RBM – 1 ) do
          If(3)   j monitor belongs to GroupName
          Then(3)
              If(4)   CTR_CDN_RBM[j] ≠ 0  { denominator is different from 0 }
              Then(4)
              { case of one monitor with lower ratio than previous one found }
```

```
                    If(5)   ( ( CTR_COMP_RBM[j] / CTR_CDN_RBM[j] ) <
                            RATIO_TMP_RBM )
                    Then(5)
                            RATIO_TMP_RBM = CTR_COMP_RBM[j] /
                            CTR_CDN_RBM[j]
                            IDX_TMP_RBM = j
                            Fill-in ListRbmDataByGroup with numerator =
                            CTR_COMP_RBM[j] and denominator =
                            CTR_CDN_RBM[j]
                    Else(5)
                            { case of two monitors with same ratios }
                            If(6)   ( ( CTR_COMP_RBM[j] / CTR_CDN_RBM[j] ) =
                                    RATIO_TMP_RBM )
                            Then(6)
                                    If(7)   CTR_CDN_RBM[j] >
                                            CTR_CDN_RBM[IDX_TMP_RBM]
                                    Then(7)
                                            Fill-in ListRbmDataByGroup with numerator
                                            = CTR_COMP_RBM[j] and denominator =
                                            CTR_CDN_RBM[j]
                                            IDX_TMP_RBM = j
                                    Endif(7)
                            Endif(6)
                    Endif(5)
                Else(4)
                    { denominator equals 0 }
                Endif(4)
            Endif(3)
        Endfor
        Exit while loop
      Else(1)
            { m monitor doesn't belong to GroupName }
            m = m +1
            Fill-in ListRbmDataByGroup with numerator = 0 and denominator = 0
      Endif(1)
  Endwhile
  ResultRbmDataByGroup = POSITIVE_RESPONSE
```

27.27.8.3 API to clear all Rate-Based Monitoring statistics

General information:

Regulation text :
"Each number shall be reset to zero only when a non-volatile memory reset occurs (e.g., reprogramming event, etc.) or, if the numbers are stored in keep-alive memory (KAM), when KAM is lost due to an interruption in electrical power to the control module (e.g., battery disconnect, etc.). Numbers may not be reset to zero under any other circumstances including when a scan tool command to clear fault codes or reset KAM is received."

The service provided in this paragraph shall be used to clear all the Rate-Based Monitoring statistics after a reprogramming session of ECU software and/or calibration data.

This service shall not be called when a Scan Tool command to clear faults (Mode 04$) is received.

Description:

Syntax :	ACTION_ERRM_ClearRbmStatistics (OUT < ResultClrInfo>)
Parameter(in) :	No parameter.
Parameter(out) :	ResultClrInfo to inform if RBM statistics are cleared or not
Short Description :	This API clear all the Rate-Based Monitoring statistics

Remark : For design and CPU resource reasons, this API shall be called while engine is stopped.

Application conditions:

Initialization :	-
Recurrence :	-
Activation :	at Action call
Deactivation :	-

Formula section:

```
ResultClrInfo = STATISTICS_NOT_CLEARED
CTR_IGK_CYC_RBM = 0
CTR_CDN_OBD_RBM = 0
For m = 0 to ( NC_NR_DIAG_RBM –1 ) do
    CTR_CDN_RBM[m] = 0
    CTR_COMP_RBM[m] = 0
Endfor
ResultClrInfo = STATISTICS_CLEARED
```

Configuration data:

Name	Dim	Hex. Limits	Phys. Limits	Resol.	Unit
NC_ERR_DTC_REQ_OBD	-	0...1H	0...1	1	[-]
This bit shall be set to 0 (SYMPTOM) to read OBD DTC by symptomThis bit shall be set to 1 (FAILURE) to read OBD DTC by failure					
NC_ERR_DTC_REQ_CUS	-	0...1H	0...1	1	[-]
This bit shall be set to 0 (SYMPTOM) to read customer DTC by symptomThis bit shall be set to 1 (FAILURE) to read customer DTC by failure					
NLC_OBD_FRF_PND	-	0...1H	0...1	1	[-]
Defines the EOBD/CARB freeze frame strategy the following ways :0 : EOBD/CARB freeze frame data of Mode 02h will be returned after detection of a confirmed failure .1 : EOBD/CARB freeze frame data of Mode 02h will be returned :after detection of a confir					

Configuration data detailed description:

NC_ERR_DTC_REQ_OBD : This bit shall be set to 0 (SYMPTOM) to read OBD DTC by symptom.
This bit shall be set to 1 (FAILURE) to read OBD DTC by failure.

NC_ERR_DTC_REQ_CUS : This bit shall be set to 0 (SYMPTOM) to read Customer DTC by symptom.
This bit shall be set to 1 (FAILURE) to read Customer DTC by failure.

NLC_OBD_FRF_PND : Definition the EOBD/CARB freeze frame strategy, as follow :
0 : EOBD/CARB freeze frame data of Mode 02h will be returned after detection of a confirmed failure.
1 : EOBD/CARB freeze frame data of Mode 02h will be returned :

- after detection of a confirmed emission relevant failure if confirmed in memory
- after pending emission relevant detection if not confirmed in memory

27.28 Communication Interface (Appl. Inc.)

27.28.1 Communication interface for BMW diagnosis tool

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_TYPE_BYTE[NC_NR_ERR_DYN]	O	0...FFH	0...255	1	[-]
Fault-type-byte					
ERR_TYPE_EXT_BYTE[NC_NR_ERR_DYN]	O	0...FFH	0...255	1	[-]
Fault-type-extension-byte					

Input data:

LV_END_DIAG_XX	LV_DC_MAX[NC_NR_ERR_DYN]	LV_ERR_XX	WAL_CONF_XX
LV_CDN_DIAG_XX	CTR_WUP_CYC[NC_NR_ERR_DYN]	ENVD_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN][NC_NR_FRF_SET][NC_NR_ERR_DYN]	LV_ERR_DC[NC_NR_ERR_DYN]
LV_ERR_CFM[NC_NR_ERR_DYN]	LV_ERR_TMP[NC_NR_ERR_DYN]	LV_READY_XX	LV_ERR_MKD[NC_NR_ERR_DYN]
CTR_DC[NC_NR_ERR_DYN]			

FUNCTION DESCRIPTION:

General information:

This communication interface is to define the data contents needed for the BMW diagnosis information. The Information in the Fault-type-byte and Fault-type-extension-byte is bit-coded and requested via KWP communication.

Application conditions:

Initialisation: managed in the specific chapters

Recurrence: managed in the specific chapters

Activation: see chapter “KWP communication”

Formula section:

Definition of ERR_TYPE_BYTE[x]

Bit No:	Description	Input
0	Specifies the error symptom: 0001-Short circuit to VBATT or signal / value above threshold (Max)	**IF** ENVD_CUS_SET_CMN[1][x] = 1H
1	Specifies the error symptom: 0010-Short circuit to ground or signal / value below threshold (Min)	**IF** ENVD_CUS_SET_CMN[1][x] = 2H
2	Specifies the error symptom: 0100-Open circuit or no signal (Sig)	**IF** ENVD_CUS_SET_CMN[1][x] = 4H
3	Specifies the error symptom: 1000-Implausible signal / condition (Plaus)	**IF** ENVD_CUS_SET_CMN[1][x] = 8H
4	BMW Readiness flag. The bit is set to 0 if the test conditions are met and the diagnostic runs to completion at least once after the fault memory has been cleared, or after power failure	**IF** LV_READY_XX = 0
5	This bit is set when the fault has been saved to the fault memory as OBD debounced.	**IF** LV_DC_MAX[x] = 1
6	This bit is set when the fault is currently present	**IF** LV_ERR_XX = 1
7	MIL calibration status, specifies wheather the MIL is to be activated	**IF** MIL_ON bit of WAL_CONF_XX =1

Definition of ERR_TYPE_EXT_BYTE[x]

Bit No:	Description	Input
0	Only for Diagnostic diagnosis; This bit is set if the diagnosis is currently running	**For diagnostic diagnosis**** LV_END_DIAG_XX = 0 **AND** LV_CDN_DIAG_XX = 1
1	Only for Diagnostic diagnosis; This bit is set if the diagnosis is interrupted for this driving cycle	**For diagnostic diagnosis**** Not defined yet
2	Cycle flag. The bit is set if the diagnosis runs to completion in this driving cycle	**IF** LV_END_DIAG_XX = 1
3	This bit is to be set if the fault is stored as a result of an action by an tester	**IF** LV_ERR_MKD[x] = 1
4	This bit is to be set if the fault is currently requesting the MIL to be illuminated.	**IF** LV_DC_MAX[x] = 1 **AND** MIL_ON bit of WAL_CONF_XX =1
5	This bit is set if MIL debouncing has been achieved, the MIL is not currently being requested for this fault and output via the Scan Tool interface has not yet occured (HLC =0 & DLC >0). (former OBDII error)	**IF** LV_ERR_CFM[x] = 1 **AND** CTR_DC[x] = 0 **AND** MIL_ON bit of WAL_CONF_XX =1 **AND** CTR_WUP_CYC[x] > 0
6	Not yet defined	
7	Not yet defined	

**Diagnostic diagnosis are all sequential diagnosis (see “Table of failures – CARB status”)

27.29 Diagnosis Trouble Code Management (DTC)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_MEM[NC_NR_ERR_DYN]	V/O/S	0...FH	0...15	1	[-]
Memorized detected symptom of the failure IDX (for the DTC building)					
ERR_SYM_LST[NC_NR_ERR_DYN]	V/O/S	0...FH	0...15	1	[-]
Memorized the last detected symptom of the failure IDX (for the DTC building)					
#IF NLC_TREAT_DIAG_MIS = 0					
SYM_CYL_MEM_MIS_A	V/O/S	0...FFFFH	0...65535	1	[-]
memorized cylinder with misfire A (for the DTC building)					
SYM_CYL_MEM_MIS_B1	V/O/S	0...FFFFH	0...65535	1	[-]
memorized cylinder with misfire B1 (for the DTC building)					
SYM_CYL_MEM_MIS_B4	V/O/S	0...FFFFH	0...65535	1	[-]
memorized cylinder with misfire B4 (for the DTC building)					
SYM_CYL_LST_MIS_A	V/O/S	0...FFFFH	0...65535	1	[-]
Memorized the last cylinder of the misfire A failure (for the DTC building)					
SYM_CYL_LST_MIS_B1	V/O/S	0...FFFFH	0...65535	1	[-]
Memorized the last cylinder of the misfire B1 failure (for the DTC building)					
SYM_CYL_LST_MIS_B4	V/O/S	0...FFFFH	0...65535	1	[-]
Memorized the last cylinder of the misfire B4 failure (for the DTC building)					
#ENDIF					
ERR_DTC[NC_NR_ERR_DYN]	V/O/S	0...FFFFH	0...65535	1	[-]
DTC code stored in memory for failure IDX					

Input data:

ERR_SYM_XX	SYM_CYL_MIS_A	SYM_CYL_MIS_B1	SYM_CYL_MIS_B4
NC_CYL_NR	NLC_TREAT_DIAG_MIS		

Export actions :

ACTION_ERRM_StoreDtc (IN <IDX>)
This action stores the memorized symptom of the first occurrence of the failure IDX to build the DTC
ACTION_ERRM_StoreDtcLst (IN <IDX>)
This action stores the memorized symptom of the last occurrence of the failure IDX to build the DTC
ACTION_ERRM_EraseDtc (IN <IDX>)
This action erases the memorized symptom of the failure IDX

FUNCTION DESCRIPTION:

General information:

Purpose of this module is to catch some data (ERR_SYM_XX) when a failure occurs. Then, these data (ERR_SYM_MEM[IDX] and ERR_SYM_LST_[IDX]) are used to generate a code called DTC upon communication tool request (see Communication interface module). Additionally, for visualization (for communication tool, developpment purposes), a DTC code is stored in memory (ERR_DTC_[IDX]).

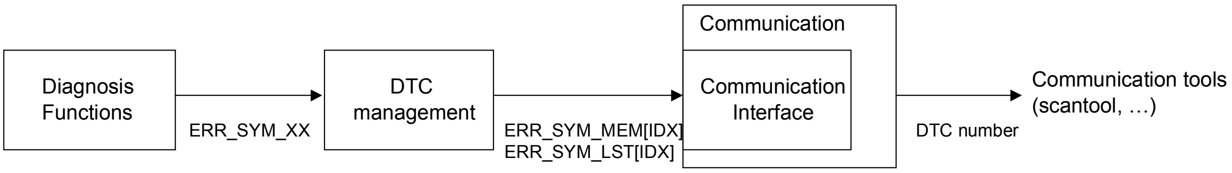
The DTC is a code used to identify the symptom of the failure or the failure (global failure). It is an interface between the diagnosis and the communication.

The customer and/or EOBD/CARB according to the standard J2012 define this DTC.

DTC encoding for regulation EOBD/CARB shall be done using J2012 definition (2 bytes interpreted [0000h to FFFFh] in hex, [P0000 to U3FFF] in J2012).

DTC encoding for customer is free. It can be in decimal, or applying J2012 definition, or any other encoding definition.

Signal flow diagram:



27.29.1 DTC Storage

27.29.1.1 DTC Storage / First occurence

Description:

When the failure IDX gets present (1st occurrence), the detected of the first symptom is stored in the 2nd layer memory by calling the action ACTION_ERRM_StoreDtc (IN<IDX>). Additionally, a DTC code is stored in dynamic memory for visualization :

- related to the symptom DTC if NC_ERR_DTC_CONF = 0
- related to the failure OBD DTC if NC_ERR_DTC_CONF = 1
- related to the customer specific symptom DTC if NC_ERR_DTC_CONF = 2
- related to the customer specific failure DTC if NC_ERR_DTC_CONF = 3

Syntax :	ACTION_ERRM_StoreDtc (IN <IDX>)	
Parameter (in) :	IDX	Index of failure in 2nd layer memory to store the DTC
Parameter (out) :	-	
Short description :	This action stores the memorised symptom of the first occurrence of the failure IDX to build the DTC	

Application conditions:

Initialization: ERR_SYM_MEM[IDX] is restored from the NVMY

SYM_CYL_MEM_MIS_A, SYM_CYL_MEM_MIS_B1 and

SYM_CYL_MEM_MIS_B4 are restored from NVMY

In case of damaged NVMY, ERR_SYM_MEM[IDX] = 0

SYM_CYL_MEM_MIS_A = 0,

SYM_CYL_MEM_MIS_B1 = 0,

SYM_CYL_MEM_MIS_B4 = 0,

ERR_DTC[IDX] = 0

Recurrence: -

Activation: at Action request

Formula section:

```
If      NLC_TREAT_DIAG_MIS = 0
        (Misfire are treated by type (LV_ERR_MIS_A, MIS_B1 and MIS_B4 is defined))
Then
        For a misfiring diagnosis:
        (XX stands for MIS_A, MIS_B1, and MIS_B4) (Misfire by type NLC_TREAT_DIAG_MIS=0)
                SYM_CYL_MEM_XX = SYM_CYL_XX

                If      NC_ERR_DTC_CONF = 0 or 1 or 4                   (J2012 DTC is stored)
                Then    If      SYM_CYL_MEM_XX [12] = 1         (Random)
                        Then    ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 0 ] [ NC_CYL_NR+1 ]
                        Else    If      SYM_CYL_MEM_XX [13] = 1         (Multiple cylinder)
                                        Or many cylinder bits are set in SYM_CYL_MEM_XX
                                Then    ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 0 ] [ NC_CYL_NR ]
                                Else    (only one cylinder bit is set in SYM_CYL_MEM_XX)
                                                i = bit number which is set in SYM_CYL_MEM_XX
                                                ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 0 ] [ i ]
                                Endif
                        Endif
                Endif

                If      NC_ERR_DTC_CONF = 2 or 3 or 5 (customer encoded DTC is stored)
                Then    If      SYM_CYL_MEM_XX [12] = 1         (Random)
                        Then    ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 1 ] [ NC_CYL_NR+1 ]
                        Else    If      SYM_CYL_MEM_XX [13] = 1         (Multiple cylinder)
                                        Or many cylinder bits are set in SYM_CYL_MEM_XX
                                Then    ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 1 ] [ NC_CYL_NR ]
                                Else    (only one cylinder bit is set in SYM_CYL_MEM_XX)
                                                i = bit number which is set in SYM_CYL_MEM_XX
                                                ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 1 ] [ i ]
                                Endif
                        Endif
                Endif
Endif
```

For others diagnosis :

(also for misfire diagnosis if treated by cylinder (LV_ERR_MIS_x is defined))

```
ERR_SYM_MEM[IDX] = ERR_SYM_XX
If      NC_ERR_DTC_CONF = 0 or 4        (1st symptom J2012 DTC is stored)
Then    If      only one bit is set in ERR_SYM_MEM[IDX]
        Then    i = bit number which is set in ERR_SYM_MEM[IDX]
                ERR_DTC[IDX] =  ID_ERR_DTC_XX [0] [ i ] with XX = F(IDX)
        Else    (many bit is set in ERR_SYM_MEM[IDX])
                ERR_DTC[IDX] =  ID_ERR_DTC_XX [0] [4] with XX = F(IDX)
        Endif
Endif

If      NC_ERR_DTC_CONF = 1             (global J2012 DTC is stored)
Then    ERR_DTC[IDX] =  ID_ERR_DTC_XX [0] [4] with XX = F(IDX)
Endif

If      NC_ERR_DTC_CONF = 2 or 5        (customer encoded DTC is stored)
Then    If      only one bit is set in ERR_SYM_MEM[IDX]
        Then    i = bit number which is set in ERR_SYM_MEM[IDX]
                ERR_DTC[IDX] =
                ID_ERR_DTC_XX [NC_NR_DTC_FMT] [ i ] with XX = F(IDX)
        Else    (many bit is set in ERR_SYM_MEM[IDX])
                ERR_DTC[IDX] =
                ID_ERR_DTC_XX [NC_NR_DTC_FMT] [5 - NC_NR_DTC_FMT]
                with XX = F(IDX)
        Endif
Endif

If      NC_ERR_DTC_CONF = 3    (global customer encoded DTC is stored)
Then    ERR_DTC[IDX] =
        ID_ERR_DTC_XX [NC_NR_DTC_FMT] [5 - NC_NR_DTC_FMT]
        with XX = F(IDX)
Endif
```

27.29.1.2 DTC Storage / Last occurence

Description:

For each occurence of the failure IDX (Last occurrence), the detected of the last symptom is stored in the 2^{nd} layer memory by calling the action ACTION_ERRM_StoreDtcLst (IN<IDX>).

Syntax : ACTION_ERRM_StoreDtcLst (IN<IDX>)

Parameter (in) : IDX Index of failure in 2^{nd} layer memory to store the DTC

Parameter (out) : -

Short description : This action stores the memorised symptom of the last occurrence of the failure IDX to build the DTC

Application conditions:

Initialization: ERR_SYM_LST[IDX] is restored from the NVMY

SYM_CYL_LST_MIS_A, SYM_CYL_LST_MIS_B1 and

SYM_CYL_LST_MIS_B4 are restored from NVMY

In case of damaged NVMY, ERR_SYM_LST[IDX] = 0

SYM_CYL_LST_MIS_A = 0,

SYM_CYL_LST_MIS_B1 = 0,

SYM_CYL_LST_MIS_B4 = 0,

Recurrence: -

Activation: at Action request

Formula section:

```
ERR_SYM_LST[IDX] = ERR_SYM_XX
If    NLC_TREAT_MIS_DIAG = 0
      (Misfire are treated by type)
then        For a misfiring diagnosis :
      (XX stands for MIS_A, MIS_B1 and MIS_B4)
      SYM_CYL_LST_XX = SYM_CYL_ XX
endif
```

```
If    NLC_TREAT_DIAG_MIS = 0
      (Misfire are treated by type (LV_ERR_MIS_A, MIS_B1 and MIS_B4 is defined))
Then        For a misfiring diagnosis:

            If    NC_ERR_DTC_CONF = 4                          (J2012 DTC is stored)
            Then  If    SYM_CYL_LST_XX [12] = 1 (Random)
                  Then  ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 0 ] [ NC_CYL_NR+1 ]
                  Else  If    SYM_CYL_LST_XX [13] = 1 (Multiple cylinder)
                              Or many cylinder bits are set in SYM_CYL_LST_XX
                        Then  ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 0 ] [ NC_CYL_NR ]
                        Else  (only one cylinder bit is set in SYM_CYL_LST_XX)
                                    i = bit number which is set in SYM_CYL_LST_XX
                                    ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 0 ] [ i ]
                        Endif
                  Endif
            Endif

            If    NC_ERR_DTC_CONF = 5                    (customer DTC is stored)
            Then  If    SYM_CYL_LST_XX [12] = 1 (Random)
                  Then  ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 1 ] [ NC_CYL_NR+1 ]
                  Else  If    SYM_CYL_LST_XX [13] = 1 (Multiple cylinder)
                              Or many cylinder bits are set in SYM_CYL_LST_XX
                        Then  ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 1 ] [ NC_CYL_NR ]
                        Else  (only one cylinder bit is set in SYM_CYL_LST_XX)
                                    i = bit number which is set in SYM_CYL_LST_XX
                                    ERR_DTC[IDX] = ID_ERR_DTC_MIS [ 1 ] [ i ]
                        Endif
                  Endif
            Endif
```

For others diagnosis :

(also for misfire diagnosis if treated by cylinder (LV_ERR_MIS_x is defined))

```
If      NC_ERR_DTC_CONF = 4             (last symptom J2012 DTC is stored)
Then    If      only one bit is set in ERR_SYM_LST[IDX]
        Then    i = bit number which is set in ERR_SYM_LST[IDX]
                ERR_DTC[IDX] =  ID_ERR_DTC_XX [0] [ i ] with XX = F(IDX)
        Else    (many bit is set in ERR_SYM_LST[IDX])
                ERR_DTC[IDX] =  ID_ERR_DTC_XX [0] [4] with XX = F(IDX)
        Endif
Endif


If      NC_ERR_DTC_CONF = 5             (customer encoded DTC is stored)
Then    If      only one bit is set in ERR_SYM_LST[IDX]
        Then    i = bit number which is set in ERR_SYM_LST[IDX]
                ERR_DTC[IDX] =
                ID_ERR_DTC_XX [NC_NR_DTC_FMT] [ i ] with XX = F(IDX)
        Else    (many bit is set in ERR_SYM_LST[IDX])
                ERR_DTC[IDX] =
                ID_ERR_DTC_XX [NC_NR_DTC_FMT] [5 - NC_NR_DTC_FMT]
                with XX = F(IDX)
        Endif
Endif
```

27.29.2 DTC storage for the Misfiring (with NLC_TREAT_DIAG_MIS = 0, Misfire by type)

Description:

As describe above, the cylinder in default is memorised at the first occurrence. In this part each new cylinder in default will be memorised.

Application conditions:

Initialization:	treated above
Recurrence:	For Misfire A, B1 and B4: Misfire segment synchronous
Activation:	-

Formula section:

For a misfiring A diagnosis :

SYM_CYL_MEM_MIS_A is updated with the new additional bits (cylinder, multiple and random) of SYM_CYL_MIS_A

For a misfiring B1 diagnosis :

SYM_CYL_MEM_MIS_B1 is updated with the new additional bits (cylinder, multiple and random) of SYM_CYL_MIS_B1

For a misfiring B4 diagnosis :

SYM_CYL_MEM_MIS_B4 is updated with the new additional bits (cylinder, multiple and random) of SYM_CYL_MIS_B4.

27.29.3 DTC deleting

Description:

When the failure IDX is erased from 2nd layer memory, the memorised detected symptom is also erased by calling the action ACTION_ERRM_EraseDtc (IN <IDX>).

Syntax :	ACTION_ERRM_EraseDtc (IN <IDX>)	
Parameter (in) :	IDX	Index of failure in 2nd layer memory to store the DTC
Parameter (out) :	-	
Short description :	This action erases the memorised symptom of the failure IDX	

Application conditions:

Initialization:	-
Recurrence:	-
Activation:	at Action request

Formula section:

```
If    NLC_TREAT_DIAG_MIS = 0
      (Misfire are treated by type (LV_ERR_MIS_A, MIS_B1 and MIS_B4 is defined))
Then        For a misfiring A diagnosis :
            SYM_CYL_MEM_MIS_A = 0
            SYM_CYL_LST_MIS_A = 0
            ERR_DTC[IDX] = 0
            For a misfiring B1 diagnosis :
            SYM_CYL_MEM_MIS_B1 = 0
            SYM_CYL_LST_MIS_B1 = 0
            ERR_DTC[IDX] = 0
            For a misfiring B4 diagnosis :
            SYM_CYL_MEM_MIS_B4 = 0
            SYM_CYL_LST_MIS_B4 = 0
            ERR_DTC[IDX] = 0
Endif
```

For other diagnoses, including misfire diagnoses if treated by cylinder (LV_ERR_MIS_x is defined) :

```
            ERR_SYM_MEM[IDX] = 0
            ERR_SYM_LST[IDX] = 0
            ERR_DTC[IDX] = 0
```

27.29.4 Calibratable DTC number

Description:

The DTC numbers are defined by a calibration table called ID_ERR_DTC_XX.

This table defined for each failure XX :

a) with NC_NR_DTC_FMT = 0

- a DTC number for each symptom of the failure XX
- a global OBD DTC number for the failure XX
- a global customer specific DTC number

b) with NC_NR_DTC_FMT = 1

- a OBD DTC number for each symptom of the failure XX
- a global OBD DTC number for the failure XX
- a customer specific DTC number for each symptom of the failure XX
- a global customer specific DTC number

When a DTC is request by a diagnostic tool, the Communication Interface module build up the DTC number with information :

- ERR_SYM_MEM[IDX] and / or ERR_SYM_LST[IDX] stored in 2^{nd} layer memory ;
- Contents of table ID_ERR_DTC_XX .

There is a particular case for misfiring (MIS_A, MIS_B1, MIS_B4) diagnosis. The DTC numbers are defined by cylinder in a calibration table called ID_ERR_DTC_MIS.

(used if NLC_TREAT_DIAG_MIS = 0; Misfire by type)

- ID_ERR_DTC_MIS[0] [i] is the DTC for misfiring detection on the cylinder number i +1
- ID_ERR_DTC_MIS[0] [NC_CYL_NR] is the DTC for multiple misfiring
- ID_ERR_DTC_MIS[0] [NC_CYL_NR +1] is the DTC for random misfiring

- ID_ERR_DTC_MIS[1] [i] is the customer DTC for misfiring detection on the cylinder number i+1
- ID_ERR_DTC_MIS[1] [NC_CYL_NR] is the customer DTC for multiple misfiring
- ID_ERR_DTC_MIS[1] [NC_CYL_NR +1] is the customer DTC for random misfiring

Example 1:

The failure TCO (coolant temperature) is stored in the 2nd layer memory. The calibratable table of DTC has the following value :

a) with NC_NR_DTC_FMT = 0

DTC of the 1st symptom	DTC of the 2nd symptom	DTC of the 3rd symptom	DTC of the 4th symptom	Global DTC of the failure	Specific customer code
ID_ERR_DTC_ TCO [0] [0] = 0118h (P0118)	ID_ERR_DTC_ TCO [0] [1] = 0117h (P0117)	ID_ERR_DTC_ TCO [0] [2] = 0119h (P0119)	ID_ERR_DTC_ TCO [0] [3] = 0116h (P0116)	ID_ERR_DTC_ TCO [0] [4] = 0115h (P0115)	ID_ERR_DTC_ TCO [0] [5] = 30

b) with NC_NR_DTC_FMT = 1

	DTC of the 1st symptom	DTC of the 2nd symptom	DTC of the 3rd symptom	DTC of the 4th symptom	Global DTC of the failure
OBD DTC	ID_ERR_DTC_ TCO [0] [0] = 0118h (P0118)	ID_ERR_DTC_ TCO [0] [1] = 0117h (P0117)	ID_ERR_DTC_ TCO [0] [2] = 0119h (P0119)	ID_ERR_DTC_ TCO [0] [3] = 0116h (P0116)	ID_ERR_DTC_ TCO [0] [4] = 0115h (P0115)
Customer DTC	ID_ERR_DTC_ TCO [1] [0] = 31	ID_ERR_DTC_ TCO [1] [1] = 32	ID_ERR_DTC_ TCO [1] [2] = 33	ID_ERR_DTC_ TCO [1] [3] = 34	ID_ERR_DTC_ TCO [1] [4] = 30

Upon request from the diagnosis tool, the DTC number sent to the tool will be one of the numbers defined in ID_ERR_DTC_TCO regards the symptom detected.

Example 2 :

One failure misfiring (LV_ERR_MIS_A, LV_ERR_MIS_B1 or LV_ERR_MIS_B4) is stored in the 2nd layer memory. The application is a NC_CYL_NR -cylinder engine. The calibratable table of DTC has the following value :

	Misfiring on 1st cylinder	Misfiring on 2nd cylinder	Misfiring on 3rd cylinder	Misfiring on 4th cylinder	...	Misfiring on NC_CYL_NR th cylinder	Multiple Misfiring	Random Misfiring
OBD DTC	ID_ERR_ DTC_MIS [0] [0] = 0301h (P0301)	ID_ERR_ DTC_MIS [0] [1] = 0302h (P0302)	ID_ERR_ DTC_MIS [0] [2] = 0303h (P0303)	ID_ERR_ DTC_MIS [0] [3] = 0304h (P0304)	...	ID_ERR_ DTC_MIS [0] [NC_CYL_NR - 1] = 03xxh (P03xx)	ID_ERR_ DTC_MIS [0] [NC_CYL_NR] = 0300h (P0300)	ID_ERR_ DTC_MIS [0] [NC_CYL_NR + 1] = 0300h (P0300)
Customer DTC	ID_ERR_ DTC_MIS [1] [0] = 11	ID_ERR_ DTC_MIS [1] [1] = 12	ID_ERR_ DTC_MIS [1] [2] = 13	ID_ERR_ DTC_MIS [1] [3] = 14	...	ID_ERR_ DTC_MIS [1] [NC_CYL_NR - 1] = 16	ID_ERR_ DTC_MIS [1] [NC_CYL_NR] = 10	ID_ERR_ DTC_MIS [1] [NC_CYL_NR + 1] = 10

Upon request from the diagnosis tool, the DTC number sent to the tool will be one of the numbers defined in ID_ERR_DTC_MIS regards the symptom detected.

Calibration data:

Name	Dim	Hex. Limits	Phys. Limits	Resol.	Unit
ID_ERR_DTC_XX	NC_LDP_1_DTC_TABLE_SIZE * NC_LDP_2_DTC_TABLE_SIZE	0...FFFFH	0...65535	1	[-]
LDP_NR_1_ID_ERR_DTC_XX	NC_LDP_1_DTC_TABLE_SIZE	0...FFH	0...255	1	[-]
LDP_NR_2_ID_ERR_DTC_XX	NC_LDP_2_DTC_TABLE_SIZE	0...FFH	0...255	1	[-]
DTC table declaration for the diagnosis instance XX, J2012 shall be applied on OBD/CARB part					
ID_ERR_DTC_MIS	2* NC_LDP_2_DTC_MIS_TABLE_SIZE	0...FFFFH	0...65535	1	[-]
LDP_NR_1_ID_ERR_DTC_MIS	2	0...FFH	0...255	1	[-]
LDP_NR_2_ID_ERR_DTC_MIS	NC_LDP_2_DTC_MIS_TABLE_SIZE	0...FFH	0...255	1	[-]
DTC table declaration for misfiring diagnosis, J2012 shall be applied on OBD/CARB part					

With :

NC_LDP_1_DTC_TABLE_SIZE = NC_NR_DTC_FMT +1

NC_LDP_2_DTC_TABLE_SIZE = 6 - NC_NR_DTC_FMT

NC_LDP_2_DTC_MIS_TABLE_SIZE = NC_CYL_NR +2

Configuration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
NC_ERR_DTC_CONF	1	0...5H	0...5	1	[-]
Type of DTCs displayed through ERR_DTC[IDX]					
NC_NR_DTC_FMT	1	0...1H	0...1	1	[-]
Selection of 6 or 10 DTCs configurations					
NC_LDP_1_DTC_TABLE_SIZE	1	0...FH	0...15	1	[-]
Table size definition for ID_ERR_DTC_XX; set to NC_NR_DTC_FMT+1					
NC_LDP_2_DTC_TABLE_SIZE	1	0...FH	0...15	1	[-]
Table size definition for ID_ERR_DTC_XX; set to 6-NC_NR_DTC_FMT					
NC_LDP_2_DTC_MIS_TABLE_SIZE	1	0...FH	0...15	1	[-]
Table size definition for ID_ERR_DTC_MIS; set to NC_CYL_NR+2					

Configuration data detailed description:

NC_NR_DTC_FMT	= 0	6 DTCs configuration.
	= 1	10 DTCs configuration.

27.30 DTC Management (Appl. Inc.)

27.30.1 Definition of additional DTC information

FUNCTION DESCRIPTION:

General information:

For several diagnosis services it is necessary to transmit an additional information for each symptom.

Description:

These additional information can be calibrated for each symptom of a failure XX with the table ID_ERR_DTC_INFO_XX.

For this project it's used to define the DTCFailureTypeByte according to ISO 14229.

Example for diagnostic instance V_TCO_RNG:

additional Info for SYM_0	additional Info for SYM_1	additional Info for SYM_2	additional Info for SYM_3
ID_ERR_DTC_ INFO_V_TCO_RNG[0] = 05h	ID_ERR_DTC_ INFO_V_TCO_RNG[1] = 01h	ID_ERR_DTC_ INFO_V_TCO_RNG[2] = 00h	ID_ERR_DTC_ INFO_V_TCO_RNG[3] = 00h

Formula section:

not used yet

27.31 History memory

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_ERR_HIS_NR	V/O/S	0...FFH	0...255	1	[-]
Number of failures stored in the history memory					
ERR_HIS_DTC[NC_NR_ERR_HIS]	V/O/S	0...FFFFH	0...65535	1	[-]
DTC of failure in history memory					
ERR_HIS[NC_NR_HIS][NC_NR_ERR_HIS]	V/O/S	0...FFH	0...255	1	[-]
Others data in history memory					

Output data detailed description:

ERR_HIS_DTC[NC_NR_ERR_HIS] array counts NC_NR_ERR_HIS elements (1… NC_NR_ERR_HIS).

ERR_HIS[NC_NR_HIS][NC_NR_ERR_HIS] array counts NC_NR_HIS x NC_NR_ERR_HIS elements (1… NC_NR_HIS, 1…NC_NR_ERR_HIS).

Input data:

ERR_HIS_CONF[NC_NR_HIS]	ERR_DTC[NC_NR_ERR_DYN]		

Export actions:

ACTION_ERRM_StoreHistory (IN <IDX>)
This action stores the failure IDX in the history memory

FUNCTION DESCRIPTION:

General information:

This module is called by the dynamic error management module just before erasing a failure in the 2nd layer memory (dynamic memory).

The principle is to copy some data related to this failure in a separate memory, called history memory. So the failure become historic failure.

The number of historic failure is limited to a maximum of NC_NR_ERR_HIS. When this maximum number is reach, new historic failure entry deletes the oldest historic failure in memory (“first in first out” principle).

History memory is defined by :

- ERR_HIS_DTC[y] with : y = index of failure in history memory

  A DTC number related to the failure.

- ERR_HIS[x][y] with : x = index of data stored for failure y in history memory

  y = index of failure in history memory

  Some others data.

Contents of history memory is the same for each failure and defined in History memory (Appl. Inc.) module.

The history memory can be read and erased separately by dedicated diagnosis tool service.

The history memory can be deleted by setting the calibration LC_ERR_FMY_HIS_CLR to 1.

27.31.1 Store a failure in history memory

Description:

Syntax : ACTION_ERRM_StoreHistory (IN <IDX>)

Parameter (in) : IDX Index of failure to store in history memory

Parameter(out) : -

Short description : This action stores a failure in history memory

Application conditions:

Initialization: after system initialization the history memory is restored from non-volatile memory or in case of a damaged non volatile memory the history memory is initialized with zero

Recurrence: -

Activation: at action request

Formula section:

```
If     CTR_ERR_HIS_NR < NC_NR_ERR_HIS
Then   CTR_ERR_HIS_NR = CTR_ERR_HIS_NR + 1
Else   ERR_HIS_DTC[y] = ERR_HIS_DTC[y+1]    (“first in first out” principle)
       ERR_HIS[x][y] = ERR_HIS[x][y+1]      (with x = 1..NC_NR_HIS, y = 1...CTR_ERR_HIS_NR–1)
Endif
ERR_HIS_DTC[y] = ERR_DTC[IDX]
ERR_HIS[x][y] = ERR_HIS_CONF[x]
( with x = 1.. NC_NR_HIS, y = CTR_ERR_HIS_NR )
```

27.31.2 Clear the History Memory

Description:

The history memory can be erased by setting the calibration bit LC_ERR_FMY_HIS_CLR at 1.

Application conditions:

Initialization:	after system initialization the history memory is restored from non-volatile memory or in case of a damaged non volatile memory the history memory is initialized with zero
Recurrence:	-
Activation:	at LC_ERR_FMY_HIS_CLR 0→1 transition

Formula section:

ERR_HIS_DTC[y] = 0
ERR_HIS[x][y] = 0
(with x = 1 ... NC_NR_HIS
y = 1 ... NC_NR_ERR_HIS)
CTR_ERR_HIS_NR = 0

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_ERR_FMY_HIS_CLR	1	0...1H	0...1	1	[-]
Calibration to clear the history memory (when set to 1)					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_ERR_HIS	1	0...FFH	0...255	1	[-]
Max number of failure in the historic memory					
NC_NR_HIS	1	0...FFH	0...255	1	[-]
Max number of data (in bytes) to stored for a failure in the historic memory					

27.32 History memory (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_HIS_CONF[NC_NR_HIS]	O	0...FFH	0...255	1	[-]
Contents of history memory					

Input data:

NC_NR_ERR_HIS	NC_NR_HIS	ERR_TYPE_BYTE_IDX	CTR_FRC_IDX
ENVD_CUS_SET_CMN[NC_NR_ENVD_CUS_SET_CMN][NC_NR_FRF_SET][NC_NR_ERR_DYN]	C_ERR_CLAS_XX		

FUNCTION DESCRIPTION:

General information:

This file defines size and contents of the History memory.

History memory	Caracteristics	Value
	Max number of failure	NC_NR_ERR_HIS = 10
	Number of data to store for a failure (in bytes)	NC_NR_HIS = 9
	Contents	ERR_HIS_CONF[1] = ERR_TYPE_BYTE_IDX *BMW error type byte*
		ERR_HIS_CONF[2] = CTR_FRC_IDX *Frequency counter of error IDX*
		ERR_HIS_CONF[3] = ENVD_CUS_SET_CMN_2_1_IDX *Freeze frame data km-counter (high byte) 1st. occur.*
		ERR_HIS_CONF[4] = ENVD_CUS_SET_CMN_3_1_IDX *Freeze frame data km-counter (low byte) 1st. occur.*
		ERR_HIS_CONF[5] = ENVD_CUS_SET_CMN_2_2_IDX *Freeze frame data km-counter (high byte) 2st. occur.*
		ERR_HIS_CONF[6] = ENVD_CUS_SET_CMN_3_2_IDX *Freeze frame data km-counter (low byte) 2st. occur.*
		ERR_HIS_CONF[7] = ENVD_CUS_SET_CMN_2_3_IDX *Freeze frame data km-counter (high byte) last occur.*

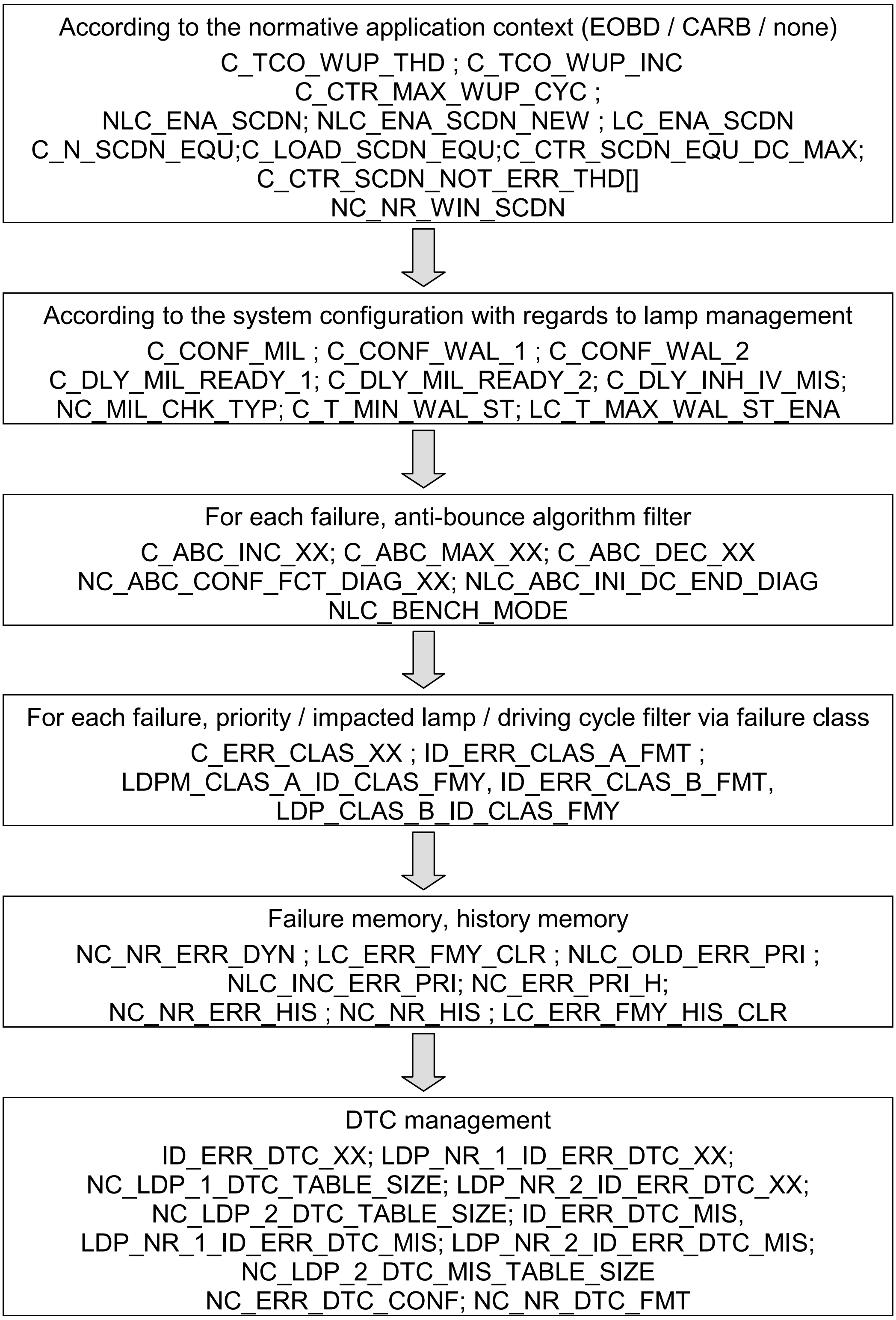
		ERR_HIS_CONF[8] = ENVD_CUS_SET_CMN_3_3_IDX *Freeze frame data km-counter (low byte) last occur.*
		ERR_HIS_CONF[9] = C_ERR_CLAS_XX *Error class*

27.33 Error management (gen)

27.33.1 Calibration interfering function

27.33.2 Calibration flowchart

Although, no strict sequence of calibration is needed for the Dynamic Error Management, a logical order is suggested below.



According to the normative application context (EOBD / CARB / none)
C_TCO_WUP_THD ; C_TCO_WUP_INC
C_CTR_MAX_WUP_CYC ;
NLC_ENA_SCDN; NLC_ENA_SCDN_NEW ; LC_ENA_SCDN
C_N_SCDN_EQU;C_LOAD_SCDN_EQU;C_CTR_SCDN_EQU_DC_MAX;
C_CTR_SCDN_NOT_ERR_THD[]
NC_NR_WIN_SCDN

According to the system configuration with regards to lamp management
C_CONF_MIL ; C_CONF_WAL_1 ; C_CONF_WAL_2
C_DLY_MIL_READY_1; C_DLY_MIL_READY_2; C_DLY_INH_IV_MIS;
NC_MIL_CHK_TYP; C_T_MIN_WAL_ST; LC_T_MAX_WAL_ST_ENA

For each failure, anti-bounce algorithm filter
C_ABC_INC_XX; C_ABC_MAX_XX; C_ABC_DEC_XX
NC_ABC_CONF_FCT_DIAG_XX; NLC_ABC_INI_DC_END_DIAG
NLC_BENCH_MODE

For each failure, priority / impacted lamp / driving cycle filter via failure class
C_ERR_CLAS_XX ; ID_ERR_CLAS_A_FMT ;
LDPM_CLAS_A_ID_CLAS_FMY, ID_ERR_CLAS_B_FMT,
LDP_CLAS_B_ID_CLAS_FMY

Failure memory, history memory
NC_NR_ERR_DYN ; LC_ERR_FMY_CLR ; NLC_OLD_ERR_PRI ;
NLC_INC_ERR_PRI; NC_ERR_PRI_H;
NC_NR_ERR_HIS ; NC_NR_HIS ; LC_ERR_FMY_HIS_CLR

DTC management
ID_ERR_DTC_XX; LDP_NR_1_ID_ERR_DTC_XX;
NC_LDP_1_DTC_TABLE_SIZE; LDP_NR_2_ID_ERR_DTC_XX;
NC_LDP_2_DTC_TABLE_SIZE; ID_ERR_DTC_MIS,
LDP_NR_1_ID_ERR_DTC_MIS; LDP_NR_2_ID_ERR_DTC_MIS;
NC_LDP_2_DTC_MIS_TABLE_SIZE
NC_ERR_DTC_CONF; NC_NR_DTC_FMT

Environmental data (freeze frame)

NC_NR_ENVD_OBD; NC_NR_ENVD_PREV ; NC_NR_ENVD_CUS_CMN;
NC_ENVD_CUS_CMN_UPD;
NC_NR_ENVD_CUS_SET_CMN ; NC_NR_ENVD_CUS_SET_SPC
NC_NR_FRF_SET ; ID_ENVD_FAC[255]; NC_ID_ENVD_FAC;
ID_ERR_ENVD_XX; LDP_1_ID_ERR_ENVD_XX

Communication interface (API)

C_STATE_READY_OBD_1; C_STATE_READY_OBD_2;
C_STATE_READY_OBD_3; C_STATE_READY_OBD_CMPL_OBD_1;
NC_ERR_DTC_REQ_OBD ; NC_ERR_DTC_REQ_CUS ;
NLC_OBD_FRF_PND; LC_STATE_READY_OBD_CARB_CC

Rate-based monitoring

NLC_OBD_RBM_ENA ; LC_RBM_CLR
NC_NR_DIAG_RBM ; C_AMP_MIN_RBM ; C_DLY_AST_RBM ;
C_T_AST_RBM ; C_T_IS_RBM ; C_T_VS_RBM ;
C_TAM_MIN_RBM ; C_VS_THD_RBM ; C_VS_THD_IS_CDN_RBM ;
C_PV_IS_RBM ;

27.33.3 Calibration process

27.33.3.1 Warm-up cycle definition

A warm-up cycle is detected as soon as the coolant temperature reaches a threshold value and the gap with starting temperature exceeds another value. Another calibration value defined the number of warm up cycle to wait, to erase the failure in memory. The decrementation of the warm up counter begins when the MIL is off and the failure is no more pending.

For Warm up cycle detection, light different value are given by the standards:

For gasoline engines (GS):

	EOBD	CARB
C_TCO_WUP_INC	22 Kelvin / 22°C	40 Fahrenheit / 22.22°C
C_TCO_WUP_THD	343 Kelvin / 69.85°C	160 Fahrenheit / 71,11°C
C_CTR_MAX_WUP	40	40

For diesel engines (DS):

	EOBD	CARB
C_TCO_WUP_INC	22 Kelvin / 22°C	40 Fahrenheit / 22.22°C

C_TCO_WUP_THD	343 Kelvin / 69.85°C	140 Fahrenheit / 60°C
C_CTR_MAX_WUP	40	40

27.33.3.2 Similar conditions

Similar conditions are additional conditions based on engine and load status to erase a failure in memory (for Misfire and Fuel system failure only).

⇨ For EOBD or non CARB application
Similar conditions are only required for CARB application, so it is possible to inhibit them for non-CARB application.
NLC_ENA_SCDN = 0
LC_ENA_SCDN = 0 or 1
Note : whatever the value of LC_ENA_SCDN, when NLC_ENA_SCDN = 0, it's impossible to enable similar conditions.

⇨ For CARB application
According CARB standard similar conditions are required with the following value:
NLC_ENA_SCDN = 1
LC_ENA_SCDN = 1
Note : if NLC_ENA_SCDN=1, it's possible to enable or disable similar conditions. But for CARB application it's mandatory to activate them with LC_ENA_SCDN = 1.
With the configuration NLC_ENA_SCDN_NEW the new SCDN functionality (SCDN window around the point, when failure is set in ERRM) could be used.

Exemple for a CARB application:

Calibration name	Physical value	Example of calibration value
LC_ENA_SCDN	1	1
C_N_SCDN_EQU	375 rpm	0B (Hex)
C_LOAD_SCDN_EQU	20 %	-
C_CTR_SCDN_EQU_DC_MAX	80	50 (Hex)

If new SCDN functionality is used additonal the calibration C_CTR_SCDN_NOT_ERR_THD[m] must be adapted to the system needs for each SCDN instance m.

The data NC_NR_WIN_SCDN is configured with the total number of instances used for similar condition calculation.

27.33.3.3 Lamp management

All warning lamp (MIL, WAL_1, WAL_2) configurations are managed with a dedicated calibration:
- C_CONF_MIL for MIL
- C_CONF_WAL_1 for WAL_1
- C_CONF_WAL_2 for WAL_2

It is possible to inhibit a lamp illumination, to inhibit a pre-drive check for a lamp, to allow or not external request for lamp illumination and to link MIL illumination with another lamp (WAL_1, WAL_2).

For vehicles which have a MIL blinking for readiness status displaying during pre-drive check, NC_MIL_CHK_TYP, C_DLY_MIL_READY_1 and C_DLY_MIL_READY_2 must be defined.

Bit 1 of NC_MIL_CHK_TYP = 1: The readiness status are displayed
C_DLY_MIL_READY_1: Delay to start MIL blinking for readiness display during pre-drive check (ARB: 15-20s)
C_DLY_MIL_READY_2: MIL blinking duration for readiness display during pre-drive check (ARB: 5-10s)

Example 1:

Calibration name	Physical value	Example of calibration value
C_CONF_MIL	-	2 (hex) / 0000 0010 (binary)
C_CONF_WAL_1	-	2 (hex) / 0000 0010 (binary)
C_CONF_WAL_2	-	0
C_DLY_MIL_READY_1	20 seconds	28 (Hex)
C_DLY_MIL_READY_2	10 seconds	14 (Hex)

In this example:

- The ECU manages three lamps
- MIL and WAL_1 support external request
- WAL_2 doesn’t support external request
- The pre-drive check is managed by the ECU

As regard Misfiring strategies, if cylinder shut-off is used in case of misfire A detection, C_CONF_MIL and C_DLY_INH_IV_MIS shall be adapted.

27.33.3.4 Anti-bounce Algorithms

27.33.3.4.1 Anti-bounce counter and statistic filters for diagnostics

The anti-bounce algorithm is used to filter a detected failure. The anti-bounce counter is managed with three calibrations:

- C_ABC_INC_XX, to increment the counter while an error is detected
- C_ABC_MAX_XX, to indicate the maximum value of the anti-bounce counter. If the maximum is reached the failure gets present.
- C_ABC_DEC_XX, to decrement the counter when an error is not detected.

Additionally the calibration LC_ABC_BENCH permits to switch on the bench mode for set / reset the failure manually (useful in development phase). The calibration LC_ERR_DET_UPD permits to manage the visualisation of the sporadic error. Both strategies must be included in software by configurations.

The bench mode is useful to set / reset failure manually for development purposes:
i) LC_ABC_BENCH = 1: bench mode is active.
ii) LC_ERR_DET_UPD = 1: sporadic visualisation mode is active.
iii) Then, when LC_ABC_BENCH = 1, use C_ABC_INC_XX to set / reset the failure.
C_ABC_INC_XX = 255: the failure is set
C_ABC_INC_XX = 0: the failure is reset

Important : It is NOT recommended to have the bench mode activated for the serial calibration.

With bench mode activated and C_ABC_INC_XX = 0:
ERR_SYM_XX = 0 LV_ERR_XX = 0 CTR_ABC_XX = 0
LV_CDN_DIAG_XX = 1 LV_END_DIAG_XX = 1

With bench mode activated and C_ABC_INC_XX = FFH:
ERR_SYM_XX = 0Fh LV_ERR_XX = 1 CTR_ABC_XX = C_ABC_MAX_XX
LV_CDN_DIAG_XX = 1 LV_END_DIAG_XX = 1

Without bench mode and C_ABC_INC_XX = 0:
ERR_SYM_XX = 0 LV_ERR_XX = 0 CTR_ABC_XX = 0
LV_END_DIAG_XX = 0 LV_CDN_DIAG_XX = 0 or 1

Each diagnosis has a specified filter type : STD_INI, STD, DEC_CAL, MEM, MEM_INI, NO.

Filter types

Meaning / Result	STD	STD_ INI	DEC_ CAL	MEM	MEM_ INI	STC	NO
LV_ERR_XX and CTR_ABC_XX are saved	**X**		**X**				**Not used**
LV_ERR_XX and CTR_ABC_XX aren't saved		**X**		**X**	**X**		**Not used**
LV_ERR_XX and CTR_ABC_XX are restored from NVMY at ECU reset	**X**		**X**				**Not used**
LV_ERR_XX and CTR_ABC_XX are initialized at LV_IGK 0 → 1 transition or reset		**X**		**X**	**X**	**X**	**Not used**
LV_ERR_XX and CTR_ABC_XX are initialized with the 0 value		**X**	**X**	**X**		**X**	**Not used**
LV_ERR_XX is initialized to 0 and CTR_ABC_XX is initialized with (C_ABC_MAX_XX - C_ABC_INC_XX)					**X**		**Not used**
The anti-bounce counter is decremented when the diagnosis condition are fulfilled without detected failure	**X**	**X**	**X**				**Not used**
The anti-bounce counter is decremented when the diagnosis condition are fulfilled without detected failure only if it hasn't reached the maximum value.				**X**	**X**		**Not used**
The anti-bounce decrement is equal to 1	**X**	**X**		**X**	**X**		**Not used**
The anti-bounce decrement is calibratable			**X**				**Not used**
Filter type is an anti-bounce mechanism	**X**	**X**	**X**	**X**	**X**		**Not used**
The anti-bounce decrement is calibratable						**X**	**Not used**

Important :
- For statistical filtering (STC), it's mandatory to respect the following criterion :
C_ABC_MAX_XX > C_ABC_INC_XX

27.33.3.4.2 Multicondition debounce algorithm

The multicondition debounce algorithm is used to filter a detected failure. This filter is used in case of the condition activation to do the test is not the same for each symptom. (e.g. output power diagnostic)

The anti-bounce counter is managed with two calibrations:
- C_ABC_INC_XX, to increment the counter while an error is detected
- C_ABC_MAX_XX, to indicate the maximum value of the anti-bounce counter. If the maximum is reached the failure gets present.

Bench mode and sporadic errors detection are also available with multicondition debounce algorithm (see §1.3.4.1)

For anti-bounce algorithm calibration (C_ABC_INC_XX, C_ABC_MAX_XX) depends on the sensor sensibility and the diagnosis recurrence.

Note : for each failure the only possible filtering configuration is MPL_STD_INI :

Meaning / Result	Filter type MPL_STD_INI
LV_ERR_XX and CTR_ABC_XX are saved	
LV_ERR_XX and CTR_ABC_XX aren't saved	X
LV_ERR_XX and CTR_ABC_XX are restored from NVMY at ECU reset	
LV_ERR_XX and CTR_ABC_XX are initialized at LV_IGK 0 → 1 transition or reset	X
LV_ERR_XX and CTR_ABC_XX are initialized with the 0 value	X
LV_ERR_XX is initialized to 0 and CTR_ABC_XX is initialized with (C_ABC_MAX_XX - C_ABC_INC_XX)	
The anti-bounce counter is decremented when the diagnosis condition are fulfilled without detected failure	X
The anti-bounce counter is decremented when the diagnosis condition are fulfilled without detected failure only if it hasn't reached the maximum value.	
The anti-bounce decrement is equal to 1	X
The anti-bounce decrement is calibratable	

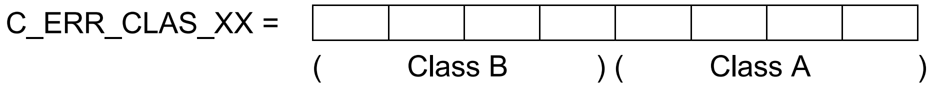
27.33.3.5 Failure Classes

The calibration of a failure, regarding failure treatment within the error management, is made via failure classes with a calibration named C_ERR_CLAS_XX. Each class defines the type of this failure as emission relevant or not, as MIL handling or not, as using driving cycle for MIL illumination, etc...

Definition of a failure class:

The failure class number is a byte divided in two subclasses named class B (defined by the most significant bits) and class A (defined by the less significant bits).

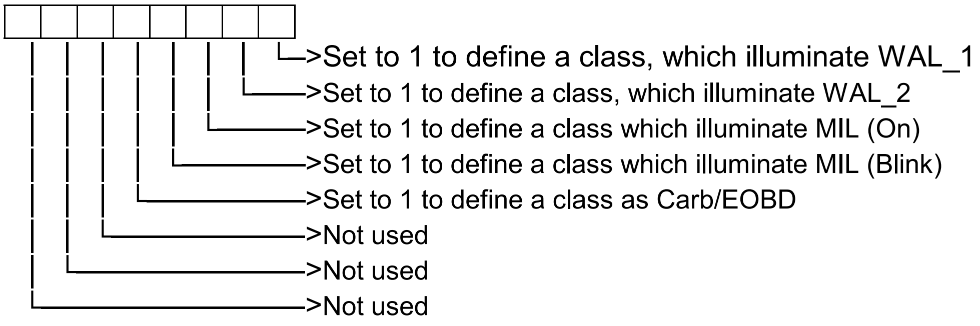
Failure class number for failure XX



Sub classes definitions are given with ID_ERR_CLAS_A_FMT and ID_ERR_CLAS_B_FMT:

Sub class A definition: (16 different subclasses A can be defined) Defines the impact of a failure on the different lamps (MIL, WAL_1 and WAL_2). To define subclass A number i, use ID_ERR_CLAS_A_FMT [i].

ID_ERR_CLAS_A_FMT [i]



WAL_1 (Warning Lamp 1), WAL_2 (Warning Lamp 2) and MIL (Malfunction Indicator Lamp) are lamps to indicate malfunction on the vehicle.

- When a failure occurs, WAL_1 can be immediately illuminated according the class of the failure. Idem for WAL_2 lamp.
- When a failure occurs, MIL can be illuminated when its driving cycle counter reach the maximum value according the class of the failure.
- When a failure occurs, MIL can blinked immediately according the class of the failure.

Sub class B definition: to be used to define priority and driving cycle counter (maximum, increment, decrement) of the failure.

To define sub class B number i use:

- ID_ERR_CLAS_B_FMT [i] [0] = increment value of the driving cycle counter
- ID_ERR_CLAS_B_FMT [i] [1] = maximum value of the driving cycle counter
- ID_ERR_CLAS_B_FMT [i] [2] = decrement value of the driving cycle counter
- ID_ERR_CLAS_B_FMT [i] [3] = priority of the failure

Definition of a class for a failure : to define the class of the failure XX, the calibration C_ERR_CLAS_XX must be used. This calibration contains two hexadecimal digits values :

- first digit represents the sub class B value of the failure XX (from 0H to FH)
- second digit represents the sub class A value of the failure XX (from 0H to FH)

Sub class A definition : to make easier the calibration, some pre-defined value are given (sub class 0 to 4), the others value is free for customer specific definitions (sub class 5 to FH) :

Sub Class A definition : ID_ERR_CLAS_A_FMT (1 byte x 16)

Subclass A Value (i) in hex / Name	**Description**	**ID_ERR_CLAS_A_FMT [i]**
0 / NO_LAMP	*- Predefined value –* Failure without impact on any lamp	%0000 0000 (in binary)
1 / WAL_1_ON	*- Predefined value -* Failure with impact on WAL1 lamp	%0000 0001 (in binary)
2 / WAL_2_ON	*- Predefined value –* Failure with impact on WAL2 lamp	%0000 0010 (in binary)
3 / MIL_ON	*- Predefined value -* Failure with impact on Mil (on)	%0001 0100 (in binary)
4 / MIL_ON_FLL	*- Predefined value –* Failure with impact on Mil (on & blinking)	%0001 1100 (in binary)
5 / LAMP_CUS_1	*Free for customer definition*	*Free for customer definition*
6 / LAMP_CUS_2	*Free for customer definition*	*Free for customer definition*
7 / LAMP_CUS_3	*Free for customer definition*	*Free for customer definition*
8 / LAMP_CUS_4	*Free for customer definition*	*Free for customer definition*
9 / LAMP_CUS_5	*Free for customer definition*	*Free for customer definition*
A / LAMP_CUS_6	*Free for customer definition*	*Free for customer definition*
B / LAMP_CUS_7	*Free for customer definition*	*Free for customer definition*
C / LAMP_CUS_8	*Free for customer definition*	*Free for customer definition*
D / LAMP_CUS_9	*Free for customer definition*	*Free for customer definition*
E / LAMP_CUS_10	*Free for customer definition*	*Free for customer definition*
F / LAMP_CUS_11	*Free for customer definition*	*Free for customer definition*

For failure without impact on emissions (non-CARB failure) : These failures have no impact on MIL ; according customer requirement, these failures might have impact on other lamp.

⇨ For non-CARB failures, use sub class A number 0 (NO_LAMP), 1 (WAL_1) or 2 (WAL_2) according customer requirement

For failure with impact on emissions (CARB failure) : These failures have impact on MIL. For CARB failure, use sub class A number 3 (MIL_ON)

Remark: A failure is considered as a CARB failure (emission relevant) when the bit 4 of ID_ERR_CLAS_A_FMT[i] = 1 (failure visible with scantools, failure can set OBD freeze frame for mode 2).

Sub class B definition : To make easier the calibration, some pre-defined value are given (sub class 0 to 7), the others value is free for customer definition (sub class 8 to FH) :

Sub Class B definition : ID_ERR_CLAS_B_FMT (4 bytes x 16)

		ID_ERR_CLAS_B_FMT			
Subclass B Value (i) in hex/ Name	**Description**	**[i] [0] DC inc**	**[i] [1] DC max**	**[i] [2] DC dec**	**[i] [3] Priority**
0 / NO_LAW_L	*- Predefined value –* Non Carb/EOBD failure with low priority	1	1	1	0
1 / NO_LAW_M	*- Predefined value -* Non Carb/EOBD failure with medium priority	1	1	1	1
2 / NO_LAW_H	*- Predefined value –* Non Carb/EOBD failure with high priority	1	1	1	2
3 / LAW_CC_L	*- Predefined value -* Carb/EOBD failure with low priority Comprehensive component	3	3	1	4
4 / LAW_L	*- Predefined value -* Carb/EOBD failure with low priority	3 / 2 *)	6	2	4
5 / LAW_M	*- Predefined value -* Carb/EOBD failure with medium priority	3 / 2 *)	6	2	5
6 / LAW_H	*- Predefined value -* Carb/EOBD failure with high priority	3 / 2 *)	6	2	6
7 / VALID	For validation	1	1	1	4
8 / NO_ERASE	*- Predefined value -* failure with high priority, not erasable by warm-up cycles.	3	3	0	6
9 / DC_CUS_1	*Free for customer definition*	-	-	-	-
A / DC_CUS_2	*Free for customer definition*	-	-	-	-
B / DC_CUS_3	*Free for customer definition*	-	-	-	-
C / DC_CUS_4	*Free for customer definition*	-	-	-	-
D / DC_CUS_5	*Free for customer definition*	-	-	-	-
E / DC_CUS_6	*Free for customer definition*	-	-	-	-
F / DC_CUS_7	*Free for customer definition*	-	-	-	-

Remark : The increment of the DC counter (*) for the failures defined by the failure classes LAW_L, LAW_M and LAW_H depends on the required regulation (Carb/EOBD).

For OBD1 failure without impact on emissions (non-CARB failure) :

For such kind of failure, the driving cycle counter algorithm filtering isn’t needed because anti-bounce algorithm already exists for these electrical failures. So, it is recommended to store immediately the failure after anti-bounce filtering. In addition, these failures have no impact on the MIL.
Therefore, for such failures : C_CTR_MAX_XX_DC = C_CTR_INC_XX_DC (≥ 1)

It is also recommended to start decrementing the warm-up counter immediately when the failure is no more present.

Therefore, for such failures : C_CTR_MAX_XX_DC = C_CTR_DEC_XX_DC (≥ 1)
Concerning priority, according standards, failure without impact on emission must have the lowest priority (for customer requirement, different level of lowest priority can be defined).

⇨ For non-CARB failure, use sub class B number 0 (NO_LAW_L), 1 (NO_LAW_M) or 2 (NO_LAW_H).

For OBD1 and OBD2 failure with impact on emissions (CARB failure) :
These failures have impact on the MIL. According to EOBD / CARB standards, the MIL must be deactivated after three subsequent sequential driving cycle without failure.
Therefore, for such failures : C_CTR_MAX_XX_DC = 3 * C_CTR_DEC_XX_DC
In addition for such failures : C_CTR_MAX_XX_DC = k * C_CTR_INC_XX_DC
(with k ≤ 2 ; CARB recommendation)

With k = 2 :for failure defined in CARB / EOBD standards (Ex : Fuel System, Misfiring,...) and for failure on comprehensive components without perceptible impact for the vehicle driver (Ex : TCO).

With k = 1 :for other failure on comprehensive components with perceptible impact for the vehicle driver (Ex : Injector valve).

Concerning priority, according standards, failure with impact on pollution must have the following priority :

- fuel system and misfiring failure :	highest priority
- all other failure with impact on pollution :	medium priority

⇨ For CARB failure defined in CARB /EOBD standards, use sub class B number 4 (LAW_L), 5 (LAW_M) or 6 (LAW_H)

⇨ For others comprehensive component CARB failure, use sub class B number 3 (LAW_CC_L)

Particular case for development phase:
In development phase, it might be useful to have immediately effect on the MIL when an error is present. In this particular case, the increment, maximum and decrement calibrations must be equal. The sub class B number 7 (VALID) have been defined for this behaviour.

Particular case for failure no erase:
Some cutomers want be able to keep the failure and so no erase after 40 warm-up cycle.

27.33.3.6 Failure memory / history memory definition

Failure and history memories sizes are defined by configurations.

For development conveniences, it is often useful to clear these memories :

- The failure memory can be erased with a single LC_ERR_FMY_CLR 0 → 1 transition

- The history memory can be erased with a single LC_ERR_FMY_HIS_CLR 0 → 1 transition.

Warning : after reset of failure memory, LC_ERR_FMY_CLR must be reset to 0, nominal value of this calibration.

27.33.3.7 Data Trouble Code (DTC)

27.33.3.7.1 DTC definition for diagnostic instance and misfiring

For each diagnosis XX, a DTC indentification number is defined in ID_ERR_DTC_XX calibration table. A DTC is an hex value identifier encoded as defined in J2012 normative document and according the customer specific definition.

A default value can be found in each diagnosis specification.

If a default value is not availble, please refer to the J2012 normative document. It contains an exhaustive list of normalised DTC codes.

The calibrated DTC, upon failure detection, will be displayed on scantool, customer specific tool and ERR_DTC_IDX system variable.

This calibration shall be filled up with non null values. If one or more symptoms are unused, corresponding DTCs shall be assigned to a non null value.

ID_ERR_DTC_XX[0], ID_ERR_DTC_XX[1], ID_ERR_DTC_XX[2], ID_ERR_DTC_XX[3]
shall be filled up with non null values (NC_NR_DTC_FMT = 0).

ID_ERR_DTC_XX[0][0], ID_ERR_DTC_XX[0][1], ID_ERR_DTC_XX[0][2], ID_ERR_DTC_XX[0][3]
ID_ERR_DTC_XX[1][0], ID_ERR_DTC_XX[1][1], ID_ERR_DTC_XX[1][2], ID_ERR_DTC_XX[1][3]
shall be filled up (NC_NR_DTC_FMT = 1).

The global J2012 encoded DTC shall also be filled up. This DTC is used in Scan Tool Mode $02, if more than one symptoms are active in ERR_SYM_MEM_IDX at the same time.

ID_ERR_DTC_XX[4], ID_ERR_DTC_XX[5]
shall be filled up with non null values (NC_NR_DTC_FMT = 0).

ID_ERR_DTC_XX[0][4], ID_ERR_DTC_XX[1][4]
shall be filled up with non null values (NC_NR_DTC_FMT = 1).

a) For the case NC_NR_DTC_FMT = 0 :

- ID_ERR_DTC_XX [0] : J2012 encoded DTC indentifier linked to the first symptom of failure XX ;
- ID_ERR_DTC_XX [1] : J2012 encoded DTC indentifier linked to the second symptom of failure XX;
- ID_ERR_DTC_XX [2] : J2012 encoded DTC indentifier linked to the third symptom of failure XX ;
- ID_ERR_DTC_XX [3] : J2012 encoded DTC indentifier linked to the fourth symptom of failure XX ;

- ID_ERR_DTC_XX [4] :global J2012 encoded DTC indentifier linked to the failure XX (does not depend on the detected symptom) ;
- ID_ERR_DTC_XX [5] : global customer encoded specific DTC indentifier.

There is a particular case for misfiring (MIS_A, MIS_B1, MIS_B4) diagnoses. The DTC is defined by cylinder within the ID_ERR_DTC_MIS calibration table.

- ID_ERR_DTC_MIS [0][i] is the J2012 encoded DTC for misfiring detection on the cylinder number i +1 ;
- ID_ERR_DTC_MIS [0][NC_CYL_NR] is the J2012 encoded DTC for global (multiple) misfiring ;
- ID_ERR_DTC_MIS [1][i] is the customer encoded DTC for misfiring detection on the cylinder number i+1 ;
- ID_ERR_DTC_MIS [1][NC_CYL_NR] is the customer encoded DTC for global (multiple) misfiring.

b) For the case NC_NR_DTC_FMT = 1 :

- ID_ERR_DTC_XX [0] [0] : J2012 encoded DTC indentifier linked to the first symptom of failure XX ;
- ID_ERR_DTC_XX [0] [1] : J2012 encoded DTC indentifier linked to the second symptom of failure XX;
- ID_ERR_DTC_XX [0] [2] : J2012 encoded DTC indentifier linked to the third symptom of failure XX ;
- ID_ERR_DTC_XX [0] [3] : J2012 encoded DTC indentifier linked to the fourth symptom of failure XX ;
- ID_ERR_DTC_XX [0] [4] :global J2012 encoded DTC indentifier linked to the failure XX (does not depend on the detected symptom) ;

- ID_ERR_DTC_XX [1] [0] : customer encoded specific DTC indentifier linked to the first symptom of failure XX ;
- ID_ERR_DTC_XX [1] [1] : customer encoded specific DTC indentifier linked to the second symptom of failure XX;
- ID_ERR_DTC_XX [1] [2] : customer encoded specific DTC indentifier linked to the third symptom of failure XX ;
- ID_ERR_DTC_XX [1] [3] : customer encoded specific DTC indentifier linked to the fourth symptom of failure XX ;
- ID_ERR_DTC_XX [1] [4] : global customer encoded specific DTC indentifier.

For the particular case of misfiring (MIS_A, MIS_B1, MIS_B4) diagnosis, DTC indentifiers are defined by cylinder in a calibration table called ID_ERR_DTC_MIS. (used if NLC_TREAT_DIAG_MIS = 0; Misfire by type)

- ID_ERR_DTC_MIS[0] [i] is the DTC for misfiring detection on the cylinder number i +1
- ID_ERR_DTC_MIS[0] [NC_CYL_NR] is the DTC for multiple misfiring
- ID_ERR_DTC_MIS[0] [NC_CYL_NR +1] is the DTC for random misfiring

- ID_ERR_DTC_MIS[1] [i] is the customer DTC for misfiring detection on the cylinder number i+1
- ID_ERR_DTC_MIS[1] [NC_CYL_NR] is the customer DTC for multiple misfiring
- ID_ERR_DTC_MIS[1] [NC_CYL_NR +1] is the customer DTC for random misfiring

Example 1:

The DTC calibration table for the TCO diagnosis (coolant temperature) is defined below :

DTC of the 1st symptom	DTC of the 2nd symptom	DTC of the 3rd symptom	DTC of the 4th symptom	Global DTC of the failure	Specific customer code
ID_ERR_DTC_ TCO [0] = 0118H, (P0118)	ID_ERR_DTC_ TCO [1] = 0117H (P0117)	ID_ERR_DTC_ TCO [2] = 0116H (P0116)	ID_ERR_DTC_ TCO [3] = 0115H (P0115)	ID_ERR_DTC_ TCO [4] = 0114H (P0114)	ID_ERR_DTC_ TCO [5] = 32H

In case this TCO failure is detected and stored into memory, DTC visualization with application system variable ERR_DTC_IDX will be :

- If NC_ERR_DTC_CONF = 0

  then the symptom DTC is visible on application system :

  ERR_DTC_IDX = 0118H, if failure is detected with 1st symptom

  ERR_DTC_IDX = 0117H, if failure is detected with 2nd symptom

  ERR_DTC_IDX = 0116H, if failure is detected with 3rd symptom

  ERR_DTC_IDX = 0115H, if failure is detected with 4th symptom

  ERR_DTC_IDX = 0114H, if failure is detected with multiple symptoms (more than one symptom active at the same time)

- If NC_ERR_DTC_CONF = 1

  then the global DTC is visible on application system : ERR_DTC_IDX = 0114H

- If NC_ERR_DTC_CONF = 2

  then the customer DTC is visible on application system : ERR_DTC_IDX = 0032H

If symptom 3 and 4 are not used, corresponding DTCs can be asigned to 0114H

Example 2:

One failure misfiring (MIS_A, MIS_B1 or MIS_B4) is stored in the 2nd layer memory. The application is a 6 cylinders system. The DTC calibratable tables is defined below :

ID_ERR_DTC_MIS :

	Misfiring on 1st cylinder	Misfiring on 2nd cylinder	Misfiring on 3rd cylinder	Misfiring on 4th cylinder	Misfiring on 5th cylinder	Misfiring on 6th cylinder	Global (multiple) Misfiring
OBD DTC	ID_ERR_DTC_ MIS [0][0] = 0301H (P0301)	ID_ERR_DTC_ MIS [0][1] = 0302H (P0302)	ID_ERR_DTC_ MIS [0][2] = 0303H (P0303)	ID_ERR_DTC_ MIS [0][3] = 0304H (P0304)	ID_ERR_DTC_ MIS [0][4] = 0305H (P0305)	ID_ERR_DTC_ MIS [0][5] = 0306H (P0306)	ID_ERR_DTC_ MIS [0][6] = 0300H (P0300)

Customer DTC	ID_ERR_DTC _MIS [1][0] = 11H	ID_ERR_DTC_ MIS [1][1] = 12H	ID_ERR_DTC_ MIS [1][2] = 13H	ID_ERR_DTC_ MIS [1][3] = 14H	ID_ERR_DTC_ MIS [1][4] = 15H	ID_ERR_DTC_ MIS [1][5] = 16H	ID_ERR_DTC_ MIS [1][6] = 10H

⇨ DTC visualization with application system with ERR_DTC_IDX (if failure is stored in the memory):

- If NC_ERR_DTC_CONF = 0

then the cylinder individual OBD DTC is visible on application system :

ERR_DTC_IDX = 0301H, if misfire is detected on cylinder 1
ERR_DTC_IDX = 0302H, if misfire is detected on cylinder 2
ERR_DTC_IDX = 0303H, if misfire is detected on cylinder 3
ERR_DTC_IDX = 0304H, if misfire is detected on cylinder 4
ERR_DTC_IDX = 0305H, if misfire is detected on cylinder 5
ERR_DTC_IDX = 0306H, if misfire is detected on cylinder 6

Only in case of multiple misfire:

ERR_DTC_IDX = 0300H

- If NC_ERR_DTC_CONF = 1

then the global (multiple) misfire OBD DTC is visible on application system :

ERR_DTC_IDX = P0300

- If NC_ERR_DTC_CONF = 2

then the cylinder individual customer DTC is visible on application system :

ERR_DTC_IDX = 11H, if misfire is detected on cylinder 1
ERR_DTC_IDX = 12H, if misfire is detected on cylinder 2
ERR_DTC_IDX = 13H, if misfire is detected on cylinder 3
ERR_DTC_IDX = 14H, if misfire is detected on cylinder 4
ERR_DTC_IDX = 15H, if misfire is detected on cylinder 5
ERR_DTC_IDX = 16H, if misfire is detected on cylinder 6

Only in case of multiple misfire:

ERR_DTC_IDX = 10H

Example 3:

The failure TCO (coolant temperature) is stored. The calibratable table of DTC has the following values:

a) with NC_NR_DTC_FMT = 0

DTC of the 1st symptom	DTC of the 2nd symptom	DTC of the 3rd symptom	DTC of the 4th symptom	Global DTC of the failure	Specific customer code
ID_ERR_TCO [0][0] = 0118H (P0118)	ID_ERR_TCO [0][1] = 0117H (P0117)	ID_ERR_TCO [0][2] = 0119H (P0119)	ID_ERR_TCO [0][3] = 0116H (P0116)	ID_ERR_TCO [0][4] = 0115H (P0115)	ID_ERR_TCO [0] [5] = 30

b) with NC_NR_DTC_FMT = 1

	DTC of the 1st symptom	DTC of the 2nd symptom	DTC of the 3rd symptom	DTC of the 4th symptom	Global DTC of the failure
OBD DTC	ID_ERR_TCO [0][0] = 0118H (P0118)	ID_ERR_TCO [0] [1] = 0117H (P0117)	ID_ERR_TCO [0][2] = 0119H (P0119)	ID_ERR_TCO [0][3] = 0116H (P0116)	ID_ERR_TCO [0][4] = 0115H (P0115)
Customer DTC	ID_ERR_TCO [1] [0] = 31H	ID_ERR_TCO [1] [1] = 32H	ID_ERR_TCO [1] [2] = 33H	ID_ERR_TCO [1] [3] = 34H	ID_ERR_TCO [1] [4] = 30H

Upon request from the diagnosis tool, the DTC number sent to the tool will be one of the numbers defined in ID_ERR_DTC_TCO depending on symptom detected.

Example 4:

One failure misfiring (LV_ERR_MIS_A, LV_ERR_MIS_B1 or LV_ERR_MIS_B4) is stored in the 2nd layer memory. The application is a NC_CYL_NR -cylinder engine. The calibratable table of DTC has the following values :

	Misfiring on 1st cylinder	Misfiring on 2nd cylinder	Misfiring on 3rd cylinder	Misfiring On 4th Cylinder	...	Misfiring on NC_CYL_NR th cylinder	Multiple Misfiring	Random Misfiring
OBD DTC	ID_ERR_MIS [0][0] = 0301H (P0301)	ID_ERR_MIS [0][1] = 0302H (P0302)	ID_ERR_MIS [0][2] = 0303H (P0303)	ID_ERR_MIS [0][3] = 0304H (P0304)	...	ID_ERR_MIS [0] [NC_CYL_NR - 1] = 03xxH (P03xx)	ID_ERR_MIS [0] [NC_CYL_NR] = 0300H (P0300)	ID_ERR_MIS [0] [NC_CYL_NR + 1] = 0300H (P0300)
Customer DTC	ID_ERR_MIS [1][0] = 11H	ID_ERR_MIS [1][1] = 12H	ID_ERR_MIS [1][2] = 13H	ID_ERR_MIS [1][3] = 14H	...	ID_ERR_MIS [1] [NC_CYL_NR - 1] = 16H	ID_ERR_MIS [1] [NC_CYL_NR] = 10H	ID_ERR_MIS [1] [NC_CYL_NR + 1] = 10H

Upon request from the diagnosis tool, the DTC number sent to the tool will be one of the numbers defined in ID_ERR_DTC_MIS regards the symptom detected.

27.33.3.8 Environmental data (freeze frame size and contents)

The major part of freeze frame content has to be configured. Only the content of ENVD_CUS_SET_SPC (specific to each failure and stored many times) can be calibrated for each failure.

The content of this part is defined by using calibration ID_ERR_ENVD_XX and ID_ENVD_FAC configuration table.

ID_ERR_ENVD_XX:	Calibration table which defines for each failure XX the environmental data which should be stored in the set of freeze frame related to failure XX.

Example:

In order to store specific environmental data at each occurrence of a failure as follow:

- TIA and TCO_ST in case of TCO failure;
- MAP and AMP in case of EGR failure;

With :

ID_ENVD_FAC [1] = &TIA
ID_ENVD_FAC [2] = &TCO_ST
ID_ENVD_FAC [3] = &MAP (msb)
ID_ENVD_FAC [4] = &MAP (lsb)
ID_ENVD_FAC [5] = &AMP (msb)
ID_ENVD_FAC [6] = &AMP (lsb)

Then, for each failure, the environmental data to be stored is calibrated as follow :

ID_ERR_ENVD_TCO = [1,2,0,0]
ID_ERR_ENVD_EGR = [3,4,5,6]

27.33.3.9 Communication Interface

27.33.3.9.1 Readiness codes

Within the communication interface the behavior of functionalities in case of communication tool request must be defined:

- Readiness Code requested by law

Readiness code is requested by the CARB, with a scantool, to check if the diagnosis of each function has been performed.

C_STATE_READY_OBD_1:

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 |  |  |  |  |  |  |  └─>  bit 0 : Misfire monitoring supported
 |  |  |  |  |  |  └───>  bit 1 : Fuel system monitoring supported
 |  |  |  |  |  └─────>  bit 2 : Comprehensive component monitoring supported
 |  |  |  |  └───────>  bit 3 : 0 (reserved)
 |  |  |  └─────────>  bit 4 : X (not used)
 |  |  └───────────>  bit 5 : X (not used)
 |  └─────────────>  bit 6 : X (not used)
 └───────────────>  bit 7 : X (not used)
```

C_STATE_READY_OBD_2:

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 |  |  |  |  |  |  |  └─>  bit 0 : Catalyst monitoring supported
 |  |  |  |  |  |  └───>  bit 1 : Heated catalyst
 |  |  |  |  |  └─────>  bit 2 : Evaporative system monitoring supported
 |  |  |  |  └───────>  bit 3 : Secundary air system monitoring supported
 |  |  |  └─────────>  bit 4 : A/C system refrigerant monitoring
 |  |  └───────────>  bit 5 : Oxygen sensor monitoring supported
 |  └─────────────>  bit 6 : Oxygen sensor heater monitoring
 └───────────────>  bit 7 : EGR system monitoring supported
```

For comprehensive components (others components which may have impact indirectly on emission) the diagnosis has to be done a certain number of time to update readiness code.

An exception is made on comprehensive components, misfire and fuel system monitors. Legislation allows to set directly "ready" without computing theses 3 monitors.
This strategy can be activated with the calibration C_STATE_READY_OBD_CMPL_OBD_1 as defined below.

Readiness code calculation for CARB_MIS, CARB_FSD, CARB_CC diagnosis :
0: readiness of concerned group is calculated based on the readiness of each comprehensive component diagnostic
1: readiness of concerned group always indicates "ready"

C_STATE_READY_CMPL_OBD_1:

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 |  |  |  |  |  |  |  └─>  bit0 : 0
 |  |  |  |  |  |  └───>  bit1 : 0
 |  |  |  |  |  └─────>  bit2 : 0
 |  |  |  |  └───────>  bit3 : 0
 |  |  |  └─────────>  bit4 : set directly ready Misfire monitoring status
 |  |  └───────────>  bit5 : set directly ready Fuel system monitoring status
 |  └─────────────>  bit6 : set directly ready Comprehensive component monitoring status
 └───────────────>  bit7 : 0
```

- Readiness Code requested by customer

There is also a possibility to manage customer specific readiness with C_STATE_READY_OBD_3 defined as below :

C_STATE_READY_OBD_3:

```
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 |  |  |  |  |  |  |  └─>  bit 0 : for diagnosis classified as CARB_OTHER
 |  |  |  |  |  |  └───>  bit 1 : for diagnosis classified as NO_CARB
 |  |  |  |  |  └─────>  bit 2 : 0
 |  |  |  |  └───────>  bit 3 : 0
 |  |  |  └─────────>  bit 4 : 0
 |  |  └───────────>  bit 5 : 0
 |  └─────────────>  bit 6 : 0
 └───────────────>  bit 7 : 0
```

Typical value for readiness code requested by law

Calibration name	Example of calibration
C_STATE_READY_OBD_1	07 (Hex) / 0000 0111 (Binary)
C_STATE_READY_OBD_2	25 (Hex) / 0010 0101 (Binary)

In this example : misfire, fuel system, comprehensive component, catalyst, evaporative system and oxygen sensor monitoring are supported.

27.33.3.10 Rate-Based Monitoring

For the for Rate-Based Monitoring statistics calculation, CARB has defined standardized vehicle operations (RBM Driving Cycle), which must be met to increment counters.

A special driving cycle (LV_DC_RBM) for Rate-Based Monitoring is defined, which is set, if the following standardized vehicle operations are fulfilled:

- Cumulated trip length (time since engine start) T_AST_RBM ≥ C_T_AST_RBM

  (CARB : 600 s)
- Cumulative vehicle operation ≥ C_VS_THD_RBM for minimum of C_T_VS_RBM

  (CARB : VS ≥ 40 km/h (25mph) for at least 300 s)
- Continuous vehicle operation in idle equal or longer than C_T_IS_RBM (accelerator pedal released and vehicle speed less or equal than 1 mph)

  (CARB : 30 s)

with

- altitude such as ambient pressure AMP > C_AMP_MIN_RBM

  (CARB : 8000 feet)
- ambient temperature (TAM) ≥ C_TAM_MIN_RBM

  (CARB : 20 °F)

CARB requested values are :

C_DLY_AST_RBM = **2 s**
C_T_AST_RBM = **600 s**
C_VS_THD_RBM = **40 km/h**

C_T_VS_RBM = **300 s**
C_T_IS_RBM = **30 s**

C_VS_THD_IS_CDN_RBM = **1,6 km/h**
C_PV_IS_RBM = 0 %
C_AMP_MIN_RBM equivalent to 8000 feet ~ **2400 m**
C_TAM_MIN_RBM = **20 °F** = **- 7° C**

Calibration proposed :

C_DLY_AST_RBM = 2 s
C_T_AST_RBM = 600 s
C_VS_THD_RBM = 40,00 km/h
C_T_VS_RBM = 300 s
C_T_IS_RBM = 30 s

C_VS_THD_IS_CDN_RBM = 2 km/h (VS resolution : 1 km/h)
C_PV_IS_RBM = 0,00 %
C_AMP_MIN_RBM = 749,993 hPa
C_TAM_MIN_RBM = - 7,5 °C

28 Evaporative system control

Table of Contents

Index

28.1 EVAC Configuration Data

Input data:

NC_NR_CP_LAM_AD	NC_NR_CP_BUF	NC_PSD_DLY_CPS	

General information:

The following document describes the general rules for definition of the configuation data for EVAC agregate.

28.1.1 Local configuration data

Data	Value
NC_NR_CP_LAM_AD	9
NC_NR_CP_BUF	80
NC_PSD_DLY_CPS	2

NC_NR_CP_LAM_AD: (default value = 9)

Number of fixed cp / Lambda adaptation slices after engine start. This value is used in the time control between purge and lambda adaptation (903P) to have maximum of flexibility.

NC_NR_CP_BUF: (default value = 20)

size of the fuel buffer fuel_flow_add_buf. The default value is 20 but some applications which have long engine displacement need a greater value to simulate the distance between canister purge valve and injectors (80 for example).

NC_PSD_DLY_CPS: (default value = 2)

Canister purge valve diagnosis: Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met.

28.2 EVAC - Requirements for infrastructure

Input data:

Export actions:

ACTION_INFR_GetElDiagCPS(OUT<Cdn_diag_el_cps>, OUT<Err_diag_el_cps>)
This action reads the failure and the conditions information for each symptom from the infrastructure
ACTION_INFR_SetPWMDucyCPS(IN<Cppwm_cps>)
Sets the duty cycle of the PWM output signal
ACTION_INFR_SetPWMFrqCPS(IN<Frq_req_cp>)
Sets the frequency of the PWM output signal

Decription for actions:

ACTION_INFR_GetElDiagCPS(Cdn_diag_el_cps, Err_diag_el_cps)					
This action reads the diagnosis information and the failure symptoms (raw value) received from the infrastructure					
Name	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
Cdn_diag_el_cps	OUT	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of CPS; bit 0: diagnosis condition for symptom SYM_0; bit 1: diagnosis condition for symptom SYM_1; bit 2: diagnosis condition for symptom SYM_2					
Err_diag_el_cps	OUT	0H 1H 2H 4H	NO_SYM SYM_0 SYM_1 SYM_2	0	[-]
Raw value of error symptom for CPS (only parameter)					

ACTION_INFR_SetPWMDucyCPS(Cppwm_cps)					
By calling this action the duty cycle of the PWM output signal is changed					
Name	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
Cppwm_cps	IN	0...FFH	0...99.60937	0.390625	[%]
Corrected PWM signal for CPS opening					

ACTION_INFR_SetPWMFrqCPS(Frq_req_cp)					
By calling this action the frequency of the PWM output signal is changed					
Name	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
Frq_req_cp	IN	0...FFH	0...255	1	[Hz]
CPS Output frequency					

FUNCTION DESCRIPTION:

General information:

This module describes the interfaces to the basic software (BSW) and the PWM output opening of the canister purge valve.

Requirements for ACTION_INFR_GetElDiagCPS:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cdn_diag_el_cps	-	-	<bit coded>	Err_diag_el_cps	Diagnosis condition for each symptom **bit 0:** diagnosis condition for symptom SCP (SYM_0) **bit 1:** diagnosis condition for symptom SCG (SYM_1) **bit 2:** diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
Err_diag_el_cps	-	-	<bit coded>	Cdn_diag_el_cps	Bitcoded result of each symptom (SYM_0=SCP, SYM_1=SCG, SYM_2=OC) **0h** = NO_SYM **1h** = SYM_0 **2h** = SYM_1 **4h** = SYM_2 The relevant bit is set, if the error has been detected.

Diagnosis: ACTION_INFR_GetElDiagCPS() returns the electric diagnosis

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_SetPWMDucyCPS:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cppwm_cps	-	-	0.390625%	-	Output valve opening. Updating of duty cycle: 10ms

Diagnosis: no diagnosis

28.2.1 Coincidence requirements: -

Requirements for ACTION_INFR_SetPWMFrqCPS:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Frq_req_cp	-	-	1Hz	-	FRQ_REQ_CP is generally between 8 to 16 Hz.

Diagnosis: no diagnosis

28.2.2 Coincidence requirements: -

28.3 AGGR EVAC adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CLOSE_ACT_CP	V/O	0...1H	0...1	1	[-]
Logical value for CPS valve closed					

Input data:

LV_CP_CLOSE_ACT			

FUNCTION DESCRIPTION:

Adaptation to aggregate environment.

Application conditions:

Initialisation: at reset:
LV_CLOSE_ACT_CP = LV_CP_CLOSE_ACT from NVMY

Recurrence: 20ms

Activation: at every engine state

Deactivation: -

Formula section:

LV_CLOSE_ACT_CP = LV_CP_CLOSE_ACT

28.4 Evaporative emission control variables

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CDN_CP	V/O	0H 1H 2H 3H 4H 5H	NO_CDN CDN_NO_PURGE CDN_MIN_PURGE CDN_RAMP_OPEN CDN_RAMP_FAST CDN_NO_FAST	1	[-]
Activation conditions for evaporative emission control function					
LV_N_MAX_CP	V/O	0...1H	0...1	1	[-]
Maximum engine speed for canister purge calculations limit reached					
LV_CP_CLOSE_REQ	V/O	0...1H	0...1	1	[-]
Request to close the CPS valve					
LV_CP_WIN	V	0...1H	0...1	1	[-]
Engine operation within CP window (n & maf)					
LV_CP_DYW	V/O	0...1H	0...1	1	[-]
Dynamic window for canister purge					
LV_CP_PU	V	0...1H	0...1	1	[-]
Flag to interupt the purge in PU mode					
LV_LAM_LIM_CP	V	0...1H	0...1	1	[-]
Lambda controller reached limits (defined in lambda controller module)					
LV_LAM_OUT_CP	V	0...1H	0...1	1	[-]
Lambda controller reached limits (defined in this module)					
LV_CP_CMU_CMB	V/O	0...1H	0...1	1	[-]
Commutation combustion active					
LV_CP_CDN_MIN_PURGE_FAST	V/O	0...1H	0...1	1	[-]
flag for usage of CL-model; =1: MIN_PURGE forced, no CL-mdl usage possible					
PRS_CPS	V/O	8000...7FFFH	-2717.04... 2716.96	0.0829175	[hPa]
Pressure difference envirionment over CP line					
N_DYW_CP	V	0...1FE0H	0...8160	1	[rpm]
N average value for dynamic window					
MAF_DYW_CP	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
MAF average value for dynamic window					
TQI_DYW_CP	V	0...7FFFH	0...1023.97	0.03125	[Nm]
TQI_SP_S average value for dynamic window					
LAMB_SP_S_DYW_CP	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
LAMB_SP_S average value for dynamic window					
FAC_LAM_CP_DYW	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
FAC_LAM_CP average value for dynamic window					
T_DLY_DYW_CP	-	0...FFFFH	0...1310.7	0.02	[s]
timer for LV_CP_DYW calculation					
T_DLY_CP_PU_TMP	-	0...FFFFH	0...6553.5	0.1	[s]
timer for LV_CP_PU calculation					
T_DLY_LAM_LIM_CP_TMP	-	0...FFFFH	0...6553.5	0.1	[s]
timer for LV_LAM_LIM_CP calculation					
T_DLY_LAM_OUT_CP_TMP	-	0...FFFFH	0...6553.5	0.1	[s]
timer for LV_LAM_OUT_CP calculation					
T_DLY_CMU_CP_TMP	-	0...FFFFH	0...6553.5	0.1	[s]
timer for LV_CP_CMU_CMB calculation					
T_DLY_CP_PUC_TMP	-	0...FFFFH	0...6553.5	0.1	[s]
timer for LV_CP_PUC calculation					
T_DLY_CAT_PURGE_CP_TMP	-	0...FFFFH	0...6553.5	0.1	[s]

timer for LV_CP_CAT_PURGE calculation					
T_DLY_CAT_PURGE_CP_AFL_TMP	-	0...FFFFH	0...6553.5	0.1	[s]
timer for LV_CP_CAT_PURGE_AFL calculation					
LV_CP_PUC	V	0...1H	0...1	1	[-]
CP interruption due to engine state PUC					
LV_CP_CAT_PURGE	O/V	0...1H	0...1	1	[-]
Catalyst purge acivated					
LV_CP_CAT_PURGE_AFL	O/V	0...1H	0...1	1	[-]
Catalyst purge AFL acivated					

Input data:

N_32	MAF	TCO	TIA_THR
AMP	MAP	LV_IS	LV_PU
N	LV_LAM_LSCL[NC_CBK_EX_NR]	LV_LAM_STOP[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MIN[NC_CBK_EX_NR]
LV_FAC_LAM_LIM_MAX[NC_CBK_EX_NR]	LV_LDC_LAM_AD[NC_CBK_EX_NR]	FAC_LAM_MV_MMV_CP[NC_CBK_EX_NR]	
LV_NT_RGN_REQ	LV_CMB_TRAN_ACT	LAMB_SP_S	LV_FCUT_IND
NC_CBK_EX_NR	TQI_SP_S	T_AST	LV_HOM_ACT
FAC_LAM_CP	LV_CP_DYW_EXT	STATE_LOCK_CP	LV_INH_CP
LV_CP_CLL	LV_CP_CLOSE_2	LV_INH_CP_FAST	LV_INH_CP_CLL
STATE_CLL_DEAC_CP	PRS_CPS_EXT	LV_CP_CLOSE_1	LV_CP_CLOSE_3
LV_CAT_PURGE_REQ_POST_AFL[NC_NT_NR]	LV_PUC	STATE_CP	T_DLY_MAX_CP
LV_LAM_STOP_SHO_PER[NC_CBK_EX_NR]	LV_CL_MDL_END_CAT_PURGE		

Input data table order: External inputs – Inputs from other EVAC modules

(If the system includes an OBD sequencer, LV_CP_CLOSE_CP_x will be imported from outside the EVAC aggregate, if the system does not include an OBD sequencer, the flags will be provide within the aggregate (within the appl.inc. module))

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TIA_MIN_CP	1	0...FEH	-48...142.5	0.75	[°C]
Minimum intake air temperature for closing of the CPS					
C_N_MAX_CP	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed value for canister purge calculations (CPU load reduction)					
C_N_HYS_MAX_CP	1	0...FFH	0...8160	32	[rpm]
Hysteresis value for canister purge calculations (CPU load reduction)					
C_N_CP_BOL	1	0...FFH	0...8160	32	[rpm]
Minimum speed limit for activation of the RAMP_OPEN operation (homogeneous)					
C_N_CP_TOL	1	0...FFH	0...8160	32	[rpm]
Maximum speed limit for activation of the RAMP_OPEN operation (homogeneous)					
C_N_2_CP_BOL	1	0...FFH	0...8160	32	[rpm]
Minimum speed limit for activation of the RAMP_OPEN operation (stratified)					
C_N_2_CP_TOL	1	0...FFH	0...8160	32	[rpm]
Maximum speed limit for activation of the RAMP_OPEN operation (stratified)					
C_FAC_LAM_CP_BOL	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Limit on FAC_LAM_CP value to start RAMP_OPEN					
C_FAC_LAM_CP_TOL	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Limit on FAC_LAM_CP value to start RAMP_OPEN					
C_LAMB_SP_S_CP_BOL	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Bottom limit on the LAMB_SP_S value					
C_LAMB_SP_S_CP_TOL	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Top limit on the LAMB_SP_S value					
C_PRS_MAX_CPS	1	8000...7FFFH	-2717.04... 2716.96	0.0829175	[hPa]
Pressure difference limit (environment-intake manifold) for activation of the canister purge					
C_PRS_MAX_CPS_2	1	8000...7FFFH	-2717.04... 2716.96	0.0829175	[hPa]
Pressure difference limit (environment-intake manifold) for direct switch to NO_PURGE (no ramp)					
C_DLY_N_MAF_DYW_CP	1	0...FFFFH	0...1310.7	0.02	[s]
Time delay to have stabilization condition					
C_T_DLY_PU_CPS	1	1...FFFFH	0.1...6553.5	0.1	[s]
CPS interrupt time during PU					
C_T_LAM_LIM_CP	1	0...FFFFH	0...6553.5	0.1	[s]
Additionnal time to stay in MIN_PURGE after LV_FAC_LAM_LIM_XXX					
C_T_DLY_LAM_CP	1	1...FFFFH	1...65535	1	[s]
Delay on lambda control stabilization to restart RAMP_OPEN					
IP_T_DLY_CMU_CPS	6	0...FFFFH	0...6553.5	0.1	[s]
LDP_N_32_IP_T_DLY_CMU_CPS	6	0...FFH	0...8160	32	[rpm]
Stabilization delay on lambda regulation after commutation					
C_MAF_DYW_CP	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
MAF window for limited dynamics					
C_N_DYW_CP	1	0...1FE0H	0...8160	1	[rpm]
engine speed window for limited dynamics					
C_N_2_DYW_CP	1	0...1FE0H	0...8160	1	[rpm]
engine speed window for limited dynamics					
C_TQI_DYW_CP	1	0...7FFFH	0...1023.97	0.03125	[Nm]
Dynamic window value on TQI_SP_S					
C_FAC_LAM_CP_DYW	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Dynamic window value on FAC_LAM_CP					
C_LAMB_SP_S_DYW_CP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Dynamic window on LAMB_SP_S value					
C_CRLC_N_DYW_CP	1	0...FFH	0...0.99609	3.9063e-3	[-]
Filter constant for limited dynamics calculation on engine speed					
C_CRLC_MAF_DYW_CP	1	0...FFH	0...0.99609	3.9063e-3	[-]
Filter constant for limited dynamics calculation on MAF					
C_CRLC_LAMB_SP_S_DYW_CP	1	0...FFH	0...0.99609	3.9063e-3	[-]

Filter constant for limited dynamics calculation on lambda setpoint in stratified					
C_CRLC_LAM_DYW_CP	1	0...FFH	0...0.99609	3.9063e-3	[-]
Filter constant for limited dynamics calculation on lambda controller controller in non-normal purge					
IP_MAF_CP_BOL	8	0...FFH	0...1389	5.4470588	[mg/stk]
LDPM_N_32_3_EVAC	8	0...FFH	0...8160	32	[rpm]
Minimum mass air flow limit for activation of the RAMP_OPEN operation (homogeneous)					
IP_MAF_CP_TOL	8	0...FFH	0...1389	5.4470588	[mg/stk]
LDPM_N_32_3_EVAC	8	0...FFH	0...8160	32	[rpm]
Maximum mass air flow limit for activation of the RAMP_OPEN operation (homogeneous)					
C_MAF_CP_BOL_HYS	1	0...FFH	0...1389	5.4470588	[mg/stk]
Hysteresis for minimum mass air flow limit for activation of the RAMP_OPEN operation (homogeneous					
IP_TQI_CP_BOL	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32_3_EVAC	8	0...FFH	0...8160	32	[rpm]
Minimum torque limit for sctivation of the RAMP_OPEN operation (stratified)					
IP_TQI_CP_TOL	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32_3_EVAC	8	0...FFH	0...8160	32	[rpm]
Maximum torque limit for sctivation of the RAMP_OPEN operation (stratified)					
C_TQI_CP_BOL_HYS	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Hysteresis for minimum mass air flow limit for activation of the RAMP_OPEN operation (homogeneous					
C_T_DLY_PUC_CP	1	1...FFFFH	0.1...6553.5	0.1	[s]
caniser purge interrupt time after PUC					
C_T_DLY_CAT_PURGE_CP	1	1...FFFFH	0.1...6553.5	0.1	[s]
Delay time after cathalyst purge					
C_T_DLY_CAT_PURGE_AFL_CP	1	1...FFFFH	0.1...6553.5	0.1	[s]
Delay time at strart cathalyst purge AFL					
C_DLY_N_TQI_DYW_CP_1	1	0...FFFFH	0...1310.7	0.02	[s]
Time delay to have stabilization condition for RampOpen					
C_DLY_N_TQI_DYW_CP_2	1	0...FFFFH	0...1310.7	0.02	[s]
Time delay to have stabilization condition for RampOpenFast and MaxPurge					
C_CRLC_TQI_DYW_CP_1	1	0...FFH	0...0.99609	3.9063e-3	[-]
Filter constant for limited dynamics calculation on tqi_sp in stratified					
C_CRLC_TQI_DYW_CP_2	1	0...FFH	0...0.99609	3.9063e-3	[-]
Filter constant for limited dynamics calculation on tqi_sp in stratified for RampOpenFast and MaxPurge					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_CP_CLL_S_INH	-	0...1H	0...1	1	[-]
Selection of MIN_PURGE during stratified mode (=1)					
LC_CP_CLL_INH_MAN	-	0...1H	0...1	1	[-]
switch for deactivation of closed loop canister purge ("normal purge")					
LC_CP_INH_MAN	-	0...1H	0...1	1	[-]
switch for deactivation of canister purge and force NO_PURGE					
LC_CP_DYW_CLC_ENA	-	0...1H	0...1	1	[-]
1: calculation of dynamic window conditions					
LC_CP_DYW_LAM_AD	-	0...1H	0...1	1	[-]
1: use the limited dynamics flag of the lambda adaptation (common flag lamb ad / cp)					
LC_CP_DYW_CLC_MAX_ENA	-	0...1H	0...1	1	[-]
1: calculate lim.dynamics / lambda contr.limits etc. also with n_max_cp exceeded					
LC_LAM_DYW_CP_INI_DEAC	-	0...1H	0...1	1	[-]
1: deactivate dynamic window on lambda control at deactivated normal purge					
LC_STATE_CDN_CP_MAN	-	0...1H	0...1	1	[-]
switch to activate test mode // manual setting of STATE_CDN_CP					
LC_PRS_CPS_EXT	-	0...1H	0...1	1	[-]
switch to use externally calculated value for PRS_CPS					

C_STATE_WIN_CP_NOT	-	0H 1H 2H	NONE NO_PURGE MIN_PURGE	1	[-]
state to be forced if engine operation outside CP window					
C_STATE_CDN_CP_MAN	-	0H 1H 2H 3H 4H 5H	NO_CDN CDN_NO_PURGE CDN_MIN_PURGE CDN_RAMP_OPEN CDN_RAMP_FAST CDN_NO_FAST	1	[-]
manual selection of STATE_CDN_CP					
C_STATE_LAM_LIM_CP	-	0H 1H 2H	NONE NO_PURGE MIN_PURGE	1	[-]
state to be forced if lambda control limitation reached					
LC_CP_CAT_PURGE_OFF	1	0...1H	0...1	1	[-]
Canister purge off during catalyst purge					
LC_CP_CAT_PURGE_OPL	1	0...1H	0...1	1	[-]
Canister purge in open loop during cathalyst purge					
LC_CP_CAT_PURGE_AFL_ACT	1	0...1H	0...1	1	[-]
Canister purge AFL in closed loop with CL-Model					

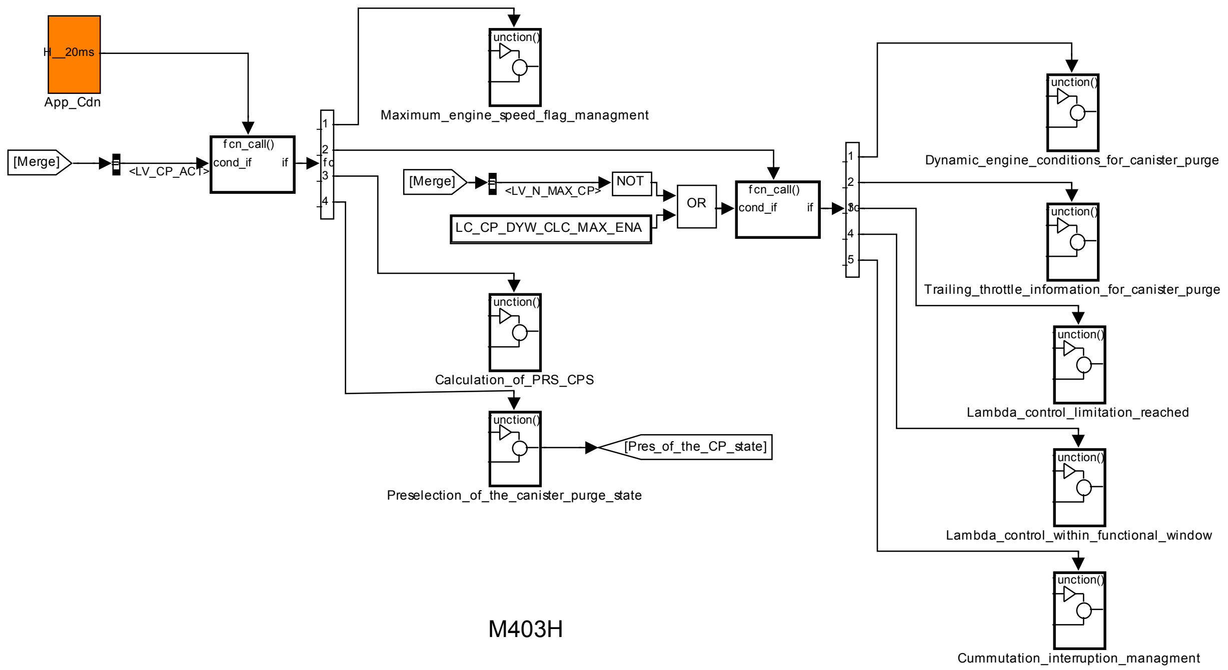
General information:

--

Application conditions:

Initialisation: --
Recurrence: **20** ms
Activation: --
Deactivation: --

Signal flow diagram:



Formula section:

```
if      LV_CP_ACT = 1
then call   “Maximum engine speed flag management”
        if     [ LV_N_MAX_CP = 0
        or     LC_CP_DYW_CLC_MAX_ENA = 1 ]
        then  call   “Dynamic engine conditions for canister purge”
              call   “Trailling throttle information for canister purge”
              call   "Canister purge prohibition after "trailing fuel cut off""
              call   “Lambda control limitation reached”
              call   “Lambda control within functional window”
              call   “Commutation interruption management”
              call   "Catalyst purge interruption management"
        endif
        call   ”Calculation of PRS_CPS”
        call   “Preselection of the canister purge state”
end
```

28.4.1 Maximum engine speed flag management

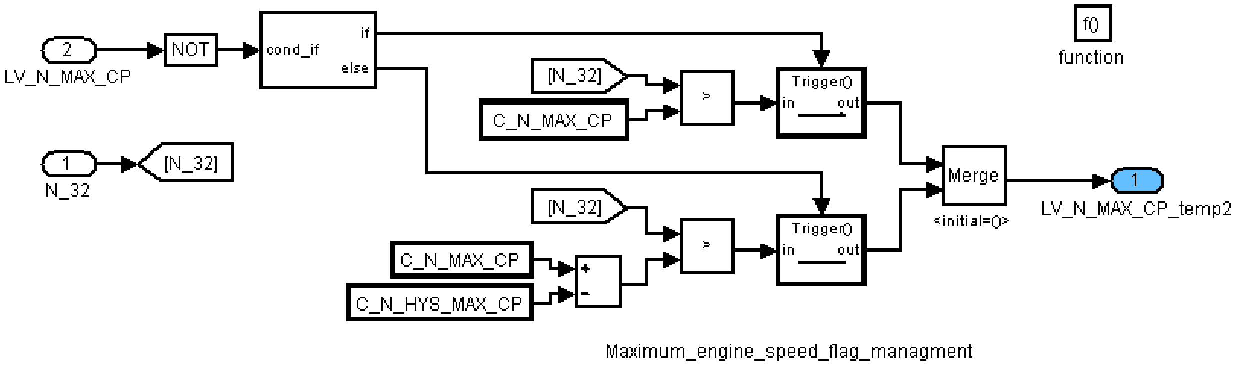
General information:

When the CPU load is critical it is possible to disable some of the canister purge calculations when the engine speed is above a threshold.

Application conditions:

Initialisation: at RST or IGKON
LV_N_MAX_CP = 0
Recurrence: called by superior block
Activation: called by superior block
Deactivation: --

Signal flow diagram:



Formula section:

if LV_N_MAX_CP = 0
then LV_N_MAX_CP = [N_32 > C_N_MAX_CP]
else LV_N_MAX_CP = [N_32 > C_N_MAX_CP – C_N_HYS_MAX_CP]
endif
end

28.4.2 Dynamic engine conditions for canister purge

General information:

The limited dynamic condition for a variable exist if the current value stays inside a tolerance window.
If the values of n & maf (homogeneous) rsp. n, tqi_sp_s and lamb_sp_s (stratified) stay within a window around the moving mean value of this variable for a certain time, limited dynamic is detected.

C_FAC_LAM_CP_DYW should be always greater than C_FAC_LAM_DIF_MAX_CP.

Application conditions:

Initialisation: at RST or IGKON
N_DYW_CP = 0
MAF_DYW_CP = 0
TQI_DYW_CP = 0
LAMB_SP_S_DYW_CP = 0
FAC_LAM_CP_DYW = 0
T_DLY_DYW_CP = 0

Recurrence: called by superior bloc
Activation: called by superior block
Deactivation: --

Formula section:

```
if      LC_CP_DYW_CLC_ENA     = 1
and     LC_CP_CLL_INH_MAN     = 0
then    call   "Dynamic window calculation"
        else   LV_CP_DYW = 1
endif
if      LC_CP_DYW_LAM_AD = 1         // use dyw flag from lambda adaptation
then    LV_CP_DYW = [LV_CP_DYW and LV_LDC_LAM_AD [1] and LV_LDC_LAM_AD [2]]
endif
end
```

28.4.2.1 Dynamic window calculation

```
if      [      LV_HOM_ACT = 1              // limited dynamic cond. for homogeneous operation
        and    abs  (N_DYW_CP      - N)            < C_N_DYW_CP
        and    abs  (MAF_DYW_CP  - MAF)                 < C_MAF_DYW_CP
        and    abs  (FAC_LAM_CP_DYW  - FAC_LAM_CP)       < C_FAC_LAM_ CP_DYW]
or      [      LV_HOM_ACT = 0              // limited dynamic conditions for stratified operation
        and    abs  (N_DYW_CP      - N)            < C_N_2_DYW_CP
        and    abs  (TQI_DYW_CP   - TQI_SP_S)           < C_TQI_DYW_CP
        and    abs  (LAMB_SP_S_DYW_CP - LAMB_SP_S)       < C_LAMB_SP_S_DYW_CP ]
then    call   "Timer and MMV calculation"        // LDC fulfilled
else    call   "Timer and MMV init"               // LDC not fulfilled
endif
LV_CP_DYW =
    [ T_DLY_DYW_CP  = 0
and     LV_CP_DYW_EXT= 1 ]                        // external dyn.win.flag from appl.inc. module
end
```

28.4.2.2 Timer and MMV calculation

```
N_DYW_CP$_{n}$ = N_DYW_CP$_{n-1}$ + (N - N_DYW_CP$_{n-1}$) * C_CRLC_N_DYW_CP
if LV_HOM_ACT = 1
        MAF_DYW_CP$_{n}$  = MAF_DYW_CP$_{n-1}$ +
        (MAF - MAF_DYW_CP$_{n-1}$) * C_CRLC_MAF_DYW_CP
        if   LV_CP_CLL  = 1               // no ldc on lambda controller in normal purge
        or [STATE_CLL_DEAC_CP ≠ CP_NO_DEAC and LC_LAM_DYW_CP_INI_DEAC = 1]
        then  FAC_LAM_ CP_DYW          = FAC_LAM_CP
        else  FAC_LAM_CP_DYW$_{n}$          = FAC_LAM_CP_DYW$_{n-1}$ +
              (FAC_LAM_CP – FAC_LAM_CP_DYW$_{n-1}$) * C_CRLC_LAM_DYW_CP
        endif
else
    if (STATE_CP = RAMP_OPEN_FAST or STATE_CP = MAX_PURGE
    or STATE_CLL_DEAC_CP = MAX_PURGE_DEAC
    or STATE_CLL_DEAC_CP = RAMP_FAST_DEAC
```

```
          or T_DLY_MAX_CP > 0 )
        then  TQI_DYW_CP$_n$  = TQI_DYW_CP$_{n-1}$ +
              (TQI_SP_S - TQI_DYW_CP$_{n-1}$) * C_CRLC_TQI_DYW_CP_2
        else  TQI_DYW_CP$_n$  = TQI_DYW_CP$_{n-1}$ +
              (TQI_SP_S - TQI_DYW_CP$_{n-1}$) * C_CRLC_TQI_DYW_CP_1
        endif
        LAMB_SP_S_DYW_CP$_n$ = LAMB_SP_S_DYW_CP$_{n-1}$   +
              (LAMB_SP_S - LAMB_SP_S_DYW_CP$_{n-1}$) * C_CRLC_LAMB_SP_S_DYW_CP
    endif
    T_DLY_DYW_CP = max (0, T_DLY_DYW_CP – TA)   // timer decrementation
    end
```

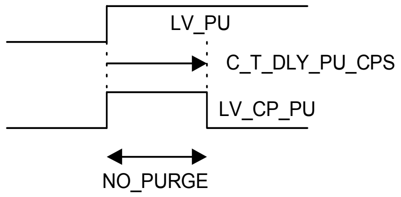
28.4.2.3 Timer and MMV init

```
    N_DYW_CP = N
    if      LV_HOM_ACT = 1
            MAF_DYW_CP          = MAF
            FAC_LAM_CP_DYW      = FAC_LAM_CP
            T_DLY_DYW_CP        = C_DLY_N_MAF_DYW_CP
    else
            TQI_DYW_CP                  = TQI_SP_S
            LAMB_SP_S_DYW_CP            = LAMB_SP_S
            if (STATE_CP = RAMP_OPEN_FAST or STATE_CP = MAX_PURGE
                  or STATE_CLL_DEAC_CP = MAX_PURGE_DEAC
                  or STATE_CLL_DEAC_CP = RAMP_FAST_DEAC
                  or T_DLY_MAX_CP > 0 )
            then  T_DLY_DYW_CP          = C_DLY_N_TQI_DYW_CP_2
            else  T_DLY_DYW_CP          = C_DLY_N_TQI_DYW_CP_1
            endif
    endif
    end
```

28.4.3 Trailling throttle information for canister purge

General information:

In case of trailling throttle condition LV_PU, the fuel coming from the purge could be important compared to the fuel injected. To prevent any malfunction we need to close immediatly the purge valve during a delay C_T_DLY_PU_CPS. During this time the flag LV_CP_PU is set to 1.



Application conditions:

Initialisation: at RST or IGKON
LV_CP_PU = 0
T_DLY_CP_PU_TMP = 0
Recurrence: called by superior block
Activation: called by superior block
Deactivation: --

Formula section:

```
if     LV_PU  0 → 1
then   T_DLY_CP_PU_TMP = C_T_DLY_PU_CPS
else   T_DLY_CP_PU_TMP = max (0, T_DLY_CP_PU_TMP – TA)
endif
LV_CP_PU = [T_DLY_CP_PU_TMP > 0 ]
end
```

28.4.4 Canister purge prohibition after "trailing fuel cut off"

After a fuel cut off phase (PUC) the engine needs a ceratain time to stabilize the air fuel ratio. During this phase the lambda controller has a deviation. To prevent that this deviation has an influence on the CL calculation, canister purge will be disabled after a PUC phase.

Application conditions:

Initialisation: at RST or IGKON
LV_CP_PUC = 0
T_DLY_CP_PUC_TMP = 0
Recurrence: called by superior block
Activation: called by superior block
Deactivation: --

Formula section:

```
if     LV_PUC  = 1
then   T_DLY_CP_PUC_TMP = C_T_DLY_PUC_CP
else   T_DLY_CP_PUC_TMP = max (0, T_DLY_CP_PUC_TMP – TA)
endif
LV_CP_PUC = [T_DLY_CP_PUC_TMP > 0 ]
end
```

28.4.5 Lambda control limitation reached

General information:

When the lambda control reaches a minimum or maximum value, the information LV_FAC_LAM_LIM_MIN rsp. LV_FAC_LAM_LIM_MAX is set. As soon as the lambda control value is again in its limits, the bit information is reset. In order to memorize this information and to wait for a lambda control value stabilized, an information is managed LV_LAM_LIM_CP.

This value is set to TRUE as long as any lambda control limit is exceeded and reset after a delay C_T_LAM_LIM_CP after the lambda control is within its limits again.

Application conditions:

Initialisation: at RST or IGKON
LV_LAM_LIM_CP = 0
T_DLY_LAM_LIM_CP_TMP = 0
Recurrence: called by superior block
Activation: called by superior block
Deactivation: --

Formula section:

```
if   LV_FAC_LAM_LIM_MIN [i]  = 1          at least one bench
or   LV_FAC_LAM_LIM_MAX [i]  = 1          at least one bench
then T_DLY_LAM_LIM_CP_TMP = C_T_LAM_LIM_CP
else T_DLY_LAM_LIM_CP_TMP = max (0, T_DLY_LAM_LIM_CP_TMP – TA)
endif
LV_LAM_LIM_CP = [T_DLY_LAM_LIM_CP_TMP > 0 ]
end
```

28.4.6 Lambda control within functional window

General information:

Parallel to the observation of lamba control limits using the flags LV_FAC_LAM_LIM_MIN and LV_FAC_LAM_LIM_MAX coming from the lambda controller, the canister purge strategy provides two independent limits C_FAC_LAM_CP_BOL & C_FAC_LAM_CP_TOL.
As long as at least one lambda controller is not running in a window defined by [C_FAC_LAM_CP_BOL, C_FAC_LAM_CP_TOL] an information is set to TRUE and reset after a delay C_T_DLY_LAM_CP after all lambda controllers are within [C_FAC_LAM_CP_BOL, C_FAC_LAM_CP_TOL] again.

Application conditions:

Initialisation: at RST or IGKON
LV_LAM_OUT_CP = 0
T_DLY_LAM_OUT_CP_TMP = 0
Recurrence: called by superior block
Activation: called by superior block
Deactivation: --

Formula section:

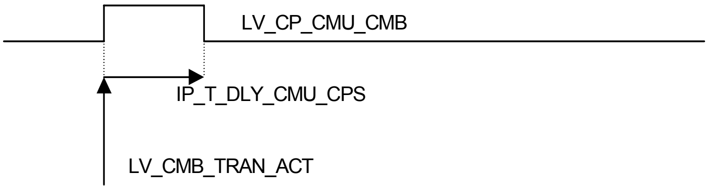
```
if   FAC_LAM_MV_MMV_CP [i] <= C_FAC_LAM_CP_BOL          at least one bench
or   FAC_LAM_MV_MMV_CP [i] >= C_FAC_LAM_CP_TOL          at least one bench
then T_DLY_LAM_OUT_CP_TMP = C_T_DLY_LAM_CP
else T_DLY_LAM_OUT_CP_TMP = max (0, T_DLY_LAM_OUT_CP_TMP – TA)
endif
```

```
LV_LAM_OUT_CP = [T_DLY_LAM_OUT_CP_TMP > 0 ]
end
```

28.4.7 Commutation interruption management

General information:

Before a combustion commutation, the CPS valve may be closed to purge the intake manifold. A delay will be added to wait for lambda stabilisation and manifold purge. During the commutation the evaporative emission control is in state NO_PURGE



Application conditions:

Initialisation: at RST or IGKON
LV_CP_CMU_CMB = 0
T_DLY_CMU_CP_TMP = 0
Recurrence: called by superior block
Activation: called by superior block
Deactivation: --

Formula section:

```
if    LV_CMB_TRAN_ACT
then  T_DLY_CMU_CP_TMP = IP_T_DLY_CMU_CPS (N_32)
else  T_DLY_CMU_CP_TMP = max (0, T_DLY_CMU_CP_TMP – TA)
endif
LV_CP_CMU_CMB = [T_DLY_CMU_CP_TMP > 0 ]
end
```

28.4.8 Catalyst purge interruption management

General information:

During NO_x regeneration of the catalyst and a time delay after that, the canister purge can be set configurable to off, open loop or closed loop.

Application conditions:

Initialisation: at RST or IGKON
LV_CP_CAT_PURGE = 0
T_DLY_CAT_PURGE_CP_TMP = 0
Recurrence: called by superior block
Activation: called by superior block
Deactivation: --

Formula section:

```
if    LV_HOM_ACT = 1
      and  [LV_NT_RGN_REQ
      or   (LV_CAT_PURGE_REQ_POST_AFL[i]          // at least one of these flags
           and LC_CP_CAT_PURGE_AFL_ACT)]
then  T_DLY_CAT_PURGE_CP_TMP = C_T_DLY_CAT_PURGE_CP
else  T_DLY_CAT_PURGE_CP_TMP =  max (0, T_DLY_CAT_PURGE_CP_TMP – TA)
endif
LV_CP_CAT_PURGE = [T_DLY_CAT_PURGE_CP_TMP > 0 ]
end

if    [LV_HOM_ACT = 1
      and  LV_CAT_PURGE_REQ_POST_AFL[i] transition 0 -> 1 // at least one of these
flags
then  T_DLY_CAT_PURGE_CP_AFL_TMP = C_T_DLY_CAT_PURGE_AFL_CP
else  T_DLY_CAT_PURGE_CP_AFL_TMP
                             = max (0, T_DLY_CAT_PURGE_CP_AFL_TMP – TA)
endif
LV_CP_CAT_PURGE_AFL = [T_DLY_CAT_PURGE_CP_AFL_TMP > 0 ]
end
```

28.4.9 Calculation of PRS_CPS

General information:

PRS_CPS is the pressure drop over the CP system, i.e. the difference between ambient air pressure and intake manifold pressure

Application conditions:

Initialisation: at RST or IGKON
PRS_CPS = 0
Recurrence: called by superior block
Activation: called by superior block
Deactivation: --

Formula section:

```
if    LC_PRS_CPS_EXT
then  PRS_CPS  = PRS_CPS_EXT
else  PRS_CPS  = AMP – MAP
```

end

28.4.10 Preselection of the canister purge state

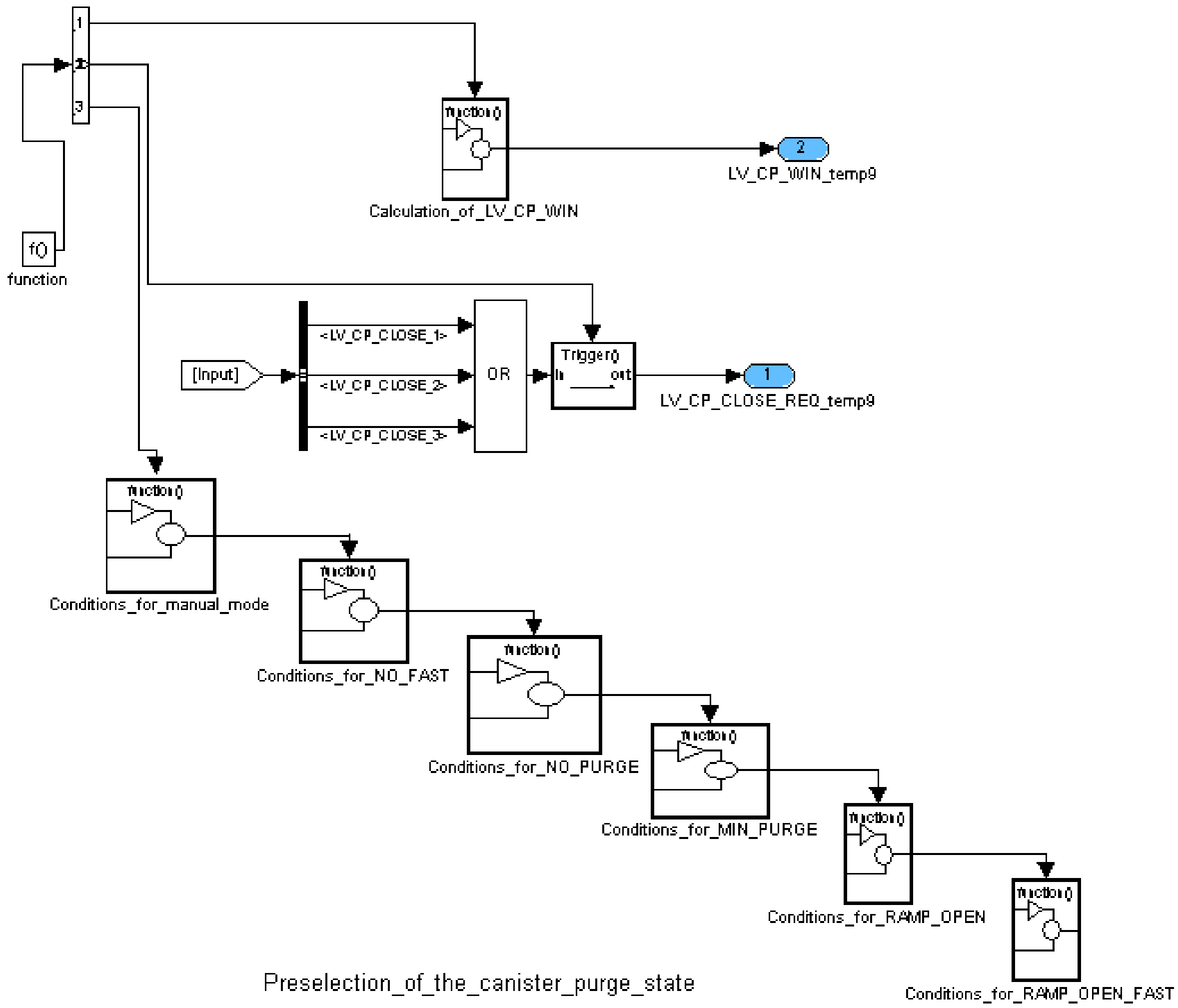
General information:

After activation of the function (LV_CP_ACT=1) there is a continuous check of the activation conditions.

Application conditions:

Initialisation:	at RST or IGKON all the values are set to zero.
Recurrence:	called by superior block
Activation:	called by superior block
Deactivation:	--

Signal flow diagram:



Formula section:

```
if      LV_HOM_ACT = 1
then  if LV_CP_WIN = 1
        then LV_CP_WIN =                          // Engine operation window for CP, hom.
               [      N_32        > C_N_CP_BOL
               and   N_32        < C_N_CP_TOL
               and   MAF         > [ IP_MAF_CP_BOL (N_32) - C_MAF_CP_BOL_HYS ]
               and   MAF         < IP_MAF_CP_TOL (N_32)]
        else LV_CP_WIN =                          // Engine operation window for CP, hom.
               [      N_32        > C_N_CP_BOL
               and   N_32        < C_N_CP_TOL
               and   MAF         > [ IP_MAF_CP_BOL (N_32) + C_MAF_CP_BOL_HYS ]
               and   MAF         < IP_MAF_CP_TOL (N_32)]
        endif
else  if LV_CP_WIN = 1
        then LV_CP_WIN =                          // Engine operation window for CP, strat.
               [      N_32        > C_N_2_CP_BOL
               and   N_32        < C_N_2_CP_TOL
               and   TQI_SP_S > [ IP_TQI_CP_BOL (N_32) - C_TQI_CP_BOL_HYS ]
               and   TQI_SP_S < IP_TQI_CP_TOL (N_32)]
        else  LV_CP_WIN =                         // Engine operation window for CP, strat.
               [      N_32        > C_N_2_CP_BOL
               and   N_32        < C_N_2_CP_TOL
               and   TQI_SP_S > [ IP_TQI_CP_BOL (N_32) + C_TQI_CP_BOL_HYS ]
               and   TQI_SP_S < IP_TQI_CP_TOL (N_32)]
        endif
endif

LV_CP_CLOSE_REQ =
[       LV_CP_CLOSE_1                // request to close CPS with high speed
or      LV_CP_CLOSE_2                // request to close CPS with medium speed
or      LV_CP_CLOSE_3 ]              // request to close CPS with normal speed
```

Conditions for manual mode:

1. Test mode activated
   (LC_STATE_CDN_CP_MAN = 1)
   // in this case the conditions below are no more checked

➔ STATE_CDN_CP = C_STATE_CDN_CP_MAN

Conditions for immediate switch to NO_PURGE (without ramp):

1. Fuel cut off active
   (LV_FCUT_IND = 1) **or**
2. Trailing throttle operation started
   (LV_CP_PU = 1) **or**
3. Prohibition after PUC
   (LV_CP_PUC = 1) **or**
4. Deactivation due to high engine speed / CPU load
   (LV_N_MAX_CP = 1) **or**

5. Difference pressure intake manifold-environment below threshold
   (relevant for supercharger engine or high engine load)
   (PRS_CPS < C_PRS_MAX_CPS_2) **or**
6. Request for fast closing from appl.inc. module
   (LV_INH_CP_FAST = 1)

➔ STATE_CDN_CP = CDN_NO_FAST

Conditions for NO_PURGE:
(closing of the CPS required)

1. Conditions for NO_FAST are not fulfilled **and**
2. {canister purge manually forced to NO_PURGE
   (LC_CP_INH_MAN=1) **or**
3. Application incidences flag to inhibit canister purge and switch to NO_PURGE
   (LV_INH_CP=1) **or**
4. NO_PURGE locked for a certain time by EVAC control
   (STATE_LOCK_CP = NO_PURGE) **or**
5. Request to close CPS from OBD2 sequenceur
   (LV_CP_CLOSE_REQ=1) **or**
6. Difference pressure intake manifold-environment below threshold
   (relevant for supercharger engine)
   (PRS_CPS < C_PRS_MAX_CPS) **or**
7. Idle speed active and ambient air temperature below threshold
   (acoustic problems with the CPS at low ambient temperatures)
   [(LV_IS=1) **and** (TIA_THR < C_TIA_MIN_CP)] **or**
8. Combustion commutation and stabilization
   (LV_CP_CMU_CMB=1) **or**
9. Engine operation point outside CP window and NO_PURGE selected for this event
   [(LV_CP_WIN = 0) **and** (C_STATE_WIN_CP_NOT = NO_PURGE)] **or**
10. Lambda controller reached limits and no purge state is desired
    [(LV_LAM_LIM_CP = 1) **and** (C_STATE_LAM_LIM_CP = NO_PURGE)] **or**
11. Catalyst purge requests state NoPurge
    [(LV_CP_CAT_PURGE = 1
    and (LC_CP_CAT_PURGE_OFF = 1 or LV_CL_MDL_END_CAT_PURGE = 1)] **or**
    [LV_CP_CAT_PURGE_AFL = 1]

➔ STATE_CDN_CP = CDN_NO_PURGE

Conditions for MIN_PURGE:

1) "hard" conditions for MIN_PURGE:

The "hard" condtitions for MIN_PURGE force MIN_PURGE, i.e. no usage of the canister load model is possible.

LV_CP_CDN_MIN_PURGE_FAST =

1. {closed loop canister purge (“normal puge”) manually deactivated
   (LC_CP_CLL_INH_MAN=1) **or**
2. Application incidences flag to inhibit normal purge and switch to MIN_PURGE
   (LV_INH_CP_CLL=1)
   or
3. MIN_PURGE locked for a certain time by EVAC control

(STATE_LOCK_CP = MIN_PURGE) **or**

4. Trailing throttle
   (LV_PU = 1) **or**
5. Switch for not allowing NORMAL_PURGE in stratified set
   (LV_HOM_ACT=0 **and** LC_CP_CLL_S_INH=1) **or**
6. Engine operation point outside CP window and MIN_PURGE selected for this event
   [(LV_CP_WIN = 0) **and** (C_STATE_WIN_CP_NOT = MIN_PURGE)] } **or**
7. Catalyst purge requests open loop
   [(LV_CP_CAT_PURGE = 1) and (LC_CP_CAT_PURGE_OPL = 1)]

2) combination of all conditions for MIN_PURGE:

1. Conditions for NO_FAST are not fulfilled **and**
2. Conditions for NO_PURGE are not fulfilled **and**
3. [Any of the "hard" conditions for MIN_PURGE is given
   (LV_CP_CDN_MIN_PURGE_FAST = 1) **or**
4. Lambda control in open-loop state in homogeneous mode
   (LV_HOM_ACT=1 **and** LV_LAM_LSCL[i] = 0)
   (at least one bank) **or**
5. Lambda control stopped in homogeneous mode
   (LV_HOM_ACT=1 **and** LV_LAM_STOP[i] = 1 **and** LV_LAM_STOP_SHO_PER[i] = 0)
   (at least one bank, both conditions has to be true for one bank) **or**
6. Lambda controller reached limits and min purge state is desired
   [(LV_LAM_LIM_CP=1) **and** (C_STATE_LAM_LIM_CP = MIN_PURGE)]

➔ STATE_CDN_CP = CDN_MIN_PURGE

Conditions for RAMP_OPEN:

1. Conditions for NO_FAST are not fulfilled **and**
2. Conditions for NO_PURGE are not fulfilled **and**
3. Conditions for MIN_PURGE are not fulfilled **and**
4. Limited dynamics for canister purge fulfilled
   (LV_CP_DYW=1) **and**
5. [Lambda control in defined limits in homogeneous mode
   (LV_HOM_ACT=1 **and** LV_LAM_OUT_CP=0) **or**
6. Lambda setpoint within defined limits in stratified mode
   [LV_HOM_ACT=0 **and** (C_LAMB_SP_S_CP_BOL < LAMB_SP_S <C_LAMB_SP_S_CP_TOL)]]
   and
7. Engine operation within CP window (LV_CP_WIN = 1)

➔ STATE_CDN_CP = CDN_RAMP_OPEN

Conditions for RAMP_OPEN_FAST :

1. Conditions for NO_FAST are not fulfilled **and**
2. Conditions for NO_PURGE are not fulfilled **and**
3. Conditions for MIN_PURGE are not fulfilled **and**
4. Limited dynamics for canister purge fulfilled
   (LV_CP_DYW=1) **and**
5. [Lambda control in defined limits in homogeneous mode
   (LV_HOM_ACT=1 **and** LV_LAM_OUT_CP=0) **or**
6. Lambda setpoint within defined limits in stratified mode
   [LV_HOM_ACT=0 **and** (C_LAMB_SP_S_CP_BOL < LAMB_SP_S <C_LAMB_SP_S_CP_TOL)]]

➔ STATE_CDN_CP = CDN_RAMP_FAST

(the only difference between conditions for RAMP_OPEN and for RAMP_OPEN_FAST is the engine operation window.)

if none of the above conditions are fulfilled

➔ STATE_CDN_CP = NO_CDN

The calculation of STATE_CDN_CP is done in the following order:

```
if          Conditions for manual mode fulfilled
then        STATE_CDN_CP = C_STATE_CDN_CP_MAN
elseif          Conditions for NO_FAST fulfilled
then        STATE_CDN_CP = CDN_NO_FAST
elseif          Conditions for NO_PURGE fulfilled
then        STATE_CDN_CP = CDN_NO_PURGE
elseif          Conditions for MIN_PURGE fulfilled
then        STATE_CDN_CP = CDN_MIN_PURGE
elseif          Conditions for RAMP_OPEN fulfilled
then        STATE_CDN_CP = CDN_RAMP_OPEN
elseif          Conditions for RAMP_OPEN_FAST fulfilled
then        STATE_CDN_CP = CDN_RAMP_FAST
else        STATE_CDN_CP = NO_CDN
endif
end
```

28.5 Torque reserve for canister purge at idle

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_ADD_CP	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
Torque reserve for canister purge at idle					

Input data:

LV_HOM_RUN	LV_IS
CL_MMV	STATE_CP

Input data table order: External inputs – Inputs from other EVAC modules

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQ_ADD_CP	8	0...FFFFH	-1024...1023.96875	0.03125	[Nm]
LDP_CL_MMV_IP_TQ_ADD_CP	8	0...FFH	0...1.99218	0.0078125	[-]
Map for determination of TQ_ADD_CP in dependence of CL_MMV					
C_CRLC_TQ_ADD_CP	1	0...FFH	0...0.99609	3.91E-03	[-]
Filter constant for canister purge torque reserve					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TQ_ADD_CP_INH	1	0...1H	0...1	1	[-]
Logical variable for inhibiting the torque reserve for canister purge in idle					

General Information:

The torque reserve for canister purge in idle should be active, when the canister purge function is in a controlled mode, so if STATE_CP = MAX_PURGE.

By increasing the engine load without increasing the torque output, a higher purge rate is possible.

Signal flow diagram:

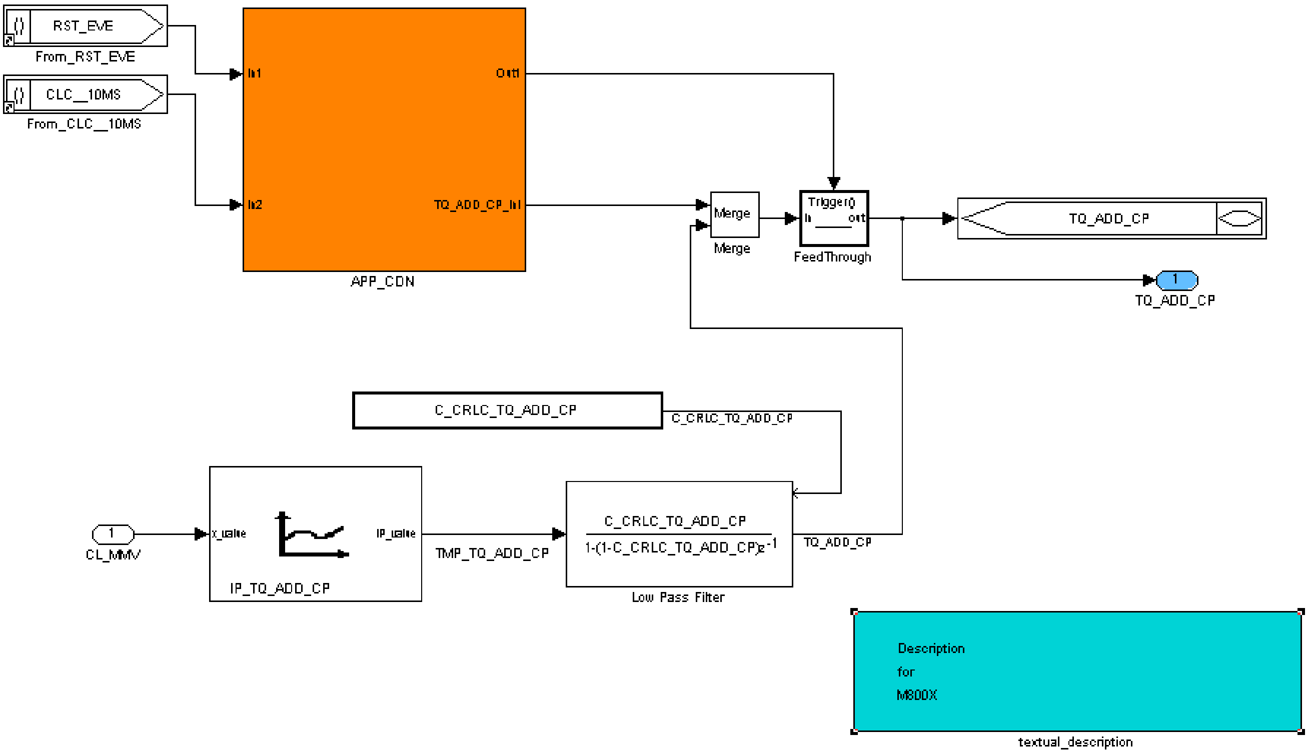


Figure 1: *Torque reserve for canister purge at idle*

FUNCTION DESCRIPTION:

General information:

Activation:

LV_IS = 1
and STATE_CP = MAX_PURGE
and LC_TQ_ADD_CP_INH = 0
and LV_HOM_RUN = 1

Deactivation:

LV_IS = 0
or STATE_CP ≠ MAX_PURGE
or LC_TQ_ADD_CP_INH = 1
or LV_HOM_RUN = 0

Initialization: at reset or deactivation
TQ_ADD_CP = 0

Recurreny: **20** ms

Formula section:

TMP_TQ_ADD_CP = IP_TQ_ADD_CP (CL_MMV)

$TQ\_ADD\_CP_n$= // filtered value of *TMP_TQ_ADD_CP*

$TQ\_ADD\_CP_{n-1} + C\_CRLC\_TQ\_ADD\_CP * (TMP\_TQ\_ADD\_CP - TQ\_ADD\_CP_{n-1})$

28.6 EVAC flow setpoints

DI version

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FLOW_CPS_SP_MIN_PURGE	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Setpoint value for the flow through the CPS for MIN_PURGE mode					
MFF_MAX_CP	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Maximum limit on the fuel flow mass					
FLOW_MAX_CPS	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Maximum setpoint value for the flow through the CPS					
FLOW_MAX_PHY_CPS	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Physical limit: flow at 100% opened valve					
MFF_KGH_MAX_CP	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Fuel flow max from the ACF					
RATIO_FUEL_SP_MAX_PURGE	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Setpoint value for the relative part of fuel from CPS compared to total fuel in cylinder for normal operation					
FAC_FLOW_SP_MIN_PURGE	V	0...FFFFH	0...3.99993	0.061e-3	[-]
correction factor for the flow setpoint in MIN_PURGE when CL_MMV known					
FAC_FLOW_COR_CLL_CP	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Flow target correction factor					

Input data:

TQI_SP_S	MFF_SP_HOM_ENG[NC_CBK_EX_NR]	LV_CP_ACT	LC_LAM_CBK_CPS
N_32	NC_CYL_NR	LV_CP_ENA	CL_MMV
TCO	MAF_CYL	MFF_SP_MV	
LV_CL_CLC_VLD	TQI_SP_S	LV_HOM_ACT	PRS_CPS
STATE_OPM_ENG_CP	MFF_SP_MV	AMP	AMP_DELTA_IT_CP
FTL			

Input data table order: External inputs – Inputs from other EVAC modules

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FLOW_MAX_CPS	12*12	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MAF_2_EVAC	12	0...FFH	0...1389	5.4470588	[mg/stk]
flow setpoint for MAX_PURGE					
IP_FLOW_MAX_2_CPS	12*12	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
flow setpoint for MAX_PURGE in stratifid mode					
IP_FLOW_CPS	16	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_PRS_CPS_1_EVAC	16	0...FFFFH	-2717... 2716.91708	0.0829163	[hPa]
FLOW_CPS for fully opened CPS (CPPWM=100%)					
IP_FLOW_CPS_ENG_MOD_0	12*12	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MAF_2_EVAC	12	0...FFH	0...1389	5.4470588	[mg/stk]
flow setpoint for MIN_PURGE at ENG_MOD_0					
IP_FLOW_CPS_ENG_MOD_1	12*12	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_TQI_SP_S_3_EVAC	12	0...7FFFH	0...1023.97	0.03125	[Nm]
flow setpoint for MIN_PURGE at ENG_MOD_1					
IP_FLOW_CPS_ENG_MOD_2	12*12	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_TQI_SP_S_3_EVAC	12	0...7FFFH	0...1023.97	0.03125	[Nm]
flow setpoint for MIN_PURGE at ENG_MOD_2					
IP_RATIO_FUEL_MAX_CP_MOD_0	12*12	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MAF_2_EVAC	12	0...FFH	0...1389	5.4470588	[mg/stk]
Maximum fuel flow through the CPS at MOD_0					
IP_RATIO_FUEL_MAX_CP_MOD_1	12*12	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDP_TQI_SP_S_IP_RATIO_FUEL_MAX	12	0...7FFFH	0...1023.96875	0.03125	[Nm]
Maximum fuel flow through the CPS at MOD_1					
IP_RATIO_FUEL_MAX_CP_MOD_2	12*12	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDP_TQI_SP_S_IP_RATIO_FUEL_MAX	12	0...7FFFH	0...1023.96875	0.03125	[Nm]
Maximum fuel flow through the CPS at MOD_2					
IP_RATIO_FUEL_MAX_CP_COR	5	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_TCO_IP_RATIO_FUEL_MAX_CP	5	0...FEH	-48...142.5	0.75	[°C]
Correction factor for MFF_MAX_CP					
IP_FAC_FLOW_SP_MIN_PURGE	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_CL_MMV_IP_FAC_FLOW_MIN	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
weighting factor for the CL deviation from the worstcase CL					
C_MFF_MIN_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Minimum limit on mass fuel flow					
C_CL_MAX_MIN_PURGE	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
on this worst case CL is based the base calibration of MIN_PURGE flow setpoint					
C_FAC_FLOW_SP_MIN_PURGE_MAX	1	0...FFFFH	0...3.99993	0.061e-3	[-]
maximum correction of MIN_PURGE flow setpoint					
C_FTL_MIN_FLOW_COR_CP	1	0...7FH	0...127	1	[l]
Lower fuel tank level threshold for flow correction					
C_FAC_FLOW_COR_FTL_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Flow corrrection factor if fuel tank level condition is fulfilled					
IP_FAC_FLOW_COR_AMP_CP	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_AMP_IP_FAC_FLOW_COR_CP	8	0...FFFFH	0...5434	0.0829175	[hPa]
Flow correction factor via AMP					
IP_FAC_FLOW_COR_AMP_DELTA_CP	8	0...FFFFH	0...0.99998	0.0153e-3	[-]

LDP_AMP_DELTA_IP_FAC_FLOW_COR	8	0...FFFFH	-2717.04145... 2716.95854	0.0829175	[hPa]
Flow correction factor via AMP_DELTA_IT_CP					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_FAC_FLOW_SP_MIN_PURGE	-	0...1H	0...1	1	[-]
switch to activate / deactiate "intelligent MIN_PURGE"					

General information:

This module provides the flow setpoint for MIN_PURGE and the FUEL_FLOW setpoint for closed loop purge (“normal purge”).

Application conditions:

Initialisation: *on activation and on deactivation:*

FLOW_CPS_SP_MIN_PURGE = 0
RATIO_FUEL_SP_MAX_PURGE = 0
MFF_MAX_CP = 0
FLOW_MAX_CPS = 0
FLOW_MAX_PHY_CPS = 0

Recurrence: **100** ms
Activation: LV_CP_ACT = 1
Deactivation LV_CP_ACT = 0

Formula section:

```
// Flow setpoint for MIN_PURGE & fuel flow ratio for closed loop

case selection of STATE_OPM_ENG_CP
ENG_MOD_0:
        FLOW_CPS_SP_MIN_PURGE =
                IP_FLOW_CPS_ENG_MOD_0 (N_32, MAF_SP_TQI)
        RATIO_FUEL_SP_MAX_PURGE =
                IP_RATIO_FUEL_MAX_CP_ENG_MOD_0 (N_32, MAF_SP_TQI)
ENG_MOD_1:
        FLOW_CPS_SP_MIN_PURGE =
                IP_FLOW_CPS_ENG_MOD_1 (N_32, TQI_SP_S)
        RATIO_FUEL_SP_MAX_PURGE =
                IP_RATIO_FUEL_MAX_CP_ENG_MOD_1 (N_32, TQI_SP_S)
ENG_MOD_2:
        FLOW_CPS_SP_MIN_PURGE =
                IP_FLOW_CPS_ENG_MOD_2 (N_32, TQI_SP_S)
        RATIO_FUEL_SP_MAX_PURGE =
            IP_RATIO_FUEL_MAX_CP_ENG_MOD_2 (N_32, TQI_SP_S)
End case selection


if      LC_LAM_CBK_CPS=0                                  // cylinder bank selection
then TMP_MFF_SP = MFF_SP_HOM_ENG[1]
```

```
else  TMP_MFF_SP = MFF_SP_HOM_ENG[2]
endif
end

// Flow setpoint for MAX_PURGE
FLOW_MAX_PHY_CPS      = IP_FLOW_CPS (PRS_CPS)

if FTL < C_FTL_MIN_FLOW_COR_CP
then
FAC_FLOW_COR_CLL_CP =
IP_FAC_FLOW_COR_AMP_CP(AMP)
* IP_FAC_FLOW_COR_AMP_DELTA_CP(AMP_DELTA_IT_CP)
else
FAC_FLOW_COR_CLL_CP =
IP_FAC_FLOW_COR_AMP_CP(AMP)
* IP_FAC_FLOW_COR_AMP_DELTA_CP(AMP_DELTA_IT_CP)
* C_FAC_FLOW_COR_FTL_CP
endif

if    STATE_OPM_ENG_CP = 0
then  FLOW_MAX_CPS          = min [ IP_FLOW_MAX_CPS (N_32, MAF_SP_TQI),
                              FLOW_MAX_PHY_CPS] * FAC_FLOW_COR_CLL_CP
else  FLOW_MAX_CPS          = min [ IP_FLOW_MAX_2_CPS (N_32, MFF_SP_MV),
                              FLOW_MAX_PHY_CPS] * FAC_FLOW_COR_CLL_CP
endif


// "intelligent MIN_PURGE"
if    LV_CL_CLC_VLD                        = 1
and   LC_FAC_FLOW_SP_MIN_PURGE        = 1
then  FAC_FLOW_SP_MIN_PURGE =
      min [ C_FAC_FLOW_SP_MIN_PURGE_MAX,
            C_CL_MAX_MIN_PURGE / CL_MMV
          * IP_FAC_FLOW_SP_MIN_PURGE (CL_MMV) ]
else  FAC_FLOW_SP_MIN_PURGE                = 1
endif
FLOW_CPS_SP_MIN_PURGE =
      min [ FLOW_CPS_SP_MIN_PURGE * FAC_FLOW_SP_MIN_PURGE,
            FLOW_MAX_CPS]

// maximum CP fuel mass for closed loop
MFF_KGH_MAX_CP = [TMP_MFF_SP - C_MFF_MIN_CP] * (N_32 * NC_CYL_NR * 3) / 10^5
MFF_MAX_CP =
      min   [MFF_SP_MV * RATIO_FUEL_SP_MAX_PURGE *
            IP_RATIO_FUEL_MAX_CP_COR (TCO), MFF_KGH_MAX_CP]
end
```

28.7 Evaporative emission control

Should be calculated before LACO calculation!

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CP	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH BH	CP_NOT_ACT NO_PURGE RAMP_TO_NO_PURGE WAIT_RAMP_OPEN MIN_PURGE - - - RAMP_OPEN RAMP_OPEN_FAST MAX_PURGE RAMP_CLOSE	1	[-]
State of Evaporative Emission Control Function					
STATE_CLL_DEAC_CP	V/O	0H 1H 2H 3H	CP_NO_DEAC RAMP_OPEN_DEAC MAX_PURGE_DEAC RAMP_FAST_DEAC	1	[-]
Deactivation states of the Evaporative Emission Control					
STATE_LOCK_CP	V/O	0H 1H 2H	UNLOCKED NO_PURGE_LOCKED MIN_PURGE_LOCKED	1	[-]
STATE_CP locked in this state					
FLOW_TAR_CPS	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
CPS flow setpoint, target flow setpoint to be reached after ramping					
FLOW_CTL_CPS	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
CPS flow setpoint, EVAC internal setpoint					
FLOW_SP_CPS	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
CPS flow setpoint, after merge of all flow requests					
FLOW_COR_CPS	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
CPS flow setpoint, ambient conditions corrected, passed to valve control					
FLOW_CPS	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
CPS flow, pseudo-feedback value (limited to physical maximum)					
FLOW_GRD_CPS	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow gradient (flow change / 20ms) to be applied (may be added or substracted)					
FLOW_CTL_CPS_DI	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow through the CPS for disabled NORMAL_PURGE operation					
FLOW_DLY_CP	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Padé filter: filtered value of FLOW_CPS					
FLOW_DLY_MMV_CP	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Lowpass filter: filtered value of FLOW_DLY_CP					
FLOW_DLY_OLD_1_CP	-	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Old value (n-1) of FLOW_DLY_CP for Padé Filter calculation for FLOW_DLY_CP					
FLOW_DLY_OLD_2_CP	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Old value (n-2) of FLOW_DLY_CP for Padé Filter calculation for FLOW_DLY_CP					
FLOW_CPS_OLD_1	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]

Old value (n-1) of FLOW_CPS for Padé Filter calculation for FLOW_DLY_CP					
FLOW_CPS_OLD_2	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Old value (n-2) of FLOW_CPS for Padé Filter calculation for FLOW_DLY_CP					
FLOW_TOT_CPS	V	0...FFFFFFFFH	0...1455.99999	0.339e-6	[g]
Total of FLOW_CPS during emission cycle					
LV_CP_CDN_RAMP_OPEN	V/O	0...1H	0...1	1	[-]
RAMP_OPEN now possible					
LV_CP_CDN_RAMP_OPEN_FAST	V	0...1H	0...1	1	[-]
RAMP_OPEN_FAST now possible					
LV_CP_RAMP_OPEN_ACT	V/O	0...1H	0...1	1	[-]
Logical value for active RAMP_OPEN operation					
LV_RAMP_OPEN_ACT	V/O	0...1H	0...1	1	[-]
Logical value for active RAMP_OPEN operation					
LV_STATE_MEM_CP_CLL	V/O	0...1H	0...1	1	[-]
=1: last STATE_CP is a NORMAL_PURGE state					
LV_T_DLY_REQ_CP	V	0...1H	0...1	1	[-]
C_T_DLY_REQ_CP time elapsed					
LV_DLY_GAS_CP	V	0...1H	0...1	1	[-]
=0: waiting for gas transfer time before further flow incrementatin in RAMP_OPEN / RAMP_OPEN_FAST					
LV_CL_CLC_AVL	V/O	0...1H	0...1	1	[-]
interface flag to cat.diag.: cl known;					
LV_CL_CLC_VLD	V/O	0...1H	0...1	1	[-]
actual CL value usable for "intelligent MIN_PURGE"					
LV_CP_ACT_REQ	V/O	0...1H	0...1	1	[-]
Request to inhit Catalyst and O2 sensor diagnosis for allowing purge canister activation					
LV_CP_CLL	V/O	0...1H	0...1	1	[-]
=1: CP currently in close loop operation (=NORMAL_PURGE)					
LV_CP_AFL_IT	-	0...1H	0...1	1	[-]
temporary variable: internal copy of lv_afl (0: rich; 1: lean)					
LV_CL_MDL_CAT_PURGE_ACT_CP	O/V	0...1H	0...1	1	[-]
Max purge state active at catalyst purge with canister load model					
CL_MMV_CLC_END	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
Canister load at the end of MAX_PURGE operation					
T_DLY_MAX_CP	O/V	0...FFFFH	0...6553.5	0.1	[s]
Time counter between the end of a MAX_PURGE and a new learning request (distinction between RAMP_OPEN and RAMP_OPEN_FAST)					
T_DI_CP	V	0...FFFFH	0...6553.5	0.1	[s]
time counter for disabled RAMP_OPEN, RAMP_OPEN_FAST or MAX_PURGE					
T_DLY_CP	V	0...FFFFH	0...6553.5	0.1	[s]
lambda controller stabilisation time counter					
T_DLY_CP_OPEN	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter for gas transfer time waiting in RAMP_OPEN and RAMP_OPEN_FAST					
PQ_CP_SP	V/O	0...FFFFH	0...0.99998	0.0153e-3	[-]
PQ-setpoint for canister purge					
FLOW_FAC_CP	V	0...FFFFH	0...3.99993	0.061e-3	[-]
correction factor for flow decrementation during RAMP_CLOSE ramp					
FAC_TIA_AMP_CP	V	0...FFH	0...1.99218	0.0078125	[-]
Flow correction versus air temperature and ambient pressure					
MAF_CPS	O/V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
MAF through the CPS					
MAF_CP	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
(MAF_KGH throttle) MAF_KGH_THR + (Mass air flow through the canister purge valve) MAF_CPS					
MAF_DLY_CP	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Padé filter: filtered value of MAF_CP					
MAF_DLY_MMV_CP	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Lowpass filter: filtered value of MAF_DLY_CP					
MAF_DLY_OLD_1_CP	-	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Old value (n-1) of MAF_DLY_CP for Padé Filter calculation for MAF_DLY_CP					
MAF_DLY_OLD_2_CP	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Old value (n-2) of MAF_DLY_CP for Padé Filter calculation for MAF_DLY_CP					
MAF_CP_OLD_1	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]

Old value (n-1) of MAF_CP for Padé Filter calculation for MAF_DLY_CP					
MAF_CP_OLD_2	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Old value (n-2) of MAF_CP for Padé Filter calculation for MAF_DLY_CP					
T_DLY_REQ_CP_TMP	-	0...FFFFH	0...65535	1	[s]
temporary variable: timer for LV_T_DLY_REQ_CP calculation					
PQ_SP_CP_TMP	-	0...FFFFH	0...0.99998	0.0153e-3	[-]
temporary variable for PQ_SP_CP calculation					
FLOW_INC_CTR_TMP	-	0...FFH	0...255	1	[-]
temporary variable: counter of incrementation steps during RAMP_OPEN & RAMP_OPEN_FAST					
T_LOCK_CP_TMP	V	0...FFFFH	0...1310.7	0.02	[s]
Time counter for locking MIN_PURGE or NO_PURGE after canister empty detection					
MAF_DLY_CPS	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Padé filter: filtered value of MAF_CPS					
MAF_DLY_MMV_CPS	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Lowpass filter: filtered value of MAF_DLY_CPS					
MAF_DLY_OLD_1_CPS	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Old value (n-1) of MAF_DLY_CPS for Padé Filter calculation for MAF_DLY_CPS					
MAF_DLY_OLD_2_CPS	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Old value (n-2) of MAF_DLY_CPS for Padé Filter calculation for MAF_DLY_CPS					
MAF_CPS_OLD_1	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Old value (n-1) of MAF_CPS for Padé Filter calculation for MAF_DLY_CPS					
MAF_CPS_OLD_2	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Old value (n-2) of MAF_CPS for Padé Filter calculation for MAF_DLY_CPS					
T_DLY_CL_MDL	V/O	0...FFFFH	0...6553.5	0.1	[s]
Time counter to activate CL-Model if lambda gradient greater than a threshold					
MAF_INT_CP	V	0...FFFFH	0...1820.42	0.0277778	[g]
MAF integal for lambda controller stabilisation in Wait Ramp Open					
T_MAF_INT_CP	V	0...FFFFH	0...6553.5	0.1	[s]
Time limitation for MAF integration at state Wait Ramp Open					

Input data:

AMP	CL	CL_MMV	FAC_FLOW_COR_CLL_CP
FAC_FLOW_COR_CP_EXT	FAC_FLOW_TAR_COR_CP	FLOW_CPS_SP_MIN_PURGE	FLOW_MAX_CPS
FLOW_MAX_PHY_CPS	FLOW_SP_CPS_EVAP	LC_CP_CLL_INH_MAN	LV_CL_MDL_ENA
LV_CP_ACT	LV_CP_AFL	LV_CP_CDN_MIN_PURGE_FAST	LV_CP_CLOSE_1
LV_CP_CLOSE_2	LV_CP_CLOSE_REQ	LV_CP_CMU_CMB	LV_CP_ENA
LV_CP_NEW_RAMP_OPEN	LV_CP_NEW_RAMP_OPEN_FAST	LV_CPS_AD_ACT	LV_DIAGCP_CPS_ACT
LV_FAC_LAM_SHIFT_CP_AVL	LV_FLOW_TAR_COR_CP	LV_HOM_ACT	LV_HOM_AFL_ACT
LV_N_MAX_CP	MAF	MAF_CYL	MAF_KGH_FG_PRED_COR_CP
MAF_THR	MFF_BUF_CP	MFF_CP	MFF_MAX_CP
MFF_SP_MV	N_32	NC_CYL_NR	PRS_CPS
STATE_CDN_CP	STATE_OPM_ENG_CP	T_RAMP_OPEN_STAB	TIA_THR
TQI_SP_S	LV_CP_CAT_PURGE	LV_CL_MDL_ACT	

Input data table order: External inputs – Inputs from other EVAC modules

Import actions:

ACTION_EVAM_Purge ()
[imported from EVAM, spec B02Q, "Activation conditions for evaporative system monitoring]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CL_CAT_DIAG	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Canister load threshold to enable cat.diag. in MIN_PURGE (if cl < thd)					
C_CL_MMV_PQ	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
CL_MMV threshold to force PQ in Max purge during strat. or Hom. lean burn modes					
C_CL_MMV_REQ	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
CL_MMV limit value to give priority to purge canister					
C_CRLC_PQ_SP_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
correlation constant to calculate PQ_SP_CP					
C_FAC_DEC_COR_1_CP	1	0...FFFFH	0...3.99993	0.061e-3	[-]
correction factor for closing ramp speed when high closing speed requested from OBD sequencer					
C_FAC_DEC_COR_2_CP	1	0...FFFFH	0...3.99993	0.061e-3	[-]
correction factor for closing ramp speed when medium closing speed requested from OBD sequencer					
C_FAC_DEC_COR_CMU_CP	1	0...FFFFH	0...3.99993	0.061e-3	[-]
correction factor for closing ramp speed when combustion commutation requested					
C_FAC_FLOW_MAX_INI_CP	1	0...FFH	0...0.99609	3.9063e-3	[-]
Factor to limit FLOW_CTL_CPS at re-entering MAX_PURGE for low canister load					
C_FAC_FLOW_CAT_PURGE_CP	1	0...FFFFH	0...3.99993	0.061e-3	[-]
Flow reduction factor at MaxPurge at Cat-purge with CL-Model activation					
C_FAC_FLOW_CAT_PURGE_INI_CP	1	0...FFFFH	0...3.99993	0.061e-3	[-]
Flow reduction factor at reentry of MaxPurge at Cat-purge with CL-Model activation					
C_FAC_T_DLY_PURGE_TUBE_CP	1	0...FFFFH	0...3.99993	0.061e-3	[-]
Factor for Gas delay time in Wait Ramp Open					
C_FLOW_CPS_AS	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
FLOW_SP_CPS correction of application system					
C_FLOW_CPS_GRD_PURGE_TUBE	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow gradient for tube purging in WAIT_RAMP_OPEN					
C_FLOW_CPS_GRD_PURGE_TUBE_CLOSE	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow gradient for tube purging in WAIT_RAMP_OPEN (at closing cps)					
C_FLOW_CPS_SP_PURGE_TUBE	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow setpoint for tube purging in WAIT_RAMP_OPEN					
C_FLOW_DLY_CRLC_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for FLOW_DLY_CP-filter					
C_FLOW_INI_CPS	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
FLOW_CTL_CPS initialization value					
C_MAF_DLY_CRLC_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for MAF_DLY_CP-filter					
C_MAF_DLY_CRLC_CPS	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for MAF_CPS_DLY-filter					
C_MAF_INT_MAX_CP_0	1	0...FFFFH	0...1820.42	0.0277778	[g]
MAF integral threshold for lambda controller stabilisation in Wait Ramp Open (ENG_MODE = 0)					
C_MAF_INT_MAX_CP_1	1	0...FFFFH	0...1820.42	0.0277778	[g]
MAF integral threshold for lambda controller stabilisation in Wait Ramp Open (ENG_MODE != 0)					
C_MFF_MAX_CPS_DI	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Max. MFF for flow through cps on reentering MAX_PURGE					
C_MFF_RATIO_FLOW_INI_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Fuel Ratio for calculation of flow setpoint at re-entry in Max Purge					
C_T_CP_PURGE_TUBE	1	0...FFFFH	0...6553.5	0.1	[s]
Time at the start of WAIT_RAMP_OPEN wherein the purge tube may be purged itself					
C_T_DI_MAX_CP_0	1	1...FFFFH	0.1...6553.5	0.1	[s]
Maximum time for disabling MAX_PURGE operation (ENG_MODE = 0)					

C_T_DI_MAX_CP_1	1	1...FFFFH	0.1...6553.5	0.1	[s]
Maximum time for disabling MAX_PURGE operation (ENG_MODE != 0)					
C_T_DI_RAMP_FAST_CP_0	1	1...FFFFH	0.1...6553.5	0.1	[s]
Maximum time for disabling RAMP_OPEN_FAST operation (ENG_MODE = 0)					
C_T_DI_RAMP_FAST_CP_1	1	1...FFFFH	0.1...6553.5	0.1	[s]
Maximum time for disabling RAMP_OPEN_FAST operation (ENG_MODE != 0)					
C_T_DI_RAMP_OPEN_CP_0	1	1...FFFFH	0.1...6553.5	0.1	[s]
Maximum time for disabling RAMP_OPEN operation (ENG_MODE = 0)					
C_T_DI_RAMP_OPEN_CP_1	1	1...FFFFH	0.1...6553.5	0.1	[s]
Maximum time for disabling RAMP_OPEN operation (ENG_MODE != 0)					
C_T_DLY_CL_MDL	1	0...FFFFH	0...6553.5	0.1	[s]
Time to activate CL-Model					
C_T_DLY_CP	1	1...FFFFH	0.1...6553.5	0.1	[s]
Time for lambda control stabilization before to start RAMP_OPEN or RAMP_OPEN_FAST					
C_T_DLY_MAX_CP_0	1	0...FFFFH	0...6553.5	0.1	[s]
Time between two MAX_PURGE operation for RAMP_OPEN_FAST activation (ENG_MODE = 0)					
C_T_DLY_MAX_CP_1	1	0...FFFFH	0...6553.5	0.1	[s]
Time between two MAX_PURGE operation for RAMP_OPEN_FAST activation (ENG_MODE != 0)					
C_T_DLY_REQ_CP	1	0...FFFFH	0...65535	1	[s]
Time to set LV_T_DLY_REQ_CP					
C_T_LOCK_CP	1	0...FFFFH	0...1310.7	0.02	[s]
Locking time for locking MIN_PURGE or NO_PURGE after canister empty detection					
C_T_MAX_MAF_INT_CP_0	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum runtime of MAF integration as stabilisation time for the lambda controller (ENG_MODE = 0)					
C_T_MAX_MAF_INT_CP_1	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum runtime of MAF integration as stabilisation time for the lambda controller (ENG_MODE != 0)					
C_T_MIN_MAF_INT_CP_0	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum runtime of MAF integration as stabilisation time for the lambda controller (ENG_MODE = 0)					
C_T_MIN_MAF_INT_CP_1	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum runtime of MAF integration as stabilisation time for the lambda controller (ENG_MODE != 0)					
ID_FLOW_INC_NR_CP	4	1...FFH	1...255	1	[-]
LDP_N_32_ID_FLOW_INC_NR_CP	4	0...FFH	0...8160	32	[rpm]
Increment number for FLOW_CTL_CPS calculation during RO and ROF (steps à 20ms)					
IP_FAC_AMP_CP	6	0...FFH	0...1.99218	0.0078125	[-]
LDPM_AMP_1_EVAC	6	0...FFFFH	0...5434	0.0829175	[hPa]
Correction factor for ambient pressure					
IP_FAC_TIA_CP	6	0...FFH	0...1.99218	0.0078125	[-]
LDP_TIA_THR_IP_FAC_TIA_CP	6	0...FEH	-48...142.5	0.75	[°C]
Correction factor for intake air temperature					
IP_FLOW_CPS_DEC	6	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_MFF_SP_MV_1_EVAC	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
Decrement value on FLOW_CTL_CPS during MAX_PURGE					
IP_FLOW_CPS_INC	6	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_MFF_SP_MV_1_EVAC	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
Increment value on FLOW_CTL_CPS during MAX_PURGE					
IP_FLOW_DEC_CP	6	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_MFF_SP_MV_1_EVAC	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
Decrement value for ramp decrease of FLOW_CTL_CPS					
IP_FLOW_FAC_CP	6	0...FFFFH	0...3.99993	0.061e-3	[-]
LDP_CL_IP_FLOW_FAC	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
Correction factor for ramp increase of FLOW_CTL_CPS					
IP_FLOW_INC_COR_HOM_CP	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correction of FLOW_GRD_CPS in homogen mode					
IP_FLOW_INC_CP	6	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_MFF_SP_MV_1_EVAC	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
Increment value for ramp increase of FLOW_CTL_CPS in Ramp open mode					
IP_FLOW_INC_FAST_CP	6	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_MFF_SP_MV_1_EVAC	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
Increment value for ramp increase of FLOW_CTL_CPS in Ramp Open Fast mode					

IP_FLOW_MAX_OPEN_CPS	12*12	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MAF_2_EVAC	12	0...FFH	0...1389	5.4470588	[mg/stk]
Maximum limit for FLOW_CTL_CPS value for RAMP_OPEN & RAMP_OPEN_FAST					
IP_FLOW_MAX_OPEN_CPS_2	12*12	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_N_32_2_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_MV_2_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Max. limit for FLOW_CTL_CPS value for RAMP_OPEN & RAMP_OPEN_FAST with STATE_OPM_ENG_CP = 1					
IP_PQ_CP_SP	6*6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_N_32_1_EVAC	6	0...FFH	0...8160	32	[rpm]
LDPM_TQI_SP_S_2_EVAC	6	0...7FFFH	0...1023.97	0.03125	[Nm]
Forced PQ during stratified mode for RAMP_OPEN & MAX_PURGE					
IP_T_DLY_CL_MMV_DYW	6	1...FFFFH	0.1...6553.5	0.1	[s]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Stabilization delay on CL_MMV value during RAMP_OPEN					
IP_T_DLY_CP	6	0...FFFFH	0...63.99902	0.9766e-3	[-]
LDPM_MAF_KGH_COR_IP_T_DLY_CP	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Time delay for MAF_CYL (delay time (s) / time recurrence(0.1s))					
IP_T_DLY_CP_2	6	0...FFFFH	0...63.99902	0.9766e-3	[-]
LDPM_MAF_KGH_COR_IP_T_DLY_CP	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Time delay for MAF_CYL (delay time (s) / time recurrence(0.1s)) with STATE_OPM_ENG_CP = 1					

Hint: For each operating point, the corresponding value of IP_FLOW_MAX_OPEN_CPS and IP_FLOW_MAX_OPEN_CPS_2 must not exceed the corresponding value of IP_FLOW_MAX_CPS.

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_LAM_LIN_CPS	-	0...1H	0...1	1	[-]
Type of O2 sensor (0 -> binary sensor , 1 -> linear sensor)					
LC_PQ_SP_CP_HOM_AFL	-	0...1H	0...1	1	[-]
switch to enable forced pq during hom_afl operation					
LC_FLOW_CPS_AS	-	0...1H	0...1	1	[-]
C_FLOW_CPS_AS activation					
LC_CP_TRAN_DEAC_CLL	-	0...1H	0...1	1	[-]
1: enable transition back to RAMP_OPEN_FAST after interruption					
LC_CP_TRA_MIN_PURGE_DFT_ENA	-	0...1H	0...1	1	[-]
1: enable transition NO_PURGE (MIN_PURGE with STATE_CDN_CP = NO_CDN					
LC_CP_TRA_MIN_RAMP_OPEN_DIS	-	0...1H	0...1	1	[-]
enable trans. NO_PURGE (MIN_PURGE, if no RO or ROF is possible although STATE_CDN_CP=RO or ROF					
LC_CP_TRA_MAX_MIN_CDN_ENA	-	0...1H	0...1	1	[-]
enable transition MAX_PURGE (MIN_PURGE with STATE_CDN_CP = NO_CDN					
LC_CP_TRA_OPL_CDN_NOT	-	0...1H	0...1	1	[-]
enable transition WAIT_RAMP_OPEN (NO_PURGE if STATE_CDN_CP = NO_CDN					
LC_CP_TRA_OPL_CDN_CLL_NOT	-	0...1H	0...1	1	[-]
enable transition WAIT_RAMP_OPEN (NO_PURGE if LV_CP_CDN_RAMP_OPEN, _FAST no more given					
LC_CP_FLOW_LIM_OPEN_FAST	-	0...1H	0...1	1	[-]
enable old calc. of FLOW_TAR_CPS during ROF: fuel mass based					
LC_CP_TRAN_OPEN_MAX_ENA	-	0...1H	0...1	1	[-]
enable transition RO (MaxP if flow target reached					
C_STATE_MOD_SEL_CP	-	0H 1H 2H 3H	SEL_NO_PURGE SEL_MIN_PURGE SEL_MAX_PURGE SEL_RAMP_CLOSE	1	[-]
selection of the purge mode with empty canister detected at end of RAMP_OPEN					
LC_CP_CAT_PURGE_MDL_ACT	1	0...1H	0...1	1	[-]
transition to Wait-RO at cat-purge will inhibitied due to usage of CL modell					

28.7.1 General information

General information:

The evaporative emissions of the fuel tank system are stored in an activated charcoal filter ACF to prevent an escape of the gas to the environment. The loading capacity of the filter is limited, so the filter must be purged temporarily. Therefore the canister purge solenoid CPS. that is positioned between the charcoal filter and the intake manifold is opened by the engine management system. To prevent driveability and emission problems, the opening of the CPS and so the purge flow MFF_CP must be controlled.

With the Evaporative Emission Control function, the opening of the CPS is controlled in dependence with the fuel mass stored in the canister. Therefore the canister load cl is calculated using the measured mass air flow MAF, the deviation of the lambda-control when the CPS is opened, the mass flow through the CPS and the stoichiometric constant of the fuel (14.7). The opening of the CPS is controlled in dependence with the canister load CL.

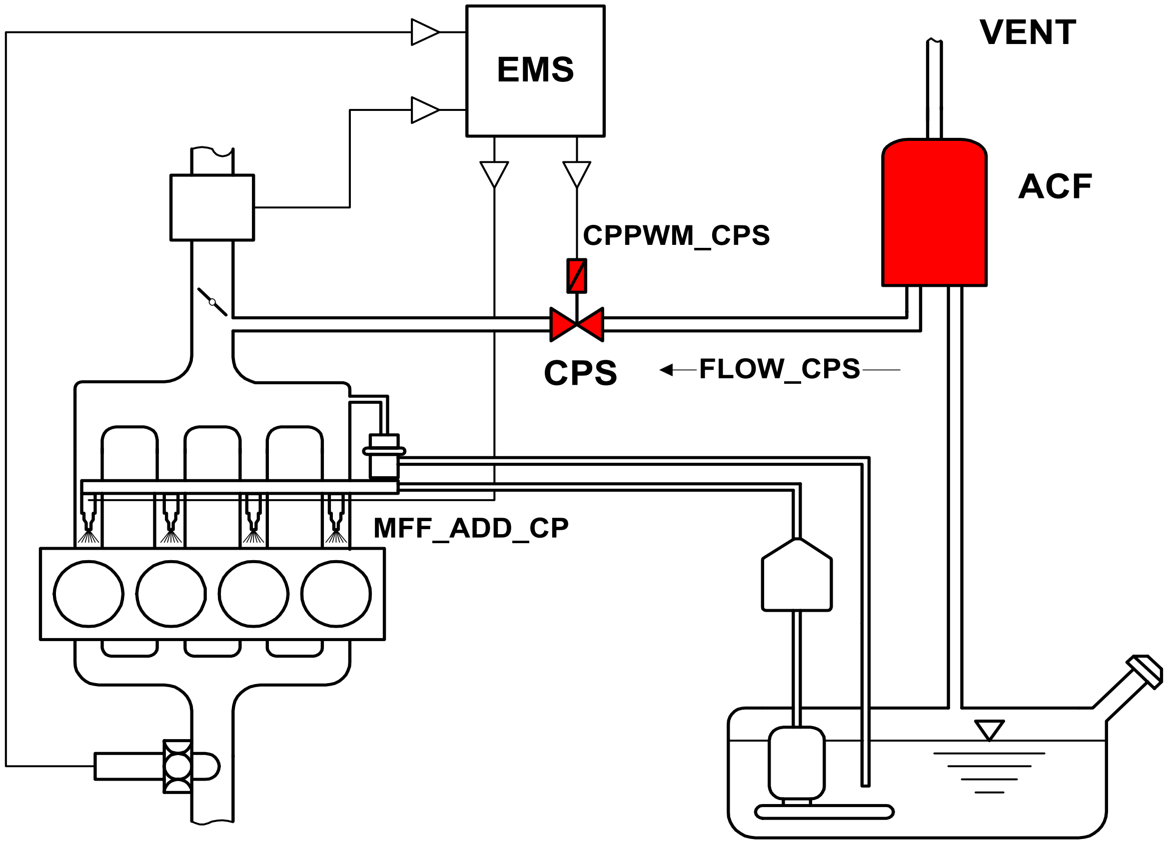


figure 1: evaporative emission control system

Application conditions:

Initialisation: at RST, IGKON or transition to CP_NOT_ACT:

PQ_CP_SP = 1
STATE_CP = CP_NOT_ACT
STATE_CLL_DEAC_CP = CP_NO_DEAC
STATE_LOCK_CP = UNLOCKED
all other values = 0

Recurrence: **20** ms
Activation: --
Deactivation: --

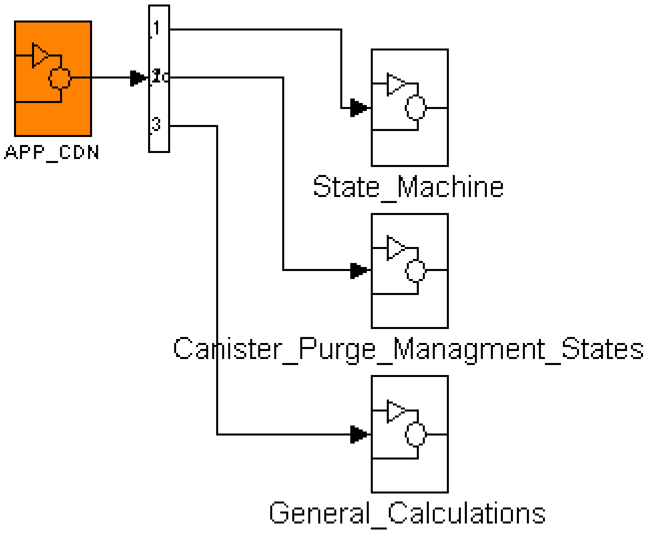


figure 2: structure of the evaporative emission control

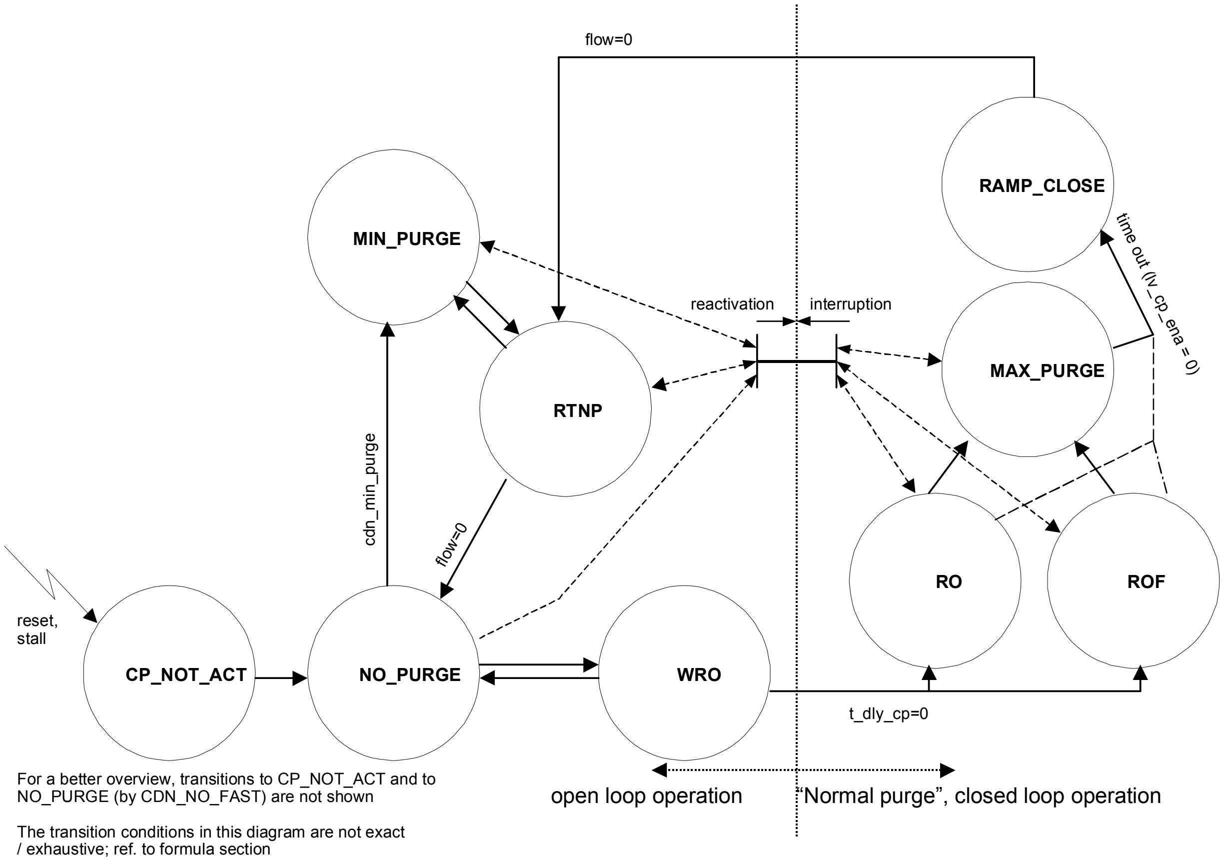
Formula section:

call “State machine”
call “Canister purge management states”
call “General calculations”
end

Temporary adaptation flags :

LV_RAMP_OPEN_ACT = LV_CP_RAMP_OPEN_ACT

28.7.2 State machine



For a better overview, transitions to CP_NOT_ACT and to NO_PURGE (by CDN_NO_FAST) are not shown

The transition conditions in this diagram are not exact / exhaustive; ref. to formula section

figure 3: State machine (simplified)

The canister purge control has the following states:

CP_NOT_ACT	function inactive, initial state
NO_PURGE	Valve closed, no purging.
RAMP_TO_NO_PURGE	Flow is currently ramped down from purging level to zero
MIN_PURGE	open loop purge with low flow
WAIT_RAMP_OPEN	waiting for lambda control stabilisation before leaning mode
RAMP_OPEN	Slow opening of the valve, learning mode
RAMP_OPEN_FAST	Faster opening of the valve; cl known from previous purge
MAX_PURGE	Controlled canister purge with high flow through the ACF
RAMP_CLOSE	Closing of the valve at the end of controlled operation

28.7.2.1 Brief description of the activation conditions of the different states

<u>General information:</u>

In the following, a brief description of the conditions of each state is given, this description is only for information and is <u>not precise</u>. In general, if the conditions below are met, the state described will be active.

The detailed conditions for transitions from each state to each other possible state and the actions to be done on the transitions are described in the next chapter “State Transitions”.

The “controlled states” are RAMP_OPEN, RAMP_OPEN_FAST, MAX_PURGE & RAMP_CLOSE.

The “non-controlled states” are NO_PURGE, RAMP_TO_NO_PURGE, MIN_PURGE and WAIT_RAMP_OPEN.

NORMAL_PURGE Operation (closed loop operation)

If the conditions for NORMAL_PURGE are fulfilled, the NORMAL_PURGE operation is proceeded in the following sequences:

RAMP_OPEN → MAX_PURGE → RAMP_CLOSE

or RAMP_OPEN_FAST → MAX_PURGE → RAMP_CLOSE

These sequences can change, if one of the phases is disabled for a short time and the interrupt is not longer than an applicable time constant

- C_T_DI_RAMP_OPEN_CP_0/_1 (STATE_CLL_DEAC_CP = RAMP_OPEN_DEAC) **or**
- C_T_DI_MAX_CP_0/_1 (STATE_CLL_DEAC_CP = MAX_PURGE_DEAC) **or**
- C_T_DI_RAMP_FAST_CP_0/_1 (STATE_CLL_DEAC_CP = RAMP_FAST_DEAC)

If RAMP_OPEN, RAMP_OPEN_FAST or MAX_PURGE operation was interrupted (any non-controlled state active) and the interruption time is lower than the adjustable times C_T_DI_RAMP_OPEN_CP_0/_1, C_T_DI_RAMP_FAST_CP_0/_1 rsp. C_T_DI_MAX_CP_0/_1, RAMP_OPEN, RAMP_OPEN_FAST or MAX_PURGE operation starts immediately with the values which were stored when the operation was interrupted.

When the interruption times are longer than the corresponding constants, NORMAL_PURGE states are re-started with a new RAMP_OPEN / RAMP_OPEN_FAST phase. (FLOW_CTL_CPS starts with zero.)

CP_NOT_ACT

Function inactive, initial state

1. Engine stopped or stalled
2. coolant temperature threshold not once exceeded

NO_PURGE

Valve closed, no purging.

1. STATE_CDN_CP = CDN_NO_PURGE or CDN_NO_FAST **and**
2. [Fast shut-off necessary **or**
3. ramp from purge states to NO_PURGE finished]

RAMP_TO_NO_PURGE

Flow is currently ramped down from purging level to zero

If fast shut-off is necessary, the state machine will directly switch to NO_PURGE (immediate valve closing without ramp), otherwise, RAMP_TO_NO_PURGE will be active as long as the valve is closed by a ramp

1. STATE_CDN_CP = CDN_NO_PURGE **and**
2. Fast shut-off not necessary **and**
3. Ramp from purge states to NO_PURGE not finished **and**
4. [Previous state was MIN_PURGE or WAIT_RAMP_OPEN **and** time out from time scheduler LV_CP_ENA = 0]

An immediate switch to NO_PURGE without ramp is requested by STATE_CDN_CP = CDN_NO_FAST. In this case, FLOW_CTL_CPS is directly set to 0, no ramping down will occur.

MIN_PURGE

No controlled purge possible, open loop purge with low flow, thus low disturbance of lambda contro.

1. STATE_CDN_CP = CDN_MIN_PURGE **or**
2. Empty canister detected at the end of RAMP_OPEN

RAMP_OPEN

Slow opening of the valve, canister load not known, will be learned during RAMP_OPEN

1. STATE_CDN_CP = CDN_RAMP_OPEN **and**
2. Long time since the last MAX_PURGE (old cl value no more useable) **and**
3. The time delay for lambda control stabilization is elapsed or not necessary (due to MAX_PURGE or RAMP_OPEN disabling).

RAMP_OPEN_FAST

Faster opening of the valve than in RAMP_OPEN; cl known from previous normal purge operation

1. STATE_CDN_CP = CDN_RAMP_FAST **and**
2. Short time since the last MAX_PURGE (old cl value still useable) **and**
3. The time delay for lambda control stabilization is elapsed or not necessary (due to MAX_PURGE or RAMP_OPEN disabling)

MAX_PURGE

Controlled canister purge with high flow through the ACF

1. NO_PURGE / RAMP_TO_NO_PURGE / MIN_PURGE not requested **and**
2. RAMP_OPEN or RAMP_OPEN_FAST normally finished **and**
3. Can be directly reached from the non-controlled states, if MAX_PURGE was previously disabled and disabling time was lower than a threshold (STATE_CLL_DEAC_CP = MAX_PURGE_DEAC)
4. Can be entered at catalyst purge with canister load model running

RAMP_CLOSE

Closing of the valve at the end of controlled operation

1. NO_PURGE / RAMP_TO_NO_PURGE / MIN_PURGE not requested **and**
2. Canister purge time slice elablsed / Lambda adaption time active

28.7.2.2 Preliminary calculations

```
LV_CP_CDN_RAMP_OPEN_FAST =
    { [      T_DLY_MAX_CP       > 0                   // cl from prev. purge useable
or     STATE_CLL_DEAC_CP = RAMP_FAST_DEAC ]           // re-entry with T_.. = 0
and [ STATE_CDN_CP        = CDN_RAMP_FAST
or     STATE_CDN_CP        = CDN_RAMP_OPEN ]
and   LV_CP_NEW_RAMP_OPEN = 0}
       // CDN_RAMP_OPEN may be given, although RAMP_OPEN_FAST is preferable

LV_CP_CDN_RAMP_OPEN =
     [ T_DLY_MAX_CP        = 0
and   STATE_CDN_CP        = CDN_RAMP_OPEN ]
end
```

28.7.2.3 CP_NOT_ACT

```
if          LV_CP_ACT       = 0
then        STATE_CP        = CP_NOT_ACT
else        call  "INITIALIZATION-VALUES on entering NO_PURGE"
            STATE_CP        = NO_PURGE
endif
end
```

28.7.2.4 NO_PURGE

```
if          LV_CP_ACT       = 0
then        STATE_CP        = CP_NOT_ACT

elseif            STATE_CDN_CP = CDN_NO_PURGE
or          STATE_CDN_CP = CDN_NO_FAST
or          LV_CP_ENA       = 0
then        STATE_CP        = NO_PURGE                  // stay in NO_PURGE

elseif            STATE_CDN_CP = CDN_MIN_PURGE
then        call  "INITIALIZATION-VALUES on leaving NO_PURGE"
            call  "INITIALIZATION-VALUES on entering MIN_PURGE"
            STATE_CP        = MIN_PURGE

            // re-entry in Max Purge with activated canister load model at catalyst purge
elseif            STATE_CDN_CP = CDN_RAMP_OPEN
      and   LV_CP_CAT_PURGE = 1
      and   LV_CL_MDL_ACT = 1
then        call  "INITIALIZATION-VALUES on leaving NO_PURGE"
            call  "INITIALIZATION-VALUES on entering MAX_PURGE"
            STATE_CP        = MAX_PURGE

            // re-entry in normal purge states after short deactivation
```

```
elseif          STATE_CLL_DEAC_CP ≠ CP_NO_DEAC
          and   LV_CP_NEW_RAMP_OPEN = 0
          and   LV_CP_NEW_RAMP_OPEN_FAST = 0
then  case selection on STATE_CLL_DEAC_CP
          RAMP_OPEN_DEAC:
          if      LV_CP_CDN_RAMP_OPEN
          then call   "INITIALIZATION-VALUES on leaving NO_PURGE"
                call  "INITIALIZATION-VALUES on entering RAMP_OPEN"
                STATE_CP = RAMP_OPEN
          endif

          RAMP_FAST_DEAC:
          if    [ LV_CP_CDN_RAMP_OPEN_FAST
          and   LC_CP_TRAN_DEAC_CLL ]   // switch to enable old functionality
          then call   "INITIALIZATION-VALUES on leaving NO_PURGE"
                call  "INITIALIZATION-VALUES on entering RAMP_OPEN_FAST"
                STATE_CP = RAMP_OPEN_FAST
          endif

          MAX_PURGE_DEAC:
          call  "INITIALIZATION-VALUES on leaving NO_PURGE"
          call  "INITIALIZATION-VALUES on entering MAX_PURGE"
          STATE_CP      = MAX_PURGE
      end case selection

elseif          (LV_CP_CDN_RAMP_OPEN_FAST or LV_CP_CDN_RAMP_OPEN)
      and   (!LV_CP_CAT_PURGE or !LC_CP_CAT_PURGE_MDL_ACT)
then        call  "INITIALIZATION-VALUES on leaving NO_PURGE"
            call  "INITIALIZATION-VALUES on entering WAIT_RAMP_OPEN"
            STATE_CP      = WAIT_RAMP_OPEN

elseif         {[STATE_CDN_CP = CDN_RAMP_FAST        // No RO possible
            and         LC_CP_TRA_MIN_RAMP_OPEN_DIS ]
      or    [ STATE_CDN_CP = NO_CDN                  // NO_CDN-management
            and         LC_CP_TRA_MIN_PURGE_DFT_ENA ]}
      and   (!LV_CP_CAT_PURGE or !LC_CP_CAT_PURGE_MDL_ACT)
then        call  "INITIALIZATION-VALUES on leaving NO_PURGE"
            call  "INITIALIZATION-VALUES on entering MIN_PURGE"
            STATE_CP      = MIN_PURGE

else        STATE_CP      = NO_PURGE                 // default
endif
end
```

28.7.2.5 RAMP_TO_NO_PURGE

```
if          LV_CP_ACT     = 0
then        STATE_CP      = CP_NOT_ACT
```

```
elseif          STATE_CDN_CP = CDN_NO_FAST
or          FLOW_CTL_CPS = 0                              // flow reduction finished
then        call  "INITIALIZATION-VALUES on leaving RAMP_TO_NO_PURGE"
            call  "INITIALIZATION-VALUES on entering NO_PURGE"
            STATE_CP        = NO_PURGE

elseif          STATE_CDN_CP = CDN_NO_PURGE
or          LV_CP_ENA       = 0
then        STATE_CP        = RAMP_TO_NO_PURGE                   // stay in actual state

elseif          STATE_CDN_CP = CDN_MIN_PURGE
then        call  "INITIALIZATION-VALUES on leaving RAMP_TO_NO_PURGE"
            call  "INITIALIZATION-VALUES on entering MIN_PURGE"
            STATE_CP        = MIN_PURGE

            // re-entry in normal purge states after short deactivation
elseif          STATE_CLL_DEAC_CP ≠ CP_NO_DEAC
            and  LV_CP_NEW_RAMP_OPEN = 0
            and  LV_CP_NEW_RAMP_OPEN_FAST = 0
then  case selection on STATE_CLL_DEAC_CP
            RAMP_OPEN_DEAC:
            if      LV_CP_CDN_RAMP_OPEN
            then  call  "INITIALIZATION-VALUES on leaving RAMP_TO_NO_PURGE"
                  call  "INITIALIZATION-VALUES on entering RAMP_OPEN"
                  STATE_CP = RAMP_OPEN
            endif

            RAMP_FAST_DEAC:
            if     [ LV_CP_CDN_RAMP_OPEN_FAST
            and   LC_CP_TRAN_DEAC_CLL ]  //switch to enable old functionality
            then call  "INITIALIZATION-VALUES on leaving RAMP_TO_NO_PURGE"
                  call  "INITIALIZATION-VALUES on entering RAMP_OPEN_FAST"
                  STATE_CP = RAMP_OPEN_FAST
            endif

            MAX_PURGE_DEAC:
            call  "INITIALIZATION-VALUES on leaving RAMP_TO_NO_PURGE"
            call  "INITIALIZATION-VALUES on entering MAX_PURGE"
            STATE_CP        = MAX_PURGE
      end case selection

elseif         [ STATE_CDN_CP = NO_CDN           // NO_CDN-management
and         LC_CP_TRA_MIN_PURGE_DFT_ENA ]
then        call  "INITIALIZATION-VALUES on leaving RAMP_TO_NO_PURGE"
            call  "INITIALIZATION-VALUES on entering MIN_PURGE"
            STATE_CP        = MIN_PURGE
```

```
else  STATE_CP            = RAMP_TO_NO_PURGE             // default
endif
end
```

28.7.2.6 MIN_PURGE

```
if          LV_CP_ACT     = 0
then        STATE_CP      = CP_NOT_ACT

elseif            STATE_CDN_CP = CDN_NO_FAST
then        call  “INITIALIZATION-VALUES on leaving MIN_PURGE”
            call  “INITIALIZATION-VALUES on entering NO_PURGE”
            STATE_CP      = NO_PURGE

elseif            STATE_CDN_CP = CDN_NO_PURGE
or          LV_CP_ENA     = 0
or         [ STATE_CLL_DEAC_CP = CP_NO_DEAC           // “wait ramp open”
      and  (LV_CP_CDN_RAMP_OPEN_FAST
      or    LV_CP_CDN_RAMP_OPEN) ]
then        call  “INITIALIZATION-VALUES on leaving MIN_PURGE”
            call  “INITIALIZATION-VALUES on entering RAMP_TO_NO_PURGE”
            STATE_CP      = RAMP_TO_NO_PURGE

elseif            STATE_CDN_CP = CDN_MIN_PURGE
then        STATE_CP      = MIN_PURGE                      // stay in MIN_PURGE

            // re-entry in normal purge states after short deactivation
elseif            STATE_CLL_DEAC_CP ≠ CP_NO_DEAC
            and   LV_CP_NEW_RAMP_OPEN = 0
            and   LV_CP_NEW_RAMP_OPEN_FAST = 0
then  case selection on STATE_CLL_DEAC_CP
            RAMP_OPEN_DEAC:
            if    LV_CP_CDN_RAMP_OPEN
            then call   “INITIALIZATION-VALUES on leaving MIN_PURGE”
                  call  “INITIALIZATION-VALUES on entering RAMP_OPEN”
                  STATE_CP = RAMP_OPEN
            endif

            RAMP_FAST_DEAC:
            if    [ LV_CP_CDN_RAMP_OPEN_FAST
            and   LC_CP_TRAN_DEAC_CLL ]   //switch to enable old functionality
            then call   “INITIALIZATION-VALUES on leaving MIN_PURGE”
                  call  “INITIALIZATION-VALUES on entering RAMP_OPEN_FAST”
                  STATE_CP = RAMP_OPEN_FAST
            endif

            MAX_PURGE_DEAC:
            if    [ STATE_CDN_CP ≠ NO_CDN
```

```
                 or     LC_CP_TRA_MAX_MIN_CDN_ENA = 0 ]
                 then   call  "INITIALIZATION-VALUES on leaving MIN_PURGE"
                        call  "INITIALIZATION-VALUES on entering MAX_PURGE"
                        STATE_CP = MAX_PURGE
                 endif
          end case selection

   else          STATE_CP      = MIN_PURGE                  // default
   endif
   end
```

28.7.2.7 WAIT_RAMP_OPEN

```
   if            LV_CP_ACT     = 0
   then          STATE_CP      = CP_NOT_ACT

   elseif               STATE_CDN_CP = CDN_NO_FAST
   or            STATE_CDN_CP = CDN_NO_PURGE
   or            STATE_CDN_CP = CDN_MIN_PURGE
   or            LV_CP_ENA     = 0
   or           [STATE_CDN_CP = NO_CDN
          and    LC_CP_TRA_OPL_CDN_NOT ]
   or           [LV_CP_CDN_RAMP_OPEN                        = 0
          and    LV_CP_CDN_RAMP_OPEN_FAST          = 0
          and    LC_CP_TRA_OPL_CDN_CLL_NOT ]
   then          call  "INITIALIZATION-VALUES on leaving WAIT_RAMP_OPEN"
                 call  "INITIALIZATION-VALUES on entering NO_PURGE"
                 STATE_CP      = NO_PURGE

   elseif               {      (MAF_INT_CP < C_MAF_INT_MAX_CP_0
                        and    T_MAF_INT_CP < C_T_MAX_MAF_INT_CP_0
                        and    STATE_OPM_ENG_CP = ENG_MODE_0)
                 or     (MAF_INT_CP < C_MAF_INT_MAX_CP_1
                        and    T_MAF_INT_CP < C_T_MAX_MAF_INT_CP_1
                        and    STATE_OPM_ENG_CP != ENG_MODE_0)}]
          or     T_DLY_CP > 0
          or     (T_MAF_INT_CP < C_T_MIN_MAF_INT_CP_0
                 and   STATE_OPM_ENG_CP = ENG_MODE_0)
          or     (T_MAF_INT_CP < C_T_MIN_MAF_INT_CP_1
                 and   STATE_OPM_ENG_CP != ENG_MODE_0)
          or     LV_CPS_AD_ACT = 1
   then          STATE_CP      = WAIT_RAMP_OPEN             // stay in WRO

   elseif               LV_CP_CDN_RAMP_OPEN
   then          call  "INITIALIZATION-VALUES on leaving WAIT_RAMP_OPEN"
                 call  "INITIALIZATION-VALUES on entering RAMP_OPEN"
                 STATE_CP      = RAMP_OPEN
```

```
elseif          LV_CP_CDN_RAMP_OPEN_FAST
then        call  "INITIALIZATION-VALUES on leaving WAIT_RAMP_OPEN"
            call  "INITIALIZATION-VALUES on entering RAMP_OPEN_FAST"
            STATE_CP        = RAMP_OPEN_FAST

else        call  "INITIALIZATION-VALUES on leaving WAIT_RAMP_OPEN"
            call  "INITIALIZATION-VALUES on entering NO_PURGE"
            STATE_CP        = NO_PURGE                // default, robustness
endif
end
```

28.7.2.8 RAMP_OPEN

```
if          LV_CP_ACT       = 0
then        STATE_CP        = CP_NOT_ACT

elseif          STATE_CDN_CP = CDN_NO_FAST
then        call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
            call  "INITIALIZATION-VALUES on entering NO_PURGE"
            STATE_CP        = NO_PURGE

elseif          STATE_CDN_CP = CDN_NO_PURGE
then        call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
            call  "INITIALIZATION-VALUES on entering RAMP_TO_NO_PURGE"
            STATE_CP        = RAMP_TO_NO_PURGE

elseif          LV_CP_CDN_RAMP_OPEN = 0
        or  LV_CP_NEW_RAMP_OPEN = 1
        or  LV_CP_NEW_RAMP_OPEN_FAST = 1
then        call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
            call  "INITIALIZATION-VALUES on entering MIN_PURGE"
            STATE_CP        = MIN_PURGE

elseif          LV_CP_ENA       = 0
then        call  "Storage of Values for OBDII EVAP-System Monitoring"
            call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
            call  "INITIALIZATION-VALUES on entering RAMP_CLOSE"
            STATE_CP        = RAMP_CLOSE

elseif          [ ( FLOW_CTL_CPS >= FLOW_TAR_CPS
        and LC_CP_TRAN_OPEN_MAX_ENA = 1)
        or  MFF_CP >= MFF_MAX_CP
        or  T_RAMP_OPEN_STAB = 0 ]                        //  time  counter
for RO elapsed
        and  LV_FLOW_TAR_COR_CP = 0
        then // RAMP_OPEN finished
```

```
                LV_CL_CLC_AVL = [CL_MMV <= C_CL_CAT_DIAG]
                LV_CL_CLC_VLD = 1
                if    [ LV_FAC_LAM_SHIFT_CP_AVL  = 0           // no lambda contr. shift done
                and   T_RAMP_OPEN_STAB = 0 ]                   // but time out
                then  call  "Selection of purge mode with empty canister"
                else  // regular transition to MAX_PURGE
                      call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
                      call  "INITIALIZATION-VALUES on entering MAX_PURGE"
                      STATE_CP = MAX_PURGE
                endif

else            STATE_CP        = RAMP_OPEN                    // default
endif
end
```

28.7.2.8.1 Selection of purge mode with empty canister

```
case selection on C_STATE_MOD_SEL_CP:
NO_PURGE:
        STATE_LOCK_CP       = NO_PURGE
        call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
        call  "INITIALIZATION-VALUES on entering RAMP_TO_NO_PURGE"
        STATE_CP            = RAMP_TO_NO_PURGE
MIN_PURGE:
        STATE_LOCK_CP       = MIN_PURGE
        call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
        call  "INITIALIZATION-VALUES on entering MIN_PURGE"
        STATE_CP            = MIN_PURGE
MAX_PURGE:
        call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
        call  "INITIALIZATION-VALUES on entering MAX_PURGE"
        STATE_CP            = MAX_PURGE
RAMP_CLOSE:
        call  "INITIALIZATION-VALUES on leaving RAMP_OPEN"
        call  "INITIALIZATION-VALUES on entering RAMP_CLOSE"
        STATE_CP            = RAMP_CLOSE
end case selection
end
```

28.7.2.9 RAMP_OPEN_FAST

```
if              LV_CP_ACT       = 0
                STATE_CP        = CP_NOT_ACT

elseif                STATE_CDN_CP = CDN_NO_FAST
then            call  "INITIALIZATION-VALUES on leaving RAMP_OPEN_FAST"
                call  "INITIALIZATION-VALUES on entering NO_PURGE"
                STATE_CP        = NO_PURGE
```

```
elseif          STATE_CDN_CP = CDN_NO_PURGE
then      call  "INITIALIZATION-VALUES on leaving RAMP_OPEN_FAST"
          call  "INITIALIZATION-VALUES on entering RAMP_TO_NO_PURGE"
          STATE_CP        = RAMP_TO_NO_PURGE

elseif          STATE_CDN_CP = CDN_MIN_PURGE
      or  STATE_CDN_CP = NO_CDN
      or  LV_CP_NEW_RAMP_OPEN = 1
      or  LV_CP_NEW_RAMP_OPEN_FAST = 1
then      call  "INITIALIZATION-VALUES on leaving RAMP_OPEN_FAST"
          call  "INITIALIZATION-VALUES on entering MIN_PURGE"
          STATE_CP        = MIN_PURGE

elseif          LV_CP_ENA       = 0
then      call  "Storage of Values for OBDII EVAP-System Monitoring"
          call  "INITIALIZATION-VALUES on leaving RAMP_OPEN_FAST"
          call  "INITIALIZATION-VALUES on entering RAMP_CLOSE"
          STATE_CP        = RAMP_CLOSE

elseif         [ FLOW_CTL_CPS >= FLOW_TAR_CPS
      or  MFF_CP >= MFF_MAX_CP ]
      and LV_FLOW_TAR_COR_CP = 0
      then call "INITIALIZATION-VALUES on leaving RAMP_OPEN_FAST"
          call  "INITIALIZATION-VALUES on entering MAX_PURGE"
          STATE_CP        = MAX_PURGE

else      STATE_CP        = RAMP_OPEN_FAST    // stay in current state
endif
end
```

28.7.2.10 MAX_PURGE

```
if        LV_CP_ACT       = 0
then      STATE_CP        = CP_NOT_ACT

elseif          STATE_CDN_CP = CDN_NO_FAST
then      call  "INITIALIZATION-VALUES on leaving MAX_PURGE"
          call  "INITIALIZATION-VALUES on entering NO_PURGE"
          STATE_CP        = NO_PURGE

elseif          STATE_CDN_CP = CDN_NO_PURGE
then      call  "INITIALIZATION-VALUES on leaving MAX_PURGE"
          if    LV_CP_CLOSE_REQ
          then  call "Storage of Values for OBDII EVAP-System Monitoring"
          endif
          call  "INITIALIZATION-VALUES on entering RAMP_TO_NO_PURGE"
          STATE_CP        = RAMP_TO_NO_PURGE
```

```
elseif          STATE_CDN_CP = CDN_MIN_PURGE
      and [ LV_CL_MDL_ENA = 0                          //  CL-mdl not enabled
      or    LV_CP_CDN_MIN_PURGE_FAST = 1 ]          // "hard" conditions for MIN_PURGE
      or  [ STATE_CDN_CP = NO_CDN                  // NO_CDN-management
      and   LC_CP_TRA_MAX_MIN_CDN_ENA ]
      or    LV_CP_NEW_RAMP_OPEN = 1
      or    LV_CP_NEW_RAMP_OPEN_FAST = 1
      or  [ LV_CL_MDL_CAT_PURGE_ACT_CP = 1
      and   { LV_CL_MDL_ACT = 0 or LV_CP_CAT_PURGE = 0} ]
then      call  "INITIALIZATION-VALUES on leaving MAX_PURGE"
          call  "INITIALIZATION-VALUES on entering MIN_PURGE"
          STATE_CP       = MIN_PURGE

elseif          LV_CP_ENA      = 0
then      call  "Storage of Values for OBDII EVAP-System Monitoring"
          call  "INITIALIZATION-VALUES on leaving MAX_PURGE"
          call  "INITIALIZATION-VALUES on entering RAMP_CLOSE"
          STATE_CP       = RAMP_CLOSE

else      STATE_CP       = MAX_PURGE                   // stay in current state
endif
end
```

28.7.2.11 RAMP_CLOSE

```
if        LV_CP_ACT      = 0
then      STATE_CP       = CP_NOT_ACT

elseif          STATE_CDN_CP = CDN_NO_FAST
then      call  "INITIALIZATION-VALUES on leaving RAMP_CLOSE"
          call  "INITIALIZATION-VALUES on entering NO_PURGE"
          STATE_CP       = NO_PURGE

elseif          FLOW_CTL_CPS = 0                    // RAMP_CLOSE finished
      or  STATE_CDN_CP = CDN_NO_PURGE
      or  STATE_CDN_CP = CDN_MIN_PURGE
      then call  "INITIALIZATION-VALUES on leaving RAMP_CLOSE"
          call  "INITIALIZATION-VALUES on entering RAMP_TO_NO_PURGE"
          STATE_CP       = RAMP_TO_NO_PURGE

else      STATE_CP       = RAMP_CLOSE          // stay in current state
endif
```

28.7.2.12 INITIALIZATIONS on transitions

The following submodules are called on transitions between the different states of canister purge control

28.7.2.12.1 Storage of Values for OBDII EVAP-System Monitoring (RO, ROF, MaxP)

This submodule is called, if RAMP_OPEN, RAMP_OPEN_FAST or MAX_PURGE is left due to purge time elapsed (LV_CP_ENA = 0).

In this case, the last calculated value of the canister load CL_MMV is stored as CL_MMV_CLC_END before RAMP_CLOSE operation starts. This value is used as input for the OBDII EVAP-System Monitoring.

In addition, this subroutine may be called if MAX_PURGE is left due to a valve closing request from OBD II function.

```
CL_MMV_CLC_END = CL_MMV
end
```

28.7.2.12.2 INITIALIZATION-VALUES on entering NO_PURGE

This submodule is called at every possible transition from any state to NO_PURGE except the transition CP_NOT_ACT → NO_PURGE.

On entering NO_PURGE, the initialisations listed below are done.

```
// Initialisation of the time counter for lambda controller stabilisation
FLOW_GRD_CPS          = 0
FLOW_TAR_CPS          = 0
FLOW_CTL_CPS          = 0
LV_CP_CLL             = 0
LV_STATE_MEM_CP_CLL   = 0
PQ_CP_SP              = 1           // In this mode no forced PQ is required
end
```

28.7.2.12.3 INITIALIZATION-VALUES on leaving NO_PURGE

If the NO_PURGE operation is left towards any other state, the following initialisations are done.

```
---
end
```

28.7.2.12.4 INITIALIZATION-VALUES on entering RAMP_TO_NO_PURGE

This submodule is called at every possible transition from any state to RAMP_TO_NO_PURGE.

On entering RAMP_TO_NO_PURGE, the initialisations listed below are done.

As cl is not updated during RAMP_TO_NO_PURGE, it’s sufficient to define the correction factor for closing speed FLOW_FAC_CP on transition from any state to RAMP_TO_NO_PURGE.

The start value of FLOW_CTL_CPS for the ramp has to be defined.

```
LV_CP_CLL          = 0
FLOW_TAR_CPS       = 0
FLOW_CTL_CPS       = min [FLOW_CTL_CPS, FLOW_MAX_CPS]
PQ_CP_SP           = 1                       // In this mode no forced PQ is required
if   STATE_CP      = MIN_PURGE
then FLOW_FAC_CP   = 1                       // ramp decrement cor. factor
else FLOW_FAC_CP   = IP_FLOW_FAC_CP (CL)
endif
end
```

28.7.2.12.5 INITIALIZATION-VALUES on leaving RAMP_TO_NO_PURGE

If the RAMP_TO_NO_PURGE operation is left towards any other state, the following initialisations are done.

```
---
end
```

28.7.2.12.6 INITIALIZATION-VALUES on entering MIN_PURGE

This submodule is called at every possible transition from any state to MIN_PURGE.
On entering MIN_PURGE, the initialisations listed below are done.
The start value of FLOW_CTL_CPS for the ramp has to be defined.

```
LV_CP_CLL          = 0
FLOW_CTL_CPS       = min [FLOW_CTL_CPS, FLOW_MAX_CPS]
PQ_CP_SP           = 1                  // In this mode no forced PQ is required
end
```

28.7.2.12.7 INITIALIZATION-VALUES on leaving MIN_PURGE

If the MIN_PURGE operation is left towards any other state, the following initialisations are done.

```
---
end
```

28.7.2.12.8 INITIALIZATION-VALUES on entering WAIT_RAMP_OPEN

This submodule is called at every possible transition from any state to WAIT_RAMP_OPEN.
On entering WAIT_RAMP_OPEN, the initialisations listed below are done.

```
T_DI_CP                  = 0          // reset of time counter for disabled normal purge
STATE_CLL_DEAC_CP        = CP_NO_DEAC
```

```
T_DLY_CP            = C_T_CP_PURGE_TUBE+
                IP_T_DLY_CP(MAF_KGH_FG_PRED_COR_CP) *
                C_FAC_T_DLY_PURGE_TUBE_CP
MAF_INT_CP          = 0
T_MAF_INT_CP        = 0
end
```

28.7.2.12.9 INITIALIZATION-VALUES on leaving WAIT_RAMP_OPEN

If the WAIT_RAMP_OPEN operation is left towards any other state, the following initialisations are done.

```
end
```

28.7.2.12.10 INITIALIZATION-VALUES on entering RAMP_OPEN

This submodule is called at every possible transition from any state to RAMP_OPEN.

If the RAMP_OPEN operation was disabled (Conditions for RAMP_OPEN were not fulfilled) for a time < C_T_DI_RAMP_OPEN_CP_0/_1 (T_DI_CP > 0, STATE_CLL_DEAC_CP = RAMP_OPEN_DEAC), initialisations for disabled RAMP_OPEN are done, otherwise the standard initialisations for RAMP_OPEN are done.

```
if      STATE_CLL_DEAC_CP       = RAMP_OPEN_DEAC
then                            // Initialisations for disabled RAMP_OPEN
        FLOW_CTL_CPS            = FLOW_CTL_CPS_DI
else                            // Standard-initialisations for RAMP_OPEN
        FLOW_CTL_CPS            = C_FLOW_INI_CPS
endif
LV_CP_CLL                       = 1
LV_CP_RAMP_OPEN_ACT             = 1
LV_DLY_GAS_CP                   = 0
LV_CP_AFL_IT                    = 1                     // lean
STATE_CLL_DEAC_CP               = CP_NO_DEAC
T_DI_CP                         = 0
if      STATE_OPM_ENG_CP = ENG_MODE_0
then    T_DLY_CP_OPEN                   =
        max[IP_T_DLY_CP (MAF_KGH_FG_PRED_COR_CP),1hex]
else    T_DLY_CP_OPEN                   =
        max[IP_T_DLY_CP_2 (MAF_KGH_FG_PRED_COR_CP),1hex]
endif
FLOW_CPS                        = FLOW_CTL_CPS
FLOW_CPS_OLD_1                  = FLOW_CTL_CPS
FLOW_CPS_OLD_2                  = FLOW_CTL_CPS
FLOW_DLY_OLD_1_CP               = FLOW_CTL_CPS
FLOW_DLY_OLD_2_CP               = FLOW_CTL_CPS
MAF_CP_OLD_1                    = MAF_CP
MAF_CP_OLD_2                    = MAF_CP
MAF_DLY_OLD_1_CP                = MAF_CP
```

```
MAF_DLY_OLD_2_CP              = MAF_CP
MAF_CPS_OLD_1                 = MAF_CPS
MAF_CPS_OLD_2                 = MAF_CPS
MAF_DLY_OLD_1_CPS             = MAF_CPS
MAF_DLY_OLD_2_CPS             = MAF_CPS
end
```

28.7.2.12.11 INITIALIZATION-VALUES on leaving RAMP_OPEN

If the RAMP_OPEN operation is left towards any other state, the following actual values for FLOW_CTL_CPS_DI are stored before the interruption of the operation:

```
LV_CP_RAMP_OPEN_ACT           = 0
LV_STATE_MEM_CP_CLL           = 1
FLOW_CTL_CPS_DI               = FLOW_CTL_CPS
if     (LV_CP_NEW_RAMP_OPEN = 0
       and  LV_CP_NEW_RAMP_OPEN_FAST = 0)
and    STATE_LOCK_CP          = UNLOCKED
then   STATE_CLL_DEAC_CP      = RAMP_OPEN_DEAC
       if    STATE_OPM_ENG_CP = ENG_MODE_0
       then  T_DI_CP          = C_T_DI_RAMP_OPEN_CP_0
       else  T_DI_CP          = C_T_DI_RAMP_OPEN_CP_1
       endif
else   if    STATE_LOCK_CP    ≠ UNLOCKED
       then  T_LOCK_CP_TMP    = C_T_LOCK_CP
       endif
endif
end
```

28.7.2.12.12 INITIALIZATION-VALUES on entering RAMP_OPEN_FAST

This submodule is called at every possible transition from any state to RAMP_OPEN_FAST.

If the RAMP_OPEN_FAST operation was disabled (Conditions for RAMP_OPEN were not fulfilled) for a time < C_T_DI_RAMP_FAST_CP_0/_1 (T_DI_CP > 0, STATE_CLL_DEAC_CP = RAMP_FAST_DEAC), initialisations for disabled RAMP_OPEN_FAST are done, otherwise the standard initialisations for RAMP_OPEN are done.

```
if     STATE_CLL_DEAC_CP      = RAMP_FAST_DEAC
then                          // Initialisations for disabled RAMP_OPEN_FAST
       FLOW_CTL_CPS           = FLOW_CTL_CPS_DI
else                          // Standard-initialisations for RAMP_OPEN_FAST
       FLOW_CTL_CPS           = C_FLOW_INI_CPS
endif
LV_CP_CLL                     = 1
LV_CP_RAMP_OPEN_ACT           = 1
LV_DLY_GAS_CP                 = 0
```

```
LV_CP_AFL_IT                 = 1                    // lean
STATE_CLL_DEAC_CP            = CP_NO_DEAC
T_DI_CP                      = 0
if     STATE_OPM_ENG_CP = ENG_MODE_0
then   T_DLY_CP_OPEN                  =
       max[IP_T_DLY_CP (MAF_KGH_FG_PRED_COR_CP),1hex]
else   T_DLY_CP_OPEN                  =
       max[IP_T_DLY_CP_2 (MAF_KGH_FG_PRED_COR_CP),1hex]
endif
FLOW_CPS                     = FLOW_CTL_CPS
FLOW_CPS_OLD_1               = FLOW_CTL_CPS
FLOW_CPS_OLD_2               = FLOW_CTL_CPS
FLOW_DLY_OLD_1_CP            = FLOW_CTL_CPS
FLOW_DLY_OLD_2_CP            = FLOW_CTL_CPS
MAF_CP_OLD_1                 = MAF_CP
MAF_CP_OLD_2                 = MAF_CP
MAF_DLY_OLD_1_CP             = MAF_CP
MAF_DLY_OLD_2_CP             = MAF_CP
MAF_CPS_OLD_1                = MAF_CPS
MAF_CPS_OLD_2                = MAF_CPS
MAF_DLY_OLD_1_CPS            = MAF_CPS
MAF_DLY_OLD_2_CPS            = MAF_CPS
end
```

28.7.2.12.13 INITIALIZATION-VALUES on leaving RAMP_OPEN_FAST

If the RAMP_OPEN_FAST operation is left towards any other state, the following initialisations are done.

```
LV_CP_RAMP_OPEN_ACT          = 0
LV_STATE_MEM_CP_CLL          = 1
FLOW_CTL_CPS_DI              = FLOW_CTL_CPS
if     (LV_CP_NEW_RAMP_OPEN = 0
       and   LV_CP_NEW_RAMP_OPEN_FAST = 0 )
then   STATE_CLL_DEAC_CP      = RAMP_FAST_DEAC
       if     STATE_OPM_ENG_CP = ENG_MODE_0
       then T_DI_CP                   = C_T_DI_RAMP_FAST_CP_0
       else T_DI_CP                   = C_T_DI_RAMP_FAST_CP_1
       endif
endif
end
```

28.7.2.12.14 INITIALIZATION-VALUES on entering MAX_PURGE

This submodule is called at every possible transition from any state to MAX_PURGE.

On entering MAX_PURGE, the initialisations listed below are done.

If the MAX_PURGE operation was disabled (Conditions for MAX_PURGE were not fulfilled) for a time t < C_T_DI_MAX_CP_0/_1 (T_DI_CP > 0, STATE_CLL_DEAC_CP = MAX_PURGE_DEAC), the initialisations for disabled MAX_PURGE are done additionally to the standard initialisations for MAX_PURGE.

```
if      STATE_CLL_DEAC_CP        = MAX_PURGE_DEAC
then    FLOW_CTL_CPS             = MIN (FLOW_CTL_CPS_DI;
                                   FLOW_MAX_CPS * C_FAC_FLOW_MAX_INI_CP;
                                   MFF_SP_MV * C_MFF_RATIO_FLOW_INI_CP
                                   * [N_32 * NC_CYL_NR * 3) / $10^5$ ] / CL_MMV)
                                   // [mg/stk] -> [kg/h]
else    ACTION_EVAM_Purge ()
endif
// standard initialisations
LV_CP_CLL                        = 1
STATE_CLL_DEAC_CP                = CP_NO_DEAC
T_DI_CP                          = 0
if          STATE_CP = RAMP_OPEN
then        T_DLY_REQ_CP_TMP         = C_T_DLY_REQ_CP
elseif          STATE_CP = NO_PURGE
            and LV_CP_CAT_PURGE = 1
            and LV_CL_MDL_ENA = 1
then        LV_CL_MDL_CAT_PURGE_ACT_CP = 1
            FLOW_CTL_CPS         = MIN (FLOW_MAX_CPS * C_FAC_FLOW_MAX_INI_CP;
                                   MFF_SP_MV * C_MFF_RATIO_FLOW_INI_CP
                                   * [N_32 * NC_CYL_NR * 3) / $10^5$ ] / CL_MMV)
                                   * C_FAC_FLOW_CAT_PURGE_INI_CP
                                   // [mg/stk] -> [kg/h]
endif

// Additionnal initialization in case of RAMP_OPEN_FAST to MAX_PURGE transition:
if      STATE_CP = RAMP_OPEN_FAST
then    FLOW_CPS_OLD_1           = FLOW_CPS
        FLOW_CPS_OLD_2           = FLOW_CPS
        FLOW_DLY_OLD_1_CP        = FLOW_CPS
        FLOW_DLY_OLD_2_CP        = FLOW_CPS
endif
end
```

28.7.2.12.15 INITIALIZATION-VALUES on leaving MAX_PURGE

If the MAX_PURGE operation is disabled or ends, the following actual values for FLOW_CTL_CPS_DI are stored before the interruption of the operation:

```
LV_STATE_MEM_CP_CLL              = 1
FLOW_CTL_CPS_DI                  = FLOW_CTL_CPS
T_DLY_REQ_CP_TMP                 = 0
if      (LV_CP_NEW_RAMP_OPEN     = 0
```

```
        and   LV_CP_NEW_RAMP_OPEN_FAST = 0
        and   LV_CL_MDL_CAT_PURGE_ACT_CP = 0)
  then  STATE_CLL_DEAC_CP            = MAX_PURGE_DEAC
        if    STATE_OPM_ENG_CP = MODE_0
        then  T_DI_CP                = C_T_DI_MAX_CP_0
        else  T_DI_CP                = C_T_DI_MAX_CP_1
        endif
  endif
  LV_CL_MDL_CAT_PURGE_ACT_CP         = 0
  end
```

28.7.2.12.16 INITIALIZATION-VALUES on entering RAMP_CLOSE

This submodule is called at every possible transition from any state to RAMP_CLOSE.

On entering RAMP_CLOSE, the initialisations listed below are done.

The Initialization value of FLOW_CTL_CPS for RAMP_CLOSE is the last calculated value of min (FLOW_CTL_CPS, FLOW_MAX_CPS) if the preceding operation was MAX_PURGE. If the RAMP_OPEN operation was preceding the Initialization value is the last calculated value of FLOW_CTL_CPS of the RAMP_OPEN operation.

```
STATE_CLL_DEAC_CP       = CP_NO_DEAC
T_DI_CP                 = 0
PQ_CP_SP                = 1              // In this mode no forced PQ is required
FLOW_TAR_CPS            = 0
if    STATE_CP          = MAX_PURGE
then  FLOW_CTL_CPS      = min [FLOW_CTL_CPS, FLOW_MAX_CPS]
endif
end
```

28.7.2.12.17 INITIALIZATION-VALUES on leaving RAMP_CLOSE

If the RAMP_CLOSE operation is left towards any other state, the following initialisations are done.

```
LV_STATE_MEM_CP_CLL = 1
end
```

28.7.3 Canister purge management states

28.7.3.1 NO_PURGE

General information:

In the NO_PURGE state of the Evaporative Emission Control no calculation of the CPS flow set point flow_ctl_cps, the fuel flow from the activated charcoal filter (ACF) MFF_CP, the Canister Load cl and the additive fuel flow for injection correction will be done.

Formula section:

```
call  "Time counter for disabled RAMP_OPEN or MAX_PURGE"
end
```

28.7.3.2 RAMP_TO_NO_PURGE

Formula section:

```
FLOW_GRD_CPS = IP_FLOW_DEC_CP (MFF_SP_MV) * FLOW_FAC_CP
if          LV_CP_CMU_CMB
then        FLOW_GRD_CPS = FLOW_GRD_CPS * C_FAC_DEC_COR_CMU_CP
elseif          LV_CP_CLOSE_1
then        FLOW_GRD_CPS = FLOW_GRD_CPS * C_FAC_DEC_COR_1_CP
elseif          LV_CP_CLOSE_2
then        FLOW_GRD_CPS = FLOW_GRD_CPS * C_FAC_DEC_COR_2_CP
endif
call  "FLOW_CTL_CPS ramping"
call  "Time counter for disabled RAMP_OPEN or MAX_PURGE"
end
```

28.7.3.3 MIN_PURGE

General information:

In the MIN_PURGE state, the CPS flow is calculated from calibration datas depending on combustion mode.

```
FLOW_TAR_CPS = FLOW_CPS_SP_MIN_PURGE * FAC_FLOW_TAR_COR_CP
if    FLOW_CTL_CPS$_{n-1}$ > FLOW_TAR_CPS                // actual value higher than SP
then  FLOW_GRD_CPS = IP_FLOW_DEC_CP (MFF_SP_MV)// decreasing flow
else  if LV_HOM_ACT = 1
      then FLOW_GRD_CPS = IP_FLOW_INC_CP (MFF_SP_MV)
                       * IP_FLOW_INC_COR_HOM_CP (MAF_CYL) // increasing flow
      else FLOW_GRD_CPS = IP_FLOW_INC_CP (MFF_SP_MV)   // increasing flow
      endif
      LV_STATE_MEM_CP_CLL = 0               // transition from normal purge finished
endif
call  "FLOW_CTL_CPS ramping"
call  "Time counter for disabled RAMP_OPEN or MAX_PURGE"
end
```

28.7.3.4 WAIT_RAMP_OPEN

General information:

The WAIT_RAMP_OPEN state is reached if the conditions for normal entry in RAMP_OPEN or RAMP_OPEN_FAST are given but the time counter for lambda controller stabilisation is not yet elapsed or CPS adaptation is active. It can only be reached by NO_PURGE.

To speed up the reaction of the lambda sensor in RAMP_OPEN or RAMP_OPEN_FAST, the purge tube may be purged itself for a short instance in WAIT_RAMP_OPEN. Using this functionality, the purge valve is opened rapidly and wide for a very short time to purge the diluted gas in the tubes and fill the purge tubes with "fresh" gas from the ACF.

For the calcuation of the air mass flow integral the unit conversion:
[g] = [kg/h] * [ms] * [(g*h) / (kg*ms)] / 3600
has to be considered.

Formula section:

```
if      LV_CPS_AD_ACT = 0
then
MAF_INT_CP = MAF_INT_CP + (MAF_CYL * TA / 3600)  // MAF Integral for Lambda
                                                       controller stabilisation
T_MAF_INT_CP = T_MAF_INT_CP + TA  // Total time of the MAF integration
                                              for min/max limitation
T_DLY_CP = max [$T\_DLY\_CP_{n-1}$ – TA, 0]         // Purge tube  time counter
// "purging the purge tube"
endif
 if     T_MAF_INT_CP          < C_T_CP_PURGE_TUBE
then
        FLOW_TAR_CPS        = C_FLOW_CPS_SP_PURGE_TUBE
        FLOW_GRD_CPS        = C_FLOW_CPS_GRD_PURGE_TUBE
else
        FLOW_TAR_CPS        = 0
        FLOW_GRD_CPS        = C_FLOW_CPS_GRD_PURGE_TUBE_CLOSE
endif
call   “FLOW_CTL_CPS ramping”
end
```

28.7.3.5 RAMP_OPEN

General information:

After a reset, an engine stall or if a certain time since the last MAX_PURGE operation has passed, the load cl of the canister is considered as not known. In this case the canister load value cl is initialized to a constant value c_cl_st_cp and the CPS is opened slowly, using a ramp function. So a high deviation of the lambda control shall be prevented.

The RAMP_OPEN operation is finished when the fuel flow MFF_CP is greater than a maximum value MFF_MAX_CP or when the CL_MMV value stay in a dynamic window. Then the MAX_PURGE operation starts, if the cl is above a threshold, otherwise the reaction of the system is configurable.

At the end of the RAMP_OPEN operation the remaining time of purge is reevaluated versus Canister Load cl is reevaluated.

Formula section:

```
call “calculation of PQ_CP_SP”                        (common)
call “calculation of the flow setpoint”               (common)
end
```

28.7.3.6 RAMP_OPEN_FAST Operation

General information:

After a short interruption of the MAX_PURGE operation the load cl of the ACF stay around the last value calculated. In this case the flow_ctl_cps is calculated as a function of the actual FLOW_CTL_CPS value and the objective value.

The RAMP_OPEN_FAST operation is finished. when one of the following conditions is set:

- FLOW_CTL_CPS is greater or equal to the value FLOW_TAR_CPS
- The fuel flow MFF_CP is greater than a maximum value MFF_MAX_CP

Then the MAX_PURGE operation starts. If the purge time is elapsed (LV_CP_ENA=0), the RAMP_OPEN_FAST operation ends. the RAMP_CLOSE operation starts immediately.

At the end of the RAMP_OPEN_FAST operation the purge time is reevaluated (see chapter “time scheduler”).

Formula section:

```
PQ_CP_SP =1            //  In this mode no forced PQ is required by canister purge
call “calculation of the flow setpoint”                     (common)
end
```

28.7.3.7 MAX_PURGE Operation

General information:

The MAX_PURGE operation is started when the RAMP_OPEN operation or the RAMP_OPEN_FAST operation is finished.

The MAX_PURGE operation ends when the purge time is elapsed. Then the RAMP_CLOSE operation starts.

During MAX_PURGE, the time counter T_DLY_MAX_CP is initialised to C_T_DLY_MAX_CP_0/_1. When the MAX_PURGE operation ends or when the MAX_PURGE is disabled, the time T_DLY_MAX_CP starts to count backwards down to 0.

Calculation of FLOW_CTL_CPS for MAX_PURGE

The value of FLOW_CTL_CPS during MAX_PURGE operation is calculated as a function of the filtered canister load CL_MMV and the maximum allowable fuel flow from the CPS MFF_MAX_CP (see flow setpoints module).

Formula section:

```
call “calculation of PQ_CP_SP”                               (common)
call “Calculation of LV_T_DLY_REQ_CP”
FLOW_TAR_CPS = min [MFF_MAX_CP / CL_MMV, FLOW_MAX_CPS] *
                                                  FAC_FLOW_TAR_COR_CP
if    $FLOW\_CTL\_CPS_{n-1}$ > FLOW_TAR_CPS                  // actual value higher than SP
then  FLOW_GRD_CPS = IP_FLOW_CPS_DEC (MFF_SP_MV)      // decreasing flow
else  if LV_HOM_ACT = 1
```

```
        then FLOW_GRD_CPS = IP_FLOW_CPS_INC (MFF_SP_MV)
                                * IP_FLOW_INC_COR_HOM_CP (MAF_CYL)// increasing flow
        else FLOW_GRD_CPS = IP_FLOW_CPS_INC (MFF_SP_MV)  // increasing flow
        endif
    endif
    (gradient limitation moved here from flow calculation chain)
    call   "FLOW_CTL_CPS ramping"
    end
```

28.7.3.7.1 Calculation of LV_T_DLY_REQ_CP

This management is usefull esssentially during an emision cycle. The aim is to stop the purge in case of request from the catalyst diagnosis (or an other OBD2 diagnosis), one time during the cycle.

LV_T_DLY_REQ_CP is set C_T_DLY_REQ_CP seconds after a Ramp_Open and stay to 1 until next trip or engine stalling.

The flag is set after having been for a time C_T_DLY_REQ_CP in an un-interrupted MAX_PURGE coming from RAMP_OPEN and stay to 1 until next trip or engine stalling .

When coming from any other state than RAMP_OPEN, the flag is not calculated.

Signal flow diagram:

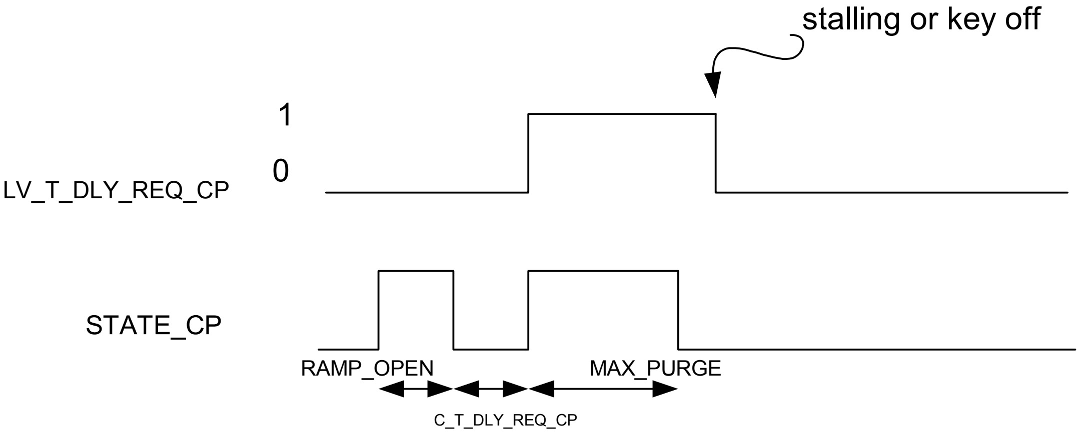


figure 4: LV_T_DLY_REQ_CP

Formula section:

```
if      T_DLY_REQ_CP_TMP > 1h
then    T_DLY_REQ_CP_TMP = max (1h, T_DLY_REQ_CP_TMP – TA)
endif
if      T_DLY_REQ_CP_TMP = 1h
then    LV_T_DLY_REQ_CP    = 1
endif
end
```

28.7.3.8 RAMP_CLOSE Operation

General information:

The RAMP_CLOSE operation starts if the purge time is elapsed.

The RAMP_CLOSE operation ends when the value of FLOW_CTL_CPS is equal to 0.

Calculation of FLOW_CTL_CPS for RAMP_CLOSE

When the RAMP_CLOSE operation is activated FLOW_CTL_CPS is decremented. The decrement value depends on the map values of IP_FLOW_DEC_CP. The closing speed of the CPS is so controlled by the mass fuel flow setpoint MFF_SP_MV.

Formula section:

```
FLOW_GRD_CPS = IP_FLOW_DEC_CP (MFF_SP_MV) * IP_FLOW_FAC_CP (CL)
call   “FLOW_CTL_CPS ramping”
end
```

28.7.4 Common submodules

The keywords in brackets at the end of the headline of each following submodule indicates, by which STATE_CP the corresponding submodule is called. If STATE_CP = CP_NOT_ACT, none of the following submodules is called. The following abbreviations are used:

NP	NO_PURGE
RTNP	RAMP_TO_NO_PURGE
MinP	MIN_PURGE
WRO	WAIT_RAMP_OPEN
RO	RAMP_OPEN
ROF	RAMP_OPEN_FAST
MaxP	MAX_PURGE
RC	RAMP_CLOSE

28.7.4.1 Time counter for disabled RAMP_OPEN or MAX_PURGE (NP, RTNP, MinP)

General information:

This calculation is called in NO_PURGE, RAMP_TO_NO_PURGE and MIN_PURGE.

If the RAMP_OPEN or the MAX_PURGE operation was interrupted (NO_PURGE, RAMP_TO_NO_PURGE or MIN_PURGE active) and the time counter T_DI_CP is greater than zeor which means that the interrupt-time is lower than the adjustable times C_T_DI_RAMP_OPEN_CP_0/_1 (RAMP_OPEN) or C_T_DI_MAX_CP_0/_1 (MAX_PURGE), a direct switch back to the previous state RAMP_OPEN or MAX_PURGE is possible, if the conditions for this previous state are once again given.

Otherwise (time counter T_DI_CP reached zero), NORMAL_PURGE has to restart with RAMP_OPEN.

TA = recurrence

Formula section:

```
If      (LV_CP_NEW_RAMP_OPEN = 0
        and LV_CP_NEW_RAMP_OPEN_FAST = 0
then    $T\_DI\_CP_n$                  = max [0, $T\_DI\_CP_{n-1}$ – TA]
else    T_DI_CP = 0
endif
$T\_LOCK\_CP_n$                  = max [0, $T\_LOCK\_CP_{n-1}$ – TA]
if      T_DI_CP                = 0
then    STATE_CLL_DEAC_CP = CP_NO_DEAC
endif
if      T_LOCK_CP = 0
then    STATE_LOCK_CP          = UNLOCKED
endif
end
```

28.7.4.2 Calculation of PQ_CP_SP (RO, MaxP)

General information:

This calculation is called only in Ramp_Open and MAX_PURGE, and available only in stratified and homogeneous lean burn modes.

In this mode a forced PQ is required by canister purge in order to increase PRS_CPS and to purge with more efficiency. IP_PQ_CP_SP is depending on engine speed and torque setpoint.

Formula section:

```
if      [(LV_HOM_ACT = 0)
or      (LV_HOM_AFL_ACT = 1 and LC_PQ_SP_CP_HOM_AFL=1) ]
then    if      (CL_MMV > C_CL_MMV_PQ)
        then  PQ_SP_CP_TMP = IP_PQ_CP_SP (N_32, TQI_SP_S)
        else  PQ_SP_CP_TMP = 1
        endif
        $PQ\_CP\_SP_n = PQ\_CP\_SP_{n-1}$ +
                    (PQ_SP_CP_TMP - $PQ\_CP\_SP_{n-1}$) * C_CRLC_PQ_SP_CP
else    $PQ\_CP\_SP_n$ = 1
endif
```

28.7.4.3 Calculation of the flow setpoint (RO, ROF)

General information:

This calculation is called in RAMP_OPEN and RAMP_OPEN_FAST.

The opening of the CPS is controlled by the value of FLOW_CTL_CPS. This value describes the flow requested through the CPS. So a too high deviation of the lambda control at transient engine operation should be avoided. The opening sequence is as follows:

- When entering RAMP_OPEN or RAMP_OPEN_FAST, FLOW_CTL_CPS is initialised

- After start or re-start of RAMP_OPEN or RAMP_OPEN_FAST, the system waits for the gas transfer time between CPS valve and upstream lambda sensor (IP_T_DLY_CP).
- When this delay is elapsed or when using a binary lambda control and a transition rich to lean occurs, the next period of increment steps starts. The number of increment steps for this phase is defined from ID_FLOW_INC_NR_CP. This number is defined once at the beginning of the phase. For each increment the FLOW_CTL_CPS increase is fixed by IP_FLOW_INC_CP corrected by IP_FLOW_FAC_CP. The opening speed of the CPS is so controlled by the mass air flow MAF_CYL and the load CL of the activated charcoal canister ACF to avoid lambda regulation deviation.
- After the increment steps are carried out, the function waits again for gas transfer time or binary lambda sensor rich-lean transition.
- FLOW_CTL_CPS is limited to FLOW_TAR_CPS, the target flow setpoint.

The Pjump used depends of the bank selected (see canister load and lambda deviation module). The Pjump is related to gas enrichment.

The aim is to increase the FLOW_CTL_CPS with incremented values to prevent unexpected lambda deviation due to the lambda controller.

The increment number ID_FLOW_INC_NR_CP carried out is tuneable and depending on engine speed.

The time delay will allow to synchronize the flow variation, the gas transfer and the lambda control. As long as the maximum flow FLOW_TAR_CPS is not reached the sequence will start over.

Signal flow diagram:

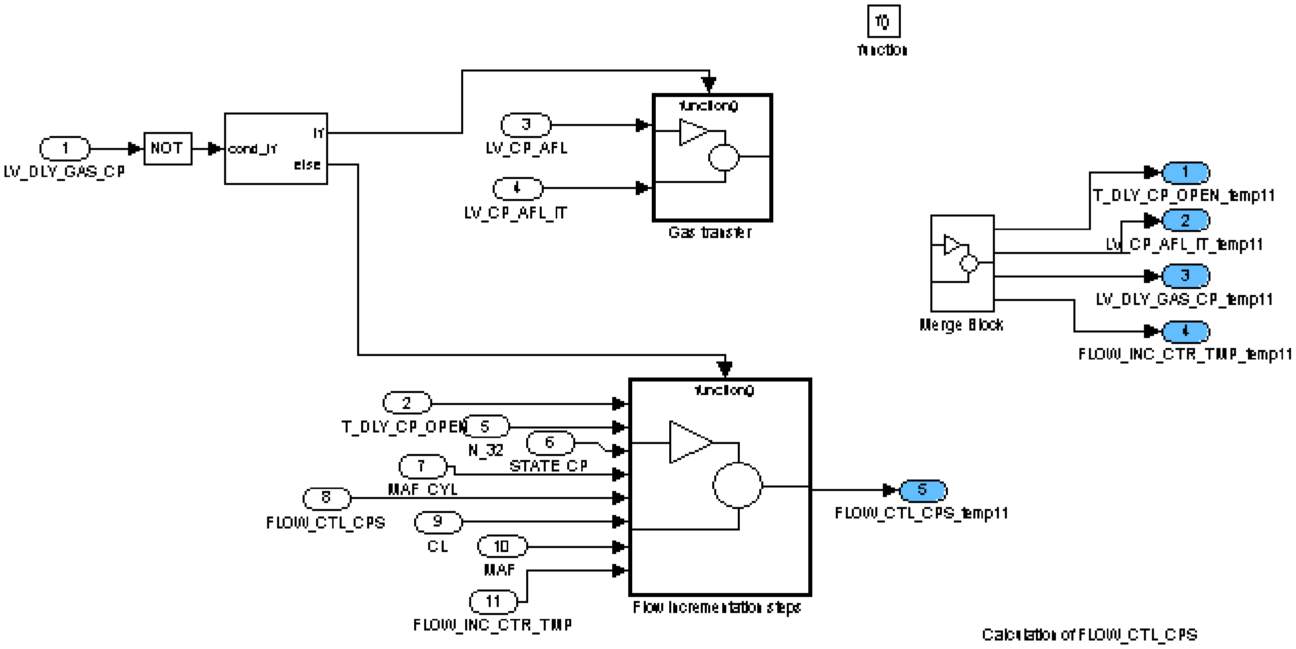


figure 6: calculation of FLOW_CTL_CPS for RAMP_OPEN_FAST

Formula section:

```
if      LV_DLY_GAS_CP = 0
then call   "Gas transfer"                    // waiting for gas transfer timer
```

```
else  call  ”Flow incrementation steps”
endif
end
```

28.7.4.3.1 Gas transfer

```
T_DLY_CP_OPEN      = max [ 0, T_DLY_CP_OPEN - TA]
if    LC_LAM_LIN_CPS = 0                                // binary lambda control
then  LV_CP_AFL_IT  = LV_CP_AFL
else  LV_CP_AFL_IT  = 1                                 // lean

endif
if    T_DLY_CP_OPEN = 0
or    LV_CP_AFL_IT  0 → 1          // rich → lean
then  LV_DLY_GAS_CP            = 1
      FLOW_INC_CTR_TMP         = ID_FLOW_INC_NR_CP (N_32)
endif
end
```

28.7.4.3.2 Flow incrementation steps

```
if    STATE_CP = RAMP_OPEN
then  if LV_HOM_ACT = 1
      then FLOW_GRD_CPS = IP_FLOW_INC_CP (MFF_SP_MV) *
                  IP_FLOW_INC_COR_HOM_CP (MAF_CYL) // RO
      else  FLOW_GRD_CPS = IP_FLOW_INC_CP (MFF_SP_MV) // RO
      endif
else  if LV_HOM_ACT = 1
      then  FLOW_GRD_CPS = IP_FLOW_INC_FAST_CP (MFF_SP_MV) *
                  IP_FLOW_INC_COR_HOM_CP (MAF_CYL) // ROF
      else FLOW_GRD_CPS = IP_FLOW_INC_FAST_CP (MFF_SP_MV) // ROF
      endif
endif
FLOW_GRD_CPS = FLOW_GRD_CPS * IP_FLOW_FAC_CP (CL)

if    STATE_CP = RAMP_OPEN_FAST
and   LC_CP_FLOW_LIM_OPEN_FAST = 0
then  FLOW_TAR_CPS       = min [MFF_MAX_CP / CL_MMV, FLOW_MAX_CPS]
else  [If STATE_OPM_ENG_CP = ENG_MODE_0
         then  FLOW_TAR_CPS       = IP_FLOW_MAX_OPEN_CPS (N_32, MAF)
                                           * FAC_FLOW_COR_CLL_CP
         else  FLOW_TAR_CPS       = IP_FLOW_MAX_OPEN_CPS_2 (N_32,
                                     MFF_SP_MV) * FAC_FLOW_COR_CLL_CP
      endif]
```

```
endif
FLOW_TAR_CPS = FLOW_TAR_CPS * FAC_FLOW_TAR_COR_CP
call   "FLOW_CTL_CPS ramping"
FLOW_INC_CTR_TMP              = max [ 0, FLOW_INC_CTR_TMP – 1]

if     FLOW_INC_CTR_TMP        = 0
then   // gas delay timer init
       LV_DLY_GAS_CP           = 0
       If STATE_OPM_ENG_CP = ENG_MODE_0
       then T_DLY_CP_OPEN            =
              max[IP_T_DLY_CP (MAF_KGH_FG_PRED_COR_CP),1hex]
       else T_DLY_CP_OPEN            =
              max[IP_T_DLY_CP_2 (MAF_KGH_FG_PRED_COR_CP),1hex]
       endif
       // T_DLY_CP_OPEN will be set to IP_T_DLY_CP / 10
       LV_CP_AFL_IT            = 1                          // lean
       FLOW_GRD_CPS       = 0
endif
end
```

28.7.4.4 FLOW_CTL_CPS ramping

The ramping up and down of the internal flow setpoint FLOW_CTL_CPS is defined by the target (final setpoint to be reached) FLOW_TAR_CPS and the flow gradient (increment rsp. decrement) FLOW_GRD_CPS.

This calculation is applied in all states with any flow ramps.

```
if     FLOW_CTL_CPS$_{n-1}$ > FLOW_TAR_CPS          // actual value higher than target
then   FLOW_CTL_CPS$_n$ =                           // decreasing flow
              max [FLOW_TAR_CPS, FLOW_CTL_CPS$_{n-1}$ - FLOW_GRD_CPS]
else   FLOW_CTL_CPS$_n$ =                           // increasing flow
              min [FLOW_TAR_CPS, FLOW_CTL_CPS$_{n-1}$ + FLOW_GRD_CPS]
endif
```

28.7.5 General calculations

General information:

This submodule is calculated in every state_cp except CP_NOT_ACT

When the maximum engine speed limitation is reached (LV_N_MAX_CP) some of the calculations for the injection correction must be stopped to decrease the CPU load. If LV_N_MAX_CP is active then the normal calculations of the following variables are not done and the values of these variables are set to 0.

Application conditions:

Initialisation: at RST or IGKON. values are set to zero.
Recurrence: called by superior block
Activation: --
Deactivation: --

Formula section:

```
call       "Time counter for time after MAX_PURGE operation"
if         LV_N_MAX_CP = 1
then  call "Default values for disabled calculations"
else  call "Commutation on diagnosis detection"
      call "OBD2 diagnosis activation"
      call "Calculation of FLOW_CPS and FLOW_COR_CPS"
endif
if         LC_CP_CLL_INH_MAN = 0
then  call "Padé-filter and PT1-Filter calculations"
endif
end
```

28.7.5.1 Time counter for time after MAX_PURGE operation

Depending on this time counter, RAMP_OPEN (counter=0) or RAMP_OPEN_FAST (>0) are activated for CL learning. As long as the counter has not reached zero, the CL used in the previous MAX_PURGE phase is considered as usable.

Formula section:

```
if         STATE_CP = MAX_PURGE
then       if    STATE_OPM_ENG_CP = ENG_MOD_0
           then  T_DLY_MAX_CP = C_T_DLY_MAX_CP_0           // counter init
           else  T_DLY_MAX_CP = C_T_DLY_MAX_CP_1
           T_DLY_CL_MDL = max [0, T_DLY_CL_MDL - TA]
elseif           LV_CP_CLL = 0
then       T_DLY_MAX_CP = max [0, T_DLY_MAX_CP - TA]
           T_DLY_CL_MDL = C_T_DLY_CL_MDL
else       $T\_DLY\_MAX\_CP_n = T\_DLY\_MAX\_CP_{n-1}$
           $T\_DLY\_CL\_MDL_n = T\_DLY\_CL\_MDL_{n-1}$
```

```
Endif

if      LV_CP_NEW_RAMP_OPEN = 1
then T_DLY_MAX_CP = 0
endif

if      T_DLY_MAX_CP        = 0
then LV_CL_CLC_VLD       = 0                    // switch off int. MIN_PURGE
endif
end
```

28.7.5.2 Default values for disabled calculations

When the maximum engine speed limitation is reached (LV_N_MAX_CP) some of the calculations for the injection correction must be stopped to decrease the CPU load. If LV_N_MAX_CP is active then the normal calculations of the following variables are not done and the values of these variables are set to 0.

Formula section:

```
FLOW_SP_CPS                   = 0
FLOW_COR_CPS             = 0
FLOW_CPS                 = 0
FLOW_TOT_CPS             = 0
MAF_CPS                  = 0
end
```

28.7.5.3 OBD2 diagnosis activation

General information:

In order to manage the priority between canister purge and catalyst diagnosis or between canister purge and O2 sensor diagnosis, the bit information LV_CP_ACT_REQ will be sent to the diagnosis OBD2 manager. Two cases are possible :

- first of all, when the conditions to purge are set in order to learn the canister load (RAMP_OPEN phase).
- when LV_T_DLY_REQ_CP=1 if the canister load (CL_MMV) is greater than a calibrated value C_CL_MMV_REQ.

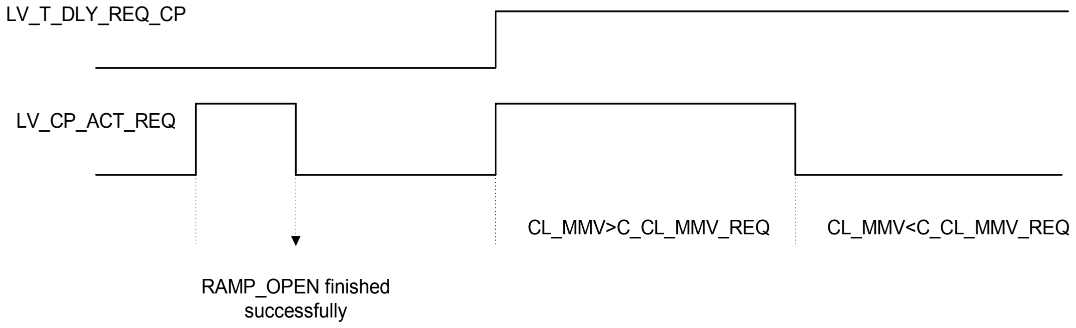


figure 7: LV_CP_ACT_REQ

Formula section:

```
LV_CP_ACT_REQ =
    [ LV_T_DLY_REQ_CP and (CL_MMV > C_CL_MMV_REQ) ]
or  STATE_CP = RAMP_OPEN
end
```

28.7.5.4 Calculation of FLOW_CPS and FLOW_COR_CPS

The setpoint of the mass flow through the CPS (FLOW_COR_CPS) is calculated using the actual calculated values of FLOW_CTL_CPS, a ramp gradient depending on MAF_CYL and the correction factors on ambient pressure and ambient temperature.

If the OBD II evaporative system monitoring (vaccum method) function is active (LV_DIAGCP_CPS_ACT=1), the opening of the CPS is controlled by the value of FLOW_SP_CPS_EVAP (see chapter OBDII functions).

The opening of the CPS can also be controlled by the application system using the switch LC_FLOW_CPS_AS and C_FLOW_CPS_AS.

The calculation of FLOW_SP_CPS is limited using a gradient value.

In RAMP_OPEN, RAMP_OPEN_FAST, NO_PURGE and RAMP_TOP_NO_PURGE, the gradient limitation is no more applied.

The flow request through the CPS is limited by the physical maximum flow at fully openend valve defined in FLOW_MAX_PHY_CPS (depending on pressure difference ambient air – intake manifold)

The value FLOW_TOT_CPS permits to measure the quantity of flow during an emission cycle.

Signal flow diagrams:

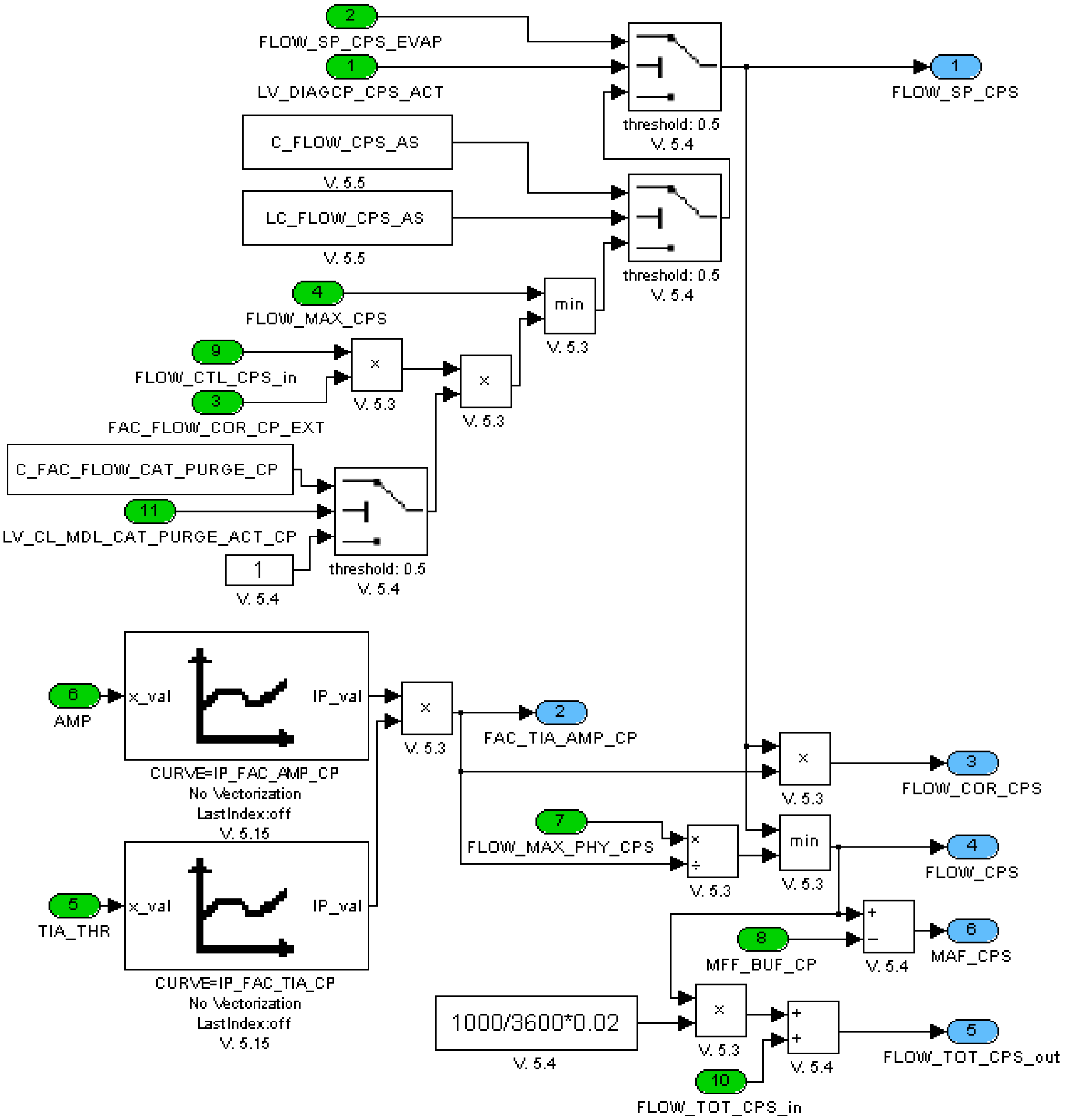


figure 8: Calculation of FLOW_CPS, FLOW_COR_CPS

Formula section:

```
if        LV_DIAGCP_CPS_ACT              // Interface with leak detection by vaccum
method
then      FLOW_SP_CPS        = FLOW_SP_CPS_EVAP
elseif         LC_FLOW_CPS_AS            // manual setting of flow_sp_cps
then      FLOW_SP_CPS = C_FLOW_CPS_AS
else      if LV_CL_MDL_CAT_PURGE_ACT_CP = 0
          then FLOW_SP_CPS = min
```

```
                    [FLOW_CTL_CPS * FAC_FLOW_COR_CP_EXT, FLOW_MAX_CPS]
          else FLOW_SP_CPS = min
                    [FLOW_CTL_CPS * FAC_FLOW_COR_CP_EXT *
                         C_FAC_FLOW_CAT_PURGE_CP, FLOW_MAX_CPS]
endif

// correction factor for ambient air temperature and ambient air pressure
FAC_TIA_AMP_CP =
          IP_FAC_AMP_CP     (AMP)
        * IP_FAC_TIA_CP     (TIA_THR)

FLOW_COR_CPS = FLOW_SP_CPS * FAC_TIA_AMP_CP
FLOW_CPS = min [
        FLOW_SP_CPS,
        FLOW_MAX_PHY_CPS / FAC_TIA_AMP_CP]

FLOW_TOT_CPS        = FLOW_TOT_CPS + FLOW_CPS / 360
MAF_CPS             = FLOW_CPS - MFF_BUF_CP
end
```

28.7.5.5 Padé- and Low pass filter calculations

(Calculation of FLOW_DLY_CP, FLOW_DLY_MMV_CP, MAF_CP, MAF_DLY_CP and MAF_DLY_MMV_CP)

Application conditions:

Initialisation: --
Recurrence: **100** ms
Activation: --
Deactivation: --

Signal flow diagram:

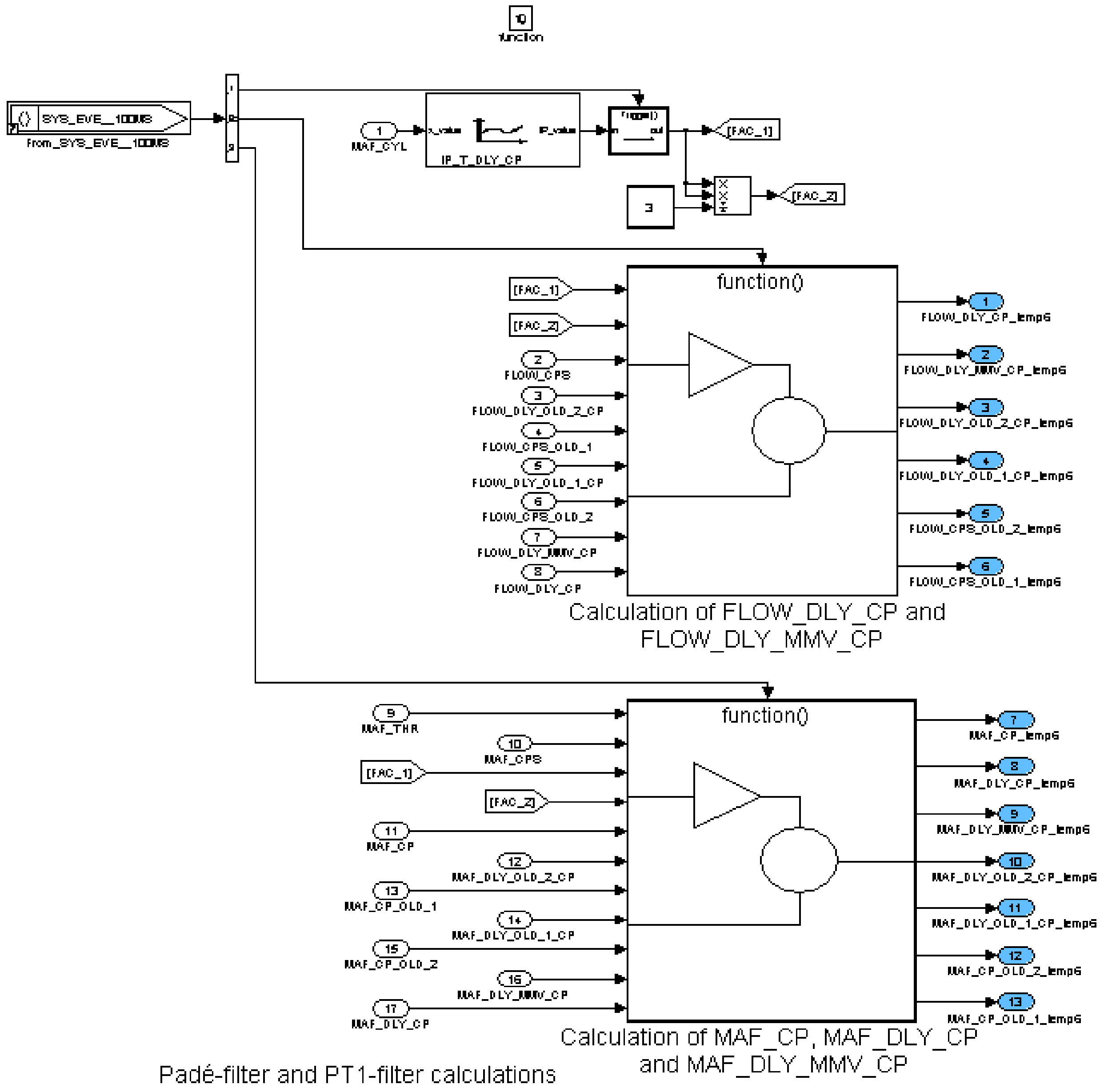


figure 9: Padé- and low pass filters

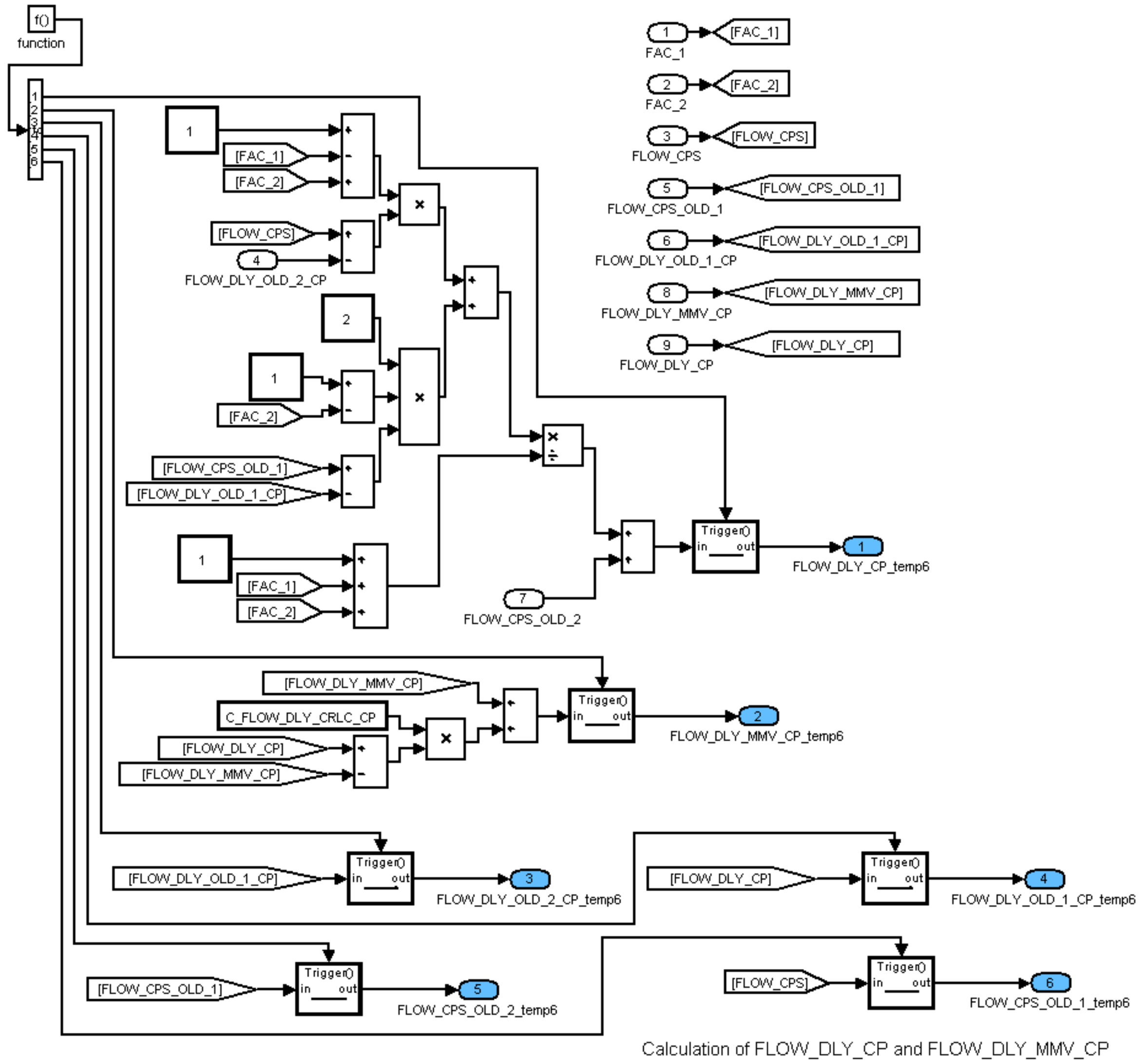


figure 10: calculation of FLOW_DLY_CP and FLOW_DLY_MMV_CP
the calc. of MAF_DLY_CP and MAF_DLY_MMV_CP is done following the same scheme

Formula section:

Padéfilter-parameters

if STATE_OPM_ENG_CP = ENG_MODE_0
then FAC_1 = IP_T_DLY_CP (MAF_KGH_FG_PRED_COR_CP)
else FAC_1 = IP_T_DLY_CP_2 (MAF_KGH_FG_PRED_COR_CP)
FAC_2 = FAC_1 * FAC_1 / 3

Calculation of FLOW_DLY_CP and FLOW_DLY_MMV_CP:

// Padé-filter
FLOW_DLY_CP=
1 / (1 + FAC_1 + FAC_2)
* [(1 - FAC_1 + FAC_2)
* (FLOW_CPS - FLOW_DLY_OLD_2_CP)
+ 2 * (1 - FAC_2) * (FLOW_CPS_OLD_1 - FLOW_DLY_OLD_1_CP)]
+ FLOW_CPS_OLD_2

```
// Low pass-filter
FLOW_DLY_MMV_CP$_n$= FLOW_DLY_MMV_CP$_{n-1}$ +
        (FLOW_DLY_CP$_n$ - FLOW_DLY_MMV_CP$_{n-1}$) * C_FLOW_DLY_CRLC_CP

// memorization for next recurrence
FLOW_DLY_OLD_2_CP      = FLOW_DLY_OLD_1_CP
FLOW_DLY_OLD_1_CP      = FLOW_DLY_CP
FLOW_CPS_OLD_2         = FLOW_CPS_OLD_1
FLOW_CPS_OLD_1         = FLOW_CPS
```

Calculation of MAF_CP, MAF_DLY_CP and MAF_DLY_MMV_CP:

```
MAF_CP = MAF_THR + MAF_CPS

// Padé-filter
MAF_DLY_CP(n)=
        1 / (1 + FAC_1 + FAC_2)
        * [ (1 - FAC_1 + FAC_2)
        * (MAF_CP$_n$ - MAF_DLY_OLD_2_CP)
        + 2 * (1 - FAC_2) * (MAF_CP_OLD_1 - MAF_DLY_OLD_1_CP) ]
        + MAF_CP_OLD_2

// Lowpass-filter
MAF_DLY_MMV_CP$_n$ = MAF_DLY_MMV_CP$_{n-1}$ +
        (MAF_DLY_CP$_n$ - MAF_DLY_MMV_CP$_{n-1}$) * C_MAF_DLY_CRLC_CP

// memorization for next recurrence
MAF_DLY_OLD_2_CP       = MAF_DLY_OLD_1_CP
MAF_DLY_OLD_1_CP       = MAF_DLY_CP
MAF_CP_OLD_2           = MAF_CP_OLD_1
MAF_CP_OLD_1           = MAF_CP
```

Calculation of MAF_DLY_CPS and MAF_DLY_MMV_CPS:

```
// Padé-filter
MAF_DLY_CPS(n)=
        1 / (1 + FAC_1 + FAC_2)
        * [ (1 - FAC_1 + FAC_2)
        * (MAF_CPS$_n$ - MAF_DLY_OLD_2_CPS)
        + 2 * (1 - FAC_2) * (MAF_CPS_OLD_1 - MAF_DLY_OLD_1_CPS) ]
        + MAF_CPS_OLD_2

// Lowpass-filter
MAF_DLY_MMV_CPS$_n$ = MAF_DLY_MMV_CPS$_{n-1}$ +
        (MAF_DLY_CPS$_n$ - MAF_DLY_MMV_CPS$_{n-1}$) * C_MAF_DLY_CRLC_CPS

// memorization for next recurrence
MAF_DLY_OLD_2_CPS      = MAF_DLY_OLD_1_CPS
MAF_DLY_OLD_1_CPS      = MAF_DLY_CPS
```

```
MAF_CPS_OLD_2          = MAF_CPS_OLD_1
MAF_CPS_OLD_1          = MAF_CPS
end
```

28.8 Evaporative emission control (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_AST_CP	O/V	0...FFFFH	0...6553.5	0.1	[s]
Time after first start in driving cycle					
LV_CP_ACT	O/V	0...1H	0...1	1	[-]
Logical value for EVAP-Control active					
LV_INH_CP	O/V	0...1H	0...1	1	[-]
Inhibits any purge operation, forces NO_PURGE (via a ramp)					
LV_INH_CP_FAST	O/V	0...1H	0...1	1	[-]
Inhibits any purge operation, forces a direct jump to NO_PURGE					
LV_INH_CP_CLL	O/V	0...1H	0...1	1	[-]
Inhibits closed loop operation, leads to MIN_PURGE, if no other reasons force NO_PURGE					
LV_CP_DYW_EXT	O/V	0...1H	0...1	1	[-]
flag collecting project specific dynamic window conditions					
LV_INH_LAM_AD_SDL	O/V	0...1H	0...1	1	[-]
flag collecting project specific reasons to inhibit LAM_AD via time control / permanently enable LAM_AD					
LV_INH_CP_SDL	O/V	0...1H	0...1	1	[-]
flag collecting project specific reasons to inhibit CP via time control / permanently enable CP					
LV_T_SDL_EXT_REQ	O/V	0...1H	0...1	1	[-]
flag collecting project specific reasons to temporarilly disabling CP and lambda adaptation					
T_DLY_CP_LAMB_CH	V	0...FFFFH	0...1310.7	0.02	[s]
Delay time after LAMB_CH has finished to enable NO_PURGE					
LV_T_DLY_CP_LAMB_CH	-	0...1H	0...1	1	[-]
flag indicating that delay time to enable NO_PURGE after LAMB_CH has finished was started					
T_DLY_CP_OPM_AV	V	0...FFFFH	0...1310.7	0.02	[s]
Delay time after OPM_AV change					
LV_CPS_AD_INH	O/V	0...1H	0...1	1	[-]
CPS adaptation inhibited					
CTR_DLY_CLL_IS_S_CP	V	0...FFFFH	0...65535	1	[-]
counter to disabled CLL at IS at unknown CL					
LV_INH_CLL_IS_S_CP	O/V	0...1H	0...1	1	[-]
Flag for disabled CLL at IS at unknown CL					
FAC_FLOW_COR_CP_EXT	O/V	0...FFH	0...3.98437	0.015625	[-]
External factor to modify absolute flow					
FAC_FLOW_COR_CP_MIN_PURGE	O/V	0...FFH	0...3.98437	0.015625	[-]
External factor to modify absolute flow for OPM_AV_CP = 1					
PRS_CPS_EXT	O/V	8000...7FFFH	-2717.04145... 2716.95854	0.0829175	[hPa]
Pressure difference envirionment over CP line					
LV_CP_CLOSE_1	O/V	0...1H	0...1	1	[-]
Request to inhibit the canister purge function with high closing speed					
LV_CP_CLOSE_2	O/V	0...1H	0...1	1	[-]
Request to inhibit the canister purge function with medium closing speed					
LV_CP_CLOSE_3	O/V	0...1H	0...1	1	[-]
Request to inhibit the canister purge function with normal closing speed					
LV_C_FAC_LAM_0_CP_WIDE	O/V	0...1H	0...1	1	[-]
Set the limitations of C_FAC_LAM_0_CP to wide threshold					
T_LAM_DEV_CHK_CP	V	0...FFFFH	0...1310.7	0.02	[s]
Time couter for lambda deviation check					
T_LAM_DEV_SYM_CP	V	0...FFFFH	0...1310.7	0.02	[s]
Time counter for the symptom in lambda deviation check					
STATE_OPM_ENG_CP	O/V	0H 1H 2H	ENG_MODE_0 ENG_MODE_1 ENG_MODE_2	1	[-]
Engine mode for CP					
LV_STATE_OPM_CHG_S_CP	O/V	0...1H	0...1	1	[-]
Engine mode change ENG_MODE_1 <-> ENG_MODE_2					

LV_STATE_OPM_CHG_HOM_CP	O/V	0...1H	0...1	1	[-]
Engine mode change ENG_MODE_0 <-> ENG_MODE_1or2					
FAC_FLOW_TAR_COR_CP	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Reduction factor for the flow target at close engine mode swich					
LV_FLOW_TAR_COR_CP	O/V	0...1H	0...1	1	[-]
Reduction for the flow target at close engine mode swich					
MFF_RED_OPM_THD_CP	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Area around the engine mode switch threshold to reduce the flow					
MFF_GRD_CP	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
MFF gradient for canister purge					
MFF_GRD_MMV_CP	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Filtered MFF gradient for canister purge					
LV_STATE_T_SDL_CP_LAM_AD_EXT	O/V	0...1H	0...1	1	[-]
External condition to force the time management to switch to lambda adaptation time					

Input data:

LV_ES	LV_ST	LV_AST	TCO_ST
T_AST			
LV_ERR_LAM_ADJ[NC_CBK_EX_NR]	LV_ENA_CHK_FUC_ROUGH_LEAK	LV_ACT_VLS_EOL_EXT_ADJ	LV_DIAG_OPL_REQ_CBK[NC_CBK_EX_NR]
LV_ACT_SA_EOL	LV_ERR_PVS_DOUBLE	LV_ERR_TPS	OPM_AV
LV_ACT_SAP_EXT_ADJ	LV_ERR_TIA	LV_ACT_SAV_EXT_ADJ	LV_ERR_MAF
	LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_ERR_LOAD_TPS_PLAUS	LV_DIAGCPS_MIN_MOD
LV_ERR_AMP	LV_MIS_STATE_A	LV_ERR_IGC	ERR_SYM_DMTLS
LV_ERR_CHG_LS_UP	LV_NO_PURGE_DMTL	MAP_MES	LV_TLDV_REQ
LV_ERR_EL_CPS	LV_SA_END	LV_TI_CH	LV_VAR_LSH_UP
LV_ERR_DIAGCPS	LV_ST_END	AMP	LV_CP_INH_CUS
STATE_ERR_IV	LV_CP_CLL	MFF_SP_MV	T_AST
TCO	LV_CYL_BAL_LAM_AD_EOL	MFF_MAX_HOMS_CUS	MFF_MAX_S_CUS
CL_MMV	LV_ERR_MAP_DIP_PLAUS	LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_SHIFT
LV_LAM_AD_EXT_ADJ			
MAP_DIP_MES_BAS_2SEG			
FAC_LAM_LIM[NC_CBK_EX_NR]	FAC_LAM_AD_OUT[NC_CBK_EX_NR]	FAC_MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	FAC_LAM_AD_CUS[NC_CBK_EX_NR]
STATE_CP	LV_LAM_LSCL[NC_CBK_EX_NR]	LV_LAM_AD_EXT	LC_LAM_CBK_CPS
LAMB_SP			
LV_LAM_LIM_LAM_AD[NC_CBK_EX_NR]	LV_CYL_BAL_LAM_AD_REQ_CUS		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_CP_ACT	1	0...1H	0...1	1	[-]
central switch for activation / deactivation of canister purge functions					
IP_T_ST_AST_TCO_ST_CPS_MIN	6	1...FFFFH	0.1...6553.5	0.1	[s]
LDP_TCO_ST_IP_T_ST	6	0...FEH	-48...142.5	0.75	[°C]
Minimum temperature for EVAP-Control active					
LC_CP_AST_ACT	1	0...1H	0...1	1	[-]
Use of LV_AST for activation of evaporative emission control function					
C_T_DLY_CP_LAMB_CH	1	0...FFFFH	0...1310.7	0.02	[s]
Calibrateable delay time after LAMB_CH has finished to enabe NO_PURGE					
C_T_DLY_CP_OPM_AV	1	0...FFFFH	0...1310.7	0.02	[s]
Calibrateable time delay for NO_PURGE after STATE_OPM_AV changed					

C_T_AST_MAX_INH_CP_SDL	1	0...FFFFH	0...6553.5	0.1	[s]
Max time after start to reset LV_INH_CP_SDL					
C_TCO_MAX_INH_CP_SDL	1	0...FEH	-48...142.5	0.75	[°C]
Max cooling water temperature to reset LV_INH_CP_SDL					
C_CTR_DLY_CLL_IS_S_MAX_CP	1	0...FFFFH	0...65535	1	[-]
Max value for counter to disabled CLL at IS at unknown CL					
C_CTR_DLY_CLL_IS_S_INC_CP	1	0...FFFFH	0...65535	1	[-]
Increment step for counter to disabled CLL at IS at unknown CL					
C_CTR_DLY_CLL_IS_S_DEC_CP	1	0...FFFFH	0...65535	1	[-]
Decrement step for counter for to disabled CLL at IS at unknown CL					
C_CL_MMV_CLL_IS_S_MAX	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum allowed CL_MMV for closed loop operation in stratified engine operation mode at idle speed					
LC_ERR_MAP_MIN_PURGE_CP	1	0...1H	0...1	1	[-]
Enables Min Purge operation at MAP error, otherwise switched to No Purge					
LC_ERR_AMP_INH_CLL	1	0...1H	0...1	1	[-]
Inhibits the EVAC to be in close loop if an AMP error is detected					
IP_FAC_REL_FLOW_CPS_WOUT_CAT	12*12	0...FFH	0...3.98437	0.015625	[-]
LDPM_N_32_IP_REL_FLOW_MAX_CPS	12	0...FFH	0...8160	32	[rpm]
LDPM_MAF_IP_REL_FLOW_MAX_CPS	12	0...FFH	0...1389	5.4470588	[mg/stk]
Factor for CP MIN_PURGE					
C_STATE_CONF_TLDV_CP	1	0...3H	0...3	1	[-]
Configuration byte for closing speed of canister purge before leak detection test started					
C_STATE_CONF_CAT_CP	1	0...3H	0...3	1	[-]
Configuration byte for closing speed of canister purge before catalyst monitor test started					
LC_TLDV_NOT_SEQ	1	0...1H	0...1	1	[-]
Configuration bit to select if the OBD2 sequencer is present (0) or not (1)					
ID_STATE_OPM_ENG_CP	8	0...2H	0...2	1	[-]
LDPM_OPM_AV	8	0...8H	0...8	1	[-]
Mode selection for canister purge depending on engine operation mode					
IP_MFF_RED_OPM_CP	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_MFF_SP_MV_IP_MFF_RED_CP	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
Area around the engine mode switch threshold to reduce the flow					
C_CRLC_MFF_GRD_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation factor for MFF_GRD_CP calculation					
C_FAC_FLOW_TAR_COR_MIN_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Lower thershold for the flow correction factor					
IP_FAC_LAM_DEV_MIN_CP	4	0...FFFFH	-100...99.99694	3.0518e-3	[%]
LDPM_LAMB_SP_CP	4	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambdacontroller minimum threshold for possibility of fuel system erreor dependend on LAMB_SP[i]					
IP_FAC_LAM_DEV_MAX_CP	4	0...FFFFH	-100...99.99694	3.0518e-3	[%]
LDPM_LAMB_SP_CP	4	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambdacontroller maximum threshold for possibility of fuel system erreor dependend on LAMB_SP[i]					
LC_LAM_DEV_RESU_CP	1	0...1H	0...1	1	[-]
Resume the lambda deviation detection					
C_T_LAM_DEV_CHK_MAX_CP	1	0...FFFFH	0...1310.7	0.02	[s]
Time to detect the possibility of an error in the fuel system					
C_T_LAM_DEV_SYM_MAX_CP	1	0...FFFFH	0...1310.7	0.02	[s]
Threshold time for the symptom to detect the possibility of an error in the fuel system					
LC_LAM_LIM_AD_ENA	1	0...1H	0...1	1	[-]
switch to enable forced lambda adaptation					

28.8.1 Project specific definition for "Time after start"

General information:

T_AST_CP has to be written in general with T_AST. Only special configurations, e.g. for stop-start functionality, T_AST_CP has to be written with other values (e.g. T_AST_DC).

Application conditions:

Initialisation: *T_AST_CP = 0 after reset*

Recurrence: 100ms

Activation: all engine states

Deactivation: --

Formula section:

T_AST_CP = T_AST

28.8.2 Activation of the Evaporative Emission Control Function

General information:

The Evaporative Emission Control function is activated (LV_CP_ACT=1), if the time after start exceeds IP_T_ST_AST_TCO_ST_CPS_MIN, which depends on coolant temperature after start. The function is active during all engine operating states except the following states:

- engine stopped (LV_ES=1)
- engine start (LV_ST=1)

For the MSA stop/start functionality it is necessary to calculate a seperat T_AST_CP, which represents the time after first start in an driving cycle.

Application conditions:

Initialisation: LV_CP_ACT = 0 after reset or stalling

Recurrence: 20ms
Activation:
Deactivation: --

Formula section:

LV_CP_ACT =
[T_AST_CP > IP_T_ST_AST_TCO_ST_CPS_MIN(TCO_ST)
and LV_ES = 0
and LV_ST = 0
and [LV_AST = 0 **or** LC_CP_AST_ACT = 0]
and LC_CP_ACT = 1]

28.8.3 Cross-locking table

General information:

This chapter describes the evaporative emission control inhibition.

Application conditions:

Initialisation: at reset:

LV_INH_CP = 1
LV_INH_CP_FAST = 0
LV_INH_CP_CLL = 0
LV_CP_DYW_EXT = 1 (never changed after initialisation)
LV_INH_LAM_AD_SDL = 0 (never changed after initialisation)
LV_INH_CP_SDL = 0
LV_T_DLY_CP_LAMB_CH = 0
T_DLY_CP_LAMB_CH = 0
LV_CPS_AD_INH = 0 (never changed after initialisation)
T_DLY_CP_OPM_AV = 0

at LV_IGK 0 -> 1

LV_T_DLY_CP_LAMB_CH = 0
T_DLY_CP_LAMB_CH = 0
CTR_DLY_CLL_IS_S_CP = C_CTR_DLY_CLL_IS_S_MAX_CP
LV_INH_CLL_IS_S_CP = 1
T_DLY_CP_OPM_AV = 0

Recurrence: **20** ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

```
If      LV_SA_END = 1       and
        LV_TI_CH = 0        and
        $LV\_T\_DLY\_CP\_LAMB\_CH_{n} = 0$
Then    $LV\_T\_DLY\_CP\_LAMB\_CH_{n-1} = 1$
        T_DLY_CP_LAMB_CH  =  C_T_DLY_CP_LAMB_CH
Endif

If      LV_SA_END = 1       and
        LV_TI_CH = 0        and
        T_DLY_CP_LAMB_CH > 0
Then    $T\_DLY\_CP\_LAMB\_CH = T\_DLY\_CP\_LAMB\_CH_{n-1} - 20$ ms
```

```
Endif


If          $OPM\_AV_n$  != $OPM\_AV_{n-1}$
Then        T_DLY_CP_OPM_AV = C_T_DLY_CP_OPM_AV
Else        $T\_DLY\_CP\_OPM\_AV_n = T\_DLY\_CP\_OPM\_AV_{n-1}$ – 20 ms
Endif
Counter to prevent RO/ROF at idle speed and stratified engine mode at unknown CL_MMV
If          LV_CP_CLL = 1
Then        CTR_DLY_CLL_IS_S_CP = max(0,CTR_DLY_CLL_IS_S_CP –
                                        C_CTR_DLY_CLL_IS_S_DEC_CP)
            If      CTR_DLY_CLL_IS_S_CP = 0
            Then    LV_INH_CLL_IS_S_CP = 0
            endif
Else        CTR_DLY_CLL_IS_S_CP =
                    min(C_CTR_DLY_CLL_IS_S_MAX_CP,CTR_DLY_CLL_IS_S_CP +
                                                C_CTR_DLY_CLL_IS_S_INC_CP)
            If      CTR_DLY_CLL_IS_S_CP = C_CTR_DLY_CLL_IS_S_MAX_CP
            Then    LV_INH_CLL_IS_S_CP = 1
            endif


endif
```

For activation of **NO_PURGE (via a ramp)** and setting of the flag LV_INH_CP the following conditions must be fulfilled:

```
If          LV_ERR_EL_CPS =1                        or
            LV_ERR_DIAGCPS =1                       or
            {(LV_ERR_MAP_DIP_PLAUS          or
            LV_ERR_MAP_DIP_SENS             or
            LV_ERR_MAP_DIP_SHIFT)           and
            LC_ERR_MAP_MIN_PURGE_CP = 0}            or
            LV_ACT_SAV_EXT_ADJ =1                   or
            LV_ACT_SAP_EXT_ADJ =1                   or
            LV_ACT_SA_EOL =1                        or
            LV_NO_PURGE_DMTL = 1                    or
            ERR_SYM_DMTLS = “SYM_1”                 or
            LV_MIS_STATE_A = 1                      or
```

```
              T_DLY_CP_OPM_AV > 0                  or
              LV_SA_END = 0                        or
              LV_TI_CH = 1                         or
              LV_ENA_CHK_FUC_ROUGH_LEAK = 1        or
              LV_CP_INH_CUS = 1                    or
              ( LV_SA_END = 1                  and
              LV_TI_CH = 0                     and
              T_DLY_CP_LAMB_CH > 0 )
Then          LV_INH_CP = 1
Else          LV_INH_CP = 0
Endif

If            LV_DIAG_OPL_REQ_CBK[i] = 1           or   //for any i
              LV_ACT_VLS_EOL_EXT_ADJ = 1
Then          LV_INH_CP_FAST = 1
Else          LV_INH_CP_FAST = 0
Endif

If            ( LV_CYL_BAL_LAM_AD_EOL = 1          or
              T_AST > C_T_AST_MAX_INH_CP_SDL       or
              TCO > C_TCO_MAX_INH_CP_SDL )         and
              LV_LAM_AD_EXT_ADJ = 0
Then          LV_INH_CP_SDL = 0
Else          LV_INH_CP_SDL = 1
Endif
```

For activation of **MIN_PURGE** and setting of the flag LV_INH_CP_CLL the following conditions must be fulfilled:

```
If            LV_ERR_MAF = 1                       or
              LV_ERR_TPS = 1                       or
              LV_ERR_TIA = 1                       or
              ( LV_ERR_AMP = 1                     and
              LC_ERR_AMP_INH_CLL = 1 )             or
```

```
        STATE_ERR_IV ≠ 0                        or
        LV_ERR_LS_UP[i] =1                      or
        LV_ERR_PVS_DOUBLE = 1                   or
        LV_ERR_IGC = 1                          or
        LV_ERR_LOAD_TPS_PLAUS = 1               or
        LV_ERR_LAM_ADJ[i] = 1                   or
        LV_ERR_CHG_LS_UP = 1                    or
        LV_DIAGCPS_MIN_MOD = 1                  or
        {(LV_ERR_MAP_DIP_PLAUS          or
        LV_ERR_MAP_DIP_SENS             or
        LV_ERR_MAP_DIP_SHIFT)           and
        LC_ERR_MAP_MIN_PURGE_CP = 1}            or
        // only open loop at idle speed and stratified engine mode and unknown CL
        // If CL is known, closed loop only for low CL_MMV
        [(LV_IS = 1                             and
        STATE_OPM_ENG_CP != ENG_MODE_0 and
        {LV_INH_CLL_IS_S_CP = 1
        or CL_MMV > C_CL_MMV_CLL_IS_S_MAX } ) ]

Then    LV_INH_CP_CLL = 1
Else    LV_INH_CP_CLL = 0
Endif
```

28.8.4 Project specific factor for MIN purge flow

General information:

With this function, the MIN_PURGE flow can be varied by a factor from 0 to 4 for vehicles without catalyst and lambda sensors or in case of using different engine operation modes (STATE_OPM_ENG_CP).

Application conditions:

Initialisation: FAC_FLOW_COR_CP_EXT = 1 *at reset*
FAC_FLOW_COR_CP_MIN_PURGE = 1 *at reset*

Recurrence: 100ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

```
If (1)  LV_VAR_LSH_UP = 1
```

```
Then (1)
        If (2) STATE_OPM_ENG_CP = 0
        Then (2) FAC_FLOW_COR_CP_MIN_PURGE = 1
        Else (2) FAC_FLOW_COR_CP_MIN_PURGE =
                    IP_FAC_REL_FLOW_CPS_WOUT_CAT
        Endif (2)
        FAC_FLOW_COR_CP_EXT  = 1
Else (1) FAC_FLOW_COR_CP_EXT  = IP_FAC_REL_FLOW_CPS_WOUT_CAT
        FAC_FLOW_COR_CP_MIN_PURGE = 1
Endif (1)
```

28.8.5 External calculations

General information:

With this function, it is possible to supply the “Evaporative emission control variables” function with a measured canister purge differential preasure. For further information on the use of this “external” differential preasure please refer to chapter “Evaporative emission control variables”.

Application conditions:

Initialisation: PRS_CPS_EXT = 0 at reset

Recurrence: 20ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

PRS_CPS_EXT = MAP_DIP_MES_BAS_2SEG

28.8.6 Management of LV_CP_CLOSE_CP_x with no OBD sequencer

General information:

This chapter describes the recommendations for evaporative emission control inhibition when no OBD sequencer is used. In this case the flags LV_CP_CLOSE_1, LV_CP_CLOSE_2, and LV_CP_CLOSE_3 must be managed within this module.

Examples that need to close purge are OBD diagnostics for leak detection and catalyst monitor.

Normally the flags LV_CP_CLOSE_CP_x are managed and defined by the OBD sequencer, if there is no sequencer it is recommended to set LC_TLDV_NOT_SEQ =1 and define and calculate these flags in the application inc. spec.

C_STATE_CONF_TLDV_CP and C_STATE_CONF_CAT_CP are used to configure the closing ramp of the CPS valve to inhibit purge during an OBD test.

Values of C_STATE_CONF_TLDV_CP and C_STATE_CONF_CAT_CP:

0: Canister purge remains active, no request to close the valve

1: Canister purge closed with highest speed closing ramp

2: Canister purge closed with medium speed closing ramp

3: Canister purge closed with normal speed closing ramp

Application conditions:

Initialisation: at reset or power loss:
LV_CP_CLOSE_1 = 0 (never changed after initialisation)
LV_CP_CLOSE_2 = 0 (never changed after initialisation)
LV_CP_CLOSE_3 = 0 (never changed after initialisation)

Recurrence: only initialisation

28.8.7 Selection of STATE_OPM_ENG_CP in case of engine operation mode change (OPM_AV)

General information:

This chapter describes the canister purge behaviour in case of changing engine operation mode (OPM_AV).

Application conditions:

Initialisation: at reset or power loss STATE_OPM_ENG_CP = 0

Recurrence: *100 ms*

Activation: every engine state

Formula Section:

STATE_OPM_ENG_CP = ID_STATE_OPM_ENG_CP

Setting of the flag LV_STATE_OPM_CHG_HOM_CP to detect the engine mode change from homogenious (ENG_MODE_0) to stratified (ENG_MODE_1/2) and vice versa and the flag LV_STATE_OPM_CHG_S_CP to detect the schange of the stratified engine mode (ENG_MODE_1 to ENG_MODE_2 and vice versa.

```
if    [(STATE_OPM_ENG_CP$_n$ = ENG_MODE_0
      and STATE_OPM_ENG_CP$_{n-1}$ != ENG_MODE_0)
      or  (STATE_OPM_ENG_CP$_n$ != ENG_MODE_0
      and STATE_OPM_ENG_CP$_{n-1}$ = ENG_MODE_0)]
then        LV_STATE_OPM_CHG_HOM_CP = 1
else        LV_STATE_OPM_CHG_HOM_CP = 0
endif

if     [(STATE_OPM_ENG_CP$_n$ = ENG_MODE_1
```

```
        and $\text{STATE\_OPM\_ENG\_CP}_{n-1}$ = ENG_MODE_2)
        or ($\text{STATE\_OPM\_ENG\_CP}_{n}$ = ENG_MODE_2
        and $\text{STATE\_OPM\_ENG\_CP}_{n-1}$ = ENG_MODE_1)]
    then    LV_STATE_OPM_CHG_S_CP = 1
    else    LV_STATE_OPM_CHG_S_CP = 0

    endif
```

28.8.8 end

28.8.9 Flow Reduction near engine operation mode switch

General information:

The two thresholds for an engine operating mode change is given by the two input values. Around this fuel mass point the flow through the CPS valve is reduced, to prevent disturbences at engine mode change by canister purge.

Application conditions:

Initialisation: *all values = 0 at reset*

Recurrence: *20 ms*

Activation: every engine state

Formula Section:

$\text{MFF\_GRD\_CP} = \mathbf{abs}(\text{MFF\_SP\_MV}_{n-1} - \text{MFF\_SP\_MV}_{n})$

$\text{MFF\_GRD\_MMV\_CPn} =$

$\text{MFF\_GRD\_MMV\_CP}_{n-1} + (\text{MFF\_GRD\_CP} - \text{MFF\_GRD\_MMV\_CP}_{n-1}) * \text{C\_CRLC\_MFF\_GRD\_CP}$

MFF_RED_OPM_THD_CP = IP_MFF_RED_OPM_CP(MFF_GRD_MMV_CP)

```
if      abs(MFF_SP_MV–MFF_MAX_S_CUS) < MFF_RED_OPM_THD_CP // HOMS<->S
then    LV_FLOW_TAR_COR_CP = 1
        FAC_FLOW_TAR_COR_CP =  max{ C_FAC_FLOW_TAR_COR_MIN_CP,
            abs(MFF_SP_MV–MFF_MAX_S_CUS)/ MFF_RED_OPM_THD_CP}
elseif  abs(MFF_SP_MV–MFF_MAX_HOMS_CUS) < MFF_RED_OPM_THD_CP
                                                        // HOMS<->HOM
then    LV_FLOW_TAR_COR_CP = 1
        FAC_FLOW_TAR_COR_CP = max{ C_FAC_FLOW_TAR_COR_MIN_CP,
            abs(MFF_SP_MV–MFF_MAX_HOMS_CUS)/ MFF_RED_OPM_THD_CP}
else    LV_FLOW_TAR_COR_CP = 0
```

```
            FAC_FLOW_TAR_COR_CP = 1
endif


end
```

28.8.10 Calculation of LV_C_FAC_LAM_0_CP_WIDE

General information:

At the start of each driving cycle the lambda deviation has to be monitored to detect the possibility of an error in the fuel system. If the possibility of an error is detected, the flag LV_C_FAC_LAM_0_CP_WIDE is set. This will open the limitations of FAC_LAM_0_CP.
TA = recurrence

Application conditions:

Initialisation:

```
at RST or LV_IGK=1
T_LAM_DEV_CHK_CP = C_T_LAM_DEV_CHK_MAX_CP
T_LAM_DEV_SYM_CP = 0
LV_C_FAC_LAM_0_CP_WIDE = 0
if     LC_LAM_CBK_CPS=0
then idx_cp = 0
else idx_cp = 1
endif
```

Recurrence: *20ms*
Activation: *every engine state*

Deactivation: --

Formula section:

```
if     LC_LAM_DEV_RESU_CP = 1
       and STATE_CP = MAX_PURGE
then   T_LAM_DEV_CHK_CP = C_T_LAM_DEV_CHK_MAX_CP
       T_LAM_DEV_SYM_CP = 0
       LV_C_FAC_LAM_0_CP_WIDE = 0
endif

if     LV_LAM_LSCL[idx_cp] = 1            // lambda controller acting
       and T_LAM_DEV_CHK_CP > 0
       and (STATE_CP = CP_NOT_ACT
            or STATE_CP = NO_PURGE)
then   T_LAM_DEV_CHK_CP = max (0, T_LAM_DEV_CHK_CP – TA)
       if     {LV_LAM_AD_EXT = 0
              and   [((FAC_LAM_LIM[idx_cp] + FAC_LAM_AD_OUT[idx_cp] +
                    FAC_MFF_ADD_LAM_AD_OUT[idx_cp]) >=
                    IP_FAC_LAM_DEV_MAX_CP(LAMB_SP))
```

```
                  or((FAC_LAM_LIM[idx_cp] + FAC_LAM_AD_OUT[idx_cp] +
                  FAC_MFF_ADD_LAM_AD_OUT[idx_cp]) <=
                  IP_FAC_LAM_DEV_MIN_CP(LAMB_SP))]}
              or
              {LV_LAM_AD_EXT = 1
              and [((FAC_LAM_LIM[idx_cp] + FAC_LAM_AD_CUS[idx_cp]) >=
                  IP_FAC_LAM_DEV_MAX_CP(LAMB_SP))
                  or((FAC_LAM_LIM[idx_cp] + FAC_LAM_AD_CUS[idx_cp]) <=
                  IP_FAC_LAM_DEV_MIN_CP(LAMB_SP))]}
         then T_LAM_DEV_SYM_CP ++
              if   T_LAM_DEV_SYM_CP = C_T_LAM_DEV_SYM_MAX_CP
              then LV_C_FAC_LAM_0_CP_WIDE = 1
              endif
          endif
    endif
    end
```

28.8.11 External request for Time slice "Lambda adaption"

General information:

With the flag LV_STATE_T_SDL_CP_LAM_AD_EXT the time management for lambda adaptation and canister purge can be forced to change and stay in one of the lambda adaptaiton time slices.

Application conditions:

Initialisation: *LV_STATE_T_SDL_CP_LAM_AD_EXT = 0 after reset*

Recurrence: 20ms

Activation: all engine states

Deactivation: --

Formula section:

```
If   ( LV_LAM_LIM_LAM_AD [i] = 1     // shall be true, if at least one bench is true
     and  LC_LAM_LIM_AD_ENA = 1) // immediate lambda adaptation required
     or LV_CYL_BAL_LAM_AD_REQ_CUS = 1
then LV_STATE_T_SDL_CP_LAM_AD_EXT = 1
else LV_STATE_T_SDL_CP_LAM_AD_EXT = 0
```

28.9 Injection calculation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MFF_BUF_CP	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Fuel flow from the CPS to the Cylinder					
MFF_ADD_BUF[NC_NR_CP_BUF]	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Array of MFF_ADD_CP_KGH values [NC_NR_CP_BUF elements default value= 20]					
MFF_TOT_CP	V	0...FFFFFFFFH	0...1455.99999	0.339e-6	[g]
Total of MFF_BUF_CP during emission cycle					
MFF_ADD_LAM_CP	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Additive adaptive correction of the mass fuel flow injection with active canister purge					
MFF_CPS	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
fuel mass calculation for torque correction in stratified combustion					
MFF_ADD_CYL_CP	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass from the canister in the cylinder					
TQI_ADD_CP	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Canister purge torque gain					
BUF_INP_CP	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Input for the fuel delay buffer					
BUF_OUT_CP	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Output of the fuel delay buffer					
IDX_CP	-	0...FFH	0...255	1	[-]
Current position within the ring buffer					
IDX_RD_CP	-	0...FFH	0...255	1	[-]
Reading position within the ring buffer					
IDX_STEP_CP	-	0...FFH	0...255	1	[-]
number of elements to fill with actual input beginning with actual position					
MAF_KGH_FG_PRED_COR_CP	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Predicted fresh gas out of manifold corrected by MAP and TIA for CP					

Input data:

N_32	TQI_SP_S	NC_CYL_NR	LV_HOM_ACT
MAF	MAF_KGH_FG_PRED	MAP	TIA[NC_SENS_NR_TIA]
MAF_CPS	MAF_CYL		
LV_FAC_LAM_SHIFT_CP	LC_CP_CLL_INH_MAN	LV_N_MAX_CP	MFF_ADD_CP_KGH
LV_CP_ACT	STATE_OPM_ENG_CP		

Input data table order: External inputs – Inputs from other EVAC modules

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_MFF_DLY_CP	6	0...FFH	0...255	1	[-]
LDPM_MAF_CYL_1_EVAC_MFF_DLY_CP	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Delay for MFF_BUF_CP calculation; maximum allowed calibration value is NC_NR_CP_BUF-1					
IP_MFF_DLY_CP_2	6	0...FFH	0...255	1	[-]
LDPM_MAF_CYL_1_EVAC_MFF_DLY_CP	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Delay for MFF_BUF_CP calculation with STATE_OPM_ENG_CP = 1					
IP_CRLC_MFF_BUF_CP	6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Ratio for MFF_BUF_CP calculation					
IP_IDX_STEP_CP	12	0...FFH	0...255	1	[-]
LDPM_MAF_KGH_CLC_ID_IDX_STEP_CP	12	0...FFFFH	0...2047.96875	0.03125	[kg/h]
number of elements+1 to fill with actual input beginning with actual position					
IP_IDX_STEP_CP_2	12	0...FFH	0...255	1	[-]
LDPM_MAF_KGH_CLC_ID_IDX_STEP_CP	12	0...FFFFH	0...2047.96875	0.03125	[kg/h]
number of elements+1 to fill with actual input beginning with actual position at STATE_OPM_ENG_CP = 1					
IP_MAF_KGH_PRED_COR_MAP_TIA	6*3	0...FFFFH	0...7.99987	0.1221e-3	[-]
LDPM_MAP_1_EVAC	6	0...FFFFH	0...5434	0.0829175	[hPa]
LDPM_TIA_1_EVAC	3	0...FEH	-48...142.5	0.75	[°C]
Correction factor for MAF_KGH_FG_PRED (MAP and TIA)					
IP_MAF_KGH_PRED_COR_MAP_TIA_2	6*3	0...FFFFH	0...7.99987	0.1221e-3	[-]
LDPM_MAP_1_EVAC	6	0...FFFFH	0...5434	0.0829175	[hPa]
LDPM_TIA_1_EVAC	3	0...FEH	-48...142.5	0.75	[°C]
Correction factor for MAF_KGF_FG_PRED (MAP and TIA) with of STATE_OP_ENG_CP = 1					
C_MAF_THR_MIN_BUF_CP	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Minimum MAF_THR value for Ringbuffer input calculation.					

28.10 Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
LC_FLOW_CPS_CAL	-	0...1H	0...1	1	[-]
Choice of recurence for MFF_ADD_LAM_CP calculation					
LC_MAF_COR_ACT_CP	-	0...1H	0...1	1	[-]
Activate the usage of the corrected MAF for reading the buffer					
NC_NR_CP_BUF	-	0...FFH	0...255	1	[-]
Size of the fuel buffer FUEL_FLOW_ADD_BUF					

General information:

When the Evaporative Emission Control is active, the fuel injection is corrected by an additive value MFF_ADD_LAM_CP to avoid emission and driveability problems caused by the fuel flow through the CPS. This correction is done all the time.

During RAMP_OPEN when the lambda control is shifted the buffer will be initialized to the last MFF_ADD_CP_KGH value.

The MFF_BUF_CP value is calculated to take into account the distance between the CPS VALVE and the injectors.

The value MFF_ADD_LAM_CP is added with a minus sign to the total formula of the injection calculation (see chapter INJECTION).

The value MFF_TOT_CP allow to measure the quantity of fuel during an emission cycle.

After the initialization phase the buffer must be calculated all the time, in order to have the buffer empty when the evaporative function is stopped.

Application conditions:

Initialisation: at reset and engine stop
MFF_TOT_CP = 0
at reset, engine stop and deactivation
MFF_ADD_BUF[all] = 0
MFF_BUF_CP = 0
MFF_ADD_LAM_CP = 0
MFF_CPS = 0
MFF_ADD_CYL_CP = 0
TQI_ADD_CP = 0
BUF_INP_CP = 0
BUF_OUT_CP = 0
IDX_CP = pointer at MFF_ADD_BUF[0]

Recurrence: **20** ms

Activation: [LV_CP_ACT = 1 **and** LC_CP_CLL_INH_MAN = 0]
Deactivation: [LV_CP_ACT = 0 **or** LC_CP_CLL_INH_MAN = 1]

Formula section:

```
if      LV_N_MAX_CP = 1
then    call  "Default values for disabled calculations"
else    call "handling of the ring buffer"
        if      LV_HOM_ACT = 1
        then    call "Injection correction calculation"
        else    call "Torque Correction calculation"
        endif
        MFF_TOT_CP = MFF_TOT_CP + MFF_BUF_CP / 360

endif
end
```

28.10.1 Default values for disabled calculations

When the maximum engine speed limitation is reached (LV_N_MAX_CP) some of the calculations for the injection correction must be stopped to decrease the CPU load. If LV_N_MAX_CP is active then the normal calculations of the following variables are not done and the values of these variables are set to 0.

Formula section:

```
MFF_ADD_BUF          = 0   (all values of the array)
MFF_BUF_CP           = 0
MFF_ADD_LAM_CP       = 0
TQI_ADD_CP           = 0
BUF_INP_CP           = 0
MFF_CPS              = 0
BUF_OUT_CP           = 0
MFF_ADD_CYL_CP       = 0
end
```

28.10.2 Handling of the ring buffer

The recurrence of this subchapter is configurable by LC_FLOW_CPS_CAL:

if LC_FLOW_CPS_CAL=0 then reccurency **20** ms
if LC_FLOW_CPS_CAL=1 then reccurency **40** ms
The buffer MFF_ADD_BUF has NC_NR_CP_BUF elements

IDX_CP, IDX_RD_CP and IDX_STEP_CP are pointers within the ring buffer; they have to be checked for exceeding the limits of the ring buffer.

IDX_CP is the current position within the ring buffer MFF_ADD_BUF. This position is incremented every 20 rsp. 40 ms, depending on LC_FLOW_CPS_CAL

Formula section:

Correction of MAF_KGH_FG_PRED (MAP and TIA)

```
If      STATE_OPM_ENG_CP = ENG_MODE_0
        Then MAF_KGH_FG_PRED_COR_CP = MAF_KGH_FG_PRED *
             IP_MAF_KGH_PRED_COR_MAP_TIA(MAP,TIA)
        Else MAF_KGH_FG_PRED_COR_CP = MAF_KGH_FG_PRED *
             IP_MAF_KGH_PRED_COR_MAP_TIA_2(MAP,TIA)

// The overflow handling of the buffer indices is not described here and has to be done software-internal

// reading
If LC_MAF_COR_ACT_CP = 0
then
        if   STATE_OPM_ENG_CP = ENG_MODE_0
        then IDX_RD_CP     = IDX_CP - IP_MFF_DLY_CP (MAF_CYL)
        else IDX_RD_CP     = IDX_CP - IP_MFF_DLY_CP_2 (MAF_CYL)
        endif
else
        if   STATE_OPM_ENG_CP = ENG_MODE_0
        then IDX_RD_CP     = IDX_CP -
             IP_MFF_DLY_CP (MAF_KGH_FG_PRED_COR_CP)
        else IDX_RD_CP     = IDX_CP -
             IP_MFF_DLY_CP_2 (MAF_KGH_FG_PRED_COR_CP)
        endif
endif

BUF_OUT_CP   = MFF_ADD_BUF[IDX_RD_CP] * MAF * N_32 * NC_CYL_NR * 3 /  $10^5$


// writing → IDX_STEP_CP elements are filled
if      MFF_ADD_CP_KGH = 0
then    BUF_INP_CP = 0
else    BUF_INP_CP     = MFF_ADD_CP_KGH /
             (max(MAF_THR,C_MAF_THR_MIN_BUF_CP) + MAF_CPS)
endif
if      LV_FAC_LAM_SHIFT_CP =1
        then MFF_ADD_BUF [all]    = BUF_INP_CP
             BUF_OUT_CP           = MFF_ADD_CP_KGH
             MFF_BUF_CP           = MFF_ADD_CP_KGH
        else if STATE_OPM_ENG_CP = ENG_MODE_0
             then IDX_STEP_CP    = IP_IDX_STEP_CP (MAF_KGH_FG_PRED_COR_CP)
             else  IDX_STEP_CP   =
                   IP_IDX_STEP_CP_2 (MAF_KGH_FG_PRED_COR_CP)
             endif
             for k = IDX_CP to IDX_CP + IDX_STEP_CP do
                   MFF_ADD_BUF[k] = BUF_INP_CP
             endfor
```

```
        endif
    // moving the position index
    IDX_CP = IDX_CP + IDX_STEP_CP + 1

    // final filtered buffer output
    $MFF\_BUF\_CP_n$ = [$MFF\_BUF\_CP_{n-1}$ + IP_CRLC_MFF_BUF_CP (MAF_CYL)
                              * (BUF_OUT_ CP – $MFF\_BUF\_CP_{n-1}$)]
    end
```

calibration hint:

If a modification of the recurrence of the ring buffer is done (modification on LC_FLOW_CPS_CAL), the delay times, which are calibrated as index, have to be updated according to the new recurrence.

28.10.3 Injection correction calculation

This correction is applied for homogeneous mode only. In stratified mode, correction is applied by torque (see next paragraph)

```
if    N_32 > 0
then  MFF_ADD_LAM_CP   = (MFF_BUF_CP * $10^5$) / (N_32 * NC_CYL_NR * 3)
else  MFF_ADD_LAM_CP   = 0
endif
MFF_ADD_CYL_CP = MFF_ADD_LAM_CP
TQI_ADD_CP           = 0
end
```

28.10.4 Torque Correction calculation

During stratified combustion, the impact of the premixed fuel flow coming from the CPS has to be compensated in an alternative way. It was demonstrated that the torque increase due to canister purge is dependent on the mass fuel flow from the CPS, the engine speed, and, additionally, on the mass flow of fuel that originates from the injector. The corrective torque will be calculated in the project spezific Torque coordination.

Formula section:

```
if    N_32 > 0
then  MFF_CPS          = MFF_BUF_CP * $10^5$ / (N_32 * NC_CYL_NR * 3)
else  MFF_CPS = 0
endif
MFF_ADD_CYL_CP = MFF_CPS
MFF_ADD_LAM_CP           = 0
TQI_ADD_CP           = 0
end
```

28.11 EVAC lambda deviation and canister load

DI version

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CP_MEM	O/V	0H	CP_NOT_ACT	1	[-]
		1H	NO_PURGE		
		2H	RAMP_TO_NO_PURGE		
		3H	WAIT_RAMP_OPEN		
		4H	MIN_PURGE		
		5H	-		
		6H	-		
		7H	-		
		8H	RAMP_OPEN		
		9H	RAMP_OPEN_FAST		
		AH	MAX_PURGE		
		BH	RAMP_CLOSE		
locally memorised version of STATE_CP					
STATE_CLL_DEAC_CP_MEM	-	0H	CP_NO_DEAC	1	[-]
		1H	RAMP_OPEN_DEAC		
		2H	MAX_PURGE_DEAC		
		3H	RAMP_FAST_DEAC		
locally memorised version of STATE_CLL_DEAC_CP					
LV_FAC_LAM_SHIFT_CP	O/V	0...1H	0...1	1	[-]
Logical value for lambda control correction during RAMP_OPEN operation					
LV_FAC_LAM_SHIFT_CP_AVL	O/V	0...1H	0...1	1	[-]
=1: At least one lambda control shift during RAMP_OPEN done					
LV_CP_AFL	O/V	0...1H	0...1	1	[-]
copy of (bench-selective) LV_AFL (0: rich; 1: lean)					
LV_CP_NEW_RAMP_OPEN	O/V	0...1H	0...1	1	[-]
Request for new Ramp Open, exit of normal purge					
MFF_CP	O/V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Fuel flow from the ACF					
MFF_ADD_CP_KGH	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Additive adaptive fuel flow from the ACF for the injection fuel mass correction					
MFF_ADD_CP_OLD	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
stored value of MFF_ADD_CP_OLD on disabled RAMP_OPEN					
MFF_LAM_CP	V	8000...7FFFH	-4...3.99987	0.1221e-3	[kg/h]
Fuel flow lambda control part from the ACF					
CL	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Canister load					
CL_MMV	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Moving mean value of the canister load CL					
FAC_LAM_0_CP	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Start value of the lambda control correction FAC_LAM_CP before the CPS is opened					
FAC_LAM_CP	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lambda control correction value used for canister load calculation					
FAC_LAM_DIF_CP	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lambda control deviation with active canister purge					
FAC_LAM_SHIFT_CP	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Correction value for lambda control correction during RAMP_OPEN operation					

CRLC_FAC_LAM_MV_MMV_CP	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
Filter constant for FAC_LAM_MV_MMV_CP calculation					
CRLC_CL	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
temporary variable: correlation constant for CL_MMV calculation					
CRLC_CL_TMP_INI	-	0...FFFFH	0...0.99998	0.0153e-3	[-]
temporary variable: init value for CRLC_CL_TMP in RAMP_OPEN					
CRLC_CL_TMP_0	-	0...FFFFH	0...0.99998	0.0153e-3	[-]
temporary variable: target value for CRLC_CL in RAMP_OPEN					
LV_FAC_LAM_DIF_CP_TMP	-	0...1H	0...1	1	[-]
temporary variable: FAC_LAM_DIF_CP limits reached in RAMP_OPEN					
LV_CRLC_CL_TRA	-	0...1H	0...1	1	[-]
flag: transition of fast to slow filter in MAX_PURGE ongoing					
LV_LAMB_AFS_CP	O/V	0...1H	0...1	1	[-]
Homogeneous lambda 1 combustion request					
LAMB_MMV_CP	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Filtered value of Lambda for canister purge					
LAMB_LPF_CP	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Low pass flter Lambda for canister purge					
T_RAMP_OPEN_STAB	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter for cl stabilisation during RAMP_OPEN					
CL_MMV_DYW_MIN	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Minimum value of the dynamic window during RAMP_OPEN					
CL_MMV_DYW_MAX	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum value of the dynamic window during RAMP_OPEN					
RATE_FLOW_DLY	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Relative flow delay (FLOW_DLY_CP / MAF_DLY_MMV_CP)					
FAC_LAM_GRD	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Gradient of lambda control output					
FAC_LAM_GRD_MMV	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Moving mean value of lambda gradient					
FAC_CRLC_CL	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Reduction factor at CL filter constant determination					
CL_MMV_MEM_CL_MDL	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Last stored CL_MMV value before FAC_LAM_GRD_MMV exceeds a threshold					
CL_MMV_MEM_CL_MDL_OLD	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Old stored CL_MMV value before FAC_LAM_GRD_MMV exceeds a threshold					
LV_CL_MMV_CAL_ACT	O/V	0...1H	0...1	1	[-]
Conditions for calculation of CL_MMV are fulfilled					
AMP_ST_PRS_MEM_CP	V	0...FFFFH	0...5434	0.0829175	[hPa]
Ambient preasure after reset					
AMP_DELTA_IT_CP	V/O	8000...7FFFH	-2717.04145...2716.95854	0.0829175	[hPa]
Ambient preasure difference between AMP and AMP at reset					
LAMB_MMV_0_CP	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Filtered value of Lambda for canister purge at start of RO					
LAMB_LPF_0_CP	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Low pass flter Lambda for canister purge at start of RO					
LAMB_LPF_DIF_IT_CP	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Low pass flter Lambda for canister purge at start of RO					
LAMB_MMV_0_TMP_CP	-	0...7FFFH	0...31.99902	0.9766e-3	[-]
Temp. sampling point for calculation of LAMB_MMV_COR_CP					
LAMB_MMV_COR_CP	V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Estimated lambda value for correction of FAC_LAM_0_CP					
LAMB_MMV_0_GRD_NEW_CP	-	0...FFFFH	0...1.99996	0.0305e-3	[-]
Actually lambda value for correction of FAC_LAM_0_CP					
LAMB_MMV_0_GRD_CP	V/S	0...FFFFH	0...1.99996	0.0305e-3	[-]
Sampling point for straight line for correction of FAC_LAM_0_CP					
T_LAMB_MMV_COR_UPD_CP	V	1...FFFFH	0.02...1310.7	0.02	[s]
Timer since last adaption of correction straight line for FAC_LAM_0_CP					

T_CP_PER_RUN_CP	V	1...FFFFH	0.02...1310.7	0.02	[s]
Time of actual purge period					
LV_CP_PER_END_REQ_S_CP	V	0...1H	0...1	1	[-]
Request to change to no purge					
LV_CL_GRD_CHK_MAX	V	0...1H	0...1	1	[-]
CL value conformity check exceeds					
ABC_LAMB_AFS_AMP_CP	V	8000...7FFFH	-32768...32767	1	[-]
anti bounce counter for setting LV_LAMB_AFS_CP depending on AMP					
ABC_LAMB_AFS_T_AST_CP	V	8000...7FFFH	-32768...32767	1	[-]
anti bounce counter for setting LV_LAMB_AFS_CP depending on T_AST					
LV_LAMB_AFS_T_AST_CP	O/V	0...1H	0...1	1	[-]
Homogeneous lambda 1 combustion request by threshold f(T_AST)					
LV_LAMB_AFS_AMP_CP	O/V	0...1H	0...1	1	[-]
Homogeneous lambda 1 combustion request by threshold f(delta_AMP)					
T_LAMB_AFS_T_AST	V	0...FFFFH	0...1310.7	0.02	[s]
Time for homogenous engine mode request					
LV_CP_NEW_RAMP_OPEN_FAST	O/V	0...1H	0...1	1	[-]
Request for new Ramp Open Fast or Ramp Open, exit of normal purge					
MFF_SP_0_CP	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint at initialisation of FAC_LAM_0_CP					
N_32_0_CP	V	0...FFH	0...8160	32	[rpm]
Engine speed at initialisation of FAC_LAM_0_CP					
LV_CP_OPP_WIN_1	O/V	0...1H	0...1	1	[-]
Operating point exceed small window					
LV_CP_OPP_WIN_2	O/V	0...1H	0...1	1	[-]
Operating point exceed large window					
T_LAMB_AFS_AMP_VLD_CP	V	0...FFFFH	0...1310.7	0.02	[s]
Timer that limits homogeneous request by CL_MMV evaluation for AMP					
T_LAMB_AFS_T_AST_VLD_CP	V	0...FFFFH	0...1310.7	0.02	[s]
Timer that limits homogeneous request by CL_MMV evaluation for T_AST					

Input data:

MAF_CYL	LV_FAC_LAM_SHIFT_CP_END[NC_CBK_EX_NR]	STATE_CP	LV_STATE_MEM_CP_CLL
LAMB_SP_HOM[NC_CBK_EX_NR]	FAC_LAM_MV_MMV_CP[NC_CBK_EX_NR]	LV_CP_ACT	STATE_CLL_DEAC_CP
LV_AFL[NC_CBK_EX_NR]		LV_CP_CLL	FLOW_DLY_MMV_CP
LV_HOM_ACT	LAMB_LS_UP[NC_CBK_EX_NR]	FLOW_CPS	MAF_DLY_MMV_CP
LV_HOM_AFL_ACT	LAMB_SP_FIL_S[NC_CBK_EX_NR]	PRS_CPS	LV_CP_DYW
CL_MDL	LV_CL_MDL_ACT	ERR_SYM_EL_CPS	IP_T_DLY_CL_MMV_DYW
FLOW_DLY_CP	CPPWM_CPS	AMP	LV_ERR_AMP
LV_END_DIAG_AMP	STATE_OPM_ENG_CP	LV_STATE_OPM_CHG_S_CP	LV_STATE_OPM_CHG_HOM_CP
VS	MFF_SP_MV	LV_CL_MDL_ENA	LV_FLOW_TAR_COR_CP
LV_CPS_AD_HOM_REQ	T_AST_CP	LV_C_FAC_LAM_0_CP_WIDE	N_32

Input data table order: External inputs – Inputs from other EVAC modules

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_CL_CRLC_RAMP_OPEN	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correlation constant for CL-filter for RAMP_OPEN					
IP_CL_CRLC_OPEN_FAST	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correlation constant for CL-filter for RAMP_OPEN_FAST					
IP_CL_CRLC	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correlation constant for CL-filter for MAX_PURGE / RAMP_CLOSE					
IP_CL_CRLC_INI	3	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_CL_MMV_1_EVAC	3	0...FFFFH	0...1.99996	0.0305e-3	[-]
Initialization value for correlation factor value in MAX_PURGE					
IP_CRLC_FAC_LAM_MV_MMV_CP_CLL	6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
correlation constant for FAC_LAM_MV_MMV_CP calculation in case of Purge in closed loop					
IP_CRLC_FAC_LAM_MV_MMV_CP_OPL	6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
correlation constant for FAC_LAM_MV_MMV_CP calculation in case of Purge in open loop					
IP_NR_STEP_CL	3	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_CL_MMV_1_EVAC	3	0...FFFFH	0...1.99996	0.0305e-3	[-]
Number of step to change of correlation factor value					
C_CL_MAX_CP	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum limit for canister load CL					
C_CL_ST_CP	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Initialization value for canister load CL					
C_RAF_CLC_CP	1	0...FFFFH	0...255.99609	3.9063e-3	[-]
stoiciometric fuel constant: ca. 14,7					
C_FAC_LAM_DIF_MAX_CP	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Maximum lambda deviation for injection time correction during RAMP_OPEN					
C_CL_FAC_LAM_DIF_MAX	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper limit on FAC_LAM_DIF_CP value variation					
C_CL_FAC_LAM_DIF_MIN	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lower limit on FAC_LAM_DIF_CP value variation					
C_PRS_MAX_2_CPS	1	8000...7FFFH	-2717.04145... 2716.95854	0.0829175	[hPa]
Pressure difference limit (environment-intake manifold) for reactivation of the canister purge					
C_FLOW_MIN_CPS	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Limit on FLOW_CPS value for CL_MMV calculation					
C_CL_DYW_MAX_CP	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum limit for canister load cl difference					
C_FAC_LAM_DIF_CP_AS	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Application system value to apply a lambda deviation					
C_CPPWM_MIN_CL	1	0...FFH	0...99.60937	0.390625	[%]
minimum valve pwm signal to enable CL calcualation					
C_T_DLY_MAX_RAMP_OPEN	1	1...FFFFH	0.1...6553.5	0.1	[s]
Time out value during RAMP_OPEN operation					
C_CL_MMV_DYW	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Dynamic window on CL_MMV value during RAMP_OPEN					
IP_CRLC_LAMB_LPF_CP	6	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_LAMB_LPF_CP_IP_CRLC_CP	6	0...7FFFH	0...31.99902	0.9766e-3	[-]
Correlation constant for LAMB_MMV_CP calculation					
IP_CRLC_MMV_CP	6	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_LAMB_MMV_CP_IP_CRLC_CP	6	0...7FFFH	0...31.99902	0.9766e-3	[-]
Correlation constant for LAMB_MMV_CP calculation					
C_FAC_LAM_0_MIN	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Minimum limit for FAC_LAM_0_CP					
C_FAC_LAM_0_MAX	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]

Maximum limit for FAC_LAM_0_CP					
IP_FAC_CRLC_CL	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_RATE_IP_FAC_CRLC_CL	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
Reduction Factor at very low flow rates					
IP_FAC_CPPWM	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_CPPWM_IP_FAC_CPPWM	6	0...FFH	0...99.60937	0.390625	[%]
Reduction Factor at very low flow rates, dependent on CPPWM					
IP_FAC_LAM_GRD_MMV	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_FAC_IP_FAC_LAM_GRD_MMV	6	0...FFFFH	-50...49.99847	1.5259e-3	[%]
Reduction Factor dependent on mean value of gradient of lamda control output					
C_CRLC_FAC_LAM_GRD	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for lambda gradient					
C_FAC_LAM_THD_CL_MEM	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Threshold for FAC_LAM_GRD to save the last CL_MMV value					
C_FLOW_MIN_CPS_RO	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Limit on FLOW_CPS value for CL_MMV calculation (during RO and ROF)					
IP_CL_MMV_HOM_PRS_UP_CP	4	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_AMP_DELTA_CP_IP_CL	4	8000...7FFFH	-2717.04145... 2716.95854	0.0829175	[hPa]
Limit on canister load to stay in homogenous mode depending on rising ambient preasure difference					
IP_CL_MMV_HOM_PRS_DOWN_CP	4	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_AMP_DELTA_CP_IP_CL	4	8000...7FFFH	-2717.04145... 2716.95854	0.0829175	[hPa]
Limit on canister load to go to stratified mode depending on falling ambient preasure difference					
C_LAMB_LPF_AD_MIN_CP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Minimum value of LAMB_LPF_AD for learning the straight line for LAMB_MMV_0_CP					
C_LAMB_LPF_AD_MAX_CP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Maximum value of LAMB_LPF_AD for learning the straight line for LAMB_MMV_0_CP					
C_LAMB_LPF_CLC_GRD_MIN_CP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Treshold for low pass flter Lambda for canister purge to activate adaption of the correction for FAC_LAM_0_CP					
C_CRLC_LAMB_MMV_0_GRD_CP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for straight line for FAC_LAM_0_CP correction					
IP_T_CP_PER_END_CP	4*4	1...FFFFH	0.02...1310.7	0.02	[s]
LDP_LAMB_DIF_IP_T_END_CP	4	0...7FFFH	0...31.99902	0.9766e-3	[-]
LDP_T_LAMB_COR_UPD_IP_T_END_CP	4	0...FFFFH	0...1310.7	0.02	[s]
Maximum time to run a purge period in stratified mode					
IP_T_CP_PER_END_COR_CP	3	0...FFH	0...1.99218	0.0078125	[-]
LDPM_CL_MMV_1_EVAC	3	0...FFFFH	0...1.99996	0.0305e-3	[-]
Canister load correction of maximum time to run a purge period in stratified mode					
C_LAMB_AFS_AMP_MAX_CP	1	0...8000H	0...32768	1	[-]
Maximum value for the anti bounce counter for setting LV_LAMB_AFS_CP					
C_ABC_INC_LAMB_AFS_AMP_CP	1	0...8000H	0...32768	1	[-]
anti bounce counter incrementation for setting LV_LAMB_AFS_CP					
C_ABC_DEC_LAMB_AFS_AMP_CP	1	0...8000H	0...32768	1	[-]
anti bounce counter decrementation for setting LV_LAMB_AFS_CP					
C_LAMB_AFS_T_AST_MAX_CP	1	0...8000H	0...32768	1	[-]
Maximum value for the anti bounce counter for setting LV_LAMB_AFS_CP					
C_ABC_INC_LAMB_AFS_T_AST_CP	1	0...8000H	0...32768	1	[-]
anti bounce counter incrementation for setting LV_LAMB_AFS_CP					
C_ABC_DEC_LAMB_AFS_T_AST_CP	1	0...8000H	0...32768	1	[-]
anti bounce counter decrementation for setting LV_LAMB_AFS_CP					
ID_CL_MMV_HOM_T_AST_UP_CP	3	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_T_AST_CP_ID_CL	3	0...FFFFH	0...6553.5	0.1	[s]
Limit on canister load to stay in homogenous mode depending on rising ambient preasure difference					
ID_CL_MMV_HOM_T_AST_DOWN_CP	3	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_T_AST_CP_IP_CL	3	0...FFFFH	0...6553.5	0.1	[s]
Limit on canister load to stay in homogenous mode depending on T_AST					
C_VS_MIN_LAMB_AFS	1	0...FFH	0...255	1	[km/h]
Minimum speed for homogenous engine mode request					
C_T_LAMB_AFS_T_AST	1	0...FFFFH	0...1310.7	0.02	[s]

Minimum time for homogenous engine mode request at low CL					
IP_CL_CRLC_RAMP_OPEN_2	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correlation constant for CL-filter for RAMP_OPEN at STATE_OPM_ENG_CP != MODE0					
IP_CL_CRLC_OPEN_FAST_2	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correlation constant for CL-filter for RAMP_OPEN_FAST at STATE_OPM_ENG_CP != MODE0					
IP_CL_CRLC_2	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_MAF_CYL_1_EVAC	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correlation constant for CL-filter for MAX_PURGE at STATE_OPM_ENG_CP != MODE0					
IP_CL_CRLC_INI_2	3	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_CL_MMV_1_EVAC	3	0...FFFFH	0...1.99996	0.0305e-3	[-]
Initialization value for correlation factor value in MAX_PURGE at STATE_OPM_ENG_CP != MODE0					
IP_MFF_SP_OPP_MAX	12*12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_N_32_4_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_3_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Upper MFF_SP threshold for operating point window					
IP_MFF_SP_OPP_MIN	12*12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_N_32_4_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_3_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower MFF_SP threshold for operating point window					
IP_N_32_OPP_MAX	12*12	0...FFH	0...8160	32	[rpm]
LDPM_N_32_4_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_3_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Upper N_32 threshold for operating point window					
IP_N_32_OPP_MIN	12*12	0...FFH	0...8160	32	[rpm]
LDPM_N_32_4_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_3_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower N_32 threshold for operating point window					
C_FAC_OPP_CP_1	1	0...FFFFH	0...1	0.0153e-3	[-]
Factor to reduce the MFF and N thresholds for the smaller OPP window					
C_T_LAMB_AFS_T_AST_VLD_CP	1	0...FFFFH	0...1310.7	0.02	[s]
Maximum time for homogeneous request by CL_MMV evaluation for T_AST					
C_T_LAMB_AFS_AMP_VLD_CP	1	0...FFFFH	0...1310.7	0.02	[s]
Maximum time for homogeneous request by CL_MMV evaluation for AMP					
C_CL_ST_MIN_CP	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Minimum Initialization value for canister load CL					
C_CL_ST_MAX_CP	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum initialization value for canister load CL					
C_CL_ST_FAC_CP	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Weight the CL_MMV deviation for initialization of CL/ CL_MMV at entering RO					
C_FAC_LAM_0_MIN_WIDE	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Minimum limit for FAC_LAM_0_CP to sharpen FSD					
C_FAC_LAM_0_MAX_WIDE	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Maximum limit for FAC_LAM_0_CP to sharpen FSD					
IP_MFF_SP_OPP_MAX_1	12*12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_N_32_4_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_3_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Upper MFF_SP threshold for operating point window for 2nd operation mode					
IP_MFF_SP_OPP_MIN_1	12*12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_N_32_4_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_3_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower MFF_SP threshold for operating point window for 2nd operation mode					
IP_N_32_OPP_MAX_1	12*12	0...FFH	0...8160	32	[rpm]
LDPM_N_32_4_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_3_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Upper N_32 threshold for operating point window for 2nd operation mode					
IP_N_32_OPP_MIN_1	12*12	0...FFH	0...8160	32	[rpm]
LDPM_N_32_4_EVAC	12	0...FFH	0...8160	32	[rpm]
LDPM_MFF_SP_3_EVAC	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower N_32 threshold for operating point window for 2nd operation mode					

C_N_32_MIN_OPP_WIN_CP	1	0...FFH	0...8160	32	[rpm]
N threshold below the usage of CL model for operation window is not used					
C_MFF_SP_MIN_OPP_WIN_CP	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
MFF threshold below the usage of CL model for operation window is not used					
C_PRS_MAX_2_CPS_1	1	8000...7FFFH	-2717.04145... 2716.95854	0.0829175	[hPa]
Pressure diff. limit (environment-intake manifold) for reactivation of the canister purge for ENG_MODE not 0					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_LAM_CBK_CPS	-	0...1H	0...1	1	[-]
Lambda controller selection (0 -> bank 0 , 1 -> bank1)					
LC_CL_FIL_INI_TRA_DEAC_CLL	-	0...1H	0...1	1	[-]
Correlation factor calculation in case of NO_PURGE or MIN_PURGE -> MAX_PURGE transition					
LC_LAMB_AFS_CP_CPS_SCG	-	0...1H	0...1	1	[-]
switch to force hom1 operation if cps valve short circuit to ground error present					
LC_INIT_CL_CRLC_RAMP_OPEN_FAST	-	0...1H	0...1	1	[-]
Correlation factor calculation in case of RAMP_OPEN_FAST -> MAX_PURGE transition					
C_STATE_LAMB_SEL_S_CP	-	0H 1H 2H	BANK_0 BANK_1 MEAN_VALUE	1	[-]
Lambda controller selection for stratified mode					
LC_STATE_OPM_CHG_S_CP	-	0...1H	0...1	1	[-]
Enabels setting of LV_CP_NEW_RAMP_OPEN at LV_STATE_OPM_CHG_S_CP =1					
LC_STATE_OPM_CHG_HOM_CP	-	0...1H	0...1	1	[-]
Enabels setting of LV_CP_NEW_RAMP_OPEN at LV_STATE_OPM_CHG_HOM_CP =1					
LC_FAC_LAM_0_COR_ACT_CP	1	0...1H	0...1	1	[-]
Enables the FAC_LAM_0_CP correction for stratified engine operation					

General information:

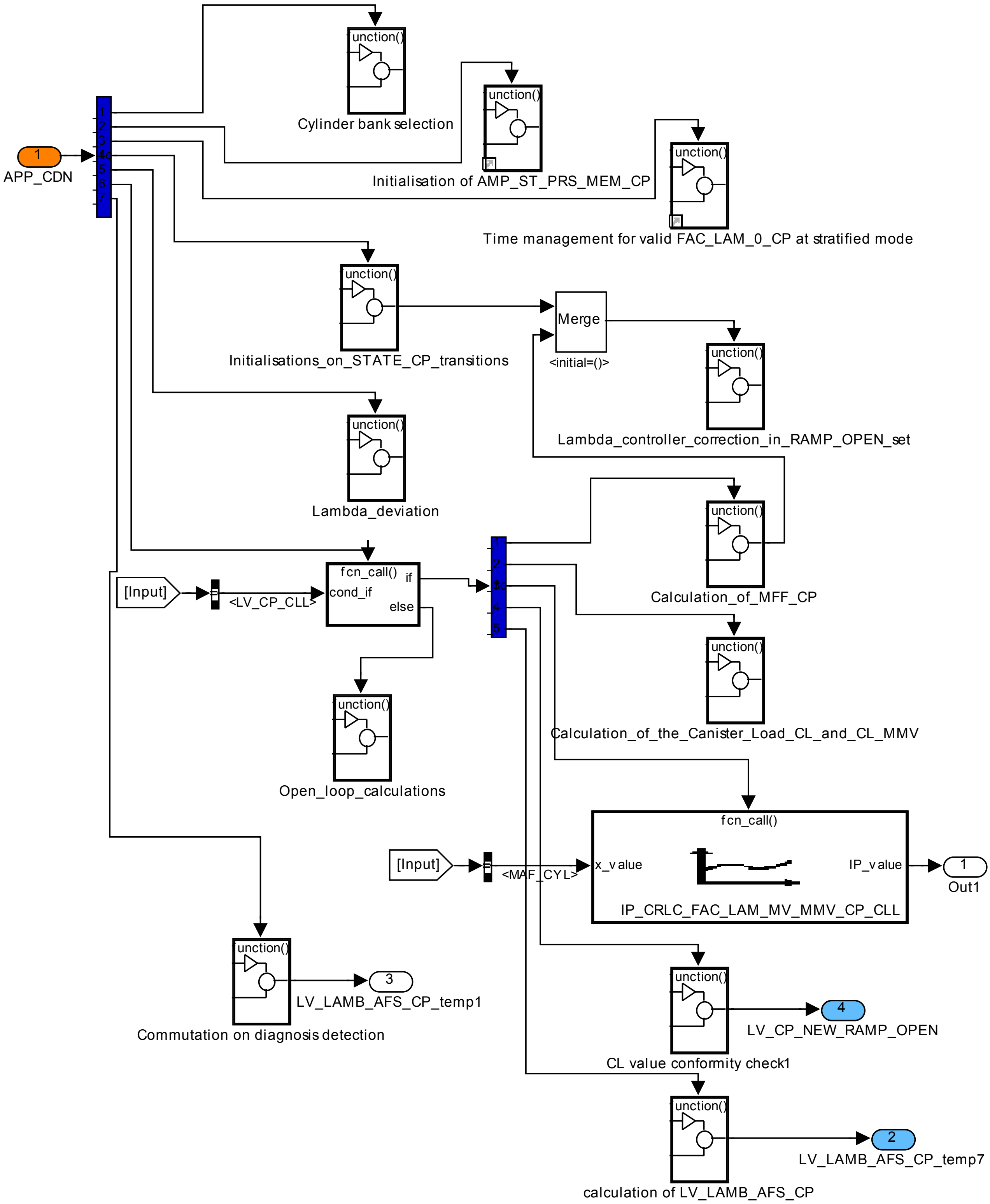
This module calculates the lambda deviation caused by the opening of the CPS valve and a canister load value using the lambda deviation value. The canister load is the concentration of fuel vapors within the mixture flow coming from the ACF.

The opening period cppwm of the CPS is controlled in dependence with the canister load CL.

Application conditions:

Initialisation: at RST, IGKON or deactivation
all values = 0
CL = C_CL_ST_CP
CL_MMV = C_CL_ST_CP
At first initialisation or NVMY error: LAMB_MMV_0_GRD_CP= 1
Recurrence: **20** ms
Activation: LV_CP_ACT = 1
Deactivation: LV_CP_ACT = 0

Signal flow diagram:



Formula section:

if LC_LAM_CBK_CPS=0

then $idx\_cp = 1$

else $idx\_cp = 2$

```
endif
call  "Initialisation of AMP_ST_PRS_MEM_CP"
call  “Initialisations on STATE_CP transitions”
call  “Lambda deviation”
$T\_LAMB\_MMV\_COR\_UPD\_CP_n = T\_LAMB\_MMV\_COR\_UPD\_CP_{n-1} + 0.02$
if    LV_CP_CLL          = 1                         // “Normal purge” active
and   LV_CL_MDL_ACT      = 0
then  call  "Time management for valid FAC_LAM_0_CP at stratified mode"
      call  “calculation of MFF_CP”
      call  “calculation of the Canister Load CL, CL_MMV”
      (moved to lambda deviation)
      call  “CL value conformity check”
else  call  “Open loop calculations”
endif
if    LV_CP_CLL          = 1
      call  "CL_MMV evaluation for homogeneous engine operation mode request"
endif
call  "Calculation of LV_LAMB_AFS_CP"
call  "Engine operating point window"
call  "Request for new RAMP OPEN and RAMP OPEN FAST"
end
```

28.11.1 INITIALIZATION of AMP_ST_PRS_MEM_CP

Save the first valid ambient preasure AMP once after reset:

Application conditions:

Initialisation: AMP_ST_PRS_MEM_CP = 0 at RST
Recurrence: **100** ms
Activation: --
Deactivation: --

```
if    LV_ERR_AMP = 0 and LV_END_DIAG_AMP = 1 and AMP_ST_PRS_MEM_CP = 0
then  AMP_ST_PRS_MEM_CP = AMP
end
```

28.11.2 INITIALIZATIONS on STATE_CP transitions

The following submodules are to be calculated on transitions between the different states of canister purge control.

To follow a transition in CP control module, STATE_CP and STATE_CLL_DEAC_CP have to be memorised in this module.

At the case of STATE_OPM_ENG_CP != MODE_0 the intialization value for CL and CL_MMV is corrected in the direction of the last known CL_MMV. The correction can be weighted and limited.

Formula section:

```
if      STATE_CP ≠ STATE_CP_MEM              // transition detected
then    if      STATE_CP_MEM = RAMP_OPEN
        then    call ”INITIALIZATION-VALUES on leaving RAMP_OPEN”
                if      STATE_CP = MAX_PURGE
                and     FAC_LAM_DIF_CP > 0
                then    call “lambda controller correction in RAMP_OPEN (set)”
                endif
        endif
        case selection on STATE_CP:
        RAMP_OPEN:          call “INITIALIZATION-VALUES on entering RAMP_OPEN”
        RAMP_OPEN_FAST:     call “INITIALIZATION-VALUES on ent. RAMP_OPEN_FAST”
        MAX_PURGE:          call “INITIALIZATION-VALUES on entering MAX_PURGE”
        end case selection
endif
// memorisation for next recurrence
STATE_CP_MEM                 = STATE_CP
STATE_CLL_DEAC_CP_MEM        = STATE_CLL_DEAC_CP
end
```

28.11.2.1 INITIALIZATION-VALUES on leaving RAMP_OPEN

```
MFF_ADD_CP_OLD    = MFF_ADD_CP_KGH
end
```

28.11.2.2 INITIALIZATION-VALUES on entering RAMP_OPEN

```
if      STATE_CLL_DEAC_CP_MEM = RAMP_OPEN_DEAC
then                                 // Initialisations for disabled RAMP_OPEN
        CL                           = CL_MMV
        MFF_ADD_CP_KGH               = MFF_ADD_CP_OLD
else                                 // Standard-initialisations for RAMP_OPEN
        if      STATE_OPM_ENG_CP = STATE_ENG_0
        then    CL                   = C_CL_ST_CP
                CL_MMV               = C_CL_ST_CP
        else    CL_MMV               = MAX[MIN[C_CL_ST_CP+(CL_MMV-C_CL_ST_CP)
                            *C_CL_ST_FAC_CP, C_CL_ST_MAX_CP],C_CL_ST_MIN_CP]
                CL                   = CL_MMV
        endif
        CL_0_0                       = CL_MMV
        CL_0_1                       = 0
        CL_MMV_DYW_MAX               = 0
        CL_MMV_DYW_MIN               = FFFFH
        MFF_ADD_CP_KGH               = 0
        LV_FAC_LAM_SHIFT_CP_AVL = 0
        LV_FAC_LAM_SHIFT_CP     = 0


        if LV_HOM_ACT = 1
```

```
        then
            if LV_C_FAC_LAM_0_CP_WIDE = 1
            then FAC_LAM_0_CP =      max[C_FAC_LAM_0_MIN_WIDE,
                                 min[C_FAC_LAM_0_MAX_WIDE, FAC_LAM_CP]]
            else FAC_LAM_0_CP =
                max[C_FAC_LAM_0_MIN, min[C_FAC_LAM_0_MAX, FAC_LAM_CP]]
            endif
            LAMB_MMV_CP        = LAMB_LS_UP [idx_cp]
            LAMB_LPF_CP              = LAMB_SP_FIL_S [idx_cp]
        else
            N_32_0_CP                = N_32
            MFF_SP_0_CP              = MFF_SP_MV
            Case selection of C_STATE_LAMB_SEL_S_CP:
            BANK_0:
                LAMB_MMV_CP                  = LAMB_LS_UP [0]
                LAMB_LPF_CP                  = LAMB_SP_FIL_S [0]
            BANK_1:
                LAMB_MMV_CP                  = LAMB_LS_UP [1]
                LAMB_LPF_CP                  = LAMB_SP_FIL_S [1]
            MEAN_VALUE:
                LAMB_MMV_CP                  = SUM{LAMB_LS_UP [i=0 to
                                 NC_CBK_EX_NR]} /NC_CBK_EX_NR
                LAMB_LPF_CP                  = SUM{LAMB_SP_FIL_S  [i=0 to
                                 NC_CBK_EX_NR]} /NC_CBK_EX_NR
            End of case
            call "Initalisation at RO and ROF for FAC_LAM_0_CP correction
                at stratified mode"
            call "Calculation of the straight line for LAMB_MMV_COR_CP"
            call " Calculation of corrected FAC_LAM_0_CP in stratified"
    endif
        // RAMP_OPEN time out initialization
        T_RAMP_OPEN_STAB       = C_T_DLY_MAX_RAMP_OPEN
    endif
    end
```

28.11.2.3 INITIALIZATION-VALUES on entering RAMP_OPEN_FAST

```
    if      STATE_CLL_DEAC_CP_MEM = RAMP_FAST_DEAC
    then    --                          // Initialisations for disabled RAMP_OPEN_FAST
    else                                // Standard-initialisations for RAMP_OPEN_FAST
        CL_0_0                          = CL_MMV
        CL_0_1                          = 0
        if LV_HOM_ACT = 1
        then
            if LV_C_FAC_LAM_0_CP_WIDE = 1
            then FAC_LAM_0_CP =      max[C_FAC_LAM_0_MIN_WIDE,
                                 min[C_FAC_LAM_0_MAX_WIDE, FAC_LAM_CP]]
             else FAC_LAM_0_CP =
                max[C_FAC_LAM_0_MIN, min[C_FAC_LAM_0_MAX, FAC_LAM_CP]]
```

```
            endif
        else
            N_32_0_CP                     = N_32
            MFF_SP_0_CP                   = MFF_SP_MV
            call "Initalisation at RO and ROF for FAC_LAM_0_CP correction
                at stratified mode"
            call "Calculation of the straight line for LAMB_MMV_COR_CP"
            call " Calculation of corrected FAC_LAM_0_CP in stratified"
        endif
    endif
    CL                                = CL_MMV
    end
```

28.11.2.4 INITIALIZATION-VALUES on entering MAX_PURGE

```
    if   STATE_CLL_DEAC_CP_MEM     = MAX_PURGE_DEAC
    then CL                        = CL_MMV
    endif
    end
```

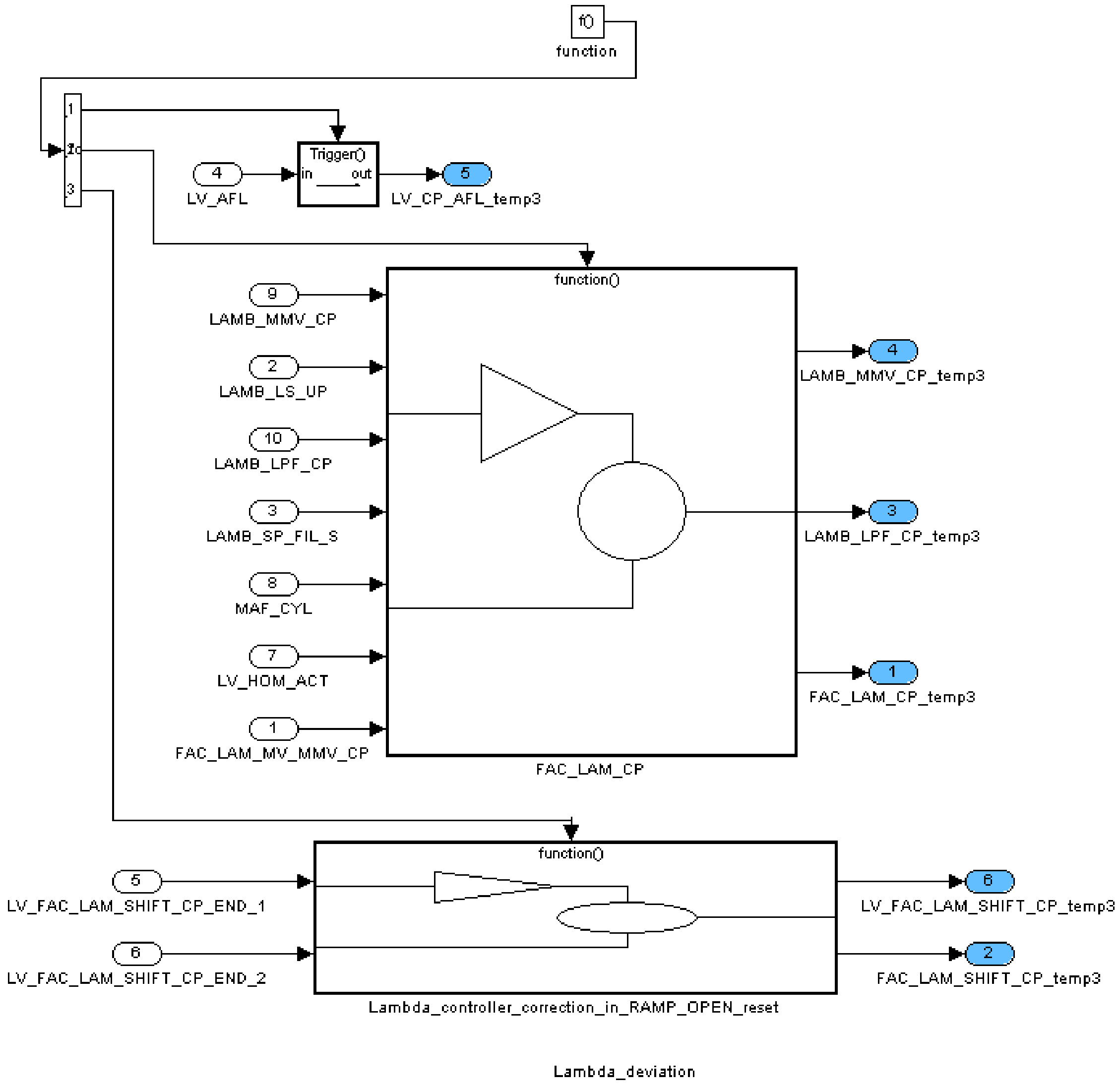
28.11.3 Lambda deviation

In case of two bank exhaust line configuration, the master-exhaust bank for evaporative emission control has to be defined using LC_LAM_CBK_CPS.

The additive constant C_FAC_LAM_DIF_CP_AS should be set to 0 in normal operation; it is only provided for validation purposes

The calculation of the filtered mean value of lambda controller output FAC_LAM_MV_MMV_CP is done inside LACO aggregate. Indeed, the filter constant for this calculation - CRLC_FAC_LAM_MV_MMV_CP - is provided by this module, depending on the STATE_CP.

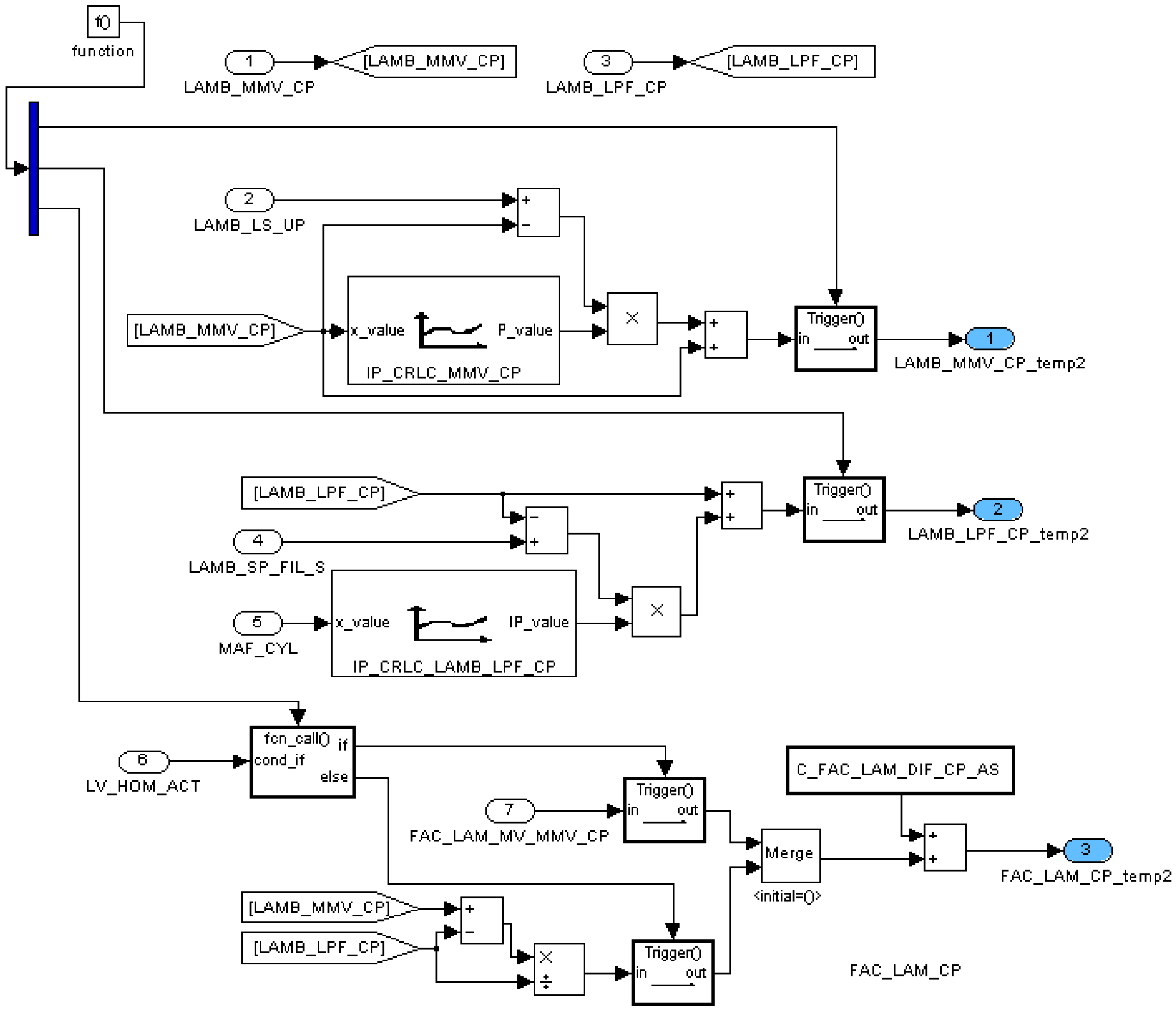
In open loop, a strong filtering is desired to obtain a stable value for FAC_LAM_0_CP on entering the closed loop. Within the closed loop operation, the filtering should be less severe in order to enable a quick learning of CL and fast reactions on lambda deviations caused by canister purge.



Formula section:

```
// determination of the filter constant for FAC_LAM_MV_MMV_CP
// this value is passed to LACO
if      LV_CP_CLL = 1                                          // "Normal purge" active
then CRLC_FAC_LAM_MV_MMV_CP   = IP_CRLC_FAC_LAM_MV_MMV_CP_CLL (MAF_CYL)
else  CRLC_FAC_LAM_MV_MMV_CP   = IP_CRLC_FAC_LAM_MV_MMV_CP_OPL (MAF_CYL)
endif
LV_CP_AFL = LV_AFL [idx_cp]

// determination of the lambda deviation
// FAC_LAM_MV_MMV_CP is received from LACO where it is calculated using the filter constant defined above
call   "FAC_LAM_CP"
call   "lambda controller correction in RAMP_OPEN (reset)"  // check in every state_cp
end
```

28.11.3.1 FAC_LAM_CP



```
if      LV_HOM_ACT = 1
then
        $LAMB\_MMV\_CP_n$ = $LAMB\_MMV\_CP_{n-1}$ +
                (LAMB_LS_UP [idx_cp] - $LAMB\_MMV\_CP_{n-1}$) *
                IP_CRLC_MMV_CP (LAMB_MMV_CP)
        $LAMB\_LPF\_CP_n$= $LAMB\_LPF\_CP_{n-1}$ +
                (LAMB_SP_FIL_S [idx_cp] - $LAMB\_LPF\_CP_{n-1}$) *
                IP_CRLC_LAMB_LPF_CP (MAF_CYL)
        FAC_LAM_CP = FAC_LAM_MV_MMV_CP [idx_cp]
else
        case selsction of C_STATE_LAM_SEL_S_CP:
        BANK_0:
                $LAMB\_MMV\_CP_n$ = $LAMB\_MMV\_CP_{n-1}$ +
```

```
                    (LAMB_LS_UP [0] - LAMB_MMV_CP$_{n-1}$) *
                  IP_CRLC_MMV_CP (LAMB_MMV_CP)
          LAMB_LPF_CP$_{n}$= LAMB_LPF_CP$_{n-1}$ +
                  (LAMB_SP_FIL_S [0] - LAMB_LPF_CP$_{n-1}$)
                  *IP_CRLC_LAMB_LPF_CP(MAF_CYL)
      BANK_1:
          LAMB_MMV_CP$_{n}$ = LAMB_MMV_CP$_{n-1}$ +
                    (LAMB_LS_UP [1] - LAMB_MMV_CP$_{n-1}$) *
                  IP_CRLC_MMV_CP (LAMB_MMV_CP)
          LAMB_LPF_CP$_{n}$= LAMB_LPF_CP$_{n-1}$ +
                  (LAMB_SP_FIL_S [1] - LAMB_LPF_CP$_{n-1}$) *
                  IP_CRLC_LAMB_LPF_CP (MAF_CYL)
      MEAN_VALUE:
          LAMB_MMV_CP$_{n}$ = LAMB_MMV_CP$_{n-1}$ +
                    (SUM[LAMB_LS_UP [i=0 to NC_CBK_EX_NR]]/NC_CBX_EX_NR
                    - LAMB_MMV_CP$_{n-1}$) *IP_CRLC_MMV_CP (LAMB_MMV_CP)
          LAMB_LPF_CP$_{n}$= LAMB_LPF_CP$_{n-1}$ +
                  (SUM[LAMB_SP_FIL_S [i=0 to NC_CBX_EX_NR]]/NC_CBX_EX_NR -
                  LAMB_LPF_CP$_{n-1}$) *IP_CRLC_LAMB_LPF_CP (MAF_CYL)
      end of case
      FAC_LAM_CP =(LAMB_MMV_CP - LAMB_LPF_CP) / LAMB_LPF_CP
  endif
  FAC_LAM_CP = FAC_LAM_CP + C_FAC_LAM_DIF_CP_AS
  end
```

28.11.3.2 Lambda controller correction in RAMP_OPEN (reset)

The flag requesting a shift of the lambda controller LV_FAC_LAM_SHIFT_CP is reset after all acknowledge flags from lambda controller are received.

Formula section:

```
if    LV_FAC_LAM_SHIFT_CP_END [i] = 1          // all benches have to be =1
then  LV_FAC_LAM_SHIFT_CP         = 0
      FAC_LAM_SHIFT_CP            = 0
endif
end
```

28.11.4 Calculation of MFF_CP

General information:

This calculation is calculated in RAMP_OPEN, RAMP_OPEN_FAST, MAX_PURGE and RAMP_CLOSE.

The fuel flow from the ACF when the CPS is opened causes a deviation of the lambda control. This fuel flow can be calculated as a function of the difference of the lambda control values with closed / opened CPS, the mass air flow MAF_DLY_MMV_CP, the lambda value LAMB_LPF_CP and a constant taking into account the stoiciometric ratio C_RAF_CLC_CP.

The mass air flow MAF_DLY_MMV_CP is the filtered value of maf_cp using a Padé-filter and a moving mean value calculation to consider the delay time between the mass air flow maf_cp and the FAC_LAM_CP calculation. The fuel flow calculation MFF_LAM_CP is the result of the lambda control deviation.

The additive adaptive fuel mass flow MFF_ADD_CP_KGH caused by the canister purge is also regarded in the calculation.

When the deviation of the lambda control FAC_LAM_DIF_CP exceeds C_FAC_LAM_DIF_MAX_CP during RAMP_OPEN operation or when the RAMP_OPEN operation is finished successfully, the fuel injection time is corrected by an additive correction value and the lambda controller value FAC_LAM_CP is corrected by the value of FAC_LAM_DIF_CP. FAC_LAM_DIF_CP is then set to 0 because the lambda deviation is corrected.

During MAX_PURGE. RAMP_OPEN_FAST and RAMP_CLOSE operation there is a continuous additive adaptive correction of the fuel. The additive fuel flow MFF_ADD_CP_KGH is calculated from the calculated mass flow through the CPS FLOW_CPS and the filtered canister load CL_MMV.

During RAMP_OPEN, the MFF_ADD_CP_KGH is updated as soon as FAC_LAM_DIF_CP exceeds the threshold C_FAC_LAM_DIF_MAX_CP and a last time on the transition RAMP_OPEN to MAX_PURGE, so MFF_ADD_CP_KGH will change periodically during RAMP_OPEN. If RAMP_OPEN is disabled for a short time, the current value of MFF_ADD_CP_KGH is stored in MFF_ADD_CP_OLD on leaving RAMP_OPEN and restored on re-entering RAMP_OPEN (see initialisations).

Formula section:

```
if LV_HOM_ACT = 0
then call " Calculation of corrected FAC_LAM_0_CP in stratified"
endif
FAC_LAM_DIF_CP = FAC_LAM_0_CP - FAC_LAM_CP // calculation of FAC_LAM_DIF_CP
if    STATE_CP = RAMP_OPEN                      // lambda controller correction
then  if LV_HOM_ACT = 1
      then if    FAC_LAM_DIF_CP > C_FAC_LAM_DIF_MAX_CP
           then MFF_ADD_CP_KGH = CL_MMV * FLOW_CPS
                call "Lambda controller correction in RAMP_OPEN (set)"
           endif
      else MFF_ADD_CP_KGH = CL_MMV * FLOW_CPS
           if    FAC_LAM_DIF_CP > C_FAC_LAM_DIF_MAX_CP
           then LV_FAC_LAM_SHIFT_CP_AVL = 1
           endif
      endif
else  MFF_ADD_CP_KGH = CL_MMV * FLOW_CPS
endif
call  "MFF_LAM_CP"
MFF_CP = MFF_LAM_CP + MFF_ADD_CP_KGH
end
```

28.11.4.1 Lambda controller correction in RAMP_OPEN (set)

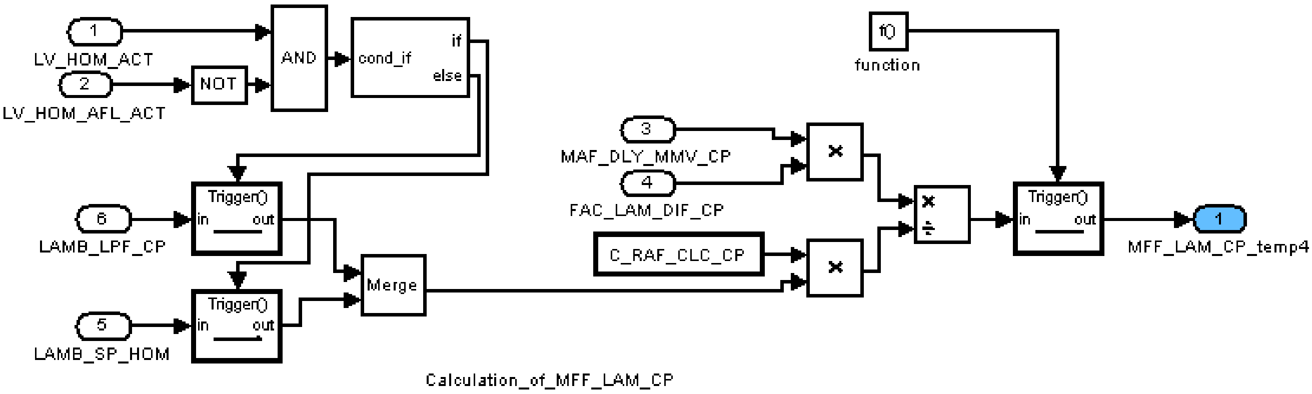
Called by:

- by RAMP_OPEN actions, when FAC_LAM_DIF_CP exceeds threshold
- On transition RAMP_OPEN to MAX_PURGE, if FAC_LAM_DIF_CP > 0

Formula section:

```
if      LV_FAC_LAM_SHIFT_CP     = 0
then    LV_FAC_LAM_SHIFT_CP     = 1
        FAC_LAM_SHIFT_CP        = FAC_LAM_DIF_CP
endif
FAC_LAM_DIF_CP                  = 0
LV_FAC_LAM_SHIFT_CP_AVL         = 1
end
```

28.11.4.2 MFF_LAM_CP



```
if      LV_HOM_ACT = 1 and LV_HOM_AFL_ACT= 0
then    MFF_LAM_CP =
          (MAF_DLY_MMV_CP * FAC_LAM_DIF_CP) / (C_RAF_CLC_CP * LAMB_SP_HOM)
else    MFF_LAM_CP =
          (MAF_DLY_MMV_CP * FAC_LAM_DIF_CP) / (C_RAF_CLC_CP * LAMB_LPF_CP)
endif
end
```

28.11.5 Calculation of the Canister Load CL, CL_MMV

General information:

This calculation is called in RAMP_OPEN, RAMP_OPEN_FAST, MAX_PURGE and RAMP_CLOSE.

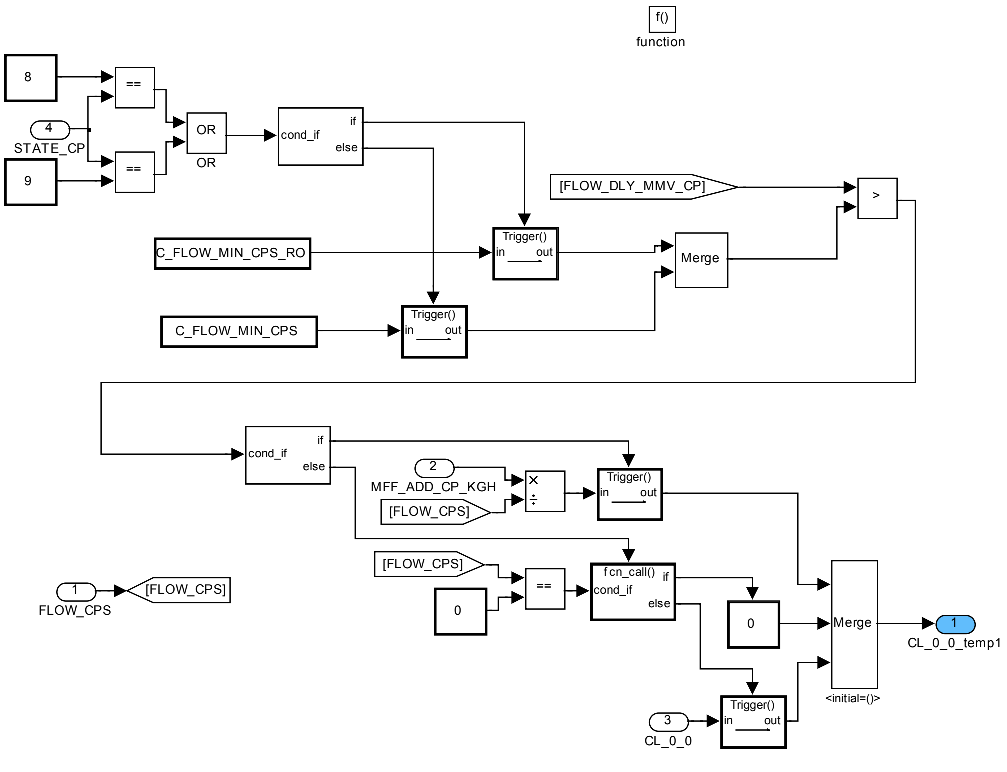
The load CL of the ACF is calculated as the ratio of the fuel flow MFF_LAM_CP and the flow FLOW_DLY_MMV_CP added to the ratio of the fuel flow MFF_ADD_CP_KGH and the flow FLOW_CPS.

The flow FLOW_DLY_MMV_CP is the filtered value of FLOW_CPS using a Padé-filter and a moving mean value calculation to consider the delay between the FLOW_CPS and the MFF_LAM_CP calculation. Those calculations are described in a separate chapter.

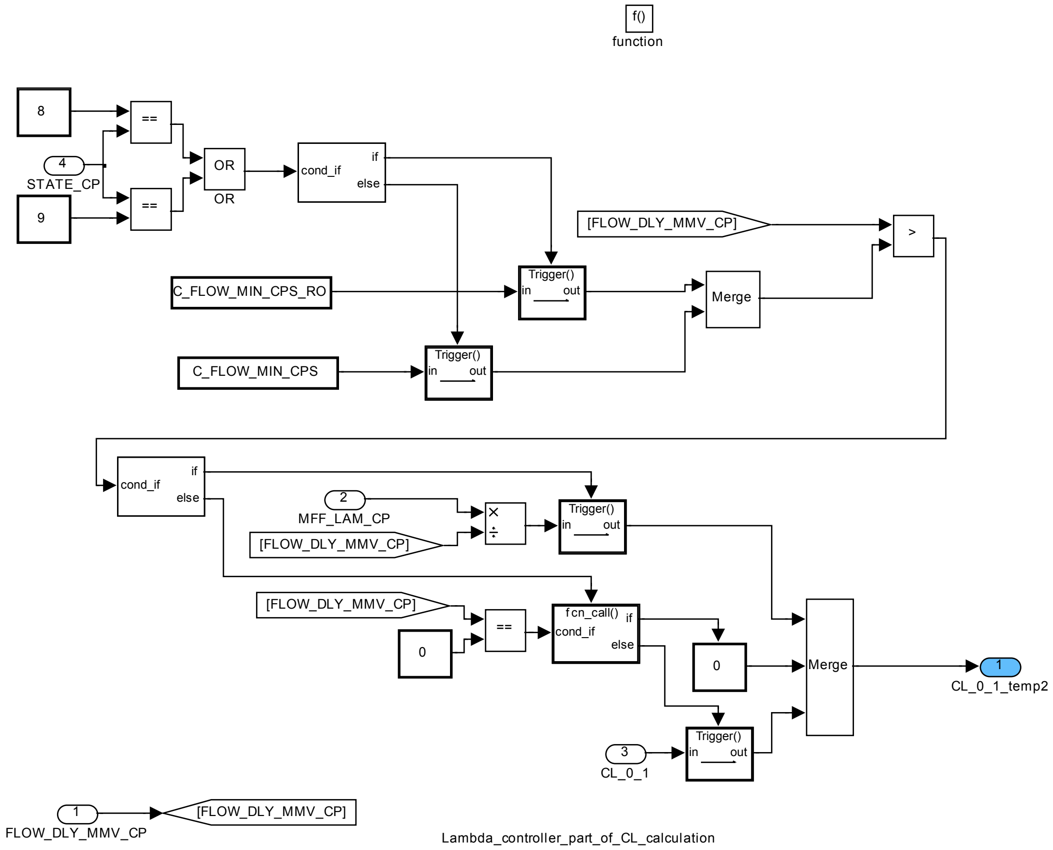
The minimum and maximum values of this ratio are limited by 0 and C_CL_MAX_CP. The value of CL is filtered using a moving mean value calculation. The value of CL is only calculated when the engine is operated at limited dynamics (LV_CP_DYW=1).

Application conditions:

Initialisation: all values = 0 at RST or IGKON
Recurrence: **100** ms
Activation: -- (called by superior block)
Deactivation: -- (called by superior block)



Injection_part_of_CL_calculation



Formula section:

```
if STATE_CP = RAMP_OPEN or RAMP_OPEN_FAST
then
        if      [ LV_CP_DYW            = 1                               //limited dynamics
        and     [(STATE_OPM_ENG_CP = ENG_MODE_0
                and PRS_CPS            > C_PRS_MAX_2_CPS) //minimum pressure difference
                or (STATE_OPM_ENG_CP != ENG_MODE_0
                and PRS_CPS            > C_PRS_MAX_2_CPS_1)] //minimum pressure diff.
        and FLOW_CPS > C_FLOW_MIN_CPS_RO //min. FLOW_CPS for RO/ROF
        and FLOW_DLY_MMV_CP > C_FLOW_MIN_CPS_RO // min FLOW at lambda
                                                            sensor for RO and
ROF
        and     CPPMW_CPS              > C_CPPWM_MIN_CL ]     // min valve opening
        then    LV_CL_MMV_CAL_ACT = 1            // calculation of CL/CL_MMV active
                CL_0_0 = MFF_ADD_CP_KGH / FLOW_CPS        // injection part of CL
                CL_0_1 = MFF_LAM_CP / FLOW_DLY_MMV_CP // lambda ctl. part of CL
                CL = min [CL_0_0 + CL_0_1, C_CL_MAX_CP]    // limitation of CL
                call "CL filter constant determination"          // CRLC_CL calculation
                $CL\_MMV_n = CL\_MMV_{n-1} + (CL - CL\_MMV_{n-1}) * CRLC\_CL$
                call   "calculation of time out"
        else    LV_CL_MMV_CAL_ACT  = 0
```

```
            CL_MMV$_{n}$ = CL_MMV$_{n-1}$
        endif
    else
        if    [ LV_CP_DYW          = 1                          // limited dynamics
        and   [(STATE_OPM_ENG_CP = ENG_MODE_0
              and PRS_CPS          > C_PRS_MAX_2_CPS) //minimum pressure difference
              or (STATE_OPM_ENG_CP != ENG_MODE_0
              and PRS_CPS          > C_PRS_MAX_2_CPS_1)] //minimum pressure diff.
        and   FLOW_CPS > C_FLOW_MIN_CPS          // minimum FLOW_CPS
        and   FLOW_DLY_MMV_CP > C_FLOW_MIN_CPS       // minimum FLOW_CPS at
                                                        lambda sensor
        and   CPPMW_CPS            > C_CPPWM_MIN_CL ]   // minimum valve opening
        then  LV_CL_MMV_CAL_ACT = 1                     // calculation of CL/CL_MMV
    active
              CL_0_0 = MFF_ADD_CP_KGH / FLOW_CPS       // injection part of CL
              CL_0_1 = MFF_LAM_CP / FLOW_DLY_MMV_CP // lambda control part of CL
              CL = min [CL_0_0 + CL_0_1, C_CL_MAX_CP]    // limitation of CL
              call “CL filter constant determination”      // CRLC_CL calculation
              CL_MMV$_{n}$ = CL_MMV$_{n-1}$ + (CL – CL_MMV$_{n-1}$) * CRLC_CL
              call   "calculation of time out"
        else  LV_CL_MMV_CAL_ACT  = 0
        CL_MMV$_{n}$ = CL_MMV$_{n-1}$
        endif
    endif
    end
```

28.11.5.1 Calculation of time out

(moved here from EVAC mng.)

```
If    LV_FLOW_TAR_COR_CP = 0
then  T_RAMP_OPEN_STAB = max [0, T_RAMP_OPEN_STAB–TA ]
endif
if    LV_FAC_LAM_SHIFT_CP_AVL
and [ CL_MMV < CL_MMV_DYW_MIN
or    CL_MMV > CL_MMV_DYW_MAX ]
then  CL_MMV_DYW_MIN   = CL_MMV -  C_CL_MMV_DYW
      CL_MMV_DYW_MAX  = CL_MMV + C_CL_MMV_DYW
      // timer re-initialisation
      T_RAMP_OPEN_STAB= IP_T_DLY_CL_MMV_DYW (MAF_CYL)
endif
end
```

28.11.5.2 CL filter constant determination

```
FAC_LAM_GRD          = (FAC_LAM_CP – FAC_LAM_CP$_{n-1}$)
FAC_LAM_GRD_MMV= FAC_LAM_GRD_MMV$_{n-1}$ + (FAC_LAM_GRD –
                                   FAC_LAM_GRD_MMV$_{n-1}$) * C_CRLC_FAC_LAM_GRD
if    FAC_LAM_GRD                < C_FAC_LAM_THD_CL_MEM
then  CL_MMV_MEM_CL_MDL_OLD = CL_MMV_MEM_CL_MDL
      CL_MMV_MEM_CL_MDL     = CL_MMV
```

```
RATE_FLOW_DLY     = FLOW_DLY_CP / MAF_DLY_MMV_CP
FAC_CRLC_CL  =  IP_FAC_CRLC_CL  (RATE_FLOW_DLY)  *  IP_FAC_CPPWM
(CPPWM_CPS) * IP_FAC_LAM_GRD (FAC_LAM_GRD_MMV)
case selection on STATE_CP:
RAMP_OPEN:
    if    STATE_OPM_ENG_CP = ENG_MODE_0
    then  CRLC_CL = IP_CL_CRLC_RAMP_OPEN (MAF_CYL) * FAC_CRLC_CL
    else  CRLC_CL = IP_CL_CRLC_RAMP_OPEN_2 (MAF_CYL) * FAC_CRLC_CL
RAMP_OPEN_FAST:
    if    STATE_OPM_ENG_CP = ENG_MODE_0
    then  CRLC_CL = IP_CL_CRLC_OPEN_FAST (MAF_CYL) * FAC_CRLC_CL
    else  CRLC_CL = IP_CL_CRLC_OPEN_FAST_2 (MAF_CYL) * FAC_CRLC_CL
RAMP_CLOSE:
    CRLC_CL = IP_CL_CRLC (MAF_CYL) * FAC_CRLC_CL
MAX_PURGE:
    if    STATE_OPM_ENG_CP = ENG_MODE_0
    then  CRLC_CL_TMP_0         = IP_CL_CRLC (MAF_CYL) * FAC_CRLC_CL
          CRLC_CL_TMP_INI       = IP_CL_CRLC_INI (CL_MMV)
    else  CRLC_CL_TMP_0         = IP_CL_CRLC_2 (MAF_CYL) * FAC_CRLC_CL
          CRLC_CL_TMP_INI       = IP_CL_CRLC_INI_2 (CL_MMV)
    endif
    LV_FAC_LAM_DIF_CP_TMP =
          [(FAC_LAM_DIF_CP   > C_CL_FAC_LAM_DIF_MAX)
    or    (FAC_LAM_DIF_CP   < C_CL_FAC_LAM_DIF_MIN)]

    if          STATE_CP_MEM = RAMP_OPEN                   // switch to init map
    or          [ STATE_CP_MEM = RAMP_OPEN_FAST and
                LC_INIT_CL_CRLC_RAMP_OPEN_FAST = 1]
    or         [(      STATE_CP_MEM = NO_PURGE
          or    STATE_CP_MEM = RAMP_TO_NO_PURGE
          or    STATE_CP_MEM = MIN_PURGE)
          and   LC_CL_FIL_INI_TRA_DEAC_CLL=1]
    or    [(LV_FAC_LAM_DIF_CP_TMP$_{n-1}$ = 0) and (LV_FAC_LAM_DIF_CP_TMP$_n$ = 1)]
    then  CRLC_CL$_n$ = CRLC_CL_TMP_INI
          LV_CRLC_CL_TRA = 1
    else  if    LV_CRLC_CL_TRA = 0
          then  CRLC_CL$_n$ = CRLC_CL_TMP_0
          else  CRLC_CL$_n$ = CRLC_CL$_{n-1}$
                      – (CRLC_CL_TMP_INI - CRLC_CL_TMP_0)
                      * IP_NR_STEP_CL (CL_MMV)
                if    CRLC_CL  $\le$ CRLC_CL_TMP_0
                then  CRLC_CL  = CRLC_CL_TMP_0
                      LV_CRLC_CL_TRA      = 0
                endif
          endif
    endif
end case selection
end
```

Note: All values of IP_CL_CRLC_INI have to be bigger than the highest value of IP_CL_CRLC; otherwise, the filter constant CRLC_CL will be unstable.

28.11.5.3 CL value conformity check :

General information:

This calculation is called in RAMP_OPEN, RAMP_OPEN_FAST and MAX_PURGE.

When the calculation of the canister load create a high difference between the previous value CL_{n-1} and the new value CL_n a transition to MIN_PURGE is requested by setting of LV_CL_GRD_CHK_MAX.

Formula section:

```
LV_CL_GRD_CHK_MAX = [abs [$CL_{n-1}$ - $CL_n$] > C_CL_DYW_MAX_CP ]
end
```

28.11.6 CL_MMV evaluation for homogeneous engine operation mode request

Homogeneous lambda one combustion is requested by setting of the bit information LV_LAMB_AFS_CP (see chapter "Calculation of LV_LAMB_AFS_CP"). Two possibilities are given. If the canister load CL_MMV is greater then the threshold IP_CL_MMV_HOM_PRS_UP_CP. The threshold is taken from a map depending on the difference between the ambient preasure at reset and the actual ambient preasure. The bit information is reset if the canister load falls below the threshold IP_CL_MMV_HOM_PRS_DOWN_CP. The setting and reseting of the bit LV_LAMB_AFS_CP is debounced by the counters ABS_LAMB_AFS_CP.
The other possibility is given for lower CL_MMV values, if the vehicle drives not in the city. There is the increasement for purge flow much bigger then driving in the city. During driving in the city the fuel consumption increase is too high to request homogenious mode.

Application conditions:

Initialisation: all values = 0 at RST or IGKON
Recurrence: **100** ms
Activation: -- (called by superior block)
Deactivation: -- (called by superior block)

Formula section:

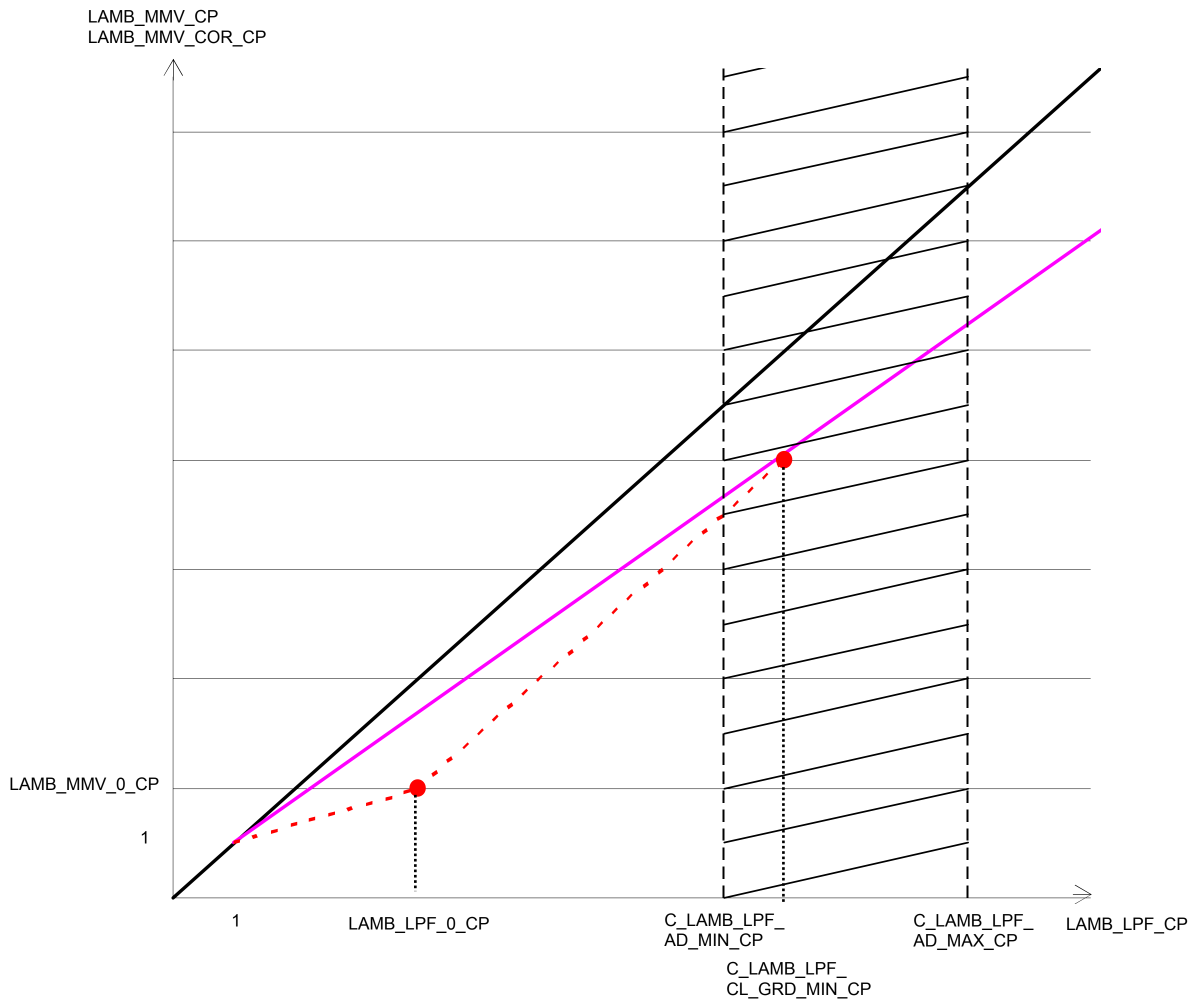
```
if LV_ERR_AMP = 0 and L_END_DIAG_AMP = 1
then AMP_DELTA_IT_CP = AMP - AMP_ST_PRS_MEM_CP
else AMP_DELTA_IT_CP = 0
endif


if      CL_MMV <= IP_CL_MMV_HOM_PRS_DOWN_CP(AMP_DELTA_IT_CP)
then    // under  the lower threshold, count ABC down
        ABC_LAMB_AFS_AMP_CP = MAX[(-1)*C_LAMB_AFS_AMP_MAX_CP,
                ABC_LAMB_AFS_AMP_CP- C_ABC_DEC_LAMB_AFS_AMP_CP]
        if ABC_LAMB_AFS_AMP_CP <= (-1)*C_LAMB_AFS_AMP_MAX_CP
```

```
        then LV_LAMB_AFS_AMP_CP = 0
        endif
else    if      CL_MMV > IP_CL_MMV_HOM_PRS_UP_CP(AMP_DELTA_IT_CP)
        then    // above the upper threshold, count ABC up
                ABC_LAMB_AFS_AMP_CP = MIN[C_LAMB_AFS_AMP_MAX_CP ,
                ABC_LAMB_AFS_AMP_CP+ C_ABC_INC_LAMB_AFS_AMP_CP]
                if      ABC_LAMB_AFS_AMP_CP >= C_LAMB_AFS_AMP_MAX_CP
                then    LV_LAMB_AFS_AMP_CP = 1
                endif
        else // between the theresholds, count ABC to zero
                if ABC_LAMB_AFS_AMP_CP > 0
                then    ABC_LAMB_AFS_AMP_CP =
                        MAX[0,ABC_LAMB_AFS_AMP_CP- C_ABC_DEC_LAMB_AFS_AMP_CP]
                else if ABC_LAMB_AFS_AMP_CP < 0
                        then    ABC_LAMB_AFS_AMP_CP =
                                MIN[0, ABC_LAMB_AFS_AMP_CP+
                                C_ABC_INC_LAMB_AFS_AMP_CP]
                        endif
                endif
        endif
endif

if      CL_MMV <= ID_CL_MMV_HOM_T_AST_DOWN_CP(T_AST_CP)
then    // under  the lower threshold, count ABC down
        ABC_LAMB_AFS_T_AST_CP = MAX[(-1)*C_LAMB_AFS_T_AST_MAX_CP,
                ABC_LAMB_AFS_T_AST_CP- C_ABC_DEC_LAMB_AFS_T_AST_CP]
        if ABC_LAMB_AFS_T_AST_CP <= (-1)*C_LAMB_AFS_T_AST_MAX_CP
        then LV_LAMB_AFS_T_AST_CP = 0
        endif
else    if      CL_MMV > ID_CL_MMV_HOM_T_AST_UP_CP(T_AST_CP)
        then    // above the upper threshold, count ABC up
                ABC_LAMB_AFS_T_AST_CP = MIN[C_LAMB_AFS_T_AST_MAX_CP ,
                ABC_LAMB_AFS_T_AST_CP+ C_ABC_INC_LAMB_AFS_T_AST_CP]
                if      ABC_LAMB_AFS_T_AST_CP >= C_LAMB_AFS_T_AST_MAX_CP
                then    LV_LAMB_AFS_T_AST_CP = 1
                endif
        else // between the theresholds, count ABC to zero
                if ABC_LAMB_AFS_T_AST_CP > 0
                then    ABC_LAMB_AFS_T_AST_CP =
                        MAX[0,ABC_LAMB_AFS_T_AST_CP-
                        C_ABC_DEC_LAMB_AFS_T_AST_CP]
                else if ABC_LAMB_AFS_T_AST_CP < 0
                        then    ABC_LAMB_AFS_T_AST_CP =
                                MIN[0 , ABC_LAMB_AFS_T_AST_CP+
                                C_ABC_INC_LAMB_AFS_T_AST_CP]
                        endif
                endif
        endif
endif
```

end

28.11.7 Adaptation of FAC_LAM_0_CP in stratified mode

In stratified mode it is necessary to adapt the start value of the lambda deviation concerning to the lambda setpoint during canister purge. The deviation between lambda setpoint and measured lambda value is component specific. Therefor a straigt line is approximated at everv start of RO or ROF. If the lambda setpount is in a defined window, the straight line will be adapted to the actual start value. Out of this straight line, a corrected lambda value LAMB_MMV_COR_CP will be calculated.



28.11.7.1 Initalisation at RO and ROF for FAC_LAM_0_CP correction at stratified mode

At initialization of RO or ROF the filtered lambda and lambda setpoint are stored.

LAMB_MMV_0_CP = LAMB_MMV_CP
LAMB_LPF_0_CP = LAMB_LPF_CP
T_CP_PER_RUN_CP = 0

28.11.7.2 Calculation of the straight line for LAMB_MMV_COR_CP

If LAMB_LPF_CP is in the window of C_LAMB_LPF_AD_MIN_CP and C_LAMB_LPF_AD_MAX_CP the straight line will be learned. The straight line is defined by the sampling point [LAMB_LPF_CP=1; LAMB_MMV_CP=1] and the gradient LAMB_MMV_0_GRD_CP.

```
if ((LAMB_LPF_CP > C_LAMB_LPF_AD_MIN_CP)
and  (LAMB_LPF_CP < C_LAMB_LPF_AD_MAX_CP))
then
    // new gradient calculated by the sampling points 1;1
    and LAMB_MMV_0_CP; LAMB_LPF_0_CP //
    LAMB_MMV_0_GRD_NEW_CP = (LAMB_MMV_CP-1)/(LAMB_LPF_CP-1)

    // Filter for the gradient of the straight line //
    LAMB_MMV_0_GRD_CP$_n$= LAMB_MMV_0_GRD_CP$_{n-1}$+
                    (LAMB_MMV_0_GRD_NEW_CP- LAMB_MMV_0_GRD_CP$_{n-1}$)
                    *C_CRLC_ LAMB_MMV_0_GRD_CP

    T_LAMB_MMV_COR_UPD_CP = 0
endif
```

28.11.7.3 Calculation of corrected FAC_LAM_0_CP in stratified

In stratified mode the value FAC_LAM_0_CP has to be corrected if the lambda setpoint changes. The calculation of the straight line depends on LAMB_LPF_CP and the start value LAMB_LPF_0_CP. If the start value is bigger then C_LAMB_LPF_AD_MIN_CP, the LAMB_MMV_COR_CP value will be on the solid line. If the start value is smaller then C_LAMB_LPF_AD_MIN_CP the value LAMB_MMV_COR_CP will be on the dotted line for LAMB_LPF_CP smaller then > C_LAMB_LPF_CLC_GRD_MIN_CP.

```
if    LC_FAC_LAM_0_COR_ACT_CP = 1
then
    if     (LAMB_LPF_0_CP  > C_LAMB_LPF_AD_MIN_CP or
            LAMB_LPF_CP > C_LAMB_LPF_CLC_GRD_MIN_CP)
    then          // LAMB_MMV_COR_CP is on the solid straight line //
        LAMB_MMV_COR_CP = 1 + [LAMB_MMV_0_GRD_CP *(LAMB_LPF_CP - 1)]
        // linear equation with the sampling points 1;1 and the gradient
        // LAMB_MMV_0_GRD_CP
    elseif LAMB_LPF_CP > LAMB_LPF_0_CP
    then LAMB_MMV_0_TMP_CP =
            1+ [LAMB_MMV_0_GRD_CP * (C_LAMB_LPF_CLC_GRD_MIN_CP - 1)]
        // y-value for the sampling point C_LAMB_LPF_CLC_GRD_MIN_CP,
        // needed for the calculation of the dotted straight line
        LAMB_MMV_COR_CP = LAMB_MMV_0_CP +
                [(LAMB_MMV_0_TMP_CP - LAMB_MMV_0_CP) *
                (LAM_LPF_CP - LAMB_LPF_0_CP)/
                (C_LAMB_LPF_CLC_GRD_MIN_CP - LAMB_LPF_0_CP)]
        // linear equation with the sampling points LAMB_LPF_0_CP;LAMB_MMV_0_CP
        // and C_LAMB_LPF_CLC_GRD_MIN_CP; LAMB_MMV_0_TMP_CP
        else  LAMB_MMV_COR_CP =
```

```
                    1 + [(LAMB_MMV_0_CP-1) * (LAMB_LPF_CL-1) / (LAMB_LPF_0_CP-1)]
                // linear equation with the sampling points 1;1 and
                // LAMB_LPF_0_CP;LAMB_MMV_0_CP
                endif
        endif
        FAC_LAM_0_CP = (LAMB_MMV_COR_CP – LAMB_LPF_CP) / LAMB_LPF_CP
else    FAC_LAM_0_CP = (LAMB_MMV_0_CP – LAMB_LPF_0_CP) / LAMB_LPF_0_CP
endif
```

28.11.7.4 Time management for valid FAC_LAM_0_CP at stratified mode

The run time of the current purge period is measured in T_CP_PER_RUN_CP. This time is compared with a thereshold caculated on the time since the last adaption of the straight line T_LAMB_MMV_COR_UPD_CP and the difference between the lambda setpoint at start of the purge period and the current setpoint. The threshold is weighted with a factor depending on the canister load.

```
T_CP_PER_RUN_CP = T_CP_PER_RUN_CP + 0.2
LAMB_LPF_DIF_IT_CP = ABS(LAMB_LPF_0_CP - LAMB_LPF_CP)
if      (LV_HOM_ACT = 0)
        & (T_CP_PER_RUN_CP >
        (IP_T_CP_PER_END_CP(LAMB_LPF_DIF_IT_CP,T_LAMB_MMV_COR_UPD_CP)
        *IP_T_CP_PER_END_COR_CP(CL_MMV)))
then    LV_CP_PER_END_REQ_S_CP = 1
else    LV_CP_PER_END_REQ_S_CP = 0
endif
```

28.11.8 Open loop calculations

Formula section:

```
if      LV_CL_MDL_ACT       = 1
then    CL_MMV$_n$            = CL_MDL
else    CL_MMV$_n$            = CL_MMV$_{n-1}$
endif
if      LV_STATE_MEM_CP_CLL
or      LV_CL_MDL_ACT       = 1
then    MFF_ADD_CP_KGH      = CL_MMV * FLOW_CPS
else    MFF_ADD_CP_KGH      = 0
endif
CL$_n$                        = CL$_{n-1}$            // no calculation
if      LV_CL_MDL_ACT       = 1
then    MFF_CP              = MFF_ADD_CP_KGH
else    MFF_CP              = 0
endif
(moved to lambda deviation)
LV_CL_GRD_CHK_MAX           = 0
FAC_LAM_DIF_CP              = 0
LV_CP_PER_END_REQ_S_CP = 0
end
```

28.11.9 Calculation of LV_LAMB_AFS_CP

General information:

A commutation to homogeneous lambda one combustion with the flag LV_LAMB_AFS_CP is requested if a short circuit to the ground of the CPS is detected, if the CL-MMV evaluation or the CPS adaptation request this mode. As CL_MMV is only evalutated in closed loop operation, there is the risk of getting stuck in homogeneous operation. Therefore timers limit the request for both criteria (T_AST and AMP) to an applicable time after which a homogeneous request is cancelled.

Application conditions:

Initialisation: all values = 0 at RST or IGKON
Recurrence: **20** ms
Activation: -- (called by superior block)
Deactivation: -- (called by superior block)

Formula section:

```
// Timer calculation
T_LAMB_AFS_T_AST               = max[T_LAMB_AFS_T_AST – TA, 0]
T_LAMB_AFS_AMP_VLD_CP          = max[T_LAMB_AFS_AMP_VLD_CP – TA, 0]
T_LAMB_AFS_T_AST_VLD_CP        = max[T_LAMB_AFS_T_AST_VLD_CP – TA, 0]

if      (LV_LAMB_AFS_T_AST_CP = 1 and VS > C_VS_MIN_LAMB_AFS)
then    T_LAMB_AFS_T_AST = C_T_LAMB_AFS_T_AST
endif

if      LV_CP_CLL = 1
then    T_LAMB_AFS_AMP_VLD_CP      = C_T_LAMB_AFS_AMP_VLD_CP
        T_LAMB_AFS_T_AST_VLD_CP    = C_T_LAMB_AFS_T_AST_VLD_CP
end

// sum of the conditions for LV_LAMB_AFS_CP
LV_LAMB_AFS_CP =
    [(ERR_SYM_EL_CPS = 2h and LC_LAMB_AFS_CP_CPS_SCG = 1)    or
    LV_CPS_AD_HOM_REQ = 1                                   or
    (LV_LAMB_AFS_AMP_CP = 1 and T_LAMB_AFS_AMP_VLD_CP > 0)  or
    (T_LAMB_AFS_T_AST > 0 and T_LAMB_AFS_T_AST_VLD_CP > 0)]

end
```

28.11.10 Engine operating point window

General information:

If the operating point (N,MFF) has a big distance to the operating point stored at entering RO or ROF at stratified engine mode, it is useful to start a new purge phase, due to imprecision in the lambda signal. If the engine operating point will exceed a small window (window_1), the CL-model will be activated, if the CL-model is not valid, a new RO or ROF will be requested. If the operating point will exceed a large window (window_2), a new RO or ROF will be requested.

Formula section:

```
if$_1$ LV_CP_CLL = 1
then$_1$ if$_2$ STATE_OPM_ENG_CP = ENG_MODE_2
    then$_2$ if$_3$ [(N_32_0_CP – N_32) >
        (IP_N_32_OPP_MIN(N_32_0_CP, MFF_SP_0_CP)* C_FAC_OPP_CP_1)
        or (N_32 - N_32_0_CP) >
        (IP_N_32_OPP_MAX(N_32_0_CP, MFF_SP_0_CP)* C_FAC_OPP_CP_1)
        or (MFF_SP_0_CP – MFF_SP_MV) >
        (IP_MFF_SP_OPP_MIN(N_32_0_CP, MFF_SP_0_CP) * C_FAC_OPP_CP_1)
        or (MFF_SP_MV – MFF_SP_0_CP) >
        (IP_MFF_SP_OPP_MAX(N_32_0_CP, MFF_SP_0_CP)* C_FAC_OPP_CP_1)]
        then$_3$ LV_CP_OPP_WIN_1 = 1
            if$_4$ (N_32 < C_N_32_MIN_OPP_WIN_CP
                and MFF_SP_MV < C_MFF_SP_MIN_OPP_WIN_CP)
            or [(N_32_0_CP – N_32) >
            IP_N_32_OPP_MIN(N_32_0_CP, MFF_SP_0_CP)
            or (N_32 - N_32_0_CP) >
            IP_N_32_OPP_MAX(N_32_0_CP, MFF_SP_0_CP)
            or (MFF_SP_0_CP – MFF_SP_MV) >
            IP_MFF_SP_OPP_MIN(N_32_0_CP, MFF_SP_0_CP)
            or (MFF_SP_MV – MFF_SP_0_CP) >
            IP_MFF_SP_OPP_MAX(N_32_0_CP, MFF_SP_0_CP)]
            then$_4$     LV_CP_OPP_WIN_2 = 1
            else$_4$     LV_CP_OPP_WIN_2 = 0
            endif$_4$
        else$_3$ LV_CP_OPP_WIN_1 = 0
            LV_CP_OPP_WIN_2 = 0
        endif$_3$
    else$_2$ if$_3$ STATE_OPM_ENG_CP = ENG_MODE_1
        then$_3$ if$_4$ [(N_32_0_CP – N_32) >
            (IP_N_32_OPP_MIN_1(N_32_0_CP, MFF_SP_0_CP)* C_FAC_OPP_CP_1)
            or (N_32 - N_32_0_CP) >
            (IP_N_32_OPP_MAX_1(N_32_0_CP, MFF_SP_0_CP)* C_FAC_OPP_CP_1)
            or (MFF_SP_0_CP – MFF_SP_MV) >
            (IP_MFF_SP_OPP_MIN_1(N_32_0_CP, MFF_SP_0_CP) *
   C_FAC_OPP_CP_1)
            or (MFF_SP_MV – MFF_SP_0_CP) >
            (IP_MFF_SP_OPP_MAX_1(N_32_0_CP, MFF_SP_0_CP)*
 C_FAC_OPP_CP_1)]
            Then$_4$     LV_CP_OPP_WIN_1 = 1
                If$_5$ (N_32 < C_N_32_MIN_OPP_WIN_CP
                    and MFF_SP_MV < C_MFF_SP_MIN_OPP_WIN_CP)
                or [(N_32_0_CP – N_32) >
                IP_N_32_OPP_MIN_1(N_32_0_CP, MFF_SP_0_CP)
                or (N_32 - N_32_0_CP) >
```

```
                    IP_N_32_OPP_MAX_1(N_32_0_CP, MFF_SP_0_CP)
                    or (MFF_SP_0_CP – MFF_SP_MV) >
                    IP_MFF_SP_OPP_MIN_1(N_32_0_CP, MFF_SP_0_CP)
                    or (MFF_SP_MV – MFF_SP_0_CP) >
                    IP_MFF_SP_OPP_MAX_1(N_32_0_CP, MFF_SP_0_CP)]
                    Then$_5$     LV_CP_OPP_WIN_2 = 1
                    Else$_5$     LV_CP_OPP_WIN_2 = 0
                    Endif$_5$
              Else$_4$ LV_CP_OPP_WIN_1 = 0
                    LV_CP_OPP_WIN_2 = 0
              Endif$_4$
Else$_1$ LV_CP_OPP_WIN_1 = 0
      LV_CP_OPP_WIN_2 = 0
Endif$_1$

end
```

28.11.11 Request for new RAMP OPEN and RAMP OPEN FAST :

General information:

When the "CL value conformity check" detects a non plausible CL value or when the "Time management for valid FAC_LAM_0_CP at stratified mode" requests an new RAMP Open or the enginge operating mode switches or the engine opeating point exceeds a window a transition to MIN_PURGE is requested by setting of LV_CP_NEW_RAMP_OPEN.

In this case no re-entering in the normal purge is possible, instead, WAIT_RAMP_OPEN is activated when the conditions are fullfilled and the transition to RAMP_OPEN or RAMP_OPEN_FASTis done after WAIT_RAMP_OPEN.

Formula section:

```
LV_CP_NEW_RAMP_OPEN = LV_CL_GRD_CHK_MAX

LV_CP_NEW_RAMP_OPEN_FAST =
      [LV_CP_PER_END_REQ_S_CP
      or    (LV_STATE_OPM_CHG_HOM_CP and LC_STATE_OPM_CHG_HOM_CP)
      or    (LV_STATE_OPM_CHG_S_CP and LC_STATE_OPM_CHG_S_CP)
      or    (LV_CL_MDL_ENA = 0 and LV_CP_OPP_WIN_1 = 1)
      or    LV_CP_OPP_WIN_2]
end
```

28.12 EVAC canister load model

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CL_MDL_0	V	0H 1H	CL_MDL_OFF CL_MDL_ON	1	[-]
global state of the CL model (on/off)					
STATE_CL_MDL	V	0H 1H 2H	DISABLED ENABLED ACTIVE	1	[-]
state of the CL model, if switched on					
LV_CL_MDL_ENA	O/V	0...1H	0...1	1	[-]
Activation of the CL model enabled					
LV_CL_MDL_ACT	O/V	0...1H	0...1	1	[-]
CL model active					
CL_MDL	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Value of Canister Load Model					
T_CL_MDL_CLL_ENA	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter in CP closed loop operation to enable canister load model usage (counting backward)					
T_CL_MDL_ACT	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter for canister load usage time (counting forward)					
T_CL_MDL_CAT_PURGE_ENA	V	0...FFFFH	0...6553.5	0.1	[s]
TIme counter to allow activation the model in Cat-Purge phase					
CL_MDL_SLOP_INT	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Integral of CL_MDL_SLOP					
CL_MDL_SLOP_FAC	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
final CL-model correction factor on CL_MMV					
CL_MDL_SLOP	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
slope of CL_MMV					
CL_MMV_MEM_MDL	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
stored value of CL_MMV					
LV_LAM_THD	V	0...1H	0...1	1	[-]
Lambda gradient exceeds threshold					
LV_CL_MDL_END_CAT_PURGE	O/V	0...1H	0...1	1	[-]
CL-model used in Cat-Purge phase					

Input data:

STATE_CP	STATE_CDN_CP	LV_CP_ACT	LV_CP_CDN_MIN_PURGE_FAST
CL_MMV	LV_CP_CLL	T_DLY_CL_MDL	FAC_LAM_GRD_MMV
CL_MMV_MEM_CL_MDL	CL_MMV_MEM_CL_MDL_OLD	LV_CP_OPP_WIN_1	LV_CP_CAT_PURGE

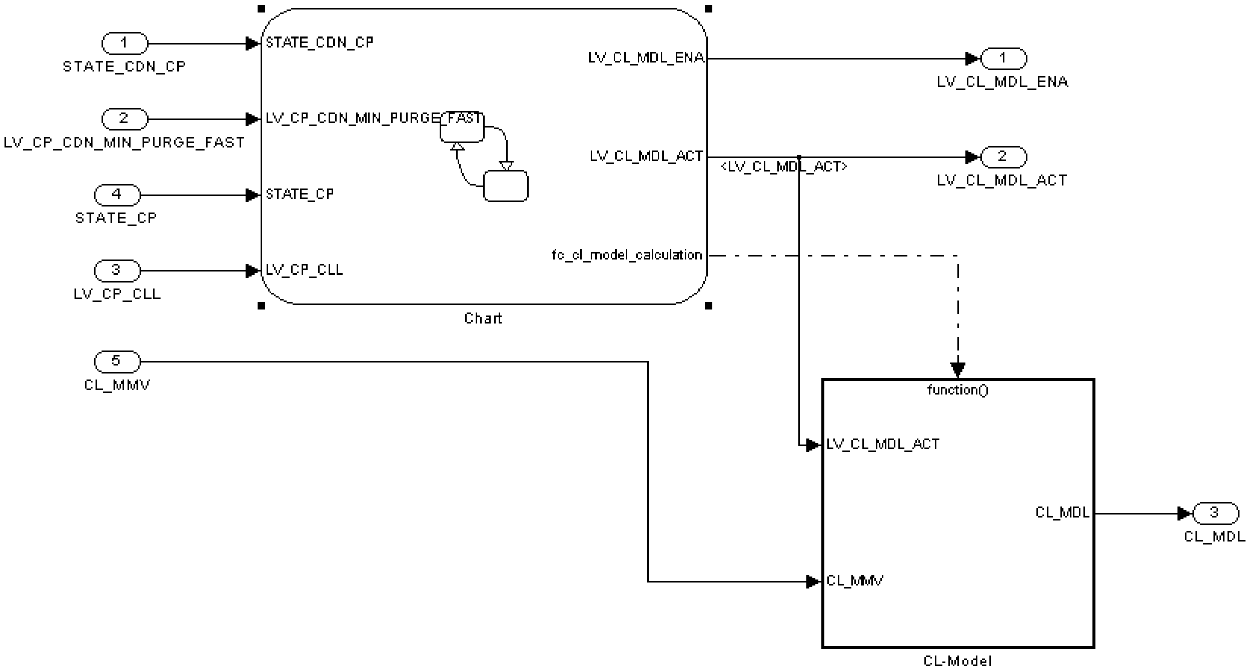
Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_CL_MDL_CLL_ENA_INC	1	0...FFFFH	0...6553.5	0.1	[s]
Increment for enable time counter if not in CP closed loop operation					
C_T_CL_MDL_ACT_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum time of CL model usage					
C_T_CL_MDL_ACT_MAX_CAT_PURGE	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum time of CL model usage at Catalyst purge					
C_T_CL_MDL_ENA	1	0...FFFFH	0...6553.5	0.1	[s]
Time for enabled CL-Model after closed loop operation					
C_T_CL_MDL_ACT_OFS	1	0...FFFFH	0...6553.5	0.1	[s]
offset for re-init of CL mdl ena time counter after regaular leaving of active mode					
C_T_CL_MDL_CAT_PURGE_ENA	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum time in open loop for activaton of the model in Cat-Purge phase					
C_FAC_T_CL_MDL_ACT	1	0...FFFFH	0...3.99993	0.061e-3	[-]
correction factor for re-init of CL mdl ena time counter after regaular leaving of active mode					
C_CRLC_CL_MDL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for CL_MDL filtering					
IP_CL_MDL_SLOP	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_CL_MDL_SLOP_0_IP_CL_SLOP	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
Timeshape over CL_MDL_SLOP					
IP_CL_MDL_VALUE	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_CL_MDL_SLOP_INT_IP_CL_VALUE	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
Increment value on FLOW_SP_CPS					
LC_CL_MDL_ACT	1	0...1H	0...1	1	[-]
Switch for activation of CL model calculation					
LC_CL_MDL_ENA	1	0...1H	0...1	1	[-]
Switch for enabling CL model usage					
C_FAC_LAM_THD	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper threshold for lamgda control output gradient					
LC_FAC_LAM_GRD	1	0...1H	0...1	1	[-]
Switch for enabling lambda control output gradient check					
IP_T_CL_MDL_CLL_ENA	6	0...FFFFH	0...6553.5	0.1	[s]
LDP_CL_IP_T_CL_MDL_CLL_ENA	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
Minimum time in CP closed loop operation to enable CL model usage					

General information:

The evaporative emissions of the fuel tank system are stored in an activated charcoal filter ACF to prevent an escape of the gas to the environment. The loading capacity of the filter is limited, so the filter must be purged temporarily. Therefore the canister purge solenoid CPS. that is positioned between the charcoal filter and the intake manifold is opened by the engine management system. To prevent driveability and emission problems, the opening of the CPS and the purge flow MFF_CP must be controlled.

With the canister load modelling function (CL-Model), the continuity of the CL factor over time during phases of disturbed calculation is guarantied. Therefore the canister load CL will be substituted with the output of a SIMULINK model, which is a calculation result of a normalized exponential function.



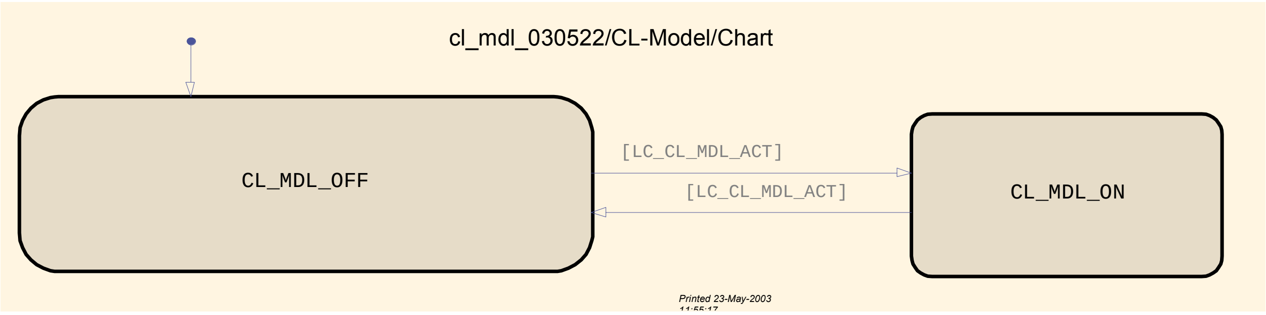
Application conditions:

Initialisation: at RST, IGKON or deactivation
STATE_CL_MDL_0 = CL_MDL_OFF
STATE_CL_MDL = DISABLED
all other values = 0
Recurrence: **100** ms
Activation: LV_CP_ACT = 1
Deactivation: LV_CP_ACT = 0

Formula section:

call "CL model state machine 0 (superstate)"
end

28.12.1 CL model state machine 0 (superstate)



28.12.1.1 CL_MDL_OFF

```
Transitions:
if      LC_CL_MDL_ACT     = 1
then    STATE_CL_MDL_0    = CL_MDL_ON
        STATE_CL_MDL            = DISABLED
endif
end

Actions:
LV_CL_MDL_ACT      = 0
LV_CL_MDL_ENA      = 0
T_CL_MDL_CLL_ENA = IP_T_CL_MDL_CLL_ENA(CL_MMV)          // init
T_CL_MDL_ACT       = 0
end
```

28.12.1.2 CL_MDL_ON

```
Transitions:
if      LC_CL_MDL_ACT     = 0
then    STATE_CL_MDL_0    = CL_MDL_OFF
endif
end

Actions:
if    ~~LV_CP_CLL                   = 1~~
      STATE_CP = Max_Purge
then  If    LV_CP_CAT_PURGE = 0
      then  T_CL_MDL_CLL_ENA        = max [0, T_CL_MDL_CLL_ENA - TA]
            LV_CL_MDL_END_CAT_PURGE = 0
      endif
      T_CL_MDL_CAT_PURGE_ENA = C_T_CL_MDL_CAT_PURGE_ENA
else  T_CL_MDL_CLL_ENA        = min [IP_T_CL_MDL_CLL_ENA(CL_MMV),
            T_CL_MDL_CLL_ENA + TA * C_T_CL_MDL_CLL_ENA_INC]
      T_CL_MDL_CAT_PURGE_ENA = max[0,T_CL_MDL_CAT_PURGE_ENA – TA]
endif
if    (T_DLY_CL_MDL = 0
      and FAC_LAM_GRD_MMV > C_FAC_LAM_THD)
      and LC_FAC_LAM_GRD

      then  LV_LAM_THD = 1
      else  LV_LAM_THD = 0
endif

call  "CL model state machine"
call  "canister load modelling"
end
```

28.12.2 CL model state machine

28.12.2.1 DISABLED

Transitions:

```
if          {(T_CL_MDL_CLL_ENA       = 0
                 and LV_CP_CAT_PURGE = 0
            or ( LV_CP_CAT_PURGE    = 1
                 and T_CL_MDL_CAT_PURGE_ENA > 0
                 and LV_T_CL_MDL_END_CAT_PURGE = 0)}
            and         LC_CL_MDL_ENA        = 1
then        STATE_CL_MDL        = ENABLED          // switch to enable CL model usage
endif
end
```

Actions:

```
LV_CL_MDL_ACT                  = 0
LV_CL_MDL_ENA                  = 0
T_CL_MDL_ACT                   = 0
end
```

28.12.2.2 ENABLED

Transitions:

```
if          T~~_CL_MDL_CLL_ENA > C_T_CL_MDL_ENA~~
            (T_CL_MDL_CLL_ENA        > 0
                  or LV_CP_CAT_PURGE = 1)
            and (LV_CP_CAT_PURGE = 0
                  or T_CL_MDL_CAT_PURGE_ENA = 0)
then        STATE_CL_MDL        = DISABLED

elseif            [STATE_CP              = MAX_PURGE
            and   {(STATE_CDN_CP = CDN_MIN_PURGE
                  and LV_CP_CDN_MIN_PURGE_FAST = 0)
            or    LV_LAM_THD = 1
            or    LV_CP_OPP_WIN_1 = 1}]
        or  [LV_CP_CAT_PURGE = 1
            and {STATE_CDN_CP = CDN_RAMP_OPEN
            or STATE_CDN_CP = CDN_RAMP_FAST}]
then        STATE_CL_MDL        = ACTIVE
endif
end
```

Actions:

```
LV_CL_MDL_ACT              = 0
LV_CL_MDL_ENA              = 1
end
```

28.12.2.3 ACTIVE

Transitions:

```
// canister purge closed loop operation abandoned
if          T_CL_MDL_CLL_ENA > C_T_CL_MDL_ENA
            and LV_CP_CAT_PURGE = 0
then        STATE_CL_MDL          = DISABLED

// max. time of CL model usage elapsed
elseif          (T_CL_MDL_ACT ≥ C_T_CL_MDL_ACT_MAX and LV_CP_CAT_PURGE
= 0)
        or  (T_CL_MDL_ACT ≥ C_T_CL_MDL_ACT_MAX_CAT_PURGE
            and LV_CP_CAT_PURGE = 1)
then        T_CL_MDL_CLL_ENA = IP_T_CL_MDL_CLL_ENA(CL_MMV)     // init
            if LV_CP_CAT_PURGE = 1
            then LV_CL_MDL_END_CAT_PURGE = 1
            endif
            STATE_CL_MDL          = DISABLED

// no more usage of CL model necessary; switch to "normal" MAX_PURGE
elseif          [(STATE_CDN_CP ≠ CDN_MIN_PURGE and LV_LAM_THD = 0
            and LV_CP_OPP_WIN_1 = 0)
            and {LV_CP_CAT_PURGE = 0
                or (STATE_CDN_CP ≠ CDN_RAMP_OPEN
                and STATE_CDN_CP ≠ CDN_RAMP_FAST)}]
then        T_CL_MDL_CLL_ENA = min [IP_T_CL_MDL_CLL_ENA(CL_MMV),
                T_CL_MDL_ACT* C_FAC_T_CL_MDL_ACT +
                C_T_CL_MDL_ACT_OFS]
            if LV_CP_CAT_PURGE = 1
            then LV_CL_MDL_END_CAT_PURGE = 1
            endif
            STATE_CL_MDL          = DISABLED
endif
end
```

Actions:

```
LV_CL_MDL_ACT           = 1
LV_CL_MDL_ENA           = 1
T_CL_MDL_ACT            = T_CL_MDL_ACT + TA
end
```

28.12.3 Canister Load modelling

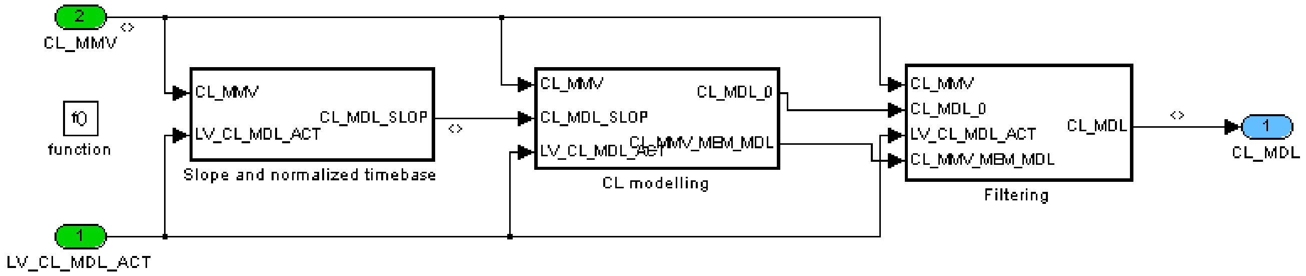


figure: structure of the canister load model

Formula section:

call “Slope and normalized timebase”
call “Modelling and output of CL_MDL”
call “Evaluation CL versus CL_MDL”
end

Calculation information:

The overview shows that the complete function is divided into three submodules, the used inputs are the recurrence time described in chapter Application conditions, the logical value LV_CL_MDL_VALUE_ACT indicating instationary driving conditions it is the result of the Data Type Conversion block with STATE_CDN_CL_MDL as input and the value of Canister Load.

The outgoing CL_MDL value is chosen via a switch block between either the Canister Load during stationary driving conditions at the one hand and the modelled Canister Load Load during instationary driving conditions at the other hand indicated by the variable LV_CL_MDL_VALUE_ACT.

28.12.3.1 Slope and normalized timebase

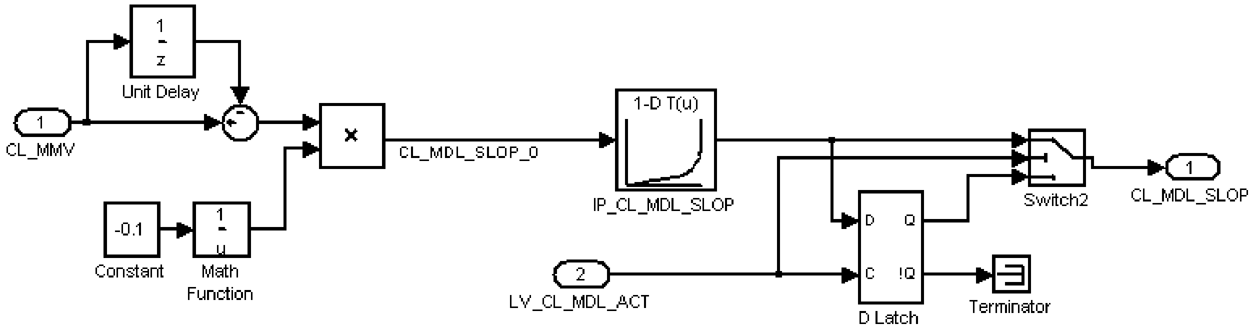


figure: slope and normalized timebase

This submodule calculates the (inverted) slope of CL_MMV by building the difference CL_MMV_n minus CL_MMV_{n-1}. The slope is used with the Map IP_CL_MDL_SLOP to read out the normalized time constant CL_MDL_SLOP which indicates the time position on a exponential function using the slope as an input. The outgoing CL_MDL_SLOP value is frozen when LV_CL_MDL_ACT=1.

Formula section:

```
if    LV_CL_MDL_ACT       = 0
then  CL_MDL_SLOP_0       = -(CL_MMV_MEM_CL_MDL-CL_MMV_MEM_CL_MDL_OLD)
      CL_MDL_SLOP         = IP_CL_MDL_SLOP (CL_MDL_SLOP_0)
else  $CL\_MDL\_SLOP_n$   = $CL\_MDL\_SLOP_{n-1}$
endif
end
```

28.12.3.2 Modelling and output of CL_MDL

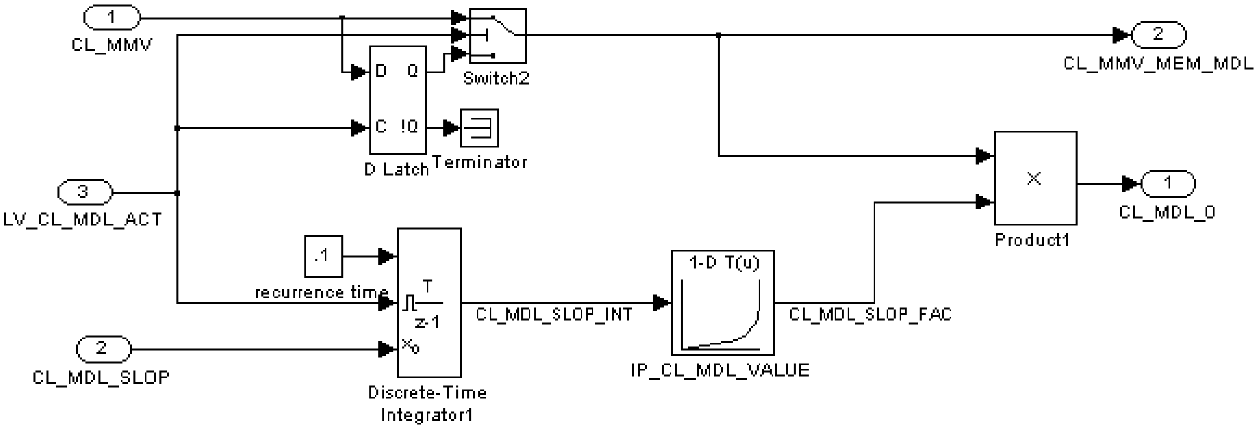


figure: modelling and output of CL_MDL

The output of IP_CL_MDL_VALUE is the normalized value of CL_MDL over $CL\_MDL\_SLOP_{integral}$. The input of IP_CL_MDL_VALUE is calculated with the Discrete-Time Integrator block. The starting point of integration is CL_MDL_SLOP.

Formula section:

```
if    LV_CL_MDL_ACT       = 0
then  CL_MDL_0            = CL_MMV_MEM_CL_MDL
      CL_MMV_MEM_MDL      = CL_MMV_MEM_CL_MDL
      CL_MDL_SLOP_INT     = 0
else  CL_MDL_SLOP_INT     = $CL\_MDL\_SLOP\_INT_{n-1}$ + CL_MDL_SLOP
      CL_MDL_SLOP_FAC     = IP_CL_MDL_VALUE (CL_MDL_SLOP_INT)
      CL_MDL_0            = CL_MMV_MEM_MDL * CL_MDL_SLOP_FAC
end
```

28.12.3.3 Evaluation CL_MMV versus CL_MDL

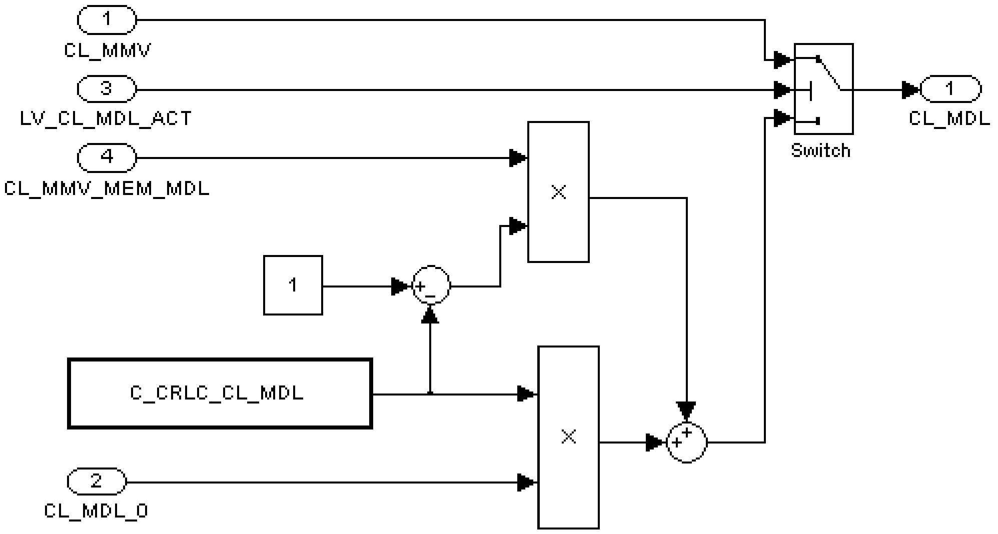


figure: evaluation CL versus CL_MDL

This submodul evaluates the CL_MMV_MEM_MDL value (frozen if LV_CL_MDL_ACT) versus the CL_MDL_0 value by using the filter constant C_CRLC_CL_MDL.

Formula section:

```
if    LV_CL_MDL_ACT      = 0
then  CL_MDL             = CL_MMV
else  CL_MDL             = CL_MDL_0 * C_CRLC_CL_MDL
                         + CL_MMV_MEM_MDL * (1-C_CRLC_CL_MDL)
end
```

28.13 Time control between EVAP and Lambda adaption

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_T_SDL_CP	O/V	0H 1H 2H 3H 4H 5H 6H 7H 8H	PASSIVE CP_FIXED LAM_FIXED WAIT_LAM LAM_DYN CP_DYN WAIT_CP OFF_CP EXT_REQ	1	[-]
State of Time Control Function					
LV_LAM_AD_ENA	O/V	0...1H	0...1	1	[-]
Lambda adaptation enabled					
LV_CP_ENA	O/V	0...1H	0...1	1	[-]
Canister purge enabled					
T_CP	V	0...FFFFH	0...6553.5	0.1	[s]
Time of actual CP time range					
T_AD	V	0...FFFFH	0...6553.5	0.1	[s]
Time of actual lambda adaption time range					
T_MAX_AD	V	0...FFFFH	0...6553.5	0.1	[s]
Maximum adaption time; used to disable CP when canister is detected empty					
LV_STATE_SDL_LOCK_CP	V	0...1H	0...1	1	[-]
Fixed lambda adaptation is left due to finished lambda adaptation					
LV_INH_SDL_CP	-	0...1H	0...1	1	[-]
flag: normal scheduling disabled					
LV_T_SDL_NEW_RAMP_OPEN_CP	V	0...1H	0...1	1	[-]
Flag: new Ramp Open after a lambda adaptation					
T_TCO_ST_STAT_SDL_CP	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter for canister purge after start at hot conditions					
LV_TCO_ST_STAT_SDL_CP	V	0...1H	0...1	1	[-]
Time slice for canister purge after start at hot conditions					

Input data:

LV_LAM_AD_END	LV_CYL_BAL_LAM_AD_ENA_DC	STATE_CP	LV_CP_CLOSE_ACT
LV_LAM_AD_CDN	LV_IGK	CL_MMV	LV_INH_CP_SDL
T_PRI_TOT_LAM_AD	LV_STATE_T_SDL_CP_LAM_AD_EXT	LV_CP_ACT	NC_CBK_EX_NR
TCO	LV_ES		
LV_INH_LAM_AD_SDL	LV_T_SDL_EXT_REQ	T_AST_CP	

Input data table order: External inputs – Inputs from other EVAC modules

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_T_MAX_CP	6	1...FFFFH	0.1...6553.5	0.1	[s]
LDP_CL_MMV_IP_T_MAX_CP	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
Duration of maximum purge versus Canister Load					
C_T_2_CP	1	1...FFFFH	0.1...6553.5	0.1	[s]
Duration of purge phase					
C_T_DI_CP	1	1...FFFFH	0.1...6553.5	0.1	[s]
Disable time of purge phase when canister detected empty					
C_T_MFF_AD_MAX	1	1...FFFFH	0.1...6553.5	0.1	[s]
Time out for adaption phase					
C_T_SDL_EXT_REQ	1	1...FFFFH	0.1...6553.5	0.1	[s]
Time for external request					
C_CL_MMV_T_BOL	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Time management canister load bottom limit					
C_CL_MMV_T_TOL	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Time management canister load top limit					
C_CL_MMV_T_WAIT_LAM	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
CL threshold to switch off CP during WAIT_LAM					
C_TCO_DYN_T_CTL	1	0...FEH	-48...142.5	0.75	[°C]
Minimum temperature for activation of dynamic time control between EVAP and lambda adaptation					
ID_T_SET_T_AST_LAM_CP	NC_NR_CP_LAM_AD	1H 2H 3H	CP_FIXED LAM_FIXED DYN_SDL	1	[-]
LDP_T_AST_SET_ID_T_AST	NC_NR_CP_LAM_AD	0...FFFFH	0...6553.5	0.1	[s]
Indexiation for fixed time slices					
C_TCO_COLD_DYN_T_CTL_CP	1	0...FEH	-48...142.5	0.75	[°C]
Maximum cold temperatur to activate the dymanic time slices at start of DC					
C_T_TCO_ST_STAT_SDL_CP	1	0...FFFFH	0...6553.5	0.1	[s]
Time for canister purge after start at hot conditions					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_MFF_AD_ENA_PAS	1	0...1H	0...1	1	[-]
switch to enable lambda adaptation while cp passive					
LC_CP_ENA_WAIT_LAM	1	0...1H	0...1	1	[-]
switch to permantently enable canister purge in WAIT_LAM					
LC_CL_CP_DYN_END_ENA	1	0...1H	0...1	1	[-]
switch to enable end of CP phase when reaching low canister load					
LC_SDL_LAM_AD_CDN_OFF	1	0...1H	0...1	1	[-]
switch to ignore LV_LAM_AD_CDN (adapt. ongoing) for end of lambda adapt. time slice calculation					
C_STATE_MAN_SDL_CP	1	0H 1H 2H 3H	AUTOMATIC SDL_NONE SDL_CP LAM_AD	1	[-]
selection of manual scheduling mode					
NC_NR_CP_LAM_AD	1	0...FFH	0...255	1	[-]
Number of fixed CP / Lambda adaptation slices after engine start					
LC_CYL_BAL_LAM_AD_INH_CP	1	0...1H	0...1	1	[-]
Inhibition of: transition to LAM_DYN controlled by CYBL					

General information:

Lambda adapation and canister purge both use a lambda deviation for their purposes. Indeed, as the canister purge itself causes a certain lambda deviation, both functions can not work simultaneous. As a consequence, a time management has to be installed partitioning the engine opertating time and enabling either canister purge or lambda adaptation while disabling the other function.

Signal flow diagram:

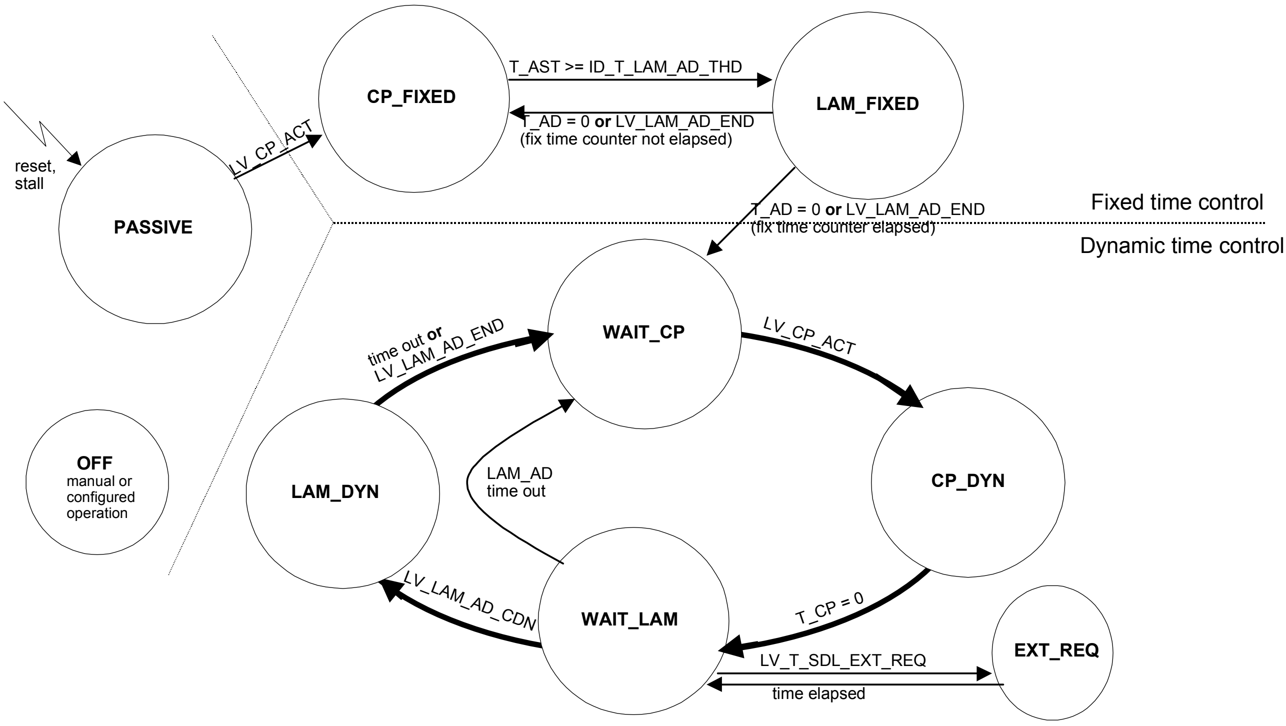


figure 1: time scheduler state diagram

The state diagram above only contains the recent transitions; all possible transitions are listed in the formula section below.

The following table indicates whether canister purge or lambda adapatation is enabled/ disabled depending on the state STATE_T_SDL_CP of the time scheduler.

STATE_T_SDL_CP	CP	LAM_AD
PASSIVE	✗	C
CP_FIXED	✓	✗
CP_DYN	✓	✗
WAIT_LAM	✓	✗
LAM_FIXED	✗	✓
LAM_DYN	✗	✓
WAIT_CP	✗	✓
EXT_REQ	✗	✗

configurable C

Lambda adaptation is enabled in the corresponding states after the CPS is closed

CP is enabled in WAIT_LAM for a calibratable time

figure 2: enabling of canister purge or lambda adaptation depending on state

In general the time control is splited into two main parts:

- At leaving start the time split between lambda adaption and canister purge is defined by a certain number of fixed time frames. The number of these fixed time frames is given by NC_NR_CP_LAM_AD -1 (last defines the start of dynamic time slices).
- After performing the fixed time frames, the time periods of canister purge and lambda adaption are depending on canister load and the necessity of lambda adaption, indicated by their corresponding functions, so dynamic time frames will be applied.
- For Hybrid and Stop/Start configurations the timers for the dynamic time slices are stopped during engine off time.

After the start phase has ended and canister purge is possible (LV_CP_ACT = 1), the manager starts with a time slice depending on TCO. For very cold condition, the dynamic time slices will be performed immediatily. For a mid range temperature, the fixed slices defined in ID_T_SET_T_AST_LAM_CP are done and for a high temperature at first one fixed canister purge time frame will perfaomed. After a defined time the manager will switch to the dynamic time slices.

During CP_FIXED period the canister purge is possible if the conditions for the function are fulfilled. If they're not fulfilled, neither canister purge nor lambda adaptation will be active.

During LAM_FIXED period the canister purge is disabled and the lambda adaptation is enabled. During this period and after the canister purge valve is closed, the lambda adaptation is possible if the conditions for the function are fulfilled. If they're not fulfilled, neither canister purge nor lambda adaptation will be active.

If the lambda adaptation sends its ready bit LV_LAM_AD_END, the lambda adaptation time will immediately end and the state switches to CP_FIXED.

This sequence (canister purge / lambda adapation) is repeated until ID_T_SET_T_AST_LAM_CP defines dynamic slices; for each period, the start time is defined by the corresponding value of the indexiation table ID_T_SET_T_AST_LAM_CP.

The dynamic time frames start with WAIT_CP after the fixed time frames are elapsed, see description of the different states below.

Description of the different time control states

PASSIVE

After each reset or engine stalling the time control starts with state PASSIVE and switches to CP_FIXED or WAIT_CP depending on TCO and when canister purge starts to be active, i.e. LV_CP_ACT = 1 is fulfilled. Lambda adaptation can be activated during this state (configurable)

CP_FIXED

Only canister purge is enabled. This state is active as long as no lambda adaptation is triggered by the corresponding time threshold. If a lambda adaptation is urgently requested, LAM_FIXED is selected.

LAM_FIXED

Only lambda adaption is enabled. This state is left, if the adaptation is finished or the next fixed or dynamic slice starts. If immediate lambda adpatation is requested via the flag LV_STATE_T_SDL_CP_LAM_AD_EXT, this state is not left until these flags have been reset. Afterwards, the fixed time slices are resumed.

CP_DYN

Canister purge is enabled. This state is left, if the timecounter T_CP elapses or an urgent lambda adaptation is requested.

LAM_DYN

Lambda adaption is enabled. To leave this state, a time counter initialised when detecting an empty canister has to have elapsed. The state is left, if the standard adaptation time counter has elapsed and no adaptation is currently ongoing or if the end of an adaptation cycle is signaled by lambda adaptation itself.

WAIT_LAM

Within WAIT_LAM, canister purge is enabled; this enabling may expire after a calibratable time. As soon as the conditions for a lambda adaptation are given, this state is left towards LAM_DYN and lambda adaptation starts. If the time counter for lambda adaptation C_T_MFF_AD_MAX is elapsed, WAIT_CP is selected and canister purge is enabled.

WAIT_CP

Within WAIT_CP, lambda adaptation is enabled. As soon as canister purge function is active (LV_CP_ACT = 1), this state is left towards CP_DYN and canister purge is enabled.

EXT_REQ

If functions other than canister purge and lambda adaptation request a certain time slice with both, lambda adaptation and canister purge deactivated, the time scheduler may switch to the state EXT_REQ for a calibratable time.

In all states with enabled lambda adaptation, the enabling starts when the canister purge valve is closed (LV_CP_CLOSE_ACT = 1).

At the “regular” entry of MAX_PURGE (i.e. not a re-entry after a short deactivation but an entry after finishing canister load learning phases RAMP_OPEN or RAMP_OPEN_FAST) the time slices for CP and lambda adaptation are re-evaluated; remaining time for canister purge is the minimum of IP_T_MAX_CP (depending on canister load) and the lambda adaption priorisation time T_PRI_TOT_LAM_AD. A small time T_PRI_TOT_LAM_AD from lambda adaptation is related to a high priority for lambda adaptation as the canister purge time will be decreased.

Three cases are handled:

High canister load

If the canister is detected full at the end of the learning phase the time control gives priority to the evaporative emission control by increasing the remaining purge time in order to decrease the canister load. The purge time IP_T_MAX_CP is used to purge the canister.

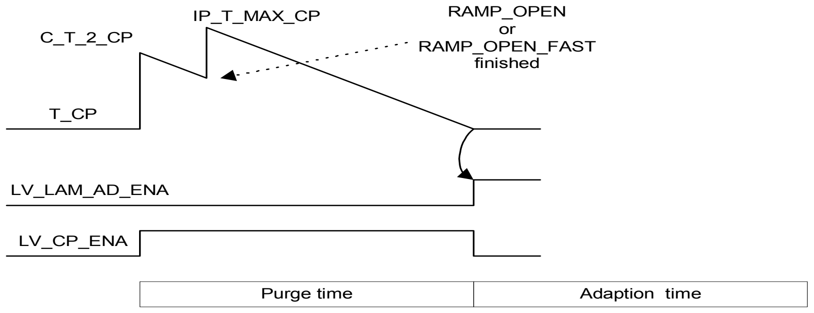


figure 3: canister purge time at high canister load

Medium canister load

If the canister is detected with a medium load, the purge time is calculated as the minimum value between the needs of the lambda adaption and canister purge using the lambda adaption priorisation time T_PRI_TOT_LAM_AD. A small time T_PRI_TOT_LAM_AD from lambda adaptation is related to a high priority for lambda adaptation as the canister purge time will be decreased.

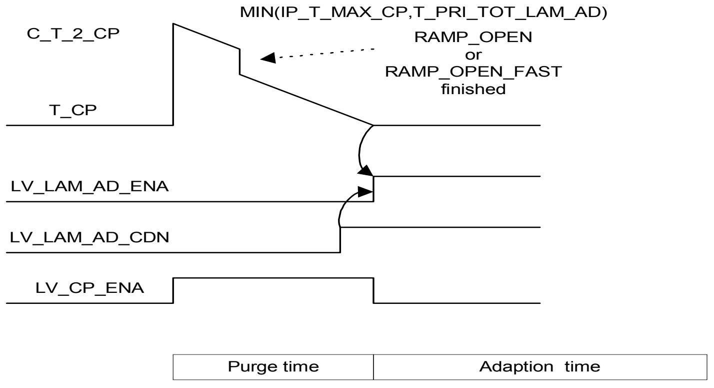


figure 4: canister purge time at medium canister load

Empty canister

If the canister is detected empty at the end of the learning phase, the canister purge is disabled for a time C_T_DI_CP and the lambda adaption is enabled.

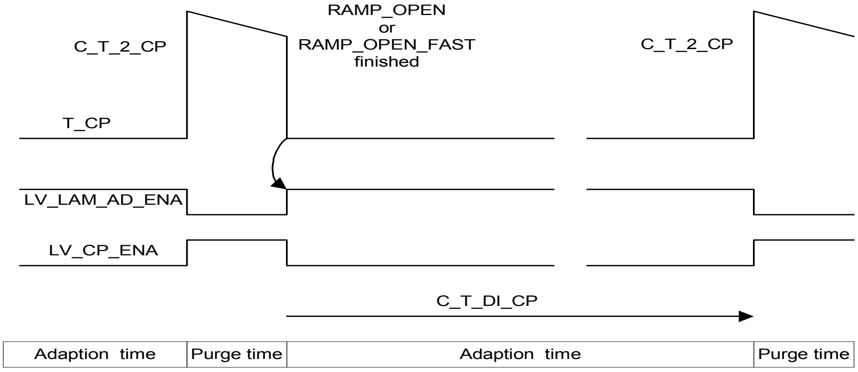


figure 5: canister purge time at low canister load

Application conditions:

Initialisation: at RST or IGKON:
STATE_T_SDL_CP = PASSIVE
all other values = 0
Recurrence: **100** ms
Activation: LV_IGK = 1
Deactivation: --

Formula section:

```
LV_INH_SDL_CP =
     [ LV_INH_LAM_AD_SDL        = 1
or     LV_INH_CP_SDL            = 1
or     C_STATE_MAN_SDL_CP       ≠ AUTOMATIC ]

call   "Time control states"    (corresponding transition part)
call   "Time control states"    (corresponding action part)

if     LV_INH_SDL_CP            = 0
then   call "Calculation of the time slices for canister purge and lambda adaption"
else   STATE_T_SDL_CP           = OFF_CP          // transition possible from all states
endif
end
```

28.13.1 Time control states

28.13.1.1 PASSIVE

Transitions:

```
if          LV_CP_ACT              = 1
then  if$_1$    TCO <  C_TCO_COLD_DYN_T_CTL_CP
      then$_1$  STATE_T_SDL_CP     = WAIT_CP
      else$_1$  if$_2$ TCO < C_TCO_DYN_T_CTL
            then$_2$ if$_3$ ID_T_SET_T_AST_LAM_CP (T_AST_CP) = DYN_SDL
                  then$_3$ STATE_T_SDL_CP     = WAIT_CP
                  else$_3$   STATE_T_SDL_CP    = CP_FIXED
                  endif$_3$
            else$_2$ T_TCO_ST_STAT_SDL_CP = C_ T_TCO_ST_STAT_SDL_CP
                  LV_TCO_ST_STAT_SDL_CP = 1
                  STATE_T_SDL_CP     = CP_FIXED
            endif$_2$
      endif$_1$
endif
end
```

Actions:

```
LV_CP_ENA                       = 0              // inits
LV_LAM_AD_ENA                   = [ LC_MFF_AD_ENA_PAS and LV_CP_CLOSE_ACT ]
T_CP                            = 0
T_AD                            = 0
T_MAX_AD                        = 0
end
```

28.13.1.2 CP_FIXED

Transitions:

```
if          LV_CP_ACT              = 0
then        STATE_T_SDL_CP         = PASSIVE

elseif            LV_STATE_T_SDL_CP_LAM_AD_EXT = 1 // immediate lam. adapt. required
            or    (ID_T_SET_T_AST_LAM_CP (T_AST_CP) = LAM_FIXED
                  and LV_STATE_SDL_LOCK_CP = 0
                  and LV_TCO_ST_STAT_SDL_CP = 0)
then        STATE_T_SDL_CP         = LAM_FIXED
elseif            LV_TCO_ST_STAT_SDL_CP = 1
            and T_TCO_ST_STAT_SDL_CP = 0
then        STATE_T_SDL_CP         = WAIT_LAM
elseif            ID_T_SET_T_AST_LAM_CP (T_AST_CP) = DYN_SDL
then        STATE_T_SDL_CP         = WAIT_CP
endif
end
```

Actions:

```
LV_CP_ENA                          = 1
LV_LAM_AD_ENA                      = 0
T_TCO_ST_STAT_SDL_CP = max(0, T_TCO_ST_STAT_SDL_CP - TA)
if     ID_T_SET_T_AST_LAM_CP (T_AST_CP) = CP_FIXED
then   LV_STATE_SDL_LOCK_CP = 0
endif
end
```

28.13.1.3 LAM_FIXED

Transitions:

```
if      LV_STATE_T_SDL_CP_LAM_AD_EXT = 0
then    if          ID_T_SET_T_AST_LAM_CP (T_AST_CP) = DYN_SDL
        then        STATE_T_SDL_CP     = WAIT_CP
        elseif              ID_T_SET_T_AST_LAM_CP (T_AST_CP) = CP_FIXED
        then        STATE_T_SDL_CP     = CP_FIXED
        elseif              LV_LAM_AD_END = 1
        then        STATE_T_SDL_CP     = CP_FIXED
                    LV_STATE_SDL_LOCK_CP = 1
        endif
endif
end
```

Actions:

```
LV_CP_ENA                          = 0
LV_LAM_AD_ENA                      = LV_CP_CLOSE_ACT
endif
end
```

28.13.1.4 WAIT_CP

Transitions:

```
if          LV_CP_ACT              = 1
then        T_CP                          = C_T_2_CP
            STATE_T_SDL_CP         = CP_DYN
endif
end
```

Actions:

```
if LV_T_SDL_NEW_RAMP_OPEN_CP             = 1
then  LV_CP_ENA                          = 0
endif
LV_LAM_AD_ENA                      = LV_CP_CLOSE_ACT
LV_T_SDL_NEW_RAMP_OPEN_CP     = 0
end
```

28.13.1.5 CP_DYN

Transitions:

```
if       LV_STATE_T_SDL_CP_LAM_AD_EXT = 1 // immediate lam. adapt. required
or       LV_CP_ACT            = 0                  // CP disabled
or       T_CP                          = 0                  // regular end of CP time
or      [ CL_MMV <= C_CL_MMV_T_BOL   // canister empty
and      STATE_CP = MAX_PURGE
and      LC_CL_CP_DYN_END_ENA = 1 ]
then     T_AD                          = C_T_MFF_AD_MAX
         T_MAX_AD             = 0
         STATE_T_SDL_CP       = WAIT_LAM
endif
end
```

Actions:

```
LV_CP_ENA                     = 1
LV_LAM_AD_ENA                 = 0
if LV_ES = 0
then  T_CP                            = max [0, T_CP – TA]
endif
```

28.13.1.6 WAIT_LAM

Transitions:

```
if       LV_STATE_T_SDL_CP_LAM_AD_EXT = 1 // immediate lam. adapt. required
then     STATE_T_SDL_CP       = LAM_DYN

elseif          LV_T_SDL_EXT_REQ = 1
then     T_CP                          = C_T_SDL_EXT_REQ
         STATE_T_SDL_CP       = EXT_REQ

elseif          (LV_LAM_AD_CDN       = 1
      and  (LV_CYL_BAL_LAM_AD_ENA_DC = 1
           or LC_CYL_BAL_LAM_AD_INH_CP = 1))
then     STATE_T_SDL_CP       = LAM_DYN

elseif      [ T_AD                    = 0
      and  T_MAX_AD                = 0 ]
then     STATE_T_SDL_CP       = WAIT_CP
endif
end
```

Actions:

```
LV_CP_ENA                     = [CL_MMV ≥ C_CL_MMV_T_WAIT_LAM]
LV_LAM_AD_ENA                 = 0
if LV_ES = 0
then  T_AD                            = max [0, T_AD – TA]
      T_MAX_AD                        = max [0, T_MAX_AD – TA]
endif
```

```
end
```

28.13.1.7 LAM_DYN

Transitions:

```
if          T_MAX_AD              = 0          // CP disabling time after low cl detection
  and       LV_STATE_T_SDL_CP_LAM_AD_EXT = 0
  and  [ LV_LAM_AD_END  = 1          // adapt. finished
       or  (T_AD                  = 0          // time out
       and {LV_LAM_AD_CDN         = 0   // and adapt. not ongoing
       or   LC_SDL_LAM_AD_CDN_OFF = 1}) ]
then        STATE_T_SDL_CP        = WAIT_CP
endif
end
```

Actions:

```
LV_CP_ENA                         = 0
LV_LAM_AD_ENA                     = LV_CP_CLOSE_ACT
if LV_ES = 0
then  T_AD                        = max [0, T_AD – TA]
      T_MAX_AD                    = max [0, T_MAX_AD – TA]
Endif
LV_T_SDL_NEW_RAMP_OPEN_CP = 1
end
```

28.13.1.8 EXT_REQ

Transitions:

```
if          LV_STATE_T_SDL_CP_LAM_AD_EXT = 1 // immediate lam. adapt. required
or          T_CP                        = 0              // regular end of CP time
or          LV_T_SDL_EXT_REQ = 0                 // no more external request
then        STATE_T_SDL_CP   = WAIT_LAM
endif
end
```

Actions:

```
LV_CP_ENA                         = 0
LV_LAM_AD_ENA                     = 0
if LV_ES = 0
then  T_CP                        = max [0, T_CP – TA]
endif
// T_CP is used as counter to economize on a special counter T_EXT_REQ
end
```

28.13.1.9 OFF_CP

Transitions:

```
if          LV_INH_SDL_CP         = 0
```

```
then     STATE_T_SDL_CP     = PASSIVE
endif
end

Actions:
T_CP                        = 0           // inits
T_AD                        = 0
T_MAX_AD                    = 0

case selection on C_STATE_MAN_SLD_CP
    SDL_NONE:
    LV_CP_ENA               = 0
    LV_LAM_AD_ENA           = 0

    SDL_CP:
    LV_CP_ENA               = 1
    LV_LAM_AD_ENA           = 0

    LAM_AD:
    LV_CP_ENA               = 0
    LV_LAM_AD_ENA           = LV_CP_CLOSE_ACT

    AUTOMATIC:
    if   LV_INH_LAM_AD_SDL = 1              // both inhibited
    and  LV_INH_CP_SDL      = 1
    then LV_CP_ENA          = 0
         LV_LAM_AD_ENA      = 0

    elseif    LV_INH_CP_SDL      = 0
    then LV_CP_ENA          = 1
         LV_LAM_AD_ENA      = 0

    elseif LV_INH_LAM_AD_SDL = 0
    then LV_CP_ENA          = 0
         LV_LAM_AD_ENA      = LV_CP_CLOSE_ACT
    endif
end case selection
end
```

28.13.2 Calculation of the time slices for canister purge and lambda adaption

```
// also to be called with empty canister and C_STATE_MOD_SEL_CP = SEL_MAX_PURGE
if   ( STATE_CP n-1          = RAMP_OPEN
or     STATE_CP n-1          = RAMP_OPEN_FAST)
and    STATE_CP n            = MAX_PURGE
and    STATE_T_SDL_CP        > LAM_FIXED
then call  "time slices calculation"
endif
```

```
end
```

28.13.2.1 time slices calculation

```
// high canister load
if      CL_MMV              ≥ C_CL_MMV_T_TOL
then    T_CP                     = IP_T_MAX_CP (CL_MMV)

// medium canister load
elseif        CL_MMV             ≥ C_CL_MMV_T_BOL
then    T_CP                = min [IP_T_MAX_CP (CL_MMV),
                                   T_PRI_TOT_LAM_AD]

// low canister load
else    STATE_T_SDL_CP      = WAIT_LAM
        T_AD                     = C_T_MFF_AD_MAX
        T_MAX_AD            = C_T_DI_CP
        T_CP                     = 0
endif
end
```

28.14 Canister Purge Valve Control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CPPWM_CPS	V/O	0...FFH	0...99.60937	0.390625	[%]
Corrected PWM signal for CPS opening					
CPPWM	V	0...FFH	0...99.60937	0.390625	[%]
PWM signal for CPS opening					
FRQ_REQ_CP	V/O	0...FFH	0...255	1	[Hz]
CPS Output frequency					
LV_CP_CLOSE_ACT	V/O	0...1H	0...1	1	[-]
Logical value for CPS valve closed					
LV_FRQ_1_CPS	V	0...1H	0...1	1	[-]
Logical value for CPS PWM frequency					

Input data:

N_32	VB	LV_CPPWM_EXT_ADJ	CPPWM_EXT_ADJ
LV_ERR_EL_CPS	CPPWM_LIH	FLOW_COR_CPS	CPPWM_ADD_AD
LV_N_MAX_CP	PRS_CPS	FRQ_CPS_AD	LV_CPS_AD_ACT

Input data table order: External inputs – Inputs from other EVAC modules

Import actions:

ACTION_INFR_SetPWMDucyCPS(IN<Cppwm_cps>)
This action chages the duty cycle of the PWM output signal
ACTION_INFR_SetPWMFrqCPS(IN<Frq_req_cp>)
This action changes the frequency of the PWM output signal

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_CPPWM	16*16	0...FFH	0...99.60937	0.390625	[%]
LDP_FLOW_COR_CPS_IP_CPPWM	16	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
LDPM_PRS_CPS_1_EVAC	16	0...FFFFH	-2717... 2716.91708	0.0829163	[hPa]
CPS valve characteristic					
IP_CPPWM_COR_FRQ	8	0...FFH	0...1.99218	0.0078125	[-]
LDP_CPPWM_IP_CPPWM_COR_FRQ	8	0...FFH	0...99.60937	0.390625	[%]
PWM correction following frequency value					
IP_CPPWM_VB_ADD_1	8	0...FFH	-50...49.60937	0.390625	[%]
LDPM_VB_1_EVAC	8	0...FFH	0...25.89843	0.1015625	[V]
Additive VB correction for CPPWM and frequency 1					
IP_CPPWM_VB_ADD_2	8	0...FFH	-50...49.60937	0.390625	[%]
LDPM_VB_1_EVAC	8	0...FFH	0...25.89843	0.1015625	[V]
Additive VB correction for CPPWM and frequency 2					
IP_CPPWM_VB_FAC	8	0...FFH	0...1.99218	0.0078125	[-]
LDPM_VB_1_EVAC	8	0...FFH	0...25.89843	0.1015625	[V]
Multiplicative VB correction for CPPWM					
IP_CPPWM_VB_FAC_PRS_CPS	6	0...FFH	0...3.98437	0.015625	[-]
LDP_PRS_CPS_IP_CPPWM_VB_FAC	6	0...FFFFH	-2717... 2716.91708	0.0829163	[hPa]
Multiplicative VB correction for CPPWM					
C_N_FRQ_1_CPS	1	0...FFH	0...8160	32	[rpm]
Threshold on engine speed to shift CPS frequency output					
C_N_FRQ_2_CPS	1	0...FFH	0...8160	32	[rpm]
Threshold on engine speed to shift CPS frequency output					
C_N_FRQ_HYS_CPS	1	0...FFH	0...8160	32	[rpm]
Hysteresis threshold on engine speed to shift CPS frequency output					
C_CPPWM_FRQ_1_CPS	1	0...FFH	0...99.60937	0.390625	[%]
Threshold on CPPWM to shift CPS frequency output					
C_CPPWM_FRQ_2_CPS	1	0...FFH	0...99.60937	0.390625	[%]
Threshold on CPPWM to shift CPS frequency output					
C_CPPWM_FRQ_HYS_CPS	1	0...FFH	0...99.60937	0.390625	[%]
Hysteresis threshold on CPPWM to shift CPS frequency output					
C_CPPWM_FRQ_1	1	0...FFH	0...255	1	[Hz]
First CPS output frequency					
C_CPPWM_FRQ_2	1	0...FFH	0...255	1	[Hz]
Second CPS output frequency					
C_CPPWM_MAX_FRQ_1_LIM_CP	1	0...FFH	0...99.60937	0.390625	[%]
CPPWM_CPS maximum limit to assure the diagnosis with frequency 1					
C_CPPWM_MAX_FRQ_2_LIM_CP	1	0...FFH	0...99.60937	0.390625	[%]
CPPWM_CPS maximum limit to assure the diagnosis with frequency 2					
C_CPPWM_MIN_FRQ_1_LIM_CP	1	0...FFH	0...99.60937	0.390625	[%]
CPPWM_CPS minimum limit to assure the diagnosis with frequency 1					
C_CPPWM_MIN_FRQ_2_LIM_CP	1	0...FFH	0...99.60937	0.390625	[%]
CPPWM_CPS minimum limit to assure the diagnosis with frequency 2					
C_CPPWM_AS	1	80...7FH	-100...99.21875	0.78125	[%]
Application system value for CPS opening					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_CPPWM_LIM_DIAG	1	0...1H	0...1	1	[-]
Selection of diagnosis limit values on CPPWM_CPS					
LC_CPPWM_AS	1	0...1H	0...1	1	[-]
C_CPPWM_AS activation					

General information:

The PWM-setpoint to control the opening of the canister purge valve CPPWM is calculated from a map as a function of the flow setpoint and the pressure difference. This flow setpoint is deliverd from evapoarative emission control module.

The influence of the battery voltage VB on the opening of the CPS is corrected by the additive value IP_CPPWM_VB_1_ADD or IP_CPPWM_VB_2_ADD and multiplicative value IP_CPPWM_VB_FAC.

Two frequencies of the CPS are possible; the selection of the current frequency is dependent on engine speed and valve opening.

An additional PWM offset can be applied using C_CPPWM_AS (the offset) and LC_CPPWM_AS (the switch to turn the consideration of the offset on).

The term CPPWM_ADD_AD includes the valve adaptation value delivered from the canister purge valve characteristic adapation function (if included).

An external setpoint CPPWM_EXT_ADJ is applied, if LV_CPPWM_EXT_ADJ is true.

If a fault is detected on the canister purge valve (LV_ERR_EL_CPS=1), the opening of the CPS is controlled by the value of CPPWM_LIH (see chapter OBDI functions).

In case of high CPU load / high engine speed (LV_N_MAX_CP=1), the regular control of the valve is stopped to save CPU load.

The physical range for the frequency request FRQ_REQ_CP is 5 to 32 Hz.

Application conditions:

Initialisation:	LV_CP_CLOSE_ACT=1 and all other values =0	at reset or stalling
Recurrence:	**20** ms	
Activation:	*every engine state*	
Deactivation	--	

Signal flow diagrams:

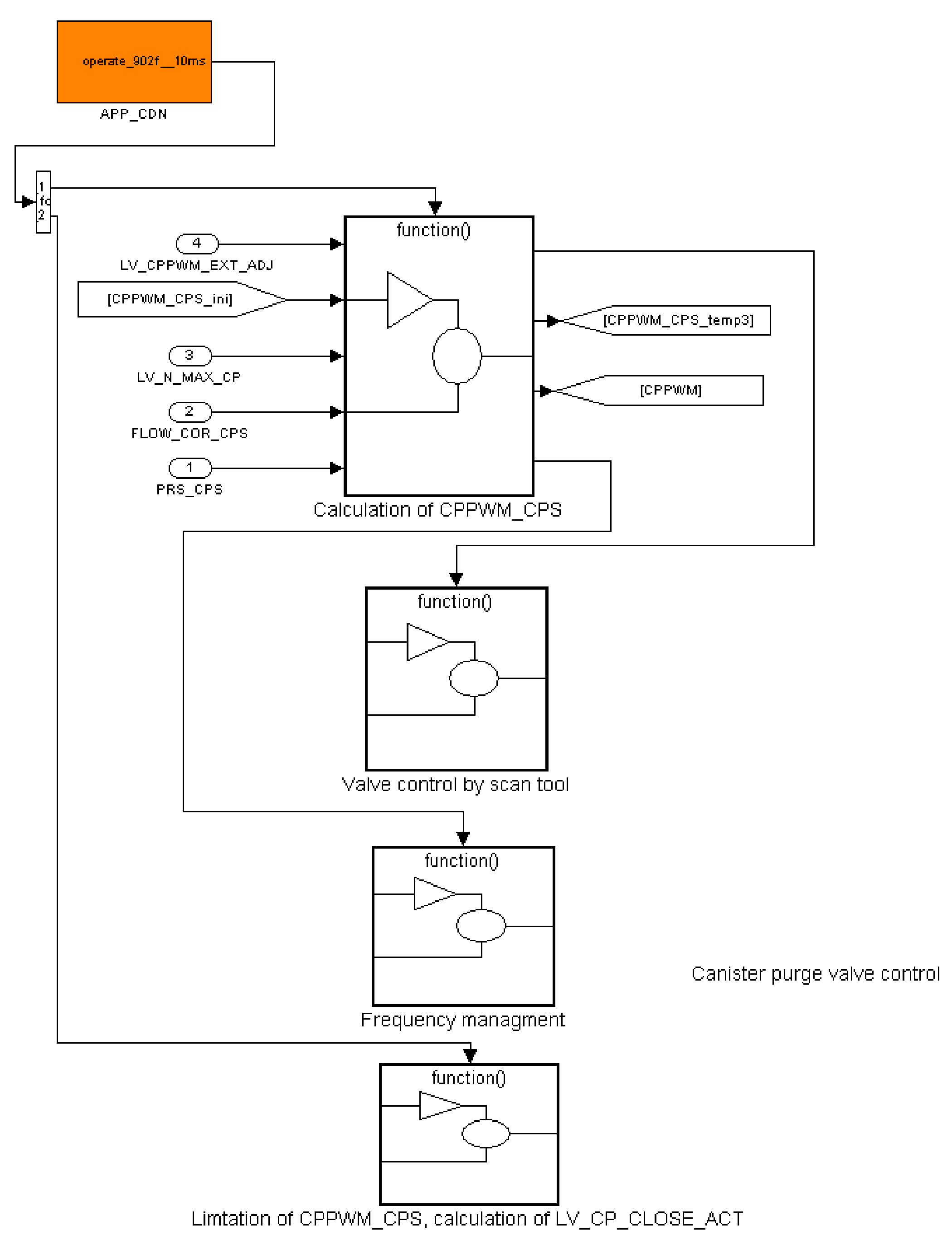


figure 1: overview valve control

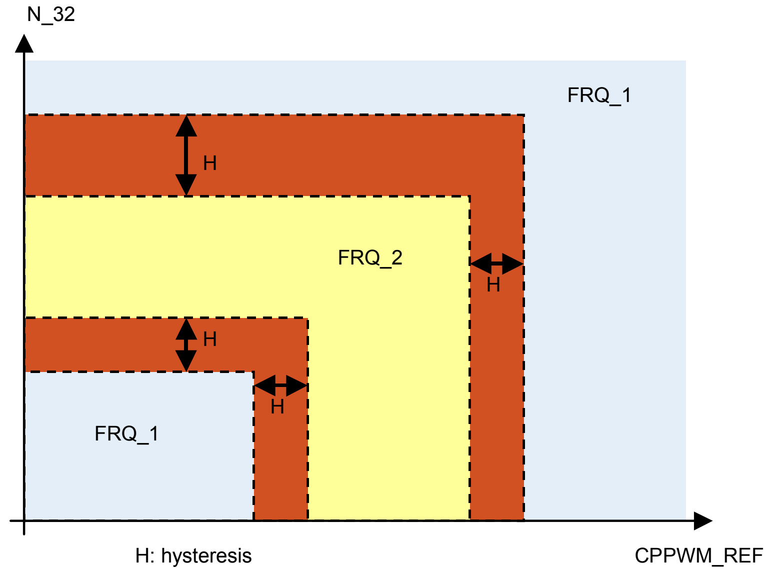


figure 3: frequency management

Formula section:

Calculation of CPPWM_CPS

```
if    LV_CPPWM_EXT_ADJ = 1
then  call "Valve control by scan tool"
else  if    LV_N_MAX_CP  = 1
      then  CPPWM_CPS    = 0
      else  if        LV_ERR_EL_CPS = 1
            then      CPPWM    = CPPWM_LIH
            elseif         LV_CPS_AD_ACT = 1
            then      CPPWM    = CPPWM_ADD_AD
            elseif         FLOW_COR_CPS = 0
            then      CPPWM    = 0
            else      CPPWM    = max [0, IP_CPPWM (FLOW_COR_CPS, PRS_CPS)
                                 + CPPWM_ADD_AD]
            endif
            if        LC_CPPWM_AS = 1
            then      CPPWM          = CPPWM + C_CPPWM_AS
            endif

            call "Frequency management"
      endif
endif
call  "Limitation of CPPWM_CPS, calculation of LV_CP_CLOSE_ACT"
end
```

28.14.1 Valve control by scan tool

```
if    LV_CPPWM_EXT_ADJ = 1
then  FRQ_REQ_CP          = C_CPPWM_FRQ_2
      LV_FRQ_1_CPS        = 0
      CPPWM               = CPPWM_EXT_ADJ
```

```
          CPPWM_CPS           = CPPWM_EXT_ADJ
          TMP_MAX_CPS_DIAG = C_CPPWM_MAX_FRQ_2_LIM_CP
          TMP_MIN_CPS_DIAG = C_CPPWM_MIN_CPS_DIAG_FRQ_2
    endif
    end
```

28.14.2 Frequency management

```
    frequency selection
    if    LV_CPS_AD_ACT       = 1
    then  FRQ_REQ_CP          = FRQ_CPS_AD
          LV_FRQ_1_CPS        = 0

    if    LV_FRQ_1_CPS = 1
    then  if    (CPPWM          < C_CPPWM_FRQ_2_CPS)
          and   (N_32           < C_N_FRQ_2_CPS)
          and   [ (CPPWM        > C_CPPWM_FRQ_1_CPS + C_CPPWM_FRQ_HYS_CPS)
          or    (N_32           > C_N_FRQ_1_CPS     + C_N_FRQ_HYS_CPS) ]
          then  FRQ_REQ_CP     = C_CPPWM_FRQ_2
                LV_FRQ_1_CPS   = 0
          else  FRQ_REQ_CP     = C_CPPWM_FRQ_1
                LV_FRQ_1_CPS   = 1
          endif
    else  if    (CPPWM          > C_CPPWM_FRQ_2_CPS + C_CPPWM_FRQ_HYS_CPS)
          or    (N_32           > C_N_FRQ_2_CPS     + C_N_FRQ_HYS_CPS)
          or    [ (CPPWM        < C_CPPWM_FRQ_1_CPS)
          and   (N_32           < C_N_FRQ_1_CPS) ]
          then  FRQ_REQ_CP     = C_CPPWM_FRQ_1
                LV_FRQ_1_CPS   = 1
          else  FRQ_REQ_CP     = C_CPPWM_FRQ_2
                LV_FRQ_1_CPS   = 0
          endif
    endif


    battery voltage correction & diagnosis limit determination
    if    CPPWM > 0
    then  if    LV_FRQ_1_CPS = 1
          then  CPPWM_CPS =
                CPPWM
                * IP_CPPWM_COR_FRQ (CPPWM)
                * IP_CPPWM_VB_FAC (VB)
                + IP_CPPWM_VB_ADD_1 (VB) * IP_CPPWM_VB_FAC_PRS_CPS (PRS_CPS)
                TMP_MIN_CPS_DIAG  = C_CPPWM_MIN_FRQ_1_LIM_CP
                TMP_MAX_CPS_DIAG = C_CPPWM_MAX_FRQ_1_LIM_CP
          else  CPPWM_CPS =
                CPPWM
                * IP_CPPWM_VB_FAC (VB)
                + IP_CPPWM_VB_ADD_2 (VB) *  IP_CPPWM_VB_FAC_PRS_CPS (PRS_CPS)
```

```
                TMP_MIN_CPS_DIAG  = C_CPPWM_MIN_FRQ_2_LIM_CP
                TMP_MAX_CPS_DIAG = C_CPPWM_MAX_FRQ_2_LIM_CP
            endif
    else    CPPWM_CPS = 0
            TMP_MIN_CPS_DIAG = C_CPPWM_MIN_FRQ_2_LIM_CP
            // TMP_MIN_CPS_DIAG is used for calculation of LV_CP_CLOSE_ACT, should be
            defined although CPPWM_CPS=0.
    endif
    end
```

28.14.3 Limitation of CPPWM_CPS, calculation of LV_CP_CLOSE_ACT

```
    if      LC_CPPWM_LIM_DIAG = 1
    then    if      CPPWM_CPS > 0
            then    CPPWM_CPS = min [ max [
                    CPPWM_CPS, TMP_MIN_CPS_DIAG], TMP_MAX_CPS_DIAG]
            endif
            LV_CP_CLOSE_ACT = [CPPWM_CPS <= TMP_MIN_CPS_DIAG]
    else    LV_CP_CLOSE_ACT = [CPPWM_CPS = 0]
    endif
    end
```

Call infrastructure to update the PWM duty-cycle

ACTION_INFR_SetPWMDucyCPS(CPPWM_CPS)

Call infrastructure to update the PWM frequency

ACTION_INFR_SetPWMFrqCPS(FRQ_REQ_CP)

28.15 Canister purge valve adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CPS_AD_RUN_CDN_OK	O/V	0...1H	0...1	1	-
all conditions for adaptation (including counter) fulfilled, adaptation can run					
LV_CPS_AD_ACT	O/V	0...1H	0...1	1	-
if positive: adaptation is active or waiting					
FRQ_CPS_AD	O/V	0...FFH	0...255	1	Hz
current CPS frequency for adaptation					
CPPWM_ADD_AD	O/V	0...7FFFH	0...99.996948	0.00305176	%
Additive adaptive CPPWM correction					
LV_CPS_AD_HOM_REQ	O/V	0...1H	0...1	1	-
Flag to request homogeneous engine operation mode for CPS adaptation					
CPPWM_ADD_AD_MEM	V/S	0...FFFFH	0...99.998474	0.00152588	%
final output of function: additive STO-parameter written to non-volatile memory					
CTR_1_CPS_AD	V	0...FFFFH	0...1.3107E+3	0.02	s
function duration time (in APP_CDN)					
CTR_2_CPS_AD	V	0...FFFFH	0...1.3107E+3	0.02	s
internal time counter for wait-states and ramps					
CTR_CYCNR_CPS_AD	V	0...FFH	0...255	1	-
number of successful adaptation cycles during driving cycle					
FAC_FLOW_CPS_AD	V	0...FFH	0...0.99609375	0.00390625	-
STO-weighting factor depending on FLOW_COR_CPS					
FAC_LAMB_START_CPS_AD	V	8000...7FFFH	-50...49.9984741	0.00152588	%
lambda deviation at begin of CPS_AD to eliminate a constant lambda-offset					
FAC_PRS_CPS_AD	V	0...FFH	0...1.9921875	0.0078125	-
STO-weighting factor depending on PRS_CPS					
FAC_PWM_CPS_AD	V	0...FFH	0...0.99609375	0.00390625	-
STO-weighting factor depending on CPPWM_CPS					
LV_CL_OK_CPS_AD	V	0...1H	0...1	1	-
CL is sufficient for adaptation					
LV_CTR_OK_CPS_AD	V	0...1H	0...1	1	-
CPS_AD has not yet successfully finished the required number of cycles					
LV_CPS_AD_ST_CDN_OK	V	0...1H	0...1	1	-
all conditions for adaptation (including CL) fulfilled, adaptation can start					
T_CPS_AD_HOM_REQ	V	0...FFFFH	0...1.3107E+3	0.02	s
For this time homogeneous operation mode is requested (LV_CPS_AD_HOM_REQ = 1)					
CTR_CPS_AD_HOM_REQ	V	0...FFH	0...255	1	-
Number of homogeneous operation mode requests from CPS adaptation.					
T_DLY_CPS_AD_HOM_REQ	V	0...FFFFH	0...1.3107E+3	0.02	s
Time until the next homogeneous engine operation mode requests by CPS adaptation is possible					
LV_CPS_OK_CPS_AD	V	0...1H	0...1	1	-
CP-Sytem is ready for adaptation (STATE_CP etc.)					
LV_ENG_OK_CPS_AD	V	0...1H	0...1	1	-
engine conditions (N, etc.) are OK for adaptation					
LV_ERR_CPS_AD	V	0...1H	0...1	1	-
during current driving cycle one or more adaptations of CPS have failed					
LV_T_OK_CPS_AD	V	0...1H	0...1	1	-
all relevant time counters are OK for adaptation					
RATIO_FRQ	V	0...FFH	0...15.9375	0.0625	-
ratio FRQ used for adaptation / minimum purge frequency					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CPS_AD	V	0H 1H 2H 4H 5H 6H	INACTIVE WAITING LEARNING INTERRUPTED FAILED FINISHED	1	-
current state of CPS-adaptation reported by function					
STATE_CPS_AD_REQ	V	0H 1H 2H 4H 5H 6H	INACTIVE WAITING LEARNING INTERRUPTED FAILED FINISHED	1	-
state of CPS-adaptation requested by APP_CDN (=activate etc.)					

Input data:

STATE_CP	LV_CL_CLC_VLD	LV_CPS_AD_INH	LV_CP_CLOSE_REQ
LV_HOM_ACT	N	TQI_AV	FAC_LAM_MV[NC_CBK_EX_NR]
T_AST_CP	CL_MMV	PRS_CPS	FLOW_COR_CPS
CPPWM_CPS			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CL_MIN_CPS_AD	1	0...FFFFH	0...1.999969	3.05176E-5	-
minimum canister load for CPS-adaptation					
C_CPPWM_ADD_AD_MEM_NVMY_INI	1	0...FFFFH	0...99.9984741	0.00152588	%
First value of the additive STO parameter written to non-volatile memory at NVMY_INI					
C_CPPWM_AD_RATIO	1	0...FFH	0...0.99609375	0.00390625	-
ratio new measured / old memorized value for new memorized CPPWM_ADD_AD_MEM					
C_CPPWM_DELTA_AD	1	0...FFFFH	0...99.998474	0.00152588	%
PWM-distance to old STO for start adaptation					
C_CPPWM_DLY_LAM	1	0...FFFFH	0...99.998474	0.00152588	%
delta PWM between STO and lambda reaction (time delay compensation)					
C_CPPWM_INC_AD	1	0...FFFFH	0...99.998474	0.00152588	%
PWM-increment for stop after aborted adaptation					
C_CPPWM_INC_STOP_AD	1	0...FFFFH	0...99.998474	0.00152588	%
PWM-increment for stop after aborted adaptation					
C_CPPWM_MAX_AD	1	0...FFFFH	0...99.998474	0.00152588	%
maximum PWM for CPS-adaptation					
C_CPPWM_MIN_AD	1	0...FFFFH	0...99.998474	0.00152588	%
minimum PWM-threshold for CPS-adaptation					
C_CTR_CPS_AD_HOM_REQ_MAX	1	0...FFH	0...255	1	-
Maximum number of homogeneous operation mode requests from CPS adaptation.					
C_CYCNR_CPS_AD	1	0...FFH	0...255	1	-
maximum number of adaption-cycles per driving cycle					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_LAMB_DELTA_MIN_CPS_AD	1	8000...7FFFH	-50...49.9984741	0.00152588	%
lowest detectable lambda deviation					
C_FAC_LAMB_MAX_CPS_AD	1	8000...7FFFH	-50...49.9984741	0.00152588	%
threshold for CPS-adaptation-abortion					
C_FLOW_CLC_DIF_CPS_AD	1	0...FFFFH	0...7.99987793	1.2207E-4	kg/h
Difference between C_FLOW_CLC_MAX_CPS_AD and begin of ramp					
C_FLOW_CLC_MAX_CPS_AD	1	0...FFFFH	0...7.99987793	1.2207E-4	kg/h
Maximum FLOW threshold for weighting-STO ramp					
C_FRQ_CLC_1_CPS_AD	1	0...FFH	0...255	1	Hz
numerator for adaptation frequency calculation					
C_FRQ_CLC_2_CPS_AD	1	80...7FH	-128...127	1	Hz
offset for adaptation frequency calculation					
C_FRQ_MAX_MEC_CPS	1	0...FFH	0...255	1	Hz
maximum allowed mechanical frequency for CPS					
C_FRQ_MIN_CPS_AD	1	0...FFH	0...255	1	Hz
start-frequency for finding adaptation frequency					
C_N_MAX_CPS_AD	1	0...1FE0H	0...8.16E+3	1	rpm
upper threshold of dynamic idle-window for CPS-adaptation					
C_N_MIN_CPS_AD	1	0...1FE0H	0...8.16E+3	1	rpm
lower threshold of dynamic idle-window for CPS-adaptation					
C_PRS_CPS_MAX_CPS_AD	1	8000...7FFFH	-2.717E+3 ... 2.71696E+3	0.08291752	hPa
Maximum PRS_CPS-threshold for CPS-adaptation					
C_PWM_CLC_DIF_CPS_AD	1	0...FFH	0...99.609375	0.390625	%
Difference between CL_PWM_CLC_MAX_CPS_AD and begin of ramp					
C_PWM_CLC_MAX_CPS_AD	1	0...FFH	0...99.609375	0.390625	%
Maximum CPPWM_CPS threshold for weighting-STO-ramp					
C_TQI_AV_MAX_CPS_AD	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
upper threshold of dynamic idle-window for CPS-adaptation					
C_TQI_AV_MIN_CPS_AD	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
lower threshold of dynamic idle-window for CPS-adaptation					
C_T_AST_DEAC_CPS_AD	1	0...FFFFH	0...6.5535E+3	0.1	s
Maximum start time for adaptation					
C_T_CPS_AD_HOM_REQ	1	0...FFFFH	0...1.3107E+3	0.02	s
Time while hom-request is maintained (even if due to operation mode change not all CPS adaptation conditions are met)					
C_T_DLY_CPS_AD_HOM_REQ	1	0...FFFFH	0...1.3107E+3	0.02	s
Minimum time between two homogeneous engine operation mode requests by CPS adaptation					
C_T_DLY_FAC_LAMB_CPPWM_AD	1	0...FFFFH	0...6.5535E+3	0.1	s
time delay after change of fuel massflow during detectable lambda deviation (expecting 0 or above)					
C_T_INC_CPS_CLOSE	1	0...FFFFH	0...6.5535E+3	0.1	s
time increment between cps-closing-steps					
C_T_MAX_CPS_AD	1	0...FFFFH	0...6.5535E+3	0.1	s
maximum function duration time					
C_T_ST_CPS_AD	1	0...FFFFH	0...6.5535E+3	0.1	s
minimum time after start engine to allow CPS-adaptation					
C_T_WAIT_CPS_AD	1	0...FFFFH	0...6.5535E+3	0.1	s
time delay after function call until start open CPS					
C_T_WAIT_LAMB_RGN	1	0...FFFFH	0...6.5535E+3	0.1	s
time after adaptation for "regeneration" of lambda-regualtion					
LC_CPS_AD_CHK	1	0...1H	0...1	1	-
Run adaptation for check, overriding conditions					
IP_CPPWM_OPEN_CPS	6	0...FFH	0...1.9921875	0.0078125	-
LDP_PRS_CPS_IP_CPPWM_OPEN_CPS	6	8000...7FFFH	-2.717E+3 ... 2.71696E+3	0.08291752	hPa
Weighting factor for STO according to CPS characteristic					

28.15.1 General information

The purpose of the CPS_AD function is to find the STO(start-to-open-point) of the canister-purge-solenoid (CPS): The PWM-value (valid for the default frequency C_FRQ_MIN_CPS_AD) at which the CPS (canister-purge-solenoid-valve) has reached the end of it’s mechatronical delay time and starts to open (STO). Therefore the CPS is opened using a ramp, meanwhile monitoring the lambda-adaptation (FAC_LAMB) to detect a deviation caused by fuel purged through the CPS. Due to the fact that at a low canister-load (CL_MMV) the adaptation is getting less accurate or even impossible the purge-massflow during adaptation can be increased by rising the frequency (to have more opening-periods per second). The frequency is calculated according to CL_MMV. For PWM is a relative value (valid only for a special frequency) all PWM-values have to be corrected after computing the frequency.

The function CPS_AD (CPS-adaptation) is part of the EVAC-aggregate. To allow a fast purge of the ACF the PWM-ramp for opening the CPS has to start at the STO.

APP_CDN:

This chart is the function’s internal state-machine.

When the conditions (i.e. engine speed, canister load) are sufficient (checked by the function START_CDN_OK), the state active (substate adaptation) is entered.

For transparency the conditions are split into the flags: LV_CPS_AD_ST_CDN_OK, LV_CPS_AD_RUN_CDN_OK, LV_ENG_OK_CPS_ AD, LV_CPS_OK_CPS_AD, LV_T_OK_CPS_ AD, LV_CL_OK_CPS_ AD, LV_CTR_OK_CPS_AD. For details see APP_CDN-block.

The function LEARN_STO ist triggered during state active (by triggering the containing subsystem LEARN_STO). The output value STATE_CPS_AD_REQ is used to control the function’s internal state while STATE_CPS_AD is the reporting value from the function. After an interrupt or failed adaptation the flag LV_ERR_CPS_AD is set. After that in state closing the valve is closed and in state waiting there is time to compensate any lambda-deviation caused by the function.

The value LC_CPS_AD_CHK allws an override of the CPS_AD-conditions.

If maybe once a LEARN_SLOPE-function will be added, the state active will have to be divided into the substates LEARN_STO_active and LEARN_SLOPE_active.

At abortion (to avoid a lambda-step by suddenly closing the CPS) the function has to be shutdown by entering state CLOSE. This can be caused by changing conditions like N or by flags like LV_CPS_AD_INH. In this case the function goes into state CLOSING to prevent any further influence on the engine.

A decreasing CL_MMV value during adaptation is normal but – after a sufficient CL-value has been assured before function start – does not require a shutdown; the value LV_CPS_AD_RUN_CDN_OK is still 1.

If all conditions for CPS adaptation are met but the engine is not in homogeneous operation mode, this mode is requested via the flag LV_CPS_AD_HOM_REQ.

This request is hold up for the time C_T_CPS_AD_HOM_REQ or - if the engine mode changes - until the adaptation is finished or was interrupted. After a non-successful request or after the adaptation is finished / was interrupted, for the time C_T_DLY_CPS_AD_HOM_REQ no request for homogeneous operation mode is allowed.

After a non-successful request or after the adaptation is finished/ was interrupted, for the time C_T_DLY_CPS_AD_HOM_REQ no requests for homogeneous mode are allowed.

Per driving cycle only C_CTR_CPS_AD_HOM_REQ requests are allowed, afterwards CPS adaptation can only take place if the engine is in homogeneous operation mode due to other conditions.

Application Condition

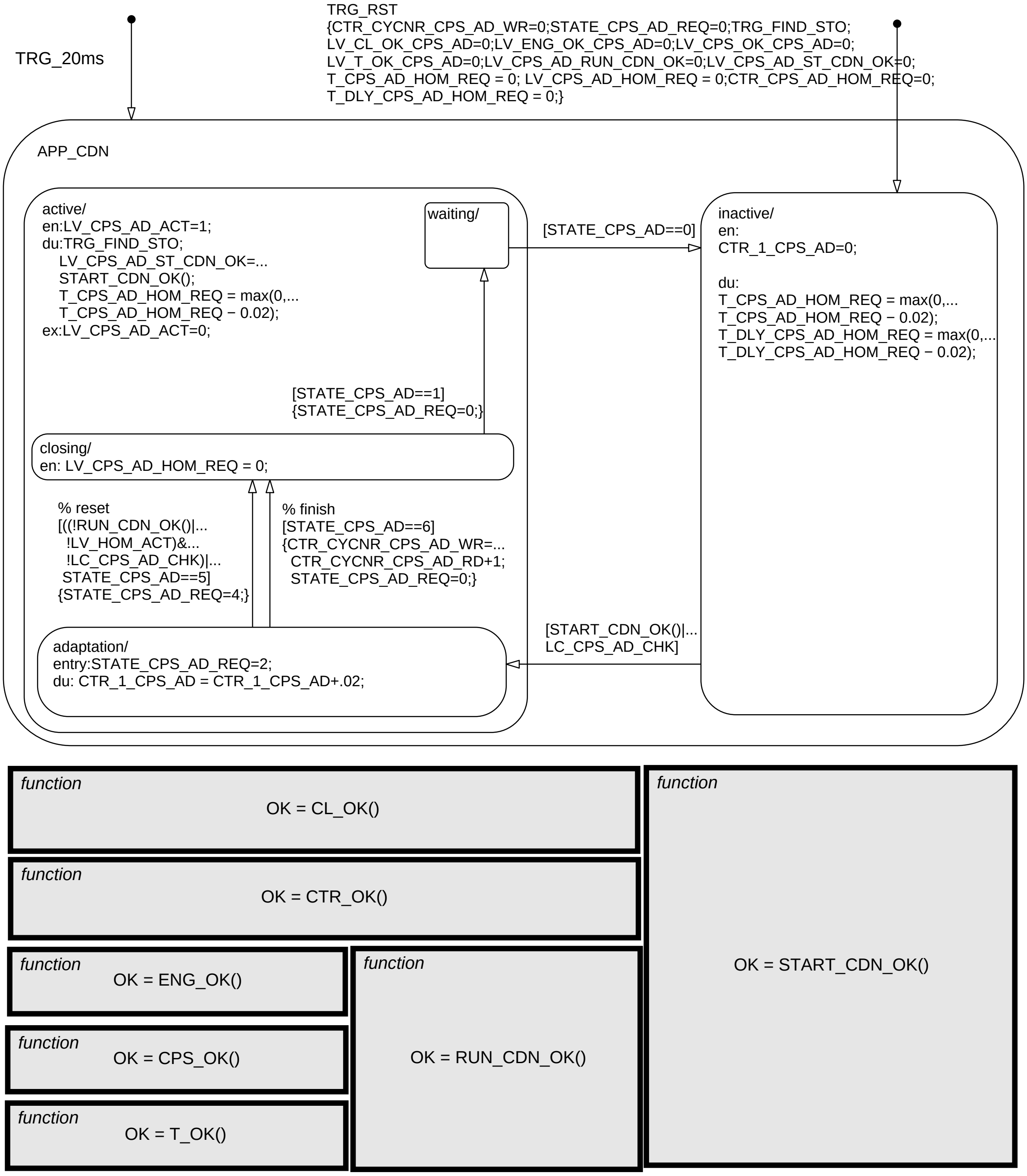


Figure 1 EVAC_CPS_AD/ APP_CDN/ APP_CDN_CHART

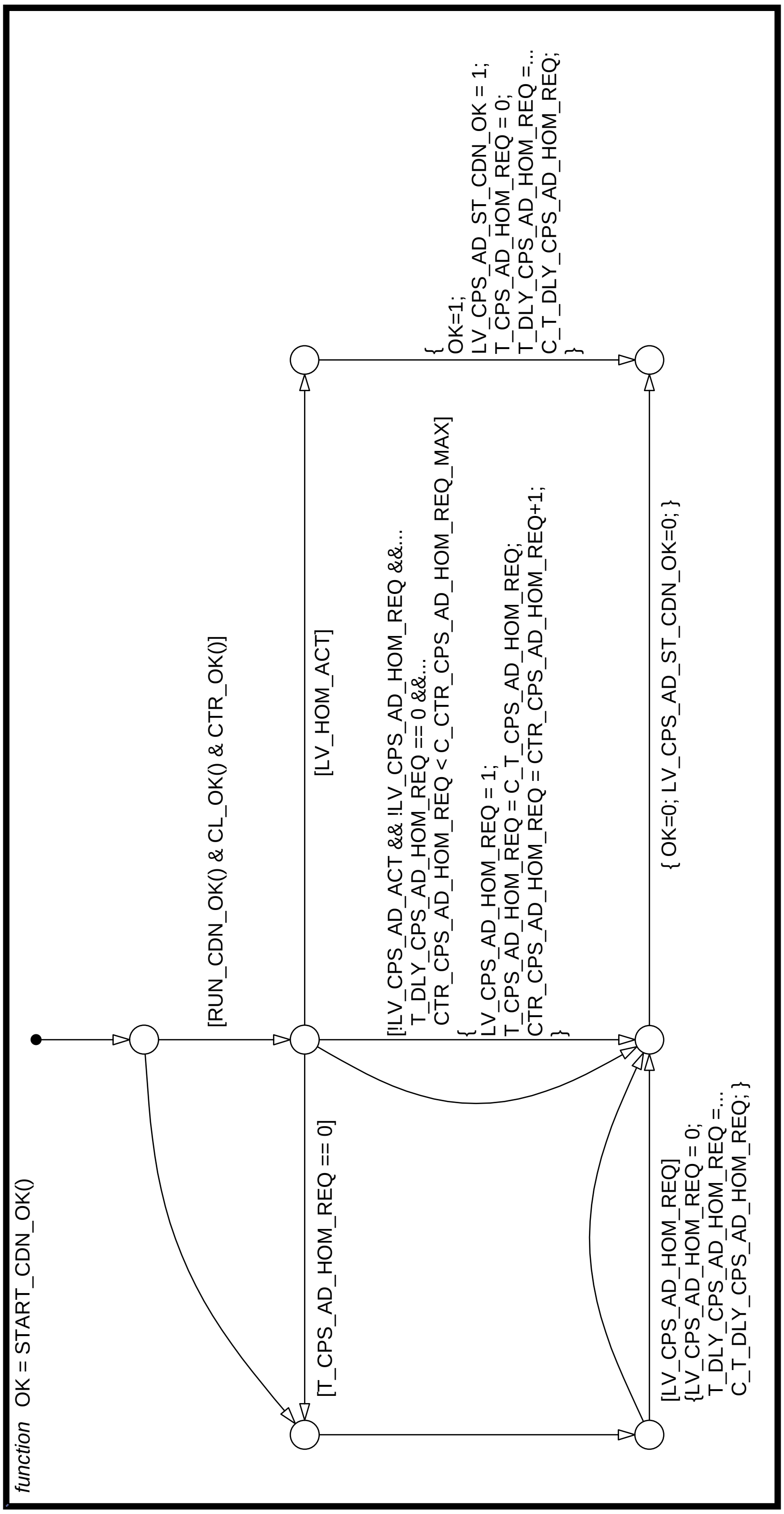


Figure 2 EVAC_CPS_AD/ APP_CDN/ APP_CDN_CHART/ START_CDN_OK

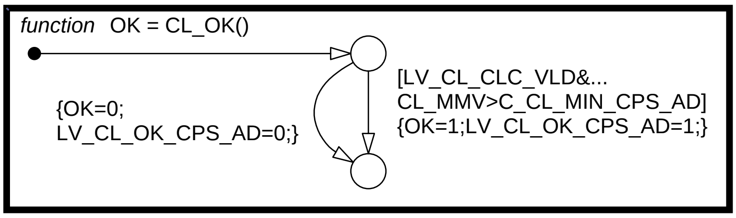


Figure 3 EVAC_CPS_AD/ APP_CDN/ APP_CDN_CHART/ CL_OK

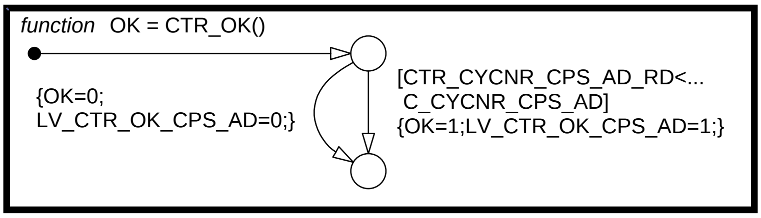


Figure 4 EVAC_CPS_AD/ APP_CDN/ APP_CDN_CHART/ CTR_OK

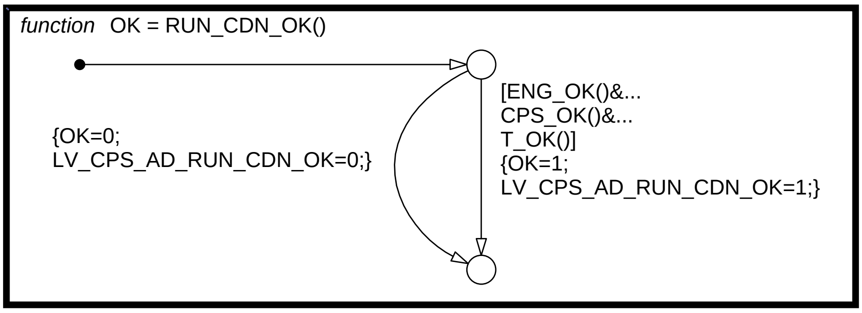


Figure 5 EVAC_CPS_AD/ APP_CDN/ APP_CDN_CHART/ RUN_CDN_OK

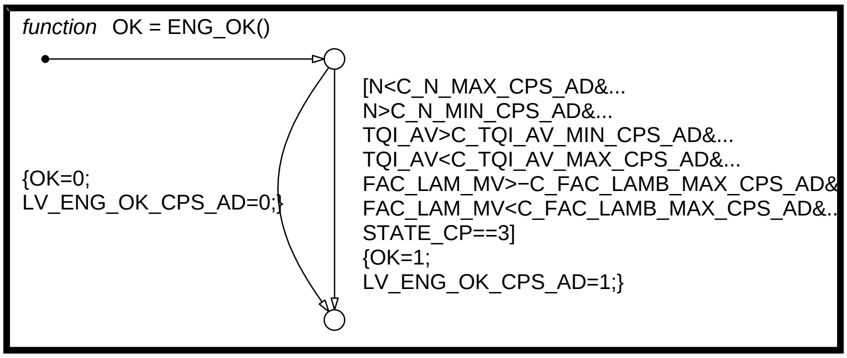


Figure 6 EVAC_CPS_AD/ APP_CDN/ APP_CDN_CHART/ ENG_OK

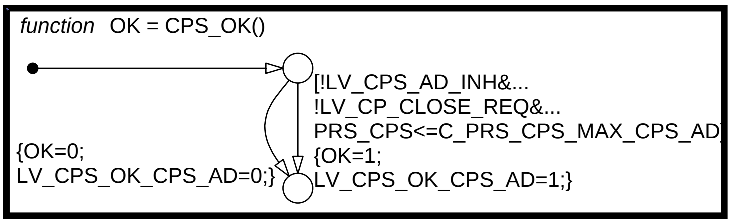


Figure 7 EVAC_CPS_AD/ APP_CDN/ APP_CDN_CHART/ CPS_OK

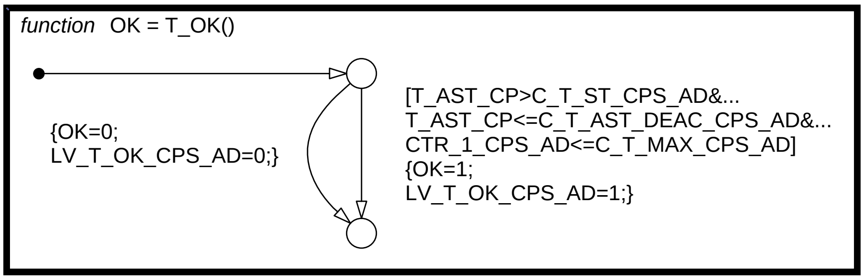
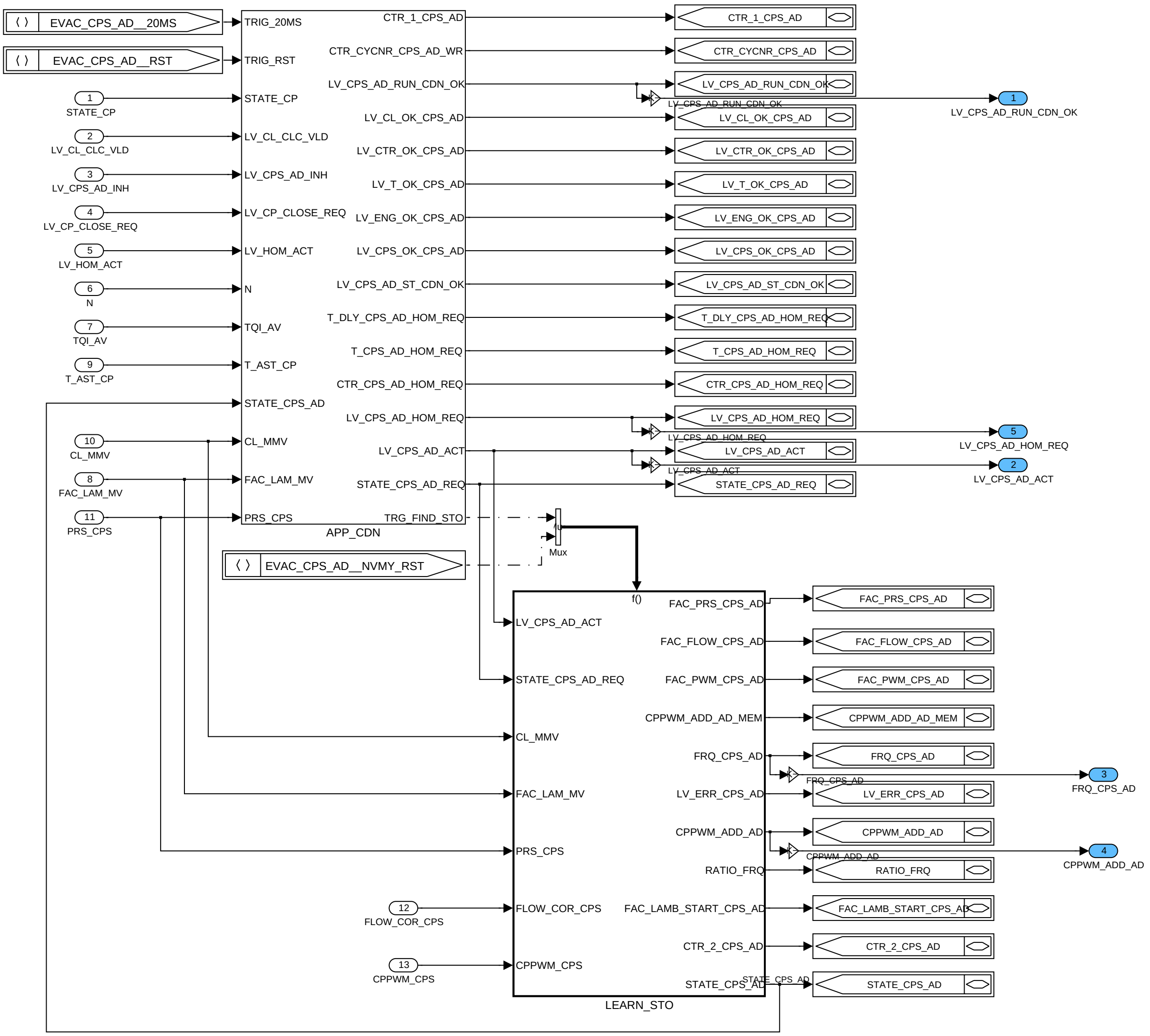


Figure 8 EVAC_CPS_AD/ APP_CDN/ APP_CDN_CHART/ T_OK

Function Description



SDA_SRS / SDA 4.0 15−Sep−2005

Figure 9 EVAC_CPS_AD

28.15.1.1 SUBFUNCTION: LEARN_STO

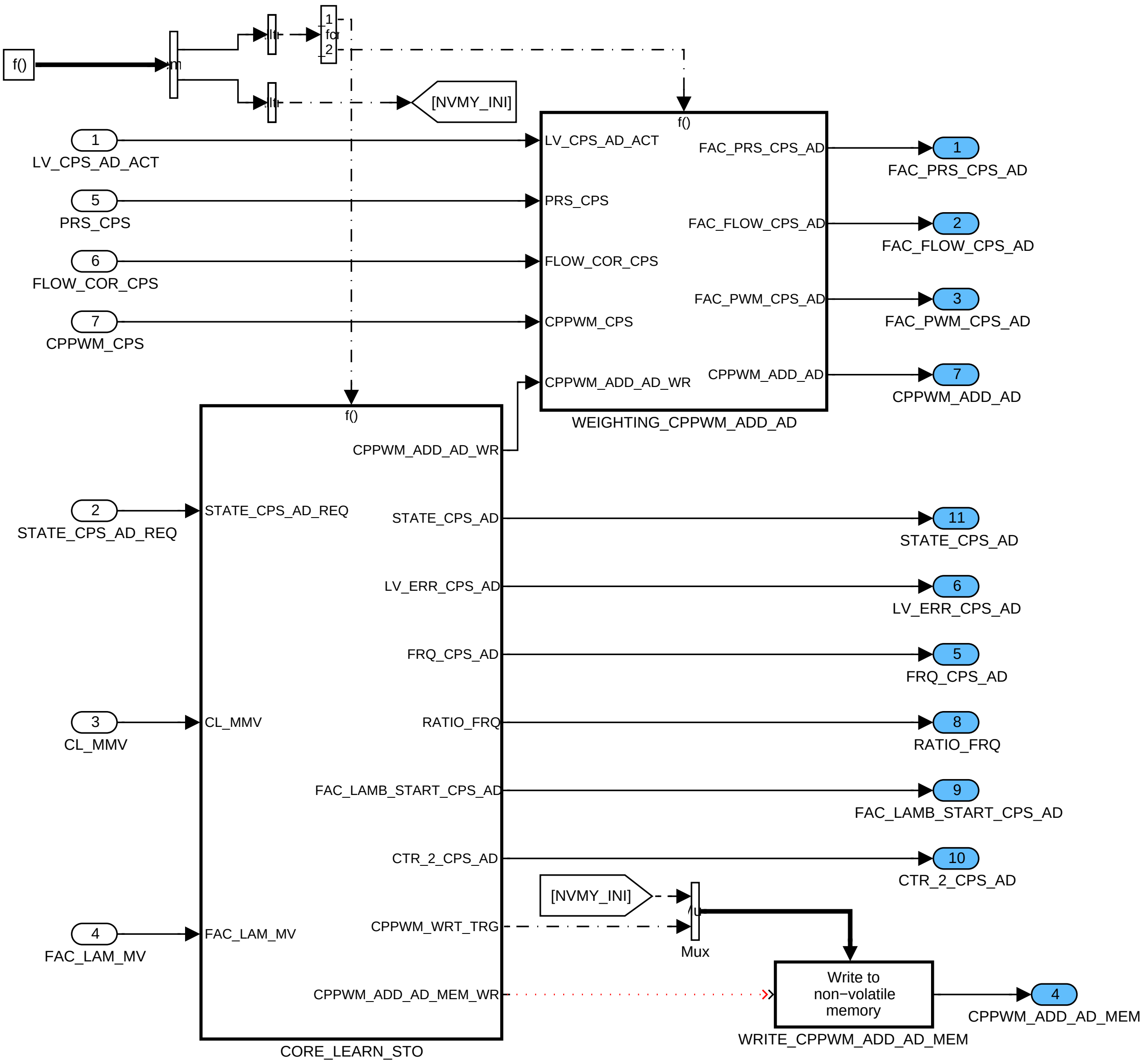


Figure 10 EVAC_CPS_AD/ LEARN_STO

<u>EVAC_CPS_AD/LEARN_STO/CORE_LEARN_STO</u>

This chart is the core of the CPS_AD-function. The states are:

Entry:

When entering the chart outputs are set to default values.

In case of a CPPWM_ADD_AD_MEM > maximum threshold (danger of engine disturbance) the memory value is set to minimum threshold.

CHECK_CONDITIONS:

This first substate is to prevent a new initialisation (state INIT) of the chart at the last trigger before it's state machine (APP_CDN) is leaving the state active. If PWM-adaptation is demanded by APP_CDN, the state CHECK_CONDITION will be left because of STATE_CPS_AD_REQ==2.

INIT_WAIT:

STATE_CPS_AD is set to 1 eporting that the function is in a wait-state (to allow manipulations during implementation using T_WAIT or provide some additional time for lambda recreation after any disturbance in the past).

CPPWM_ADD_AD_MEM_WR (output variable for CPPWM_ADD_AD_MEM in the non-volatile memory) is set to the last known value of CPPWM_ADD_AD_MEM.

calculations:

FAC_LAMB_START is the memory for the current FAC_LAMB-value to calculate a lambda-deviation caused by the function eliminating a constant lambda-offset.

To minimize the measuring error during STO-adaptation the adaptation has to be done at highest possible frequency to measure the highest possible fuel massflow. This is the frequency where the lambda-deviation caused by the opening-increment (CPPWM_INC_AD) is just small enough not to disturb the engine. CPPWM_INC_AD just as several other values are here adapted to the new frequency, using RATIO_FRQ which is calculated using the canister load CL_MMV.

Before entering INCREASE_PWM STATE_CPS is set to 7 reporting that PWM-adaptation is in process.

The two junctions are to avoid a calculated FRQ which is above C_FRQ_MAX_MEC_CPS.

IMPORTANT:

The code *{FRQ=abs(...* is paid to a stateflow-bug for *abs(* in stateflow is really *abs(fix(* and there aint't no *fix(* in stateflow. The correct mathematical expression here is *{FRQ=fix(...* to cut out the comma.

INCREASE_PWM:

Here the STO is detected. Frequent entering of the junction beside allows the monitoring of the lambda-deviation while opening the CPS. The time delay C_T_DLY_FAC_LAMB is to let the lambda-sensor detect the HC purged into the manifold.

If the function is interrupted (STATE_CPS_AD_REQ==4), the state ist left to go directly to CLOSE.

CLOSE:

The state CLOSE can be reached from INCREASE_PWM via regular ("success") or error-transition. Here the measured value is checked against calibration-thresholds. Each transition reports via STATE_CPS. The success-transition calculates the memorized STO (for FRQ_MIN_CPS_AD) and mixes it with the old one using C_CPPWM_AD_RATIO.

The CPS is closed using a ramp until STO is reached (assuming a massflow=0 at STO), then the CPS is completely closed.

WAIT_CLOSE:

This stat is to provide a time delay (time increment) for the closing ramp.

WAIT_LAMB_RGN:

The state gives the engine some time to „recover“ (C_T_WAIT_LAMB_RGN) from the lambda-deviation caused by the adaptation, i.e. to preserve a „calm“ idle-state for the next adaptation cycle.

WRITE_CPPWM_ADD_AD_MEM:

This block defines the write-to-non-volatile-memory of CPPWM_ADD_AD_MEM.

Due to change requirements the INTERNAL value CPPWM_ADD_AD has been deleted in this chart; there rests only the output CPPWM_ADD_AD_WR which is now replacing CPPWM_ADD_AD in the internal calculations.

At NVMY_INI, i.e. when the ECU is flashed with a new software, the adaptation value CPPWM_ADD_AD_MEM is initialized with the value C_CPPWM_ADD_AD_MEM_NVMY_INI and this value is written to the non-volatile memory. By this it can be assured that a reasonable start-to-open point is used for the purging cycles before the first CPS adaptation could be done.

For correct processing, the check of CPPWM_ADD_AD_MEM has been transferred from state CHECK_CONDITIONS to chart entry.

At NVMY_INI, i.e. when the ECU is flashed with a new software, the adaptation value CPPWM_ADD_AD_MEM is initialized with the value C_CPPWM_ADD_AD_MEM_NVMY_INI and this value is written to the non-volatile memory. By this it can be assured that a reasonable start-to-open point is used for the purging cycles before the first CPS adaptation could be done.

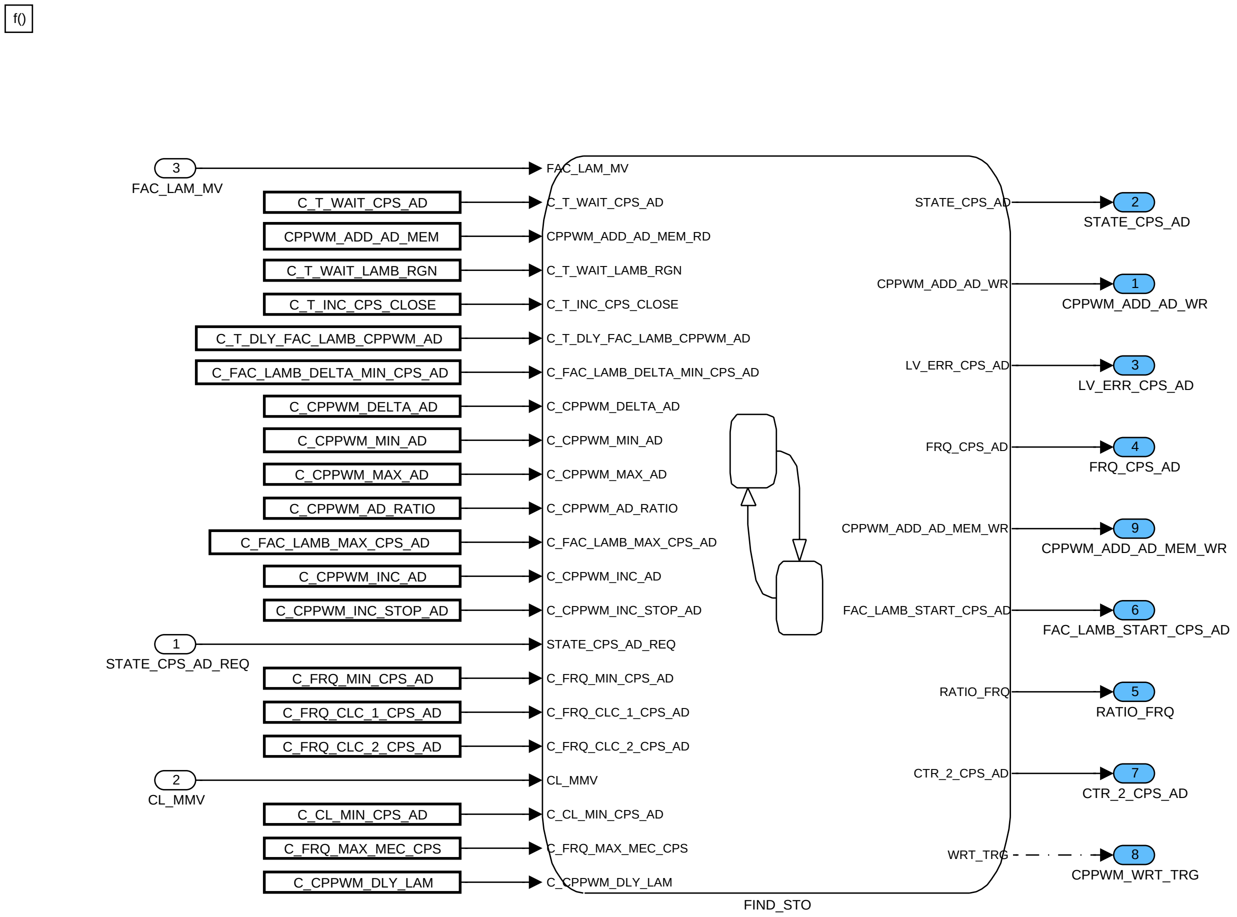


Figure 11 EVAC_CPS_AD/ LEARN_STO/ CORE_LEARN_STO

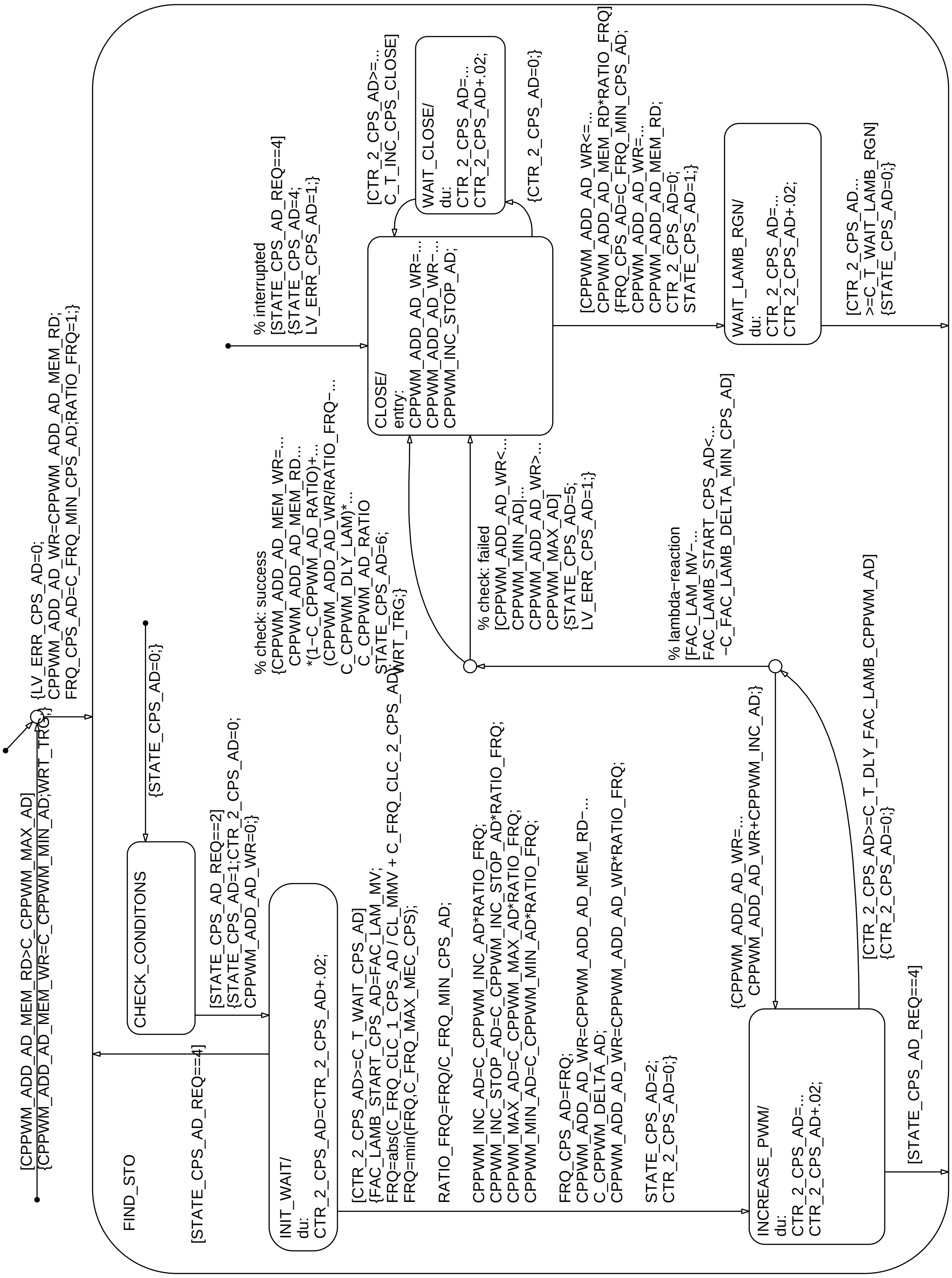


Figure 12 EVAC_CPS_AD/ LEARN_STO/ CORE_LEARN_STO/ FIND_STO

EVAC_CPS_AD/LEARN_STO/WEIGHTING_CPPWM_ADD_AD

This submodule does a weighting of the STO CPPWM_ADD_AD_WR to calculate the output CPPWM_ADD_AD. The PRS_CPS-weighting is to pay respect to a nonlinear behaviour of the CPS (to be defined by manufacturer, used for calibration of IP_CPPWM_OPEN_CPS). The FLOW- and PWM-weighting is to 'softly' eliminate the factor when reaching an operating state where STO is of almost no influence.

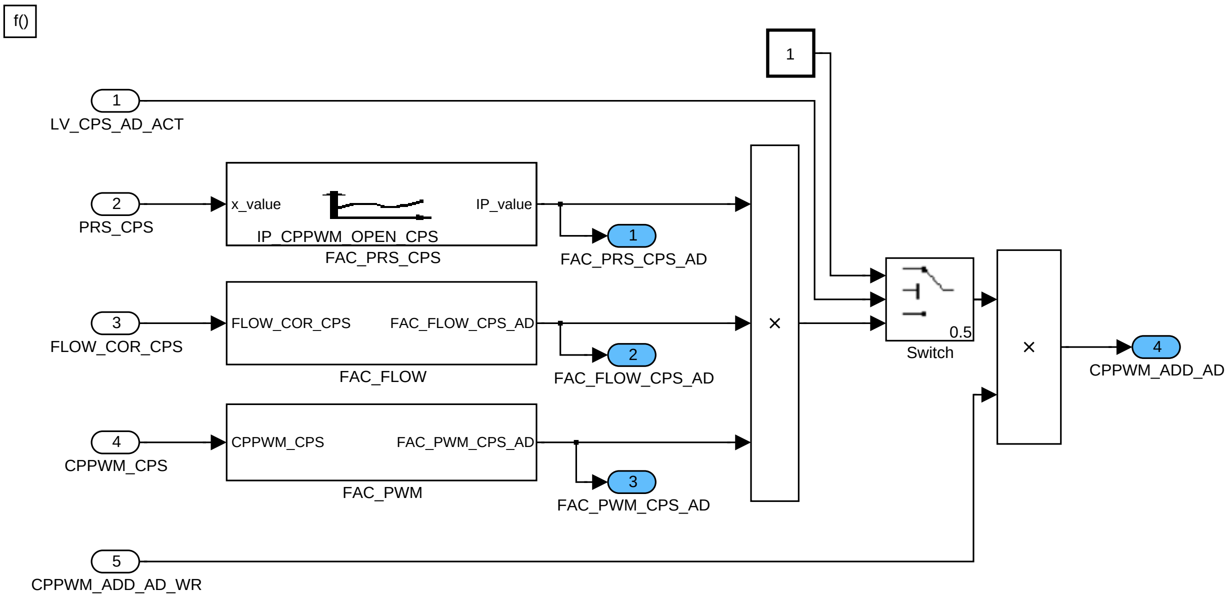


Figure 13 EVAC_CPS_AD/ LEARN_STO/ WEIGHTING_CPPWM_ADD_AD

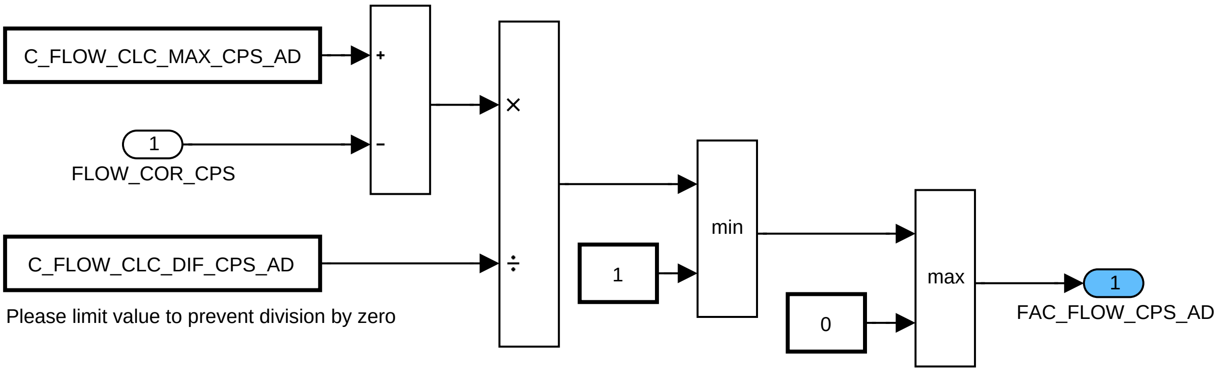


Figure 14 EVAC_CPS_AD/ LEARN_STO/ WEIGHTING_CPPWM_ADD_AD/ FAC_FLOW

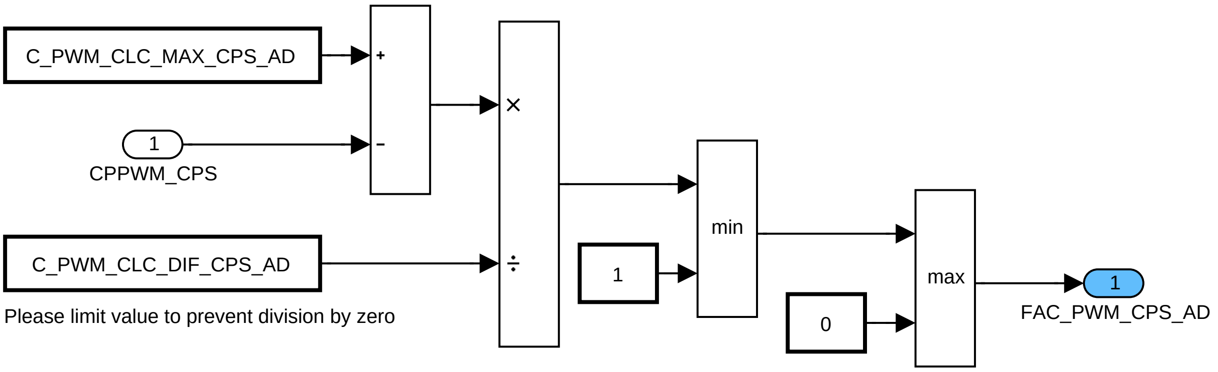


Figure 15 EVAC_CPS_AD/ LEARN_STO/ WEIGHTING_CPPWM_ADD_AD/ FAC_PWM

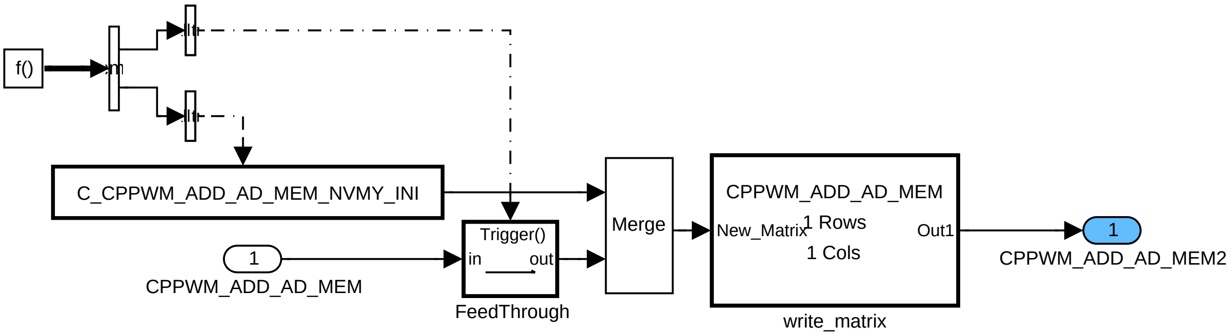


Figure 16 EVAC_CPS_AD/ LEARN_STO/ WRITE_CPPWM_ADD_AD_MEM

28.16 Canister purge valve diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_EL_CPS	V/O/S	0...1H	0...1	1	[-]
Logical value for present electrical canister purge valve failure					
ERR_SYM_EL_CPS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Symptoms of the failure					
CDN_DIAG_EL_CPS	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of CPS; bit 0: diagnosis condition for symptom SYM_0; bit 1: diagnosis condition for symptom SYM_1; bit 2: diagnosis condition for symptom SYM_2					
ERR_DIAG_EL_CPS	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Raw value of error symptom for CPS (only parameter)					
CPPWM_LIH	V/O	0...FFH	0...99.60937	0.390625	[%]
PWM signal for CPS opening limp home value					

Input data:

LV_IGK	LV_INH_DIAG_EL_CPS	CPPWM_CPS	LV_ST

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX>, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX >, IN<C_ABC_MAX_XX>, OUT< LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result
ACTION_INFR_GetElDiagCPS (OUT< CDN_DIAG_EL_CPS>, OUT< ERR_DIAG_EL_CPS >)
This action reads the failure and the conditions information for each symptom from the infrastructure

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CPPWM_DIAG_MIN_SCP	1	0...FFH	0...99.60937	0.390625	[%]
Minimum threshold for SCP diagnosis window					
C_CPPWM_DIAG_MAX_SCG	1	0...FFH	0...99.60937	0.390625	[%]
Maximum threshold for SCG diagnosis window					
C_CPPWM_DIAG_MIN_OC	1	0...FFH	0...99.60937	0.390625	[%]
Minimum threshold for OC diagnosis window					
C_ABC_INC_EL_CPS_DIAG	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_EL_CPS_DIAG	1	0...FFH	0...255	1	[-]
Maximum value for antibounce counter					
C_CPPWM_SUB_DIAG_1	1	0...FFH	0...99.60937	0.390625	[%]
cppwm substitute when SCG failure					
C_CPPWM_SUB_DIAG_2	1	0...FFH	0...99.60937	0.390625	[%]
cppwm substitute when SCB or OL failure					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_CPS	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (typical 2)					

Configuration for diagnostic symptoms :

Diagnostic CPS	Symptom description	Symptom	Filter type
CPS Diagnostic	*SCP*	*SYM_0*	*MPL_STD_INI*
	SCG	*SYM_1*	
	OC	*SYM_2*	

General information:

The CPS is driven by the ECU via an output driver. The failure detection is done by ECU Hardware.
The purpose of the diagnosis is

The purpose is to perform the electrical diagnosis of the CPS actuator to detect electrical faults as defined by OBD I requirements. Three symptoms are distinguished:

Error-symptoms and conditions: are defined to this diagnosis function as following

```
□□□□
 | | | └─> Short circuit to battery (SCP)      (= SYM_0)
 | | └───> Short circuit to ground (SCG)       (= SYM_1)
 | └─────> Open circuit (OC)                   (= SYM_2)
 └───────> Not used                            (= SYM_3)
```

Application conditions:

Initialisation: LV_ERR_EL_CPS is initialized according filter-type.
at reset: CDN_DIAG_EL_CPS = 0

Recurrence: 100ms

Activation: -

Deactivation: -

Formula section:

// Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_EL_CPS and diagnosis conditions CDN_DIAG_EL_CPS) received from the infrastructure:

ACTION_INFR_GetElDiagCPS (OUT<CDN_DIAG_EL_CPS>, OUT<ERR_DIAG_EL_CPS>, SYNCRONIZATION<CALL>)

// Diagnosis conditions are set according infrastructure information: CDN_DIAG_EL_CPS. Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_EL_CPS

```
if      LV_IGK                      = 1
and     LV_ST                       = 0
~~and     LV_CDN_VB_OBD1              = 1~~
~~and     LV_ERR_SPI_MPS              = 0~~
and     LV_INH_DIAG_EL_CPS          = 0
then    if      activation conditions are met for NC_PSD_DLY_CPS recurrencies
        then // additional diagnosis conditions
                // conditions that disable SCP detection
                if      CPPWM_CPS < C_CPPWM_DIAG_MIN_SCP
                then    bit 0 of CDN_DIAG_EL_CPS = 0        // diagnosis of SCP is not possible
                endif

                // conditions that disable SCG detection
                if      CPPWM_CPS > C_CPPWM_DIAG_MAX_SCG
                then    bit 1 of CDN_DIAG_EL_CPS = 0        // diagnosis of SCG is not possible
                endif

                // conditions that disable OC detection
                if      CPPWM_CPS < C_CPPWM_DIAG_MIN_OC
                or      CPPWM_CPS > C_CPPWM_DIAG_MAX_SCG
                then    bit 2 of CDN_DIAG_EL_CPS = 0        // diagnosis of OC is not possible
                endif
        else            CDN_DIAG_EL_CPS = 0
else                    CDN_DIAG_EL_CPS = 0
endif
```

// Failure filtering and error management treatment

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_EL_CPS and ERR_DIAG_EL_CPS; this algorithm determines:

ERR_SYM_EL_CPS	(detected error symptom for CPS diagnosis)
LV_ERR_EL_CPS	(Error flag for debounced error of CPS)
LV_CDN_DIAG_EL_CPS	(Diagnosis condition information)
LV_END_DIAG_EL_CPS	(End of diagnosis information)

ACTION_ERRM_FilterMulticondition (IN<*CPS*>, IN<CDN_DIAG_EL_CPS>, IN<ERR_DIAG_EL_CPS>, IN<C_ABC_INC_EL_CPS_DIAG>, IN<C_ABC_MAX_EL_CPS_DIAG>, OUT< LV_ERR_CPS>, SYNCRONIZATION<CALL>)

```
// Limp-home value
if      LV_ERR_EL_CPS = 1
then    case selection on ERR_SYM_EL_CPS
        SCG:        CPPWM_LIH = C_CPPWM_SUB_DIAG_1
        SCP, OC:    CPPWM_LIH = C_CPPWM_SUB_DIAG_2
        NO_SYM:     CPPWM_LIH = $CPPWM\_LIH_{n-1}$
        end case selection
```

```
else  CPPWM_LIH = 0
endif
end
```

28.17 Application Incidences for Electrical CPS Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_EL_CPS	V/O	0...1H	0...1	1	[-]
Inhibits canister purge valve diagnosis					

Input data:

LV_IGK	LV_CDN_VB_OBD1	LV_ERR_SPI_MPS	

General information:

This chapter describes the canister purge valve diagnosis inhibition.

Application conditions:

Initialisation: LV_INH_DIAG_EL_CPS = 0 at LV_IGK = 0->1 or reset

Recurrence: 100 ms & every transition of an error flag.

Activation: at every engine operating state

Deactivation: --

Formula section:

```
If      LV_IGK = 1              and
        LV_CDN_VB_OBD1 = 1      and
        LV_ERR_SPI_MPS = 0
Then    LV_INH_DIAG_EL_CPS = 0
Else    LV_INH_DIAG_EL_CPS = 1
Endif
```

28.18 Calibration hints for EVAC control (gen)

28.18.1 Calibration interfering functions

Before to start the calibration work, it is absolute mandatory to have stable operating

- Lambda-Control and Injection Prec-Control
- Intake Manifold Model
- Idle Speed Control

Deficiencies in the calibration quality of the above mentioned may block the calibration/operation or demand a recalibration. Especially the Lambda Control and the Injection Pre-Control must be at a mature standard (Lambda Control output < 5% in stationary engine conditions).

28.18.2 Main operating situations

NO_PURGE

During this mode, CPS valve is closed

This mode is used

- when conditions to authorize evap control are not fulfilled (TCO, ...)
- while the lambda adaptation phase is ongoing
- in case of failures (CPS valve, ...)
- in case of OBD2 diagnosis request (catalyst check...)
- in case of evaporative system monitoring active

MIN_PURGE

The MIN_PURGE or open loop operation operation is used when the charcoal canister has to be purged even if its load cannot be learned.

The conditions are:

- Vehicle is not equipped with A/F control
- A/F regulation component failures (lambda sensor...)
- Transient driving conditions. In this case, the purge flow is adjusted to have a maximum of 5% A/F ratio deviation with a saturated canister.

NORMAL_PURGE / closed loop operation

The normal operation is used when the determination of the canister load is finished.

Purge flow is adjusted to have the maximum of A/F ratio deviation with a saturated canister.

The NORMAL_PURGE operation is proceeded in the following sequences:

RAMP_OPEN → MAX_PURGE → RAMP_CLOSE
RAMP_OPEN_FAST → MAX_PURGE → RAMP_CLOSE

RAMP_OPEN

At first start of the NORMAL_PURGE the canister load is unknown. So, the purge valve is opened slowly using a ramp (a high deviation of the lambda control shall be prevented). During this learning the fuel injection time is corrected to avoid emission and driveability problems. The Canister Load is determined by the fuel $_{purge}$ / flow $_{purge}$ ratio.

The RAMP_OPEN operation is finished when :

- CL is learned and stable or
- The canister fuel flow (MFF_CP) is greater than a threshold

RAMP_OPEN_FAST

When the canister load is considered as known, indicated by beeing within a certatin time interval since last MAX_PURGE, the closed loop operation may start with a faster learning phase / ramp. This RAMP_OPEN_FAST operation is finished when :

- The purge flow greater than a threshold
- The canister fuel flow (MFF_CP) is greater than a threshold.

MAX_PURGE

During MAX_PURGE operation the CPS valve is controlled by the canister load, thus a constant fuel ratio (fuel from cps / total fuel) is applied and the valve opening is controlled to keep this constant fuel ratio by adjusting the total purge flow.

The target is to reach high efficiency purging. When the CL is low the purge valve opening will be increased to find the maximum limit of canister fuel flow.

RAMP_CLOSE

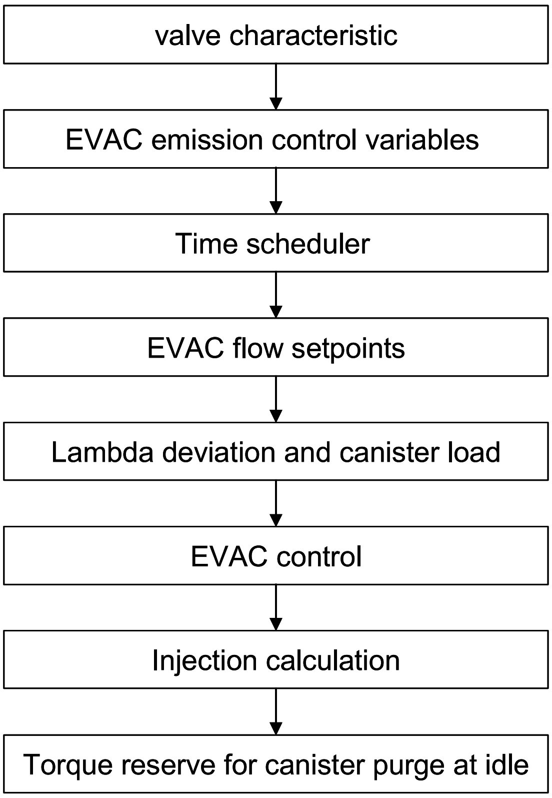
When the purge time is elapsed the purge flow is decreased via a ramp in RAMP_CLOSE operation.

28.18.3 Syntax

CalibrationDataName	description
	[defaultValue]

28.18.4 Calibration flowchart

The following flowchart shows the order to calibrate the EVAC functionalities. A continuous refining in this order is mandatory to achieve best results.



28.18.5 Valve characteristic

IP_CPPWM	CPS valve characteristic This table allows to define the valve opening (CPPWM) from two physical parameters such as the purge flow setpoint (FLOW_COR_SP_CPS) and the upstream/downstream differential pressure applied (PRS_CPS). The calibration values are gained from the flowbench measurements. An excel file IP_CPPWM.XLS allows to prepare easily the flowbench results (flow = f [pressure, valve opening]) to fill up IP_CPPWM (valve opening = f [flow, pressure]). Breakpoints recommended values for FLOW_COR_SP_CPS axe: • The breakpoint values allow to avoid to spent to much time at the beginning of Ramp Open mode (small opening). • The two first breakpoints have to following rule: • The first breakpoint will be zero (0 kg/h). The entire raw (valve opening) will be filled up with = 0 % whatever the differential pressure PRS_DIF_CPS The second breakpoint will be a small purge flow value like 0.05 kg/h. The raw (valve opening) will be filled out with the “start to open” beginning of flow depending on the differential pressure PRS_DIF_CPS. [-]
IP_CPPWM_VB_ADD_1	Additive VB correction for CPPWM at frq 1 [-]
IP_CPPWM_VB_ADD_2	Additive VB correction for CPPWM at frq 2 This additive battery correction is applied to adapt the valve opening due to battery voltage variation. The value is depending on the frequency selected.
IP_CPPWM_VB_FAC_1/2	Multiplicative VB correction for CPPWM at frequency 1 or 2. This factor correction is applied to adapt the valve opening due to battery voltage variation. The value is depending on the type of purge

	valve selected. [1]
IP_CPPWM_VB_FAC_PRS_CPS_1/2	Multiplicative VB correction for CPPWM at frequency 1 or 2. This factor corrects the "Additive VB correction for CPPWM at frq1" due to pressure CPS changes. [-]

28.18.5.1 Frequency management

C_N_FRQ_1_CPS	as required or specified
C_N_FRQ_2_CPS	
C_CPPWM_FRQ_1_CPS	
C_CPPWM_FRQ_2_CPS	

28.18.5.2 Thresholds on engine speed and CPPWM to shift CPS frequency.

C_N_FRQ_HYS_CPS **C_CPPWM_FRQ_HYS_CPS**	Hysteresis thresholds on engine speed and CPPWM to shift CPS frequency. These parameters allow to get stable frequency switching. The hysteresis value should be twice the engine speed and opening valve fluctuations. [-]
C_CPPWM_FRQ_1 **C_CPPWM_FRQ_2**	First CPS output frequency These calibrations will define the valve frequency value applied depending on engine speed and purge valve opening. The selection of the frequency allows to control the purge flow with a minimum of random variation. The intake manifold back pressure due to engine valves activities could disturb the pulsed purge flow. Then the purge fuel control becomes out of the target. [10]
IP_CPPWM_COR_FRQ	PWM correction following frequency value This factor correction is applied on the purge valve opening defined for the reference frequency control (FREQ1). From these measurements calculate the ratio between the purge valve opening at

the different frequencies and for the same conditions (purge flow, pressure). Calculate for each opening value the best ratio to satisfy most of cases.

[1]

28.18.6 EVAC emission control variables (403H)

28.18.6.1 Activation, engine speed limit

C_N_MAX_CP	Maximum engine speed value for canister purge calculations (CPU load reduction)
C_N_HYS_MAX_CP	Hysteresis value for canister purge calculations (CPU load reduction)
LC_CP_DYW_CLC_MAX_ENA	can be set to 0 if further CPU load reduction at high engine speed is needed

C_MAF_DYW_CP
C_N_DYW_CP
C_FAC_LAM_CP_DYW
C_DLY_N_MAF_DYW_CP
C_TQI_DYW_CP
C_N_2_DYW_CP
C_LAMB_SP_S_DYW_CP
C_LAMB_SP_S_DYW_CP
C_LAMB_SP_S_DYW_CP
C_LAMB_SP_S_DYW_CP
C_DLY_N_TQI_DYW_CP_1
C_DLY_N_TQI_DYW_CP_2

Dynamic window constants defining stable engine conditions for canister purge function during homogeneous combustion.

When stable engine conditions are reached, a timer is started for C_DLY_N_MAF_DYW_CP (hom.) rsp. C_DLY_N_TQI_DYW_CP (strat., hom.lean)

When the delay is elapsed the CL_MMV calculation is authorized.

The time delay allows the lambda controller to be centered again.

These calibrations have to be refined depending on the standard driving cycle used. The transient time duration will define the values.

Method:

- engine bench or Chassis Dyno
- stable engine conditions
- canister purge function not active
- Choose 6 engine load points, give preference to the stable phases of the emission cycle.
- For each engine load measure the deviation and variation(MIN and MAX) of the engine speed (N) and mass air flow value (MAF) (hom.) rsp. torque set

point value (TQI_SP_S) (lean).

- The dynamic window will be define using the formula [3 * (max-min) / 2]
- The division by 2 is the result of the specification description where the dynamic window is define as half window.

The time delay to set the stable condition information is tune on the emission cycle in order to filter the lambda deviations during transients. The condition engine stable must be active when lambda deviations (without purge) are elapsed.

C_CRLC_MAF_DYW_CP	Filter constant for limited dynamics calculation on MAF [0,1]
C_CRLC_N_DYW_CP	Filter constant for limited dynamics calculation on engine speed [0,1]
C_CRLC_LAM_DYW_CP	Filter constant for limited dynamics calculation on lambda controller controller in non-normal purge [0,1]
C_CRLC_LAMB_SP_S_DYW_CP	Filter constant for limited dynamics calculation on lambda setpoint in stratified [0,1]
C_CRLC_TQI_DYW_CP_1 **C_CRLC_TQI_DYW_CP_2**	Filter constant for limited dynamics calculation on tqi_sp in stratified, _2 for RAMP_OPEN_FAST and MAX_PURGE, _1 for all other states. [0,1]
C_CRLC_LSMB_SP_S_DYW_CP	Filter constant for limited dynamics calculation on lambda setpoint in stratified [0,1]
IP_TQI_CP_BOL	Minimum torque limit for activation of the RAMP_OPEN operation (stratified) [-]
IP_TQI_CP_TOL	Maximum torque limit for activation of the RAMP_OPEN operation (stratified [-]
C_MAF_CP_BOL_HYS	Hysteresis for minimum mass air flow limit for activation of the RAMP_OPEN

	operation (homogeneous) [0]
C_TQI_CP_BOL_HYS	Hysteresis for minimum torque limit for activation of the RAMP_OPEN operation (stratified) [0]
LC_CP_DYW_CLC_ENA	1: calculation of dynamic window conditions 0: LC_CP_DYW allways one [1]
LC_CP_DYW_CLC_MAX_ENA	1: calculate lim.dynamics / lambda contr.limits etc. also with n_max_cp exceeded [1]
LC_CP_DYW_LAM_AD	1: use the limited dynamics flag of the lambda adaptation (common flag lamb ad / cp) 0: use the CP internal calculation of the limited dynamics falg [0]
LC_LAM_DYW_CP_INI_DEAC	1: deactivate dynamic window on lambda control at deactivated normal purge [0]
C_T_DLY_CAT_PURGE_CP	Stabilisation time after catalyst regeneration

28.18.6.2 Preselection of the canister purge state

C_STATE_CDN_CP_MAN	manual selection of STATE_CDN_CP
C_PRS_MAX_CPS	[Barometric pressure – Manifold intake pressure] limit of the difference of the pressure applied for activation of the canister purge One condition of NO_PURGE is AMP – MAP < C_PRS_MAX_CPS. The value C_PRS_MAX_CPS = 0 enables the MAX_PURGE mode when MAP > AMP (case of supercharger engine) [0 mb]
C_PRS_MAX_CPS_2	To prevent backflow from intake manifold in the acf with charged engines, this threshold leads to a direct shut of the canister purge valve (no ramp) [AMP – MAP] < C_PRS_MAX_2_CPS

	disables close loop operation [10 mb]
C_STATE_WIN_CP_NOT	state to be forced if engine operation outside CP window; selection of "NONE" leads to an exiting of RAMP_OPEN, but not of RAMP_OPEN_FAST or MAX_PURGE
C_STATE_LAM_LIM_CP	state to be forced if lambda control limitation reached .
LC_STATE_CDN_CP_MAN	switch to activate test mode // manual setting of STATE_CDN_CP
C_TIA_MIN_CP	Minimum intake air temperature to enable canister purge function at idle. One of the condition of NO_PURGE is TIA < C_TIA_MIN_CP. Freezing conditions or acoustic problems can appear at idle. This value has to be checked during cold condition tests. The freezing conditions are depending on the location of the valve mounting. The acoustic problems are depending on the valve mounting support. [5° C]
C_LAMB_SP_S **C_LAMB_SP_S_CP_TOL** **C_LAMB_SP_S_CP_BOL**	Bottom and top limits on the LAMB_SP_S value [1,5] [2,5]
C_FAC_LAM_THD	Upper threshold for lamgda control output gradient If the lambda gradient exceeds the threshold conditions for CL_MDL or MinP are fulfilled. [-]
LC_FAC_LAM_GRD	Switch for enabling lambda control output gradient check [0]
LC_CP_INH_MAN	Switch for deactivation of canister purge and force NO_PURGE [0]
LC_PRS_CPS_EXT	switch to use externally calculated value for PRS_CPS [0]
LC_CP_CLL_INH_MAN	switch for deactivation of closed loop canister purge ("normal purge")

	[0]
C_CP_CLL_S_INH	Selection of MIN_PURGE during stratified mode (=1) [0]
LC_CP_CAT_PURGE_OPL **LC_CP_CAT_PURGE_OFF**	Defines the behaviour of the system during NOx regeneration of the catalyst and a time delay after that. The canister purge can be set configurable to off (LC_CP_CAT_PURGE_OFF = 1, OPL =0), open loop (LC_CP_CAT_PURGE_OPL = 1, OFF = 0) or closed loop (both = 0).

28.18.7 Time scheduler (903P)

C_STATE_MAN_SDL_CP	selection of manual scheduling mode [AUTOMATIC]
NC_NR_CP_LAM_AD **ID_T_SET_T_AST_LAM_CP** **LDP_T_AST_SET_ID_T_AST**	Set NC_NR_CP_LAM_AD equal to the total sum of fixed time slices (lambda adaptation and caniser purge) plus one for the dynamic time slices. Fill in LDP_T_AST_SET_ID_T_AST the start time of the fixed time slices and the last value for the dynamic time slices. The first value has to be zero. In the ID_T_SET_ID_T_AST the corresponding fixed (CP_FIXED or LAM_FIXED) states are filled in. Fill the last element with DYN_SDL to end the fixed time slices.
C_CL_MMV_T_BOL **C_CL_MMV_T_TOL** **C_T_DI_CP**	When an empty canister is detected at the end of the learning modes RAMP_OPEN or RAMP_OPEN_FAST (empty canister meaning CL_MMV < C_CL_MMV_T_BOL), (and C_STATE_MOD_SEL_CP="MAX_PURGE") the evaporative system control is stopped for the delay C_T_DI_CP. If the canister is detected full CL_MMV>C_CL_MMV_T_TOL the time control gives the priority to the CP time request. In case of middle load the time control check the request of lambda adaptation and canister purge. This value is checked during hot trip (worst case conditions: canister full and vapor generation) and canister purge cycle.

The goal is to optimize the lambda adaptation accuracy and the purge efficiency (no fuel vapor escape to the atmosphere).

C_T_MFF_AD_MAX

Initialisation time for dynamic lambda adaptation time slice. The timer counts as well while waiting for the lambda adaptation conditions to be met (in this period, canister purge may still be enabled). The lambda adaptation phase may be left if the feedback flag from lambda adaptation is set.

[240 s]

C_T_2_CP

Initialisation time for dynamic canister purge time slice, will be corrected at the end of RAMP_OPEN.

This value must be checked on emission cycle.

[240 s]

IP_T_MAX_CP

Dynamic canister purge time depending on canister load. At the end of RAMP_OPEN or RAMP_OPEN_FAST the purge time duration is modified to fit the canister load learnt.

IP_T_MAX_CP < C_T_2_CP.

This calibration is defined by measuring CL decrease on standard driving cycle (FTP, EURO 2000...) with worst case condition like ambient temperature... For different CL adjust the purging time duration to avoid canister purge breakthrough.

CL_MMV	[-]	0	0	0	0	1	1
IP_T_MAX_CP	[s]	60	90	200	230	260	280

C_CL_MMV_T_WAIT_LAM

CL threshold to switch off CP during WAIT_LAM if CL_MMV is below the threshold

[0,6]

C_TCO_COLD_DYN_T_CTL_CP

C_TCO_DYN_T_CTL

C_T_TCO_ST_STAT_SDL_CP

If the coolant temperature is below the first threshold at the start of a driving cycle, dynamic time slices are chosen. Between both thresholds, the fixed time slices are applied. Above the second threshold, one single fixed time slice for purging is chosen. The time for the fixed slice is given by C_T_TCO_ST_STAT_SDL_CP.

LC_CL_CP_DYN_END_ENA

Switch to enable end of CP phase when reaching low canister load (CP_DYN ->

	WAIT_LAM) [0]
LC_LAM_LIM_AD_ENA	Switch to enable forced lambda adaptation during canister purge [1]
LC_MFF_AD_ENA_PAS	Switch to enable lambda adaptation while cp passive [1]
LC_SDL_LAM_AD_CDN_OFF	Switch to ignore LV_LAM_AD_CDN (adapt. ongoing) for end of lambda adapt. time slice calculation; lambda adapt. will be interrupted if timer exceeds. [0]
LC_CYL_BAL_LAM_AD_INH_CP	Switch to deactivate the cylinder balancing manager influence on enabling lambda adaptation

28.18.8 EVAC flow setpoints (9076)

IP_FLOW_CPS_HOM

IP_FLOW_CPS_HOM_AFL

These values are the maximum canister purge flow in MIN_PURGE mode and homogeneous lambda 1 rsp. hom. lean combustion.

This ratio will limit the purge valve opening because no fuel compensation is done in this mode.

ΔLambda CP is the maximum lambda control deviation allowed to avoid driveability trouble (acceleration) when the lambda controller is not centered. This maximum deviation is obtained with charcoal canister full loaded. The recommended value of ΔLambda CP is about 5-6 % maximum.

From this maximum deviation the maximum purge flow can be evaluated for each engine load :

- Canister Load = (Fuel Mass CP) / Air Flow CP
- Air Flow CP = (Air Flow cylinder * ΔLambda_control CP) / $K_{stochiometric}$ * CLmax

We know the maximum Canister Load ≈ 2

and the usual $K_{stochiometric}$=14,57

- Air Flow CP= Air Flow cylinder * 0,00206

Nevertheless, we have to consider that for engine conditions like “High Load” and charcoal full loaded canister, the upstream/downstream differential pressure will be too low to generate sufficient purge flow then the lambda controller deviation will be also too low.

Method:

For each engine condition (N,MAF):

- Consider the maximum flow the purge valve can provide: IP_FLOW_CPS
- **if** IP_FLOW_CPS__PRS_CPS ≥ Air Flow cylinder * 0.00206

  then fill up IP_FLOW_CPS_HOM with Air $Flow_{cylinder}$ * 0.00206

  else fill up IP_FLOW_CPS_HOM with IP_FLOW_CPS

  endif

IP_FLOW_CPS_S

flow setpoint for MIN_PURGE / stratified operation

Method:

For stratified engine condition (N,MAF):

- Engine bench
- Bottle of gas (butane)
- Perform different opening of the valve with a canister full loaded and measure the impact on the torque.
- The calibration value will be the one in the limit of the driveability (ex 5Nm additional torque).
- The map is tuned using the canister purge valve flow FLOW_CPS

Only available in DI version

IP_RATIO_FUEL_MAX_CP_HOM

IP_RATIO_FUEL_MAX_CP_HOM_2

Maximum fuel flow ratio (fuel flow through the CPS/ total fuel flow) for homogeneous stoiciometric operation

This map allows to adapt the purge valve opening in order to purge with the maximum efficiency when the CL is not at the maximum value. These calibrations are used in RAMP_OPEN, RAMP_OPEN_FAST and MAX_PURGE modes.

These calibrations define the maximum purge fuel quantity regarding the fuel engine needs.

To avoid driveability troubles the current ratio limitation between purge fuel and total engine fuel is 25%.

- Fuel purge flow = In system air flow / K $_{\text{stochiometric}}$ * X% **{1}**
- Fuel purge = Total cylinder flow * FLOW_CPS * CL $_{\text{max}}$ **{2}**
- X% is the fuel purge/engine total fuel ratio (given by engine bench)
- FLOW_CPS is defined by IP_FLOW_MAX_CPS
- Cl_{max} = 0.7 (maximum value with butane)

Method:

- Calculate for each engine load defined by [MAF, N] the values **{1} and {2}**
- Final calibration value for each engine load defined by [MAF, N] = MIN [**{1}, {2}**]
- X%: start with 20% and following the hot trip or SHED results the percentage could be increase (less than 35%).

IP_RATIO_FUEL_MAX_CP_HOM_AFL

Only available in DI version

Maximum fuel flow ratio (fuel flow through the CPS/ total fuel flow) for homogeneous lean operation

These calibrations allow to adapt the purge valve opening in order to purge with the maximum efficiency when the CL is not at the maximum value. These calibrations are used in RAMP_OPEN, RAMP_OPEN_FAST and MAX_PURGE modes.

These calibrations define the maximum purge fuel quantity regarding the fuel engine needs.

To avoid driveability troubles the current ratio limitation between purge fuel and total engine fuel is 25%.

- Fuel purge flow = In system air flow / K $_{\text{stochiometric}}$ *lambda* X% **{1}**
- Fuel purge = Total cylinder flow * FLOW_CPS * CL $_{\text{max}}$ **{2}**

- X% is the fuel purge/engine total fuel ratio
- FLOW_CPS is defined by IP_ FLOW_MAX_CPS
- Cl$_{\text{max}}$ = 0.7 (maximum value with butane)

Method:

- Calculate for each engine load defined by [MAF, N] the values **{1} and {2}**
- Final calibration value for each engine load defined by [MAF, N] = MIN [**{1}, {2}**]
- X%: start with 10% and following the hot trip or SHED results the percentage could be increased.
- If the lean burn combustion is not used the map is tuned to 0.

IP_RATIO_FUEL_MAX_CP_S

Only available in DI version

Maximum fuel flow ratio (fuel flow through the CPS/ total fuel flow) for homogeneous stratified operation

These calibrations allow to adapt the purge valve opening in order to purge with the maximum efficiency when the CL is not at the maximum value. These calibrations are used RAMP_OPEN, RAMP_OPEN_FAST

	and MAX_PURGE modes. These calibrations define the maximum purge fuel quantity regarding the fuel engine needs. To avoid driveability troubles the ratio is limited to take into account the torque. X%: start with 10% and following the hot trip or SHED results the percentage could be increase. If the stratified MAX_PURGE is not used the map is tuned to 0. [0,25]
IP_RATIO_FUEL_MAX_CP_COR	This table is corrective factor acting on MFF_KGH_MAX_CP calculation. The aim for tuning is to allow activation of the purge function at a lower TCO. In this case (that means at low TCO), IP_RATIO_FUEL_MAX_CP_COR allow to less open the CPS. This correction has been added in case of long warm up phase, to win purge efficiency by opening the CPS in advance.
C_CL_MAX_MIN_PURGE	on this worst case CL is based the base calibration of MIN_PURGE flow setpoint for intelligent MinPurge [1]
C_FAC_FLOW_SP_MIN_PURGE_MAX	maximum correction of MIN_PURGE flow setpoint for intelligent MinPurge [1]
IP_FAC_FLOW_SP_MIN_PURGE	Weighting factor for the CL deviation from the worstcase CL The flow setpoint can be corrected depending on the last learned CL_MMV value. [-]
IP_FLOW_CPS	FLOW_CPS for fully opened CPS (CPPWM=100%) Maximum flow at full opend depending on the preasure at the CPS. [-]
IP_FLOW_MAX_CPS	Flow setpoint for MAX_PURGE
IP_FLOW_MAX_CPS_2	These calibrations define the maximum purge flow regarding the air engine needs.

Calibrate together with IP_RATO_FUEL_MAX_CP_HOM

[-]

LC_FAC_FLOW_SP_MIN_PURGE

Switch to activate / deactiate "intelligent MIN_PURGE"

[0]

C_MFF_MIN_CP

Minimum injector fuel mass.

28.18.9 Lambda deviation and canister load (9077)

28.18.9.1 Filter constants

IP_CRLC_FAC_LAM_MV_MMV_CP_CLL

correlation constant for FAC_LAM_MV_MMV_CP calculation in case of Purge in closed loop

MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_CRLC_FAC_LAM_MV_MMV_CP_CLL	[-]	0,500	0,500	0,500	0,500	0,500	0,500

IP_CRLC_FAC_LAM_MV_MMV_CP_OPL

correlation constant for FAC_LAM_MV_MMV_CP calculation in case of Purge in open loop

MAF_CYL	[kg/h]	12	45	70	140	250	50
IP_CRLC_FAC_LAM_MV_MMV_CP_OPL	[-]	0,250	0,250	0,250	0,250	0,250	0,2

IP_CRLC_LAMB_LPF_CP

Correlation constant for LAMB_MMV_CP calculation

IP_CRLC_MMV_CP

Correlation constant for LAMB_MMV_CP calculation

IP_CL_CRLC_RAMP_OPEN

CL_MMV filter coefficient during RAMP_OPEN_FAST

MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_CL_CRLC_RAMP_OPEN	[-]	0,300	0,300	0,400	0,450	0,480	0,500

IP_CL_CRLC_OPEN_FAST

When the RAMP_OPEN_FAST is ongoing the Lambda controller has to be stable (no more than 5% of deviation)

This calibration result is also depending on ramp opening value during this mode.

IP_CL_CRLC_INI

Initialization value for correlation factor value

CL_MMV	[-]	0	0	0
IP_CL_CRLC_INI	[-]	0,150	0,150	0,150

IP_CL_CRLC

This calibration is the CL_MMV filter coefficient during MAX_PURGE mode. The values will define this filter coefficient will be limited by two borders. The high one is defined by the Canister load decrease in stable engine conditions, and the low one is defined by the normal acceleration/deceleration time duration

case A, canister load decreases in stable engine conditions

Method:

- IP_CL_CRLC has to be tuned to follow the canister load (CL) decrease.
- During MAX_PURGE, CL decreases and the canister weight loss follow the same law.
- A wrong IP_CL_CRLC value could result in a meaningful difference between the average CL value (CL_MMV is used for the calculation of injection compensation) and CL.

Test conditions:

- Different engine loads in stable conditions
- Charcoal canister full loaded (butane or fuel following legal procedure).
- C_T_1_CP, C_T_2_CP and IP_T_MAX_CP_CL_MMV set to a high value to keep MAX_PURGE mode until the canister becomes empty.
- IP_CL_CRLC pre-tuned to 0,005 (for the whole mapping).
- recording of CL
- if canister is loaded with butane: CL_MAX ≈ 0.63
- if canister is loaded with fuel: CL_MAX ≈ 0.7....2

case B, normal acceleration/deceleration time duration

During transient mode the Lambda controller varies. The time duration of the lambda controller is shorter than the one during the charcoal canister weight lost. This lambda controller variation has not to be taken such as a loss of canister load.

- Different engine load in stable conditions
- Purge deactivated
- start with normal acceleration/deceleration
- Measure the time spent (T2) by the lambda controller to reach a stable value {2}

Calculate the values (CL max * recurrence) / T2 s {2}

The final calibration value will be: {2} < IP_CL_CRLC < {1a, 1c}

Usual approach:

Part and full load

IF for MAF_CYL ≥ 90 kg/h the value calculated in {1a} > 0.005 then fill up the mapping table with 0,005

And for MAF_CYL < 90 kg/h fill up the mapping table with 0.005 * value calculated in {1b}

Idle speed

At idle the average value of the lambda controller (TI_LAM_MMV) varies. Then this variation generate another CL variation. For MAF_CYL breakpoint at idle and charcoal canister empty refine the coefficient filter to prevent these CL, CL_MMV oscillations.

IP_NR_STEP_CL

Number of step to change of correlation factor value

This is a filter-filter-constant map, it aims in filtering the transition from the IP_CL_CRLC_INI map to the standard-MAX_PURGE-map IP_CL_CRLC.

C_CL_FAC_LAM_DIF_MIN
C_CL_FAC_LAM_DIF_MAX

Lower and upper limits on FAC_LAM_DIF_CP variation

If these limits are exceeded in MAX_PURGE, the CL filter constant is taken from the faster map

	IP_CL_CRLC_INI, same principle as entering MAX_PURGE.
LC_CL_CLC_INH_FAC_LAM_DIF	Switch to freeze CL calculation in case of FAC_LAM_DIF_CP exceeding the above mentioned thresholds. The CL calculation is frozen until the CL filter constant reaches again its final value to prevent interdependencies between CL and CL_MMV calculation. This functionality should be used if oscillations occur (e.g. due to slow lambda controller) when using the faster CL filter constant.
C_PRS_MAX_2_CPS/_1	Pressure difference limit (environment-intake manifold) for CL_MMV calculation depending on STATE_OPM_ENG_CP [10 mb]
C_CRLC_FAC_LAM_GRD	Correlation constant for lambda gradient For calculation of the value FAC_LAM_GRD_MMV [-]
C_VS_MIN_LAMB_AFS	The value has to be set to a vehicle speed to detect highway driving.
C_T_LAMB_AFS_T_AST	Time between two highway driving phases to keep the homogeneous request.
C_T_LAMB_AFS_T_AST_VLD_CP	Maximum valid time for homogeneous combustion mode request in canister purge open loop calculation. The time has to be bigger then C_T_LAMB_AFS_T_AST.
C_T_LAMB_AFS_AMP_VLD_CP	Maximum valid time for homogenious combustion mode request in canister purge open loop calculation.

28.18.9.2 CL calculation

C_FLOW_MIN_CPS	Purge flow threshold to authorize CL_MMV calculation in RAMP_OPEN, RAMP_OPEN_FAST and MAX purge. if FLOW_CPS is lower than this calibration the last value of CL_MMV is kept. During the purge valve opening/closing, the flow FLOW_CPS takes small values. Small flow values can produce an incorrect purge flow and an incorrect canister load learning. This difference results in values of CL that do not correspond to a physical

reality.

Calculate the purge flow obtained by 1 bit of valve opening.

Method: Biggest purge flow (100% opening P). Divide this maximum flow by the format of the valve opening□and high (CPPWM) in case of byte the divisor will be 255. The calibration will be the value regarding 2 bits.

[0,04 kg/h]

C_CL_ST_CP

Initialization value for charcoal canister load at the beginning of RAM_OPEN mode.

CL_MMV and CL use the same initialization value

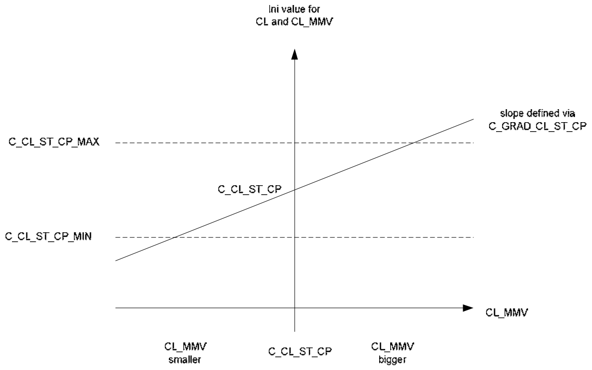
The calibrated value is about the half value of the maximum CL value encountered when the charcoal canister is loaded with butane. This value allows to reduce the work of the CL filter and to get too much deviation between CL and CL_MMV.

[0,3]

C_CL_ST_MIN_CP
C_CL_ST_MAX_CP
C_CL_ST_FAC_CP

At start of RO at stratified engine the initialization of CL and CL_MMV can be influenced with the last known CL_MMV in the limits of C_CL_ST_MIN_CP and C_CL_ST_MAX_CP. The influence of the last known canister load can be weighted with C_CL_ST_FAC_CP.

The values have to be calibrated in the way that at maximum deviation of real canister load to estimated canister load CL_MMV does not cause drivability influences.



C_CL_MAX_CP

The canister load is the fuel concentration of the A/F mixture that comes from the canister through the CPS valve.

	The mass of fuel (vapors) which is coming from the canister depends on: • the type of fuel used • the system design (pressure drop...) Charcoal canister loaded with butane • The measurements should lead to the theoretical value CL max = 0,7 Charcoal canister loaded with gasoline • The fuel from the canister is in the vapor state but the mass fuel flow is higher than the butane. This mass is depending on the type of fuel used and the in-use conditions. The C_CL_MAX_CP value equal to 2 could satisfied most of applications. Check this value during refueling in hot trip conditions (high vapor generation that can lead to a full and unstable canister load)
C_FAC_LAM_DIF_MAX_CP	Maximum Lambda controller deviation threshold during learning mode. This calibration is defined by the maximum lambda controller deviation to get no driveability trouble during accelerations while the RAMP_OPEN mode is ongoing. [5%]
C_FAC_LAM_DIF_CP_AS	Application system value to apply a lambda deviation
LC_LAM_CBK_CPS	Lambda controller selection (0 -> bank 0 , 1 -> bank1)
C_RAF_CLC_CP	Stochiometric ratio of usual fuel. This value has to be refined regarding the type of fuel defined by the application. [14,57]
LC_CL_FIL_INI_TRA_DEAC_CLL	Correlation factor calculation in case of NO_PURGE or MIN_PURGE → MAX_PURGE transition
C_CL_DYW_MAX_CP	This calibration aims to prevent wrong fuel correction when too fast variation of CL happens. This calibration is to get fuel correction robustness. In case where too high CL variation happen the MIN_MODE is forced. In RAMP_OPEN, RAMP_OPEN_FAST,

	MAX_PURGE modes [CL n-1 – CLn] is calculated. *if [CL n-1 – CLn] > C_CL_DYW_MAX_CP* *then MIN_MODE is forced*
C_CPPWM_MIN_CL	Minimum valve pwm signal to enable CL calcualation [-]
C_DEC_FAC_LAM_RAMP_OPEN	Decrement for FAC_LAM-ramp at end of RAMP_OPEN Fade out of the CRLC_FAC_LAM_CP for the FAC_LAM_CP calculation after RampOpen [-]
C_FAC_LAM_RAMP_OPEN	Weighting factor of FAC_LAM_LIM for CL calculation except at end of RAMP_OPEN
C_FAC_LAM_THD_CL_MEM	Threshold for FAC_LAM_GRD to save the last CL_MMV value
C_FLOW_MIN_CPS_RO	Limit on FLOW_CPS value for CL_MMV calculation (during RO and ROF) Avoid wrong CL calc. at low CPS flow. [-]
C_T_DLY_CP_INJ_LS_OFS	CL calculation will be enabled C_T_ ms before releasing the next lambda shift See description in the specification. [-]
IP_FAC_CPPWM	Reduction Factor at very low flow rates, dependent on CPPWM [-]
IP_FAC_CRLC_CL	Reduction Factor at very low flow rates Correction for CRLC_CL. [-]
IP_FAC_LAM_CP_LS_COR	Correction factor for FAC_LAM_CP_LS calculation [-]
IP_FAC_LAM_GRD_MMV	Reduction Factor dependent on mean value of gradient of lamda control output [-]
IP_T_DLY_CP_INJ_LS	Time delay injector -> lambda sensor [-]

LC_CL_FIL_INI_TRA_DEAC_CLL	Correlation factor calculation in case of NO_PURGE or MIN_PURGE -> MAX_PURGE transition Initialisation of CRLC_CL at transiton allowed [0]
LC_CL_FIL_INTR	switch to continue with CRLC_CL_TMP_0 initialization on entering MAX_PURGE Initialisation of CRLC_CL at transiton [0]
LC_FAC_CRLC_CL_ENA_RAMP_OPEN	switch to enable the usage of correction factor FAC_CRLC_CL in RAMP_OPEN [1]
LC_FAC_LAM_CP_DEAC	switch to deactivate the "classical" lambda control influence during RAMP_OPEN [0]
LC_FAC_LAM_CP_LS_ENA	switch for permanent enabling of the lambda deviation part for CP system deviation calculation [0]
LC_INIT_CL_CRLC_RAMP_OPEN_FAST	Correlation factor calculation in case of RAMP_OPEN_FAST -> MAX_PURGE transition [0]
LC_LAMB_DE_ENA	Switch to enable the usage of the direct lambda deviation influence [0]
LC_VVL_THR_MOD_REQ_CP	switch to deactivate "Calculations for throttled VVL mode request" [0]
C_FAC_LAM_0_MIN/MAX	Minimum/ maximun threshold for FAC_LAM_0_CP in homogenious engine mode to avoid wrong CL calculation due to short term lambda deviations.
C_FAC_LAM_0_MIN_WIDE **C_FAC_LAM_0_MAX_WIDE**	Mininmum and maximum FAC_LAM_0_CP limits in case errors in the fuel system were detected

28.18.9.2.1 DI specials

C_CL_MMV_CHK	The canister purge force the homogeneous lambda 1 combustion to increase the capacity of purge, when the canister load is under the threshold C_CL_MMV_CHK

	the stratified combustion is active again. *Method:* On specific emission cycle (Ex : MAZDA), check if the cycle if finished During hot trip and fuel vapor generation, check the capacity of the function to prevent gas to the vent of the canister.
C_CL_MMV_HOM_CP	The canister purge force the homogeneous lambda 1 combustion to increase the capacity of purge, when the canister load is upper the threshold C_CL_MMV_HOM_CP the homogeneous lambda 1 combustion stay active. *Method:* On specific emission cycle (Ex : MAZDA), check if the cycle if finished During hot trip and fuel vapor generation, check the capacity of the function to prevent gas to the vent of the canister.
LC_LAMB_AFS_CP_CPS_SCG	switch to force hom1 operation if cps valve short circuit to ground error present
C_ABC_INC_LAMB_AFS_AMP_CP **C_ABC_DEC_LAMB_AFS_AMP_CP** **C_LAMB_AFS_AMP_MAX_CP**	Incrementation/ decrementation steps for anti bounce counter and max value for counter. Depending on the ambient prasure difference at start and current ambient preasure the counter is increased or decreased. By reaching the positive threshold a homogenious engine mode is requested, by reaching the negative threshold the request is reset. To avoid canister break through at big ambient prasure change.
IP_CL_MMV_HOM_PRS_UP/DOWN_CP	Threshold for ambient preassure for homogeinious request.
C_ABC_INC_LAMB_AFS_T_AST_CP **C_ABC_DEC_LAMB_AFS_T_AST_CP** **C_LAMB_AFS_T_AST_MAX_CP**	Incrementation/ decrementation steps for anti bounce counter and max value for counter. Depending on time after start the counter is increased or decreased. By reaching the positive threshold a homogenious engine mode is requested, by reaching the negative threshold the request is reset. To avoid canister break through at city driving.
IP_CL_MMV_HOM_T_AST_UP/DOWN_CP	Threshold for time after start for homogeinious request.
C_CRLC_LAMB_MMV_0_GRD_CP	Correlation constant for learning of the

	straight line for FAC_LAM_0_CP correction due to lambda setpoint changes.
C_LAMB_LPF_AD_MIN/MAX_CP	Lambda setpoint region where an adaption of the staight line for FAC_LAM_0_CP is possible.
C_LAMB_LPF_CLC_GRD_MIN_CP	Treshold for low pass flter Lambda for canister purge to activate adaption of the correction for FAC_LAM_0_CP
IP_N_32_OPP_MIN/MAX **IP_N_32_OPP_MIN/MAX_1** **IP_MFF_SP_OPP_MIN/MAX** **IP_MFF_SP_OPP_MIN/MAX_1**	Threshold depending on STATE_OPM_ENG_CP around the engine operating point (N and MFF_SP) at start of a purge cycle. At leaving of the frame a new RO is requested. To avoid CL calculation errors due to mismatch in lambda setpoint and measured signal.
C_FAC_OPP_CP_1	Percentage to define a smaller window to calculate CL via CL model.
LC_FAC_LAM_0_COR_ACT_CP	Activation of the correction of FAC_LAM_0_CP due to lambda setpoint changes. This can be deactivated if the operation point windows are very strict.
C_N_32_MIN_OPP_WIN_CP	N threshold below the usage of CL model for operation window is not used
C_MFF_SP_MIN_OPP_WIN_CP	MFF threshold below the usage of CL model for operation window is not used

28.18.9.3 Timer constants for limits exceeded tests

C_T_DLY_PU_CPS	CPS interrupt time during PU When the engine management goes in trailling throttle (LV_PU) the injection time becomes very small and near the minimum injection time. To prevent a big lambda deviation on transition PL to PU the CPS valve is closed immediately during a time C_T_DLY_PU_CPS. When this time is elapsed the function stay in MIN_PURGE during PU. The time must allow to close the valve during the gas transfer between the CPS valve and the oxygen sensor. This time is defined by the duration of the engine gas transfer. [0.1 s]
C_T_DLY_PUC_CP	CPS interruption time after PUC When the engine management goes in

	trailling throttle with fuel cut off (LV_PUC) the CPS valve is closed immediately. After this phase, the canister purge will stay in NO_PURGE for the time C_T_DLY_PUC_CP to prevent that the CL is influenced by lambda deviations that are due to transitional effects from the PUC phase. This time should be determined by forcing PUC phases with deactivated canister purge. C_T_DLY_PUC_CP should be at least the time that is needed for lambda controller stabilization after a PUC phase.
IP_T_DLY_CMU_CPS	Stabilization delay on lambda regulation after commutation When the engine have combustion commutations (e.g. homogeneous → stratified), all the parameters are unstable during a delay. To prevent driveability problems during the commutation the canister purge valve is closed during a time IP_T_DLY_CMU_CPS__N_32. The function is re activated as soon as the engine parameters are stable. Method: Perform different type of commutations and measure the time to have stable values
C_T_LAM_LIM_CP	Additional close loop deactivation duration due to LV_LAM_LIM activation. Set, when the lambda controller sets its flags indicating "lambda control limitation reached" [5 s]
C_FAC_LAM_CP_BOL	Lambda control functional window.
C_FAC_LAM_CP_TOL	This window is additional to the control of the limiation flags from the lambda controller, see above. To prevent wrong canister load learning, the lambda control value will be checked all the time in RAMP_OPEN. If C_FAC_LAM_CP_BOL < FAC_LAM_CP < C_FAC_LAM_CP_TOL, RAMP_OPEN is disabled or stopped temporarily and MIN_PURGE is applied. This calibration prevents the learning because the reference point FAC_LAM_0_CP is not centered at RAMP_OPEN start.

	The threshold is defined by the normal error due to the flows (air & fuel). The realistic error is estimated about 5%. The 5% standard lambda controller deviation during the learning mode is defined by C_FAC_LAM_DIF_MAX_CP. These calibrations are defined by: - standard error flow + C_FAC_LAM_DIF_MAX_CP [-10%]
C_T_DLY_LAM_CP	Time to inhibit learning modes RAMP_OPEN & RAMP_OPEN_FAST after lambda controller left functional window defined above. After the time delay C_T_DLY_LAM_CP the learning conditions will be checked once more. [20 s]
C_N_CP_BOL	Minimum speed limit for activation of the RAMP_OPEN operation (homogeneous) The low limit of close loop is used as value [620 1/min]
C_N_CP_TOL	Maximum speed limit for activation of the RAMP_OPEN operation (homogeneous) This parameter is defined using the measurement of the MAP = f [MAF,N] The difference of the pressure (AMP – MAP) has to be higher than 200mbar in the area defined by C_N_CP_TOL and IP_MAF_CP_TOL. [3500 1/min]
IP_MAF_CP_BOL	Minimum mass air flow limit for activation of the RAMP_OPEN operation (homogeneous) In low MAF, the purge flow is not big enough to make change the canister saturation degree. The low limit of close loop is used as value
IP_MAF_CP_TOL	Maximum mass air flow limit for activation of the RAMP_OPEN operation (homogeneous) This parameter is defined using the measurement of the MAP = f [MAF,N] The difference of the pressure (AMP –

MAP) has to be higher than 200mbar in the area defined by C_N_CP_TOL and IP_MAF_CP_TOL.

Hint: The values defined for the window have an impact on the calibration of the chapter 6.2.1

These four calibrations will allow to select the canister load learning area during homogeneous combustion

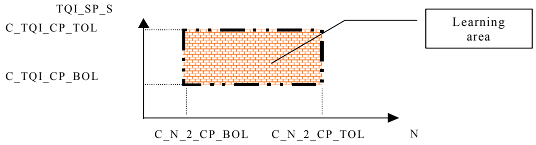
[50 mg/stroke] [400 mg/stroke]

28.18.9.4 Calculations for throttled VVL mode request

C_CL_MMV_VVL_REQ_BOL/ _TOL	Limit on CL_MMV for setting/ resetting LV_VVL_THR_MOD_REQ_CP Request for mode VVL mode with high CPS pressure to force a big purge flow at high canister load. [-]

28.18.9.5 Canister load learning window stratified

C_N_2_CP_BOL **C_N_2_CP_TOL** **C_TQI_CP_BOL** **C_TQI_CP_TOL**	Minimum / maximum engine speed / torque setpoint value limits for activation of the RAMP_OPEN operation (stratified) These four calibrations will allow to select the canister load learning area during stratified combustion



28.18.10 EVAC control (9008)

28.18.10.1 EVAC control – state machine

C_T_DLY_MAX_CP_0/1	Time between two MAX_PURGE operations for RAMP_OPEN_FAST activation for STATE_OPM_ENG_CP = ENG_MOD_0

	(_0) and STATE_OPM_ENG_CP != ENG_MOD_0 (_1). When leaving MAX_PURGE, a time counter T_DLY_MAX_CP starts to count backwards from this calibration value to 0. While the counter does not reach zero, a new learning mode via RAMP_OPEN_FAST is possible; if it reaches zero, the next learning mode has to be RAMP_OPEN [50 s]
C_T_DI_RAMP_OPEN_CP_0/1 **C_T_DI_RAMP_FAST_CP_0/1** **C_T_DI_MAX_CP_0/1**	Disable times of the states RAMP_OPEN, RAMP_OPEN_FAST and MAX_PURGE for STATE_OPM_ENG_CP = ENG_MOD_0 (_0) and STATE_OPM_ENG_CP != ENG_MOD_0 (_1). If closed loop canister purge ("normal purge") is abandoned e.g. due to high engine dynamic, the time counter T_DI_CP starts to count backwards from the corresponding calibration values to 0. While it does not reach zero, a re-entering in the last closed loop state is possible as soon as the conditions for closed loop canister purge are given again. If the counter reaches 0, closed loop canister purge has to start with a new learning mode. This does not affect the exit of closed loop canister purge by LV_CP_NEW_RAMP_OPEN. A re-entry in RAMP_OPEN_FAST is only possible, if LC_CP_TRAN_DEAC_CLL is set to 1, otherwise (switch=0), no reversible interruption of RAMP_OPEN_FAST is enabled.
C_T_LOCK_CP	Locking time for locking MIN_PURGE or NO_PURGE after canister empty detection.

	[20 s]
C_STATE_MOD_SEL_CP	If the canister is detected empty at the end of RAMP_OPEN, closed loop canister purge can be inhibited for a time C_T_LOCK_CP by setting C_STATE_MOD_SEL_CP to "NO_PURGE" or "MIN_PURGE". Then the so selected state is the maximum possible CP state until the timer T_LOCK_CP reaches zero. [MIN_PURGE]
LC_CP_TRA_MAX_MIN_CDN_ENA	enable transition MAX_PURGE → MIN_PURGE with STATE_CDN_CP
LC_CP_TRA_MIN_PURGE_DFT_ENA	enable transition NO_PURGE → MIN_PURGE with STATE_CDN_CP = NO_CDN
LC_CP_TRA_MIN_RAMP_OPEN_DIS	enable trans. NO_PURGE → MIN_PURGE, if no RAMP_OPEN or RAMP_OPEN_FAST is possible although STATE_CDN_CP=RAMP_OPEN or RAMP_OPEN_FAST [1]
LC_CP_TRA_OPL_CDN_CLL_NOT	enable transition WAIT_RAMP_OPEN → NO_PURGE if LV_CP_CDN_RAMP_OPEN, _FAST no more given [0]
LC_CP_TRA_OPL_CDN_NOT	enable transition WAIT_RAMP_OPEN → NO_PURGE if STATE_CDN_CP = NO_CDN This switch is redundant to LC_CP_TRA_OPL_CDN_CLL_NOT [0]

28.18.10.2 EVAC control – states

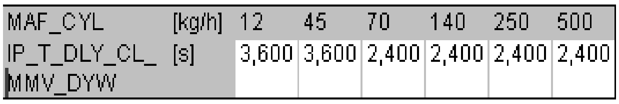
IP_PQ_CP_SP	[-]
C_CRLC_PQ_SP_CP	[0,25]
C_CL_MMV_PQ	[2]
	Setpoint value, filter constant and activation threshold on canister load for forced PQ (pressure quotient) during stratified mode for RAMP_OPEN and MAX_PURGE.

	A forced PQ is enabled in homogeneous lean burn mode if the switch LC_PQ_SP_CP_HOM_AFL is set to 1. The PQ setpoint can lead to further closing of the throttle thus leading to a higher pressure difference ambient ⇔ intake manifold thus enabling higher purge flows.
LC_PQ_SP_CP_HOM_AFL	Setpoint value, filter constant and activation threshold on canister load for forced PQ (pressure quotient) during stratified mode for RAMP_OPEN and MAX_PURGE. A forced PQ is enabled in homogeneous lean burn mode if the switch LC_PQ_SP_CP_HOM_AFL is set to 1. The PQ setpoint can lead to further closing of the throttle thus leading to a higher pressure difference ambient ⇔ intake manifold thus enabling higher purge flows. [0]
C_CL_CAT_DIAG	Canister load threshold to enable cat.diag. in MIN_PURGE (if cl < thd) If CL_MMV is below this threshold at the regular end of RAMP_OPEN, the flag LV_CL_CLC_AVL is set to 1 and passed to EGTR where it could enable a catalyst diagnosis in MIN_PURGE. [0,5]
C_CL_MMV_DYW	Dynamic window on CL_MMV value to stop leaning mode (RAMP_OPEN) and start MAX_PURGE. • The window is tuned to stop the RAMP_OPEN when the CL_MMV becomes stable inside the natural oscillation of FAC_LAM_MMV. • This tuning is done after IP_CL_CRLC_RAMP_OPEN • The tuning is done with empty canister (CL_MMV < 0,2) in idle (maximum effect by natural oscillation of lambda regulation). The window is defined as twice the value of CL_MMV oscillations. [0,024]
IP_T_DLY_CL_MMV_DYW	Minimum time duration of RAMP_OPEN when CL is inside the dynamic window

C_CL_MMV_DYW

Method:

- Charcoal canister empty
- Different stable engine load defined by MAF_CYL
- Start with RAMP_OPEN mode and measure the time spent to get CL_MMV stable and close to CL. Fill the IP_T_DLY_CL_MMV_DYW with the measured time.



MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_T_DLY_CL_MMV_DYW	[s]	3,600	3,600	2,400	2,400	2,400	2,400

C_T_DLY_MAX_RAMP_OPEN

Initialisation time duration for RAMP_OPEN mode.

If no lambda shift occurs during RAMP_OPEN mode, no re-initialsisation of the timer T_RAMP_OPEN_STAB will occur and RAMP_OPEN will be left after C_T_DLY_MAX_RAMP_OPEN.

Method:

- Charcoal canister empty
- Idle speed mode

The final valve opening is defined by FLOW_MAX value. Measure the time spent to reach about 4/5 of FLOW_MAX at idle.

[20s]

LC_LAM_LIN_CPS

Type of O2 sensor (0 → binary sensor , 1 → linear sensor)

LC_DLY_MOD_CP

Switch the flow incrementation feedback.

0: Flow incremented if gastransfer timer elapsed (T-DLY_CP_OPEN = 0) or lambda sensor switch from rich to lean (LV_CP_AFL_IT 0 -> 1)

1: Flow incremented if gastranfer timer elapsed and lambda sensor is lean.

LC_FLOW_CTL_INI_DEAC_CP

Enable the re-initialisation of FLOW_CTL_CPS at re-entrance of MAX PURGE.

C_T_CP_PURGE_TUBE_CP

This defines the time starting at entering WAIT_RAMP_OPEN while the purge valve is opened. There a two way how this can be used:

	1. For evacuating the gas in the pipeline from the ACF to the purge valve whose HC-concentration may be not corresponding to the real canister load. Then C_T_CP_PURGE_TUBE should correspond to the worst-case time that is needed to evacuate the whole pipeline. 2. For making a short (but big) opening of the CPS valve to remove adhesive effects that could affect the start-to-open point of the purge valve. Then a very short time (e.g. 100ms) should be chosen, to avoid high lambda deviations due to this opening.
C_FAC_T_DLY_PURGE_TUBE_CP	Factor that is applied to the gas delay time at the beginning of WAIT_RAMP_OPEN. Should be set to a value >1 to ensure that all the gas from 'purging the tube' has vanished from the intake manifold to not disturb the RAMP_OPEN learning. If a small C_FLOW_CPS_GRD_PURGE_TUBE_CLOSE is applied, the time until the valve is completely closed has the be taken into account by this factor.
C_T_MAX_MAF_INT_CP_0/1	Maximum time to remain in WAIT_RAMP_OPEN even if the MAF integral has not reached the below mentioned threshold (e.g. at very low MAF operating points), _0 for homogeneous and _1 for stratified combustion modes.
C_T_MIN_MAF_INT_CP_0/1	Mininum time to remain in WAIT_RAMP_OPEN even if the MAF integral has already exceeded the below mentioned threshold (e.g. at high MAF operating points), _0 for homogeneous and _1 for stratified combustion modes.
C_MAF_INT_MAX_CP_0/1	MAF integral thresholds depending on STATE_OPM_ENG_CP. This value has to be chosen according to the lambda controller stabilization MAF integral value to ensure that the controller is in stationary conditions.

28.18.10.3 EVAC control – flow calculations

LC_FLOW_CPS_AS

as required

C_FLOW_CPS_AS

Manual value for FLOW_CPS and activation switch for that.

LC_CP_FLOW_LIM_OPEN_FAST

enable old calc. of FLOW_OPEN_CPS: fuel mass based

As soon as the limit flow FLOW_OPEN_CPS is reached, RAMP_OPEN_FAST is left towards MAX_PURGE. With the switch = 1, the limit flow FLOW_OPEN_CPS can be taken from a map IP_FLOW_MAX_OPEN_CPS, this is similar to the calculation in RAMP_OPEN.

With the switch = 0, the limit flow is calculated based on the maximum possible fuel for RAMP_OPEN_FAST; this is the solution of previous canister purge versions.

ID_FLOW_INC_NR_CP

Increment number for FLOW_CTL_CPS calculation during RAMP_OPEN and

RAMP_OPEN_FAST (steps à 20ms)

The ramping-up of the flow during RAMP_OPEN and RAMP_OPEN_FAST consists of two phases: flow incrementation and gas transfer delay.

Each flow incrementation phase consists of a ID_FLOW_INC_NR_CP incrementation steps with the amount of IP_FLOW_INC_CP rsp. IP_FLOW_INC_FAST_CP. After the incrementation phase, the system waits for gas delay or rich → lean switch of a binary lambda sensor.

N_32	[1/min]	0	2048	4096	5120
ID_FLOW_INC_NR_CP	[-]	11	12	15	19

IP_FLOW_INC_CP

During RAMP_OPEN, the valve is opening smoothly in order to control the lambda

This calibration defines the slope of FLOW_CPS by incrementing directly FLOW_CTL_CPS in homogeneous mode. The first step is defined by C_FLOW_INI_CPS.

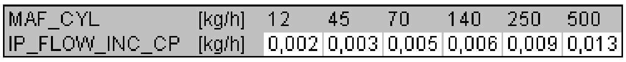
The IP_FLOW_INC_CP values are defined by calculation. These increments will be checked during driving tests in hot conditions

(charcoal canister loaded)

(Lambda max = 5-6 %, CL max = 2)

Method:

- Driving tests in hot conditions
- Different engine load for stable engine conditions
- Charcoal canister full loaded
- Ramp open mode
- Calibrate the maximum canister purge time equal to 20 s
- Calibrate the lambda adaptative duration equal to 20 s
- For different engine loads (MAF_CYL) adjust IP_FLOW_INC_CP in order to get no more than 5-6% of lambda variation during RAMP_OPEN mode.



MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_FLOW_INC_CP	[kg/h]	0,002	0,003	0,005	0,006	0,009	0,013

IP_FLOW_INC_FAST_CP

During RAMP_OPEN_FAST the CL is known so the purge valve can be opened faster than the one used in RAMP_OPEN mode. In this case the slope of the purge valve opening will be a compromise between opening speed and lambda control. This map is used for homogeneous combustion.

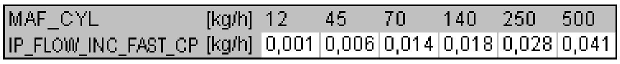
This calibration defines the slope of FLOW_CPS by incrementation of FLOW_CTL_CPS in Ramp Open fast.

Method

- Engine bench or Chassis Dyno
- Different engine loads for stable engine conditions
- Charcoal canister full loaded by butane or warmed fuel (50°C)
- Ramp open fast in Homogeneous mode
- For different engine loads (MAF_CYL) adjust IP_FLOW_INC_FAST_CP in

order to get ± 5-6 % maximum lambda variation during RAMP_OPEN mode. This calibration needs some repeated measurements. To get the repeatition fastly the time durations of some phases could be reduced (Lambda adaptation, Purge time duration)

These calibrations have to be refined during in-use tests in hot conditions (summer trip)



MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_FLOW_INC_FAST_CP	[kg/h]	0,001	0,006	0,014	0,018	0,028	0,041

IP_FLOW_DEC_CP

Purge opening decrement value for FLOW_CPS during RAMP_CLOSE mode. The slope of the closing prevents exhaust lambda variation.

This calibration defines the slope of FLOW_CPS.

Method:

- Charcoal canister full (legal procedure) and empty loads
- Canister Load learned in RAMP_OPEN and refined in MAX_PURGE.
- For different engine loads (MAF_CYL) adjust IP_FLOW_DEC_CP in order to get ± 5% maximum lambda variation at the end of RAMP_CLOSE mode. This calibration needs some repeated measurements. To get the repetition fastly the time durations of some phases could be reduced : Lambda adaptation, Purge time duration....

C_FAC_DEC_COR_CMU_CP

Purge opening decrement value for FLOW_CPS during combustion commutation.

This calibration defines the slope of FLOW_CPS.

Method:

- Charcoal canister full (legal procedure) and empty loads
- Canister Load learned in RAMP_OPEN and refined in MAX_PURGE
- Adjust C_FAC_DEC_COR_CMU_CP in

order to have no impact of the canister purge during the commutation. Compare with and without purge different types of commutation.

C_FAC_DEC_COR_1_CP	Other system functions could want to stop the canister purge function. The request of purge closing is controlled by the OBD2 controller. Depending on the type of request the closing speed is defined by C_FAC_DEC_COR_1_CP. The target of the calibration is to go in NO_PURGE mode fastly without taking care with pollution or lambda variation. Closing speed correction factor 1 is used for high speed closing *This calibration has to be done by the function transmitter of the close request* [1]
C_FAC_DEC_COR_2_CP	Other system functions could want to stop the canister purge function. The request of purge closing is controlled by the OBD2 controller. Depending on the type of request the closing speed is define by C_FAC_DEC_COR_2_CP. The target of the calibration is to go in NO_PURGE mode with a compromise between the closing speed and lambda variation. Closing speed correction factor 2 is used for medium speed closing *This calibration has to be done by the function transmitter of the close request* [1]
IP_FLOW_FAC_CP	Correction factor for FLOW_CPS applied in RAMP_OPEN, RAMP_OPEN_FAST or RAMP_CLOSE modes, for homogeneous and stratified modes. The correction value is depending of the CL value. The reference is based on canister full loaded (factor = 1). When the CL is lower the FLOW_CPS incrementation is increased. The values are defined by calculation. The values will be checked during driving tests with different charcoal canister loads. This factor must be calibrated by taken account that at each incrementation the

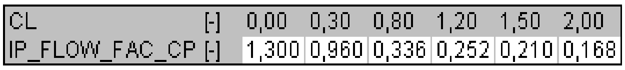
lambda variation must stay lower than 5-6%

Method:

- Driving tests, different charcoal canister loads
- Different engine load for stable engine conditions
- Ramp open mode

For different engine loads (MAF_CYL) adjust IP_FLOW_FAC_CP in order to control ± 5-6 % maximum lambda variation. This calibration needs some repeated measurements. To get the repeatition fastly the time durations of some phases could be reduced : Lambda adaptation, Purge time duration....

The calibration must be checked in Ramp open fast and ramp close mode for all the combustion modes.



CL	[-]	0,00	0,30	0,80	1,20	1,50	2,00
IP_FLOW_FAC_CP	[-]	1,300	0,960	0,336	0,252	0,210	0,168

LC_FLOW_CL_FAC_RAMP_OPEN_INH

As the accuracy of the CL value is much higher in RAMP_OPEN_FAST than in RAMP_OPEN, the weighting of the FLOW_GRD_CPS by IP_FLOW_FAC_CP should be turned off in RAMP_OPEN by setting the switch to 1, if the CL value has a high fluctuation during this state.

IP_FLOW_CPS_INC

(In MAX_PURGE, the closing of the valve must be the same as the one given by the ramp of FLOW_CTL_CPS during RAMP_CLOSE.)

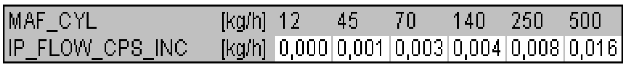
The purge valve opening/closing is done by FLOW_CTL_CPS variations.

In case where the evaporative system design is not good enough (fuel tank purged : the purge valve opening/closing has to be done slower. These calibrations allow to apply a 1st order on the purge flow needed. The purge flow integrated will define the purge valve opening.

Method:

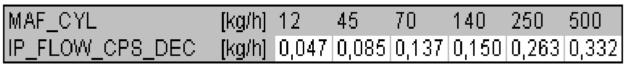
- Engine loads in stable conditions

- Charcoal canister full (legal procedure) and empty load
- CL learnt and MAX_PURGE mode acting.
- Change the purge target modifying IP_FUEL_FLOW_MAX_CP to get high variation in purge valve opening and purge flow needed.
- For different engine loads (MAF_CYL) adjust the INC and DEC in order to get ± 5% maximum lambda variation up to the valve opening setpoint is reached. This calibration needs some repeated measurements. To get the repeatition the time duration of Purge time duration could be lengthened.



MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_FLOW_CPS_INC	[kg/h]	0,000	0,001	0,003	0,004	0,008	0,016

IP_FLOW_CPS_DEC



MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_FLOW_CPS_DEC	[kg/h]	0,047	0,085	0,137	0,150	0,263	0,332

C_FLOW_INI_CPS

- This value is used to initialize FLOW_CTL_CPS at beginning of the RAMP_OPEN.
- This data is calculated to obtain most quickly the “start to open” valve opening point in idle. At the RAMP_OPEN start the purge valve opening has to be close to the “start to open” point defined in IP_CPPWM
- Method:
- Second breakpoint of purge flow axe in IP_CPPWM > C_FLOW_INI_CPS
- Check on engine bench for different engine loads that this value is enough to open correctly the valve and create e first deviation of lambda
- [0,01 kg/h]

IP_FLOW_MAX_OPEN_CPS

FLOW_CPS for fully opened CPS (CPPWM=100%)

This map defines the maximum purge flow value [kg/h] when the valve is fully opened; the values are from the flowbench measurements or from IP_CPPWM

The calibration needs the values

- MAF_CYL = f(MAF[mg/tdc], N_32[rpm]) and
- MAP = f(MAF[mg/tdc], N_32[rpm]) and

MAF_CYL ⇒ MAP ⇒ MAX FLOW CPS ⇒ IP_FLOW_MAX_CPS (MAX FLOW CPS)

An excel file FLOW_MAX.xls allows to calculate the calibration values to fill up automatically the FLOW_MAX_CP.

This map characterizes the maximum flow authorized during RAMP_OPEN or RAMP_OPEN_FAST mode. This map can be considered as flow setpoint for the learning modes.

At Idle IP_FLOW_MAX_OPEN_CPS should be limited. In order to keep sufficient idle actuator authority on purge the maximum purge flow will be limited at 10% of the maximum total air engine flow.

[-]

IP_FAC_AMP_CP

The canister purge flow ($Q_{\text{Purge Map}}$) values are defined at standard sea level barometric pressure ($P_{\text{Purge Map}}$) =1013 mb.

The factor correction by IP_FAC_AMP_CP allows to adapt the flow at the current barometric pressure.

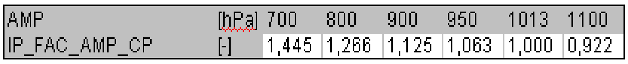
The variation of the canister purge flow is depending on the upstream pressure. This variation follows the law:

$Q_{\text{Purge}} = Q_{\text{Purge MAP}} * [P_{\text{baro current}} / P_{\text{Purge MAP}}]$

When the valve opening has to be

defined to get a needed canister purge flow the inverted law is used :

$Q_{\text{Purge MAP}} = Q_{\text{Purge}} * [P_{\text{Purge MAP}} / P_{\text{baro current}}]$



AMP	[hPa]	700	800	900	950	1013	1100
IP_FAC_AMP_CP	[-]	1,445	1,266	1,125	1,063	1,000	0,922

IP_FAC_TIA_CP

The canister purge flow ($Q_{\text{Purge Map}}$) values are defined at standard ambient temperature = 25°C

This factor correction allows to adapt the flow at the current ambient temperature

The variation of the canister purge flow follows the law:

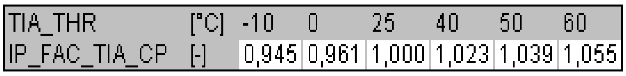
$Q_{Purge} = Q_{Purge\ MAP} * [T_{Purge\ MAP} / T_{current}]^{0,5}$

With T[°K]

When the valve opening has to be defined to get a needed canister purge flow the inverted law is used :

$Q_{Purge\ MAP} = Q_{Purge} * [T_{current} / T_{Purge\ MAP}]^{0,5}$

With T[°K]



TIA_THR	[°C]	-10	0	25	40	50	60
IP_FAC_TIA_CP	[-]	0,945	0,961	1,000	1,023	1,039	1,055

C_FLOW_CPS_SP_PURGE_TUBE

Flow setpoint for the first phase in WAIT_RAMP_OPEN. Depending on the intended usage (refer to C_T_CP_PURGE_TUBE), this flow setpoint has to be determined different.

1. Purging the tube: the flow setpoint must not be higher than the worst case MIN_PURGE flow setpoint for idle speed to not create too high lambda deviations.
2. Short, defined opening: if used to remove adhesive effects that could influence the SOP of the purge valve, a high flow setpoint (that ensures an opening of the valve, e.g. a CPPWM of about 20%) has to be chosen.

C_FLOW_CPS_GRD_PURGE_TUBE

Flow gradient for the first phase in WAIT_RAMP_OPEN. For purging the tube, smaller gradients can be chosen to avoid high lambda deviations. For a short, defined opening a big gradient should be applied to ensure that the valve is opened immediately to ensure that the target flow can be reached within the short time C_T_CP_PURGE_TUBE.

C_FLOW_CPS_GRD_PURGE_TUBE_CLOSE

Flow gradient for tube purging in WAIT_RAMP_OPEN (at closing cps). For purging the tube, smaller gradients can be chosen to avoid high lambda deviations. For a short, defined opening a big gradient should be applied to ensure that the valve is closed immediately to create only a short (but high) disturbance.

C_MAF_DLY_CRLC_CPS	Correlation constant for MAF_CPS_DLY-filter
C_T_DLY_CL_MDL	Time to activate CL-Model
IP_FLOW_MAX_OPEN_CPS_2	Max. limit for FLOW_CTL_CPS value for RAMP_OPEN & RAMP_OPEN_FAST with STATE_OPM_ENG_CP = 1
LC_CP_FLOW_LIM_OPEN_FAST	enable old calc. of FLOW_TAR_CPS during ROF: fuel mass based
LC_CP_TRA_MAX_MIN_CDN_ENA	enable transition MAX_PURGE (MIN_PURGE with STATE_CDN_CP = NO_CDN
LC_CP_TRA_MIN_PURGE_DFT_ENA	1: enable transition NO_PURGE (MIN_PURGE with STATE_CDN_CP = NO_CDN
LC_CP_TRA_MIN_RAMP_OPEN_DIS	enable trans. NO_PURGE (MIN_PURGE, if no RO or ROF is possible although STATE_CDN_CP=RO or ROF
LC_CP_TRAN_OPEN_MAX_ENA	enable transition RO (MaxP if flow target reached
C_MFF_RATIO_FLOW_INI	Maximal percentage of MFF at reentering MAX_PURGE.
C_FAC_FLOW_MAX_INI_CP	Factor to limit FLOW_CTL_CPS on re-entering MAX_PURGE in case of low canister load.

28.18.10.4 EVAC control – Padé- and Low pass filter calculations

IP_T_DLY_CP

This map allows to characterize the gas transfer duration between canister valve and oxygen sensor.

These value are used to calculate the Padé filter outputs for FLOW_CPS and MAF_CYL.

Method:

- Chassis Dyno and stable engine conditions
- Charcoal canister half loaded
- Canister purge strategy deactivated
- Opening/closing purge valve set by the application system calibration C_CPPWM_AS.

For different engine load (MAF_CYL) measurement of the time duration from the purge valve switching (open/close) to the beginning of the variations of :

- Lambda controller or
- O_2 sensor period or
- UEGO sensor

The value of IP_T_DLY_CP is calculated by dividing the measured time by the calculation recurrence of the PADÉ filter (100 ms).

MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_T_DLY_CP	[0,1 s]	4,600	3,320	2,680	2,000	1,800	0,980

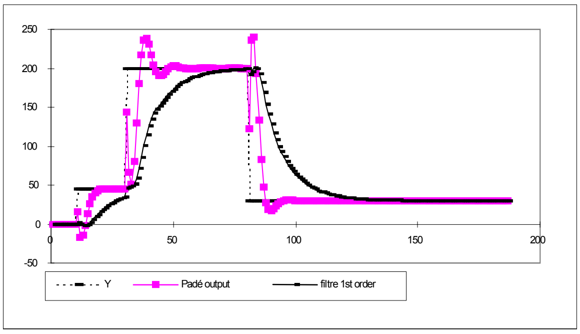
C_FLOW_DLY_CRLC_CP [0,3]

C_MAF_DLY_CRLC_CP Filter coefficient of FLOW_DLY_CP and MAF_TOT_CP_DLY_.

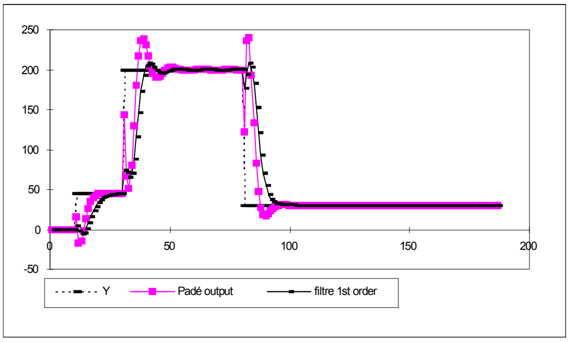
These coefficients cancel the PADE output oscillations.

The following example using two filter coefficient values show the effect on the output.

Filter coefficient = 0.1 ***Value too small because the ouput value is too much delayed***



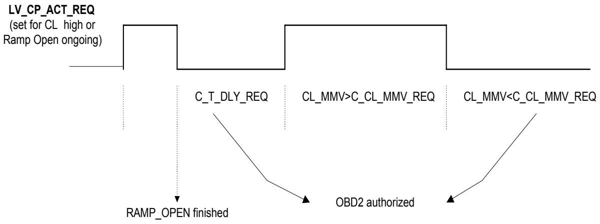
Filter coefficient : 0.3 ***Value adapted because the ouput value is close of PADE output without oscillations***



[0,3]

28.18.10.5 EVAC control – Common submodules

C_CL_MMV_REQ	When the canister load is bigger than the threshold C_CL_MMV_REQ the canister purge function does not be interrupted by another function (OBD2.....). The calibration value is defined by maximum canister load with butane. Take care this calibration value does not prevent the OBD2 and adaptive functions during SHED test or standard driving cycle when the charcoal canister is full loaded. [0,5]
C_T_DLY_REQ_CP	Time to set LV_T_DLY_REQ_CP The time C_T_DLY_REQ_CP authorizes ODB2 diagnosis activation even C_CL_MMV_REQ is set. When this time duration is elapsed the canister purge load takes highest priority. This time duration is refined in accordance with the standard driving cycle.



[120 s]

28.18.11 Injection calculation (903M)

28.18.11.1 Injection calculation – injection buffer

The injection correction model is characterized by a delay and a 1st order filter. The delay should represent the gas transfer time through the intake manifold, the appending filter should represent the flow dilution behaviour.

NC_NR_CP_BUF	Size of the fuel buffer MFF_ADD_BUF The buffer size has to be determined according to the expected delay times between CPS and injection valves.
LC_FLOW_CPS_CAL	Choice of recurence for injection buffer calculation. The injection buffer can be calculated with two different recurrencies: 20ms rsp. 40ms, LC_FLOW_CPS_CAL leading to a recurrency of 20ms. By selecting a slower update rate, higher gas delay times can be modelled with a given buffer size.
LC_DLY_GAS_CLC_VERS_CP	Determines whether the gas delay determination is done via the maps IP_MFF_DLY_CP and IP_IDX_STEP_CP (= 0) or calculated from the intake manifold volume with an temperature and pressure correction (=1). If a reliable determination of the IM volume is possible, this version should be preferred as the calibration effort for the first version is much higher.
IP_MFF_DLY_CP **IP_IDX_STEP_CP**	These calibrations allow to characterize the intake manifold gas transfer between the canister purge valve and the cylinders inlet. *Method:* • Engine bench or Chassis dyno in stable engine load conditions • Canister purge strategy deactivated • Change the injection time using the bench function (C_TI_AS) • Open and close the CPS valve via C_CPPWM_AS / LC_CPPWM_AS • For different MAF and MAP perform measurements of the gas transfer duration from the injector valve change to lambda deviation. • The time delay (valve to injector) is the difference between the map

IP_T_DLY_CP (time between the valve and the lambda sensor) and the previous measurement.

- The calibration values are the time measured divided by the calculation recurrence of MFF_BUF_CP.
- The recurrence is 20 or 40ms, depending on calibration.
- Set IP_MFF_DLY_CP all to maximum delay time!
- The buffer is filled with several elements calculated by IP_IDX_STEP_CP (high MAF -> many elements; low MAF -> less elements)

IP_MAF_KGH_PRED_COR_MAP_TIA

MAF correction as function of MAP and TIA to do a correciton in the gas delay time caused by differet intake manifold preassure

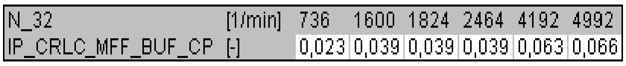
IP_CRLC_MFF_BUF_CP

The dilution behavior through the intake manifold is characterized by a 1st order filter. The filter coefficient is defined such as the exhaust lambda variations are the smallest from purging change.

Method:,

- IP_MFF_DLY_CP has to be tuned before.
- IP_CRLC_MFF_BUF_CP minimizes the lambda variation when canister fuel flow is controlled by the purge function.
- Engine bench or Chassis dyno in stable engine load conditions
- Charcoal canister half loaded
- Canister purge strategy activated in MAX_PURGE mode (CL known)
- Opening/closing purge valve set by changing the purge target (IP_FUEL_FLOW_MAX_CP)

For different engine load (N variation) measurement of the lambda variation (Linear sensor) due to purge target change. Adjust IP_CRLC_MFF_BUF_CP value before each change in order to have no exhaust lambda peak (lean or rich).



N_32	[1/min]	736	1600	1824	2464	4192	4992
IP_CRLC_MFF_BUF_CP	[-]	0,023	0,039	0,039	0,039	0,063	0,066

28.18.11.2 Injection calculation – torque correction in stratified

IP_MFF_MIN_CP_S	Canister purge margin on the fuel flow minimum value in stratified mode. This margin is used to prevent the minimum injection time in altitude or on deceleration. [5 mg/stk]
IP_FAC_TQI_CP	Factor used to convert a fuel mass in a torque. FAC_TQI_CP is a decimal factor which permit to convert a fuel mass in a torque. Because of the different resolutions of TQI_ADD_CP and MFF_CPS, the conversion (1389 mg/tdc / 1024 Nm) has to be taken into account by the calibration of FAC_TQI_CP. That means, if you want MFF_CPS = TQI_ADD_CP, then you have to calibrate FAC_TQI_CP to 0.73 (1024/1389) and not to 1. *Method:* • On engine dyno • On a reference point (N, TQI_SP_S, lambda) • Stratified mode • Open the valve with a full loaded canister (c_cppwm_as and lc_cppwm_as) • Decrease the fuel injected to go back to the reference point [0,67]

28.18.12 Torque reserve for canister purge at idle (800X)

LC_TQ_ADD_CP_INH	inhibit switch [1]
IP_TQ_ADD_CP	If not used set to [0 Nm]
C_CRLC_TQ_ADD_CP	Deactivation switch, filter constant and base map for canister purge torque reserve at idle. By increasing the engine load without increasing the torque output, a higher purge rate is possible due to the increase of engine's fuel demand (for a given purge fuel ratio). The torque reserve is only

applied in MAX_PURGE and idle speed depending on canister load CL_MMV.

Method :

- idle speed condition
- no additional load or torque reserve (cooling fan, air conditioning…)
- use the map ip_tq_n_add_is to change the torque reserve (ref chapter idle speed)
- Using the engine roughness criteria, measure the engine stability when you change torque reserve.

Make measurements for several torque reserve and record the injection time and the spark advance.
[0,5]

28.18.13 Canister Purge Valve Adaptation (902H)

Canister Purge Valve Adaptation Purpose of the canister purge solenoid (**CPS**) is to purge the tank venting charcoal filter by opening a pipe from the filter to the engine manifold; so the filter's CH-load is desorbed by fresh ambient air and sucked into the engine. For countinuous control of the purge airflow the CPS is opened pulse wide modulated (PWM).

Due to future legislation the CH-emissions of vehicles have to be cut down; one task is to accellerate purge. To keep the purge airflow as high as possible without disturbing the engine the CPS has to be controlled as accurately as possible. This means that the engine controller (which is also responsible for canister purge) has to „know" as much as possible about any car's individual CPS. The most important parameter is the start-to-open-point (**STO**); a dead time between switching on electric power and opening of the CPS; as visible in the figure beside. If the opening time (depending on PWM and frequency) is below this dead time the CPS will never open. The parameter STEP is presently assumed to be quiet low and therefore neglected while SLOPE is estimated to be almost constant as specified.

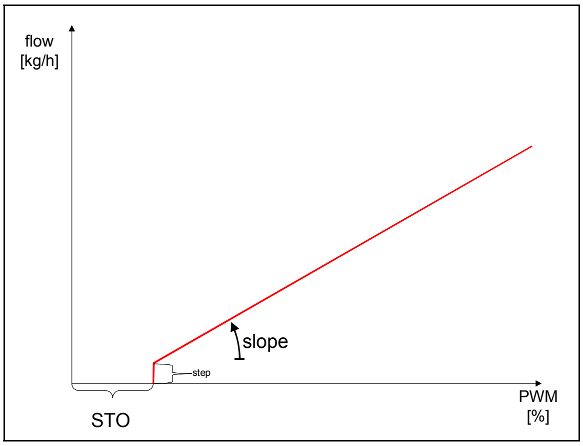


Figure: characteristic line of CPS

Up to now an average STO has been extimated, but expected future developments like high-flow-CPSs or the need for a better performance of canister purge induce an increasing demand for an exactly measured STO.

To measure the STO the CPS is ramp-opened until a decreasing lambda-value (caused by the purged HCs) is detected. For accurate measuring the HC flow has to be as high as possible without disturbing the engine. So the adaptation frequency is calculated depending on canister load (CL) and calibrated values. A too low CL interdicts adaptation.

Boundary conditions needed to start adaptation:

STATE_CP	3
LV_HOM_ACT	1
LV_CL_CLC_VLD	1
N	between C_N_MIN_CPS_AD and C_N_MAX_CPS_AD
T_AST	above C_T_ST_CPS_AD
LV_CPS_AD_INH	0
TQI_AV	between C_TQI_AV_MIN_CPS_AD and C_TQI_AV_MAX_CPS_AD
CL_MMV	above C_CL_MIN_CPS_AD
LV_CP_CLOSE_REQ	0
LC_CPS_AD_CHK	0 This flag (=1) is to start the function overriding **all** application conditions. If the function is started using this flag it will not stop until a sufficient lambda-deviation is detected (=STO learned). When function is going into wait-state after lerning, set flag back to 0. **Attention !** If no sufficient lambda-deviation is detected (i.e. if CL is too low) the CPS will be opened up to 100% PWM and stay there until the flag is reset.
FAC_LAM_CP	to take from engine model

Required for application work:
- Charcoal-filter with CL about 1,5 (75%)
- Min-flow-CPS
- Max-flow-CPS

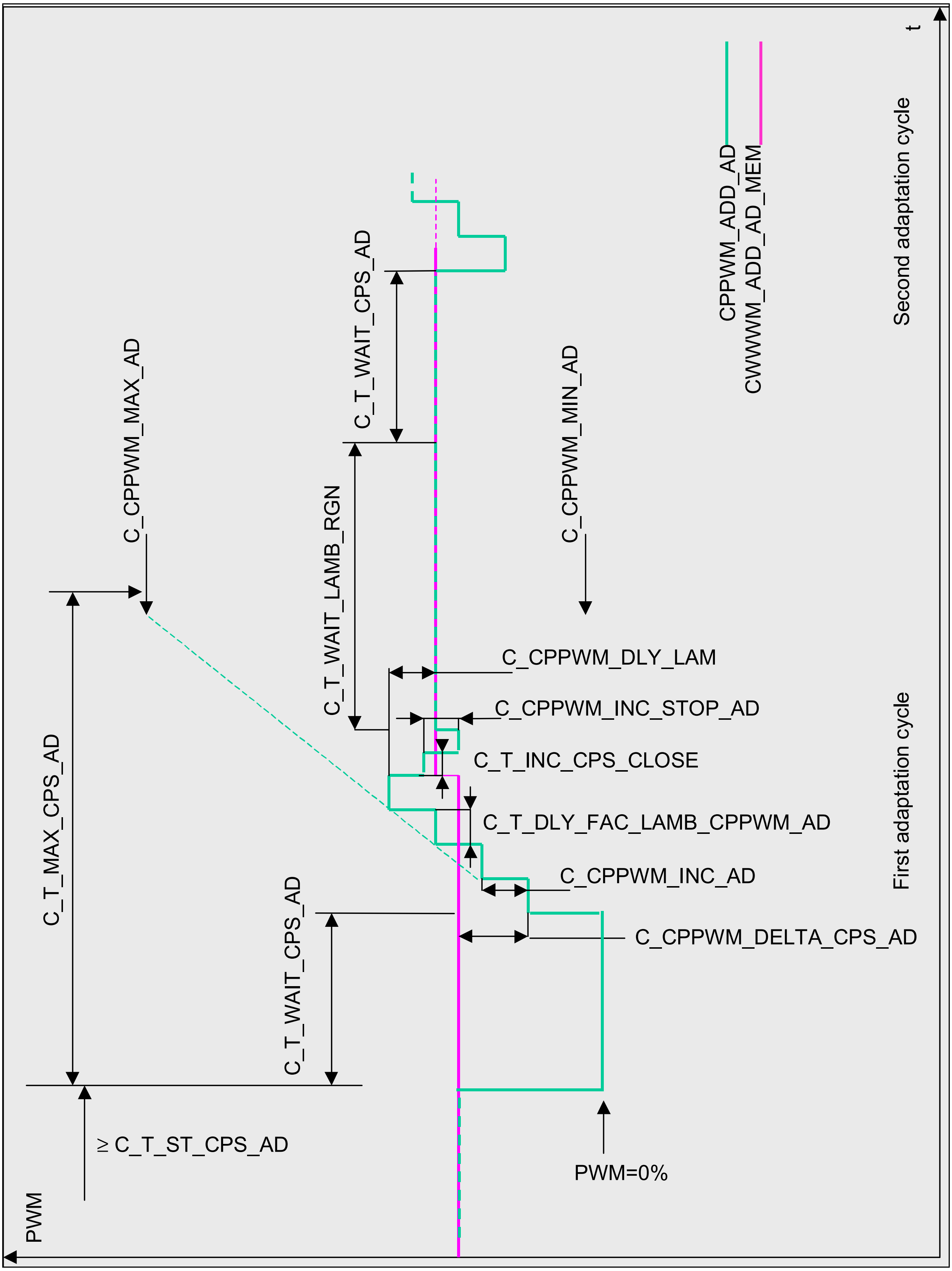


Figure: Adaptation cycle with calibraion values

Calibration method:

C_FRQ_MIN_CPS_AD	Minimum working frequency of CPS or default frequency of fixed-frequency-systems**; see CPS spec** or try 10 [Hz]
C_FRQ_MAX_MEC_CPS	Maximum allowed frequency of CPS**; see CPS spec** or try 30 [Hz].
C_FAC_LAMB_DELTA_MIN_CPS_AD	Minimum detectable lambda deviation caused by adaptation; **just above lambda-noise**. Watch FAC_LAMB at stable idle to estimate lambda noise or try 3 [%] as default.
C_FAC_LAMB_MAX_CPS_AD	Maximum allowed lambda deviation during adaptation without disturbing the engine; see engine or controller SPEC or: At low temperature (“cold morning”) open CPS stepwise, watch FAC_LAMB and see when engine is disturbed (sound/speed/vibrations).
C_CPPWM_AD_RATIO	Weighting factor of newly learned STO against old memorized value for calculation of new memorized value. 0.1 should be a useable default value. 1 can be used for monitoring during application work. This value acts like a “damper” against a badly learned STO.
C_CPPWM_DELTA_AD	Difference between PWM at start of adaptation and old memorized STO value to make sure adaptation starts below current STO. Should cover the measuring diffusion and a (smaller) possible drift of STO. This value also assures that after a “too high” learned STO the function will “climb back down”. Must be less than C_CPPWM_MIN_AD-C_CPPWM_DLY_LAM to avoid negative PWM value.Default: 5 [%]
C_CPPWM_DLY_LAM	Assuming the stepwise opening of CPS has continued between STO and lambda reaction (C_T_INC_AD < real physical time delay of C_T_DLY_FAC_LAMB_CPPWM_AD) this subtractive value compensates the PWM difference. Start application work with 0 [%]. Adapt using fully loaded canister at minimum frequency and watch FAC_LAMB. If learned STO (CPPWM_ADD_AD_MEM) is always higher than STO at beginning lambda deviation set the difference.
C_CPPWM_INC_AD	PWM increment for CPS opening during adaptation; depends on demanded function resolution and lambda regulation delay time. Has to be less than C_CPPWM_DELTA_AD. Default 1[%].
C_CPPWM_INC_STOP_AD	PWM icrement for CPS closing. Default 1[%].
C_CPPWM_MAX_AD	Maximum plausible STO-value at FRQ_MIN. See CPS spec or use 40[%] at 10Hz (= 40ms) as default.

C_CPPWM_MIN_AD	Minimum plausible STO-value at FRQ_MIN. Use 1[%] at 10Hz (= 1ms) or see CPS spec. Must be above (C_CPPWM_DELTA_AD to avoid negative PWM at next adaptation. Must be equal or higher than C_CPPWM_DLY_LAM+C_CPPWM_DELTA_AD
C_CYCNR_CPS_AD	Number of successful adaptations aimed during driving cycle. Default about 2 [-].
C_N_MAX_CPS_AD	Upper threshold of engine speed for adaptation. Default upper threshold of idle window; see engine spec or use 1200 [min^-1] as default.
C_N_MIN_CPS_AD	Lower threshold of engine speed for adaptation. Default lower threshold of idle window; see engine spec or use 600 [min^-1] as default-
C_T_DLY_FAC_LAMB_CPPWM_AD	Delay time between fuel purged into manifold and change of FAC_LAM value. Depends on engine speed and design. 1 [sec] may be a useable default value. After opening step of CPS this time passes until check for lambda deviation.
C_T_INC_CPS_CLOSE	Time increment for CPS closing, use 0,2 [sec] as default.
C_T_MAX_CPS_AD	Maximum allowed duration of one adaptation cycle. Try default 30 [sec]. As the figure shows, this time has to cover C_T_WAIT_CPS_AD and the ramp from C_CPPWM_MIN_AD-C_CPPWM_DELTA_AD to C_CPPWM_MAX_AD. So: C_T_MAX_CPS_AD≥ (C_CPPWM_MAX_AD- C_CPPWM_MIN_AD+ C_CPPWM_DELTA_AD)* C_T_INC_CPS_AD+ C_T_WAIT_CPS_AD
C_T_ST_CPS_AD	Minimum time between start engine and adaptation to ensure stable engine conditions. Default 60 [sec].
C_T_AST_DEAC_CPS_AD	Maximum time between start engine and end or abort of adaptation. If set to maximum value of T_AST this condition is disabled. Enter default 10000 [sec] to disable.
C_T_WAIT_CPS_AD	Delay time between function start and begin of CPS opening. Can be 0 [sec] if no delay demanded by application engineer.
C_T_WAIT_LAMB_RGN	Delay time after adaptation to regain stable conditions of lambda regulation. Try 10 [sec] and watch FAC_LAMB.
C_TQI_AV_MAX_CPS_AD	Upper threshold of engine torque for adaptation. Default upper threshold of idle window; see engine spec or use 40 [Nm]
C_TQI_AV_MIN_CPS_AD	Lower threshold of engine torque for adaptation.

	Default lower threshold of idle window; see engine spec or use 60 [Nm]

For first calibration or fixed frequency adaptation the frequency variation has to be terminated as follows:

C_FRQ_CLC_1_CPS_AD	=0
C_FRQ_CLC_2_CPS_AD	=C_FRQ_MIN_CPS_AD
C_CL_MIN_CPS_AD	Try as default: C_CL_MIN_CPS_AD=1 Use full canister and start purge to get current CL.Start adaptation with min-flow-CPS to get FAC_LAMB at STO. C_CL_MIN_CPS_AD=CL * C_FAC_LAMB_DELTA_MIN_CPS_AD/ FAC_LAMB

For frequency-variation:

The reliability of frequency calculation depends not only on it's own calibration but also on the precision of CL-detection which – vice versa – depends on CPS-adaptation. Therefore an iterative calibration is recommended:

1. Complete calibration as described below.
2. Wait for purge to detect CL again.
3. Recalibrate C_CL_MIN_CPS_AD, C_FRQ_MAX_CPS_AD, C_FRQ_SLOP_CPS_AD.
4. If a significant difference appears: Start again with step 2.

Use canister with CL about 1,5 (=75%).

Mount max-flow-CPS and start adaptation. Get FAC_LAMB at STO. Start purge to get current CL. Calculate:

$$\mathrm{FRQ\_MAX\_CL\_MAX} = \frac{\mathrm{C\_FRQ\_MIN\_CPS\_AD} \times \mathrm{C\_FAC\_LAMB\_MAX\_CPS\_AD} \times 2}{\mathrm{FAC\_LAMB} \times \mathrm{CL}}$$

FRQ_MAX_CL_MAX has to be ≥ C_FRQ_MIN_CPS_AD !

Mount min-flow-CPS and start adaptation. Get FAC_LAMB at STO. Start purge to get current CL. Calculate:

$$\mathrm{FRQ\_MIN\_CL\_MAX} = \frac{\mathrm{C\_FRQ\_MIN\_CPS\_AD} \times \mathrm{C\_FAC\_LAMB\_DELTA\_MIN\_CPS\_AD} \times 2}{\mathrm{FAC\_LAMB} \times \mathrm{CL}}$$

C_CL_MIN_CPS_AD	Crossing of FRQ_MIN-hyperbola and C_FRQ_MAX_MEC_CPS-line. Mount min-flow-CPS and start adaptation. Get FAC_LAMB at STO. Start purge to get current CL. Calculate: C_CL_MIN_CPS_AD=CL * C_FRQ_MIN_CPS_AD /

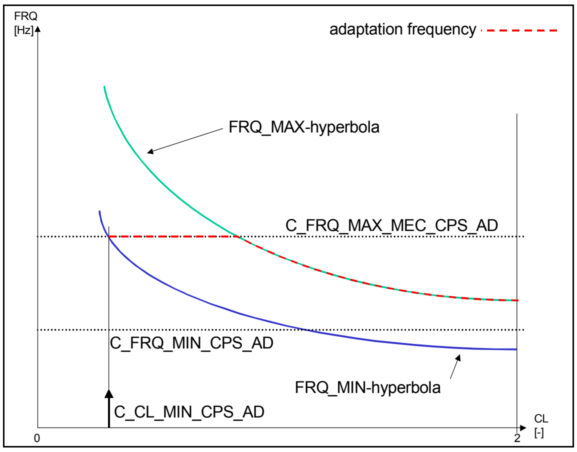
C_FRQ_MAX_MEC_CPS*
C_FAC_LAMB_DELTA_MIN_CPS_AD/
FAC_LAMB

Or try as default:
C_CL_MIN_CPS_AD=1

C_FRQ_CLC_1_CPS_AD,

C_FRQ_CLC_2_CPS_AD

The figure below shows two hyperbolas: The mimimum frequency required to generate a detectable FAC_LAMB and the maximum allowed frequency to not disturb the engine. The CL-range for adaptation is limited by the point where the minimum required frequency is getting higher than the maximum possible working frequency of the CPS.



Formula for hyperbola calculation:

$$FRQ = \frac{C\_FRQ\_CLC\_1\_CPS\_AD}{CL}$$

For precise measuring the working frequency will always be as high as possible (FRQ_MAX or C_FRQ_MAX_MEC_CPS). For individual tuning a second constant allows to move the hyperbola:

$$FRQ = \frac{C\_FRQ\_CLC\_1\_CPS\_AD}{CL} + C\_FRQ\_CLC\_2\_CPS\_AD$$

Use canister with CL about 1,5 (=75%).

Mount max-flow-CPS and start adaptation. Get FAC_LAMB at STO. Start purge to get current CL during RAMP_OPEN.
Calculate:

C_FRQ_CLC_1_CPS_AD=CL*C_FRQ_MIN_C

	PS_AD*C_FAC_LAMB_MAX_CPS_AD/ FAC_LAMB **Necessary condition, check:** C_FRQ_CLC_1_CPS_AD/2+ C_FRQ_CLC_2_CPS_AD ≥ C_FRQ_MIN_CPS_AD Fine tuning: Move frequency hyperbola by changing C_FRQ_CLS_2_CPS_AD or stretch/compress by changing C_FRQ_CLC_1_CPS_AD
C_FLOW_CLC_DIF_CPS_AD	Difference between C_FLOW_CLC_MAX_CPS_AD and begin of ramp
C_FLOW_CLC_MAX_CPS_AD	Maximum FLOW threshold for weighting-STO ramp
C_PRS_CPS_MAX_CPS_AD	Maximum PRS_CPS-threshold for CPS-adaptation
C_PWM_CLC_DIF_CPS_AD	Difference between CL_PWM_CLC_MAX_CPS_AD and begin of ramp
C_PWM_CLC_MAX_CPS_AD	Maximum CPPWM_CPS threshold for weighting-STO-ramp
IP_CPPWM_OPEN_CPS	Weighting factor for STO according to CPS characteristic
C_CPPWM_ADD_AD_MEM_NVMY_INI	Initialization value for CPPWM_ADD_AD_MEM value before first adaptation run. This value has to be subtrakted from the map IP_CPPWM taking the factors IP_CPPWM_OPEN_CPS, C_FLOW_CLC_MAX_CPS_AD, C_FLOW_CLC_DIF_CPS_AD, C_PWM_CLC_MAX_CPS_AD and C_PWM_CLC_DIF_CPS_AD.
C_T_CPS_AD_HOM_REQ	This constant should be calibrated long enough to ensure that a transition from stratified combustion mode to homogeneous combustion mode can take place and the other conditions for CPS adaptation (e.g. entering WRO state) can be fulfilled again.
C_T_DLY_CPS_AD_HOM_REQ	Minimum time between to homogeneous requests. Should be synchronized with fixed time slices / FTP cycle to ensure that homogenous combustion mode is not requested too frequently.
C_CTR_CPS_AD_HOM_REQ_MAX	Maximum number of homogenous requests per driving cycle. The maximum number of homogenous requests should also be chosen depending on the fixed time slices / FTP cycle to avoid too many homogenous requests that

	could affect emissions/ flow rate/ fuel consumption negatively.

28.18.14 Canister Load (CL) Model (907U)

C_CRLC_CL_MDL	Correlation constant for CL_MDL filtering [0,5]
C_FAC_T_CL_MDL_ACT	correction factor for re-init of CL mdl ena time counter after regaular leaving of active mode [1]
C_T_CL_MDL_ACT_MAX	Maximum time of CL model usage [40s]
C_T_CL_MDL_ACT_OFS	offset for re-init of CL mdl ena time counter after regaular leaving of active mode [30s]
IP_T_CL_MDL_CLL_ENA	Minimum time in CP closed loop operation to enable CL model usage [40s]
C_T_CL_MDL_CLL_ENA_INC	Increment for enable time counter if not in CP closed loop operation [0,1s]
IP_CL_MDL_SLOP	Timeshape over CL_MDL_SLOPE [-]
IP_CL_MDL_VALUE	Increment value on FLOW_SP_CPS [-]
LC_CL_MDL_ACT	Switch for activation of CL model calculation [0]
LC_CL_MDL_ENA	Switch for enabling CL model usage [0]

28.19 Calibration hints for EVAC control (cus)

28.19.1 Calibration interfering functions

Before to start the calibration work, it is absolute mandatory to have stable operating

- Lambda-Control and Injection Prec-Control
- Intake Manifold Model
- Idle Speed Control

Deficiencies in the calibration quality of the above mentioned may block the calibration/operation or demand a recalibration. Especially the Lambda Control and the Injection Pre-Control must be at a mature standard (Lambda Control output < 5% in stationary engine conditions).

28.20 Main operating situations

NO_PURGE

During this mode, CPS valve is closed

This mode is used

- when conditions to authorize evap control are not fulfilled (TCO, ...)
- while the lambda adaptation phase is ongoing
- in case of failures (CPS valve, ...)
- in case of OBD2 diagnosis request (catalyst check...)
- in case of evaporative system monitoring active

MIN_PURGE

The MIN_PURGE or open loop operation operation is used when the charcoal canister has to be purged even if its load cannot be learned.

The conditions are:

- Vehicle is not equipped with A/F control
- A/F regulation component failures (lambda sensor...)
- Transient driving conditions. In this case, the purge flow is adjusted to have a maximum of 5% A/F ratio deviation with a saturated canister.

NORMAL_PURGE / closed loop operation

The normal operation is used when the determination of the canister load is finished.

Purge flow is adjusted to have the maximum of A/F ratio deviation with a saturated canister.

The NORMAL_PURGE operation is proceeded in the following sequences:

RAMP_OPEN → MAX_PURGE → RAMP_CLOSE
RAMP_OPEN_FAST → MAX_PURGE → RAMP_CLOSE

RAMP_OPEN

At first start of the NORMAL_PURGE the canister load is unknown. So, the purge valve is opened slowly using a ramp (a high deviation of the lambda control shall be prevented). During this learning the fuel injection time is corrected to avoid emission and driveability problems. The Canister Load is determined by the fuel $_{purge}$ / flow $_{purge}$ ratio.

The RAMP_OPEN operation is finished when :

- CL is learned and stable or
- The canister fuel flow (MFF_CP) is greater than a threshold

RAMP_OPEN_FAST

When the canister load is considered as known, indicated by beeing within a certatin time interval since last MAX_PURGE, the closed loop operation may start with a faster learning phase / ramp. This RAMP_OPEN_FAST operation is finished when :

- The purge flow greater than a threshold
- The canister fuel flow (MFF_CP) is greater than a threshold.

MAX_PURGE

During MAX_PURGE operation the CPS valve is controlled by the canister load, thus a constant fuel ratio (fuel from cps / total fuel) is applied and the valve opening is controlled to keep this constant fuel ratio by adjusting the total purge flow.

The target is to reach high efficiency purging. When the CL is low the purge valve opening will be increased to find the maximum limit of canister fuel flow.

RAMP_CLOSE

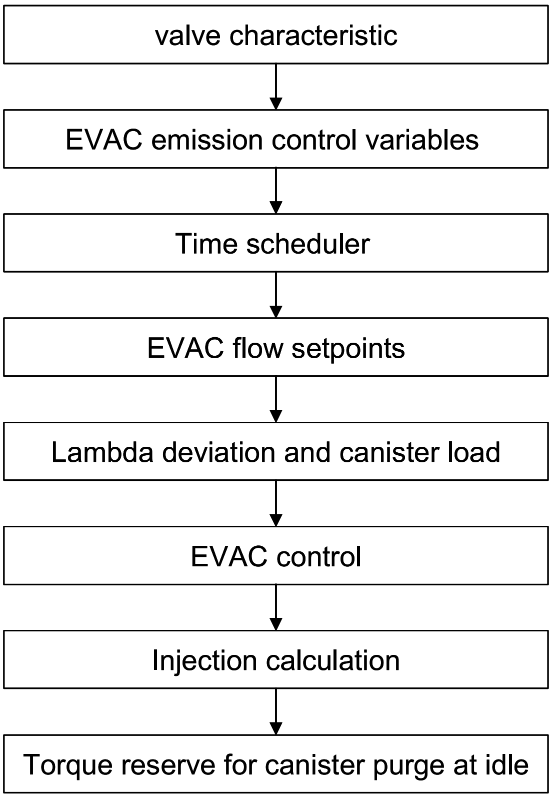
When the purge time is elapsed the purge flow is decreased via a ramp in RAMP_CLOSE operation.

28.21 Syntax

CalibrationDataName	description [defaultValue]

28.21.1 Calibration flowchart

The following flowchart shows the order to calibrate the EVAC functionalities. A continuous refining in this order is mandatory to achieve best results.



28.22 valve characteristic

IP_CPPWM

CPS valve characteristic

This table allows to define the valve opening (CPPWM) from two physical parameters such as the purge flow setpoint (FLOW_COR_SP_CPS) and the upstream/downstream differential pressure applied (PRS_CPS).

The calibration values are gained from the flowbench measurements.

An excel file IP_CPPWM.XLS allows to prepare easily the flowbench results (flow = f [pressure, valve opening]) to fill up IP_CPPWM (valve opening = f [flow, pressure]).

Breakpoints recommended values for FLOW_COR_SP_CPS axe:

- The breakpoint values allow to avoid to spent to much time at the beginning of Ramp Open mode (small opening).
- The two first breakpoints have to following rule:
- The first breakpoint will be zero (0 kg/h). The entire raw (valve opening) will be filled up with = 0 % whatever the differential pressure PRS_DIF_CPS

The second breakpoint will be a small purge flow value like 0.05 kg/h. The raw (valve opening) will be filled out with the “start to open” beginning of flow depending on the differential pressure PRS_DIF_CPS.

[-]

IP_CPPWM_VB_ADD_1

Additive VB correction for CPPWM at frq 1

[-]

IP_CPPWM_VB_ADD_2

Additive VB correction for CPPWM at frq 2

This additive battery correction is applied to adapt the valve opening due to battery voltage variation. The value is depending on the frequency selected.

IP_CPPWM_VB_FAC

Multiplicative VB correction for

CPPWM

This factor correction is applied to adapt the valve opening due to battery voltage variation. The value is depending on the type of purge valve selected.

[1]

28.22.1 Frequency management

C_N_FRQ_1_CPS **C_N_FRQ_2_CPS** **C_CPPWM_FRQ_1_CPS** **C_CPPWM_FRQ_2_CPS**	as required or specified

28.22.2 Thresholds on engine speed and CPPWM to shift CPS frequency.

C_N_FRQ_HYS_CPS
C_CPPWM_FRQ_HYS_CPS

Hysteresis thresholds on engine speed and CPPWM to shift CPS frequency.

These parameters allow to get stable frequency switching.

The hysteresis value should be twice the engine speed and opening valve fluctuations.

[-]

C_CPPWM_FRQ_1
C_CPPWM_FRQ_2

First CPS output frequency

These calibrations will define the valve frequency value applied depending on engine speed and purge valve opening. The selection of the frequency allows to control the purge flow with a minimum of random variation.

The intake manifold back pressure due to engine valves activities could disturb the pulsed purge flow. Then the purge fuel control becomes out of the target.

[10]

IP_CPPWM_COR_FRQ	PWM correction following frequency value This factor correction is applied on the purge valve opening defined for the reference frequency control (FREQ1). From these measurements calculate the ratio between the purge valve opening at the different frequencies and for the same conditions (purge flow, pressure). Calculate for each opening value the best ratio to satisfy most of cases.

[1]

28.23 EVAC emission control variables (403H)

28.23.1 Activation, engine speed limit

LC_CP_ACT	Switch for EVAC activation This switch enables / disables the canister purge LC_CP_ACT=0: canister purge deactivated
C_N_MAX_CP	Maximum engine speed value for canister purge calculations (CPU load reduction)
C_N_HYS_MAX_CP	Hysteresis value for canister purge calculations (CPU load reduction)
C_TCO_MIN_CP	Minimum temperature for EVAP-Control active The minimum threshold of the engine temperature for which the engine combustion has no instability while fuel is supplied by the canister purge system.

LC_CP_DYW_CLC_MAX_ENA	can be set to 0 if further CPU load reduction at high engine speed is needed
C_MAF_DYW_CP **C_N_DYW_CP** **C_FAC_LAM_CP_DYW** **C_DLY_N_MAF_DYW_CP** **C_TQI_DYW_CP** **C_N_2_DYW_CP** **C_LAMB_SP_S_DYW_CP** **C_LAMB_SP_S_DYW_CP** **C_LAMB_SP_S_DYW_CP** **C_LAMB_SP_S_DYW_CP** **C_DLY_N_TQI_DYW_CP**	Dynamic window constants defining stable engine conditions for canister purge function during homogeneous combustion. When stable engine conditions are reached, a timer is started for C_DLY_N_MAF_DYW_CP (hom.) rsp. C_DLY_N_TQI_DYW_CP (strat., hom.lean) When the delay is elapsed the CL_MMV calculation is authorized. The time delay allows the lambda controller to be centered again. These calibrations have to be refined depending on the standard driving cycle used. The transient time duration will define the values. *Method*: • engine bench or Chassis Dyno • stable engine conditions • canister purge function not active • Choose 6 engine load points, give preference to the stable phases of the emission cycle. • For each engine load measure the deviation and variation(MIN and MAX) of the engine speed (N) and mass air flow value (MAF) (hom.) rsp. torque set point value (TQI_SP_S) (lean). • The dynamic window will be define using the formula [3 * (max-min) / 2] • The division by 2 is the result of the specification description where the dynamic window is define as half window. • The time delay to set the stable condition information is tune on the emission cycle in order to filter the lambda deviations during transients. The condition engine stable must be active when lambda deviations (without purge) are elapsed.

C_CRLC_MAF_DYW_CP	Filter constant for limited dynamics calculation on MAF [0,1]
C_CRLC_N_DYW_CP	Filter constant for limited dynamics calculation on engine speed [0,1]
C_CRLC_LAM_DYW_CP	Filter constant for limited dynamics calculation on lambda controller controller in non-normal purge [0,1]
C_CRLC_LAMB_SP_S_DYW_CP	Filter constant for limited dynamics calculation on lambda setpoint in stratified [0,1]
C_CRLC_TQI_DYW_CP	Filter constant for limited dynamics calculation on tqi_sp in stratified [0,1]

28.23.2 Preselection of the canister purge state

C_STATE_CDN_CP_MAN	manual selection of STATE_CDN_CP
C_PRS_MAX_CPS	[Barometric pressure – Manifold intake pressure] limit of the difference of the pressure applied for activation of the canister purge One condition of NO_PURGE is AMP – MAP < C_PRS_MAX_CPS. The value C_PRS_MAX_CPS = 0 enables the MAX_PURGE mode when MAP > AMP (case of supercharger engine) [0 mb]
C_PRS_MAX_CPS_2	To prevent backflow from intake manifold in the acf with charged engines, this threshold leads to a direct shut of the canister purge valve (no ramp) [AMP – MAP] < C_PRS_MAX_2_CPS disables close loop operation [10 mb]
C_STATE_WIN_CP_NOT	state to be forced if engine operation outside CP window; selection of "NONE" leads to an exiting of RAMP_OPEN, but not of RAMP_OPEN_FAST or MAX_PURGE

C_STATE_LAM_LIM_CP	state to be forced if lambda control limitation reached .
LC_STATE_CDN_CP_MAN	switch to activate test mode // manual setting of STATE_CDN_CP
C_TIA_MIN_CP	Minimum intake air temperature to enable canister purge function at idle. One of the condition of NO_PURGE is TIA < C_TIA_MIN_CP. Freezing conditions or acoustic problems can appear at idle. This value has to be checked during cold condition tests. The freezing conditions are depending on the location of the valve mounting. The acoustic problems are depending on the valve mounting support. [5° C]
C_LAMB_SP_S **C_LAMB_SP_S_CP_TOL** **_CP_BOL**	Bottom and top limits on the LAMB_SP_S value [1,5] [2,5]

C_STATE_MAN_SDL_CP	selection of manual scheduling mode [AUTOMATIC]
NC_NR_CP_LAM_AD **ID_T_LAM_AD_THD** **ID_DUR_LAM_AD**	Number and arrays [NC_NR_CP_LAM_AD elements] for start time and duration of fixed lambda adaptation slices
C_CL_MMV_T_BOL **C_CL_MMV_T_TOL** **C_T_DI_CP**	When an empty canister is detected at the end of the learning modes RAMP_OPEN or RAMP_OPEN_FAST (empty canister meaning CL_MMV < C_CL_MMV_T_BOL), (and C_STATE_MOD_SEL_CP="MAX_PURGE") the evaporative system control is stopped for the delay C_T_DI_CP. If the canister is detected full CL_MMV>C_CL_MMV_T_TOL the time control gives the priority to the CP time request. In case of middle load the time control check the request of lambda adaptation and canister purge. This value is checked during hot trip (worst case conditions: canister full and vapor generation) and canister purge cycle. The goal is to optimize the lambda adaptation accuracy and the purge efficiency (no fuel vapor escape to the atmosphere).
C_T_MFF_AD_MAX	Initialisation time for dynamic lambda adaptation time slice. The timer counts as well while waiting for the lambda adaptation conditions to be met (in this period, canister purge may still be enabled). The lambda adaptation phase may be left if the feedback flag from lambda adaptation is set. [240 s]
C_T_2_CP	Initialisation time for dynamic canister purge time slice, will be corrected at the end of RAMP_OPEN. This value must be checked on emission cycle.

[240 s]

IP_T_MAX_CP

Dynamic canister purge time depending on canister load. At the end of RAMP_OPEN or RAMP_OPEN_FAST the purge time duration is modified to fit the canister load learnt.

IP_T_MAX_CP < C_T_2_CP.

This calibration is defined by measuring CL decrease on standard driving cycle (FTP, EURO 2000...) with worst case condition like ambient temperature... For different CL adjust the purging time duration to avoid canister purge breakthrough.

CL_MMV	[-]	0	0	0	0	1	1
IP_T_MAX_CP	[s]	60	90	200	230	260	280

28.25 EVAC flow setpoints (9076)

C_MFF_MIN_CP	Minimum limit on mass fuel flow The purge flow fuel setpoint is calculated to not lead to a remaining injector fuel mass setpoint falling below this limit. [5mg/tdc]
IP_FLOW_CPS_HOM **IP_FLOW_CPS_HOM_AFL**	These values are the maximum canister purge flow in MIN_PURGE mode and homogeneous lambda 1 rsp. hom. lean combustion.

This ratio will limit the purge valve opening because no fuel compensation is done in this mode.

ΔLambda CP is the maximum lambda control deviation allowed to avoid driveability trouble (acceleration) when the lambda controller is not centered. This maximum deviation is obtained with charcoal canister full loaded. The recommended value of ΔLambda CP is about 5-6 % maximum.

From this maximum deviation the maximum purge flow can be evaluated for each engine load :

- Canister Load = (Fuel Mass CP) / Air Flow CP
- Air Flow CP = (Air Flow cylinder * ΔLambda_control CP) / $K_{stochiometric}$ * CLmax

We know the maximum Canister Load ≈ 2 and the usual $K_{stochiometric}$=14,57

- Air Flow CP = Air Flow cylinder * 0,00206

Nevertheless, we have to consider that for engine conditions like “High Load” and charcoal full loaded canister, the upstream/downstream differential pressure will be too low to generate sufficient purge flow then the lambda controller deviation will be also too low.

Method:

For each engine condition (N,MAF):

- Consider the maximum flow the purge valve can provide: IP_FLOW_CPS
- **if** IP_FLOW_CPS__PRS_CPS ≥ Air Flow cylinder * 0.00206

  then fill up IP_FLOW_CPS_HOM with Air Flow$_{cylinder}$ * 0.00206

  else fill up IP_FLOW_CPS_HOM with IP_FLOW_CPS

  endif

IP_FLOW_CPS_S

flow setpoint for MIN_PURGE / stratified operation

Method:

For stratified engine condition (N,MAF):

- Engine bench
- Bottle of gas (butane)
- Perform different opening of the valve with a canister full loaded and measure the impact on the torque.
- The calibration value will be the one in the limit of the driveability (ex 5Nm additional torque).
- The map is tuned using the canister purge valve flow FLOW_CPS

Only available in DI version

IP_RATIO_FUEL_MAX_CP_HOM

Maximum fuel flow ratio (fuel flow through the CPS/ total fuel flow) for homogeneous stoiciometric operation

This map allows to adapt the purge valve opening in order to purge with the maximum efficiency when the CL is not at the maximum value. These calibrations are used in RAMP_OPEN, RAMP_OPEN_FAST and MAX_PURGE modes.

These calibrations define the maximum purge fuel quantity regarding the fuel engine needs.

To avoid driveability troubles the current ratio limitation between purge fuel and total engine fuel is 25%.

- Fuel purge flow = In system air flow / K $_{stochiometric}$ * X% **{1}**
- Fuel purge = Total cylinder flow * FLOW_CPS * CL $_{max}$ **{2}**

- X% is the fuel purge/engine total fuel ratio (given by engine bench)
- FLOW_CPS is defined by IP_FLOW_MAX_CPS
- Cl$_{max}$ = 0.7 (maximum value with butane)

Method:

- Calculate for each engine load defined by [MAF, N] the values **{1} and {2}**
- Final calibration value for each engine load defined by [MAF, N] = MIN [**{1}, {2}**]
- X%: start with 20% and following the hot trip or SHED results the percentage could be increase (less than 35%).

IP_RATIO_FUEL_MAX_CP_HOM_AFL

Only available in DI version

Maximum fuel flow ratio (fuel flow through the CPS/ total fuel flow) for homogeneous lean operation

These calibrations allow to adapt the purge valve opening in order to purge with the maximum efficiency when the CL is not at the maximum value. These calibrations are used in RAMP_OPEN, RAMP_OPEN_FAST and MAX_PURGE modes.

These calibrations define the maximum purge fuel quantity regarding the fuel engine needs.

To avoid driveability troubles the current ratio limitation between purge fuel and total engine fuel is 25%.

- Fuel purge flow = In system air flow / K $_{stochiometric}$ *lambda* X% **{1}**
- Fuel purge = Total cylinder flow * FLOW_CPS * CL $_{max}$

{2}

- X% is the fuel purge/engine total fuel ratio
- FLOW_CPS is defined by IP_ FLOW_MAX_CPS
- Cl_{max} = 0.7 (maximum value with butane)

Method:

- Calculate for each engine load defined by [MAF, N] the values **{1} and {2}**
- Final calibration value for each engine load defined by [MAF, N] = MIN [**{1}, {2}**]
- X%: start with 10% and following the hot trip or SHED results the percentage could be increased.
- If the lean burn combustion is not used the map is tuned to 0.

IP_RATIO_FUEL_MAX_CP_S	Only available in DI version Maximum fuel flow ratio (fuel flow through the CPS/ total fuel flow) for homogeneous stratified operation These calibrations allow to adapt the purge valve opening in order to purge with the maximum efficiency when the CL is not at the maximum value. These calibrations are used RAMP_OPEN, RAMP_OPEN_FAST and MAX_PURGE modes. These calibrations define the maximum purge fuel quantity regarding the fuel engine needs. To avoid driveability troubles the ratio is limited to take into account the torque. X%: start with 10% and following the hot trip or SHED results the percentage could be increase. If the stratified MAX_PURGE is not used the map is tuned to 0. [0,25]
IP_RATIO_FUEL_MAX_CP_	This table is corrective factor acting on

COR	MFF_KGH_MAX_CP calculation. The aim for tuning is to allow activation of the purge function at a lower TCO. In this case (that means at low TCO), IP_RATIO_FUEL_MAX_CP_COR allow to less open the CPS. This correction has been added in case of long warm up phase, to win purge efficiency by opening the CPS in advance.

28.26 Lambda deviation and canister load (9077)

28.26.1 Filter constants

IP_CRLC_FAC_LAM_MV_MMV_CP_CLL

correlation constant for FAC_LAM_MV_MMV_CP calculation in case of Purge in closed loop

MAF_CYL	[kg/h]	12	45	70	140	250	5
IP_CRLC_FAC_LAM_MV_MMV_CP_CLL	[-]	0,500	0,500	0,500	0,500	0,500	0

IP_CRLC_FAC_LAM_MV_MMV_CP_OPL

correlation constant for FAC_LAM_MV_MMV_CP calculation in case of Purge in open loop

MAF_CYL	[kg/h]	12	45	70	140	25
IP_CRLC_FAC_LAM_MV_MMV_CP_OPL	[-]	0,250	0,250	0,250	0,250	0,2

IP_CRLC_LAMB_LPF_CP

Correlation constant for LAMB_MMV_CP calculation

IP_CRLC_MMV_CP

Correlation constant for LAMB_MMV_CP calculation

IP_CL_CRLC_RAMP_OPEN

CL_MMV filter coefficient during RAMP_OPEN_FAST

MAF_CYL	[kg/h]	12	45	70	140	250
IP_CL_CRLC_RAMP_OPEN	[-]	0,300	0,300	0,400	0,450	0,480

IP_CL_CRLC_OPEN_FAST

When the RAMP_OPEN_FAST is ongoing the Lambda controller has to be stable (no more than 5% of deviation)

This calibration result is also depending on ramp opening value during this mode.

IP_CL_CRLC_INI

Initialization value for correlation factor value

CL_MMV	[-]	0	0	0
IP_CL_CRLC_INI	[-]	0,150	0,150	0,150

IP_CL_CRLC

This calibration is the CL_MMV filter coefficient during MAX_PURGE mode. The values will define this filter coefficient will be limited by two borders. The high one is defined by the Canister load decrease in stable engine conditions, and the low one is defined by the normal acceleration/deceleration time duration

case A, canister load decreases in stable engine conditions

Method:

- IP_CL_CRLC has to be tuned to follow the canister load (CL) decrease.
- During MAX_PURGE, CL decreases and the canister weight loss follow the same law.
- A wrong IP_CL_CRLC value could result in a meaningful difference between the average CL value (CL_MMV is used for the calculation of injection compensation) and CL.

Test conditions:

- Different engine loads in stable conditions
- Charcoal canister full loaded (butane or fuel following legal procedure).
- C_T_1_CP, C_T_2_CP and IP_T_MAX_CP_CL_MMV set to a high value to keep MAX_PURGE mode until the canister becomes empty.
- IP_CL_CRLC pre-tuned to 0,005 (for the whole mapping).
- recording of CL
- if canister is loaded with butane: CL_MAX ≈ 0.63
- if canister is loaded with fuel: CL_MAX ≈ 0.7....2

case B, normal acceleration/deceleration time duration

During transient mode the Lambda controller varies. The time duration of the lambda controller is shorter than the one during the charcoal canister weight lost. This lambda controller variation has not to be taken such as a loss of canister load.

- Different engine load in stable conditions
- Purge deactivated
- start with normal acceleration/deceleration
- Measure the time spent (T2) by the lambda controller to reach a stable

value {2}

Calculate the values (CL max * recurrence) / T2 s {2}

The final calibration value will be: {2} < IP_CL_CRLC < {1a, 1c}

Usual approach:

<u>Part and full load</u>

IF for MAF_CYL ≥ 90 kg/h the value calculated in {1a} > 0.005 then fill up the mapping table with 0,005

And for MAF_CYL < 90 kg/h fill up the mapping table with 0.005 * value calculated in {1b}

<u>Idle speed</u>

At idle the average value of the lambda controller (TI_LAM_MMV) varies. Then this variation generate another CL variation. For MAF_CYL breakpoint at idle and charcoal canister empty refine the coefficient filter to prevent these CL, CL_MMV oscillations.

IP_NR_STEP_CL	Number of step to change of correlation factor value This is a filter-filter-constant map, it aims in filtering the transition from the IP_CL_CRLC_INI map to the standard-MAX_PURGE-map IP_CL_CRLC.
C_CL_FAC_LAM_DIF_MIN **C_CL_FAC_LAM_DIF_MAX**	Lower and upper limits on FAC_LAM_DIF_CP variation If these limits are exceeded in MAX_PURGE, the CL filter constant is taken from the faster map IP_CL_CRLC_INI, same principle as entering MAX_PURGE [-10/10%]
C_PRS_MAX_2_CPS	Pressure difference limit (environment-intake manifold) for CL_MMV calculation [10 mb]

C_FLOW_MIN_CPS

Purge flow threshold to authorize CL_MMV calculation in RAMP_OPEN, RAMP_OPEN_FAST and MAX purge.

if FLOW_CPS is lower than this calibration the last value of CL_MMV is kept.

During the purge valve opening/closing, the flow FLOW_CPS takes small values. Small flow values can produce an incorrect purge flow and an incorrect canister load learning. This difference results in values of CL that do not correspond to a physical reality.

Calculate the purge flow obtained by 1 bit of valve opening.

Method: Biggest purge flow (100% opening P). Divide this maximum flow by the format of the valve opening□and high (CPPWM) in case of byte the divisor will be 255. The calibration will be the value regarding 2 bits.

[0,04 kg/h]

C_CL_ST_CP

Initialization value for charcoal canister load at the beginning of RAM_OPEN mode.

CL_MMV and CL use the same initialization value

The calibrated value is about the half value of the maximum CL value encountered when the charcoal canister is loaded with butane. This value allows to reduce the work of the CL filter and to get too much deviation between CL and CL_MMV.

[0,3]

C_CL_MAX_CP

The canister load is the fuel concentration of the A/F mixture that comes from the canister through the CPS valve.

The mass of fuel (vapors) which is coming from the canister depends on:

- the type of fuel used
- the system design (pressure drop...)

Charcoal canister loaded with butane

- The measurements should lead to the theoretical value CL max = 0,7

Charcoal canister loaded with gasoline

- The fuel from the canister is in the

	vapor state but the mass fuel flow is higher than the butane. This mass is depending on the type of fuel used and the in-use conditions. The C_CL_MAX_CP value equal to 2 could satisfied most of applications. Check this value during refueling in hot trip conditions (high vapor generation that can lead to a full and unstable canister load)
C_FAC_LAM_DIF_MAX_CP	Maximum Lambda controller deviation threshold during learning mode. This calibration is defined by the maximum lambda controller deviation to get no driveability trouble during accelerations while the RAMP_OPEN mode is ongoing. [5%]
C_FAC_LAM_DIF_CP_AS	Application system value to apply a lambda deviation
LC_LAM_CBK_CPS	Lambda controller selection (0 -> bank 0 , 1 -> bank1)
C_RAF_CLC_CP	Stochiometric ratio of usual fuel. This value has to be refined regarding the type of fuel defined by the application. [14,57]
LC_CL_FIL_INI_TRA_DEAC_CLL	Correlation factor calculation in case of NO_PURGE or MIN_PURGE → MAX_PURGE transition
C_CL_DYW_MAX_CP	This calibration aims to prevent wrong fuel correction when too fast variation of CL happens. This calibration is to get fuel correction robustness. In case where too high CL variation happen the MIN_MODE is forced. In RAMP_OPEN, RAMP_OPEN_FAST, MAX_PURGE modes [CL n-1 - CLn] is calculated. *if [CL n-1 - CLn] > C_CL_DYW_MAX_CP* *then MIN_MODE is forced*

28.26.2.1 DI specials

C_CL_MMV_CHK	The canister purge force the homogeneous lambda 1 combustion to increase the capacity of purge, when the canister load is under the threshold C_CL_MMV_CHK the stratified combustion is active again. *Method:* On specific emission cycle (Ex : MAZDA), check if the cycle if finished During hot trip and fuel vapor generation, check the capacity of the function to prevent gas to the vent of the canister.
C_CL_MMV_HOM_CP	The canister purge force the homogeneous lambda 1 combustion to increase the capacity of purge, when the canister load is upper the threshold C_CL_MMV_HOM_CP the homogeneous lambda 1 combustion stay active. *Method:* On specific emission cycle (Ex : MAZDA), check if the cycle if finished During hot trip and fuel vapor generation, check the capacity of the function to prevent gas to the vent of the canister.
LC_LAMB_AFS_CP_CPS_SCG	switch to force hom1 operation if cps valve short circuit to ground error present

28.26.3 Timer constants for limits exceeded tests

C_T_DLY_PU_CPS	CPS interrupt time during PU When the engine management goes in trailling throttle (LV_PU) the injection time becomes very small and near the minimum injection time. To prevent a big lambda deviation on transition PL to PU the CPS valve is closed immediately during a time C_T_DLY_PU_CPS. When this time is elapsed the function stay in MIN_PURGE during PU. The time must allow to close the valve during the gas transfer between the CPS

	valve and the oxygen sensor. This time is defined by the duration of the engine gas transfer. [0.1 s]
IP_T_DLY_CMU_CPS	Stabilization delay on lambda regulation after commutation When the engine have combustion commutations (e.g. homogeneous → stratified), all the parameters are unstable during a delay. To prevent driveability problems during the commutation the canister purge valve is closed during a time IP_T_DLY_CMU_CPS__N_32. The function is re activated as soon as the engine parameters are stable. Method: Perform different type of commutations and measure the time to have stable values
C_T_LAM_LIM_CP	Additional close loop deactivation duration due to LV_LAM_LIM activation. Set, when the lambda controller sets its flags indicating "lambda control limitation reached" [5 s]
C_FAC_LAM_CP_BOL **C_FAC_LAM_CP_TOL**	Lambda control functional window. This window is additional to the control of the limiation flags from the lambda controller, see above. To prevent wrong canister load learning, the lambda control value will be checked all the time in RAMP_OPEN. If C_FAC_LAM_CP_BOL < FAC_LAM_CP < C_FAC_LAM_CP_TOL, RAMP_OPEN is disabled or stopped temporarily and MIN_PURGE is applied. This calibration prevents the learning because the reference point FAC_LAM_0_CP is not centered at RAMP_OPEN start. The threshold is defined by the normal error due to the flows (air & fuel). The realistic error is estimated about 5%. The 5% standard lambda controller deviation during the learning mode is defined by C_FAC_LAM_DIF_MAX_CP. These calibrations are defined by: • standard error flow +

	C_FAC_LAM_DIF_MAX_CP [-10%]
C_T_DLY_LAM_CP	Time to inhibit learning modes RAMP_OPEN & RAMP_OPEN_FAST after lambda controller left functional window defined above. After the time delay C_T_DLY_LAM_CP the learning conditions will be checked once more. [20 s]
C_N_CP_BOL	Minimum speed limit for activation of the RAMP_OPEN operation (homogeneous) The low limit of close loop is used as value [620 1/min]
C_N_CP_TOL	Maximum speed limit for activation of the RAMP_OPEN operation (homogeneous) This parameter is defined using the measurement of the MAP = f [MAF,N] The difference of the pressure (AMP – MAP) has to be higher than 200mbar in the area defined by C_N_CP_TOL and IP_MAF_CP_TOL. [3500 1/min]
IP_MAF_CP_BOL **IP_MAF_CP_TOL**	Minimum mass air flow limit for activation of the RAMP_OPEN operation (homogeneous) In low MAF, the purge flow is not big enough to make change the canister saturation degree. The low limit of close loop is used as value Maximum mass air flow limit for activation of the RAMP_OPEN operation (homogeneous) This parameter is defined using the measurement of the MAP = f [MAF,N] The difference of the pressure (AMP – MAP) has to be higher than 200mbar in the area defined by C_N_CP_TOL and IP_MAF_CP_TOL. ***Hint:*** The values defined for the window have an impact on the calibration of the chapter 6.2.1 These four calibrations will allow to select

the canister load learning area during homogeneous combustion

[50 mg/stroke] [400 mg/stroke]

28.26.4 Canister load learning window stratified

Minimum / maximum engine speed / torque setpoint value limits for activation of the RAMP_OPEN operation (stratified)

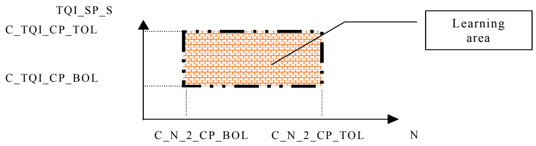
C_N_2_CP_BOL

C_N_2_CP_TOL

C_TQI_CP_BOL

C_TQI_CP_TOL

These four calibrations will allow to select the canister load learning area during stratified combustion



28.27 EVAC control (9008)

28.27.1 EVAC control - state machine

C_T_DLY_MAX_CP	Time between two MAX_PURGE operations for RAMP_OPEN_FAST activation. When leaving MAX_PURGE, a time counter T_DLY_MAX_CP starts to count backwards from this calibration value to 0. While the counter does not reach zero, a new learning mode via RAMP_OPEN_FAST is possible; if it reaches zero, the next learning mode has to be RAMP_OPEN [50 s]
C_T_DI_RAMP_OPEN_CP **C_T_DI_RAMP_FAST_CP** **C_T_DI_MAX_CP**	Disable times of the states RAMP_OPEN, RAMP_OPEN_FAST and MAX_PURGE If closed loop canister purge ("normal purge") is abandoned e.g. due to high engine dynamic, the time counter T_DI_CP starts to count backwards from the corresponding calibration values to 0. While it does not reach zero, a re-entering in the last closed loop state is possible as soon as the conditions for closed loop canister purge are given again. If the counter reaches 0, closed loop canister purge has to start with a new learning mode. This does not affect the exit of closed loop canister purge by LV_CP_NEW_RAMP_OPEN. A re-entry in RAMP_OPEN_FAST is only possible, if LC_CP_TRAN_DEAC_CLL is set to 1, otherwise (switch=0), no reversible interruption of RAMP_OPEN_FAST is enabled.
C_T_DLY_CP	Time delay to wait in WAIT_RAMP_OPEN (valve closed) before starting the learning modes. This time duration is defined by: • Duration for the engine gas

	transfer - Duration for lambda controller stabilization This value has to be checked in idle speed mode coming from MIN_PURGE mode and canister full loaded. The lambda deviation authorized in MIN_PURGE is 5% (see C_FLOW_MIN_CPS). [5 s]
C_T_LOCK_CP	Locking time for locking MIN_PURGE or NO_PURGE after canister empty detection. [20 s]
C_STATE_MOD_SEL_CP	If the canister is detected empty at the end of RAMP_OPEN, closed loop canister purge can be inhibited for a time C_T_LOCK_CP by setting C_STATE_MOD_SEL_CP to "NO_PURGE" or "MIN_PURGE". Then the so selected state is the maximum possible CP state until the timer T_LOCK_CP reaches zero. [MIN_PURGE]
LC_CP_TRA_MAX_MIN_CDN_ENA	enable transition MAX_PURGE → MIN_PURGE with STATE_CDN_CP
LC_CP_TRA_MIN_PURGE_DFT_ENA	enable transition NO_PURGE → MIN_PURGE with STATE_CDN_CP = NO_CDN
LC_CP_TRA_MIN_RAMP_OPEN_DIS	enable trans. NO_PURGE → MIN_PURGE, if no RAMP_OPEN or RAMP_OPEN_FAST is possible although STATE_CDN_CP=RAMP_OPEN or RAMP_OPEN_FAST [1]
LC_CP_TRA_OPL_CDN_CLL_NOT	enable transition WAIT_RAMP_OPEN → NO_PURGE if LV_CP_CDN_RAMP_OPEN, _FAST no more given [0]
LC_CP_TRA_OPL_CDN_NOT	enable transition WAIT_RAMP_OPEN → NO_PURGE

if STATE_CDN_CP = NO_CDN

This switch is redundant to LC_CP_TRA_OPL_CDN_CLL_NOT

[0]

28.27.2 EVAC control - states

IP_PQ_CP_SP	[-]
C_CRLC_PQ_SP_CP	[0,25]
C_CL_MMV_PQ	[2] Setpoint value, filter constant and activation threshold on canister load for forced PQ (pressure quotient) during stratified mode for RAMP_OPEN and MAX_PURGE. A forced PQ is enabled in homogeneous lean burn mode if the switch LC_PQ_SP_CP_HOM_AFL is set to 1. The PQ setpoint can lead to further closing of the throttle thus leading to a higher pressure difference ambient ⇔ intake manifold thus enabling higher purge flows.
LC_PQ_SP_CP_HOM_AFL	Setpoint value, filter constant and activation threshold on canister load for forced PQ (pressure quotient) during stratified mode for RAMP_OPEN and MAX_PURGE. A forced PQ is enabled in homogeneous lean burn mode if the switch LC_PQ_SP_CP_HOM_AFL is set to 1. The PQ setpoint can lead to further closing of the throttle thus leading to a higher pressure difference ambient ⇔ intake manifold thus enabling higher purge flows. [0]
C_CL_CAT_DIAG	Canister load threshold to enable cat.diag. in MIN_PURGE (if cl < thd) If CL_MMV is below this threshold at the regular end of RAMP_OPEN, the flag LV_CL_CLC_AVL is set to 1 and passed to EGTR where it could enable

a catalyst diagnosis in MIN_PURGE.

[0,5]

C_CL_MMV_DYW

Dynamic window on CL_MMV value to stop leaning mode (RAMP_OPEN) and start MAX_PURGE.

- The window is tuned to stop the RAMP_OPEN when the CL_MMV becomes stable inside the natural oscillation of FAC_LAM_MMV.
- This tuning is done after IP_CL_CRLC_RAMP_OPEN
- The tuning is done with empty canister (CL_MMV < 0,2) in idle (maximum effect by natural oscillation of lambda regulation).

The window is defined as twice the value of CL_MMV oscillations.

[0,024]

IP_T_DLY_CL_MMV_DYW

Minimum time duration of RAMP_OPEN when CL is inside the dynamic window C_CL_MMV_DYW

Method:

- Charcoal canister empty
- Different stable engine load defined by MAF_CYL
- Start with RAMP_OPEN mode and measure the time spent to get CL_MMV stable and close to CL. Fill the IP_T_DLY_CL_MMV_DYW with the measured time.

MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_T_DLY_CL_MMV_DYW	[s]	3,600	3,600	2,400	2,400	2,400	2,400

C_T_DLY_MAX_RAMP_OPEN

Initialisation time duration for RAMP_OPEN mode.

If no lambda shift occurs during RAMP_OPEN mode, no re-initialsisation of the timer T_RAMP_OPEN_STAB will occur and RAMP_OPEN will be left after C_T_DLY_MAX_RAMP_OPEN.

Method:

- Charcoal canister empty

- Idle speed mode

The final valve opening is defined by FLOW_MAX value. Measure the time spent to reach about 4/5 of FLOW_MAX at idle.

[20s]

LC_LAM_LIN_CPS	Type of O2 sensor (0 → binary sensor , 1 → linear sensor)

28.27.3 EVAC control - flow calculations

LC_FLOW_CPS_AS	as required
C_FLOW_CPS_AS	Manual value for FLOW_CPS and activation switch for that.
LC_CP_FLOW_LIM_OPEN_FAST	enable old calc. of FLOW_OPEN_CPS: fuel mass based As soon as the limit flow FLOW_OPEN_CPS is reached, RAMP_OPEN_FAST is left towards MAX_PURGE. With the switch = 1, the limit flow FLOW_OPEN_CPS can be taken from a map IP_FLOW_MAX_OPEN_CPS, this is similar to the calculation in RAMP_OPEN. With the switch = 0, the limit flow is calculated based on the maximum possible fuel for RAMP_OPEN_FAST; this is the solution of previous canister purge versions.
ID_FLOW_INC_NR_CP	Increment number for FLOW_CTL_CPS calculation during RAMP_OPEN and RAMP_OPEN_FAST (steps à 20ms) The ramping-up of the flow during RAMP_OPEN and RAMP_OPEN_FAST consists of two phases: flow incrementation and gas transfer delay. Each flow incrementation phase consists of a ID_FLOW_INC_NR_CP incrementation steps with the amount of IP_FLOW_INC_CP rsp. IP_FLOW_INC_FAST_CP. After the incrementation phase, the system waits for gas delay or rich → lean

switch of a binary lambda sensor.

N_32	[1/min]	0	2048	4096	5120
ID_FLOW_INC_NR_CP	[-]	11	12	15	19

IP_FLOW_INC_CP

During RAMP_OPEN, the valve is opening smoothly in order to control the lambda

This calibration defines the slope of FLOW_CPS by incrementing directly FLOW_CTL_CPS in homogeneous mode. The first step is defined by C_FLOW_INI_CPS.

The IP_FLOW_INC_CP values are defined by calculation. These increments will be checked during driving tests in hot conditions (charcoal canister loaded)

(Lambda max = 5-6 %, CL max = 2)

Method:

- Driving tests in hot conditions
- Different engine load for stable engine conditions
- Charcoal canister full loaded
- Ramp open mode
- Calibrate the maximum canister purge time equal to 20 s
- Calibrate the lambda adaptative duration equal to 20 s
- For different engine loads (MAF_CYL) adjust IP_FLOW_INC_CP in order to get no more than 5-6% of lambda variation during RAMP_OPEN mode.

MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_FLOW_INC_CP	[kg/h]	0,002	0,003	0,005	0,006	0,009	0,013

IP_FLOW_INC_FAST_CP

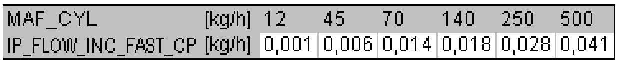
During RAMP_OPEN_FAST the CL is known so the purge valve can be opened faster than the one used in RAMP_OPEN mode. In this case the slope of the purge valve opening will be a compromise between opening speed and lambda control. This map is used for homogeneous combustion.

This calibration defines the slope of FLOW_CPS by incrementation of FLOW_CTL_CPS in Ramp Open fast.

Method

- Engine bench or Chassis Dyno
- Different engine loads for stable engine conditions
- Charcoal canister full loaded by butane or warmed fuel (50°C)
- Ramp open fast in Homogeneous mode
- For different engine loads (MAF_CYL) adjust IP_FLOW_INC_FAST_CP in order to get ± 5-6 % maximum lambda variation during RAMP_OPEN mode. This calibration needs some repeated measurements. To get the repeatition fastly the time durations of some phases could be reduced (Lambda adaptation, Purge time duration)

These calibrations have to be refined during in-use tests in hot conditions (summer trip)



MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_FLOW_INC_FAST_CP	[kg/h]	0,001	0,006	0,014	0,018	0,028	0,041

IP_FLOW_DEC_CP

Purge opening decrement value for FLOW_CPS during RAMP_CLOSE mode. The slope of the closing prevents exhaust lambda variation.

This calibration defines the slope of FLOW_CPS.

Method:

- Charcoal canister full (legal procedure) and empty loads
- Canister Load learned in RAMP_OPEN and refined in MAX_PURGE.
- For different engine loads (MAF_CYL) adjust IP_FLOW_DEC_CP in order to get

± 5% maximum lambda variation at the end of RAMP_CLOSE mode. This calibration needs some repeated measurements. To get the repetition fastly the time durations of some phases could be reduced : Lambda adaptation, Purge time duration....

C_FAC_DEC_COR_CMU_CP

Purge opening decrement value for FLOW_CPS during combustion commutation.

This calibration defines the slope of FLOW_CPS.

Method:

- Charcoal canister full (legal procedure) and empty loads
- Canister Load learned in RAMP_OPEN and refined in MAX_PURGE
- Adjust C_FAC_DEC_COR_CMU_CP in order to have no impact of the canister purge during the commutation. Compare with and without purge different types of commutation.

C_FAC_DEC_COR_1_CP

Other system functions could want to stop the canister purge function. The request of purge closing is controlled by the OBD2 controller. Depending on the type of request the closing speed is defined by C_FAC_DEC_COR_1_CP.

The target of the calibration is to go in NO_PURGE mode fastly without taking care with pollution or lambda variation.

Closing speed correction factor 1 is used for high speed closing

This calibration has to be done by the function transmitter of the close request

[1]

C_FAC_DEC_COR_2_CP

Other system functions could want to stop the canister purge function. The request of purge closing is controlled by the OBD2 controller. Depending on the type of request the closing speed is define by C_FAC_DEC_COR_2_CP.

The target of the calibration is to go in NO_PURGE mode with a compromise between the closing speed and lambda variation.

Closing speed correction factor 2 is used for medium speed closing

This calibration has to be done by the function transmitter of the close request

[1]

IP_FLOW_FAC_CP	Correction factor for FLOW_CPS applied in RAMP_OPEN, RAMP_OPEN_FAST or RAMP_CLOSE modes, for homogeneous and stratified modes. The correction value is depending of the CL value. The reference is based on canister full loaded (factor = 1). When the CL is lower the FLOW_CPS incrementation is increased.

The values are defined by calculation. The values will be checked during driving tests with different charcoal canister loads.

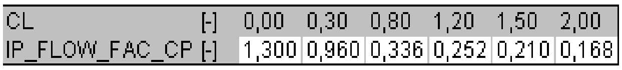
This factor must be calibrated by taken account that at each incrementation the lambda variation must stay lower than 5-6%

Method:

- Driving tests, different charcoal canister loads
- Different engine load for stable engine conditions
- Ramp open mode

For different engine loads (MAF_CYL) adjust IP_FLOW_FAC_CP in order to control ± 5-6 % maximum lambda variation. This calibration needs some repeated measurements. To get the repeatition fastly the time durations of some phases could be reduced : Lambda adaptation, Purge time duration....

The calibration must be checked in Ramp open fast and ramp close mode for all the combustion modes.



CL	[-]	0,00	0,30	0,80	1,20	1,50	2,00
IP_FLOW_FAC_CP	[-]	1,300	0,960	0,336	0,252	0,210	0,168

IP_FLOW_CPS_INC

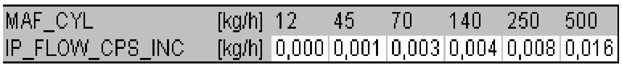
(In MAX_PURGE, the closing of the valve must be the same as the one given by the ramp of FLOW_CTL_CPS during RAMP_CLOSE.)

The purge valve opening/closing is done by FLOW_CTL_CPS variations.

In case where the evaporative system design is not good enough (fuel tank purged : the purge valve opening/closing has to be done slower. These calibrations allow to apply a 1st order on the purge flow needed. The purge flow integrated will define the purge valve opening.

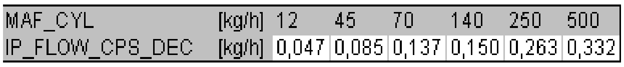
Method:

- Engine loads in stable conditions
- Charcoal canister full (legal procedure) and empty load
- CL learnt and MAX_PURGE mode acting.
- Change the purge target modifying IP_FUEL_FLOW_MAX_CP to get high variation in purge valve opening and purge flow needed.
- For different engine loads (MAF_CYL) adjust the INC and DEC in order to get ± 5% maximum lambda variation up to the valve opening setpoint is reached. This calibration needs some repeated measurements. To get the repeatition the time duration of Purge time duration could be lengthened.



MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_FLOW_CPS_INC	[kg/h]	0,000	0,001	0,003	0,004	0,008	0,016

IP_FLOW_CPS_DEC



MAF_CYL	[kg/h]	12	45	70	140	250	500
IP_FLOW_CPS_DEC	[kg/h]	0,047	0,085	0,137	0,150	0,263	0,332

C_FLOW_INI_CPS

- This value is used to initialize FLOW_CTL_CPS at beginning of the RAMP_OPEN.
- This data is calculated to obtain

most quickly the “start to open” valve opening point in idle. At the RAMP_OPEN start the purge valve opening has to be close to the “start to open” point defined in IP_CPPWM

- Method:
- Second breakpoint of purge flow axe in IP_CPPWM > C_FLOW_INI_CPS
- Check on engine bench for different engine loads that this value is enough to open correctly the valve and create e first deviation of lambda
- [0,01 kg/h]

IP_FLOW_MAX_OPEN_CPS

FLOW_CPS for fully opened CPS (CPPWM=100%)

This map defines the maximum purge flow value [kg/h] when the valve is fully opened; the values are from the flowbench measurements or from IP_CPPWM

The calibration needs the values

- MAF_CYL = f(MAF[mg/tdc], N_32[rpm]) and
- MAP = f(MAF[mg/tdc], N_32[rpm]) and

MAF_CYL ⇒ MAP ⇒ MAX FLOW CPS ⇒ IP_FLOW_MAX_CPS (MAX FLOW CPS)

An excel file FLOW_MAX.xls allows to calculate the calibration values to fill up automatically the FLOW_MAX_CP.

This map characterizes the maximum flow authorized during RAMP_OPEN or RAMP_OPEN_FAST mode. This map can be considered as flow setpoint for the learning modes.

At Idle IP_FLOW_MAX_OPEN_CPS should be limited. In order to keep sufficient idle actuator authority on purge the maximum purge flow will be limited at 10% of the maximum total air engine flow.

[-]

IP_FAC_AMP_CP

The canister purge flow ($Q_{Purge\ Map}$) values are defined at standard sea level barometric pressure ($P_{Purge\ Map}$) =1013 mb.

The factor correction by IP_FAC_AMP_CP allows to adapt the flow at the current barometric pressure.

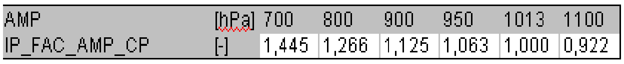
The variation of the canister purge flow is depending on the upstream pressure. This variation follows the law:

$$Q_{Purge} = Q_{Purge\ MAP} * [P_{baro\ current} / P_{Purge\ MAP}]$$

When the valve opening has to be

defined to get a needed canister purge flow the inverted law is used :

$$Q_{Purge\ MAP} = Q_{Purge} * [P_{Purge\ MAP} / P_{baro\ current}]$$



AMP	[hPa]	700	800	900	950	1013	1100
IP_FAC_AMP_CP	[-]	1,445	1,266	1,125	1,063	1,000	0,922

IP_FAC_TIA_CP

The canister purge flow ($Q_{Purge\ Map}$) values are defined at standard ambient temperature = 25°C

This factor correction allows to adapt the flow at the current ambient temperature

The variation of the canister purge flow follows the law:

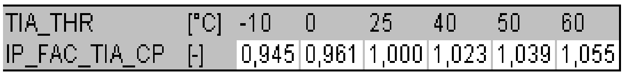
$$Q_{Purge} = Q_{Purge\ MAP} * [T_{Purge\ MAP} / T_{current}]^{0,5}$$

With T[°K]

When the valve opening has to be defined to get a needed canister purge flow the inverted law is used :

$$Q_{Purge\ MAP} = Q_{Purge} * [T_{current} / T_{Purge\ MAP}]^{0,5}$$

With T[°K]



TIA_THR	[°C]	-10	0	25	40	50	60
IP_FAC_TIA_CP	[-]	0,945	0,961	1,000	1,023	1,039	1,055

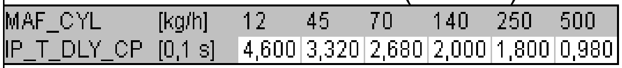
28.27.3.1 EVAC control - Padé- and Low pass filter calculations

<table>
<tr><td>IP_T_DLY_CP</td><td>
<p>This map allows to characterize the gas transfer duration between canister valve and oxygen sensor.</p>
<p>These value are used to calculate the Padé filter outputs for FLOW_CPS and MAF_CYL.</p>
<p>Method:</p>

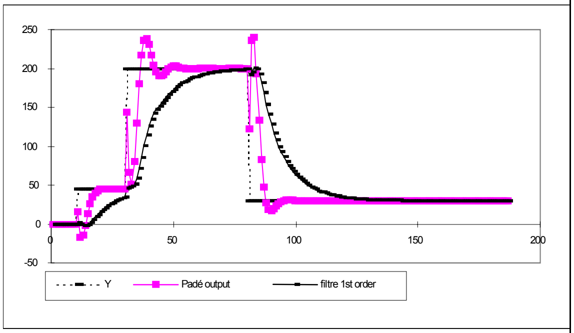
Chassis Dyno and stable engine conditions
Charcoal canister half loaded
Canister purge strategy deactivated
Opening/closing purge valve set by the application system calibration C_CPPWM_AS.

<p>For different engine load (MAF_CYL) measurement of the time duration from the purge valve switching (open/close) to the beginning of the variations of :</p>

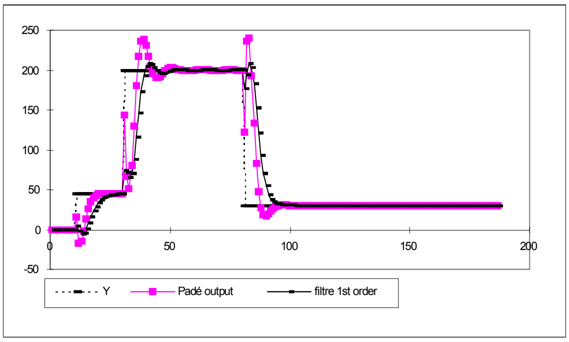
Lambda controller or
O$_2$ sensor period or
UEGO sensor

<p>The value of IP_T_DLY_CP is calculated by dividing the measured time by the calculation recurrence of the PADÉ filter (100 ms).</p>

<table>
<tr><td>MAF_CYL</td><td>[kg/h]</td><td>12</td><td>45</td><td>70</td><td>140</td><td>250</td><td>500</td></tr>
<tr><td>IP_T_DLY_CP</td><td>[0,1 s]</td><td>4,600</td><td>3,320</td><td>2,680</td><td>2,000</td><td>1,800</td><td>0,980</td></tr>
</table>
</td></tr>
<tr><td>C_FLOW_DLY_CRLC_CP</td><td>[0,3]</td></tr>
<tr><td>C_MAF_DLY_CRLC_CP</td><td>
<p>Filter coefficient of FLOW_DLY_CP and MAF_TOT_CP_DLY_.</p>
<p>These coefficients cancel the PADE output oscillations.</p>
<p>The following example using two filter coefficient values show the effect on the output.</p>
</td></tr>
</table>

Filter coefficient = 0.1 ***Value too small because the ouput value is too much delayed***



Filter coefficient : 0.3 ***Value adapted because the ouput value is close of PADE output without oscillations***



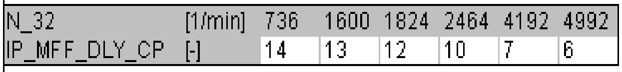
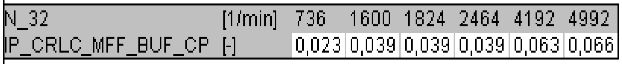
[0,3]

28.28 Injection calculation (903M)

28.28.1 Injection calculation - injection buffer

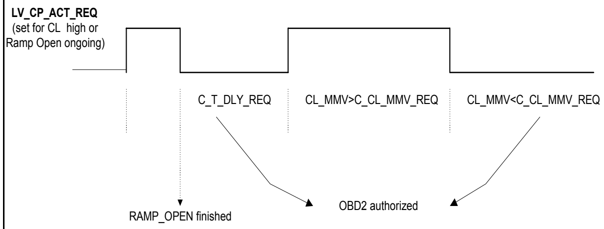
The injection correction model is characterized by a delay and a 1st order filter. The delay should represent the gas transfer time through the intake manifold, the appending filter should represent the flow dilution behaviour.

NC_NR_CP_BUF	Size of the fuel buffer MFF_ADD_BUF The buffer size has to be determined according to the expected delay times between CPS and injection valves. [20]
LC_FLOW_CPS_CAL	Choice of recurence for injection buffer calculation. The injection buffer can be calculated with two different recurrencies: 20ms rsp. 40ms, LC_FLOW_CPS_CAL leading to a recurrency of 20ms. By selecting a slower update rate, higher gas delay times can be modelled with a given buffer size. [0]
IP_MFF_DLY_CP	These calibrations allow to characterize the intake manifold gas transfer between the canister purge valve and the cylinders inlet. *Method:* • Engine bench or Chassis dyno in stable engine load conditions • Canister purge strategy deactivated • Change the injection time using the bench function (C_TI_AS) • For different engine load (N variation) measurement of the gas transfer duration from the injector valve change to lambda deviation. • The time delay (valve to injector) is the difference between the map IP_T_DLY_CP (time between the valve and the lambda sensor) and the previous measurement. • The calibration values are the time measured divided by the calculation recurrence of MFF_BUF_CP. • The recurrence is 20 or 40ms,

<table>
<tr><td></td><td>depending on calibration.
<table><tr><td>N_32</td><td>[1/min]</td><td>736</td><td>1600</td><td>1824</td><td>2464</td><td>4192</td><td>4992</td></tr><tr><td>IP_MFF_DLY_CP</td><td>[-]</td><td>14</td><td>13</td><td>12</td><td>10</td><td>7</td><td>6</td></tr></table></td></tr>
<tr><td>**IP_CRLC_MFF_BUF_CP**</td><td>The dilution behavior through the intake manifold is characterized by a 1st order filter. The filter coefficient is defined such as the exhaust lambda variations are the smallest from purging change.
Method:,
• IP_MFF_DLY_CP has to be tuned before.
• IP_CRLC_MFF_BUF_CP minimizes the lambda variation when canister fuel flow is controlled by the purge function.
• Engine bench or Chassis dyno in stable engine load conditions
• Charcoal canister half loaded
• Canister purge strategy activated in MAX_PURGE mode (CL known)
• Opening/closing purge valve set by changing the purge target (IP_FUEL_FLOW_MAX_CP)
For different engine load (N variation) measurement of the lambda variation (Linear sensor) due to purge target change. Adjust IP_CRLC_MFF_BUF_CP value before each change in order to have no exhaust lambda peak (lean or rich).
<table><tr><td>N_32</td><td>[1/min]</td><td>736</td><td>1600</td><td>1824</td><td>2464</td><td>4192</td><td>4992</td></tr><tr><td>IP_CRLC_MFF_BUF_CP</td><td>[-]</td><td>0,023</td><td>0,039</td><td>0,039</td><td>0,039</td><td>0,063</td><td>0,066</td></tr></table></td></tr>
</table>

28.28.1.1.1 EVAC control - Common submodules

C_CL_MMV_REQ	When the canister load is bigger than the threshold C_CL_MMV_REQ the canister purge function does not be interrupted by another function (OBD2.....). The calibration value is defined by

	maximum canister load with butane. Take care this calibration value does not prevent the OBD2 and adaptive functions during SHED test or standard driving cycle when the charcoal canister is full loaded. [0,5]
C_T_DLY_REQ_CP	Time to set LV_T_DLY_REQ_CP The time C_T_DLY_REQ_CP authorizes ODB2 diagnosis activation even C_CL_MMV_REQ is set. When this time duration is elapsed the canister purge load takes highest priority. This time duration is refined in accordance with the standard driving cycle.  [120 s]

28.28.2 EVAC control - Application Incidences (902C)

C_STATE_CONF_CAT_CP	[0]
C_STATE_CONF_TLDV_CP	Configuration byte for closing speed of canister purge before catalyst diagnosis / leak detection diagnosis are started. With these constants, the closing speed in case of requests from catalyst diagnosis or leak detection can be configured. Four values are possible: 0-3. "0" means no reaction on the requests, i.e. no closing of the canister purge valve. 1-3 lead to a closing of the CPS, "1" with high speed depending on the speed correction factor [0]

28.28.3 Injection calculation - torque correction in stratified

IP_MFF_MIN_CP_S	Canister purge margin on the fuel flow minimum value in stratified mode. This margin is used to prevent the minimum injection time in altitude or on deceleration. [5 mg/stk]
IP_FAC_TQI_CP	Factor used to convert a fuel mass in a torque. FAC_TQI_CP is a decimal factor which permit to convert a fuel mass in a torque. Because of the different resolutions of TQI_ADD_CP and MFF_CPS, the conversion (1389 mg/tdc / 1024 Nm) has to be taken into account by the calibration of FAC_TQI_CP. That means, if you want MFF_CPS = TQI_ADD_CP, then you have to calibrate FAC_TQI_CP to 0.73 (1024/1389) and not to 1. *Method:* • On engine dyno • On a reference point (N, TQI_SP_S, lambda) • Stratified mode • Open the valve with a full loaded canister (c_cppwm_as and lc_cppwm_as) • Decrease the fuel injected to go back to the reference point [0,67]

28.29 Torque reserve for canister purge at idle

LC_TQ_ADD_CP_INH	inhibit switch [1]
IP_TQ_ADD_CP	If not used set to [0 Nm]
C_CRLC_TQ_ADD_CP	Deactivation switch, filter constant and base map for canister purge torque reserve at idle. By increasing the engine load without increasing the torque output, a higher purge rate is possible due to the increase of engine's fuel demand (for a given purge fuel ratio). The torque reserve is only applied in MAX_PURGE and idle speed depending on canister load CL_MMV. *Method* : • idle speed condition • no additional load or torque reserve (cooling fan, air conditioning…) • use the map ip_tq_n_add_is to change the torque reserve (ref chapter idle speed) • Using the engine roughness criteria, measure the engine stability when you change torque reserve. Make measurements for several torque reserve and record the injection time and the spark advance. [0,5]

28.30 Canister Purge Valve Adaptation

Canister Purge Valve Adaptation Purpose of the canister purge solenoid (**CPS**) is to purge the tank venting charcoal filter by opening a pipe from the filter to the engine manifold; so the filter's CH-load is desorbed by fresh ambient air and sucked into the engine. For countinuous control of the purge airflow the CPS is opened pulse wide modulated (PWM).

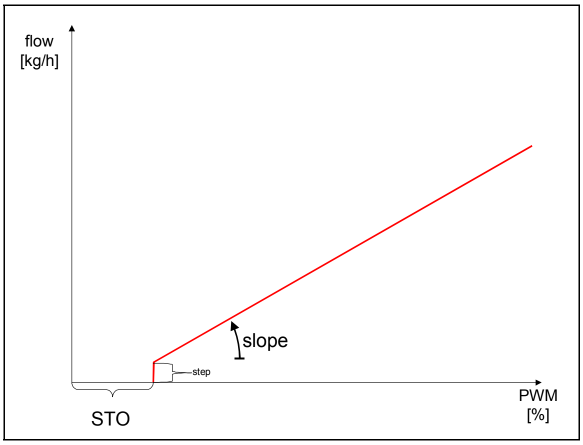


Figure: characteristic line of CPS

Due to future legislation the CH-emissions of vehicles have to be cut down; one task is to accellerate purge. To keep the purge airflow as high as possible without disturbing the engine the CPS has to be controlled as accurately as possible. This means that the engine controller (which is also responsible for canister purge) has to „know" as much as possible about any car's individual CPS. The most important parameter is the start-to-open-point (**STO**); a dead time between switching on electric power and opening of the CPS; as visible in the figure beside. If the opening time (depending on PWM and frequency) is below this dead time the CPS will never open. The parameter STEP is presently assumed to be quiet low and therefore neglected while SLOPE is estimated to be almost constant as specified.

Up to now an average STO has been extimated, but expected future developments like high-flow-CPSs or the need for a better performance of canister purge induce an increasing demand for an exactly measured STO.

To measure the STO the CPS is ramp-opened until a decreasing lambda-value (caused by the purged HCs) is detected. For accurate measuring the HC flow has to be as high as possible without disturbing the engine. So the adaptation frequency is calculated depending on canister load (CL) and calibrated values. A too low CL interdicts adaptation.

Boundary conditions needed to start adaptation:

STATE_CP	3
LV_HOM_ACT	1
LV_CL_CLC_VLD	1
N	between C_N_MIN_CPS_AD and C_N_MAX_CPS_AD
T_AST	above C_T_ST_CPS_AD
LV_CPS_AD_INH	0
TQI_AV	between C_TQI_AV_MIN_CPS_AD and C_TQI_AV_MAX_CPS_AD
CL_MMV	above C_CL_MIN_CPS_AD
LV_CP_CLOSE_REQ	0
LC_CPS_AD_CHK	0 This flag (=1) is to start the function overriding **all** application conditions. If the function is started using this flag it will not stop until a sufficient lambda-deviation is detected (=STO learned). When function is going into wait-state after lerning, set flag back to 0. **Attention !** If no sufficient lambda-deviation is detected (i.e. if CL is too low) the CPS will be opened up to 100% PWM and stay there until the flag is reset.
FAC_LAM_CP	to take from engine model

Required for application work:

- Charcoal-filter with CL about 1,5 (75%)
- Min-flow-CPS
- Max-flow-CPS

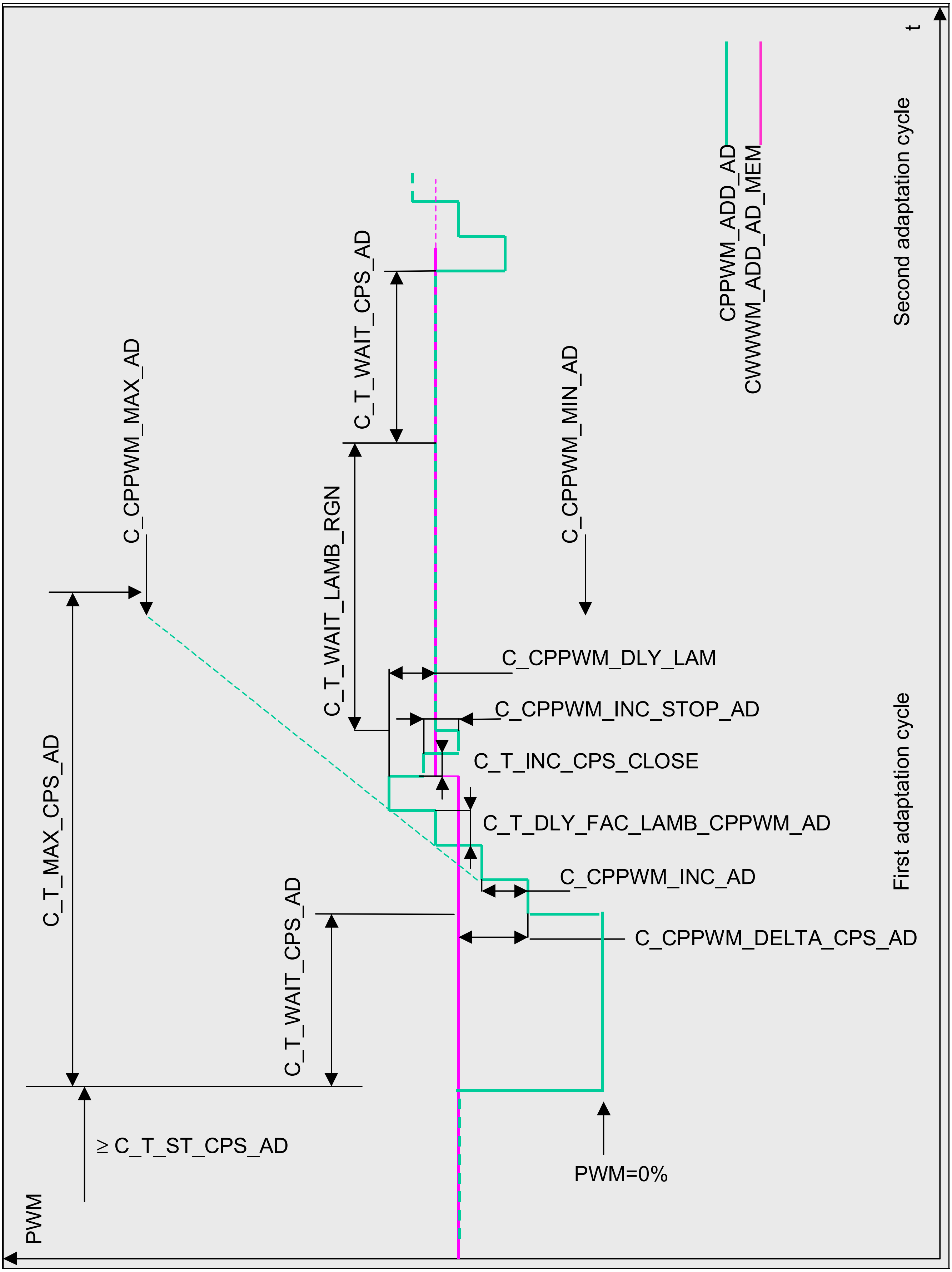


Figure: Adaptation cycle with calibraion values

Calibration method:

C_FRQ_MIN_CPS_AD	Minimum working frequency of CPS or default frequency of fixed-frequency-systems**; see CPS spec** or try 10 [Hz]
C_FRQ_MAX_MEC_CPS	Maximum allowed frequency of CPS**; see CPS spec** or try 30 [Hz].
C_FAC_LAMB_DELTA_MIN_CPS_AD	Minimum detectable lambda deviation caused by adaptation; **just above lambda-noise**. Watch FAC_LAMB at stable idle to estimate lambda noise or try 3 [%] as default.
C_FAC_LAMB_MAX_CPS_AD	Maximum allowed lambda deviation during adaptation without disturbing the engine; see engine or controller SPEC or: At low temperature (“cold morning”) open CPS stepwise, watch FAC_LAMB and see when engine is disturbed (sound/speed/vibrations).
C_CPPWM_AD_RATIO	Weighting factor of newly learned STO against old memorized value for calculation of new memorized value. 0.1 should be a useable default value. 1 can be used for monitoring during application work. This value acts like a “damper” against a badly learned STO.
C_CPPWM_DELTA_AD	Difference between PWM at start of adaptation and old memorized STO value to make sure adaptation starts below current STO. Should cover the measuring diffusion and a (smaller) possible drift of STO. This value also assures that after a “too high” learned STO the function will “climb back down”. Must be less than C_CPPWM_MIN_AD-C_CPPWM_DLY_LAM to avoid negative PWM value.Default: 5 [%]
C_CPPWM_DLY_LAM	Assuming the stepwise opening of CPS has continued between STO and lambda reaction (C_T_INC_AD < real physical time delay of C_T_DLY_FAC_LAMB_CPPWM_AD) this subtractive value compensates the PWM difference. Start application work with 0 [%]. Adapt using fully loaded canister at minimum frequency and watch FAC_LAMB. If learned STO (CPPWM_ADD_AD_MEM) is always higher than STO at beginning lambda deviation set the difference.
C_CPPWM_INC_AD	PWM increment for CPS opening during adaptation; depends on demanded function resolution and lambda regulation delay time. Has to be less than C_CPPWM_DELTA_AD. Default 1[%].
C_CPPWM_INC_STOP_AD	PWM icrement for CPS closing. Default 1[%].
C_CPPWM_MAX_AD	Maximum plausible STO-value at FRQ_MIN. See CPS spec or use 40[%] at 10Hz (= 40ms)

	as default.
C_CPPWM_MIN_AD	Minimum plausible STO-value at FRQ_MIN. Use 1[%] at 10Hz (= 1ms) or see CPS spec. Must be above (C_CPPWM_DELTA_AD to avoid negative PWM at next adaptation. Must be equal or higher than C_CPPWM_DLY_LAM+C_CPPWM_DELTA_AD
C_CYCNR_CPS_AD	Number of successful adaptations aimed during driving cycle. Default about 2 [-].
C_N_MAX_CPS_AD	Upper threshold of engine speed for adaptation. Default upper threshold of idle window; see engine spec or use 1200 [min^-1] as default.
C_N_MIN_CPS_AD	Lower threshold of engine speed for adaptation. Default lower threshold of idle window; see engine spec or use 600 [min^-1] as default-
C_T_DLY_FAC_LAMB_CPPWM_AD	Delay time between fuel purged into manifold and change of FAC_LAM value. Depends on engine speed and design. 1 [sec] may be a useable default value. After opening step of CPS this time passes until check for lambda deviation.
C_T_INC_CPS_CLOSE	Time increment for CPS closing, use 0,2 [sec] as default.
C_T_MAX_CPS_AD	Maximum allowed duration of one adaptation cycle. Try default 30 [sec]. As the figure shows, this time has to cover C_T_WAIT_CPS_AD and the ramp from C_CPPWM_MIN_AD-C_CPPWM_DELTA_AD to C_CPPWM_MAX_AD. So: C_T_MAX_CPS_AD≥ (C_CPPWM_MAX_AD- C_CPPWM_MIN_AD+ C_CPPWM_DELTA_AD)* C_T_INC_CPS_AD+ C_T_WAIT_CPS_AD
C_T_ST_CPS_AD	Minimum time between start engine and adaptation to ensure stable engine conditions. Default 60 [sec].
C_T_AST_DEAC_CPS_AD	Maximum time between start engine and end or abort of adaptation. If set to maximum value of T_AST this condition is disabled. Enter default 10000 [sec] to disable.
C_T_WAIT_CPS_AD	Delay time between function start and begin of CPS opening. Can be 0 [sec] if no delay demanded by application engineer.
C_T_WAIT_LAMB_RGN	Delay time after adaptation to regain stable conditions of lambda regulation. Try 10 [sec] and watch FAC_LAMB.
C_TQI_AV_MAX_CPS_AD	Upper threshold of engine torque for adaptation. Default upper threshold of idle window; see engine spec or use 40 [Nm]

C_TQI_AV_MIN_CPS_AD	Lower threshold of engine torque for adaptation. Default lower threshold of idle window; see engine spec or use 60 [Nm]

For first calibration or fixed frequency adaptation the frequency variation has to be terminated as follows:

C_FRQ_CLC_1_CPS_AD	=0
C_FRQ_CLC_2_CPS_AD	=C_FRQ_MIN_CPS_AD
C_CL_MIN_CPS_AD	Try as default: C_CL_MIN_CPS_AD=1 Use full canister and start purge to get current CL.Start adaptation with min-flow-CPS to get FAC_LAMB at STO. C_CL_MIN_CPS_AD=CL * C_FAC_LAMB_DELTA_MIN_CPS_AD/ FAC_LAMB

For frequency-variation:

The reliability of frequency calculation depends not only on it’s own calibration but also on the precision of CL-detection which - vice versa – depends on CPS-adaptation. Therefore an iterative calibration is recommended:

5. Complete calibration as described below.
6. Wait for purge to detect CL again.
7. Recalibrate C_CL_MIN_CPS_AD, C_FRQ_MAX_CPS_AD, C_FRQ_SLOP_CPS_AD.
8. If a significant difference appears: Start again with step 2.

Use canister with CL about 1,5 (=75%).

Mount max-flow-CPS and start adaptation. Get FAC_LAMB at STO. Start purge to get current CL. Calculate:

$$\mathrm{FRQ\_MAX\_CL\_MAX} = \frac{\mathrm{C\_FRQ\_MIN\_CPS\_AD} \times \mathrm{C\_FAC\_LAMB\_MAX\_CPS\_AD} \times 2}{\mathrm{FAC\_LAMB} \times \mathrm{CL}}$$

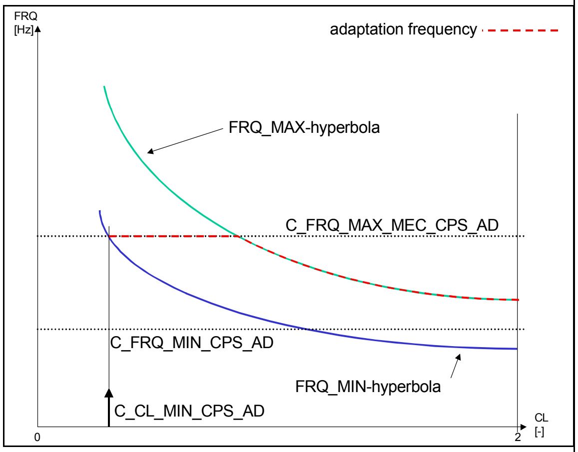
FRQ_MAX_CL_MAX has to be ≥ C_FRQ_MIN_CPS_AD !

Mount min-flow-CPS and start adaptation. Get FAC_LAMB at STO. Start purge to get current CL. Calculate:

$$\mathrm{FRQ\_MIN\_CL\_MAX} = \frac{\mathrm{C\_FRQ\_MIN\_CPS\_AD} \times \mathrm{C\_FAC\_LAMB\_DELTA\_MIN\_CPS\_AD} \times 2}{\mathrm{FAC\_LAMB} \times \mathrm{CL}}$$

C_CL_MIN_CPS_AD	Crossing of FRQ_MIN-hyperbola and C_FRQ_MAX_MEC_CPS-line. Mount min-flow-CPS and start adaptation. Get FAC_LAMB at STO. Start purge to get current CL. Calculate: C_CL_MIN_CPS_AD=CL * C_FRQ_MIN_CPS_AD / C_FRQ_MAX_MEC_CPS*

	C_FAC_LAMB_DELTA_MIN_CPS_AD/ FAC_LAMB Or try as default: C_CL_MIN_CPS_AD=1

C_FRQ_CLC_1_CPS_AD, C_FRQ_CLC_2_CPS_AD	The figure below shows two hyperbolas: The mimimum frequency required to generate a detectable FAC_LAMB and the maximum allowed frequency to not disturb the engine. The CL-range for adaptation is limited by the point where the minimum required frequency is getting higher than the maximum possible working frequency of the CPS.  Formula for hyperbola calculation: $FRQ = \frac{C\_FRQ\_CLC\_1\_CPS\_AD}{CL}$ For precise measuring the working frequency will always be as high as possible (FRQ_MAX or C_FRQ_MAX_MEC_CPS). For individual tuning a second constant allows to move the hyperbola: $FRQ = \frac{C\_FRQ\_CLC\_1\_CPS\_AD}{CL} + C\_FRQ\_CLC\_2\_CPS\_AD$ Use canister with CL about 1,5 (=75%). Mount max-flow-CPS and start adaptation. Get FAC_LAMB at STO. Start purge to get current CL during RAMP_OPEN. Calculate: C_FRQ_CLC_1_CPS_AD=CL*C_FRQ_MIN_CPS_AD*C_FAC_LAMB_MAX_CPS_AD / FAC_LAMB

	Necessary condition, check: C_FRQ_CLC_1_CPS_AD/2+ C_FRQ_CLC_2_CPS_AD ≥ C_FRQ_MIN_CPS_AD Fine tuning: Move frequency hyperbola by changing C_FRQ_CLS_2_CPS_AD or stretch/compress by changing C_FRQ_CLC_1_CPS_AD

28.31 Customer adaptation module: AGGR EVAC

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_dtev_aktiv	O/V	0...1H	0...1	1	[-]
Inhibition of MAP-control requested by fc cps					
B_eol_tev	O/V	0...1H	0...1	1	[-]
Bedingung BandendeTEV-Check aktiv					
B_te	O/V	0...1H	0...1	1	[-]
1: canister purge valve is not closed					
B_te_la1	O/V	0...1H	0...1	1	[-]
air fuel stechiometric for CP					
B_teblg	O/V	0...1H	0...1	1	[-]
Bedingung Beladungsgrad wird gelernt					
B_tev_erf	O/V	0...1H	0...1	1	[-]
Request to inhib Catalyst and O2 sensor diagnosis for allowing purge canister activation					
B_tev_zu	O/V	0...1H	0...1	1	[-]
Logical value for CPS valve closed					
Fakf	O/V	0...7FFFH	0...99.99694	3.0518e-3	[%]
Moving mean value of the canister load CL					
LV_CP_INH_CUS	O/V	0...1H	0...1	1	[-]
CPS-INH request from customer					
Mdi_res_tev	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
torque reserve for canister purge at idle					
MFF_ADD_TQ_CP	O/V	8000...7FFFH	-694.51059... 694.48940	0.0211948	[mg/stk]
TQ relevant mass flow					
MFF_MAX_CP_CUS	O/V	8000...7FFFH	-694.51059... 694.48940	0.0211948	[mg/stk]
Maximum mass flow from canister purge					
MFF_MAX_HOMS_CUS	O/V	8000...7FFFH	-694.51059... 694.48940	0.0211948	[mg/stk]
MFF threshold for engine operation mode change HOM <-> HOMS					
MFF_MAX_S_CUS	O/V	8000...7FFFH	-694.51059... 694.48940	0.0211948	[mg/stk]
MFF threshold for engine operation mode change HOMS <-> S					
Mste	O/V	0...FFFFH	0...25.59960	0.3906e-3	[kg/h]
mass flow through canister purge valve					
Rktev	O/V	8000...7FFFH	-16...15.99951	0.4883e-3	[-]
fuel correction due to canister purge					
Status_tev	O/V	0...FFH	0...255	1	[-]
State canister purge					

Input data:

B_tev_stopp	C_MAF_REF	C_MFF_FAC	CL_MMV
ECU_STATE	LV_CP_ACT_REQ	LV_CP_CLOSE_ACT	LV_EOL_CPS
LV_INH_MAP_CTL_DIAGCPS	LV_LAMB_AFS_CP	MAF_CPS	MFF_ADD_CYL_CP
Rk_vlo_hs	Rk_vlo_s	Rkte_max	Rkte_md
STATE_CP	TQ_ADD_CP		

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_DIAGCPS_EOL_HOM_REQ	1	0...1H	0...1	1	[-]
Homogeneous AFS request due to acive EOL DIAGCPS					

28.31.1 Outputs for BMW functions which are defined as EVAC exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions (for BMW variables):

Initialisation: 0 at reset

Recurrence : 10 ms

Rktev, Mste 20 ms

100ms: B_dtev_aktiv

Activation: every engine state, except power latch phase

Deactivation: power latch phase

Value at power latch phase: 0, Fakf: last valid value

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
If    ECU_STATE = "PWL"
Then
      Mdi_res_tev   = 0
      B_dtev_aktiv  = 0
      B_eol_tev     = 0
      B_tev_erf     = 0
      B_tev_zu      = 0
      Rktev         = 0
      Mste          = 0
      B_te_la1      = 0
      B_te          = 0
      Status_tev    = 0
      B_teblg       = 0
  Else
      Mdi_res_tev   =  TQ_ADD_CP
      B_dtev_aktiv  =  LV_INH_MAP_CTL_DIAGCPS
      B_eol_tev     = LV_EOL_CPS
      B_tev_erf     =  LV_CP_ACT_REQ
```

```
      B_tev_zu      =  LV_CP_CLOSE_ACT
      Rktev         =  MFF_ADD_CYL_CP / (C_MAF_REF * C_MFF_FAC)
      Status_tev    = STATE_CP
      Mste          =  MAF_CPS
      B_te_la1      =  LV_LAMB_AFS_CP or (LC_DIAGCPS_EOL_HOM_REQ and
      LV_EOL_CPS)
      B_te          =  ! LV_CP_CLOSE_ACT
      Fakf          =  CL_MMV

      If  STATE_CP = RAMP_OPEN (8H) or RAMP_OPEN_FAST (9H)
                                          or MAX_PURGE (AH)
      Then   B_teblg = 1
       Else   B_teblg = 0
       Endif
  Endif
```

28.31.2 Outputs for SV aggregates, customer → SV

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes.

Application conditions (for SVDO variables):

Initialisation: 0

Recurrence : 10 ms

Activation: at every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

LV_CP_INH_CUS = B_tev_stopp
MFF_MAX_CP_CUS = Rkte_max * C_MAF_REF * C_MFF_FAC
MFF_ADD_TQ_CP = Rkte_md * C_MAF_REF * C_MFF_FAC
MFF_MAX_S_CUS = Rk_vlo_s * C_MAF_REF * C_MFF_FAC
MFF_MAX_HOMS_CUS = Rk_vlo_hs * C_MAF_REF * C_MFF_FAC

29 Evaporative system monitoring

Table of Contents

Index

29.1 AGGR EVAM adaptation

29.1.1 Interface from/to TEV-check, leak detection and other SV internal functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CL_MMV_NORM_PURGE_END	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
Canister load at the end of MAX_PURGE operation					
LV_CL_MMV	V/O	0...1H	0...1	1	[-]
Interface flag to functional check cps					
LV_CL_MMV_TMP	V	0...1H	0...1	1	[-]
Intermediate value for interface flag to functional check cps					
LV_CP_NEW_RAMP_OPEN_MEM	V/O	0...1H	0...1	1	[-]
locally memorised version of LV_CP_NEW_RAMP_OPEN					
FLOW_SP_CPS_EVAP	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow setpoint through the CPS during functional check CPS (interface to EVAC AGGR)					
LV_ERR_EL_SOV	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on canister shut off valve command signal.					
LV_ERR_MEC_CLOSE_SOV	V/O	0...1H	0...1	1	[-]
Boolean for error currently present due to canister shut off valve stuck in closed position					
LV_ERR_MEC_OPEN_CPS	V/O	0...1H	0...1	1	[-]
Boolean for CPS stuck in open position (after debounce)					
LV_TLDV_REQ	V/O	0...1H	0...1	1	[-]
Flag indicating the activation requirement for the diagnosis (1)					
LV_ERR_CPS	V/O	0...1H	0...1	1	[-]
Flag indicating failure on canister purge valve					
LV_CL_CALC_AVL	V/O	0...1H	0...1	1	[-]
interface flag to cat.diag.: cl known					
T_DLY_PURGE_DIAGCPS	V/O	0...FFFFH	0...1310.7	0.02	[s]
Delay time for activating DIAGCPS-diagnosis (step 2) after LV_CL_MMV 0-->1					
LV_DIAGCP_CPS_ACT	V/O	0...1H	0...1	1	[-]
Flag indicating FLOW_SP_CPS_EVAP is controlled by EVAM					
LV_FAC_LAM_DIAGCP	V/O	0...1H	0...1	1	[-]
flag for lambda control to set initialize controller output					
T_DLY_CL_MMV	V	0...FFFFH	0...1310.7	0.02	[s]
Delay time for setting LV_CL_MMV 0 --> 1					
STATE_CP_MEM_AD	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH BH	CP_NOT_ACT NO_PURGE RAMP_TO_NO_PURGE WAIT_RAMP_OPEN MIN_PURGE - - - RAMP_OPEN RAMP_OPEN_FAST MAX_PURGE RAMP_CLOSE	1	[-]
Locally memorised version of STATE_CP					
T_DLY_CL_MMV_READY	V	0...FFFFH	0...1310.7	0.02	[s]
Delay time after which TEV check starts once Ramp_open has been started					
LV_T_DLY_CL_MMV_READY	V/O	0...1H	0...1	1	[-]
Flag when the first transition Wait Ramp Open to Ramp Open occurs					

Input data:

CL_MMV_CLC_END	LV_ES	STATE_CP	STATE_CLL_DEAC_CP
LV_CP_CLOSE_ACT	LV_CL_CLC_AVL	LV_ERR_EL_CPS	LV_CP_NEW_RAMP_OPEN
FLOW_SP_CP_DIAGCPS	LV_ACT_DIAGCPS	LV_IGK	LV_CL_MMV_CAL_ACT

FUNCTION DESCRIPTION:

Adaptation to (non-aggregate) TEV-check and leak detection.

Application conditions:

Initialisation: at reset **or** LV_IGK = 0 -> 1

CL_MMV_NORM_PURGE_END = 0
LV_CL_MMV = 0
T_DLY_PURGE_DIAGCPS = C_T_DLY_PURGE_DIAGCPS
T_DLY_CL_MMV = C_T_DLY_CL_MMV
LV_DIAGCP_CPS_ACT = 0
LV_ERR_CPS = 0
LV_ERR_MEC_OPEN_CPS = 0
FLOW_SP_CPS_EVAP = FLOW_SP_CP_DIAGCPS
LV_FAC_LAM_DIAGCP = 0
LV_ERR_EL_SOV = 0
LV_TLDV_REQ = 0
T_DLY_CL_MMV_READY = C_T_DLY_CL_MMV_READY
LV_T_DLY_CL_MMV_READY = 0
LV_CL_MMV_TMP = 0
LV_ERR_MEC_CLOSE_SOV = 0

at reset or LV_ES = 0 -> 1:

LV_DIAGCP_CPS_ACT = 0
LV_TLDV_REQ = 0
all other values = 0

Recurrence : 20 ms

Activation: every ECU state

Deactivation: --

Formula section:

// interface to leak detection

```
If      LV_ES = 0
Then    CL_MMV_NORM_PURGE_END = CL_MMV_CLC_END
Else    --
Endif
```

// interface to functional check cps

```
If (1)      STATE_CP_MEM_AD = WAIT_RAMP_OPEN      and
            STATE_CP = RAMP_OPEN
Then(1)     LV_CL_MMV_TMP = 0                     /* reset interface flag */
            T_DLY_PURGE_DIAGCPS = C_T_DLY_PURGE_DIAGCPS
            T_DLY_CL_MMV = C_T_DLY_CL_MMV
            LV_T_DLY_CL_MMV_READY = 1
Else(1)
    If(2)       STATE_CP_MEM_AD         = RAMP_OPEN        and
                STATE_CP ≠ RAMP_OPEN                       and
                STATE_CLL_DEAC_CP = CP_NO_DEAC             and
                LV_CP_NEW_RAMP_OPEN_MEM = 0
    Then (2)    T_DLY_CL_MMV = C_T_DLY_CL_MMV
    Else(2)
                If(3)   LV_CL_MMV_CAL_ACT = 1              and
                        STATE_CP = MAX_PURGE
                Then(3) T_DLY_CL_MMV  --           /*decrement counter
                        If(4) T_DLY_CL_MMV > 0
                        Then(4) LV_CL_MMV_TMP  = 0
                        Else(4) LV_CL_MMV_TMP  = 1 /* set temporary interface flag when
                        Endif(4)                   RAMP_OPEN   inished   and   time   delay
elapsed*
                Else(3)
                        If(5)  T_DLY_CL_MMV = 0            and
                               STATE_CP = MAX_PURGE
                        Then(5) LV_CL_MMV_TMP  = 1
                        Endif(5)
                Endif(3)
    Endif(2)
Endif
----
If          LV_T_DLY_CL_MMV_READY = 1
Then        T_DLY_CL_MMV_READY --                 /*decrement counter
Endif
----

If(1)       LV_CL_MMV_TMP  = 1
Then(1)
            If(2)       T_DLY_CL_MMV_READY = 0
            Then(2)     LV_CL_MMV =1
            Else(2)     LV_CL_MMV =0
            Endif(2)
Else(1)     LV_CL_MMV =0
Endif(1)

----
If          LV_CL_MMV = 1              and
            STATE_CP = MAX_PURGE
Then        T_DLY_PURGE_DIAGCPS  --         /*decrement counter
Endif
```

```
// memorisation of relevant variables for next recurrence

STATE_CP_MEM_AD = STATE_CP
LV_CP_NEW_RAMP_OPEN_MEM = LV_CP_NEW_RAMP_OPEN
```

calibration hint:

The RAMP_OPEN interruption time C_T_DI_RAMP_OPEN_CP should be set to minimum 40ms to prevent a wrong setting of LV_CL_MMV at short RAMP_OPEN interruption.

```
FLOW_SP_CPS_EVAP   = FLOW_SP_CP_DIAGCPS
LV_ERR_CPS         = LV_ERR_EL_CPS
LV_CL_CALC_AVL     = LV_CL_CLC_AVL

If      LV_ACT_DIAGCPS = 1 or
        (LV_ACT_DIAGCPS$_{n-1}$ = 1 and LV_ACT_DIAGCPS$_n$ = 0)
Then    LV_DIAGCP_CPS_ACT = 1
Else    LV_DIAGCP_CPS_ACT = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_PURGE_DIAGCPS	1	0...FFFFH	0...1310.7	0.02	[s]
Delay time for activating DIAGCPS-diagnosis (step 2) after LV_CL_MMV 0-->1					
C_T_DLY_CL_MMV	1	0...FFFFH	0...1310.70	0.02	s
Delay time for setting LV_CL_MMV 0 --> 1					
C_T_DLY_CL_MMV_READY	1	0...FFFFH	0...1310.7	0.02	[s]
Delay time after which TEV check starts once Ramp_open has been started					

29.2 Check filler cap during DC

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ROUGH_LEAK_SUSP_CHK_FUC	V/O/S	0...1H	0...1	1	[-]
Evap emission system suspects rough leak					
LV_CTR_DC_CHK_FUC	V/O/S	0...1H	0...1	1	[-]
Indicator of driving cycle counter enabled					
LV_ENA_CHK_FUC_REFU	V/O/S	0...1H	0...1	1	[-]
Valid refilling recognized					
LV_ENA_CHK_FUC_ROUGH_LEAK_MES	V/O	0...1H	0...1	1	[-]
Rough leak request enabled due to check filler cap					
CTR_DC_ENA_CHK_FUC	V/O/S	0...FFH	0...255	1	[-]
Number of DC since CFC request					
CTR_CHK_FUC_OPEN	V/O/S	0...FFH	0...255	1	[-]
Number of tests cfc with result filler cap open					
LV_CHK_FUC_CLOSE	V	0...1H	0...1	1	[-]
Filler cap close					
LV_CHK_FUC_OPEN	V/O/S	0...1H	0...1	1	[-]
Filler cap open					
LV_CHK_FUC_END	V	0...1H	0...1	1	[-]
Test filler cap ended					
LV_T_VS_CHK_FUC_OPEN_CAN	V	0...1H	0...1	1	[-]
Min. vehicle speed for control cfc lamp reached					
LV_T_VS_CHK_FUC_OPEN_MAX	V	0...1H	0...1	1	[-]
Recognized vehicle stops					
LV_CHK_FUC_OPEN_CAN	V/O	0...1H	0...1	1	[-]
Set check filler cap lamp in the ICL					
LV_CHK_FUC_OPEN_ACT	V/O/S	0...1H	0...1	1	[-]
Check filler cap lamp was already set					
LV_REP_CHK_FUC	V/O/S	0...1H	0...1	1	[-]
Healing test enabled					
LV_ERR_CHK_FUC	V	0...1H	0...1	1	[-]
Error flag check filler cap (open filler cap)					
LV_CDN_DIAG_CHK_FUC	V	0...1H	0...1	1	[-]
Diagnosis condition					
LV_END_DIAG_CHK_FUC	V	0...1H	0...1	1	[-]
End of diagnosis					
ERR_SYM_CHK_FUC	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom					
AMP_CHK_FUC	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
ambient pressure difference during check fuc					

Input data:

LV_ROUGH_LEAK_SUSP	LV_REFU	LV_REFU_VAL	FTL
VS	LV_ERR_FTL	LV_ERR_FTL_PLAUS	LV_ENA_CHK_FUC_ROUGH_LEAK
LV_CONF_DMTL	LV_CONF_DMTL	T_AST	LC_CONF_DMTL
LV_ERR_VS	LV_CYC_ROUGH_LEAK_SUSP	LV_ROUGH_LEAK_SUSP	STATE_DMTL
AMP_MES			

General information:

The function “check filler cap during DC” is part of the tank leakage detection package.

The function controls the conditions, which are responsible , to enable a check of open filler cap. Is such a condition available, a request LV_ENA_CHK_FUC_ROUGH_LEAK_MES can be set and a rough leak measurement shall be enabled.

The trigger for the check of open filler cap can be either a valid refueling or a given rough leak suspicion. Are the conditions available, which enable a check, that means low acceleration and low oscillation of the fuel tank level, then a check will be done, if a minimum speed of the vehicle is reached.

A test request is still present for a calibratable number of driving cycles. In case of a detected open filler cap, a calibratable number of tests can be done, after a indication lamp in the instrument cluster was set.

29.2.1 Rough leak request due to check filler cap during DC

General information:

Conditions to set the test request LV_ENA_CHK_FUC_ROUGH_LEAK_MES:

- ☐ A valid refueling is given. Setting of the nv-bit LV_ENA_CHK_FUC_REFU. The resetting of this logical value is done, if a CFC-test has finished.
- ☐ Availability of a suspected rough leak from the tank leakage detection function. The request is stored until a CFC-test has finished.
- ☐ Setting of the test request, if the vehicle speed is greater or equal than a threshold.

Conditions to reset the test request LV_ENA_CHK_FUC_ROUGH_LEAK_MES:

- ☐ Vehicle speed is lower than a threshold
- ☐ Powerful acceleration of the vehicle
- ☐ high ambient pressure deviation

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: in case of reset or LV_IGK = 0 → 1:

AMP_CHK_FUC = AMP_MES, all other variables to 0

Restore out of the NV-memory:

LV_ENA_CHK_FUC_REFU, LV_ROUGH_LEAK_SUSP_CHK_FUC
LV_CTR_DC_CHK_FUC

In case of clear FMY:

All variables to 0

AMP_CHK_FUC = AMP_MES

Recurrence: 100 ms

Activation: LC_CONF_DMTL = 1, during all ECU states

Formula section:

```
If    LV_ROUGH_LEAK_SUSP = 1                  AND          // suspected rough leak
      LC_COD_CHK_FUC = 1. Bit
      Then   LV_ROUGH_LEAK_SUSP_CHK_FUC = 1
Endif

If    LC_COD_CHK_FUC = 0. Bit        AND
      LV_REFU = 1                    AND                   // valid refueling
      LV_REFU_VAL = 1
      Then   LV_ENA_CHK_FUC_REFU = 1
Endif

If    LV_CHK_FUC_END = 1
      Then   LV_ROUGH_LEAK_SUSP_CHK_FUC = 0
             LV_ENA_CHK_FUC_REFU = 0
Endif

If    [LV_ROUGH_LEAK_SUSP_CHK_FUC = 1            OR
      (LC_COD_CHK_FUC = ! 0. Bit              AND
      LC_COD_CHK_FUC = ! 1. Bit)                 OR
      LV_ENA_CHK_FUC_REFU = 1]                             AND
      $VS \geq C\_VS\_ENA\_CHK\_FUC\_OPEN\_MIN$
      Then   LV_ENA_CHK_FUC_ROUGH_LEAK_MES = 1     // test request due to cfc
             LV_CTR_DC_CHK_FUC = 1                 // enabling DC-counter
             If     STATE_DMTL = REF_LEAK_MES → ROUGH_LEAK_MES
             Then   AMP_CHK_FUC = AMP
             Endif
Endif

If    VS < C_VS_ENA_CHK_FUC_OPEN_MAX                       OR
      $|VS_{(n)} - VS_{(n-1)}| > C\_VS\_DELTA\_CHK\_FUC\_MAX$            OR
      ((LV_ERR_FTL = 0            AND
      LV_ERR_FTL_PLAUS = 0        AND
      LC_COD_CHK_FUC = 2. Bit)                AND
      $|FTL_{(n)} - FTL_{(n-1)}| > C\_FTL\_DELTA\_CHK\_FUC\_MAX)$        OR
      LV_CHK_FUC_END = 1                                   OR
```

```
        (LV_CHK_FUC_OPEN = 1   AND
        LV_REP_CHK_FUC = 0)                                 OR
        |AMP_MES – AMP_CHK_FUC| > C_AMP_CHK_FUC_MAX
        Then  LV_ENA_CHK_FUC_ROUGH_LEAK_MES = 0      // reset of the test request
              AMP_CHK_FUC = AMP_MES
    End

    If  LV_CHK_FUC_END = 1                 OR
        LV_CHK_FUC_OPEN = 0 → 1
        Then  LV_CTR_DC_CHK_FUC = 0
    Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_COD_CHK_FUC	1	0...FFH	0...255	1	[-]
Calibration codebyte for check filler cap test					
C_VS_ENA_CHK_FUC_OPEN_MIN	1	0...FFH	0...255	1	[km/h]
Min. vehicle speed to enable cfc test					
C_VS_ENA_CHK_FUC_OPEN_MAX	1	0...FFH	0...255	1	[km/h]
Max. vehicle speed to enable cfc test					
C_VS_DELTA_CHK_FUC_MAX	1	0...FFH	0...255	1	[km/h]
Max. difference between two consecutive VS					
C_FTL_DELTA_CHK_FUC_MAX	1	0...7EH	0...126	1	[l]
Max. difference between two consecutive FTL					
C_AMP_CHK_FUC_MAX	1	0...FFFFH	0...5434	0.0829175	[hPa]
maximum deviation of the ambient pressure during check fuc					

29.2.2 Decision of check filler cap open or closed

General information:

Conditions to set LV_CHK_FUC_END:

- ❑ Number of the maximum DC are reached. This condition is reached, if the engine stops C_CTR_DC_ENA_CHK_FUC_MAX times, indicated by T_AST. The number of the counted cycles is reseted, if the filler cap test has ended or in case of the error “open filler cap”.
- ❑ Cfc test successfully ended, to be recognized by the cycleflag of the rough leak test in common with no suspected rough leak
- ❑ End of the test with C_CTR_CHK_FUC_OPEN_MAX times open filler cap.

Conditions to reset LV_CHK_FUC_END:

- ❑ By initialisation of the ECU
- ❑ With recognition of a new valid refueling

Conditions to set LV_CHK_FUC_OPEN

- ❑ Cycle flag of the rough leak measurement

- ☐ Suspected rough leak

Condition to reset LV_CHK_FUC_OPEN

- ☐ End of check filler cap test

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: in case of reset or LV_IGK = 0 → 1:

Restore out of the NV-memory:

CTR_DC_ENA_CHK_FUC, CTR_CHK_FUC_OPEN
LV_CHK_FUC_OPEN

All other variables with 0

In case of clear FMY:

All variables to 0

Recurrence: 100 ms

Activation: LC_CONF_DMTL = 1, during all ECU states

Formula section:

```
If    LV_CTR_DC_CHK_FUC = 1              AND
      LV_CTR_DC_ENA_CHK_FUC = 0
      Then  CTR_DC_ENA_CHK_FUC ++                   // driving cycle counter
            LV_CTR_DC_ENA_CHK_FUC = 1
Endif

If    LV_ROUGH_LEAK_SUSP = 0                         AND
      LV_CYC_ROUGH_LEAK_SUSP = 0 → 1                 AND
      LV_ENA_CHK_FUC_ROUGH_LEAK = 1
      Then  LV_CHK_FUC_CLOSE = 1                     // filler cap closed
      Else  LV_CHK_FUC_CLOSE = 0
Endif

If    LV_ROUGH_LEAK_SUSP = 1                         AND
      LV_CYC_ROUGH_LEAK_SUSP = 0 → 1                 AND
      LV_ENA_CHK_FUC_ROUGH_LEAK = 1
```

```
        Then   LV_CHK_FUC_OPEN = 1                                   // filler cap opened
               CTR_CHK_FUC_OPEN ++
    Endif

    If  CTR_DC_ENA_CHK_FUC = C_CTR_DC_ENA_CHK_FUC_MAX           OR
        LV_ERR_VS = 1                                           OR
        LV_CHK_FUC_CLOSE = 1                                    OR
        CTR_CHK_FUC_OPEN = C_CTR_CHK_FUC_OPEN_MAX
        Then   LV_ CHK_FUC_END = 1                                   // filler cap test ended
    Endif

    If  LC_COD_CHK_FUC = 0. Bit                  AND
        LV_ENA_CHK_FUC_REFU = 0 → 1                                  // valid refueling recognized
        Then   LV_CHK_FUC_END = 0
    Endif

    If  LV_CHK_FUC_END = 1
        Then   LV_CHK_FUC_OPEN = 0
    Endif

    If  LV_CHK_FUC_END = 1              OR
        LV_CHK_FUC_OPEN = 0 → 1
        Then   CTR_DC_ENA_CHK_FUC = 0
               CTR_CHK_FUC_OPEN = 0
    Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_AST_DC_CHK_FUC	1	0...FFFFH	0...6553.5	0.1	[s]
Threshold for AST to detect a DC					
C_CTR_DC_ENA_CHK_FUC_MAX	1	0...FFH	0...255	1	[-]
Threshold for number of DC since CFC request					
C_CTR_CHK_FUC_OPEN_MAX	1	0...FFH	0...255	1	[-]
Threshold for number of tests cfc with result filler cap open					

29.2.3 Control of the check filler cap lamp in the instrument cluster

General information:

Forming of the lamp control by LV_CHK_FUC_OPEN and the stored value LV_CHK_FUC_OPEN_ACT. If an opened filler cap is recognized and the vehicle speed is higher than a threshold for a calibratable time, the indication lamp will be set for a calibratable time. To prevent a setting of the lamp after the lamp has burned once, this information is stored in LV_CHK_FUC_OPEN_ACT. This bit shall be reseted after the end of the check filler cap test.

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: in case of reset or LV_IGK = 0 → 1:

Restore out of the NV-memory:

LV_CHK_FUC_OPEN_ACT, LV_REP_CHK_FUC

All other variables with 0

In case of clear FMY:

All variables to 0

Recurrence: 100 ms

Activation: LC_CONF_DMTL = 1, during all ECU states

Formula section:

```
If    VS ≥ C_VS_CHK_FUC_OPEN_CAN_MIN
      Then  T_VS_CHK_FUC_OPEN_CAN_MIN ++
      Else  T_VS_CHK_FUC_OPEN_CAN_MIN = 0
Endif

If    T_VS_CHK_FUC_OPEN_CAN_MIN ≥ C_T_VS_CHK_FUC_OPEN_CAN_MIN
      Then  LV_T_VS_CHK_FUC_OPEN_CAN = 1
      Else  LV_T_VS_CHK_FUC_OPEN_CAN = 0
Endif

If    VS ≤ C_VS_CHK_FUC_OPEN_MAX
      Then  T_VS_CHK_FUC_OPEN_MAX  ++
      Else  T_VS_CHK_FUC_OPEN_MAX = 0
```

```
Endif

If   T_VS_CHK_FUC_OPEN_MAX ≥ C_T_VS_CHK_FUC_OPEN_MIN
     Then  LV_T_VS_CHK_FUC_OPEN_MAX = 1
     Else  LV_T_VS_CHK_FUC_OPEN_MAX = 0
Endif

If   LV_CHK_FUC_OPEN = 1                 AND
     LV_T_VS_CHK_FUC_OPEN_CAN = 1        AND
     LV_CHK_FUC_OPEN_ACT = 0
     Then  LV_CHK_FUC_OPEN_CAN = 1
           LV_CHK_FUC_OPEN_ACT = 1
           T_CHK_FUC_OPEN_CAN = C_T_CHK_FUC_OPEN_CAN
     Else  T_CHK_FUC_OPEN_CAN --
           If    T_CHK_FUC_OPEN_CAN = 0
                 Then  LV_CHK_FUC_OPEN_CAN = 0
           Endif
Endif

If   LV_CHK_FUC_OPEN_ACT = 1             AND
     LV_T_VS_CHK_FUC_OPEN_MAX = 1
     Then  LV_REP_CHK_FUC = 1
Endif

If   LV_CHK_FUC_END = 1
     Then  LV_REP_CHK_FUC = 0
Endif
If   LV_CHK_FUC_END = 1             OR
     LV_CYC_ROUGH_LEAK_SUSP = 0 → 1           // sequencing ???
     Then  LV_CHK_FUC_OPEN_ACT = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_CHK_FUC_OPEN_CAN_MIN	1	0...FFH	0...255	1	[km/h]
Minimum vehicle speed for setting cfc lamp					
C_T_VS_CHK_FUC_OPEN_CAN_MIN	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time for vehicle speed higher than threshold					
C_VS_CHK_FUC_OPEN_MAX	1	0...FFH	0...255	1	[km/h]
Maximum vehicle speed for setting the healing					
C_T_VS_CHK_FUC_OPEN_MIN	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time for vehicle speed lower than threshold					
C_T_CHK_FUC_OPEN_CAN	1	0...FFFFH	0...6553.5	0.1	[s]
Duration of setting the cfc lamp					

29.2.4 Forming the error flag

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: in case of reset or LV_IGK = 0 → 1:

~~Restore out of the NV-memory:~~

~~LV_ERR_CHK_FUC~~

All other variables with 0

In case of clear FMY:

All variables to 0

Recurrence: 100 ms

Activation: LC_CONF_DMTL = 1, during all ECU states

Formula section:

```
If    LV_ENA_CHK_FUC_ROUGH_LEAK = 1
      Then  LV_CDN_DIAG_CHK_FUC = 1
      Else  LV_CDN_DIAG_CHK_FUC = 0
Endif

If    LV_CHK_FUC_OPEN = 0 → 1                OR
      LV_CHK_FUC_CLOSE = 0 → 1
      Then  LV_END_DIAG_CHK_FUC = 1
Endif
```

```
If    LV_CHK_FUC_OPEN = 0 → 1         AND
      LV_CHK_FUC_CLOSE = 0
      Then   ERR_SYM_CHK_FUC = MAX
             LV_ERR_CHK_FUC = 0
             LV_ERR_CHK_FUC = 1
Endif

If    LV_CHK_FUC_OPEN = 0             AND
      LV_CHK_FUC_CLOSE = 0 → 1
      Then   ERR_SYM_CHK_FUC = NO_SYM
             LV_ERR_CHK_FUC = 0
Endif
```

29.3 CPS component check via "functional check"

Output Data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DIAGCPS_MIN_MOD	V/O	0...1H	0...1	1	[-]
Indication EVAC sent into MIN_PURGE by FC CPS					
FLOW_SP_CPS_OLD	V	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Last EVAP-calculated flow setpoint through the CPS before start of FC CPS					
ERR_SYM_DIAGCPS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected failure - functional check CPS					
LV_CDN_DIAG_DIAGCPS	V/O	0...1H	0...1	1	[-]
Condition of diagnosis - functional check CPS					
LV_DIAGCPS_MIN_MOD	O/V	0...1H	0...1	1	-
Indication EVAC sent into MIN_PURGE by FC CPS					
LV_END_DIAG_DIAGCPS	V/O	0...1H	0...1	1	[-]
End of diagnosis - functional check CPS					
LV_ERR_DIAGCPS	V/O	0...1H	0...1	1	[-]
Indication for faulty functions - functional check CPS					
LV_END_RBM_DIAGCPS	V/O	0...1H	0...1	1	[-]
Indication for RBM that all three steps have been executed - functional check CPS					
STATE_DIAGCPS_RBM	V	0H 1H 2H 3H 4H 5H	STEP_INIT STEP_1 STEP_2 STEP_3 CPS_RAMP LOCK_STEP	1	[-]
State - functional check CPS					
STATE_DIAGCPS	V/O	0H 1H 2H 3H 4H 5H	STEP_INIT STEP_1 STEP_2 STEP_3 CPS_RAMP LOCK_STEP	1	[-]
State - functional check CPS					
LV_ACT_DIAGCPS	V/O	0...1H	0...1	1	[-]
Indication for functional check cps (step 2 or 3) being activated					
LV_MAF_DIAGCPS_0	V	0...1H	0...1	1	[-]
Auxiliary bit for step 3 - functional check CPS					
LV_MAF_DIAGCPS_1	V	0...1H	0...1	1	[-]
Auxiliary bit for step 3 - functional check CPS					
LV_EOL_CPS_ERR	V/O	0...1H	0...1	1	[-]
LV indicating error of functional check CPS (step 2) after EOL-test					
LV_EOL_CPS_INI	V	0...1H	0...1	1	[-]
LV indicating EOL-requested functional check CPS					
LV_OPM_AV_CNG_DIAGCPS	V	0...1H	0...1	1	[-]
LV indicating change of operation mode					
LV_N_DIF_DIAGCPS	V	0...1H	0...1	1	[-]
Auxiliary bit for step 2 - functional check CPS					
LV_N_DIF_DIAGCPS_1	-	0...1H	0...1	1	[-]
Auxiliary bit for step 2 - functional check CPS					
LV_MAP_DIF_DIAGCPS	V	0...1H	0...1	1	[-]
Auxiliary bit for step 2 - functional check CPS					
LV_MAP_DIF_DIAGCPS_1	-	0...1H	0...1	1	[-]
Auxiliary bit for step 2 - functional check CPS					
LV_INH_MAP_CTL_DIAGCPS	V/O	0...1H	0...1	1	[-]
inhibition of MAP-control during step 2 - functional check CPS					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAM_DIAGCPS	V	0...1H	0...1	1	[-]
Auxiliary bit for step 2 ¿ functional check CPS					
LV_LAM_SUM_DIAGCPS	V	0...1H	0...1	1	[-]
Auxiliary bit for step 2 ¿ functional check CPS					
LV_DI_SUM_DIAGCPS	-	0...1H	0...1	1	[-]
Auxiliary bit for step 2 - functional check CPS					
LV_SUPP_ACCIN_DIAGCPS	V/O	0...1H	0...1	1	[-]
Bit to suppress the air conditioning compressor request signal - functional check CPS					
MAF_DIAGCPS	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
MAF value on start of step 3 - functional check CPS					
N_DIAGCPS	V	0...1FE0H	0...8160	1	[rpm]
N value on starting the time C_T_DIAGCPS_N_2					
MAP_DIAGCPS	V	0...FFFFH	0...5434	0.0829175	[hPa]
MAP value on starting the time C_T_DIAGCPS_N_MAP_2					
PRS_IM_CTL_I_DIAGCPS	V	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Pressure controller I-part value on starting the time C_T_DIAGCPS_N_MAP_2					
SUM_FLOW_SP_DIAGCPS_EOL	V/O	0...FFH	0...255	1	[cyc]
Number of step 2 loops which have been passed if EOL-Test is active - fuctional check CPS					
SUM_FLOW_SP_DIAGCPS	V	0...FFH	0...255	1	[cyc]
Number of step 2 loops which have been passed - functional check CPS					
SUM_DIAG_DIAGCPS_EOL	V/O	0...FFH	0...255	1	[cyc]
Number of diagnoses if EOL-Test is active - functional check CPS					
SUM_N_DIAG_DIAGCPS	V	0...FFH	0...255	1	[cyc]
Number of diagnoses results N - functional check CPS					
SUM_MAP_DIAG_DIAGCPS	V	0...FFH	0...255	1	[cyc]
Number of diagnoses results MAP - functional check CPS					
SUM_N_DIAG_DIAGCPS_OFS	V	0...FFH	0...255	1	[cyc]
Number of diagnoses results N within 1 offset - functional check CPS					
SUM_MAP_DIAG_DIAGCPS_OFS	V	0...FFH	0...255	1	[cyc]
Number of diagnoses results MAP within 1 offset - functional check CPS					
SUM_MAF_DIAG_DIAGCPS_OFS	V	0...FFH	0...255	1	[cyc]
Number of diagnoses results MAF within 1 offset - functional check CPS					
SUM_DIAG_DIAGCPS	V	0...FFH	0...255	1	[cyc]
Number of diagnoses - functional check CPS					
T_DIAGCPS_DLY	V	1...FFH	0.1...25.5	0.1	[s]
Delay time - functional check CPS					
T_DIAGCPS_MAF	V	1...FFH	0.1...25.5	0.1	[s]
Time for active FLOW_SP offset - functional check CPS					
T_DIAGCPS_N_MAP_1	V	1...FFH	0.1...25.5	0.1	[s]
1st time interval for N-DIF-Diagnosis - functional check CPS					
T_DIAGCPS_N_MAP_2	V	1...FFH	0.1...25.5	0.1	[s]
2nd time interval for N-DIF-Diagnosis - functional check CPS					
T_DIAGCPS_N_MAP_3	V	1...FFH	0.1...25.5	0.1	[s]
3rd time interval for N-DIF-Diagnosis - functional check CPS					
T_DIAGCPS_LAM_1	V	1...FFH	0.1...25.5	0.1	[s]
1st time interval for LAM-Diagnosis - functional check CPS					
TI_LAM_DIAGCPS_DIF[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
TI_LAM_COR_i - difference while CPS-opening during STEP_2 - functional check CPS					
TI_LAM_DIAGCPS_MIN[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
TI_LAM_COR_i - min-value while CPS-opening during STEP_2 - functional check CPS					
TI_LAM_DIAGCPS_MAX[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
TI_LAM_COR_i - max-value while CPS-opening during STEP_2 - functional check CPS					
CL_MMV_SAE	V/O/S	0...FFFFH	0...1.99996	0.0305e-3	[-]
Output mode06h - Step 1					
SUM_DIAG_DIAGCPS_SAE	V/O/S	0...FFH	0...255	1	[cyc]
Output mode06h - Step 2					
MAF_DIAGCPS_SAE	V/O/S	0...FFFFH	0...1389	0.0211948	[mg/stk]
Output mode06h - Step 3					
MAF_DIAGCPS_THD_SAE	V/O/S	0...FFFFH	0...1389	0.0211948	[mg/stk]
Output mode06h - Step 3					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_DIAGCPS_DLY_1_RBM	V	1...FFH	0.1...25.5	0.1	[s]
Delay time step 2 (RBM) - functional check CPS					
T_DIAGCPS_DLY_2_RBM	V	1...FFH	0.1...25.5	0.1	[s]
Delay time step 3 (RBM) - functional check CPS					
T_DIAGCPS_MAF_RBM	V	1...FFH	0.1...25.5	0.1	[s]
Time for active FLOW_SP offset step 3 (RBM) - functional check CPS					
T_DIAGCPS_N_MAP_1_RBM	V	1...FFFFH	0.1...6553.5	0.1	[s]
time interval for N-DIF-Diagnosis step 2 (RBM) - functional check CPS					
LV_T_DIAGCPS_RBM	V	0...1H	0...1	1	[-]
Auxiliary bit timer for RBM - functional check CPS					
LV_OPM_AV_DIAGCPS_P_THR	V	0...1H	0...1	1	[-]
LV indicating change of operation mode within OPM_AV = 1 (Teildrosselung in UGD)					
MAF_SP_TQI_MMV_DIAGCPS	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Moving mean value of MAF_SP_TQI (RAM cell) - functional check CPS					
LV_MAF_SP_TQI_DYW_DIAGCPS	V/O	0...1H	0...1	1	[-]
Indication for limited dynamic condition - functional check CPS					
T_CL_MMV_DIAGCPS	V/O	0...FFH	0...25.5	0.1	[s]
Timer to count how long the CL_MMV has been above the threshold in Step 1 (UGD)					
T_CL_MMV_DIAGCPS_2	V/O	0...FFH	0...25.5	0.1	[s]
Timer to count how long the CL_MMV has been above the threshold in Step 1 (VVTNOTL1)					
FLOW_SP_CP_DIAGCPS	V/O	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow setpoint through the CPS during functional check CPS					

Input data:

LV_DI_DIAGCPS	LV_CH	N_DIF	N
CL_MMV	LV_IGK	MAF	TCO
FLOW_SP_CPS	LV_DC	LV_INH_DIAGCPS	LV_EOL_CPS
LV_IS	VS	LV_IGK	LV_TQ_ISC_I_TQ_PSTE
TI_LAM_COR[NC_CBK_EX_NR]	MAP_DIP_MES_BAS	MAF_MES	STATE_OPM_ENG_CP
T_DLY_PURGE_DIAGCPS	MAP_DIP_SP_MMV	LV_CP_CLL	LV_CL_MDL_ACT
MAF_SP_TQI	LV_DC	LV_CL_MMV_CAL_ACT	LV_HOM_ACT
PRS_IM_CTL_I			

FUNCTION DESCRIPTION:

General Information:

The diagnosis is used for the functional test of the CP solenoid. Therefore the CPS is diagnosed via its effects on the engine.

The functional check of the CPS consists of 3 steps:

- Step 1: based on the ACF load degree
- Step 2: based on
    - the engine speed change at IS,
    - the deviation of lambda-controller and
    - the measured MAP before and during CPS-opening
- Step 3: based on the difference between the measured air mass flow before and during a CPS opening

If the CPS has been detected to be OK within a step or as not OK after step 3 then the function is exited irreversibly until the next engine start.

The steps 1, 2 and 3 are processed successively, i.e. the next step may only be performed if the preceeding one has not been detected as OK. The only exception is step 1. Step 1 may also be performed parallel to step 2 and 3, which means:

After step 1 has been processed once and CPS has not been detected as OK then (if enable conditions are fulfilled) it is proceeded with step 2 and at the same time calculation of step 1 is started again. Calculation of step 1 is done until CPS is detected as OK within one of the steps (1, 2 or 3) or until CPS is dectected as definitely not OK by the end of step 3.

If the CPS is detected to be not OK after all 3 steps have been passed (end of step 3), then the error is set.

The complete diagnosis “functional check CPS” is performed only once per driving cycle.

Exception:

Activation of functional check CPS via EOL-test (LV_EOL_CPS 0 -->1):

Through the EOL-test the functional check CPS can be triggered several times per engine operation.

The diagnosis is addressed via the serial interface. Only STEP_2 will be performed.

For general enable conditions of the diagnosis see chapter "Appl. Inc. Functional Check CPS".

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                               (= SYM_0)
 │ │ └───> -                               (= SYM_1)
 │ └─────> CPS – EOL-error                 (= SYM_2)
 └───────> CPS – plausibility-error        (= SYM_3)
```

Application conditions:

Initialisation: 1) on transition LV_IGK OFF→ ON or reset:

All variables from output-data-section have to be initialized with zero:

except:
STATE_DIAGCPS_OLD = STEP_1
STATE_DIAGCPS = STEP_INIT
T_DIAGCPS_DLY = C_T_DIAGCPS_DLY
T_CL_MMV_DIAGCPS = C_T_CL_MMV_DIAGCPS
T_CL_MMV_DIAGCPS_2 = C_T_CL_MMV_DIAGCPS_2
CL_MMV_SAE = CL_MMV_SAE
SUM_DIAG_DIAGCPS_SAE =
SUM_DIAG_DIAGCPS_SAE
MAF_DIAGCPS_SAE = MAF_DIAGCPS_SAE
MAF_DIAGCPS_THD_SAE = MAF_DIAGCPS_THD_SAE

2) on clearing failure memory

All variables from output-data-section have to be initialized with zero:

except:
STATE_DIAGCPS_OLD = STEP_1
STATE_DIAGCPS = STEP_INIT
T_DIAGCPS_DLY = C_T_DIAGCPS_DLY
CL_MMV_SAE = FFFFH
SUM_DIAG_DIAGCPS_SAE = ~~FFH~~ 0H
MAF_DIAGCPS_SAE = 0
MAF_DIAGCPS_THD_SAE = FFFFH
T_CL_MMV_DIAGCPS = C_T_CL_MMV_DIAGCPS
T_CL_MMV_DIAGCPS_2 = C_T_CL_MMV_DIAGCPS_2

Recurrence: 100 ms

Activation: LV_DC = 1

Deactivation: LV_DC = 0

Conditions for End-of-line (EOL) - Test:

Description:

This is to secure that any time the functional check CPS is activated via EOL-Test :

- the functional check starts right from the beginning again
- only those steps of the functional check CPS are performed regarding the EOL-Test
- the diagnosis results of the functional check (issued via serial interface) are the results from the EOL-Test only

Formula section:

Detection of EOL-Request:

```
IF          (LV_EOL_CPS(k-1) = 0 AND LV_EOL_CPS(k) = 1)
THEN        LV_EOL_CPS_INI = 1
            STATE_DIAGCPS = STEP_INIT
ELSE        LV_EOL_CPS_INI = 0
ENDIF
```

Detection of OPM-Change:

```
IF     MAP_DIP_SP_MMV > C_MAP_DIP_SP_MMV_DIAGCPS
THEN   LV_OPM_AV_DIAGCPS_P_THR = 1       //GD, T-GD
ELSE   LV_OPM_AV_DIAGCPS_P_THR = 0       //UGD
ENDIF

IF     STATE_DIAGCPS = STEP_2       or
       STATE_DIAGCPS = STEP_3
THEN
   IF    (LV_OPM_AV_DIAGCPS_P_THR k ! =      // UGD <-> T- GD or GD
         LV_OPM_AV_DIAGCPS_P_THR k-1)
   THEN      LV_OPM_AV_CNG_DIAGCPS = 1
             STATE_DIAGCPS = STEP_INIT
   ELSE      LV_OPM_AV_CNG_DIAGCPS = 0
   ENDIF
ENDIF
```

The functionality in Step 1 which consists in counting the time during which the CL_MMV is above the threshold C_CL_MMV_DIAGCPS, needs to be reset in the following cases:

- transition between UGD/GD happens
- the system is in open loop (CL_MMV frozen)
- the canister load model is active (CL_MMV not reliable)

```
IF      (LV_OPM_AV_DIAGCPS_P_THR $_k$ ! =
         LV_OPM_AV_DIAGCPS_P_THR $_{k-1}$)        OR
        LV_CP_CLL = 0                            OR
        LV_CL_MDL_ACT = 1                        OR
        LV_CL_MMV_CAL_ACT = 0
        THEN  T_CL_MMV_DIAGCPS = C_T_CL_MMV_DIAGCPS
              T_CL_MMV_DIAGCPS_2 = C_T_CL_MMV_DIAGCPS_2
ENDIF
```

STEP_INIT:

LV_ACT_DIAGCPS = 0

LV_DIAGCPS_MIN_MOD = 0

LV_INH_MAP_CTL_DIAGCPS = 0

LV_MAF_DIAGCPS_i = 0

LV_EOL_CPS_ERR = 0

LV_N_DIF_DIAGCPS = 0

LV_N_DIF_DIAGCPS_1 = 0

LV_DI_SUM_DIAGCPS = 0

LV_MAP_DIF_DIAGCPS = 0

LV_MAP_DIF_DIAGCPS_1 = 0

SUM_MAP_DIAG_DIAGCPS_OFS = 0

SUM_N_DIAG_DIAGCPS_OFS = 0

SUM_MAF_DIAG_DIAGCPS_OFS = 0

LV_SUPP_ACCIN_DIAGCPS = 0

LV_LAM_DIAGCPS = 0

MAF_DIAGCPS = 0

N_DIAGCPS = 0

MAP_DIAGCPS = 0

PRS_IM_CTL_I_DIAGCPS = 0

SUM_DIAG_DIAGCPS_EOL = 0

SUM_FLOW_SP_DIAGCPS_EOL = 0

```
T_DIAGCPS_N_MAP_1 = 0
T_DIAGCPS_N_MAP_2 = 0
T_DIAGCPS_N_MAP_3 = 0
T_DIAGCPS_LAM_1 = 0
T_DIAGCPS_MAF = 0
TI_LAM_DIAGCPS_DIF_i = 0
TI_LAM_DIAGCPS_MIN_i = 7FFFH
TI_LAM_DIAGCPS_MAX_i = 8000H
T_DIAGCPS_DLY = C_T_DIAGCPS_DLY
T_DIAGCPS_DLY_1_RBM = 0
T_DIAGCPS_DLY_2_RBM = 0
T_DIAGCPS_N_MAP_1_RBM = 0
T_DIAGCPS_MAF_RBM = 0
LV_T_DIAGCPS_RBM = 0
T_CL_MMV_DIAGCPS = C_T_CL_MMV_DIAGCPS
T_CL_MMV_DIAGCPS_2 = C_T_CL_MMV_DIAGCPS_2


IF(1)       LV_INH_DIAGCPS = 1  OR  // Functional check CPS is exited irreversibly
            LV_EOL_CPS_INI = 1   OR  // EOL-Test active
            LV_OPM_AV_CNG_DIAGCPS = 1  // change of OPM_AV detected
THEN(1)     LV_EOL_CPS_INI = 0
            LV_OPM_AV_CNG_DIAGCPS = 0
            FLOW_SP_CP_DIAGCPS = 0
            FLOW_SP_CPS_OLD = 0
            SUM_DIAG_DIAGCPS =  0
            SUM_FLOW_SP_DIAGCPS = 0
            SUM_N_DIAG_DIAGCPS =  0
            SUM_MAP_DIAG_DIAGCPS =  0
            LV_LAM_SUM_DIAGCPS = 0

            except:     STATE_DIAGCPS_OLD = STEP_1
                        LV_ERR_DIAGCPS and ERR_SYM_DIAGCPS
                        must not be changed
ELSE(1)     FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS_OLD
```

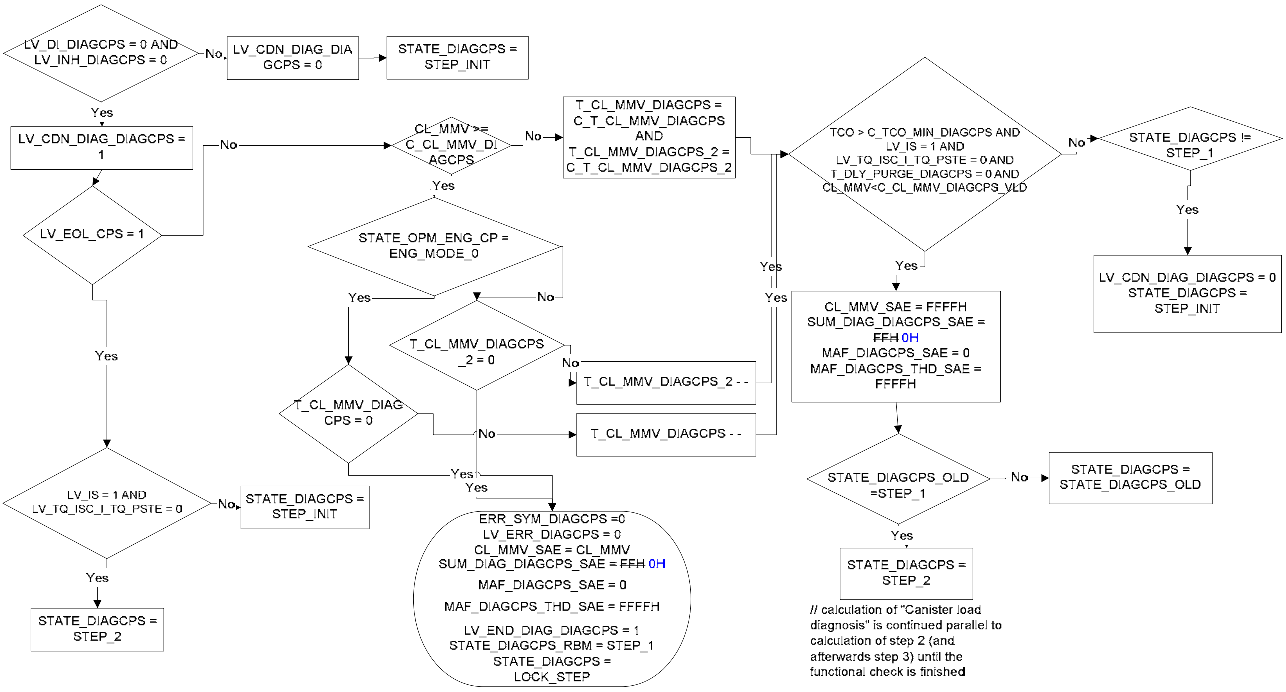
```
        IF(2)        LV_DI_DIAGCPS = 0
        THEN(2)      STATE_DIAGCPS = STEP_1
        ENDIF(2)
ENDIF(1)
```

Canister load diagnosis

Description:

Functional check of the CPS based on the ACF- load degree. The “Canister Load diagnosis” is calculated permanently until the functional check CPS is finished (LV_END_DIAGCPS = 1).

Overview:



// calculation of "Canister load diagnosis" is continued parallel to calculation of step 2 (and afterwards step 3) until the functional check is finished

Formula section:

```
IF(1)       LV_DI_DIAGCPS = 0          AND
            LV_INH_DIAGCPS = 0
THEN(1)     LV_CDN_DIAG_DIAGCPS = 1
    IF(2)   LV_EOL_CPS = 1
    THEN(2)IF(3a)     LV_IS = 1                        AND
                      LV_TQ_ISC_I_TQ_PSTE = 0
            THEN(3a)  STATE_DIAGCPS = STEP_2
            ELSE(3a)  STATE_DIAGCPS = STEP_INIT
            ENDIF(3a)
    ELSE(2)IF(3b)     CL_MMV >= C_CL_MMV_DIAGCPS
            THEN(3b) IF(3c)  STATE_OPM_ENG_CP = ENG_MODE_0
                     THEN(3c)
                            IF(3d)      T_CL_MMV_DIAGCPS=0
                            THEN(3d)
                                ERR_SYM_DIAGCPS = NO_SYM
                                LV_ERR_DIAGCPS = 0
                                LV_END_DIAG_DIAGCPS = 1
                                CL_MMV_SAE = CL_MMV
                                SUM_DIAG_DIAGCPS_SAE = ~~FFH~~ 0H
                                MAF_DIAGCPS_SAE = 0
                                MAF_DIAGCPS_THD_SAE = FFFFH
                                STATE_DIAGCPS_RBM = STEP_1
                                STATE_DIAGCPS = LOCK_STEP
                            ELSE(3d)
                                T_CL_MMV_DIAGCPS - -
                                IF(3e)  TCO > C_TCO_MIN_DIAGCPS       AND
                                        LV_IS = 1                     AND
                                        LV_TQ_ISC_I_TQ_PSTE = 0       AND
                                        T_DLY_PURGE_DIAGCPS = 0       AND
                                        CL_MMV < C_CL_MMV_DIAGCPS_VLD
                                THEN(3e)    CL_MMV_SAE = FFFFH
                                            SUM_DIAG_DIAGCPS_SAE=~~FFH~~0H
                                            MAF_DIAGCPS_SAE = 0
                                            MAF_DIAGCPS_THD_SAE = FFFFH
```

```
            IF(3f)  STATE_DIAGCPS_OLD = STEP_1
            THEN(3f)        STATE_DIAGCPS = STEP_2
            ELSE(3f)        STATE_DIAGCPS =
                            STATE_DIAGCPS_OLD
            ENDIF(3f)
        ELSE(3e) IF(3g)     STATE_DIAGCPS !=
                            STEP_1
             THEN(3g) LV_CDN_DIAG_DIAGCPS = 0
                      STATE_DIAGCPS = STEP_INIT
             ENDIF(3g)
        ENDIF(3e)
    ENDIF(3d)
ELSE(3c)
    IF(3h)  T_CL_MMV_DIAGCPS_2 = 0
    THEN(3h)
        ERR_SYM_DIAGCPS = NO_SYM
        LV_ERR_DIAGCPS = 0
        LV_END_DIAG_DIAGCPS = 1
        CL_MMV_SAE = CL_MMV
        SUM_DIAG_DIAGCPS_SAE = ~~FFH~~ 0H
        MAF_DIAGCPS_SAE = 0
        MAF_DIAGCPS_THD_SAE = FFFFH
        STATE_DIAGCPS_RBM = STEP_1
        STATE_DIAGCPS = LOCK_STEP
    ELSE(3h)
        T_CL_MMV_DIAGCPS_2 - -
        IF(3i)  TCO > C_TCO_MIN_DIAGCPS        AND
                LV_IS = 1                      AND
                LV_TQ_ISC_I_TQ_PSTE = 0        AND
                T_DLY_PURGE_DIAGCPS = 0        AND
                CL_MMV < C_CL_MMV_DIAGCPS_VLD
        THEN(3i)        CL_MMV_SAE = FFFFH
                        SUM_DIAG_DIAGCPS_SAE=~~FFH~~0H
                        MAF_DIAGCPS_SAE = 0
                        MAF_DIAGCPS_THD_SAE = FFFFH
```

```
                        IF(3j)  STATE_DIAGCPS_OLD = STEP_1
                        THEN(3j)    STATE_DIAGCPS = STEP_2
                        ELSE(3j)    STATE_DIAGCPS =
                                    STATE_DIAGCPS_OLD
                        ENDIF(3j)
                   ELSE(3i) IF(3k)  STATE_DIAGCPS !=
                                    STEP_1
                          THEN(3k)  LV_CDN_DIAG_DIAGCPS = 0
                                    STATE_DIAGCPS = STEP_INIT
                          ENDIF(3k)
                   ENDIF(3i)
              ENDIF(3h)
     ENDIF(3c)
ELSE(3b)     T_CL_MMV_DIAGCPS = C_T_CL_MMV_DIAGCPS
             T_CL_MMV_DIAGCPS_2 = C_T_CL_MMV_DIAGCPS_2
     IF(4a)  TCO > C_TCO_MIN_DIAGCPS       AND
             LV_IS = 1                     AND
             LV_TQ_ISC_I_TQ_PSTE = 0       AND
             T_DLY_PURGE_DIAGCPS = 0       AND
             CL_MMV < C_CL_MMV_DIAGCPS_VLD
     THEN(4a)     CL_MMV_SAE = FFFFH
                  SUM_DIAG_DIAGCPS_SAE = ~~FFH~~ 0H
                  MAF_DIAGCPS_SAE = 0
                  MAF_DIAGCPS_THD_SAE = FFFFH
             IF(5a)  STATE_DIAGCPS_OLD = STEP_1
             THEN(5a)     STATE_DIAGCPS = STEP_2
             ELSE(5a)     STATE_DIAGCPS =
                          STATE_DIAGCPS_OLD
             ENDIF(5a)
     ELSE(4a)     IF(5b)  STATE_DIAGCPS != STEP_1
                  THEN(5b)     LV_CDN_DIAG_DIAGCPS = 0
                               STATE_DIAGCPS = STEP_INIT
                  ENDIF(5b)
     ENDIF(4a)
```

```
            ENDIF(3b)
        ENDIF(2)
ELSE(1)     LV_CDN_DIAG_DIAGCPS = 0
            STATE_DIAGCPS = STEP_INIT
ENDIF(1)
```

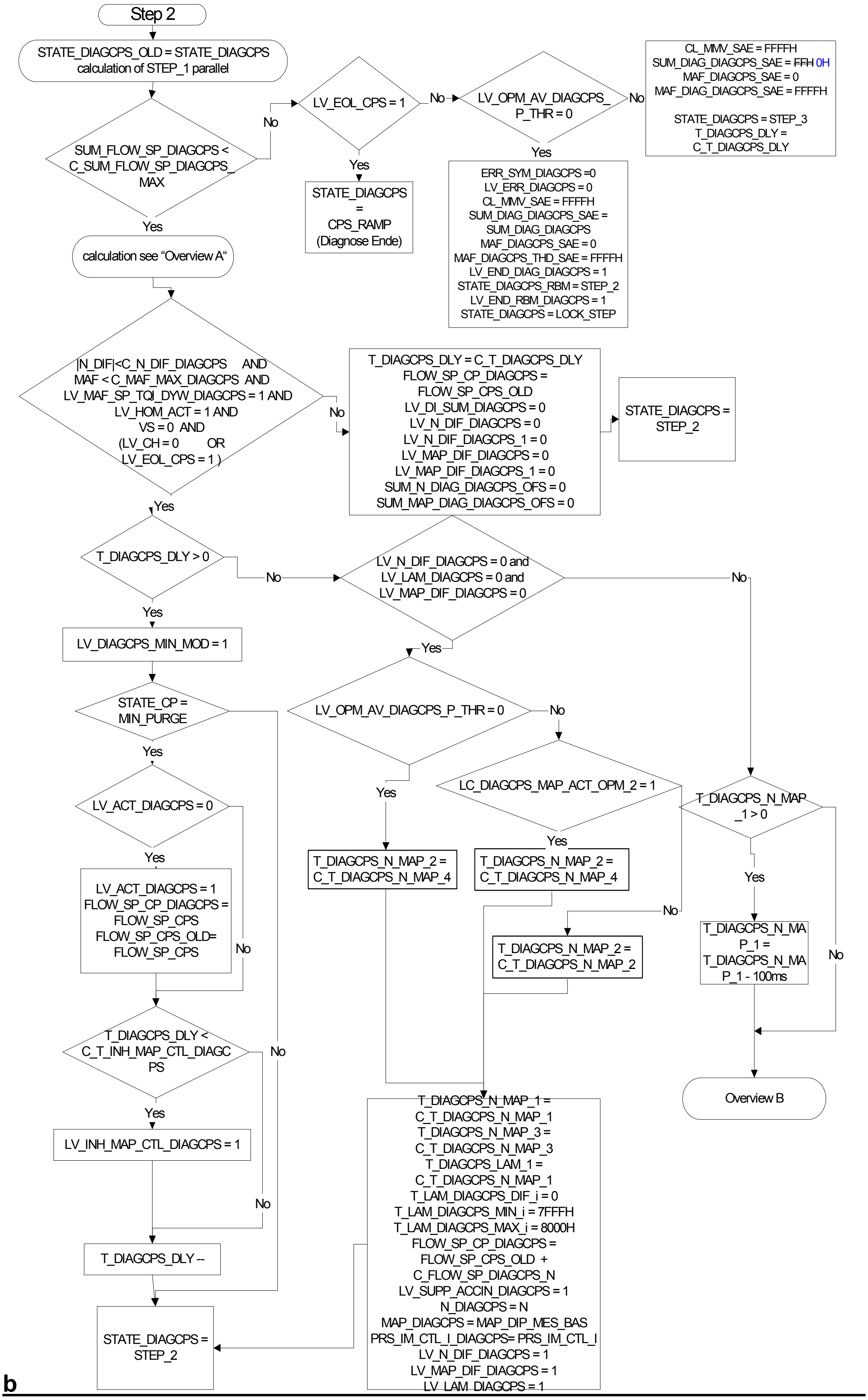
STEP 2: N-DIF-/LAM-/MAP-Diagnosis

Description:

In step 2, the CPS is evaluated based on the engine speed change (throttled mode), MAP change (unthrottled mode) and Lambda deviation at IS. To this effect, the CPS is opened for a short time and the engine speed monitored for a certain period.

After step 2 has been enabled for the first time, it is requested during each IS phase. This is repeated as long as a result has been reached in step 2. The function is not bound to one IS phase, but can be distributed to several IS phases.

Overview:


Step 2
STATE_DIAGCPS_OLD = STATE_DIAGCPS
calculation of STEP_1 parallel
SUM_FLOW_SP_DIAGCPS < C_SUM_FLOW_SP_DIAGCPS_MAX
No
Yes
LV_EOL_CPS = 1
No
LV_OPM_AV_DIAGCPS_P_THR = 0
No
CL_MMV_SAE = FFFFH
SUM_DIAG_DIAGCPS_SAE = ~~FFH~~ 0H
MAF_DIAGCPS_SAE = 0
MAF_DIAG_DIAGCPS_SAE = FFFFH
STATE_DIAGCPS = STEP_3
T_DIAGCPS_DLY = C_T_DIAGCPS_DLY
Yes
STATE_DIAGCPS = CPS_RAMP (Diagnose Ende)
Yes
ERR_SYM_DIAGCPS =0
LV_ERR_DIAGCPS = 0
CL_MMV_SAE = FFFFH
SUM_DIAG_DIAGCPS_SAE = SUM_DIAG_DIAGCPS
MAF_DIAGCPS_SAE = 0
MAF_DIAGCPS_THD_SAE = FFFFH
LV_END_DIAG_DIAGCPS = 1
STATE_DIAGCPS_RBM = STEP_2
LV_END_RBM_DIAGCPS = 1
STATE_DIAGCPS = LOCK_STEP
calculation see "Overview A"
\|N_DIF\|<C_N_DIF_DIAGCPS AND MAF < C_MAF_MAX_DIAGCPS AND LV_MAF_SP_TQI_DYW_DIAGCPS = 1 AND LV_HOM_ACT = 1 AND VS = 0 AND (LV_CH = 0 OR LV_EOL_CPS = 1)
No
T_DIAGCPS_DLY = C_T_DIAGCPS_DLY
FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS_OLD
LV_DI_SUM_DIAGCPS = 0
LV_N_DIF_DIAGCPS = 0
LV_N_DIF_DIAGCPS_1 = 0
LV_MAP_DIF_DIAGCPS = 0
LV_MAP_DIF_DIAGCPS_1 = 0
SUM_N_DIAG_DIAGCPS_OFS = 0
SUM_MAP_DIAG_DIAGCPS_OFS = 0
STATE_DIAGCPS = STEP_2
Yes
T_DIAGCPS_DLY > 0
No
LV_N_DIF_DIAGCPS = 0 and LV_LAM_DIAGCPS = 0 and LV_MAP_DIF_DIAGCPS = 0
No
Yes
LV_DIAGCPS_MIN_MOD = 1
Yes
LV_OPM_AV_DIAGCPS_P_THR = 0
No
STATE_CP = MIN_PURGE
Yes
LC_DIAGCPS_MAP_ACT_OPM_2 = 1
T_DIAGCPS_N_MAP_1 > 0
Yes
LV_ACT_DIAGCPS = 0
Yes
T_DIAGCPS_N_MAP_2 = C_T_DIAGCPS_N_MAP_4
Yes
T_DIAGCPS_N_MAP_2 = C_T_DIAGCPS_N_MAP_4
Yes
Yes
LV_ACT_DIAGCPS = 1
FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS
FLOW_SP_CPS_OLD= FLOW_SP_CPS
No
No
T_DIAGCPS_N_MAP_2 = C_T_DIAGCPS_N_MAP_2
T_DIAGCPS_N_MAP_1 = T_DIAGCPS_N_MAP_1 - 100ms
No
T_DIAGCPS_DLY < C_T_INH_MAP_CTL_DIAGCPS
No
Overview B
Yes
T_DIAGCPS_N_MAP_1 = C_T_DIAGCPS_N_MAP_1
T_DIAGCPS_N_MAP_3 = C_T_DIAGCPS_N_MAP_3
T_DIAGCPS_LAM_1 = C_T_DIAGCPS_N_MAP_1
T_LAM_DIAGCPS_DIF_i = 0
T_LAM_DIAGCPS_MIN_i = 7FFFH
T_LAM_DIAGCPS_MAX_i = 8000H
FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS_OLD + C_FLOW_SP_DIAGCPS_N
LV_SUPP_ACCIN_DIAGCPS = 1
N_DIAGCPS = N
MAP_DIAGCPS = MAP_DIP_MES_BAS
PRS_IM_CTL_I_DIAGCPS= PRS_IM_CTL_I
LV_N_DIF_DIAGCPS = 1
LV_MAP_DIF_DIAGCPS = 1
LV_LAM_DIAGCPS = 1
LV_INH_MAP_CTL_DIAGCPS = 1
No
T_DIAGCPS_DLY --
STATE_DIAGCPS = STEP_2
b

Formula section:

```
STATE_DIAGCPS_OLD = STATE_DIAGCPS
calculate STEP_1 parallely
IF(1)        SUM_FLOW_SP_DIAGCPS < C_SUM_FLOW_SP_DIAGCPS_MAX
THEN(1)IF(2a)    LV_LAM_SUM_DIAGCPS = 0 and LV_LAM_DIAGCPS = 1
     THEN(2a)
        IF(3a)  T_DIAGCPS_LAM_1 > 0
        THEN(3a)T_DIAGCPS_LAM_1 –
                // Lamb-Monitoring:
                TI_LAM_DIAGCPS_MAX_1 =
                MAX (TI_LAM_DIAGCPS_MAX_1 $_{N-1}$;TI_LAM_COR_1 $_{N}$)
                TI_LAM_DIAGCPS_MAX_2 =
                MAX (TI_LAM_DIAGCPS_MAX_2 $_{N-1}$; TI_LAM_COR_2 $_{N}$)
                TI_LAM_DIAGCPS_MIN_1 =
                MIN (TI_LAM_DIAGCPS_MIN_1 $_{N-1}$; TI_LAM_COR_1 $_{N}$)
                TI_LAM_DIAGCPS_MIN_2 =
                MIN (TI_LAM_DIAGCPS_MIN_2 $_{N-1}$; TI_LAM_COR_2 $_{N}$)
        ELSE(3a)LV_LAM_DIAGCPS = 0
                TI_LAM_DIAGCPS_DIF_1 = TI_LAM_DIAGCPS_MAX_1-
                                            TI_LAM_DIAGCPS_MIN_1
                TI_LAM_DIAGCPS_DIF_2 = TI_LAM_DIAGCPS_MAX_2-
                                                TI_LAM_DIAGCPS_MIN_2
                IF(4a)  TI_LAM_DIAGCPS_DIF_1 > C_TI_LAM_DIAGCPS_DIF   OR
                        TI_LAM_DIAGCPS_DIF_2 > C_TI_LAM_DIAGCPS_DIF
                THEN(4a) LV_LAM_SUM_DIAGCPS = 1
                        Jump to STEP_2B
                ENDIF(4a)
        ENDIF(3a)
    ENDIF(2a)
    IF(2b)      | N_DIF |  < C_N_DIF_DIAGCPS          AND
                MAF < C_MAF_MAX_DIAGCPS               AND
                LV_MAF_SP_TQI_DYW_DIAGCPS = 1         AND
                LV_HOM_ACT = 1                        AND
                VS = 0                                AND
                (LV_CH = 0              OR
                LV_EOL_CPS = 1)
    THEN(2b)
        IF(3b)  T_DIAGCPS_DLY > 0
```

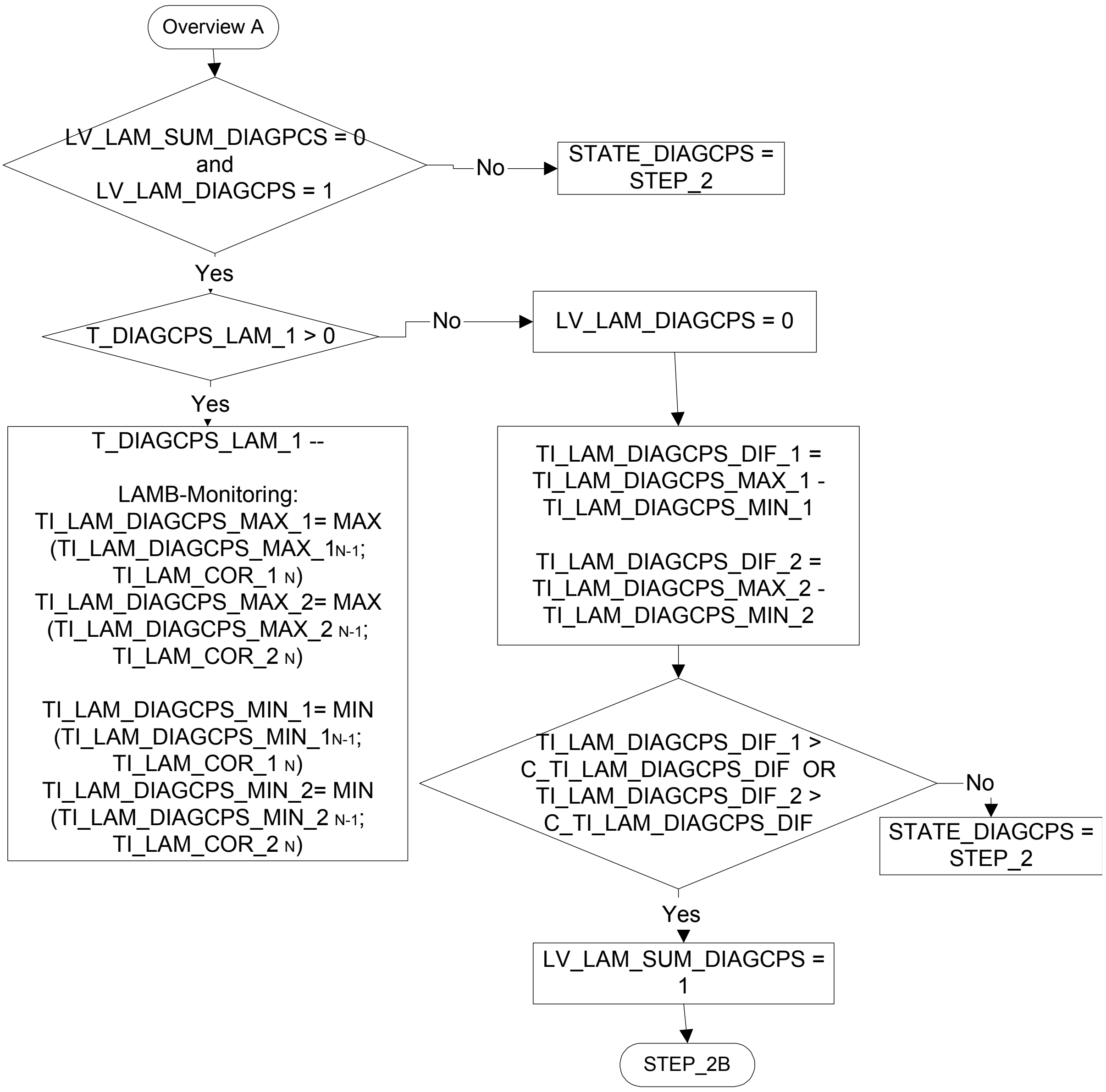
```
THEN(3b)LV_DIAGCPS_MIN_MOD = 1
        IF(4b)  STATE_CP = MIN_PURGE
        THEN(4b)
                IF(5a)    LV_ACT_DIAGCPS = 0
                THEN(5a)LV_ACT_DIAGCPS = 1
                          FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS
                          FLOW_SP_CPS_OLD = FLOW_SP_CPS
                ENDIF(5a)
                IF(5b)    T_DIAGCPS_DLY < C_T_INH_MAP_CTL_DIAGCPS
                THEN(5b) LV_INH_MAP_CTL_DIAGCPS = 1
                ENDIF(5b)
                T_DIAGCPS_DLY = T_DIAGCPS_DLY –
        ENDIF(4b)
ELSE(3b)
        IF(4c)  LV_N_DIF_DIAGCPS = 0          AND
                LV_LAM_DIAGCPS = 0            AND
                LV_MAP_DIF_DIAGCPS = 0
        THEN(4c)
                IF(5c)  LV_OPM_AV_DIAGCPS_P_THR = 0

                Then(5c) T_DIAGCPS_N_MAP_2 = C_T_DIAGCPS_N_MAP_4
                Else(5c)
                    If(6a) LC_DIAGCPS_MAP_ACT_OPM_2 = 1
                    Then(6a) T_DIAGCPS_N_MAP_2 = C_T_DIAGCPS_N_MAP_4
                    Else(6a)  T_DIAGCPS_N_MAP_2 = C_T_DIAGCPS_N_MAP_2
                    Endif(6a)
                Endif(5c)
                T_DIAGCPS_N_MAP_1 = C_T_DIAGCPS_N_MAP_1
                T_DIAGCPS_N_MAP_3 = C_T_DIAGCPS_N_MAP_3
                T_DIAGCPS_LAM_1 = C_T_DIAGCPS_N_MAP_1
                TI_LAM_DIAGCPS_MIN_i = 7FFFH
                TI_LAM_DIAGCPS_MAX_i = 8000H
                FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS_OLD +
                                                C_FLOW_SP_DIAGCPS_N
                LV_SUPP_ACCIN_DIAGCPS = 1
                N_DIAGCPS = N
                MAP_DIAGCPS = MAP_DIP_MES_BAS
                        PRS_IM_CTL_I_DIAGCPS = PRS_IM_CTL_I
                LV_N_DIF_DIAGCPS = 1
```

```
            LV_MAP_DIF_DIAGCPS = 1
            LV_LAM_DIAGCPS = 1
        ELSE(4c)
            IF(5d) T_DIAGCPS_N_MAP_1 > 0
            THEN(5d)T_DIAGCPS_N_MAP_1 = T_DIAGCPS_N_MAP_1   --
            ENDIF(5d)
            IF(5e)   T_DIAGCPS_N_MAP_2 > 0
            THEN(5e)T_DIAGCPS_N_MAP_2 = T_DIAGCPS_N_MAP_2 --
                IF(6b)     T_DIAGCPS_N_MAP_3 > 0
                THEN(6b)T_DIAGCPS_N_MAP_3 = T_DIAGCPS_N_MAP_3 --
                ELSE(6b)
                    IF(7a)FLOW_SP_CP_DIAGCPS >
                        FLOW_SP_CPS_OLD
                    THEN(7a)  FLOW_SP_CP_DIAGCPS =
                        FLOW_SP_CPS_OLD
                    ENDIF(7a)
                ENDIF(6b)
                IF(6c)     LV_OPM_AV_DIAGCPS_P_THR = 0
                THEN(6c)
                    IF(7b)       MAP_DIP_MES_BAS – MAP_DIAGCPS >
                    C_MAP_DIAGCPS_1 OR             /to use for MAF-System
                    PRS_IM_CTL_I_DIAGCPS - PRS_IM_CTL_I >
                    C_PRS_IM_CTL_I_DIAGCPS         /to use for MAP-System
                    THEN(7b) SUM_MAP_DIAG_DIAGCPS_OFS ++
                        IF(8a)SUM_MAP_DIAG_DIAGCPS_OFS >
                            C_SUM_MAP_DIAG_DIAGCPS_OFS
                        Then(8a) LV_MAP_DIF_DIAGCPS_1 = 1
                        Endif(8a)
                    ENDIF(7b)
                ELSE(6c)
                    If(7c)       LC_DIAGCPS_MAP_ACT_OPM_2 = 1
                    Then(7c)
                        IF(8b) MAP_DIP_MES_BAS – MAP_DIAGCPS >
                            C_MAP_DIAGCPS_2    OR
                            PRS_IM_CTL_I_DIAGCPS - PRS_IM_CTL_I >
                            C_PRS_IM_CTL_I_DIAGCPS
                        THEN(8b) SUM_MAP_DIAG_DIAGCPS_OFS ++
                            IF(9a) SUM_MAP_DIAG_DIAGCPS_OFS >
                                C_SUM_MAP_DIAG_DIAGCPS_OFS
                            Then(9a) LV_MAP_DIF_DIAGCPS_1 = 1
```

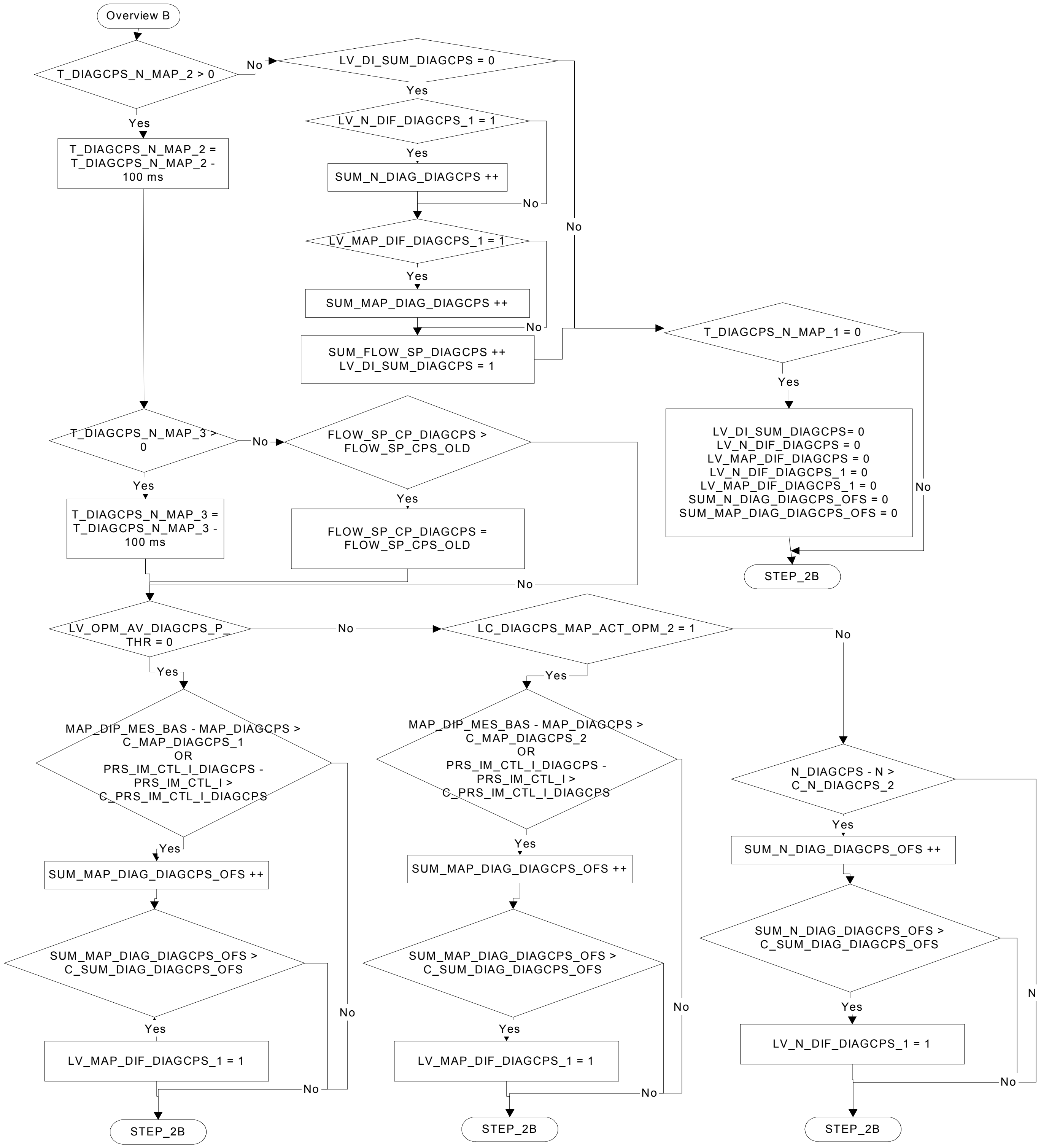
```
                            Endif(9a)
                        ENDIF(8b)
                    Else(7c)
                        IF(8c) N_DIAGCPS – N > C_N_DIAGCPS_2
                        THEN(8c)   SUM_N_DIAG_DIAGCPS_OFS ++
                            IF(9b) SUM_N_DIAG_DIAGCPS_OFS >
                                C_SUM_N_DIAG_DIAGCPS_OFS
                            Then(9b) LV_N_DIF_DIAGCPS_1 = 1
                            Endif(9b)
                        ENDIF(8c)
                    Endif(7c)
                ENDIF(6c)
            ELSE(5e)
                IF(6d)   LV_DI_SUM_DIAGCPS = 0
                THEN(6d)
                    IF(7d)      LV_N_DIF_DIAGCPS_1 = 1
                    THEN(7d) SUM_N_DIAG_DIAGCPS ++
                    ENDIF(7d)
                    IF(7e) LV_MAP_DIF_DIAGCPS_1 = 1
                    THEN(7e) SUM_MAP_DIAG_DIAGCPS ++
                    ENDIF(7e)
                SUM_FLOW_SP_DIAGCPS  ++
                LV_DI_SUM_DIAGCPS = 1
                ENDIF(6d)
                IF(6e)   T_DIAGCPS_N_MAP_1 = 0
                THEN(6e)        LV_DI_SUM_DIAGCPS = 0
                                LV_N_DIF_DIAGCPS = 0
                                LV_N_DIF_DIAGCPS_1 = 0
                                LV_MAP_DIF_DIAGCPS = 0
                                LV_MAP_DIF_DIAGCPS_1 = 0
                                SUM_MAP_DIAG_DIAGCPS_OFS = 0
                                SUM_N_DIAG_DIAGCPS_OFS = 0
                ENDIF(6e)
            ENDIF(5e)
            Jump to “STEP_2B”
        ENDIF(4c)
    ENDIF(3b)
ELSE(2b)        T_DIAGCPS_DLY = C_T_DIAGCPS_DLY
                FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS_OLD
```

```
                              LV_DI_SUM_DIAGCPS = 0
                              LV_N_DIF_DIAGCPS = 0
                              LV_N_DIF_DIAGCPS_1 = 0
                              LV_MAP_DIF_DIAGCPS = 0
                              LV_MAP_DIF_DIAGCPS_1 = 0
                              SUM_MAP_DIAG_DIAGCPS_OFS = 0
                              SUM_N_DIAG_DIAGCPS_OFS = 0
      ENDIF(2b)
  ELSE(1)IF(2c)               LV_EOL_CPS = 1
         THEN(2c)             STATE_DIAGCPS = CPS_RAMP
                              (Diagnose Ende)
         ELSE(2c)
            IF(3c)            LV_OPM_AV_DIAGCPS_P_THR = 0
            THEN(3c)ERR_SYM_DIAGCPS = NO_SYM
                              LV_ERR_DIAGCPS = 0
                              CL_MMV_SAE = ~~FFFH~~ FFFFH
                              SUM_DIAG_DIAGCPS_SAE = SUM_DIAG_DIAGCPS
                              MAF_DIAGCPS_SAE = 0
                              MAF_DIAGCPS_THD_SAE = FFFFH
                              LV_END_DIAG_DIAGCPS = 1
                              LV_END_RBM_DIAGCPS = 1
                              STATE_DIAGCPS_RBM = STEP_2
                              STATE_DIAGCPS = LOCK_STEP
            ELSE(3c) CL_MMV_SAE = FFFFH
                              SUM_DIAG_DIAGCPS_SAE = ~~FFFFH~~ 0H
                              MAF_DIAGCPS_SAE = 0
                              MAF_DIAGCPS_THD_SAE = FFFFH
                              STATE_DIAGCPS = STEP_3
                              T_DIAGCPS_DLY = C_T_DIAGCPS_DLY
            ENDIF(3c)
         ENDIF(2c)
  ENDIF(1)
```

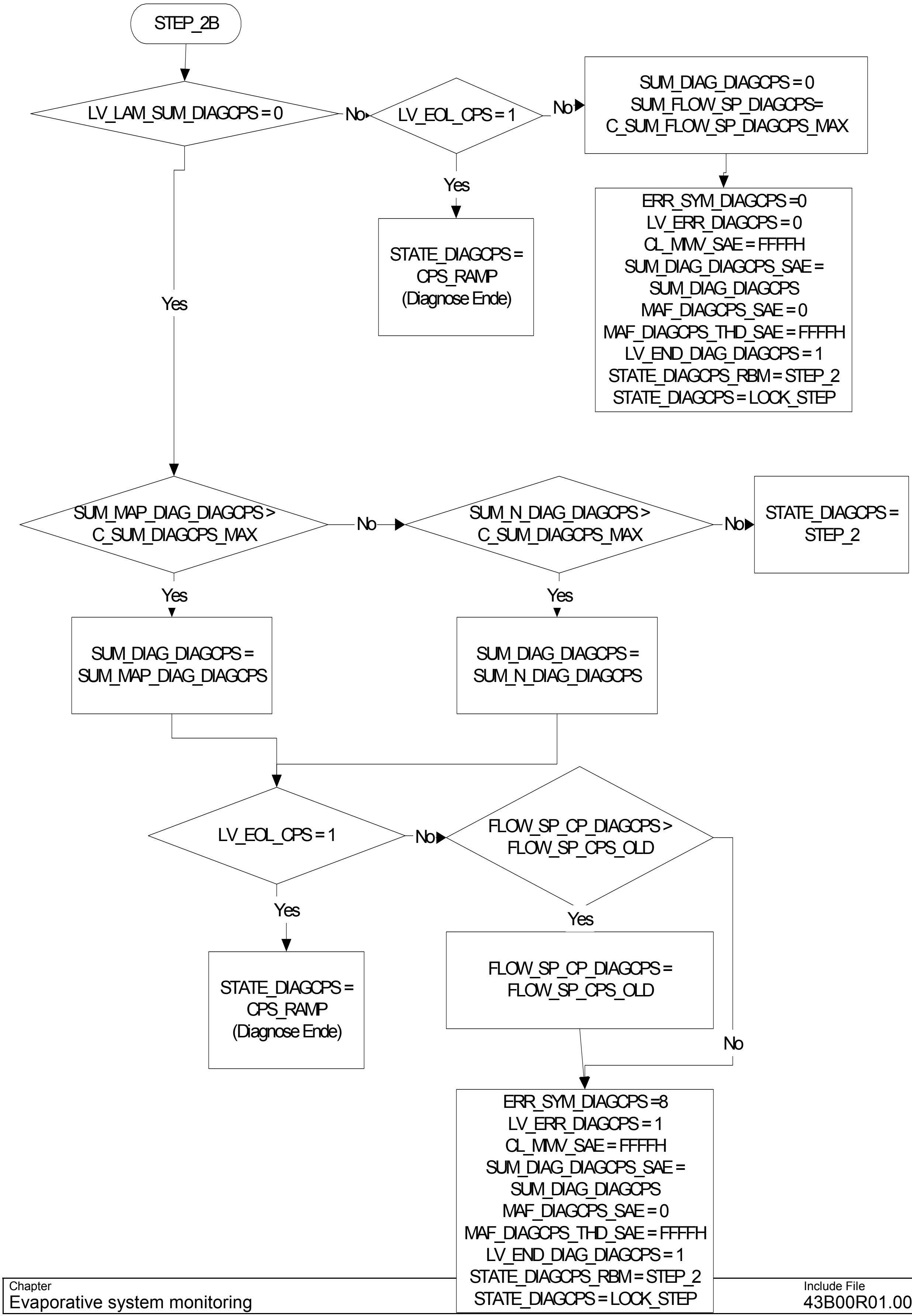
Overview A:


Overview A
LV_LAM_SUM_DIAGPCS = 0
and
LV_LAM_DIAGCPS = 1
No
STATE_DIAGCPS = STEP_2
Yes
T_DIAGCPS_LAM_1 > 0
No
LV_LAM_DIAGCPS = 0
Yes
T_DIAGCPS_LAM_1 --
LAMB-Monitoring:
TI_LAM_DIAGCPS_MAX_1= MAX (TI_LAM_DIAGCPS_MAX_1N-1; TI_LAM_COR_1 N)
TI_LAM_DIAGCPS_MAX_2= MAX (TI_LAM_DIAGCPS_MAX_2 N-1; TI_LAM_COR_2 N)
TI_LAM_DIAGCPS_MIN_1= MIN (TI_LAM_DIAGCPS_MIN_1N-1; TI_LAM_COR_1 N)
TI_LAM_DIAGCPS_MIN_2= MIN (TI_LAM_DIAGCPS_MIN_2 N-1; TI_LAM_COR_2 N)
TI_LAM_DIAGCPS_DIF_1 = TI_LAM_DIAGCPS_MAX_1 - TI_LAM_DIAGCPS_MIN_1
TI_LAM_DIAGCPS_DIF_2 = TI_LAM_DIAGCPS_MAX_2 - TI_LAM_DIAGCPS_MIN_2
TI_LAM_DIAGCPS_DIF_1 > C_TI_LAM_DIAGCPS_DIF OR TI_LAM_DIAGCPS_DIF_2 > C_TI_LAM_DIAGCPS_DIF
No
STATE_DIAGCPS = STEP_2
Yes
LV_LAM_SUM_DIAGCPS = 1
STEP_2B

Overview “B”:


Overview B
T_DIAGCPS_N_MAP_2 > 0
No
Yes
T_DIAGCPS_N_MAP_2 = T_DIAGCPS_N_MAP_2 - 100 ms
LV_DI_SUM_DIAGCPS = 0
Yes
LV_N_DIF_DIAGCPS_1 = 1
Yes
SUM_N_DIAG_DIAGCPS ++
No
No
LV_MAP_DIF_DIAGCPS_1 = 1
Yes
SUM_MAP_DIAG_DIAGCPS ++
No
SUM_FLOW_SP_DIAGCPS ++
LV_DI_SUM_DIAGCPS = 1
T_DIAGCPS_N_MAP_1 = 0
Yes
LV_DI_SUM_DIAGCPS= 0
LV_N_DIF_DIAGCPS = 0
LV_MAP_DIF_DIAGCPS = 0
LV_N_DIF_DIAGCPS_1 = 0
LV_MAP_DIF_DIAGCPS_1 = 0
SUM_N_DIAG_DIAGCPS_OFS = 0
SUM_MAP_DIAG_DIAGCPS_OFS = 0
No
STEP_2B
T_DIAGCPS_N_MAP_3 > 0
No
Yes
T_DIAGCPS_N_MAP_3 = T_DIAGCPS_N_MAP_3 - 100 ms
FLOW_SP_CP_DIAGCPS > FLOW_SP_CPS_OLD
Yes
FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS_OLD
No
LV_OPM_AV_DIAGCPS_P_THR = 0
No
LC_DIAGCPS_MAP_ACT_OPM_2 = 1
No
Yes
Yes
MAP_DIP_MES_BAS - MAP_DIAGCPS > C_MAP_DIAGCPS_1 OR PRS_IM_CTL_I_DIAGCPS - PRS_IM_CTL_I > C_PRS_IM_CTL_I_DIAGCPS
Yes
SUM_MAP_DIAG_DIAGCPS_OFS ++
SUM_MAP_DIAG_DIAGCPS_OFS > C_SUM_DIAG_DIAGCPS_OFS
Yes
LV_MAP_DIF_DIAGCPS_1 = 1
No
No
STEP_2B
MAP_DIP_MES_BAS - MAP_DIAGCPS > C_MAP_DIAGCPS_2 OR PRS_IM_CTL_I_DIAGCPS - PRS_IM_CTL_I > C_PRS_IM_CTL_I_DIAGCPS
Yes
SUM_MAP_DIAG_DIAGCPS_OFS ++
SUM_MAP_DIAG_DIAGCPS_OFS > C_SUM_DIAG_DIAGCPS_OFS
Yes
LV_MAP_DIF_DIAGCPS_1 = 1
No
No
STEP_2B
N_DIAGCPS - N > C_N_DIAGCPS_2
Yes
SUM_N_DIAG_DIAGCPS_OFS ++
SUM_N_DIAG_DIAGCPS_OFS > C_SUM_DIAG_DIAGCPS_OFS
Yes
LV_N_DIF_DIAGCPS_1 = 1
N
No
STEP_2B

STEP 2B:


STEP_2B
LV_LAM_SUM_DIAGCPS = 0
No
LV_EOL_CPS = 1
No
SUM_DIAG_DIAGCPS = 0
SUM_FLOW_SP_DIAGCPS=
C_SUM_FLOW_SP_DIAGCPS_MAX
Yes
STATE_DIAGCPS =
CPS_RAMP
(Diagnose Ende)
ERR_SYM_DIAGCPS =0
LV_ERR_DIAGCPS = 0
CL_MMV_SAE = FFFFH
SUM_DIAG_DIAGCPS_SAE =
SUM_DIAG_DIAGCPS
MAF_DIAGCPS_SAE = 0
MAF_DIAGCPS_THD_SAE = FFFFH
LV_END_DIAG_DIAGCPS = 1
STATE_DIAGCPS_RBM = STEP_2
STATE_DIAGCPS = LOCK_STEP
Yes
SUM_MAP_DIAG_DIAGCPS >
C_SUM_DIAGCPS_MAX
No
SUM_N_DIAG_DIAGCPS >
C_SUM_DIAGCPS_MAX
No
STATE_DIAGCPS =
STEP_2
Yes
Yes
SUM_DIAG_DIAGCPS =
SUM_MAP_DIAG_DIAGCPS
SUM_DIAG_DIAGCPS =
SUM_N_DIAG_DIAGCPS
LV_EOL_CPS = 1
No
FLOW_SP_CP_DIAGCPS >
FLOW_SP_CPS_OLD
Yes
Yes
STATE_DIAGCPS =
CPS_RAMP
(Diagnose Ende)
FLOW_SP_CP_DIAGCPS =
FLOW_SP_CPS_OLD
No
ERR_SYM_DIAGCPS =8
LV_ERR_DIAGCPS = 1
CL_MMV_SAE = FFFFH
SUM_DIAG_DIAGCPS_SAE =
SUM_DIAG_DIAGCPS
MAF_DIAGCPS_SAE = 0
MAF_DIAGCPS_THD_SAE = FFFFH
LV_END_DIAG_DIAGCPS = 1
STATE_DIAGCPS_RBM = STEP_2
STATE_DIAGCPS = LOCK_STEP

STEP 2B:

```
IF(1)   LV_LAM_SUM_DIAGCPS = 0
THEN(1)IF(2a)       ~~SUM_MAP_DIAG_DIAGCPS >= C_SUM_DIAGCPS_MAX~~
                    SUM_MAP_DIAG_DIAGCPS > C_SUM_DIAGCPS_MAX
        THEN(2a)    SUM_DIAG_DIAGCPS = SUM_MAP_DIAG_DIAGCPS
        ELSE(2a)    IF(3a) ~~SUM_N_DIAG_DIAGCPS >= C_SUM_DIAGCPS_MAX~~
                           SUM_N_DIAG_DIAGCPS > C_SUM_DIAGCPS_MAX
                    THEN(3a)    SUM_DIAG_DIAGCPS = SUM_N_DIAG_DIAGCPS
                    ELSE(3a)    STATE_DIAGCPS = STEP_2
                    ENDIF(3a)
        ENDIF(2a)
        IF(2b)      ~~SUM_DIAG_DIAGCPS > = C_SUM_DIAGCPS_MAX~~
                    SUM_DIAG_DIAGCPS > C_SUM_DIAGCPS_MAX
        THEN(2b)    IF(3b)  LV_EOL_CPS = 1
                    THEN(3b)    STATE_DIAGCPS = CPS_RAMP
                                (Diagnose Ende)
                    ELSE(3b)    IF(4a)  FLOW_SP_CP_DIAGCPS > FLOW_SP_CPS_OLD
                                THEN(4a)    FLOW_SP_CP_DIAGCPS =
                                                FLOW_SP_CPS_OLD
                                ENDIF(4a)
                                ERR_SYM_DIAGCPS = SYM_3
                                LV_ERR_DIAGCPS = 1
                                SUM_DIAG_DIAGCPS_SAE = SUM_DIAG_DIAGCPS
                                CL_MMV_SAE = FFFFH
                                MAF_DIAGCPS_SAE = 0
                                MAF_DIAGCPS_THD_SAE = FFFFH
                                LV_END_DIAG_DIAGCPS = 1
                                STATE_DIAGCPS_RBM = STEP_2
                                STATE_DIAGCPS = LOCK_STEP
                    ENDIF(3b)

        ELSE(2b)    jump to beginning of STEP_2
        ENDIF(2b)
ELSE(1)IF(2c)       LV_EOL_CPS = 1
        THEN(2c)    STATE_DIAGCPS = CPS_RAMP
                    (Diagnose Ende)
```

```
    ELSE(2c)    SUM_DIAG_DIAGCPS = 0
                SUM_FLOW_SP_DIAGCPS = C_SUM_FLOW_SP_DIAGCPS_MAX
                ERR_SYM_DIAGCPS = 0
                LV_ERR_DIAGCPS = 0
                SUM_DIAG_DIAGCPS_SAE = SUM_DIAG_DIAGCPS
                CL_MMV_SAE = FFFFH
                MAF_DIAGCPS_SAE = 0
                MAF_DIAGCPS_THD_SAE = FFFFH
                LV_END_DIAG_DIAGCPS = 1
                STATE_DIAGCPS_RBM = STEP_2
                STATE_DIAGCPS = LOCK_STEP
    ENDIF(2c)
ENDIF(1)
```

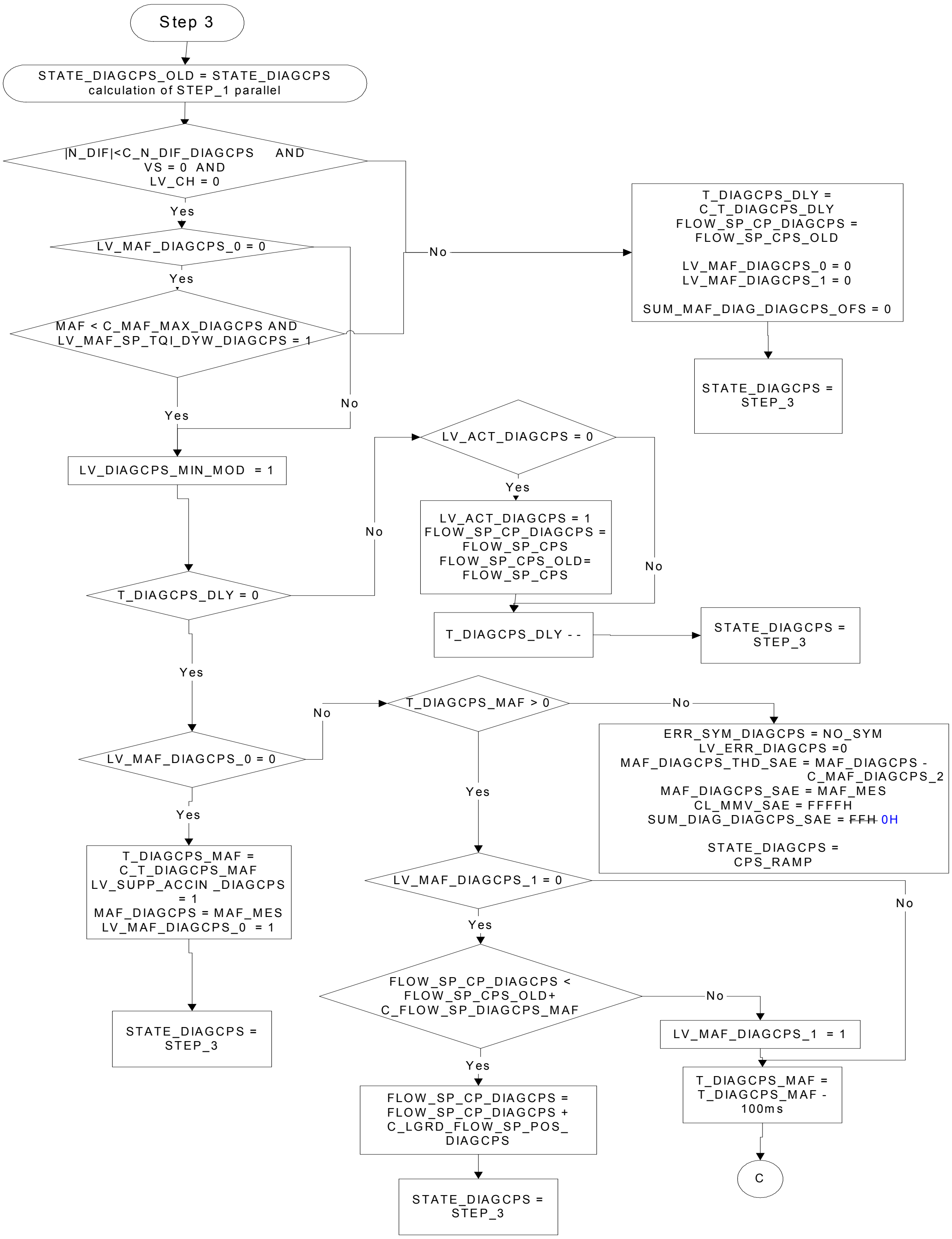
STEP 3: MAF-DIF Diagnosis

Description:

If the functional check is not recognized as OK via step 1 and via step 2, but as not OK via both steps, then step 3 is requested.

In step 3 the CPS is considered on the basis of the measured mass air flow before and during a CPS opening phase.

Overview:

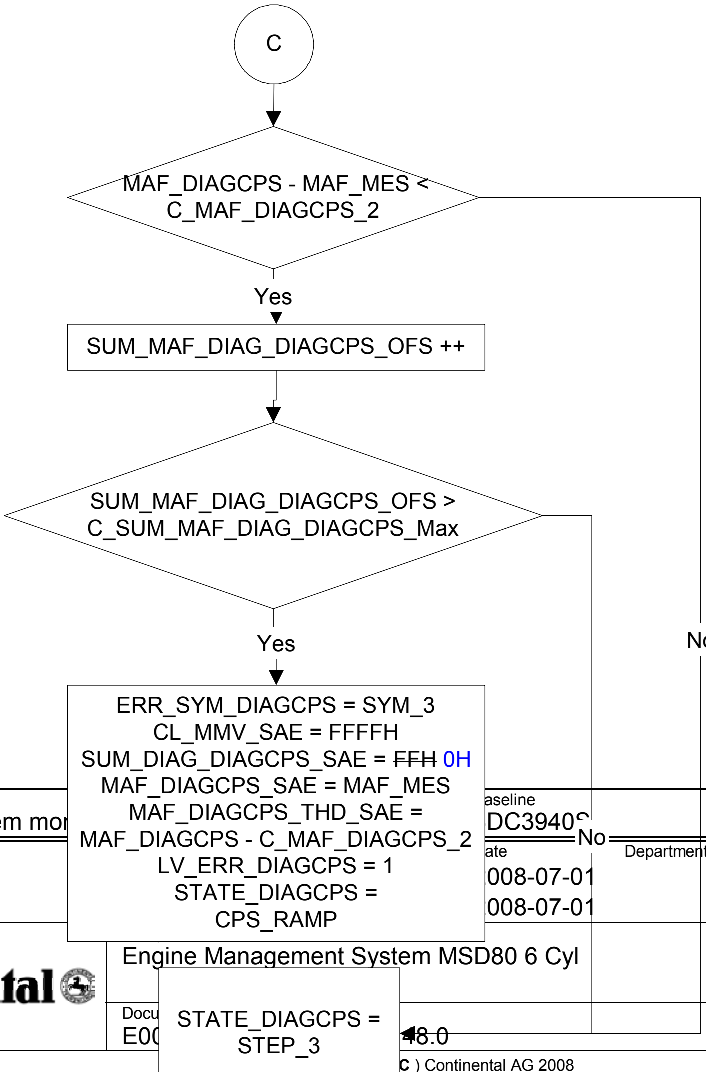


Formula Section:

```
STATE_DIAGCPS_OLD = STATE_DIAGCPS
calculate STEP_1 parallelly
IF(0)     | N_DIF | < C_N_DIF_DIAGCPS          AND
          VS = 0                               AND
          LV_CH = 0                            AND
          [(LV_MAF_DIAGCPS_0 = 0      AND
          MAF < C_MAF_MAX_DIAGCPS AND
          LV_MAF_SP_TQI_DYW_DIAGCPS = 1) OR
          LV_MAF_DIAGCPS_0 = 1]
THEN(0)   LV_DIAGCPS_MIN_MOD=1
              IF(1)   T_DIAGCPS_DLY = 0
              THEN(1)IF(2a)    LV_MAF_DIAGCPS_0 = 0
                     THEN(2a)  T_DIAGCPS_MAF = C_T_DIAGCPS_MAF
                               LV_SUPP_ACCIN_DIAGCPS = 1
                               MAF_DIAGCPS = MAF_MES
                               LV_MAF_DIAGCPS_0 = 1
                     ELSE(2a)  IF(3a)  T_DIAGCPS_MAF > 0
                               THEN(3a)IF(4a) LV_MAF_DIAGCPS_1 = 0    AND
                                              FLOW_SP_CP_DIAGCPS <
                                              FLOW_SP_CPS_OLD +
                                               C_FLOW_SP_DIAGCPS_MAF
                                       THEN(4a)FLOW_SP_CP_DIAGCPS=
                                                   FLOW_SP_CP_DIAGCPS +

                                 C_LGRD_FLOW_SP_POS_DIAGCPS
                                       ELSE(4a)LV_MAF_DIAGCPS_1 = 1
                                               T_DIAGCPS_MAF =
                                               T_DIAGCPS_MAF -100ms
                                       ENDIF(4a)
                                       Jump to C
                                ELSE(3a)    ERR_SYM_DIAGCPS = NO_SYM
                                            MAF_DIAGCPS_SAE = MAF_MES
                                            CL_MMV_SAE = FFFFH
                                            SUM_DIAG_DIAGCPS_SAE = ~~FFH~~ 0H
                                            LV_ERR_DIAGCPS = 0
                                            STATE_DIAGCPS = CPS_RAMP
```

```
                                          MAF_DIAGCPS_THD_SAE=MAF_DIAGCPS-
                                                        C_MAF_DIAGCPS_2
                              ENDIF(3a)
                    ENDIF(2a)
          ELSE(1)IF(2b)       LV_ACT_DIAGCPS = 0
                    THEN(2b)  LV_ACT_DIAGCPS = 1
                              FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS
                              FLOW_SP_CPS_OLD = FLOW_SP_CPS
                    ENDIF(2b)
                    T_DIAGCPS_DLY = T_DIAGCPS_DLY – 100ms
          ENDIF(1)
Else(0)   T_DIAGCPS_DLY = C_T_DIAGCPS_DLY
          FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS_OLD
          LV_MAF_DIAGCPS_0 = 0
          LV_MAF_DIAGCPS_1 = 0
          SUM_MAF_DIAG_DIAGCPS_OFS = 0
          STATE_DIAGCPS = STEP_3
Endif(0)
```

Overview “C”:

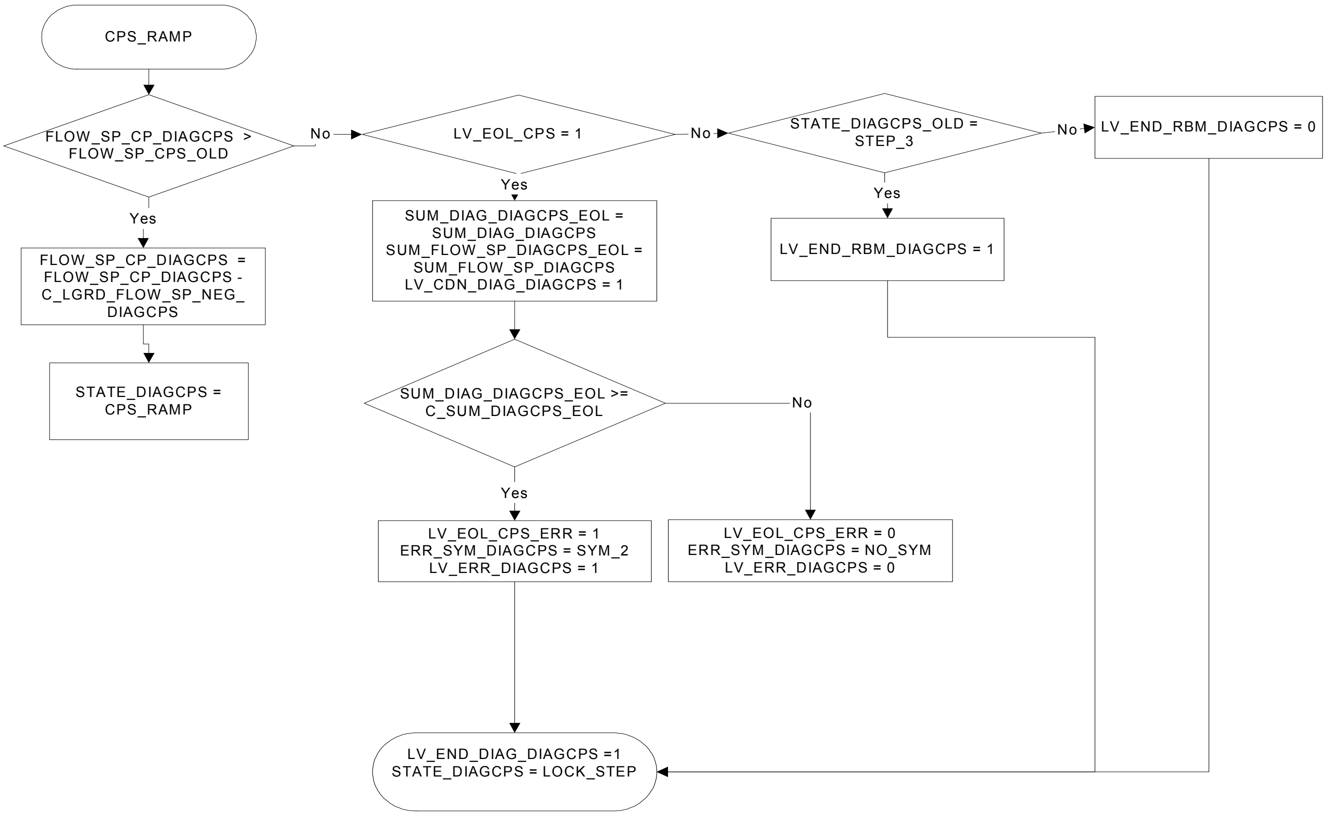


C:

```
IF(1)   MAF_DIAGCPS – MAF_MES < C_MAF_DIAGCPS_2
        THEN(1)   SUM_MAF_DIAG_DIAGCPS_OFS ++
                  IF(2)   SUM_MAF_DIAG_DIAGCPS_OFS >
                          C_SUM_MAF_DIAG_DIAGCPS_OFS
                  THEN(2)   ERR_SYM_DIAGCPS = SYM_3
                            MAF_DIAGCPS_SAE = MAF_MES
                            MAF_DIAGCPS_THD_SAE =
                                  MAF_DIAGCPS-C_MAF_DIAGCPS_2
                            CL_MMV_SAE = FFFFH
                            SUM_DIAG_DIAGCPS_SAE = ~~FFH~~ 0H
                            LV_ERR_DIAGCPS = 1
                            STATE_DIAGCPS = CPS_RAMP
                  ELSE(2)   STATE_DIAGCPS = STEP_3
                  ENDIF(2)
        ELSE(1)   STATE_DIAGCPS = STEP_3
        ENDIF(1)
```

Overview:

CPS-Ramp:



```
IF(1)      FLOW_SP_CP_DIAGCPS > FLOW_SP_CPS_OLD
THEN(1)    FLOW_SP_CP_DIAGCPS = FLOW_SP_CP_DIAGCPS –
                              C_LGRD_FLOW_SP_NEG_DIAGCPS
ELSE(1)    IF(2)   LV_EOL_CPS = 1
           THEN(2)      SUM_DIAG_DIAGCPS_EOL = SUM_DIAG_DIAGCPS
                        SUM_FLOW_SP_DIAGCPS_EOL = SUM_FLOW_SP_DIAGCPS
                        LV_CDN_DIAG_DIAGCPS  = 1
                  IF(3a)  SUM_DIAG_DIAGCPS_EOL >= C_SUM_DIAGCPS_EOL
                  THEN(3a)     LV_EOL_CPS_ERR = 1
                               ERR_SYM_DIAGCPS = SYM_2
                               LV_ERR_DIAGCPS = 1
                  ELSE(3a)     LV_EOL_CPS_ERR = 0
                               ERR_SYM_DIAGCPS = NO_SYM
                               LV_ERR_DIAGCPS = 0
                  ENDIF(3a)
           ELSE(2)      IF(3b)  STATE_DIAGCPS_OLD = STEP_3
                        THEN(3b)     LV_END_RBM_DIAGCPS = 1
                        ELSE(3b)     LV_END_RBM_DIAGCPS = 0
                        ENDIF(3b)
           ENDIF(2)
           LV_END_DIAG_DIAGCPS = 1
           STATE_DIAGCPS = LOCK_STEP
ENDIF(1)
```

Reset EOL-request-bit

LOCK_STEP:

Description:

FLOW_SP_CP_DIAGCPS = FLOW_SP_CPS_OLD

LV_ACT_DIAGCPS = 0

LV_DIAGCPS_MIN_MOD = 0

LV_INH_MAP_CTL_DIAGCPS = 0

LV_SUPP_ACCIN_DIAGCPS = 0

// RBM in case of TEV-check ended with o.k. before all 3 STEPs have been completed

```
IF          LV_T_DIAGCPS_RBM = 0
THEN        T_DIAGCPS_DLY_1_RBM = C_ T_DIAGCPS_DLY
            T_DIAGCPS_DLY_2_RBM = C_ T_DIAGCPS_DLY
            T_DIAGCPS_N_MAP_1_RBM = C_SUM_DIAGCPS_MAX
                                               * C_ T_DIAGCPS_N_MAP_1
            T_DIAGCPS_MAF_RBM = C_T_DIAGCPS_MAF
            LV_T_DIAGCPS_RBM = 1
ENDIF

IF(1)       TCO > C_TCO_MIN_DIAGCPS          and
            LV_IS = 1                        and
            VS = 0                           and
            LV_CH = 0                        and
            LV_END_RBM_DIAGCPS = 0           and
            T_DLY_PURGE_DIAGCPS = 0

THEN(1)     IF(2)            STATE_DIAGCPS_RBM = STEP_1
            THEN(2)IF(3a)         T_DIAGCPS_DLY_1_RBM > 0
                 THEN(3a)         T_DIAGCPS_DLY_1_RBM - -
                 ELSE(3a)IF(4a)        T_DIAGCPS_N_MAP_1_RBM > 0
                          THEN(4a)     T_DIAGCPS_N_MAP_1_RBM - -
                          ELSE(4a)
                               IF (5a)        LV_OPM_AV_DIAGCPS_P_THR = 0
                               THEN(5a)       LV_END_RBM_DIAGCPS = 1
                               ELSE(5a)       IF(6a)  T_DIAGCPS_DLY_2_RBM > 0
                                              THEN(6a)T_DIAGCPS_DLY_2_RBM  - -
                                              ELSE(6a)
                                                 IF(7a) T_DIAGCPS_MAF_RBM > 0
                                                 THEN(7a)T_DIAGCPS_MAF_RBM - -
                                                 ELSE(7a)LV_END_RBM_DIAGCPS = 1
                                                 ENDIF(7a)
                                              ENDIF(6a)
                               ENDIF(5a)
                          ENDIF(4a)
```

```
                ENDIF(3a)
        ELSE(2)IF(3b)          STATE_DIAGCPS_RBM = STEP_2
                THEN(3b)IF(4b)          T_DIAGCPS_DLY_2_RBM > 0
                        THEN(4b)        T_DIAGCPS_DLY_2_RBM  - -
                        ELSE(4b)IF(5b)          T_DIAGCPS_MAF_RBM > 0
                                THEN(5b)        T_DIAGCPS_MAF_RBM - -
                                ELSE(5b)        LV_END_RBM_DIAGCPS = 1
                                ENDIF(5b)
                        ENDIF(4b)
                ENDIF(3b)
        ENDIF (2)
ENDIF(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CL_MMV_DIAGCPS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Moving mean value of canister load for functional check CPS					
C_FLOW_SP_DIAGCPS_MAF	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow setpoint offset in step 3 - functional check CPS					
C_FLOW_SP_DIAGCPS_N	1	0...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Flow setpoint offset in step 2 - functional check CPS					
C_LGRD_FLOW_SP_NEG_DIAGCPS	1	1...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Limitation gradient for closing the CPS- funtional check CPS					
C_LGRD_FLOW_SP_POS_DIAGCPS	1	1...FFFFH	0...7.99987	0.1221e-3	[kg/h]
Limitation gradient for opening the CPS - functional check CPS					
C_MAF_DIAGCPS_2	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Delta MAF for MAF decrease - functional check CPS, gedrosselt					
C_MAF_MAX_DIAGCPS	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
MAF condition - functional check CPS					
C_N_DIAGCPS_2	1	0...1FE0H	0...8160	1	[rpm]
Threshold for engine speed increase in step 2 - functional check CPS, gedrosselt					
C_N_DIF_DIAGCPS	1	0...1FE0H	0...8160	1	[rpm]
Condition for IS engine speed difference - functional check CPS					
C_SUM_FLOW_SP_DIAGCPS_MAX	1	0...FFH	0...255	1	[cyc]
Max. number of step 2 loops which have been passed - functional check CPS					
C_SUM_DIAGCPS_MAX	1	0...FFH	0...255	1	[cyc]
Counter limit of C_SUM_DIAGCPS - funtional check CPS					
C_SUM_N_DIAG_DIAGCPS_OFS	1	0...FFH	0...255	1	[cyc]
Counter for n-events per offset during STEP_2 - functional check CPS					
C_SUM_MAP_DIAG_DIAGCPS_OFS	1	0...FFH	0...255	1	[cyc]
Counter for MAP-events per offset during STEP_2 - functional check CPS					
C_SUM_MAF_DIAG_DIAGCPS_OFS	1	0...FFH	0...255	1	[cyc]
Counter for MAF-events per offset during STEP_2 - functional check CPS					
C_SUM_DIAGCPS_EOL	1	0...FFH	0...255	1	[cyc]
EOL-test: Counter limit of SUM_DIAGCPS_EOL - funtional check CPS					
C_T_DIAGCPS_MAF	1	1...FFH	0.1...25.5	0.1	[s]
Time for active FLOW_SP offset C_FLOW_SP_DIAGCPS_MAF - functional check CPS					

C_T_DIAGCPS_N_MAP_1	1	1...FFH	0.1...25.5	0.1	[s]
Time interval after expiry of which the cycle of step 2 is repeated - functional check CPS					
C_T_DIAGCPS_N_MAP_2	1	1...FFH	0.1...25.5	0.1	[s]
Time for evaluaiton of N, MAP reaction - functional check CPS					
C_T_DIAGCPS_N_MAP_3	1	1...FFH	0.1...25.5	0.1	[s]
Time for increasing FLOW_SP - functional check CPS					
C_T_DIAGCPS_N_MAP_4	1	1...FFH	0.1...25.5	0.1	[s]
Time for evaluation of N, MAP reaction - functional check CPS					
C_T_DIAGCPS_DLY	1	1...FFH	0.1...25.5	0.1	[s]
Delay time - functional check CPS					
C_T_INH_MAP_CTL_DIAGCPS	1	1...FFH	0.1...25.5	0.1	[s]
time threshold for inhibition of MAP-control - functional check CPS					
C_TCO_MIN_DIAGCPS	1	0...FEH	-48...142.5	0.75	[°C]
Min. temperature for functional check CPS					
C_TI_LAM_DIAGCPS_DIF	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Min. TI_LAM_COR_i - difference while CPS-opening during STEP_2 - funcitonal check CPS					
C_MAP_DIAGCPS_1	1	0...FFFFH	0...5434	0.0829175	[hPa]
Threshold for map difference in step 2 - functional check CPS, ungedrosselt					
C_MAP_DIAGCPS_2	1	0...FFFFH	0...5434	0.0829175	[hPa]
Threshold for map difference in step 2 - functional check CPS, gedrosselt					
C_PRS_IM_CTL_I_DIAGCPS	1	0...FFFFH	0...3.99993	0.061e-3	[-]
Threshold for Pressure controller I-part difference in step 2 - functional check CPS					
C_MAP_DIP_SP_MMV_DIAGCPS	1	8000...7FFFH	-1280... 1279.96093	0.0390625	[hPa]
Threshold for map difference in step 2 - functional check CPS, gedrosselt					
LC_DIAGCPS_MAP_ACT_OPM_2	1	0...1H	0...1	1	[-]
MAP monitoring activated during STEP_2 - throttled mode					
C_CRLC_MAF_SP_TQI_DIAGCPS	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for calculation of MAF_SP_TQI_MMV_DIAGCPS					
C_T_CL_MMV_DIAGCPS	1	0...FFH	0...25.5	0.1	[s]
Timer to count how long the CL_MMV has been above the threshold in Step 1 (UGD)					
C_T_CL_MMV_DIAGCPS_2	1	0...FFH	0...25.5	0.1	[s]
Timer to count how long the CL_MMV has been above the threshold in Step 1 (VVTNOTL1)					
C_CL_MMV_DIAGCPS_VLD	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Threshold on CL_MMV to avoid going into Step 2 or Step 3 if the AKF is too full					

29.3.1 Limited dynamic condition LV_MAF_SP_TQI_DYW_DIAGCPS

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: at reset or at transition LV_IGK OFF → ON

LV_MAF_SP_TQI_DYW_DIAGCPS = 0

MAF_SP_TQI_MMV_DIAGCPS = 0

Recurrence: 100 ms

Activation: LV_DC = 1

Deactivation: LV_DC = 0

Formula section:

Calculation of MAF_SP_TQI_MMV_DIAGCPS:

MAF_SP_TQI_MMV_DIAGCPS = MAF_SP_TQI_MMV_DIAGCPS* (1-C_CRLC_MAF_SP_TQI_DIAGCPS) + C_CRLC_MAF_SP_TQI_DIAGCPS * MAF_SP_TQI

```
IF    | MAF_SP_TQI – MAF_SP_TQI_MMV_DIAGCPS |
         < C_MAF_SP_TQI _DYW_DIAGCPS

THEN        LV_MAF_SP_TQI_DYW_DIAGCPS= 1
ELSE        LV_MAF_SP_TQI_DYW_DIAGCPS= 0
            MAF_SP_TQI_MMV_DIAGCPS = MAF_SP_TQI
ENDIF
```

29.4 Inhibition for “Functional check CPS” (Applic. Inc.)

29.4.1 Diagnosis inhibition

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAGCPS	V/O	0...1H	0...1	1	[-]
Activation condition for functional check CPS, irreversible					
LV_DI_DIAGCPS	V/O	0...1H	0...1	1	[-]
Activation condition for functional check CPS, reversible					
LV_INH_DIAG_RBM_DIAGCPS	V	0...1H	0...1	1	[-]
Flag to inhibit catalyst diagnosis function if there is a OBD failure					
OPM_AV_DIAGCPS	V/O	0...2H	0...2	1	[-]
Operation mode DIAGCPS					

Input data:

LV_DC	TCO	STATE_CP	LV_ERR_CPS
LV_ERR_TPS	LV_ERR_ISC	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_FSD[NC_CBK_EX_NR]
LV_EOL_CPS	LV_ERR_VS	LV_ERR_MAF	LV_ERR_TCO
LV_ERR_MAP_DIP_SENS	C_TCO_MIN_DIAGCPS	LV_ERR_MAP_DIP_SHIFT	LV_ERR_DMTLS
LV_ERR_MAP_DIP_PLAUS	LV_ERR_DMTL_PUMP	LV_IGK	LV_ERR_DMTLH
LV_CL_MMV	LV_ERR_PVS_DOUBLE	LV_ERR_CRIT_OVL_ECU_VVL	LV_ERR_DMTLM
LV_ERR_AMP	LV_MTC_CUR_OFF	LV_ERR_DR_SC_VVL	LV_CP_ENA
LV_ERR_ROUGH_LEAK	LV_ERR_SMALL_LEAK	OPM_AV	LV_ERR_OVL_ECU_VVL
LV_ERR_DR_SC_VVL	LV_ERR_OVL_ECU_VVL	LV_ERR_RLY_VVL	LV_ERR_VVL_ROT
LV_ERR_CUR_H_VVL	LV_MIS_STATE_A	LV_MIS_STATE_B	LV_VAR_BN
LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV
LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_MAP_TPS_PLAUS	

FUNCTION DESCRIPTION:

General information:

The functional check is used for the functional test of the CP solenoid which means that the CPS is diagnosed via its effects on the engine.

There are many errors which make it necessary to stop the functional check CPS when they occure. Therefore a diagnosis has to be made whether the functional check can be enabled or not.

If LV_INH_DIAGCPS = 0 and LV_DI_DIAGCPS = 0 then the functional check is enabled.

If the bit LV_INH_DIAGCPS = 1 is set through one of the errors listed below, then the functional check is exited irreversibly until the next engine start.

If the bit LV_INH_DIAGCPS=0 and LV_DI_DIAGCPS=1 then the functional check is exited reversibly until LV_DI_DIAGCPS = 0 (recurrence 100 ms).

If one of the OBD errors ocurs as mentioned in the formula section the RBM of the functional check CPS diagnosis is inhibited by LV_INH_DIAG_RBM_DIAGCPS = 1.

Application conditions:

Initialisation: at every LV_IGK = 0→1 and reset all variables are initialized with 0

Recurrence: 100 ms

Activation: LV_DC = 1

Deactivation: LV_DC = 0

Formula section:

Inhibition due to OBDI error

If LV_ERR_MAF = 1 **or**
LV_ERR_TPS = 1 **or**
LV_ERR_LOAD_TPS_PLAUS = 1 **or**
LV_ERR_CPS = 1 **or**
LV_ERR_ROUGH_LEAK = 1 **or**
LV_ERR_SMALL_LEAK = 1 **or**
LV_ERR_DMTLH = 1 **or**
LV_ERR_DMTLM = 1 **or**
LV_ERR_DMTL_PUMP = 1 **or**
LV_ERR_DMTLS = 1 **or**
LV_ERR_AMP = 1 **or**
LV_ERR_VS = 1 **or**
LV_ERR_TCO = 1 **or**
LV_ERR_ISC = 1 **or**
LV_ERR_FSD[i] = 1 **or**
LV_ERR_MAP_DIP_SENS = 1 **or**
LV_ERR_MAP_DIP_PLAUS = 1 **or**
LV_ERR_MAP_DIP_SHIFT = 1 **or**
LV_ERR_PVS_DOUBLE = 1 **or**
LV_ERR_CRIT_OVL_ECU_VVL = 1 **or**
LV_ERR_DR_SC_VVL = 1 **or**
LV_ERR_OVL_ECU_VVL = 1 **or**
LV_ERR_RLY_VVL = 1 **or**
LV_ERR_VVL_ROT = 1 **or**
LV_ERR_CUR_H_VVL = 1 **or**
LV_MIS_STATE_A = 1 **or**
LV_MIS_STATE_B = 1 **or**
LV_ERR_FUP = 1 **or**

```
            LV_ERR_FUP_MFP_PLAUS = 1          or
            LV_ERR_H_PRS_SYS = 1              or
            LV_ERR_VCV = 1                    or
            LV_ERR_FUP_ORNG = 1               or
            LV_ERR_FUP_ST = 1                 or
            LV_ERR_MAP_TPS_PLAUS = 1



Then        LV_INH_DIAG_RBM_DIAGCPS = 1
Else        LV_INH_DIAG_RBM_DIAGCPS = 0
Endif
```

Inhibition due to environmental condition / calibration:

```
If          LV_INH_DIAG_RBM_DIAGCPS = 1       or
            LV_MTC_CUR_OFF = 1
Then        LV_INH_DIAGCPS = 1
Else        LV_INH_DIAGCPS = 0
Endif
OPM_AV_DIAGCPS = ID_OPM_AV_DIAGCPS
If          LV_INH_DIAGCPS = 0                and
            (OPM_AV_DIAGCPS =1H or OPM_AV_DIAGCPS = 2H)       and
                [ (TCO > C_TCO_MIN_DIAGCPS_1      and
                STATE_CP > 1H                     and
                LV_CP_ENA = 1                     and
                LV_CL_MMV = 1)                    or
                LV_EOL_CPS = 1 ]
Then        LV_DI_DIAGCPS = 0
Else        LV_DI_DIAGCPS = 1
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_MIN_DIAGCPS_1	1	0...FEH	-48...142.5	0.75	[°C]
Min. temperature for FC CPS/ Step1 (CL diagnosis)					
ID_OPM_AV_DIAGCPS	8	0...2H	0...2	1	[-]
LDPM_OPM_AV	8	0...8H	0...8	1	[-]
Operation mode for which DIAGCPS is enabled (0H = never, 1H = UGD, 2H = GD)					

29.4.2 Interface for Rate – Based - Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_DIAGCPS	V/O	0...FFH	0...255	1	[-]
Interface of DIAGCPS monitor with the Rate-Based Monitoring statistics					

Input data:

LV_DC	CTR_ERR_DYN_NR	LV_END_RBM_DIAGCPS	

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the DIAGCPS monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_DIAGCPS data.

Within STATE_RBM_DIAGCPS, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for DIAGCPS diagnosis)

Application conditions:

Initialisation : at LV_DC 0 → 1 transition :
bit 0, bit 1 and bit 2 of STATE_RBM_DIAGCPS = 0
on failure memory reset :
bit 1 of STATE_RBM_DIAGCPS = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_TPS_ST_CHK_2	LV_ERR_TPS_RATIO	LV_ERR_AMP	LV_ERR_EL_CPS
LV_ERR_TPS_1	LV_ERR_TPS_2	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_TPS_AD
LV_ERR_TPS_MAF_2	LV_ERR_VS	LV_ERR_MAF	LV_ERR_DMTLM
LV_ERR_ISC	LV_ERR_DMTL_PUMP	LV_ERR_DMTLH	LV_ERR_DMTLS
LV_ERR_TPS_MAF_1	LV_ERR_TCO_EL	LV_ERR_TCO_STUCK	LV_ERR_TCO_GRD
LV_ERR_TCO_PLAUS	LV_ERR_ROUGH_LEAK	LV_ERR_SMALL_LEAK	LV_ERR_TCO_STUCK_RNG
LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3	LV_ERR_MTC_DR	LV_ERR_PVS_DOUBLE
LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLA_US	LV_ERR_MAP_DIP_SHIFT	LV_ERR_FSD[NC_CBK_EX_NR]
LV_ERR_CRIT_OVL_ECU_VVL	LV_ERR_DR_SC_VVL	LV_ERR_OVL_ECU_VVL	LV_ERR_RLY_VVL
LV_ERR_VVL_ROT	LV_ERR_CUR_H_VVL	LV_ERR_MIS_0	LV_ERR_MIS_1
LV_ERR_MIS_2	LV_ERR_MIS_3	LV_ERR_MIS_4	LV_ERR_MIS_5
LV_ERR_FUP[i]	LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_VCV[i]
LV_ERR_TPS_AD_BOL	LV_ERR_FUP_ORNG	LV_ERR_FUP_ST	LV_ERR_MAP_TPS_PLAUS

```
If(1)      { CPU optimization at LV_DC 0 → 1 transition }
           CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
           While  bit 1 of STATE_RBM_DIAGCPS = 0 do
                  with each XX failure of the above list :
                  ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                  SYNCHRONIZATION<CALL>)
                  If(2)        XX has a pending status
                  Then(2)      bit 1 of STATE_RBM_DIAGCPS = 1
                  Endif(2)
           Endwhile
Else(1)    { the dynamic failure memory is empty }
           No action
Endif(1)
```

Every 1 s :

```
If     bit 0 of STATE_RBM_DIAGCPS = 0
Then
           If      LV_END_RBM_DIAGCPS = 1          and
           Then    bit 0 of STATE_RBM_DIAGCPS = 1
           Endif
Endif
```

```
If     bit 1 of STATE_RBM_DIAGCPS = 0
```

```
Then
        If      LV_INH_DIAG_RBM_DIAGCPS = 1
        Then    bit 1 of STATE_RBM_DIAGCPS = 1
        Endif
Endif
bit 2 of STATE_RBM_DIAGCPS = 1
```

29.5 Tank leakage diagnosis based on DMTL

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DMTLS	V/O	0...1H	0...1	1	[-]
controlbit of the DMTL solenoid powerstage					
LV_DMTL_PUMP	V/O	0...1H	0...1	1	[-]
controlbit of the DMTL pump powerstage					
CUR_DMTL_COMP_ROUGH_LEAK	V/O	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Pump current comparision in case of rough leak measurement					
CUR_DMTL_COMP_SMALL_LEAK_LEN	V/O	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Pump current comparision in case of 2. miniature leak measurement					
CUR_DMTL_DMTLS_TEST	V/O	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Pump current during pump test					
CUR_DMTL_DIF	V/O	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Differential pump current					
CUR_DMTL_DIF_REF_ROUGH_LEAK	V/O	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Difference, pump current between reference and min. in case of rough leak check					
CUR_DMTL_REF_LEAK	V/O	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Pump current reference leakage					
CUR_DMTL_REF_LEAK_EOL	V/O/S	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Pump current reference leakage for tester tool					
CUR_DMTL_ROUGH_LEAK_END	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Pump current at the end of the rough leak measurement					
CUR_DMTL_ROUGH_LEAK_LEN_END	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Pump current at the end of the rough leak measurement extension					
CUR_DMTL_ROUGH_LEAK_MIN	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Min. pump current in case of rough leak measurement					
CUR_DMTL_ROUGH_LEAK_MIN_EOL	O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Min. pump current in case of rough leak measurement for tester tool					
CUR_DMTL_SMALL_LEAK_END	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Pump current at the end of the miniature leak measurement					
CUR_DMTL_THD_DMTLS_TEST	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Pump current threshold in case of DMTL valve check					
CUR_DMTL_THD_ROUGH_LEAK	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Pump current threshold in case of rough leak measurement					
CUR_DMTL_THD_ROUGH_LEAK_LEN	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Pump current threshold in case of rough leak measurement extension					
T_ACT_CUR_DMTL_STAT	V/O	0...FFFFH	0...6553.5	0.1	[s]
Current time, pump current stable					
T_ACT_LEAK_MES	V/O	0...FFFFH	0...6553.5	0.1	[s]
Current time, leakage measurement					
T_ACT_REF_LEAK_MES	V/O	0...FFFFH	0...6553.5	0.1	[s]
Current time, reference leakage measurement					
LV_CHK_DMTL_CUR_DMTL_VLD	V/O	0...1H	0...1	1	[-]
Check of components on pump current thresholds valid					
LV_CUR_DMTL_DMTLS_TEST	V/O	0...1H	0...1	1	[-]
Pump current threshold reached in case of DMTL-valve check					
LV_CUR_DMTL_REF_MES_MIN	V/O	0...1H	0...1	1	[-]
Lower pump current threshold reached during reference measurement					
LV_CUR_DMTL_REF_MES_MAX	V/O	0...1H	0...1	1	[-]
Upper pump current threshold reached during reference measurement					
LV_CYC_ROUGH_LEAK_SUSP	V/O	0...1H	0...1	1	[-]
Cycle flag, rough leak suspected					
LV_CYC_SMALL_LEAK_MODE6	V/O	0...1H	0...1	1	[-]
Cycle flag, miniature leak					
LV_DMTL_ACT	V/O	0...1H	0...1	1	[-]
DMTL state machine running					
LV_DMTLS_ON	V/O	0...1H	0...1	1	[-]

DMTL valve on					
LV_DMTL_PUMP_ON	V/O	0...1H	0...1	1	[-]
DMTL pump motor on					
LV_ERR_DET_SMALL_LEAK_MIN	V/O	0...1H	0...1	1	[-]
Error, minimum value detected, miniature leakage					
LV_REF_LEAK_ROUGH_LEAK_SUSP	V/O	0...1H	0...1	1	[-]
Condition, second reference measurement due to "rough leak suspected"					
LV_ROUGH_LEAK_SUSP	V/O	0...1H	0...1	1	[-]
Rough leak suspected					
LV_ROUGH_LEAK_MES_LEN	V/O	0...1H	0...1	1	[-]
Condition, extension rough leak measurement					
LV_SMALL_LEAK_MES_LEN	V/O	0...1H	0...1	1	[-]
Condition, extension miniature leak measurement					
STATE_DMTL	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH BH CH	START REF_LEAK_MES ROUGH_LEAK_MES ROUGH_LEAK_MES_LEN ROUGH_LEAK_MES_END SMALL_LEAK_MES SMALL_LEAK_MES_LEN REF_LEAK_MES_2 TANK_PROOFED SMALL_LEAK ROUGH_LEAK MODULE_ERROR END	1	[-]
DMTL state					

STATE_DMTL_EOL	O/V/S	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH BH CH 11H 12H 21H 22H 23H 24H 25H 26H	START REF_LEAK_MES ROUGH_LEAK_MES ROUGH_LEAK_MES_LEN ROUGH_LEAK_MES_END SMALL_LEAK_MES SMALL_LEAK_MES_LEN REF_LEAK_MES_2 TANK_PROOFED SMALL_LEAK ROUGH_LEAK MODULE_ERROR END VB_OUT_OF_RANGE ELECTRICAL_ERR REFUELLING TANK_CAP_OPEN VB_GITTER T_DMTL_MAX CUR_REF_DIF_MAX THD_CUR_DMTL_DIF_MES	1	[-]
Status of DMTL EOL diagnosis					
CUR_DMTL	V/O	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Pump current tank leakage diagnosis					
CUR_DMTL_COR	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Corrected pump current					
CUR_DMTL_COR_FIL	V/O	0...FFFFFFFFH	0...399.99999	0.0931e-6	[mA]
Corrected and filtered pump current					
CUR_DMTL_COR_FIL_EOL	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Corrected and filtered pump current for tester tool					
LV_VB_DIF_MAX	V/O/S	0...1H	0...1	1	[-]
High battery voltage deviation					
VB_DMTL_REF	V/O	0...FFFFH	0...25.99960	3.97E-04	[V]
Filtered battery voltage on transition to refeence leak measurement					
VB_DMTL_FIL	V/O	0...FFFFH	0...25.99960	3.97E-04	[V]
Filtered battery voltage					
LV_CUR_DMTL_REF_DIF_MAX	V/O/S	0...1H	0...1	1	[-]
Condition "delta reference current" exceeded					
CTR_CUR_DMTL_REF_DIF_MAX	V/O/S	0...FFFFH	0...65535	1	[-]
Number of aborted measurements due to current fluctuation					
CUR_DMTL_COR_FIL_MIN	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Minimum pump current during reference leak measurement					
CUR_DMTL_COR_FIL_MAX	V/O	0...FFFFH	0...399.99389	6.10E-03	[mA]
Maximum pump current during reference leka measurement					
CUR_DMTL_REF_DIF_MAX	V	0...FFFFH	0...399.99389	6.10E-03	[mA]

Delta pump current during reference measurement					
LV_T_DLY_DMTL_PWL	V/O	0...1H	0...1	1	[-]
Waiting time DMTL start finished					
LV_REQ_PWL_DMTL	V/O	0...1H	0...1	1	[-]
Request of STG off delay, tank leakage diagnosis					
LV_ENA_ROUGH_LEAK_MES	V/O	0...1H	0...1	1	[-]
Enabling rough leak measurement					
LV_NO_PURGE_DMTL	V/O	0...1H	0...1	1	[-]
Request for no_purge of the evap.-system					
LV_ENA_MAIN_DMTL	V/O	0...1H	0...1	1	[-]
DMTL main enabling					
LV_ENA_LEAK_DMTL	V/O	0...1H	0...1	1	[-]
Enabling function request, coarse / miniature leakage measurement					
LV_ENA_MAIN_DMTL_FUC	V/O	0...1H	0...1	1	[-]
Main enabling DMTL Check Filler Cap					
LV_FTL_DMTL_VAL	V/O	0...1H	0...1	1	[-]
Validated tank level for DMTL function					
LV_FTL_DMTL_MAX	V/O	0...1H	0...1	1	[-]
Max. tank level for DMTL function					
LV_FTL_DMTL_MIN	V/O	0...1H	0...1	1	[-]
Min. tank level for DMTL function					
LV_FTL_DIAG	V/O	0...1H	0...1	1	[-]
Level sensor diagnosis					
FTL_INI	O/V/S	0...7FH	0...127	1	[l]
Ini value level for diagnostic cycle					
FCO_AV_DMTL	V/O/S	0...FFFFH	0...4294.9	6.55E-02	[l]
Currently consumed fuel					
LV_RD_FTL	V/O	0...1H	0...1	1	[-]
Level diagnosis active					
LV_FCO_H_FTL	V/O	0...1H	0...1	1	[-]
Consumed fuel greater than tank level					
LV_CLR_FTL	-	0...1H	0...1	1	[-]
set variable to reinitialize the FTL_Plaus diagnosis after clr FMY or consumption higher than tank volume					
LV_REFU_RLS_DIAG	V/O	0...1H	0...1	1	[-]
Refueling diagnosis possible					
LV_REFU_END	V/O	0...1H	0...1	1	[-]
Refueling finished					
LV_FTL_INC_IGK_ON	V/O	0...1H	0...1	1	[-]
Tank filling volume has increased in case of ignition key ON					
LV_FTL_DEC_IGK_ON	V/O	0...1H	0...1	1	[-]
Tank filling volume has decreased in case of ignition key ON					
LV_FTL_DYN	V/O	0...1H	0...1	1	[-]
Level dynamics due to refueling					
LV_FTL_INC	V/O	0...1H	0...1	1	[-]
Tank volume has increased					
LV_FTL_DEC	V/O	0...1H	0...1	1	[-]
Tank volume has decreased					
FTL_OLD	V/O/S	0...7EH	0...126	1	[l]
FTL for plausibility check					
LV_REFU_1	V/O	0...1H	0...1	1	[-]
Refueling process, internal					
LV_REFU_VAL_1	V/O	0...1H	0...1	1	[-]
Refueling bit valid, internal					
LV_REFU	V/O	0...1H	0...1	1	[-]
Refueling					
LV_REFU_VAL	V/O	0...1H	0...1	1	[-]
Refueling bit valid					
LV_ENA_FCO_DIAG	V/O	0...1H	0...1	1	[-]
Consumption diagnosis enable					
LV_FCO_DIAG_MIN	V/O	0...1H	0...1	1	[-]
Lower consumption diagnosis threshold					

LV_FCO_DIAG_MAX	V/O	0...1H	0...1	1	[-]
Upper consumption diagnosis threshold					
LV_ENA_FTL_DIAG	V/O	0...1H	0...1	1	[-]
Fuel level diagnosis enable					
LV_ERR_FTL	V/O/S	0...1H	0...1	1	[-]
Tank level error					
LV_ERR_FTL_PLAUS	V/O/S	0...1H	0...1	1	[-]
Tank level plausibility error					
LV_NO_PURGE_DMTL_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
request bit for no_purge					
LV_ENA_ROUGH_LEAK_MES_ONLY	V/O	0...1H	0...1	1	[-]
Enable only rough leak measurement					
LV_REQ_PWL_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
Request of STG off delay, rough leak diagnosis					
LV_ENA_ROUGH_LEAK_3	V/O	0...1H	0...1	1	[-]
Auxiliary bit interruption of rough leak check					
LV_REQ_ENA_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
Request rough leak measurement					
LV_ENA_ROUGH_LEAK_1	V/O	0...1H	0...1	1	[-]
Auxiliary bit enable condition for rough leak check is present					
LV_ENA_ROUGH_LEAK_2	V/O	0...1H	0...1	1	[-]
Auxiliary bit enabling rough leak check through manual activation					
LV_REQ_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
Request rough leak measurement (low volatile fuel)					
LV_TCO_ST_DMTL	V/O	0...1H	0...1	1	[-]
Coldstart condition for DMTL fulfilled					
LV_ENA_FUC_ROUGH_LEAK_MES	V/O	0...1H	0...1	1	[-]
Enable rough leak measurement due to Check Filler Cap					
CTR_CNL_ROUGH_LEAK_MES_FUC	V/O/S	0...FFH	0...255	1	[-]
Number of aborted check filler cap checks					
CTR_REP_ROUGH_LEAK_MES_FUC	V/O/S	0...FFH	0...255	1	[-]
Number of repeated check filler cap checks					
LV_ENA_FUC	V/O/S	0...1H	0...1	1	[-]
Request check filler cap					
LV_ENA_CHK_FUC_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
Enable rough leak measurement due to Check Filler Cap					
CTR_ENA_DMTL_CHK_FUC	V	0...FFH	0...255	1	[-]
Number of enabled check filler cap checks					
LV_NO_PURGE_DMTL_SMALL_LEAK	V/O	0...1H	0...1	1	[-]
Request for no_purge					
LV_ENA_SMALL_LEAK_MES	V/O/S	0...1H	0...1	1	[-]
Enable miniature leak measurement					
LV_REQ_PWL_SMALL_LEAK	V/O	0...1H	0...1	1	[-]
Request of STG off delay, miniature leak test					
LV_ENA_SMALL_LEAK_MES_END	V/O	0...1H	0...1	1	[-]
End of enable miniature leak measurement					
LV_ENA_SMALL_LEAK_1	V/O	0...1H	0...1	1	[-]
Enable for miniature leak check via test unit intervention					
LV_VB_RANGE_DMTL	V/O	0...1H	0...1	1	[-]
DMTL does not start because battery is out of band width					
LV_DMTL_RST_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
Inhibit small leak diagnosis					
CTR_TCO_ST_DMTL	V/S	0...FFH	0...255	1	[-]
Counter of recognized cold starts for DMTL small leak test					
LV_CHK_DMTL_VLD	V/O	0...1H	0...1	1	[-]
Check of components on current fluctuations valid					
LV_CTR_CNL_SMALL_LEAK_MES	V/O/S	0...1H	0...1	1	[-]
Threshold of aborted miniature leak measurements due to current fluctuations exceeded					
CTR_CNL_SMALL_LEAK_MES	V/O/S	0...FFH	0...255	1	[-]
Number of aborted miniature leak measurements due to current fluctuations					
LV_DR_DMTL	V/O	0...1H	0...1	1	[-]

Enabling the output stages of the DMTL module					
LV_DR_DMTL_PUMP	V/O	0...1H	0...1	1	[-]
DMTL pump activation					
LV_DR_DMTLS	V/O	0...1H	0...1	1	[-]
DMTL solenoid valve activation					
LV_NO_PURGE_DR_DMTL	V/O	0...1H	0...1	1	[-]
Pulse width modulation of canister-purge					
LV_DMTL_STOP	V/O/S	0...1H	0...1	1	[-]
Abnormal termination of DMTL function					
LV_DET_FUC_OPEN	V/O/S	0...1H	0...1	1	[-]
Detection of fuel cap open					
LV_DET_REFU	V/O/S	0...1H	0...1	1	[-]
Detection of refueling					
CUR_DMTL_COR_FIL_REFU_HPF	V/O	FFFF8000... 7FFFH	-200... 199.98779	6.10E-03	[mA]
Corrected, high-pass filtered pump current					
LV_CUR_DMTL_THD_DIF_MES	V/O/S	0...1H	0...1	1	[-]
Decreasing pump current during measurement					
LV_T_DMTL_MAX	V/O/S	0...1H	0...1	1	[-]
Max. time for DMTL					
LV_ERR_DET_ROUGH_LEAK_MIN	V/O/S	0...1H	0...1	1	[-]
Rough leak error					
LV_CYC_ROUGH_LEAK_MODE6	V/O	0...1H	0...1	1	[-]
Cycle flag, rough leak error					
LV_DIST_DMTL	V/O	0...1H	0...1	1	[-]
Minimum distance after suspected rough leak or refueling					
LV_REFU_DMTL	V/O/S	0...1H	0...1	1	[-]
Condition refueling					
LV_DIST_DET_ROUGH_LEAK	V/O/S	0...1H	0...1	1	[-]
Distance covered since rough leak detection					
LV_ROUGH_LEAK_SUSP_SET	V/O/S	0...1H	0...1	1	[-]
Rough leak suspicion with cycle bit rough leak suspicion present					
DIST_DMTL	V/O/S	0...FFFFFFFFH	0...65535	0.0153e-3	[m]
Distance covered for DMTL function					
LV_FUC_CAN	V/O	0...1H	0...1	1	[-]
Tank cap open – display					
LV_FUC_OPEN	V/O/S	0...1H	0...1	1	[-]
Filler cap open					
LV_ERR_DMTLM	V/O/S	0...1H	0...1	1	[-]
Error DMTL module					
LV_ERR_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
Error rough leak					
LV_ERR_SMALL_LEAK	V/O	0...1H	0...1	1	[-]
Error miniature leak					
LV_ERR_DET_DMTL_MAX	V/O/S	0...1H	0...1	1	[-]
Error maximum value DMTL					
LV_ERR_DET_DMTL_MIN	V/O/S	0...1H	0...1	1	[-]
Error minimum value DMTL					
LV_SDR_DMTL	V/O/S	0...1H	0...1	1	[-]
Error plausibility DMTL					
LV_ERR_SIG_DMTL	V/O/S	0...1H	0...1	1	[-]
Error signal DMTL					
LV_CDN_DIAG_DMTLM	V/O	0...1H	0...1	1	[-]
Diagnosis condition for DMTL module diagnosis					
LV_CDN_DIAG_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
Diagnosis condition for rough leak diagnosis					
LV_CDN_DIAG_SMALL_LEAK	V/O	0...1H	0...1	1	[-]
Diagnosis condition for small leak diagnosis					

ERR_SYM_DMTLM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected failure of each symptom for module failure					
ERR_SYM_ROUGH_LEAK	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected failure of each symptom for rough leak failure					
ERR_SYM_SMALL_LEAK	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected failure of each symptom for small leak failure					
LV_END_DIAG_DMTLM	V/O/S	0...1H	0...1	1	[-]
End of module diagnosis					
LV_END_DIAG_ROUGH_LEAK	V/O	0...1H	0...1	1	[-]
End of rough leak diagnosis					
LV_END_DIAG_SMALL_LEAK	V/O	0...1H	0...1	1	[-]
End of small leak diagnosis					
LV_DMTL_EOL	V/O	0...1H	0...1	1	[-]
End of line test active					
LV_DMTL_ASA	V/O	0...1H	0...1	1	[-]
Factory test active					
LV_HDMTL_ON	V/O	0...1H	0...1	1	[-]
Heater DMTL					
M6_CUR_DMTL_COR_FIL_CID18	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - module diagnosis minimum failure					
M6_CUR_DMTL_COR_FIL_CID19	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - module diagnosis maximum failure					
M6_CUR_DMTL_DMTLS_TEST	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - module diagnosis plausibility failure					
M6_CUR_DMTL_THD_DMTLS_TEST	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - module diagnosis plausibility diagnosis thd					
M6_CUR_DMTL_ROUGH_LEAK_END	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - DMTL, Rough-leak short cycle					
M6_CUR_DMTL_THD_ROUGH_LEAK	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - DMTL, Rough-leak short cycle thd					
M6_CUR_DMTL_SMALL_LEAK_END	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - DMTL, Small leak					
M6_CUR_DMTL_REF_LEAK	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - DMTL, Small leak thd					
M6_CTR_CNL_SMALL_LEAK_MES	V/O/S	0...FFH	0...255	1	[-]
mode06 - Module-diagnosis, Signal failure					
M6_CUR_DMTL_ROUGH_LEAK_LEN_END	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]
Mode06 - DMTL, Rough leak long cycle					
M6_CUR_DMTL_THD_ROUGH_LEAK_LEN	V/O/S	0...FFFFH	0...399.99389	6.10E-03	[mA]

Mode06 - DMTL, Rough leak long cycle thd					
FTL_ST	-	0...7FH	0...127	1	[l]
Fuel tank level at start					
DIF_FTL_AST	-	80...7FH	-128...127	1	[l]
Fuel tank level difference after start					
DIF_FCO_FTL	-	80...7FH	-128...127	1	[l]
Fuel difference to determinate upper and lower threshold of the consumption diagnosis					
FTL_AV	-	0...7FH	0...127	1	[l]
Fuel tank level actual value					
FTL_VST_IN	-	0...7FH	0...127	1	[l]
Fuel tank level at vehicle stopped and under possible refuelling conditions at IGK on and FTL reading					
DIF_FTL_VST	-	80...7FH	-128...127	1	[l]
Fuel tank level difference at vehicle stopped and under possible refuelling conditions at IGK on and FTL reading					
T_DMTL_EOL	V	0...FFFFH	0...6553.5	0.1	[s]
Time used for end of line diagnosis after test unit initiation					

Input data:

LV_CONF_DMTL	FTL	V_DMTL	VB
ECU_STATE	LV_ES	TAM	LV_ERR_TAM
AMP_MES	LV_ERR_AMP	T_AST	VS
LV_ERR_VS	CL_MMV_NORM_PURGE_END	LV_ERR_EL_CPS	STATE_EFP_CRASH_CAN
LV_ERR_DIAGCPS	LV_ERR_DMTLS	LV_ERR_DMTL_PUMP	LV_ERR_TCO
TCO_ST	LV_ERR_SPI_MPS	LV_IGK	LV_ST_END
FCO_DMTL	TRT	LV_FTL_CAN_ERR	LV_ERR_BN_KM_ICL
LV_ERR_TOUT_ICL_2	LV_T_ES_NOT_PLAUS	LV_T_REL_CAN_REG	TAM_ST
LV_CLOSE_ACT_CP	LV_PL	LV_PU	LV_PUC
LV_ACT_DMTLS_EXT_ADJ	LV_DMTL_PUMP_EXT_ADJ	LV_DMTLS_EXT_ADJ	LV_ACT_DMTL_PUMP_EXT_ADJ
LV_ENA_CHK_FUC_ROUGH_LEAK_MES	LV_IS	LV_DMTLH_EXT_ADJ	LV_ACT_DMTLH_EXT_ADJ

Calibration data:

Name	**Dim**	**Hex. limits**	**Phys. Limit**	**Resol.**	**Unit**
C_T_CUR_DMTL_ROUGH_LEAK_MIN	1	0...FFFFH	0...6553.5	0.1	[s]
Min. time pump current check in rough leak phase					
C_T_CUR_DMTL_ROUGH_LEAK_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Max. time pump current check in rough leak phase					
C_T_REF_TEST_ROUGH_LEAK	1	0...FFFFH	0...6553.5	0.1	[s]
Time for reference test in case of rough leak check					
C_T_CUR_DMTL_STAT	1	0...FFFFH	0...6553.5	0.1	[s]
Time for pump current stable					
C_CUR_DMTL_DIF_CUR_DMTL_STAT	1	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Delta pump current for pump current					
C_T_REF_LEAK_MES_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Time for reference leakage measurement					
C_CUR_DMTL_REF_LEAK_MIN	1	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Lower threshold of pump current for reference measurement					
C_CUR_DMTL_REF_LEAK_MAX	1	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Upper threshold of pump current for reference measurement					
C_CUR_DMTL_REF_OFS	1	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Reference pump current offset					
C_CUR_DMTL_COR_FIL_OFS_AMP	1	FFFF8000...7FFFH	-200...199.98779	6.1036e-3	[mA]

Pump current offset in case of ambient pressure					
C_T_REF_LEAK_MES_REP_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Time for repeat reference leakage measurement					
C_T_ACT_ROUGH_LEAK	1	0...FFFFH	0...6553.5	0.1	[s]
Time for rough leak measurement					
C_T_ACT_SMALL_LEAK	1	0...FFFFH	0...6553.5	0.1	[s]
Time for miniature leak measurement					
IP_T_ACT_ROUGH_LEAK	6	0...FFFFH	0...6553.5	0.1	[s]
LDP_FTL__T_ROUGH_LEAK_TEST	6	0...7EH	0...126	1	[l]
Time for rough leak measurement					
IP_T_ACT_SMALL_LEAK__FTL	6	0...FFFFH	0...6553.5	0.1	[s]
LDP_FTL__T_SMALL_LEAK_TEST	6	0...7EH	0...126	1	[l]
Time for miniature leak measurement					
IP_FAC_CUR_DMTL_ROUGH	0	0...FFFFH	0...0.99998	0.0153e-3	[-]
Factor delta pump current for rough leak detection					
IP_FAC_CUR_DMTL_REF	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_CUR_DMTLDIFREFROUGHLEAK_1	8	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Factor delta pump current for rough leak detection concerning reference					
IP_T_CUR_DMTL_STAT	8*8	0...FFFFH	0...6553.5	0.1	[s]
LDP_FTL__T_CUR_DMTL_STAT	8	0...7EH	0...126	1	[l]
LDPM_CUR_DMTLDIFREFROUGHLEAK_1	8	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Time for pump current stable					
C_FAC_T_CUR_DMTL_STAT_LEN	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Factor for 2. stabilization					
IP_FAC_CUR_DMTL_CHK_FUC_ROUGH	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_CUR_DMTLDIFREFROUGHLEAK_1	8	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Factor delta pump current for rouh leak detection concerning reference in case of check filler cap test					
C_T_ACT_ROUGH_LEAK_CHK_FUC	1	0...FFFFH	0...6553.5	0.1	[s]
Time for rough leak measurement in case of check filler cap					
IP_T_ACT_ROUGH_LEAK_CHK_FUC	6	0...FFFFH	0...6553.5	0.1	[s]
LDP_FTL__T_ROUGH_LEAK_TEST	6	0...7EH	0...126	1	[l]
Time for rough leak measurement in case of check filler cap and error FTL					
C_R_DMTL	1	0...FFH	0...50	0.1953125	[Ohm]
Resistance of pump current measurement					
C_T_DLY_DMTLS	1	0...FFH	0...25.5	0.1	[s]
Waiting time after change-over of solenoid valve					
C_CRLC_CUR_DMTL_FIL_LIM	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Time constant at the beginning of pump current filtering					
C_CRLC_CUR_DMTL_FIL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Time constant of pump current filtering					
IP_FAC_CUR_DMTL	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_VB_DMTL_FIL__FAC_CUR_DMTL	6	0...FFFFH	0...25.99960	0.3967e-3	[V]
Battery-voltage dependent correction factor for the pump current					
C_CRLC_VB_DMTL_FIL_LIM	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Time constant at the beginning of VB filtering					
C_CRLC_VB_DMTL_FIL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Time constant of VB filtering					
C_VB_DMTL_DIF	1	0...FFFFH	0...25.99960	0.3967e-3	[V]
Delta VB threshold for DMTL					
C_VB_DMTL_DIF_CHK_FUC	1	0...FFFFH	0...25.99960	0.3967e-3	[V]
Delta VB threshold for DMTL in case of check filler cap					
C_CUR_DMTL_REF_DIF_MAX	1	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Upper threshold of the Delta pump current					
C_T_DLY_REQ_PWL_DMTL	1	0...FFH	0...25.5	0.1	[s]
Post-operating time after no request for extending the post-operating phase is present any longer					
C_T_DLY_ES_PWL	1	0...FFH	0...25.5	0.1	[s]
Post-operating wait period until start of DMTL function					
C_AMP_DMTL_MAX	1	0...FFFFH	0...5434	0.0829175	[hPa]
Lower altitude threshold for DMTL function / upper ambient pressure for DMTL function					
C_TAM_DMTL_MIN	1	0...FEH	-48...142.5	0.75	[°C]

Lower temperature threshold for the DMTL function					
C_TAM_DMTL_MAX	1	0...FEH	-48...142.5	0.75	[°C]
Upper temperature threshold for the DMTL function					
C_CL_DMTL_MAX	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum loading degree for the DMTL function					
C_T_AST_DMTL_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Min. post-starting period for canister purge diagnosis					
C_VB_DMTL_MIN	1	0...FFH	0...25.89843	0.1015625	[V]
Minimum battery threshold for the DMTL function					
C_VB_DMTL_MAX	1	0...FFH	0...25.89843	0.1015625	[V]
Maximum battery threshold for the DMTL function					
C_T_VB_DLY_DMTL	1	0...FFFFH	0...6553.5	0.1	[s]
Time delay for Vbat monitoring					
C_VS_DMTL_MAX	1	0...FFH	0...255	1	[km/h]
Max. vehicle speed					
C_TCO_ST_DMTL_MIN	1	0...FEH	-48...142.5	0.75	[°C]
Min. TCO_ST for DMTL enable					
C_FTL_DMTL_MAX	1	0...7EH	0...126	1	[l]
Max. tank level for DMTL function					
C_FTL_DMTL_MIN	1	0...7EH	0...126	1	[l]
Min. tank level for DMTL function					
C_HYS_FTL_DMTL	1	0...7EH	0...126	1	[l]
Hysteresis for tank level bit for DMTL function					
C_T_DLY_IGK_FTL	1	0...FFH	0...25.5	0.1	[s]
Minimum time for which LV_IGK must be = 1 to issue a validated level					
C_FTL_DIAG_MAX	1	0...783H	0...126	0.0655226	[l]
maximum tank level					
C_FTL_DIAG_MIN	1	80...7FH	-128...127	1	[l]
Minimum fuel tank volume for refueling detection					
C_FTL_DEC_FTL_DIAG_MIN	1	80...7FH	-128...127	1	[l]
Minimum level decrease for refueling detection					
C_T_DLY_STOP_REFU	1	0...FFFFH	0...65535	1	[s]
Delay time stop before refueling					
C_VS_FTL_DIAG_MAX	1	0...FFH	0...255	1	[km/h]
Max. VS threshold for FTL DIAG					
C_ENA_FCO_DIAG	1	0...FFFFH	0...4294.9	0.0655365	[l]
Threshold consumed fuel for diagnosis enable					
C_FCO_DIAG_MIN	1	80...7FH	-128...127	1	[l]
Lower tolerance variable for level sensor diagnosis					
C_FCO_DIAG_MAX	1	80...7FH	-128...127	1	[l]
Upper tolerance variable for level sensor diagnosis					
C_COD_DMTL	1	0...FFH	0...255	1	[-]
Calibrationconstant					
C_TCO_ST_DMTL_COLD	1	80...7FH	-96...95.25	0.75	[°C]
TCO_ST offset of TAM to enable a DMTL diagnosis					
C_SUM_CNL_LEAK_MES_FUC_MAX	1	0...FFH	0...255	1	[-]
Maximum number of aborted check filler cap checks					
C_SUM_REP_LEAK_MES_FUC_MAX	1	0...FFH	0...255	1	[-]
Maximum number of repeated check filler cap checks					
C_COD_2_DMTL	1	0...FFH	0...255	1	[-]
2nd application code word of DMTL function					
C_TAM_DMTL_CHK_FUC_MIN	1	0...FEH	-48...142.5	0.75	[°C]
Minimum ambient temperature for enabling check filler cap check					
C_T_AST_DMTL_CHK_FUC	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time after start for enabling check filler cap check					
C_VB_DMTL_CHK_FUC_MIN	1	0...FFH	0...25.89843	0.1015625	[V]
Minimum battery voltage for enabling check filler cap check					
C_VB_DMTL_CHK_FUC_MAX	1	0...FFH	0...25.89843	0.1015625	[V]
Maximum battery voltage for enabling check filler cap check					
C_CTR_ENA_DMTL_CHK_FUC	1	0...FFH	0...255	1	[-]

maximum number of enabled check filler cap checks					
C_T_ENA_DMTL_CHK_FUC	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time between two chk filler cap checks					
C_VB_DMTL_ASA_MIN	1	0...FFH	0...25.89843	0.1015625	[V]
Minimum battery threshold at end of line for DMTL					
C_VB_DMTL_ASA_MAX	1	0...FFH	0...25.89843	0.1015625	[V]
Maximum battery threshold at end of line for DMTL					
C_CTR_TCO_ST_DMTL_MIN	1	0...FFH	0...255	1	[-]
Threshold of recognized cold start for DMTL small leak test					
C_SUM_CNL_SMALL_LEAK_MES_MAX	1	0...FFH	0...255	1	[-]
Max. number of aborted miniature leak checks due to current fluctuations					
C_T_ENA_DR_DMTL_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Max. time for function request DMTL output stages					
C_T_DLY_DET_REFU	1	0...FFFFH	0...6553.5	0.1	[s]
Wait period of refueling detection					
C_CUR_DMTL_DIF_THD_REFU	1	8000...7FFFH	-200...199.99389	6.1035e-3	[mA]
Threshold pump current change for refueling detection					
C_CUR_DMTL_DIF_THD_FUC	1	8000...7FFFH	-200...199.99389	6.1035e-3	[mA]
Threshold pump current change for fuel cap detection					
C_CRLC_CUR_DMTL_FIL_REFU	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for pump current filtering in case of refueling detection					
C_CRLC_CUR_DMTL_FIL_LIM	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Time constant at the beginning of pump current filtering					
C_T_DLY_CUR_DMTL_DIF_MES	1	0...FFFFH	0...6553.5	0.1	[s]
Current decrease detection delay during measurement					
C_T_DMTL_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum time for DMTL diagnosis					
C_CUR_DMTL_DIF_THD_MES	1	0...FFFFH	0...399.99389	6.1035e-3	[mA]
Current decrease threshold during measurement					
C_DIST_REFU_ROUGH_LEAK_MIN	1	0...FFFFH	0...65535	1	[m]
Lower threshold of distance covered between refueling and rough leak suspicion					
C_T_DLY_FUC_CAN_ON	1	0...FFFFH	0...6553.5	0.1	[s]
Delay time with which Check filler Cap - light is switched on					
C_T_DLY_FUC_CAN_OFF	1	0...FFFFH	0...6553.5	0.1	[s]
Delay time with which Check filler Cap - light is switched off					
C_VS_FUC_CAN_MIN	1	0...FFH	0...255	1	[km/h]
Minimum speed for activation Check filler cap					
C_T_DMTL_EOL_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Max. time for end of line diagnosis after test unit initiation					
C_TRT_DMTL_EOL_MAX	1	0...FFFFFFFFH	0... 119304.64708	0.0278e-3	[h]
max. TRT time to run the end of line diagnosis					

Import actions:

ACTION_ERRM_StorePrevFrf (IN <XX>)
This action stores the prestored freeze frame for the diagnosis instance XX

General information:

In the scope of the legal requirements for On-Board Diagnosis (OBD II) it must be possible to detect a rough leak with a diameter of > 1.0 mm as well as a miniature leak with a diameter of > 0.5 mm within the tank system.

The leakage diagnostic unit DMTL is installed to the fresh air connector of the CP. To measure a leakage within the tank system, an excess pressure is built up in the tank system after closing the canister purge valve and changing over a valve in the diagnostic module. An excess pressure of approx. 25 hPa is built up by means of the electrical pump in the diagnostic module. The pump's power consumption is used as a measure for the tank tightness. The power consumption in a defined reference leakage serves as a reference for stating the size of a leakage.

29.5.1 Function description

The function is basically enabled via the variant bit LV_CONF_DMTL which is set to 1 via the data record if the DMTL module is installed.
The diagnostic sequence is controlled via the state machine of the central sequencing control system. It receives from the function enabling module the enable LV_ENA_SMALL_LEAK_MES for measuring the miniature leak, and the enable LV_ENA_ROUGH_LEAK_MES for measuring the rough leak. Subsequently, the state machine starts a reference leakage measurement. After the time C_T_REF_LEAK_MES_MAX has elapsed, a transition to the rough leak diagnosis is performed, except in case the current monitoring detects a module error or the diagnosis is interrupted due to a detected diagnosis.
The system detects a rough leak suspect if the pump current does not exceed the threshold CUR_DMTL_THD_ROUGH_LEAK within the time T_DMTL_ROUGH_LEAK and if the pump current does not reach the threshold CUR_DMTL_THD_ROUGH_LEAK_LEN after the extension time T_DMTL_ROUGH_LEAK. In case of a rough leak suspect a second reference measurement is performed on the basis of which the final decision on a rough leck suspect is made. Subsequently, a transition to the miniature leak diagnosis is performed. The tank system is detected to be “tight” if the pump current exceeds the reference current CUR_DMTL_REF_LEAK.
If the pump current is detected to have reached “steady state” twice after the time T_DMTL_SMALL_LEAK, a second reference measurement is performed. If a current increase is detected, a miniature leakage is diagnosed. In case of a current reduction, the tank system is detected to be “tight”.
At the end of the diagnosis, the pump motor is switched off and the change-over valve is reset to its regeneration position.
The errors are indicated with the logic variable LV_ERR_DMTLM for a DMTL module error, LV_ERR_ROUGH_LEAK in case a rough leak is present, or LV_ERR_SMALL_LEAK if a miniature leak has been detected.

A current diagnosis is indicated with the bit LV_DMTL_ACT; interruption of diagnosis is requested via the bit LV_DMTL_STOP. The diagnosis is interrupted if tank cover open has been detected, at the beginning of refueling, in case of an excessive battery voltage fluctuation during diagnosis, in case of decreasing pump current during the measuring phase or if a short-circuit of the pump output stage has been detected.
During the reference stage the function is finished without results if humidity has been detected.
The pump current CUR_DMTL is filtered after conversion to a standard voltage of 12 V in the function block of the pump current filtering. At the beginning of the reference measurement and at the beginning of the rough leak measurement, a smaller time constant is used to enable short transient period of the filter.
Moreover, the filtered battery voltage is sampled at the end of the reference measurement, and the change is monitored in the following rough/miniature leak measurement for the threshold C_VB_DMTL_DIF. The result LV_VB_DIF_MAX = 1 results in interruption of the diagnosis.
During the reference leakage measurement it is taken care of current fluctuations which can lead to an interruption of the diagnosis.

general specification

The function block "Fuel cap and refueling check" differentiates the corrected pump current and checks for exceeding a positive threshold C_THD_CUR_DMTL_DIF_REFU, i.e. for increasing pump current to detect "refueling", as the tank pressure and the pump current rise on refueling. When the tank cap is opened, the tank pressure collapses; thus the pump current is reduced. When the negative threshold C_THD_CUR_DMTL_DIF_FUC is no longer reached, the logic variable LV_DET_FUC_OPEN is set to indicate that the fuel cap is open.

A slow decrease of C_THD_CUR_DMTL_DIF_MES of the pump current leads to an interruption of the diagnosis.

The error and cycle flags are generated in the error block.

Enabling of the minimum leakage diagnosis is essentially initiated while the control unit is closed during off delay, below a certain level, while the vehicle does not run, if the CP is not excessively loaded, at ambient temperatures within a certain band width, the battery voltage within a certain band width and while the CPS is closed.

Moreover, the system checks whether a minimum time since engine start C_T_AST_DMTL_MIN has elapsed, a cold start was given and whether the engine is running.

When the bit LV_REQ_PWL_DMTL is set, the extension of the post-operating phase is enforced up to the end of diagnosis. The diagnosis is started with the delay C_T_DLY_ES_PWL after engine start.

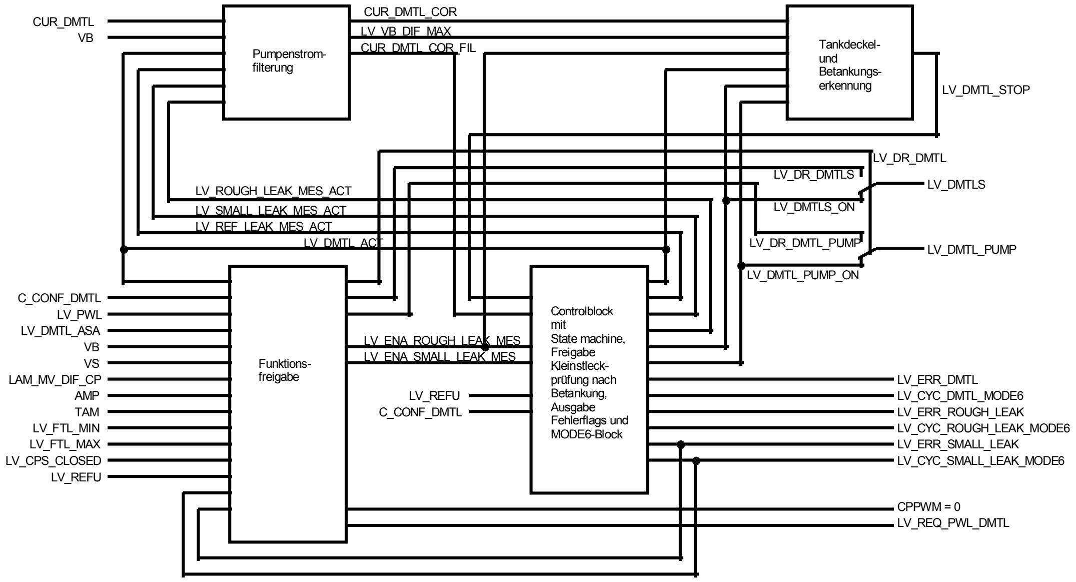
In the function block “Demand rough leak check due to Check Filler Cap” a rough leak check is activated after refuelling has been detected. With this check a tank cover check is performed. The error entry can be prevented while this check. If a rough leak has been detected a info in the instrument board will be activated in the combi instrument.

Application conditions:

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Formula section:

```
If        LV_DR_DMTL = 1
    Then  LV_DMTLS = LV_DR_DMTLS
          LV_DMTL_PUMP = LV_DR_DMTL_PUMP
    Else  LV_DMTLS = LV_DMTLS_ON
          LV_DMTL_PUMP = LV_DMTL_PUMP_ON
Endif
```

29.5.2 Central sequencing control

General information:

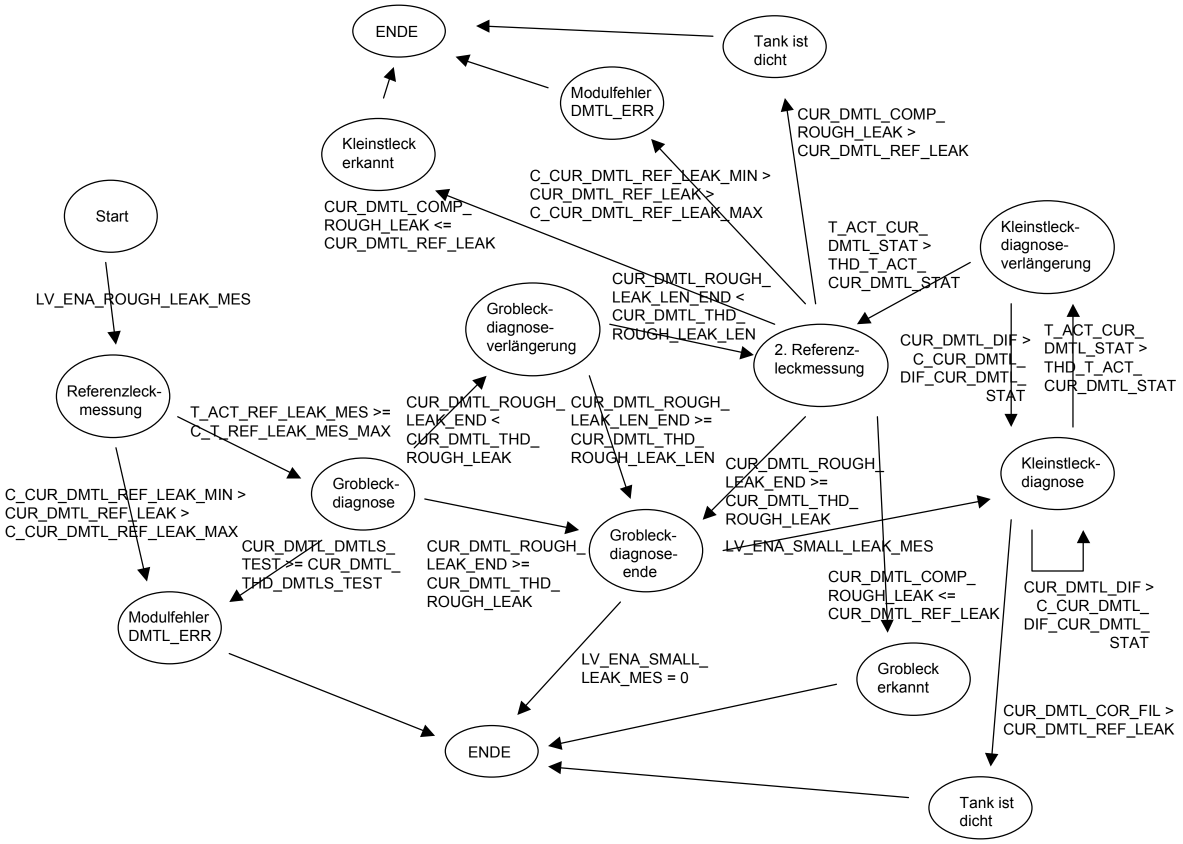
The sequence of measurements of the individual currents and coordination of the various output stages pertaining to the DMTL module is ensured by the central sequencing control.

Application conditions:

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Formula section:

Diagnosis times for rough and small leak measurement:

```
If     LV_ERR_FTL = 0                              AND
       LV_ERR_FTL_PLAUS = 0
Then       If    LV_ENA_CHK_FUC_ROUGH_LEAK = 0
           Then  T_ACT_ROUGH_LEAK        = IP_T_ACT_ROUGH_LEAK
           Else  T_ACT_ROUGH_LEAK        = IP_T_ACT_ROUGH_LEAK_CHK_FUC
           Endif
           T_ACT_SMALL_LEAK        = IP_T_ACT_SMALL_LEAK
           T_CUR_DMTL_STAT         = IP_T_CUR_DMTL_STAT
Else       If    LV_ENA_CHK_FUC_ROUGH_LEAK = 0
           Then  T_ACT_ROUGH_LEAK        = C_T_ACT_ROUGH_LEAK
```

```
        Else    T_ACT_ROUGH_LEAK        = C_T_ACT_ROUGH_LEAK_CHK_FUC
        Endif
        T_ACT_SMALL_LEAK        = C_T_ACT_SMALL_LEAK
        T_CUR_DMTL_STAT         = C_T_CUR_DMTL_STAT
Endif
```

STATE Machine:

state diagram - "start" **STATE_DMTL = 0:**

```
        Reset condition:    LV_DMTL_STOP = 1                    OR
                            LV_ENA_ROUGH_LEAK_MES = 0
                                (is approached from all states, if fulfilled)

            LV_DMTL_ACT                     = 0    Diagnosis inactive
            LV_DMTL_PUMP_ON                 = 0    Pump OFF
            LV_DMTLS_ON                     = 0    Valve OFF
            LV_CHK_DMTL_CUR_DMTL_VLD        = 0    Diagnostic module not finished
            LV_CYC_ROUGH_LEAK_SUSP          = 0    Rough leak diagn. not finished
            LV_CYC_SMALL_LEAK_MODE6         = 0    Miniature leak diag. not finished
            LV_REF_LEAK_ROUGH_LEAK_SUSP = 0
            If      LV_ERR_SIG_DMTL = 1
                    Then
                    LV_CUR_DMTL_REF_MES_MAX     = 0
                    LV_CUR_DMTL_REF_MES_MIN     = 0
                    LV_CUR_DMTL_DMTLS_TEST      = 0
            Endif

        If              LV_CONF_DMTL = 0 (no DMTL - module installed)
            Then
            LV_CUR_DMTL_REF_MES_MAX             = 0
            LV_CUR_DMTL_REF_MES_MIN             = 0
            LV_CUR_DMTL_DMTLS_TEST              = 0
            LV_CHK_DMTL_CUR_DMTL_VLD            = 0
            LV_ERR_DET_SMALL_LEAK_MIN = 0
            LV_CYC_SMALL_LEAK_MODE6     = 0    Miniature leakage diagn. not finished
```

```
                LV_ROUGH_LEAK_SUSP           = 0
                LV_CYC_ROUGH_LEAK_SUSP       = 0     Rough leakage diagn. not finished
            Endif

If          LV_ENA_ROUGH_LEAK_MES = 1
    Then    Transition to state diagram
            “reference leak measurement” / STATE_DMTL = 1H
            LV_DMTL_ACT = 1                          Diagnosis active
    Else    Remaining in state diagramm “Start” / STATE_DMTL = 0H
Endif
```

State diagram - “reference leak measurement“ STATE_DMTL = 1H:

```
Initialization: LV_DMTL_PUMP_ON              = 1     Pump ON
                T_ACT_REF_LEAK_MES           = 0
                CUR_DMTL_REF_LEAK            = 0
                LV_ROUGH_LEAK_MES_LEN        = 0
Calculations and conditions for setting:
    T_ACT_REF_LEAK_MES = T_ACT_REF_LEAK_MES + ΔT (= 100ms)
    CUR_DMTL_REF_LEAK = CUR_DMTL_REF_LEAK_EOL =
                        CUR_DMTL_COR_FIL + C_CUR_DMTL_REF_OFS

    If      T_ACT_REF_LEAK_MES > C_T_REF_LEAK_MES_MAX          max. time
                                    for reference leakage measurement elapsed
    Then    If      CUR_DMTL_REF_LEAK > C_CUR_DMTL_REF_LEAK_MAX
            Then            LV_CUR_DMTL_REF_MES_MAX    = 1    max. error detected
                            LV_CUR_DMTL_REF_MES_MIN    = 0
                            LV_CUR_DMTL_DMTLS_TEST     = 0    no plausibility error
                            LV_CHK_DMTL_CUR_DMTL_VLD   = 1
            Else if         CUR_DMTL_REF_LEAK ≤ C_CUR_DMTL_REF_LEAK_MIN:
                    Then    LV_CUR_DMTL_REF_MES_MAX    = 0
                            LV_CUR_DMTL_REF_MES_MIN    = 1    min. error detected
                            LV_CUR_DMTL_DMTLS_TEST     = 0    no plausibility error
                            LV_CHK_DMTL_CUR_DMTL_VLD   = 1
                    Endif
            Endif
    Endif
```

```
If      LV_CUR_DMTL_REF_MES_MAX = 1        OR
        LV_CUR_DMTL_REF_MES_MIN = 1
Then    transition to the state diagram “module error DMTL_ERR” / STATE_DMTL = BH
Else    transition to state diagram “rough leak measurement” / STATE_DMTL = 2H
Endif
```

State diagram - “rough leak diagnosis” STATE_DMTL = 2H:

```
Initialization:   LV_DMTLS_ON                = 1    Solenoid valve ON
                  T_ACT_LEAK_MES             = 0
CUR_DMTL_ROUGH_LEAK_MIN = CUR_DMTL_ROUGH_LEAK_MIN_EOL =
                                 CUR_DMTL_COR_FIL      minimum current
CUR_DMTL_DIF_REF_ROUGH_LEAK = CUR_DMTL_REF_LEAK -
                                 CUR_DMTL_ROUGH_LEAK_MIN
CUR_DMTL_ROUGH_LEAK_END =        CUR_DMTL_COR_FIL
If    LV_ENA_CHK_FUC_ROUGH_LEAK = 0
Then        CUR_DMTL_THD_ROUGH_LEAK = CUR_DMTL_ROUGH_LEAK_MIN +
                                      IP_FAC_CUR_DMTL_ROUGH *
                                        CUR_DMTL_DIF_REF_ROUGH_LEAK
Else        CUR_DMTL_THD_ROUGH_LEAK = CUR_DMTL_ROUGH_LEAK_MIN +
                                    IP_FAC_CUR_DMTL_CHK_FUC_ROUGH *
                                        CUR_DMTL_DIF_REF_ROUGH_LEAK
Endif
CUR_DMTL_THD_DMTLS_TEST =        CUR_DMTL_REF_LEAK +
                                 C_CUR_DMTL_COR_FIL_OFS_AMP
CUR_DMTL_DMTLS_TEST =            CUR_DMTL_ROUGH_LEAK_MIN
```

Calculations and conditions for setting:

```
    T_ACT_LEAK_MES = T_ACT_LEAK_MES + ΔT
    CUR_DMTL_ROUGH_LEAK_END = CUR_DMTL_COR_FIL      Rough leak current
    If      CUR_DMTL_COR_FIL < CUR_DMTL_ROUGH_LEAK_MIN
            Then  CUR_DMTL_ROUGH_LEAK_MIN =
                  CUR_DMTL_ROUGH_LEAK_MIN_EOL = CUR_DMTL_COR_FIL
                  CUR_DMTL_DIF_REF_ROUGH_LEAK = CUR_DMTL_REF_LEAK -
                                              CUR_DMTL_ROUGH_LEAK_MIN

                  If      LV_ENA_CHK_FUC_ROUGH_LEAK = 0
                  Then    CUR_DMTL_THD_ROUGH_LEAK =
                  CUR_DMTL_ROUGH_LEAK_MIN + IP_FAC_CUR_DMTL_ROUGH *
                                      CUR_DMTL_DIF_REF_ROUGH_LEAK
```

```
                  Else    CUR_DMTL_THD_ROUGH_LEAK =
                          CUR_DMTL_ROUGH_LEAK_MIN +
                          IP_FAC_CUR_DMTL_CHK_FUC_ROUGH *
                          CUR_DMTL_DIF_REF_ROUGH_LEAK
                  Endif
                  CUR_DMTL_DMTLS_TEST = CUR_DMTL_ROUGH_LEAK_MIN
      Endif
```

Checks:

```
      If      T_ACT_LEAK_MES > C_T_CUR_DMTL_ROUGH_LEAK_MIN   AND
              T_ACT_LEAK_MES < C_T_CUR_DMTL_ROUGH_LEAK_MAX   AND
              CUR_DMTL_DMTLS_TEST >= CUR_DMTL_THD_DMTLS_TEST
              Then  LV_CUR_DMTL_REF_MES_MAX   = 0   no maximum error, DMTL module
                    LV_CUR_DMTL_REF_MES_MIN   = 0   no minimum error, DMTL module
                    LV_CUR_DMTL_DMTLS_TEST    = 1   Plausibility error, DMTL
                    LV_CHK_DMTL_CUR_DMTL_VLD  = 1   Module diagn. concluded
                    Transition to the state diagram
                    “module error DMTL_ERR” / STATE_DMTL = BH
      Endif
      If      LV_ENA_CHK_FUC_ROUGH_LEAK = 0
      Then    If    T_ACT_LEAK_MES > T_ACT_ROUGH_LEAK                    AND
                    CUR_DMTL_ROUGH_LEAK_END < CUR_DMTL_THD_ROUGH_LEAK
              Then  Transition to state diagram
                    “rough leak diagnosis extension” / STATE_DMTL = 3H
              Endif
      Else    If    T_ACT_LEAK_MES > T_ACT_ROUGH_LEAK                    AND
                    CUR_DMTL_ROUGH_LEAK_END < CUR_DMTL_THD_ROUGH_LEAK
              Then  Transition to state diagram
                    “rough leak detected” / STATE_DMTL = AH
              Endif
      Endif

      If      T_ACT_LEAK_MES > C_T_REF_TEST_ROUGH_LEAK           AND
              CUR_DMTL_ROUGH_LEAK_END >= CUR_DMTL_THD_ROUGH_LEAK
              Then  Transition to state diagram
                    “end of rough leak diagnosis” / STATE_DMTL = 4H
```

```
Endif
```

State diagram - “rough leak diagnosis extension”   STATE_DMTL = 3H:

```
Initialization:     LV_ROUGH_LEAK_MES_LEN = 1
                    LV_DMTLS_ON = 1
                    CUR_DMTL_THD_ROUGH_LEAK_LEN = CUR_DMTL_REF_LEAK –
                                        CUR_DMTL_DIF_REF_ROUGH_LEAK *
                                        IP_FAC_CUR_DMTL_REF
                    CUR_DMTL_ROUGH_LEAK_LEN_END = CUR_DMTL_COR_FIL
Calculations and conditions for setting:
    T_ACT_LEAK_MES = T_ACT_LEAK_MES + ΔT
    CUR_DMTL_ROUGH_LEAK_LEN_END = CUR_DMTL_COR_FIL
    If      CUR_DMTL_COR_FIL < CUR_DMTL_ROUGH_LEAK_MIN
            Then    CUR_DMTL_ROUGH_LEAK_MIN =
                    CUR_DMTL_ROUGH_LEAK_MIN_EOL = CUR_DMTL_COR_FIL
                    CUR_DMTL_DIF_REF_ROUGH_LEAK = CUR_DMTL_REF_LEAK –
                                          CUR_DMTL_ROUGH_LEAK_MIN

                    CUR_DMTL_THD_ROUGH_LEAK_LEN = CUR_DMTL_REF_LEAK –
                                          IP_FAC_CUR_DMTL_REF *

                                        CUR_DMTL_DIF_REF_ROUGH_LEAK
    Endif

Checks:
    If      CUR_DMTL_ROUGH_LEAK_LEN_END >=
            CUR_DMTL_THD_ROUGH_LEAK_LEN
            Then    Transition to state diagram
                    “end of rough leak diagnosis”  / STATE_DMTL = 4H
            Else if         T_ACT_LEAK_MES > T_ACT_SMALL_LEAK
                    Then    LV_REF_LEAK_ROUGH_LEAK_SUSP = 1
                            Transition to state diagram
                            “2. reference leak diagnosis” / STATE_DMTL = 7H
            Endif
    Endif
```

State diagram – “rough leak diagnosis end”   STATE_DMTL = 4H:

```
Initialization:     LV_ROUGH_LEAK_SUSP          = 0
                    LV_CYC_ROUGH_LEAK_SUSP      = 1
    If      LV_ENA_SMALL_LEAK_MES = 0           no miniature leak check
            Then  LV_CUR_DMTL_DMTLS_TEST        = 0   no plausibility error DMTL
                  LV_CUR_DMTL_REF_MES_MIN       = 0   no minimum error, DMTL module
                  LV_CUR_DMTL_REF_MES_MAX       = 0   no maximum error, DMTL module
                  LV_CHK_DMTL_CUR_DMTL_VLD      = 1   Module diagn. concluded
                  Transition to state diagram "end" / STATE_DMTL = CH
            Else  LV_ROUGH_LEAK_MES_LEN = 0
                  Transition to state diagram
                  "miniature leak diagnosis" / STATE_DMTL = 5H
    Endif
```

State diagram – "miniature leak diagnosis"   STATE_DMTL = 5H:

```
Initialization:     T_ACT_CUR_DMTL_STAT = 0
                    CUR_DMTL_DIF = 0
                    CUR_DMTL_COMP_ROUGH_LEAK = CUR_DMTL_COR_FIL
                    LV_CYC_ROUGH_LEAK_SUSP = 0
                    LV_DMTLS_ON = 1
Calculations and conditions for setting:
                    T_ACT_CUR_DMTL_STAT = T_ACT_CUR_DMTL_STAT + ΔT
                    CUR_DMTL_DIF =  | CUR_DMTL_COR_FIL -
                                      CUR_DMTL_COMP_ROUGH_LEAK |
                    T_ACT_LEAK_MES = T_ACT_LEAK_MES + ΔT
Checks:
    If      CUR_CMTL_COR_FIL > CUR_DMTL_REF_LEAK
            Then  CUR_DMTL_SMALL_LEAK_END = CUR_DMTL_COR_FIL
                  LV_CUR_DMTL_DMTLS_TEST        = 0   no plausibility error, DMTL
                  LV_CUR_DMTL_REF_MES_MIN       = 0   no minimum error, DMTL module
                  LV_CUR_DMTL_REF_MES_MAX       = 0   no maximum error DMTL module
                  LV_CHK_DMTL_CUR_DMTL_VLD      = 1   module diagn. concluded
                  Transition to state diagram "tank is tight" / STATE_DMTL = 8H
    Endif

    If      CUR_DMTL_DIF >= C_CUR_DMTL_DIF_CUR_DMTL_STAT
            Then  Transition to state diagram
```

```
                    "miniature leak diagnosis" / STATE_DMTL = 5H
    Endif

    If      T_ACT_CUR_DMTL_STAT > T_CUR_DMTL_STAT                AND
            T_ACT_LEAK_MES > T_ACT_SMALL_LEAK
            Then    CUR_DMTL_COMP_ROUGH_LEAK = CUR_DMTL_COR_FIL
                    Transition to state diagram
                    "miniature leak diagnosis extension" / STATE_DMTL = 6H
    Endif
```

State diagram – "miniature leak diagnosis extension" STATE_DMTL = 6H:

```
Initialization:     CUR_DMTL_DIF = 0
                    T_ACT_CUR_DMTL_STAT = 0
                    CUR_DMTL_COMP_SMALL_LEAK_LEN = CUR_DMTL_COR_FIL
                    LV_SMALL_LEAK_MES_LEN = 1
Calculations and conditions for setting:
                    T_ACT_CUR_DMTL_STAT = T_ACT_CUR_DMTL_STAT + ∆T
                    T_ACT_LEAK_MES = T_ACT_LEAK_MES + ∆T
                    CUR_DMTL_DIF = |CUR_DMTL_COR_FIL –
                                      CUR_DMTL_COMP_SMALL_LEAK_LEN|

Checks:
    If      CUR_DMTL_DIF > C_CUR_DMTL_DIF_CUR_DMTL_STAT
            Then    LV_SMALL_LEAK_MES_LEN = 0
                    Transition to state diagram
                    "miniature leak diagnosis" / STATE_DMTL = 5H
    Endif

    If      T_ACT_CUR_DMTL_STAT > T_CUR_DMTL_STAT *
                                          C_FAC_T_CUR_DMTL_STAT_LEN
            Then    LV_SMALL_LEAK_MES_LEN = 0
                    CUR_DMTL_ROUGH_LEAK_END = CUR_DMTL_COR_FIL
                    Transition to state diagram
                    "2. reference leak measurement" / STATE_DMTL = 7H
    Endif
```

State diagram – “2. reference leak measurement” STATE_DMTL = 7H:

Initialtization: T_ACT_REF_LEAK_MES = 0
CUR_DMTL_REF_LEAK = 0
LV_DMTLS_ON = 0
LV_CHK_DMTL_CUR_DMTL_VLD = 0

Calculations and conditions for setting:

```
T_ACT_REF_LEAK_MES = T_ACT_REF_LEAK_MES + ΔT

CUR_DMTL_REF_LEAK = CUR_DMTL_REF_LEAK_EOL =
                CUR_DMTL_COR_FIL + C_CUR_DMTL_REF_OFS
CUR_DMTL_ROUGH_LEAK_MIN = CUR_DMTL_REF_LEAK -
                CUR_DMTL_DIF_REF_ROUGH_LEAK

CUR_DMTL_THD_ROUGH_LEAK = CUR_DMTL_ROUGH_LEAK_MIN +
                CUR_DMTL_DIF_REF_ROUGH_LEAK *
                IP_FAC_CUR_DMTL_ROUGH

CUR_DMTL_THD_ROUGH_LEAK_LEN = CUR_DMTL_REF_LEAK –
                CUR_DMTL_DIF_REF_ROUGH_LEAK *
                IP_FAC_CUR_DMTL_REF
```

Checks:

```
If    T_ACT_REF_LEAK_MES > C_T_REF_LEAK_MES_REP_MAX          AND
      CUR_DMTL_REF_LEAK > C_CUR_DMTL_REF_LEAK_MAX
      Then  LV_CUR_DMTL_DMTLS_TEST     = 0   no plausibility error, DMTL
            LV_CUR_DMTL_REF_MES_MIN    = 0   no minimum error, DMTL module
            LV_CUR_DMTL_REF_MES_MAX    = 1   maximum error, DMTL module
            LV_CHK_DMTL_CUR_DMTL_VLD   = 1   module diagn. concluded
            Transition to the state diagram
            “module error DMTL_ERR” / STATE_DMTL = BH
Endif

If    T_ACT_REF_LEAK_MES > C_T_REF_LEAK_MES_REP_MAX          AND
      CUR_DMTL_REF_LEAK < C_CUR_DMTL_REF_LEAK_MIN
      Then  LV_CUR_DMTL_DMTLS_TEST     = 0   no plausibility error, DMTL
            LV_CUR_DMTL_REF_MES_MIN    = 1   minimum error, DMTL module
            LV_CUR_DMTL_REF_MES_MAX    = 0   no maximum error, DMTL module
            LV_CHK_DMTL_CUR_DMTL_VLD   = 1   module diagn. concluded
            Transition to the state diagram
```

```
                    "module error" DMTL_ERR / STATE_DMTL = BH
    Endif
    If      [T_ACT_REF_LEAK_MES > C_T_REF_LEAK_MES_REP_MAX                AND
            CUR_DMTL_ROUGH_LEAK_LEN_END <
            CUR_DMTL_THD_ROUGH_LEAK_LEN                                   AND
            LV_ROUGH_LEAK_MES_LEN = 1]
            OR
            [T_ACT_REF_LEAK_MES > C_T_REF_LEAK_MES_REP_MAX                AND
            CUR_DMTL_ROUGH_LEAK_END < CUR_DMTL_THD_ROUGH_LEAK             AND
            LV_ROUGH_LEAK_MES_LEN = 0]
            Then  LV_CUR_DMTL_DMTLS_TEST      = 0   no plausibility error, DMTL
                  LV_CUR_DMTL_REF_MES_MIN     = 0   no minimum error, DMTL module
                  LV_CUR_DMTL_REF_MES_MAX     = 0   no maximum error, DMTL module
                  LV_CHK_DMTL_CUR_DMTL_VLD    = 1   module diagn. concluded
                  Transition to state diagram "rough leak detected" / STATE_DMTL = AH
    Endif

    If      T_ACT_REF_LEAK_MES > C_T_REF_LEAK_MES_REP_MAX                 AND
            CUR_DMTL_ROUGH_LEAK_END >= CUR_DMTL_THD_ROUGH_LEAK            AND
            LV_REF_LEAK_ROUGH_LEAK_SUSP = 1
            Then  LV_REF_LEAK_ROUGH_LEAK_SUSP      = 0
                  T_ACT_LEAK_MES                   = 0
                  Transition to state diagram
                  "end of rough leak diagnosis" / STATE_DMTL = 4H
    Endif

    If      T_ACT_REF_LEAK_MES > C_T_REF_LEAK_MES_REP_MAX                 AND
            CUR_DMTL_COMP_ROUGH_LEAK <= CUR_DMTL_REF_LEAK                 AND
            LV_REF_LEAK_ROUGH_LEAK_SUSP = 0
            Then  CUR_DMTL_SMALL_LEAK_END = CUR_DMTL_COMP_ROUGH_LEAK
                  LV_CUR_DMTL_DMTLS_TEST      = 0   no plausibility error, DMTL
                  LV_CUR_DMTL_REF_MES_MIN     = 0   no minimum error, DMTL module
                  LV_CUR_DMTL_REF_MES_MAX     = 0   no maximum error, DMTL module
                  LV_CHK_DMTL_CUR_DMTL_VLD    = 1   module diagn. concluded
                  Transition to state diagram
                  "miniature leak detected" / STATE_DMTL = 9H
```

```
Endif

If      T_ACT_REF_LEAK_MES > C_T_REF_LEAK_MES_REP_MAX            AND
        CUR_DMTL_COMP_ROUGH_LEAK > CUR_DMTL_REF_LEAK             AND
        LV_REF_LEAK_ROUGH_LEAK_SUSP = 0
        Then  CUR_DMTL_SMALL_LEAK_END = CUR_DMTL_COMP_ROUGH_LEAK
              LV_CUR_DMTL_DMTLS_TEST     = 0   no plausibility error, DMTL
              LV_CUR_DMTL_REF_MES_MIN    = 0   no minimum error, DMTL module
              LV_CUR_DMTL_REF_MES_MAX    = 0   no maximum error, DMTL module
              LV_CHK_DMTL_CUR_DMTL_VLD   = 1   module diagn. concluded
              Transition to state diagram “tank is tight” / STATE_DMTL = 8H
Endif
```

State diagram – “tank is tight”   STATE_DMTL = 8H:

```
LV_ERR_DET_SMALL_LEAK_MIN = 0    no miniature leak detected
LV_CYC_SMALL_LEAK_MODE6   = 1    miniature leak diagnisis finished
Transition to state diagram “end” / STATE_DMTL = CH
```

State diagram – “miniature leak detected”   STATE_DMTL = 9H:

```
LV_ERR_DET_SMALL_LEAK_MIN = 1    miniature leak detected
LV_CYC_SMALL_LEAK_MODE6   = 1
ERR_SYM_SMALL_LEAK = “MIN”
Transition to state diagram “end” / STATE_DMTL = CH
```

State diagram – “rough leak detected”   STATE_DMTL = AH:

```
LV_ROUGH_LEAK_SUSP        = 1    Rough leak suspected
LV_CYC_ROUGH_LEAK_SUSP    = 1    Rough leak diagnosis finished
Transition to state diagram “end” / STATE_DMTL = CH
```

State diagram – “module error DMTL_ERR”   STATE_DMTL = BH:

```
Transition to state diagram “end” / STATE_DMTL = CH
```

State diagram – “end”   STATE_DMTL = CH:

```
    LV_DMTL_PUMP_ON           = 0    Pump OFF
    LV_DMTLS_ON               = 0    valve OFF
```

LV_DMTL_ACT = 0 Diagnosis finished

29.5.3 Pump current of DMTL module

General information:

The most important analog input of the "tank leakage diagnostic function" represents the pump current of the DMTL module. The current to be evaluated is recorded and calculated by voltage measurement at the corresponding shunts. Subsequently, the calculated current is corrected and filtered before it is included in calculation of the function as an input variable.

29.5.3.1 Pump current DMTL

General information:

The current of the pump motor by means of which an excess pressure is built up in the tank system serves as measure for the detection of a leakage in the tank system.
To detect the equivalent current from the measured voltage V_DMTL which is generated in the control unit using the measurement shunt, this requirement is taken into consideration via the calibratable constant C_R_DMTL.
The module is calculated with a time base of 10 ms.

Application conditions:

Initialisation: CUR_DMTL = 0

Recurrence: 10 ms

Activation: LV_CONF_DMTL = 1

Formula section:

CUR_DMTL = V_DMTL / C_R_DMTL

29.5.3.2 Filtering of pump current

General information:

The pump current is filtered after conversion to the referenced battery voltage. At the beginning and at the end of the reference measurement, a greater time constant is used to ensure a short filter settling time.
A correction factor for the pump current is taken as a function of the calculated value VB_DMTL_FIL via a characteristic IP_FAC_CUR_DMTL. Thus, the pump current CUR_DMTL is corrected depending on the battery voltage, and results in relation to CUR_DMTL_COR.

Application conditions:

Initialisation: CUR_DMTL_COR = 0
CUR_DMTL_COR_FIL = 0

Recurrence: 10 ms

Activation: LV_CONF_DMTL = 1

Formula section:

```
CUR_DMTL_COR = CUR_DMTL * IP_FAC_CUR_DMTL

If  STATE_DMTL = 0H → 1H   OR
    STATE_DMTL = 1H → 2H   OR
    STATE_DMTL = 3H → 7H   OR
    STATE_DMTL = 6H → 7H   OR
    STATE_DMTL = 7H → 4H
    Then  T_TIMER_DLY_DMTLS = C_T_DLY_DMTLS
Endif

If  T_TIMER_DLY_DMTLS > 0
    Then  $CUR\_DMTL\_COR\_FIL_{(n)} = CUR\_DMTL\_COR\_FIL_{(n-1)} + (CUR\_DMTL\_COR_{(n)} - CUR\_DMTL\_COR\_FIL_{(n-1)}) * C\_CRLC\_CUR\_DMTL\_FIL\_LIM$
          T_TIMER_DLY_DMTLS = T_TIMER_DLY_DMTLS – 10ms
    Else  $CUR\_DMTL\_COR\_FIL_{(n)} = CUR\_DMTL\_COR\_FIL_{(n-1)} + (CUR\_DMTL\_COR_{(n)} - CUR\_DMTL\_COR\_FIL_{(n-1)}) * C\_CRLC\_CUR\_DMTL\_FIL$
Endif

CUR_DMTL_COR_FIL_EOL = CUR_DMTL_COR_FIL          #only in case of tester diag
```

29.5.3.3 VB depending pump current correction

General information:

At the end of the reference measurement, the filtered battery voltage VB_DMTL_REF is measured and monitored for a higher deviation in the following rough or miniature leak measurement. If a high battery voltage deviation is detected, LV_VB_DIF_MAX is set and the diagnosis is aborted.

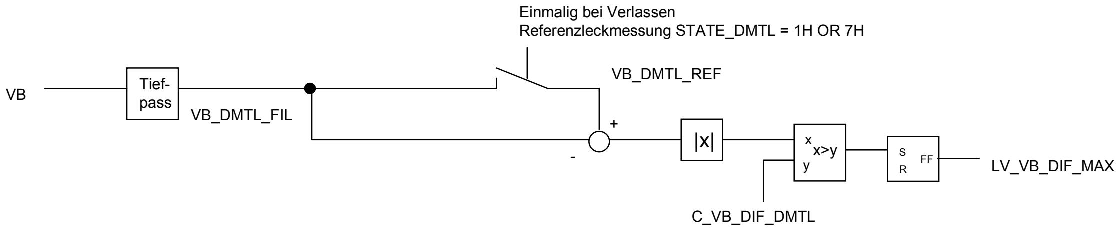
Application conditions:

Initialisation: LV_VB_DIF_MAX is restored from the NVMY
VB_DMTL_REF = VB_DMTL_FIL = 12V

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Formula section:

```
If      LV_DMTL_ACT = 0 → 1
    Then    T_DLY_VB_DMTL_FIL = C_T_DLY_DMTLS
Endif
If      LV_DMTL_ACT = 1 AND T_DLY_VB_DMTL > 0
    Then    $VB\_DMTL\_FIL_{(n)} = VB\_DMTL\_FIL_{(n-1)} + (VB_{(n)} - VB\_DMTL\_FIL_{(n-1)}) * C\_CRLC\_VB\_DMTL\_FIL\_LIM$
            T_DLY_VB_DMTL = T_DLY_VB_DMTL – 100 ms
Endif
If      LV_DMTL_ACT = 1 AND T_DLY_VB_DMTL = 0
```

```
        Then  $VB\_DMTL\_FIL_{(n)}$ = $VB\_DMTL\_FIL_{(n-1)}$ +
                    $(VB_{(n)} - VB\_DMTL\_FIL_{(n-1)})$ * C_CRLC_VB_DMTL_FIL
    Endif

    If    STATE_DMTL = 1H → 2H    OR
          STATE_DMTL = 7H → 4H
        Then  VB_DMTL_REF = VB_DMTL_FIL
        Else  VB_DMTL_REF = VB_DMTL_REF
    Endif


    If    [STATE_DMTL = 2H               OR
          STATE_DMTL = 3H                OR
          STATE_DMTL = 5H                OR
          STATE_DMTL = 6H]               AND
          [If    LV_ENA_CHK_FUC_ROUGH_LEAK = 0
           Then  |VB_DMTL_REF - VB_DMTL_FIL| > C_VB_DMTL_DIF
           Else  |VB_DMTL_REF - VB_DMTL_FIL| > C_VB_DMTL_DIF_CHK_FUC
           Endif]
        Then  LV_VB_DIF_MAX = 1
        Else  LV_VB_DIF_MAX = 0
    Endif

    If    LV_DMTL_ACT = 0 → 1
        Then  LV_VB_DIF_MAX = 0
    Endif
```

29.5.4 Interruption of function due to current fluctuations

General information:

The function block monitors the filtered and corrected pump current for excessively high current fluctuations during the reference leakage measurement. If this case is detected, the tank leakage diagnosis is aborted. The interruption is cumulated in a counter.

Application conditions:

Initialisation: LV_CUR_DMTL_REF_DIF_MAX is restored out of NVMY
CTR_CUR_DMTL_REF_DIF_MAX is restored out of NVMY

Recurrence: 100 ms

Activation: LV_DMTL_ACT = 1

Formula section:

```
If        STATE_DMTL = 0H → 1H   OR
          STATE_DMTL = 3H → 7H   OR
          STATE_DMTL = 6H → 7H
    Then  T_DLY_CUR_DMTL_REF_DIF = C_T_DLY_DMTLS
Endif

If        [STATE_DMTL = 1H OR
          STATE_DMTL = 7H]       AND
          T_DLY_CUR_DMTL_REF_DIF > 0
    Then  CUR_DMTL_COR_FIL_MAX = CUR_DMTL_COR_FIL
          CUR_DMTL_COR_FIL_MIN = CUR_DMTL_COR_FIL
          T_DLY_CUR_DMTL_REF_DIF = T_DLY_CUR_DMTL_REF_DIF – 100 ms
Endif

If        [STATE_DMTL = 1H OR
          STATE_DMTL = 7H]       AND
          T_DLY_CUR_DMTL_REF_DIF = 0
    Then  CUR_DMTL_REF_DIF_MAX =
          MAX(CUR_DMTL_COR_FIL. CUR_DMTL_COR_FIL_MAX) -
          MIN(CUR_DMTL_COR_FIL. CUR_DMTL_COR_FIL_MIN)
Endif

If        CUR_DMTL_REF_DIF_MAX > C_CUR_DMTL_REF_DIF_MAX
    Then  LV_CUR_DMTL_REF_DIF_MAX = 1
    Else  LV_CUR_DMTL_REF_DIF_MAX = 0
Endif
```

```
If          LV_DMTL_ACT = 0 → 1
      Then  LV_CUR_DMTL_REF_DIF_MAX = 0
Endif


If          LV_CUR_DMTL_REF_DIF_MAX = 0 → 1           OR
            LV_CUR_DMTL_THD_DIF_MES = 0 → 1
      Then  CTR_CUR_DMTL_REF_DIF_MAX = CTR_CUR_DMTL_REF_DIF_MAX + 1
      Else  CTR_CUR_DMTL_REF_DIF_MAX = CTR_CUR_DMTL_REF_DIF_MAX
Endif
```

29.5.5 Function enabling

Application conditions:

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

General information:

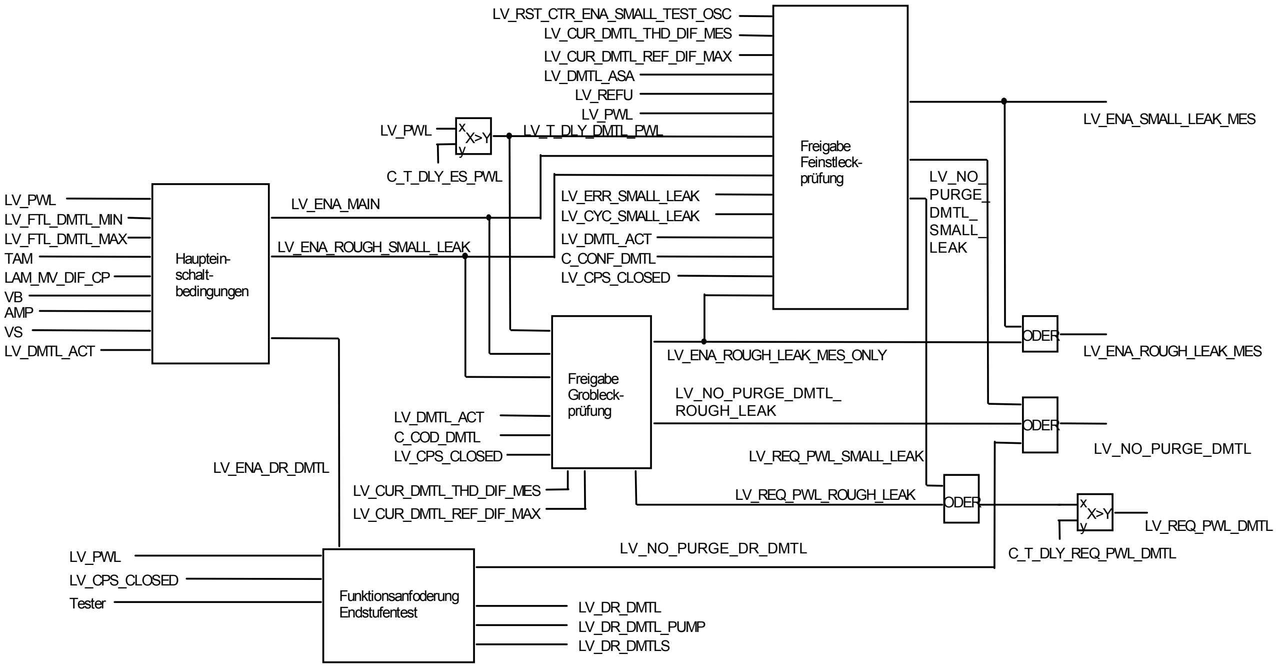
The enabling of the tank leakage diagnostic function is to be represented here in terms of an overview. The individual bits comprised are explained subsequently in the following chapters. Some remarkable points are to be mentioned in this context.

Formula section:

```
If          ECU_STATE != PWL   →  ECU_STATE = PWL
      Then  T_DLY_ES_PWL = C_T_DLY_ES_PWL
Endif


If          ECU_STATE = PWL          AND
            T_DLY_ES_PWL > 0
      Then  T_DLY_ES_PWL = T_DLY_ES_PWL – 100 ms
            LV_T_DLY_DMTL_PWL = 0
      Else  LV_T_DLY_DMTL_PWL = 1
Endif


If          LV_REQ_PWL_ROUGH_LEAK = 1          OR
            LV_REQ_PWL_SMALL_LEAK = 1
      Then  LV_REQ_PWL_DMTL = 1
```

```
                T_DLY_REQ_PWL_DMTL = 0
    Endif

    If          LV_REQ_PWL_DMTL = 1                  AND
                LV_REQ_PWL_ROUGH_LEAK = 0            AND
                LV_REQ_PWL_SMALL_LEAK = 0
        Then    T_DLY_REQ_PWL_DMTL = T_DLY_REQ_PWL_DMTL + 100 ms
        Else    T_DLY_REQ_PWL_DMTL = 0
    Endif

    If          T_DLY_REQ_PWL_DMTL > C_T_DLY_REQ_PWL_DMTL
        Then    LV_REQ_PWL_DMTL = 0
    Endif

    If          LV_ENA_SMALL_LEAK_MES = 1                    OR
                LV_ENA_ROUGH_LEAK_MES_ONLY = 1
        Then    LV_ENA_ROUGH_LEAK_MES = 1
        Else    LV_ENA_ROUGH_LEAK_MES = 0
    Endif

    If          LV_NO_PURGE_DMTL_ROUGH_LEAK = 1          OR
                LV_NO_PURGE_DMTL_SMALL_LEAK = 1                  OR
                LV_NO_PURGE_DR_DMTL = 1
        Then    LV_NO_PURGE_DMTL = 1
        Else    LV_NO_PURGE_DMTL = 0
    Endif
```

Signal flow diagram:



29.5.6 Main switch-on conditions

Application conditions:

Initialisation: LV_ENA_MAIN_DMTL = 0
LV_ENA_LEAK_DMTL = 0
LV_ENA_MAIN_DMTL_FUC = 0

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Formula section:

```
If      [LV_ERR_MEM_ROUGH_LEAK = 1          AND
         LV_ERR_CFM_ROUGH_LEAK = 1]    Error (ROUGH_LEAK) confirmed in FMY
OR
        [LV_ERR_MEM_SMALL_LEAK = 1          AND
         LV_ERR_CFM_SMALL_LEAK = 1]    Error (SMALL_LEAK) confirmed in FMY
Then     LV_ENA_LEAK_DMTL = 0
Else     If       LV_ERR_EL_CPS = 0                   AND
                  LV_ERR_DIAGCPS = 0                  AND
                  LV_ERR_DMTLS = 0                    AND
                  LV_ERR_DMTL_PUMP = 0                AND
```

```
                         LV_ERR_SPI_MPS = 0
             Then   LV_ENA_LEAK_DMTL = 1
             Else   LV_ENA_LEAK_DMTL = 0
             Endif
  Endif

  If         ECU_STATE = PWL                                   AND
             LV_ES = 1                                         AND
             [AMP_MES > C_AMP_DMTL_MAX          OR
             LV_ERR_AMP = 1]                                   AND
             [LV_ERR_TAM = 1               OR
             C_TAM_DMTL_MIN < TAM <= C_TAM_DMTL_MAX]           AND
             [TCO_ST >= C_TCO_ST_DMTL_MIN       OR
             LV_ERR_TCO = 1 in case of determination TCO_ST]   AND
             CL_MMV_NORM_PURGE_END < C_CL_DMTL_MAX             AND
             [[LV_FTL_DMTL_MIN = 0              AND
             LV_FTL_DMTL_VAL = 1]                    OR
             LV_ERR_FTL = 1                          OR
             LV_ERR_FTL_PLAUS = 1]                             AND
             LV_DMTL_ASA = 0                                   AND
             [VS < C_VS_DMTL_MAX                OR
             LV_ERR_VS = 1]                                    AND
             STATE_EFP_CRASH_CAN ≠ 2]                               AND
             [C_VB_DMTL_MIN < VB <= C_VB_DMTL_MAX         OR
             C_VB_DMTL_MIN > VB >= C_VB_DMTL_MAX for maximum
             C_T_VB_DLY_DMTL]                                  AND
             LV_ENA_LEAK_DMTL = 1
       Then  LV_ENA_MAIN_DMTL_FUC = 1
       Else  LV_ENA_MAIN_DMTL_FUC = 0
  Endif

  If         [[LV_FTL_DMTL_MAX = 0         AND
             LV_FTL_DMTL_VAL = 1]                    OR
             LV_ERR_FTL = 1                          OR
             LV_ERR_FTL_PLAUS = 1]                             AND
```

```
        T_AST >= C_T_AST_DMTL_MAX                 AND
        LV_ENA_MAIN_DMTL_FUC = 1
    Then LV_ENA_MAIN_DMTL = 1
    Else LV_ENA_MAIN_DMTL = 0
Endif
```

29.5.7 Tank level sensor

General information:

The function enabling of the tank leakage diagnostic function depends, amongst other things, on the tank being filled to between 15 % and 85 % of its capacity. The tank level information is transferred to the control unit via CAN in the INSTR2 message. If a level error is transmitted by CAN or if the CAN message is not issued, then the bit is set for a validated tank level LV_FTL_DMTL_VAL = 0.

Application conditions:

Initialisation: LV_FTL_DMTL_VAL = 0
LV_FTL_DMTL_MAX = 0
LV_FTL_DMTL_MIN = 0

Recurrence: 1 s

Activation: LV_IGK = 1

Formula section:

```
If      LV_IGK = 0 → 1
    Then T_DLY_IGK_FTL = C_T_DLY_IGK_FTL
         LV_FTL_DMTL_VAL = 0
Endif

If      LV_IGK  = 1                AND
        T_DLY_IGK_FTL >   0
    Then T_DLY_IGK_FTL = T_DLY_IGK_FTL – 1 s
Endif

If      LV_ERR_FTL = 0             AND
        T_DLY_IGK_FTL = 0
    Then LV_FTL_DMTL_VAL = 1
    Else LV_FTL_DMTL_VAL = 0
```

```
Endif

If      LV_ERR_FTL = 0
    Then LV_FTL_DMTL_MAX = LV_FTL_DMTL_MAX
         LV_FTL_DMTL_MIN = LV_FTL_DMTL_MIN
    Else LV_FTL_DMTL_MAX = 0
         LV_FTL_DMTL_MIN = 0
Endif

If      FTL > C_FTL_DMTL_MAX + C_HYS_FTL_DMTL
    Then LV_FTL_DMTL_MAX = 1
Endif

If      FTL < C_FTL_DMTL_MAX – C_HYS_FTL_DMTL
    Then LV_FTL_DMTL_MAX = 0
Endif

If      FTL < C_FTL_DMTL_MIN + C_HYS_FTL_DMTL
    Then LV_FTL_DMTL_MIN = 1
Endif

If      FTL > C_FTL_DMTL_MIN – C_HYS_FTL_DMTL
    Then LV_FTL_DMTL_MIN = 0
Endif
```

29.5.8 Tank level plausibility check

For the tank diagnostic function, it is necessary to detect a refueling process, to obtain certain tank level bits and to perform a plausibility check of the tank level signal.

29.5.8.1 Initialization of diagnostic quantities

Application conditions:

Initialisation: FCO_AV_DMTL is restored out of NVMY
FCO_DMTL_INI = 0 - FCO_AV_DMTL

Recurrence: 1 s

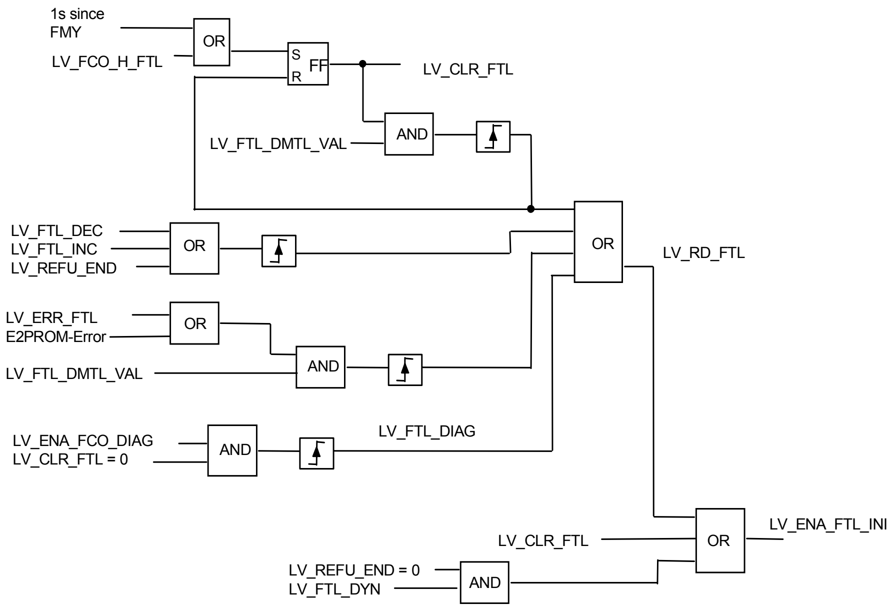
Activation: LV_IGK = 1

Deactivation: LV_IGK = 1 → 0

General information:

To perform a plausibility check of the level signal, a fuel consumption resulting from the difference between two fuel level values is compared with the fuel consumption resulting from the injection calculation.

Signal flow diagram:



In case of the initial initialization or an EEPROM error, FCO_AV_DMTL = FCO_DMTL_INI = FTL_INI = 0 is set.
In case the failure memory is deleted, FTL_INI = 0.

Formula section:

```
If      FCO_AV_DMTL > C_FTL_DIAG_MAX
  Then  LV_FCO_H_FTL = 1
Endif

If      FMY is cleared              OR
        LV_FCO_H_FTL = 1
  Then  LV_CLR_FTL = 1
Endif

If      LV_CLR_FTL = $1_{(n-1)}$    AND
        LV_FTL_DMTL_VAL = 1
  Then  LV_RD_FTL _1 = 1
        LV_CLR_FTL = $0_{(n)}$
Endif
If      LV_FTL_DEC = 1              OR
```

```
                LV_FTL_INC = 1              OR
                LV_REFU_END = 1
        Then    LV_RD_FTL _2 = 1
    Endif
    If          LV_ERR_FTL = 1              AND
                LV_FTL_DMTL_VAL = 1
        Then    LV_RD_FTL _3 = 1
    Endif
    If          LV_ENA_FCO_DIAG = 1         AND
                LV_CLR_FTL = 0
        Then    LV_RD_FTL _4 = 1
    Endif

    If          LV_RD_FTL_1 = 0 → 1                 OR
                LV_RD_FTL_2 = 0 → 1                 OR
                LV_RD_FTL_3 = 0 → 1                 OR
                LV_RD_FTL_4 = 0 → $1_{(n-1)}$       OR
                LV_CLR_FTL = 1                      OR
                [LV_REFU_END = 0            AND
                LV_FTL_DYN = 1]
        Then    LV_ENA_FTL_INI = 1
    Endif

    If          LV_ENA_FTL_INI = 1
        Then    FTL_INI = FTL
                FCO_AV_DMTL = 0
                FCO_DMTL_INI = FCO_DMTL
        Else    FTL_INI = FTL_INI
                FCO_AV_DMTL = FCO_DMTL - FCO_DMTL_INI
    Endif
```

29.5.8.2 Refueling detection case of ignition key ON and level reading

Application conditions:

Recurrence: 1 s

Activation: LV_IGK = 1

Initialization: DIF_FTL_VST = 0 in case of LV_IGK off to on

Formula section:

```
If      VS < C_VS_FTL_DIAG_MAX      UND      LV_FTL_DMTL_VAL = 1
        for a period longer than C_T_DLY_STOP_REFU
   Then LV_REFU_RLS_DIAG = 1
Endif

If      LV_FTL_DMTL_VAL = 0                        OR
        LV_ERR_VS = 1                              OR
        VS >= C_VS_FTL_DIAG_MAX
   Then LV_REFU_RLS_DIAG = 0
Endif

If      LV_REFU_RLS_DIAG = 0   AND   LV_FTL_DYN = 1
   Then LV_REFU_END = 1
   Else LV_REFU_END = 0
Endif

If      LV_ERR_VS = 1                              OR
        LV_ENA_FTL_DIAG = 0
   Then FTL_AV = FTL_VST_IN = DIF_FTL_VST = 0
Endif


If      LV_REFU_RLS_DIAG = 0 → 1
   Then FTL_VST_IN = FTL
        FCO_VST = FCO_DMTL
Endif

If      LV_REFU_RLS_DIAG = 1
   Then FTL_AV = FTL
        DIF_FTL_VST = FTL_AV - (FTL_VST_IN - (FCO_DMTL - FCO_VST))
Endif
```

Subsequently the following interrogations are processed:

```
If      LV_ERR_VS = 0                              AND
```

```
               LV_ENA_FTL_DIAG = 1                          AND
               DIF_FTL_VST > C_FTL_DIAG_MIN
         Then  LV_FTL_INC_IGK_ON = 1
         Else  LV_FTL_INC_IGK_ON = 0
Endif

If             LV_ERR_VS = 0                                AND
               LV_ENA_FTL_DIAG = 1                          AND
               DIF_FTL_VST < C_FTL_DEC_FTL_DIAG_MIN
         Then  LV_FTL_DEC_IGK_ON = 1
         Else  LV_FTL_DEC_IGK_ON = 0
Endif

If             LV_FTL_INC_IGK_ON = 1   OR        LV_FTL_DEC_IGK_ON = 1
         Then  LV_FTL_DYN = 1
         Else  LV_FTL_DYN = 0
Endif
```

In case of "delete failure memory", the following occurs:

LV_REFU_END = LV_FTL_DYN = FTL_VST_IN = FTL_AV = DIF_FTL_VST =

LV_FTL_DEC_IGK_ON = LV_FTL_INC_IGK_ON = LV_REFU_RLS_DIAG = 0

29.5.8.3 Refueling detection after restart

Application conditions:

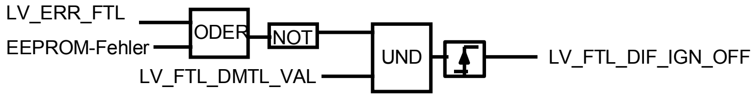
Recurrence: 1 s

Activation: LV_IGK = 1

Function description:

In the case of an EEPROM error or on initial initialization, this refueling detection must not be performed.

Signal flow diagram:



Formula section:

```
If        [LV_FTL_DMTL_VAL = 1 for minimum  one second      AND
           LV_ST_END = 1]                                       OR
          [LV_CLR_FTL = 1            AND
           LV_FTL_DIF_IGN_OFF = 1]
    Then  FTL_OLD = FTL
Endif

If        LV_FTL_DIF_IGN_OFF = 1
    Then  FTL_ST = FTL
          DIF_FTL_AST = FTL_ST - FTL_OLD
Endif

If        LV_FTL_DIF_IGN_OFF = 1
    then  If     LV_ENA_FTL_DIAG = 1                 AND
                 DIF_FTL_AST > C_FTL_DIAG_MIN
          Then   LV_FTL_INC = 1
          Else   LV_FTL_INC = 0
          Endif
          If     LV_ENA_FTL_DIAG = 1                 AND
                 DIF_FTL_AST < C_FTL_DEC_FTL_DIAG_MIN
          Then   LV_FTL_DEC = 1
          Else   LV_FTL_DEC = 0
          Endif
Endif

If        LV_REFU_RLS_DIAG = 0 → 1
Then      LV_FTL_DEC = 0
          LV_FTL_INC = 0
```

```
        DIF_FTL_AST = 0
Endif
```

In case of the initial initialization or an EEPROM error, FTL_OLD = 0 is initialized.

In case of "delete failure memory", the following occurs:

LV_FTL_IN = FTL_OLD = LV_FTL_DEC = LV_FTL_INC = FTL_ST = DIF_FTL_AST = 0

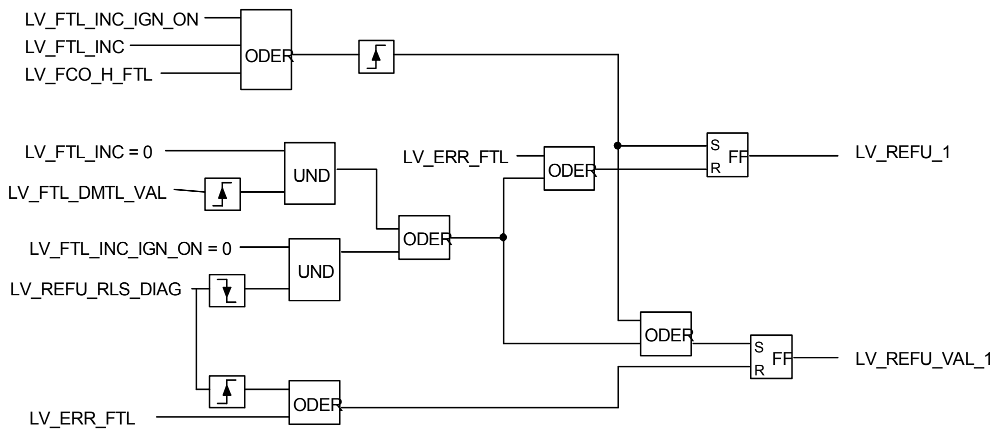
29.5.8.4 Refueling flag output

Application conditions:

Recurrence: 1 s

Activation: LV_IGK = 1

Signal flow diagram:



The output of the refueling bits LV_REFU and LV_REFU_VAL depends on the level diagnosis enable state.

The following logic is derived therefrom:

```
If          LV_ENA_FTL_DIAG = 1      AND   LV_REFU_1 = 1
    Then    LV_REFU = 1
    Else    LV_REFU = 0
Endif

If          LV_ENA_FTL_DIAG = 1      AND   LV_REFU_VAL_1 = 1
    Then    LV_REFU_VAL = 1
    Else    LV_REFU_VAL = 0
```

```
Endif
```

If no diagnosis has been enabled LV_ENA_FTL_DIAG = 0, then the refueling bits LV_REFU = 0 and LV_REFU_VAL = 0 are issued.

In case of "delete failure memory", the following occurs:
LV_REFU_1 = LV_REFU_VAL_1 = 0

29.5.8.5 Fuel level diagnosis

Application conditions:

Recurrence: 1 s

Activation: LV_IGK = 1

Formula section:

Consumption diagnosis enable:

```
If      LV_FTL_DMTL_VAL = 1                     AND
        [LV_FTL_DYN = 0          OR
        LV_REFU_END = 1]                        AND
        FCO_AV_DMTL > C_ENA_FCO_DIAG            AND
        LV_ENA_FTL_DIAG =1
    Then LV_ENA_FCO_DIAG = 1
    Else LV_ENA_FCO_DIAG = 0
Endif
```

Upper and lower threshold of the consumption diagnosis:

```
If      LV_FTL_DIAG = 1          AND    LV_ENA_FTL_DIAG = 1
    Then If     DIF_FCO_FTL = FCO_AV_DMTL - (FTL_INI - FTL) > C_FCO_DIAG_MAX
                Then   LV_FCO_DIAG_MAX = 1
                Else   LV_FCO_DIAG_MAX = 0
         Endif
         If     DIF_FCO_FTL = FCO_AV_DMTL - (FTL_INI - FTL) < C_FCO_DIAG_MIN
                Then   LV_FCO_DIAG_MIN = 1
                Else   LV_FCO_DIAG_MIN = 0
    Else LV_FCO_DIAG_MAX = 0     AND    LV_FCO_DIAG_MIN = 0
Endif
```

In case of "delete failure memory", the following occurs:

LV_FCO_DIAG_MAX = LV_FCO_DIAG_MIN = LV_ENA_FCO_DIAG = DIF_FCO_FTL = 0

29.5.8.6 Fuel level diagnosis enable

Application conditions:

Recurrence: 1 s

Activation: LV_IGK = 1

Formula section:

```
If          LV_CONF_DMTL = 1
    Then    LV_ENA_FTL_DIAG = 1
    Else    LV_ENA_FTL_DIAG = 0
Endif
```

29.5.8.7 Tank level error

Application conditions:

Initialisation: restored out of FMY

Recurrence: 1 s

Activation: LV_IGK = 1

Formula section:

```
If          LV_FTL_CAN_ERR = 1              OR
            LV_ERR_TOUT_ICL_2 = 1           OR
            LV_ERR_BN_KM_ICL = 1
    Then    LV_ERR_FTL = 1
    Else    LV_ERR_FTL = 0
Endif


If    LV_FTL_DIAG = 1
Then        If          [LV_FCO_DIAG_MAX = 1    OR
                        LV_FCO_DIAG_MIN = 1]            AND
                        LV_ERR_TOUT_ICL_2 = 0           AND
                        LV_ERR_BN_KM_ICL = 0
            Then    LV_ERR_FTL_PLAUS = 1
```

```
        Else    LV_ERR_FTL_PLAUS = 0
        Endif
    Endif
```

29.5.9 Enabling the rough leak check

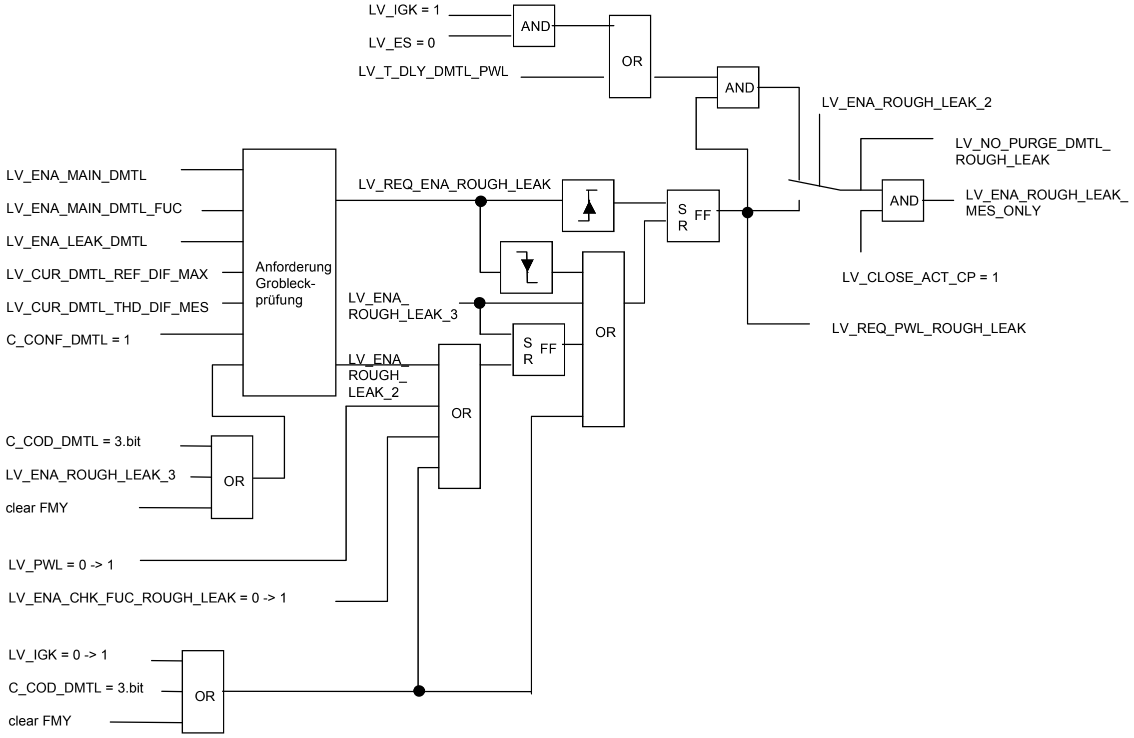
Application conditions:

Initialisation: LV_REQ_PWL_ROUGH_LEAK = 0
LV_INH_ROUGH_LEAK_MES = 0

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Formula section:

```
If      LV_REQ_ENA_ROUGH_LEAK = 0 → 1
   Then LV_REQ_PWL_ROUGH_LEAK = 1
Endif

If      LV_REQ_ENA_ROUGH_LEAK = 1 → 0        OR
        LV_ENA_ROUGH_LEAK_3 = 1              OR
        LV_INH_ROUGH_LEAK_MES = 1            OR
        LV_IGK = 0 → 1                       OR
        C_COD_DMTL = 3.bit                   OR
        clear FMY
   Then LV_REQ_PWL_ROUGH_LEAK = 0
Endif

If      LV_ENA_ROUGH_LEAK_3 = 1
```

```
        Then  LV_INH_ROUGH_LEAK_MES= 1
    Endif

    If        LV_ENA_ROUGH_LEAK_2 = 1                      OR
              LV_IGK = 0 → 1                               OR
              C_COD_DMTL = 3.bit                           OR
              clear FMY                                    OR
              ECU_STATE != PWL  → ECU_STATE = PWL                  OR
              LV_ENA_CHK_FUC_ROUGH_LEAK = 0 → 1
        Then  LV_INH_ROUGH_LEAK_MES= 0
    Endif

    If        [LV_REQ_PWL_ROUGH_LEAK = 1                   AND
              LV_ENA_ROUGH_LEAK_2 = 1]                             OR
              [LV_REQ_PWL_ROUGH_LEAK = 1                   AND
              [LV_T_DLY_DMTL_PWL = 1                 OR
              [LV_IGK = 1            AND
              LV_ES = 0]]                            AND
              LV_ENA_ROUGH_LEAK_2 = 0]
        Then  LV_NO_PURGE_DMTL_ROUGH_LEAK = 1
    Endif

        If    LV_CLOSE_ACT_CP = 1                    AND
              LV_NO_PURGE_DMTL_ROUGH_LEAK = 1
        Then  LV_ENA_ROUGH_LEAK_MES_ONLY = 1
        Else  LV_ENA_ROUGH_LEAK_MES_ONLY = 0
    Endif

    If        LV_DMTL_ACT = 1 → 0
        Then  LV_ENA_ROUGH_LEAK_3 = 1
        Else  LV_ENA_ROUGH_LEAK_3 = 0
    Endif
```

29.5.10 Request rough leak check

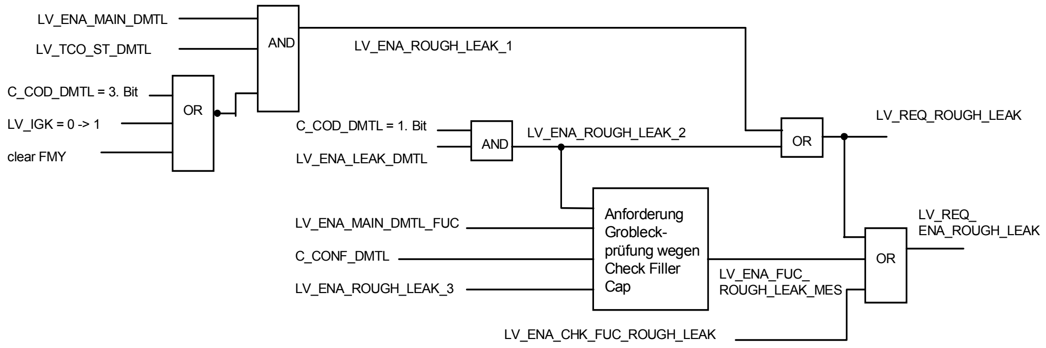
Application conditions:

Initialisation: LV_ENA_ROUGH_LEAK_1 = 0
LV_TCO_ST_DMTL = 0 in case of reset and LV_IGK = 0 → 1

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Formula section:

```
If      TCO_ST ≤ TAM_ST + C_TCO_ST_DMTL_COLD
Then    LV_TCO_ST_DMTL = 1
Else    LV_TCO_ST_DMTL = 0
Endif

If      LV_ENA_MAIN_DMTL = 1                AND
        LV_TCO_ST_DMTL = 1                  AND
        NOT(C_COD_DMTL = 3. bit OR
        LV_IGK = 0 →1           OR
        no clear FMY)
   Then LV_ENA_ROUGH_LEAK_1 = 1
   Else LV_ENA_ROUGH_LEAK_1 = 0
Endif

If      LV_ENA_LEAK_DMTL = 1          AND
```

```
          C_COD_DMTL = 1. bit
    Then  LV_ENA_ROUGH_LEAK_2 = 1
    Else  LV_ENA_ROUGH_LEAK_2 = 0
Endif


          LV_ENA_ROUGH_LEAK_1 = 1                       OR
          LV_ENA_ROUGH_LEAK_2 = 1
    Then  LV_REQ_ROUGH_LEAK = 1
    Else  LV_REQ_ROUGH_LEAK = 0
Endif


If        LV_REQ_ROUGH_LEAK = 1             OR
          LV_ENA_FUC_ROUGH_LEAK_MES = 1     OR
          LV_ENA_CHK_FUC_ROUGH_LEAK = 1
    Then  LV_REQ_ENA_ROUGH_LEAK = 1
    Else  LV_REQ_ENA_ROUGH_LEAK = 0
Endif
```

29.5.11 Rough leak check request due to Check Filler Cap

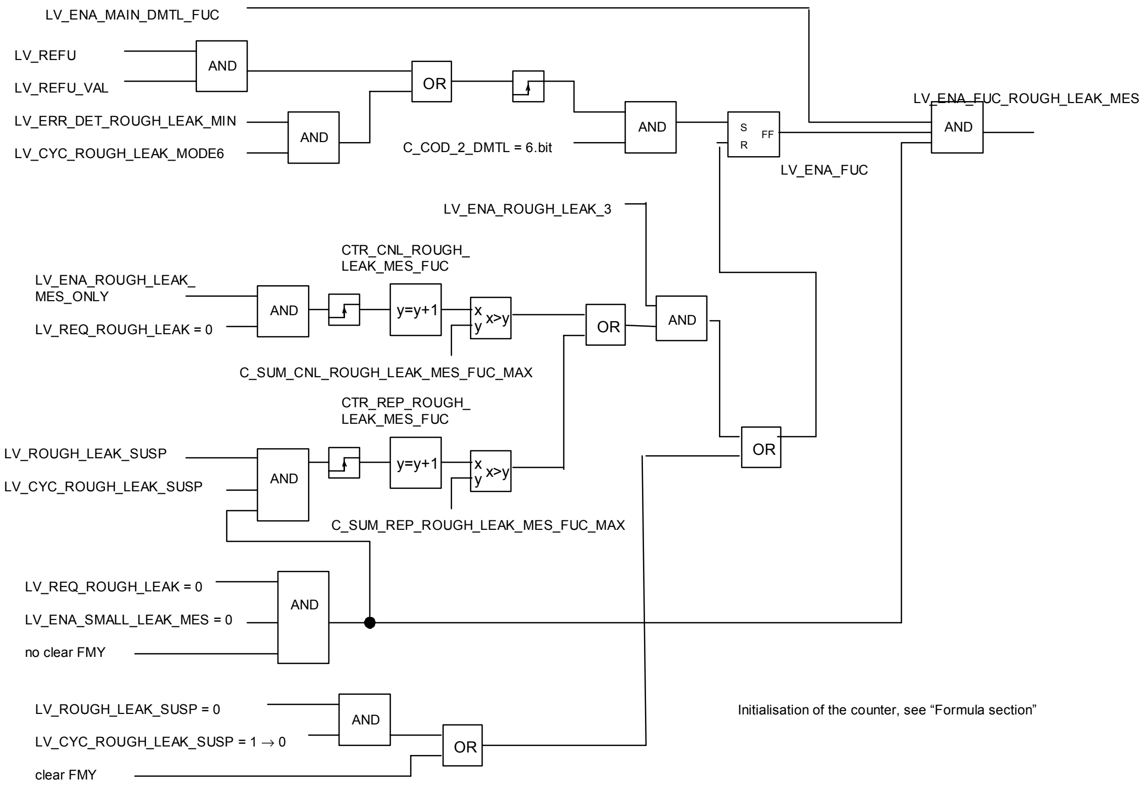
Application conditions:

Initialisation: CTR_CNL_ROUGH_LEAK_MES_FUC is restored out of the NVMY
CTR_REP_ROUGH_LEAK_MES_FUC is restored out of the NVMY
LV_ENA_FUC is restored out of the NVMY

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signalflußdiagramm :



Initialisation of the counter, see “Formula section”

Formula section:

```
If      [LV_REFU = 1                          AND
        LV_REFU_VAL = 1]                              OR
        [LV_ERR_DET_ROUGH_LEAK_MIN = 1        AND
        LV_CYC_ROUGH_LEAK_MODE6 = 1]
   Then LV_ENA_FUC_1 = 1
   Else LV_ENA_FUC_1 = 0
Endif

If      LV_ENA_FUC_1 = 0 → 1          AND
        C_COD_2_DMTL = 6.bit
   Then LV_ENA_FUC = 1
Endif

If      LV_ENA_FUC = 1                AND
        LV_ENA_MAIN_DMTL_FUC = 1      AND
```

```
                LV_REQ_ROUGH_LEAK = 0          AND
                LV_ENA_SMALL_LEAK_MES = 0      AND
                no clear FMY
        Then    LV_ENA_FUC_ROUGH_LEAK_MES = 1
        Else    LV_ENA_FUC_ROUGH_LEAK_MES = 0
    Endif

    If          LV_ENA_ROUGH_LEAK_MES_ONLY = 1          AND
                LV_REQ_ROUGH_LEAK = 0
        Then    LV_CTR_CNL_ROUGH_LEAK_MES_FUC = 1
        Else    LV_CTR_CNL_ROUGH_LEAK_MES_FUC = 0
    Endif

    If          LV_CTR_CNL_ROUGH_LEAK_MES_FUC = 0 → 1
        Then    CTR_CNL_ROUGH_LEAK_MES_FUC =
                CTR_CNL_ROUGH_LEAK_MES_FUC +1
        Else    CTR_CNL_ROUGH_LEAK_MES_FUC = CTR_CNL_ROUGH_LEAK_MES_FUC
    Endif

    If          LV_CYC_ROUGH_LEAK_SUSP = 1 → 0          OR
                LV_ENA_FUC_1 = 0 → 1                    OR
                clear FMY
        Then    CTR_CNL_ROUGH_LEAK_MES_FUC = 0
    Endif

    If          LV_ROUGH_LEAK_SUSP = 1              AND
                LV_CYC_ROUGH_LEAK_SUSP = 1          AND
                LV_REQ_ROUGH_LEAK = 0               AND
                LV_ENA_SMALL_LEAK_MES = 0           AND
                no clear FMY
        Then    LV_CTR_REP_ROUGH_LEAK_MES_FUC = 1
        Else    LV_CTR_REP_ROUGH_LEAK_MES_FUC = 0
    Endif

    If          LV_CTR_REP_ROUGH_LEAK_MES_FUC = 0 → 1
```

```
        Then  CTR_REP_ROUGH_LEAK_MES_FUC =
              CTR_REP_ROUGH_LEAK_MES_FUC +1
        Else  CTR_REP_ROUGH_LEAK_MES_FUC = CTR_REP_ROUGH_LEAK_MES_FUC
    Endif


    If        LV_ENA_FUC_1 = 0 → 1                      OR
              clear FMY
        Then  CTR_REP_ROUGH_LEAK_MES_FUC = 0
    Endif


    If        [[CTR_CNL_ROUGH_LEAK_MES_FUC >
              C_SUM_CNL_LEAK_MES_FUC_MAX                OR
              CTR_REP_ROUGH_LEAK_MES_FUC >
              C_SUM_REP_LEAK_MES_FUC_MAX]                    AND
              LV_ENA_ROUGH_LEAK_3 = 1]                            OR
              [[LV_ROUGH_LEAK_SUSP = 0             AND
              LV_CYC_ROUGH_LEAK_SUSP = 1 → 0]           OR
              clear FMY]
        Then  LV_ENA_FUC = 0
    Endif
```

29.5.12 Rough leak check request due to Check Filler Cap during trip

Application conditions:

Initialisation: CTR_ENA_DMTL_CHK_FUC = 0 at LV_IGK = 0 → 1 and clr fmy
LV_ENA_CHK_FUC_ROUGH_LEAK = 0 at LV_IGK = 0 → 1 and clr fmy
LV_T_ENA_DMTL_CHK_FUC = 0 at LV_IGK = 0 → 1 and clr fmy
LV_CTR_ENA_DMTL_CHK_FUC = 0 at LV_IGK = 0 → 1 and clr fmy
T_ENA_DMTL_CHK_FUC = 0 at LV_IGK = 0 → 1 and clr fmy
LV_T_ENA_DMTL_CHK_FUC = 0 at LV_IGK = 0 → 1 and clr fmy
LV_T_ENA_AST_DMTL_CHK_FUC = 0 at LV_IGK = 0 → 1 and clr fmy

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Formula section:

```
If      LV_ES = 0                                              and
        [LV_ERR_TAM = 1                         or
        TAM > C_TAM_DMTL_CHK_FUC_MIN]                          and
        STATE_EFP_CRASH_CAN ≠ 2                                and
        C_COD_2_DMTL = 0. bit                                  and
        C_VB_DMTL_CHK_FUC_MIN ≤ VB ≤ C_VB_DMTL_CHK_FUC_MAX     and
        CTR_ENA_DMTL_CHK_FUC < C_CTR_ENA_DMTL_CHK_FUC          and
        LV_ENA_LEAK_DMTL = 1                                   and
        LV_DMTL_ASA = 0                                        and
        LV_ENA_CHK_FUC_ROUGH_LEAK_MES = 1                      and
        LV_T_ENA_DMTL_CHK_FUC = 1
Then    LV_ENA_CHK_FUC_ROUGH_LEAK = 1
Else    LV_ENA_CHK_FUC_ROUGH_LEAK = 0
Endif

If      LV_DMTL_ACT = 1                         and
        LV_ENA_CHK_FUC_ROUGH_LEAK = 1
Then    LV_CTR_ENA_DMTL_CHK_FUC = 1
Else    LV_CTR_ENA_DMTL_CHK_FUC = 0
Endif

If      LV_CTR_ENA_DMTL_CHK_FUC = 0 → 1
Then    CTR_ENA_DMTL_CHK_FUC ++
Endif

If      LV_DMTL_ACT = 1 → 0     or
        ECU_STATE = PWL
   Then T_ENA_DMTL_CHK_FUC = 0
        LV_T_ENA_AST_DMTL_CHK_FUC = 0
        LV_ENA_CHK_FUC_ROUGH_LEAK = 0
Endif

If      LV_ENA_CHK_FUC_ROUGH_LEAK = 0           and
        LV_IGK = 1                              and
```

```
            LV_ES = 0
            Then  T_ENA_DMTL_CHK_FUC ++
Endif

If          T_AST = C_T_AST_DMTL_CHK_FUC
Then        LV_T_ENA_AST_DMTL_CHK_FUC = 1
Endif

If          T_ENA_DMTL_CHK_FUC ≥ C_T_ENA_DMTL_CHK_FUC        or
            LV_T_ENA_AST_DMTL_CHK_FUC = 1
Then        LV_T_ENA_DMTL_CHK_FUC = 1
Else        LV_T_ENA_DMTL_CHK_FUC = 0
Endif
```

29.5.13 Enable of miniature leak check

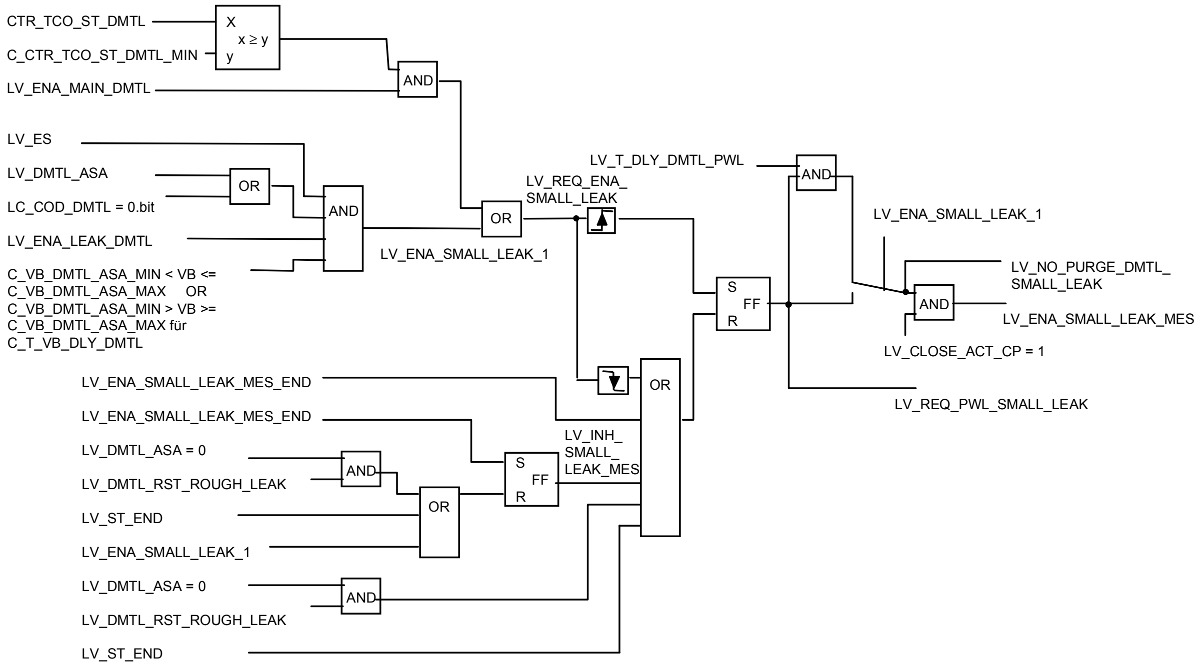
Application conditions:

Initialisation: LV_ENA_SMALL_LEAK_MES is restored out of the NVMY
CTR_TCO_ST_DMTL is restored out of the NVMY

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Formula section:

```
If      LV_TCO_ST_DMTL = 0 → 1
Then    CTR_TCO_ST_DMTL ++
Endif
If      LV_CYC_SMALL_LEAK_MODE6 = 1         OR
        LV_DMTL_RST_ROUGH_LEAK = 1
Then    CTR_TCO_ST_DMTL = 0
Endif


If      [CTR_TCO_ST_DMTL ≥ C_CTR_TCO_ST_DMTL_MIN   AND
        LV_ENA_MAIN_DMTL]                                    OR
        LV_ENA_SMALL_LEAK_1 = 1
   Then LV_REQ_ENA_SMALL_LEAK = 1
   Else LV_REQ_ENA_SMALL_LEAK = 0
Endif


If      LV_ES = 1                                    AND
```

```
            [LV_DMTL_ASA = 1            OR
            C_COD_DMTL = 0.bit]                                 AND
            LV_ENA_LEAK_DMTL                                    AND
            [C_VB_DMTL_ASA_MIN < VB ≤ C_VB_DMTL_ASA_MAX   OR
            C_VB_DMTL_ASA_MIN > VB >= C_VB_DMTL_ASA_MAX for maximum
            C_T_VB_DLY_DMTL]
    Then    LV_ENA_SMALL_LEAK_1 = 1
    Else    LV_ENA_SMALL_LEAK_1 = 0
Endif

If          [LV_DMTL_ASA = 0                    AND
            LV_DMTL_RST_ROUGH_LEAK = 1]                 OR
            LV_ST_END = 1                               OR
            LV_ENA_SMALL_LEAK_1
    Then    LV_INH_SMALL_LEAK_MES = 0
Endif

If          LV_DMTL_ACT = 0                     AND
            LV_ENA_SMALL_LEAK_MES$_{(n-1)}$ = 1
    Then    LV_ENA_SMALL_LEAK_MES_END = 1
            LV_INH_SMALL_LEAK_MES = 1
Endif

If          C_COD_DMTL = 2.bit                  OR
            clear FMY
    Then    LV_DMTL_RST_ROUGH_LEAK = 1
    Else    LV_DMTL_RST_ROUGH_LEAK = 0
Endif

If          LV_REQ_ENA_SMALL_LEAK = 0 → 1
    Then    LV_REQ_PWL_SMALL_LEAK = 1
Endif

If          LV_REQ_ENA_SMALL_LEAK = 1 → 0           OR
            LV_ENA_SMALL_LEAK_MES_END = 1           OR
```

```
            LV_INH_SMALL_LEAK_MES = 1               OR
            LV_ST_END = 0 → 1                       OR
            [LV_DMTL_ASA = 0                    AND
            LV_DMTL_RST_ROUGH_LEAK = 1]
      Then  LV_REQ_PWL_SMALL_LEAK = 0
  Endif

  If        [LV_REQ_PWL_SMALL_LEAK = 1          AND
            LV_ENA_SMALL_LEAK_1 = 1]                      OR
            [LV_REQ_PWL_SMALL_LEAK = 1          AND
            LV_T_DLY_DMTL_PWL = 1               AND
            LV_ENA_SMALL_LEAK_1 = 0]
      Then  LV_NO_PURGE_DMTL_SMALL_LEAK = 1
  Endif

  If        LV_CLOSE_ACT_CP = 1                 AND
            LV_NO_PURGE_DMTL_SMALL_LEAK = 1
      Then  LV_ENA_SMALL_LEAK_MES = 1
      Else  LV_ENA_SMALL_LEAK_MES = 0
  Endif

  If        C_VB_DMTL_ASA_MIN > VB > C_VB_DMTL_ASA_MAX              OR
            C_VB_DMTL_ASA_MIN < VB < C_VB_DMTL_ASA_MAX for maximum
            C_T_VB_DLY_DMTL
      Then  LV_VB_RANGE_DMTL = 1
      Else  LV_VB_RANGE_DMTL = 0
  Endif
```

29.5.14 Miniature leak check request due to abnormal termination through current fluctuations

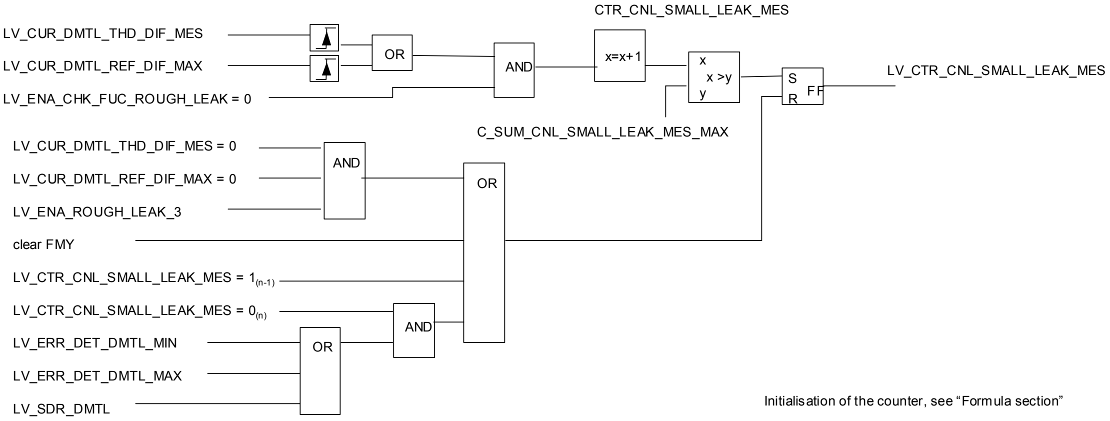
Application conditions:

Initialisation: LV_CTR_CNL_SMALL_LEAK_MES is restored out of the NVMY
CTR_CNL_SMALL_LEAK_MES is restored out of the NVMY

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Initialisation of the counter, see “Formula section”

Formula section:

```
If      [LV_CUR_DMTL_THD_DIF_MES = 0 → 1          OR
        LV_CUR_DMTL_REF_DIF_MAX = 0 → 1]              AND
        LV_ENA_CHK_FUC_ROUGH_LEAK = 0
   Then CTR_CNL_SMALL_LEAK_MES = CTR_CNL_SMALL_LEAK_MES + 1
   Else CTR_CNL_SMALL_LEAK_MES = CTR_CNL_SMALL_LEAK_MES
Endif

If      [LV_CUR_DMTL_THD_DIF_MES = 0              AND
        LV_CUR_DMTL_REF_DIF_MAX = 0               AND
        LV_ENA_ROUGH_LEAK_3 = 1]                      OR
        clear FMY                                     OR
        LV_CTR_CNL_SMALL_LEAK_MES = $1_{(n-1)}$
   Then CTR_CNL_SMALL_LEAK_MES = 0
```

```
    Else  CTR_CNL_SMALL_LEAK_MES = CTR_CNL_SMALL_LEAK_MES
Endif


If        CTR_CNL_SMALL_LEAK_MES > C_SUM_CNL_SMALL_LEAK_MES_MAX
    Then  LV_CTR_CNL_SMALL_LEAK_MES = 1
Endif


If        [LV_CUR_DMTL_THD_DIF_MES = 0              AND
          LV_CUR_DMTL_REF_DIF_MAX = 0               AND
          LV_ENA_ROUGH_LEAK_3 = 1]                       OR
          clear FMY                                      OR
          LV_CTR_CNL_SMALL_LEAK_MES = $1_{(n-1)}$        OR
          [[LV_ERR_DET_DMTL_MIN = 1     OR
          LV_ERR_DET_DMTL_MAX  = 1      OR
          LV_SDR_DMTL = 1]                     AND
          LV_CTR_CNL_SMALL_LEAK_MES = $0_{(n)}$ ]
    Then  LV_CTR_CNL_SMALL_LEAK_MES = 0
Endif


If        LV_ENA_CHK_FUC_ROUGH_LEAK = 1             OR
          LV_ENA_FUC_ROUGH_LEAK_MES = 1             OR
          LV_DMTL_STOP = 1
    Then  LV_INH_CHK_DMTL_VLD = 1
Endif
If        LV_ENA_ROUGH_LEAK_3 = 1 → 0               OR
          ECU_STATE != PWL → ECU_STATE = PWL        OR
          LV_IGK = 0 → 1
Then      LV_INH_CHK_DMTL_VLD = 0
Endif
If        (LV_INH_CHK_DMTL_VLD = 0      AND
          LV_ENA_ROUGH_LEAK_3 = 1)             OR
          LV_CTR_CNL_SMALL_LEAK_MES = 1
    Then  LV_CHK_DMTL_VLD = 1
    Else  LV_CHK_DMTL_VLD = 0
Endif
```

29.5.15 Function request "output stage test"

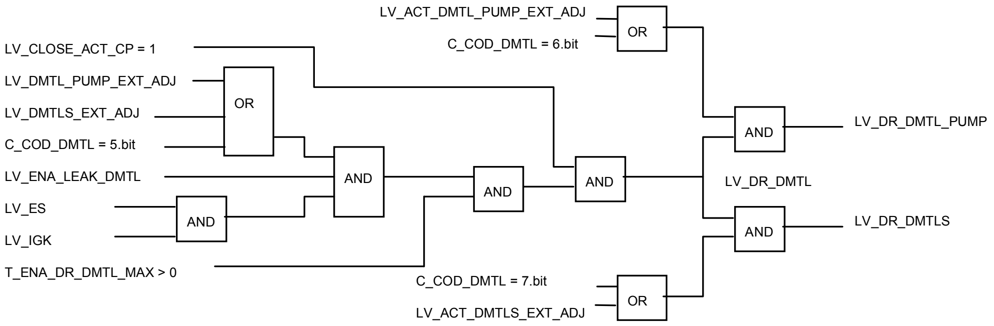
Application conditions:

Initialisation: LV_DR_DMTL_PUMP = 0
LV_DR_DMTLS = 0

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Formula section:

```
If      [LV_DMTL_PUMP_EXT_ADJ = 1    OR
        LV_DMTLS_EXT_ADJ = 1         OR
        C_COD_DMTL = 5.bit]                  AND
        LV_ENA_LEAK_DMTL = 1                 AND
        [LV_ES = 1              AND
        LV_IGK = 1]
   Then T_ENA_DR_DMTL_MAX = T_ENA_DR_DMTL_MAX + 100 ms
        LV_NO_PURGE_DR_DMTL = 1
Endif


If      [LV_DMTL_PUMP_EXT_ADJ = 1    OR
        LV_DMTLS_EXT_ADJ = 1         OR
```

```
            C_COD_DMTL = 5.bit]                       AND
            LV_ENA_LEAK_DMTL = 1                      AND
            [LV_ES = 1              AND
            LV_IGK = 1]                               AND
            T_ENA_DR_DMTL_MAX < C_T_ENA_DR_DMTL_MAX   AND
            LV_CLOSE_ACT_CP = 1
    Then    LV_DR_DMTL = 1
    Else    LV_DR_DMTL = 0
Endif

If          LV_DR_DMTL = 1                        AND
            [LV_ACT_DMTL_PUMP_EXT_ADJ = 1    OR
            C_COD_DMTL = 6.bit]
    Then    LV_DR_DMTL_PUMP = 1
    Else    LV_DR_DMTL_PUMP = 0
Endif

If          LV_DR_DMTL = 1                        AND
            [LV_ACT_DMTLS_EXT_ADJ = 1    OR
            C_COD_DMTL = 7.bit]
    Then    LV_DR_DMTLS = 1
    Else    LV_DR_DMTLS = 0
Endif
```

29.5.16 Fuel cap and refueling detection

Application conditions:

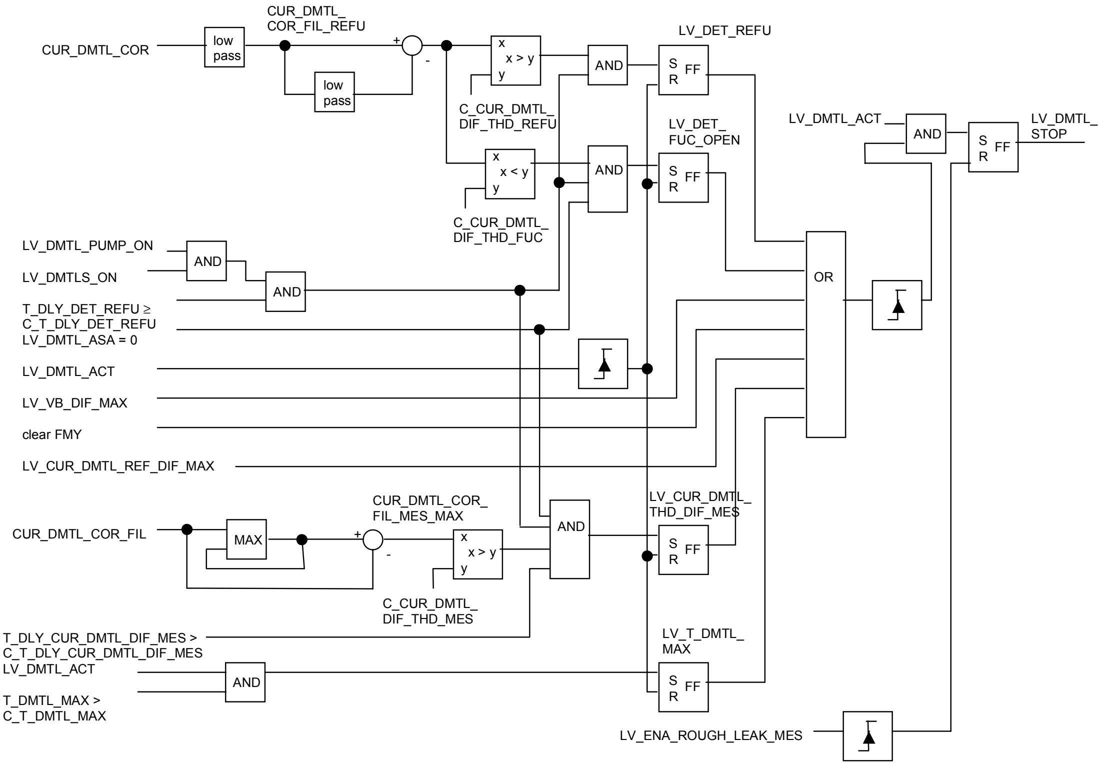
Initialisation: LV_DMTL_STOP is restored out of the NVMY
LV_DET_FUC_OPEN is restored out of the NVMY
LV_DET_REFU is restored out of the NVMY
LV_CUR_DMTL_THD_DIF_MES is restored out of the NVMY
LV_T_DMTL_MAX is restored out of the NVMY

Initialisation at
clear FMY: LV_DMTL_STOP = 1

Recurrence: 10 ms for calculation of the currents
100 ms for the use of the currents

Activation: LV_CONF_DMTL = 1

Signal flow diagram:



Formula section:

```
If      LV_DMTL_PUMP_ON = 1    AND
        LV_DMTLS_ON = 1
   Then T_DLY_DET_REFU = T_DLY_DET_REFU + 100 ms
   Else T_DLY_DET_REFU = 0
Endif

If      LV_DMTL_ACT = 1
   Then CUR_DMTL_COR_FIL_REFU$_{(n)}$ =   CUR_DMTL_COR_FIL_REFU$_{(n-1)}$ +
              (CUR_DMTL_COR$_{(n)}$ - CUR_DMTL_COR_FIL_REFU$_{(n-1)}$) *
                       C_CRLC_CUR_DMTL_FIL_LIM
Endif

If      LV_DMTL_PUMP_ON = 1    AND
        LV_DMTLS_ON = 1        AND
        T_DLY_DET_REFU < C_T_DLY_DET_REFU
Then    CUR_DMTL_COR_FIL_REFU_FIL$_{(n)}$ =
        CUR_DMTL_COR_FIL_REFU_FIL$_{(n-1)}$ +
```

```
        (CUR_DMTL_COR_FIL_REFU$_{(n)}$ - CUR_DMTL_COR_FIL_REFU_FIL$_{(n-1)}$) *
                    C_CRLC_CUR_DMTL_FIL_LIM

        T_DLY_VB_DMTL = T_DLY_VB_DMTL – 100 ms

Endif

If      LV_DMTL_PUMP_ON = 1    AND
        LV_DMTLS_ON = 1        AND
        T_DLY_DET_REFU ≥ C_T_DLY_DET_REFU
Then    CUR_DMTL_COR_FIL_REFU_FIL$_{(n)}$ =
        CUR_DMTL_COR_FIL_REFU_FIL$_{(n-1)}$ +
        (CUR_DMTL_COR_FIL_REFU$_{(n)}$ - CUR_DMTL_COR_FIL_REFU_FIL$_{(n-1)}$) *
                    C_CRLC_CUR_DMTL_FIL_REFU
Endif


CUR_DMTL_COR_FIL_REFU_HPF =    CUR_DMTL_COR_FIL_REFU -
                               CUR_DMTL_COR_FIL_REFU_FIL


If      CUR_DMTL_COR_FIL_REFU_HPF >
        C_CUR_DMTL_DIF_THD_REFU              AND
        T_DLY_DET_REFU ≥ C_T_DLY_DET_REFU
    Then LV_DET_REFU = 1
Endif

If      LV_DMTL_ACT = 0 → 1
    Then LV_DET_REFU = 0
        LV_DET_FUC_OPEN = 0
        LV_CUR_DMTL_THD_DIF_MES = 0
        T_DLY_CUR_DMTL_DIF_MES = 0
        LV_T_DMTL_MAX= 0
Endif

If      CUR_DMTL_COR_FIL_REFU_HPF <
        C_CUR_DMTL_DIF_THD_FUC                         AND
        T_DLY_DET_REFU ≥ C_T_DLY_DET_REFU              AND
        LV_DMTL_ASA = 0
    Then LV_DET_FUC_OPEN = 1
Endif


If      T_DLY_DET_REFU ≥ C_T_DLY_DET_REFU
    Then CUR_DMTL_COR_FIL_MES_MAX =
```

```
                MAX (CUR_DMTL_COR_FIL$_{(n)}$, CUR_DMTL_COR_FIL$_{(n-1)}$) –
                CUR_DMTL_COR_FIL$_{(n)}$
Endif

If              CUR_DMTL_COR_FIL_MES_MAX >
                C_CUR_DMTL_DIF_THD_MES                      AND
                T_DLY_DET_REFU ≥ C_T_DLY_DET_REFU           AND
                LV_DMTL_ASA = 0
    Then        T_DLY_CUR_DMTL_DIF_MES = T_DLY_CUR_DMTL_DIF_MES + 100 ms
    Else        T_DLY_CUR_DMTL_DIF_MES = 0
Endif

If              T_DLY_CUR_DMTL_DIF_MES > C_T_DLY_CUR_DMTL_DIF_MES
    Then        LV_CUR_DMTL_THD_DIF_MES = 1
Endif


If              LV_DMTL_ACT = 1
    Then        T_DMTL_MAX = T_DMTL_MAX  + 100 ms
    Else        T_DMTL_MAX = 0
Endif


If              T_DMTL_MAX > C_T_DMTL_MAX
    Then        LV_T_DMTL_MAX = 1
Endif


If              [LV_DET_REFU = 1                        OR
                LV_DET_FUC_OPEN = 1                     OR
                LV_VB_DIF_MAX = 1                       OR
                LV_CUR_DMTL_REF_DIF_MAX = 1             OR
                LV_CUR_DMTL_THD_DIF_MES = 1             OR
                LV_T_DMTL_MAX= 1]                           AND
                LV_DMTL_ACT = 1
    Then        LV_DMTL_STOP = 1
Endif


If              LV_ENA_ROUGH_LEAK_MES = 0 → 1
    Then        LV_DMTL_STOP = 0
```

Endif

29.5.17 Generating rough leak error from suspected rough leak error

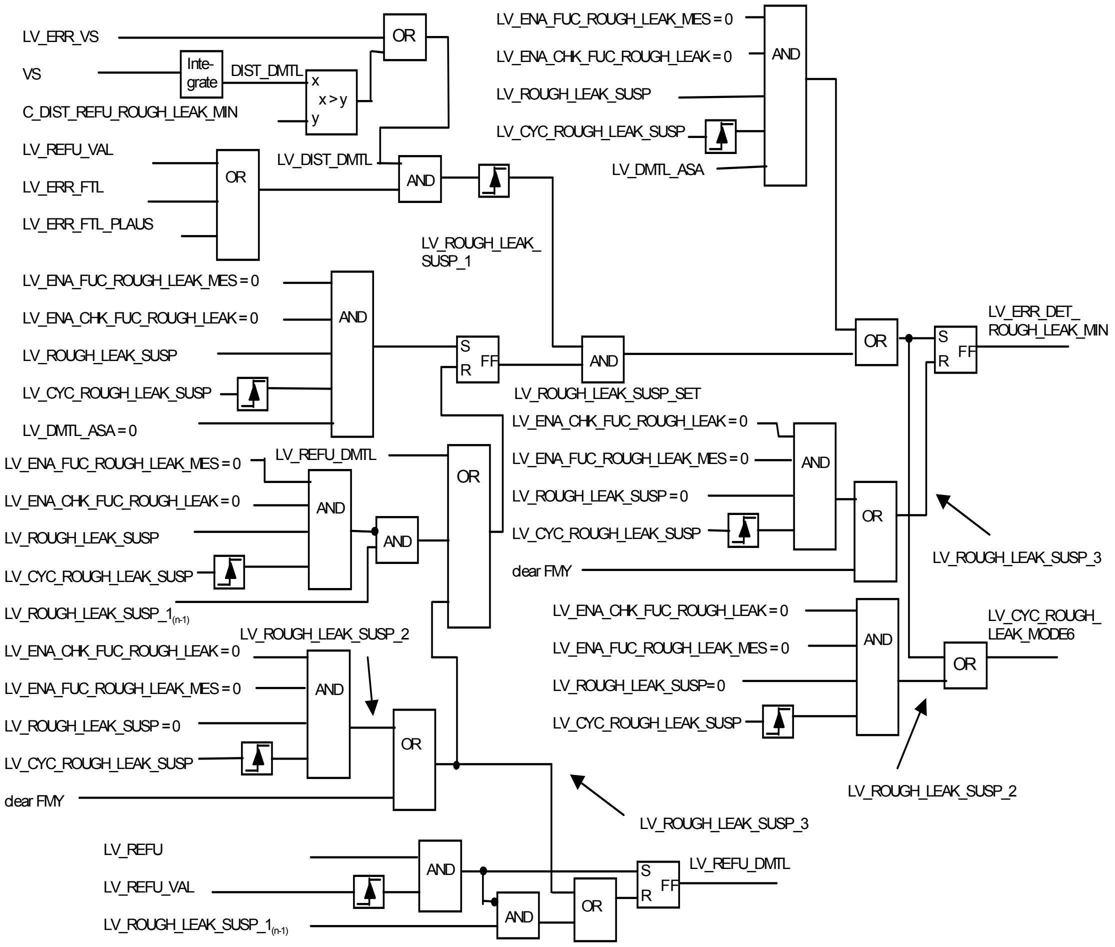
Application conditions:

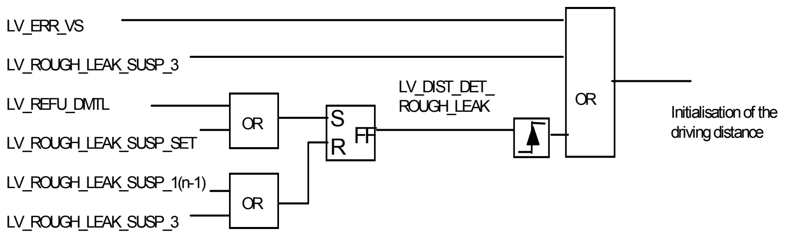
Initialisation: LV_ERR_DET_ROUGH_LEAK_MIN is restored out of the NVMY
LV_REFU_DMTL is restored out of the NVMY
LV_DIST_DET_ROUGH_LEAK is restored out of the NVMY
LV_ROUGH_LEAK_SUSP_SET is restored out of the NVMY
DIST_DMTL is restored out of the NVMY

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Signal flow diagram:





Formula section:

```
If          LV_VS_RUN = 1
     Then   $DIST\_DMTL_{(n)} = DIST\_DMTL_{(n-1)} + VS / 36$
     Else   $DIST\_DMTL_{(n)} = DIST\_DMTL_{(n-1)}$
Endif
If          DIST_DMTL > C_DIST_REFU_ROUGH_LEAK_MIN          OR
```

```
            LV_ERR_VS = 1
      Then  LV_DIST_DMTL = 1
      Else  LV_DIST_DMTL = 0
Endif
If          [LV_DIST_DMTL = 1                     AND
            [LV_REFU_VAL = 1          OR
            LV_ERR_FTL = 1            OR
            LV_ERR_FTL_PLAUS = 1]]          once          AND
            LV_ROUGH_LEAK_SUSP_1(n-1) = 0
      Then  LV_ROUGH_LEAK_SUSP_1 = 1
      Else  LV_ROUGH_LEAK_SUSP_1 = 0
Endif

If          LV_ENA_FUC_ROUGH_LEAK_MES = 0               AND
            LV_ENA_CHK_FUC_ROUGH_LEAK = 0               AND
            LV_ROUGH_LEAK_SUSP = 1                      AND
            LV_CYC_ROUGH_LEAK_SUSP = 0 → 1              AND
            LV_DMTL_ASA = 0
      Then  LV_ROUGH_LEAK_SUSP_SET = 1
            LV_CDN_DIAG_ROUGH_LEAK = 1
            ERR_SYM_ROUGH_LEAK = “MAX”
            ACTION_ERRM_StorePrevFrf (ROUGH_LEAK)
            This action stores the prestored freeze frame for the diagnosis instance ROUGH_LEAK, see
        chapter“ Environmental data, OBDII functions”
            LV_CDN_DIAG_ROUGH_LEAK = 0
Endif


If          LV_REFU_DMTL = 1                                          OR
            [Not  [LV_ENA_FUC_ROUGH_LEAK_MES = 0  AND
                  LV_ENA_CHK_FUC_ROUGH_LEAK = 0   AND
                  LV_ROUGH_LEAK_SUSP = 1          AND
                  LV_CYC_ROUGH_LEAK_SUSP = 0 → 1]          AND
            LV_ROUGH_LEAK_SUSP_1(n-1)]                                OR
            LV_ROUGH_LEAK_SUSP_3]
      Then  LV_ROUGH_LEAK_SUSP_SET = 0
Endif
```

```
If      [LV_ROUGH_LEAK_SUSP_1 = 1                AND
         LV_ROUGH_LEAK_SUSP_SET = 1]                    OR
        [LV_ENA_FUC_ROUGH_LEAK_MES = 0           AND
         LV_ENA_CHK_FUC_ROUGH_LEAK = 0           AND
         LV_ROUGH_LEAK_SUSP = 1                  AND
         LV_CYC_ROUGH_LEAK_SUSP = 0 → 1          AND
         LV_DMTL_ASA = 1]
   Then LV_ERR_DET_ROUGH_LEAK_MIN = 1
Endif

If      LV_ROUGH_LEAK_SUSP_3 = 1
   Then LV_ERR_DET_ROUGH_LEAK_MIN = 0
Endif

If      [LV_ROUGH_LEAK_SUSP_1 = 1                AND
         LV_ROUGH_LEAK_SUSP_SET = 1]                    OR
        [LV_ENA_FUC_ROUGH_LEAK_MES = 0           AND
         LV_ENA_CHK_FUC_ROUGH_LEAK = 0           AND
         LV_ROUGH_LEAK_SUSP = 1                  AND
         LV_CYC_ROUGH_LEAK_SUSP = 0 → 1          AND
         LV_DMTL_ASA = 1]                               OR
         LV_ROUGH_LEAK_SUSP_2
   Then LV_CYC_ROUGH_LEAK_MODE6 = 1
   Else LV_CYC_ROUGH_LEAK_MODE6 = 0
Endif

If      LV_ENA_FUC_ROUGH_LEAK_MES = 0            AND
        LV_ENA_CHK_FUC_ROUGH_LEAK = 0            AND
        LV_ROUGH_LEAK_SUSP = 0                   AND
        LV_CYC_ROUGH_LEAK_SUSP = 0 → 1
   Then LV_ROUGH_LEAK_SUSP_2 = 1
   Else LV_ROUGH_LEAK_SUSP_2 = 0
Endif
```

```
If      LV_ROUGH_LEAK_SUSP_2 = 1                        OR
        clear FMY
    Then LV_ROUGH_LEAK_SUSP_3 = 1
    Else LV_ROUGH_LEAK_SUSP_3 = 0
Endif

If      LV_REFU = 1                 AND
        LV_REFU_VAL = 0 → 1
    Then LV_REFU_SET = 1
         LV_REFU_DMTL = 1
    Else LV_REFU_SET = 0
Endif

If      [LV_REFU_SET = 0                      AND
        LV_ROUGH_LEAK_SUSP_1$_{(n-1)}$ = 1]             OR
        LV_ROUGH_LEAK_SUSP_3 = 1
    Then LV_REFU_DMTL = 0
Endif

If      LV_REFU_DMTL = 1                      OR
        LV_ROUGH_LEAK_SUSP_SET = 1
    Then LV_DIST_DET_ROUGH_LEAK = 1
Endif

If      LV_ROUGH_LEAK_SUSP_1$_{(n-1)}$ = 1          OR
        LV_ROUGH_LEAK_SUSP_3 = 1
    Then LV_DIST_DET_ROUGH_LEAK = 0
Endif

If      LV_ERR_VS = 1                         OR
        LV_ROUGH_LEAK_SUSP_3 = 1              OR
        LV_DIST_DET_ROUGH_LEAK = 0 → 1
    Then DIST_DMTL = 0
Endif
```

29.5.18 Combi-display activate “Check Filler Cap”

Application conditions:

Initialisation: LV_FUC_OPEN is restored out of the NVMY

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

FUNCTION DESCRIPTION:

If a rough leak suspect is detected the module has to indicate, via the CAN-signal of the combi instrument, to the driver that possibly the tank cap has not been adjusted properly.

Formula section:

```
If    LV_FUC_CAN = 0
then        if      LV_FUC_INT = 0
            then    T_DLY_FUC_CAN = C_T_DLY_FUC_CAN_ON
            else    T_DLY_FUC_CAN = T_DLY_FUC_CAN –
                    If      T_DLY_FUC_CAN = 0
                    then    LV_FUC_CAN = 1
                            T_DLY_FUC_CAN = C_T_DLY_FUC_CAN_OFF
                    endif
            endif
else        T_DLY_FUC_CAN = T_DLY_FUC_CAN –
            if      T_DLY_FUC_CAN = 0                         OR
                    (LV_ROUGH_LEAK_SUSP = 0        AND
                    LV_CYC_ROUGH_LEAK_SUSP = 0 → 1)
            then    LV_FUC_CAN = 0
                    T_DLY_FUC_CAN = C_T_DLY_FUC_CAN_ON
            endif
endif

If    LV_FUC_CAN = 1 → 0        OR
      clr FMY
then        LV_FUC_OPEN = 0
else        if      LV_CYC_ROUGH_LEAK_MODE6 = 0 → 1
```

```
            then  if    LV_ERR_ROUGH_LEAK = 0
                  then  LV_FUC_OPEN = 0
                  else  LV_FUC_OPEN = 1
                  endif
            endif
    endif
    If      LV_FUC_OPEN = 1               AND
            VS > C_VS_FUC_CAN_MIN         AND
            LV_IGK = 1
    Then    LV_FUC_INT = 1
    Else    LV_FUC_INT = 0
    Endif
```

29.5.19 Output of the errorflags

Application conditions:

Initialisation: LV_ERR_DMTLM is restored out of the NVMY
LV_ERR_DET_DMTL_MAX is restored out of the NVMY
LV_ERR_DET_DMTL_MIN is restored out of the NVMY
LV_SDR_DMTL is restored out of the NVMY
LV_ERR_SIG_DMTL is restored out of the NVMY
ERR_SYM_DMTLM 1H SYM_0 = MAX
2H SYM_1 = MIN
4H SYM_2 = SIG
8H SYM_3 = PLAUS

LV_ERR_DMTLM is reseted according NO_FIL
LV_ERR_ROUGH_LEAK is reseted according NO_FIL
LV_ERR_SMALL_LEAK is reseted according NO_FIL

In case of reset or LV_IGK = 0 -> 1:

LV_ERR_ROUGH_LEAK = 0

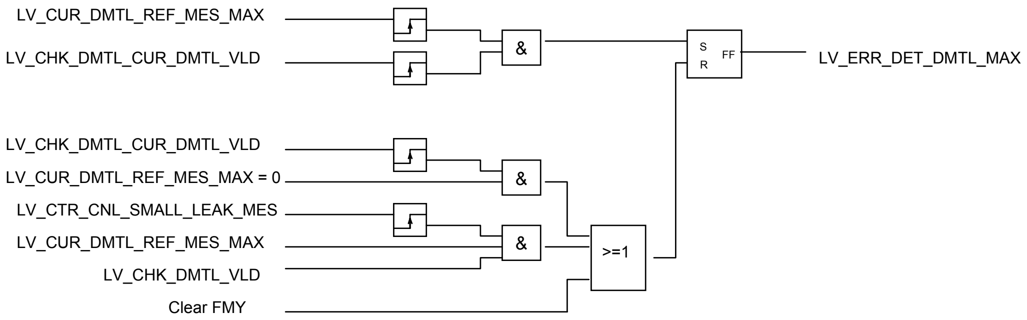
LV_ERR_SMALL_LEAK = 0

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

How maximum error DMTL is built up

Signal flow diagram:

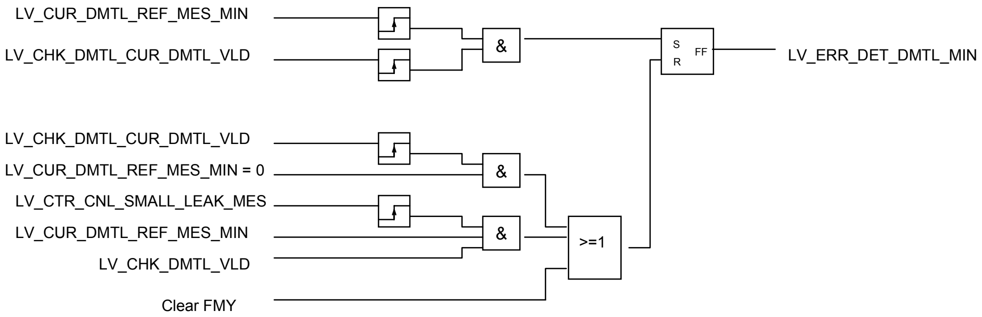


Formula section:

```
If      LV_CUR_DMTL_REF_MES_MAX = 0 → 1               AND
        LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1
   Then LV_ERR_DET_DMTL_MAX = 1
        ERR_SYM_DMTLM = “MAX”
Endif

If      [LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1             AND
        LV_CUR_DMTL_REF_MES_MAX = 0]                        OR
        [LV_CTR_CNL_SMALL_LEAK_MES = 0 → 1            AND
        LV_CUR_DMTL_REF_MES_MAX = 1                   AND
        LV_CHK_DMTL_VLD = 1]                                OR
        clear FMY
   Then LV_ERR_DET_DMTL_MAX = 0
        ERR_SYM_DMTLM = “NO_SYM”
Endif
```

How minimum error DMTL is built up

Signal flow diagram:

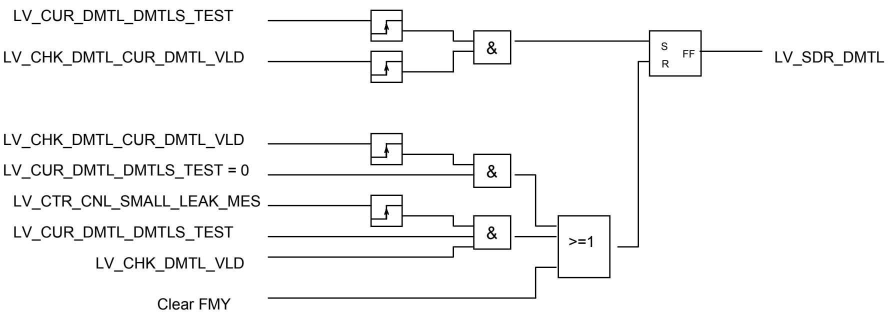


Formula section:

```
If      LV_CUR_DMTL_REF_MES_MIN = 0 → 1          AND
        LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1
  Then  LV_ERR_DET_DMTL_MIN = 1
        ERR_SYM_DMTLM = “MIN”
Endif
If      [LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1        AND
        LV_CUR_DMTL_REF_MES_MIN = 0]                   OR
        [LV_CTR_CNL_SMALL_LEAK_MES = 0 → 1       AND
        LV_CUR_DMTL_REF_MES_MIN = 1              AND
        LV_CHK_DMTL_VLD = 1]                           OR
        clear FMY
  Then  LV_ERR_DET_DMTL_MIN = 0
        ERR_SYM_DMTLM = “NO_SYM”
Endif
```

How plausibility error DMTL is built up

Signal flow diagram:

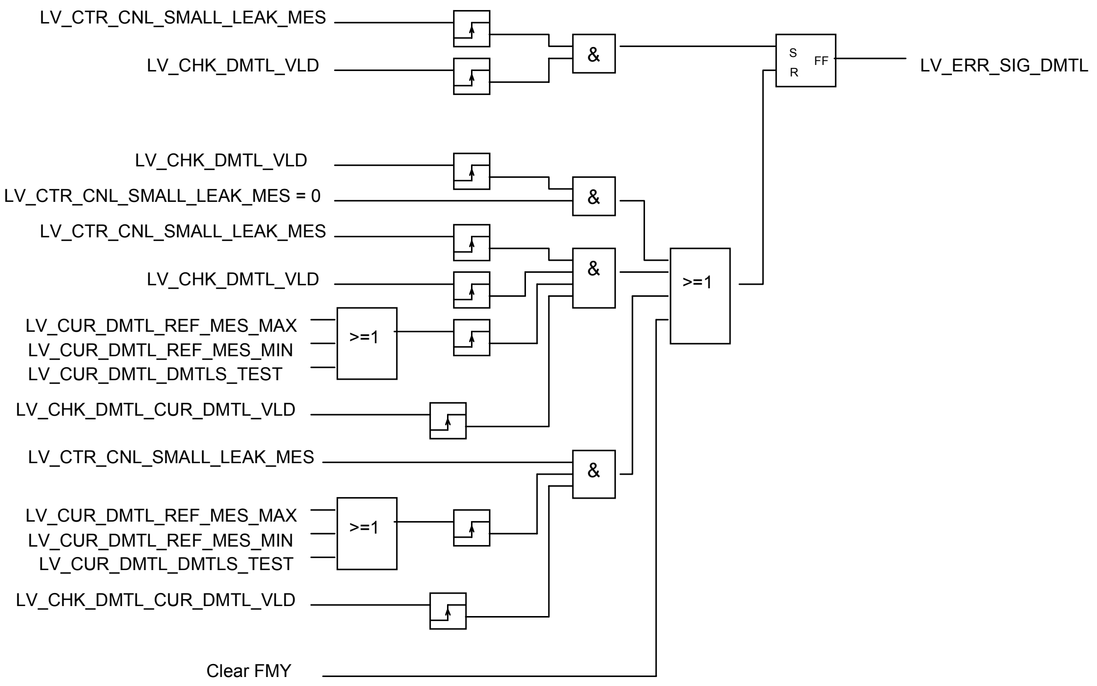


Formula section:

```
If      LV_CUR_DMTL_DMTLS_TEST = 0 → 1             AND
        LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1
   Then LV_SDR_DMTL = 1
        ERR_SYM_DMTLM = “PLAUS”
Endif
If      [LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1           AND
        LV_CUR_DMTL_DMTLS_TEST = 0]                     OR
        [LV_CTR_CNL_SMALL_LEAK_MES = 0 → 1          AND
        LV_CUR_DMTL_DMTLS_TEST = 1                  AND
        LV_CHK_DMTL_VLD = 1]                            OR
        clear FMY
   Then LV_SDR_DMTL = 0
        ERR_SYM_DMTLM = “NO_SYM”
Endif
```

How signal error DMTL is built up

Signal flow diagram:



Formula section:

If LV_CTR_CNL_SMALL_LEAK_MES = 0 → 1 **AND**

LV_CHK_DMTL_VLD = 0 → 1

Then LV_ERR_SIG_DMTL = 1

ERR_SYM_DMTLM = “SIG”

Endif

.

If [LV_CHK_DMTL_VLD = 0 → 1 **AND**

LV_CTR_CNL_SMALL_LEAK_MES = 0] **OR**

[LV_CTR_CNL_SMALL_LEAK_MES = 0 → 1 **AND**

LV_CHK_DMTL_VLD = 0 → 1 **AND**

[LV_CUR_DMTL_REF_MES_MIN = 0 → 1 **OR**

LV_CUR_DMTL_REF_MES_MAX = 0 → 1 **OR**

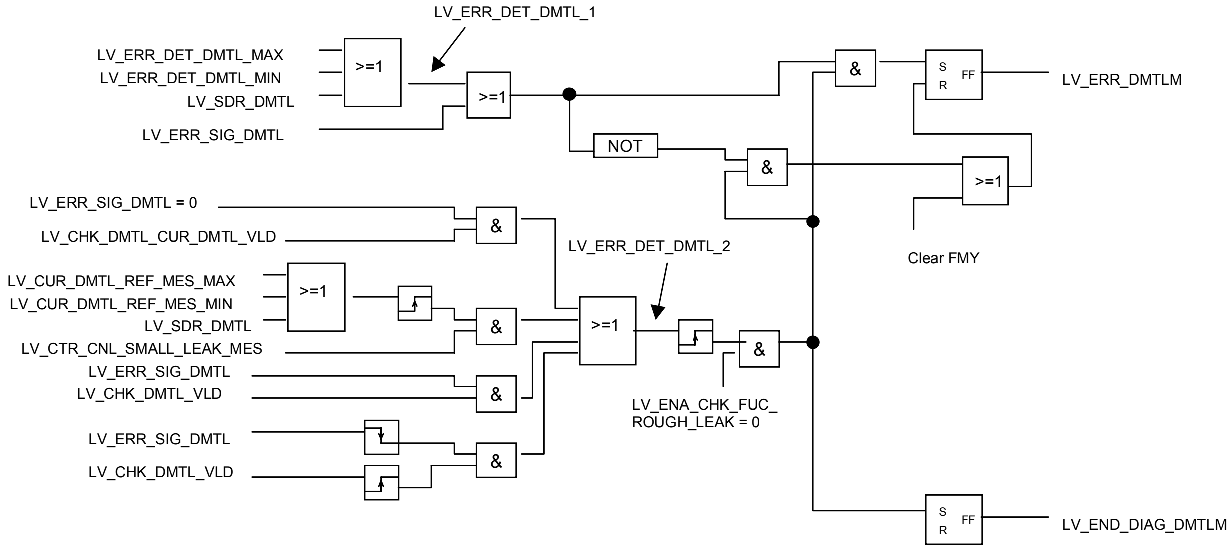
LV_CUR_DMTL_DMTLS_TEST = 0 → 1] **AND**

LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1] **OR**

```
          [LV_CTR_CNL_SMALL_LEAK_MES = 1            AND
          [LV_CUR_DMTL_REF_MES_MIN = 0 → 1 OR
          LV_CUR_DMTL_REF_MES_MAX = 0 → 1 OR
          LV_CUR_DMTL_DMTLS_TEST = 0 → 1]           AND
          LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1]                 OR
          clear FMY
     Then LV_ERR_SIG_DMTL = 0
          ERR_SYM_DMTLM = "NO_SYM"
Endif
```

How the error DMTL module is built up

Signal flow diagram:



Formula section:

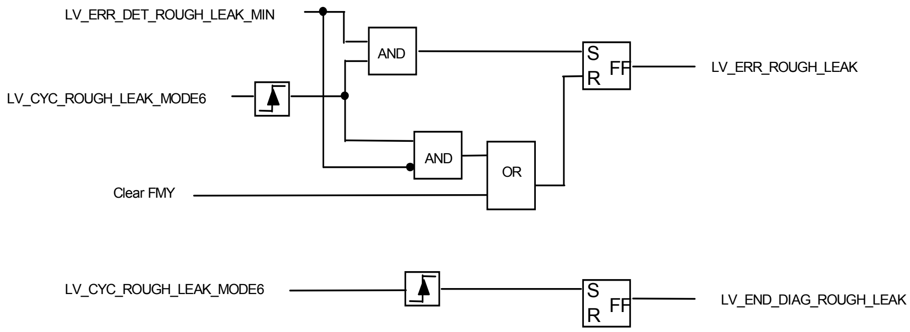
```
If        LV_SDR_DMTL = 1                 OR
          LV_ERR_DET_DMTL_MIN = 1         OR
          LV_ERR_DET_DMTL_MAX = 1
     Then LV_ERR_DET_DMTL_1 = 1
     Else LV_ERR_DET_DMTL_1 = 0
Endif
```

```
If      [LV_ERR_DET_DMTL_1 = 0 → 1           AND
        LV_CTR_CNL_SMALL_LEAK_MES = 1]                  OR
        [LV_ERR_SIG_DMTL = 0                  AND
        LV_CHK_DMTL_CUR_DMTL_VLD = 1]                   OR
        [LV_ERR_SIG_DMTL = 1                  AND
        LV_CHK_DMTL_VLD = 1]                            OR
        [LV_ERR_SIG_DMTL = 1 → 0              AND
        LV_CHK_DMTL_VLD = 0 → 1]
   Then LV_ERR_DET_DMTL_2 = 1
   Else LV_ERR_DET_DMTL_2 = 0
Endif


If      [LV_ERR_DET_DMTL_1 = 1        OR
        LV_ERR_SIG_DMTL = 1]                      AND
        LV_ERR_DET_DMTL_2 = 0 → 1                 AND
        LV_ENA_CHK_FUC_ROUGH_LEAK = 0
   Then LV_CDN_DIAG_DMTLM = 1
        If    LV_ERR_DMTLM = 1
        Then  LV_ERR_DMTLM = 1
              LV_END_DIAG_DMTLM = 1 (result delivered to Error Managment)
              LV_ERR_DMTLM = 0      (result delivered to Error Managment)
              LV_ERR_DMTLM = 1      (result delivered to Error Managment)
        Else  LV_ERR_DMTLM = 1
              LV_END_DIAG_DMTLM = 1 (result delivered to Error Managment)
        Endif
        LV_CDN_DIAG_DMTLM = 0
Endif


If      [LV_ERR_DET_DMTL_1 = 0        AND
        LV_ERR_SIG_DMTL = 0           AND
        LV_ERR_DET_DMTL_2 = 0 → 1     AND
        LV_ENA_CHK_FUC_ROUGH_LEAK = 0]            OR
        clear FMY
   Then LV_CDN_DIAG_DMTLM = 1
        LV_ERR_DMTLM = 0
```

```
            LV_END_DIAG_DMTLM = 1
            LV_CDN_DIAG_DMTLM = 0
Endif


If          LV_ERR_DET_DMTL_2 = 0 → 1             AND
            LV_ENA_CHK_FUC_ROUGH_LEAK = 0
    Then    LV_CDN_DIAG_DMTLM = 1
            LV_END_DIAG_DMTLM = 1
            LV_CDN_DIAG_DMTLM = 0
Endif
```

How rough leak error is built

Signal flow diagram:



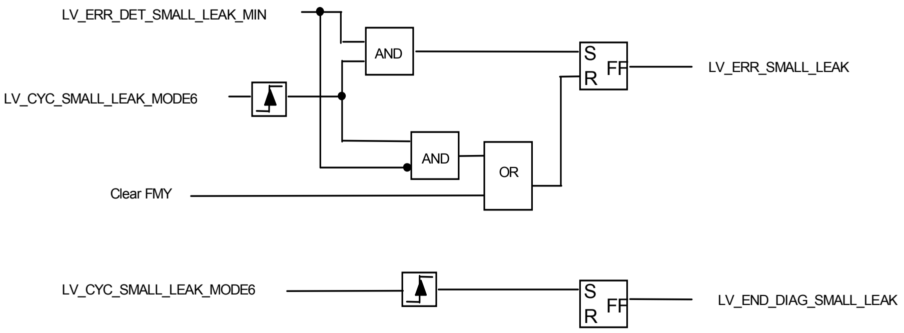
Formula section:

```
If          LV_ERR_DET_ROUGH_LEAK_MIN = 1             AND
            LV_CYC_ROUGH_LEAK_MODE6 = 0 → 1
    Then    LV_CDN_DIAG_ROUGH_LEAK = 1
            LV_ERR_ROUGH_LEAK = 1
            ERR_SYM_ROUGH_LEAK = “MAX”
            LV_CDN_DIAG_ROUGH_LEAK = 0
Endif
If          [LV_ERR_DET_ROUGH_LEAK_MIN = 0            AND
            LV_CYC_ROUGH_LEAK_MODE6 = 0 → 1]                OR
            clear FMY
```

```
    Then LV_CDN_DIAG_ROUGH_LEAK = 1
         LV_ERR_ROUGH_LEAK = 0
         ERR_SYM_ROUGH_LEAK = “NO_SYM”
         LV_CDN_DIAG_ROUGH_LEAK = 0
Endif

If       LV_CYC_ROUGH_LEAK_MODE6 = 0 → 1
    Then LV_END_DIAG_ROUGH_LEAK = 1
Endif
```

How small leak error is built

Signal flow diagram:



Formula section:

```
If       LV_ERR_DET_SMALL_LEAK_MIN = 1          AND
         LV_CYC_SMALL_LEAK_MODE6 = 0 → 1
    Then LV_CDN_DIAG_SMALL_LEAK = 1
         LV_ERR_SMALL_LEAK = 1
         ERR_SYM_SMALL_LEAK = “MIN”
         LV_CDN_DIAG_ROUGH_LEAK = 0
Endif
If       [LV_ERR_DET_SMALL_LEAK_MIN = 0         AND
         LV_CYC_ROUGH_SMALL_MODE6 = 0 → 1]           OR
         clear FMY
    Then LV_CDN_DIAG_ROUGH_LEAK = 1
         LV_ERR_SMALL_LEAK = 0
```

```
              ERR_SYM_DMTL_LEAK = NO_SYM
              LV_CDN_DIAG_ROUGH_LEAK = 0
Endif

If            LV_CYC_SMALL_LEAK_MODE6 = 0 → 1
    Then      LV_END_DIAG_SMALL_LEAK = 1
Endif
```

29.5.20 End of line test and factory test of DMTL function

Application conditions:

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Initialisation: LV_DMTL_ASA = 0

FUNCTION DESCRIPTION:

The end of line test and the factory test must permit execution of the tank leak diagnostic function via test unit.

The fundamental difference is the fact that at the end of line, the timeout monitoring of the test unit signal is deleted and that a change-over of "ignition key" into both directions does not interrupt the function. Another particularity of the end of line function is the fact that the function is started anew without any further test unit intervention if it has been aborted due to "engine speed detected, but START not exited", provided that all enable conditions are present.

On the other hand, activation in the factory is only admissible with the test unit connected and ignition key ON.

Formula section:

```
If            (STATE_EOL_KWP_DMTL  = 0 (Start Command from Tester received) )
    Then      LV_DMTL_ASA = 1
    Else if (STATE_EOL_KWP_DMTL  = 7 (Stop Command from Tester received) )
              Then   LV_DMTL_ASA = 0
              Endif
Endif


If            LV_DMTL_ASA = 1   AND
```

```
            TRT < C_TRT_DMTL_EOL_MAX
    Then    LV_DMTL_EOL = 1
Endif


If          LV_PUC = 1 OR LV_PU = 1 OR LV_PL = 1 OR LV_IS = 1            OR
            LV_DMTL_ACT = 1 → 0                                      OR
            LV_DMTL_STOP = 1                                         OR
            (STATE_EOL_KWP_DMTL  = 7 (Stop Command from Tester received) )  OR
            T_DMTL_EOL > C_T_DMTL_EOL_MAX
    Then    LV_DMTL_EOL = 0
Endif
If          LV_DMTL_EOL = 1
    Then    T_DMTL_EOL = T_DMTL_EOL + 100 ms
    Else    T_DMTL_EOL = 0
Endif



If LV_DMTL_EOL
    Then
    /*DIAGNOSIS CURRENTLY RUNNING,
     OUTPUT OF STATE VARIABLE */
    STATE_DMTL_EOL = STATE_DMTL
    Else /* DIAGNOSIS CANCELLED OR FINISHED */
    If !LV_VB_RANGE_DMTL
            Then
            /* VB IS OUT OF RANGE FOR EOLT */
            STATE_DMTL_EOL = VB_OUT_OF_RANGE;

            Else
            If !LV_ENA_LEAK_DMTL
                    Then
                    /* ELECTRICAL ERROR */
                    STATE_DMTL_EOL = ELECTRICAL_ERR;

                    Else
                    If T_DMTL_EOL >= C_T_DMTL_EOL_MAX
                            Then
                            /* MAX. DURATION OF DIAGNOSIS REACHED */
                            STATE_DMTL_EOL = T_DMTL_MAX;

                            Else
                            If LV_DET_REFU
                                    Then
                                    /* REFUELLMENT */
```

```
                    STATE_DMTL_EOL = REFUELLING;

                    Else
                    If LV_VB_DIF_MAX
                        Then
                        /* GITTER VB TOO LARGE */
                        STATE_DMTL_EOL = VB_GITTER;

                        Else
                        If LV_CUR_DMTL_REF_DIF_MAX
                            Then
                            /* GITTER REFERENCE CURRENT TOO
                            LARGE */
                            STATE_DMTL_EOL = CUR_REF_DIF_MAX;

                            Else
                            If (LV_CUR_DMTL_THD_DIF_MES)
                                Then
                                /* FALLING CURRENT DURING
                                MEASUREMENT */
                                STATE_DMTL_EOL =
                                THD_CUR_DMTL_DIF_MES;
                            Endif
                        Endif
                    Endif
                Endif
            Endif
        Endif
    Endif
Endif
```

29.5.21 Heater DMTL module

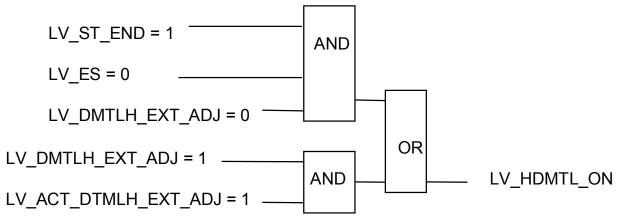
Application conditions:

Recurrence: 100 ms

Activation: LV_CONF_DMTL = 1

Initialisation: LV_HDMTL_ON = 0

Signal flow diagram:



Formula section:

```
If          [(1)LV_DMTLH_EXT_ADJ = 1                AND
            LV_ACT_DMTLH_EXT_ADJ = 1] (1)                 OR
            [(6)LV_DMTLH_EXT_ADJ = 0                                      AND
            LV_ST_END = 1                                                 AND
            LV_ES = 0                                                     AND
     Then   LV_HDMTL_ON = 1
     Else   LV_HDMTL_ON = 0
Endif
```

29.5.22 Mode06 DMTL

General information:

Testvalues and Tresholds has to be prepared for output via obdII-communication mode06. All M6_XX has to be set to 0 in case of clearing failure memory.

Formula section:

Module-Diagnosis, minimum failure:

```
IF   LV_END_DIAG_DMTLM = 1
     THEN
        M6_CUR_DMTL_COR_FIL_CID18 = CUR_DMTL_COR_FIL
     ELSE
        M6_CUR_DMTL_COR_FIL_CID18 = M6_CUR_DMTL_COR_FIL_CID18
ENDIF
```

Module-Diagnosis, maximum failure:

```
IF   LV_END_DIAG_DMTLM = 1
     THEN
          M6_ CUR_DMTL_COR_FIL _CID19 = CUR_DMTL_COR_FIL
```

```
        ELSE
            M6_CUR_DMTL_COR_FIL_CID19 = M6_CUR_DMTL_COR_FIL_CID19
ENDIF

Module-Diagnosis, plausibility failure:
IF  LV_END_DIAG_DMTLM = 1
        THEN
            M6_CUR_DMTL_DMTLS_TEST = CUR_DMTL_DMTLS_TEST
        ELSE
            M6_CUR_DMTL_THD_DMTLS_TEST = CUR_DMTL_THD_DMTLS_TEST
ENDIF

Module-diagnosis, Signal failure:
IF  LV_END_DIAG_DMTLM = 1
        THEN
            M6_CTR_CNL_SMALL_LEAK_MES = CTR_CNL_SMALL_LEAK_MES
        ELSE
            M6_CTR_CNL_SMALL_LEAK_MES = M6_CTR_CNL_SMALL_LEAK_MES
ENDIF


DMTL, Rough-leak, short cycle:
IF  LV_CYC_ROUGH_LEAK_MODE6 = 1 AND LV_ROUGH_LEAK_MES_LEN = 0
        THEN
            M6_CUR_DMTL_ROUGH_LEAK_END = CUR_DMTL_ROUGH_LEAK_END
            M6_CUR_DMTL_THD_ROUGH_LEAK = CUR_DMTL_THD_ROUGH_LEAK
        ELSE
            M6_CUR_DMTL_ROUGH_LEAK_END = M6_CUR_DMTL_ROUGH_LEAK_END
            M6_CUR_DMTL_THD_ROUGH_LEAK = M6_CUR_DMTL_THD_ROUGH_LEAK
ENDIF

DMTL, Small-leak:
IF  LV_CYC_SMALL_LEAK_MODE6 = 1
        THEN
            M6_CUR_DMTL_SMALL_LEAK_END = CUR_DMTL_SMALL_LEAK_END
            M6_CUR_DMTL_REF_LEAK = CUR_DMTL_REF_LEAK
        ELSE
            M6_CUR_DMTL_SMALL_LEAK_END = M6_CUR_DMTL_SMALL_LEAK_END
            M6_CUR_DMTL_REF_LEAK = M6_CUR_DMTL_REF_LEAK
ENDIF

DMTL, Rough-leak, long cycle:
IF  LV_CYC_ROUGH_LEAK_MODE6 = 1 AND LV_ROUGH_LEAK_MES_LEN = 1
        THEN
            M6_CUR_DMTL_ROUGH_LEAK_LEN_END = CUR_DMTL_ROUGH_LEAK_LEN_END
            M6_CUR_DMTL_THD_ROUGH_LEAK_LEN = CUR_DMTL_THD_ROUGH_LEAK_LEN
        ELSE
            M6_CUR_DMTL_ROUGH_LEAK_LEN_END = M6_CUR_DMTL_ROUGH_LEAK_LEN_END
            M6_CUR_DMTL_THD_ROUGH_LEAK_LEN=M6_CUR_DMTL_THD_ROUGH_LEAK_LEN
ENDIF
```

29.6 Tank leakage diagnosis - DMTL (Applic. Inc.)

29.6.1 Inhibition of diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_RBM_DMTL	V/O	0...1H	0...1	1	[-]
Flag to inhibit DMTL diagnosis function if there is a OBD failure					

Input data:

LV_DC	LV_ERR_EL_CPS	LV_ERR_DIAGCPS	LV_ERR_DMTLS
LV_ERR_DMTL_PUMP	LV_ERR_TCO	LV_ERR_VS	LV_ERR_SPI_MPS

FUNCTION DESCRIPTION:

Description:

If one of the following components fail the DMTL diagnosis function is stopped by LV_INH_DIAG_RBM_DMTL = 1.

Application conditions:

Initialization: 0 at reset

Recurrence: 0.1 sec

Activation: LV_DC transition 0 -> 1 (start driving cycle)

Deactivation: LV_DC transition 1 -> 0 (driving cycle terminated)

Formula section:

Inhibition due to OBDI error

```
If      LV_ERR_EL_CPS = 1
    or  LV_ERR_DIAGCPS = 1
    or  LV_ERR_DMTLS = 1
    or  LV_ERR_DMTL_PUMP = 1
    or  LV_ERR_TCO = 1
    or  LV_ERR_VS = 1
    or  LV_ERR_SPI_MPS = 1
Then    LV_INH_DIAG_RBM_DMTL = 1
            (DMTL diagnosis inhibited due to OBD error )
Else    LV_INH_DIAG_RBM_DMTL = 0
            (DMTL diagnosis not inhibited)
Endif
```

29.6.2 Interface for Rate – Based – Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_DMTLM	V/O	0...7H	0...7	1	[-]
Interface of DMTLM monitor with the Rate-Based Monitoring statistics					
STATE_RBM_ROUGH_LEAK	V/O	0...7H	0...7	1	[-]
Interface of ROUGH_LEAK monitor with the Rate-Based Monitoring					
STATE_RBM_SMALL_LEAK	V/O	0...7H	0...7	1	[-]
Interface of SMALL_LEAK monitor with the Rate-Based Monitoring statistics					
CTR_CNL_SMALL_LEAK_MES_VIRT	V/O/S	0...FFH	0...255	1	[-]
Virtuel number of aborted miniature leak measurements dur to current fluctuations					
LV_CTR_CNL_SMALL_LEAK_VIRT	V/O/S	0...1H	0...1	1	[-]
Virtuel aborted miniature leak measurements due to current fluctuations exceeded					
LV_CHK_DMTL_CUR_DMTL_VLD_VIRT	V/O/S	0...1H	0...1	1	[-]
Virtuel check of component concerning valid pump current					
LV_DET_SMALL_LEAK	V/O/S	0...1H	0...1	1	[-]
Stored bit, that in the last DC small leak was recognized					
LV_DET_NO_SMALL_LEAK	V/O/S	0...1H	0...1	1	[-]
Stored bit, that in the last DC no small leak was recognized					
LV_DET_NO_ROUGH_LEAK	V/O/S	0...1H	0...1	1	[-]
Condition leak check done and no rough leak detected					
T_AST_DMTL_RBM	V	0...FFFFH	0...6553.5	0.1	[s]
comulative time since engine start for incrementation of the evaporative system monitor denominator					

Input data:

LV_DC	CTR_ERR_DYN_NR	LV_END_DIAG_DMTLM	LV_CHK_DMTL_CUR_DMTL_VLD
LV_CHK_DMTL_VLD	C_SUM_CNL_SMALL_LEAK_MES_MAX	LV_ERR_DMTLM	LV_CYC_ROUGH_LEAK_MODE6
LV_CTR_CNL_SMALL_LEAK_MES	LV_ERR_DET_SMALL_LEAK_MIN	LV_CYC_SMALL_LEAK_MODE6	LV_IGK
LV_REFU	LV_REFU_VAL	C_DIST_REFU_ROUGH_LEAK_MIN	LV_ERR_DET_ROUGH_LEAK_MIN
TCO_ST	TAM_ST	TAM	VS
LV_FTL_DMTL_VAL	T_ES_CUS		

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the tank leakage detection monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_DMTLM, STATE_RBM_ROUGH_LEAK and STATE_RBM_SMALL_LEAK data.

Within STATE_RBM_xxx, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)

  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)

  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

  (not valid for DMTL function)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_DMTLM = 0
bit 0, bit 1 and bit 2 of STATE_RBM_ROUGH_LEAK = 0
bit 0, bit 1 and bit 2 of STATE_RBM_SMALL_LEAK = 0
T_AST_DMTL_RBM = 0
LV_REFU_VAL_DC = 0
LV_DET_ROUGH_LEAK_VIRT = 0
LV_DET_SMALL_LEAK_VIRT = 0
DIST_DET_ROUGH_LEAK_VIRT = 0
T_DLY_FTL_DMTL_VAL = C_T_DLY_FTL_DMTL_VAL

on failure memory reset :

bit 1 of STATE_RBM_DMTLM = 0
bit 1 of STATE_RBM_ROUGH_LEAK = 0
bit 1 of STATE_RBM_SMALL_LEAK = 0
CTR_CNL_SMALL_LEAK_MES_VIRT = 0
LV_CTR_CNL_SMALL_LEAK_VIRT = 0
LV_CHK_DMTL_CUR_DMTL_VLD_VIRT = 0
LV_DET_SMALL_LEAK = 0
LV_DET_NO_SMALL_LEAK = 0
LV_DET_NO_ROUGH_LEAK = 0
LV_DET_ROUGH_LEAK = 0
DIST_DET_ROUGH_LEAK_VIRT = 0

at reset:

CTR_CNL_SMALL_LEAK_MES_VIRT out of NVMY
LV_CTR_CNL_SMALL_LEAK_VIRT out of NVMY
LV_CHK_DMTL_CUR_DMTL_VLD_VIRT out of NVMY
LV_DET_SMALL_LEAK out of NVMY
LV_DET_NO_SMALL_LEAK out of NVMY
LV_DET_NO_ROUGH_LEAK out of NVMY
T_DLY_FTL_DMTL_VAL = C_T_DLY_FTL_DMTL_VAL

Recurrence: 100ms

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_DIAGCPS	LV_ERR_EL_CPS	LV_ERR_TCO_EL	LV_ERR_TCO_GRD
LV_ERR_TCO_STUCK	LV_ERR_TCO_PLAUS	LV_ERR_VS	LV_ERR_DMTLS
LV_ERR_DMTL_PUMP	LV_ERR_SPI_MPS		

```
If(1)     { CPU optimization at LV_DC 0 → 1 transition }
          CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
          While  bit 1 of STATE_RBM_DMTLM = 0 do
                 with each XX failure of the above list :
                 ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                 SYNCHRONIZATION<CALL>)
                 If(2)          XX has a pending status
                 Then(2)
                                bit 1 of STATE_RBM_DMTLM = 1
                 Endif(2)
          Endwhile
Else(1)
          { the dynamic failure memory is empty }
          No action
Endif(1)

If(1)     { CPU optimization at LV_DC 0 → 1 transition }
          CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
```

```
            While  bit 1 of STATE_RBM_ROUGH_LEAK = 0 do
                   with each XX failure of the above list :
                   ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                   SYNCHRONIZATION<CALL>)
                   If(2)          XX has a pending status
                   Then(2)
                                  bit 1 of STATE_RBM_ROUGH_LEAK = 1
                   Endif(2)
            Endwhile
Else(1)
            { the dynamic failure memory is empty }
            No action
Endif(1)


If(1)       { CPU optimization at LV_DC 0 → 1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While  bit 1 of STATE_RBM_SMALL_LEAK = 0 do
                   with each XX failure of the above list :
                   ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                   SYNCHRONIZATION<CALL>)
                   If(2)          XX has a pending status
                   Then(2)
                                  bit 1 of STATE_RBM_SMALL_LEAK = 1
                   Endif(2)
            Endwhile
Else(1)
            { the dynamic failure memory is empty }
            No action
Endif(1)
```

numerator calculation DMTL module error

Every 100 ms

```
If      LV_CHK_DMTL_VLD = 0 → 1
Then    CTR_CNL_SMALL_LEAK_MES_VIRT ++
Endif

If      CTR_CNL_SMALL_LEAK_MES_VIRT >
        C_SUM_CNL_SMALL_LEAK_MES_MAX
Then    LV_CTR_CNL_SMALL_LEAK_VIRT = 1
        CTR_CNL_SMALL_LEAK_MES_VIRT = 0
Endif

If      (LV_CHK_DMTL_VLD = 1                    AND
        LV_CTR_CNL_SMALL_LEAK_MES = 1)
Then    CTR_CNL_SMALL_LEAK_MES_VIRT = 0
Endif

If      LV_CHK_DMTL_CUR_DMTL_VLD = 0 → 1          OR
        LV_CTR_CNL_SMALL_LEAK_MES = 0 → 1
Then    LV_CHK_DMTL_CUR_DMTL_VLD_VIRT = 1
Endif

If      bit 0 of STATE_RBM_DMTLM = 0
Then
        If      [LV_CTR_CNL_SMALL_LEAK_VIRT = 1             OR
                [LV_ERR_DMTLM = 1          AND
                LV_CHK_DMTL_CUR_DMTL_VLD_VIRT = 1]]               AND
                LV_IGK = 1
        Then    bit 0 of STATE_RBM_DMTLM = 1
                LV_CTR_CNL_SMALL_LEAK_VIRT = 0
                LV_CHK_DMTL_CUR_DMTL_VLD_VIRT = 0
        Endif
Endif
```

numerator calculation DMTL rough leak error

Every 100 ms:

```
If      LV_FTL_DMTL_VAL = 0
Then    T_DLY_FTL_DMTL_VAL –
        If      T_DLY_FTL_DMTL_VAL > 0
        Then    LV_FTL_DMTL_VAL_VIRT = 0
        Else    LV_FTL_DMTL_VAL_VIRT = 1
        Endif
Else    T_DLY_FTL_DMTL_VAL = C_T_DLY_FTL_DMTL_VAL
        LV_FTL_DMTL_VAL_VIRT = 0
Endif

If      ((LV_REFU_VAL_DC = 0            AND
        LV_REFU = 0                     AND
        LV_REFU_VAL = 1)                        OR
        (LV_FTL_DMTL_VAL_VIRT = 1       AND
        C_COD_DMTL_RBM = 0. bit))                       AND
        T_ES_CUS > C_T_ES_DMTL_DIAG_MAX
Then    LV_T_ES_NO_REFU = 1
Else    LV_T_ES_NO_REFU = 0
Endif

If      LV_ERR_DET_ROUGH_LEAK_MIN = 0   AND
        LV_CYC_ROUGH_LEAK_MODE6 = 1
Then    LV_DET_NO_ROUGH_LEAK = 1
Endif

If      LV_DET_NO_ROUGH_LEAK = 1
Then    If      LV_REFU_VAL = 0 → 1     AND
                LV_REFU = 1
        Then    DIST_DET_ROUGH_LEAK_VIRT = 0
        Endif

        If      [LV_FTL_DMTL_VAL = 1    AND
                LV_ERR_VS = 0           AND
```

```
                LV_IGK = 1                          AND
                LV_VS_RUN = 1]
        Then    DIST_DET_ROUGH_LEAK_VIRT$_{(n)}$ =
                DIST_DET_ROUGH_LEAK_VIRT$_{(n-1)}$ + (VS / 36)
        Endif

        If      LV_ERR_VS = 1                       OR
                LV_FTL_DMTL_VAL_VIRT = 1            OR
                DIST_DET_ROUGH_LEAK_VIRT ≥
                C_DIST_REFU_ROUGH_LEAK_MIN
        Then    LV_DIST_DMTL_VIRT = 1
        Else    LV_DIST_DMTL_VIRT = 0
        Endif
Endif

If      LV_ERR_DET_ROUGH_LEAK_MIN = 1       AND
        LV_CYC_ROUGH_LEAK_MODE6 = 1
Then    LV_DET_ROUGH_LEAK = 1
        LV_DET_NO_ROUGH_LEAK = 0
Endif
If      LV_DET_NO_ROUGH_LEAK = 1    AND
        LV_T_ES_NO_REFU =1
Then    LV_DET_ROUGH_LEAK_VIRT = 1
Endif

If      bit 0 of STATE_RBM_ROUGH_LEAK = 0
Then
        If      [[LV_DIST_DMTL_VIRT = 1             AND
                LV_DET_ROUGH_LEAK_VIRT = 1]             OR
                LV_DET_ROUGH_LEAK = 1]                          AND
                LV_IGK = 1
        Then    bit 0 of STATE_RBM_ROUGH_LEAK = 1
                LV_DET_NO_ROUGH_LEAK = 0
                LV_DET_ROUGH_LEAK = 0
        Endif
Endif
```

```
If      [T_ES_CUS ≤ C_T_ES_DMTL_DIAG_MAX          OR
        (LV_REFU_VAL_DC = 0                 AND
        LV_REFU = 1                  AND
        LV_REFU_VAL = 1)]                                  AND
        LV_IGK = 1
Then    LV_T_ES_REFU = 1
        LV_DET_NO_ROUGH_LEAK = 0
Else    LV_T_ES_REFU = 0
Endif
```

numerator calculation DMTL small leak error

Every 100 ms

```
If      LV_ERR_DET_SMALL_LEAK_MIN = 1       AND
        LV_CYC_SMALL_LEAK_MODE6 = 1
Then    LV_DET_SMALL_LEAK = 1
        LV_DET_NO_SMALL_LEAK = 0
Endif
If      LV_ERR_DET_SMALL_LEAK_MIN = 0       AND
        LV_CYC_SMALL_LEAK_MODE6 = 1
Then    LV_DET_NO_SMALL_LEAK = 1
Endif
If      LV_DET_NO_SMALL_LEAK = 1            AND
        LV_T_ES_NO_REFU = 1
Then    LV_DET_SMALL_LEAK_VIRT = 1
Endif

If      bit 0 of STATE_RBM_SMALL_LEAK = 0
Then    If      [LV_DET_SMALL_LEAK = 1          OR
                LV_DET_SMALL_LEAK_VIRT = 1]           AND
                LV_IGK = 1
        Then    bit 0 of STATE_RBM_SMALL_LEAK = 1
                LV_DET_SMALL_LEAK = 0
                LV_DET_NO_SMALL_LEAK = 0
        Endif
Endif
```

```
          If    LV_T_ES_REFU = 1
          Then  LV_DET_NO_SMALL_LEAK = 0
          Endif
          If    LV_REFU_VAL = 1
          Then  LV_REFU_VAL_DC = 1
          Endif



If   bit 1 of STATE_RBM_DMTLM = 0
Then
          If    LV_INH_DIAG_RBM_DMTL = 1
          Then  bit 1 of STATE_RBM_DMTLM = 1
          Endif
Endif
If   bit 1 of STATE_RBM_ROUGH_LEAK = 0
Then
          If    LV_INH_DIAG_RBM_DMTL = 1
          Then  bit 1 of STATE_RBM_ROUGH_LEAK = 1
          Endif
Endif
If   bit 1 of STATE_RBM_SMALL_LEAK = 0
Then
          If    LV_INH_DIAG_RBM_DMTL = 1
          Then  bit 1 of STATE_RBM_SMALL_LEAK = 1
          Endif
Endif

### denominator definition

If   C_TAM_MIN_DMTL_RBM ≤ TAM ≤ C_TAM_MAX_DMTL_RBM          AND
     LV_ST_END = 1
Then      if    T_AST_DMTL_RBM < C_T_AST_DMTL_RBM
                T_AST_DMTL_RBM ++
          Endif
```

```
Endif

If      bit 2 of STATE_RBM_DMTLM = 0
Then If     TCO_ST < TAM_ST + C_TAM_ST_HYS_DMTL_RBM                      AND
        C_TCO_ST_MIN_DMTL_RBM ≤ TCO_ST ≤ C_TCO_ST_MAX_DMTL_RBM          AND
        T_AST_DMTL_RBM ≥ C_T_AST_DMTL_RBM
            Then        bit 2 of STATE_RBM_DMTLM = 1
            Endif
Endif

If      bit 2 of STATE_RBM_ROUGH_LEAK = 0
Then If     TCO_ST < TAM_ST + C_TAM_ST_HYS_DMTL_RBM                      AND
        C_TCO_ST_MIN_DMTL_RBM ≤ TCO_ST ≤ C_TCO_ST_MAX_DMTL_RBM          AND
        T_AST_DMTL_RBM ≥ C_T_AST_DMTL_RBM
            Then        bit 2 of STATE_RBM_ROUGH_LEAK = 1
            Endif
Endif

If      bit 2 of STATE_RBM_SMALL_LEAK = 0
Then If     TCO_ST < TAM_ST + C_TAM_ST_HYS_DMTL_RBM                      AND
        C_TCO_ST_MIN_DMTL_RBM ≤ TCO_ST ≤ C_TCO_ST_MAX_DMTL_RBM          AND
        T_AST_DMTL_RBM ≥ C_T_AST_DMTL_RBM
            Then        bit 2 of STATE_RBM_SMALL_LEAK = 1
            Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_FTL_DMTL_VAL	1	0...FFFFH	0...6553.5	0.1	[s]
Delay time for a valid FTL for DMTL_RBM					
C_T_ES_DMTL_DIAG_MAX	1	0...FFFFH	0...65535	1	[min]
Maximum time for a DMTL diagnosis					
C_TAM_MIN_DMTL_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Minimum ambient temperature threshold for DMTL_RBM					
C_TAM_MAX_DMTL_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Maximum ambient temperature threshold for DMTL_RBM					
C_T_AST_DMTL_RBM	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum comulative time since engine start for incrementation of the evaporative system monitor denominator					
C_TAM_ST_HYS_DMTL_RBM	1	80...7FH	-96...95.25	0.75	[°C]
Ambient temperature hysteresis for DMTL_RBM					
C_TCO_ST_MIN_DMTL_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Minimum engine start temperature threshold for DMTL_RBM					
C_TCO_ST_MAX_DMTL_RBM	1	0...FEH	-48...142.5	0.75	[°C]
Maximum engine start temperature threshold for DMTL_RBM					
C_COD_DMTL_RBM	1	0...FFH	0...255	1	[-]
configuration byte for DMTL RBM					

30 Exhaust system temperature control

Table of Contents

Index

30.1 AGGR EXTC adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CH_MOD_IVVT	V/O	0...FFH	0...255	1	[-]
Bit-coded mask, containing combustion-mode-request for catalyst heating and the desulfation strategy; 2. Prio					
STATE_CH_MOD_IVVT_REQ	V/O	0...FFH	0...255	1	[-]
Bit-coded mask, containing combustion-mode-request for catalyst heating and the desulfation strategy; 1. Prio					

Input data:

STATE_CH_MOD_REQ	STATE_CH	STATE_OPM_IVVT_CH	LV_SO2P_REQ

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

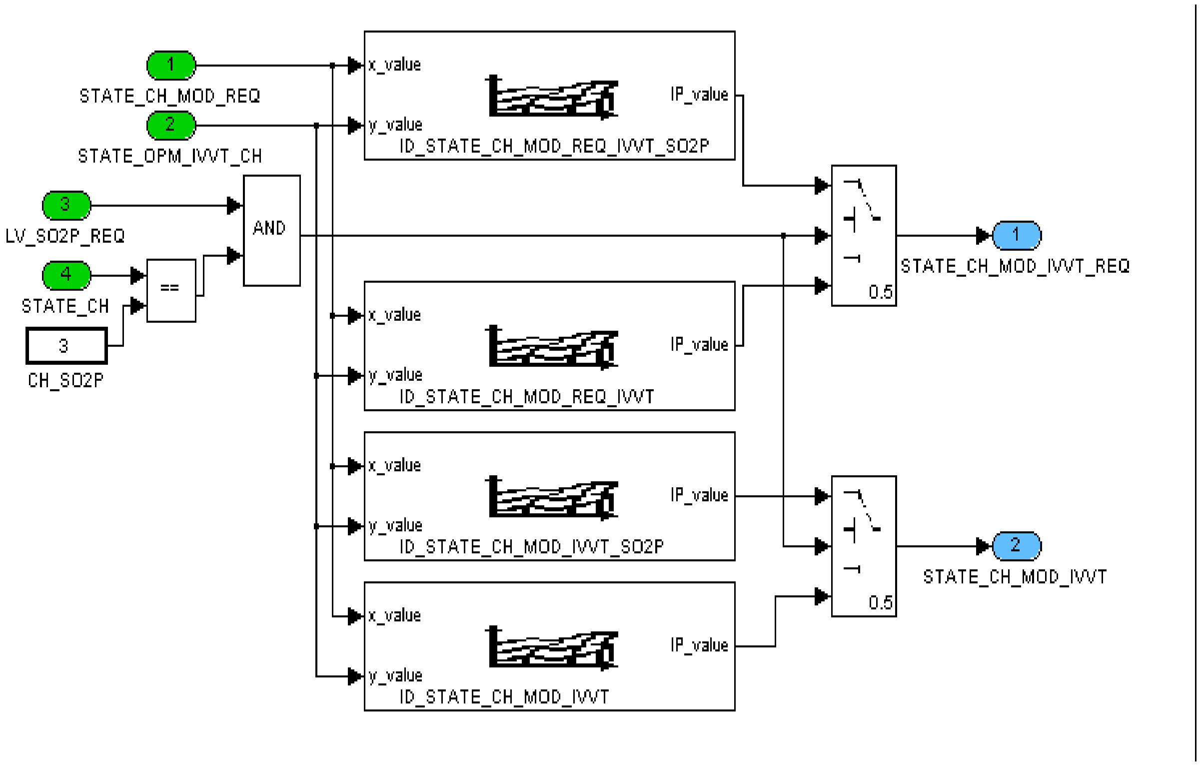
Application conditions:

Initialisation: *0 at reset except*

Recurrence : 10 ms

Activation: *every engine state*

Formula section:



Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_STATE_CH_MOD_REQ_IVVT_SO2P	5*3	0...FFH	0...255	1	[-]
LDPM_STATE_CH_MOD_REQ	5	0...4H	0...4	1	[-]
LDPM_STATE_OPM_IVVT_CH	3	0...2H	0...2	1	[-]
Matrix of Bit-mask in case of desulfation for 1.Prio-Request					
ID_STATE_CH_MOD_REQ_IVVT	5*3	0...FFH	0...255	1	[-]
LDPM_STATE_CH_MOD_REQ	5	0...4H	0...4	1	[-]
LDPM_STATE_OPM_IVVT_CH	3	0...2H	0...2	1	[-]
Matrix of Bit-mask for 1.Prio-Request					
ID_STATE_CH_MOD_IVVT_SO2P	5*3	0...FFH	0...255	1	[-]
LDPM_STATE_CH_MOD_REQ	5	0...4H	0...4	1	[-]
LDPM_STATE_OPM_IVVT_CH	3	0...2H	0...2	1	[-]
Matrix of Bit-mask in case of desulfation for 2.Prio-Request					
ID_STATE_CH_MOD_IVVT	5*3	0...FFH	0...255	1	[-]
LDPM_STATE_CH_MOD_REQ	5	0...4H	0...4	1	[-]
LDPM_STATE_OPM_IVVT_CH	3	0...2H	0...2	1	[-]
Matrix of Bit-mask for 2.Prio-Request					

30.2 Catalyst heating functions and engine warm up

30.2.1 Catalyst Heating functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CH_N_SP_IS	V/O	0...1H	0...1	1	[-]
Catalyst Heating Function "increased idle speed setpoint"					
LV_TI_CH	V/O	0...1H	0...1	1	[-]
Catalyst Heating Function "lean injection"					

Input data:

STATE_CH			

General information:

Logical catalyst heating variables that have been used up to now are set depending on STATE_CH.

Application conditions:

Initialisation: all outputs = 0 at ECU initialization

Recurrence: 100 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

```
If STATE_CH = 1
    Then
    LV_TI_CH = 1
    Else
    LV_TI_CH = 0
    Endif

IF STATE_CH = 1 OR 2
    Then
    LV_CH_N_SP_IS = 1
    Else
    LV_CH_N_SP_IS = 0
    endif
```

30.2.2 Warm-up function : (LV_WUP)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_WUP	V/O	0...1H	0...1	1	[-]
Warm up functions active					

Input data:

LV_ES	TI_WUP_1_ADD		

FUNCTION DESCRIPTION:

General information:

The injection is corrected with the warm-up enrichment.

The states warm-up (LV_WUP) and catalyst heating (LV_TI_CH) are used parallel.

Application conditions:

Initialisation :	LV_WUP = 1
Recurrence :	once per segment (720 °crk / NC_CYL_NR)
Activation :	at all engine operating states
Deactivation :	TI_WUP_1_ADD = 0 (refer to chapter : ” Injection ”). Is activated again at STATE_ENG = LV_ES

Formula section:

```
if    LV_ES = 1
  then      LV_WUP  =  1
endif


if    LV_WUP = 1
   then      if       TI_WUP_1_ADD > 0
                then          LV_WUP  unchanged  (stays at value 1)
                else          LV_WUP  =  0
             endif
   else      LV_WUP  unchanged  (stays at value 0)
endif
```

30.3 Catalyst heating coordination

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CH_AST_CDN	V	0... 1H	0... 1	1	[-]
Indicates if conditions for after start catalyst heating are fulfilled					
LV_CH_L_CDN	V	0... 1H	0... 1	1	[-]
Indicates if conditions for low load catalyst heating are fulfilled					
LV_CH_L_RGN	O/V	0... 1H	0... 1	1	[-]
Low load catalyst heating for NOx-trap - heating after regeneration is active					
LV_CH_SO2P_CDN	V	0... 1H	0... 1	1	[-]
Indicator for conditions of desulfation catalyst heating					
STATE_CH	O/V	0 1 2 3	CH_OFF CH_AST CH_L_LOAD CH_SO2P	-	[-]
State of catalyst heating					
TEMP_CAT_DIF_CH_L	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[K]
Difference between actual catalyst temperature and deactivation threshold for low load CH					
TNT_DIF_CH_L	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[K]
Difference between actual NOx-trap-temperature and deactivation threshold for low load CH					
TNT_MDL_L_COR_CH_L	V	0... FFFFH	0... 1023.984375	0.015625	[°C]
Lowest NOx-Trap temperature - adapted for low load catalyst heating activation					
TQI_REQ_FAST_FIL_CH_SO2P	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Filtered torque request (fast path) for desulfation catalyst heating activation					
VS_FIL_CH_SO2P	V	0... FFH	0... 255	1	[km/h]
Filtered vehicle speed					

Input Data:

T_AST_COR_CH	T_ES_CUS	TCO_ST	TEMP_CAT
TNT_MDL_H	TQI_REQ_FAST	N_32	LV_HOM_ACT
LV_SO2P_REQ_2	VS	LV_ST_END	LV_NT_SO2P_EXT_ADJ_ACT
CTR_KM_CAN	TCO	FTL	NT_AGI
TNT_MDL_L	LV_CH_AST_REQ_TCHA_DIAG	LV_NT_RGN_REQ	TEMP_CAT_DYN_MDL [NC_CBK_EX_NR]
LV_CH_AST_ENA	LV_STATE_CH_SO2P_ENA	LV_CH_L_ENA	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_FAC_TQI_FIL_CH_SO2P	1	0... FFFFH	0... 1	15.259e-6	[-]
Correlation factor to filter the requested torque					
C_FAC_VS_FIL_CH_SO2P	1	0... FFFFH	0... 1	15.259e-6	[-]
Correlation factor to filter the vehicle speed					
C_FTL_HYS_CH_SO2P	1	0... 7FH	0... 127	1	[l]
Hysteresis of fuel tank level threshold for deactivation of desulfation catalyst heating					
C_FTL_MIN_CH_SO2P	1	0... 7FH	0... 127	1	[l]
Minimum fuel tank level, deactivates desulfation catalyst heating					
C_N_32_MIN_NT_SO2P_EXT_ADJ_CH	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed to start catalyst heating for desulfation, activated by external adjustment					

C_N_32_THD_CH_L_OFF	1	0... FFH	0... 8160	32	[rpm]
Threshold of engine speed that disables CH_L					
C_N_32_THD_CH_L_ON	1	0... FFH	0... 8160	32	[rpm]
Threshold of engine speed that enables CH_L					
C_STATE_CH_MAN	1	0 1 2 3	CH_OFF CH_AST CH_L_LOAD CH_SO2P	-	[-]
Manual setting of state of cat. heating					
C_T_MAX_CH_L_RGN	1	0... FFFFH	0... 6553.5	0.1	[s]
Maximum duration of low load heating after regeneration phase					
C_TCO_HYS_CH_SO2P	1	0... FEH	-48... 142.5	0.75	[°C]
Hysteresis of coolant temperature threshold for activation of catalyst heating for desulfation					
C_TCO_MIN_CH_SO2P	1	0... FEH	-48... 142.5	0.75	[°C]
Minimum coolant temperature for activation of catalyst heating for desulfation					
C_TEMP_CAT_DYN_MDL_THD_CH_SO2P	1	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
Threshold of catalyst temperature for desulfation catalyst heating					
C_TEMP_CAT_MAX_CH_AST	1	0... FFFFH	-33... 990.984375	0.015625	[°C]
Threshold of catalyst temperature for catalyst heating after start					
C_TEMP_CAT_THD_CH_L_HOM_ON	1	0... FFFFH	-33... 990.984375	0.015625	[°C]
Threshold of catalyst temperature for homogeneous CH_L					
C_TEMP_CAT_THD_CH_L_OFF	1	0... FFFFH	-33... 990.984375	0.015625	[°C]
Threshold of catalyst temperature for deactivation of low load CH					
C_TEMP_CAT_THD_CH_L_S_ON	1	0... FFFFH	-33... 990.984375	0.015625	[°C]
Threshold of catalyst temperature for stratified CH_L					
C_TNT_HYS_CH_L_RGN	1	8000... 7FFFH	-512... 511.984375	0.015625	[°C]
Hysteresis for activation of low load catalyst heating for heating after regeneration					
C_TNT_MAX_CH_AST	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Maximum nox trap temperature for catalyst heating after start					
C_TNT_MAX_CH_SO2P	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Maximum temperature of TNT for desulfation catalyst heating					
C_TNT_MIN_CH_L_RGN	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Minimum NOx-Trap temperature for activation of low load catalyst heating for heating after regeneration					
C_TNT_MIN_CH_SO2P	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Minimum temperature of TNT for desulfation catalyst heating					
C_TNT_THD_CH_L_HOM_ON	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Threshold NOx-trap-temperature that activates CH in homogeneous					
C_TNT_THD_CH_L_OFF	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Threshold NOx-trap-temperature that deactivates CH in stratified and homogeneous					
C_TNT_THD_CH_L_RGN	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
If NOx-Trap temperature falls below this threshold low load catalyst heating for heating after regeneration is activated					
C_TNT_THD_CH_L_S_ON	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Threshold NOx-trap-temperature that activates CH in stratified					
C_TQI_REQ_FAST_THD_CH_L_OFF	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Threshold of TQI_REQ_FAST that disables CH_L					
C_TQI_REQ_FAST_THD_CH_L_ON	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Threshold of TQI_REQ_FAST that enables CH_L					
C_TQI_THD_CH_SO2P	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Threshold of requested torque to activate desulfation catalyst heating					
C_VS_MAX_CH_AST	1	0... FFH	0... 255	1	[km/h]
Vehicle speed threshold for after start catalyst heating					
C_VS_THD_CH_SO2P	1	0... FFH	0... 255	1	[km/h]
threshold of vehicle speed to activate desulfation catalyst heating					

IP_T_AST_THD_CH_AST	4*8	0... FFFFH	0... 6553.5	0.1	[s]
LDP_TCO_ST_IP_T_AST_THD_CH_AST	8	0... FEH	-48... 142.5	0.75	[°C]
LDPM_CTR_KM_CAN_1_EXTC	4	0... FFFFH	0... 655350	10	[km]
Threshold of time after start for catalyst heating after start					
IP_T_ES_THD_CH_AST	8	0... FFFFH	0... 65535	1	[min]
LDP_TCO_ST_IP_T_ES_THD_CH_AST	8	0... FEH	-48... 142.5	0.75	[°C]
Threshold of engine off duration time which stops catalyst heating after start					
IP_TNT_ADD_CH_L_NT_AGI	4	0... FFFFH	-1024... 1023.96875	0.03125	[K]
LDP_NT_AGI_IP_TNT_ADD_CH_L	4	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Additive temperature value on low-load-catalyst-heating-activation-thresholds					
IP_TQI_REQ_FAST_THD_CH_L_RGN	4	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_TQI_REQ_CH_L_RGN	4	0... FFH	0... 8160	32	[rpm]
Torque - threshold to deactivate low load catalyst heating after regeneration					
LC_STATE_CH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of state of cat. heating					

General Information

This module has to be calculated before any other catalyst heating module.
In this module the decision is made if catalyst heating after start, in low load or for desulfation is active. STATE_CH indicates the actual status and triggers other modules.

Application Conditions

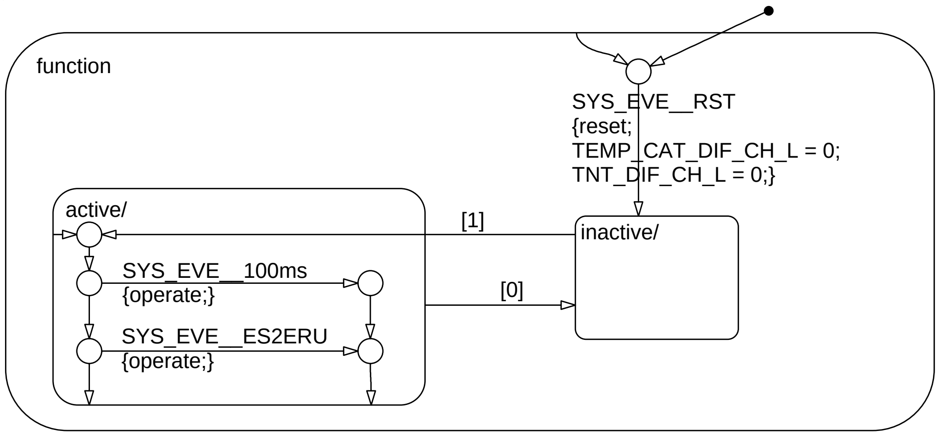
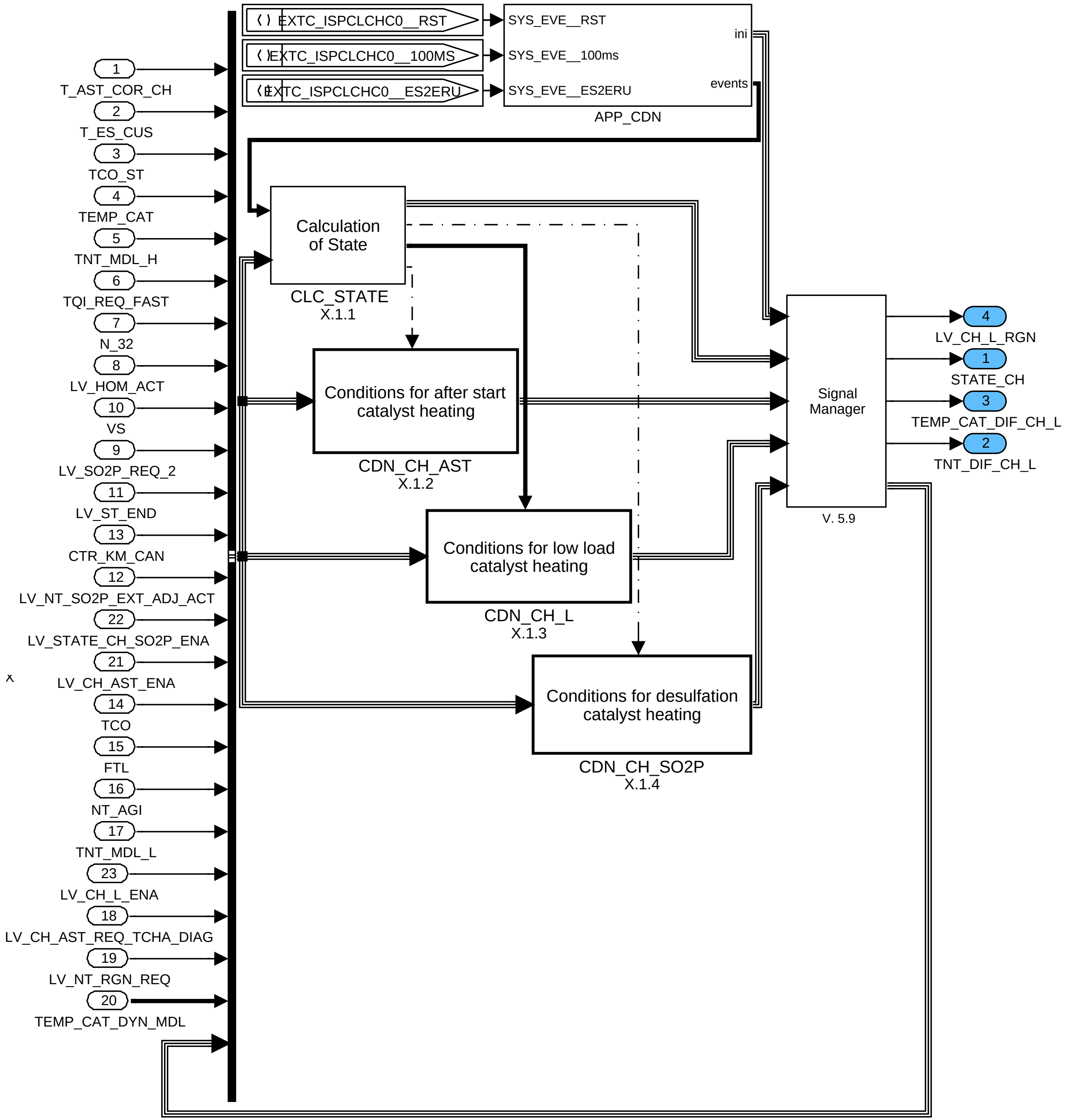


Figure 1:
Path: EXTC_ISPCLCHC0/APP_CDN/Chart

Function description



SDA_SRS / SDA V 5.0.6 07−Dec−2006

Figure 2:
Path: EXTC_ISPCLCHC0

30.3.1 State machine

30.3.1.1 State machine - Subsystem

The conditions for activation of several catalyst heating measures are not calculated if another measure is active see stateflow chart.

With to LC_STATE_CH_MAN_ACT it is possible to switch STATE_CH manually.

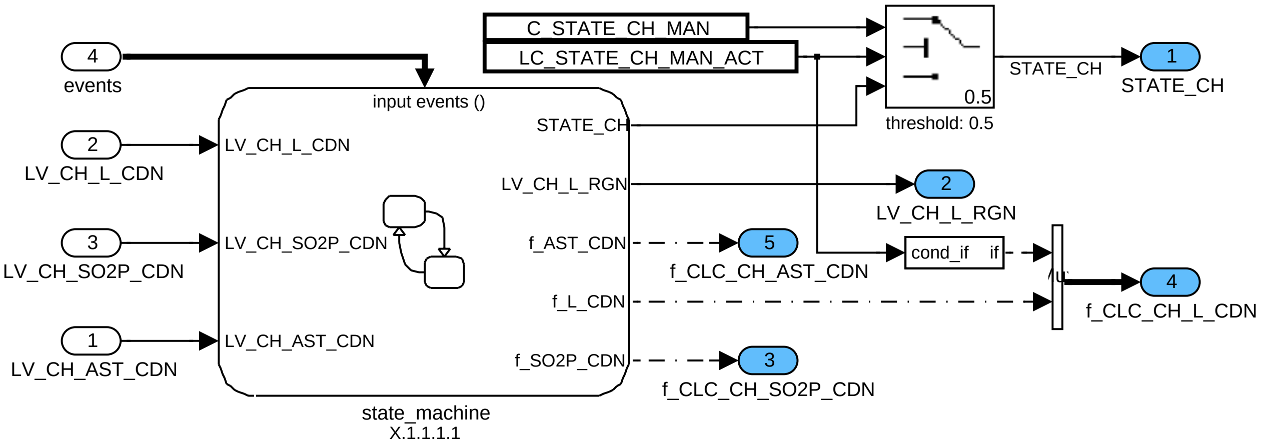


Figure 3:
Path: EXTC_ISPCLCHC0/CLC_STATE/SUB

30.3.1.1.1 state_machine

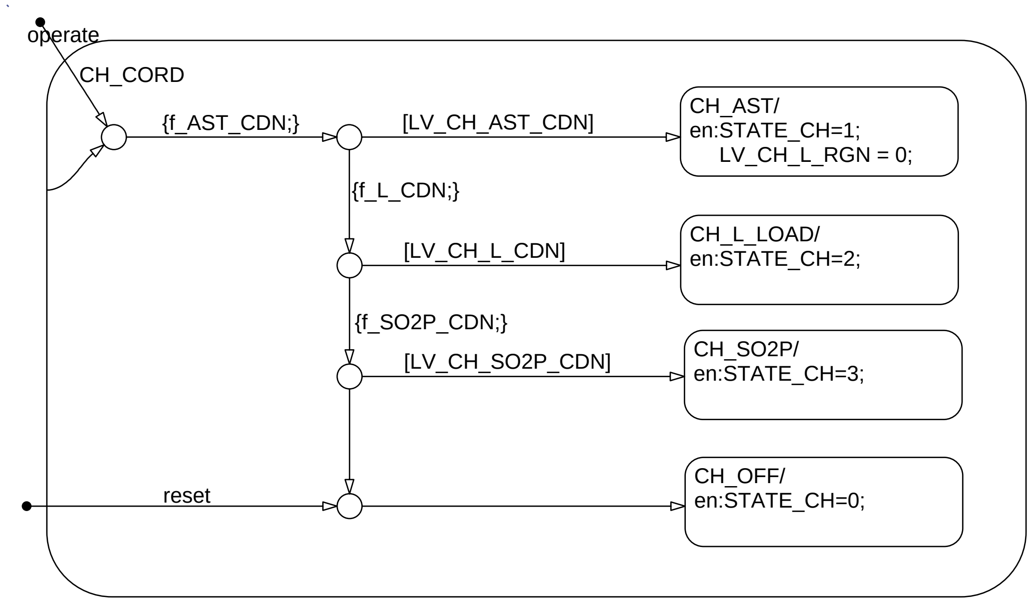


Figure 4:
Path: EXTC_ISPCLCHC0/CLC_STATE/SUB/state_machine

30.3.2 Conditions for after start heating

30.3.2.1 Conditions for after start heating - Subsystem

T_ES_CUS is the time with stopped engine. If one of the conditions stops catalyst heating, it only can be re-activated after next start.

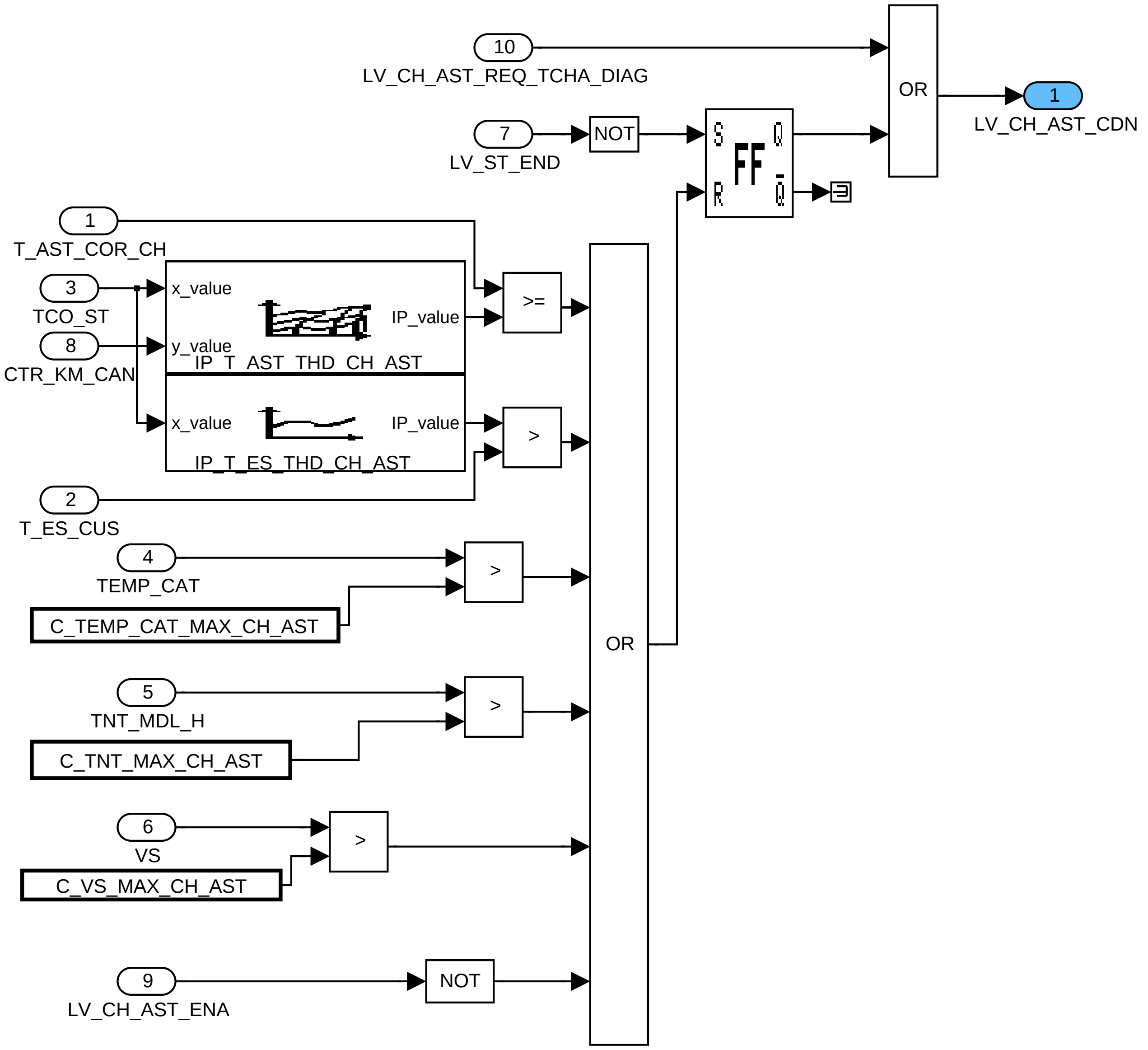


Figure 5:
Path: EXTC_ISPCLCHC0/CDN_CH_AST/SUB

30.3.3 Conditions for low load heating

30.3.3.1 Conditions for low load heating - Subsystem

TNT_MDL_L can be adjusted depending on NOx-trap aging.
Low load catalyst heating can be activated due to low temperatures in NOx-Trap or Pre-catalyst ('CH_L_CDN') or after regeneration of NOx-trap. In 2nd case the lambda-split function is activated (LV_CH_L_RGN = 1).
The differences between the actual temperatures and the thresholds TNT_DIF_CH_L and TEMP_CAT_DIF_CH_L are used as mapinputs in the setpoint calculation modules e.g. for torque reserve.

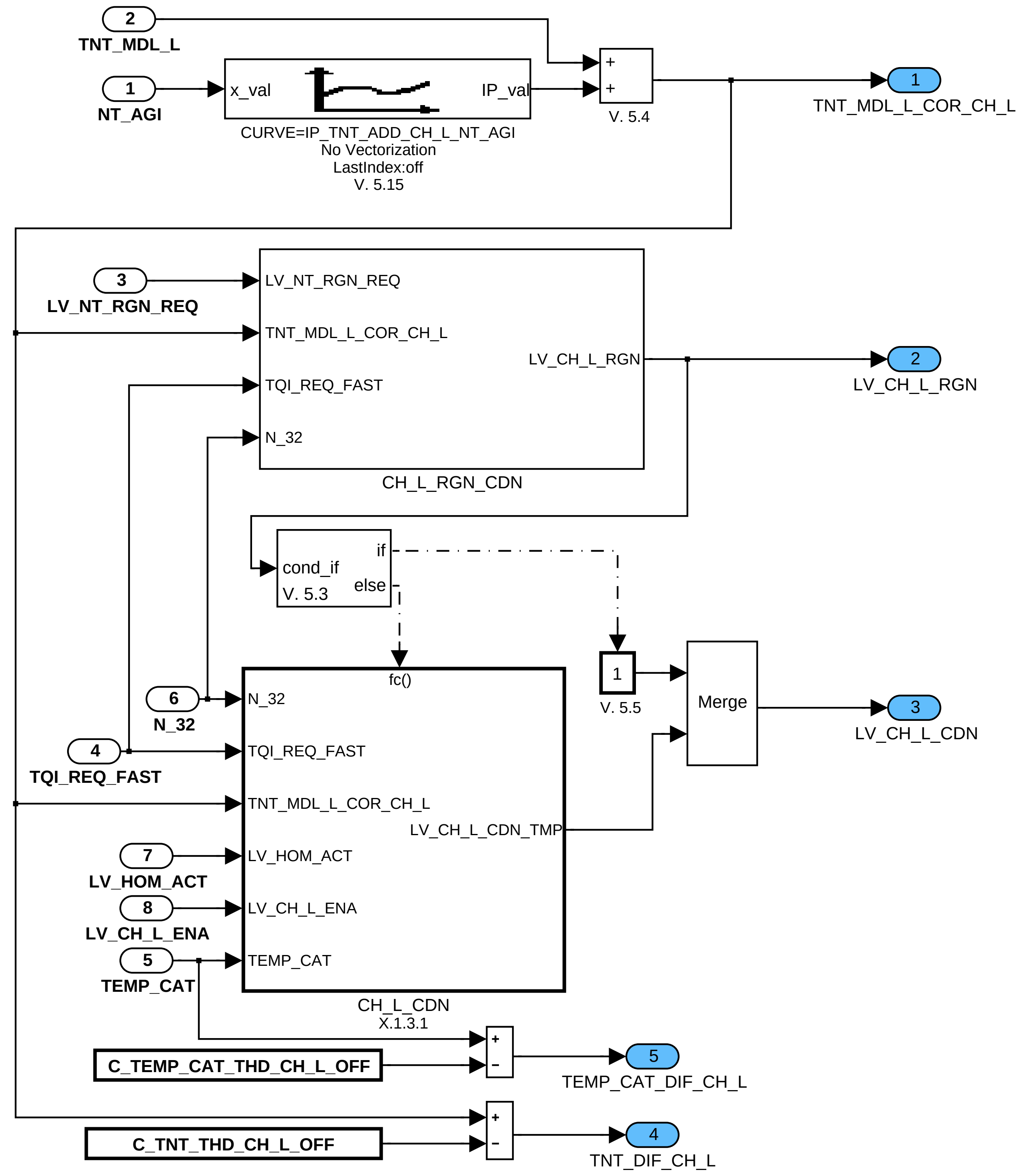


Figure 6:
Path: EXTC_ISPCLCHC0/CDN_CH_L/SUB

30.3.3.1.1 Conditions for low load heating - Subsystem

For torque, engine speed and temperature there are each two thresholds for hysteresis application. For homogenous and stratified mode there are several activation thresholds for catalyst temperature. Attention: In low load catalyst heating one measure is to request homogeneous mode at too low temperatures! (See "Catalyst heating low load strategy"). These temperature thresholds have to be calibrated cohesive.

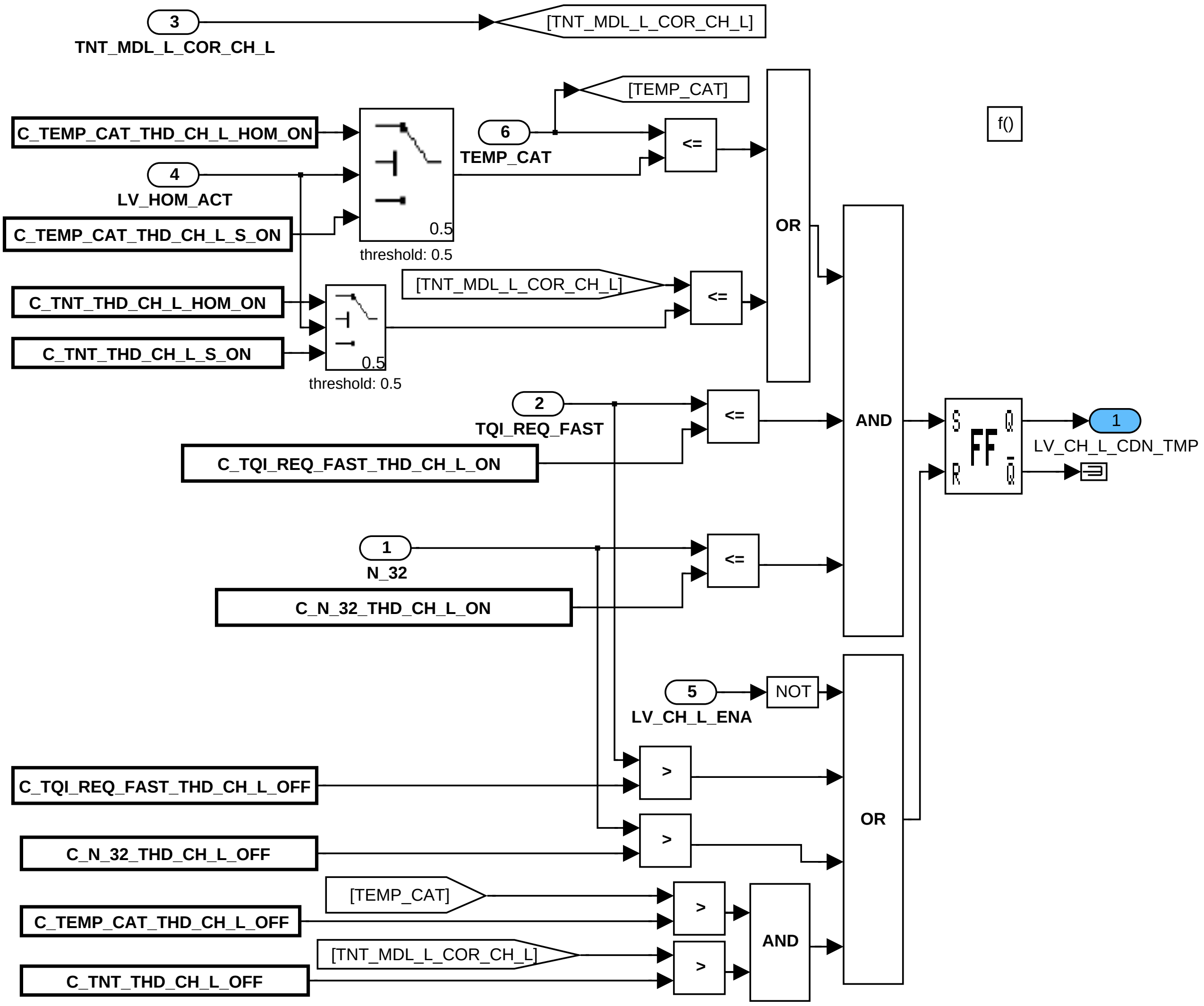


Figure 7:
Path: EXTC_ISPCLCHC0/CDN_CH_L/SUB/CH_L_CDN

30.3.3.1.2 Conditions for low load heating after regeneratin (lambda split function)

LV_NT_RGN_REQ is 1 at active regeneration. The function can be deactivated depending on load-, temperature- or time threshold.

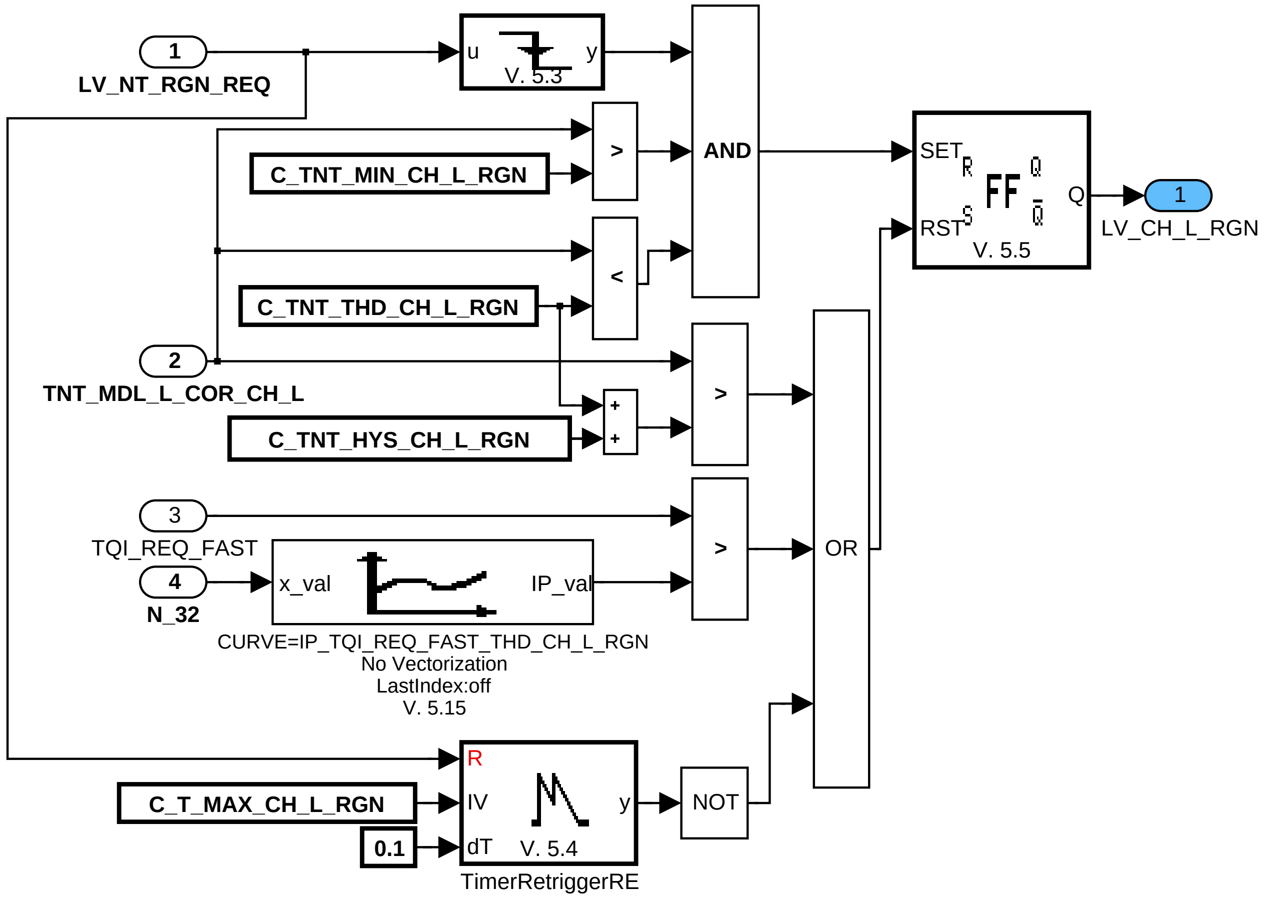


Figure 8:
Path: EXTC_ISPCLCHC0/CDN_CH_L/SUB/CH_L_RGN_CDN

30.3.4 Conditions for desulfation catalyst heating

30.3.4.1 Conditions for desulfation catalyst heating - 1st part

Vehicle speed and requested torque are filtered to avoid often on and off switching. TCO and FTL thresholds can only be re-set at engine off.

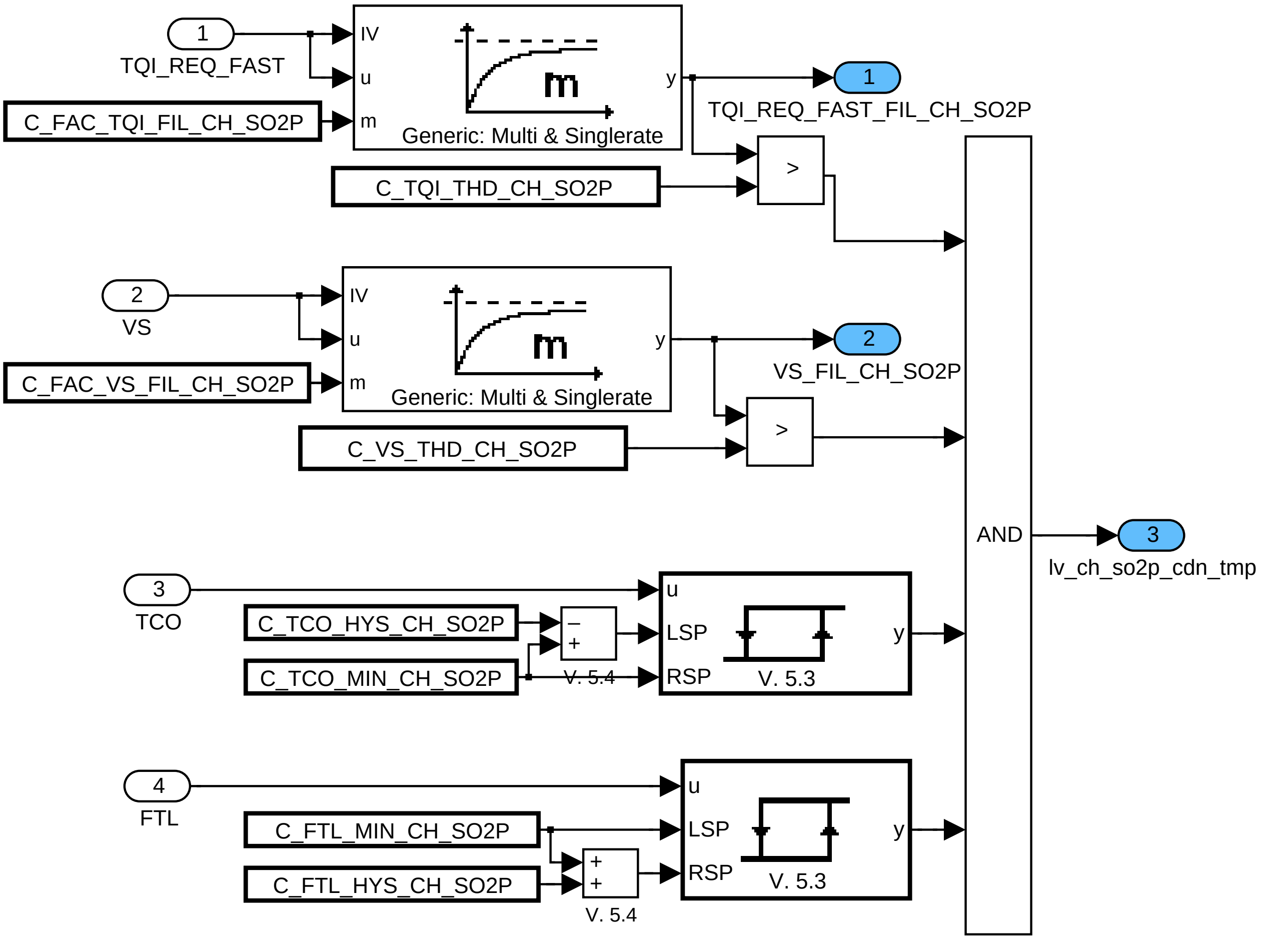


Figure 9:
Path: EXTC_ISPCLCHC0/CDN_CH_SO2P/SUB_1

30.3.4.2 Conditions for desulfation catalyst heating - 2nd part

LV_SO2P_REQ_2 indicates that catalyst heating is necessary due to imperatively desulfation.

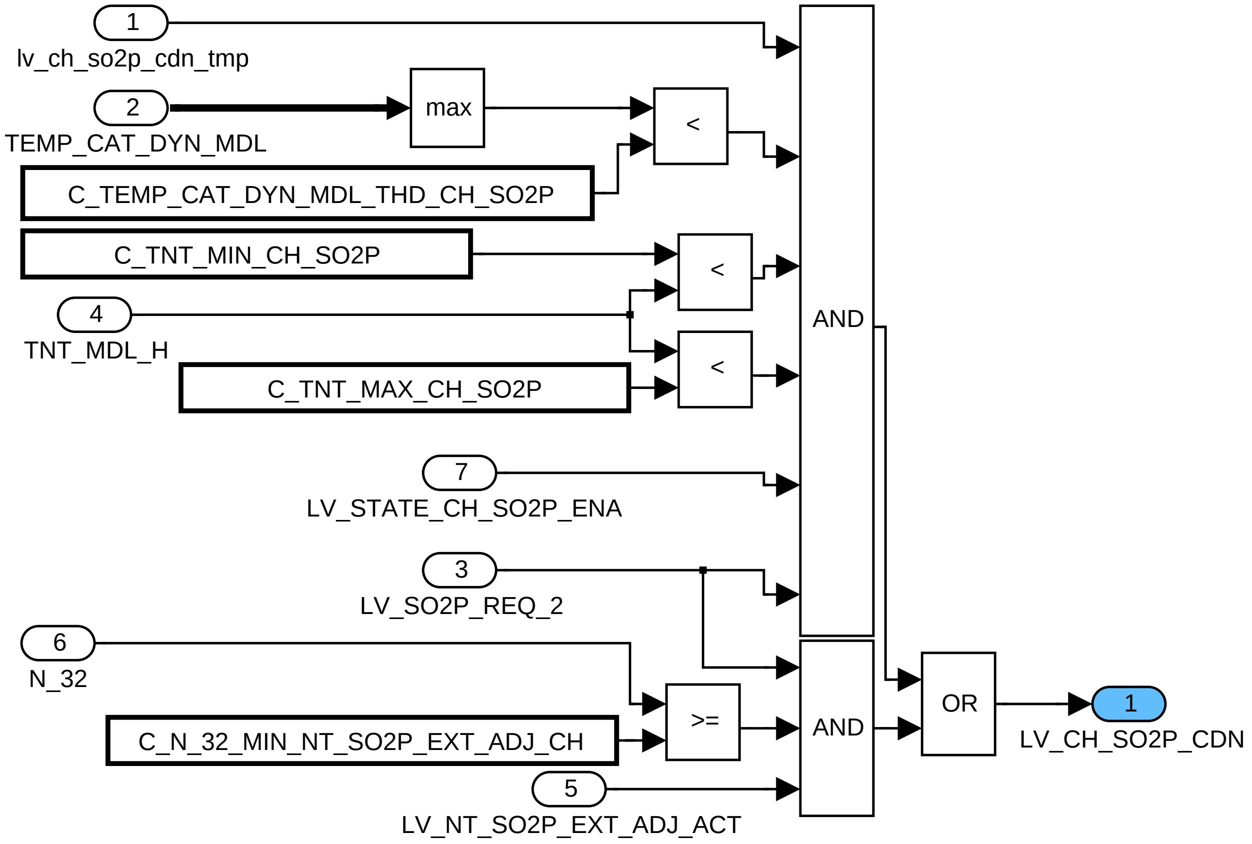


Figure 10:
Path: EXTC_ISPCLCHC0/CDN_CH_SO2P/SUB_2

30.4 Catalyst heating coordination (Appl. Inc.)

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_CH_SO2P	O/V/S	0... FFH	0... 255	1	[-]
Counter of not effectual trials to heat the NOx - Trap for desulfation					
CTR_KM_CH_SO2P_INH	O/V/S	0... FFFFH	0... 655350	10	[km]
Mileage at inhibition of catalyst heating for desulfation due to CTR_CH_SO2P or T_CH_SO2P_ACT					
CTR_KM_SO2P_END	O/V/S	0... FFFFH	0... 655350	10	[km]
Mileage at last successful desulfation					
DIST_CH_SO2P_INH	O/V	0... FFFFH	0... 655350	10	[km]
Km - counter since inhibition of catalyst heating for desulfation due to CTR_CH_SO2P or T_CH_SO2P_ACT					
DIST_SO2P_END	O/V	0... FFFFH	0... 655350	10	[km]
Km - counter since last end of desulfation					
LV_CH_AST_ENA	O/V	0... 1H	0... 1	1	[-]
indicates if after start catalyst heating is enabled					
LV_CH_AST_INH_OPM	V	0... 1H	0... 1	1	[-]
Inhibition of catalyst heating (input from operation mode manager)					
LV_CH_L_ENA	O/V	0... 1H	0... 1	1	[-]
Inhibition of catalyst heating in low load					
LV_CH_OPM_REQ_CUS_CLC	V	0... 1H	0... 1	1	[-]
Calculation of operation mode manager - conditions for after start catalyst heating active					
LV_CTR_KM_CH_SO2P_INH	V	0... 1H	0... 1	1	[-]
Start of mileage counter CTR_KM_CH_SO2P_INH					
LV_STATE_CH_SO2P_ENA	O/V	0... 1H	0... 1	1	[-]
Catalyst heating for desulfation is enabled					
T_CH_SO2P_ACT	O/V/S	0... FFFFH	0... 6553.5	0.1	[s]
Time with active catalyst heating for desulfation without active desulfation					

Input Data:

C_TNT_SP_CH_SO2P	NT_AGI_SUL	TNT_MDL_MV	LV_SO2P_REQ_2
CTR_KM_CAN	LV_SO2P_REQ	LV_CH_INH	OPM_REQ_CUS
LV_CH	LV_ERR_TEG_PCAT_DOWN	LV_CH_SO2P_WOUT_LIM	STATE_CH

Calibration Data:

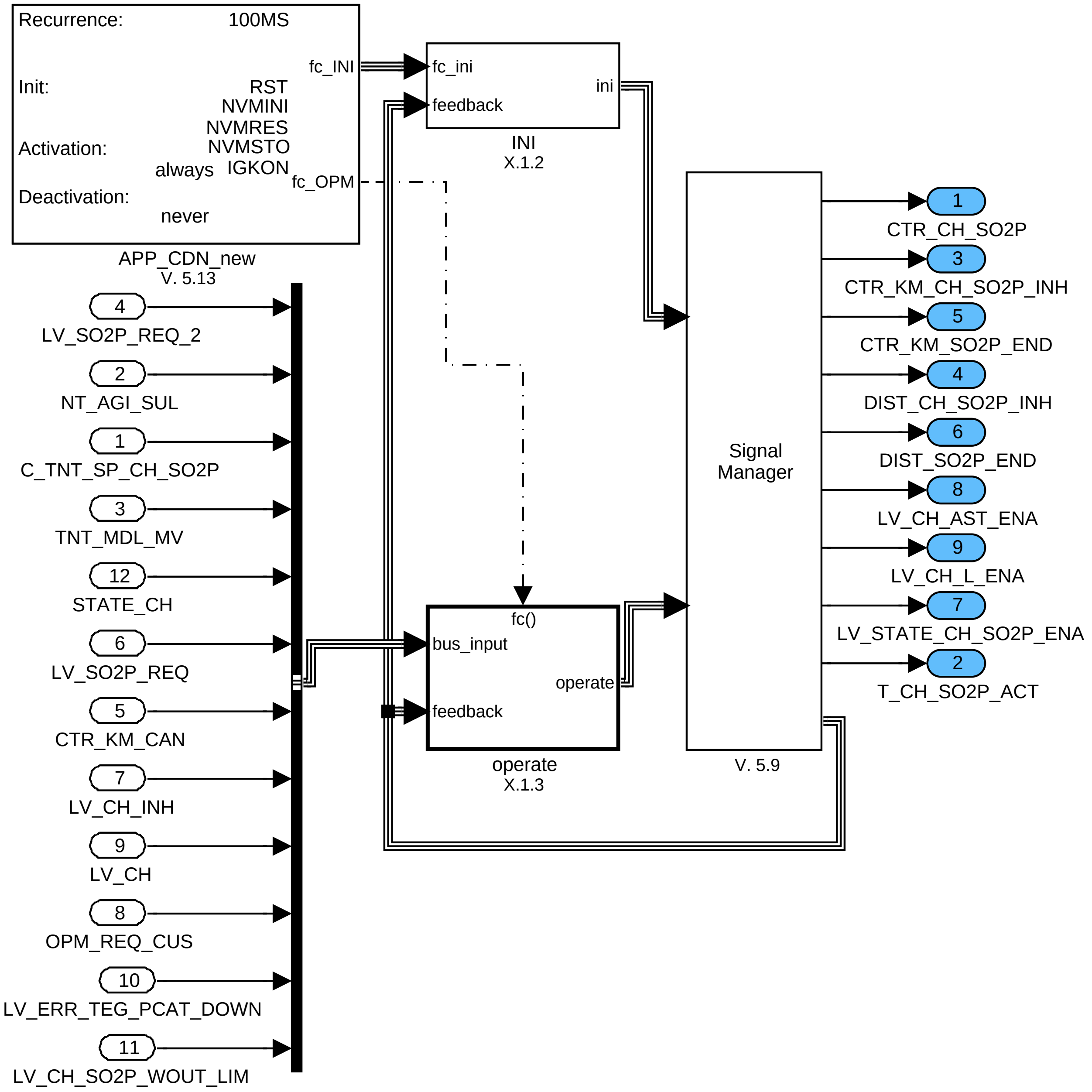
Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CTR_CH_SO2P_MAX	1	0... FFH	0... 255	1	[-]
Maximum allowed heating activations within calibratable km - distance					
C_DIST_CH_SO2P_INH_MAX	1	0... FFFFH	0... 655350	10	[km]
Maximum km-counter since inhibition of catalyst heating for desulfation - to allow heating again					
C_DIST_SO2P_END_MIN	1	0... FFFFH	0... 655350	10	[km]
Minimum km since last desulfation to activate catalyst heating for desulfation					
C_T_CH_SO2P_ACT_MAX	1	0... FFFFH	0... 6553.5	0.1	[s]
Maximum heating time without activation of desulfation					
IP_TNT_DELTA_MAX_CH_SO2P	6	0... FFFFH	0... 1023.984375	0.015625	[°C]
LDP_NT_AGI_SUL_IP_TNT_CH_SO2P	6	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Permissible temperature difference to be bridged					

In this module additional conditions for catalyst heating for desulfation of NOx Trap are defined.

Application Conditions

Initialization: RST, NVMINI, NVMRES, NVMSTO, IGKON
Recurrence: 100MS
Activation: always
Deactivation: never

Function description



x

SDA_SRS / SDA V 5.0.6 13−Oct−2006

Figure 11:
Path: EXTC_ISPCLCHCAI0

30.4.1 Initialisation / NVMY - Handling

30.4.1.1 Initialisation at reset

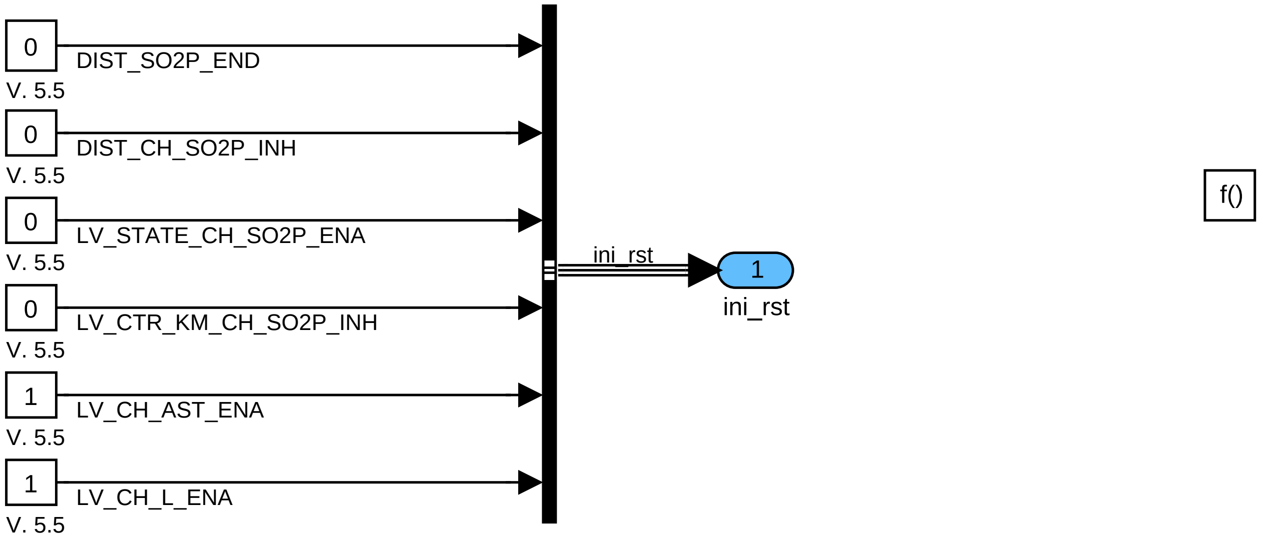


Figure 12:
Path: EXTC_ISPCLCHCAI0/INI/RST_INI

30.4.1.2 Initialisation at ignition key on

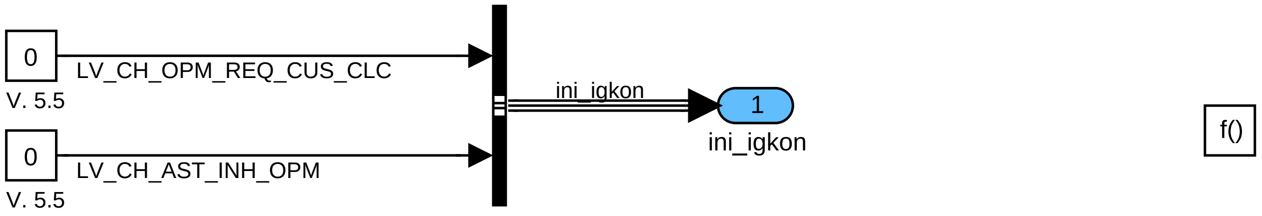


Figure 13:
Path: EXTC_ISPCLCHCAI0/INI/INI_IGKON

30.4.2 Operate Subsystem

30.4.2.1 Conditions for desulfation - catalyst heating

LV_CTR_KM_CH_SO2P_INH_INI starts the kilometre counter DIST_CH_SO2P_INH.

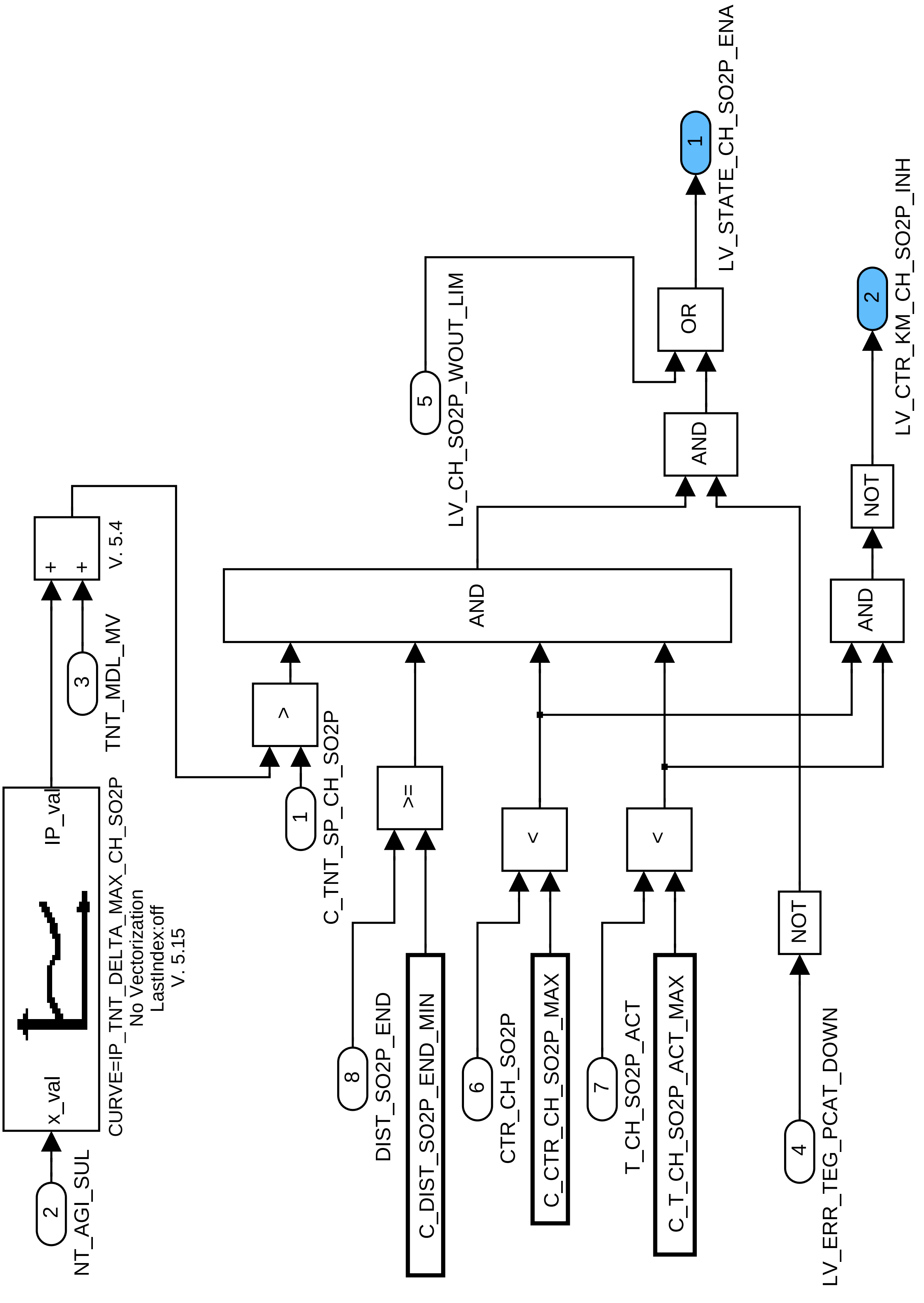


Figure 14:
Path: EXTC_ISPCLCHCAI0/operate/CLC_SO2P_CDN

30.4.2.2 Calculation of timer and counter - for desulfation catalyst heating

CTR_CH_SO2P counts the ineffective tryings to heat up the catalyst. T_CH_SO2P_ACT is the time, the catalyst is heated without started desulfation (temperature for desulfation not reached).DIST_SO2P_END are the kilometres that have been driven since the last successful desulfation. DIST_CH_SO2P_INH are the kilometres that have been driven since the time- or counter-threshold was reached. After C_DIST_CH_SO2P_INH_MAX is reached, the timer and the counter are initialized and the catalyst is heated again.

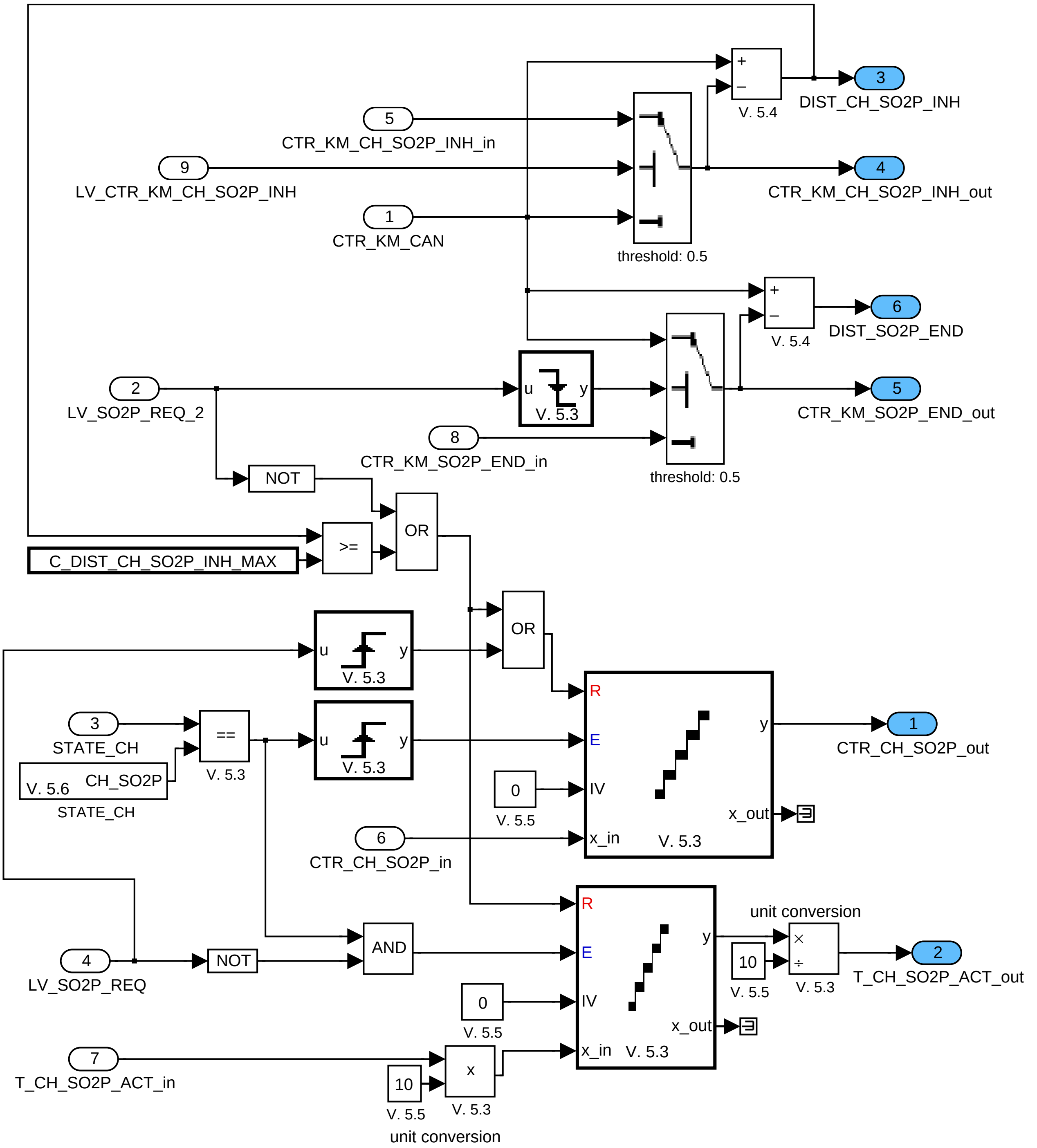


Figure 15:
Path: EXTC_ISPCLCHCAI0/operate/CLC_T_DIST_CTR

30.4.2.3 External conditions for after start catalyst heating

LV_CH_AST_INH_OPM and LV_CH_AST_INH_OPM are set to 0 at system event ignition key on.

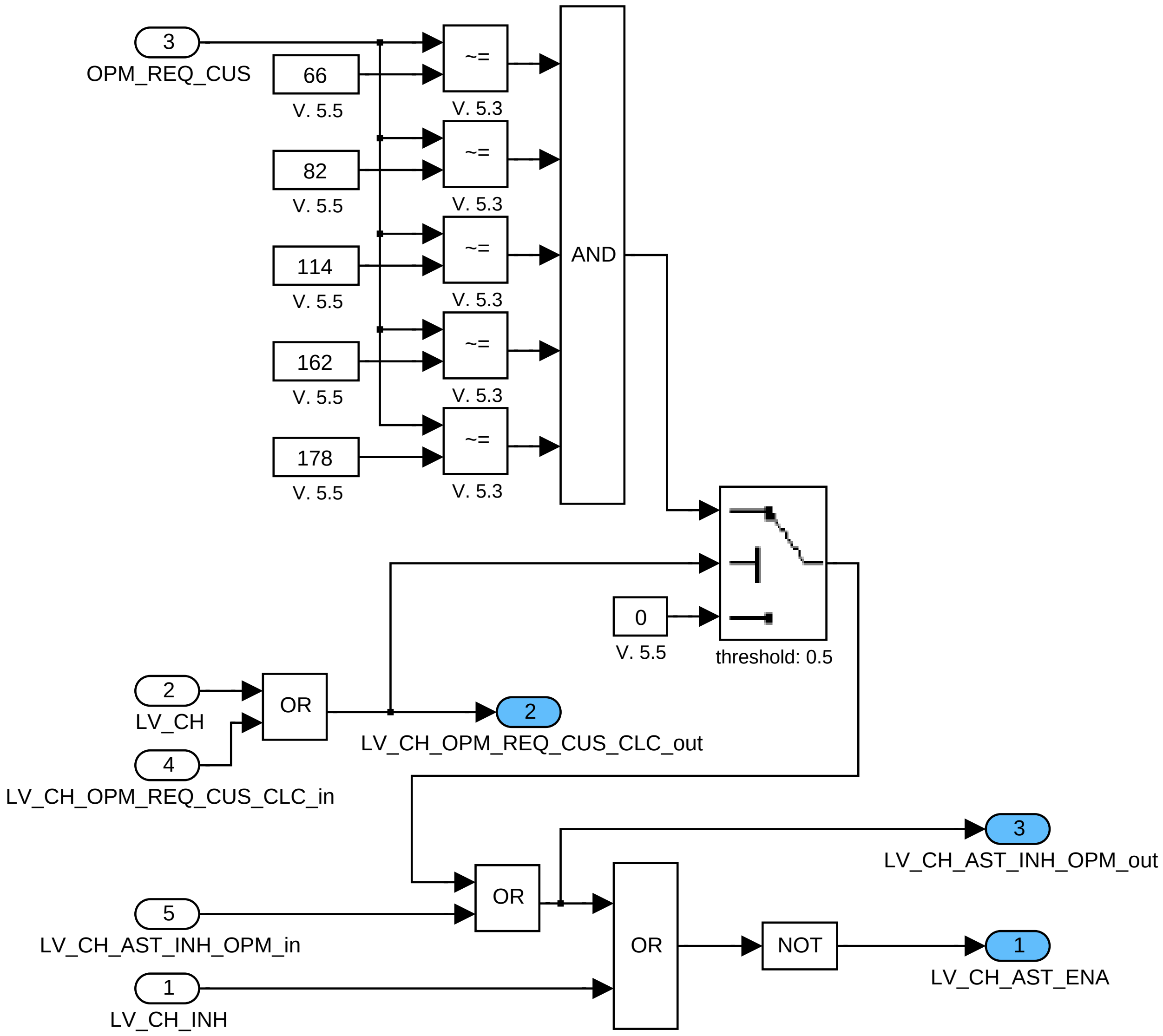


Figure 16:
Path: EXTC_ISPCLCHCAI0/operate/CH_AST_CH_L

30.5 Catalyst heating strategy

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
NR_PRI_CH_MOD	O/V	80... 7FH	-128 ...127	1	[-]
Priority of catalyst heating strategy					
NR_PRI_CH_MOD_REQ	O	80... 7FH	-128 ...127	1	[-]
Priority of requested catalyst heating strategy					
STATE_CH_MOD	O/V	0 1 2 3 4	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	-	[-]
Catalyst heating strategy					
STATE_CH_MOD_REQ	O	0 1 2 3 4	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	-	[-]
Requested CH-strategy (combustion mode) for catalyst heating					

Input Data:

STATE_CH_MOD_L	STATE_CH_MOD_REQ_L	NR_PRI_CH_MOD_REQ_L	STATE_CH_MOD_AST
STATE_CH_MOD_REQ_AST	STATE_CH	STATE_CH_MOD_SO2P	STATE_CH_MOD_REQ_SO2P
LV_DLY_CH_MOD_REQ_SO2P_HOM	LV_CH_AST_REQ_TCHA_DIAG		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_NR_PRI_CH_MOD	1	80... 7FH	-128 ...127	1	[-]
Priority of catalyst heating strategy					
C_NR_PRI_CH_MOD_REQ	1	80... 7FH	-128 ...127	1	[-]
Priority of requested catalyst heating strategy					
C_NR_PRI_CH_MOD_REQ_TCHA_DIAG	1	80... 7FH	-128 ...127	1	[-]
Priority of requested strategy - at activated catalyst heating function for turbo charger diagnosis					

General Information

This modul merges the STATE_.. values from after start and low load catalyst heating.

Application Conditions

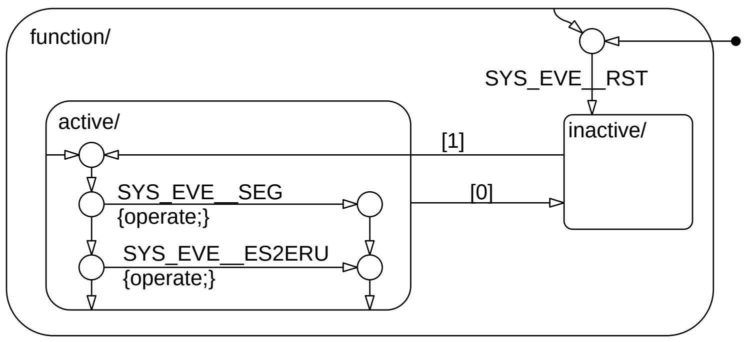
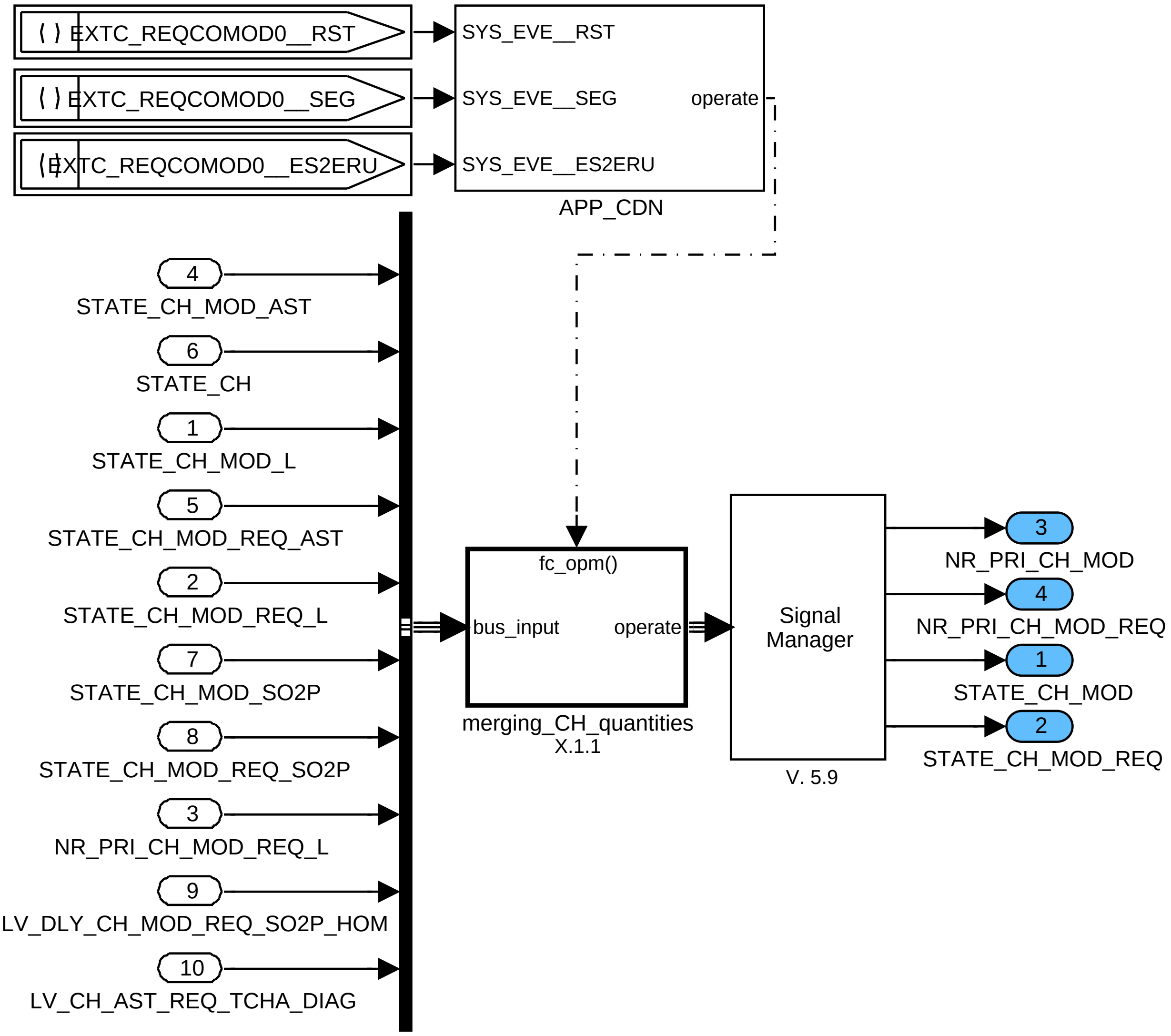


Figure 17:
Path: EXTC_REQCOMOD0/APP_CDN/Chart

Function description



x SDA_SRS / SDA V 5.0.4 30−Mar−2006

Figure 18:
Path: EXTC_REQCOMOD0

30.5.1 Operate Subsystem

30.5.1.1 Generation of STATE_CH_MOD

STATE_CH indicates the actual active catalyst heating state CH_AST, CH_L_LOAD or CH_SO2P. STATE_CH_MOD is set to the corresponding value.

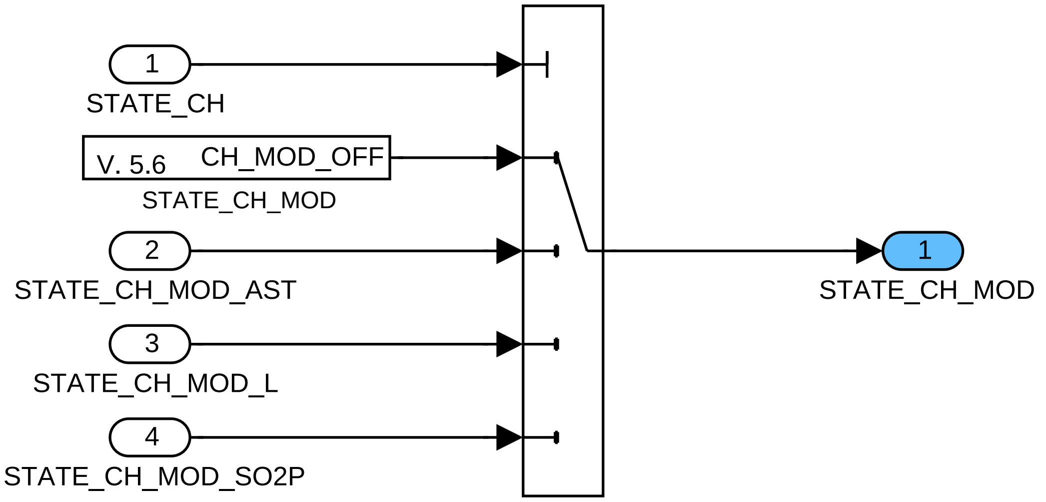


Figure 19:
Path: EXTC_REQCOMOD0/merging_CH_quantities/CLC_STATE_CH_MOD

30.5.1.2 Generation of STATE_CH_MOD_REQ

Same procedere as for STATE_CH_MOD. At LV_DLY_CH_MOD_REQ_SO2P_HOM = 1, it is possible that STATE_CH_MOD_REQ requests homogeneous mode although STATE_CH is CH_OFF. At LV_CH_AST_REQ_TCHA_DIAG = 1, MPLH is requested.

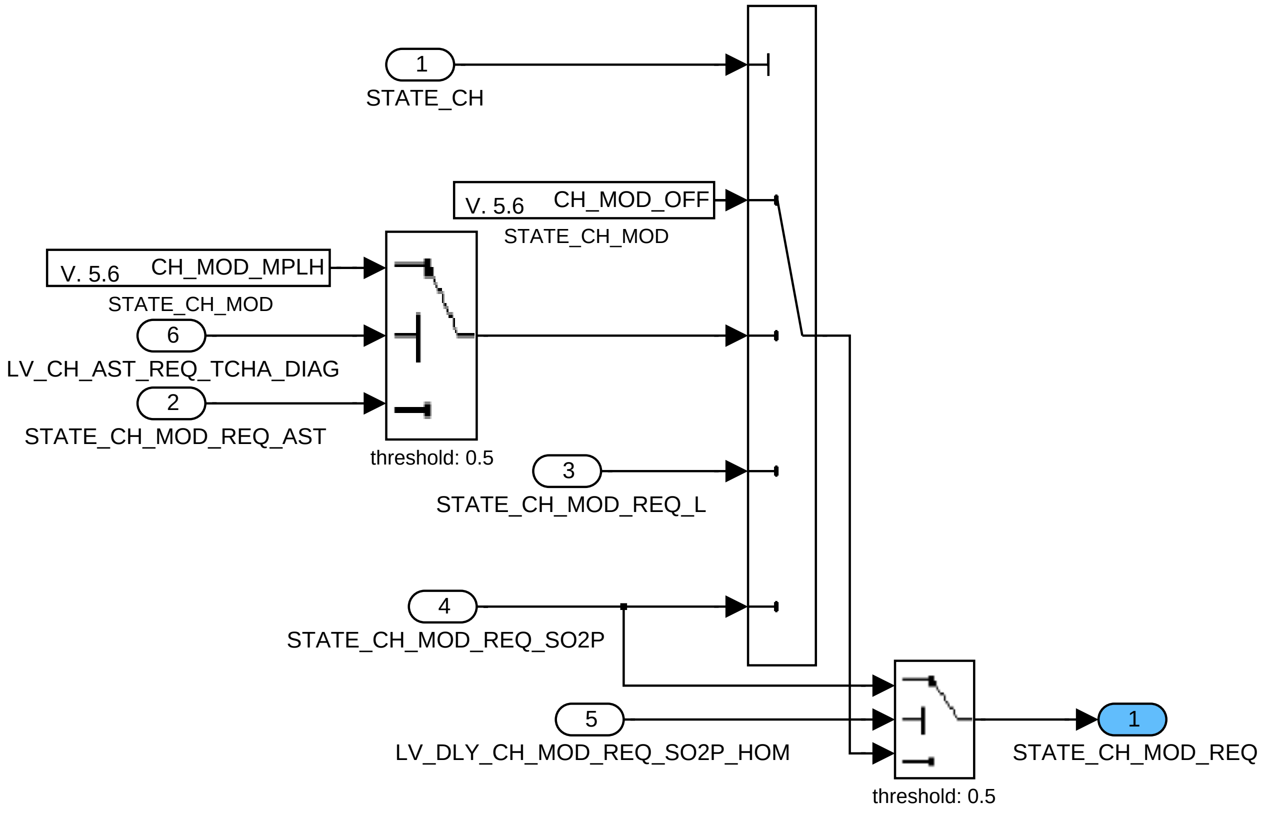


Figure 20:
Path: EXTC_REQCOMOD0/merging_CH_quantities/CLC_STATE_CH_MOD_REQ

30.5.1.3 Generation of priority for operation mode requests

At active turbo diagnosis a separate value is used.

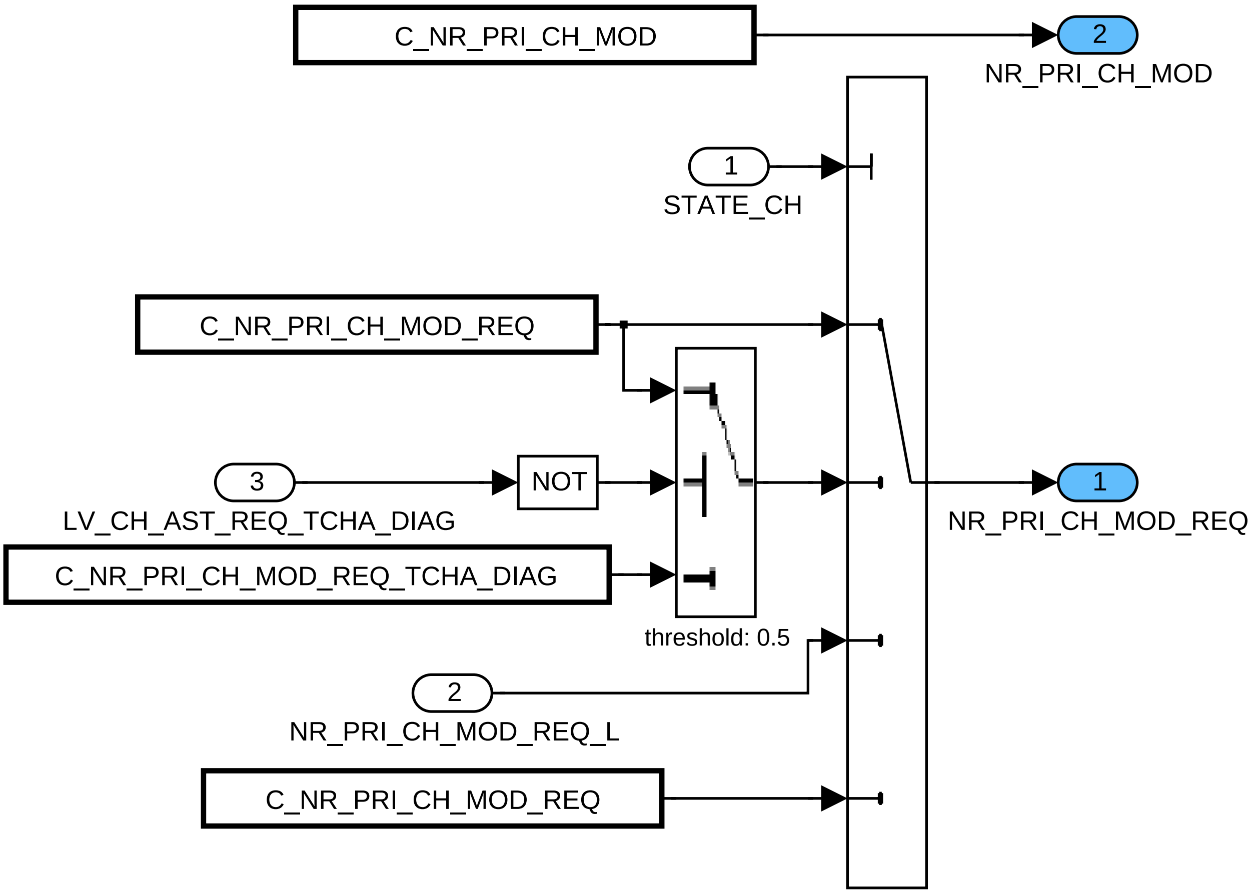


Figure 21:
Path: EXTC_REQCOMOD0/merging_CH_quantities/CLC_NR_PRI

30.6 Catalyst heating after start strategy

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CH_AST_MPLH_CDN	V	0... 1H	0... 1	1	[-]
Indicates if conditions for after start catalyst heating with multiple injection in homogenous are fulfilled					
LV_CH_AST_MPLP_CDN	V	0... 1H	0... 1	1	[-]
Indicates if conditions for after start catalyst heating with post injection in stratified are fulfilled					
STATE_CH_MOD_AST	O/V	0 1 2 3 4	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	-	[-]
Catlyst heating strategy					
STATE_CH_MOD_REQ_AST	O/V	0 1 2 3 4	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	-	[-]
Requested CH-strategy (combustion mode) for after start catalyst heating					
T_AST_COR_CH	O/V	0... FFFFH	0... 6553.5	0.1	[s]
Catalyst age depending correction of T_AST for after start catalyst heating					

Input Data:

STATE_CH	LV_HOM_ACT	LV_MPLH_ACT	LV_S_ACT
LV_MPLP_ACT	T_AST	TQI_REQ_FAST	N_32
TCO_ST	VS	LV_S_ENA	LV_HOM_ENA
LV_HOMS_ACT	PV	LV_ST_END	CTR_KM_CAN
LV_ES			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CH_MOD_MAN	1	0 1 2 3 4	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	-	[-]
Manual setting of catalyst heating strategy					
C_N_MAX_MPLH_CH_AST	1	0... FFH	0... 8160	32	[rpm]
Maximum engine speed for after start catalyst heating in homogeneous mode with split injection					
C_N_MAX_MPLP_CH_AST	1	0... FFH	0... 8160	32	[rpm]
Maximum engine speed for after start catalyst heating in stratified mode with post injection					
C_N_MIN_MPLH_CH_AST	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for after start catalyst heating in homogeneous mode with split injection					
C_N_MIN_MPLP_CH_AST	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for after start catalyst heating in stratified mode with post injection					
C_T_MIN_PV_THD_MPLH_CH_DEAC	1	0... FFFFH	0... 6553.5	0.1	[s]
Minimum time to deactivate MPLH_CH after exceeded pedal value					
C_TQI_MAX_MPLP_CH_AST	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Maximum of indicated torque request for after start catalyst heating with post injection					
C_TQI_MIN_MPLP_CH_AST	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]

Minimum of indicated torque request for after start catalyst heating with post injection					
C_VS_MAX_MPLP_CH_AST	1	0... FFH	0... 255	1	[km/h]
Maximum vehicle speed for post injection during after start catalyst heating					
ID_CH_MOD_BAS	8*8	0... 3H	0... 3	1	[-]
LDPM_T_AST_COR_CH_1_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDP_TCO_ST_ID_CH_MOD_BAS	8	0... FEH	-48... 142.5	0.75	[°C]
Basic catalyst heating strategy					
IP_PV_THD_MPLH_CH_AST	8	0... 3FFH	0... 99.90234375	0.0976563	[%]
LDP_TCO_ST_IP_PV_THD_MPLH_CH	8	0... FEH	-48... 142.5	0.75	[°C]
Pedal value threshold for multiple injection in homogeneous for after start catalyst heating					
IP_T_AST_COR_CH	6	0... FFFFH	0... 6553.5	0.1	[s]
LDP_CTR_KM_CAN_IP_T_AST_COR_CH	6	0... FFFFH	0... 655350	10	[km]
Catalyst age (in fact driven km) corrected time after start for after start catalyst heating					
LC_CH_MOD_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of catalyst heating strategy					

General Information

The engine states during catalyst heating can be homogeneous (HOM), homogeneous with multiple injection (MPLH), stratified with post injection (MPLP) and homogeneous-stratified (HOMS). STATE_CH_MOD_AST, generated in this module, contains the information which catalyst heating strategy is active and is used for CH-setpoint calculation (for example fuel mass setpoint or lambda). Further the request which combustion mode is the most efficient for catalyst heating is generated in this module.

Application Conditions

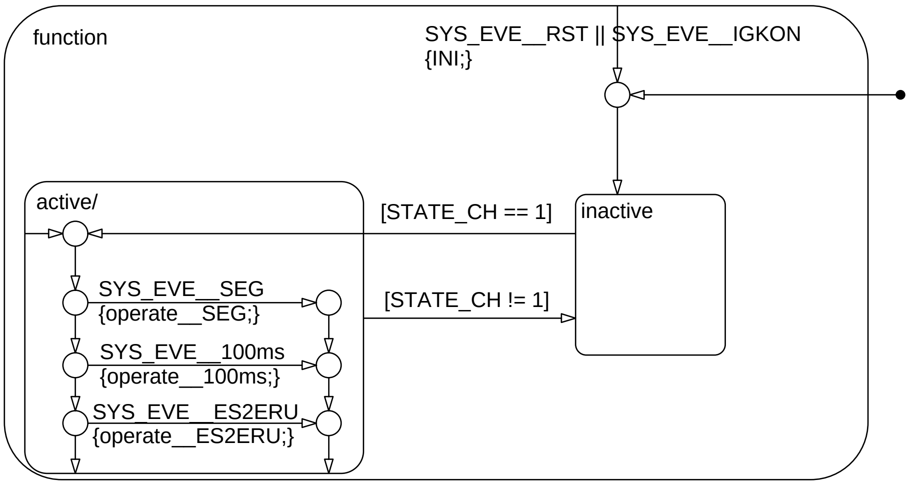
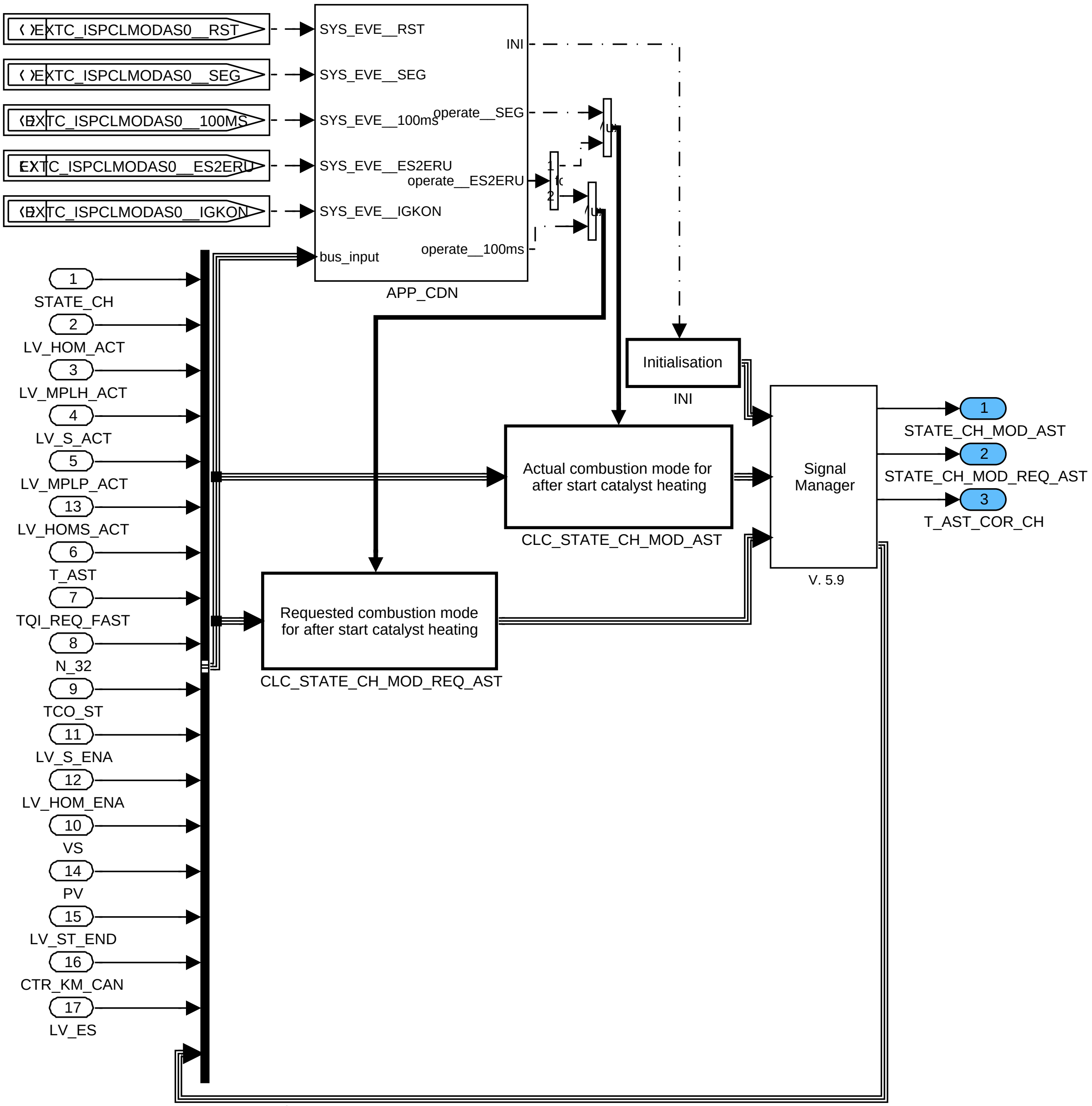


Figure 22:
Path: EXTC_ISPCLMODAS0/APP_CDN/Chart

Function description



x

Figure 23:
Path: EXTC_ISPCLMODAS0

30.6.1 Initialisation

30.6.1.1 No title given

The outputs are initialised at reset and function deactivation.

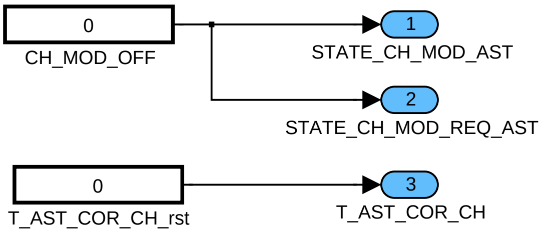


Figure 24:
Path: EXTC_ISPCLMODAS0/INI/SUB

30.6.2 Actual combustion mode for after start catalyst heating

30.6.2.1 No title given

STATE_CH_MOD_AST is set according to the bit-variables that show the actual mode (HOM, MPLH, HOMS or MPLP). If no mode is requested for catalyst heating (STATE_CH_MOD_REQ_AST = 0) also the actual mode is set to 0. It is also possible to set STATE_CH_MOD_AST manually.

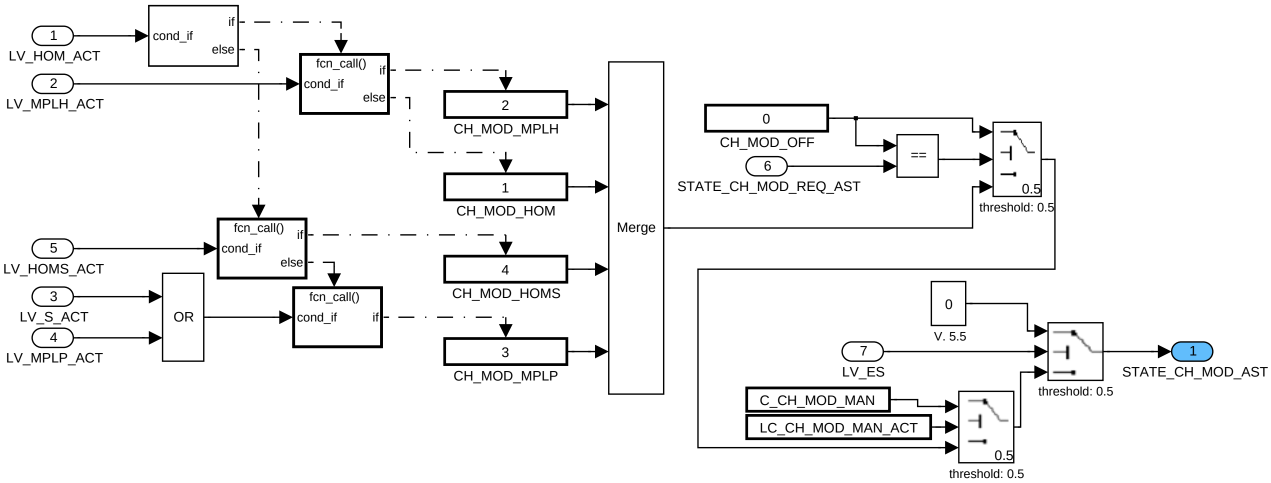


Figure 25:
Path: EXTC_ISPCLMODAS0/CLC_STATE_CH_MOD_AST/SUB

30.6.3 Requested combustion mode for catalyst heating

30.6.3.1 No title given

ID_CH_MOD_BAS contains the strategy-order for catalyst heating.

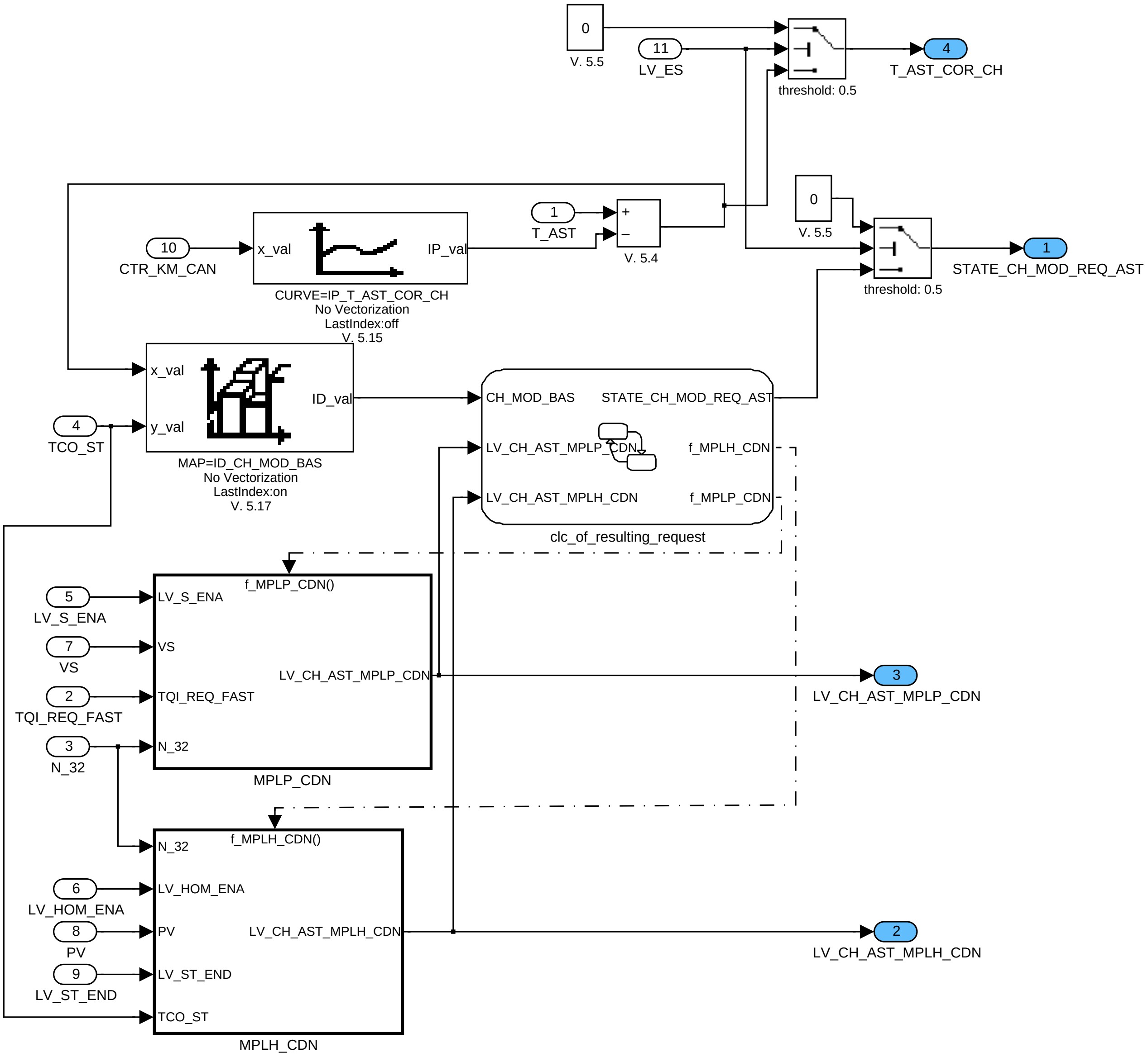
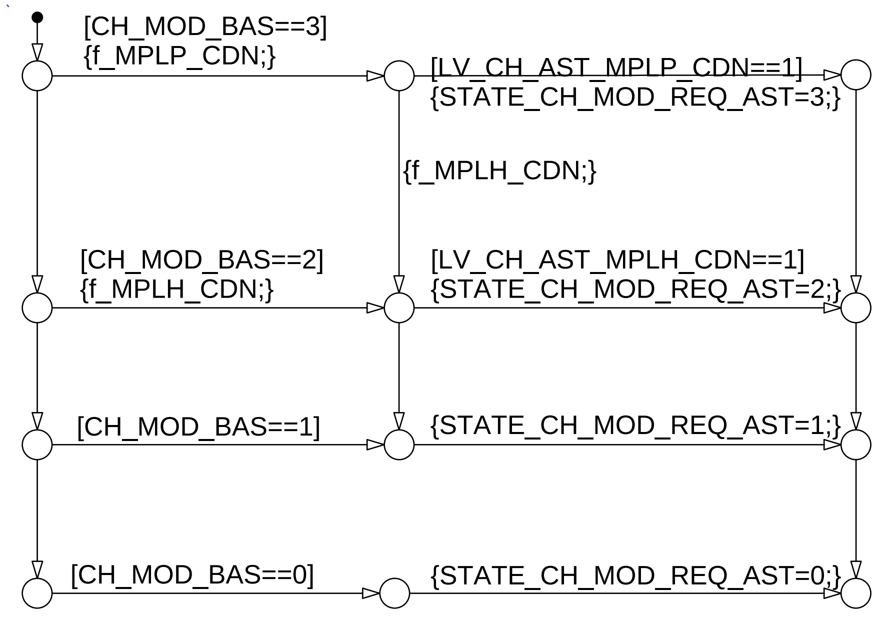


Figure 26:
Path: EXTC_ISPCLMODAS0/CLC_STATE_CH_MOD_REQ_AST/SUB

30.6.3.1.1 clc_of_resulting_request



Via function−call (e.g. f_MPLP_CDN) some conditions for the "optimal" combustion mode CH_MOD_BAS are checked. If this conditions (s. next subsystems) are not fulfilled, the next mode of the priority−order MPLP−MPLH−HOM is checked.

Figure 27:
Path: EXTC_ISPCLMODAS0/CLC_STATE_CH_MOD_REQ_AST/SUB/clc_of_resulting_request

30.6.3.1.2 Conditions for homogenous mode with multiple injection

LV_CH_AST_MPLH_CDN indicates if conditions for MPLH for catalyst heating are fulfilled. It is set to 0 if the engine speed is not in-between a calibratable range, LV_HOM_ENA is 0 or the pedal value exceeds C_PV_MAX_MPLH_CH for a certain time (C_T_MIN_PV_THD_MPLH_CH_DEAC). The bit LV_CH_AST_MPLH_CDN can only be set to 1 again after next engine start.

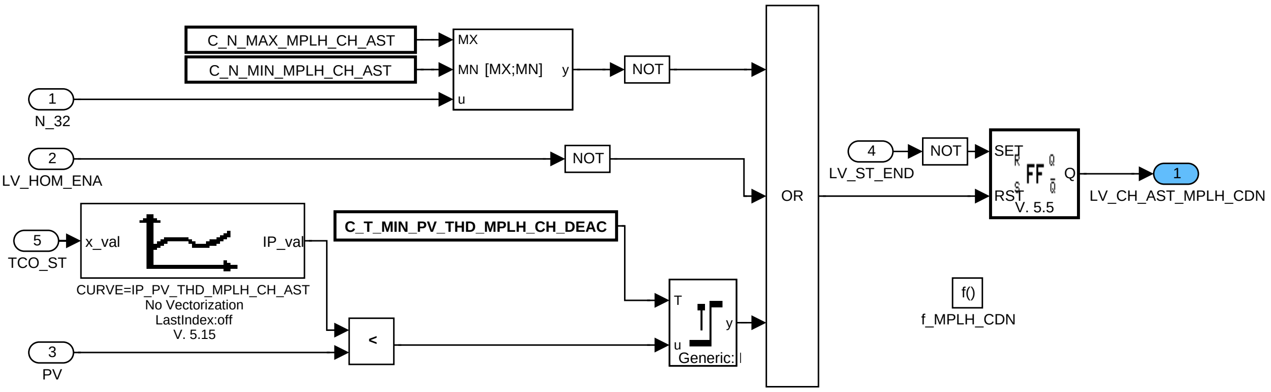


Figure 28:
Path: EXTC_ISPCLMODAS0/CLC_STATE_CH_MOD_REQ_AST/SUB/MPLH_CDN

30.6.3.1.3 Conditions for stratified mode

For stratified mode with post injection the torque request and the engine speed has to be in-between a calibratable range, the vehicle speed must be below a limit and LV_S_ENA must indicate that stratified mode is enabled.

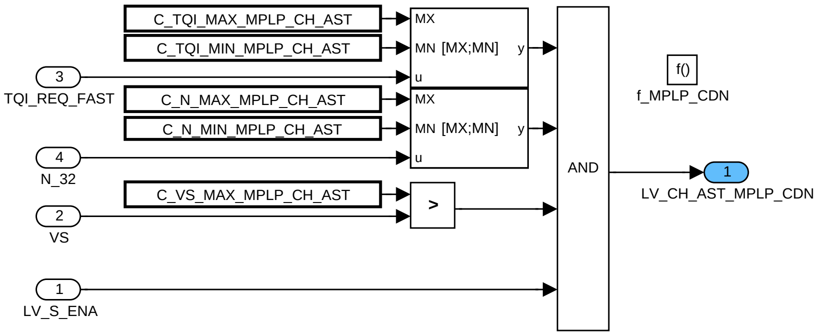


Figure 29:
Path: EXTC_ISPCLMODAS0/CLC_STATE_CH_MOD_REQ_AST/SUB/MPLP_CDN

30.7 Catalyst heating low load strategy

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CH_MOD_L	O/V	0H 1H 2H 3H 4H	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	1	-
Catalyst heating strategy for low load heating					
STATE_CH_MOD_REQ_L	O/V	0H 1H 2H 3H 4H	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	1	-
Requested CH-strategy (combustion mode) for low load catalyst heating					
NR_PRI_CH_MOD_REQ_L	O/V	80...7FH	-128...127	1	-
Priority of requested catalyst heating strategy for low load catalyst heating					

Input data:

STATE_CH	LV_S_ACT	LV_MPLP_ACT	C_CH_MOD_MAN
LC_CH_MOD_MAN_ACT	TEMP_CAT	TNT_MDL_H	LV_HOMS_ACT

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NR_PRI_CH_MOD_REQ_L_ACT	1	80...7FH	-128...127	1	-
Priority of requested catalyst heating strategy for switching to homogeneous mode					
C_NR_PRI_CH_MOD_REQ_L_PAS	1	80...7FH	-128...127	1	-
Priority of passive requested catalyst heating strategy in low load catalyst heating					
C_TEMP_CAT_MIN_SWI_HOMS_CH_L	1	0...FFFFH	-33...990.984375	0.015625	°C
Below this catalyst temperature the homogeneous-stratified mode is requested for low load catalyst heating					
C_TEMP_CAT_MIN_SWI_HOM_CH_L	1	0...FFFFH	-33...990.984375	0.015625	°C
Below this catalyst temperature the homogeneous mode is requested for low load catalyst heating					
C_TEMP_HYS_SWI_HOMS_CH_L	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Hysteresis of temperature threshold for switching in homogenous-stratified mode in low load catalyst heating					
C_TEMP_HYS_SWI_HOM_CH_L	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Hysteresis of temperature threshold for switching in homogenous mode in low load catalyst heating					
C_TNT_MIN_SWI_HOMS_CH_L	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Below this NOx-trap temperature the homogeneous-stratified mode is requested for low load catalyst heating					
C_TNT_MIN_SWI_HOM_CH_L	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Below this NOx-trap temperature the homogeneous mode is requested for low load catalyst heating					

30.7.1 General information

This module has to be calculated after “Catalyst heating coordination” and “Catalyst heating after start strategy” and before any other catalyst heating module.

The output STATE_CH_MOD_L will be merged in “Catalyst heating strategy” into STATE_CH_MOD. This value is used to trigger setpoint calculation modules. STATE_CH_MOD_REQ_L will be merged into STATE_CH_MOD_REQ.

Application Condition

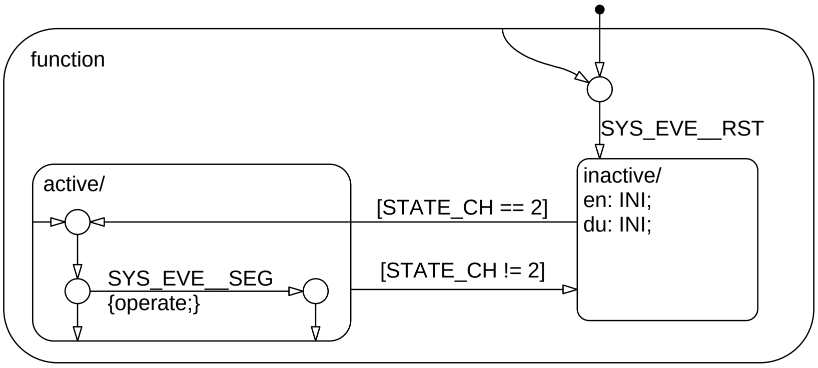
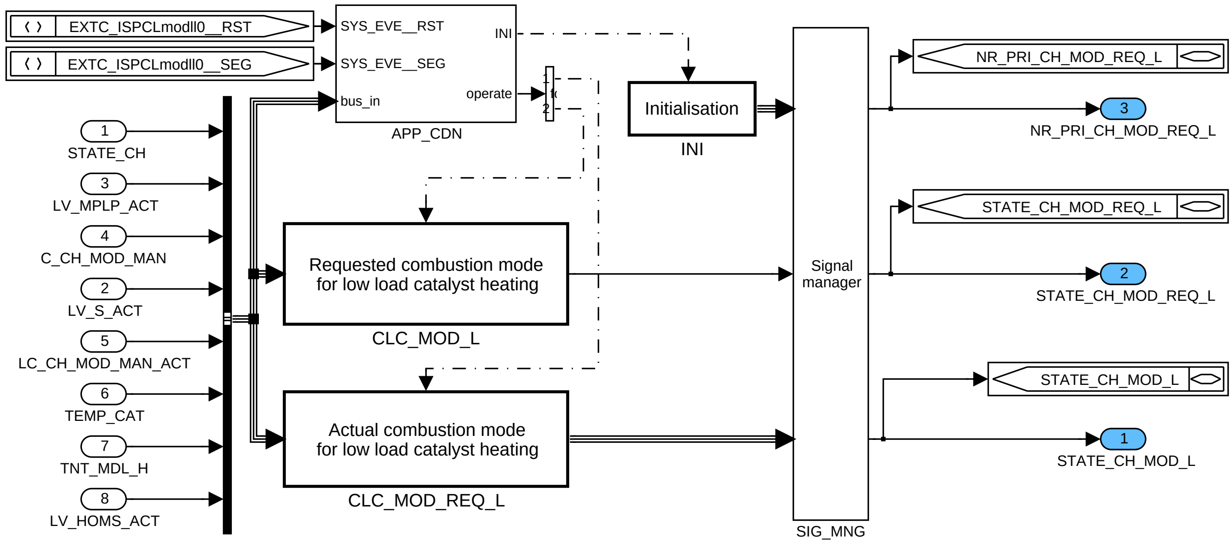


Figure 30 EXTC_ISPCLmodll0/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 22−Jun−2005

Figure 31 EXTC_ISPCLmodll0

30.7.1.1 Initialization

All outputs are set to 0 at reset and deactivated function.

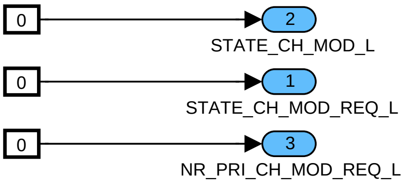


Figure 32 EXTC_ISPCLmodll0/ INI/ INI_SUB

30.7.1.2 Actual combustion mode for low load catalyst heating

Depending on the actual active combustion mode, STATE_CH_MOD_L is set to HOMS, MPLP or HOM. With (L)C_CH_MOD_MAN(_ACT) it is possible to set it manually.

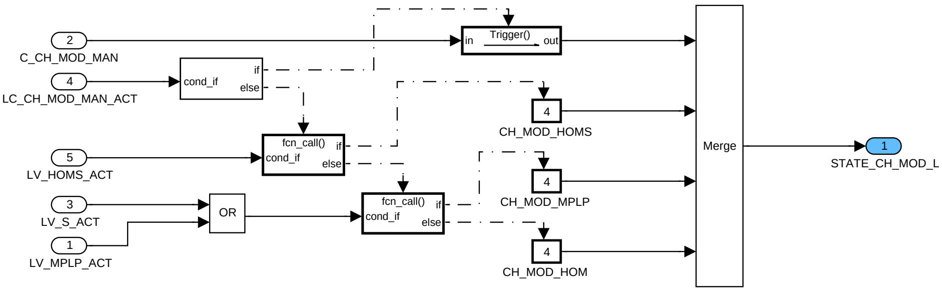


Figure 33 EXTC_ISPCLmodll0/ CLC_MOD_L/ CLC_MOD_L_SUB

30.7.1.3 Requested combustion mode for low load catalyst heating

If catalyst or NOx-Trap temperature falls below C_TEMP_CAT_MIN_SWI_HOMS_CH_L or C_TNT_MIN_SWI_HOMS_CH_L the HOMS – mode is requested to inhibit that the catalyst cools down under the working temperature. If one of the temperatures falls even below C_TEMP_CAT_MIN_SWI_HOM_CH_L or C_TNT_MIN_SWI_HOM_CH_L homogenous mode is requested. NR_PRI_CH_MOD_REQ_L is increased via C_NR_PRI_CH_MOD_REQ_L_ACT if the request shell be active.

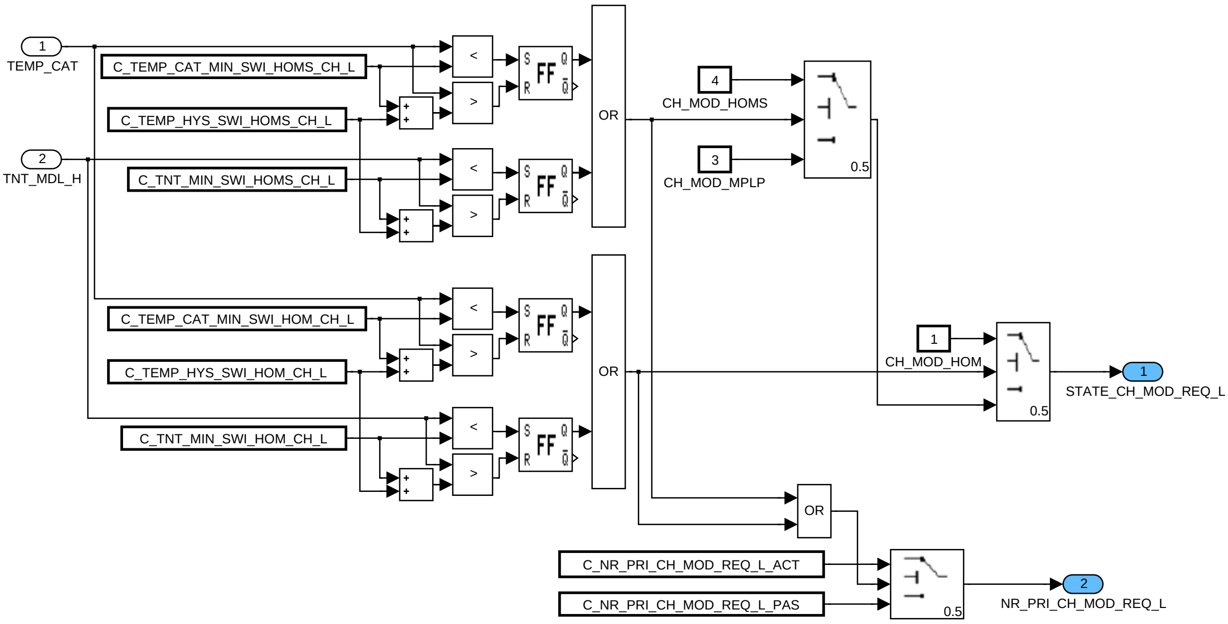


Figure 34 EXTC_ISPCLmodll0/ CLC_MOD_REQ_L/ CLC_MOD_REQ_L_SUB

30.8 Catalyst heating for desulfation strategy

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_DLY_CH_MOD_REQ_SO2P_HOM	O/V	0... 1H	0... 1	1	[-]
Indicates if homogeneous mode is requested after active desulfation catalyst heating					
STATE_CH_MOD_REQ_SO2P	O/V	0 1 2 3 4	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	-	[-]
Requested CH-strategy (combustion mode) for desulfation catalyst heating					
STATE_CH_MOD_SO2P	O/V	0 1 2 3 4	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH CH_MOD_MPLP CH_MOD_HOMS	-	[-]
Catalyst heating strategy for desulfation catalyst heating					
TNT_DIF_CH_SO2P	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[K]
Difference between desulfation setpoint and actual value of TNT					

Input Data:

STATE_CH	LV_S_ACT	TNT_MDL_L	N_32
MAF	LV_MPLH_ACT	LV_NT_SO2P_EXT_ADJ_ACT	LV_STATE_CH_SO2P_ENA

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_MAF_SO2P_ACT_HYS	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Mass air flow hysteresis to avoid toggle of switch over the SO2P heating request					
C_N_32_SO2P_ACT_HYS	1	0... FFH	0... 8160	32	[rpm]
Engine speed hysteresis to avoid toggle of switch over the SO2P heating request					
C_STATE_CH_MOD_SO2P_MAN	1	0 1 2	CH_MOD_OFF CH_MOD_HOM CH_MOD_MPLH	-	[-]
Manual setting of catalyst heating strategy for desulfation					
C_T_DLY_STATE_CH_MOD_REQ_SO2P	1	0... FFFFH	0... 6553.5	0.1	[s]
Time to request homogeneous mode after deactivation of desulfation catalyst heating					
C_TNT_SP_CH_SO2P	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Setpoint of TNT for desulfation catalyst heating					
C_TNT_SP_CH_SO2P_EXT_ADJ	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Setpoint of TNT for desulfation catalyst heating activated by external adjustment					
ID_STATE_CH_MOD_REQ_SO2P	6*6	0... 2H	0... 2	1	[-]
LDP_N_32_ID_CH_MOD_REQ_SO2P	6	0... FFH	0... 8160	32	[rpm]
LDP_MAF_ID_CH_MOD_REQ_SO2P	6	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Requested combustion mode for desulfation catalyst heating					
LC_STATE_CH_MOD_SO2P_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of catalyst heating strategy for desulfation					

General Information

general specification

This module has to be calculated after Catalyst heating coordination , Catalyst heating after start strategy , Catalyst heating low load strategy and before any other catalyst heating module.

The engine states during catalyst heating for desulfation can be homogeneous (HOM) and homogeneous with multiple injection (MPLH). STATE_CH_MOD_SO2P, generated in this module, contains the information which catalyst heating strategy is active and is used for CH-setpoint calculation (for example fuel mass setpoint or lambda). It will be merged in Catalyst heating strategy into STATE_CH_MOD. STATE_CH_MOD_REQ_SO2P will be merged into STATE_CH_MOD_REQ. Further a request is generated that contains the most efficient combustion mode for desulfation catalyst heating.

Application Conditions

Initialization: RST
Activation: 100MS: STATE_CH == 3
SEG: STATE_CH == 3
Deactivation: STATE_CH != 3 && STATE_CH_MOD_REQ_SO2P == 0

Function description

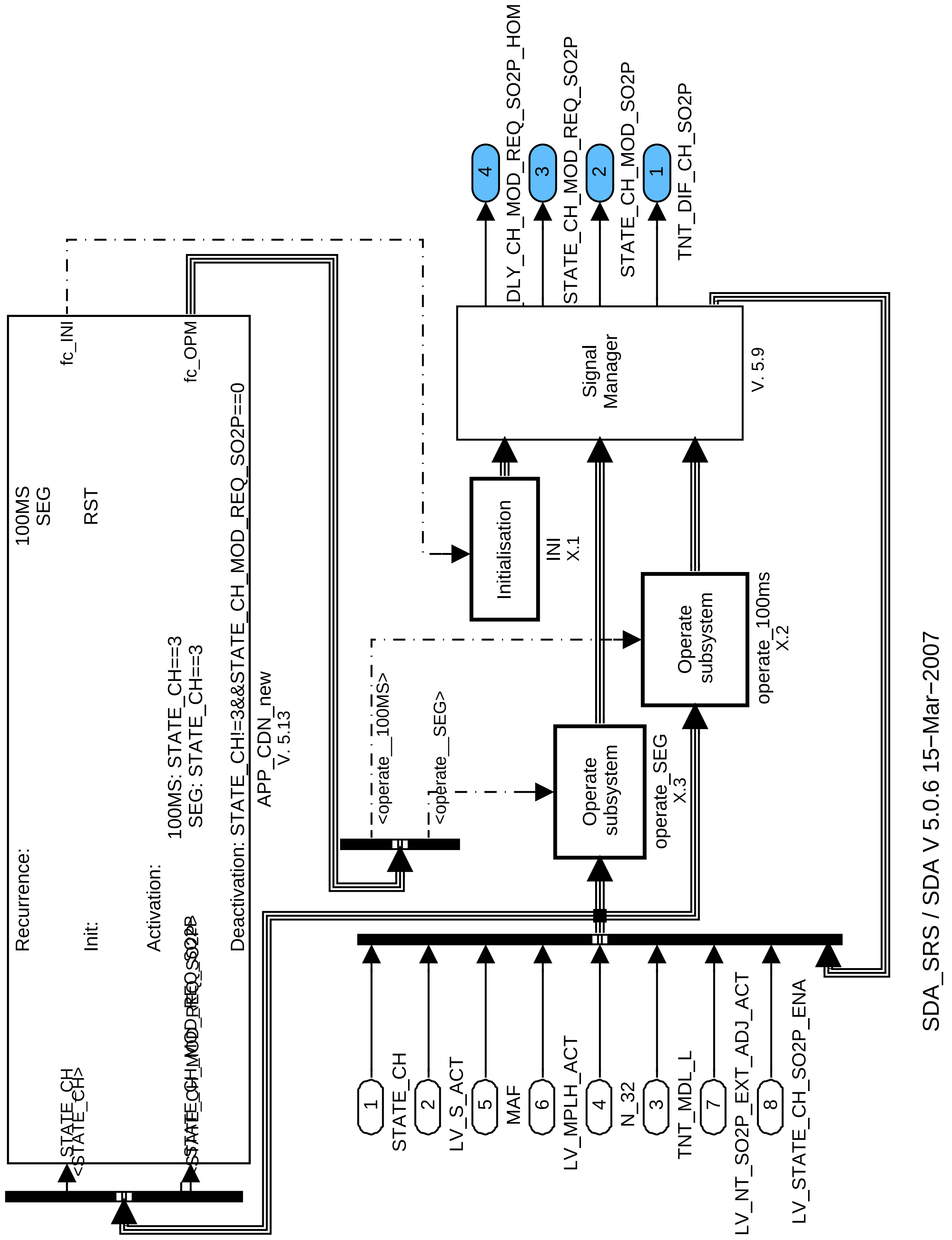


Figure 35:
Path: EXTC_ISPCLMODSO0

30.8.1 Initialisation

30.8.1.1 Initialisation - Subsystem

The outputs are initialised at reset and function deactivation.

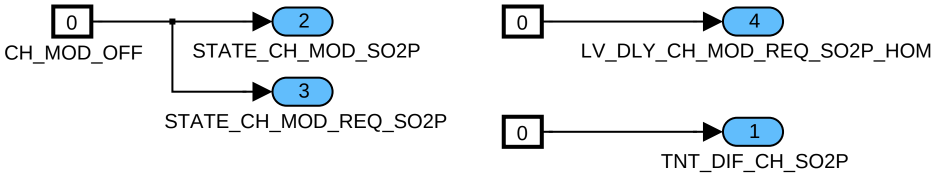


Figure 36:
Path: EXTC_ISPCLMODSO0/INI/SUB

30.8.2 100 ms - Task

30.8.2.1 100 ms - Task - Subsystem

At deactivation of desulfation catalyst heating due to torque or vehicle speed threshold (STATE_CH == 0 and LV_STATE_CH_SO2P_ENA=1) there is the possibility to request homogenous mode for a calibratable time.

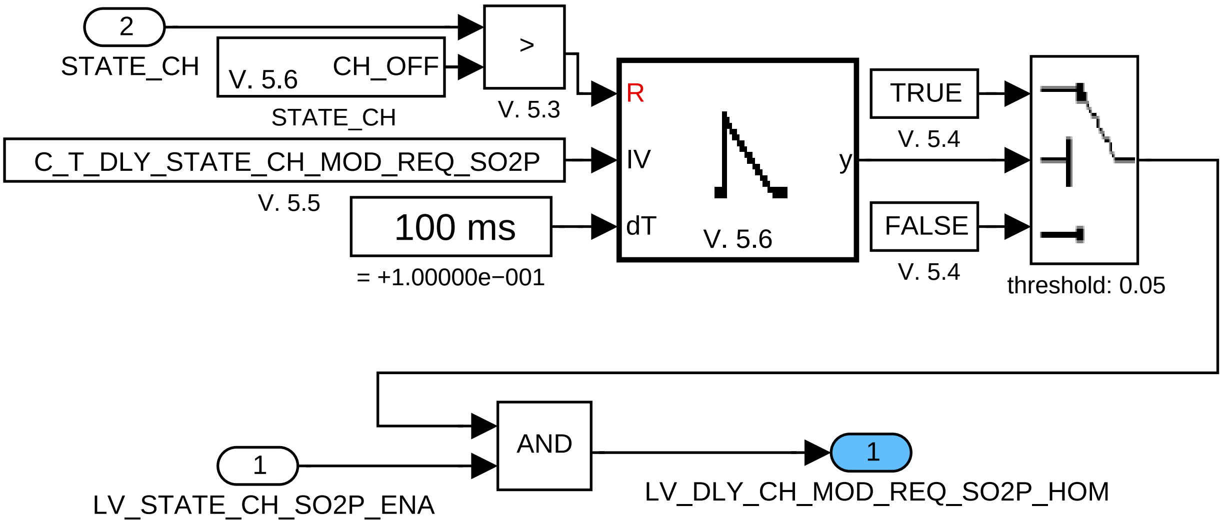


Figure 37:
Path: EXTC_ISPCLMODSO0/operate_100ms/SUB

30.8.3 Segment synchronous task

30.8.3.1 Segment synchronous task - Subsystem

TNT_DIF_CH_SO2P is the difference between the NOx-trap temperature setpoint and the actual value. It is negative if there is necessity to heat. The ID and the calibration data for manual setting have other physical meaning as the output of the spec because for catalyst heating for desulfation only HOM and MPLH can be used.

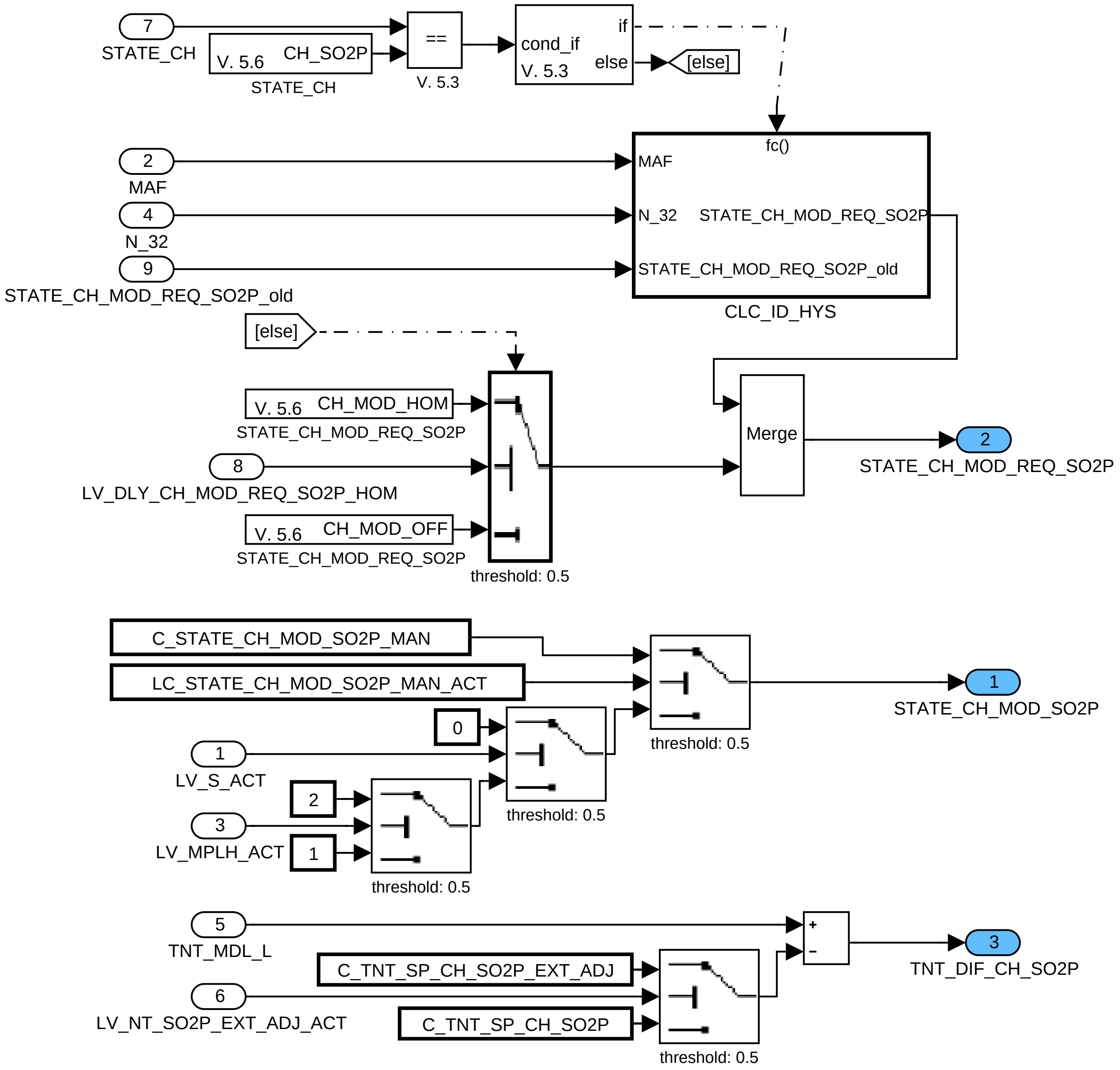


Figure 38:
Path: EXTC_ISPCLMODSO0/operate_SEG/SUB

30.8.3.1.1 EXTC_ISPCLMODSO0/OPERATE_SEG/SUB/CLC_ID_HYS

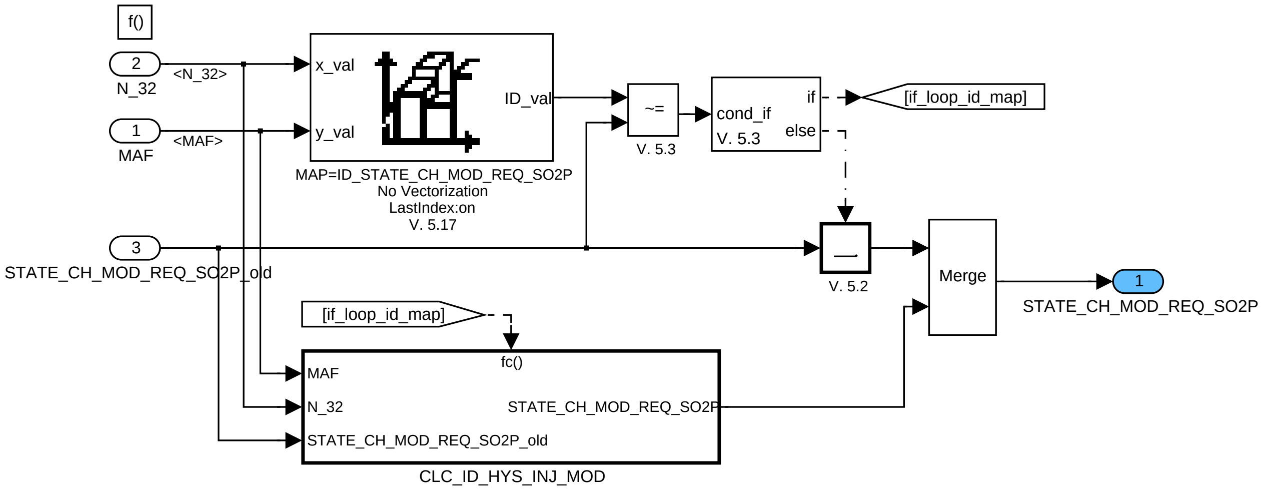


Figure 39:
Path: EXTC_ISPCLMODSO0/operate_SEG/SUB/CLC_ID_HYS

30.8.3.1.1.1 Change of injection mode

The basic for the changing of the injection mode is the hysteresis of N_32 and MAF to avoid the toggle between the injection modes. The actual values of N_32 and MAF are detected by the corresponding breakpoints. These indexes are going to the map ID_STATE_CH_MOD_REQ_SO2P. The hysteresis is calculated with the value of the corresponding breakpoint and the calibration constant. The changing of the injection mode is only possible after reaching the maximum threshold of the hysteresis. Example: Actual engine speed is 3000rpm. Corresponding breakpoint is 1632rpm. Calibration value is 200rpm. Result of the hysteresis is 1632rpm + 200rpm = 1832rpm. The calculation of MAF is the same.

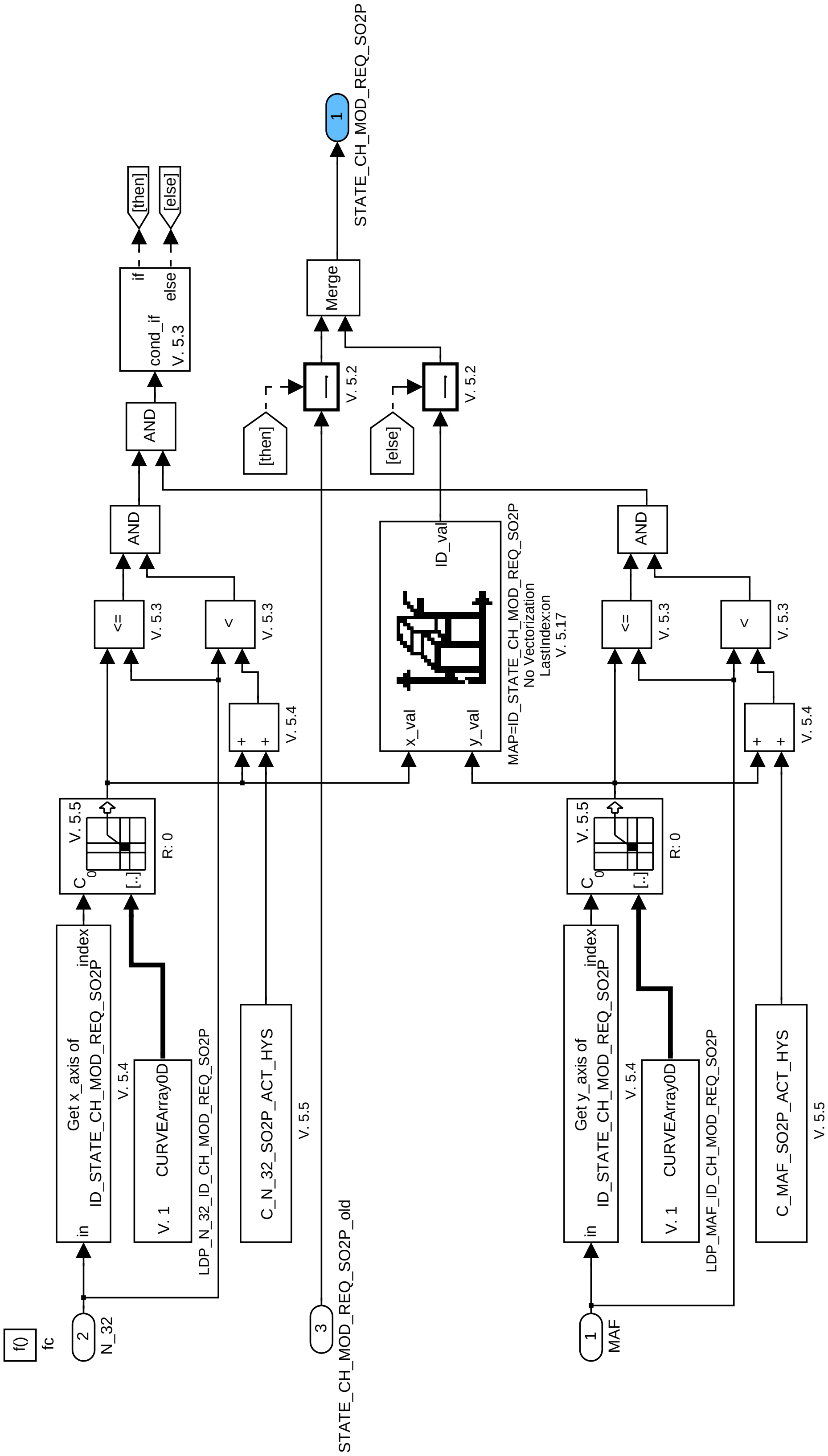


Figure 40:
Path: EXTC_ISPCLMODSO0/operate_SEG/SUB/CLC_ID_HYS/CLC_ID_HYS_INJ_MOD

30.9 Minimum IGA limitation for exhaust gas temperature protection

Output data:

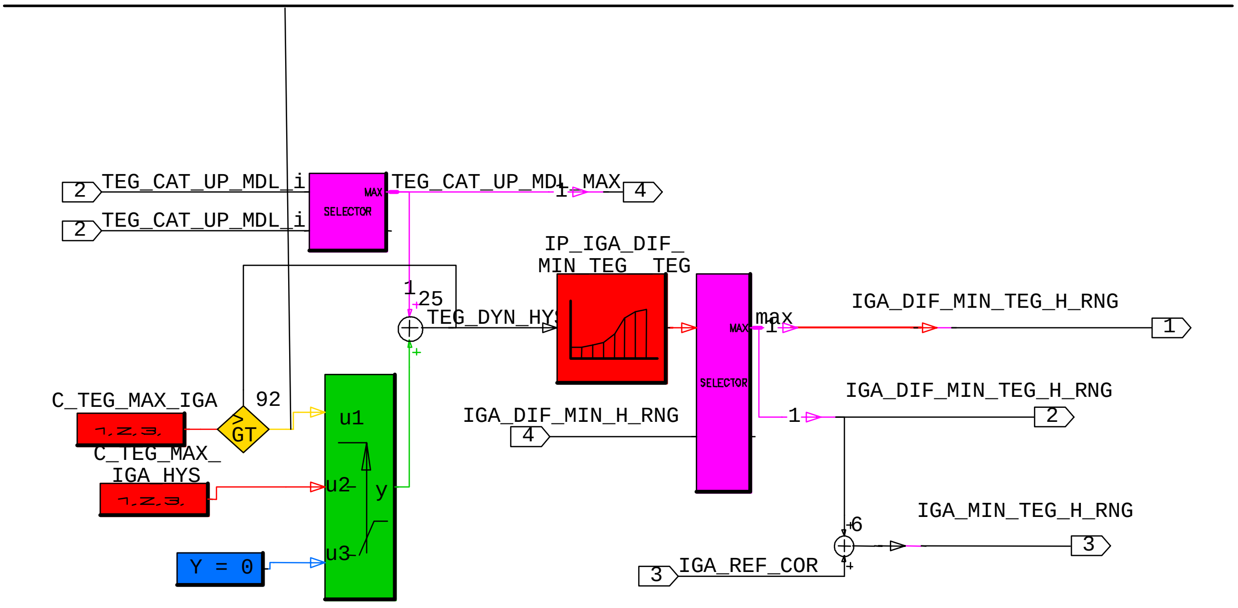
Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_MIN_TEG_H_RNG	V/O	FA60h...5A0h	-90...90	0.0625	[°CRK]
minimum ignition angle					
IGA_DIF_MIN_TEG_H_RNG	V/O	0h...B40h	0...-180	0.0625	[°CRK]
Difference of minimum ignition angle with reference one - TEG correction included					
IGA_DIF_MIN_TEG_1_H_RNG	-	0h...B40h	0...-180	0.0625	[°CRK]
Difference of minimum ignition angle with reference one according to TEG value - Intermediate variable					
TEG_DYN_HYS	V	0...7FF0H	0...2047	0.0625	[°C]
Temperature of exhaust gas with hysterisis for IGA_MIN limitation					
TEG_CAT_UP_MDL_MAX	V/O	0...7FF0H	0...2047	0.0625	[°C]
Maximal Temperature of exhaust gas between all cylinder bank					

Input data:

LV_HOM_RUN	LV_ST	LV_ES	TEG_CAT_UP_MDL[NC_CBK_EX_NR]
IGA_DIF_MIN_H_RNG	IGA_REF_COR		

FUNCTION DESCRIPTION:

Discrete SuperBlock	Sample Period	Sample Skew	Inputs	Outputs	Enable Signal
IGA_MIN_for_TEG	1.	0.	4	4	Parent



General information:

The minimum ignition angle IGA_MIN is the minimum allowable ignition angle that is possible to apply to the engine. Hencce, it only includes the combustion stability concern. It does not take into account any thermal effect. It should be used during the torque intervention phases in order to produce a fast reduction of the torque.
Applying IGA_MIN as actual ignition angle results in efficiency losses that should lead to an exhaust gas overheating and/or engine damages.

In order to avoid overheating of the components in the exhaust-system branch, minimum spark retard has to be limited in accordance with the actual exhaust gas temperature TEG_CAT_UP_MDL[NC_CBK_EX_NR].

Moreover to determine the maximum possible torque reserve for catalyst heating, the torque reserve co-ordination module needs the minimum ignition angle efficiency and the minimum ignition angle with exhaust branch protection efficiency in order to limit and choose the different torque reserve requests.

Hence another minimum spark retard difference have to be calculated IGA_DIF_MIN_TEG_H_RNG which takes into account the exhaust gas system protection. It is the maximum between the one calculated versus the maximal exhaust gases temperature (TEG_CAT_UP_MDL_MAX) and the basic one IGA_DIF_MIN_H_RNG.
Classically IGA_DIF_MIN is calculated from a basic value of the minimum ignition angle corrected with a temperature dependent additive term. The basic minimum spark advance is here extracted from a calibration map.
To calculate the minimum spark retard limitation according to the exhaust gas temperature the map IP_IGA_DIF_MIN_TEG_H_RNG__TEG is used. TEG_DYN_HYS is used as input for this map. This intermediate variable TEG_DYN_HYS represents the exhaust gas temperature TEG_CAT_UP_MDL[NC_CBK_EX_NR] increased by an hysterisis in order to anticipate the exhaust-system branch overheat.

The influence of the ignition angle on the exhaust gas temperature results from the ignition angle efficiency. Thus - in order to calibrate easily the map IP_IGA_DIF_MIN_TEG_H_RNG__TEG - one should use the efficiencies resulting from IGA_DIF_MIN_H_RNG values with the map IP_EFF_IGA__IGA_DIF and limit these efficiencies according to the actual exhaust gas temperature to avoid exhaust-system branch overheat. Then, thanks to the reverse map IP_IGA_DIF__EFF or thanks to the torque model for ignition angle efficiency, it is possible to calculate the value of the map IP_IGA_DIF_MIN_TEG_H_RNG__TEG.

So, in all practical cases IGA_DIF_MIN_H_RNG is a negative value that is the maximum allowable spark retard for engine stable running and pollutant conditions. It includes the coolant temperature correction. The exhaust gas temperature limitation is then taken into account in IGA_DIF_MIN_TEG_H_RNG:

The actual exhaust gas temperature TEG_CAT_UP_MDL[NC_CBK_EX_NR] can be either measured by a sensor on the exhaust-system branch or calculated from an exhaust gas temperature model.

For engines with multiple cylinder bamks, this temperature TEG_CAT_UP_MDL[NC_CBK_EX_NR] of exhaust gas before catalyst is defined and dedicated for each cylinder banks individualy. Since here the limitation of IGA_MIN is just a protective function versus the catalyst overheating, only the maximal value from all those cylinder bank related TEG_CAT_UP_MDL[NC_CBK_EX_NR] temperature will be taken into account.

Note: Two different update rates are used in the following calculations: for everything related to TEG only 1000ms rate is used to update calculation in synchronisation with TEG updates; for calculations related to ignition angle a 10 time faster update rate of 100ms.
These calculations are only performed when engine is running (*not stalled*) in homogenous mode after the end of start phase.

Application conditions:

Activation: LV_HOM_RUN = 1 **and** LV_ST = 0 **and** LV_ES = 0
active only in homogenous mode after the end of start phase before engine stall

Deactivation: LV_HOM_RUN = 0 **or** LV_ST = 1 **or** LV_ES = 1

Initialisation:
IGA_MIN_TEG_H_RNG = -90 °CRK
IGA_DIF_MIN_TEG_H_RNG = -180 °CRK
IGA_DIF_MIN_TEG_1_H_RNG = -180 °CRK
TEG_DYN_HYS = 0 at reset

Update Rate:
TEG_DYN_HYS => 1000ms **and** synchronised with TEG_CAT_UP_MDL[NC_CBK_EX_NR] updates.
IGA_DIF_MIN_TEG_H_RNG => 100ms
IGA_MIN_TEG_H_RNG => 100ms
EFF_IGA_MIN_TEG => 100ms

Formula section:

Maximal cylinder bank exhaust gas temperature determination:

every 1000ms
in synchronisation with TEG_CAT_UP_MDL[NC_CBK_EX_NR] updates

$$TEG\_CAT\_UP\_MDL\_MAX = \underset{0}{\overset{NC\_CBK\_NR-1}{Max}} \left(TEG\_CAT\_UP\_MDL\_i\right)$$

Hysteris function on exhaust gas temperature for ignition effects anticipation:

every 1000ms
in synchronisation with TEG_CAT_UP_MDL[NC_CBK_EX_NR] updates

```
If      TEG_DYN_HYS_{n-1}  > C_TEG_MAX_IGA
Then    TEG_DYN_HYS_n      = TEG_CAT_UP_MDL_MAX_n + C_TEG_MAX_IGA_HYS
Else    TEG_DYN_HYS_n      = TEG_CAT_UP_MDL_MAX_n
Endif
```

IGA_DIF_MIN_TEG_1_H_RNG = IP_IGA_DIF_MIN_TEG_H_RNG__TEG (TEG_DYN_HYS)

IGA_DIF_MIN_TEG_1_H_RNG is just an intermediate variable for calculation to name the result of the interpolation

Limitation of Maximum possible spark retard according to exhaust temperature TEG_DYN_HYS:

every 100ms

IGA_DIF_MIN_TEG_H_RNG = Max (IGA_DIF_MIN_H_RNG . IGA_DIF_MIN_TEG_1_H_RNG)

Minimum ignition angle limited for exhaust temperature protection: every 100ms

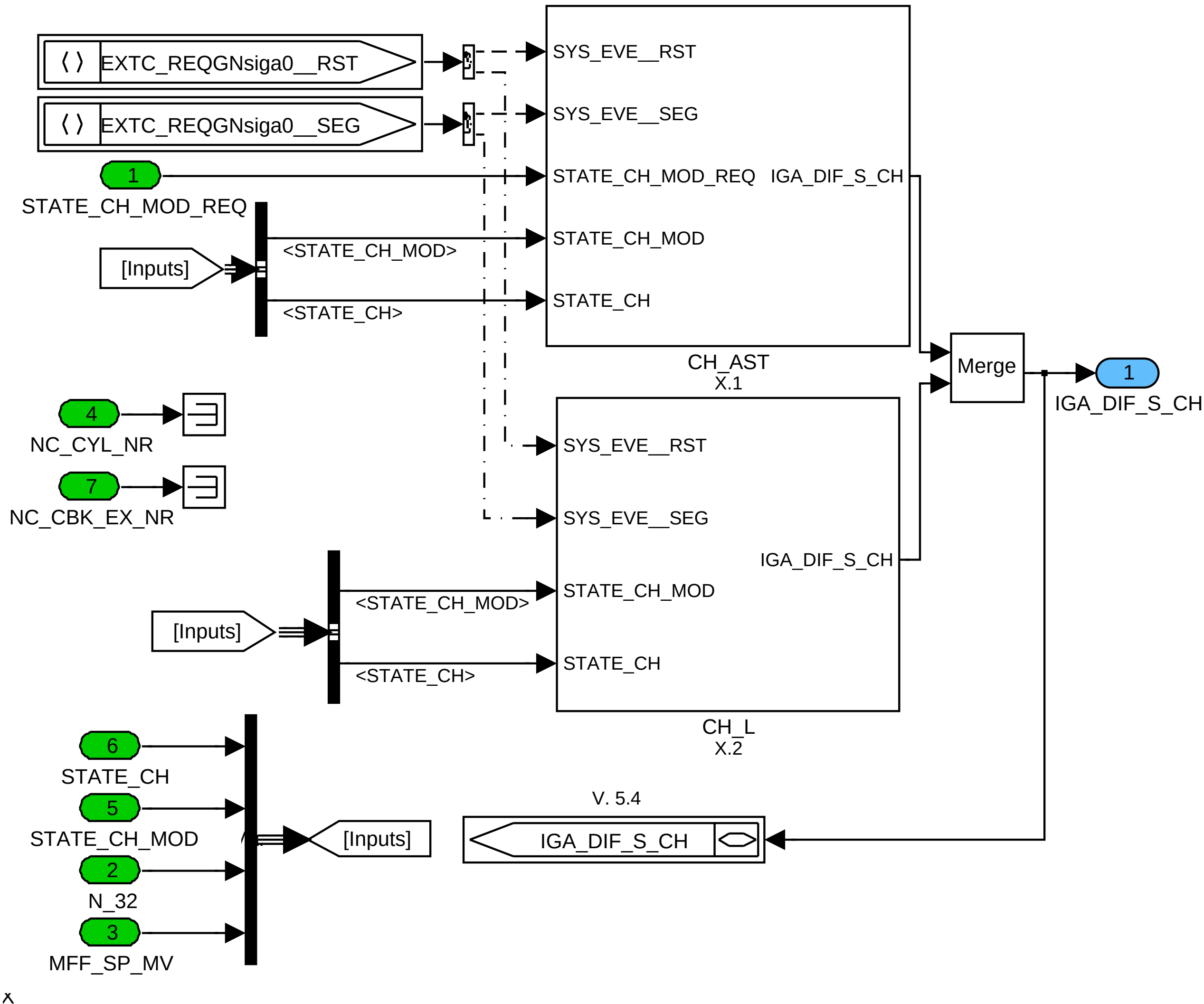
IGA_MIN_TEG_H_RNG = IGA_REF_COR + IGA_DIF_MIN_TEG_H_RNG

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEG_MAX_IGA_HYS	1	0...7FF0H	0...2047	0.0625	[°C]
Hysterisis on exhaust gas temperature for spark retard control					
C_TEG_MAX_IGA	1	0...7FF0H	0...2047	0.0625	[°C]
Maximum allowable exhaust gas temperature for spark retard control					
IP_IGA_DIF_MIN_TEG_H_RNG__TEG	16	0h...B40h	0...-180	0.0625	[°CRK]
LDP_TEG_DYN_HYS	16	0...7FF0H	0...2047	0.0625	[°C]
Minimum ignition angle versus exhaust gas temperature					

30.10 Ignition angle difference for catalyst heating with post injection

Overview



SDA_SRS / SDA V 5.0.4 15−Nov−2006

Figure 41:
Path: EXTC_REQGNsiga0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
IGA_DIF_S_CH	O/V	0... FFH	-35.625 ...60	0.375	[°CRK]
Ignition angle difference for stratified catalyst heating with post injection					

Input Data:

STATE_CH_MOD_REQ	N_32	MFF_SP_MV	NC_CYL_NR
STATE_CH_MOD	STATE_CH	NC_CBK_EX_NR	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_IGA_DIF_ADD_MFF_POST_CH_L	1	0... FFH	-35.625 ...60	0.375	[°CRK]
Mass of post injection depending correction for ignition angle difference for low load catalyst heating					
C_IGA_DIF_S_CH_AST_BAS	1	0... F00H	-720 ...720	0.375	[°CRK]
Ignition angle difference for stratified catalyst hating with post injection					
C_IGA_DIF_S_CH_L_BAS	1	0... F00H	-720 ...720	0.375	[°CRK]
Ignition angle difference for stratified low load catalyst heating with post injection					
C_IGA_DIF_S_CH_MAN	1	0... FFH	-35.625 ...60	0.375	[°CRK]
Manual setting of ignition angle difference for stratified catalyst heating with post injection					
LC_IGA_DIF_S_CH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of ignition angle difference for stratified catalyst heating with post injection					

General Information

For catalyst heating in stratified mode possibly with post injection there is an increase of exhaust gas temperature due to retarding the ignition angle. In this module a difference of the ignition angle is generated.

The function is divided into two parts: After start and low load catalyst heating.

30.10.1 Catalyst heating after start

Application Conditions

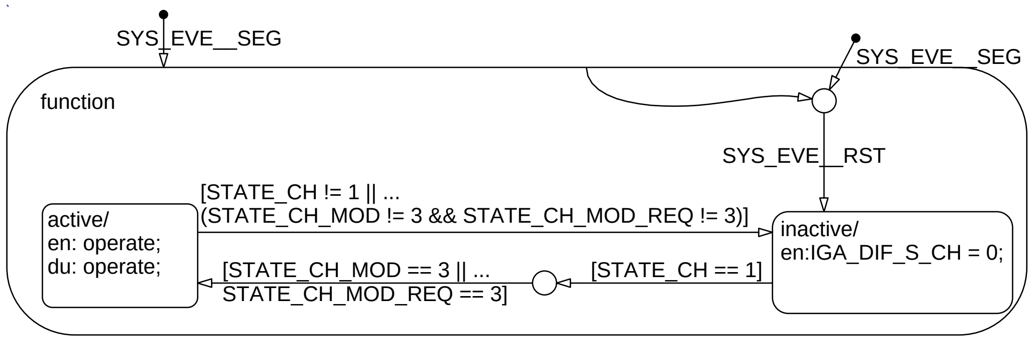


Figure 42:
Path: EXTC_REQGNsiga0/CH_AST/APP_CDN/Chart

Function description

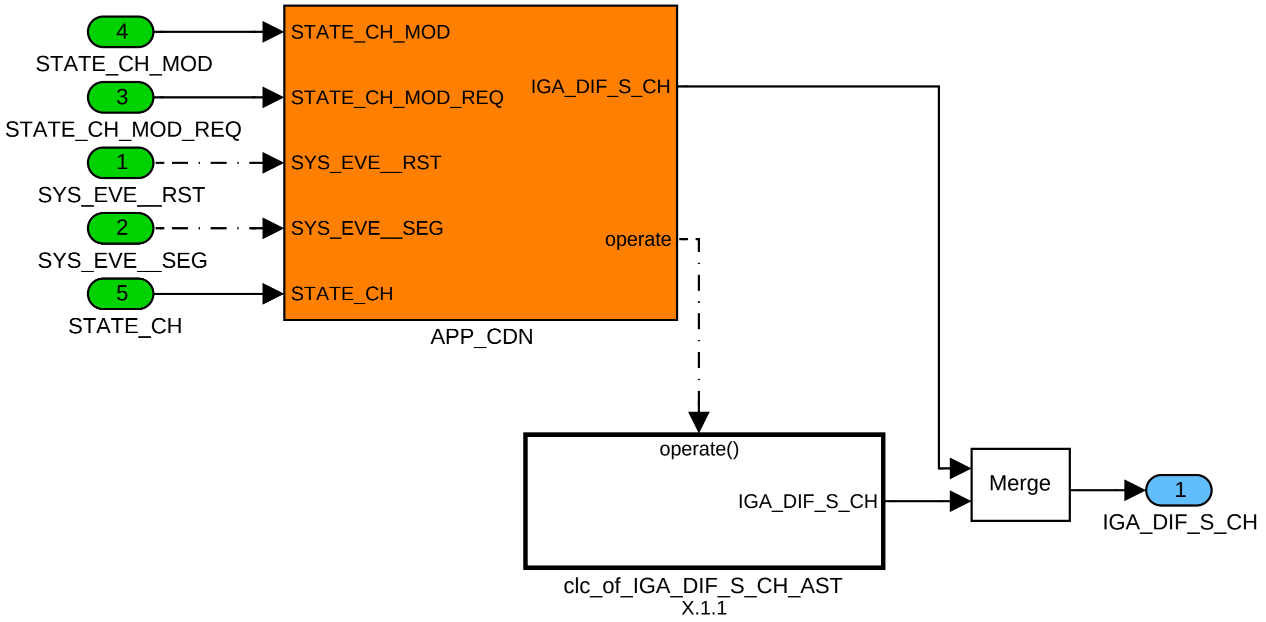


Figure 43:
Path: EXTC_REQGNsiga0/CH_AST

30.10.1.1 Function: Catalyst heating after start

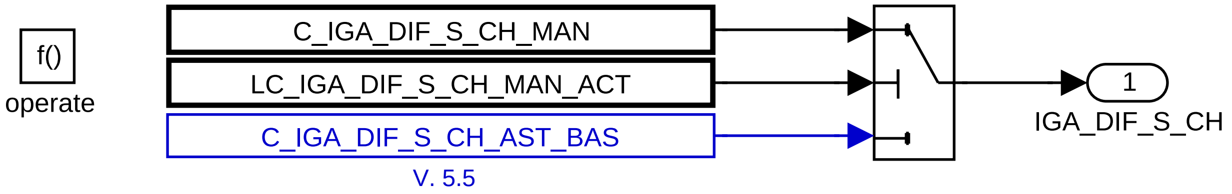


Figure 44:
Path: EXTC_REQGNsiga0/CH_AST/clc_of_IGA_DIF_S_CH_AST

30.10.2 Low load catalyst heating

Application Conditions

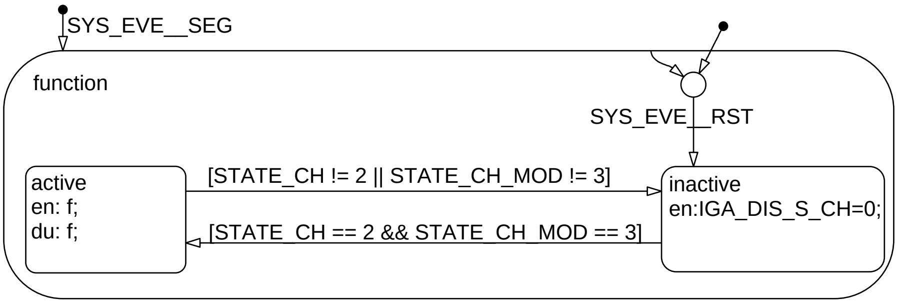


Figure 45:
Path: EXTC_REQGNsiga0/CH_L/APP_CDN/Chart

Function description

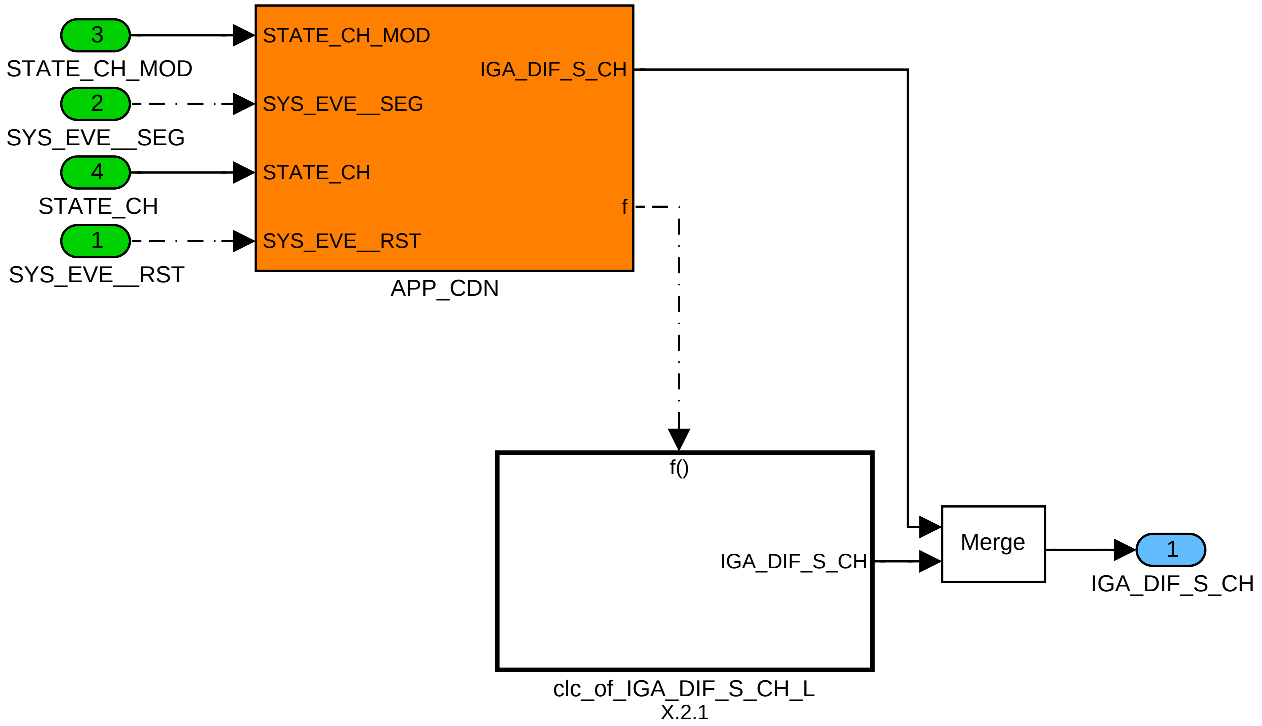


Figure 46:
Path: EXTC_REQGNsiga0/CH_L

30.10.2.1 Function: Low load catalyst heating

For low load CH there is a second correction depending on the actual fuel mass of post injection beside the CH_ASTfunctionality. The read_vector block picks out one element (here the 1st) of the cylinder individual MFF_POSTvector.

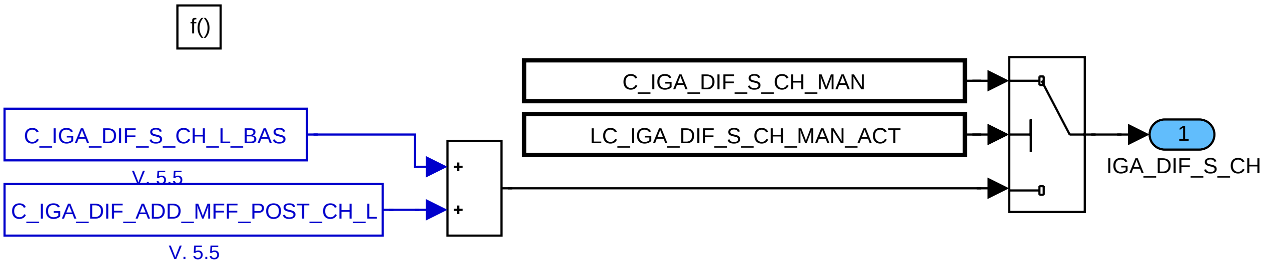


Figure 47:
Path: EXTC_REQGNsiga0/CH_L/clc_of_IGA_DIF_S_CH_L

30.11 Hom catalyst heating with multiple injection 1. pulse

Overview

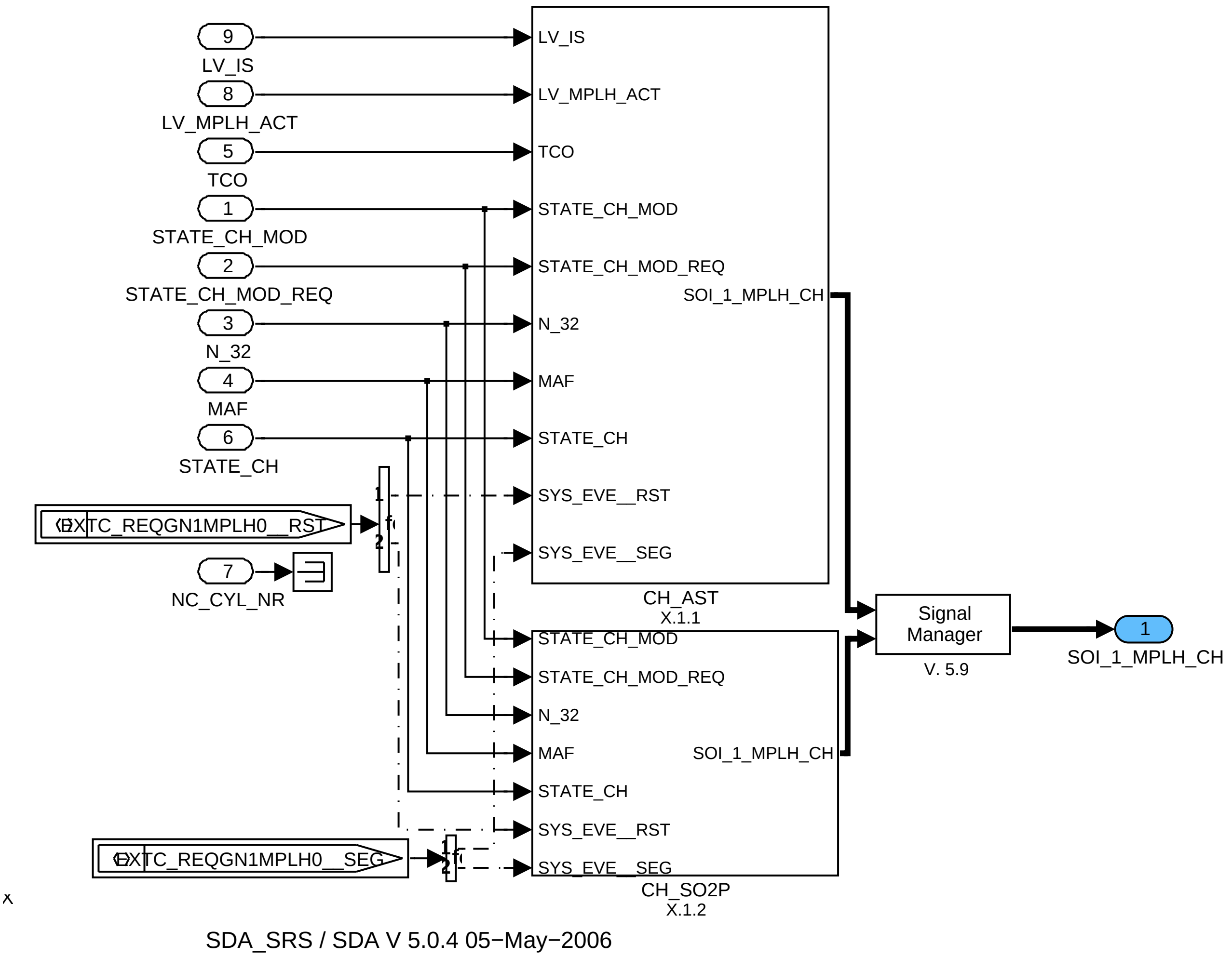


Figure 48:
Path: EXTC_REQGN1MPLH0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
SOI_1_MPLH_CH [NC_CYL_NR]	O/V	0... 780H	0... 720	0.375	[°CRK]
Start of 1st injection for catalyst heating with multiple injection in homogeneous					

Input Data:

STATE_CH_MOD	STATE_CH_MOD_REQ	N_32	MAF
TCO	STATE_CH	NC_CYL_NR	LV_MPLH_ACT
LV_IS			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_SOI_1_MPLH_CH_MAN	1	0... 780H	0... 720	0.375	[°CRK]
Manual setting of start of injection for catalyst heating					
IP_SOI_1_CH_ADD_TCO	6	0... F00H	-720 ...720	0.375	[°CRK]
LDP_TCO_IP_SOI_1_CH_ADD_TCO	6	0... FEH	-48... 142.5	0.75	[°C]
Additive value for end of 1st injection related to ignition angle depending on coolant temperature					
IP_SOI_1_MPLH_CH_AST	6*6	0... 780H	0... 720	0.375	[°CRK]
LDPM_N_32_7_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_1_EXTC	6	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Basic start of injection for CH					
IP_SOI_1_MPLH_CH_AST_IS	6*6	0... 780H	0... 720	0.375	[°CRK]
LDPM_N_32_5_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDP_MAF_IP_SOI_1_MPLH_CH_AST_IS	6	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Basic start of injection for CH in idle					
IP_SOI_1_MPLH_CH_SO2P	8*8	0... 780H	0... 720	0.375	[°CRK]
LDPM_N_32_8_EXTC	8	0... FFH	0... 8160	32	[rpm]
LDPM_MAF_2_EXTC	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Basic start of injection for MPLH during catalyst heating for desulfation					
LC_SOI_1_MPLH_CH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of start of injection for catalyst heating					

General Information

In this module a special phasing of the 1st pulse in homogeneous split injection in case of catalyst heating is generated.

30.11.1 Catalyst heating after start

Application Conditions

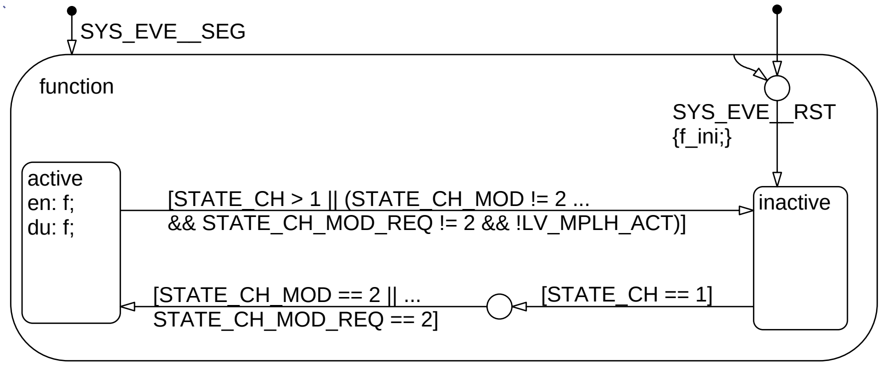


Figure 49:
Path: EXTC_REQGN1MPLH0/CH_AST/APP_CDN/Chart

Function description

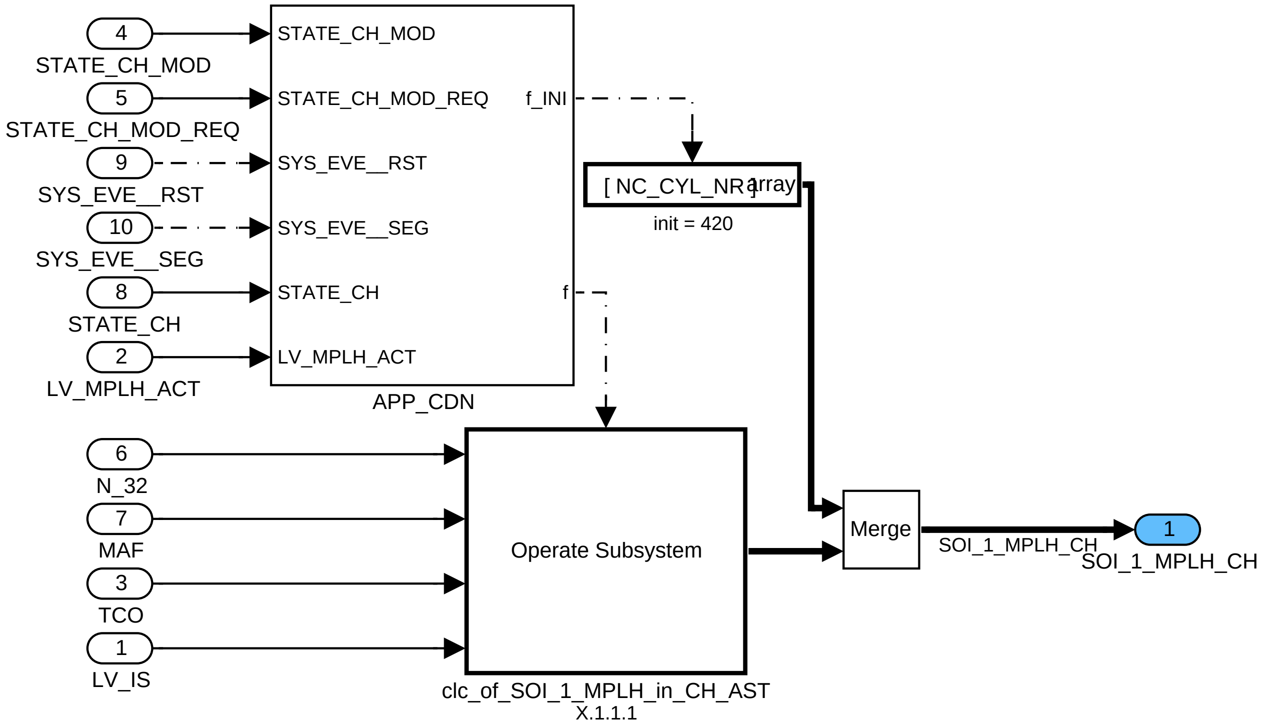


Figure 50:
Path: EXTC_REQGN1MPLH0/CH_AST

30.11.1.1 Catalyst heating after start - Subsystem

In the For-system the scalar value is written to each row of the NC_CYL_NR-dimensional vector.

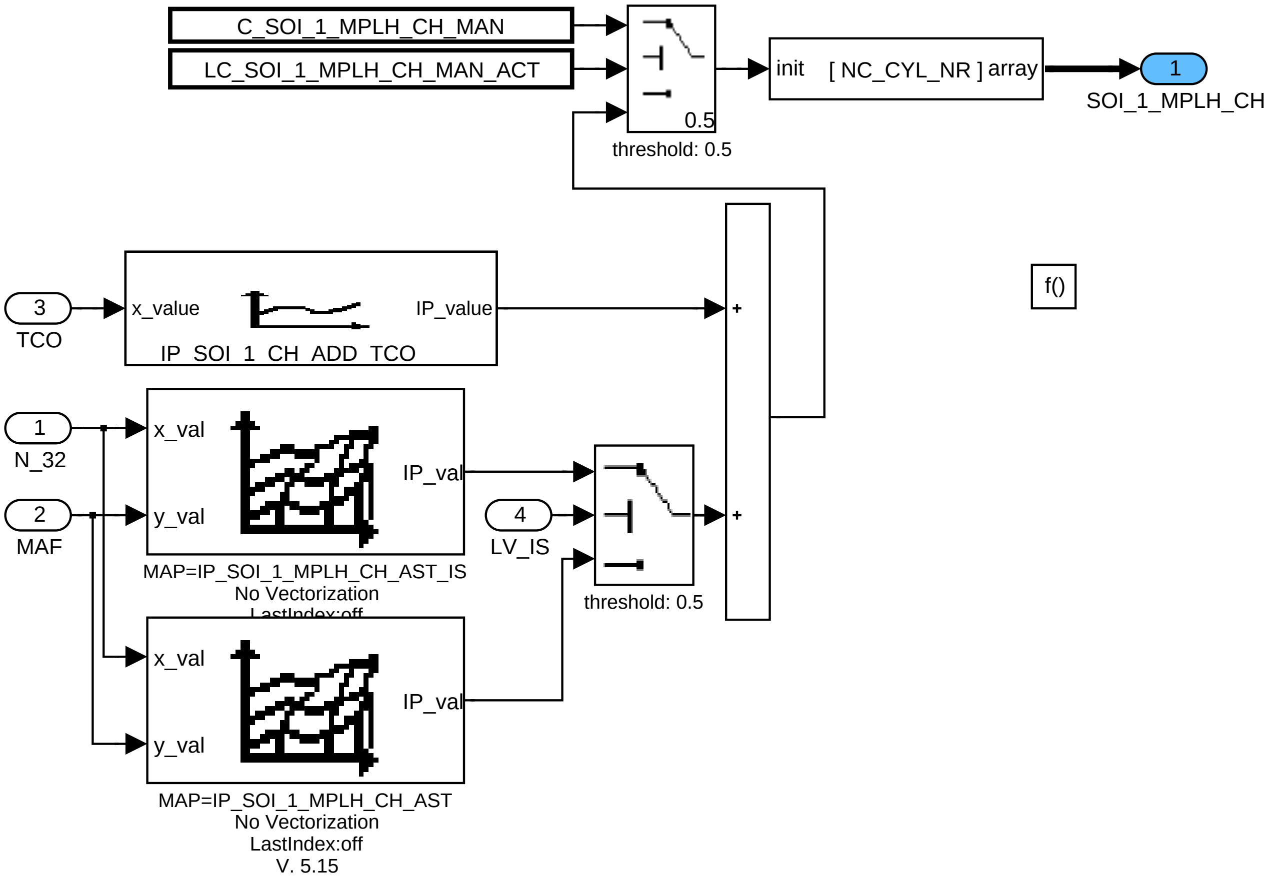


Figure 51:
Path: EXTC_REQGN1MPLH0/CH_AST/clc_of_SOI_1_MPLH_in_CH_AST

30.11.2 Catalyst heating for desulfation

Application Conditions

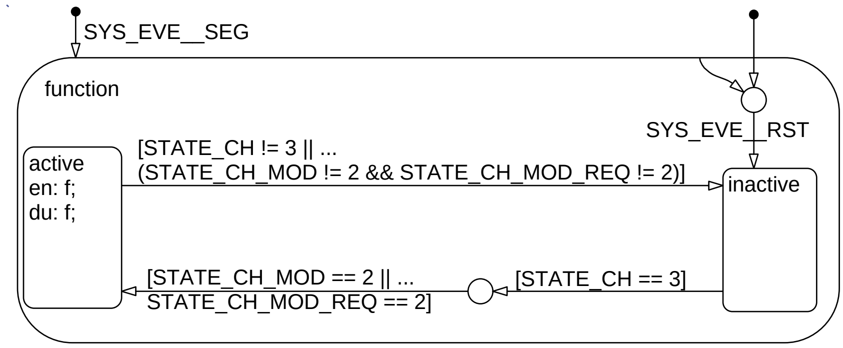


Figure 52:
Path: EXTC_REQGN1MPLH0/CH_SO2P/APP_CDN/Chart

Function description

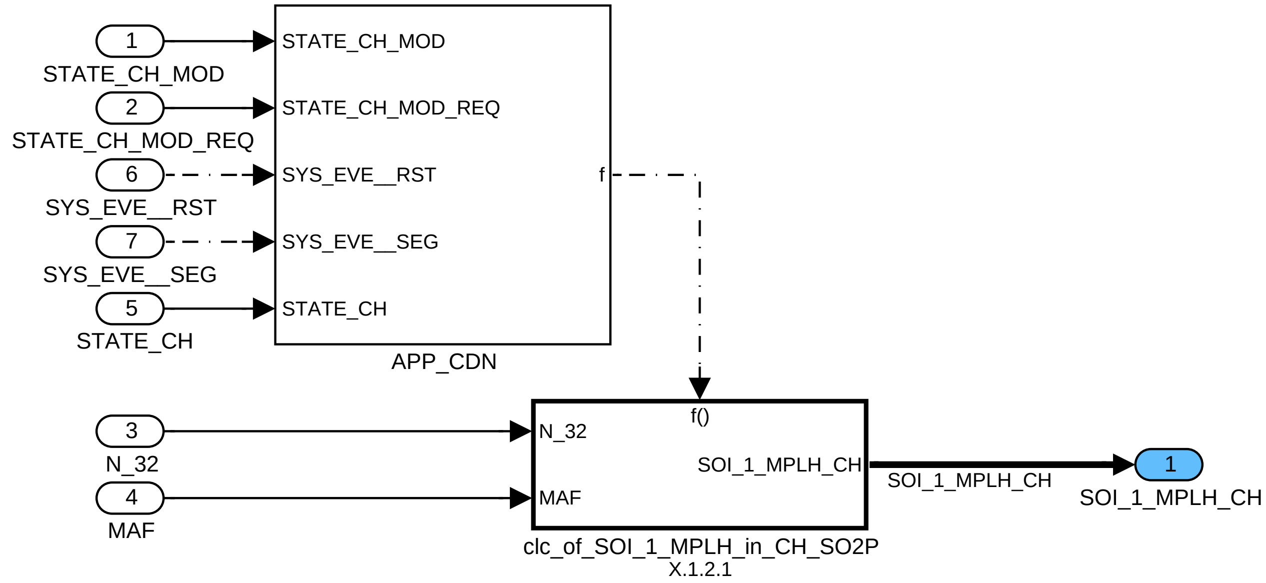


Figure 53:
Path: EXTC_REQGN1MPLH0/CH_SO2P

30.11.2.1 Catalyst heating for desulfation - Subsystem

In the For-system the scalar value is written to each row of the NC_CYL_NR-dimensional vector.

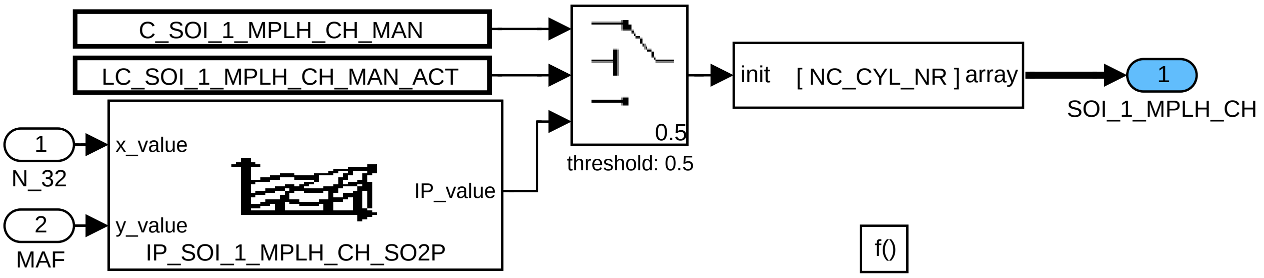
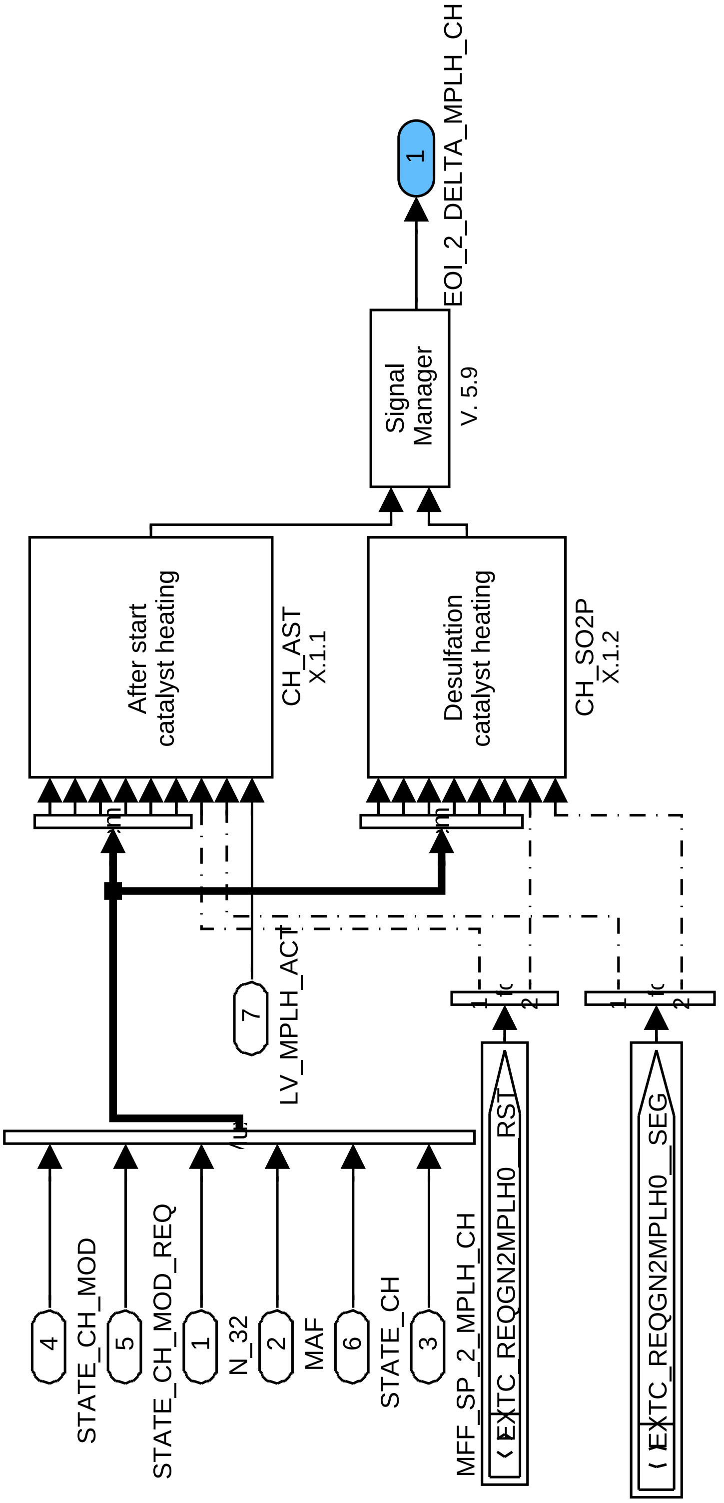


Figure 54:
Path: EXTC_REQGN1MPLH0/CH_SO2P/clc_of_SOI_1_MPLH_in_CH_SO2P

30.12 Hom catalyst heating with double injection 2. pulse

Overview



SDA_SRS / SDA V 5.0.1 02–Dec–2005

Figure 55:
Path: EXTC_REQGN2MPLH0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
EOI_2_DELTA_MPLH_CH	O/V	0... 780H	0... 720	0.375	[°CRK]
End of injection of the second injection for catalyst heating - related to ignition angle					

Input Data:

LV_MPLH_ACT	MAF	MFF_SP_2_MPLH_CH	N_32
STATE_CH	STATE_CH_MOD	STATE_CH_MOD_REQ	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_EOI_2_DELTA_MPLH_CH_MAN	1	0... 780H	0... 720	0.375	[°CRK]
Manual setting of end of injection of the second injection for catalyst heating					
IP_EOI_2_DELTA_MPLH_CH_AST	6x6	0... 780H	0... 720	0.375	[°CRK]
LDPM_N_32_7_EXTC	6	0... FFH	0... 8.16e+3	32	[rpm]
LDPM_MAF_1_EXTC	6	0... FFFFH	0... 1.389e+3	0.0211948	[mg/stk]
End of 2nd injection in after start catalyst heating in homogeneous with split injection- related to ignition angle					
IP_EOI_2_DELTA_MPLH_CH_SO2P	8x8	0... 780H	0... 720	0.375	[°CRK]
LDPM_N_32_8_EXTC	8	0... FFH	0... 8.16e+3	32	[rpm]
LDPM_MAF_2_EXTC	8	0... FFFFH	0... 1.389e+3	0.0211948	[mg/stk]
End of 2nd injection in desulfation catalyst heating in homogeneous with split injection- related to ignition angle					
IP_EOI_2_MPLH_CH_ADD_MFF_2	6	0... F00H	-720... 720	0.375	[°CRK]
LDP_MFF_2_IP_EOI_2_MPLH_CH_ADD	6	0... FFFFH	0... 1.389e+3	0.0211948	[mg/stk]
Additive value for end of 2nd injection					
LC_EOI_2_DELTA_MPLH_CH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of end of injection of the second injection for catalyst heating					

General Information

For catalyst heating in homogeneous split mode an additional injection pulse is generated. This module contains the phasing of this pulse.
EOI_2_DELTA_MPLH_CH therefore is defined as crankshaft angle difference between end of injection of 2nd pulse and ignition angle.

30.12.1 After start catalyst heating

Application Conditions

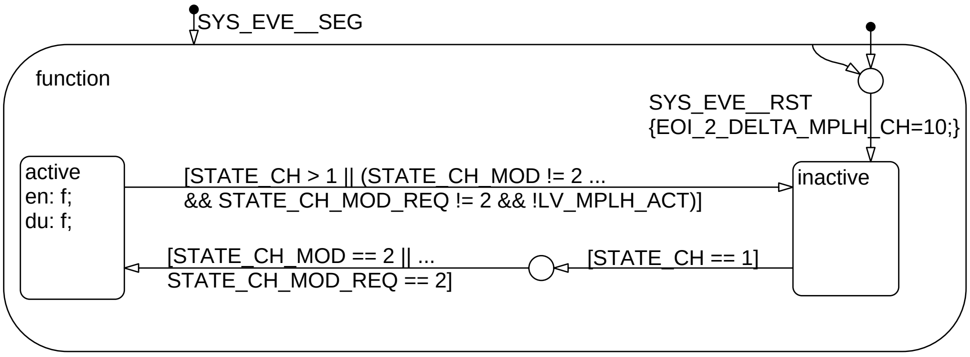


Figure 56:
Path: EXTC_REQGN2MPLH0/CH_AST/APP_CDN/Chart

Function description

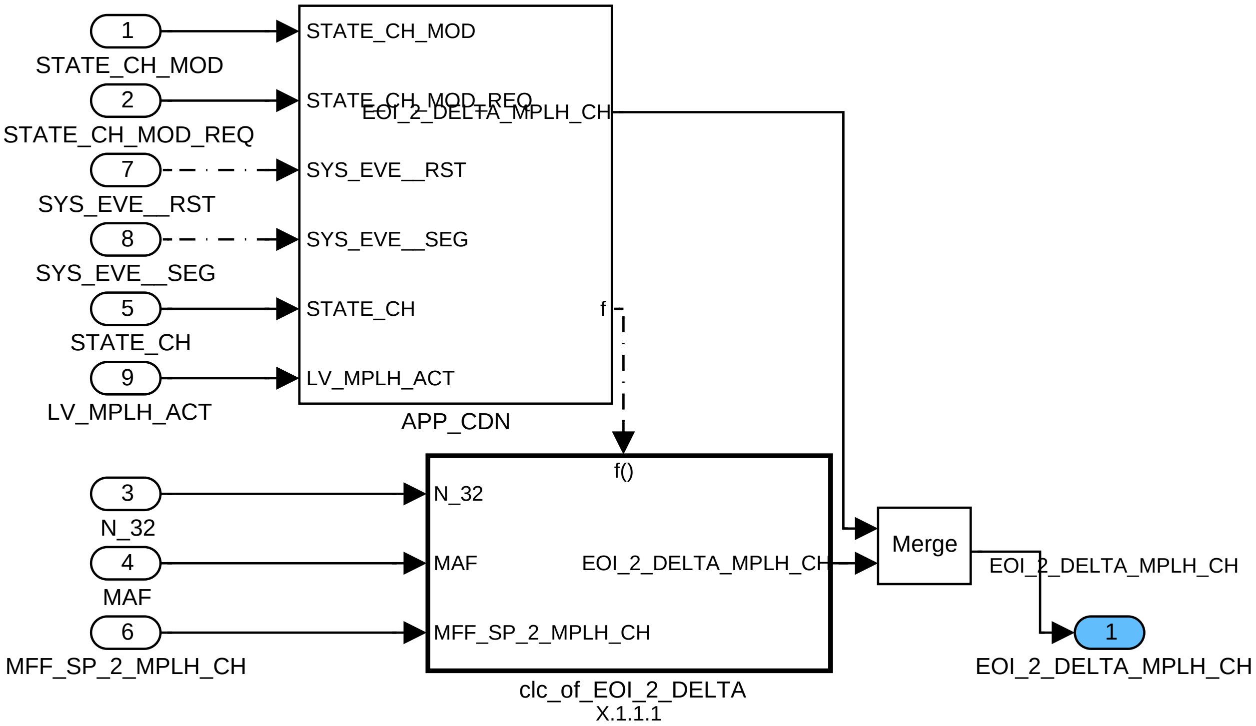


Figure 57:
Path: EXTC_REQGN2MPLH0/CH_AST

30.12.1.1 Calculation of phasing for 2nd pulse

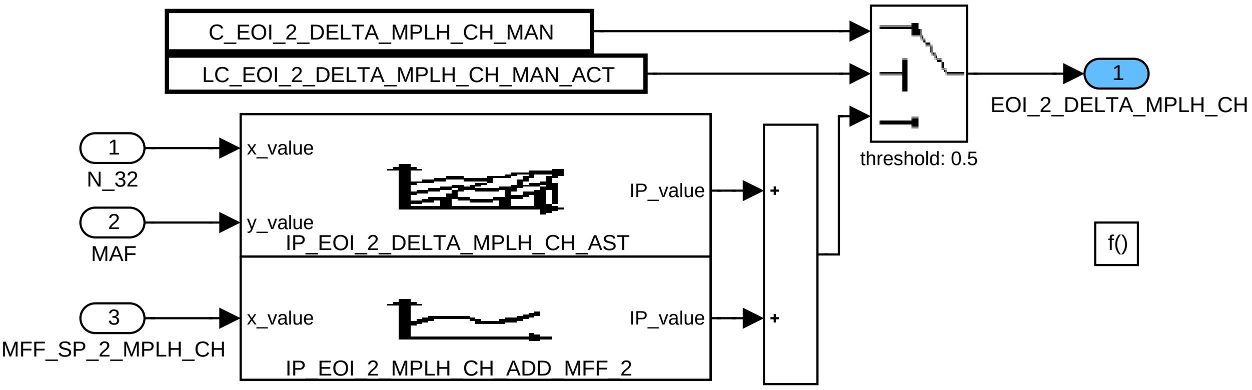


Figure 58:
Path: EXTC_REQGN2MPLH0/CH_AST/clc_of_EOI_2_DELTA

30.12.2 Desulfation catalyst heating

<u>Application Conditions</u>

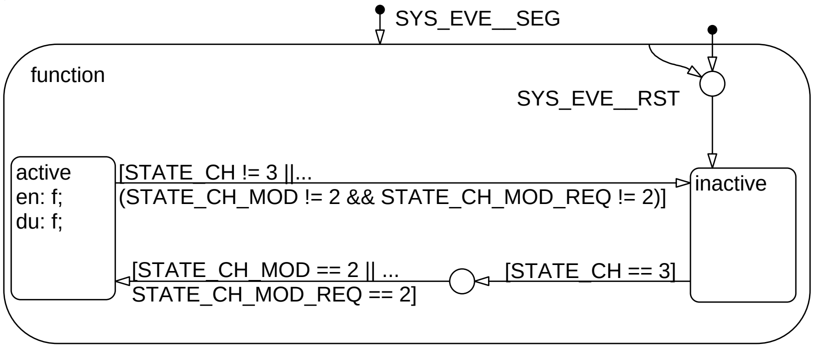


Figure 59:
Path: EXTC_REQGN2MPLH0/CH_SO2P/APP_CDN/Chart

Function description

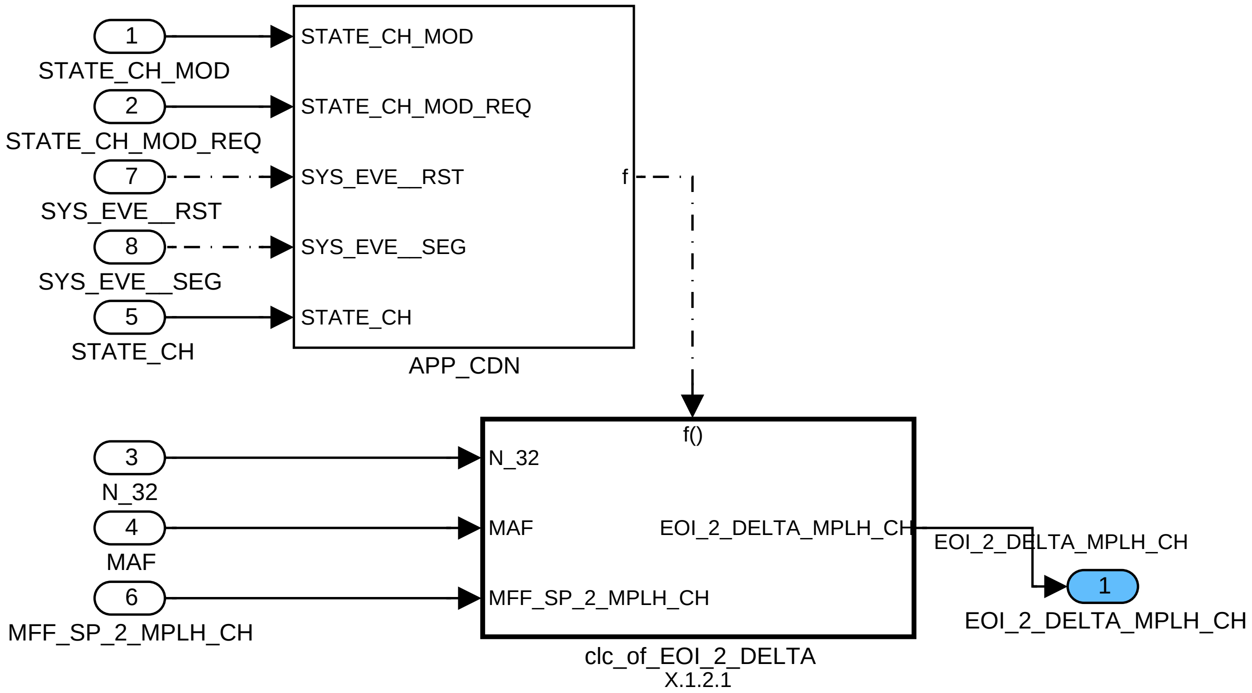


Figure 60:
Path: EXTC_REQGN2MPLH0/CH_SO2P

30.12.2.1 Calculation of phasing for 2nd pulse

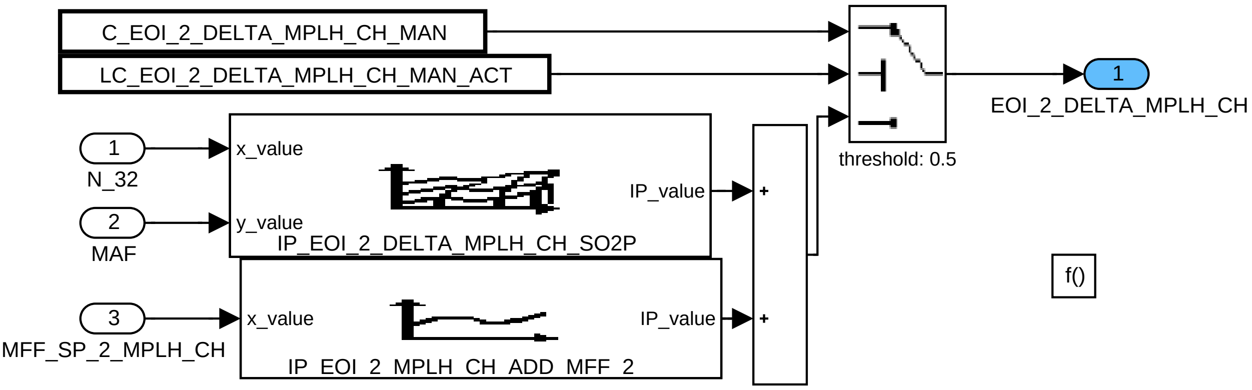


Figure 61:
Path: EXTC_REQGN2MPLH0/CH_SO2P/clc_of_EOI_2_DELTA

30.13 Fuel mass setpoint of second injection for hom. catalyst heating

Overview

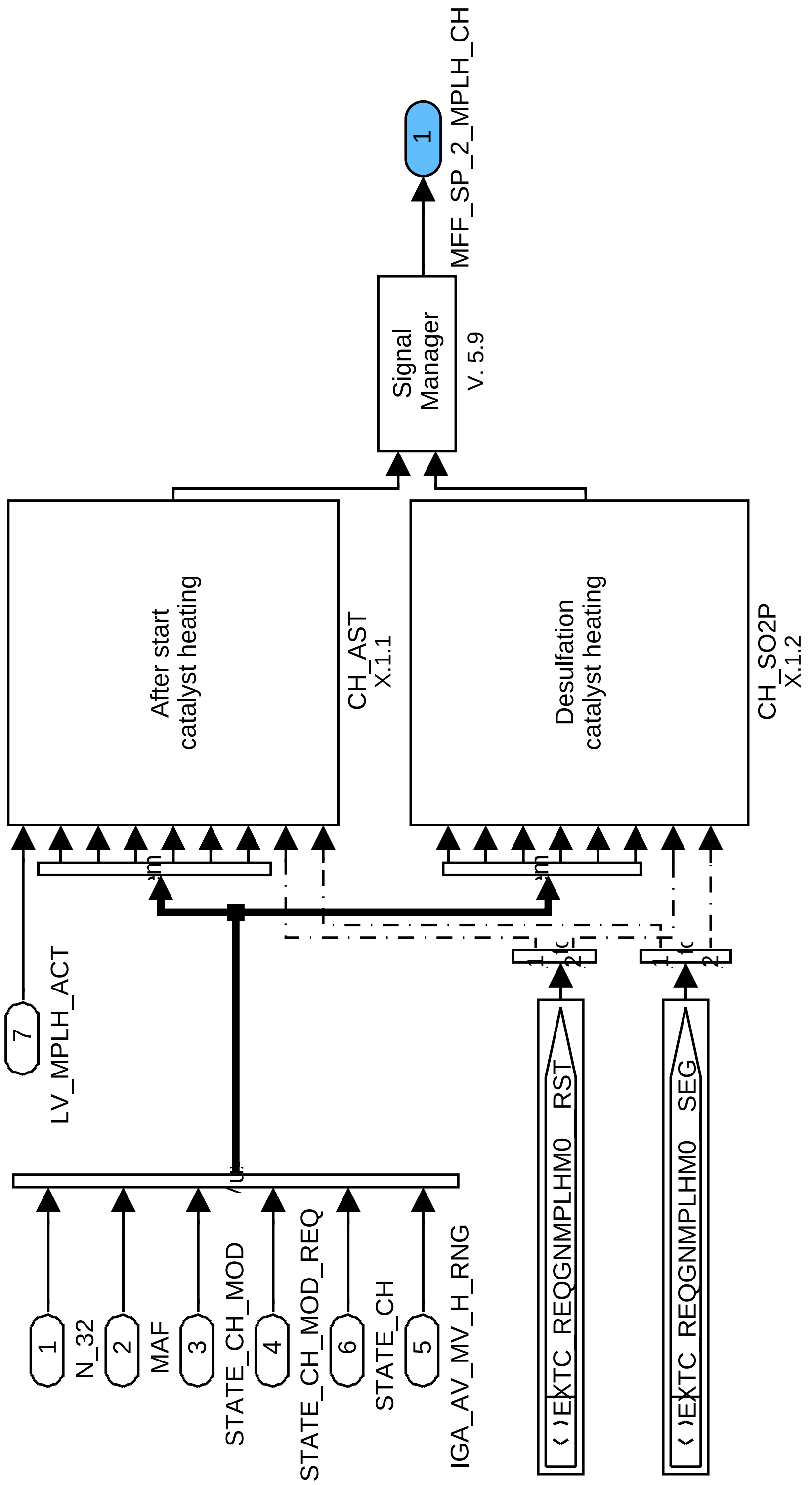


Figure 62:
Path: EXTC_REQGNMPLHM0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
MFF_SP_2_MPLH_CH	O/V	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Fuel mass for second injection pulse					

Input Data:

IGA_AV_MV_H_RNG	LV_MPLH_ACT	MAF	N_32
STATE_CH	STATE_CH_MOD	STATE_CH_MOD_REQ	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_MFF_SP_2_MPLH_CH_MAN	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Manual setting of fuel mass for second injection pulse					
IP_MFF_SP_2_ADD_IGA_CH	4	0... FFFFH	-694.5... 694.4788	0.0211945	[mg/stk]
LDP_IGA_H_RNG_IP_MFF_2_MPLH_CH	4	0... B40H	-90... 90	0.0625	[°CRK]
Ignition angle depending correction of fuel mass for second injection pulse					
IP_MFF_SP_2_MPLH_CH_AST	6x6	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDPM_N_32_7_EXTC	6	0... FFH	0... 8.16e+3	32	[rpm]
LDPM_MAF_1_EXTC	6	0... FFFFH	0... 1.389e+3	0.0211948	[mg/stk]
Basic fuel mass for second injection pulse in after start catalyst heating					
IP_MFF_SP_2_MPLH_CH_SO2P	8x8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDPM_N_32_8_EXTC	8	0... FFH	0... 8.16e+3	32	[rpm]
LDPM_MAF_2_EXTC	8	0... FFFFH	0... 1.389e+3	0.0211948	[mg/stk]
Fuel mass for second injection pulse in desulfation catalyst heating					
LC_MFF_SP_2_MPLH_CH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of fuel mass for second injection pulse					

General Information

To heat up the catalyst a 2nd fuel mass pulse is injected in homogeneous mode short time before ignition. In this module the mass of this pulse is generated.
The ignition angle is being retarded to increase the exhaust temperature. The 2nd injection mass MFF_SP_2_MPLH_CH should be calculated depending on ignition angle. That is done through IGA_AV_MV_H_RNG.

30.13.1 After start catalyst heating

Application Conditions

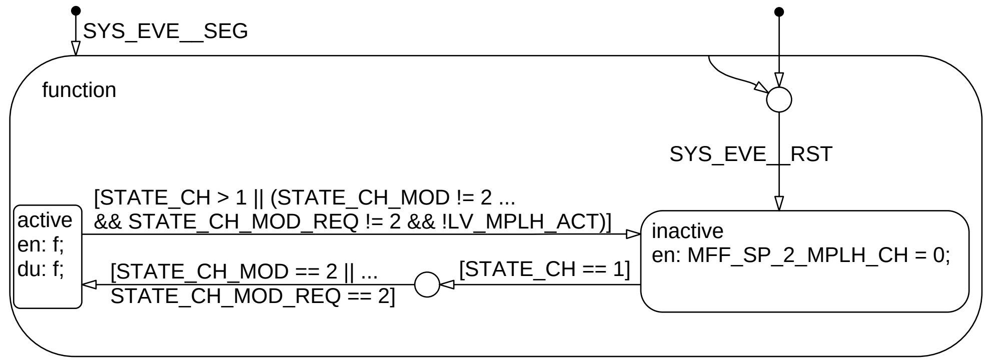


Figure 63:
Path: EXTC_REQGNMPLHM0/CH_AST/APP_CDN/Chart

Function description

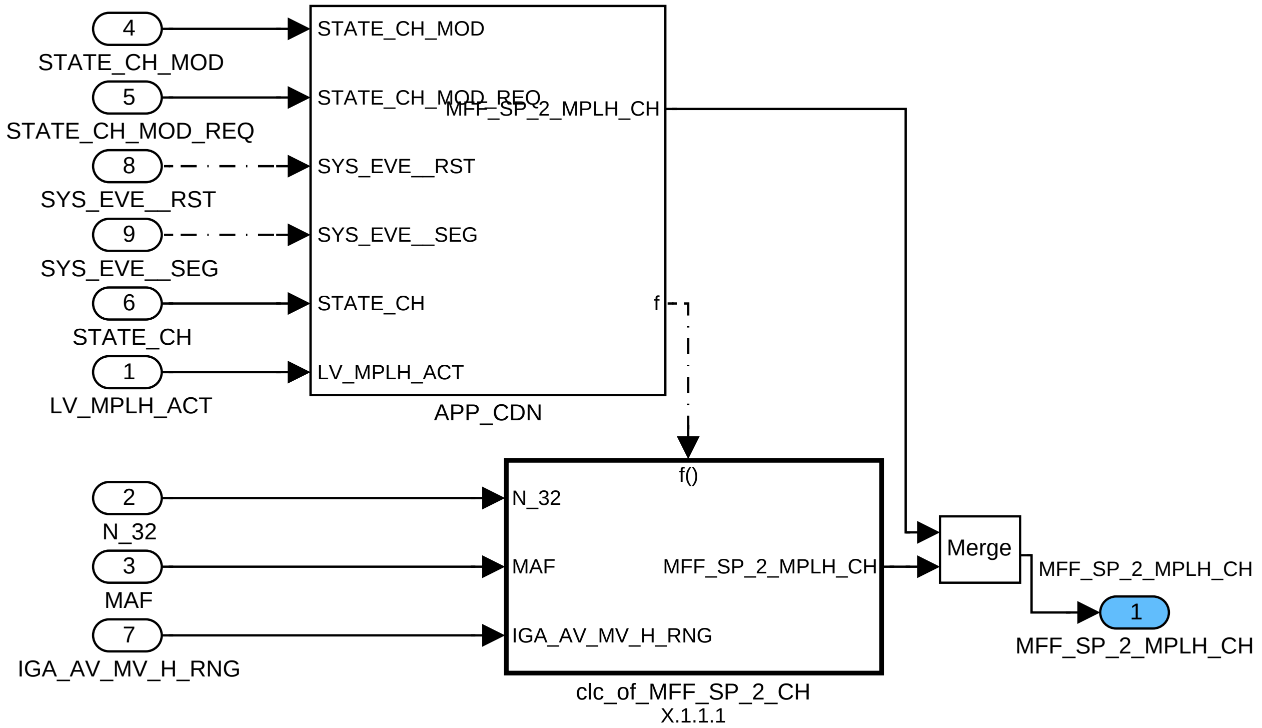


Figure 64:
Path: EXTC_REQGNMPLHM0/CH_AST

30.13.1.1 Function: After start catalyst heating

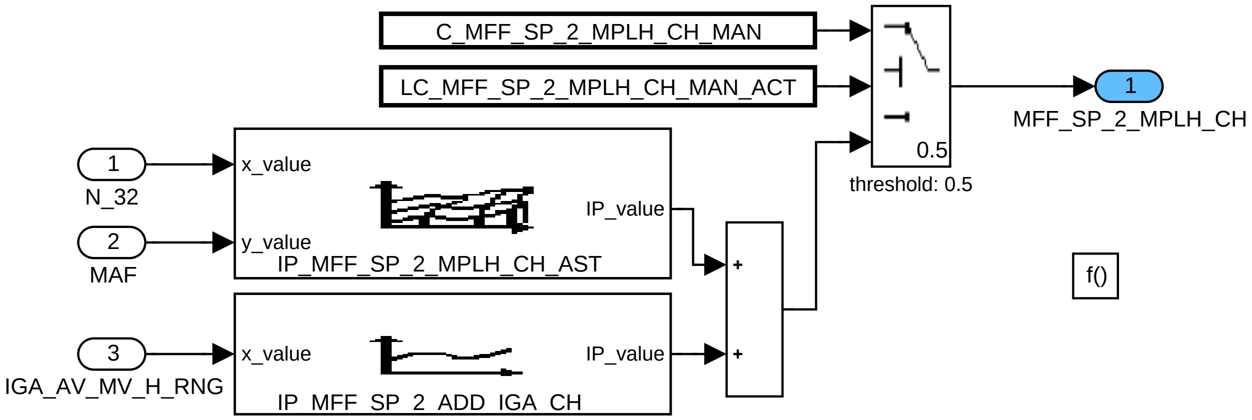


Figure 65:
Path: EXTC_REQGNMPLHM0/CH_AST/clc_of_MFF_SP_2_CH

30.13.2 Desulfation catalyst heating

Application Conditions

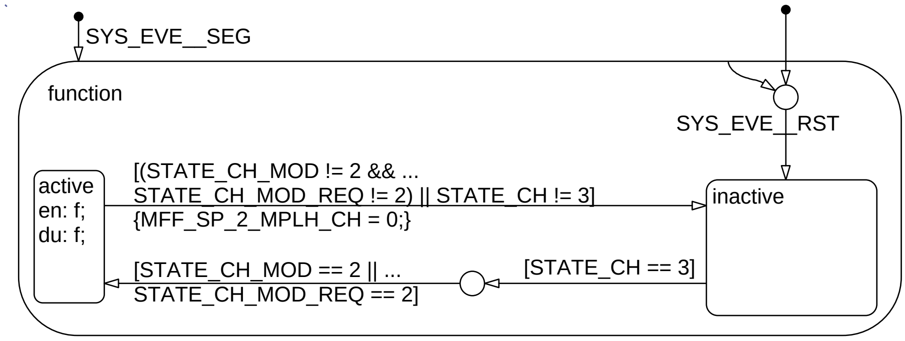


Figure 66:
Path: EXTC_REQGNMPLHM0/CH_SO2P/APP_CDN/Chart

Function description

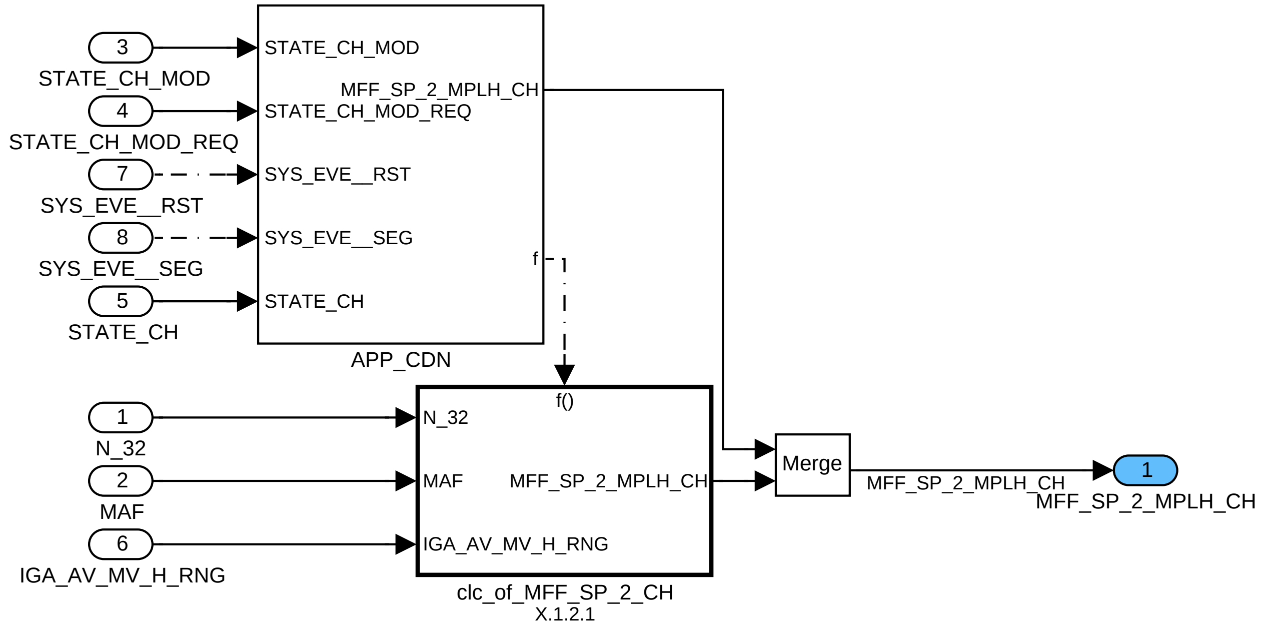


Figure 67:
Path: EXTC_REQGNMPLHM0/CH_SO2P

30.13.2.1 Function: Desulfation catalyst heating

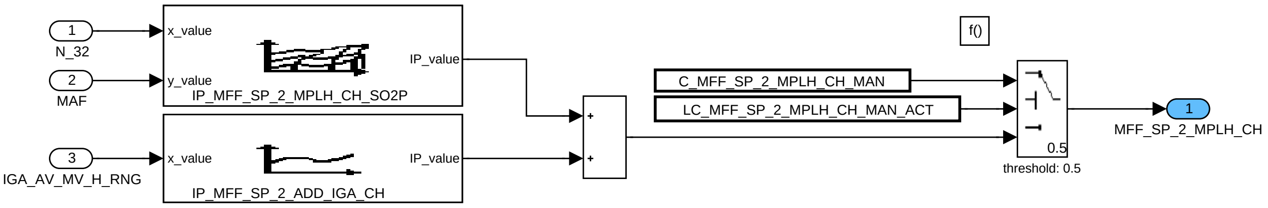


Figure 68:
Path: EXTC_REQGNMPLHM0/CH_SO2P/clc_of_MFF_SP_2_CH

30.14 Lambda setpoint for catalyst heating

Overview

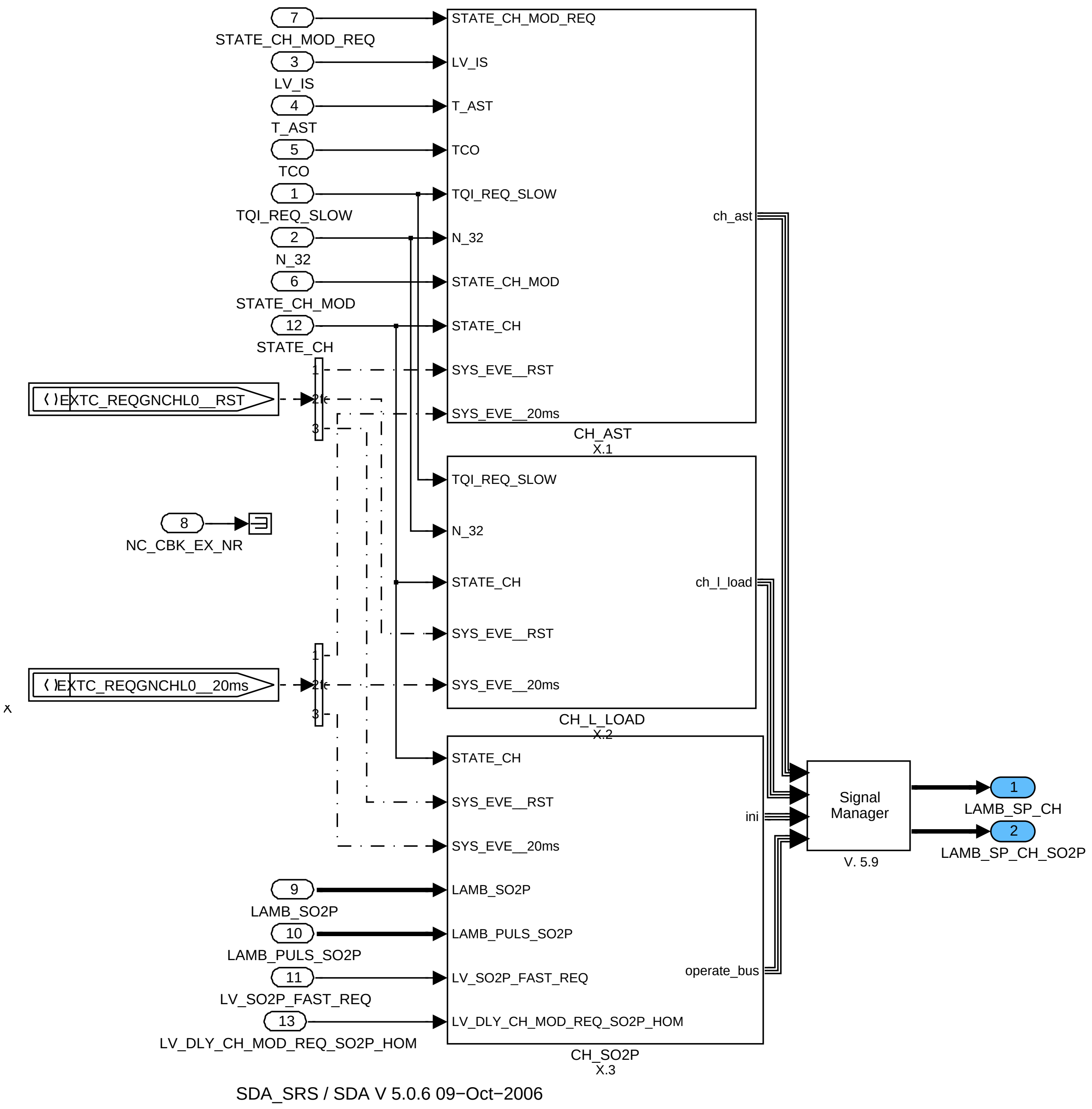


Figure 69:
Path: EXTC_REQGNCHL0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LAMB_SP_CH [NC_CBK_EX_NR]	O/V	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda setpoint for catalyst heating - bankselective					
LAMB_SP_CH_SO2P [NC_CBK_EX_NR]	O/V	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda setpoint for catalyst heating - bankselective, during desulfation					

Input Data:

TQI_REQ_SLOW	N_32	LV_IS	T_AST
TCO	STATE_CH_MOD	STATE_CH_MOD_REQ	NC_CBK_EX_NR
LAMB_SO2P [NC_CBK_EX_NR]	LAMB_PULS_SO2P [NC_CBK_EX_NR]	LV_SO2P_FAST_REQ	STATE_CH
LV_DLY_CH_MOD_REQ_SO2P_HOM			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_LAMB_MV_CH_L_RGN	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Mean lambda value during catalyst heating with lambda split					
C_LAMB_SP_CH_MAN [NC_CBK_EX_NR]	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Manual setting of lambda setpoint for catalyst heating - bankselective					
IP_LAMB_ADD_CH_AST_HOM	8*6	0... 7FFFH	-16... 15.9990234375	976.563e-6	[-]
LDPM_T_AST_4_EXTC	6	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_5_EXTC	8	0... FEH	-48... 142.5	0.75	[°C]
T_AST and TCO depending additional correction for lambda setpoint for catalyst heating					
IP_LAMB_ADD_CH_AST_MPLH	8*6	0... 7FFFH	-16... 15.9990234375	976.563e-6	[-]
LDPM_T_AST_4_EXTC	6	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_5_EXTC	8	0... FEH	-48... 142.5	0.75	[°C]
T_AST and TCO depending additional correction for lambda setpoint for catalyst heating with dual injection					
IP_LAMB_CH_AST_HOM	6*6	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_N_32_10_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_SLOW_3_EXTC	6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Lambda setpoint for homogeneous after start catalyst heating in idle speed					
IP_LAMB_CH_AST_HOM_IS	4*4	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_N_32_9_EXTC	4	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_SLOW_4_EXTC	4	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Lambda setpoint for homogeneous after start catalyst heating in idle speed					
IP_LAMB_CH_AST_MPLH	6*6	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_N_32_10_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_SLOW_3_EXTC	6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Lambda setpoint for homogeneous after start catalyst heating with dual injection					
IP_LAMB_CH_AST_MPLH_IS	4*4	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]

LDPM_N_32_9_EXTC	4	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_SLOW_4_EXTC	4	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Lambda setpoint for homogeneous catalyst heating with dual injection in idle speed					
IP_LAMB_CH_L_RGN	6*6	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDP_N_32_IP_LAMB_CH_L_RGN	6	0... FFH	0... 8160	32	[rpm]
LDP_TQI_REQ_SLOW_IP_LAMB_CH_RGN	6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Lambda split value for enriched bank.					
LC_LAMB_SO2P_FAST_ENA	1	0... 1H	0... 1	1	[-]
Enables using of LAMB_SO2P in case of LV_SO2P_FAST_REQ (fast desulfation without wobble)					
LC_LAMB_SP_CH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of LAMB_SP_CH					
LC_LAMB_SP_CH_SO2P_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of LAMB_SP_CH_SO2P					

General Information

In this module a lambda-setpoint for homogeneous catalyst heating is generated.

30.14.1 After start catalyst heating

In stratified mode the lambda setpoint is also calculated if homogenous (split) mode is requested by STATE_CH_MOD_REQ.

Application Conditions

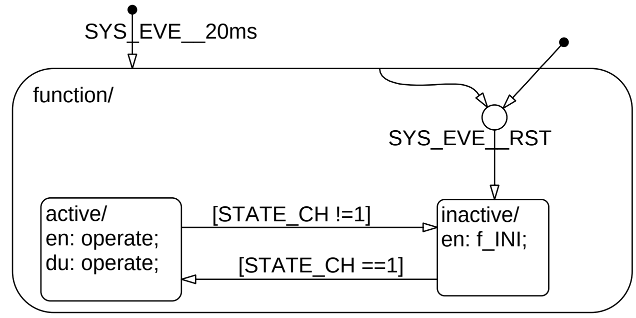


Figure 70:
Path: EXTC_REQGNCHL0/CH_AST/APP_CDN/Chart

Function description

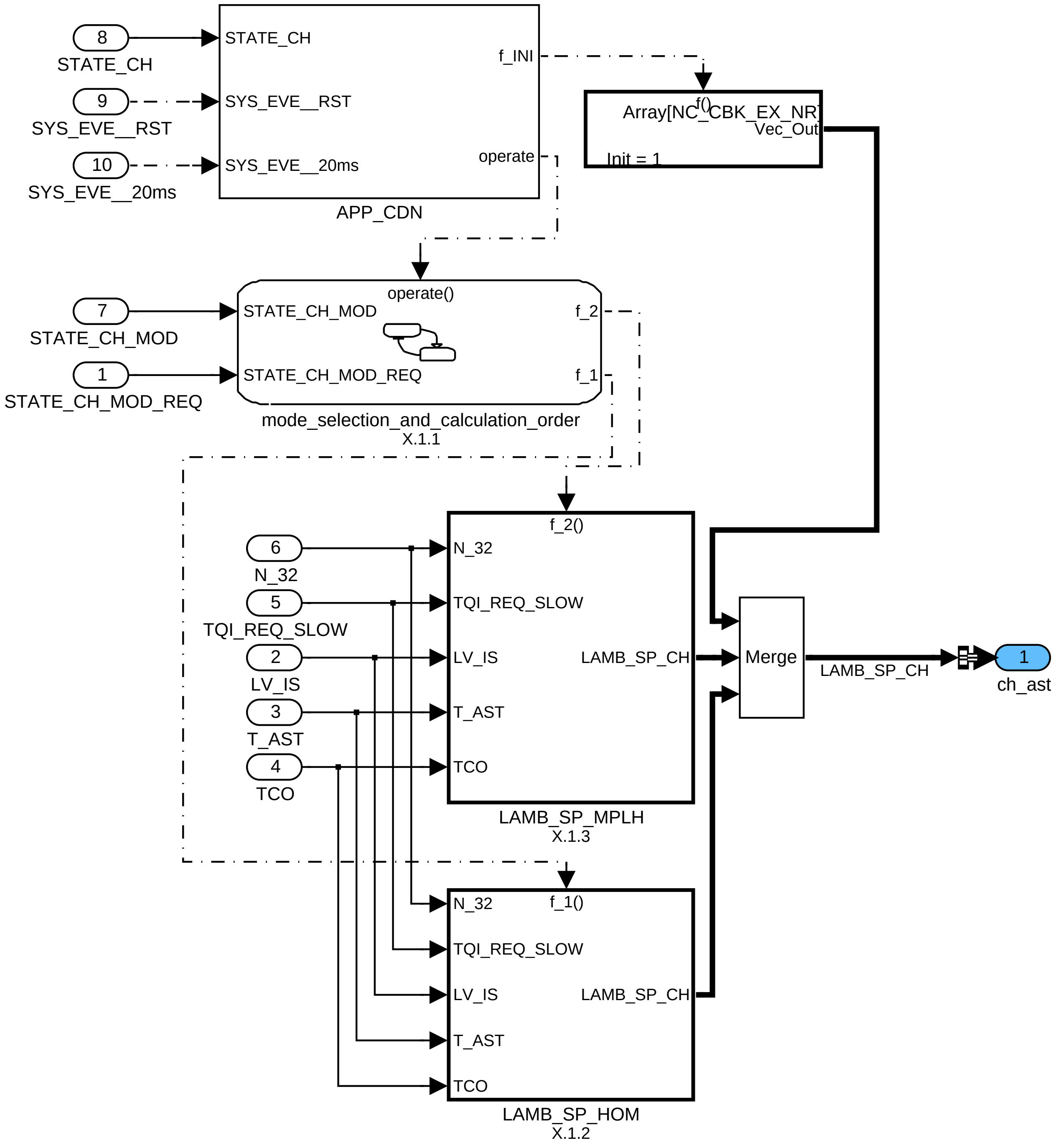


Figure 71:
Path: EXTC_REQGNCHL0/CH_AST

30.14.1.1 No title given

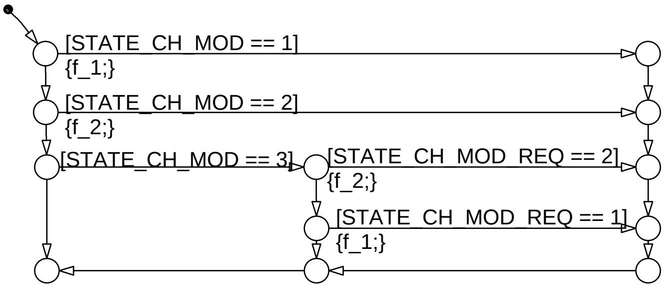


Figure 72:
Path: EXTC_REQGNCHL0/CH_AST/mode_selection_and_calculation_order

30.14.1.2 After start catalyst heating, single injection

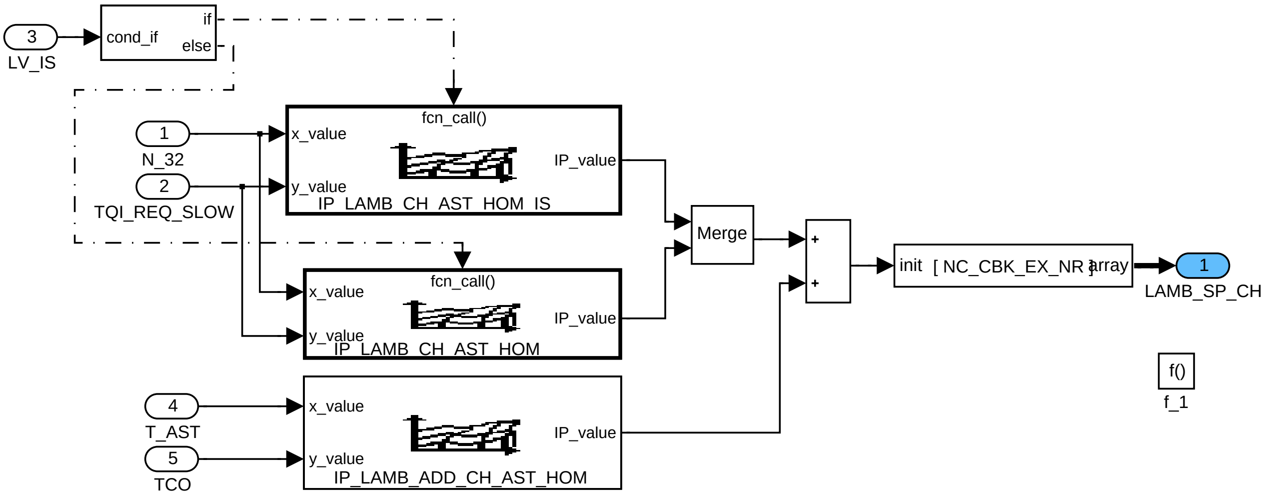


Figure 73:
Path: EXTC_REQGNCHL0/CH_AST/LAMB_SP_HOM

30.14.1.3 After start catalyst heating, multiple injection

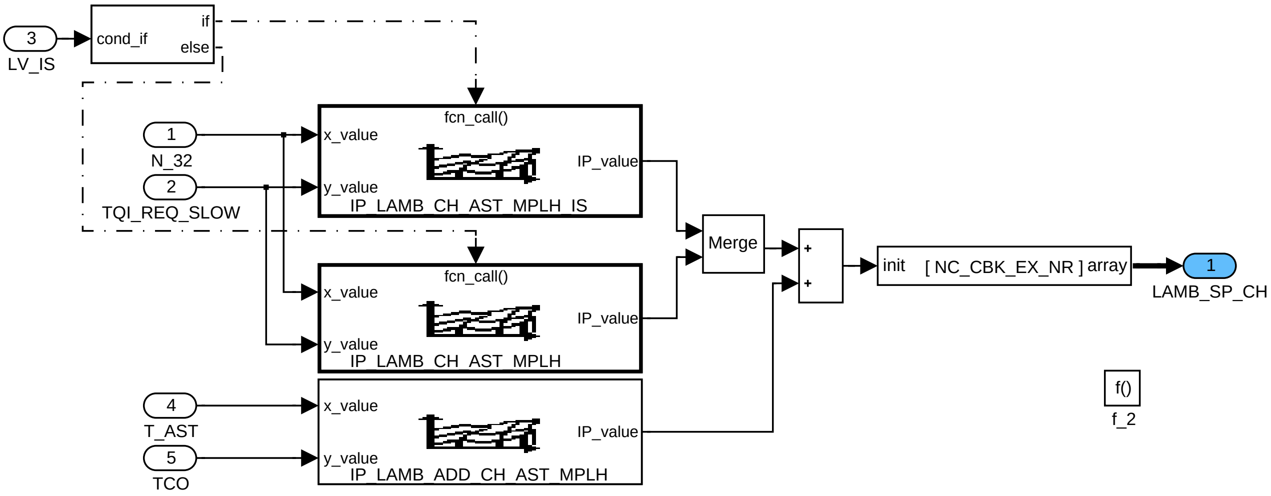


Figure 74:
Path: EXTC_REQGNCHL0/CH_AST/LAMB_SP_MPLH

30.14.2 Low load catalyst heating

Low load catalyst heating is activated after regeneration. With this function it is possilble to apply different lambda-setpoints for each bank. The function can only be used with Y exhaust-systems because the exhaust gases of the lean and the rich cylinder banks are mixed upstream the NOx-trap and cause exothermic reaction.

Application Conditions

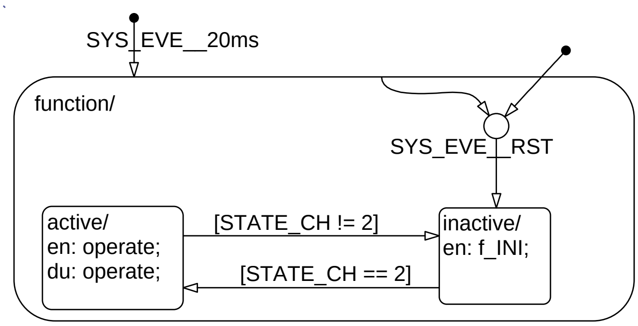


Figure 75:
Path: EXTC_REQGNCHL0/CH_L_LOAD/APP_CDN/Chart

Function description

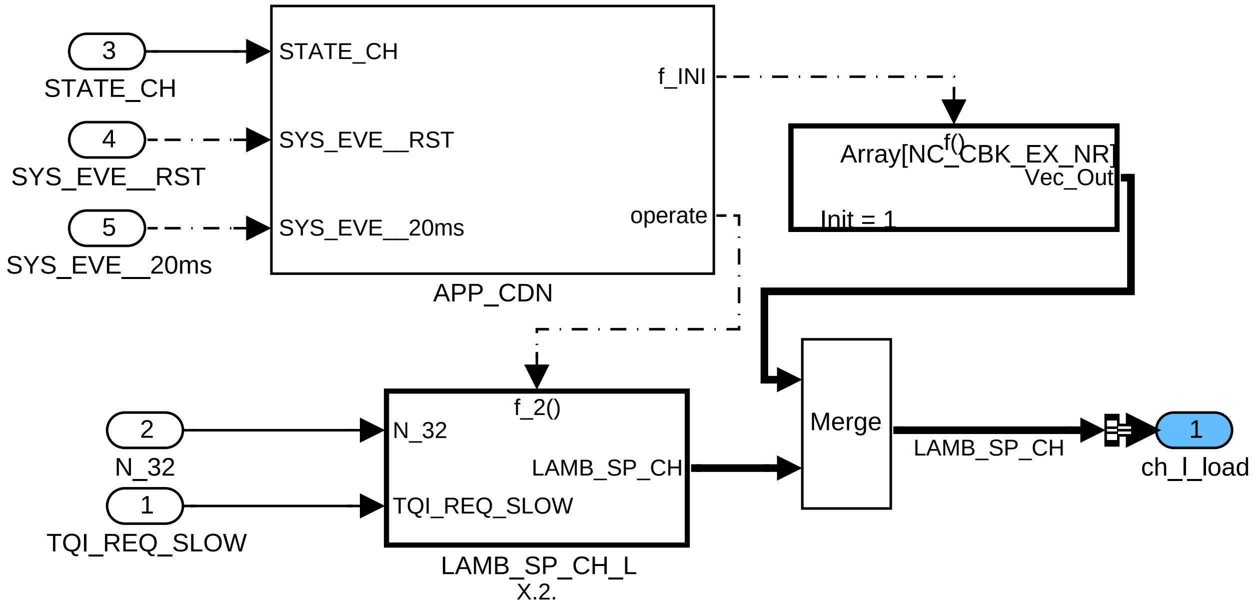


Figure 76:
Path: EXTC_REQGNCHL0/CH_L_LOAD

30.14.2.1 Function: Low load catalyst heating

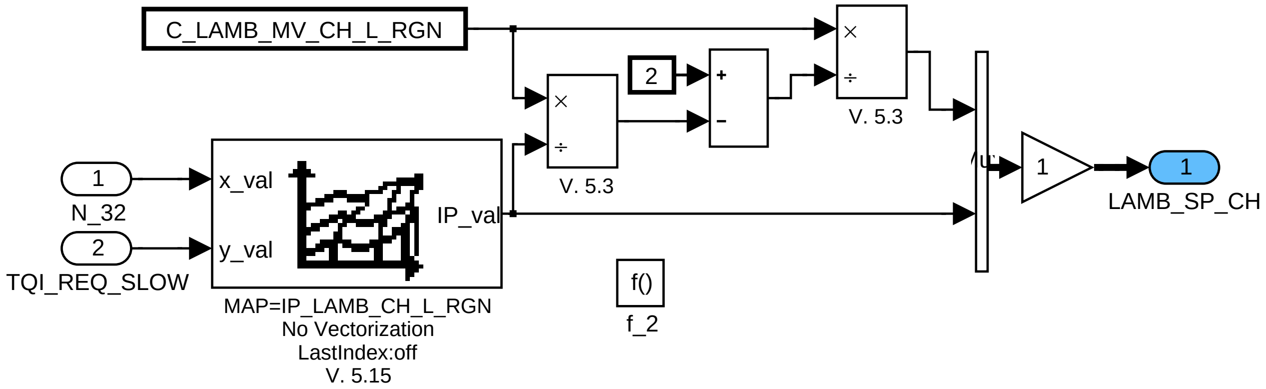


Figure 77:
Path: EXTC_REQGNCHL0/CH_L_LOAD/LAMB_SP_CH_L

30.14.3 Desulfation catalyst heating

Application Conditions

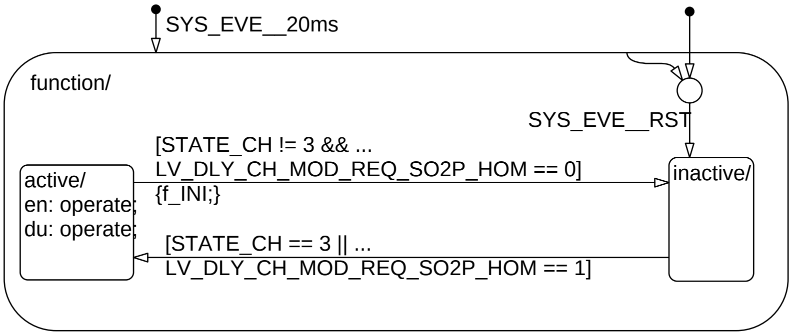


Figure 78:
Path: EXTC_REQGNCHL0/CH_SO2P/APP_CDN/Chart

Function description

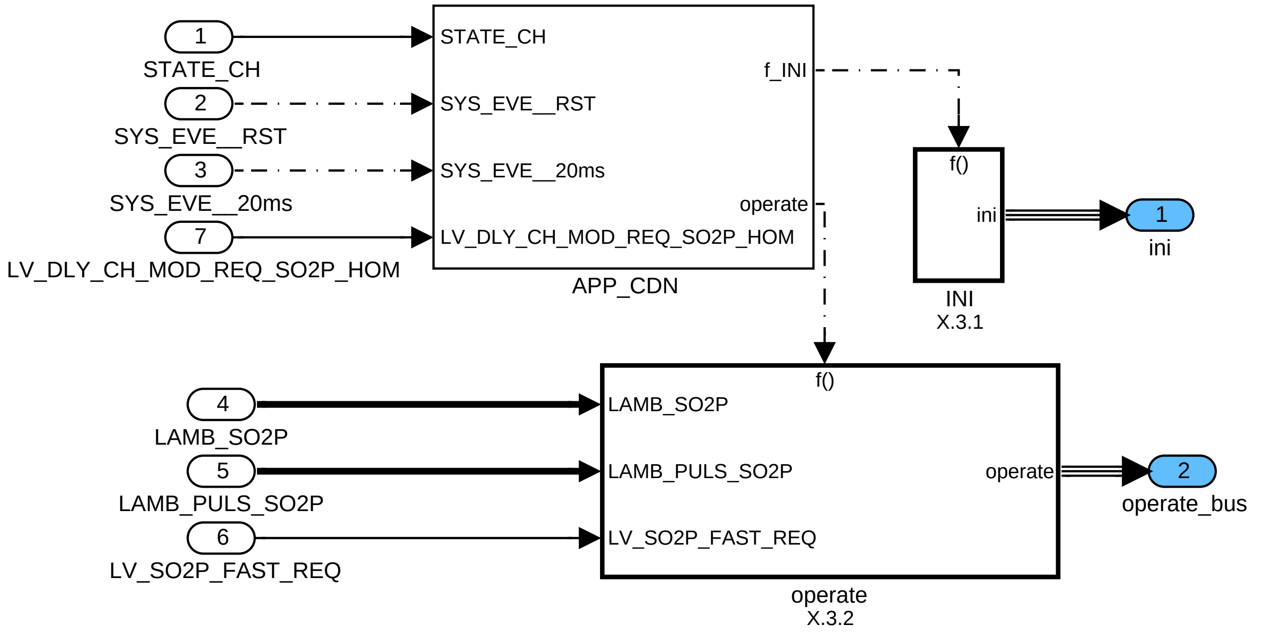


Figure 79:
Path: EXTC_REQGNCHL0/CH_SO2P

30.14.3.1 Initialisation

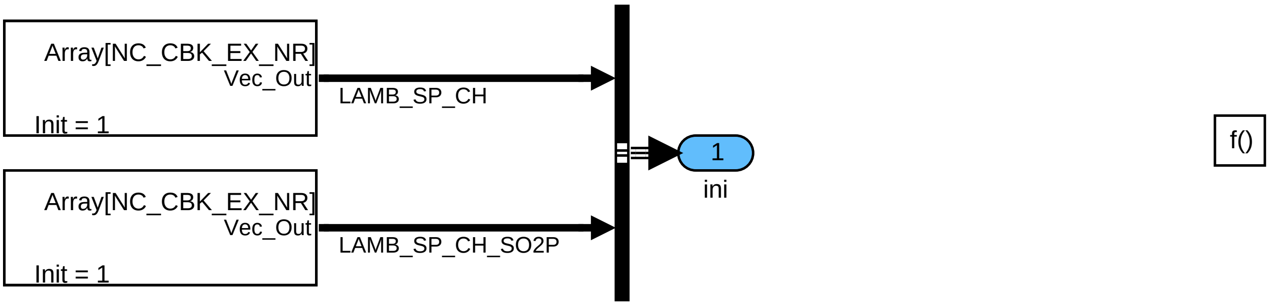


Figure 80:
Path: EXTC_REQGNCHL0/CH_SO2P/INI

30.14.3.2 Function: Desulfation catalyst heating

The basic lambda value for catalyst desulfation catalyst heating is applicatable for single or splitted injection. A temperature-depending factor weighs the difference between this basic-value and lambda=1. The resulting lambda setpoint is used for both cylinder banks. LAMB_SP_CH is used during catalyst heating phase, LAMB_SP_CH_SO2P is used during desulfation if there was catalyst heating active before. Due to C_LAMB_SP_CH_MAN it is possible to set the lambda setpoint cylinder individual.

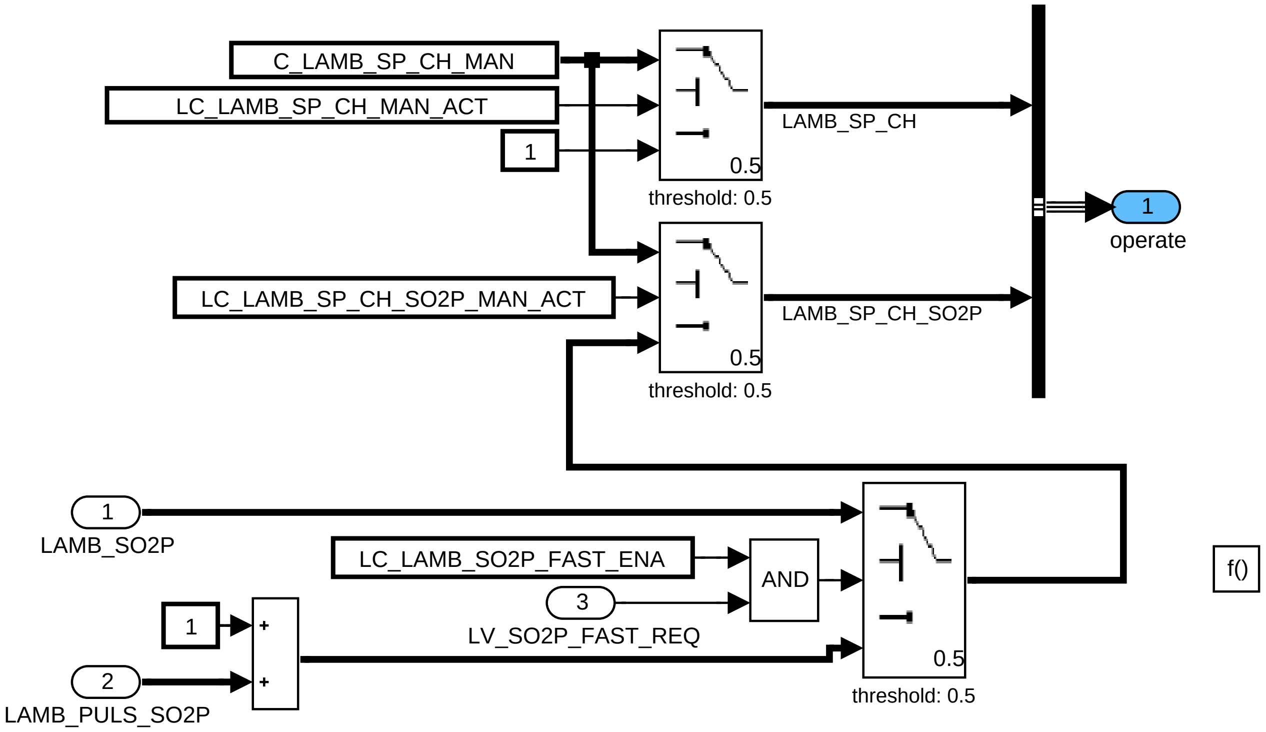


Figure 81:
Path: EXTC_REQGNCHL0/CH_SO2P/operate

30.15 Fuel mass setpoint of post injection for catalyst heating

Overview

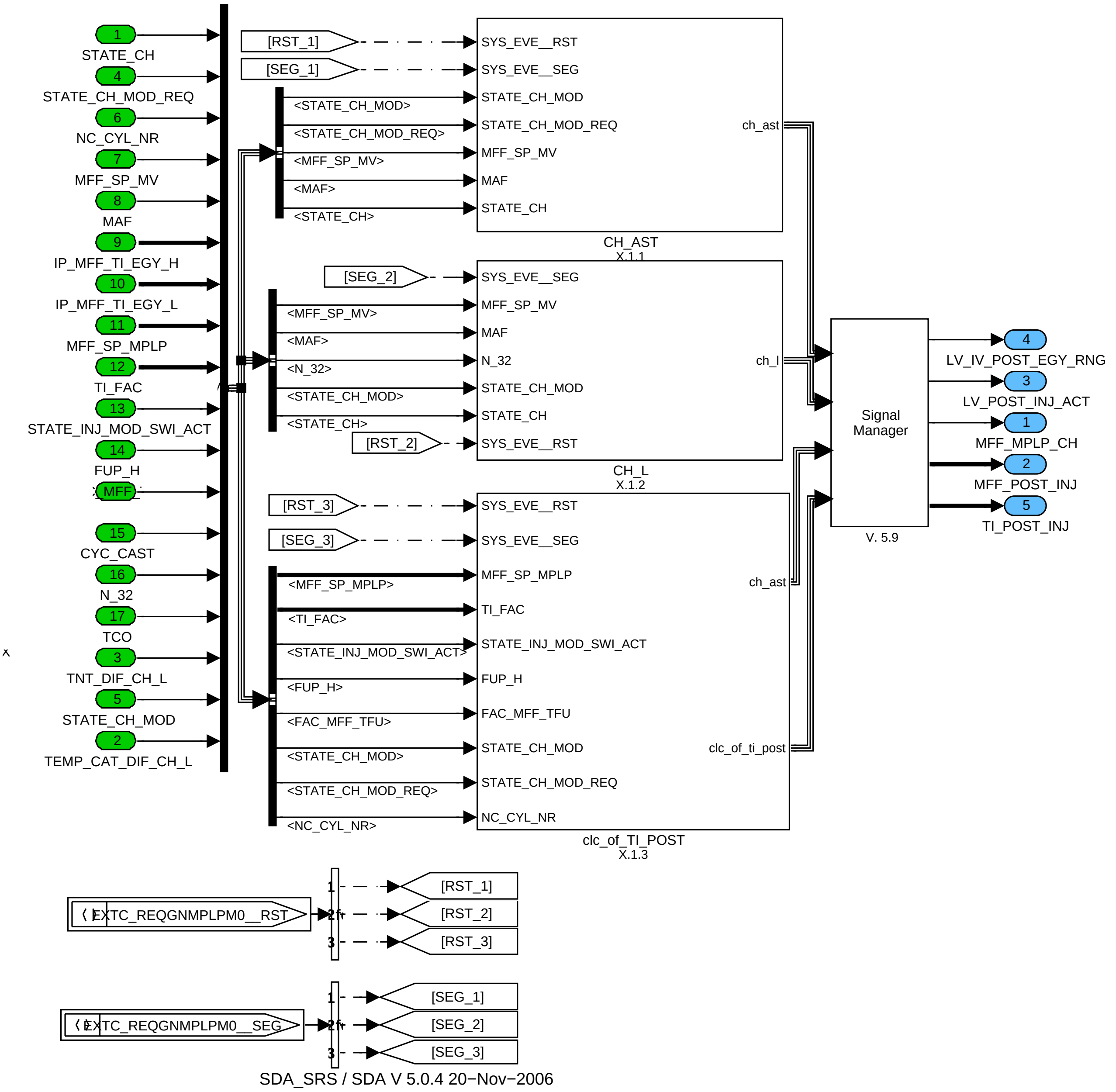


Figure 82:
Path: EXTC_REQGNMPLPM0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LAMB_CMB_S	V	0... FFFFH	0... 31.9995117188	488.281e-6	[-]
Lambda of stratified combustion - without post pulse					
LV_IV_POST_EGY_RNG	O/V	0... 1H	0... 1	1	[-]
Boolean variable indicating used piezo IV energy range for post injection					
LV_POST_INJ_ACT	O/V	0... 1H	0... 1	1	[-]
Boolean variable indicating the post injection is active					
MFF_MPLP_CH	O/V	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Fuel mass for post injection					
MFF_POST_INJ [NC_CYL_NR]	O/V	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Mass fuel flow of the post injection					
TI_POST_INJ [NC_CYL_NR]	O	0... FFFFH	0... 65.535	1e-3	[ms]
Injection time of the post injection					

Input Data:

STATE_CH	TEMP_CAT_DIF_CH_L	TNT_DIF_CH_L	STATE_CH_MOD_REQ
STATE_CH_MOD	NC_CYL_NR	MFF_SP_MV	MAF
IP_MFF_TI_EGY_H	IP_MFF_TI_EGY_L	MFF_SP_MPLP [NC_CYL_NR]	TI_FAC [NC_CYL_NR]
STATE_INJ_MOD_SWI_ACT	FUP_H	CYC_CAST	N_32
TCO	FAC_MFF_TFU		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_FAC_MFF_MPLP_CH_AST	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Correction factor depending on TCO and CYC_CAST for the mass fuel flow of the post injection					
C_FAC_MFF_MPLP_CH_AST_N_32	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Engine speed correction factor for the mass fuel flow of the post injection					
C_FAC_MFF_MPLP_CH_L	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Difference between cat. temperatures and target temp. depending correction for TQ_ADD_CH_L					
C_FAC_TI_POST_PRS_CYL	1	0... FFFFH	0... 15.9997558594	244.141e-6	[-]
Correction factor to compensate the influence of the cylinder counter pressure					
C_FUP_COR_POST_INJ	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Manual additive correction factor for injector pressure difference at post injection					
C_MAF_FAC_POST_INJ	1	0... FFFFH	0... 31.9995117188	488.281e-6	[-]
Constant for stoichiometric air/ fuel ratio (=14.7)					
C_MFF_MPLP_CH_AST	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Basic map for the mass fuel flow of the post injection for after start catalyst heating					
C_MFF_MPLP_CH_L	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Basic map for the mass fuel flow of the post injection for low load CH					
C_MFF_MPLP_CH_MAN	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Manual mass fuel flow for the post injection for catalyst heating					
IP_FAC_COR_POST_INJ	6	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDP_N_32_IP_FAC_COR_POST_INJ	6	0... FFH	0... 8160	32	[rpm]
Engine speed correction factor for the mass fuel flow of the post injection					

LC_IV_POST_EGY_RNG	1	0... 1H	0... 1	1	[-]
Switch to select energy range					
LC_MFF_MPLP_CH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Switch for manual mass fuel flow for the post injection for catalyst heating					

General Information

For catalyst heating an additional fuel mass pulse is injected after the stratified injected main combustion has occurred. In this module the mass of this so called post injection is generated. The module has to be calculated before Injection phase of post injection for catalyst heating , MAF correction for catalyst heating in stratified mode and Ignition angle difference for catalyst heating with post injection .

30.15.1 After start catalyst heating

Application Conditions

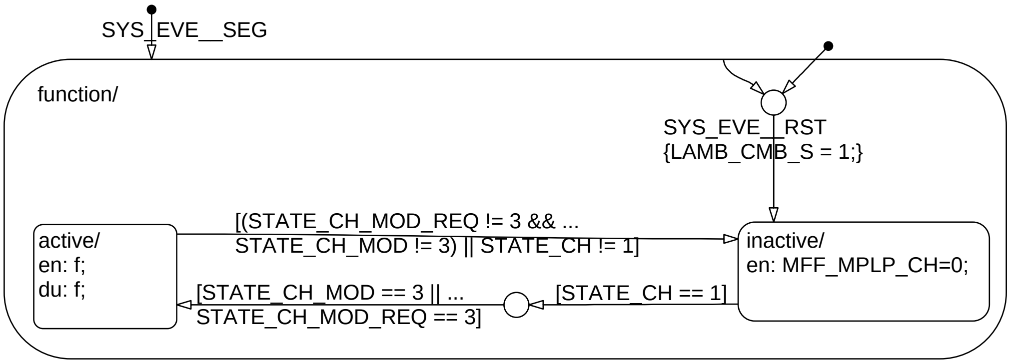


Figure 83:
Path: EXTC_REQGNMPLPM0/CH_AST/APP_CDN/Chart

Function description

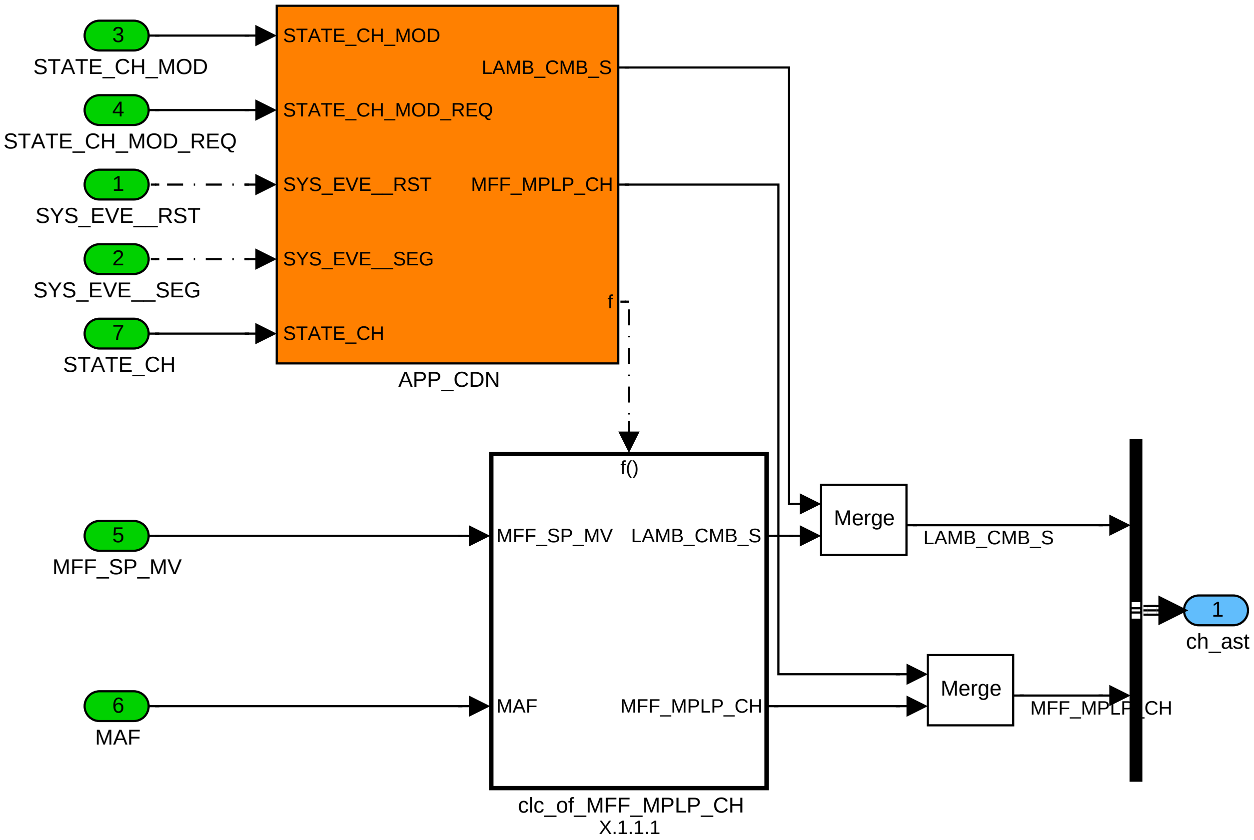


Figure 84:
Path: EXTC_REQGNMPLPM0/CH_AST

30.15.1.1 Calculation of MFF_MPLP_CH in after start catalyst heating

LAMB_CMB_S is the global lambda of the stratified main combustion without post injection. If MFF_SP is 0, LAMB_CMB_S should be set to its maximum limit. C_MAF_FAC_POST_INJ shell be 14.7 .

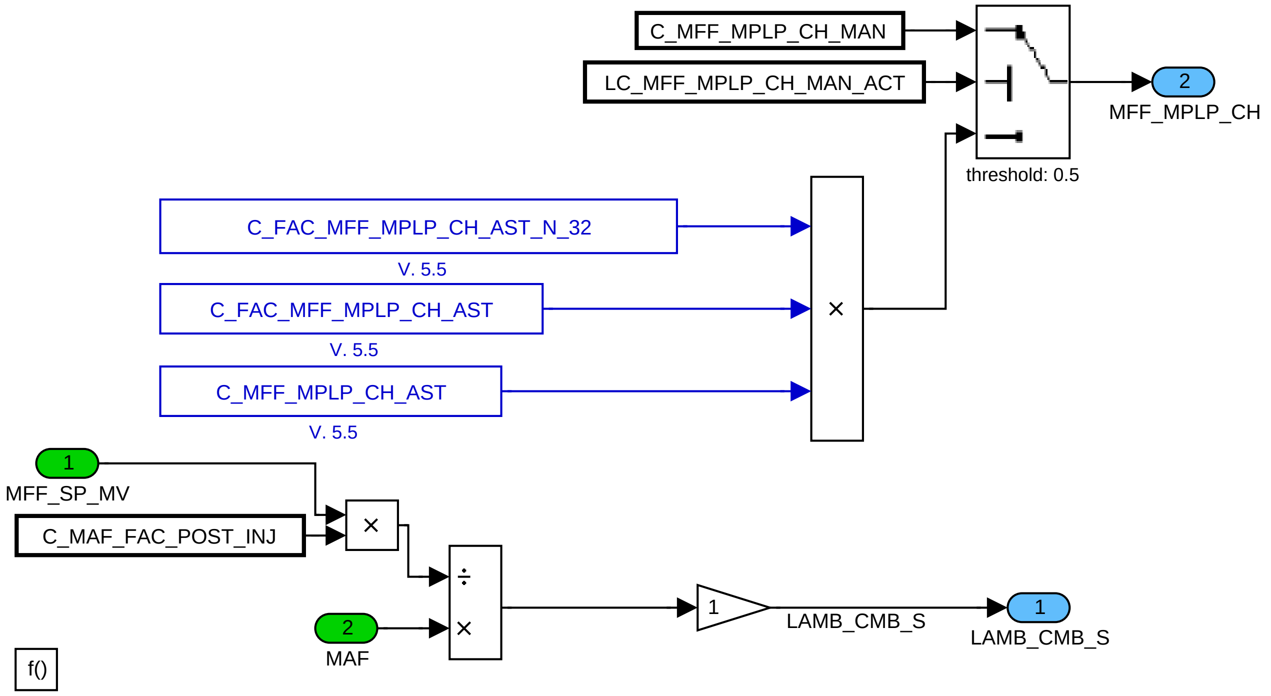


Figure 85:
Path: EXTC_REQGNMPLPM0/CH_AST/clc_of_MFF_MPLP_CH

30.15.2 Low load catalyst heating

Application Conditions

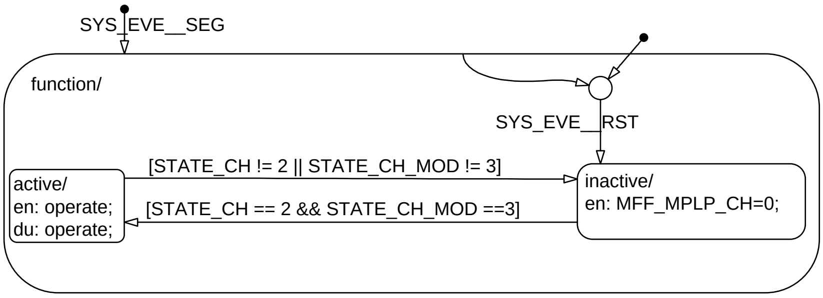


Figure 86:
Path: EXTC_REQGNMPLPM0/CH_L/APP_CDN/Chart

Function description

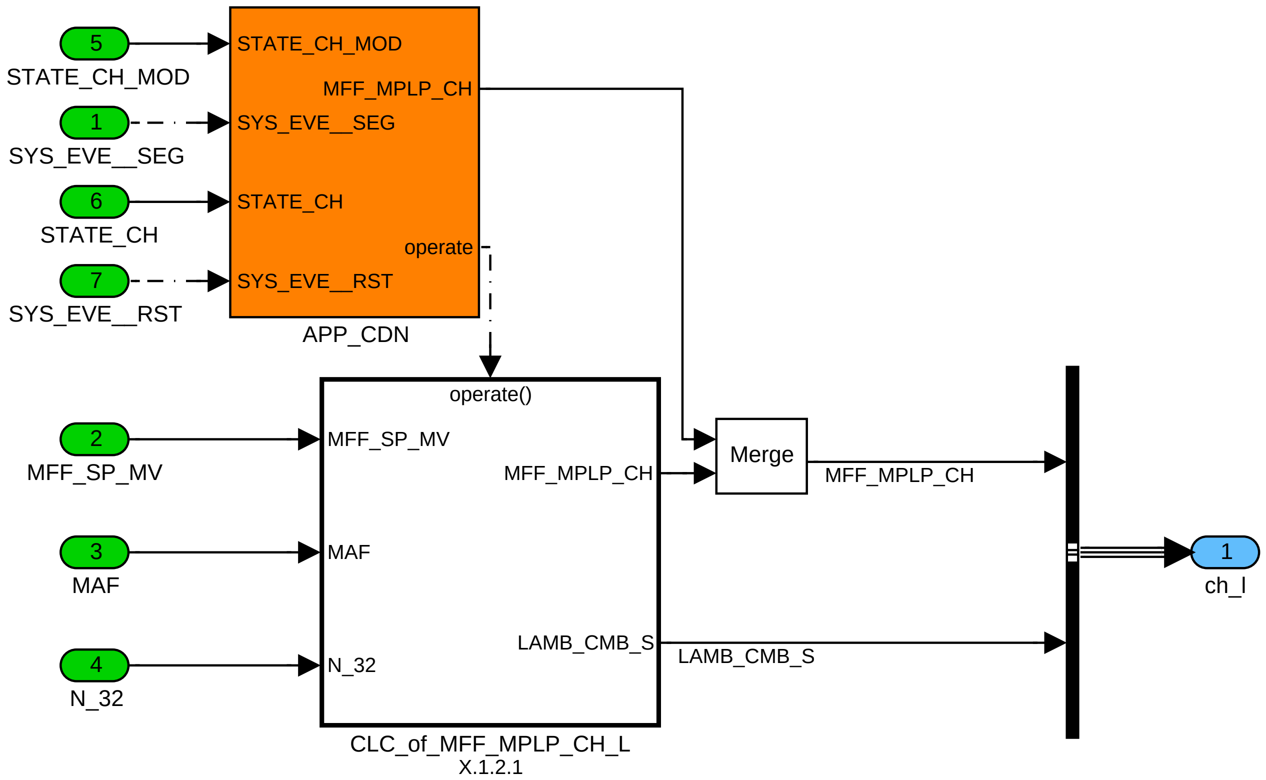


Figure 87:
Path: EXTC_REQGNMPLPM0/CH_L

30.15.2.1 Calculation of MFF_MPLP_CH in low load catalyst heating

LAMB_CMB_S is the global lambda of the stratified main combustion without post injection.

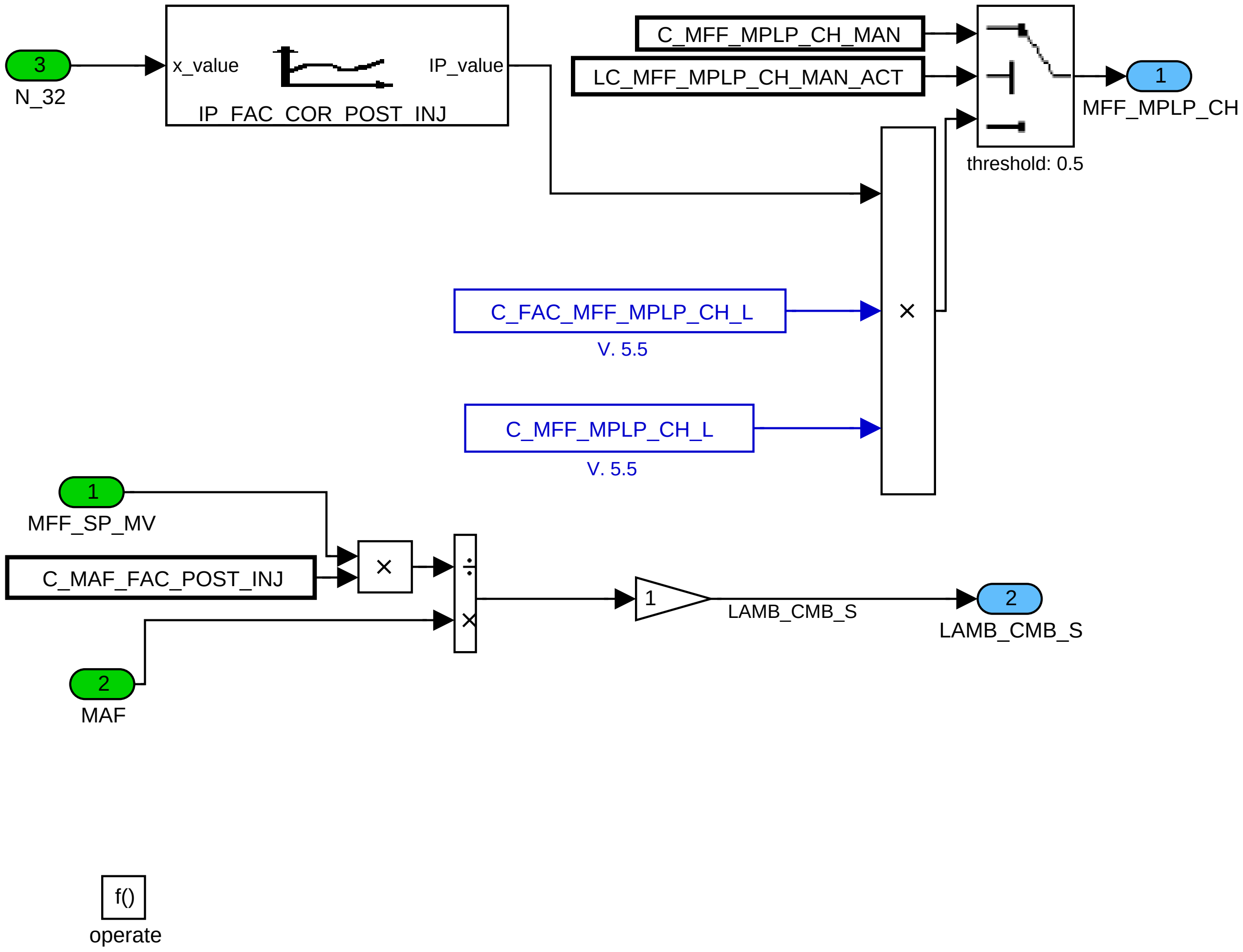


Figure 88:
Path: EXTC_REQGNMPLPM0/CH_L/CLC_of_MFF_MPLP_CH_L

30.15.3 Injection time of post injection

Application Conditions

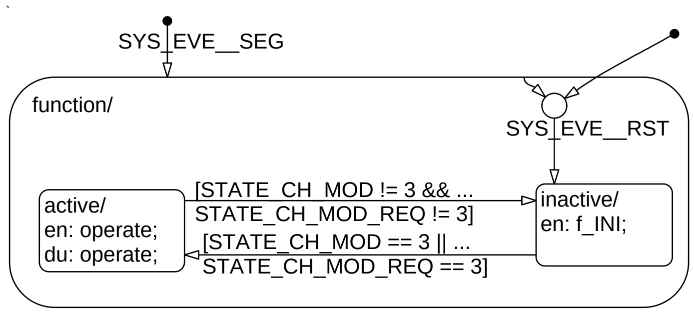


Figure 89:
Path: EXTC_REQGNMPLPM0/clc_of_TI_POST/APP_CDN/Chart

Function description

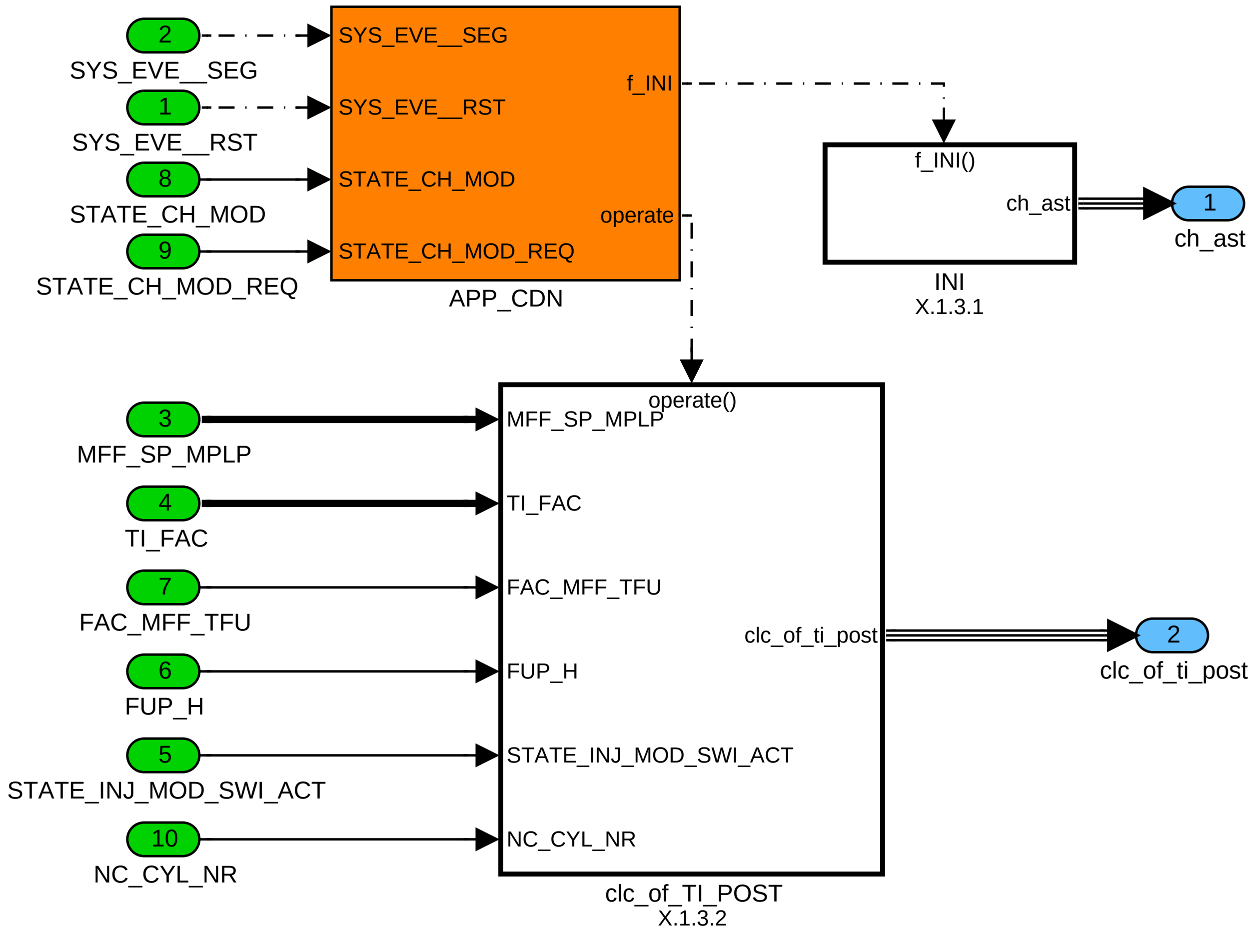


Figure 90:
Path: EXTC_REQGNMPLPM0/clc_of_TI_POST

30.15.3.1 Initialization

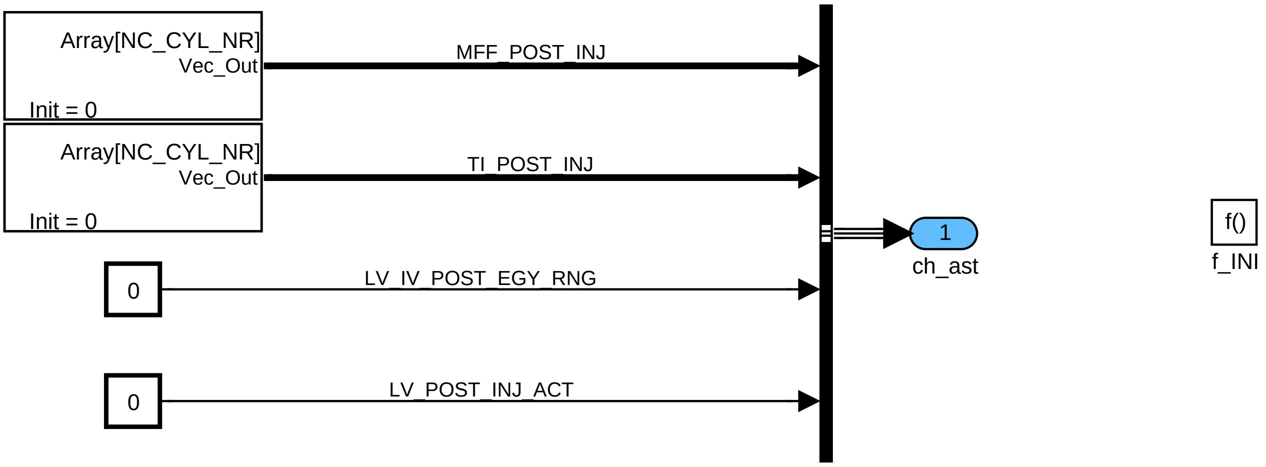


Figure 91:
Path: EXTC_REQGNMPLPM0/clc_of_TI_POST/INI

30.15.3.2 Calculation of post injection

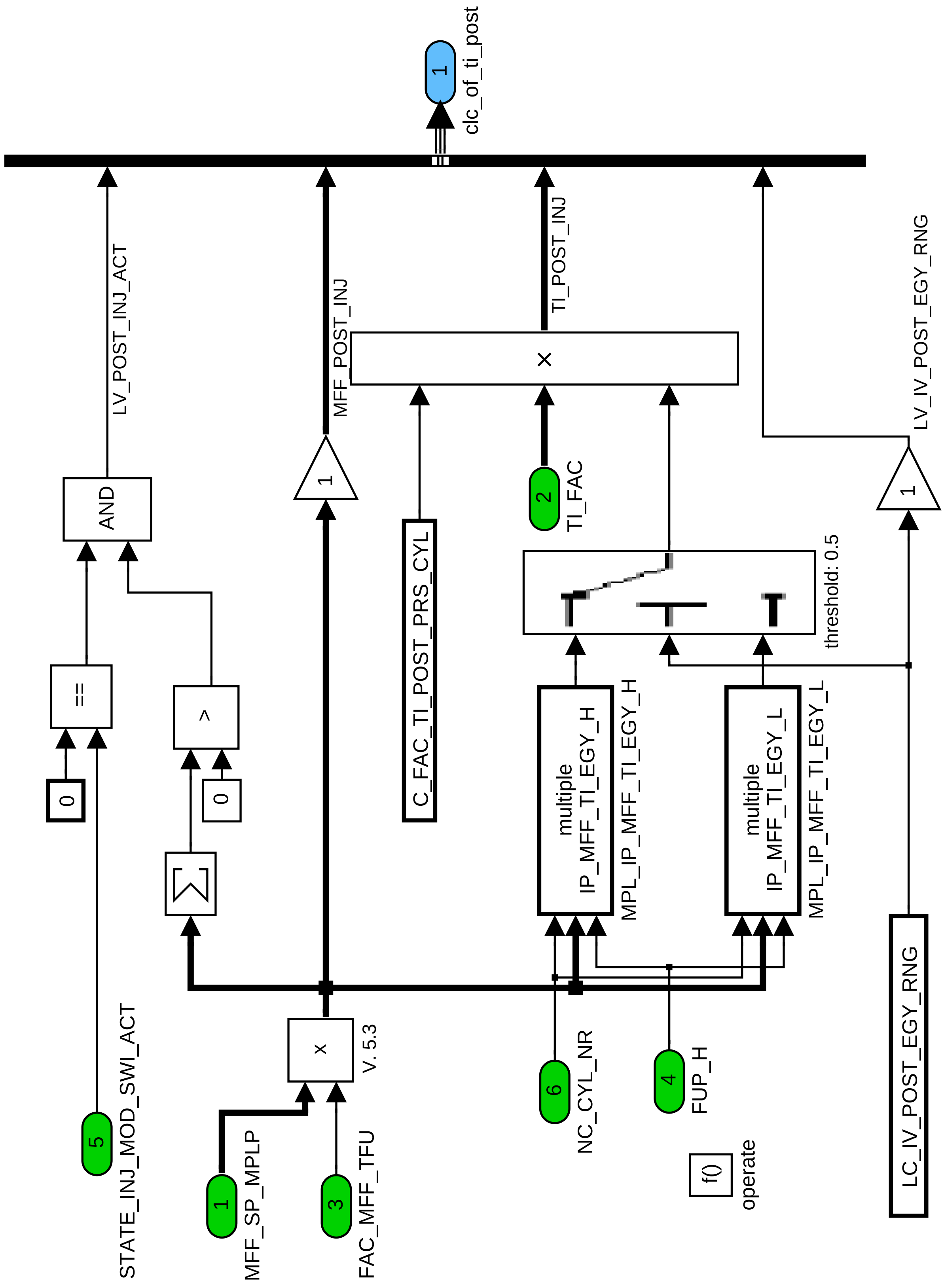
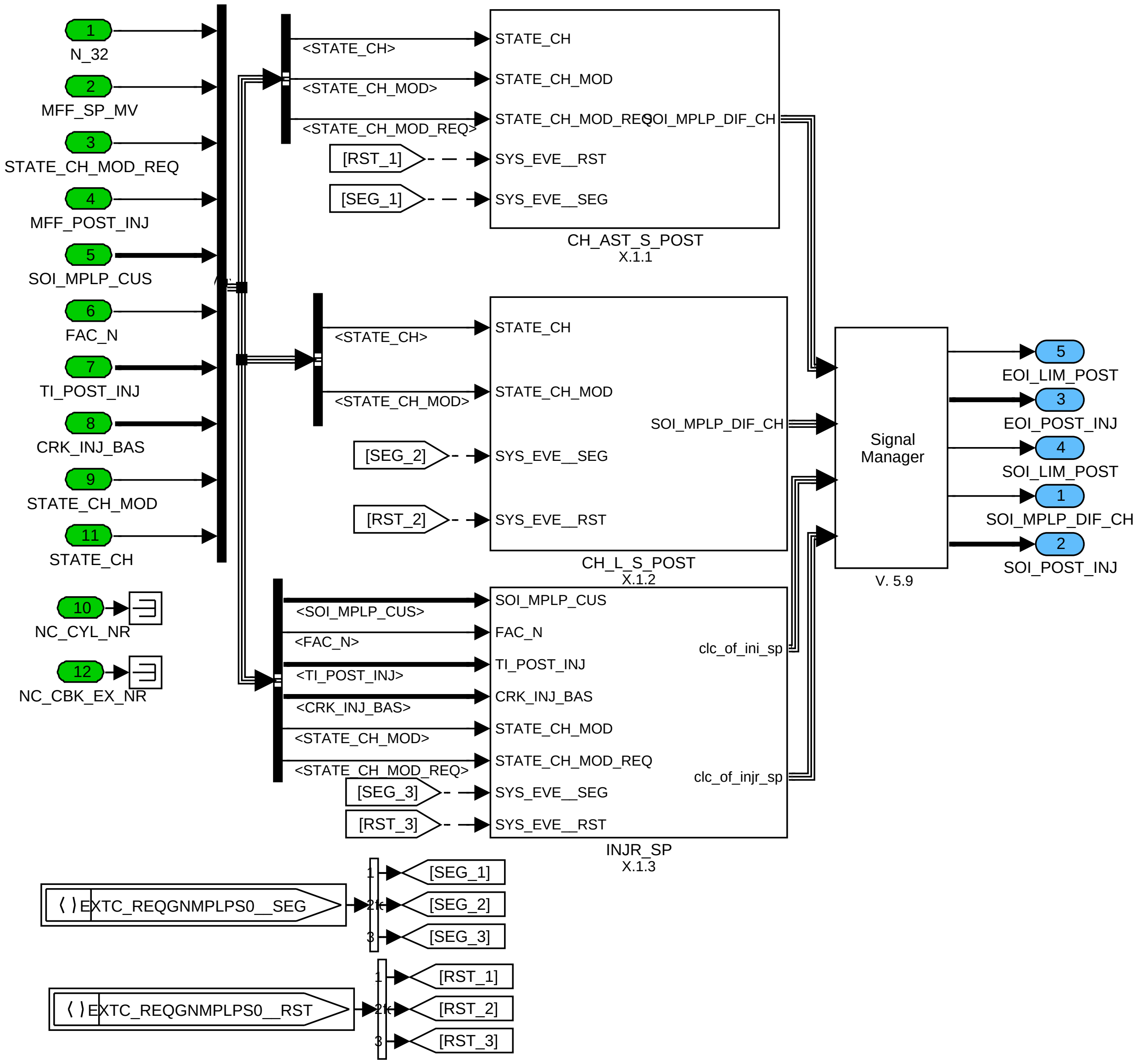


Figure 92:
Path: EXTC_REQGNMPLPM0/clc_of_TI_POST/clc_of_TI_POST

30.16 Injection phase of post injection for catalyst heating

Overview



SDA_SRS / SDA V 5.0.4 15−Nov−2006

Figure 93:
Path: EXTC_REQGNMPLPS0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
EOI_LIM_POST	O/V	0... 780H	0... 720	0.375	[°CRK]
Latest possible EOI for post injection					
EOI_POST_INJ [NC_CYL_NR]	O/V	0... 780H	0... 720	0.375	[°CRK]
End of the post injection					
SOI_LIM_POST	O/V	0... 780H	0... 720	0.375	[°CRK]
Earliest possible SOI for post injection					
SOI_MPLP_DIF_CH	O/V	0... 780H	0... 720	0.375	[°CRK]
Start of injection of the second injection for catalyst heating					
SOI_POST_INJ [NC_CYL_NR]	O/V	0... 780H	0... 720	0.375	[°CRK]
Start of the post injection to EOI_LIM					

Input Data:

N_32	MFF_SP_MV	STATE_CH_MOD_REQ	MFF_POST_INJ
SOI_MPLP_CUS [NC_CYL_NR]	FAC_N	TI_POST_INJ [NC_CYL_NR]	CRK_INJ_BAS [NC_CYL_NR]
STATE_CH_MOD	NC_CYL_NR	STATE_CH	NC_CBK_EX_NR

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_EOI_LIM_POST	1	0... 780H	0... 720	0.375	[°CRK]
Latest possible EOI for post injection relatet to EOI_LIM					
C_SOI_LIM_POST	1	0... 780H	0... 720	0.375	[°CRK]
Earliest possible SOI for post injection related to EOI_LIM					
C_SOI_MPLP_ADD_MFF_POST_CH_L	1	0... 780H	0... 720	0.375	[°CRK]
Mass of post injection depending correction of start of injection of post injection for low load strat. CH					
C_SOI_MPLP_DIF_CH_AST	1	0... 780H	0... 720	0.375	[°CRK]
Start of injection of the post injection for catalyst heating related to ignition angle					
C_SOI_MPLP_DIF_CH_L	1	0... 780H	0... 720	0.375	[°CRK]
Start of injection of post injection for low load stratified catalyst heating					
C_SOI_MPLP_MAN	1	0... 780H	0... 720	0.375	[°CRK]
Manual start of injection for the post injection					
C_SOI_POST_INJ_MAN	1	0... 780H	0... 720	0.375	[°CRK]
Manual start of injection for the post injection					
LC_SOI_MPLP_MAN_ACT	1	0... 1H	0... 1	1	[-]
Switch for manual start of injection for the post injection					
LC_SOI_POST_INJ_MAN_ACT	1	0... 1H	0... 1	1	[-]
Switch for manual start of injection for the post injection					

General Information

For catalyst heating an additional fuel mass pulse is injected after the stratified injected main combustion has occurred. This will increase the temperature in the exhaust system, especially in the catalyst. In this module the injection phase of this so called post injection is generated.
The function is splitted into 3 parts: the phasing for after start catalyst heating, the phasing for low load heating and the calculation of further setpoints for the INJR aggregate.
SOI_MPLP_DIF_CH is an input for BMWfunctions where SOI_3_S_CUS is the (eventually changed) output. SOI_POST_INJ and EOI_POST_INJ are the actually phasings.

SOI_MPLP_DIF_CH means the difference crankshaft angle between the ignition angle and the start of the post injection.

30.16.1 Post injection for after start catalyst heating

Application Conditions

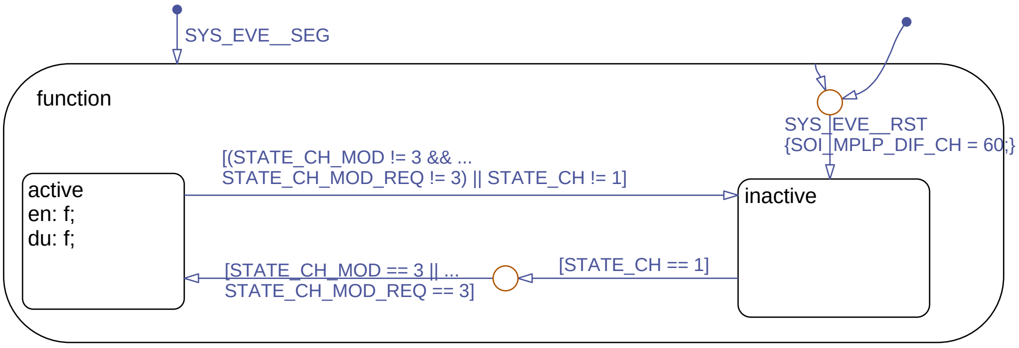


Figure 94:
Path: EXTC_REQGNMPLPS0/CH_AST_S_POST/APP_CDN/Chart

Function description

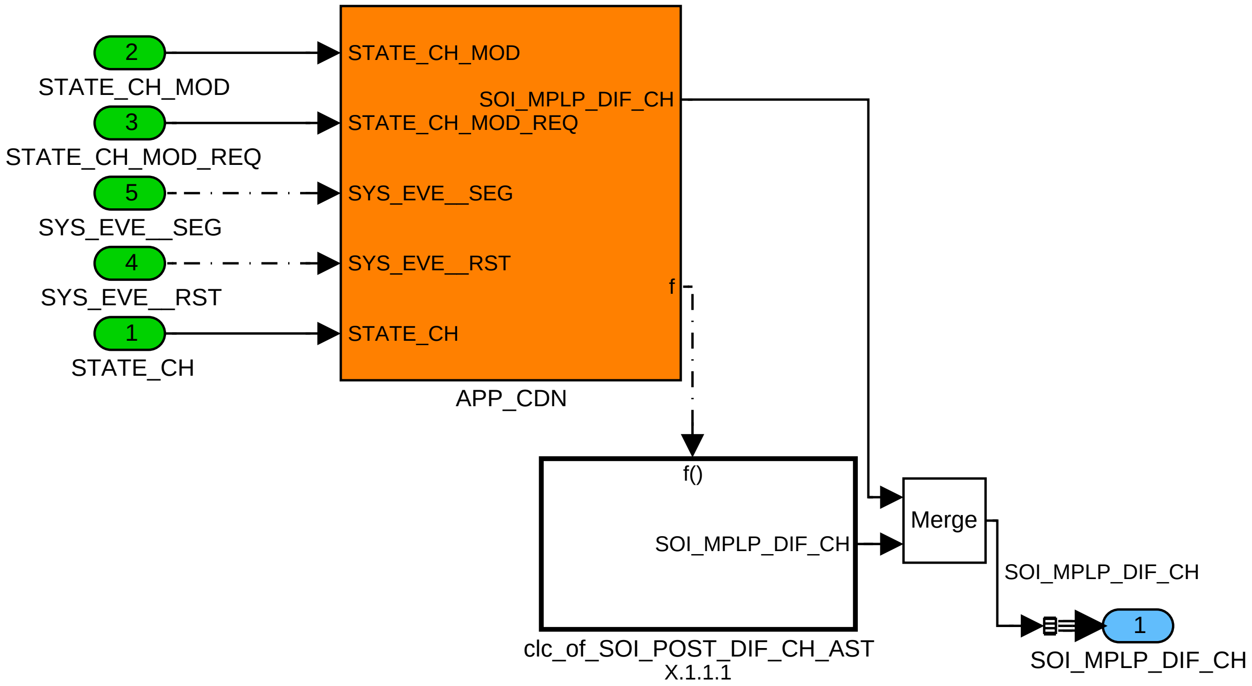


Figure 95:
Path: EXTC_REQGNMPLPS0/CH_AST_S_POST

30.16.1.1 Function: Post injection for after start catalyst heating

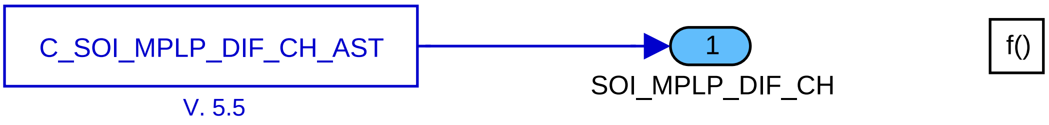


Figure 96:
Path: EXTC_REQGNMPLPS0/CH_AST_S_POST/clc_of_SOI_POST_DIF_CH_AST

30.16.2 Post injection for low load catalyst heating

Application Conditions

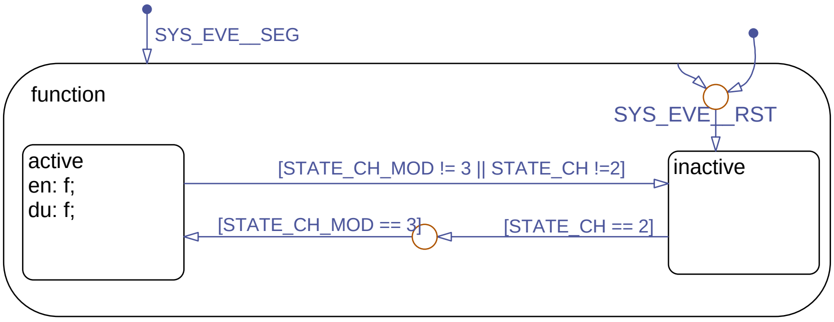


Figure 97:
Path: EXTC_REQGNMPLPS0/CH_L_S_POST/APP_CDN/Chart

Function description

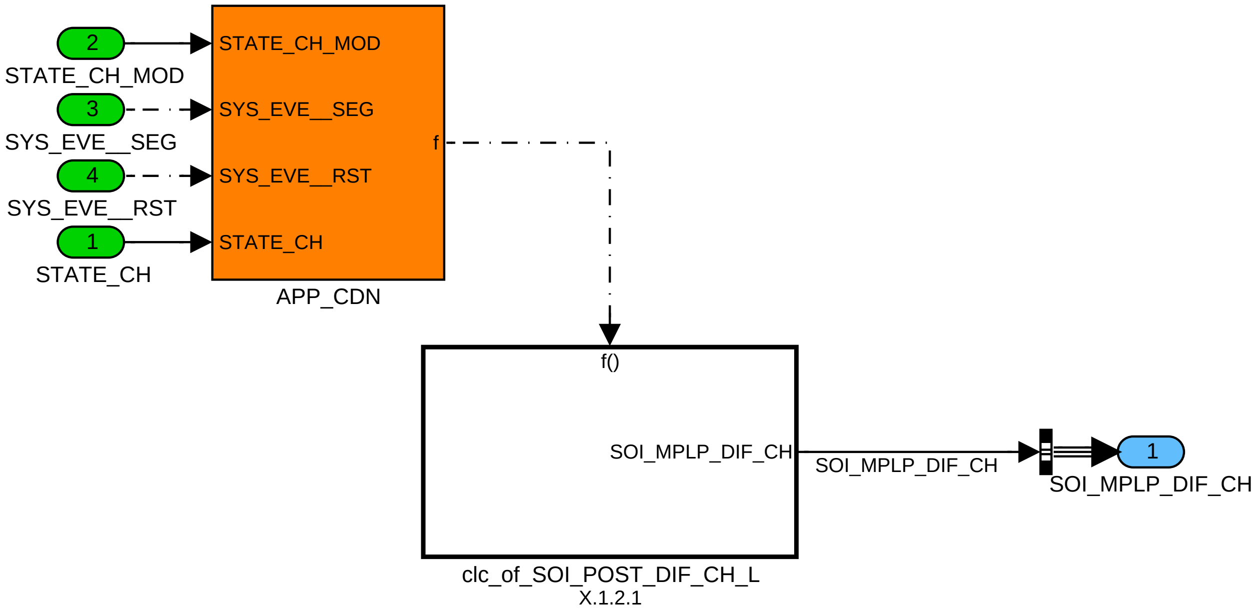


Figure 98:
Path: EXTC_REQGNMPLPS0/CH_L_S_POST

30.16.2.1 Function: Post injection for low load heating

For low load CH there is a second correction dependend on the actual fuel mass of post injection beside the CH_AST-functionality. The read_vector block picks out one element (here the 1st)of the cylinder indiviuell MFF_POST-vector.

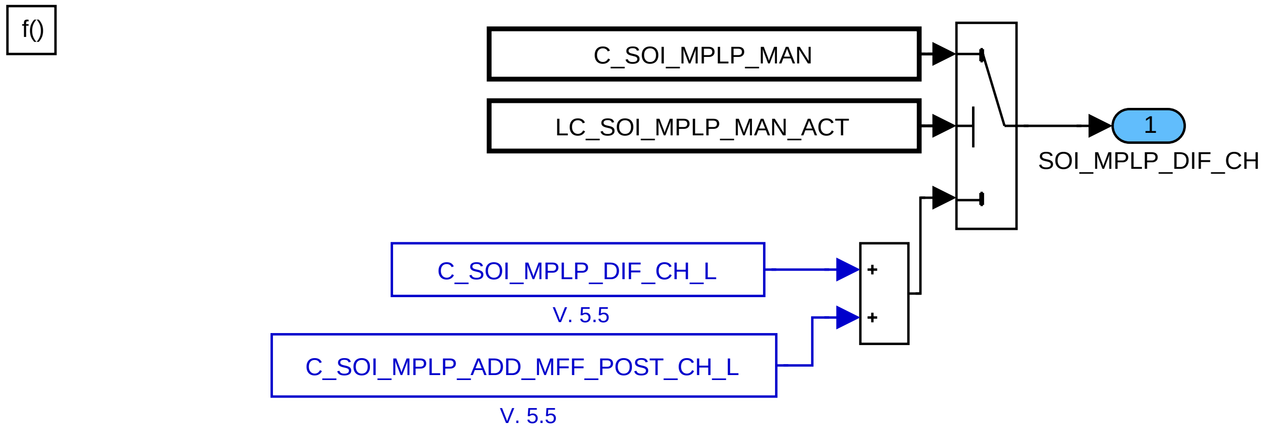


Figure 99:
Path: EXTC_REQGNMPLPS0/CH_L_S_POST/clc_of_SOI_POST_DIF_CH_L

30.16.3 Setpoint calculation for the INJR aggregate

Up to now the calculation of EOI_POST_INJ and the limitation values are not done in INJR (injection realization).

Application Conditions

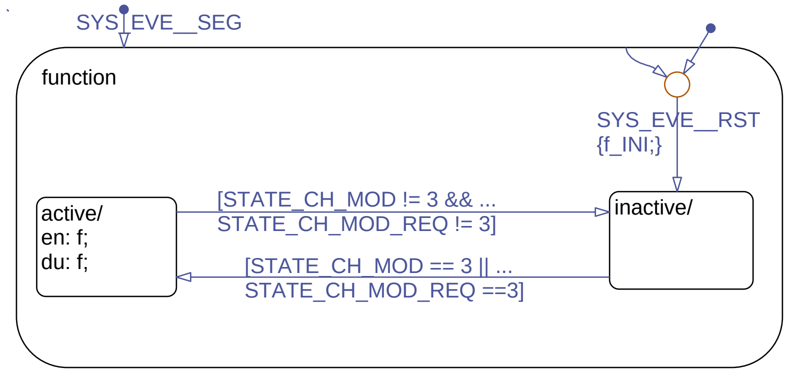


Figure 100:
Path: EXTC_REQGNMPLPS0/INJR_SP/APP_CDN/Chart

Function description

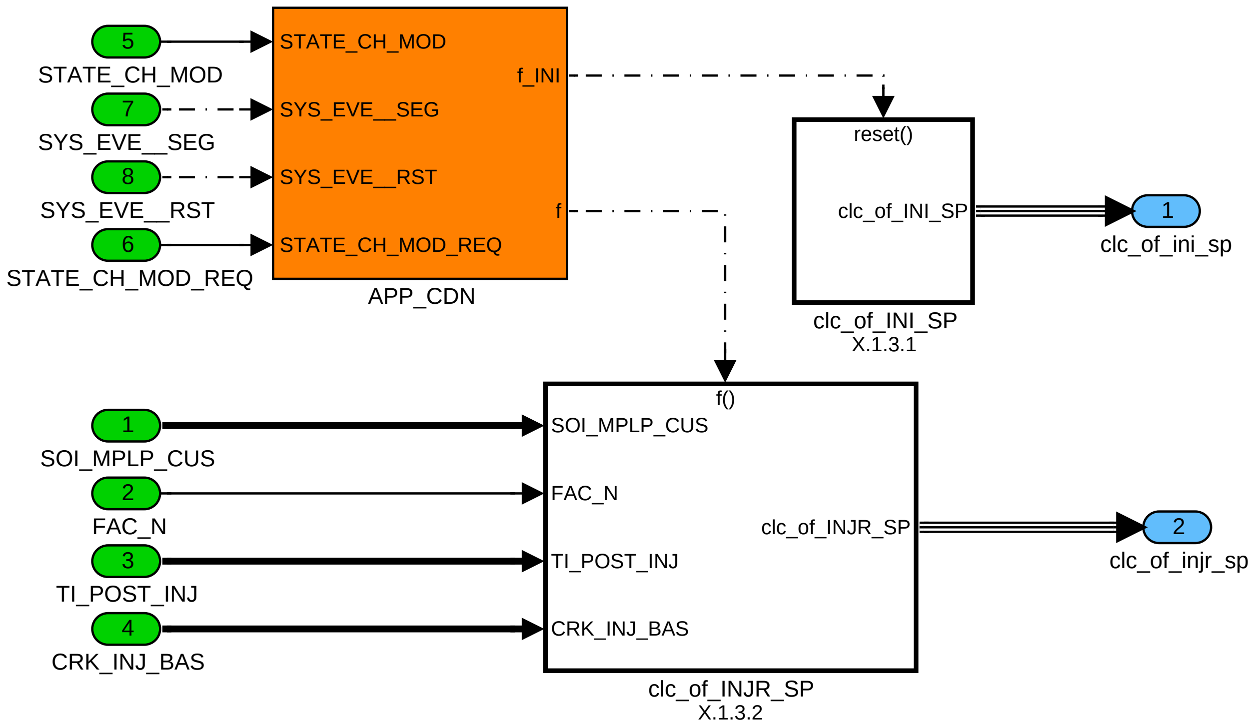


Figure 101:
Path: EXTC_REQGNMPLPS0/INJR_SP

30.16.3.1 Initialization

The limitations are initialized by the calibratable value. The set_vector block initializes the Vec_out with the value Init.

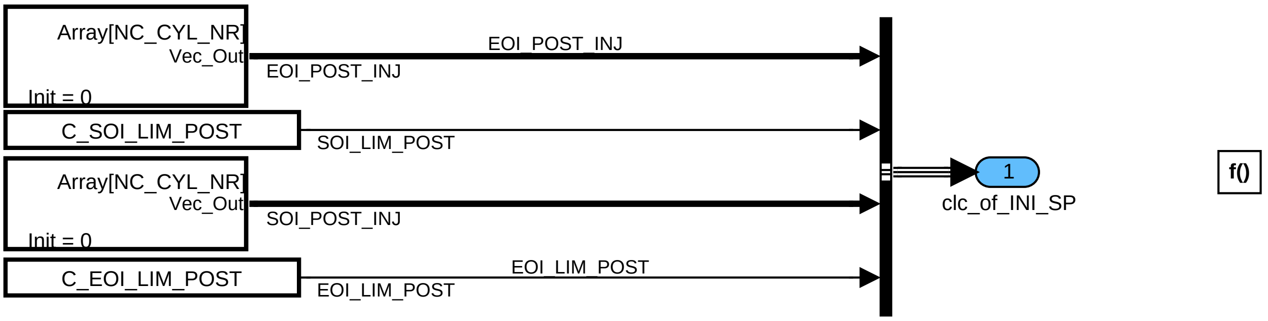


Figure 102:
Path: EXTC_REQGNMPLPS0/INJR_SP/clc_of_INI_SP

30.16.3.2 Function: Setpoint calculation for the INJR aggregate

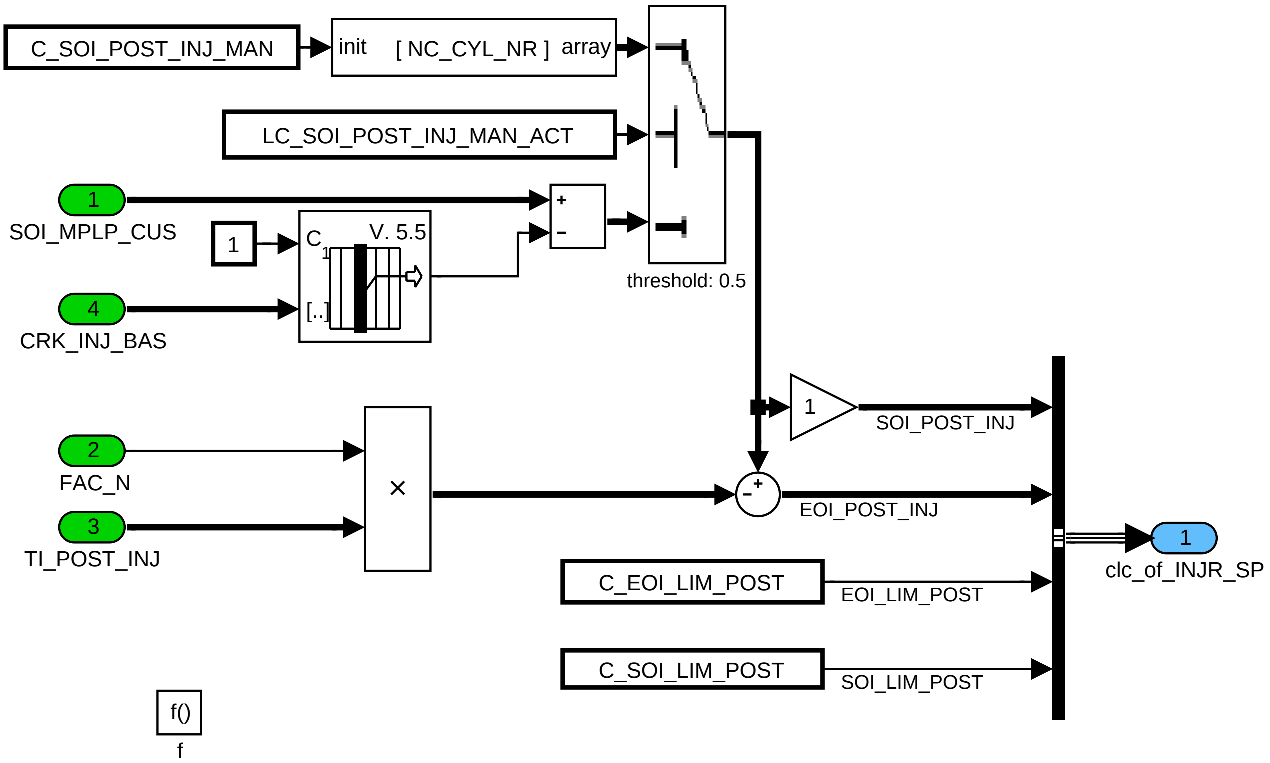


Figure 103:
Path: EXTC_REQGNMPLPS0/INJR_SP/clc_of_INJR_SP

30.17 Lambda Catalyst Overheating Prevention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_COP_CUS_i	V/O	0...7FFFh	0.. 31.9990234	9.8E-4	-
lambda for catalyst overheating prevention					
LAMB_TEG_CAT_UP_MDL_COP_i	V	0...7FFFh	0.. 31.9990234	9.8E-4	-
calculation value for catalyst overheating prevention upstream of the precatalyst					
LAMB_TEG_CAT_DOWN_COP_i	V	0...7FFFh	0.. 31.9990234	9.8E-4	-
calculation value for catalyst overheating prevention downstream the precatalyst					
LAMB_TNT_MDL_H_COP_i	V	0...7FFFh	0.. 31.9990234	9.8E-4	-
calculation value for catalyst overheating prevention in the NOx-trap catalyst					
LV_TEG_CAT_UP_MDL_COP_i	V	00...01h	0...1	1	-
flag for catalyst overheating prevention active					
LV_TEG_CAT_DOWN_COP_i	V	00...01h	0...1	1	-
flag for catalyst overheating prevention active					
LV_TNT_MDL_H_COP_i	V	00...01h	0...1	1	-
flag for catalyst overheating prevention active					
LV_LAMB_COP_CUS_i	V/O	00...01h	0...1	1	-
general flag for catalyst overheating prevention active and output					
TEG_CAT_DOWN_i	V	0 ... 7FF0H	0...2047	0.0625	°C
Exhaust gas temperature downstream the catalyst					

Input data:

LAMB_SP_i	LAMB_BAS_i		
TEG_CAT_UP_MDL_i	TEG_CAT_DOWN_MDL_i	TEG_PCAT_DOWN_i	TNT_MDL_H_i
LV_ST_END	LV_AST	LV_PUC	
LV_SCC_i			

FUNCTION DESCRIPTION:

Application conditions:

Recurrence: every 1 s.

Initialisation: at reset LAMB_COP_CUS_i = C_LAMB_COP_PAS

Activation: If LV_ST_END = 1

Deactivation: -

General information:

If the temperature in the exhaust line exceeds a certain limit, the catalytic converter or even the exhaust line can be destroyed. To reduce the temperature the lambda is ramped down.

The exhaust line of a gasoline direct injection engine consists of a precatalyst and a NO_X-trap catalyst. The actual temperatures upstream (TEG_CAT_UP_MDL_i) and downstream of the precatalyst (TEG_CAT_DOWN_MDL_i or TEG_PCAT_DOWN_i, respectively) and the temperature in the NO_X-trap (TNT_MDL_H_i) are compaired to thresholds, separate for each temperature. The difference between an actual temperature and a threshold is input for an I-controller. Three different I-controllers are working in parallel.

Descriptions of LAMB_COP_i is the output LAMB_COP_CUS_i

In the following, **xx** serves as substitute for specification purposes. This is in detail:

xx = TEG_CAT_UP_MDL	for I-controller 1 (temperature upstream of the precatalyst)
xx = TEG_CAT_DOWN	for I-controller 2 (temperature downstream of the precat.)
xx = TNT_MDL_H	for I-controller 3 (temperature in the NO_X-trap catalyst)

Formula section:

30.17.1 Maximal selection between model or sensor based temperatur downstream the precatalyst

If a temperature sensor downstream the precatalyst is available the maximum value between the sensor based temperature TEG_PCAT_DOWN_i and the the model based temperature TEG_CAT_DOWN_MDL_i is choosen.

TEG_CAT_DOWN_i = MAX (TEG_PCAT_DOWN_i, TEG_CAT_DOWN_MDL_i)

Note:

If no temperature sensor downstream the precatalyst is available the variable has to be initialized as follows:

TEG_PCAT_DOWN_i = 0

30.17.2 Activation and Deactivation of the Controller

A single controller is activated (LV_xx_COP_i = 1) if the temperature xx_i exceeds its threshold C_xx_THD_COP. A singel controller is deactivated (LV_xx_COP_i = 0) if the temperature xx_i is below the threshold C_xx_THD_MIN_COP or if the temperature xx_i is below the threshold C_xx_THD_COP after the lambda was ramped up and has reached the the basic lambda setpoint LAMB_BAS_i.

All controllers are deactivated if one of the following logical variables is set to one:

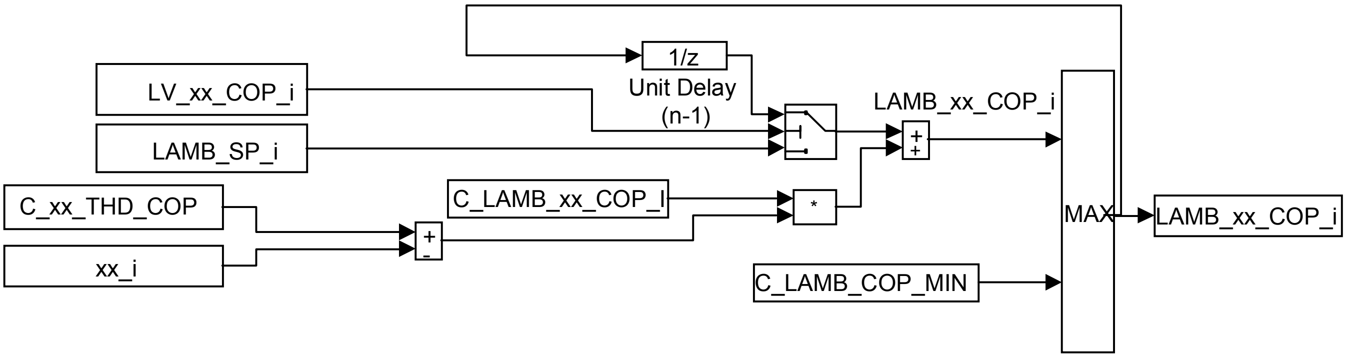
LV_AST, LV_PUC, LV_SCC_i.

For that case the lambda for catalyst overheat protection LAMB_COP_i is set to the passive value C_LAMB_COP_PAS.

```
(1) If      LV_AST = 1 or LV_PUC = 1or LV_SCC_i = 1
(1) then    LV_xx_COP_i = 0
(1) else
            (2) if      xx_i >= C_xx_THD_MIN_COP
            (2) then
                        (3) if      xx_i >= C_xx_THD_COP
                        (3) then    LV_xx_COP_i = 1
                        (3) else
                                    (4) if      LAMB_xx_COP_i > LAMB_BAS_i
                                    (4) then    LV_xx_COP_i = 0
                                    (4) endif
                        (3) endif
            (2) else  LV_xx_COP_i = 0
            (2) endif
(1) endif
```

At transition LV_xx_COP_i = 0 -> 1 the variable LAMB_xx_COP_i is initialized with LAMB_SP_i .

30.17.3 Controller Equations



```
If      LV_xx_COP_i = 1                                ...... (controller active)
then
LAMB_xx_COP_i (n) = LAMB_xx_COP_i (n-1) + C_LAMB_xx_COP_I *  (C_xx_THD_COP – xx_i)
else   LAMB_xx_COP_i = C_LAMB_COP_PAS            ...... (controller inactive)
endif
```

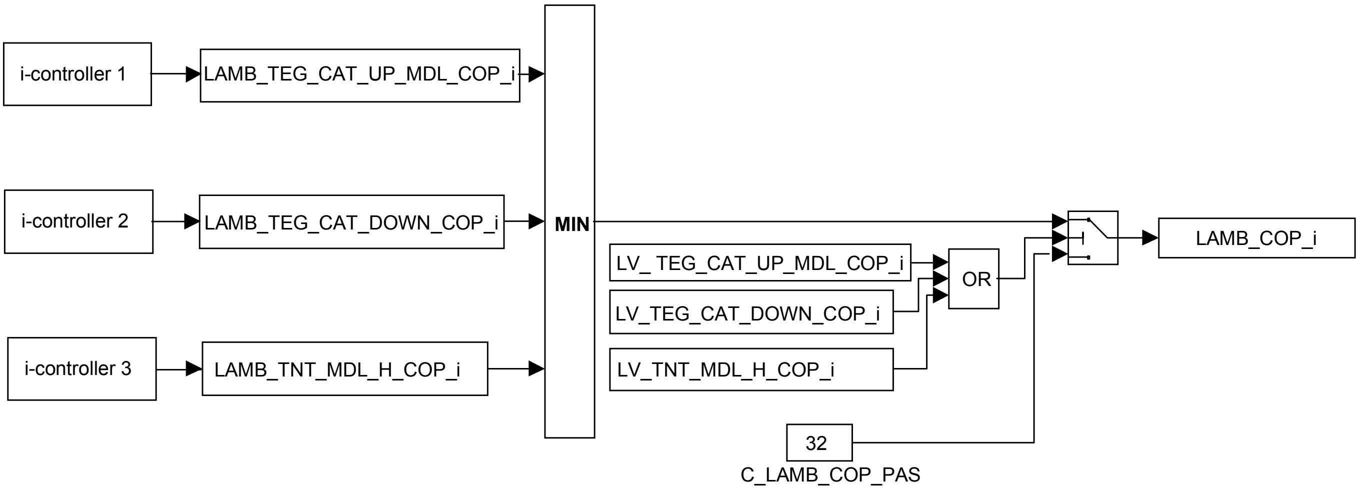
Application Hint: The passive value C_ LAMB_COP_PAS shoud be set to 32.

The value LAMB_xx_COP_i can not be smaller than the minimum allowed lambda value C_LAMB_COP_MIN:

LAMB_xx_COP_i = MAX (LAMB_xx_COP_i, C_LAMB_COP_MIN)

30.17.4 Calculation of LAMB_COP (Resulting lambda request)

System flow diagramm:



If at least one controller is active (LV_xx_COP_i = 1) the minimum controller output is selected for LAMB_COP_i:

```
if      LV_TEG_CAT_UP_MDL_COP_i = 1
        or
        LV_TEG_CAT_DOWN_COP_i = 1
        or
        LV_TNT_MDL_H_COP_i = 1
then    LAMB_COP_i = MIN (LAMB_xx_COP_i)
        LV_LAMB_COP_i = 1
else    LAMB_COP_i = C_LAMB_COP_PAS
        LV_LAMB_COP_i = 0
endif
```

If one of the temperatures **xx_i** exceeds the maximum threshold C_xx_THD_MAX_COP, the lambda enrichment is set to the minimum allowed lambda value C_LAMB_COP_MIN immediately.

```
If
TEG_CAT_UP_MDL_i >= C_TEG_CAT_UP_MDL_THD_MAX_COP
or
TEG_CAT_DOWN_MDL_i >= C_TEG_CAT_DOWN_THD_MAX_COP
or
```

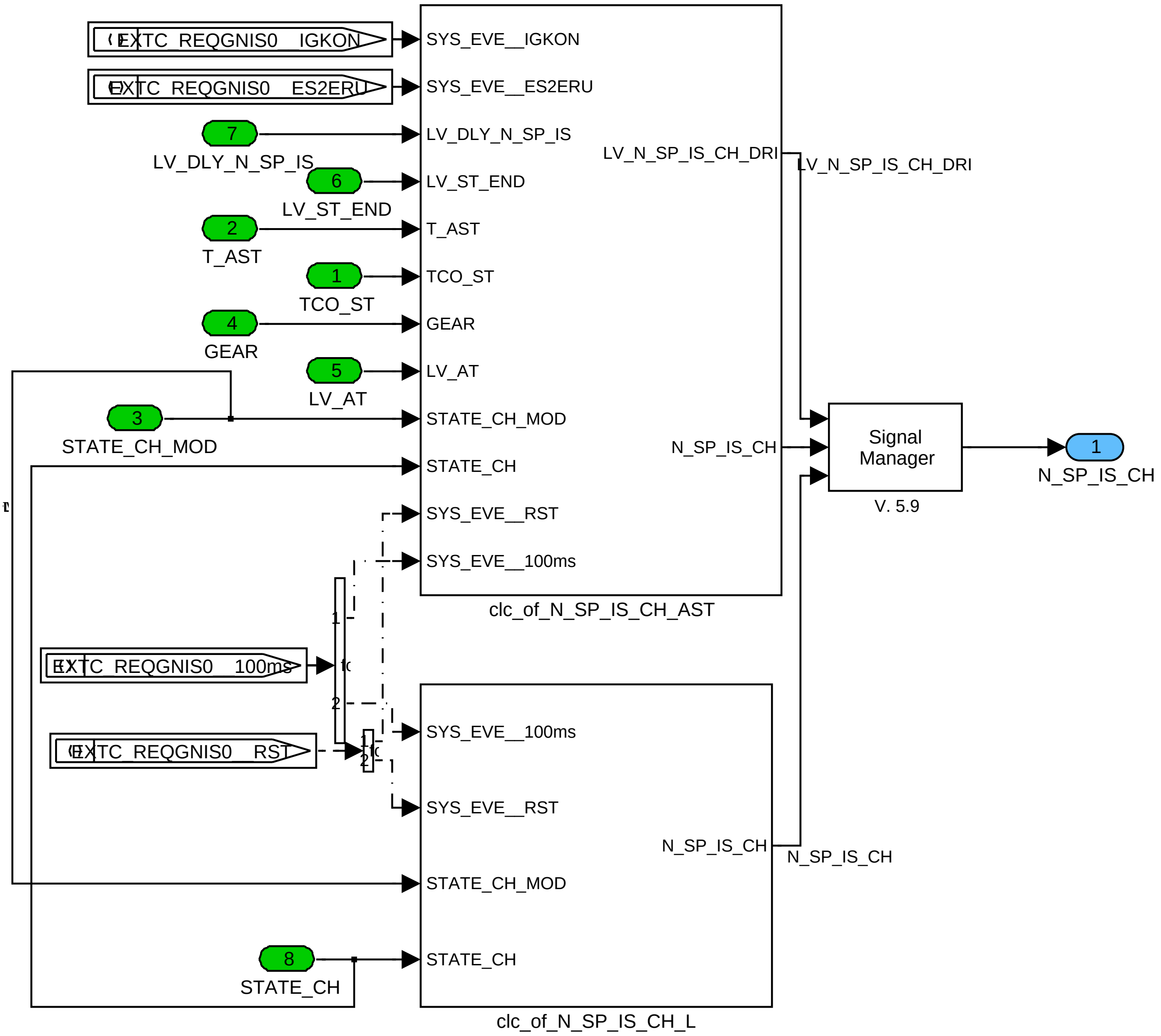
```
TNT_MDL_H_i >= C_TNT_MDL_H_THD_MAX_COP

then LAMB_COP_i = C_LAMB_COP_MIN
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_COP_MIN	1	0...7FFFh	0.31.9990234	9.8e-4	-
minimum value for catalyst overheating protection					
C_LAMB_COP_PAS	1	0...7FFFh	0.31.9990234	9.8e-4	-
passive value for the lambda catalyst overheating prevention					
C_LAMB_TEG_CAT_UP_MDL_COP_I	1	0...FFFFh	0... 0.003922	5.98e-8	1/°C
I-part of controler					
C_LAMB_TEG_CAT_DOWN_COP_I	1	0...FFFFh	0... 0.003922	5.98e-8	1/°C
I-part of controler					
C_LAMB_TNT_MDL_H_COP_I	1	0...FFFFh	0... 0.003922	5.98e-8	1/°C
I-part of controler					
C_TEG_CAT_UP_MDL_THD_MIN_COP	1	0...7FF0h	0... 2047	0.0625	°C
bottom threshold for exhaust gas temperature upstream of the precatalyst					
C_TEG_CAT_DOWN_THD_MIN_COP	1	0..7FF0H	0..2047	0.0625	°C
bottom threshold for exhaust gas temperature downstream of the precatalyst					
C_TNT_MDL_H_THD_MIN_COP	1	0...FFFFh	0... 1024	0.015625	°C
bottom threshold for temperature in the NOx Trap catalyst					
C_TEG_CAT_UP_MDL_THD_COP	1	0..7FF0H	0..2047	0.0625	°C
threshold for exhaust gas temperature upstream of the precatalyst – I-controller activation					
C_TEG_CAT_DOWN_THD_COP	1	0..7FF0H	0..2047	0.0625	°C
threshold for exhaust gas temperature downstream of the precatalyst – I-controller activation					
C_TNT_MDL_H_THD_COP		0..FFFFH	0..1024	0.015625	°C
threshold for temperature in the NOx Trap catalyst – I-controller activation					
C_TEG_CAT_UP_MDL_THD_MAX_COP	1	0..7FF0H	0..2047	0.0625	°C
Maximum threshold for exhaust gas temperature upstream of the precatalyst					
C_TEG_CAT_DOWN_THD_MAX_COP	1	0..7FF0H	0..2047	0.0625	°C
Maximum threshold for exhaust gas temperature downstream of the precatalyst					
C_TNT_MDL_H_THD_MAX_COP		0..FFFFH	0..1024	0.015625	°C
maximum threshold for temperature in the NOx Trap catalyst					

30.18 Idle speed setpoint for catalyst heating

Overview



SDA_SRS / SDA V 5.0.4 07−Dec−2006

Figure 104:
Path: EXTC_REQGNIS0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_N_SP_IS_CH_DRI	V	0... 1H	0... 1	1	[-]
Logical variable for driving position engaged (switch for N_SP_IS_CH maps)					
N_SP_IS_CH	O/V	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint for catalyst heating					

Input Data:

TCO_ST	T_AST	STATE_CH_MOD	GEAR
LV_AT	LV_ST_END	LV_DLY_N_SP_IS	STATE_CH

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_N_SP_IS_CH_AST_MPLP	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint for stratified catalyst heating with post injection					
C_N_SP_IS_CH_AST_MPLP_DRI	1	0... 1FE0H	0... 8160	1	[rpm]
idle speed setpoint for stratified catalyst heating with post injection, engaged gear					
C_N_SP_IS_CH_L_HOM	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint for homogenous low load CH					
C_N_SP_IS_CH_L_S	1	0... 1FE0H	0... 8160	1	[rpm]
Idle speed setpoint for stratified low load CH					
IP_N_SP_IS_CH_AST_HOM	6*8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_T_AST_5_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_5_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for homogeneous catalyst heating					
IP_N_SP_IS_CH_AST_HOM_DRI	6*8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_T_AST_5_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_5_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for homogeneous catalyst heating, engaged gear					
IP_N_SP_IS_CH_AST_HOM_DRI_REV	6*8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_T_AST_5_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_5_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for homogeneous catalyst heating, engaged reverse gear					
IP_N_SP_IS_CH_AST_MPLH	6*8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_T_AST_5_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_5_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for homogeneous catalyst heating with multiple injection					
IP_N_SP_IS_CH_AST_MPLH_DRI	6*8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_T_AST_5_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_5_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for homogeneous catalyst heating with multiple injection, engaged gear					
IP_N_SP_IS_CH_AST_MPLH_DRI_REV	6*8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_T_AST_5_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_5_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for homogeneous catalyst heating with multiple injection, engaged reverse gear					
IP_N_SP_IS_CH_AST_MPLP_DRI_REV	6*8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_T_AST_5_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_5_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Idle speed setpoint for stratified catalyst heating with post injection, engaged reverse gear					
LC_DRI_ACT_MT	1	0... 1H	0... 1	1	[-]
Usage of gear-dependency of N_SP_IS_CH-maps for manual transmision					

General Information

For catalyst heating an increased idle speed is needed. In this modul there is a setpointcalculation for each strategy which is only calculated if actually needed.

N_SP_IS_CH is initialized with 0 because N_SP_IS is generated by a maximum-selection.

30.18.1 After start catalyst heating

There are several maps for engaged gear and not engaged. Via LC it is possible to use this dependency also for manual transmission.

Application Conditions

Function description

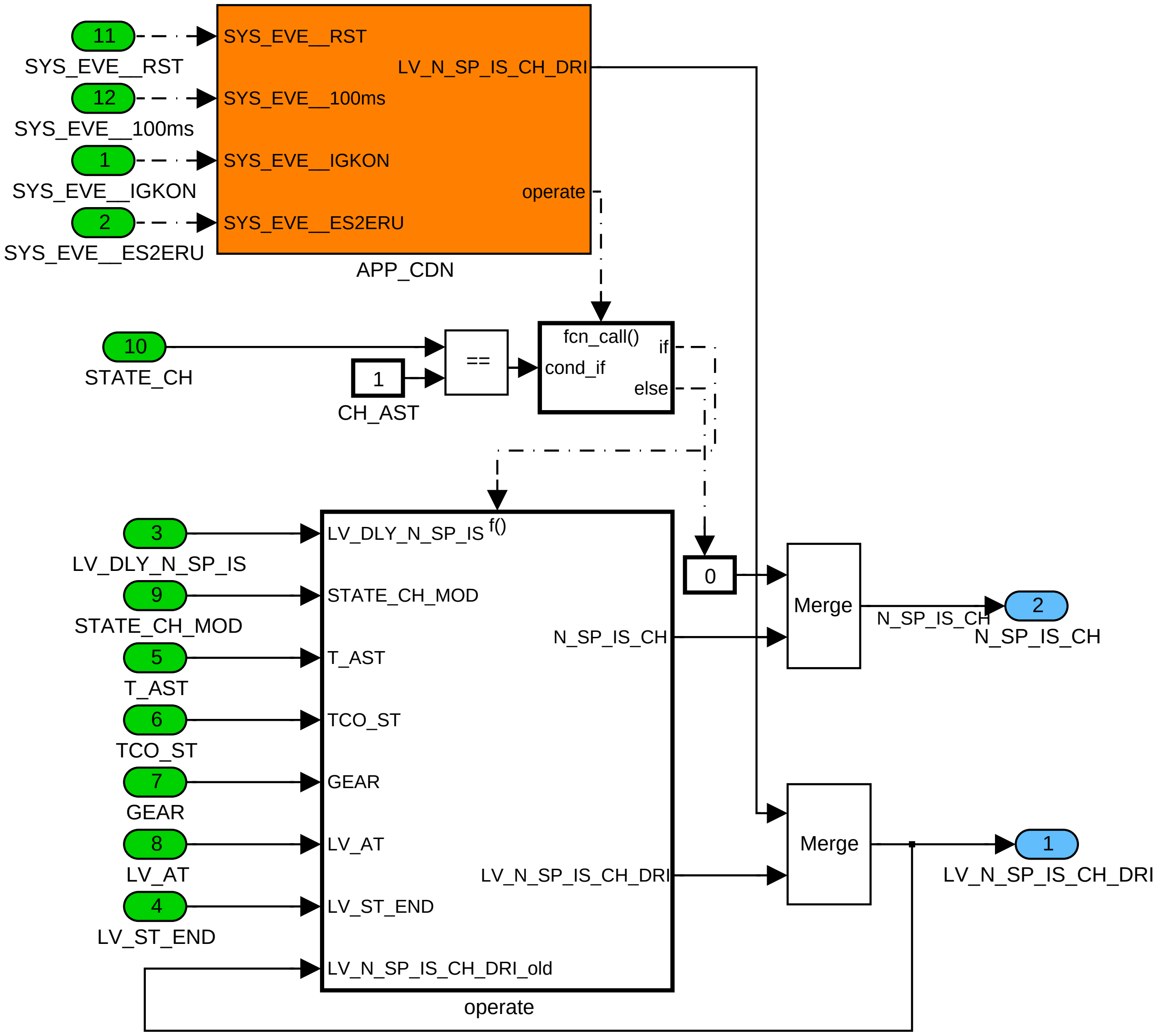


Figure 105:
Path: EXTC_REQGNIS0/clc_of_N_SP_IS_CH_AST

30.18.1.1 Operate subsystem: after start catalyst heating

For Gear > 0 there are several maps. With LC_DRI_ACT_MT it is possible to use the duplicated maps also for manual transmission. If these maps are used once they will be used until next engine start.

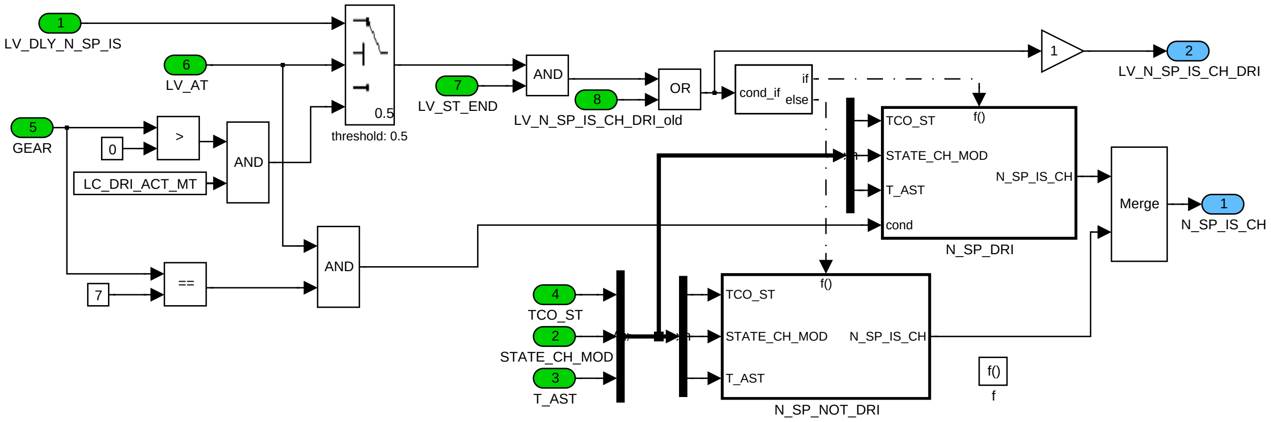


Figure 106:
Path: EXTC_REQGNIS0/clc_of_N_SP_IS_CH_AST/operate

30.18.1.1.1 Calculation of N_SP_IS_CH

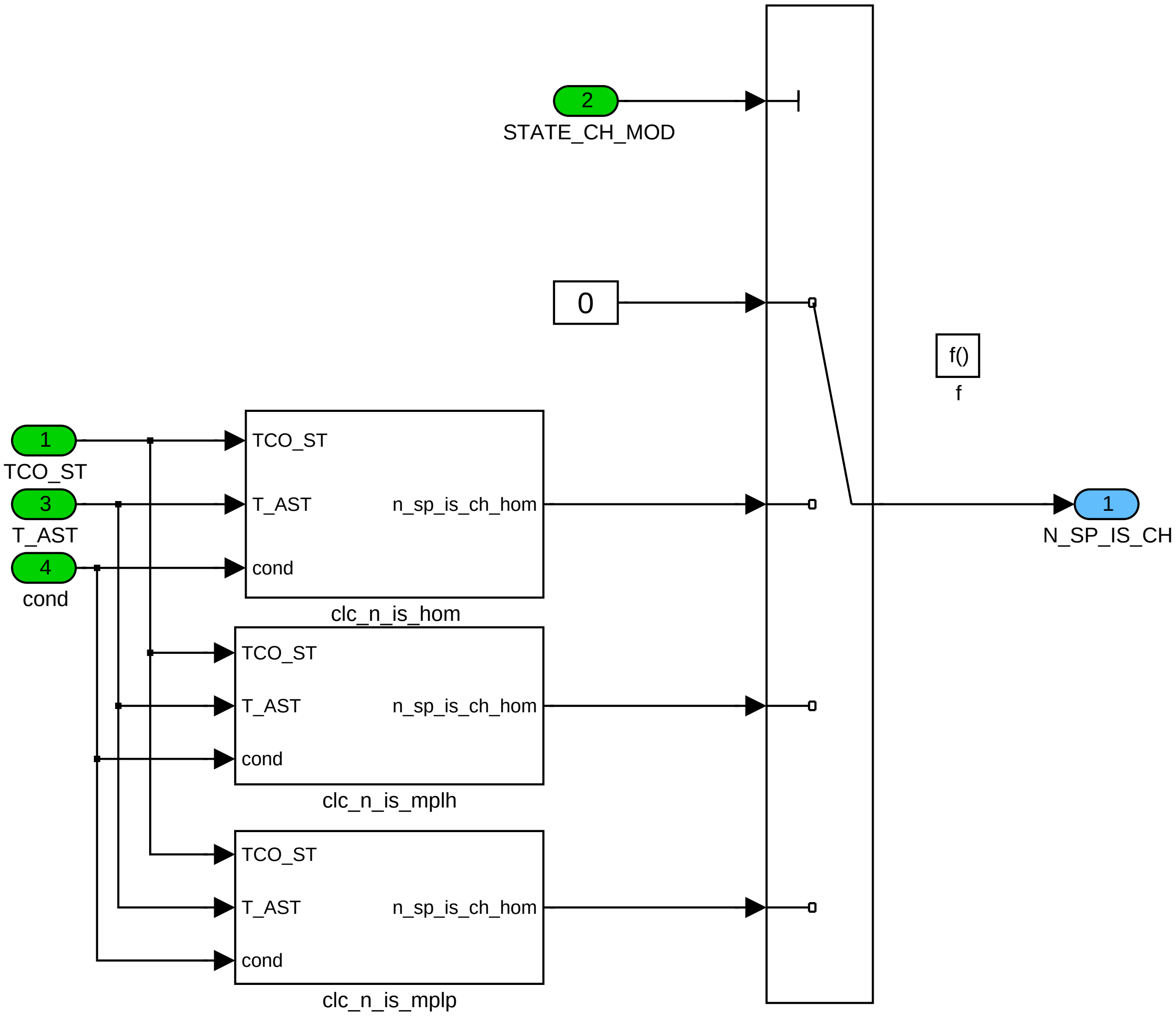


Figure 107:
Path: EXTC_REQGNIS0/clc_of_N_SP_IS_CH_AST/operate/N_SP_DRI

30.18.1.1.1.1 Calculation of N_IS_HOM

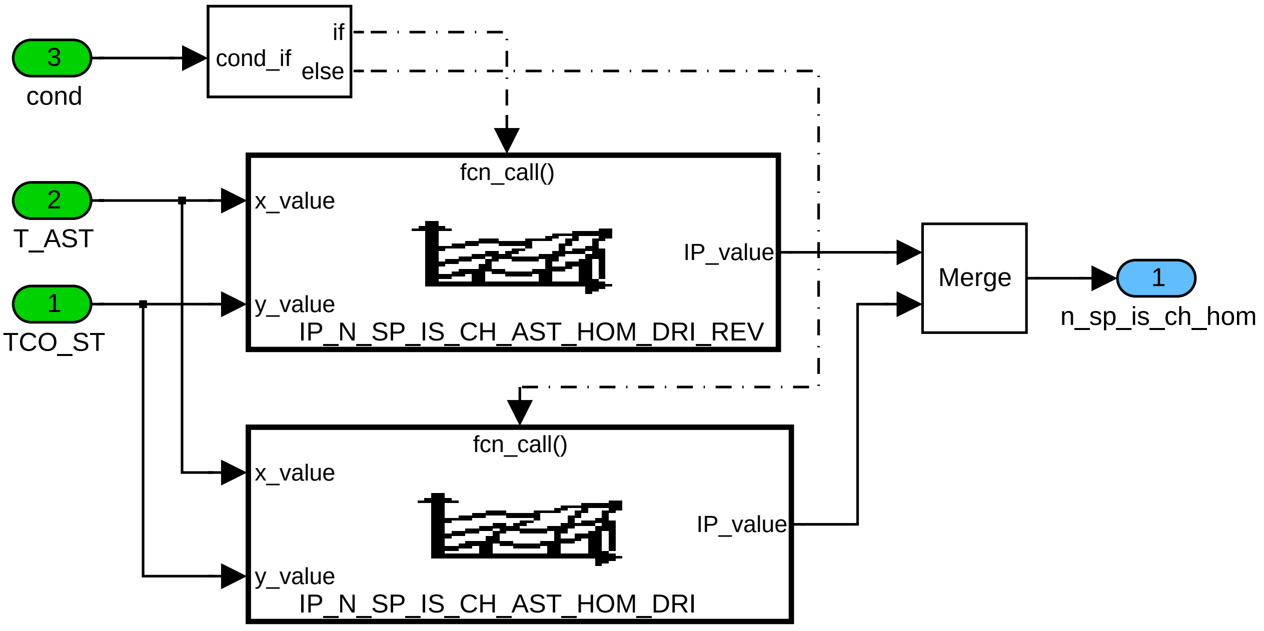


Figure 108:
Path: EXTC_REQGNIS0/clc_of_N_SP_IS_CH_AST/operate/N_SP_DRI/clc_n_is_hom

30.18.1.1.1.2 Calculation of N_IS_MPLH

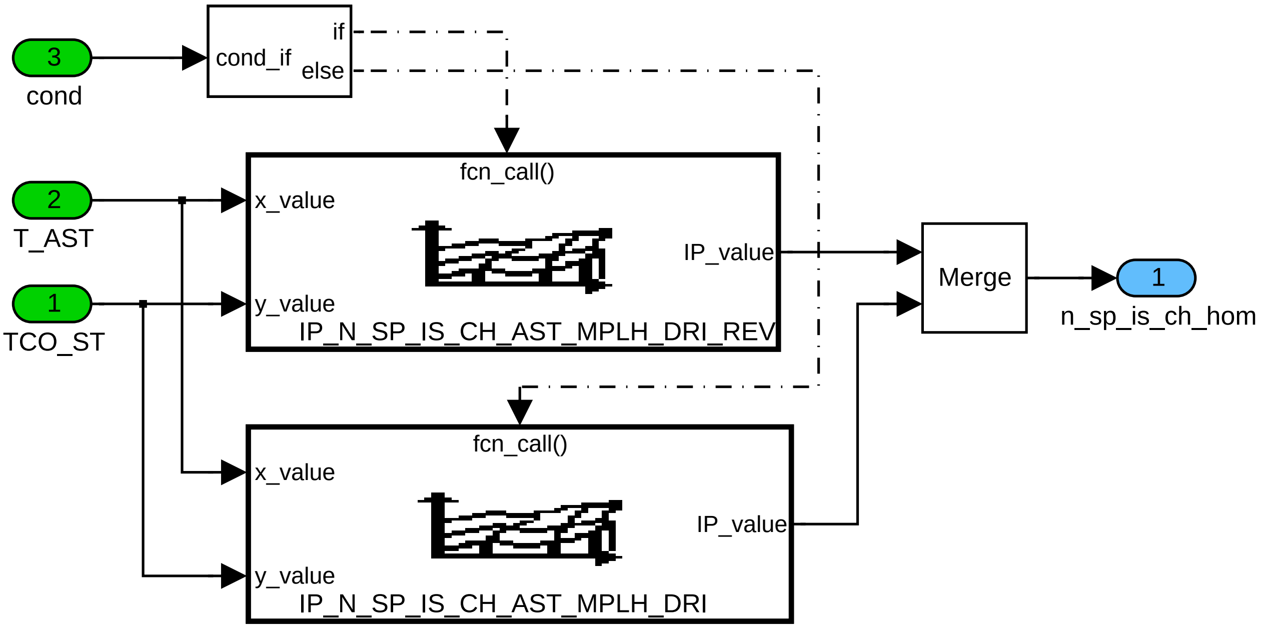


Figure 109:
Path: EXTC_REQGNIS0/clc_of_N_SP_IS_CH_AST/operate/N_SP_DRI/clc_n_is_mplh

30.18.1.1.1.3 Calculation of N_IS_MPLP

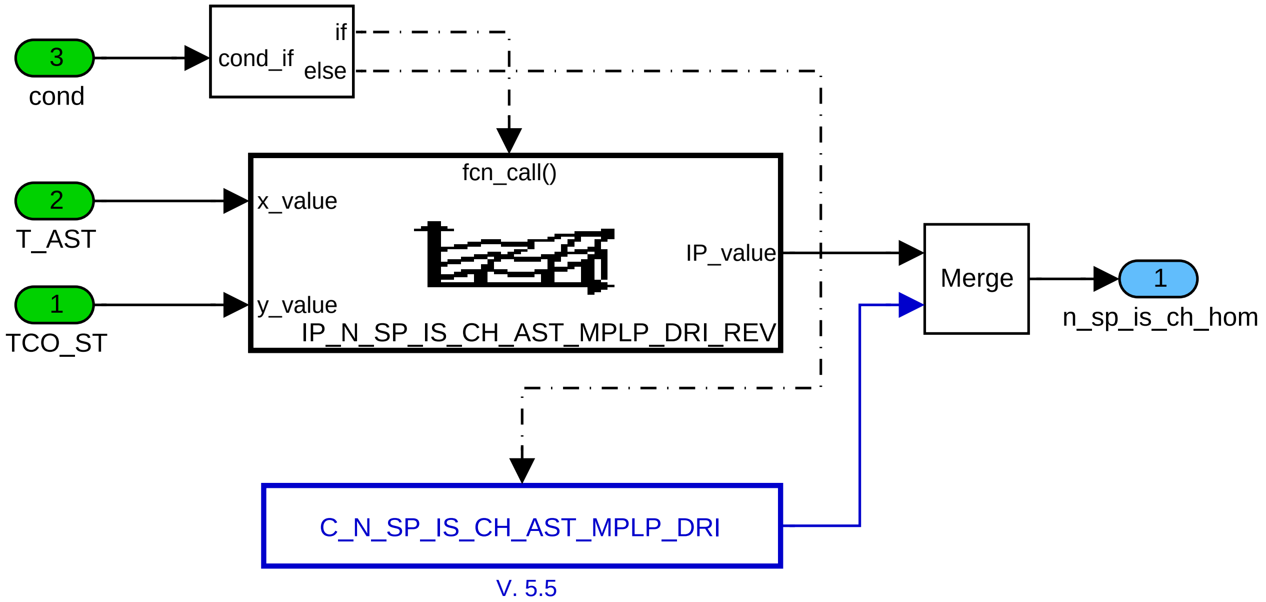


Figure 110:
Path: EXTC_REQGNIS0/clc_of_N_SP_IS_CH_AST/operate/N_SP_DRI/clc_n_is_mplp

30.18.1.1.2 Calculation of N_SP_IS_CH_AST

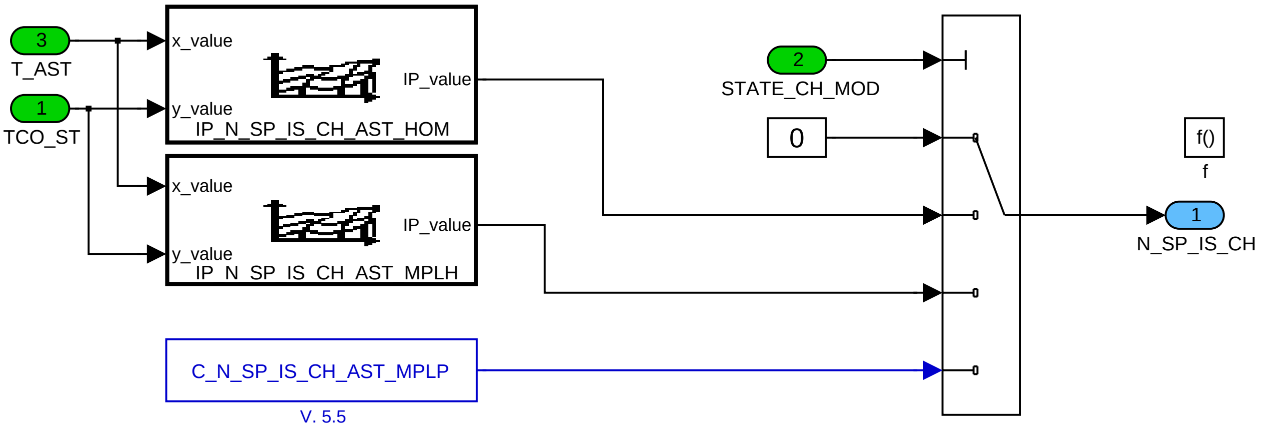


Figure 111:
Path: EXTC_REQGNIS0/clc_of_N_SP_IS_CH_AST/operate/N_SP_NOT_DRI

30.18.2 Low load catalyst heating

During low load catalyst heating there is no multiple injection in homogeneous mode.

Application Conditions

Function description

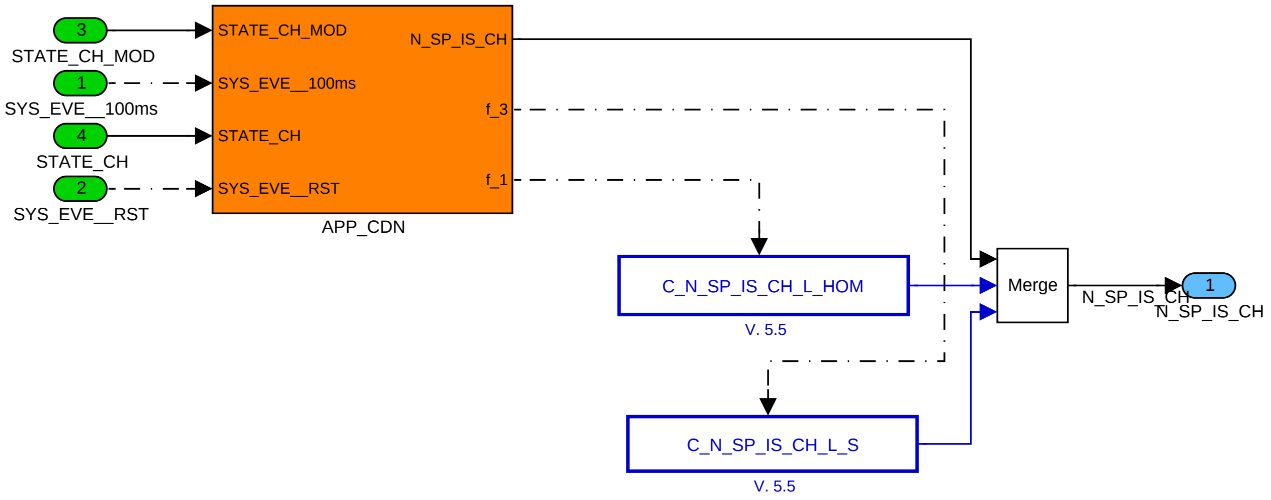


Figure 112:
Path: EXTC_REQGNIS0/clc_of_N_SP_IS_CH_L

30.19 Camshaft setpoint for catalyst heating

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CAM_SP_CH_EX	O/V	8000... 7FFFH	-768... 767.9765625	0.0234375	[°CRK]
Camshaft position setpoint during cat heating; exhaust					
CAM_SP_CH_IN	O/V	8000... 7FFFH	-768... 767.9765625	0.0234375	[°CRK]
Camshaft position setpoint during cat heating; inlet					
FAC_EFF_IGA_CH_SP	O/V	0... FFH	0... 0.99609375	3.90625e-3	[-]
Setpoint of factor for fading ignition angle efficiency in case of catalyst heating					
STATE_OPM_IVVT_CH	O/V	0... 2H	0... 2	1	[-]
IVVT - strategy request; 0 = no request ; 1 = engine roughness-optimal ; 2 = emission-optimal					

Input Data:

STATE_CH_MOD	N	FAC_MAF_REL_EGR_COR	FAC_EFF_IGA_CH
TCO	T_AST	STATE_CH_MOD_REQ	LV_SO2P_LAMB_PULS
STATE_CH	LV_IS	FAC_CAM_CH	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_STATE_OPM_IVVT_CH_L	1	0... 2H	0... 2	1	[-]
IVVT - strategy request for low load catalyst heating					
ID_STATE_OPM_IVVT_CH_HOM_SO2P	6*6	0... 2H	0... 2	1	[-]
LDP_N_ID_OPM_IVVT_CH_HOM_SO2P	6	0... 1FE0H	0... 8160	1	[rpm]
LDP_FAC_MAF_EGR_ID_CH_HOM_SO2P	6	0... BB8H	0... 300	0.1	[%]
IVVT - strategy request for catalyst heating for desulfation					
ID_STATE_OPM_IVVT_CH_MPLH_SO2P	6*6	1... 2H	1... 2	1	[-]
LDP_N_ID_OPM_IVVT_CH_MPLH_SO2P	6	0... 1FE0H	0... 8160	1	[rpm]
LDP_FAC_MAF_EGR_ID_CH_MPLH_SO2P	6	0... BB8H	0... 300	0.1	[%]
IVVT - strategy request for catalyst heating for desulfation, multiple injection					
IP_CAM_CH_COLD_EX	10*10	0... FFH	-40... -135.625	-0.375	[°CRK]
LDPM_N_2_EXTC	10	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_2_EXTC	10	0... BB8H	0... 300	0.1	[%]
Outlet camshaft setpoint for catalyst heating					
IP_CAM_CH_COLD_EX_IS	10*8	0... FFH	-40... -135.625	-0.375	[°CRK]
LDPM_N_4_EXTC	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_4_EXTC	10	0... BB8H	0... 300	0.1	[%]
Outlet camshaft setpoint for catalyst heating in idle					
IP_CAM_CH_COLD_IN	10*10	0... FFH	60... 155.625	0.375	[°CRK]
LDPM_N_2_EXTC	10	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_2_EXTC	10	0... BB8H	0... 300	0.1	[%]
Inlet camshaft setpoint for catalyst heating					
IP_CAM_CH_COLD_IN_IS	10*8	0... FFH	60... 155.625	0.375	[°CRK]
LDPM_N_4_EXTC	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_4_EXTC	10	0... BB8H	0... 300	0.1	[%]
Inlet camshaft setpoint for catalyst heating in idle					
IP_CAM_CH_HOM_HOT_EX	10*16	0... FFH	-40... -135.625	-0.375	[°CRK]
LDPM_N_1_EXTC	16	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_1_EXTC	10	0... BB8H	0... 300	0.1	[%]
Outlet camshaft setpoint for catalyst heating with single injection					

IP_CAM_CH_HOM_HOT_EX_IS	10*8	0... FFH	-40... -135.625	-0.375	[°CRK]
LDPM_N_3_EXTC	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_3_EXTC	10	0... BB8H	0... 300	0.1	[%]
Outlet camshaft setpoint for catalyst heating with single injection in idle					
IP_CAM_CH_HOM_HOT_IN	10*16	0... FFH	60... 155.625	0.375	[°CRK]
LDPM_N_1_EXTC	16	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_1_EXTC	10	0... BB8H	0... 300	0.1	[%]
Inlet camshaft setpoint for catalyst heating with single injection					
IP_CAM_CH_HOM_HOT_IN_IS	10*8	0... FFH	60... 155.625	0.375	[°CRK]
LDPM_N_3_EXTC	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_3_EXTC	10	0... BB8H	0... 300	0.1	[%]
Inlet camshaft setpoint for catalyst heating with single injection in idle					
IP_CAM_CH_MPLH_HOT_EX	10*16	0... FFH	-40... -135.625	-0.375	[°CRK]
LDPM_N_1_EXTC	16	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_1_EXTC	10	0... BB8H	0... 300	0.1	[%]
Outlet camshaft setpoint for catalyst heating with multiple injection					
IP_CAM_CH_MPLH_HOT_EX_IS	10*8	0... FFH	-40... -135.625	-0.375	[°CRK]
LDPM_N_3_EXTC	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_3_EXTC	10	0... BB8H	0... 300	0.1	[%]
Outlet camshaft setpoint for catalyst heating with multiple injection in idle					
IP_CAM_CH_MPLH_HOT_IN	10*16	0... FFH	60... 155.625	0.375	[°CRK]
LDPM_N_1_EXTC	16	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_1_EXTC	10	0... BB8H	0... 300	0.1	[%]
Inlet camshaft setpoint for catalyst heating with multiple injection					
IP_CAM_CH_MPLH_HOT_IN_IS	10*8	0... FFH	60... 155.625	0.375	[°CRK]
LDPM_N_3_EXTC	8	0... 1FE0H	0... 8160	1	[rpm]
LDPM_FAC_MAF_REL_EGR_COR_3_EXTC	10	0... BB8H	0... 300	0.1	[%]
Inlet camshaft setpoint for catalyst heating with multiple injection in idle					
IP_FAC_EFF_IGA_CH_SP	8*8	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_T_AST_6_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_6_EXTC	8	0... FEH	-48... 142.5	0.75	[°C]
Setpoint of factor for fading ignition angle efficiency in case of catalyst heating					
IP_FAC_EFF_IGA_CH_SP_MPLH	8*8	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_T_AST_6_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_6_EXTC	8	0... FEH	-48... 142.5	0.75	[°C]
Setpoint of factor for fading ignition angle efficiency in case of catalyst heating with multiple injection					

General Information

In this module setpoints for the camshafts during catalyst heating are generated.

Application Conditions

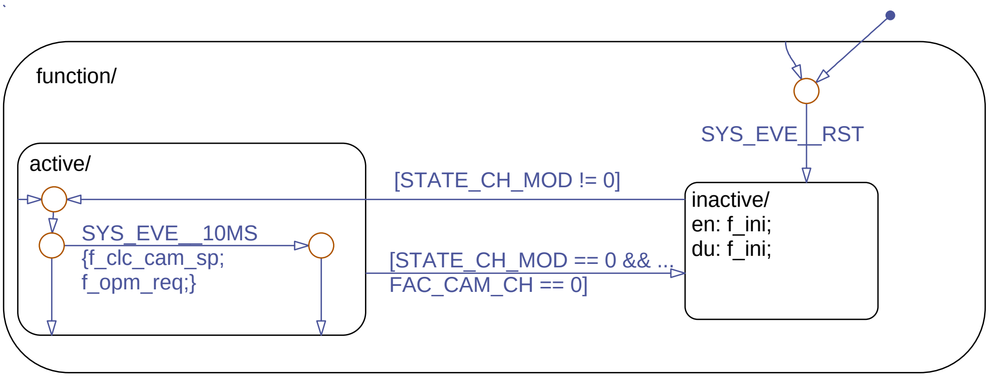


Figure 113:
Path: EXTC_REQGNcam0/APP_CDN/Chart

Function description

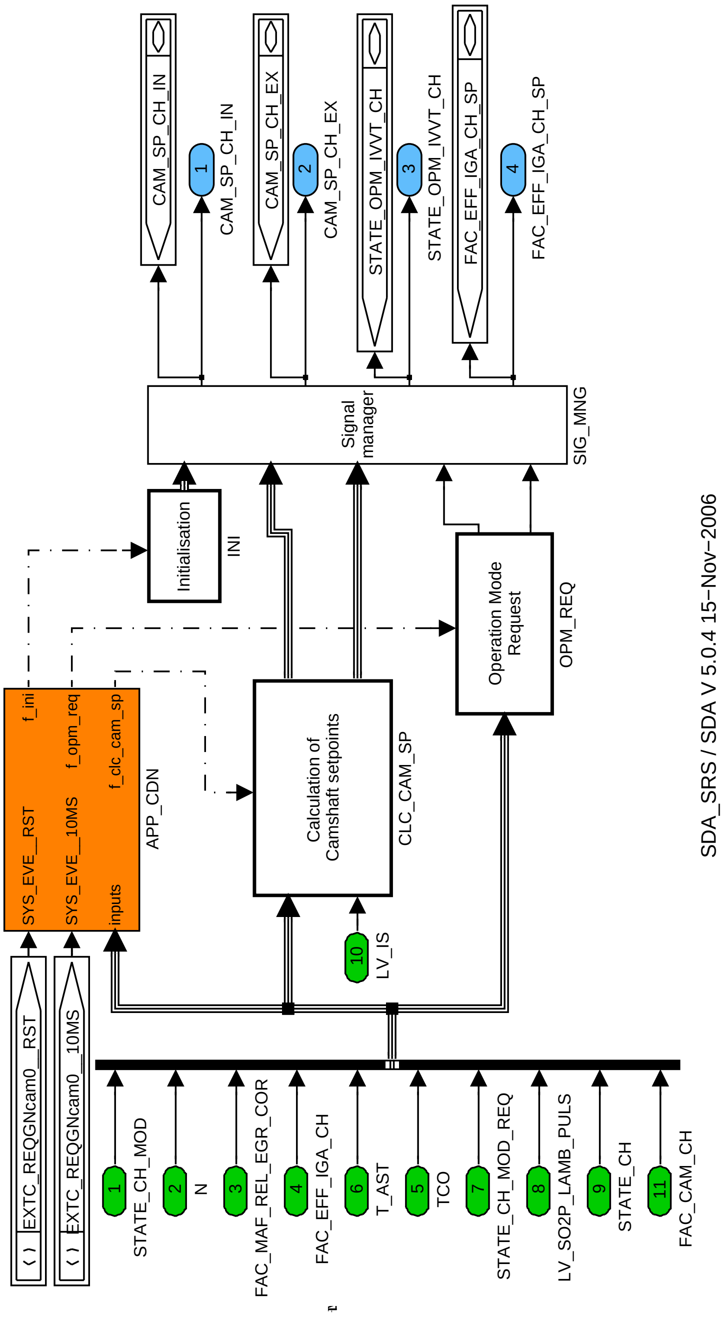


Figure 114:
Path: EXTC_REQGNcam0

30.19.1 Initialisation

30.19.1.1 Initialisation subsystem

The camshaft positions are initialised at reset and deactivated function with extreme-position-values

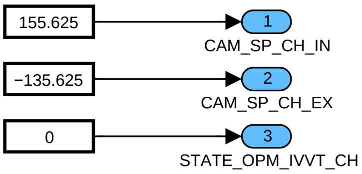


Figure 115:
Path: EXTC_REQGNcam0/INI/INI_SUB

30.19.2 Calculation of camshaft setpoints

All maps for camshaft setpoints are twice available for idle and for part load.

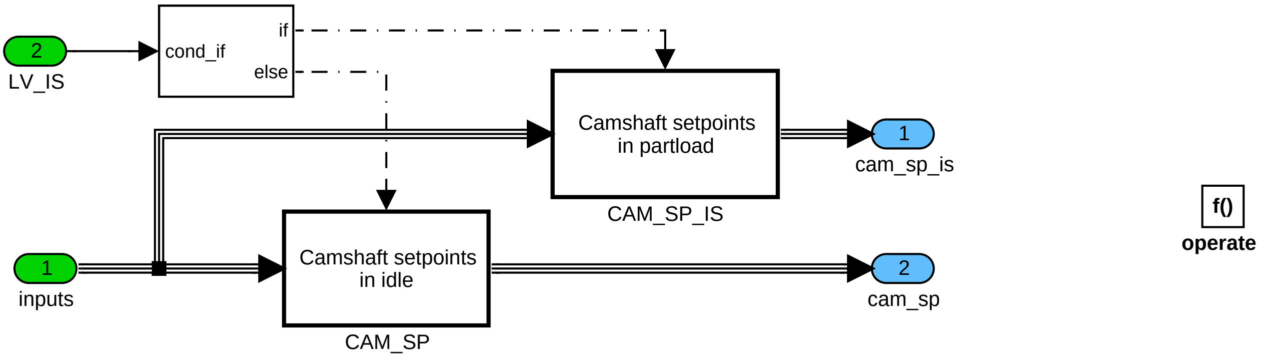


Figure 116:
Path: EXTC_REQGNcam0/CLC_CAM_SP

30.19.2.1 EXTC_REQGNCAM0/CLC_CAM_SP/CAM_SP_IS

30.19.2.1.1 Camshaft setpoints in idle

There are 4 maps for camshaft-setpoints for catalyst heating which shell be used for emission-optimal calibration (2 for single and 2 for multiple injection). FAC_EFF_IGA_CH interpolates between the map-outputs and the outputs of 2 COLD -maps which are engine-roughness-optimal calibrated.

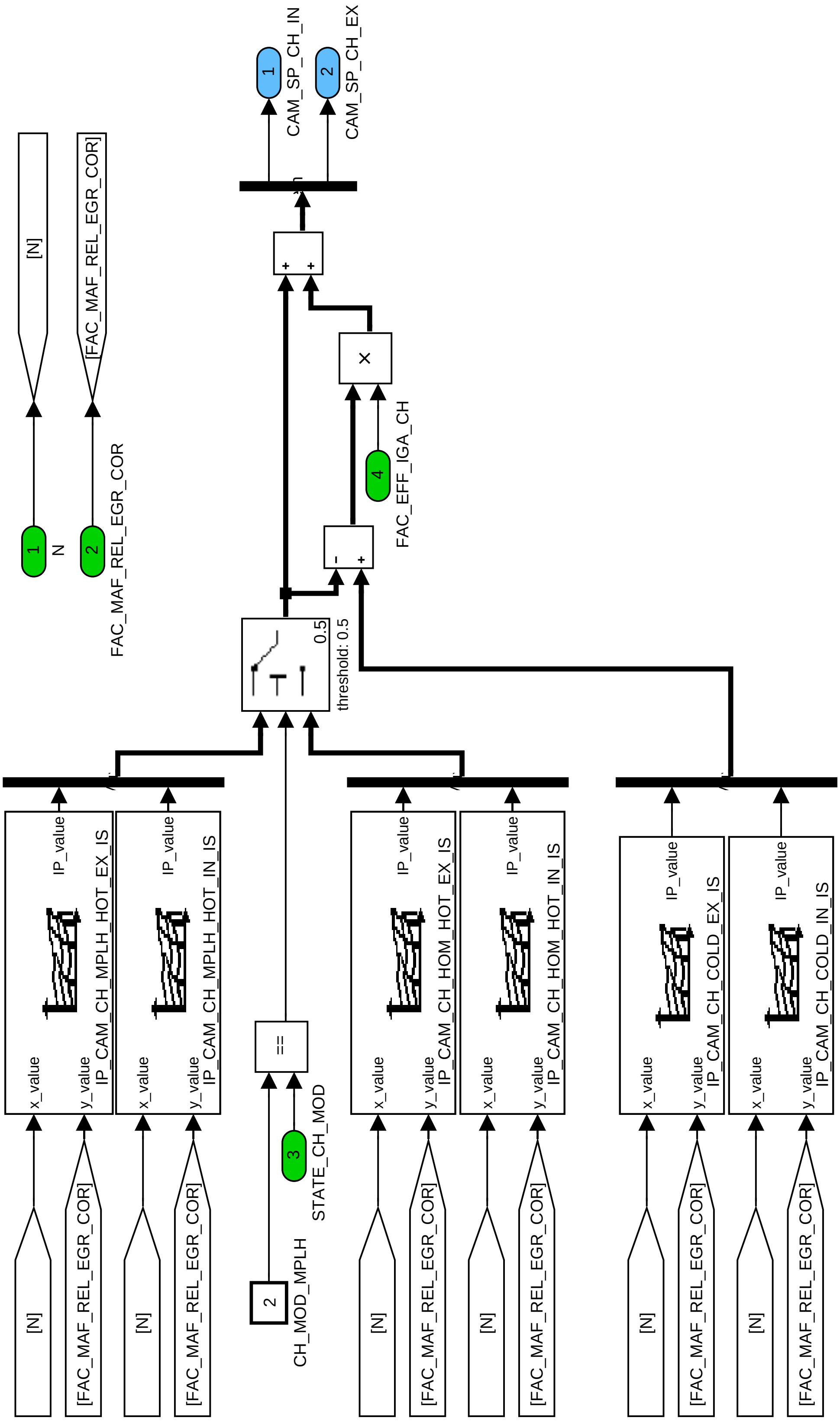


Figure 117:
Path: EXTC_REQGNcam0/CLC_CAM_SP/CAM_SP_IS/SUB

30.19.2.2 EXTC_REQGNCAM0/CLC_CAM_SP/CAM_SP

30.19.2.2.1 Camshaft setpoints in partload

The function is dublicated from idle-speed-function.

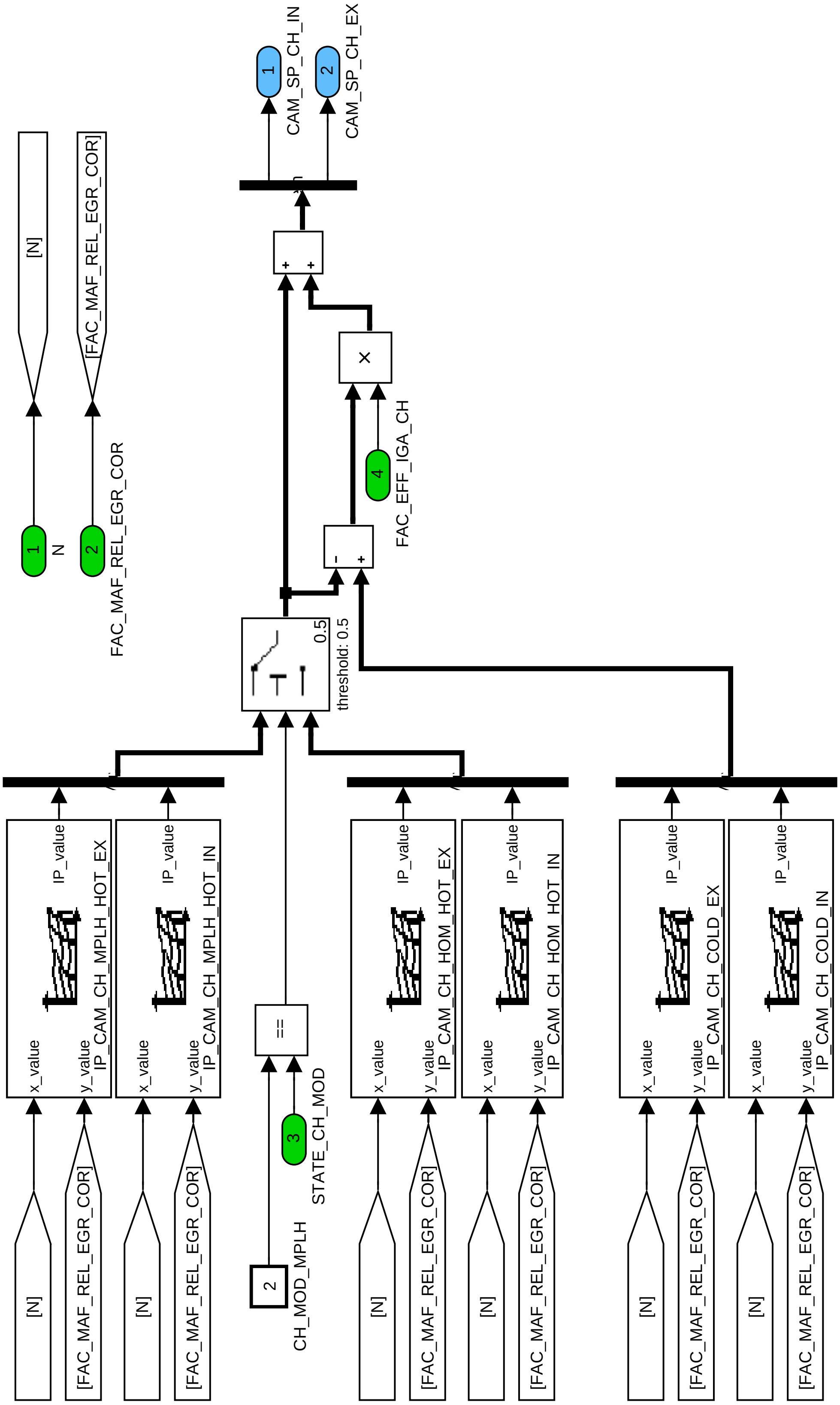


Figure 118:

Path: EXTC_REQGNcam0/CLC_CAM_SP/CAM_SP/SUB

30.19.3 Operation mode request

30.19.3.1 Calculation

Depending on STATE_CH the calculation is separated for after start, low load and desulfation catalyst heating.

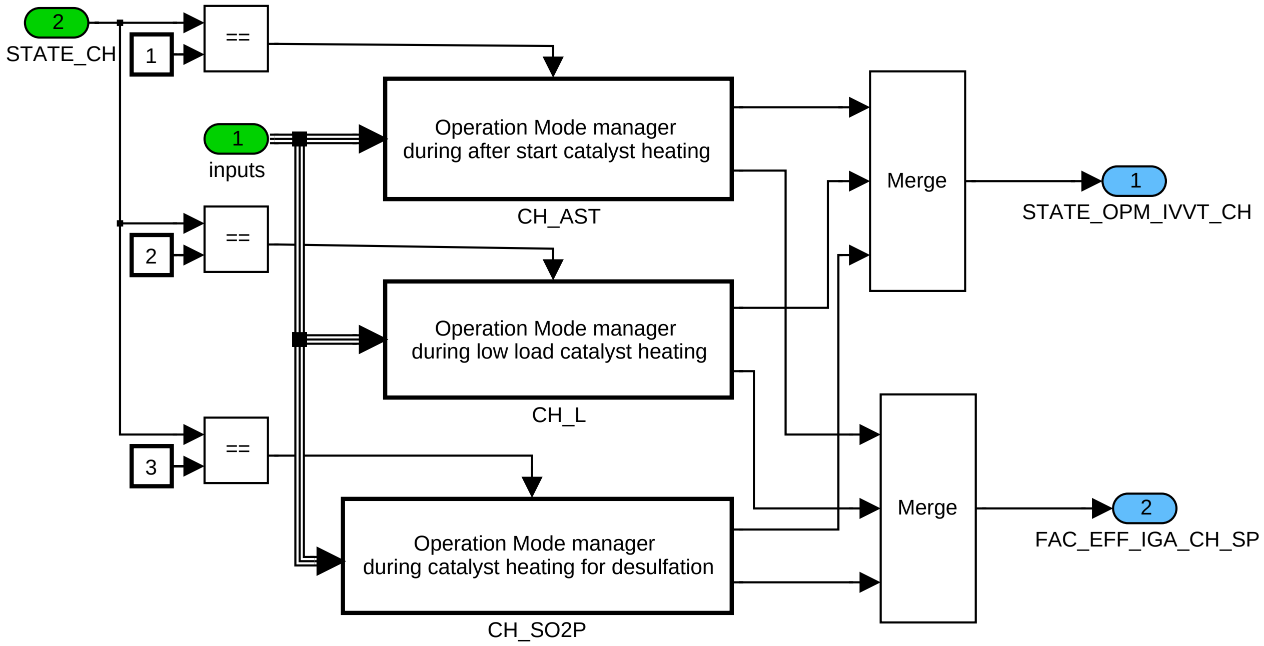


Figure 119:
Path: EXTC_REQGNcam0/OPM_REQ/SUB

30.19.3.1.1 EXTC_REQGNCAM0/OPM_REQ/SUB/CH_AST

30.19.3.1.1.1 Operation Mode manager during after start catalyst heating

FAC_EFF_IGA_CH_SP = 0 means that emission-optimal settings are to be chosen. FAC_EFF_IGA_CH_SP = 1 means that engine-roughness-optimal settings of camshafts shell be chosen. This setpoint value shell result in FAC_EFF_IGA_CH which is used for interpolation betweet emission- and engine-roughness-optimal maps. STATE_OPM_IVVT_CH contains the same information in logical way. 2 means emission-optimum, 1 means engine-roughness-optimum, 0 means no request from catalyst heating side.

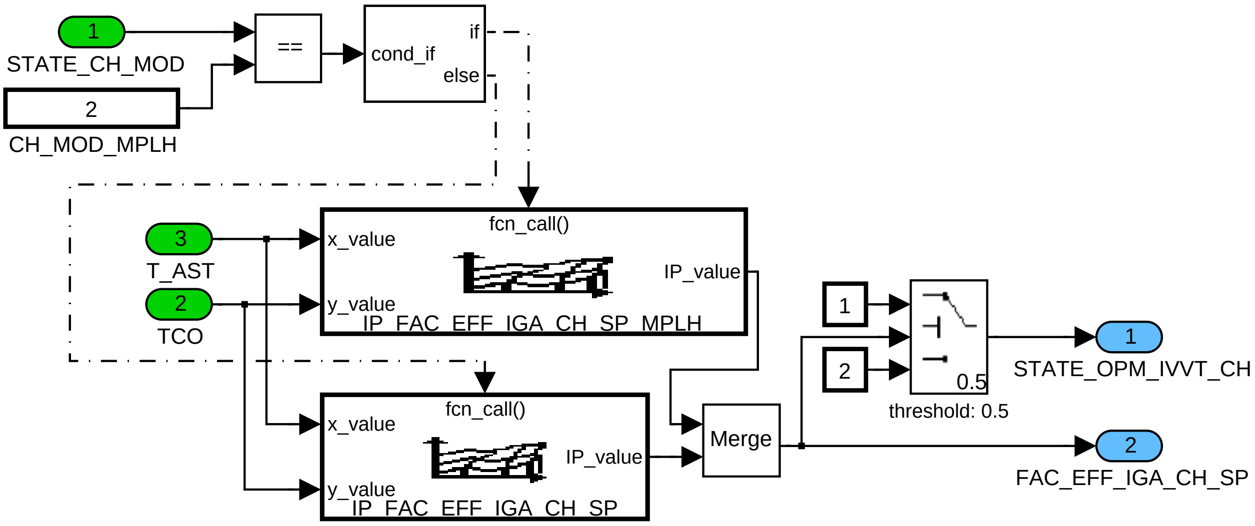


Figure 120:
Path: EXTC_REQGNcam0/OPM_REQ/SUB/CH_AST/SUB

30.19.3.1.2 EXTC_REQGNCAM0/OPM_REQ/SUB/CH_L

30.19.3.1.2.1 Operation Mode manager during low load catalyst heating

Only for homogeneous mode with single injection the setting of the operation mode request is calibratable. In all other cases '0' that means no alternation in camshaft setpoints is generated. FAC_EFF_IGA_CH_SP is set depending on the STATE so no values between 0 and 1 are possible.

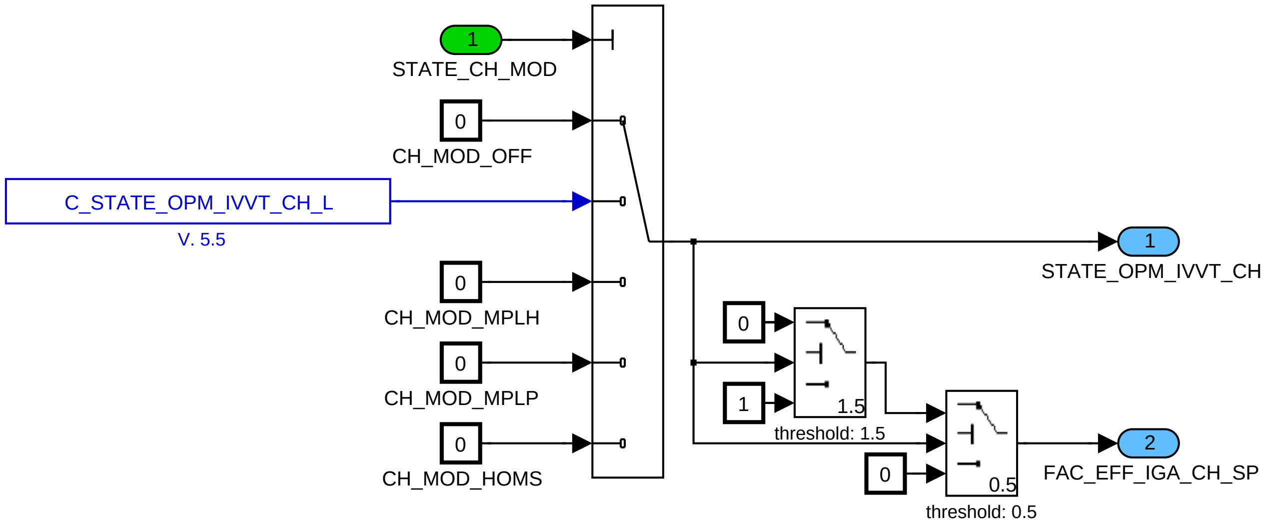


Figure 121:
Path: EXTC_REQGNcam0/OPM_REQ/SUB/CH_L/SUB

30.19.3.1.3 EXTC_REQGNCAM0/OPM_REQ/SUB/CH_SO2P

30.19.3.1.3.1 Operation Mode manager during catalyst heating for desulfation

For homogenous mode with single and with multiple injection the setting of the operation mode request is calibratable. In all other cases '0' that means no alternation in camshaft setpoints is generated. FAC_EFF_IGA_CH_SP is set depending on the STATE so no values between 0 and 1 are possible.

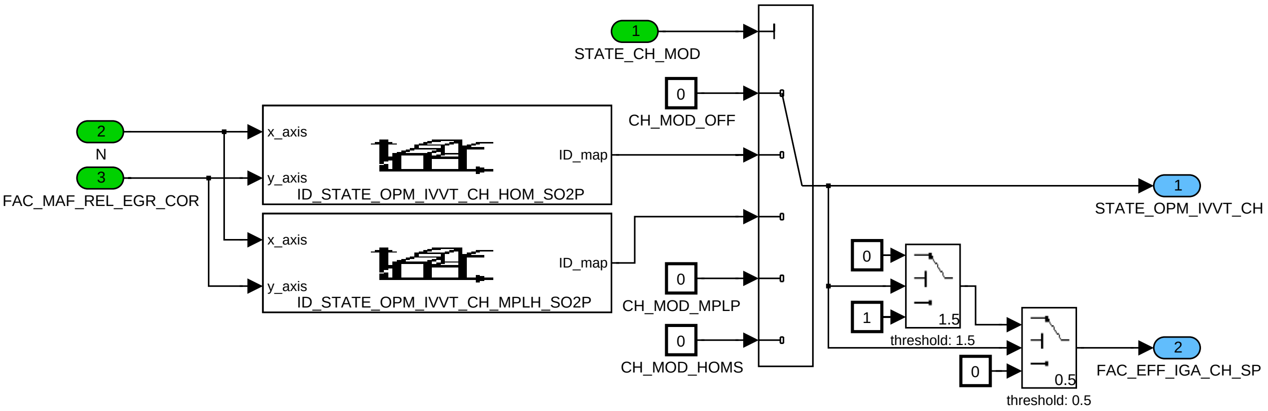


Figure 122:
Path: EXTC_REQGNcam0/OPM_REQ/SUB/CH_SO2P/SUB

30.20 Fuel pressure for catalyst heating

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
FUP_SP_CH	O/V	0... FFFFH	0... 347776	5.306722	[hPa]
Fuel pressure setpoint for catalyst heating					

Input Data:

LV_MPLH_ACT	N_32	STATE_CH_MOD	STATE_CH_MOD_REQ
TCO_ST	TQI_REQ_TRA	T_AST	

Calibration Data:

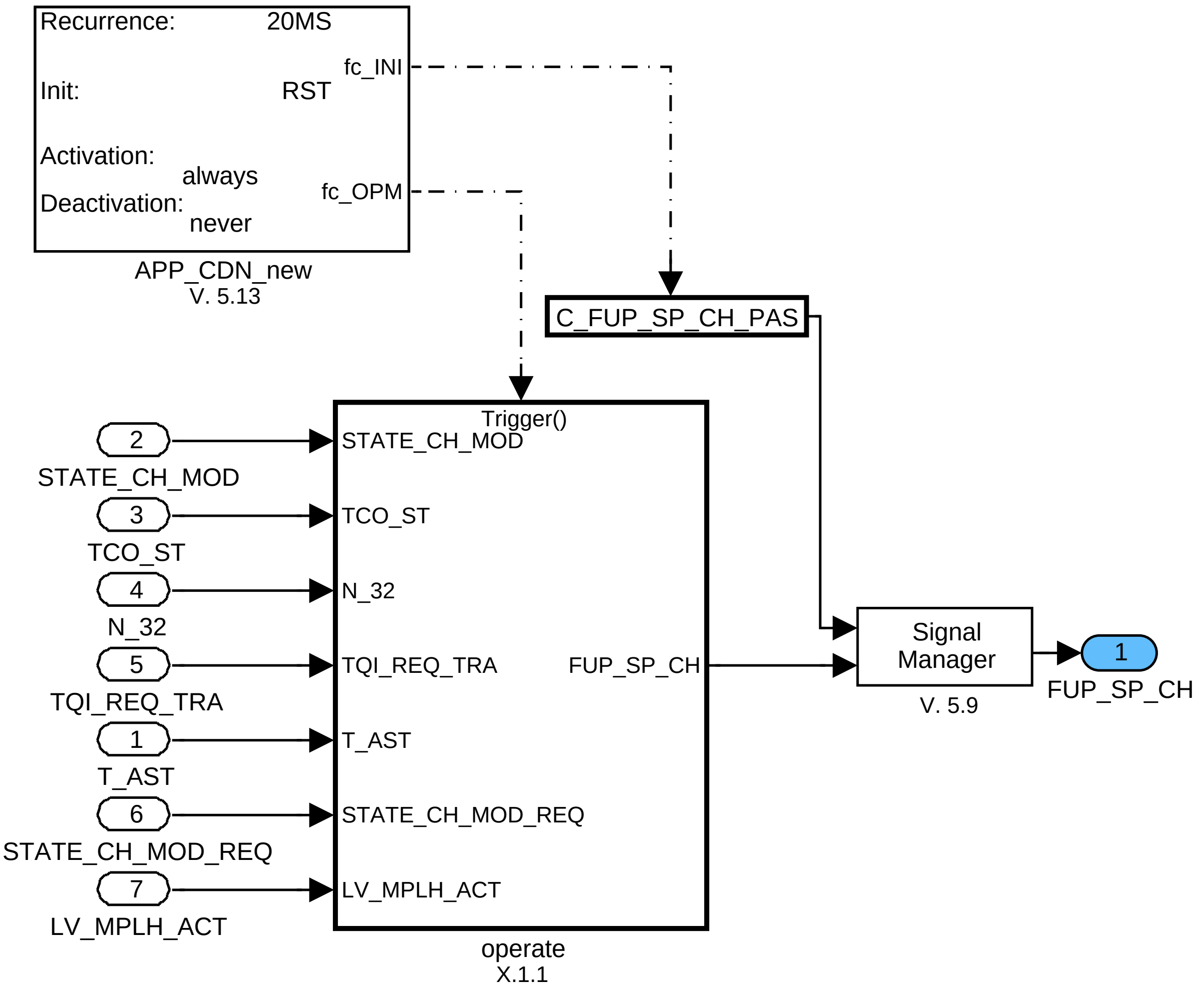
Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_FUP_SP_CH_PAS	1	0... FFFFH	0... 347776	5.306722	[hPa]
Passive value for fuel pressure setpoint for catalyst heating					
IP_FAC_FUP_SP_CH	4x6	0... FFH	0... 0.9960931	3.9063e-3	[-]
LDP_TCO_ST_IP_FAC_FUP_SP_CH	4	0... FEH	-48... 142.5	0.75	[C]
LDP_T_AST_IP_FAC_FUP_SP_CH	6	0... FFFFH	0... 6.554e+3	0.1	[s]
Factor for fuel pressure setpoint during (MPLH) - catalyst heating					
IP_FUP_SP_CH	6x6	0... FFFFH	0... 347776	5.306722	[hPa]
LDP_N_32_IP_FUP_SP_CH	6	0... FFH	0... 8.16e+3	32	[rpm]
LDP_TQI_REQ_TRA_IP_FUP_SP_CH	6	0... 7FFFH	0... 1.024e+3	0.03125	[Nm]
Fuel pressure setpoint for catalyst heating					

General Information

During homogeneous mode with double injection (MPLH mode) it is necessary to use a higher fuel pressure due to the 2nd "stratified" injection.

Application Conditions

Initialization: RST
Recurrence: 20MS
Activation: always
Deactivation: never



SDA_SRS / SDA V 5.0.1 02−Dec−2005

Figure 123:
Path: EXTC_REQGNFUP0

30.20.1 Operate Subsystem

If catalyst heating in MPLH-mode is active STATE_CH_MOD OR STATE_CH_MOD_REQ = 2 = CH_MOD_MPLH the IP values are used. This can be during after start catalyst heating or catalyst heating for desulfation. In all other cases the passive value C_FUP_SP_CH_PAS is used.

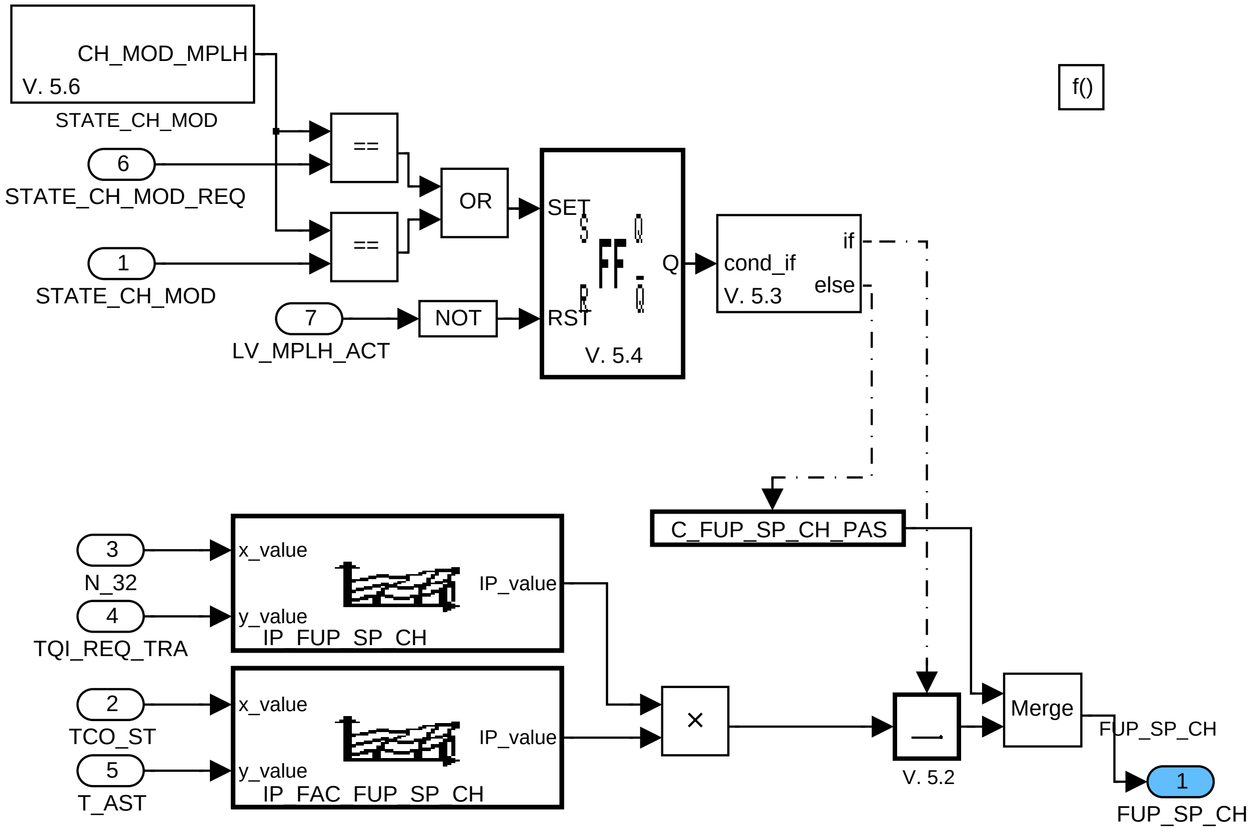


Figure 124:
Path: EXTC_REQGNFUP0/operate

30.21 MAF correction for catalyst heating in stratified mode

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
MAF_DIF_S_CH	O/V	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Mass air flow difference for cat. heating in stratified mode (possibly with post injection)					

Input Data:

STATE_CH_MOD	NC_CYL_NR	STATE_CH

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_MAF_DIF_ADD_MFF_POST_CH_L	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Mass of post injection depending correction for MAF_DIF					
C_MAF_DIF_S_CH_L_BAS	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Basic difference of mass of air flow for low load stratified catalyst heating					
C_MAF_DIF_S_CH_MAN	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Manual setting of MAF-difference for stratified catalyst heating					
LC_MAF_DIF_S_CH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of MAF-difference for stratified catalyst heating					

General Information

During catalyst heating in stratified mode (possibly with post injection) it is necessary to reduce the mass of air flow and so the cooling effect. In this module the difference of MAF is generated.

Application Conditions

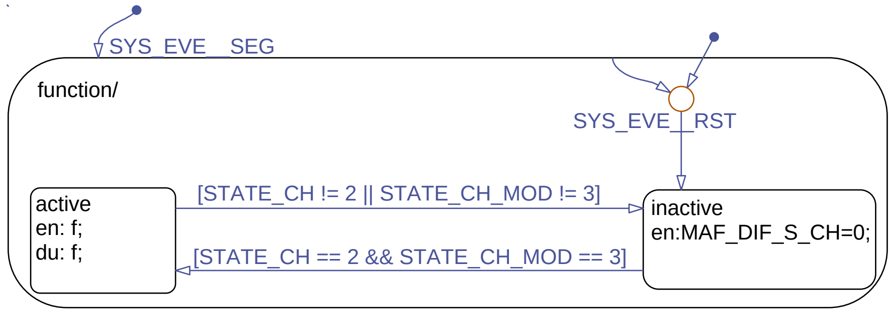
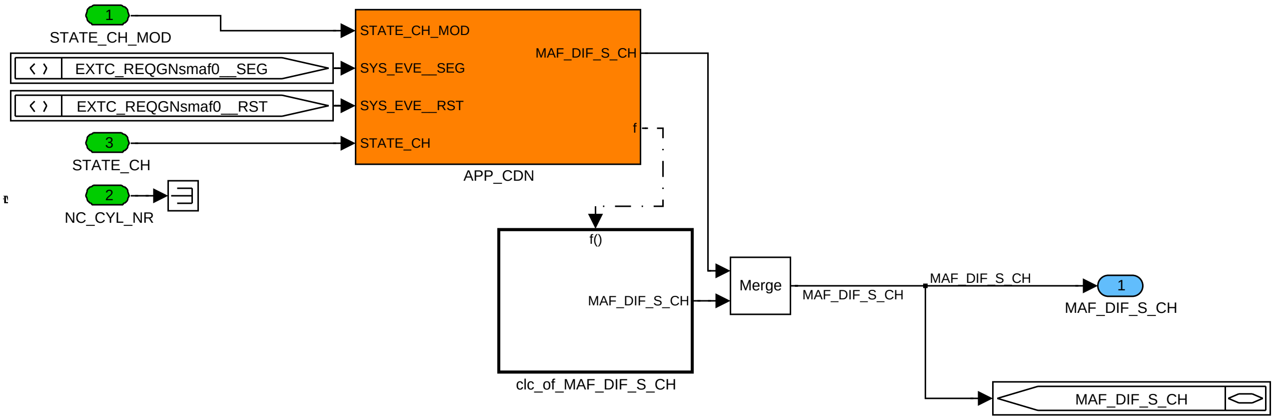


Figure 125:
Path: EXTC_REQGNsmaf0/APP_CDN/Chart

Function description



SDA_SRS / SDA V 5.0.4 07−Nov−2006

Figure 126:
Path: EXTC_REQGNsmaf0

30.21.1 Calculation of MAF_DIF_S_CH in low load catalyst heating

There is a load/engine speed depending map, an external correction that is not used up to now and a 2nd correction depending on the actual mass of post injection. The read_vector block picks out one element (here the 1st) of the cylinder individuell MFF_POSTvector.

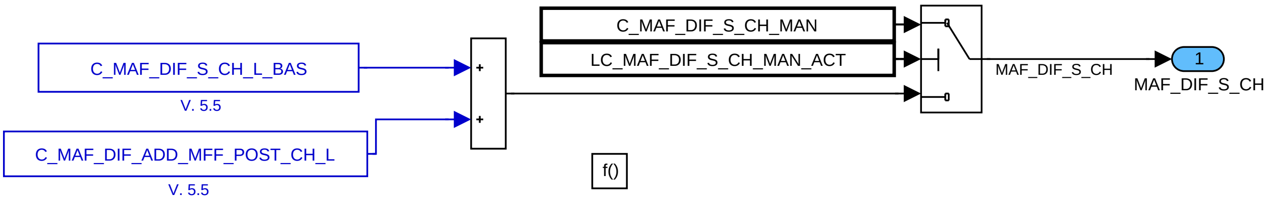
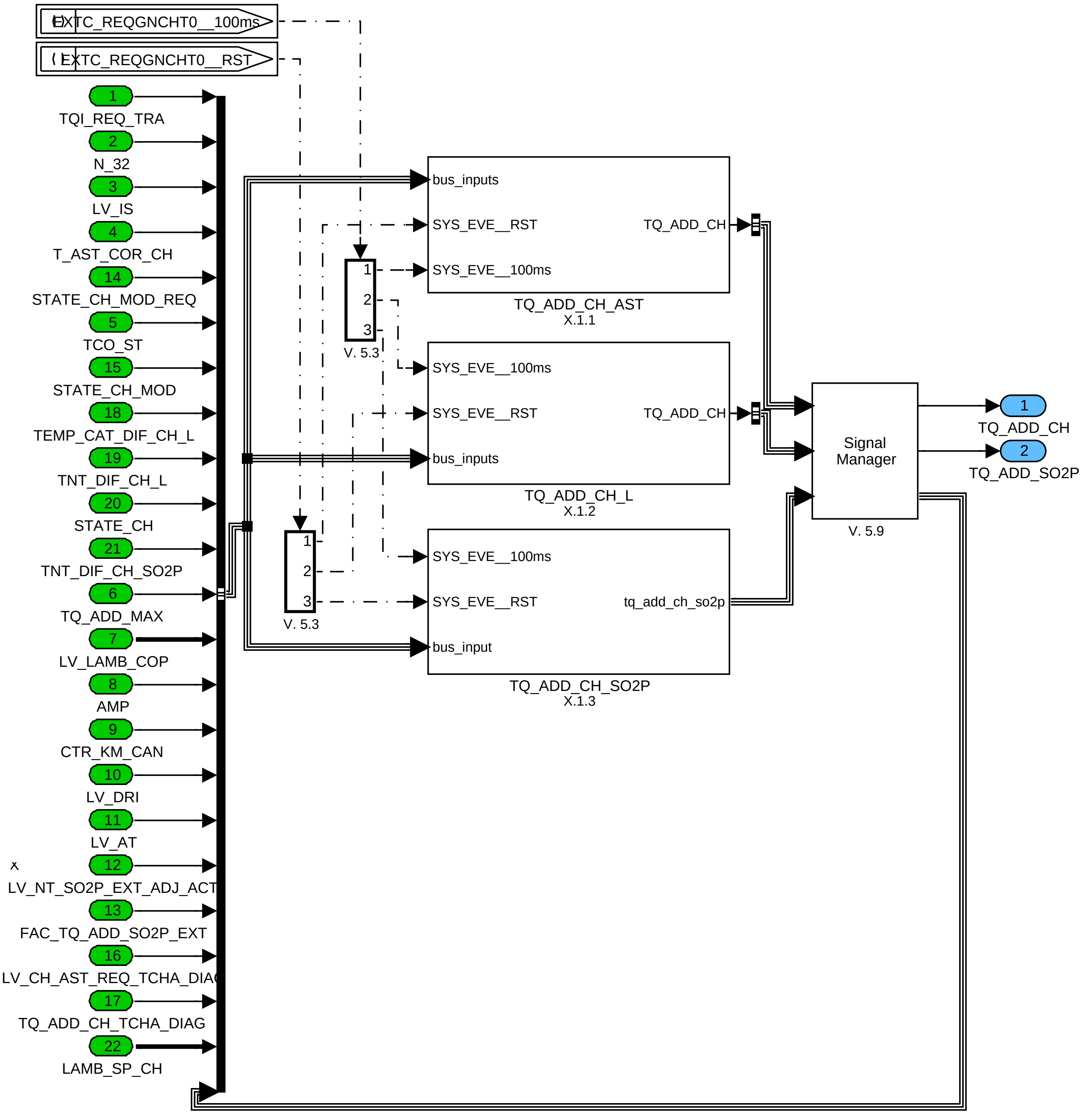


Figure 127:
Path: EXTC_REQGNsmaf0/clc_of_MAF_DIF_S_CH

30.22 Torque based catalyst heating

Overview



SDA_SRS / SDA V 5.0.4 17−Nov−2006

Figure 128:
Path: EXTC_REQGNCHT0

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
TQ_ADD_CH	O	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for catalyst heating					
TQ_ADD_SO2P	O	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve requested for desulfatisation catalyst heating					

Input Data:

TQI_REQ_TRA	N_32	LV_IS	T_AST_COR_CH
TCO_ST	TQ_ADD_MAX	LV_LAMB_COP [NC_CBK_EX_NR]	AMP
CTR_KM_CAN	LV_DRI	LV_AT	LV_NT_SO2P_EXT_ADJ_ACT
FAC_TQ_ADD_SO2P_EXT	STATE_CH_MOD_REQ	STATE_CH_MOD	LV_CH_AST_REQ_TCHA_DIAG
TQ_ADD_CH_TCHA_DIAG	TEMP_CAT_DIF_CH_L	TNT_DIF_CH_L	STATE_CH
TNT_DIF_CH_SO2P	LAMB_SP_CH [NC_CBK_EX_NR]		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_FAC_TQ_ADD_CH_L	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Difference between cat. temperatures and target temp. depending correction for TQ_ADD_CH_L					
C_FAC_TQ_ADD_SO2P_PCTL_MPLH	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Factor to weigh the precontrolled torque reserve in case of homogen split - mode					
C_TQ_ADD_CH_HOM_MAN	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Manual setting of torque reserve for catalyst heating in homogeneous mode					
C_TQ_ADD_CH_L_HOM	1	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Basic torque reserve for homogenous low load catalyst heating					
C_TQ_ADD_CH_L_LGRD	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limitation gradient for torque reserve in low load cat. heating					
C_TQ_ADD_CH_LGRD_OFF	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Gradient for decreasing TQ_ADD_CH in after start catalyst heating					
C_TQ_ADD_CH_LGRD_ON	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Gradient for increasing TQ_ADD_CH in after start catalyst heating					
C_TQ_ADD_CH_MPLH_MAN	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Manual setting of torque reserve for catalyst heating in homogeneous mode with split injection					
C_TQ_ADD_SO2P_I	1	0... FFFFH	0... 1	15.259e-6	[Nm/K]
Correlation constant for the I-share of the controller					
C_TQ_ADD_SO2P_I_MAX	1	0... 7FFFH	0... 127.9961	3.90625e-3	[Nm]
Maximium of the I-share of the controller					
C_TQ_ADD_SO2P_MAX_EXT_ADJ	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Maximum torque reserve for desulfation catalyst heating in case of external adjustment					
C_TQ_ADD_SO2P_P	1	0... FFFFH	0... 2	30.518e-6	[Nm/K]
Correlation constant for the P-share of the controller					
C_TQ_ADD_SO2P_P_MAX	1	0... 7FFFH	0... 127.9961	3.90625e-3	[Nm]
Maximium of the P-share of the controller					
C_TQ_ADD_SO2P_PCTL_EXT_ADJ	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Precontrolled torque reserve for desulfation catalyst heating in case of external adjustment					

C_TQ_ADD_SO2P_RAMP_DEC	1	0... FFFFH	0... 255.99609	3.90625e-3	[Nm/100 ms]
Ramp gradient for gradual decrease of the torque reserve					
C_TQ_ADD_SO2P_RAMP_INC	1	0... FFFFH	0... 255.99609	3.90625e-3	[Nm/100 ms]
Ramp gradient for gradual increase of the torque reserve					
IP_FAC_TQ_ADD_CH_AMP	6*8	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDP_TCO_ST_IP_FAC_TQ_ADD_CH_AMP	8	0... FEH	-48... 142.5	0.75	[°C]
LDP_AMP_IP_FAC_TQ_ADD_CH_AMP	6	0... FFFFH	0... 5434	0.0829175	[hPa]
Reduction of torque reserve due to ambient pressure					
IP_FAC_TQ_ADD_CH_CTR_KM	4	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_CTR_KM_CAN_1_EXTC	4	0... FFFFH	0... 655350	10	[km]
Reduction of torque reserve depending on catalyst age					
IP_FAC_TQ_ADD_CH_HOM	6*8	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_T_AST_COR_CH_1_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_4_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Coolant temperature and time after start depending correction TQ_ADD_CH					
IP_FAC_TQ_ADD_CH_HOM_AT	6*8	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_T_AST_COR_CH_1_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_4_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Coolant temperature and time after start depending correction TQ_ADD_CH, for AT					
IP_FAC_TQ_ADD_CH_HOM_DRI	6*6	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_N_32_3_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_TRA_1_EXTC	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Factor on torque reserve for homogeneous catalyst heating at engaged drivetrain (only for AT relevant)					
IP_FAC_TQ_ADD_CH_L_RGN	4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDP_LAMB_DIF_IP_TQ_ADD_CH_L	4	0... 7FFFH	-16... 15.9990234375	976.563e-6	[-]
Factor on TQ_ADD_CH_L depending on the split up of lambda setpoints for catalyst heating					
IP_FAC_TQ_ADD_CH_MPLH	6*8	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_T_AST_COR_CH_1_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_4_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Coolant temperature and time after start depending correction TQ_ADD_CH_HSP					
IP_FAC_TQ_ADD_CH_MPLH_AT	6*8	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_T_AST_COR_CH_1_EXTC	8	0... FFFFH	0... 6553.5	0.1	[s]
LDPM_TCO_ST_4_EXTC	6	0... FEH	-48... 142.5	0.75	[°C]
Coolant temperature and time after start depending correction TQ_ADD_CH, for AT					
IP_FAC_TQ_ADD_CH_MPLH_DRI	6*6	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDPM_N_32_3_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_TRA_1_EXTC	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Factor on torque increase for catalyst heating at homogeneous split injection at engaged drivetrain (only for AT relevant)					
IP_TQ_ADD_CH_HOM	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32_3_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_TRA_1_EXTC	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Basic torque reserve for homogeneous catalyst heating					
IP_TQ_ADD_CH_HOM_AT	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32_3_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_TRA_1_EXTC	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Basic torque reserve for homogeneous catalyst heating with automatic transmission					
IP_TQ_ADD_CH_HOM_IS	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32_5_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_TRA_2_EXTC	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Basic torque reserve for homogeneous catalyst heating in idle speed					

IP_TQ_ADD_CH_L_HOM_IS	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_TQ_ADD_CH_L_IS	6	0... FFH	0... 8160	32	[rpm]
LDP_TQI_REQ_IP_TQ_ADD_CH_L_IS	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Torque reserve for low load catalyst heating in idle speed					
IP_TQ_ADD_CH_MPLH	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32_3_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_TRA_1_EXTC	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Basic torque increase for catalyst heating at homogeneous split injection					
IP_TQ_ADD_CH_MPLH_AT	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32_3_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_TRA_1_EXTC	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Basic torque increase for catalyst heating at homogeneous split injection with automatic transmission					
IP_TQ_ADD_CH_MPLH_IS	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32_5_EXTC	6	0... FFH	0... 8160	32	[rpm]
LDPM_TQI_REQ_TRA_2_EXTC	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Basic torque increase for catalyst heating for homogeneous split					
IP_TQ_ADD_SO2P_MAX	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_TQ_SO2P_MAX	6	0... FFH	0... 8160	32	[rpm]
LDP_TQI_REQ_TRA_IP_TQ_SO2P_MAX	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Maximum torque reserve for desulfation catalyst heating					
IP_TQ_ADD_SO2P_PCTL	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_TQ_SO2P_PCTL	6	0... FFH	0... 8160	32	[rpm]
LDP_TQI_REQ_TRA_IP_TQ_SO2P_PCTL	6	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Precontrolled torque reserve for desulfation catalyst heating					
LC_TQ_ADD_CH_HOM_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of torque reserve for catalyst heating in homogeneous mode					
LC_TQ_ADD_CH_MPLH_MAN_ACT	1	0... 1H	0... 1	1	[-]
Activation of manual setting of torque reserve for catalyst heating in homogeneous mode, split injection					

General Information

This module concerns the calculation of a torque reserve which causes an ignition angle retardation and so exhaust gas temperature increase.

There are three functional parts: Cat. heating after (cold-) start, low load and for desulfation.

30.22.1 Catalyst heating after start

There are several setpoints for single and multiple injection. The output is ramped via a difference limiter.

Application Conditions

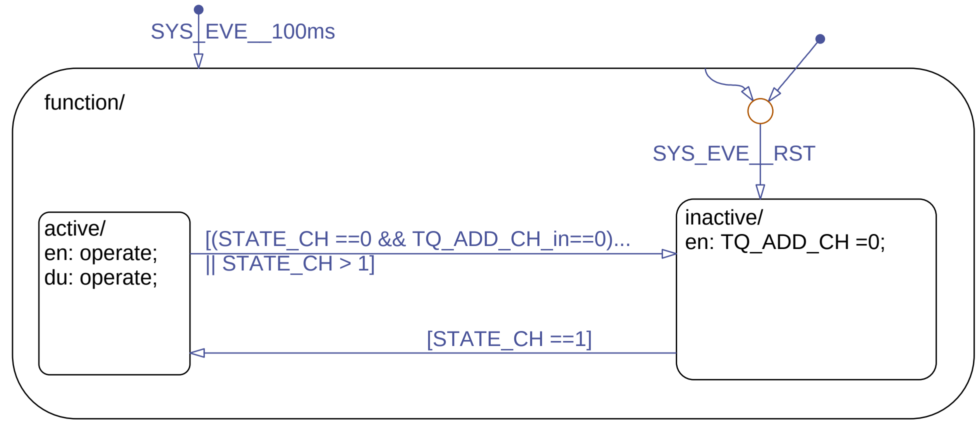


Figure 129:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_AST/APP_CDN/Chart

Function description

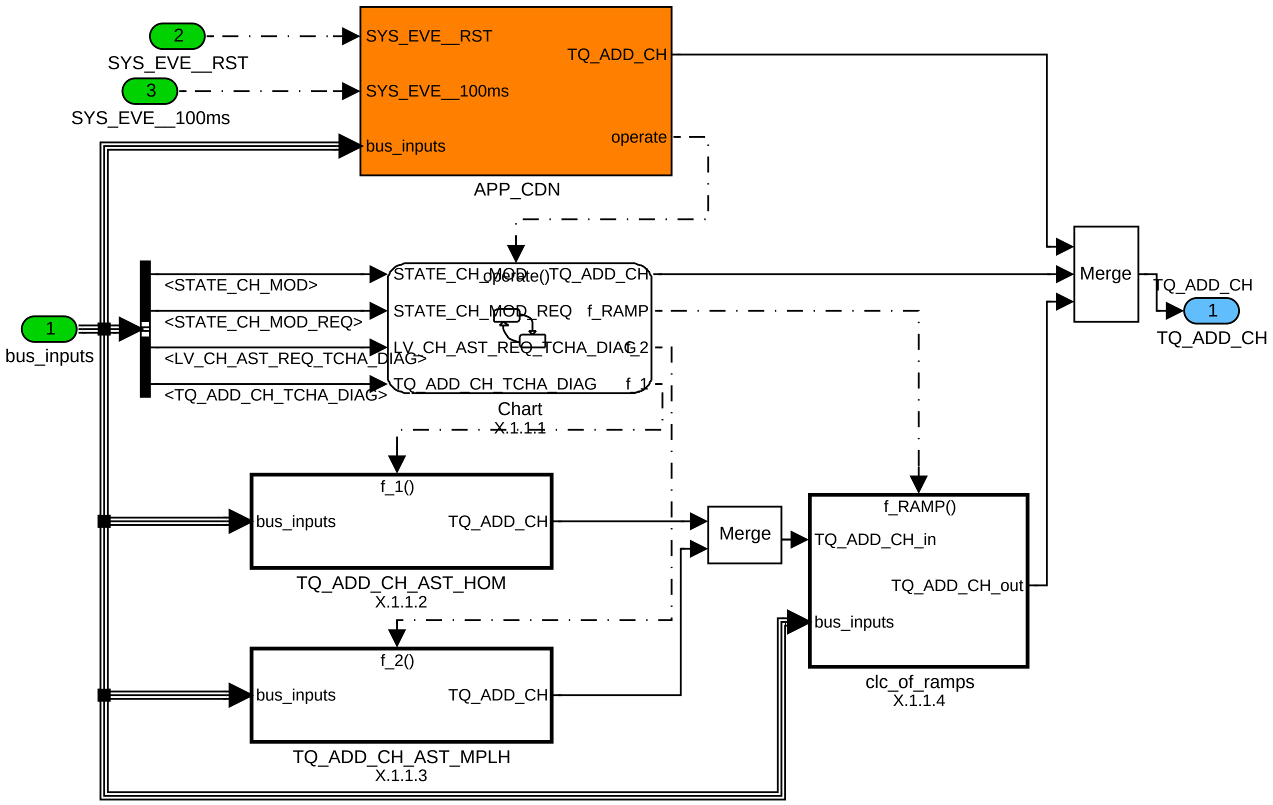


Figure 130:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_AST

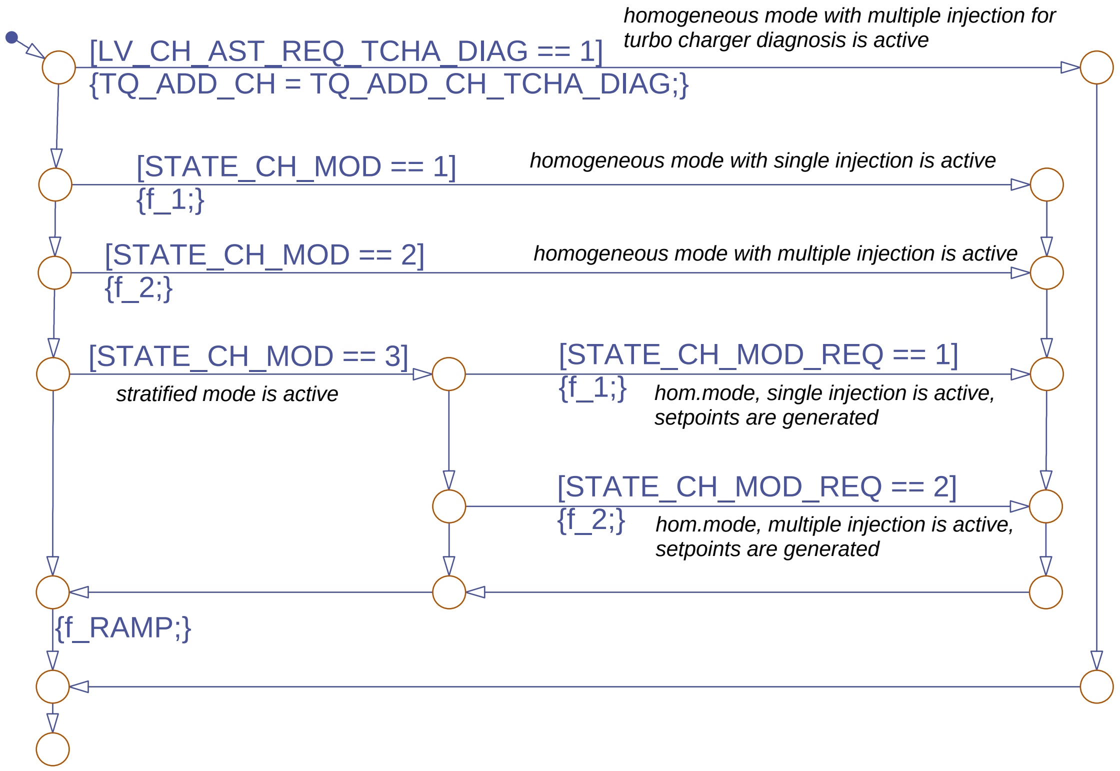


Figure 131:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_AST/Chart

30.22.1.1 Homogenous mode, single injection

30.22.1.1.1 Homogenous mode, single injection - Subsystem

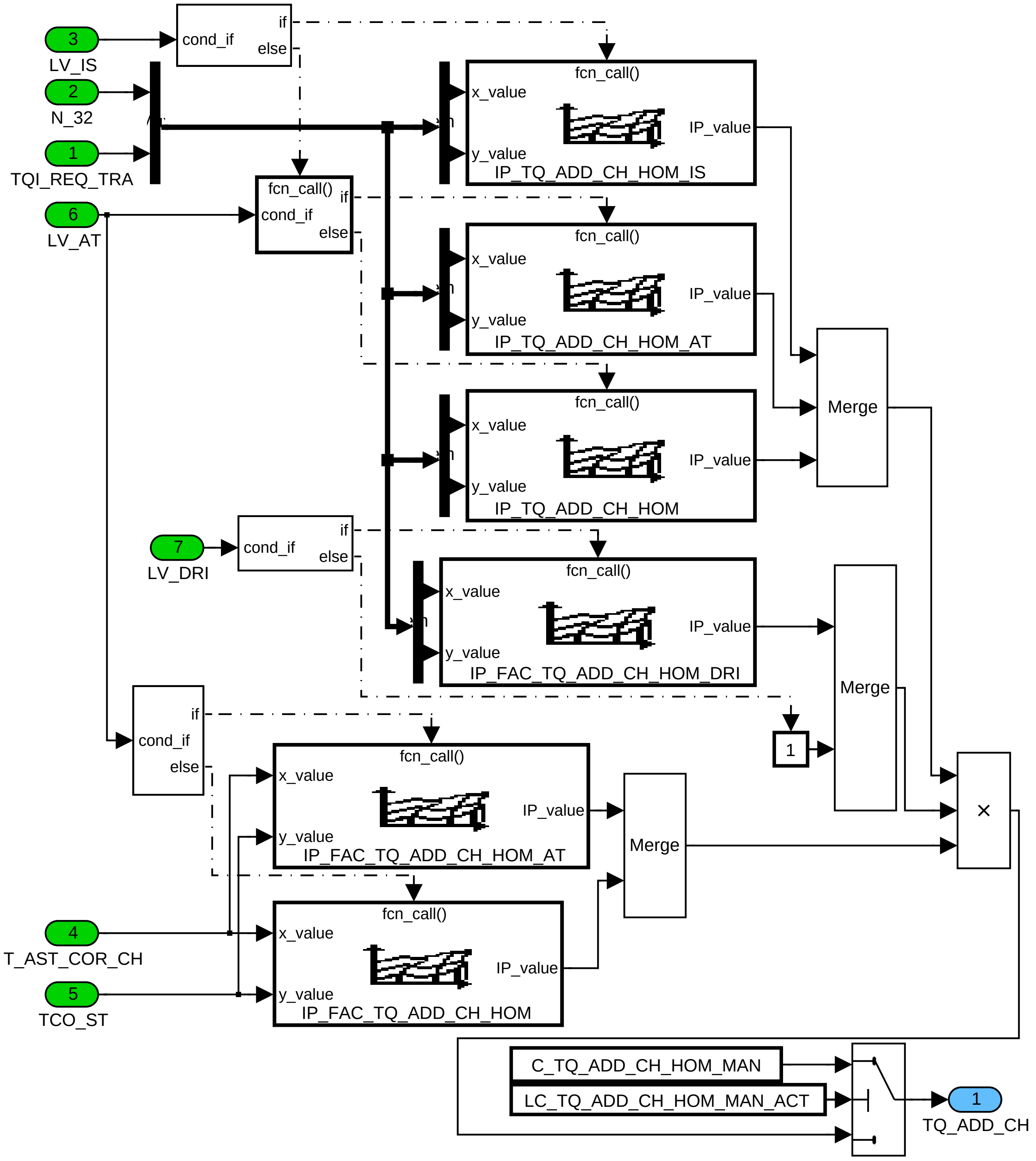


Figure 132:

Path: EXTC_REQGNCHT0/TQ_ADD_CH_AST/TQ_ADD_CH_AST_HOM/SUB

30.22.1.2 Homogenous mode, multiple injection

30.22.1.2.1 Homogenous mode, multiple injection - Subsystem

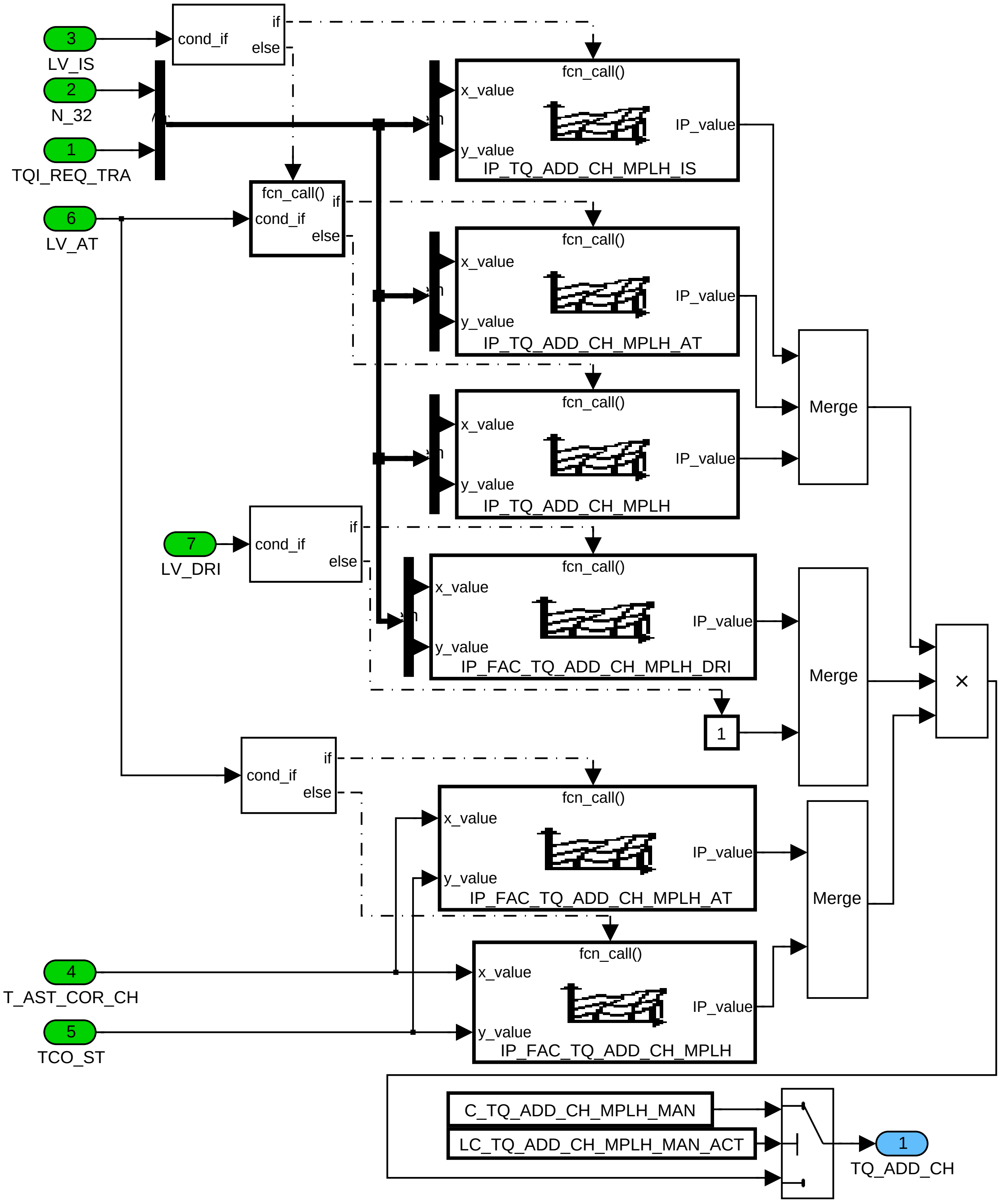


Figure 133:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_AST/TQ_ADD_CH_AST_MPLH/SUB

30.22.1.3 Ramp-function

30.22.1.3.1 Ramp-function - Subsystem

With IP_FAC_TQ_ADD_CH_AMP the torque reserve can be reduced with decreasing ambient pressure and so increasing elevation. The resulting torque reserve is filtered by a difference limiter.

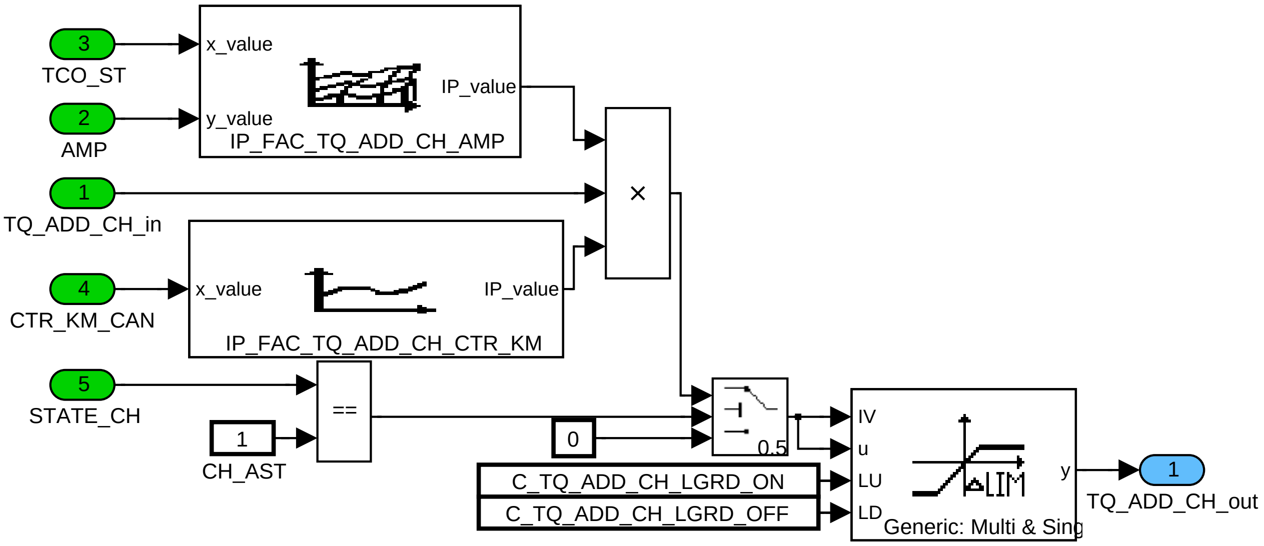


Figure 134:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_AST/clc_of_ramps/SUB

30.22.2 Catalyst heating in low load

In low load catalyst heating there is no homogeneous mode with split injection.

Application Conditions

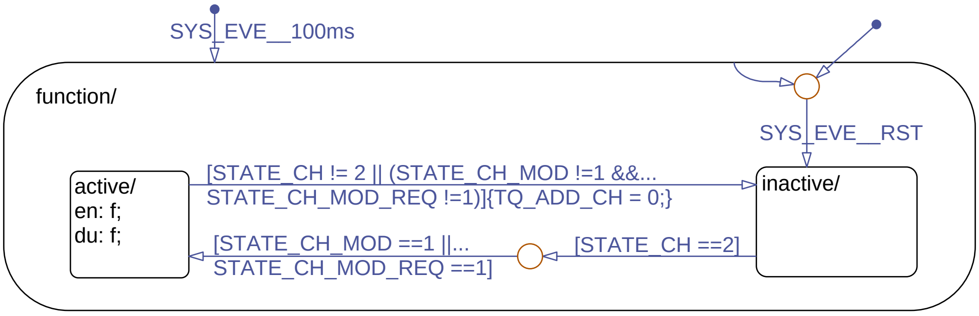


Figure 135:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_L/APP_CDN/Chart

Function description

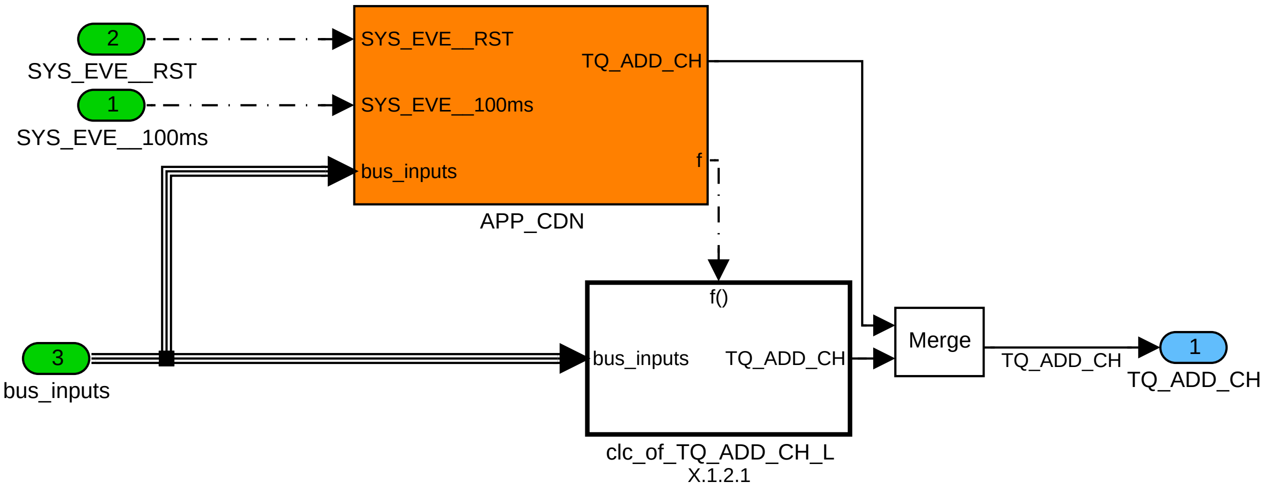


Figure 136:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_L

30.22.2.1 Torque reserve at low load catalyst heating

30.22.2.1.1 Torque reserve at low load catalyst heating - Subsystem

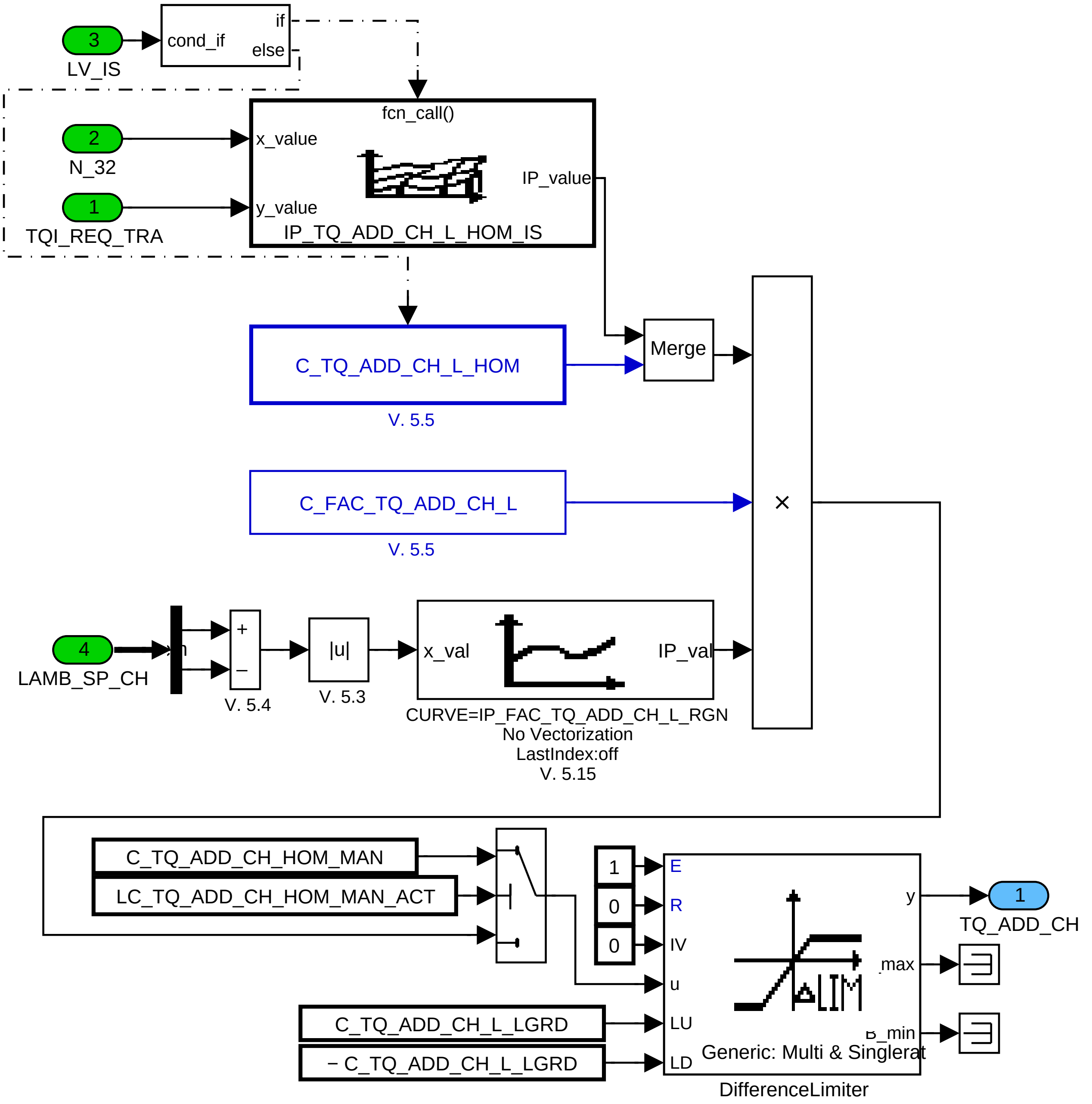


Figure 137:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_L/clc_of_TQ_ADD_CH_L/SUB

30.22.3 Catalyst heating for desulfation

TQ_ADD_CH is set by a controller, which controls the lowest NOx-trap temperature. Therefore TNT_DIF_CH_SO2P is the difference between the temperature setpoint for desulfation and the actual temperature (set in "Catalyst heating for desulfation strategy"). It is negative if temperature is too low.

Application Conditions

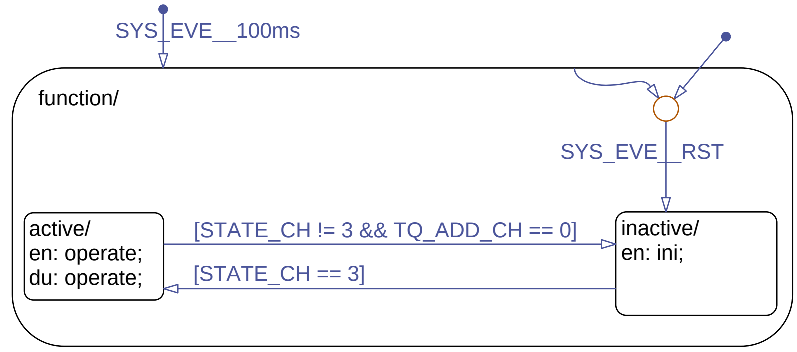


Figure 138:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_SO2P/APP_CDN/Chart

Function description

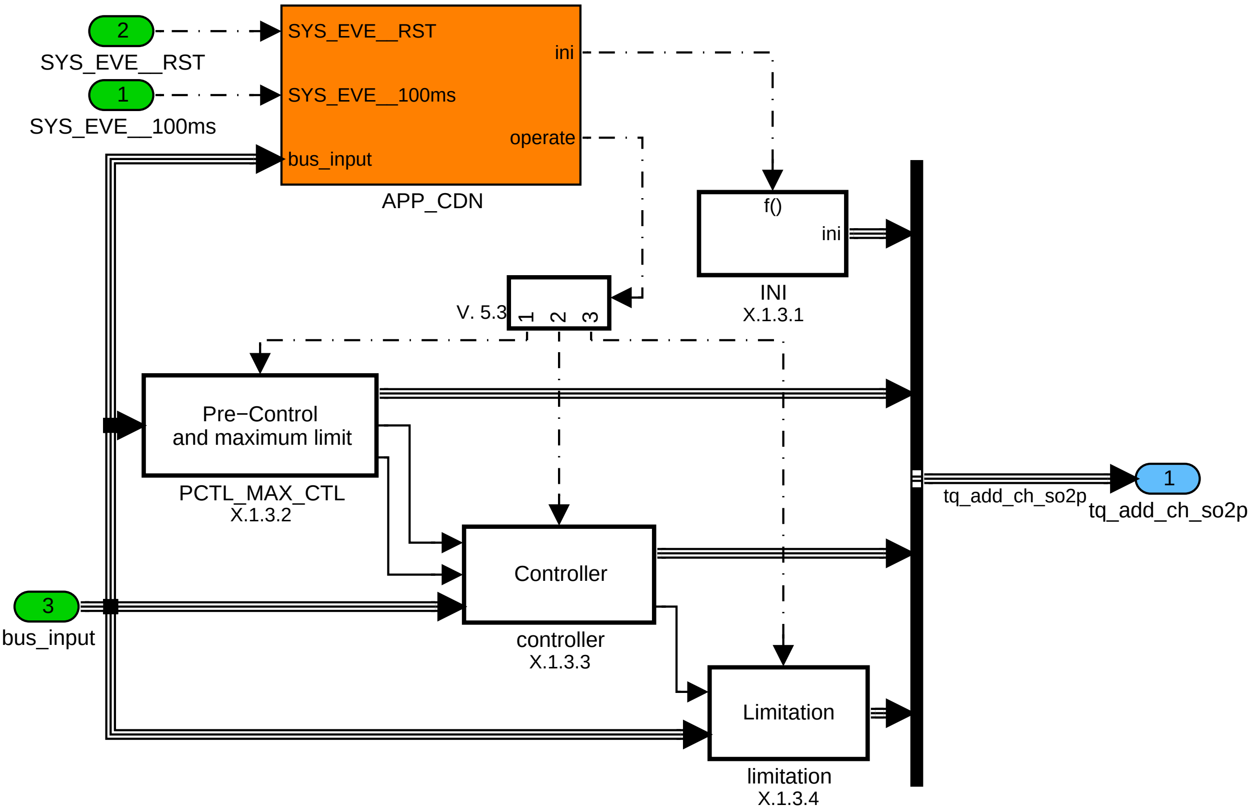


Figure 139:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_SO2P

30.22.3.1 Initialisation

The values are set to 0 at reset and once at function deactivation.

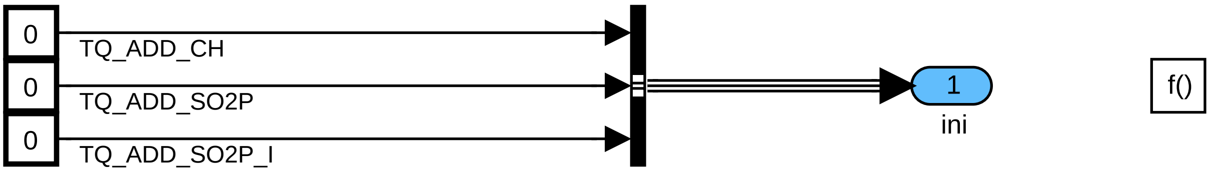


Figure 140:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_SO2P/INI

30.22.3.2 Precontrol and maximum limit

30.22.3.2.1 Precontrol and maximum limit - Subsystem

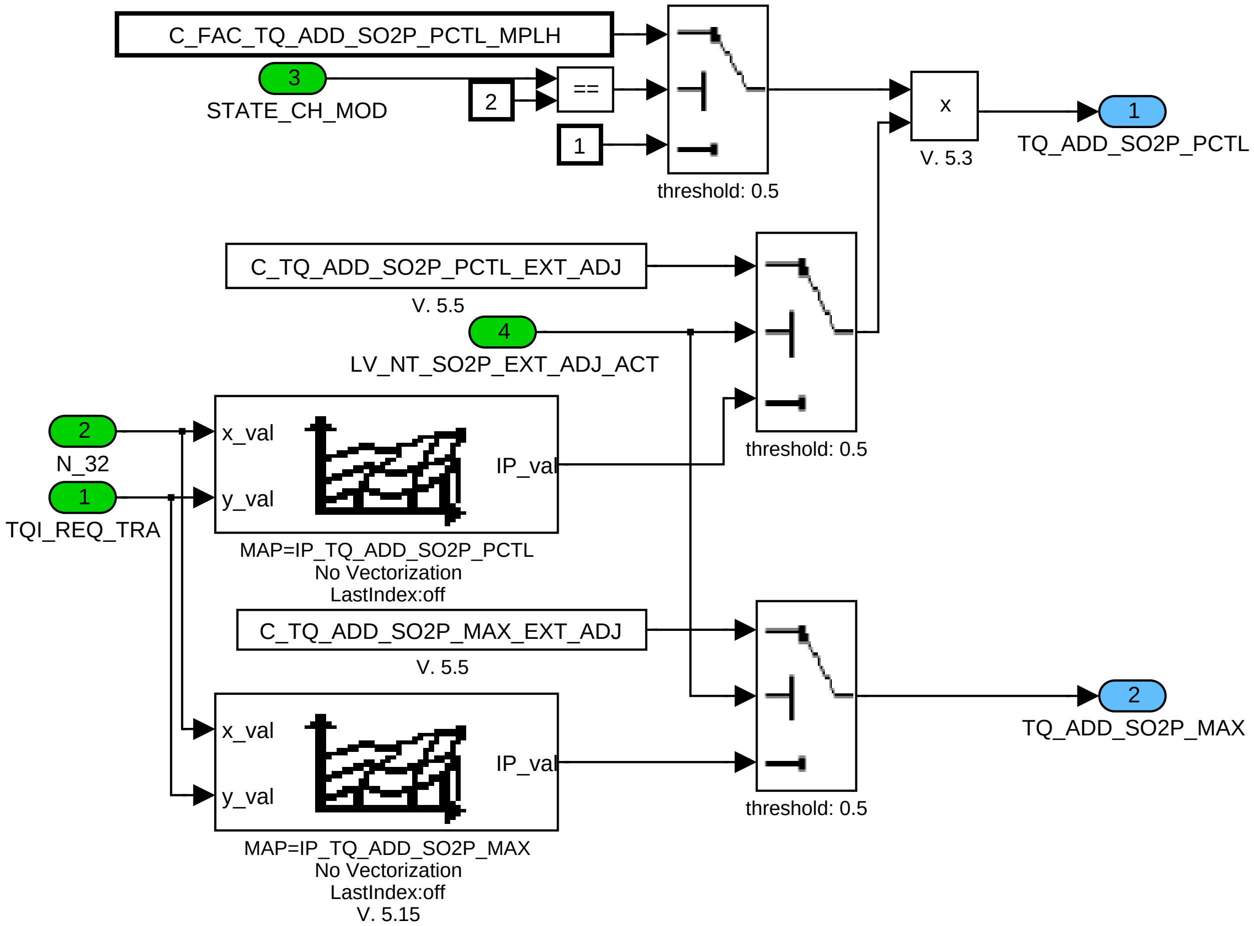


Figure 141:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_SO2P/PCTL_MAX_CTL/SUB

30.22.3.3 Controller of NOx-trap temperature

30.22.3.3.1 Controller of NOx-trap temperature - Subsystem

The controller consists of an integral and a proportional part (PI-controller). The I-part is frozen if the overall controller output exceeds the maximum available torque reserve TQ_ADD_MAX or the map-value.
The pre-controlled value for TQ_ADD_SO2P is added to the PI-controller output.

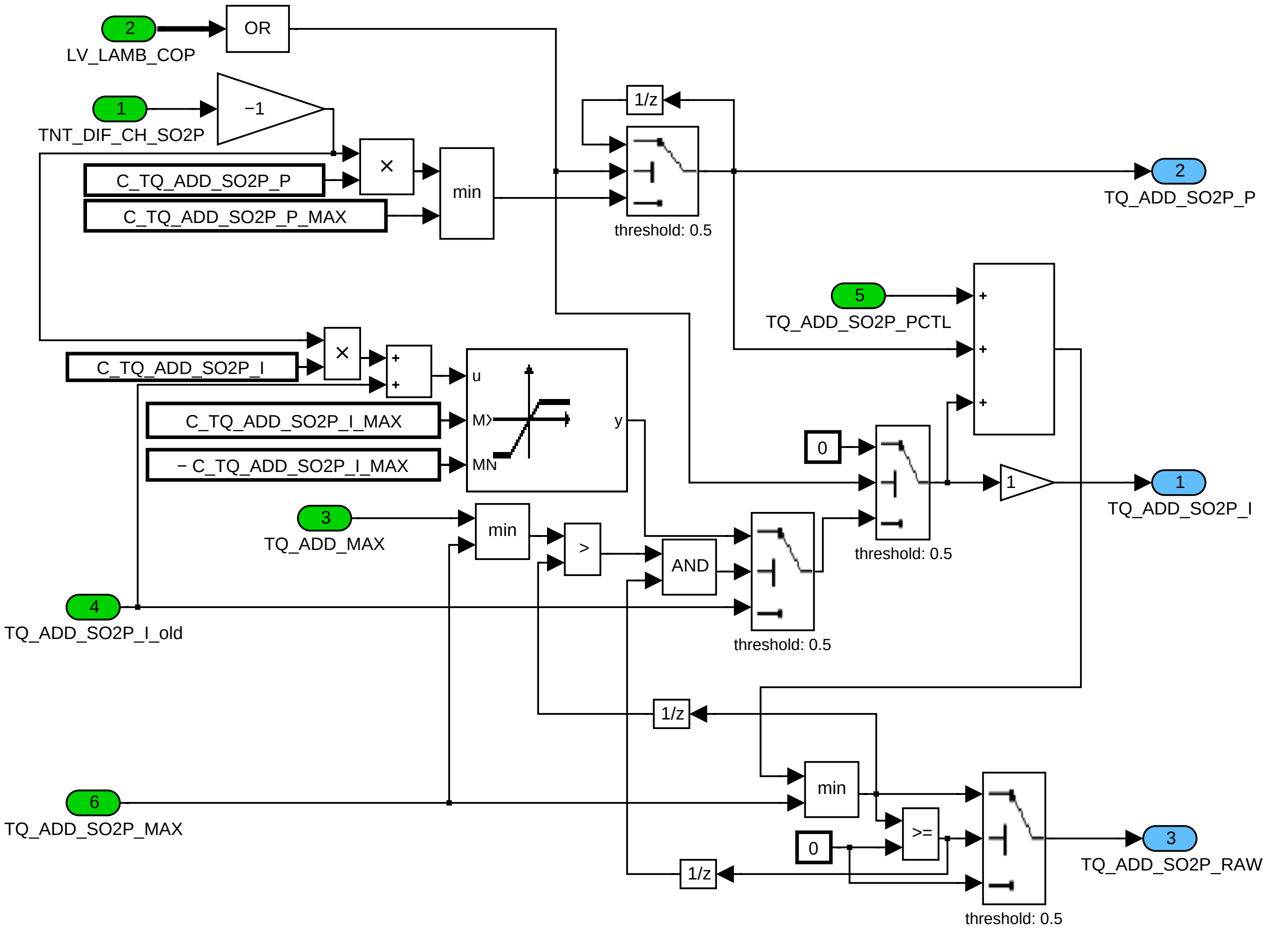


Figure 142:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_SO2P/controller/SUB

30.22.3.4 Limitation

30.22.3.4.1 Limitation - Subsystem

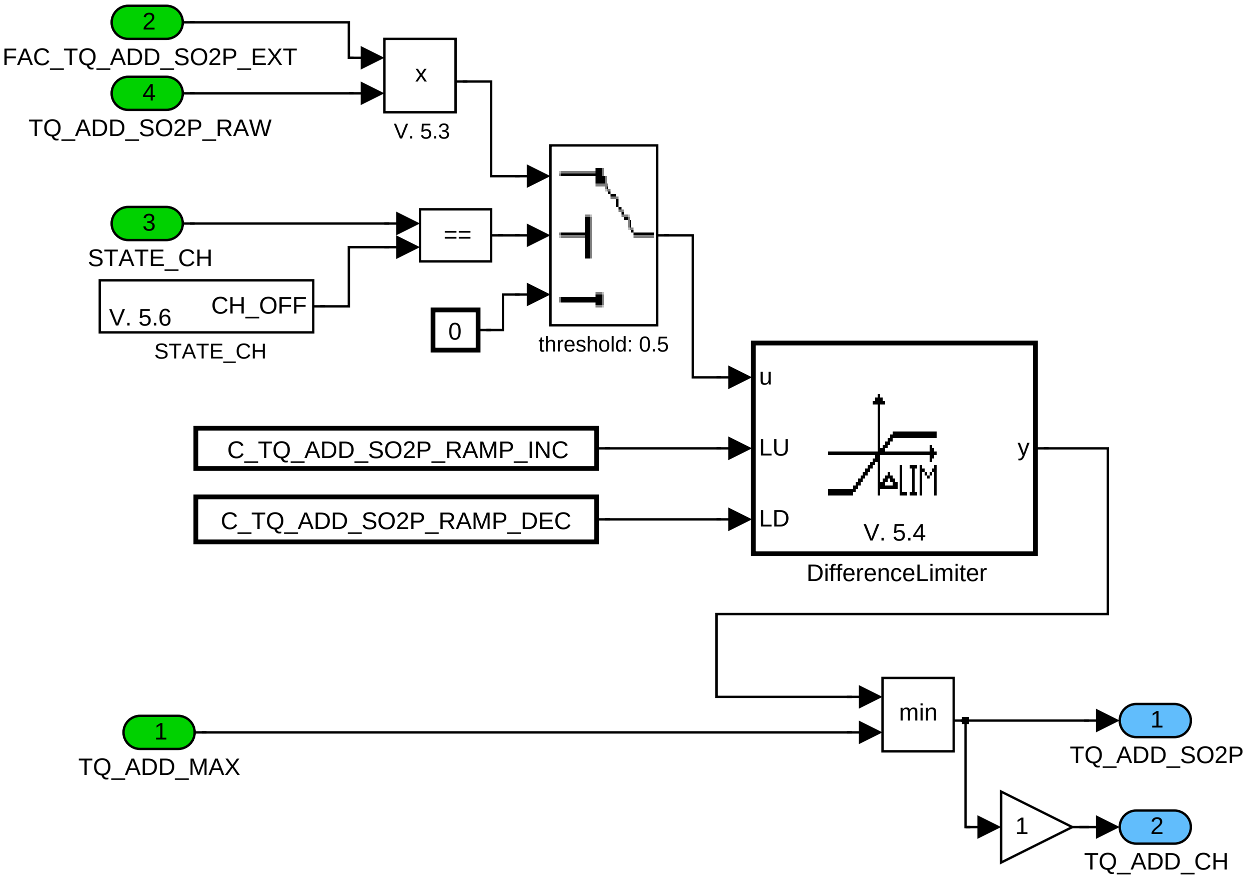


Figure 143:
Path: EXTC_REQGNCHT0/TQ_ADD_CH_SO2P/limitation/SUB

30.23 Customer adapation module :AGGR EXTC

30.23.1 Outputs for BMW functions which are defined as EXTC exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_atlsvc_if	O/V	0...1H	0...1	1	[-]
Testeranfordung für Diagnosefunktion der Abgasturbolader					
B_bsakt_zuli1	O/V	0...1H	0...1	1	[-]
Boolean for active catalyst overheating control bank 2					
B_bsakt_zuli2	O/V	0...1H	0...1	1	[-]
Boolean for active catalyst overheating control bank 1					
B_kath	O	0...1H	0...1	1	[-]
Catalyst heating function general					
Baw_katman1	O/V	0...FFFFH	0...65535	1	[-]
Statuswort Katheizanforderung - "priorisierter" Wunsch					
Baw_katman2	O/V	0...FFFFH	0...65535	1	[-]
Statuswort Katheizanforderung - "alternativ" Wunsch					
Drfkh_s	O/V	8000...7FFFH	-327.68...327.67	0.01	[%]
differences for S - CH					
Dwesb3_kh_s	O/V	E200...5A00H	-180...540	0.0234375	[°CRK]
Delta Zündwinkel Einspritzbeginn 3.EinspritzungSchicht					
Dwese2_kh_h	O/V	E200...5A00H	-180...540	0.0234375	[°CRK]
Delta Zündwinkel Einspritzende 2. Einspritzung homogen					
Dzw_kh_s	O/V	FE0C...258H	-50...60	0.1	[°CRK]
Catalyst heating ignition angle difference (stratified)					
FAC_MAF_REL_EGR_COR	O/V	0...BB8H	0...300	0.1	[%]
Relative filling, EGR-corrected					
FAC_TQ_ADD_SO2P_EXT	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
External factor reducing torque reserve for desulfation catalyst heating					
La_bszuli1	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda for catalyst overheating prevention bank 1					
La_bszuli2	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda for catalyst overheating prevention bank 2					
La_kh1	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda catalyst heating					
La_kh2	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda catalyst heating					
LAMB_COP[NC_CBK_EX_NR]	O/V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda for catalyst overheating prevention bank 1+2					
LV_CH	O/V	0...1H	0...1	1	[-]
Auxiliary Function " Catalyst Heating Spark"					
LV_CH_AST_REQ_TCHA_DIAG	O/V	0...1H	0...1	1	[-]
Activation of catalyst heating function requested for turbo charger diagnosis					
LV_CH_INH	O/V	0...1H	0...1	1	[-]
Inhibition of general catalyst heating					
LV_LAMB_COP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Lambda setpoint calculation for catalyst overheating prevention is active					
LV_PUC_INH_TEMP_CAT	O/V	0...1H	0...1	1	[-]
Flag to inhibit PUC operation due to too high TEMP_CAT					
LV_TCHA_DIAG_REQ	O/V	0...1H	0...1	1	[-]
Request of ATL-Diagnosis					
LV_TCHA_DIAG_REQ_OFF	V	0...1H	0...1	1	[-]
request to switch off ATL-diagnosis					
Mdi_res_kh	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Torque reserve for catalyst heating					
P_rail_kh	O/V	0...FFFFH	0...39.99938	0.6104e-3	[MPa]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Rail pressure from catalyst heating manager					
Prio_katman1	O/V	80...7FH	-128...127	1	[-]
Priority number (highest wish) of catalyst heating manager					
Prio_katman2	O/V	80...7FH	-128...127	1	[-]
Priority number (second wish) of catalyst heating manager					
TQ_ADD_CH_TCHA_DIAG	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for turbocharger diagnosis - to be realized via catalyst heating					
Wesb1kh_h[NC_CYL_NR]	O/V	E200...5A00H	-180...540	0.0234375	[°CRK]
Einspritzbegin 1. Einspritzung homogen					

Input data:

B_atlsvc	B_atlsvc_khm	B_bs1	B_bs2
B_vb_sa	B_vbkhm_notl	Baw_ist	C_MAF_REF
CRK_INJ_BAS[NC_CYL_NR]	EOI_2_DELTA_MPLH_CH	F_bsmdres	FUP_SP_CH
IGA_DIF_S_CH	La_bs1	La_bs2	LAMB_COP_CUS_i
LAMB_SP_CH[NC_CBK_EX_NR]	LV_TCHA_DIAG_EXT_REQ	MAF_DIF_S_CH	Md_res_atlsvc
NC_CBK_EX_NR	NC_CYL_NR	NR_PRI_CH_MOD	NR_PRI_CH_MOD_REQ
Rfv_vns	SOI_1_MPLH_CH[NC_CYL_NR]	SOI_MPLP_DIF_CH	STATE_CH
STATE_CH_MOD_IVVT	STATE_CH_MOD_IVVT_REQ	TQ_ADD_CH	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation at reset or at exit power latch phase:

0 ,except La_bszuli1/2 = 2.00, La_kh1/2 = 1.00

Recurrence : seg Dwesb3_kh_s, Dwese2_kh_h,
Wesb1kh_h[ANZAHL_ZYLINDER], P_rail_kh, Dzw_kh_s
10 ms Mdi_res_kh, La_kh1/2,
La_bszuli1, La_bszuli2, Drfkh_s, Baw_katman1, Baw_katman2
100ms Prio_katman1/2, B_kath, B_atlsvc_if
200ms B_bsakt_zuli1, B_bsakt_zuli2

Activation: every engine state, except power latch phase
Deactivation: at power latch phase, except B_kath
Values at Deactivation: 0, except P_rail_kh: last calculated value

Formula section:

Remark: all formulas are valid in a **physical** meaning

Mdi_res_kh = TQ_ADD_CH
La_bszuli1 = LAMB_COP_CUS_1
La_bszuli2 = LAMB_COP_CUS_2
B_bsakt_zuli1 = LV_LAMB_COP_CUS_1
B_bsakt_zuli2 = LV_LAMB_COP_CUS_2
Dwesb3_kh_s = SOI_MPLP_DIF_CH
Dwese2_kh_h = EOI_2_DELTA_MPLH_CH
La_kh1 = LAMB_SP_CH[1]
La_kh2 = LAMB_SP_CH[2]

Baw_katman1 = STATE_CH_MOD_IVVT_REQ

Baw_katman2 = STATE_CH_MOD_IVVT

Dzw_kh_s = IGA_DIF_S_CH

Wesb1kh_h[ANZAHL_ZYLINDER]
= SOI_1_MPLH_CH[NC_CYL_NR] + CRK_INJ_BAS[0]

Drfkh_s = MAF_DIF_S_CH / C_MAF_REF
Prio_katman1 = NR_PRI_CH_MOD_REQ
Prio_katman2 = NR_PRI_CH_MOD
P_rail_kh = FUP_SP_CH
B_kath = LV_CH

B_atlsvc_if = LV_TCHA_DIAG_REQ

30.23.2 Outputs for SV aggregates

FUNCTION DESCRIPTION:

Adaptation to BMW environment

Application conditions:

Initialisation at reset: 0, except LAMB_COP[NC_CBK_EX_NR] = 2.00
FAC_TQ_ADD_SO2P_EXT = 1.00

Initalisation at LV_IGK 0-> 1: LV_TCHA_DIAG_REQ_OFF = 0

Recurrence: 10ms: LAMB_COP[NC_CBK_EX_NR],
LV_LAMB_COP[NC_CBK_EX_NR],
LV_LAMB_COP, FAC_MAF_REL_EGR_COR, LV_CH ,
LV_CH_INH, TQ_ADD_CH_TCHA_DIAG

100ms: LV_PUC_INH_TEMP_CAT,
LV_TCHA_DIAG_REQ, LV_TCHA_DIAG_REQ_OFF,
LV_CH_AST_REQ_TCHA_DIAG

200ms: FAC_TQ_ADD_SO2P_EXT

Activation: every engine state

Formula section:

LAMB_COP[1] = La_bs1
LAMB_COP[2] = La_bs2

```
LV_LAMB_COP[1]          =  B_bs1
LV_LAMB_COP[2]          =  B_bs2
LV_LAMB_COP             =  B_bs2 or B_bs1
LV_PUC_INH_TEMP_CAT     =  B_vb_sa
FAC_MAF_REL_EGR_COR =  Rfv_vns
if     Baw_ist = Baw_katman1 or Baw_ist = Baw_katman2
then          LV_CH = 1
else          LV_CH = 0
endif
LV_CH_INH =  B_vbkhm_notl
FAC_TQ_ADD_SO2P_EXT =  F_bsmdres
TQ_ADD_CH_TCHA_DIAG   = Md_res_atlsvc
if              B_atlsvc switch 1 -> 0
then            LV_TCHA_DIAG_REQ_OFF = 1
endif
LV_TCHA_DIAG_REQ =        (LC_TCHA_DIAG_EXT_REQ or LV_TCHA_DIAG_EXT_REQ)
                       and   not(LV_TCHA_DIAG_REQ_OFF)
LV_CH_AST_REQ_TCHA_DIAG = B_atlsvc_khm
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TCHA_DIAG_EXT_REQ	1	0...1H	0...1	1	[-]
switch for manual start of ATL-diagnosis (default: 0)					

31 Exhaust system temperature determination

Table of Contents

Index

31.1 EXTD – Requirements for infrastructure

Input data:

NC_NR_TEG_SENS			

Export actions:

ACTION_INFR_GetVpTeg (IN <CtrTegSens>, OUT <VpTegMes>,)
Action to read the sensor voltage of the different exhaust gas temperature sensors from the ADC

Description for actions:

ACTION_INFR_GetVpTeg (CtrTegSens, VpTegMes)					
The input parameter <CtrTegSens> gives the basic software the information which of the NC_NR_TEG_SENS exhaust gas temperature sensor voltages has to be provided to the application sofware. Every time the action is called the value of the corresponding AD channel is read out and delivered to the application sofware. Default SYNCHRONISATION is ASYNCHRONOUS					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
CtrTegSens	IN	0...5	1...6	1	-
Index of the exhaust gas temperature sensor					
VpTegMes	OUT	0...7FFFH	0...4.999847412109375	1.52587890625e-4	V
Exhaust gas temperature sensor voltage					

FUNCTION DESCRIPTION:

General information:

This chapter describes the acquistion of exhaust gas temperature sensor signals based on an analog sensor voltage.

Requirements for ACTION_INF_GetVpTeg:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
CtrTegSens	1			VpTegMes is the digitalizied voltage of the sensor with the index CtrTegSens	
VpTegMes	Not relevant	< 1% of reference supply voltage for sensor	ADC: 10bit Value: 15bit	The digitalisation of the value has to be carried out less than 10ms before the action calls the result.	

Diagnosis: no electrical diagnosis done

Coincidence requirements: The AD conversion is performed autonomously by the infrastructure. The recurrence has to be less than 10ms to guarantee that digitalisation value is not older than these 10ms.

31.2 AGGR EXTD Adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TEG_DYN_STOP	V/O/S	0...7FF0H	0...2047	0.0625	°C
Modelled exhaust gas temperature at last engine stop					

Input data:

TEG_DYN			

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: at Engine run to Engine stop (ERU_to_ES)

TEG_DYN_STOP = TEG_DYN

Recurrence: -

Activation: -

Deactivation: -

Formula section:

-

31.3 Exhaust Gas Temperature

31.3.1 Aquisition of VP_TEG_PCAT_DOWN

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VP_TEG_PCAT_DOWN	O/V	0...7FFFH	0...4.99984	0.1526e-3	[V]
Catalyst temperature sensor raw acquisition					

Import actions:

ACTION_INFR_GetVpTeg (IN <>, OUT <>)

Sampling Rate: Ta = 0.1sec

Application conditions:

Activation: always calculated

Deactivation: -

Global configuration data

Data	Value
NC_NR_TEG_SENS	1

ACTION_INFR_GetVpTeg(1,VP_TEG_PCAT_DOWN)

// all sensor values are imported and provided as output

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_TEG_SENS	1	1...6H	1...6	1	[-]
number of exhaust gas temperature sensor in system					

31.3.2 Calculation of Exhaust Gas Temperatures

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TEG_PCAT_DOWN[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Exhaust gas temperature upstream of precat					
TEG_PCAT_DOWN_RAW	V	0...7FFFH	0...2047.9375	0.0625	[K]
Exhaust gas temp downstream of pre-catalyst, raw value i.e. not replaced by substitute value, wide range					
TEG_PCAT_DOWN_ST	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Exhaust gas temperature downstream of pre-catalyst at engine start					
TEG_PCAT_DOWN_WIDE_RNG	V	0...7FFFH	0...2047.9375	0.0625	[K]
Exhaust gas temperature downstream of pre-catalyst with wide range, unit: Kelvin					
TEG_PCAT_DOWN_WIDE_RNG_ST	O/V	0...7FFFH	0...2047.9375	0.0625	[K]
Exhaust gas temperature downstream of pre-catalyst at engine start with wide range, unit: Kelvin					

Input data:

LV_ERR_PREL_TEG_PCAT_DOWN	LV_ERR_TEG_PCAT_DOWN	LV_IGK	MAF_KGH
TEG_CAT_DOWN_MDL[NC_CBK_EX_NR]			

General information:

The exhaust gas temperature sensor acquisition is suitable for single branch (only one exhaust bank) and twin branch exhaust lines (two exhaust banks).

For a twin branch exhaust line the exhaust gas temp. acquisition has the same structure for both exhaust lines. To differentiate variables and calibration data the **index i** is used:

i = 1 refers to temp. sensor in the first branch of a twin branch exhaust line or the temp. sensor in a single branch exhaust line

i = 2 refers to temp. sensor in the second branch of a twin branch exhaust line

This version is intended for an engine with a configuration that uses one temperature sensor. The exhaust gas temp. downstream of pre-cat.). TEG_PCAT_DOWN is used for the temperature modelling of the NOx catalyst monolith temperature (for GDI engine).

TEG_PCAT_DOWN_ST is the exhaust gas temperature at engine start used for dew point calculation of the NOx catalyst.

As the range of TEG_PCAT_DOWN and TEG_PCAT_DOWN_ST is limited to 0..1024°C, TEG_PCAT_DOWN_WIDE_RNG and TEG_PCAT_DOWN_WIDE_RNG_ST are calculated in a similar manner with a range of 0..2048K (equals -273..1775°C) to provide information about subzero and very high temperatures.

Sampling Rate: Ta = 0.1sec

Application conditions:

Activation: Ignition on LV_IGK = 1

Deactivation: Ignition off LV_IGK = 0

Initialisation:

At the transition LV_IGK 0 -> 1 (including reset) TEG_PCAT_DOWN_RAW is initialised with IP_TEG_WIDE_RNG__VP_TEG(VP_TEG_PCAT_DOWN). In that way the value TEG_PCAT_DOWN_RAW is available for the module “Exhaust gas temperature sensor diagnosis (TEG)” to make the “Plausibility check: Cold start”. That is necessary since the module “Exhaust gas temperature sensor diagnosis (TEG)” must have been performed since a valid LV_ERR_PREL_TEG_PCAT_DOWN is needed to determine TEG_PCAT_DOWN_ST and TEG_PCAT_DOWN_WIDE_RNG_ST.

At the first calculation of the module “Exhaust gas temperature” and at transition LV_ES 1 -> 0 TEG_PCAT_DOWN_ST and TEG_PCAT_DOWN_WIDE_RNG_ST are initialised:

```
If   LV_ERR_PREL_TEG_PCAT_DOWN = 0
     Then
          TEG_PCAT_DOWN_ST = IP_TEG__VP_TEG(VP_TEG_PCAT_DOWN)
          TEG_PCAT_DOWN_WIDE_RNG_ST =
                IP_TEG_WIDE_RNG__VP_TEG(VP_TEG_PCAT_DOWN)
     Else
          TEG_PCAT_DOWN_ST = C_TEG_PCAT_DOWN_ST_INI
          TEG_PCAT_DOWN_WIDE_RNG_ST = TEG_PCAT_DOWN_ST
End
```

FUNCTION DESCRIPTION:

The voltage VP_TEG_PCAT_DOWN is converted with a linear interpolation to an exhaust gas temperature downstream of the pre-catalyst TEG_PCAT_DOWN / TEG_PCAT_DOWN_WIDE_RNG.

The logical variable LV_ERR_PREL_TEG_PCAT_DOWN indicates a signal out of the specified range (short to ground or short to UB or line broken) or a not plausible temperature value. If active (LV_ERR_PREL_TEG_PCAT_DOWN = 1), the last valid temperature value is frozen until the error is debounced (LV_ERR_TEG_PCAT_DOWN = 1) or until a valid temp. value is obtained again.

With debounced error (LV_ERR_TEG_PCAT_DOWN = 1) a substitute value TEG_CAT_DOWN_MDL is assigned.

In normal operating mode (LV_ERR_PREL_PCAT_DOWN = 0 and LV_ERR_PCAT_DOWN = 0) the temperature depends on the temperature sensor voltage via map.

Formula section:

```
TEG_PCAT_DOWN_RAW$_N$ = IP_TEG_WIDE_RNG__VP_TEG(VP_TEG_PCAT_DOWN$_N$)
If   LV_ERR_TEG_PCAT_DOWN = 1
Then
     TEG_PCAT_DOWN_1 = TEG_CAT_DOWN_MDL - C_TEG_PCAT_DOWN_ADD_ERR
     TEG_PCAT_DOWN_2 = TEG_PCAT_DOWN_1
     TEG_PCAT_DOWN_WIDE_RNG = TEG_PCAT_DOWN_1
Else
     If  LV_ERR_PREL_TEG_PCAT_DOWN = 1
     Then
          TEG_PCAT_DOWN_1$_N$ = TEG_PCAT_DOWN_1$_{N-1}$
```

```
            $TEG\_PCAT\_DOWN\_2_{N} = TEG\_PCAT\_DOWN\_2_{N-1}$
            $TEG\_PCAT\_DOWN\_WIDE\_RNG_{N} = TEG\_PCAT\_DOWN\_WIDE\_RNG_{N-1}$
    else
            $TEG\_PCAT\_DOWN\_1_{N} = IP\_TEG\_\_VP\_TEG$
            $TEG\_PCAT\_DOWN\_2_{N} = TEG\_PCAT\_DOWN\_1_{N}$
            $TEG\_PCAT\_DOWN\_WIDE\_RNG_{N} = IP\_TEG\_WIDE\_RNG\_\_VP\_TEG$
    end
end
```

Calibration data:

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEG_PCAT_DOWN_ADD_ERR	1	0...FFH	0...510	2	[°C]
Additional offset on the temperature model in case of sensor error					
C_TEG_PCAT_DOWN_ST_INI	1	0...FFFFH	0...1023.98437	0.015625	[°C]
Initialisation value of TEG_PCAT_DOWN_ST_i used in case of sensor error					
IP_TEG__VP_TEG	16	0...FFFFH	0...1023.98437	0.015625	[°C]
LDP_VP_TEG__TEG	16	0...7FFFH	0...4.99984	0.1526e-3	[V]
conversion table of exhaust gas temp.					
IP_TEG_WIDE_RNG__VP_TEG	8	0...FFFFH	0...2047.96875	0.03125	[K]
LDP_VP_TEG__TEG_WIDE_RNG	8	0...7FFFH	0...4.99985	0.0001526	[V]
conversion table of exhaust gas temp. with extended range					

31.4 Exhaust gas temperature model and dew-point recognition

31.4.1 Detection of dew point

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_COLD_ST	O/V/S	0...FFH	0...255	1	[-]
Number of cold starts					
INT_TEG_MAF	V	0...FFFFH	0...9102.08	0.1388888	[g]
sum of the weighted air mass flow values used for the dew point recognition					
LV_TEG_CAT_DOWN_MIN_THD	O/V	0...1H	0...1	1	[-]
Dew-point recognition downstream of catalyst					
LV_TEG_MIN_THD	O/V	0...1H	0...1	1	[-]
Dew-point recognition					
EFF_IGA_AV_DEW	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Ignition efficiency actual ignition angle bank selective for dew point					

Input data:

LV_PUC	LV_ST	LV_IGK	LV_ES
MAF_KGH	EFF_IGA_AV	TCO_ST	TIA
LV_HOM_ACT			

FUNCTION DESCRIPTION:

General information:

The exhaust gas temperature model has been removed from Siemens VDO side and is now modelled in the BMW objects. This module does satisfy the open interfaces of the dew point recognition between Siemens VDO and BMW.

Application conditions:

Initialisation: **IF** (there are no adaptation values stored **OR** the EEPROM does not work)
THEN CTR_COLD_ST = 0 (default value is set)
ELSE CTR_COLD_ST = CTR_COLD_ST stored in EEPROM (previously stored value is assigned)
END
At transition LV_ES = 1 -> 0
INT_TEG_MAF = 0 (Mass flow integral for dew point detection)
LV_TEG_MIN_THD = 0 (State: dew point not passed)
LV_TEG_CAT_DOWN_MIN_THD = 0 (State: dew point downstream of cat. not passed)

Recurrence: 1000ms

Activation: LV_ES and LV_ST = 0 (after leaving the engine state ST)

Deactivation: LV_ES = 1 (engine state ES is active)

Formula section:

Calculation of the number of cold starts:

After leaving the engine state ST the value of CTR_COLD_ST is incremented by the increment one.

The value CTR_COLD_ST is a measure of the number of cold-start tests and has to be stored in the flash-memory when the engine is switched off (LV_IGK = 0). After a new start of the engine CTR_COLD_ST is initialized with the stored value.

```
If        LV_HOM_ACT = 1
Then      EFF_IGA_AV_DEW = EFF_IGA_AV
Else      EFF_IGA_AV_DEW = 1
Endif;
```

```
IF   LV_PUC = 0  AND (LV_TEG_MIN_THD = 0 OR LV_TEG_CAT_DOWN_MIN_THD = 0)
     % For PUC active no increment of INT_TEG_MAF$_N$
THEN
     INT_TEG_MAF$_N$  = INT_TEG_MAF$_{N-1}$ + IP_INT_TEG(EFF_IGA_AV_DEW ,MAF_KGH) *
     MAF_KGH$_N$ * 1000ms
END
```

Dew point upstream of catalyst:

```
IF   INT_TEG_MAF$_N$ < IP_INT_TEG_THD(TCO_ST, TIA) *
       ID_CTR_COLD_ST(CTR_COLD_ST)
THEN
     LV_TEG_MIN_THD = 0                   % dew point not passed
ELSE
     LV_TEG_MIN_THD = 1                   % dew point passed
END
```

Dew point downstream of catalyst:

```
IF   INT_TEG_MAF$_N$ < IP_INT_TEG_CAT_DOWN_THD(TCO_ST, TIA) *
     ID_CTR_COLD_ST(CTR_COLD_ST)
THEN
     LV_TEG_CAT_DOWN_MIN_THD = 0          % dew point not passed
ELSE
     LV_TEG_CAT_DOWN_MIN_THD = 1          % dew point passed
END
```

Reset of cold start counter CTR_COLD_ST:

```
IF   LV_TEG_MIN_THD = 1 AND LV_TEG_CAT_DOWN_MIN_THD = 1
THEN
     CTR_COLD_ST = 0                      % reset of cold start counter
END
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_CTR_COLD_ST	6	0...FFH	0...3.98437	0.015625	[-]
LDP_CTR_COLD_ST__CTR_COLD_ST	6	0...FFH	0...255	1	[-]
weighting factor for the consideration of the number of cold-start tests					
IP_INT_TEG	6*6	0...FFH	0...15.9375	0.0625	[-]
LDP_EFF_IGA_AV_DEW__INT_TEG	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_MAF_KGH__INT_TEG	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
weighting factor for the consideration of the ignition angle and MAF_KGH influence for dew point determination					
IP_INT_TEG_THD	8*6	0...FFFFH	0...9102.08	0.1388888	[g]
LDPM_TCO_ST_1_EXTD	8	0...FEH	-48...142.5	0.75	[°C]
LDPM_TIA_1_EXTD	6	0...FEH	-48...142.5	0.75	[°C]
threshold air mass flow integral for detection of dew point upstream of catalyst					
IP_INT_TEG_CAT_DOWN_THD	8*6	0...FFFFH	0...9102.08	0.1388888	[g]
LDPM_TCO_ST_1_EXTD	8	0...FEH	-48...142.5	0.75	[°C]
LDPM_TIA_1_EXTD	6	0...FEH	-48...142.5	0.75	[°C]
threshold air mass flow integral for detection of dew point downstream of catalyst					

31.5 Exhaust Gas temperature model (Appl. Inc.)

31.5.1 Mean value calculations for the TEG model

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NR_CYL_SCC_TEG[NC_CBK_EX_NR]	V/O	0...1AH	0...13	0.5	[-]
Number of shut off cylinders for TEG calculation					

Input data:

INH_CYC_IV	INH_SWI_IV	LV_IGK	

General information:

The TEG model is calculated at the recurrence of 500 ms. For this it is nessesary to calculate mean values by high frequency influency variables. The sampling of the input datas are 10 ms.

Application conditions:

Initialisation:	-
Recurrence:	10ms
Activation:	LV_IGK = 1
Deactivation:	-

FUNCTION DESCRIPTION:

Cylinder shut off regonisation:

This function shows bank selective the mean value of shut off cylinders about 500 ms. After this time the sampling is starting again.

N53 / N54:

⇐ Logic Cylinder Number

INH_CYC_IV = |X|X|X|X|X|X|X|X|X|

Bank 1 (log): 0; 2; 4 Log: 0 1 2 3 4 5
Bank 2 (log): 1; 3; 5 Pys.: R6 Cyl: 1 5 3 6 2 4

INH_SWI_IV = |X|X|X|X|X|X|X|X|X|

Bank 1 (log): 0; 2; 4 Log: 0 1 2 3 4 5
Bank 2 (log): 1; 3; 5 Pys.: R6 Cyl: 1 5 3 6 2 4

N43:

⇐ Logic Cylinder Number

INH_CYC_IV = |X|X|X|X|X|X|X|X|X|

Bank 1 (log): 0; 2 Log: 0 1 2 3

Bank 2 (log): 1; 3; Pys.: R4 Cyl: 1 3 4 2

INH_SWI_IV = |X|X|X|X|X|X|X|X|X|

Bank 1 (log): 0; 2; Log: 0 1 2 3

Bank 2 (log): 1; 3; Pys.: R4 Cyl: 1 3 4 2

NR_CYL_SCC_TEG_RAW[i]: Number of shut off cylinder bank i

MAX (INH_SWI_IV ; INH_CYC_IV)

Formula section:

```
IF      counter < 50  (Sampling 10ms)
Then    NR_CYL_SCC_TEG_RAW[i]        =
            (NR_CYL_SCC_TEG_RAW[i] $_{=n-1}$+ Number of shut off cylinder bank i )
        counter + 1
Else    counter = 50
        NR_CYL_SCC_TEG[i]        = NR_CYL_SCC_TEG_RAW[i] / 50
        NR_CYL_SCC_TEG_RAW[i] and counter reset to 0
Endif
```

31.5.2 Conversion of temperature outputs for interfaces

31.5.2.1 Exhaust temperature for interfaces

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TEG_MIN_THD[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Dew-point recognition					
LV_TEG_CAT_DOWN_MIN_THD[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
dew point has been reached pre/post cat. specific bank selective downstream					

Input data:

LV_TEG_MIN_THD	LV_TEG_CAT_DOWN_MIN_THD		

Description:

A conversion is necessary to support the different resolutions in other modules.

Application conditions:

Initialisation: -

Recurrence: 500ms

Activation: at all engine operating states

Deactivation: -

Formula section:

LV_TEG_MIN_THD_1 = LV_TEG_MIN_THD
LV_TEG_MIN_THD_2 = LV_TEG_MIN_THD
LV_TEG_CAT_DOWN_MIN_THD_1 = LV_TEG_CAT_DOWN_MIN_THD
LV_TEG_CAT_DOWN_MIN_THD_2 = LV_TEG_CAT_DOWN_MIN_THD

31.6 NOx Catalyst Monolith Temperature Model

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TEG_PCAT_DOWN_COR[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Sensor input NOx catalyst monolith temperature					
LV_TNT_MIN_THD_2	O/V	0...1H	0...1	1	[-]
dew point recognition (second monolith)					
EGY_DEW_END_NT	O/V	0...FFFFFFFFH	0... 536870911.875	0.125	[Ws]
Total amount of energy until nox trap dew point detection					
EGY_DEW_NT_STOP	O/V/S	0...FFFFFFFFH	0... 536870911.875	0.125	[Ws]
additional amount of energy for nox trap dew point detection from last engine run without passed dew point					
CTR_COLD_ST_NT	O/V/S	0...FFH	0...255	1	[-]
Number of cold starts for NOx catalyst					
EGY_DEW_END_NT_INT	O/V	0...FFFFFFFFH	0... 536870911.875	0.125	[Ws]
Amount of energy brought into exhaust system for nox trap dew point detection					
MAF_KGH_MIN_EGY_DEW_NT	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
minimum air-mass flow that has to exceeded for incrementing CTR_EGY_DEW_NT					
CTR_EGY_DEW_NT	V	0...FFFFH	0...6553.5	0.1	[-]
Counter for correction of energy threshold for nox trap dew point detection					
FAC_EGY_DEW_END_NT	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
factor for correction of energy threshold for nox trap dew point detection					
FAC_EGY_DEW_NT_STOP	V	0...FFH	-1...1.55	0.01	[-]
factor for correction of energy amount that is stored in NVMY					
RATIO_EGY_DEW_NT	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Ratio of energy brought into exhaust system to energy threshold					

Input data:

LV_ES	TEG_WALL_NT_DOWN_MDL	TEG_WALL_NT_DOWN_MDL_ST	MAF_KGH
LV_ST_END	TCO_ST	T_AST	LV_ST
TEG_PCAT_DOWN[NC_CBK_EX_NR]	TEG_PCAT_DOWN_ADD	LV_CHECK_MDL_ST	EGY_DEW_NT_DOWN
TAM	CAN_R_RATIO_NOX_SENS[NC_NOX_SENS_CONF]	TAM_ST	T_ES_2
CAN_HW_NS[NC_NOX_SENS_CONF]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DLY_DEW_NT	1	0...FFH	0...25.5	0.1	[s]
time delay for nox trap dew point detection					
C_FAC_EGY_DEW_NT_STOP	1	0...FFH	0...0.99609	3.9063e-3	[-]
Weighting factor on restart for nox trap dew point detection					
C_TEG_WALL_NT_HYS	1	0...1FFFH	0...188.393	0.023	[°C]
Reference temperature hysteresis to set nox trap dew point after cooled exhaust pipe again					
C_TEG_WALL_NT_RED	1	0...FFFFH	-48...1459.305	0.023	[°C]
Reference temperature to erase nox trap dew point at cooled exhaust pipe					
ID_MAF_KGH_MIN_EGY_DEW_NT	8	0...FFFFH	0...1023.98437	0.015625	[kg/h]

LDP_TEG_WALL_NT_DOWN_MDL_ID_MAF	8	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
MAF threshold for incrementing CTR_EGY_DEW_END_NT					
IP_EGY_DEW_END_NT	8*8	0...FFFFFFFFH	0... 536870911.875	0.125	[Ws]
LDP_TCO_ST__EGY_DEW_END_NT	8	0...FEH	-48...142.5	0.75	[°C]
LDP_TEG_WALL_NT_DOWN_MDL_ST	8	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Energy threshold for nox trap dew point detection					
IP_EGY_DEW_NT_MAX	8*8	0...FFFFFFFFH	0... 536870911.875	0.125	[Ws]
LDP_TCO_ST__EGY_DEW_NT_MAX	8	0...FEH	-48...142.5	0.75	[°C]
LDP_TAM_ST__EGY_DEW_NT_MAX	8	0...FEH	-48...142.5	0.75	[°C]
Maximum energy that has to be brought into a full flooded exhaust manifold until all water is condensed					
IP_FAC_EGY_DEW_END_NT	8*8	0...7FH	0...1.98437	0.015625	[-]
LDP_CTR_EGY_DEW_NT	8	0...FFFFH	0...6553.5	0.1	[-]
LDP_MAF_KGH_IP_FAC_EGY_DEW_END	8	0...FFFFH	0...2047.96875	0.03125	[kg/h]
factor for correction of energy threshold for nox trap dew point detection					
IP_FAC_EGY_DEW_END_NT_STOP	16*2	0...FFH	-1...1.55	0.01	[-]
LDP_RATIO_EGY_DEW_NT	16	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_TAM__FAC_EGY_DEW_END_NT	2	0...FEH	-48...142.5	0.75	[°C]
factor for correction of energy amount that is stored in NVMY					
IP_FAC_EGY_DEW_END_NT_TAM_T_ES	4*4	0...FFH	-4...3.96875	0.03125	[-]
LDP_T_ES_2__IP_FAC_EGY_DEW_END	4	1...FFFFH	1...65535	1	[s]
LDP_TAM__IP_FAC_EGY_DEW_END	4	0...FEH	-48...142.5	0.75	[°C]
Factor for correction of energy-threshold dependent on TAM and T_ES_2					
IP_R_RATIO_NOX_SENS_MIN_DEW	8*8	0...FFFFH	-32768...32767	1	[-]
LDP_TAM__IP_R_RATIO_NOX_SENS	8	0...FEH	-48...142.5	0.75	[°C]
LDP_NR_CAN_HW_NS	8	0...FFH	0...255	1	[-]
Minimum sensor element temperature depended resistance ratio as 1000 * R/ R25 for dew point detection dependant on NOx sensor hardware version					
IP_TEG_WALL_NT_DOWN_ST	8	0...FFFFH	-48...1459.305	0.023	[°C]
LDP_TAM__TEG_WALL_NT_DOWN_ST	8	0...FEH	-48...142.5	0.75	[°C]
temperature treshold for nox trap dew point detection					
C_EGY_DEW_NT_STOP	1	0...FFFFFFFFH	0... 536870911.875	0.125	[Ws]
Initialization value of energy for nox trap dew point detection from last engine run					

Export actions:

ACTION_EXTD_InitNTEgyStop()
Initialization of energy for nox trap dew point detection from last engine run

Description for ACTION_EXTD_InitNTEgyStop

ACTION_EXTD_InitNTEgyStop()

Initialization of energy for nox trap dew point detection from last engine run

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit

EGY_DEW_NT_STOP = C_EGY_DEW_NT_STOP % initialization via tester service job

31.6.1 Dew point determination

Application conditions:

Activation: LV_ST = 0 and LV_ES = 0

Deactivation: LV_ES = 1

Recurrence: 100 ms

Initialisation:

Initialisation at engine start: transition LV_ES = 1 -> 0

```
LV_TNT_MIN_THD_2      =        0
LV_TNT_MIN_THD_READY_2   = 0
```

Initialisation at engine stop: transition LV_ES = 0 -> 1

```
 LV_TNT_MIN_THD_READY_2   = 0
LV_TNT_MIN_THD_2             = 0      % for a system with a NOx-sensor, the sensor
                                      % heating is activated by LV_TNT_MIN_THD_2.
                                      % Hence for a stopped engine the bit must be
                                      % initialized
                                      % Initialisation is done after calculation of
                                      EGY_DEW_NT_STOP
```

Function description:

On engine start, the calculated manifold wall temperature is compared to the reference value IP_TEG_WALL_TNT_DOWN_ST. If that temperature is higher than the reference value, it is assumed that no condensed water condenses from the exhaust gas. The status regarding reaching the dew point is represented by the bit LV_TNT_MIN_THD_2.
If the manifold wall temperature calculated is lower than the reference value (end of dew point not reached LV_TNT_MIN_THD_2 = 0), an additional amount of heat must be introduced into the exhaust-system branch. If the wall temperature exceeds the reference temperature or the amount of energy introduced into the exhaust system exceeds a certain threshold, end of dew point reached (LV_TNT_MIN_THD_2 = 1).
LV_TNT_MIN_THD_READY_2 is an internal flag to detect the first readyness after function start. To trigger the heater management function depending on low wall temperature (long PUC phase) the dew point detection is reseted below C_TEG_WALL_TNT_RED.
To set the dew point again, the hysteresis C_TEG_WALL_TNT_HYS must be passed.
To ensure, that values for TEG_WALL_NT_DOWN_MDL and TEG_WALL_NT_DOWN_MDL_ST are valid, the dew point detection is delayed by C_DLY_DEW_NT.

Formula section:

End of dew point after start:

```
If(1)       T_AST > C_DLY_DEW_NT
Then (1)
    If(2)        LV_TNT_MIN_THD_READY_2 == 0                    and
                 (TEG_WALL_NT_DOWN_MDL > IP_TEG_WALL_NT_DOWN_ST and
                 EGY_DEW_END_NT_INT ≥ EGY_DEW_END_NT) and
                 CAN_R_RATIO_NOX_SENS ≥ IP_R_RATIO_NOX_SENS_MIN_DEW
    then(2)      LV_TNT_MIN_THD_2  =  1
            if(3a)       TEG_WALL_NT_DOWN_MDL ≥ C_TEG_WALL_NT_RED
```

```
        then(3a)        LV_TNT_MIN_THD_READY_2 =  1
        endif(3a)
    else(2)         // Hysteresis of dew point during engine run
        if(3b)          LV_TNT_MIN_THD_READY_2 == 1
      then(3b)
        if(4)           TEG_WALL_NT_DOWN_MDL  < C_TEG_WALL_NT_RED
        then(4)         LV_TNT_MIN_THD_2   =  0
        else(4)
            if(5)           TEG_WALL_NT_DOWN_MDL >
                            C_TEG_WALL_NT_RED + C_TEG_WALL_NT_HYS
            then(5)         LV_TNT_MIN_THD_2      =  1
            endif(5)
        endif(4)
        else(3b)            LV_TNT_MIN_THD_2   =  0
        endif(4b)
    endif(2)
  endif (1)
```

31.6.2 Calculation of the NOx catalyst monolith temperature (sensor input)

Formula section:

TEG_PCAT_DOWN_COR[NC_CBK_EX_NR]

= TEG_PCAT_DOWN[NC_CBK_EX_NR] +TEG_PCAT_DOWN_ADD

31.6.2.1 Calculation of number of cold starts for Nox catalyst

After leaving the engine state ST the value of CTR_COLD_ST_NT is incremented by the increment one. At the same time dew point state detection is initialized LV_TNT_MIN_THD_2 = 0, i.e. the exhaust gas temperature is less than the temperature that characterizes the dew point.

The value CTR_COLD_ST_NT is a measure of the number of cold-start tests and has to be stored in the flash-memory when the engine is switched off (LV_IGK = 0). After a new start of the engine CTR_COLD_ST_NT is initialised with the stored value.

CTR_COLD_ST_NT is set to zero when the dew point of Temp_Model_2 is passed (LV_TNT_MIN_THD_2 = 1). The initialisation depends on Temp_Model_2 since it takes the longest for the second part of the Nox catalyst monolith to pass the dew-point.

In that way CTR_COLD_ST_NT is incremented for each engine start if the dew point of Temp_Model_2 was not passed before.

Application conditions:

Initialisation: from nonvolatile memory or = 0 if not possible

Recurrence: only once per engine run %CTR_COLD_ST_NT is only incremented per engine run

Activation: LV_ES =0

Formula section:

```
If      LV_TNT_MIN_THD_2 = 0     % at each transition from LV_ES = 1 -> 0
then    CTR_COLD_ST_NT ++        % CTR_COLD_ST_NT is incremented; if there is a
else    CTR_COLD_ST_NT = 0       % a transition of LV_TNT_MIN_THD =  0 -> 1 within the
endif                            % same DC, CTR_COLD_ST_NT has to be set to 0
```

31.6.3 Increasing the required energy

Function description:

On each restart, the amount of condensed water first increases and then decreases until the dew point is reached. To this effect, the energy introduced into the exhaust system EGY_DEW_END_NT_INT is calculated from the start and saved in the non-volatile memory at "engine stop" and after the latching phase has elapsed.
The amount of energy that has been brought into the exhaust system is weighted with FAC_EGY_DEW_NT_STOP, a factor dependent on the ambient temperature and the ratio of EGY_DEW_END_NT_INT and the threshold EGY_DEW_END_NT.
The result of this whole calculation is stored as EGY_DEW_NT_STOP in non volatile memory.
If the dewpoint is passed, EGY_DEW_NT_STOP is set to 0.

Application conditions:

Initialisation: at reset: from nonvolatile memory or = 0 if not possible

Recurrence: only once after transition from LV_ES 0 ->1

Activation: at transition from LV_ES 0->1

Formula section:

RATIO_EGY_DEW_NT = EGY_DEW_END_NT_INT / EGY_DEW_END_NT

FAC_EGY_DEW_NT_STOP = IP_FAC_EGY_DEW_END_NT_STOP

```
If        LV_TNT_MIN_THD_2    = 0
  then if FAC_EGY_DEW_NT_STOP >= 0
      then EGY_DEW_NT_STOP$_{(new)}$  =  EGY_DEW_NT_STOP$_{(last)}$
                         + IP_EGY_DEW_END_NT  * FAC_EGY_DEW_NT_STOP
      else EGY_DEW_NT_STOP$_{(new)}$  =  EGY_DEW_NT_STOP$_{(last)}$
                         + EGY_DEW_NT_STOP$_{(last)}$* FAC_EGY_DEW_NT_STOP
      endif
  else    EGY_DEW_NT_STOP $_{(new)}$  =  0
endif
```

31.6.4 Integration of energy brought into the exhaust system

Function description:

For dew point detection of the NOX-Trap, the energy, that is brought into the exhaust system, has to be integrated.

Application conditions:

Initialisation: EGY_DEW_END_NT_INT = 0 at transition LV_ES 1 -> 0

Recurrence: 200ms

Activation: LV_ST_END= 1 and LV_ES = 0

Deactivation: LV_ES =1

Formula section:

$$EGY\_DEW\_END\_NT\_INT = EGY\_DEW\_END\_NT\_INT_{(n-1)} + EGY\_DEW\_NT\_DOWN * 0,2s$$

31.6.5 Calculation of the total energy until end of dew point

Application conditions:

Activation: LV_CHECK_MDL_ST = 1

Deactivation: LV_ES = 1

Recurrence: 100ms

Initialisation:

Initialisation at engine start: transition LV_ES = 1 -> 0
EGY_DEW_END_NT = FFFFFFFFH
CTR_EGY_DEW_NT = 0

Function description:

To enable detection end of dew point, the necessary energy that has to be introduced into the exhaust system must be calculated first.

The value for the missing amount of energy is obtained from the map IP_EGY_DEW_END_NT. Input for this map is TEG_WALL_NT_DOWN_MDL_ST and TCO_ST.

ID_MAF_KGH_MIN_EGY_DEW_NT is a curve versus the wall temperature TEG_WALL_NT_DOWN_MDL which describes the minimum mass air flow, that has to exceeded to trigger the incrementing of CTR_EGY_DEW_NT.

MAF_KGH and of CTR_EGY_DEW_NT are inputs for IP_FAC_EGY_DEW_END_NT which will provide another factor for decreasing EGY_DEW_END_NT.

Formula section:

MAF_KGH_MIN_EGY_DEW_NT = ID_MAF_KGH_MIN_EGY_DEW_NT

```
If      MAF_KGH > MAF_KGH_MIN_EGY_DEW_NT
Then    CTR_EGY_DEW_NT ++                       % increment by 0.1
Else    CTR_EGY_DEW_NT = 0                      % reset counter
```

FAC_EGY_DEW_END_NT = IP_FAC_EGY_DEW_END_NT
FAC_EGY_DEW_END_NT_TAM_T_ES = IP_FAC_EGY_DEW_END_NT_TAM_T_ES

EGY_DEW_END_NT_TMP = **MIN**{ [(IP_EGY_DEW_END_NT * FAC_EGY_DEW_END_NT_TAM_T_ES + EGY_DEW_NT_STOP)*FAC_EGY_DEW_END_NT] **;** IP_EGY_DEW_NT_MAX * FAC_EGY_DEW_END_NT }

EGY_DEW_END_NT = **MIN**(EGY_DEW_END_NT_TMP **;** $\text{EGY\_DEW\_END\_NT}_{(n-1)}$)

31.7 NOx Catalyst Monolith Temperature Model (Appl. Inc.)

31.7.1 Additive correction of the input temperature TEG_PCAT_DOWN

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TEG_PCAT_DOWN_ADD	V/O	8000...7FFFH	-512...512	0.015625	°C
Temperature offset to correct the influence of the Exhaust flap and temp. sensor position					
FAC_TNT_MIN_COR_n	V/O	0 ... FFH	0 ... 1.99219	0.007812	-
Factor to reduce dew-point time of NOx cat. monolith temp. model					

Input data:

LV_ST	LV_ES		

FUNCTION DESCRIPTION:

For systems containing an exhaust flap a correction of the model input temperature TEG_PCAT_DOWN can be necessary.

Application conditions:

Activation:	- engine state LV_ST = 0
	- engine state LV_ES = 0
Deactivation:	- engine state ES is active (LV_ES = 1)
Recurrence:	100 ms

Formula section:

TEG_PCAT_DOWN_ADD = 0

FAC_TNT_MIN_COR_1 = 1 % default value, no reduction of dew point time
FAC_TNT_MIN_COR_2 = 1 % default value, no reduction of dew point time

31.8 Exhaust Gas Temperature Sensor Diagnosis (TEG)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TEG_PCAT_DOWN	O/V	0...1H	0...1	1	[-]
Temp. after precatalyst error flag					
LV_ERR_PREL_TEG_PCAT_DOWN	O/V	0...1H	0...1	1	[-]
Temp. after precatalyst antibouce flag					
CTR_TQ_MIN_TEG_SENS_DIAG	V	0...FFFFH	0...65535	1	[-]
Counter to control the temperature sensor diagnosis activation and deactivation					
ERR_SYM_TEG_PCAT_DOWN	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of the diagnostic instance TEG_PCAT_DOWN					
LV_END_DIAG_TEG_PCAT_DOWN	O/V	0...1H	0...1	1	[-]
End of diagnostics of the temperature sensor downstream catalyst					
LV_CDN_DIAG_TEG_PCAT_DOWN	O/V	0...1H	0...1	1	[-]
Condition for diagnostic of temperature sensor downstream catalyst					
CTR_SAMPLE_ACT_TEG_SENS_DIAG	V	0...FFFFH	0...65535	1	[-]
Sample counter for continuous temperature sensor diagnosis					
CTR_TQ_MIN_TEG_SENS_DIAG_1	V	0...FFFFH	0...65535	1	[-]
Counter to control the temperature sensor continuous diagnosis activation and deactivation					
RATE_SAMPLE_STC_TEG_SENS_DIAG[32]	V	0...FFFFH	0...65535	1	[-]
Number of samples per calibrated time interval for continuous TEG diagnosis					
CTR_TEG_SENS_DIAG_SUM	V	0...FFFFH	0...65535	1	[-]
Sum counter for wehgting factors for continuous TEG diagnosis					
LV_TEG_SENS_DIAG_RLS	V	0...1H	0...1	1	[-]
Release temperatur sensor diagnosis flag					

Input data:

VP_TEG_PCAT_DOWN	LV_IGK	T_AST_DIAG	VS
TQ_AV	TEG_CAT_DOWN_MDL	TAM	LV_VAR_TCHA
STATE_ENG			

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis shall be to detect electrical faults as defined by OBD I requirements.
VP_TEG_PCAT_DOWN is monitored for invalid gradients and values (see also “exhaust gas temperature ” in chapter “system variables”).

The diagnosis for "short circuit to Vbatt or open load" is only performed:

- either by reaching a certain voltage on the pin or after a calibrateable time after start or counter for activation of diagnosis reached calibrateable value
- and at the moment of diagnosis additional conditions are fullfilled

general specification

This functionality shell prevent a wrong failure detection due to too low sensor temperatures, i.e. at cold start.

Description:

Error-symptoms are defined to this diagnosis function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └> Short circuit to Vbatt or Open load    (= SYM_0)
 │ │ └──> Short circuit to GND                    (= SYM_1)
 │ └────> -                                       (= SYM_2)
 └──────> -                                       (= SYM_3)
```

Application conditions

Initialization: according configuration MEM
(0 at LV_IGK 0->1 or reset)

Recurrence: 100ms

Activation: **see formula section**

Formula section:

```
If(1) LV_IGK = 1
Then(1) CTR_SAMPLE_ACT_TEG_SENS_DIAG++

If(2) TQ_AV> C_TQ_MIN_TEG_SENS_DIAG_1

Then(2) CTR_TQ_MIN_TEG_SENS_DIAG_1++

Endif(2)

If(3) CTR_SAMPLE_ACT_TEG_SENS_DIAG >=
      C_RATE_SAMPLE_TEG_SENS_DIAG_1

Then(3)

    For(3) (x = 31; x = 1; x --)
           { RATE_ SAMPLE _STC_TEG_SENS_DIAG [x] =
           RATE_ SAMPLE _STC_TEG_SENS_DIAG [x-1] }

 RATE_ SAMPLE _STC_TEG_SENS_DIAG [0] = CTR_TQ_MIN_TEG_SENS_DIAG_1
 CTR_TQ_MIN_TEG_SENS_DIAG_1 = 0
 CTR_SAMPLE_ACT_TEG_SENS_DIAG = 0

Endif(3)
```

Short circuit to GND:

```
If(4a)      VP_TEG_PCAT_DOWN  <  C_VP_TEG_PCAT_DOWN_DIAG_MIN
and         LV_VAR_TCHA = 0
Then(4a)    LV_CDN_DIAG_TEG_PCAT_DOWN = 1
            LV_ERR_PREL_TEG_PCAT_DOWN = 1
            ERR_SYM_TEG_PCAT_DOWN = SYM_1

Else(4a)    LV_CDN_DIAG_TEG_PCAT_DOWN = 0
Endif(4a)
```

Short circuit to Vbatt or Open load:

```
If(4b)      LV_TEG_SENS_DIAG_RLS = 0
    and     CTR_TQ_MIN_TEG_SENS_DIAG <                           | calculation of
                        C_CTR_MIN_TQ_MIN_TEG_SENS_DIAG           | counter for
    and     TQ_AV  >=  C_TQ_MIN_TEG_SENS_DIAG                    | activation of
    and     VS       >=  C_VS_MIN_TEG_SENS_DIAG                  | diagnosis
    and     LV_VAR_TCHA = 0
Then(4b)    CTR_TQ_MIN_TEG_SENS_DIAG++
Else(4b)    freeze CTR_TQ_MIN_TEG_SENS_DIAG
Endif(4b)

If(4c)      LV_TEG_SENS_DIAG_RLS = 0
Then(4c)
If(4d)      CTR_TQ_MIN_TEG_SENS_DIAG >=
                       C_CTR_MIN_TQ_MIN_TEG_SENS_DIAG
    or
            (VP_TEG_PCAT_DOWN  <  C_VP_TEG_SENS_ACT_DIAG  and
            T_AST_DIAG  >= C_T_AST_MIN_TEG_SENS_DIAG)
Then(4d)    LV_TEG_SENS_DIAG_RLS = 1
Endif(4d)
Endif(4c)
If(4e)      LV_TEG_SENS_DIAG_RLS = 1
Then4(e)    LV_CDN_DIAG_TEG_PCAT_DOWN = 1
            CTR_TQ_MIN_TEG_SENS_DIAG = 0
            CTR_TEG_SENS_DIAG_SUM = 0
            For(4e)        (x = 1; x <= C_CTR_CYC_TEG_SENS_DIAG_1; x++)
            { CTR_TEG_SENS_DIAG_SUM = CTR_TEG_SENS_DIAG_SUM +
            C_FAC_TEG_SENS_DIAG[x-1] * RATE_ SAMPLE _STC_TEG_SENS_DIAG [x-1] }
    If(5)           VP_TEG_PCAT_DOWN  >  C_VP_TEG_PCAT_DOWN_DIAG_MAX
    Then(5)
            If(6)     (CTR_TEG_SENS_DIAG_SUM > IP_CTR_TQ_MIN_TEG_SENS_DIAG
            and       STATE_ENG != PUC
            and       RATE_ SAMPLE _STC_TEG_SENS_DIAG [0] >
```

```
                C_RATE_SAMPLE_MIN_TEG_SENS_DIAG)
            or
                TEG_CAT_DOWN_MDL[1] > C_ TEMP_THD _TEG_SENS_DIAG_1

            Then(6)     LV_ERR_PREL_TEG_PCAT_DOWN = 1
                        ERR_SYM_TEG_PCAT_DOWN = SYM_0
                        LV_ERR_TEG_PCAT_DOWN = 1
            Endif(6)
        Else(5)         ERR_SYM_TEG_PCAT_DOWN = NO_SYM
                        LV_ERR_PREL_TEG_PCAT_DOWN = 0
        Endif(5)
    Else(4e)    LV_CDN_DIAG_TEG_PCAT_DOWN = 0
    Endif(4e)

    Calculation of the gradient control
    If (4f)     I VP_TEG_PCAT_DOWN(n) - VP_TEG_PCAT_DOWN(n-1) I > C_VP_TEG_MES_GRD_MAX
    Then (4f)   LV_ERR_PREL_TEG_PCAT_DOWN = 1
    Endif (4f)

    Endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VP_TEG_PCAT_DOWN_DIAG_MIN	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Minimum voltage threshold for error detection					
C_VP_TEG_PCAT_DOWN_DIAG_MAX	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Maximum voltage threshold for error detection					
C_VP_TEG_MES_GRD_MAX	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Maximum voltage gradient threshold for error detection					
C_ABC_INC_TEG_PCAT_DOWN	1	0...FFH	0...255	1	[-]
Debounce counter increment ¿ TEG Diagnosis					
C_ABC_MAX_TEG_PCAT_DOWN	1	1...FFH	1...255	1	[-]
Debounce counter maximum ¿ TEG Diagnosis					
C_CTR_MIN_TQ_MIN_TEG_SENS_DIAG	1	0...FFFFH	0...65535	1	[-]
Minimum counter value for temperature sensor diagnosis dependent on TQ					
C_TQ_MIN_TEG_SENS_DIAG	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
calibratable threshold as minimum requested torque for activating diagnosis of exhaust gas temperature sensors					
C_VS_MIN_TEG_SENS_DIAG	1	0...FFH	0...255	1	[km/h]
Vehicle speed threshold for activating diagnosis of exhaust gas temperature sensors					
C_VP_TEG_SENS_ACT_DIAG	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Exhaust gas temperature sensor signal voltage threshold for activation of diagnosis of TEG sensors					
C_T_AST_MIN_TEG_SENS_DIAG	1	0...FFFFH	0...32767.5	0.5	[s]
Minimum time delay after start for exhaust gas temperature sensor diagnosis					
C_TQ_MIN_TEG_SENS_DIAG_1	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Min. requested torque for counter activating for continuous TEG diagnosis					
C_CTR_CYC_TEG_SENS_DIAG_1	1	0...20H	0...32	1	[-]
Number of cycles for sum calculating for continuous TEG diagnosis					
C_RATE_SAMPLE_TEG_SENS_DIAG_1	1	0...400H	0...10240	10	[-]
Number of samples for save cycle for continuous TEG diagnosis					
C_TEMP_THD_TEG_SENS_DIAG_1	1	0...400H	0...1024	1	[°C]
Exhaust gas temperature threshold for activation of TEG sensor diagnosis					
C_FAC_TEG_SENS_DIAG[32]	1	0...80H	0...1	0.0078125	[-]
Array for wehgting factor for continuous TEG diagnosis					
IP_CTR_TQ_MIN_TEG_SENS_DIAG	8	0...FFFFH	0...65535	1	[-]
LDP_TAM_IP_CTR_TQ_MIN_TEG_SENS	8	0...FEH	-48...142.5	0.75	[°C]
Max value from sum. counter for TQ_AV check for continuous TEG diagnosis					
C_RATE_SAMPLE_MIN_TEG_SENS_DIAG	1	0...FFFFH	0...65535	1	[-]
Min number of samples per calibrated time interval for continuous TEG sensor diagnosis					

31.9 Cus adap module: EXTD

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Tkat_mod	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
modelled catalyst temperature					
Tabg_sens1	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Signal exhaust gas temperature sensor 1					
Tabg_sens2	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Signal exhaust gas temperature sensor 2					
B_atm_tpvk1	V/O	0...1H	0...1	1	[-]
Bedingung Taupunktende erreicht Lambdasonde vor Kat1					
B_atm_tpvk2	V/O	0...1H	0...1	1	[-]
Bedingung Taupunktende erreicht Lambdasonde vor Kat2					
B_atm_tpnk1	V/O	0...1H	0...1	1	[-]
Dew point has been reached post cat. specific bank 1 downstream					
B_atm_tpnk2	V/O	0...1H	0...1	1	[-]
Dew point has been reached post cat. specific bank 2 downstream					
EGY_DEW_NT_DOWN	O/V	8000...7FFFH	-20480... 20479.375	0.625	[W]
amount of energy brought into exhaust system for nox trap dew point detection					
LV_CHECK_MDL_ST	O/V	0...1H	0...1	1	[-]
Flag for modelling start					
TEG_CAT_DOWN_MDL	O/V	0...7FF0H	0...2047	0.0625	[°C]
Bank selektive modelled exhaust gas temperature downstream for interfaces					
TEG_CAT_DOWN_MDL_BAS[NC_CBK_EX_NR]	O/V	0...FFFFH	-48...1459.305	0.023	[°C]
Model of exhaust temperature post catalyst downstream					
TEG_CAT_DOWN_MDL_i	O	0...7FF0H	0...2047	0.0625	[°C]
Exhaust gas temperature downstream Cat					
TEG_CAT_UP_MDL_i	O	0...7FF0H	0...2047	0.0625	[°C]
Bank selektive modelled exhaust gas temperature upstream for interfaces					
TEG_CAT_UP_MDL[NC_CBK_EX_NR]	O/V	0...7FF0H	0...2047	0.0625	[°C]
Modelled exhaust gas temperature; engine out (bank selective)					
TEG_DYN	O/V	0...7FF0H	0...2047	0.0625	[°C]
Modelled exhaust gas temperature upstream for interfaces					
TEG_STAT_MAX	O/V	0...7FF0H	0...2047	0.0625	[°C]
Gemittelte max. Temperatur für Desulfatisierung					
TEG_STAT_MAX_1	O/V	0...7FF0H	0...2047	0.0625	[°C]
Max. Temperatur für Desulfatisierung Bank 1					
TEG_STAT_MAX_2	O/V	0...7FF0H	0...2047	0.0625	[°C]
Max. Temperatur für Desulfatisierung Bank 2					
TEG_WALL_CAT_DOWN_MDL[NC_CBK_EX_NR]	O/V	0...FFFFH	-48...1459.305	0.023	[°C]
Model of exhaust wall temperature post catalyst downstream					
TEG_WALL_CAT_UP_MDL[NC_CBK_EX_NR]	O/V	0...FFFFH	-48...1459.305	0.023	[°C]
Model of exhaust wall temperature pre catalyst upstream					
TEG_WALL_NT_DOWN_MDL	O/V	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
exhaust manifold wall temperature at NOX-sensor					
TEG_WALL_NT_DOWN_MDL_ST	O/V	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
exhaust manifold wall temperature at NOX-sensor at start					
TEMP_CAT	O/V	0...FFFFH	-33...990.98437	0.015625	[°C]
Modelled exhaust gas temperature downstream for interfaces					
TEMP_CAT_DYN_MDL[NC_CBK_EX_NR]	O/V	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
Modelled catalyst temperature under dynamic conditions					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TEMP_CAT_STAT_MDL[NC_CBK_EX_NR]	O/V	0...7FFFH	-273.15... 1774.7875	0.0625	[°C]
modeled catalyst monolith temperature (bank selective)					
TNT_MDL_1[NC_NT_NR]	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Modelled NOx catalyst temperature of the first monolith					
TNT_MDL_2[NC_NT_NR]	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Modelled NOx catalyst temperature of the second monolith					
TNT_MDL_H	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Maximum of modelled NOx catalyst monolith temperatures					
TNT_MDL_H_1	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Maximum of modelled NOx catalyst monolith temperatures, bank 1					
TNT_MDL_H_2	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Maximum of modelled NOx catalyst monolith temperatures, bank 2					
TNT_MDL_L	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Minimum of modelled NOx catalyst monolith temperatures					
TNT_MDL_MAX_AVL	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Maximum attainable monolith temp. for active Nox cat. desulfation					
TNT_MDL_MV	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Temperature NoX-trap mean value					
TNT_MDL_MV_SNG[NC_NT_NR]	O/V	0...FFFFH	0...1023.98437	0.015625	[°C]
Mean value of the two modelled NOx catalyst monolith temperatures for each exhaust branch					

Input data:

LV_ST_END	LV_TEG_CAT_DOWN_MIN_THD[NC_CBK_EX_NR]	LV_TEG_MIN_THD[NC_CBK_EX_NR]	NC_CBK_EX_NR
NC_NT_NR	Q_abgnox	T_ikat1	T_ikat1_stat
T_ikat2	T_ikat2_stat	Tabg_demax1	Tabg_demax2
Tabg_demaxav	Tabg_desul	Tabg_ikat_av	Tabg_inok1
Tabg_inok2	Tabg_nk_av	Tabg_nk1	Tabg_nk2
Tabg_nnok1	Tabg_nnok2	Tabg_nok_av	Tabg_nok_av1
Tabg_nok_av2	Tabg_nokmax	Tabg_nokmax1	Tabg_nokmax2
Tabg_nokmin	Tabg_r10	Tabg_r9	Tabg_vk_av
Tabg_vk1	Tabg_vk2	TEG_PCAT_DOWN_COR[NC_CBK_EX_NR]	Tw_m1
Tw_m2	Tw_nk1	Tw_nk2	Tw_noxsens
LV_STST_STOP_CYC_MEM			

31.9.1 Output for BMW functions which are not defined as EXTD exported data

FUNCTION DESCRIPTION:

Remark: all formulas are valid in a **physical** meaning

Application conditions:

Initialisation at reset or at exit power latch phase for variables with BMW-naming convention:

0 °C:	Tkat_mod
0:	B_atm_tpnk1, B_atm_tpnk2, B_atm_tpvk1, B_atm_tpvk2
First valid value:	Tabg_sens1, Tabg_sens2, Tabg_sensturb1, Tabg_sensturb2

Recurrence: 200ms

Activation: at every engine state

Deactivation: ---

Formula section:

```
B_atm_tpvk1  =  LV_TEG_MIN_THD[1]
B_atm_tpvk2  =  LV_TEG_MIN_THD[2]
B_atm_tpnk1     =         LV_TEG_CAT_DOWN_MIN_THD[1]
B_atm_tpnk2     =         LV_TEG_CAT_DOWN_MIN_THD[2]

If ECU_STATE is not"PWL"
then
   Tkat_mod    = TEG_CAT_DOWN_MDL
   Tabg_sens1     =         TEG_PCAT_DOWN_COR[1]
     If NC_NT_NR = 2
       then
          Tabg_sens2 = TEG_PCAT_DOWN_COR[2]
       else
          Tabg_sens2 = TEG_PCAT_DOWN_COR[1]

Endif
```

31.9.2 Outputs for SV aggregates

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes.

Application conditions:

Initialisation for variables with Siemens-naming convention: 0 °C

Except:

EGY_DEW_NT_DOWN = Q_abgnox

TEG_CAT_DOWN_MDL_BAS[1] = Tabg_r9

TEG_CAT_DOWN_MDL_BAS[2] = Tabg_r10

TEG_CAT_UP_MDL[1] = Tabg_vk1

TEG_CAT_UP_MDL[2] = Tabg_vk2

TEG_CAT_UP_MDL_1 = Tabg_vk1

TEG_CAT_UP_MDL_2 = Tabg_vk2

TEG_STAT_MAX_1 = Tabg_demax1

TEG_STAT_MAX_2 = Tabg_demax2

TEG_STAT_MAX = Tabg_demaxav

TEMP_CAT = Tabg_ikat_av

TEG_CAT_DOWN_MDL[1] = Tabg_nk1

TEG_CAT_DOWN_MDL[2] = Tabg_nk2

TEG_CAT_DOWN_MDL_1 = Tabg_nk1

TEG_CAT_DOWN_MDL_2 = Tabg_nk2

TEG_CAT_DOWN_MDL = Tabg_nk_av

TEG_WALL_NT_DOWN_MDL_ST = Tw_noxsens

TEG_WALL_NT_DOWN_MDL = Tw_noxsens

TEG_DYN = Tabg_vk_av

TEMP_CAT_DYN_MDL[1] = T_ikat1

TEMP_CAT_DYN_MDL[2] = T_ikat2

TEMP_CAT_STAT_MDL[1] = T_ikat1_stat

TEMP_CAT_STAT_MDL[2] = T_ikat2_stat

TNT_MDL_MAX_AVL = Tabg_desul

TNT_MDL_1[1] =Tabg_inok1

TNT_MDL_1[2] =Tabg_inok2

TNT_MDL_2[1] =Tabg_nnok1

TNT_MDL_2[2] =Tabg_nnok2

TNT_MDL_MV =Tabg_nok_av

TNT_MDL_MV_SNG[1]=Tabg_nok_av1

TNT_MDL_MV_SNG[2]=Tabg_nok_av2

TNT_MDL_H =Tabg_nokmax

TNT_MDL_H_1 =Tabg_nokmax1

TNT_MDL_H_2 =Tabg_nokmax2

TNT_MDL_L =Tabg_nokmin

Recurrence : 200 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

EGY_DEW_NT_DOWN = Q_abgnox

TEG_WALL_CAT_UP_MDL[1] = Tw_m1

TEG_WALL_CAT_UP_MDL[2] = Tw_m2

TEG_WALL_CAT_DOWN_MDL[1] = Tw_nk1

TEG_WALL_CAT_DOWN_MDL[2] = Tw_nk2

TEG_CAT_DOWN_MDL_BAS[1] = Tabg_r9

TEG_CAT_DOWN_MDL_BAS[2] = Tabg_r10

TEG_CAT_UP_MDL[1] = Tabg_vk1

TEG_CAT_UP_MDL[2] = Tabg_vk2

TEG_STAT_MAX_1 = Tabg_demax1

TEG_STAT_MAX_2 = Tabg_demax2

TEG_STAT_MAX = Tabg_demaxav

TEMP_CAT = Tabg_ikat_av

TEG_CAT_DOWN_MDL[1] = Tabg_nk1

TEG_CAT_DOWN_MDL[2] = Tabg_nk2

TEG_CAT_DOWN_MDL = Tabg_nk_av

TEG_WALL_NT_DOWN_MDL = Tw_noxsens

TEG_DYN = Tabg_vk_av

TEMP_CAT_DYN_MDL[1] = T_ikat1

TEMP_CAT_DYN_MDL[2] = T_ikat2

TEMP_CAT_STAT_MDL[1] = T_ikat1_stat

TEMP_CAT_STAT_MDL[2] = T_ikat2_stat

TNT_MDL_MAX_AVL = Tabg_desul

TNT_MDL_1[1] =Tabg_inok1

TNT_MDL_1[2] =Tabg_inok2

TNT_MDL_2[1] =Tabg_nnok1

TNT_MDL_2[2] =Tabg_nnok2

TNT_MDL_MV =Tabg_nok_av

TNT_MDL_MV_SNG[1]=Tabg_nok_av1

```
TNT_MDL_MV_SNG[2]=Tabg_nok_av2
TNT_MDL_H       =Tabg_nokmax
TNT_MDL_H_1     =Tabg_nokmax1
TNT_MDL_H_2     =Tabg_nokmax2
TNT_MDL_L       =Tabg_nokmin

If(1) LV_ES = 1
Then(1)
    LV_CHECK_MDL_ST = 0
Else(1)
    If(2)   LV_CHECK_MDL_ST = 0
      And   TEG_WALL_NT_DOWN_MDL(n-1) <> TEG_WALL_NT_DOWN_MDL(n)
    Then(2)
            LV_CHECK_MDL_ST = 1
            If      LV_STST_STOP_CYC_MEM = 0
            Then    TEG_WALL_NT_DOWN_MDL_ST = TEG_WALL_NT_DOWN_MDL
            Endif
    Endif(2)
Endif(1)
```

32 Function management

Chapter Function management	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 5435 of 9643

Table of Contents

Index

32.1 Total running time

Output data:

Name	Mode	Hex. Limit	Phys. Limit	Resol.	Unit
TRT	V/O/S	0...FFFFFFFFH	0...119304.64708	2.78E-05	[h]
Total running time					

Input data:

LV_IGK	LV_ES		

Function description:

The total running time is incremented by 100ms each if the conditions "ignition key ON" and "engine running" has been detected.

The total running time is saved in the non-volatile memory at the end of the driving cycle.

Formula section:

IF LV_IGK = 1 UND LV_ES = 0

THEN TRT = TRT + 2,7778e-5h

ELSE TRT = TRT

32.2 ECU power latch management - power down

Export actions:

ACTION_FCTM_SetPwlLockCdn (IN <Param1>)
Set related flag at pwl_lock_cdn
ACTION_FCTM_ClrPwlLockCdn (IN <Param1>)
Clear related flag at pwl_lock_cdn

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PWL_LOCK_CDN	V/O	0...FFFFH	0...65535	1	[-]
Power latch extension by project-specific configuration:see configuration-values in bitmask NC_PWL_LOCK_CDN_XXX					
T_PWL	V/O	0...FFFFH	0...6553.5	0.1	[s]
incrementing timer during PWL phase active					
LV_INH_PWL_TRAN_ES_EL	V/O	0...1H	0...1	1	[-]
flag active in shutdown-phase: IGK on is not accepted without reset					
LV_PWL_ACT	V/O	0...1H	0...1	1	[-]
power latch phase active (Do not use as Output)					

Input data:

NC_PWL_LOCK_CDN_XXX	STATE_LRN_ECU	LV_DIAG_END_RLY_MAIN_DLY	

FUNCTION DESCRIPTION:

Initialisations:

T_PWL = 0 at RST **or** EXIT_PWL

LV_PWL_ACT = 0

LV_PWL_ACT = 1 at TO_PWL

Recurrence: 100 ms

Activation: LV_PWL_ACT = 1

General information:

The relevant informations for setting PWL_LOCK_CDN are transferred with the help of the action call.

The ECU keeps active for a minimum time after entering the powerlatch-phase.
The minimum ECU power latch time (C_T_MIN_PWL) can only be extended by other functions locking a PWL-resource using the bitword PWL_LOCK_CDN (see project specifc list of PWL configuration data (definition of NC_PWL_LOCK_CDN_XXX)). If the maximum powerlatch time C_T_PWL_MAX is reached the ECU will be shut down and all not saved data will be lost.

Bit-position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWL_LOCK_CDN																

Each other function is responsible itself to lock / unlock its own PWL-resource at entering / leaving the powerlatch-phase !

Formula section:

```
T_PWL = T_PWL + 100ms

If      (T_PWL > C_T_MIN_PWL                    and
        PWL_LOCK_CDN = 0)
Or      (T_PWL > C_T_MAX_PWL                    and
        at PWL_LOCK_CDN no NVMY Bit is set)
Or      LV_INH_PWL_TRAN_ES_EL = 1 (already in shut down)
Then    activate shut-down mechanism
Else    do nothing
Endif
```

Actions for shut-down mechanism:

reactivation by switching IGK on is forbidden

- LV_INH_PWL_TRAN_ES_EL = 1
- disable Monitoring-unit (MU)
- wait until MU-disable is confirmed **or** Communication with MU failed **or** LV_DIAG_END_RLY_MAIN_DLY = 0
- all function specific initializations with “end of power latch phase” are done.
- The power-supply-pin of the ECU is switched off.
- Call 5V reset detection.

Requirements for ACTION_FCTM_SetPwlLockCdn(<param>):

param: NC_PWL_LOCK_CDN_xxx (see Appl. INCI)

Set related flag at pwl_lock_cdn

pwl_lock_cdn |= 1 shifted left by <param>

Requirements for ACTION_FCTM_ClrPwlLockCdn (IN <Param1>)

param: NC_PWL_LOCK_CDN_xxx (see Appl. INCI)

Clear related flag at pwl_lock_cdn

pwl_lock_cdn &= not (1 shifted left by <param>)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MIN_PWL	1	0...FFFFH	0...6553.5	0.1	[s]
minimum power-latch time for ECU switch off					
C_T_MAX_PWL	1	0...FFFFH	0...6553.5	0.1	[s]
maximum power-latch time for ECU switch off (if reached => not saved data will be lost)					

32.2.1 Configuration of PWL-resources

General information:

The bitmask PWL LOCK CDN shows all functions by a defined bit-position, which has locked a PWL resource for PWL extension.

Bit-position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWL_LOCK_CDN																

It’s also possible, that more than one function has locked the PWL.
Here all functions must unlock its request before the ECU is switched off.

Bitmask-name	Bitposition	PWL-lock-configuration-Description
NC_PWL_LOCK_CDN_XXX	0...15	
Values for project		
NC_PWL_LOCK_CDN_NVMY	0	PWL for NVMY storage
NC_PWL_LOCK_CDN_RLY_MAIN	1	PWL for main relay diagnosis
NC_PWL_LOCK_CDN_EGR_AD	2	PWL for EGR adaptation
NC_PWL_LOCK_CDN_BN2000	3	PWL for BN2000
NC_PWL_LOCK_CDN_DMTL	4	PWL for DMTL diagnosis
NC_PWL_LOCK_CDN_ROM_CHK	5	PWL for ROM check
NC_PWL_LOCK_CDN_CUS	6	PWL for CUS functions required
NC_PWL_LOCK_CDN_HPDI	7	PWL for HDPI (shaker function)
project-specific configuration of PWL resources, which can be locked		

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PWL_LOCK_CDN_NVMY	1	0...FFH	0...255	1	[-]
PWL for NVMY storage					
NC_PWL_LOCK_CDN_RLY_MAIN	1	0...FFH	0...255	1	[-]
PWL for main relay diagnosis					
NC_PWL_LOCK_CDN_EGR_AD	1	0...FFH	0...255	1	[-]
PWL for EGR adaptation					
NC_PWL_LOCK_CDN_BN2000	1	0...FFH	0...255	1	[-]
PWL for BN2000					
NC_PWL_LOCK_CDN_DMTL	1	0...FFH	0...255	1	[-]
PWL for DMTL diagnosis					
NC_PWL_LOCK_CDN_ROM_CHK	1	0...FFH	0...255	1	[-]
PWL for ROM check					
NC_PWL_LOCK_CDN_CUS	1	0...FFH	0...255	1	[-]
PWL for CUS functions required					
NC_PWL_LOCK_CDN_HPDI	1	0...FFH	0...255	1	[-]
PWL for HPDI					

33 Fuel mass setpoint

Chapter Fuel mass setpoint	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 5444 of 9643

Table of Contents

Index

33.1 Administration of calculation optimization

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CLC_RED	V/O	0 ... FFH	0 ... 255	1	-
Calculation reduction state for runtime optimization					
CTR_CYL_NR_ST_CLC	V/O	0 ... 7H	0 ... 7	1	-
Number of first cylinder in calculation order					
CTR_CYL_NR_STOP_CLC	V/O	0 ... 7H	0 ... 7	1	-
Number of last cylinder in calculation order					
CTR_CBK_IN_NR_ST_CLC	V/O	0 ... 1H	1 ... 2	1	-
Start number of intake bank for calculations					
CTR_CBK_IN_NR_STOP_CLC	V/O	0 ... 1H	1 ... 2	1	-
Stop number of intake bank for calculations					
CTR_CBK_EX_NR_ST_CLC	V/O	0 ... 1H	1 ... 2	1	-
Start number of exhaust bank for calculations					
CTR_CBK_EX_NR_STOP_CLC	V/O	0 ... 1H	1 ... 2	1	-
Stop number of exhaust bank for calculations					

Input data:

NC_CYL_NR	LV_ST_END	N_32	SEG_NR
NC_IN_REF	NC_LAMB_REF		

FUNCTION DESCRIPTION:

General information:

This module is used for administration of runtime reduction and calculation optimization steps.

Pay Attention! This module has to be placed in the operating system immediately after the calculation of N_32 and SEG_NR, which are calculated by the ENSD aggregate. All aggregates, which are affected by calculation optimization, have to be calculated afterwards.

33.1.1 Runtime reduction

Application conditions :

Activation : every engine state

Deactivation : -

Initialization: at reset: STATE_CLC_RED = NC_STATE_CLC_RED_SEG_1

CTR_CYL_NR_ST_CLC = 0

CTR_CYL_NR_STOP_CLC = (NC_CYL_NR - 1)

CTR_CBK_IN_NR_ST_CLC = 1 (physical meaning)

CTR_CBK_IN_NR_STOP_CLC = NC_CBK_IN_NR

CTR_CBK_EX_NR_ST_CLC = 1 (physical meaning)

CTR_CBK_EX_NR_STOP_CLC = NC_CBK_EX_NR

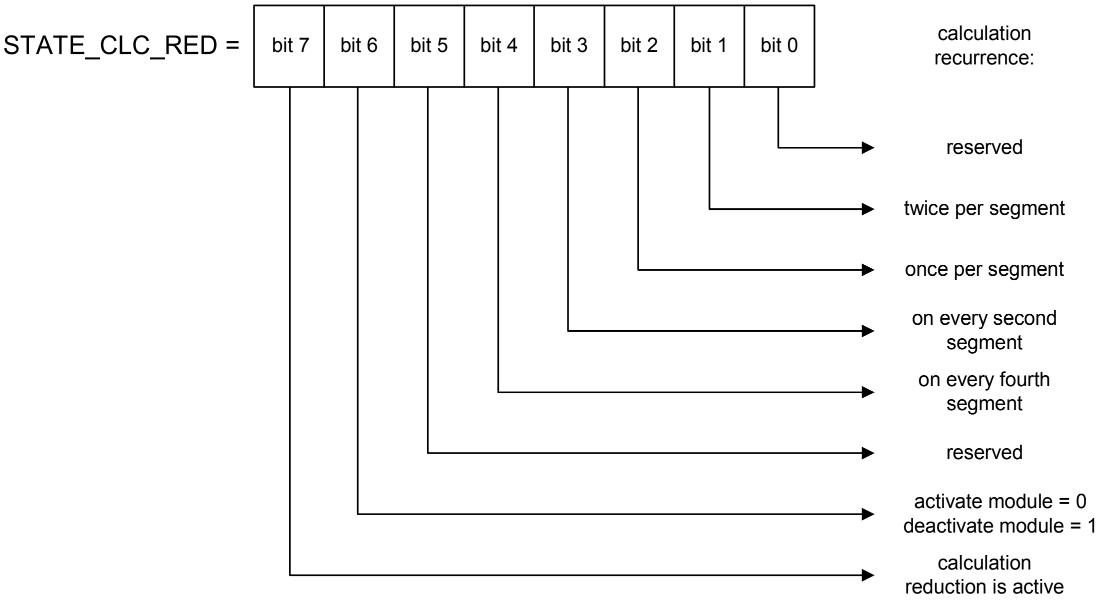
Recurrence : LV_ST_END = 0: 10 ms

LV_ST_END = 1: segment synchronous

This chapter describes runtime reduction steps. First step is to reduce cylinder dependent calculations from all cylinders to one cylinder, dependent on the engine speed. An additional step is to reduce bank dependent calculations to the intake or exhaust bank, which is allocated to the related cylinder by NC_IN_REF and NC_LAMB_REF.

Formula Section :

33.1.1.1 Definition of STATE_CLC_RED



33.1.1.1.1 Definitions of the non calibrateable constants to generate STATE_CLC_RED

NC_STATE_CLC_RED_SEG_05 = 0x02H

NC_STATE_CLC_RED_SEG_1 = 0x04H

NC_STATE_CLC_RED_SEG_2 = 0x08H

NC_STATE_CLC_RED_SEG_4 = 0x10H

NC_STATE_CLC_RED_CLC_DEAC = 0x40H

NC_STATE_CLC_RED_ACT = 0x80H

NC_STATE_CLC_RED_MASK_1 = 0x1EH

NC_STATE_CLC_RED_MASK_2 = 0xC0H

33.1.1.2 Check runtime reduction activation conditions

```
(1) IF  (STATE_CLC_RED AND NC_STATE_CLC_RED_ACT) = 0            ... bitwise
        AND
        N_32 > C_N_32_THD_CLC_RED
        AND
        SEG_NR = C_SEG_NR_ST_CLC_RED
(1) THEN   (runtime reduction has to be activated)
        STATE_CLC_RED =        NC_STATE_CLC_RED_ACT
                          OR   NC_STATE_CLC_RED_SEG_2            ... bitwise


(1) ELSE
        (2) IF (STATE_CLC_RED AND NC_STATE_CLC_RED_ACT) =                 ... bitwise
               NC_STATE_CLC_RED_ACT
               AND
               N_32 < (C_N_32_THD_CLC_RED - C_N_32_HYS_CLC_RED)
               AND
               SEG_NR = C_SEG_NR_ST_CLC_RED
        (2) THEN
               STATE_CLC_RED = NC_STATE_CLC_RED_SEG_1
               (runtime reduction will be deactived)
        (2) ENDIF
(1) ENDIF
```

33.1.1.3 Determine cylinder and bank numbers for follow up calculations

```
(3) IF  (STATE_CLC_RED AND NC_STATE_CLC_RED_ACT) =                        ... bitwise
        NC_STATE_CLC_RED_ACT
(3) THEN
        STATE_CLC_RED =          STATE_CLC_RED
                          ExOR   NC_STATE_CLC_RED_CLC_DEAC                ... bitwise

        CTR_CYL_NR_ST_CLC = ID_CTR_CYL_NR_CLC
        (SEG_NR is input data for ID_CTR_CYL_NR_CLC)
        CTR_CYL_NR_STOP_CLC = CTR_CYL_NR_ST_CLC
        (4) IF  Bit related to cylinder CTR_CYL_NR_ST_CLC of NC_IN_REF = 1
        (4) THEN
               CTR_CBK_IN_NR_ST_CLC = 2              (physical meaning)
               CTR_CBK_IN_NR_STOP_CLC = 2            (physical meaning)
```

```
        (4) ELSE
                CTR_CBK_IN_NR_ST_CLC = 1                  (physical meaning)
                CTR_CBK_IN_NR_STOP_CLC = 1                (physical meaning)
        (4) ENDIF
        (5) IF  Bit related to cylinder CTR_CYL_NR_ST_CLC of NC_LAMB_REF = 1
        (5) THEN
                CTR_CBK_EX_NR_ST_CLC = 2                  (physical meaning)
                CTR_CBK_EX_NR_STOP_CLC = 2                (physical meaning)
        (5) ELSE
                CTR_CBK_EX_NR_ST_CLC = 1                  (physical meaning)
                CTR_CBK_EX_NR_STOP_CLC = 1                (physical meaning)
        (5) ENDIF
    (3) ELSE
        CTR_CYL_NR_ST_CLC = 0
        CTR_CYL_NR_STOP_CLC = (NC_CYL_NR - 1)
        CTR_CBK_IN_NR_ST_CLC = 1                          (physical meaning)
        CTR_CBK_IN_NR_STOP_CLC = NC_CBK_IN_NR             (physical meaning)
        CTR_CBK_EX_NR_ST_CLC = 1                          (physical meaning)
        CTR_CBK_EX_NR_STOP_CLC = NC_CBK_EX_NR             (physical meaning)
    (3) ENDIF
```

Application hint

ID_CTR_CYL_NR_CLC describe the cylinder number, which will be calculated exclusive on the segment with the number SEG_NR, even if the runtime reduction is active. Therefor you have to specify the segment cylinder relation here proper to your engine type. For example (8 cyl. engine):

SEG_NR Segment number	ID_CTR_CYL_NR_CLC Exclusive calculated cylinder
0	3
1	4
2	5
3	6
4	7
5	0
6	1
7	2

Note that the related intake and exhaust banks will be calculated automatically by NC_IN_REF and NC_LAMB_REF.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_CTR_CYL_NR_CLC	NC_CYL_NR	0...7H	0...7	1	-
LDP_SEG_NR_ID_CTR_CYL_NR_CLC	NC_CYL_NR	0...7H	0...7	1	-
Cylinder and segment relation for runtime reduction					
C_N_32_THD_CLC_RED	1	0...FFH	0...8160	32	rpm
Engine speed threshold for activation of runtime reduction					
C_N_32_HYS_CLC_RED	1	0...3FH	0...2016	32	rpm
Engine speed hysteresis for deactivation of runtime reduction					
C_SEG_NR_ST_CLC_RED	1	0...7H	0...7	1	-
Start segment number for runtime reduction activation					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_STATE_CLC_RED_SEG_05	1	0 ... FFH	0 ..255	1	-
Constant defined to indicate that the calculation of a module is done twice per segment					
NC_STATE_CLC_RED_SEG_1	1	0 ... FFH	0 ..255	1	-
Constant defined to indicate that the calculation of a module is done on every segment					
NC_STATE_CLC_RED_SEG_2	1	0 ... FFH	0 ..255	1	-
Constant defined to indicate that the calculation of a module is done on every second segment					
NC_STATE_CLC_RED_SEG_4	1	0 ... FFH	0 ..255	1	-
Constant defined to indicate that the calculation of a module is done on every fourth segment					
NC_STATE_CLC_RED_CLC_DEAC	1	0 ... FFH	0 ..255	1	-
Constant defined to indicate that the calculation of a module has not been done at next activation call					
NC_STATE_CLC_RED_ACT	1	0 ... FFH	0 ..255	1	-
Constant defined to indicate that calculation reduction is active					
NC_STATE_CLC_RED_MASK_1	1	0 ... FFH	0 ..255	1	-
Mask to isolate calculation recurrence information					
NC_STATE_CLC_RED_MASK_2	1	0 ... FFH	0 ..255	1	-
Mask to isolate activation information					

33.2 Run Time Optimization by Recurrency Switch

Output Data:

Name	Mode	Hex. Limit	Phys. Limit	Resol.	Unit
LV_CLC_2SEG_ENA	V/O	0...1H	0...1	1	-
activation flag for correction modes for non segment synchronuously calculation					
LV_CLC_2SEG	V/O	0...1H	0...1	1	-
Toggle-Bit for activation/deactivation of the segment synchronuously calculation					

Input Data:

N_32			

FUNCTION DESCRIPTION:

Application conditions:

Recurrence: every segment

Initialisation: LV_CLC_2SEG = 0; LV_CLC_2SEG_ENA = 0

Activation: every other engine state

Deactivation: engine stopped or engine start detected: LV_ES = 1 or LV_ST = 1

General Information:

At a high engine speed, run time problems may occur. Overstepping a certain engine speed threshold, the Toggle-Bit created in this module can evoke modules calculated segment synchronuously to switch to a recurrency of every second segment.

Formula section:

Activation of Toggle-Bit:

As soon as the engine speed exceeds a threshold C_N_SEG_2_ENA, the Toggle-Bit LV_CLC_2SEG gets activated and changes its value from zero to one the next segment and back to zero again the subsequent segment.

Activation of Correction-Bit:

At the same time, the bit LV_CLC_2SEG_ENA is set to one. This bit enables calculation corrections in files that use a segment counter or have to be corrected by any other influences caused by a calculation recurrence only every second segment.

Deactivation:

The deactivation of both bits is done as soon as the engine speed threshold is underspent again.

```
if    N_32  >    C_N_SEG_2_ENA
then        LV_CLC_2SEG_ENA      =    1
            LV_CLC_2SEG          =    0 → 1→ 0 → 1 → 0 → 1 → 0 → ...(toggling)
else        LV_CLC_2SEG_ENA = 0
            LV_CLC_2SEG  = 0
endif
```

Calibration Data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
C_N_SEG_2_ENA	1	0...FFH	0...8160	32	1/min
engine speed threshold for the activation of the calculation recurrence change to every second segment					

33.3 Restart function (FAC_ST_REST)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_ST_REST	O/V	0...FFH	0...1.9921875	0.0078125	-
Restart-factor for the correction of mass fuel flow at start					

Input data:

TCO	LV_REST	TCO_STOP	T_ES_2
T_AST_REST	LV_STALL	LV_T_REL_CAN_REG	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_STALL	8	0...FFH	0...0.99609307	0.00390625	-
LDPM_TCO__FAC_STALL	8	0...FEH	-48...142.5	0.75	°C
Special cold start reduction factor in case of re-start after engine stall out					
IP_FAC_STND_REST	8	0...FFH	0...0.99609307	0.00390625	-
LDPM_TCO__TI_ST_REST	8	0...FEH	-48...142.5	0.75	°C
Cold start reduction factor in case of re-start with no engine stop time T_ES available					
IP_REST	10	0...FFH	0...0.99609307	0.00390625	-
LDP_T_ES_2_REST	10	0...FFFFH	0...1.09225E+3	0.01666667	min
Considering the shut-off time for re-start calculation					
IP_FAC_REST	8x8	0...FFH	0...0.99609307	0.00390625	-
LDP_T_AST_REST	8	0...FFFFH	0...6.5535E+3	0.1	s
LDP_TCO_STOP__FAC_REST	8	0...FEH	-48...142.5	0.75	°C
Considering coolant stop temperature and after start time for re-start calculation					

33.3.1 FMSP_M7003

FUNCTION DESCRIPTION:

General information:

In order to improve re-starts, a correction factor of mass fuel flow during pre-injection and start injection is calculated. This factor is multiplicative and decreases the mass fuel flow. In order to prevent, that no calculation of T_ES was done, the T_ES calculation task is called.

Application Condition

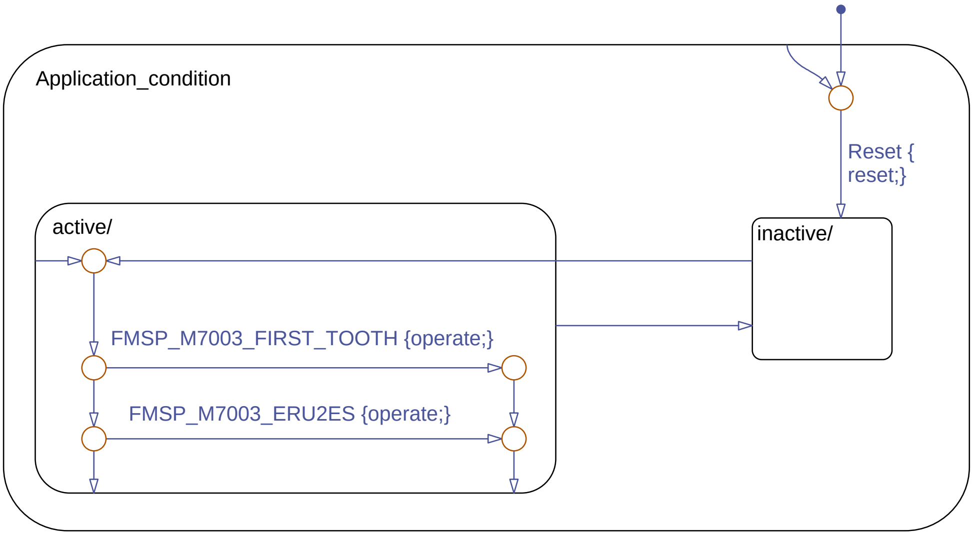


Figure 1 FMSP_M7003/ APP_CDN/ Chart

Function Description

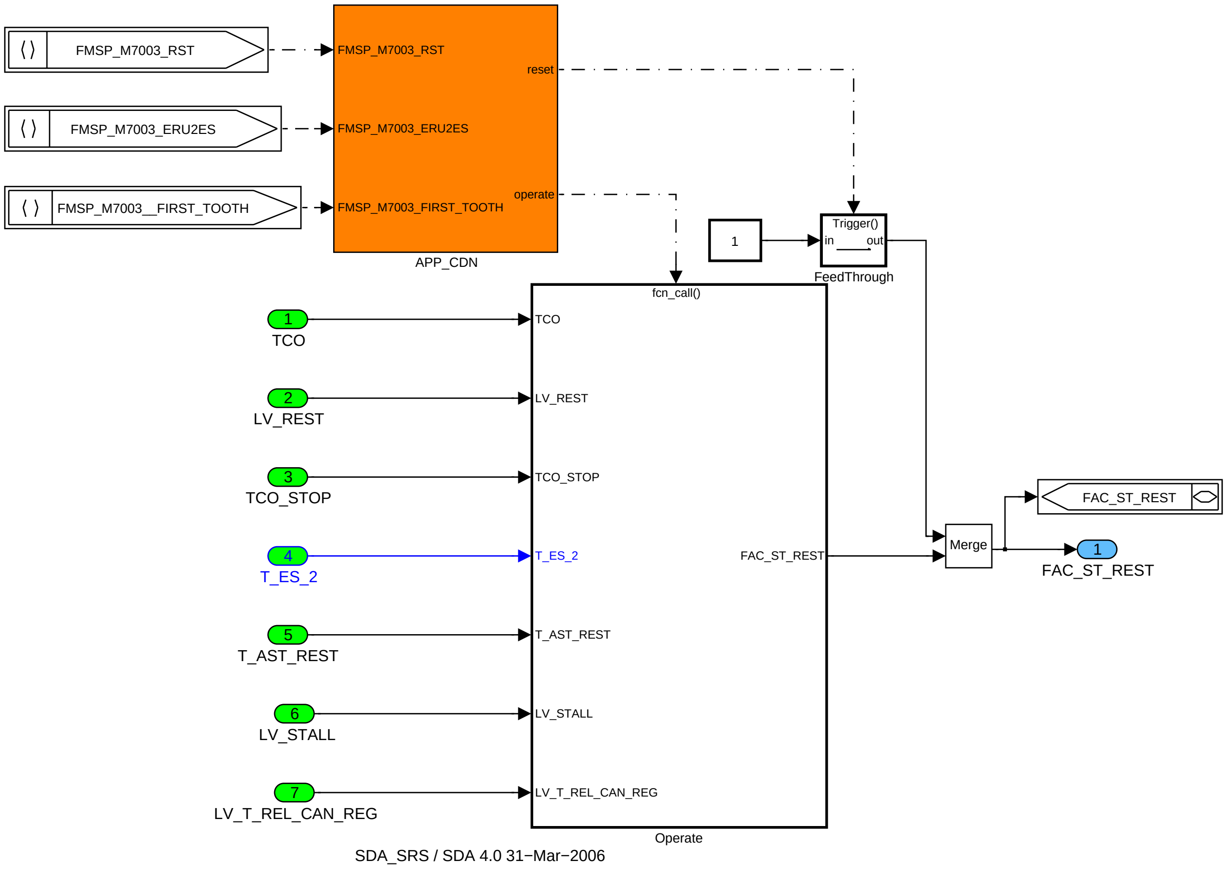


Figure 2 FMSP_M7003

33.3.1.1 SUBFUNCTION: Operate

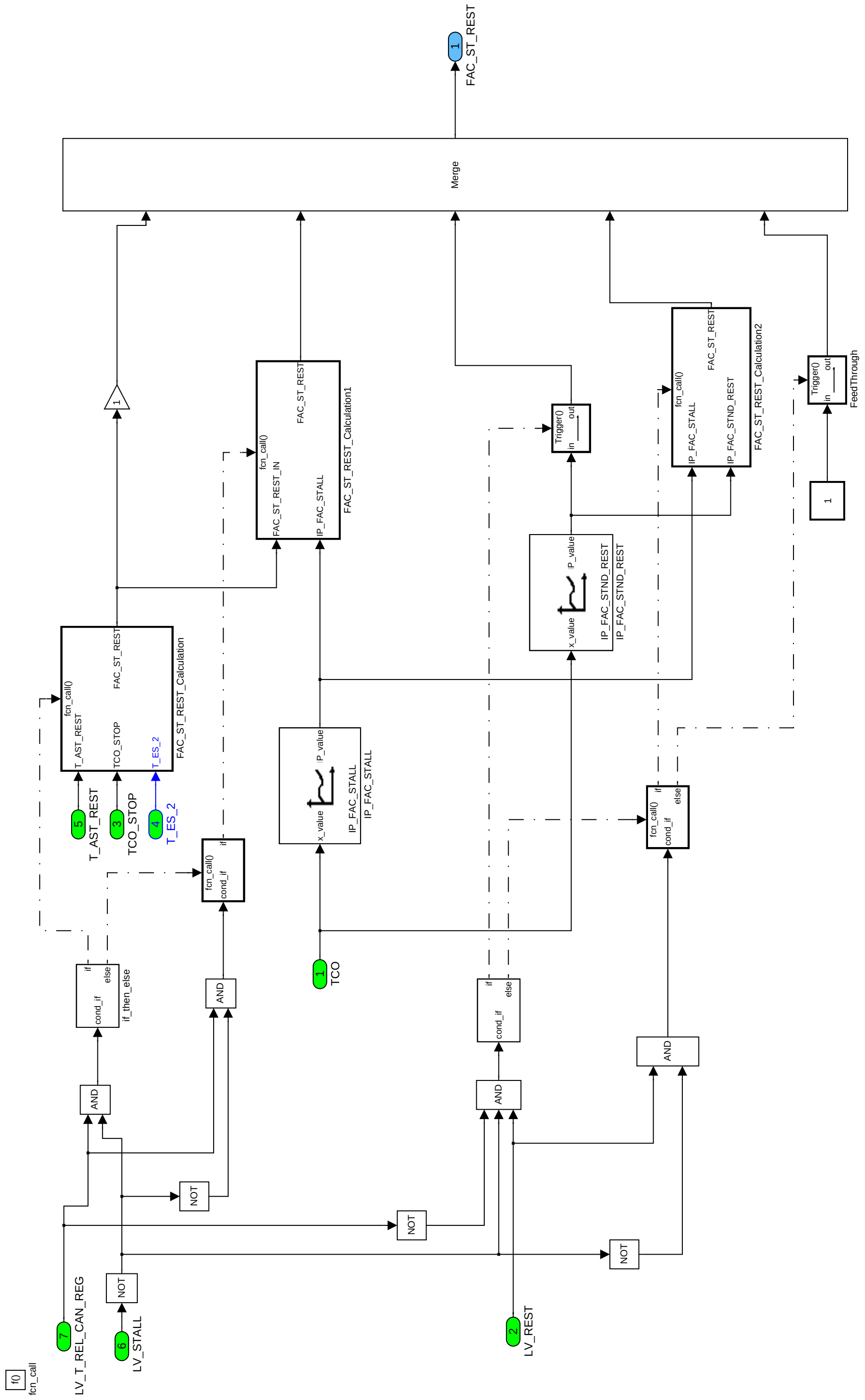


Figure 3 FMSP_M7003/ Operate

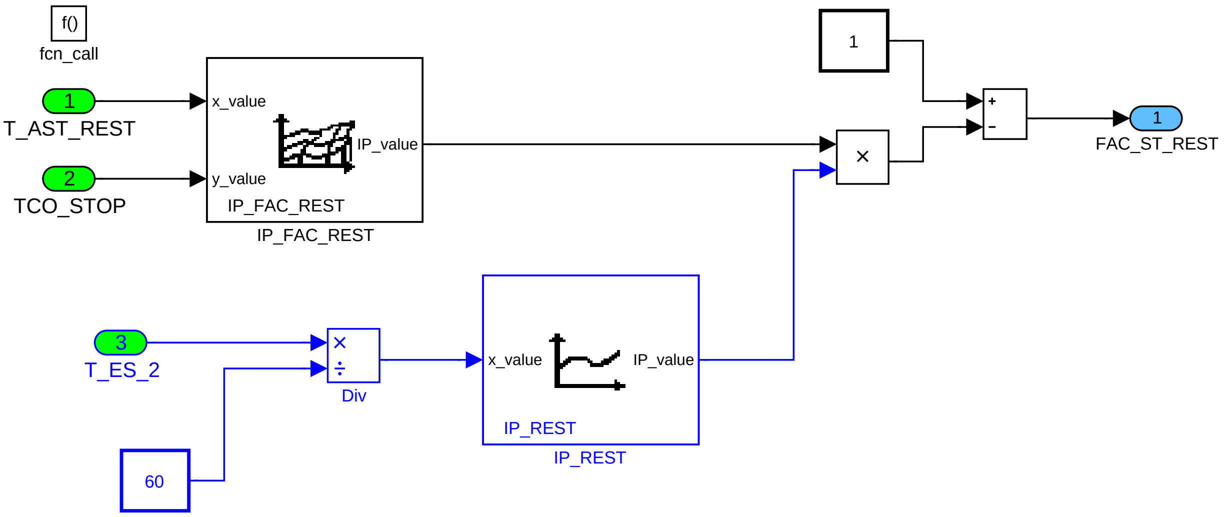


Figure 4 FMSP_M7003/ Operate/ FAC_ST_REST_Calculation

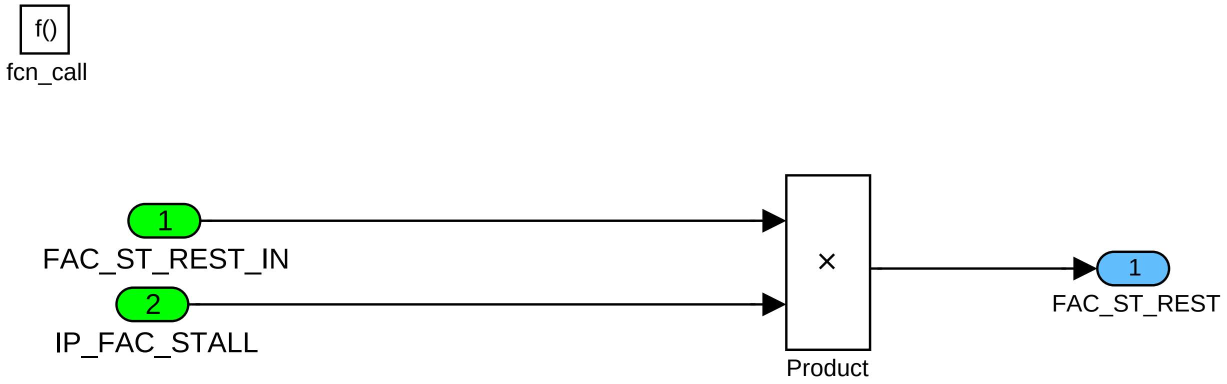


Figure 5 FMSP_M7003/ Operate/ FAC_ST_REST_Calculation1

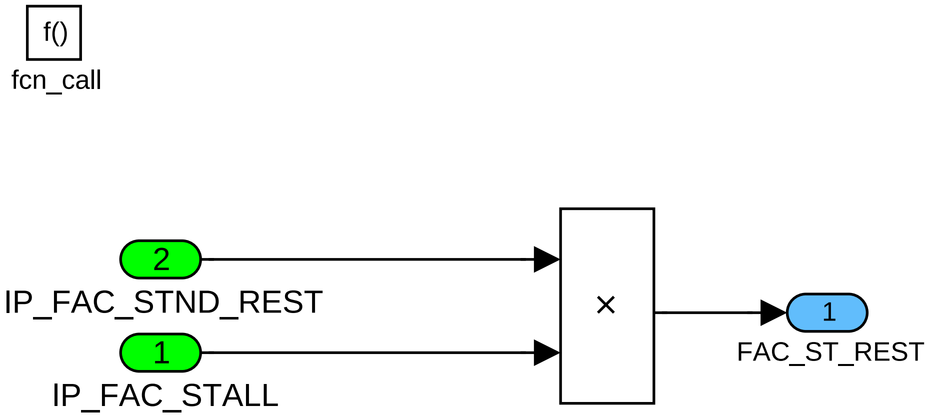


Figure 6 FMSP_M7003/ Operate/ FAC_ST_REST_Calculation2

33.3.1.2 Ambient pressure correction

Overview:

The purpose of this strategy is to correct the injection time depending on the ambient pressure before the MAF (manifold air pressure) variable is available.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_ST_AMP	V/O	0...FFH	0...1.9922	0.0078	-
Injection correction at start upon AMP					

Input data:

AMP	LV_ST_END		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_ST_AMP	6	0...FFH	0...1.9922	0.0078	-
LDP_AMP_IP_FAC_ST_AMP	6	0...FFFFH	0...5434	0.083	hPa
Cranking injection time correction upon ambient pressure					

Application conditions:

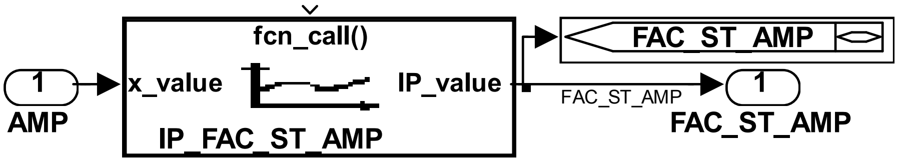
Activation: LV_ST_END =0

Deactivation: -

Initialization: -

Recurrence: 10 ms

FUNCTION DESCRIPTION:



33.4 Cold post start correction (TI_CAST)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TI_CAST	O/V	0...FFFFH	0...3.99993896	6.10352E-5	-
cold post start injection time correction					
LV_AST_END	O/V	0...1H	0...1	1	-
logical bit for after-start end					
TI_CAST_H_PRS	V	0...FFFFH	0...3.99993896	6.10352E-5	-
cold start fuel mass flow at high fuel pressure					
TI_CAST_LIH_L_PRS	V	0...FFFFH	0...3.99993896	6.10352E-5	-
cold start fuel mass flow at limb home low fuel pressure					
FAC_TI_CAST_FUP	V	0...FFH	0...1.9921875	0.0078125	-
Factor for weighting the influence of fuel pressure on cold post start injection					

Input data:

LV_AST	LV_CLC_2SEG	LV_CLC_2SEG_ENA	LV_ST_END
TCO	TIA	CYC_CAST	FAC_ST_REST
FUP	MAF_HB	N_32	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_FAC_TI_CAST_FUP	1	0...FFH	0...0.99609375	0.00390625	-
Low pass filter correlation constant for the weighting factor of fuel pressure					
IP_FAC_TI_CAST_FUP	6	0...FFH	0...1.9921875	0.0078125	-
LDP_FUP_FAC_TI_CAST	6	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Factor for weighting the influence of fuel pressure on cold post start injection					
IP_FAC_REST_TI_CAST_H_PRS	8x6	0...FFH	0...0.99609375	0.00390625	-
LDPM_CYC_CAST_1	8	0...FFFFH	0...6.5535E+4	1	-
LDPM_FAC_ST_REST	6	0...FFH	0...1.9921875	0.0078125	-
Correction of TI_CAST at restart high fuel pressure					
IP_FAC_REST_TI_CAST_LIH_L_PRS	8x6	0...FFH	0...0.99609375	0.00390625	-
LDPM_CYC_CAST_1	8	0...FFFFH	0...6.5535E+4	1	-
LDPM_FAC_ST_REST	6	0...FFH	0...1.9921875	0.0078125	-
Correction of TI_CAST at restart low fuel pressure					
IP_FAC_TI_CAST_H_PRS	8x8	0...FFH	0...1.9921875	0.0078125	-
LDPM_MAF_6	8	0...FFH	0...1.389E+3	5.44705882	mg/stk
LDPM_N_3_9	8	0...FFH	0...8.16E+3	32	rpm
Cold after start injection correction at high fuel pressure (engine operating point)					
IP_FAC_TI_CAST_LIH_L_PRS	8x8	0...FFH	0...1.9921875	0.0078125	-
LDPM_MAF_6	8	0...FFH	0...1.389E+3	5.44705882	mg/stk
LDPM_N_3_9	8	0...FFH	0...8.16E+3	32	rpm
Cold after start injection correction at limp home low fuel pressure (engine operating point)					
IP_TI_CAST_H_PRS	2x10	0...FFH	0...3.984375	0.015625	-
LDPM_TIA_IP_TI_CAST	2	0...FEH	-48...142.5	0.75	°C
LDPM_TCO__MFF_DEAC_CST	10	0...FEH	-48...142.5	0.75	°C
Initialization value of post start enrichment factor at high fuel pressure					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TI_CAST_LIH_L_PRS	2x10	0...FFH	0...3.984375	0.015625	-
LDPM_TIA_IP_TI_CAST	2	0...FEH	-48...142.5	0.75	°C
LDPM_TCO__MFF_DEAC_CST	10	0...FEH	-48...142.5	0.75	°C
Initialization value of post start enrichment factor at low fuel pressure					
IP_TI_DEAC_CAST_H_PRS	16x10	0...FFH	0...0.99609307	0.00390625	-
LDPM_CYC_CAST_2	16	0...FFFFH	0...6.5535E+4	1	-
LDPM_TCO__MFF_DEAC_CST	10	0...FEH	-48...142.5	0.75	°C
Deactivation factor of post start enrichment at high fuel pressure					
IP_TI_DEAC_CAST_LIH_L_PRS	16x10	0...FFH	0...0.99609307	0.00390625	-
LDPM_CYC_CAST_2	16	0...FFFFH	0...6.5535E+4	1	-
LDPM_TCO__MFF_DEAC_CST	10	0...FEH	-48...142.5	0.75	°C
Deactivation factor of post start enrichment at low fuel pressure					

33.4.1 FMSP_M7009

FUNCTION DESCRIPTION:

General information:

This enrichment takes place just after start in order to ensure the first combustions in cold conditions.

CYC_CAST is first the number of cycles (segments) at start. At the transition to after start, the counter is reset and represents then the number of after-start cycles during LV_AST=1.

Depending on fuel pressure FUP the functionality weighs up 2 cold post start injection times. Both injection times are calculated continuously and depending on a factor based on fuel pressure a interpolation between high and low pressure path estimates the executed after start injection TI_CAST.

Application conditions:

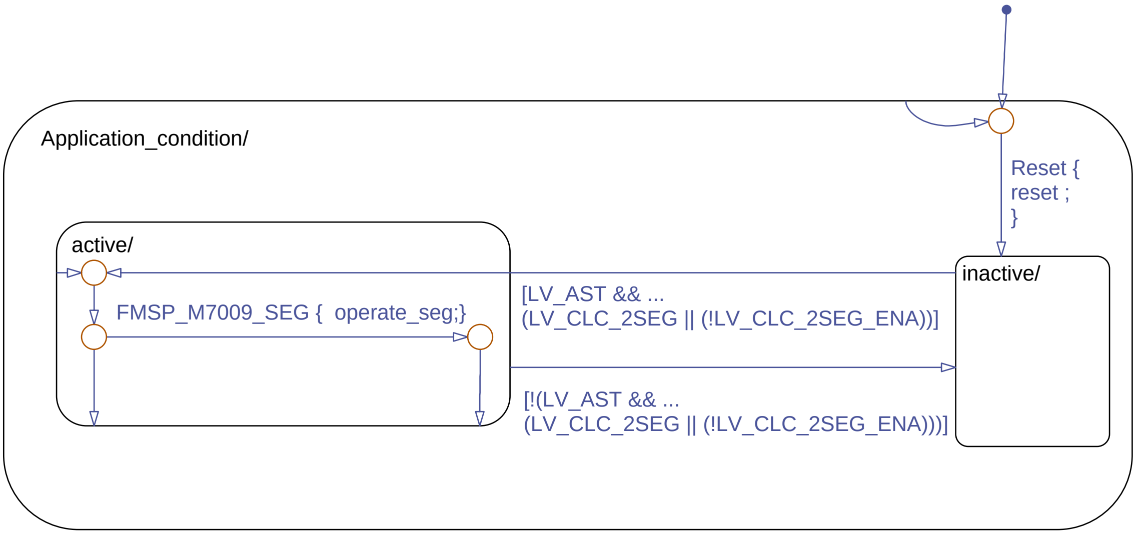
Initialisation: all outputs = 0 at ECU reset

Recurrence: segmentsynchronous

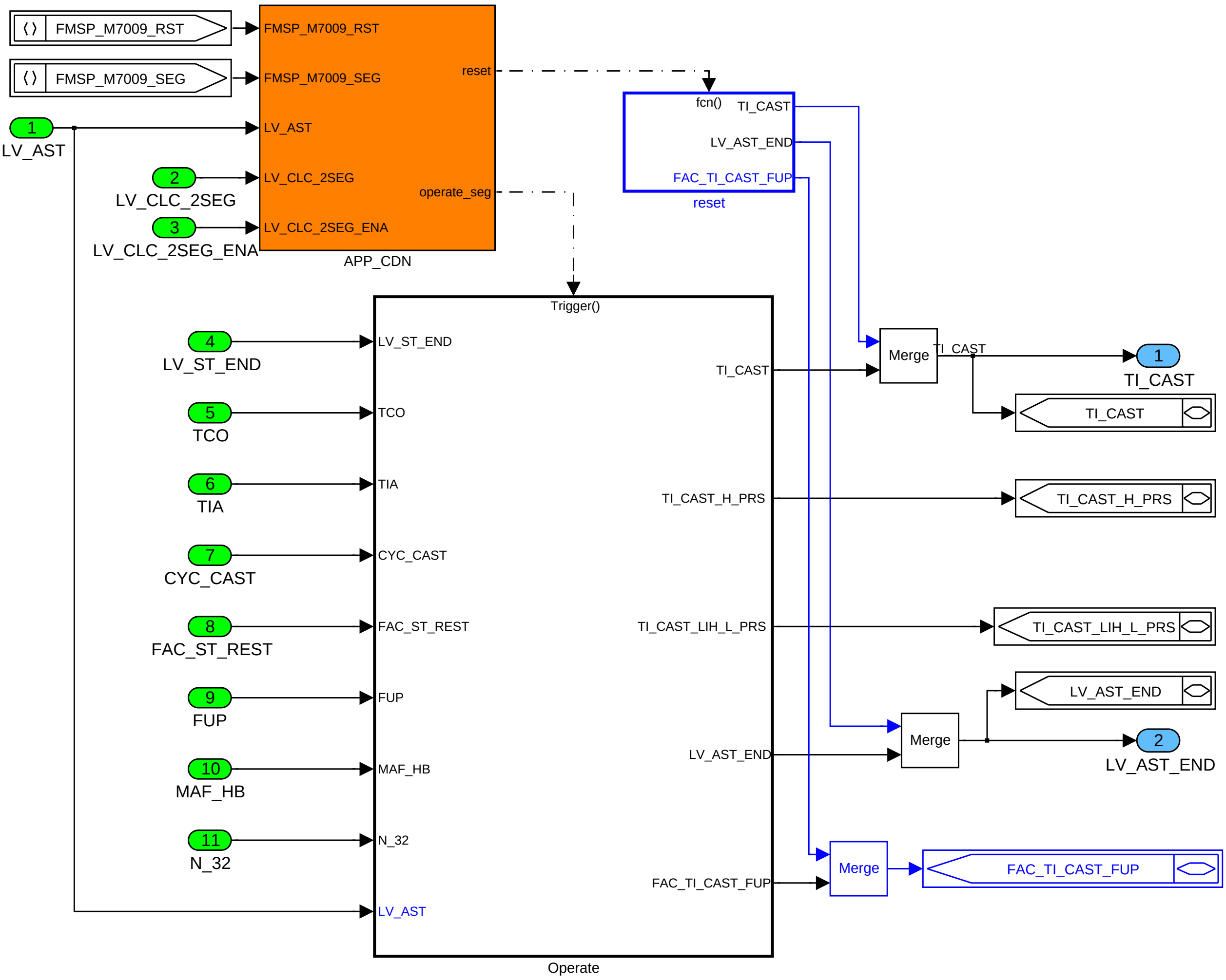
Activation: Post start auxiliary function (LV_AST) activated (exit from LV_ST to LV_IS or LV_PL) and (LV_CLC_2SEG = 1 or LV_CLC_2SEG_ENA = 0)

Deactivation: not Activation

Application Condition



Function Description



SDA_SRS / SDA 4.0 26−Mar−2007

Figure 7 FMSP_M7009

Chapter Fuel mass setpoint	Baseline 4DC3940S		Include File 43700901.00D
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 5468 of 9643

33.4.1.1 SUBFUNCTION: Operate

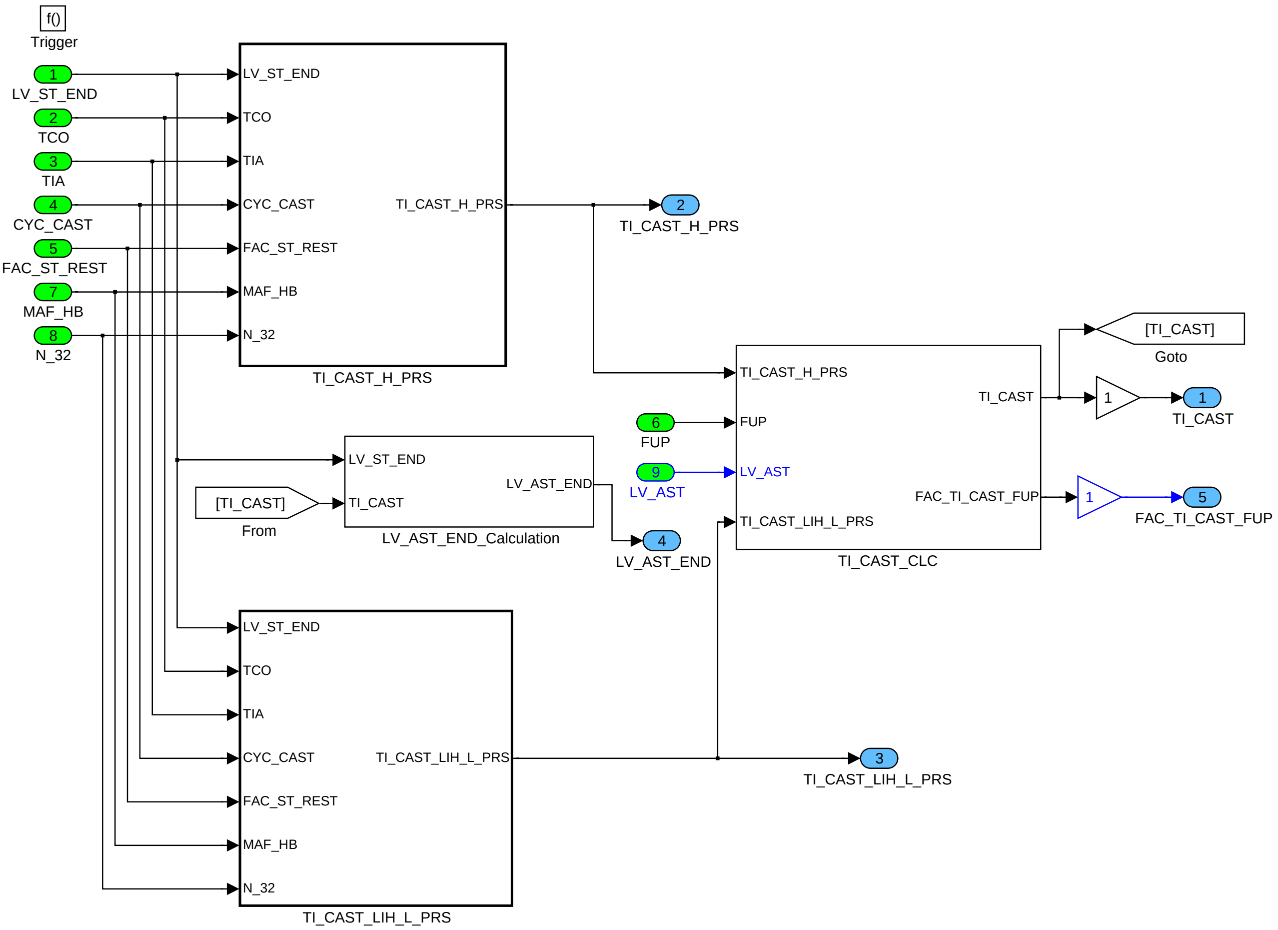


Figure 8 FMSP_M7009/ Operate

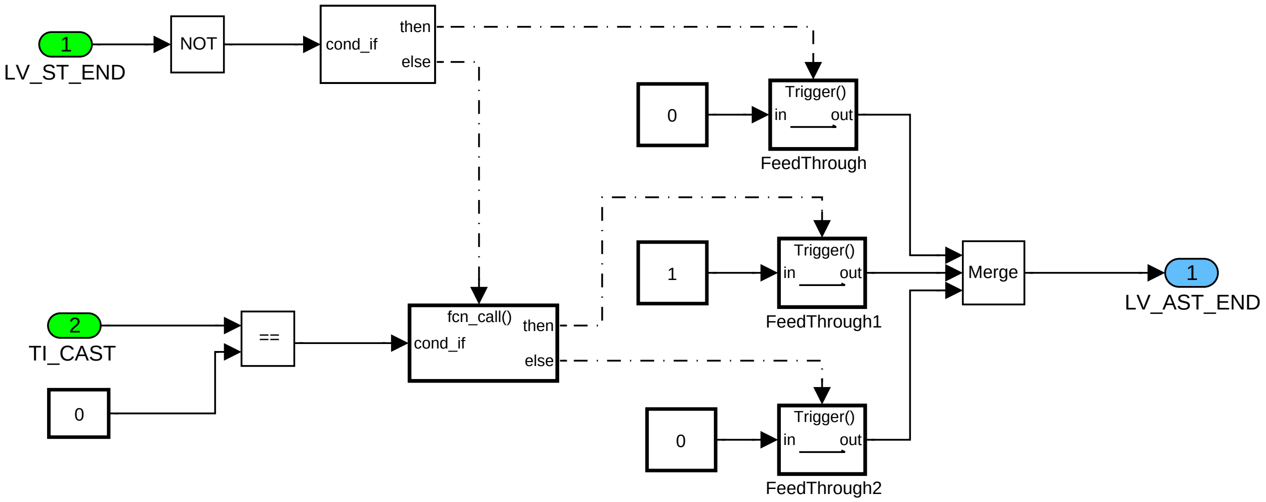


Figure 9 FMSP_M7009/ Operate/ LV_AST_END_Calculation

TI_CAST_SWI

There are two input values TI_CAST_H_PRS and TI_CAST_LIH_L_PRS. Each input value belongs to the fuel pressure. Moreover FUP indicates which path is more weighted. It is ensured that a continuous cross-fading in the range TI_CAST_H_PRS and

general specification

TI_CAST_LIH_L_PRS is active in the after start time interval. The algorithms are calculated until LV_AST drops.

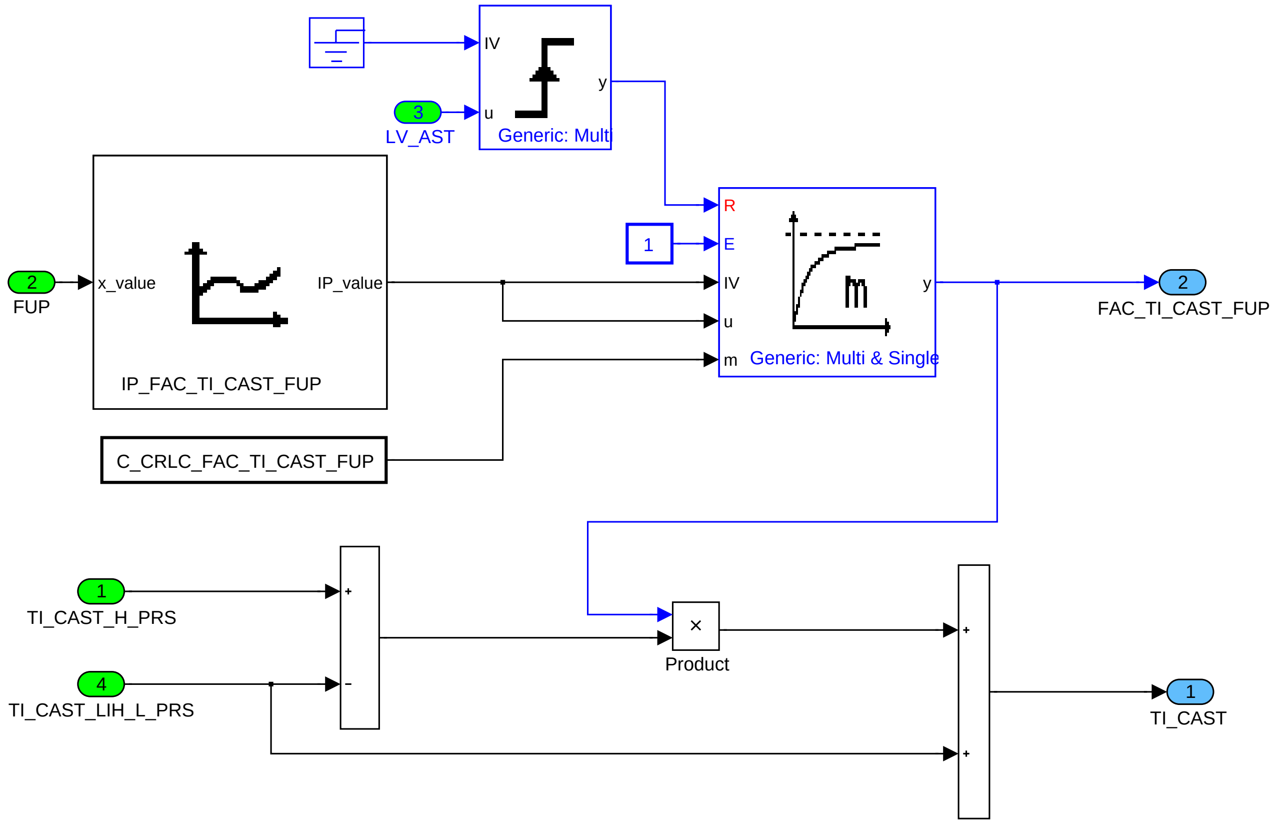


Figure 10 FMSP_M7009/ Operate/ TI_CAST_CLC

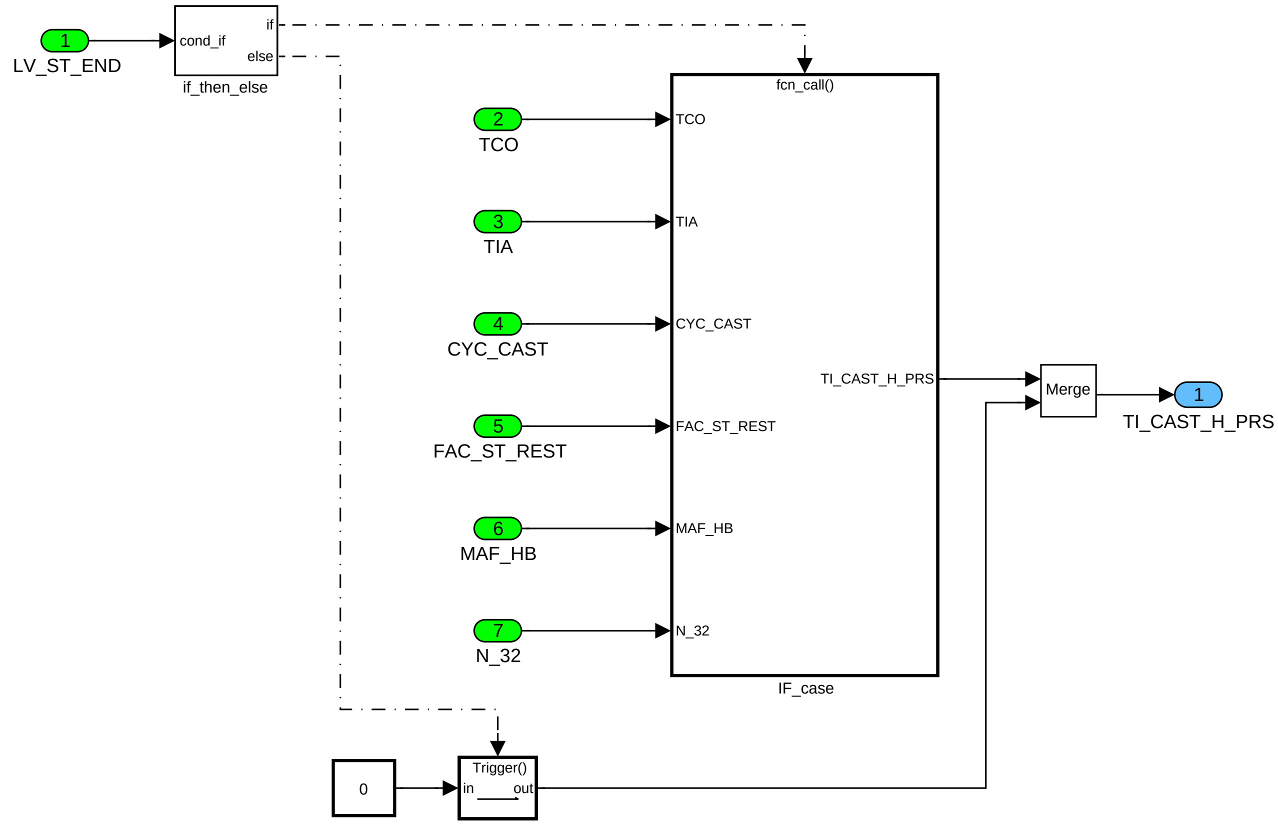


Figure 11 FMSP_M7009/ Operate/ TI_CAST_H_PRS

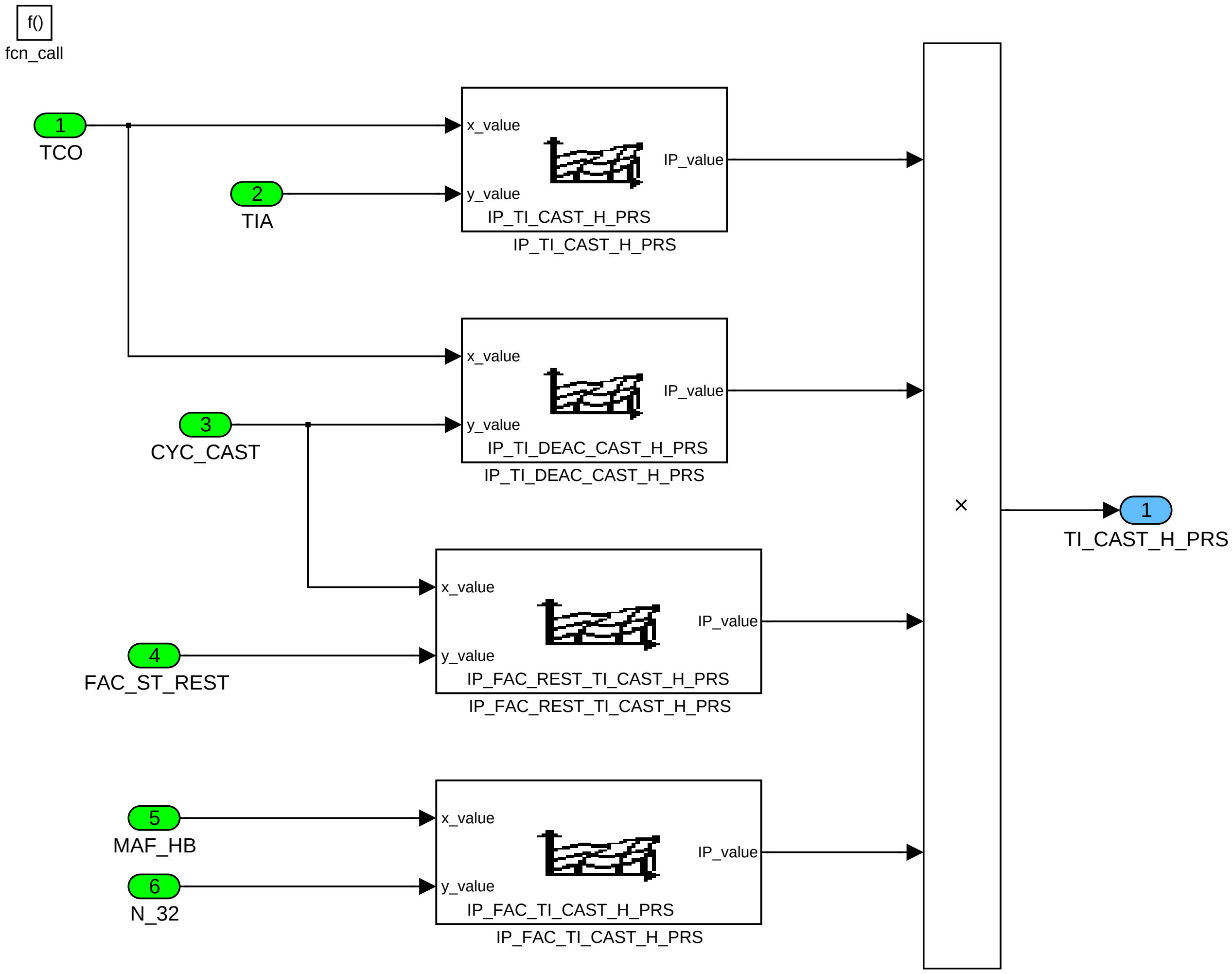


Figure 12 FMSP_M7009/ Operate/ TI_CAST_H_PRS/ IF_case

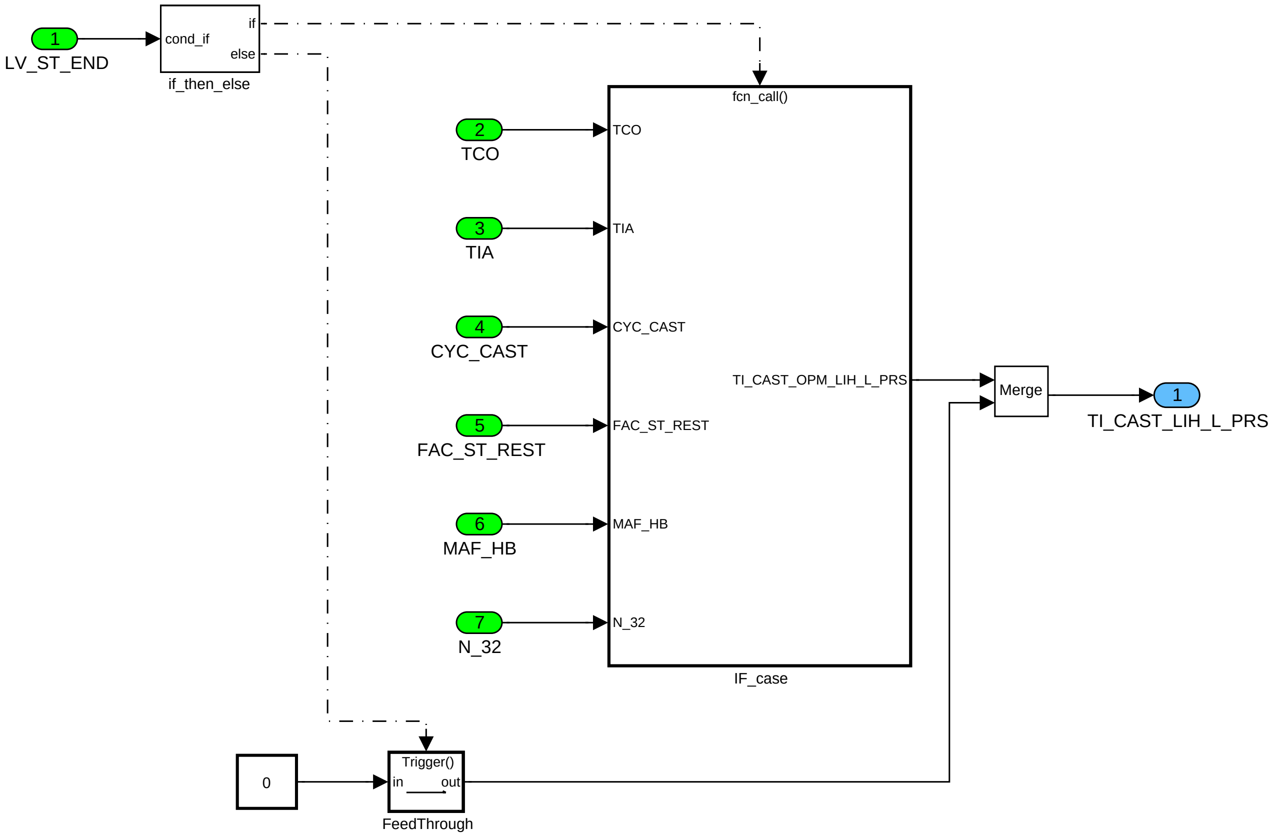


Figure 13 FMSP_M7009/ Operate/ TI_CAST_LIH_L_PRS

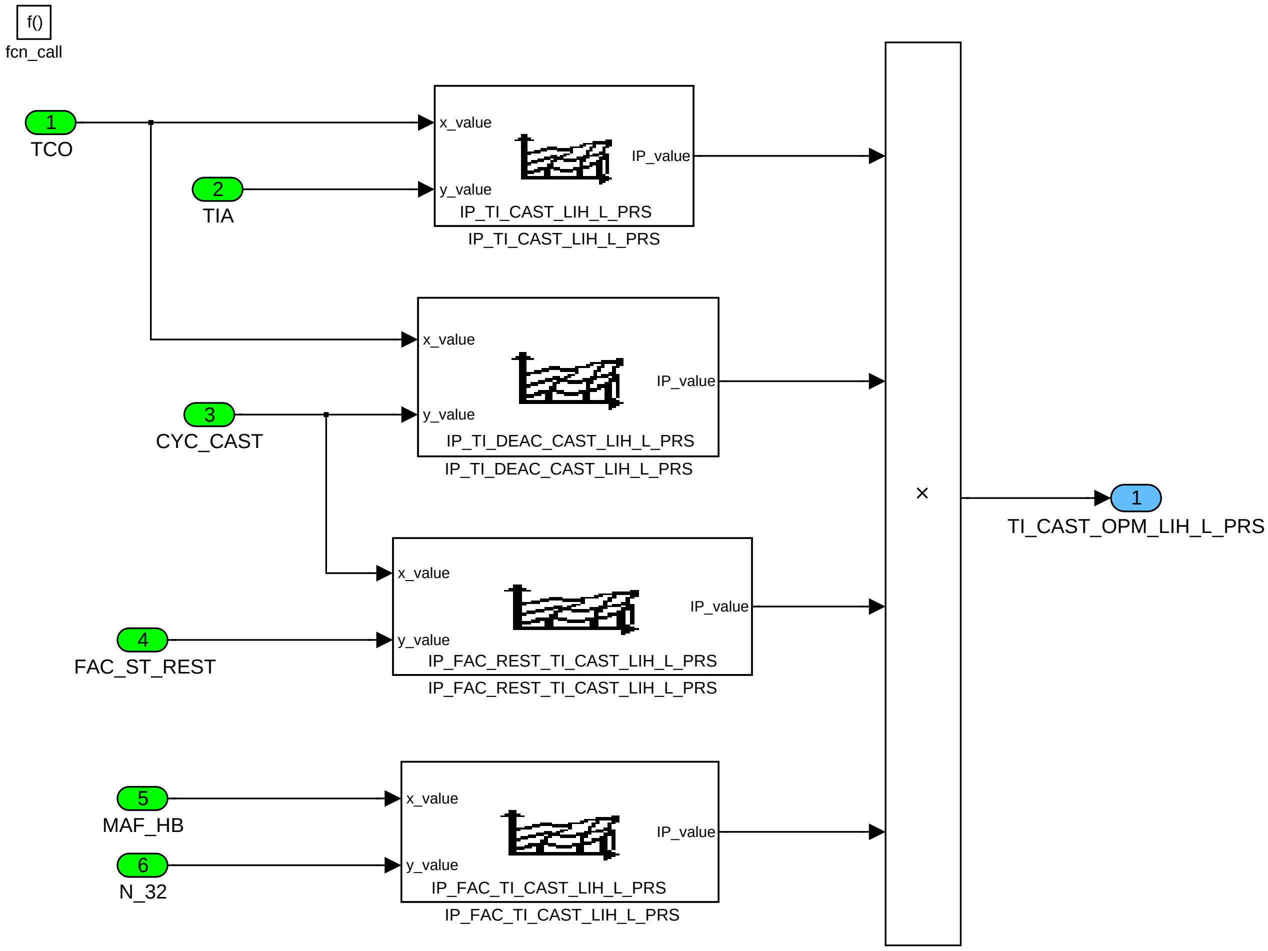


Figure 14 FMSP_M7009/ Operate/ TI_CAST_LIH_L_PRS/ IF_case

33.4.1.2 SUBFUNCTION: reset

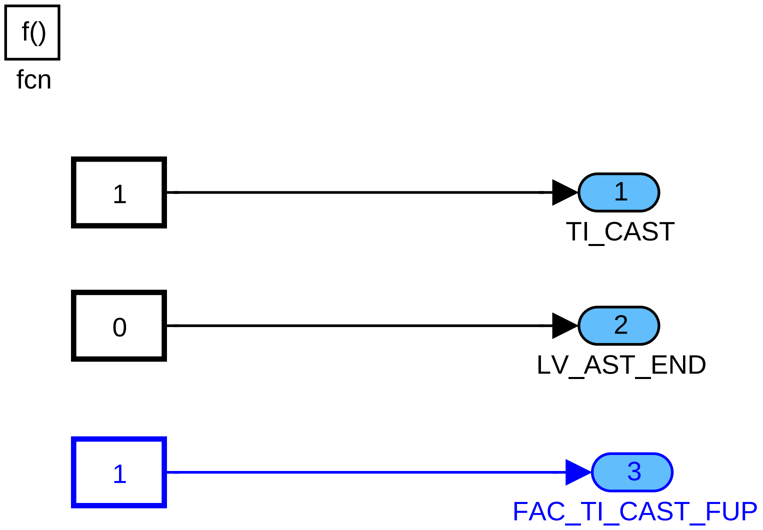


Figure 15 FMSP_M7009/ reset

33.5 Warm up correction

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TI_WUP	O/V	0...FFFFH	0...3.99993896	6.10352E-5	-
Warm-up enrichment, total value					
TI_WUP_1_ADD	O/V	0...FFFFH	0...3.99993896	6.10352E-5	-
Warm-up enrichment, additive part of intermediate value					
TI_WUP_1	V	0...FFFFH	0...3.99993896	6.10352E-5	-
Warm-up enrichment part after operation mode manager, intermediate value					
FAC_TI_WUP_FUP	V	0...FFH	0...1.9921875	0.0078125	-
Factor for weighting the influence of fuel pressure on warm up injection					
TI_TEMP	V	0...FFFFH	0...3.99993896	6.10352E-5	-
Warm-up enrichment part, intermediate, permanent part					
TI_WUP_1_H_PRS	V	0...FFFFH	0...3.99993896	6.10352E-5	-
Warm-up enrichment part at high fuel pressure					
TI_WUP_1_ADD_H_PRS	V	0...FFFFH	0...3.99993896	6.10352E-5	-
Warm-up enrichment, additive part of intermediate value at high fuel pressure					
TI_WUP_1_LIH_L_PRS	V	0...FFFFH	0...3.99993896	6.10352E-5	-
Warm-up enrichment part at low fuel pressure					
TI_WUP_1_ADD_LIH_L_PRS	V	0...FFFFH	0...3.99993896	6.10352E-5	-
Warm-up enrichment, additive part of intermediate value at low fuel pressure					

Input data:

LV_CLC_2SEG_ENA	LV_CLC_2SEG	MAF_HB	AMP
TOIL	N_32	MAP	LV_WUP
FUP	TCO	TCO_ST	TIA
LV_ES	INJ_MOD_HOM_REQ		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_FAC_TI_WUP_FUP	1	0...FFH	0...0.99609375	0.00390625	-
Low pass filter correlation constant for the weighting factor of fuel pressure					
IP_FAC_TI_TCO_WUP_H_PRS	8	0...FFH	0...1.9921875	0.0078125	-
LDPM_TCO__TI_WUP_H_PRS	8	0...FEH	-48...142.5	0.75	°C
warm up basic factor at high fuel pressure					
IP_FAC_TI_TCO_WUP_LIH_L_PRS	8	0...FFH	0...1.9921875	0.0078125	-
LDPM_TCO__TI_WUP_LIH_L_PRS	8	0...FEH	-48...142.5	0.75	°C
warm up basic factor at low fuel pressure					
IP_FAC_TI_WUP_FUP	6	0...FFH	0...1.9921875	0.0078125	-
LDP_FUP_FAC_TI_WUP	6	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Factor for weighting the influence of fuel pressure on warm up injection					
IP_TI_FAC_TCO	3	0...FFH	0...0.99609375	0.00390625	-
LDP_TCO__TI_FAC_TCO	3	0...FEH	-48...142.5	0.75	°C
Injection time weighting for temperature influence f(coolant temperature)					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_TI_VO_WUP	8x8	0...FFH	0...1.9921875	0.0078125	-
LDP_MAP_DIF__FAC_TI_VO_WUP	8	0...FFFFH	0...5.434E+3	0.08291752	hPa
LDPM_N_32__TI_WUP	8	0...FFH	0...8.16E+3	32	rpm
Factor map for TCO/VO dependent warm up correction					
IP_TI_FAC_N_MAP	8x8	0...FFH	0...0.99609375	0.00390625	-
LDP_MAP__TI_FAC_N_MAP	8	0...FFFFH	0...5.434E+3	0.08291752	hPa
LDP_N_32__TI_FAC_N_MAP	8	0...FFH	0...8.16E+3	32	rpm
Injection time weighting for temperature influence f(operating point)					
IP_TI_FAC_TIA_TOIL	6x6	0...FFH	0...1.9921875	0.0078125	-
LDP_TIA_TI_FAC_TIA_TOIL	6	0...FEH	-48...142.5	0.75	°C
LDP_TOIL_TI_FAC_TIA_TOIL	6	0...C8H	-40...160	1	°C
Injection time weighting for temperature influence f(TIA, TOIL)					
IP_TI_TCO_WUP_H_PRS	6x8	0...FFH	0...1.9921875	0.0078125	-
LDP_TCO_ST_TI_TCO_WUP_H_PRS	6	0...FEH	-48...142.5	0.75	°C
LDPM_TCO__TI_WUP_H_PRS	8	0...FEH	-48...142.5	0.75	°C
TCO/TCO_ST-dependent warm up correction at high fuel pressure					
IP_TI_TCO_WUP_LIH_L_PRS	6x8	0...FFH	0...1.9921875	0.0078125	-
LDP_TCO_ST_TI_TCO_WUP_LIH_L_PRS	6	0...FEH	-48...142.5	0.75	°C
LDPM_TCO__TI_WUP_LIH_L_PRS	8	0...FEH	-48...142.5	0.75	°C
TCO/TCO_ST-dependent warm up correction at low fuel pressure					
IP_TI_WUP_H_PRS	8x8	0...FFH	0...0.99609375	0.00390625	-
LDPM_MAF_HB__TI_WUP	8	0...FFH	0...1.389E+3	5.44705882	mg/stk
LDPM_N_32__TI_WUP	8	0...FFH	0...8.16E+3	32	rpm
TI warm up correction at high fuel pressure					
IP_TI_WUP_H_PRS_SNGH	8x8	0...FFH	0...0.99609375	0.00390625	-
LDPM_MAF_HB__TI_WUP	8	0...FFH	0...1.389E+3	5.44705882	mg/stk
LDPM_N_32__TI_WUP	8	0...FFH	0...8.16E+3	32	rpm
TI warm up correction at high fuel pressure for SNGH					
IP_TI_WUP_LIH_L_PRS	8x8	0...FFH	0...0.99609375	0.00390625	-
LDPM_MAF_HB__TI_WUP	8	0...FFH	0...1.389E+3	5.44705882	mg/stk
LDPM_N_32__TI_WUP	8	0...FFH	0...8.16E+3	32	rpm
TI warm up correction at low fuel pressure					

33.5.1 FMSP_M700A

FUNCTION DESCRIPTION:

General information:

To cover the increased fuel requirement while the engine is cold, injection is corrected via the warm up enrichment.

The function splits up 2 warm-up enrichment parts -intermediate value- (applied for 2 different fuel pressure modes). In accordance with the received fuel pressure FUP the warm up injection represents the weighted enrichment parts.

Application conditions:

Activation : at every engine operating state, when (LV_CLC_2SEG = 1 **or** LV_CLC_2SEG_ENA = 0)

Deactivation : **not** Activation

Initialization : at ECU-Initialisation, LV_ES = 0 ->1, LV_ST = 0 -> 1:

TI_WUP, TI_TEMP and FAC_TI_WUP_FUP = 1,0

TI_WUP_1 = 1,000061, TI_WUP_1_ADD = 0,000061

Update rate : 1/segment

Application Condition

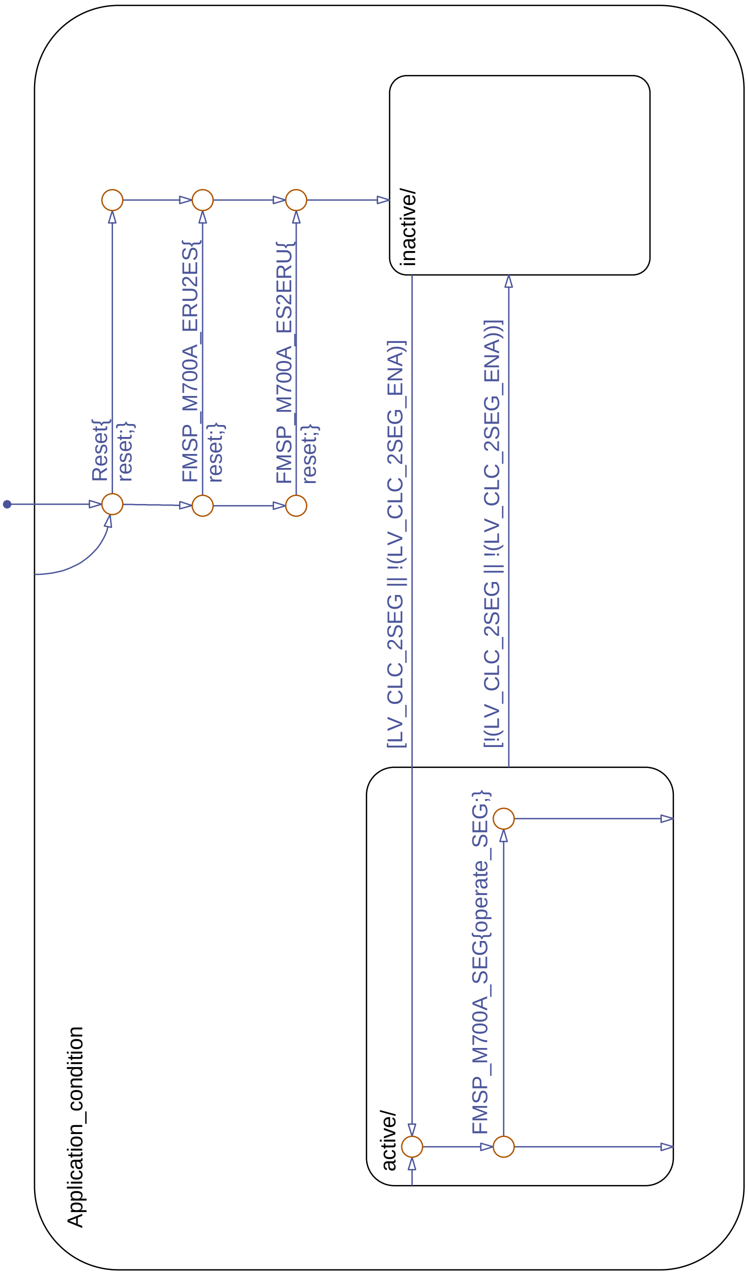
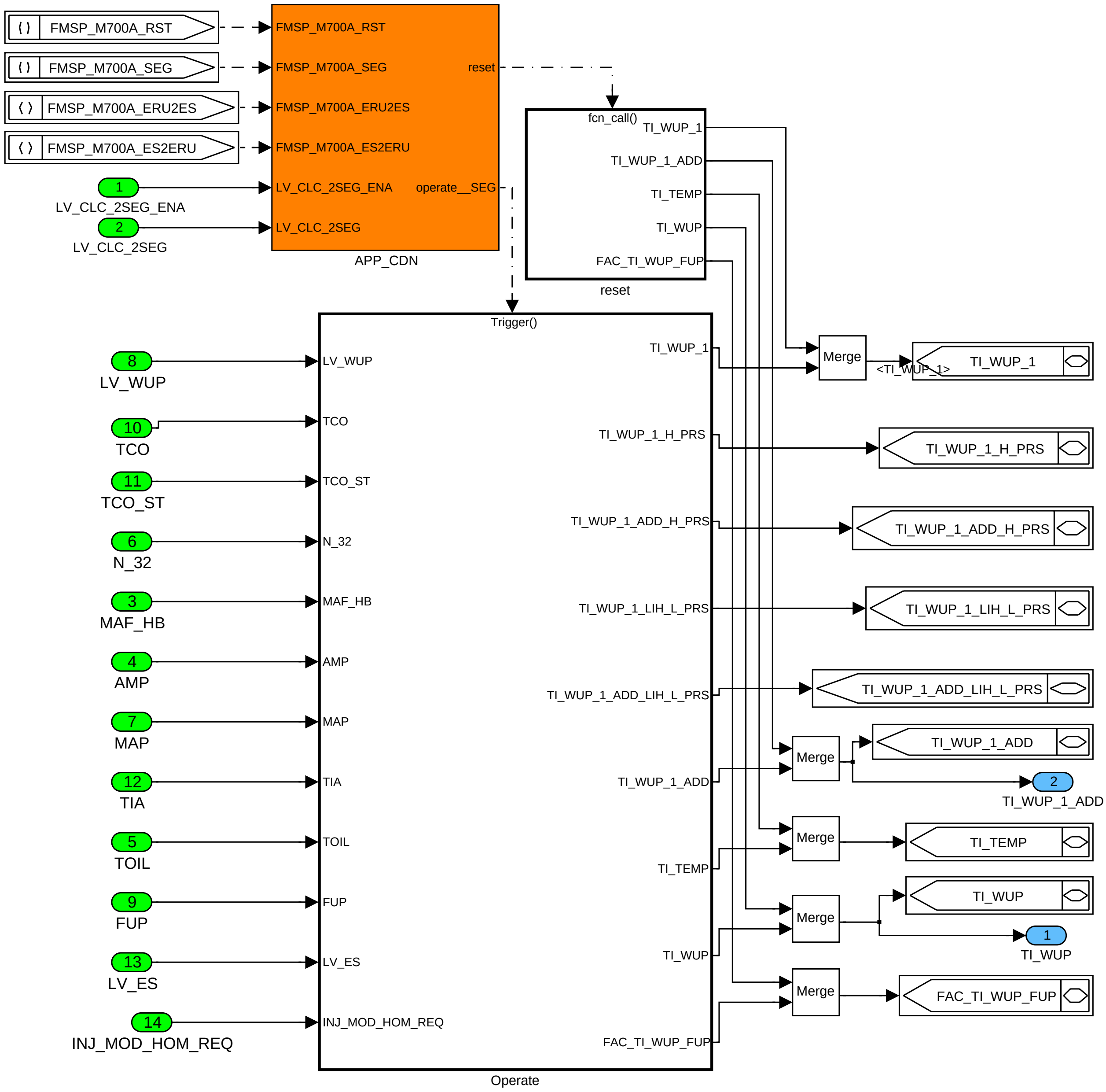


Figure 16 FMSP_M700A/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 26−Jan−2007

Figure 17 FMSP_M700A

33.5.1.1 SUBFUNCTION: Operate

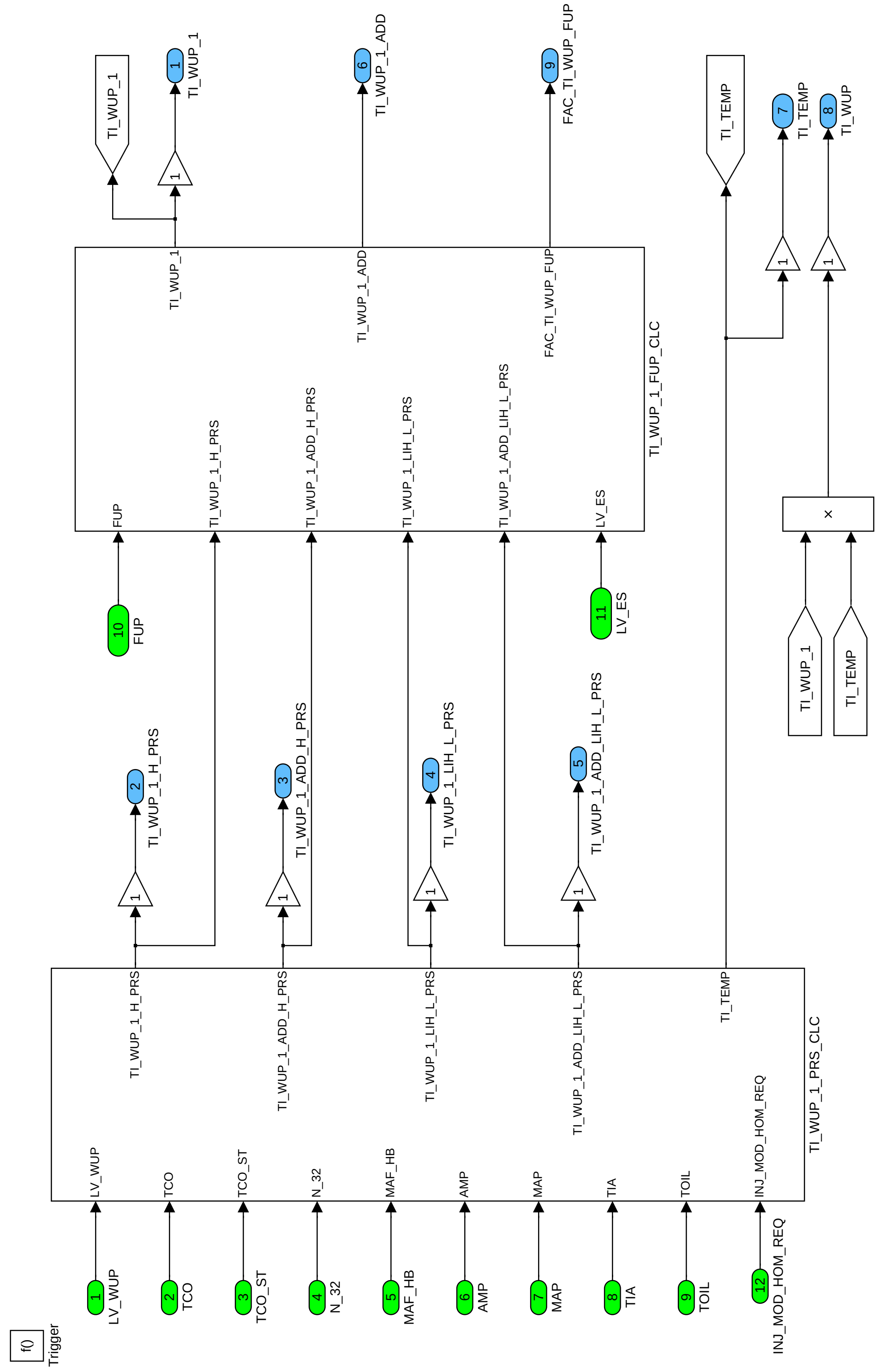


Figure 18 FMSP_M700A/ Operate

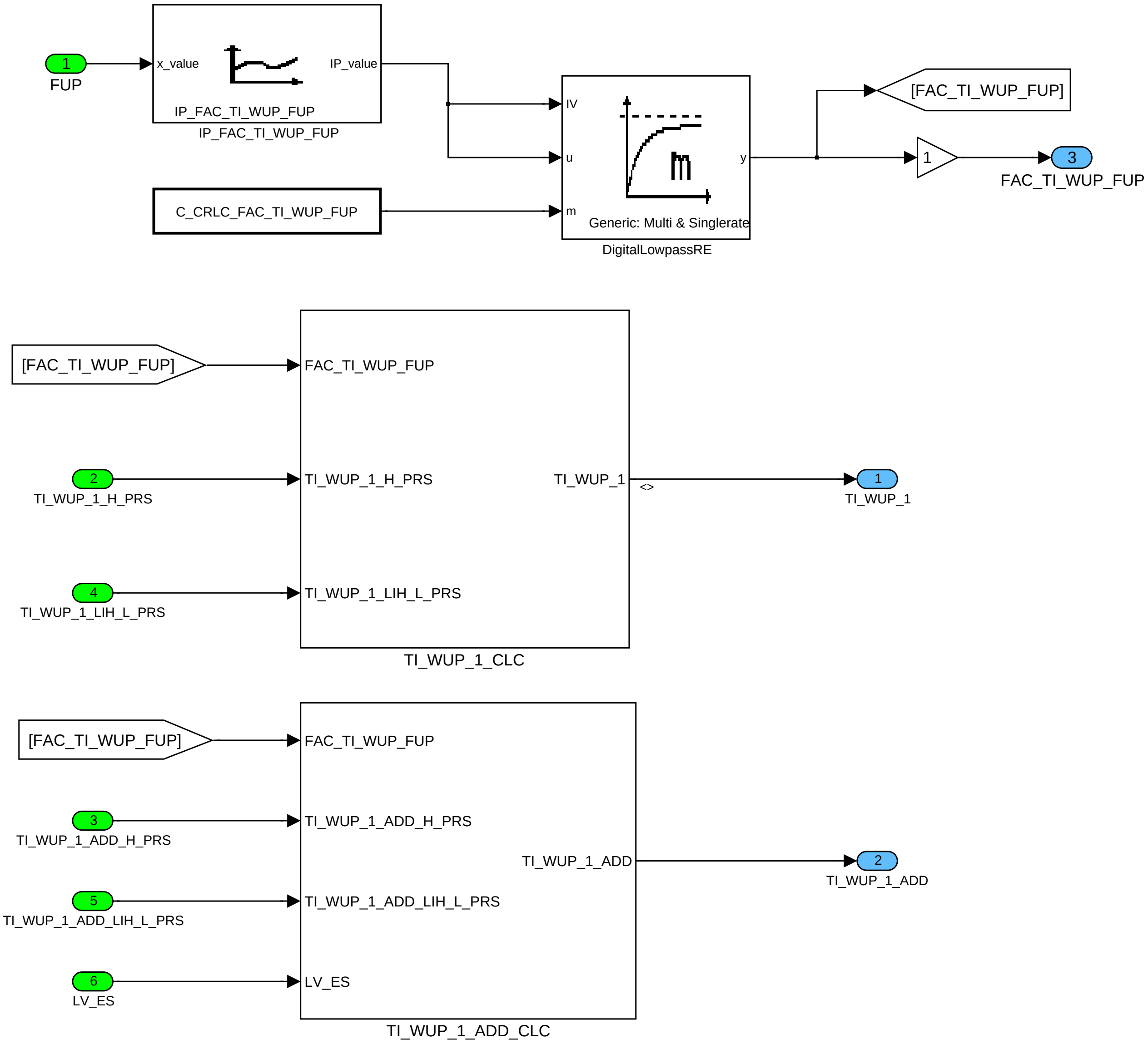


Figure 19 FMSP_M700A/ Operate/ TI_WUP_1_FUP_CLC

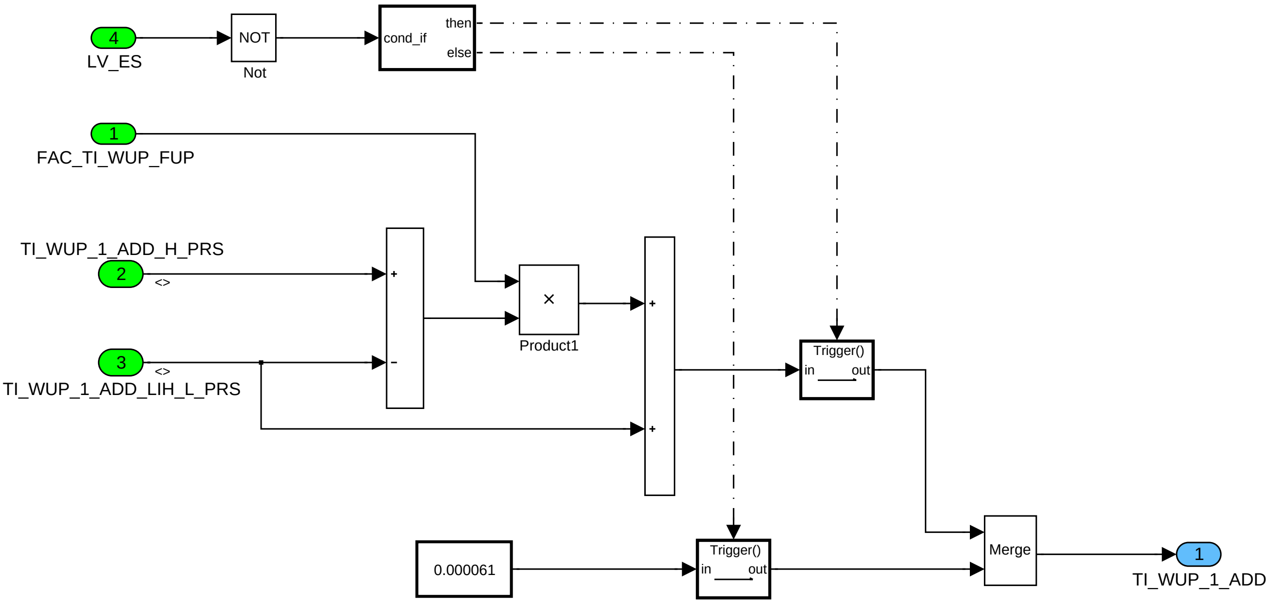


Figure 20 FMSP_M700A/ Operate/ TI_WUP_1_FUP_CLC/ TI_WUP_1_ADD_CLC

FMSP_M700A/OPERATE/TI_WUP_1_FUP_CLC/TI_WUP_1_CLC

The input values TI_WUP_1_H_PRS and TI_WUP_1_LIH_L_PRS are calculated for the fuel high pressure and the fuel limp home low pressure mode. Each input value belongs to a fuel pressure mode. Depending on FUP a factor indicates which path is executed as a result of the interpolation.

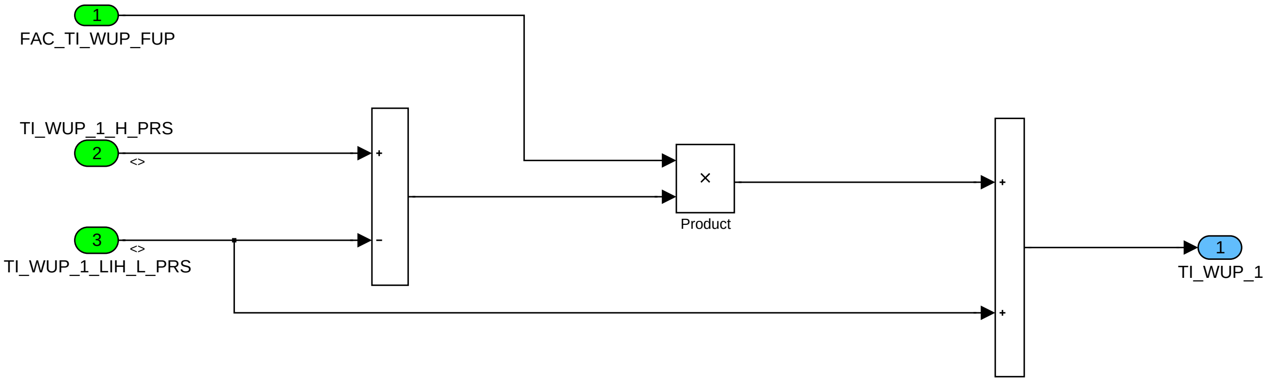


Figure 21 FMSP_M700A/ Operate/ TI_WUP_1_FUP_CLC/ TI_WUP_1_CLC

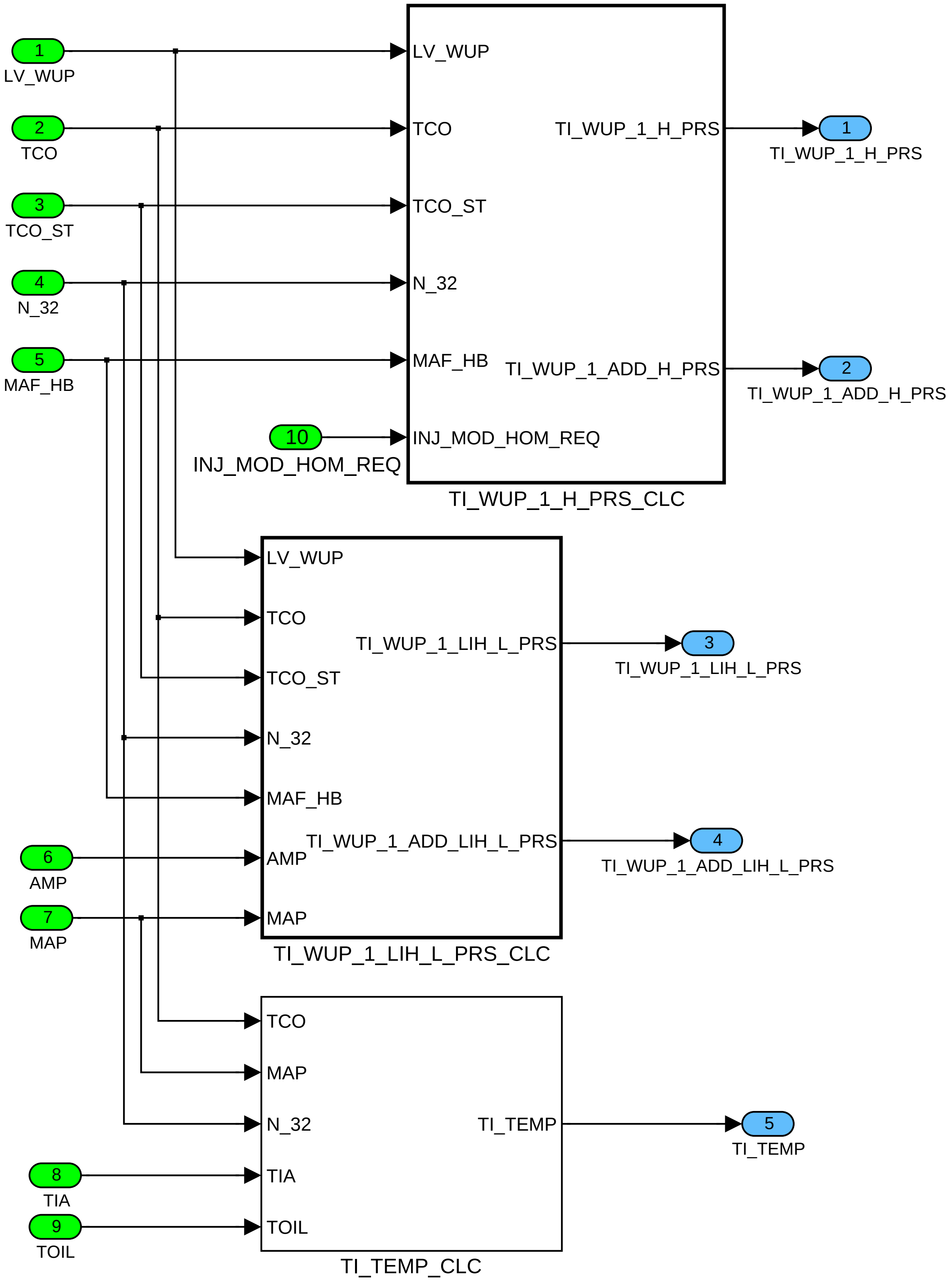


Figure 22 FMSP_M700A/ Operate/ TI_WUP_1_PRS_CLC

FMSP_M700A/OPERATE/TI_WUP_1_PRS_CLC/TI_TEMP_CLC

content

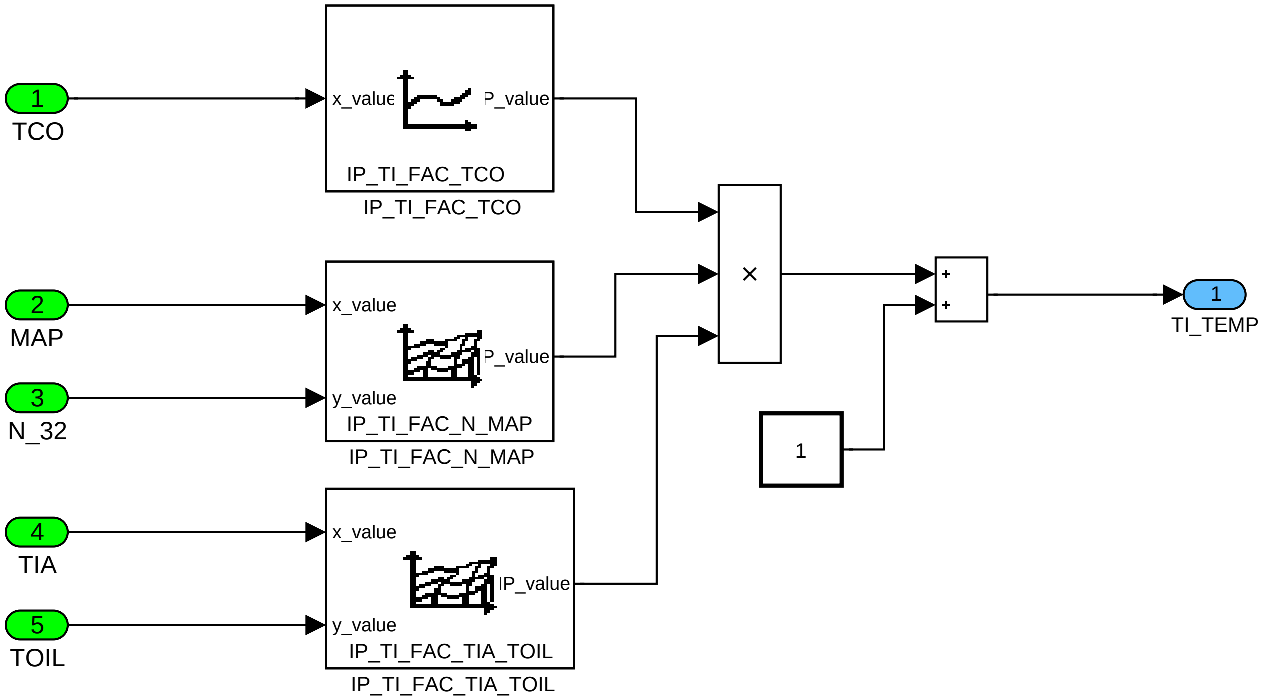


Figure 23 FMSP_M700A/ Operate/ TI_WUP_1_PRS_CLC/ TI_TEMP_CLC

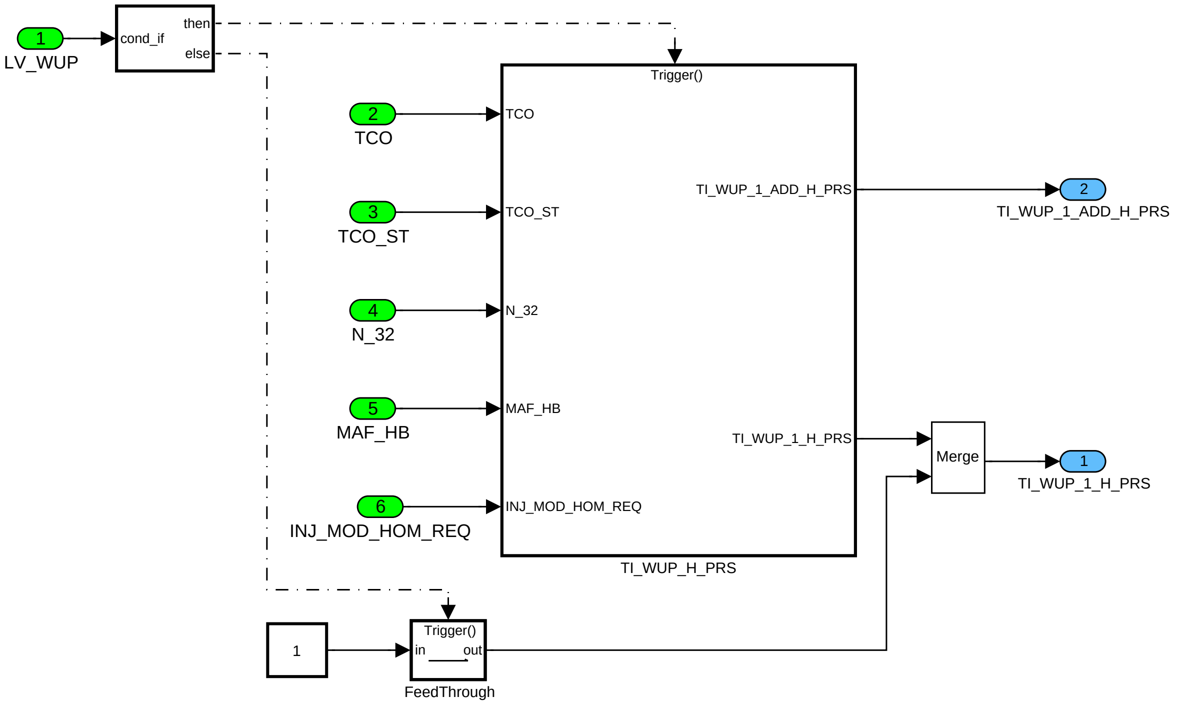


Figure 24 FMSP_M700A/ Operate/ TI_WUP_1_PRS_CLC/ TI_WUP_1_H_PRS_CLC

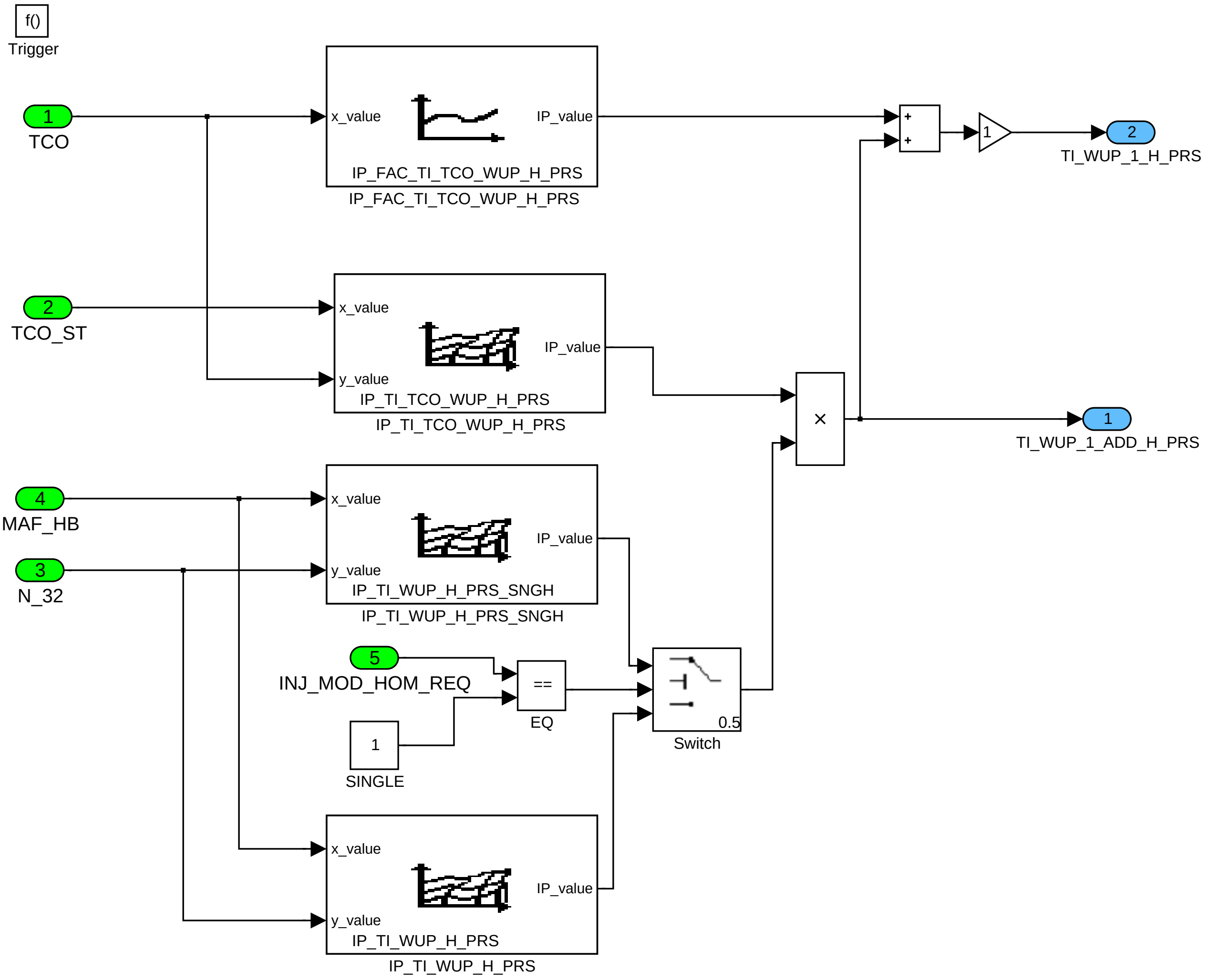


Figure 25 FMSP_M700A/ Operate/ TI_WUP_1_PRS_CLC/ TI_WUP_1_H_PRS_CLC/ TI_WUP_H_PRS

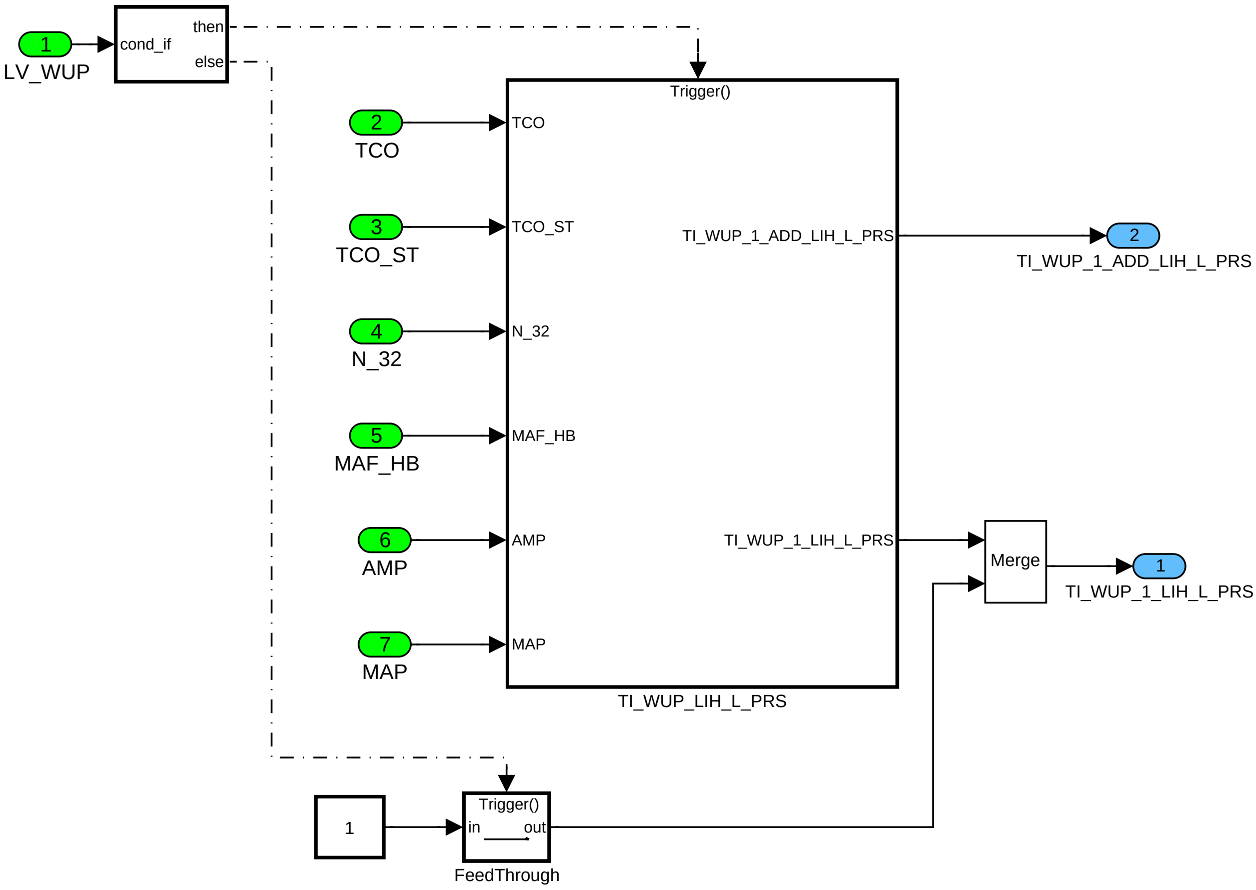


Figure 26 FMSP_M700A/ Operate/ TI_WUP_1_PRS_CLC/ TI_WUP_1_LIH_L_PRS_CLC

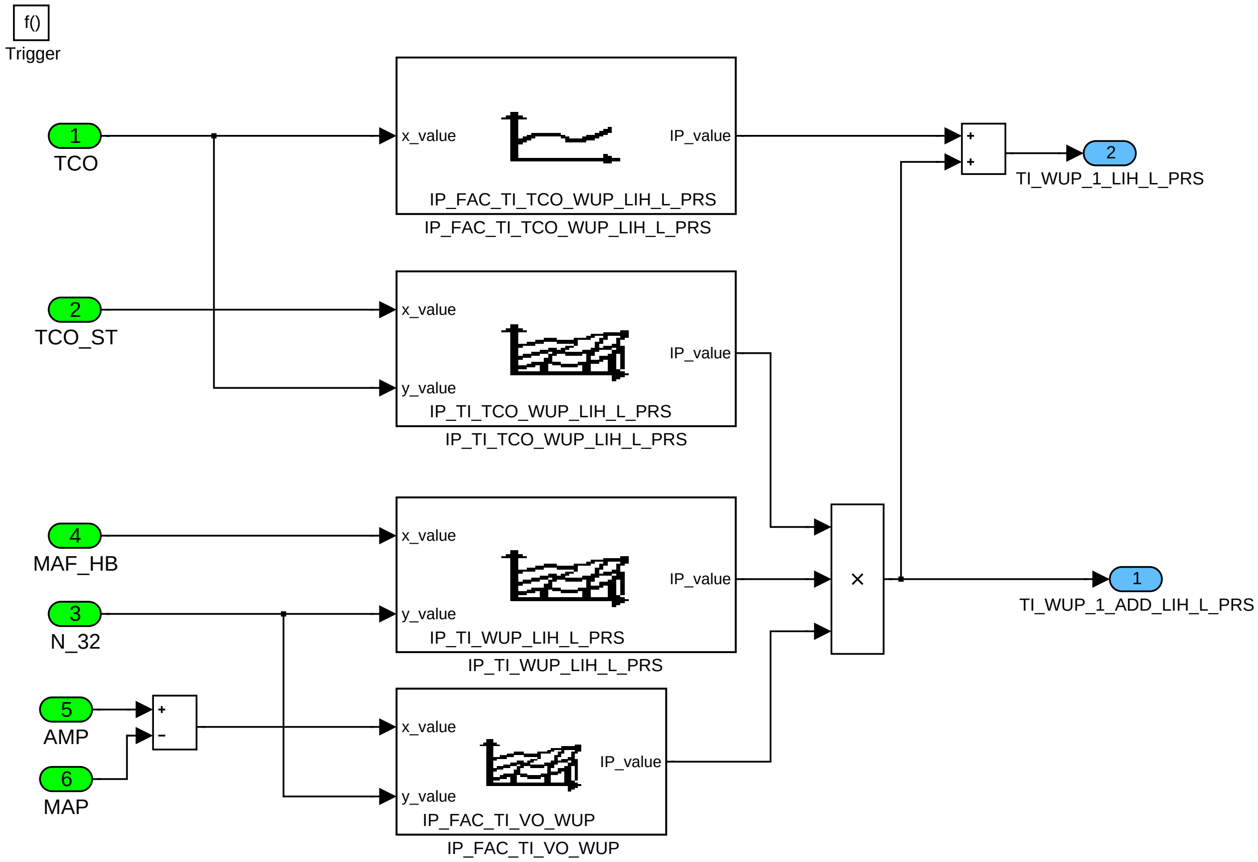


Figure 27 FMSP_M700A/ Operate/ TI_WUP_1_PRS_CLC/ TI_WUP_1_LIH_L_PRS_CLC/ TI_WUP_LIH_L_PRS

33.5.1.2 SUBFUNCTION: reset

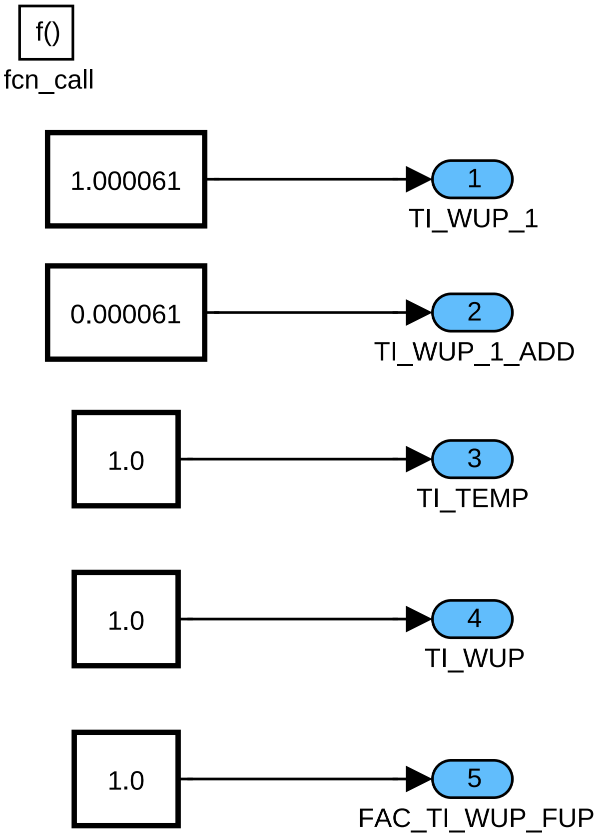


Figure 28 FMSP_M700A/ reset

33.6 Calculation of MFF setpoint

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MFF_SP_MV	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint after combustion selection					
MFF_SP[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint after combustion selection, bank selective					
MFF_SP_FUP_CTL	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint for fuel pressure control					
MFF_SP_HOM[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint for homogeneous mode, bank selective					
MFF_SP_HOM_ENG[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint after combustion selection					
MFF_SP_HOM_BAS_MV	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Arithmetic Mean of mass fuel flow setpoint for homogeneous mode					

Input data:

LV_ST_END	LV_IGA_AND_INJ_SWI	MFF_SP_2_HOM[NC_CYL_NR]	MFF_SP_1_HOM[NC_CYL_NR]
SUM_INH_INJ	MFF_SP_1_S[NC_CYL_NR]	MFF_SP_2_S[NC_CYL_NR]	MFF_SP_3_S[NC_CYL_NR]
MFF_SP_HOM_BAS[NC_CBK_EX_NR]			MFF_SP_3_HOM[NC_CYL_NR]

FUNCTION DESCRIPTION:

General information:

MFF_SP_MV is set to MFF_SP_HOM or MFF_SP_S depending on LV_IGA_AND_INJ_SWI.

MFF_SP serves as input, e.g. for the function „Fuel Pressure Control“.

Application conditions:

Initialisation: at reset

Recurrence: If LV_ST_END = 0: 10 ms
If LV_ST_END = 1: every TDC

Activation: every engine state

Deactivation: -

Formula section:

Calculation of MFF_SP_MV and MFF_SP_FUP_CTL:

```
IF      LV_IGA_AND_INJ_SWI =  1         (homogeneous mode)
THEN    MFF_SP_MV           =  (MFF_SP_1_HOM[x] + MFF_SP_2_HOM[x] +
                               MFF_SP_3_HOM[x]) / NC_CYL_NR
        MFF_SP_FUP_CTL      =  MFF_SP_MV * (1-(SUM_INH_INJ / NC_CYL_NR))
ELSE    (stratified mode)
        MFF_SP_MV           =  (MFF_SP_1_S[x] + MFF_SP_2_S[x] + MFF_SP_3_S[x]) /
                               NC_CYL_NR
        MFF_SP_FUP_CTL      =  MFF_SP_MV * (1-(SUM_INH_INJ / NC_CYL_NR))
ENDIF
```

Calculation of MFF_SP[NC_CBK_EX_NR]:

```
IF      NC_CBK_EX_NR > 1                      (please note physical meaning)
THEN
        FOR i = 1 TO NC_CBK_EX_NR             (please note physical meaning)
            MFF_SP[i] = 0
        ENDFOR
        FOR x = 0 TO (NC_CYL_NR – 1)
            IF (NC_LAMB_REF AND $2^x$) ≠ 0          (bitwise)
            THEN
                Bank 2:
                IF      LV_IGA_AND_INJ_SWI = 1        (homogeneous mode)
                THEN    MFF_SP[2] += MFF_SP_1_HOM[x] + MFF_SP_2_HOM[x]
                                     + MFF_SP_3_HOM[x]
                ELSE    (stratified mode)
                        MFF_SP[2] += MFF_SP_1_S[x] + MFF_SP_2_S[x] +
                                     MFF_SP_3_S[x]
                ENDIF
            ELSE
                Bank 1:
                IF      LV_IGA_AND_INJ_SWI = 1        (homogeneous mode)
                THEN    MFF_SP[1] += MFF_SP_1_HOM[x] + MFF_SP_2_HOM[x]
                                     + MFF_SP_3_HOM[x]
                ELSE    (stratified mode)
                        MFF_SP[1] += MFF_SP_1_S[x] + MFF_SP_2_S[x] +
                                     MFF_SP_3_S[x]
                ENDIF
```

```
        ENDIF
    ENDFOR
    FOR i = 1 TO NC_CBK_EX_NR
        MFF_SP[i] = MFF_SP[i] / (NC_CYL_NR / NC_CBK_EX_NR)
    ENDFOR
ELSE
    MFF_SP[1] = 0
    FOR x = 0 TO (NC_CYL_NR – 1)
        IF      LV_IGA_AND_INJ_SWI = 1      (homogeneous mode)
        THEN    MFF_SP[1] += MFF_SP_1_HOM[x] + MFF_SP_2_HOM[x]
                                  + MFF_SP_3_HOM[x]
        ELSE    (stratified mode)
                MFF_SP[1] += MFF_SP_1_S[x] + MFF_SP_2_S[x] +
                             MFF_SP_3_S[x]
        ENDIF
    ENDFOR
    MFF_SP[1] = MFF_SP[1] / NC_CYL_NR
ENDIF
```

Calculation of MFF_SP_HOM[NC_CBK_EX_NR]:

```
IF      NC_CBK_EX_NR > 1                    (please note physical meaning)
THEN
    FOR i = 1 TO NC_CBK_EX_NR               (please note physical meaning)
        MFF_SP_HOM[i] = 0
    ENDFOR


    FOR x = 0 TO (NC_CYL_NR – 1)
        IF (NC_LAMB_REF AND 2^x) ≠ 0               (bitwise)
        THEN
            Bank 2:
            MFF_SP_HOM[2] += MFF_SP_1_HOM[x] + MFF_SP_2_HOM[x]
                                  + MFF_SP_3_HOM[x]
        ELSE
            Bank 1:
            MFF_SP_HOM[1] += MFF_SP_1_HOM[x] + MFF_SP_2_HOM[x]
                                  + MFF_SP_3_HOM[x]
        ENDIF
    ENDFOR
```

```
        FOR i = 1 TO NC_CBK_EX_NR
            MFF_SP_HOM[i] = MFF_SP_HOM[i] / (NC_CYL_NR / NC_CBK_EX_NR)
            MFF_SP_HOM_ENG[i] = MFF_SP_HOM[i]
        ENDFOR

    ELSE
        MFF_SP_HOM[1] = 0
        FOR x = 0 TO (NC_CYL_NR – 1)
            MFF_SP_HOM[1] += MFF_SP_1_HOM[x] + MFF_SP_2_HOM[x]
                              + MFF_SP_3_HOM[x]
        ENDFOR
        MFF_SP_HOM[1] = MFF_SP_HOM[1] / NC_CYL_NR
        MFF_SP_HOM_ENG[1] = MFF_SP_HOM[1]
    ENDIF
```

Calculation of MFF_SP_HOM_BAS_MV:

MFF_SP_HOM_BAS_MV = **SUM(** MFF_SP_HOM_BAS[NC_CBK_EX_NR] **)**
/ NC_CBK_EX_NR

33.7 Fuel consumption calculation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_VS_MIN_FCO	V	0...FFH	0...255	1	[-]
Counter indicating time since VS is below threshold					
FAC_FCO_FIL[2]	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Filter factor for FCO filter					
FCO	O/V	0...FFFFFFFFH	0...4294967295	1	[µl]
integrated fuel consumption					
FCO_AV	V	0...FFFFH	0...65535	1	[µl]
actual fuel consumption					
FCO_AV_1	O/V	0...FFFFH	0...65535	1	[µl]
actual fuel consumption					
FCO_AV_2	O/V	0...FFFFH	0...65535	1	[µl]
actual fuel consumption					
FCO_AV_FIL[2]	V	0...FFFFFFFFH	0...65535	0.0153e-3	[µl]
actual fuel consumption filtered					
FCO_AV_MMV	O/V	0...FFFFH	0...65535	1	[µl]
Moving mean value of the actual fuel consumption					
FCO_DMTL	O/V	0...FFFFFFFFH	0...4294967295	1	[µl]
integrated fuel consumption DMTL					
FCO_FIL_DIF	V	80000000... 7FFFFFFFH	-2147483648... 2147483647	1	[µl]
difference of actual and filtered actual fuel consumption					
FCO_FIL_DIF_FB	V	80000000... 7FFFFFFFH	-2147483648... 2147483647	1	[µl]
Feedback part to FCO calculation of difference of actual and filtered actual fuel consumption					
FCO_MFF_ADD_CYL_CP	-	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass from the canister in the cylinder, temp. value for FCO calculation					
T_FAC_FCO_FIL[2]	V	0...FFH	0...25.5	0.1	[s]
Filter time for FCO filter					
T_RAMP_FCO_FIL	V	0...FFFFH	0...655.35	0.01	[s]
Timer to ramp down the FCO-filter					

Input data:

MFF_SP[NC_CBK_EX_NR]	LV_PUC	EFF_SCC_AV	FAC_FCO_KWP
N_32	NC_CYL_NR	LV_IGK	LV_REQ_ISC
OPM_AV	VS	MFF_SP_FCO_COR	MFF_ADD_CYL_CP
NC_LAMB_REF	LV_FCO_COR_REQ	MFF_SP_BAS	SEG_NR
LV_PUC_REQ	LV_PUC_INH_TEMP_CAT	LV_PUC_LOCK_TNT	LV_LDM_PUC_INH
LV_DCC_PUC_INH	LV_AMT_ACT	LV_PUC_SA_INH	LV_DCC_INC_ACT
LV_GS_INC_ACT	LV_INH_PUC_CUS	LV_ES	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_MFF_FCO	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Transition factor MFF to FCO					
C_CRLC_FCO_AV	1	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation constant for moving mean value calculation of FCO					
C_FCO_DIF_MAX	1	0...FFFFH	0...65535	1	[µl]
threshold to activate balance controller					
C_FCO_DIF_GAIN	1	0...FFFFH	0...1	0.0153e-3	[-]
Gain for P-controller					

C_T_FAC_FCO_FIL_CUS[2]	1	0...FFH	0...25.5	0.1	[s]
Filter constant for FCO filter during custumer corrected fuel mass setpoint					
C_T_FAC_FCO_FIL_ISC[2]	1	0...FFH	0...25.5	0.1	[s]
Filter constant for FCO filter during active idle speed controller in stratified mode					
C_T_RAMP_FCO_FIL	1	0...FFFFH	0...655.35	0.01	[s]
Constant indicating speed of ramping down the FCO-filter					
C_CTR_VS_MIN_FCO	1	0...FFH	0...255	1	[-]
Threshold of a counter (used in FCO calculation)					
C_VS_MIN_FCO	1	0...FFH	0...255	1	[km/h]
Threshold for VS used in FCO calculation					
LC_FCO_MFF_ADD_CYL_CP	1	0...1H	0...1	1	[-]
use Canister fuel flow for FCO calculation					
LC_FCO_COR_REQ_ENA	1	0...1H	0...1	1	[-]
use corrected MFF_SP during NOx Trap regeneration and desulfurization for FCO calculation					
C_FCO_FIL_DIF_FB_MAX	1	80000000... 7FFFFFFFH	-2147483648... 2147483647	1	[µl]
Maximum value of FCO_FIL_DIF_FB					
C_FCO_FIL_DIF_FB_MIN	1	80000000... 7FFFFFFFH	-2147483648... 2147483647	1	[µl]
Minimun value of FCO_FIL_DIF_FB					
C_FAC_MFF_FCO_LAM_AD	1	0...FFH	0...0.99609	3.9063e-3	[-]
weighting factor for using or not of lambda adaptation value for calculation of FCO					
LC_FCO_COR_REQ_PUC	1	0...1H	0...1	1	[-]
Freeze FCO calculation on LV_PUC = 1					
LC_FCO_COR_REQ_PUC_INH	1	0...1H	0...1	1	[-]
Freeze FCO calculation on LV_PUC_REQ = 1 and inhibit LV_PUC					

General information:

The actual fuel consumption value (FCO_AV_1) is calculated from MFF_SP (incl. Lambdaadaptation) or MFF_SP_BAS (without Lam-ad) and several factors. During NOx regeneration additional FCO_AV_2 is calculated. This is not the real fuel mass, but it is used to avoid strong peaks in the FCO_AV signal. If NOx regeneration, desulfurization or the idle speed controller during stratifid mode is active, FCO_AV is additional filtered to avoid high gradients in the FCO signal. To ensure that the integrated fuel consumption (FCO) is still correct despite of the filtering and the use of FCO_AV_2 (-> FCO_FIL_DIF = 0), a balance controller (C_FCO_DIF_GAIN) is activated to balance FCO over a certain time.

Application conditions:

Initialisation: at reset **or** LV_IGK 0->1: all = 0, except FCO_DMTL = 0 only at reset and FCO_FIL_DIF =0 only at reset

Recurrence: every segment,

except calculation of T_RAMP_FCO_FIL and FCO_FIL_DIF_FB, 10ms recurrence

Activation: LV_ES = 0

Formula section:

```
// calculate FCO_AV_1
If      LC_FCO_MFF_ADD_CYL_CP = 1 // use Canister fuel flow for FCO calculation?
then    FCO_MFF_ADD_CYL_CP = MFF_ADD_CYL_CP
else    FCO_MFF_ADD_CYL_CP = 0
```

```
endif
// calculate FCO_AV_1using MFF_SP with Lambdaadatation (C_FAC_MFF_FCO_LAM_AD=1) or without
FCO_AV_1 = (MFF_SP[NC_LAMB_REF[SEG_NR] ] * C_FAC_MFF_FCO_LAM_AD +
            MFF_SP_BAS * (1- C_FAC_MFF_FCO_LAM_AD) +
            FCO_MFF_ADD_CYL_CP) * C_FAC_MFF_FCO * 46 * EFF_SCC_AV *
            (1 + FAC_FCO_KWP)


//calculate a second FCO during regeneration and desulfurization (not real mass fuel flow,
used for filtering strong amplitudes due to regeneration):
If      LV_FCO_COR_REQ = 1          and
        LC_FCO_COR_REQ_ENA = 1
Then    FCO_AV_2 = MFF_SP_FCO_COR * C_FAC_MFF_FCO * 46 * EFF_SCC_AV *
        ( 1 + FAC_FCO_KWP )
        FCO_AV = FCO_AV_2
Else    FCO_AV = FCO_AV_1
        FCO_AV_2 = FCO_AV_1
Endif


// calculate FCO_DMTL and FCO_AV_MMV
$FCO\_DMTL_n$       =   $FCO\_DMTL_{n-1}$ + FCO_AV
$FCO\_AV\_MMV_{(n)}$  =   $FCO\_AV\_MMV_{(n-1)}$
                        + C_CRLC_FCO_AV * (FCO_AV – $FCO\_AV\_MMV_{(n-1)}$ )


// initialize filter factors if requested
If      LV_FCO_COR_REQ = 1
Then    //activation of filter in case of custumer corrected fuel mass setpoint:
        T_FAC_FCO_FIL[i]  =  C_T_FAC_FCO_FIL_CUS[i]
Elseif  (OPM_AV = 1 or OPM_AV = 3)  and  LV_REQ_ISC = 1
Then    //activation of filter in case of active idle speed controller in stratified mode:
        T_FAC_FCO_FIL[i]  =  C_T_FAC_FCO_FIL_ISC[i]
Endif


// calculate filter outputs:
If      T_FAC_FCO_FIL[1] > 0   or   T_FAC_FCO_FIL[2] > 0
Then    if      LV_FCO_COR_REQ = 0              and
                 [(OPM_AV != 1 and OPM_AV != 3)  or  LV_REQ_ISC = 0]
        then    // ramp filter time/ factors -> until FAC_FCO_FIL[i] = 0:
```

```
                if      T_RAMP_FCO_FIL >= C_T_RAMP_FCO_FIL
                then    T_RAMP_FCO_FIL = 0
                        T_FAC_FCO_FIL[i] = T_FAC_FCO_FIL[i] - 0,1s
                else    do nothing
                endif
        Endif
        // calculate filter factor FAC_FCO_FIL from filter time T_FAC_FCO and segment time
        (calculated from N_32):
        FAC_FCO_FIL[i] = 1 / ( 1 + T_FAC_FCO_FIL[i] / (120 / (N_32 * NC_CYL_NR)) )
        // calculate filter outputs:
        $FCO\_AV\_FIL[1]_{(n)}$    =    FAC_FCO_FIL[1]  * FCO_AV
                                        + (1 – FAC_FCO_FIL[1]) * $FCO\_AV\_FIL[1]_{(n-1)}$
        $FCO\_AV\_FIL[2]_{(n)}$    =    FAC_FCO_FIL[2]  * $FCO\_AV\_FIL[1]_{(n)}$
                                        + (1 – FAC_FCO_FIL[2]) * $FCO\_AV\_FIL[2]_{(n-1)}$
Else    FCO_AV_FIL[i]         =    FCO_AV      //converted
Endif


// calculate FCO:
If      [LV_PUC = 1 and LC_FCO_COR_REQ_PUC = 1]                                   or
        // no FCO display on fuel cut off


        [LV_PUC_REQ = 1 and LV_PUC = 0 and LC_FCO_COR_REQ_PUC_INH = 1
                                                                                 and
        (LV_PUC_INH_TEMP_CAT = 1 or LV_PUC_LOCK_TNT = 1)                         and
        LV_LDM_PUC_INH = 0 and LV_DCC_PUC_INH = 0 and LV_AMT_ACT = 0 and
        LV_PUC_SA_INH = 0 and LV_DCC_INC_ACT = 0 and LV_GS_INC_ACT = 0
        and LV_INH_PUC_CUS = 0]
        // no FCO display on requested but inhibited fuel cut off
Then
        // fuel consumption display in instrument cluster is deactived
        $FCO_{(n)} = FCO_{(n-1)}$
        $FCO\_FIL\_DIF_{(n)} = FCO\_FIL\_DIF_{(n-1)} + FCO\_AV\_1$
Else
    If      abs( FCO_FIL_DIF) > C_FCO_DIF_MAX     or
            LV_FCO_COR_REQ = 1                    or
            [(OPM_AV = 1 or OPM_AV = 3)  and  LV_REQ_ISC = 1]
    Then
```

```
        //check speed of vehicle, in case VS < threshold -> fuel consumption display in
        instrument cluster is deactived -> thus complete adding of FCO difference
        (FCO_FIL_DIF) to FCO signal can be done:
        IF      VS < C_VS_MIN_FCO
        THEN    CTR_VS_MIN_FCO ++
        ELSE    CTR_VS_MIN_FCO = 0
        ENDIF
        // for the following calculations FCO_AV_FIL[2] has to be converted!
        IF      CTR_VS_MIN_FCO <= C_CTR_VS_MIN_FCO
        THEN
                // calculate integrated fuel consumption with filter output and controller
                output:
                $FCO_{(n)} = FCO_{(n-1)} + FCO\_AV\_FIL[2] + FCO\_FIL\_DIF\_FB$

                //calculate fuel difference between FCO_AV_1 and filter+controller output
                (integrated):
                $FCO\_FIL\_DIF_{(n)} = FCO\_FIL\_DIF_{(n-1)}$
                                    $+ [ FCO\_AV\_1 - FCO\_AV\_FIL[2] ]$
                                    $- FCO\_FIL\_DIF\_FB$

                //reset of FCO_FIL_DIF_FB:
                FCO_FIL_DIF_FB = 0
        ELSE
                // calculate integrated fuel consumption with FCO_AV_1 (real FCO) and
                add complete FCO difference:
                $FCO_{(n)} = FCO_{(n-1)} + FCO\_AV\_1 + FCO\_FIL\_DIF$

                //reset FCO_FIL_DIF:
                $FCO\_FIL\_DIF_{(n)} = 0$

                //reset of FCO_FIL_DIF_FB:
                FCO_FIL_DIF_FB = 0
        ENDIF

Else    // calculate integrated fuel consumption with FCO_AV:
        $FCO_{(n)} = FCO_{(n-1)} + FCO\_AV$
        // initialize values:
        FCO_FIL_DIF     = 0
```

```
    Endif
Endif
```

Calculation of Timer T_RAMP_FCO_FIL (calculation with 10ms recurrence!):

```
If      T_RAMP_FCO_FIL < 655,35 s
Then    T_RAMP_FCO_FIL = T_RAMP_FCO_FIL + 0,01s
Else    do nothing
Endif
```

Calculation of FCO_FIL_DIF_FB (calculation with 10ms recurrence!):

```
IF      abs(FCO_FIL_DIF) > 0
Then    if      abs(FCO_FIL_DIF) < C_FCO_FIL_DIF_FB_MIN
        Then
                FCO_FIL_DIF_FB = FCO_FIL_DIF
        Else
                //calculate FCO_FIL_DIF_FB and apply min/max check:
                FCO_FIL_DIF_FB_1 = min(C_FCO_FIL_DIF_FB_MAX,
                                        abs(FCO_FIL_DIF)* C_FCO_DIF_GAIN)
                FCO_FIL_DIF_FB   = max(C_FCO_FIL_DIF_FB_MIN,
                                                  FCO_FIL_DIF_FB_1)
                //add correct sign to FCO_FIL_DIF_FB:
                If      FCO_FIL_DIF < 0
                Then    FCO_FIL_DIF_FB =  - FCO_FIL_DIF_FB
                Else    unchanged
                Endif
        Endif
Endif
```

33.8 Cus adap module: FMSP

33.8.1 Outputs for BMW functions which are defined as FMSP exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_dosp_akt	O	0...FFH	0...255	1	[-]
Injection mode requested					
B_wst	O	0...1H	0...1	1	[-]
Auxiliary function "Re-start"					
F_start	O	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
cold start enrichment					
F_vst	O	0...FFFFH	0...63.99902	0.9766e-3	[-]
mixture enrichment due to warm-up and cold after start enrichment					
Rk_1	O	0...FFFFH	0...31.99951	0.4883e-3	[-]
relative fuel mass cylinder bank i					
Rk_2	O	0...FFFFH	0...31.99951	0.4883e-3	[-]
relative fuel mass cylinder bank i					
Rk_kh_s	O/V	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Mass fuel flow of the post injection during CH					
Rk2_kh_h	O/V	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
relative fuel mass cylinder bank i					

Input data:

C_MFF_REF	ECU_STATE	FAC_ST_AMP	FAC_ST_REST
INJ_MOD_GLOBAL	LV_REST	MFF_MPLP_CH	MFF_SP_2_MPLH_CH
MFF_SP_HOM[NC_CBK_EX_NR]	NC_INJ_MOD_MASK_2	NC_INJ_MOD_SINGLE	TI_CAST
TI_WUP			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

i = 1, 2 (index cylinder bank)

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: at reset **and** at exit ECU_STATE "PWL":
0, except F_vst = 1.00

Recurrence : segment : (Mk_hom_ext_1,Mk_hom_ext_2)
(Mk_hom_ext), B_dosp_akt, F_vst,Rk_i, Rk2_kh_h, Rk2_kh_s

10 ms: B_wst, F_start
At "First valid tooth" and before module M904B (High pressure pump control)

general specification

Activation: every engine state

Deactivation: -

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
if  ECU_STATE "PWL"
then

    B_dosp_akt      = 0
    B_wst           = 0
    F_start         = 0
    F_vst           = 1.00
    Rk_1            = 0
    Rk_2            = 0
    Rk2_kh_h        = 0
    Rk_kh_s         = 0

else

        B_wst   =   LV_REST

      If    ((INJ_MOD_GLOBAL & NC_INJ_MOD_MASK_2) == NC_INJ_MOD_SINGLE)
      Then     B_dosp_akt = 0          (single injection activ)
      Else     B_dosp_akt = 1          (multi injection activ)
      Endif

    F_vst  =  TI_WUP + TI_CAST
    $Rk\_i = \frac{MFF\_SP\_HOM\_i}{C\_MAF\_REF * C\_MFF\_FAC}$

    Umrechnung von SVDO [mg/str] in [%] mit:
    Rk2_kh_h = MFF_SP_2_MPLH_CH / C_MFF_REF*100%
    Rk_kh_s = MFF_MPLP_CH / C_MFF_REF*100%

    F_start = FAC_ST_REST * FAC_ST_AMP *100%
endif
```

33.8.2 Outputs for SV-aggregates

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_1_HOM_COR[NC_CYL_NR]	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
MFF additive correction on 1st. injection pulse (homogeneous modus)					
FAC_1_S_COR[NC_CYL_NR]	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
MFF additive correction on 1st. injection pulse (stratified modus)					
FAC_2_HOM_COR[NC_CYL_NR]	V	0...FFFFH	0...1	0.0153e-3	[-]
MFF additive correction on 2nd. injection pulse (homogeneous modus)					
FAC_2_S_COR[NC_CYL_NR]	V	0...FFFFH	0...1	0.0153e-3	[-]
MFF additive correction on 2nd. injection pulse (stratified modus)					
FAC_3_HOM_COR[NC_CYL_NR]	V	0...FFFFH	0...1	0.0153e-3	[-]
MFF additive correction on 3rd. injection pulse (homogeneous modus)					
FAC_3_S_COR[NC_CYL_NR]	V	0...FFFFH	0...1	0.0153e-3	[-]
MFF additive correction on 3rd. injection pulse (stratified modus)					
FAC_MFF_COR_INJ_MOD	O/V	0...FFH	0...1.99218	0.0078125	[-]
Mass fuel flow correction at different homogeneous injection mode					
FAC_MFF_COR_INJ_MOD_S	O/V	0...FFH	0...1.99218	0.0078125	[-]
Mass fuel flow correction at different injection stratified mode					
FAC_MFF_CST_OPM_SEL	O/V	0...FFH	0...0.99609	3.9062e-3	[-]
start fuel mass flow interpolation factor for operation switch manager					
FAC_MFF_WUP_HOMS	O/V	0...FFH	0...1.99218	0.0078125	[-]
Mass fuel flow correction during homogeneous lean warm-up phase					
FAC_MFF_WUP_S	O/V	0...FFH	0...1.99218	0.0078125	[-]
Mass fuel flow correction stratified warm-up phase					
FAC_MPLP_COR[NC_CYL_NR]	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
MFF additive correction on post injection pulse (stratified modus)					
FAC_TI_CAST_OPM_SEL	O/V	0...FFH	0...0.99609	3.9062e-3	[-]
post start injection time correction interpolation factor for operation switch manager					
FAC_TI_WUP_OPM_SEL	O/V	0...FFH	0...0.99609	3.9062e-3	[-]
warm up injection time correction interpolation factor for operation switch manager					
INJ_MOD_HOM_REQ	O/V	0...FFH	0...255	1	[-]
injection mode selection hom (single / multi)					
INJ_MOD_S_REQ	O/V	0...FFH	0...255	1	[-]
injection mode selection strat (single / multi)					
LV_ADD_PULSE_ENA	O/V	0...1H	0...1	1	[-]
Switch to enable additional injection pulse at homogeneous mode, single injection mode (injection update at transient operation)					
LV_EOI_ST_ENA	O/V	0...1H	0...1	1	[-]
Indicates if an EOI based start is requested					
LV_FCO_COR_REQ	O/V	0...1H	0...1	1	[-]
Anforderung Verbrauchsberechnung mit korrigierter Kraftstoffmasse					
LV_ST_H_PRS	-	0...1H	0...1	1	[-]
Flag indicating high pressure start requested					
LV_ST_INJ_AUTH	O/V	0...1H	0...1	1	[-]
Flag indicating injection authorized					
LV_TI_COR_WF_OPM_1_ACT	O/V	0...1H	0...1	1	[-]
wall film operation mode switch					
MFF_ADD_WF	O/V	8000...7FFFH	-694.5... 694.47880	0.0211945	[mg/stk]
Total fuel amount for wall film compensation					
MFF_SP_1_EXT_COR[NC_CYL_NR]	O/V	8000...7FFFH	-694.5... 694.47880	0.0211945	[mg/stk]
Fuel mass setpoint external correction for the first pulse (aging and coding corr.)					
MFF_SP_1_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual mass fuel flow setpoint for homogeneous mode, first pulse					
MFF_SP_1_HOMS[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]

Fuel mass setpoint for the first pulse at homogeneous-stratified mode					
MFF_SP_1_S[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the first pulse at stratified mode					
MFF_SP_1_S_TMP	-	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the third pulse at stratified mode, for temporary calculations					
MFF_SP_2_EXT_COR[NC_CYL_NR]	O/V	8000...7FFFH	-694.5... 694.47880	0.0211945	[mg/stk]
Fuel mass setpoint external correction for the second pulse (aging and coding corr.)					
MFF_SP_2_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the second pulse at homogeneous mode					
MFF_SP_2_HOMS[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the second pulse at homogeneous-stratified mode					
MFF_SP_2_S[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the second pulse at stratified mode					
MFF_SP_2_S_TMP	-	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the second pulse at stratified mode, for temporary calculations					
MFF_SP_3_EXT_COR[NC_CYL_NR]	O/V	8000...7FFFH	-694.5... 694.47880	0.0211945	[mg/stk]
Fuel mass setpoint external correction for the third pulse (aging and coding corr.)					
MFF_SP_3_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the third pulse at homogeneous mode					
MFF_SP_3_HOMS[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the second pulse at homogeneous-stratified mode					
MFF_SP_3_S[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the third pulse at stratified mode					
MFF_SP_BAS	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Momentary fuel mass setpoint without adaptation					
MFF_SP_BAS_S	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Momentary stratified fuel mass setpoint without adaptation					
MFF_SP_FCO_COR	O/V	8000...7FFFH	-694.5... 694.47880	0.0211945	[mg/stk]
Korrigierte Kraftstoffmasse für Kraftstoffverbrauchsanzeige					
MFF_SP_HOM_BAS[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint for homogeneous mode (exhaust gas bank specific)					
MFF_SP_HOM_MPG_TMP[NC_NR_IV_PLS]	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint for homogeneous mode (temporary value) for FUMU.					
MFF_SP_HOM_TMP[NC_NR_IV_PLS]	-	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint for homogeneous mode (temporary value).					
MFF_SP_HOMS	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass flow setpoint at homogeneous-stratified combustion mode					
MFF_SP_HOMS_PRED	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass flow setpoint if homogeneous-stratified combustion mode would be active					
MFF_SP_MPLP[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the post pulse at stratified mode					
MFF_SP_MPLP_EXT_COR[NC_CYL_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint external correction for the post pulse (aging and coding corr.)					
MFF_SP_S	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass in stratified combustion					
MFF_SP_S_PRED	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Predicted mass fuel flow setpoint if stratified combustion mode would be active					
MFF_SP_S_SWI_HOM	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow expected after switching from strattified to homogeneous mode					
NR_INJ_PLS_HOM_REQ	O/V	0...3H	0...3	1	[-]
Number of requested injection pulses for homogeneous mode					
NR_INJ_PLS_HOM_REQ_TMP	-	0...FFH	0...255	1	[-]
Number of requested injection pulses for homogeneous mode, for temporary calculations					
NR_INJ_PLS_S_REQ	O/V	0...3H	0...3	1	[-]
Number of requested injection pulses for stratified or homogeneous-stratified mode					
LF_MFF_HOM	O/V	0...FFH	0...255	1	[-]

Bitfield to indicate homogeneous injection mapping and ignition coupling.

Input data:

B_espr_start	B_hd_start	B_rkkor_kva	Ba_switch_uek
C_INJ_MOD_SP_MAN	CTR_CYL_NR_ST_CLC	CTR_CYL_NR_STOP_CLC	Espr_mod_soll[NC_CYL_NR]
F_nst_ba	F_rkkorwl_s	F_rkkorwl_hs	F_st_ba
F_wl_ba	FAC_LAM_AD_BAL[NC_CBK_EX_NR]	FAC_LAM_ADJ_COR_LAM_AD_CUS[NC_CBK_EX_NR]	LC_INJ_MOD_SP_MAN_ENA
LV_CH	LV_HOM_AFL_ACT	LV_IGA_AND_INJ_SWI_HOMS	LV_S_ACT
LV_S_CLC	LV_ST_END	MFF_ADD_BAL[NC_CYL_NR]	MFF_ADD_BAL_EXT[NC_CYL_NR]
MFF_ADD_LAM_AD_INJ[NC_CYL_NR]	MFF_LAM_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	MFF_SP_MV	Mk_soll_h[NC_CYL_NR][NC_NR_IV_PLS]
	Mk_soll_s	Mkzyl_soll_hs[NC_CYL_NR][NC_NR_IV_PLS]	N
NC_INJ_MOD_MULTI	NC_INJ_MOD_SINGLE	NC_LAMB_REF	NC_NR_IV_PLS
Rk_add_wf	Rk_kor_h	Rk_kor_kva	Rk_korres_s
Rk_korres_hs	SEG_NR		

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset: 0

```
Expect:
INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE
INJ_MOD_S_REQ = NC_INJ_MOD_SINGLE
FAC_MFF_WUP_S = 1
FAC_MFF_WUP_HOMS = 1
FAC_MFF_COR_INJ_MOD = 1
FAC_MFF_COR_INJ_MOD_S = 1
For x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
    FAC_1_S_COR[x] = 1
    FAC_2_S_COR[x] = 0
    FAC_3_S_COR[x] = 0
    FAC_MPLP_COR[x] = C_FAC_MPLP_COR
Endfor
```

Recurrence :	LV_ST_END = 0:	10 ms
	LV_ST_END = 1:	segment synchronous

Note: Please pay attention before integrating this function in the SW. This function shall be calculated before the module 'Calculation of MFF setpoint'.

CTR_CYL_NR_ST_CLC , CTR_CYL_NR_STOP_CLC and LV_S_CLC must be calculated before calculation of this module

Any other order of calculation should not be changed

FAC_MFF_CST_OPM_SEL,

FAC_TI_CAST_OPM_SEL,

FAC_TI_WUP_OPM_SEL

Expect: 10ms : MFF_SP_HOMS_PRED, MFF_SP_S_PRED

Activation: every engine state

Deactivation: -

Formula section:

Remark: all formulas are valid in a **physical** meaning

LV_FCO_COR_REQ = B_rkkor_kva
MFF_SP_FCO_COR = Rk_kor_kva * C_MAF_REF * C_MFF_FAC

MFF_ADD_WF = Rk_add_wf*C_MAF_REF*C_MFF_FAC

MFF_SP_S_SWI_HOM = Rk_kor_h * C_MAF_REF * C_MFF_FAC

FAC_MFF_WUP_S = F_rkkorwl_s
FAC_MFF_WUP_HOMS = F_rkkorwl_hs

```
IF LC_MFF_MPG_NEW = 1
THEN
    Only element 0 is used to indicate injection mode
    Bit 0 of LF_MFF_HOM = Bit 0 of Espr_mod_soll[0]
    Bit 1 of LF_MFF_HOM = Bit 2 of Espr_mod_soll[0]
    Bit 2 of LF_MFF_HOM = Bit 3 of Espr_mod_soll[0]
    Bit 3 of LF_MFF_HOM = Bit 4 of Espr_mod_soll[0]
    Bit 4 of LF_MFF_HOM = Bit 5 of Espr_mod_soll[0]
    Note: Bits 0, 1 and 2 of LF_MFF_HOM[x] state that the fuel mass on the respective pulse is unequal to zero (=1) or equal to zero (=0), which means that this pulse shall not be carried out.
    Bits 3 and 4 state that pulse 2 resp. pulse 3 are ignition coupled (=1) or not (=0).
ENDIF
```

NR_INJ_PLS_HOM_REQ_TMP = 0

```
If    LC_MFF_SP_HOM_MAN_ACT = 1
Then
      For   x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
      MFF_SP_1_HOM[x]   = C_MFF_SP_1_HOM_MAN
      MFF_SP_2_HOM[x]   = C_MFF_SP_2_HOM_MAN
      MFF_SP_3_HOM[x]   = C_MFF_SP_3_HOM_MAN
      Endfor

      IF MFF_SP_1_HOM[0] > 0
      THEN
            NR_INJ_PLS_HOM_REQ_TMP = NR_INJ_PLS_HOM_REQ_TMP OR 0x1   ...bitwise
      ENDIF
      IF MFF_SP_2_HOM[0] > 0
      THEN
            NR_INJ_PLS_HOM_REQ_TMP = NR_INJ_PLS_HOM_REQ_TMP OR 0x2   ...bitwise
      ENDIF
      IF MFF_SP_3_HOM[0] > 0
      THEN
            NR_INJ_PLS_HOM_REQ_TMP = NR_INJ_PLS_HOM_REQ_TMP OR 0x4   ...bitwise
      ENDIF
Else
      For   x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC

      IF LC_MFF_MPG_NEW = 1
      THEN

         IF LV_ST_END = 0 AND (LF_MFF_HOM BitwiseAND 7h = 4) // only third bit is set
         THEN – high pressure start with fuel mass on pulse 3
            MFF_SP_1_HOM[x]    =       0
            MFF_SP_2_HOM[x]    =       Mk_soll_h[x][2]  // zero based indexing
            MFF_SP_3_HOM[x]    =       0
         ELSE
            IF LV_ST_END = 0 AND (LF_MFF_HOM BitwiseAND 7h = 2)
                                                         // only second bit is set
            THEN – high pressure start with fuel mass on pulse 2
                    MFF_SP_1_HOM[x]    =       0
                    MFF_SP_2_HOM[x]    =       Mk_soll_h[x][1]  // zero based indexing
                    MFF_SP_3_HOM[x]    =       0
            ELSE
                    tmp = 0  // Note: tmp is a temporary counter counting from 0 to 3.
                    MFF_SP_HOM_MPG_TMP[ ] = 0      // all NC_NR_IV_PLS elements

                    FOR i = 0 TO NC_NR_IV_PLS – 1 DO

                          IF Bit i of LF_MFF_HOM = 1
                          THEN
                              MFF_SP_HOM_MPG_TMP [tmp] = Mk_soll_h[x][i]
                              tmp = tmp + 1
                          ENDIF
                    ENDFOR

                    MFF_SP_1_HOM[x]    = MFF_SP_HOM_MPG_TMP [0]
                    MFF_SP_2_HOM[x]    = MFF_SP_HOM_MPG_TMP [1]
                    MFF_SP_3_HOM[x]    = MFF_SP_HOM_MPG_TMP [2]
```

```
                ENDIF
            ENDIF
        ELSE

                MFF_SP_1_HOM[x]    =      Mk_soll_h[x][0]
                MFF_SP_2_HOM[x]    =      Mk_soll_h[x][1]
                MFF_SP_3_HOM[x]    =      Mk_soll_h[x][2]

        ENDIF

        MFF_SP_HOM_BAS[[NC_LAMB_REF[bit x]] new =
                MFF_SP_HOM_BAS[[NC_LAMB_REF[bit x]] +
                (MFF_SP_1_HOM[x] + MFF_SP_2_HOM[x] + MFF_SP_3_HOM[x])


        IF MFF_SP_1_HOM[x] > 0
        THEN
                NR_INJ_PLS_HOM_REQ_TMP = NR_INJ_PLS_HOM_REQ_TMP OR 0x1   ...bitwise
        ENDIF
        IF MFF_SP_2_HOM[x] > 0
        THEN
                NR_INJ_PLS_HOM_REQ_TMP = NR_INJ_PLS_HOM_REQ_TMP OR 0x2   ...bitwise
        ENDIF
        IF MFF_SP_3_HOM[x] > 0
        THEN
                NR_INJ_PLS_HOM_REQ_TMP = NR_INJ_PLS_HOM_REQ_TMP OR 0x4   ...bitwise
        ENDIF
        Endfor
    Endif
    % mean homogeneous mass fuel flow set point (sum over all pulses) assigned to each
    exhaust gas bank, without any correction. or adaptation
    MFF_SP_HOM_BAS[[NC_LAMB_REF[bit x]] new =
        MFF_SP_HOM_BAS[[NC_LAMB_REF[bit x]] / (NC_CYL_NR / NC_CBK_EX_NR)

    % Fuel mass setpoint of current cylinder [x] without adaptation for the fuel consumption
    calculation
            MFF_SP_BAS = MFF_SP_1_HOM[SEG_NR] + MFF_SP_2_HOM[SEG_NR] +
            MFF_SP_3_HOM[SEG_NR]

    IF NR_INJ_PLS_HOM_REQ_TMP > 3
    THEN
        INJ_MOD_HOM_REQ = NC_INJ_MOD_MULTI
        NR_INJ_PLS_HOM_REQ = 3
        IF (LC_FAC_MFF_HOM_COR_ENA = 1)
            FAC_MFF_COR_INJ_MOD = IP_FAC_MFF_COR_MPLH_3(MFF_SP_BAS, N)
        ELSE
            FAC_MFF_COR_INJ_MOD = 1
        ENDIF

    ELSE
        IF NR_INJ_PLS_HOM_REQ_TMP = 3
        THEN
```

```
            INJ_MOD_HOM_REQ = NC_INJ_MOD_MULTI
            NR_INJ_PLS_HOM_REQ = 2
            IF ( LC_FAC_MFF_HOM_COR_ENA = 1 AND
                ( LV_CH =0 OR LC_MFF_COR_CH_ENA =1) AND
                ( LV_ST_END=1 OR LC_MFF_COR_ST_ENA=1) )
                FAC_MFF_COR_INJ_MOD = IP_FAC_MFF_COR_MPLH(MFF_SP_BAS, N)
            ELSE
              FAC_MFF_COR_INJ_MOD = 1
             ENDIF
        ELSE
            INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE
            NR_INJ_PLS_HOM_REQ = 1
            IF (LC_FAC_MFF_HOM_COR_ENA = 1 AND
                ( LV_ST_END=1 OR LC_MFF_COR_ST_ENA=1) )
            THEN
                FAC_MFF_COR_INJ_MOD = IP_FAC_MFF_COR_SNGH(MFF_SP_BAS, N)
            ELSE
                FAC_MFF_COR_INJ_MOD = 1
            ENDIF

        ENDIF
    ENDIF

    If(4) LV_S_CLC = 1
    Then(4)
        IF      LC_MFF_SP_HOMS_MAN_ACT = 1
            Then
                    For x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
                    MFF_SP_1_HOMS[x]  =         C_MFF_SP_1_HOMS_MAN
                    MFF_SP_2_HOMS[x]  =         C_MFF_SP_2_HOMS_MAN
                    MFF_SP_3_HOMS[x]  =         C_MFF_SP_3_HOMS_MAN
                    Endfor
            Else
                    For x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
                    MFF_SP_1_HOMS[x]  =         Mkzyl_soll_hs[x][0]
                    MFF_SP_2_HOMS[x]  =         Mkzyl_soll_hs[x][1]
                    MFF_SP_3_HOMS[x]  =         Mkzyl_soll_hs[x][2]
                    Endfor
                    MFF_SP_HOMS = MFF_SP_1_HOMS[x] +MFF_SP_2_HOMS[x] +
                                                              MFF_SP_3_HOMS[x]
                                                (x = latest segment)
        Endif

        If      LC_MFF_SP_S_MAN_ACT = 1
            Then
                    For x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
                    MFF_SP_1_S[x]         = C_MFF_SP_1_S_MAN
                    MFF_SP_2_S[x]         = C_MFF_SP_2_S_MAN
                    MFF_SP_3_S[x]         = C_MFF_SP_3_S_MAN
                    MFF_SP_MPLP[x]        = C_MFF_SP_MPLP_MAN
                    Endfor
```

```
            MFF_SP_S =    C_MFF_SP_1_S_MAN + C_MFF_SP_2_S_MAN +
                          C_MFF_SP_3_S_MAN + C_MFF_SP_MPLP_MAN
        Else
            For x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
            MFF_SP_1_S[x]        =       Mkzyl_soll_s[x][0]
            MFF_SP_2_S[x]        =       Mkzyl_soll_s[x][1]
            MFF_SP_3_S[x]        =       Mkzyl_soll_s[x][2]
            MFF_SP_MPLP[x]       =       Mkzyl_soll_s[x][3]
            Endfor
            MFF_SP_S      =       Mk_soll_s

    Endif


    If      ((LC_INJ_MOD_SP_MAN_ENA = 0) and (LV_IGA_AND_INJ_SWI_HOMS = 1))
            or
            ((LC_INJ_MOD_SP_MAN_ENA = 1) and (    C_INJ_MOD_SP_MAN = 'HOMS' or
                                                  C_INJ_MOD_SP_MAN = 'HOMS-SNG' or
                                                  C_INJ_MOD_SP_MAN = 'HOMS+PLS3'))
    Then
        MFF_SP_S_TMP = 0

        For x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
        MFF_SP_1_S[x]        =       MFF_SP_1_HOMS[x]
        MFF_SP_2_S[x]        =       MFF_SP_2_HOMS[x]
        MFF_SP_3_S[x]        =       MFF_SP_3_HOMS[x]
        MFF_SP_MPLP[x] = 0
        MFF_SP_S_TMP=     MFF_SP_S_TMP + MFF_SP_1_HOMS[x] +
                          MFF_SP_2_HOMS[x] + MFF_SP_3_HOMS[x]


        Endfor
        MFF_SP_S = MFF_SP_S_TMP /
                (CTR_CYL_NR_STOP_CLC - CTR_CYL_NR_ST_CLC + 1)

    Endif


    % Fuel mass setpoint of current cylinder [x] without adaptation for the fuel consumption
    calculation
    IF LV_S_ACT = 1 OR LV_HOM_AFL_ACT = 1
    THEN
            MFF_SP_BAS = MFF_SP_1_S[SEG_NR] + MFF_SP_2_S[SEG_NR] +
            MFF_SP_3_S[SEG_NR]
    ENDIF

    MFF_SP_BAS_S = MFF_SP_1_S[SEG_NR] + MFF_SP_2_S[SEG_NR] +
            MFF_SP_3_S[SEG_NR]

Endif(4)
```

Injection mode selection: at BMW the main pulse are the 2nd pulse if there is no 1st pulse the 2nd pulse will copied to the 1st puls:

```
If(1)       LV_ST_END = 1
Then(1)
At start end:
MFF_SP_1_S_TMP      =  0
MFF_SP_2_S_TMP      =  0
   If(2)    LC_MFF_SP_HOM_MAN_ACT = 1
   Then(2)
     For     x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
         Check, if all fuel masses for the first and the second pulse strat. Mode are zero
        MFF_SP_1_S_TMP      = MFF_SP_1_S_TMP OR MFF_SP_1_S[x]
                                                                              ...bitwise
        MFF_SP_2_S_TMP      = MFF_SP_2_S_TMP OR MFF_SP_2_S[x]
                                                                              ...bitwise
     Endfor
   Else(2)

     For     x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
        % distribution factors of MFF correction between the injection pulses
        MFF_SP_HOM_TMP = MFF_SP_1_HOM[x]+ MFF_SP_2_HOM[x]+
                                 MFF_SP_3_HOM[x]

        FAC_1_HOM_COR[x] = MFF_SP_1_HOM[x] / MFF_SP_HOM_TMP
        FAC_2_HOM_COR[x] = MFF_SP_2_HOM[x] / MFF_SP_HOM_TMP
        FAC_3_HOM_COR[x] = 1- (FAC_1_COR[x]+ FAC_2_COR[x])


        % MFF correction of 1st. pulse due to Lambda Adaptation and Cylinder Balancing
        MFF_SP_1_HOM[x] =
        (MFF_LAM_ADD_LAM_AD_OUT[NC_LAMB_REF[bit x]] + MFF_ADD_BAL[x]) *
        FAC_1_HOM_COR[x] +
        MFF_SP_1_HOM[x]* (1+ (FAC_LAM_AD_BAL[NC_LAMB_REF[bit x]] / 100%))

        % MFF correction of 1st. pulse due to injector coding , aging predictive correction and
        minimum fuel mass adaptation
        MFF_SP_1_EXT_COR[x] =
                            MFF_ADD_BAL_EXT[x] *
                            IP_FAC_MFF_ADD_BAL_EXT(MFF_SP_1_HOM[x])
                            +
                            MFF_ADD_LAM_AD_INJ[x] *
                            IP_FAC_MFF_ADD_LAM_AD_INJ(MFF_SP_1_HOM[x])

        MFF_SP_1_HOM[x] = MFF_SP_1_HOM[x] * FAC_MFF_COR_INJ_MOD +
                                MFF_SP_1_EXT_COR[x]

        If(3) MFF_SP_2_HOM[x] > 0
        Then(3)

            % 2nd. pulse MFF correction
            MFF_SP_2_HOM[x] =
            (MFF_LAM_ADD_LAM_AD_OUT[NC_LAMB_REF[bit x]] + MFF_ADD_BAL[x]) *
            FAC_2_HOM_COR[x] +
            MFF_SP_2_HOM[x]* (1+ (FAC_LAM_AD_BAL[NC_LAMB_REF[bit x]] / 100%))
```

```
                % MFF correction of 2nd pulse due to injector coding , aging predictive correction
                and minimum fuel mass adaptation
                MFF_SP_2_EXT_COR[x] =
                                MFF_ADD_BAL_EXT[x] *
                                IP_FAC_MFF_ADD_BAL_EXT(MFF_SP_2_HOM[x])
                                +
                                MFF_ADD_LAM_AD_INJ[x] *
                                IP_FAC_MFF_ADD_LAM_AD_INJ(MFF_SP_2_HOM[x])

                MFF_SP_2_HOM[x] = MFF_SP_2_HOM[x] * FAC_MFF_COR_INJ_MOD +
                                  MFF_SP_2_EXT_COR[x]
            Endif(3)

            If(4) MFF_SP_3_HOM[x] > 0
            Then(4)
                % 3rd. pulse MFF correction
                MFF_SP_3_HOM[x] =
                  (MFF_LAM_ADD_LAM_AD_OUT[NC_LAMB_REF[bit x]] + MFF_ADD_BAL[x]) *
                  FAC_3_HOM_COR[x] +
                  MFF_SP_3_HOM[x]* (1+ (FAC_LAM_AD_BAL[NC_LAMB_REF[bit x]] / 100%))

                % MFF correction of 3rd. pulse due to injector coding, aging predictive correction
                and minimum fuel mass adaptation
                MFF_SP_3_EXT_COR[x] =
                  MFF_ADD_BAL_EXT[x] * IP_FAC_MFF_ADD_BAL_EXT(MFF_SP_3_HOM[x])
                  +
                  MFF_ADD_LAM_AD_INJ[x] *
                                IP_FAC_MFF_ADD_LAM_AD_INJ(MFF_SP_3_HOM[x])

                  MFF_SP_3_HOM[x] = MFF_SP_3_HOM[x] * FAC_MFF_COR_INJ_MOD +
                                    MFF_SP_3_EXT_COR[x]
            Endif(4)

            Check, if all fuel masses for the first and the second pulse strat. Mode are zero
            MFF_SP_1_S_TMP    = MFF_SP_1_S_TMP OR MFF_SP_1_S[x]
                                                                    ...bitwise
            MFF_SP_2_S_TMP    = MFF_SP_2_S_TMP OR MFF_SP_2_S[x]
                                                                    ...bitwise

        Endfor
      Endif(2)
    Else(1)
    At start:
    MFF_SP_1_S_TMP    =  0
    MFF_SP_2_S_TMP    =  0
    LV_EOI_ST_ENA = 0

        For    x  = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
            Check, if all fuel masses for the first and the second pulse strat. Mode are zero
            MFF_SP_1_S_TMP    = MFF_SP_1_S_TMP OR MFF_SP_1_S[x]
                                                                    ...bitwise
```

```
            MFF_SP_2_S_TMP      = MFF_SP_2_S_TMP OR MFF_SP_2_S[x]
                                                                  ...bitwise
        Endfor


        If NR_INJ_PLS_HOM_REQ_TMP = 2
        Then (high pressure start)
          LV_EOI_ST_ENA = 1
          For       x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC

                MFF_SP_1_HOM[x]  =        MFF_SP_2_HOM[x]*
                  (1+(FAC_LAM_ADJ_COR_LAM_AD_CUS[NC_LAMB_REF[bit x]] / 100%))

                MFF_SP_2_HOM[x]  =        0
          Endfor
        Else (low pressure start)
          For       x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC

                MFF_SP_1_HOM[x]=MFF_SP_1_HOM[x]*
                  (1+ (FAC_LAM_ADJ_COR_LAM_AD_CUS [NC_LAMB_REF[bit x]] / 100%))
                MFF_SP_2_HOM[x]=MFF_SP_2_HOM[x]*
                  (1+ (FAC_LAM_ADJ_COR_LAM_AD_CUS [NC_LAMB_REF[bit x]] / 100%))
                MFF_SP_3_HOM[x]=MFF_SP_3_HOM[x]*
                  (1+ (FAC_LAM_ADJ_COR_LAM_AD_CUS [NC_LAMB_REF[bit x]] / 100%))
          Endfor
        Endif
    Endif(1)


    If(11)      LV_S_CLC = 1
    Then(11)
        In case of homogeneous-stratified mode, copy the fuel masses for homs to the fuel
        masses for stratified mode (mapping):
        IF(1a) ((LC_INJ_MOD_SP_MAN_ENA = 0) and (LV_IGA_AND_INJ_SWI_HOMS = 1))
               or
               ((LC_INJ_MOD_SP_MAN_ENA = 1) and (C_INJ_MOD_SP_MAN = 'HOMS' or
                                                 C_INJ_MOD_SP_MAN = 'HOMS-SNG' or
                                               C_INJ_MOD_SP_MAN = 'HOMS+PLS3'))
        THEN(1a) homs mode
               IF (LC_FAC_MFF_HOMS_COR_ENA = 1)
                   FAC_MFF_COR_INJ_MOD_S =
                                   IP_FAC_MFF_COR_HOMS(MFF_SP_BAS_S, N)
               ELSE
                   FAC_MFF_COR_INJ_MOD_S = 1
               ENDIF

               IF MFF_SP_2_S_TMP > 0
               THEN homs tripple injection – no pulse mapping
                       INJ_MOD_S_REQ = NC_INJ_MOD_MULTI
                       NR_INJ_PLS_S_REQ = 3
               ELSE
                IF MFF_SP_1_S_TMP > 0
```

```
          THEN homs double injection
                INJ_MOD_S_REQ = NC_INJ_MOD_MULTI
                NR_INJ_PLS_S_REQ = 2
                For     x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
                        MFF_SP_2_S[x] = MFF_SP_3_S[x]
                MFF_SP_3_S[x] =     0
                Endfor
          ELSE homs single injection
                INJ_MOD_S_REQ = NC_INJ_MOD_SINGLE
                NR_INJ_PLS_S_REQ = 1
                For     x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
                MFF_SP_1_S[x] = MFF_SP_3_S[x]
                MFF_SP_2_S[x] =     0
                MFF_SP_3_S[x] =     0
                Endfor
          ENDIF
         ENDIF

ELSE(1a) s mode has to be calculated
          IF        MFF_SP_1_S_TMP > 0
          THEN s tripple injection – no pulse mapping
             INJ_MOD_S_REQ = NC_INJ_MOD_MULTI
             NR_INJ_PLS_S_REQ = 3
             IF (LC_FAC_MFF_S_COR_ENA = 1)
                FAC_MFF_COR_INJ_MOD_S=
                   IP_FAC_MFF_COR_MPLS_3(MFF_SP_BAS_S,N)
             ELSE
                 FAC_MFF_COR_INJ_MOD_S = 1
             ENDIF
          ELSE
                IF (LC_FAC_MFF_S_COR_ENA = 1)
                   FAC_MFF_COR_INJ_MOD_S=
                               IP_FAC_MFF_COR_MPLS(MFF_SP_BAS_S,N)
                ELSE
                   FAC_MFF_COR_INJ_MOD_S = 1
                ENDIF
                IF MFF_SP_2_S_TMP > 0
                THEN s double injection
                        INJ_MOD_S_REQ = NC_INJ_MOD_MULTI
                        NR_INJ_PLS_S_REQ = 2
                        For     x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
                        MFF_SP_1_S[x] = MFF_SP_2_S[x]
                        MFF_SP_2_S[x] = MFF_SP_3_S[x]
                        MFF_SP_3_S[x] =     0
                        Endfor
                ELSE s single injection
                        INJ_MOD_S_REQ = NC_INJ_MOD_SINGLE
                        NR_INJ_PLS_S_REQ = 1
                         For  x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC
                         MFF_SP_1_S[x]      =        MFF_SP_3_S[x]
                          MFF_SP_2_S[x]     =        0
                         MFF_SP_3_S[x]      =        0
```

```
                        Endfor
                    ENDIF
                ENDIF
ENDIF(1a) puls mapping

Statement usefull for ECM2
IF LV_IGA_AND_INJ_SWI = 0
THEN FAC_MFF_COR_INJ_MOD = FAC_MFF_COR_INJ_MOD_S
ENDIF

Corrections for fuel masses due to cylinder balancing
IF(3)       LC_MFF_ADD_AD_LAM_BAL_MAN = 1     AND
            LV_ST_END = 1                     AND
            LC_MFF_SP_S_MAN_ACT = 0
THEN(3)     corrections have to be calculated


    Calculation of correction factors

        For x = CTR_CYL_NR_ST_CLC to CTR_CYL_NR_STOP_CLC

            % MFF correction distribution factors

            IF(5a) LC_FAC_S_COR_ENA = 1
            THEN(5a)
                % Calculation of the additive Lambda Adaption distribution factor over all pulses
                (default value: the additive adaptation considered only on the first pulse).

                IF(5b) LC_FAC_S_COR_MAN = 0
                Then(5b)
                    FAC_1_S_COR[x] = MFF_SP_1_S[x] / (MFF_SP_1_S[x] + MFF_SP_2_S[x]
                    + MFF_SP_3_S[x] + MFF_SP_MPLP[x])

                    FAC_2_S_COR[x] = MFF_SP_2_S[x] / (MFF_SP_1_S[x] + MFF_SP_2_S[x]
                    + MFF_SP_3_S[x] + MFF_SP_MPLP[x])

                    FAC_3_S_COR[x] = MFF_SP_3_S[x] / (MFF_SP_1_S[x] + MFF_SP_2_S[x]
                    + MFF_SP_3_S[x] + MFF_SP_MPLP[x])

                    FAC_MPLP_COR[x] = 1 – (FAC_1_S_COR[x] + FAC_2_S_COR[x] +
                    FAC_3_S_COR[x])
                Else(5b)
                    FAC_1_S_COR[x] = MFF_SP_1_S[x] / (MFF_SP_1_S[x] + MFF_SP_2_S[x]
                    + MFF_SP_3_S[x])

                    FAC_2_S_COR[x] = MFF_SP_2_S[x] / (MFF_SP_1_S[x] + MFF_SP_2_S[x]
                    + MFF_SP_3_S[x])

                    FAC_3_S_COR[x] = MFF_SP_3_S[x] / (MFF_SP_1_S[x] + MFF_SP_2_S[x]
                    + MFF_SP_3_S[x])

                    FAC_MPLP_COR[x] = C_FAC_MPLP_COR
                Endif(5b)
```

```
Endif(5a)
```

Note: In case of a division by zero, the corresponding variable FAC_SP_... should be set to zero!

Calculation of corrected fuel masses due to cylinder balancing:

```
% 1st. pulse MFF correction – first pulse is always calculated
MFF_SP_1_S[x] = (MFF_LAM_ADD_LAM_AD_OUT[NC_LAMB_REF[bit x]] +
MFF_ADD_BAL[x]) * FAC_1_S_COR[x] +
MFF_SP_1_S[x]* (1+ (FAC_LAM_AD_BAL[NC_LAMB_REF[bit x]] / 100%))

% MFF correction of 1st. pulse due to injector coding, aging predictive correction
and minimum fuel mass adaptation
MFF_SP_1_EXT_COR[x] =
                MFF_ADD_BAL_EXT[x] *
                IP_FAC_MFF_ADD_BAL_EXT(MFF_SP_1_S[x])
                +
                MFF_ADD_LAM_AD_INJ[x] *
                IP_FAC_MFF_ADD_LAM_AD_INJ(MFF_SP_1_S[x])

MFF_SP_1_S[x] = MFF_SP_1_S[x] * FAC_MFF_COR_INJ_MOD_S +
                     MFF_SP_1_EXT_COR[x]

IF (NR_INJ_PLS_S_REQ >= 2)
THEN second pulse is calculated in case of double or triple injection
      % 2nd. pulse MFF correction
      MFF_SP_2_S[x] = (MFF_LAM_ADD_LAM_AD_OUT[NC_LAMB_REF[bit x]] +
      MFF_ADD_BAL[x]) * FAC_2_S_COR[x] +
      MFF_SP_2_S[x]* (1+ (FAC_LAM_AD_BAL[NC_LAMB_REF[bit x]] / 100%))

      % MFF correction of 2nd. pulse due to injector, aging predictive correction and
      minimum fuel mass adaptation
         MFF_SP_2_EXT_COR[x] =
                MFF_ADD_BAL_EXT[x] *
                IP_FAC_MFF_ADD_BAL_EXT(MFF_SP_2_S[x])
                +
                MFF_ADD_LAM_AD_INJ[x] *
                IP_FAC_MFF_ADD_LAM_AD_INJ(MFF_SP_2_S[x])

      MFF_SP_2_S[x] = MFF_SP_2_S[x] * FAC_MFF_COR_INJ_MOD_S +
                           MFF_SP_2_EXT_COR[x]
ELSE
      MFF_SP_2_S[x] = 0
ENDIF

IF (NR_INJ_PLS_S_REQ >= 3)
THEN third pulse is calculated in case of triple injection
      % 3rd. pulse MFF correction
      MFF_SP_3_S[x] = (MFF_LAM_ADD_LAM_AD_OUT[NC_LAMB_REF[bit x]] +
      MFF_ADD_BAL[x]) * FAC_3_S_COR[x] +
```

```
                MFF_SP_3_S[x]* (1+ (FAC_LAM_AD_BAL[NC_LAMB_REF[bit x]] / 100%))

                % MFF correction of 3$^{rd}$. pulse due to injector coding , aging predictive
                correction and minimum fuel mass adaptation
                  MFF_SP_3_EXT_COR[x] =
                      MFF_ADD_BAL_EXT[x] *
                      IP_FAC_MFF_ADD_BAL_EXT(MFF_SP_3_S[x])
                      +
                      MFF_ADD_LAM_AD_INJ[x] *
                      IP_FAC_MFF_ADD_LAM_AD_INJ(MFF_SP_3_S[x])

                MFF_SP_3_S[x] = MFF_SP_3_S[x] * FAC_MFF_COR_INJ_MOD_S +
                             MFF_SP_3_EXT_COR[x]

                MFF_SP_MPLP[x] = 0    (No post injection at triple injection!)
            ELSE
                MFF_SP_3_S[x] = 0

                Post pulse is possible in case of no triple injection:
                MFF_SP_MPLP[x] = (MFF_LAM_ADD_LAM_AD_OUT[NC_LAMB_REF[bit x]]
                + MFF_ADD_BAL[x]) * FAC_MPLP_COR[x] +
                MFF_SP_MPLP[x]* (1+ (FAC_LAM_AD_BAL[NC_LAMB_REF[bit x]] / 100%))

                % MFF correction of post pulse due to injector coding, aging predictive
                correction and minimum fuel mass adaptation
                MFF_SP_MPLP_EXT_COR[x] =
                      MFF_ADD_BAL_EXT[x] *
                      IP_FAC_MFF_ADD_BAL_EXT(MFF_SP_MPLP[x])
                      +
                      MFF_ADD_LAM_AD_INJ[x] *
                      IP_FAC_MFF_ADD_LAM_AD_INJ(MFF_SP_MPLP[x])

                MFF_SP_MPLP[x] = MFF_SP_MPLP[x] + MFF_SP_MPLP_EXT_COR[x]
            ENDIF

           Endfor

ENDIF(3)
Endif(11)

If(5)      LC_INJ_MOD_SP_MAN_ENA = 1
Then(5)
           If       (6)     C_INJ_MOD_SP_MAN = 'SNGH'
                    or
                    C_INJ_MOD_SP_MAN = 'DISABLE'
           Then(6)
                    INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE
                    INJ_MOD_S_REQ = NC_INJ_MOD_SINGLE
           Endif(6)
           If(7)    C_INJ_MOD_SP_MAN = 'MPLH'
                    or
                    C_INJ_MOD_SP_MAN = 'MPLH+PLS3'
           Then(7)
```

```
                INJ_MOD_HOM_REQ = NC_INJ_MOD_MULTI
                INJ_MOD_S_REQ = NC_INJ_MOD_SINGLE
        Endif(7)

        If      (8)     (C_INJ_MOD_SP_MAN = 'SNGS'
                or
                C_INJ_MOD_SP_MAN = 'HOMS-SNG')
                and
                LV_S_CLC=1
        Then(8)
                INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE
                INJ_MOD_S_REQ = NC_INJ_MOD_SINGLE
        Endif(8)

        If(9)   (C_INJ_MOD_SP_MAN = 'MPLS'
                or
                C_INJ_MOD_SP_MAN = 'MPLS+PLS3')
                and
                LV_S_CLC=1
        Then(9)
                INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE
                INJ_MOD_S_REQ = NC_INJ_MOD_MULTI
        Endif(9)

        If      (10)    (C_INJ_MOD_SP_MAN = 'HOMS'
                or
                C_INJ_MOD_SP_MAN = 'HOMS+PLS3')
                and
                LV_S_CLC=1
        Then(10)
                INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE
                INJ_MOD_S_REQ = NC_INJ_MOD_MULTI
        Endif(10)
Endif(5)
FAC_MFF_CST_OPM_SEL        =  F_st_ba
FAC_TI_CAST_OPM_SEL        =  F_nst_ba
FAC_TI_WUP_OPM_SEL         =  F_wl_ba
LV_TI_COR_WF_OPM_1_ACT     =  Ba_switch_uek (vice versa)
LV_ST_H_PRS                =  B_hd_start
LV_ST_INJ_AUTH             =  B_espr_start
```

Calculation of LV_ADD_PULSE_ENA

```
IF (C_ADD_PULSE_ENA = 1)
   OR
   ((C_ADD_PULSE_ENA = 2) AND (LV_ST_END = 1))
THEN
      LV_ADD_PULSE_ENA = 1
```

```
ELSE
    LV_ADD_PULSE_ENA = 0
ENDIF



/* with 10ms recurrence */
MFF_SP_HOMS_PRED   = Rk_korres_hs* C_MAF_REF * C_MFF_FAC
MFF_SP_S_PRED      = Rk_korres_s* C_MAF_REF * C_MFF_FAC
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ADD_PULSE_ENA	1	0...2H	0...2	1	[-]
Constant for enabling additional pulses in case of injection time update at transient conditions, homogeneous mode (0 = disable, 1 = enable, 2 = enable at start end)					
C_FAC_MPLP_COR	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Constant defining how quantitatively the additive adaptation value shall be used in the post injection pulse					
C_MAF_REF	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Reference air mass flow for the calculation of MAF and MAF_MAX_COR					
C_MFF_FAC	1	0...FFFFH	0...0.12499	1.9073e-6	[-]
Constant for stoichiometric air/fuel ratio (=1/14.7)					
C_MFF_SP_1_HOM_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the first pulse in homogeneus mode (manual adjustment)					
C_MFF_SP_1_HOMS_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the first pulse in homogeneous-stratified mode (manual adjustment)					
C_MFF_SP_1_S_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the first pulse in stratified mode (manual adjustment)					
C_MFF_SP_2_HOM_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the second pulse in homogeneus mode (manual adjustment)					
C_MFF_SP_2_HOMS_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the second pulse in homogeneous-stratified mode (manual adjustment)					
C_MFF_SP_2_S_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the second pulse in stratified mode (manual adjustment)					
C_MFF_SP_3_HOM_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the third pulse in homogeneous mode (manual adjustment)					
C_MFF_SP_3_HOMS_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the third pulse in homogeneous-stratified mode (manual adjustment)					
C_MFF_SP_3_S_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the third pulse in stratified mode (manual adjustment)					
C_MFF_SP_MPLP_MAN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass setpoint for the post pulse in stratified mode (manual adjustment)					
IP_FAC_MFF_ADD_BAL_EXT	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_MFF_IP_FAC_MFF_ADD_EXT	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Weighting factor for additive aging and coding corrections on MFF					
IP_FAC_MFF_ADD_LAM_AD_INJ	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_MFF_IP_FAC_MFF_LAM_AD_INJ	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Weighting factor for the additive corrections from the function Minimum Fuel Mass Adaptation					
IP_FAC_MFF_COR_HOMS	12*10	0...FFH	0...1.99218	0.0078125	[-]
LDP_MFF_IP_FAC_MFF_COR_HOMS	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_IP_FAC_MFF_COR_HOMS	10	0...1FE0H	0...8160	1	[rpm]
Fuel mass correction factor for operation mode HOMS					
IP_FAC_MFF_COR_MPLH	12*10	0...FFH	0...1.99218	0.0078125	[-]
LDP_MFF_IP_FAC_MFF_COR_MPLH	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_IP_FAC_MFF_COR_MPLH	10	0...1FE0H	0...8160	1	[rpm]
Fuel mass correction factor for injection mode double injection - homogenous					
IP_FAC_MFF_COR_MPLH_3	12*10	0...FFH	0...1.99218	0.0078125	[-]
LDP_MFF_IP_FAC_MFF_COR_MPLH_3	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_IP_FAC_MFF_COR_MPLH_3	10	0...1FE0H	0...8160	1	[rpm]
Fuel mass correction factor for injection mode triple injection - homogenous					
IP_FAC_MFF_COR_MPLS	12*10	0...FFH	0...1.99218	0.0078125	[-]
LDP_MFF_IP_FAC_MFF_COR_MPLS	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_IP_FAC_MFF_COR_MPLS	10	0...1FE0H	0...8160	1	[rpm]
Fuel mass correction factor for injection mode double injection - statified					
IP_FAC_MFF_COR_MPLS_3	12*10	0...FFH	0...1.99218	0.0078125	[-]
LDP_MFF_IP_FAC_MFF_COR_MPLS_3	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_IP_FAC_MFF_COR_MPLS_3	10	0...1FE0H	0...8160	1	[rpm]
Fuel mass correction factor for injection mode triple injection - statified					
IP_FAC_MFF_COR_SNGH	12*10	0...FFH	0...1.99218	0.0078125	[-]
LDP_MFF_IP_FAC_MFF_COR_SNGH	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_IP_FAC_MFF_COR_SNGH	10	0...1FE0H	0...8160	1	[rpm]

Fuel mass correction factor for injection single injection - homogenous					
LC_FAC_MFF_HOM_COR_ENA	1	0...1H	0...1	1	[-]
Switch for activation of fuel mass correction in homogenous mode					
LC_FAC_MFF_HOMS_COR_ENA	1	0...1H	0...1	1	[-]
Switch for activation of fuel mass correction in HOMS mode					
LC_FAC_MFF_S_COR_ENA	1	0...1H	0...1	1	[-]
Switch for activation of fuel mass correction in stratified mode					
LC_FAC_S_COR_ENA	1	0...1H	0...1	1	[-]
Enable additive compensation over all stratified injection pulses					
LC_FAC_S_COR_MAN	1	0...1H	0...1	1	[-]
Manual switch to enable additive compensation of third injection pulse per calibration constant					
LC_MFF_ADD_AD_LAM_OUT_MAN	1	0...1H	0...1	1	[-]
Manual switch to consider lambda adaptation values for MFF_SP_1_S calculation					
LC_MFF_COR_CH_ENA	1	0...1H	0...1	1	[-]
Switch for activation of injection mode dependend fuel mass correction during catalyst heating					
LC_MFF_COR_ST_ENA	1	0...1H	0...1	1	[-]
Switch for activation of injection mode dependend fuel mass correction during start					
LC_MFF_MPG_NEW	1	0...1H	0...1	1	[-]
Switch to activate new fuel mass mapping within layer.					
LC_MFF_SP_HOM_MAN_ACT	1	0...1H	0...1	1	[-]
Switch for manual homogeneus mode active					
LC_MFF_SP_HOMS_MAN_ACT	1	0...1H	0...1	1	[-]
Switch for manual homogeneous-stratified mode active					
LC_MFF_SP_S_MAN_ACT	1	0...1H	0...1	1	[-]
Switch for manual stratified mode active					

34 Fuel supply

Table of Contents

Index

34.1 FUSL Configuration data

Output data:

FUNCTION DESCRIPTION:

34.1.1 Local configuration data ECU with current feedback of VCV via ATIC 39

Data	Value (physical)
NC_V_CUR_VCV_MES	2 [2**H**]

The following data is stored in the flash at end of ECU production and the values will never change during ECU live. The values are used to reduce the tolerances of the current feedback of the ATIC 39 for the VCV current control.

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_CUR_VCV_BOOT_H	1	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Measured voltage at high current					
C_V_CUR_VCV_BOOT_H_CMPL	1	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Measured voltage at high current complement					
C_V_CUR_VCV_BOOT_L	1	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Measured voltage at low current					
C_V_CUR_VCV_BOOT_L_CMPL	1	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Measured voltage at low current complement					

Configuration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
NC_V_CUR_VCV_MES	1	0...2H	0...2	1	[-]
Value to select ECU with VCV shunt or not (0= not shunt, 1= shunt available, 2= feedback from ATIC)					

34.2 Acquisition of fuel pressure (V_FUP)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
V_FUP_SUM	V/O	0...FFFFFFFFH	0... 655359.99984	0.1526e-3	[V]
Accumulated mainfold air pressure measurements for segment number (2n-1), n=1,2,3,...					
V_FUP_SUM_1	V/O	0...FFFFFFFFH	0... 655359.99984	0.1526e-3	[V]
Accumulated mainfold air pressure measurements for segment number (2n), n=1,2,3,...					
CTR_V_FUP	V/O	0...FFFFH	0...65535	1	[-]
Number of samples in buffer V_FUP_SUM					
CTR_V_FUP_1	V/O	0...FFFFH	0...65535	1	[-]
Number of samples in buffer V_FUP_SUM_1					
LV_FUP_SEG_OLD	O	0...1H	0...1	1	[-]
Logical variable indicating the buffer to read the sensor voltage in BSW old value					
V_FUP	V	0...7FFFH	0...4.99984	0.1526e-3	[V]
LPG pressure in the fuel rail, voltage raw value (10 bits).					

Input data:

LV_ES	LV_FUP_SEG		

Import actions:

ACTION_INFR_GetVFupSens(OUT<V_Fup>)
This action gets the FUP sensor voltage

FUNCTION DESCRIPTION:

Signal flow diagram:

General information:

The raw value (voltage) for FUP is measured by continous conversion every 1 ms. The values (10 bit) are summed up in two alternating buffers, V_FUP_SUM or V_FUP_SUM_1. The numbers of values are counted in comparable buffers CTR_V_FUP or CTR_V_FUP_1. This mechanism is necessary to synchronize the measurement and the calculation of FUP_MES (build mean value with a standardized range)

Depending on the logical variable LV_FUP_SEG one of the two buffers — V_FUP_SUM or V_FUP_SUM_1 — and the respective counter — CTR_V_FUP or CTR_V_FUP_1 — are selected for writing, the contents of the other buffer (counter) is used for calculations and diagnosis (after this buffer and the corresponding counter have been read they are cleared).

After having read out one buffer — V_FUP_SUM or V_FUP_SUM_1 — the buffer has to be cleared as well as the corresponding counter — CTR_V_FUP or CTR_V_FUP_1.

A two-buffer-system is used to avoid incorrect FUP calculation.

The buffers will be filled when the segment is changing.

Application conditions:

Initialisation: **At** Reset

After HW reset, the buffers - V_FUP_SUM and V_FUP_SUM_1 - and the corresponding counters - CTR_V_FUP and CTR_V_FUP_1 - have to be cleared.

V_FUP_SUM = 0
V_FUP_SUM_1 = 0
CTR_V_FUP = 0
CTR_V_FUP_1 = 0
LV_FUP_SEG_OLD = 0

Recurrence: 1ms
Activation: -
Deactivation: -

Formula section:

```
If(1) ( LV_FUP_SEG = 0 and LV_FUP_SEG_OLD = 1 )
Then(1)
    V_FUP_SUM =
    CTR_V_FUP = 0
Else(1)
    If(2) ( LV_FUP_SEG =1 and LV_FUP_SEG_OLD = 0 )
    Then(2)
        V_FUP_SUM_1 =
        CTR_V_FUP_1 = 0
    Else(2)
        Do nothing
    Endif(2)
Endif(1)
```

$LV\_FUP\_SEG\_OLD_{(n)} = LV\_FUP\_SEG_{(n-1)}$

```
If(1) ( LV_FUP_SEG = 0 )
Then(1)
    $V\_FUP\_SUM = V\_FUP_{(n)} + V\_FUP_{(n+1)}$
    $CTR\_V\_FUP_{(n)} = CTR\_V\_FUP_{(n-1)} + 1$
Else(1)
    $V\_FUP\_SUM\_1 = V\_FUP_{(n)} + V\_FUP_{(n+1)}$
    $CTR\_V\_FUP\_1_{(n)} = CTR\_V\_FUP\_1_{(n-1)} + 1$
Endif(1)
```

34.3 FUSL - Requirements to Infrastructure (Multi piston pump with PWM controlled volume control valve (VCV))

Input data:

Export actions:

ACTION_INFR_GetElDiagVcv(OUT <Cdn_diag_vcv>, OUT <Err_diag_vcv>**)**
This action reads the failure and condition information for a symptom of the VCV actuator power stage.
ACTION_INFR_SetPwmVcv(IN <ducy>**)**
This action change the dutycycle of the VCV control PWM to increase or reduce fuel amount
ACTION_INFR_GetVFupSens(OUT <v_fup>**)**
This action gets the FUP sensor voltage of high pressure pipe system into the injection rail
ACTION_INFR_SetFrqPwmVcv(IN <frq>**)**
This action sets the frequency of the VCV control PWM
ACTION_INFR_GetVFupEfpSens(OUT <v_fup_efp>**)**
This action gets the FUP sensor voltage of low pressure pipe system to the high pressure pump
ACTION_INFR_GetVVcvCurMes(OUT <v_cur_vcv_mes>**)**
This action gets the shunt voltage from the maximum VCV current peak

Description for actions:

ACTION_INFR_GetElDiagVcv(OUT <Cdn_diag_vcv>, OUT <Err_diag_vcv>) This action reads the failure and condition information for a symptom of VCV power stage. The readout of the power stage is performed autonomous and the information is gathered. When calling this Action the information inside the Infrastructure is reset after returning the OUT parameters.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Cdn_diag_vcv	OUT	0 ... 7H	0 ... 7	1	-
Diagnosis condition for symptom: Bit 0: Set, if diagnosis condition for symptom SCP (SYM_0) is fulfilled Bit 1: Set, if diagnosis condition for symptom SCG (SYM_1) is fulfilled Bit 2: Set, if diagnosis condition for symptom OC (SYM_2) is fulfilled (all combinations are possible)					
Err_diag_vcv	OUT	0 ... 7H	0 ... 7	1	-
Raw value of error symptom: Bit 0: Set, if raw value of error symptom SCP (SYM_0) is set Bit 1: Set, if raw value of error symptom SCG (SYM_1) is set Bit 2: Set, if raw value of error symptom OC (SYM_2) is set (all combinations are possible)					
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

ACTION_INFR_SetPwmVcv(IN <ducy> **)**					
This action change the dutycycle of the VCV control PWM to increase or reduce fuel amount					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
ducy	IN	0…FFFFH	0…99.609375	0.0015259	%
VCV driver PWM duty cycle					

ACTION_INFR_GetVFupSens(OUT <v_fup> **)**					
This action gets the FUP sensor voltage of high pressure pipe system into the injection rail					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
v_fup	OUT	0...7FFFH	0..4.9998474	0.000152587890625	V
FUP high pressure sensor voltage					

ACTION_INFR_SetFrqPwmVcv(IN <frq>**)**					
This action sets the frequency of the VCV control PWM					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
frq	IN	A...3FFH	10...1023	1	Hz
VCV frequency					

ACTION_INFR_GetVFupEfpSens(OUT <v_fup_efp> **)**					
This action gets the FUP sensor voltage of low pressure pipe system to the high pressure pump					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
v_fup_efp	OUT	0...7FFFH	0..4.9998474	0.000152587890625	V
FUP EFP low pressure sensor voltage					

ACTION_INFR_GetVVcvCurMes(OUT <v_cur_vcv_mes>**)**					
This action gets the shunt voltage from the maximum VCV current peak					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
v_cur_vcv_mes	OUT	0...7FFFH	0...4.9998474	0.000152587890625	V
Shunt voltage from VCV					

FUNCTION DESCRIPTION:

General information:

ACTION_INFR_GetElDiagVcv() returns the result of the electrical diagnosis of VCV actuator power stage.

_ The device readout is performed autonomous by the Infrastructure each 10ms.
_ The error informations of every symptom is gathered in the Infrastructure (or-ed symptom) until the Application reads out the information by calling ACTION_INFR_GetElDiagVcv().
_ After having read out the information by calling ACTION_INFR_GetElDiagVcv(), the data inside the Infrastructure is reset. Resetting of Cdn_diag_vcv avoids ambigious results in case of too short calling reccurrence of ACTION_INFR_GetElDiagVcv(): Reset Cdn_diag_vcv indicates, that the gathering of the information is not completely finished.

Requirements for ACTION_INFR_GetElDiagVcv:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cdn_diag_vcv	-	-	<bit coded>	Err_diag_vcv	Diagnosis condition for each symptom **bit 0:** diagnosis condition for symptom SCP (SYM_0) **bit 1:** diagnosis condition for symptom SCG (SYM_1) **bit 2:** diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
Err_diag_vcv	-	-	<bit coded>	Cdn_diag_vcv	Bitcoded result of each symptom **bit 0:** raw value of error symptom SCP (SYM_0) **bit 1:** raw value of error symptom SCG (SYM_1) **bit 2:** raw value of error symptom OC (SYM_2) The relevant bit is set, if the raw error has been detected.

Diagnosis: ACTION_INFR_GetElDiagVcv returns the electric diagnosis for VCV power stage

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_SetPwmVcv:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
ducy	Not relevant	Not relevant	0.0015259%	Not relevant	By the initialization the PWM value set to 0%.

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_GetVFupSens:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
v_fup	Not relevant	0,1 %	0.000152587890625 V	Not relevant	Not relevant

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_SetFrqPwmVcv:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
frq	Not relevant	Not relevant	1 Hz	Not relevant	Not relevant

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_GetVFupEfpSens:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
v_fup_efp	Not relevant	0,1 %	0.000152587890625 V	Not relevant	Not relevant

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

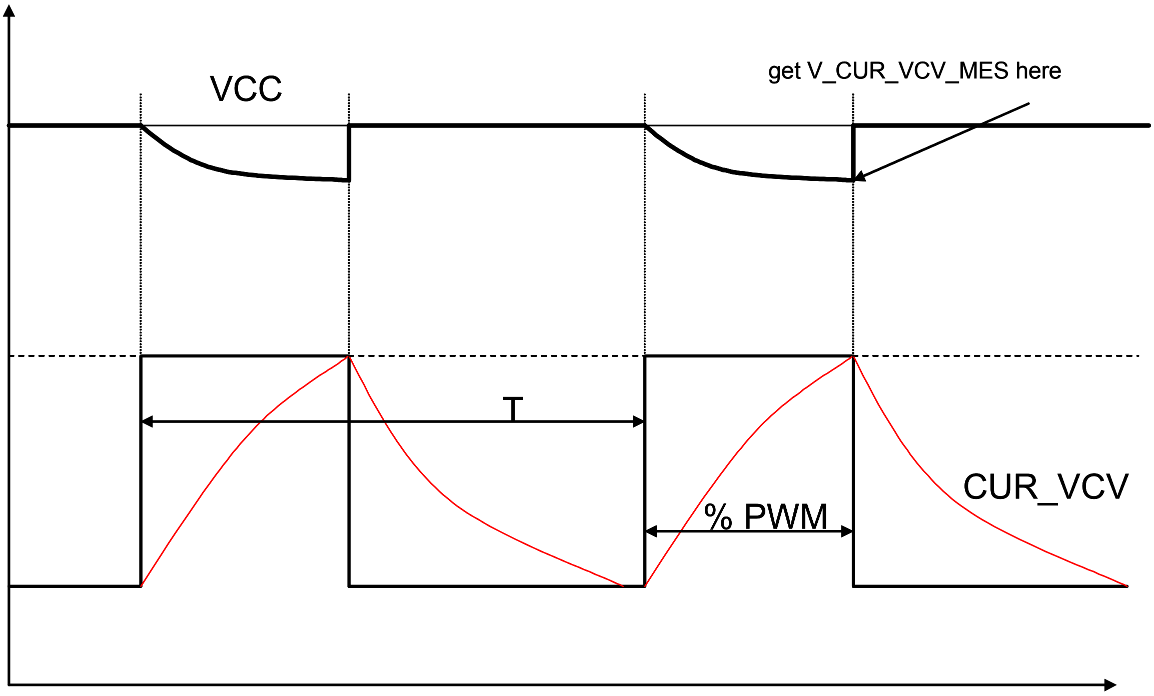
Requirements for ACTION_INFR_ GetVVcvCurMes(OUT <v_cur_vcv_mes>):

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
v_cur_vcv_mes	Not relevant	Not relevant	0.000152587890625 V	Not relevant	Not relevant

Diagnosis: no electrical diagnosis done here

Coincidence requirements:

The current measurement have to capture the maximum current peak nearly from the switch OFF event from the output line driver. Therefore the IO driver have to handle the timing of the measure start event.



For projects with the curent feedback from shunt, the value v_cur_vcv_mes is provided every segment (if segments are available) or every 10ms (if no segments are available).

The PWM signal is updated every segment or ervery 10ms.

For projects with the curent feedback from ATIC 39, the feedback voltage has to be provided at the end of actual duty cycle. See picture below.

The value will be taken every segment (if segments are available) or every 10ms (if no segments are available).

For projects without curent feedback the value v_cur_vcv_mes has to set to 0V.

34.4 AGGR FUSL adaptation

34.4.1 Outputs for SV aggregates, SV internally

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VFF_EFP	V/O	0...FFH	0...255	1	[l/h]
Required amount of fuel					
EFPPWM_CAN	V/O	0...FFH	0...99.60937	0.390625	[%]
Pump speed of the electrical fuel pump as 8 bit PWM signal					
FUP_AD	V/O	8000...7FFFH	-173890... 173884	5.3066911	[hPa]
Adaptation value for fuel pressure adaptation					

Input data:

VFF_EFP_REQ	EFPPWM	

FUNCTION DESCRIPTION:

Adaptation to aggregate environment.

Application conditions:

Initialisation at reset:

VFF_EFP = VFF_EFP_REQ
MIN (VFF_EFP, FEH)
EFPPWM_CAN = EFPPWM
MIN (EFPPWM_CAN, FEH)
FUP_AD = 0

Recurrence : If LV_ST_END == 1
then the recurrency is every segment
If LV_ST_END == 0
then the recurrency is every 10 ms

Activation: every engine operating state

Formula section:

VFF_EFP = VFF_EFP_REQ
MIN (VFF_EFP, FEH)
EFPPWM_CAN = EFPPWM
MIN (EFPPWM_CAN, FEH)
FUP_AD = 0

34.4.2 Miscellaneous

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ENG_OFF_N_CON	O	0...1H	0...1	1	[-]
Logical value engine stop mode active					
LV_PIN_ICH	O	0...1H	0...1	1	[-]
Independent car heater request for electrical fuel pump					
LV_ACR_EFPPWM_TEST_REQ	O	0...1H	0...1	1	[-]
Electrical fuel pump actuator test active					
LV_FUP_EFP_SP_EXT_REQ	O	0...1H	0...1	1	[-]
External request of fuel pressure setpoint (low pressure)					
LV_FUP_LIH_REQ	O	0...1H	0...1	1	[-]
Limp home mode for HPP requested					
NC_NR_N_FUP_SP	O	0...1FH	1...32	1	[-]
Number of engine speed datapoints for FUP_SP					
NC_NR_MFF_SP_FUP_SP	O	0...1FH	1...32	1	[-]
Number of mass fuel flow datapoints for FUP_SP					
FUP_RES_H_SP_CH	O	0...FFFFH	0...255.99609	3.9063e-3	[MPa]
Fuel pressure setpoint during catalyst heating					
MFF_SP_HOM_MV	O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint for homogeneous mode, mean value					
STATE_MPLH_MOD	O	0H 1H 2H 3H	MPLH_OFF MPLH_CH MPLH_OPP MPLH_OHP	1	[-]
State of the multiple injection in hom. mode					
FUP_RNG_H_SP_CH	O	0...FFFFH	0...255.99609	3.9063e-3	[MPa]
Fuel pressure setpoint for catalyst heating					
FUP_RNG_H_SP_OHP	O	0...FFFFH	0...255.99609	3.9063e-3	[MPa]
Fuel pressure setpoint for overheating prevention					
NR_CYL_VVL_H_ACT	O	0...8H	0...8	1	[-]
Number of active cylinder for high mode					
LV_PWL	O	0...1H	0...1	1	[-]
The power-latch phase has started					
LV_LAM_STOP_AE	O	0...1H	0...1	1	[-]
stop-flag for lambda controller					
LV_ERR_RLY_VCV	O	0...1H	0...1	1	[-]
Boolean for error currently present on RLY_VCV command signal					
LV_VCV_RLY	O	0...1H	0...1	1	[-]
Flag for controlling the volume control valve relay					

Input data:

MFF_SP_FUP_CTL	LV_PWL_ACT	LV_ERR_RLY_MAIN	LV_RLY_MAIN

FUNCTION DESCRIPTION:

NC_NR_N_FUP_SP = 8

NC_NR_MFF_SP_FUP_SP = 10

Application conditions:

Initialisation at reset:

LV_ENG_OFF_N_CON = 0
LV_PIN_ICH = 0
LV_ACR_EFPPWM_TEST_REQ = 0
LV_FUP_EFP_SP_EXT_REQ = 0
LV_FUP_LIH_REQ = 0
FUP_RES_H_SP_CH = 0
MFF_SP_HOM_MV = MFF_SP_FUP_CTL
STATE_MPLH_MOD = 0
FUP_RNG_H_SP_CH = 0
FUP_RNG_H_SP_OHP = 0
NR_CYL_VVL_H_ACT = 0
LV_PWL = LV_PWL_ACT
LV_LAM_STOP_AE = 0
LV_ERR_RLY_VCV = LV_ERR_RLY_MAIN
LV_VCV_RLY = LV_RLY_MAIN

Recurrence : 100ms
Activation: *every engine operating state*

Formula section:

LV_PWL = LV_PWL_ACT
LV_ERR_RLY_VCV = LV_ERR_RLY_MAIN
LV_VCV_RLY = LV_RLY_MAIN
MFF_SP_HOM_MV = MFF_SP_FUP_CTL

34.5 Fuel temperature

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TFU	O/V	0...FEH	-48...142.5	0.75	°C
Fuel temperature					
TFU_SUB_STOP	O/V/S	0...FEH	-48...142.5	0.75	°C
Modeled fuel temperature at last engine stop					
TEMP_DIF_TCO_TIA	O/V	0...FEH	-48...142.5	0.75	°C
Temperature difference between TCO_EX and TCO or TIA and TCO					
TFU_IV_SUB	O/V	0...FEH	-48...142.5	0.75	°C
Modeled injektor fuel temperature depending from TCO, N, MFF_SP, and EWP					
TFU_IV	O	0...FEH	-48...142.5	0.75	°C
Injektor fuel temperature depending from TCO, N, MFF_SP, and EWP					
TFU_EFP	O/V	0...FEH	-48...142.5	0.75	°C
Fuel temperature for the low pressure pump					
TFU_IV_SUB_STOP	O/V/S	0...FEH	-48...142.5	0.75	°C
Modeled injection valve temperature TFU_IV at last engine stop, stored in NVMY					
TFU_LPF_TCO	V	0...FEH	-48...142.5	0.75	°C
Modeled fuel temperature with low pass filter					
TFU_RAW	V	0...FEH	-48...142.5	0.75	°C
Modeled fuel temperature raw value					
TFU_IV_SUB_BAS	V	0...FEH	-48...142.5	0.75	°C
Modeled injector fuel temperature depending an TCO, IGA_AV and MFF_SP_MV					
TFU_IV_SUB_RAW	V	0...FEH	-48...142.5	0.75	°C
Modeled injektor fuel temperature depending from TCO, N, MFF_SP, and EWP, raw-value after first low pass filter					
TFU_IV_ST	V	0...FEH	-48...142.5	0.75	°C
Modeled injection valve temperature at engine start, depending on TCO, T_ES, TFU_IV_SUB_STOP and TEG_DYN_STOP					
LV_TFU_IV_ES	V	0...1H	0...1	1	-
Flag for TFU_IV calculation after engine stop and before power latch active					
TFU_IV_SUB_MV	V	0...FEH	-48...142.5	0.75	°C
Modeled injector fuel temperature depending on TFU_IV_SUB_BAS and TEMP_CAPA_IV_MV					
TFU_ADD_FUEL_FLOW	V	0...FEH	-48...142.5	0.75	°C
Fuel temperature increase / decrease due to fuel flow					
TEMP_FUEL_RAIL_WALL	V	0...FEH	-48...142.5	0.75	°C
Modeled fuel rail wall temperature					
TFU_OFS_AIR	V	0...FEH	-48...142.5	0.75	°C
Fuel temperature offset depending on ambient conditions					
TEMP_FUEL_RAIL_SUB	V	0...FEH	-48...142.5	0.75	°C
Modeled fuel rail temperature depending from TCO and N					

Input data:

TCO	N	TCO_EX	VS
ECFPWM_ECF	TAM	TEG_DYN_STOP	T_AST
MFF_SP_MV	N_REL_CWP	VFF_MFF_SP_FUP_CTL	LV_RUN_ENG
TCO_ST	LV_T_REL_CAN_REG	T_ES	TEMP_CAPA_IV_MV
IGA_AV_MV_H_RNG			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_FAC_TEMP_DIF_TCO_TIA	1	0...FFH	0...0.99609375	0.00390625	-
Low pass filter correlation constant for difference of coolant temperature					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_FAC_TEMP_IV	1	0...FFH	0...0.99609375	0.00390625	-
Low pass filter correlation constant for the injection valve temperature out of coolant temperature					
C_CRLC_FAC_VFF_MFF_SP_FUP_CTL	1	0...FFH	0...0.99609375	0.00390625	-
Low pass filter correlation constant for the weighting factor of volume fuel flow through the injectors					
C_CRLC_FAC_VS_ECFPWM_ECF	1	0...FFH	0...0.99609375	0.00390625	-
Low pass filter correlation constant for the weighting factor for air speed					
C_CRLC_TFU_IV_ES	1	0...FFFFH	0...23.9996338	3.66211E-4	°C
Limited Gradient step for calculation of TFU_IV at engine stop befor power latch or LV_IGK = 1 after engine stop					
C_TFU_EFP_OFS	1	0...FEH	-48...142.5	0.75	°C
Offset for the modelling of the fuel temperature of the low pressure pump					
C_TFU_IV_SUB_LIM_DEC	1	0...FFFFH	0...23.9996338	3.66211E-4	°C
Limited Gradient step for TFU_IV_SUB decrement					
C_TFU_IV_SUB_LIM_INC	1	0...FFFFH	0...23.9996338	3.66211E-4	°C
Limited Gradient step for TFU_IV_SUB increment					
C_TFU_RAW_LIM_DEC	1	0...FFFFH	0...23.9996338	3.66211E-4	°C
Limited Gradient step for TFU_RAW decrement					
C_TFU_RAW_LIM_INC	1	0...FFFFH	0...23.9996338	3.66211E-4	°C
Limited Gradient step for TFU_RAW increment					
C_TFU_RAW_SWI_INI	1	0...AH	0...10	1	-
Switch for initial value for gradient limitation of TFU_RAW after reset: 1 = 0hex (-48°C), 2 = TCO, 3 = TFU_LPF_TCO					
C_TFU_SUB_STOP_INI	1	0...FEH	-48...142.5	0.75	°C
TFU_SUB_STOP initialisation value					
C_T_AST_SWI_TFU_IV	1	0...FFFFH	0...6.5535E+3	0.1	s
Timer after start for activation of TEMP_CAPA_IV_MV based function					
C_T_TFU_CLC	1	1...FFH	1...255	1	s
Time recurrence for TFU calculation					
C_T_TFU_IV_SUB_CLC	1	1...FFH	1...255	1	s
Time recurrence for TFU_IV_SUB calculation					
C_T_TFU_IV_SUB_RAW_CLC	1	1...FFH	1...255	1	s
Time recurrence for TFU_IV_SUB_RAW calculation					
LC_TCO_TFU_IV_SUB	1	0...1H	0...1	1	-
Switch between TCO and TFU_IV_SUB for TFU_IV					
LC_TEMP_CAPA_TFU_IV_SWI	1	0...1H	0...1	1	-
Switch between TEMP_CAPA based function (1) and TCO based function (0)					
LC_TEMP_FUEL_RAIL_WALL_TAM	1	0...1H	0...1	1	-
Logical switch TEMP_FUEL_RAIL_WALL and TAM for TFU_RAW calculation					
LC_TFU_TIA_TCO_EX	1	0...1H	0...1	1	-
Logical switch TIA and TCO_EX for TFU_OFS_AIR calculation					
IP_CRLC_TFU_IV_SUB_VFF_MFF_SP	9	0...FFFFH	0...0.99998474	1.52588E-5	-
LDP_VFF_MFF_SP_IP_CRLC_TFU_IV	9	0...FFFFH	0...255	0.00389105	l/h
Low pass filter correlation depending on VFF_MFF_SP_FUP_CTL					
IP_CRLC_VFF_MFF_TFU_IV_SUB_MV	9	0...FFH	0...0.99609375	0.00390625	-
LDPM_VFF_MFF_SP_FUP_CTL	9	0...FFFFH	0...255	0.00389105	l/h
Fading factor depending on VFF_MFF_SP_FUP_CTL					
IP_FAC_TCO_TEMP_IV_COR	6	0...FFH	0...1.9921875	0.0078125	-
LDP_TCO_IP_FAC_TCO_TEMP_IV_COR	6	0...FEH	-48...142.5	0.75	°C
Correction factor for correlation between TCO and injector temperature					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_TEMP_IV_CWP_COR	6	0...FFH	0...1.9921875	0.0078125	-
LDP_N_IP_FAC_TEMP_IV_CWP_COR	6	0...FFH	0...255	1	-
Factor for correlation between RPM of cooling water pump and injector temperature					
IP_FAC_TIV_EWP_COR	6	0...FFH	0...1.9921875	0.0078125	-
LDP_TIV_EWP_COR_IP_FAC_TIV_EWP	6	0...FFH	0...255	1	-
Factor for correlation between RPM of electrical water pump and injector temperature					
IP_FAC_TIV_TCO_COR	6	0...FFH	0...1.9921875	0.0078125	-
LDP_TIV_TCO_COR_IP_FAC_TIV_TCO	6	0...FEH	-48...142.5	0.75	°C
Factor for correlation between TCO and injector temperature					
IP_FAC_T_ES	9	0...FFH	0...1.9921875	0.0078125	-
LDP_T_ES_IP_FAC_T_ES	9	0...FFFFH	0...6.5535E+4	1	min
Factor for correction of Engine off duration time					
IP_FAC_T_ES_TFU_IV	9	0...FFH	0...1.9921875	0.0078125	-
LDP_T_ES_IP_FAC_T_ES_TFU_IV	9	0...FFFFH	0...6.5535E+4	1	s
Factor for correction of Engine off duration time for TFU_IV_ST					
IP_FAC_VFF_MFF_SP_FUP_CTL	9	0...FFH	0...1.9921875	0.0078125	-
LDPM_VFF_MFF_SP_FUP_CTL	9	0...FFFFH	0...255	0.0038910 5	l/h
Factor for conversion of required fuel volume to cooling					
IP_TAM_ADD_TFU_IV	8	0...FEH	-48...142.5	0.75	°C
LDP_TAM_ADD_TFU_IV	8	0...FEH	-48...142.5	0.75	°C
Correction of the influence of ambient temperatur on TFU_IV					
IP_TAM_ADD_TFU_RAW	8	0...FEH	-48...142.5	0.75	°C
LDP_TAM_ADD_TFU_RAW	8	0...FEH	-48...142.5	0.75	°C
Correction of the influence of ambient temperatur on TFU_RAW					
IP_TCO_TEMP_DIF_TCO_TIA	6	0...FEH	-48...142.5	0.75	°C
LDP_TEMP_DIF_TCO_TIA_IP_TCO	6	0...FEH	-48...142.5	0.75	°C
Factor for correction of the coolant temperature					
IP_CRLC_TFU_T_AST_VFF_MFF_SP	6x6	0...FFH	0...0.99609375	0.0039062 5	-
LDP_T_AST_IP_CRLC_TFU_VFF_MFF	6	0...FFFFH	0...6.5535E+4	1	s
LDP_VFF_MFF_SP_FUP_CTL_IP_CRLC	6	0...FFFFH	0...255	0.0038910 5	l/h
Weighting factor for correction of the fuel temperature					
IP_D_TIV_NORM_N_MFF_SP	6x4	0...FEH	-48...142.5	0.75	°C
LDP_N_IP_D_TIV_NORM_N_MFF_SP	6	0...1FE0H	0...8.16E+3	1	rpm
LDP_MFF_SP_IP_D_TIV_NORM_N_MFF	4	0...FFFFH	0...1.389E+3	0.0211947 8	mg
load-dependent temperatur differential to coolant temperature					
IP_FAC_TEG_T_ES_TEG_DYN_STOP	6x6	0...FFH	0...1.9921875	0.0078125	-
LDP_TEG_DYN_STOP_IP_FAC_TEG	6	0...7FF0H	0...2.047E+3	0.0625	°C
LDP_T_ES_IP_FAC_TEG_T_ES	6	0...FFFFH	0...6.5535E+4	1	min
Weighting factor for exhaust temperature					
IP_FAC_TEG_T_ES_TFU_IV	6x6	0...FFH	0...1.9921875	0.0078125	-
LDP_TEG_IP_FAC_TFU_IV	6	0...7FF0H	0...2.047E+3	0.0625	°C
LDP_T_ES_IP_FAC_TFU_IV	6	0...FFFFH	0...6.5535E+4	1	min
Weighting factor for exhaust temperature and engine off duration time for TFU_IV_ST					
IP_FAC_VS_ECFPWM_ECF	6x6	0...FFH	0...1.9921875	0.0078125	-
LDP_VS_IP_FAC_VS_ECFPWM_ECF	6	0...FFH	0...255	1	km/h
LDP_ECFPWM_ECF_IP_FAC	6	0...FFH	0...99.609375	0.390625	%
Weighting factor for air speed					
IP_TEMP_DIF_N_MFF_SP_IV	6x4	0...FEH	-48...142.5	0.75	°C
LDP_N_IP_TEMP_DIF_IV	6	0...1FE0H	0...8.16E+3	1	rpm
LDP_MFF_SP_IP_TEMP_DIF_IV	4	0...FFH	0...1.389E+3	5.4470588 2	mg/stk
load dependent injektor temperatur differential to coolant temperature					
IP_TEMP_OFS_IGA_MFF_SP	8x8	0...FEH	-48...142.5	0.75	°C
LDP_IGA_IP_TEMP_OFS_MFF_SP	8	0...B40H	-90...90	0.0625	°CRK
LDP_MFF_SP_IP_TEMP_OFS_MFF_SP	8	0...FFFFH	0...1.389E+3	0.0211947 8	mg/stk
Temperature offset to TCO based on IGA_AV_MV and MFF_SP_MV					

34.5.1 FUSL_M400K

The temperature of the fuel has influence to the density and the volume of the fuel. This has consequences to several points of the function within the fuel supply aggregate and the injection timing. The influence is a lower efficiency of the pump and a influence on the injection valves due to higher temperature of different calculations of the fuel mass in the rail. Therefore two different temperature are modelled, TFU for the fuel pump control and TFU_IV for the injection valve temperature

Both fuel temperatures are expected to depend mainly on the coolant temperature. Beyond this there is an influence of the fuel volume flow, the engine temperature, the ambient temperature, the vehicle speed and time since the last engine stop and the air temperature.

Application condition:

Activation: LV_RUN_ENG = 1;

Deactivation: At power latch;

Initialisation: At reset: TFU_IV_SUB_RAW, TFU_IV_SUB, TFU_IV, TFU, TFU_RAW, TFU_LPF_TCO, TFU_IV_ST = TCO_ST;

At IGK off to on:

PT1 filter constant of TFU_RAW = 0

local variables of gradient limiters tfu_raw_step_clc and tfu_iv_sub_step_clc = 0

local counter for calculation delay of TFU values = 0

Recurrence: every 1S

Except the the following calculation:

TFU_LPF_CLC, TFU_IV_ST_CLC Activation: LV_T_REL_CAN_REG = 1 and LV_RUN_ENG = 0 Recurrence: 100 ms

TFU_INIT_NO_T_REL_CAN Activation: LV_T_REL_CAN_REG = 0 and LV_RUN_ENG = 0 Recurrence: 100 ms

TFU, which is calculated every C_T_TFU_CLC

TFU_IV_SUB_RAW, which is calculated every C_T_TFU_IV_SUB_RAW_CLC

TFU_IV_SUB, which is calculated every C_T_TFU_IV_SUB_CLC

Application Condition

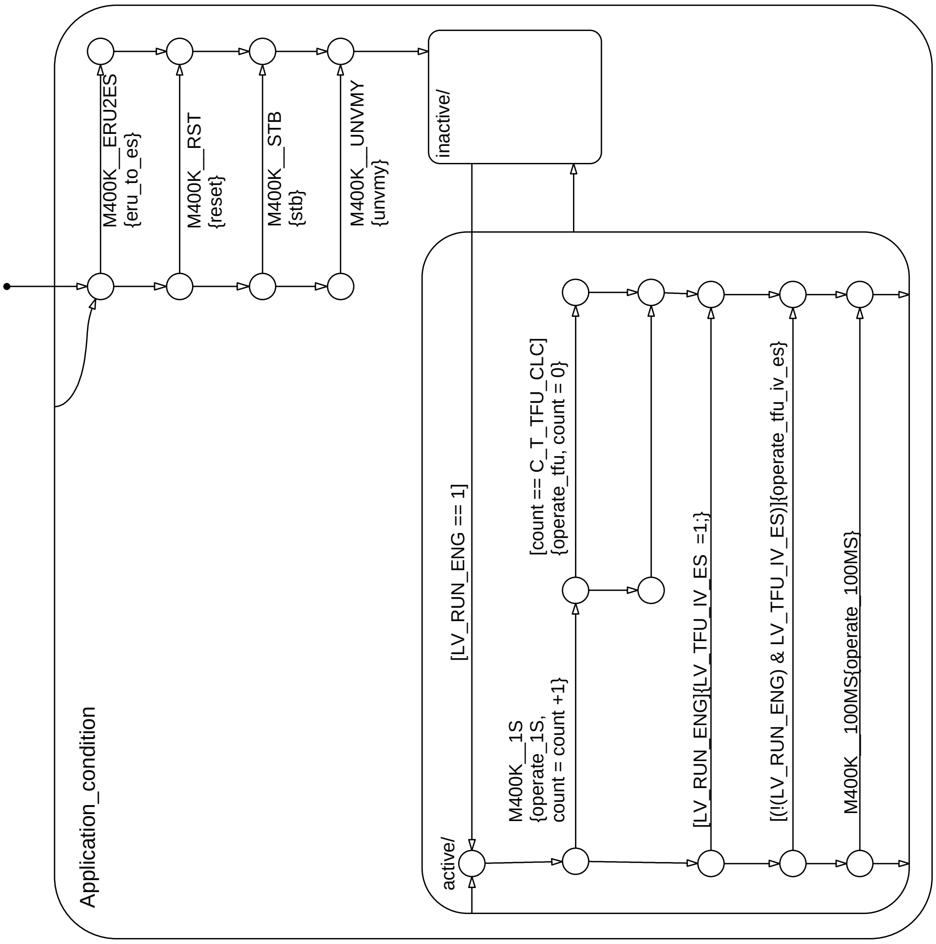
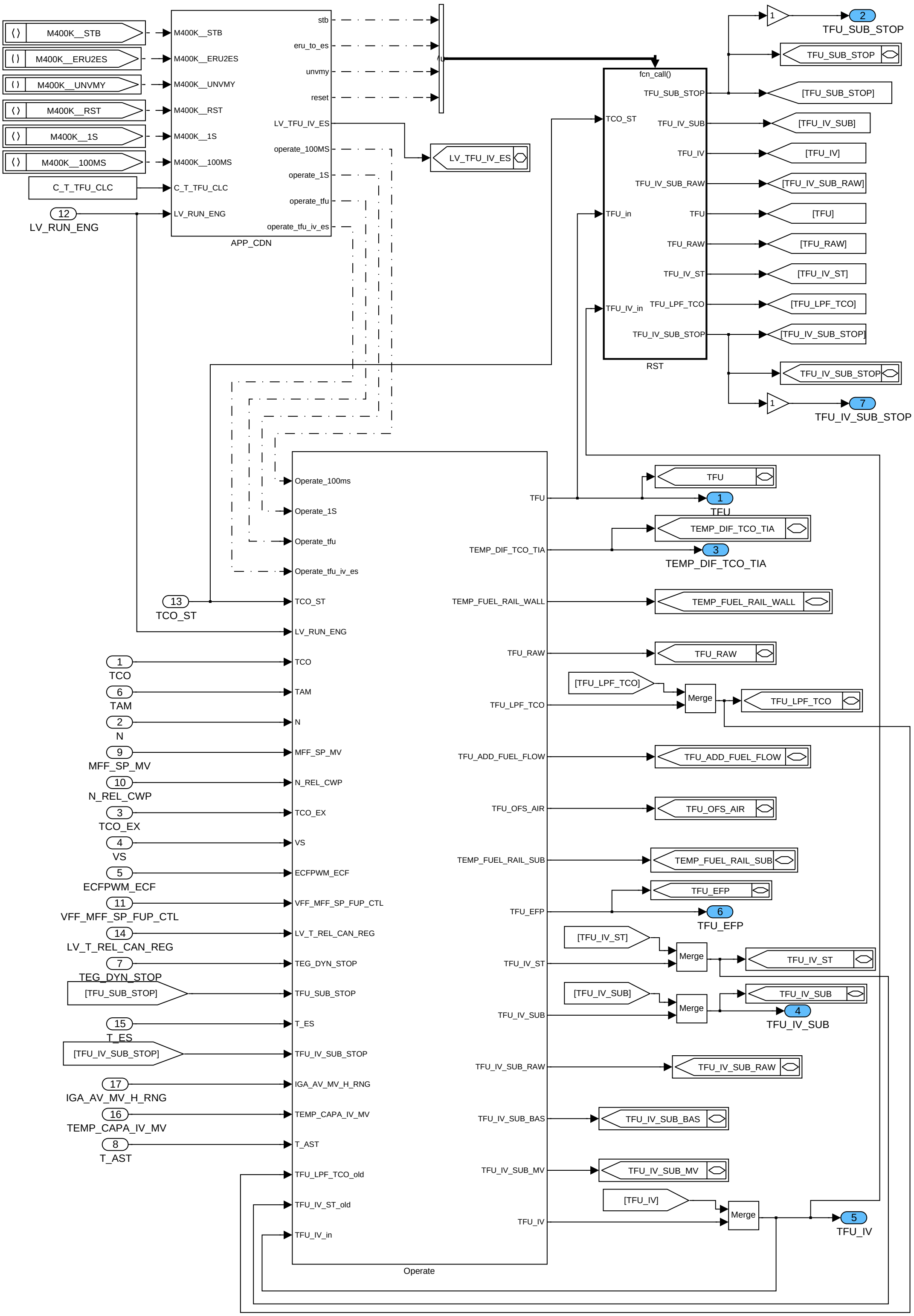


Figure 1 FUSL_M400K/ APP_CDN/ Chart1

Function Description



SDA_SRS / SDA 4.0 22−Jun−2007

Figure 2 FUSL_M400K

34.5.1.1 SUBFUNCTION: Operate

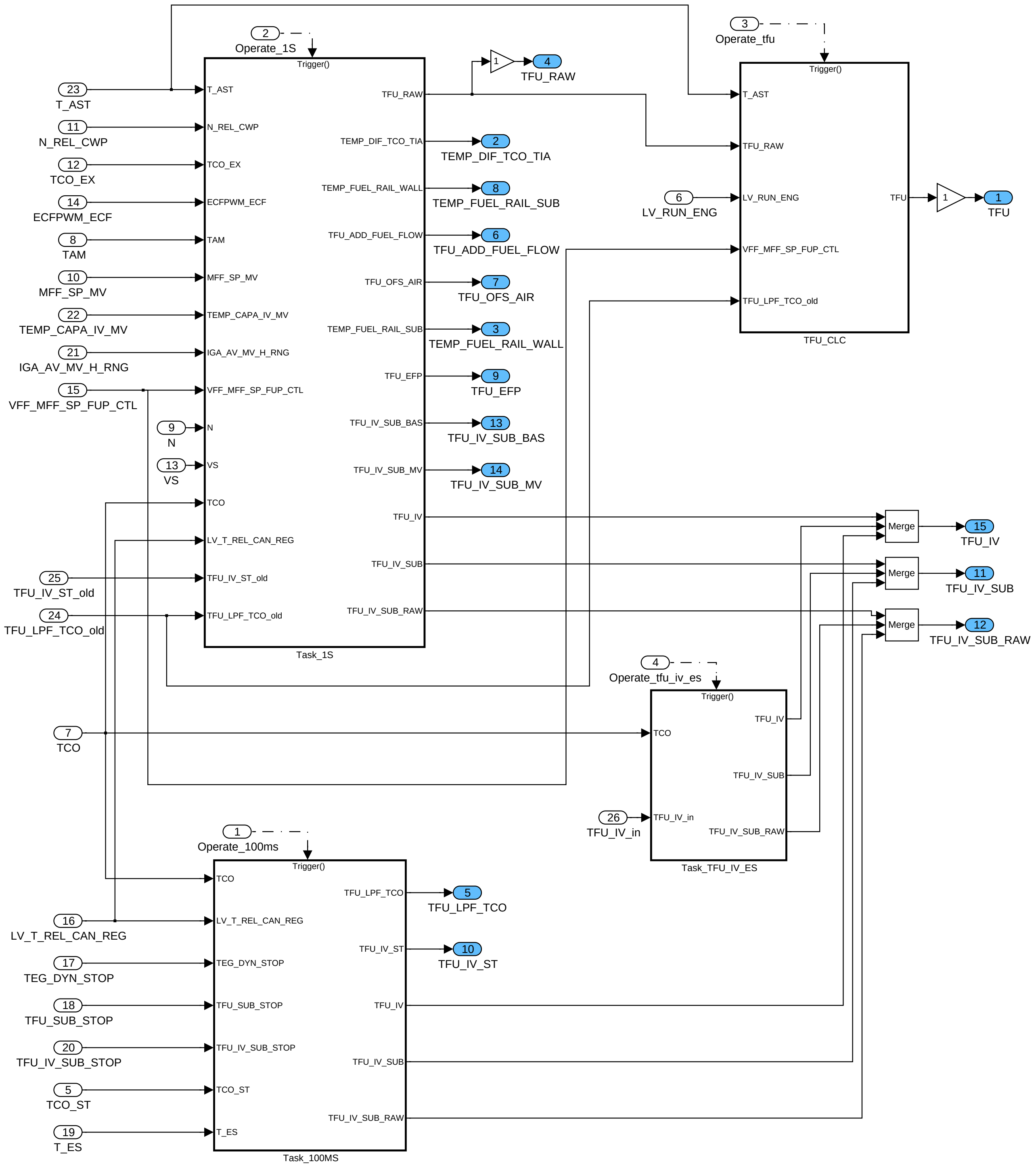


Figure 3 FUSL_M400K/ Operate

<u>FUSL_M400K/OPERATE/TFU_CLC</u>

Low pass filter of TFU_RAW to TFU. The initial value TFU_LPF_TCO for the low pass filter depends on TCO, the exhaust gas temperature at engine stop, the engine of time and the fuel temperature at engine stop.

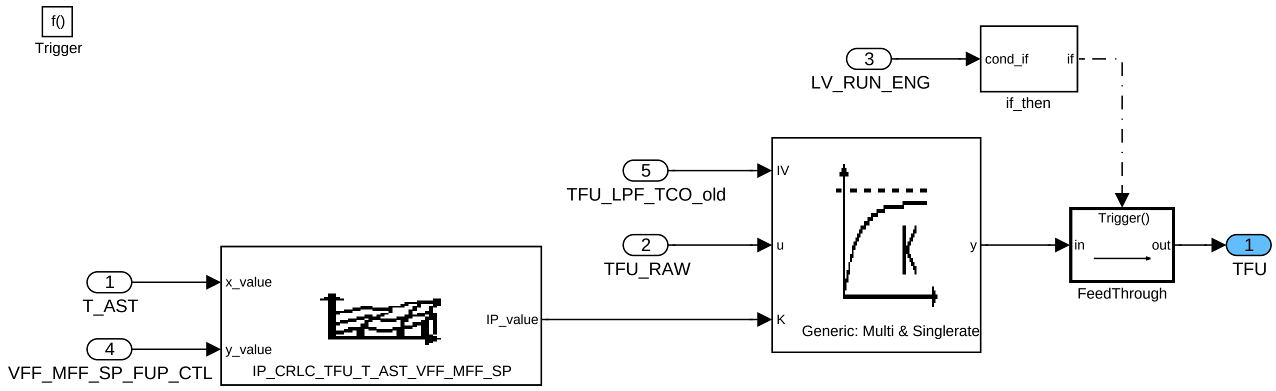


Figure 4 FUSL_M400K/ Operate/ TFU_CLC

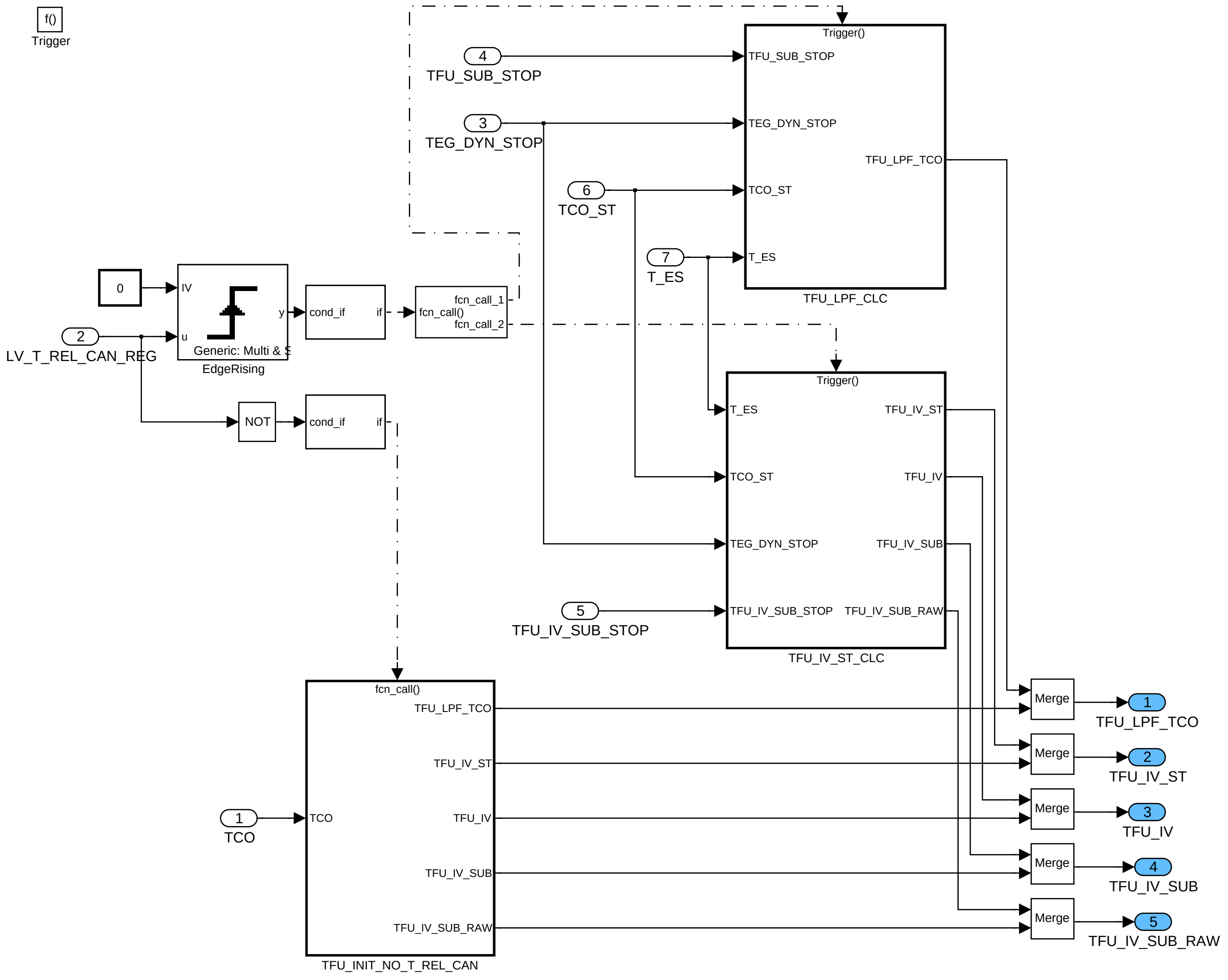


Figure 5 FUSL_M400K/ Operate/ Task_100MS

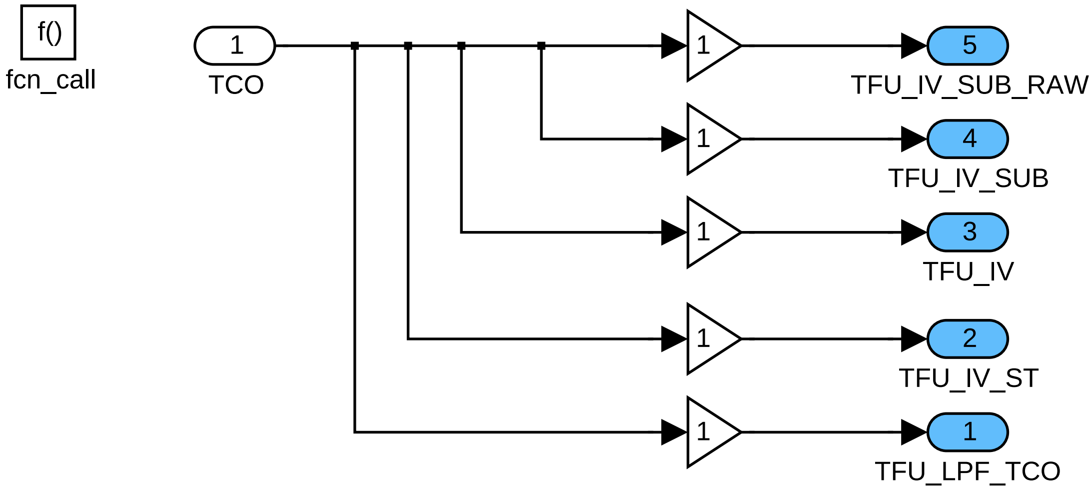


Figure 6 FUSL_M400K/ Operate/ Task_100MS/ TFU_INIT_NO_T_REL_CAN

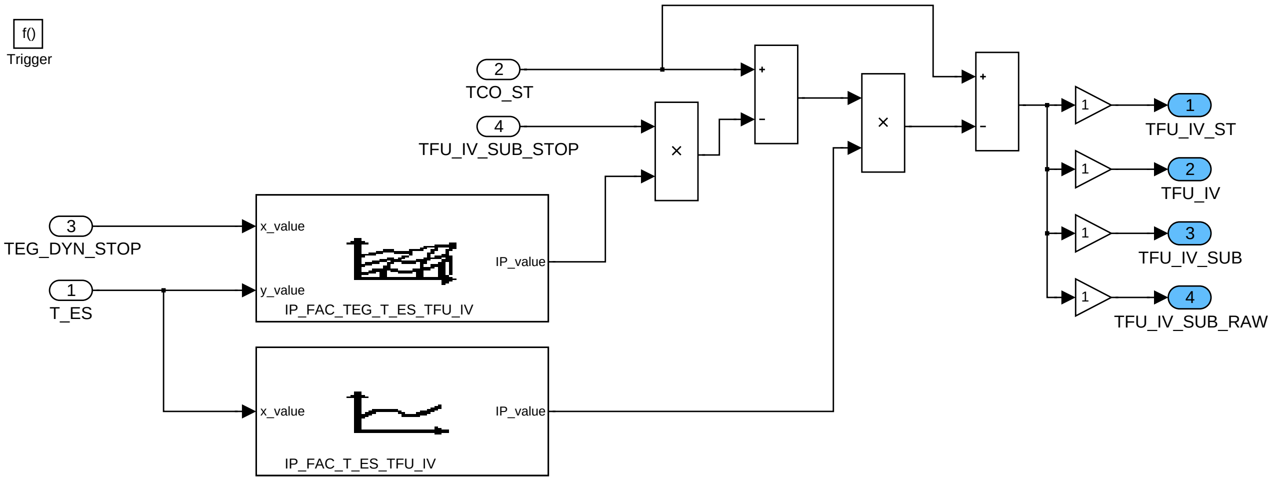


Figure 7 FUSL_M400K/ Operate/ Task_100MS/ TFU_IV_ST_CLC

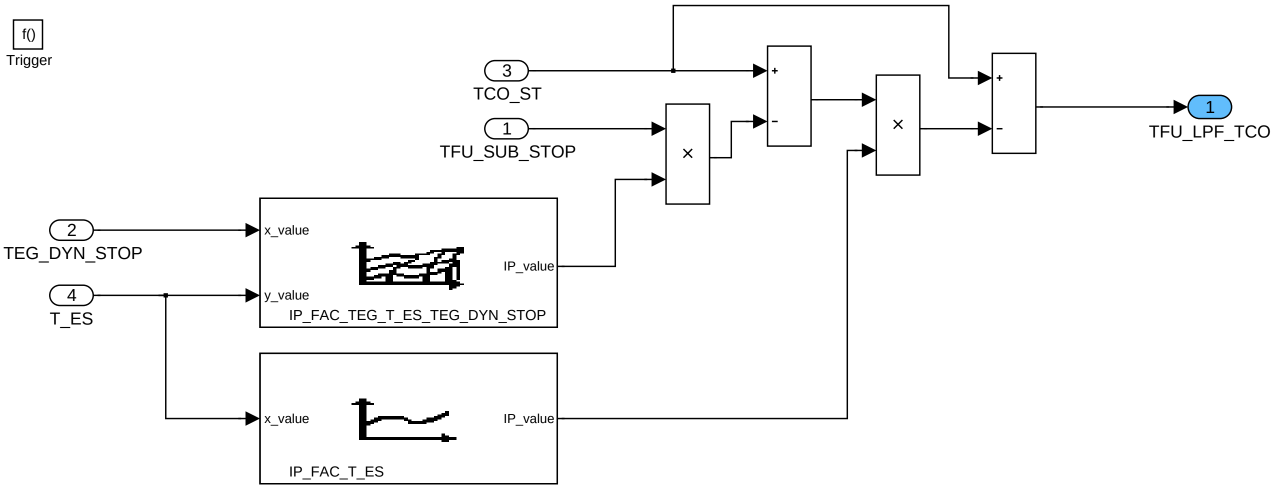


Figure 8 FUSL_M400K/ Operate/ Task_100MS/ TFU_LPF_CLC

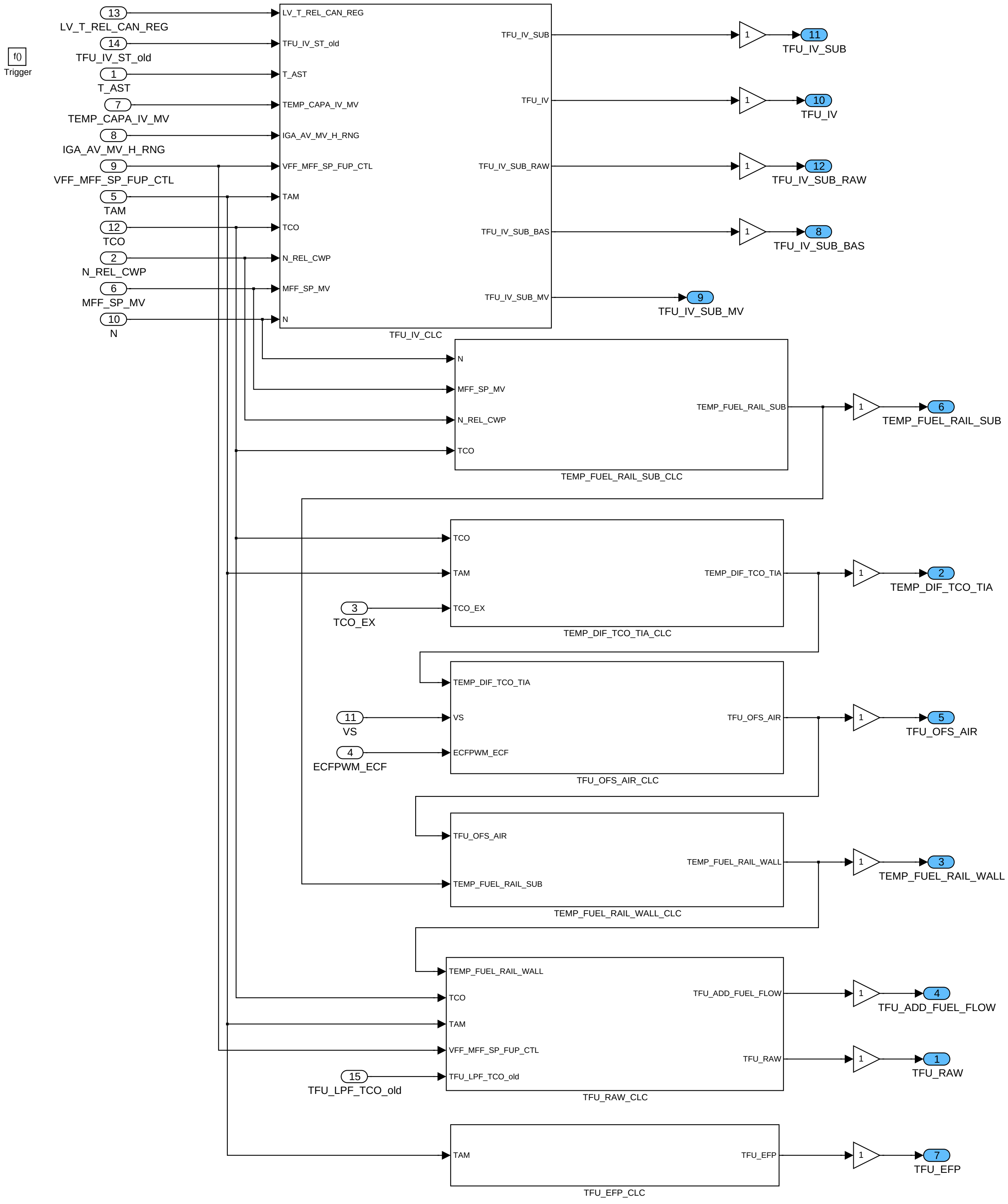


Figure 9 FUSL_M400K/ Operate/ Task_1S

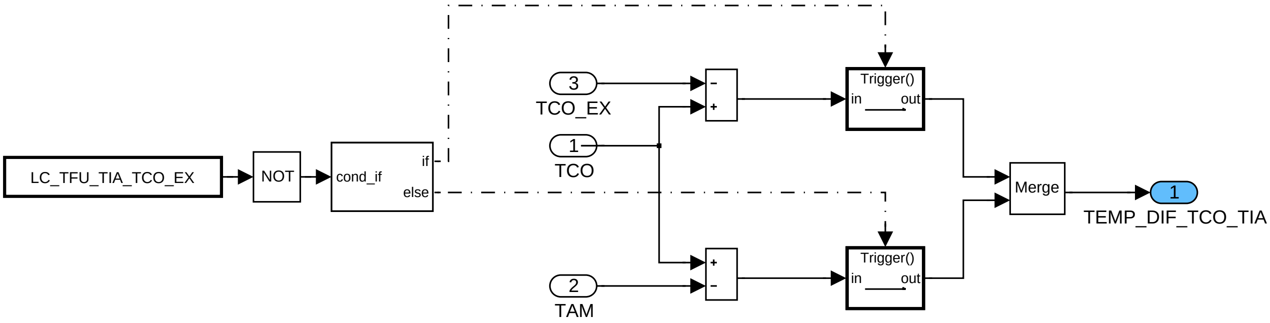


Figure 10 FUSL_M400K/ Operate/ Task_1S/ TEMP_DIF_TCO_TIA_CLC

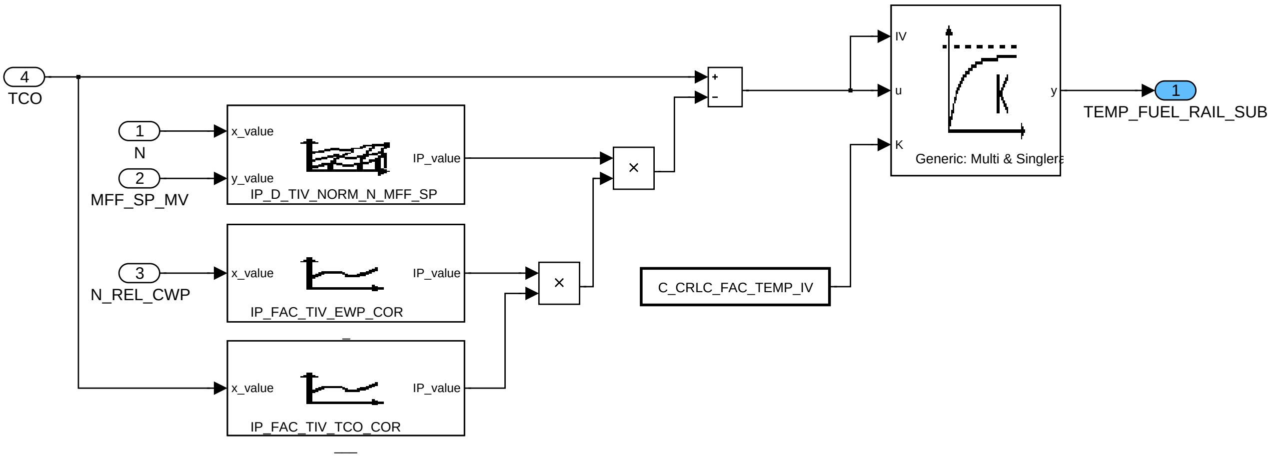


Figure 11 FUSL_M400K/ Operate/ Task_1S/ TEMP_FUEL_RAIL_SUB_CLC

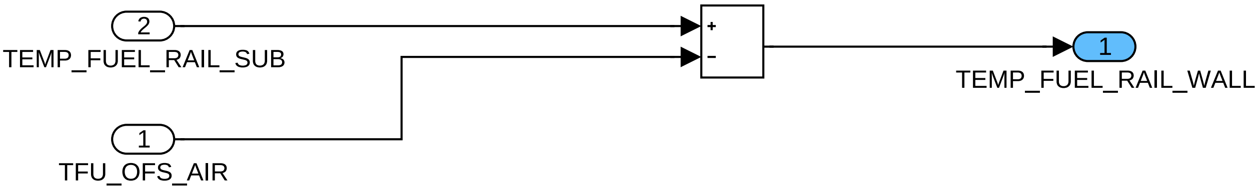


Figure 12 FUSL_M400K/ Operate/ Task_1S/ TEMP_FUEL_RAIL_WALL_CLC

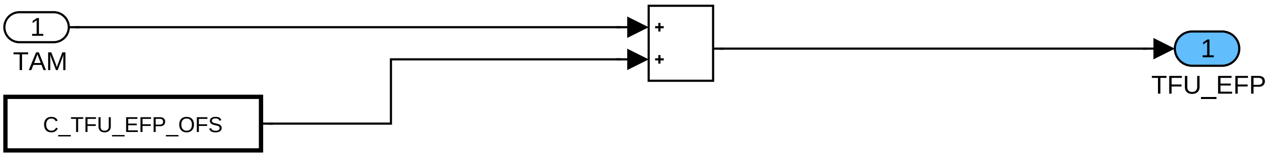


Figure 13 FUSL_M400K/ Operate/ Task_1S/ TFU_EFP_CLC

FUSL_M400K/OPERATE/TASK_1S/TFU_IV_CLC

For the calculation of the injection valve temperature, two functions are available, switchable by the logical constant LC_TEMP_CAPA_TFU_IV_SWI.

1. - For TFU_IV_SUB_MV a mixed temperature of the heating influence of the combustion chamber and the fuel temperature is calculated. The heating influence of the combustion chamber rests on TCO, MFF_SP_MV and ignition angle. The fuel temperature influence is

described by TEMP_CAPA_IV_MV (Temperature of the piezo stacks derived from their capacities).

With higher fuel flows through the injectors, the injector temperature is more influenced by the fuel temperature. The modeled injector temperature can be shifted by a fading factor based on the fuel flow to TEMP_CAPA_IV_MV.

The heat up phase of the injector tip during start is faster than the heating of the fuel temperature. During start phase and short after start, the fading function between TEMP_CAPA_IV_MV and TFU_IV_SUB_RAW can be blocked for a certain time by using C_T_AST_SWI_TFU_IV.

2. - The second function depends mainly on the coolant temperature, influenced by

engine speed, the mass flow through the engine, the ambient temperature and the rpm of the cooling water pump.

The output TFU_IV_SUB_RAW is filtered by a PT1 function. Because of the minimum resolution of the TFU values of 0,75°/digit, the input value of the PT1 is stored internal (not visible) and calculated with an finer resolution of 0,00036°/digit until it reaches the minimum digit.

By using C_T_TFU_IV_SUB_RAW_CLC or C_T_TFU_IV_SUB_CLC

the calculation recurrence of TFU_IV_SUB_RAW (PT1 funktion) or TFU_IV_SUB (GradientLimiter) can be reduced.

The gradient limiters for rising and falling edge, C_TFU_IV_SUB_LIM_DEC and C_

TFU_IV_SUB _LIM_INC have a finer resolution than TFU_IV_SUB. It takes several

calculation cycles until TFU_IV_SUB changes. During this calculation cycles of TFU_IV_SUB a internal variable (not visible) is incremented or decremented depending on rising or failing gradient until it reaches the minimum digit of TFU_IV_SUB (0.75°C).

If the CAN values are not valid (LV_T_REL_CAN_REG = 0), TFU_IV, TFU_IV_SUB and

TFU_IV_SUB_RAW = TCO.

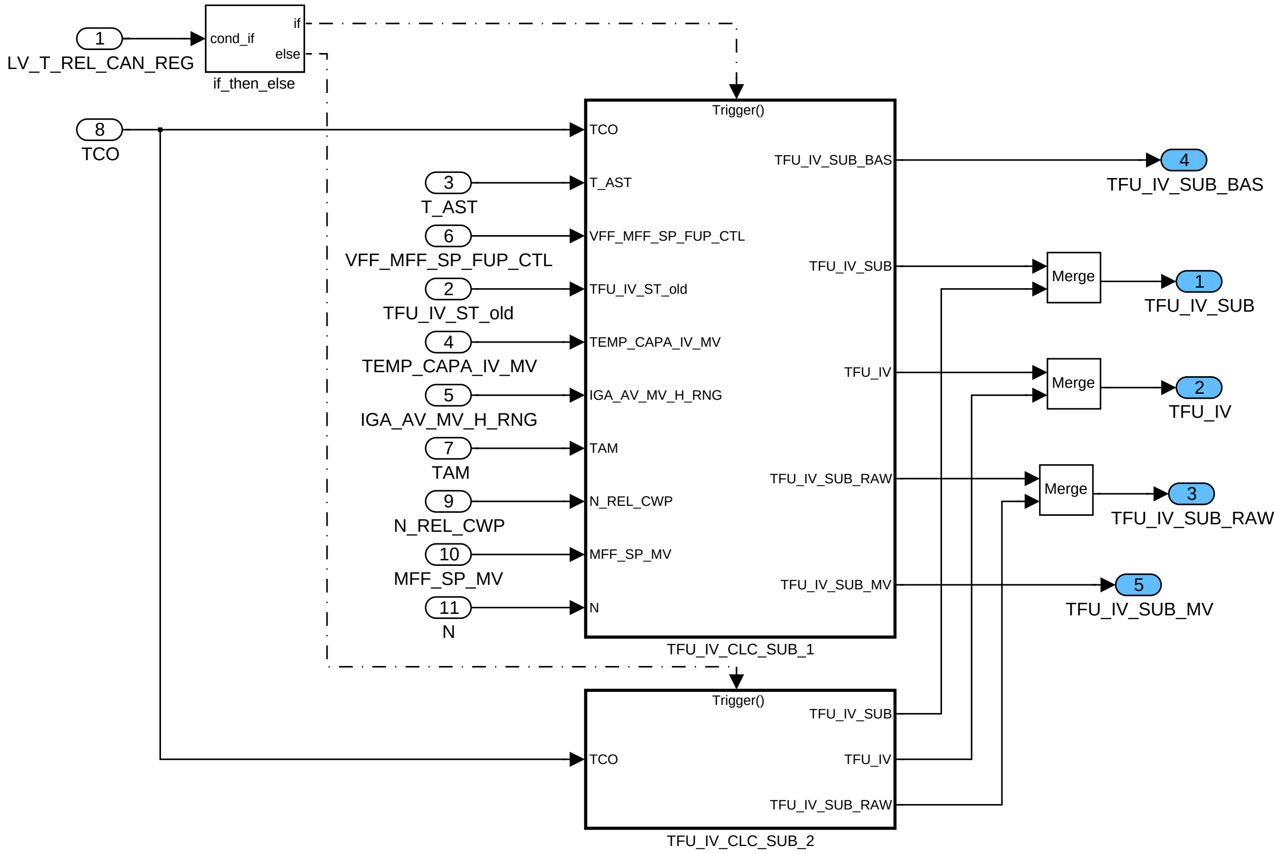


Figure 14 FUSL_M400K/ Operate/ Task_1S/ TFU_IV_CLC

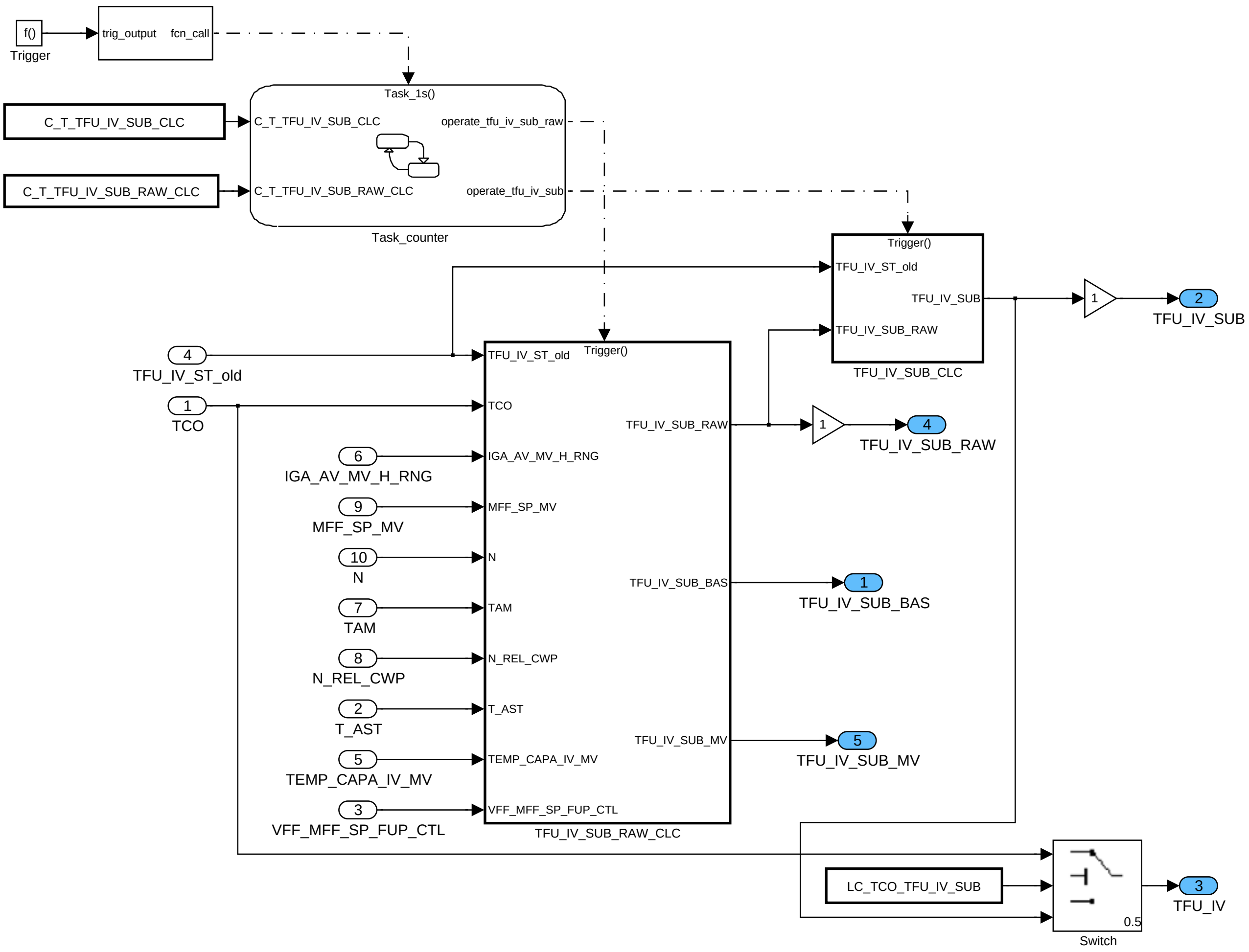


Figure 15 FUSL_M400K/ Operate/ Task_1S/ TFU_IV_CLC/ TFU_IV_CLC_SUB_1

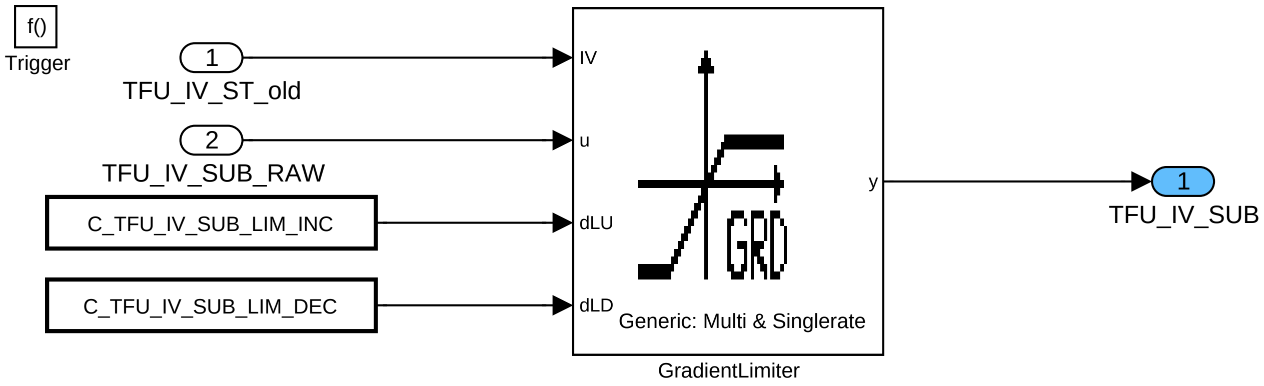


Figure 16 FUSL_M400K/ Operate/ Task_1S/ TFU_IV_CLC/ TFU_IV_CLC_SUB_1/ TFU_IV_SUB_CLC

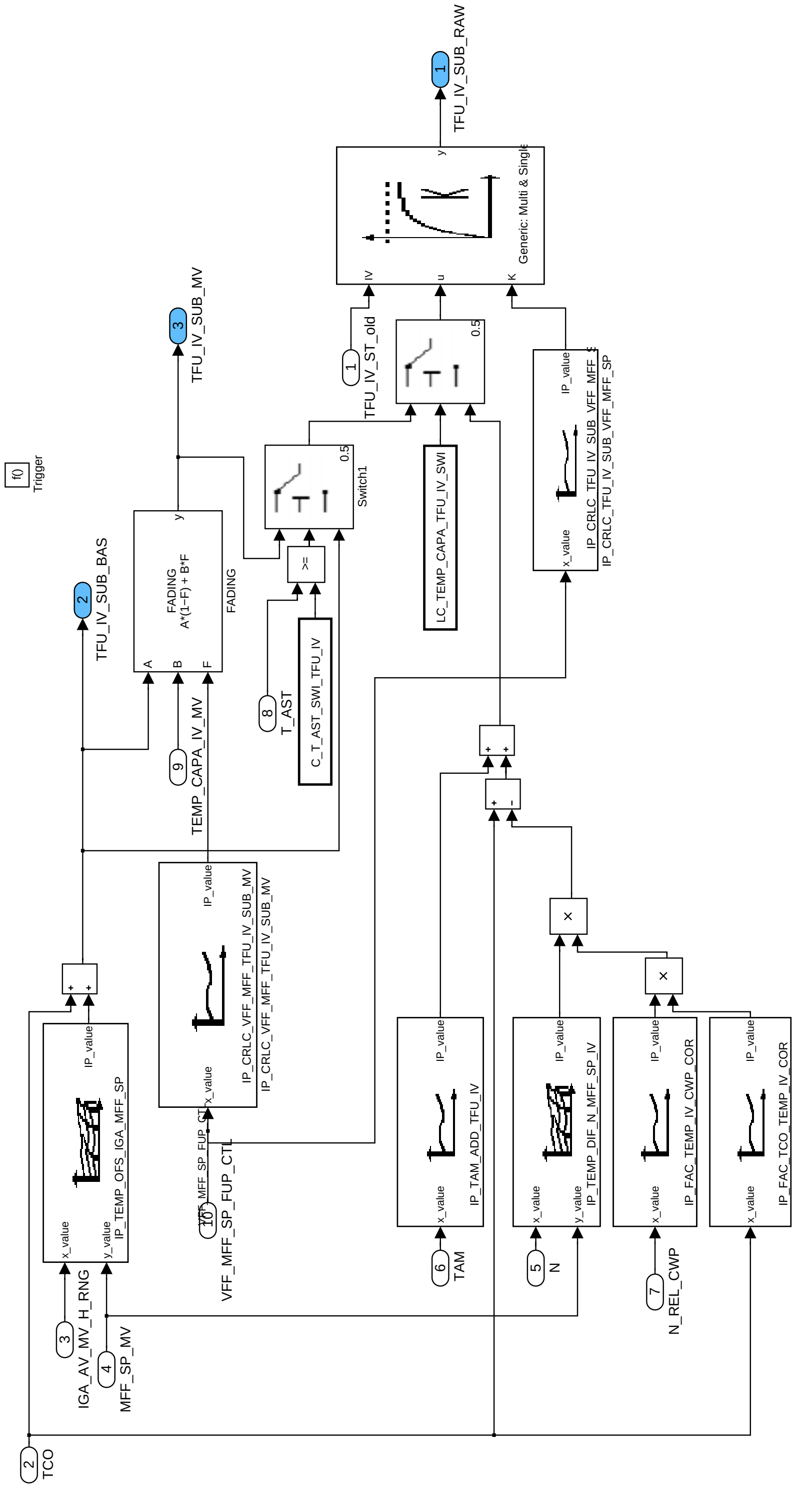


Figure 17 FUSL_M400K/ Operate/ Task_1S/ TFU_IV_CLC/ TFU_IV_CLC_SUB_1/ TFU_IV_SUB_RAW_CLC

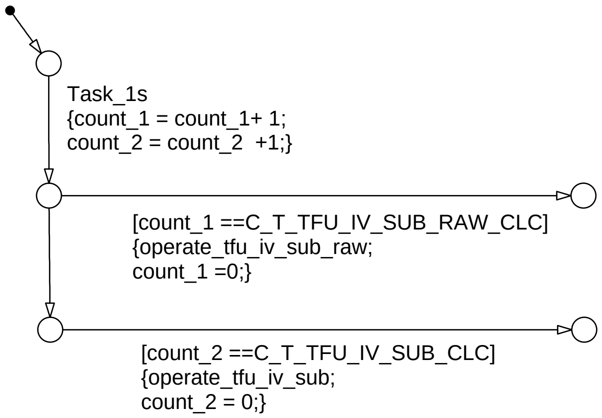


Figure 18 FUSL_M400K/ Operate/ Task_1S/ TFU_IV_CLC/ TFU_IV_CLC_SUB_1/ Task_counter

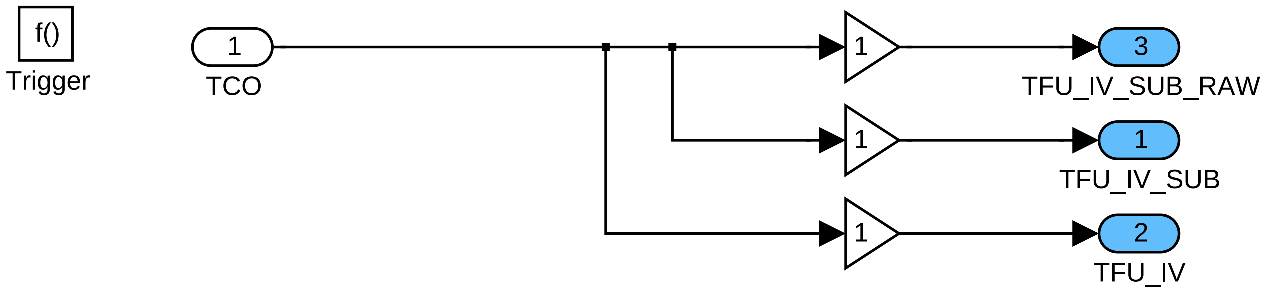


Figure 19 FUSL_M400K/ Operate/ Task_1S/ TFU_IV_CLC/ TFU_IV_CLC_SUB_2

FUSL_M400K/OPERATE/TASK_1S/TFU_OFS_AIR_CLC

Calculation of ambient air influence on fuel temperature due to vehicle speed and electrical cooling fan.

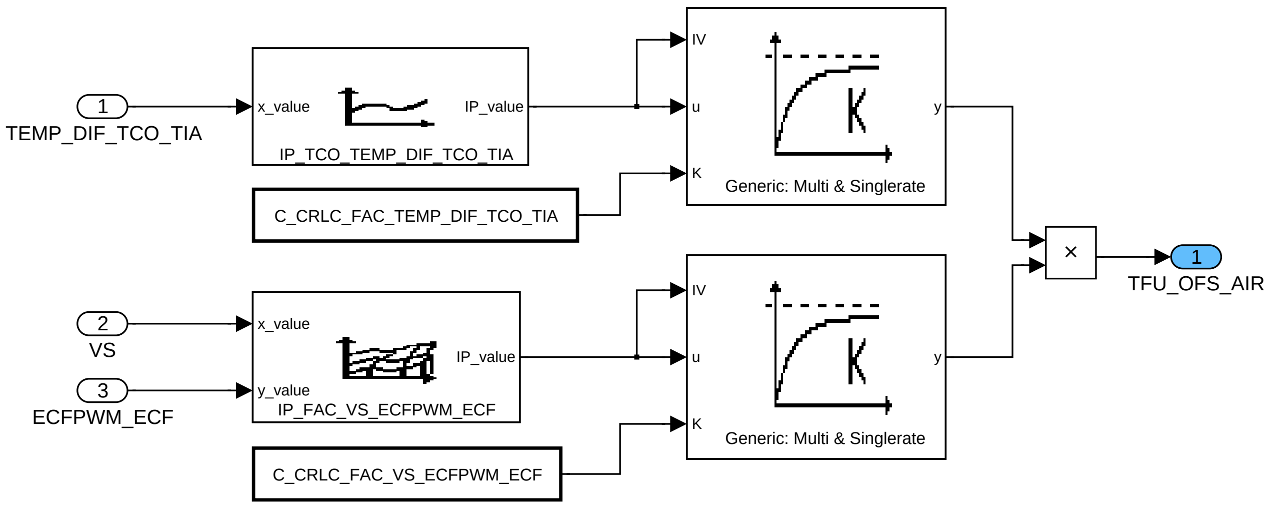


Figure 20 FUSL_M400K/ Operate/ Task_1S/ TFU_OFS_AIR_CLC

FUSL_M400K/OPERATE/TASK_1S/TFU_RAW_CLC

The temperature TFU_RAW is calculated from the fuel rail wall temperature reduced by the decrease of the temperature due to fuel flow through the fuel rail.

C_TFU_RAW_LIM_DEC and C_TFU_RAW_LIM_INC have a finer resolution than TFU_RAW. It takes several calculation cycles (GradientLimiter) until TFU_RAW changes. During this calculation cycles of TFU_RAW the local variable tfu_raw_step_clc is incremented or decremented depending on rising or failing gradient until it reaches the minimum digit of TFU_RAW (0.75°C). After a step of TFU_RAW the lasting value of tfu_raw_step_clc will be used for the next calculation.

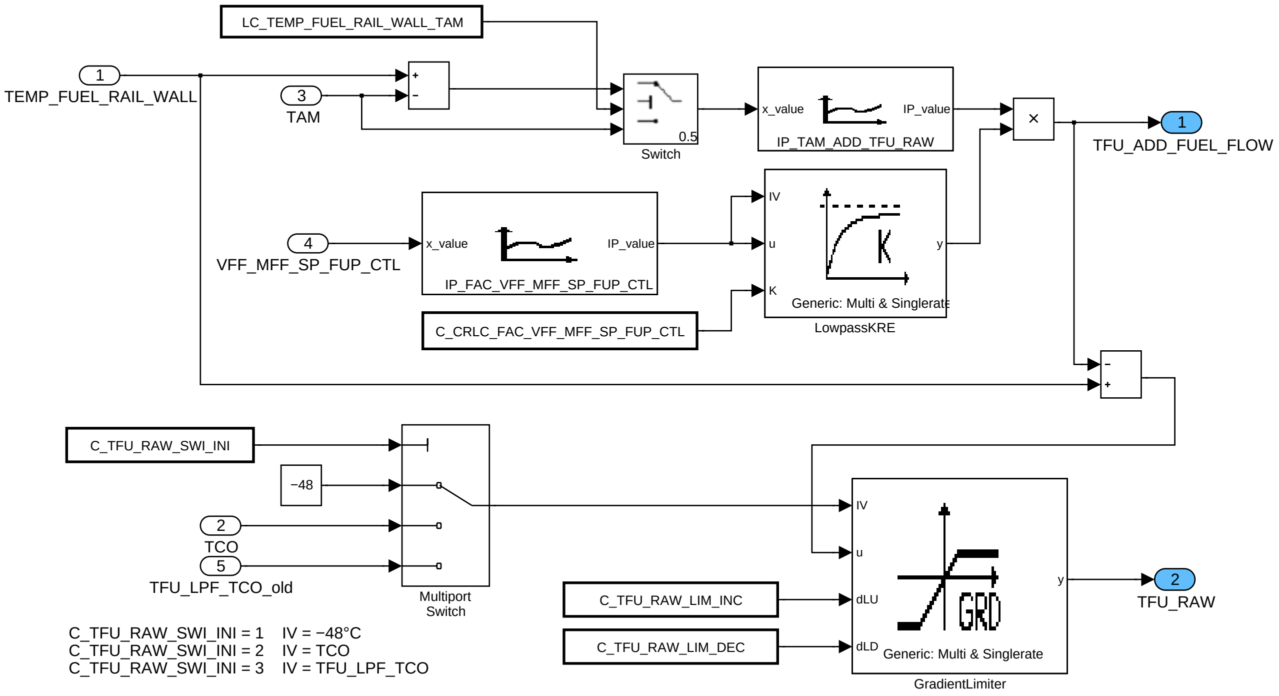


Figure 21 FUSL_M400K/ Operate/ Task_1S/ TFU_RAW_CLC

FUSL_M400K/OPERATE/TASK_TFU_IV_ES

At engine stop, a modelled injector temperature will be calculated based on TCO. Starting with the last modelled temperatures TFU_IV_SUB_RAW, TFU_IV_SUB and TFU_IV the values were incremented by the value C_CRLC_TFU_IV_ES until TCO is reached.

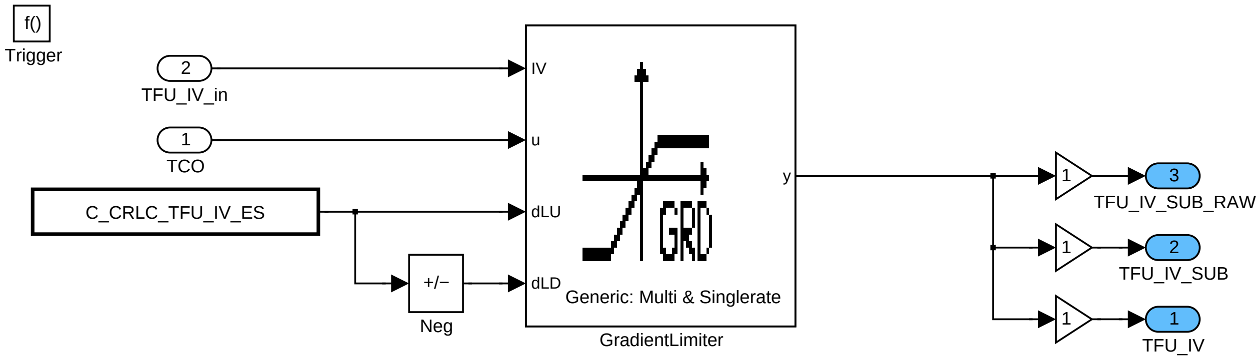


Figure 22 FUSL_M400K/ Operate/ Task_TFU_IV_ES

34.5.1.2 SUBFUNCTION: RST

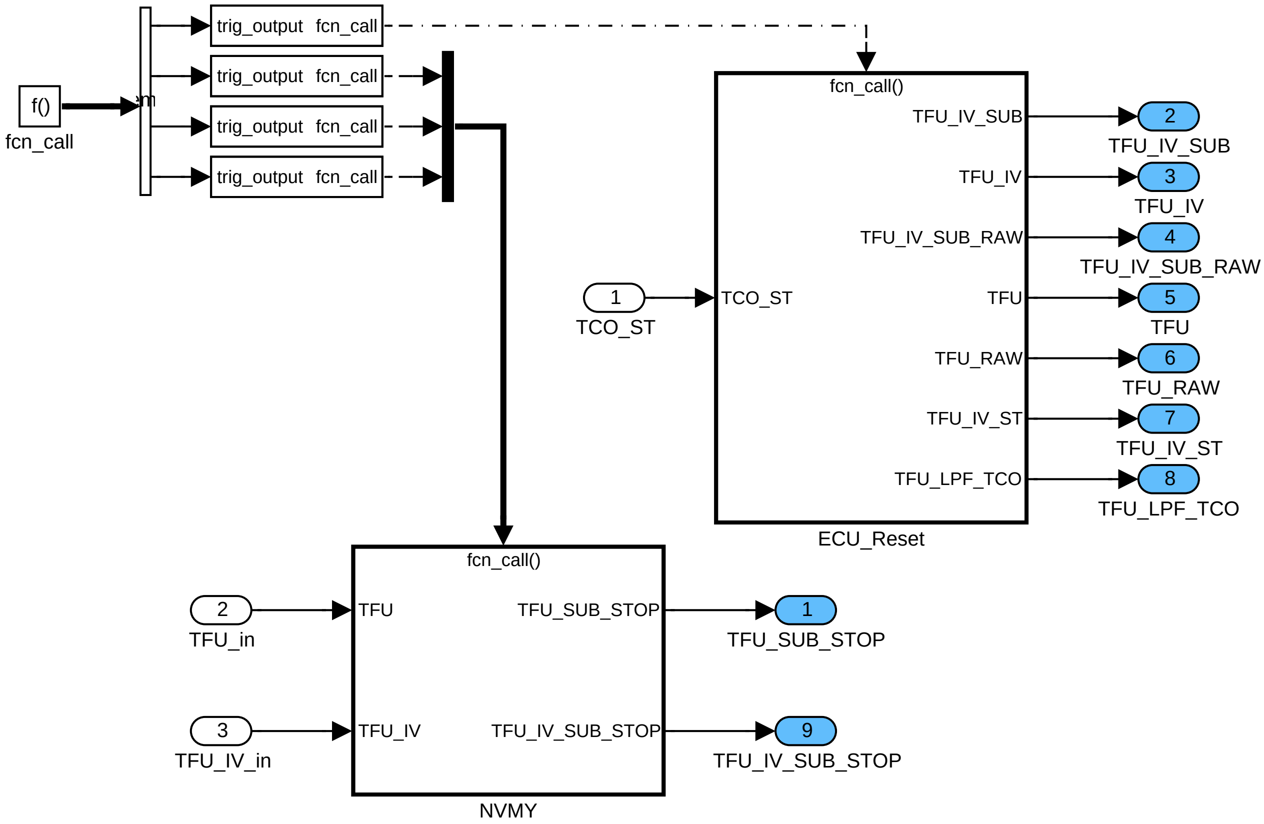


Figure 23 FUSL_M400K/ RST

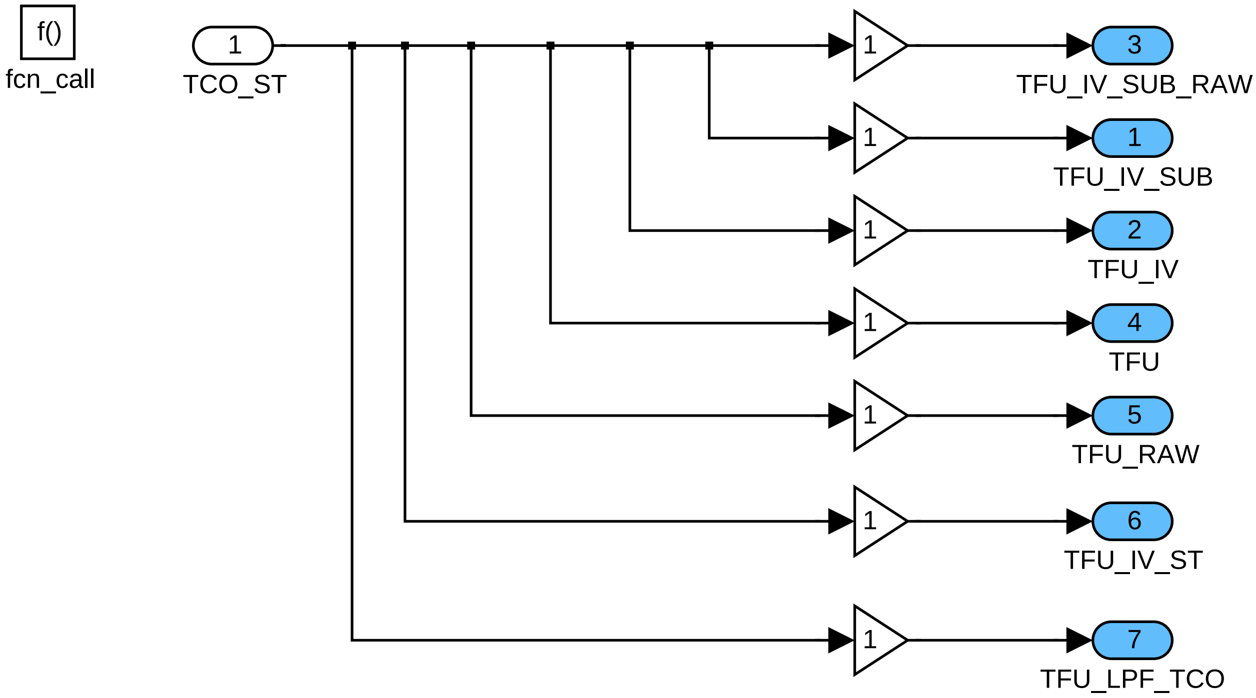


Figure 24 FUSL_M400K/ RST/ ECU_Reset

FUSL_M400K/RST/NVMY

Overview of non volatile memory functions

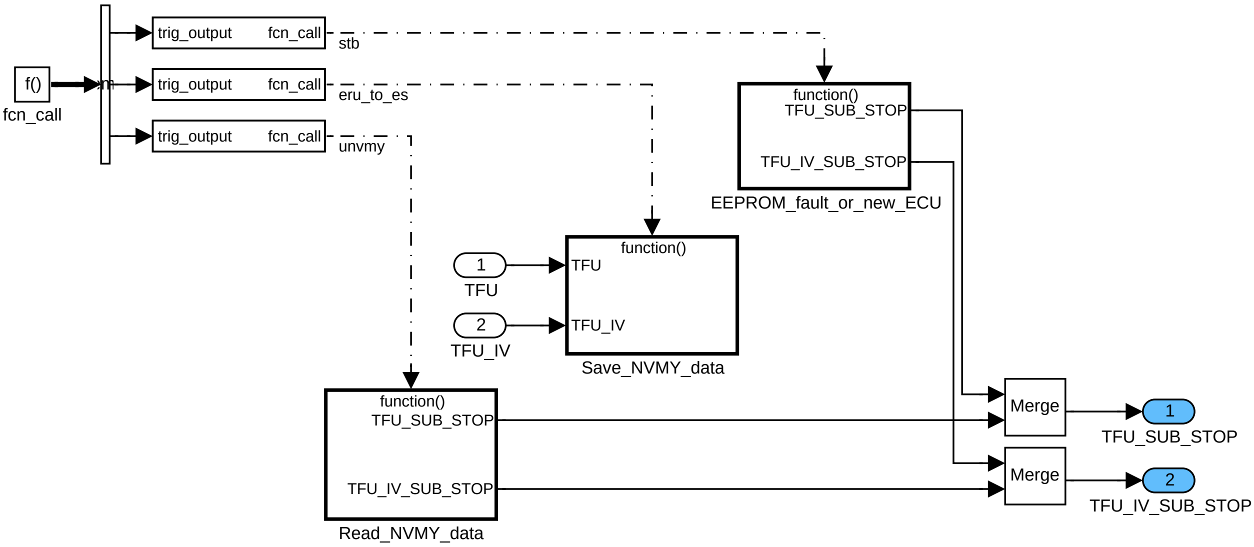


Figure 25 FUSL_M400K/ RST/ NVMY

FUSL_M400K/RST/NVMY/EEPROM_FAULT_OR_NEW_ECU

At EEPROM failure or by using a new ECU, no NVMY are available therefore this constant C_TFU_SUB_STOP_INI will be used as initial value.

Figure 26 FUSL_M400K/ RST/ NVMY/ EEPROM_fault_or_new_ECU

FUSL_M400K/RST/NVMY/READ_NVMY_DATA

Reading stored non volatile values from last engine run.

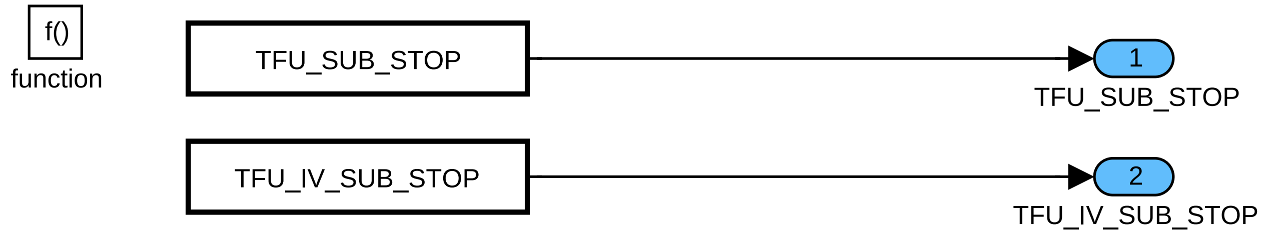


Figure 27 FUSL_M400K/ RST/ NVMY/ Read_NVMY_data

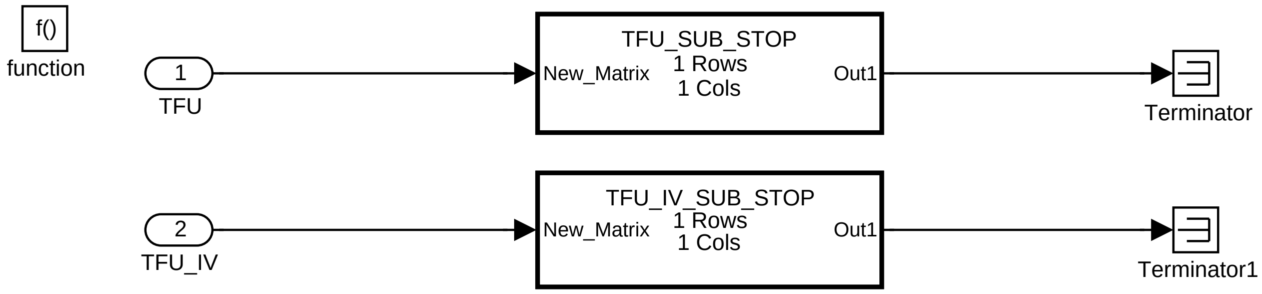


Figure 28 FUSL_M400K/ RST/ NVMY/ Save_NVMY_data

34.6 Fuel pressure

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FUP	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel pressure					
LV_FUP_SEG	O/V	0...1H	0...1	1	[-]
Logical variable indicating the buffer to read the sensor voltage in BSW					
FUP_H	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel pressure, high range					
V_FUP_MV	O/V	0...7FFFH	0...4.99984	0.1526e-3	[V]
Mean value of the acquired sensor voltage					
FUP_RNG_H	O/V	0...FFFFH	0...255.99609	3.9063e-3	[MPa]
Fuel pressure					
FUP_MES	V	0...FFFFH	0...347776	5.3067216	[hPa]
Measured fuel pressure					
FUP_MES_MMV	V	0...FFFFH	0...347776	5.3067216	[hPa]
Measured mean value fuel pressure					
FUP_RNG_H_MES	O/V	0...FFFFH	0...255.99609	3.9063e-3	[MPa]
Measured fuel pressure					

Input data:

V_FUP_SUM	CTR_V_FUP	V_FUP_SUM_1	CTR_V_FUP_1
LV_ST_END	LC_FUP_EFP_AVL	FUP_EFP	LV_FUP_LIH_L_PRS_CTL_REQ
FUP_AD			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FUP_MES_CRLC	1	0...FFH	0...0.99609	3.9063e-3	[-]
Filter constant for the measured fuel pressure value					
C_FUP_OFS_LIH_L_PRS_CTL_REQ	1	0...FFFFH	0...173888	2.6533608	[hPa]
Offset for FUP at low pressure control limphome					
IP_FUP_MES	8	0...FFFFH	0...347776	5.3067216	[hPa]
LDP_V_FUP_MV_IP_FUP_MES	8	0...7FFFH	0...4.99984	0.1526e-3	[V]
Fuel pressure sensor linearisation table					

34.6.1 FUSL_M4018

The fuel pressure sensor is delivering a voltage signal corresponding to the fuel pressure in the rail. Out of this sensor voltage a fuel pressure value is calculated via liniarisation table. An additional filter can be applied.

Application Condition

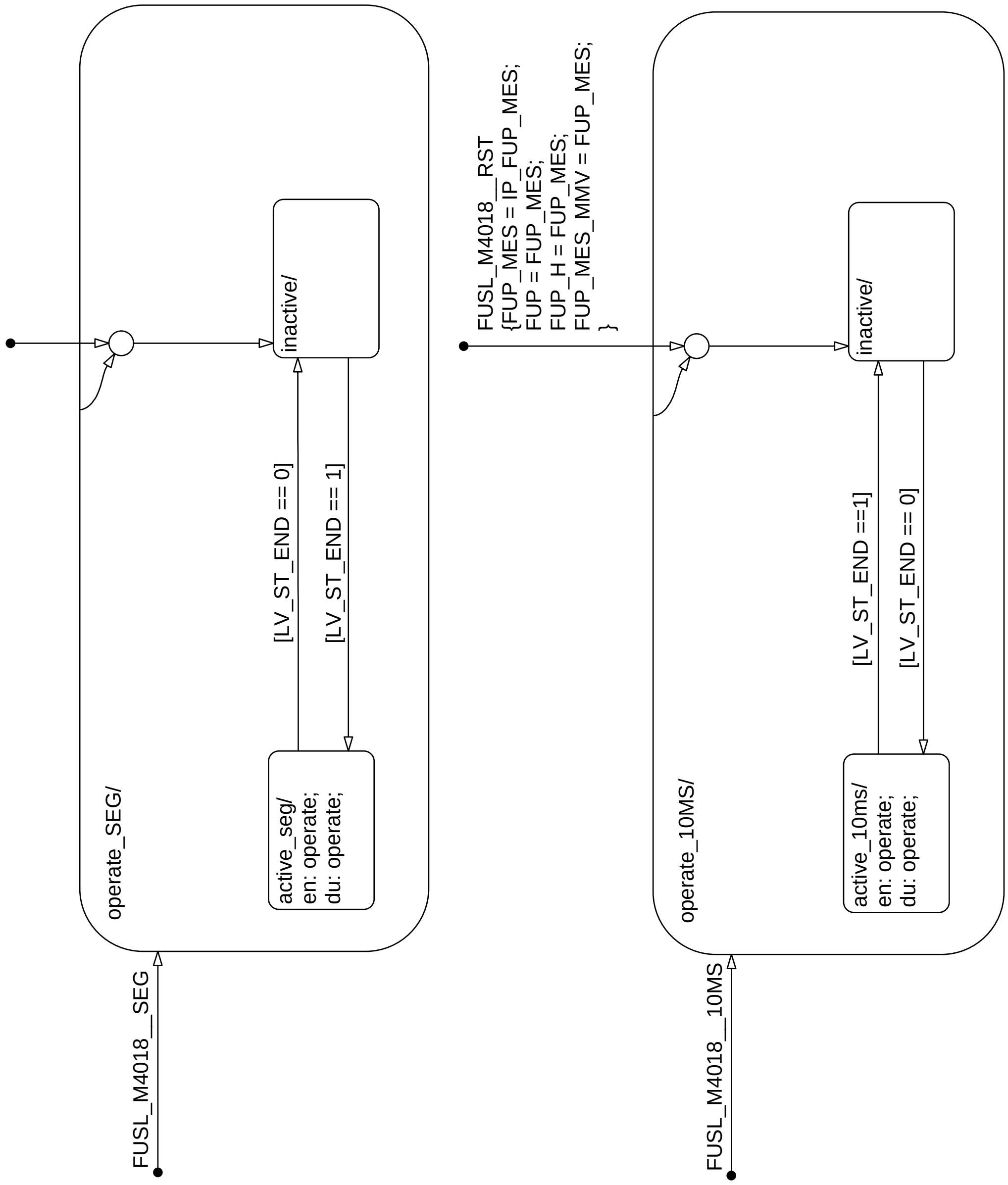


Figure 29 FUSL_M4018/ APP_CDN/ Chart

Function Description

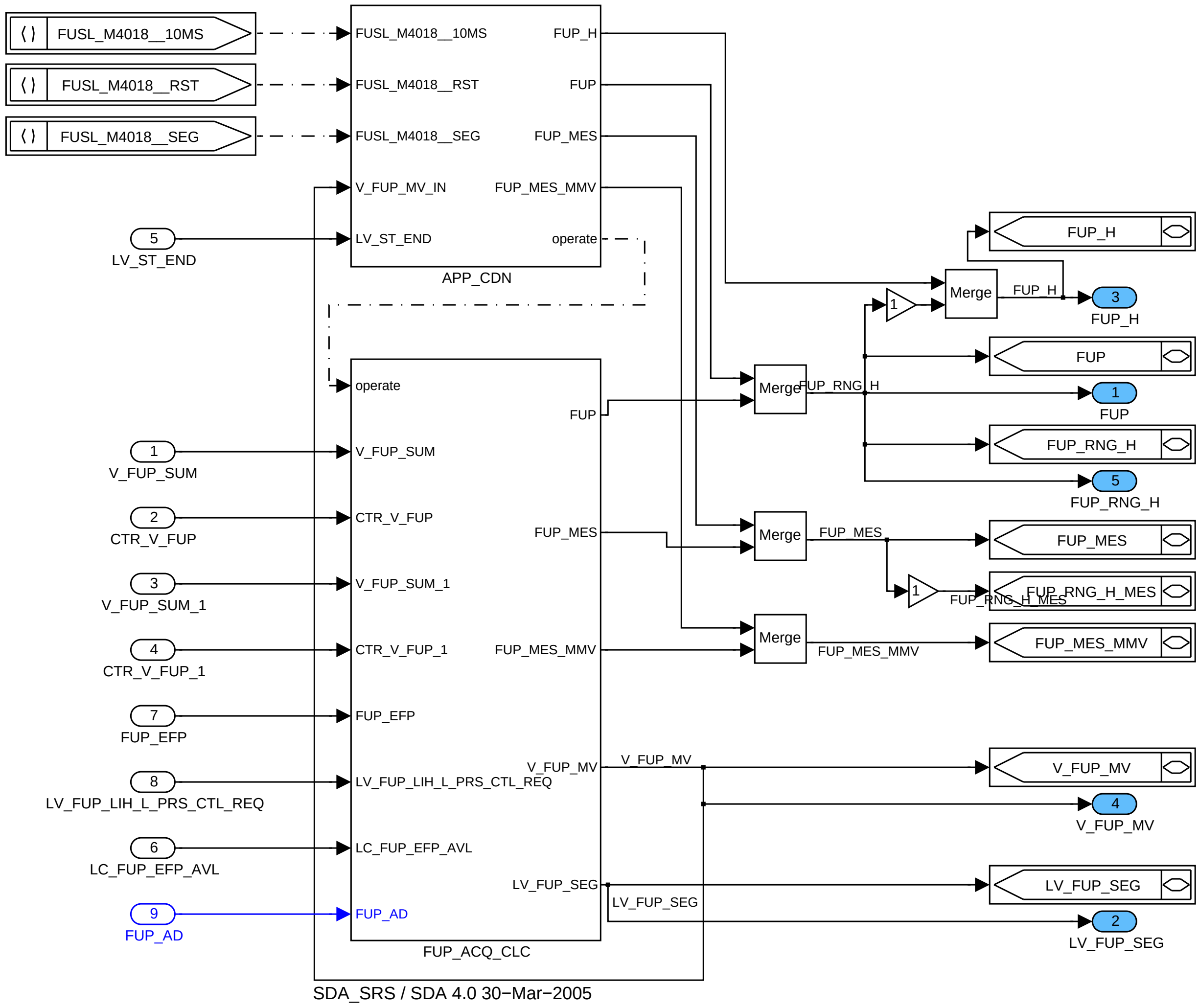


Figure 30 FUSL_M4018

34.6.1.1 FUSL_M4018/FUP_ACQ_CLC

The fuel pressure calculation is split into two parts.

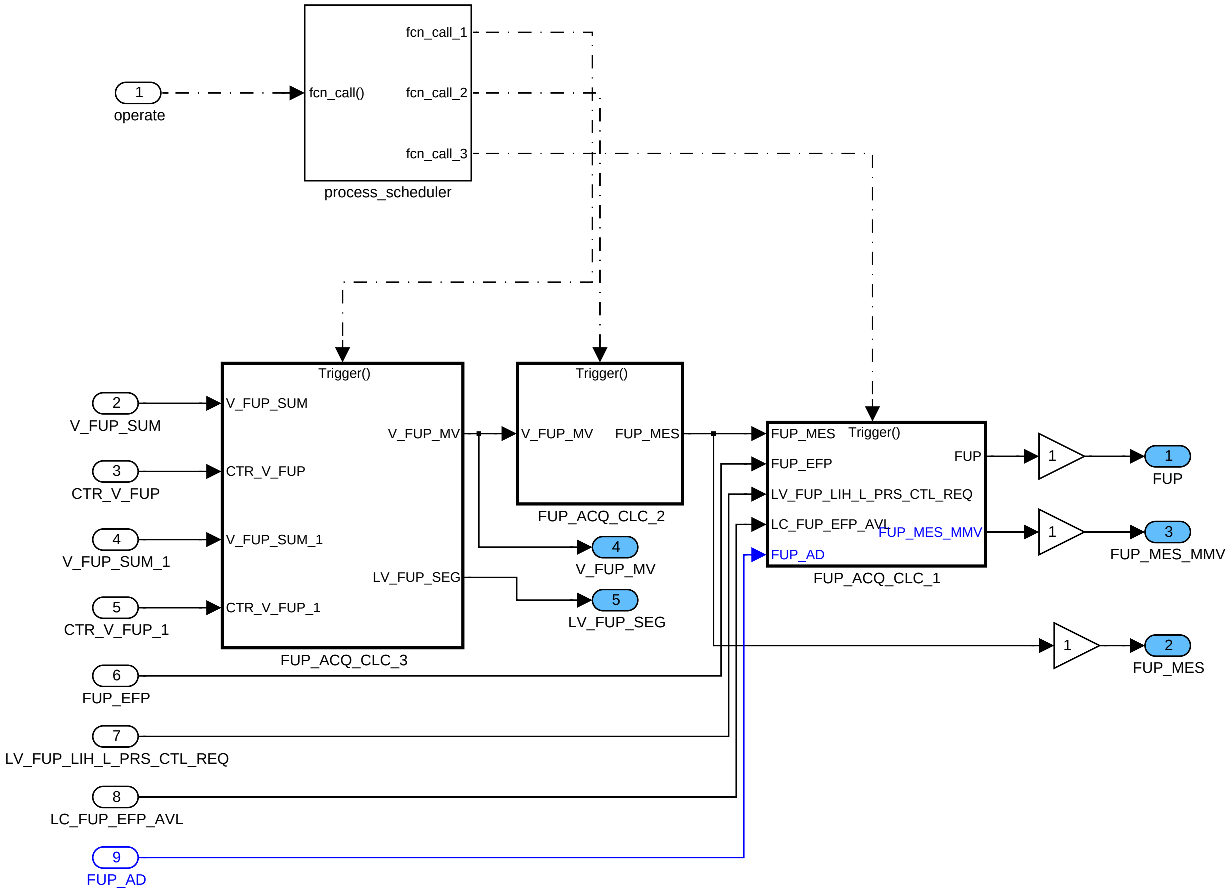


Figure 31 FUSL_M4018/ FUP_ACQ_CLC

FUSL_M4018/FUP_ACQ_CLC/FUP_ACQ_CLC_1

The liniarisation table gives the relationship between the sensor signal and the fuel pressure. This data has to be delivered by the sensor supplier.

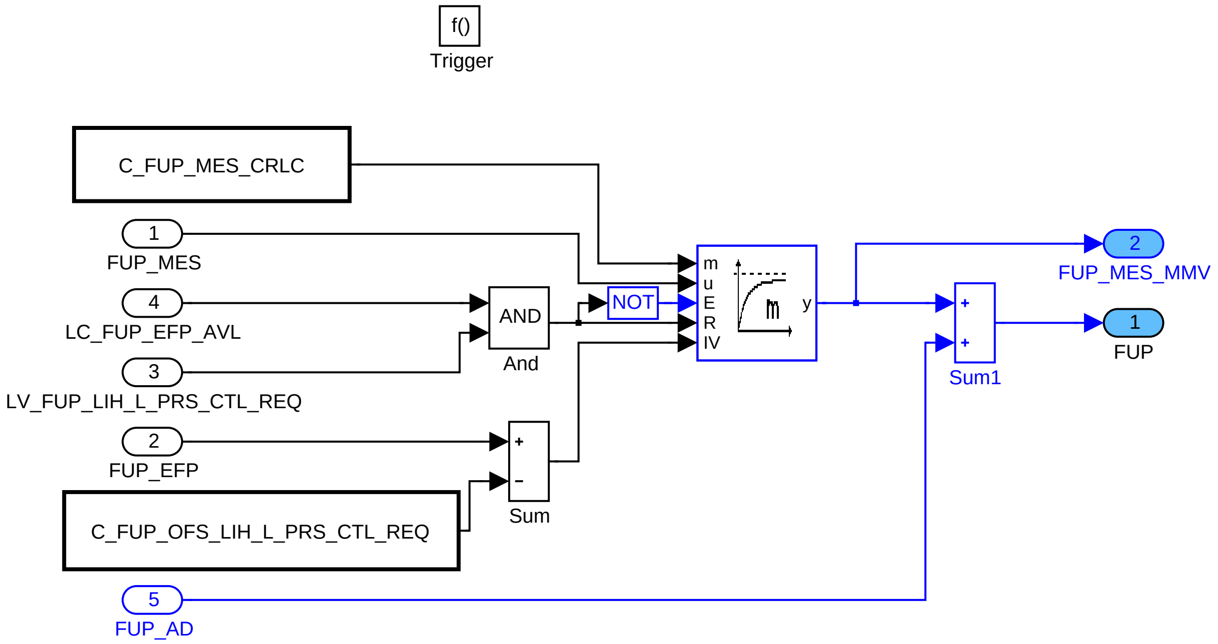


Figure 32 FUSL_M4018/ FUP_ACQ_CLC/ FUP_ACQ_CLC_1

FUSL_M4018/FUP_ACQ_CLC/FUP_ACQ_CLC_2

The correlation constant gives the filter constant for the fuel pressure. This filtering could be necessary to get a more stable signal for the fuel pressure.

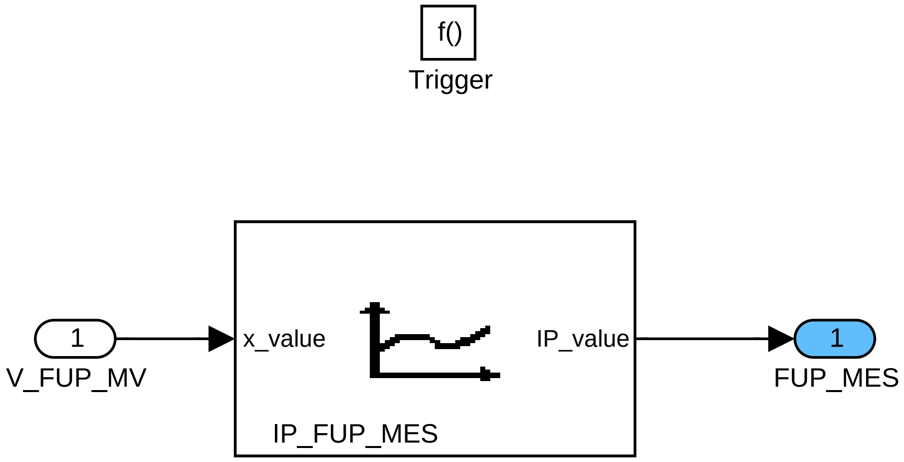
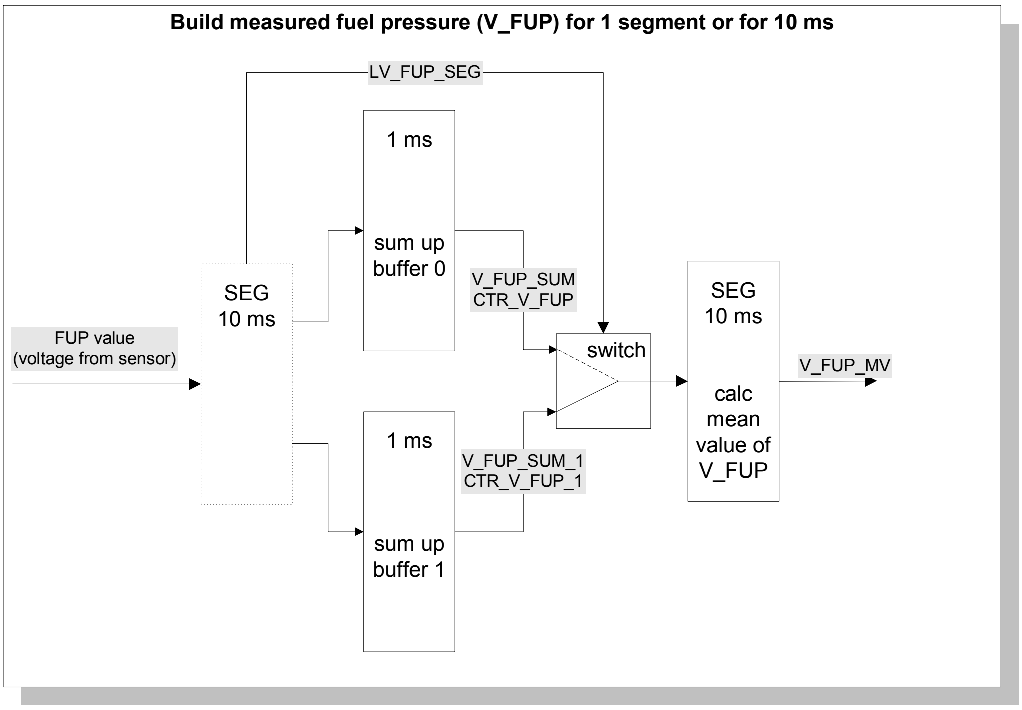


Figure 33 FUSL_M4018/ FUP_ACQ_CLC/ FUP_ACQ_CLC_2

FUSL_M4018/FUP_ACQ_CLC/FUP_ACQ_CLC_3



The raw value (voltage) for FUP is measured by continous conversion every 1 ms. The values (10 bit) are summed up in two alternating buffers, SUM_V_FUP or SUM_V_FUP_1. The numbers of values are counted in comparable buffers CTR_FUP or CTR_FUP_1. This mechanism is necessary to synchronize the measurement and the calculation of FUP_MES (build mean value with a standardized range) and will be done within BSW.

Depending on the logical variable LV_FUP_SEG one of the two buffers — SUM_V_FUP or SUM_V_FUP_1 — and the respective counter — CTR_V_FUP or CTR_V_FUP_1 — are selected for writing, the contents of the other buffer (counter) is used for calculations and diagnosis (after this buffer and the corresponding counter have been read they are cleared).

After having read out one buffer — SUM_V_FUP or SUM_V_FUP_1 — the buffer has to be cleared as well as the corresponding counter — CTR_V_FUP or CTR_V_FUP_1.

A two-buffer-system is used to avoid incorrect FUP calculation.

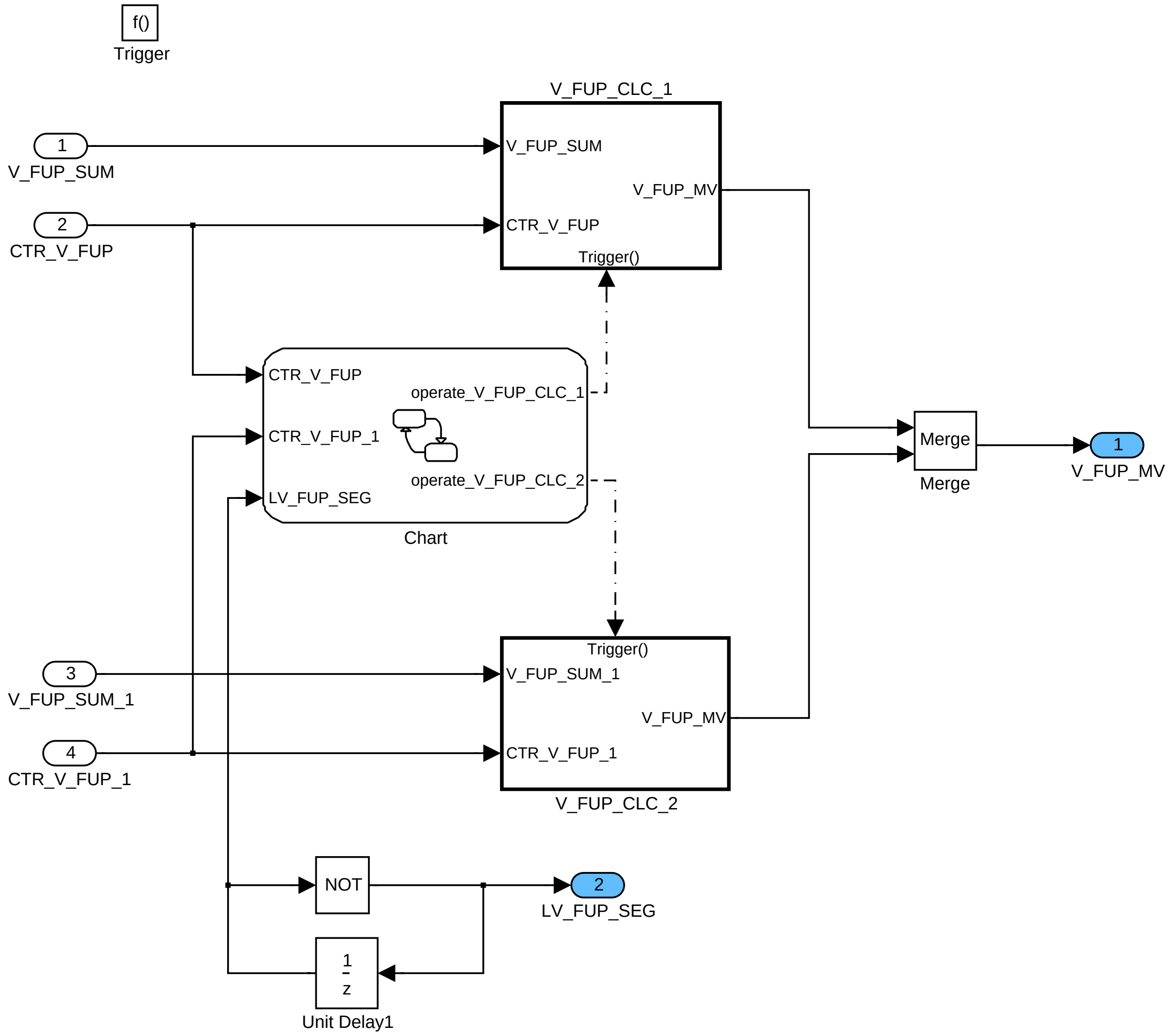


Figure 34 FUSL_M4018/ FUP_ACQ_CLC/ FUP_ACQ_CLC_3

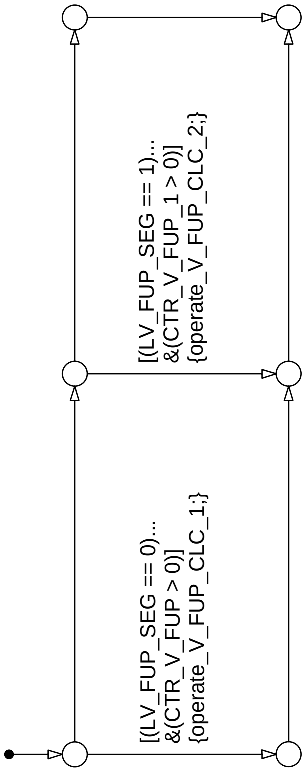


Figure 35 FUSL_M4018/ FUP_ACQ_CLC/ FUP_ACQ_CLC_3/ Chart

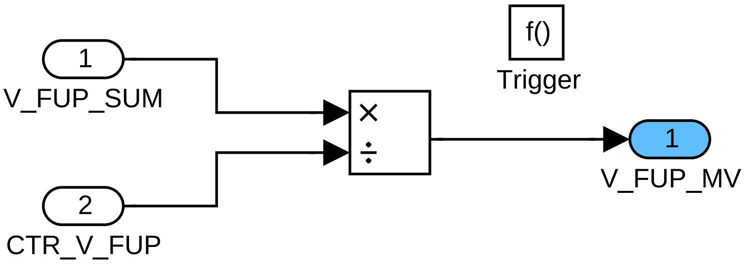


Figure 36 FUSL_M4018/ FUP_ACQ_CLC/ FUP_ACQ_CLC_3/ V_FUP_CLC_1

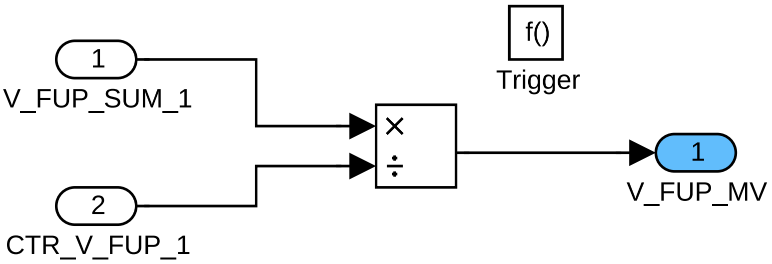


Figure 37 FUSL_M4018/ FUP_ACQ_CLC/ FUP_ACQ_CLC_3/ V_FUP_CLC_2

34.7 Fuel pressure acquisition of the electrical fuel pump

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FUP_EFP	O/V	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Fuel pressure EFP					
FUP_EFP_MES	O/V	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Measured Fuel pressure value in the feedline					
V_FUP_EFP_MV	O/V	0...7FFFH	0...4.99984741	1.52588E-4	V
Voltage of the low fuel pressure sensor (for diagnosis)					
FUP_EFP_MMV	V	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Mean fuel pressure value in the feedline					
V_FUP_EFP	V	0...7FFFH	0...4.99984741	1.52588E-4	V
Low fuel pressure EFP sensor raw acquisition					
FUP_EFP_CTR	-	0...FFFFH	0...6.5535E+4	1	-
Number of samples in buffer FUP_EFP_sum					
FUP_EFP_CTR_1	-	0...FFFFH	0...6.5535E+4	1	-
Number of samples in buffer FUP_EFP_sum_1					
FUP_EFP_SUM	-	0...FFFFFFFFH	0...2.09715E+7	0.00488281	V
Accumulated low fuel pressure sensor measurements for segment number (2n-1), n=1,2,3,...					
FUP_EFP_SUM_1	-	0...FFFFFFFFH	0...2.09715E+7	0.00488281	V
Accumulated low fuel pressure sensor measurements for segment number (2n), n=1,2,3,...					
LV_FUP_EFP_SEG	-	0...1H	0...1	1	-
idicates if segment number is even or odd					

Input data:

LV_ERR_FUP_EFP	LV_ES	ERR_SYM_FUP_EFP	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FUP_EFP_LIH	1	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Limp home value for fuel pressure of low pressure circuit (opening pressure of safety valve)					
C_FUP_EFP_MES_CRLC	1	0...FFH	0...0.99609375	0.00390625	-
Correlation constant for fuel pressure filter for the low pressure pump					
IP_FUP_EFP	3	0...FFFFH	0...1.73888E+5	2.6533608	hPa
LDP_V_FUP_EFP_IP_FUP_EFP	3	0...7FFFH	0...4.99984741	1.52588E-4	V
Fuel pressure sensor linearisation table					

34.7.1 Fuel pressure acquistion of low pressure signal

The raw value (voltage) for FUP_EFP is measured by continous conversion every 1 ms. The values (10 bit) are summed up in two alternating buffers, FUP_EFP_SUM or FUP_EFP_SUM_1. The numbers of values are counted in comparable buffers FUP_EFP_CTR or FUP_EFP_CTR_1. This mechanism is necessary to synchronize the measurement and the calculation of FUP_EFP_MES (build mean value with a standardized range)

Depending on the logical variable LV_FUP_EFP_SEG one of the two buffers — FUP_EFP_SUM or FUP_EFP_SUM_1 — and the respective counter — FUP_EFP_CTR or

FUP_EFP_CTR_1 — are selected for writing, the contents of the other buffer (counter) is used for calculations and diagnosis (after this buffer and the corresponding counter have been read they are cleared).

After having read out one buffer — FUP_EFP_SUM or FUP_EFP_SUM_1 — the buffer has to be cleared as well as the corresponding counter — FUP_EFP_CTR or FUP_EFP_CTR_1.

A two-buffer-system is used to avoid incorrect FUP_EFP calculation.

The fuel pressure of the feedline is calculated by means of the fuel pressure sensor for the feedline. Additionally in case of sensor error a substitude value will be taken.

Use **ACTION_INFR_GetVFupEfpSens(OUT** <v_fup_efp>**)**

Application Condition

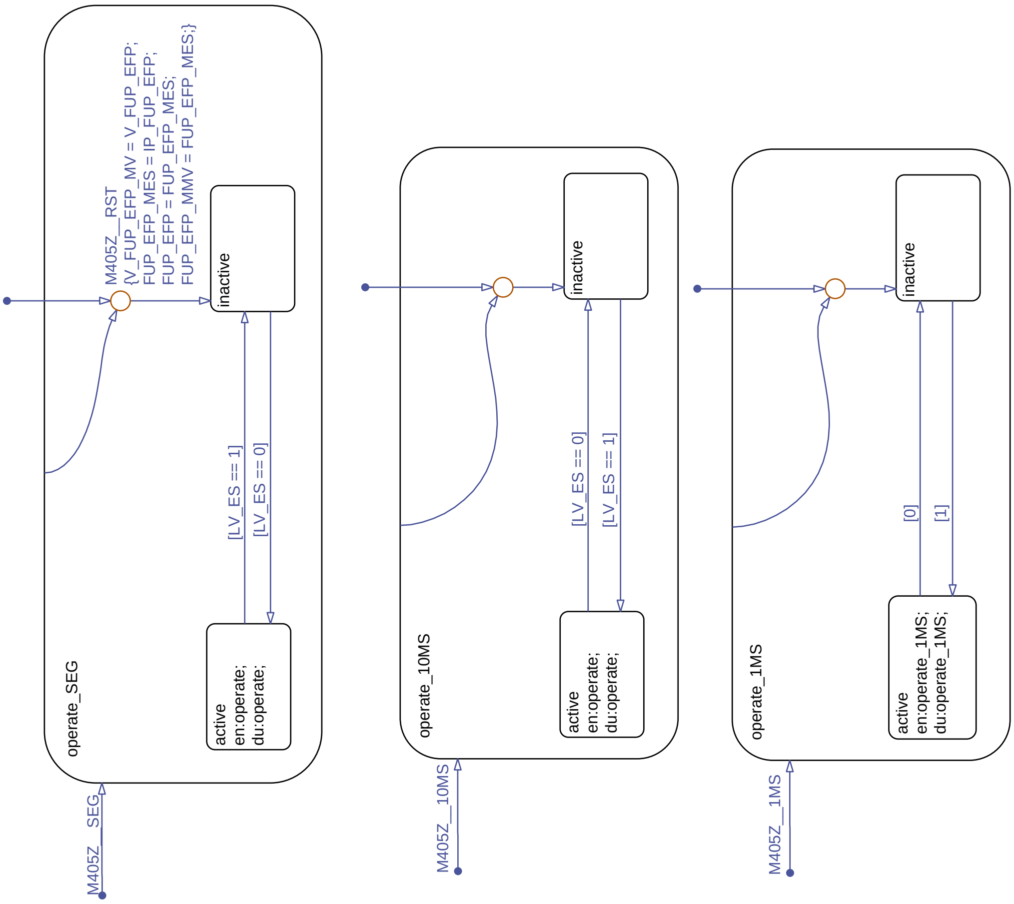


Figure 38 M405Z/ APP_CDN/ Chart

Function Description

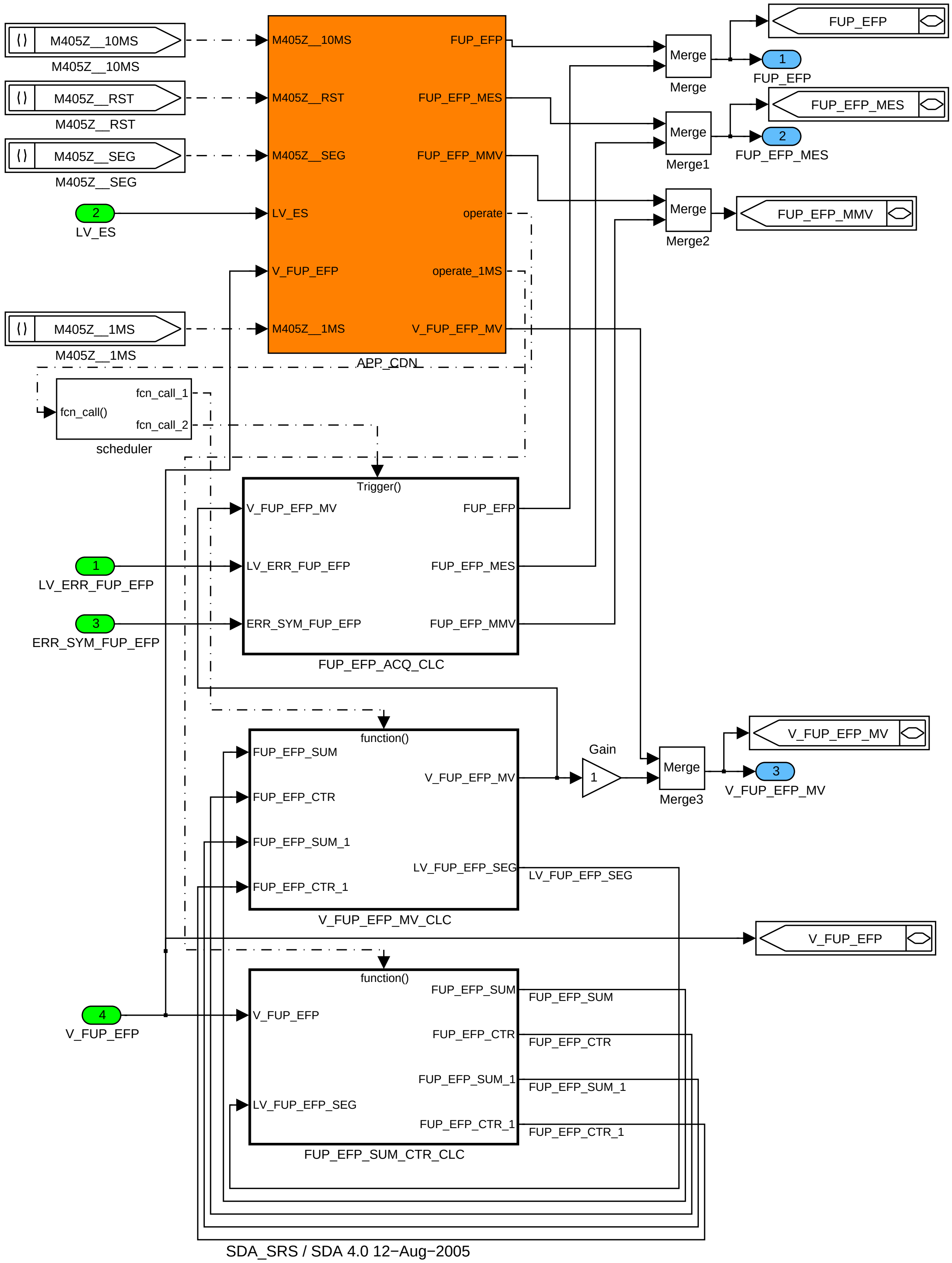


Figure 39 M405Z

34.7.1.1 <HEADER MODULE="M405Z/V_FUP_EFP_MV_CLC">

Calculation of V_FUP_EFP_MV.

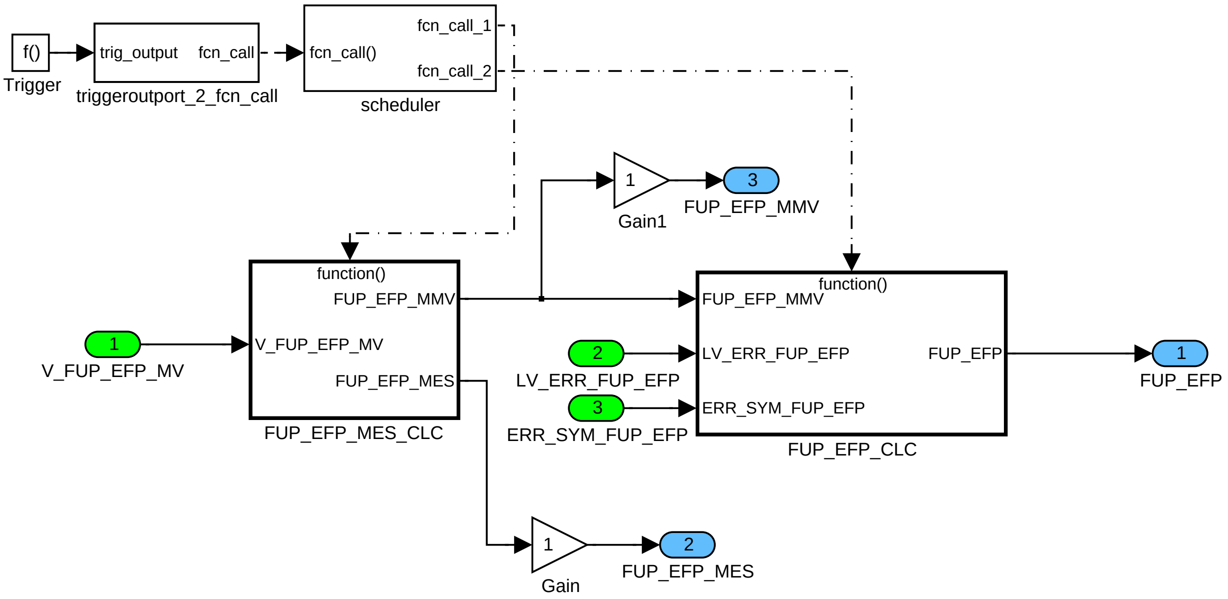


Figure 40 M405Z/ FUP_EFP_ACQ_CLC

Fuel pressure of the electrical fuel pump calculation

Depending on different failure bits the mean value of the fuel pressure sensor or a subsitude value (opening point overpressure valve) is taken as FUP_EFP.

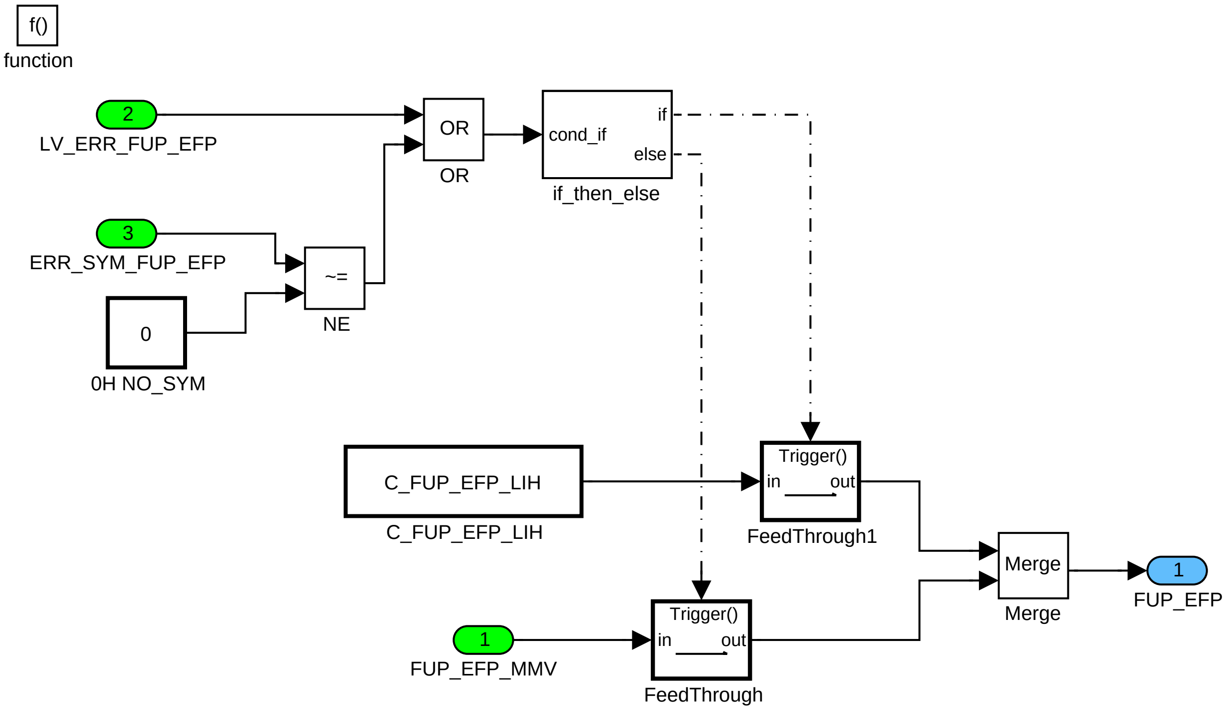


Figure 41 M405Z/ FUP_EFP_ACQ_CLC/ FUP_EFP_CLC

Sensor conversion

The sensor value is converted via calibrated curve into a measured value. This value is then filtered.

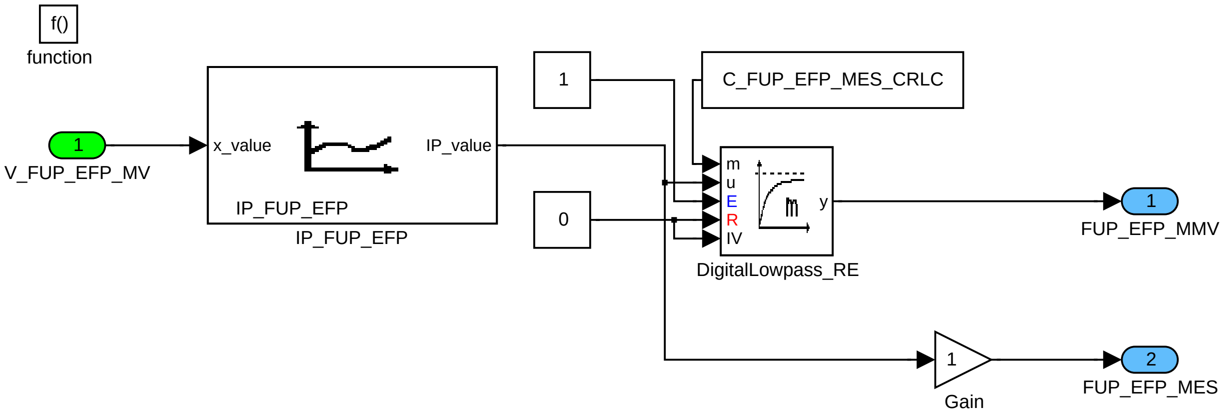


Figure 42 M405Z/ FUP_EFP_ACQ_CLC/ FUP_EFP_MES_CLC

34.7.1.2 M405Z/FUP_EFP_SUM_CTR_CLC

Calculation of FUP_EFP_SUM, FUP_EFP_CTR, FUP_EFP_SUM_1 and FUP_EFP_CTR_1

general specification

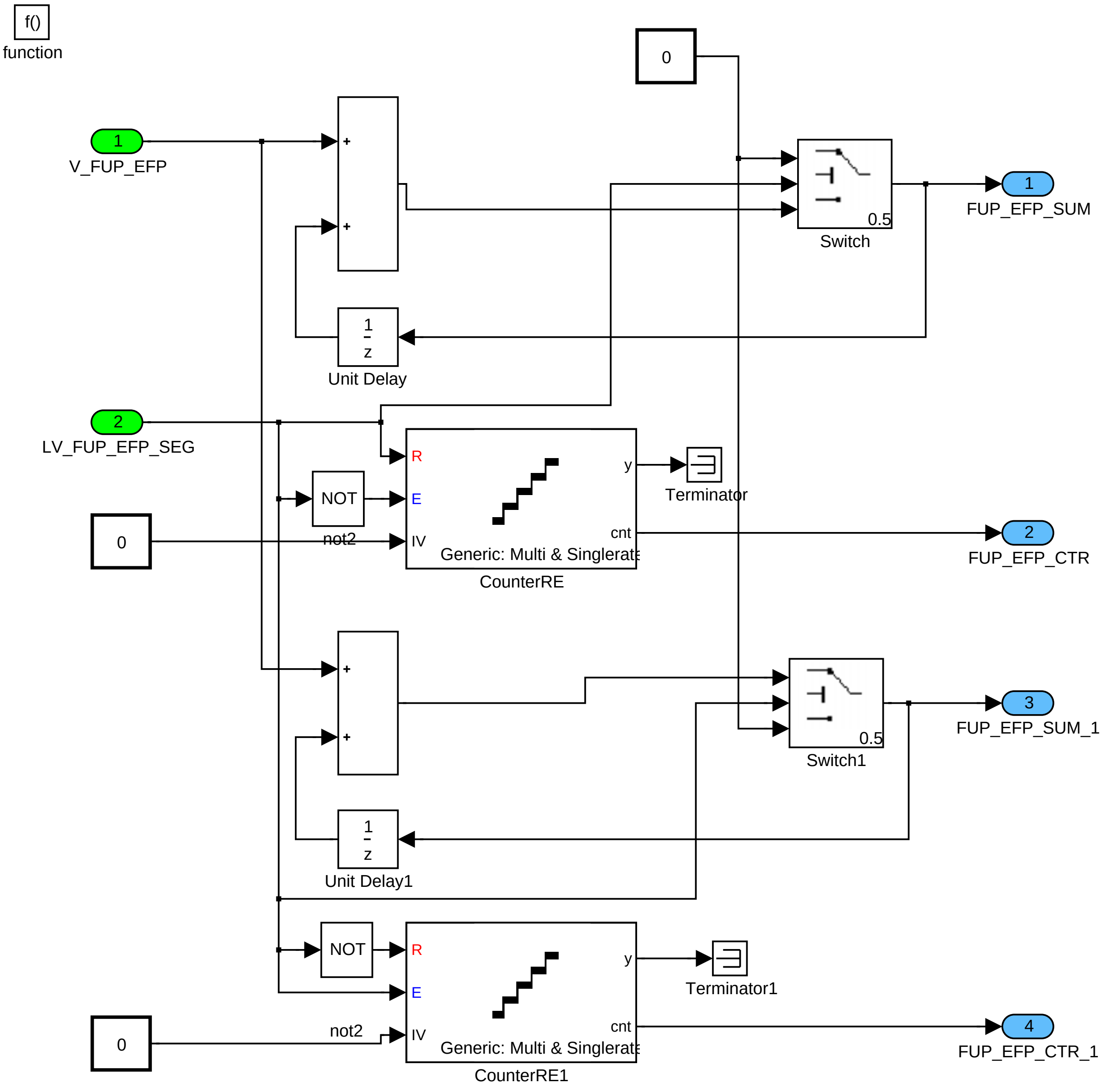


Figure 43 M405Z/ FUP_EFP_SUM_CTR_CLC

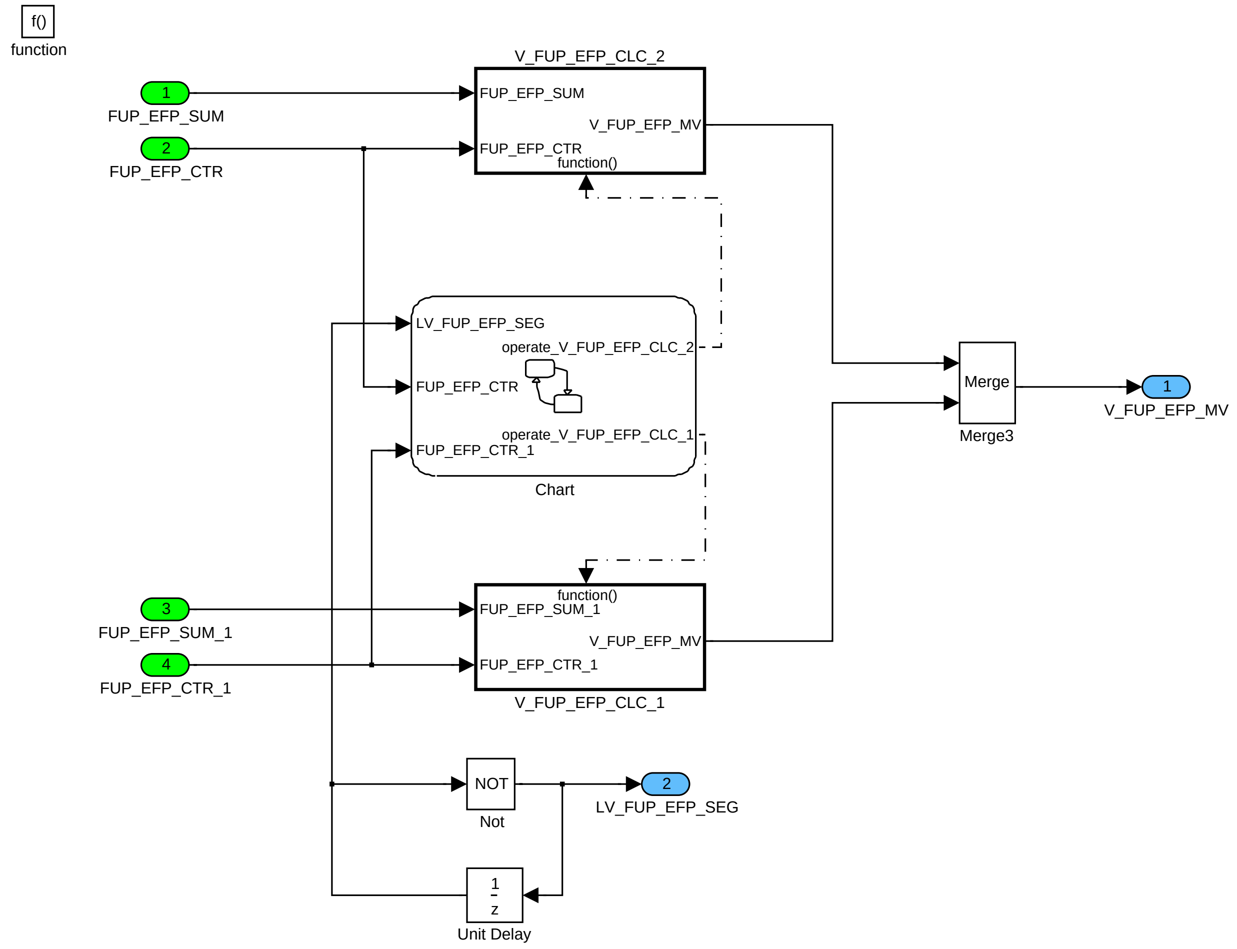


Figure 44 M405Z/ V_FUP_EFP_MV_CLC

1.1.3 M405Z/ V_FUP_EFP_MV_CLC/ Chart

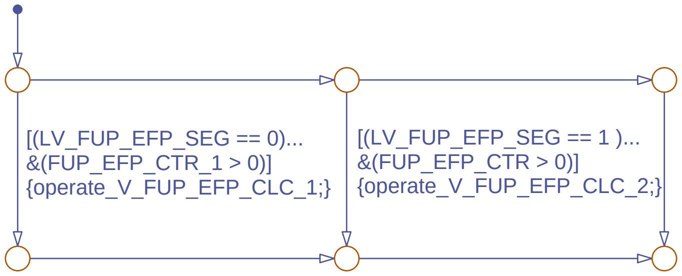


Figure 45 M405Z/ V_FUP_EFP_MV_CLC/ Chart

M405Z/V_FUP_EFP_MV_CLC/V_FUP_EFP_CLC_1

Calculation of V_FUP_EFP_MV.

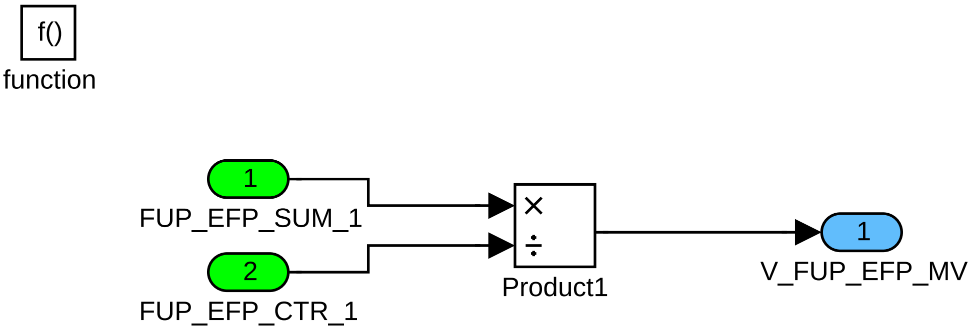


Figure 46 M405Z/ V_FUP_EFP_MV_CLC/ V_FUP_EFP_CLC_1

M405Z/V_FUP_EFP_MV_CLC/V_FUP_EFP_CLC_2

Calculation of V_FUP_EFP_MV.

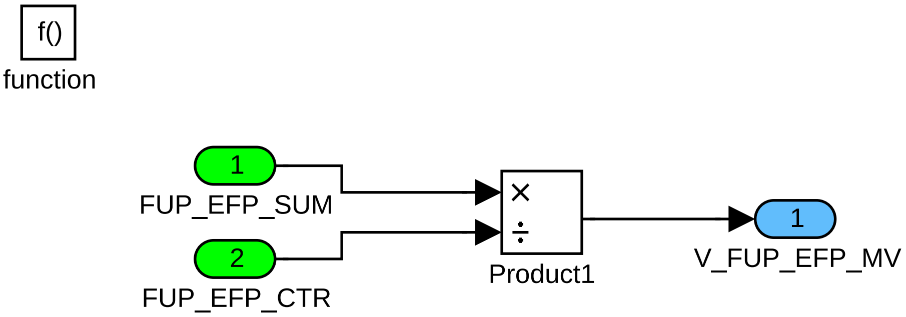


Figure 47 M405Z/ V_FUP_EFP_MV_CLC/ V_FUP_EFP_CLC_2

34.8 Fuel pressure setpoint of electrical fuel pump

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FUP_EFP_SP	O/V	0...FFFFH	0...1.73888E+5	2.653361	hPa
Fuel pressure setpoint of electrical fuel pump					
FUP_EFP_SP_TMP	V	0...FFFFH	0...1.73888E+5	2.653361	hPa
Temporary value of fuel pressure setpoint of electrical fuel pump before limitation					

Input data:

LV_PU	LV_PUC	MFF_SP_FUP_CTL	N_32
STATE_EFP	TFU	LV_FUP_EFP_SP_EXT_REQ	LV_FUP_LIH_L_PRS_CTL_REQ
TCO_ST	T_AST		

Calibration data:

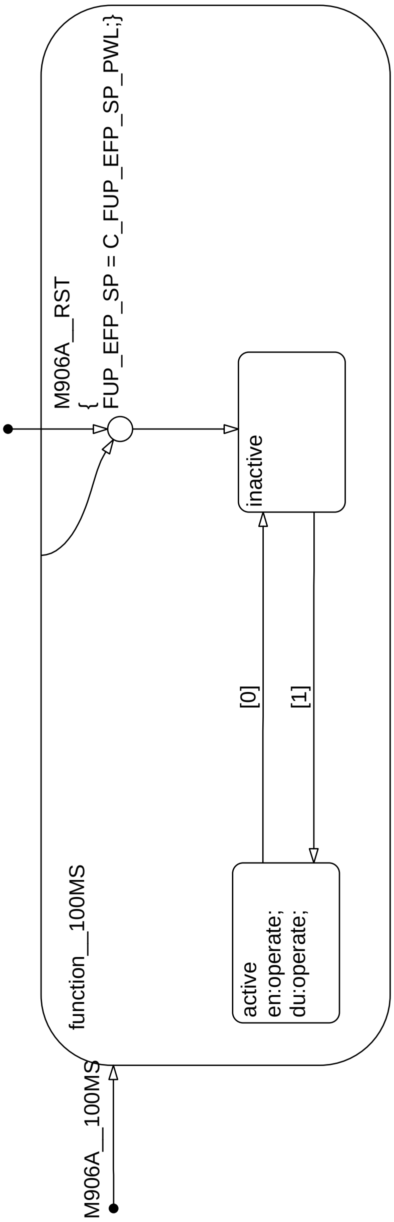
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FUP_EFP_SP_EXT_REQ	1	0...FFFFH	0...1.73888E+5	2.653361	hPa
External FUP_EFP_SP					
C_FUP_EFP_SP_HPP_LIH	1	0...FFFFH	0...1.73888E+5	2.653361	hPa
Fuel pressure setpoint for high pressure pump limp home					
C_FUP_EFP_SP_LIM_DEC	1	0...FFFFH	0...1.73888E+5	2.653361	hPa
Decrement for fuel pressure setpoint limitation					
C_FUP_EFP_SP_LIM_INC	1	0...FFFFH	0...1.73888E+5	2.653361	hPa
Increment for fuel pressure setpoint limitation					
C_FUP_EFP_SP_MAN	1	0...FFFFH	0...1.73888E+5	2.653361	hPa
Manual FUP_EFP_SP					
C_FUP_EFP_SP_PUC	1	0...FFFFH	0...1.73888E+5	2.653361	hPa
Low fuel pressure setpoint in PU/PUC					
C_FUP_EFP_SP_PWL	1	0...FFFFH	0...1.73888E+5	2.653361	hPa
Low fuel pressure setpoint during power latch phase of EFP					
LC_FUP_EFP_SP_MAN_ACT	1	0...1H	0...1	1	-
Switch for manual FUP_EFP_SP					
IP_FUP_EFP_SP_HST	8	0...FFFFH	0...1.73888E+5	2.653361	hPa
LDPM_MFF_SP_FUP_CTL_IP_EFP_MIN	8	0...FFFFH	0...1.389E+3	0.021194781	mg
Map with the FUP_EFP setpoint for the hot start					
IP_T_FUP_EFP_SP_HST	3	1...FFFFH	0.01...655.35	0.01	s
LDP_TCO_ST_IP_T_FUP_EFP_SP_HST	3	0...FEH	-48...142.5	0.75	°C
Map with the time for the FUP_EFP setpoint for the hot start					
IP_FUP_EFP_SP	4x4	0...FFFFH	0...1.73888E+5	2.653361	hPa
LDP_MFF_SP_FUP_CTL_IP_EFP_SP	4	0...FFFFH	0...1.389E+3	0.021194781	mg
LDP_N_32_IP_FUP_EFP_SP	4	0...FFH	0...8.16E+3	32	rpm
Fuel pressure setpoint of low pressure pump for normal operation					
IP_FUP_EFP_SP_MIN	8x8	0...FFFFH	0...1.73888E+5	2.653361	hPa
LDP_TFU_IP_FUP_EFP_SP	8	0...FEH	-48...142.5	0.75	°C
LDPM_MFF_SP_FUP_CTL_IP_EFP_MIN	8	0...FFFFH	0...1.389E+3	0.021194781	mg
Minimum fuel pressure setpoint of low pressure pump					

34.8.1 Fuel pressure setpoint of low pressure pump

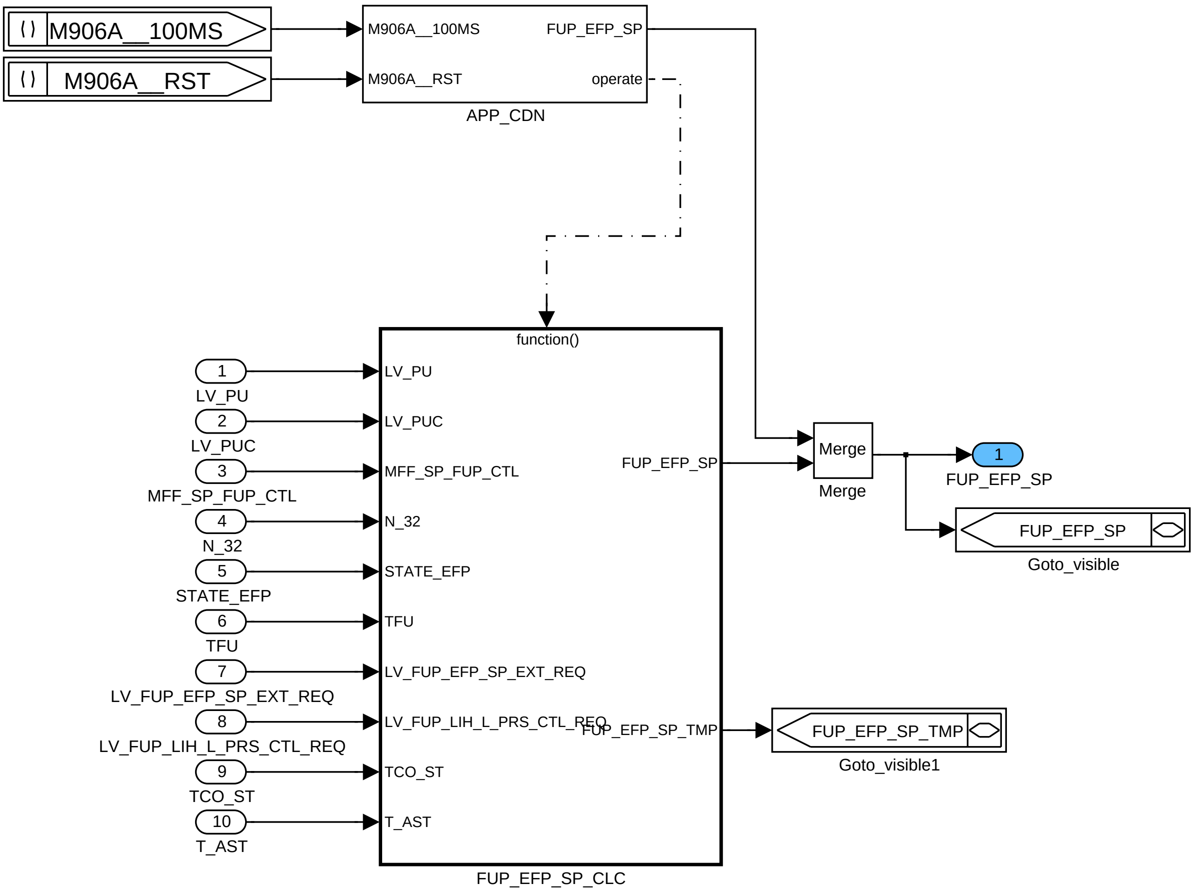
The module fuel pressure setpoint of electrical fuel pump delivers a fuel setpoint to the module fuel pressure control.

The fuel pressure setpoint of the low pressure pump is normally taken out of a map depending on engine speed and injected fuel mass. Exemptions are power latch phase of the electrical fuel pump and the pull mode. Additionally a testbench function is included. In order to avoid to big fuel pressure setpoint gradients a certain gradient limitation is included.

Application Condition



Function Description

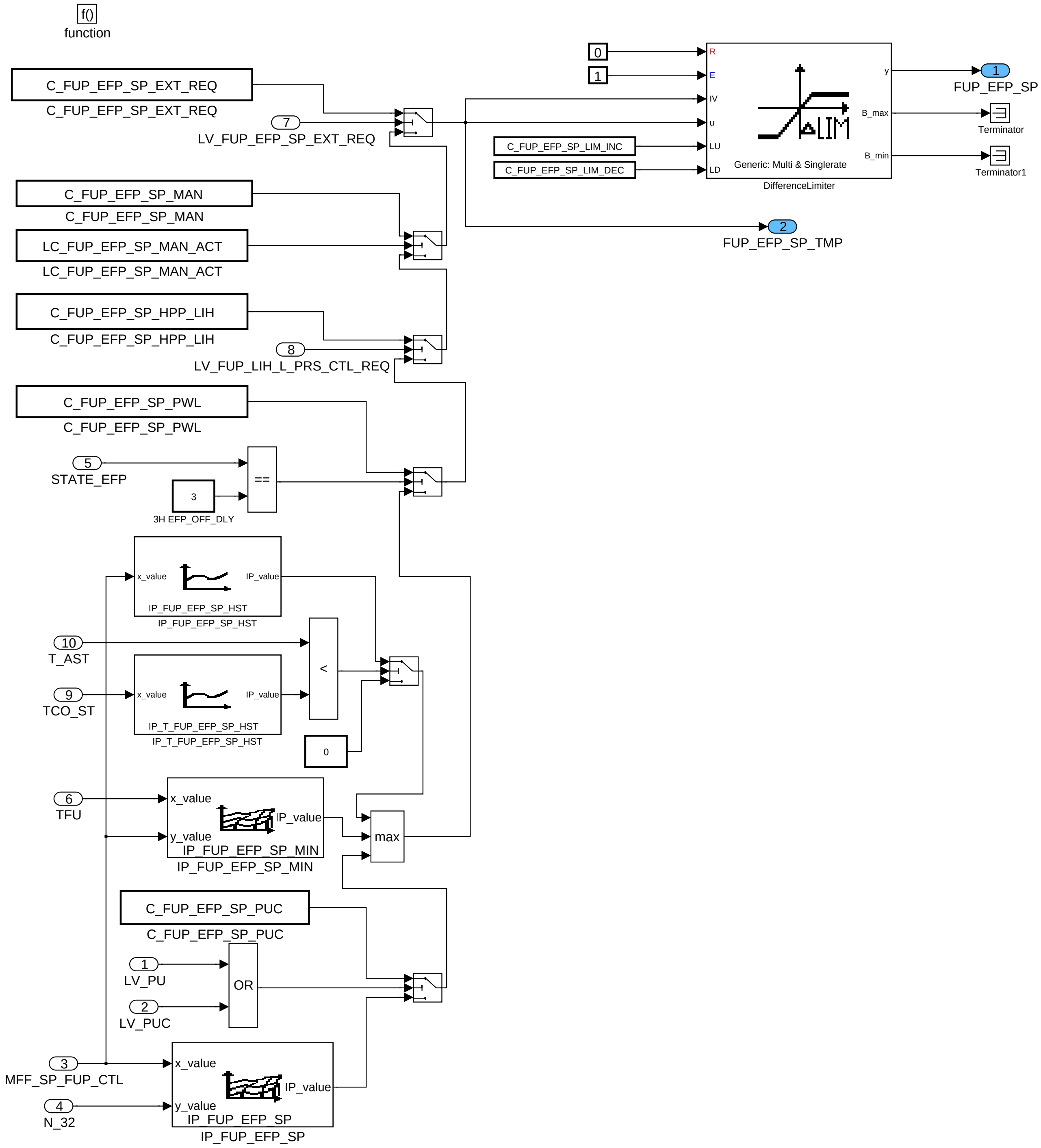


SDA_SRS / SDA 3.1 07−Jul−2004

M906A

34.8.1.1 Fuel pressure setpoint calculation

In this part the fuel pressure setpoint is calculated.



M906A/FUP_EFP_SP_CLC

34.9 Electrical fuel pump control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EFPPWM	O/V	0...FFFFH	0...99.9984741	0.00152588	%
Pump speed of the electrical fuel pump as PWM signal					
LV_EFP	O/V	0...1H	0...1	1	-
Logical variable for electrical fuel pump active					
STATE_EFP	O/V	0H 1H 2H 3H 4H 5H 6H	EFP_OFF EFP_ACR_TEST EFP_ON EFP_OFF_DLY EFP_EXT_ADJ EFP_CRASH EFP_LIH	1	-
States of electrical fuel pump function					
EFPPWM_I	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
I-Part of the controller					
EFPPWM_I_AD	O/V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
Adaptive I-Part of the controller					
EFPPWM_MIN_AD	O/V/S	0...FFFFH	0...99.9984741	0.00152588	%
Minimum pump speed of the electrical fuel pump as PWM signal					
LV_EFPPWM_AD_STOP	O/V	0...1H	0...1	1	-
Logical variable for adaptation stop					
LV_EFPPWM_I_AD_VLD	O/V/S	0...1H	0...1	1	-
Logical variable showing valid adaptive I-Part of EFPPWM					
LV_EFPPWM_MIN_VLD	O/V/S	0...1H	0...1	1	-
Logical variable showing valid minimum of EFPPWM					
VFF_EFP_REQ	O/V	0...FFFFH	0...255	0.00389105	l/h
Requested amount of fuel from the pump					
EFPPWM_BAS	V	0...FFFFH	0...99.9984741	0.00152588	%
Precontroller value of EFPPWM					
CTR_SEG_EFPPWM_I_AD	V	0...FFFFH	0...6.5535E+4	1	-
Segment counter before I-part adaptation is allowed					
FUP_EFP_GRD	V	8000...7FFFH	-8.6945E+4 ... 8.69427E+4	2.6533608	hPa/10ms
Calculated requested amount of fuel from the pump					
VFF_EFP_CLC	V	0...FFFFH	0...255	0.00389105	l/h
Calculated requested amount of fuel from the pump					
LV_EFPPWM_CTL_STOP	V	0...1H	0...1	1	-
Logical variable for I-part freeze					
LV_EFPPWM_MIN_AD_ACT	V	0...1H	0...1	1	-
Logical variable showing active minimum EFPPWM adaptation					
LV_FUP_EFP_LIM	V	0...1H	0...1	1	-
Logical variable indicating VFF limitation for the EFP					
VFF_EFP_COR	V	0...FFFFH	0...255	0.00389105	l/h
Volume fuel flow for correction of the EFP volume fuel flow					
FUP_EFP_DIF	V	8000...7FFFH	-8.6945E+4 ... 8.69427E+4	2.6533608	hPa
Deviation of the fuel pressure after fuel pump from the setpoint					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_EFPPWM_CTL_RST	V	0...1H	0...1	1	-
Logical variable for controller reset					
EFPPWM_I_TMP	V	8000...7FFFH	-50...49.9984741	0.00152588	%
Temporary I-Part of the controller					
EFPPWM_I_AD_DIF	V	8000...7FFFH	-50...49.9984741	0.00152588	%
Adaptive I-Part difference of the controller					
EFPPWM_P	V	8000...7FFFH	-50...49.9984741	0.00152588	%
P-Part of the controller					
CTR_SEG_EFPPWM_MIN_AD	V	0...FFFFH	0...6.5535E+4	1	-
Segment counter before minimum PWM adaptation is allowed					

Input data:

STATE_EFP_TRAN	MFF_SP_FUP_CTL	TFU_EFP	VFF_EFP_ADD
TCO	LV_PUC	T_AST	FUP_EFP_SP
FUP_EFP	ERR_SYM_FUP_EFP	T_SEG_AV	N_32
LV_PIN_ICH	LV_ST_END	FUP_SP_TMP	VFF_VCV
LV_FUP_LIH_HOM_VCV_OPEN_REQ	TFU	FUP_DIF	LV_EFPPWM_EXT_ADJ
EFPPWM_EXT_ADJ	LV_EFP_CTL_AD_CLR_EXT_REQ		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_EFPPWM_I_AD	1	0...FFH	0...0.99609375	0.00390625	-
Correlation constant for I-part adaptation					
C_CTR_FUP_SP_LIM_VFF_EFP_COR	1	0...FFFFH	0...6.5535E+4	1	-
Counter for duration of the limitation for VFF_EFP					
C_CTR_SEG_EFPPWM_I_AD_MAX	1	0...FFFFH	0...6.5535E+4	1	-
Maximum segment counter for the I-part adaptation					
C_CTR_SEG_EFPWPM_MIN_AD_MAX	1	0...FFFFH	0...6.5535E+4	1	-
Maximum segment counter for the minimum EFPPWM adaptation					
C_EFPPWM_ACR_TEST_INC	1	0...FFFFH	0...99.9984741	0.00152588	%
Increment for the actuator test					
C_EFPPWM_CRASH	1	0...FFFFH	0...99.9984741	0.00152588	%
Min. electrical fuel pump speed at state EFP_CRASH					
C_EFPPWM_DEAC	1	0...FFFFH	0...99.9984741	0.00152588	%
Minimum EFPPWM at ACR_TEST					
C_EFPPWM_I_AD_MAX	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Max. value for the adaptive I-part of the controller					
C_EFPPWM_I_AD_MIN	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Min. value for the adaptive I-part of the controller					
C_EFPPWM_I_MAX	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Max. value for the I-part of the controller					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EFPPWM_I_MAX_LIH_HOM_VCV_OPEN	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Max. value for the I-part of the controller in case of VCV open limphome					
C_EFPPWM_I_MIN	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Min. value for the I-part of the controller					
C_EFPPWM_I_MIN_LIH_HOM_VCV_OPEN	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Min. value for the I-part of the controller in case of VCV open limphome					
C_EFPPWM_LIH_ADD	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Additive EFPPWM in case of limp home					
C_EFPPWM_MAN_ACT	1	0...FFFFH	0...99.9984741	0.00152588	%
Manual value for the EFPPWM					
C_EFPPWM_MAX	1	0...FFFFH	0...99.9984741	0.00152588	%
Max. electrical fuel pump speed					
C_EFPPWM_MIN_1	1	0...FFFFH	0...99.9984741	0.00152588	%
Min. electrical fuel pump speed at state EFP_ON					
C_EFPPWM_MIN_2	1	0...FFFFH	0...99.9984741	0.00152588	%
Min. electrical fuel pump speed at state EFP_OFF					
C_EFPPWM_MIN_AD_INI	1	0...FFFFH	0...99.9984741	0.00152588	%
Initial value for the EFPPWM_MIN adaptation					
C_EFPPWM_MIN_AD_TOL	1	0...FFFFH	0...99.9984741	0.00152588	%
Maximum value for the EFPPWM_MIN adaptation					
C_EFPPWM_MIN_COR	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Correction value for the EFPPWM to reach ‘zero delivery’					
C_EFPPWM_MIN_LIH_HOM_VCV_OPEN	1	0...FFFFH	0...99.9984741	0.00152588	%
Min. electrical fuel pump speed in case of VCV open limphome					
C_FUP_EFP_DIF_MAX_EFPPMW_AD	1	8000...7FFFH	-8.6945E+4 ... 8.69427E+4	2.6533608	hPa
Maximum deviation of the fuel pressure after fuel pump from the setpoint for adaptation					
C_FUP_EFP_GRD_MAX	1	8000...7FFFH	-8.6945E+4 ... 8.69427E+4	2.6533608	hPa/10ms
Fuel pressure gradient for freezing EFPPWM during PUC					
C_FUP_EFP_SP_MAX_EFPPWM_MIN_AD	1	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Maximum FUP EFP setpoint for the minimum adaptation of EFPPWM					
C_FUP_EFP_SP_MIN_EFPPWM_MIN_AD	1	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Minimum FUP EFP setpoint for the minimum adaptation of EFPPWM					
C_FUP_RHO_FUEL_EFP	1	153B...FFFFH	74.6395056...900	0.01373312	mg/cm**3
Density of the fuel					
C_FUP_SP_DIF_LIM_VFF_EFP_COR	1	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Fuel pressure setpoint difference threshold for activation of the VFF_EFP correction					
C_MFF_SP_MAX_EFPPWM_I_AD	1	0...FFFFH	0...1.389E+3	0.02119478	mg
Maximum injected fuel mass for allowing I-part adaptation					
C_MFF_SP_MAX_EFPPWM_MIN_AD	1	0...FFFFH	0...1.389E+3	0.02119478	mg
Maximum injected fuel mass for allowing minimum EFPPWM adaptation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MFF_SP_MIN_EFPPWM_I_AD	1	0...FFFFH	0...1.389E+3	0.02119478	mg
Minimum injected fuel mass for allowing I-part adaptation					
C_MFF_SP_MIN_EFPPWM_MIN_AD	1	0...FFFFH	0...1.389E+3	0.02119478	mg
Minimum injected fuel mass for allowing minimum EFPPWM adaptation					
C_N_MAX_EFPPWM_I_AD	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for allowing I-part adaptation					
C_N_MAX_EFPPWM_MIN_AD	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for allowing minimum EFPPWM adaptation					
C_N_MIN_EFPPWM_I_AD	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for allowing I-part adaptation					
C_N_MIN_EFPPWM_MIN_AD	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for allowing minimum EFPPWM adaptation					
C_TCO_MAX_EFPPWM_AD	1	0...FEH	-48...142.5	0.75	°C
Maximum coolant temperature for allowing adaptation					
C_TCO_MIN_EFPPWM_AD	1	0...FEH	-48...142.5	0.75	°C
Minimum coolant temperature for allowing adaptation					
C_T_AST_EFPPWM_MIN_AD	1	0...FFFFH	0...6.5535E+3	0.1	s
Time after engine start to allow minimum EFPPWM adaptation					
C_VFF_DIF_LIM_VFF_EFP_COR	1	0...FFFFH	0...255	0.00389105	l/h
Constant for manual VFF difference for VFF_EFP correction					
C_VFF_EFP_COR_GRD_DEC	1	0...FFFFH	0...255	0.00389105	l/h
Decrement gradient limitation for VFF_EFP_COR					
C_VFF_EFP_REQ_LIH	1	0...FFFFH	0...255	0.00389105	l/h
EFP volume fuel flow value for limp home of the fuel low pressure system					
LC_EFPPWM_I_AD_CLR	1	0...1H	0...1	1	-
Logical variable to clear the adaptive I-Part of the controller					
LC_FUP_EFP_AVL	1	0...1H	0...1	1	-
Logical variable indicating ‘ fuel pressure sensor for low pressure line available’					
LC_VFF_DIF_LIM_VFF_EFP_COR_MAN	1	0...1H	0...1	1	-
Logical constant for switch to manual VFF_DIF for VFF_EFP correction					
IP_EFPPWM_BAS_L_POW	8	0...FFFFH	0...99.9984741	0.00152588	%
LDP_VFF_EFP_REQ_IP_EFPPWM_BAS	8	0...FFFFH	0...255	0.00389105	l/h
Basic value of the EFPPWM for low power version					
IP_EFPPWM_I	8	0...FFFFH	-50...49.9984741	0.00152588	%
LDP_FUP_EFP_DIF_IP_EFPPWM_I	8	0...FFFFH	-8.6945E+4 ... 8.69427E+4	2.6533608	hPa
I-part of the controller					
IP_EFPPWM_I_VCV_LIH	8	0...FFFFH	-50...49.9984741	0.00152588	%
LDP_FUP_DIF_IP_EFPPWM_I_VCV_LIH	8	0...FFFFH	-1.7389E+5 ... 1.73884E+5	5.30669108	hPa
I-part of the controller in case of VCV open limphome					
IP_EFPPWM_P	8	0...FFFFH	-50...49.9984741	0.00152588	%
LDP_FUP_EFP_DIF_IP_EFPPWM_P	8	0...FFFFH	-8.6945E+4 ... 8.69427E+4	2.6533608	hPa
P-part of the controller					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_EFPPWM_P_VCV_LIH	8	0...FFFFH	-50...49.9984741	0.00152588	%
LDP_FUP_DIF_IP_EFPPWM_P_VCV_LIH	8	0...FFFFH	-1.7389E+5 ... 1.73884E+5	5.30669108	hPa
p-part of the controller in case of VCV open limphome					
IP_FUP_EFP_SP_COR_VCV_OPEN	8	0...FFFFH	-8.6945E+4 ... 8.69427E+4	2.6533608	hPa
LDP_TFU_IP_FUP_EFP_SP_COR_VCV	8	0...FEH	-48...142.5	0.75	°C
Correction for EFP in case of VCV open lih					
IP_FAC_EFPPWM_BAS_L_POW	6x8	0...FFFFH	0...1.99996948	3.05176E-5	-
LDP_TFU_EFP_IP_FAC_EFPPWM_BAS	6	0...FEH	-48...142.5	0.75	°C
LDP_FUP_EFP_SP_IP_FAC_EFPPWM	8	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Correction factor for the basic value of the EFPPWM for low power version					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_T_DLY_EFP_ACR_TEST_MAX	1	0...FFFFH	0...655.35	0.01	s
Duration time for the actuator test at EFPPWM = 100 % till switch off (value: 5 s)					

34.9.1 Electrical fuel pump control with fuel pressure sensor for feed pressure

General information:

The electrical fuel pump is controlled by the EFPPWM.

The PI-controller is used to compensate deviations from the nominal fuel pressure after the fuel pump. If no sensor is applied the EFPPWM is only calculated with the help of the precontrol map, which corresponds to the pump data.

In order to control also deviations caused by external influences or aging two adaptation strategies are applied. One adaptation is used in a nominal operating point where the I-part of the controller is taken as an adaptive I-part. The other adaptation gives the minimum EFPPWM for ensuring the correct fuel pressure for the jet pumps without increasing the fuel pressure to the level where the overpressure release valve opens.

Application Condition

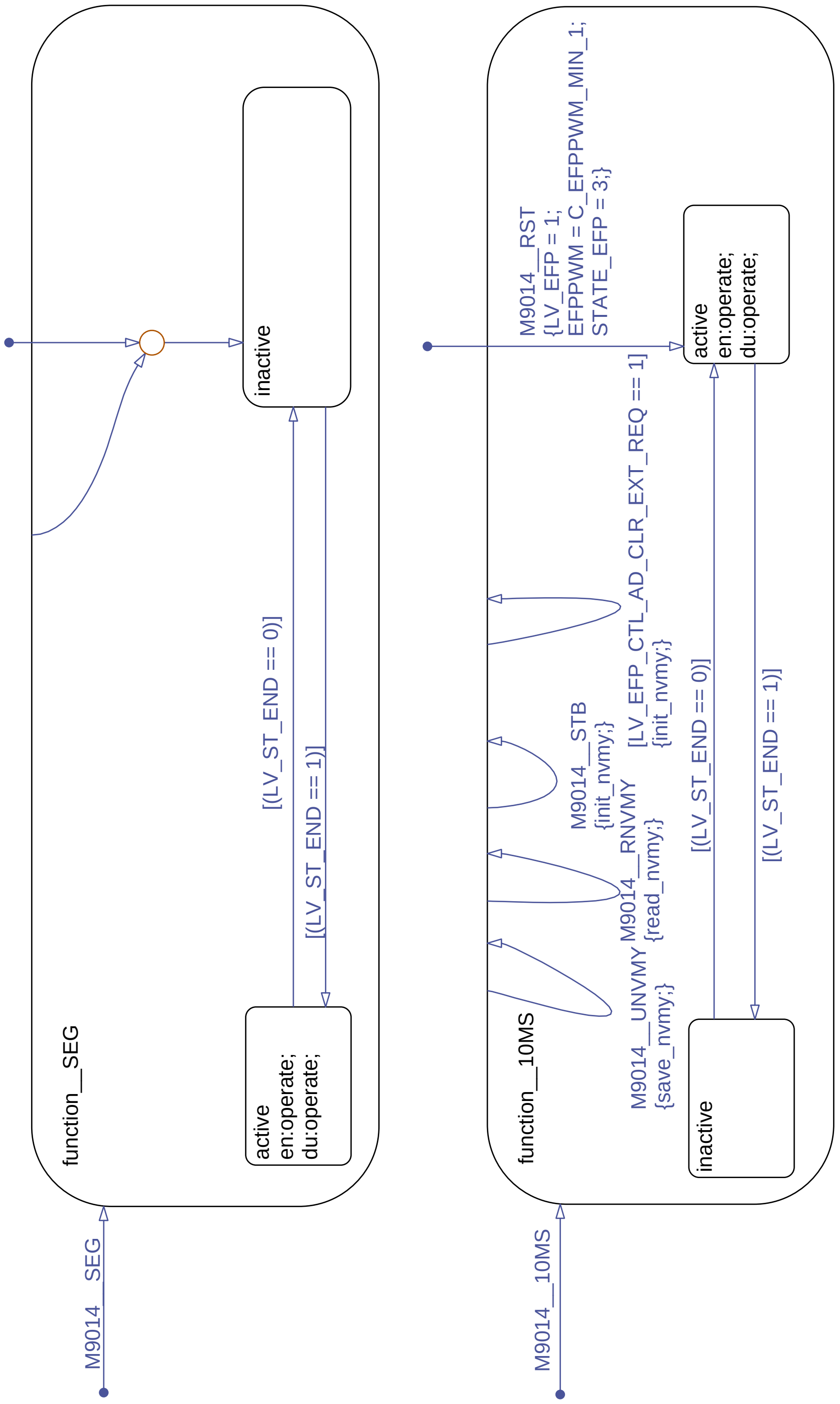
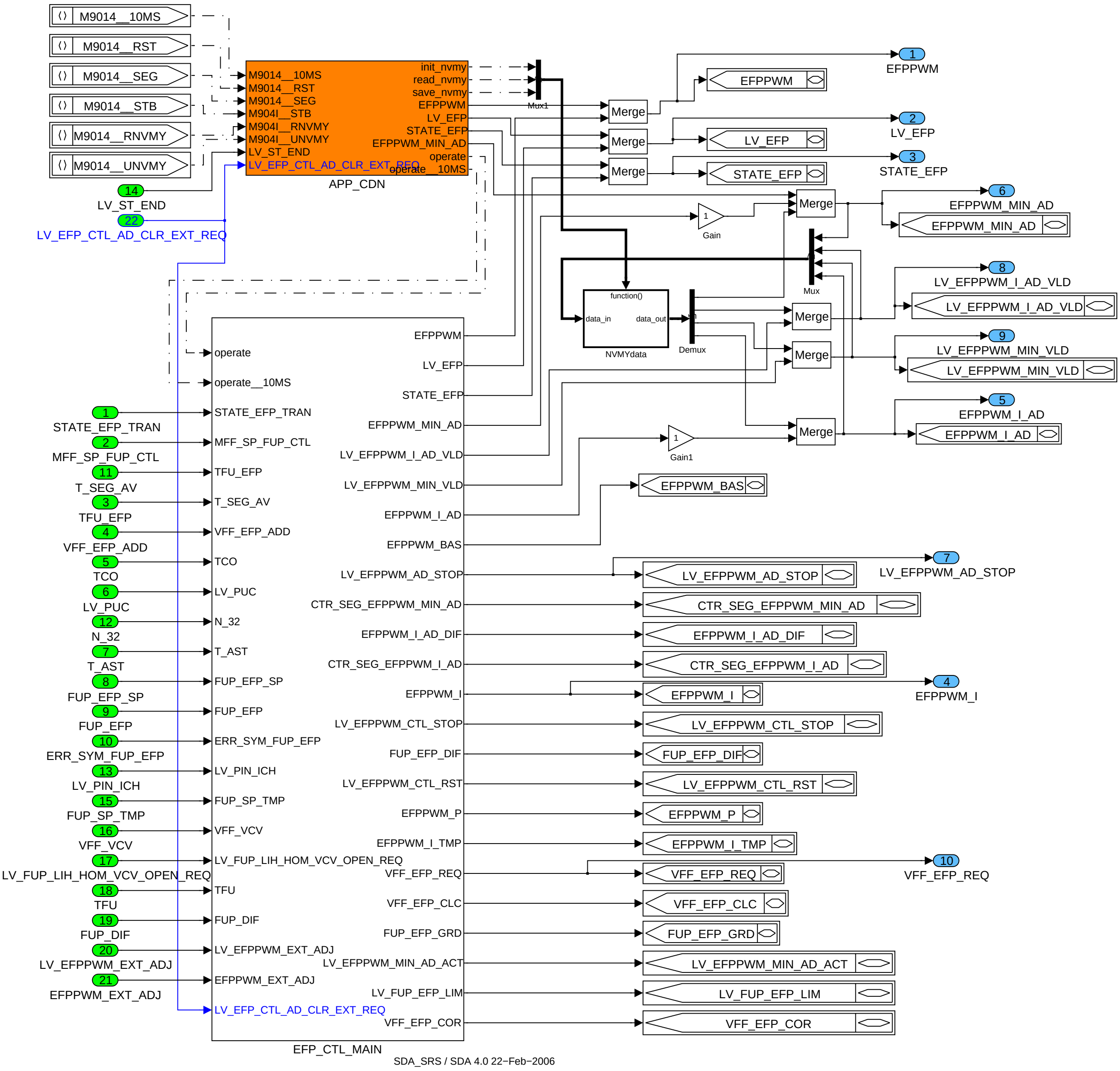


Figure 48 M9014/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 22−Feb−2006

Figure 49 M9014

34.9.1.1 Electrical fuel pump main

This picture shows an overview of the structure.

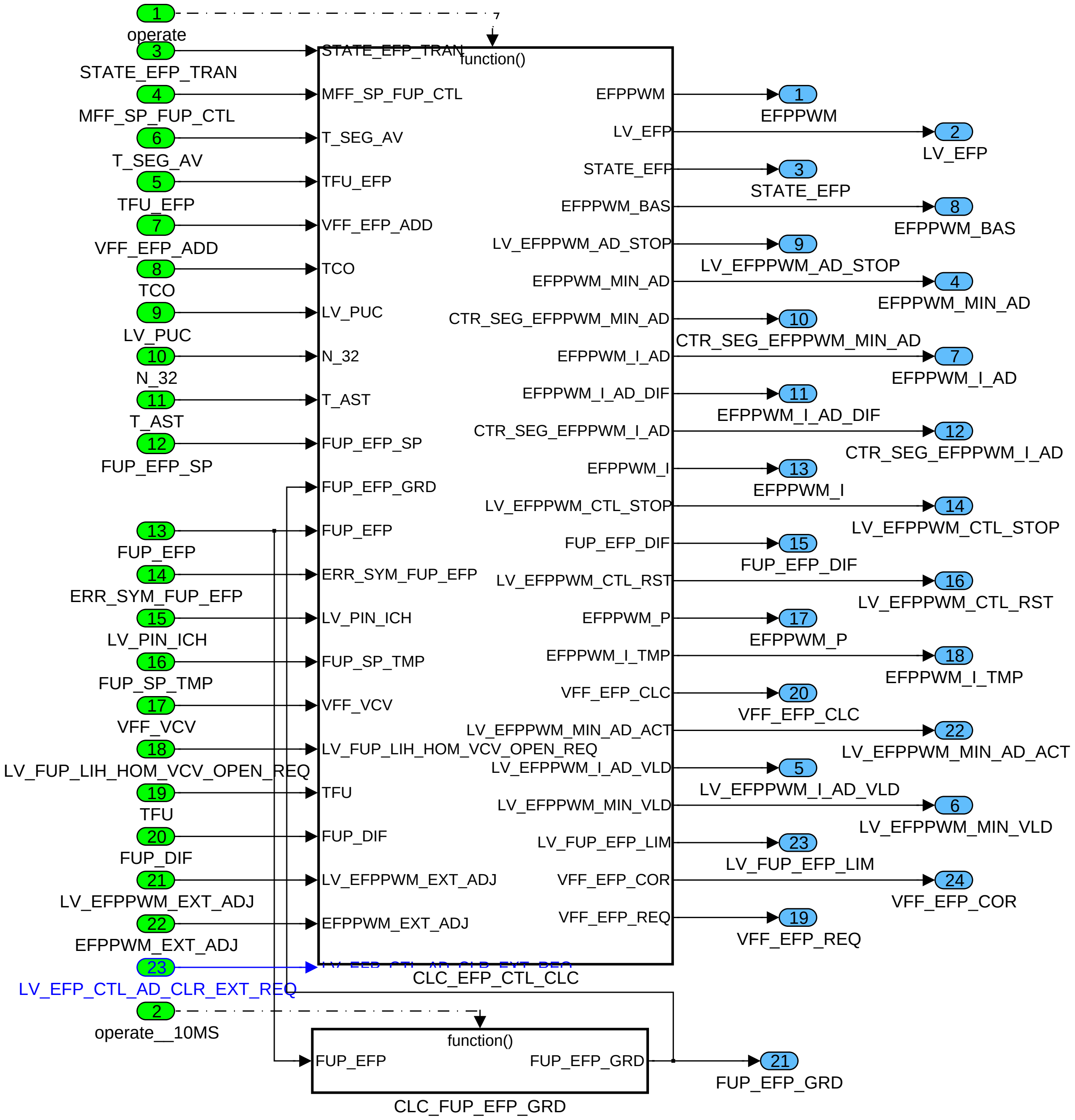


Figure 50 M9014/ EFP_CTL_MAIN

Electrical fuel pump calculation

Calculation of the main function of the electrical fuel pump controller. The function is split into four parts corresponding to the calculated state.

STATE_EFP_CLC:

The different modules have to be calculated due to state diagram below. The input STATE_EFP_TRAN will be produced in the file Electrical fuel pump (Appl. Inc.).

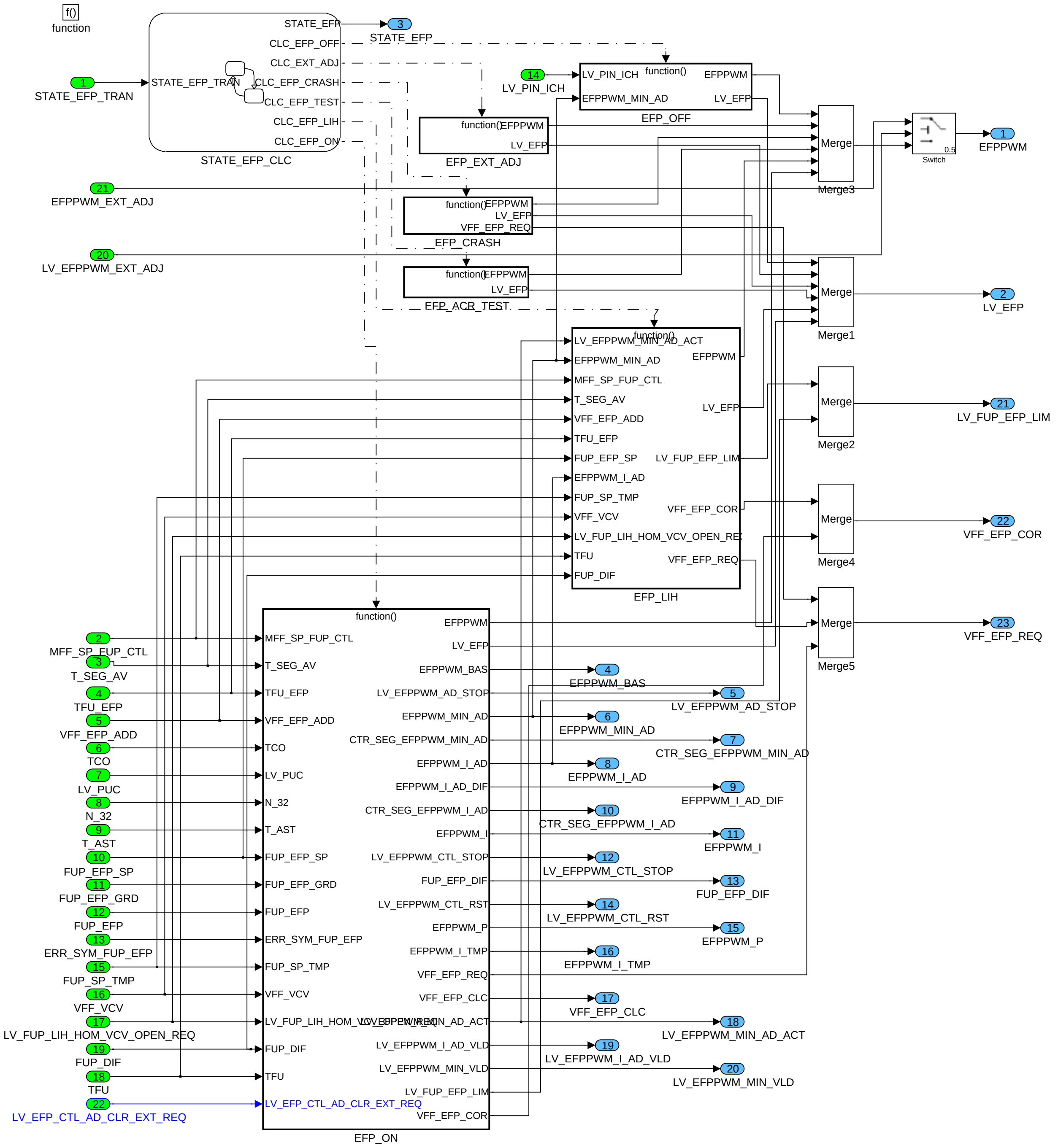


Figure 51 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC

Electrical fuel pump actuator test

Actuator test

This testmode is request by an external customer test device. It forces the EFPPWM to have a certain testpattern defined in that function.

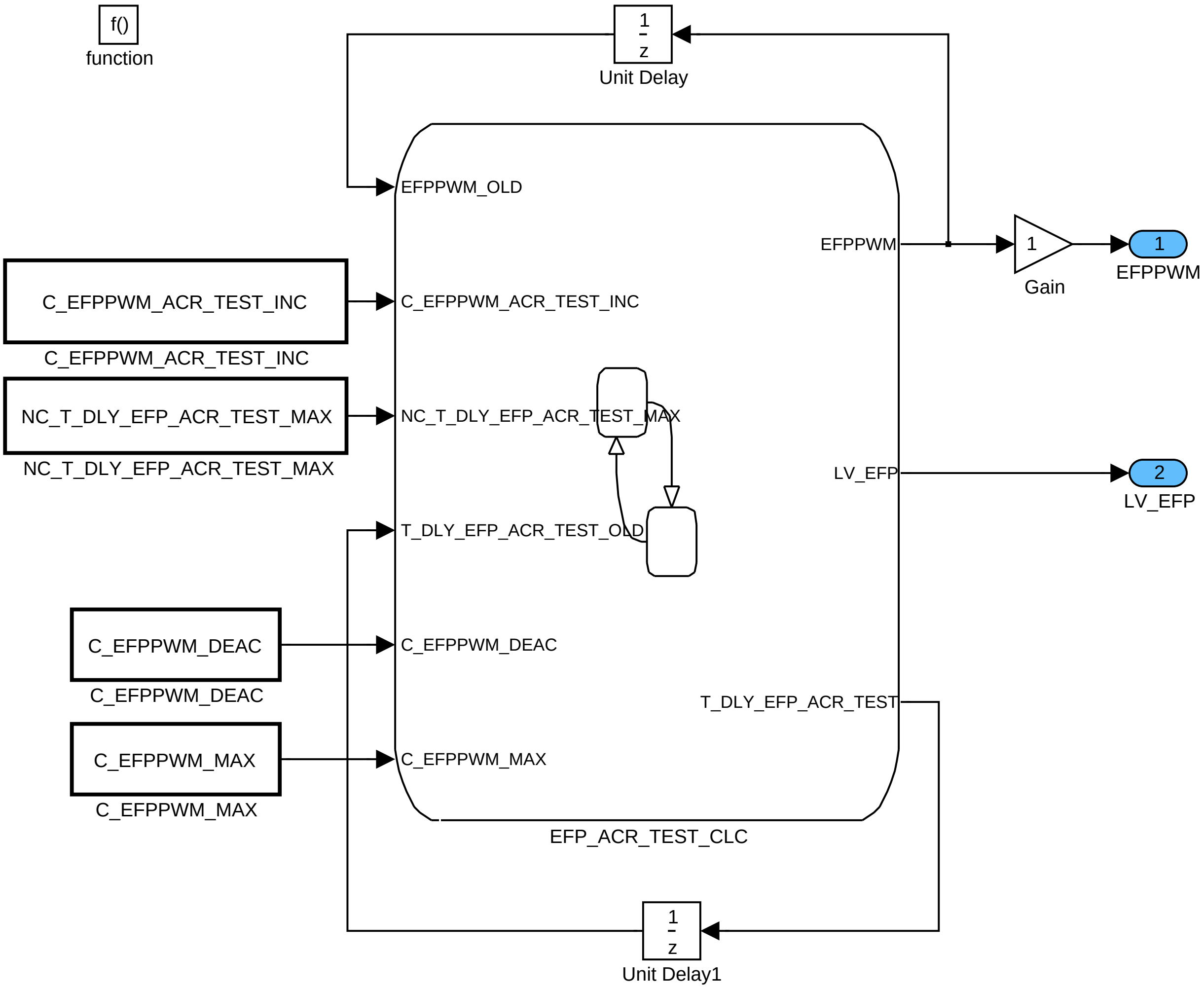


Figure 52 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ACR_TEST

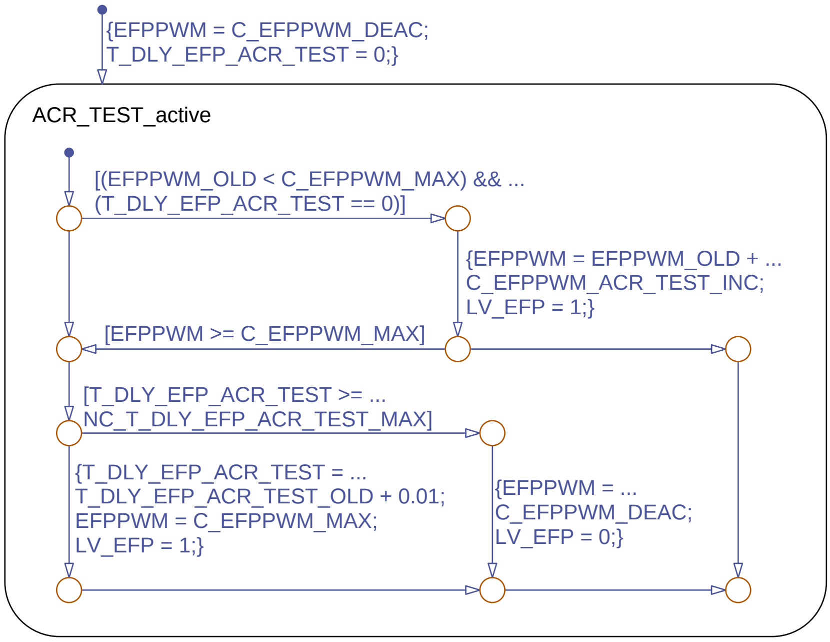


Figure 53 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ACR_TEST/ EFP_ACR_TEST_CLC

Electrical fuel pump crash

Calculation of EFPPWM in state "crash"

In case of crash a special EFPPWM should be given.

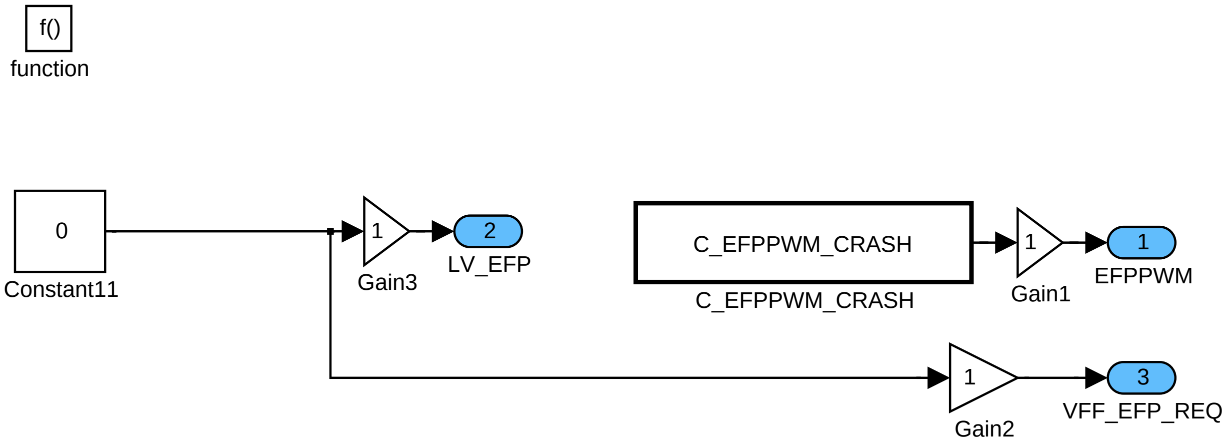


Figure 54 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_CRASH

Electrical fuel pump external adjustment

External adjustment of the EFPPWM

For testing reasons this module can generate a manual EFPPWM.

f()

function

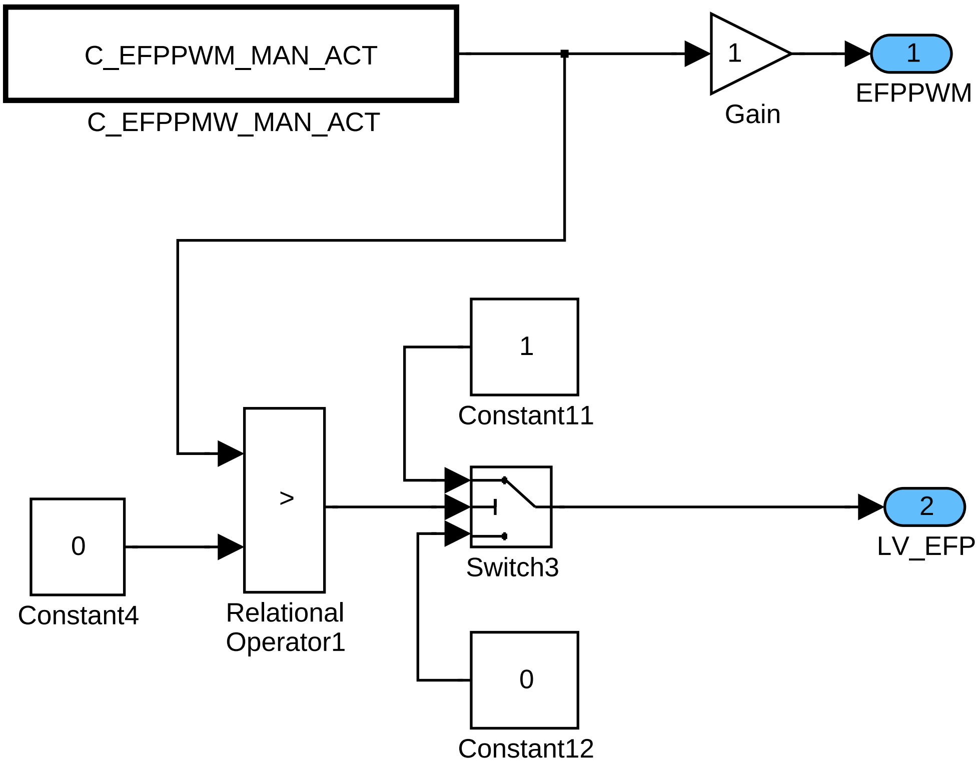


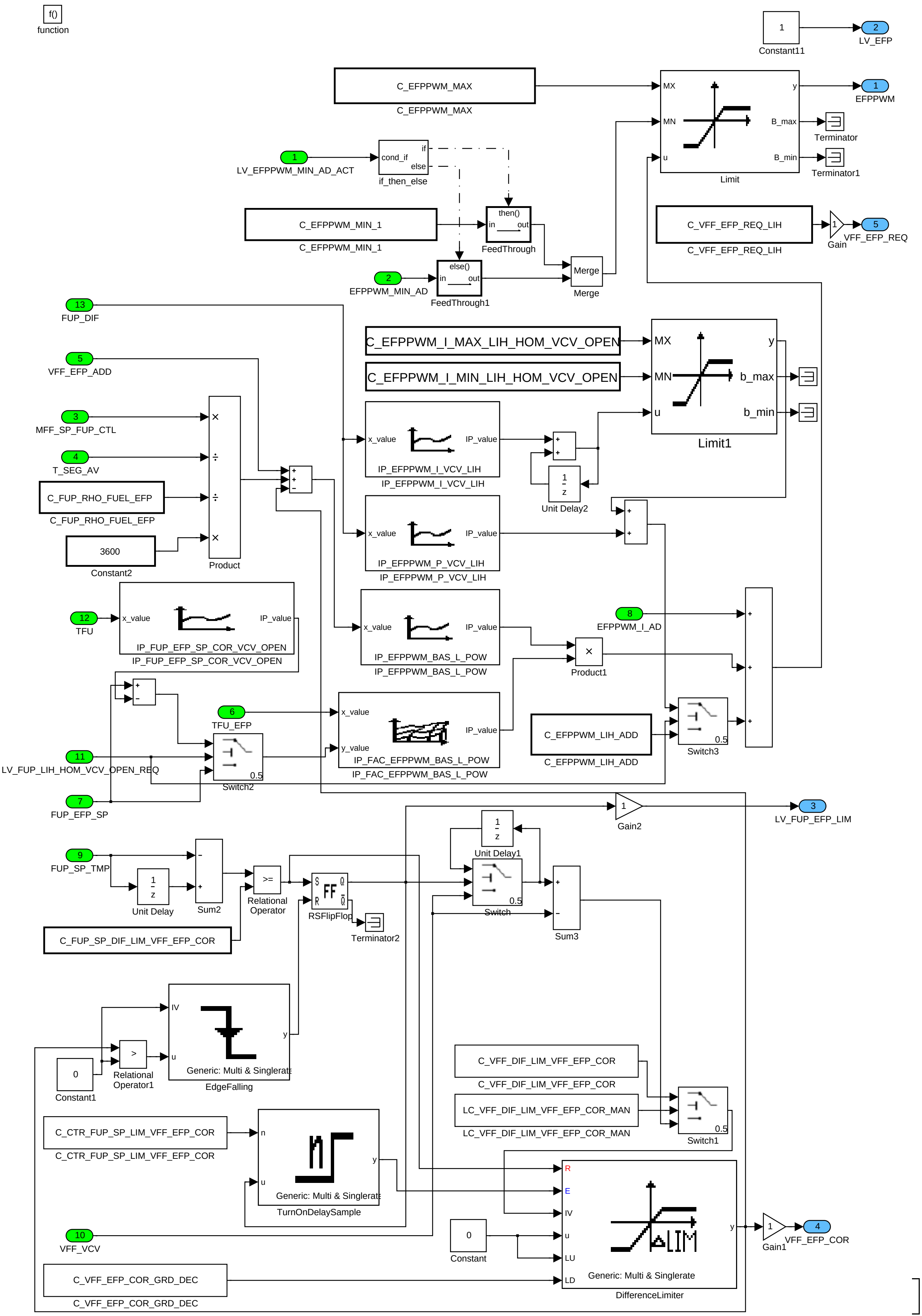
Figure 55 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_EXT_ADJ

Electrical fuel pump limp home

Calculation of EFPPWM limp home

In case of limp home the EFPPWM is at maximum.

general specification



Cha
Fu

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Regensburg(RBG)

A4 : 2004-06

Figure 56 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_LIH

Electrical fuel pump off

Calculation of the controller in state "off"

If no fuel delivery is required the ECU has to give a minimum EFPPWM. The only operating point where this rule can not be applied is at request from the independent car heater. At this point the adapted minimum PWM is applied to the pump.

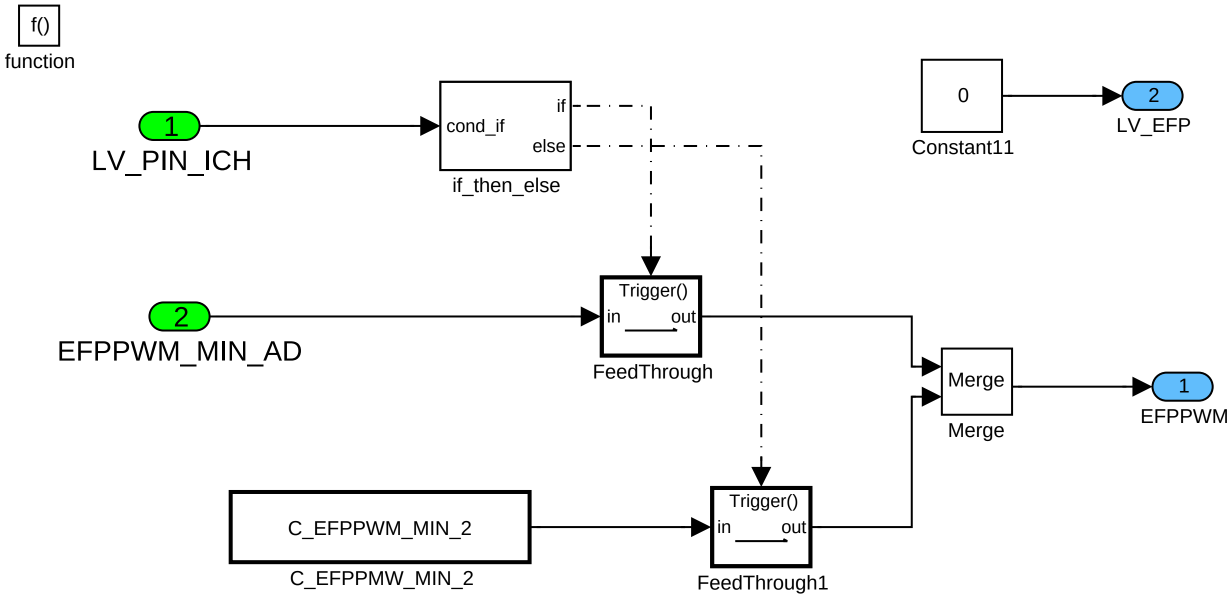


Figure 57 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_OFF

Electrical fuel pump on

Calculation of the Controller in state "on"

The functionality for the electrical fuel pump in status "on" is given here. It is split into a precontrol calculation part, a limitation signal generating part (mainly for PUC), an EFPPWM calculation part and the controller output part.

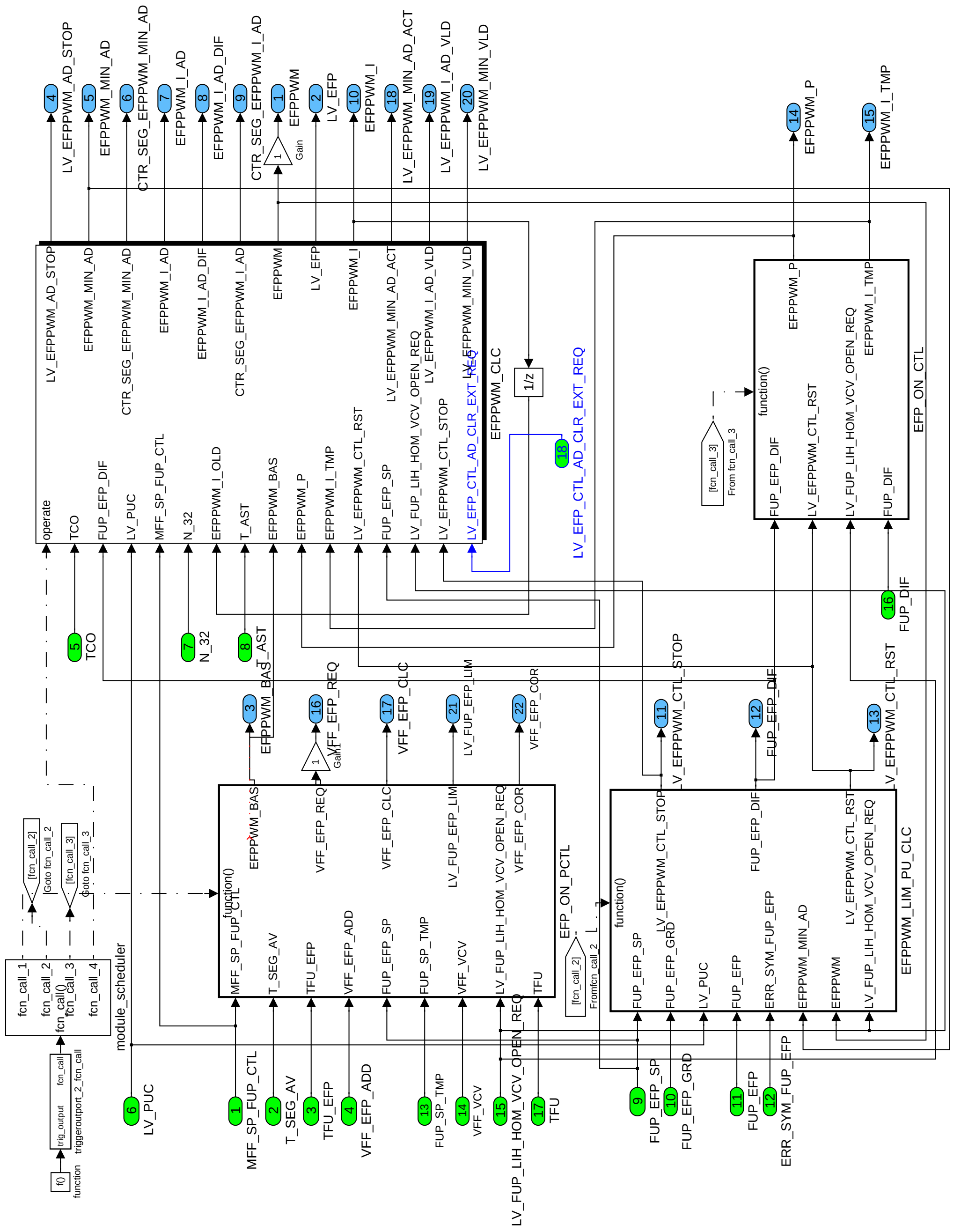


Figure 58 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON

Electrical fuel pump pwm signal calculation

Calculation of the final output signal

The EFPPWM calculation part is split into two parts. One is calculating the adaptation and the other part calculate the final EFPPWM output.

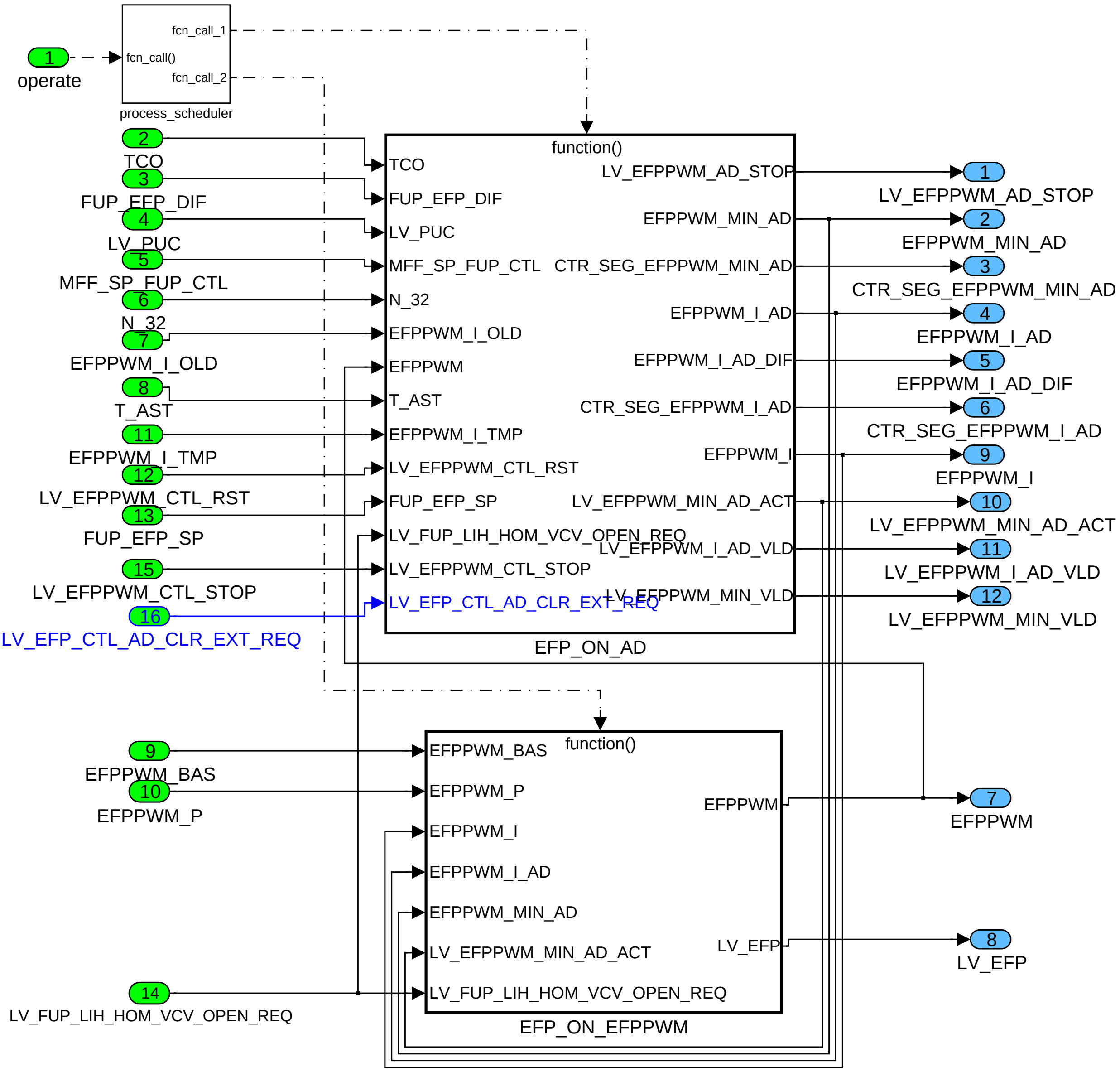


Figure 59 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC

<u>Electrical fuel pump on adaptive</u>

Calculation of the adaptation

The adaptation part is split into the main adaptation inbit flag calculating module, the calculation part for the minimum EFPPWM adaptation part and two modules for the I–part adaptation.

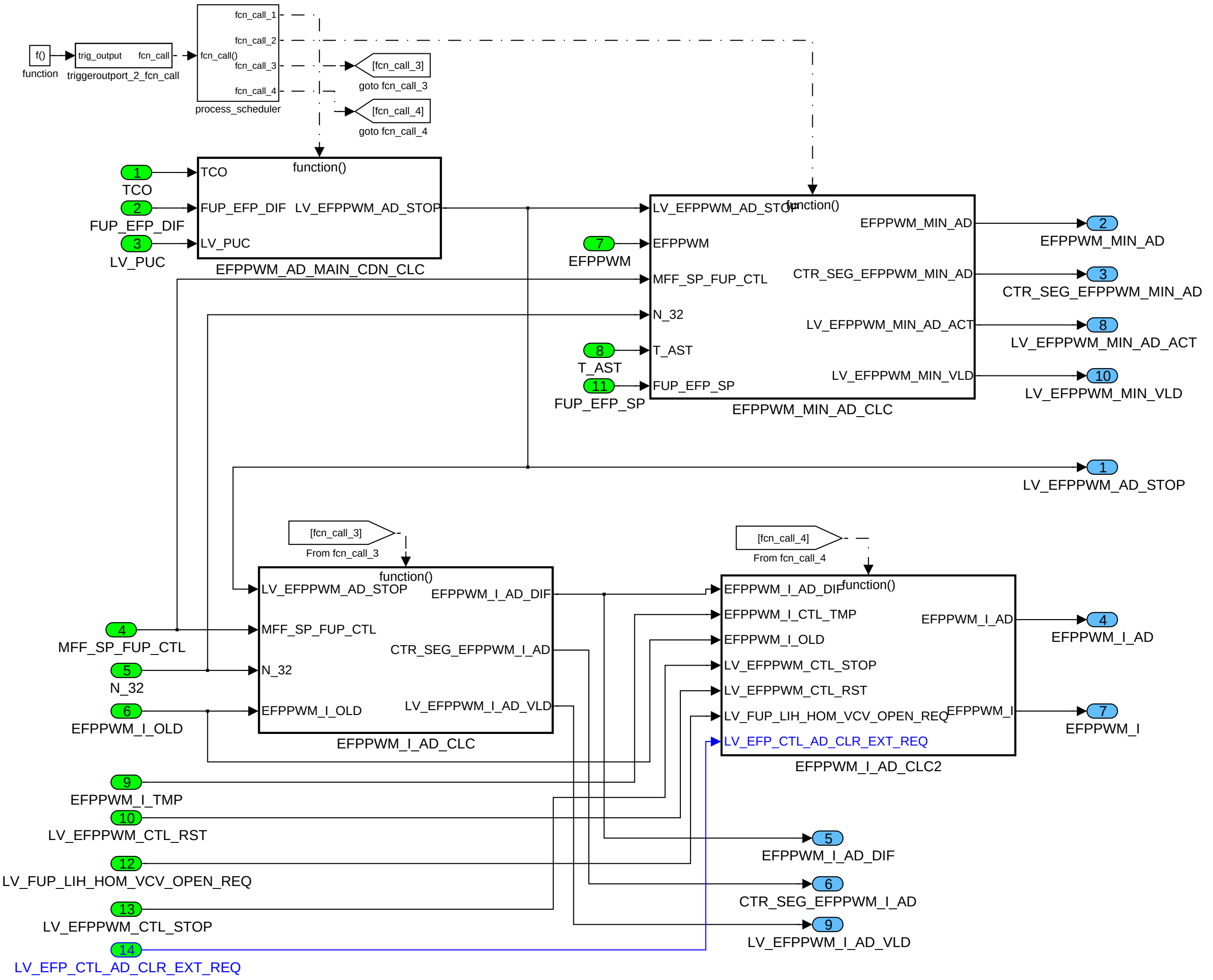


Figure 60 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC/ EFP_ON_AD

Electrical fuel pump pwm signal adaptive main condition calculation

Main conditions for the adaptation

Within this module the inhibition flag for the adaptation is created.

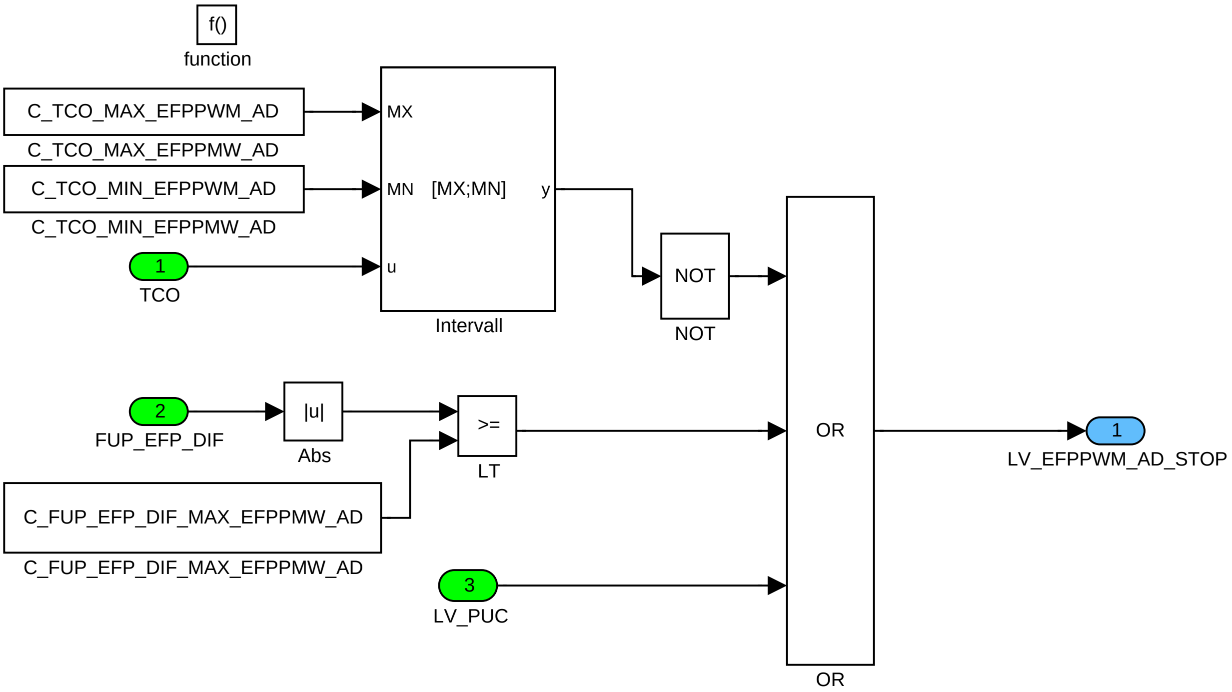


Figure 61 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC/ EFP_ON_AD/ EFPPWM_AD_MAIN_CDN_CLC

Electrical fuel pump pwm signal integral part of a control structure adaptive calculation

Calculation of the adaptation via I – part of the controller is calculated within that module. The main function is descripted in the state diagram.

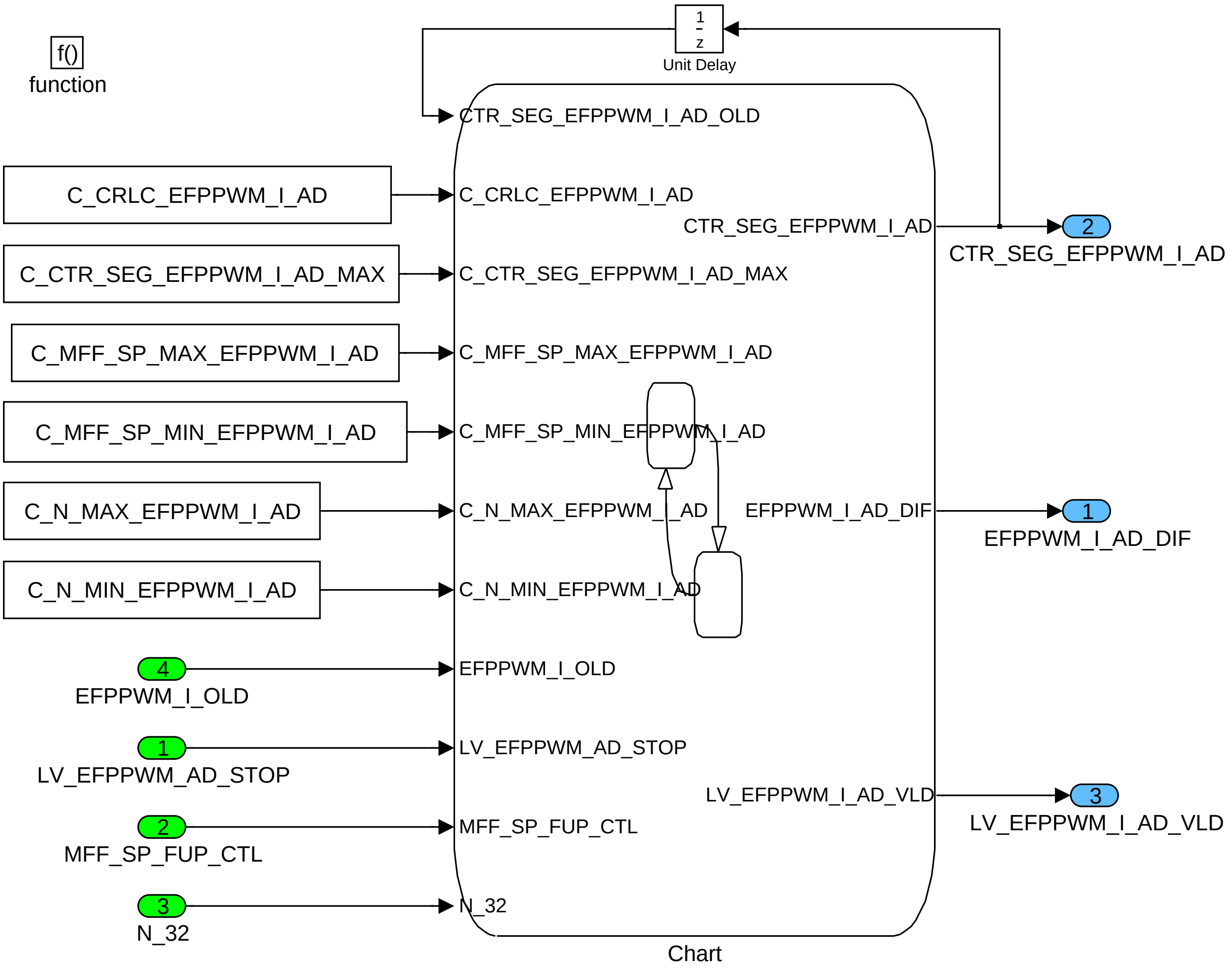


Figure 62 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC/ EFP_ON_AD/ EFPPWM_I_AD_CLC

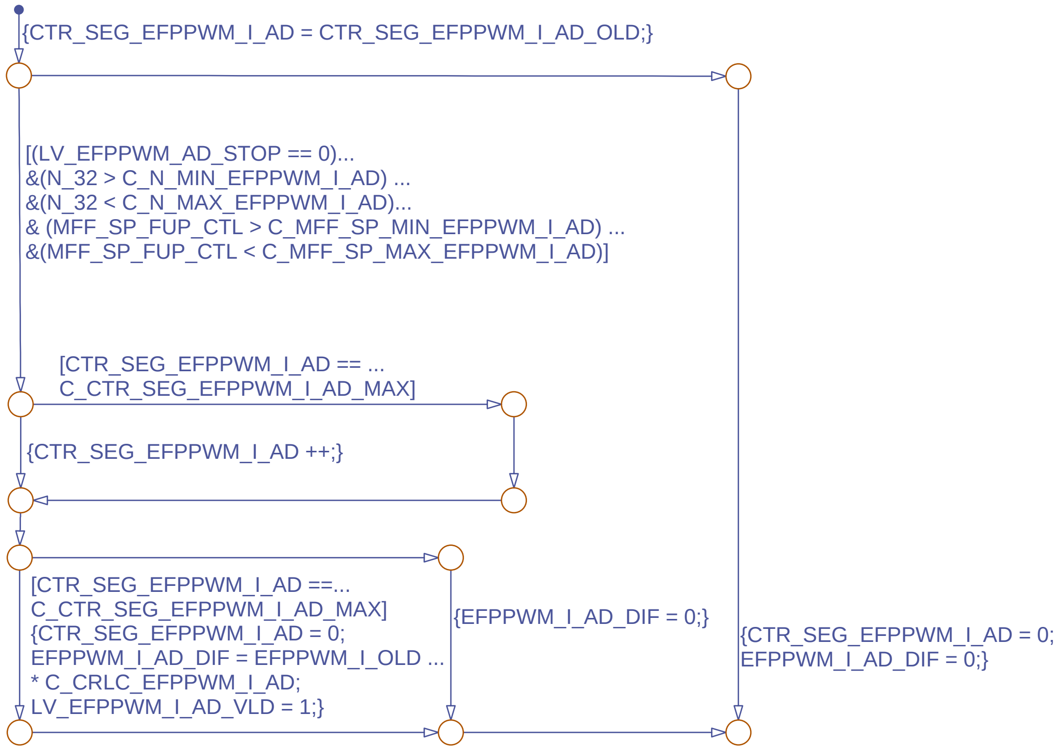


Figure 63 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC/ EFP_ON_AD/ EFPPWM_I_AD_CLC/ Chart

<u>Electrical fuel pump pwm signal integral part of a control structure adaptive calculation 2</u>

Final calculation of the I – part adaptation and the I – part of the controller

Out of the I – part of the last calculation cycle and the difference of the adaptive I – part of the controller and some other signals the final I – part and the adaptive I – part of the controller is calculated.

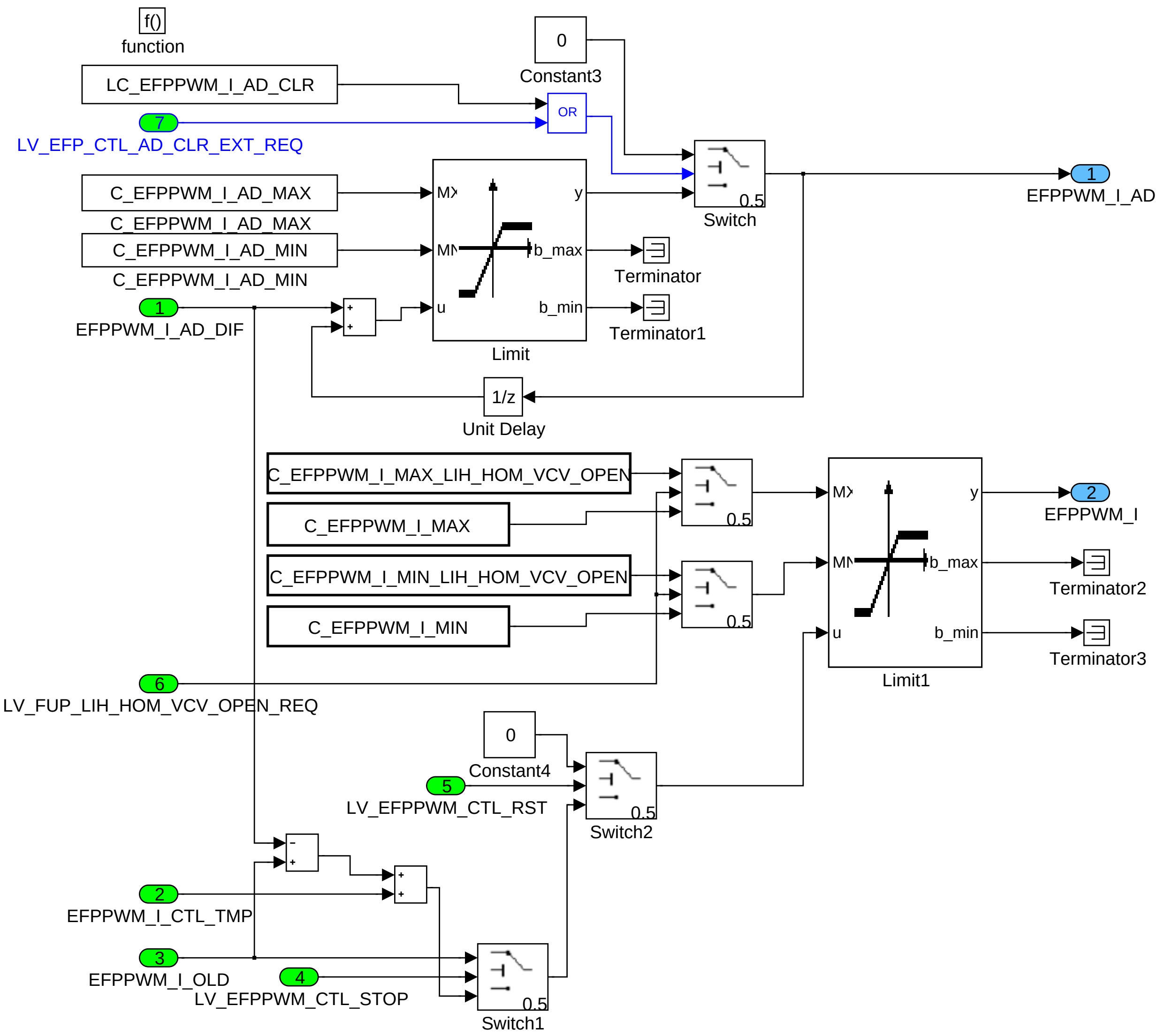


Figure 64 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC/ EFP_ON_AD/ EFPPWM_I_AD_CLC2

Electrical fuel pump pwm signal minimum adaptive calculation

Calculation of the minimum EFPPWM

The minimum EFPPWM for the pump enabling the correct function of the jet pump is adapted within that function. The base is descripted in the state diagram.

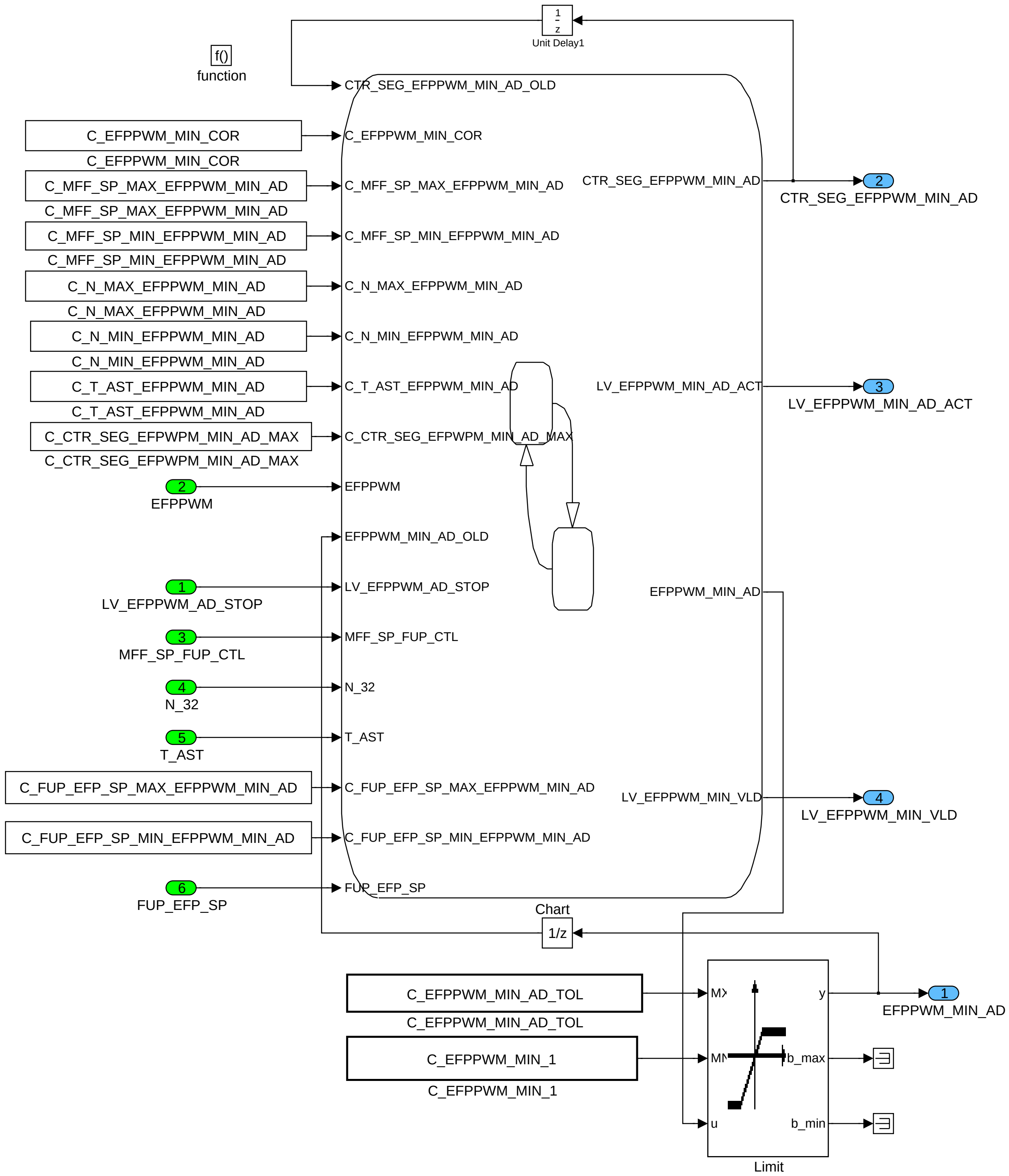


Figure 65 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC/ EFP_ON_AD/ EFPPWM_MIN_AD_CLC

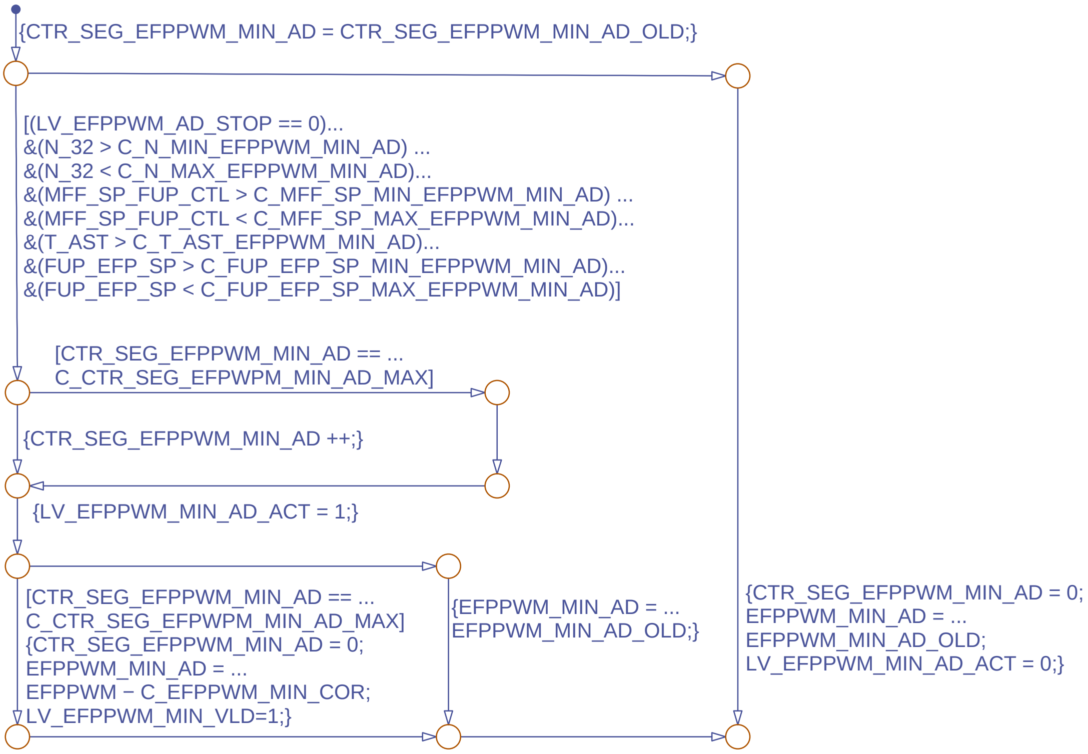


Figure 66 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC/ EFP_ON_AD/ EFPPWM_MIN_AD_CLC/ Chart

<u>Electrical fuel pump on electrical fuel pump pwm signal</u>

Calculation of the final output for the EFPPWM

In this module the final EFPPWM is calculated.

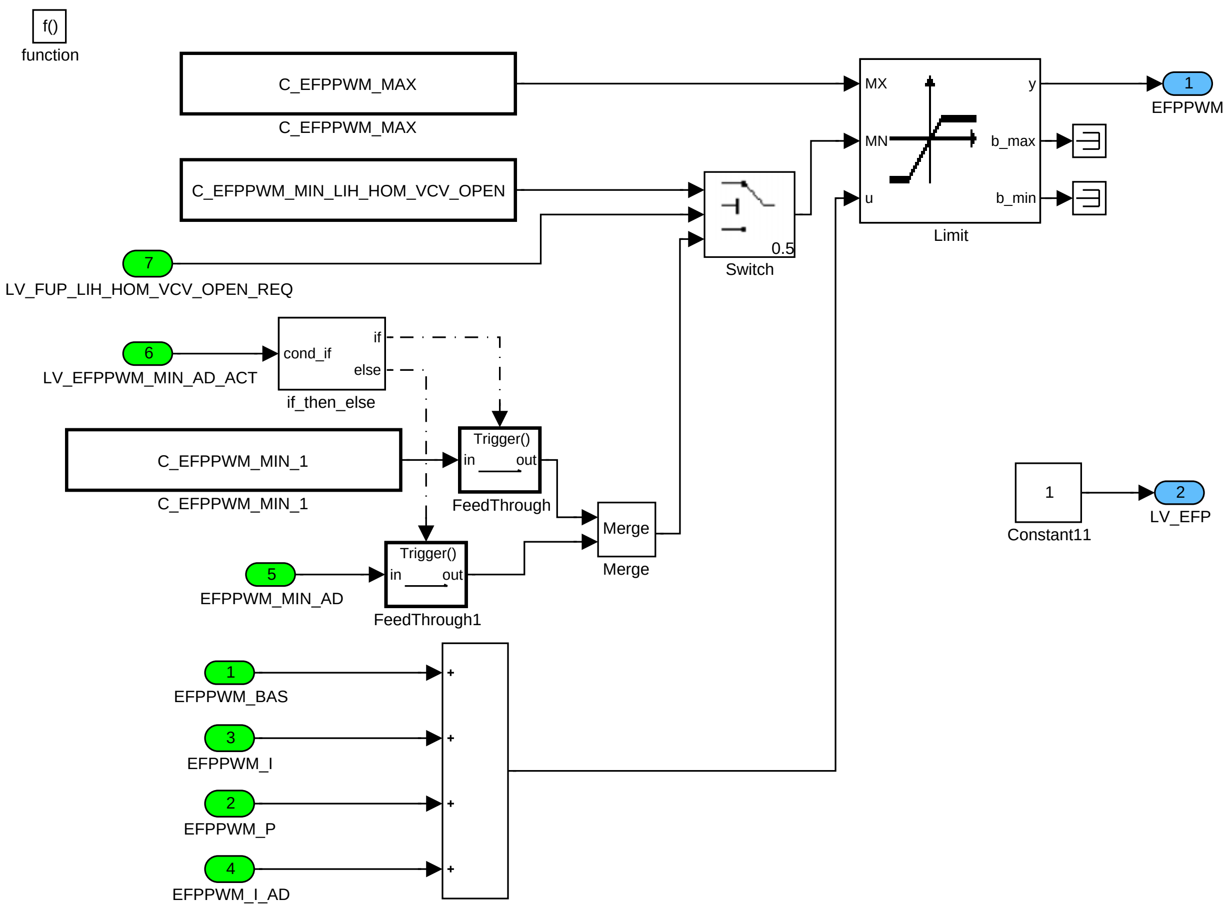


Figure 67 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_CLC/ EFP_ON_EFPPWM

<u>Electrical fuel pump pwm signal limitation pull calculation</u>

Limitation of the controller

Given by special conditions the controller is stopped. This is done within PUC when a too high fuel pressure gradient is measured and the fuel pressure is over the setpoint.

Also the total reset of the controller output can be necessary when there is a sensor error or no sensor is available (given by the flag).

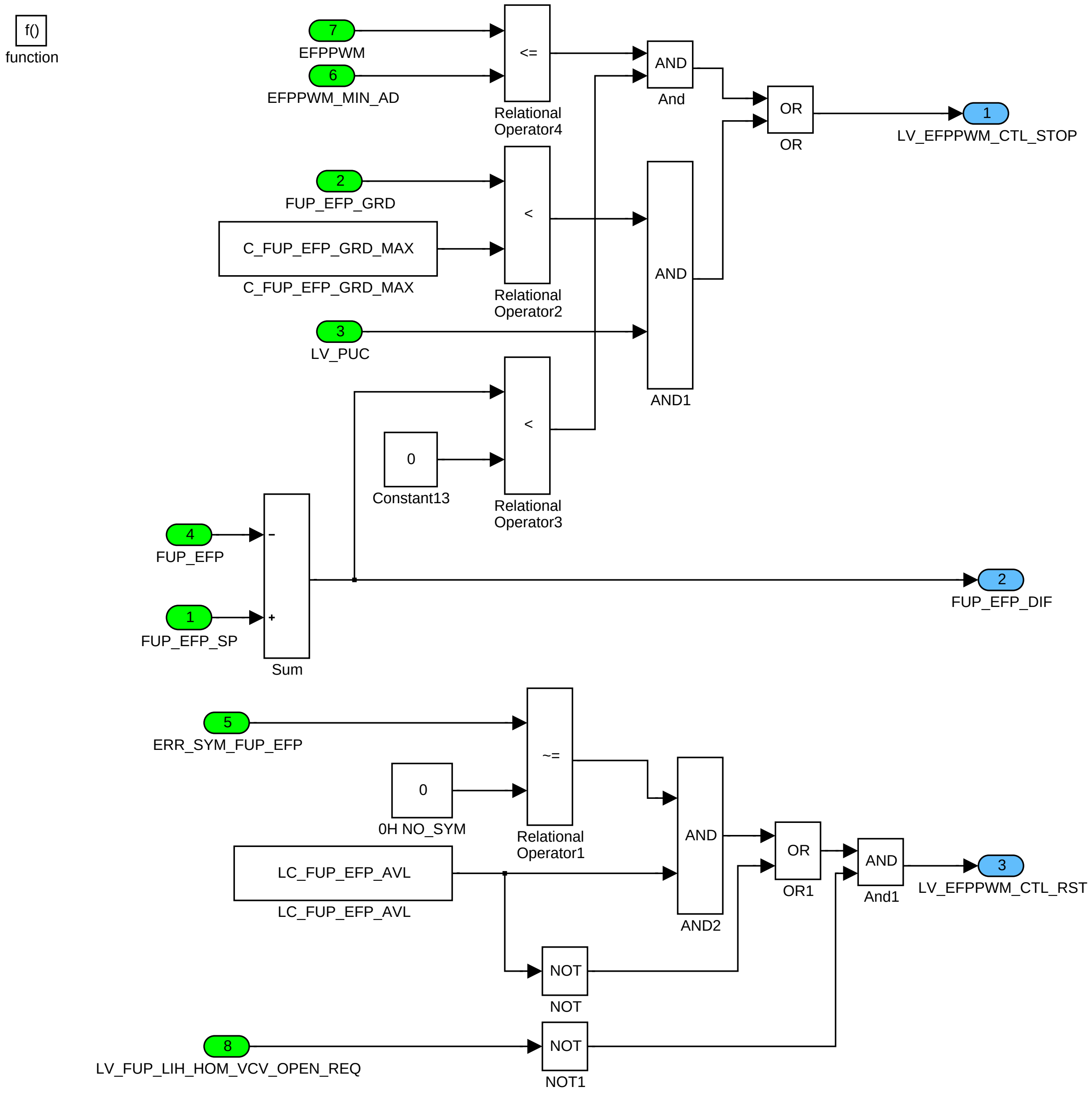


Figure 68 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFPPWM_LIM_PU_CLC

Electrical fuel pump on control

Calculation of the controller

The controller part gives the output of the controller maps for the I – part. In case of given reset flag the I – part and the P – part are zero.

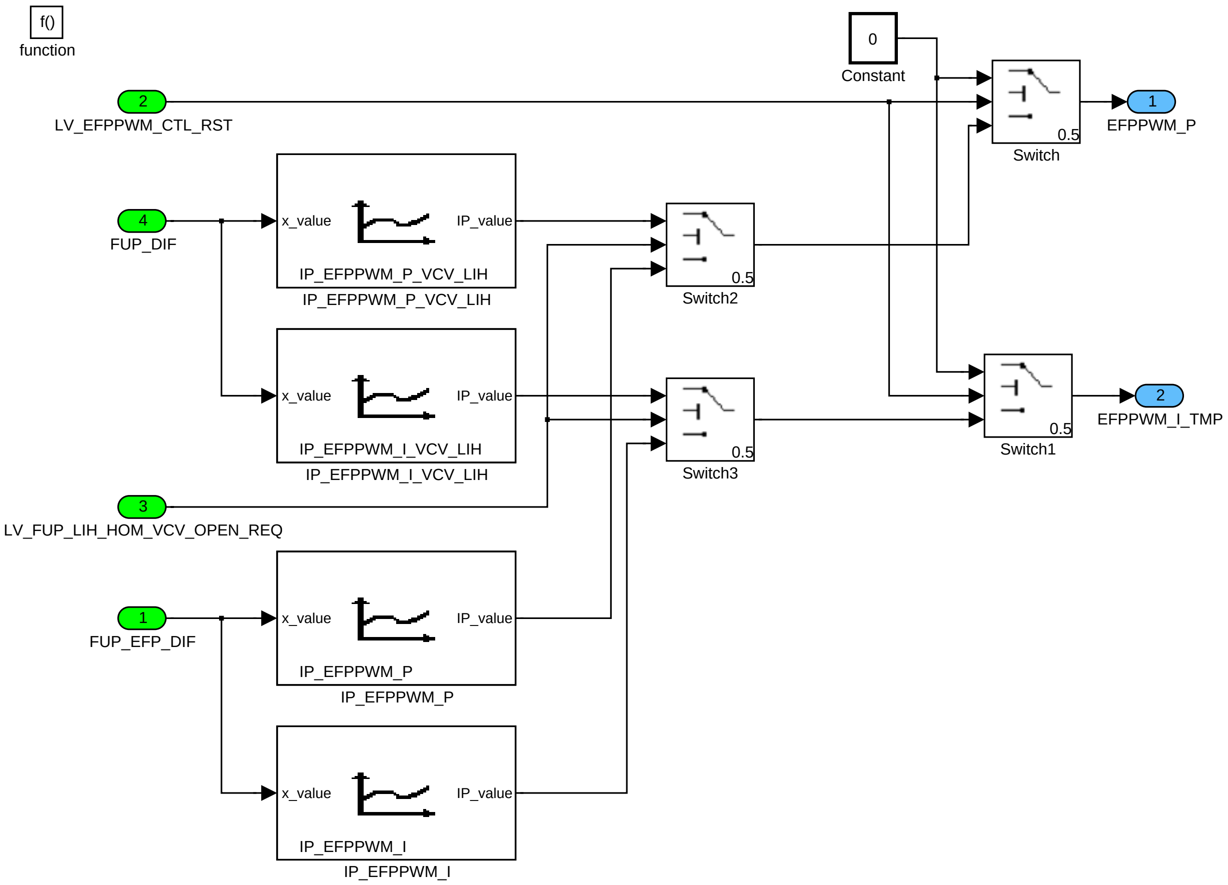


Figure 69 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFP_ON_CTL

Electrical fuel pump on pre-control

Calculation of the precontroller

The precontrol part calculated the precontroller value out of the required fuel mass for the injection. Within the given two maps the basic controller output is calculated.

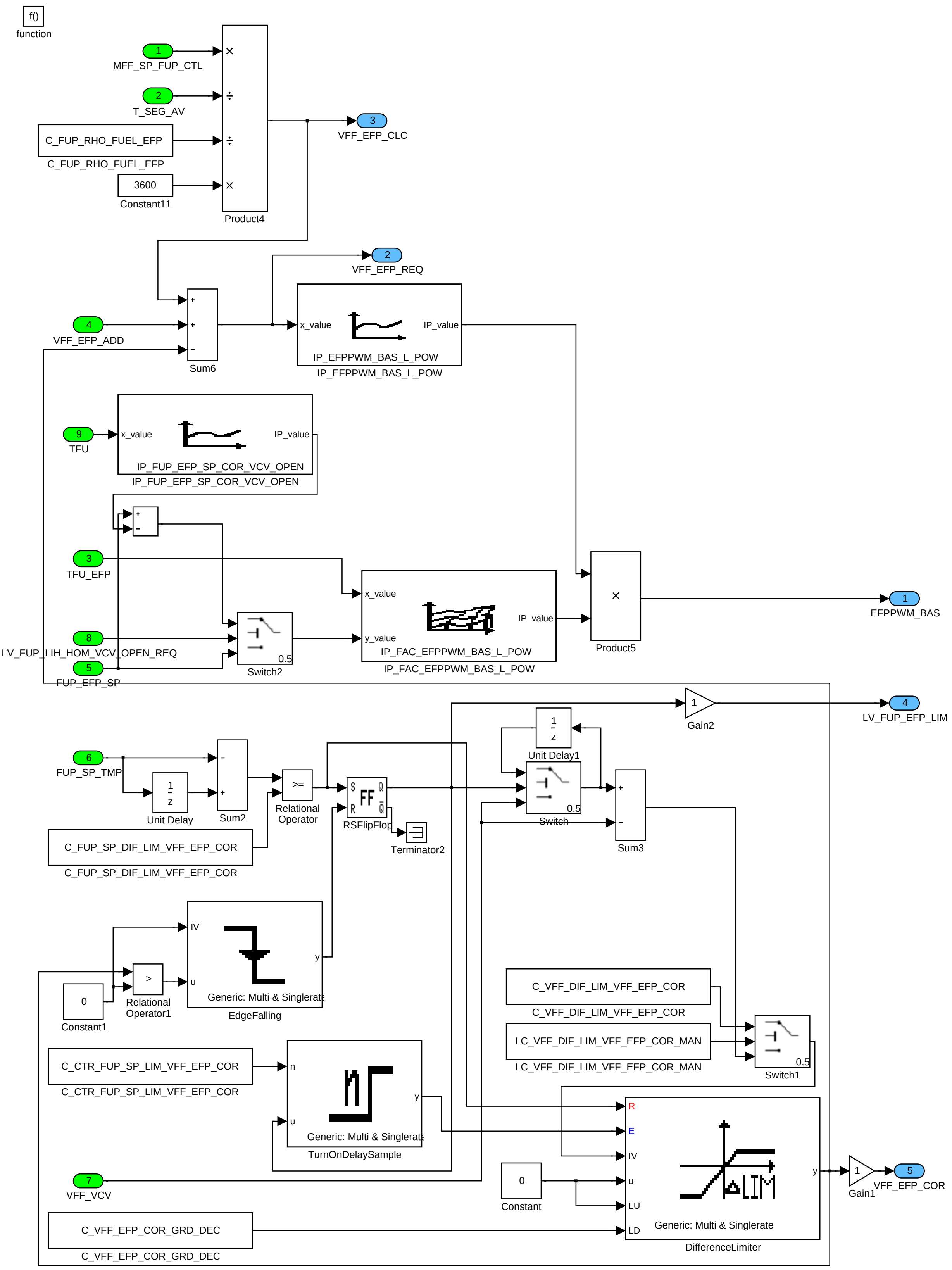


Figure 70 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ EFP_ON/ EFP_ON_PCTL

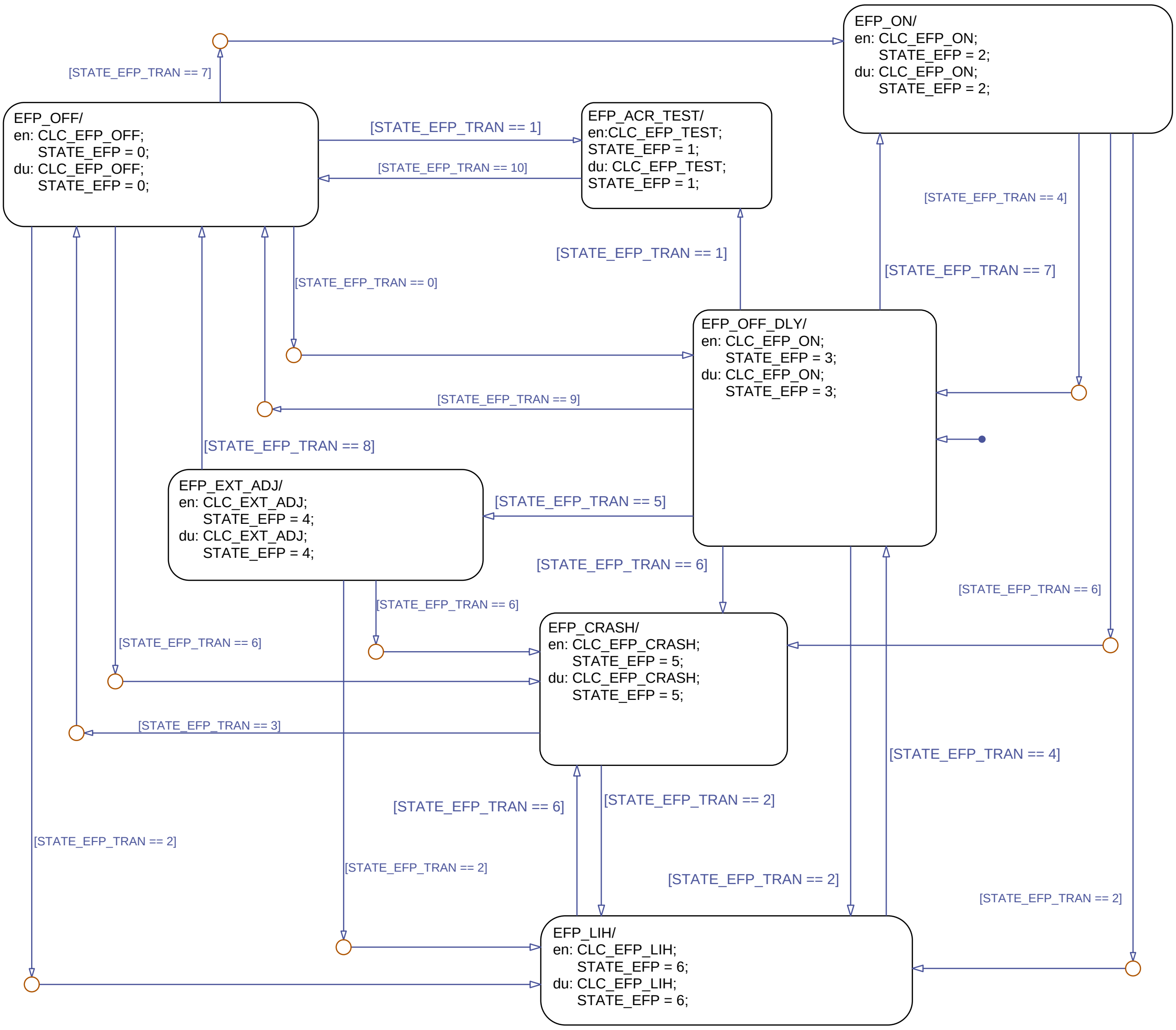


Figure 71 M9014/ EFP_CTL_MAIN/ CLC_EFP_CTL_CLC/ STATE_EFP_CLC

Fuel pressure electrical fuel pump ground

Subfunction description:

Calculation of the gradient of the fuel pressure in the low pressure circuit.

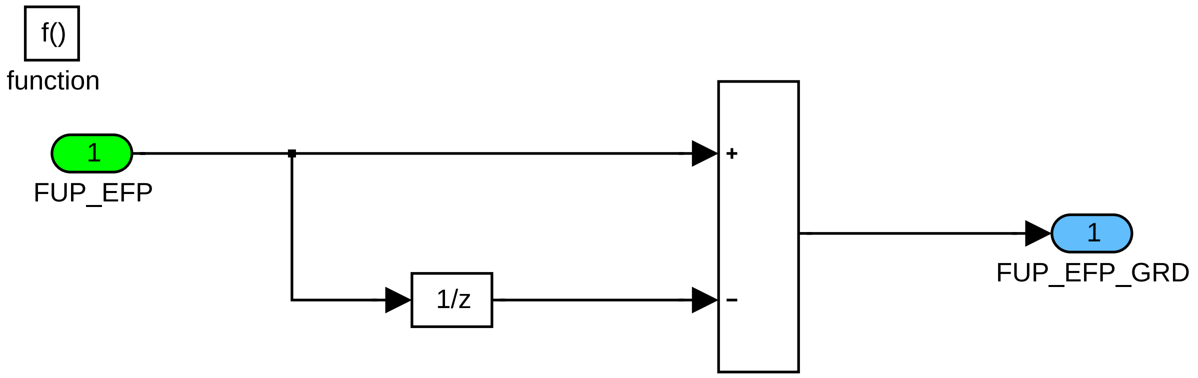


Figure 72 M9014/ EFP_CTL_MAIN/ CLC_FUP_EFP_GRD

34.9.1.2 Initialization

In this part outputs are initialised.

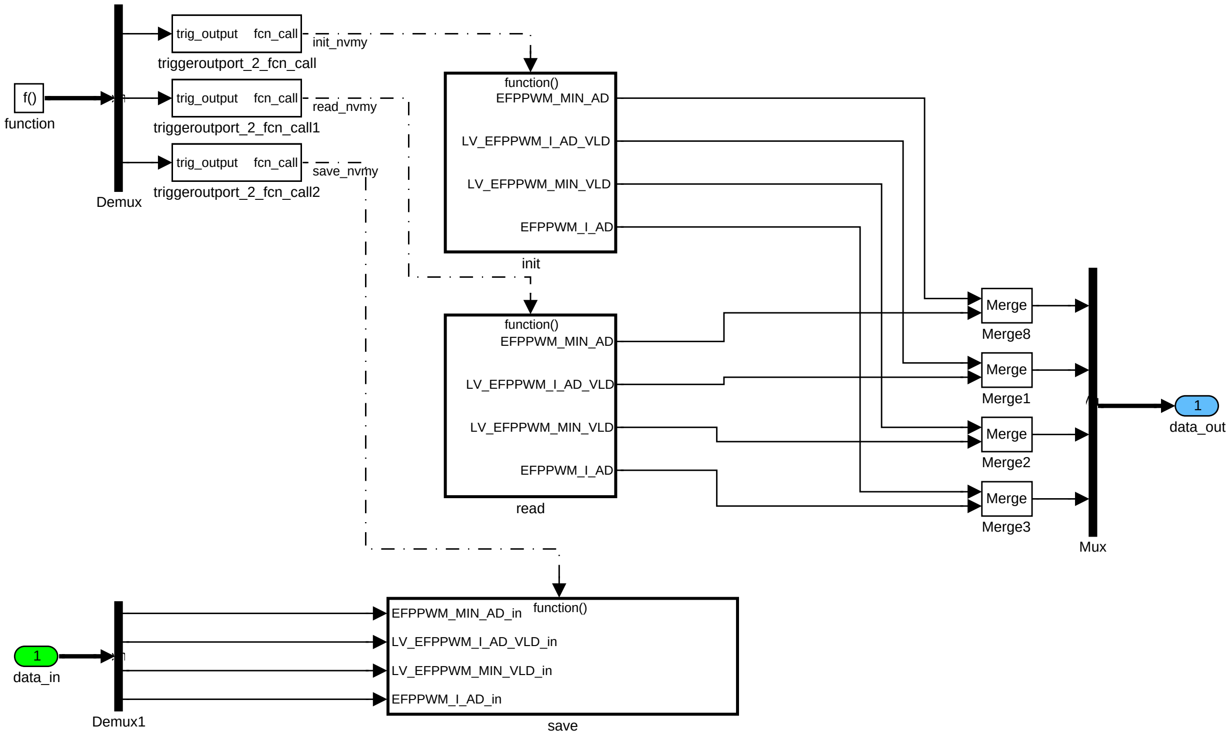


Figure 73 M9014/ NVMYdata

Initialization with constants

If EEPROM error or ECU brand new or cleared by service tool then the outputs are initialised with constants.

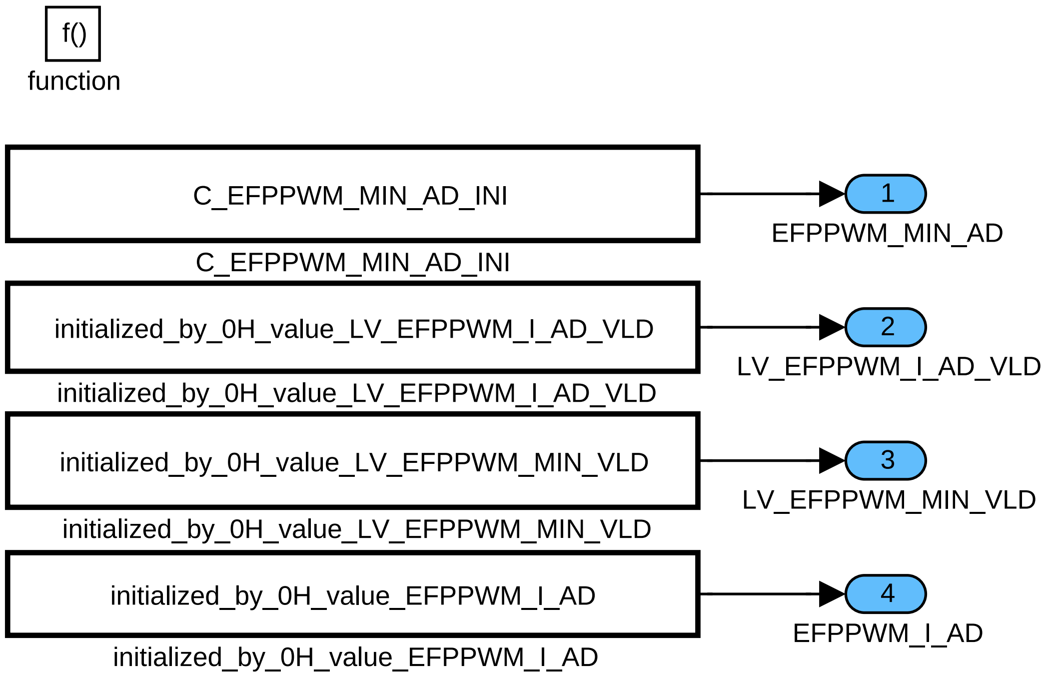


Figure 74 M9014/ NVMYdata/ init

Output read

In this part outputs are initialised by EEPROM value.

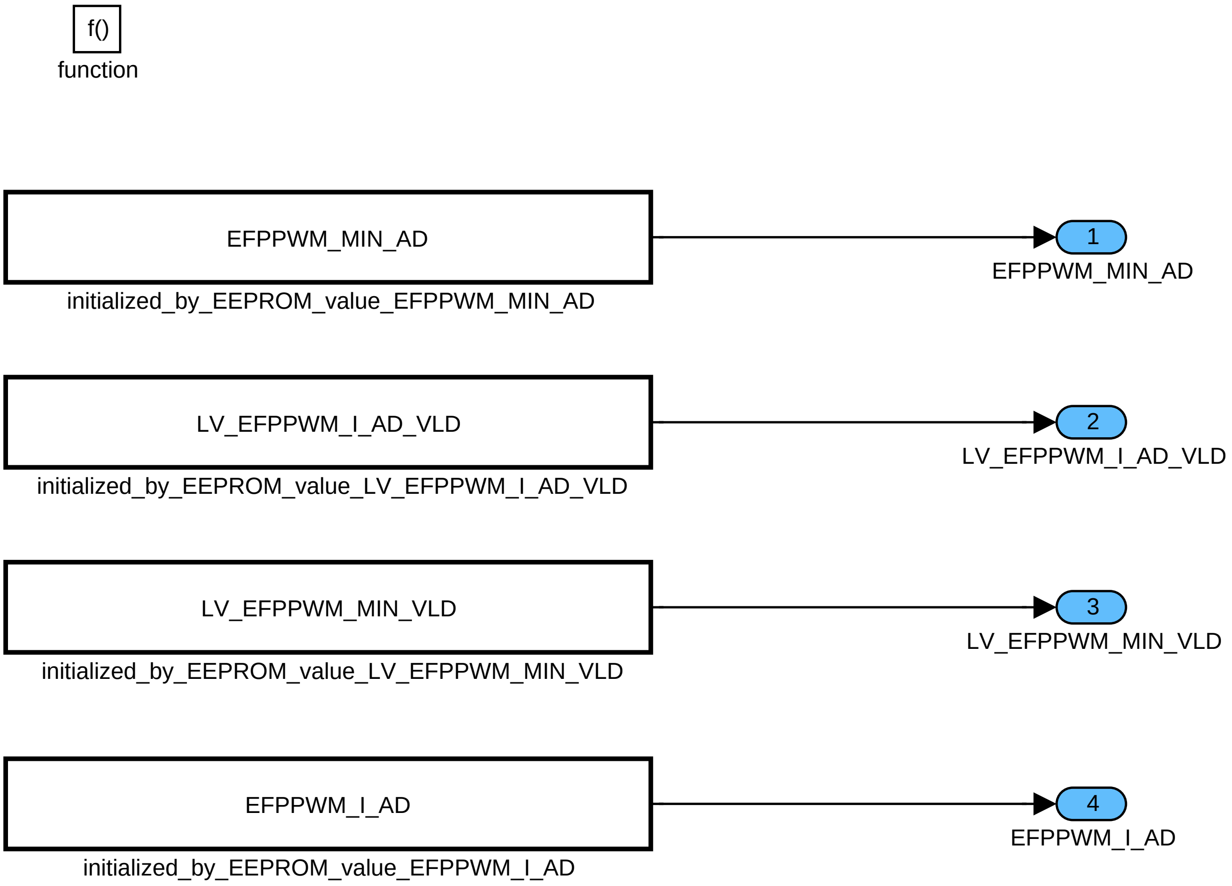


Figure 75 M9014/ NVMYdata/ read

Output save

In this part outputs are saved on the EEPROM.

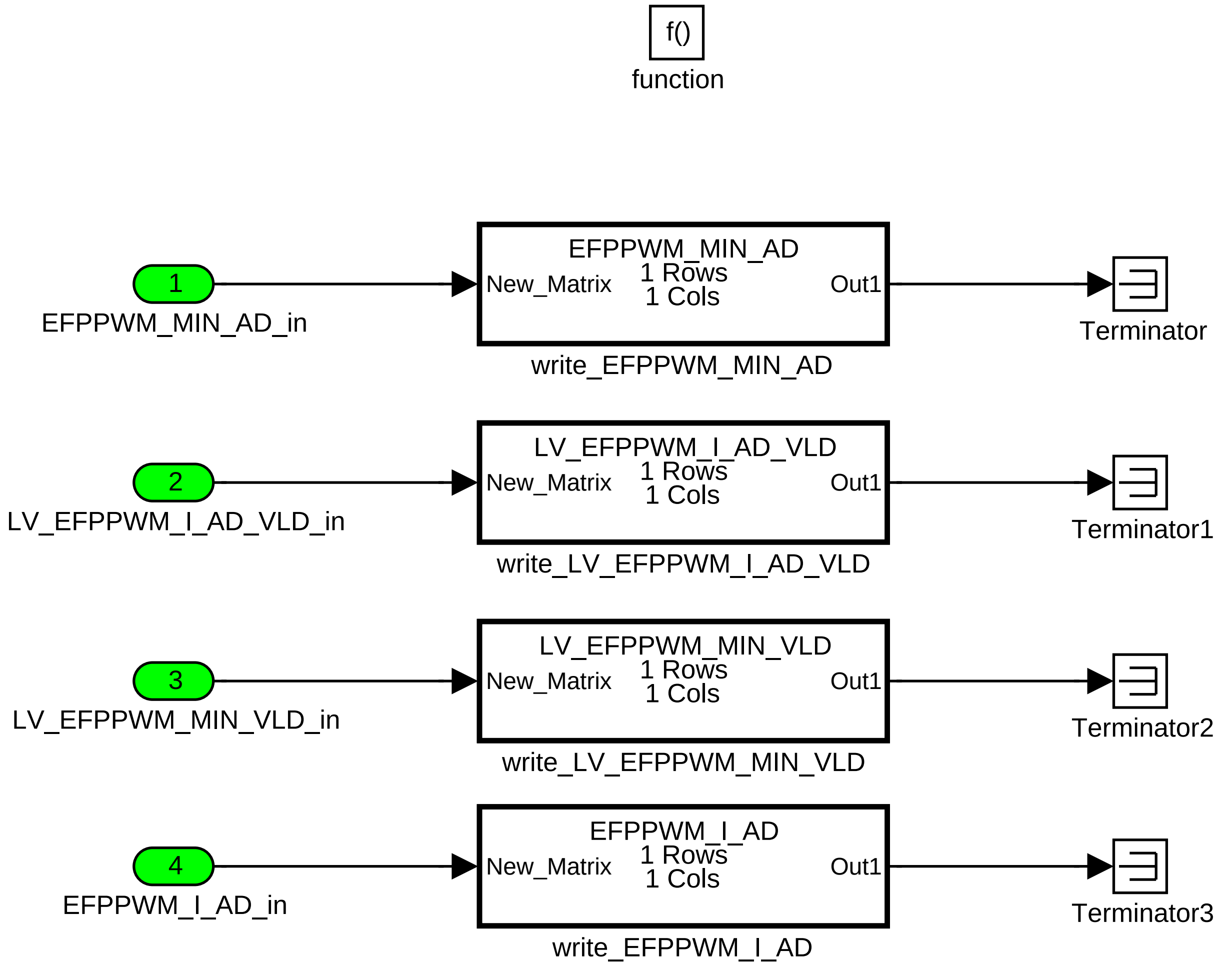


Figure 76 M9014/ NVMYdata/ save

34.10 Electrical fuel pump control (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_EFP_TRAN	O/V	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH	OFF_DLY ACR_TEST LIH CRASH2OFF ON2OFF_DLY ON2EXT_ADJ CRASH OFF_DLY2ON EXT_ADJ2OFF OFF_DLY2OFF ACR_TEST2OFF	1	-
States of electrical fuel pump function					
VFF_EFP_ADD	O/V	0...FFH	0...255	1	l/h
Additive value for the amount of fuel from the pump					
T_EFP_MEM_H	O/V/S	0...FFFFH	0...655.35	0.01	s
Time for the pressure increase					
T_EFP_MEM_L	O/V/S	0...FFFFH	0...655.35	0.01	s
Time for the pressure decrease					
FUP_EFP_STOP_H	O/V/S	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Pressure at observation stop for the pressure increase					
FUP_EFP_STOP_L	O/V/S	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Pressure at observation stop for the pressure decrease					
FUP_EFP_ST_H	O/V/S	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Pressure at observation start for the pressure increase					
FUP_EFP_ST_L	O/V/S	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Pressure at observation start for the pressure decrease					
TCO_MEM_FUP_EFP_H	O/V/S	0...FEH	-48...142.5	0.75	°C
Coolant temperature for the pressure increase					
TCO_MEM_FUP_EFP_L	O/V/S	0...FEH	-48...142.5	0.75	°C
Coolant temperature for the pressure decrease					
TCO_TMP_L	V	0...FEH	-48...142.5	0.75	°C
Temporary coolant temperature for the pressure decrease					
T_EFP_ACT_L	V	0...FFFFH	0...655.35	0.01	s
Timer for the pressure decrease					
FUP_EFP_TMP_H	V	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Temporay pressure at observation start for the pressure increase					
TCO_TMP_H	V	0...FEH	-48...142.5	0.75	°C
Temporary coolant temperature for the pressure increase					
T_EFP_ACT_H	V	0...FFFFH	0...655.35	0.01	s
Timer for the pressure increase					
FUP_EFP_TMP_L	V	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Temporay pressure at observation start for the pressure decrease					
LV_ERR_FUP_EFP_LIH	V	0...1H	0...1	1	-
Logical bit indicating limp home of EFP pump requested					
LV_PUMP_PRE_RUN_STOP	V	0...1H	0...1	1	-
Logical bit indicating pre run of the pump stopped					
T_EFP	V	0...FFFFH	0...655.35	0.01	s
Time counter for activated EFP					

Input data:

STATE_EFP	LV_IGK	LV_ES	LV_ENG_OFF_N_CON
STATE_EFP_CRASH_CA	LV_FUP_LIH_REQ	LV_ERR_FUP_EFP	ERR_SYM_FUP_EFP

N			
LV_ST_END	FUP_EFP_MES	FUP_EFP_SP	LV_PIN_ICH
LV_ACR_EFPPWM_TEST_REQ	FUP_EFP	TCO	TIA
FTL	LV_ERR_EFPPWM_PLAUS	LC_FUP_EFP_AVL	LV_EFP_CTL_AD_CLR_EXT_REQ

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FUP_EFP_MEM_L_DIF	1	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Fuel pressure difference to store values for start pressure decrease observation					
C_FUP_EFP_ST_H_MIN	1	0...FFFFH	0...1.73888E+5	2.6533608	hPa
Start fuel pressure level to store values for start pressure increase observation in any case					
C_TCO_MEM_FUP_EFP_L_TOL	1	0...FEH	-48...142.5	0.75	°C
Coolant temperature threshold to start fuel pressure decrease observation					
C_TIA_FUP_EFP_L_HYS	1	0...FEH	0...190.5	0.75	°C
Intake temperature hysteresis for fuel pressure decrease observation					
C_T_EFP	1	0...FFFFH	0...655.35	0.01	s
Time counter for activated EFP within power latch					
C_VFF_EFP_ADD	1	0...FFH	0...255	1	l/h
Additive value for the amount of fuel from the pump					
LC_EFPPWM_MAN_ACT	1	0...1H	0...1	1	-
Switch for manual EFPPWM					

34.10.1 Flow controlled electrical fuel pump

The electrical fuel pump is controlled by the EFPPWM.

The PI-controller is used to compensate deviations from the nominal fuel pressure after the fuel pump.

In this module the switching conditions for the main module are calculated. This includes also limp home and crash reactions. Application Condition

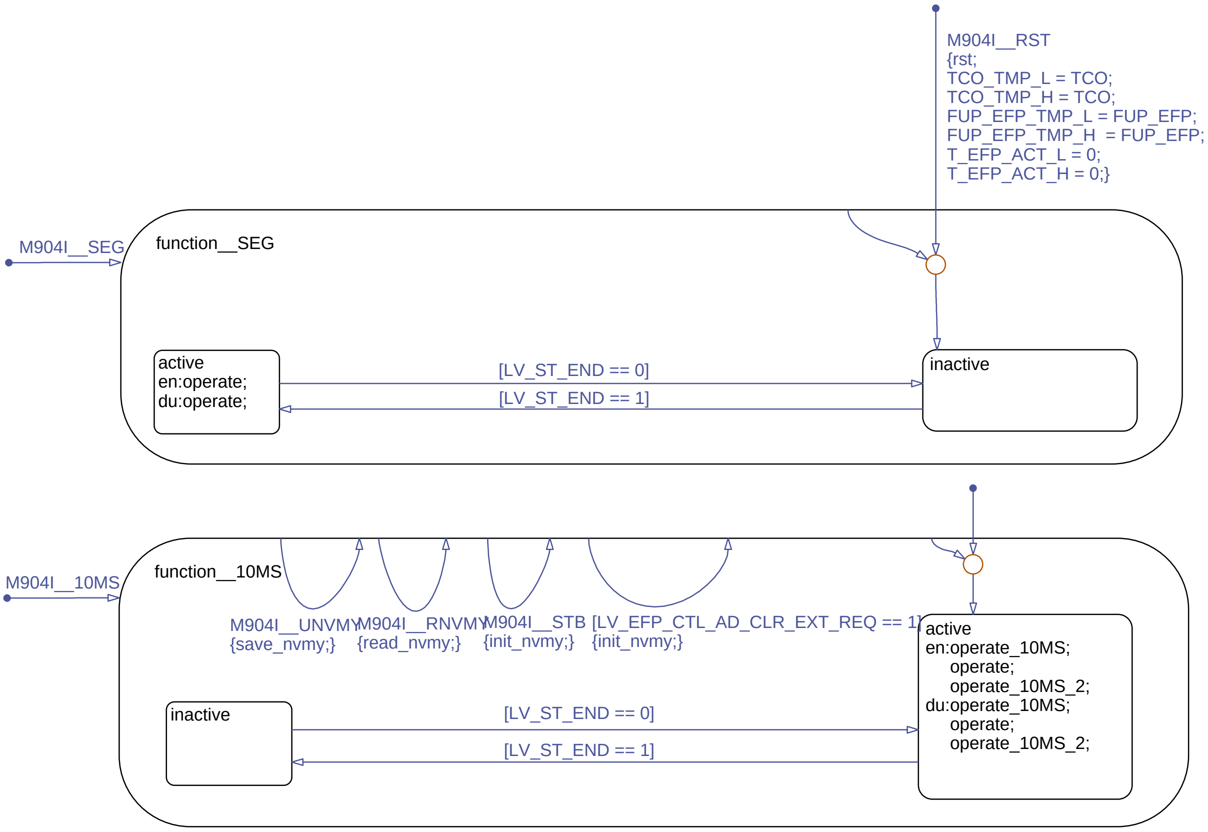


Figure 77 M904I/ APP_CDN/ Chart

Function Description

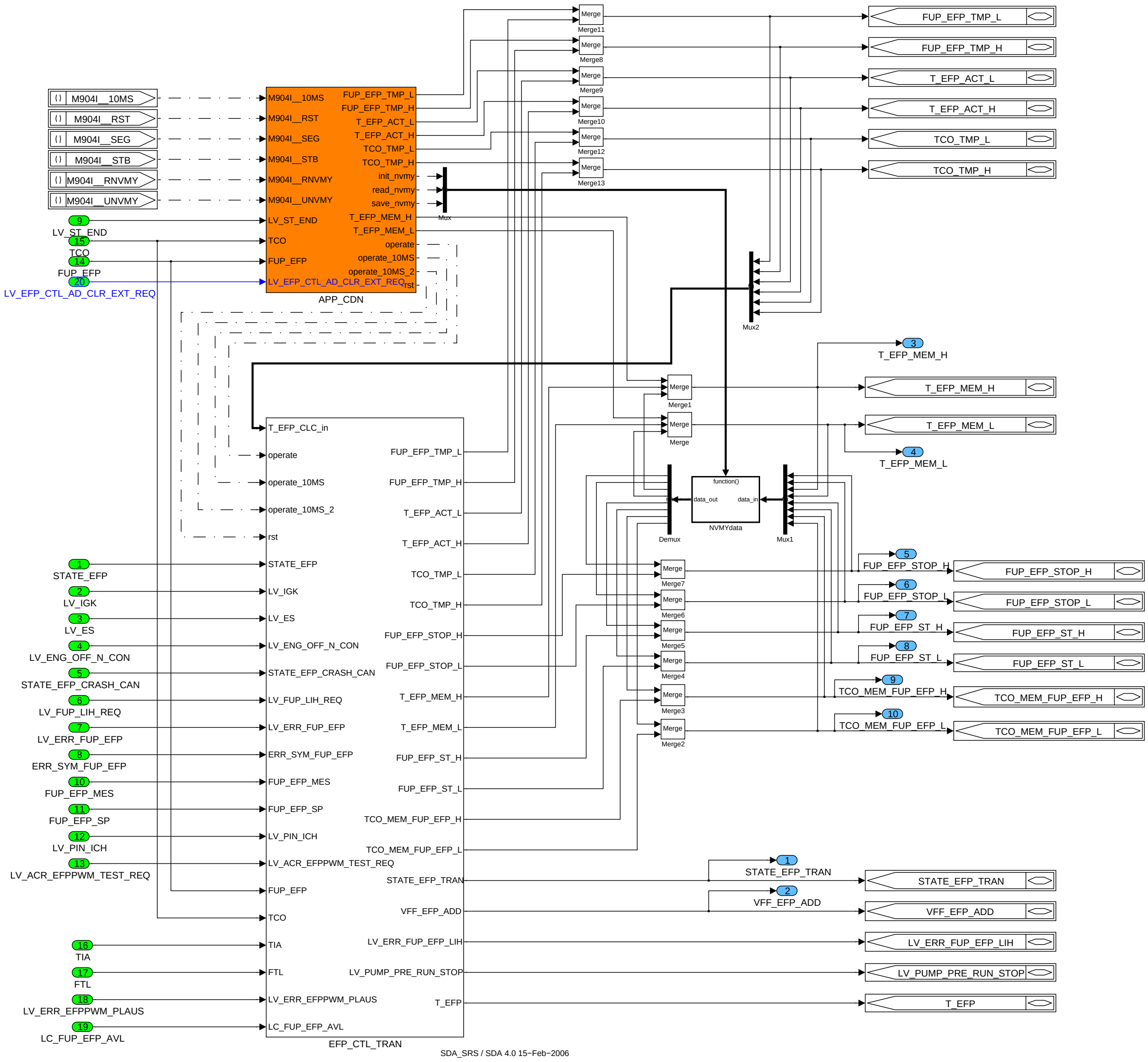


Figure 78 M904I

34.10.1.1 Electrical fuel pump control transition

The function is consisting out of the timer calculation for the EFP and the transition and limp home calculation. Additionally a calculation part for pressure increase and decrease observation is included.

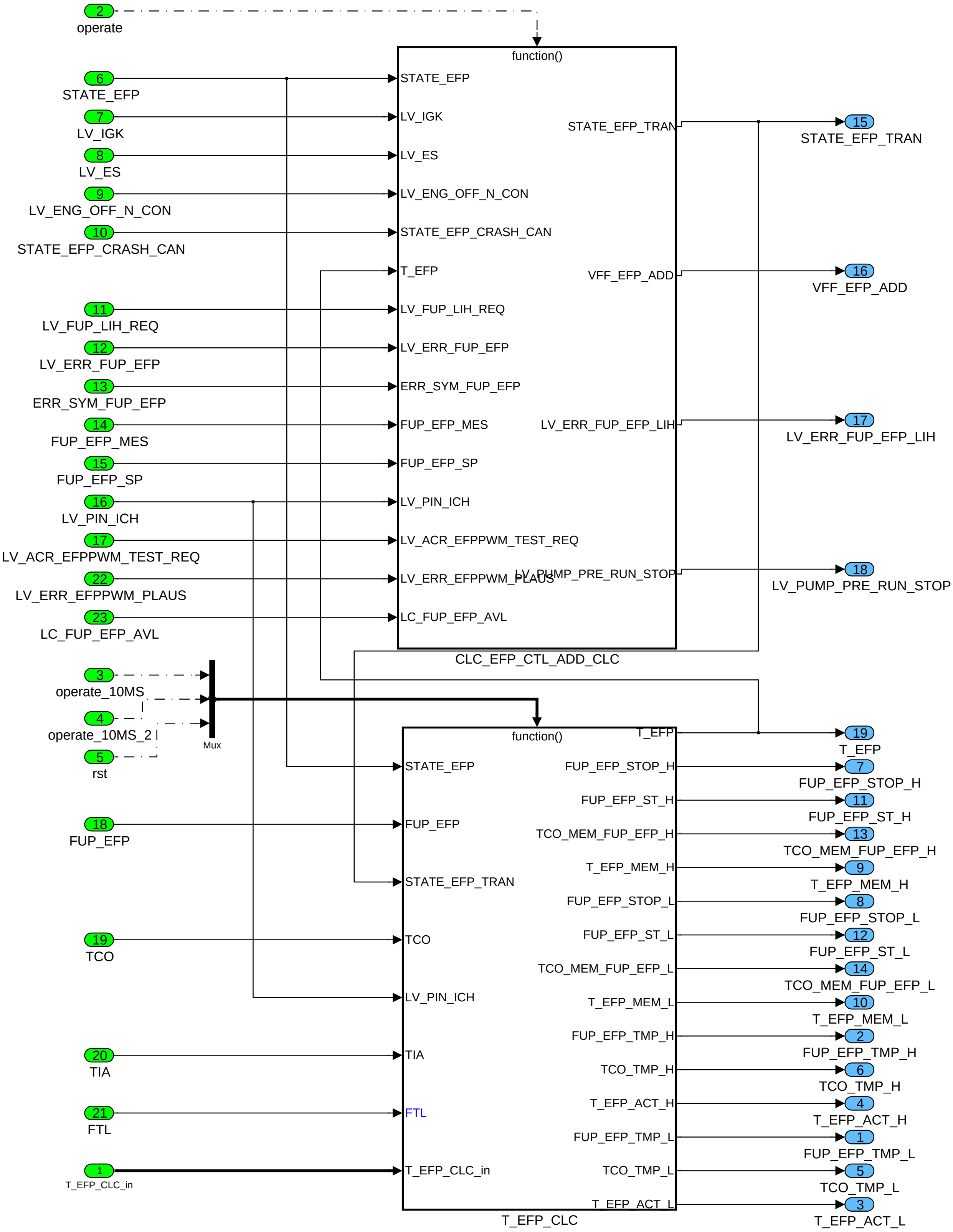


Figure 79 M904I/ EFP_CTL_TRAN

Control additive calculation

This module is consisting out of the transition condition calculation, the addition flow calculation and the limp home determination.

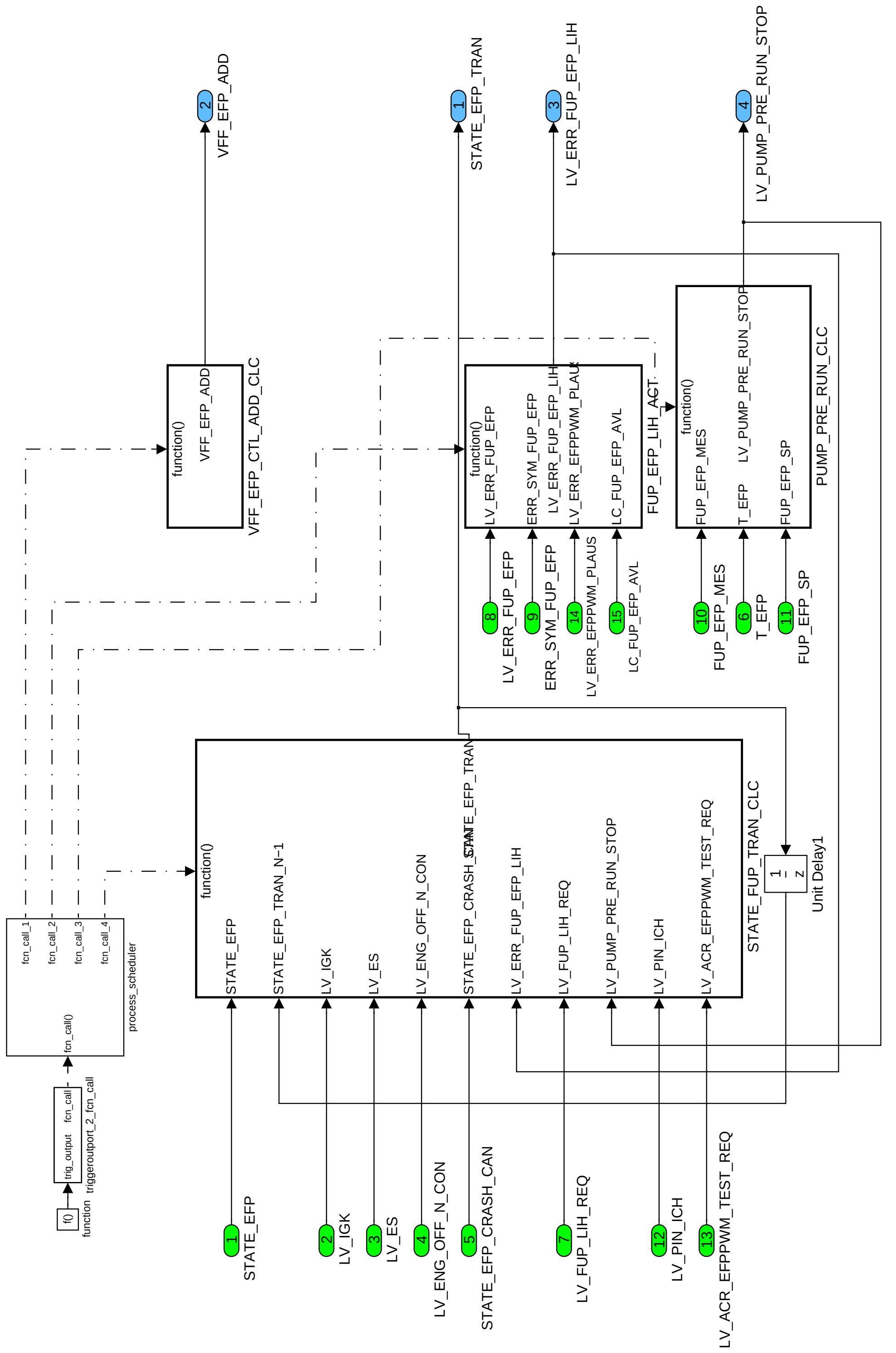


Figure 80 M904I/ EFP_CTL_TRAN/ CLC_EFP_CTL_ADD_CLC

Chapter Fuel supply	Baseline 4DC3940S		Include File 30904I02.00B
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 5635 of 9643

Limp home active

The limp home for the low pressure fuel pump is triggered by different error flags.

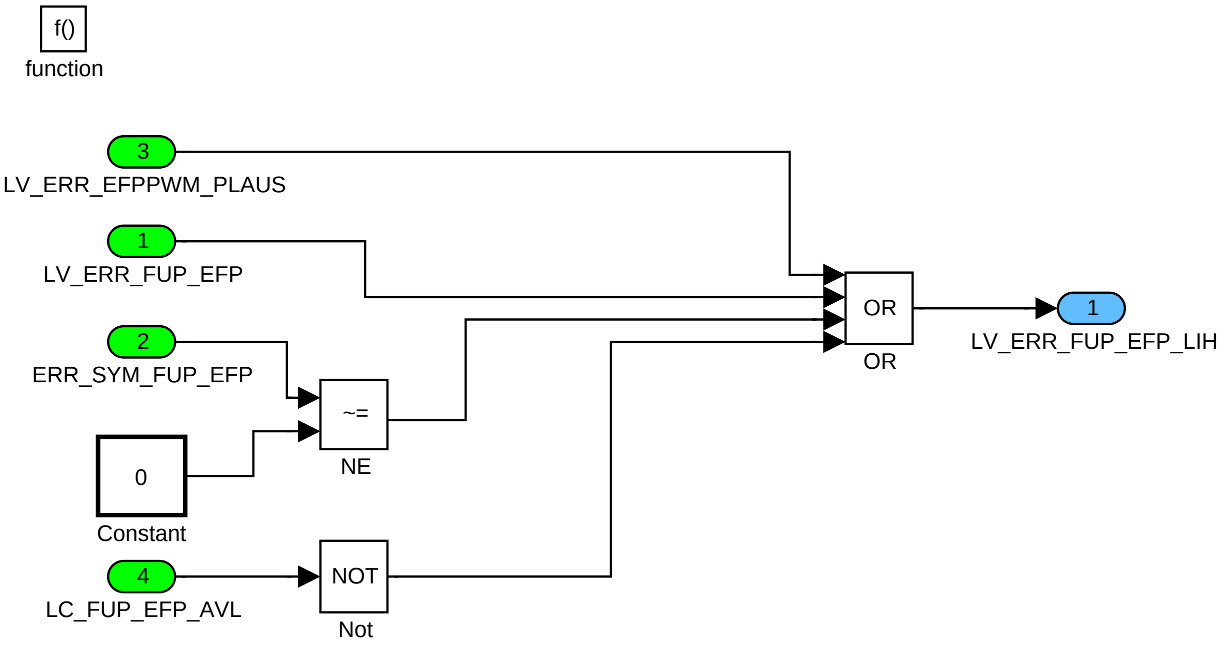
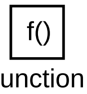


Figure 81 M904I/ EFP_CTL_TRAN/ CLC_EFP_CTL_ADD_CLC/ FUP_EFP_LIH_ACT

Pump pre-run calculation

The pre-run of the low pressure pump is stopped when the fuel pressure setpoint of the low pressure pump has been reached by the measured fuel pressure (without filtering) or the timer has expired.



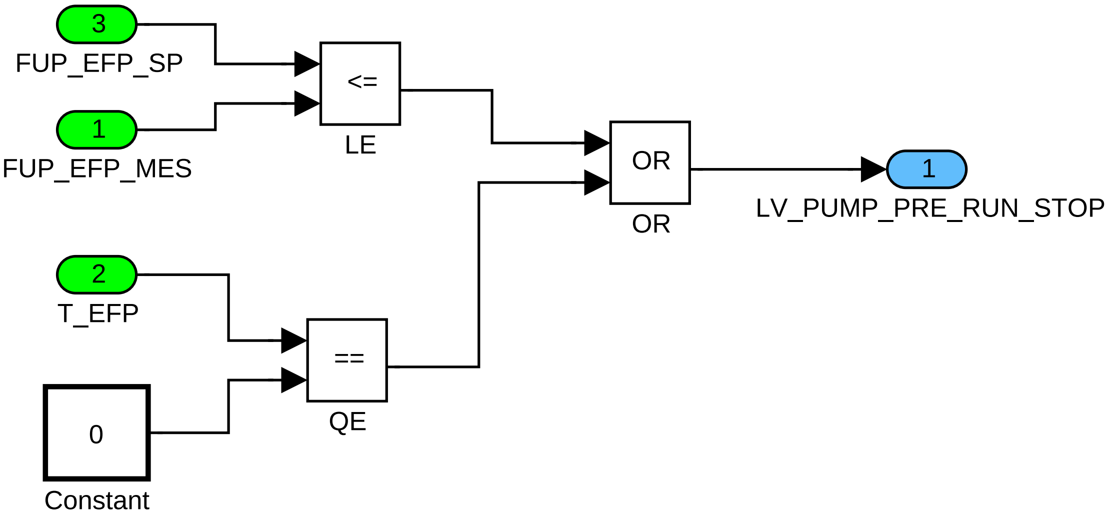


Figure 82 M904I/ EFP_CTL_TRAN/ CLC_EFP_CTL_ADD_CLC/ PUMP_PRE_RUN_CLC

State fuel pressure transition calculation

The transition conditions are split into two parts due to a better overview.

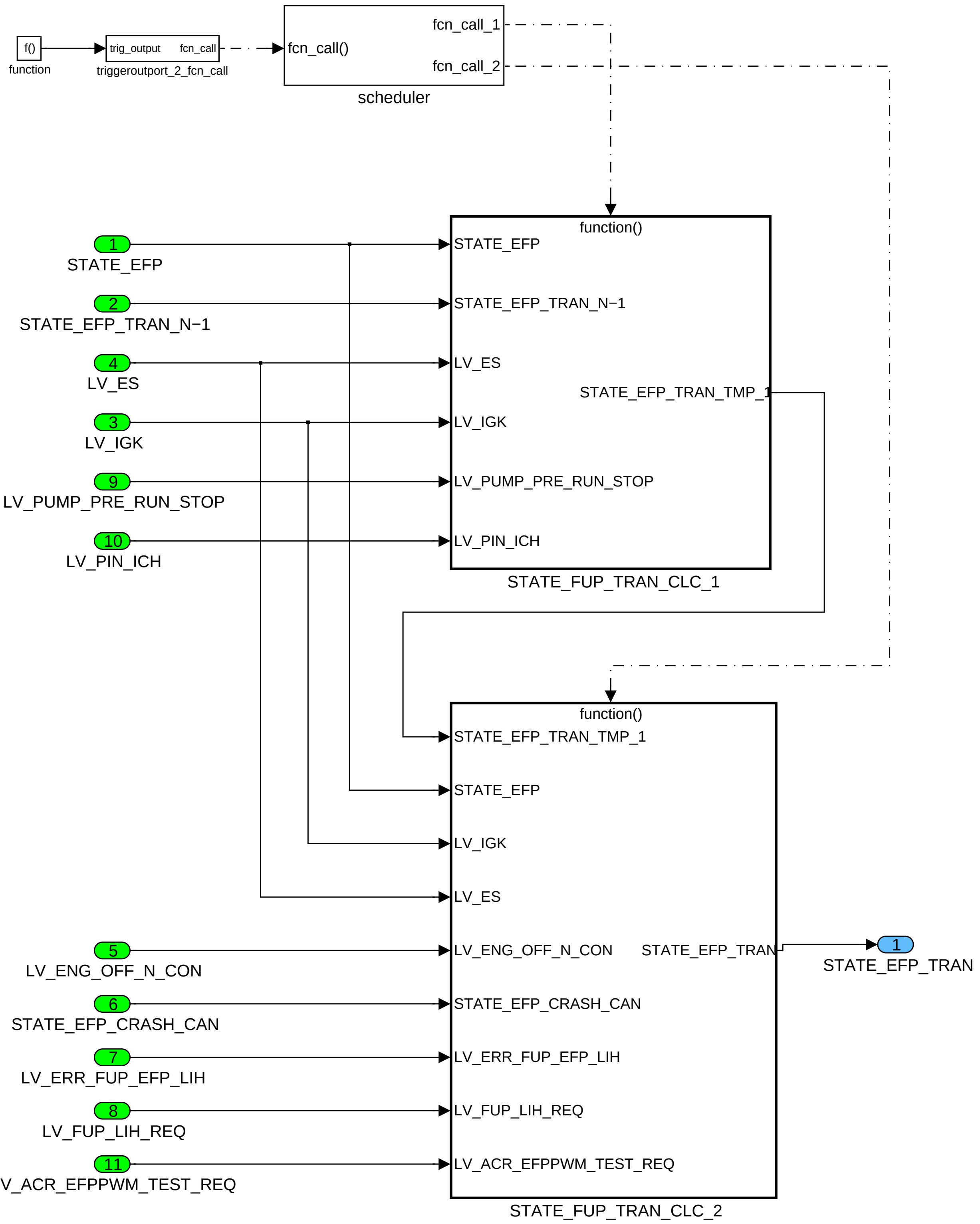


Figure 83 M904I/ EFP_CTL_TRAN/ CLC_EFP_CTL_ADD_CLC/ STATE_FUP_TRAN_CLC

State fuel pressure transition calculation 1

In this part the first part of the transition conditions for the EFP state diagram are calculated.

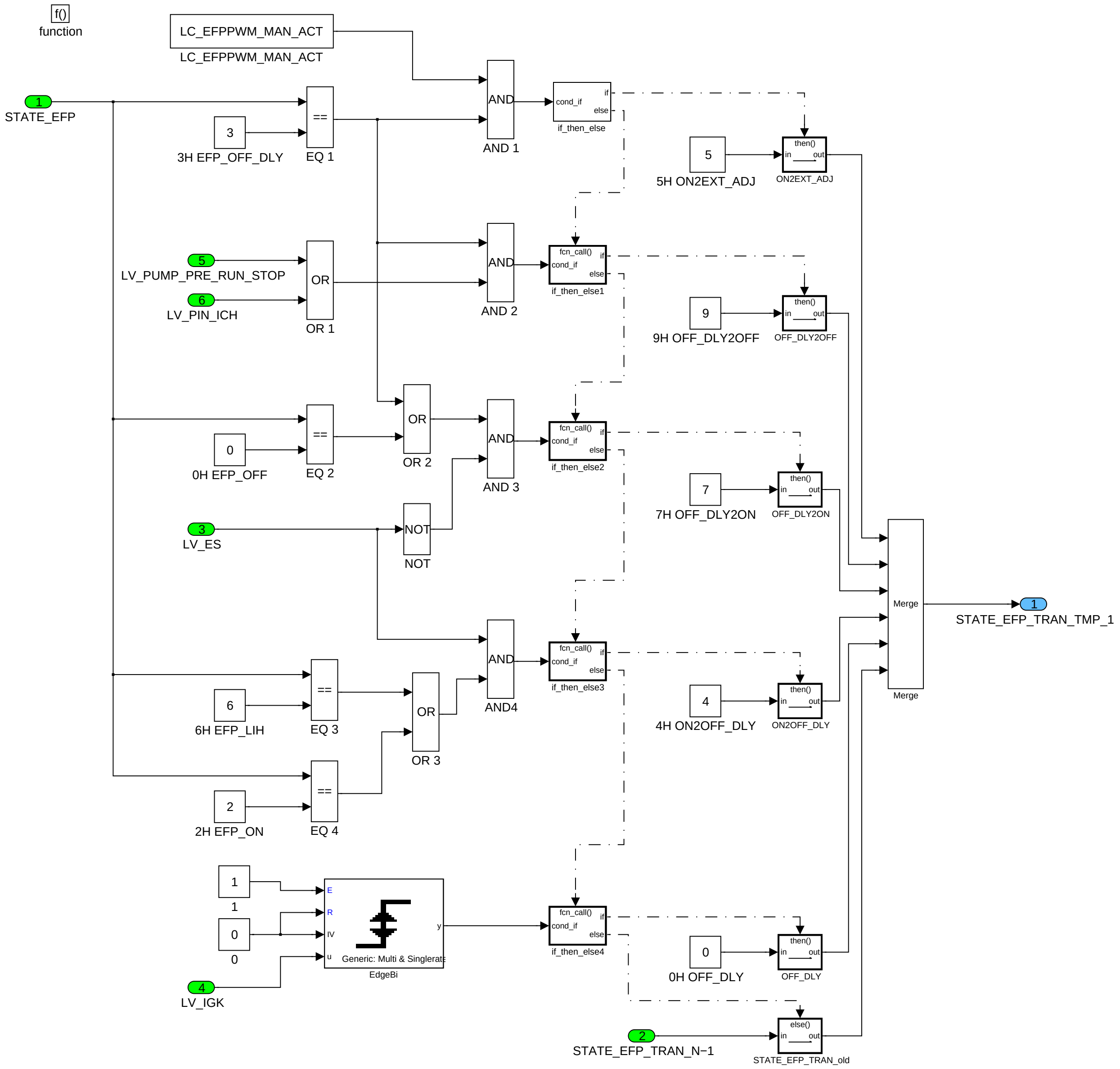
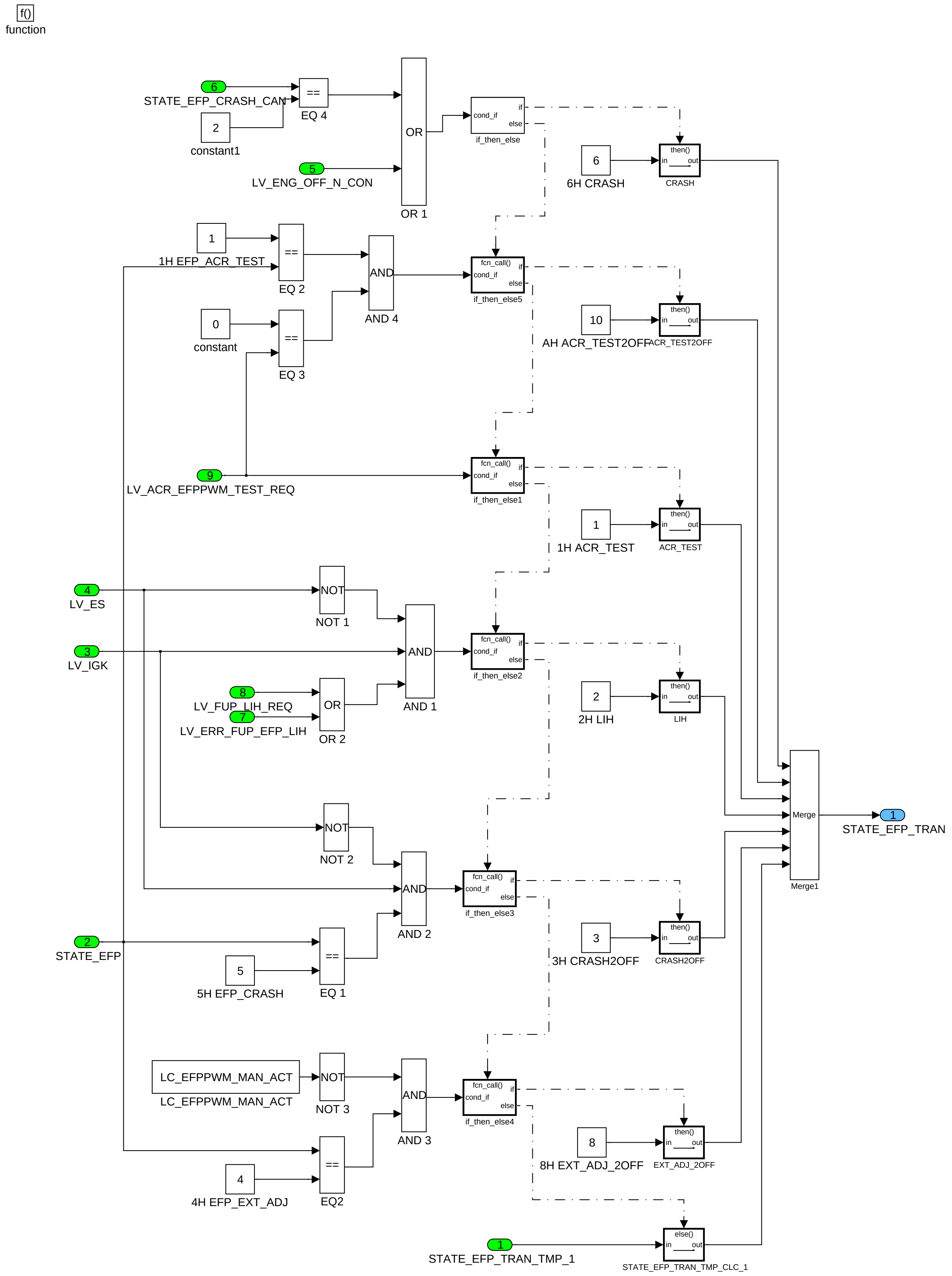


Figure 84 M904I/ EFP_CTL_TRAN/ CLC_EFP_CTL_ADD_CLC/ STATE_FUP_TRAN_CLC/ STATE_FUP_TRAN_CLC_1

State fuel pressure transition calculation 2

In this part the first part of the transition conditions for the EFP state diagram are calculated. Main parts are the limp home conditions and the safety and security reasons.



Chapter Fuel supply	Baseline 4DC3940S		Include File 30904I02.00B
Designed by Released by	Date 2008-07-01 2008-07-01	Department	Sign
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 5642 of 9643

Figure 85 M904I/ EFP_CTL_TRAN/ CLC_EFP_CTL_ADD_CLC/ STATE_FUP_TRAN_CLC/ STATE_FUP_TRAN_CLC_2

Volume fuel flow

An additional offset for the requested flow by the low pressure pump can be adjusted with the following constant.

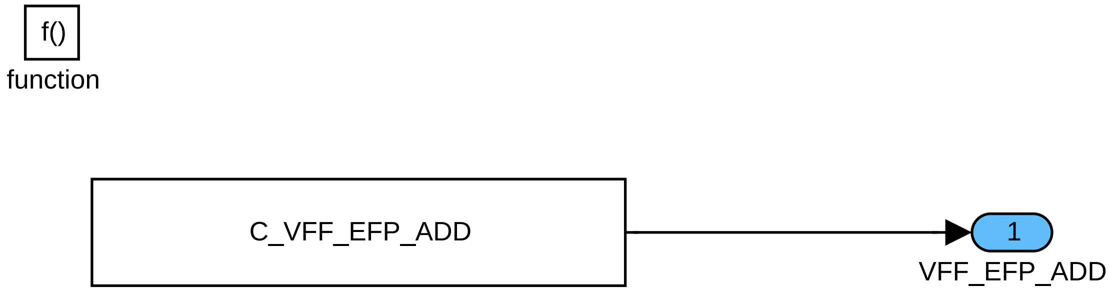


Figure 86 M904I/ EFP_CTL_TRAN/ CLC_EFP_CTL_ADD_CLC/ VFF_EFP_CTL_ADD_CLC

Time calculation

The calculation of the timer for the EFP is done within this module. A reset of this timer is done by the reset or a calculated condition. The calculation part for the pressure increase and decrease is placed within the subblock.

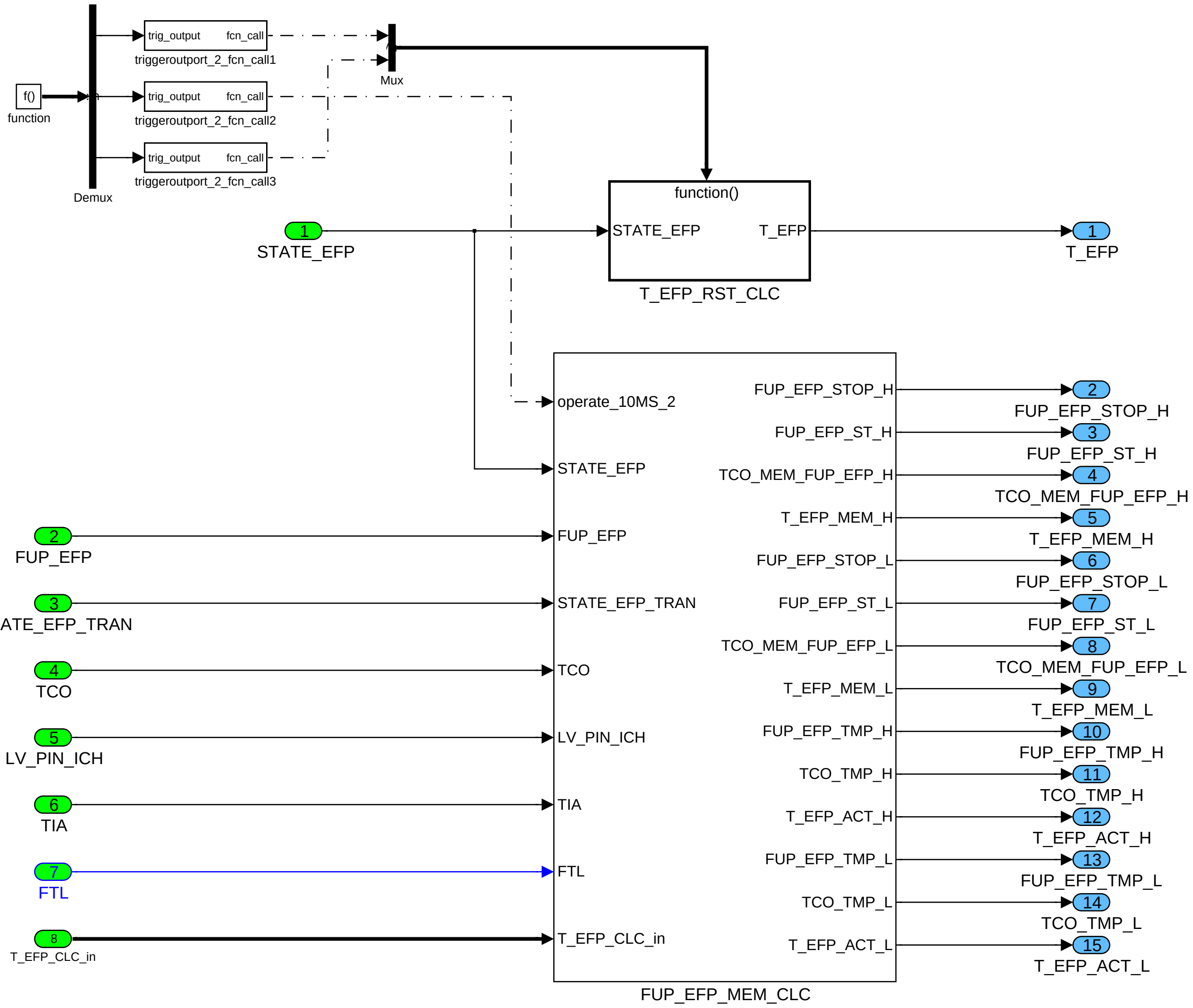


Figure 87 M904I/ EFP_CTL_TRAN/ T_EFP_CLC

Electrical fuel pump memorize calculation

The calculation of the low pressure system observation is split into a calculation for the high part and on calculation for the low part. Each part has an output for the start pressure, stop pressure, coolant temperature and the time for rising/falling pressure.

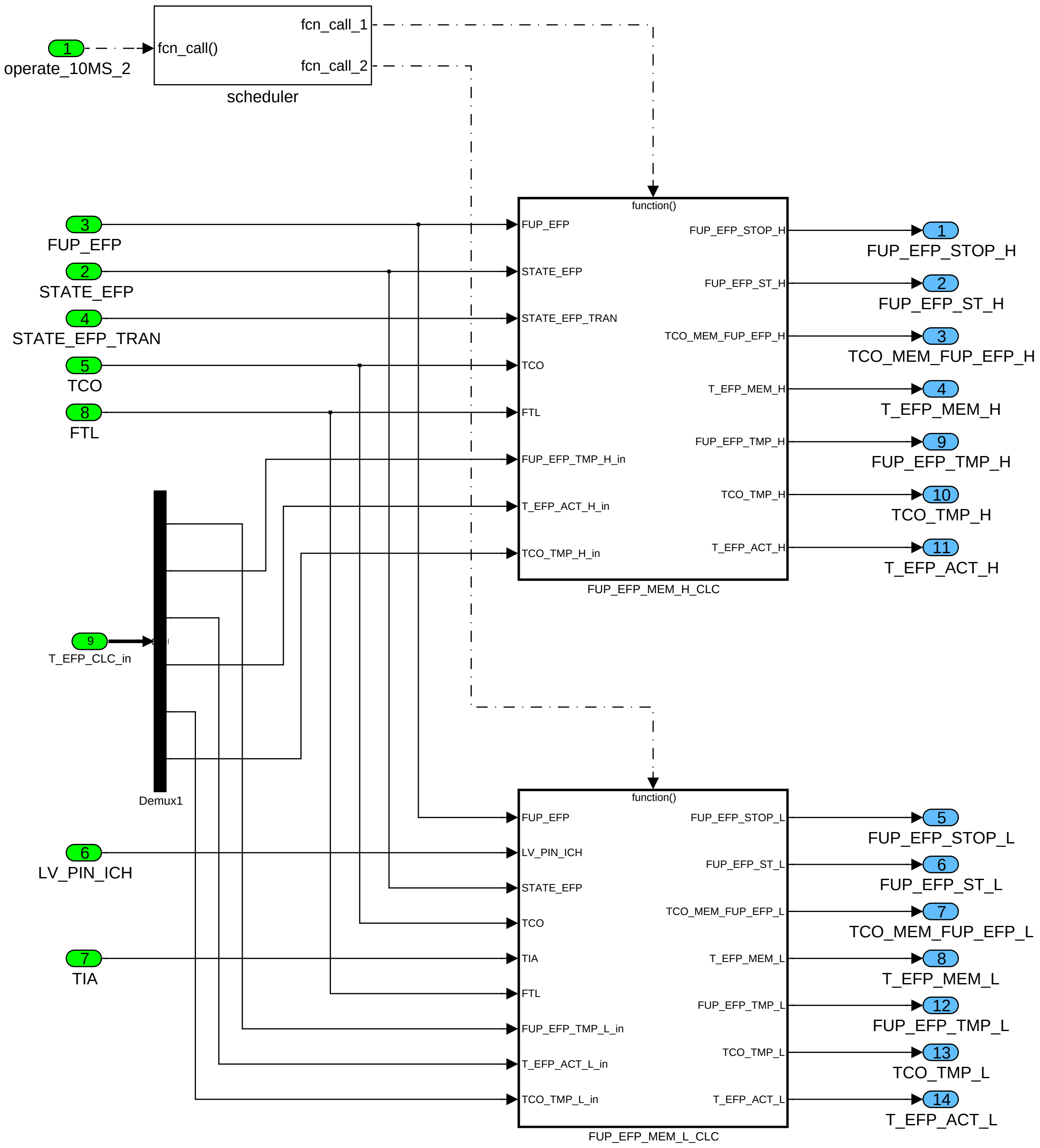


Figure 88 M904I/ EFP_CTL_TRAN/ T_EFP_CLC/ FUP_EFP_MEM_CLC

Electrical fuel pump memorize high calculation

In this part the high signals are calculated.

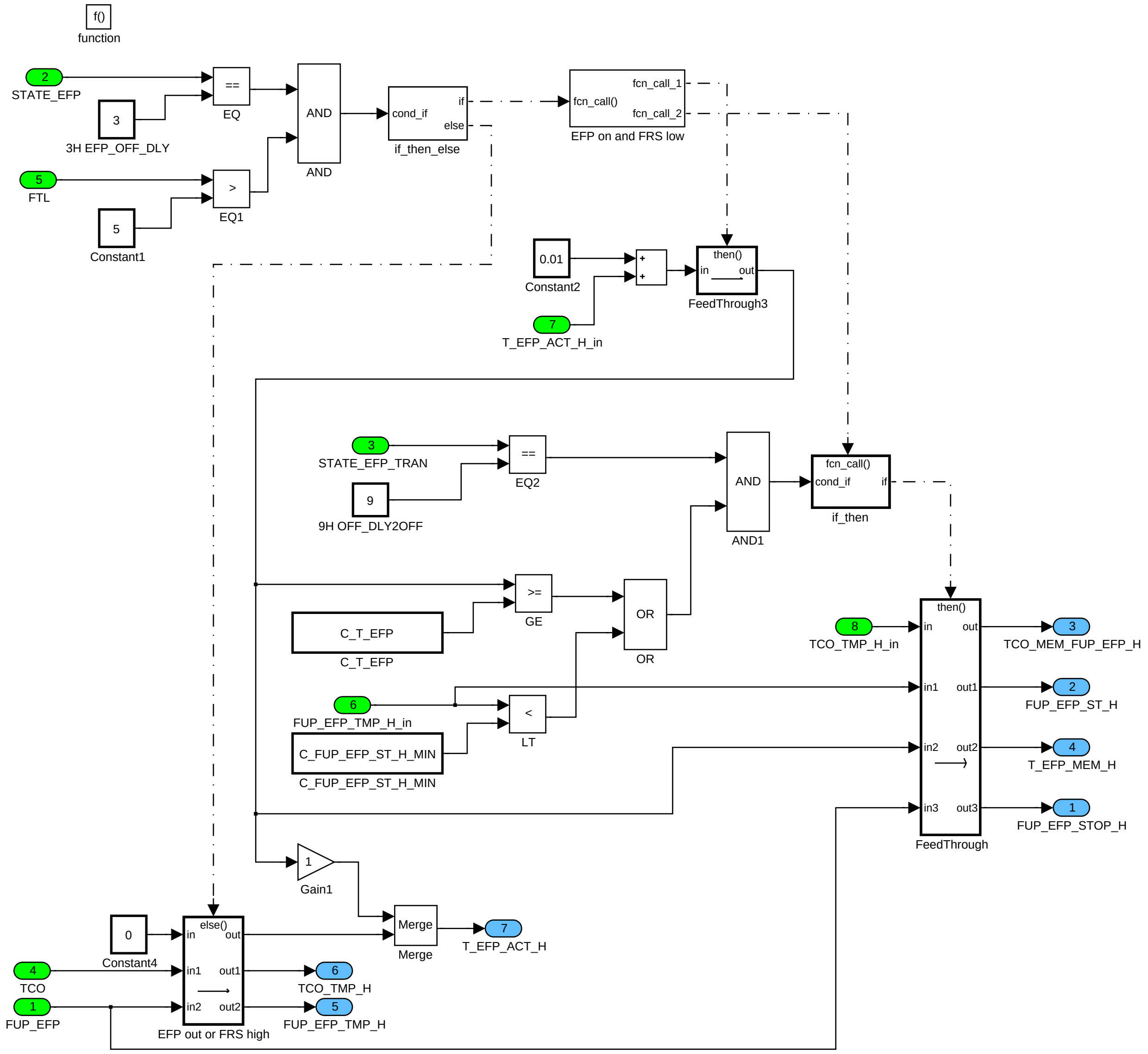


Figure 89 M904I/ EFP_CTL_TRAN/ T_EFP_CLC/ FUP_EFP_MEM_CLC/ FUP_EFP_MEM_H_CLC

Electrical fuel pump memorize low calculation

In this part the low signals are calculated.

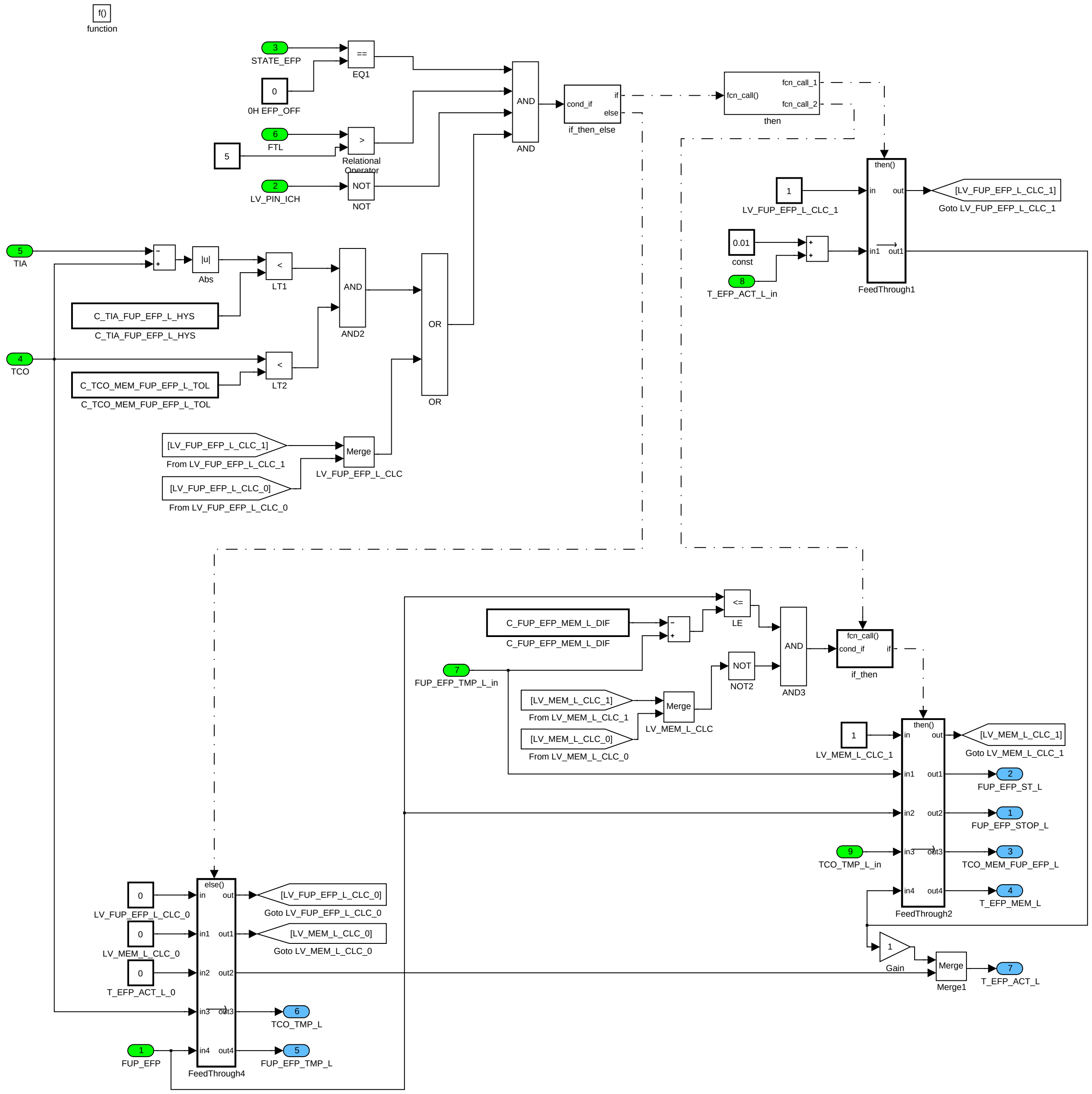


Figure 90 M904I/ EFP_CTL_TRAN/ T_EFP_CLC/ FUP_EFP_MEM_CLC/ FUP_EFP_MEM_L_CLC

Reset of the timer

The reset of the timer is calculated according to the following module.

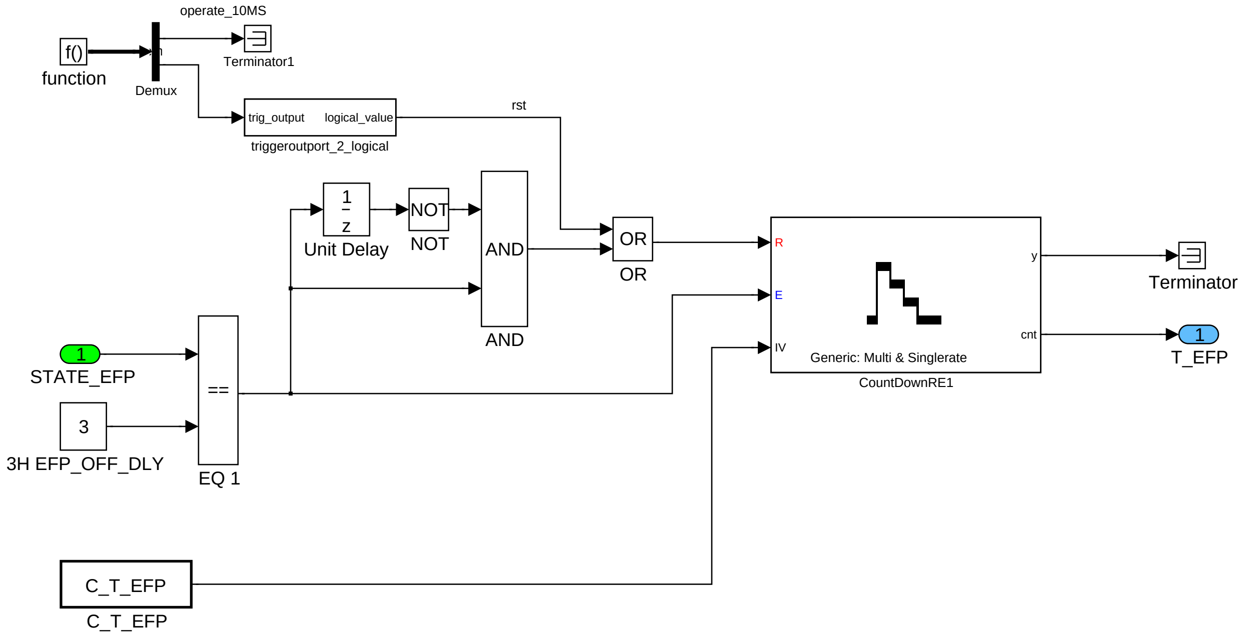


Figure 91 M904I/ EFP_CTL_TRAN/ T_EFP_CLC/ T_EFP_RST_CLC

34.10.1.2 Initialization

In this part outputs are initialised.

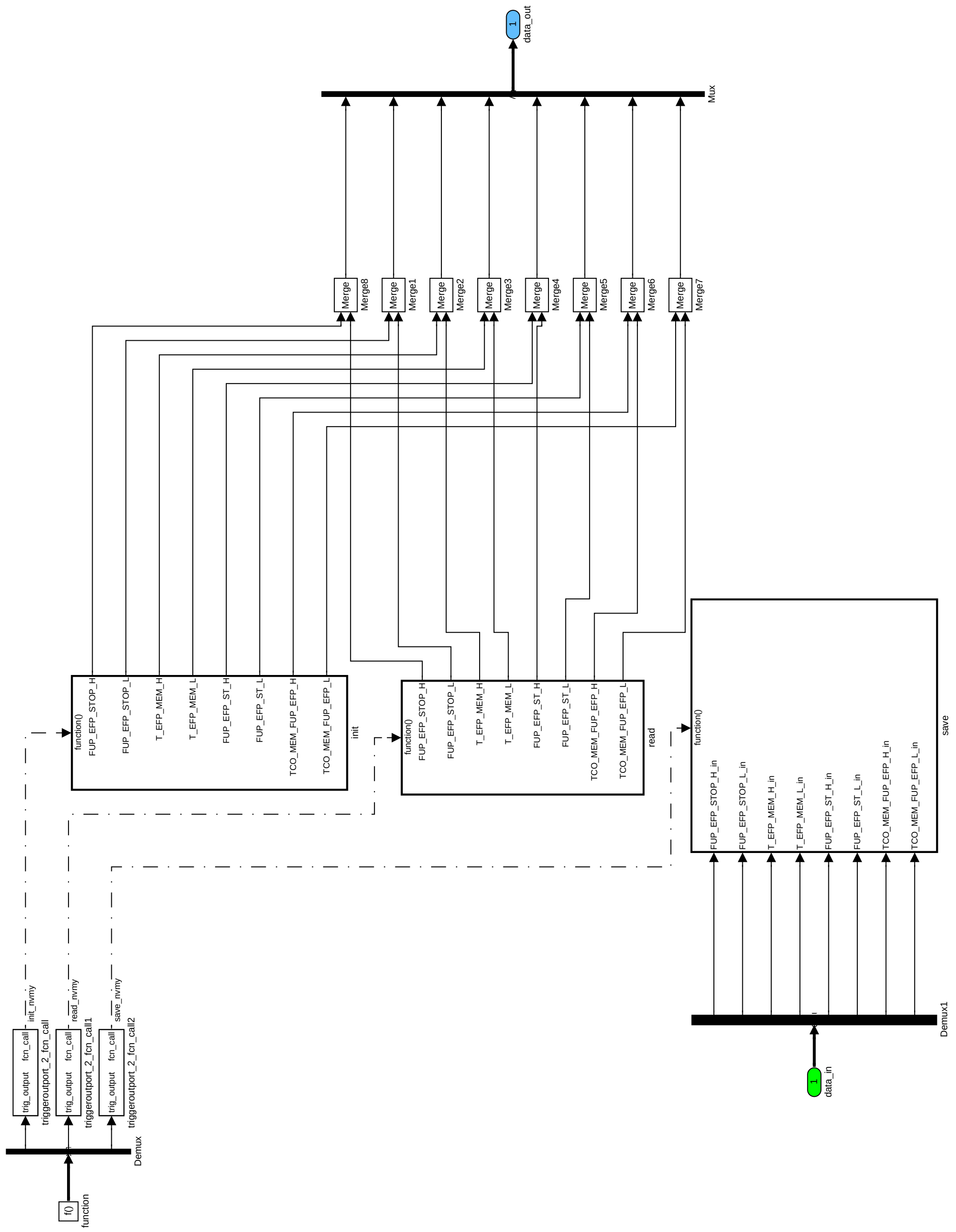


Figure 92 M904I/ NVMYdata

Initialization with FFFFH or FEH

If EEPROM error or ECU brandnew or cleared by service tool then the outputs are initialised with FFFFH or FEH.

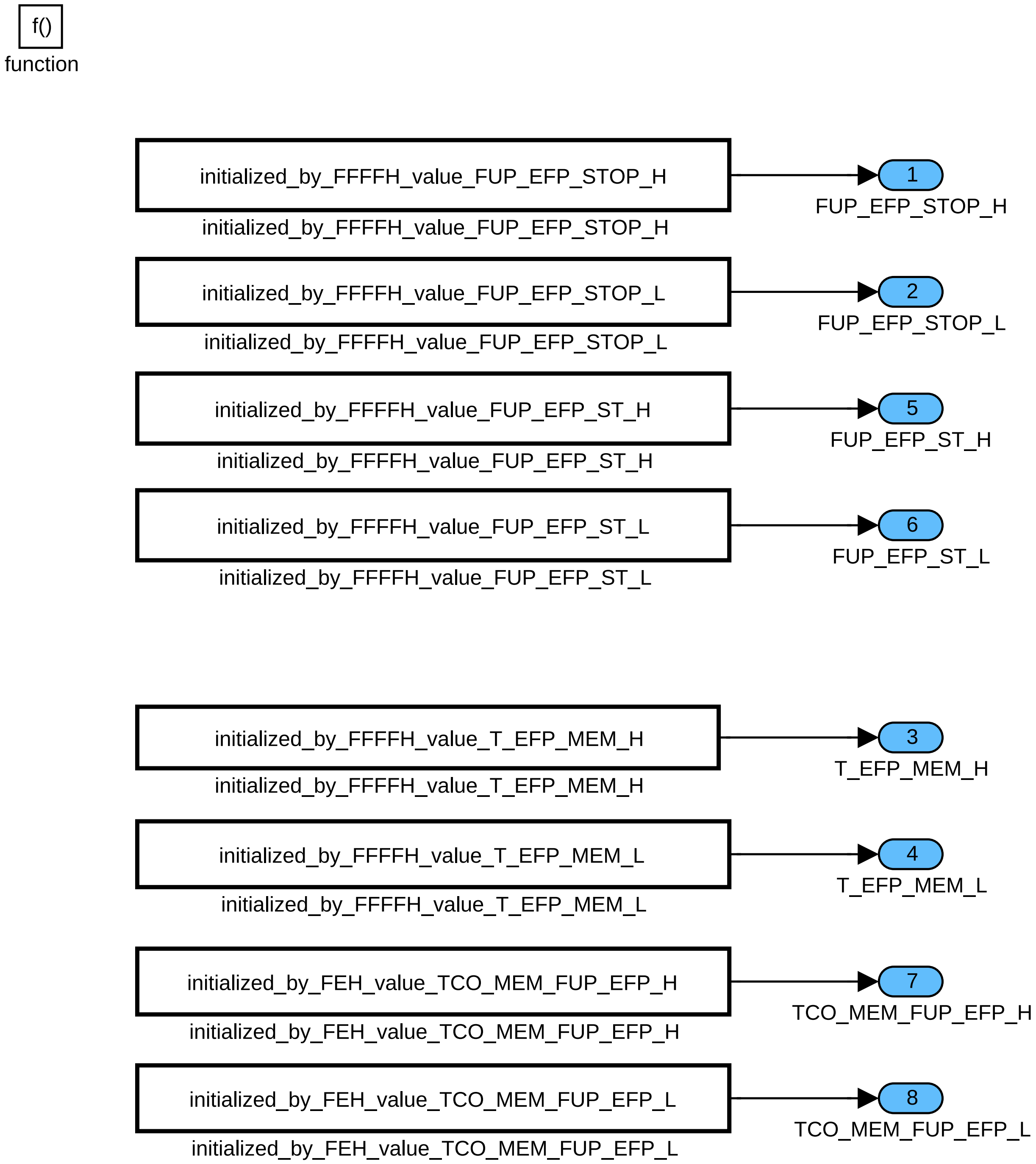


Figure 93 M904I/ NVMYdata/ init

Output read

In this part outputs are initialised by EEPROM value.

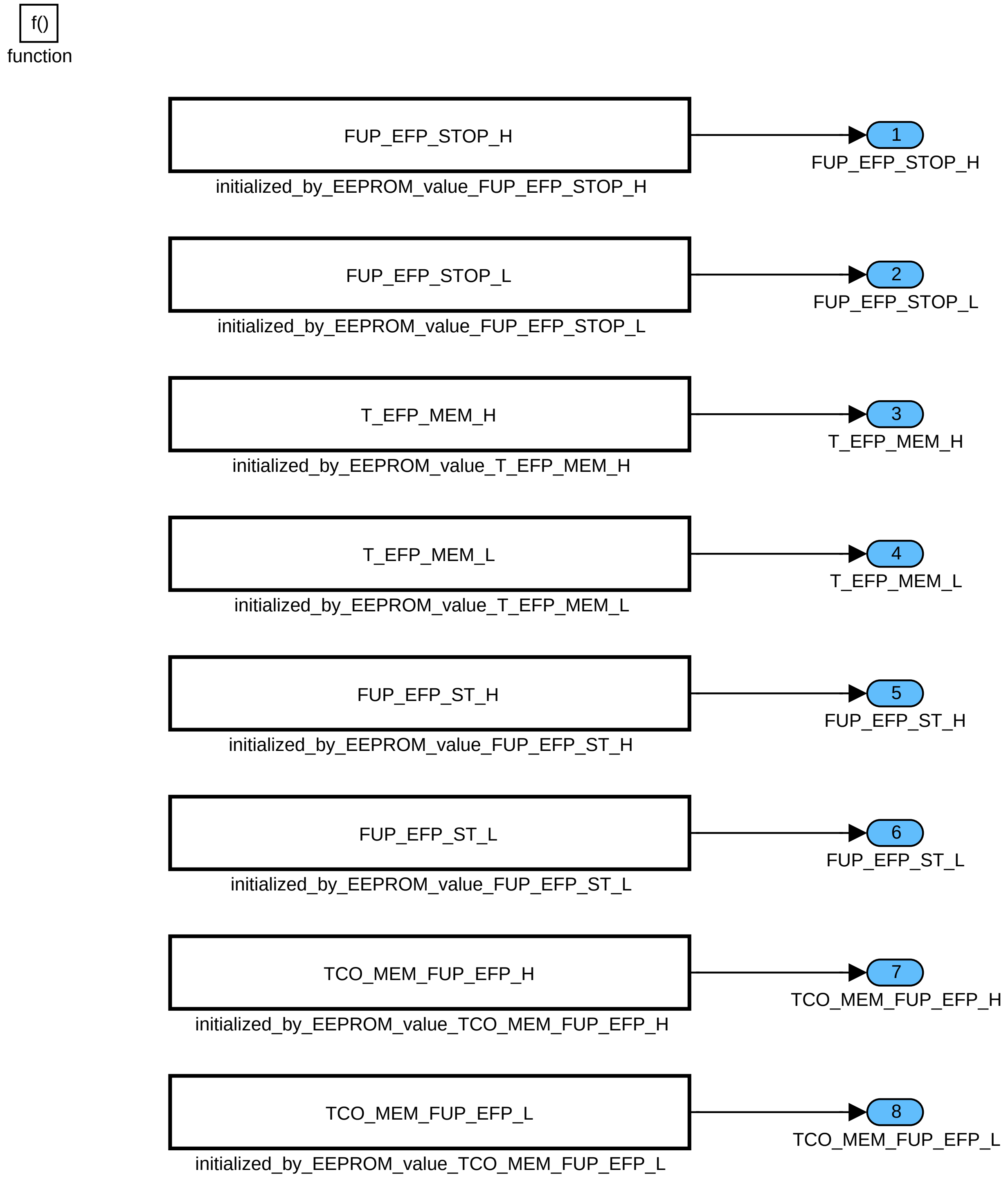


Figure 94 M904I/ NVMYdata/ read

Output save

In this part outputs are saved on the EEPROM.

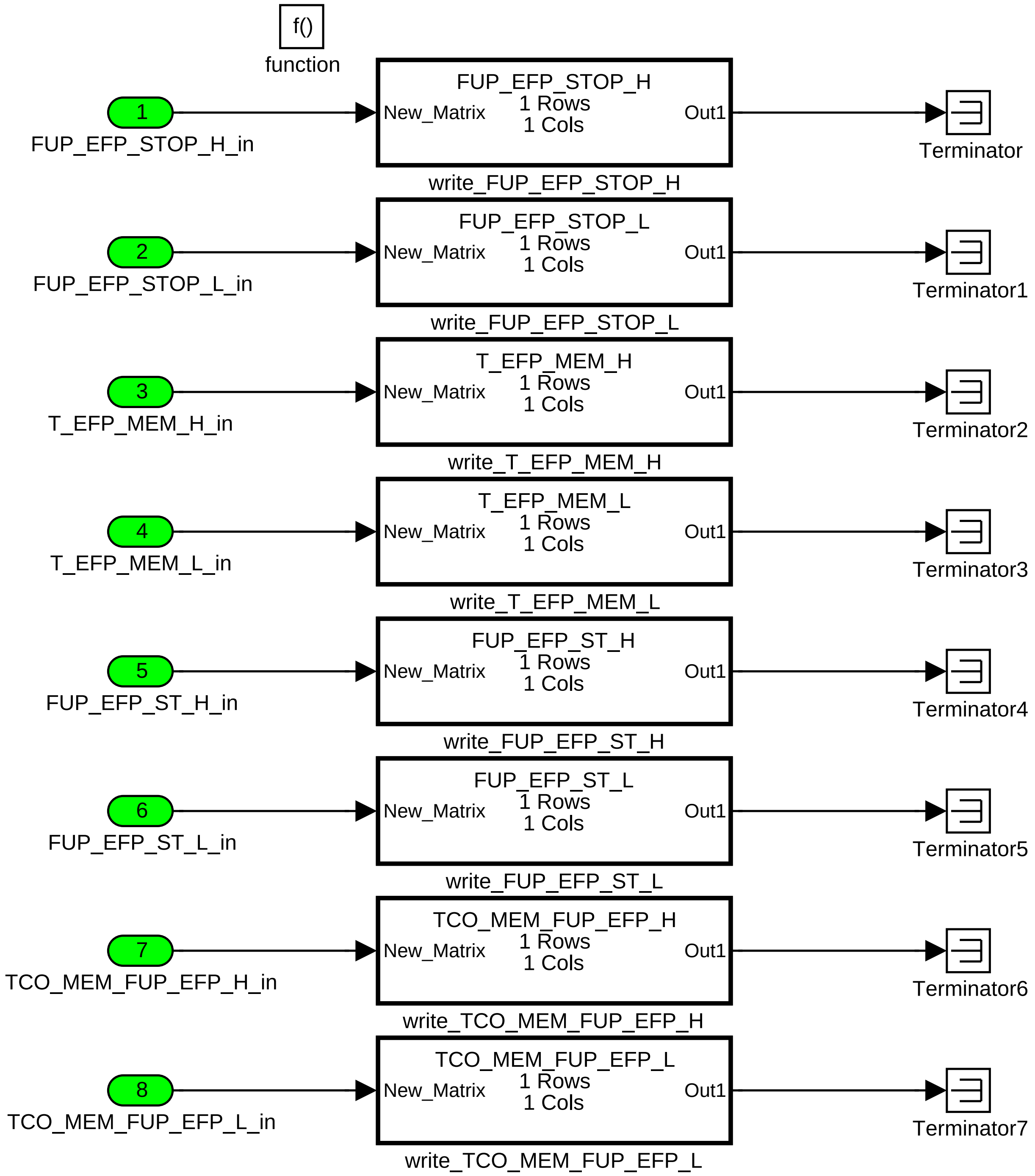


Figure 95 M904I/ NVMYdata/ save

34.11 Fuel pressure setpoint calculation

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
FUP_H_SP_S	O/V	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint for stratified mode, high range					
FUP_RNG_H_SP	O/V	0... FFFFH	0... 255.99609375	3.90625e-3	[MPa]
Fuel pressure setpoint					
FUP_RNG_H_SP_S	O/V	0... FFFFH	0... 255.99609375	3.90625e-3	[MPa]
Fuel pressure setpoint for stratified mode					
FUP_SP	O/V	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint					
FUP_SP_GRD_ACT	V	0... FH	0... 15	1	[-]
Indicator for gradient limitation out of start active					
FUP_SP_ST	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure for start					
FUP_SP_TMP	O/V	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint (before limitation)					
FUP_SP_TMP_1	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Intermediate value for FUP_SP filter					
LV_FUP_SP_ADD	O/V	0... 1H	0... 1	1	[-]
Logical variable indicating that the additional fuel pressure setpoint in the PUC phase is enable					

Input Data:

N_32	MFF_SP_FUP_CTL	LV_FUP_SP_SWI	LV_PU
LV_PUC	LV_ST_END	LV_FUP_SP_EXT_REQ	LV_ES
FUP_SP_EXT	FUP_H_SP_S_EXT	FUP_RNG_H_SP_CH	STATE_CH
TCO_ST	STATE_MPLH_MOD	FUP_RNG_H_SP_OHP	NR_CYL_VVL_H_ACT
C_CTR_MIN_VFF_VCV_MIN_AD_SWI	NC_NR_N_FUP_SP	LV_ST_H_PRS	NC_NR_MFF_SP_FUP_SP
LV_IGK	LV_FIRST_VLD_TOOTH	LV_FUP_LIH_HOM_VCV_OPEN_REQ	LV_FUP_LIH_L_PRS_CTL_REQ
LV_FUP_LIH_HOM_REQ	CTR_MIN_VFF_VCV_MIN_AD_SWI	T_AST	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CTR_FUP_SP_VVL_LGRD	1	0... FFH	0... 255	1	[-]
Counter for fuel pressure setpoint gradient limitation after switching of VVL					
C_FAC_FUP_SP	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Factor for the fuel pressure setpoint temp change influence to the FUP_SP					
C_FUP_SP_ADD	1	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Additional fuel pressure setpoint in the PUC phase					
C_FUP_SP_CRLC_2	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlation constant for the FUP_SP					

C_FUP_SP_IGK_NOT	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint at IGK off					
C_FUP_SP_IGK_NOT_LIM_DEC	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint decrement during IGK off					
C_FUP_SP_IGK_NOT_LIM_INC	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint increment during IGK off					
C_FUP_SP_L_PRS_ST	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint for low pressure start					
C_FUP_SP_LIH_HOM	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint for homogenous limphome					
C_FUP_SP_LIH_VCV_OPEN	1	0... FFFFH	0... 347776	5.3067216	[hPa]
FUP setpoint for VCV open limphome					
C_FUP_SP_LIM_DEC	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint decrement during normal operation					
C_FUP_SP_LIM_INC	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint increment during normal operation					
C_FUP_SP_MAN	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Manual fuel pressure setpoint					
C_FUP_SP_MPLH_OPP	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint for double injection mode					
C_FUP_SP_PU_LIM_DEC	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint decrement during PU/PUC					
C_FUP_SP_PU_LIM_INC	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint increment during PU/PUC					
C_FUP_SP_ST_LGRD_DEC	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint decrement out of start					
C_FUP_SP_ST_LGRD_INC	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint increment till st end					
C_FUP_SP_VVL_LGRD	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for fuel pressure setpoint increment and decrement after switching of VVL					
C_T_AST_FUP_SP_GRD_SWI	1	0... FFFFH	0... 6553.5	0.1	[s]
Time after start to switch the fuel pressure gradient					
C_T_FUP_SP_ADD_1	1	0... FFFFH	0... 655.35	0.01	[s]
Time of counter number 1 for additional fuel pressure setpoint in the PUC phase					
C_T_FUP_SP_ADD_2	1	0... FFFFH	0... 655.35	0.01	[s]
Time of counter number 2 for additional fuel pressure setpoint in the PUC phase					
IP_FUP_SP_HOM	10*8	0... FFFFH	0... 347776	5.3067216	[hPa]
LDPM_N_32_2_FUSL	8	0... FFH	0... 8160	32	[rpm]
LDPM_MFF_SP_FUP_CTL_1_FUSL	10	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Fuel pressure setpoint for homogenous engine operation					
IP_FUP_SP_PU	10*8	0... FFFFH	0... 347776	5.3067216	[hPa]
LDPM_N_32_2_FUSL	8	0... FFH	0... 8160	32	[rpm]
LDPM_MFF_SP_FUP_CTL_1_FUSL	10	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Fuel pressure setpoint for PU and PUC engine operation					
IP_FUP_SP_S	10*8	0... FFFFH	0... 347776	5.3067216	[hPa]
LDPM_N_32_2_FUSL	8	0... FFH	0... 8160	32	[rpm]
LDPM_MFF_SP_FUP_CTL_1_FUSL	10	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Fuel pressure setpoint for stratified engine operation					
IP_FUP_SP_ST	6*6	0... FFFFH	0... 347776	5.3067216	[hPa]
LDPM_TCO_ST_1_FUSL	6	0... FEH	-48... 142.5	0.75	[°C]
LDP_N_32_IP_FUP_SP_ST	6	0... FFH	0... 8160	32	[rpm]
Fuel pressure setpoint for the start					
LC_FUP_SP_MAN_ACT	1	0... 1H	0... 1	1	[-]
Switch for manual fuel pressure setpoint					

General Information

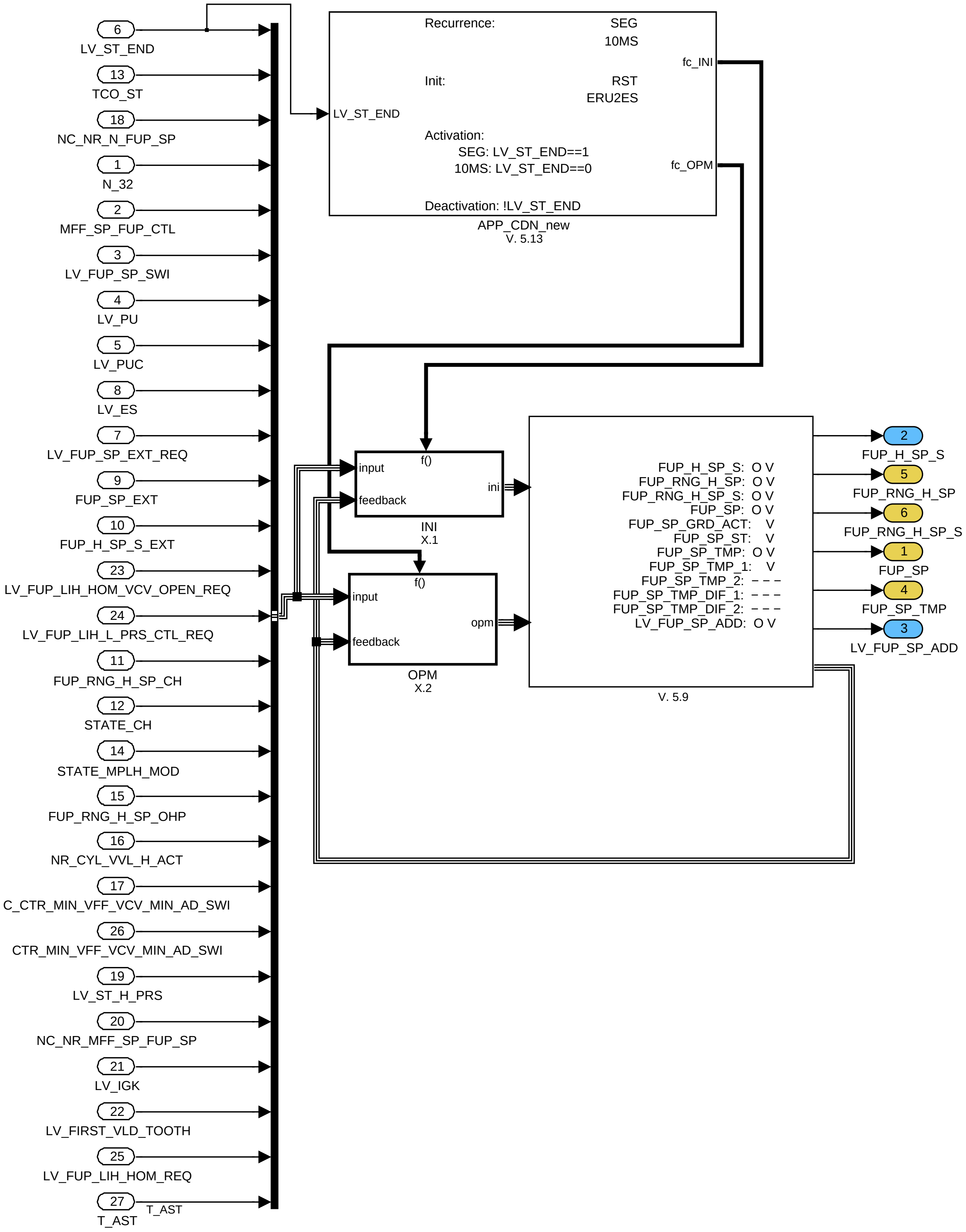
There are maps for homogeneous mode and stratified mode according to the variable, which shows the actual engine mode. In PU and PUC a different map for the fuel pressure setpoint is selected.

Additionally a testbench function is also included. In order to avoid too big fuel pressure setpoint gradients a certain gradient limitation is included. That limitation is consisting out of a increment and decrement limitation. For PU and PUC a different limitation for increment and decrement is selected.

Application Conditions

Initialization: RST, ERU2ES
Recurrence: SEG activated if LV_ST_END==1
10MS activated if LV_ST_END==0
Deactivation: !LV_ST_END

Function description



SDA_SRS / SDA V 5.0.4 30−Nov−2006

Figure 96:

Path: FUSL_M904A

34.11.1 Calculation overview for initialisation

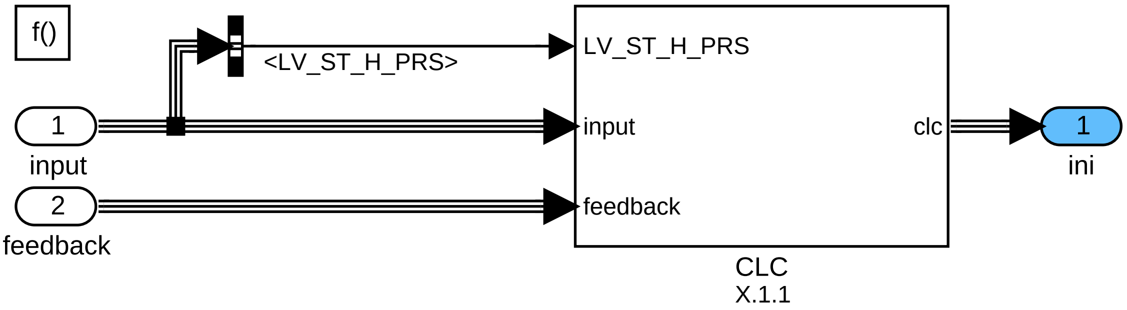


Figure 97:

Path: FUSL_M904A/INI

34.11.1.1 Initialisation

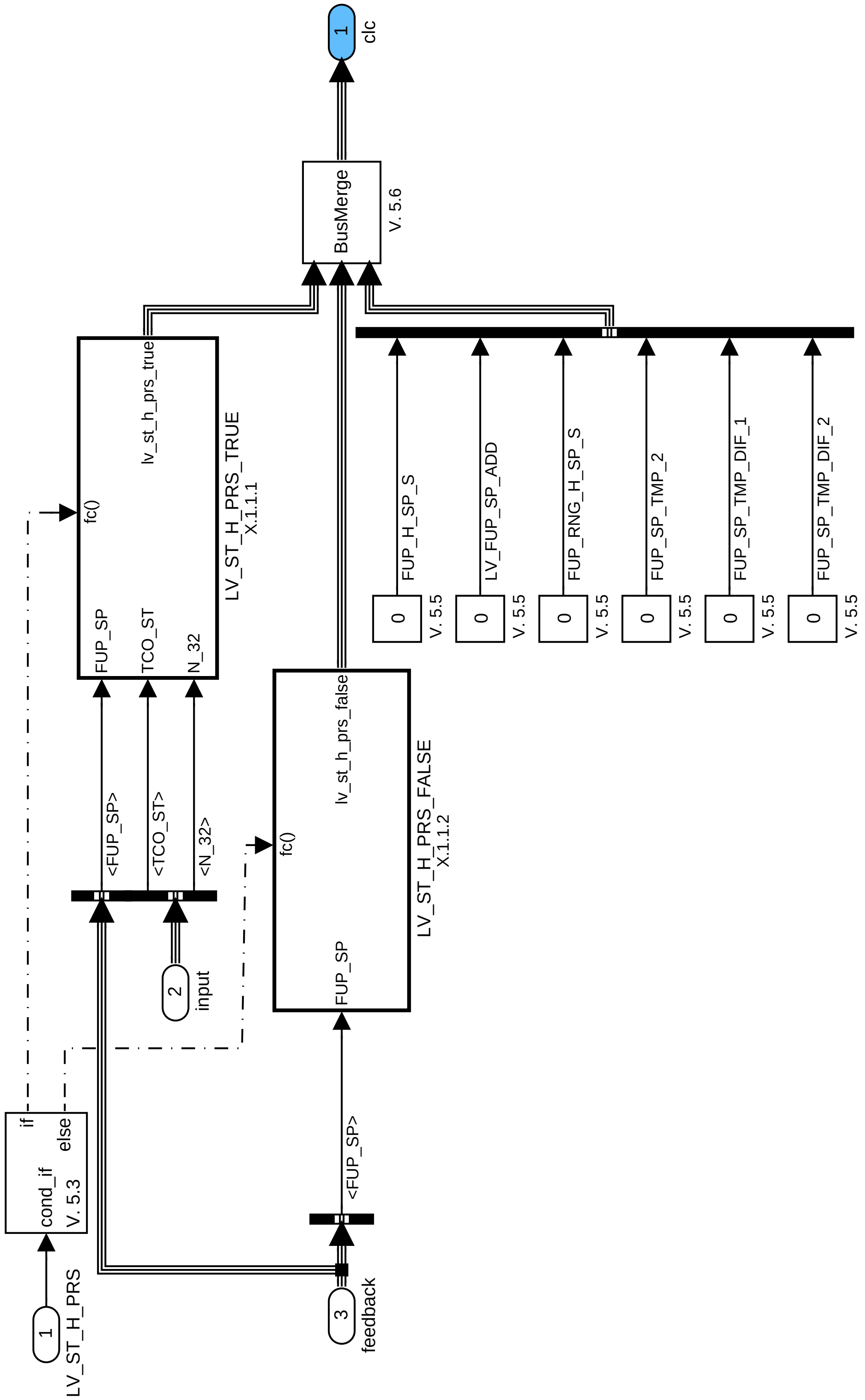


Figure 98:

Path: FUSL_M904A/INI/CLC

34.11.1.1.1 Initialisation if LV_ST_H_PRS is false

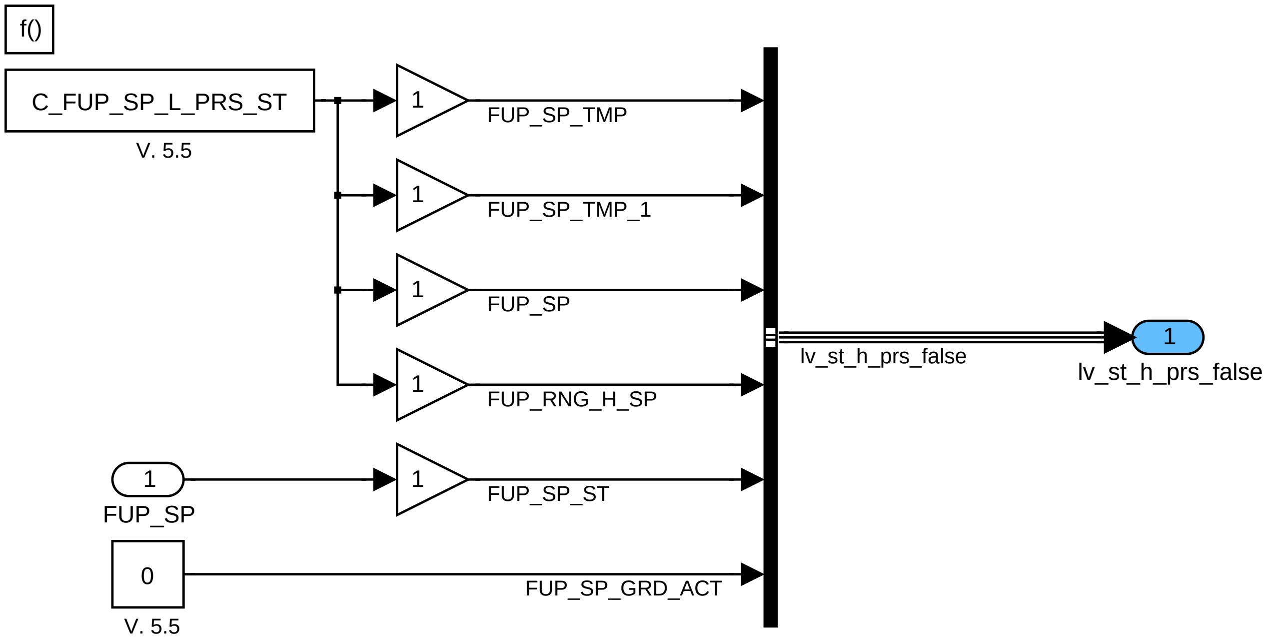


Figure 99:

Path: FUSL_M904A/INI/CLC/LV_ST_H_PRS_FALSE

34.11.1.1.2 Initialisation if LV_ST_H_PRS is true

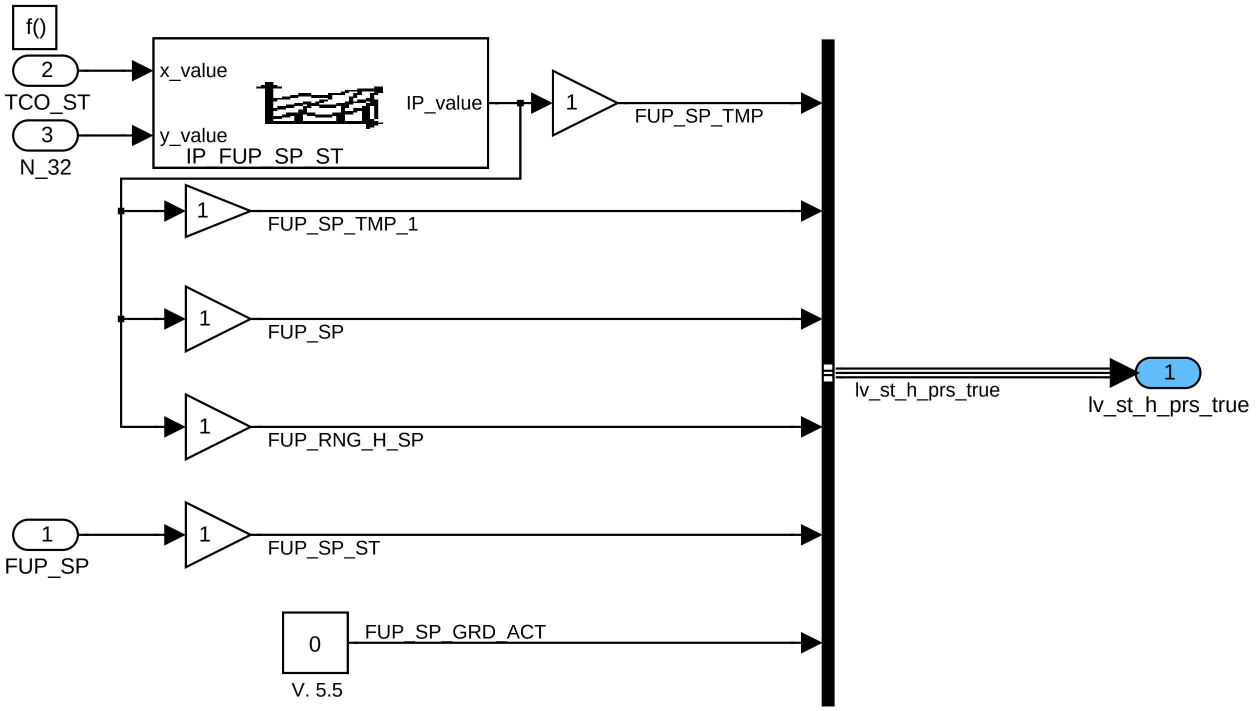


Figure 100:

Path: FUSL_M904A/INI/CLC/LV_ST_H_PRS_TRUE

34.11.2 Formula section

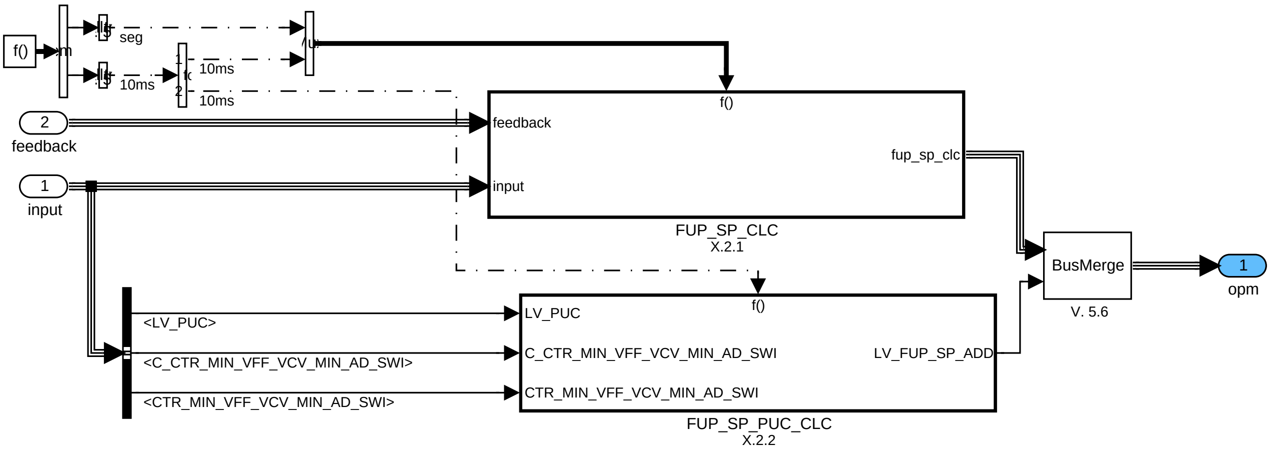


Figure 101:
Path: FUSL_M904A/OPM

34.11.2.1 Fuel pressure setpoint calculation overview

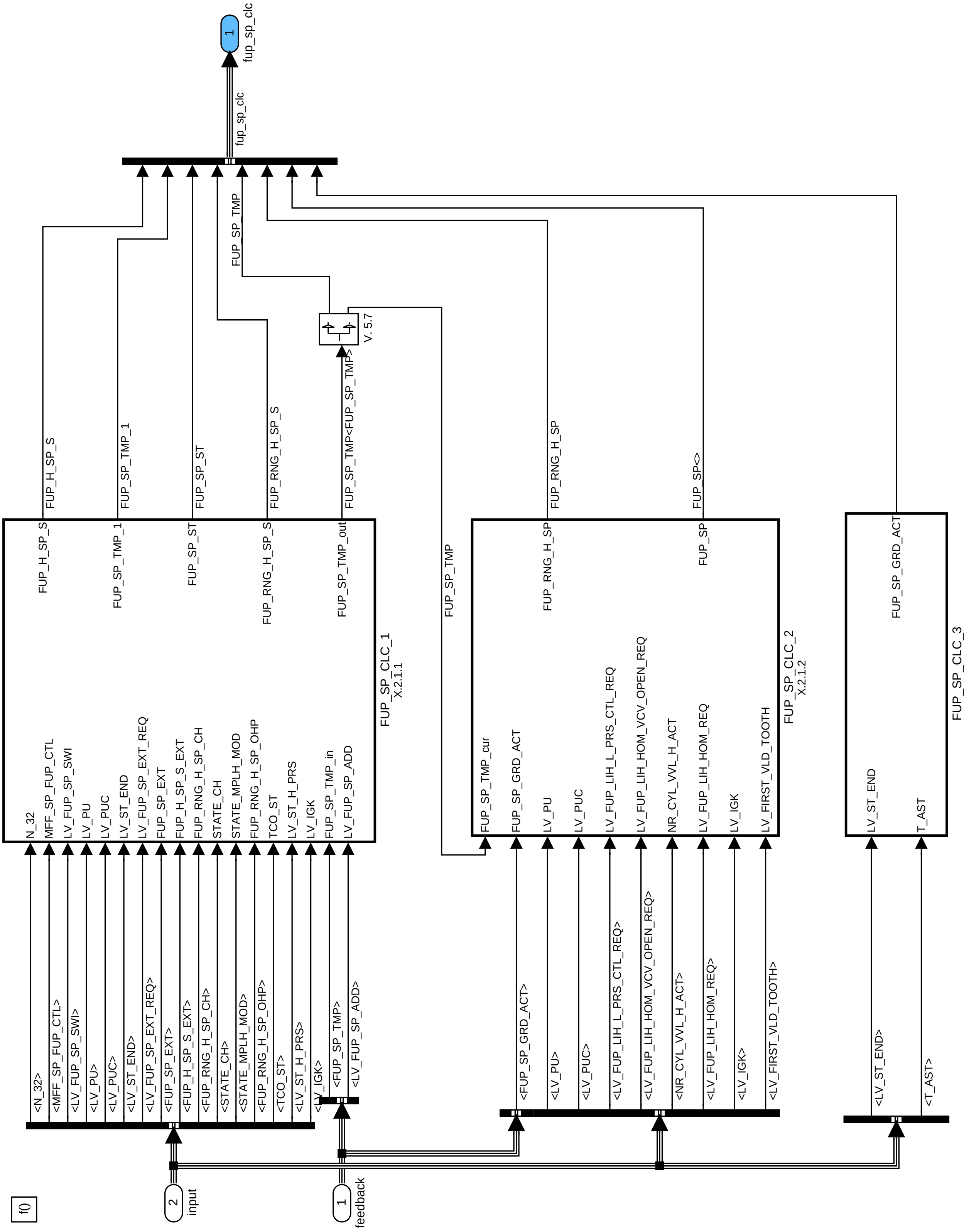


Figure 102:
Path: FUSL_M904A/OPM/FUP_SP_CLC

34.11.2.1.1 Calculation of fuel pressure high setpoint

The actual fuel pressure setpoint is taken out the two maps. Only for PU and PUC mode different fuel pressure setpoints can be selected due to injector requirement.

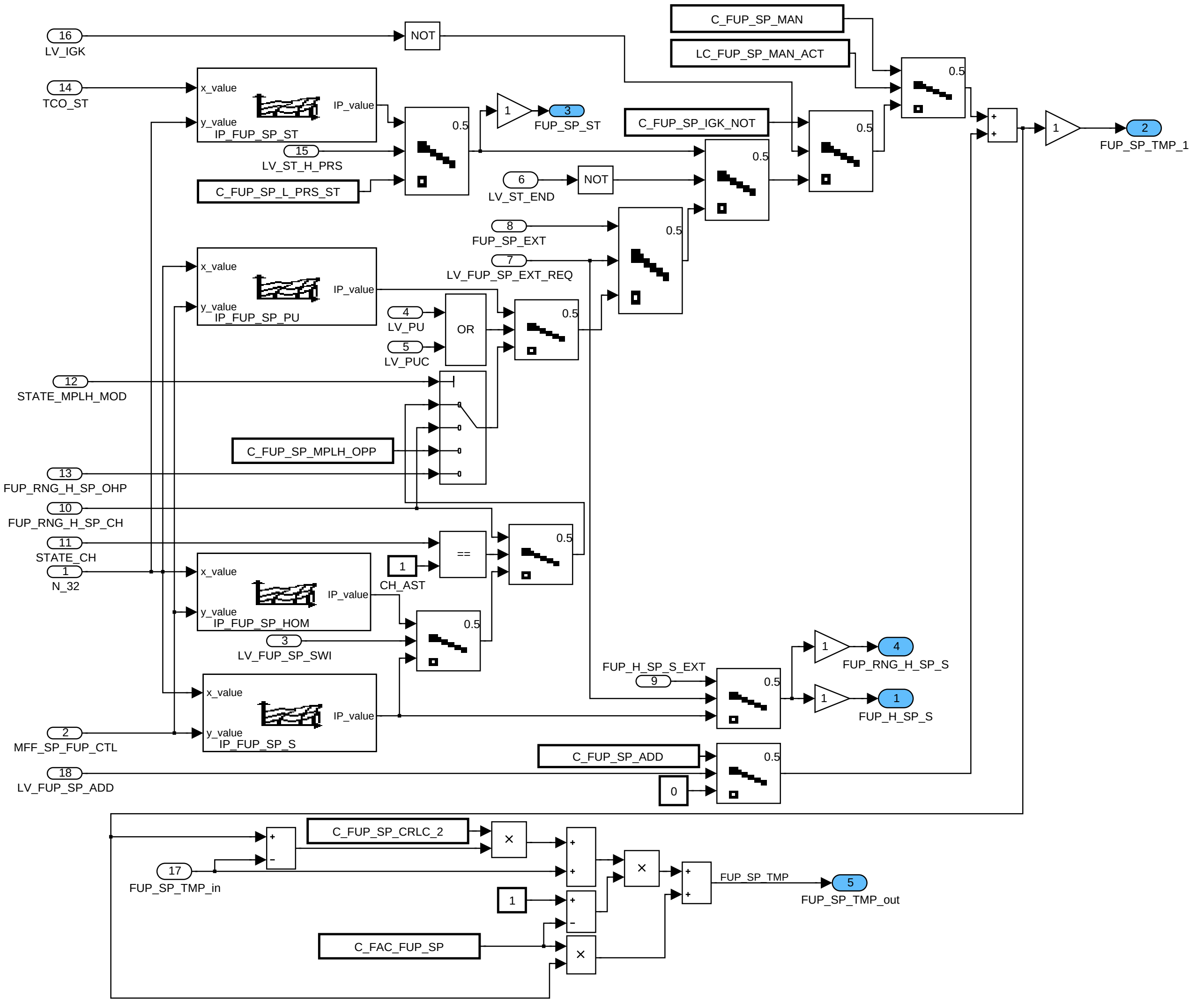


Figure 103:
Path: FUSL_M904A/OPM/FUP_SP_CLC/FUP_SP_CLC_1

34.11.2.1.2 Calculation of fuel pressure setpoint

For the gradient limitation the calibration should be started with the highest value for all gradient limitations. The values should be lowered by applying jumps the FUP_SP via manual intervention.

The internal value in the difference limiter has to be initialised with FUP_SP_TMP at reset.

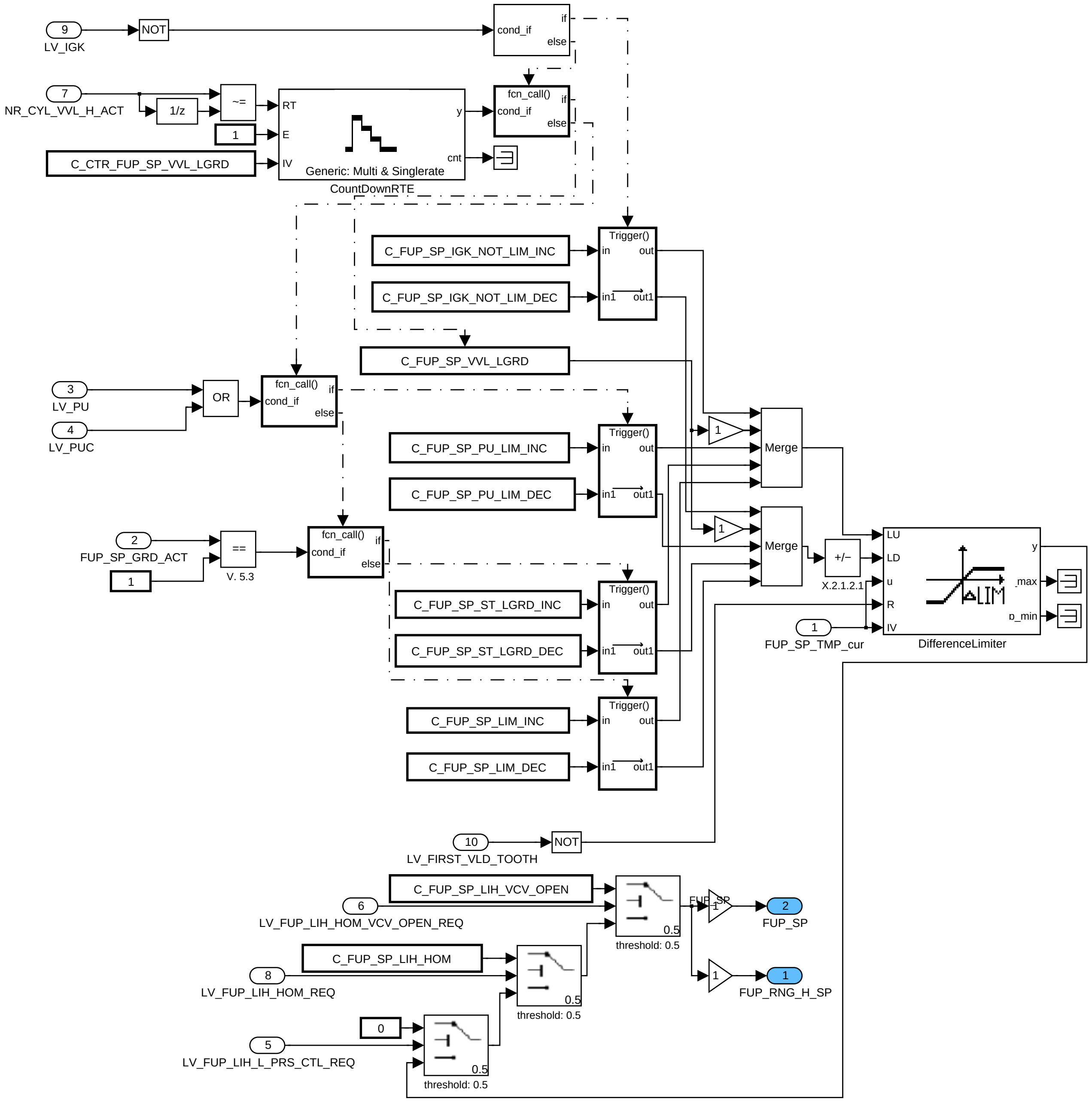


Figure 104:

Path: FUSL_M904A/OPM/FUP_SP_CLC/FUP_SP_CLC_2

34.11.2.1.3 Fuel pressure setpoint gradient calculation

The gradient limitation out the start phase will be stopped until the FUP_SP is reached.

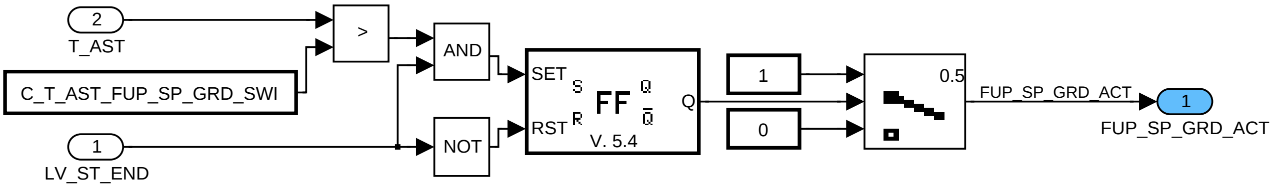


Figure 105:

Path: FUSL_M904A/OPM/FUP_SP_CLC/FUP_SP_CLC_3

34.11.2.2 Calculation of LV_FUP_SP_AD

This block is triggered every time at 10ms without the LV_ST_END == 0 condition.

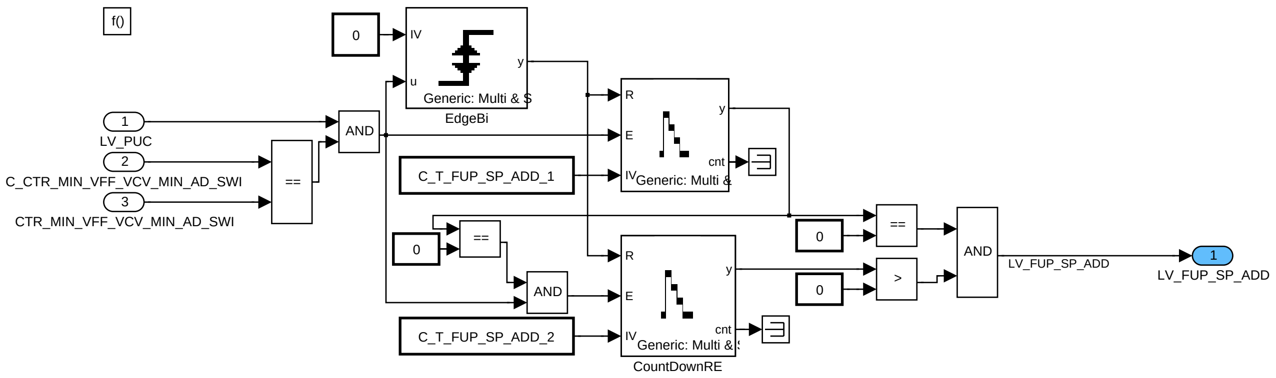


Figure 106:
Path: FUSL_M904A/OPM/FUP_SP_PUC_CLC

34.12 Fuel Supply general

Overview

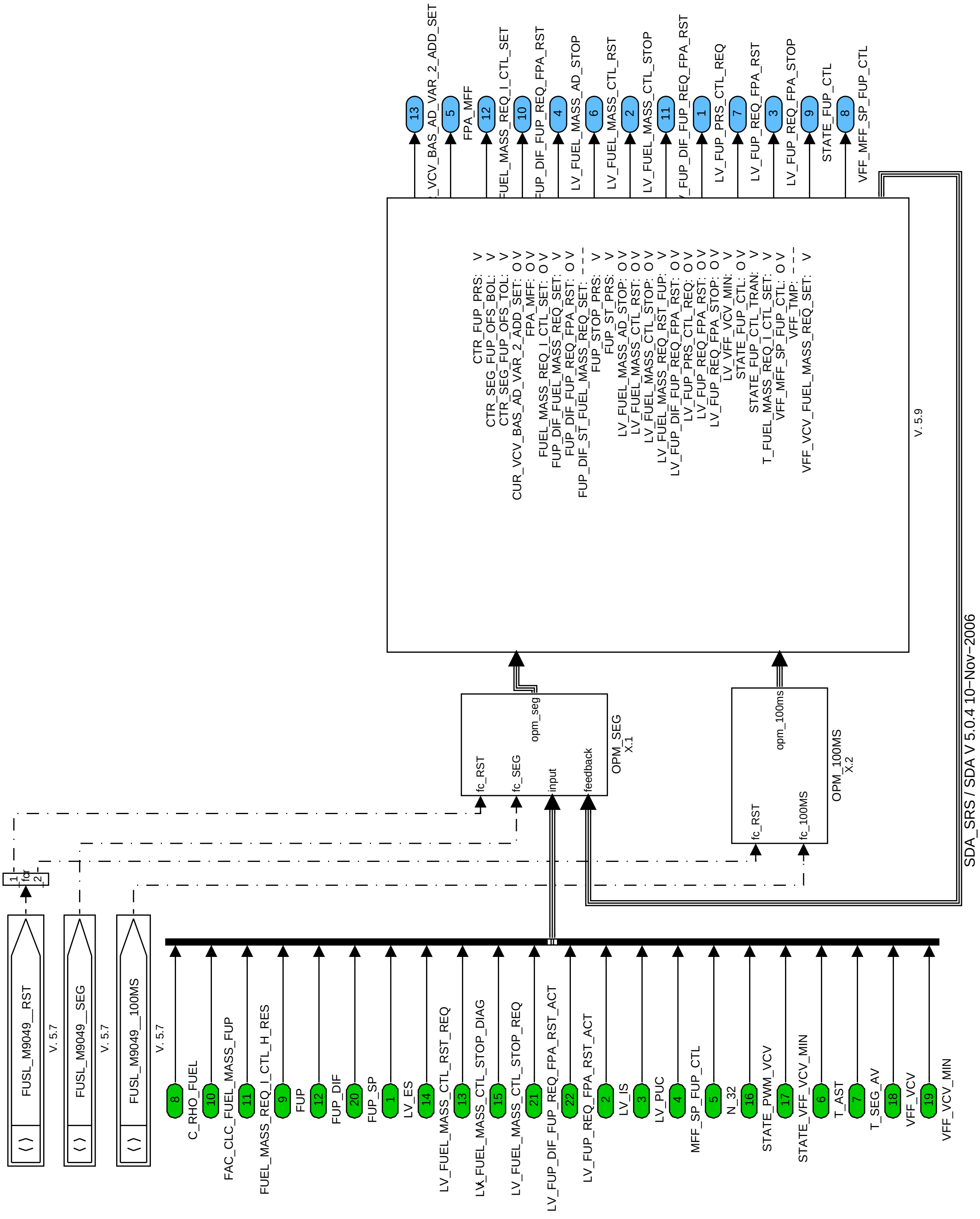


Figure 107:
Path: FUSL_M9049

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_FUP_PRS	V	0... FFFFH	0... 65535	1	[-]
Counter for maximum segment cycles for setting of the pressure controller in the intervention mode					
CTR_SEG_FUP_OFS_BOL	V	0... FFH	0... 255	1	[-]
Counter for forced leaving of pressure control					
CTR_SEG_FUP_OFS_TOL	V	0... FFH	0... 255	1	[-]
Counter for forced leaving of flow control					
CUR_VCV_BAS_AD_VAR_2_ADD_SET	O/V	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Setting value due to intervention mode for additive component of second variant of VCV flow adaptation					
FPA_MFF	O/V	0... FFFFH	0... 255	3.89105e-3	[l/h]
Fuel flow through the regulator					
FUEL_MASS_REQ_I_CTL_SET	O/V	80000000... 7FFFFFFFH	-694.510597391 ...694.489402609	323.402e-9	[mg]
Setting value of the I-part of the flow controller in interwention mode					
FUP_DIF_FUEL_MASS_REQ_SET	V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Deviation of fuel pressure for setting of the I-part of the flow controller					
FUP_DIF_FUP_REQ_FPA_RST	O/V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Fuel pressure difference at switching from pressure to flow control in intervention mode					
FUP_DIF_ST_FUEL_MASS_REQ_SET	-	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Deviation of fuel pressure for setting of the I-part of the flow controller					
FUP_ST_PRS	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure in the beginning of the pressure control phase for the intervention mode					
FUP_STOP_PRS	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure in the end of the pressure control phase for the intervention mode					
LV_FUEL_MASS_AD_STOP	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting adaption stop of controller for demand control					
LV_FUEL_MASS_CTL_RST	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting I-part reset of controller for flow control					
LV_FUEL_MASS_CTL_STOP	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting I-part freeze of controller for flow control					
LV_FUEL_MASS_REQ_RST_FUP	V	0... 1H	0... 1	1	[-]
Logical variable indicating resetting of the I-Part of the flow controller becouse of overpressure					
LV_FUP_DIF_FUP_REQ_FPA_RST	O/V	0... 1H	0... 1	1	[-]
Logical variable indicating the setting of the pressure I-part of the controller in intervention mode					
LV_FUP_PRS_CTL_REQ	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting pressure control mode					
LV_FUP_REQ_FPA_RST	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting I-part reset of controller for pressure control					
LV_FUP_REQ_FPA_STOP	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting I-part freeze of controller for pressure control					
LV_VFF_VCV_MIN	V	0... 1H	0... 1	1	[-]
Logical variable indicating volume flow through VCV reaches limit					
STATE_FUP_CTL	O/V	0 1 2 3	FLOW_CONTROL FLOW2PRS PRESSURE_CONTROL PRS2FLOW	-	[-]
Operating state for the global fuel pressure regulation strategy					

STATE_FUP_CTL_TRAN	V	0 1 2 3	FLOW_REQ FLOW2PRS_REQ PRS_REQ PRS2FLOW_REQ	-	[-]
Transistion conditions for the STATE_FUP_CTL					
T_FUEL_MASS_REQ_I_CTL_SET	V	0... FFFFH	0... 655.35	0.01	[s]
Maximum time in the flow control mode for setting of the I-part of the flow controller					
VFF_MFF_SP_FUP_CTL	O/V	0... FFFFH	0... 255	3.89105e-3	[l/h]
Volume fuel flow through the injectors					
VFF_TMP	-	0... FFFFH	0... 255	3.89105e-3	[l/h]
Temporary value for volume fuel flow					
VFF_VCV_FUEL_MASS_REQ_SET	V	0... FFFFH	0... 255	3.89105e-3	[l/h]
Setting volume fuel flow for the I-part of the flow controller					

Input Data:

LV_ES	LV_IS	LV_PUC	MFF_SP_FUP_CTL
N_32	T_AST	T_SEG_AV	C_RHO_FUEL
FUP	FAC_CLC_FUEL_MASS_FUP	FUEL_MASS_REQ_I_CTL_H_RES	FUP_DIF
LV_FUEL_MASS_CTL_STOP_DIAG	LV_FUEL_MASS_CTL_RST_REQ	LV_FUEL_MASS_CTL_STOP_REQ	STATE_PWM_VCV
STATE_VFF_VCV_MIN	VFF_VCV	VFF_VCV_MIN	FUP_SP
LV_FUP_DIF_FUP_REQ_FPA_RST_ACT	LV_FUP_REQ_FPA_RST_ACT		

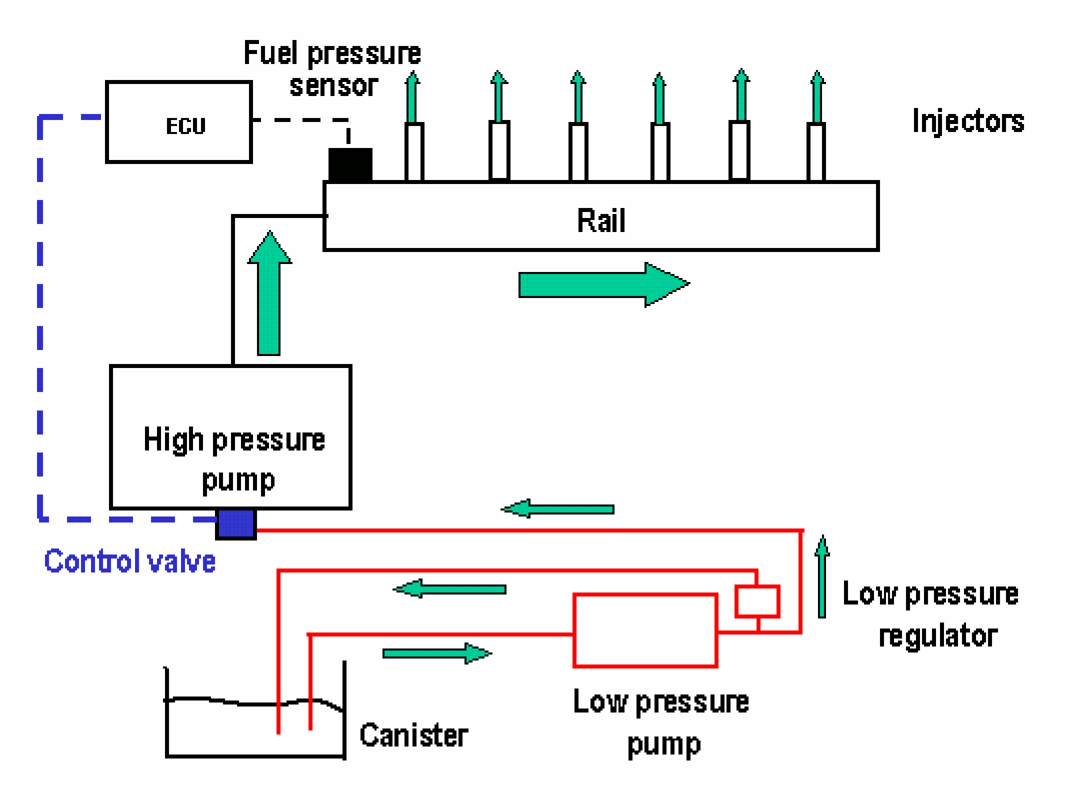
Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CTR_FUP_PRS	1	0... FFFFH	0... 65535	1	[-]
Constant for maximal segment cycle number for setting of the pressure controller in the intervention mode					
C_CTR_SEG_FUP_OFS_BOL_MAX	1	0... FFH	0... 255	1	[-]
Max. value for the counter for forced leaving of pressure control					
C_CTR_SEG_FUP_OFS_TOL_MAX	1	0... FFH	0... 255	1	[-]
Max. value for the counter for forced leaving of flow control					
C_CUR_VCV_BAS_AD_VAR_2_ADD_SET	1	0... FFFFH	-32.768 ...32.767	1e-3	[A]
Setting value for the adaptation at setting of the flow controller					
C_FUP_CTL_MOD	1	0 1 2	AUTOMATIC FLOW_CONTROL PRESSURE_CONTROL	-	[-]
Request for operating state of the global fuel pressure regulation strategy					
C_FUP_OFS_BOL	1	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Offset to the FUP for the forced leaving of pressure control					
C_FUP_OFS_TOL	1	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Offset to the FUP for the forced leaving of flow control					
C_FUP_SP_DEC_TOL	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Offset to the FUP for the forced leaving of flow control					
C_T_AST_MIN_I_CTL_SET	1	0... FFFFH	0... 6553.5	0.1	[s]
Minimum time after start for enabling the setting of the flow controller and adaptation value					
C_T_FUEL_MASS_REQ_I_CTL_SET	1	0... FFFFH	0... 655.35	0.01	[s]
Constant for the maximum time in the flow control mode for setting of the I-part of the flow controller					
C_VFF_MFF_SP_VCV_BAS_AD_SET_LIM	1	0... FFFFH	0... 255	3.89105e-3	[l/h]

Limit VFF_MFF_SP_FUP_CTL value for setting of the I-part of the flow controller					
C_VFF_SWI_THD	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
Switch threshold selecting VFF_VCV or VFF_MFF_SP_FUP_CTL as input for struc. change					
C_VFF_VCV_BAS_AD_SET_LIM	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
Limit VFF_VCV value for setting of the I-part of the flow controller					
C_VFF_VCV_MIN_BOL_HYS	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
Hysteresis to the lower switch limit					
C_VFF_VCV_MIN_TOL_HYS	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
Hysteresis to the upper switch limit					
IP_FAC_FUEL_MASS_REQ_SET_N	8	0... FFFFH	0... 31.999	488.273e-6	[mg/(l/h)]
LDP_IP_FAC_FUEL_MASS_SET	8	0... FFH	0... 8160	32	[rpm]
Calculation of the mg value for setting of the I-part of the flow controller due to engine speed					
IP_FUP_DIF_FUP_REQ_FPA_RST	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
LDP_FUP_DIF_IP_FUP_REQ_FPA_RST	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
Interpolation curve for setting of the pressure controller in the intervention mode					
IP_VFF_VCV_I_CTL_SET	6	0... FFFFH	-255 ...0	3.89105e-3	[l/h]
LDP_VFF_VCV_IP_VFF_VCV_CTL_SET	6	0... FFFFH	0... 255	3.89105e-3	[l/h]
Calculation map for the setting value of the I-part of the flow controller due to fuel volume flow					
LC_FUEL_MASS_CTL_RST	1	0... 1H	0... 1	1	[-]
Logical switch requesting I-part reset of controller for flow control at leaving pressure control					
LC_FUP_REQ_FPA_RST	1	0... 1H	0... 1	1	[-]
Logical switch requesting I-part reset of controller for pressure control at leaving flow control					
LC_INH_SWI_THD_PUC	1	0... 1H	0... 1	1	[-]
Logical constant for inhibit of the fuel pressure intervention mode in PUC					
LC_VFF_SWI	1	0... 1H	0... 1	1	[-]
Logical switch selecting VFF_VCV or VFF_MFF_SP_FUP_CTL as input for struc. Change (1:VFF_VCV)					

General Information

A variant of the fuel pressure control system is a threewayvalve as a control mode, which combines the volume control valve and the regulator functionality. In that case the regulator is obsolete.



General scheduling

Segment synchronious tasks:

1. Fuel pressure of low pressure pump
2. Fuel pressure
3. {Injection related parts of SW}
4. Electrical fuel pump control (Appl. Inc.)
5. Electrical fuel pump
6. Fuel supply general
7. Fuel pressure setpoint calculation
8. Fuel pressure control
9. High pressure pump control

1 ms tasks:

1. Fuel pressure of low pressure pump

10 ms tasks:

1. Fuel pressure of low pressure pump
2. Fuel pressure
3. {Injection related parts of SW}
4. Electrical fuel pump control (Appl. Inc.)
5. Electrical fuel pump
6. Fuel pressure setpoint calculation
7. Fuel pressure control (Appl. Inc.)
8. Fuel pressure control
9. High pressure pump control (Appl. Inc.)

10. High pressure pump control

20 ms tasks:

1. Fuel system pressure diagnosis for low pressure (Appl. Inc)
2. Fuel system pressure diagnosis for low pressure
3. Fuel system pressure diagnosis (Appl. Inc.)
4. Fuel system pressure diagnosis

40 ms tasks:

1. Fuel pressure sensor diagnosis for low pressure (Appl. Inc.)
2. Fuel pressure sensor diagnosis for low pressure
3. Fuel pressure sensor diagnosis (Appl. Inc.)
4. Fuel pressure sensor diagnosis

100 ms tasks:

1. Electrical diagnosis of fuel pressure actuator (Appl. Inc.)
2. Electrical diagnosis of fuel pressure actuator
3. Fuel pressure setpoint of electrical fuel pump
4. Fuel pressure control (Appl. Inc.)
5. Fuel pressure control
6. High pressure pump control
7. Fuel supply general

1000 ms tasks:
Fuel pressure actuator temperature calculation

Management of the pressure and flow controlers in the intervention mode

Condition VFF_VCV under the C_VFF_SWI_THD

In idle speed:

Reset of the I part of the flow controller
Set of the I part of the pressure controller
No reset of the I part of the pressure controller
No Execution of the adaptations

In PUC:

No reset of the I part of the flow controller
Set of the I part of the pressure controller
Reset of the I part of the pressure controller depending on the reset LC's
Alternative execution of the pressure /leakage adaptations

In PU:

No reset of the I part of the flow controller
Set of the I part of the pressure controller

Reset of the I part of the pressure controller depending on the reset LC's
No Execution of the adaptations.

34.12.1 Function description of 100MS task

Application Conditions

Initialization: RST
Recurrence: 100MS
Activation: always
Deactivation: never

Function description

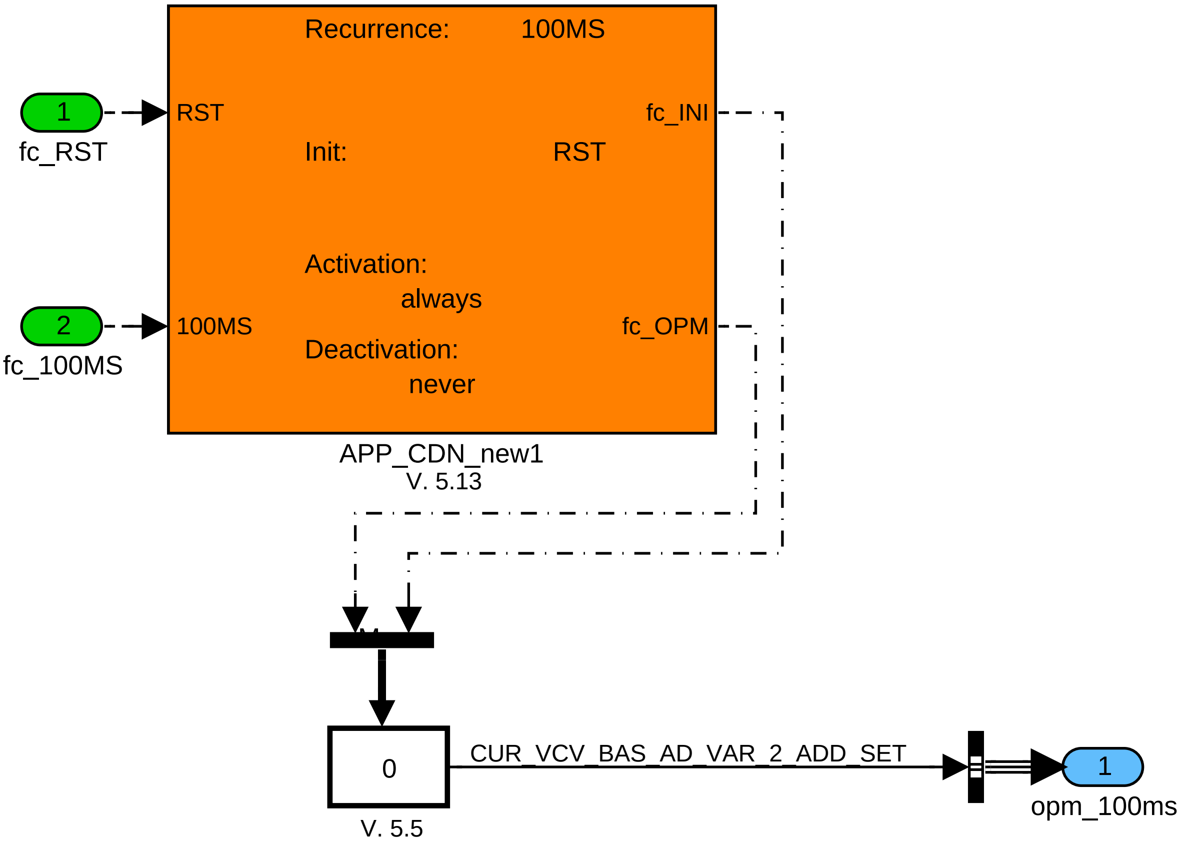
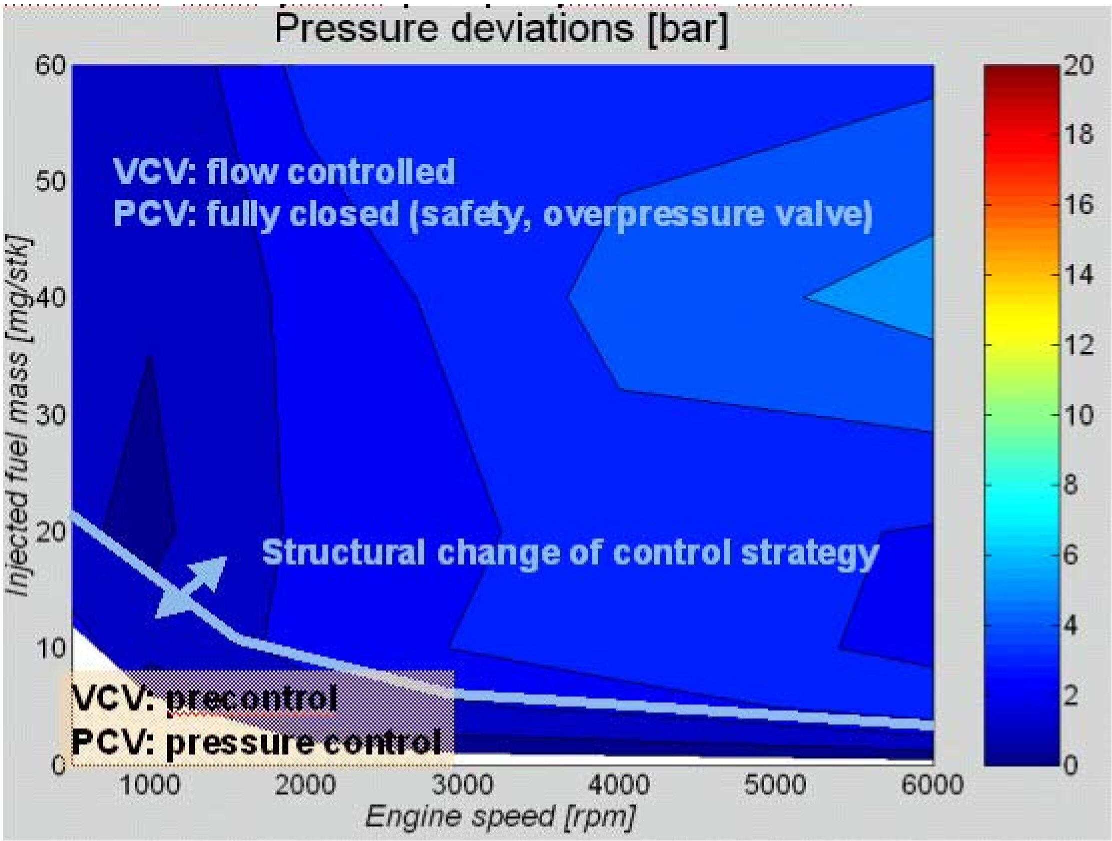


Figure 108:
Path: FUSL_M9049/OPM_100MS

34.12.2 Function Description of segment synchronous task

The fuel supply general organizes the different fuel pressure control possiblities. It can be achieved an automatic mode where the flow control and pressure control mode is chosen by the flow through the volume control valve. Also only flow control or pressure control can be selected. The pressure control part can be divided into pressure control with or without flow precontrol and the flow control can be combined with or without pressure control.



Application Conditions

Initialization: RST
Recurrence: SEG
Activation: LV_ES == 0
Deactivation: LV_ES == 1

Function description

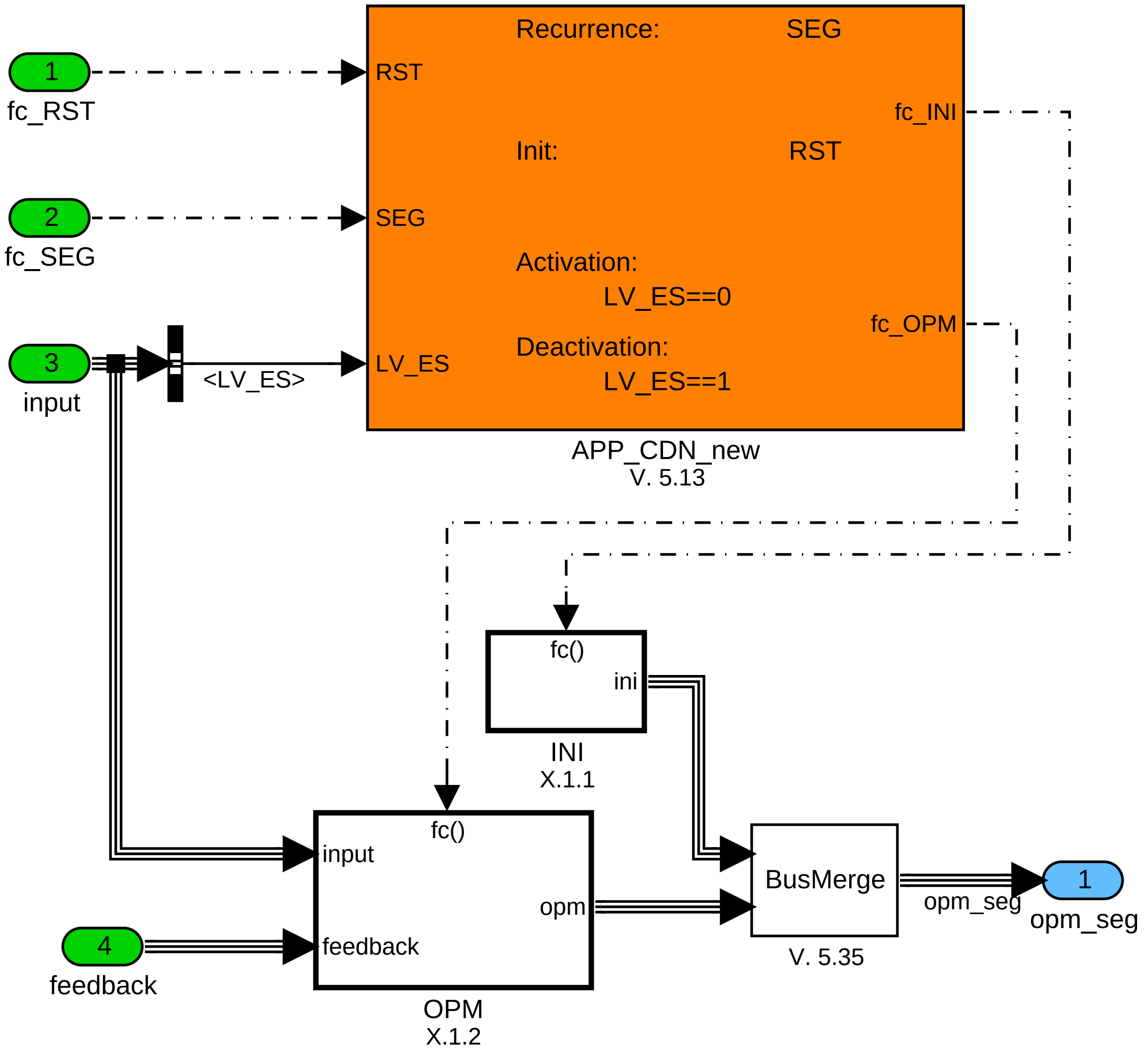


Figure 109:
Path: FUSL_M9049/OPM_SEG

34.12.2.1 Initialisation at reset event

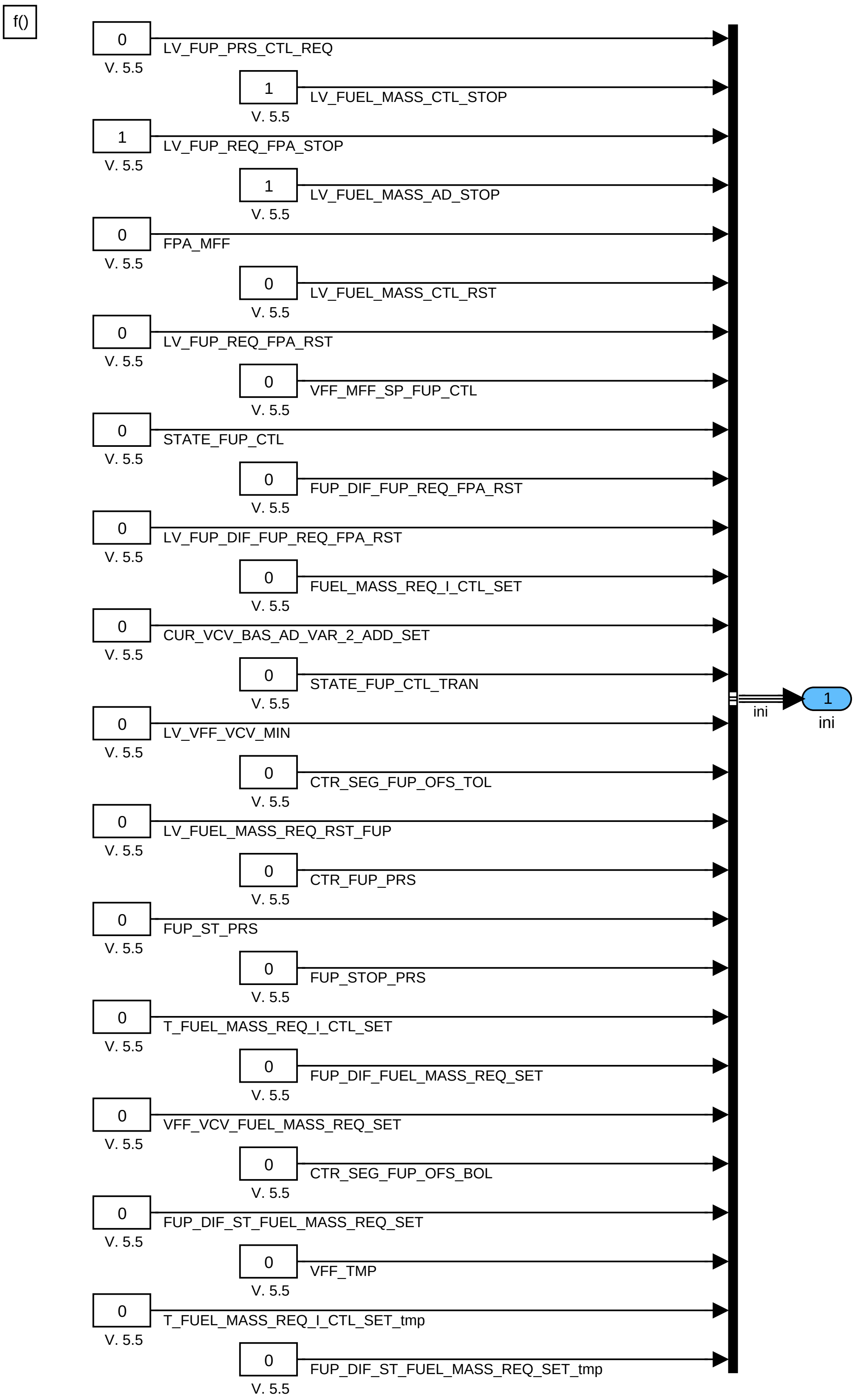


Figure 110:
Path: FUSL_M9049/OPM_SEG/INI

34.12.2.2 Calculation

For better visibility the functionality is split into three subsystems

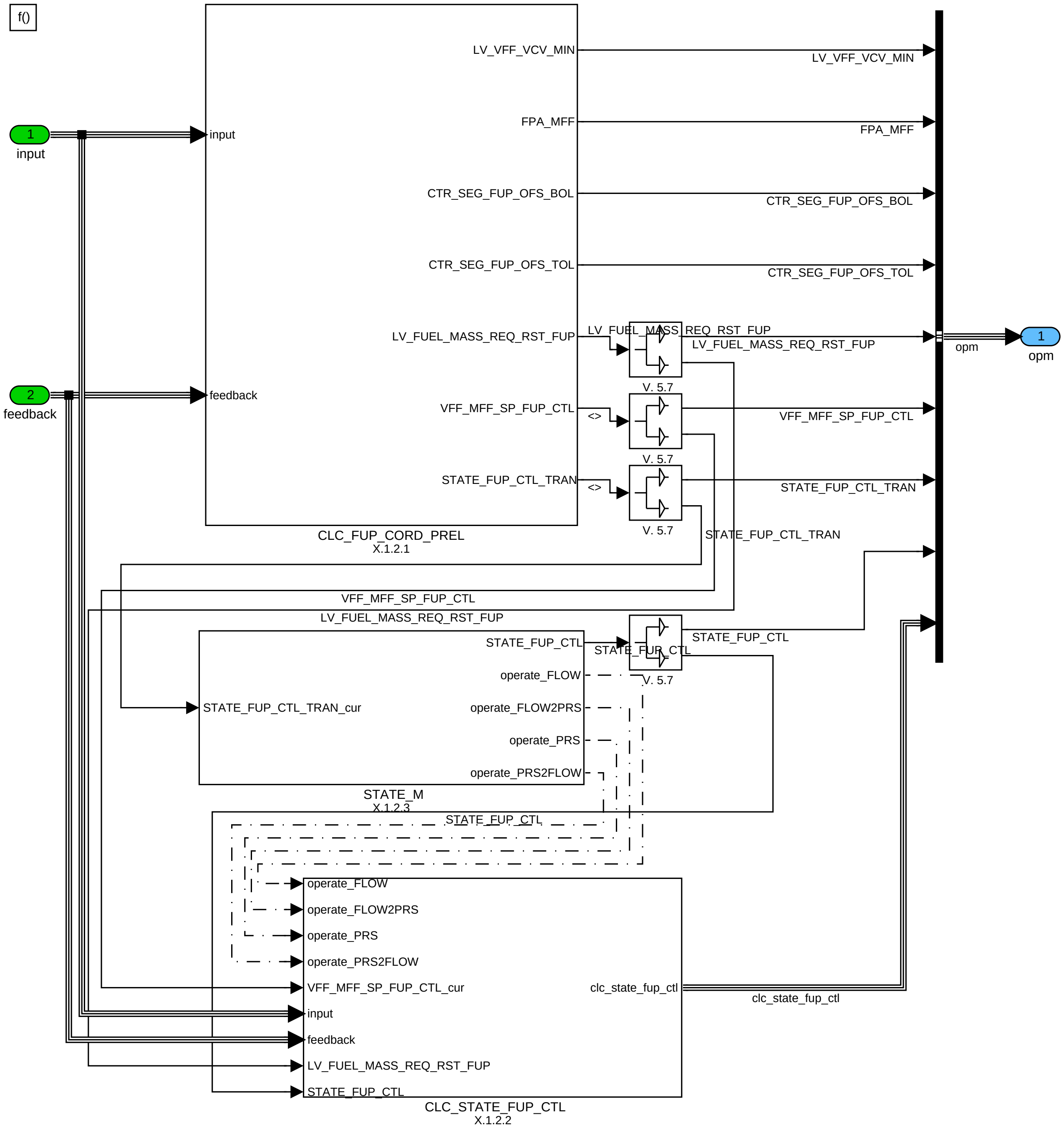


Figure 111:
Path: FUSL_M9049/OPM_SEG/OPM

34.12.2.2.1 Preliminary fuel pressure co-ordination

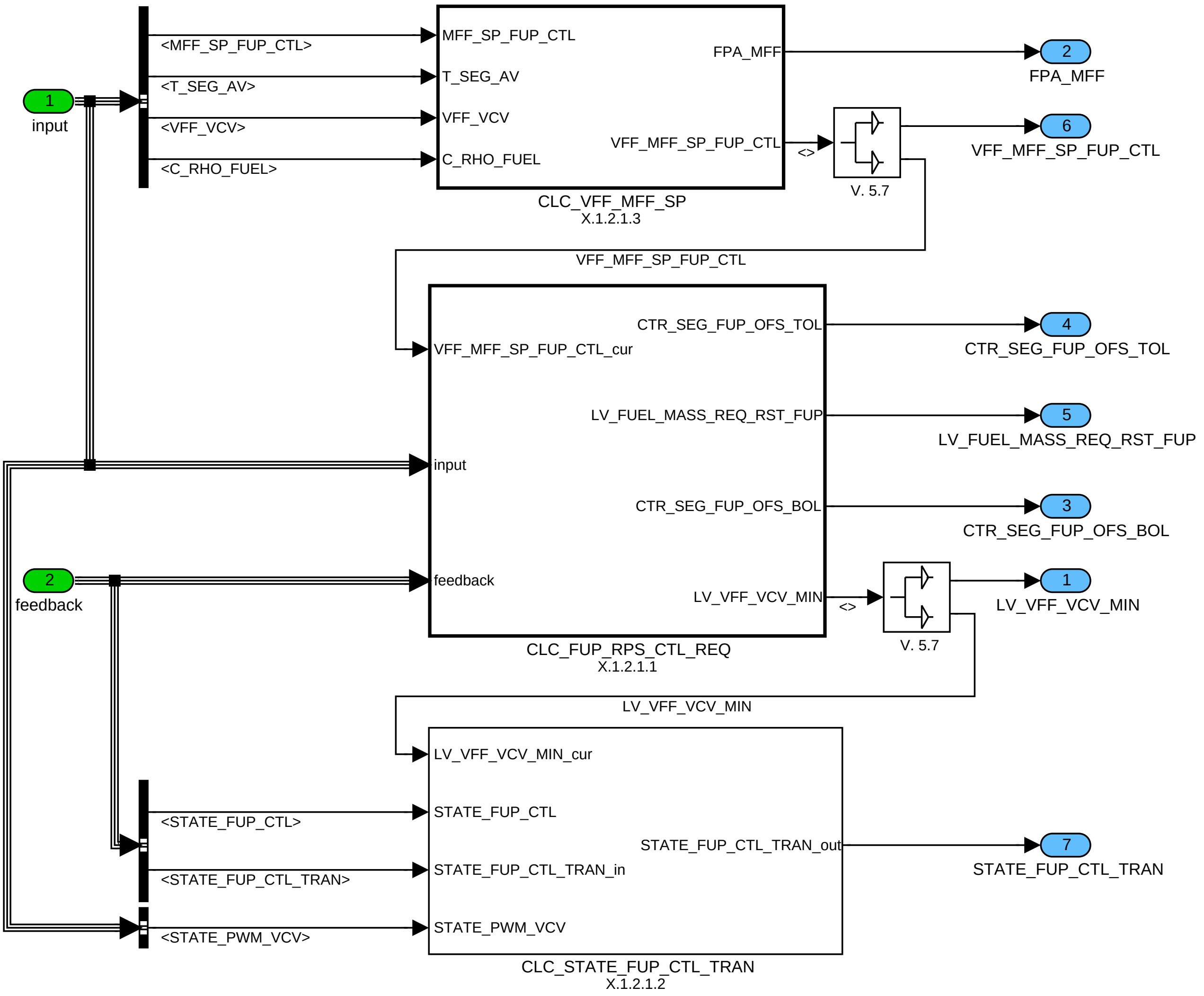


Figure 112:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_FUP_CORD_PREL

34.12.2.2.1.1 Fuel pressure request control

The following blocks calculate the switching between pressure and flow control mode

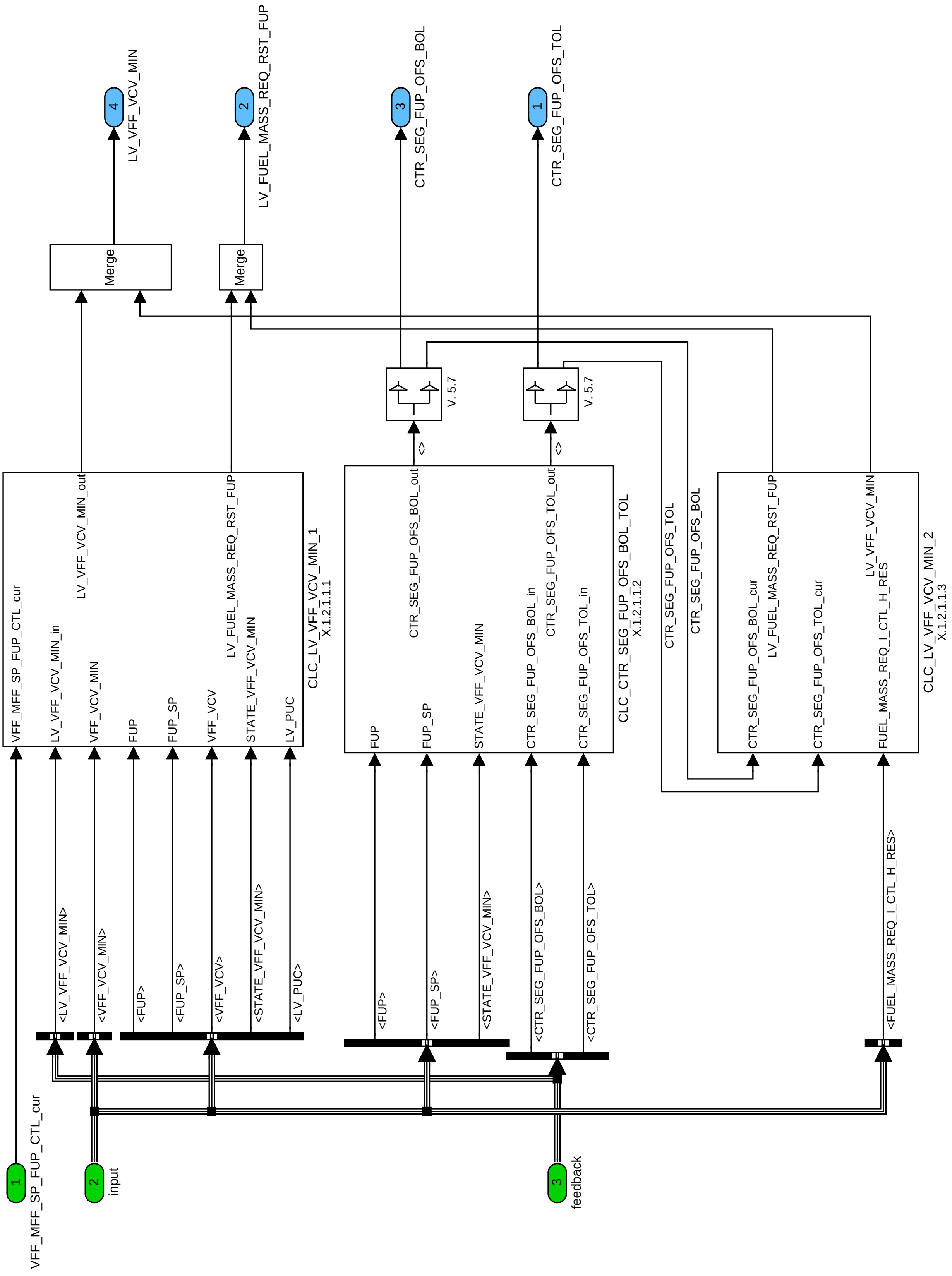


Figure 113:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_FUP_CORD_PREL/CLC_FUP_RPS_CTL_REQ

34.12.2.2.1.1.1 Calculation of fuel pressure limits

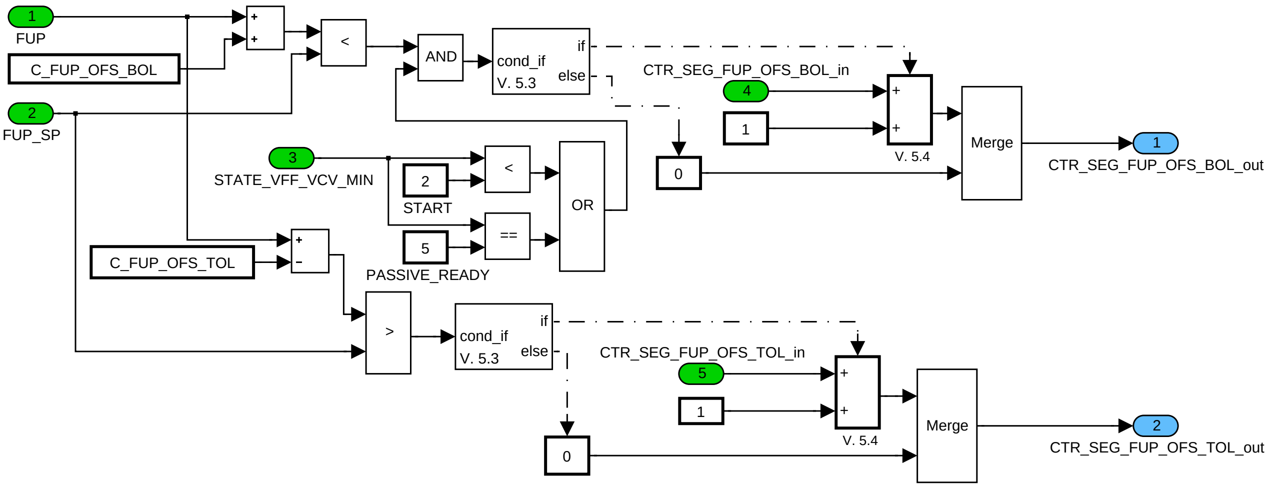


Figure 114:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_FUP_CORD_PREL/CLC_FUP_RPS_CTL_REQ/CLC_CTR_SEG_FUP_OFS_BOL_TOL

34.12.2.2.1.1.2 Minimum volumetric fuel flow calculation

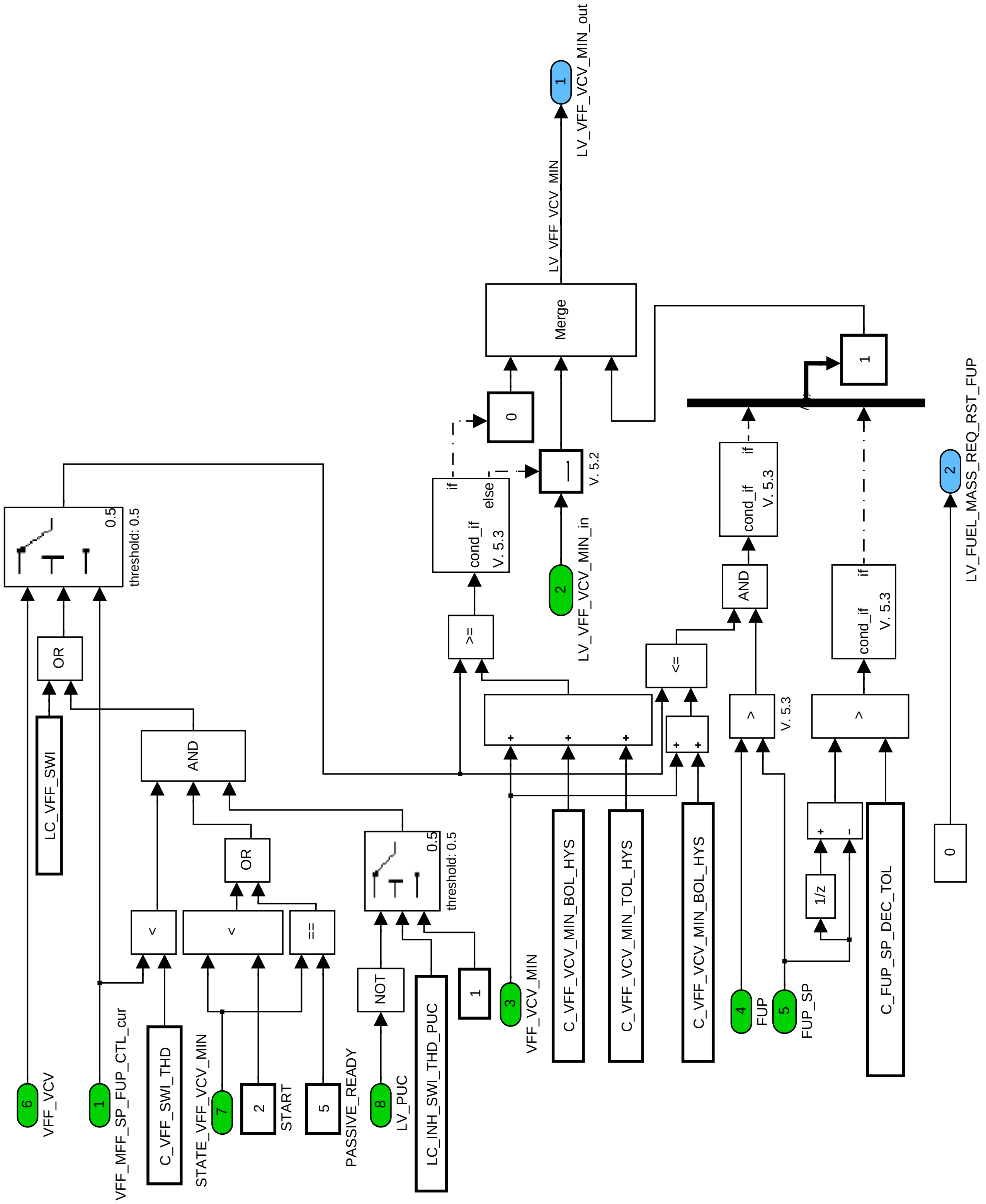


Figure 115:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_FUP_CORD_PREL/CLC_FUP_RPS_CTL_REQ/CLC_LV_VFF_VCV_MIN_1

34.12.2.2.1.1.3 Minimum fuel flow based on fuel pressure offset

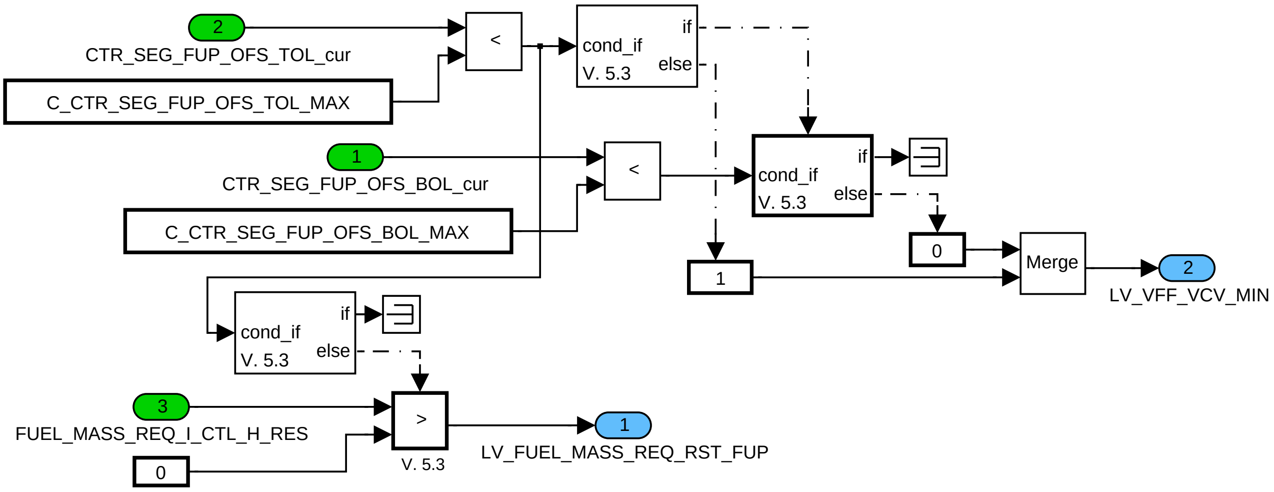


Figure 116:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_FUP_CORD_PREL/CLC_FUP_RPS_CTL_REQ/CLC_LV_VFF_VCV_MIN_2

34.12.2.2.1.2 Fuel pressure control transition

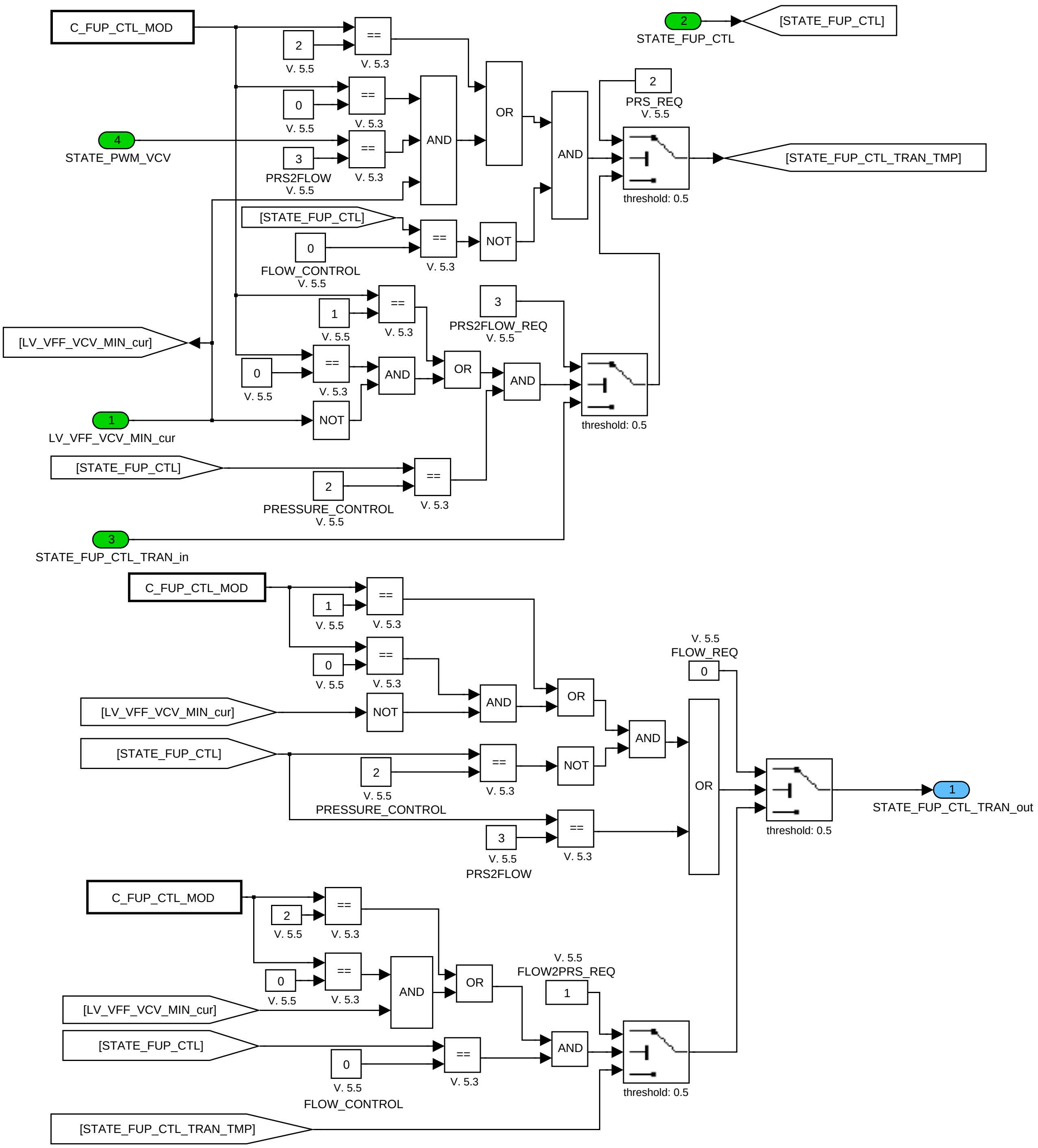


Figure 117:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_FUP_CORD_PREL/CLC_STATE_FUP_CTL_TRAN

34.12.2.2.1.3 Calculation of Fuel pressure control

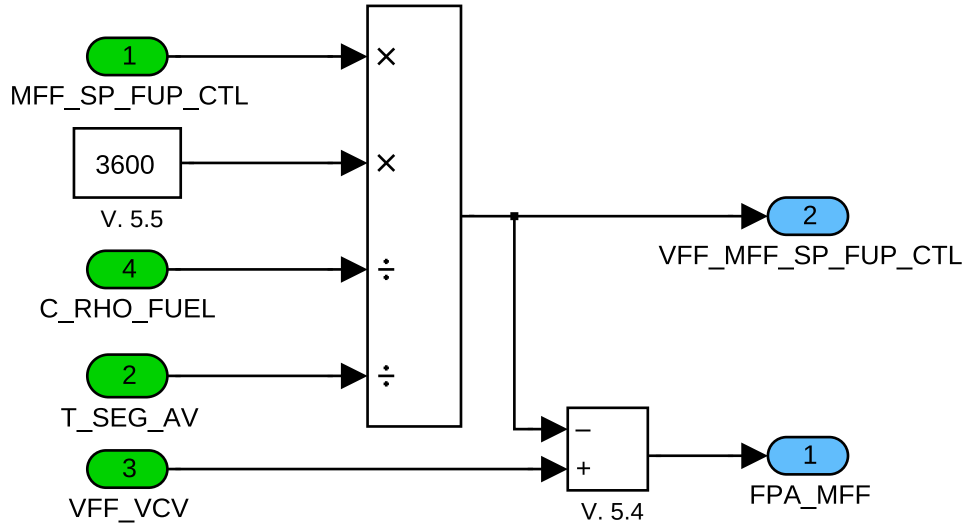


Figure 118:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_FUP_CORD_PREL/CLC_VFF_MFF_SP

34.12.2.2.2 State Machine

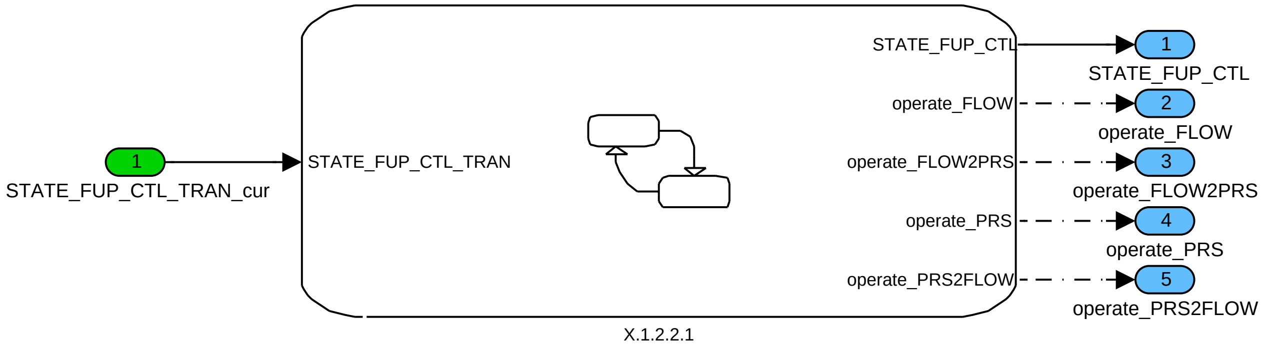


Figure 119:
Path: FUSL_M9049/OPM_SEG/OPM/STATE_M

34.12.2.2.2.1 Chart

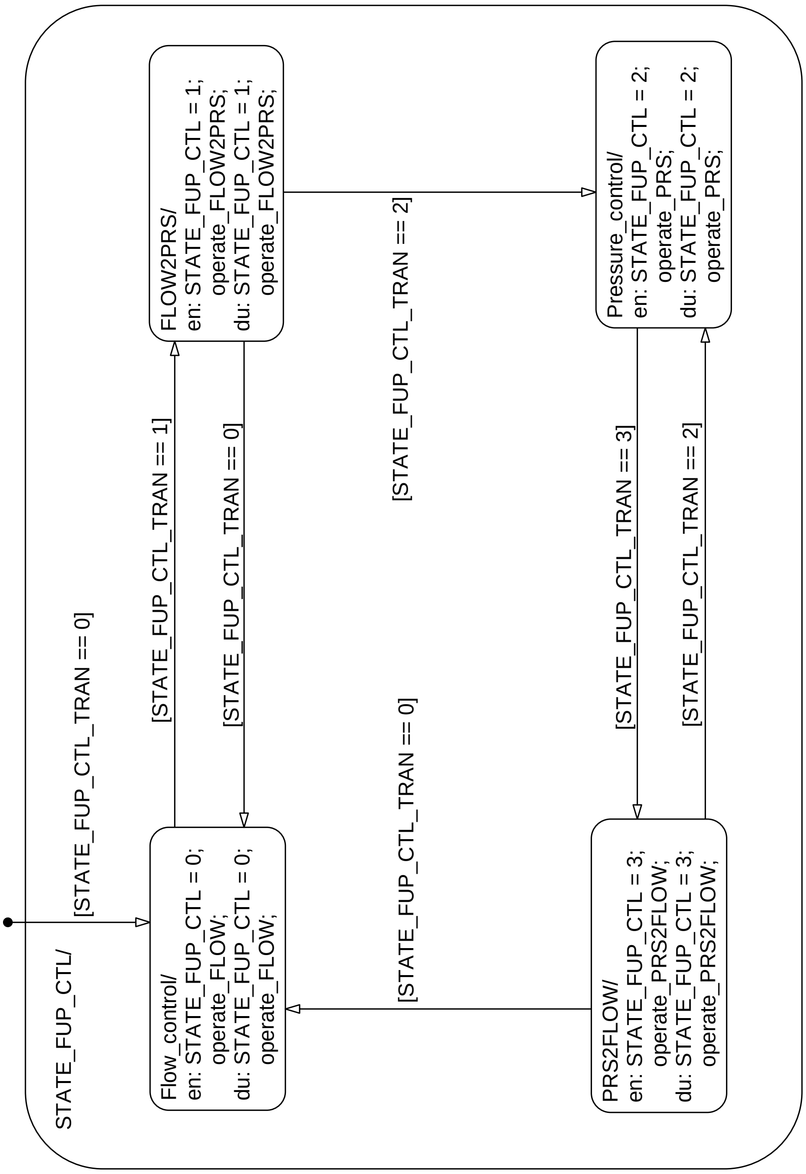


Figure 120:
Path: FUSL_M9049/OPM_SEG/OPM/STATE_M/Chart

34.12.2.2.3 State action calculation overview

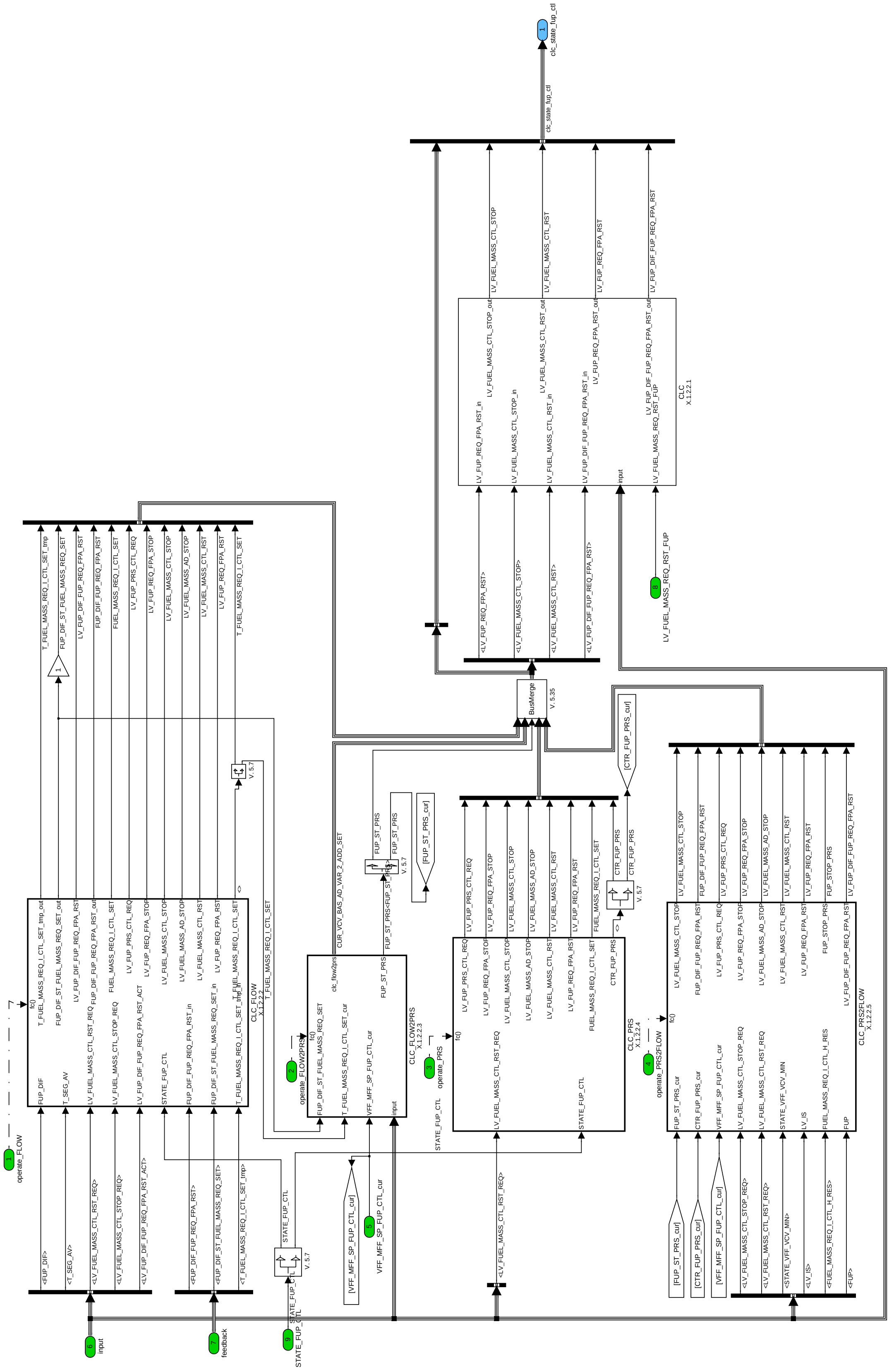


Figure 121:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_STATE_FUP_CTL

34.12.2.2.3.1 Calculation

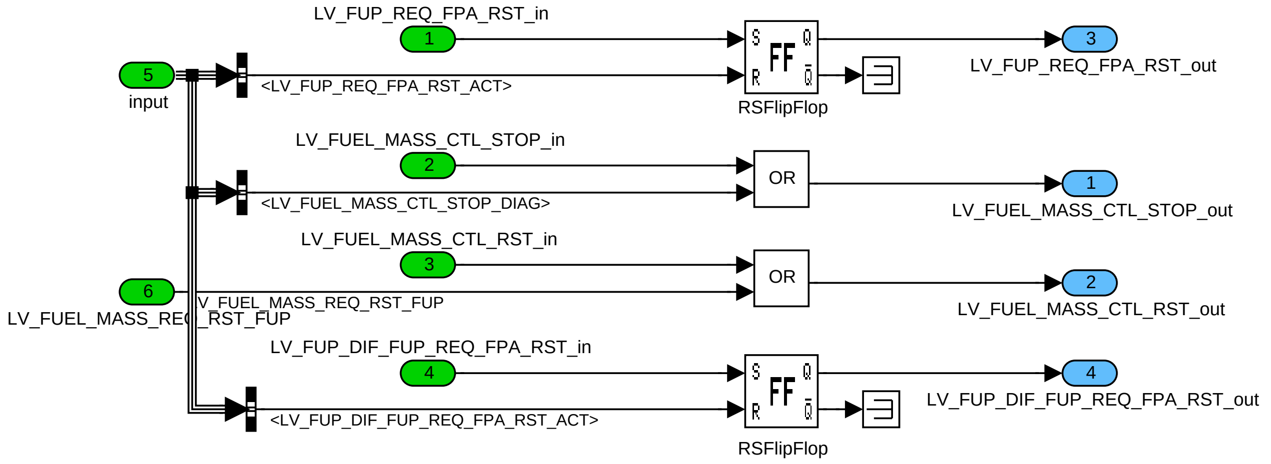


Figure 122:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_STATE_FUP_CTL/CLC

34.12.2.2.3.2 Flow calculation

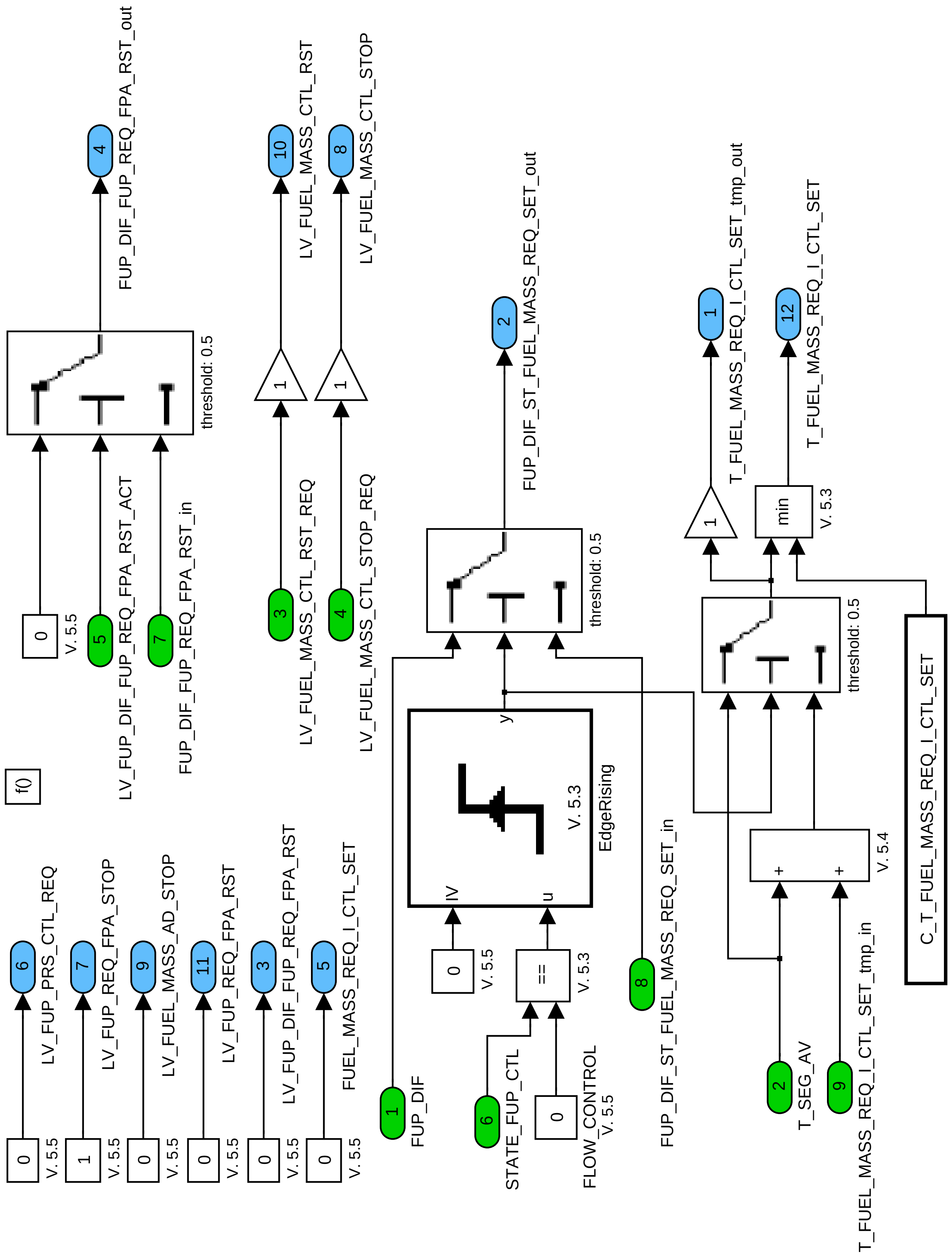


Figure 123:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_STATE_FUP_CTL/CLC_FLOW

34.12.2.2.3.3 Flow to pressure calculation overview

For better visibility the functionality is divided into two different systems

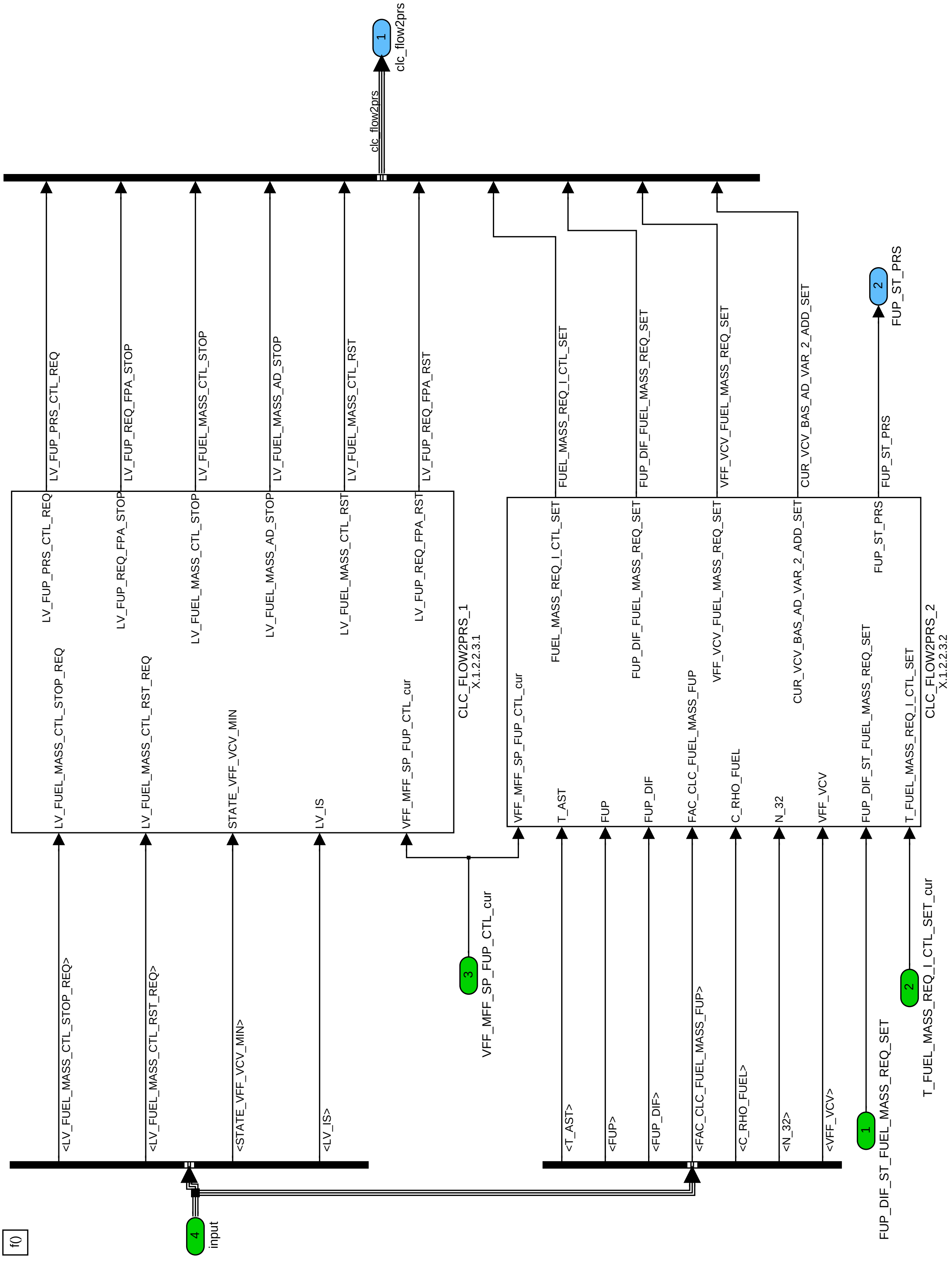


Figure 124:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_STATE_FUP_CTL/CLC_FLOW2PRS

34.12.2.2.3.3.1 Fuel pressure actuator reset calculation

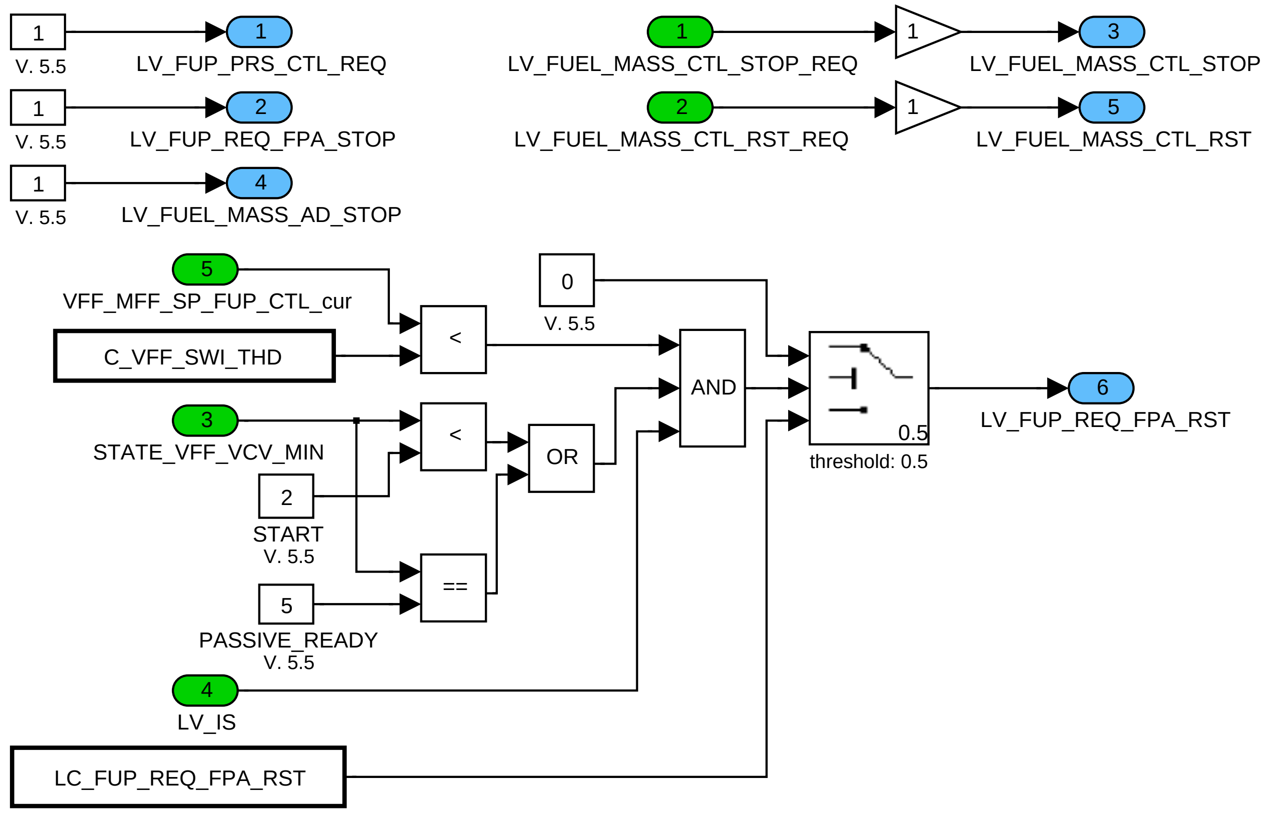


Figure 125:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_STATE_FUP_CTL/CLC_FLOW2PRS/CLC_FLOW2PRS_1

34.12.2.2.3.3.2 Fuel mass request control & set calculation

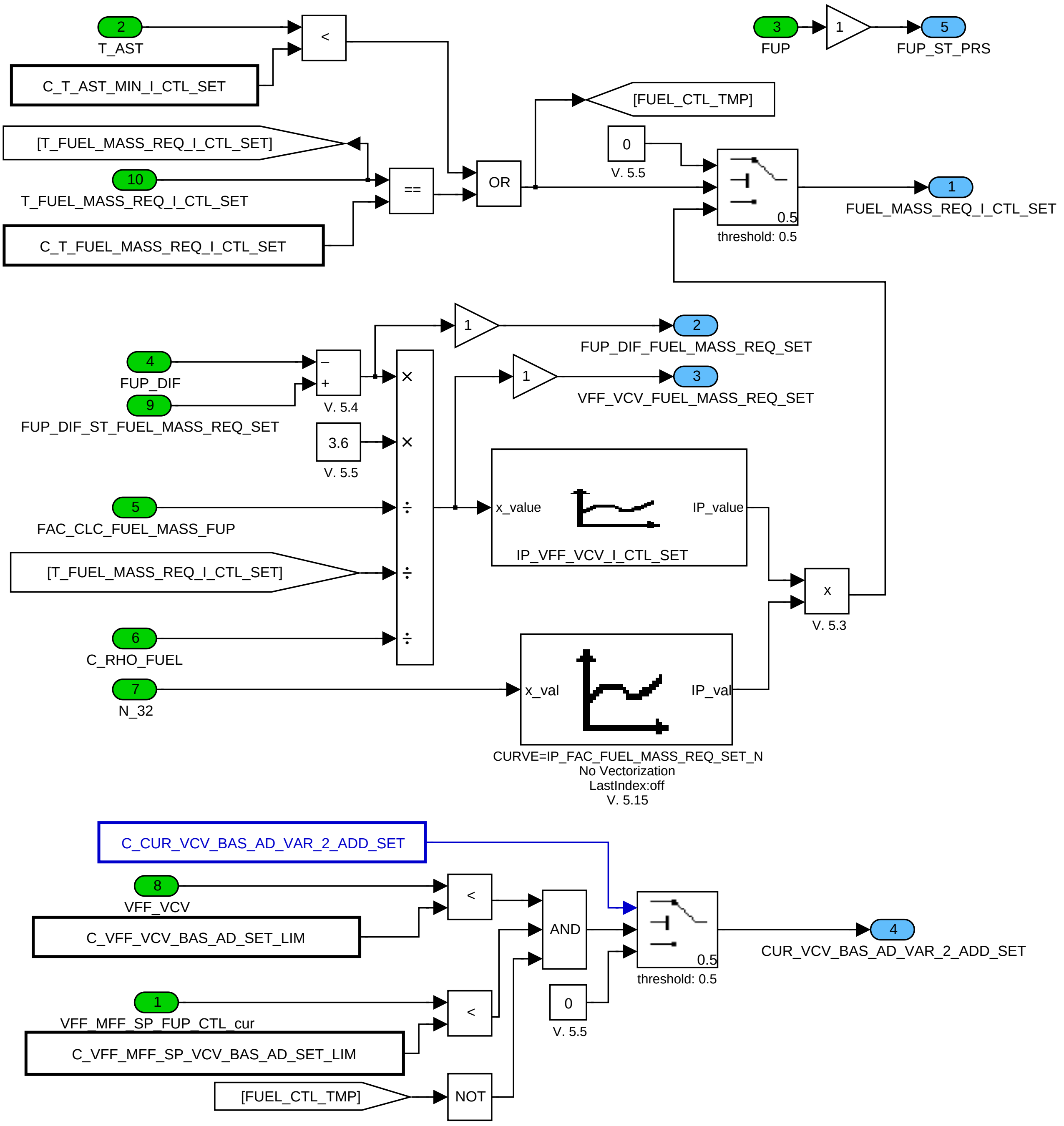


Figure 126:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_STATE_FUP_CTL/CLC_FLOW2PRS/CLC_FLOW2PRS_2

34.12.2.2.3.4 Pressure calculation

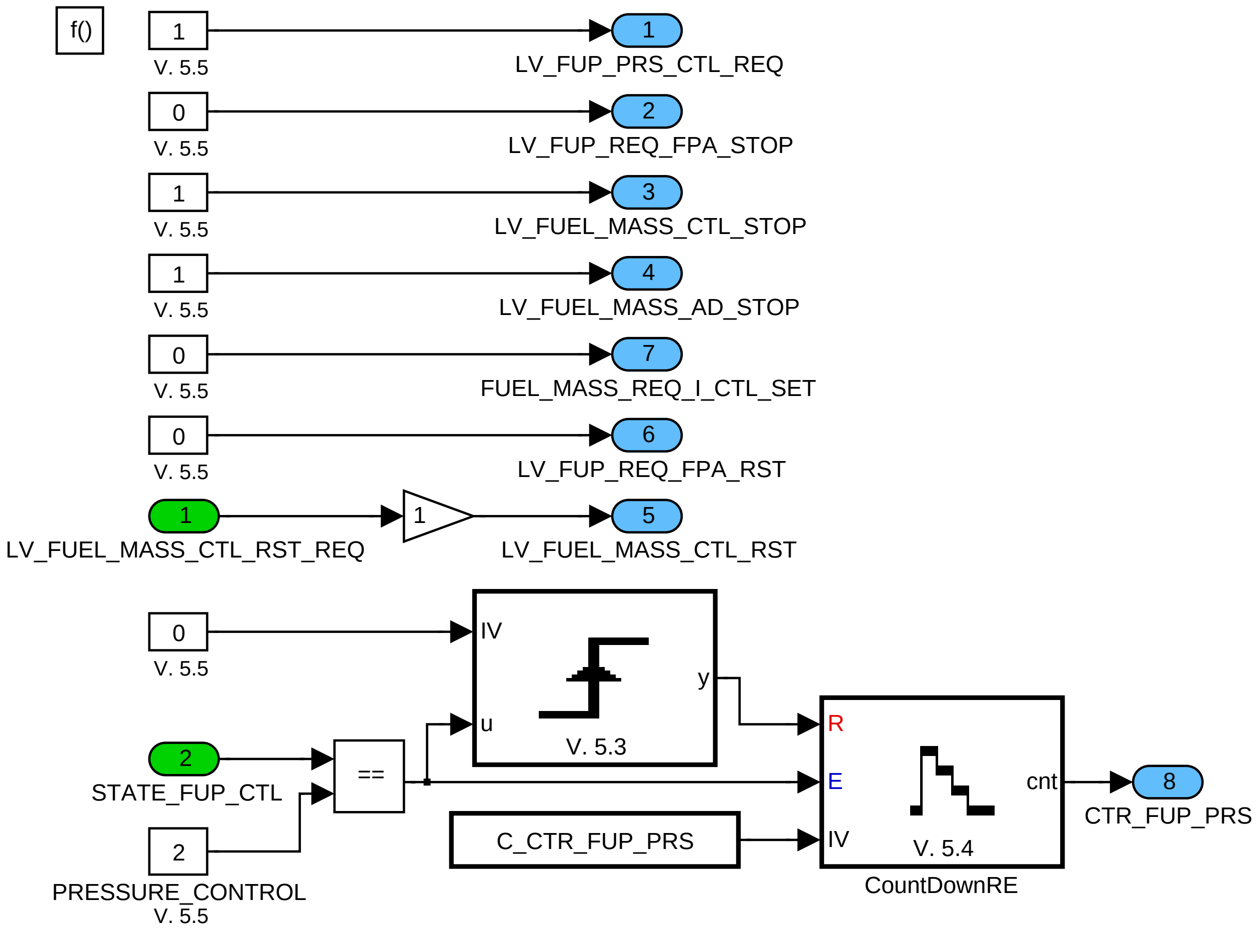


Figure 127:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_STATE_FUP_CTL/CLC_PRS

34.12.2.2.3.5 Pressure to flow calculation

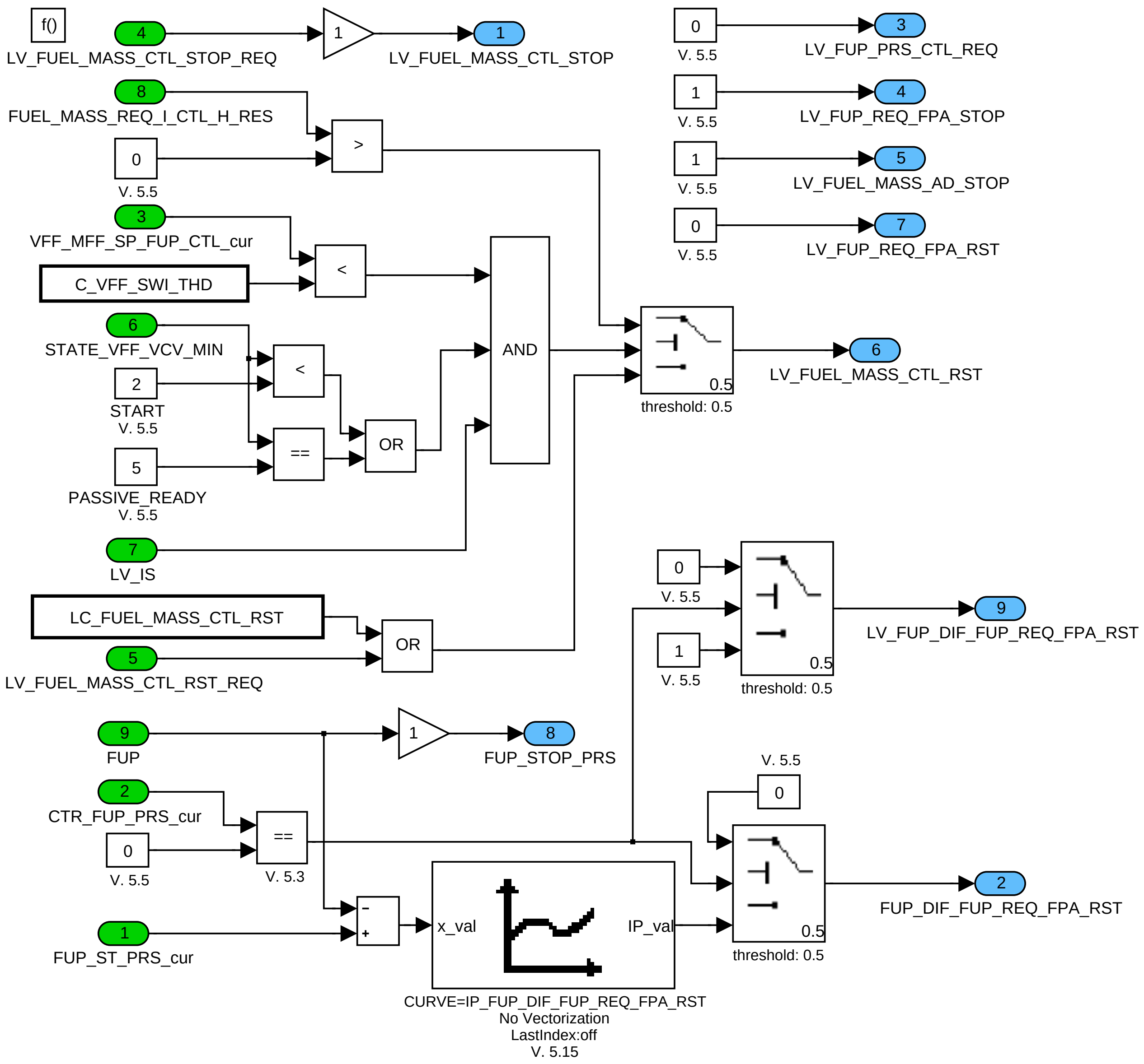
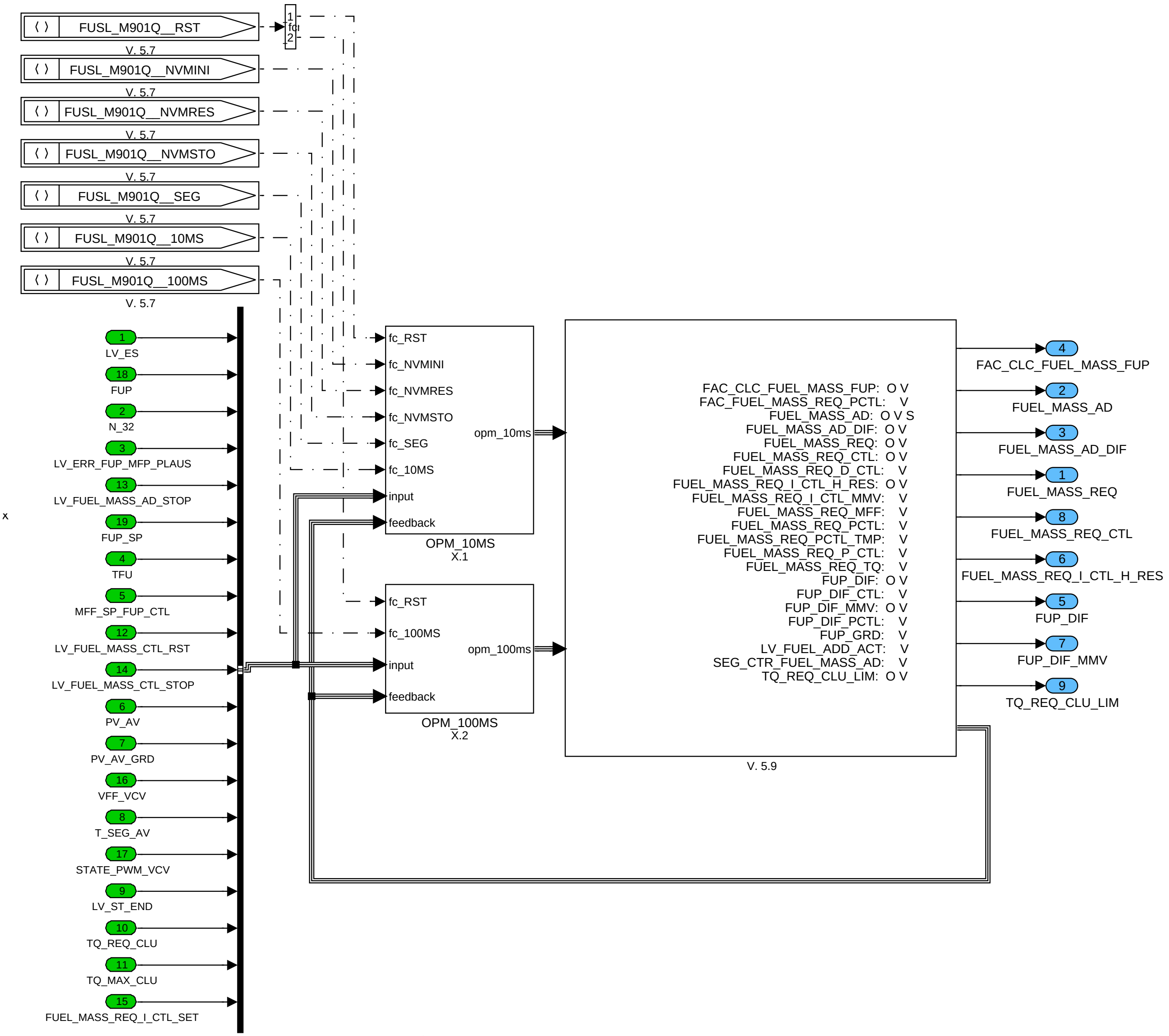


Figure 128:
Path: FUSL_M9049/OPM_SEG/OPM/CLC_STATE_FUP_CTL/CLC_PRS2FLOW

34.13 Fuel pressure control

Overview



SDA_SRS / SDA V 5.0.4 10−Nov−2006

Figure 129:
Path: FUSL_M901Q

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
FAC_CLC_FUEL_MASS_FUP	O/V	0... FFH	0... 510	2	[hPa/mg]
Factor for compressibility and density of gasoline and rail					
FAC_FUEL_MASS_REQ_PCTL	V	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Weighting factor for the precontroller					
FUEL_MASS_AD	O/V/S	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Adaptive fuel mass					
FUEL_MASS_AD_DIF	O/V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Deviation of the fuel mass for the adaption					
FUEL_MASS_REQ	O/V	0... FFFFH	0... 1389	0.0211948	[mg]
Requested fuel mass from the pump					
FUEL_MASS_REQ_CTL	O/V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Controller output					
FUEL_MASS_REQ_D_CTL	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
D-part of controller					
FUEL_MASS_REQ_I_CTL_H_RES	O/V	80000000... 7FFFFFFFH	-694.510597391 ...694.489402609	323.402e-9	[mg]
I-part of the controller					
FUEL_MASS_REQ_I_CTL_MMV	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Mean value of the I-part of the controller					
FUEL_MASS_REQ_MFF	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Additive fuel mass (DT1) due to injection fuel mass					
FUEL_MASS_REQ_P_CTL	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
P-part of the controller					
FUEL_MASS_REQ_PCTL	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Requested fuel mass from the precontroller					
FUEL_MASS_REQ_PCTL_TMP	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Requested fuel mass from the precontroller (without the influence of the weighting factor)					
FUEL_MASS_REQ_TQ	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Additive fuel mass (DT1) due to Torque request					
FUP_DIF	O/V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Deviation of fuel pressure setpoint and measured fuel pressure					
FUP_DIF_CTL	V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Deviation of measured fuel pressure and FUP_MDL or FUP_SP (depending on LC_USE_FUP_MDL_CTL)					
FUP_DIF_MMV	O/V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Deviation of fuel pressure setpoint and measured fuel pressure moving mean value					
FUP_DIF_PCTL	V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Deviation of fuel pressure setpoint and FUP_MDL or FUP (depending on LC_USE_FUP_MDL_PCTL)					
FUP_GRD	V	8000... 7FFFH	-86945.33 ...86942.67	2.6533608	[hPa/10ms]
Fuel pressure gradient					

LV_FUEL_ADD_ACT	V	0... 1H	0... 1	1	[-]
Logical variable indicating that additional fuel mass is added to precontrol					
LV_FUEL_MASS_REQ_CTL_CLR	-	0... 1H	0... 1	1	[-]
Logical variable for reset of CTL/AD part at fast pressure decrease					
SEG_CTR_FUEL_MASS_AD	V	0... FFFFH	0... 65535	1	[-]
Segment counter before adaptation is allowed					
TQ_REQ_CLU_LIM	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Limited requested torque at clutch for FUSL DT1 activation					

Input Data:

LV_ES	N_32	LV_ERR_FUP_MFP_PLAUS	TFU
MFF_SP_FUP_CTL	PV_AV	PV_AV_GRD	T_SEG_AV
LV_ST_END	TQ_REQ_CLU	TQ_MAX_CLU	LV_FUEL_MASS_CTL_RST
LV_FUEL_MASS_AD_STOP	LV_FUEL_MASS_CTL_STOP	FUEL_MASS_REQ_I_CTL_SET	VFF_VCV
STATE_PWM_VCV	FUP	FUP_SP	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CLC_FUEL_PUMP_CTL	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Amount of segments of calculation recurrency of high pressure pump control					
C_CRLC_FUP_DIF_MMV	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Filter constant for the fuel pressure difference					
C_CTR_CYC_FUEL_ADD	1	0... FFFFH	0... 65535	1	[-]
Number of segments where adding additional fuel mass to precontrol is active					
C_DLY_FUP_SP	1	1... 6H	1... 6	1	[-]
Delay of the FUP_SP in calculation cycles					
C_FAC_FUEL_MASS_TQ	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[mg/Nm]
Correlation factor from Nm to mg/stk					
C_FAC_N_MASS_FUEL_REQ_PCTL	1	0... FFH	0... 9.9609375	0.0390625	[-]
Factor for the precontroller depending on engine speed.					
C_FAC_TQ_REQ_FUEL_MASS	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Factor to limit the TQ_REQ_CLU to the real value (no over Request)					
C_FUEL_MASS_AD_COR_CRLC	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Filter constant for the fuel mass adaptation					
C_FUEL_MASS_AD_MAX	1	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Maximum adaptive fuel mass					
C_FUEL_MASS_AD_MIN	1	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Minimum adaptive fuel mass					
C_FUEL_MASS_REQ_ADD	1	0... FFFFH	0... 1389	0.0211948	[mg]
Additive fuel mass depending on engine speed.					
C_FUEL_MASS_REQ_I_CTL_INI	1	0... FFFFH	-694.5... 694.478806	0.0211945	[mg]
Initialisation of the I-part of the flow controller depending on the fuel temperature					
C_FUEL_MASS_REQ_I_CTL_MAX	1	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]
Maximum I-part of the controller					
C_FUEL_MASS_REQ_I_CTL_MIN	1	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg]

Minimum I-part of the controller					
C_FUEL_MASS_REQ_I_CTL_MMV_CRLC	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Filter constant for moving mean value calculation of the I-part					
C_N_FUEL_MASS_AD_MAX	1	0... FFH	0... 8160	32	[rpm]
Maximum engine speed threshold for allowing fuel mass adaption					
C_N_FUEL_MASS_AD_MIN	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed threshold for allowing fuel mass adaption					
C_PV_GRD_THD_FUEL_ADD	1	0... FFH	-1250... 1240.234375	9.765625	[%/s]
Pedal value gradient threshold for adding fuel mass					
C_PV_THD_FUEL_ADD	1	0... FFH	0... 99.609375	0.390625	[%]
Pedal value threshold for adding fuel mass					
C_SEG_CTR_FUEL_MASS_AD_MAX	1	0... FFFFH	0... 65535	1	[-]
Number of segments with stable conditions before an adaptation occurs					
C_VFF_VCV_I_CTL_STOP_MAX	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
Last breakpoint of the characteristic line of the VCV					
C_VFF_VCV_I_CTL_STOP_MIN	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
First breakpoint of the characteristic line of the VCV					
ID_DLY_FUP_GRD	8	1... 5H	1... 5	1	[-]
LDP_N_32_ID_DLY_FUP_GRD	8	0... FFH	0... 8160	32	[rpm]
Number of driftersteps for FUP_GRD calculation					
IP_FAC_CLC_FUEL_MASS_FUP	3	0... FFH	0... 510	2	[hPa/mg]
LDP_TFU_IP_FAC_CLC_FUEL_MASS	3	0... FEH	-48... 142.5	0.75	[°C]
Factor for compressibility and density of gasoline and rail					
IP_FAC_FUEL_MASS_REQ_D_CTL	8	0... FFH	0... 3.984375	0.015625	[-]
LDPM_N_32_FUSL_1	8	0... FFH	0... 8160	32	[rpm]
Weighting factor for the controller					
IP_FAC_FUEL_MASS_REQ_I_CTL	8	0... FFH	0... 3.984375	0.015625	[-]
LDPM_N_32_FUSL_1	8	0... FFH	0... 8160	32	[rpm]
Weighting factor for the controller					
IP_FAC_FUEL_MASS_REQ_P_CTL	8	0... FFH	0... 3.984375	0.015625	[-]
LDPM_N_32_FUSL_1	8	0... FFH	0... 8160	32	[rpm]
Weighting factor for the controller					
IP_FAC_FUEL_MASS_REQ_PCTL	8	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
LDP_FUP_DIF_PCTL_IP_FAC_FUEL	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
Weighting factor for the precontroller					
IP_FUEL_MASS_REQ_ADD_MFF_SP	8	0... FFH	0... 255	1	[seg]
LDPM_MFF_SP_FUP_CTL_2_FUSL	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
DT1 factor (output = input*K/T1) depending on injection fuel mass					
IP_FUEL_MASS_REQ_ADD_TQ	8	0... FFH	0... 255	1	[seg]
LDPM_TQ_REQ_CLU_LIM_1_FUSL	8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
DT1 Factor (output = input*K/T1) depending on Torque request					
IP_FUEL_MASS_REQ_D_CTL	12*12	0... FFFFH	-694.5... 694.478806	0.0211945	[mg]
LDP_FUP_GRD_IP_FUEL_MASS_REQ	12	0... FFFFH	-86945.33 ...86942.67	2.6533608	[hPa/10 ms]
LDP_FUP_DIF_CTL_IP_FUEL_MASS	12	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
D-part of the controller					
IP_FUEL_MASS_REQ_I_CTL	8	0... FFFFH	-694.5... 694.478806	0.0211945	[mg]
LDPM_FUP_DIF_CTL_1_FUSL	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
I-part of the controller					

IP_FUEL_MASS_REQ_P_CTL	8	0... FFFFH	-694.5... 694.478806	0.0211945	[mg]
LDPM_FUP_DIF_CTL_1_FUSL	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
P-part of the controller					
IP_T1_FUEL_MASS_ADD_MFF_SP	8	0... FFH	0... 255	1	[seg]
LDPM_MFF_SP_FUP_CTL_2_FUSL	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Constant time (in segment) for additive fuel mass (DT1) depending on injection fuel mass					
IP_T1_FUEL_MASS_ADD_TQ	8	0... FFH	0... 255	1	[seg]
LDPM_TQ_REQ_CLU_LIM_1_FUSL	8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Constant time (in segment) for additive fuel mass (DT1) depending on Torque request					
LC_AD_CLR_FUEL_MASS_AD	1	0... 1H	0... 1	1	[-]
Switch for reseting the fuel mass adaptation					
LC_FUEL_MASS_REQ_CTL_CLR	1	0... 1H	0... 1	1	[-]
Switch for enabling CTL/AD part reset at fast pressure decrease					
LC_USE_FUEL_MASS_REQ_CTL	1	0... 1H	0... 1	1	[-]
Switch for using the controller output for the fuel mass requested calculation					
LC_VFF_VCV_I_CTL_RST	1	0... 1H	0... 1	1	[-]
Enabling of I-Part controler reset if the first or last breakpoint of the characteristic line for the VCV is reached					

General Information

The control algorithm can be used for all kind of flow controlled pump. This function is the interface of the fuel pressure acquisition and the fuel pressure setpoint calculation to the fuel pump control.

It calculates basically a requested fuel mass to be delivered into the fuel rail by the HP pump. This calculation is based on the fuel pressure setpoint, measrured fuel pressure, the fuel mass flow setpoint and the feedback of the pump interface, which is the fuel mass delivered by the pump. Additionally there is the input fuel temperature, which is needed to correct the calculation of the fuel mass in the rail.

34.13.1 Function Description of 10MS

The functionality is split into several parts. First part is the precalcualtion of the fuel mass. In this module first the deviation between the fuel pressure setpoint and the measured fuel pressure is calculated. This deviation is the basis for the calculation of the requested fuel mass from the precontroller. This input will be corrected by a weighting factor and a conversion factor, which includes the density of the fuel and the compressibility of the rail and the fuel.

To get a correct input for the mass fuel flow this will be corrected within the module mass fuel flow setpoint for fuel pressure control with the effective fuel cut off ratio.

The controller is split into the fuel mass adaptation and the actual controller. The calculated fuel mass adaptation takes as a basis the old value to the adaptive fuel mass and uses a actual calculated fuel mass adapation deviation. The adaptive fuel mass is min/max limited. The fuel mass adapation deviation is only calculated if the engine speed and the pump fuel mass is within certain limits. Another criteria is a segment counter, which allows the adaptation only after a certain amount of segments with stable conditions. The basis for the adapation is a filtered Ipart of the controller.

The input of the actual controller is also a fuel pressure deviation. This time the deviation is calculated out of the measured fuel pressure fuel pressure setpoint. Out of this deviation the P part and the I part of the controller is calculated out of maps. After that calculation the P part can be reseted by certain conditions. The I part taken out of the map will be added up with the old value. This value can be 'frozen or reseted. Anyway there is limitation to min/max values. The I part and the P part are finally added up to the controller output.

With this controller output, the precalculation and the corrected mass fuel flow setpoint value the requested fuel mass is calculated.

Application Conditions

Initialization: RST, NVMINI, NVMRES, NVMSTO
Activation: SEG: LV_ES == 0 && LV_ST_END == 1
10MS: LV_ES == 0 && LV_ST_END == 0
Deactivation: LV_ES == 1

Function description

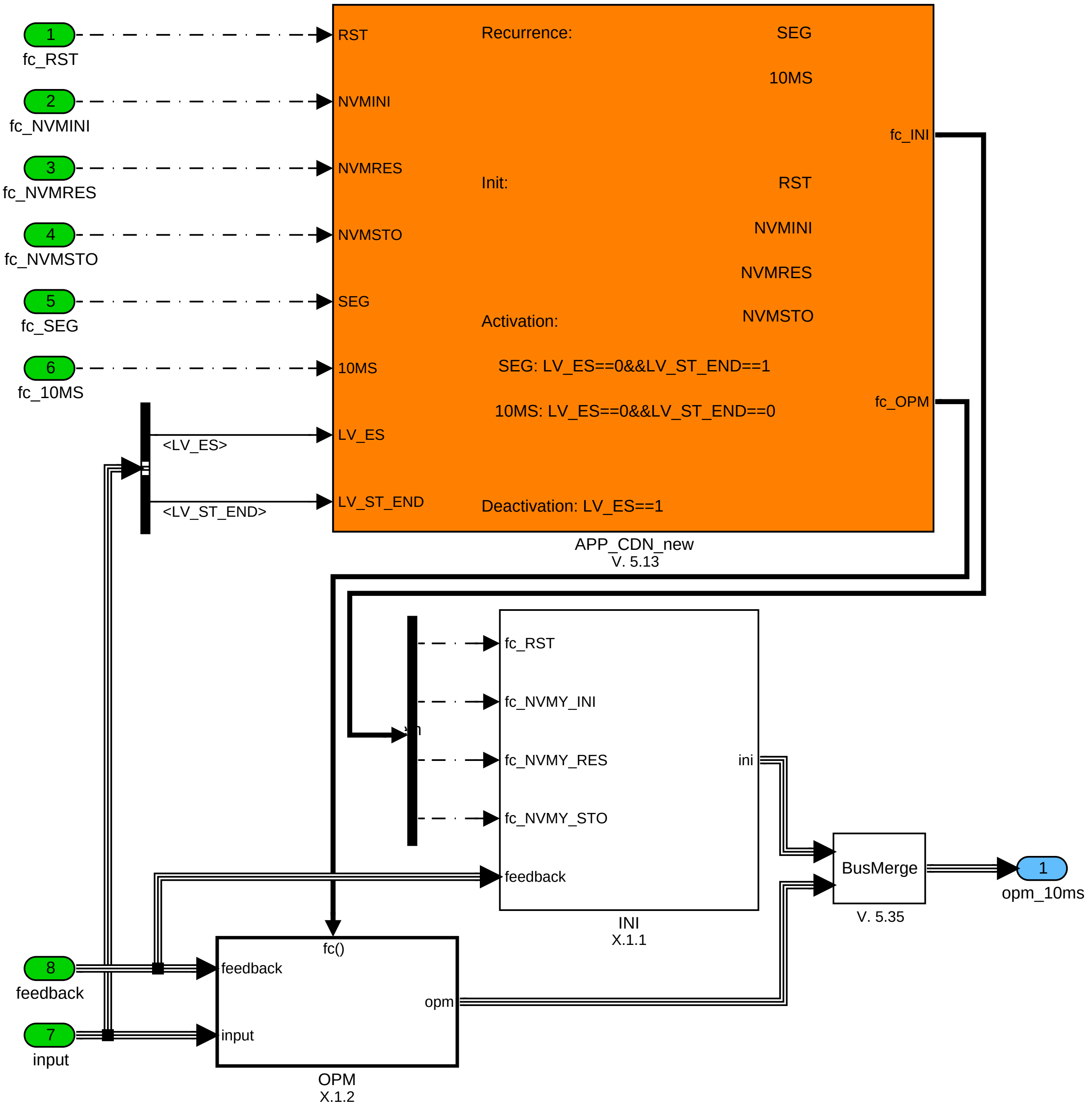


Figure 130:
Path: FUSL_M901Q/OPM_10MS

34.13.1.1 Initialisation

Initialisation at reset event & Non volatile memory management is done here.

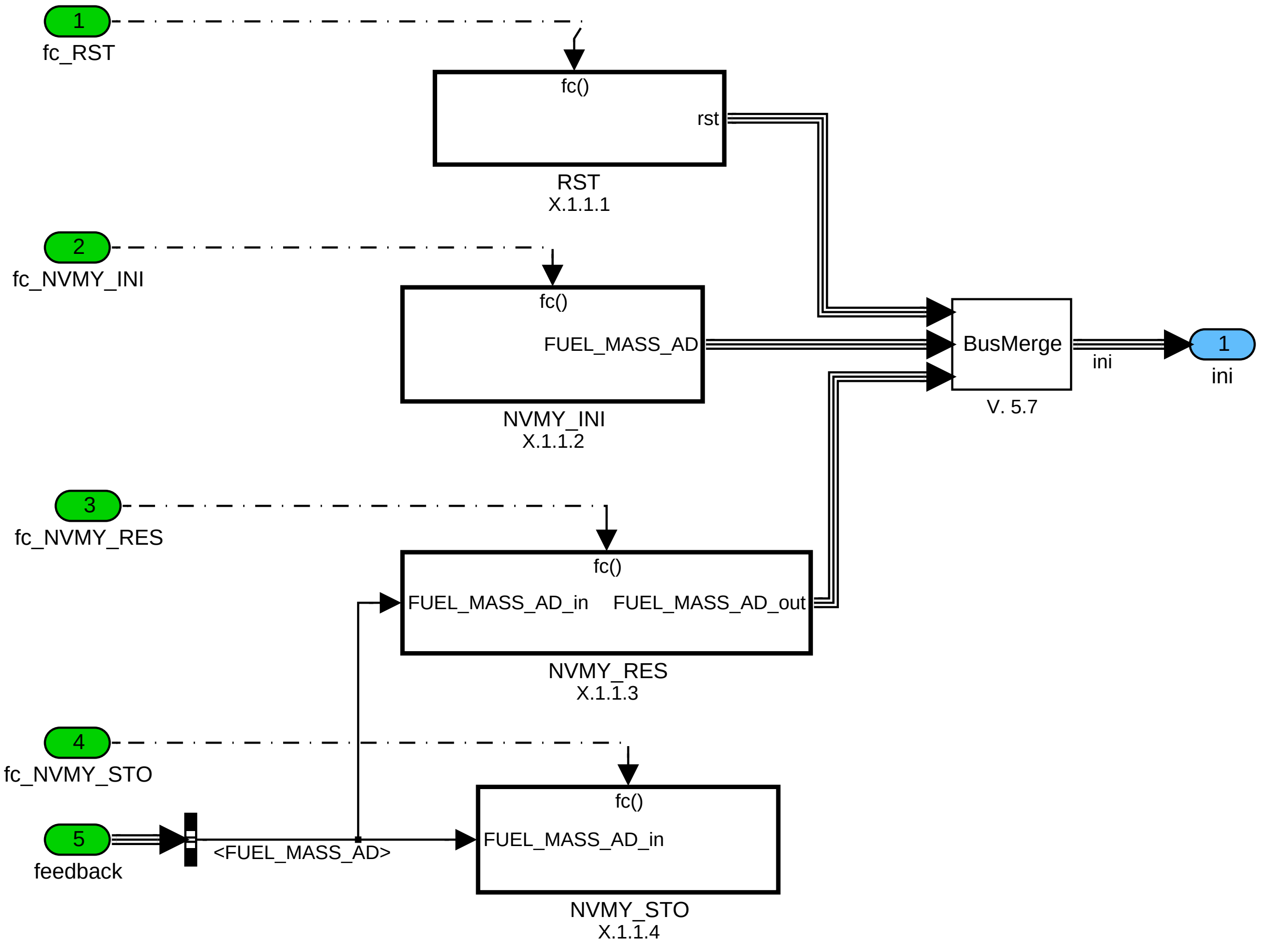


Figure 131:
Path: FUSL_M901Q/OPM_10MS/INI

34.13.1.1.1 Initialisation at Non-Volatile memory

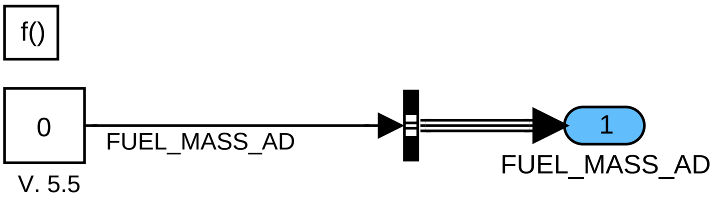


Figure 132:
Path: FUSL_M901Q/OPM_10MS/INI/NVMY_INI

34.13.1.1.2 Restore management at Non-Volatile memory

Figure 133:
Path: FUSL_M901Q/OPM_10MS/INI/NVMY_RES

34.13.1.1.3 Store management at Non-Volatile memory

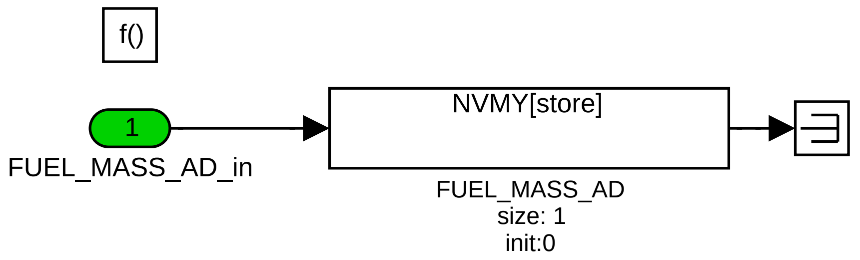


Figure 134:
Path: FUSL_M901Q/OPM_10MS/INI/NVMY_STO

34.13.1.1.4 Initialisaion at reset event

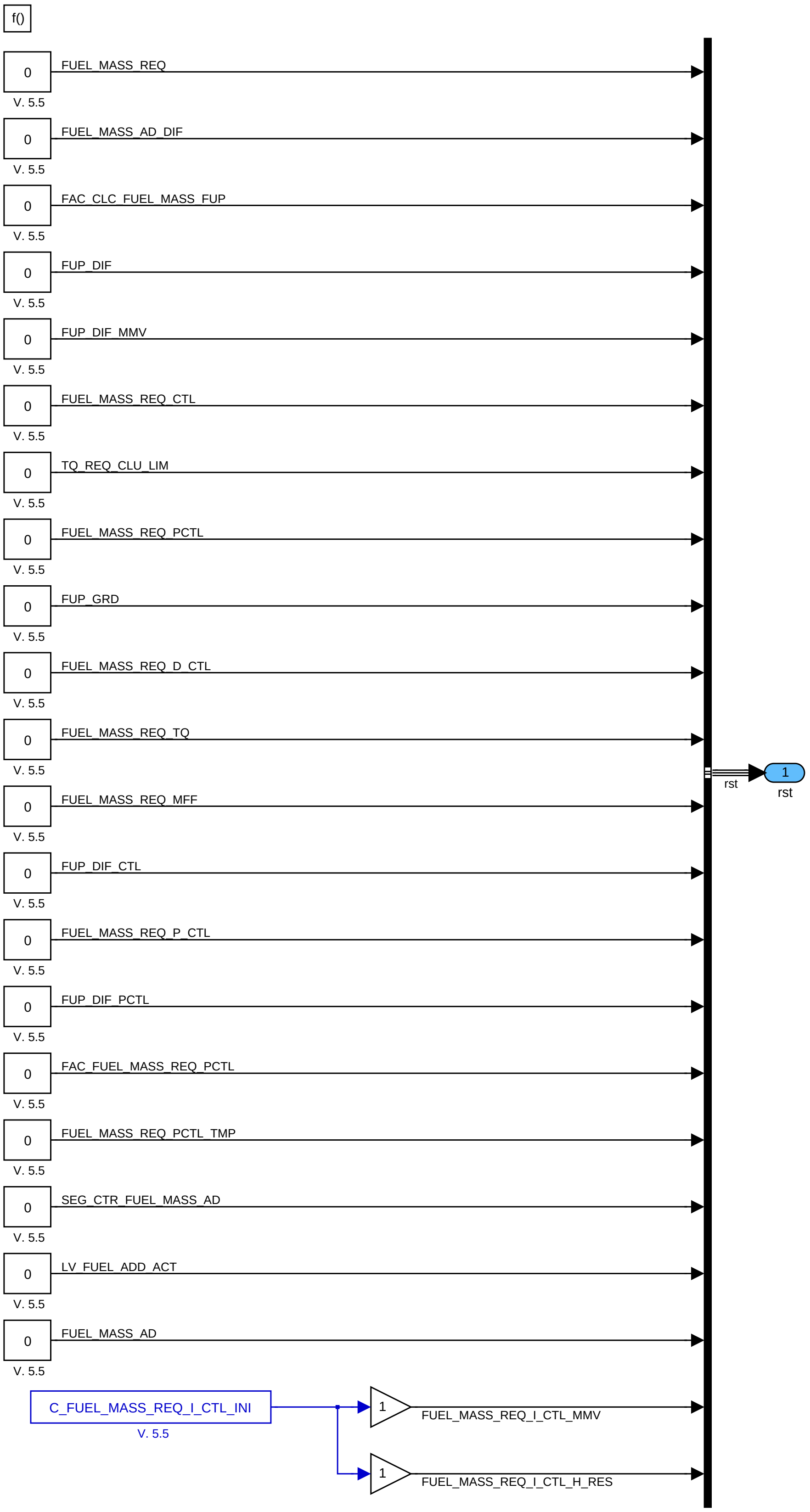


Figure 135:
Path: FUSL_M901Q/OPM_10MS/INI/RST

34.13.1.2 Fuel mass & fuel pressure control

For better visibility of the function the calculation is split into 3 parts.

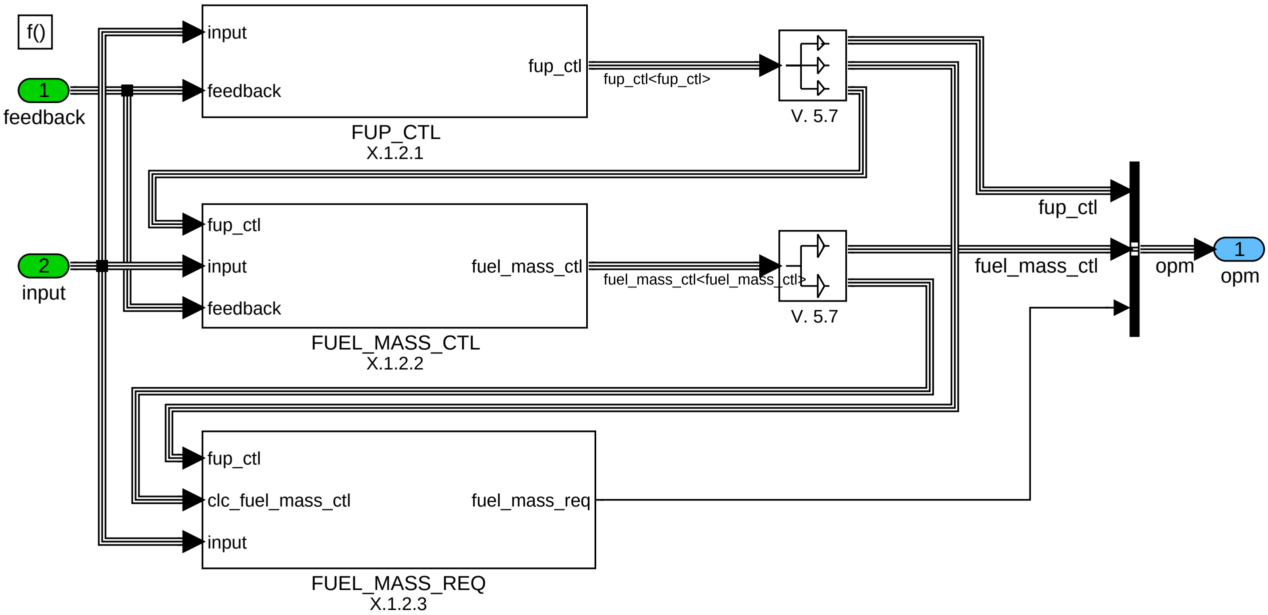


Figure 136:
Path: FUSL_M901Q/OPM_10MS/OPM

34.13.1.2.1 Calculation of Fuel pressure control

This subblock is split into three other blocks in which the precontrol, the adaptive parts and the conditions for a additive fuel mass for the precontrol are calculated

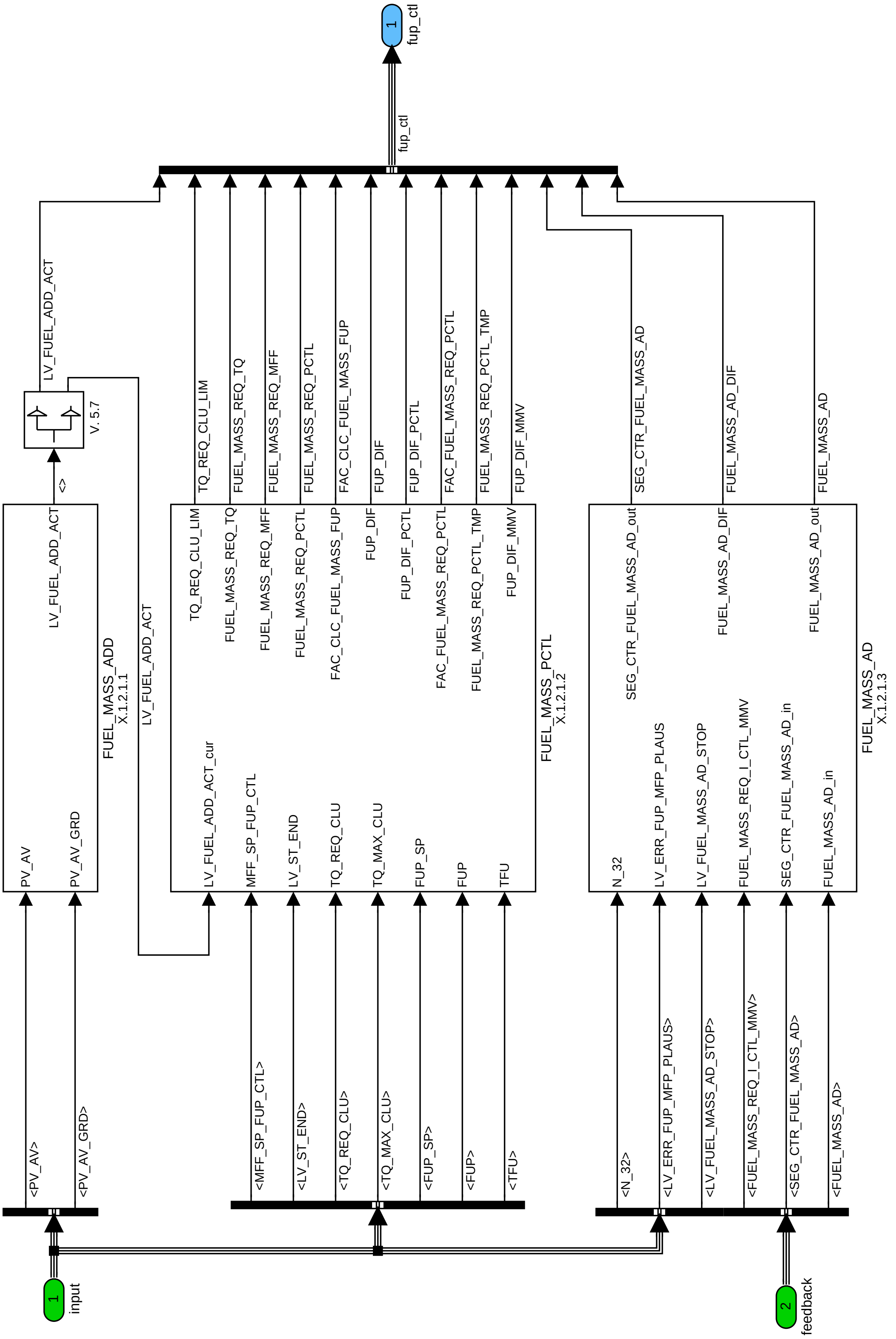


Figure 137:
Path: FUSL_M901Q/OPM_10MS/OPM/FUP_CTL

34.13.1.2.1.1 Additional fuel mass active

Depending on the pedal value and the pedal value gradient a fuel mass is added to the precontroller for a certain amount of cycles. This is indicated by a flag.

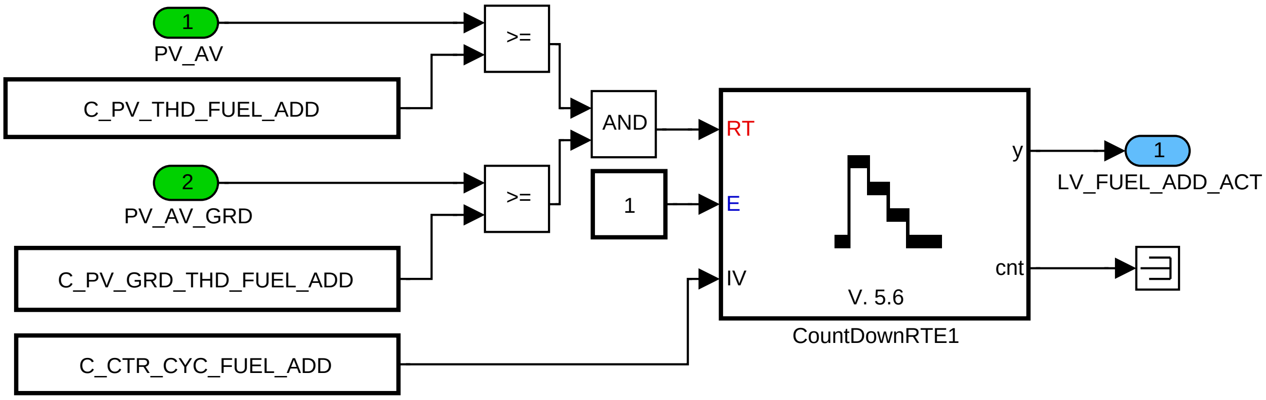


Figure 138:
Path: FUSL_M901Q/OPM_10MS/OPM/FUP_CTL/FUEL_MASS_ADD

34.13.1.2.1.2 Calculation

Out of the fuel pressure deviation related to the used fuel pressure the precontrol value is calculated. This value can be modified by the weighting factor given by the map. Out of this calculated precontrol fuel pressure value a precontrol fuel mass must be chosen. This is done by the second map. Additionally to that a fuel mass can be added to the precontrol.

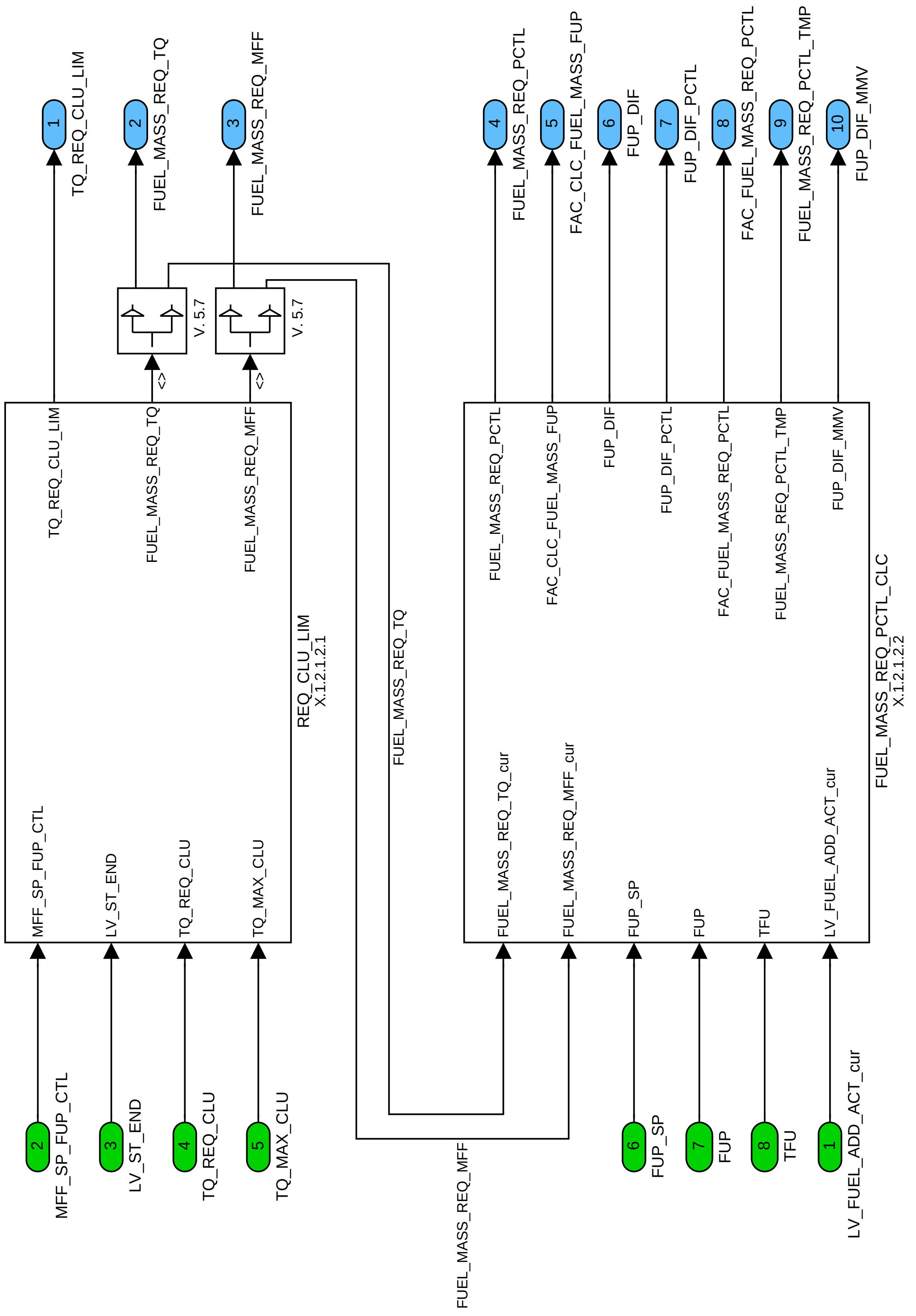


Figure 139:
Path: FUSL_M901Q/OPM_10MS/OPM/FUP_CTL/FUEL_MASS_PCTL

34.13.1.2.1.2.1 Calulcation of Mass fuel flow

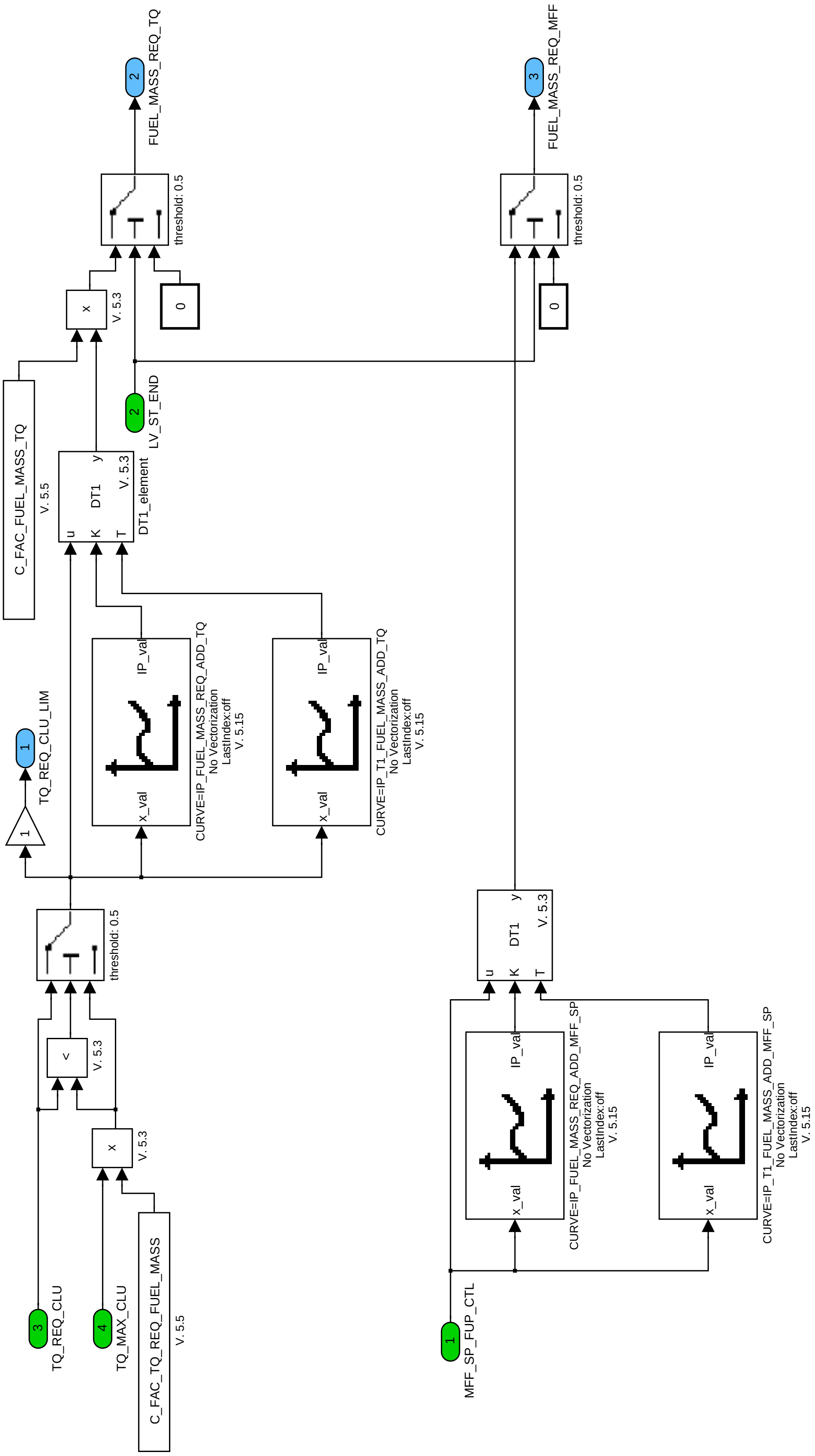


Figure 140:
Path: FUSL_M901Q/OPM_10MS/OPM/FUP_CTL/FUEL_MASS_PCTL/REQ_CLU_LIM

34.13.1.2.1.2.2 Calculation of requested fuel mass from precontroller

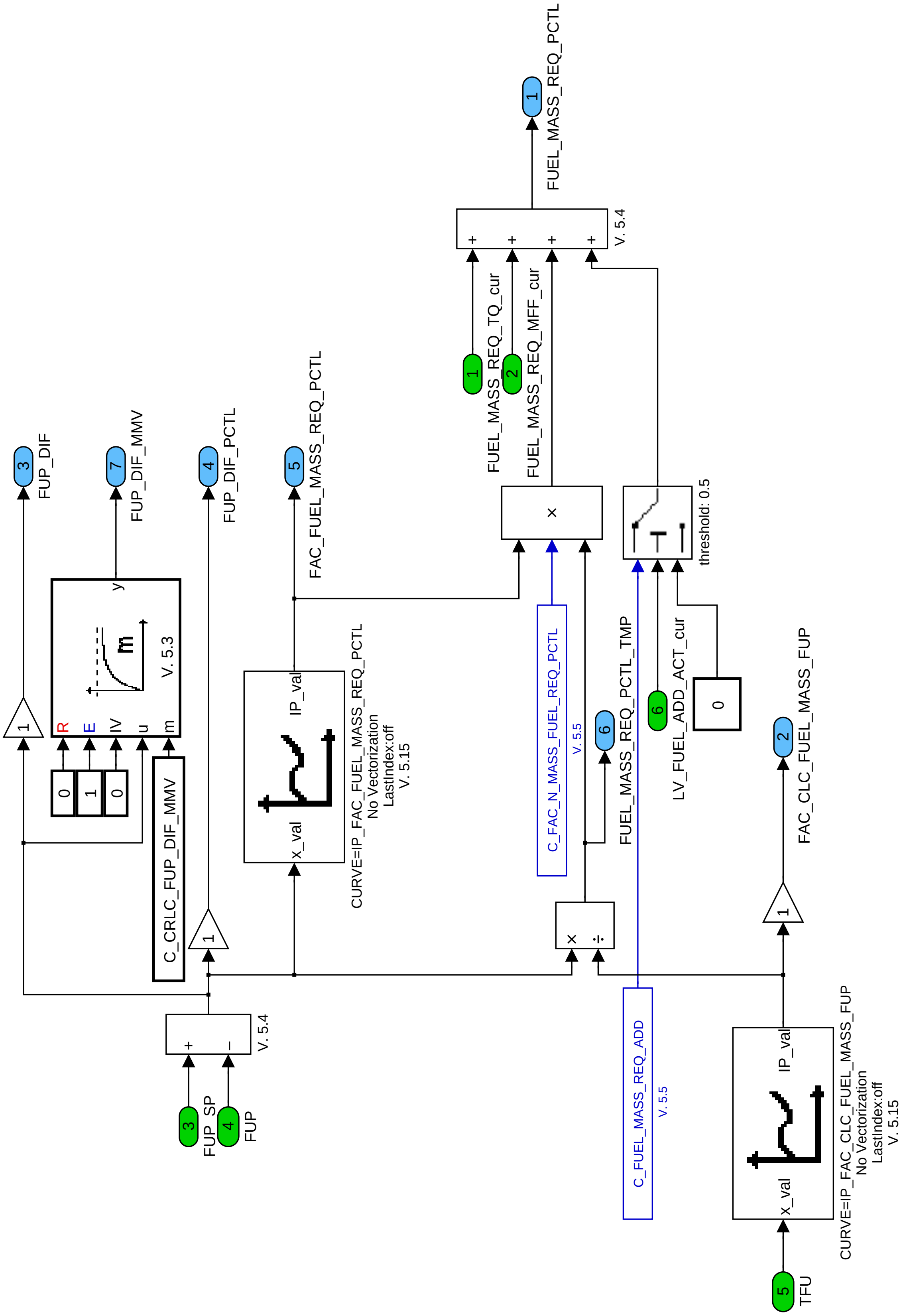


Figure 141:
Path: FUSL_M901Q/OPM_10MS/OPM/FUP_CTL/FUEL_MASS_PCTL/FUEL_MASS_REQ_PCTL_CLC

34.13.1.2.1.3 Calculation of fuel mass adaptation

The adaptation is split into two parts. One is calculated the adaptive fuel mass difference and one is responsible for the actual fuel mass determination

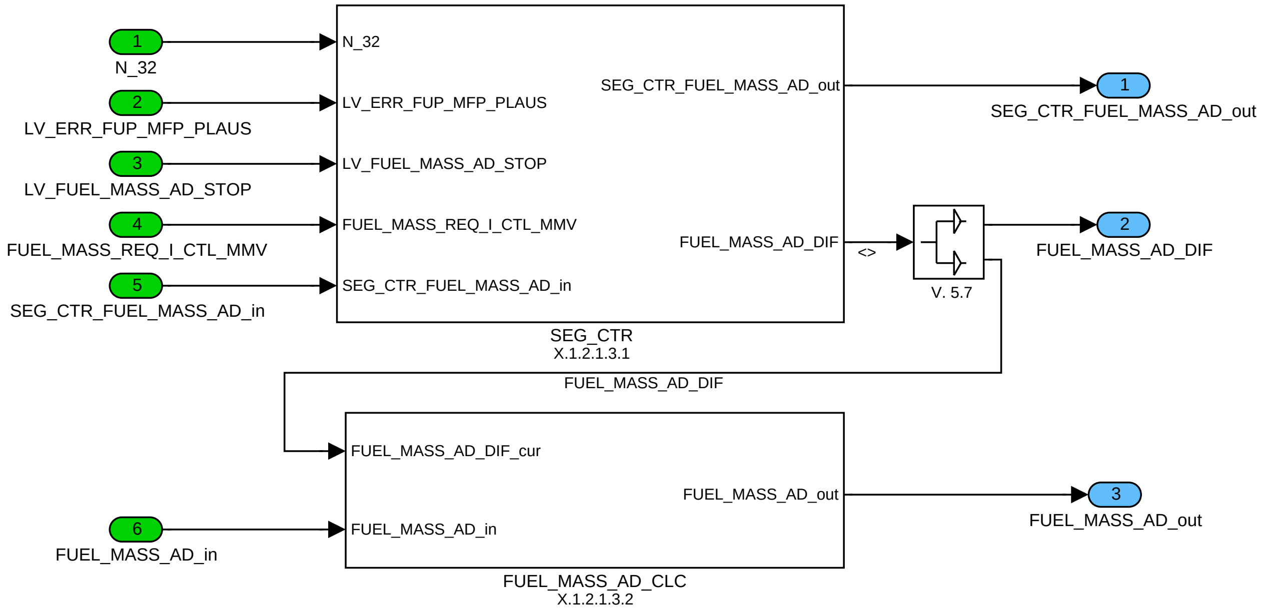


Figure 142:
Path: FUSL_M901Q/OPM_10MS/OPM/FUP_CTL/FUEL_MASS_AD

34.13.1.2.1.3.1 Calculation of adaptive fuel mass

The adapation value should not be above a maximum value and below a minimum value.

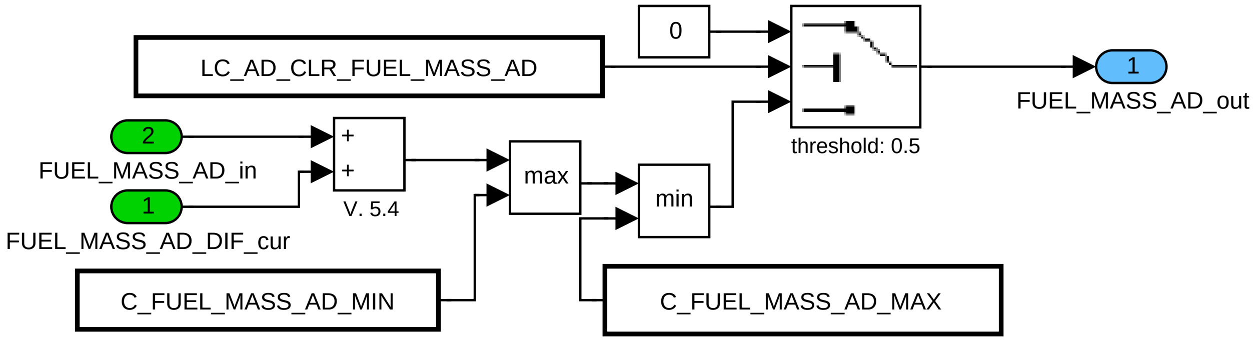


Figure 143:
Path: FUSL_M901Q/OPM_10MS/OPM/FUP_CTL/FUEL_MASS_AD/FUEL_MASS_AD_CLC

34.13.1.2.1.3.2 Calculation of adaptive fuel mass condition

In this block the adaptation condition are determined

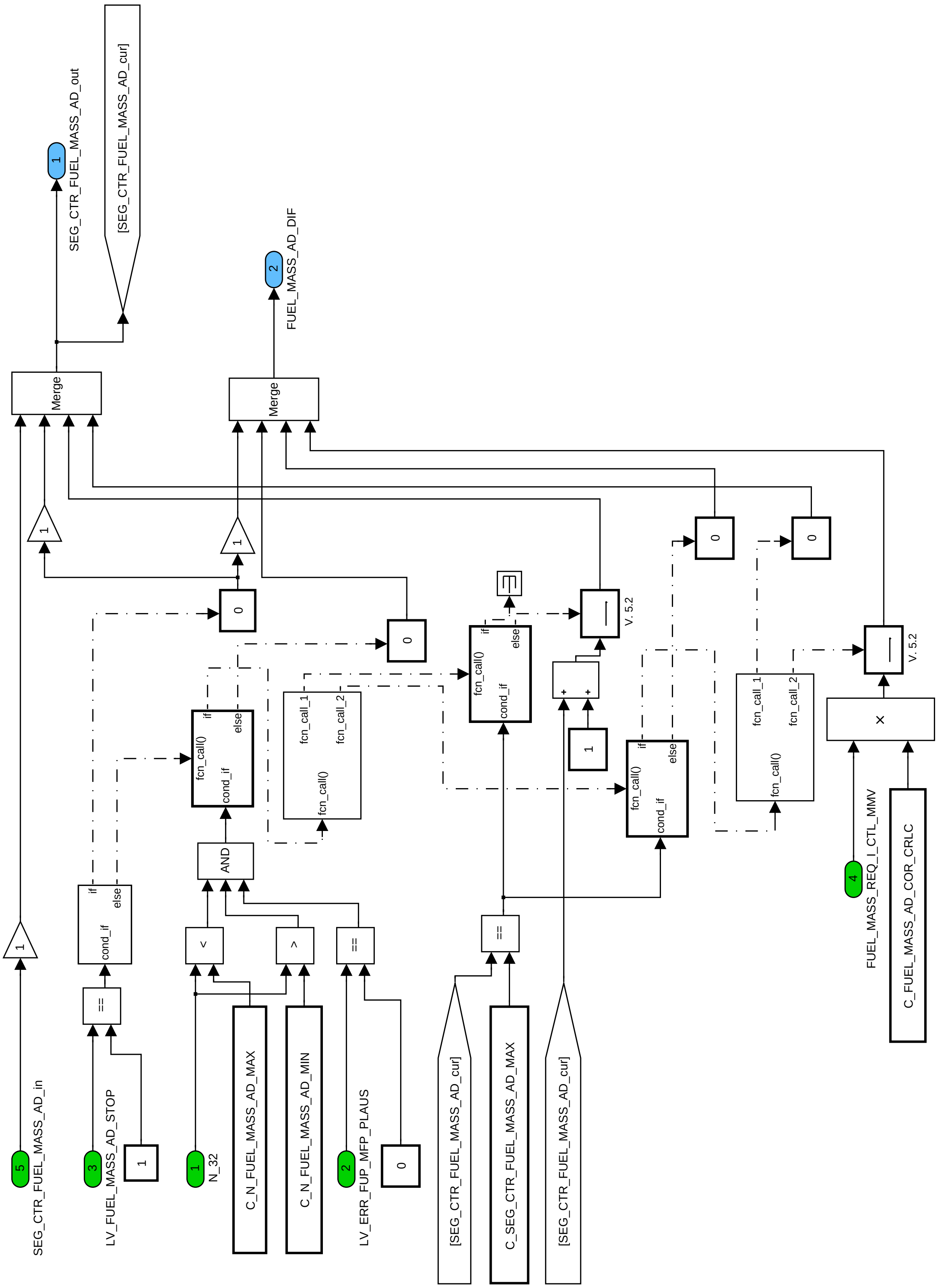


Figure 144:
Path: FUSL_M901Q/OPM_10MS/OPM/FUP_CTL/FUEL_MASS_AD/SEG_CTR

34.13.1.2.2 Fuel mass control

For a better overview the controller is split into two parts.

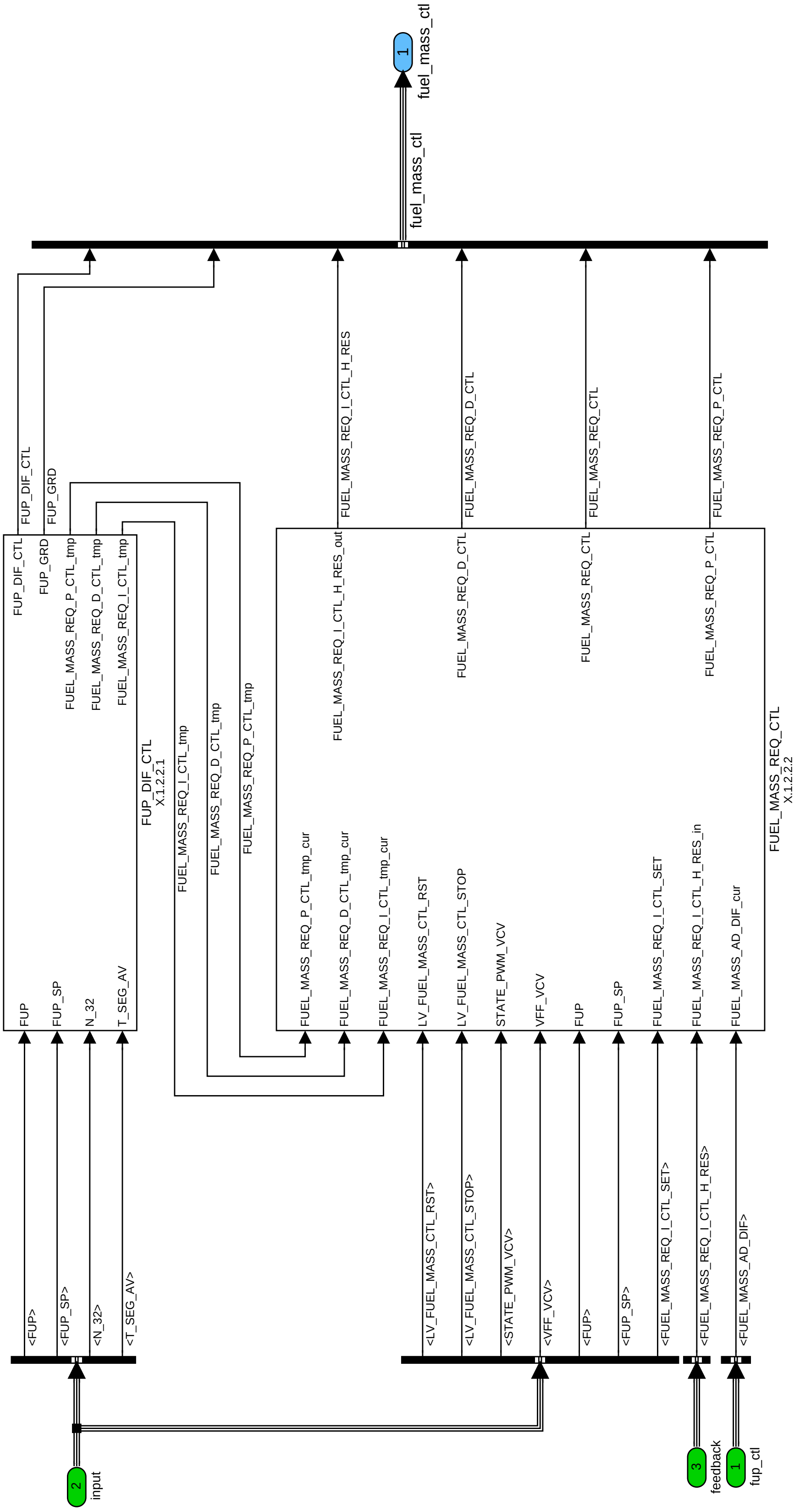


Figure 145:
Path: FUSL_M901Q/OPM_10MS/OPM/FUEL_MASS_CTL

34.13.1.2.2.1 Calculation of requested fuel mass by controller

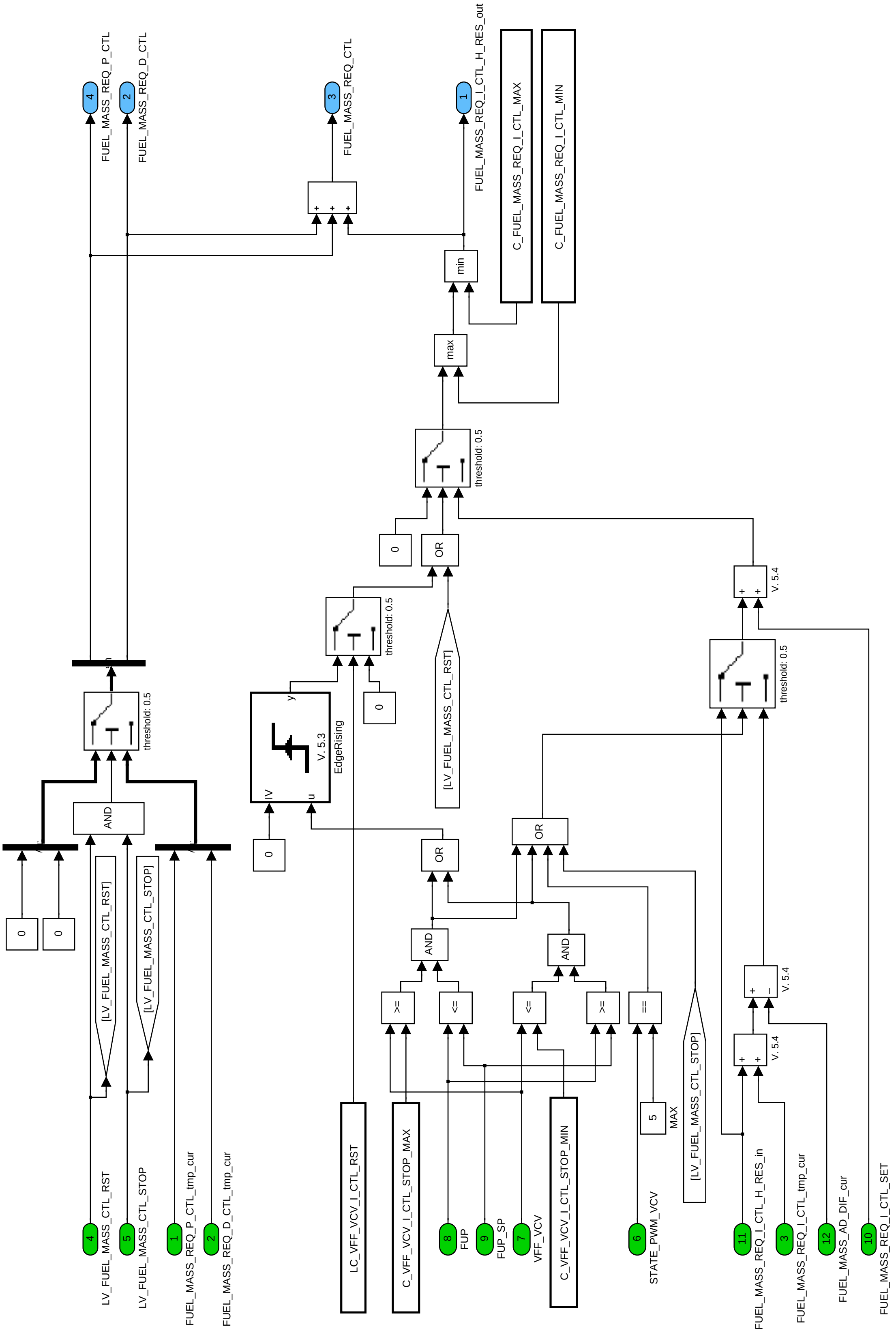


Figure 146:
Path: FUSL_M901Q/OPM_10MS/OPM/FUEL_MASS_CTL/FUEL_MASS_REQ_CTL

34.13.1.2.2.2 Calculation of fuel pressure deviation

The controller values are calculated out of the fuel pressure setpoint value with a delay and the fuel pressure.

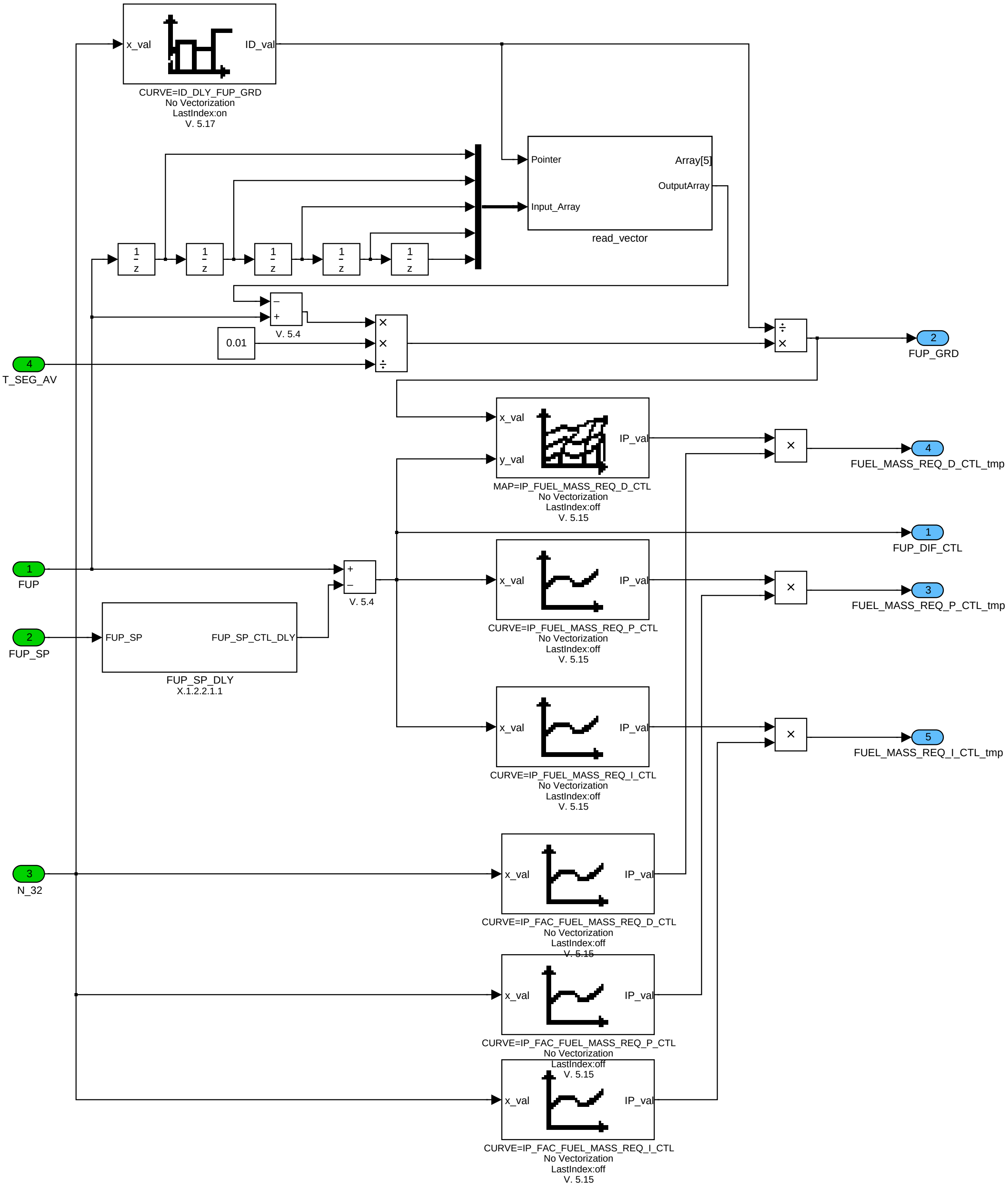


Figure 147:
Path: FUSL_M901Q/OPM_10MS/OPM/FUEL_MASS_CTL/FUP_DIF_CTL

34.13.1.2.2.2.1 Calculation of fuel pressure setpoint with delay

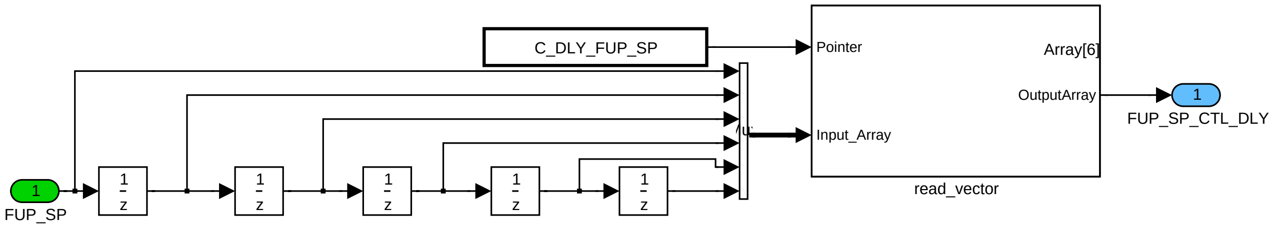


Figure 148:
Path: FUSL_M901Q/OPM_10MS/OPM/FUEL_MASS_CTL/FUP_DIF_CTL/FUP_SP_DLY

34.13.1.2.3 Calculation of fuel mass request

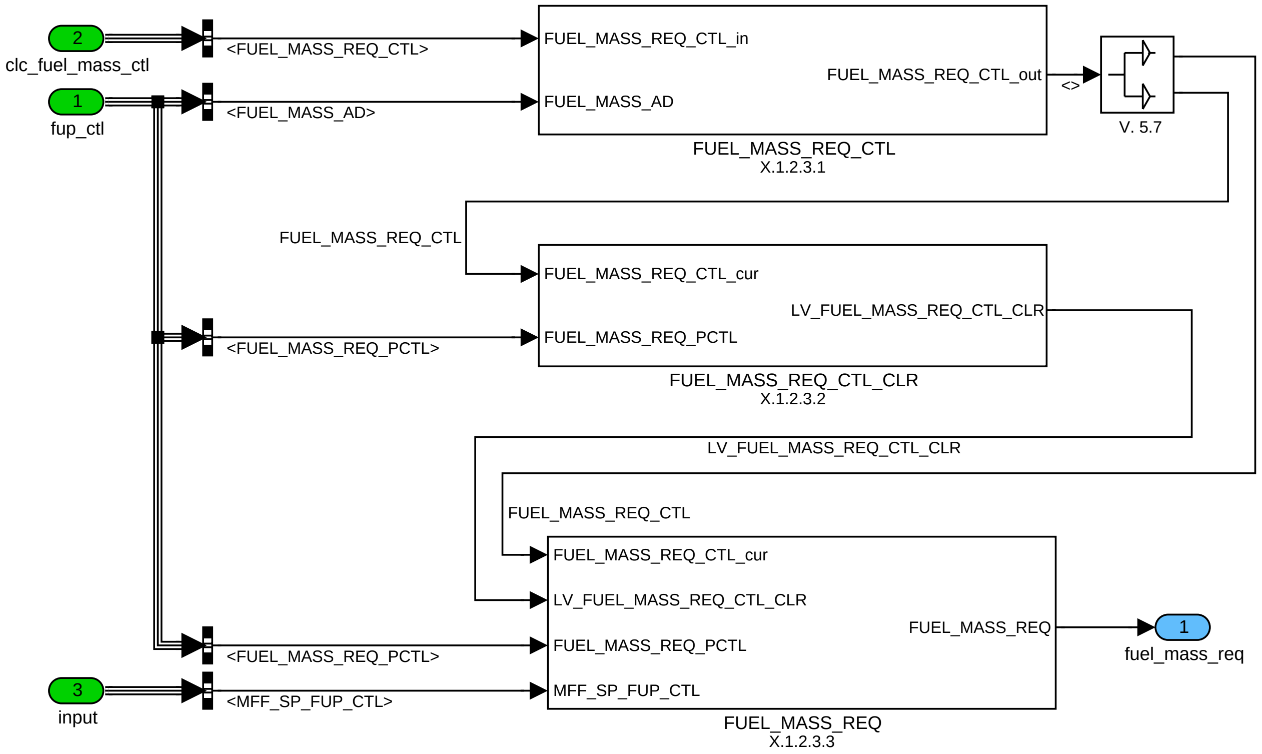


Figure 149:
Path: FUSL_M901Q/OPM_10MS/OPM/FUEL_MASS_REQ

34.13.1.2.3.1 Calculation of Fuel mass request control

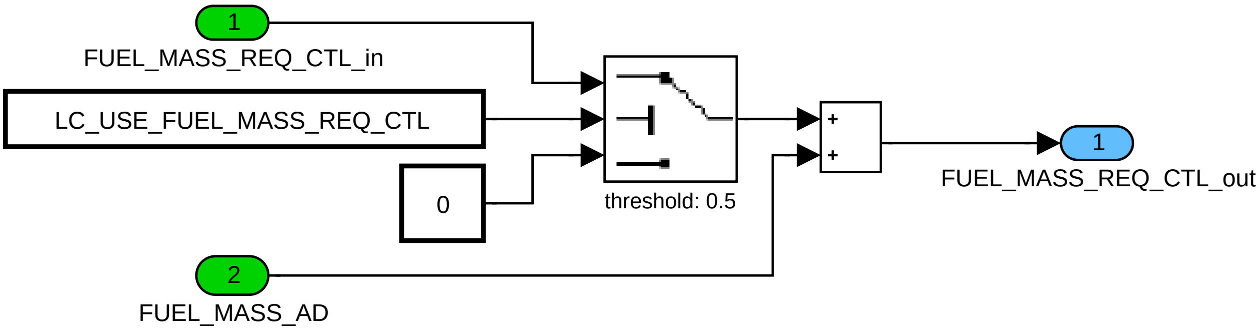


Figure 150:
Path: FUSL_M901Q/OPM_10MS/OPM/FUEL_MASS_REQ/FUEL_MASS_REQ_CTL

34.13.1.2.3.2 Calculation of logical variable for reset of CTL/AD part of fast pressure

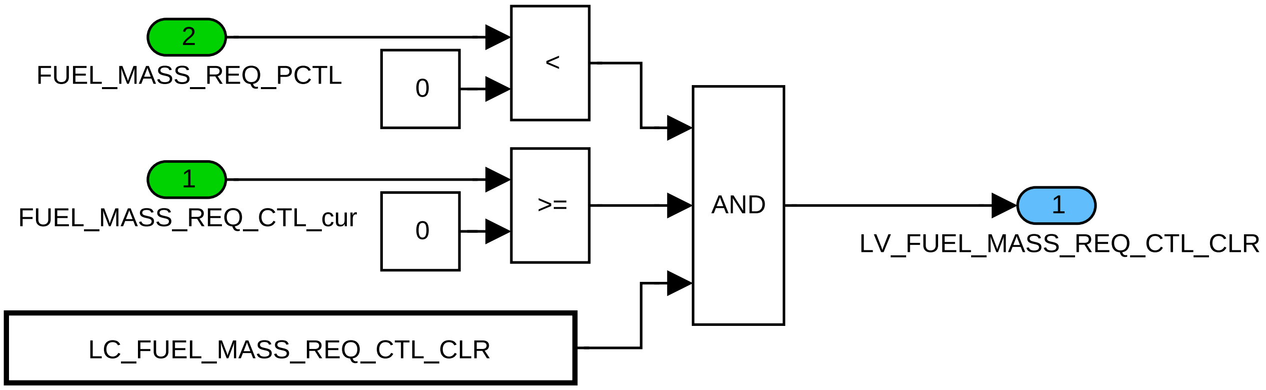


Figure 151:
Path: FUSL_M901Q/OPM_10MS/OPM/FUEL_MASS_REQ/FUEL_MASS_REQ_CTL_CLR

34.13.1.2.3.3 Calculation of fuel mass request

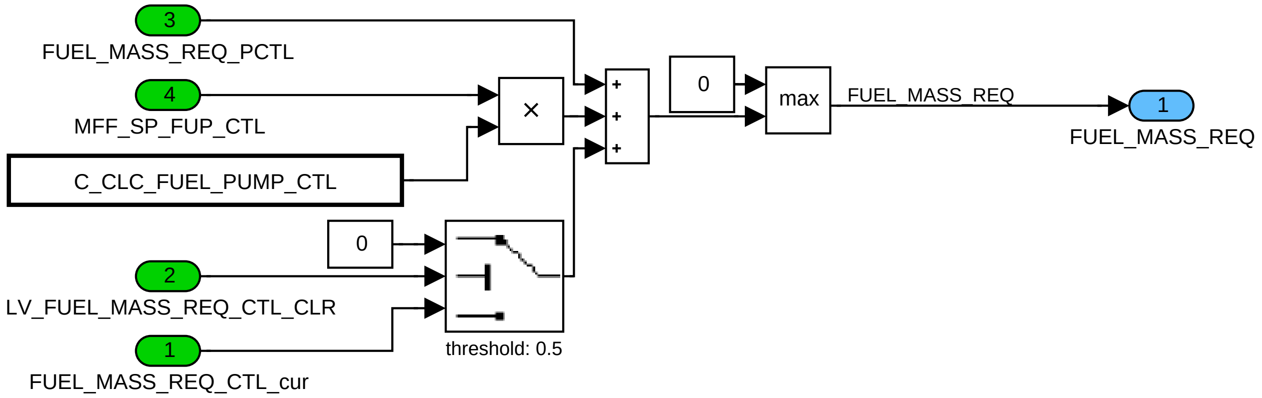


Figure 152:
Path: FUSL_M901Q/OPM_10MS/OPM/FUEL_MASS_REQ/FUEL_MASS_REQ

34.13.2 Function Description of 100MS

Only the module FUEL_MASS_MDL_DIF_I_MMV_CLC will be calculated every 100 ms.

Application Conditions

Initialization: RST
Recurrence: 100MS
Activation: LV_ES == 0
Deactivation: LV_ES == 1

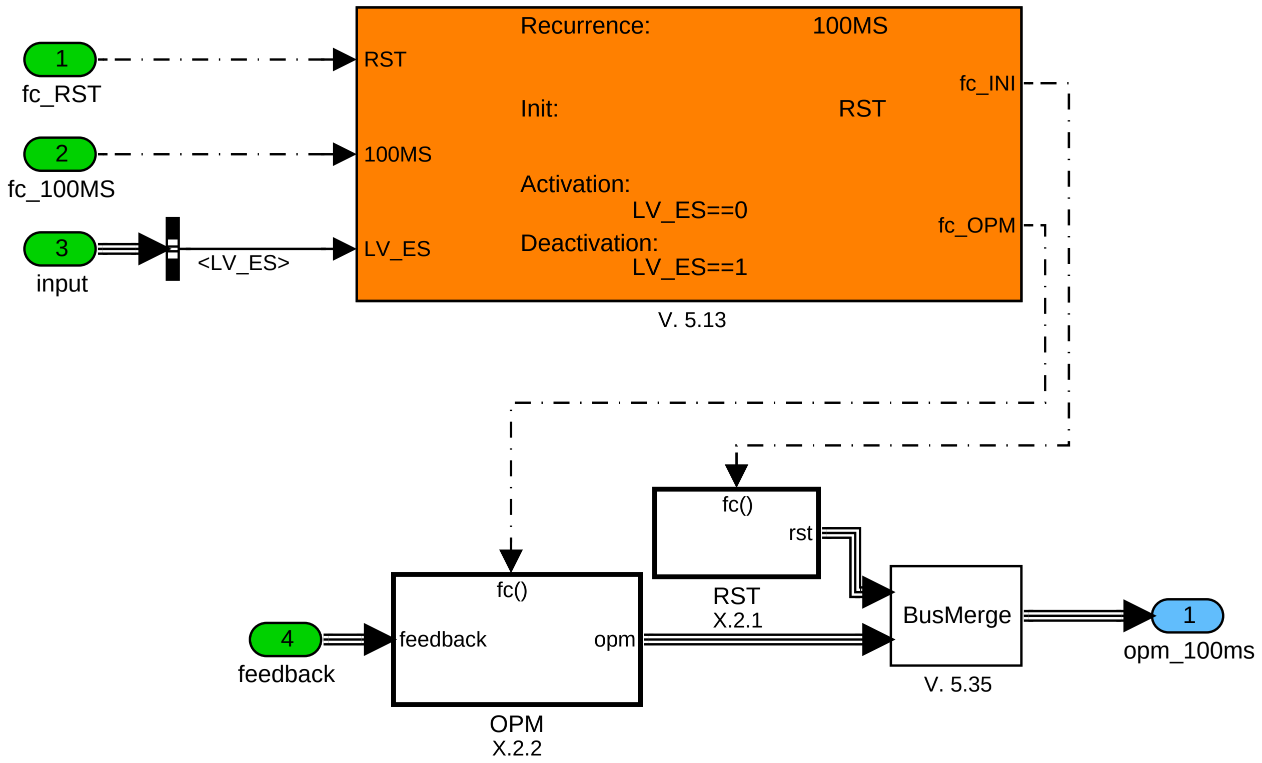


Figure 153:
Path: FUSL_M901Q/OPM_100MS

34.13.2.1 Formula Section

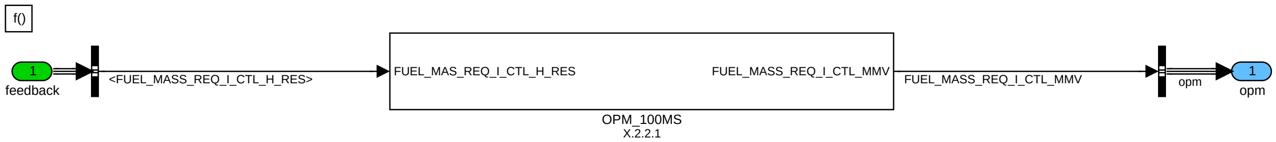


Figure 154:
Path: FUSL_M901Q/OPM_100MS/OPM

34.13.2.1.1 Moving mean value of fuel mass request control

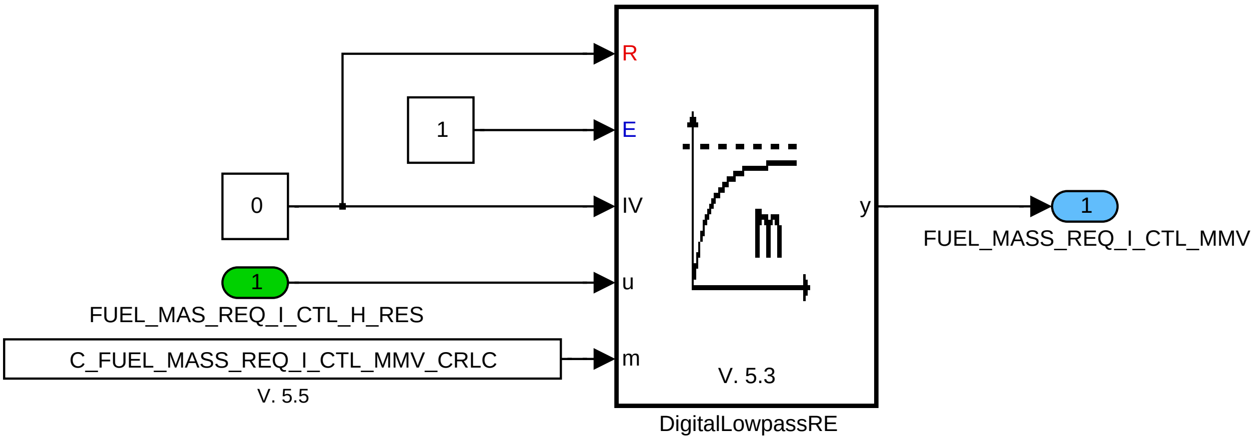


Figure 155:
Path: FUSL_M901Q/OPM_100MS/OPM/OPM_100MS

34.13.2.2 Initialisation at reset event

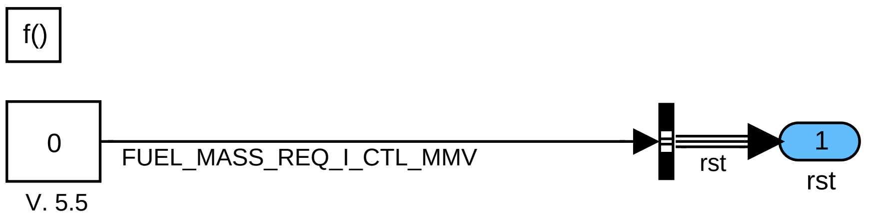


Figure 156:
Path: FUSL_M901Q/OPM_100MS/RST

34.14 Fuel pressure control (Appl. Inc.)

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
FUP_DIF_DLY	V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Deviation of FUP_SP and FUP with the delay of FUP_SP					
FUP_REQ_FPA	O/V	0... FFFFH	0... 347776	5.3067216	[hPa]
Required fuel pressure from the FPA					
FUP_REQ_FPA_I	O/V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
I-part of FUP_REQ_FPA					
FUP_REQ_FPA_I_TMP	V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Temporary I-part of FUP_REQ_FPA					
FUP_REQ_FPA_P	V	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
P-part of FUP_REQ_FPA					
FUP_REQ_FPA_PCTL	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Requested fuel pressure by the precontrol					
FUP_SP_FIL	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Filtered fuel pressure setpoint					
LV_DLY_FPA_CLL_REQ	V	0... 1H	0... 1	1	[-]
Logical variable for closed-loop control request to avoid overheating					
LV_FUP_DIF_FUP_REQ_FPA_RST_ACT	O	0... 1H	0... 1	1	[-]
Logical variable indicating that the setting of the pressure I-part of the controller in intervention mode is active					
LV_FUP_REQ_FPA_AD_INH	V	0... 1H	0... 1	1	[-]
Logical variable requesting inhibition of the adaptation of the controller					
LV_FUP_REQ_FPA_RST_ACT	O	0... 1H	0... 1	1	[-]
Logical variable indicating resetting of the I-part of the pressure controler					
STATE_FUP	O	0 1 2 3 4 5	LOW_PRESSURE_STATE FAST_PRESSURE_INCREASE PRESSURE_CONTROL OVER_PRESSURE OPEN_LOOP EMERGENCY	-	[-]
State of the fuel pressure controller					
STATE_FUP_TRAN	-	0 1 2 3 4 5 6 7 8	EMERGENCY LP LP_FPLN FPLN_PCTL OPL PCTL_OVP OVP_PCTL OPL_PCTL PCTL_FPLN	-	[-]
Transistion conditions for the state of the fuel pressure controller					
T_DLY_ERR_FUP	-	0... FFFFH	0... 655.35	0.01	[s]
Delay time for a fuel pressure error to start controller invention					
T_DLY_FPA_CLL_REQ	-	0... FFFFH	0... 655.35	0.01	[s]
Delay time for closed loop control to avoid overheating					
T_DLY_FPA_PROT	-	0... FFFFH	0... 655.35	0.01	[s]
Delay time for protection of the component					
T_DLY_FUP_MAX	-	0... FFFFH	0... 655.35	0.01	[s]

Delay time for the maximum fuel pressure to start controller invention

Input Data:

LV_ERR_CRASH_SIG	LV_ERR_EFP	LV_ERR_FPA	LV_ERR_FUP
LV_ERR_TCO	LV_ERR_VB	LV_ES	LV_RLY_ST
LV_ST_H_PRS	N_32	STATE_ERR_IV	FUP
FUP_DIF_FUP_REQ_FPA_RST	LV_FUP_DIF_FUP_REQ_FPA_RST	LV_FUP_REQ_FPA_RST	LV_FUP_REQ_FPA_STOP
STATE_VFF_VCV_MIN	FUP_SP	LV_FIRST_VLD_TOOTH	LV_IGK

Calibration Data:

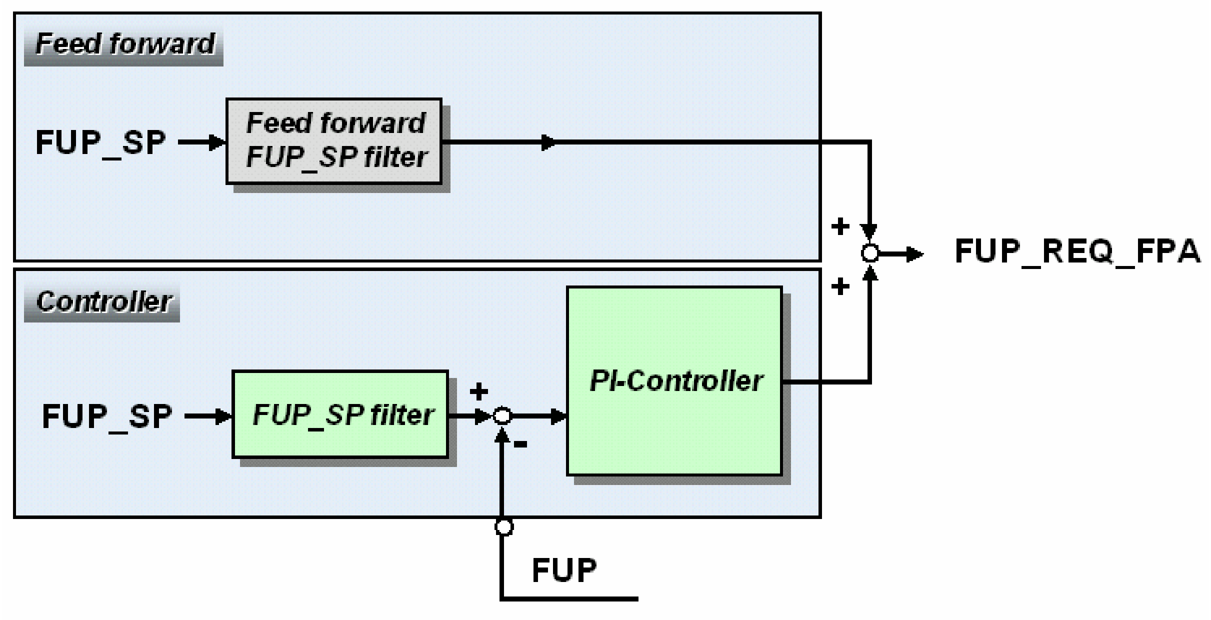
Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_FAC_FUP_SP_PCTL	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Factor for fuel pressure setpoint change influence to the feed forward controller					
C_FUP_MAX	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Maximum fuel pressure without controller intervention					
C_FUP_OFS_CTL	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Offset to the fuel pressure setpoint for controller strategy switch					
C_FUP_OFS_ST	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Offset to the fuel pressure setpoint for controller strategy switch out of start					
C_FUP_REQ_FPA_I_MAX	1	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Maximum value for integral part of FUP_REQ_FPA					
C_FUP_REQ_FPA_I_MIN	1	8000... 7FFFH	-173890 ...173884	5.3066911	[hPa]
Minimum value for integral part of FUP_REQ_FPA					
C_FUP_REQ_FPA_INI	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Initial value for FUP_REQ_FPA					
C_FUP_REQ_FPA_MAX	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Maximum value for FUP_REQ_FPA					
C_FUP_REQ_FPA_MIN	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Minimum value for FUP_REQ_FPA					
C_FUP_SP_CRLC	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlation constant for the FUP_SP for the PI-control					
C_FUP_SP_PCTL_CRLC	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
FUP_SP correlation constant for the precontrol of FPA_REQ_FPA					
C_N_32_FPA	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for leaving low pressure state					
C_T_DLY_ERR_FUP	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for a fuel pressure error to start controller invention					
C_T_DLY_FPA_CLL_REQ	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for closed loop control to avoid overheating					
C_T_DLY_FPA_PROT	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for protection of the component					
C_T_DLY_FUP_MAX	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for the maximum fuel pressure to start controller invention					
IP_FUP_REQ_FPA_I	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
LDPM_FUP_DIF_DLY_FUSL_1	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
Integral parameter of FUP_REQ_FPA					
IP_FUP_REQ_FPA_P	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]

LDPM_FUP_DIF_DLY_FUSL_1	8	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
Proportional parameter of FUP_REQ_FPA					

General Information

Application hint: NC_CLC_FUEL_PUMP_CTL = 1 for inlet controlled multi piston pump

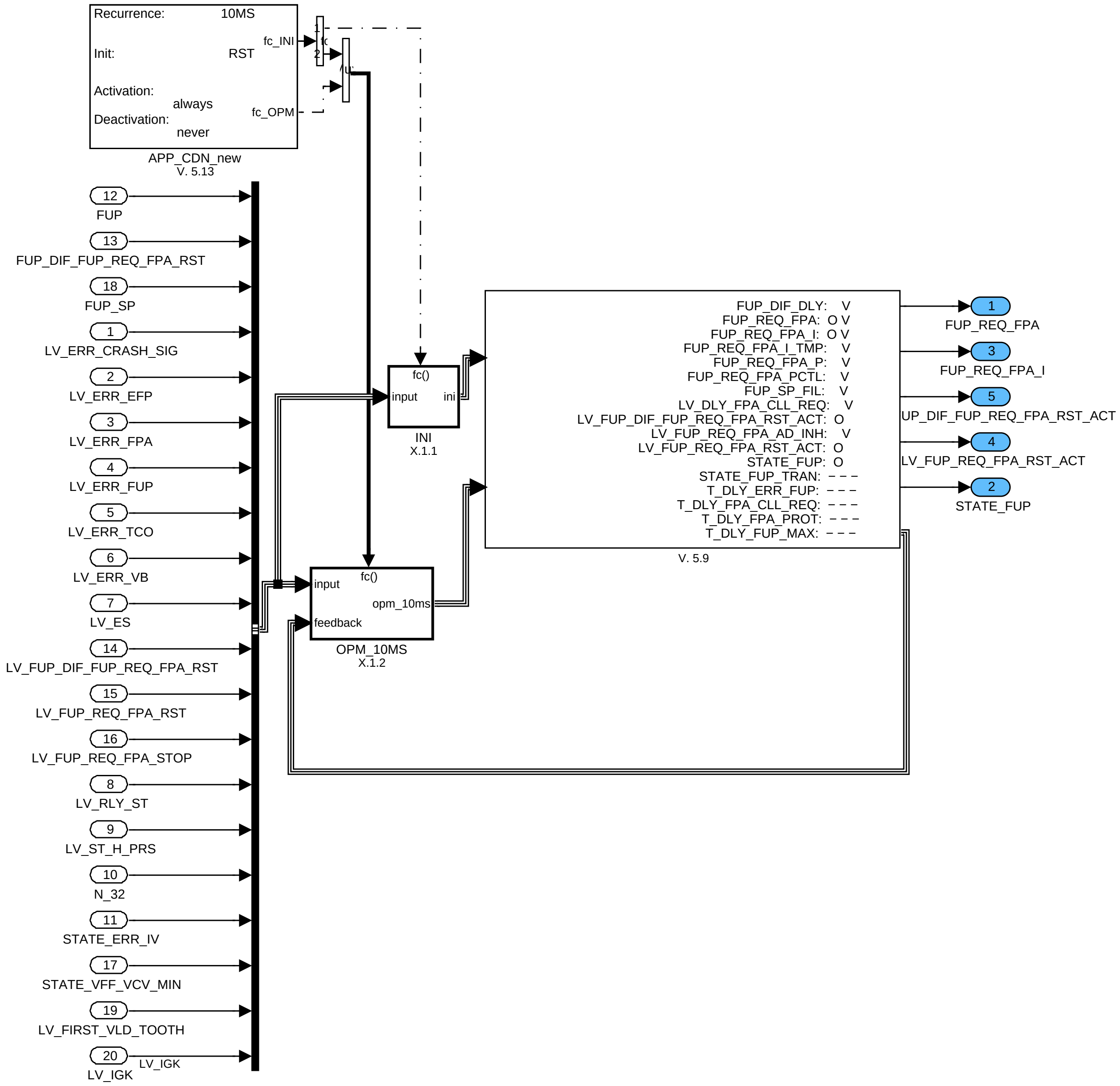
The fuel pressure control (Appl. Inc.) provides a fuel pressure demand for the regulator. It is depending on fuel pressure setpoint and the fuel pressure. The structure is constiting out of a feed forward and a controller part.



Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: 1
Deactivation: 0

Function description



SDA_SRS / SDA V 5.0.6 30−May−2007

Figure 157:
Path: FUSL_M903W

34.14.1 INITILISATION of Variables at Reset

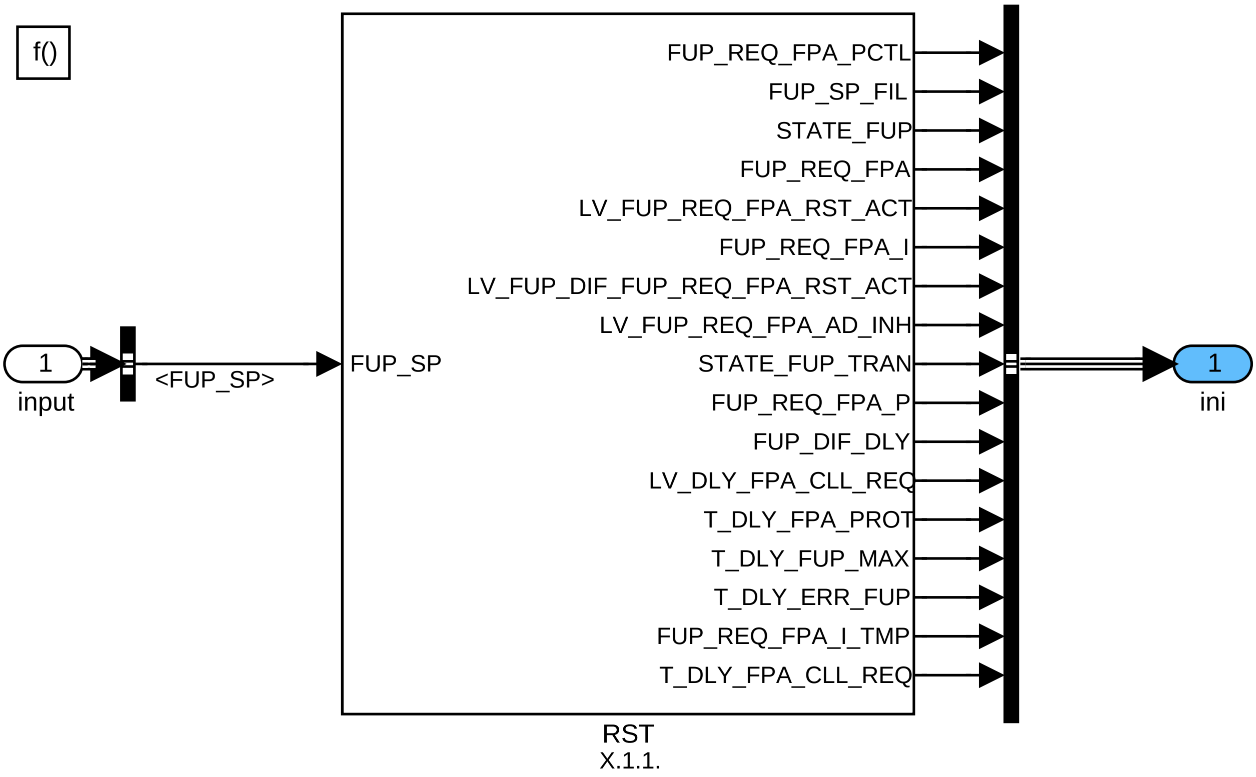


Figure 158:
Path: FUSL_M903W/INI

34.14.1.1 Initilisation

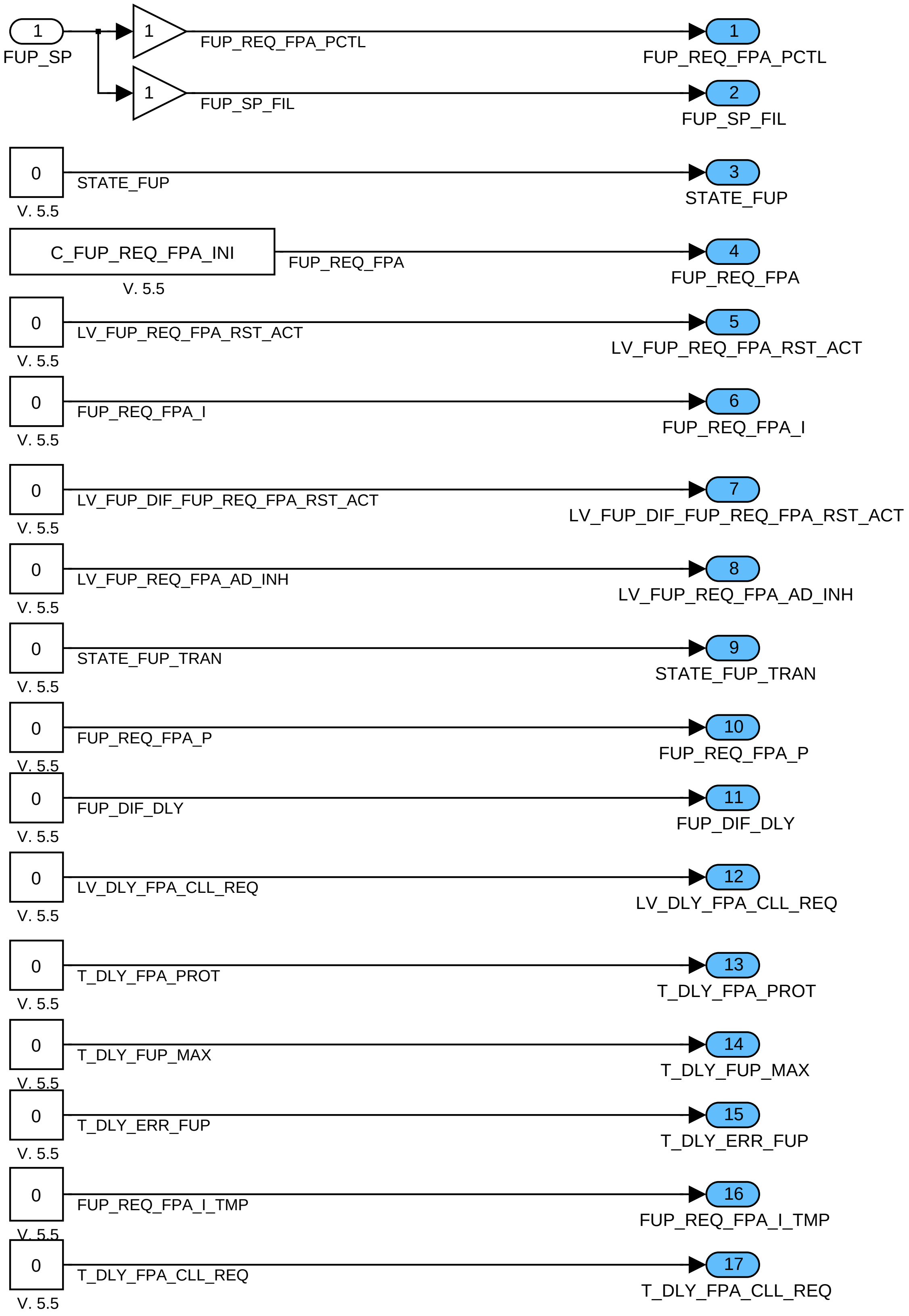


Figure 159:
Path: FUSL_M903W/INI/RST

34.14.2 Formula Section

The functionality is consiting mainly out of transistion calculation part and a controller part.

The transistion are calculated out of several external inputs and the controller is taken into account this external influences.

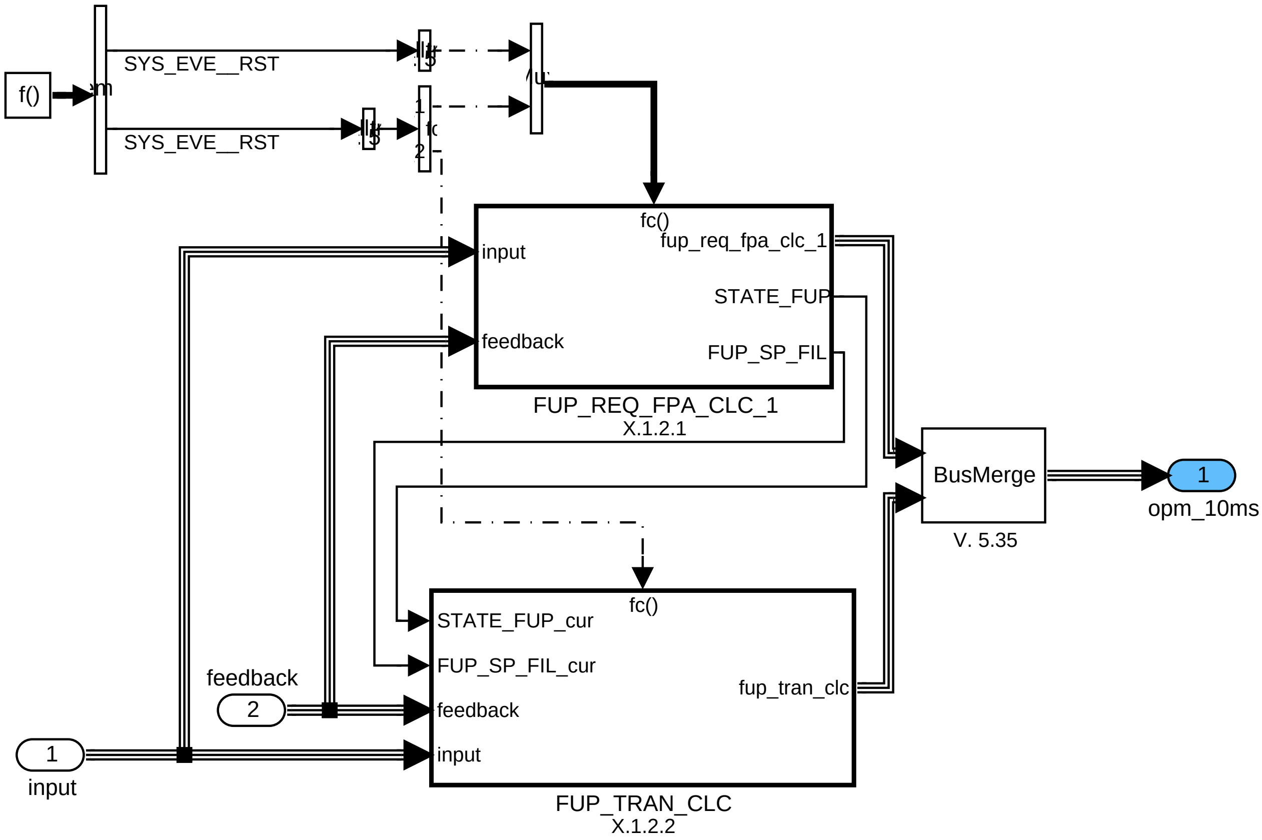


Figure 160:
Path: FUSL_M903W/OPM_10MS

34.14.2.1 Calculation

Out of the result of the state diagram a certain PWM is calculated within that subblocks.

The state diagram is corresponding to the different states of the controller. All the transition conditions are defined in the application incidence of fuel pressure control.

In Low pressure state the FPAPWM signal has minimum value. Also all the initialization is done in this state. Then in order to reach the fuel pressure setpoint the transition is done to Fast pressure increase . In this state the regulator has its maximum FPAPWM value in order to ensure the maximum pressure increase. In the state Pressure control the feed forward controller and the PIcontroller are active. The state Overpressure is selected when the pressure in the rail is too high. Here the FPAPWM signal has minimum value. In the state Limp home the control loop is opened and only the feed forward controller is active.

When a low pressure start is required the controller starts in Low pressure state until STATE_FUP_TRAN = 2. This variable is defined in application incidences. After the start phase the fuel pressure should reach the setpoint in state Fast pressure increase . After fulfilling STATE_FUP_TRAN = 3 (setpoint is reached) the state Pressure control will be entered.

In case of a high pressure start the controller remains in Low pressure state until STATE_FUP_TRAN = 2 (starter relay is activated). This is done for security reasons in order to protect the regulator for overheating. In state Fast pressure increase the fasted pressure increase is performed. After fulfilling STATE_FUP_TRAN = 3 (setpoint is reached) the state Pressure control will be activated.

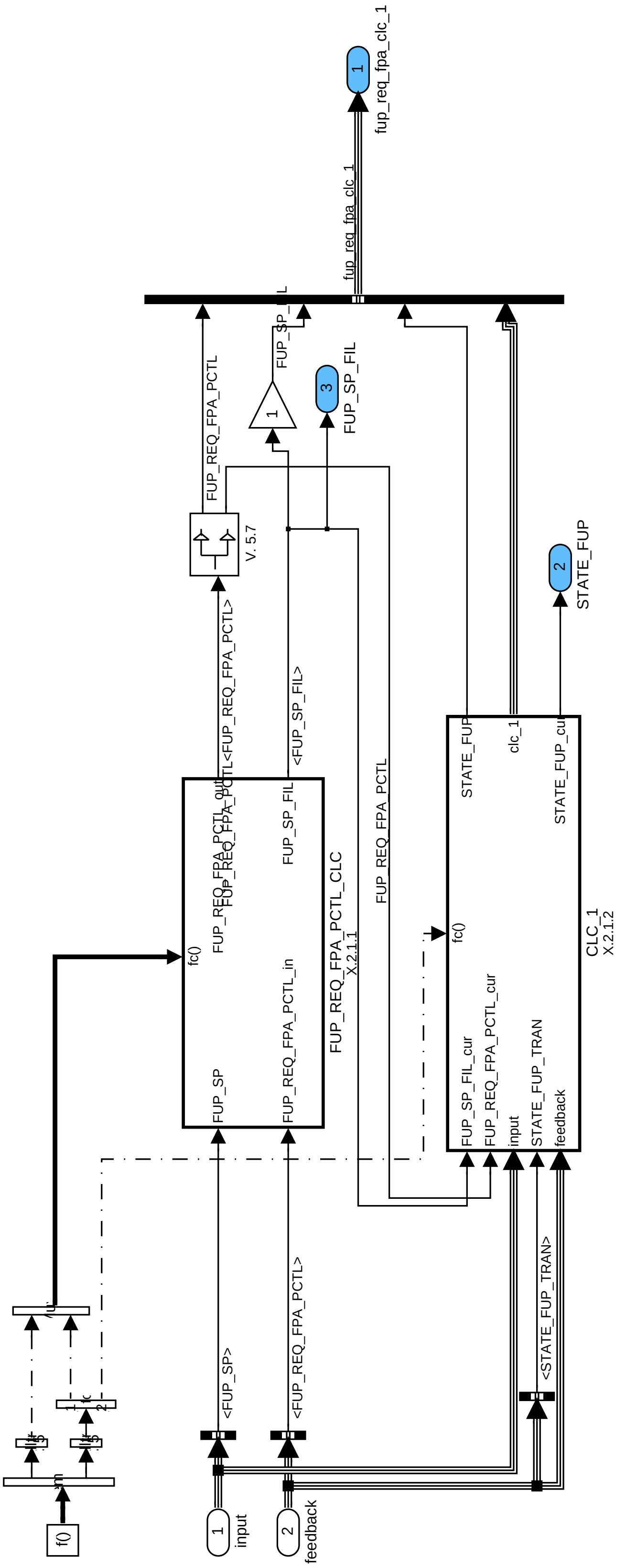


Figure 161:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1

34.14.2.1.1 FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1

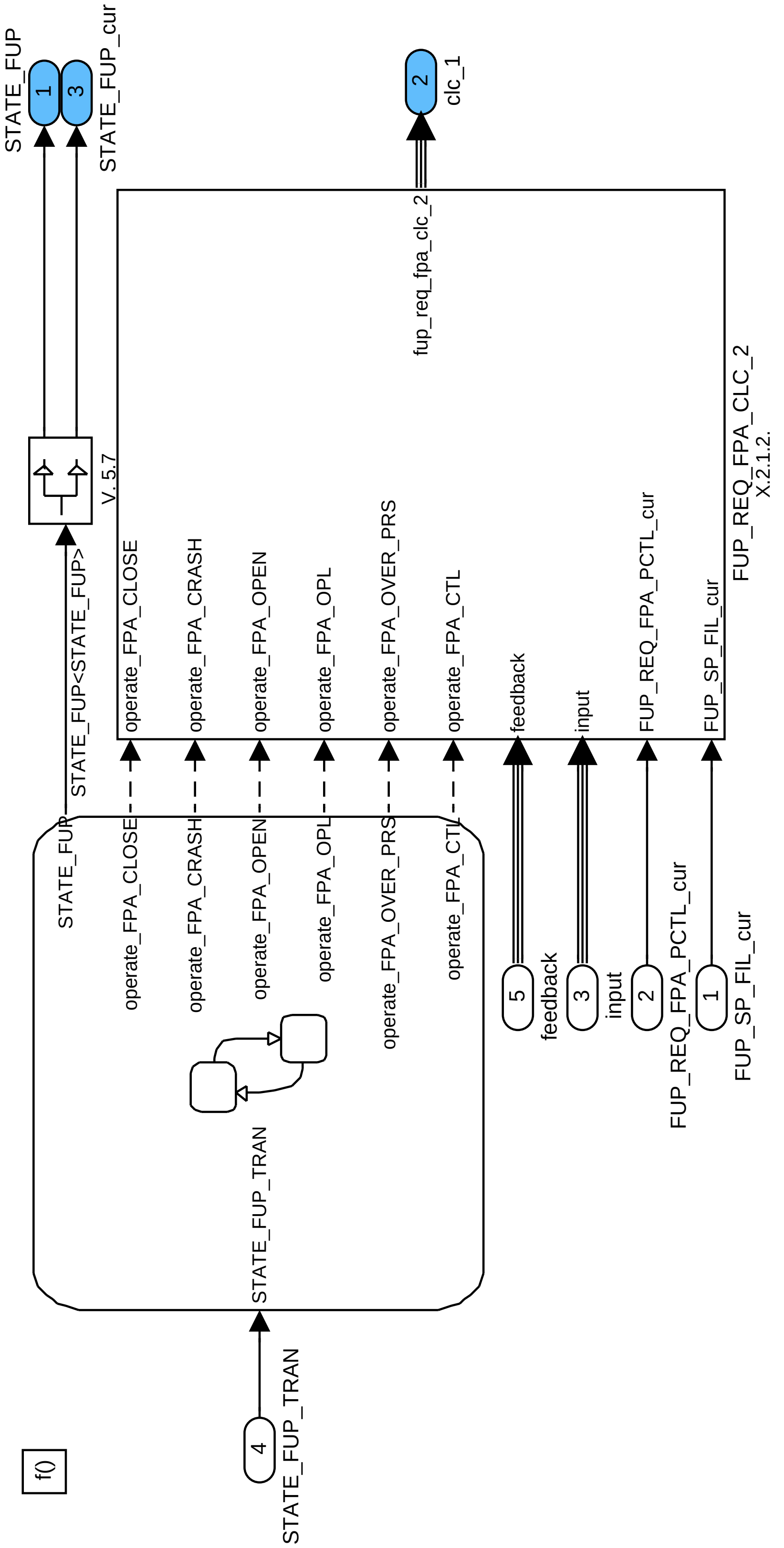


Figure 162:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1

34.14.2.1.1.1 Chart

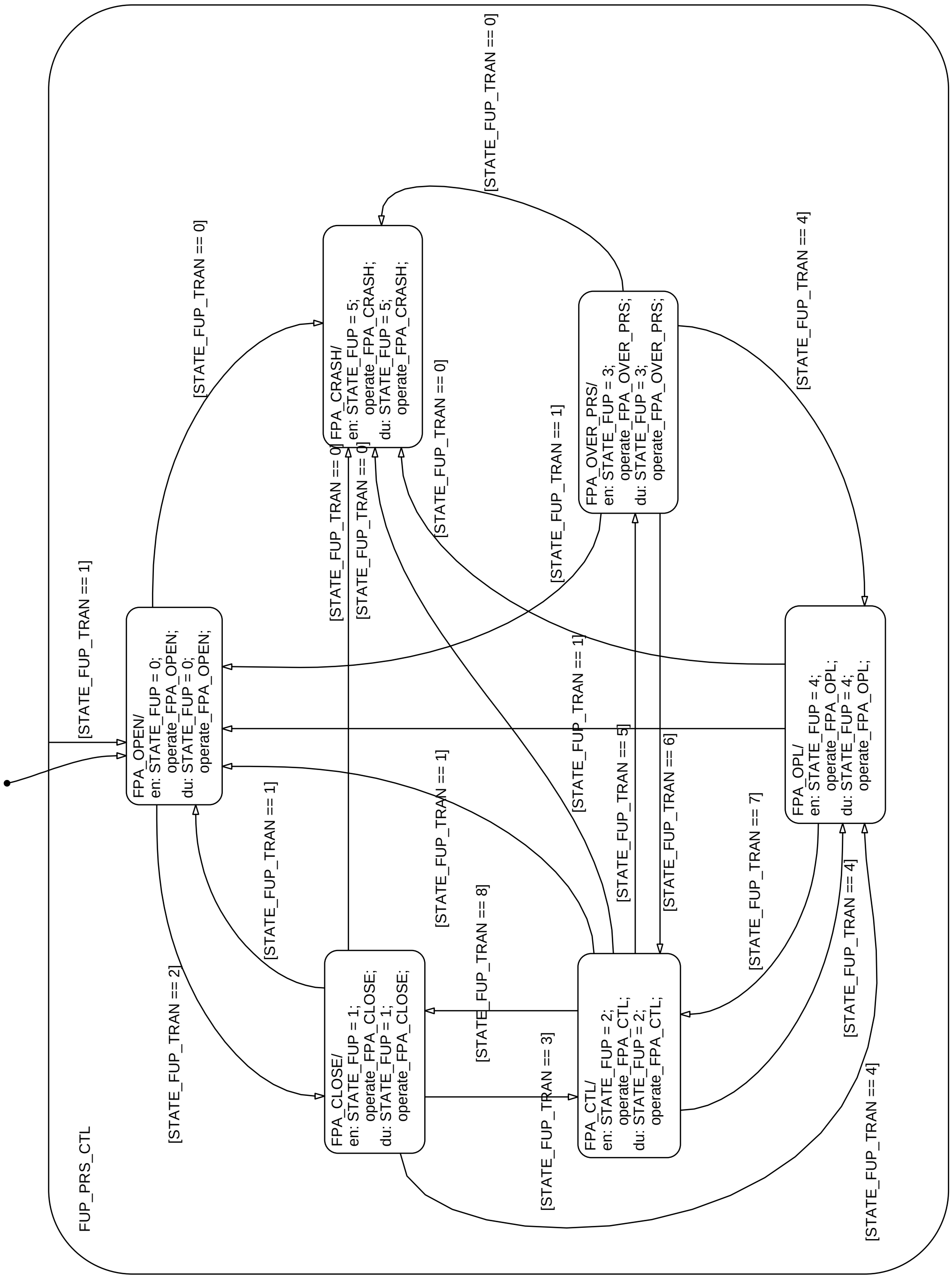


Figure 163:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/Chart

34.14.2.1.1.2 Calculation of Variables based on State Variables

Within that blocks the different signals are calculated depending on the state of the state diagram

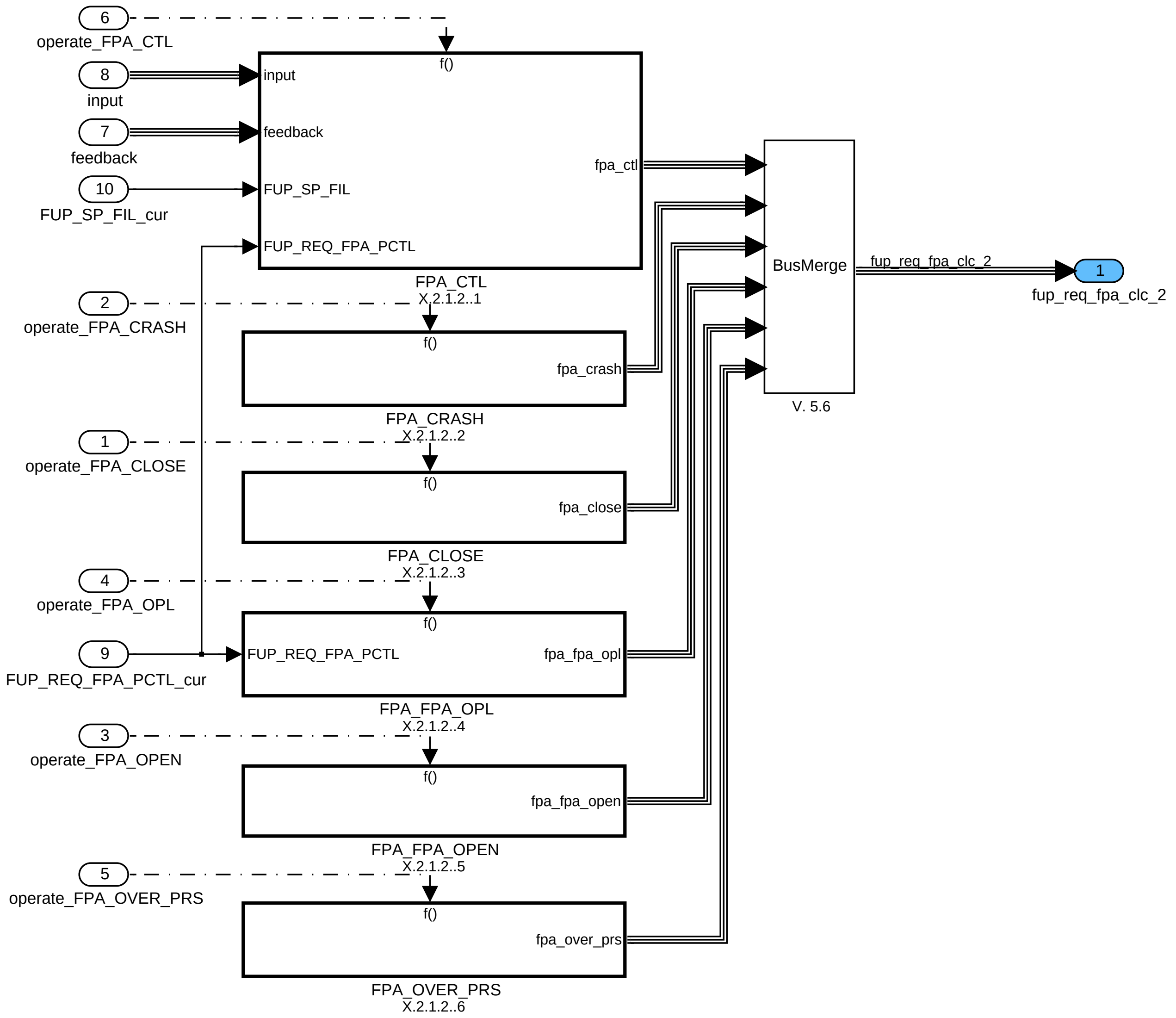


Figure 164:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2

34.14.2.1.1.2.1 Calculation at FPA_CLOSE

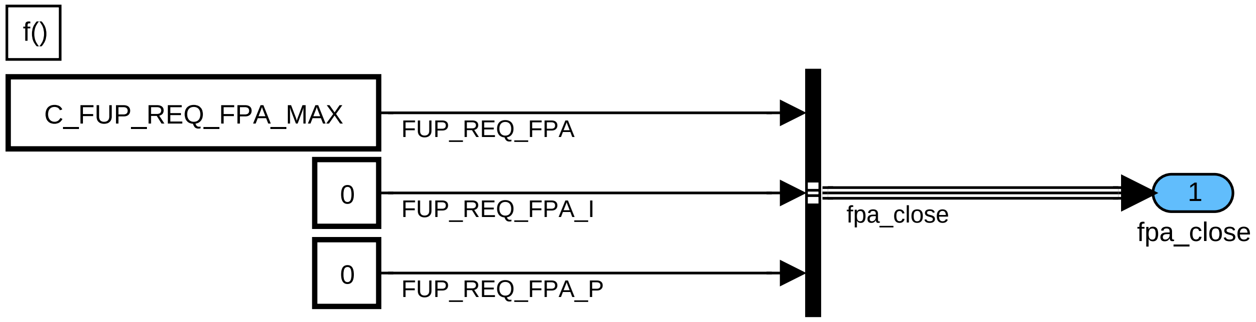


Figure 165:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2/FPA_CLOSE

34.14.2.1.1.2.2 Calculation at FUP_CRASH

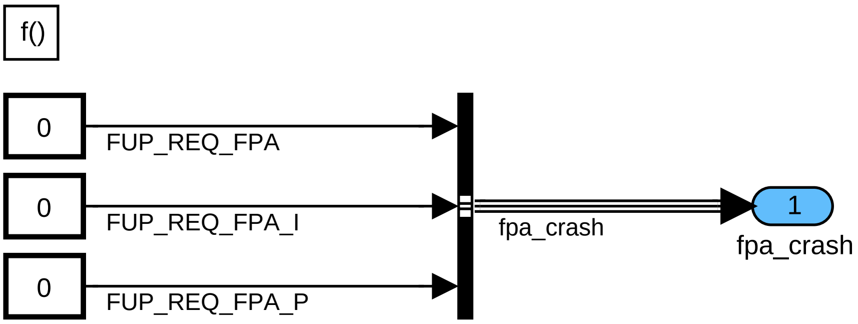


Figure 166:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2/FPA_CRASH

34.14.2.1.1.2.3 Calculation

Withing that two blocks the adaptation and the fuel pressure request for the FPA are calculated.

The calculation of the I and Ppart is done with the help of the fuel pressure deviation and two maps. The Ipart is not allowed to be outside of certain limits. Out of the precontrol, Ipart and Ppart the final fuel pressure demand for the high pressure pump control is generated.

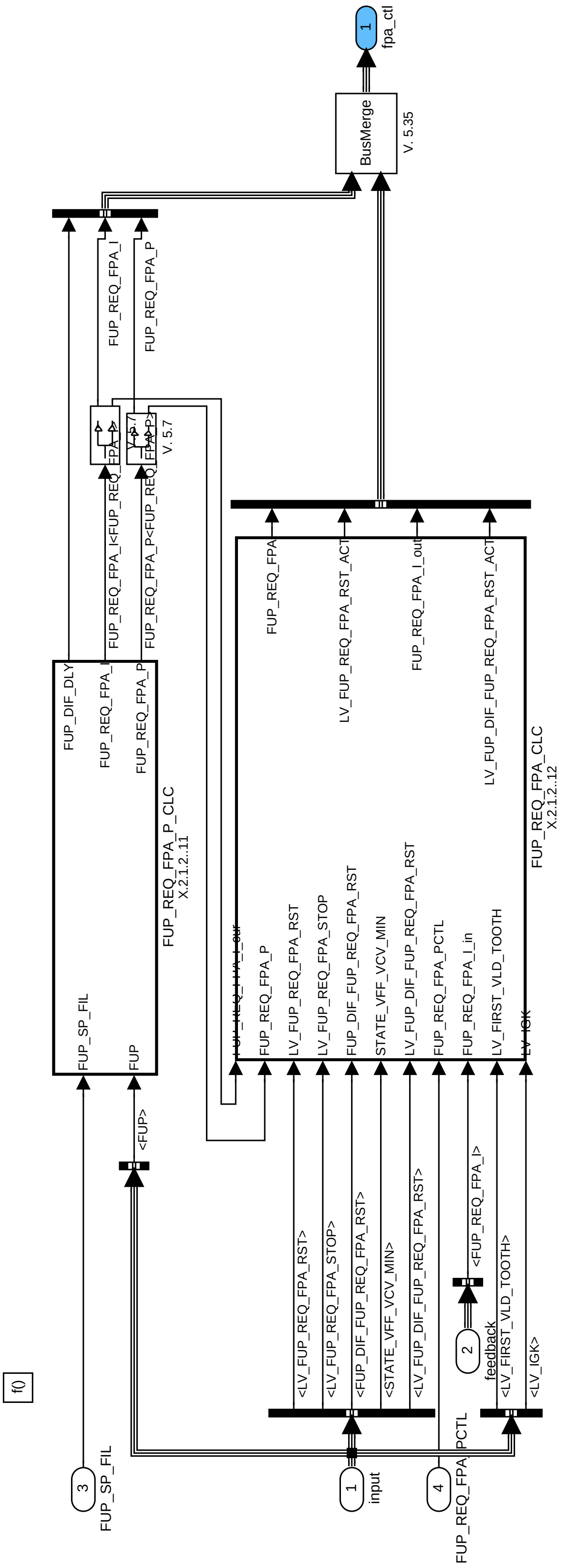


Figure 167:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2/FPA_CTL

34.14.2.1.1.2.3.1 Calculation of FUP and LV variables

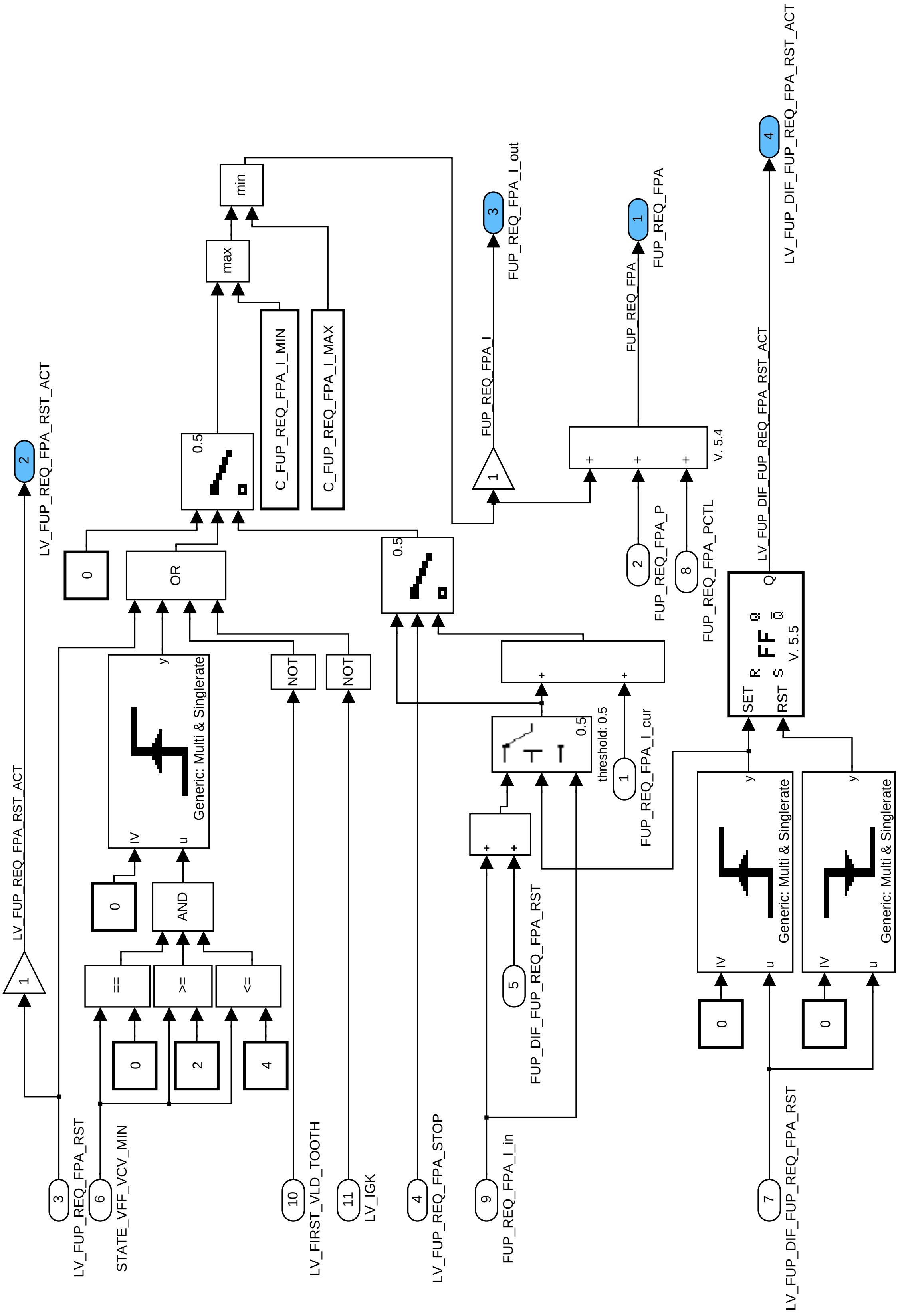


Figure 168:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2/FPA_CTL/FUP_REQ_FPA_CLC

34.14.2.1.1.2.3.2 Calculation of FUP variables

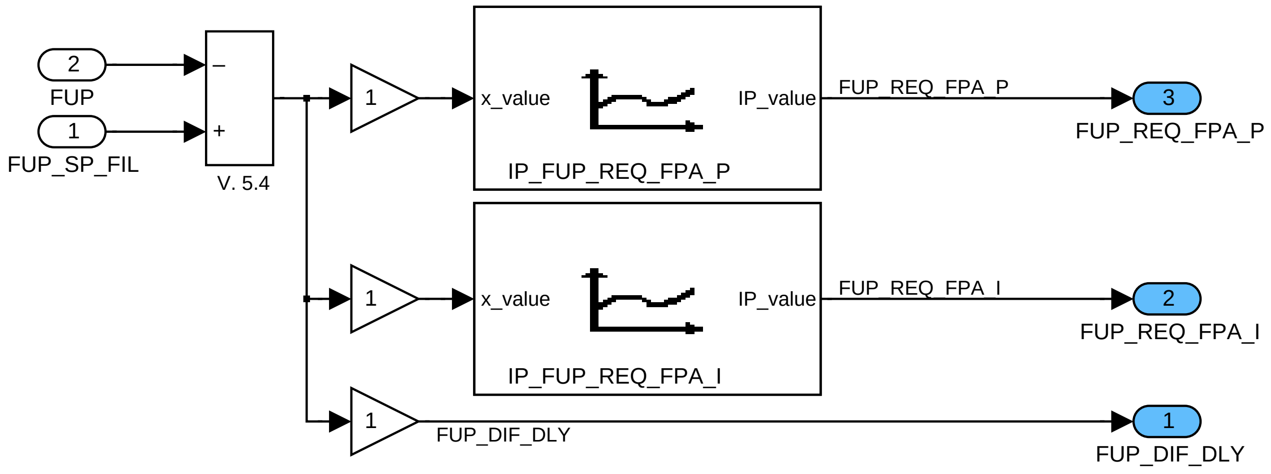


Figure 169:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2/FPA_CTL/FUP_REQ_FPA_P_CLC

34.14.2.1.1.2.4 Calculation at FPA_FPA_OPEN

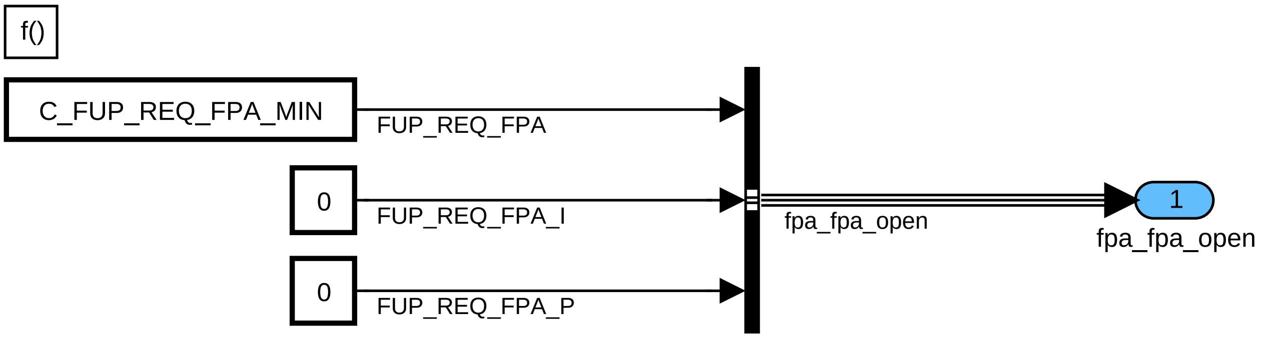


Figure 170:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2/FPA_FPA_OPEN

34.14.2.1.1.2.5 Calculation at FPA_FPA_OPL

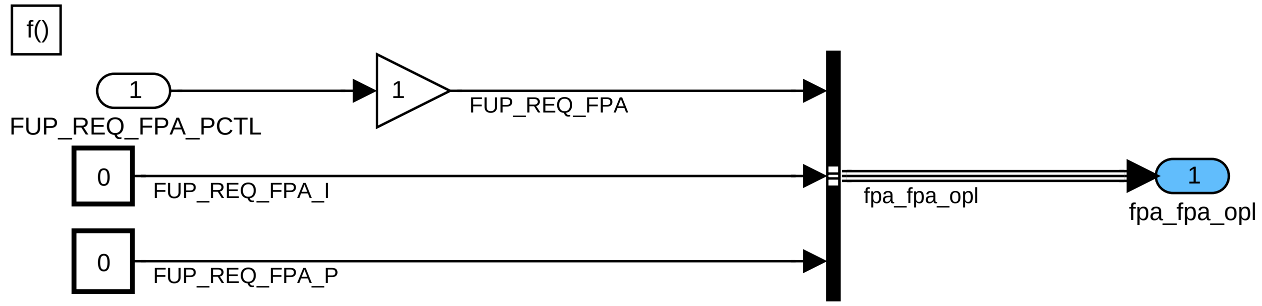


Figure 171:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2/FPA_FPA_OPL

34.14.2.1.1.2.6 Calculation at FPA_OVER_PRS

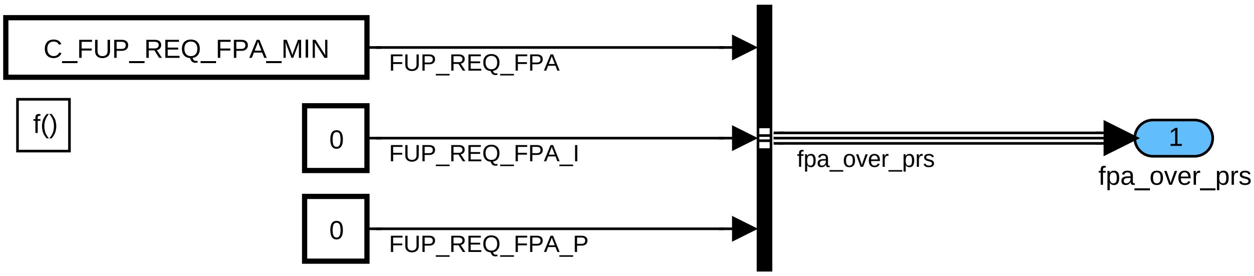
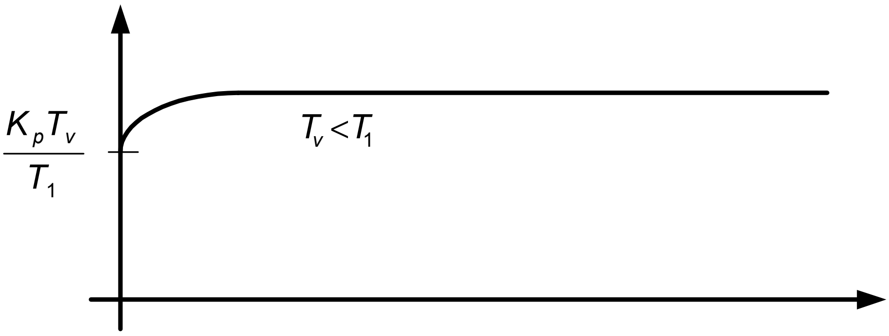


Figure 172:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/CLC_1/FUP_REQ_FPA_CLC_2/FPA_OVER_PRS

34.14.2.1.2 Set point calculation

Within this subblock the filtered fuel pressure setpoint and the precontroller is calculated.

The aim of the precontroller is to prevent overshoots or undershoots of the fuel pressure at big fuel pressure setpoint slopes. This whole block is basically representing a PPT1block.



Source: PPT1-response.vsd

The factor KpTv/T1 ist represented by FAC_FUP_SP_PCTL. This factor is depending on the FUP_SP change. This factor will be multiplicated with FUP_SP, which is the basis for the FPAPWM value. The FUP_SP value is also filtered with a modified PT1filter.

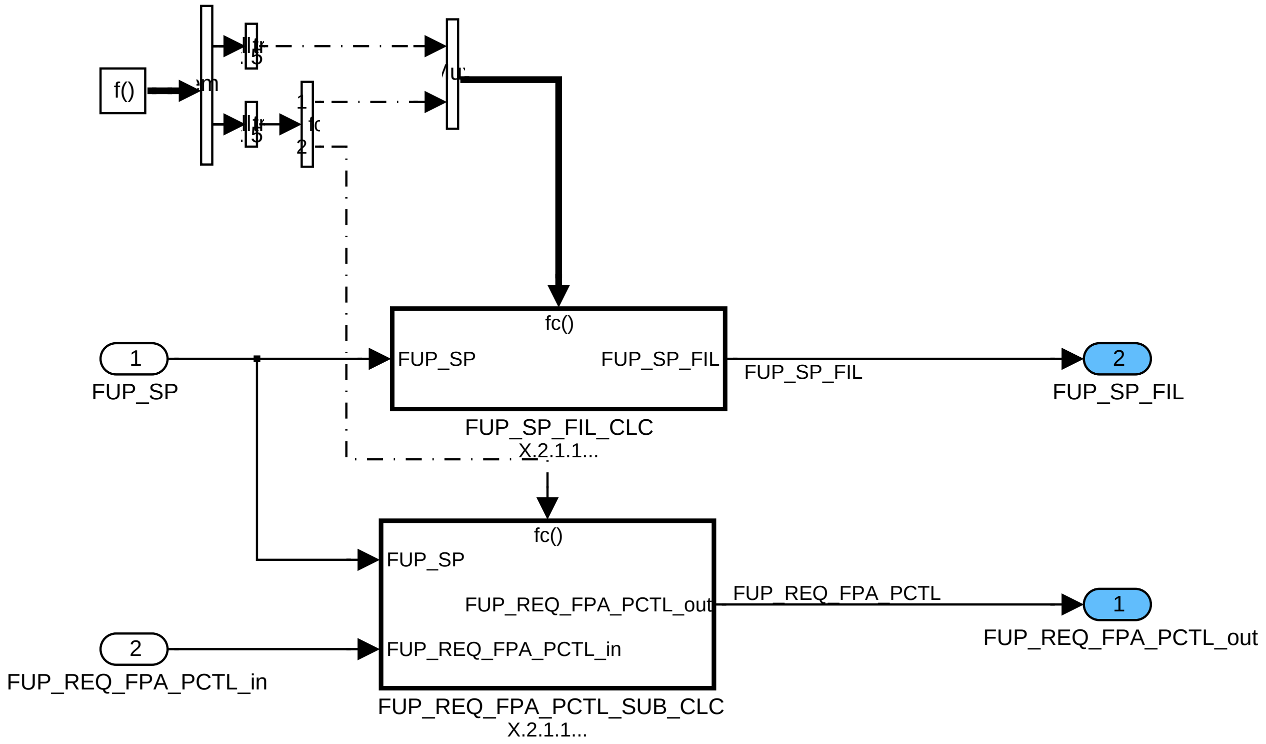


Figure 173:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/FUP_REQ_FPA_PCTL_CLC

34.14.2.1.2.1 Calculation

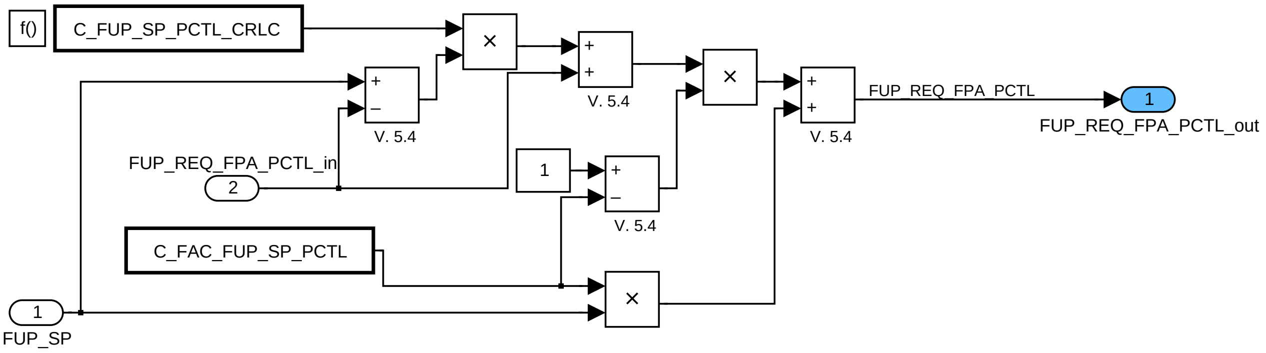


Figure 174:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/FUP_REQ_FPA_PCTL_CLC/FUP_REQ_FPA_PCTL_SUB_CLC

34.14.2.1.2.2 Filter

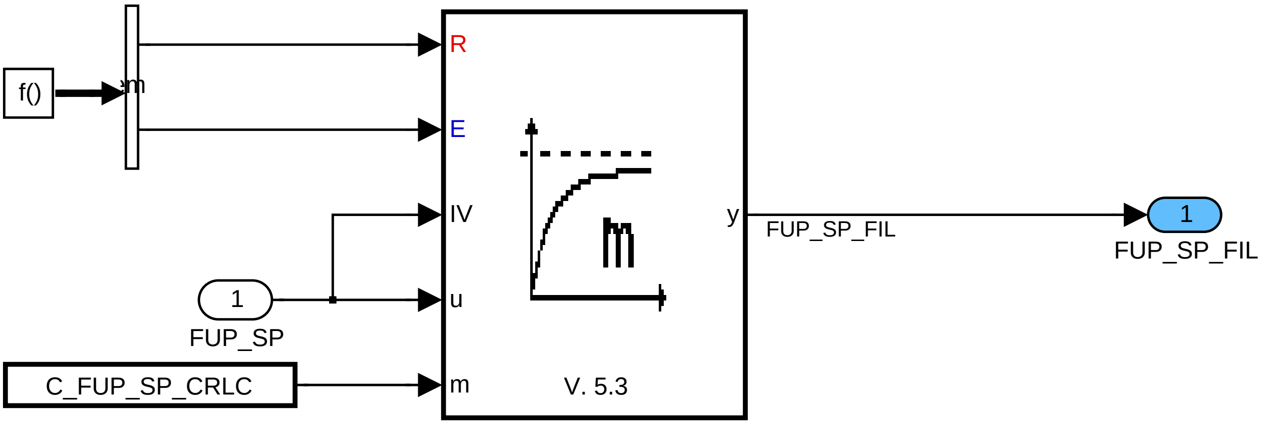


Figure 175:
Path: FUSL_M903W/OPM_10MS/FUP_REQ_FPA_CLC_1/FUP_REQ_FPA_PCTL_CLC/FUP_SP_FIL_CLC

34.14.2.2 Calculation of FUP_TRAN_CLC

In this overview the calculation blocks of the transitions, the adaptation inhibition and the different timers for component protections are listed

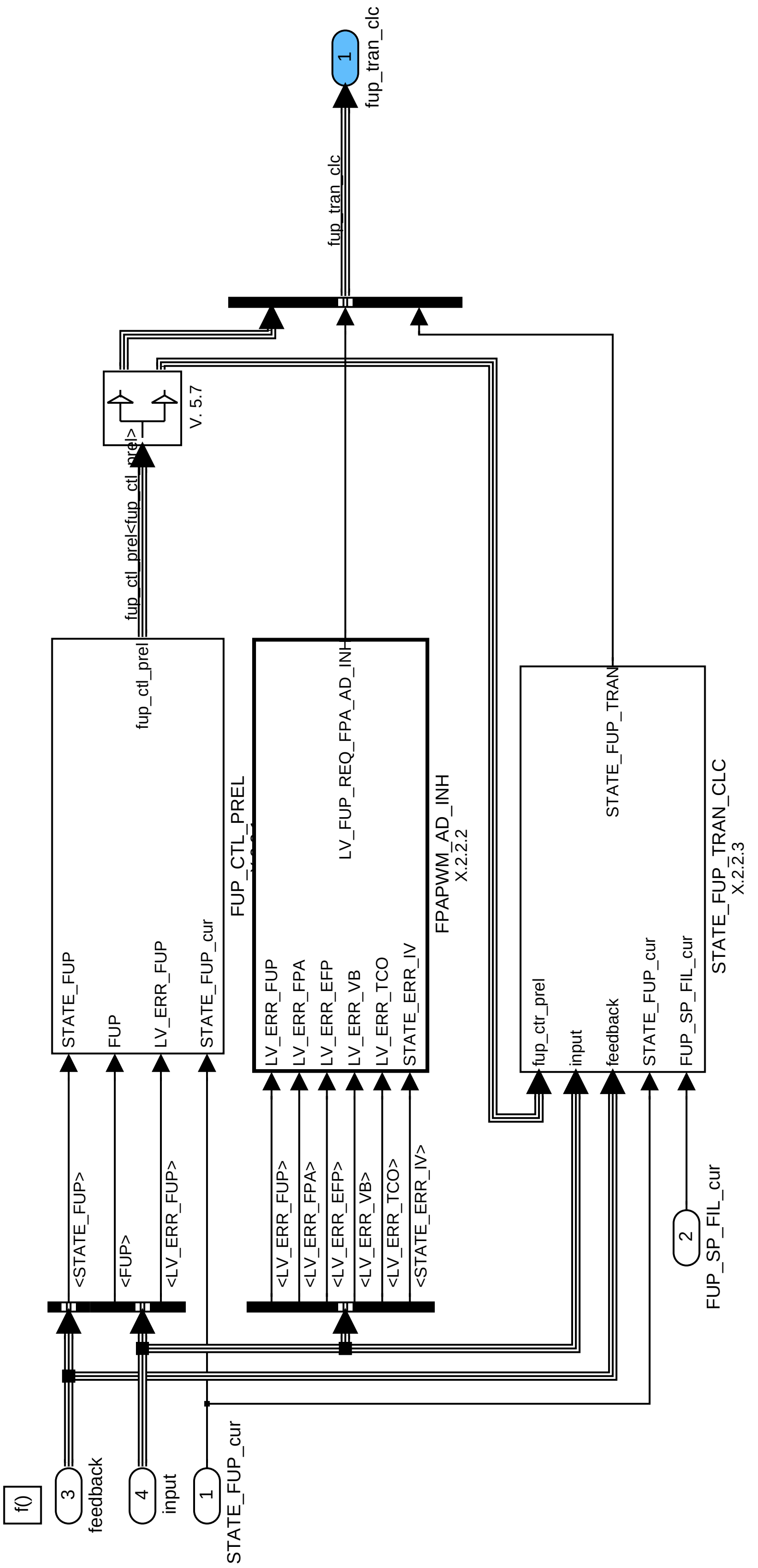


Figure 176:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC

34.14.2.2.1 Calculation of FPAPWM_AD_INH

Several external errors can forbid the adaptation

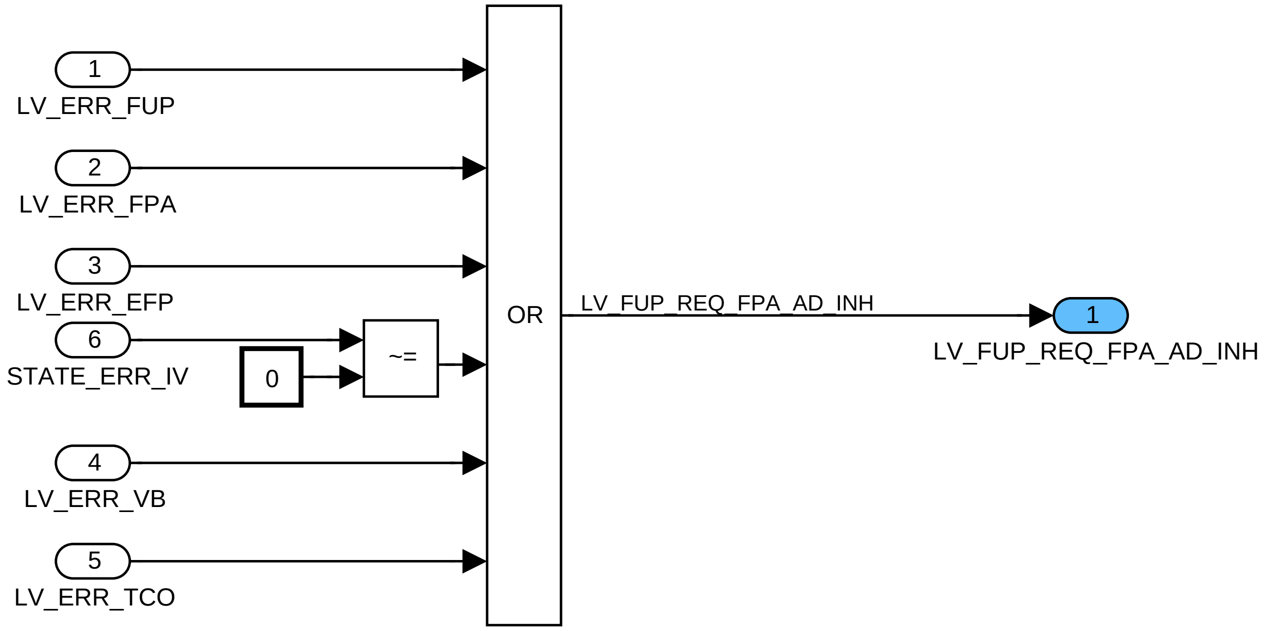


Figure 177:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/FPAPWM_AD_INH

34.14.2.2.2 Timer Calculation

The different timers for the component protection are calculated within that subblock

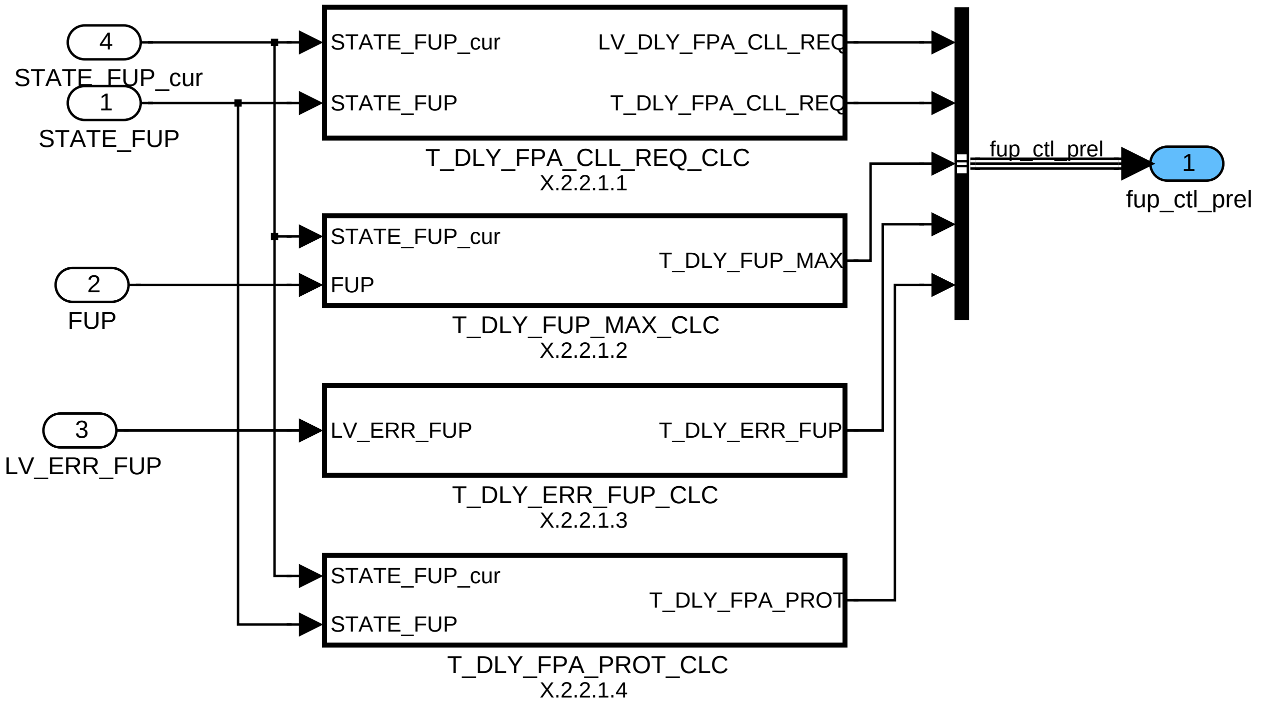


Figure 178:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/FUP_CTL_PREL

34.14.2.2.2.1 Timer Calculation

In case of a fuel pressure error a timer is started

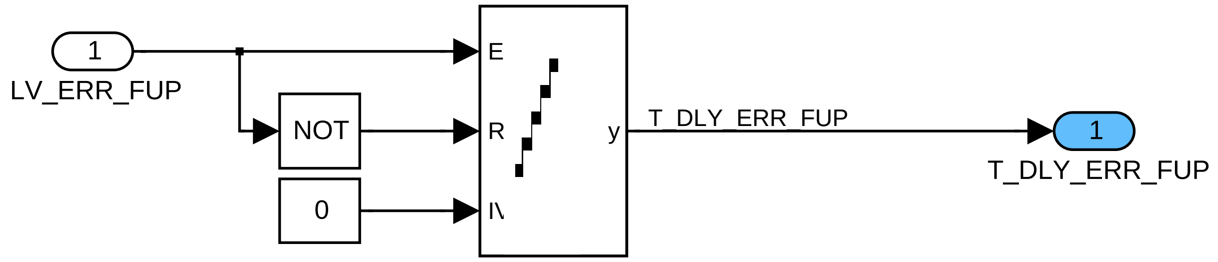


Figure 179:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/FUP_CTL_PREL/T_DLY_ERR_FUP_CLC

34.14.2.2.2.2 Calculation of Timer and LV variales

For the start a higher PWM than allowed can be requested to enable a fast pressure increase. This timer prevents a severe component damage due to overheating
Note: Due to the functionality high pressure pump control the regulator can maybe not respond to that strategy!

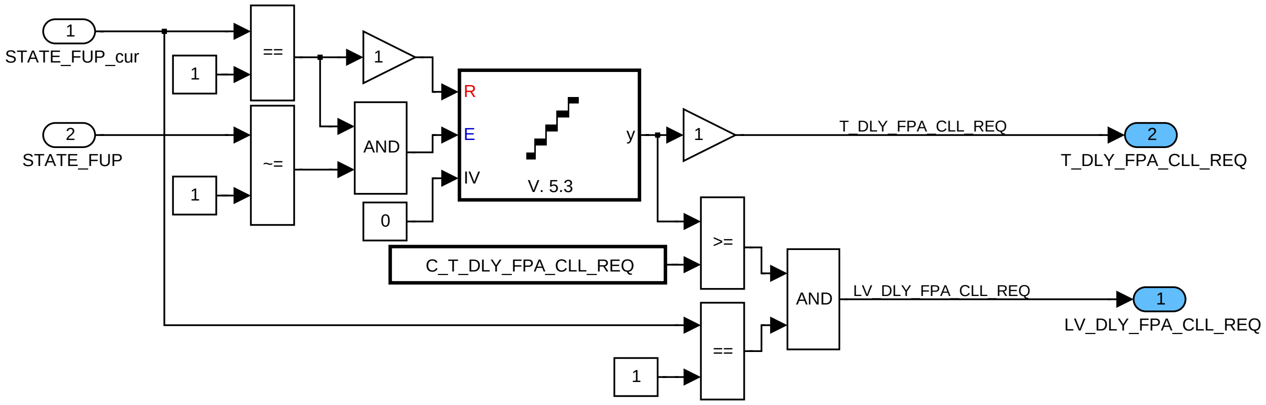


Figure 180:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/FUP_CTL_PREL/T_DLY_FPA_CLL_REQ_CLC

34.14.2.2.2.3 Timer Calculation

Corresponding to the timer before a certain time has to be waited until reaching high PWM again.
Note: Due to the functionality high pressure pump control the regulator can maybe not respond to that strategy!

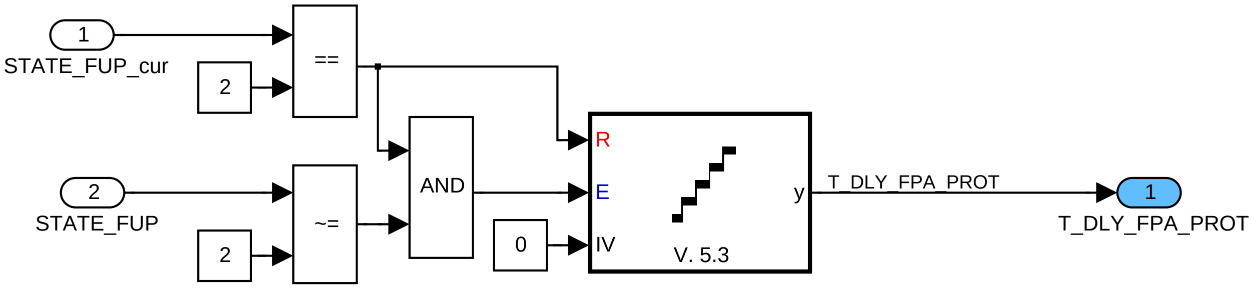


Figure 181:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/FUP_CTL_PREL/T_DLY_FPA_PROT_CLC

34.14.2.2.2.4 Timer Calculation

This timer is observing to overpressure and gives signals to the state diagram.

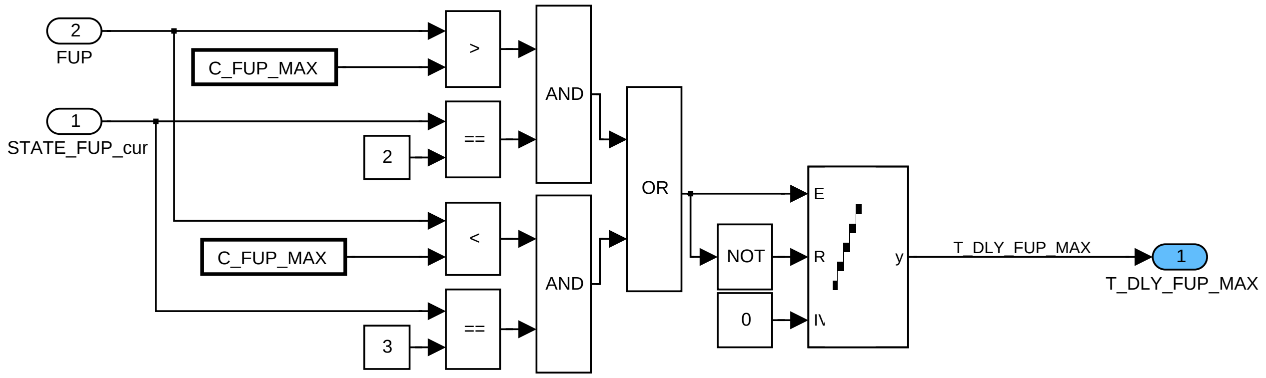


Figure 182:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/FUP_CTL_PREL/T_DLY_FUP_MAX_CLC

34.14.2.2.3 Calculation

Due to complexibility of the transition calculation this function part is split into three parts.

In the first part the crash state will be achieved in case of crash event. When the engine stops the low pressure state will be selected. In case of overpressure more than a certain time the overpressure state should be reached. After disappearing of the overpressure more than a certain time this state will be left. In case of fuel pressure sensor failure the open loop state should be selected.

The second part gives possiblities to reach close state. This state might be not valid due to external influences at the actual actuator control done in high pressure pump control.

In the third part mainly start conditions are mentioned. In case of required high pressure start the high request for the FPA control is given as soon as the starter relay has closed. Otherwise a certain engine speed has to be reached.

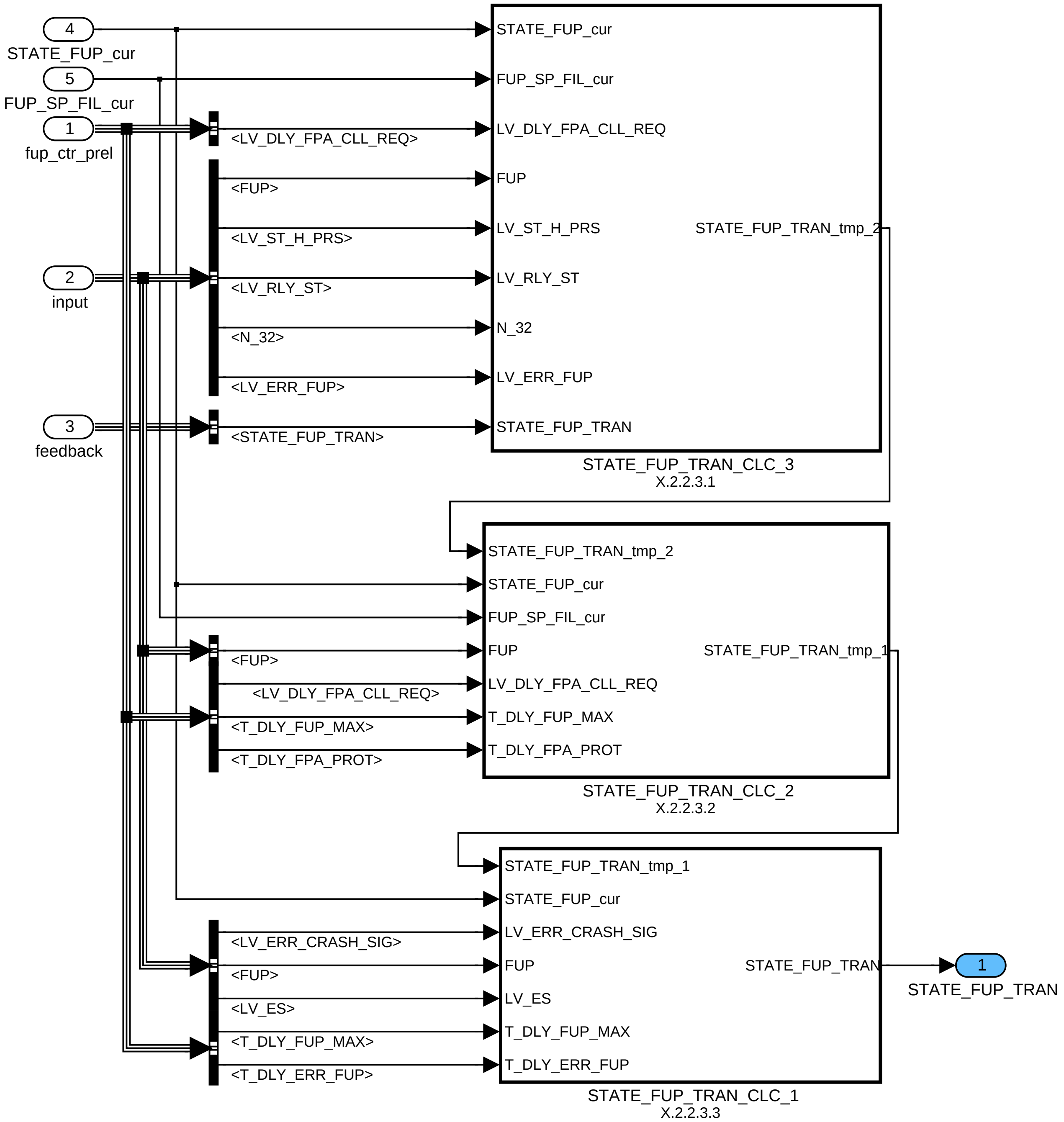


Figure 183:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/STATE_FUP_TRAN_CLC

34.14.2.2.3.1 Calculation at STATE_FUP_TRAN_CLC_1

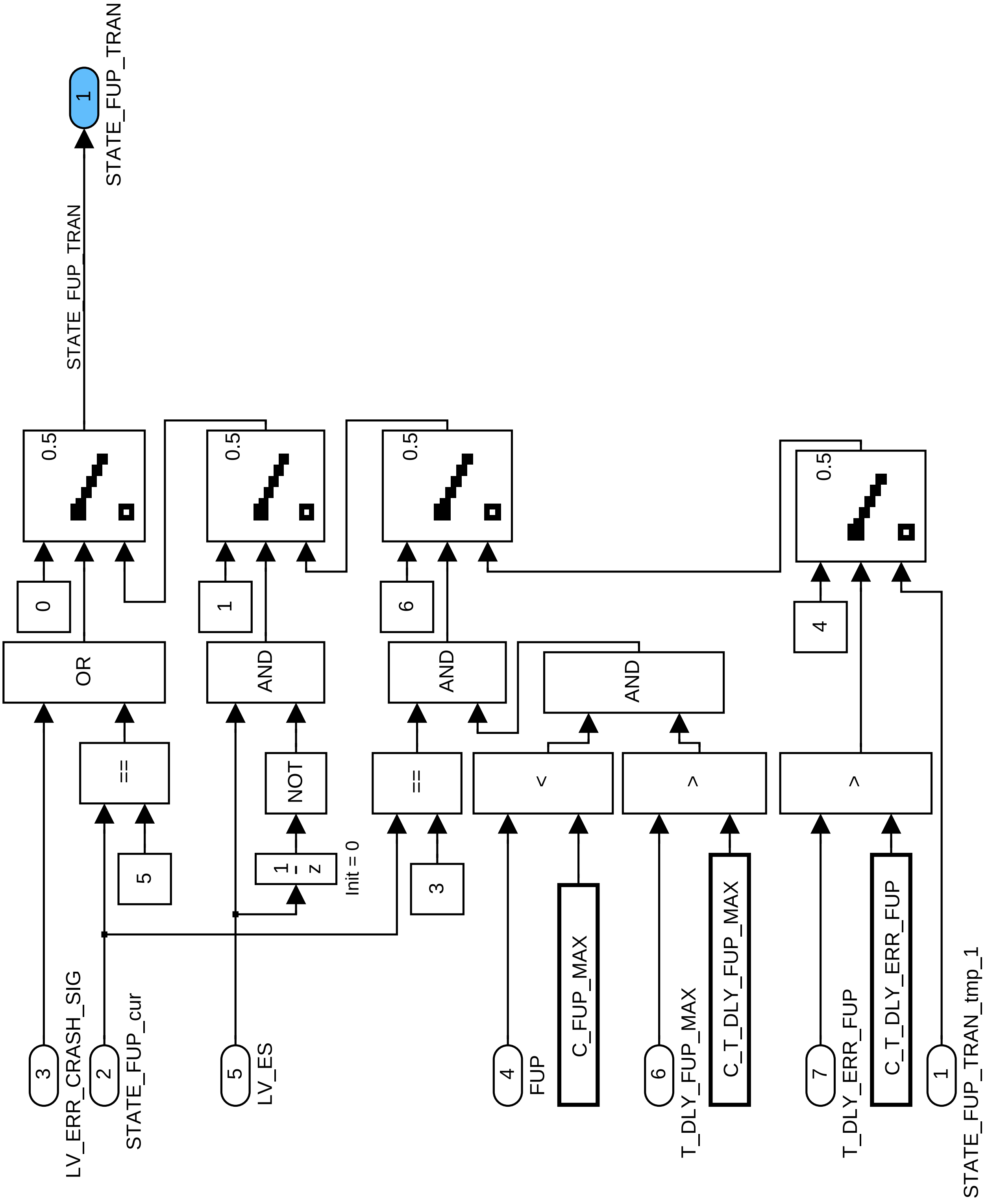


Figure 184:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/STATE_FUP_TRAN_CLC/STATE_FUP_TRAN_CLC_1

34.14.2.2.3.2 Calculation at STATE_FUP_TRAN_CLC_2

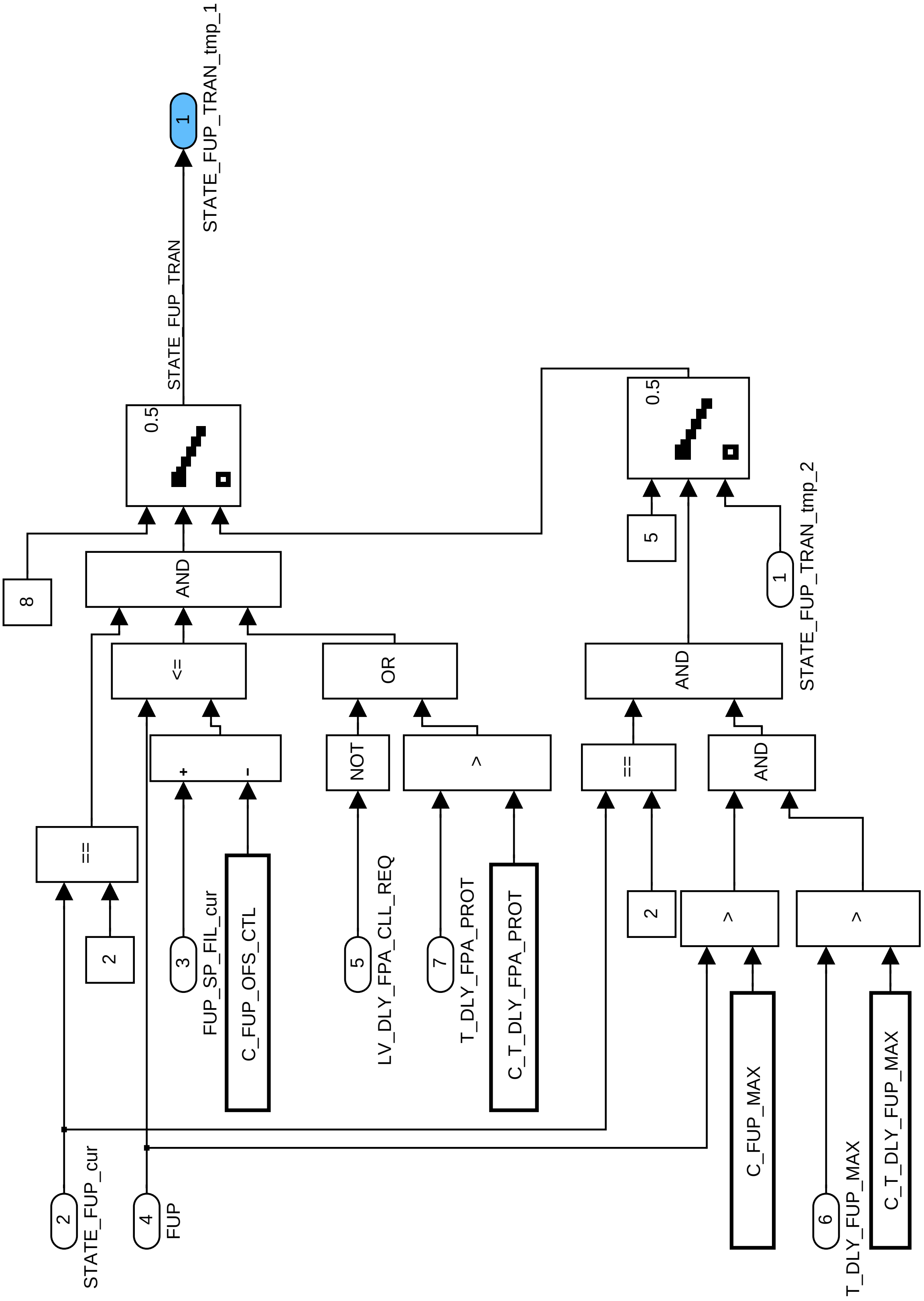


Figure 185:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/STATE_FUP_TRAN_CLC/STATE_FUP_TRAN_CLC_2

34.14.2.2.3.3 Calculation at STATE_FUP_TRAN_CLC_3

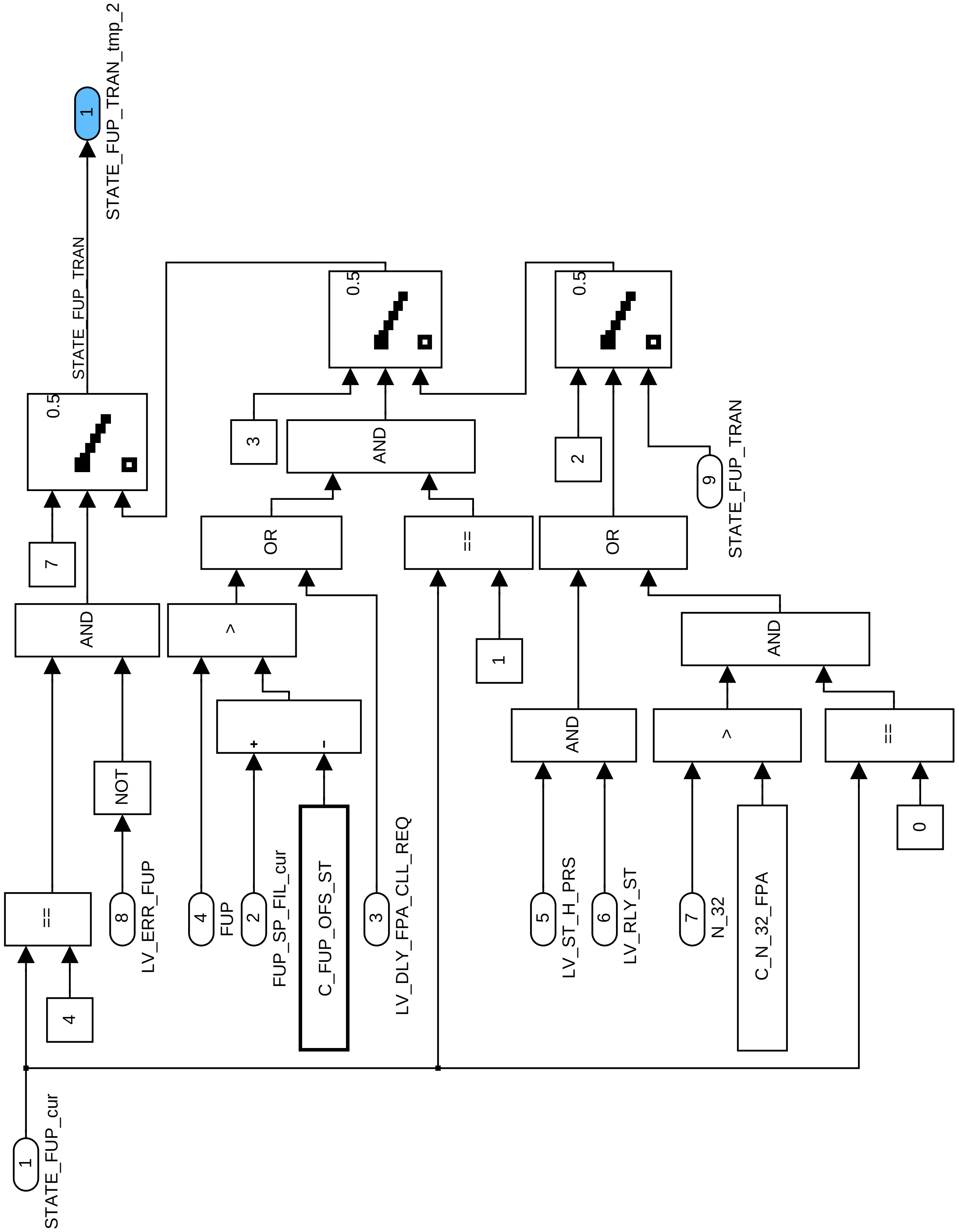


Figure 186:
Path: FUSL_M903W/OPM_10MS/FUP_TRAN_CLC/STATE_FUP_TRAN_CLC/STATE_FUP_TRAN_CLC_3

34.15 High pressure pump control (VCV)

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_CUR_VCV_AST_MOVE_ACT	S	0... FFH	0... 255	1	[-]
Countdown for CUR pulses to move the VCV after engine stop					
CTR_CUR_VCV_MAX	V	0... FFH	0... 255	1	[-]
Counter in Operation max mode					
CTR_MIN_VFF_VCV_MIN_AD_SWI	O/V	0... FFH	0... 255	1	[-]
Variable indicating the cycle of the frequency of the lekage adaptation					
CUR_VCV	V	0... FFFFH	0... 65.535	1e-3	[A]
Actulal current of the VCV					
CUR_VCV_BAS	O/V	0... FFFFH	0... 65.535	1e-3	[A]
Desired value for the current of the VCV					
CUR_VCV_BAS_AD_VAR_2_ADD	O/V/S	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Additive component of second variant of VCV flow adaptation					
CUR_VCV_CTL	V	0... FFFFH	0... 65.535	1e-3	[A]
Requested current for the VCV in CTL mode					
CUR_VCV_DIF	V	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Current difference of VCV current controller					
CUR_VCV_DIF_REL	O/V	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Relative current deviation of VCV current controller					
CUR_VCV_DIF_REL_MMV	V	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Filtered relative deviation of the VCV current controller					
CUR_VCV_MAX	V	0... FFFFH	0... 65.535	1e-3	[A]
Maximum allowed current for the VCV					
CUR_VCV_MIN	V	0... FFFFH	0... 65.535	1e-3	[A]
Requested for the VCV in pressure control mode					
CUR_VCV_MIN_AD	O/V	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Adapted current of VCV in pressure control mode					
CUR_VCV_MIN_AD_1	O/V/S	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Adapted CUR of the first point pressure control part					
CUR_VCV_MIN_AD_2	O/V/S	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Adapted CUR of the second point pressure control part					
CUR_VCV_REQ	V	0... FFFFH	0... 65.535	1e-3	[A]
Requested current for the VCV					
FAC_CUR_VCV_BAS_AD_VAR_2	O/V/S	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Multiplicative component of second variant of VCV flow adaptation					
FAC_CUR_VCV_BAS_AD_VAR_2_TMP	O/V/S	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Temporary variable for the multiplicative component of second variant of VCV flow adaptation					
FAC_CUR_VCV_CTL	V	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Output of VCV current controller					
FAC_CUR_VCV_MIN_AD	V	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[A/hPa]
Factor for the correction of the pressure control part					
FAC_V_SLOP_CUR_VCV	O/V	0... FFFFH	0... 3.99993896484	61.0352e-6	[-]
Slope/ factor value of the VCV current measurement					
FAC_VFF_FPA	V	0... FFH	0... 0.99609375	3.90625e-3	[-]
Factor for the flow through the regulator part of 3W-valve (depending on VFF_VCV_MIN)					
FAC_VFF_VCV_MIN	O/V	0... FFH	0... 9.9609375	0.0390625	[-]

Correction factor for the leakage adaptation					
FRQ_REQ_VCV	O/V	0... 3FFH	0... 1023	1	[Hz]
Required frequency for the volume control valve (typical range 10 - 1023 Hz)					
FUP_CUR_VCV_MIN_AD_1	O/V/S	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure first adaptation point pressure control					
FUP_CUR_VCV_MIN_AD_2	O/V/S	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure second adaptation point pressure control					
FUP_SP_VFF_VCV_MIN_AD_END	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint level for ending the VFF_VCV_MIN adaptation					
FUP_SP_VFF_VCV_MIN_AD_ST	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint level for starting the VFF_VCV_MIN adaptation					
FUP_VFF_VCV_MIN_AD_ST	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Initial fuel pressure where the VCV_MIN calculation starts at					
LV_CUR_VCV_AST_MOVE_ACT	V	0... 1H	0... 1	1	[-]
Logical value for move VCV after engine stop active					
LV_CUR_VCV_CTL_AD_ENA	V	0... 1H	0... 1	1	[-]
Logical value to enable VCV PWM adaption due to CUR CTL					
LV_CUR_VCV_CTL_PRE_RUN	O/V	0... 1H	0... 1	1	[-]
Logical value for VCV_PRE_RUN					
LV_CUR_VCV_CTL_PRE_RUN_ACT	V	0... 1H	0... 1	1	[-]
Logical value for VCV_PRE_RUN active					
LV_CUR_VCV_CTL_PRE_RUN_EDGE	V	0... 1H	0... 1	1	[-]
Logical value for start of VCV_PRE_RUN					
LV_CUR_VCV_ST_ACT	V	0... 1H	0... 1	1	[-]
Logical value for setting of CUR_VCV at start					
LV_FUEL_MASS_CTL_RST_REQ	O/V	0... 1H	0... 1	1	[-]
Logical variable for reseting P and I-part of the controller					
LV_FUEL_MASS_CTL_STOP_REQ	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting freezing the I-part of the controller					
LV_PWL_LOCK_CDN_HPDI	S	0... 1H	0... 1	1	[-]
Power latch extension for CUR pulses to move the VCV after engine stop					
LV_STATE_VFF_VCV_MIN_RST	V	0... 1H	0... 1	1	[-]
Logical bit requesting STATE_VFF_VCV_MIN reset					
LV_VFF_VCV_MIN_AD_ENA	V	0... 1H	0... 1	1	[-]
Adaptation conditions for the minimum flow through the VCV are fullfilled					
PWM_VCV	O/V	0... FFFFH	0... 99.9984741211	1.52588e-3	[%]
PWM signal for the VCV					
PWM_VCV_BAS	V	0... FFFFH	0... 99.998474	1.52588e-3	[%]
Base PWM signal for the VCV					
PWM_VCV_MAX	V	0... FFFFH	0... 99.998474	1.52588e-3	[%]
Maximum PWM signal for the VCV					
STATE_PWM_VCV	O/V	0 1 2 3 4 5 6	VCV_TEST START MFP_CTL VCV_CLOSE VCV_CRASH MFP_MAX VCV_LIH	-	[-]
State diagram for the VCV control					
STATE_PWM_VCV_TRAN	V	0 1 2 3 4 5 6 7	CRASH OPEN CLOSE CONTROL MANUAL MANUAL_OPEN MAXIMUM VCV_LIH	-	[-]
Transition state diagram for the VCV control					

STATE_VFF_VCV_MIN	O/V	0 1 2 3 4 5	PASSIVE READY START ACTIVE END PASSIVE_READY	-	[-]
Minimum VCV adaptation state					
T_CUR_VCV_CTL_PRE_RUN	V	0... FFH	0... 25.5	0.1	[s]
Time for VCV_PRE_RUN active					
T_CUR_VCV_CTL_PRE_RUN_ACT	V	0... FFFFH	0... 655.35	0.01	[s]
Time indicating that VCV pre run is active					
T_VFF_VCV_LIM	V	0... FFFFH	0... 655.35	0.01	[s]
Maximum time for leakage adaption					
T_VFF_VCV_MIN	V	0... FFFFH	0... 655.35	0.01	[s]
Time measured for the pressure increase during adaptation					
T_VFF_VCV_MIN_AD_READY	O/V	0... FFFFH	0... 655.35	0.01	[s]
Time for waiting in the Passive_ready state of the leakage adaptation					
T_VFF_VCV_MIN_AD_ST	O/V	0... FFFFH	0... 655.35	0.01	[s]
Timer to start the pressure increase during adaptation					
V_CUR_VCV_MES	-	0... 7FFFH	0... 4.99984741211	152.588e-6	[V]
Measured voltage of shunt for VCV current controll					
V_CUR_VCV_MES_MV	V	0... 7FFFH	0... 4.99984741211	152.588e-6	[V]
Measured voltage of shunt for VCV current controll					
V_OFS_CUR_VCV	O/V	8000... 7FFFH	-5... 4.99984741211	152.588e-6	[V]
Offset value of the VCV current measurement					
VB_VCV_MMV	V	0... FFH	0... 25.8984375	0.1015625	[V]
Filtered battery voltage for VCV control					
VFF_VCV	O/V	0... FFFFH	0... 255	3.89105e-3	[l/h]
Requested volume fuel flow through the VCV					
VFF_VCV_MIN	O/V	0... FFFFH	0... 255	3.89105e-3	[l/h]
Adapted minimum volume fuel flow through the VCV					
VFF_VCV_MIN_BAS	O/V/S	0... FFFFH	0... 255	3.89105e-3	[l/h]
Adapted minimum volume fuel flow through the VCV (normalized value)					
VFF_VCV_MIN_MMV	V	0... FFFFH	0... 255	3.89105e-3	[l/h]
Adapted minimum volume fuel flow through the VCV after filter					
VFF_VCV_MIN_MMV_INI	V	0... FFFFH	0... 255	3.89105e-3	[l/h]
Adapted minimum volume fuel flow through the VCV (normalized value), Ini value for VFF_VCV_MIN_MMV					

Input Data:

LV_ERR_CRASH_SIG	LV_FIRST_VLD_TOOTH	MFF_SP_FUP_CTL	LV_PUC
T_SEG_AV	VB	TFU	LV_ST_H_PRS
LV_ST_END	N	C_FUP_SP_ADD	TCO_ST
NC_V_CUR_VCV_MES	LV_HPP_CTL_AD_CLR_EXT_REQ	LV_ERR_FUP_STOP	LV_PWL
LV_ES	TCO	C_V_CUR_VCV_BOOT_L	C_V_CUR_VCV_BOOT_L_CMPL
C_V_CUR_VCV_BOOT_H	C_V_CUR_VCV_BOOT_H_CMPL	LV_VCV_RLY	V_CUR_VCV_MES_ACTION_INFR
TECU	LV_IGK	NC_PWL_LOCK_CDN_HPDI	FUP
FAC_CLC_FUEL_MASS_FUP	FUEL_MASS_REQ	FUEL_MASS_REQ_I_CTL_H_RES	FUP_DIF
LV_FUP_LIH_L_PRS_CTL_REQ	LV_FUP_LIH_HOM_REQ	LV_FUP_LIH_HOM_VCV_OPEN_REQ	LV_FUP_PRS_CTL_REQ

VFF_MFF_SP_FUP_CTL	STATE_FUP_CTL	CUR_VCV_BAS_AD_VAR_2_ADD_SET	FUP_SP
LV_FUP_SP_ADD	FUP_REQ_FPA	FUP_REQ_FPA_I	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CRLC_FUP_CUR_VCV_MIN_AD	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlation constant for the MMV calculation of FUP for the adaptation					
C_CRLC_VB_VCV_MMV	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Averaging constant of VB filter for VCV control					
C_CRLC_VFF_VCV_MIN_BAS_MMV	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Filter constant for moving mean value calculation of the VFF_VCV_MIN_BAS					
C_CRLC_VFF_VCV_MIN_MMV	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Filter constant for moving mean value calculation of the VFF_VCV_MIN					
C_CTR_CUR_VCV_AST_MOVE_ACT	1	0... FFH	0... 255	1	[-]
Number of CUR pulses to move the VCV after engine stop					
C_CTR_CUR_VCV_BAS_AD_VAR_2_MAX	1	0... FFH	0... 255	1	[-]
Counter for the second variant of the flow control adaptation					
C_CTR_CUR_VCV_CTL_STOP	1	0... FFFFH	0... 65535	1	[-]
Counter for stopping the VCV current control in intervention mode					
C_CTR_CUR_VCV_MAX	1	0... FFH	0... 255	1	[-]
Calibration for Counter in Operation max mode					
C_CTR_CUR_VCV_MIN_AD_MAX	1	0... FFH	0... 255	1	[-]
Counter for the pressure control part adaptation					
C_CTR_MIN_VFF_VCV_MIN_AD_INI	1	0... FFH	0... 255	1	[-]
Constant for initialisation of counter for frequency of the lekage adaptation					
C_CTR_MIN_VFF_VCV_MIN_AD_SWI	1	0... FFH	0... 255	1	[-]
Constant deciding the frequency of the lekage adaptation					
C_CUR_BAS_CUR_VCV_CTL_MAX	1	0... FFFFH	0... 65.535	1e-3	[A]
Maximum current to enable VCV current controler					
C_CUR_BAS_CUR_VCV_CTL_MIN	1	0... FFFFH	0... 65.535	1e-3	[A]
Minimun current to enable VCV current controler					
C_CUR_VCV_AD_CORD_2	1	0... FFFFH	0... 99.9984e-6	1.52588e-9	[A/hPa]
Correlation constant for transfering the I-part into the PWM_VCV_BAS curve					
C_CUR_VCV_AD_VAR_2_CORD_1	1	0... FFFFH	0... 0.049999237	762.939e-9	[A/mg]
Correlation constant for transfering the I-part into the flow control curve for second variant of the VCV flow adaptation					
C_CUR_VCV_AST_MOVE	1	0... FFFFH	0... 65.535	1e-3	[A]
PWM for moving the VCV after engine stop					
C_CUR_VCV_AST_MOVE_MIN	1	0... FFFFH	0... 65.535	1e-3	[A]
VCV current threshold to activate the VCV after engine stop movement wthout LV_ERR_FUP_STOP					
C_CUR_VCV_BAS_AD_VAR_2_ADD_INI	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Initialisation value for the additive component ot the second variant of the VCV flow adaptation					
C_CUR_VCV_BAS_AD_VAR_2_ADD_MAX	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Maximum value for the additive component in the second variant of the VCV flow adaptation					
C_CUR_VCV_BAS_AD_VAR_2_ADD_MIN	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Minimum value for the additive component in the second variant of the VCV flow adaptation					
C_CUR_VCV_CRASH	1	0... FFFFH	0... 65.535	1e-3	[A]
CUR for VCV in CRASH					
C_CUR_VCV_CTL_AD_MAX	1	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Maximum allowed current deviation to enable PWM adaption					
C_CUR_VCV_DIF_REL_PRE_RUN_MIN	1	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Deviation of the VCV current controler to enable VCV_PRE_RUN					
C_CUR_VCV_LIH	1	0... FFFFH	0... 65.535	1e-3	[A]
CURfor VCV in limphome					

C_CUR_VCV_LIM_INC	1	0... FFFFH	0... 65.535	1e-3	[A]
Gradient limitation for increment of CUR_VCV					
C_CUR_VCV_MAN	1	0... FFFFH	0... 65.535	1e-3	[A]
Manual CURsignal for the VCV					
C_CUR_VCV_MIN_AD_1_INI	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Initialisation value CUR_VCV first point of pressure control part					
C_CUR_VCV_MIN_AD_1_MAX	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Maximum value for the adaptation of point 1 of the pressure control part					
C_CUR_VCV_MIN_AD_1_MIN	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Minimum value for the adaptation of point 1 of the pressure control part					
C_CUR_VCV_MIN_AD_2_INI	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Initialisation value CUR_VCV second point of pressure control part					
C_CUR_VCV_MIN_AD_2_MAX	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Maximum value for the adaptation of point 2 of the pressure control part					
C_CUR_VCV_MIN_AD_2_MIN	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Minimum value for the adaptation of point 2 of the pressure control part					
C_CUR_VCV_MIN_AD_MAX	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Maximum value for the adaptation of the pressure control part					
C_CUR_VCV_MIN_AD_MIN	1	8000... 7FFFH	-32.768 ...32.767	1e-3	[A]
Minimum value for the adaptation of the pressure control part					
C_CUR_VCV_MIN_OPL	1	0... FFFFH	0... 65.535	1e-3	[A]
Minimum CURsignal for the VCV in open loop (at engine stop)					
C_CUR_VCV_MIN_TOL	1	0... FFFFH	0... 65.535	1e-3	[A]
Maximum value for CUR_VCV_MIN					
C_CUR_VCV_PRE_RUN	1	0... FFFFH	0... 65.535	1e-3	[A]
Current for VCV_PRE_RUN					
C_CUR_VCV_SP_MAX	1	0... FFFFH	0... 65.535	1e-3	[A]
Maximum allowed current for VCV					
C_CUR_VCV_VFF_VCV_MIN_AD	1	0... FFFFH	0... 65.535	1e-3	[A]
CUR value for the leakage adaptation					
C_FAC_CUR_VCV_AD_VAR_2_CORD_2	1	1... FFH	392.157e-6... 0.100000035	392.157e-6	[1/mg]
Correlation constant for transfering the I-part into the CUR_VCV_BAS curve for second variant of the VCV flow adaptation					
C_FAC_CUR_VCV_BAS_AD_2_TMP_INI	1	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Initialisation value for the temporary variable for the multiplicative component ot the second variant of the VCV flow adaptation					
C_FAC_CUR_VCV_BAS_AD_VAR_2_MAX	1	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Maximum value for the multiplicative component of the second variant of VCV flow adaptation					
C_FAC_CUR_VCV_BAS_AD_VAR_2_MIN	1	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Minimum value for the multiplicative component of the second variant of VCV flow adaptation					
C_FAC_CUR_VCV_CTL_MAX	1	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Maximum output of VCV current controller					
C_FAC_CUR_VCV_CTL_MIN	1	8000... 7FFFH	-2... 1.99993896484	61.0352e-6	[-]
Minimum output of VCV current controller					
C_FAC_PWM_VCV_MIN	1	0... FFH	0... 9.9609375	0.0390625	[-]
Correction factor for the PWM_VCV_MIN by the flow through the regulator part					
C_FRQ_PWM_VCV_OPL	1	0... 3FFH	0... 1023	1	[Hz]
Required frequency for the VCV for open position (typical 10 - 1023 Hz); Initialisation with 100Hz					
C_FRQ_PWM_VCV_PRE_RUN	1	0... 3FFH	0... 1023	1	[Hz]
Required frequency for the VCV for pre run (typical 10 - 1023 Hz); Initialisation with 100Hz					
C_FRQ_REQ_VCV_AST_MOVE	1	0... 3FFH	0... 1023	1	[Hz]
Required frequency for moving the VCV after engine stop (typical 10 - 1023 Hz)					

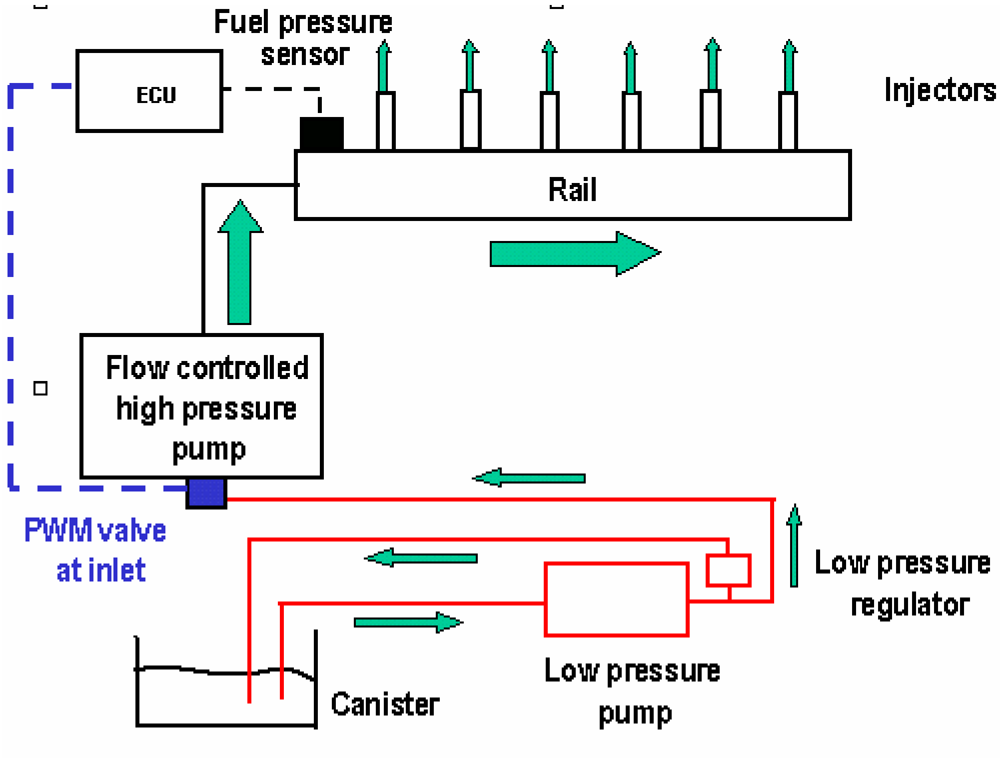
C_FUP_CUR_VCV_MIN_AD_1	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Mean fuel pressure for adaptation of point 1 of pressure control part					
C_FUP_CUR_VCV_MIN_AD_1_INI	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Initialisation value for the fuel pressure first adaptation point pressure control part					
C_FUP_CUR_VCV_MIN_AD_2	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Mean fuel pressure for adaptation of point 2 of pressure control part					
C_FUP_CUR_VCV_MIN_AD_2_INI	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Initialisation value for the fuel pressure second adaptation point pressure control part					
C_FUP_CUR_VCV_MIN_HYS_1	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Hysteresis of fuel pressure for adaptation of point 1 of pressure control part					
C_FUP_CUR_VCV_MIN_HYS_2	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Hysteresis of fuel pressure for adaptation of point 2 of pressure control part					
C_FUP_DIF_CUR_VCV_AD	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Max. fuel pressure deviation to allow adaptation					
C_FUP_DIF_CUR_VCV_AD_VAR_2	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Max. fuel pressure deviation to allow the second variant of the flow adaptation					
C_FUP_DIF_MAX_VFF_VCV_MIN_AD	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Maximum pressure deviation allowed for adaptation of VFF_VCV_MIN					
C_FUP_MAX_PRE_RUN_ACT	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Maximum fuel pressure for activation of prerun					
C_FUP_OFS_MFP_MAX	1	0... FFFFH	-173890 ...173884	5.3066911	[hPa]
FUP offset for high pressure start					
C_FUP_SP_VFF_VCV_MIN_AD_END	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint additional value for ending the VFF_VCV_MIN adaptation					
C_FUP_VCV_AST_MOVE_MAX	1	0... FFFFH	0... 347776	5.3067216	[hPa]
FUP threshold to activate the VCV after engine stop movement wthout LV_ERR_FUP_STOP					
C_FUP_VFF_VCV_MIN_AD_HYS	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure difference for VFF_VCV_MIN adaptation					
C_FUP_VFF_VCV_MIN_AD_RED	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure reduction value due to temperature for VFF_VCV_MIN adaptation					
C_FUP_VFF_VCV_MIN_AD_ST	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure value for VFF_VCV_MIN adaptation start					
C_MFF_SP_VCV_BAS_AD_MAX	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Maximum mass fuel flow to allow the VCV flow adaptation					
C_PWM_VCV_MIN_OPL	1	0... FFFFH	0... 99.998474	1.52588e-3	[%]
Minimum PWM signal for the VCV in open loop (at engine stop)					
C_RHO_FUEL	1	153B... FFFFH	74.6395056077 ...900	0.0137331	[mg/cm**3]
Density of the fuel					
C_T_CUR_VCV_AST_MOVE	1	0... FFFFH	0... 655.35	0.01	[s]
Time for the CUR to move the VCV after engine stop					
C_T_CUR_VCV_CTL_PRE_RUN	1	0... FFH	0... 25.5	0.1	[s]
Time for the VCV_PRE_RUN					
C_T_CUR_VCV_CTL_PRE_RUN_ACT_MIN	1	0... FFFFH	0... 655.35	0.01	[s]
Minimum time for VCV pre run duration					
C_T_DLY_CUR_VCV_AST_MOVE	1	0... FFFFH	0... 655.35	0.01	[s]
Delaytime for the CUR to move the VCV after engine stop					
C_T_FIRST_DLY_CUR_VCV_AST_MOVE	1	0... FFFFH	0... 655.35	0.01	[s]
First delaytime for the CUR to move the VCV after engine stop					
C_T_VFF_VCV_LIM_MAX	1	0... FFFFH	0... 655.35	0.01	[s]
Maximum time for the leakage adaptation					
C_T_VFF_VCV_MIN_AD_READY	1	0... FFFFH	0... 655.35	0.01	[s]
Maximum time for the leakage adaptation					
C_T_VFF_VCV_MIN_AD_ST_MAX	1	0... FFFFH	0... 655.35	0.01	[s]
Time for VFF_VCV_MIN adaptation start					
C_TFU_CUR_VCV_AD_MAX	1	0... FEH	-48... 142.5	0.75	[°C]

Maximum temperature for the adaptation					
C_TFU_CUR_VCV_AD_MIN	1	0... FEH	-48... 142.5	0.75	[°C]
Minimum temperature for the adaptation					
C_TFU_VFF_VCV_MIN_CLC_MAX	1	0... FEH	-48... 142.5	0.75	[°C]
Maximum temperature for the leakage adaptation					
C_TFU_VFF_VCV_MIN_CLC_MIN	1	0... FEH	-48... 142.5	0.75	[°C]
Minimum temperature for the leakage adaptation					
C_V_CUR_VCV_BOOT_H_INI	1	0... 7FFFH	0... 4.99984741211	152.588e-6	[V]
Measured voltage at high current					
C_V_CUR_VCV_BOOT_H_STD	1	0... 7FFFH	0... 4.99984741211	152.588e-6	[V]
Measured voltage at high current standard value					
C_V_CUR_VCV_BOOT_L_INI	1	0... 7FFFH	0... 4.99984741211	152.588e-6	[V]
Measured voltage at low current initial value					
C_V_CUR_VCV_BOOT_L_STD	1	0... 7FFFH	0... 4.99984741211	152.588e-6	[V]
Measured voltage at low current standard value					
C_VB_CUR_VCV_CTL_AD_MAX	1	0... FFH	0... 25.8984375	0.1015625	[V]
Maximum battery voltage to enable PWM adaption					
C_VB_CUR_VCV_CTL_AD_MIN	1	0... FFH	0... 25.8984375	0.1015625	[V]
Minimum battery voltage to enable PWM adaption					
C_VB_CUR_VCV_CTL_PRE_RUN_MAX	1	0... FFH	0... 25.8984375	0.1015625	[V]
Maximum battery voltage to enable VCV_PRE_RUN					
C_VB_CUR_VCV_CTL_PRE_RUN_MIN	1	0... FFH	0... 25.8984375	0.1015625	[V]
Minimum battery voltage to enable VCV_PRE_RUN					
C_VFF_CUR_VCV_BAS_AD_2_BOL_1	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
VFF bottom limit to allow the adaptation of the additive component in the second variant of the VCV flow adaptation					
C_VFF_CUR_VCV_BAS_AD_2_BOL_2	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
VFF bottom limit to allow the adaptation of the multiplicative component in the second variant of the VCV flow adaptation					
C_VFF_CUR_VCV_BAS_AD_2_TOL_1	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
VFF top limit to allow the adaptation of the additive component in the second variant of the VCV flow adaptation					
C_VFF_CUR_VCV_BAS_AD_2_TOL_2	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
VFF top limit to allow the adaptation of the multiplicative component in the second variant of the VCV flow adaptation					
C_VFF_VCV_MIN_BOL	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
VFF_VCV_MIN adaptation bottom value					
C_VFF_VCV_MIN_INI	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
VFF_VCV_MIN_BAS initial value					
C_VFF_VCV_MIN_TOL	1	0... FFFFH	0... 255	3.89105e-3	[l/h]
VFF_VCV_MIN adaptation top value					
ID_CUR_VCV_LIM_DEC	4	0... FFFFH	0... 65.535	1e-3	[A]
LDP_FUP_ID_CUR_VCV_LIM_DEC	4	0... FFFFH	0... 347776	5.3067216	[hPa]
Gradient limitation for decrement of CUR_VCV					
ID_FRQ_PRS_PWM_VCV_STND	8	0... 3FFH	0... 1023	1	[Hz]
LDP_N_ID_FRQ_PRS_PWM_VCV_STND	8	0... 1FE0H	0... 8160	1	[rpm]
Required frequency for the VCV for standard operation in pressure mode					
ID_FRQ_PWM_VCV_STND	8	0... 3FFH	0... 1023	1	[Hz]
LDP_N_ID_FRQ_PRS_PWM_VCV_STND	8	0... 1FE0H	0... 8160	1	[rpm]
Required frequency for the VCV for standard operation (typical range 10 - 1023 Hz)					
IP_CRLC_CUR_VCV_DIF_REL	6	0... FFH	0... 0.99609375	3.90625e-3	[-]
LDP_CUR_DIF_REL_IP_CRLC_CUR	6	0... FFFFH	-32.768 ...32.767	1e-3	[A]
Corelation value for filtering of relative current deviation VCV current controll					
IP_CUR_SP_FLOW_CTL_VCV	16	0... FFFFH	0... 65.535	1e-3	[A]
LDP_VFF_VCV_IP_CUR_SP_FLOW	16	0... FFFFH	0... 255	3.89105e-3	[l/h]
Current setpoint of VCV in flow controll					
IP_CUR_SP_PRS_CTL_VCV	16	0... FFFFH	0... 65.535	1e-3	[A]

LDP_FUP_REQ_FPA_IP_CUR_SP_PRS	16	0... FFFFH	0... 347776	5.3067216	[hPa]
Current setpoint of VCV in pressure controll					
IP_CUR_VCV_MES	6	0... FFFFH	0... 65.535	1e-3	[A]
LDP_V_CUR_IP_V_CUR_VCV_MES	6	0... 7FFFH	0... 4.99984741211	152.588e-6	[V]
Transfer from measured voltage to current of the VCV					
IP_CUR_VCV_ST	6	0... FFFFH	0... 65.535	1e-3	[A]
LDPM_TCO_ST_1_FUSL	6	0... FEH	-48... 142.5	0.75	[°C]
CUR_VCV for the start after the event 1TOOTH					
IP_FAC_CUR_VCV_I_CTL	8	0... FFFFH	-2... 1.99993896484	61.0352e-6	[-]
LDP_CUR_DIF_MMV_IP_CUR_I_CTL	8	0... FFFFH	-2... 1.99993896484	61.0352e-6	[-]
I- share of VCV current controller					
IP_FAC_FRQ_V_CUR_VCV_MES	8	0... FFFFH	0... 3.99993896484	61.0352e-6	[-]
LDP_FRQ_REQ_V_CUR_VCV_MES	8	0... 3FFH	0... 1023	1	[Hz]
Frequency correction of the measured feedback					
IP_FAC_PWM_VCV_TFU	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_TFU_IP_FAC_PWM_VCV	8	0... FEH	-48... 142.5	0.75	[°C]
Correction factor for the PWM_VCV by the fuel temperature					
IP_FAC_PWM_VCV_VB	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDPM_VB_FUSL	8	0... FFH	0... 26	0.1019608	[V]
Correction factor for the PWM_VCV by the battery voltage					
IP_FAC_TECU_V_CUR_VCV_MES	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_TECU_IP_FAC_V_CUR_VCV_MES	8	0... FEH	-48... 142.5	0.75	[°C]
Temperature correction of the measured feedback					
IP_FAC_VB_PWM_V_CUR_VCV_MES	8*9	0... FFFFH	0... 7.99987792969	122.07e-6	[-]
LDP_VB_IP_FAC_V_CUR_VCV_MES	9	0... FFH	0... 26	0.1019608	[V]
LDP_PWM_VCV_IP_FAC_V_CUR_VCV	8	0... FFFFH	0... 99.9984741211	1.52588e-3	[%]
Correction factor for current feedback VB vs PWM					
IP_FAC_VFF_VCV_MIN	4	0... FFH	0... 9.9609375	0.0390625	[-]
LDP_TFU_IP_FAC_VFF_VCV_MIN	4	0... FEH	-48... 142.5	0.75	[°C]
Correction factor for the leakage adaptation					
IP_PWM_VCV_STND	6	0... FFFFH	0... 99.9984741211	1.52588e-3	[%]
LDP_CUR_VCV_IP_PWM_BAS	6	0... FFFFH	0... 65.535	1e-3	[A]
Transfer from desired current to standard PWM					
IP_VCV_STATE_CUR_VCV_MAX	6	0... FFFFH	0... 65.535	1e-3	[A]
LDPM_TCO_ST_1_FUSL	6	0... FEH	-48... 142.5	0.75	[°C]
CUR signal for VCV in STATE_PWM_VCV = Max					
LC_CUR_VCV_CTL_RST	1	0... 1H	0... 1	1	[-]
Switch to reset the VCV current controller					
LC_CUR_VCV_CTL_STOP	1	0... 1H	0... 1	1	[-]
Switch to stop the VCV current controller					
LC_CUR_VCV_MAN_ACT	1	0... 1H	0... 1	1	[-]
Switch for manual CUR_VCV					
LC_HPP_CTL_AD_CLR	1	0... 1H	0... 1	1	[-]
Clearing of adaptation values of high pressure pump control					
LC_PRS_H_ST	1	0... 1H	0... 1	1	[-]
Logical constant for selecting the high pressure start					
LC_V_CUR_VCV_BOOT_INI_MAN	1	0... 1H	0... 1	1	[-]
Switch to use INI values for current measurement					
LC_VCV_AST_MOVE_ACT_MAN	1	0... 1H	0... 1	1	[-]
Logical variable to activate the VCV after engine stop movement without LV_ERR_FUP_STOP					
LC_VCV_ST_CLOSE_TRAN	1	0... 1H	0... 1	1	[-]

General Information

The fuel pressure in the rail can be controlled by an volume control valve at the inlet side. Depending on the requested fuel mass to be delivered by the fuel pump the inlet valve to the pump has to be controlled.



In case of checksum error or new ECU initial values are taken for the corresponding variables. Otherwise the stored values are taken. All adapted values can be cleared by the variable LC_HPP_CTL_AD_CLR.

```
Use     ACTION_INFR_SetPwmVcv(IN<ducy>)
        ACTION_INFR_SetFrqPwmVcv(IN<frq>)
        IF      NC_V_CUR_VCV_MES > 0      (Shunt for VCV current is present in ECU)
        THEN    ACTION_INFR_GetVVcvCurMes(OUT <v_cur_vcv_mes>)
        ELSE
        END

At reset call
        ACTION_INFR_SetPwmVcv(IN<ducy>)
        ACTION_INFR_SetFrqPwmVcv(IN<frq>)

IF              LV_ST_END ==0
THEN             call at 10ms
                IF      NC_V_CUR_VCV_MES > 0      (Shunt for VCV current is present in ECU)
                THEN    ACTION_INFR_GetVVcvCurMes(OUT <v_cur_vcv_mes>)
                ELSE
                END
                IF      NOT (LV_PWL_LOCK_CDN_HPDI(n-1)) AND
```

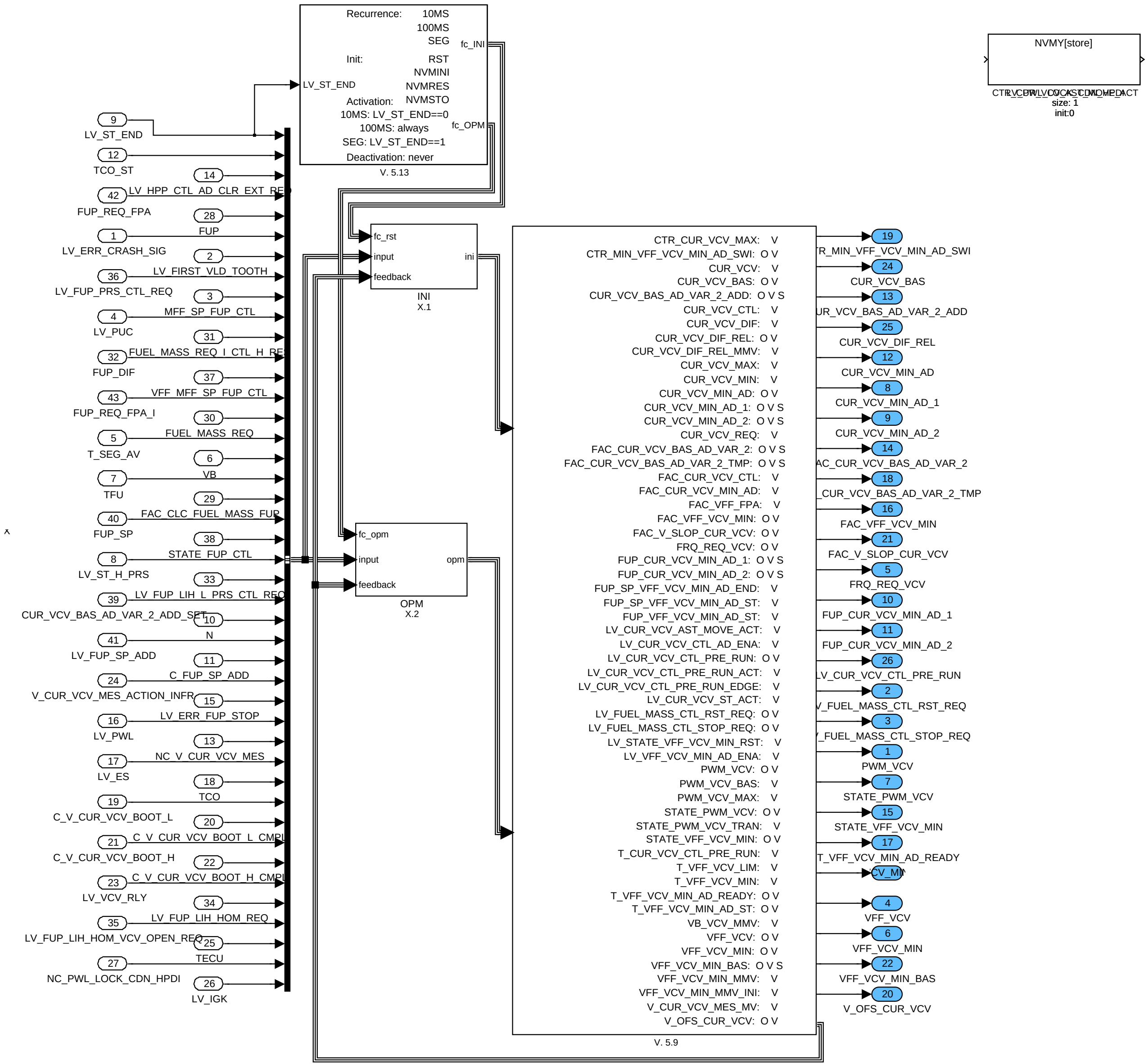
```
                LV_PWL_LOCK_CDN_HPDI(n)
        THEN ACTION_FCTM_SetPwlLockCdn(IN  <nc_pwl_lock_cdn_hpdi>)
        END
        IF      LV_PWL_LOCK_CDN_HPDI(n-1) AND
                NOT (LV_PWL_LOCK_CDN_HPDI(n))
        THEN ACTION_FCTM_ClrPwlLockCdn(IN  <nc_pwl_lock_cdn_hpdi>)
        END

        by  final calculation of pwm_vcv and frq_req_vcv
        ACTION_INFR_SetPwmVcv(IN<ducy>)
        ACTION_INFR_SetFrqPwmVcv(IN<frq>)
ELSE           call at segment
        IF      NC_V_CUR_VCV_MES > 0     (Shunt for VCV current is present in ECU)
        THEN   ACTION_INFR_GetVVcvCurMes(OUT <v_cur_vcv_mes>)
        ELSE
        END
        by  final calculation of pwm_vcv and frq_req_vcv
        ACTION_INFR_SetPwmVcv(IN<ducy>)
        ACTION_INFR_SetFrqPwmVcv(IN<frq>)
END
```

V_CUR_VCV_MES_ACTION_INFR isn't a real input and isn't generated by an other specification. It is only the voltage from the ACTION_INFR_GetVVcvCurMes(OUT <v_cur_vcv_mes>)

Application Conditions

Initialization: RST, NVMINI, NVMRES, NVMSTO
Recurrence: 10MS activated if LV_ST_END==0
100MS activated if always
SEG activated if LV_ST_END==1
Deactivation: never

Function description



SDA_SRS / SDA V 5.0.6 13−Jul−2007

Figure 187:

34.15.1 Initialization:

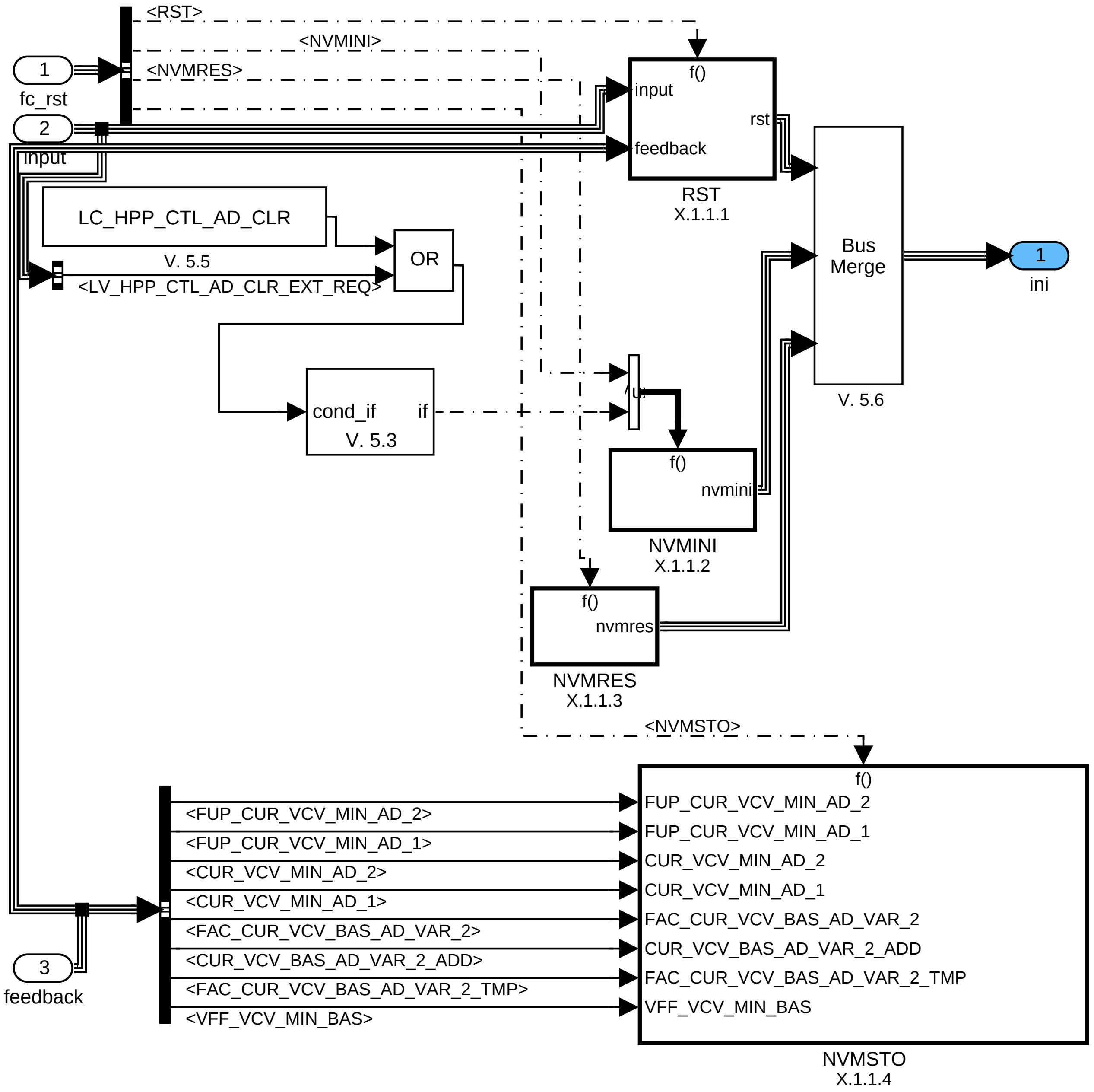


Figure 188:

34.15.1.1 NVMY Initialization:

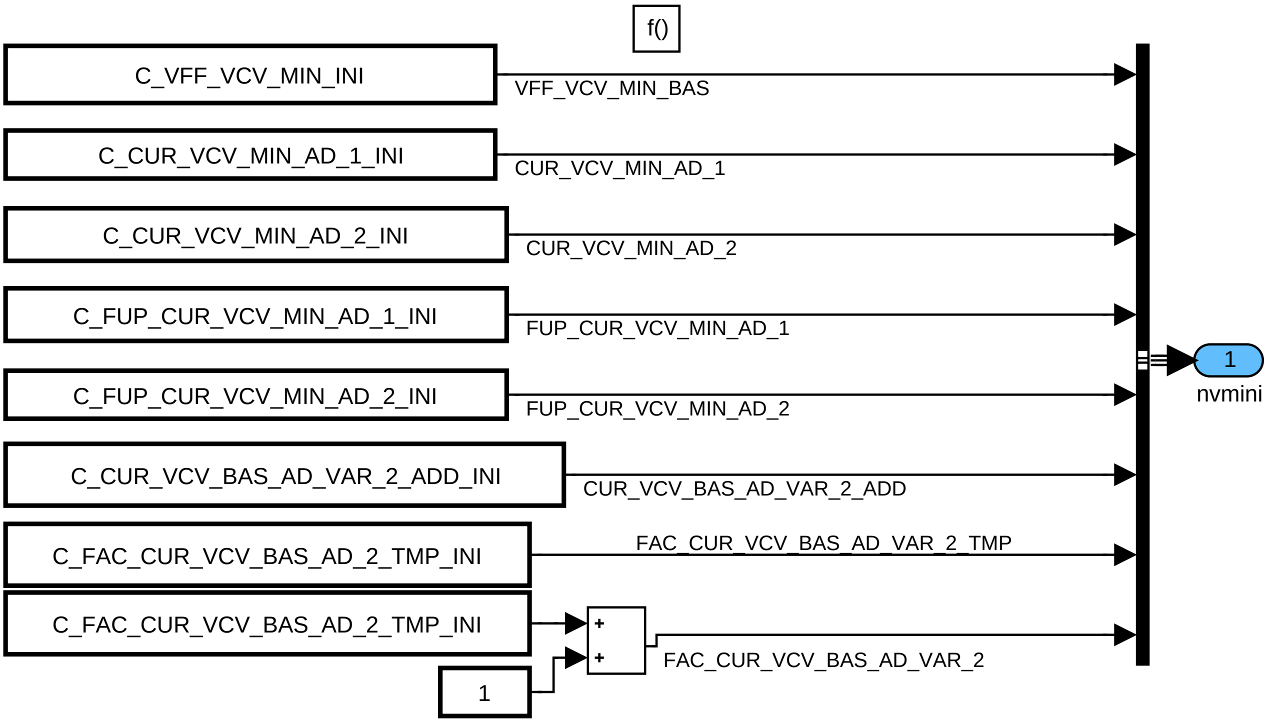


Figure 189:

34.15.1.2 NVMY Restore

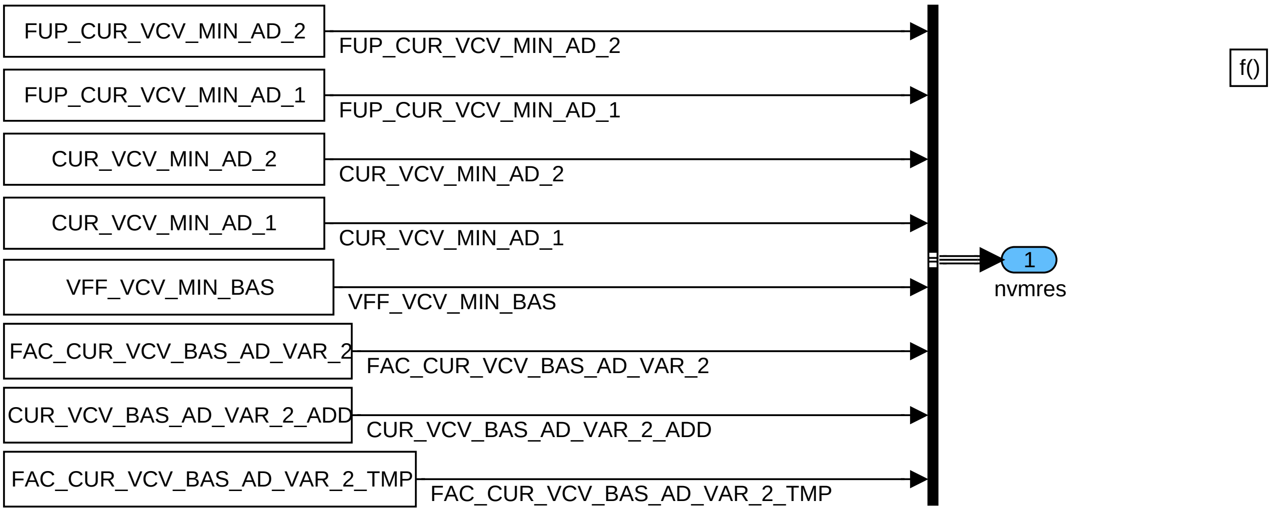


Figure 190:

34.15.1.3 NVMY Store

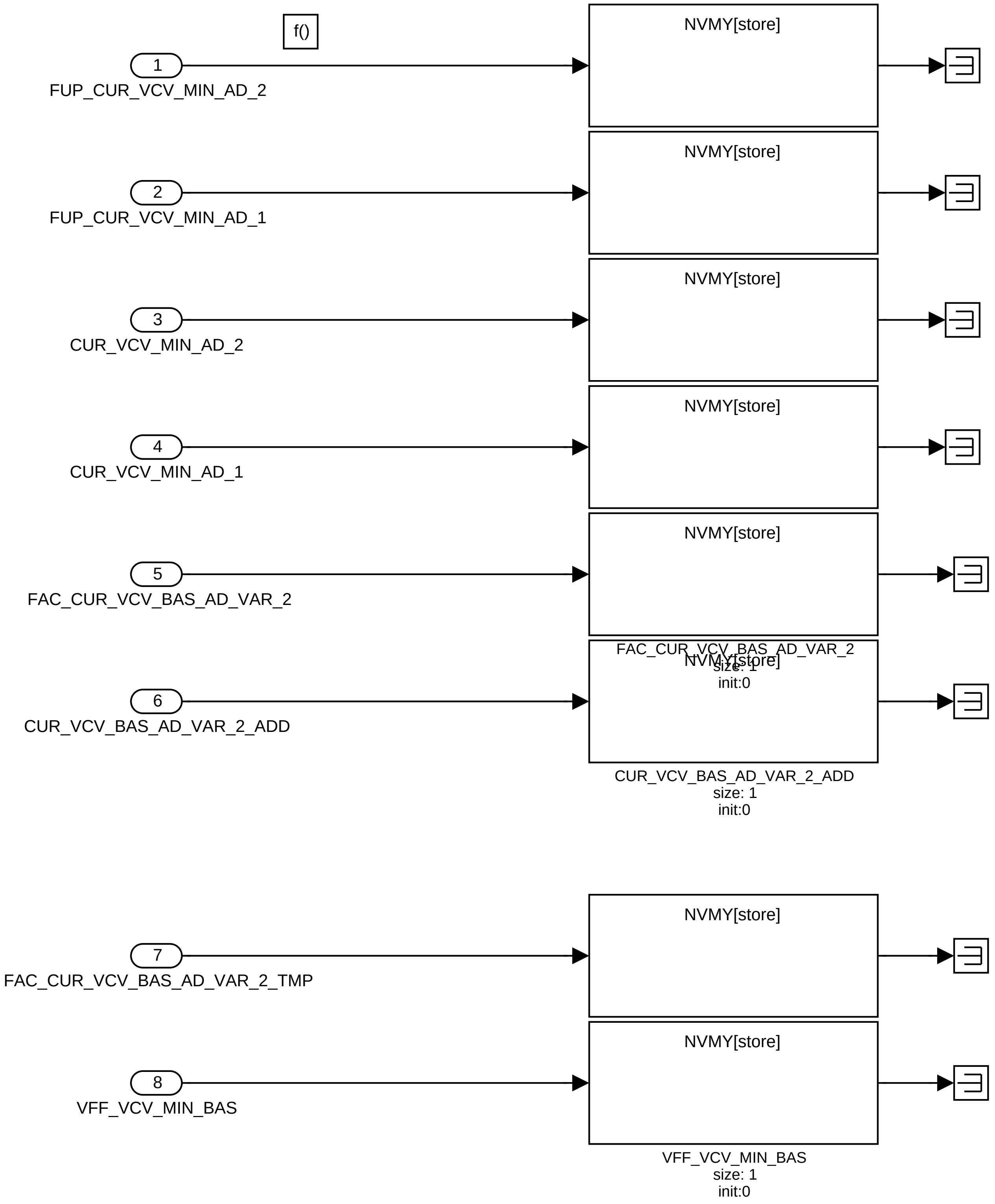


Figure 191:

34.15.1.4 Reset

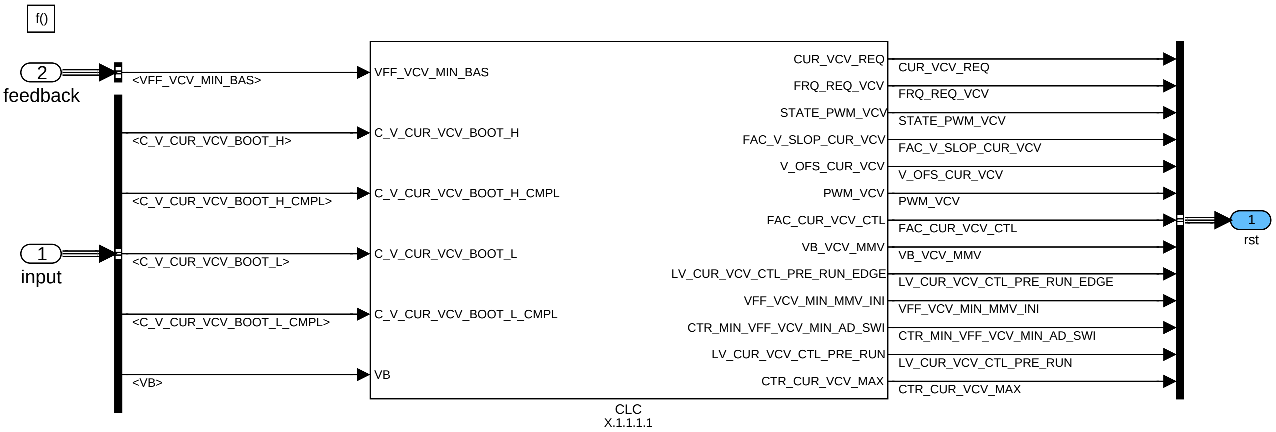


Figure 192:

34.15.1.4.1 Calculation:

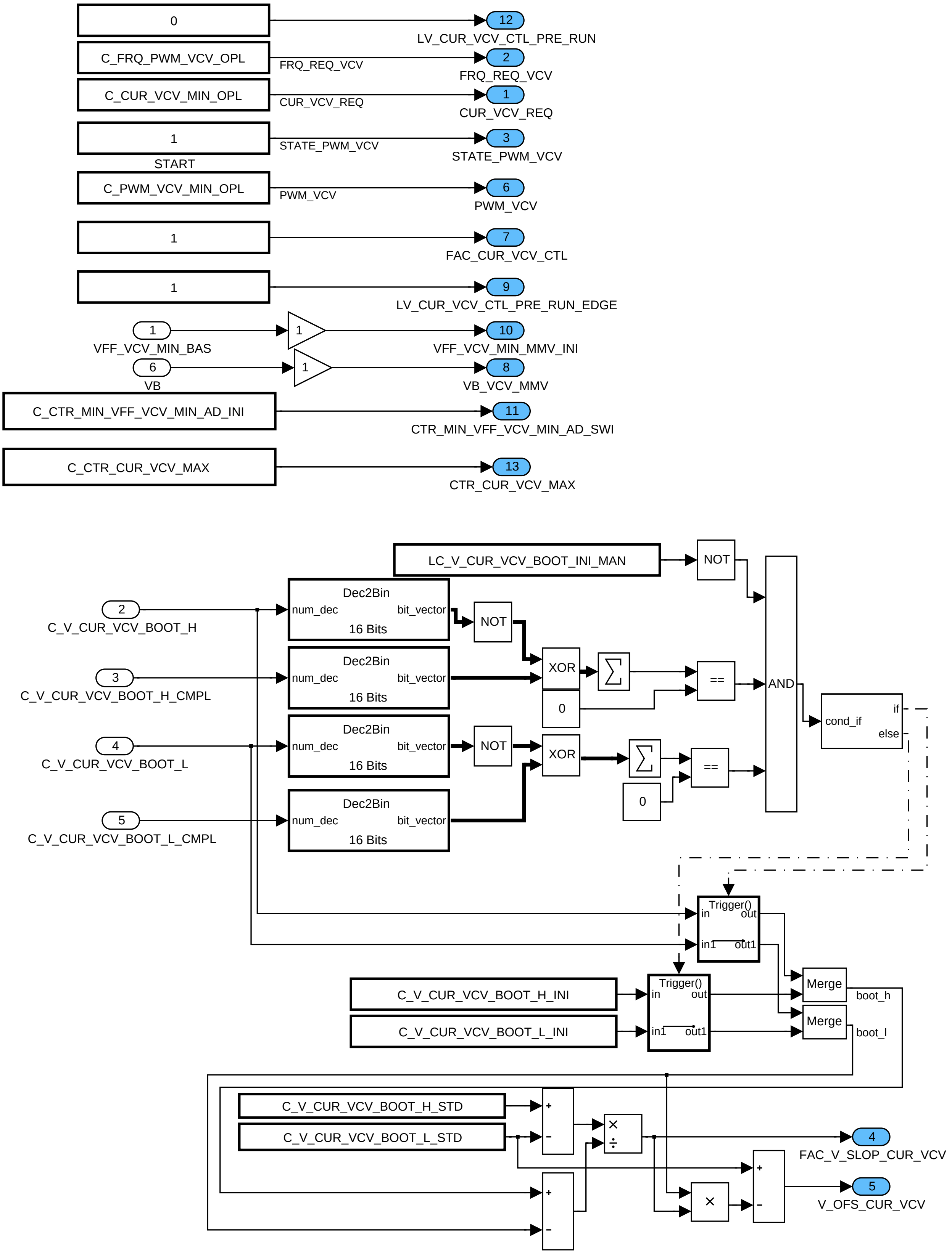


Figure 193:

34.15.2 Formula Section

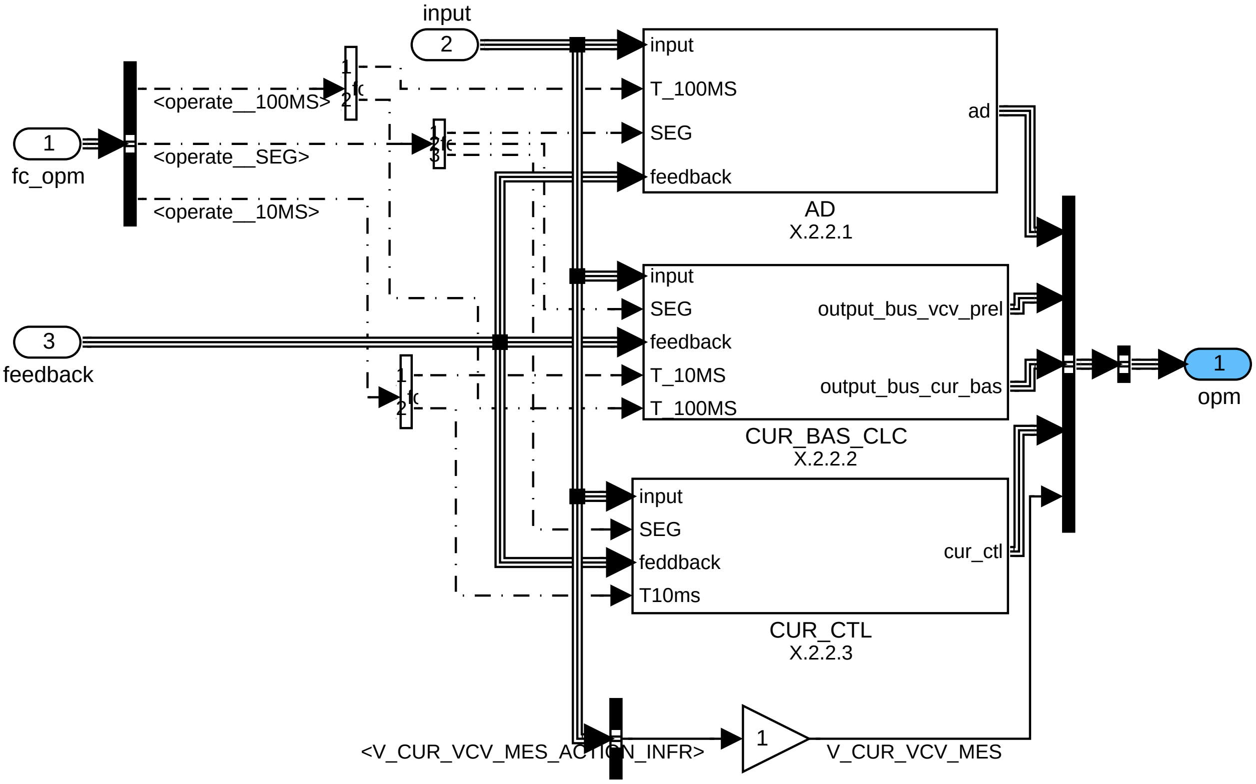


Figure 194:

34.15.2.1 Adaptation

This function part provides the adaptation values of the VCV.

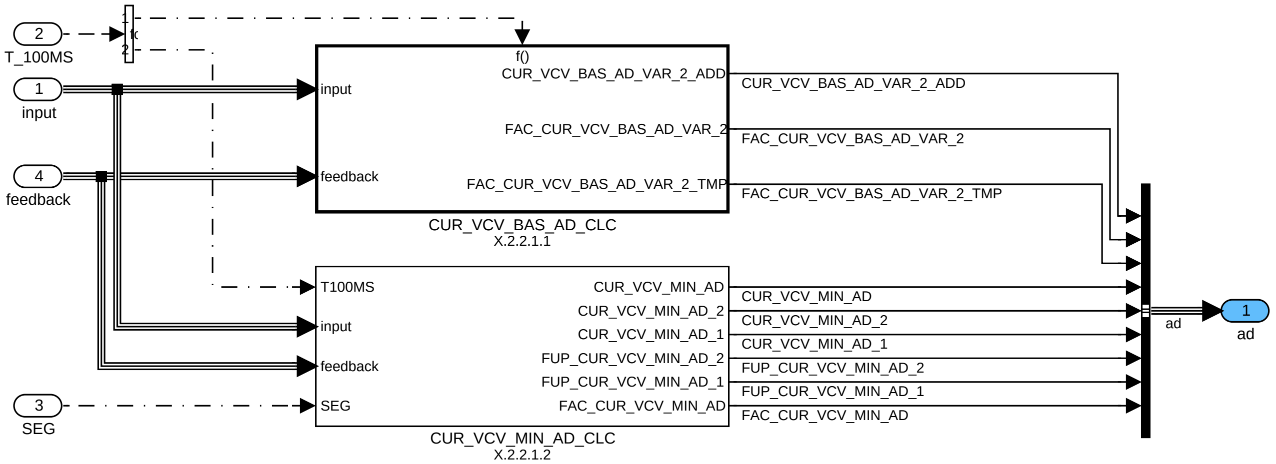


Figure 195:

34.15.2.1.1 Adaptation of valve characteristics:

In this block the factor of the second variant of the VCV_BAS adaptation is calculated.

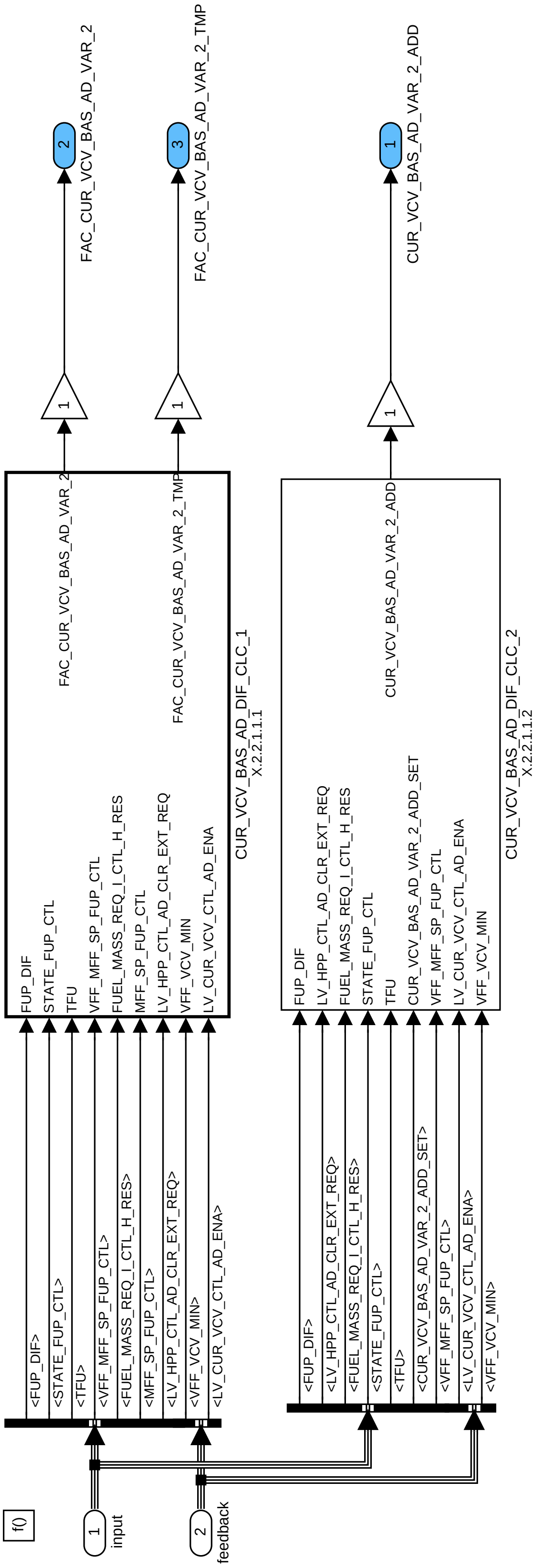

FAC_CUR_VCV_BAS_AD_VAR_2
FAC_CUR_VCV_BAS_AD_VAR_2_TMP
CUR_VCV_BAS_AD_VAR_2_ADD
2
3
1
1
1
1
FAC_CUR_VCV_BAS_AD_VAR_2
FAC_CUR_VCV_BAS_AD_VAR_2_TMP
CUR_VCV_BAS_AD_VAR_2_ADD
CUR_VCV_BAS_AD_DIF_CLC_1
X.2.2.1.1.1
CUR_VCV_BAS_AD_DIF_CLC_2
X.2.2.1.1.2
FUP_DIF
STATE_FUP_CTL
TFU
VFF_MFF_SP_FUP_CTL
FUEL_MASS_REQ_I_CTL_H_RES
MFF_SP_FUP_CTL
LV_HPP_CTL_AD_CLR_EXT_REQ
VFF_VCV_MIN
LV_CUR_VCV_CTL_AD_ENA
FUP_DIF
LV_HPP_CTL_AD_CLR_EXT_REQ
FUEL_MASS_REQ_I_CTL_H_RES
STATE_FUP_CTL
TFU
CUR_VCV_BAS_AD_VAR_2_ADD_SET
VFF_MFF_SP_FUP_CTL
LV_CUR_VCV_CTL_AD_ENA
VFF_VCV_MIN
<FUP_DIF>
<STATE_FUP_CTL>
<TFU>
<VFF_MFF_SP_FUP_CTL>
<FUEL_MASS_REQ_I_CTL_H_RES>
<MFF_SP_FUP_CTL>
<LV_HPP_CTL_AD_CLR_EXT_REQ>
<VFF_VCV_MIN>
<LV_CUR_VCV_CTL_AD_ENA>
<FUP_DIF>
<LV_HPP_CTL_AD_CLR_EXT_REQ>
<FUEL_MASS_REQ_I_CTL_H_RES>
<STATE_FUP_CTL>
<TFU>
<CUR_VCV_BAS_AD_VAR_2_ADD_SET>
<VFF_MFF_SP_FUP_CTL>
<LV_CUR_VCV_CTL_AD_ENA>
<VFF_VCV_MIN>
f()
1
input
2
feedback

Figure 196:

34.15.2.1.1.1 Additive component calculation

In this block the additive component of the second variant of the VCV_BAS adaptation is calculated.

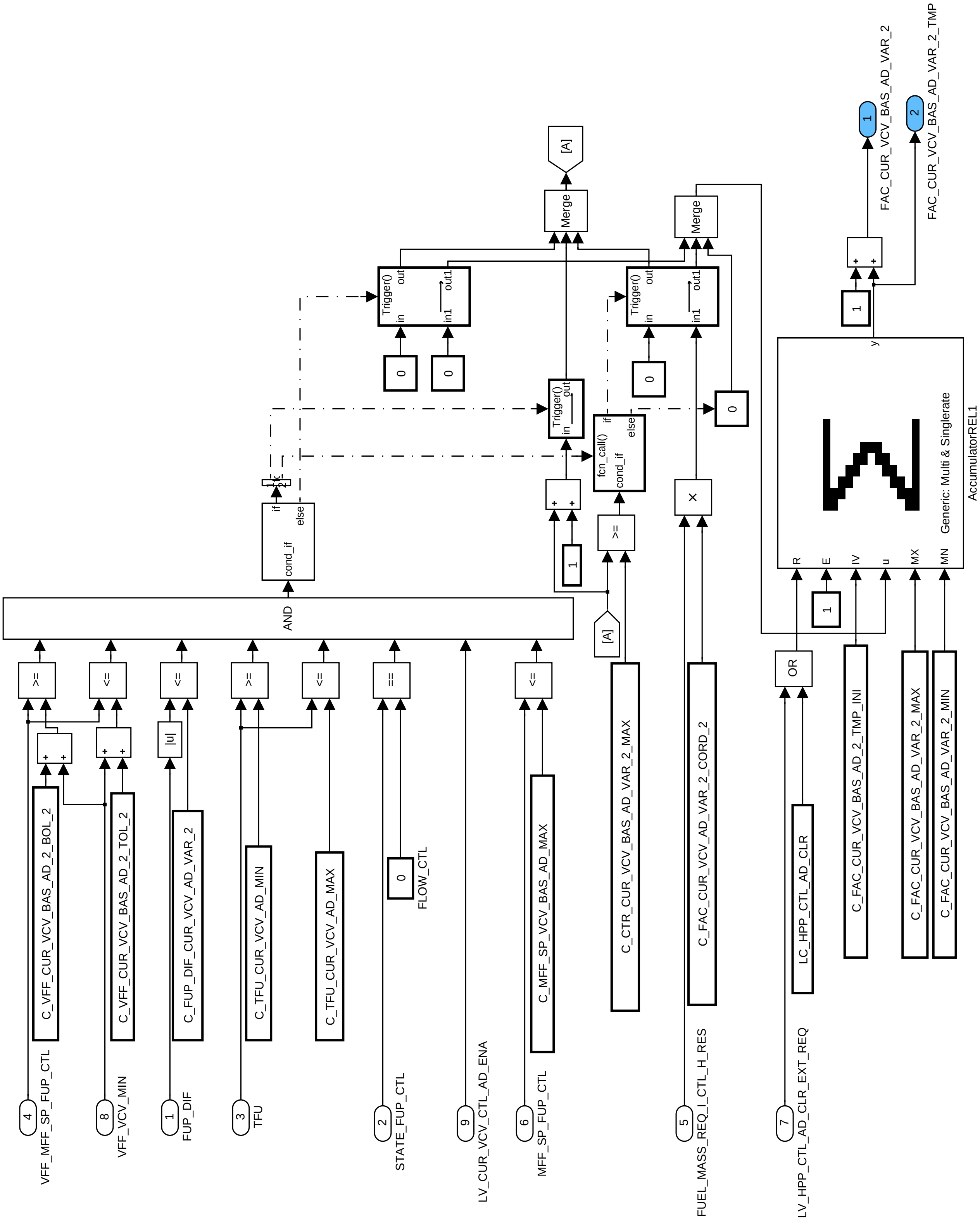


Figure 197:

34.15.2.1.1.2 Factor Calculation

In this block the factor of the second variant of the VCV_BAS adaptation is calculated.

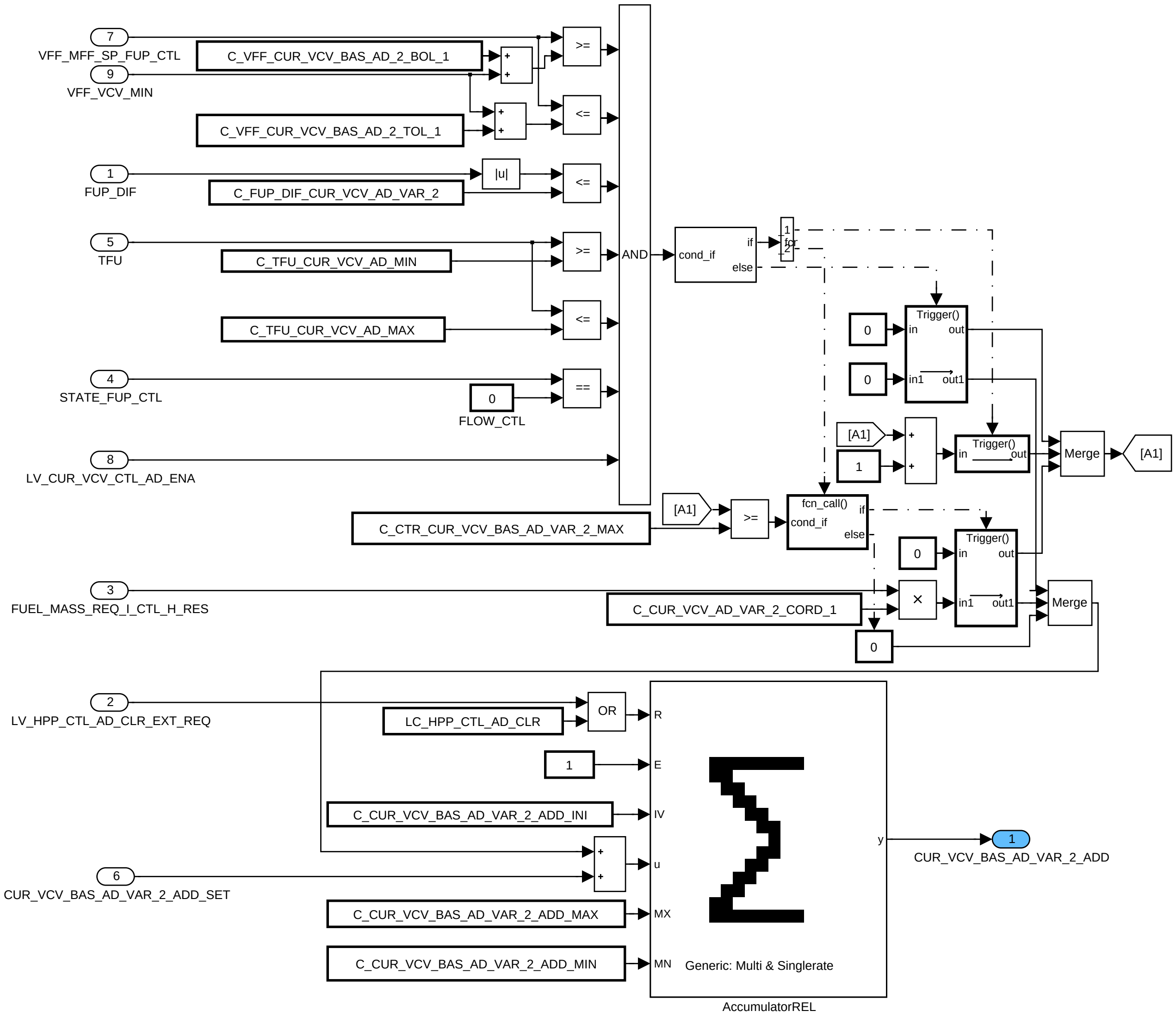


Figure 198:

34.15.2.1.2 Pressure control line adaptation of the VCV valve

This calculation is split into the calculations of the two points, the calculation of a factor and the final calculation of the adaptation value for the pressure control part of the valve.
! CUR_VCV_MIN_AD_CLC has to be calculated at first inside segment synchronous task.

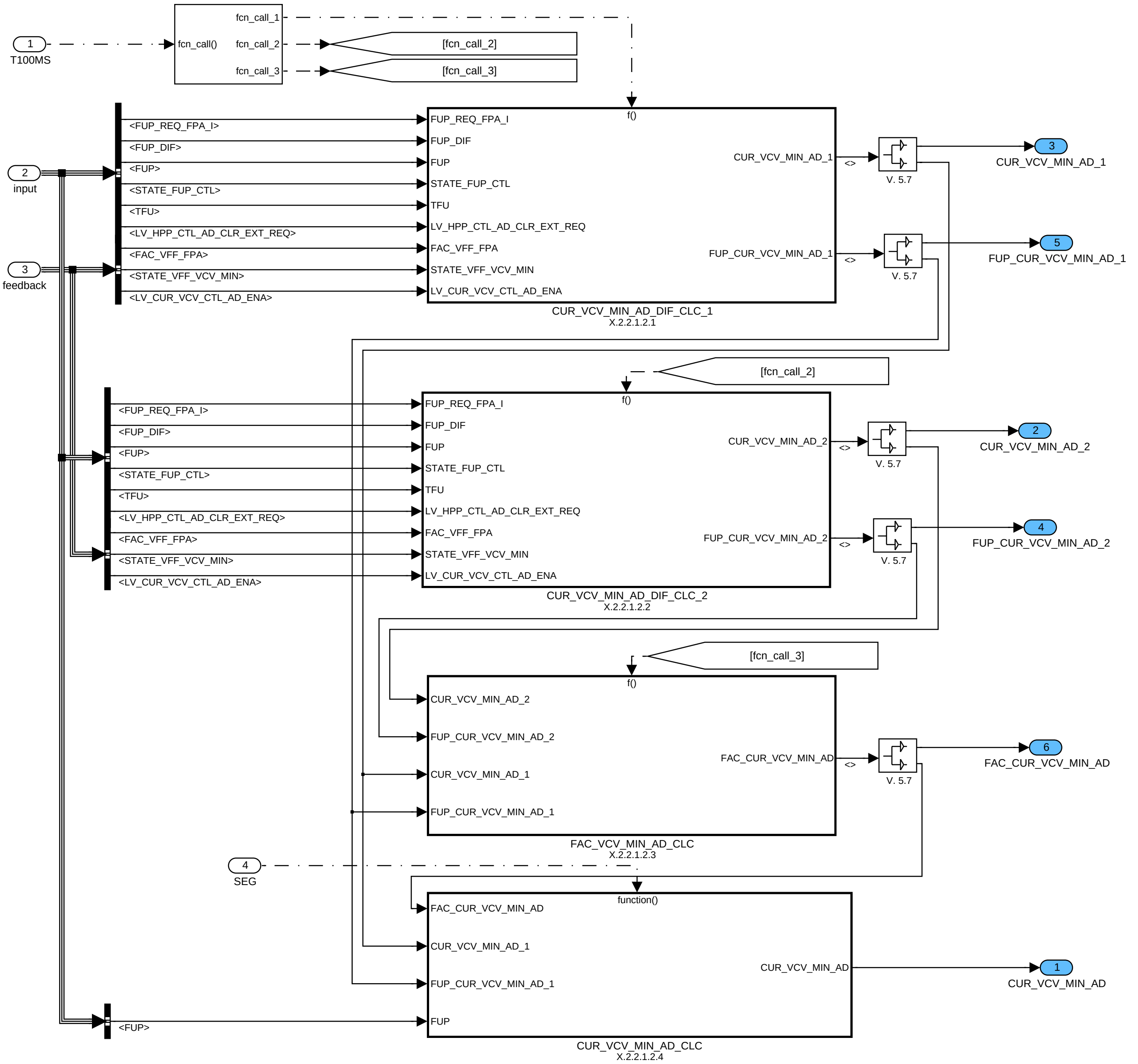


Figure 199:

34.15.2.1.2.1 Calculation of the final pressure control line adaptation value

The final adaptation value for the pressure control part of the valve is calculated in that block.

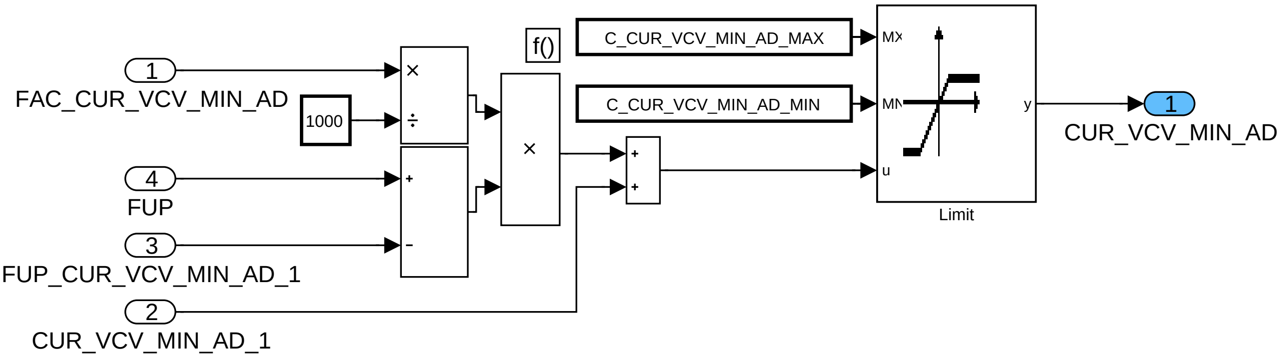


Figure 200:

34.15.2.1.2.2 Adaptation of point 2 of pressure control line

The calculation of the first point of the adaptation of the pressure control line is constisting out of a MMV calculation of the flow and the corresponding state chart

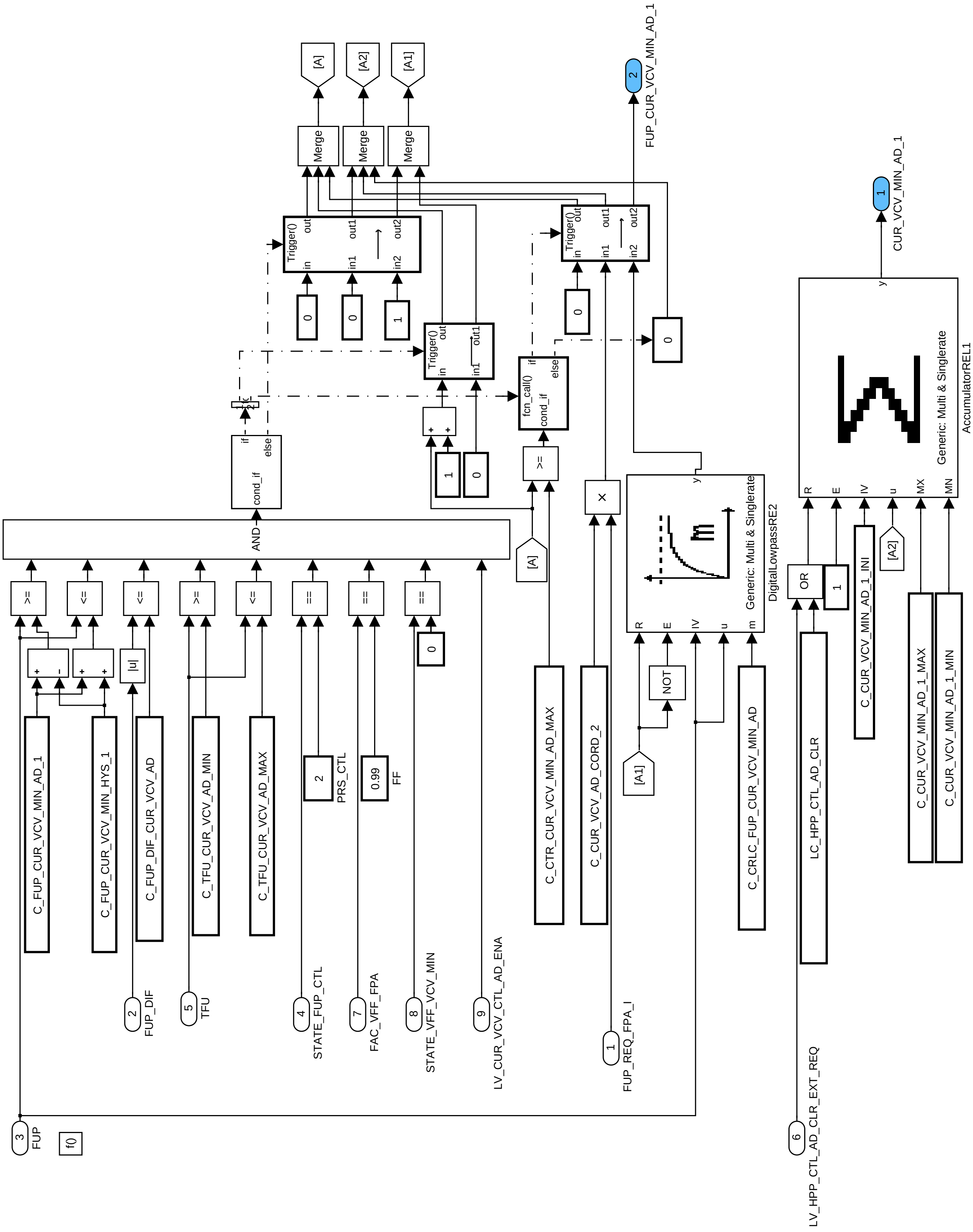


Figure 201:

34.15.2.1.2.3 Calculation of point 1 of pressure control line

The calculation of the second point of the adaptation of the pressure control line is consisting out of a MMV calculation of the flow and the corresponding state chart.

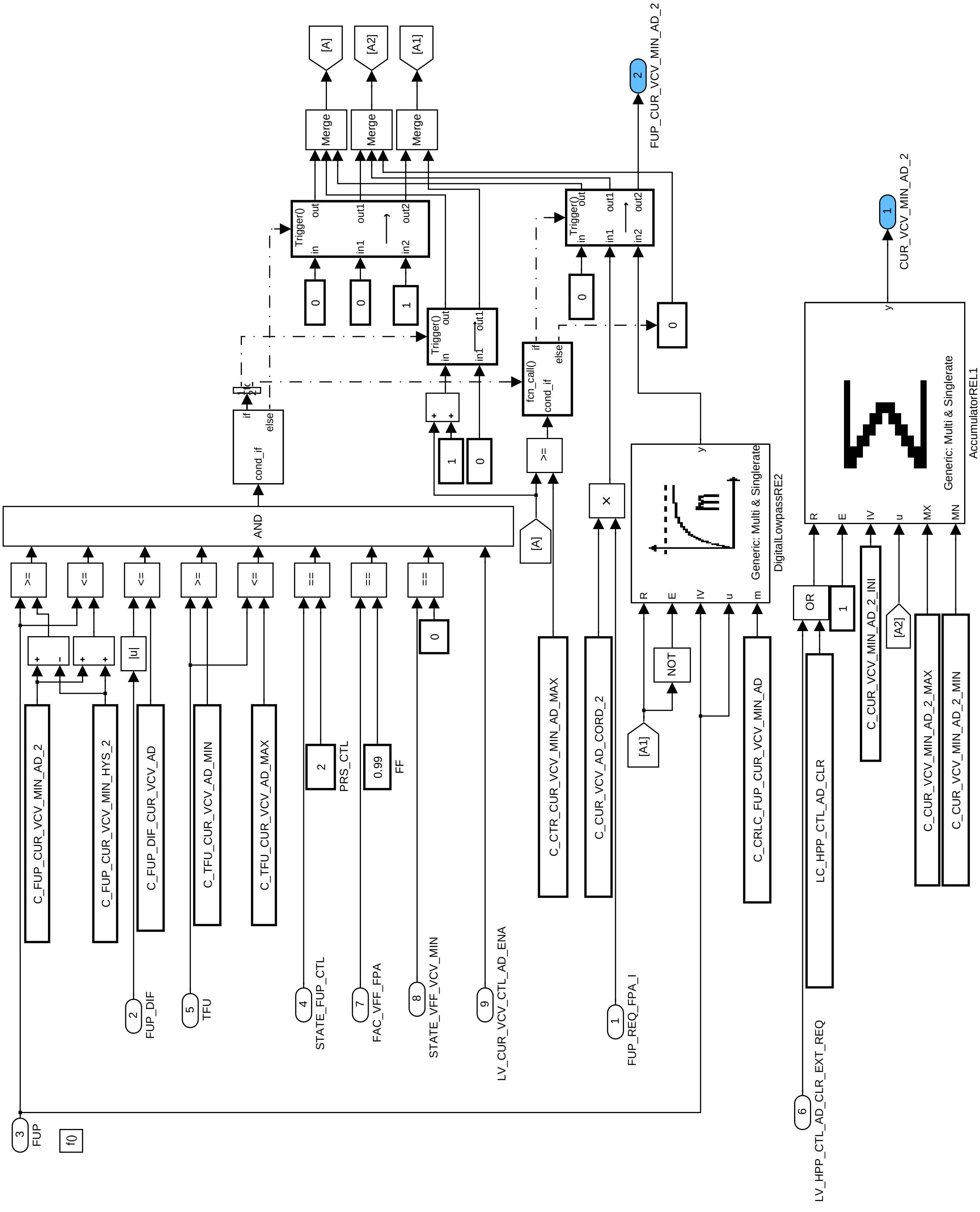


Figure 202:

34.15.2.1.2.4 Calculation of the slope of the pressure control line

In this block the slope for the correction of the pressure control part of the valve is calculated.

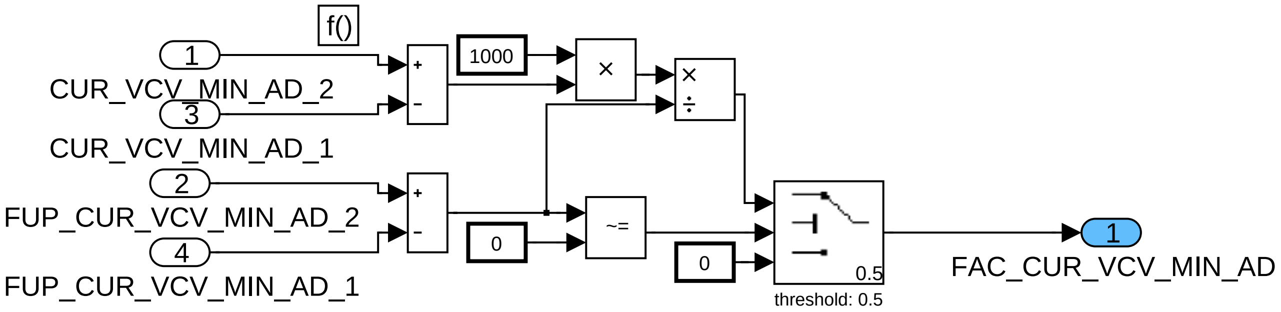


Figure 203:

34.15.2.2 Basic VCV Current Calculation

Overview of basic VCV current calculation. This function part provides the desired VCV current.

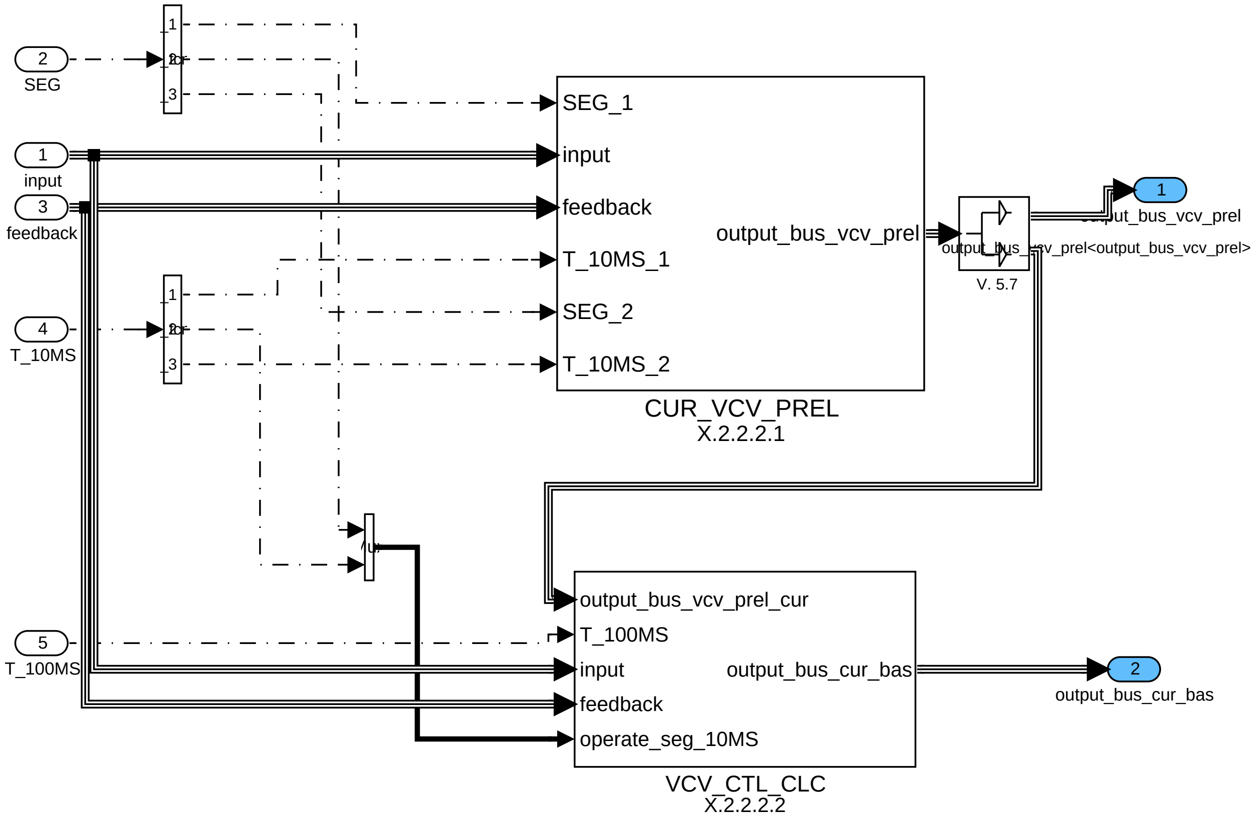


Figure 204:

34.15.2.2.1 Calculation of the basic fuel flows through the valve

The requested volume fuel flow through the volume control valve is calculated by the requested fuel mass and a conversion calculation taken into account the segment time and the density of the fuel. The real pumped fuel flow volume is calculated out of the calculated pump fuel mass.

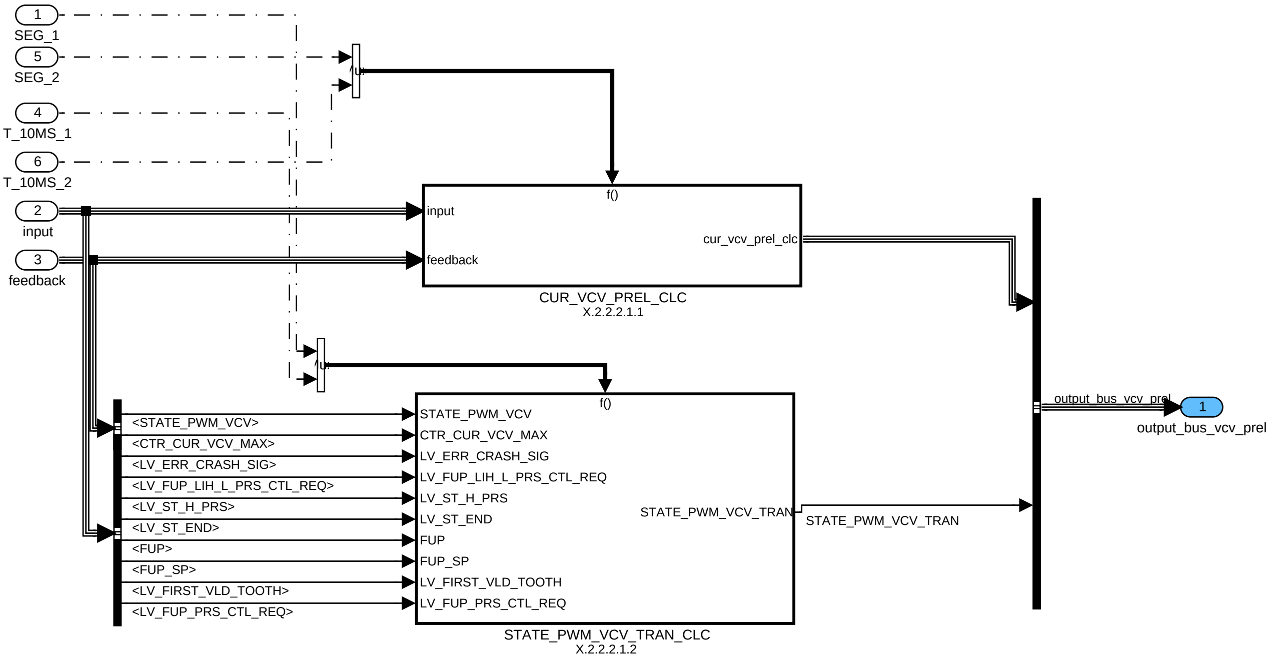


Figure 205:

34.15.2.2.1.1 CUR_VCV_PREL_CLC

This part has been divided into 3 parts

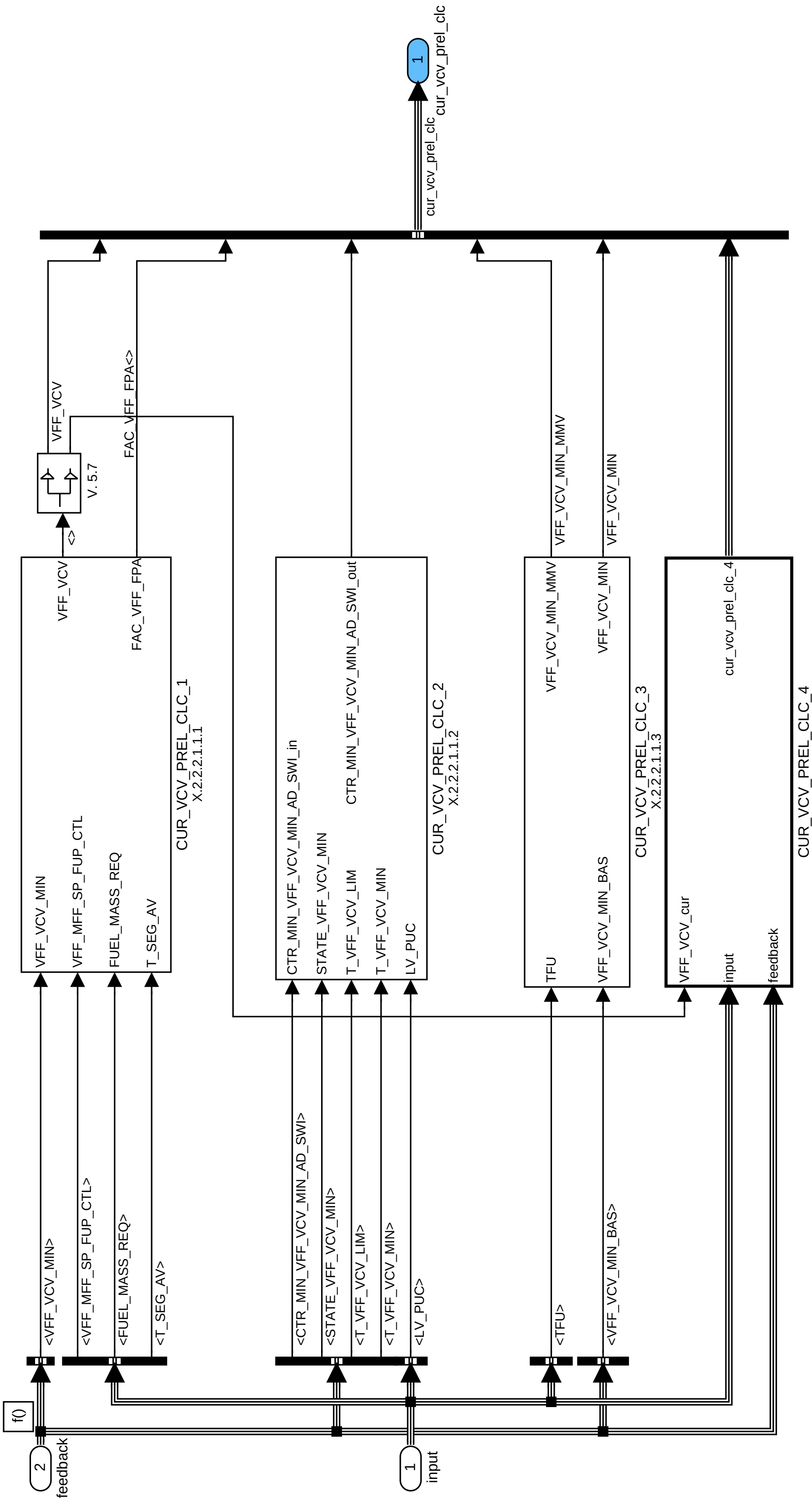

1
cur_vcv_prel_clc
cur_vcv_prel_clc
VFF_VCV
FAC_VFF_FPA<>
V. 5.7
<>
VFF_VCV
FAC_VFF_FPA
VFF_VCV_MIN
VFF_MFF_SP_FUP_CTL
FUEL_MASS_REQ
T_SEG_AV
CUR_VCV_PREL_CLC_1
X.2.2.2.1.1.1
CTR_MIN_VFF_VCV_MIN_AD_SWI_out
CTR_MIN_VFF_VCV_MIN_AD_SWI_in
STATE_VFF_VCV_MIN
T_VFF_VCV_LIM
T_VFF_VCV_MIN
LV_PUC
CUR_VCV_PREL_CLC_2
X.2.2.2.1.1.2
VFF_VCV_MIN_MMV
VFF_VCV_MIN
VFF_VCV_MIN_MMV
VFF_VCV_MIN
TFU
VFF_VCV_MIN_BAS
CUR_VCV_PREL_CLC_3
X.2.2.2.1.1.3
cur_vcv_prel_clc_4
VFF_VCV_cur
input
feedback
CUR_VCV_PREL_CLC_4
X.2.2.2.1.1.4
<VFF_VCV_MIN>
<VFF_MFF_SP_FUP_CTL>
<FUEL_MASS_REQ>
<T_SEG_AV>
<CTR_MIN_VFF_VCV_MIN_AD_SWI>
<STATE_VFF_VCV_MIN>
<T_VFF_VCV_LIM>
<T_VFF_VCV_MIN>
<LV_PUC>
<TFU>
<VFF_VCV_MIN_BAS>
f()
2
feedback
1
input

Figure 206:

34.15.2.2.1.1.1 CUR_VCV_PREL_CLC_1

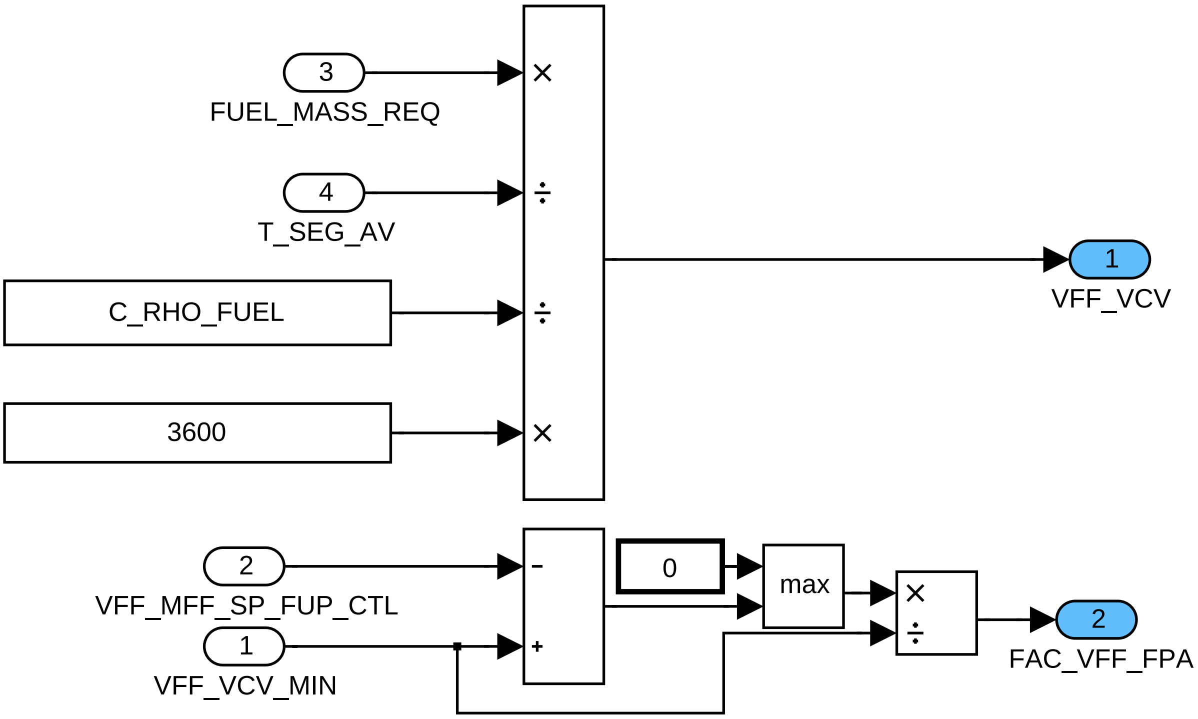


Figure 207:

34.15.2.2.1.1.2 CUR_VCV_PREL_CLC_2

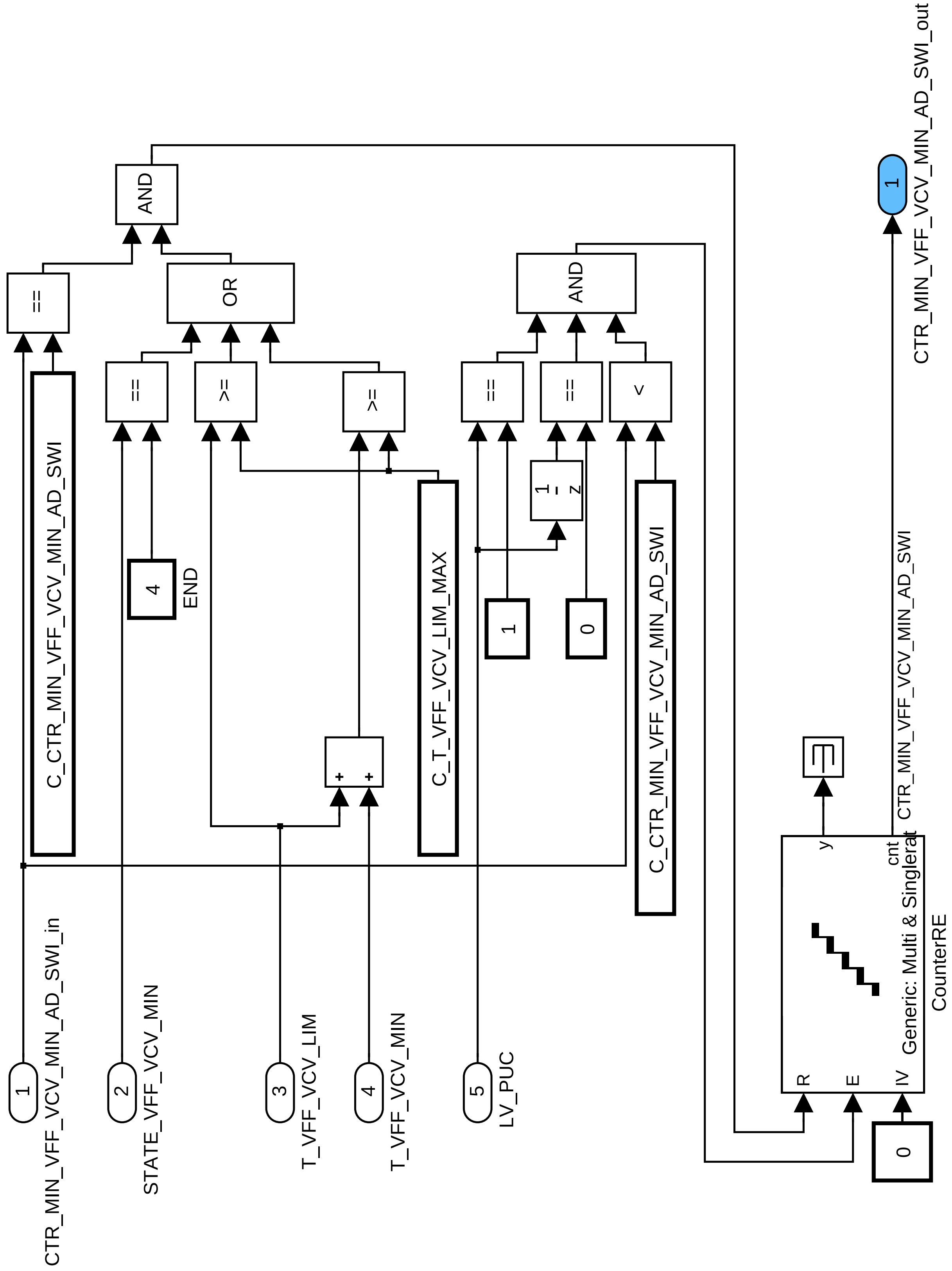


Figure 208:

34.15.2.2.1.1.3 CUR_VCV_PREL_CLC_3

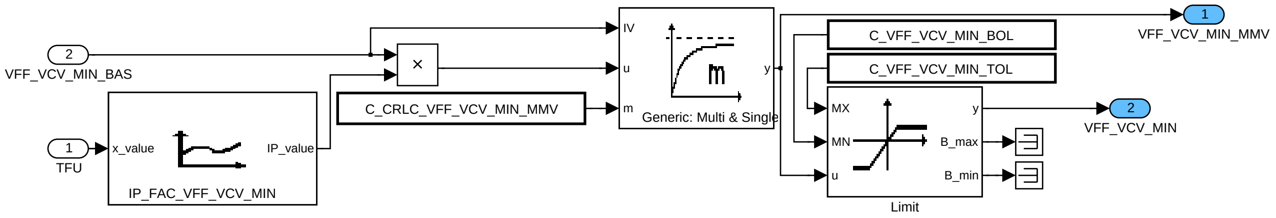


Figure 209:

34.15.2.2.1.1.4 CUR_VCV_PREL_CLC_4

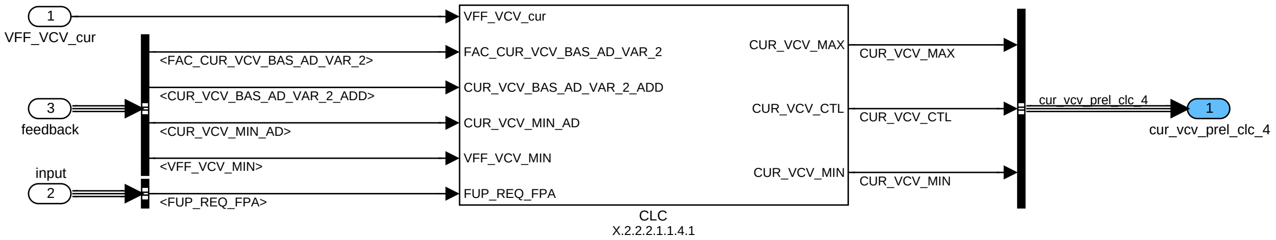


Figure 210:

34.15.2.2.1.1.4.1 Calculation

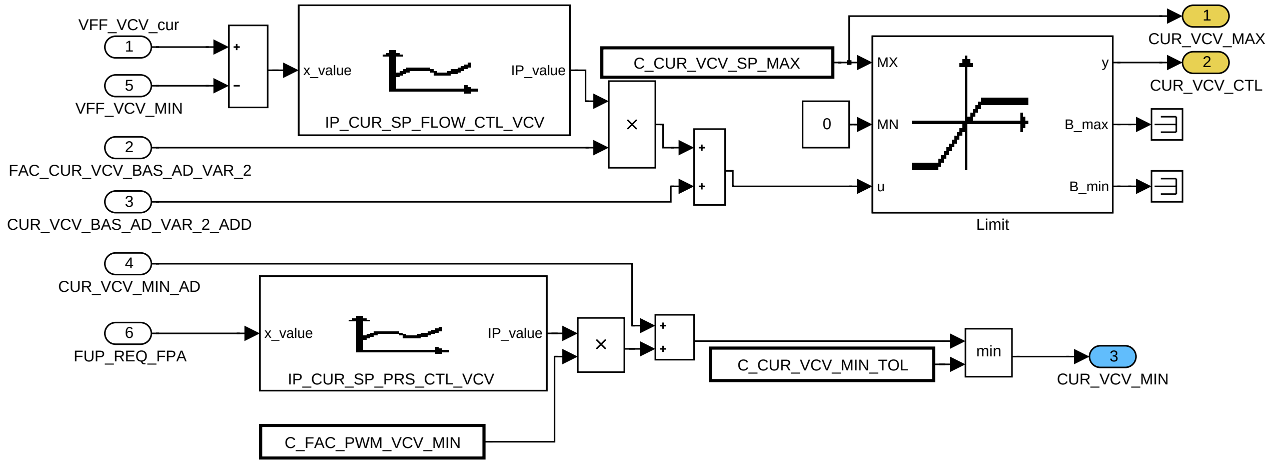


Figure 211:

34.15.2.2.1.2 Calculation of the transition conditions for the different working areas of the volume control valve

In this block the conditions for operating the state diagram are given. At crash event a safe state should be reached. In case of pull the valve should be closed and in case of engine stop the valve should be open. Manual operation can be selected by a flag. If none of the above mentioned cases are valid the normal operation mode can be done.

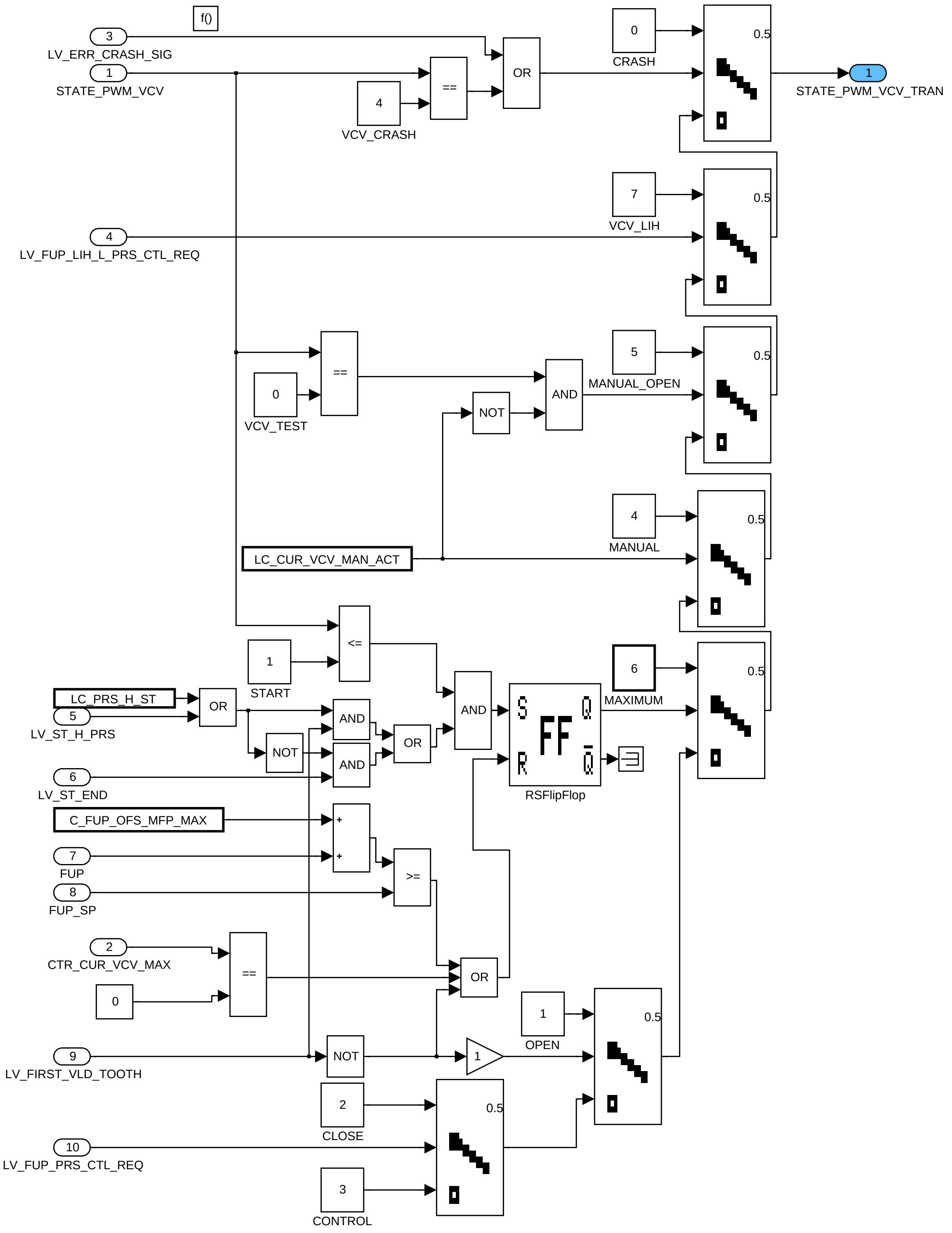

f()
3
LV_ERR_CRASH_SIG
1
STATE_PWM_VCV
4
VCV_CRASH
==
OR
0
CRASH
0.5
1
STATE_PWM_VCV_TRAN
4
LV_FUP_LIH_L_PRS_CTL_REQ
7
VCV_LIH
0.5
0
VCV_TEST
==
NOT
AND
5
MANUAL_OPEN
0.5
LC_CUR_VCV_MAN_ACT
4
MANUAL
0.5
<=
1
START
LC_PRS_H_ST
5
LV_ST_H_PRS
OR
AND
NOT
AND
OR
AND
S
R
Q
FF
RSFlipFlop
6
MAXIMUM
0.5
6
LV_ST_END
C_FUP_OFS_MFP_MAX
+
+
7
FUP
>=
8
FUP_SP
2
CTR_CUR_VCV_MAX
==
0
OR
1
OPEN
0.5
9
LV_FIRST_VLD_TOOTH
NOT
1
2
CLOSE
0.5
10
LV_FUP_PRS_CTL_REQ
3
CONTROL

Figure 212:

34.15.2.2.2 Final calculation of the output signals

In the state diagram the states are set corresponding to the transition conditions defined before. Out of the conditions the PWM can be calculated with the two blocks following the state diagram.

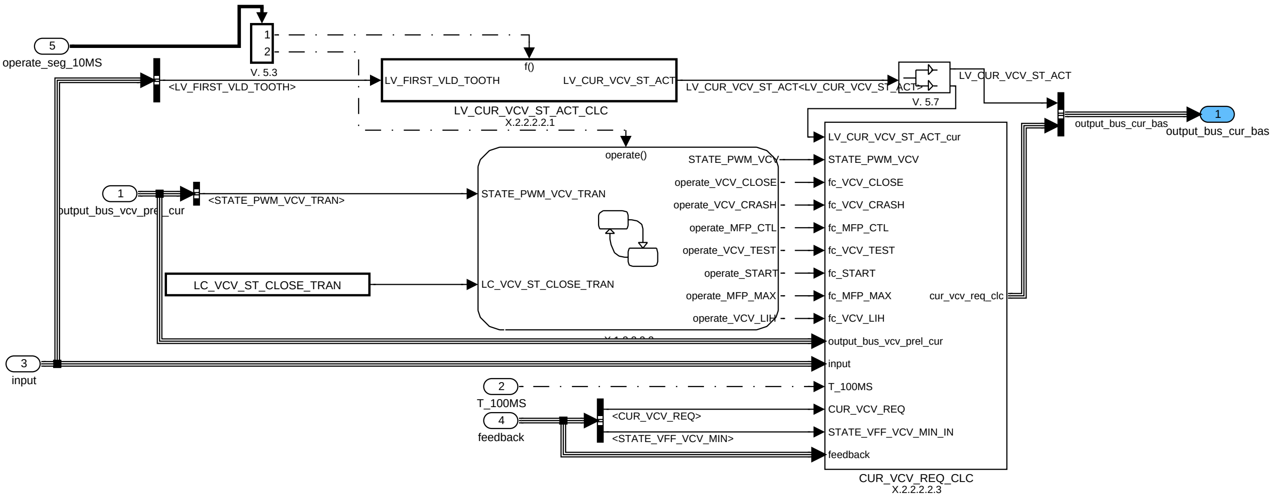


Figure 213:

34.15.2.2.2.1 Calculation of CUR_VCV_REQ_CLC

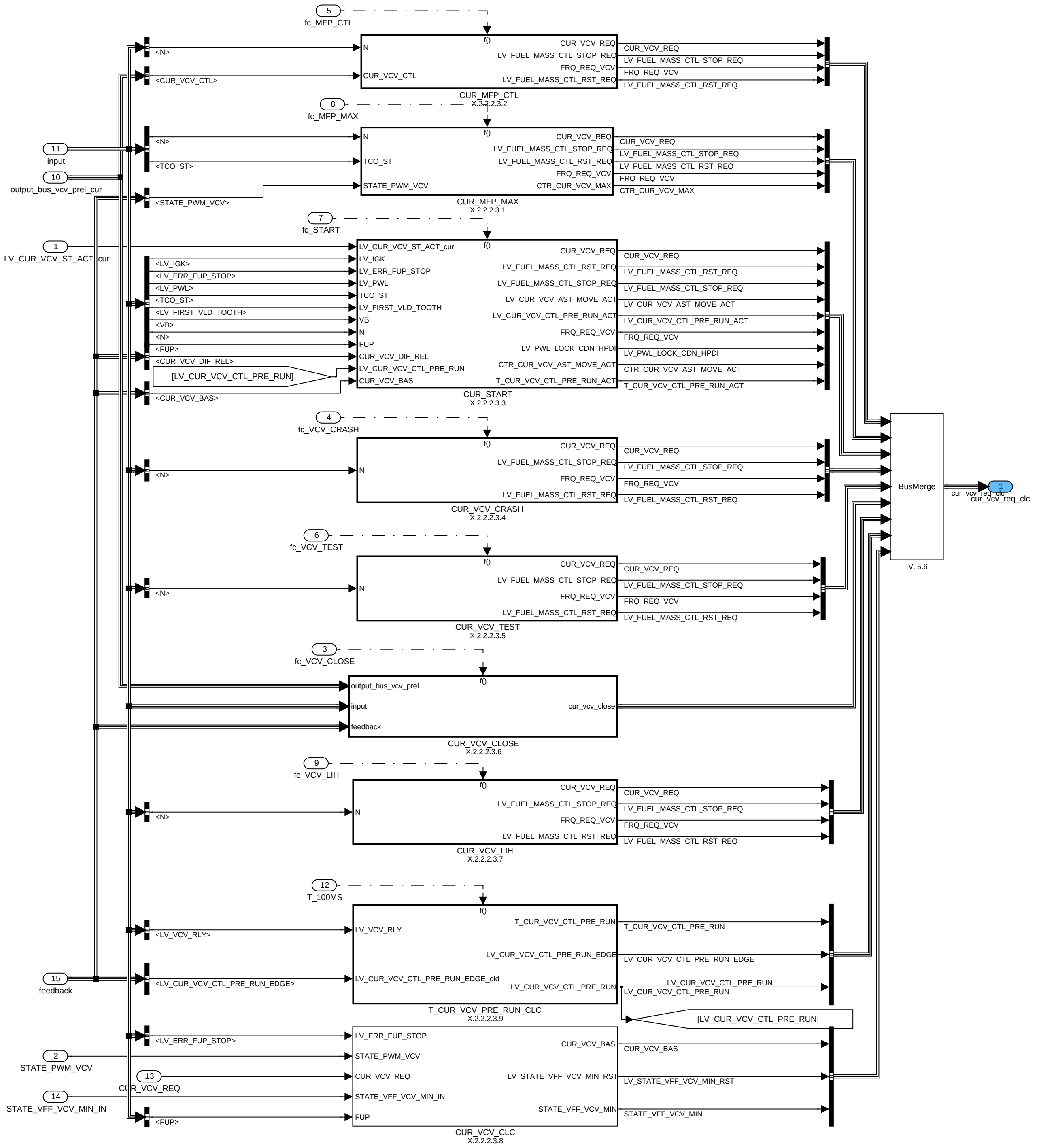


Figure 214:

34.15.2.2.2.1.1 CUR_MFP_CTL

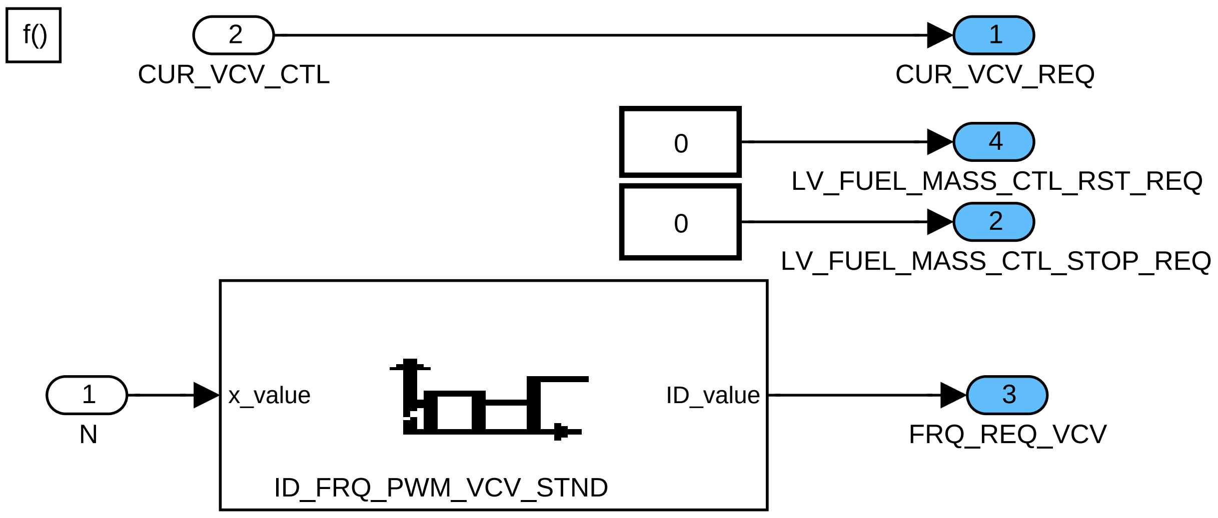


Figure 215:

34.15.2.2.2.1.2 CUR_MFP_MAX

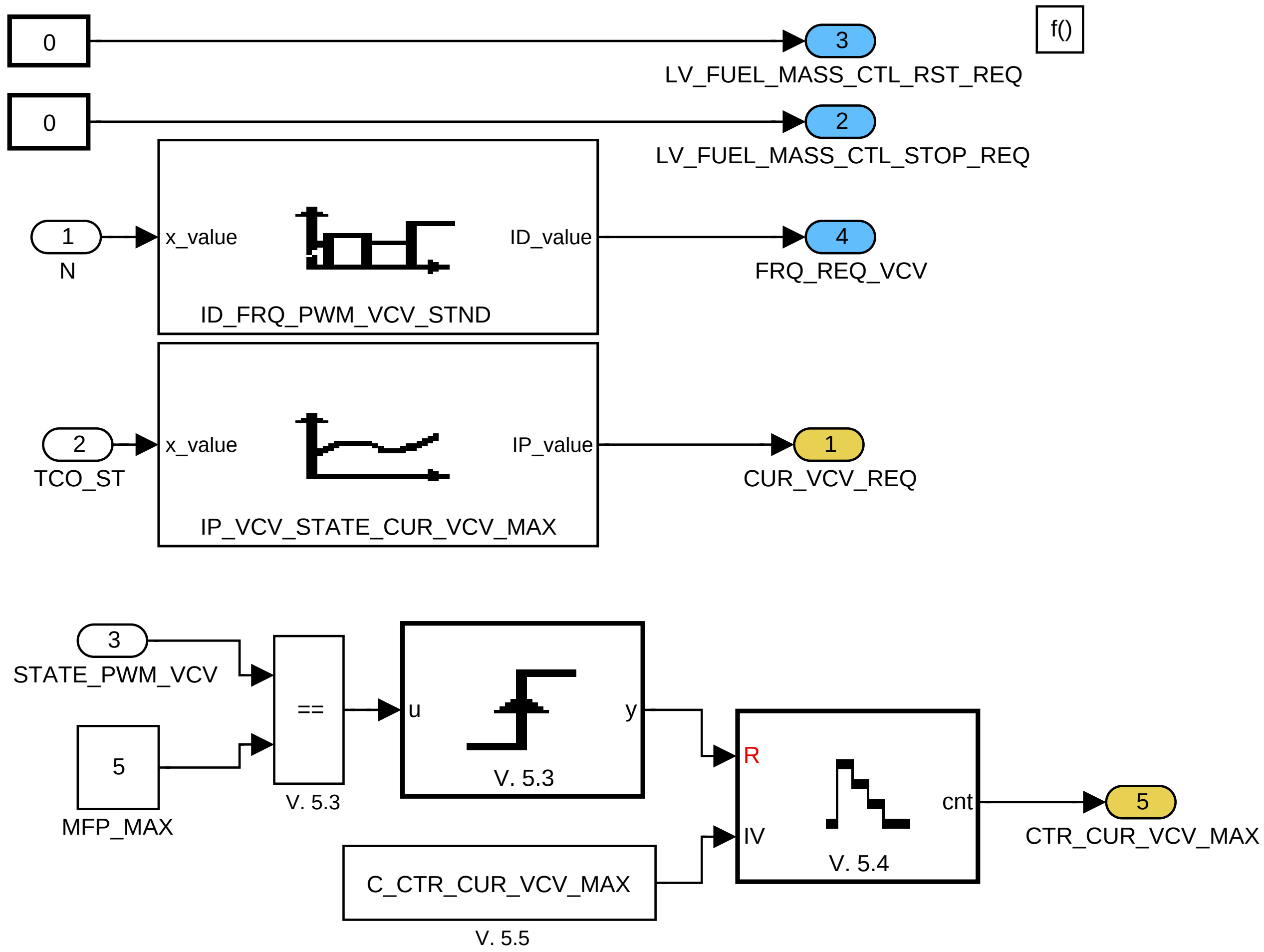


Figure 216:

34.15.2.2.2.1.3 CUR_START

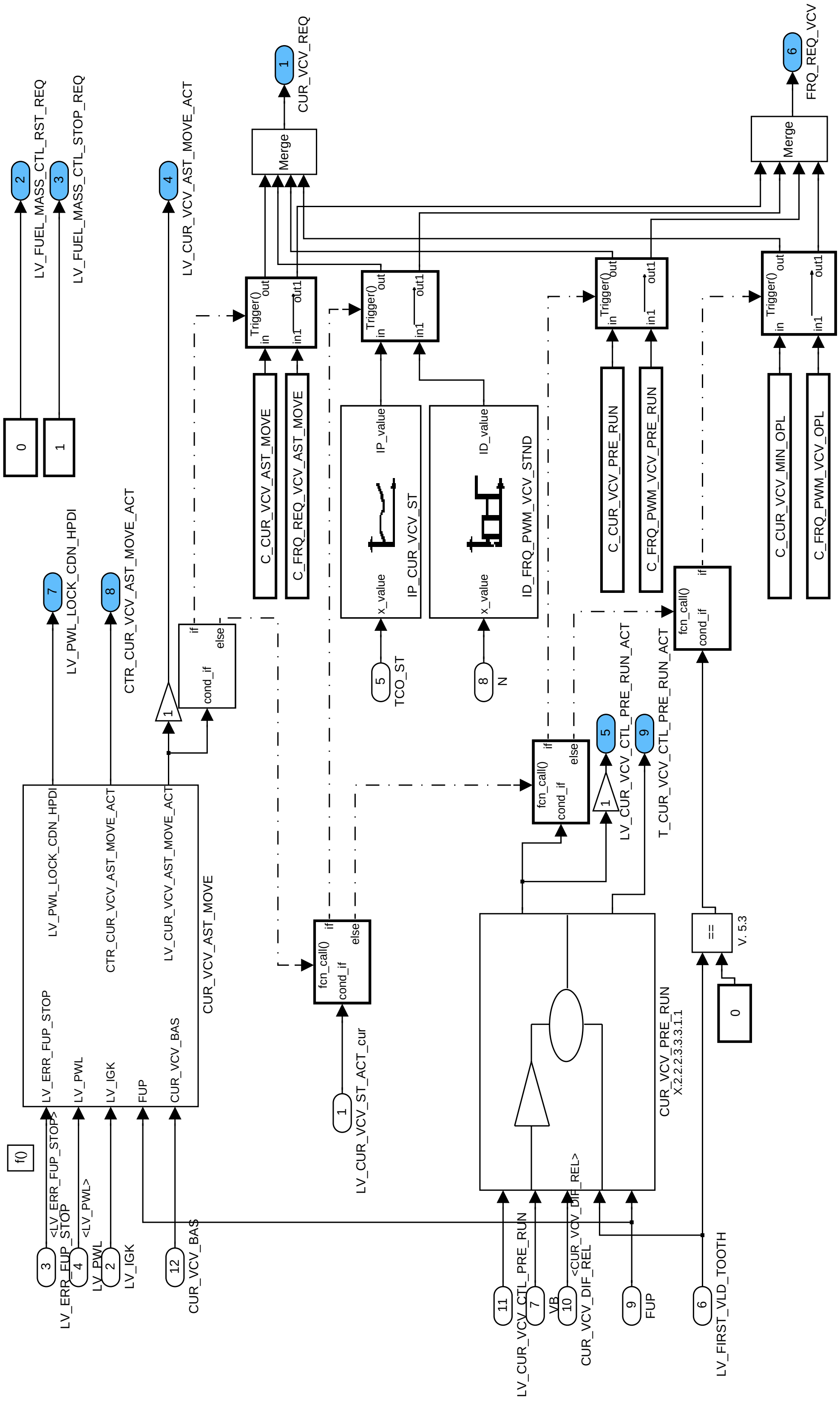


Figure 217:

34.15.2.2.2.1.3.1 Movement of VCV after engine stop.

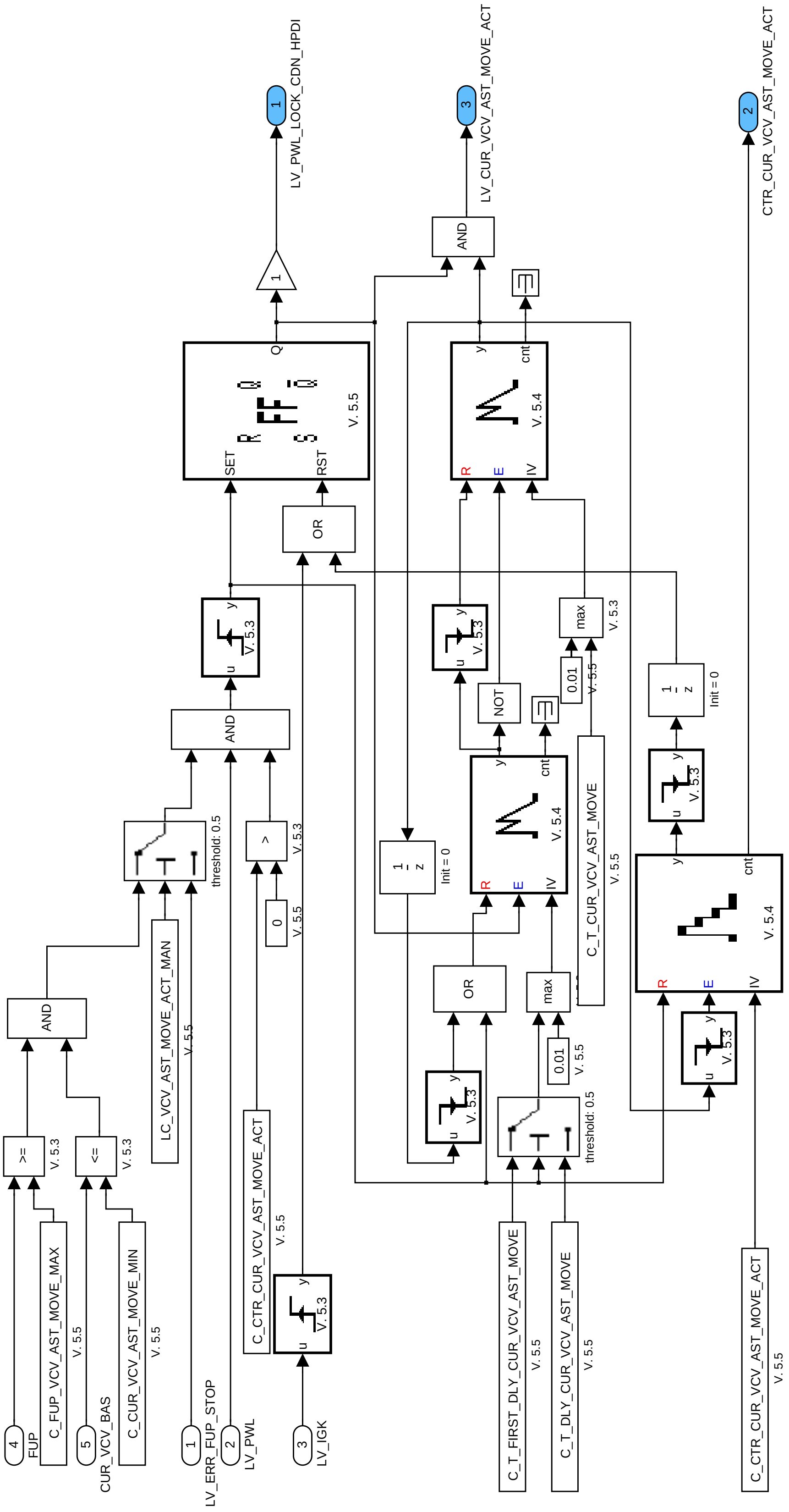


Figure 218:

34.15.2.2.2.1.3.2 Calculation of LV_CUR_VCV_CTL_PRE_RUN_ACT

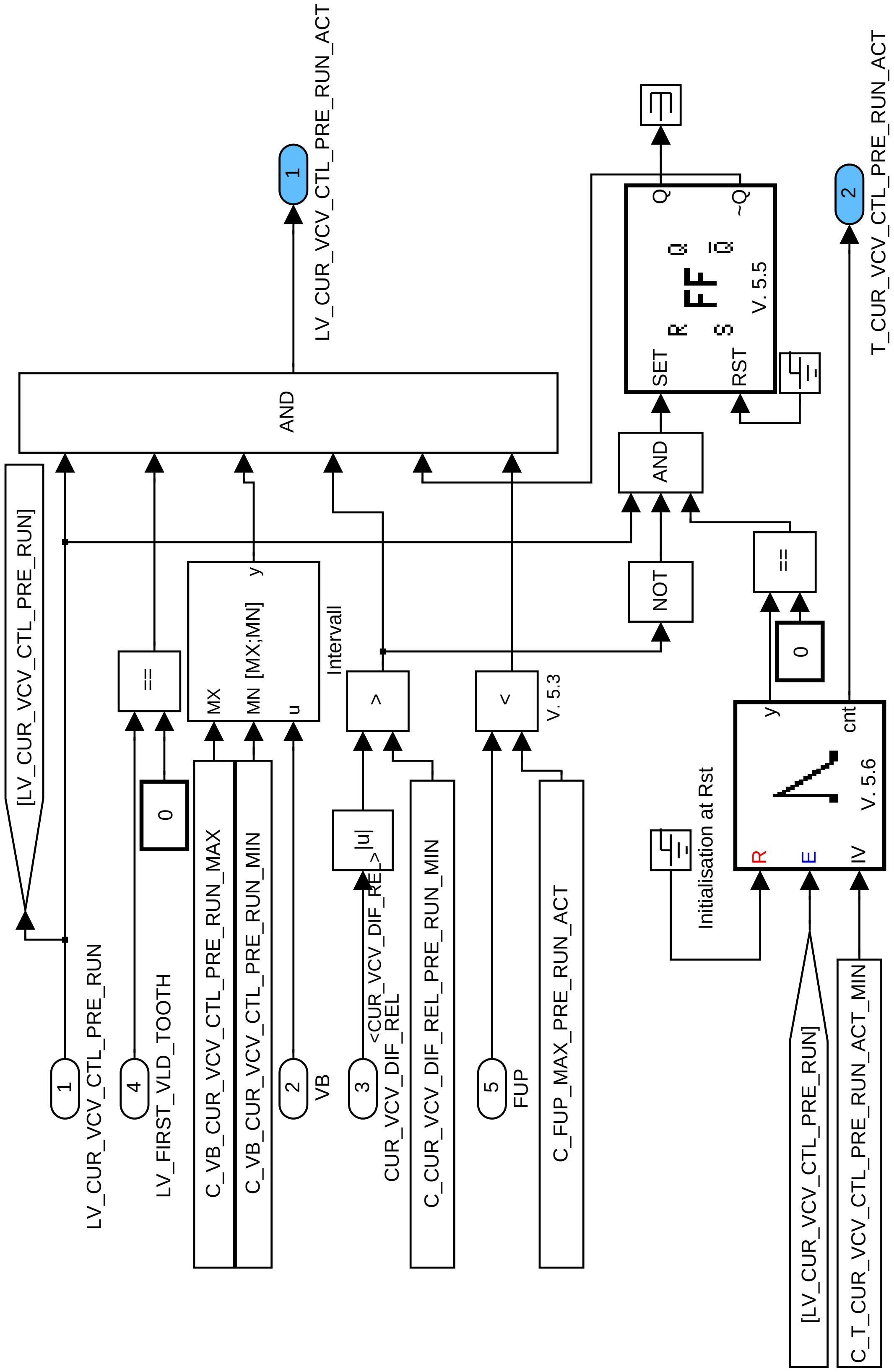


Figure 219:

34.15.2.2.2.1.4 Final calculation of the CUR_VCV

The gradient limitation will only be applied at special states to avoid big PWM jumps. There are several constants for positive PWM change and negative PWM change.

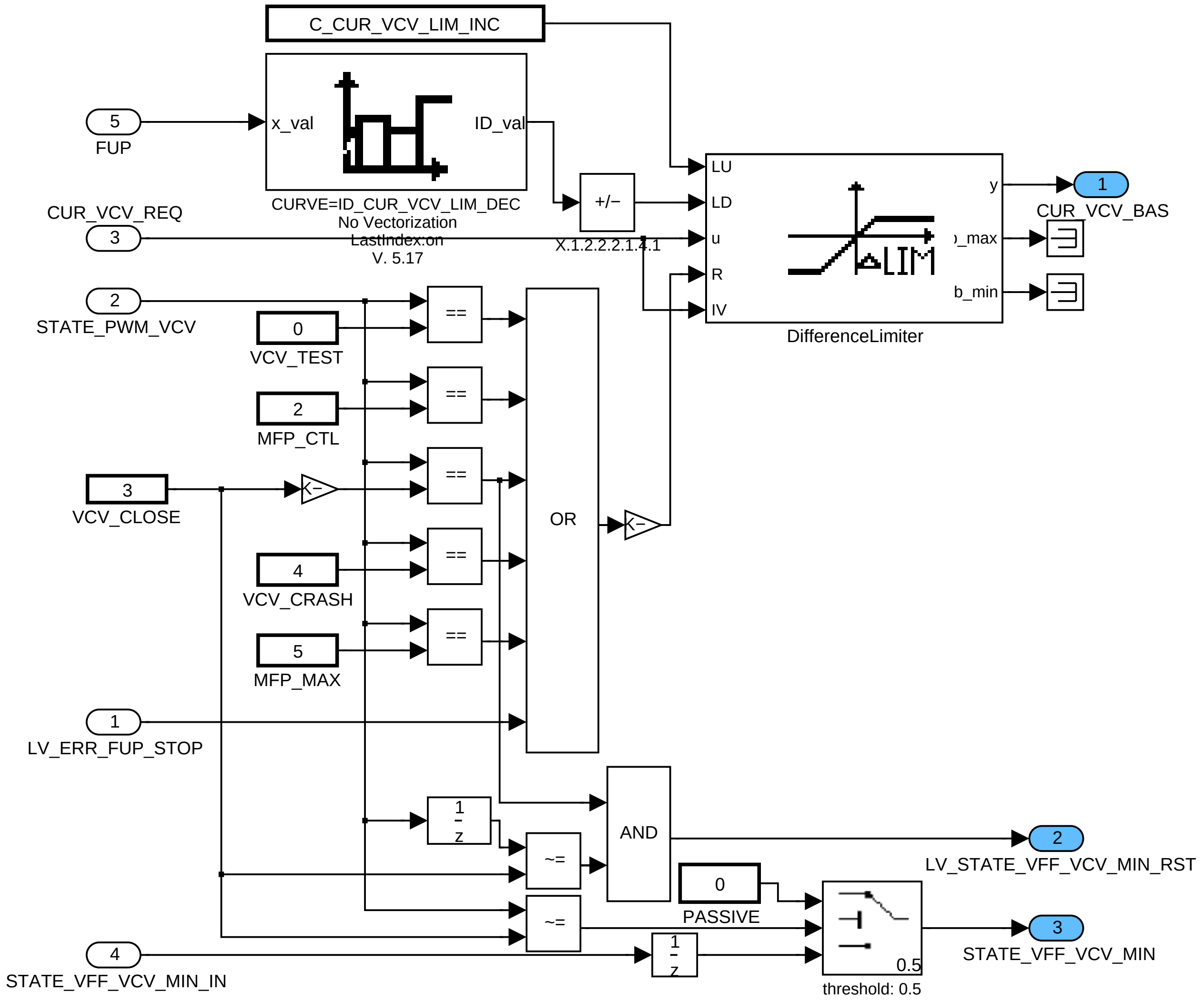
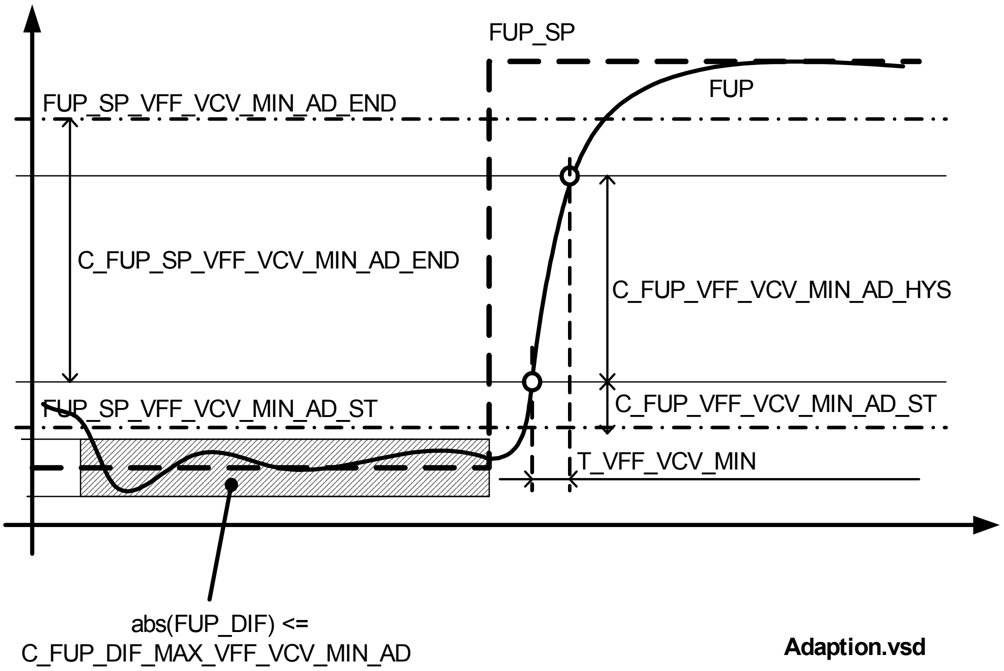


Figure 220:

34.15.2.2.2.1.5 Leakage adaptation in the state 'close'

The minimum flow through the VCV can be adapted by the following principle.



! The internal value of VFF_VCV_MIN_BAS digital lowpass filter has to be initialised by VFF_VCV_MIN_BAS value at reset.

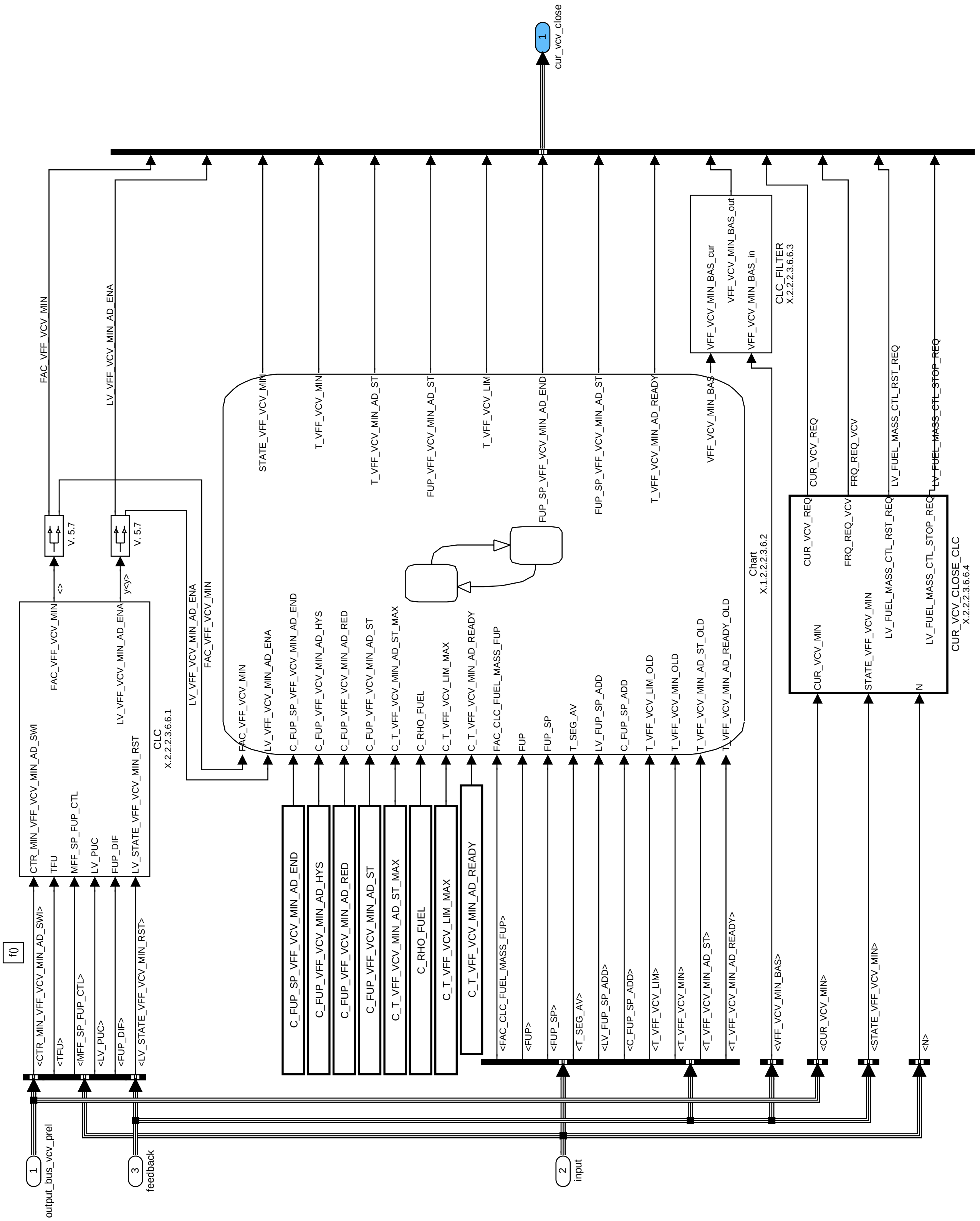


Figure 221:

34.15.2.2.2.1.5.1 FUSL_M904B/OPM/CUR_BAS_CLC/VCV_CTL_CLC/CUR_VCV_REQ_CLC/CUR_VCV_CLOSE/CLC

CONTENT

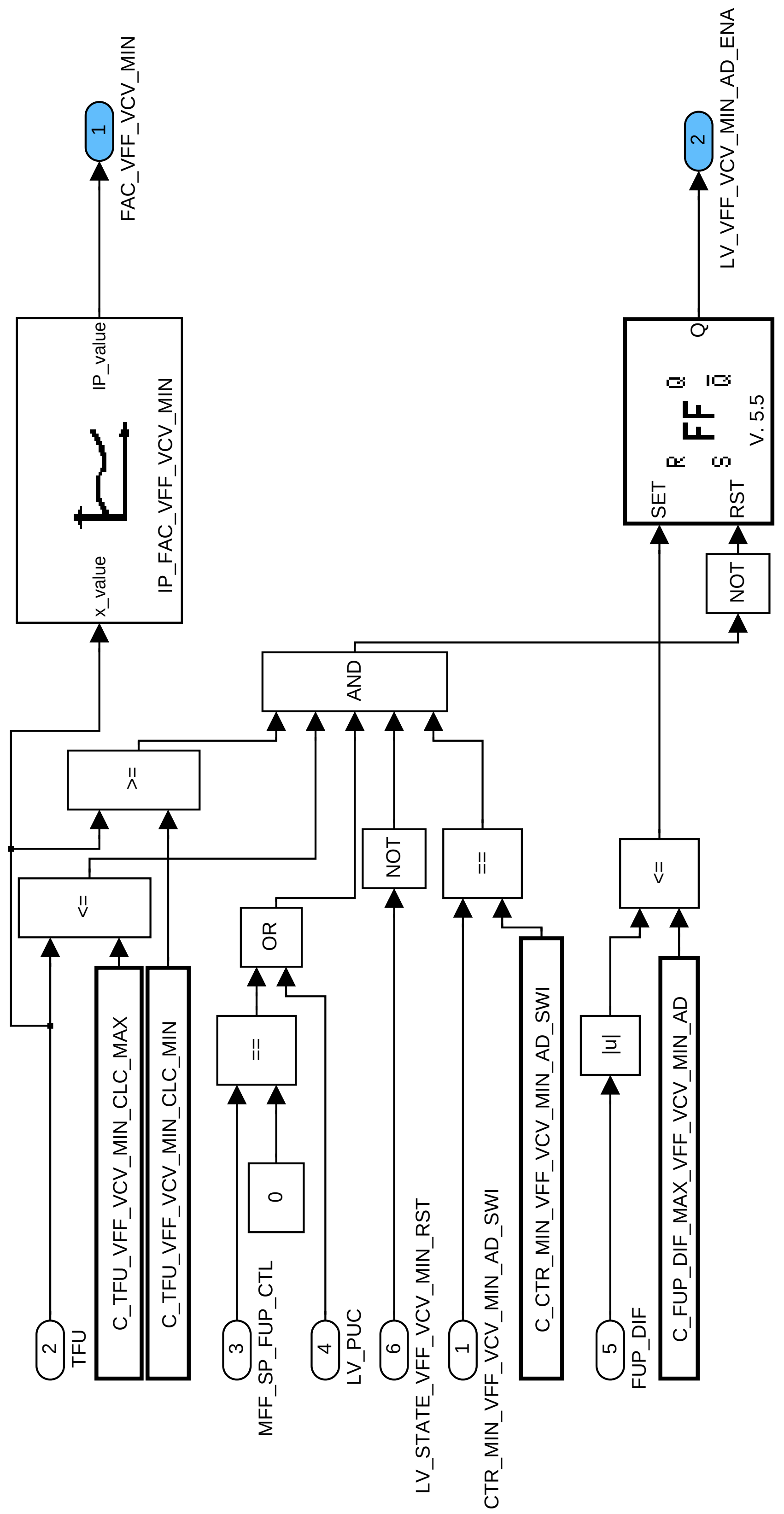

1
FAC_VFF_VCV_MIN
2
LV_VFF_VCV_MIN_AD_ENA
IP_value
x_value
IP_FAC_VFF_VCV_MIN
Q
SET
R
FF
Q
S
Q̄
RST
V. 5.5
NOT
AND
>=
<=
NOT
==
<=
OR
==
0
|u|
2
TFU
C_TFU_VFF_VCV_MIN_CLC_MAX
C_TFU_VFF_VCV_MIN_CLC_MIN
3
MFF_SP_FUP_CTL
4
LV_PUC
6
LV_STATE_VFF_VCV_MIN_RST
1
CTR_MIN_VFF_VCV_MIN_AD_SWI
C_CTR_MIN_VFF_VCV_MIN_AD_SWI
5
FUP_DIF
C_FUP_DIF_MAX_VFF_VCV_MIN_AD

Figure 222:

34.15.2.2.2.1.5.2 FUSL_M904B/OPM/CUR_BAS_CLC/VCV_CTL_CLC/CUR_VCV_REQ_CLC/CUR_VCV_CLOSE/CLC_FILTER

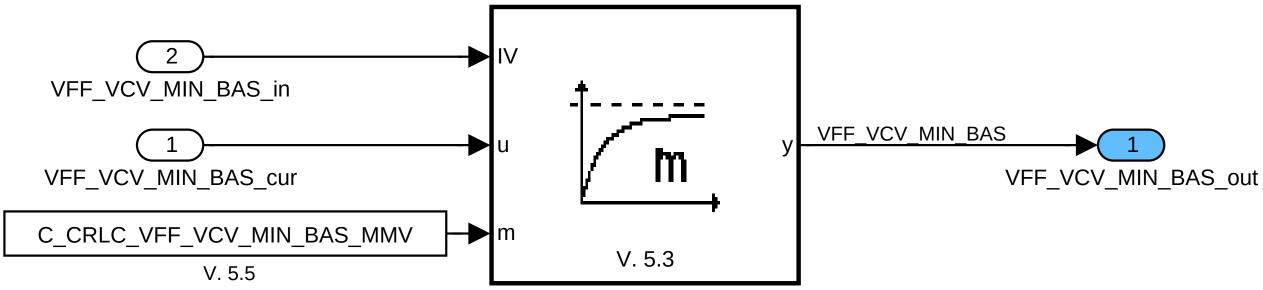


Figure 223:

34.15.2.2.2.1.5.3 FUSL_M904B/OPM/CUR_BAS_CLC/VCV_CTL_CLC/CUR_VCV_REQ_CLC/CUR_VCV_CLOSE/CUR_VCV_CLOSE_CLC

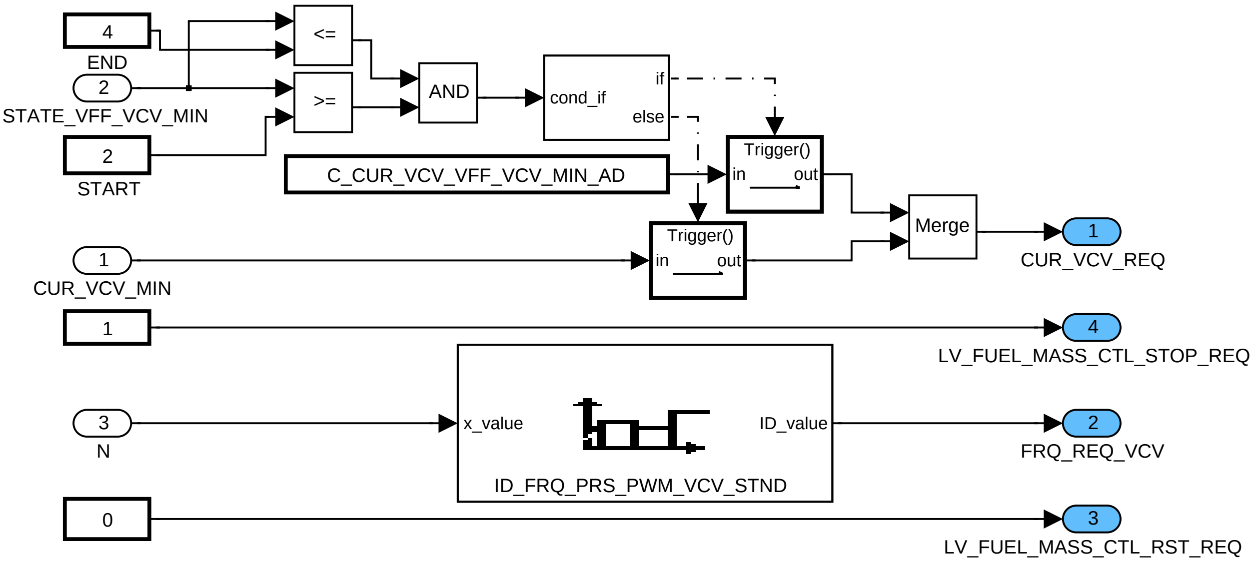


Figure 224:

34.15.2.2.2.1.5.4 Chart

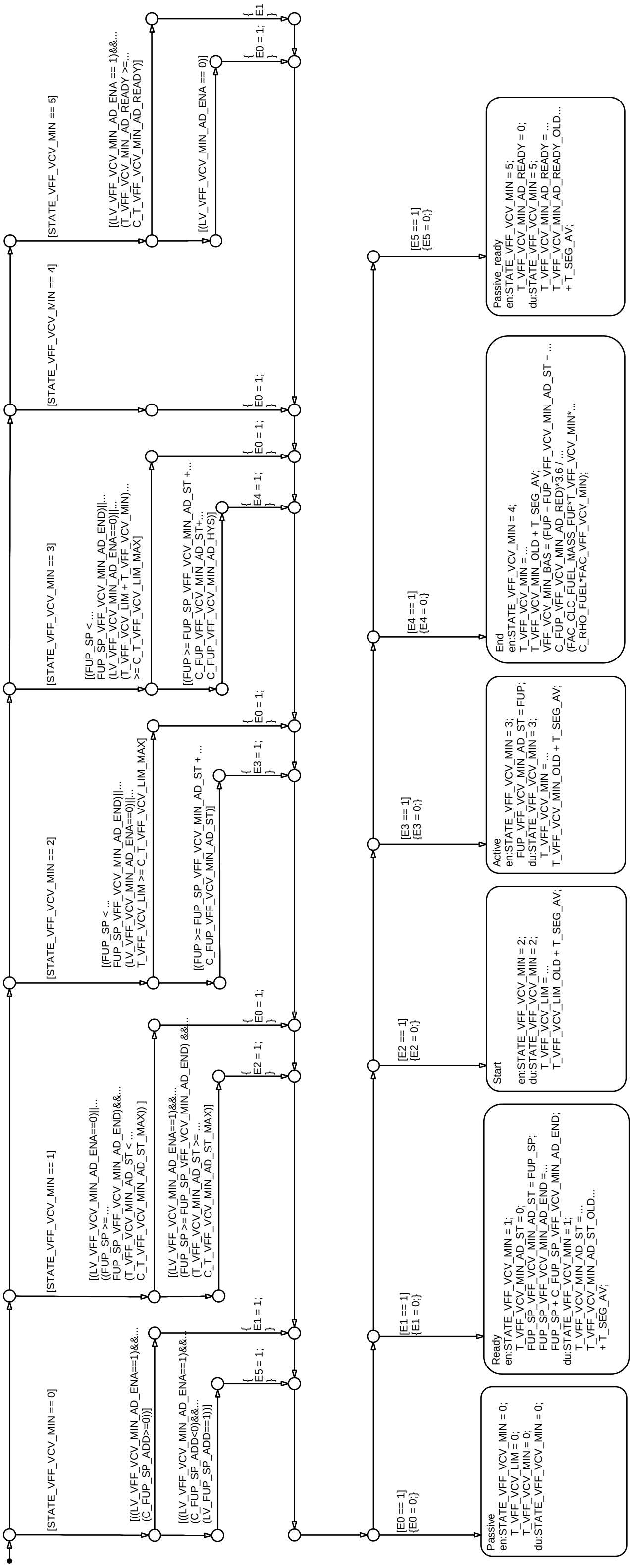


Figure 225:

34.15.2.2.2.1.6 CUR_VCV_CRASH

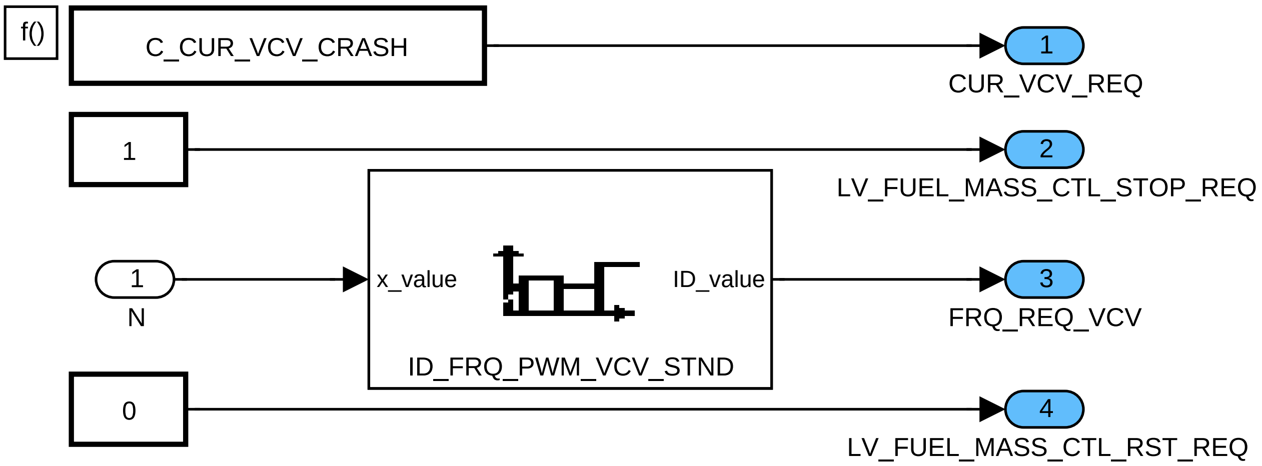


Figure 226:

34.15.2.2.2.1.7 CUR_VCV_LIH

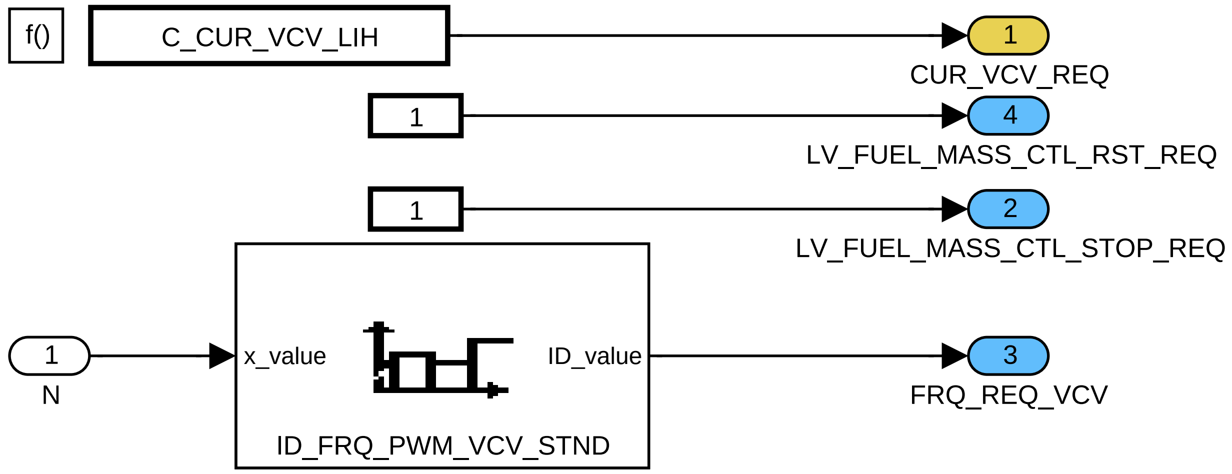


Figure 227:

34.15.2.2.2.1.8 CUR_VCV_TEST

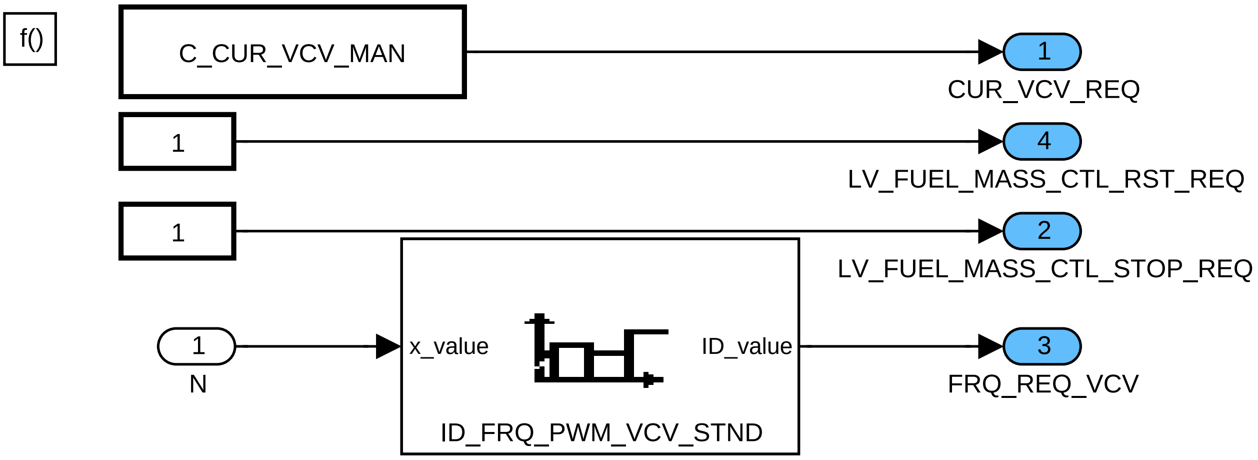


Figure 228:

34.15.2.2.2.1.9 T_CUR_VCV_PRE_RUN_CLC

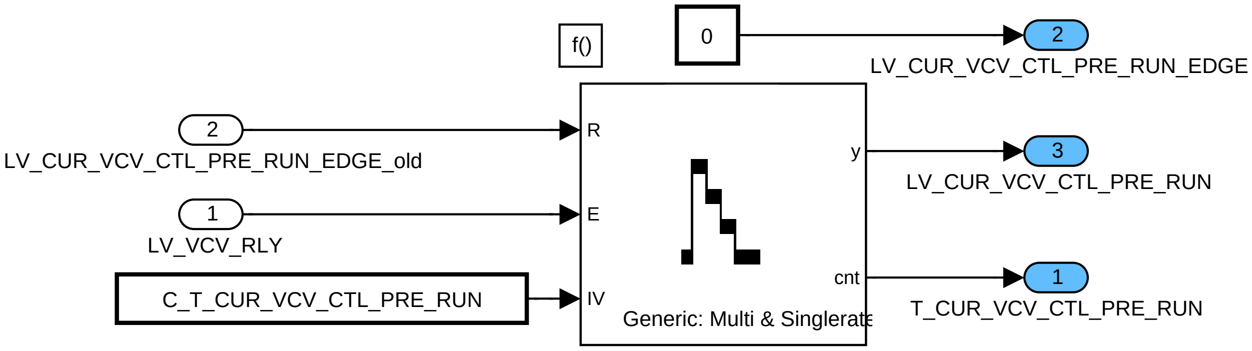


Figure 229:

34.15.2.2.2.2 Chart1

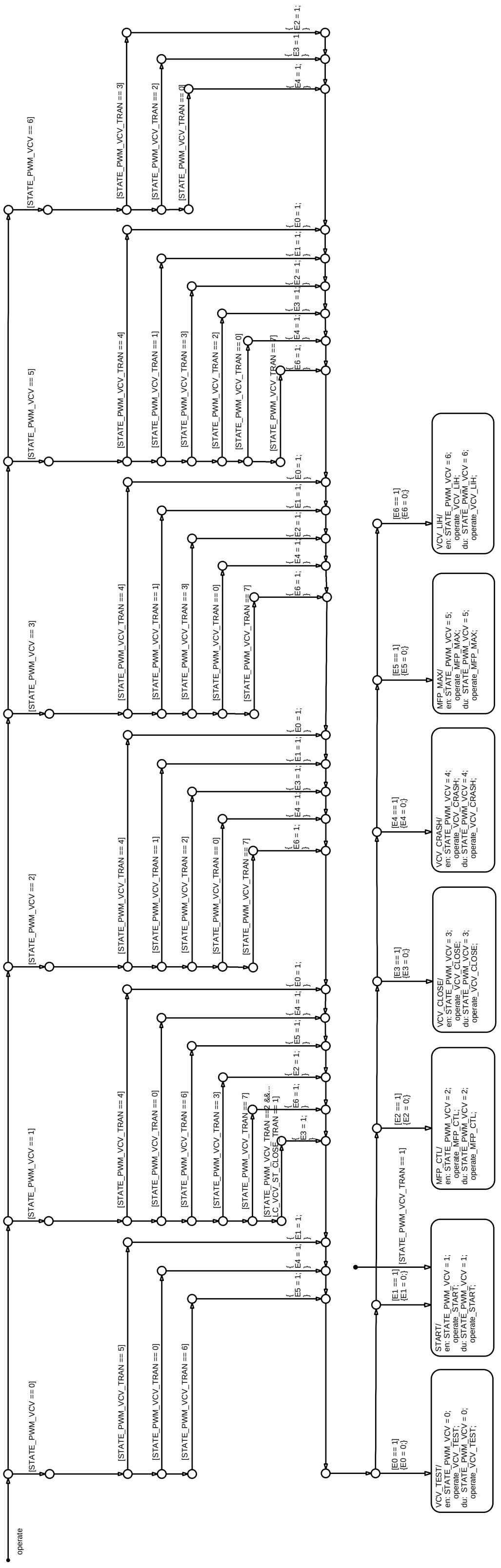


Figure 230:

34.15.2.2.2.3 Calculation of LV_CUR_VCV_ST_ACT_CLC

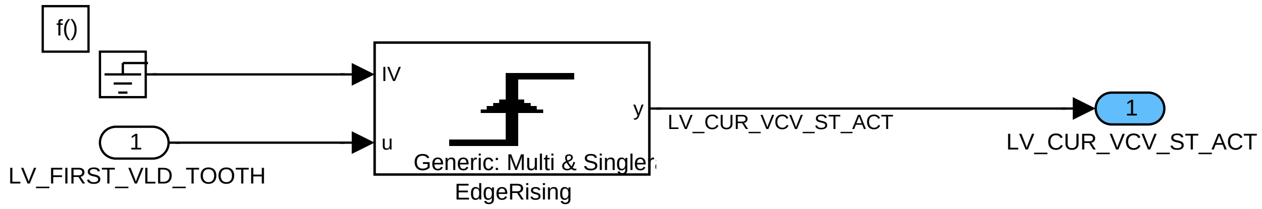


Figure 231:

34.15.2.3 Overview of VCV current controller

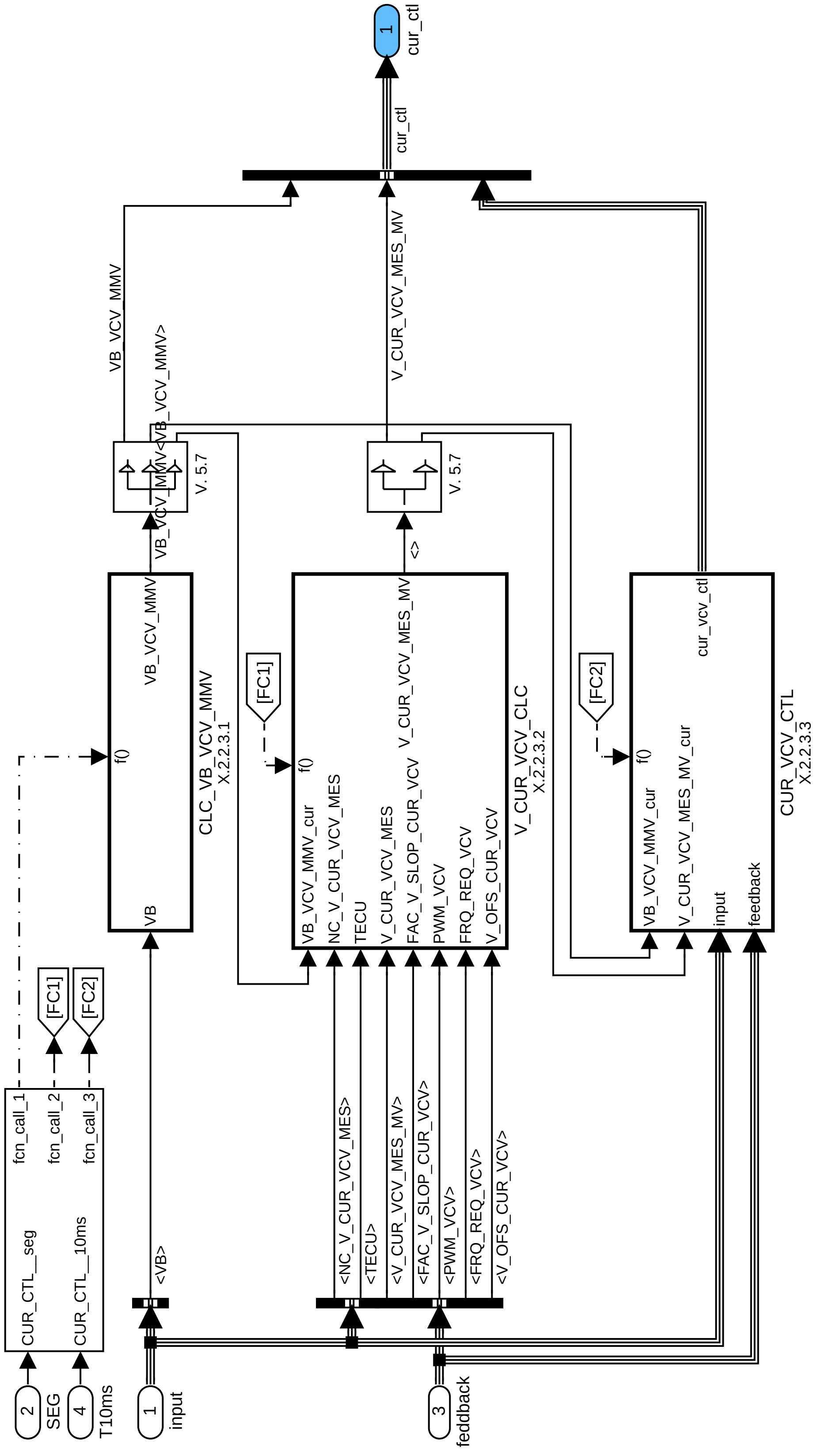


Figure 232:

34.15.2.3.1 Calculation of VB_VCV_MMV

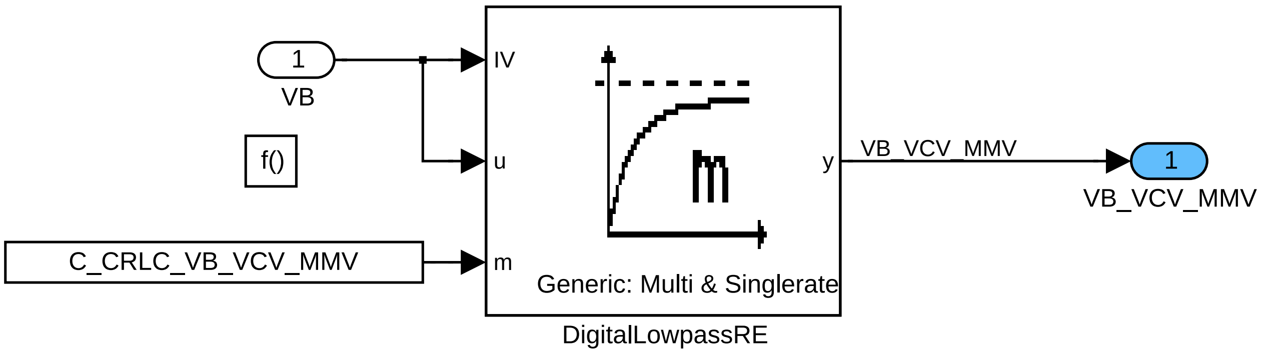


Figure 233:

34.15.2.3.2 FUSL_M904B/OPM/CUR_CTL/CUR_VCV_CTL

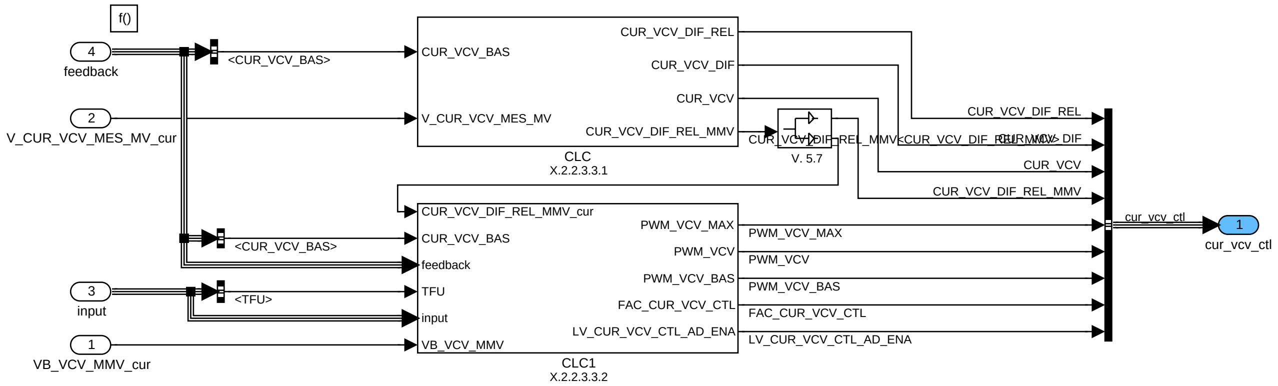


Figure 234:

34.15.2.3.2.1 FUSL_M904B/OPM/CUR_CTL/CUR_VCV_CTL/CLC

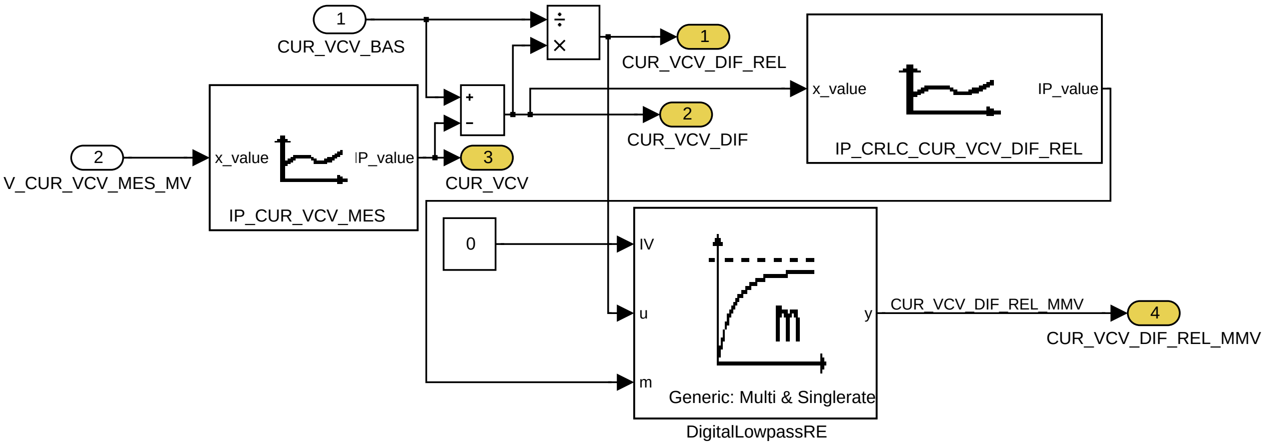


Figure 235:

34.15.2.3.2.2 Calculation 1

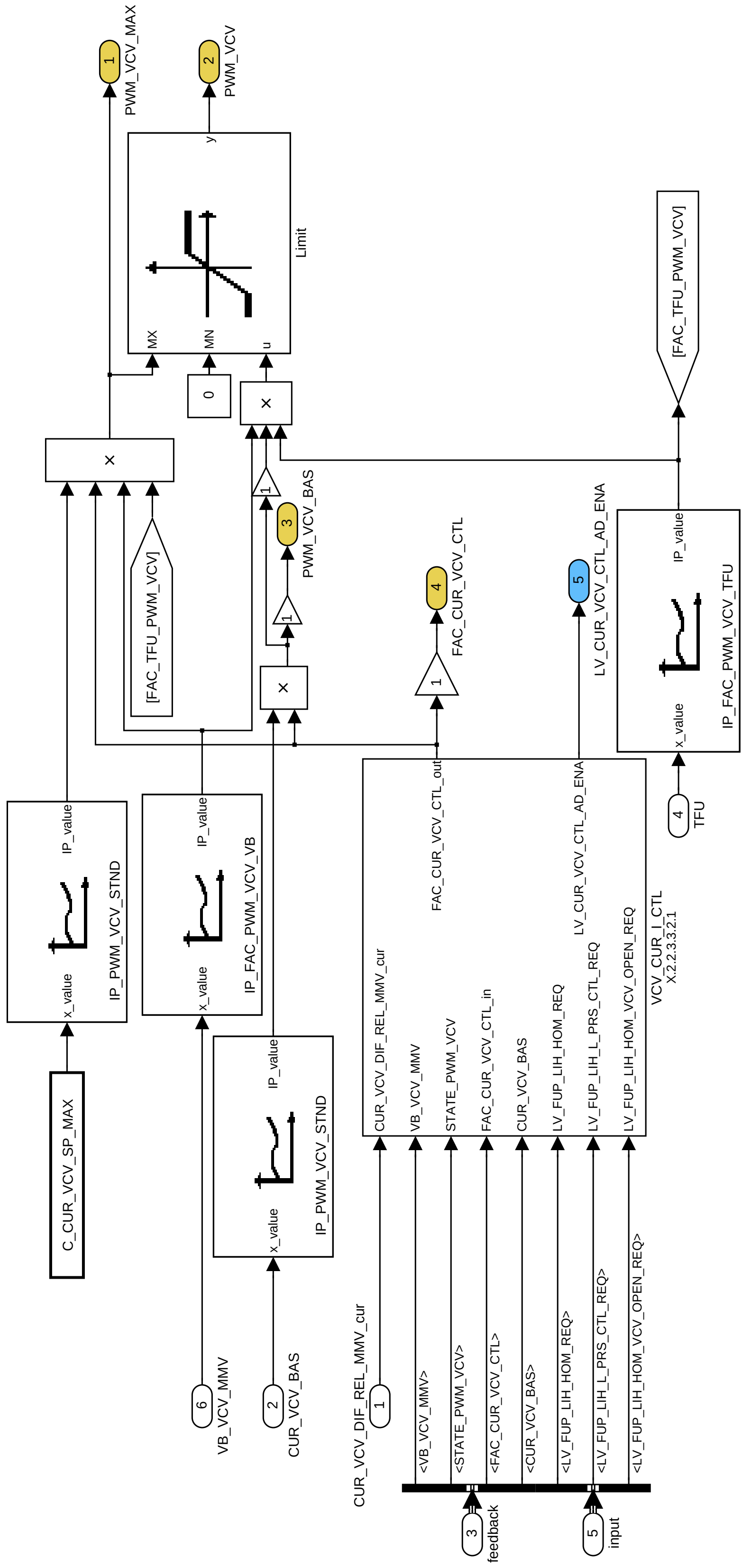


Figure 236:

34.15.2.3.2.2.1 VCV_CUR_I_CTL

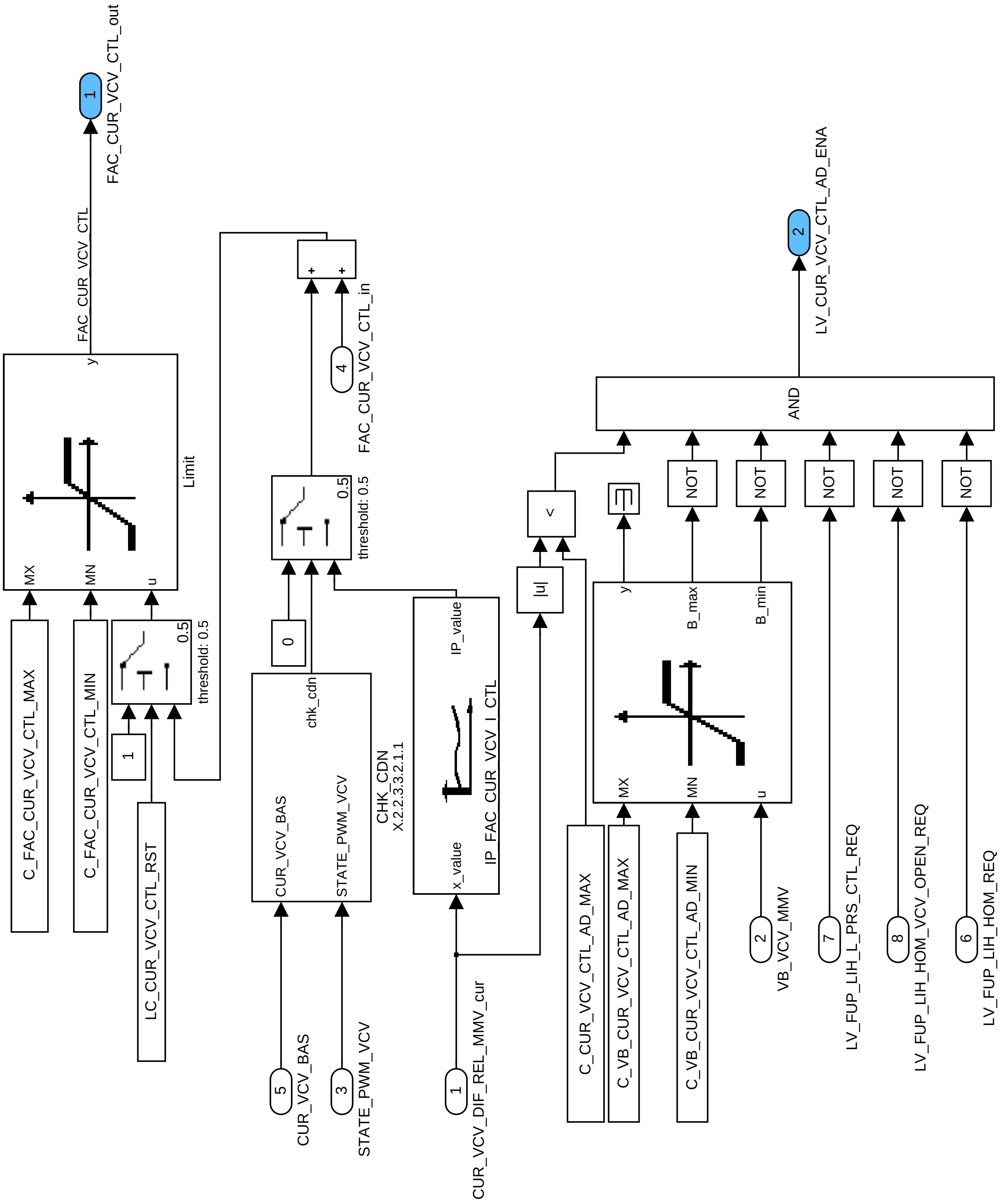


Figure 237:

34.15.2.3.2.2.1.1 Condition Check

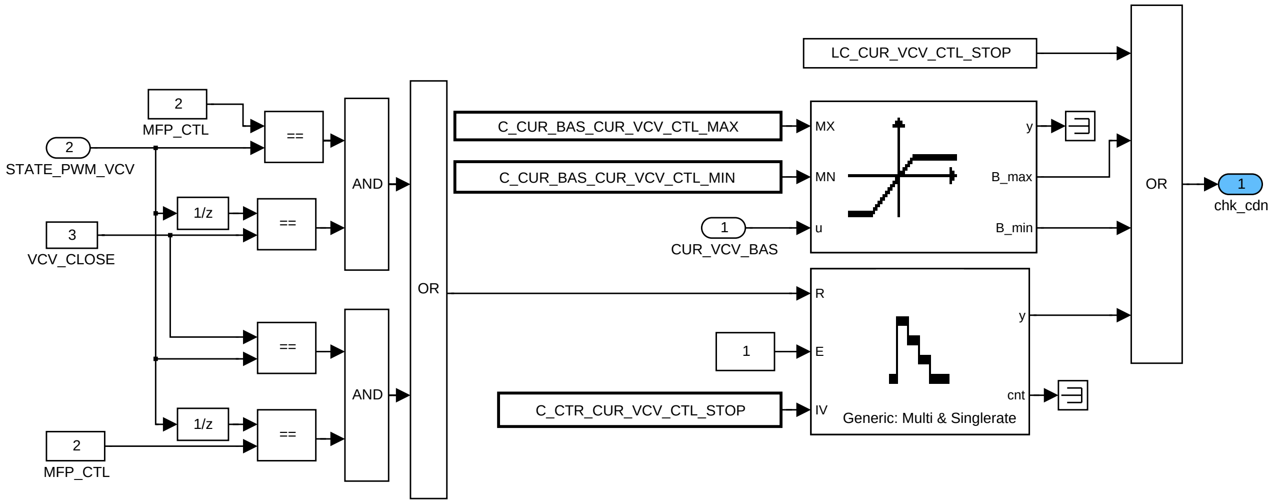


Figure 238:

34.15.2.3.3 Calculation of V_CUR_VCV_CLC

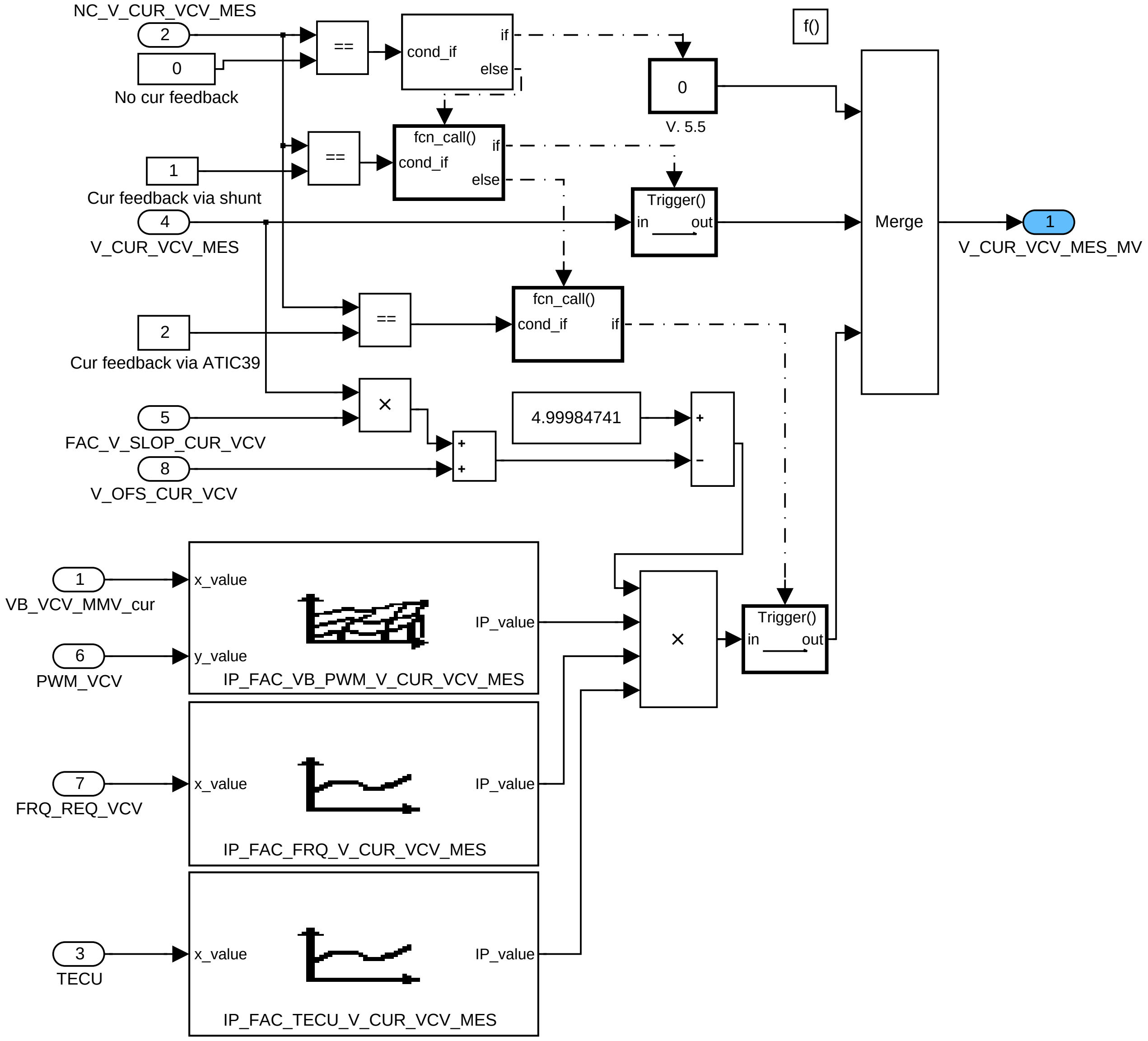


Figure 239:

34.16 High pressure pump control (Appl. Inc.)

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_FUEL_MASS_CTL_STOP_DIAG	O/V	0... 1H	0... 1	1	[-]
Logical variable freeze the controller while the limphome phase					
LV_FUP_LIH_HOM_REQ	O/V	0... 1H	0... 1	1	[-]
Logical variable indicating LIH mode for HPP requested					
LV_FUP_LIH_HOM_VCV_OPEN_REQ	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting limphome for VCV open					
LV_FUP_LIH_L_PRS_CTL_REQ	O/V	0... 1H	0... 1	1	[-]
Logical variable requesting fuel low pressure control limphome					

Input Data:

FUP	LV_ERR_CRASH_SIG	LV_ERR_FUP	LV_ERR_FUP_MFP_PLAUS
LV_ERR_FUP_ORNG	LV_ERR_H_PRS_SYS	LV_ERR_VCV	LV_ERR_VCV_PLAUS
LV_ES	ERR_SYM_H_PRS_SYS		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_FUP_LIH_VCV_OPEN	1	0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure threshold for switching to VCV open limphome mode					
C_T_FUP_LIH_HOM_VCV_OPEN_REQ	1	0... FFFFH	0... 655.35	0.01	[s]
Timer for activation of the VCV open limphome					
C_T_FUP_LIH_L_PRS_CTL_REQ	1	0... FFFFH	0... 655.35	0.01	[s]
Timer for freeze the controller in FUP_LIH_L_PRS_CTL					
LC_T_FUP_LIH_L_PRS_CTL_REQ	1	0... 1H	0... 1	1	[-]
Switch for enabling of the timer for freeze the controller in FUP_LIH_L_PRS_CTL					

General Information

In this module the limphome bits for the high pressure pump control are calculated.

Application Conditions

Initialization: all outputs to zero at RST
Recurrence: 10MS
Activation: always
Deactivation: never

Function description

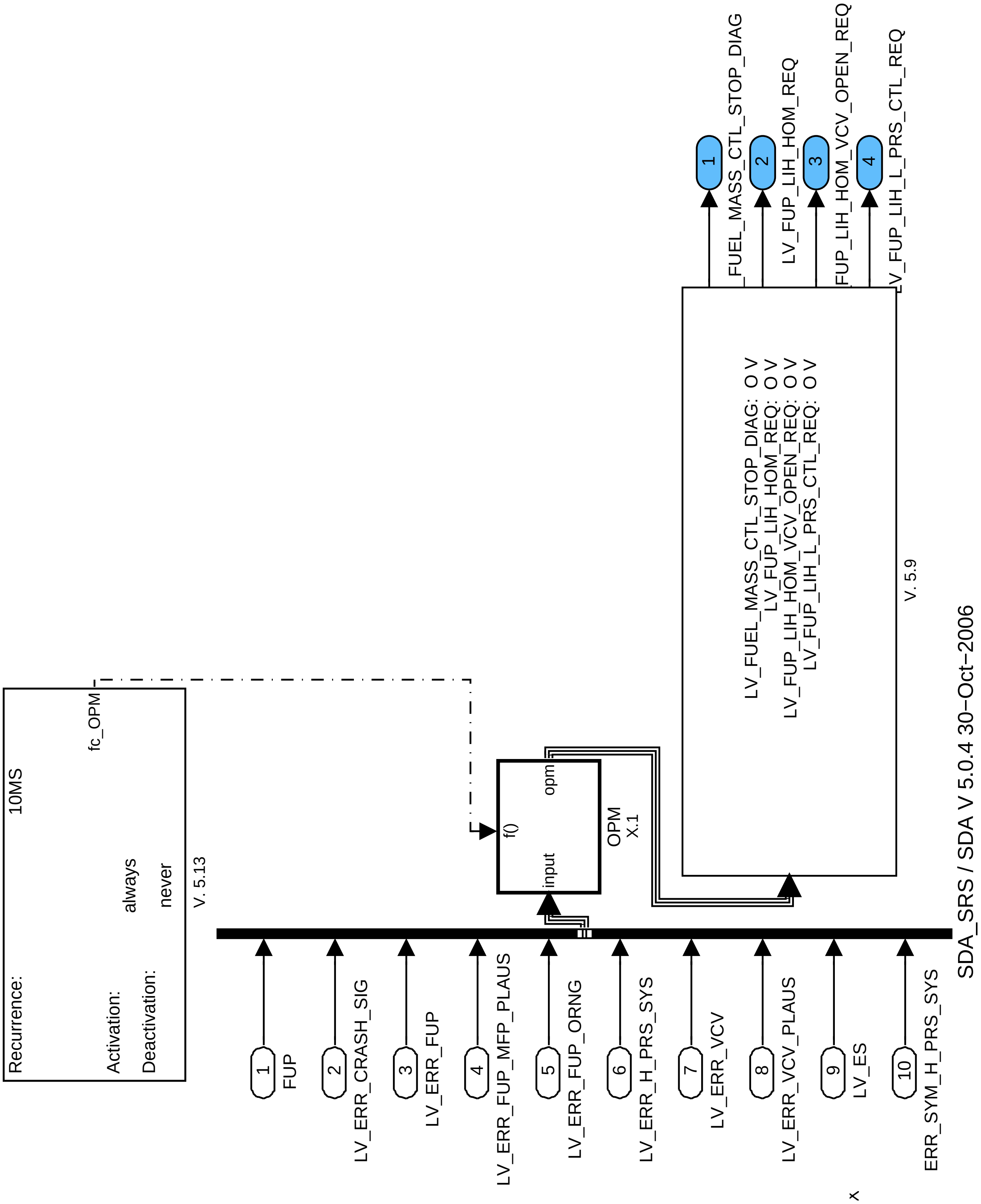


Figure 240:
Path: FUSL_M904R

34.16.1 Formula Section

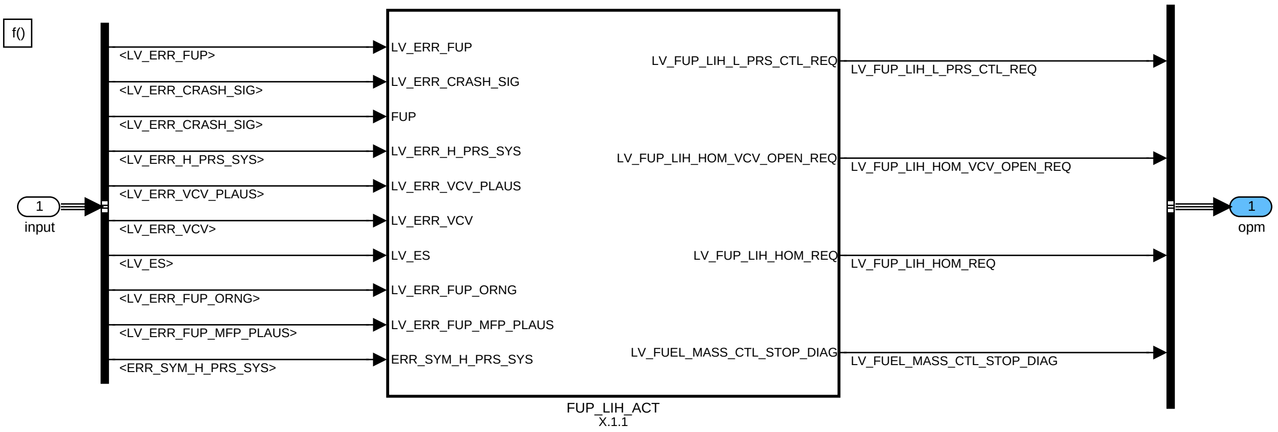


Figure 241:
Path: FUSL_M904R/OPM

34.16.1.1 FUP_LIH_ACT

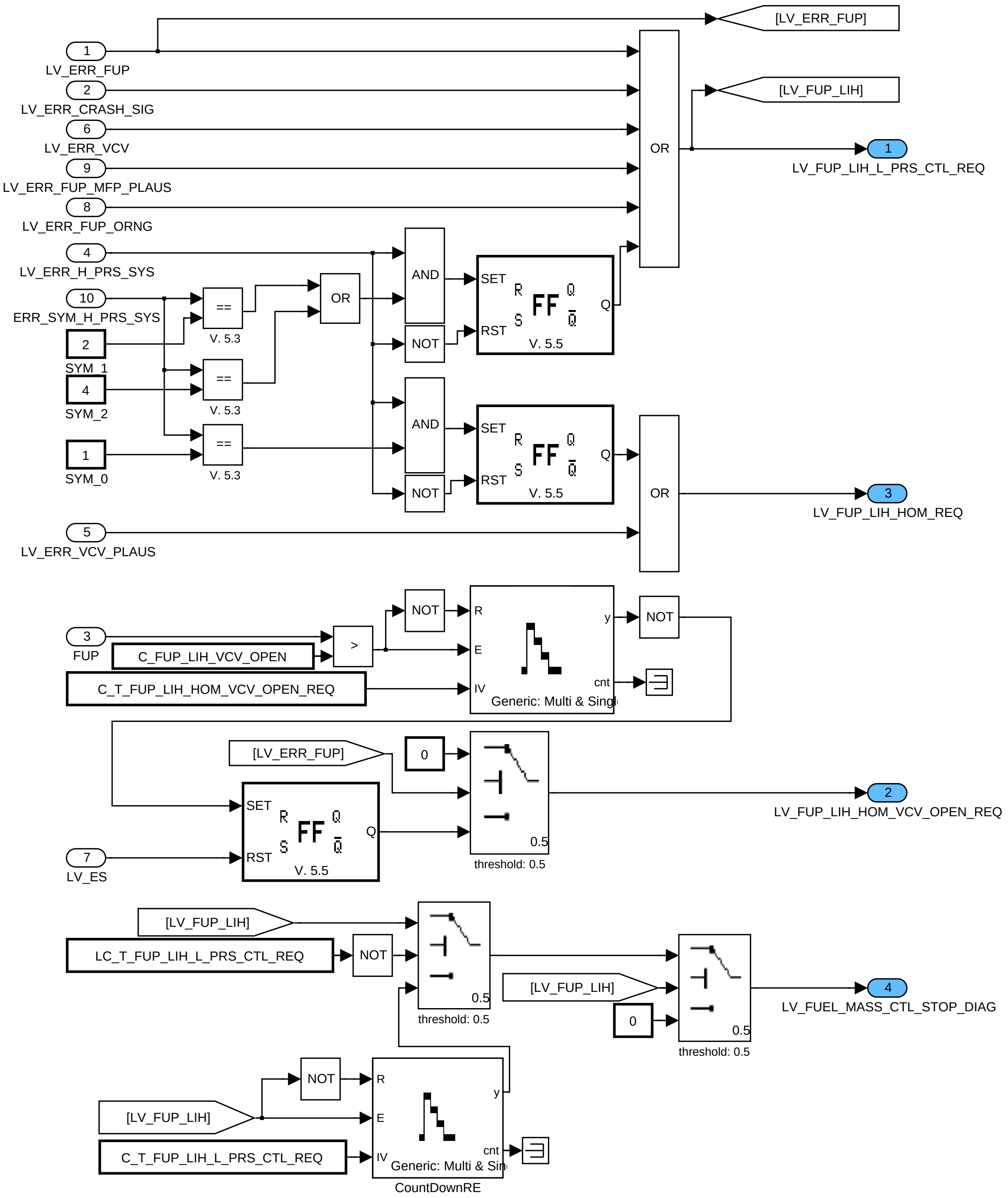


Figure 242:
Path: FUSL_M904R/OPM/FUP_LIH_ACT

34.17 Fuel pressure actuator temperature

Output data:

Name	Mode	Hex	Phys	Res.	Unit
TFPA	O/V	0... FEH	-48... 142.5	0.75	°C
Fuel pressure acutuator temperature					

Input data:

TFU			

34.17.1 Fuel pressure actuator temperature at available fuel temperature

General Information

In case of an available temperature of the fuel either modelled or measured this temperature is taken as a substitute for the temperature for the fuel pressure actuator

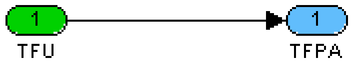
Application conditions:

Initialisation: at reset: TFPA = TFU

Recurrence: 1 s

Activation: every engine state

FUNCTION DESCRIPTION:



34.18 Diagnosis of EFP (via PT-CAN)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_EFP	O/V	0...1H	0...1	1	[-]
Logical variable error in case of EFP-error (selfdiagnosis EFP via CAN)					
LV_CDN_DIAG_EFP	O/V	0...1H	0...1	1	[-]
Diagnosis condition EFP diagnosis					
ERR_SYM_EFP	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom EFP diagnosis					
LV_END_DIAG_EFP	O/V	0...1H	0...1	1	[-]
End of Diagnosis EFP diagnosis					
LV_ERR_EFP_CRASH	O/V	0...1H	0...1	1	[-]
logical variable error in case of crash switch off EFP					
LV_CDN_DIAG_EFP_CRASH	O/V	0...1H	0...1	1	[-]
Diagnosis condition CRASH_EFP diagnosis					
ERR_SYM_EFP_CRASH	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom CRASH_EFP diagnosis					
LV_END_DIAG_EFP_CRASH	O/V	0...1H	0...1	1	[-]
End of Diagnosis CRASH_EFP diagnosis					

Input data:

LV_CDN_VB_CAN_DIAG	CONF_SWI_EFP_OUT	LV_IGK	STATE_DIAG_EFP
LV_VAR_BN	LV_ERR_CAN_BOFF	LV_ERR_BN_EFP	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_EFP	1	0...FFH	0...255	1	[-]
Increment of EFP-error					
C_ABC_MAX_EFP	1	0...FFH	0...255	1	[-]
MAX-threshold EFP-error					
C_ABC_INC_EFP_CRASH	1	0...FFH	0...255	1	[-]
Increment of EFP-error					
C_ABC_MAX_EFP_CRASH	1	0...FFH	0...255	1	[-]
MAX-threshold EFP-error					

FUNCTION DESCRIPTION

General information:

In case of EKP on PT-CAN (CONF_SWI_EFP_OUT = 2), the EKP sends diagnosis-information to the DME, the following diagnoses are defined:

- EFP Selfdiagnosis
- EFP switch off in case of vehicle crash

34.18.1 EFP selfdiagnosis

General information:

The state of the self diagnosis of the EFP module is transmitted via PT-CAN. The error handling for the diagnosis results will be done by the ECU.

Description:

Error symptoms are defined for this diagnosis function as:

```
□□□□
 │ │ │ └─> N_high                       (= SYM_0)
 │ │ └───> N_Low                        (= SYM_1)
 │ └─────> LIH                          (= SYM_2)
 └───────> Temp                         (= SYM_3)
```

Application conditions:

Initialisation: *All values are initialized according Filter-type STD_INI*

Recurrence: *1s*

Activation:

```
If      LV_CDN_VB_CAN_DIAG =  1         and
        CONF_SWI_EFP_OUT > 0            and
        {[LV_VAR_BN = 0                 and
        (CONF_SWI_EFP_OUT = 2           and
        LV_ERR_TOUT_EFP_CAN = 0         and
        LV_ERR_CAN_BOFF = 0)]           or
        [LV_VAR_BN = 1                  and
         LV_VAR_BN_EFP = 1              and
        LV_ERR_BN_EFP = 0               and
        LV_ERR_CAN_BOFF = 0]}
Then    LV_CDN_DIAG_EFP = 1
        LV_END_DIAG_EFP = 1
         (Calculated by error management)
Else    LV_CDN_DIAG_EFP =  0
Endif
```

Formula section:

Calculation of detected error

```
If      STATE_DIAG_EFP = 3H
Then    ERR_SYM_EFP = SYM_0        Symptom rpm to high (N_high)
```

```
ElseIf   STATE_DIAG_EFP = 2H
Then     ERR_SYM_EFP = SYM_1          Symptom rpm to low (N_low)
ElseIf   STATE_DIAG_EFP = 5H
Then     ERR_SYM_EFP = SYM_2          Symptom limp home mode (LIH)
ElseIf   STATE_DIAG_EFP = 4H
Then     ERR_SYM_EFP = SYM_3          Symptom high temperature (Temp)
Else     ERR_SYM_EFP = NO_SYM
Endif
```

Calculation of detected error

```
If       ERR_SYM_EFP is not 0
Then     LV_ERR_EFP = 1     set after debounce
Else     LV_ERR_EFP = 0     reset after rebounce
Endif
```

Configuration for diagnostic symptoms :

Diagnostic EFP	Symptom description	Symptom	Filter type
EFP Selfdiagnosis	*Symptom rpm to high (N_high)*	*SYM_0*	*STD_INI*
	Symptom rpm to low (N_low)	*SYM_1*	
	Symptom limp home mode (LIH)	*SYM_2*	
	Symptom high temperature (Temp)	*SYM_3*	

// calculation of end of diagnosis according to ERRM

34.18.2 EFP switch off in case off vehicle crash

General information:

The state of the self diagnosis of the EFP module is transmitted via PT-CAN. The error handling for the diagnosis results will be done by the ECU.

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                 (= SYM_0)
 │ │ └───> -                 (= SYM_1)
 │ └─────> Crash switch off  (= SYM_2)
 └───────> -                 (= SYM_3)
```

Application conditions:

Initialisation: *All values are initialized according Filter-type STD_INI*

Recurrence: *1s*

Activation:

```
If    LV_CDN_VB_CAN_DIAG =  1         and
      CONF_SWI_EFP_OUT > 0            and
      {[LV_VAR_BN = 0                 and
      (CONF_SWI_EFP_OUT = 2           and
      LV_ERR_TOUT_EFP_CAN = 0         and
      LV_ERR_CAN_BOFF = 0)]           or
      [LV_VAR_BN = 1                  and
       LV_VAR_BN_EFP = 1              and
      LV_ERR_BN_EFP = 0               and
      LV_ERR_CAN_BOFF = 0]}
Then  LV_CDN_DIAG_EFP_CRASH = 1
      LV_END_DIAG_EFP_CRASH = 1
       (Calculated by error management)
Else  LV_CDN_DIAG_EFP_CRASH =  0
Endif
```

Formula section:

Calculation of detected error

```
If      STATE_DIAG_EFP = 1H
Then    ERR_SYM_EFP_CRASH = SYM_2 Symptom switch off due to vehicle crash
        LV_ERR_EFP_CRASH = 1            set after debounce
Else    ERR_SYM_EFP_CRASH = NO_SYM
        LV_ERR_EFP_CRASH = 0            reset after rebounce
Endif
```

// calculation of end of diagnosis according to ERRM

Configuration for diagnostic symptoms :

Diagnostic EFP_CRASH	**Symptom description**	**Symptom**	**Filter type**
EFP switch off in case of vehicle crash	*Symptom switch off due to vehicle crash*	*SYM_2*	*STD_INI*

34.19 Fuel pressure sensor diagnosis (FUP_EFP)

Output data:

Name	Mod.	Hex. Limit	Phys. Limit	Resol.	Unit
LV_ERR_FUP_EFP	V/O	0...1H	0...1	1	[-]
FUP sensor error detected					
LV_CDN_DIAG_FUP_EFP	V	0...1H	0...1	1	[-]
Status of diagnosis for FUP sensor					
ERR_SYM_FUP_EFP	V	0H 1H 2H	NO_SYM SYM_0 SYM_1	1	[-]
Detected symptom FUP sensor					
LV_END_DIAG_FUP_EFP	V	0...1H	0...1	1	[-]
End of diagnosis for FUP sensor diagnosis					

Input data:

V_FUP_EFP_MV	LV_IGK	LV_INH_DIAG_FUP_EFP	

Import actions:

ACTION_ERRM_FilterSymptom(IN< XX >, IN< LV_CDN_DIAG_XX >, IN< ERR_SYM_XX >,
IN< C_ABC_INC_XX >, IN< C_ABC_DEC_XX >, IN< C_ABC_MAX_XX >, OUT< LV_ERR_XX >)

FUNCTION DESCRIPTION:

General Information:

Analog input signal in the A/D-Input from the Microprocessor.

Description

Error-symtoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> Short circuit in signal wire to VB or wire break      (= SYM_0)
 │  │  └───> Short circuit in signal wire to ground                 (= SYM_1)
 │  └─────> -                                                        (= SYM_2)
 └───────> -                                                         (= SYM_3)
```

Remark: Calculation of LV_END_DIAG_FUP_EFP see generic calculation “End of diagnosis” in anti bounce algorithm.

Application conditions:

Initialization:
LV_CDN_DIAG_FUP_EFP = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_FUP_EFP = *Refer to filtering configuration for the initialisation value*
LV_ERR_FUP_EFP = *Refer to filtering configuration for the initialisation value*

ERR_SYM_FUP_EFP = *Refer to filtering configuration for the initialisation value*

Recurrence: 40 ms

Activation:

```
if      LV_IGK = 1 AND LV_INH_DIAG_FUP_EFP = 0
Then    LV_CDN_DIAG_FUP_EFP = 1
        (diagnosis is active)
else    LV_CDN_DIAG_FUP_EFP = 0
        (diagnosis is inactive)
endif
```

Deactivation:

When the activation condition is not fulfilled then LV_CDN_DIAG_FUP_EFP = 0.

Formula section:

```
{ SYM_0 : Short circuit in signal wire to VB or wire break }
    If      V_FUP_EFP_MV > C_V_FUP_EFP_MV_MAX_DIAG
    then
            ERR_SYM_ FUP_EFP = SYM_0            { Symptom SCP or OC is active }

    else
        { SYM_1 : Short circuit in signal wire to ground }
        if      V_FUP_EFP_MV      <      C_V_FUP_EFP_MV_MIN_DIAG
                then
                ERR_SYM_FUP_EFP = SYM_1     { Symptom SCG is active }
                else
                ERR_SYM_FUP_EFP = NO_SYM    { No failure has been detected }
                endif
    endif
```

Symptoms are **unchanged** if condition for doing symptopm detection is not fulfilled.

For failure and error management treatment the anti-bounce mechanism is called with the parameters LV_CDN_DIAG_FUP_EFP and ERR_SYM_FUP_EFP.

ACTION_ERRM_FilterSymptom (IN< FUP_EFP >, IN< LV_CDN_DIAG_FUP_EFP >,
IN< ERR_SYM_FUP_EFP >, IN< C_ABC_INC_FUP_EFP >, IN< 1 >,
IN< C_ABC_MAX_FUP_EFP >, OUT< LV_ERR_FUP_EFP >)

This algorithm determines LV_ERR_FUP_EFP and LV_END_DIAG_FUP_EFP and delivers the result to Error Management.

Calibration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
C_V_FUP_EFP_MV_MAX_DIAG	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Threshold value V_FUP_EFP_MV to detect Short circuit in signal wire to VB					
C_V_FUP_EFP_MV_MIN_DIAG	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Threshold value V_FUP_EFP_MV to detect Short circuit in signal wire to ground or wire break					
C_ABC_INC_FUP_EFP	1	0...FFH	0...255	1	[-]
Increment debounce counter					
C_ABC_MAX_FUP_EFP	1	1...FFH	1...255	1	[-]
Maximum value debounce counter					

34.20 FUP_EFP sensor diagnosis (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_FUP_EFP	V/O	0...1H	0...1	1	[-]
Inhibition bit for the fuel pressure sensor diagnosis (low pressure circuit)					

Input data:

LV_IGK	LV_FUP_LIH_HOM_VCV_OPEN_REQ		

FUNCTION DESCRIPTION:

General Information:

Depending on certain conditions the fuel pressure sensor diagnosis for the low pressure system must be inhibited.

Application conditions:

Initialization (at reset and LV_IGK = 0--> 1): LV_INH_DIAG_FUP_EFP = 0

Recurrence: 40 ms

Activation: LV_IGK = 1

Deactivation: When the activation condition is not fulfilled

Formula section:

```
If     LV_FUP_LIH_HOM_VCV_OPEN_REQ = 1
then   LV_INH_DIAG_FUP_EFP = 1
else   LV_INH_DIAG_FUP_EFP = 0
endif
```

Configuration for diagnostic symptoms:

Diagnosis	Symptom	Nr	ABC type
Fuel pressure sensor (low pressure circuit) diagnosis with short circuit signal wire to VB or wire break	Short circuit in signal wire to VB or wire break (SCP_OC)	*SYM_0*	STD_INI
FUP_EFP			
Fuel pressure sensor (low pressure circuit) diagnosis short circuit in signal wire to ground	short circuit in signal wire to ground (SCG)	*SYM_1*	STD_INI
FUP_EFP			

34.21 Fuel pressure sensor diagnosis (FUP)

Output data:

Name	Mod.	Hex. Limit	Phys. Limit	Resol.	Unit
LV_ERR_FUP	V/O	0...1H	0...1	1	[-]
FUP sensor error detected					
LV_CDN_DIAG_FUP	V	0...1H	0...1	1	[-]
Status od diagnosis for FUP sensor					
ERR_SYM_FUP	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom FUP sensor					
LV_END_DIAG_FUP	V	0...1H	0...1	1	[-]
End of diagnosis for FUP sensor diagnosis					

Input data:

V_FUP_MV	LV_INH_DIAG_FUP	N_32	LV_IGK

Import actions:

ACTION_ERRM_FilterSymptom(IN< XX >, IN< LV_CDN_DIAG_XX >, IN< ERR_SYM_XX >,
IN< C_ABC_INC_XX >, IN< C_ABC_DEC_XX >, IN< C_ABC_MAX_XX >, OUT< LV_ERR_XX >)

FUNCTION DESCRIPTION:

General Information:

Analog input signal in the A/D-Input from the Microprocessor.

Description

Error-symtoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit in signal wire to VB or wire break   (= SYM_0)
 │ │ └───> Short circuit in signal wire to ground             (= SYM_1)
 │ └─────> -                                                  (= SYM_2)
 └───────> -                                                  (= SYM_3)
```

Remark: Calculation of LV_END_DIAG_FUP see generic calculation “End of diagnosis” in anti bounce algorithm.

Application conditions:

Initialization:

LV_CDN_DIAG_FUP = *Refer to filtering configuration for the initialisation value*

LV_END_DIAG_FUP = *Refer to filtering configuration for the initialisation value*
LV_ERR_FUP = *Refer to filtering configuration for the initialisation value*
ERR_SYM_FUP = *Refer to filtering configuration for the initialisation value*

Recurrence: 40 ms

Activation:

```
if      LV_IGK = 1 AND LV_ INH_DIAG_FUP = 0
        AND N_32 <= C_N_32_FUP_DIAG_MAX
Then    LV_CDN_DIAG_FUP = 1
        (diagnosis is active)
else    LV_CDN_DIAG_FUP = 0
        (diagnosis is inactive)
endif
```

Deactivation:

When the activation condition is not fulfilled then LV_CDN_DIAG_FUP = 0.

Formula section:

```
{ SYM_0 : Short circuit in signal wire to VB or wire break }
If    V_FUP_MV > C_V_FUP_MV_MAX_DIAG
then
      ERR_SYM_ FUP = SYM_0                    { Symptom SCP or OC is active }
else
      { SYM_1 : Short circuit in signal wire to ground }
      If      V_FUP_MV < C_V_FUP_MV_MIN_DIAG
      then
              ERR_SYM_ FUP = SYM_1            { Symptom SCG is active }
      else
              ERR_SYM_FUP = NO_SYM            { No failure has been detected }
      endif

endif
```

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

For failure and error management treatment the anti-bounce mechanism is called with the parameters LV_CDN_DIAG_FUP and ERR_SYM_FUP.

ACTION_ERRM_FilterSymptom (IN< FUP >, IN< LV_CDN_DIAG_FUP >,
IN< ERR_SYM_FUP >, IN< C_ABC_INC_FUP >, IN< 1 >,
IN< C_ABC_MAX_FUP >, OUT< LV_ERR_FUP >)

This algorithm determines LV_ERR_FUP and LV_END_DIAG_FUP and delivers the result to Error Management.

Calibration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
C_V_FUP_MV_MAX_DIAG	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Threshold value V_FUP_MV to detect Short circuit in signal wire to VB					
C_V_FUP_MV_MIN_DIAG	1	0...7FFFH	0...4.99984	0.1526e-3	[V]
Threshold value V_FUP_MV to detect Short circuit in signal wire to ground or wire break					
C_N_32_FUP_DIAG_MAX	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed for FUP diagnosis					
C_ABC_INC_FUP	1	0...FFH	0...255	1	[-]
Increment debounce counter					
C_ABC_MAX_FUP	1	1...FFH	1...255	1	[-]
Maximum value debounce counter					

34.22 Fuel pressure sensor diagnosis (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_FUP	V/O	0...1H	0...1	1	-
Inhibition bit for the fuel pressure sensor diagnosis					

Input data:

LV_IGK			

FUNCTION DESCRIPTION:

General Information:

Depending on certain conditions the fuel pressure sensor diagnosis must be inhibited.

Application conditions:

Initialization (at reset and LV_IGK = 0--> 1): LV_INH_DIAG_FUP = 0

Recurrence: 40 ms

Activation:

LV_IGK = 1

Deactivation:

When the activation condition is not fulfilled

Formula section:

LV_INH_DIAG_FUP = 0

Configuration for diagnostic symptoms:

Diagnostic FUP	Symptom description	Symptom	Filter type
Fuel pressure sensor diagnosis with short circuit signal wire to VB or to ground or wire break	*SCP_OC*	*SYM_0*	STD
	SCG	*SYM_1*	

34.23 Electrical diagnosis of fuel pressure actuator

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_VCV	V/O	0 ... 1H	0 ... 1	1	-
Present failure: failure after filtering of diagnosis VCV					
CDN_DIAG_VCV	V	0 ... 7H	0 ... 7	1	-
Diagnosis condition for each symptom of VCV bit 0: diagnosis condition for symptom SYM_0 bit 1: diagnosis condition for symptom SYM_1 bit 2: diagnosis condition for symptom SYM_2 bit 3: diagnosis condition for symptom SYM_3					
ERR_DIAG_VCV	V/O	0 ... 7H	0 ... 7	1	-
Raw value of error symptom for VCV (only parameter) bit 0: Set, if raw value of error symptom SCP (SYM_0) is set bit 1: Set, if raw value of error symptom SCG (SYM_1) is set bit 2: Set, if raw value of error symptom OC (SYM_2) is set (all combinations are possible)					

Note :

- The data ERR_SYM_VCV, LV_END_DIAG_VCV and LV_CDN_DIAG_VCV are not present in output data, because they are not used by other functions. Despite the data are not present in output these data are provided in error management and always visible.

Input data:

LV_IGK	LV_INH_DIAG_VCV	PWM_VCV	LV_CDN_VB_OBD1

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>, OUT<LV_ERR_XX>)
ACTION_INFR_GetElDiagVcv(OUT< Cdn_diag_vcv>, OUT< Err_diag_vcv >)

Note :

ACTION_INFR_GetElDia*g*Vcv() is defined in the IRS (Infrastructure requirement specification)

FUNCTION DESCRIPTION:

General information:

The Volume controlled valve (VCV) is driven by the ECU via an output driver. The failure detection is done by ECU Hardware.
The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements.

Description:

After activation conditions are met, the diagnosis activation is delayed for NC_PSD_DLY_VCV executions, to avoid the usage of wrong infrastructure information.

Error-symptoms and conditions: are defined to this diagnosis function as following

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to battery (SCP)      (= SYM_0)
 │ │ └───> Short circuit to ground (SCG)       (= SYM_1)
 │ └─────> Open circuit (OC)                   (= SYM_2)
 └───────> -                                   (= SYM_3)
```

Application conditions:

Initialisation: at transition lv_igk 0->1:
LV_ERR_VCV = 0 and

ERR_SYM_VCV,

LV_END_DIAG_VCV

(according Filter-type)

Set delay counter for NC_PSD_DLY_VCV

at reset:

CDN_DIAG_VCV = 0,

LV_ERR_VCV = 0 and

ERR_SYM_VCV = 0,

LV_CDN_DIAG_VCV = 0,

LV_END_DIAG_VCV = 0,

(according Filter-type)

Set delay counter for NC_PSD_DLY_VCV

Recurrence: 100 ms

Activation: -

Deactivation: When the activation condition is not fulfilled

Formula section:

If LV_IGK = 1
and LV_CDN_VB_OBD1 = 1
and LV_INH_DIAG_VCV = 0

Then

Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_VCV and basic diagnosis conditions CDN_DIAG_VCV) received from the infrastructure:

ACTION_INFR_GetElDiagVcv (OUT<Cdn_diag_vcv>, OUT<Err_diag_vcv>)

Basic diagnosis conditions are set according infrastructure information: CDN_DIAG_VCV
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_VCV

If Activation conditions are met for the NC_PSD_DLY_VCV recurrence of

```
                    the diagnosis
            Then    { Additional diagnosis conditions }
                    If      C_PWM_VCV_MIN_DIAG_SCP > PWM_VCV
                    then  bit 0 of CDN_DIAG_VCV  = 0 {Diagnosis of SCP not possible }
                    Endif
                    If      PWM_VCV > C_PWM_VCV_MAX_DIAG_SCG
                    then  bit 1 of CDN_DIAG_VCV  = 0 { Diagnosis of SCG not possible }
                    Endif
                    If      PWM_VCV < C_PWM_VCV_MIN_DIAG_OC
                    Or      PWM_VCV > C_PWM_VCV_MAX_DIAG_SCG
                    then  bit 2 of CDN_DIAG_VCV  = 0 { Diagnosis of OC not possible }
                    Endif
            Else    CDN_DIAG_VCV = 0
            Endif
Else
            CDN_DIAG_VCV = 0
Endif
```

Failure filtering and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_VCV and ERR_DIAG_VCV.

ACTION_ERRM_FilterMulticondition (IN<*VCV*>, IN<CDN_DIAG_VCV>, IN<ERR_DIAG_VCV>, IN<C_ABC_INC_VCV>, IN<C_ABC_MAX_VCV>, OUT<LV_ERR_VCV>)
This algorithm determines:

ERR_SYM_VCV	(detected error symptom for VCV diagnosis)
LV_ERR_VCV	(Error flag for debounced error of VCV)
LV_CDN_DIAG_VCV	(Diagnosis condition information)
LV_END_DIAG_VCV	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_PWM_VCV_MIN_DIAG_SCP	1	0..FFH	0..99.609375	0.391	%
Minimum threshold for SCP diagnosis window					
C_PWM_VCV_MAX_DIAG_SCG	1	0..FFH	0..99.609375	0.391	%
Maximum threshold for SCG diagnosis window					
C_PWM_VCV_MIN_DIAG_OC	1	0..FFH	0..99.609375	0.391	%
Minimum threshold for OC diagnosis window					
C_ABC_INC_VCV	1	0..FFH	0..255	1	
Antibounce counter increment					
C_ABC_MAX_VCV	1	1..FFH	1..255	1	
Maximum value for antibounce counter					

Configuration for diagnostic symptoms :

Diagnostic VCV	Symptom description	Symptom	Filter type
Diagnostic	*SCP*	*SYM_0*	*MPL_STD_INI*

description	*SCG*	*SYM_1*	
	OC	*SYM_2*	

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_VCV	1	1 ... FH	1 ... 15	1	-
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (= 2)					

34.24 Electrical diagnosis of fuel pressure actuator (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_VCV	V/O	0...1H	0...1	1	-
Inhibition bit for the fuel pressure actuator diagnosis					

Input data:

FUNCTION DESCRIPTION:

General information:

Depending on certain conditions the fuel pressure actuator diagnosis must be inhibited.

Application conditions:

Initialisation (at reset and ignition key off to on): LV_INH_DIAG_VCV = 0

Recurrence: 100 ms

Formua section:

LV_INH_DIAG_VCV = 0

34.25 Fuel system pressure diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_FUP_ST	O/V	0...1H	0...1	1	-
Error flag for fuel pressure at start too low					
LV_ERR_FUP_MFP_PLAUS	O/V	0...1H	0...1	1	-
Boolean that indicates inconsistencies between fuel pressure and mass fuel pump					
LV_ERR_H_PRS_SYS	O/V	0...1H	0...1	1	-
Boolean that indicates abnormal fuel pressure value in fuel system					
LV_ERR_FUP_ORNG	O/V	0...1H	0...1	1	-
Boolean that indicates fuel pressure out of range					
LV_ERR_VCV_PLAUS	O/V	0...1H	0...1	1	-
Boolean that indicates abnormal value of the VCV adaptation					
LV_ERR_FUP_STOP	O/V	0...1H	0...1	1	-
Boolean that indicates to high fuel pressure after engine stop					
FAC_LAM_DIF_MFP_PLAUS	V	8000...7FFFH	-50...49.9984741	0.00152588	%
Deviation of fuel mass controller for MFP_PLAUS					
LV_CDN_DIAG_FUP_ORNG	V	0...1H	0...1	1	-
Status of diagnosis flag for FUP_ORNG check					
LV_CDN_DIAG_FUP_ST	V	0...1H	0...1	1	-
Status of diagnosis flag for FUPstart					
ERR_SYM_FUP_MFP_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom FUP/MFP plausibility check					
LV_END_DIAG_FUP_MFP_PLAUS	V	0...1H	0...1	1	-
End of diagnosis flag for FUP/MFP plausibility check					
LV_END_DIAG_FUP_ST	V	0...1H	0...1	1	-
End of diagnosis flag for FUP too low at engine start					
ERR_SYM_FUP_ORNG	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom FUP_ORNG check					
LV_END_DIAG_FUP_ORNG	V	0...1H	0...1	1	-
End of diagnosis flag for FUP sensor ORNG check					
LV_CDN_DIAG_FUP_STOP	V	0...1H	0...1	1	-
Status of diagnosis flag for FUP check after engine stop					
LV_END_DIAG_H_PRS_SYS	V	0...1H	0...1	1	-
End of diagnosis flag for high pressure system monitoring					
LV_END_DIAG_FUP_STOP	V	0...1H	0...1	1	-
End of diagnosis flag for FUP check after engine stop					
ERR_SYM_FUP_STOP	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom FUP check after engine stop					
LV_CDN_DIAG_VCV_PLAUS	V	0...1H	0...1	1	-
Status of diagnosis flag for VCV adaption plausibility check					
LV_END_DIAG_VCV_PLAUS	V	0...1H	0...1	1	-
End of diagnosis flag for dadption plausibility check					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_VCV_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom for adaptation plausibility check					
T_FUP_STOP	V	0...FFFFH	0...1.3107E+3	0.02	s
Time for detection of to high fuel pressure in the power latch					
LV_CDN_DIAG_FUP_MFP_PLAUS_1	V	0...1H	0...1	1	-
Status of diagnosis flag for FUP/MFP plausibility check SYM 1					
LV_CDN_DIAG_H_PRS_SYS_2	V	0...1H	0...1	1	-
Status of diagnosis flag for high pressure system monitoring SYM 2					
VFF_VCV_DIAG_CLC	V	0...FFFFH	0...255	0.00389105	l/h
Calculated VCV flow for diagnosis					
LV_CDN_DIAG_H_PRS_SYS	V	0...1H	0...1	1	-
Status of diagnosis flag for high pressure system monitoring					
LV_CDN_DIAG_FUP_MFP_PLAUS	V	0...1H	0...1	1	-
Status of diagnosis flag for FUP/MFP plausibility check					
ERR_SYM_H_PRS_SYS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom for high pressure system monitoring					
ERR_SYM_FUP_ST	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected symptom FUP start					

Input data:

LV_IGK	N_32	LV_ST_END	LV_ST_H_PRS
LV_DC	LV_INH_FUP_ORNG	FUP	LV_INH_FUP_MFP_PLAUS
STATE_FUP_CTL	LV_INH_H_PRS_SYS	MFF_SP_FUP_CTL	LV_CAT_PURGE_ACT[NC_CBK_EX_NR]
LV_LAM_STOP_AE	LV_WUP	LV_FL	LV_LAM_LSCL[NC_CBK_EX_NR]
LV_LAMB_OHP[NC_CBK_EX_NR]	LV_CH	FAC_MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	FAC_LAM_AD_OUT[NC_CBK_EX_NR]
FAC_LAM_LIM[NC_CBK_EX_NR]	NC_CBK_EX_NR	ERR_SYM_FSD[NC_CBK_EX_NR]	LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]
ERR_SYM_FSD_LAM_LIM[NC_CBK_EX_NR]	LV_ERR_FSD_H_RNG[NC_CBK_EX_NR]	ERR_SYM_FSD_H_RNG[NC_CBK_EX_NR]	LV_TI_1_HOM_MIN
LV_TI_2_HOM_MIN	FUEL_MASS_REQ_CTL	LV_ERR_FSD[NC_CBK_EX_NR]	FUP_DIF
LV_ST_INJ_AUTH	FAC_CUR_VCV_BAS_AD_VAR_2	CUR_VCV_BAS_AD_VAR_2_ADD	LV_INH_FUP_ST_BOL
NC_IDX_DIAG_FUP_MFP_PLAUS	NC_IDX_DIAG_FUP_ORNG	NC_IDX_DIAG_H_PRS_SYS	NC_IDX_DIAG_FUP_ST
TFU	LV_INH_VCV_PLAUS	LV_INH_FUP_STOP	STATE_PWM_VCV
CUR_VCV_MIN_AD	NC_IDX_DIAG_VCV_PLAUS	NC_IDX_DIAG_FUP_STOP	LV_PWL
VFF_MFF_SP_FUP_CTL	CUR_VCV_BAS	CUR_VCV_DIF_REL	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_FUP_MFP_PLAUS	1	0...FFH	0...255	1	-
Increment of the plausibility check of FUP/MFP anti-bounce counter					
C_ABC_INC_FUP_ORNG	1	0...FFH	0...255	1	-
Increment of the FUP ORNG anti-bounce counter					
C_ABC_INC_FUP_ST	1	0...FFH	0...255	1	-
Increment of FUP at start monitoring anti-bounce counter					
C_ABC_INC_FUP_STOP	1	0...FFH	0...255	1	-
Increment of the FUP_STOP check anti-bounce counter					
C_ABC_INC_H_PRS_SYS	1	0...FFH	0...255	1	-
Increment of high pressure system monitoring anti-bounce counter					
C_ABC_INC_VCV_PLAUS	1	0...FFH	0...255	1	-
Increment of the VCV adaption check anti-bounce counter					
C_ABC_MAX_FUP_MFP_PLAUS	1	1...FFH	1...255	1	-
Threshold to be reached, before plausibility check error of FUP/MFP					
C_ABC_MAX_FUP_ORNG	1	1...FFH	1...255	1	-
Threshold to be reached, before FUP ORNG error					
C_ABC_MAX_FUP_ST	1	1...FFH	1...255	1	-
Threshold to be reached, before permanently activating FUP too low at start monitoring error					
C_ABC_MAX_FUP_STOP	1	1...FFH	1...255	1	-
Threshold to be reached, before FUP_STOP check error					
C_ABC_MAX_H_PRS_SYS	1	1...FFH	1...255	1	-
Threshold to be reached, before permanently activating high pressure system monitoring error					
C_ABC_MAX_VCV_PLAUS	1	1...FFH	1...255	1	-
Threshold to be reached, before VCV adaption check error					
C_CUR_VCV_AD_MFP_DIAG_MAX	1	8000...7FFFH	-32.768...32.767	0.001	A
Maximum adapted additive curent value					
C_CUR_VCV_AD_MFP_DIAG_MAX_1	1	8000...7FFFH	-32.768...32.767	0.001	A
Maximum adapted additive curent value SYM 1					
C_CUR_VCV_AD_MFP_DIAG_MAX_2	1	8000...7FFFH	-32.768...32.767	0.001	A
Maximum adapted additive curent value SYM 2					
C_CUR_VCV_AD_MFP_DIAG_MIN	1	8000...7FFFH	-32.768...32.767	0.001	A
Minimum adapted additive curent value					
C_CUR_VCV_BAS_AD_VCV_PLAUS_MAX	1	8000...7FFFH	-32.768...32.767	0.001	A
Maximum adapted additive curent value					
C_CUR_VCV_BAS_AD_VCV_PLAUS_MIN	1	8000...7FFFH	-32.768...32.767	0.001	A
Minimum adapted additive curent value					
C_CUR_VCV_DIF_REL_DIAG_MAX	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
Maximum relative current deviation to enable FUP_MFP_PLAUS and H_PRS_SYS diagnosis					
C_CUR_VCV_MIN_AD_VCV_PLAUS_MAX	1	8000...7FFFH	-32.768...32.767	0.001	A
Maximum adapted minimum curent value					
C_CUR_VCV_MIN_AD_VCV_PLAUS_MIN	1	8000...7FFFH	-32.768...32.767	0.001	A
Minimum adapted minimum curent value					
C_FAC_CUR_VCV_AD_MFP_DIAG_MAX	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
Maximum adapted factor value					
C_FAC_CUR_VCV_AD_MFP_DIAG_MAX_1	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
Maximum adapted factor value SYM 1					
C_FAC_CUR_VCV_AD_MFP_DIAG_MAX_2	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
Maximum adapted factor value SYM 2					
C_FAC_CUR_VCV_AD_MFP_DIAG_MIN	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
Minimum adapted factor value					
C_FAC_CUR_VCV_PLAUS_MAX	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
Maximum adapted factor value					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_CUR_VCV_PLAUS_MIN	1	8000...7FFFH	-2...1.99993896	6.10352E-5	-
Minimum adapted factor value					
C_FAC_LAM_AD_MFP_PLAUS_0	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Limit of FAC Lambda contr. range for MFP_PLAUS SYM_0					
C_FAC_LAM_AD_MFP_PLAUS_1	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Limit of FAC Lambda contr. range for MFP_PLAUS SYM_1					
C_FAC_LAM_H_PRS_SYS_MAX_0	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Limit of Lambda contr. range for H_PRS_SYS SYM_0					
C_FAC_LAM_H_PRS_SYS_MAX_1	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Limit of Lambda contr. range for H_PRS_SYS SYM_1					
C_FAC_LAM_H_PRS_SYS_MAX_2	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Limit of Lambda contr. range for H_PRS_SYS SYM_2					
C_FAC_LAM_H_PRS_SYS_MIN_0	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Limit of Lambda contr. range for H_PRS_SYS SYM_0					
C_FAC_LAM_H_PRS_SYS_MIN_1	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Limit of Lambda contr. range for H_PRS_SYS SYM_1					
C_FAC_LAM_H_PRS_SYS_MIN_2	1	8000...7FFFH	-50...49.9984741	0.00152588	%
Limit of Lambda contr. range for H_PRS_SYS SYM_2					
C_FUEL_REQ_CTL_MFP_DIAG_MAX	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg
Maximum threshold for highpressure controller					
C_FUEL_REQ_CTL_MFP_DIAG_MAX_1	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg
Maximum threshold for highpressure controller SYM1					
C_FUEL_REQ_CTL_MFP_DIAG_MAX_2	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg
Maximum threshold for highpressure controller SYM 2					
C_FUEL_REQ_CTL_MFP_DIAG_MIN	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg
Minimum threshold for highpressure controller					
C_FUEL_REQ_CTL_MFP_DIAG_MIN_0	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg
Minimum threshold for highpressure controller SYM0					
C_FUP_BOL_1	1	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Bottom fuel pressure threshold 1					
C_FUP_DIF_BOL	1	8000...7FFFH	-1.7389E+5 ... 1.73884E+5	5.30669108	hPa
Min FUP difference for H_PRS_SYS SYM 0					
C_FUP_DIF_TOL	1	8000...7FFFH	-1.7389E+5 ... 1.73884E+5	5.30669108	hPa
Top limit of FUP difference					
C_FUP_STOP	1	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Constant for the limit of fuel pressure after engine stop					
C_FUP_ST_BOL	1	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Fuel pressure at start of injection to detect too low start pressure					
C_FUP_TOL_1	1	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Top fuel pressure threshold 1					
C_FUP_TOL_2	1	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Top fuel pressure threshold 2					
C_MFF_SP_FUP_MFP_PLAUS_MAX	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Maximum Mass fuel flow for FUP_MFP_PLAUS diagnosis					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MFF_SP_FUP_MFP_PLAUS_MAX_1	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Maximum Mass fuel flow for FUP_MFP_PLAUS diagnosis SYM_1					
C_MFF_SP_FUP_MFP_PLAUS_MIN	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Minnimum Mass fuel flow for FUP_MFP_PLAUS diagnosis					
C_MFF_SP_FUP_MFP_PLAUS_MIN_1	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Minnimum Mass fuel flow for FUP_MFP_PLAUS diagnosis SYM_1					
C_MFF_SP_H_PRS_SYS_BOL_0	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Bottom limit of MFF for detection of stucked VCV					
C_MFF_SP_H_PRS_SYS_BOL_2	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Bottom limit of MFF for detection of stucked VCV					
C_MFF_SP_H_PRS_SYS_MAX_2	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Maximum Mass fuel flow for FUP_MFP_PLAUS diagnosis SYM_2					
C_MFF_SP_H_PRS_SYS_MIN_2	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Minnimum Mass fuel flow for FUP_MFP_PLAUS diagnosis SYM_2					
C_N_32_FUP_DIAG	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for FUP/FPAPWM plausibility check (Default value 800)					
C_N_32_FUP_MFP_PLAUS_MAX	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for the plausibility check					
C_N_32_FUP_MFP_PLAUS_MAX_1	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for the plausibility check SYM_1					
C_N_32_FUP_MFP_PLAUS_MIN	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for the plausibility check					
C_N_32_FUP_MFP_PLAUS_MIN_1	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for the plausibility check SYM_1					
C_N_32_H_PRS_SYS_MAX_2	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for the plausibility check SYM_2					
C_N_32_H_PRS_SYS_MIN_2	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for the plausibility check SYM_2					
C_N_32_VCV_PLAUS_MAX	1	0...FFH	0...8.16E+3	32	rpm
Maximum engine speed for the VCV adaption plausibility check					
C_N_32_VCV_PLAUS_MIN	1	0...FFH	0...8.16E+3	32	rpm
Minimum engine speed for VCV adapation plausibility check					
C_TFU_VCV_PLAUS_MAX	1	0...FEH	-48...142.5	0.75	°C
Maximum threshold for the VCV adapation plausibility check					
C_TFU_VCV_PLAUS_MIN	1	0...FEH	-48...142.5	0.75	°C
Minimum threshold for the VCV adaptatin plausibility check					
C_T_FUP_STOP	1	0...FFFFH	0...1.3107E+3	0.02	s
Time constant for detection of the fuel low pressure in the power latch					
C_T_MAX_H_PRS_SYS_0	1	0...FFFFH	0...655.35	0.01	s
Maximum delay time to debounce H_PRS_SYS SYM_0					
C_T_MAX_H_PRS_SYS_1	1	0...FFFFH	0...655.35	0.01	s
Maximum delay time to debounce H_PRS_SYS SYM_1					
C_T_MAX_H_PRS_SYS_2	1	0...FFFFH	0...655.35	0.01	s
Maximum delay time to debounce H_PRS_SYS SYM_2					
C_VFF_VCV_DIF_DIAG_0	1	0...FFFFH	-127.5...127.5	0.00389105	l/h
Threshold of VCV flow deviation to detect SYM 0					
C_VFF_VCV_DIF_DIAG_1	1	0...FFFFH	-127.5...127.5	0.00389105	l/h
Threshold of VCV flow deviation to detect SYM 1					
C_VFF_VCV_DIF_DIAG_2	1	0...FFFFH	-127.5...127.5	0.00389105	l/h
Threshold of VCV flow deviation to detect SYM 2					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VFF_VCV_DIF_DIAG_MAX	1	0...FFFFH	-127.5...127.5	0.00389105	l/h
Threshold of VCV flow deviation to detect FUP_MFP_PLAUS_SYM_1					
C_VFF_VCV_DIF_DIAG_MIN	1	0...FFFFH	-127.5...127.5	0.00389105	l/h
Threshold of VCV flow deviation to detect FUP_MFP_PLAUS_SYM_0					
LC_FUP_MFP_PLAUS_VAR_2_ACT	1	0...1H	0...1	1	-
Switch to enable the second variant of MFP_PLAUS and H_PRS_SYS diagnosis					
IP_FLOW_CUR_BAS_VCV	16	0...FFFFH	0...255	0.00389105	l/h
LDP_CUR_VCV_IP_FLOW_CUR_BAS	16	0...FFFFH	0...65.535	0.001	A
Map to convert CUR_VCV_BAS into VFF_VCV_DIAG_CLC					

Import actions:

ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure
ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter

34.25.1 FUNCTION DESCRIPTION:

The purpose is to diagnose mechanical errors.

Specific diagnosis information to the project are defined in chapters ‘Diagnosis information’ (Freeze frame, DTC, error code number, symptom number, data for MIL management ...).

General information:

After debounce the corresponding error bit LV_ERR... is set to 1 and the corresponding symptom is active. Only if the driving cycle is finished the error bit LV_ERR... and the debounced counter are set to 0 and the corresponding symptom is not active.

The fuel system pressure diagnosis will be executed in dependency on the status of the fuel pressure sensor.

If the diagnosis is not prohibited the MFP plausibility check, the FUP sensor plausibility check and the HP system monitoring can be executed. The MFP plausibility check will be calculated only in state MFP_CTL.

The high-pressure system check indicates mainly an abnormal fuel pressure value in fuel system.

If a plausibility check error or the high-pressure system check error is detected a request for Lambda=1 mode is set to identify the origin of the non-plausibility.

The flags LV_CDN_DIAG_XXX exept the flag for FUP_STOP are ==0 if the function is not active. LV_CDN_DIAG_FUP_STOP == 1 if the function is not active.

Application Condition

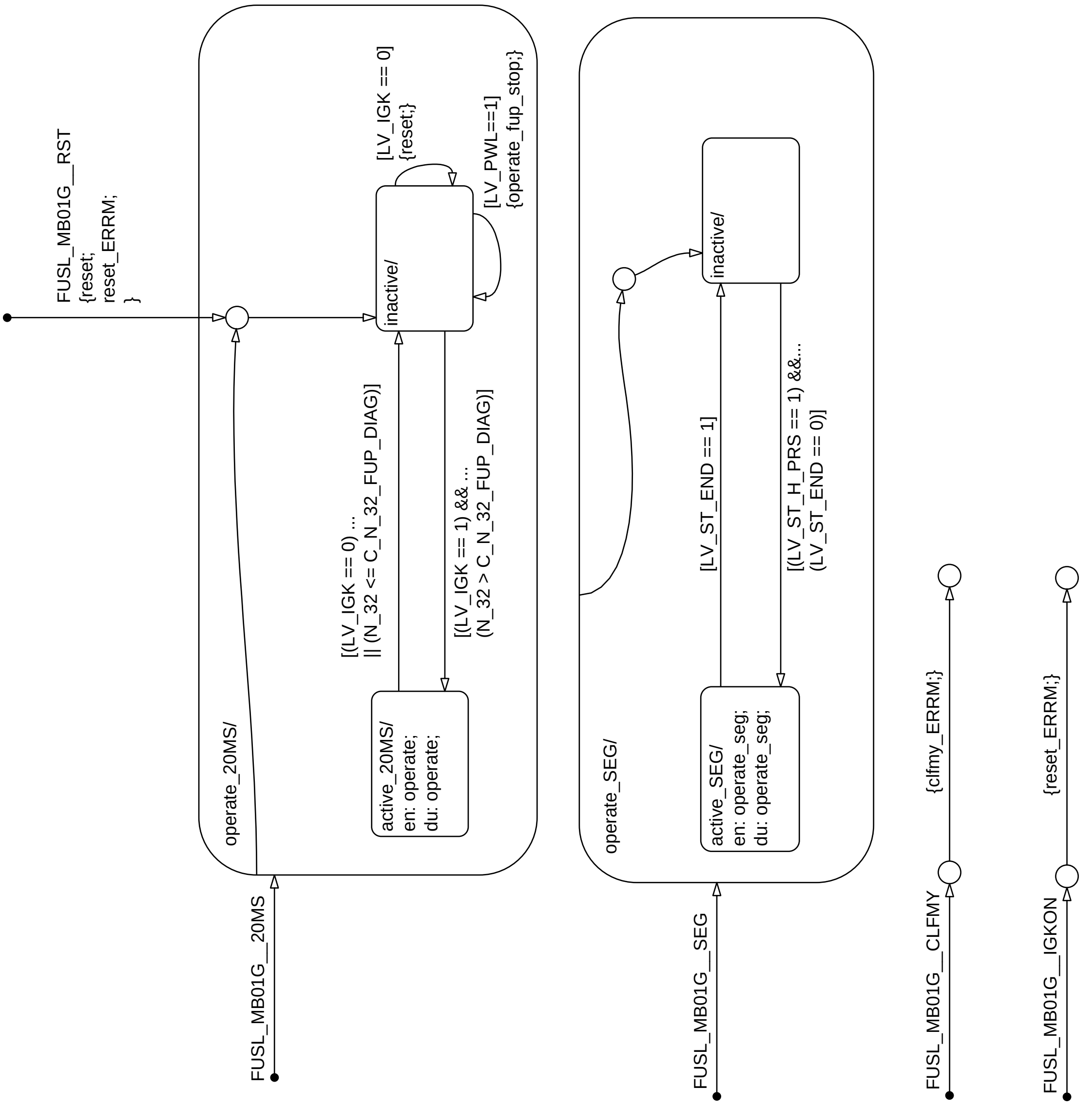
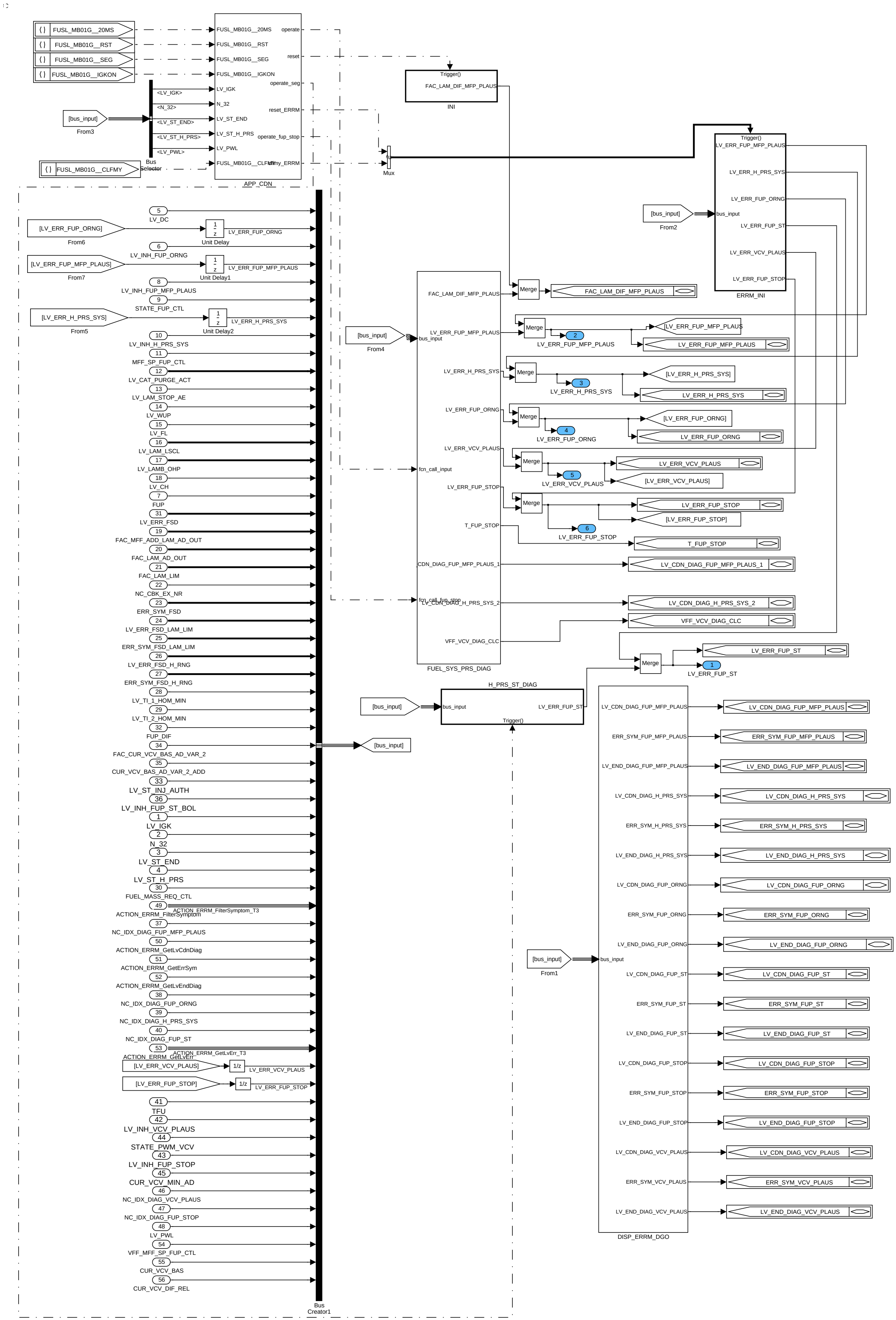


Figure 243 FUSL_MB01G/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 14−Nov−2006

Chapter	Baseline	Include File
Fuel supply	4DC3940S	43B01G01.00A

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Pages
5826 of 9643

Figure 244 FUSL_MB01G

34.25.1.1 SUBFUNCTION: DISP_ERRM_DGO

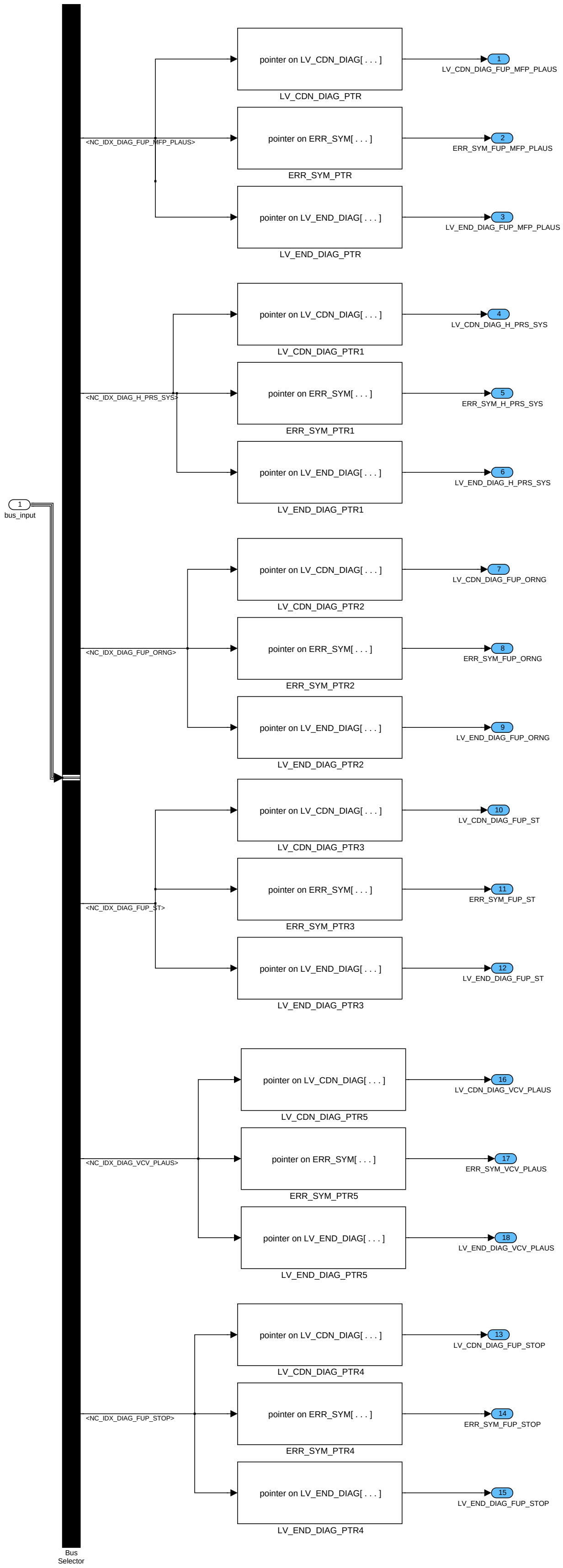


Figure 245 FUSL_MB01G/ DISP_ERRM_DGO

34.25.1.2 SUBFUNCTION: ERRM_INI

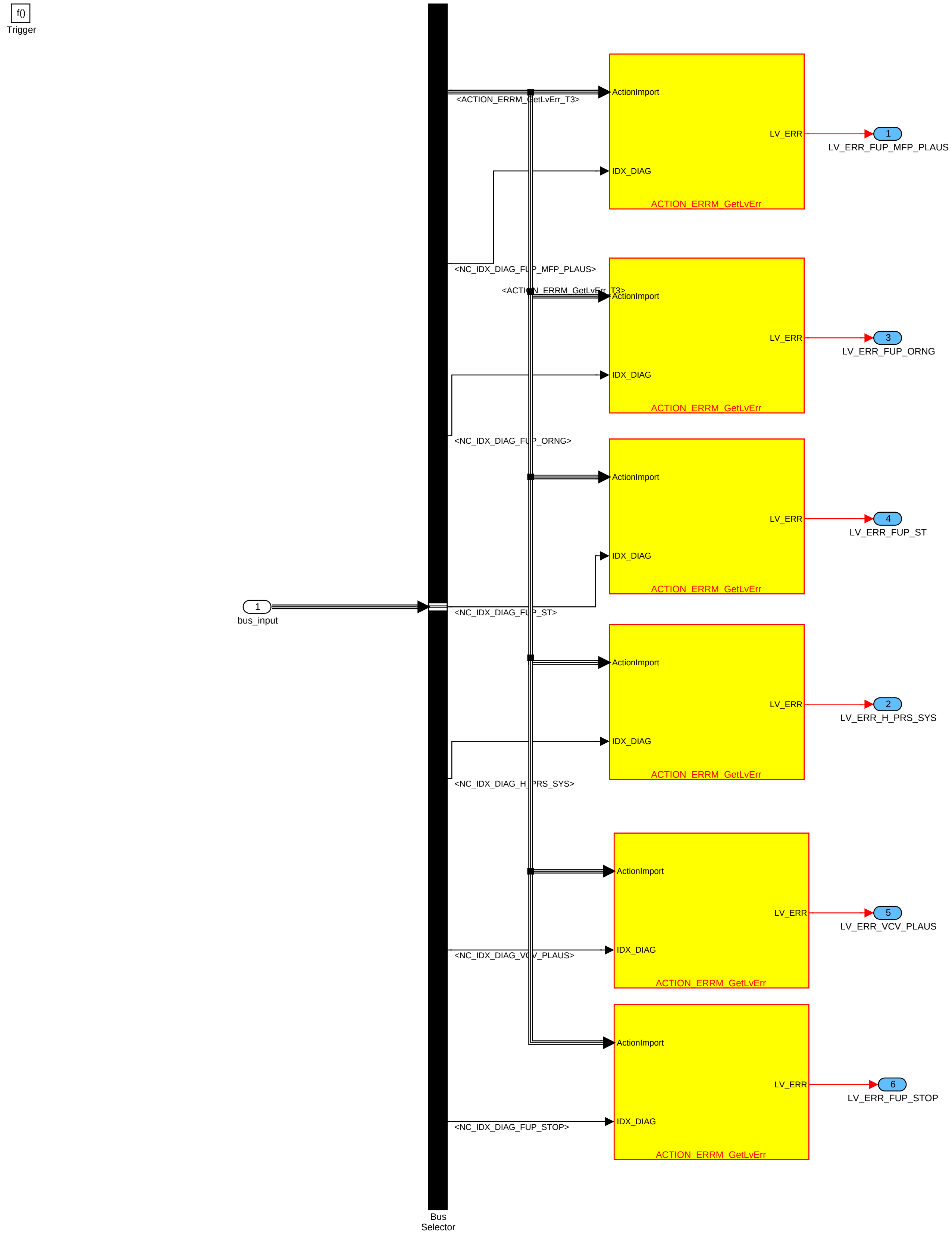


Figure 246 FUSL_MB01G/ ERRM_INI

34.25.1.3 FUSL_MB01G/FUEL_SYS_PRS_DIAG

General overview of the parts of the diagnosis function.

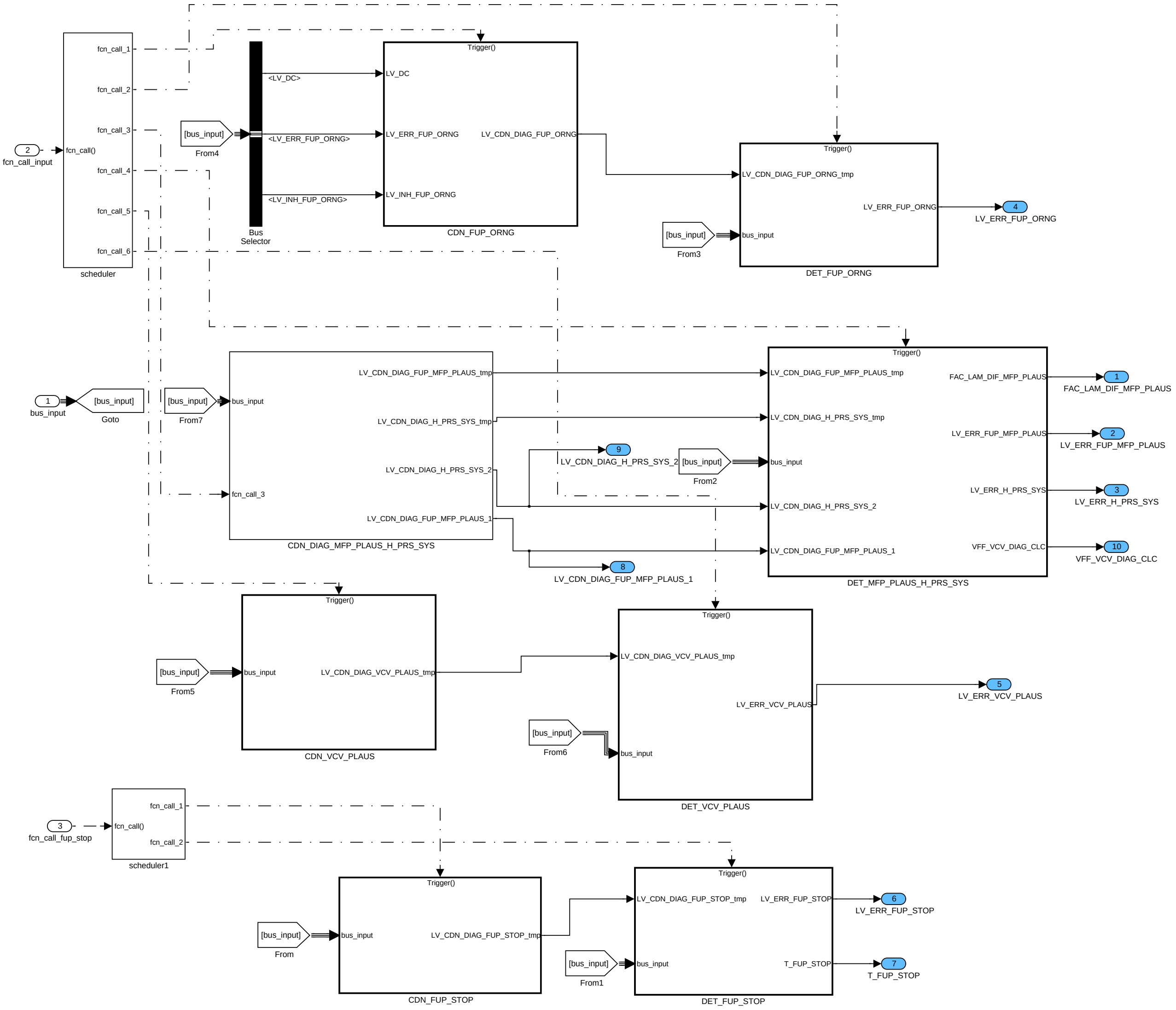


Figure 247 FUSL_MB01G/ FUEL_SYS_PRS_DIAG

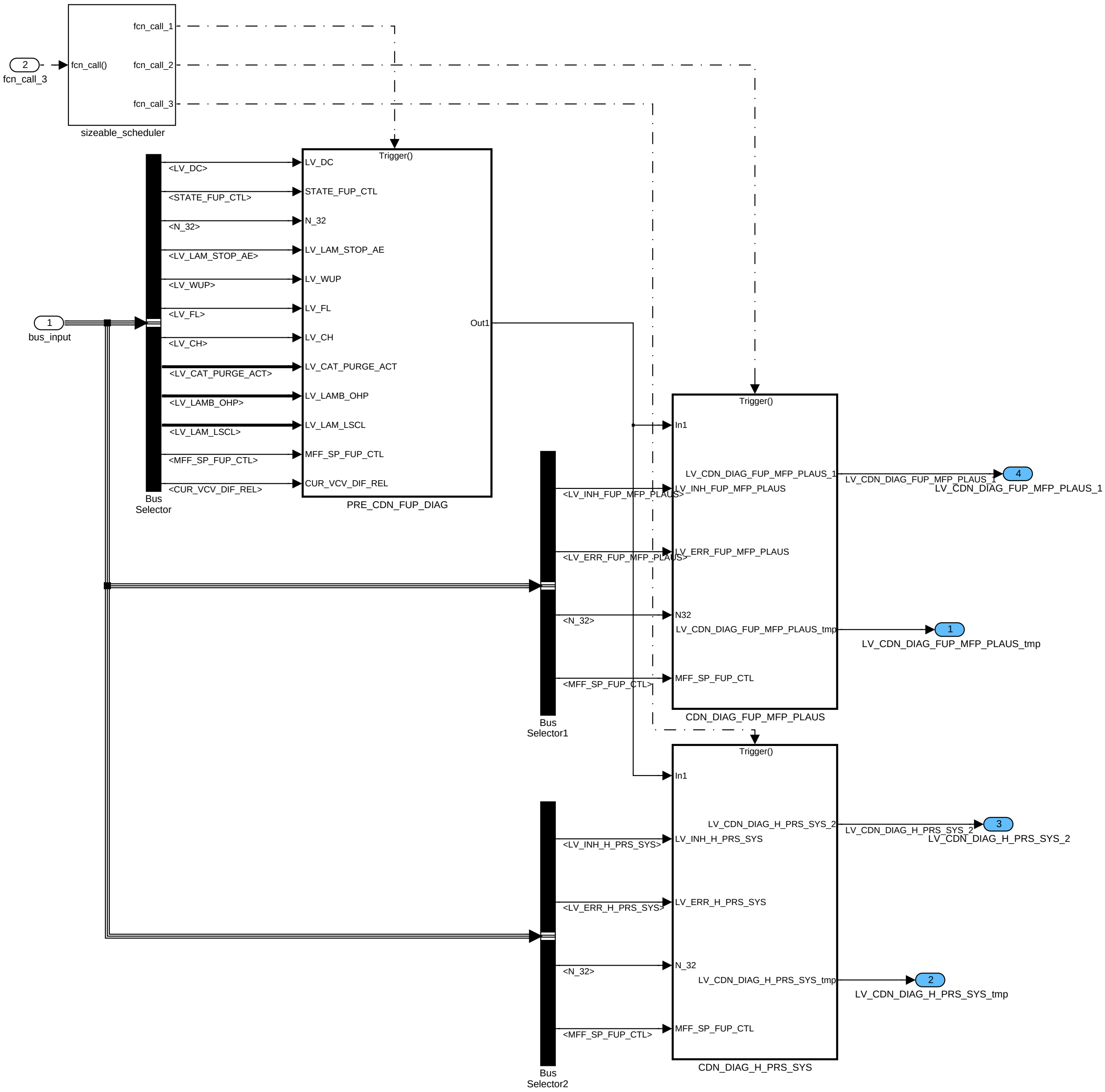


Figure 248 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ CDN_DIAG_MFP_PLAUS_H_PRS_SYS

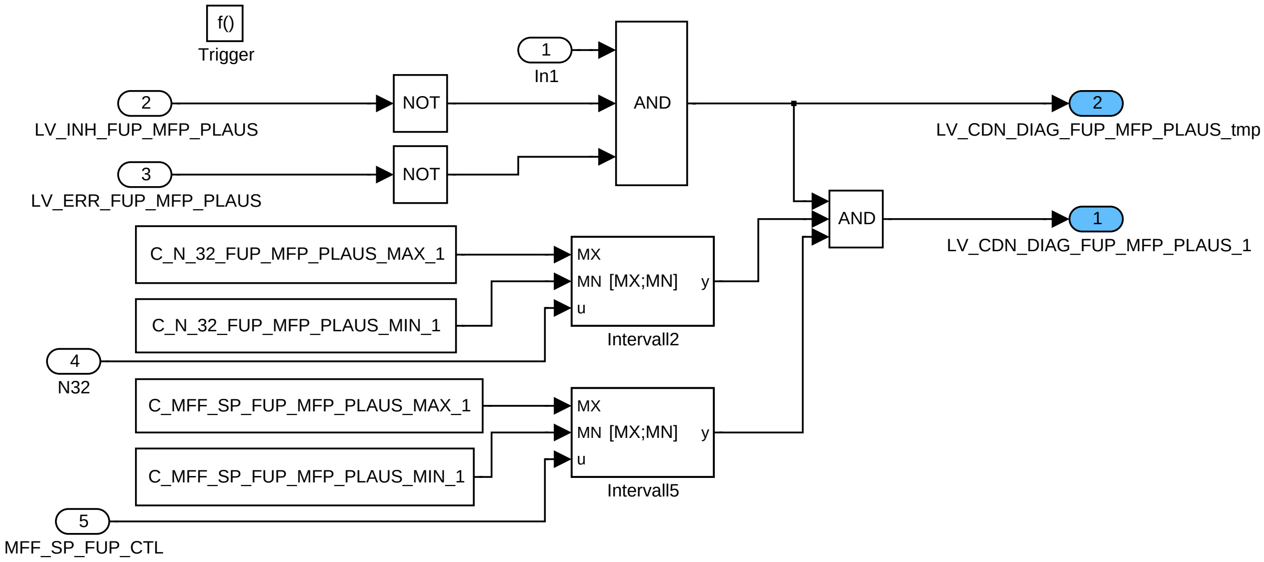


Figure 249 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ CDN_DIAG_MFP_PLAUS_H_PRS_SYS/ CDN_DIAG_FUP_MFP_PLAUS

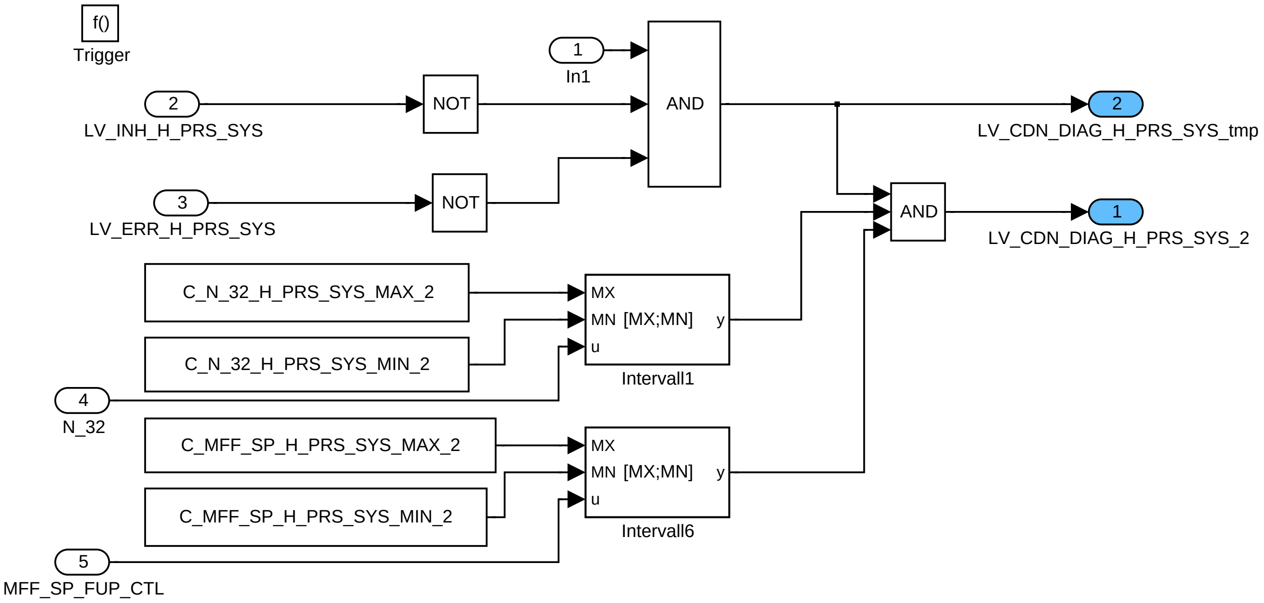


Figure 250 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ CDN_DIAG_MFP_PLAUS_H_PRS_SYS/ CDN_DIAG_H_PRS_SYS

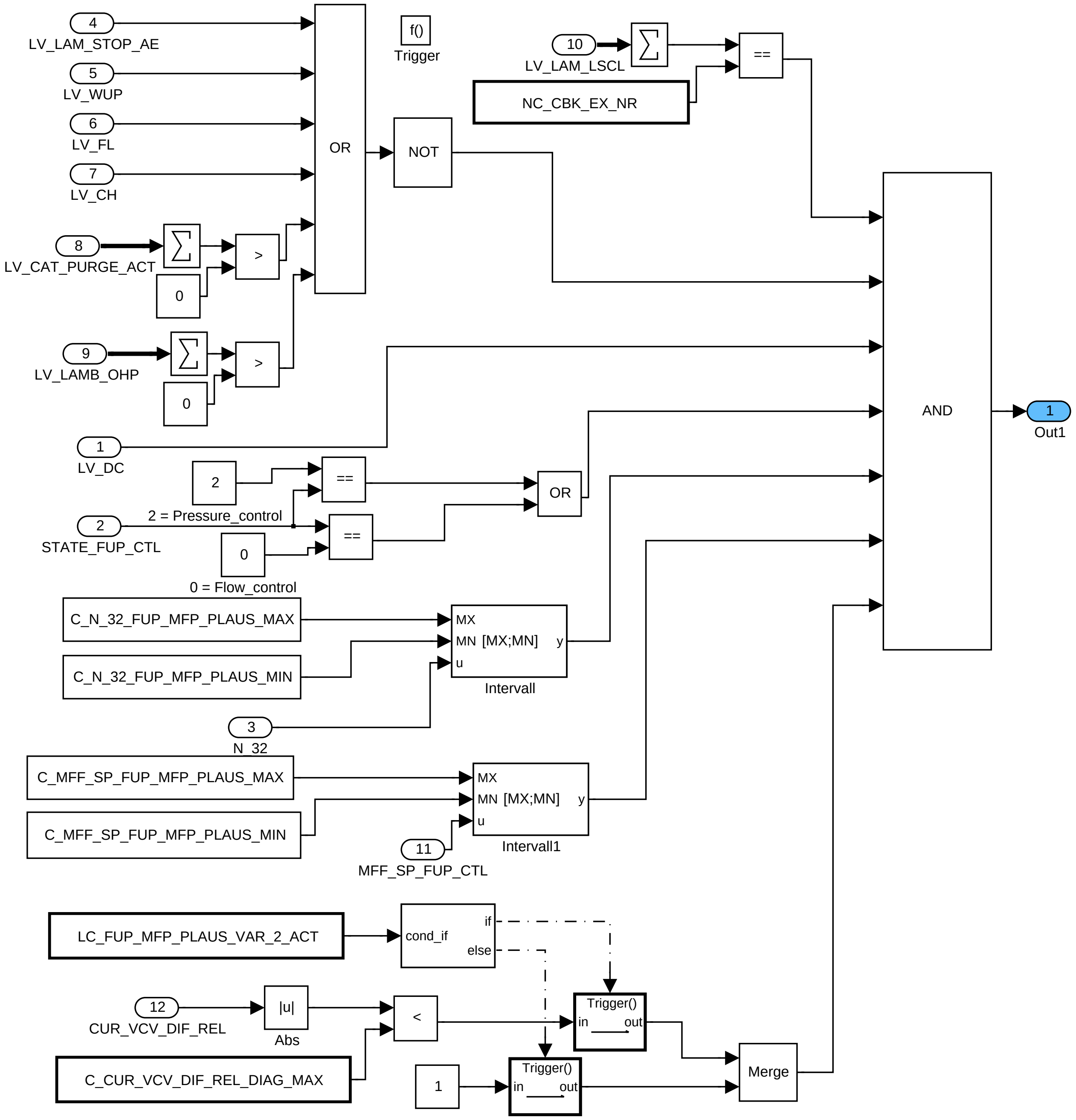


Figure 251 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ CDN_DIAG_MFP_PLAUS_H_PRS_SYS/ PRE_CDN_FUP_DIAG

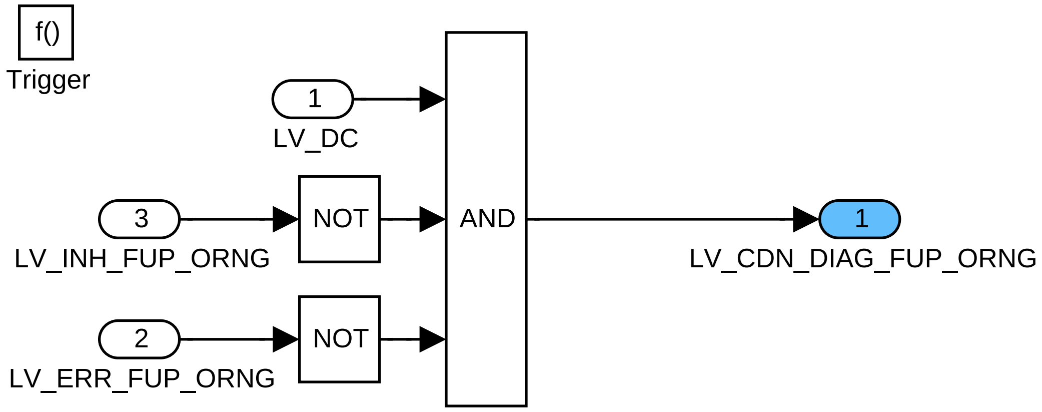


Figure 252 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ CDN_FUP_ORNG

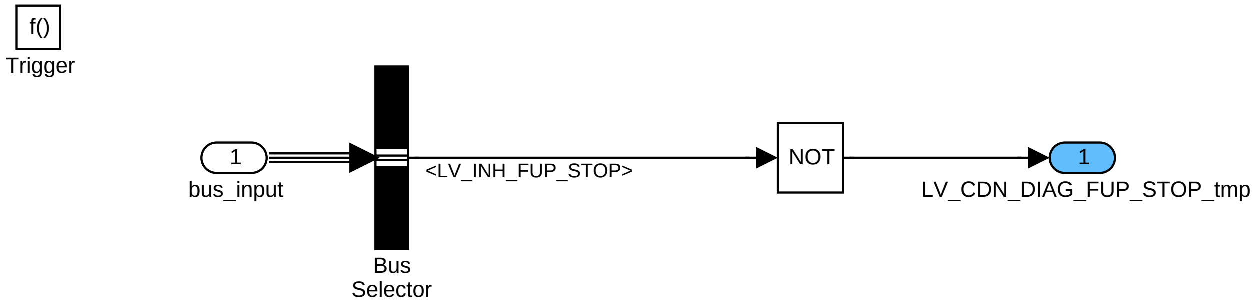


Figure 253 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ CDN_FUP_STOP

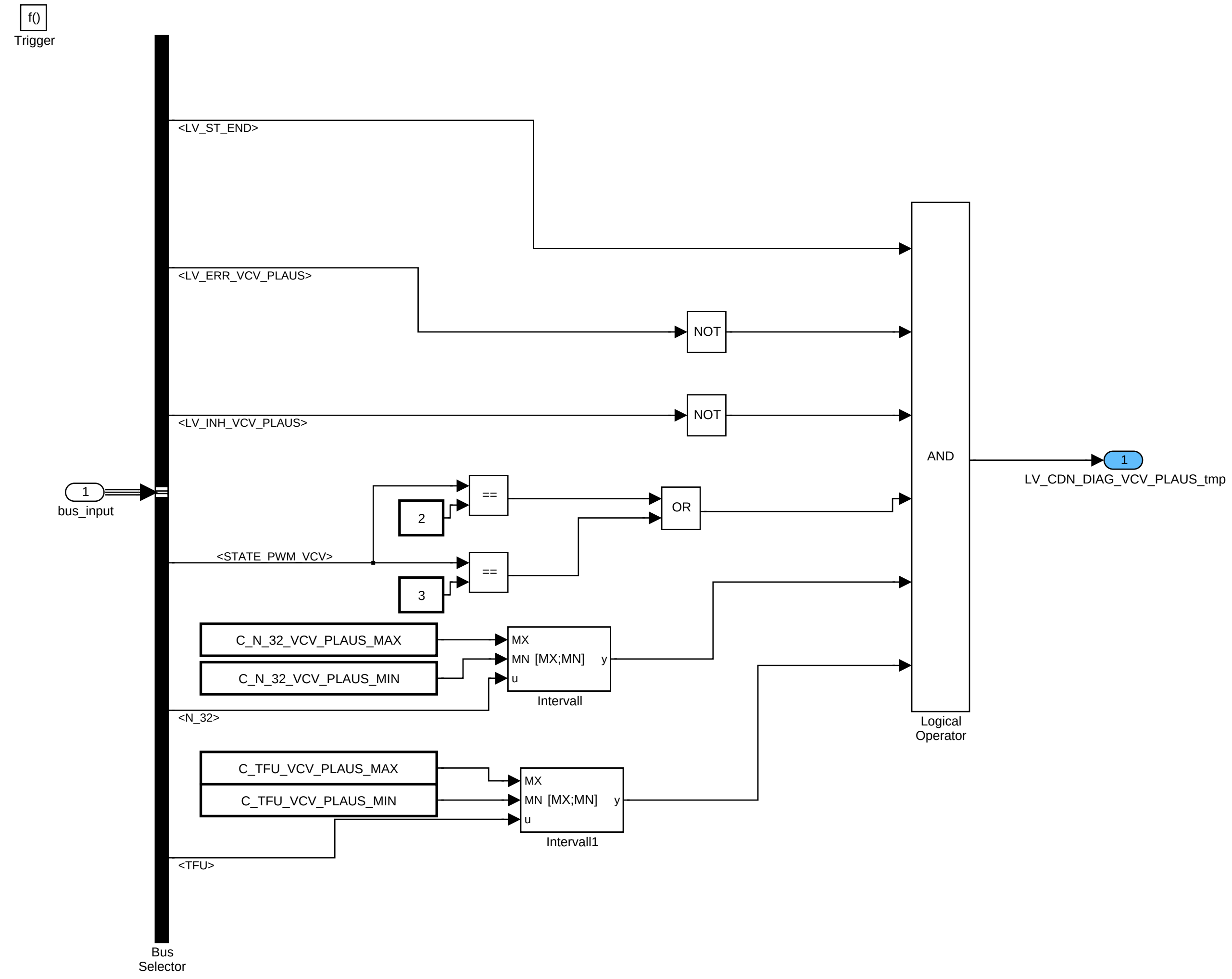


Figure 254 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ CDN_VCV_PLAUS

FUSL_MB01G/FUEL_SYS_PRS_DIAG/DET_FUP_ORNG

Detection of fuel pressure out of range.

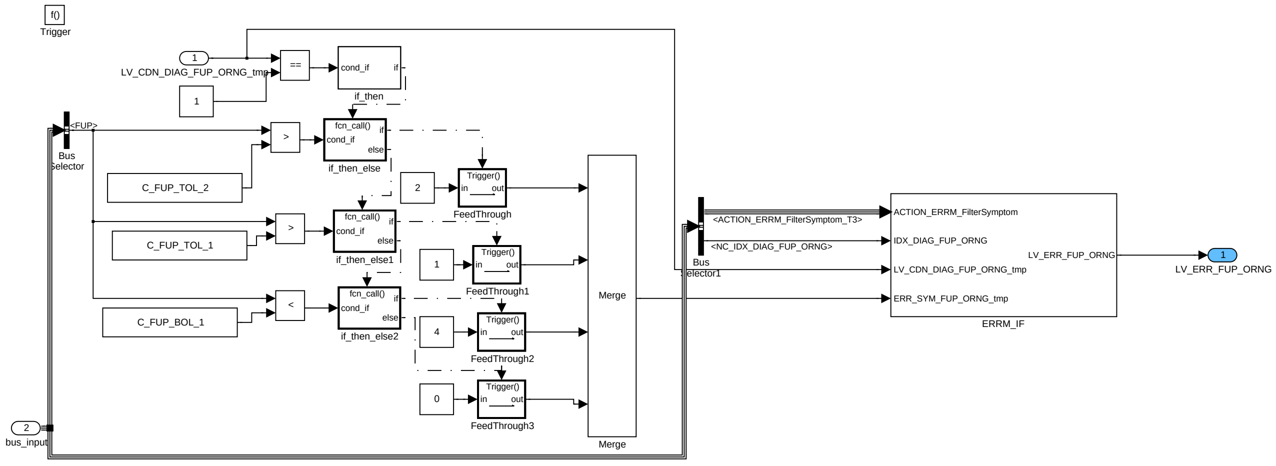


Figure 255 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_FUP_ORNG

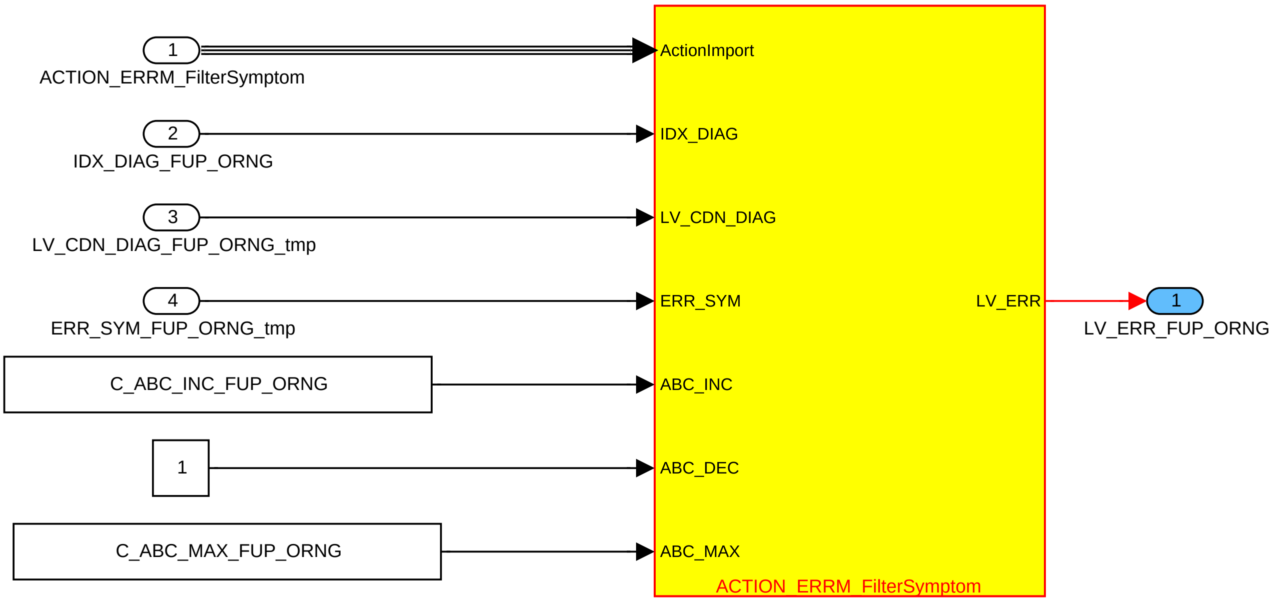


Figure 256 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_FUP_ORNG/ ERRM_IF

FUSL_MB01G/FUEL_SYS_PRS_DIAG/DET_FUP_STOP

Detection of fuel pressure at engine stop to high.

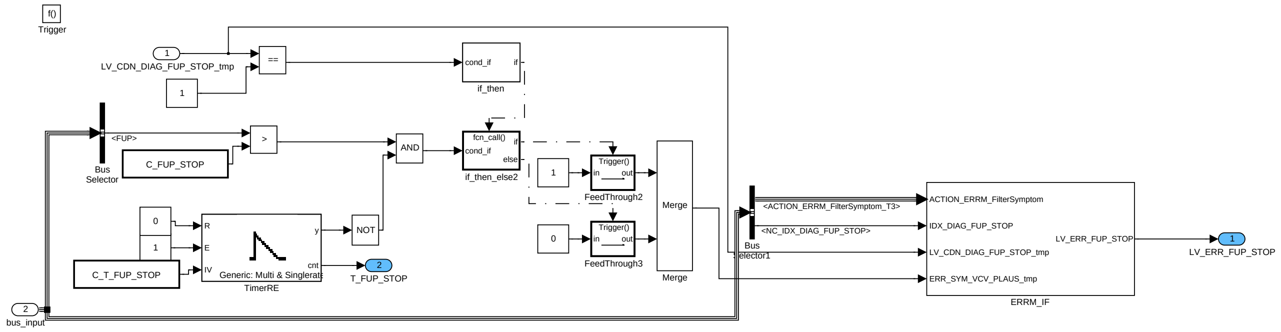


Figure 257 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_FUP_STOP

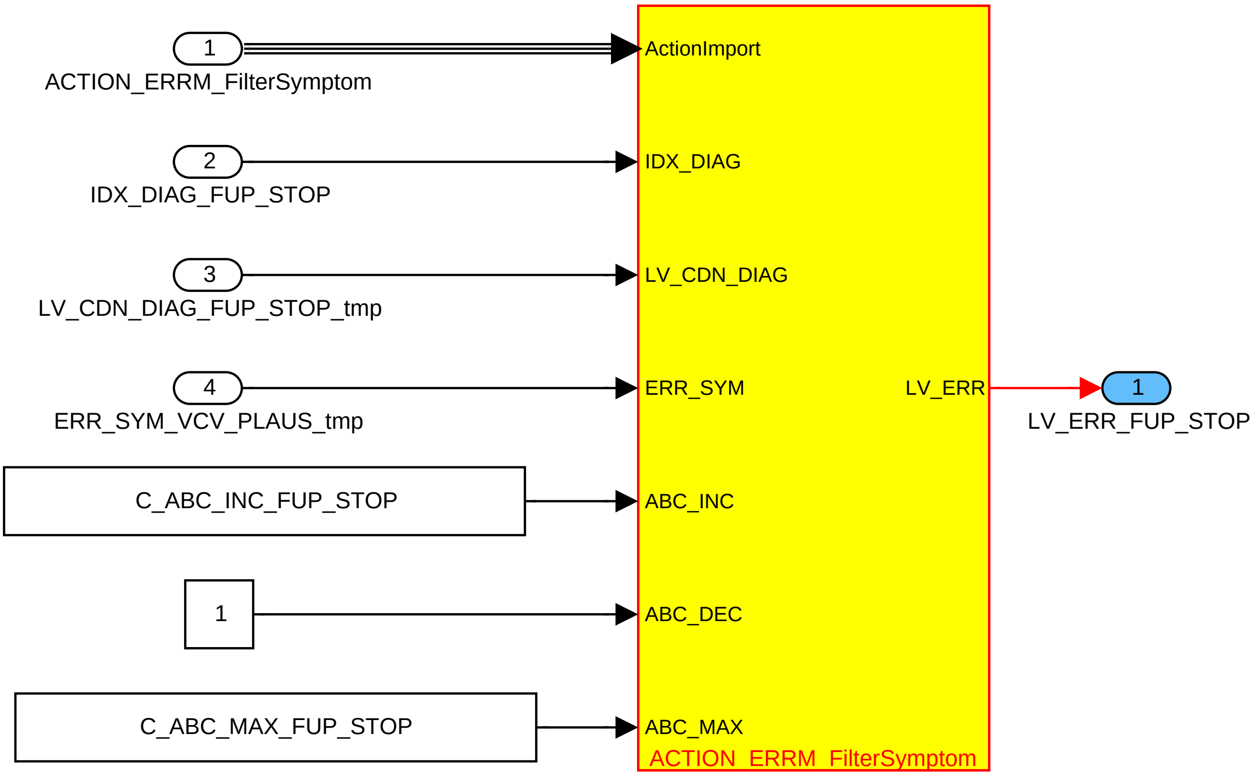


Figure 258 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_FUP_STOP/ ERRM_IF

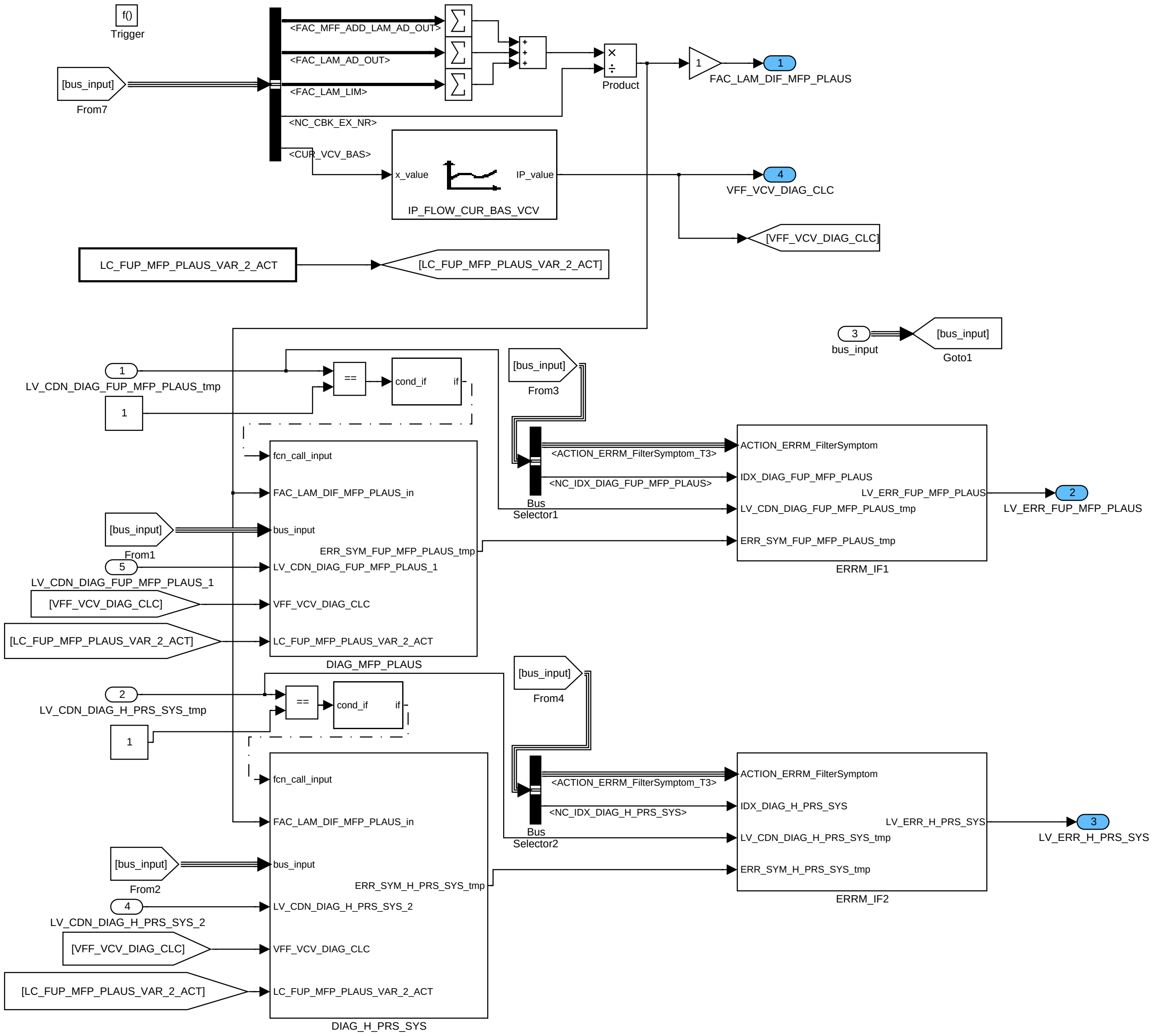


Figure 259 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS

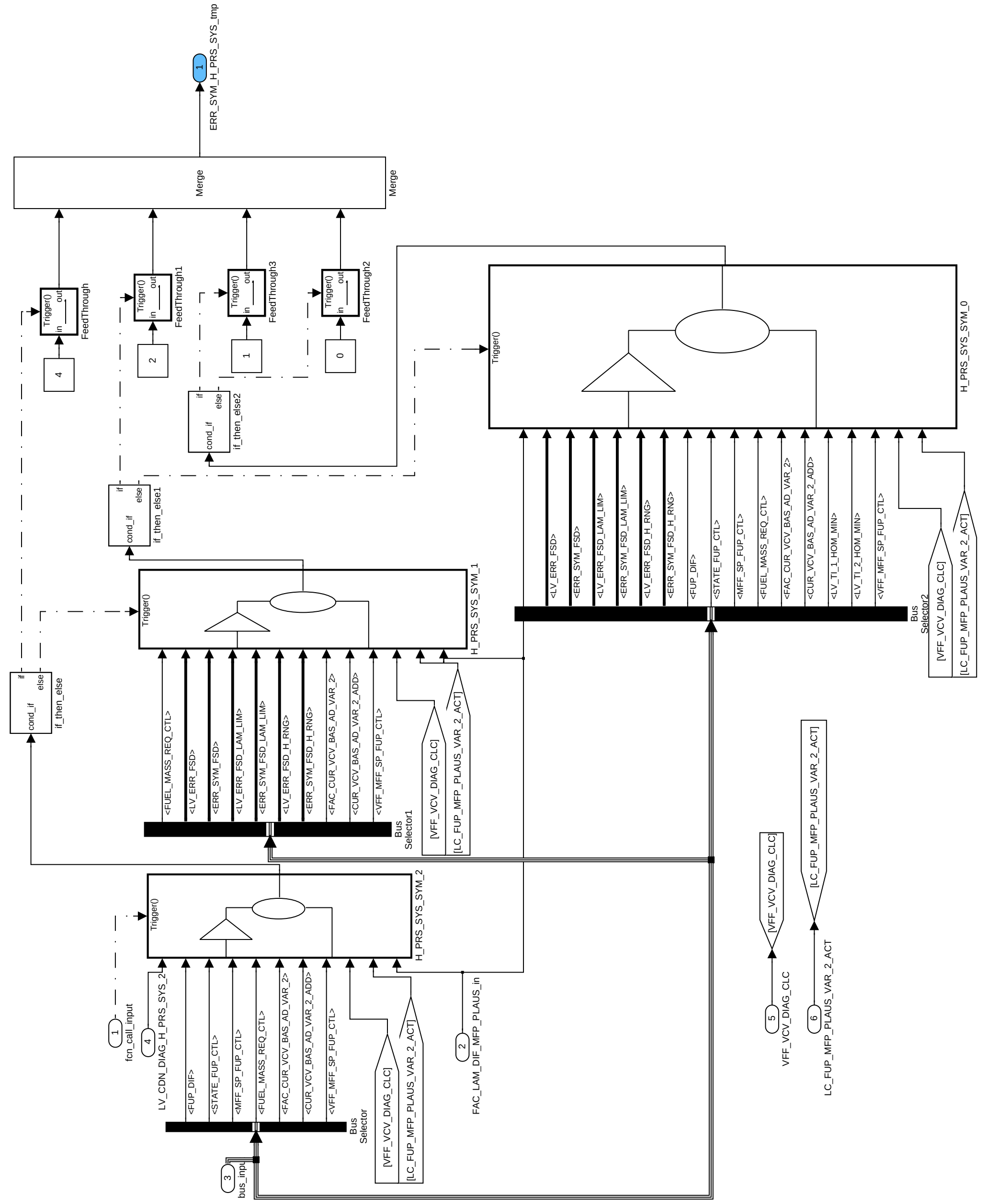


Figure 260 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ DIAG_H_PRS_SYS

FUSL_MB01G/FUEL_SYS_PRS_DIAG/DET_MFP_PLAUS_H_PRS_SYS/DIAG_H_PRS_SYS/H_PRS_SYS_SYM_0

Detection of H_PRS_SYS SYM_0, delivery of pump is too high.

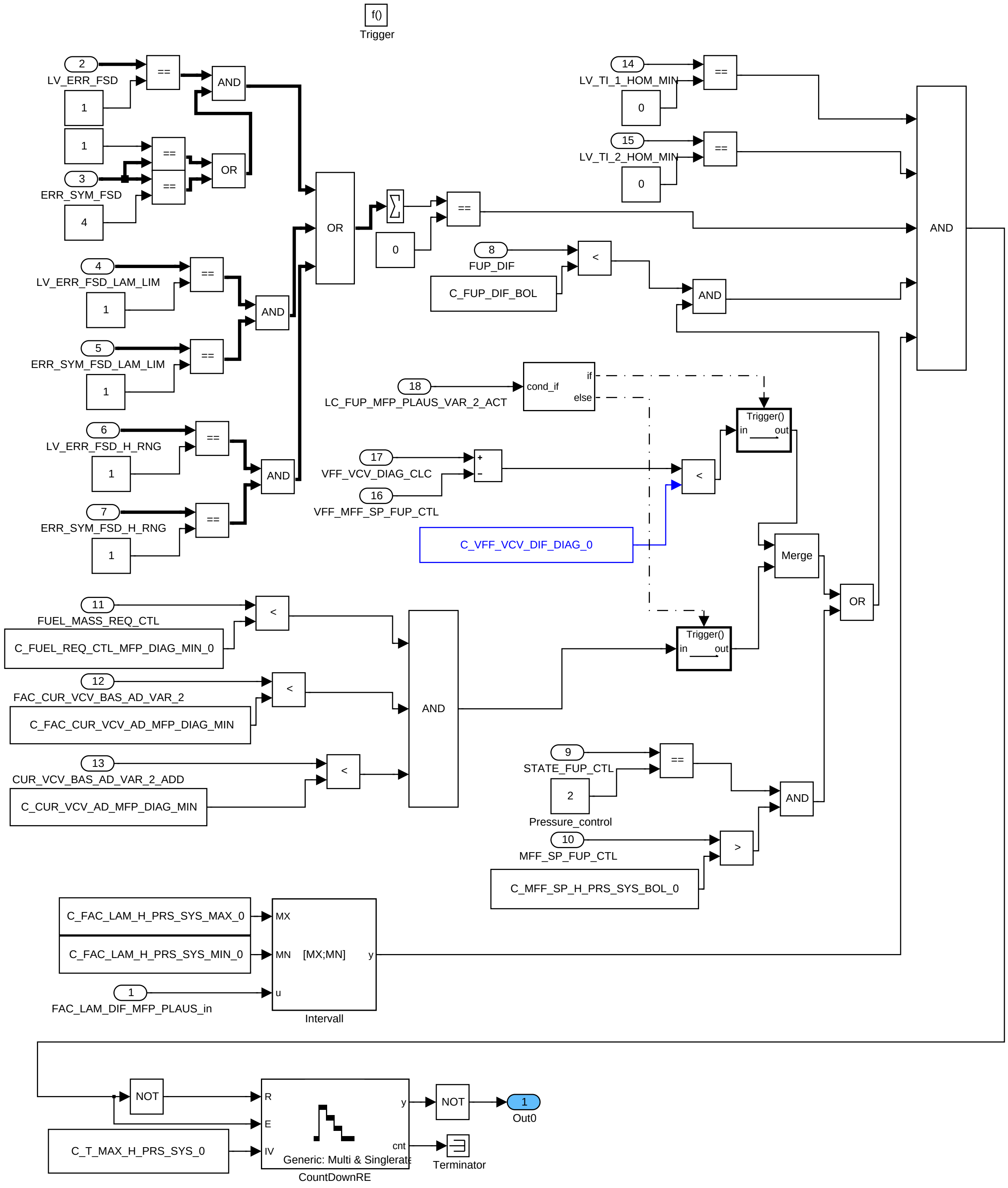


Figure 261 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ DIAG_H_PRS_SYS/ H_PRS_SYS_SYM_0

FUSL_MB01G/FUEL_SYS_PRS_DIAG/DET_MFP_PLAUS_H_PRS_SYS/DIAG_H_PRS_SYS/H_PRS_SYS_SYM_1

Detection of H_PRS_SYS SYM_1, delivery of pump is too low.

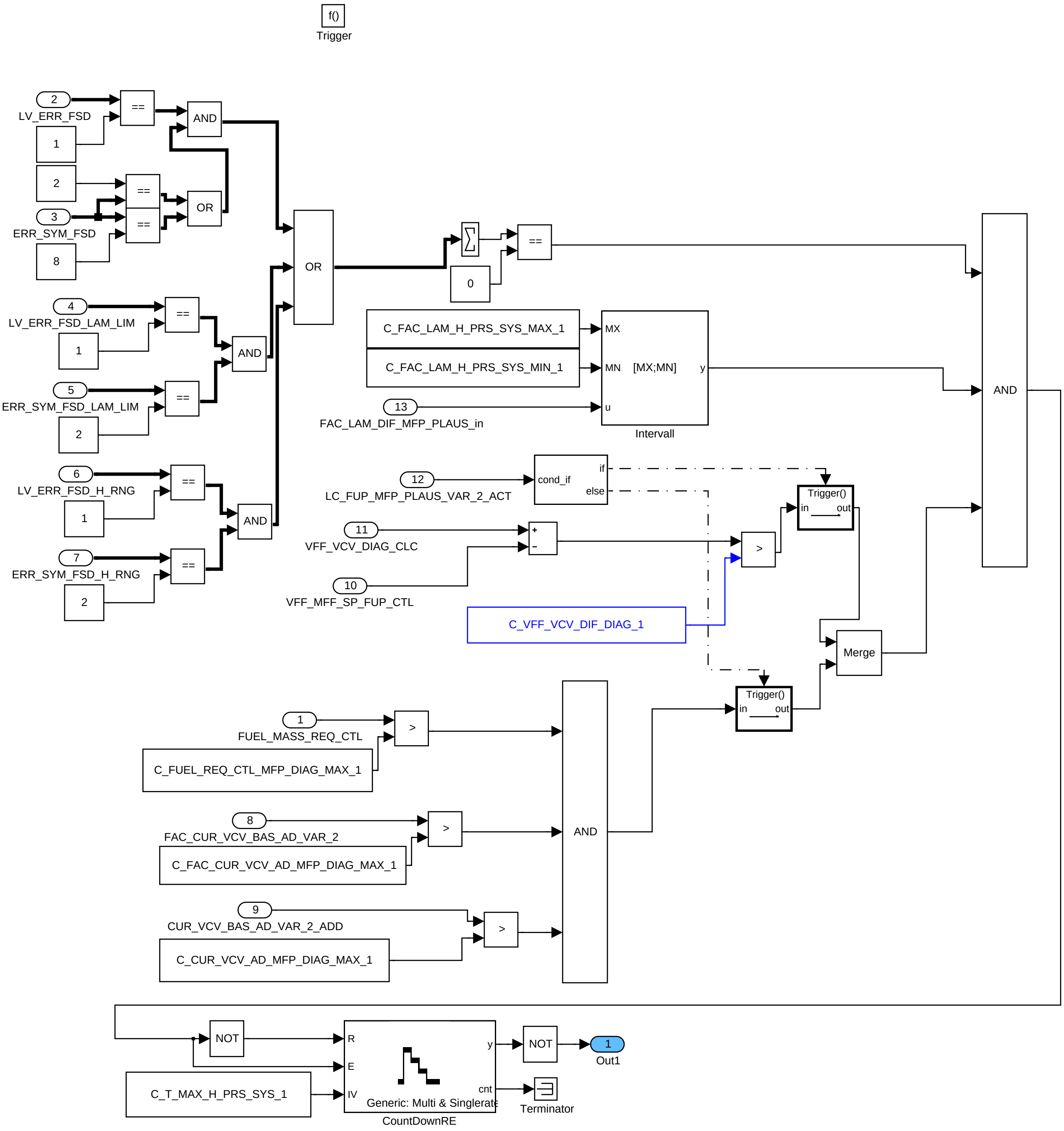


Figure 262 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ DIAG_H_PRS_SYS/ H_PRS_SYS_SYM_1

FUSL_MB01G/FUEL_SYS_PRS_DIAG/DET_MFP_PLAUS_H_PRS_SYS/DIAG_H_PRS_SYS/H_PRS_SYS_SYM_2

Detection of H_PRS_SYS SYM_2, delivery of pump is too low, or OPV is open, or VCV stuck at low delivery.

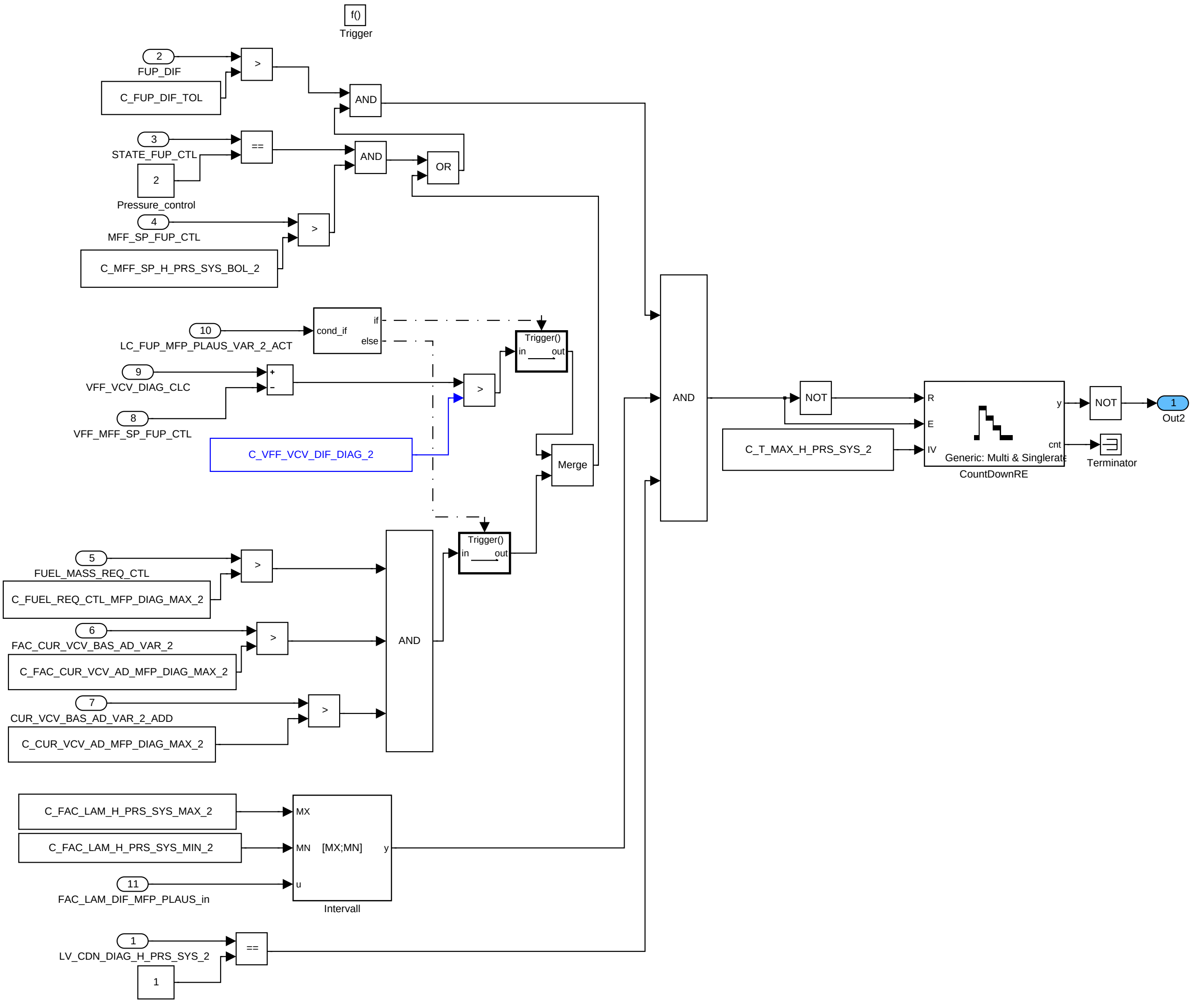


Figure 263 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ DIAG_H_PRS_SYS/ H_PRS_SYS_SYM_2

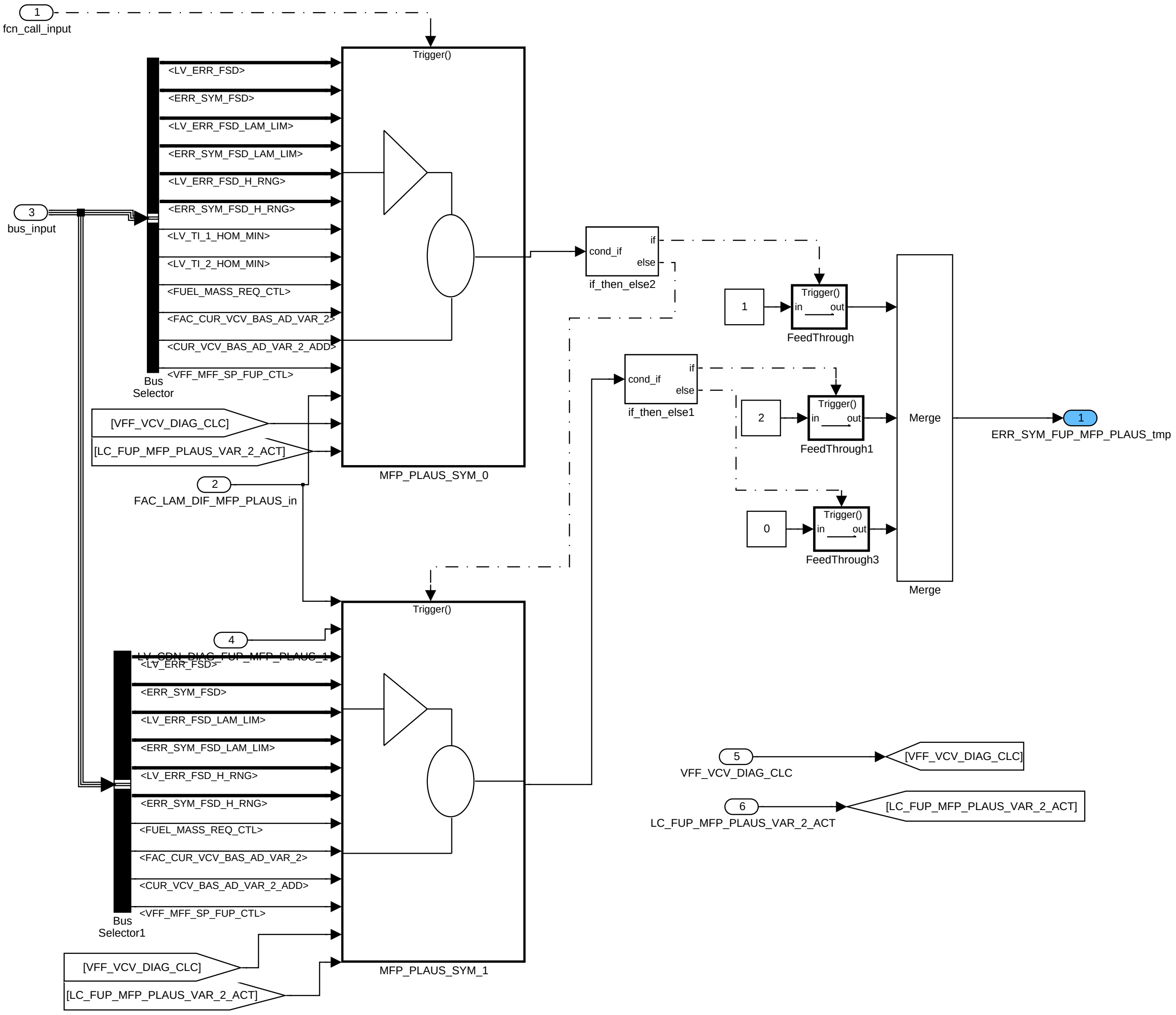
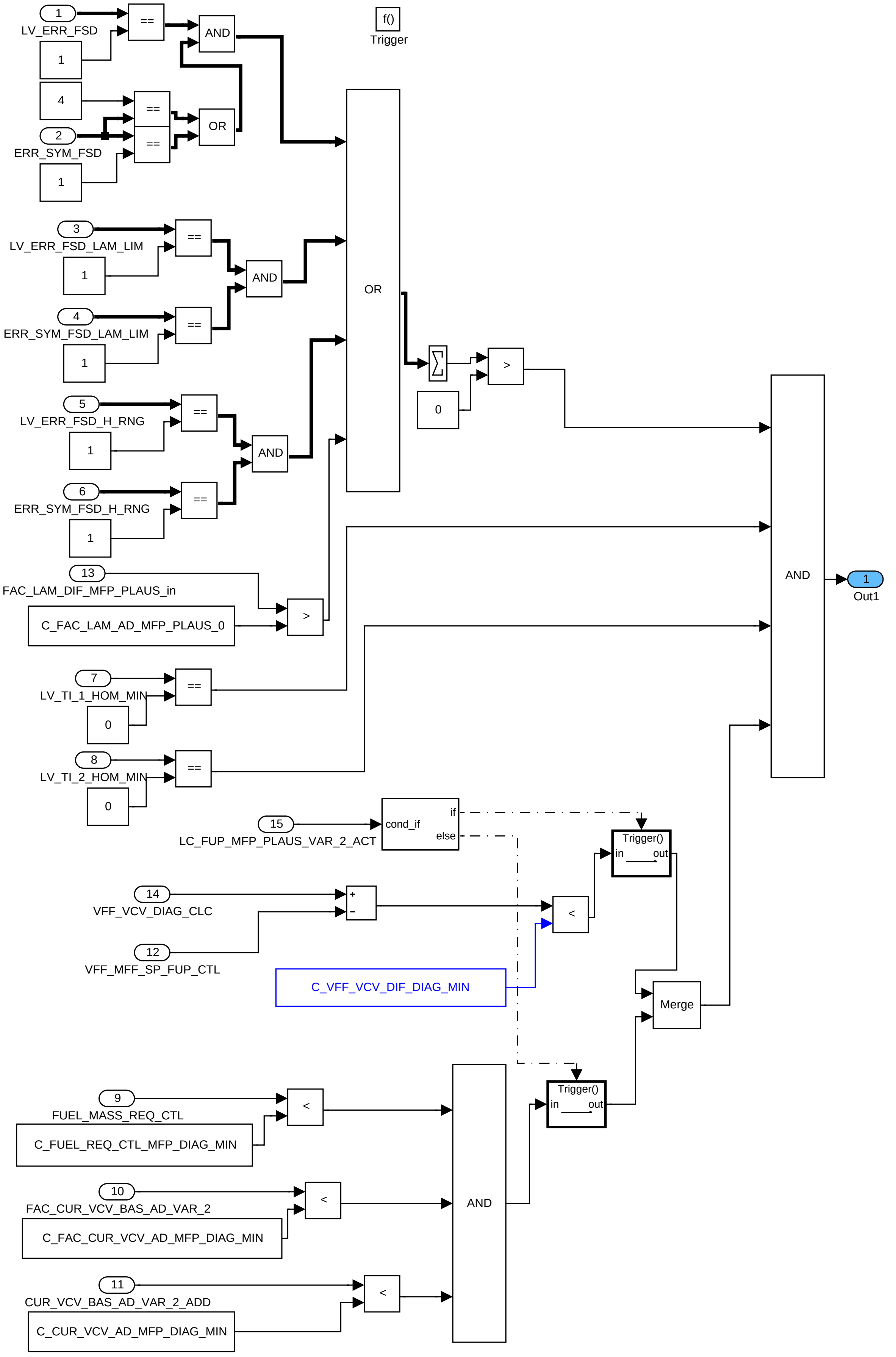


Figure 264 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ DIAG_MFP_PLAUS

FUSL_MB01G/FUEL_SYS_PRS_DIAG/DET_MFP_PLAUS_H_PRS_SYS/DIAG_MFP_PLAUS/MFP_PLAUS_SYM_0

Detection of FUP_MFP_PLAUS SYM_0, FUP sensor shows a too high value.



Chapter Fuel supply	Baseline 4DC3940S		Include File 43B01G01.00A
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

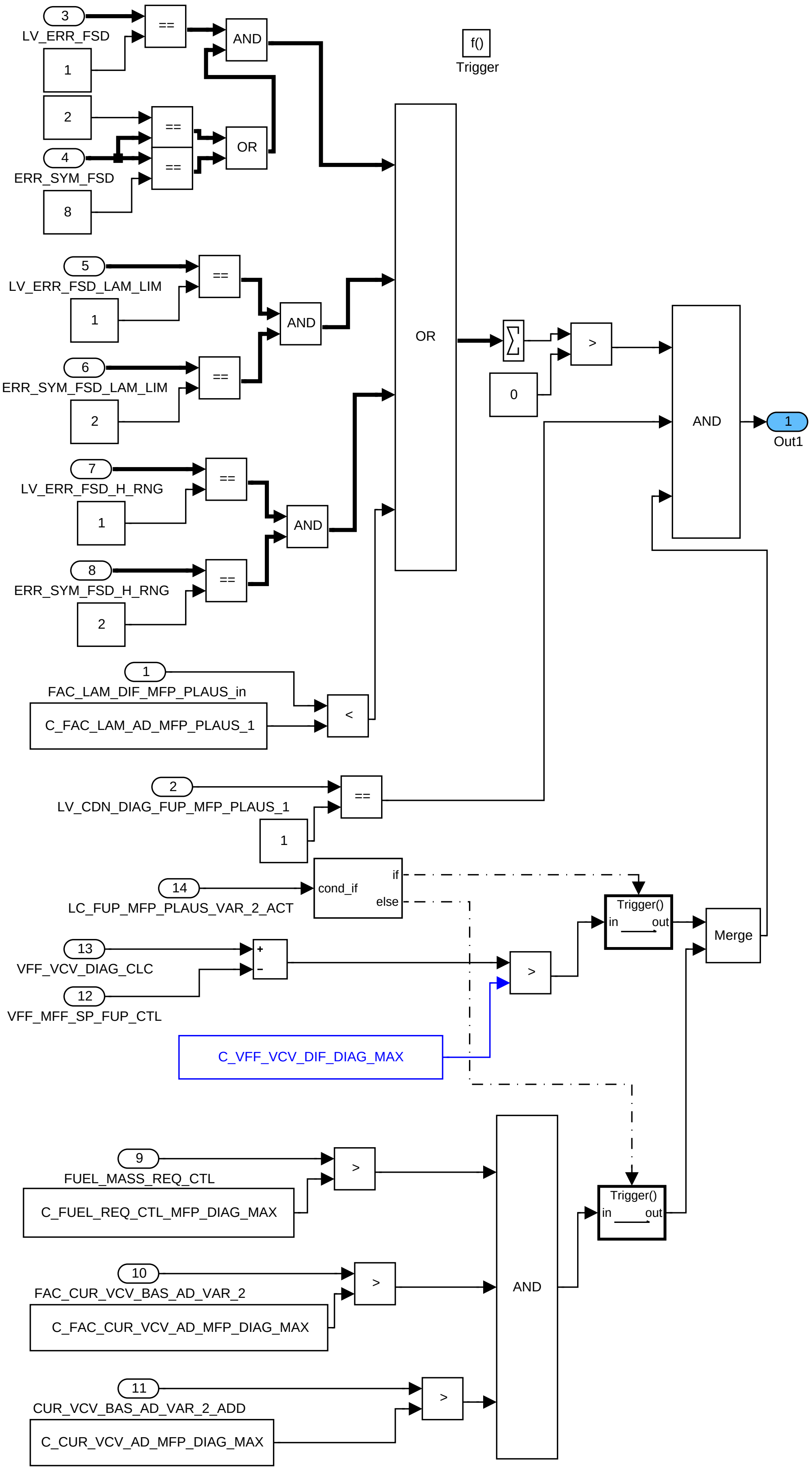
Document Key
E002-190.49.02 SPE 000 48.0

Pages
5846 of 9643

Figure 265 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ DIAG_MFP_PLAUS/ MFP_PLAUS_SYM_0

FUSL_MB01G/FUEL_SYS_PRS_DIAG/DET_MFP_PLAUS_H_PRS_SYS/DIAG_MFP_PLAUS/MFP_PLAUS_SYM_1

Detection of FUP_MFP_PLAUS SYM_1, FUP sensor shows a too low value.



3
LV_ERR_FSD
1
==
AND
2
4
ERR_SYM_FSD
8
==
==
OR
f()
Trigger
5
LV_ERR_FSD_LAM_LIM
1
==
AND
6
ERR_SYM_FSD_LAM_LIM
2
==
OR
∑
>
0
AND
1
Out1
7
LV_ERR_FSD_H_RNG
1
==
AND
8
ERR_SYM_FSD_H_RNG
2
==
1
FAC_LAM_DIF_MFP_PLAUS_in
C_FAC_LAM_AD_MFP_PLAUS_1
<
2
LV_CDN_DIAG_FUP_MFP_PLAUS_1
==
1
14
LC_FUP_MFP_PLAUS_VAR_2_ACT
cond_if
if
else
Trigger()
in
out
Merge
13
VFF_VCV_DIAG_CLC
+
-
12
VFF_MFF_SP_FUP_CTL
>
C_VFF_VCV_DIF_DIAG_MAX
9
FUEL_MASS_REQ_CTL
>
C_FUEL_REQ_CTL_MFP_DIAG_MAX
Trigger()
in
out
10
FAC_CUR_VCV_BAS_AD_VAR_2
>
C_FAC_CUR_VCV_AD_MFP_DIAG_MAX
AND
11
CUR_VCV_BAS_AD_VAR_2_ADD
>
C_CUR_VCV_AD_MFP_DIAG_MAX

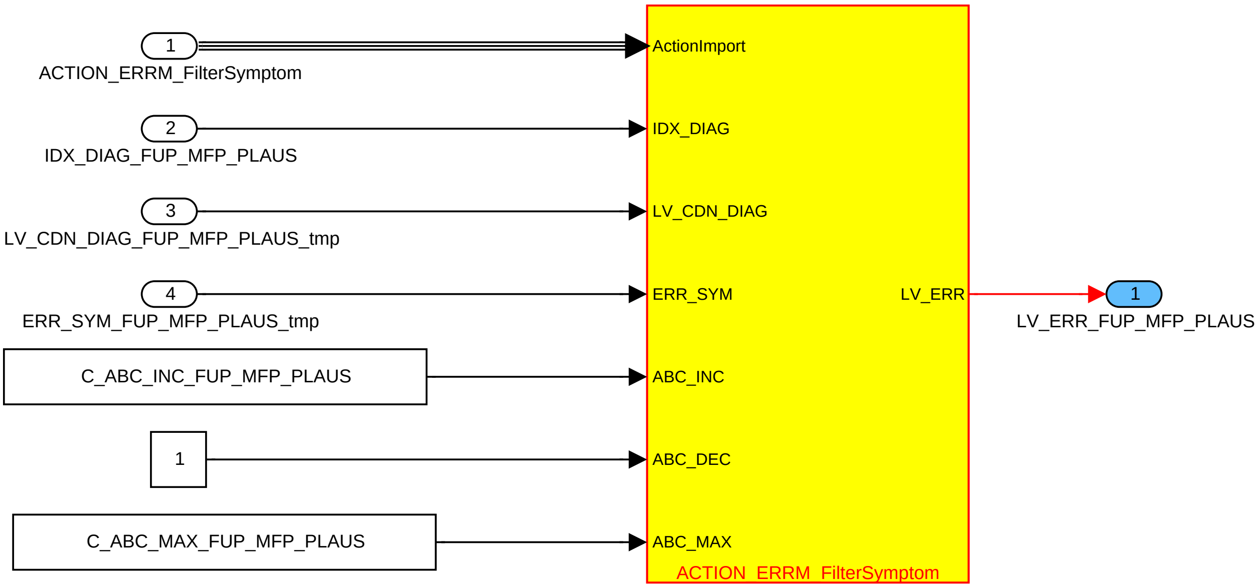


Figure 266 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ DIAG_MFP_PLAUS/ MFP_PLAUS_SYM_1

Figure 267 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ ERRM_IF1

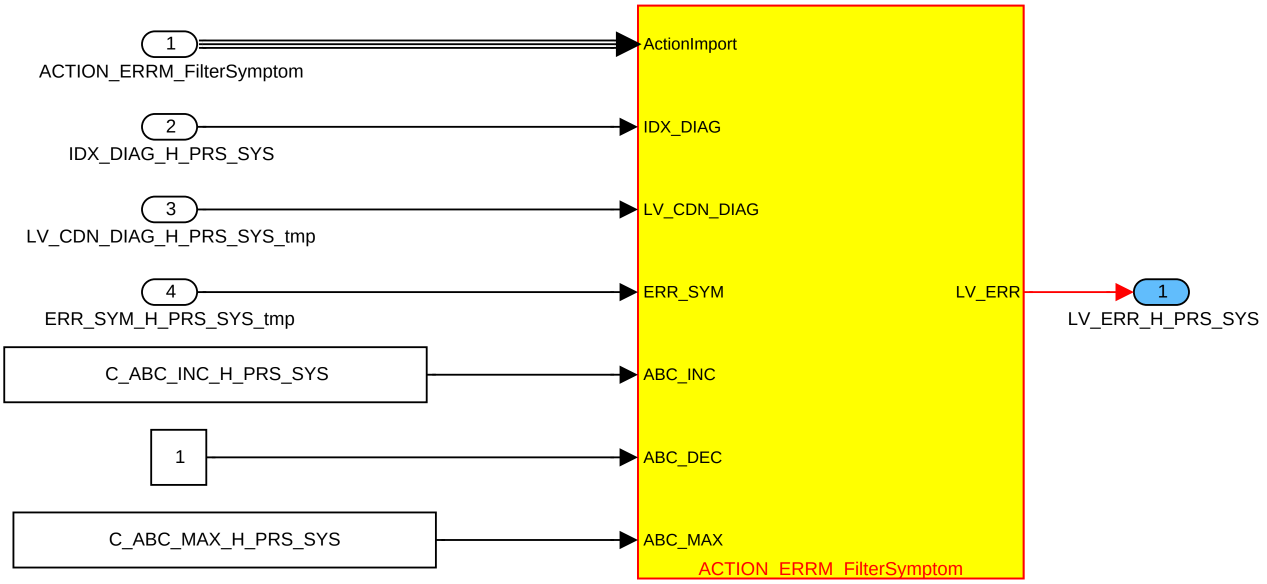
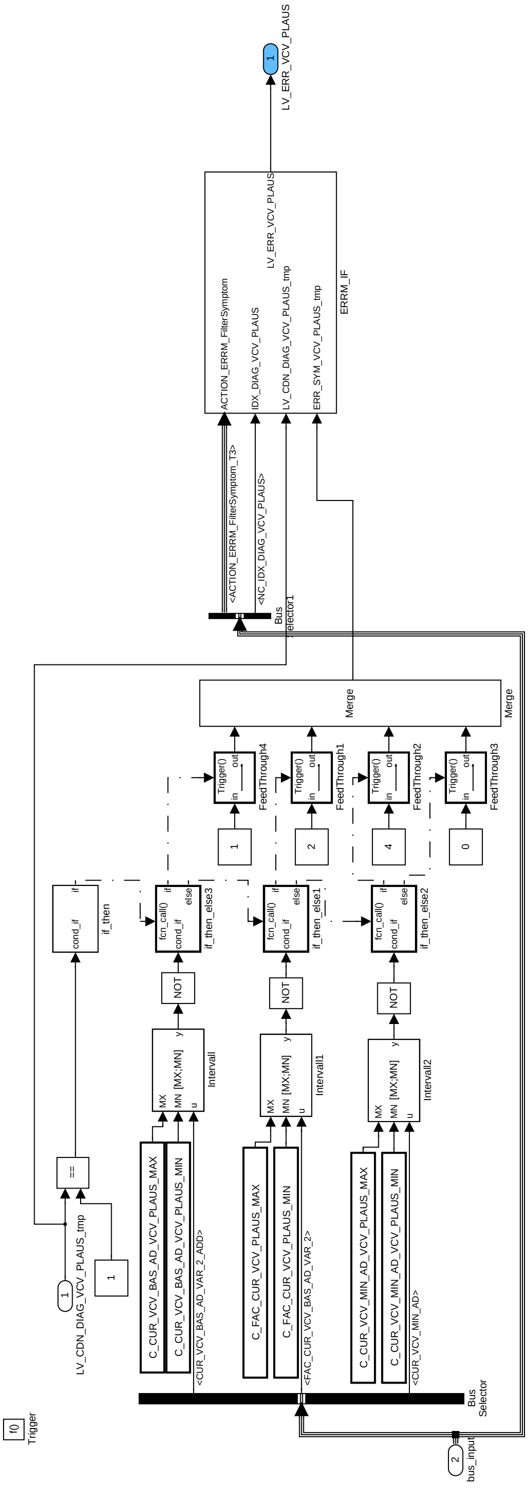


Figure 268 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_MFP_PLAUS_H_PRS_SYS/ ERRM_IF2

FUSL_MB01G/FUEL_SYS_PRS_DIAG/DET_VCV_PLAUS



Plausibility check of VCV adaptation values.

Figure 269 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_VCV_PLAUS

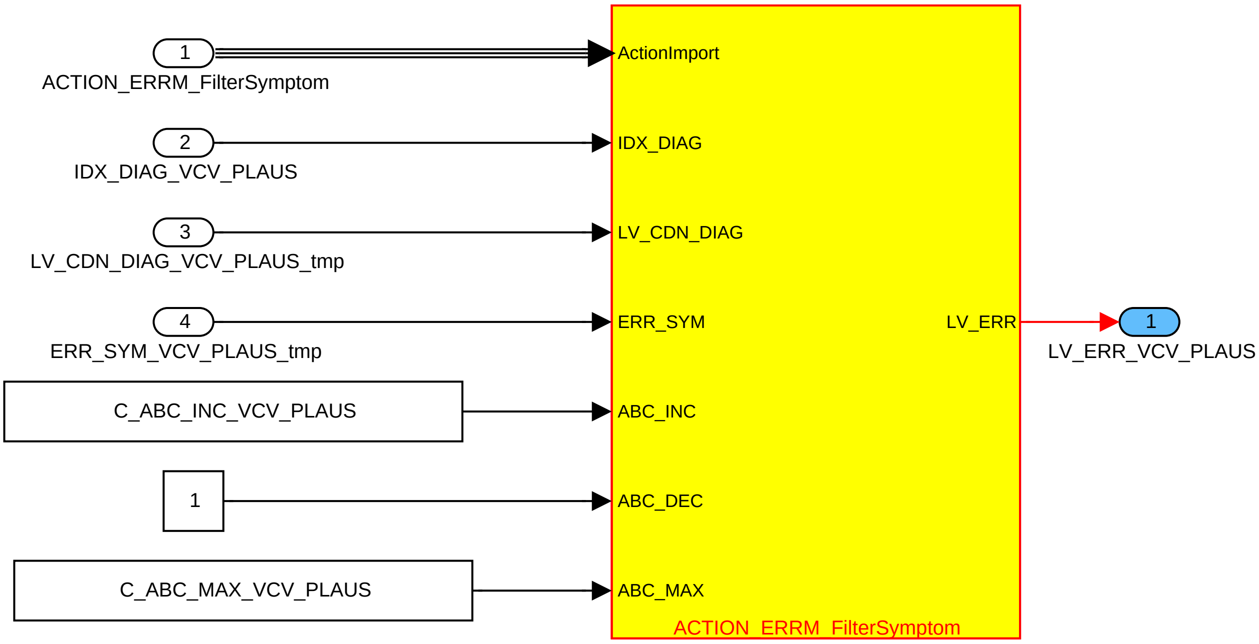


Figure 270 FUSL_MB01G/ FUEL_SYS_PRS_DIAG/ DET_VCV_PLAUS/ ERRM_IF

34.25.1.4 FUSL_MB01G/H_PRS_ST_DIAG

Detection of fuel pressure at start of injection is to low.

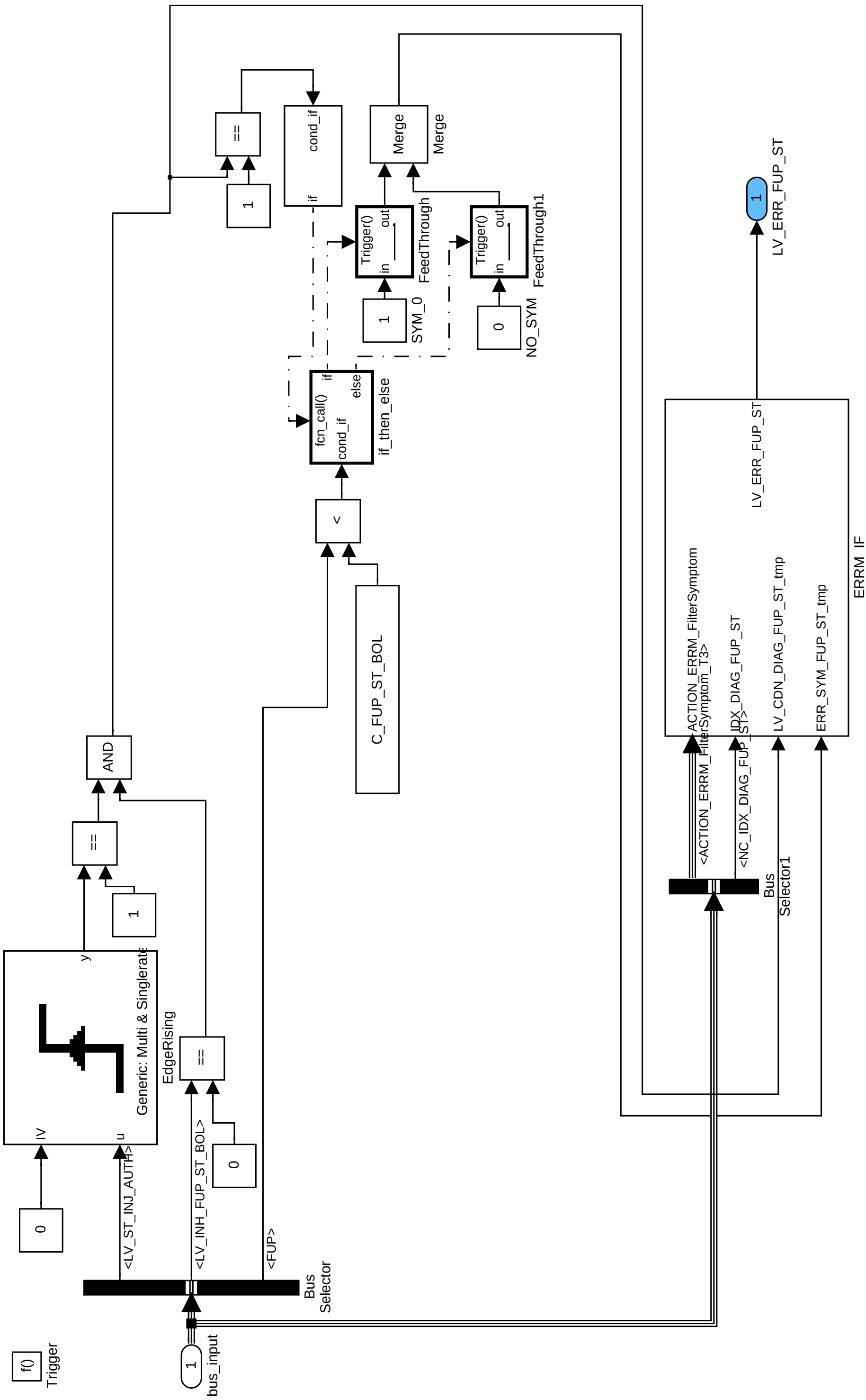


Figure 271 FUSL_MB01G/ H_PRS_ST_DIAG

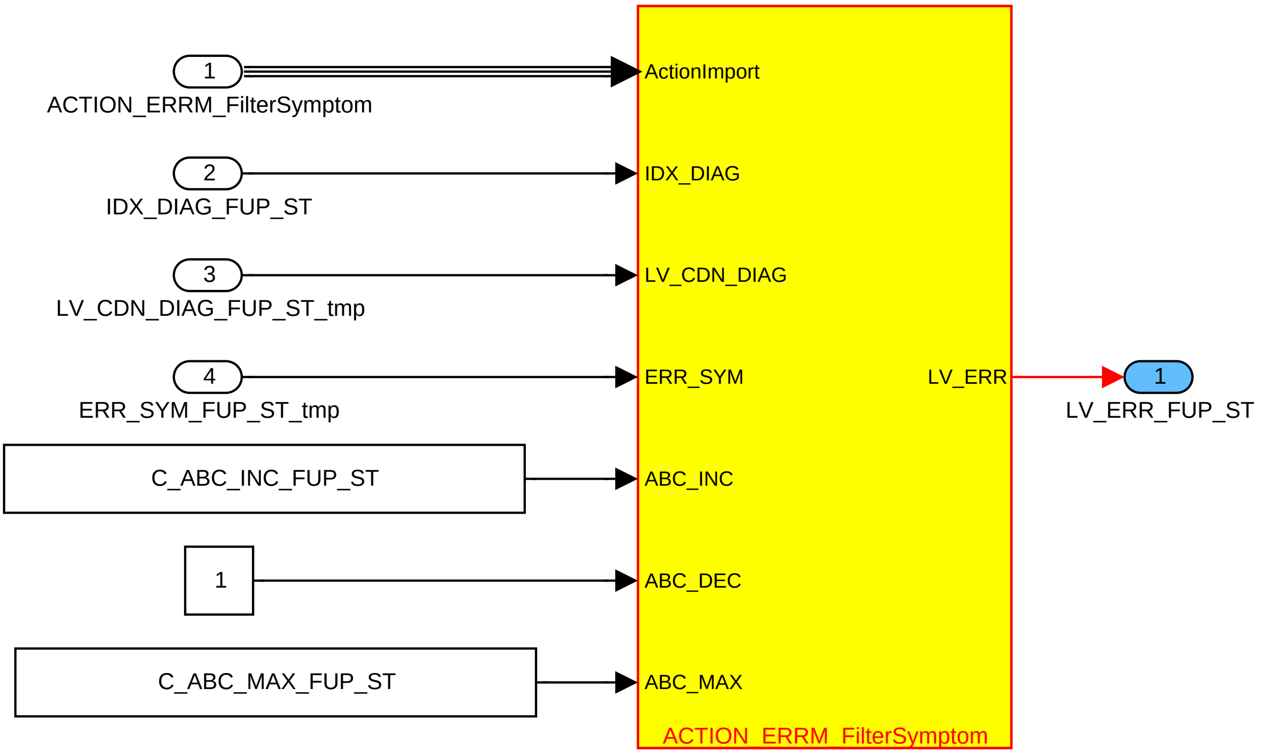


Figure 272 FUSL_MB01G/ H_PRS_ST_DIAG/ ERRM_IF

34.25.1.5 SUBFUNCTION: INI

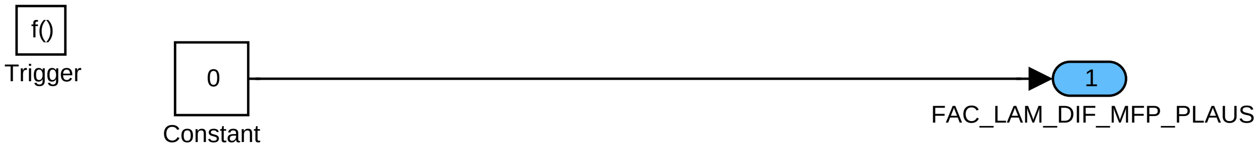


Figure 273 FUSL_MB01G/ INI

34.26 Application incidences for fuel system pressure diagnosis

Data Definition:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_INH_FUP_MFP_PLAUS	O/V	0... 1H	0... 1	1	[-]
Boolean flag to stop the plausibility check of FUP/MFP					
LV_INH_FUP_ORNG	O/V	0... 1H	0... 1	1	[-]
Boolean flag to stop the FUP ORNG check					
LV_INH_FUP_SENS_PLAUS	O/V	0... 1H	0... 1	1	[-]
Boolean flag to stop the FUP sensor plausibility check					
LV_INH_FUP_ST_BOL	O/V	0... 1H	0... 1	1	[-]
Boolean flag to stop the diagnosis of FUP at start of injection					
LV_INH_FUP_STOP	O/V	0... 1H	0... 1	1	[-]
Boolean flag to stop the FUP_STOP check					
LV_INH_H_PRS_SYS	O/V	0... 1H	0... 1	1	[-]
Boolean flag to stop the high pressure system monitoring					
LV_INH_VCV_PLAUS	O/V	0... 1H	0... 1	1	[-]
Boolean flag to stop the VCV_PLAUS check					

Input Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CL_MMV		0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Moving mean value of the canister load CL					
FUP_SP		0... FFFFH	0... 347776	5.3067216	[hPa]
Fuel pressure setpoint					
LV_ACT_DIAGCPS		0... 1H	0... 1	1	[-]
Indication for functional check cps (step 2 or 3) being activated					
LV_ERR_EL_CPS		0... 1H	0... 1	1	[-]
Logical value for present electrical canister purge valve failure					
LV_ERR_FUP		0... 1H	0... 1	1	[-]
FUP sensor error detected					
LV_ERR_FUP_MFP_PLAUS		0... 1H	0... 1	1	[-]
Boolean that indicates inconsistencies between fuel pressure and mass fuel pump					
LV_ERR_FUP_ORNG		0... 1H	0... 1	1	[-]
Boolean that indicates fuel pressure out of range					
LV_ERR_LS_DOWN [NC_CBK_EX_NR]		0... 1H	0... 1	1	[-]
Final diagnostic of the downstream oxygen sensor					
LV_ERR_LS_UP [NC_CBK_EX_NR]		0... 1H	0... 1	1	[-]
Boolean error flag, fault currently present on upstream oxygen sensor signal					
LV_ERR_MEC_OPEN_CPS		0... 1H	0... 1	1	[-]
Mechanical failure on canister purge valve					
LV_ERR_RLY_VCV		0... 1H	0... 1	1	[-]
Boolean for error currently present on RLY_VCV command signal					
LV_ERR_VCV		0... 1H	0... 1	1	[-]
Boolean for error currently present on VCV					
LV_IGK		0... 1H	0... 1	1	[-]
Ignition key on					
LV_MIS_STATE_A		0... 1H	0... 1	1	[-]
CARB A misfire criterion confirmed (debounced)					
LV_MIS_STATE_B		0... 1H	0... 1	1	[-]
CARB B (B1 or B4) misfire criterion confirmed (debounced)					
LV_VCV_RLY		0... 1H	0... 1	1	[-]

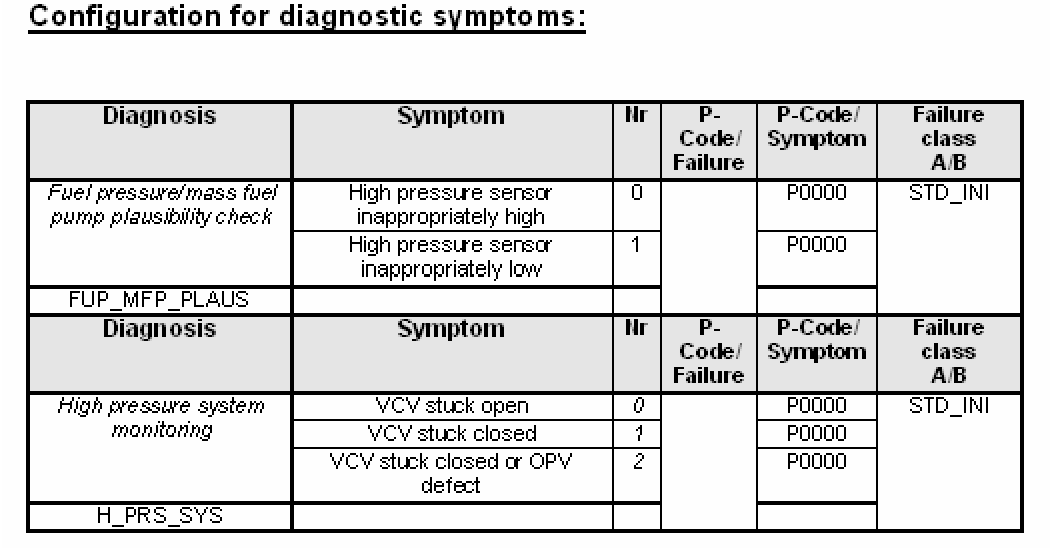
Flag for controlling the volume control valve relay					
N_32		0... FFH	0... 8160	32	[rpm]
Engine speed -Resolution 32 rpm					
STATE_CP		0 1 2 3 4 5 6 7 8 9 10 11	CP_NOT_ACT NO_PURGE RAMP_TO_NO_PURGE WAIT_RAMP_OPEN MIN_PURGE - - - RAMP_OPEN RAMP_OPEN_FAST MAX_PURGE RAMP_CLOSE	-	[-]
State of Evaporative Emission Control Function					
STATE_ERR_IV		0... FFH	0... 255	1	[-]
Error pattern after debouncing for all injectors					
T_AST		0... FFFFH	0... 6553.5	0.1	[s]
Time after exit start to after start					
VFF_MFF_SP_FUP_CTL		0... FFFFH	0... 255	3.89105e-3	[l/h]
Volume fuel flow through the injectors					

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CL_MAX_DIAG_MFP_PLAUS	V	0... FFFFH	0... 7.99987792969	122.07e-6	[-]
Maximum canister load to detect a fault in highpressure system					
C_FUP_SP_DIAG_MAX	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Maximum FUP_SP to enable high pressure dianosis					
C_FUP_SP_DIAG_MIN	V	0... FFFFH	0... 347776	5.3067216	[hPa]
Minimum FUP_SP to enable high pressure dianosis					
C_T_AST_H_PRS_DIAG_INH	V	0... FFFFH	0... 6553.5	0.1	[s]
Inhibit time for deactivation of the high pressure system diagnosis					
C_T_FUP_ORNG_DIAG_INH	V	0... FFFFH	0... 1310.7	0.02	[s]
Inhibit time for deactivation of the FUP ORNG diagnosis					
C_T_H_PRS_DIAG_INH	V	0... FFFFH	0... 1310.7	0.02	[s]
Inhibit time for deactivation of the high pressure system diagnosis					
IP_VFF_MAX_FUP_DIAG	V	0... FFFFH	0... 255	3.89105e-3	[l/h]
LDP_N_32_IP_VFF_MAX_FUP_DIAG	4	0... FFH	0... 8160	32	[rpm]
Maximun fuel flow to enable high pressure diagnosis					

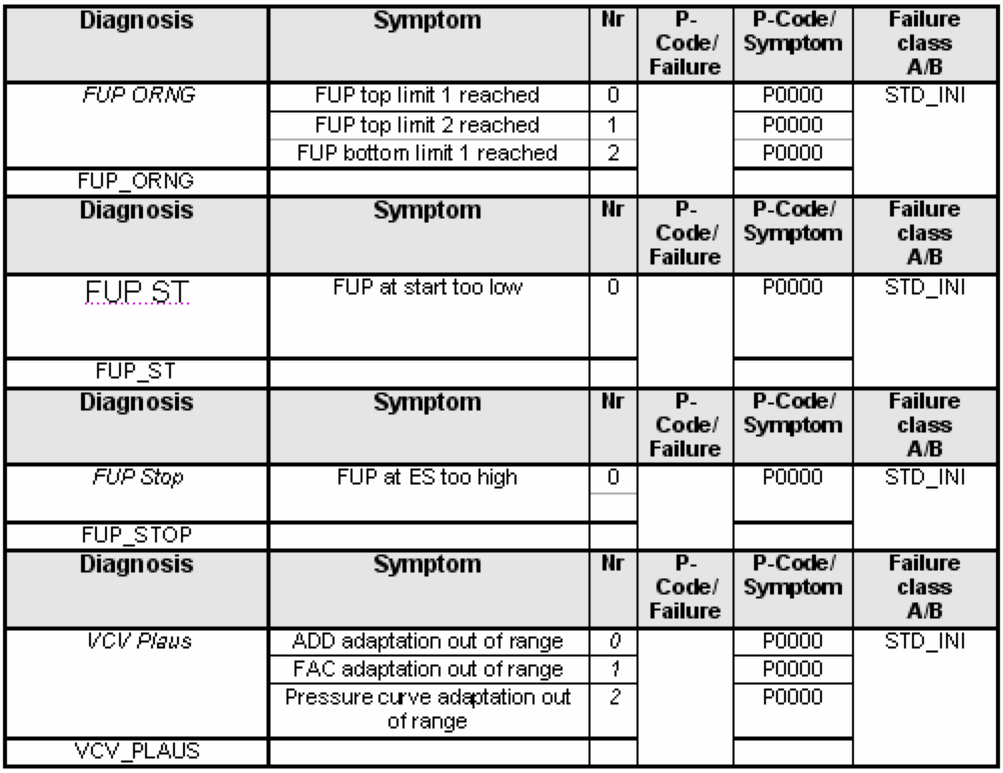
General Information

To inhibit the fuel system pressure diagnosis, several Diagnostic errors examined.



Configuration for diagnostic symptoms:

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
Fuel pressure/mass fuel pump plausibility check	High pressure sensor inappropriately high	0		P0000	STD_INI
	High pressure sensor inappropriately low	1		P0000	
FUP_MFP_PLAUS					
Diagnosis	**Symptom**	**Nr**	**P-Code/ Failure**	**P-Code/ Symptom**	**Failure class A/B**
High pressure system monitoring	VCV stuck open	*0*		P0000	STD_INI
	VCV stuck closed	*1*		P0000	
	VCV stuck closed or OPV defect	*2*		P0000	
H_PRS_SYS					

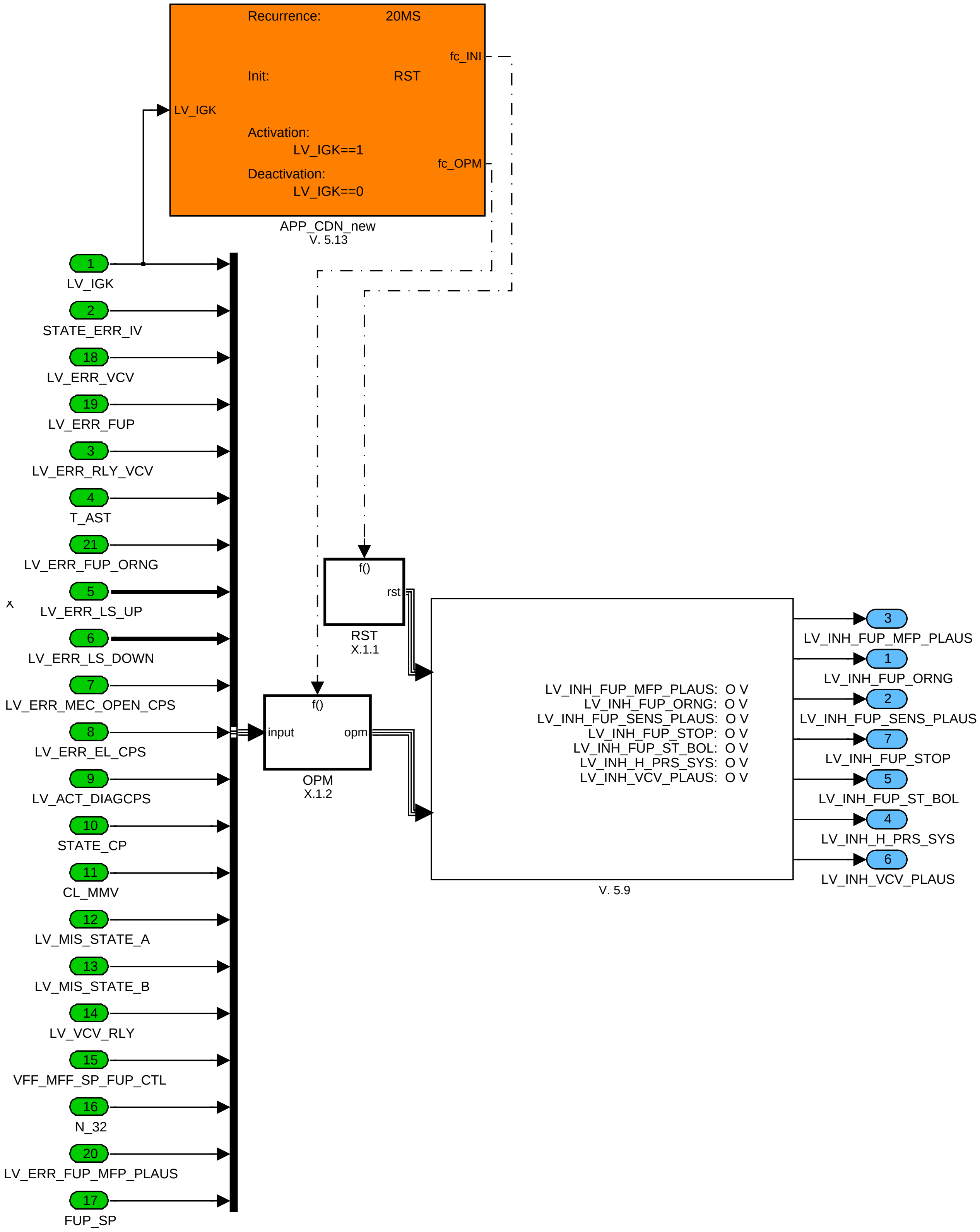


Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
FUP ORNG	FUP top limit 1 reached	0		P0000	STD_INI
	FUP top limit 2 reached	1		P0000	
	FUP bottom limit 1 reached	2		P0000	
FUP_ORNG					
Diagnosis	**Symptom**	**Nr**	**P-Code/ Failure**	**P-Code/ Symptom**	**Failure class A/B**
FUP ST	FUP at start too low	0		P0000	STD_INI
FUP_ST					
Diagnosis	**Symptom**	**Nr**	**P-Code/ Failure**	**P-Code/ Symptom**	**Failure class A/B**
FUP Stop	FUP at ES too high	0		P0000	STD_INI
FUP_STOP					
Diagnosis	**Symptom**	**Nr**	**P-Code/ Failure**	**P-Code/ Symptom**	**Failure class A/B**
VCV Plaus	ADD adaptation out of range	*0*		P0000	STD_INI
	FAC adaptation out of range	*1*		P0000	
	Pressure curve adaptation out of range	*2*		P0000	
VCV_PLAUS					

Application Conditions

Initialization: RST
Recurrence: 20MS
Activation: LV_IGK==1
Deactivation: LV_IGK==0

Function description



SDA_SRS / SDA V 5.0.4 24−Mar−2006

Figure 274:
Path: FUSL_MB01I

34.26.1 Function Description:

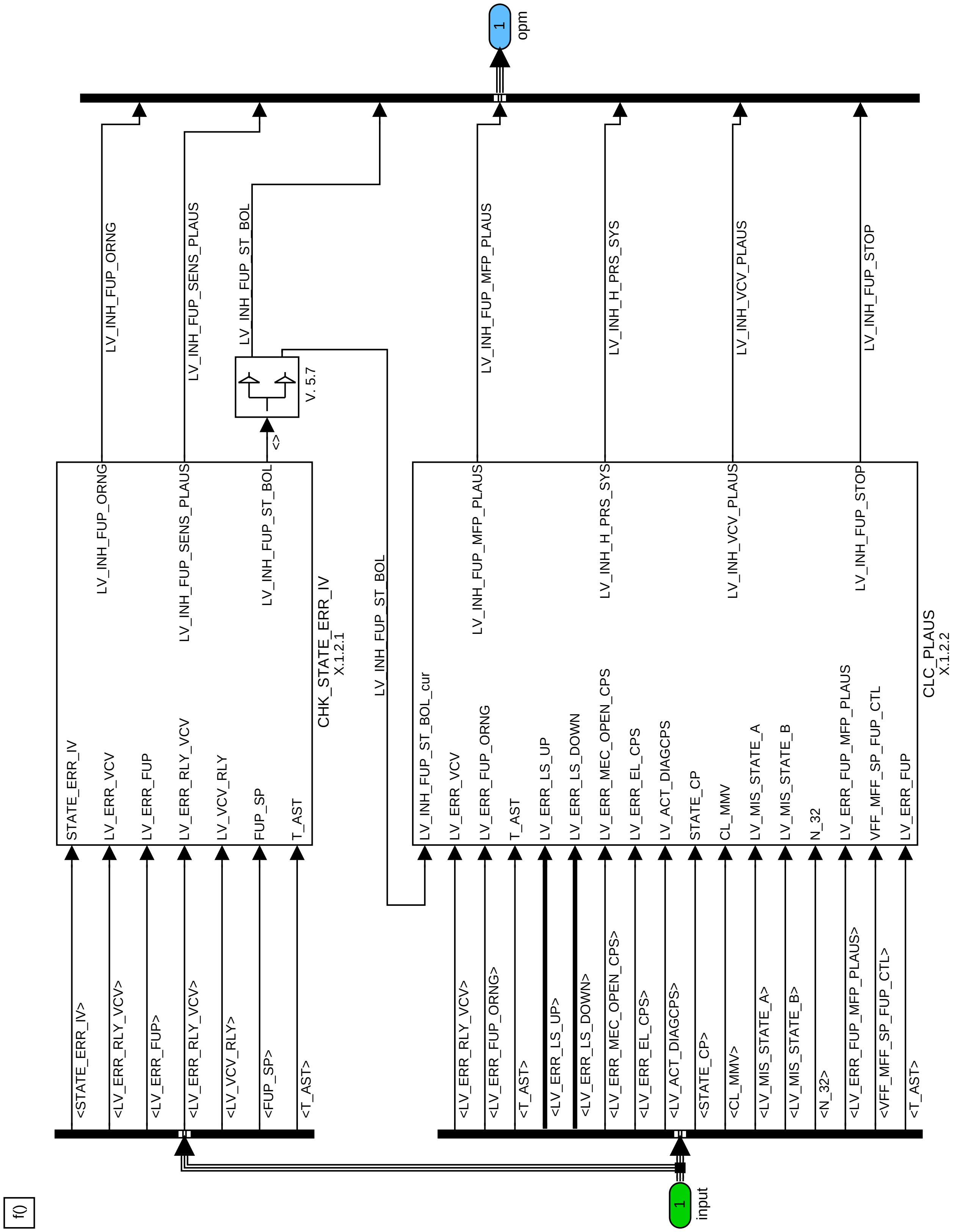


Figure 275:
Path: FUSL_MB01I/OPM

34.26.1.1 Calculation of inhibition fuel pressure:

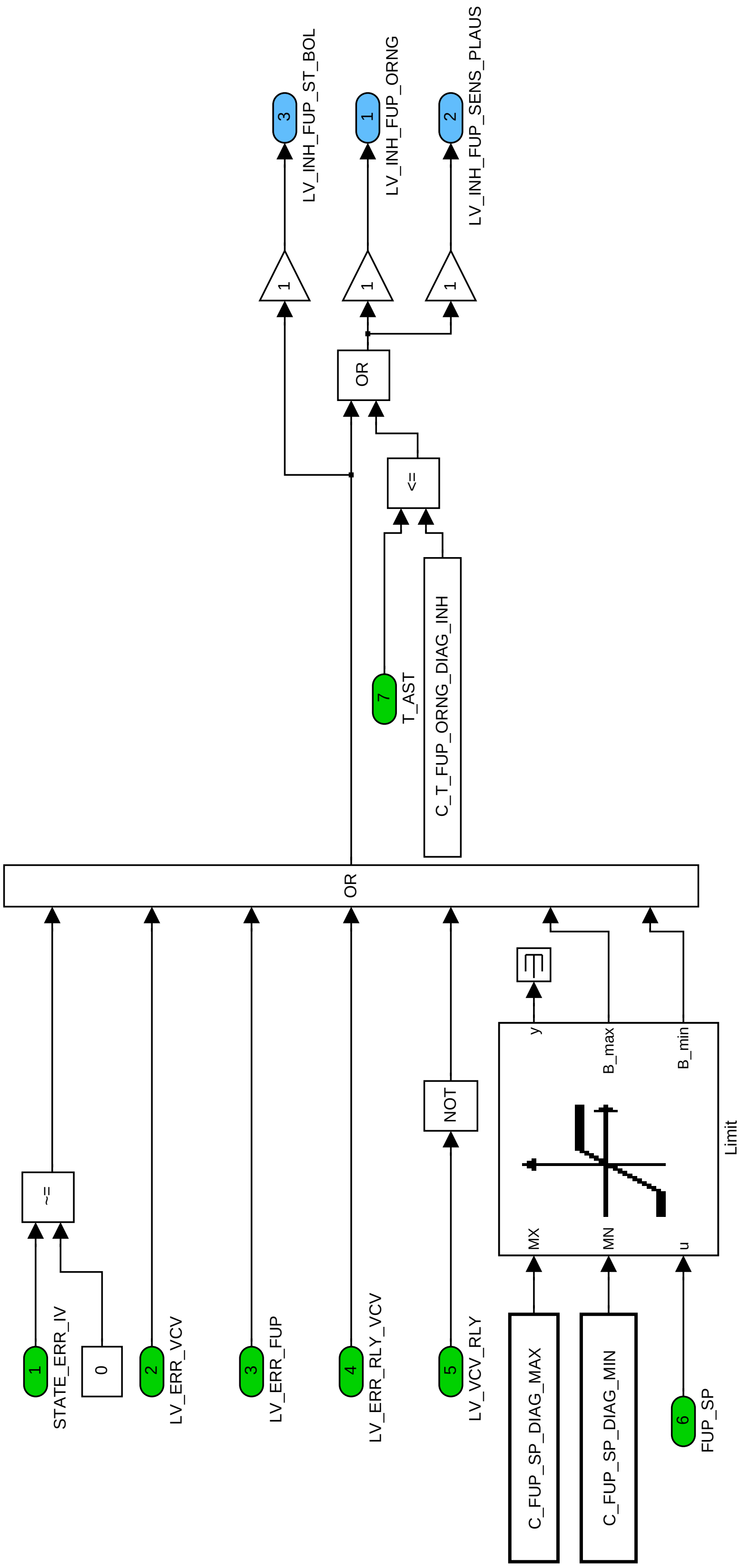


Figure 276:
Path: FUSL_MB01I/OPM/CHK_STATE_ERR_IV

34.26.1.2 Calculation for mass fuel pump:

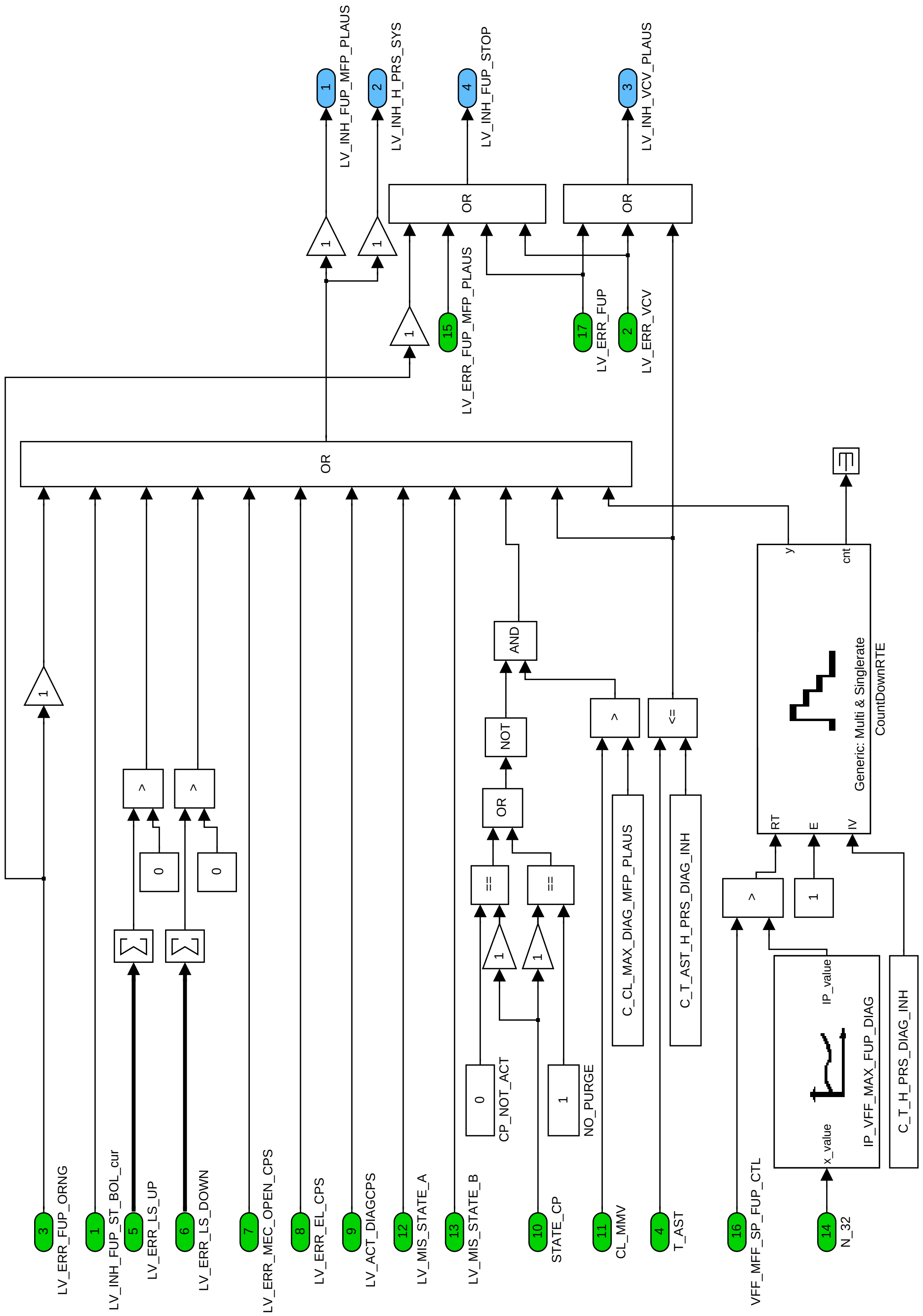


Figure 277:
Path: FUSL_MB01I/OPM/CLC_PLAUS

34.26.2 Initialisation:

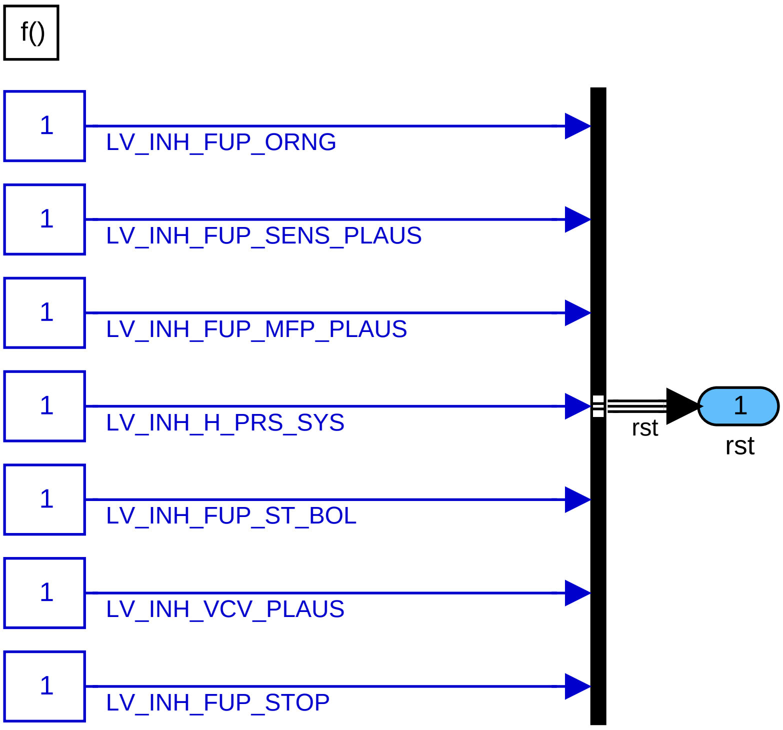
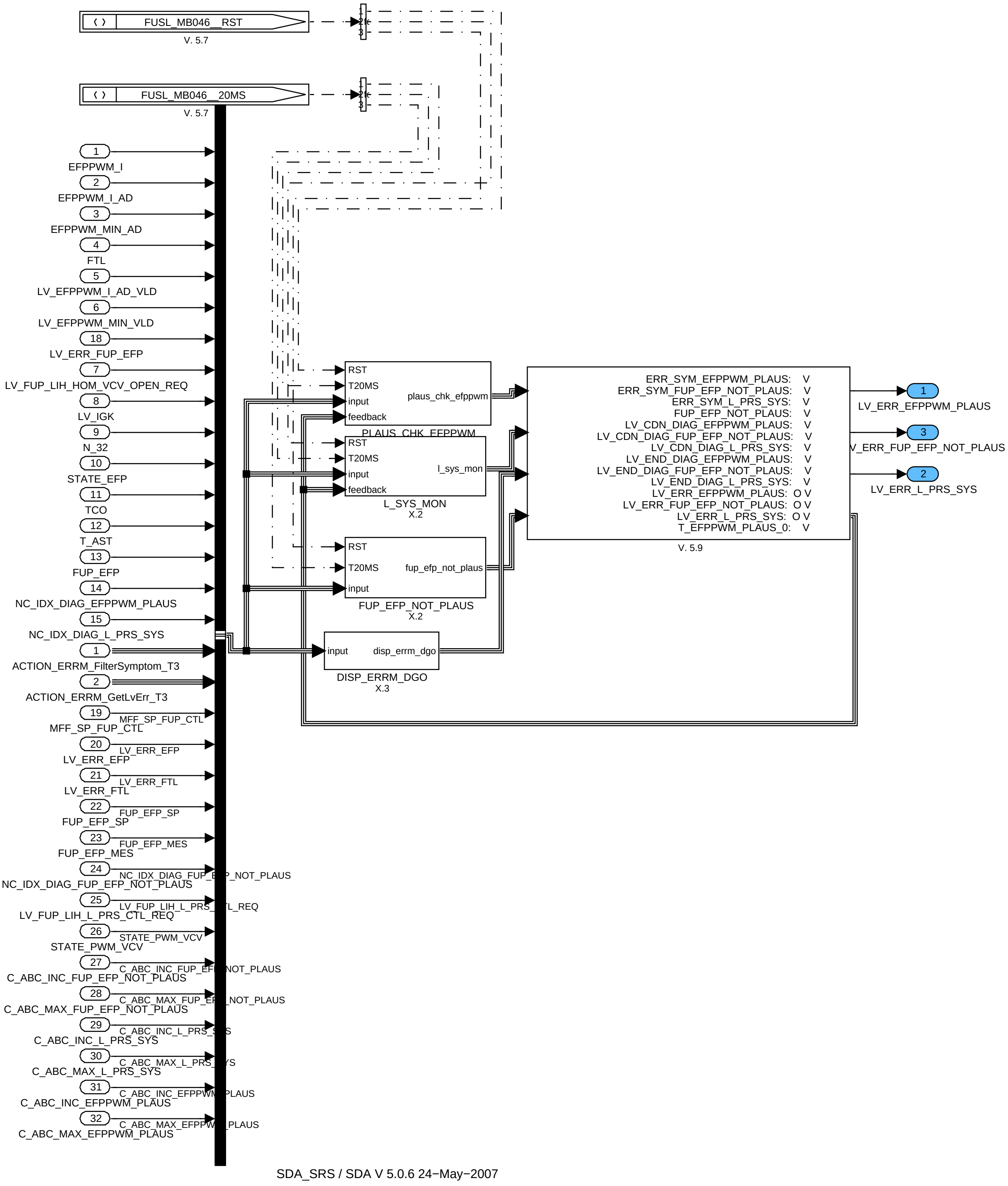


Figure 278:
Path: FUSL_MB01I/RST

34.27 Fuel system pressure diagnosis for low pressure

Overview



SDA_SRS / SDA V 5.0.6 24−May−2007

Figure 279:
Path: FUSL_MB046

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
ERR_SYM_EFPPWM_PLAUS	V	0 1 2 4 8	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	-	[-]
Detected symptom EFPPWM plausibility check					
ERR_SYM_FUP_EFP_NOT_PLAUS	V	0 1 2 4 8	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	-	[-]
Detected symptom for FUP_EFP_NOT_PLAUS diagnosis					
ERR_SYM_L_PRS_SYS	V	0 1 2 4 8	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	-	[-]
Detected symptom for low pressure system monitoring					
FUP_EFP_NOT_PLAUS	V	0... FFFFH	0... 173888	2.6533608	[hPa]
Filtered FUP_EFP for FUP_EFP_NOT_PLAUS diagnosis					
LV_CDN_DIAG_EFPPWM_PLAUS	V	0... 1H	0... 1	1	[-]
Status of diagnosis flag for EFPPWM plausibility check					
LV_CDN_DIAG_FUP_EFP_NOT_PLAUS	V	0... 1H	0... 1	1	[-]
Condition flag for FUP_EFP_NOT_PLAUS diagnosis					
LV_CDN_DIAG_L_PRS_SYS	V	0... 1H	0... 1	1	[-]
Status of diagnosis flag for low pressure system monitoring					
LV_END_DIAG_EFPPWM_PLAUS	V	0... 1H	0... 1	1	[-]
End of diagnosis flag for EFPPWM plausibility check					
LV_END_DIAG_FUP_EFP_NOT_PLAUS	V	0... 1H	0... 1	1	[-]
End diag flag for FUP_EFP_NOT_PLAUS diagnosis					
LV_END_DIAG_L_PRS_SYS	V	0... 1H	0... 1	1	[-]
End of diagnosis flag for low pressure system monitoring					
LV_ERR_EFPPWM_PLAUS	O/V	0... 1H	0... 1	1	[-]
Boolean that indicates inconsistencies of the EFPPWM signal					
LV_ERR_FUP_EFP_NOT_PLAUS	O/V	0... 1H	0... 1	1	[-]
Error flag for FUP_EFP_NOT_PLAUS diagnosis					
LV_ERR_L_PRS_SYS	O/V	0... 1H	0... 1	1	[-]
Boolean that indicates abnormal fuel pressure value in low pressure fuel system					
T_EFPPWM_PLAUS_0	V	0... FFH	0... 5.1	0.02	[s]
Delaytime for EFPPWM_PLAUS SYM 0					

Input Data:

EFPPWM_I	EFPPWM_I_AD	EFPPWM_MIN_AD	FTL
LV_EFPPWM_I_AD_VLD	LV_EFPPWM_MIN_VLD	LV_FUP_LIH_HOM_VCV_OPEN_REQ	LV_IGK
N_32	STATE_EFP	TCO	T_AST
FUP_EFP	NC_IDX_DIAG_EFPPWM_PLAUS	NC_IDX_DIAG_L_PRS_SYS	LV_ERR_FUP_EFP
MFF_SP_FUP_CTL	LV_ERR_EFP	LV_ERR_FTL	FUP_EFP_SP
FUP_EFP_MES	NC_IDX_DIAG_FUP_EFP_NOT_PLAUS	LV_FUP_LIH_L_PRS_CTL_REQ	STATE_PWM_VCV
C_ABC_INC_FUP_EFP_NOT_PLAUS	C_ABC_MAX_FUP_EFP_NOT_PLAUS	C_ABC_INC_L_PRS_SYS	C_ABC_MAX_L_PRS_SYS
C_ABC_INC_EFPPWM_PLAUS	C_ABC_MAX_EFPPWM_PLAUS		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CRLC_FUP_EFP_NOT_PLAUS	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlation constant for FUP_EFP_NOT_PLAUS calculation					
C_EFPPWM_I_AD_DIAG_MAX	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Maximum threshold for adaptive I-part of the EFPPWM					
C_EFPPWM_I_AD_DIAG_MIN	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Minimum threshold for adaptive I-part of the EFPPWM					
C_EFPPWM_I_DIAG_MAX	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Maximum threshold for I-part of the EFPPWM					
C_EFPPWM_I_DIAG_MIN	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Minimum threshold for I-part of the EFPPWM					
C_EFPPWM_MIN_AD_DIAG_MAX	1	0... FFFFH	0... 99.9984741211	1.52588e-3	[%]
Maximum threshold for adaptive minimum EFPPWM					
C_EFPPWM_MIN_AD_DIAG_MIN	1	0... FFFFH	0... 99.9984741211	1.52588e-3	[%]
Minimum threshold for adaptive minimum EFPPWM					
C_FTL_MIN_EFPPWM_PLAUS_DIAG	1	0... 7FH	0... 127	1	[l]
Minimum fuel tank level for the EFPPWM plausibility diagnosis					
C_FTL_MIN_L_PRS_SYS_DIAG	1	0... 7FH	0... 127	1	[l]
Minimum fuel tank level for the low pressure system diagnosis					
C_FUP_EFP_BOL_1	1	0... FFFFH	0... 173888	2.6533608	[hPa]
Bottom fuel pressure threshold 1					
C_FUP_EFP_NOT_PLAUS_DEV_MAX	1	0... FFFFH	0... 173888	2.6533608	[hPa]
Maximum FUP EFP deviation for FUP_EFP_NOT_PLAUS error					
C_FUP_EFP_SP_DIF_DIAG_FUP_EFP	1	0... FFFFH	0... 173888	2.6533608	[hPa]
Condition for stable FUP EFP for NOT_PLAUS error					
C_FUP_EFP_TOL_1	1	0... FFFFH	0... 173888	2.6533608	[hPa]
Top fuel pressure threshold 1					
C_FUP_EFP_TOL_2	1	0... FFFFH	0... 173888	2.6533608	[hPa]
Top fuel pressure threshold 2					
C_MFF_SP_MAX_FUP_EFP_NOT_PLAUS	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Maximum injected fuel mass for the FUP_EFP_NOT_PLAUS diagnosis					
C_MFF_SP_MIN_FUP_EFP_NOT_PLAUS	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Minimum injected fuel mass for the FUP_EFP_NOT_PLAUS diagnosis					
C_N_32_FUP_EFP_DIAG	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for EFPWPM plausibility check (Default value 800)					
C_N_32_MAX_EFPPWM_PLAUS	1	0... FFH	0... 8160	32	[rpm]
Maximum engine speed for the plausibility check					
C_N_32_MAX_FUP_EFP_NOT_PLAUS	1	0... FFH	0... 8160	32	[rpm]
Maximum engine speed for the FUP_EFP_NOT_PLAUS diagnosis					
C_N_32_MAX_L_PRS_SYS	1	0... FFH	0... 8160	32	[rpm]
Maximum engine speed for the plausibility check					
C_N_32_MIN_EFPPWM_PLAUS	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for the plausibility check					
C_N_32_MIN_FUP_EFP_NOT_PLAUS	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for the FUP_EFP_NOT_PLAUS diagnosis					
C_N_32_MIN_L_PRS_SYS	1	0... FFH	0... 8160	32	[rpm]
Minimum engine speed for the plausibility check					
C_T_AST_FUP_EFP_DIAG	1	0... FFFFH	0... 6553.5	0.1	[s]
Time for inhibit diagnosis for the low pressure system					
C_T_AST_FUP_EFP_NOT_PLAUS	1	0... FFFFH	0... 6553.5	0.1	[s]

Time after start to allow FUP_EFP_NOT_PLAUS diagnosis					
C_T_EFPPWM_PLAUS_0	1	0... FFH	0... 5.1	0.02	[s]
INI value of delaytime for EFPPWM_PLAUS SYM 0					
C_T_FUP_EFP_NOT_PLAUS_PWM_SWI	1	0... FFFFH	0... 1310.7	0.02	[s]
Time after STATE_PWM_SWI to allow FUP_EFP_NOT_PLAUS diagnosis					
C_TCO_MAX_EFPPWM_PLAUS	1	0... FEH	-48... 142.5	0.75	[°C]
Maximum temperature for the plausibilty check					
C_TCO_MIN_EFPPWM_PLAUS	1	0... FEH	-48... 142.5	0.75	[°C]
Minimum temperature for the plausibility check					
LC_EFPPWM_I_AD_ENA	1	0... 1H	0... 1	1	[-]
Logical constant for enabling of the I-Part adaptive					
LC_EFPPWM_MIN_ENA	1	0... 1H	0... 1	1	[-]
Logical constant for enabling of the minimum EFPPWM					

Import Actions:

ACTION_ERRM_GetLvErr(**IN** <IDX_DIAG>, **OUT** <LV_ERR>)
ACTION_ERRM_FilterSymptom(**IN** <IDX_DIAG>, **IN** <LV_CDN_DIAG>, **IN** <ERR_SYM>, **IN** <ABC_INC>, **IN** <ABC_DEC>, **IN** <ABC_MAX>, **OUT** <LV_ERR>)

General Information

The purpose is to diagnose mechanical errors.
Specific diagnosis information to the project are defined in chapters Diagnosis information (Freeze frame, DTC, error code number, symptom number, data for MIL management ...).

After debounce the correponding error bit LV_ERR... is set to 1 and the corresponding symptom is active. Only if the driving cycle is finished the error bit LV_ERR... and the debounced counter are set to 0 and the corresponding symptom is not active.
The fuel system pressure diagnosis of low pressure will be executed in dependency on the status of the low pressure fuel pressure sensor.

The low-pressure system check indicates mainly an abnormal fuel pressure value in fuel system.

34.27.1 Pointers to ERRM data

Application Conditions

Function description

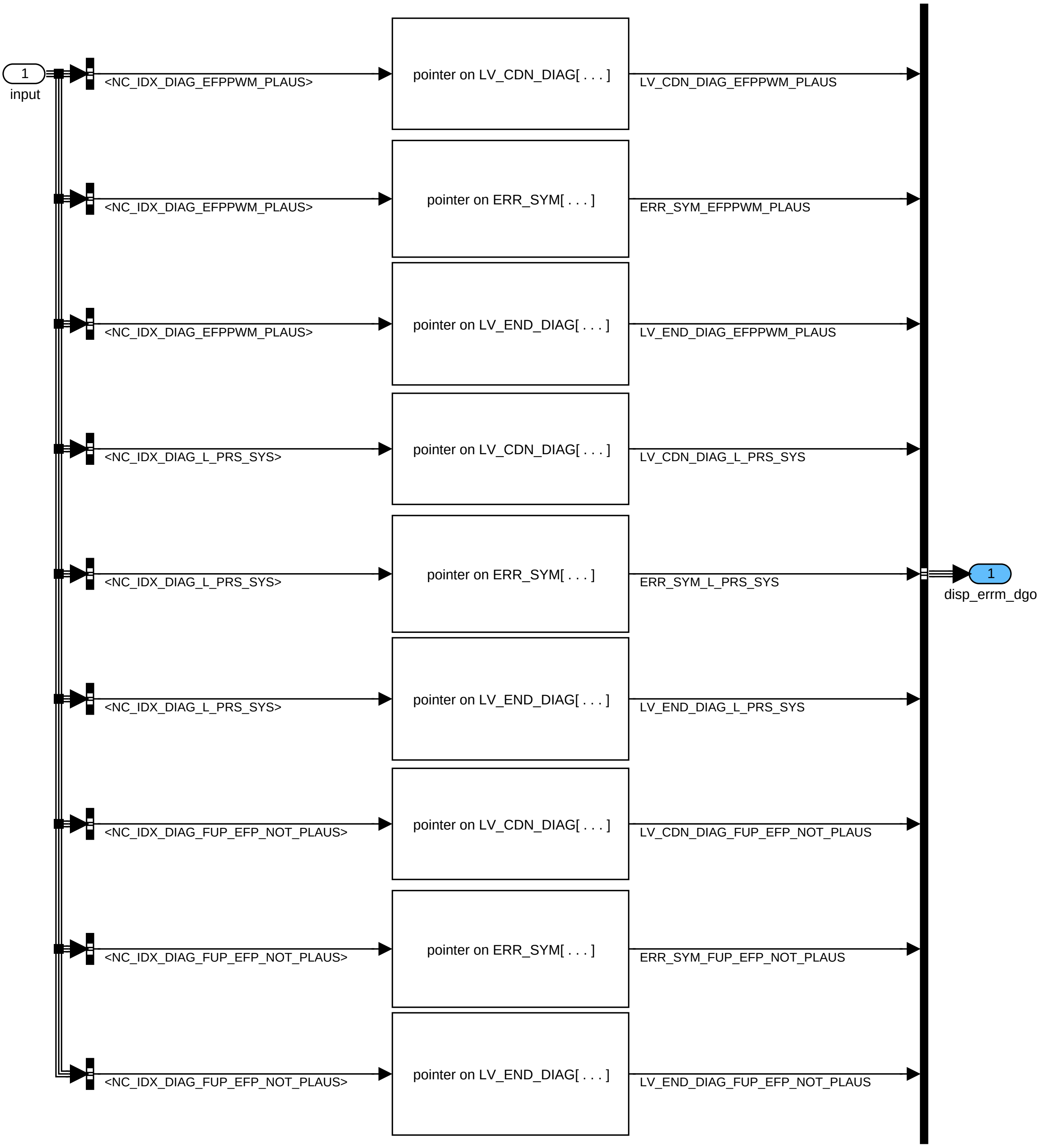


Figure 280:
Path: FUSL_MB046/DISP_ERRM_DGO

34.27.2 FUSL_MB046/FUP_EFP_NOT_PLAUS

CONTENT

Application Conditions

Initialization: RST
Recurrence: 20MS
Activation: always
Deactivation: never

Function description

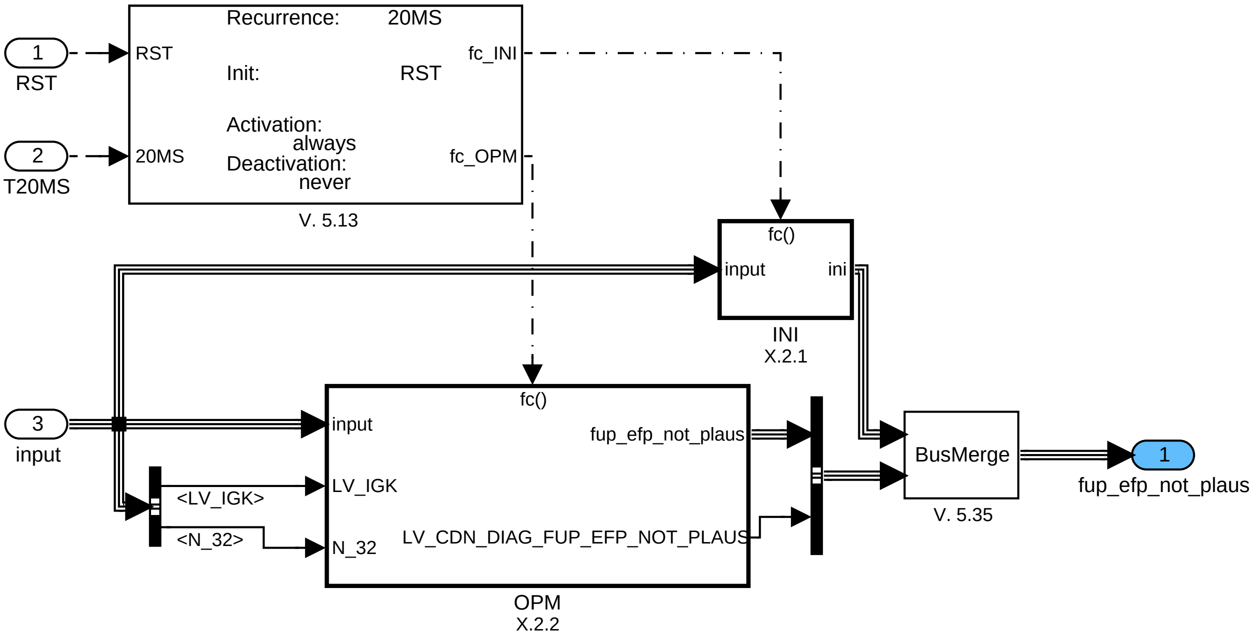


Figure 281:
Path: FUSL_MB046/FUP_EFP_NOT_PLAUS

34.27.2.1 FUSL_MB046/FUP_EFP_NOT_PLAUS/INI

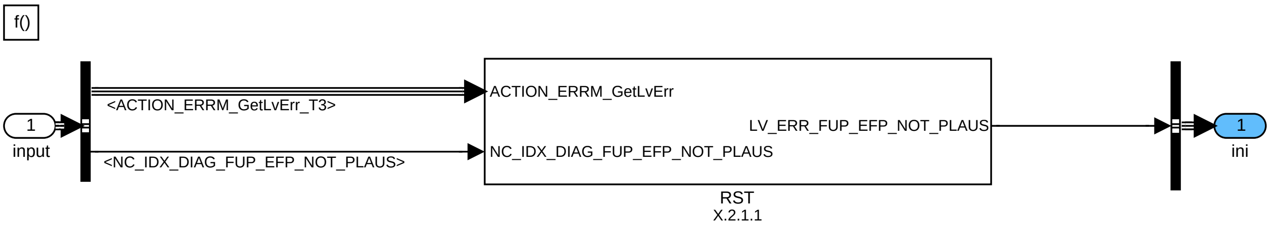


Figure 282:
Path: FUSL_MB046/FUP_EFP_NOT_PLAUS/INI

34.27.2.1.1 FUSL_MB046/FUP_EFP_NOT_PLAUS/INI/RST

CONTENT

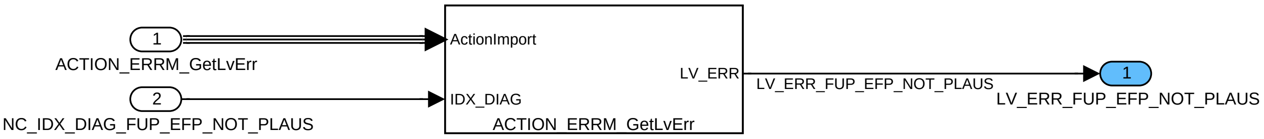


Figure 283:
Path: FUSL_MB046/FUP_EFP_NOT_PLAUS/INI/RST

34.27.2.2 FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM

CONTENT

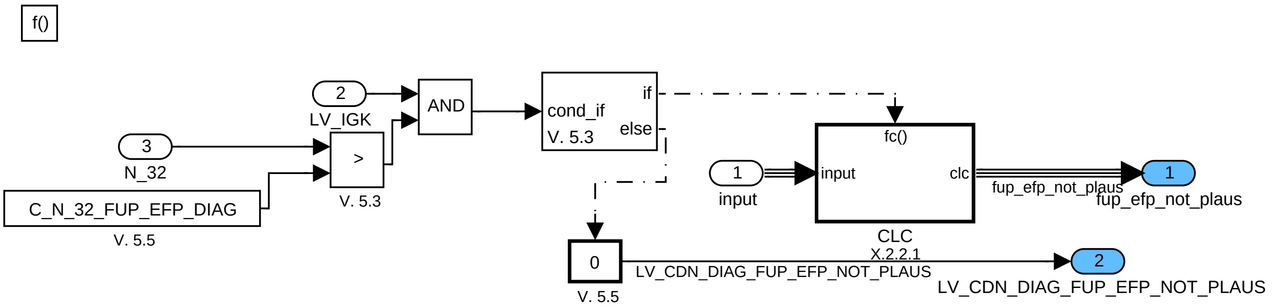


Figure 284:
Path: FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM

34.27.2.2.1 FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM/CLC

CONTENT

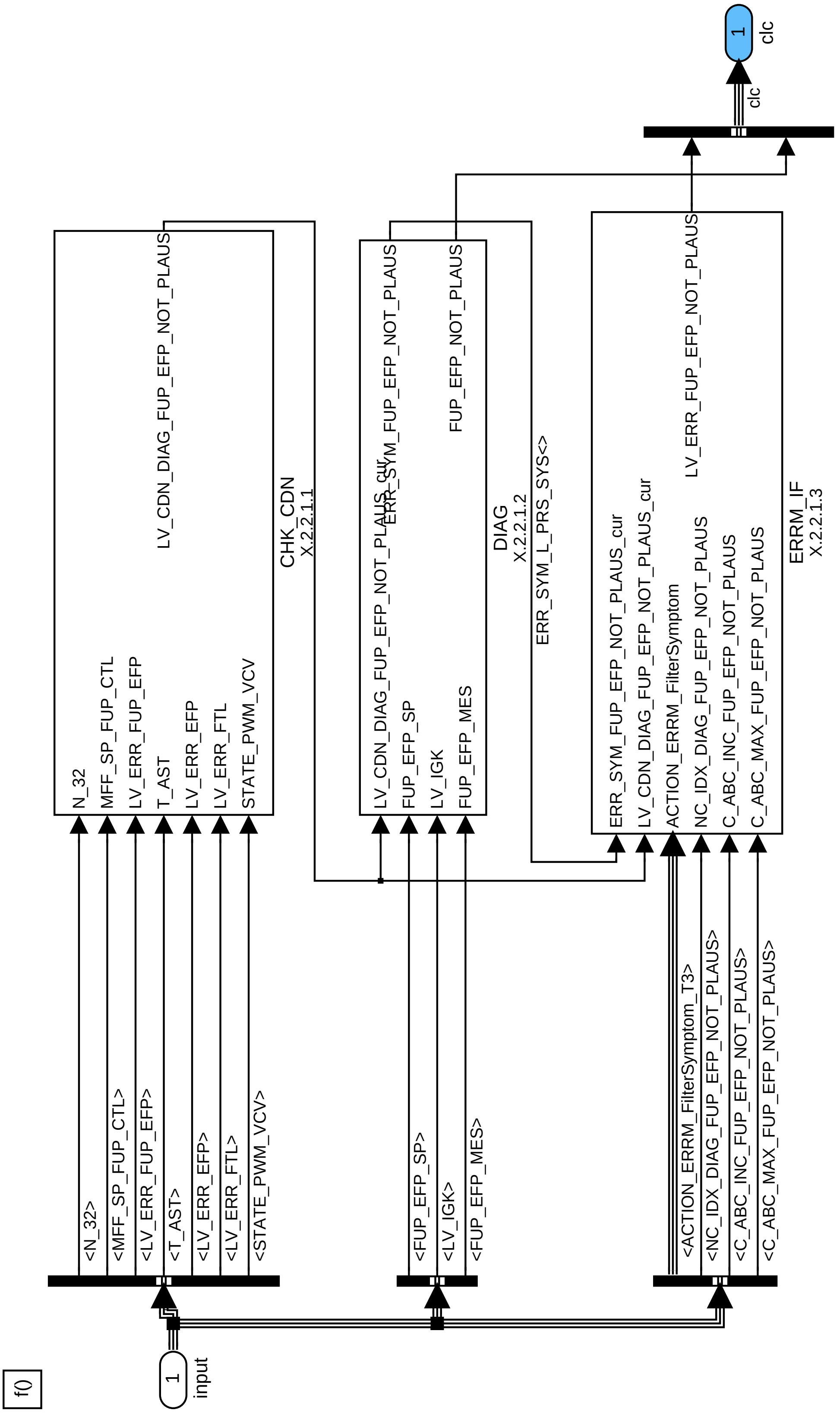


Figure 285:
Path: FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM/CLC

34.27.2.2.1.1 FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM/CLC/CHK_CDN

CONTENT

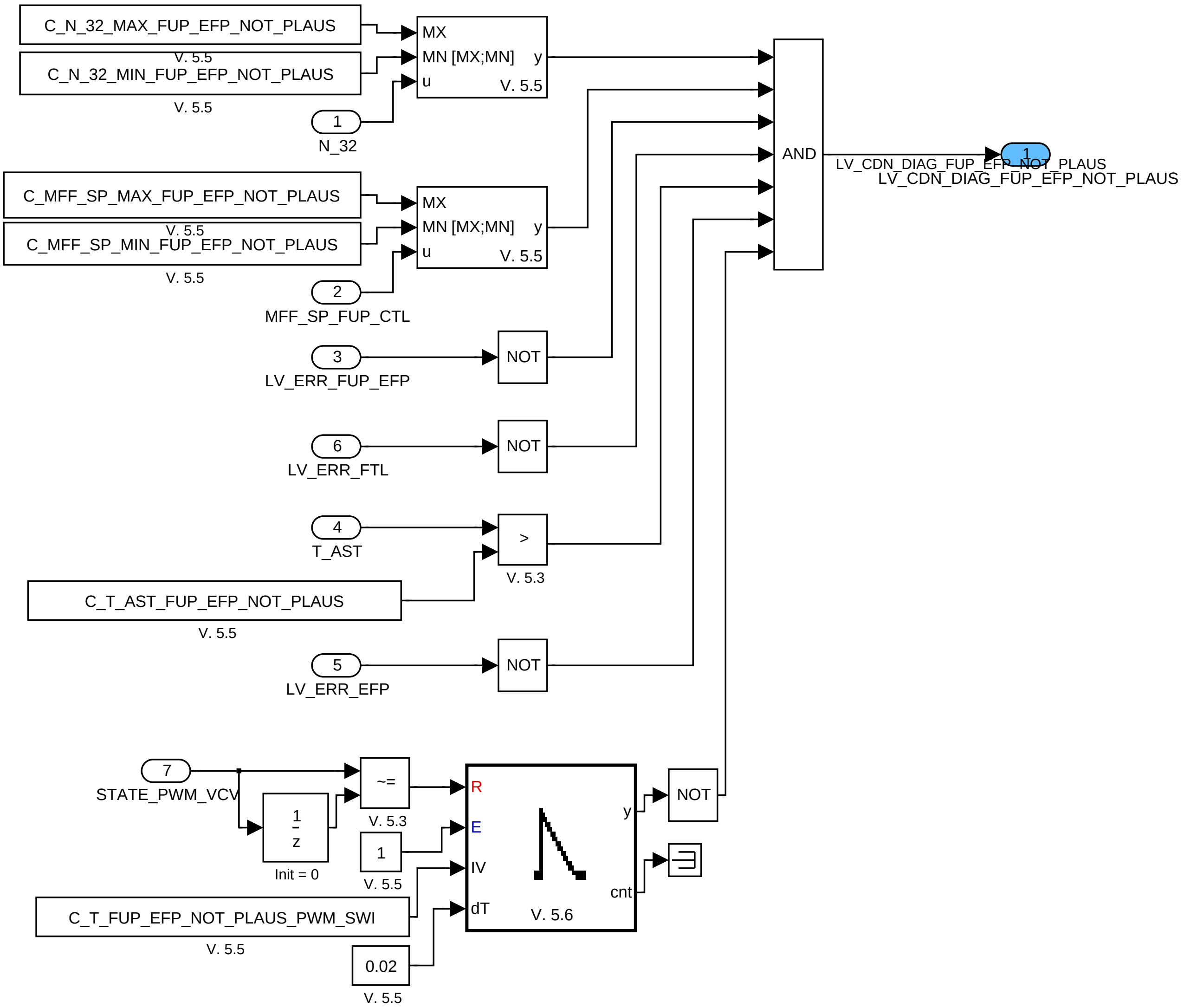


Figure 286:
Path: FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM/CLC/CHK_CDN

34.27.2.2.1.2 FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM/CLC/DIAG

CONTENT

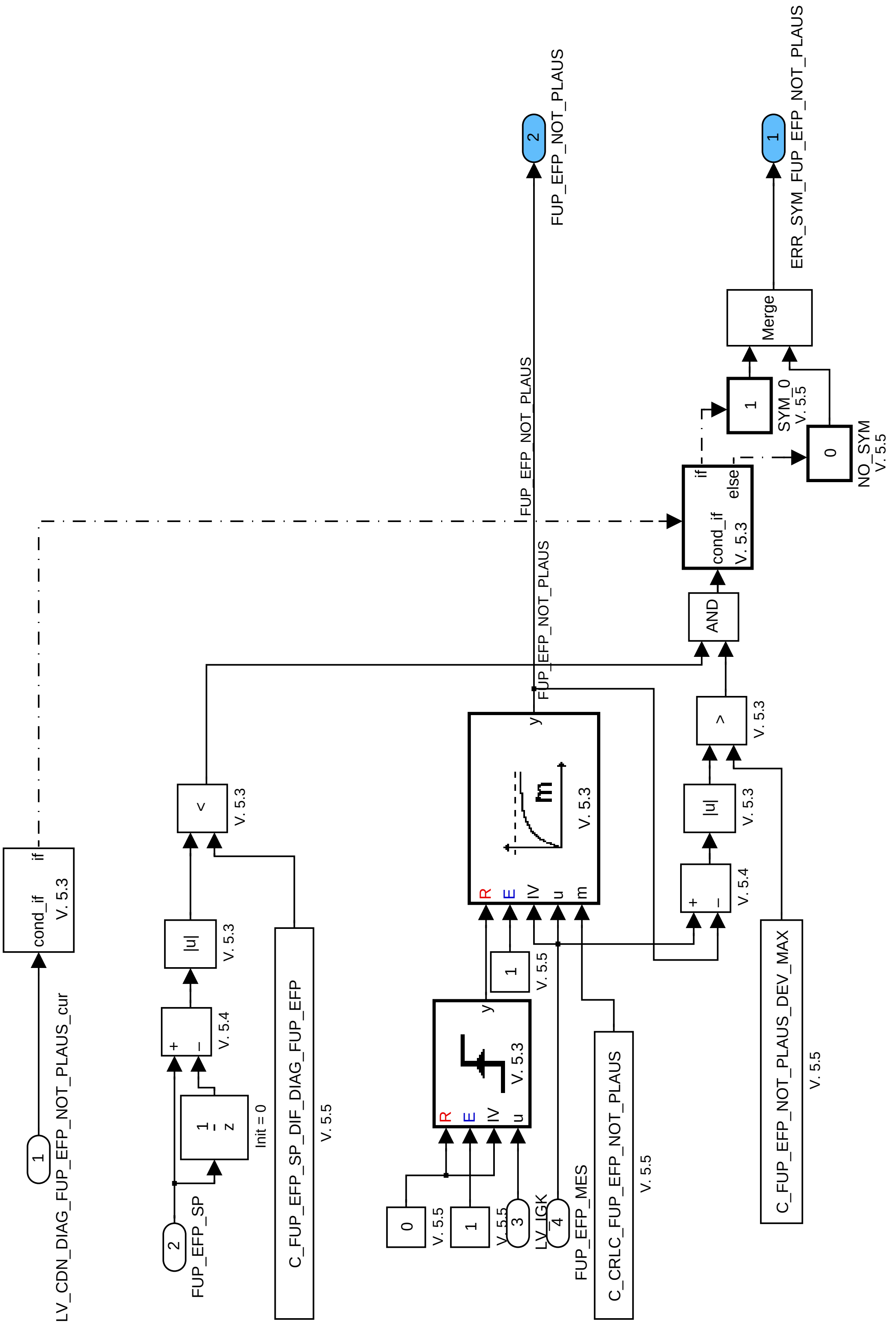


Figure 287:

Path: FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM/CLC/DIAG

34.27.2.2.1.3 FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM/CLC/ERRM_IF

CONTENT

Chapter Fuel supply	Baseline 4DC3940S		Include File 30B04602.00C
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 5873 of 9643

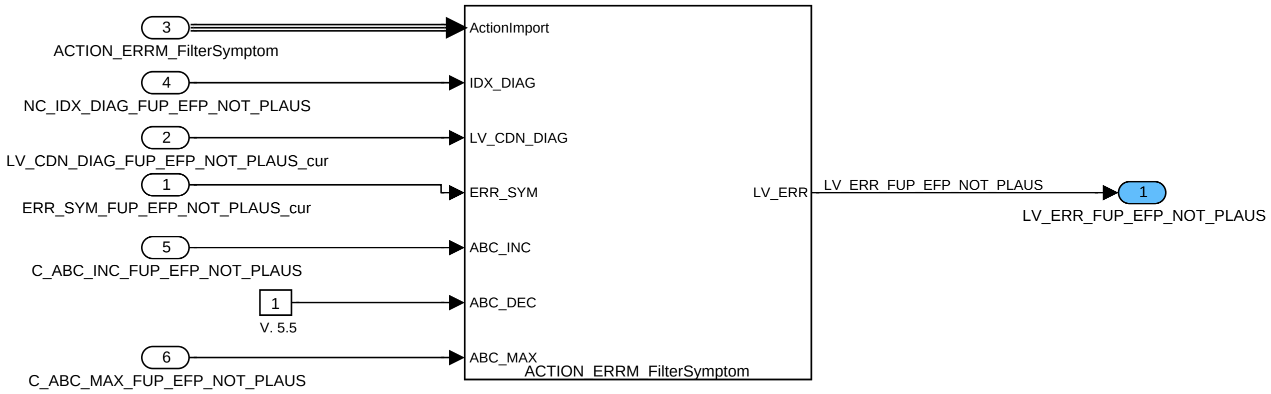


Figure 288:
Path: FUSL_MB046/FUP_EFP_NOT_PLAUS/OPM/CLC/ERRM_IF

34.27.3 LP system monitoring

General information:
Certain pressure thresholds are observed in order to detect mechanical faults of the components of the fuel circuit. There are two pressure limits to detect overpressure and there is one to detect pressure too low.
FUP_EFP > Tol2 indicates an overpressure with an error in the HP pump pressure relief valve. The pressure is higher than the pressure where the relief valve should open.
Tol1 < FUP_EFP < Tol2 indicates a pressure which is under the critical value of the LP pump pressure relief valve. This could happen when there is a mechanical fault, but the overpressure relief valve is still ok
FUP < Bol1 indicates a excessive loss of pressure. Mechanical failure in the pump, regulator or leakage be detected.

Application Conditions

Initialization: RST
Recurrence: 20MS
Activation: always
Deactivation: never

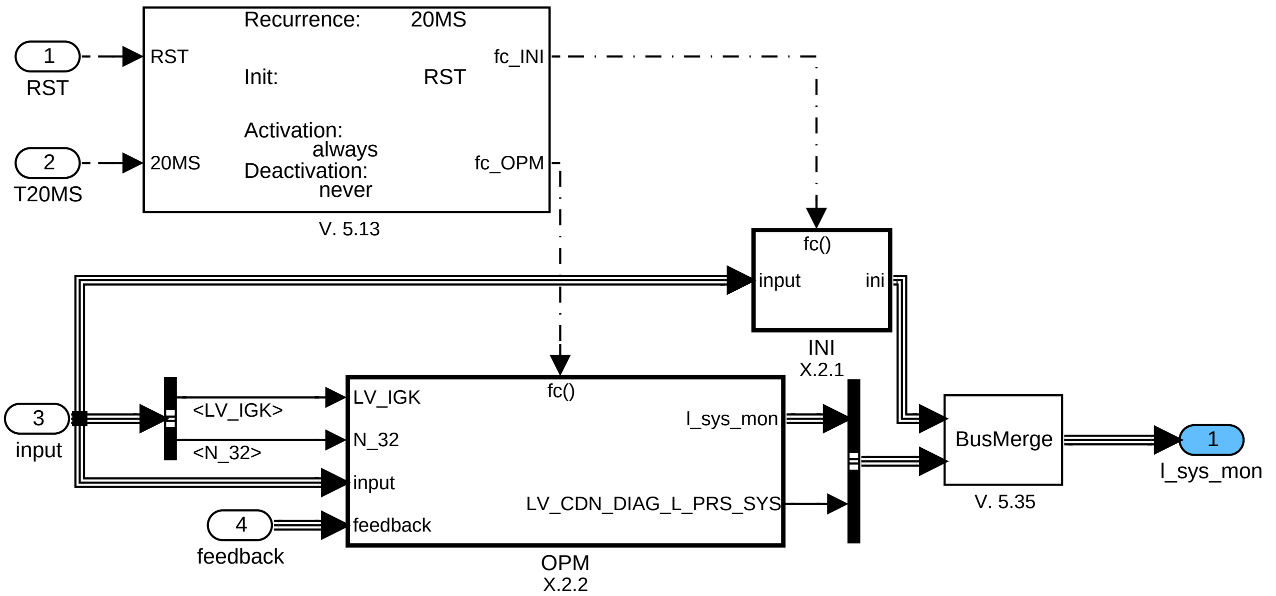


Figure 289:
Path: FUSL_MB046/L_SYS_MON

34.27.3.1 Initialization

All values are initialized according Filter-type.

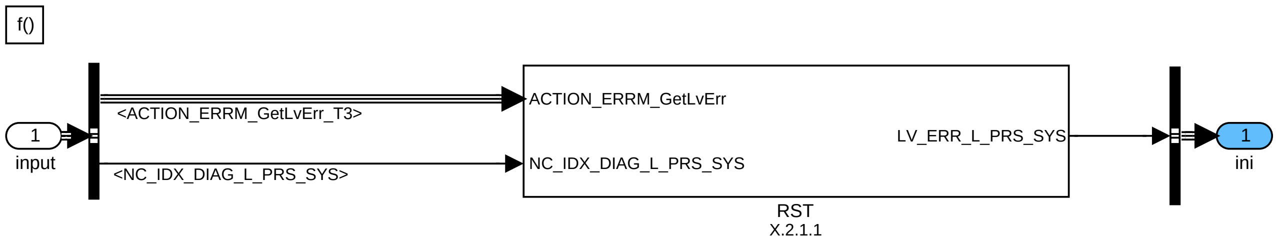


Figure 290:
Path: FUSL_MB046/L_SYS_MON/INI

34.27.3.1.1 Reset

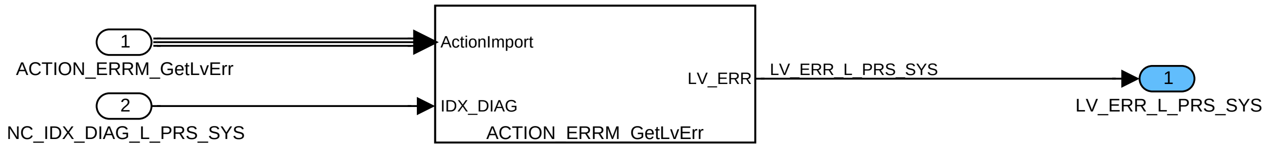


Figure 291:
Path: FUSL_MB046/L_SYS_MON/INI/RST

34.27.3.2 Formula Section

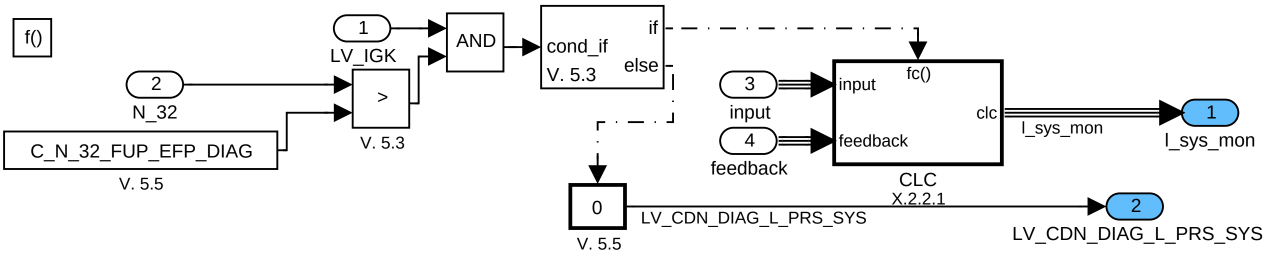


Figure 292:
Path: FUSL_MB046/L_SYS_MON/OPM

34.27.3.2.1 Activation True

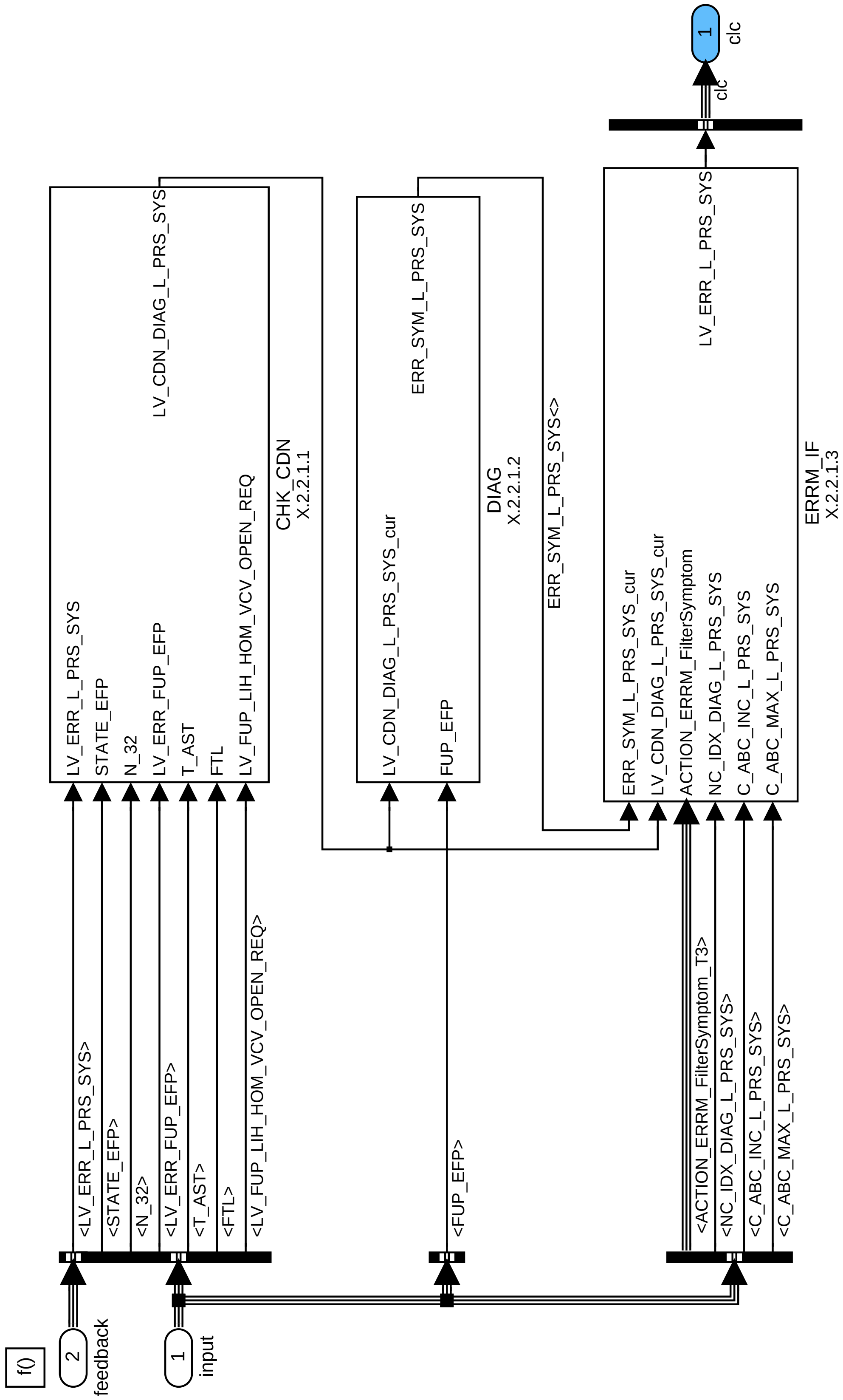


Figure 293:
Path: FUSL_MB046/L_SYS_MON/OPM/CLC

34.27.3.2.1.1 Conditional Check

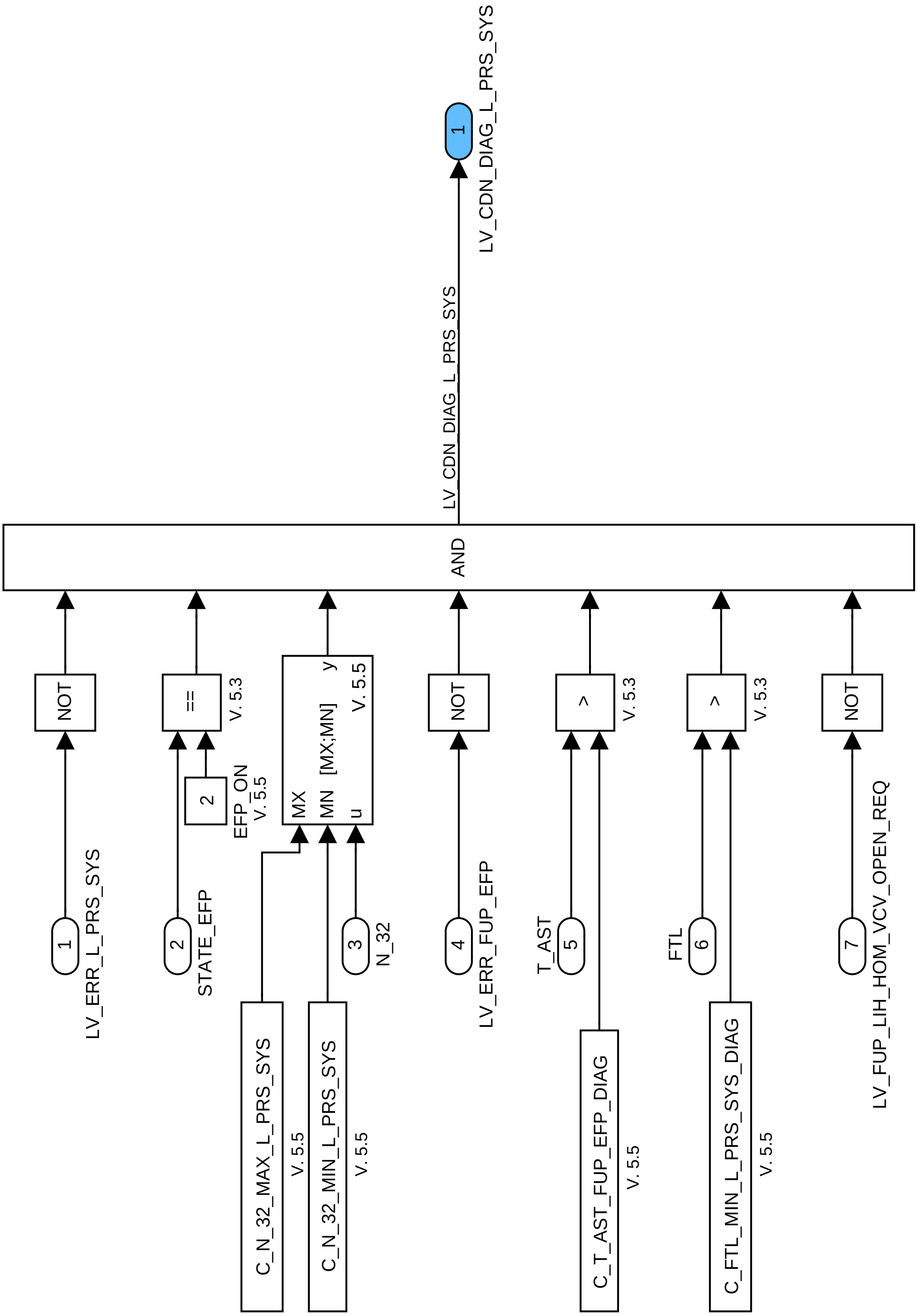


Figure 294:
Path: FUSL_MB046/L_SYS_MON/OPM/CLC/CHK_CDN

34.27.3.2.1.2 Diagnosis

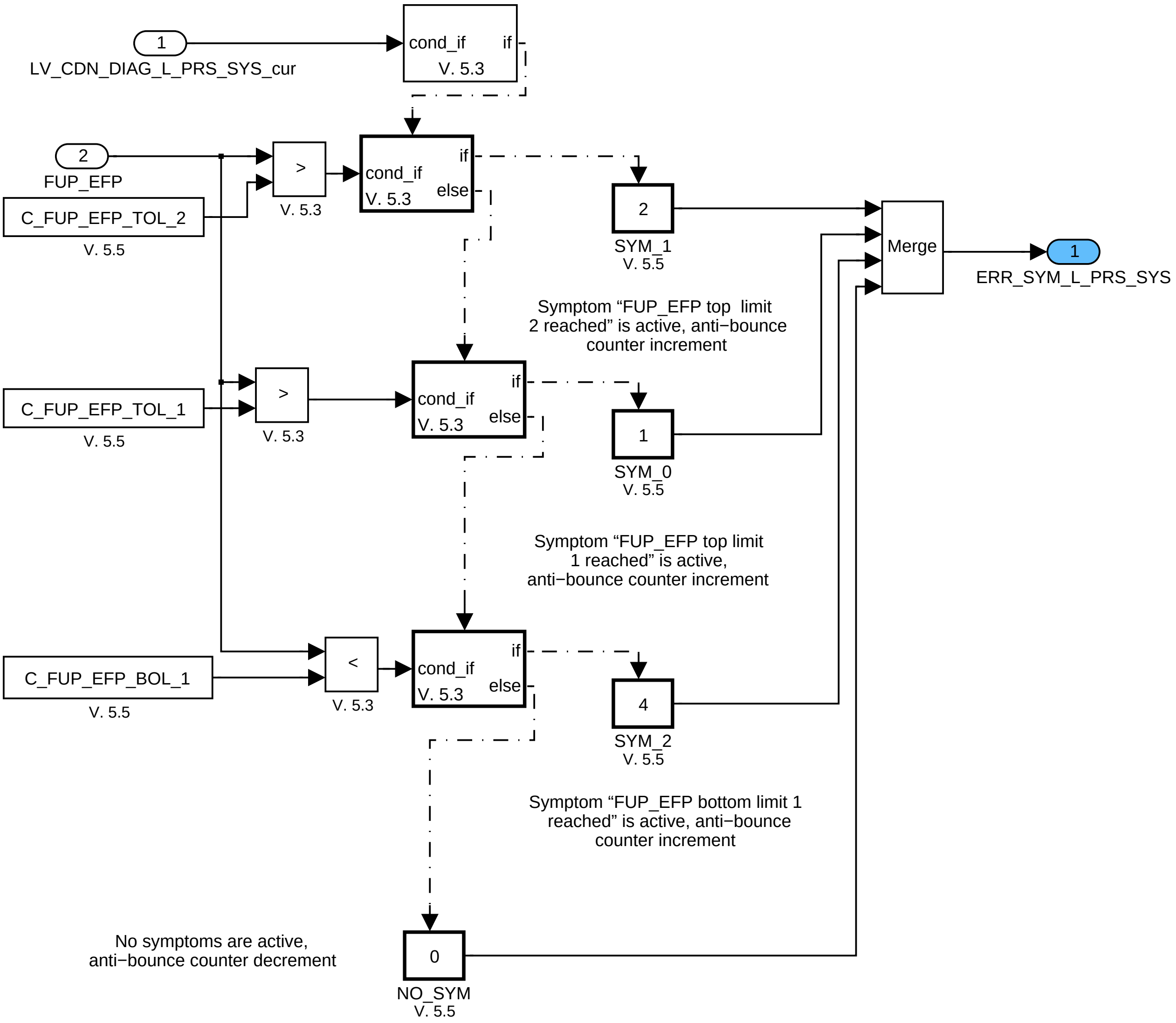


Figure 295:
Path: FUSL_MB046/L_SYS_MON/OPM/CLC/DIAG

34.27.3.2.1.3 ERRM Interface

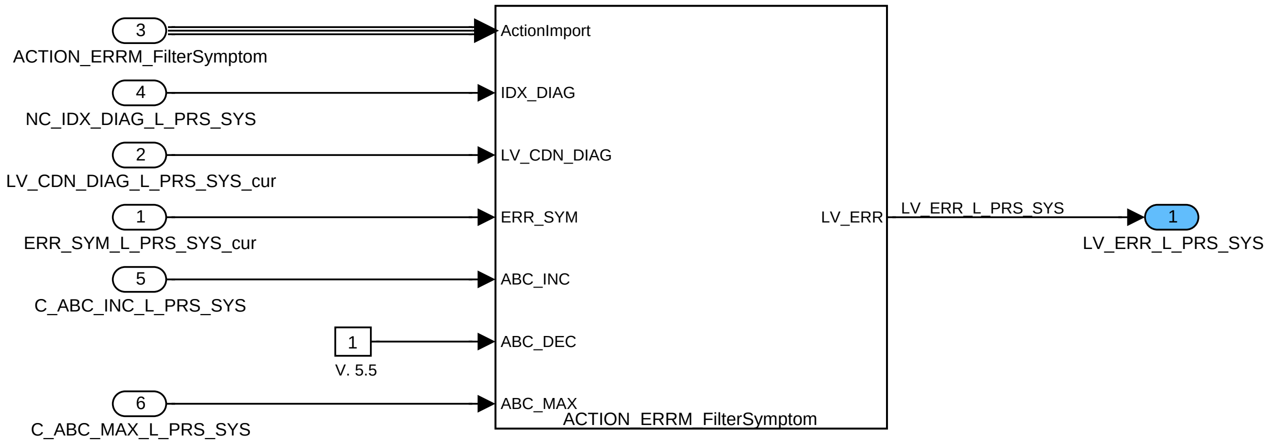


Figure 296:
Path: FUSL_MB046/L_SYS_MON/OPM/CLC/ERRM_IF

34.27.4 Plausibility check EFPPWM

General information:
The plausibility check for the EFPPWM consists out of three parts. First a check of the I-part of the controller, which indicates mainly rapid changes in the performance of the pump or other low pressure system related topics. The adaptive integral part can be used as an indicator for life time topics of the fuel pump. The adaptive minimum EFPPWM gives the functionality of the fuel pump at zero delivery to the high pressure system.

Application Conditions

Initialization: RST
Recurrence: 20MS
Activation: always
Deactivation: never

Function description

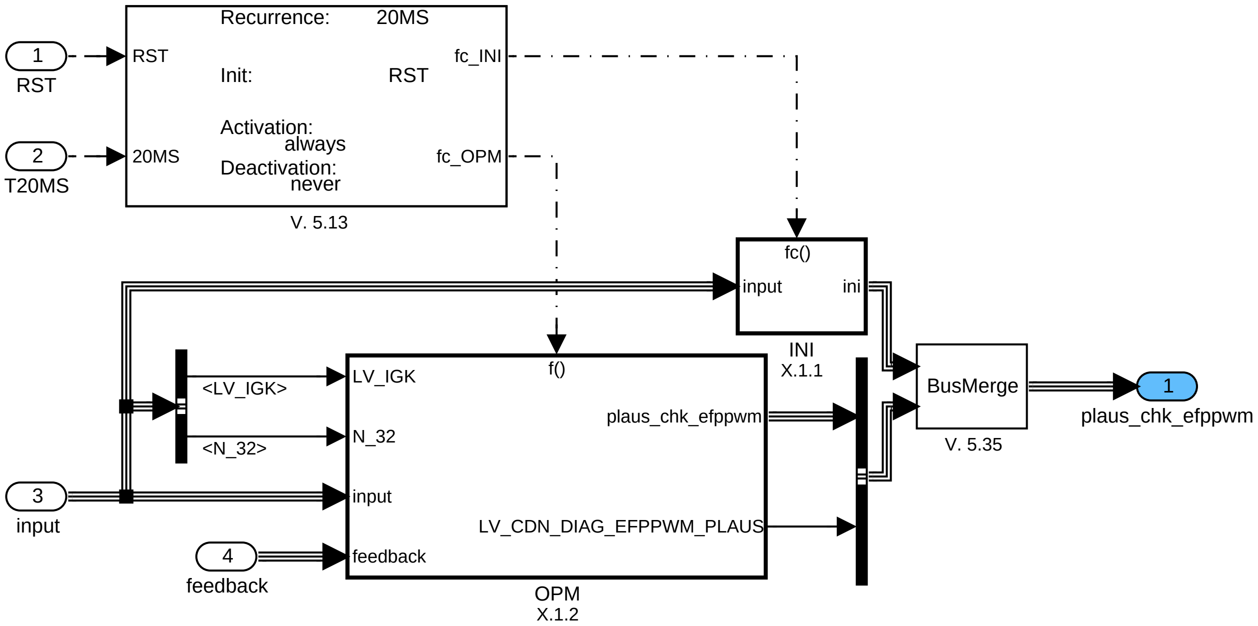


Figure 297:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM

34.27.4.1 Initialization

All values are initialized according Filter-type.

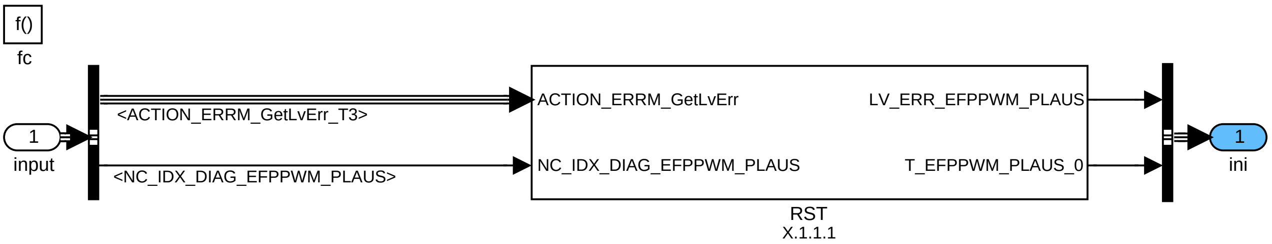


Figure 298:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM/INI

34.27.4.1.1 Reset

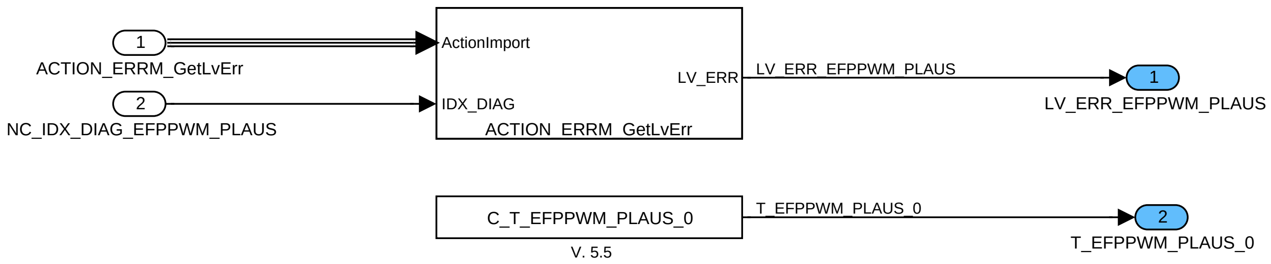


Figure 299:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM/INI/RST

34.27.4.2 Formula Section

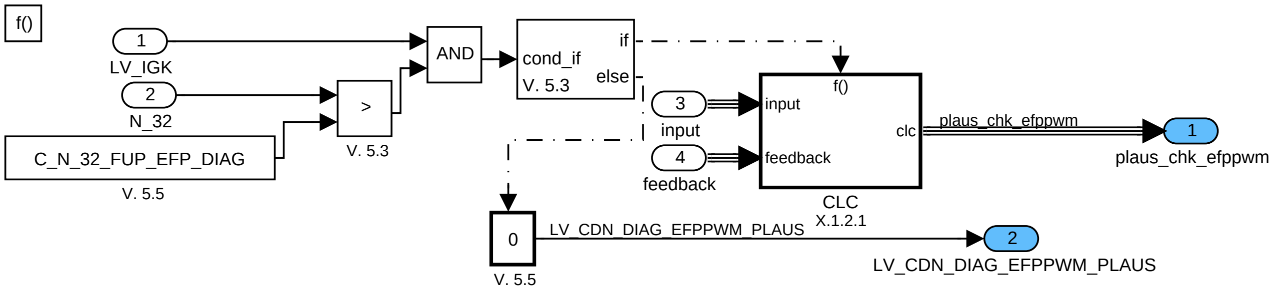


Figure 300:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM/OPM

34.27.4.2.1 Activation True

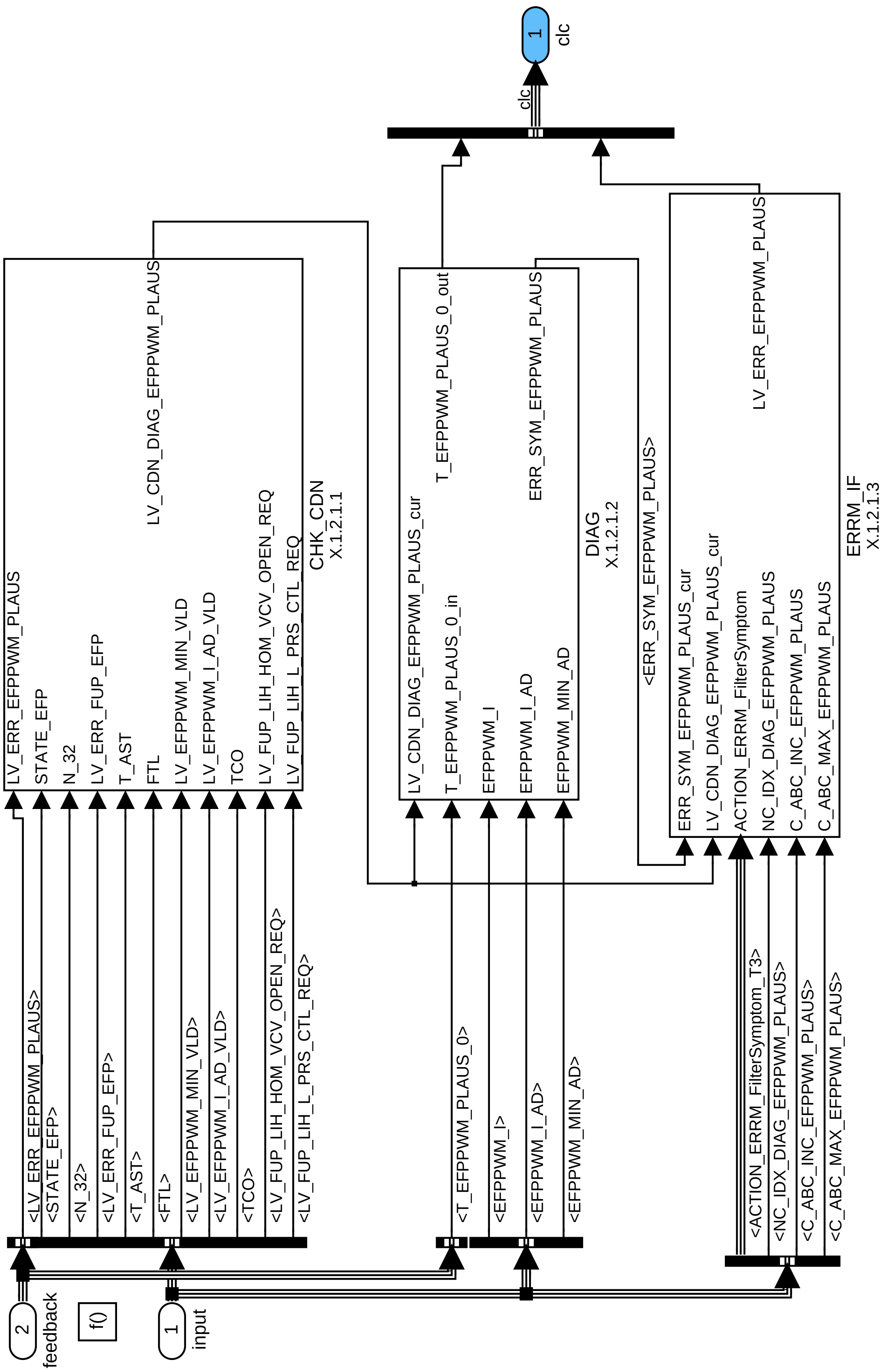


Figure 301:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM/OPM/CLC

34.27.4.2.1.1 Conditional Check

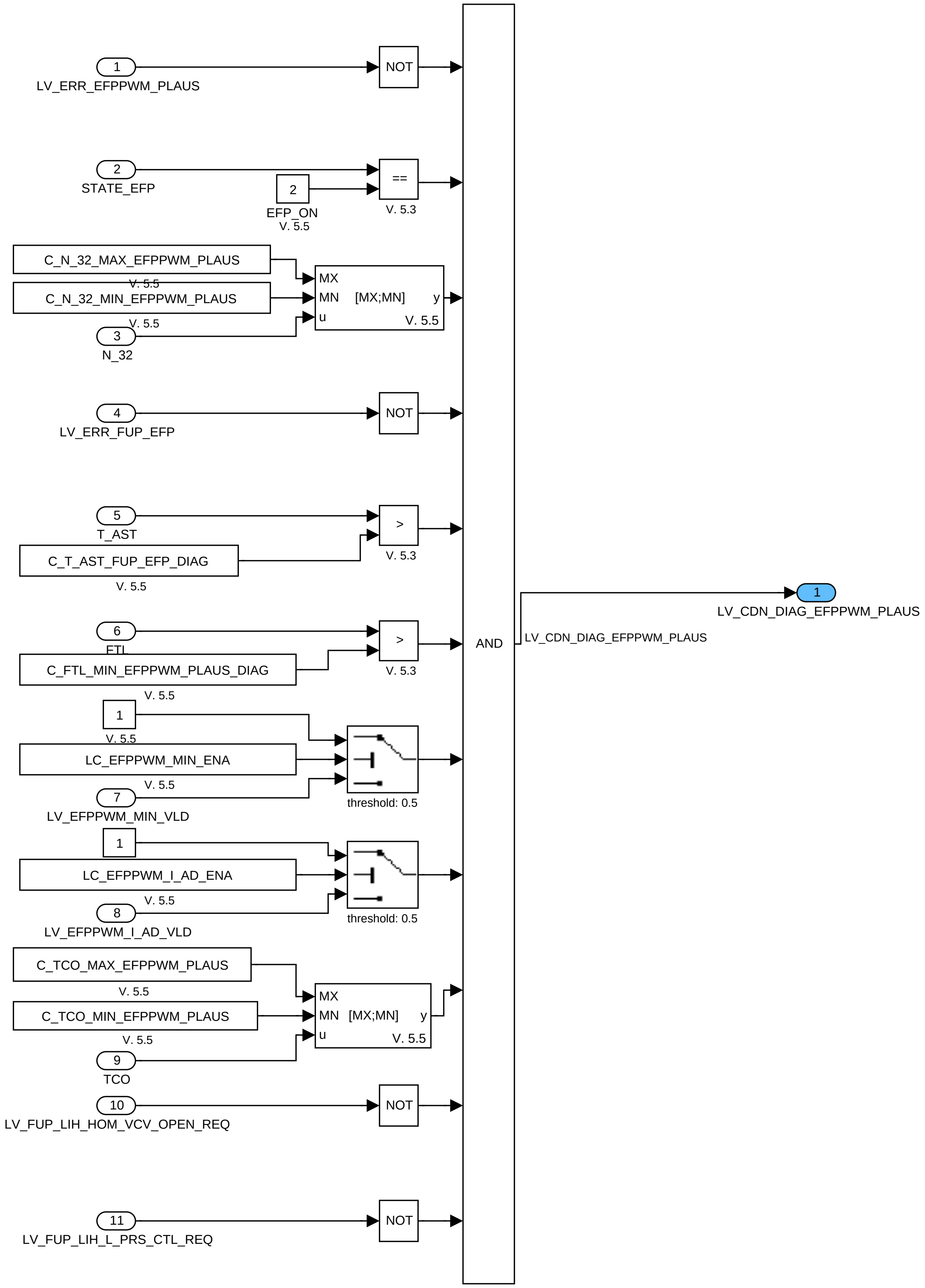


Figure 302:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM/OPM/CLC/CHK_CDN

34.27.4.2.1.2 Diagnosis

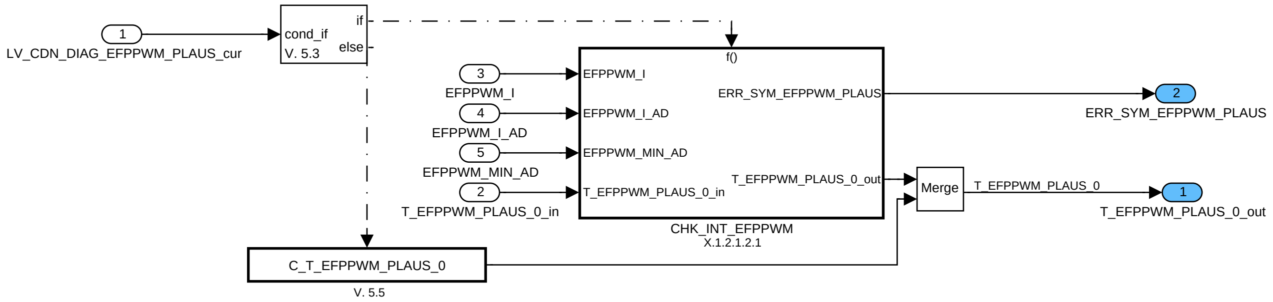


Figure 303:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM/OPM/CLC/DIAG

34.27.4.2.1.2.1 Check the integral part of EFPPWM

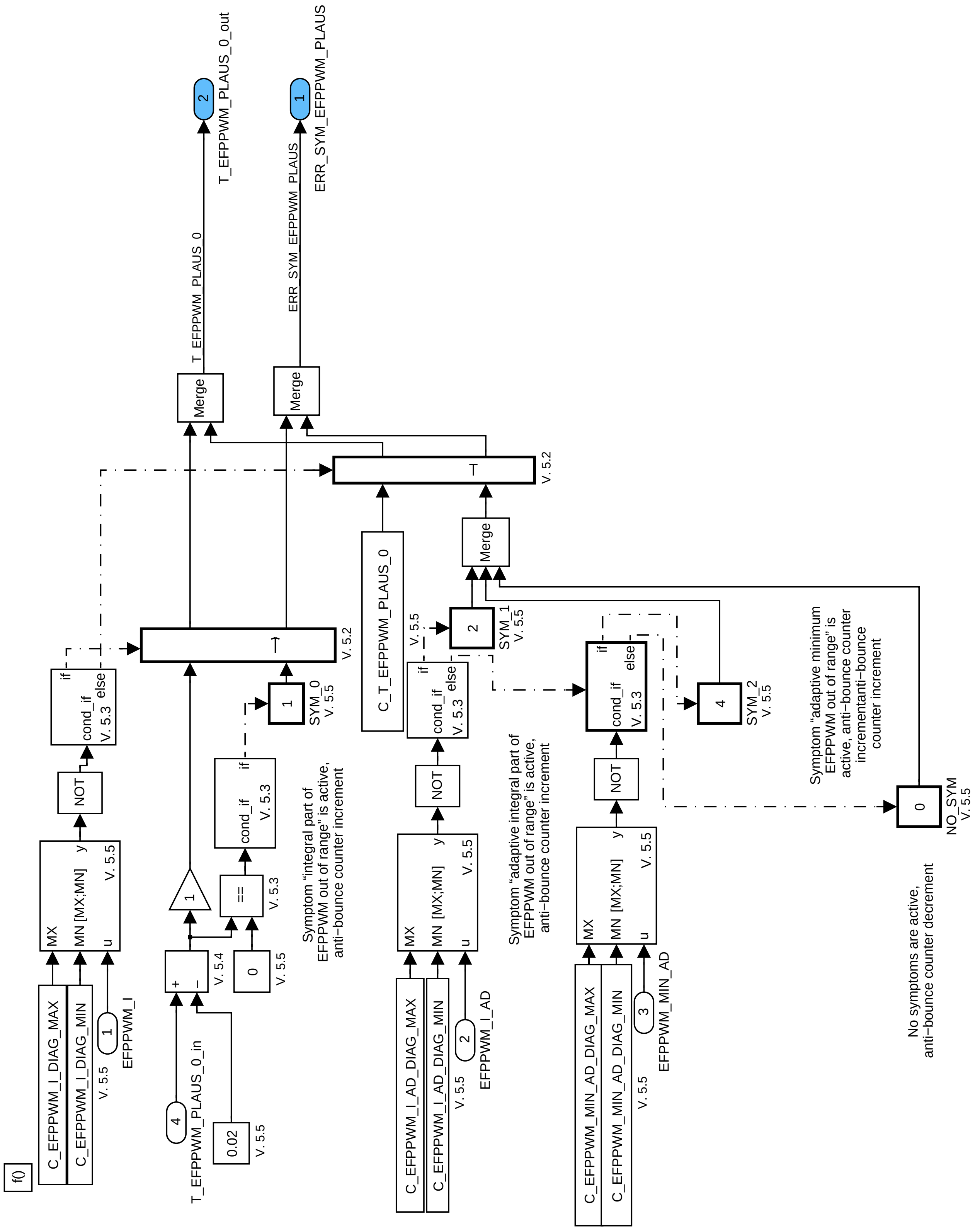


Figure 304:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM/OPM/CLC/DIAG/CHK_INT_EFPPWM

34.27.4.2.1.3 ERRM Interface

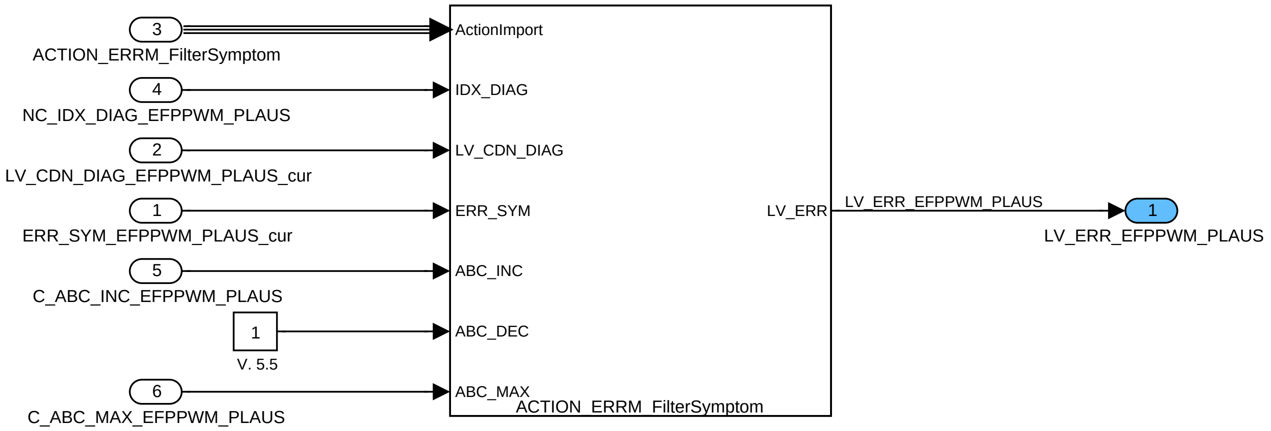


Figure 305:
Path: FUSL_MB046/PLAUS_CHK_EFPPWM/OPM/CLC/ERRM_IF

34.28 Fuel system pressure diagnosis for low pressure (Appl. Inc.)

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ABC_INC_EFPPWM_PLAUS	V	0... FFH	0... 255	1	[-]
Increment of the plausibility check of EFPPWM anti-bounce counter					
C_ABC_INC_FUP_EFP_NOT_PLAUS	V	0... FFH	0... 255	1	[-]
Increment of the loose contact check of FUP EFP anti-bounce counter					
C_ABC_INC_L_PRS_SYS	V	0... FFH	0... 255	1	[-]
Increment of low pressure system monitoring anti-bounce counter					
C_ABC_MAX_EFPPWM_PLAUS	V	1... FFH	1... 255	1	[-]
Threshold to be reached, before plausibility check error of EFPPWM					
C_ABC_MAX_FUP_EFP_NOT_PLAUS	V	1... FFH	1... 255	1	[-]
Threshold to be reached, before FUP_EFP loose contact					
C_ABC_MAX_L_PRS_SYS	V	1... FFH	1... 255	1	[-]
Threshold to be reached, before permanently activating low pressure system monitoring error					

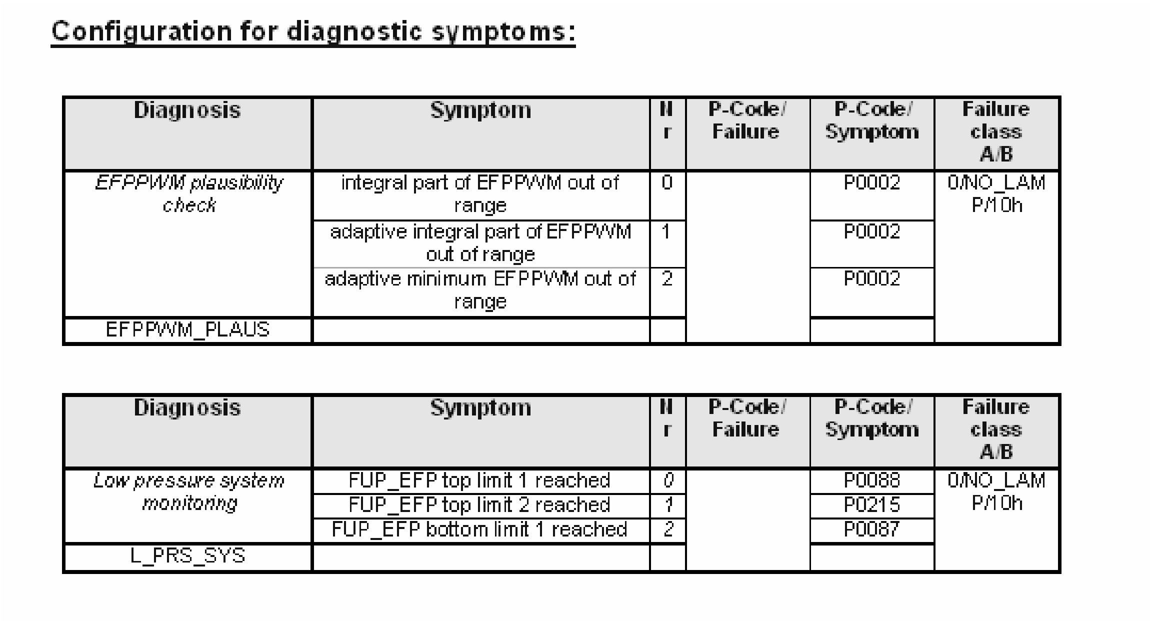
General Information

The low pressure system diagnosis is not inhibited.

Error treatment:
Error debounce:
Debounce counter increment: C_ABC_INC_L_PRS_SYS
Debounce counter maximum value: C_ABC_MAX_L_PRS_SYS

Debounce counter increment: C_ABC_INC_EFPPWM_PLAUS
Debounce counter maximum value: C_ABC_MAX_EFPPWM_PLAUS

- Debounce counter increment: C_ABC_INC_FUP_EFP_NOT_PLAUS
- Debounce counter maximum value: C_ABC_MAX_FUP_EFP_NOT_PLAUS



Configuration for diagnostic symptoms:

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
EFPPWM plausibility check	integral part of EFPPWM out of range	0		P0002	0/NO_LAM P/10h
	adaptive integral part of EFPPWM out of range	1		P0002	
	adaptive minimum EFPPWM out of range	2		P0002	
EFPPWM_PLAUS					

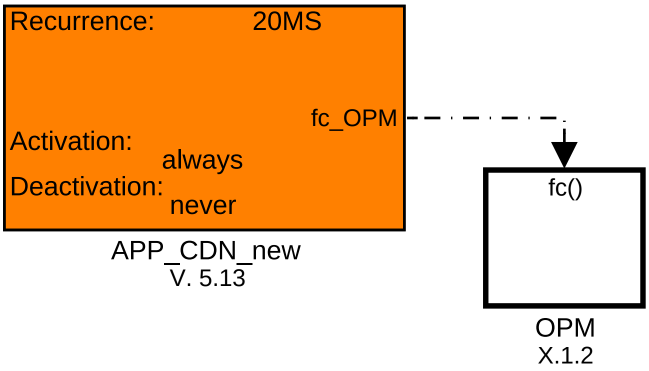
Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
Low pressure system monitoring	FUP_EFP top limit 1 reached	*0*		P0088	0/NO_LAM P/10h
	FUP_EFP top limit 2 reached	*1*		P0215	
	FUP_EFP bottom limit 1 reached	*2*		P0087	
L_PRS_SYS					

Diagnosis	Symptom	Nr	P-Code / Failure	P-Code/ Symptom	Failure class A/B
FUP EFP loose contact	Low pressure sensor loose contact	0		P0000	STD_INI
FUP_EFP_NOT_PLAUS					

Application Conditions

Initialization: all outputs to zero at RST
Recurrence: 20MS
Activation: 1
Deactivation: 0

Function description



SDA_SRS / SDÂ V 5.0.4 24−Mar−2006

Figure 306:
Path: FUSL_MB073

34.28.1 CALCULATION

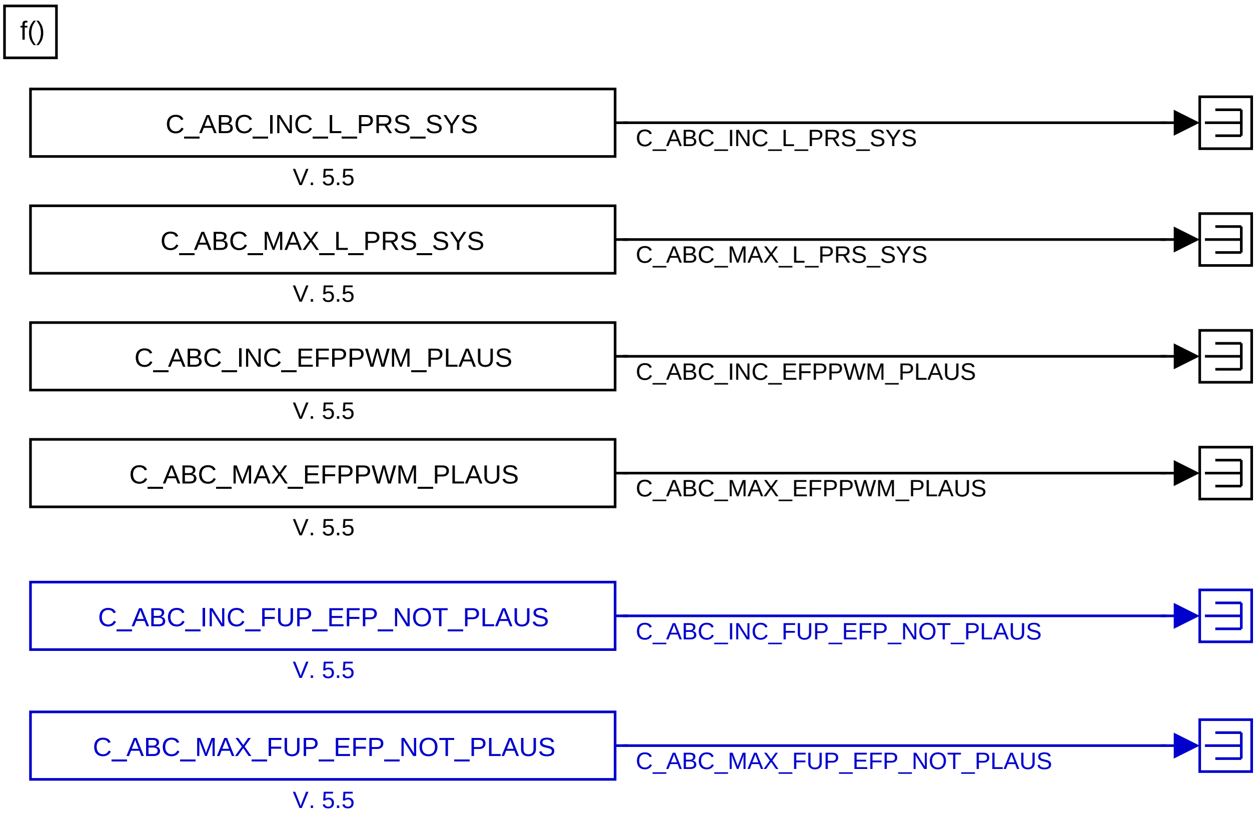


Figure 307:
Path: FUSL_MB073/OPM

34.28.2 Outputs for BMW which are defined as FUSL exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_apraiLhom	V/O	0...1H	0...1	1	[-]
Anforderung Diagnose-Raildruck Homogen					
B_aprail_ndnotl	V/O	0...1H	0...1	1	[-]
Anforderung Diagnose-Raildruck Niederdrucknotlauf					
Prail_ist	V/O	0...FFFFH	0...40	0,61e-3	MPa
Rail pressure					
B_prail_h	V/O	0...1H	0...1	1	-
Request homogen					
B_prailvcv_h	V/O	0...1H	0...1	1	-
Request homogen VCV open					
B_prailvfd_h	V/O	0...1H	0...1	1	-
Request LIH pre pressure					

Input data:

LV_FUP_HOM_REQ	LV_FUP_LIH_REQ	LV_FUP_LIH_HOM_REQ
LV_FUP_LIH_HOM_VCV_OPEN_REQ	LV_FUP_LIH_L_PRS_CTL_REQ	FUP

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: 0

Activation: all engine operating states

Recurrences: 10 ms

Formula section:

B_aprail_hom = LV_FUP_HOM_REQ

B_aprail_ndnotl = LV_FUP_LIH_REQ

B_prail_h = LV_FUP_LIH_HOM_REQ

B_prailvcv_h = LV_FUP_LIH_HOM_VCV_OPEN_REQ

B_prailvfd_h = LV_FUP_LIH_L_PRS_CTL_REQ

Prail_ist = FUP

34.28.3 Outputs for SV aggregates, BMW→SV

Output data:

Name	Mode	Hex	Phys	Res.	Unit
FUP_SP_EXT	V/O	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel pressure setpoint BMW					
FUP_H_SP_S_EXT	V/O	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel pressure setpoint for stratified mode, high range					
LV_FUP_SP_EXT_REQ	V/O	0...1H	0...1	1	[-]
External request of fuel pressure setpoint (high pressure)					

Input data:

Prail_out_s	Prail_soll	B_prailsw	LV_FUP_SP_REQ_EXT_ADJ
FUP_SP_EXT_ADJ			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes.

Application conditions:

Initialisation:	at resat: all = 0hex
Recurrence :	10 ms
Activation:	every engine state In power latch phase: LV_FUP_SP_EXT_REQ = 0

Formula section:

Remark: all formulas are valid in a **physical** meaning
Endif

```
FUP_H_SP_S_EXT = Prail_out_s
If    LV_FUP_SP_REQ_EXT_ADJ = 0
then
   FUP_SP_EXT = Prail_soll
else
     FUP_SP_EXT  =  FUP_SP_EXT_ADJ
LV_FUP_SP_EXT_REQ =   B_prailsw
```

35 Fuel tank level

Table of Contents

Index

35.1 Fuel reserve diagnosis for temperature protection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_FTL_MIN_ACT	V	0...1H	0...1	1	[-]
Activation conditions for Diagnosis passed					
LV_ERR_FTL_MIN	V/O	0...1H	0...1	1	[-]
Fuel tank level low Injection intervention					
RATIO_FTL	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Ratio of counter					
RATIO_FTL_INI	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Ratio of initialisation counter					
LV_CDN_DIAG_FTL_MIN	V	0...1H	0...1	1	[-]
Condition diag					
LV_END_DIAG_FTL_MIN	V	0...1H	0...1	1	[-]
End of diagnosis					
CTR_FTL_MIN_ACT	V	0...FFFFH	0...65535	1	[-]
Pre counter FTL events					
ER_FTL_MIN	V	8000...7FFFH	-32768...32767	1	[µs]
Engine roughness for FTL_MIN diagnosis					
CTR_ABC_FTL_MIN	V	0...FFH	0...255	1	[-]
Debounce counter					
ERR_SYM_FTL_MIN	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptome					
INH_IV_FTL_MIN	V/O	0...FFH	0...255	1	[-]
Bit coded inhbit condition cylinder selective for fuel tank low level					
LV_FUP_AVL_DIAG_TMP	V	0...1H	0...1	1	[-]
Logical variable indicating fuel pressure falls below a threshold of the FTL diagnosis					
LV_LAMB_PLAUS_FTL_MIN_DIAG_TMP	V	0...1H	0...1	1	[-]
Logical variable indicating an unplausible lambda signal for FTL diagnosis					
LV_ERR_FUP_DIAG_ACT_HPRS	V	0...1H	0...1	1	[-]
Logical variable indicating the use of error flags for diagnosis condition in High Pressure System					
LV_ERR_FUP_DIAG_ACT_LPRS	V	0...1H	0...1	1	[-]
Logical variable indicating the use of error flags for diagnosis condition in Low Pressure System					
STATE_ERR_FTL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
State Defining the Source of the Error symptom					
FTL_RI_MMV_H_RES	V	0...7F00H	0...127	3.9063e-3	[l]
Moving mean value of FTL_RI with high resolution					
FTL_LE_MMV_H_RES	V	0...7F00H	0...127	3.9063e-3	[l]
Moving mean value of FTL_LE with high resolution					
FTL_RI_MMV	V	0...7FH	0...127	1	[l]
Moving mean value of FTL_RI					
FTL_LE_MMV	V	0...7FH	0...127	1	[l]
Moving mean value of FTL_LE					
LV_FTL_LE_RE_ACT	V	0...1H	0...1	1	[-]
Flag indicating left and right side of tank below theshold					
T_DIAG_FTL_MIN	V	0...FFH	0...25.5	0.1	[s]
Timer to stop ERR_SYM_FTL_MIN = SYM_0 diagnosis and reset inhibit flag					

Input data:

LV_ERR_SAP	LV_ES	OPM_AV	LV_ERR_FUP_MFP_PLAUS
LV_ERR_MAF	LV_FTL_CAN_ERR	LV_ERR_SAP	LV_ERR_FTL_OBD
LV_ERR_LOAD_TPS_PLAUS	LV_LAM_STOP[NC_CBK_EX_NR]	FAC_LAM_LIM[NC_CBK_EX_NR]	LV_LAMB_LS_UP_VLD[NC_CBK_EX_NR]
LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_PUC	TEG_DYN	LV_ERR_T_SEG_ER
LAMB_SP_MV	T_AST	LV_ERR_LAM_ADJ[NC_CBK_EX_NR]	FUP_SP
LV_ERR_CHG_LS_UP	LV_ERR_EL_CPS	LV_ERR_DIAGCPS	LV_ERR_FSD[NC_CBK_EX_NR]
LAMB_LS_UP[NC_CBK_EX_NR]	LV_ERR_SA_SAV	LV_ERR_SA_SAV_LSL	LV_ERR_SAV
LV_LAM_LSCL[NC_CBK_EX_NR]	LV_ERR_SEG_AD_ER	LV_ERR_AMP	LV_ERR_AMP_PLAUS
FTL_RI	FTL_LE	FUP	FUP_EFP
LAMB_SP_FIL_HOM[NC_CBK_EX_NR]	LV_ERR_FUP_EFP	LV_ERR_FUP	LV_ERR_SLV_IVVT_IN[NC_NR_CBK_IVVT]
LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_MAP_TPS_PLAUS	LV_FUP_LIH_HOM_REQ
LV_FUP_LIH_L_PRS_CTL_REQ	LV_FUP_LIH_HOM_VCV_OPEN_REQ	ER	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_RATIO_FTL_INI_MIN	1	0...FFFFH	0...655.35	0.01	[-]
Correction factor					
C_FAC_RATIO_FTL_INI_MAX	1	0...FFFFH	0...655.35	0.01	[-]
Correction factor					
C_CTR_FTL_MAX	1	0...FFFFH	0...65535	1	[-]
Maximum of increments					
C_ABC_INC_FTL_MIN	1	0...FFH	0...255	1	[-]
Increment					
C_ABC_DEC_FTL_MIN	1	0...FFH	0...255	1	[-]
Decrement					
C_ABC_MAX_FTL_MIN	1	1...FFH	1...255	1	[-]
Maximum					
C_FUP_EFP_DIAG_FTL_MIN	1	0...FFFFH	0...173888	2.6533608	[hPa]
Fuel pressure of low pressure threshold for FTL diagnosis					
LC_LAMB_PLAUS_FTL_MIN_DIAG	1	0...1H	0...1	1	[-]
Logical constant indicating the non plausible lambda signal for FTL diagnosis					
C_FAC_LAM_FTL_MIN	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Threshold of limited value of lambda controller output for FTL diagnosis					
LC_ERR_FUP_DIAG_ACT_HPRS	1	0...1H	0...1	1	[-]
Logical constant indicating the use of error flags for diagnosis condition at High Pressure					
LC_ERR_FUP_DIAG_ACT_LPRS	1	0...1H	0...1	1	[-]
Logical constant indicating the use of error flags for diagnosis condition at Low Pressure					
C_LAMB_DIF_FIL_MAX	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Threshold of lambda difference for FTL diagnosis					
LC_ER_FTL_MIN_ACT	1	0...1H	0...1	1	[-]
Logical constant indicating the use of engine roughness flags for diagnosis condition					
C_FUP_DIF_FTL_MAX	1	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel Pressure Difference Maximum for FTL diagnosis					
LC_STATE_FTL_MIN_ENA	1	0...1H	0...1	1	[-]
Function enable condition					
C_FTL_MIN_DIAG_LE	1	0...7FH	0...127	1	[l]
Threshold fuel level left					
C_FTL_MIN_DIAG_RI	1	0...7FH	0...127	1	[l]
Threshold fuel revel right					

C_TEG_DYN_FTL_MIN	1	0...7FF0H	0...2047	0.0625	[°C]
Temperture threshold					
C_TEG_DYN_FTL_HYS	1	0...7FF0H	0...2047	0.0625	[°C]
Temperature hysteresis					
C_FTL_MIN_DIAG_HYS	1	0...7FH	0...127	1	[l]
Hysteresis					
C_CRLC_FTL_MMV	1	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation constant for calculation of moving mean value of FTL_LE/RI_MMV					
IP_FUP_DIAG_FTL_MIN	4	0...FFFFH	0...347776	5.3067216	[hPa]
LDP_N_32_FUP_DIAG_FTL_MIN	4	0...FFH	0...8160	32	[rpm]
Fuel pressure threshold for FTL diagnosis					
C_FTL_RATIO_MAX	1	8000...7FFFH	-32768...32767	1	[µs]
Maximum of increments					
IP_T_AST_FTL_ACT	4	0...FFFFH	0...6553.5	0.1	[s]
LDP_TEMP_CAT_T_AST_FTL_ACT	4	0...FFFFH	-33...990.98437	0.015625	[°C]
Wait time after start to activate diag_ftl_min					
C_INH_IV_FTL_MIN	1	0...FFH	0...255	1	[-]
Bitmask of inhibit cylinders					
C_T_FTL_PUC_MAX	1	0...FFH	0...25.5	0.1	[s]
Maximum for Timer to stop ERR_SYM_FTL_MIN = SYM_0 diagnosis					
C_T_INH_IV_FTL_MAX	1	0...FFH	0...25.5	0.1	[s]
Maximum for Timer to reset cylinder inhibition due to cross reaction with other inhibit flags					

35.1.1 Activation of Diagnosis

Description:

Application conditions:

Initialisation: *at reset **or (**LV_IGK = 0->1 **and** LV_ES = 0**):** LV_FTL_MIN_ACT = 0*

Recurrence: 100 ms

Activation: always

Deactivation: otherwise

Formula section:

```
IF  LV_IGK == 1                    and
    LV_ES == 0                     and
    LV_FTL_CAN_ERR == 0            and
    LV_ERR_LS_UP[0] == 0           and
    LV_ERR_LS_UP[1] == 0           and
    LV_ERR_FTL_OBD == 0            and
    LV_ERR_MAF == 0                and
    LV_ERR_LOAD_TPS_PLAUS == 0 and
    LV_ERR_LAM_ADJ[0] == 0         and
    LV_ERR_LAM_ADJ[1] == 0         and
    LV_ERR_CHG_LS_UP == 0          and
    LV_ERR_EL_CPS == 0             and
```

```
    LV_ERR_DIAGCPS == 0           and
    LV_ERR_FSD[0] == 0            and
    LV_ERR_FSD[1] == 0            and
    LV_ERR_SA_SAV == 0            and
    LV_ERR_SA_SAV_LSL == 0        and
    LV_ERR_SAV == 0               and
    LV_ERR_T_SEG_ER == 0          and
    LV_ERR_SEG_AD_ER == 0         and
    LV_ERR_AMP == 0               and
    LV_ERR_SAP == 0               and
    LV_ERR_AMP_PLAUS == 0         and
    LV_ERR_SLV_IVVT_IN == 0       and
    LV_ERR_OFS_LSL_UP[0] == 0     and
    LV_ERR_OFS_LSL_UP[1] == 0     and
    LV_ERR_CTL_LSL_UP[0] == 0     and
    LV_ERR_CTL_LSL_UP[1] == 0     and
    LV_ERR_MAP_TPS_PLAUS == 0     and
    T_AST > IP_T_AST_FTL_ACT

THEN    LV_FTL_MIN_ACT = 1

ELSE    LV_FTL_MIN_ACT = 0
ENDIF
```

35.1.2 Diagnosis Function

General information:

The intention of this diagnosis is to deteced a fuel level getting empty. An analysis is done based on the two chambers fuel level of the fuel tank.

Application conditions:

Initialisation: **at** reset **and** LV_IGK 0 to 1

LV_ERR_FTL_MIN, RATIO_FTL, RATIO_FTL_INI, LV_CDN_DIAG_FTL_MIN, CTR_FTL_MIN_ACT, ER_FTL_MIN, ERR_SYM_FTL_MIN, LV_LAMB_PLAUS_FTL_MIN_DIAG_TMP, LV_FUP_AVL_DIAG_TMP, LV_FTL_LE_RE_ACT

all with 0 physical

FTL_RI_MMV	*= FTL_RI*
FTL_RI_MMV_H_RES	*= FTL_RI*
FTL_LE_MMV	*= FTL_LE*
FTL_LE_MMV_H_RES	*= FTL_LE*

Recurrence: 100ms

Activation: LV_FTL_MIN_ACT =1 **and** LV_ERR_FTL_MIN = 0

Deactivation: LV_FTL_MIN_ACT = 0 **or** LV_ERR_FTL_MIN = 1

Description:

Error-symptoms are defined to this diagnosis function as following :

```
□□□□
 |  |  |  └─> low pressure intervention in high pressure system      (= SYM_0)
 |  |  └────> low pressure intervention in low pressure system       (= SYM_1)
 |  └───────> -                                                       (= SYM_2)
 └──────────> low fuel level intervention                             (= SYM_3)
```

Formula section:

Reset of variables on FMY_clear:

```
If(1)     failure memory is cleared
THEN(1)        LV_LAMB_PLAUS_FTL_MIN_DIAG_TMP = 0
               LV_FUP_AVL_DIAG_TMP = 0
               STATE_ERR_FTL = 0
ENDIF(1)
```

Timer Setting to stop SYM_0 detection:

```
IF(1)          LV_PUC = 1->0        (Flank from 1 to 0)
THEN(1)        T_DIAG_FTL_MIN = C_T_FTL_PUC_MAX
ENDIF(1)
```

Lambda plausibility condition:

```
IF(1)   LC_LAMB_PLAUS_FTL_MIN_DIAG = 1 and OPM_AV = 2H  ('AFS' = HOM)
                                              and LV_PUC = 0
THEN(1)        IF(2)   {[(LV_LAMB_LS_UP_VLD[0] = 1                              and
                       LAMB_LS_UP[0] - LAMB_SP_FIL_HOM[0] >
                                              C_LAMB_DIF_FIL_MAX )     or
                       (LV_LAMB_LS_UP_VLD[1] = 1                              and
                       LAMB_LS_UP[1] - LAMB_SP_FIL_HOM[1] >
                                              C_LAMB_DIF_FIL_MAX ) ]
```

```
                      or

            [(LV_LAM_LSCL[0] = 1                      and
            LV_LAM_STOP[0] = 0                        and
            FAC_LAM_LIM[1] > C_FAC_LAM_FTL_MIN )      or
            (LV_LAM_LSCL[1] = 1                       and
            LV_LAM_STOP[1] = 0                        and
            FAC_LAM_LIM[2] > C_FAC_LAM_FTL_MIN      ) ]}
                                                      and
            T_DIAG_FTL_MIN = 0
        THEN(2)   LV_LAMB_PLAUS_FTL_MIN_DIAG_TMP = 1
        ELSE(2)   LV_LAMB_PLAUS_FTL_MIN_DIAG_TMP = 0
        ENDIF(2)
ELSE(1) LV_LAMB_PLAUS_FTL_MIN_DIAG_TMP = 0
ENDIF(1)
```

High Pressure Condition:

```
IF(1)   LC_ERR_FUP_DIAG_ACT_HPRS = 1
THEN(1) IF(2) LV_ERR_FUP_MFP_PLAUS = 0   and
              LV_ERR_FUP = 0
        THEN(2)   LV_ERR_FUP_DIAG_ACT_HPRS = 1
        ELSE(2)   LV_ERR_FUP_DIAG_ACT_HPRS = 0
        ENDIF(2)
ELSE(1) LV_ERR_FUP_DIAG_ACT_HPRS = 0
ENDIF(1)
```

Low Pressure Condition:

```
IF(1)   LC_ERR_FUP_DIAG_ACT_LPRS = 1
THEN(1) IF(2) LV_ERR_FUP_EFP = 0
        THEN(2)   LV_ERR_FUP_DIAG_ACT_LPRS = 1
        ELSE(2)   LV_ERR_FUP_DIAG_ACT_LPRS = 0
        ENDIF(2)
ELSE(1) LV_ERR_FUP_DIAG_ACT_LPRS = 0
ENDIF(1)
```

Individual Tank Levels Condition:

Right:

```
IF(1)        FTL_RI – FTL_RI_MMV > 0
THEN(1)      FTL_RI_MMV_H_RES = FTL_RI
             FTL_RI_MMV = FTL_RI_MMV_H_RES
ELSE(1)      FTL_RI_MMV_H_RES = FTL_RI_MMV_H_RES + (FTL_RI –
             FTL_RI_MMV_H_RES) * C_CRLC_FTL_MMV
             FTL_RI_MMV = FTL_RI_MMV_H_RES
ENDIF(1)
```

Left:

```
IF(1)        FTL_LE – FTL_LE_MMV > 0
THEN(1)      FTL_LE_MMV_H_RES = FTL_LE
             FTL_LE_MMV = FTL_LE_MMV_H_RES
ELSE(1)      FTL_LE_MMV_H_RES = FTL_LE_MMV_H_RES + (FTL_LE –
             FTL_LE_MMV_H_RES) * C_CRLC_FTL_MMV
             FTL_LE_MMV = FTL_LE_MMV_H_RES
ENDIF(1)
```

Empty Tank Decision based on tank level condition:

```
IF(1a)       LC_STATE_FTL_MIN_ENA == 1                    and
             FTL_LE_MMV < = C_FTL_MIN_DIAG_LE             and
             FTL_RI_MMV < = C_FTL_MIN_DIAG_RI             and
             TEG_DYN => C_TEG_DYN_FTL_MIN
THEN(1a)     LV_FTL_LE_RE_ACT = 1

ELSEIF(1b)   FTL_LE_MMV > C_FTL_MIN_DIAG_LE + C_FTL_MIN_DIAG_HYS  or
             FTL_RI_MMV > C_FTL_MIN_DIAG_RI+ C_FTL_MIN_DIAG_HYS   or
             TEG_DYN < C_TEG_DYN_FTL_MIN - C_TEG_DYN_FTL_HYS
THEN(1b)     LV_FTL_LE_RE_ACT = 0
ELSE(1)      LV_FTL_LE_RE_ACT(n) = LV_FTL_LE_RE_ACT(n-1)
ENDIF(1)
```

Error Decision:

```
IF(1)        OPM_AV = 2H  ('AFS' = HOM)
THEN(1)      IF(2)   LV_LAMB_PLAUS_FTL_MIN_DIAG_TMP = 1
```

```
          THEN(2)
                  IF(2a)        LV_FTL_LE_RE_ACT = 1                        and
                                (STATE_ERR_FTL = SYM_0                      or
                                STATE_ERR_FTL = SYM_1)                      and
                                LV_FUP_AVL_DIAG_TMP = 1
                  THEN(2a)      LV_FUP_AVL_DIAG_TMP = 1
                                STATE_ERR_FTL = SYM_3

                  ELSEIF(2b)    LV_FUP_AVL_DIAG_TMP = 1
                  THEN(2b)      STATE_ERR_FTL(n) = STATE_ERR_FTL(n-1)

                  ELSEIF(2c)    FUP_SP – FUP > C_FUP_DIF_FTL_MAX            and
                                FUP < IP_FUP_DIAG_FTL_MIN                   and
                                LV_ERR_FUP_DIAG_ACT_HPRS = 1
                  THEN(2c)      LV_FUP_AVL_DIAG_TMP = 1
                                STATE_ERR_FTL = SYM_0

                  ELSEIF(2d)    FUP_EFP < C_FUP_EFP_DIAG_FTL_MIN            and
                                LV_ERR_FUP_DIAG_ACT_HPRS = 0                and
                                LV_ERR_FUP_DIAG_ACT_LPRS = 1
                  THEN(2d)      LV_FUP_AVL_DIAG_TMP = 1
                                STATE_ERR_FTL = SYM_1

                  ELSE(2d)      LV_FUP_AVL_DIAG_TMP = 0
                                STATE_ERR_FTL= NO_SYM
                  ENDIF(2d)

                  ENDIF(2c)
                  ENDIF(2b)
                  ENDIF(2a)
          ENDIF(2)
ELSE(1)   LV_FUP_AVL_DIAG_TMP(n) = LV_FUP_AVL_DIAG_TMP(n-1)
ENDIF(1)
```

Error Setting:

```
IF(0)    LC_ER_FTL_MIN_ACT = 1
THEN(0)
   IF(1)      LV_FUP_AVL_DIAG_TMP = 1
   THEN(1)  LV_CDN_DIAG_FTL_MIN = 0
        IF (2)  RATIO_FTL_INI == 0
        Then(2)        IF (3)  CTR_FTL_MIN_ACT < C_CTR_FTL_MAX
                Then(3)        CTR_FTL_MIN_ACT = CTR_FTL_MIN_ACT + 1
                               ER_FTL_MIN = ER_FTL_MIN + abs(ER)
                ELSE(3)   RATIO_FTL_INI = ER_FTL_MIN / C_FTL_RATIO_MAX
                             CTR_FTL_MIN_ACT = 0
                             ER_FTL_MIN = 0
                Endif(3)
        else(2)
           IF(3)         CTR_FTL_MIN_ACT < C_CTR_FTL_MAX
           Then(3)       CTR_FTL_MIN_ACT = CTR_FTL_MIN_ACT + 1
                         ER_FTL_MIN = ER_FTL_MIN + abs(ER)
           ELSE(3)       RATIO_FTL  = ER_FTL_MIN / C_FTL_RATIO_MAX
                IF(4)    RATIO_FTL > RATIO_FTL_INI * C_FAC_RATIO_FTL_INI_MAX or
                         RATIO_FTL < RATIO_FTL_INI * C_FAC_RATIO_FTL_INI_MIN
                Then(4)        LV_CDN_DIAG_FTL_MIN = 1
                               IF(5)   STATE_ERR_FTL = SYM_0
                               THEN(5)        ERR_SYM_FTL_MIN = SYM_0

                               ELSEIF(5)      STATE_ERR_FTL = SYM_1
                               THEN(5)        ERR_SYM_FTL_MIN = SYM_1

                               ELSEIF(5)      STATE_ERR_FTL = SYM_3
                               THEN(5)        ERR_SYM_FTL_MIN = SYM_3

                               ELSE(5)        ERR_SYM_FTL_MIN = NO_SYM
                               ENDIF(2c)
                               CTR_FTL_MIN_ACT = 0
                               ER_FTL_MIN = 0
                               debounce LV_ERR_FTL_MIN
                               C_ABC_INC_FTL_MIN; C_ABC_MAX_FTL_MIN
```

```
                Else(4)      reset   ER_FTL_MIN = 0
                             CTR_FTL_MIN_ACT = 0
                Endif(4)
            Endif(3)
        Endif(2)
    ELSE(1)     LV_CDN_DIAG_FTL_MIN = 1
                ERR_SYM_FTL_MIN = NO_SYM
                Rebounce with C_ABC_DEC_FTL_MIN
                RATIO_FTL_INI =  RATIO_FTL = 0;
                ER_FTL_MIN = 0
                CTR_FTL_MIN_ACT = 0
    Endif(1)
ELSE(0)
        IF(1)   LV_FUP_AVL_DIAG_TMP = 1
                IF(2)   STATE_ERR_FTL = SYM_0
                THEN(2)      ERR_SYM_FTL_MIN = SYM_0
                             LV_CDN_DIAG_FTL_MIN = 1
                             debounce LV_ERR_FTL_MIN
                             C_ABC_INC_FTL_MIN; C_ABC_MAX_FTL_MIN

                ELSEIF(2)    STATE_ERR_FTL = SYM_1
                THEN(2)      ERR_SYM_FTL_MIN = SYM_1
                             LV_CDN_DIAG_FTL_MIN = 1
                             debounce LV_ERR_FTL_MIN
                             C_ABC_INC_FTL_MIN; C_ABC_MAX_FTL_MIN

                ELSEIF(2)    STATE_ERR_FTL = SYM_3
                THEN(2)      ERR_SYM_FTL_MIN = SYM_3
                             LV_CDN_DIAG_FTL_MIN = 1
                             debounce LV_ERR_FTL_MIN
                             C_ABC_INC_FTL_MIN; C_ABC_MAX_FTL_MIN

                ELSE(2)      ERR_SYM_FTL_MIN = NO_SYM
                             Rebounce with C_ABC_DEC_FTL_MIN
                ENDIF(2)
        ENDIF(1)
```

ENDIF(0)

35.1.3 Inhibition Flag

Application conditions:

Initialisation: *INH_IV_FTL_MIN* = 0 ; *T_DIAG_FTL_MIN = 0*

Recurrence: 100ms

Activation: LV_IGK ==1

Deactivation: LV_IGK ==0

Formula section:

Timer Setting to stop SYM_0 detection:

Decrement T_DIAG_FTL_MIN

```
IF (1a)         LV_ERR_FTL_MIN = 1                              and
                STATE_ERR_FTL = SYM_0                           and
                (
                LV_FUP_LIH_L_PRS_CTL_REQ = 1                    or
                LV_FUP_LIH_HOM_VCV_OPEN_REQ = 1                 or
                LV_FUP_LIH_HOM_REQ = 1
                )                                               and
                T_DIAG_FTL_MIN = 0
THEN(1a)        INH_IV_FTL_MIN = 00H

ELSEIF (1b)     LV_ERR_FTL_MIN = 1
THEN(1b)        INH_IV_FTL_MIN = C_INH_IV_FTL_MIN
    IF(2a)          STATE_ERR_FTL = SYM_0                       and
                    LV_FUP_LIH_L_PRS_CTL_REQ = 0                and
                    LV_FUP_LIH_HOM_VCV_OPEN_REQ = 0             and
                    LV_FUP_LIH_HOM_REQ = 0
    THEN(2a)        T_DIAG_FTL_MIN = C_ T_INH_IV_FTL_MAX
    ENDIF(2a)

ELSE(1b)        INH_IV_FTL_MIN = 00H
                "BIT mask for selection of  cylinder to inhibit"
ENDIF(1b)
```

35.2 Fuel tank level (Plausibility error)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_FTL_OBD	V/O	0...1H	0...1	1	[-]
error flag fuel tank level					
LV_CDN_DIAG_FTL_OBD	V/O	0...1H	0...1	1	[-]
Diagnosis condition					
ERR_SYM_FTL_OBD	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH BH CH DH EH FH	NO_SYM SYM_0 SYM_1 SYM_0, SYM_1 SYM_2 SYM_0, SYM_2 SYM_1, SYM_2 SYM_0, SYM_1, SYM_2 SYM_3 SYM_0, SYM_3 SYM_1, SYM_3 SYM_0, SYM_1, SYM_3 SYM_2, SYM_3 SYM_0, SYM_2, SYM_3 SYM_1, SYM_2, SYM_3 SYM_0, SYM_1, SYM_2, SYM_3	0	[-]
Detected error					
LV_END_DIAG_FTL_OBD	V/O	0...1H	0...1	1	[-]
End of Diagnosis					

Input data:

LV_ST	LV_ERR_FTL_PLAUS	LV_IGK	C_CONF_DMTL
LV_ES	LV_FTL_DIAG		

FUNCTION DESCRIPTION:

General information:

This diagnosis is for collecting the failure bit of the fuel tank level plausibility function.

Description:

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                               (= SYM_0)
 │ │ └───> -                               (= SYM_1)
 │ └─────> -                               (= SYM_2)
 └───────> FTL plausibility error          (= SYM_3)
```

Application conditions:

Initialisation: at reset **or** LV_IGK = 0 → 1 **or** clear fmy all 0

Recurrence: 1 s

Activation:

```
    If      C_CONF_DMTL = 1                     and
            LV_IGK = 1                          and
            LV_ES = 0                           and
            LV_ST = 0                           and
            LV_FTL_DIAG = 1
    Then    LV_CDN_DIAG_FTL_OBD = 1
    Else    LV_CDN_DIAG_FTL_OBD = 0
    Endif
```

Formula section:

```
If    LV_ERR_ FTL_PLAUS = 1
Then        ERR_SYM_FTL_OBD = SYM_3
            LV_ERR_FTL_OBD = 1
Else        ERR_SYM_FTL_OBD = NO_SYM
            LV_ERR_FTL_OBD = 0
Endif
LV_END_DIAG_FTL_OBD  = 1
```

35.3 Fuel tank level (CAN error)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_FTL_LE_CAN	V/O	0...1H	0...1	1	[-]
error flag for left (second) fuel tank level					
LV_CDN_DIAG_FTL_LE_CAN	V/O	0...1H	0...1	1	[-]
Diagnosis condition					
ERR_SYM_FTL_LE_CAN	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error					
LV_END_DIAG_FTL_LE_CAN	V/O	0...1H	0...1	1	[-]
End of Diagnosis					
LV_ERR_FTL_RI_CAN	V/O	0...1H	0...1	1	[-]
error flag for right (first) fuel tank level					
LV_CDN_DIAG_FTL_RI_CAN	V/O	0...1H	0...1	1	[-]
Diagnosis condition					
ERR_SYM_FTL_RI_CAN	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error					
LV_END_DIAG_FTL_RI_CAN	V/O	0...1H	0...1	1	[-]
End of Diagnosis					

Input data:

LV_ST	LV_ERR_TOUT_ICL_7	LV_IGK	C_CONF_DMTL
LV_ES	LV_STATE_ERR_CAN_TOUT	STATE_ERR_SYM_FTL_RI_CAN	LV_FTL_TOT_CAN_ERR
LV_FTL_LE_CAN_ERR	LV_FTL_RI_CAN_ERR	STATE_ERR_SYM_FTL_LE_CAN	

FUNCTION DESCRIPTION:

General information:

This diagnosis is for collecting the failure bits for LV_ERR_FTL_LE_CAN and LV_ERR_FTL_RI_CAN from CAN (Kombi).

Description:

Error-symptoms are defined for LV_ERR_FTL_LE_CAN and LV_ERR_FTL_RI_CAN diagnosis function as following :

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  |  |  |  └─> short circuit to vbatt or open line      (= SYM_0)
  |  |  └────> short circuit to ground                  (= SYM_1)
  |  └───────> -                                        (= SYM_2)
  └──────────> FTL error hw based from ICL              (= SYM_3)
```

Application conditions:

Initialisation: **std_ini**

Recurrence: 1 s

```
Activation:  If      C_CONF_DMTL = 1              and
                     LV_IGK = 1                   and
                     LV_ES = 0                    and
                     LV_ST = 0
             Then    LV_CDN_DIAG_FTL_LE_CAN = 1
                     LV_CDN_DIAG_FTL_RI_CAN = 1
             Else    LV_CDN_DIAG_FTL_LE_CAN = 0
                     LV_CDN_DIAG_FTL_RI_CAN = 0
             Endif
```

Formula section:

```
//calculation or LV_ERR_FTL_LE_CAN:
IF       STATE_ERR_SYM_FTL_LE_CAN = SCP
Then     ERR_SYM_FTL_LE_CAN = SYM_0
Elseif   STATE_ERR_SYM_FTL_LE_CAN = SCG
Then     ERR_SYM_FTL_LE_CAN = SYM_1
Elseif   LV_FTL_TOT_CAN_ERR = 1          or
         LV_FTL_LE_CAN_ERR = 1           or
         LV_STATE_ERR_CAN_TOUT = 1       or
         LV_ERR_TOUT_ICL_7 = 1
Then     ERR_SYM_FTL_LE_CAN = SYM_3
Else     ERR_SYM_FTL_LE_CAN = NO_SYM
Endif
LV_ERR_FTL_LE_CAN, LV_END_DIAG_FTL_LE_CAN set by ERRM after debounce

//calculation or LV_ERR_FTL_RI_CAN:
IF       STATE_ERR_SYM_FTL_RI_CAN = SCP
Then     ERR_SYM_FTL_RI_CAN = SYM_0
Elseif   STATE_ERR_SYM_FTL_RI_CAN = SCG
Then     ERR_SYM_FTL_RI_CAN = SYM_1
Elseif   LV_FTL_TOT_CAN_ERR = 1          or
         LV_FTL_RI_CAN_ERR = 1           or
```

```
          LV_STATE_ERR_CAN_TOUT = 1   or
          LV_ERR_TOUT_ICL_7 = 1
Then      ERR_SYM_FTL_RI_CAN = SYM_3
Else      ERR_SYM_FTL_RI_CAN = NO_SYM
Endif
```

LV_ERR_FTL_RI_CAN, LV_END_DIAG_FTL_RI_CAN set by ERRM after debounce

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_FTL_LE_CAN	1	0...FFH	0...255	1	[-]
Increment for debouncing diagnosis					
C_ABC_MAX_FTL_LE_CAN	1	1...FFH	1...255	1	[-]
Max threshold for diagnosis					
C_ABC_INC_FTL_RI_CAN	1	0...FFH	0...255	1	[-]
Increment for debouncing diagnosis					
C_ABC_MAX_FTL_RI_CAN	1	1...FFH	1...255	1	[-]
Max threshold for diagnosis					

35.4 Fuel tank level diagnosis (Applic. Inc.)

35.4.1 Interface for Rate – Based – Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_FTL_OBD	V/O	0...7H	0...7	1	[-]
Interface of FTL_OBD monitor with the Rate-Based Monitoring statistics Bit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1) Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1) Bit 2: individual RBM conditions of the monitor were encountered within this DC (bit 2 = 1)					

Input data:

LV_DC	LV_END_DIAG_FTL_OBD		

FUNCTION DESCRIPTION:

General information:

With this module the interface between the FTL_OBD monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_FTL_OBD data.

Within STATE_RBM_FTL_OBD, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for FTL_OBD diagnosis)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_FTL_OBD = 0

on failure memory reset :

bit 1 of STATE_RBM_FTL_OBD = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

Bit 0:

```
If    bit 0 of STATE_RBM_FTL_OBD = 0
then
      if      LV_END_DIAG_FTL_OBD = 1
      then    bit 0 of STATE_RBM_FTL_OBD = 1
      endif
endif
```

Bit 1:

bit 1 of STATE_RBM_FTL_OBD = 0

Bit 2:

bit 2 of STATE_RBM_FTL_OBD = 1

36 General

Table of Contents

Index

36.1 Cylinder numbering

In the software the cylinders are numbered logically with 0...5. The relating physical cylinders are numbered from 1...6. If in the specification a cylinder-specific-variable is mentioned with ..._x or ...[CYL] then always the logic value is ment. The correct firing order of any 6 cylinder engine has to be guaranteed by the wiring harness, see system picture.

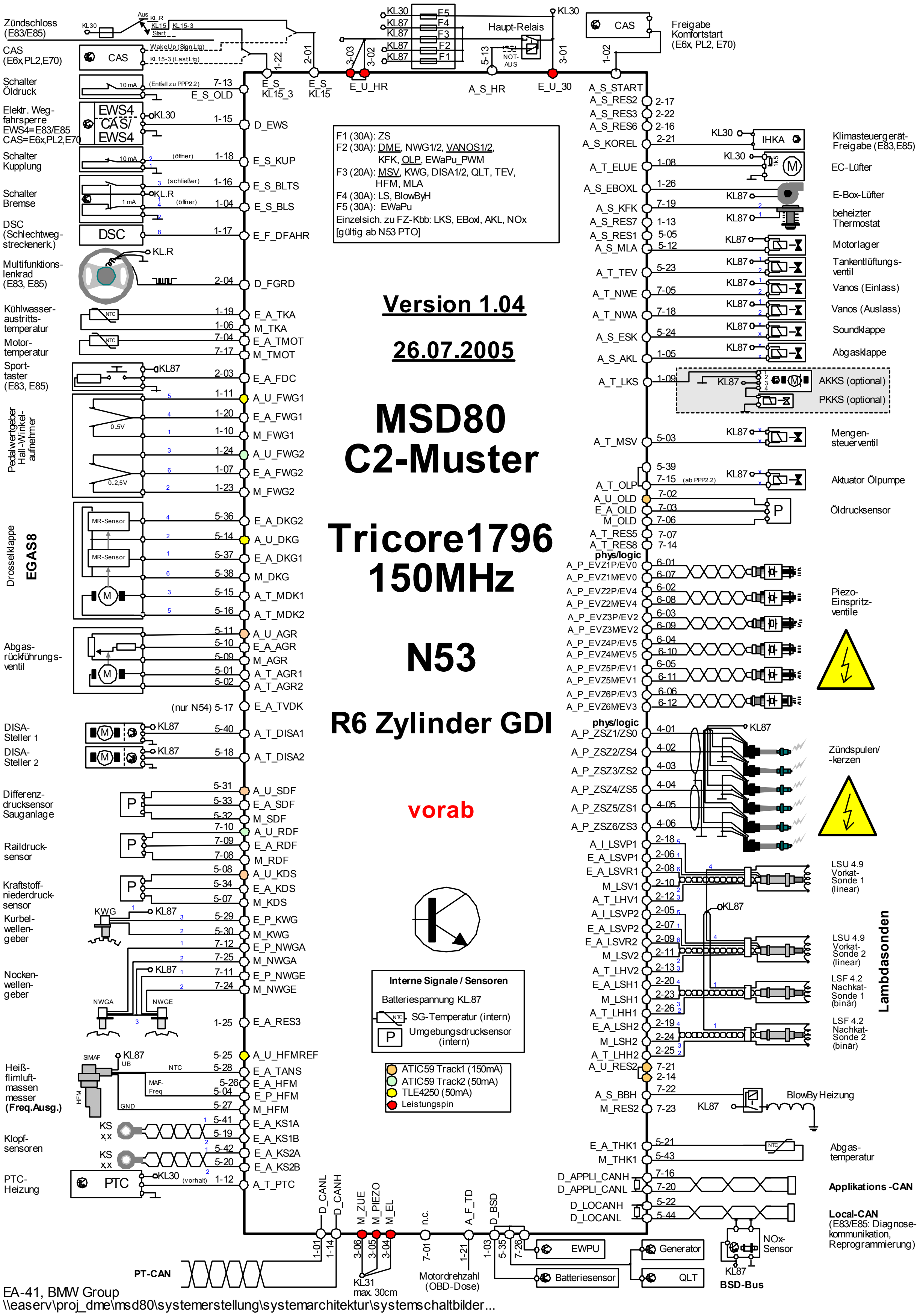
The following graphic shows examples of physical cylinder numbers:

M54 / M56 / NG R6
Flywheel
6
5
4
3
2
1

The relation between logical and physical cylinder numbers is as follows:

Logic	Physical M5x / NG R6	bank
0	1	1
1	5	2
2	3	1
3	6	2
4	2	1
5	4	2

36.2 System Pictures



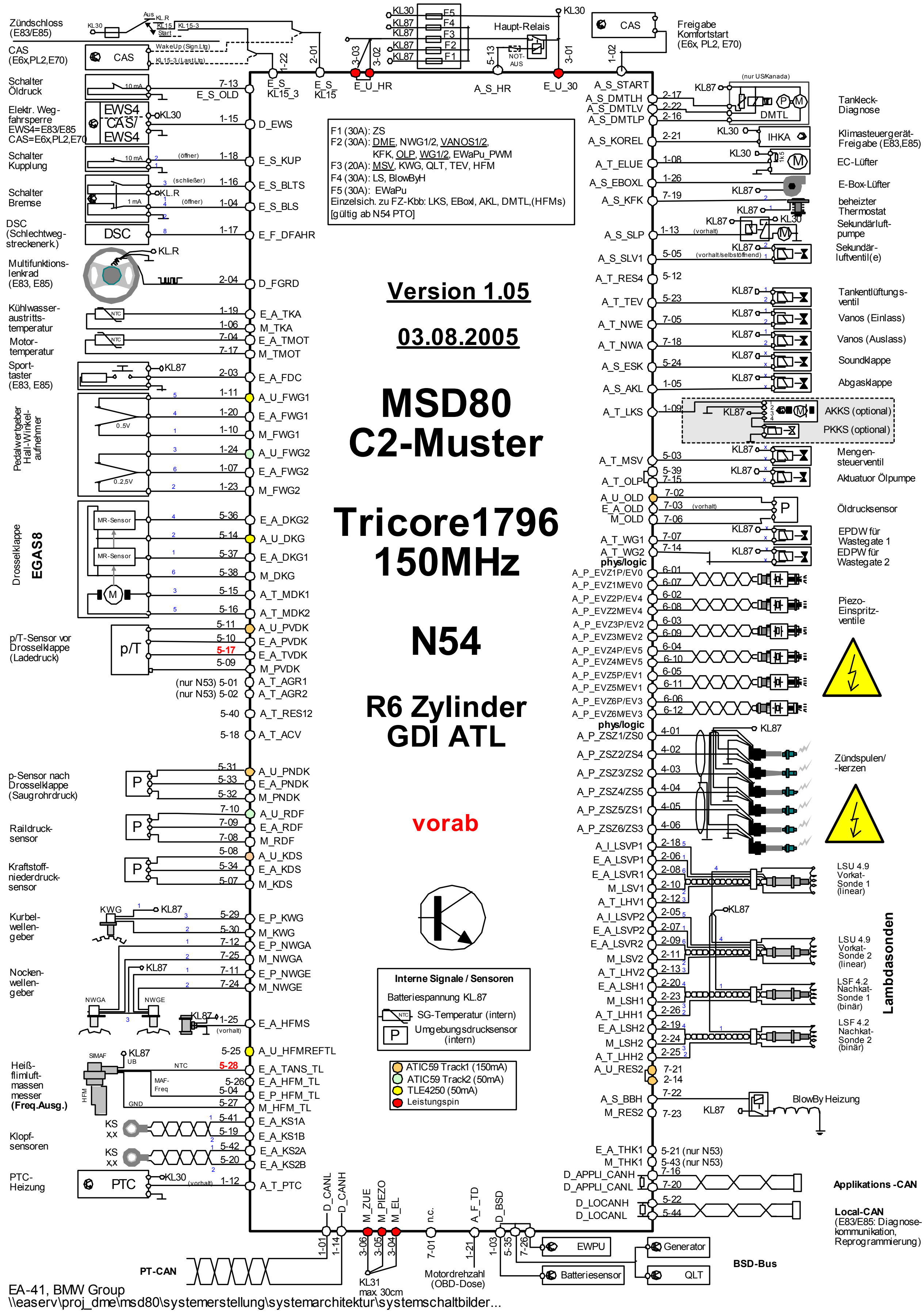
F1 (30A): ZS
F2 (30A): DME, NWG1/2, VANOS1/2, KFK, OLP, EWaPu_PWM
F3 (20A): MSV, KWG, DISA1/2, QLT, TEV, HFM, MLA
F4 (30A): LS, BlowByH
F5 (30A): EWaPu
Einzelsich. zu FZ-Kbb: LKS, EBoxl, AKL, NOx
[gültig ab N53 PTO]

Interne Signale / Sensoren

Batteriespannung KL.87
NTC SG-Temperatur (intern)
P Umgebungsdrucksensor (intern)

🟧 ATIC59 Track1 (150mA)
🟩 ATIC59 Track2 (50mA)
🟨 TLE4250 (50mA)
🟥 Leistungspin

EA-41, BMW Group
\\easerv\proj_dme\msd80\systemerstellung\systemarchitektur\systemschaltbilder...



EA-41, BMW Group
\\easerv\proj_dme\msd80\systemerstellung\systemarchitektur\systemschaltbilder...

Chapter	Baseline	Include File
General	4DC3940S	43100103.00E

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Regensburg(RBG)

A4 : 2004-06

36.3 Dataset identification and checksum calculation

FUNCTION DESCRIPTION:

General information:

The INCA application tool doesn’t store a dataset identifier when a dataset is sent to the ECU. Therefore it is not possible to empathise which dataset really is in an ECU. To solve this problem three calibratable variables C_DATA_ID_0, C_DATA_ID_1 and C_DATA_ID_2 are introduced. Every variable can be calibrated with 5 ASCII signs, wich are the identifier of the dataset. The variables are only visible in INCA (SAM doesn't support ascii format).

To activate or deactivate the checksum calculation over the dataset with the help of INCA, the constant C_CHECK_SUM has been introduced. If the value of C_CHECK_SUM is = 0000h, the checksum calculation is deactivated. If the value of C_CHECK_SUM is <> 0000h, the checksum calculation is active.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DATA_ID_0	1	0...FFH	ASCII	1	[-]
Dataset identifier string 1					
C_DATA_ID_1	1	0...FFH	ASCII	1	[-]
Dataset identifier string 2					
C_DATA_ID_2	1	0...FFH	ASCII	1	[-]
Dataset identifier string 3					
C_CHECK_SUM	1	0...FFFFH	0...65535	1	[-]
Constant to activate the dataset checksum calculation					

36.4 Nomenclature of the variables used

Keyword	English definition	German definition
0, 1, ...	Index 0, 1, ... n	Index
A, B, ...	Index A ... Z	Index
AAI	Air assisted injector	luftumfaßte Einspritzventile
ABC	Anti-Bounce counter	Entprellzähler
ABS	Antilock Breaking System	Antiblockiersystem
ABSV	absolute value	Betrag
AC	acceleration	Beschleunigung
ACC	Air conditioning compressor	Klimakompressor
ACCIN	Air conditioning compressor request signal	Klimakompressor
ACCOUT	Air conditioning compressor output signal	Klimakompressor Ausgangssignal
ACF	Aktivated charcoal filter	Aktivkohlefilter
ACIN	Air conditioning request signal	Klimaanlage
ACK	Acknowledge	Bestätigung
ACLS	air conditioning load signal	Klimalastsignal
ACP	Air conditioning pressure	Druck Klimaanlage
ACPPWM	Air condition pressure PWM	Tastverhältnis Klimadruck
ACQ	Acquisition	Akquisition
ACR	actuator	Ansteller
ACT	active	aktiv
AD	Adaptive	Adaption
ADC	Advance (early)	früh
ADD	Additive	Additive Korrektur
ADJ	adjustment	Verstellung
ADR	Address	Adresse
AE	Acceleration enrichment	Beschleunigungs-anreicherung
AEB	active engine brackets	aktive Motorlager
AETCU	Autarkic electronic transmission control unit	autarke elektronische Getriebesteuerung
AFL	Air fuel lean	mageres Gemisch
AFR	Air fuel rich	fett
AFS	Air fuel stoichiometric	stöchiometrisches Gemisch
AGI	aging	Alterung
AI	air injection	Lufteinblasung
AIRB	airbag	Airbag
AJ	Anti-jerk-function	Anti-Ruckel-Funktion
ALS	Alarm system	Diebstahlwarnanlage
ALTER	alternator	Lichtmaschine
ALTI	altitude	Höhe
ALTPWM	alternator PWM	Tastverhältnis Lichtm.
AM	ambient	Umgebung
AMP	Ambient pressure	Umgebungsdruck
AMPL	amplitude	Amplitude
AN	Analog	analog

Keyword	English definition	German definition
ANTI	Anti	anti
APP	Application generated parameter	
AR	Area	Fläche
ARP	air rail pressure	
AS	Application system	Applikationssystem
ASA	After sale service	Kundendienst
ASC	Asynchronous serial communication interface	
ASR	Anti slip regulation	Antischlupfregelung
AST	After-start	Nachstart
AT	Automatic shifted transmission	Automatikgetriebe
ATJS	automatic transmission jump start	Knallstart
AUT	Automatic	automatisch
AUTH	Authorize	autorisieren
AV	Actual value	Istwert
AVL	available	verfügbar
BACK	back	zurück
BAS	Base	Basis
BD	burning duration	Brenndauer
BDC	Bottom dead center	unterer Totpunkt
BEG	Begin	Anfang
BFS	bi fuel switch	
BHC	Burner heated catalyst	Brennerkat
BIT	Bit	Bit
BLS	Brake light switch	Bremslichtschalter
BOFF	bus off mode	
BOL	Bottom limit	untere Grenze
BOOT	boot	
BOP	Boost pressure (diesel)	Ladedruck
BPAPWM	PWM of boost pressure adjustment (diesel)	Tastverhältnis Ladedrucksteller
BRAKE	brake	Bremse
BRI	Bridge contact	Brücke
BTS	Brake test switch	Bremslichttest-schalter
BUF	buffer	
BUZ	buzzer	Summer
BWL	Byte word long	
BYTE	byte	byte
C	calibration data changeable with the application system	vom Applikationssystem änderbare Daten
CAL	calibration	Abstimmung, Eichung
CALL	call	
CAM	Camshaft	Nockenwellensignal
CAN	Controlled area network	Controlled area network
CAP	charge air pressure	Ladeluftdruck
CASE	case	Fall (-unterscheidung)
CAST	Cold after-start	Kaltnachstart
CAT	Catalyst	Katalysator

Keyword	English definition	German definition
CBG	continuous gearbox	
CBK	cylinder bank	Zylinderbank
CC	Converter clutch	Wandlerkupplung
CCH	Combustion chamber heating function	Brennraumaufheiz-funktion
CDN	Condition	Bedingung
CEI	combustion efficiency index	
CFA	Cooling fan	Ventilator
CFB	current feedback	Stromrückmeldung
CFM	Confirmation	Bestätigung
CH	catalyst heating	Katheizen
CHG	change	Änderung
CHK	check	Überprüfung
CIS	carried idle switch	mitgeführter Leerlauf-kontakt
CKS	Checksum	Prüfsumme
CL	canister load	Beladungsgrad Aktivkohlefilter
CLC	calculated	berechnet
CLDT	cold test	Kalttest
CLK	clock	clock
CLOSE	close	geschlossen
CLR	Clear	Löschsignal
CLU	Clutch	Kupplung
CMB	combustion	Verbrennung
CMD	Command	Befehl
CMP	compression	Verdichtung
CMT	comfort manual transmission	
CNL	cancel	aufheben, streichen
CNV	convection	Konvektion
COD	Code	Code
COL	collector	Sammler
COLD	Cold	kalt
COM	Communication	Kommunikation
COMP	comparison	Vergleich
CON	Constant	Konstante
CONC	concentration	Konzentration
CONF	Configuration	Konfiguration
CONV	converter	Wandler
COP	catalyst overheating prevention	Kat-Schutz
COR	Correction	Korrektur
CORD	Coordination	Koordination
CP	Canister purge	Tankentlüftung
CPL	Complement	
CPP	canister purge pump	
CPPWM	Pulse width modulation of canister-purge	Tastverhältnis Tankentlüftung
CPS	Canister purge solenoid	Tankentlüfungsventil
CQ	characteristic quantity	Kenngröße
CR	controlled reset	Rücksteuerung

Keyword	English definition	German definition
CRASH	Crash	Zusammenstoß
CRIT	critical	kritisch
CRK	Crankshaft/ Crankshaft angle	Kurbelwelle/Kurbel-wellenwinkel
CRLC	Correlation constant	Mittelungskonstante
CRU	Cruise control	Fahrgeschwindigkeits-regelung (Tempomat)
CRUS	Cruise control switch	Fahrgeschwindigkeitsregelungsschalter
CS	Clutch switch	Kupplungsschalter
CSI	Cold start injector	Kaltstart Einspritzdüse
CST	Cold start	Kaltstart
CT	closed throttle	geschlossene DK
CTC	clamped tube check	clamped tube check
CTL	Control	Regelung
CTR	Counter	Zähler
CTS	Closed throttle switch	Leerlaufschalter
CUR	Current	Strom
CUS	customer	Kunde
CVC	canister vent check	canister vent check
CWP	cooling water pump	Kühlwasserpumpe
CYC	Cycle	Zyklus
CYCNR	Number of Cycles	Zyklenanzahl
CYL	Cylinder	Zylinder
D	Differential part of PID	differentieller Anteil
DAMP	damping	Dämpfung
DC	driving cycle	driving cycle
DCC	distance cruise control	Abstandsregel-tempomat
DE	deviation	Abweichung
DEAC	deactivation	Deaktivierung
DEC	Decrementation	Dekrementieren
DECE	deceleration	Verzögerung
DELTA	delta	Delta
DEP	Depression	Unterdruck
DET	Detection	Erkennung
DEV	development	Entwicklung
DFCT	defective	
DFT	Default	Voreinstellung
DGO	Diagnostic output	Diagnoseausgang
DHL	down hill	bergab
DHP	Dashpot	Dashpot
DI	disable	ausschalten, nicht zulassen
DIAG	Diagnosis	Diagnose
DIAGCP	Diagnosis canister purge	Diagnose Tankentlüftung
DIAGCPS	diagnosis canister purge solenoid	Diagnose Tankentlüftungsventil
DIF	Difference	Differenz
DIG	Digital	Digital
DIGIS	digital idle speed stabilization with ignition	digitale Leerlaufstabilisierung
DIP	differential pressure	Differenzdruck

Keyword	English definition	German definition
DIR	direction	Richtung
DIRE	directory	Verzeichnis
DIS	Discrete	Diskret
DISP	Display	Anzeige
DIST	distance	Entfernung
DIV	Division	Division
DL	Data line	Datenleitung
DLL	data link layer	
DLY	Delay	Totzeit
DOWN	down	nach
DPS	Diagnostic power stage	Diagnosetreiber IC
DR	driver (HW)	Endstufe
DRI	Drive (Gear engaged)	Fahrstufe
DRIFT	Drift	Drift
DRIV	driver	Fahrer
DROF	drive off	anfahren
DRV0	Static value	statischer Wert
DRV1	Derivative 1	1. Ableitung
DRV2	Derivative 2	2. Ableitung
DS	Diagnostic service	Diagnosetelegramm
DS2	DS2 protocol (BMW)	
DSK	disk	Disk
DSLIP	Deceleration slip	Verzögerungsschlupf
DT	drive train	Antriebsstrang
DTC	Diagnostic trouble code	
DTP	Differential fuel tank pressure	Differenzdruck Tank
DUCY	duty cycle	Tastverhältnis
DUI	dual injection	
DUMMY	dummy	Dummy
DUR	Duration	Dauer
DVC	device	
DYN	Dynamic	dynamisch
DYNO	dynamometer	Prüfstand
DYW	Dynamic window	Dynamikfenster/Band-breite/Toleranzwert
E2P	E2PROM	E2PROM
EBOX	electronic housing	
ECF	electronically controlled cooling fan	elektronisch geregelter Lüftermotor
ECFPWM	electronic coolant fan PWM	Tastverhältnis elektronischer Lüfter
ECT	electronically controlled thermostat	elektronisch geregelter Thermostat
ECTPWM	electronically controlled thermostat	Tastverhältnis elektr. geregelter Thermostat
ECU	Electronic control unit	Steuergerät
EDGE	edge	Flanke
EDIT	edit	eingeben
EF	exhaust flap	Abgasklappe
EFC	effective	effektiv
EFF	efficiency	Wirkungsgrad

Keyword	English definition	German definition
EFP	Electric fuel pump	elektronische Kraftstoffpumpe
EG	Exhaust gas	Abgas
EGBP	exhaust gas back pressure	Abgasgegendruck
EGR	Exhaust gas recirculation	Abgasrückführung
EGRPWM	PWM of exhaust gas recirculation	Tastverhältnis Abgasrückführsteller
EGRV	EGR valve	EGR Ventil
EGY	Energy	Energie
EHC	Electrical heated catalyst	elektrisch beheizter Katalysator
EL	Electrical	elektrisch
EMS	Engine management system	elektronische Motorsteuerung
ENA	enable	zulassen
END	End	Ende
ENG	Engine	Motor
ENVD	Environmental Data	Umweltbedingungen
EOA	end of air	Ende Lufteinblasung
EOF	End of filling (diesel)	Füllendewinkel
EOI	End of injection	Einspritzendewinkel
EOL	end of line	Bandende
EOLP	end of line programming	Bandende-programmierung
EP	Enrichment potentiometer	Anreicherungs-messung
EQU	equal	gleich
EQUIP	equipment	Ausstattung
ER	Engine roughness	Laufunruhe
ERR	Error	Fehler
ERU	engine running	Motor läuft
ES	Engine stop	Motor steht
ESB	enable Symptom Byte	Fehlerspeicherfreigabe
ESP	electronic stability program	elektronisches Stabilitätsprogramm
ESS	Engine speed signal	Drehzahlsignal
ESTIM	estimated	geschätzt
ET	emission test	Abgasuntersuchung
ETC	Electronic throttle control	elektronische Motor-leistungssegelung
ETCU	Electronic transmission control unit	elektronische Getriebesteuerung
ETCUPWM	Electronic transmission control unit PWM	Tastverhältnis elektronische Getriebesteuerung
EVAP	evaporative system	Tankentlüftungssystem
EVE	Event	Ereignis
EVSL	engine vehicle speed limiter	motorischer Geschw.-begrenzer
EWRN	Error warning status mode	
EX	Exhaust	Auslaß
EXO	exotherm	
EXPV	expansion valve	Entspannungsventil
EXT	external	extern
EXV	Exhaust valve	Auslaßventil
FAC	Factor , coefficient	Faktor
FAC	factor, coefficient	Faktor

Keyword	English definition	German definition
FAD	fuel air delay	Verzögerung Kraftstoff- Gemischeinspritzung
FALL	falling	fallend
FAN	fan	Lüfter
FAST	Fast	Schnell
FB	Feed back	Rückmeldung
FCO	Fuel consumption	Kraftstoffverbrauchsanzeige
FCOPWM	Fuel consumption PWM	Tastverhältnis Kraftstoffverbrauch
FCT	functional	Funktional
FCUT	Fuel cutoff	Kraftstoffabschaltung
FDIN	fade in	einblenden
FDOUT	fade out	ausblenden
FE	fuel enrichment	Anfettung
FIL	Filter	Filter
FILE	file	Datei
FINE	fine	fein
FIRST	first	zuerst, erst
FL	Full load	Vollast
FLL	flash light	Blinkcode
FLOOD	Flood	Absaufen des Motors
FLOW	flow	Fluß
FLS	Flash	
FLUC	fluctuation	Schwankung
FM	frequency modulation	Frequenzmodulation
FMT	Format	Format
FMY	Failure memory	Fehlerspeicher
FN	front	vorn
FOL	Follower	folgend
FP	Fast pulse	Fast pulse
FPA	Fuel pressure actuator	Kraftstoffdrucksteller
FPAPWM	Fuel pressure PWM	Tastverhältnis Kraftstoffdruck
FPS	fuel pressure sensor	Benzindrucksensor
FPV	Flash programming voltage enable (switch)	Programmierspannungsfreigabe (Schalter)
FRC	frequency counter	Häufigkeitszähler
FRF	freeze frame	OBD II Umweltbedingungen
FRQ	Frequency	Frequenz
FRS	Fuel reserve signal (4-liter-signal)	Füllstandssignal Tank (4-Liter-Signal)
FSD	fuel system diagnosis	Kraftstoffsystem-diagnose
FTL	Fuel tank level	Füllstandsniveau
FUC	fuel cap	Tankdeckel
FUEL	fuel	Kraftstoff
FUP	Fuel pressure	Kraftstoffdruck
FW	freewheeling	Freilaufkreis
GAC	actual gear code	aktueller Gang, kodiert
GAIN	Gain	Verstärkung
GAP	gap	Lücke
GAS	gas	Gas

Keyword	English definition	German definition
GB	gear box	Getriebe
GC	Gear change	Getriebeeingriff
GCB	Gear change begin	Getriebeeingriff-beginn
GCD	Gear change duration	Getriebeeingriffs-dauer
GEAR	gear	Gang
GEN	generator	Generator
GENC	Generator charge connection	Generatorladeleitung
GL	Glow	glühen
GND	Ground	Masse
GP	gear protection	Getriebeschutz
GR	gear ratio	Gang-Verhältnis
GRD	Gradient/slope	Gradient
GS	Gear shift signal	Getriebeschaltsignal
GSC	Gear shift characteristic	Schaltcharakteristik
GSPWM	gear shift signal PWM	Tastverhältnis Getriebeschaltsignal
H	High	high
HAST	Hot after-start	Heißnachstart
HB	high byte	high byte
HC	hydro carbon	Kohlenwasserstoff
HCHPWM	PWM of HC heater	
HD	header	
HDLP	head lamp	Frontscheinwerfer
HEAT	Heat	Beheizen
HFM	Hot film air mass meter	Heißfilmluftmassen-messer
HIV	heater inlet valve	Heizungsvorlaufventil
HLD	Hold	halten
HOM	homogeneous	homogen
HOT	Hot	heiß
HPF	Hight pass filter	Hochpaßfilter
HST	hot start	Heißstart
HTP	heating performance	Heizleistung
HYS	Hysteresis	Hysterese
I	Integral part of PID	Integraler Anteil
I2C	I^2C Bus (similar SPI)	
IC	integrated circuit	integrated circuit
ICL	instrument cluster	Kombiinstrument
ICO	intercooler	Ladeluftkühler
ID	index table with arbitrary many inputs without interpolation	Tabelle mit n-Eingängen ohne Interpolation.
IDM	ID table which is used in different modules with different input data	Tabelle, die in mehreren Modulen mit unterschiedlichen Eingangsgrößen benutzt wird.
IDX	index	Index
IF	Interface	Interface
IGA	Ignition angle	zylinderindividueller Gesamtzündwinkel
IGC	Ignition coil	Zündspule
IGCFB	Ignition current feedback	Zündstromrückmeldung

Keyword	English definition	German definition
IGK	Ignition key	Zündspannung
IGN	ignition	Zündung
IL	interactive layer	
ILSH	Current in the lambda sensor heater	Strom an der Lambda sondenheizung
IM	Intake manifold	Saugrohr
IMOB	Immobilizer	elektronische Wegfahrsperre
IMP	Impulse	Impuls
IN	Intake	Einlaß
INC	Increase	Inkrementierung
IND	Individual	individuell
INDU	inductive	induktiv
INF	intercept function	Abfangfunktion
INFO	information	Information
INH	Inhibition	hemmen
INI	Initialization	Initialisierungswert
INJ	injection	Einspritzung
INP	Input	Eingang
INT	Integral	Integral
INTER	Intermediary	Zwischenwert
INTM	intermittent	diskontinuierlich, sporadisch
INTR	interrupt	Unterbrechung
INTV	intervention	eingreifen
INV	intake valve	Einlaßventil
IP	index table with arbitrary many inputs with interpolation	Tabelle mit n-Engängen mit Interpolation.
IPLSL	pumping current of LSL	Strom an der linearen Lambdasonde
IPLSLPWM	PWM of pumping current of LSL	Tastverhältnis Strom an der linearen LS
IPM	IP table which is used in different modules with different input data	Kennfeld, das in mehreren Modulen mit unterschiedlichen Eingangsgrößen benutzt wird.
IPS	Idle position sensor	Leerlaufstellersensor
IS	Idle Speed	Leerlauf/Leerlauf-regler
ISA	Idle speed actuator	Leerlauffüllungssteller
ISAPWM	Pulse width modulation of idle speed actuator	Tastverhältnis Leerlauffüllungssteller
ISC	idle speed controler	Leerlaufregler
IT	internal	intern
ITC	integration time constant	Integrationszeit-konstante
IV	Injection valve	Einspritzventil
IVS	invers	invers
IVVT	infinitely variable valve timing	Regelvanos
IVVTHPWM	Holding pulse width modulation of infinitely variable valve timing	Haltetastverhältnis Regelvanos
IVVTPWM	Pulse width modulation of infinitely variable valve timing	Tastverhältnis Regelvanos
JAM	jam	klemmen
JERK	Jerk	Ruckeln
JUMP	Jump-start	Jump-start
KD	kick down	Kick down
KGH	Kilogram per hour	Kilogramm / Stunde

Keyword	English definition	German definition
KM	kilometer	
KNK	Knock control	Klopfregelung
KNKS	Knock control signal	Kopfregelungssignal
KNKWB	Beginning of knock control window	Klopfregelungsfenster-Beginnwinkel
KNKWD	Knock control window duration	Klopfregelungsfenster-Dauerwinkel
KNKWE	End of knock window	Klopfregelungs-fenster-Endewinkel
KWP	Keyword protocol 2000	
L	Low	Low
LAM	Lambda controler	Lambda/Lambda-regler
LAMB	Lambda	Lambda
LAT	lateral	
LB	low byte	
LBG	line breakage	Leitungsbruch
LC	logic calibration data	logische Konstante
LD	Loading degree determination	Beladungsgrad-ermittlung
LDC	limited dynamic conditions	begrenzte Dynamik
LDP	list of data points	Stützstellen einer Eingangs-variablen einer Tabelle.
LDPM	repeatedly used list of data points	Stützsellen einer Eingangs-variablen einer Tabelle, die mehrfach verwendet wird
LE	left	links
LEN	Length	Länge
LEVEL	level	Niveau
LFC	Long fuel circulation	großer Kraftstoffkreislauf
LGD	Low gear detection	Low Gang Erkennung
LGRD	Limitating gradient	Änderungs-begrenzung
LGT	longitudinal	
LID	local identifier	
LIH	Limp home	Notlauf
LIM	Limitation	Begrenzung
LINP	logical input	
LOAD	load	Last
LOADPWM	Load output	Tastverhältnis Lastsignal
LOC	local variable	lokale Variable
LOCK	lock	sperren
LOIL	Oil level	Ölniveau
LOILPWM	Oil level PWM	Tastverhältnis Ölniveau
LOSS	loss	Verlust
LOST	lost	verloren
LOUT	logical output	
LPF	Low pass filter	Tiefpaßfilter
LS	Lambda sensor	Lambdasonde
LSCL	Lamda sensor close loop	geschlossener Kreis
LSH	Lambda sensor heater	Lambdasondenheizung
LSHPWM	Pulse width modulation of lambda sensor heater	Tastverhältnis Lambdasondenheizung
LSL	linear lambda sensor	lineare Lambdasonde

Keyword	English definition	German definition
LSOL	Lambda sensor open loop	offener Kreis
LST	last	zuletzt, letzter
LTC	load transient correction	Lastschlagdämpfung
LV	Logic variables data	logische Variable
MAF	Mass air flow	Luftmasse
MAFM	Mass air flow meter	Luftmassenmesser
MAIN	Main	hauptsächlich
MAN	Manual	von Hand
MAP	Manifold air pressure	Saugrohrdruck
MASK	mask	maskieren, verbergen
MASS	mass	Masse
MAT	mathematical	mathematisch
MAX	Maximum	Maximum
MC	main controller	
MCL	missing cap lamp	Diagnoselampe für fehlenden Tankdeckel
MCPS	magnetic crank shaft position sensor	induktiver KW Sensor
MDL	model	Modell
MEC	mechanic	mechanisch
MEM	Memorise	Speicher
MES	measured	gemessen
MF	Fuel mass (diesel)	Einspritzmasse
MFF	mass fuel flow	Kraftstoffmassenstrom
MIL	Malfunction indication light	Diagnose-Fehler, -Lampe
MILM	Malfunction indication light memory	Fehlerspeicher für Anzeigelampe
MIN	Minimum	Minimum
MIS	Misfire	Verbrennungsaussetzer
MISS	missing	fehlen
MMV	Moving mean value	gleitender Mittelwert
MOD	Mode	Modus
MON	monitoring	überwachen
MON2	copy of process monitoring	Kopie der Prozeßüberwachung
MOSE	model series	Baureihe
MOT	motor	Motor
MOVE	movement	Bewegung
MPG	mapping	
MPH	miles per hour	Meilen / Stunde
MPL	Multiple	mehrfach
MSG	message	
MSR	Deceleration slip control	Motorschlepp-momentenregelung
MST	master	Master
MSW	multifunctional steering wheel	Multifunktionslenkrad
MT	Manual shifted transmission	Handschaltgetriebe
MTC	Main throttle control	Motordrosselklappe
MTCPWM	PWM-signal of MTC	Tastverhältnis MDK
MU	monitoring unit	Kontroll Einheit

Keyword	English definition	German definition
MUX	Multiplexer	Multiplexer
MV	Mean value	Mittelwert
MWSS	magnetic wheel speed sensor	induktiver Raddrehzahlsensor
N	Engine speed	Motordrehzahl
NACK	no acknowledge	keine Bestätigung
NAME	name	Name
NAT	Natural	natürlich
NC	no calibrated data	vom Applikationssystem nicht änderbare Daten
NEG	Negative	negativ
NEUT	neutral	neutral
NEW	new	neu
NF	Natural frequency	natural frequency
NL	Noise level	Geräuschwert
NLC	non calibratible logical constant	nicht applizierbare logische Konstante
NOISE	noise	Rauschen
NOT	Not ...	nicht
NOX	NOX	NOX
NR	Number	Nummer
NT	NOX trap	NOX Speicherkat.
NTL	NOX trap loading	NOX Speicherkatbeladung
NTLD	NOX trap loading degree	NOX Speicherkat-beladungsgrad
NULL	null	Null
NVMY	non volatile memory	nichtflüchtiger Speicher
O2	oxygen	Sauerstoff
O2L	oxygen loading	Sauerstoffbeladung
OBD	On board diagnosis	On board diagnosis
OBS	observe	beobachten
OC	Open circuit (hard.)	offener Schaltkreis
OCT	Octal power stage (ATM 39)	
OFF	Off	aus
OFS	Offset	Offset
OIL	oil	Öl
OK	ok	ok (bestanden, erledigt)
OLD	old	alt
ON	On	ein
OP	Opening period (diesel)	Offnungsdauer
OPEN	open	offen
OPL	Open loop	offener Regelkreis
OPM	operating mode	Betriebsart
OPP	operating mode	Betreibspunkt / Arbeitspunkt
OPT	optional	optional
OPTM	optimum	optimal
ORNG	out of range	außerhalb des Bereichs
OS	operating system	
OSC	Oscillate	oszillieren

Keyword	English definition	German definition
OUT	output	Ausgang
OV	output value	
OVB	overboost	
OVER	over	über
OVF	overflow	
OVS	oversteering	übersteuern
P	Proportional part of PID	proportionaler Anteil
PAR	parity	Parität
PARK	park	parken
PAS	Passive	passiv
PAT	Pattern	Muster
PCD	pressure control device	Druckregler
PCTL	pre-control	Vorsteuerung
PCU	Pump control unit (diesel)	Pumpensteuergerät
PCV	high pressure pump pressure control valve	
PEAK	peak	
PER	Period	Periode
PERC	per cent	Prozent
PGAS	gas pressure	Gasdruck
PHA	Phase	Phase
PHY	physical	physikalisch
PIA	parallel interface adapter	
PIN	pin	Pin
PL	Part load	Teillast
PLAUS	plausible	plausibel
PM	Puls modulator	Phasengeber
PN	Park neutral	Park neutral
POIL	Oilpressure	Öldruck
PONCLOFA	Power on to closure offset angle (diesel)	Magnetventil-Anzugswinkel
POP	post operating phase	Rechnernachlauf
PORT	port	Durchgang, Öffnung
PORTPWM	Port PWM	Tastverhältnis Drallklappe
POS	Positive	positiv
POST	Post	nach
POTI	Potentiometer	Potentiometer
POW	power	Leistung
PQ	Pressure quotient	Druckquotient
PR	positive response	
PRED	predicted	prädiziert
PREL	preliminary	vorläufig
PRES	Present	momentan
PREV	Previous	vor, vorab
PRG	progression	Progression
PRI	priority	Vorrang
PRIM	primary	primär
PRJ	project	Projekt

Keyword	English definition	German definition
PROG	programming	programmieren
PROP	Proportional	proportional
PROT	protection	Schutz
PRS	Pressure	Druck
PS	Pump shaft (diesel)	Pumpenwelle
PSI	Flow rate	Durchflußwert
PSN	position	Position
PSTE	Power steering	Servolenkung
PTR	pointer	Zeiger
PU	Pull	Schub
PUB	Pull begin	Schubbeginn
PUC	Pull fuel cutoff	Schubabschaltung
PUE	Pull end	Schubende
PULS	pulsation	Pulsation
PUR	Pull reduction	Schubreduzierung
PV	Pedal value	Pedalwert
PVS	Pedal value sensor (diesel)	Pedalwert
PVSL	permanent vehicle speed limiter	permanenter Geschw.-begrenzer
PWL	power latch	Hauptrelais
PWM	Pulse width modulation	Tastverhältnis
PWR	Power supply	Stromversorgung
QC	quantified charge	normierte Füllung
QF	Fuel quantity (diesel)	Einspritzmenge
QGAS	gas quality	Gasqualität
QOIL	Oil quality	Ölqualität
QOILPWM	Oil quality PWM	Tastverhältnis Ölqualität
QPS	Quad power stage (ATM 36 oder ATM 38)	
QUO	quotient	Quotient
R	Resistance	Widerstand
RA	general gas constant	allg. Gaskonstante
RAA	run away avoidance	Hochlaufunterdrückung
RAE	reactivation acceleration enrichment	Wiedereinsetzen BA
RAF	ratio of air fuel	Luftverhältnis
RAM	RAM	RAM
RAMP	ramp	Rampe
RAS	Radiator shutter	Kühlerjalousie
RATE	rate	Rate
RATIO	Ratio	Verhältnis
RAW	raw value	Rohwert
RCV	receive	erhalten, empfangen
RD	read	lesen
RE	rear	hinten
REAC	reactivation	Wiedereinsetzen
READY	readiness identification	Bereitschaftserkennung
REC	record	aufzeichnen
RED	Reduction	Reduktion

Keyword	English definition	German definition
REF	Reference	Referenz
REFU	refuel	betanken
REINF	reinforce	
REL	Relative	relativ
REQ	Request	Anforderung
RESI	residual	Rest
RESP	response	Antwort
REST	Restart	Wiederholstart
RESU	resume	wiederaufnehmen
REV	Revolution	Umdrehung
RFL	Reactivation Full Load	Wiedereinsetzen Vollast
RFP	recirculation flap	Umluftklappe
RFPPWM	recirculation flap PWM	Tastverhältnis Umluftklappe
RFU	Returnless fuel system	
RG	Residual exhaust gas	Restgas
RGL	regular	normal
RGN	regeneration	Regenerierung
RHO	Density	Dichte
RI	right	rechts
RIS	Reactivation Idle Speed	Wiedereinsetzen Leerlauf
RISE	rising	steigend
RLS	release	Freigabe
RLV	relevant	relevant
RLY	Relay	Relais
RND	random	Zufall
RNG	range	Bereich
ROLL	Rolling	Rollen
ROM	read only memory	
RON	research octane number	
RPG	reprogrammation	
RPL	Reactivation Part Load	Wiedereinsetzen Teillast
RPU	Reactivation Pull	Wiedereinsetzen Schub
RR	Rough road	Schlechtwegstrecke
RS	reed switch	Reed switch
RST	Reset	rücksetzen
RT	runtime	Laufzeit
RTD	Retard	spät
RUL	Running losses	running losses
RUN	Runing (working)	laufen
RVL	Reversing light	Rückfahrlicht
RWUP	Rich warm up	fetter Warmlauf
S	stratified	geschichtet
SA	Secondary air	Sekundärluft
SAF	secondary air flow	Sekundärluft-massenstrom
SAM	Siemens Application and Measurement System	

Keyword	English definition	German definition
SAMPLE	sample	Sample
SAMS	SAMS	SAMS
SAP	Secondary air pump	Sekundärluftpumpe
SAPPWM	secondary air pump PWM	
SAS	Safety Cut Off	Sicherheits-schubabschalten
SAV	Secondary air valve	Sekundärluftventil
SAVE	save	speichern
SAVPWM	Secondary air valve PWM	Tastverhältnis Sekundärluftventil
SAWUP	secondary air warm up	
SC	supercharged	aufgeladen
SCA	Scale	Skala
SCAN	scan	
SCB	short circuit brush	Schleiferkurzschluß
SCC	Single cylinder cut off (MSR)	Einzelzylinder-abschaltung
SCG	Short circuit ground	Masseschluß
SCHA	supercharger	mechanischer Lader
SCP	Short circuit plus	auf Betriebsspannung
SDL	scheduler	Verteiler
SDR	status diagnostic result	Status Diagnoseergebnis
SDWN	Shift down	zurückschalten
SECU	Security	Sicherheit
SEED	Seed	Zugangsberechtigung
SEG	Segment	Segment
SEL	selective	selektiv
SEM	successive error management	Folgefehlerbetrachtung
SENS	Sensor	Sensor
SEQ	Sequential	sequentiell
SER	serie	Serie
SERVO	servo	Servo
SET	Set	einstellen
SF	status flag	Status Flag
SFC	Short fuel circulation	kleiner Kraftstoffkreislauf
SFT	sensor frequency time	Sensorfrequenzzeit
SHD	Shield	Schirmung
SHIFT	shift	Verschiebung
SIF	seriel interface	serielle Schnittstelle
SIG	signal	Signal
SIM	Simulation	Simulation
SLAVE	Slave	
SLIP	Slip	Schlupf
SLM	Smoke limitation (diesel)	Rauchdichtebegrenzung
SLOP	slope	Steigung
SLOW	Slow	langsam
SLV	solenoid valve	Magnetventil
SND	send	

Keyword	English definition	German definition
SO2	sulphur dioxide	Schwefeldioxid
SO2P	desulfuration	Entschwefelung
SOA	start of air	Start Lufteinblasung
SOI	Start of injection	Einspritzbeginnwinkel
SOIL	Oil sensor signal	Ölsensorsignal
SOP	Start of pumping (diesel)	Förderbeginnwinkel
SOV	Shut off valve of evap function	Aktivkohlefilterabsperrventil
SP	Setpoint	Sollwert
SPARE	Spare	Reserve
SPI	Serial peripheral interface	
SPR	spring	Feder
SRC	Source	Quelle
ST	Starting	Start
STAB	stability	
STALL	Stalling	Motor abwürgen
STAT	stationary	stationär
STATE	State	Zustand
STB	Standby	Standby
STC	statistic	statistisch
STC	storage capacity	Speicherkapazität
STD	standard deviation	Standardabweichung
STE	Start enable	Startfreigabe
STEP	Step	Stufe
STK	stroke	Hub
STND	Standardization	Normierung
STOP	stop	stop
STP	Stepper motor	Schrittmotor
STR	starter relais	Anlasser Relais
SUB	Substitude	Ersatzgröße
SUBT	Subtractive	subtraktive Korrektur
SUM	Sum	Anzahl
SUP	Shift up	hochschalten
SUPP	suppress	unterdrücken
SUSP	suspicion	Verdacht
SVL	Service vehicle light	Servicelampe
SW	software	Software
SWI	Switch	Schalter
SWT	sensor switching time	Sensorsprungzeit
SYM	Symptom	Symptom
SYN	Synchro	Synchron
SYS	system	System
T	time	Zeit
TAC	After catalyst temperature	Abgastemperatur nach Katalysator
TACE	evaporative temp.air conditioning	
TAM	ambient temperature	Umgebungstemperatur
TAR	target	Ziel

Keyword	English definition	German definition
TBA	Boost air temperature (diesel)	Ladelufttemperatur
TBAT	battery temperature	Batterietemperatur
TBL	table	
TC	throttle control	
TCC	test cycle counter	Prüfintervallzähler
TCI	coolant temperature in	TKW Motoreintritt
TCO	Coolant temperature	Kühlwassertemperatur
TCOPWM	Coolant temperature PWM	Tastverhältnis Kühltemperatur
TCS	traction control system	Traktionskontrolle
TCSPWM	Traction control PWM	Tastverhältnis Traktionskontrolle
TCU	Time control unit	
TCYL	cylinder air temperature	Lufttemp. im Zylinder
TD	Time dwell	Schließzeit
TDC	Top dead center	oberer Totpunkt
TEG	Exhaust gas temperature	Abgastemperatur
TEGR	EGR temperature	Temperatur AGR
TEMP	Temperature	Temperatur
TEST	test	Test
TFU	Fuel temperature	Kraftstofftemperatur
TGAS	gas temperature	Gastemperatur
TGT	tightness	Dichtigkeit
TH	Thermostat	Thermostat
THD	Threshold	Schwellwert
THE	Theoritical	theoretisch
THEAT	heater temperature	Heizlufttemperatur
THERMO	thermo	thermisch
THM	Thermic inertia of the engine	
THR	Throttle	Drosselklappe
TI	Injection time	Einspritzzeit
TIA	Intake air temperature	Ansauglufttemperatur
TIP	tip	getastet
TIPO	Post injection time	Nacheinspritzer-Einspritzzeit
TIPR	Pre injection time	Vorabeinspritzer-Einspritzzeit
TIT	Titanium	Titan
TLDP	Tank leakage detection pump	Tankleckagediagnosepumpe
TLDV	tank leakage detection valve	Leckdiagnoseabschaltventil
TMAG	coil temperature of magnetic valve IVVT	Spulentemperatur VANOS Magnetventil
TMP	temporary	temporär
TNT	temperature Nox trap	
TOG	toggle	toggle
TOIL	Oil-Temperature	Öltemperatur
TOILPWM	Oil temperature PWM	Tastverhältnis Öltemperatur
TOL	Top limit	obere Grenze
TOOTH	Tooth	Zahn
TORS	torsion	Torsion
TOT	Total	gesamt

Keyword	English definition	German definition
TOUT	time out	
TPC	Pre catalysator temperature	Abgastemperatur vor Katalysator
TPS	throttle position sensor	Drosselklappensensor
TPSPWM	Throttle position sensor PWM	Tastverhältnis TPS
TQ	Torque	Moment
TQB	braking torque	Bremsmoment
TQD	Driving torque at clutch	Antriebsmoment an der Kupplung
TQDW	Torque at driving wheels	Moment an den Antriebsrädern
TQE	effective engine torque	effektives Motormoment
TQFR	Friction torque	Reibmoment
TQI	Indicated engine torque	Indiziertes Motormoment
TQPWM	Torque PWM	Tastverhältnis Moment
TQR	Torque reduction (MSR)	Motormomentenreduktion
TQZ	Zero driving torque	Nullmoment
TRA	Transient	instationär
TRAN	transition	Übergang
TRANS	transmission	Getriebe
TRCOM	Trip computer signal	Trip computer signal
TREAT	treatment	Behandlung
TRIG	Trigger	Triggerung
TRIP	trip related parameter	TRIP bezogener Parameter
TRM	transmit	übertragen, senden
TRO	trouble	Störung
TRT	total running time	Betriebsstundenzähler
TSCHA	air temperature supercharger	Lufttemperatur mech. Lader
TSLIP	Traction slip	Antriebsschlupf
TUR	turbine	Turbine
TURB	Turbulence	Turbulenz
TYP	type	
UP	up	vor
UPD	update	Aktualisierung
USE	Compiler switch	
V	Voltage	Spannung
VAC	vacuum	Vakuum
VAL	Valuated	gewichtet
VALUE	value	Wert
VAP	vapor	Dampf
VAR	variant	Variante
VB	Battery-voltage	Batteriespannung (KL30)
VCC	Supply voltage	Versorgungsspannung (5V)
VCV	high pressure pump volume control valve	
VEH	vehicle	Fahrzeug
VEL	Velocity	Verstellgeschwindig-keit
VEPU	Venturi pump	Saugstrahlpumpe
VERS	version	Version
VIC	Variable intake control	variabler Einlaß
VIM	Variable intake manifold	Schaltsaugrohr

Keyword	English definition	German definition
VIRT	virtual	virtuell
VLD	Valid	gültig
VLFT	valve lift	Ventilhub
VLS	Lambda sensor voltage	Lambdasonden-Spannung
VLSL	linear lambda sensor voltage	Spannung lineare Lanbdasonde
VO	Valve overlap	Ventilüberschneidung
VOL	Volume	Volumen
VP	pumping voltage	Pumpspannung
VPROG	Programming voltage	Programmier-spannung
VS	Vehicle speed	Tachosignal
VST	vehicle stopped	Fahrzeug steht
VVL	variable valve lift	variabler Ventilhub
VVSL	variable vehicle speed limiter	variabler Geschwindigkeitsbegrenzer
VVSLS	variable vehicle speed limiter switch	Schalter für variabl. Geschwindigkeitsbegr.
VVT	Variable valve timing	variable Nockenwellensteuerung, Schaltvanos
VVTPWM	PWM signal of VVT	Tastverhältnis Schaltvanos
WAIT	wait	warten
WAL	warning lamp	Warnlampe
WARN	warning	warnen
WF	Wall fllm	Wandfilm
WG	Waste gate	
WGPWM	Waste gate pulse width modulation	
WHEEL	Wheel speed	Rad
WIN	Window	Fenster
WIND	Windscreen	Windschutzscheibe
WKU	Wake up	wake up
WKUL	Wake up line	Reizleitung-Signal
WORD	word	word
WORK	work	Arbeit
WOUT	without	ohne
WR	write	schreiben
WUP	Warm up	Warmlauf
ZR	Zirconium	Zirkon

The designation of the variables used informs about the type (index table, table, constant), the input and output variables (e.g. TI, TCO etc.), the operating state (e.g. FL, PL, IS) and the dependent parameters, if any (e.g. f (engine speed)).

The designation index table (IP_) means that the index table value is obtained by interpolation in a unidimensional or multidimensional area.

A table (ID_) also can be uni- or multidimensional.
The table value is not obtained by interpolation.

The allocation of the table values to the data points is defined as follows:
A table value i applies if the input value is greater than or equal to the data point i.
Exception: The table value 1 applies in case of input values less than the data point 2.

```
                                                                                     LDP 8
                                                                           LDP 7       | 8
                                                             LDP 6           | 7        |______
                                                   LDP 5       | 6           |______
                                     LDP 4           | 5       |_____
                           LDP 3       | 4           |______
               LDP 2         | 3       |_____
Table            | 2         |_____
value 1          |_____
_______
     |
     |
  LDP 1 (=Input value  <  LDP 2)
```

Unless otherwise specified, the data points are not subject to hysteresis.

Example of nomenclature:

The term "IP_TIB__N__MAF" means that the index table value TIB (basic injection period) is obtained by a two-dimensional interpolation.

The dependence of the index table value is defined by the suffix __N__MAF :
TIB is a function of the engine speed data point and of the air mass data point, i.e. of the general engine speed and air mass data points.

The descriptions of variables and of the free adjustable data points, together with their value ranges and resolutions, are shown in the Annex.

36.5 AT Library - short description

36.5.1 General

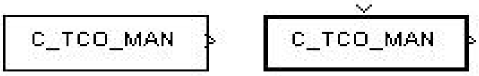
This document is a short reference to all AT Library blocks. It aims to help the reading of SDA based specifications. The complete library reference is available with the current SDA installation.

Optional inputs and outputs: Some blocks can have optional inputs and outputs, which are made available when necessary. These outputs are normally not shown in the example pictures, but described on the explanations.

Maps, curves and axis: Maps and curves are related to their axis definitions. This information can be consulted at the calibration section in the given specification.

Index base: Block inputs for indices can be one-based (standard) or zero-based (special cases). The index base is shown subscripted. For instance R0 stands for "row index, zero-based". Other examples: bit_sel0, x1, y0. See "selector" block example, on this document in chapter 1.8, page 9.

Triggered blocks: Most of the blocks have an optional trigger input. This port accepts function-call connection, which determinate when these blocks shall be calculated. Example:



36.5.2 Block versioning

The block version is shown since SDA 5.0 release. Please note the following:

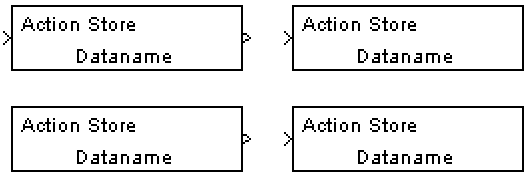
- Blocks from previous SDA versions do not show any version information.
- The version information is shown on the block icon or below it in the format "*V X.Y*".
- Functional change or relevant appearance changes reflects on version information update (only on "X"). A change in "Y" is not functional or appearance relevant. For instance, the versions 5.1, 5.4, 5.13 have all the same appearance and functional behavior.
- The part "X" of the block version number is not coupled to SDA release number. The first block version since SDA 5.0 is "5." for all libraries though.
- All functional changes done to AT Library blocks between SDA releases are shown in this document and in the according library documentation within the SDA installation.
-

36.5.3 Functional behaviour

The blocks of version 6.x have the same functional behaviour than the blocks of version 5.x. The changes in the blocks are only related to code generation and/or to adaptation to the ASAM MBFS standard.

36.5.4 Action Handling

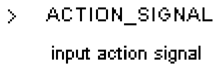
Action_Data_Store (V. 5) **Action_Data_Store_Init (V. 5),**
Read_Action_Data_Store (V. 5) **Write_Action_Data_Store (V. 5)**



These blocks are used to model the initialization and read/write access to an action internal storage that can be accessed by all instances of this action and inside the module defining the action.

If this storage is only used inside the action instance or inside the defining module it is listed as local data on the specification.

Input_Action_Signal (V. 5)



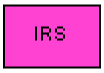
This block is used to model nested Actions (reused Action inside an Action). Here the Action Signal needed for the reused Action is fed in by an Action Input.

Input_To_Action (V. 5) **Output_From_Action (V. 5)**

The Input_To_Action block is used to model an input to an Action that is unique for all action instances and calculated outside of the module that defines the action.

The Output_From_Action block is used to model an output of an action (shared storage) and of the defining module. The value can be set by all instances of the action and inside the module defining the action.

IRS (V. 5)



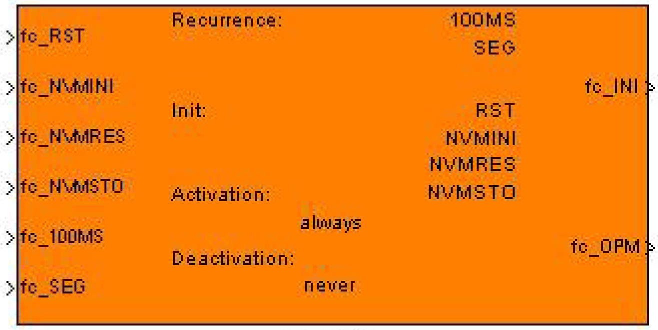
This block indicates that the according subsystem belongs to an Infrastructure Requirement Specification (IRS).

Action Parameters (V. 5.)

The blocks PAR_IN, PAR_OUT and PAR_INOUT are used to model action input, output and input-output parameters.

36.5.5 Application Condition

APP_CDN (V. 5) (V. 6)



APP_CDN

The Application Condition block displays the activation conditions for the according module. It controls when the module is activated or deactivated. All function calls triggering or initializing the functionality of a module are input to the APP_CDN block. Dependent on the activation condition the function calls are fed through the block or not. The Application Conditions are displayed in textual form within the specification.

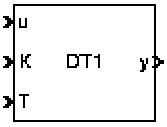
In the above example of the block the according module is always active and never inactive. The initialization is triggered by the Reset (RST) event and the Nonvolatile Memory (NVM...) events. The according function calls are output in bus of function calls fc_INI.

The operate mode is triggered by the 100ms event and the segment synchronous event. The according function calls are output in a bus of function calls fc_OPM.

36.5.6 Basic Algorithms

DT1 Element (V. 5.) (V. 6)

The output signal y is calculated from the input signal u, the amplification factor K and the time constant T according to the following functions:

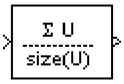


DT1_element

$$z1[n] = (u*K/T);$$
$$z2[n] = z2[n-1]+ (Teff / (T+Teff))*(z1[n] - z2[n-1]);$$
$$y[n] = z1[n] - z2[n];$$

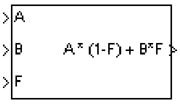
T	Time Constant
u	Input signal
K	Amplification factor

mean value (V. 5.) (V. 6)



This block outputs mean value of elements of a single input vector.

fading (V. 5.) (V. 6)



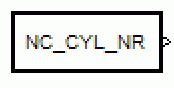
This block implements the fading calculation:

Inputs: A, B, F.

Output: Out=A*(1-F)+B*F.

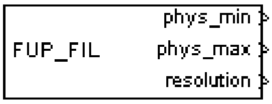
36.5.7 Calibration_Data_Handling

Constant (V. 5.) (V. 6)



Outputs the value specified by a constant or parameter, which can be a scalar or array. The example's output gives the value contained in the constant NC_CYL_NR (a configuration data).

from_data_definition (V. 5.) (V. 6)



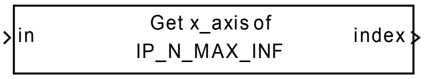
This block reads chosen fields from data definition of specified data (e.g. FUP_FIL) and gives them as outputs. The possible fields to output: minimal physical value, maximal physical value, resolution and states (only for state variables).

Get Curve Index (V. 5.)

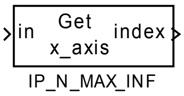
This block outputs the next smallest index from the input axis value. It considers the axis of the selected curve. Indexing starts from zero.

In	Axis value on the axis of the curve
Out	Index of the axis value on the axis of the curve

Get Map Index (V. 5.) (V. 6)

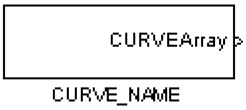


This block outputs the next smallest index of the input axis value on the selected axis (x_axis or y_axis) of the selected map. Indexing starts from zero.



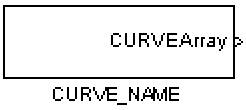
in	Axis value of the specified axis in a map
index	Index of the axis value on the specified axis of the map

getCurveArray (V. 5.)



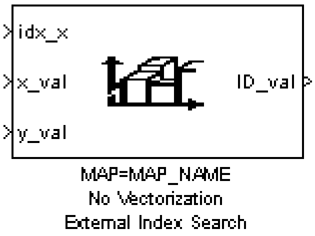
Generates a signal which carries the values from the selected calibration data (an array of curves).

getMapArray (V. 5.) (V. 6)



Generates a signal which carries the values from the selected calibration data (an array of maps).

id_lookup (V. 5.) (V. 6)



This block implements the lookup operation over a non interpolated map or array of maps. A single map has one dimension (calibration curve) or two dimensions (calibration map).

The name of the map is specified by "MAP".

The block supports 3 different modes:

- "local": The map is read from base workspace and only the look up value is exported via the output
- "exported": The map is read from base workspace and the look up value together with the complete map array is exported.
- "imported": The map is read from an inport, as an array signal.

The block outputs the lookup value "ID_val" connected with the next smaller index to the axis values "x_val" and "y_val". There is no interpolation or extrapolation between axis values.

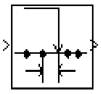
With the option "External Index Search", the search for the axis indices can be done externally (see index_search block). The ports "x_val" and "y_val" are named "x_frac" and "y_frac". In this case, the values on "x_frac" and "y_frac" are directly used as index and fraction values for the map lookup.

The ports "idx_x", "idx_y" and "idx_z" specify the indices selecting a map to be read from the given map array, if an array of maps is used. Their availability as input ports depends on the array dimensions, which can be consulted at the calibration section in the given specification.

For access to an array of maps, the block can be set to:

- "Scalar vectorization" mode: All maps in the array are read with the same inputs for "x_val" and "y_val". The output is a vector carrying the lookup values.
- "Array vectorization" mode: All maps in the array are read with separate values from the inputs "x_val" and "y_val", which are arrays. The output is a vector carrying the lookup values.
- "No vectorization" mode: The lookup operation is calculated for one map which is selected through the inputs "idx_x", "idx_y" and "idx_z". Values for "x_val" and "y_val" are than scalar, as also the output lookup value.

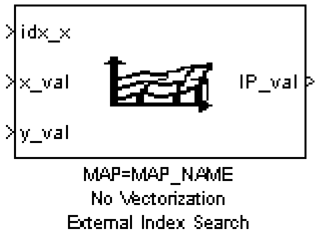
index_search (V. 5.) (V. 6)



The index-search block calculates the zero-based indices and interval fractions for the input value for a selected axis.

The block generates a pair of outputs for each input value by calculating the zero-based index of the breakpoint set element that is less than or equal to the input value and the resulting fractional value (*f* were: 0 <= *f* < 1). This represents the input value's normalized position between the index and the next index value for in-range input.

ip_lookup (V. 5.) (V. 6)



This block is analogous to the id_lookup block. The difference is the output lookup value, which is interpolated.

IP_val is calculated by a linear interpolation through the map values. There is linear interpolation between axis values.

The name of the map is specified by "MAP".

The block supports 3 different modes:

- "local": The map is read from base workspace and only the look up value is exported via the output
- "exported": The map is read from base workspace and the look up value together with the complete map array is exported.
- "imported": The map is read from an inport, as an array signal.

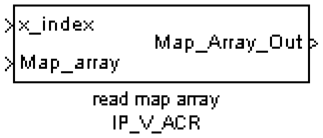
With the option "External Index Search", the search for the axis indices can be done externally (see index_search block). The ports "x_val" and "y_val" are named "x_frac" and "y_frac". In this case, the values on "x_frac" and "y_frac" are directly used as index and fraction values for the map interpolation.

The ports "idx_x", "idx_y" and "idx_z" specify the indices selecting a map to be read from the given map array, if an array of maps is used. Their availability as input ports depends on the array dimensions, which can be consulted at the calibration section in the given specification.

For access to an array of maps, the block can be set to:

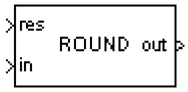
- "Scalar vectorization" mode: All maps in the array are read with the same inputs for "x_val" and "y_val". The output is a vector carrying the lookup values.
- "Array vectorization" mode: All maps in the array are read with separate values from the inputs "x_val" and "y_val", which are arrays. The output is a vector carrying the lookup values.
- "No vectorization" mode: The lookup operation is calculated for one map which is selected through the inputs "idx_x", "idx_y" and "idx_z". Values for "x_val" and "y_val" are than scalar, as also the output lookup value.

read_Map_Array (V. 5.) (V. 6)



This block selects and output the map specified by the input indexes "x_index", "y_index" and "z_index" as a signal. Not shown indexes have standard value "1". The input is an array of maps.

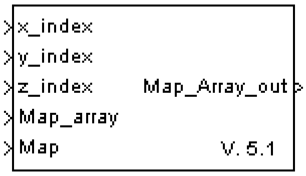
to_resolution (V. 5.) (V. 6)



This block provides the possibility to reduce data with a higher resolution to a lower one. Three types of reduction are possible: truncation (ceils or floor) and round-up to a given resolution. The chosen conversion method is shown in the block icon: ROUND, CEIL or FLOOR. The resolution value can be obtained from the optional input port "res" or given as a block annotation.

write Map Array (V. 5.) (V. 6)

Outputs a map array signal "Map_Array_out" which is a copy from the input array signal "Map_Array", where the single map specified by the input indexes "x_index", "y_index" and "z_index" is substituted by the values of the input "Map". Not shown indexes have standard value "1".

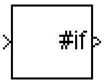


write map array
MAP_NAME

36.5.8 Control_Structures

Compiler switch if (V. 5.) (V. 6) **compiler switch (V. 5.) (V. 6)**

These blocks are used to model compiler switches.



compiler_switch_if : If condition is true then output function call '#if'.

compiler_switch: If condition is true then output function call '#if' else output function call '#else'

do while loop

do while loop (V. 5.) **while loop (V. 5.)**

While-Control Structure and Do-While-Control Structure:

As long as condition is true the function call "while" is cast. The only difference between the blocks is when the condition is checked for the first time. The while_loop block checks the condition before it performs first function call. do_while_loop performs the function call at first and only after that checks the condition. The optional output 'i' indicates the current number of execution.

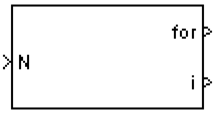
while loop

To omit an endless loop the while is only executed as long as the number of iterations is smaller than the defined maximum (default 500, for model simulation).

for loop (V. 5.)

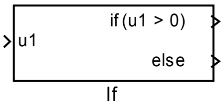
For-Loop-Control Structure:

Executes for-loops function-calls, where the optional output "i" runs from 0 (or 1) to N. The output "for" is a function call signal triggering the according subsystem that contains the functionality within the for loop.



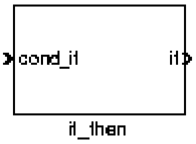
for i = 1 to N

If (V. 6)



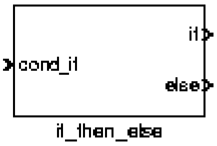
If...Then...Else-Control Structure: If condition 'u1' is true then output action call 'if' else output action call 'else'. In difference to the following if-then-else blocks this block outputs SL action calls instead of function calls.

If...Then (V. 5.) (V. 6)



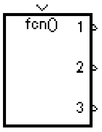
If...Then-control structure: If condition 'cond_if' is true then output function-call 'if'.

If...Then...Else (V. 5.) (V. 6)



If...Then...Else-Control Structure: If condition 'cond_if' is true then output function call 'if' else output function call 'else'.

sequencer (V. 5.) (V. 6)

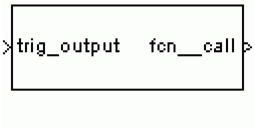


This block generates an ordered sequence of function calls. The number of output ports is variable. The user has a possibility to select in the block mask between three operation variants:

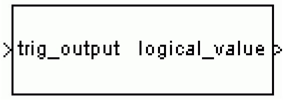
- "not triggered" - the output function calls are generated when the block is calculated;
- "function call triggered" - the output function calls are generated when the block is triggered. In this case block has an extra function call trigger input;
- "logical triggered" - the block has an extra input port "trigg". The output function calls are generated when this input has a value different than zero.

Triggeroutport 2 fcn call (V. 5)

This block generates a function call if the input is different to zero. The block can be used in connection with a trigger block output.



Triggeroutport 2 logical (V. 5)

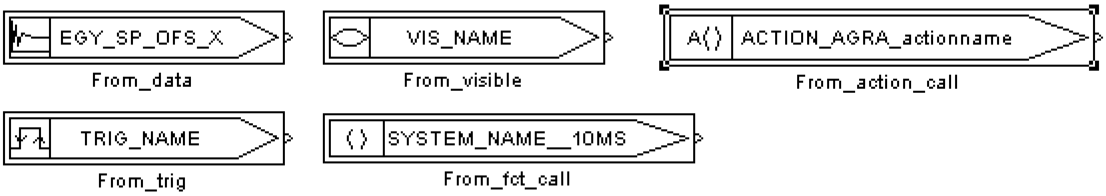


Conversion of a trigger signal into a logical value. This block generates a logical 1 if the input is different to zero. The output is zero otherwise. The block can be used in connection with a trigger block output.

36.5.9 Signal Types

FROM_DATA (V. 5) **FROM_FCT_CALL (V. 5)** **FROM_TRIG (V. 5)**

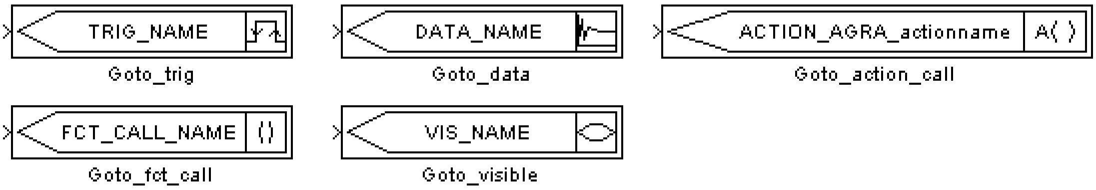
FROM_VISIBLE (V. 5) **FROM_ACTION_CALL (V. 5)**



From_data, From_fct_call, From_trig or From_visible is a From block with scope global for the use with global data, function call, trigger or visible signals only. This From blocks get~~s~~ the signal from a corresponding Goto block.

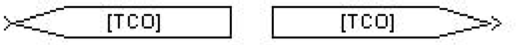
GOTO_DATA (V. 5.) **GOTO_FCT_CALL (V. 5.)** **GOTO_TRIG (V. 5.)**

GOTO_VISIBLE (V. 5.) **GOTO_ACTION_CALL (V. 5.)**



Goto_data, Goto_fct_call, Goto_trig or Goto_visible is a Goto block with scope global for the use with global data, function call, trigger or visible signals only. The signal is sent to a corresponding From block.

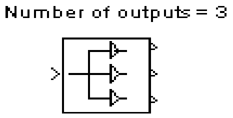
GOTO **FROM**

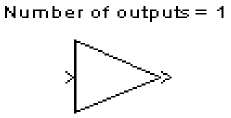


Goto and From blocks with scope local. The blocks represent a signal line between the Goto and the From block. In the example the signal TCO is submitted between the blocks.

36.5.10 Signals

branch (V. 5.) (V. 6)

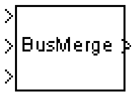




This block creates one or more copies from the input signal, which can than be directly used as input to a merge block.

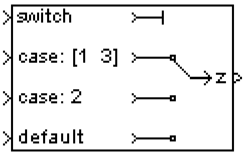
Input signal can be a scalar, a vector or a bus. In case input signal is a bus, every output signal is also a bus with the same bus structure as the input.

BusMerge (V. 5.) (V. 6)



This block constructs a bus output holding the union of all elements of the input busses in a flat bus structure. Flat structure means that the exported bus has only one hierarchy level. Signals that are input on more than one of the input busses are merged separately before being added to the output bus.

case multiport switch (V. 5.) (V. 6)



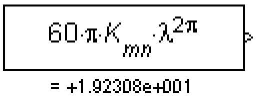
The block connects one of the "case" input ports (or default port) to the output port.

It receives a value on the input port "switch", which is compared to case conditions that determine which of the other input ports is connected to the output port.

If the value on the control port "switch" does not match any case condition, the "default" input is connected to the output.

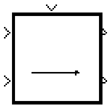
In the example picture, if the value of "switch" is 1 or 3, the port "case: [1 3]" is connected to the output. If its value is 2, than the port "case: 2". Else, the default port is used.

constant value (V. 5.) (V. 6)



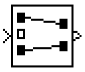
Outputs a constant value with the possibility to show the icon as formula expression. The calculated value is shown on the block annotation.

FeedThrough (V. 5.) (V. 6)



The FeedThrough block outputs the input values if the function-call occurs. The output maintains its last changed value. The number of input and output ports is variable.

Selector (V. 5.) (V. 6)

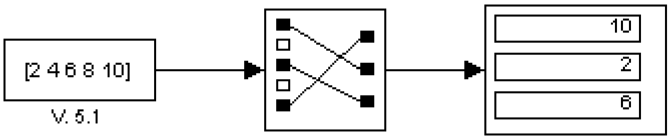


The Selector block generates as output selected elements of an input vector or matrix. It accepts either vector or matrix signals as input. The block icon changes to reflect the type of input used. The way the block determines the elements to select differs slightly, depending on the type of input.

Vector Input

If the input type is vector, a Selector block outputs a vector of selected elements. The block determines the indices of the elements to select either from the block's Elements parameter or from an external signal.

The elements to be selected must be specified as a vector unless only one element is being selected. For example, the following model shows the Selector block icon and the output for an input vector of [2 4 6 8 10] and an Elements parameter value of [5 1 3].



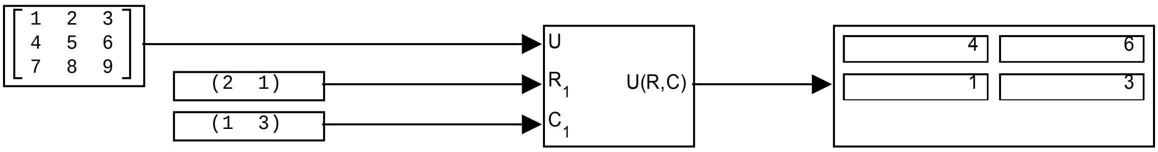
The block icon displays the ordering of input vector elements graphically if the block icon is large enough. If external source for element indices is used, the block adds an input port for the element indices signal.

Matrix Input

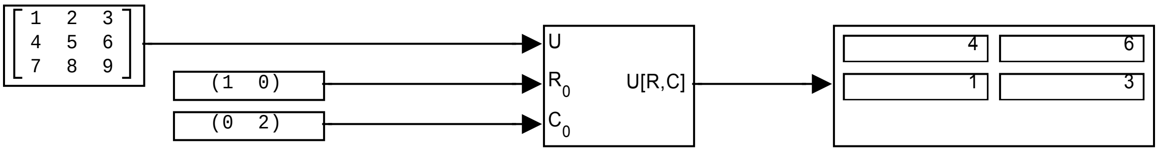
If the input type is matrix, the Selector block outputs a matrix of elements selected from the input matrix. The block determines the row and column indices of the elements to select either from its Rows and Columns parameters or from external signals (input ports "R", for row, and "C" for column selection).

The indices of the row and columns to be selected must be specified as vectors (or a scalar if only one row or column is to be selected). For example, the Rows expression [2 1] and the Columns expression [1 3] specifies output of a 2x2 matrix whose first row contains the first and third elements of the input matrix's second row and whose second row contains the first and third elements of the input matrix's first row.

Using one-based indexing:

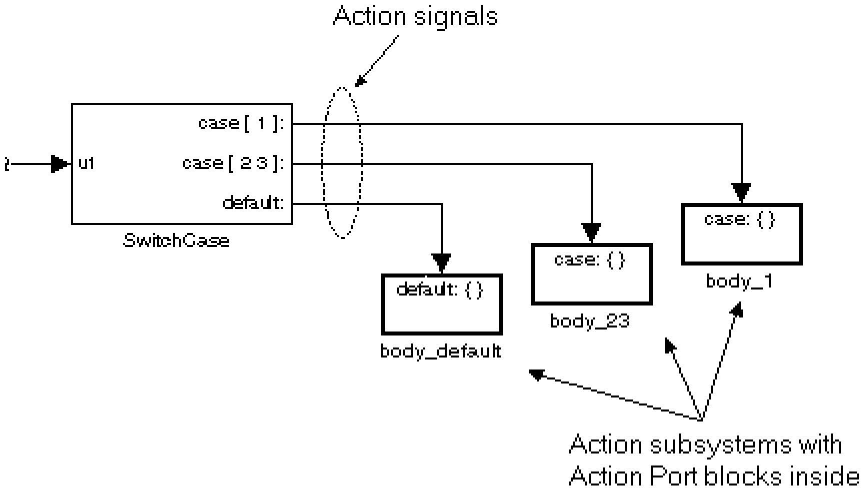


Using zero-based indexing:



Switch Case (V. 5.) (V. 6)

The following shows a completed Simulink C-like switch control flow statement in the subsystem of the SwitchCase block.



As shown in the above example, a Switch_Case block receives a single input, which it uses to form case conditions that determine which Action subsystem to execute. Each output port case condition is attached to an Action subsystem. The cases are evaluated topdown starting with the top case. If a case value (in brackets) corresponds to the actual value of the input, its Action subsystem is executed.

The above switch control flow statement can be represented by the following pseudo code:

```
switch (u1) {
    case [u1=1]:
        body_1;
        break;
    case [u1=2 or u1=3]:
        body_23;
        break;
    default:
        bodydefault;
}
```

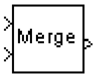
Note: As demonstrated in the pseudo code example above, cases for the SwitchCase block contain an implied break after their Action subsystem is executed. There is no "fall through" behaviour for the Simulink switch control flow statement as found in standard C switch statements.

Note: The naming "Action" in this block is not connected to "Actions" defined in the SV Action concept.

logical value (V. 5.) (V. 6)

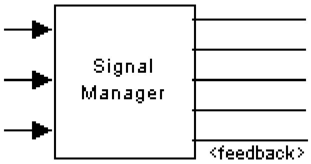
Block outputs logical TRUE or FALSE.

Merge (V. 5.) (V. 6)



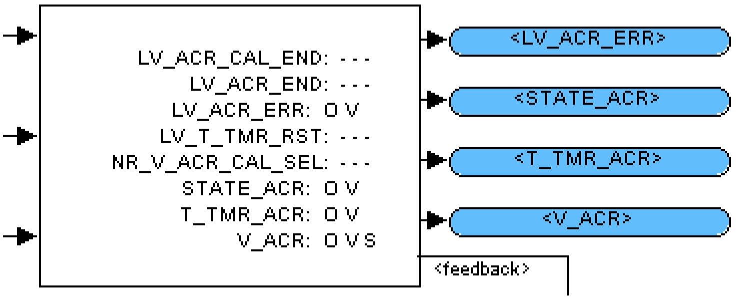
The Merge block combines its inputs into a single output line whose value at any time is equal to the most recently computed output of its driving blocks. The Merge block often is used to merge a calculation signal with a reset signal.

SignalManager (V. 5.) (V. 6)



This block merges signals contained in the input busses. It supports two or more input busses. Optional configurations:

- Output with "feedback" bus, which contains all merged signals. The feedback bus is routed back to the operate subsystem.
- "Show signal names on the mask". Where:



- Outports represent the module outputs;

"O": module output;

"V": visible variable;

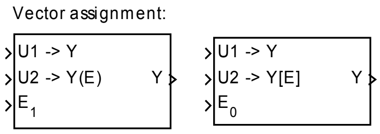
"S": NVMY data;

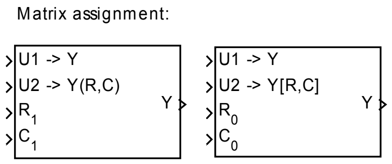
"---": Local variable

Note: Only local signals that are present on one of the input busses to this block are displayed on the block mask

36.5.11 Vectors and Matrices

Assignment (V. 5.) (V. 6)





The Assignment block assigns values from input U2 to selected elements of array input U1, producing the array output Y. Equivalent operation:

Vector assignment:

```
Y = U1;
Y[E] = U2;
```

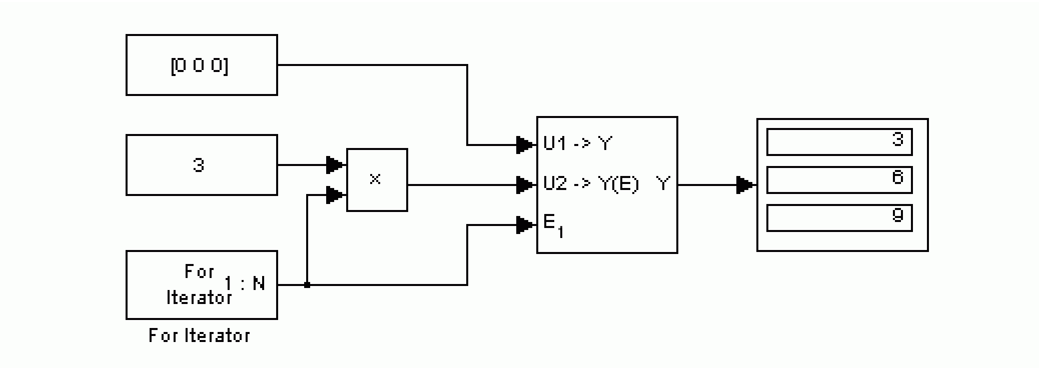
Matrix assignment:

```
Y = U1;
Y[R,C] = U2;
```

On a vector assignment, the input E is used to select the assignment position. On a matrix assignment, the inputs R (row) and C (column) are used to select the assignment position. The assignment position can be a scalar, a vector or a matrix, depending on the dimension of E, or the combined dimensions of R and C.

Example: iterated Assignment

The Assignment block can be used to assign values computed in an iterator (For or While) loop to a vector or matrix signal. The iterator block generates the indices required by the Assignment block. Two iterator blocks are needed to compute row and column indices separately. For example, the following model uses a For block to create a vector signal each of whose elements equals 3*i where i is the index of the assignment element.



Dec2Bin (V. 5.) (V. 6) Bin2Dec (V. 5.) (V. 6)

These blocks convert an integer into a binary number and vice versa. The number of bits can be set to 8, 16, 32 or 64. For decimal numbers the digits after the comma will be cut off. The binary numbers are displayed with leading highest exponent: [2^n, ...,2^1,2^0].

Important: these blocks suffered functional modification from version 3.1 to 4.0. See below.

Pseudo-code since SDA 4.0

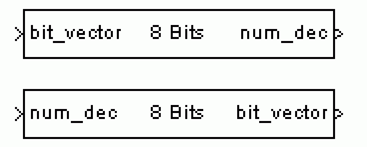
```
Dec2Bin:
for ( i=0; i<no_bits; i++) {
    bit_vector[ no_bits - i] =
                      num_dec.i;
}
```

Pseudo-code for SDA 3.1

```
Dec2Bin:
for ( i=0; i<8; i++) {
    8bit_vector[i+1] = num_dec.i;
}

Bin2Dec:
```

Bin2Dec:

```
num_dec = 0;
for ( i=0; i<no_bits; i++) {
    num_dec = num_dec +
              bit_vector[no_bits-i]
              * 2^i;
}
```



First element in bit vector is most significant bit. E.g. **10000000 is equal to 128**

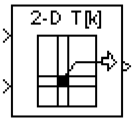
```
num_dec = 0;
for ( i=0; i<8; i++) {
    num_dec = num_dec +
              8bit_vector[i+1]
              * 2^i;
}
```

Dec2Bin

Bin2Dec

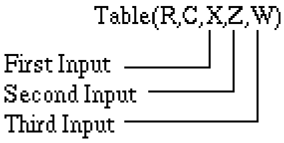
First element in bit vector is less significant bit. E.g. **10000000 is equal to 1**

DirectLookUp (V. 5.) (V. 6)



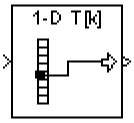
Index into an N-dimensional table to retrieve a scalar, vector or 2-D matrix. The Direct Look-Up Table (n-D) block uses its block inputs as zero-based indices into an n-D table. The number of inputs varies with the shape of the output desired. The output can be a scalar, a vector, or a 2-D matrix. The look-up table uses **zero-based** indexing.

A set of output values can be defined as the Table data parameter. The output can be a scalar, a vector or a 2-D matrix. The first input specifies the zero-based index to the first dimension higher than the number of dimensions in the output, the second input specifies the index to the next table dimension, and so on, as shown by this figure:

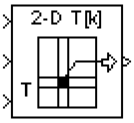


The figure shows a 5-D table with an output shape set to "2-D Matrix"; the output is a 2-D Matrix with R rows and C columns.

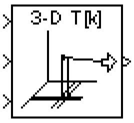
This figure shows the set of all the different icons that the Direct Look-Up Table block shows (depending on which options you choose in the block's dialog box).



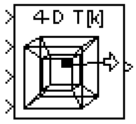
Direct Look-Up Table (n-D)



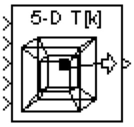
Direct Look-Up Table (n-D)3



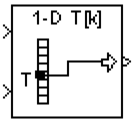
Direct Look-Up Table (n-D)6



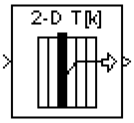
Direct Look-Up Table (n-D)9



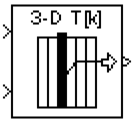
Direct Look-Up Table (n-D)12



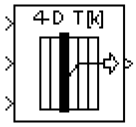
Direct Look-Up Table (n-D)1



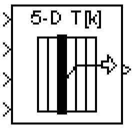
Direct Look-Up Table (n-D)4



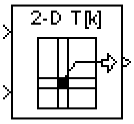
Direct Look-Up Table (n-D)7



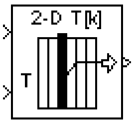
Direct Look-Up Table (n-D)10



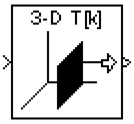
Direct Look-Up Table (n-D)13



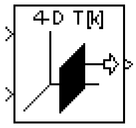
Direct Look-Up Table (n-D)2



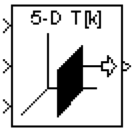
Direct Look-Up Table (n-D)5



Direct Look-Up Table (n-D)8



Direct Look-Up Table (n-D)11



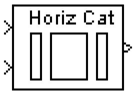
Direct Look-Up Table (n-D)14

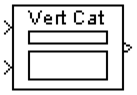
With dimensions higher than 4, the icon matches the 4-D icons, but will show the exact number of dimensions in the top text, e.g., "8-D T[k]." The top row of icons is used when the block output is made from one or more single-element lookups on the table. The blocks labelled "n-D Direct Table Lookup5," 6, 8 and 12 are configured to extract a column from the table and the two blocks ending in 7 and 9 are extracting a plane from the table. Blocks in the figure ending in 10, 11 and 12 are configured to have the table be an input instead of a parameter.

get_bit (V. 5.) (V. 6)

This block gets the values of specific bits of the decimal number obtained from the input port "num_dec". The output can be either single value or vector depending on the input given through "bit_sel" input port. The input "bit_sel" can be a single value or a vector. It selects the bits used in this operation. Bits are selected by its position number from the less significant bit (position 0 or 1) to the most significant bit (7, 15, 31, 63 or 8, 16, 32, 64). The user has possibility to select between zero or one based indexing for "bit_sel".

MatrixConcatenation (V. 5.) (V. 6)





The MatrixConcatenation block concatenates input matrices u_1, u_2, ..., u_n along rows or columns, where n is specified by the number of input ports.

Horizontal Matrix Concatenation

In the horizontal concatenation mode, the block concatenates the input matrices along rows.

y = [u1 u2 u3 ... un] % Equivalent MATLAB code

For horizontal concatenation, inputs must all have the same row dimension, M, but may have different column dimensions. The output matrix has dimension M-by- Ni, where Ni is the number of columns in input ui (i = 1, 2, ..., n). When some of the inputs are length-M 1-D vectors while others are M-by-Ni matrices, the vector inputs are treated as M-by-1 matrices.

Vertical Matrix Concatenation

In the vertical concatenation mode, the block concatenates the input matrices along columns.

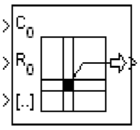
y = [u1;u2;u3;...;un]   % Equivalent MATLAB code

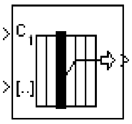
For vertical concatenation, inputs must all have the same column dimension, N, but may have different row dimensions. The output matrix has dimension Mi-by-N, where Mi is the number of rows in input ui (i = 1, 2, ..., n). When some of the inputs are length-Mi 1-D vectors while others are Mi-by-1 matrices, the vector inputs are treated as Mi-by-1 matrices. (1-D vector inputs are not accepted for vertical concatenation when the other inputs have column dimension greater than 1.)

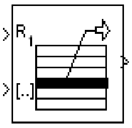
1-D Vector Concatenation

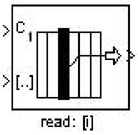
When all inputs to the Matrix Concatenation block are length-Mi 1-D vectors, the output is a Mi-by-1 matrix containing all input elements concatenated in port order: the elements in the vector input to the top port appear as the first elements in the output, and the elements in the vector input to the bottom port appear as the last elements in the output.

read_array (V. 5.) (V. 6)









This block reads elements (values, rows or columns) from input array. There are three possible modes:

- "read single value" - block reads the single value A(R,C) from the input array A at the given positions C (columns) and R (rows).
- "read row(s)" - block reads row(s) A(R) from the input array A at the given position(s) R.
- "read column(s)" - block reads column(s) A(C) from the input array A at the given position(s) C.

In the modes "read row(s)" and "read column(s)" it is possible to give a vector as an input for coordinates R or C. So it is possible to get more than one row or column in the output at once.

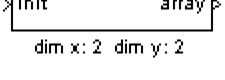
Each coordinate R and C can be obtained from the corresponding input port or as a block parameter. The user has a possibility to select between zero or one based indexing for coordinates R and C, indicated as subscribed 0 or 1 (e.g. C_0 or C_1).

Bus support: The input [..] can be either an array or a bus of arrays. If the input signal is a bus, the output signal is a bus with the same structure as the input. Every element of the bus signal gets the suffix specified below the block (read: [i]). The forth block shows the annotation "read: [i]" which means that every signal name in the output bus gets the suffix [i] to indicate that the i-th element of the array is selected.

set_array (V. 5.) (V. 6)

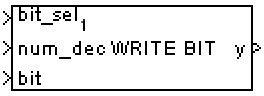
Reduce annotations 'on'

Reduce annotations 'off'



This block outputs the array filled with the given initial value. The initial value can be obtained from the block input ('external source') or can be entered as the block parameter ('internal source'). The size of the output array can be entered directly as the block parameter or can be obtained from the existent data definition.

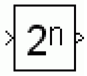
set_bit (V. 5.) (V. 6)



The block sets or resets the bits of decimal number "dec_num" chosen through input port "bit_sel" depending on the value of "bit". User can specify "bit" value either in the parameter "Bit value" or as an input port. The "bit" input port value can be a vector. In this case "bit" vector length should be the same as in "bit_sel" vector.

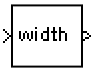
The decimal value can be obtained from the input port "num_dec" or it can be stored in internal memory and initialized by zero. The user has the possibility to select between zero or one based indexing for "bit_sel", indicated as subscribed 0 or 1 (e.g. bit_sel_0 or bit_sel_1).

Two_power_n (V. 5.) (V. 6)



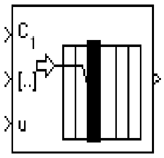
This block calculates 2 to the power of n. The input n has to be an integer. Otherwise the digits after the comma are cut off (floor[n]).

Width (V. 5) (V. 6)



Output the width of the input signal. If an the input is an array of size 3 the output is 3.

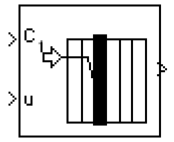
write_array (V. 5.) (V. 6)



This block writes elements (values, rows or columns) into input array. It is possible to select between zero or one based indexing for coordinates.

There are three possible operation modes:

- "write single value" - block writes the single value A(R,C) into the input array A at the given positions R (columns) and C (rows).
- "write row(s)" - block writes row(s) A(R) into the input array A at the given position(s) R.
- "write column(s)" - block writes column(s) A(C) into the input array A at the given position(s) C.



size: [NC_CYL_NR]

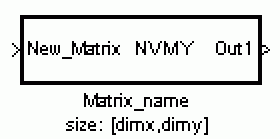
In the modes "write row(s)" and "write column(s)" it is possible to give a vector as an input for coordinates R or C. So it is possible to write more than one row or column in the output array at once.

The input row or column in "write row(s)" or "write column(s)" mode should either be scalar (single value) or have the same length as the according dimension of the input array.

It is possible to obtain the input array A from an input port or from an internal memory. When block uses internal memory to store the array A the size of this array is shown as block annotation like in the second block example. The size of the array in this case is NC_CYL_NR. The dimension can also be obtained from an existent data definition.

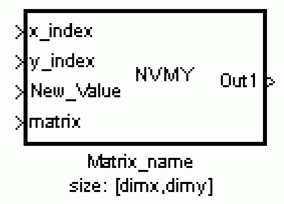
Bus support: The input "[..]" can be either an array or a bus of arrays. When using a bus of arrays, the input "u" can be a single signal or a bus of signals. When both inputs are buses, there is a correspondence between signals with the same names: the signal "X_i" from "u" is used to write on the array "X" from "[..]".

Write Matrix (V. 5.) (V. 6)



The block writes a Matrix 'Matrix_name' directly in the Workspace. This block is used if data in the simulation has to be saved in the nonvolatile memory NVMY (e.g. at PWL_OFF). In the data definition table of the specification the matrix name is set to Mode S. The parameters 'Matrix_name', number of rows 'dimx Rows' and number of columns 'dimy Col' are shown in the block icon.

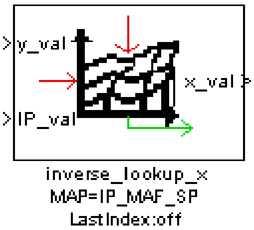
Write To Matrix (V. 5.)



The Block writes a new value to a given position in a 2-dimensional Matrix 'MatrixName' not only in the Simulation but also directly in the Workspace. The new value is given by 'New_Value' and the position in the matrix is given by the indices 'x_index' and 'y_index' (Indexing starts from zero).

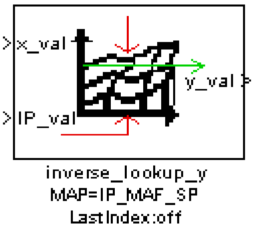
This block can be used if data in the simulation has to be saved in the nonvolatile memory NVMY (e.g. at PWL_OFF). In the specification the matrix name is set to Mode S. The parameters 'Matrix_name', number of rows 'dimx Rows' and number of columns 'dimy Col' are shown in the block icon.

inverse_lookup_x (V. 5.) (V. 6)



This block extracts the value on the x_axis for the map given by Mapname based on the input of the actual map value and the value for the y-axis.

inverse_lookup_y (V. 5.) (V. 6)

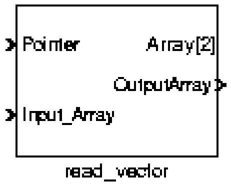


This block extracts the value on the y_axis for the map given by Mapname based on the input of the actual map value and the value for the x-axis.

36.5.12 SDA 4.0 and 3.1 specific blocks

Note: The version information is not available on block icon for blocks from SDA 4.0 and 3.1 releases.

Read Vector

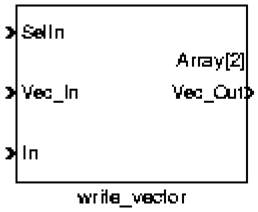


Description

This block returns the value of the array element at the position selected by 'Pointer'. Indexing in this block starts from 1. The size of the array can be directly given in the 'Size of Array' parameter or by a variable set in Workspace.

Pointer	Position of the selected element in the input vector (array index)
Input_Array	input vector
OutputArray	Selected element of the input vector
Array[NC_...]	Size of array

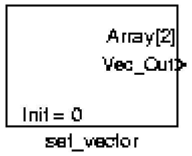
Write Vector



Writes a given value to the selected position in the vector. The block writes a given value to the selected position in the vector. The value is given by 'In' and the position in the vector by 'SelIn' (Indexing starts from 1).The dimension of the array is specified by the 'Size of Array' parameter.

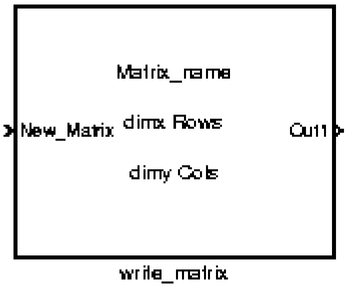
SelIn	Index of the vector position
Vec_In	Input vector
In	Value that is written to vector position 'SelIn'
Vec_Out	Output vector
Array[NC_...]	Size of array

Set Vector



The block initializes all values in a vector with the value 'Init'. The size of the vector can be given directly by the 'Size of Array' parameter or by a variable set in Workspace. 'Vec_Out' outputs the initialized vector.

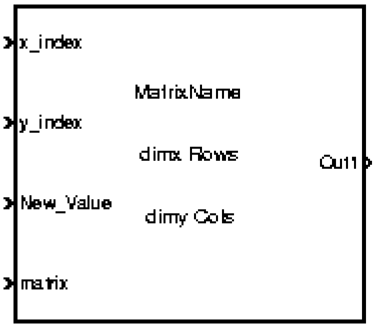
Write Matrix



The block writes a Matrix 'Matrix_name' directly in the Workspace. This block can be used if data in the simulation has to be saved in the nonvolatile memory NVMY (e.g. at PWL_OFF). In the specification the matrix name is set to Mode S. The parameters 'Matrix_name', number of rows 'dimx Rows' and number of columns 'dimy Col' are shown in the block icon.

New_Matrix	Matrix Elements
Out	2-dimensional Matrix, written in the Workspace

Write To Matrix



write_to_matrix

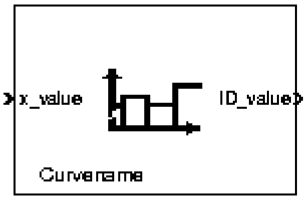
The Block writes a new value to a given position in a 2-dimensional Matrix 'MatrixName' not only in the Simulation but also directly in the Workspace. The new value is given by 'New_Value' and the position in the matrix is given by the indices 'x_index' and 'y_index' (Indexing starts from zero).

This block can be used if data in the simulation has to be saved in the nonvolatile memory NVMY (e.g. at PWL_OFF). In the specification the matrix name is set to Mode S. The parameters 'Matrix_name', number of rows 'dimx Rows' and number of columns 'dimy Col' are shown in the block icon.

x_index	x-index of the matrix position
y_index	y-index of the matrix position
matrix	Input matrix
New_Value	Value that is written to matrix position 'x_index' and 'y_index'
Out	Output matrix

Lookup ID Curve

Looks up an ID_Curve (No interpolation). Outputs the 'ID_value' connected with the next smallest index to the axis value 'x_value'. There is no interpolation between axis values. The names of the axis are given by list_curve in the Workspace. There is also no extrapolation outside the axes values.



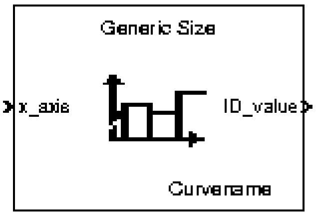
lookup_id_curve

triggered (optional)

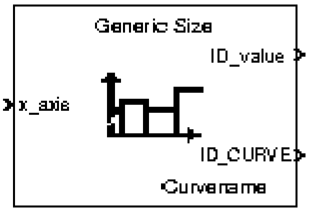
x_value	Input axis value
ID_value	Output curve value connected with the next smallest index to the axis value

Lookup ID Curve Array

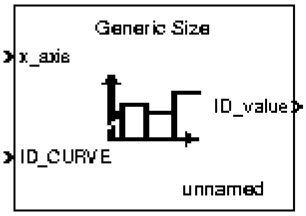
Looks up an ID_Curve_array (No interpolation). [triggered (optional)]



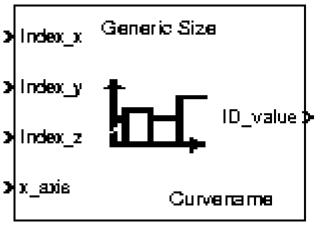
lookup_id_curve_array



lookup_id_curve_array1

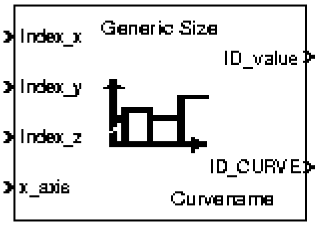


lookup_id_curve_array2



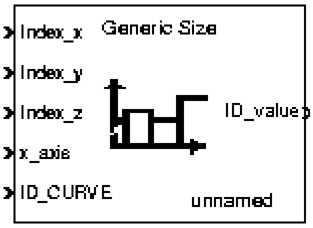
lookup_id_curve_array

local



lookup_id_curve_array

exported



lookup_id_curve_array

imported

Modes

Mode local: 'Curvename' is defined within the block in block parameter 'ID_CURVE'.

Mode exported: 'Curvename' is defined within the block in block parameter 'ID_CURVE' and is exported via an output (if lookup table is input to another specification).

Mode imported: 'Curvename' is imported via an inport (if lookup table is defined in another specification).

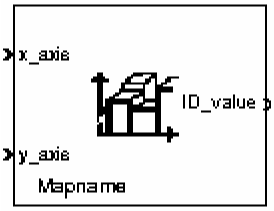
Description

Outputs the 'ID_map' value connected with the next smallest index to the axis value 'x_axis'. There is no interpolation between axis values. Index_x, Index_y and Index_z specify the indices of the ID_Curve in the curve array. The names of the axes and the size of the curve array are given by list_curve in the Workspace. There is no extrapolation outside the axes values.

x_axis	Input x-axis value
ID_map	Output curve value connected with the next smallest index to the x-axis value
Index_x	Input: x-index of the curve in the curve array (optional)
Index_y	Input: y-index of the curve in the curve array (optional)
Index_z	Input: z-index of the curve in the curve array (optional)
ID_CURVE	Input: imports the name of the curve (Mode: imported) (optional)
ID_CURVE	Ouput: exports the name of the curve (Mode: exported) (optional)

Lookup ID Map

Looks up an ID_Map (No interpolation).



lookup_id_map

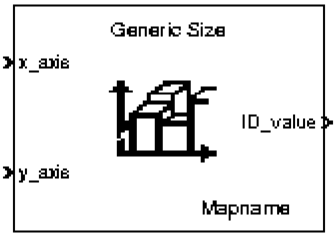
triggered (optional)

Outputs the 'ID_map' value connected with the next smallest index to the axis values 'x_axis' and 'y_axis'. There is no interpolation between axis values. The names of the axes are given by list_map in the Workspace. There is no extrapolation outside the axes values.

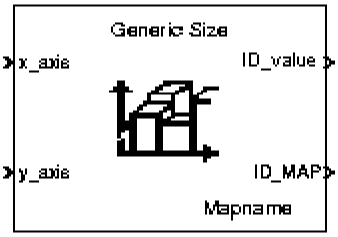
x_axis	Input: x-axis value
y_axis	Input: y-axis value
ID_map	Output: Map value connected with the next smallest index to the x- and y-axis values

Lookup ID Map Array

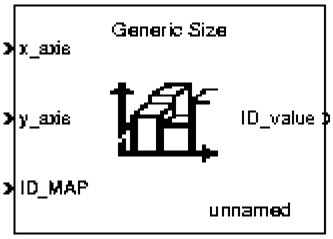
Looks up an ID_Map_array (No interpolation). [triggered (optional)]



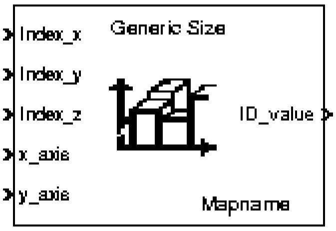
lookup_id_map_array



lookup_id_map_array1

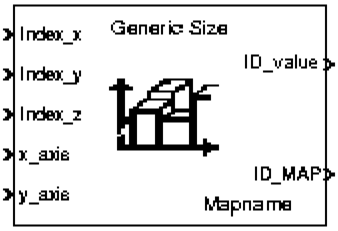


lookup_id_map_array2



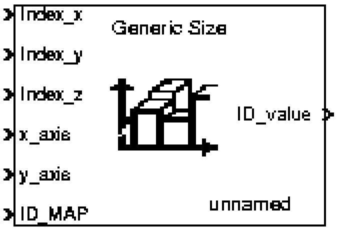
lookup_id_map_array

local



lookup_id_map_array

exported



lookup_id_map_array

imported

Modes

Mode local: 'Mapname' is defined within the block in block parameter 'ID_Map'.

Mode exported: 'Mapname' is defined within the block in block parameter 'ID_Map' and is exported via an outport (if lookup table is input to another specification) .

Mode imported: 'Mapname' is imported via an inport (if lookup table is defined in another specification).

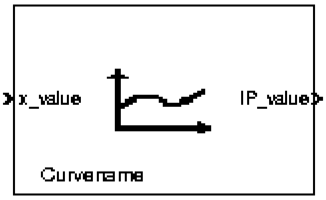
Description

Outputs the 'ID_map' value connected with the next smallest index to the axis values 'x_axis' and 'y_axis'. There is no interpolation between axis values. Index_x, Index_y and Index_z spezify the indices of the ID_Map in the Map array. The names of the axis and the size of the map array are given by list_map in the Workspace. There is no extrapolation outside the axes values.

x_axis	Input x-axis value
y_axis	Input y-axis value
ID_map	Map value connected with the next smallest index to the x-axis and y-axis value
Index_x	x-index of the map in the map array (optional)
Index_y	y-index of the map in the map array (optional)
Index_z	z-index of the map in the map array (optional)
ID_MAP	Imports the name of the map (Mode: imported) (optional)
ID_MAP	Exports the name of the map (Mode: exported) (optional)

Lookup IP Curve

Looks up an IP_Curve (With interpolation).



lookup_ip_curve

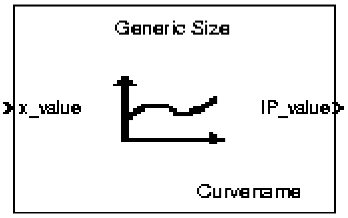
triggered (optional)

Outputs the 'IP_value' connected with the linear interpolated index of the axis value 'x_value'. There is a linear interpolation between axis values. The names of the axes are given by list_curve in the Workspace. There is no extrapolation outside the axes values.

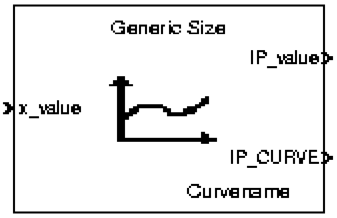
x_axis	Input x-axis value
IP_value	Value obtained interpolating the curve

Lookup IP Curve Array

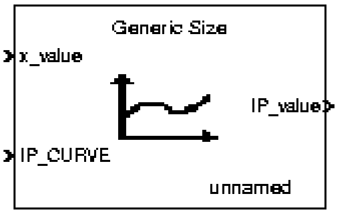
Looks up an IP_Curve_array (With interpolation). [triggered (optional)]



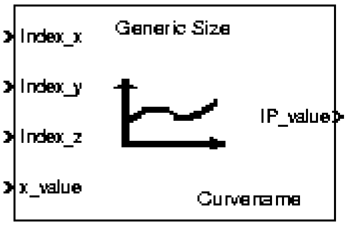
lookup_ip_curve_array



lookup_ip_curve_array1

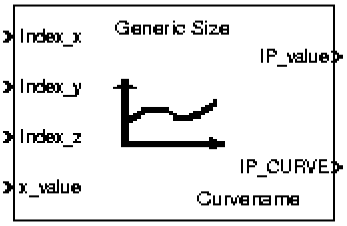


lookup_ip_curve_array2



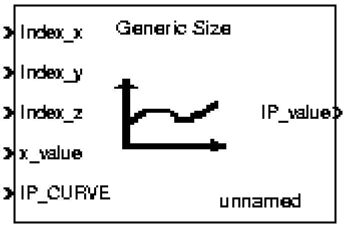
lookup_ip_curve_array

local



lookup_ip_curve_array

exported



lookup_ip_curve_array

imported

Modes

Mode local: 'Curvename' is defined within the block in block parameter 'ID_CURVE'.

Mode exported: 'Curvename' is defined within the block in block parameter 'ID_CURVE and is exported via an outport (if lookup table is input to another specification).

Mode imported: 'Curvename' is imported via an inport (if lookup table is defined in another specification).

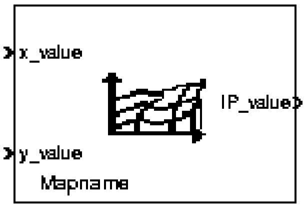
Description

Outputs the 'IP_value' connected with the linear interpolated index to the axis value 'x_value'. There is a linear interpolation between axis values. Index_x, Index_y and Index_z spezify the indices of the IP_Curve in the curve array. The names of the axis and the size of the curve array are given by list_curve in the Workspace. There is no extrapolation outside the axes values.

x_axis	Input x-axis value
IP_value	Value obtained interpolating the curve
Index_x	Input: x-index of the curve in the curve array
Index_y	Input: y-index of the curve in the curve array
Index_z	Input: z-index of the curve in the curve array
IP_CURVE	Imports the name of the curve (Mode: imported)
IP_CURVE	Exports the name of the curve (Mode: exported)

Lookup IP Map

Looks up an IP_Map (With interpolation).



lookup_ip_map

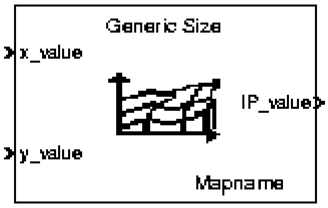
triggered (optional)

Outputs the 'IP_value' connected with the linear interpolated index of the axis values 'x_value' and 'y_value'. There is a linear interpolation between axis values. The names of the axis are given by list_map in the Workspace. There is no extrapolation outside the axes values.

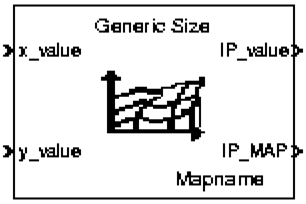
x_value	x-axis value
y_value	y-axis value
IP_value	Map value connected with the linear interpolated indeces to the x- and y-axis values

Lookup IP Map Array

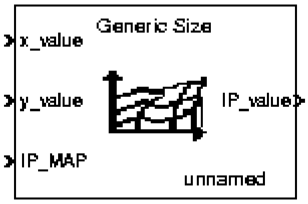
Looks up an IP_Map_array (With interpolation).[triggered (optional)]



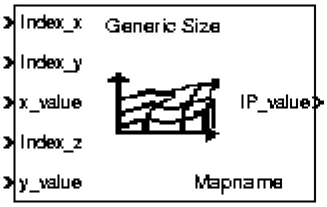
lookup_ip_map_array



lookup_ip_map_array2

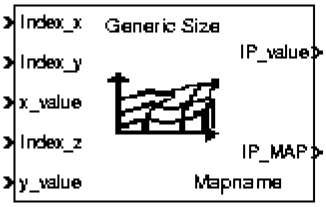


lookup_ip_map_array3



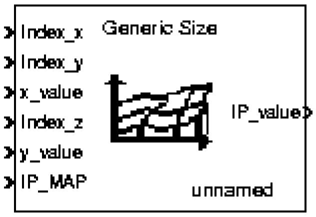
lookup_ip_map_array

local



lookup_ip_map_array

exported



lookup_ip_map_array

imported

Modes

Mode local: 'Mapname' is defined within the block in block parameter 'IP_Map'.

Mode exported: 'Mapname' is defined within the block in block parameter 'IP_Map' and is exported via an outport (if lookup table is input to another specification).

Mode imported: 'Mapname' is imported via an inport (if lookup table is defined in another specification).

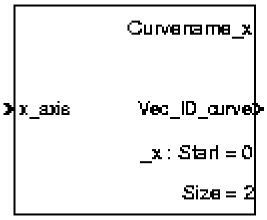
Description

Outputs the 'IP_value' connected with the linear interpolated index to the axis values 'x_value' and 'y_value'. There is linear interpolation between axis values, but there is no extrapolation outside the axes values. 'Index_x',' Index_y' and 'Index_z' specify the indices of the IP_Map in the map array. The names of the axes and the size of the map array are given by list_map in the Workspace.

x_value	x-axis value
y_value	y-axis value
IP_value	Map value connected with the linear interpolated index to the x-axis and y-axis value
Index_x	x-index of the map in the map array
Index_y	y-index of the map in the map array
Index_z	z-index of the map in the map array
IP_MAP	Imports the name of the map (Mode: imported)
IP_MAP	Exports the name of the map (Mode: exported)

Lookup Multiple ID Curve

Looks up multiple ID_Curves (No interpolation).



lookup_multiple_id_curve

triggered (optional)

If the parameter 'Curvename' contains one of the tokens (_x, _i, _j), this is replaced automatically by a given range from 1 to 'Size' or 0 to ('Size'-1). The number of curves accessed is given by 'Size'.

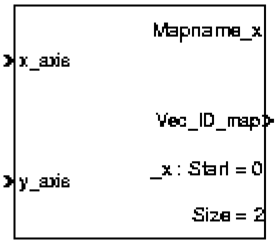
Outputs a vector of length 'Size' of ID_values 'Vec_ID_curve' connected with the next smallest index of the axis values 'x_value' of the different curves.

There is no interpolation between the axis values and no extrapolation outside the axes values. The axis names are given by list_curve in the Workspace.

x_axis	Axis value
Vec_ID_curve	Vector of Curve values with dimension 'Size' connected with the next smallest index of the x-axis value

Lookup Multiple ID Map

Looks up multiple ID_Maps (No interpolation).



lookup_multiple_id_map

triggered (optional)

If the parameter 'Curvename' contains one of the tokens (_x, _i, _j), this is replaced automatically by a given range from 1 to 'Size' or 0 to ('Size'-1). The number of maps accessed is then given by 'Size'.

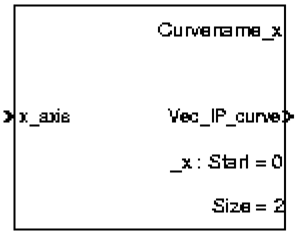
Outputs a vector of length 'Size' of ID_values 'Vec_ID_map' connected with the next smallest indeces to the axis values 'x_axis' and 'y_axis'. There is no interpolation between the axis values.

The axis names are given by list_curve in the Workspace. There is no extrapolation outside the axes values.

x_axis	x-axis value
y_axis	y-axis value
Vec_ID_map	Vector of Curve values connected with the next smallest index of the x-axis and y-axis value

Lookup Multiple IP Curve

Looks up multiple IP_Curves (With interpolation).



lookup_multiple_ip_curve

triggered (optional)

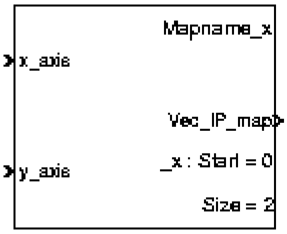
If the parameter 'Curvename' contains one of the tokens (_x, _i, _j), this is replaced automatically by a given range from 1 to 'Size' or 0 to ('Size'-1). The number of curves accessed is then given by 'Size'.

Outputs a vector of length 'Size' of IP_values 'Vec_IP_curve' connected with the linear interpolated index of the axis value 'x_axis'. There is linear interpolation between the axis values and no extrapolation outside the axes values. The axis names are given by list_curve in the Workspace.

x_axis	Axis value
Vec_IP_curve	Vector of Curve values connected with the linear interpolated index to the x-axis value

Lookup Multiple IP Map

Looks up multiple IP_Maps (With interpolation).



lookup_multiple_ip_map

triggered (optional)

If the parameter 'Curvename' contains one of the tokens (_x, _i, _j), this is replaced automatically by a given range from 1 to 'Size' or 0 to ('Size'-1). The number of maps accessed is then given by 'Size'.

Outputs a vector of length 'Size' of IP_values 'Vec_IP_map' connected with the linear interpolated index to the axis values 'x_axis' and 'y_axis'.

There is linear interpolation between the axis values, but there is no extrapolation outside the axes values. The axis names are given by list_map in the Workspace.

x_axis	x-axis value
y_axis	y-axis value
Vec_IP_map	Curve value connected with the linear interpolated index to the x-axis and y-axis value

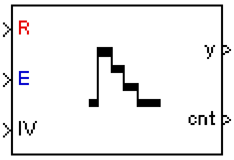
36.6 AT Basic Blocks - short description

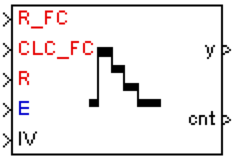
36.6.1 Remarks for block version V 5.x

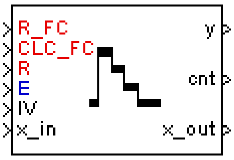
36.6.2 Multi-rate and single-rate block versions

Some blocks have two versions: multi-rate and single-rate. The multi-rate version (with R_FC and CLC_FC inputs) is used in case of system resets have to be simulated. R_FC and CLC_FC are shortcuts for "reset function call" and "calculate function call" respectively.

Various blocks have also optional inputs and outputs, which are chosen according to the model specific needs. The following picture shows an example of a block in its both versions, single-rate and multi-rate. The third picture also shows optional inputs and outputs (internal state):







For each block the functionality is described by an "Init-code" (executed at the start of simulation) and pseudo code (for methods run(), finalize(), pass(), out() and reset()) executed during simulation).

The block functionality, given with the pseudo-code and init-code, remains the same in both versions, multi-rate and single-rate.

In the multi-rate version, the block is only calculated if a function-call triggers the input R_FC or CLC_FC. Triggering the block through the input R_FC simulates a system reset. The calculation done when a R_FC triggers the block is shown in the given pseudo-code "R_FC". The block outputs are not calculated in the R_FC code. Triggering the block through the input CLC_FC runs the execution of this block, calculating a new output value. This calculation is shown in the given pseudo-code for each block.

In the single-rate version, the block calculation is executed every time the subsystem containing the block is triggered. This means, that the block calculation occurs as for every other non triggered block or subsystem. The order of calculation in the single-rate mode is exactly the same as for CLC_FC.

Besides the system reset (R_FC function call) the blocks also implement a logical reset (functional reset). There are two versions of the logical reset which can occur in single-rate mode or in multi-rate mode when CLC_FC triggers the block:

- *State based*: Reset occurs as long as signal R equals 1 (blocks with the suffix _RE)
- *Edge based*: Reset occurs on signal RT rising edge (blocks with the suffix _RTE)

Unlike reset via the R_FC function call, when the logical reset occurs, the block calculates its outputs.

While no logical reset occurs in single-rate mode or while CLC_FC function-call in multi-rate mode, the block computations are defined by the value of the Enable signal E. While E is not equal to zero, the block performs the necessary computation and updates its internal state(s). While E is equal to zero, the internal states are not updated and the output of the block is based on the previously stored value of its internal state.

CLC_FC function uses the following sub-functions:

general specification

- reset() initializes internal state(s) with initial value(s) (IV, IV1, IV2...)
- run() performs general block computations while no logical reset is occurred and the enable signal E is not equal to zero
- finalize() performs general block computations while the logical reset is occurred or the enable signal E is not equal to zero
- pass() performs necessary computations while no logical reset is occurred and enable signal E is equal to zero.
- out() assigns values to output signal

The current implementation of these blocks is described with the following C-pseudo code:

State detecting reset blocks (-RE blocks)

```
function R_FC ( )
{
 reset();
}
```

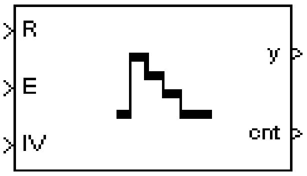
```
function CLC_FC ( )
{
 if (R) {
   reset();
   finalize();
 }
 else if (E) {
   run();
   finalize();
 }
 else {
   pass();
 }
 out();
}
```

Edge detecting reset blocks (-RTE blocks)

```
function R_FC ( )
{
 reset();
 R_1 = 0;
}
```

```
function CLC_FC ( )
{
 if (R & !R_1) {
   reset();
   finalize();
 }
 else if (E) {
   run()
   finalize();
 }
 else {
   pass();
 }
  out();
 R_1 = R;
}
```

In the description later on, for these kinds of blocks, the C-like code for each of these sub-functions is given beside the block icon as shown in the example below:



```
Variables:
  E: logic
  R: logic
  IV: real
  y: logic
  x: int
  cnt: int
```

```
Init-code:
  x = 0;

reset ( )
  x = round(IV);

run ( )
    if (x > 0) {
      x = x - 1;
    }
```

```
finalize( )

pass ( )

out ( )
  y = (x>0);
  cnt = x;
```

The additional inputs and outputs used to set and get internal states have the suffixes "_in" and "_out" after the referred state name respectively. These inputs and outputs are optional.

The following list shows the standard input and output names, with their corresponding descriptions:

Inputs

Port	Description
E	Enable: suppresses the block calculation while 0
R	Resets the block states to their initial values while 1
RT	Resets the block states to their initial values on rising edge 0 -->1
IV	Initial value set upon reset
R_FC	System reset (multirate version)
CLC_FC	System calculation request (multirate version)
dT	Sample time
MX	Upper state limit
MN	Lower state limit
x_in	Direct input of an internal state if the block has one internal state
x1_in	Direct input of an internal state 1 if the block has 2 internal states
x2_in	Direct input of an internal state 2 if the block has 2 internal states

Outputs

Port	Description
B_max	True if the upper limitation active
B_min	True if the lower limitation active
x_out	Direct output of an internal state if the block has one internal state
x1_out	Direct output of a internal state 1 if the block has 2 internal states
x2_out	Direct output of a internal state 2 if the block has 2 internal states
y	Out
cnt	Internal counter state

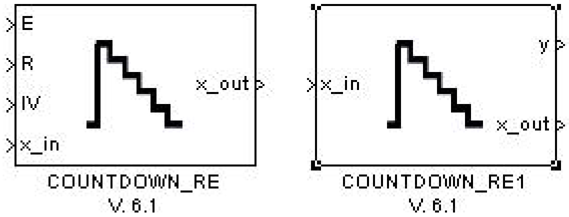
36.6.3 Remarks for block version V 6.x

36.6.4 Functional behaviour

The blocks of version 6.x have the same functional single rate behaviour than the blocks of version 5.x. The changes in the blocks are only related to code generation and to adaptation to the ASAM MBFS standard.

36.6.5 Single-rate block version

Version 6 blocks do not support the multi-rate option. Many blocks have optional inputs and outputs, which are chosen according to the model specific needs. The following picture shows an example for block configuration:



In the single-rate version, the block calculation is executed every time the subsystem containing the block is triggered. This means, that the block calculation occurs as for every other non triggered block or subsystem.

While no logical reset occurs in single-rate mode, the block computations are defined by the value of the Enable signal E. While E is not equal to zero, the block is calculated and updates its internal state(s). While E is equal to zero, the internal states are not updated and the output of the block is based on the previously stored value of its internal state.

x_in and x_out interface

The inputs x_in and outputs x_out are used to export and import the internal state x of the blocks. The internal state is exported in order to be able to initialize the state at system reset via the SignalManager on top module level. This system reset handling replaces the multirate option of the blocks.

36.6.5.1 Input/Output port details for V6 blocks:

Inputs

Port	Description
E	Enable: suppresses the block calculation while 0
R	Resets the block states to their initial values while 1
RT	Resets the block states to their initial values on rising edge 0 -->1
IV	Initial value set upon reset
dT	Sample time
MX	Upper state limit
MN	Lower state limit
x_in	Direct input of an internal state if the block has one internal state
x1_in	Direct input of an internal state 1 if the block has 2 internal states
x2_in	Direct input of an internal state 2 if the block has 2 internal states

Outputs

Port	Description
B_max	True if the upper limitation active
B_min	True if the lower limitation active
x_out	Direct output of an internal state if the block has one internal state

Port	Description
x1_out	Direct output of a internal state 1 if the block has 2 internal states
x2_out	Direct output of a internal state 2 if the block has 2 internal states
y	Output

36.6.6 Directly triggered blocks

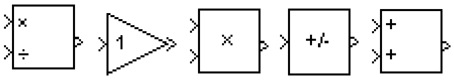
The blocks without internal states (and, as a result, without reset) have the 'Triggered block' parameter. This is a checkbox which can be 'on' or 'off'. When this option is 'off' the calculation for these blocks is executed every time the subsystem, which they are placed in, is triggered. When this option is 'on', the additional control input port is added at the top of these blocks. This port accepts function-call connection (as CLC_FC does) and these blocks are calculated only when a function-call triggers the control port.

36.6.7 Block versioning

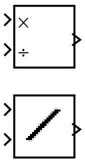
The block version is shown since SDA 5.0 release. Please note the following:

- Blocks from previous SDA versions do not show any version information.
- The version information is shown on the block icon or below it in the format "*V X.Y*".
- Functional change or relevant appearance changes reflects on version information update (only on "X"). A change in "Y" is not functional or appearance relevant. For instance, the versions 5.1, 5.4, 5.13 have all the same appearance and functional behavior.
- The part "X" of the block version number is not coupled to SDA release number. The first block version since SDA 5.0 is "5." for all libraries though.
- All functional changes done to AT Library blocks between SDA releases are shown in this document and in the according library documentation within the SDA installation.

36.6.8 Arithmetic Operators

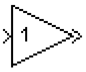


Div **(V. 5) (V. 6)**



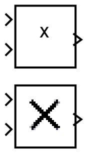
The block divides the elements of the first (upper) input by the elements of the second (lower) input. Case the upper input is a vector or matrix, it performs an element-wise division.

Gain **(simulink block)**



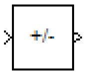
The Gain block generates its output by multiplying its input by a specified gain factor. The input and gain can be a scalar, vector, or matrix. Only unitary gain is allowed in SDA models.

Mul **(V. 5) (V. 6)**



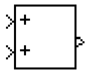
The block multiplies the elements of the first (upper) input by the elements of the second (lower) input (element wise multiplication).

Neg **(V. 5) (V. 6)**



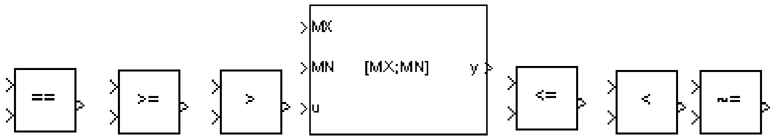
Negates the input signal. The input signal can be a scalar, a vector or a matrix.

Sum **(V. 5) (V. 6)**



The Sum block performs addition ("+" input) or subtraction ("-" input) on its inputs. This block can add or subtract scalar, vector, or matrix inputs. It can also collapse the elements of a single input vector.

36.6.9 Comparison Operators



Interval **(V. 5) (V. 6)**

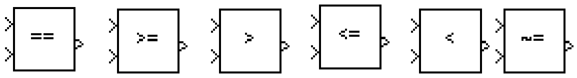
Output y is true if the input signal u lies between MN and MX.

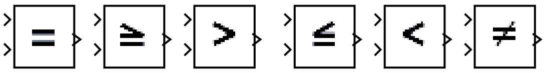
	**Variables:**
	MX: real
	MN: real
	u: real
	y: logic
		run ( )
		if((u < MX) && (u > MN)){
		y = 1;
		}
		else {
		y = 0;
		}

The braces are used to indicate if the interval limits MX and MN are contained ("[MX...", "...MN]") or not ("]MX...", "...MN[") in the interval. The ">" or "<" signs are then changed in ">=" or "<=", accordingly.

Port	I / O	Description
u	IN	Input signal
MX	IN	Maximum value for input signal
MN	IN	Minimum value for input signal
y	OUT	Boolean value, true if u lies in the Interval [MN,MX]

EQ, GE, GT, LE, LT, NE (V. 5) (V. 6)





The blocks EQ, GE, GT, LE, LT and NE performs a relational operation on its two inputs and produces output according to the following table.

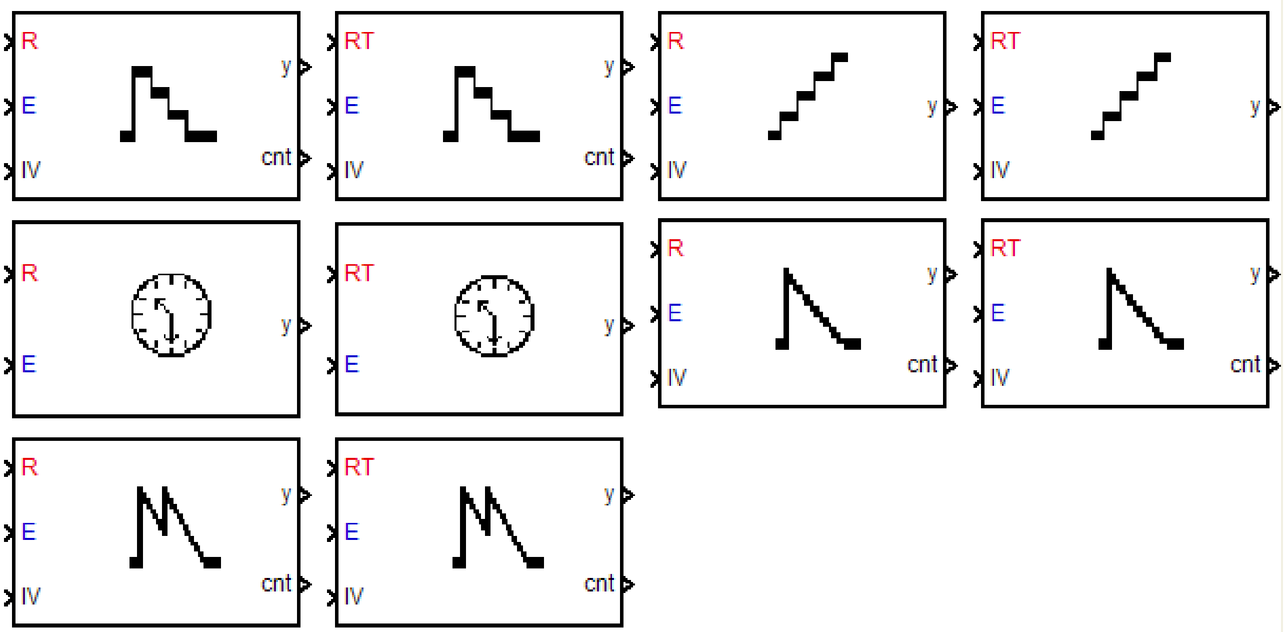
Operator	Output
==	TRUE if the first input is equal to the second input
~=	TRUE if the first input is not equal to the second input
<	TRUE if the first input is less than the second input
<=	TRUE if the first input is less than or equal to the second input
>=	TRUE if the first input is greater than or equal to the second input
>	TRUE if the first input is greater than the second input

If the result is TRUE, the output is 1; if FALSE, it is 0. Inputs can be specified as scalars, arrays, or a combination of a scalar and an array:

- For scalar inputs, the output is a scalar.
- For array inputs, the output is an array of the same dimensions, where each element is the result of an elementwise comparison of the input arrays.
- For mixed scalar/array inputs, the output is an array, where each element is the result of a comparison between the scalar and the corresponding array element.

The block icon displays the selected operator.

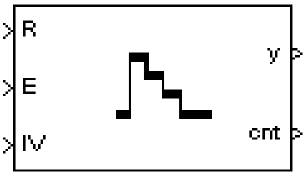
36.6.10 Counter and Timer



Optional ports:	R, E, RT, R_FC, CLC_FC, x_in, x_out, dT

CountDownRE (V. 5) (V. 6)

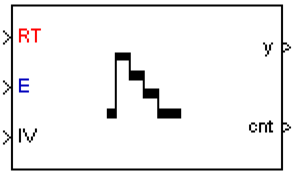
Output y is true, if the number of block evaluations since the last reset is less than the initial number of block evaluations IV. The block has state based reset.

	**Variables:**
	E: logic
	R: logic
	IV: real
	y: logic
	x: int
	cnt: int
		Init-code:
		x = 0;
		
		reset ( )
		x = round(IV);
		
		run ( )
		if (x > 0) {
		x = x - 1;
		}
			finalize( )
			
			pass ( )
			
			out ( )
			y = (x>0);
			cnt = x;

CountDownRTE (V. 5) (V. 6)

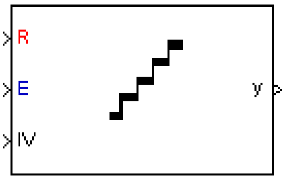
Output is true, if the number of block evaluations since the last reset is less than the initial number of block evaluations IV. The block has edge based reset.

general specification

	**Variables:**
	E: logic
	RT: logic
	IV: real
	y: logic
	x: int
	cnt: int
		Init-code:
		x = 0;
		
		reset ( )
		x = round(IV);
		
		run ( )
		if (x1 > 0) {
		x = x - 1;
		}
			finalize( )
			
			pass ( )
			
			out ( )
			y = (x>0);
			cnt = x;

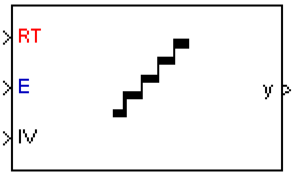
CounterRE (V. 5) (V. 6)

Counts up and gives out the number of block evaluations since the last reset. The block has state based reset.

	**Variables:**
	E: logic
	R: logic
	IV: real
	y: real
	x: int
		Init-code:
		x = 0;
		
		reset ( )
		x = round(IV);
		
		run ( )
		x = x + 1;
			finalize( )
			
			pass ( )
			
			out ( )
			y = x;

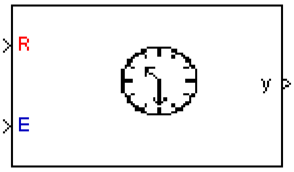
CounterRTE (V. 5) (V. 6)

Counts up and gives out the number of block evaluations since last reset. The block has edge based reset.

	**Variables:**
	E: logic
	RT: logic
	IV: real
	y: real
	x: real
		Init-code:
		x = 0;
		
		reset ( )
		x = IV;
		
		run ( )
		x = x + 1;
		pass ( )
			finalize( )
			
			out ( )
			y = x;

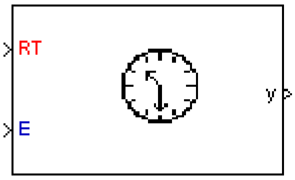
StopWatchRE (V. 5) (V. 6)

The block outputs the time since the system init or the last reset. The block has state based reset.

	**Variables:** E: logic R: logic y: real x: real	**Init-code:** x = 0.0; **reset ( )** x = 0.0; **run ( )** x = x + dT	**finalize( )** **pass ( )** **out ( )** y = x;

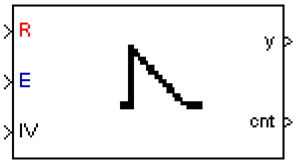
StopWatchRTE (V. 5) (V. 6)

The block outputs the time since the system init or the last reset. The block has edge based reset.

	**Variables:**
	E: logic
	RT: logic
	y: real
	x: real
		Init-code:
		x = 0.0;
		
		reset ( )
		x =0.0;
		
		run ( )
		x = x + dT;;
			finalize( )
			
			pass ( )
			
			out ( )
			y = x;

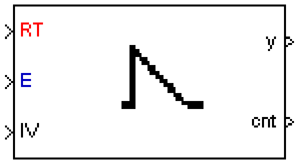
TimerRE (V. 5) (V. 6)

The block indicates true, if the initial time IV has not passed since the last reset of the block. Reset can only be performed, if the counter has run down to zero. The block has state based reset.

	**Variables:**
	E: logic
	R: logic
	IV: real
	y: logic
	x: real
	cnt: real
		Init-code:
		x = 0.0;
		
		reset ( )
		if (x < = 0) {
		x = IV;
		}
		
		run ( )
		x = MAX(x - dT, 0.0);
			finalize( )
			
			pass ( )
			
			out ( )
			cnt = x;
			y = (x>0);

TimerRTE (V. 5) (V. 6)

The block indicates true, if the initial time IV has not passed since the last reset of the block. Reset can only be performed, if the counter has run down to zero. The block has edge based reset.



Variables:
```
E: logic
R: logic
IV: real
y: logic
x: real
cnt: real
```

Init-code:
```
x = 0.0;
```

reset ( )
```
if(x<= 0.0) {
 x = IV;
}
```

run ( )
```
x = x - 1;
```

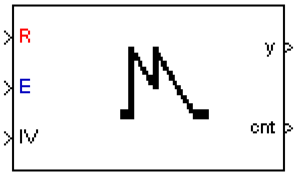
finalize( )

pass ( )

out ( )
```
cnt = x;
y = (x1>0);
```

TimerRetriggerRE (V. 5) (V. 6)

The block indicates true, if the initial time IV has not passed since the last reset of the block. Reset is always possible and independent of the current timer value. The block has state based reset.



Variables:
```
E: logic
R: logic
IV: real
y: logic
x: real
cnt: real
```

Init-code:
```
x = 0.0;
```

reset ( )
```
x = IV;
```

run ( )
```
cnt = x;
x = MAX(x - dT, 0.0);
```

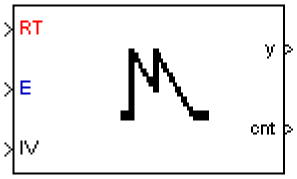
finalize( )

pass ( )

out ( )
```
y = (x>0);
```

TimerRetriggerRTE (V. 5) (V. 6)

The block indicates true, if the initial time IV has not passed since the last reset of the block. Reset is always possible and independent of the current timer value. The block has edge based reset.



Variables:
```
E: logic
RT: logic
IV: real
y: logic
x: real
cnt: real
```

Init-code:
```
x = 0.0;
```

reset ( )
```
x = IV;
```

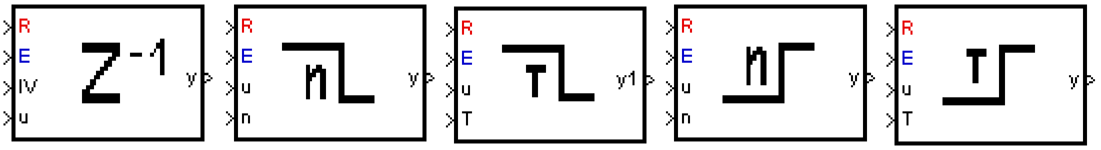
out ( )
```
cnt = x;
y = (x>0);
```

finalize( )

run ( )
```
x = MAX(x - dT, 0.0);
```

pass ( )

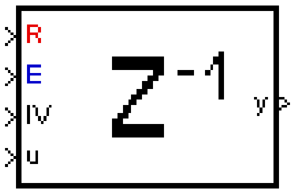
36.6.11 Delay Blocks



Optional ports: R, E, R_FC, CLC_FC, x_in, x_out, dT

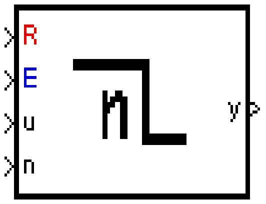
Delay_RE (V. 5) (V. 6)

Delays the input signal u by one sample time. The reset value IV has direct influence on the output. DelayResetEnabled, Delay_RE and Unit Delay depend on individual simulation purposes.

	**Variables:**
	E: logic
	R: logic
	IV: real
	u: real
	y: real
	x: real
		Init-code:
		x = 0;
		
		reset ( )
		x = IV;
		y = IV
		
		run ( )
		y = x;
		x = u
			finalize( )
			
			pass ( )
			y = x;
			
			out ( )

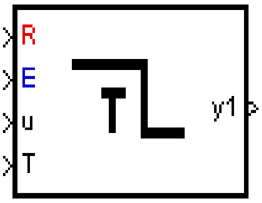
TurnOffDelaySample (V. 5) (V. 6)

A falling edge of the input signal u is delayed by n block evaluations (E).

	**Variables:**
	E: logic
	R: logic
	n: real
	u: logic
	y: logic
	x: int
	temp: int
		Init-code:
		x = 0;
		
		reset ( )
		x = round(n);
		
		pass ( )
		
		out ( )
		y = ((temp>0) || u);
			finalize( )
			temp = x;
			if(u) {
			x = round(n);
			}
			else {
			if(temp > 0) {
			x = x - 1;
			}
			}
			
			run ( )

TurnOffDelayTime (V. 5) (V. 6)

A falling edge of the input signal u is delayed by the time T, if the input remains false for minimum the whole period T.



Variables:
T: real
R: logic
E: logic
y: logic
x: real
temp: real

Init-code:
```
x = 0.0;
```

reset ( )
```
x = T;
```

pass ( )

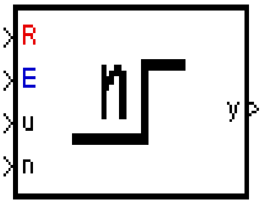
out ( )
```
y = ((temp>0) || u);
```

finalize( )
```
temp = x;
if(u) {
 x = T;
}
else {
 if(temp > 0.0) {
  x = x - dT;
 }
}
```

run ( )

TurnOnDelaySample (V. 5) (V. 6)

A rising edge of the input signal u is delayed by n block evaluations.



Variables:
E: logic
R: logic
n: real
u: logic
y: logic
x: int
temp: int

Init-code:
```
x = 0;
```

reset ( )
```
x = round(n);
```

pass ( )

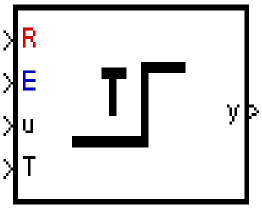
out ( )
```
y =((temp<=0) && u);
```

finalize( )
```
temp = x;
if(u) {
 if(temp > 0) {
  x = x - 1;
 }
}
else {
 x = round(n);
}
```

run ( )

TurnOnDelayTime (V. 5) (V. 6)

A rising edge of the input signal u is delayed by the time T, if the input signal remains high (true) for minimum this period T.



Variables:
E: logic
R: logic
T: real
y: logic
x: real
temp: real

Init-code:
```
x = 0.0;
```

reset ( )
```
x = T;
```

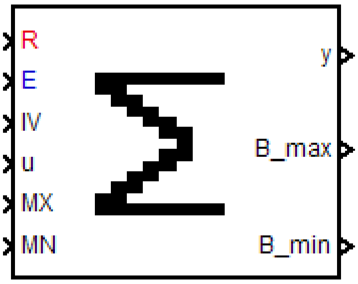
pass ( )

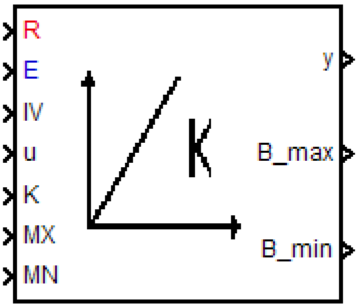
out ( )
```
y = ((temp<=0) && u);
```

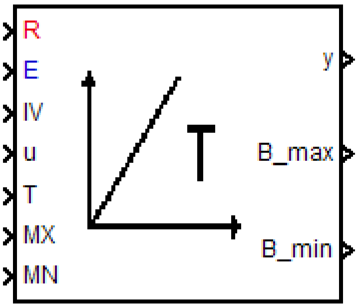
finalize( )
```
temp = x;
if(u) {
 if(temp > 0.0) {
  x = x - dT;
 }
}
else {
 x = T;
}
```

run ( )

36.6.12 Integrators



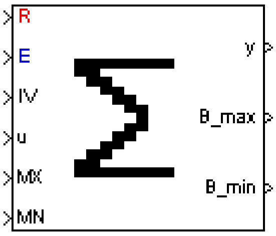




Optional ports: R, E, R_FC, CLC_FC, x_in, x_out, dT, B_max, B_min

AccumulatorREL (V. 5) (V. 6)

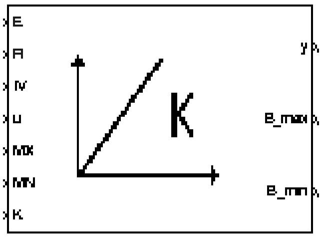
A time discrete integrator. The integrator value y is limited to the maximum MX and the minimum MN.

	**Variables:**
	MX: real
	MN: real
	u: real
	E: logic
	R: logic
	IV: real
	y: real
	B_max: logic
	B_min: logic
	x: real
		Init-code:
		x = 0.0;
		reset ( )
		x = IV;
		run ( )
		x = x+ u;
		pass ( )
		finalize ( )
			out ( )

```
if (x > MX) {
    x = MX;
    B_max = 1;
    B_min = 0;
 }
 else if (x < MN) {
    x = MN;
    B_max = 0;
    B_min = 1;
 }
 else {
   B_max = 0;
   B_min = 0;
 }
 y = x;
```

IntegratorKREL (V. 5) (V. 6)

A time discrete integrator with gain K. The integrated value and therefore the output y are limited by the inputs MX and MN.

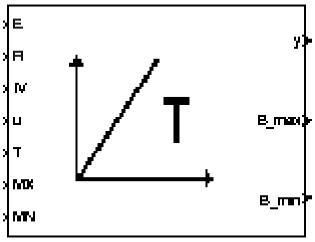
	**Variables:**
	MX: real
	MN: real
	K: real
	u: real
	E: logic
	R: logic
	IV: real
	y: real
	B_max: logic
	B_min: logic
	x: real
		Init-code:
		x = 0.0;
		reset ( )
		x = IV;
		run ( )
		x = x + K*u*dT;
		pass ( )
		finalize( )
			out ( )

```
 if (x > MX) {
    x = MX;
    B_max = 1;
    B_min = 0;
 }
 else if (x < MN) {
    x = MN;
    B_max = 0;
    B_min = 1;
 }
 else {
```

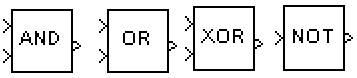
			B_max = 0; B_min = 0; } y = x;

IntegratorTREL **(V. 5) (V. 6)**

A time discrete integrator with gain 1/T. The integrated value and therefore the output y are limited by the inputs MX and MN.

	**Variables:**
	MX: real
	MN: real
	T: real
	u: real
	E: logic
	R: logic
	IV: real
	y: real
	B_max: logic
	B_min: logic
	x: real
		Init-code:
		x = 0.0;
		
		reset ( )
		x = IV;
		
		run ( )
		x = x + u*dT/T;
		
		pass ( )
		
		finalize( )
			out ( )
			if (x > MX) {
			x = MX;
			B_max = 1;
			B_min = 0;
			}
			else if (x < MN) {
			x = MN;
			B_max = 0;
			B_min = 1;
			}
			else {
			B_max = 0;
			B_min = 0;
			}
			y = x;

36.6.13 Logical Operators (V. 5) (V. 6)



Logical Operators

These blocks performs the logical operations on its inputs: AND, OR, XOR, and NOT. The output depends on the number of inputs, and their dimensionality. The output is 1 if TRUE and 0 if FALSE. The block icon shows the corresponding operator.

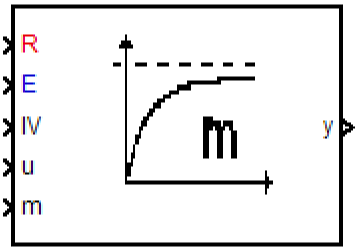
If the block has more than one input, the output has the same dimensions as the inputs (after scalar expansion) and each output element is the result of applying the specified logical operation to the corresponding input elements. For example, if the operation is ‘AND’ and the inputs are 2-by-2 arrays, the output is a 2-by-2 array whose top, left element is the result of applying AND to the top, left elements of the inputs, etc.

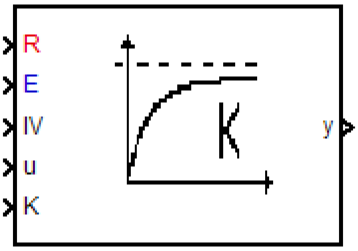
If the block has a single input and the block is not the NOT operator, the input must be vector-like, i.e. a scalar, a 1-D array, or a one-row or one-column 2-D array. The output is a scalar value equal to the result of applying the operation to the elements of the input.

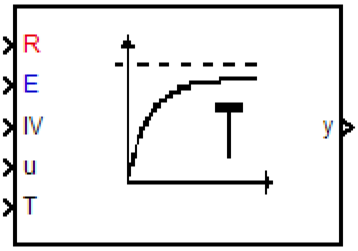
When configured as a multi-input XOR gate, this block performs an addition modulo two operations as mandated by the IEEE standard for logic elements.

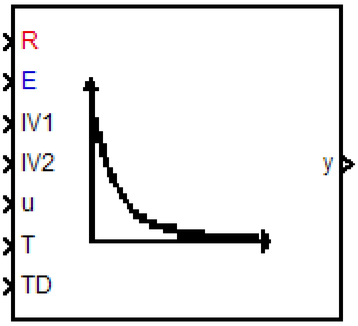
All blocks support **bitwise** operations. In this case, the logical operation is calculated bit by bit between the input ports. The block mode is identified by the annotation "bitwise" under its icon.

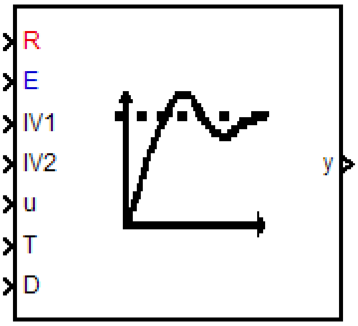
36.6.14 Low and highpass











Optional ports: R, E, RT, R_FC, CLC_FC, x_in, x_in, x1_in, x2_in, x1_out, x2_out, dT

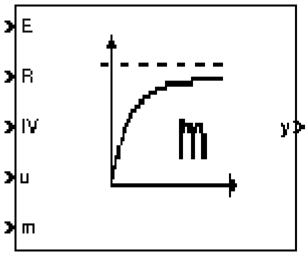
DigitalLowpassRE **(V. 5) (V. 6)**

A discrete time first order lowpass. Value m indicates the percentage of the current approximation of the output value to the aimed input value.

Relationship between the cut-off-frequency (fc) and m:

$$m = 1 - e^{(-f_c \cdot dT \cdot 2 \cdot \pi)} \qquad f_c = -\frac{\ln(1-m)}{2 \cdot \pi \cdot dT}$$

with sample Time dT.

	**Variables:**
	E: logic
	R: logic
	IV: real
	u: real
	m: real
	y: real
	x: real
		Init-code:
		x = 0.0;
		
		reset ( )
		x = IV;
		
		run ( )
		x = x + m*(u - x);
			finalize( )
			
			pass ( )
			
			out ( )
			y = x;

HighpassTRE (V. 5) (V. 6)

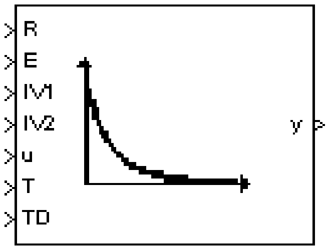
This block performs a discretized first order high pass filter with the continuous time Laplace notation:

$$G(s) = \frac{TD \cdot s}{1 + T \cdot s}$$

If TD=T the CSD discrete highpass is recovered. In this case the relationship between the highpass cut-off-frequency (fc) and T is :

$$T = \left(1 - e^{(-f_c \cdot dT \cdot 2\pi)}\right) \cdot dT \qquad f_c = -\frac{\ln\left(1 - \frac{T}{dT}\right)}{2 \cdot \pi \cdot dT}$$

with sample Time dT.

	**Variables:**
	E: logic
	R: logic
	IV: real
	y: logic
	x1: real
	x2:real
	T: real
	TD: real
		Init-code:
		x1 = 0.0;
		x2 = 0.0;
		
		pass ( )
		
		out ( )
		y = x1;
			finalize( )
			
			reset ( )
			x1 = IV1;
			x2 = IV2;
			
			run ( )
			x1 = x1 + (TD*(u-x2) - x1*dT)/T;
			x2 = u;

Input IV1	Initial value (for time step n) set upon reset
Input IV2	Initial value (for time step n-1) set upon reset

LowpassKRE (V. 5) (V. 6)

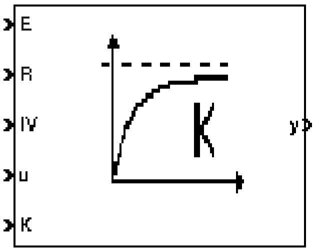
A discrete time first order lowpass filter with time constant 1/K.

Relationship between the cut-off-frequency (f_c) and K:

$$K = \frac{\left(1 - e^{\left(-f_c \cdot dT \cdot 2 \cdot \pi\right)}\right)}{dT} \qquad f_c = -\frac{\ln\left(1 - K \cdot dT\right)}{2 \cdot \pi \cdot dT}$$

with sample Time dT. The input K is the inverse time constant.

	Variables:	**Init-code:**	**finalize( )**

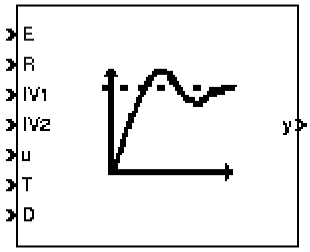
	E: logic
	R: logic
	IV: real
	u: real
	K: real
	y: real
	x: real
		x = 0.0;
		
		reset ( )
		x = IV;
		
		run ( )
		x = x + K*dT*(u-x);
			
			pass ( )
			
			out ( )
			y = x;

<u>**LowpassSecOrdRE**</u> **(V. 5) (V. 6)**

A discrete time second order lowpass filter with time constant T and damping factor D (PT2) with continuous time Laplace notation:

$$G(s) = \frac{1}{s^2 \cdot T^2 + s \cdot 2 \cdot D \cdot T + 1}$$

	Variables:	**Init-code:**	**finalize( )**

	E: logic
	R: logic
	IV1: real
	IV2: real
	u: real
	T: real
	D: real
	y: real
	x1: real
	x2: real
	temp: real
	temp1: real
		x1 = 0.0;
		x2 = 0.0;
		
		reset ( )
		x1 = IV1;
		x2 = IV2;
		
		pass ( )
		
		out ( )
		y = x1;
			
			run ( )
			temp = x2;
			temp1 = pow(T,2) + 2*D*T*dT;
			x2 = x1;
			x1 = x2 + (pow(dT,2)*(u-x2))/temp1 + pow(T,2)*(x2-temp)/temp1;

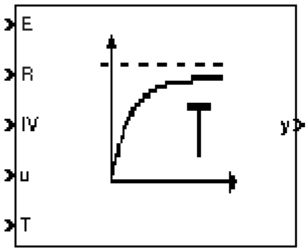
T	Time constant (real)
D	Damping factor (real)
IV1	Initial value set upon reset
IV2	Initial value set upon reset

LowpassTRE (V. 5) (V. 6)

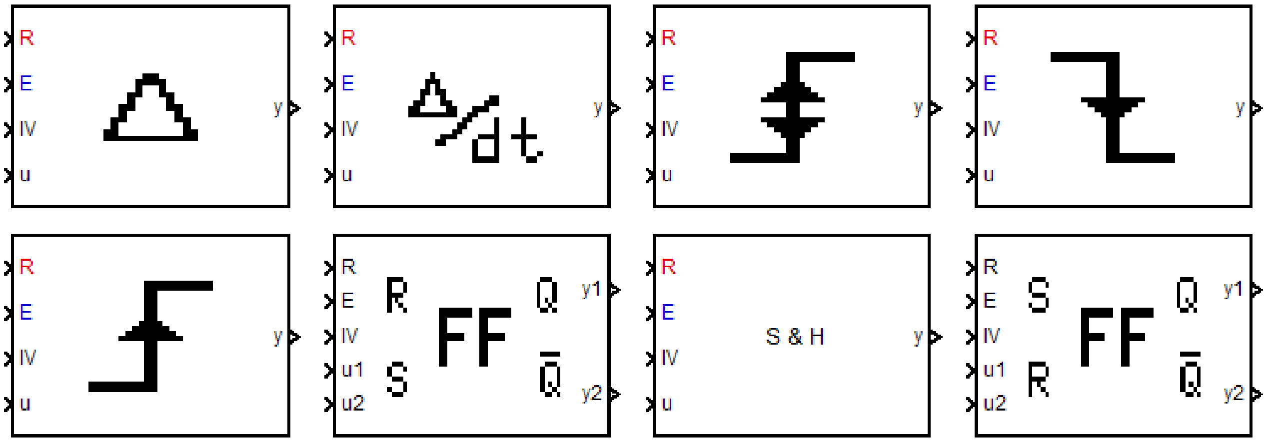
This block performs a first order lowpass filter with T as time constant. Relationship between the cut-off-frequency (fc) and T:

$$K = \frac{\left(1 - e^{(-f_c \cdot dT \cdot 2 \cdot \pi)}\right)}{dT} \qquad f_c = -\frac{\ln\left(1 - K \cdot dT\right)}{2 \cdot \pi \cdot dT}$$

with sample Time dT.

	**Variables:**
	E: logic
	R: logic
	IV: real
	u: real
	T: real
	y: real
	x: real
		Init-code:
		x = 0.0;
		
		reset ( )
		x = IV;
		
		run ( )
		x = x + dT*(u-x)/T;
			finalize( )
			
			pass ( )
			
			out ( )
			y = x;

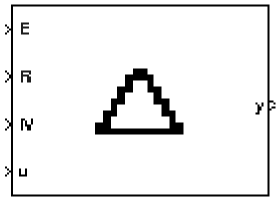
36.6.15 Memory Blocks



Optional ports: R, E, RT, R_FC, CLC_FC, x_in, x_out, dT

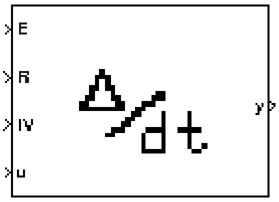
DeltaOneStep (V. 5) (V. 6)

Calculates the difference between the current and the last input value.

	**Variables:**
	E: logic
	R: logic
	IV: real
	u: real
	y: real
	x: real
		Init-code:
		x = 0.0;
		reset ( )
		x = IV;
		run ( )
			finalize( )
			y = u - x;
			x = u;
			pass ( )
			out ( )

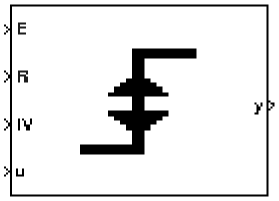
DifferenceQuotient (V. 5) (V. 6)

Calculates the rate of change of the input signal over time.

	**Variables:**
	E: logic
	R: logic
	IV: real
	u: real
	y: real
	x: real
		Init-code:
		x = 0.0;
		reset ( )
		x = IV;
		run ( )
			finalize( )
			y = (u - x)/dT;
			x = u;
			pass ( )
			out ( )

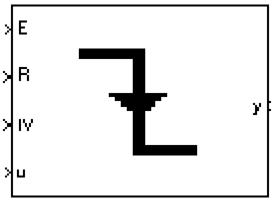
EdgeBi (V. 5) (V. 6)

The output y indicates true at any change of the logical input value.

	**Variables:**
	E: logic
	R: logic
	IV: real
	u: logic
	y: logic
	x: logic
		Init-code:
		x = 0;
		reset ( )
		x = IV;
		run ( )
			finalize( )
			y = (x ! = u);
			x = u;
			pass ( )
			out ( )

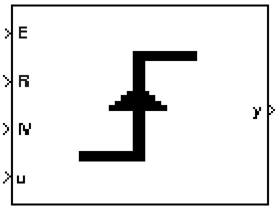
EdgeFalling (V. 5) (V. 6)

The output y indicates true if the input value changes from true to false.

	**Variables:**
	E: logic
	R: logic
	IV: logic
	u: logic
	y: logic
	x: logic
		Init-code:
		x = 0;
		
		reset ( )
		x = IV;
		
		run ( )
			finalize( )
			y = (x && !u);
			x = u;
			
			pass ( )
			
			out ( )

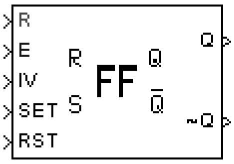
EdgeRising (V. 5) (V. 6)

The output y indicates true if the input value changes from false to true.

	**Variables:**
	E: logic
	R: logic
	IV: logic
	u: logic
	y: logic
	x: logic
		Init-code:
		x = 0;
		
		reset ( )
		x = IV;
		
		run ( )
			finalize( )
			y = (!x && u);
			x = u;
			
			pass ( )
			
			out ( )

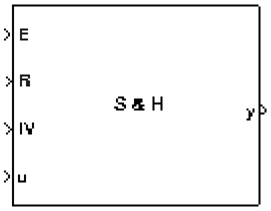
RSFlipFlop (V. 5) (V. 6)

A flip flop. The second input (reset) dominates the first input (set).

	**Variables:**
	SET: logic
	RST: logic
	Q: logic
	~Q: logic
	x: logic
		Init-code:
		x = 0;
		
		reset ( )
		x = IV;
		
		pass ( )
		
		out ( )
		Q = x;
		~Q = !x;
			finalize( )
			
			run ( )
			if(RST) {
			x = 0;
			}
			else if(SET) {
			x = 1;
			}

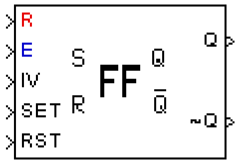
SampleAndHoldRE (V. 5) (V. 6)

Sample and Hold or just a simple memory block with enable and reset ports.

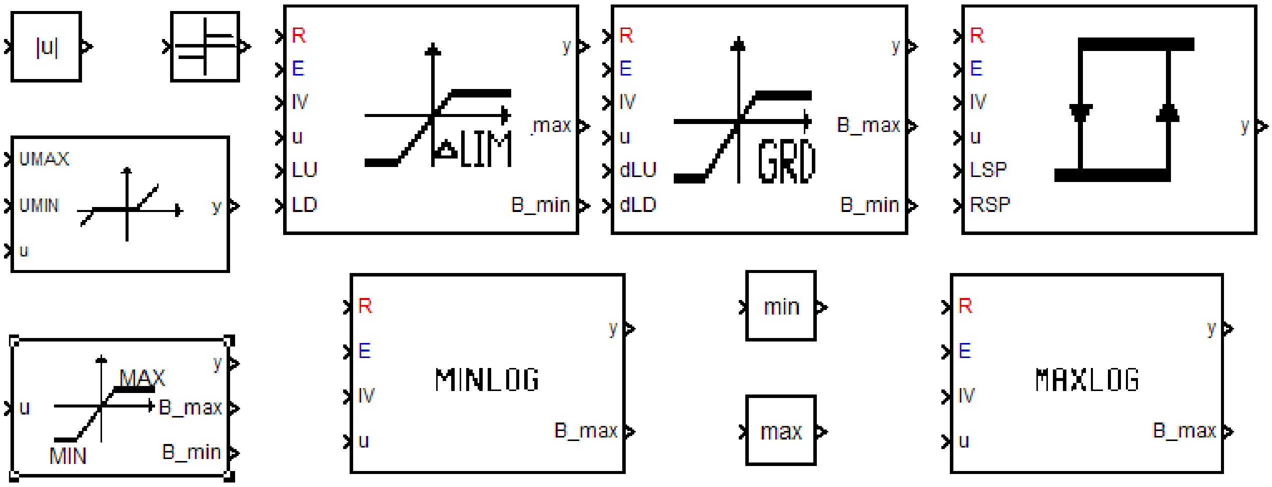
	**Variables:**
	E: logic
	R: logic
	IV: real
	u: real
	y: real
	x: real
		Init-code:
		x = 0.0;
		
		reset ( )
		x = IV;
		
		run ( )
		x = u;
			finalize( )
			
			pass ( )
			
			out ( )
			y = x;

SRFlipFlop (V. 5) (V. 6)

A flip flop. The first input (set) dominates the second input (reset).

	**Variables:**
	SET: logic
	RST: logic
	Q: logic
	~Q: logic
	x: logic
		Init-code:
		x = 0;
		
		reset ( )
		x = IV;
		
		pass ( )
		
		out ( )
		Q = x;
		~Q = !x;
			finalize( )
			
			run ( )
			if(SET) {
			x = 1;
			}
			else if(RST) {
			x = 0;
			}

36.6.16 Nonlinear Blocks - automotive blockset

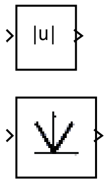


Some blocks have multirate version. The Multirate version of these blocks (with R_FC and CLC_FC inputs) is used if system resets have to be simulated. Other optional inputs and outputs in these blocks are:

Optional ports:	R, E, RT, R_FC, CLC_FC, x_in, x_out, dT, B_max, B_min

Abs **(V. 5) (V. 6)**

The Abs block generates as output the absolute value of the input.



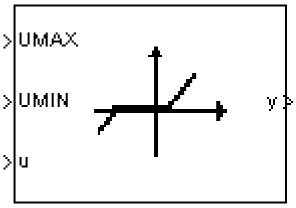
Variables:
u: real
y: real

run ( )

```
if(u >= 0) {
  y = u;
}
else {
  y = -u;
}
```

Dead Band **(V. 5) (V. 6)**

Outputs zero, if the input value is between UMIN and UMAX. Otherwise the output signal is the input signal reduced by the input limits.



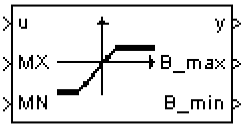
Variables:
UMAX: real
UMIN: real
u: real
y: real

run ( )

```
if(u < UMAX && u > UMIN) {
 y = 0.0;
}
else if(u >= UMAX) {
 y = u - UMAX;
}
else {
 y = u - UMIN;
}
```

Limit **(V. 5) (V. 6)**

The input signal u is limited between maximum value MX and minimum value MN. The boolean output flags B_MAX and B_MIN represent an active limitation in both directions.



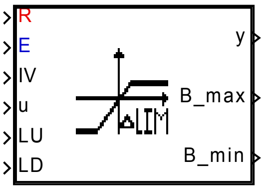
Variables:
MX: real
MN: real
u: real
y: real
B_min: logic
B_max: logic

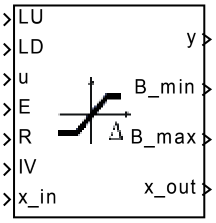
run ( )

```
if(u > MX) {
  y = MX;
  B_max = 1;
  B_min = 0;
}
else if(u < MN) {
  y = MN;
  B_max = 0;
  B_min = 1;
}
else {
  y = u;
  B_max = 0;
  B_min = 0;
}
```

DifferenceLimiter (V. 5) (V. 6)

The output value is limited in its difference to the last input value. The maximum valid difference is LU, the minimum valid difference is LD. Only for a sampling time of 1s this block behaves like a Gradient limiter. For other sampling times this time has to be taken into account in setting the correct values for LU and LD.





Variables:
E: logic
R: logic
IV: real
u: real
LU: real
LD: real
y: logic
B_max: logic
B_min: logic
x: real

run ( )

```
if((u-x)>LU) {
  x = x + LU;
  B_min = 0;
  B_max = 1;
}
else if((u - x)< LD) {
  x = x + LD;
  B_min = 1;
  B_max = 0;
}
else {
  x = u;
  B_min = 0;
  B_max = 0;
}
```

finalize( )

pass ( )

out ( )

```
  y = x;
```

Init-code:

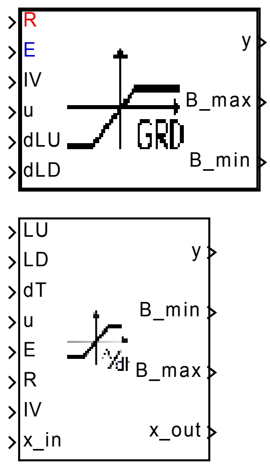
```
  x = 0.0;
```

reset ( )

```
  x = IV;
```

GradientLimiter (V. 5) (V. 6)

The output value is limited in its gradient to the last input value. The maximum valid difference is dLU, the minimum valid difference is dLD. The values of dLD and dLU have to be given per second.



Variables:

```
 E: logic
 R: logic
 IV: real
 u: real
 dLU: real
 dLD: real
 y: real
 B_min: logic
 B_max: logic
 x: real
```

Init-code:

```
 x = 0.0;
```

reset ( )

```
 x = IV;
```

pass ( )

finalize( )

out ( )

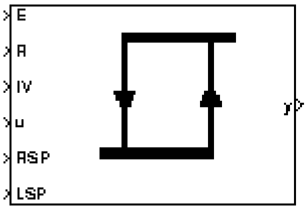
```
 y = x;
```

run ( )

```
if((u - x)/dT > dLU) {
  x = x + dLU*dT;
  B_min = 0;
  B_max = 1;
}
else if((u - x)/dT < dLD) {
  x = x + dLD*dT;
  B_min = 1;
  B_max = 0;
}
else {
  x = u;
  B_min = 0;
  B_max = 0;
}
```

Hysteresis (V. 5) (V. 6)

The Output of this block is a logical 1 if the input value is greater than the RSP and logical 0 if the input is below the LSP.



Variables:

```
 E: logic
 R: logic
 IV: logic
 u: real
 RSP: real
 LSP: real
 y: logic
 x: logic
```

Init-code:

```
 x = 0;
```

reset ( )

```
 x = IV;
```

pass ( )

out ( )

```
 y = x;
```

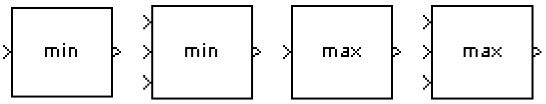
finalize( )

```
 if(u > RSP) {
   x = 1;
 }
 else if(u < LSP) {
   x = 0;
 }
```

run ( )

RSP	Upper threshold value
LSP	Lower threshold value

Min, Max (V. 5) (V. 6)



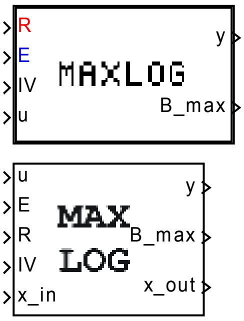
The MinMax block outputs either the minimum or the maximum element or elements of the input(s).

If the block has one input port, the input must be a scalar or a vector. The block outputs a scalar equal to the minimum or maximum element of the input vector.

If the block has multiple input ports, the nonscalar inputs must all have the same dimensions. The block expands any scalar inputs to have the same dimensions as the nonscalar inputs. The block outputs a signal having the same dimensions as the input. Each output element equals the minimum or maximum of the corresponding input elements.

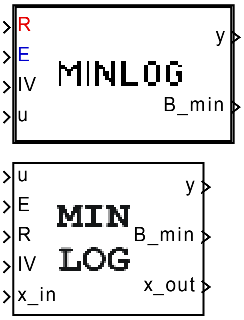
MaxLogRE (V. 5) (V. 6)

The output value is the maximum value of all input values u between resets, with respect to the initial value IV.

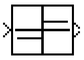
	**Variables:**
	E: logic
	R: logic
	IV: real
	u: real
	y: real
	B_max: logic
	x: real
		Init-code:
		x = 0.0;
		B_max = 0;
		
		reset ( )
		x = IV;
		
		out ( )
		y = x;
			finalize( )
			
			run ( )
			B_max = 0;
			if(u > x) {
			x = u;
			B_max = 1;
			}
			
			pass ( )

MinLogRE (V. 5) (V. 6)

The output value y is the minimum value of all input values u between resets, with respect to the initial value IV.

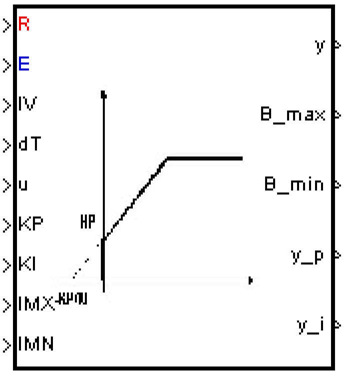
	**Variables:**
	E: logic
	R: logic
	IV: real
	u: real
	y: logic
	B_min: logic
	x: real
		Init-code:
		x = 0.0;
		B_min = 0;
		
		reset ( )
		x = IV;
		
		out ( )
		y = x;
			finalize( )
			
			run ( )
			B_min = 0;
			if(u < x) {
			x = u;
			B_min = 1;
			}
			
			pass ( )

Sign (V. 5) (V. 6)

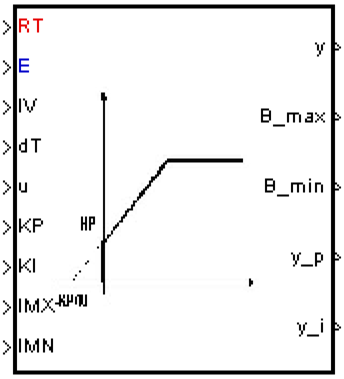

The Sign block indicates the sign of the input:

- The output is 1 when the input is greater than zero.
- The output is 0 when the input is equal to zero.
- The output is -1 when the input is less than zero.

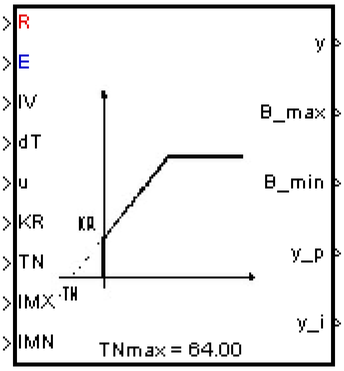
36.6.17 Controller_Blocks (AT_Extended_Lib)



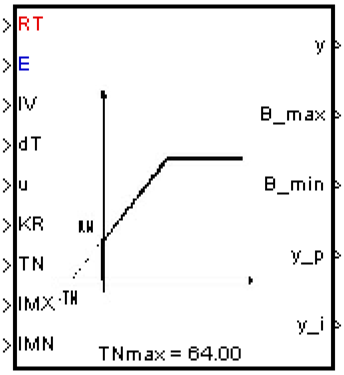
PIControllerK_RE



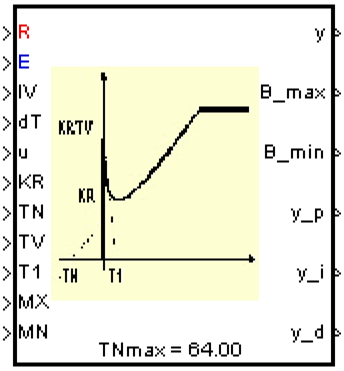
PIControllerK_RTE



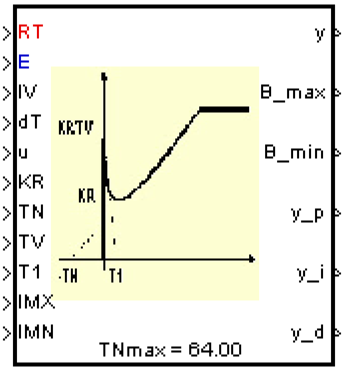
PIControllerT_RE



PIControllerT_RTE



PIDControllerT_RE



PIDControllerT_RTE

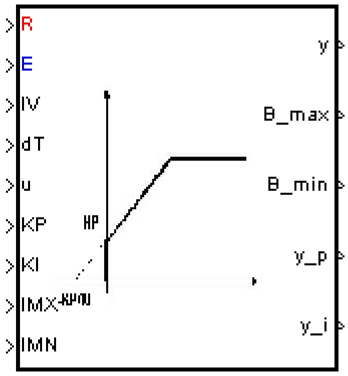
The blocks could be figured for single- and multi-rate version. Multirate version (with inport R_FC and CLC_FC) are used for system reset. Each of PIControllerK, PIControllerT and PIDControllerT has two version with different function reset inport: R (state based) and RT (edge based).

Controller blocks can be configured for single- and multirate mode.

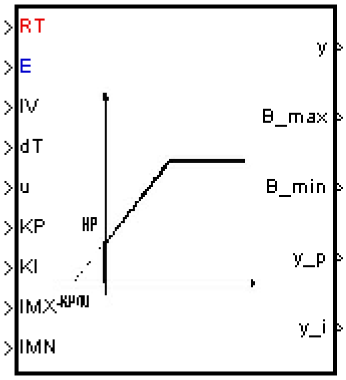
Optional ports:	R, E, RT, R_FC, CLC_FC, IV, xi_in, xd_in, xi_out, xd_out, y_p, y_i, y_d, B_max, B_min

IMX	I-state upper limitation value
IMN	I-state lower limitation value
y_p	Level of proportional part of the controller output
y_i	Level of integral part of the controller output

PIControllerK_RE (V.5) and PIControllerK_RTE (V.5)



PIControllerK_RE



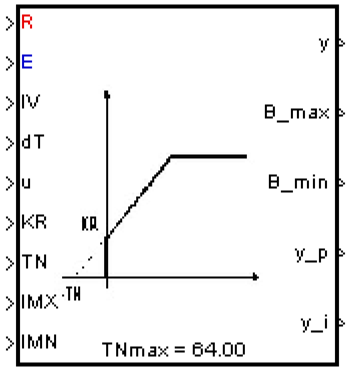
PIControllerK_RTE

This block presents a resetable (integral part) PI controller with variable parameters and state limitation. It is based on the transfer function : G(s) = KP + KI/s, but with full consideration of parameter variability and with the discretization (used method: euler backward) in integrator part.

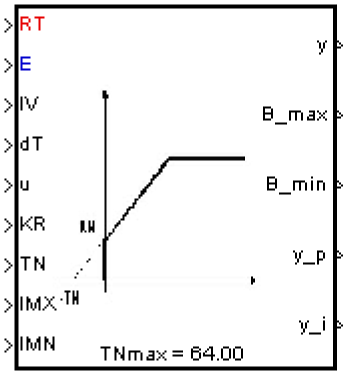
The limits affect the integral state only. The output is the unlimited sum of the unlimited proportional part and the limited integral part.

KP	Proportional gain
KI	Integration gain
y_p	Level of proportional part of the controller output
y_i	Level of integral part of the controller output

PIControllerT_RE (V.5) and PIControllerT_RTE (V.5)



PIControllerT_RE



PIControllerT_RTE

The block represents a resetable (integral part) PI controller with variable parameters and state limitation. It is based on the transfer function :G(s) = KR * (1 + 1/(TN*s)), but with full consideration of parameter variability and with the discretization (used method: euler backward) in integrator part.

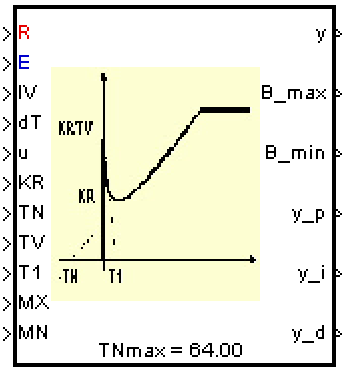
The limits affect the integral state only. The output is the unlimited sum of the unlimited proportional part and the limited integral part.

Setting TN to or below the sample time disables the proportional part (TN <= dt ↔ KR = 0).

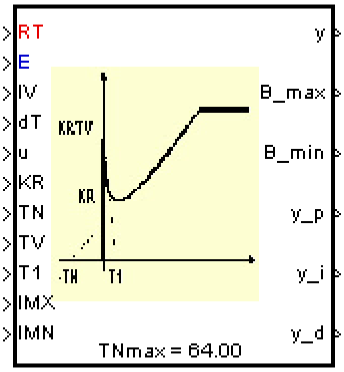
Setting TN to or above the maximum value disables integration (TN >= TNmax ↔ The integrator's part is constant).

KR	Controller amplification
TN	Integrator adjusting time

PIDControllerT_RE (V.5) and PIDControllerT_RTE (V.5)



PIDControllerT_RE



PIDControllerT_RTE

This block represents resetable (integral and derivative part), limitable PID controller with variable parameters. It is based on the transfer function:

G(s) = KR * (1 + 1/(TN*s) + TV*s/(1 + T1*s))

but with full consideration of parameter variability. The discretization (used method: euler backward) is used in integrator part and derivative part (the discretization of each part is separately).

Setting TN to or below the sample time disables the proportional part.

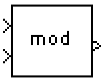
Setting TN to or above the maximum value disables integration.

To disable the derivative part, set TV=0.

KR	Controller amplification
TN	Integrator adjusting time
TV	Derivative adjusting time
T1	Derivative adjusting time
y_d	Level of derivative part of the controller output

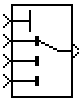
36.6.18 Nonlinear

Modulo (V. 5) (V. 6)



The Modulo block outputs the modulus after division. The inputs must be real arrays of the same size, or real scalars.

Multiport Switch (Simulink block)



The Multiport Switch block chooses between a number of inputs.

The first (top) input is the control input and the other inputs are data inputs. The value of the control input determines which data input to pass through to the output port.

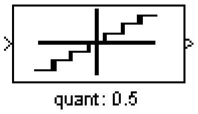
If the control input is not an integer value, the Multiport Switch truncates the value to the nearest integer and issues a warning. If the (truncated) control input is less than one or greater than the number of input ports, the switch issues an out-of-bounds error. Otherwise, the switch passes the data input that corresponds to the (truncated) control input. The following table summarizes the Multiport Switch's behavior.

(Truncated) Control Input	Passes This Data Input
Less than 1	Out of bounds error
1	First input
2	Second input
etc.	etc.
Greater than the number of data inputs	Out of bounds error

Data inputs can be scalar or vector. The control input can be a scalar or a vector. The block output is determined by these rules:

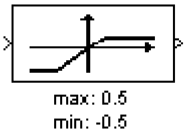
- If inputs are scalar, the output is a scalar.
- If the block has more than one data input, at least one of which is an array, the output is an array. Any scalar inputs are expanded to arrays.
- If the block has only one data input, the input must be a scalar or a vector (1-D array). If the input is a vector, the block output is the element of the vector that corresponds to the truncated value of the control input.

Quantizer **(V. 5)**



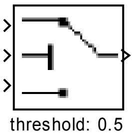
The Quantizer block passes its input signal through a stair-step function so that many neighbouring points on the input axis are mapped to one point on the output axis. The effect is to quantize a smooth signal into a stair-step output. The output is computed using the round-to-nearest method, which produces an output that is symmetric about zero: y = q * round(u/q), where y is the output, u the input, and q the Quantization interval parameter.

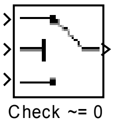
Saturation **(V. 5)**



The Saturation block imposes upper and lower bounds on a signal. When the input signal is within the range specified by the Lower limit and Upper limit parameters, the input signal passes through unchanged. When the input signal is outside these bounds, the signal is clipped to the upper or lower bound. When the parameters are set to the same value, the block outputs that value.

Switch **(V. 5) (V. 6)**



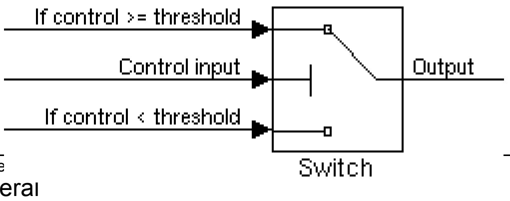


The Switch block propagates one of two inputs to its output depending on the value of a third input, called the control input.

Switch with threshold: If the signal on the control (second) input is greater than or equal to the Threshold parameter, the block propagates the first input; otherwise, it propagates the third input. This figure shows the use of the block ports.

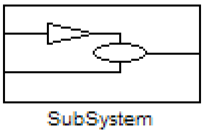
Switch with check unequal 0: If the signal on the control (second) input is unequal to 0, the block propagates the first input; otherwise, it propagates the third input.

This figure shows the use of the block ports:



36.6.19 Signals & Systems

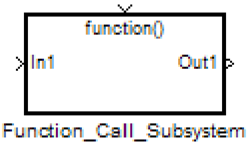
Subsystem



Represent a nested subsystem within another system.

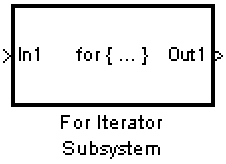
Subsystems can be virtual or atomic, depending on the value of its Treat as Atomic Unit parameter. Simulink ignores virtual subsystem boundaries when determining block update order. By contrast, Simulink executes all blocks within an atomic subsystem before moving on to the next block. Conditionally executed subsystems are atomic. Unconditionally executed subsystems are virtual by default. You can, however, designate an unconditionally executed subsystem as atomic. This is useful if you need to ensure that a subsystem is executed in its entirety before any other block is executed. An atomic subsystem is identified by bold boundaries.

Function Call Subsystem



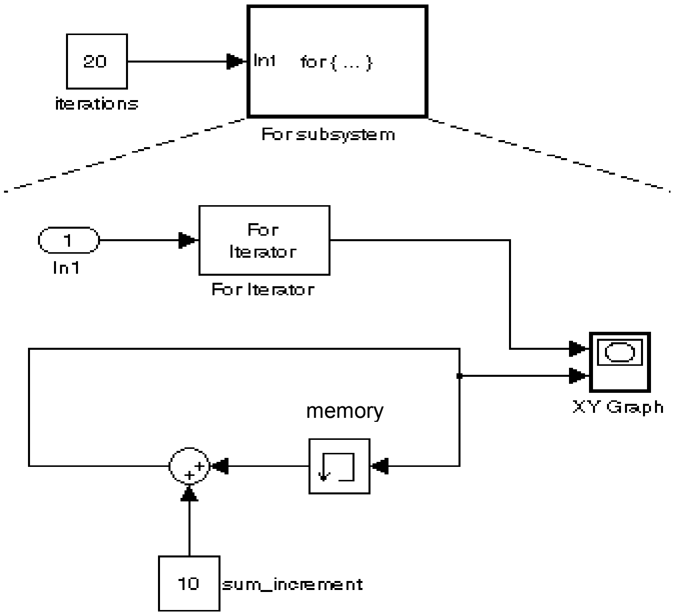
Represent a subsystem that can be invoked as a function by another block. Function-call subsystems are not executed directly by Simulink; rather, the Stateflow charts or other blocks determine when to execute the subsystem. They can be executed repeatedly during the same simulation time step. When the subsystem completes execution, control returns to the caller.

For Iterator Subsystem

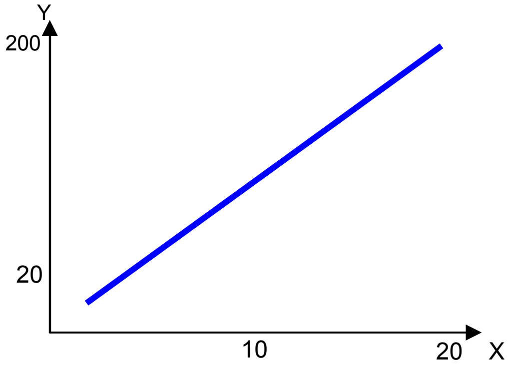


This block is a Subsystem block that is preconfigured to serve as a starting point for creating a subsystem that executes repeatedly during a simulation time step. It contains the For Iterator block.

The For Iterator block, when placed in a subsystem, implements a C-like for control flow statement in Simulink as a For subsystem. In the For subsystem, the For Iterator block has iterative control over any Simulink blocks present. For each iteration value of the For Iterator block, the accompanying blocks execute. The number of iterations is set internally for the For Iterator block or externally with data input. The following example shows a completed for control flow statement that increments an initial value of zero by 10 over 20 iterations.



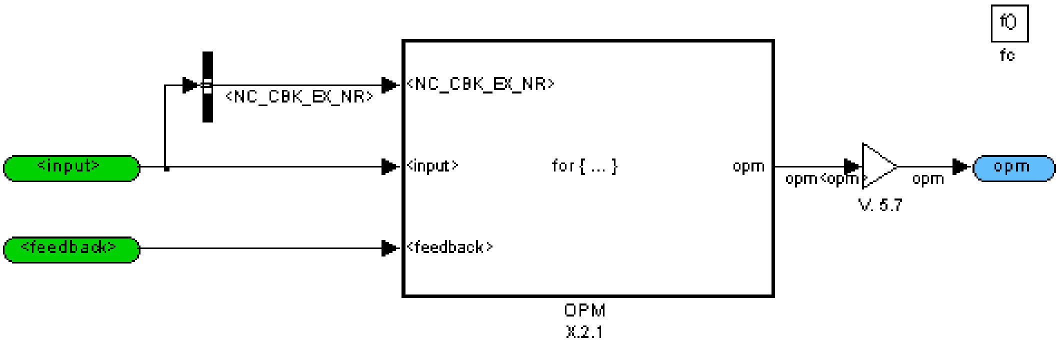
In the example above, a For subsystem receives an input, which it passes to the For Iterator block inside. The For Iterator block uses this input to determine the number of times it executes the blocks of its subsystem, in this case, 20 times. Each time the blocks execute, a value of 10 is added to a sum, which is initially zero. In addition, for each time the blocks of the subsystem execute, the For Iterator block outputs a value equal to the number of times that the blocks have executed, including the current execution. This is referred to as the iterator value. This value, along with the sum value, is sent to an XY Graph block:

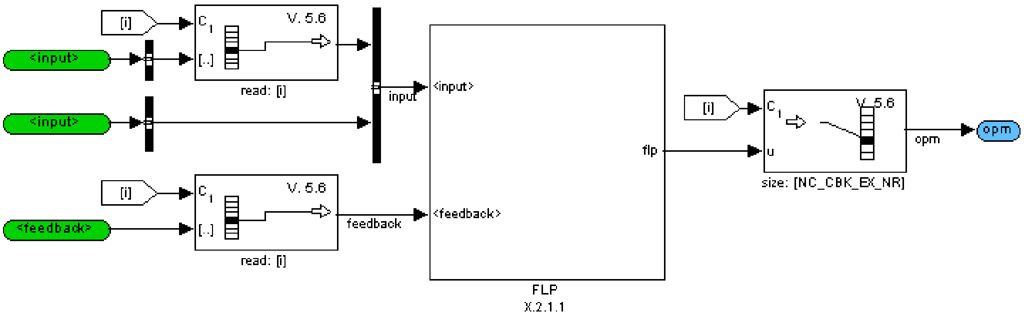


The control flow statement example above can be represented by the following pseudo code.

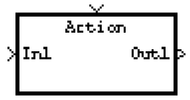
```
sum = 0;
iterations = 20;
sum_increment = 10;
for (i = 0; i < iterations;
i++) {
    sum = sum + sum_increment;
}
```

Example of a For loop:





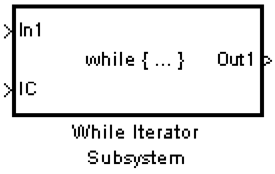
Switch Case Action Subsystem

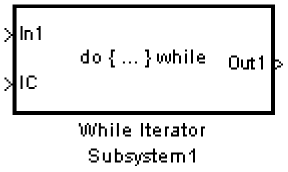


This block is a Subsystem block that is preconfigured to serve as a starting point for creating a subsystem whose execution is triggered by a Switch Case block. It contains the Action Port that links to a signal from a switch case block:

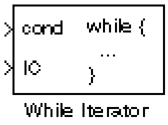
Note: The naming "Action" within this block is different to "Actions" which are connected to the SV Action concept

While Iterator Subsystem



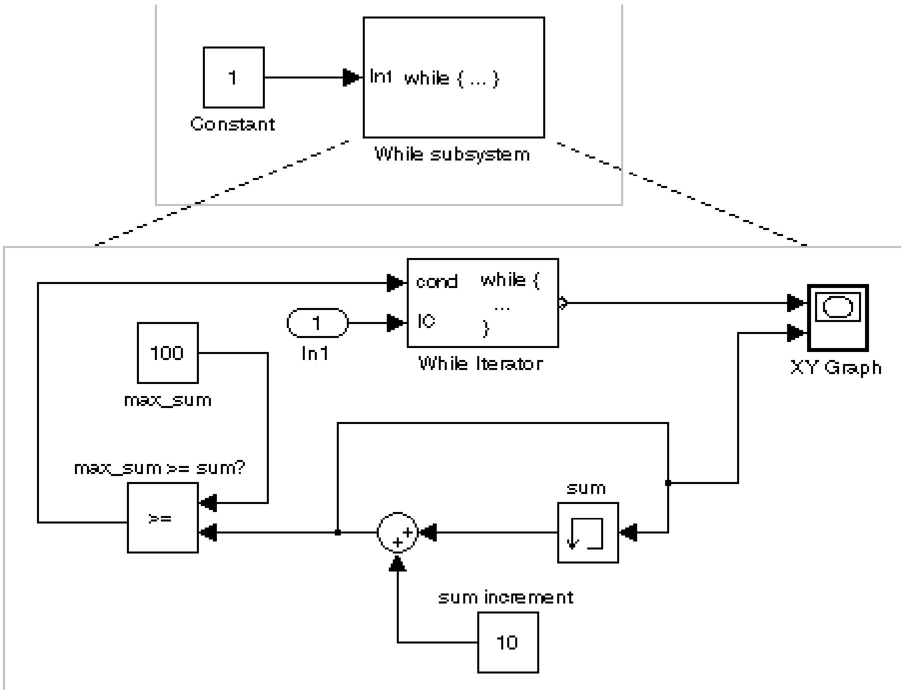


Represent a subsystem that executes repeatedly while a condition is satisfied during a simulation time step. This block is a Subsystem block that is preconfigured to serve as a starting point for creating a subsystem that executes repeatedly while a condition is satisfied during a simulation time step. It contains the While Iterator block.



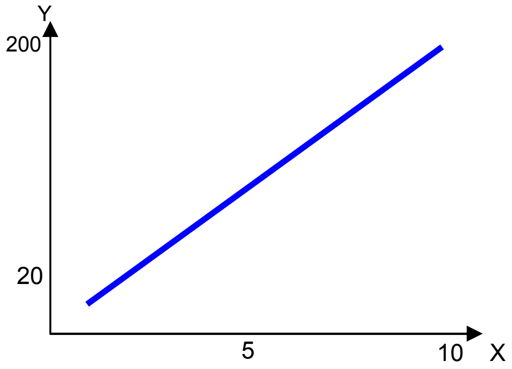
The While Iterator block, when placed in a subsystem, implements a C-like while or do-while control flow statement in Simulink as a While subsystem. It has iterative control over any accompanying Simulink block programming placed in the same subsystem with it.

For each iteration of the While Iterator block, the accompanying blocks perform one execution. Iteration takes place as long as the input conditions are true. This applies to an initial condition for the first execution (input port labeled IC) and a condition for succeeding executions (input port labeled cond). The following While subsystem example increments an initial value of 0 by 10 for every execution.



In the above example, a subsystem with a While block receives an input, which it passes to the IC (initial condition) port of the While block inside. If this value is true, the While block executes the blocks of the subsystem it is in. Since this value is 1 (true), the blocks execute and a value of 10 is added to a sum, which is initially 0. The sum is then compared to a value of 100. If the sum is less than or equal to 100, a value of true is passed to the While block through the cond (condition) port. This causes the blocks to execute again and again until the value passed to the While Iterator block is false and execution ceases.

In addition, for each time the blocks of the subsystem execute, the While block outputs a value equal to the number of times that the blocks have executed, including the current execution. This value, along with the sum value, is sent to the x and y coordinate inputs, respectively, of an XY Graph block with the following result.



The above while control flow statement example can be represented by the following pseudo code.

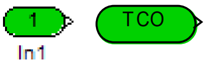
```
sum = 0;
IC = 1;
iteration_number = 0;
cond = IC;
while (cond != 0) {
    iteration_number = iteration_number + 1;
    sum = sum + sum_increment;
    if (sum >= 100 OR iterations > max_iterations)
        cond = 0;
}
```

The equivalent pseudo code for a do-while iteration in the example above (minus the IC port) is as follows:

```
sum = 0;

iteration_number = 0;
cond = 1;
do {
    iteration_number = iteration_number + 1;
    sum = sum + sum_increment;
    if (sum >= 100 AND iterations > max_iterations) cond = 0;
} while (cond);
```

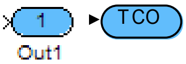
When you change the While Iterator block to do-while operation, the IC (initial condition) input disappears from the While Iterator block. The important distinction between the while and do-while modes of the While Iterator block is that the do-while mode runs the While subsystem at least once. In while mode, the While subsystem may not run its blocks at all depending on the value of the initial condition (IC).

Inport



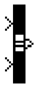
Inport blocks are the links from outside a system into the system. The signal name (here: TCO) can optionally be displayed on the input

Outport



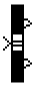
Outport blocks are the links from a system to a destination outside the system. The signal name (here: TCO) can optionally be displayed on the output.

Bus Creator



The Bus Creator block combines a set of signals into a bus, i.e., a group of signals represented by a single line in a block diagram. The Bus Creator block, when used in conjunction with the Bus Selector block, reduces number of lines required to route signals from one part of a diagram to another.

Bus Selector



Select signals from an incoming bus. The Bus Selector block accepts input from a Bus Creator block or another Bus Selector block. This block has one input port. The number of output ports depends on the state of the Muxed output check box. If you check Muxed output, then the signals are combined at the output port and there is only one output port; otherwise, there is one output port for each selected signal.

Mux

Combine several input signals into a vector or bus output signal. The Mux block combines its inputs into a single output. An input can be a scalar, vector, or matrix signal. Depending on its inputs, the output of a Mux block is a vector or a composite signal, i.e., a signal containing both matrix and vector elements. If all of a Mux block's inputs are vectors or vector-like, the block's output is a vector. A vector-like signal is any signal that is a scalar (one-element vector), a vector, or a single-column or single-row matrix. If any input is a nonvector-like matrix signal, the output of the Mux is a bus signal. Bus signals can drive only virtual blocks, e.g., a Demux, Subsystem, or Go To block.

Demux

Extract and output the elements of a bus or vector signal. The Demux block extracts the components of an input signal and outputs the components as separate signals. The block accepts either vector (1-D array) signals or bus signals.

Ground

The Ground block can be used to connect blocks whose input ports are not connected to other blocks. The Ground block outputs a signal with zero value. The data size of the signal is the same as that of the port to which it is connected.

Terminator

The Terminator block can be used to cap blocks whose output ports are not connected to other blocks.

Enable

Adding an Enable block to a subsystem makes it an enabled subsystem. An enabled subsystem executes while the input received at the Enable port placed at the top of the subsytem is greater than zero.

Trigger and function call

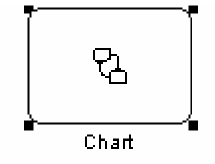
Adding a Trigger block to a subsystem makes it a triggered subsystem. A triggered subsystem executes once on each simulation step when the value of the signal that passes through the trigger port changes in a specifiable way (described below). A subsystem can contain no more than one Trigger block.

The Trigger type parameter allows you to choose the type of event that triggers execution of the subsystem:

- rising triggers execution of the subsystem when the control signal rises from a negative or zero value to a positive value (or zero if the initial value is negative).
- falling triggers execution of the subsystem when the control signal falls from a positive or a zero value to a negative value (or zero if the initial value is positive).
- either triggers execution of the subsystem when the signal is either rising or falling.
- function-call causes execution of the subsystem to be controlled by logic internal to another block (and the subsystem becomes a Function_Call_Subsystem, see description above).

36.6.20 Stateflow

Stateflow Chart



This block indicates a Stateflow Chart, which is mainly used to model state machines. The chart can be triggered and contain variable number of outputs and inputs.

36.6.21 Functional differences between blocks from SDA 5.0 and blocks from 4.0 and 3.1

Version information

- All SDA 5.0 blocks in Automotive Blockset show version information on the block icon. Every block has own version 5.XX.
- Blocks from SDA 4.0 and 3.1 releases do not show version information on the block icon.

Functional differences

All SDA 5.0 blocks from Automotive Blockset are compatible with the MSR specification (V. 1.12). The previous (4.0 and 3.1) block versions have different functional behavior, which is listed below:

1. Group 1: Blocks which perform reset at the first calculation step after Reset signal (R) becomes zero.
2. Group 2: Blocks which do not perform functional reset (R) correctly if on first step after reset the enable signal E is not zero. In this case, the calculation uses a new state value equal to current input IV instead of stored input IV during the functional reset.
3. Group 3: Blocks which do not assign value to output if enable (E) signal is equal to zero and no logical reset (R or RT) is requested. It can produce mistakes in multi-rate mode after R_FC function call.
4. Group 4: Blocks which have wrong simulation behavior.

Block	***1***	***2***	***3***	***4***	***Block***	***1***	***2***	***3***	***4***
CountDownRE	*x*	*x*			*DigitalLowpassRE*	*x*	*x*	*x*	
CountDownRTE					*HighpassTRE*	*x*			*x*
CounterRE	*x*	*x*			*LowpassKRE*	*x*	*x*	*x*	
CounterTRE					*LowpassSecOrdRE*	*x*	*x*		
StopWatchRE	*x*	*x*	*x*		*LowpassTRE*	*x*	*x*	*x*	
StopWatchTRE			*x*		*DeltaOneStep*	*x*	*x*		
TimerRetriggerRE	*x*	*x*			*DifferenceLimiter*	*x*	*x*	*x*	
TimerRetriggerRTE					*GradientLimiter*	*x*	*x*	*x*	
TimerRE			*x*		*DiffirenceQuotient*	*x*	*x*	*x*	
TimerRTE					*EdgeBi*	*x*	*x*	*x*	*x*
DelayResetEnabled	*x*				*EdgeFalling*	*x*	*x*	*x*	*x*
TurnOffDelaySample	*x*	*x*			*EdgeRising*	*x*	*x*	*x*	*x*
TurnOffDelayTime	*x*	*x*			*SampleAndHoldRE*	*x*	*x*	*x*	
TurnOnDelaySample	*x*	*x*			*MaxLogRE*	*x*			*x*
TurnOnDelayTime	*x*	*x*			*MinLogRE*	*x*			*x*
AccumulatorREL	*x*	*x*			*Hysteresis*	*x*		*x*	*x*
IntegratorKREL	*x*	*x*			*Delay_RE*				
IntegratorTREL	*x*	*x*							

5. TimerRE: Functional reset is different from MSR specification (see pseudo-code).

Pseudo-code from previous versions

This section lists the pseudo-code for blocks from previous versions (4.0 and 3.1) which have RE or RTE modes.

Standard ports:
- E: Suppresses the block calc.
- R: Resets to init values
- IV: Initial value set upon reset

Optional ports:
- State1_in: Direct input of internal state (x)
- State1_out: Direct output of internal state (x)
- R_FC: System reset request
- CLC_FC: System calculation request

CountDownRE

Port I/O Description
y: if counter state is zero
cnt: Out internal counter state

System-Init-Code
```
x = 0;
R1 = 0;
y = 0;
cnt = 0;
```

Init-Code (R_FC)
```
x = round(IV);
R1 = R;
```

Run-Code (CLC_FC)
```
if (R || R1) {
  x = round(IV);
}
else if (E) {
  if (x > 0){
    x = x - 1;
  }
}
y = (x > 0);
cnt = x;
R1 = R;
```

StopWatchRE

Port I/O Description
y Out Time since last reset
cnt: Out Internal counter state

Optional ports:
dT: Sample Time input

System-Init-Code
```
x = 0.0;
R1 = 0;
cnt = 0.0;
y = 0.0;
```

Init-Code (R_FC)
```
x = 0.0;
R1 = R;
```

Run-Code (CLC_FC)
```
if (R || R1) {
  x = 0.0;
}
else if (E) {
  x = x + dT;
}
y = x;
cnt = x;
R1 = R;
```

CounterRE

Port I/O Description
y: Out counter value
cnt: Out Internal counter state

System-Init-Code
```
x = 0;
R1 = 0;
cnt = 0;
y = 0;
```

Init-Code (R_FC)
```
x = round (IV);
R1 = R;
```

Run-Code
```
if (R || R1) {
  x = round(IV);
}
else if (E) {
  x = x + 1;
}
y = x;
cnt = x;
R1 = R;
```

CountDownRTE

Port I/O Description
y: Out counter value
cnt: Out Internal counter state

System-Init-Code
```
x = 0;
RT_1 = 0;
cnt = 0;
```

Run-Code
```
if (RT && !RT_1) {
  x = round(IV);
}
else if (E) {
  if (x > 0) {
    x = x - 1;
  }
}
```

```
y = 0;
```

Init-Code (R_FC)

```
x = round (IV);
RT_1 = 0;
```

```
RT_1 = RT;
y = (x > 0);
cnt = x;
```

CounterRTE

Port I/O Description

y: Out counter value
cnt: Out Internal counter state

System-Init-Code

```
x = 0;
cnt = 0;
y = 0;
```

Init-Code (R_FC)

```
x = round (IV);
RT_1 = 0;
```

Run-Code

```
if (RT && !RT_1) {
  x = round(IV);
}
else if (E) {
  x = x + 1;
}
RT_1 = RT;
y = x;
cnt = x;
```

StopWatchRTE

Port I/O Description

y Out Time since last reset
cnt: Out Internal counter state

Optional ports:

dT: Sample Time input

System-Init-Code

```
x = 0.0;
cnt = 0.0;
y = 0.0;
```

Init-Code (R_FC)

```
x = 0.0;
RT_1 = 0;
```

Run-Code (CLC_FC)

```
if (RT && !RT_1) {
  x = 0.0;
}
else if (E) {
  x = x + dT;
}
RT_1 = RT;
y = x
cnt = x;
```

Delay_RE

Port I/O Description

u: Source signal (real)
y: Output signal, delay by 1

System-Init-Code

```
x = 0;
y = 0;
```

Init-Code (R_FC)

Not implemented.

Run-Code (CLC_FC)

```
if (R) {
  y = IV;
  x = y = IV;
}
else if (E) {
  y = x;
  x = u;
}
else {
  y = x;
}
```

TimerRetriggerRE

Port I/O Description

y: Timer is running (time >0)

Init-Code (R_FC)

x = IV;

cnt: Out Internal counter state

Optional ports:
dT: Sample Time input

System-Init-Code
x = 0.0;
R1 = 0;
cnt = 0.0;
y = 0.0;

Run-Code (CLC_FC)
```
if (R || R1) {
  x = IV;
}
else if (E) {
  x = MAX(x-dT, 0.0);
}
y = (x > 0.0);
cnt = x;
R1 = R;
```

TimerRetriggerRTE

Port I/O Description
y: Timer is running (time >0)
cnt: Out Internal counter state

Optional ports:
dT: Sample Time input

System-Init-Code
```
x = 0.0;
cnt = 0.0;
y = 0.0;
```

Init-Code (R_FC)
x = IV;
```
RT_1 = 0;
```

Run-Code (CLC_FC)
```
if (RT && !RT_1) {
  x = IV;
}
else if (E) {
  x = MAX(x-dT,0.0);
}
RT_1 = RT;
y = (x > 0.0);
cnt = x;
```

TimerRE (V. 4.0)

Port I/O Description
y: Timer is running (time >0)
cnt: Out Internal counter state

Optional ports:
dT: Sample Time input

System-Init-Code
```
x = 0.0;
cnt = 0.0;
y = 0.0;
```

Init-Code (R_FC)
```
x = IV;
```

Run-Code (CLC_FC)
```
if (R && (x <= 0.0)) {
  x = IV;
}
else if (E) {
  x = MAX(x-dT, 0.0);
}
y = (x > 0.0);
cnt = x;
```

TimerRE (V. 3.1)

Port I/O Description
y: Timer is running (time >0)
cnt: Out Internal counter state

Optional ports:
dT: Sample Time input

System-Init-Code
```
x = 0.0;
cnt = 0.0;
y = 0.0;
```

Init-Code (R_FC)
```
if (x <= 0.0){
  x = IV;
}
```

Run-Code (CLC_FC)
```
if (R){
  if (x <= 0.0){
     x = IV;
  }
}
else if (E) {
  x = MAX(x-dT, 0.0);
}
y = (x > 0.0);
cnt = x;
```

DelayResetEnabled

Port I/O Description
u: Source signal (real)
y: Output signal, delay by 1

System-Init-Code

```
x = 0;
R1 = 0;
y = 0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R) {
  x = IV;
  y = x;
}
else {
  if (R1) {
    x = IV;
  }
  if (E) {
    y = x;
    x = u;
  }
  else {
    y = x;
  }
}
R1 = R;
```

TurnOffDelaySample

Port I/O Description
n: Number of delays
u: Source signal (boolean)
y: Falling edge u, delay by n

System-Init-Code

```
x = 0; /* counter */
y1 = 0; /* y */
y = 0;
R1 = 0;
```

Init-Code (R_FC)

```
x = round(n);
y1 = (u || (x>0));
if (!u && (x>0)) {
  x = x - 1;
}
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = round(n);
  y1 = (u || (x>0));
  if (!u && (x>0)) {
    x = x - 1;
  }
}
else if(E) {
  y1 = (u || (x>0));
  if (u) {
    x = round(n);
  }
  else if (x > 0) {
    x = x - 1;
  }
}
y = y1;
R1 = R;
```

TurnOffDelayTime

Port I/O Description
T: Delay time
u: Source signal (boolean)
y: Falling edge u, delay by T

Optional ports:
dT: Sample Time input

System-Init-Code

```
x = 0.0; /* timer */
y1 = 0; /* y */
y = 0;
R1 = 0;
```

Init-Code (R_FC)

```
x = T
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = T; /* reset */
  y1 =(u || (x>0.0));
  if (!u && (x>0)) {
    x = x - dT;
  }
}
else if (E) {
  y1=(u||(x>0.0));
  if (u) {
    x = T;
  }
  else if (x > 0.0) {
    x = x - dT;
  }
}
y = y1;
R1 = R;
```

TurnOnDelaySample

Port I/O Description
n: Number of delays
u: Source signal (boolean)
y: Rising edge u, delay by n

System-Init-Code
```
x = 0; /* counter */
y1 = 0; /* y */
y = 0;
R1 = 0;
```

Init-Code (R_FC)
```
x = round(n);
y1 = (u && (x<=0));
if (u && (x>0)) {
  x = x - 1;
}
R1 = R;
```

Run-Code (CLC_FC)
```
if (R || R1) {
  x = round(n);
  y1 = (u && (x<=0));
  if (u && (x>0)) {
    x = x - 1;
  }
}
else if(E) {
  y1 = (u && (x<=0));
  if (!u) {
    x = round(n);
  }
  else if (x > 0) {
    x = x - 1;
  }
}
y = y1;
R1 = R;
```

TurnOnDelayTime

Port I/O Description
T: Delay time
u: Source signal (boolean)
y: Rising edge u, delay by n

System-Init-Code
```
x = 0.0; /* timer */
y1 = 0; /* y */
y = 0;
R1 = 0;
```

Init-Code (R_FC)
```
x = T
```

Run-Code (CLC_FC)
```
if (R || R1) {
  x = T;
  y1=(u&&(x<=0.0));
  if (u&&(x>0.0) {
    x = x - dT;
  }
}
else if (E) {
  y1=(u&&(x<=0.0));
  if (!u) {
    x = T;
  }
  else if (x > 0.0) {
    x = x - dT;
  }
}
y = y1;
R1 = R;
```

AccumulatorREL

Port I/O Description
u: Input signal to accumulate
MX: Upper state limit
MN: Lower state limit
y: u accumulated (running sum)
B_max Out True if upper limitation active
B_min Out True if lower limitation active

System-Init-Code
```
x = 0;
R1 = 0;
y = 0;
B_min = 0;
B_max = 0;
```

Init-Code (R_FC)

Run-Code (CLC_FC)
```
if (R || R1) {
  x = IV;
}
else if (E) {
  x = x + u;
}
/* limit */
if (x > MX) {
  x = MX;
  B_max = 1;
  B_min = 0;
}
else if (x < MN) {
  x = MN;
  B_max = 0;
  B_min = 1;
```

```
x = IV;
if (x > MX) {
  x = MX;
}
else if (x < MN) {
  x = MN;
}
R1 = R;
```

```
}
else {
  B_max = 0;
  B_min = 0;
}
y = x; /* out */
R1 = R;
```

IntegratorKREL

Port I/O Description

u: Input signal to integrate
MX: Upper state limit
MN: Lower state limit
K: Gain
y: integrated value of u
B_max Out True if upper limitation active
B_min Out True if lower limitation active

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
B_min = 0;
B_max = 0;
```

Init-Code (R_FC)

```
x = IV;
if (x > MX) {
  x = MX;
}
else if (x < MN) {
  x = MN;
}
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
}
else if (E) {
  x = x + K*u*dT;
}
/* limit */
if (x > MX) {
  x = MX;
  B_max = 1;
  B_min = 0;
}
else if (x < MN) {
  x = MN;
  B_max = 0;
  B_min = 1;
}
else {
  B_max = 0;
  B_min = 0;
}
y = x; /* out */
R1 = R;
```

IntegratorTREL

Port I/O Description

u: Input signal to integrate
MX: Upper state limit
MN: Lower state limit
T: Time constant (1/Gain)
y: integrated value of u
B_max Out True if upper limitation active
B_min Out True if lower limitation active

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
B_min = 0;
B_max = 0;
```

Init-Code (R_FC)

```
x = IV;
if (x > MX) {
  x = MX;
}
else if (x < MN) {
  x = MN;
}
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
}
else if (E) {
  x = x + u*dT/T;
}
/* limit */
if (x > MX) {
  x = MX;
  B_max = 1;
  B_min = 0;
}
else if (x < MN) {
  x = MN;
  B_max = 0;
  B_min = 1;
}
else {
  B_max = 0;
  B_min = 0;
}
y = x; /* out */
R1 = R;
```

Optional ports:

dT: Sample Time input

DigitalLowpassRE

Port I/O Description
u: Source signal (real)
m: Filter factor
y: Filtered signal

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
}
else if (E) {
  x = x+ m*(u - x);
}
y = x;
R1 = R;
```

LowpassKRE

Port I/O Description
u: Source signal (real)
K: Filter factor
y: Filtered signal

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
}
else if (E) {
  x = x+ K*dT*(u-x);
}
y = x;
R1 = R;
```

HighpassTRE

Port I/O Description
IV1: IV (for timestep n) at reset
IV2: IV (for timestep n-1) at reset
u: Source signal (real)
T: Time constant
TD: Time constant
y: Highpass filtered signal

Optional ports:
State1_in: Direct input of internal state (x1)
State2_in: Direct input of internal state (x2)
State1_out: Direct output of internal state (x1)
State2_out: Direct output of internal state (x2)

System-Init-Code

```
x1 = 0.0;
x2 = 0.0;
R1 = 0;
y = 0.0;
```

Note: Wrong behavior on simulation if IV values are not constant.

Init-Code (R_FC)

```
x1 = IV1;
x2 = IV2;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R) {
  x1 = IV1;
  x2 = IV2;
}
else {
if (R1) {
  x1 = IV1;
  x2 = IV2;
}
if (E) {
  x1=x1+(TD*u-TD*x2-
     dT*x1)/T;
  }
}
y = x1;
x2 = u;
R1 = R;
```

LowpassSecOrdRE

Port I/O Description
IV1: IV (for timestep n) at reset
IV2: IV (for timestep n-1) at reset
u: Source signal (real)
T:Time constant
D: Damping constant
Y: Highpass filtered signal

Optional ports:
State1_in: Direct input of internal state (x1)
State2_in: Direct input of internal state (x2)
State1_out: Direct output of internal state (x1)
State2_out: Direct output of internal state (x2)

System-Init-Code

```
x1 = 0.0;
x2 = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x1 = IV1;
x2 = IV2;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x1 = IV1;
  x2 = IV2;
}
else if (E) {
temp1 =T*T+2*D*T*dT;
x1 = x1+[dT*dT*(u–x1)
     + T*T*(x1–x2)] /
     temp1;
}
y = x1;
x2 = x1;
R1 = R;
```

LowpassTRE

Port I/O Description
u: Source signal (real)
T: time constant
y: Filtered signal

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
}
else if (E) {
  x = x+ dT/T*(u-x);
}
y = x;
R1 = R;
```

DeltaOneStep

Port I/O Description
u: Source signal (real)
y: Difference signal

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
  y = u - x;
}
else if (E) {
  y = u - x;
}
x = u;
R1 = R;
```

EdgeBi

Port I/O Description
u: Source signal (boolean)
y: Edge pulse

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
  y = (x != u);
  x = u;
}
else if (E) {
  y = (x != u);
  x = u;
}
R1 = R;
```

EdgeFalling

Port I/O Description

u: Source signal (boolean)
y: Edge pulse

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
  y = (x && !u);
  x = u;
}
else if (E) {
  y = (x && !u);
  x = u;
}
R1 = R;
```

EdgeRising

Port I/O Description

u: Source signal (boolean)
y: Edge pulse

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
  y = (!x && u);
  x = u;
}
else if (E) {
  y = (!x && u);
  x = u;
}
R1 = R;
```

SampleAndHold

Port I/O Description

u: Source signal (real)
y: Captured signal
B_max: True if upper limitation active
B_min: True if lower limitation active

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
}
else if (E) {
  x = u;
}
y = x;
R1 = R;
```

Hysteresis

Port I/O Description
u: Source signal (real)
RSP: Upper threshold (real)
LSP: Lower threshold (real)
y: Hysteresis signal (boolean)

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
B_max = 0;
B_min = 0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
  y = x;
}
if(R || R1 || E){
  if (u > RSP) {
    x = 1;
  }
  else if (u < LSP) {
    x = 0;
  }
  y = x;
}
R1 = R;
```

DifferenceQuotient

Port I/O Description
u: Source signal (real)
y: Rate of change (gradient)

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
  y = (u - x)/dT;
  x = u;
}
else if (E) {
  y = (u - x)/dT;
  x = u;
}
R1 = R;
```

DifferenceLimiter

Port I/O Description
u: Source signal (real)
LU: Upper limit of difference (real)
LD: Lower limit of difference (real)
y: Limited signal (real)
B_min: Lower limit exceeded (boolean)
B_max: Upper limit exceeded (boolean)

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
B_max = 0;
B_min = 0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
  y = x;
}
else if (E) {
if ( (u - x)> LU ) {
  x = x+ LU;
  B_min = 0;
  B_ max = 1;
}
else if ( (u - x)< LD ) {
  x = x+ LD;
  B_ min = 1;
  B_ max = 0;
}
else {
x = u;
  B_ min = 0;
  B_ max = 0;
}
  y = x;
}
R1 = R;
```

DifferenceLimiter

Port I/O Description
u: Source signal (real)
dLU: Upper gradient [unit(u)/s] (real)
dLD: Lower gradient [unit(u)/s] (real)
y: Limited signal (real)
B_min: Lower limit exceeded (boolean)
B_max: Upper limit exceeded (boolean)

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
B_max = 0;
B_min = 0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
if (R || R1) {
  x = IV;
  y = x;
}
else if (E) {
  if ((u-x)>dLU*dT) {
    x = x+ dLU * dT;
    B_min = 0;
    B_max = 1;
  }
  else if((u-x)<
         dLD*dT)
  {
   x = x+ dLD * dT;
    B_min = 1;
    B_max = 0;
  }
  else {
    x = u;
    B_min = 0;
    B_max = 0;
  }
  y = x;
}
R1 = R;
```

MaxLogRE

Port I/O Description
u: Source signal (real)
y: Maximum value
B_max: Current input is new max. (boolean)

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
B_max = 0;
```

Init-Code (R_FC)

```
x = IV;
R1 = R;
```

Run-Code (CLC_FC)

```
B_max = 0;
if (R) {
  x = IV;
}
else {
  if (R1) {
    x = IV;
  }
  if (E) {
    if (u > x) {
      x = u;
      B_max = 1;
    }
  }
}
y = x;
R1 = R;
```

MaxLogRE

Port I/O Description
u: Source signal (real)
y: Minimum value
B_min: Current input is new min. (boolean)

System-Init-Code

```
x = 0.0;
R1 = 0;
y = 0.0;
```

Run-Code (CLC_FC)

```
B_min = 0;
if (R) {
  x = IV;
}
else {
  if (R1) {
    x = IV;
  }
```

`B_min = 0;`
Init-Code (R_FC)
`x = IV;`
`R1 = R;`
	See code below

```
  if (E) {
    if (u < x) {
      x = u;
      B_min = 1;
    }
  }
}
y = x;
R1 = R;
```

36.7 How to read SDA based specifications

36.7.1 Introduction

The aim of this document is to enable the reader to understand the formalism of a SDA-model based specification. It is dedicated to customers or users dealing with the graphical specification.

SDA (System Design Automation) is the Matlab/Simulink/Stateflow[1] based function development environment used at Siemens VDO Automotive AG.

To understand the benefit of this new specification structure and formalism, this document first gives a very brief overview to a SDA model. SDA models are set up as Simulink/Stateflow models and have a clearly defined and unique structure as described in the following sections. Each SDA based specification is based on part of the complete and executable Model.

36.7.2 Matlab/Simulink/Stateflow.

This subchapter describes with a few words the tool used for modeling the behavior of a complete ECU. This tool is called Matlab. Matlab abbreviates "Matrix Laboratory" and has been developed for Technical Computing.

Matlab is:

- an interactive programming environment
- a high performance language
- a developers tool and
- a library of mathematical functions.

There are various software tools featured by Matlab the so-called **Matlab-Toolboxes**. They belong to the Matlab product family and are application-specific libraries of Matlab functions that customize Matlab for solving particular classes of problems. One of these toolboxes is Simulink.

Simulink is a graphical mouse-driven program for:

- the modeling of linear and non-linear dynamic systems,
- the simulation of them
- and of their analysis.

Simulink models are developed and parameterized as block diagrams. Pictures of these block diagrams are used to describe the functionality in the automatic generated specification.

Simulink-Blocksets are application-specific libraries for Simulink models. One of these blocksets is Stateflow.

Stateflow is a special blockset of Simulink for the design of:

- complex control structures,

---

[1] Matlab, Simulink, and Stateflow are registered trademarks of TheMathworks, Inc.
Other product or brand names are trademarks or registered trademarks of their respective holders.

- i.e. for event driven systems,
- for their simulation and
- analysis.

Stateflow models are developed and programmed as state machines. Pictures of these state machines are used to describe the functionality in the automatic generated specification.

36.7.3 SDA model overview

Model

Although the final specification represents only a sub part of a complete SDA model, this section gives a short overview of a complete SDA - model.

A SDA model consists of five main parts:

- Stimuli (STI_PRJ_AGGR)
  The stimuli subsystem provides all signals and data inputs needed for the simulation of the model

- Display (DISP_PRJ_AGGR)
  In the display subsystem all outputs of the models are collected and visualized or saved to the Matlab Workspace.

- Plant (PLA_PRJ_AGGR)
  The plant contains a model of the system that has to be controlled by the Controller Unit. The behavior of the plant can be bypassed with input data in the Stimuli.

- Controller (PRJ_AGGR)
  The Engine Control Unit (ECU) holds the functionality of the controller. It is subdivided into aggregates (see below) which are again subdivided into modules.

For each module a separate specification is generated.

Aggregates

An aggregate holds the functionality of a specific function together with its inputs and outputs. The main part of an aggregate in the SDA environment is the subsystem called "*modules_<AGGR>*", where <AGGR> stands for the aggregate name (see Figure 1). In Simulink a subsystem is a graphical mean to group any portion of a model.
The subsystem "*modules_AGGR*" includes all modules of the aggregate. The modules themselves are once again subsystems. The signal flow between the modules is represented by signal lines and signal buses. Bus selectors are used to select and distribute the appropriate signal to the modules. All data inputs and outputs are defined on this level of the aggregate as shown in the following figure. In future this figure will be used in the specification to give an overview of an aggregate.

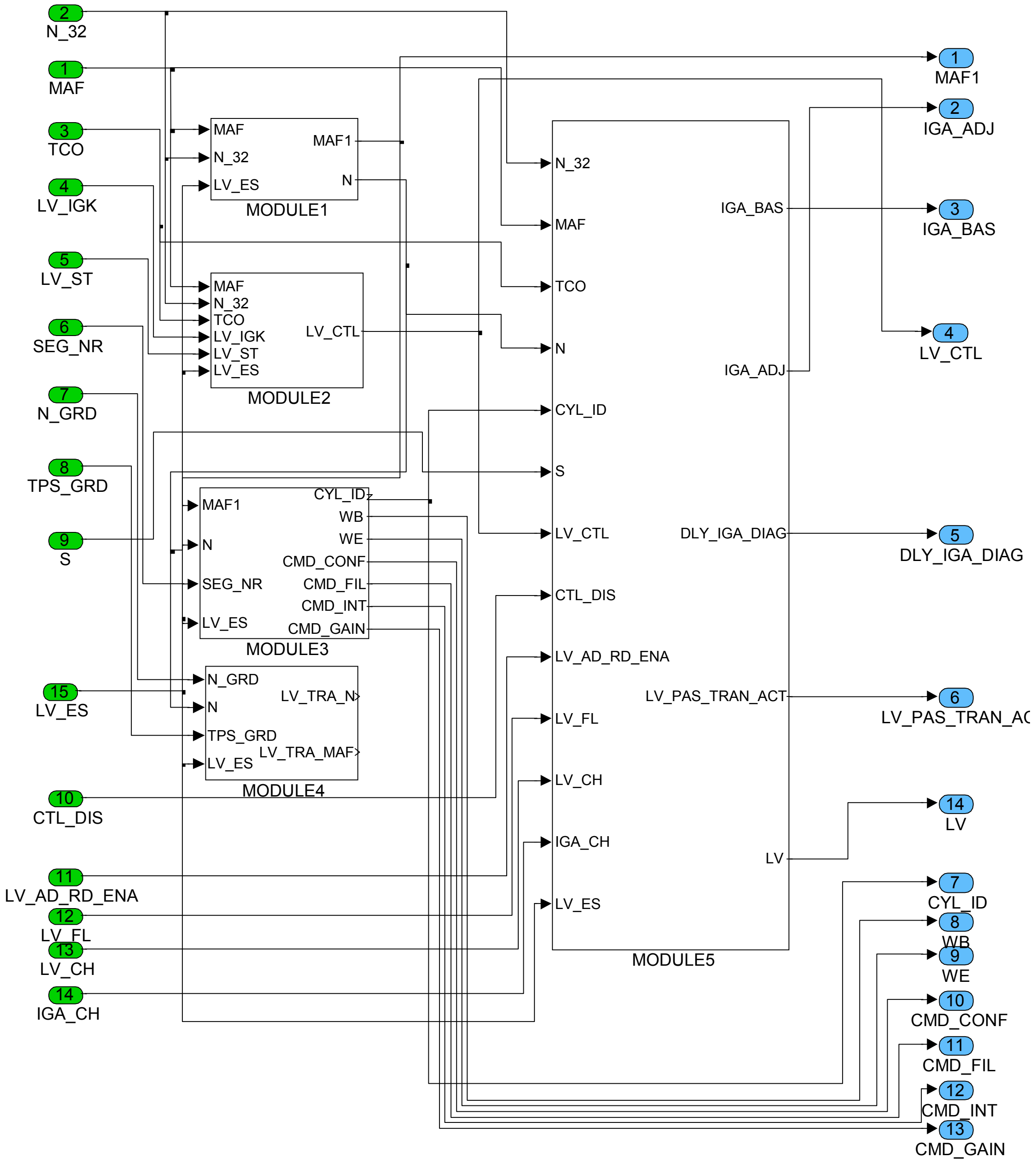


Figure 1 - Schematic picture of the content of an aggregate. This picture shows the subsystem named "modules_<AGGR>" including all the five modules (Module1 to Module5) of the aggregate. For each module a separate specification is generated.

Modules

A module is the lowest level modular component in the model. All modules are included in the subsystem "modules_AGGR". The whole functionality of the model is placed inside triggered subsystems within the modules. There is no restriction to the number of subsystem levels. Content of the subsystems in the modules is shown in the graphical software requirement specification (SDA-SRS). This means that only the part relevant for the calibration is shown in the pictures in the SRS. Beside the pictures there is also some textual description for the separate functionality. These text explanations are directly followed by their respective pictures, as shown in the graphical software requirement specification (SDA-SRS).

36.7.4 SDA specific Simulink notation

The functionality of the modules is build up with Simulink blocks that are placed inside triggered subsystems. The pictograms on the Simulink blocks represent a special functionality.

The blocks that are allowed for modeling within the SDA environment are included in separate libraries:

- the AT_Basic_Blocks ("Automotive Blockset")
- the AT_Lib library

In "Documentation Automotive Blockset" and "Documentation AT_Lib" a short description of the different blocks is given. The blocks are connected by signal lines, which represent the signal flow between them. Moreover the control flow in the model is driven by function calls that trigger the different subsystems, i.e. a function call is a signal that initiates the calculation of a functionality grouped in a subsystem.

SDA generated SRS include some special layout features that are listed here and demonstrated in the example in Figure 2:

- Thin lines mark scalar signals.
- Thick lines mark multidimensional signals e.g. arrays, vectors...
- Threefold lines.

Threefold lines represent bus signals, i.e. different signals that are collected together and represented by one "bus"-signal. This is similar to a data structure in Software. The block "bus creator" is used to collect the data "signal2" from the input "In1" and the data "Signal1" from the block "Constant1".

The block "bus_selector" is used to select dedicated signals from the bus. An example is shown in the figure with "Signal1" and "signal2". This example shows also that:

- Dashed lines represent function calls or triggers. In the example a so-called "Function-call generator" is used to initiate the evaluation of the constant block "Constant1" which write the array [1,2,3] to its output.
- Instead of signal lines local Goto/From blocks can be used to connect two subsystems as a kind of invisible signal line.
- Signal names are written beside the signal line.
- Input ports (e.g. the block "In1") and output ports (the block "Out1" or "Out2") provide the interface of the subsystem to the environment, i.e. they show the input and output data of a subsystem.

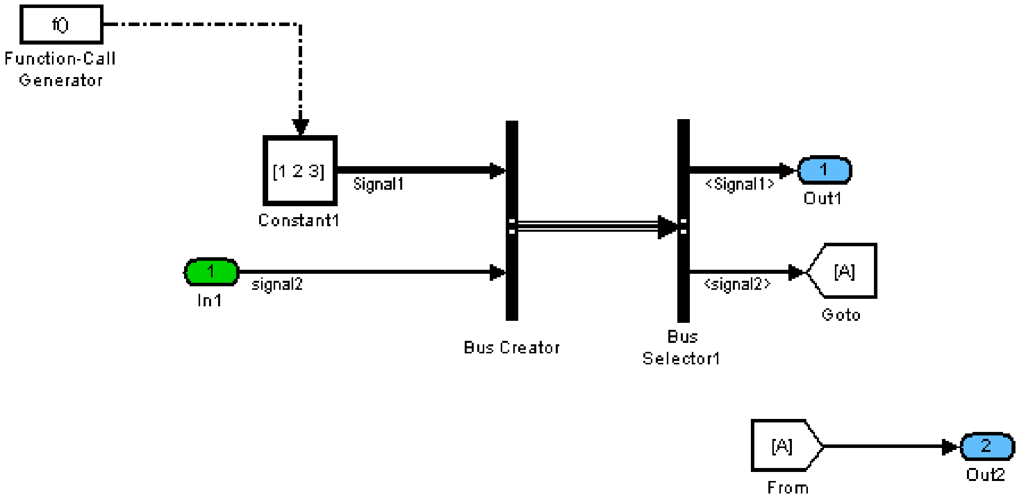
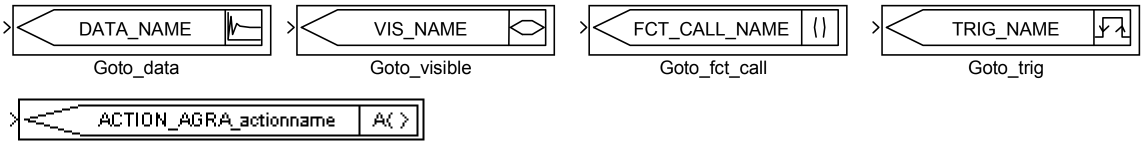


Figure 2 - Example

- The port names are shown on the respective subsystem (in this example: "In1", "Out1" and "Out2".
- There exist different **global** Goto/From blocks for data signals, visibles, function calls and triggers. E.g. the Goto blocks:



- All visibles are available via the application system.
- All goto/from data are outputs/inputs of modules.
- If a subsystem has an input signal that is also an output signal of this subsystem, the suffixes "_in" are added to the input port name and "_out" to the output port name.

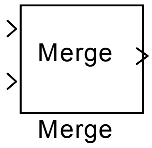
This means for example if the signal "A1" is both, input and output data of a subsystem, the input port is named "A1_in" and the output port "A1_out".

36.7.5 Structure of the SDA-Specification

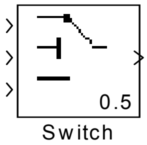
The SDA-SRS layout is based on the layout of the standard WinWord Specifications. The first section includes tables of all output, input, calibration and configuration data. For output data the name, mode, hex. limits, phys. limits, resolution and unit is given in the table, whereas for the calibration and configuration data the dimension is given instead of mode. The input data table only holds the names of the variables.

An optional textual description and an overview picture of the module follow this section. Afterwards the contents of the subsystems with optional textual descriptions are included in hierarchical order. The pictures in a SRS are always shown after their respective text description.

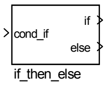
36.7.6 Special Simulink blocks and their usage



The **Merge block** combines its inputs into a single output line. The output value at any time is equal to the most recently computed output of its driving blocks. The Merge block mainly is used to merge a "normal" operate signal with a reset signal. In order to achieve the right performance it has to be taken into account the execution order of its driving blocks.



The **Switch block** propagates one of two inputs to its output, depending on the value of a third input, called the control input. If the signal on the control (second) input is greater than or equal to the Threshold parameter (shown in the block) the first input; otherwise, it propagates the third input.



The **if-then-else block** provides the possibility to implement an if-then-else control structure in Simulink. The input of this block is the condition "cond_if". If the condition is true (= 1) then the function call "if" is sent out. Else the function call "else" is sent out. These function calls can be used to trigger the corresponding subsystems.

36.7.7 How to read SDA conform Stateflow charts

Introduction

The aim of this chapter is to enable the reader of SDA based specifications to understand the Stateflow charts used in SDA models. This document is not a Stateflow reference, it shows only the usage according to the SDA Guidelines.

Stateflow objects overview

Graphical objects

Name	**Description**[2]
State	States are the primary objects of Stateflow. They represent modes of a system.
Transition	A transition is a pathway for a chart or state to change from one mode (state) to another.
Default transition	Default transitions tell Stateflow which of several possible states to enter first for a chart or superstate.
Connective junction	A connective junction represents a decision point between alternative transition paths.
History junction	A history junction records the most recently active state of the chart or superstate in which it is placed.
Function	Functions that are graphically defined by a flow graph. The flow graph is contained inside a box with the function name.
Box	Boxes are used to group parts of a diagram.

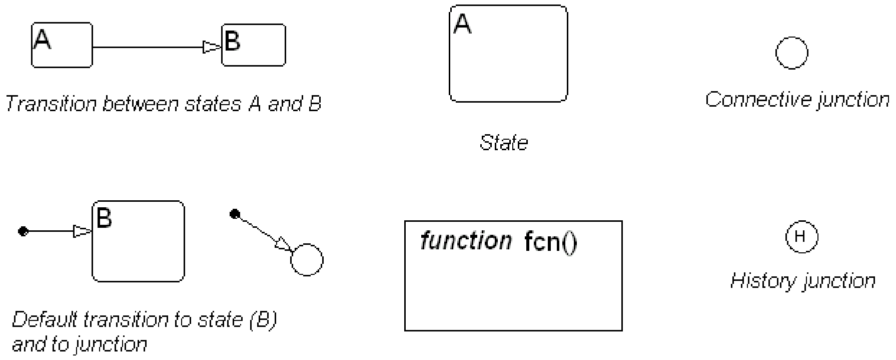


Figure 3 - Symbols used to represent graphical Stateflow objects

Non graphical objects:

Data object: data used in a state chart to control its execution. It can be created in the Stateflow chart itself (local data) or be imported from the Simulink model that calls the chart (input data). Simulink models can also read the Stateflow data, when it is declared as an output from the chart (output data).

Event object: an event can trigger whole Stateflow charts or individual actions in a chart. A Stateflow chart reacts on events and can broadcast events to its internal objects or send events to the Simulink model. Events are triggers or function calls.

Interface to Simulink

Figure 4 shows the interface between a Stateflow chart and the Simulink model. As shown, function calls and triggers can awake the chart. The chart can also send events to the model, use data as inputs and produce data as outputs.

The chart runs only one event input at each time.

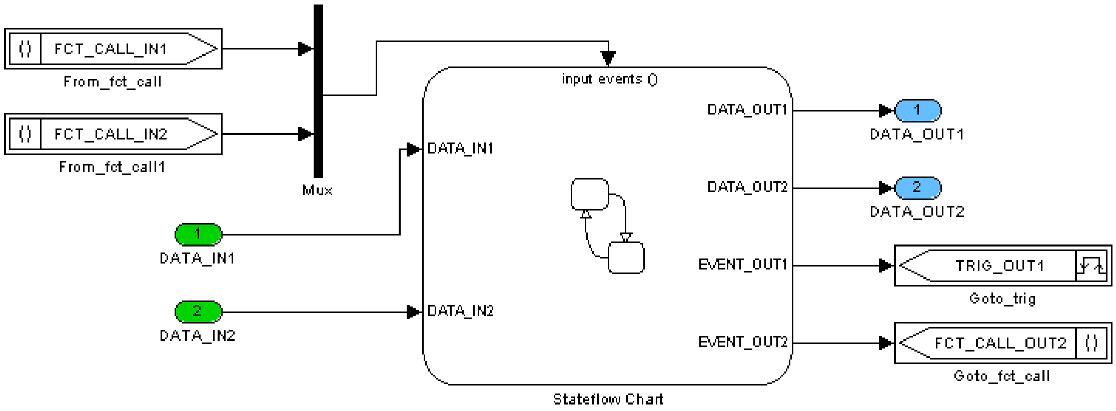
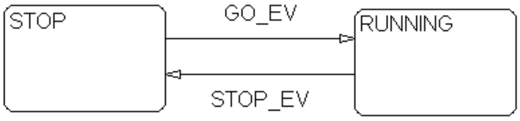


Figure 4 - Interface between a Simulink model and a Stateflow chart

States

A state describes a mode of a reactive Stateflow chart. The chart always has one or more active states. An active state tells that the chart is in this state's mode. Figure 5 shows a chart that can assume two modes: stop and running.



Assume: State "STOP" is at the beginning active.

Chart reaction on certain events:

1. Event "STOP_EV" awakes the chart …nothing happens.
2. Event "GO_EV" awakes the chart.
   "STOP" is deactivated.
   "RUNNING" is activated.
3. Event "STOP_EV" awakes the chart.
   "RUNNING" is deactivated
   "STOP" is activated.

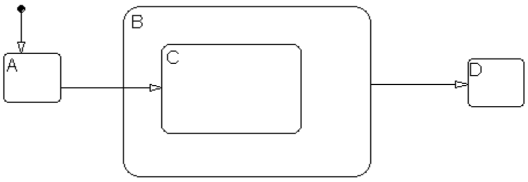
Figure 5 – Transition between two states

Activation

A transition between two states, state A to state B, changes state A to inactive and state B to active. A transition occurs always from an active state to an inactive state. The transitions are driven by events and controlled by conditions. The whole transition path between both active and inactive states must be evaluated and be valid so that the transition can occur.

Hierarchy

Every state is part of a hierarchy. States can be superstates, substates or just states. A state is a superstate if it contains other states, which are called substates. When a superstate turns active, one of its substates - if existent - turns also active. In the same way, when a substate turns active, it makes it superstate also active. See in Figure 6 some examples. The Stateflow chart itself is, in a SDA model, the highest hierarchical element.



The chart is awaken for the first time:
Default transition occurs and state A is activated.

The chart is awaken for the second time:

1. Transition between A and C is valid
2. A is deactivated
3. Transition occurs
4. B is activated
5. C is activated

The chart is awaken for the third time:

1. Transition between B and D is valid
2. C is deactivated
3. B is deactivated
4. Transition occurs
5. D is activated

Figure 6 – Hierarchical relation between states

Decomposition

Each state can be decomposed in substates. There are two kinds of decomposition: AND and EXCLUSIVE OR. See an example on Figure 7.

Parallel AND decomposition: one or more substates can be active at the same time. The substates are drawn with dashed borders.

Parallel EXCLUSIVE OR decomposition: only one substate can be active. The substates are drawn with continuous borders.

Substates of an superstate with parallel AND decomposition are drawn with dashed borders. When such a superstate activates, its substates are then activated in the order they appear - beginning at the top left side and finishing at the bottom right side. Only superstates with AND parallel decomposition can have two or more active substates at the

same time.

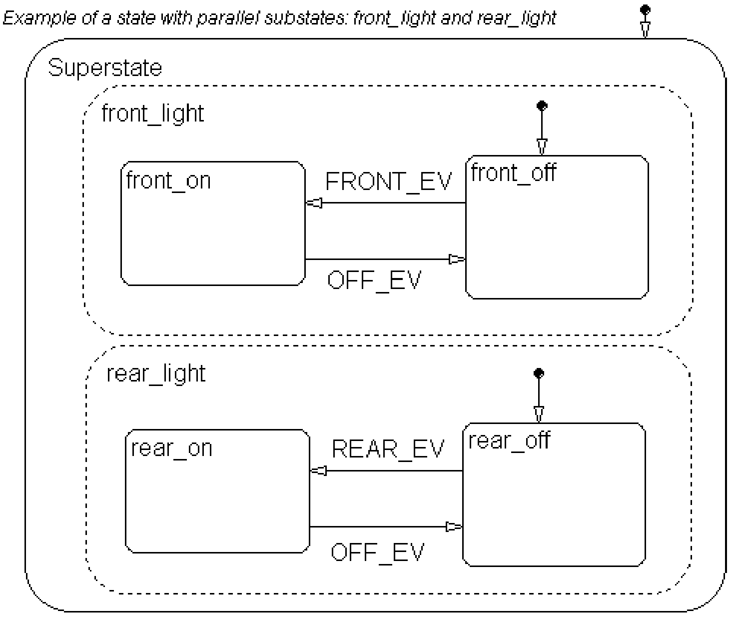


Figure 7 - Example of parallel state decomposition

This example shows a chart with three hierarchy levels:

1. First level: is the chart itself, contains the state "superstate" inside.
2. Second: content from state "superstate".
3. Third: content from the states "front_light" and "rear_light"

Decomposition:

- "superstate" has AND decomposition.
- "front_light","rear_light" and the chart itself have EXCLUSIVE OR decomposition.

Activation order in the AND decomposition:

- "front_light" is activated first ("1") and then "rear_light" ("2"). Both are active during the same time.
- The deactivation of parallelal substates occurs in the inverse order.
- In this example, if "superstate" will be deactivated, first "rear_light" will is deactivated, than "front_light" and finally "superstate".

History

A state also has history that applies to its level of hierarchy in the Stateflow diagram. When a superstate deactivates, it implies that its substates are also turning deactivated. When a history junction is present, activating the superstate again will reactivate the same substate that was activated before the last deactivation.

Stateflow actions

Stateflow action type

Actions are statements with mathematical expressions or event broadcasting. The Figure 8 shows some examples of actions.

In SDA Stateflow charts, actions can be either event broadcasting or value assignment. If a data is an input from Simulink, it can be used only on right hand side operations (is not possible to assign new values to it). If a data is an output to Simulink or a local data, it can be used on both sides. Multiple actions can be ordered in a list, separated with semi-colons to each other. The actions in a list are executed in the order they appear.

State actions

Four action types can be specified for one state: entry:, during:, exit:, or on: <<event_name>>

actions. This actions are specified in the state label, after the state name followed by a slash:
STATE_NAME/
entry: << action list >>; /* Actions are executed right after the state activation. */
during: << action list >>; /* Actions are executed if the state is active and there is no valid transition to another state. */
exit: << action list >> /* Actions are executed right after the state deactivation. */
on <<event_name>>: << action list >> /* Actions are executed when the specified event occurs */

Note: entry: and exit: actions are executed always after a transition has occurred. A during: action is execution as long as no transition has occurred. Entry, during and exit are also referred as en:, du: and ex:.

Transition evaluation

Transition path

A transition path is a sequence of transitions joined by connective junctions. A path doesn't necessary lead to a valid transition between two states.

Evaluating a transition doesn't mean that this transition will occur. A transition (between two states) occurs only if the hole path is valid. Nevertheless, the evaluation of all possible transitions follows always the sequence established by their precedence rules.

*That means, if a transition containing a condition statement is valid, its condition action (if it exists) runs at the evaluation time, without taking care if this transition is in a path that will lead to a transition between states or not. The same is not true for transition actions: they run only if the whole path containing them is true and also only **after** the transition between both states (initial and final) has occurred.*

Transition label

The switching from one state to another may be driven by events (function call, trigger) or conditions that are labeled to the transition arrow. The label notation has the following syntax:

event_name[condition_statement]{condition_ actions;}/transition_ action;

All three parts of the label notation are optional: event_name, transition action, and the set condition expression plus condition action. It is not obligatory to specify a condition action for a condition expression.

*A transition action happens only if the whole path containing the transition is true. The action happens after the transition has occurred. Transition actions are declared after a slash at the end of the transition label. Transition actions are **not** allowed in SDA transition action labels.*

The condition label [condition_statement] is evaluated as an expression with Boolean result. A transition containing a condition statement can only be valid if the condition result is true.

A condition action happens whenever a condition is evaluated and is true. The action *happens right after the condition evaluation. Condition actions are declared inside brackets {<< action list >>} after the condition declaration.*

A condition statement can contain a call to an *In(<<state_name>>)* function, that is evaluated true when the state *"state_name"* is active. See the tables at the end of this documentation for more information about operations precedence and syntax. See in Figures 5, 7 and 8 some transition label examples.

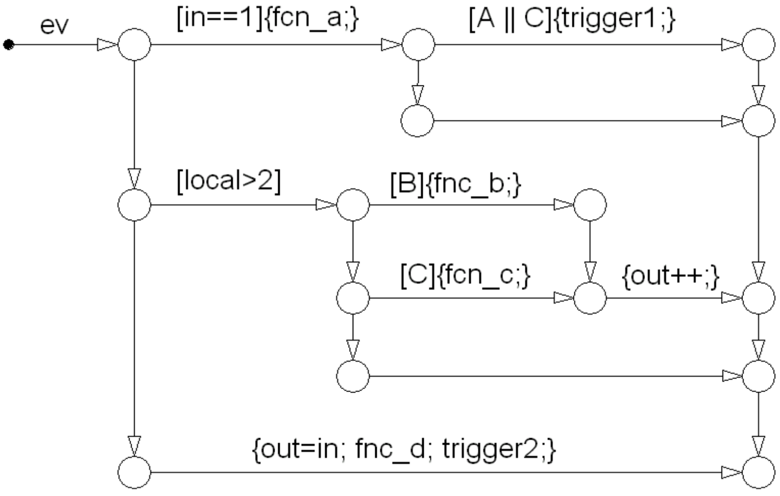
Precedence rules for transition evaluation

The evaluation of transitions coming from a single source obeys the following precedence order:

1. **End point hierarchy:** *transitions whose end points are attached to higher hierarchical levels are evaluated first.*
2. **Label:** *transitions are ordered for evaluation according to the types of action language present in their labels, with precedence as follows:*
   2.1. Labels with events and conditions.
   2.2. Labels with events
   2.3. Labels with conditions
   2.4. No label
3. **Transition angular position in its source:** *evaluation order based on the angular position of the transition on the surface of the originating object:*
   A - Multiple outgoing transitions from states
   Are evaluated in a clockwise progression starting at the upper left corner of the source state.
   B - Multiple outgoing transitions from connective junctions
   Are evaluated in a clockwise progression starting at the twelve o'clock position on the junction.

Figure 8 - Example: if - then - else construction

The following flow chart was made observing the SDA guidelines:



Events:
ev, trigger1, trigger2, fnc_a, fnc_b, fnc_c, fnc_d

Data:
in, out, local, A, B, C, D

Conditions present in the chart:
ev, [in==1], [A||C], [B], [C], [local>2]

Actions present in the chart:
{fnc_a;}, {trigger1;}, {fnc_b;}, {fnc_c}, {out++}, {out=in, fnc_d; trigger2;}

Attempting the precedence rules between transitions, a possible representation of this flow chart in c-like pseudo-code could be (this is not a suggestion!):

```
transition_occurred_flag=0;
if(arriving_event==ev){
      if(in==1){
            fcn_a();
            if(A||C){
                  transition_occurred_flag=1;
                  trigger(trigger1);
            }
            if(1)
                  transition_occurred_flag=1;
      }
      if(!transition_occurred_flag && local>2){
            if(B){
                  transition_occurred_flag=1;
                  fnc_b();
            }
            if(!transition_occurred_flag && C){
                  transition_occurred_flag=1;
                  fnc_c();
            }
            if(transition_occurred_flag)
                  out++;
            }
            if(1)
                  transition_occurred_flag=1;
      }
      if(!transition_occurred_flag){
            out=in;
            fnc_d();
            trigger(trigger2);
            transition_occurred_flag=1;
      }

}
```

Inner transition

An inner transition is a transition that does not exit the source state. It is represented as an internal transition from a superstate to a substate or to the superstate itself. When a inner transition occurs, it has the same result as when the superstate was deactivated and then reactivated, but the state remains active all the time.

The inner transition is part of the state it belongs to. It is evaluated only during this state, like other transitions inside the same state.

That means: when the inner transition occurs, the active substate exits and a new substate is activated. The new active substate is chosen by the inner transition. When it points to the superstate itself, the rules are the same as if the superstate is being activated from an external state. In this case, a default transition inside the superstate or a history junction will take effect. In Figure 9, the transitions 1, 3 and are inner transition of state A. Transition 4 is a inner transitions of state B. Transition 2 is self-loop transition, not a inner transition.

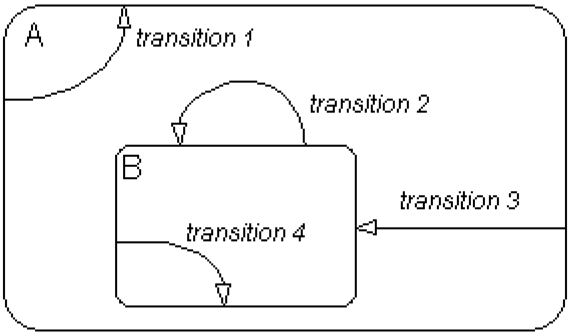


Figure 9 - Inner transition examples: transitions 1, 3 and 4.

Direct event broadcast

It is possible to broadcast an event directly to another state in the same chart (local event) using the action command *send*:

send(event_name,state_name): Broadcasts the event "*event_name*" to the state "*state_name*" and any of its offspring in the hierarchy. "state_name" must be already active in order to receive the sent event.

Notes

1. Matlab, Simulink, and Stateflow are registered trademarks of TheMathworks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.
2. From the Matlab, Simulink and Stateflow documentation.

Tables

The following tables show the possible Stateflow operations used to write statements (Stateflow action language) and conditions. The precedence order must be considered.

Assignment operations

Example	Description[2]
`a = expression`	Simple assignment
`a += expression`	Equivalent to `a = a + expression`
`a -= expression`	Equivalent to `a = a - expression`
`a *= expression`	Equivalent to `a = a * expression`
`a /= expression`	Equivalent to `a = a / expression`

In C-like bit operations:

Example	Description[2]
`a |= expression`	Equivalent to `a = a | expression` (bit operation).
`a &= expression`	Equivalent to `a = a & expression` (bit operation).
`a ^= expression`	Equivalent to `a = a ^ expression` (bit operation).

Unary operations

Example	Description[2]
~a	Logical NOT of a
!a	Logical NOT of a
-a	Negative of a

Unary actions

Example	Description[2]
a++	Increment a
a--	Decrement a

Binary and Bitwise operations

Example	Precedence	Description[2]
a * b	10	Multiplication
a / b	10	Division
a %% b	10	Modulus
a + b	9	Addition
a - b	9	Subtraction
a >> b	8	Shift operand a right by b bits. Noninteger operands for this operator are first cast to integers before the bits are shifted.
a << b	8	Shift operand a left by b bits. Noninteger operands for this operator are first cast to integers before the bits are shifted.
a > b	7	Comparison of the first operand greater than the second operand
a < b	7	Comparison of the first operand less than the second operand
a >= b	7	Comparison of the first operand greater than or equal to the second operand
a <= b	7	Comparison of the first operand less than or equal to the second operand
a == b	6	Comparison of equality of two operands
a ~= b	6	Comparison of inequality of two operands
a != b	6	Comparison of inequality of two operands
a <> b	6	Comparison of inequality of two operands
a & b	5	Logical AND of two operands or Bitwise AND of two operands
a ^ b	4	Operand a raised to power b or Bitwise XOR of two operands
a | b	3	Logical OR of two operands or Bitwise OR of two operands
a && b	2	Logical AND of two operands
a || b	1, last precedence	Logical OR of two operands

36.8 Initialization of variables from non implemented functions

All defined LV_ERR_XXX shall not be given ENVD data for the .a2L file (normally automatically defined)

EXCEPT: LV_ERR_TEG_PCAT_DOWN[NC_CBK_EX_NR]
LV_ERR_EGR

36.8.1 Miscellaneous

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CYC_ST	O/V	0...FFH	0...255	1	[-]
cycle counter at start and after start					
F_reib_ba	-	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for operation mode transition					
GR_SUB_MT	O	0...7H	0...7	1	[-]
Substitute gear ratio with error					
IGA_TRA_KNK	O	0...80H	-48...0	0.375	[°CRK]
Spark retard till high MAP_DRV1 applied to engnie					
IGC_x_EXT_ADJ	O	FA60...5A0H	-90...90	0.0625	[°CRK]
External Ignition Adjustment					
LAMB_AV_COR	O	0...7FFFH	0...31.99902	0.9766e-3	[-]
average lambda value					
LV_CMB_TRAN_ACT	O	0...1H	0...1	1	[-]
Combustion mode transition active					
LV_DI_TQ_REQ_CAN_MPI_GDI	O	0...1H	0...1	1	[-]
logical variable for disabling ARS intervention due to EMS error					
LV_EFPPWM_EXT_ADJ	O	0...1H	0...1	1	[-]
Logical variable indicating external controling of EFPPWM by service tool					
LV_EOL_OBD	O	0...1H	0...1	1	[-]
Flag which indicates that EOL is required					
LV_FAN_VAR_AD	O	0...1H	0...1	1	[-]
-					
LV_FPAPWM_DIAG_AFS_REQ	O	0...1H	0...1	1	[-]
hom. mod is requested by fuel pressure pwm diagnosis					
LV_FUP_HOM_REQ	O	0...1H	0...1	1	[-]
Anforderung Diagnose-Raildruck Homogen					
LV_IGC_x_EXT_ADJ	O	0...1H	0...1	1	[-]
External Adjustment for Ignition Angle					
LV_INH_CYL_BAL_ER_LIH_CTL	O	0...1H	0...1	1	[-]
Inhibit engine roughness adaptation because limp home mode					
LV_INH_CYL_BAL_LAM_LIH_CTL	O	0...1H	0...1	1	[-]
Inhibit cylinder individual lambda adaptation because limpe home mode					
LV_INH_S_MAN	O	0...1H	0...1	1	[-]
manual inhibiting startified mode by application system					
LV_ISC_INH_EXT_ADJ	O	0...1H	0...1	1	[-]
Logical variable to switch off the idle speed controller					
LV_NT_AGI_INI_EXT_ADJ_NEW_CAT	O	0...1H	0...1	1	[-]
initialization signal for a new cat via scan tool					
LV_NT_AGI_INI_EXT_ADJ_NEW_ECU	O	0...1H	0...1	1	[-]
initialization signal for a new ECU via scan tool					
LV_S_REQ_EGR	O	0...1H	0...1	1	[-]
Logical constant stratified mode requested for EGR LIH					
LV_STST_STOP_CYC	O/V	0...1H	0...1	1	[-]
Logical variable indicating MSA stop cycle active (from stop request until start end)					
LV_TEMP_BOL	O	0...1H	0...1	1	[-]
Activation of homogeneous request due to coolant temperature reasons					
LV_TQI_BOL_SET_S	O	0...1H	0...1	1	[-]
Logical variable is set if TQI_SP_S falls below the minimim allowable torque in stratified					
LV_VAR_SAP	O	0...1H	0...1	1	[-]
Vehicle with secundary air pump					
LV_VAR_SAV	O	0...1H	0...1	1	[-]
vehicle with secundary air valve (dsk_byte_0 bit 5)					
LV_VAR_STST	O	0...1H	0...1	1	[-]
Variant for engine start stop automatic (MSA)					
MAF_HB	O/V	0...FFH	0...1389	5.4470588	[mg/stk]

set to 0 at transition ERU -> ES					
MAF_SP_S	O	0...FFFFH	0...1389	0.0211948	[mg/stk]
MAF setpoint for stratified mode					
Mrnn_test_pr	O	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Testerwert NN PR-Adaption					
PWM_VALUE_HIV_PERC	O	0...FH	0...150	10	[%]
control signal for heater inlet valve					
TEG_TUR_UP_MES[NC_NR_TEG_SENS]	O	0...FFE0H	-273.15...1773.85	0.03125	[°C]
measured exhaust gas temperature upstream of turbo charger					
TQ_ADD_ST	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve at engine start					
TQI_SP_SLOW	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Slow indicated engine torque					
VLS_NT_DOWN[NC_NOX_SENS_CONF]	O/V	0...3FFH	0...4.99511	4.8828e-3	[V]
voltage of lambda sensor downstream NOx-catalyst					
LV_ACT_SA_EOL	O	0...1H	0...1	1	[-]
Activation bit for secondary air EOL test					
STATE_DIAG_SA_SAFM	O			1	[-]
States of secondary air system diagnosis					
FAC_LAM_DIAGCP[NC_CBK_EX_NR]	O	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
lambda controller mean value at the beginning of CPS opening					
AC_VEH	V/O	8000...7FFFH	-27.77088... 27.77003	0.8475e-3	[m/s²]
Momentane Fahrzeugbeschleunigung					
N_VS	V/O	0...FFH	0...255	1	[rpm/(km/h)]
Division n/vs					
VS_HIGH_RES	V/O	0...7FFFH	0...327.67	0.01	[km/h]
Fahrzeuggeschwindigkeit (Wert mit hoher Auflösung)					
TQI_BAS_MAX	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Maximum available torque at basic conditions EFF_TOT_BAS					

LV_ERR_CLK_DRIFT_PBK_IV[NC_CYL_NR]	O	0...1H	0...1	1	[-]
Error flag for clock drift failure in ATIC88 or too high input voltage in injector					
LV_AMT_CRAWL_ON	O	0...1H	0...1	1	[-]
Active AMT clutch torque					
STATE_TQ_AMT_PLAUS	O	0...FFH	0...255	1	[-]
Bitwise coded State for AMTintervention state					
STATE_ERR_AMT_CAN	O	0...5H	0...5	1	[-]
State of AMT Error intervention CAN 11H					
TQI_AMT_SLOW_INC	O	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque increase for slow torque intervention during gear shift					
LV_AMT_INC_ACT	O	0...1H	0...1	1	[-]
Logical variable for torque intervention due to gear shift					
TQI_AMT_FAST_INC	O	0...7FFFH	0...1023.96875	0.03125	[Nm]
Torque increase for fast torque intervention during gear shift					
TQI_AMT_FAST_DEC	O	0...7FFFH	0...1023.96875	0.03125	[Nm]
Torque decrement for fast torque intervention during gear shift					
LV_AMT_DEC_ACT	O	0...1H	0...1	1	[-]
Logical variable for torque intervention due to gear shift					
LV_INH_FCUT_AMT	O	0...1H	0...1	1	[-]
Logical variable inhibiting FCUT during AMT-intervention					
TQI_AMT_SLOW_DEC	O	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque decrement for slow torque intervention during gear shift					
LV_STST_STOP_CYC_MEM	O	0...1H	0...1	1	[-]
Memorized flag LV_STST_STOP_CYC for the time after end of start until engine stop					

Input data:

CYC_CAST	LAMB_LS_UP[NC_CBK_EX_NR]	LAMB_SP	LV_EFP_EXT_ADJ
LV_ES	MAF_SP	MAF_SP	NC_CBK_EX_NR
NC_NOX_SENS_CONF			
TQI_MAX	Prnn_test	STATE_CH	TEG_CAT_DOWN_MDL[NC_CBK_EX_NR]
TEG_CAT_UP_MDL[NC_CBK_EX_NR]	TEG_PCAT_DOWN_i	TQI_REQ_SLOW	TQI_REF_MAX
AC_VEH_LGT_TCS	N_VS_RATIO	VS_H	EFF_TOT_BAS

Application conditions:

Configuration values: NC_CMB_CONF = AFS_AFL_S (4H)

NC_CBK_HPP_NR = 1

Initialisation at reset: *0,*

except:

IGA_TRA_KNK = 0 °crk

PWM_VALUE_HIV_PERC = 100%

TQI_AMT_FAST_DEC = 1024 Nm

TQI_AMT_SLOW_DEC = 1024 Nm

Recurrence: same recurrence as corresponding input data

Formula section:

CYC_ST = CYC_CAST

LAMB_AV_COR = 0.5 * (LAMB_LS_UP[1] + LAMB_LS_UP[2])

MAF_HB = MAF

MAF_SP_S = MAF_SP

TQI_SP_SLOW = TQI_REQ_SLOW

LV_EFPPWM_EXT_ADJ = LV_EFP_EXT_ADJ
Mrnn_test_pr = Prnn_test

AC_VEH = AC_VEH_LGT_TCS
N_VS = N_VS_RATIO
VS_HIGH_RES = VS_H
TQI_BAS_MAX = MIN (TQI_MAX, TQI_REF_MAX * EFF_TOT_BAS)

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CMB_CONF	1	0H 1H 2H 3H	AFS AFS_S AFS_AFL AFS_AFL_S	1	[-]
Engine combustion modes target					
NC_STATE_STST_ENA	O	0...1H	0...1	1	[-]
Switch to indicate engine stop start automatic enabled (= 1) or disabled (= 0)					
NC_CBK_HPP_NR	1	0...FFH	0...255	1	[-]
System constant for number of used high pressure pumps (1 = one pump, 2 = two pumps, ...; SW: 0 = one pump; 1 = 2 pumps, ...) - Anzahl Hochdruckpumpen im System					

36.8.2 Variables definition for missing Secondary air function

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_LAMB_SP_SA	O	0...FFFFH	0...3.99993	0.061e-3	[-]
Lambda deviation through secondary air					
LV_ERR_CAN_STST	O	0...1H	0...1	1	[-]
CAN error of MSA relevant signals					
LV_ERR_SA_SAP	O	0...1H	0...1	1	[-]
Present failure: Minimum secondary air flow rate failure after filtering					
LV_ERR_SA_SAV	O	0...1H	0...1	1	[-]
Present failure: Mechanically jammed SAV after filtering					
LV_ERR_SA_SYS	O	0...1H	0...1	1	[-]
Present failure: Minimum secondary air flow rate failure after filtering					
LV_ERR_SAP	O	0...1H	0...1	1	[-]
Boolean for error currently present on secondary air pump command signal.					
LV_ERR_SAV	O	0...1H	0...1	1	[-]
Boolean for error currently present on secondary air valve command signal					
LV_PUC_SA_INH	O	0...1H	0...1	1	[-]
Inhibition of PUC during SA active for the first time					
LV_SA_END	O	0...1H	0...1	1	[-]
secondary air function ended for this cycle (lambda adaptation obstruct)					
LV_SAP	O	0...1H	0...1	1	[-]
Flag for secondary air pump activation					
LV_SAV	O	0...1H	0...1	1	[-]
Flag for secondary air valve activation					
LV_SAWUP	O	0...1H	0...1	1	[-]
secondary air injection time correction active					
LV_TEMP_MAX_SAP_COIL	O	0...1H	0...1	1	[-]
Inhibition bit for SA-function					
SAF_DIAG_MAX	O	0...FF00H	0...255	3.9063e-3	[kg/h]
Mean diagnosis value of maximum air flow					
SAF_DIAG_MAX_SAE	O	0...FF00H	0...255	3.9063e-3	[kg/h]
Mean diagnosis value of maximum secondary air flow for SAE					
SAF_DIAG_MIN	O	0...FF00H	0...255	3.9063e-3	[kg/h]
Mean diagnosis value of minimum air flow					
SAF_DIAG_MIN_SAE	O	0...FF00H	0...255	3.9063e-3	[kg/h]
Mean diagnosis value of minimum secondary air flow for SAE					
SAF_KGH	O	0...FFFFH	0...1023.98437	0.015625	[kg/h]
Corrected value of SA mass flow					
STATE_OBD_SA	O	0...FFH	0...255	1	[-]
Commanded SA-Status for Scan-Tool					
STATE_SA	O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH BH CH	SA_INACTIVE SA_DELAY SA_ACTIVE INTERRUPT SAP_DELAY FINISHED CANCELLED EXT_ADJ EXT_ADJ_FIN EOL_SA_ACT EOL_SA_SAP_ ACT EOL_SA_FIN SAV_CLN	1	[-]
State of Secondary Air Function					
V_SAF	O	0...FFH	0...4.998	0.0195313	[V]
Secondary air mass flow sensor raw acquisition					

Application conditions:

Initialisation at reset: 0, except:

FAC_LAMB_SP_SA = 1.00
LV_SA_END = 1
STATE_OBD_SA =00000100b
(SAP deactivated, Atmosphere / off)

36.8.3 Error Flags

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_CAM_DE_IVVT_EX[NC_NR_CBK_IVVT]	O	0...1H	0...1	1	[-]
Present failure: failure after filtering of diagnosis enduring exhaust camshaft deviation					
LV_ERR_CAM_DE_IVVT_IN[NC_NR_CBK_IVVT]	O	0...1H	0...1	1	[-]
Present failure: failure after filtering of diagnosis crankshaft to inlet camshaft mechanics violation					
LV_ERR_CRASH_SIG	O	0...1H	0...1	1	[-]
Boolean for error currently present on location "Crash Signal front impact"					
LV_ERR_CRIT_OVL_ECU_VVL	O	0...1H	0...1	1	[-]
Boolean for error currently present on driver overload					
LV_ERR_CUR_H_VVL	O	0...1H	0...1	1	[-]
Error motor current critical high					
LV_ERR_DR_SC_VVL	O	0...1H	0...1	1	[-]
Short circuit error detected of VVL driver					
LV_ERR_EGR	O	0...1H	0...1	1	[-]
Error currently present on exhaust gas recirculation system					
LV_ERR_FPA	O	0...1H	0...1	1	[-]
error bit indicating fuel pressure actuator					
LV_ERR_FPS	O	0...1H	0...1	1	[-]
Error currently present on fuel pressure sensor					
LV_ERR_OVL_ECU_VVL	O	0...1H	0...1	1	[-]
Boolean for error currently present on OVL_ECU_VVL					
LV_ERR_PORT[NC_PORTPWM]	O	0...1H	0...1	1	[-]
Flag for Port-flap power stage electrical failure currently prensent					
LV_ERR_RLY_VVL	O	0...1H	0...1	1	[-]
Boolean for error currently present on relay controll / capacitor pre-charge					
LV_ERR_SA_SAV_LSL	O	0...1H	0...1	1	[-]
Present failure: Mechanical jammed SAV after filtering					
LV_ERR_TEG_PCAT_DOWN[NC_CBK_EX_NR]	O	0...1H	0...1	1	[-]
Temp. after precatalyst error flag					
LV_ERR_VB	O	0...1H	0...1	1	[-]
Present failure after filtering of diagnosis of battery voltage					
LV_ERR_VVL_ROT	O	0...1H	0...1	1	[-]
Error flag for rotation sense diagnosis					

Input data:

LV_ERR_ACR_AD	LV_ERR_ACR_CTL	LV_ERR_ACR_DR	LV_ERR_CAM_TOT
LV_ERR_SENS_ACR	NC_NR_CBK_IVVT		

Hinweis: Die hier definierten Bits LV_ERR_XXX unterliegen nicht der alg. Fehlerspeicherverwaltung.

Application conditions:

Initialisation at reset: 0

Activation: at every engine state

Recurrence: same recurrence as corresponding input data

Formula section:

LV_ERR_TEG_PCAT_DOWN[NC_CBK_EX_NR] = LV_ERR_TEG_PCAT_DOWN
LV_ERR_EGR = LV_ERR_ACR_AD **or** LV_ERR_ACR_CTL **or** LV_ERR_ACR_DR **or** LV_ERR_SENS_ACR

36.8.4 Interfaces to chapter serial communication

In the chapter I 'Serial Communication' are a lot of variables used which are not defined yet. Since there is no update of this chapter all unsatified inputs are defined here.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_mareghub_ad	O	0...1H	0...1	1	[-]
Flag indicating "Adaption Massenstromregler auf Hub erstmalig erfolgt"					
St_vbrvs_aus	O	0...FFFFH	0...65535	1	[-]
Status Verbrennungsregelung					
St_vbrvs_ein	O	0...FFH	0...255	1	[-]
Status Verbrennungsregelung					
LV_CRC_DONE_BOOT	-	0...1H	0...1	1	[-]
-					
LV_CRC_DONE_CAL	-	0...1H	0...1	1	[-]
-					
LV_CRC_DONE_ECU	-	0...1H	0...1	1	[-]
-					
LV_CRC_ERR_BOOT	-	0...1H	0...1	1	[-]
-					
LV_CRC_ERR_CAL	-	0...1H	0...1	1	[-]
-					
LV_CRC_ERR_ECU	-	0...1H	0...1	1	[-]
-					
LV_CRC_STATUS_MU	-	0...1H	0...1	1	[-]
Flag indicating an error in the Standard ROM Check					

Application conditions:

Initialisation:	0FFFFh:	St_vbrvs_aus
	0FFh:	St_vbrvs_ein
	0:	other

36.8.5 Outputs due to aggregate implementation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EOI_LIM[NC_CYL_NR]	O	F100...F00H	-1440...1440	0.375	[°CRK]
Cylinder individual zero position for phasing					
INH_CYC_IV	O	0...FFH	0...255	1	[-]
Shut-off request for cylinder bitmask					
LV_IND_FCUT	O	0...1H	0...1	1	[-]
At least one cylinder is shut off					
LV_REQ_SEG_AD_RST_EOL	O	0...1H	0...1	1	[-]
Variable to reset SAG_AD variables due to a EOL request					
SOI_MAX	O	0...780H	0...720	0.375	[°CRK]
Earliest possible SOI					

Input data:

CRK_INJ_BAS[NC_CYL_NR]	INH_IV	LV_CLC_2SEG	LV_CLC_2SEG_ENA
LV_FCUT_IND	SOI_LIM		

FUNCTION DESCRIPTION: ---

Application conditions:

Initialisation: 0

Recurrence : every segment

Activation: every state if LV_CLC_2SEG = 1 **or** LV_CLC_2SEG_ENA = 0

Configuration data: NC_IN_REF = 42 *// one intake bank*

NC_LAMB_REF = 42 *//(101010B)*

Formula section:

EOI_LIM[NC_CYL_NR] = CRK_INJ_BAS[NC_CYL_NR]

INH_CYC_IV = INH_IV

LV_IND_FCUT = LV_FCUT_IND

SOI_MAX = SOI_LIM

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_IN_REF	1	0...FFH	0...255	1	[-]
Pattern for allocation of physical cylinders to intake bank					
NC_LAMB_REF	1	0...FFH	0...255	1	[-]
Pattern for allocation of physical cylinders to exhaust bank					

36.8.6 Adaptation module for non implemented VVL functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ANG_EXC_VVL	O	0...238EH	0...200	0.0219732	[°]
Selected current value in grad after jitter filtration					
MAP_DIP_SP_MMV	O	8000...7FFFH	-1280... 1279.96093	0.0390625	[hPa]
Filtered setpoint of manifold differential pressure					
Mrnn_test_vvt	O	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
Testerwert NN VVT-Adaption					
VLFT_MIN	O/S	0...FFFFH	0...65.535	0.001	[mm]
Result of adaptation of minimal variable valve lift					

FUNCTION DESCRIPTION: ---

Application conditions:

Initialisation: 0, except: ANG_EXC_VVL = 200 °

36.8.7 Adaptation module for non implemented STST (MSA) functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DRIV_BELT_CLOSE	O	0...1H	0...1	1	[-]
Driver seat belt fastened					
LV_STST_DEAC_ACT	O	0...1H	0...1	1	[-]
MSA deactivation due to permanent errors					
LV_STST_INH_ACT	O	0...1H	0...1	1	[-]
MSA inhibition due to reversible errors					
LV_STST_INH_CDN_AD	O	0...1H	0...1	1	[-]
MSA inhibition due to adaption					
LV_STST_SWI_ACT	O	0...1H	0...1	1	[-]
Engine start stop automatic switched on					
PBSU	O	0...FFFFH	0...5434	0.0829175	[hPa]
Absolute pressure in the brake servo					
PSN_ENG_REL	O	0...780H	0...720	0.375	[°CRK]
Relative engine position before next TDC					
PSN_ENG_REL_ST	O	0...780H	0...720	0.375	[°CRK]
Relative engine position before next TDC at re-start					
STATE_STST_REQ_CAN	O	0H 1H 2H 3H 4H 5H 6H 7H	NO_REQ START_REQ STOP_REQ START_INH STOP_INH NOT_USED NOT_USED INVALID_ SIGNAL	1	[-]
State indicating request from CAN for STST (MSA) functionality					
LV_STST_ES	O	0...1H	0...1	1	[-]
Logical variable indicating engine stop caused by start stop automatic					
LV_ENG_RUN_CMB	O	0...1H	0...1	1	[-]
Running engine without starter, only through combustion					
INH_IV_STST	O	0...FFH	0...255	1	[-]
Cylinder shut off pattern for engine shut down due to stop request					
LV_RLY_ST_STST	-	0...1H	0...1	1	[-]
HW-control of the starter relay via MSA					
LV_STST_ST_REQ	O	0...1H	0...1	1	[-]
Logical variable indicating the validated start request from start stop manager					
LV_STST_STOP_REQ	O	0...1H	0...1	1	[-]
Logical variable indicating the validated stop request from start stop manager					
LV_IGK_OFF_ACK_PERM_DEAC	O	0...1H	0...1	1	[-]
Bit indicating permanent deactivation of ignition off acknowledgement					
LV_IGK_OFF_ACK_TMP_DEAC	O	0...1H	0...1	1	[-]
Bit indicating temporay deactivation of ignition off acknowledgement					

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: 0

36.8.8 Adaptation module for non implemented TÖNS functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_EDGE_FALL_LOIL	O				
CTR_SIG_NOT_VLD_LOIL	O				
LV_SOIL_VLD_LOIL	O	0...1H	0...1	1	[-]
Oil temperature valid					
LV_STATE_RUN_LOIL	O	0...1H	0...1	1	[-]
State machine once passed					
LV_T_COOL_VLD_LOIL	O	0...1H	0...1	1	[-]
Cool off time valid					
T_COOL_LOIL	O	0...FFFFH	0...6553.5	0.1	[-]
Cool off time of LOIL sensor					
T_EDGE_FALL_LOIL	O				
T_REF_SIG_LEN_LOIL	O				
T_TEMP_SIG_LEN_LOIL	O				

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: 0

36.8.9 For non implemented serielle communication

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Exwinkkor	O	DC72...238EH	-200...200	0.0219732	[°]
Korrekturwinkel Excenterwelle zur Hubkorrektur					
F_minhub	O	0...FFFFH	0...0.99998	0.0153e-3	[-]
Faktor Ein-/Ausblendung Minhub über Tmot und Nkw					
F_tikorrvr[NC_CYL_NR]	O	0...FFFFH	0...1.99996	0.0305e-3	[-]
Korrekturfaktor für Einspritzung aus Verbrennungsregelung					
Minhub	O	0...FFFFH	0...65.535	0.001	[mm]
Ergebnis der Minhub-Adaption					
Minhub_roh	O	0...FFFFH	0...65.535	0.001	[mm]
Minhubrohwert aus Adaption					
Minhubvs	O	0...FFFFH	0...65.535	0.001	[mm]
Vorgabe Minhub über Tester					
Minhubvs_ist	O	0...FFFFH	0...65.535	0.001	[mm]
Tatsächlich wirksamer Minhub aus Verstelleingriff (vor Ein-/Ausblendungsfaktoren)					
Minhubvsnv	O/S	0...FFFFH	0...65.535	0.001	[mm]
dauerhaft fest programmierter Minhub					
N_WHEEL_FN_LE	O	8000...7FFFH	-512...511.703	0.0156207	[1/s]
Actual value for RPM wheel front left					
N_WHEEL_FN_RI	O	8000...7FFFH	-512...511.703	0.0156207	[1/s]
Actual value for RPM wheel front right					
N_WHEEL_RE_LE	O	8000...7FFFH	-512...511.703	0.0156207	[1/s]
Actual value for RPM wheel rear left					
N_WHEEL_RE_RI	O	8000...7FFFH	-512...511.703	0.0156207	[1/s]
Actual value for RPM wheel rear right					
S_vsmnhb	O	0...FFH	0...255	1	[-]
Schalter für Testereingriff					
S_vsmnhbnv	O/S	0...FFH	0...255	1	[-]
Schalter für Testereingriff					
STATE_DIAG_DR_VVL	O	0...FFH	0...255	1	[-]
State of ATIC61 self- diagnosis					
STATE_TBL_DRIV[8][6]	O	0...FFH	0...99.60937	0.390625	[%]
Table of driving profile over torque request and engine speed (saved in NVMY)					
STATE_TQ_WHEEL_DRIV_ASI	O	0...FFH	0...255	1	[-]
qualifier / state of signal TQ_WHEEL_DRIV_ASI					
STATE_TQ_WHEEL_TCS_SLOW	O	0...FFH	0...255	1	[-]
Qualifier concept to target wheel torque on powertrain					
TQ_WHEEL_TCS_FAST	O	8000...7FFFH	-32768...32767	1	[Nm]
Target wheel torque on powertrain for fast stabilization					
TQ_WHEEL_TCS_SLOW	O	8000...7FFFH	-32768...32767	1	[Nm]
Target wheel torque on powertrain for slow stabilization					
Zylhubkor	O	0...FFH	0...255	1	[-]
Für Hubkorrektur ausgewählter Zylinder					

FUNCTION DESCRIPTION: only for satisfaction of ISCAN-tools

Application conditions:

Initialisation:	0FFFFh:	Exwinkkor, F_minhub, F_tikorrvr[0..7], Minhub, Minhub_roh, Minhubvs, Minhubvs_ist, Minhubvsnv
	0FFh:	S_vsmnhb, S_vsmnhbnv, Zylhubkor, STATE_TQ_WHEEL_TCS_SLOW, STATE_TQ_WHEEL_DRIV_ASI
	-32000 Nm:	TQ_WHEEL_TCS_SLOW, TQ_WHEEL_TCS_FAST
	0:	other

Formula section: ---

36.8.10 Variable definitions for missing CAN11hex functionality

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQI_GS_FAST_REQ_CAN	O	0...7FFFH	0...1023.96875	0.03125	[Nm]
relative torque request during gear shift intervention - fast					
TQI_GS_SLOW_REQ_CAN	O	0...7FFFH	0...1023.96875	0.03125	[Nm]
relative torque request during gear shift intervention - slow					
TQI_GS_FAST_REQ	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
absolute torque request during gear shift intervention - fast					
TQI_GS_SLOW_REQ	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
absolute torque request during gear shift intervention - slow					
LV_CTOP	O	0...1H	0...1	1	[-]
State convertible top: 0 = closed, 1 = open					
TQI_ASR_FAST_CAN	O	0...FFH	0...99.60937	0.390625	[%]
Relative indicated engine torque requested by ASR (fast path)					
TQI_MSR_CAN	O	0...FFH	0...99.60937	0.390625	[%]
Relative indicated engine torque requested by MSR					
TQI_ASR_FAST_REQ_CAN	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Relative indicated engine torque requested by ASR (fast path)					
TQI_MSR_REQ_CAN	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Relative indicated engine torque requested by MSR					
TQI_ASR_SLOW_REQ_CAN	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Relative indicated engine torque requested by ASR (slow path)					
LV_ASR_PLAUS	O	0...1H	0...1	1	[-]
Logical variable plausibility ASR					
LV_MSR_PLAUS	O	0...1H	0...1	1	[-]
Logical variable plausibility MSR					
LV_CITY	O	0...1H	0...1	1	[-]
City mode SSG					
TQI_AMT_REQ_CAN	O	0...FFH	0...99.60937	0.390625	[%]
Torque request SSG via CAN					
TQI_AMT_REQ	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque request SSG					
LV_ERR_TOUT_ASR_3	O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message ASC3					
LV_ERR_TOUT_ETCU_1	O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message EGS1					
LV_ERR_TOUT_ETCU_2	O	0...1H	0...1	1	[-]
logical variable timeout error CAN message EGS2					
LV_ERR_TOUT_AMT_1	O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message SSG1					
LV_ERR_TOUT_ICL_2	O	0...1H	0...1	1	[-]
logical variable timeout error CAN message INSTR2					
LV_ERR_TOUT_ICL_3	O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message INSTR3					
LV_ERR_TOUT_PSTE_1	O	0...1H	0...1	1	[-]
logical variable timeout error CAN message LWS1					
LV_ERR_TOUT_ASR_1	O	0...1H	0...1	1	[-]
logical variable timeout error CAN message ASC1					
TQI_SLOW_AMT_REQ	O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque request (slow) SSG					
LV_GP	O	0...1H	0...1	1	[-]

LV indicating mode gearbox - AMT					
LV_ERR_TOUT_ICL_7	O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message INSTR7					
LV_ERR_TOUT_EFP_CAN	O	0...1H	0...1	1	[-]
logical variable timeout error CAN message EKP					

Application conditions:

Initialisation: 0

Formula section: ---

36.8.11 Variables only necessary for IScan

Output data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IGA_CH	O	0...80H	-48...0	0.375	[°CRK]
dummy					
LV_ALTER_RD_TBL_6	O	0...1H	0...1	1	[-]
logical variable: software switch for condition read register 6					
LV_BRAKE_REQ	O	0...1H	0...1	1	[-]
hom. mod is requested by critical break under pressure					
LV_CAM_SP_CH	O	0...1H	0...1	1	[-]
--					
LV_CH_TQ_ADD	O	0...1H	0...1	1	[-]
Catalyst Heating Function "torque reserve" (late ignition angle)					
LV_CH_TQ_ADD_IS	O	0...1H	0...1	1	[-]
Catalyst Heating Function "torque reserve for first idle phases"					
LV_INH_DPS_REG_CPSDIAG	O	0...1H	0...1	1	[-]

LV_LAMB_CH	O	0...1H	0...1	1	[-]
indicates that lambda catalyst heating is active					
LV_MFF_S_POST_CH	O	0...1H	0...1	1	[-]

MAF_SUB_COR_MMV_MON	O	0...FFH	0...1389	5.4470588	[mg/stk]
Low pass filtered MAF_SUB_COR_MON					
PRS_EX_PCAT_UP	O	0...FFFFH	0...5434	0.0829175	[hPa]
Exhaust pressure pre catalyst					
STATE_CUT_OFF_DT	O	0...FFH	0...255	1	[-]
state cut off drivetrain					
STATE_INI_DT	O	0...FFH	0...255	1	[-]
state initialisation drivetrain					
STATE_N_MAX_MON	O	0...FFFFH	0...65535	1	[-]
Error bit indication					
LV_END_DIAG_SA_SAFM	O	0...1H	0...1	1	[-]
End of SAFM diagnosis					
LV_END_DIAG_SA_SAP	O	0...1H	0...1	1	[-]
End of SA minimum flow rate - diagnosis					
LV_END_DIAG_SA_SAV	O	0...1H	0...1	1	[-]
End of Mechanically jammed SAV-diagnosis					
LV_END_DIAG_SA_SYS	O	0...1H	0...1	1	[-]
End of SA minimum flow rate diagnosis					
SAF_KGH_MES_BAS	O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Raw secondary air-mass flow measured					
LV_ERR_MEC_IVVT_EX[NC_NR_CBK_IVVT]	O	0...1H	0...1	1	[-]
Error Flag for functional error on the exhaust CAM adjustment					
LV_ERR_SA_SAFM	O	0...1H	0...1	1	[-]
Present failure: Component Secondary air flow meter failure					
St_dvovrld	O	0...FFH	0...255	1	[-]
carrierbyte for status of VVLI overload (overload diagnosis ECU / DC motor)					
St_vvt_err	O	0...FFH	0...255	1	[-]
carrierbyte for status of VVL failures (indicating the ability of the position control)					
ANG_1_RAW_VVL	O	0...1FFFH	0...179.97802	0.0219727	[°]
Recent as valid selected raw value in actual sampling time of sensor1					
ANG_DE_ABSV_PLAUS_CHK_VVL	O	0...1FFFH	0...179.97802	0.0219727	[°]
Absolute value of angle deviation between sensor 1 and 2 for plausibility check					
ANG_REL_VVL	O	0...1FFFH	0...99.98779	0.012207	[%]
Selected Actulal Value of VVL angle position for position controller					

ANG_SP_CTL_VVL	O	0...1FFFH	0...99.98779	0.012207	[%]
Setpoint for the position controller					
ANG_SP_VVL	O	0...238EH	0...200	0.0219732	[°]
Desired position of the excenter shaft in degree					
ANG_SP_VVL_CUS	O	0...238EH	0...200	0.0219732	[°]
Desired position of the excenter shaft in degree for customer position controller					
ANG_STK_VVL	O	0...238EH	0...200	0.0219732	[°]
Effective stroke (linearised value)					
CUR_MOT_VVL[NC_SENS]	O	F380...C80H	-200...200	0.0625	[A]
Actual signed motor current. The current is positive when the source - drain current of the SWI_4 Transistor is positive (k=0...4)					
LV_STATE_RLY_VVL	O	0...1H	0...1	1	[-]
Pre-charge relay state, 0 = relay open, 1 = relay closed					
PWM_DR_OUT_SET_VVL	O	0...1FFH	0...99.80468	0.1953125	[%]
PWM to be set by PWM generator see VVL chopper control too					
TEMP_MOT_VVL	O	E800...7FFFH	-48...255.99218	0.0078125	[°C]
Estimated DC motor temperature					
TEMP_SWI_MES_MAX_VVL	O	8000...7FFFH	-1024... 1023.96875	0.03125	[°C]
Maximal measured temperature of the chopper switch transistors.					
V_VCC_SENS_VVL	O	0...FFFFH	0...511.99218	0.0078125	[V]
Sensor supply voltage					
V_VCC_SENS_VVL_RAW	O	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage supply for VVL sensor					
VCC_DR_VVL	O	0...FFFFH	0...511.99218	0.0078125	[V]
Supply voltage filtered with fast filter					
B_msadxcav	O/V	0...1H	0...1	1	[-]
Bedingung MSA-Abscaltverhinderer vom DxC					
CAT_DIAG[NC_CBK_EX_NR]	O/S	0...7FH	0...1.98437	0.015625	[-]
At ended diagnosis final value for catalyst conversion capability					
EAC_COD_SENS	O	0...FFH	0...255	1	[-]
result of decoding process of the EAC sensor					
Exwink_ist	O	0...238EH	0...200	0.0219732	[°]
Actual position of the excenter shaft in degree					
Ivvtmot[NC_CYL_NR]	O	8000...7FFFH	-2048... 2047.9375	0.0625	[A]
Array of VVLI DC motor current					
LOAD_MAX_SCDN_MIS	O/S	0...FFH	0...99.60937	0.390625	[%]
Maximal load of similar condition window for LV_ERR_MIS					
LOAD_MIN_SCDN_MIS	O/S	0...FFH	0...99.60937	0.390625	[%]
Minimal load of similar condition window for LV_ERR_MIS					
LV_ACT_ECRAS_EXT_ADJ	O	0...1H	0...1	1	[-]
adjustment value for radiator shutters 0 = off, 1 = on					
LV_VAR_EAC	O/S	0...1H	0...1	1	[-]
variant EAC					
LV_VAR_ECRAS	O/S	0...1H	0...1	1	[-]
variant of ECRAS recognized					
MFF_AD_ADD_MMV_REL[NC_CBK_EX_NR]	O	8000...7FFFH	-694.51059... 694.48940	0.0211948	[mg/stk]
Relative additive adaptive factor					
MFF_AD_FAC_MMV_REL[NC_CBK_EX_NR]	O	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Relative multiplicative adaptive factor					
MFF_ADD_EXT_ADJ[NC_CBK_EX_NR]	O	8000...7FFFH	-694.5... 694.47880	0.0211945	[mg/stk]
Mass fuel flow for CO adjustment at idling, value for injection correction					
MFF_ADD_EXT_ADJ_NVMY	O/S	8000...7FFFH	-694.5... 694.47880	0.0211945	[mg/stk]
Mass fuel flow for CO adjustment at idling, value for storing in NVMY					
Oz_manip	O	0...FFH	0...255	1	[-]
Manipulationsbyte					

VS_H_N_MAX	O/S	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed at highest event					
Vvt_soll	O	0...238EH	0...200	0.0219732	[°]
Desired position of the excenter shaft in degree for customer position controller					
LV_KWP_PROG_DATA	O	0...1H	0...1	1	[-]
State variable for requesting for programming the calibration data after power latch phase					
STATE_RST_TYP_ACT	-	0...FFH	0...255	1	[%]
Status-byte: reset type during current ECU-run					
CTR_WRST	-	0...FFH	0...255	1	[-]
Number of atypical warm-resets since last power-up (from BSW) / RESET SAFE					
STATE_RST_INFO_ADD[NC_NR_DBG_NVMY]	S	0...FFFFFFFFH	0...4294967295	1	[-]
Additional reset info (cause) / RESET SAFE					
LV_DBG_INFO_VLD[NC_NR_DBG_NVMY]	S	0...1H	0...1	1	[-]
background info for last reset valid / RESET SAFE					
STACK_ADR_RST[NC_NR_DBG_NVMY][NC_NR_STACK_ADR_RST]	S	0...FFFFFFFFH	0...4294967295	1	[-]
Info last available caller addresses (default 0) / RESET SAFE					
STATE_RST_TYP[NC_NR_DBG_NVMY]	S	0...FFH	0...255	1	[%]
Reset typ / RESET SAFE					

FUNCTION DESCRIPTION: only for satisfaction of ISCAN-tools

Application conditions:

Initialisation at reset: 0,

except:

IGA_CH = 0 °crk

LV_END_DIAG_SA_SAFM = 1

LV_END_DIAG_SA_SAP = 1

LV_END_DIAG_SA_SAV = 1

LV_END_DIAG_SA_SYS = 1

Formula section: ---

PRS_EX_PCAT_UP = PRS_EX

If STATE_CH = 1h

Then LV_LAMB_CH = 1

Else LV_LAMB_CH = 0

Endif

TEG_PCAT_DOWN[NC_CBK_EX_NR] = TEG_PCAT_DOWN_i

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CAM_OP_EX	1	0...FFH	175.125...270.75	0.375	[°CRK]
Opening period of exhaust valve					
LC_AD_CLR_CPS	1	0...1H	0...1	1	[-]
Logical constant for initializing CPS nvmy data					
LC_AD_CLR_IDLE	1	0...1H	0...1	1	[-]
Logical constant for initializing IDLE nvmy data					
LC_AD_CLR_IMM	1	0...1H	0...1	1	[-]
Logical constant for initializing IMM nvmy data					
LC_AD_CLR_SA	1	0...1H	0...1	1	[-]
Logical constant for initializing SA nvmy data					
LC_AD_CLR_VVL_STATE_NVMY	1	0...1H	0...1	1	[-]
Calibration data to reset adaptation status flags					

36.8.12 Variables required by Software

Output data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LV_CAT_DIAG_REQ_EOL	O	0...1H	0...1	1	[-]
Flag to activate short trip of catalyst efficiency diagnosis					
CTR_GB_NEUT_NOT_PLAUS_SUM	O/S	0...FFH	0...255	1	[-]
Sum counter for confirmed neutral gear plausibility errors					
CHA_CDN_BAT	O	0...FFH	0...127.5	0.5	[-]
LV_ECU_SLA	O	0...1H	0...1	1	[-]
ECU is in slave-mode 0=Master (or no Master/Slave system); 1=Slave					
VP_TCO_COPL[NC_NR_TCO_SENS]	O	0...7FFFH	0...4.99984	0.1526e-3	[V]
Sensor raw acquisition of all available coolant temperature sensors on master ECU					

Input data:

VP_TCO			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_BAS_COR_MIN	O	0...7FFFH	0...1.99993	0.061e-3	[-]
Minimal value for lambda setpoint					
C_SAF_DIAG_MIN	O	0...FF00H	0...255	3.9063e-3	[kg/h]
Error detection threshold for minimum deviation					
C_SAF_DIAG_MAX	O	0...FF00H	0...255	3.9063e-3	[kg/h]
Error detection threshold for maximum deviation					

Application conditions:

Initialisation: 0

Formula section: ---

VP_TCO_COPL[NC_NR_TCO_SENS] = VP_TCO[NC_NR_TCO_SENS]

36.8.13 Interfaces required for nonimplemented NOx-functionality

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_NS_SHIFT_DIAG_ACT_EXT_ADJ[NC_NOX_SENS_CONF]	O	0...1H	0...1	1	[-]
Activation of the diagnosis by service tool					
LV_END_DIAG_NS_RAW[NC_NOX_SENS_CONF]	O	0...1H	0...1	1	[-]
Diagnostic performed at last one time for NOx signal RAW adaptation					
LV_ERR_NS_RAW[NC_NOX_SENS_CONF]	O	0...1H	0...1	1	[-]
Present failure after filtering of the NOx sensor - NOx raw emission diagnosis					

Export Actions:

void ACTION_NOXD_StartNNShiftDiag(void)
dummy for StartNNShiftDiag without any functionality
voidACTION_NOXD_EndNNShiftDiag(void)
dummy for EndNNShiftDiag without any functionality

36.9 Logic outputs

36.9.1 General management of logic outputs

FUNCTION DESCRIPTION:

General information:

Logic outputs are outputs which are piloted by software without PWM or CAPCOM.

Initialization:

At initialization phase, each logical output is set or reset versus function need.
See detail file of each output.

Logic state:

Positive logic means the connector output is "active" or "ON" (current present through the load) when the corresponding boolean is true.

The output is “ON” when the voltage level is low at the ECU connector and is “OFF” when the voltage level is high at the ECU connector.

Update recurrence:

All logical outputs are updated at application recurrence except those piloted by SPI octal link which are updated at 10 ms with jitter < 1ms.

Diagnostic:

Several options are proposed. For detail see chapter on output diagnostic.

36.10 PWM outputs

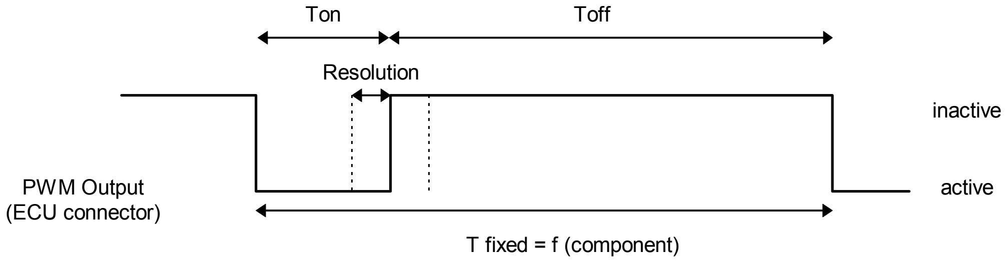
36.10.1 General management of PWM outputs :

FUNCTION DESCRIPTION:

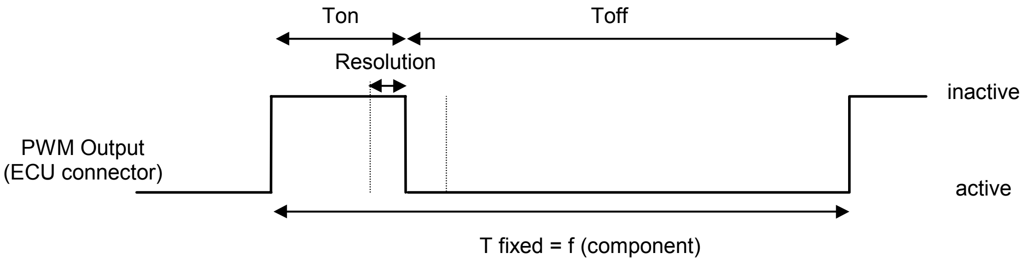
General information:

PWM management principle :

“Positive” PWM :



“Negative” PWM :



Duty cycle = Ton / T

PWM outputs are 2 types :

- True PWM using a POUT µP output. In this case the duty cycle range could be 0 . . 100%
- Pseudo PWM using a CAPCOM µP output. In this case, minimum Ton and minimum Toff are 300 µs with standard priority.

Initalization:

During hard reset and software initalization, all PWM outputs are inactive. As soon as application software start, the PWM outputs are piloted. Inactive output means inactive driver output.

36.11 PWM outputs

36.11.1 List of PWM outputs

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IVVTPWM_0	V/O	0...FFFFH	0...99.99847	1.53E-03	[%]
PWM for inlet VANOS					
IVVTPWM_1	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
PWM for outlet VANOS					

Input data:

MTCPWM	CPPWM_CPS	LSHPWM_UP[NC_CBK_EX_NR]	LSHPWM_DOWN[NC_CBK_EX_NR]
ECFPWM_ECF	ECTPWM	VIMPWM_1	VCVPWM
ECRASPWM	VIMPWM_2	WGPWM[NC_CBK_EX_NR]	PWM_ACR
FPAPWM		LV_VAR_TCHA	LC_POIL_CTL_ENA
POIL_PWM			

Data definition:

Name	ECU Pin Name (Output)	Definition / Remark	Frequency (Hz)
MTCPWM	A_T_MDK	Main throttle control	600
LSHPWM_UP_1	A_T_LHV1	LSH bank 1 upstream	See EGCP configuration
LSHPWM_UP_2	A_T_LHV2	LSH bank 2 upstream	
LSHPWM_DOWN_1	A_T_LHH1	LSH bank 1 downstream	
LSHPWM_DOWN_2	A_T_LHH2	LSH bank 2 downstream	
ECTPWM	A_S_KFK	electronically controlled thermostat	1
IVVTPWM_0	A_T_NWE1	VANOS unit 1 (inlet 1 or inlet)	244
IVVTPWM_1	A_T_NWE2	VANOS unit 2 (inlet 2 or outlet)	244
CPPWM_CPS	A_T_TEV	canister purge valve	Def. in EVAC
ECFPWM_ECF	A_T_ELUE1	cooling fan	100 (10 during power latch)
VCVPWM	A_T_MSV	Flow control valve	201
PWM_ACR	A_T_AGR1/2	Exhaust gas recirculation valve	976
ECRASPWM	A_T_LKS	Electronical controlled radiator shutter	200

Switchover between FPAPWM and POIL_PWM

If LC_POIL_CTL_ENA = 0

then

FPAPWM	A_T_DSV	Pressure control valve	1)

else

POIL_PWM	A_T_DSV	Electronically controlled oil pressure actuator	2)

endif

1.) According module ' High pressure pump control (Appl. Inc.)'

2.) According module 'Cus adap module: ENLU'

Additional for LV_VAR_TCHA = 0

Name	ECU Pin Name (Output)	Definition / Remark	Frequency (Hz)
VIMPWM_1	A_T_DISA1	variable intake manifold chanel 1	200
VIMPWM_2	A_T_DISA2	variable intake manifold chanel 2	200

Additional for LV_VAR_TCHA = 1

Name	ECU Pin Name (Output)	Definition / Remark	Frequency (Hz)
WGPWM[2]	A_T_WG2	wastegate chanel 2	300
WGPWM[1]	A_T_WG1	wastegate chanel 1	300

List of corresponding logic values:

Name	Definition
LV_MTC_DIR	direction of MTC
LV_MTC_DIS	disable MTC

36.11.2 Output PWM for inlet and outlet VANOS

The PWM outputs for both VANOS actuators are driven directly by BMW- objects via function call. To know the actual PWM- value in the system, the registers of the PWM-outputs are mirrored into system variables (IVVTPWM_0 and IVVTPWM_1). To reach a permanent micro-movement of the control piston in the valve a variable Frequency-Control is implemented driven directly by BMW- objects via function call: PWM_FRQ_NWA1(), PWM_FRQ_NWE1().

The minimum frequency, which can be adjusted, is 75 Hz

36.11.3 Output for HPDI relay

CH_PWM_DCDC = 80%

36.12 Interface definitions - Outputs from BMW functions to aggregate adaptation modules

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Agr_rate	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
EGR rate					
Amo_05	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Amplitude 0,5 Motorordnung					
Amo_10	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Amplitude 1,0 Motorordnung					
Amo_15	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Amplitude 1,5 Motorordnung					
Amo_20	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Amplitude 2,0 Motorordnung					
Amo_30	V/O	0...FFFFH	0...15.99975	0.2441e-3	[-]
Amplitude 3. Motorordnung					
Amo_40	V/O	0...FFFFH	0...15.99975	0.2441e-3	[-]
Amplitude 4. Motorordnung					
Aspr	V/O	0...FFH	0...204	0.8	[°CRK]
setpoint outlet camshaft					
Atlr	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Stellgröße Regler und Vorsteuerung					
Atlvst	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Stellgröße Anteil Vorsteuerung					
Atlsvc_dpvdk1	O/V	8000...7FFFH	-1280... 1279.96093	0.0390625	[hPa]
Differenzdruck vor DK beider Turbolader					
Atlsvc_dpvdk2	O/V	8000...7FFFH	-1280... 1279.96093	0.0390625	[hPa]
Differenzdruck vor DK mit Turbolader 1					
Atlsvc_dpvdk3	O/V	8000...7FFFH	-1280... 1279.96093	0.0390625	[hPa]
Differenzdruck vor DK mit Turbolader 1					
Ba_ist	V/O	0...FFH	0...255	1	[km/m]
Actual engine operation mode					
Ba_soll	V/O	0...FFH	0...255	1	[-]
requested engine operation mode					
Ba_subist	V/O	0...FFH	0...255	1	[-]
Actual engine operation mode					
Ba_subsoll	V/O	0...FFH	0...255	1	[-]
requested engine operation mode					
Baw_ist	V/O	0...FFFFH	0...65535	1	[-]
Betriebsart und Subbetriebsart IST					
Baw_prio	V/O	0...FFFFH	0...65535	1	[-]
priorisierte Betriebsart und Subbetriebsart					
Baw_soll	V/O	0...FFFFH	0...65535	1	[-]
Betriebsart und Subbetriebsart SOLL					
Bosbtvfbk	V/O/S	0...FFH	0...255	1	[%]
Verfügbarkeit Motoröl					
Bosconf	V/O/S	0...FFFFFFFFH	0...4294967295	1	[-]
Bos-Teilnehmer: Bit 1 = OZ; Bit 8 = PM					
Bosmanip	V/O	0...FFH	0...255	1	[-]
Manipulationsbyte					
Bosrlsm	V/O/S	8000...7FFFH	-327680... 327670	10	[km]
Restlaufstrecke Motoröl					
Bosstate	V/O	0...FFH	0...255	1	[-]
Einheit für Restlaufstrecke					

Bostoken	V/O	0...FFH	0...255	1	[-]
Teilen der BOS-Schnittstelle zwischen OZ und PM (0= frei, 1= besetzt vom OZ, 8= besetzt vom PM)					
Dfmonitor	V/O	0...FFH	0...100	0.3921569	[%]
condition of battery load					
Dffgen	V/O	0...FFH	0...100	0.3921569	[%]
calculated DF-signal from alternator					
Dla_soll_puls[NC_CBK_EX_NR]	O/V	8000...7FFFH	-8...7.99975	0.2441e-3	[-]
Lambdadifferenzen aus Pulsation					
Drf_spuel	V/O	8000...7FFFH	-327.68...327.67	0.01	[%]
Delta RF das in der Überschneidung durchgespuelt wird					
Dzw_agr_hs	O/V	FE0C...258H	-50...60	0.1	[°CRK]

Deltazündwinkel AGR-Korrektur (homogen-schicht)					
Dzw_agr_kor	O/V	FE0C...258H	-50...60	0.1	[°CRK]
ignition angle correction for external exhaust gas recirculation					
Dzw_annm	V/O	FE0C...258H	-50...60	0.1	[°CRK]
Referenzwert für B_kftkr, mittlerer adaptierter Spätziehwinkel					
Eco_jobstat1	V/O	0...FFH	0...255	1	[-]
Eco Job 1 Rueckgabewer					
Eco_result1	V/O	0...FFFFH	0...81.99874	1.2512e-3	[A]
ECOS Ruhestrom 1					
Eisyagr_korfak_b	V/O	0...FFH	0...1.99218	0.0078125	[-]
Abgleich AGR-Modell (Faktor)					
Eisydk_korfak_b	V/O	0...FFH	0...1.99218	0.0078125	[-]
Abgleich DK-Modell (Faktor)					
Eisydk_koroff_b	V/O	80...7FH	-1024...1016	8	[kg/h]
Abgleich DK-Modell (Offset)					
Eisyev_korfak_b	V/O	0...FFH	0...1.99218	0.0078125	[-]
Abgleich EV-Modell (Faktor)					
Eisyev_koroff_b	V/O	80...7FH	-1024...1016	8	[kg/h]
Abgleich EV-Modell (Offset)					
Espr	V/O	0...FFH	0...204	0.8	[°CRK]
setpoint inlet camshaft					
Espr_mod_soll[NC_CYL_NR]	O/V	0...FFH	0...255	1	[-]
Aktueller Einspritzmodus (Sollvorgabe)					
Eta_md_uesp	O/V	8000...7FFFH	-200...199.99389	6.1035e-3	[%]
Wirkungsgradkorrektur beim Überspülen					
Eta_mdkh	O/V	8000...7FFFH	-200...199.99389	6.1035e-3	[%]
Wirkungsgradkorrektur bei Katheizen					
Eta_zw_minres	V/O	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Minimal möglicher Zündwinkelwirkungsgrad für Momentenreserve bezogen auf min. Zündwinkel					
F_atlad	V/O/S	0...FFFFH	0...1.99996	0.0305e-3	[-]
Atl-Regler Adaptionsfaktor					
F_atldyn	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
Überspülfaktor					
F_bsmdres	O/V/S	0...FFH	0...0.99609	3.9063e-3	[-]
Faktor Bauteilschutz zur Begrenzung der Drehmomentreserve (bei Nox-Katheizen)					
F_llr_ba	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for operation mode transition					
F_nst_ba	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for operation mode transition for TI_CAST					
F_st_ba	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for operation mode transition for MFF_CST					
F_tikorzsk[NC_CYL_NR]	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
zylinderindividueller Einspritzkorrektur aus VBR					
F_wl_ba	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for operation mode transition for TI_WUP_1					
F_vl_aus	V/O	0...FFFFH	0...3.99993	0.061e-3	[-]
load factor for switching off full load					
F_vwi_ba	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Faktor zur Überblendung des Vorlagerungswinkels					
Genitest_tol	V/O	0...FFH	0...99.60937	0.390625	[%]
Toleranzbereich für Abweichung vom Sollwert Strom					
Geniutest_ab	V/O	0...FFH	0...255	1	[-]
GENIUTEST Abbruchbedingung					
Geniutest_err	V/O	0...FFH	0...255	1	[-]
GENIUTESTFehler					
Genutest_tol	V/O	0...FFH	0...99.60937	0.390625	[%]
Def_f_ub_b100					
I_craweng	V/O	8000...7FFFH	-32768...32767	1	[A]
Strombedarf Motor					
I_gen	V/O	0...FFH	0...255	1	[A]
Generatorstrom					

I_gentest	V/O	0...FFH	0...255	1	[A]
Modellierter Generatorstrom für Tester für Generatortest					
Id_bosmg	V/O	0...FFFFH	0...65535	1	[-]
Id für Motoröl (BOS Kombi)					
Id_bosmgt	V/O	0...FFH	0...255	1	[-]
ID für Motoröl (BOS-Tester)					
Id_bosmsg	V/O	0...FFH	0...255	1	[-]
Id für Motoröl (BOS Kombi)					
Id_bosrtak	O/V	0...FFH	0...255	1	[-]
Acknowledge reset durchgeführt					
Igrinfo[30]	V/O/S	0...FFH	0...255	1	[-]
information storage for IGR (intelligente Generatorreglung)					
Injekt_hub_h	O/V	0...FFH	0...255	1	[-]
Injektorhub Homogen					
Injekt_hub_hs	O/V	0...FFH	0...255	1	[-]
Injektorhub Homogen-Schicht					
Injekt_hub_s	O/V	0...FFH	0...255	1	[-]
Injektorhub Schicht					
Klann_mw1	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
Lambdaadaptionsmittelwert Bank1					
Klann_mw2	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
Lambdaadaptionsmittelwert Bank2					
Klann_test1	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
Lambdaadaptionswert für Tester Bank1					
Klann_test2	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
Lambdaadaptionswert für Tester Bank2					
Krnn_test	V/O	FE0C...258H	-50...60	0.1	[°CRK]
Testerwert NN KR-Adaption					
La_bs1	V/O	0...FFFFH	0...15.99975	0.2441e-3	[-]
lambda for catalyst overheating prevention bank 1					
La_bs2	V/O	0...FFFFH	0...15.99975	0.2441e-3	[-]
lambda for catalyst overheating prevention bank 2					
Leminfo[40]	V/O/S	0...FFH	0...255	1	[-]
information storage for LEM (Leistungskoordination Elektrisch Mechanisch)					
Md_can_dmee	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
theoretisches Moment für Getriebesteuerung					
Md_gennm_na	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
filtered torque losses due to alternator					
Md_getriebe_hs	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
torque losses for manual gear					
Md_max_klima	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
air-conditioning-compressor limit torque					
Md_na_ges	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
total torque losses for additive engine devices					
Md_na_ges_f	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
filtered total torque losses for additive engine devices					
Md_rad_istlm	O/V	80000000...7FFFFFFFH	-214748364.8...214748364.7	0.1	[Nm]
Ist-Radmoment aus Luftmasse ohne externe Eingriffe					
Md_reib_sa	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Reibmoment					
Md_res_max	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
max mögliche Momentenreserve					
Mdi_fzdyn	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres Wunschkupplungsmoment nach Fahrdynamikfilterung					
Mdi_fzdyn_res	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres Moment nach Fahrdynamikfilterung mit Momentenreserve					
Mdi_max	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
maximales inneres Moment					
Mdi_min_hs	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Minimal mögliche Moment in Schicht					

Mdi_min_s	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres minimales Moment im Schicht					
Mdi_opt_l1	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
reference indicated engine torque					
Mdi_nregl_plus	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Inneres Moment Drehzahlregler langsamer Pfad (erhöhend)					
Mdi_nregs_plus	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Inneres Moment Drehzahlregler schneller Pfad (erhöhend)					
Mdi_reib	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
torque losses					
Mdi_res_max	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres Moment incl. max mögliche Momentenreserve ohne Einfluss LL					
Mdi_res_maxll	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres Moment incl. max mögliche Momentenreserve mit Einfluss LL					
Mdi_soll	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres Sollmoment					
Mdi_soll_k	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres Sollmoment					
Mdi_soll_lk	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres Soll-Moment langsamer Pfad ohne Sicherheitseingriff					
Mdi_wunsch	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Inneres Fahrerwunschmoment mit Begrenzung ohne Eingriffe					
Mdi_zw_soll	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
torque setpoint from ignition path					
Mdi_zw_sollcan	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
inneres Sollmoment-über den Zw einzustellen					
Mdk_min	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Minimales Kupplungsmoment					
Mdk_wunsch	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
torque setpoint at clutch before filter					
Md_res_atlsvc	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Angeforderte Momentenreserve während der Turboladerdiagnose					
Mrnn_test_dk	V/O	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
Testerwert NN DK-Adaption					
Ms_res_max	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
max Luftmassenstrom der durch Momentenreserve angefordert werden kann					
Msa_arravrs[60]	V/O/S	0...FFH	0...255	1	[-]
Ringspeicher Abschaltverhinderer					
Msa_indexrs	V/O/S	0...FFH	0...255	1	[-]
Index fuer den Ringspeicher Abschaltverhinderer					
Msainfo[50]	V/O/S	0...FFH	0...255	1	[-]
information storage for MSA (Motor-Start/Stop-Automatik)					
Msastz	V/O/S	0...FFFFH	0...65535	1	[-]
Zaehler aller Startversuche					
Msastzmsa	V/O/S	0...FFFFH	0...65535	1	[-]
Zaehler aller MSA-Startversuche					
Msdk	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
mass flow through throttle					
Msdk_f	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
mass flow through throttle					
Msnlgofs_tmp	V/O/S	8000...7FFFH	-1024... 1023.96875	0.03125	[kg/h]
Leckluft Adaptionswert					
Mszyl	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
mass flow to cylinder					
Mszyl_ges	V/O	0...FFFFH	0...6553.5	0.1	[kg/h]
total inducted air mass, including EGR					
N_dzm	O/V	0...7FFFH	0...32767	1	[rpm]
Angezeigte Drehzahl					
N_gen	V/O	0...7FFFH	0...32767	1	[rpm]
Generatordrehzahl					

N_nstart	V/O	0...7FFFH	0...32767	1	[rpm]
Motordrehzahl nach Motorstart					
Nelueft_wm	V/O	0...FFH	0...99.60937	0.390625	[%]
Solldrehzahl E-Lüfter Vorgabe WM					
Newp_soll	V/O	0...FFH	0...255	1	[-]
Solldrehzahl elektrische Wasserpumpe					
Nkw_poel_notl	V/O	0...7FFFH	0...32767	1	[rpm]
Max. Drehzahl bei defektem Ölsystem (Bauteilschutz)					
Nkw_poel_soll	V/O	0...7FFFH	0...32767	1	[rpm]
Soll LL-Drehzahl aus Öldruckregelung (für HLL, Testerbetrieb...)					
Nslb	V/O	0...7FFFH	0...32767	1	[rpm]
engine speed correction due to battery load-bilanz					
Nsl_koor	O/V	8000...7FFFH	-32768...32767	1	[rpm]
LL-Solldrehzahl, koordiniert					
Pmbackup[7]	V/O	0...FFH	0...255	1	[-]
Backupspeicher					
Prnn_test	O/V	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Testerwert NN PR-Adaption					
Prnn_test_agr	O/V	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Testerwert NN AGR-Adaption					
Qv_h2o	V/O/S	0...FFFFH	0...63.99902	0.9766e-3	[-]
bisheriger Wasserverlust					
Qv_h2oquali	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Qualtitätswert für Wasserverlust					
Qv_h2ostatus	O/V/S	0...FFH	0...255	1	[Ah]
Statuswert für Entwicklung Wasserverlust					
Oz_krz2cnt	V/O	0...FFH	0...255	1	[-]
Zähler 20er- Packet Niveaumessung					
Oz_krzcnt	V/O	0...FFH	0...255	1	[-]
Zähler 10er- Packet Niveaumessung					
Oz_krzkor2k	V/O	0...FFH	0...74.70703	0.2929688	[-]
Niveaumessung 20er- Packet					
Oz_krzkor	V/O	0...FFH	0...74.70703	0.2929688	[-]
Niveaumessung OrientierungsMw					
Oz_kvbog	V/O/S	0...FFFFH	0...65535	1	[-]
Zugeteilte Bonuskraftstoffmenge					
Oz_kvbsm_ul	V/O/S	0...FFFFFFFFH	0...524288	0.1221e-3	[-]
Kraftstoffverbrauch seit letztem Ölwechsel					
Oz_lf1t	V/O/S	0...FFH	0...2.55	0.01	[-]
Länderfaktor 1 codiert					
Oz_lf2t	V/O/S	0...FFH	0...2.55	0.01	[-]
Länderfaktor 2 codiert					

Oz_lgmwcnt	V/O	0...FFH	0...255	1	[-]
Zähler LangzeitMw Niveaumessung					
Oz_lp	V/O	0...FFH	0...255	1	[-]
Für "Ölkannenanzeige im Display"					
Oz_lv	V/O	0...FFH	0...255	1	[-]
Erforderliche Nachfüllmenge Motoröl					
Oz_nivkor	V/O	0...FFH	0...74.70703	0.2929688	[-]
Niveaumessung korrigiertes Ölniveau					
Oz_oelkm	V/O/S	0...FFFFH	0...655350	10	[km]
Oelkilometer					
Oz_oelzeit	V/O	0...FFFFH	0...65535	1	[-]
Öllaufzeit					
Oz_oricnt	V/O	0...FFH	0...255	1	[-]
Zähler OriMw Niveaumessung					
Oz_permbog	V/O/S	0...FFFFH	0...5.99990	0.0916e-3	[-]
Zugeteilter Permittivitätsbonus					
Oz_permex	V/O/S	0...FFFFH	0...5.99990	0.0916e-3	[-]
Permittivität für Bewertung aufgereitet (extrapoliert)					
Oz_permlow	V/O/S	0...FFFFH	0...5.99990	0.0916e-3	[-]
Aufbereitete Permittivität bei letztem Ölwechsel					
Oz_permoff	V/O	0...FFFFH	0...5.99990	0.0916e-3	[-]
Offset für Permittivitätskorrektur					
Oz_rwkvb	O/V	8000...7FFFH	-327680... 327670	10	[km]
Restweg aus Kraftstoffverbrauch abgeleitet					
Oz_rwperm	O/V	8000...7FFFH	-327680... 327670	10	[km]
Restweg aus Permittivität abgeleitet					
Oz_vormw	V/O	0...FFH	0...74.70703	0.2929688	[-]
Niveaumessung VorMw					
Oz_vormwcnt	V/O	0...FFH	0...255	1	[-]
Zähler VorMw Niveaumessung					
Poel_fpwm	V/O	0...FFH	0...255	1	[Hz]
Frequenz PWM Ansteuerung Öldruck-Regelventi					
P_oel_soll	V/O	0...FFFFH	0...65535	1	[hPa]
Sollwert Öldruck					
Pldr_soll	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
Ladedruck Sollwert					
Ps	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
manifold pressure					
Ps_ist	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
measured manifold pressure					
Pspvdk	V/O	0...FFFFH	0...1	0.0153e-3	[-]
pressure quotient at throttle					
Pspvdk_soll	V/O	0...FFFFH	0...3.99993	0.061e-3	[-]
Druckverhältnis über Drosselklappe Sollwert					
Psreg_i	O/V	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Saugrohrdruckregler I-Anteil					
Q_abgnox	O/V	8000...7FFFH	-20480... 20479.375	0.625	[W]
Wärmestrom [W] von Abgas an Abgasanlage hinter Noxkat					
Qv_nv_ezm	V/O/S	8000...7FFFH	-32...31.99902	0.9766e-3	[-]
mittlerer Fehler über alle Hystereseberechnungen					
Qv_nv_start	V/O/S	80...7FH	-128...127	1	[A]
Aus Start berechneter Kapazitätsverlust					
Qv_nv_zh	V/O/S	0...FFFFFFFFH	0...4294967295	1	[-]
Zahl der Hystereseberechnungen					
Qv_out_1	V/O/S	80...7FH	-128...127	1	[A]
Kapazitätsverlust letzter Start					
Qv_out_2	V/O/S	80...7FH	-128...127	1	[A]
Kapazitätsverlust 2. letzter Start					

Qv_out_3	V/O/S	80...7FH	-128...127	1	[A]
Kapazitätsverlust 3. letzter Start					
Qv_out_4	V/O/S	80...7FH	-128...127	1	[A]
Kapazitätsverlust 4. letzter Start					
Qv_out_5	V/O/S	80...7FH	-128...127	1	[A]
Kapazitätsverlust 5. letzter Start					
Qv_out_m	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Gültiger gemittelter Kapazitätsverlust					
Qv_quali_1	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Qualitätsindex für letzten Qv-Wert					
Qv_quali_2	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Qualitätsindex für 2. letzten Qv-Wert					
Qv_quali_3	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Qualitätsindex für 3. letzten Qv-Wert					
Qv_quali_4	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Qualitätsindex für 4. letzten Qv-Wert					
Qv_quali_5	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Qualitätsindex für 5. letzten Qv-Wert					
Qv_quali_m	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Qualitätsindex für gemittelten Qv-Wert					
Qv_status	V/O/S	80...7FH	-128...127	1	[-]
Prozessstatus / Trend für gemittelten Qv-Wert					
Qv_td1	V/O/S	0...FFFFH	0...65535	1	[h]
Zeit seit Qv_out_1 - Berechnung					
Qv_td2	V/O/S	0...FFFFH	0...65535	1	[h]
Zeitabstand zwischen Qv_out_1 und Qv_out_2					
Qv_td3	V/O/S	0...FFFFH	0...65535	1	[h]
Zeitabstand zwischen Qv_out_2 und Qv_out_3					
Qv_td4	V/O/S	0...FFFFH	0...65535	1	[h]
Zeitabstand zwischen Qv_out_3 und Qv_out_4					
Qv_td5	V/O/S	0...FFFFH	0...65535	1	[h]
Zeitabstand zwischen Qv_out_4 und Qv_out_5					
Qvc_status_1	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Ausgang für Schlüsselgröße 1					
Qvc_status_2	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Ausgang für Schlüsselgröße 2					
Qvc_status_3	O/V/S	0...FFH	0...255	1	[Ah]
Ausgang für Schlüsselgröße 3					
Qvc_status_4	V/O/S	80...7FH	-128...127	1	[-]
Ausgang für Schlüsselgröße 4					
Rf	V/O	0...7530H	0...300	0.01	[%]
relative cylinder filling					
Rf_ig_ist	V/O	0...7530H	0...300	0.01	[%]
Inertgas-Anteil im Brennraum					
Rf_res_max	V/O	0...7530H	0...300	0.01	[%]
Maximale rel. Füllung aus MD-Reserve für Katheizen					
Rf_soll	V/O	0...7530H	0...300	0.01	[%]
setpoint relative cylinder filling					
Rf_vl	V/O	0...7530H	0...300	0.01	[%]
maximum setpoint relative cylinder filling					
Rfv_vns	V/O	0...7530H	0...300	0.01	[%]
Relative Füllung für Vanos					
Rk_add_wf	O/V	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
Wandfilmkorrektur DI Homogen					
Rk_kor_h	O/V	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
rel. Kraftstoffmasse homogen stationär					
Rk_kor_kva	O/V	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
Korrigierte Kraftstoffmasse für Kraftstoffverbrauchsanzeige					

Rk_korres_hs	O/V	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
Korrigiertes Rk_soll mit Momentenreserve (homogen-schicht)					
Rk_korres_s	O/V	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
Korrigiertes Rk_soll mit Momentenreserve (schicht)					
Rk_vlo_s	V/O	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
Umschaltgrenze Schicht					
Rk_vlo_hs	V/O	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
Umschaltgrenze Homogenschicht					
Rkte_max	V/O	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
Max zul. Menge Kraftstoff aus TE					
Rkte_md	V/O	8000...7FFFH	-1600... 1599.95117	0.0488281	[%]
Momentenrelevanter Teil der Kraftstoffmasse					
Rt_bastatg_h	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Relative Zeit Homogen Betrieb gesamter Motorlauf					
Rt_bastatg_hs	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Relative Zeit Homogen-Schicht-Betrieb gesamter Motorlauf					
Rt_bastatg_s	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Relative Zeit Schicht-Betrieb gesamter Motorlauf					
Rt_bastatg_sa	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Relative Zeit Homogen Betrieb gesamter Motorlauf					
Spa_art	V/O	0...FFH	0...255	1	[-]
Anzeigeart für Schaltpunktanzeige					
Spa_art_2	V/O	0...FFH	0...255	1	[-]
Anzeigeart für Schaltpunktanzeige 2					
Spa_gang	V/O	0...FFH	0...255	1	[-]
Gang für die Schaltpunktanzeige					
St_atldiag2_out	O/V	0...FFH	0...255	1	[-]
Statusbyte Interface Ausgang					
St_atlstat	V/O	0...FFH	0...255	1	[-]
Statuswort Laderstatus					
St_atlsvc	O/V	0...FFH	0...255	1	[-]
Status während der Diagnose der Abgasturbolader					
St_atlsvc_pvdk	O/V	0...FFH	0...255	1	[-]
Ergebnis der Diagnosefunktion der Abgasturbolader					
St_atlsvc_start	O/V	0...FFH	0...255	1	[-]
Status nach Aufruf der Diagnosefunktion der Abgastutbolader					
St_dgen0	V/O	0...FFH	0...255	1	[-]
Statusbyte DGEN					
St_dgenerrst_md1	V/O/S	0...FFFFH	0...65535	1	[-]
Statusbyte Fehlerzuweisung zu Messdatensatz1					
St_dgenerrst_md2	V/O/S	0...FFFFH	0...65535	1	[-]
Statusbyte Fehlerzuweisung zu Messdatensatz2					
St_dgengrenz1	V/O/S	0...FFH	0...31.875	0.125	[A]
Mittelwert der Erregerstrombegrenzung über applizierbare Zeit X (z.B. 10min)					
St_dgengrenz1_md1	V/O/S	0...FFH	0...31.875	0.125	[A]
Mittelwert der Erregerstrombegrenzung über applizierbare Zeit X (z.B. 10min) zu 1. Messdatensatz					
St_dgengrenz1_md2	V/O/S	0...FFH	0...31.875	0.125	[A]
Mittelwert der Erregerstrombegrenzung über applizierbare Zeit X (z.B. 10min) zu 2. Messdatensatz					
St_dgengrenz2	V/O/S	0...FFH	0...31.875	0.125	[A]
Mittelwert der Erregerstrombegrenzung über applizierbare Zeit Y (z.B. 30min)					
St_dgengrenz2_md1	V/O/S	0...FFH	0...31.875	0.125	[A]

Mittelwert der Erregerstrombegrenzung über applizierbare Zeit Y (z.B. 30min) zu 1. Messdatensatz					
St_dgengrenz2_md2	V/O/S	0...FFH	0...31.875	0.125	[A]
Mittelwert der Erregerstrombegrenzung über applizierbare Zeit Y (z.B. 30min) zu 2. Messdatensatz					
St_dgengrenzerr	V/O/S	0...FFH	0...255	1	[-]
Fehlerstati zur Erregerstrombegrenzung					
St_dgengrenzerr_md1	O/V/S	0...FFH	0...255	1	[-]
Fehlerstati zur Erregerstrombegrenzung zu 1. Messdatensatz					
St_dgengrenzerr_md2	V/O/S	0...FFH	0...255	1	[-]
Fehlerstati zur Erregerstrombegrenzung zu 2. Messdatensatz					
St_dgengrenznz	V/O/S	0...FFH	0...31.875	0.125	[A]
Mittelwert der Erregerstrombegrenzung über applizierbare Zeit Z (z.B. 2min)					
St_dgengrenznz_md1	V/O/S	0...FFH	0...31.875	0.125	[A]
Mittelwert der Erregerstrombegrenzung über applizierbare Zeit Z (z.B. 2min) zu 1. Messdatensatz					
St_dgengrenznz_md2	V/O/S	0...FFH	0...31.875	0.125	[A]
Mittelwert der Erregerstrombegrenzung über applizierbare Zeit Z (z.B. 2min) zu 2. Messdatensatz					
St_dgenub1	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Mittelwert der Batteriespannung über applizierbare Zeit X (z.B. 10min)					
St_dgenub1_md1	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Mittelwert der Batteriespannung über applizierbare Zeit X (z.B. 10min) zu 1. Messdatensatz					
St_dgenub1_md2	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Mittelwert der Batteriespannung über applizierbare Zeit X (z.B. 10min) zu 2 Messdatensatz					
St_dgenub2	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Mittelwert der Batteriespannung über applizierbare Zeit X (z.B. 30min)					
St_dgenub2_md1	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Mittelwert der Batteriespannung über applizierbare Zeit X (z.B. 30min) zu 1. Messdatensatz					
St_dgenub2_md2	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Mittelwert der Batteriespannung über applizierbare Zeit X (z.B. 30min) zu 2. Messdatensatz					
St_dgenuberr	V/O/S	0...FFH	0...255	1	[-]
Fehlerstati zur Batteriespannung					
St_dgenuberr_md1	V/O/S	0...FFH	0...255	1	[-]
Fehlerstati zur Batteriespannung zu 1. Messdatensatz					
St_dgenuberr_md2	V/O/S	0...FFH	0...255	1	[-]

Fehlerstati zur Batteriespannung zu 2. Messdatensatz					
St_dgenubnz	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Statuswort Messung Spannung Ub					
St_dgenubnz_md1	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Mittelwert der Batteriespannung über applizierbare Zeit Z (z.B. 2min) zu 1. Messdatensatz					
St_dgenubnz_md2	V/O/S	0...FFFFH	0...6173.397	0.0942	[V]
Mittelwert der Batteriespannung über applizierbare Zeit Z (z.B. 2min) zu 2. Messdatensatz					
St_dgenugen1	V/O/S	0...FFFFH	0...6553.5	0.1	[V]
Mittelwert der Generatorsollspannung über applizierbare Zeit X (z.B. 10min)					
St_dgenugen1_md1	V/O/S	0...FFFFH	0...6553.5	0.1	[V]
Mittelwert der Generatorsollspannung über applizierbare Zeit X (z.B. 10min) zu 1. Messdatensatz					
St_dgenugen1_md2	V/O/S	0...FFFFH	0...6553.5	0.1	[V]
Mittelwert der Generatorsollspannung über applizierbare Zeit X (z.B. 10min) zu 2. Messdatensatz					
St_dgenugen2	V/O/S	0...FFFFH	0...6553.5	0.1	[V]
Mittelwert der Generatorsollspannung über applizierbare Zeit Y (z.B. 30min)					
St_dgenugen2_md1	V/O/S	0...FFFFH	0...6553.5	0.1	[V]
Mittelwert der Generatorsollspannung über applizierbare Zeit X (z.B. 30min) zu 1. Messdatensatz					
St_dgenugen2_md2	V/O/S	0...FFH	0...25.5	0.1	[V]
Mittelwert der Generatorsollspannung über applizierbare Zeit X (z.B. 30min) zu 2. Messdatensatz					
St_dgenugenerr	V/O/S	0...FFH	0...255	1	[-]
Fehlerstati zur Generatorsollspannung					
St_dgenugenerr_md1	V/O/S	0...FFH	0...255	1	[-]
Fehlerstati zur Generatorsollspannung zu 1. Messdatensatz					
St_dgenugenerr_md2	V/O/S	0...FFH	0...255	1	[-]
Fehlerstati zur Generatorsollspannung zu 2. Messdatensatz					
St_dgenugennz	V/O/S	0...FFFFH	0...6553.5	0.1	[V]
Statuswort Messung Spannung Ugen					
St_dgenugennz_md1	V/O/S	0...FFFFH	0...6553.5	0.1	[V]
Mittelwert der Generatorsollspannung über applizierbare Zeit Z (z.B. 2min) zu 1. Messdatensatz					
St_dgenugennz_md2	V/O/S	0...FFFFH	0...6553.5	0.1	[V]
Mittelwert der Generatorsollspannung über applizierbare Zeit Z (z.B. 2min) zu 2. Messdatensatz					
St_dsc_mradist	O/V	0...FFH	0...255	1	[-]
Status Ist-Radmoment					
St_eisydiag1	V/O	0...FFH	0...255	1	[-]
Diagnose Mslam					
St_eisydiag2	V/O	0...FFH	0...255	1	[-]
Diagnose Mslam					

St_ecostat2	V/O/S	0...FFH	0...255	1	[-]
Ecos Statusbyte 2					
St_eisy_hfm	V/O	0...FFH	0...255	1	[-]
Status Eisy adaption					
St_eisyad_read	V/O/S	0...FFH	0...255	1	[-]
Status Eisy adaption					
St_entlast_loc	V/O	0...FFH	0...255	1	[-]
?????					
St_fas_mradist	O/V	0...FFH	0...255	1	[-]
Status Radmoment Ist					
St_fw	V/O	0...FFH	0...255	1	[-]
Statuswort mdfw					
St_gen	V/O	0...FFH	0...255	1	[-]
Status Generator					
St_gentest	V/O	0...FFH	0...255	1	[-]
Status Generatortest					
St_gl_adapt	V/O	0...FFH	0...255	1	[-]
Statusbyte adaption KH					
St_ldstgen	V/O	0...FFH	0...255	1	[-]
Lastzustand Generator					
St_igroutnv	V/O/S	0...FFH	0...255	1	[-]
Statusbyte St_igrout					
St_injad_quit	O/V	0...FFH	0...255	1	[-]
Quittung Adaptionsanforderung d. Kleinstmengen					
St_klann_ad1	V/O	0...FFH	0...255	1	[-]
Statusbyte fuer klann_ad1					
St_klann_ad2	V/O	0...FFH	0...255	1	[-]
Statusbyte fuer klann_ad2					
St_llranh0	V/O	0...FFH	0...255	1	[-]
Statusbyte Leerlaufdrehzahlanhebung					
St_mdar0	V/O	0...FFH	0...255	1	[-]
Statuswort mdar 0					
St_mdar1	V/O	0...FFH	0...255	1	[-]
Statuswort mdar 1					
St_mdbanl2	V/O	0...FFH	0...255	1	[-]
Statusbyte Notlaufmanager					
St_mdbanl_pwgll	O/V	0...FFH	0...255	1	[-]
Statuswort für Bits					
St_mdk	V/O	0...FFH	0...255	1	[-]
Status Momentenstruktur					
St_mdrk2	V/O	0...FFH	0...255	1	[-]
Statusbyte					
St_msdssm	V/O	0...FFH	0...255	1	[-]
Statuswort mdssm					
St_mini	V/O	0...FFH	0...255	1	[-]
Statuswort für Bits					
St_msaa	V/O	0...FFH	0...255	1	[-]
Statusbyte msaa					
St_msaanz	V/O	0...FFFFH	0...65535	1	[-]
Statusword msaanz					
St_msabs	V/O	0...FFH	0...255	1	[-]
Statusbyte MSA-Betriebstrategie					
St_ozbg0	V/O	0...FFH	0...255	1	[-]
Statuswort OZBG					
St_nmax_ba	V/O	0...FFH	0...255	1	[-]
Statuswort für CC Meldung Drehzahlbegrenzung aus MDBA					
St_poelreg2	V/O	0...FFH	0...255	1	[-]
Status Öldruckregelung 2					
St_prailpulskor	O/V	0...FFH	0...255	1	[-]
Container für Bits					
St_prestoppmsa	O/V	0...FFH	0...255	1	[-]

Statuswort f. MSA-Prestopp					
St_qvc1	V/O	0...FFH	0...255	1	[-]
Status QVC					
St_rk_kva	O/V	0...FFH	0...255	1	[-]
Container für Bits					
St_sondenheiz	O/V	0...FFH	0...255	1	[-]
Statusbyte St_sondenheiz					
St_statmkor1	V/O	0...FFH	0...255	1	[-]
Statuswort					
St_testpoelsys	V/O	0...FFH	0...255	1	[-]
Statuswort für Überprüfung Ölsystem im Testerbetrieb					
St_testpoelsys2	V/O	0...FFH	0...255	1	[-]
Zusätzliche Info zu Überprüfung Öldrucksystem im Testerbetrieb					
St_ti_b	V/O	0...FFH	0...255	1	[-]
Status "Begrenzung der Einspritzzeit"					
St_ugenkor0	V/O	0...FFH	0...255	1	[-]
Status Ugenkor					
St_vbrzyl_aus	V/O	0...FFFFH	0...65535	1	[-]
Status Zylindergleichstellung					
St_wesb_s	O/V	0...FFH	0...255	1	[-]
Status Einspritzwinkel 3. Einspritzung Schicht Wesb/Wese					
St_zwbts1	V/O	0...FFH	0...255	1	[-]
Status Bauteilschutzfunktion					
St_zwgrund	V/O	0...FFH	0...255	1	[-]
Statuswort ZWGRUND					
Stat_fofumx_anf	O/V	0...FFFFH	0...65535	1	[-]
Anforderung maximale Folgefunken (Zylinderselektiv)					
Stat_lvs	O/V	0...FFH	0...255	1	[-]
Stufe Laufruheverbesserung in Schicht					
Stat_zyl_aus	O/V	0...FFH	0...255	1	[-]
Zylinderausblendanforderung					
T_batt	V/O	FE0C...5DCH	-50...150	0.1	[°C]
battery temperature					
Tevt_mdl	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Temperatur am Einlassventil modelliert					
Tget_b1	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Temperatur < 80°C					
Tget_b2	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
80°C =< Temperatur =< 109°C					
Tget_b3	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
110°C =< Temperatur =< 124°C					
Tget_b4	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
125°C =< Temperatur =< 129°C					
Tget_b5	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Temperatur > 129°C					
Tlrgen	V/O	0...FFH	0...25.5	0.1	[s]
physical input for load response time					
Tmot_b1	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Temperatur < 98°C					
Tmot_b2	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
98°C =< Temperatur =< 112°C					
Tmot_b3	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
113°C =< Temperatur =< 120°C					
Tmot_b4	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
121°C =< Temperatur =< 125°C					
Tmot_b5	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Temperatur > 125°C					
Toel	V/O	FE0C...5DCH	-50...150	0.1	[°C]
Öltemperaturausgabe					
Toel_oz	V/O	FE0C...5DCH	-50...150	0.1	[°C]
Öltemperaturausgabe					

Toel_b1	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Temperatur < 80°C					
Toel_b2	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
80°C =< Temperatur =< 110°C					
Toel_b3	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
110°C =< Temperatur =< 135°C					
Toel_b4	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
135°C =< Temperatur =< 150°C					
Toel_b5	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Temperatur > 150°C					
Tumg_abst	V/O	FE0C...5DCH	-50...150	0.1	[°C]
Abstell-Aussentemperatur					
Tumg_battab	V/O	FE0C...5DCH	-50...150	0.1	[°C]
Abstell-Umgebungstemperatur Batterie					
Tumg_b1	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Temperatur < 3°C					
Tumg_b2	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
3°C =< Temperatur =< 19°C					
Tumg_b3	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
20°C =< Temperatur =< 29°C					
Tumg_b4	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
30°C =< Temperatur =< 39°C					
Tumg_b5	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Temperatur > 39°C					
Tv_kft	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Ausgegebenes TV für KFT					
Tv_poel	V/O	0...FFFFH	0...0.99998	0.0153e-3	[-]
Tastverhältnis Öeldruck-Regelventil					
Tvneutral	O/V/S	0...FFFFH	0...655.35	0.01	[%]
Tastverhältnis Nullgangsensor					
Tw_noxsens	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Rohrwandtemperatur am Noxsensor					
U_gen	V/O	0...FFFFH	0...6553.5	0.1	[V]
generator voltage					
U_gentest	V/O	0...FFFFH	0...6553.5	0.1	[V]
Sollspannung für Tester für Generatortest					
Vsa_spri	V/O	0...FFFFH	0...6553.5	0.1	[°CRK]
spread angle outlet camshaft					
Vsa_sprn	V/O	0...FFFFH	0...6553.5	0.1	[°CRK]
spread angle outlet camshaft at reference conditions (warm engine)					
Vse_spri	V/O	0...FFFFH	0...6553.5	0.1	[°CRK]
spread angle inlet camshaft					
Vse_sprn	V/O	0...FFFFH	0...6553.5	0.1	[°CRK]
spread angle intlet camshaft at reference conditions (warm engine)					
Wdk_soll	V/O	0...1000H	0...100	0.0244141	[%]
setpoint throttle opening					
Wdk_diag	V/O	0...1000H	0...100	0.0244141	[%]
Berechneter Winkel der Drosselklappe aus Hauptfüllungsignal					
Wese3_h[NC_CYL_NR]	V/O	E200...5A00H	-180...540	0.0234375	[°CRK]
Einspritzendewinkel 3. Einspritzung Homogen absolut zum ZOT					
Wes_soll_h[NC_CYL_NR][NC_PLS_NR]	O/V	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Soll Einspritzwinkel homogen					
Wesb_s[NC_CYL_NR]	O/V	E200...5A00H	-180...540	0.0234375	[°CRK]
Winkel Einspritzbeginn Haupteinspritzung schicht zylinderindividuell					
Zrbosmld	V/O/S	0...FFH	0...255	1	[-]
Servicezähler für BOS-Meldung					
Zr_lvs_0	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufe 0 LVS					
Zr_lvs_1	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufe 1 LVS					
Zr_lvs_2	O/V/S	0...FFFFH	0...65535	1	[-]

Anzahl Stufe 2 LVS					
Zr_lvs_3	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufe 3 LVS					
Zr_lvs_ll_reakt	O/V*/S	0...FFH	0...255	1	[-]
Anzahl der schnellen LVS Reaktionen im Leerlauf					
Zr_lvssekt_0	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 0					
Zr_lvssekt_1	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 1					
Zr_lvssekt_2	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 2					
Zr_lvssekt_3	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 3					
Zr_lvssekt_4	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 4					
Zr_lvssekt_5	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 5					
Zr_lvssekt_6	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 6					
Zr_lvssekt_7	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 7					
Zr_lvssekt_8	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung in Sektor 8					
Zr_lvszyl_0	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung durch Zylinder 0					
Zr_lvszyl_1	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung durch Zylinder 1					
Zr_lvszyl_2	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung durch Zylinder 2					
Zr_lvszyl_3	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung durch Zylinder 3					
Zr_lvszyl_4	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung durch Zylinder 4					
Zr_lvszyl_5	O/V/S	0...FFFFH	0...65535	1	[-]
Anzahl Stufenerhöhung durch Zylinder 5					
Zw_offkorrvr[NC_CYL_NR]	O/V	FE0C...258H	-50...60	0.1	[°CRK]
Zündwinkelkorrektur durch Verbrennungsregelung					
Agr_soll	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Exhaused gas setpoint					
Agrpos_soll	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
AGR-Ventil Sollposition					
Ba_switch_uek	V/O	0...FFH	0...255	1	[-]
Schalter für Wandfilmkompensation					
Ctrcbr	V/O	0...FFH	0...255	1	[-]
Steuerung Stromzweige					
Ctrpcos	V/O	0...FFH	0...255	1	[-]
Steuerung Standverbraucher					
Ctrprio	V/O	0...FFH	0...255	1	[-]
Steuerung Spitzenreduzierung Priorität					
Ctrpriocf	V/O	0...FFH	0...255	1	[-]
Steuerung Spitzenreduzierung Priorität Komfort					
Ctrpwrcos	V/O	0...FFFFH	0...65535	1	[-]
Steuerung Leistung Sonderverbraucher					
Ctrpwspcos	V/O	0...FFH	0...127.5	0.5	[%]
Steuerung Leistung Sonderverbraucher					
Dwese2_h[NC_CYL_NR]	V/O	E200...5A00H	-180...540	0.0234375	[°CRK]
Differenz zw. Einspritzende 2. Einspritzung Homogen und Zündung					
Eta_labas	V/O	8000...7FFFH	-200...199.99389	6.1035e-3	[%]
Lambdawirkungsgrad bei Lambda_bas_cor					
Eta_labas_sol	V/O	8000...7FFFH	-200...199.99389	6.1035e-3	[%]
Lambda-Wirkungsgrad					

Eta_lambda_1	V/O	8000...7FFFH	-200...199.99389	6.1035e-3	[%]
Lambda-Wirkungsgrad, Bank1					
Eta_lambda_2	V/O	8000...7FFFH	-200...199.99389	6.1035e-3	[%]
Lambda-Wirkungsgrad, Bank 2					
F_lradap1	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
Lambdaadaptionswert Bank1					
F_lradap2	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
Lambdaadaptionswert Bank2					
F_rkkorwl_hs	V/O	0...FFFFH	0...3.99993	0.061e-3	[-]
Warmlaufkorrektur Homogen-Schicht nur für 1. Einspritzung					
F_rkkorwl_s	V/O	0...FFFFH	0...3.99993	0.061e-3	[-]
Warmlaufkorrektur Schicht nur für 1. Einspritzung					
F_sicher_zw	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Überblendfaktor aus Verbrennungsregelung für Zündwinkel					

<table>
<tr><td>I_genmax</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[A]</td></tr>
<tr><td colspan="6">Generatormaximalstrom</td></tr>
<tr><td>Ierrgrenz</td><td>V/O</td><td>0...FFH</td><td>0...31.875</td><td>0.125</td><td>[A]</td></tr>
<tr><td colspan="6">Begrenzter Erregerstrom Generator 1</td></tr>
<tr><td>La_abgas1</td><td>V/O</td><td>0...FFFFH</td><td>0...15.99975</td><td>0.2441e-3</td><td>[-]</td></tr>
<tr><td colspan="6">Berechnete Lambda in Abgas inklusiv Durchspülung in sch, hos und hom</td></tr>
<tr><td>La_abgas2</td><td>V/O</td><td>0...FFFFH</td><td>0...15.99975</td><td>0.2441e-3</td><td>[-]</td></tr>
<tr><td colspan="6">Berechnete Lambda in Abgas inklusiv Durchspülung in sch, hos und hom</td></tr>
<tr><td>La_bas1</td><td>V/O</td><td>0...FFFFH</td><td>0...15.99975</td><td>0.2441e-3</td><td>[-]</td></tr>
<tr><td colspan="6">Basis-Lambda soll Bank 1</td></tr>
<tr><td>La_bas2</td><td>V/O</td><td>0...FFFFH</td><td>0...15.99975</td><td>0.2441e-3</td><td>[-]</td></tr>
<tr><td colspan="6">Basis-Lambda soll Bank 2</td></tr>
<tr><td>La_ref_homm</td><td>V/O</td><td>0...FFFFH</td><td>0...15.99975</td><td>0.2441e-3</td><td>[-]</td></tr>
<tr><td colspan="6">Lambda Sollwert Homogen mager</td></tr>
<tr><td>Mdi_ist_m</td><td>V/O</td><td>D8F0...2710H</td><td>-1000...1000</td><td>0.1</td><td>[Nm]</td></tr>
<tr><td colspan="6">Über alle Zylinder gemitteltes inneres Motormoment</td></tr>
<tr><td>Mdi_max_l1</td><td>V/O</td><td>D8F0...2710H</td><td>-1000...1000</td><td>0.1</td><td>[Nm]</td></tr>
<tr><td colspan="6">Maximales inneres Moment</td></tr>
<tr><td>Mdi_opt_s</td><td>V/O</td><td>D8F0...2710H</td><td>-1000...1000</td><td>0.1</td><td>[Nm]</td></tr>
<tr><td colspan="6">Motormoment Schicht</td></tr>
<tr><td>Mdi_s_istm</td><td>V/O</td><td>D8F0...2710H</td><td>-1000...1000</td><td>0.1</td><td>[Nm]</td></tr>
<tr><td colspan="6">Ist-Moment im Schichtbetrieb über alle Zylinder gemittelt</td></tr>
<tr><td>Mdi_soll_l</td><td>V/O</td><td>8000...7FFFH</td><td>-3276.8...3276.7</td><td>0.1</td><td>[Nm]</td></tr>
<tr><td colspan="6">inneres Sollmoment</td></tr>
<tr><td>Mk_soll_h[NC_CYL_NR][NC_NR_IV_PLS]</td><td>V/O</td><td>0...FFFFH</td><td>0...1389</td><td>0.0211948</td><td>[mg/stk]</td></tr>
<tr><td colspan="6">Kraftstoffmasse Homogen als Sollwert</td></tr>
<tr><td>Mk_soll_s</td><td>V/O</td><td>0...FFFFH</td><td>0...1389</td><td>0.0211948</td><td>[mg/stk]</td></tr>
<tr><td colspan="6">Kraftstoffmasse Schicht als Sollwert</td></tr>
<tr><td>Mkzyl_soll_hs[NC_CYL_NR][NC_NR_IV_PLS]</td><td>O/V</td><td>0...FFFFH</td><td>0...1389</td><td>0.0211948</td><td>[mg/stk]</td></tr>
<tr><td colspan="6">Kraftstoffmasse Homogenschicht als Sollwert (zylinderindividuell, je Einspritzung)</td></tr>
<tr><td>Mkzyl_soll_s[NC_CYL_NR][4]</td><td>O/V</td><td>0...FFFFH</td><td>0...1389</td><td>0.0211948</td><td>[mg/stk]</td></tr>
<tr><td colspan="6">Kraftstoffmasse Schicht als Sollwert (zylinderindividuell, je Einspritzung)</td></tr>
<tr><td>Msagr</td><td>V/O</td><td>0...FFFFH</td><td>0...2047.96875</td><td>0.03125</td><td>[kg/h]</td></tr>
<tr><td colspan="6">AGR-Massenstrom</td></tr>
<tr><td>Prail_soll</td><td>V/O</td><td>0...FFFFH</td><td>0...39.99938</td><td>0.6104e-3</td><td>[MPa]</td></tr>
<tr><td colspan="6">Raildruckvorgabe von BMW, gültig nur wenn B_prailsw gesetzt ist.</td></tr>
<tr><td>Prail_out_s</td><td>V/O</td><td>0...FFFFH</td><td>0...39.99938</td><td>0.6104e-3</td><td>[MPa]</td></tr>
<tr><td colspan="6">Raildruckvorgabe von BMW, gültig nur wenn B_prailsw gesetzt ist.</td></tr>
<tr><td>Pminfo1[37]</td><td>V/O/S</td><td>0...FFFFH</td><td>0...65535</td><td>1</td><td>[%]</td></tr>
<tr><td colspan="6">Infospeicher 1</td></tr>
<tr><td>Pminfo2[29]</td><td>V/O/S</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Infospeicher 2</td></tr>
<tr><td>Pssol</td><td>V/O</td><td>0...FFFFH</td><td>0...2559.96093</td><td>0.0390625</td><td>[hPa]</td></tr>
<tr><td colspan="6">Soll-Saugrohrdruck</td></tr>
<tr><td>Rqpcos</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anforderung Standverbraucher</td></tr>
<tr><td>Selspcos</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Selection special consumers</td></tr>
<tr><td>St_ba_agf</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Status Betriebsarten angefordert</td></tr>
<tr><td>St_blsdisa2</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Statuswort BLSDISA Ausgang</td></tr>
<tr><td>St_blshub</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Statuswort BLSHUB</td></tr>
<tr><td>St_disa_bmw</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Status Disa</td></tr>
<tr><td>St_imdgen0</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Statuswort Imdgen</td></tr>
<tr><td>St_kath_ena</td><td>V/O</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Statuswort erlaubte Katheizmassnahmen</td></tr>
</table>

St_mdbafak	V/O	0...FFH	0...255	1	[-]
Status MDBAFAK					
St_mdbafak2	V/O	0...FFH	0...255	1	[-]
Status MDBAFAK					
St_oz_o	V/O	0...FFH	0...255	1	[-]
Statusbyte für Ausgänge OZ					
St_pmprsvcalc1	V/O	0...FFH	0...255	1	[-]
Statuswort PMPRSVCALC					
St_pr	V/O	0...FFH	0...255	1	[-]
Status Raildruck					
St_tiausgang	V/O	0...FFH	0...255	1	[-]
Status Einspritzung					
Stpcos	V/O	0...FFH	0...255	1	[-]
Status Standverbraucher					
Td_wese23min_hs[NC_CYL_NR]	O/V	0...FFFFH	0...262.14	0.004	[ms]
minimaler Zeitabstand Einspritzung zw. 2 und 3 in Homogen-Schicht					
Td_wese[NC_CYL_NR][2]	O/V	0...FFFFH	0...262.14	0.004	[ms]
Zeitabstände Einspritzung Schicht (Zylinder, 1-2)					
Tgen	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Generatortemperatur					
Ubt	V/O	0...FFFFH	0...61.425	0.015	[V]
voltage battery					
ulev	O	0...FFH	0...255	1	[-]
Spannungsebene (internal used)					
Ulev	O	0...FFH	0...255	1	[-]
Spannungsebene (internal used)					
Wesb1_h[NC_CYL_NR]	V/O	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Einspritzbeginn 1. Einspritzung Homogen					
Wesb1_hs[NC_CYL_NR]	V/O	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Einspritzbeginn 1. Einspritzung Homogen-stratified					
Wesb2max_h	O/V	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Frühest möglicher Einspritzbeginnwinkel 2. Einspritzung Homogen					
Wesb4_s[NC_CYL_NR]	O/V	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Einspritzbeginn 3. Einspritzung Schicht					
Wesbkh_s[NC_CYL_NR]	O/V	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Einspritzbeginn Katheizeinspritzung Schicht					
Wese2_h[NC_CYL_NR]	V/O	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Einspritzendewinkel 2. Einspritzung Homogen absolut zum ZOT					
Wese2_hs[NC_CYL_NR]	V/O	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Einspritzende 2. Einspritzung Homogen-stratified					
Wese3_hs[NC_CYL_NR]	O/V	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Einspritzendewinkel 3. Einspritzung Homogen-Schicht					
Wese_s[NC_CYL_NR]	V/O	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Winkel Einspritzende Haupteinspritzung Schicht zylinderindividuell					
Zw_grund1	V/O	FE0C...258H	-50...60	0.1	[°CRK]
Grundzündwinkel ohne Kr mit Kr- und ROZ-Adaption Bank 1					
Zw_grund2	V/O	FE0C...258H	-50...60	0.1	[°CRK]
Grundzündwinkel ohne Kr mit Kr- und ROZ-Adaption Bank 2					
Zw_opt1	V/O	FE0C...258H	-50...60	0.1	[°CRK]
Optimaler Zündwinkel mit Tmot- und Lambdaeinfluss Bank 1					
Zw_opt2	V/O	FE0C...258H	-50...60	0.1	[°CRK]
Optimaler Zündwinkel mit Tmot- und Lambdaeinfluss Bank 2					
Zw_soll_s[NC_CYL_NR]	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°CRK]
Zündwinkel der über die TPU ausgegeben wird. Dieser Zündwinkel ist für den Homogen- und den Schichtbetrieb gültig.					
Zw_soll_hs[NC_CYL_NR]	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°CRK]
Zündwinkel der über die TPU ausgegeben wird. Dieser Zündwinkel ist für den Homogen- Schichtbetrieb gültig.					
Dzw_krkorll	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°CRK]
Verschiebung ZW-Frühgrenze bei Klopfen im LL					
Absch_korr	V/O/S	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
Korrekturwert Abschaltung					

Ba_wm_ist	V/O	0...FFH	0...255	1	[-]
Ist-Betriebsart Wärmemanagementkoordinator					
Bosfxid2	V/O/S	0...FFFFH	0..65535	1	[-]
ID_Funktion_BOS_Meldung_2					
Bosid2	V/O/S	0...FFH	0...255	1	[-]
ID2_BOS_Meldung_2					
D_soc	V/O/S	0...7FFFH	0...99.99694	3.0518e-3	[-]
Abstand zur Startfähigkeitsgrenze					
F_wnwkh	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for fading ignition angle in case of catalyst heating					
F_zw_hsplit	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Faktor zur Überblendung von Größen des Zündpfads					
Ibhwversi	V/O	0...FFH	0...255	1	[-]
IBS Hardwareversion					
Ibsderrs1	V/O/S	0...FFH	0...255	1	[-]
Dibs Fehlerzähler BSD					
Ibsderrs2	V/O/S	0...FFFFH	0...65535	1	[-]
Dibs Fehlerzähler EBSD					
Ibspreco2	V/O/S	0...FFH	0...255	1	[-]
Vorteiler für Ibsderrs2					
Ibswbase	V/O	0...FFH	0...255	1	[-]
IBS Softwarebaseline					
Ibswchang	V/O	0...FFH	0...255	1	[-]
IBS SW-Änderungsstatus					
Ktupcsctr	V/O/S	0...FFFFH	0...65535	1	[-]
Fehlerzähler EBSD Checksumme					
Md_rad_ksoll	V/O	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
koordiniertes Wunschradmoment					
Md_rad_ist	V/O	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Ist-Moment an den Antriebsrädern					
Md_rad_max	V/O	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Max. Radmoment					
Md_rad_min	V/O	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Schleppmoment unbefeuert					
Nmax_ba	V/O	0...7FFFH	0...32767	1	[rpm]
Maximal-Drehzahl bei aktueller Betriebsart					
Oz_nivakt	V/O	0...FFH	0...74.70703	0.2929688	[-]
Aktuelles Ölniveau für den DIS-Tester					
Oz_nivkrzt	V/O	0...FFH	0...74.70703	0.2929688	[-]
Kurzzeit-Ölniveau-Mittelwert für den DIS-Tester					
Oz_nivlangt	V/O/S	0...FFH	0...74.70703	0.2929688	[-]
Langzeit-Ölniveau-Mittelwert für den DIS-Tester					
Oz_reset	V/O	0...FFH	0...255	1	[-]
Status der Reset-Art an Kombi					
Ptc_pwr	V/O	0...FFH	0...127.5	0.5	[%]
Maximale Leistung, die der PTC einstellen darf					
Snibs	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Seriennummer IBS					
St_dibs0	V/O/S	0...FFH	0...255	1	[-]
Bitbasis dibs					
Bosjziel	V/O/S	0...FFH	0...255	1	[-]
Zieltermin_Jahr_BOS_Meldung_2					
Bosmziel	V/O/S	0...FFH	0...255	1	[month]
Zieltermin_Monat_BOS_Meldung_2					
Bosprog2	V/O/S	0...FFH	0...255	1	[month]
Prognose_Intervall_Zeit_BOS_Meldung_2					
Bosres	V/O/S	0...FFH	0...255	1	[-]

BOS_Reserve					
Bosrw2	V/O/S	0...FFH	0...255000	1000	[km]
Prognose_Intervall_Weg_BOS_Meldung_2					
Tabg_r10	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemperatur nach Rohr 10, nach Kat, Bank2					
Tabg_r9	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemperatur nach Rohr 9, nach Kat, Bank1					
Tabg_vk1	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemperatur vor Kat, Bank1					
Tabg_vk2	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemperatur vor Kat, Bank2					
Tw_m1	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Rohrwandtemperatur vor Kat, Bank 1					
Tw_m2	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Rohrwandtemperatur vor Kat, Bank 2					
U_batt	V/O	0...FFFFH	6...22.38375	0.00025	[V]
Batteriespannung von IBS gemessen					
Zbibs	V/O/S	0...FFFFFFFFH	0...4294967295	1	[-]
Zusbaunummer					
St_mdinfo_ges	V/O	0...FFH	0...255	1	[-]
Status Momenteneingriffe					
St_ldm_kupp	V/O	0...FFH	0...255	1	[-]
Status_Kraftschluss					
Dzwo_lam_kor	V/O	80...7FH	-64...63.5	0.5	[°CRK]
Grundreferenzzündwinkel Lambdakorrektur					
Dzwot	V/O	80...7FH	-64...63.5	0.5	[°CRK]
Grundreferenzzündwinkel Temperaturkorrektur					
I_batt	V/O	0...FFFFH	-200...5042.8	0.08	[A]
battery current					
Mdi_ml_sollvb	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Soll-Moment Luftpfad vor Sicherheitsbegrenzung					
Mdi_zw_sollvb	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Soll-Moment Zündpfad vor Sicherheitsbegrenzung					
Mdk_wunsch_filt	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Gefiltertes Wunschmoment					
Oz_permakt	V/O	0...FFFFH	0...5.99990	0.0916e-3	[-]
Aktuelle Ölpermittivität für den DIS-Tester					
Oz_tempakt	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Aktuelle Öltemperatur für den DIS-Tester					
Pm_klemmstat	V/O	0...FFH	0...255	1	[-]
Klemmenstatus bei Tiefentladung					
Stat_sv_reg1	V/O	0...FFH	0...255	1	[-]
Status Standverbraucher registriert Teil 1					
Stat_sv_reg2	V/O	0...FFH	0...255	1	[-]
Status Standverbraucher registriert Teil 2					
T2histshort	V/O	0...FFH	0...3808	14.933333	[min]
Zeit, indem der Ruhestrom bei 80..200mA liegt					
T3histshort	V/O	0...FFH	0...3808	14.933333	[min]
Zeit, indem der Ruhestrom bei 200..1000mA liegt					
T4histshort	V/O	0...FFH	0...3808	14.933333	[min]
Zeit, indem der Ruhestrom größer als 1000mA liegt					
Tabg_demax1	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Max. Temperatur für Desulfatisierung Bank1					
Tabg_demax2	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Max. Temperatur für Desulfatisierung Bank2					
Tabg_demaxav	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Gemittelte max.Temperatur für Desulfatisierung					
Tabg_ikat_av	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemp. im Kat gemittelt					
Tabg_nk1	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemp. nach Kat Bank 1					

Tabg_nk2	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemp. nach Kat Bank 2					
Tabg_nk_av	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Gemittelte Temperatur nach Kat					
Tabg_vk_av	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Gemittelte Temperatur vor Kat					
Tvwg	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Wastegate 1 PWM Ausgang					
Tvwg2	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Wastegate 2 PWM Ausgang					
St_ar1	V/O	0...FFH	0...255	1	[-]
Statuswort Antiruckel					
St_bns	V/O/S	0...FFH	0...255	1	[-]
Statusbyte Bordnetzstabilität					
St_deavns2	V/O/S	0...FFH	0...255	1	[-]
Status Bedingung Vanos im Anschlag					
St_hhs	V/O	0...FFH	0...255	1	[-]
carrierbyte HHS					
St_ngang	O/V/S	0...FFH	0...255	1	[-]
Status Nullgangerkennung					
St_ngang0	V/O	0...FFH	0...255	1	[-]
Status Nullgangerkennung					
St_pmdiag0	V/O	0...FFH	0...255	1	[-]
Statuswort PMDIAG					
St_pmdiag1	V/O	0...FFH	0...255	1	[-]
Statuswort PMDIAG					
St_rail	V/O	0...FFH	0...255	1	[-]
Statuswort Raildruckregler					
St_rf1	V/O	0...FFH	0...255	1	[-]
Statusbyte					
St_sa	V/O	0...FFH	0...255	1	[-]
Statuswort Schubabschaltung					
St_sa_atmph	V/O	0...FFH	0...255	1	[-]
Statusbyte Ausgang Schubabschaltung von ATMPH					
St_sleep	V/O	0...FFH	0...255	1	[-]
Statusbyte Sleep					
St_ubw	V/O/S	0...FFH	0...255	1	[-]
carrierbyte "Spannungswaechter"					
St_vsa	V/O	0...FFH	0...255	1	[-]
Status Anschlagadaption Auslass					
St_vse	V/O	0...FFH	0...255	1	[-]
Status Anschlagadaption Einlass					
St_wese	V/O	0...FFH	0...255	1	[-]
Status Einspritzwinkel					
St_wm1	V/O	0...FFH	0...255	1	[-]
Statusbyte fuer Ausgang WM-Koordinator					
Stmsa	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Status MSA und Funktionsabschalter					
Stmsaaa	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Status MSA Abschaltaufforderer					
Stmsaea	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Status MSA Einschaltaufforderer					
Stmsaav	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Status MSA aktiv und bereit und Abschaltverhinderer					
Stmsaev	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Status MSA Einschaltverhinderer					
T_ikat1	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemperatur im Kat, Bank1					
T_ikat2	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Abgastemperatur im Kat, Bank2					
T_ikat1_stat	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]

stationäre Abgastemperatur im Kat, Bank1					
T_ikat2_stat	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
stationäre Abgastemperatur im Kat, Bank2					
Tw_nk1	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Rohrwandtemperatur nach Kat, Bank 1					
Tw_nk2	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Rohrwandtemperatur nach Kat, Bank 2					
Tvulv	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Tastverhältnis Umluftventil					
St_dps	V/O	0...FFH	0...255	1	[-]
Statuswort					
Mdk_wunsch_ipm	O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Torque request for IPM					
F_wnwkhk	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Überblendfaktor (warm/kalt) für Spreizung und Zündhaken während Katheizen					
Mdk_max	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
max. torque at clutch					
Mdi_ml_soll	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
Max. torque at air					
Bszsi	V/O/S	0...FFFFFFFFH	0...4294967295	1	[-]
Betriebsstundenzaehler					
Igenk	O/V/S	F32	-	FLOAT	[-]
Generator aufintegriert					
Dfds[16]	V/O/S	0...FFH	0...255	1	[-]
Generatorauslastungsprofil					
Lurabs_f[NC_CYL_NR]	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[-]
gefilterte Laufruhedeltas eines Zylinders					
Lurdif_f[NC_CYL_NR]	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[-]
gefilterte Laufruhedeltas eines Zylinders					
Rt_bastatg_sa	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Betriebsstundenzaehler homogen					
Tabg_desul	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
max. Temperatur im Unterbodenkat für Desulfatisierung					
Tabg_inok1	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Temperatur im Noxkat 1.Mono Bank1					
Tabg_inok2	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Temperatur im Noxkat 1.Mono Bank2					
Tabg_nnok1	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Temperatur im Noxkat 2.Mono Bank1					
Tabg_nnok2	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Temperatur im Noxkat 2.Mono Bank2					
Tabg_nok_av	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Mittelwert der Temperaturen im Monolith Unterbodenkat					
Tabg_nok_av1	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Mittelwert der Temperaturen im Monolith Unterbodenkat Bank1					
Tabg_nok_av2	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Mittelwert der Temperaturen im Monolith Unterbodenkat Bank2					
Tabg_nokmax	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Maximum der Temperaturen im Monolith Unterbodenkat					
Tabg_nokmax1	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Maximum der Temperaturen im Monolith Unterbodenkat Bank1					
Tabg_nokmax2	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Maximum der Temperaturen im Monolith Unterbodenkat bank 2					
Tabg_nokmin	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Minimum der Temperaturen im Monolith Unterbodenkat					
Tabg_asens	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
Modellierte Abgastemperatur am Temperatursensor (Rohr 15)					
I_ges	O/V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Gesamtübersetzung					
I_ha	O/V	0...FFFFH	0...63.99902	0.9766e-3	[-]
Übersetzung Hinterachse					

La_sollreg	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda setpoint for controller (with lambda pulsation)					
St_aekh	O/V/S	0...FFH	0...255	1	[-]
Carrier Byte "Aussetzererkennung Katheizen"					
St_aekh_ae	O/V	0...FFH	0...255	1	[-]
Statuswort Aussetzer erkannt zylinderselektiv					

General information:

Data description of BMW outputs. For more detailed information on datatypes and short description please refer to the BMW layer specification.

Application conditions:

Initialisation: 0

except :

I_ha: 63
Bosconf : from nonvolatile memory
Eisyagr_korfak_b: 1
F_atlad: from nonvolatile memory
F_ffboost : 1
F_szboost : 1
La_bs1/2 : 2
Oz_kvbsm_ul : from nonvolatile memory
Ps : Pu
St_ldstgen : 2
Tumg_abst : from nonvolatile memory
Tumg_battab : from nonvolatile memory
Eta_labas, Eta_labas_sol: 100%
Eta_lambda_1, Eta_lambda_2: 100 %
Nmax_ba: 9000 rpm
Absch_korr: from nonvolatile memory
Bosfxid2: from nonvolatile memory
Bosid2: from nonvolatile memory
D_soc: from nonvolatile memory
Eta_md_uesp: 100%
Eta_mdkh: 100%
F_bsmdres: 1.0
Ibsderrs1: from nonvolatile memory
Ibsderrs2: from nonvolatile memory
Ibspreco2: from nonvolatile memory
Ktupcsctr: from nonvolatile memory
La_sollreg[NC_CBK_EX_NR]: 1.0
Msa_arravrs[60] from nonvolatile memory
Msa_indexrs from nonvolatile memory
Msastz from nonvolatile memory
Msastzmsa from nonvolatile memory
Nmax_ba: 9000
Oz_nivlangt: from nonvolatile memory
Oz_oelkm: from nonvolatile memory
St_dibs0 from nonvolatile memory
Bosjziel from nonvolatile memory
Bosmziel from nonvolatile memory

Bosprog2	from nonvolatile memory
Bosres	from nonvolatile memory
Bosrw2	from nonvolatile memory
Tabg_r10	first valid value
Tabg_r9	first valid value
Tabg_vk1	first valid value
Tabg_vk2	first valid value
Tw_m1	first valid value
Tw_m2	first valid value
Stat_sv_reg1	first valid value
Stat_sv_reg2	first valid value
Tabg_demax1	first calculated value
Tabg_demax2	first calculated value
Tabg_demaxav	first calculated value
Tabg_ikat_av	first calculated value
Tabg_nk1	first calculated value
Tabg_nk2	first calculated value
Tabg_nk_av	first calculated value
Tabg_vk_av	first calculated value
Bosbtvfbk:	from nonvolatile memory
Bosrlsm:	from nonvolatile memory
Oz_kvbog:	from nonvolatile memory
Pminfo1[37]:	from nonvolatile memory
Pminfo2[29]:	from nonvolatile memory
Zrbosmld:	from nonvolatile memory
St_bns	from nonvolatile memory
St_ubw	from nonvolatile memory
St_deavns2	last calculated value
St_sleep	1
Tw_nk1	first calculated value
Tw_nk2	first calculated value
Oz_permbog	from non volatile memory
Oz_permex	from non volatile memory
Oz_permlow	from non volatile memory
Oz_lf1t:	from non volatile memory
Oz_lf2t:	from non volatile memory
Igrinfo[30]:	from non volatile memory
Leminfo[40]:	from non volatile memory
Msainfo[50]:	from non volatile memory
St_igroutnv:	from non volatile memory
Pspvdk_soll:	1.00
Psreg_i:	1.00
Prnn_test:	last calculated value
Prnn_test_agr:	last calculated value
Poel_fpwm:	244 Hz

Tabg_desul, Tabg_inok1, Tabg_inok2, Tabg_nnok1, Tabg_nnok2, Tabg_nok_av, Tabg_nok_av1, Tabg_nok_av2, Tabg_nokmax, Tabg_nokmax1, Tabg_nokmax2, Tabg_nokmin, Q_abgnox, Tw_noxsens:

first calculated value

Read from NVRAM:

Bszsi, Dfds[16], Igenk, Genitest_tol, Genitest_bol, Qv_h2o, Qv_h2oquali, Qv_h2ostatus, St_ecostat2, St_spa2, Zr_lvs_ll_reakt

Initialisation by BMW:

Bosstate, Id_bosmg, Id_bosmsg, Oz_lp, Oz_lv, Ba_ist,Krn_test, U_gen, Wdk_diag, Zbibs, Zw_grund1, Zw_grund2, Zw_opt1, Zw_opt2, Mrnn_test_dk, Qv_nv_ezm, Qv_nv_ezm, Qv_nv_start, Qv_nv_zh, Qv_out_1, Qv_out_2, Qv_out_3, Qv_out_4, Qv_out_5, Qv_out_m, Qv_quali_1, Qv_quali_2, Qv_quali_3, Qv_quali_4, Qv_quali_5, Qv_quali_m, Qv_status, Qv_td1, Qv_td2, Qv_td3, Qv_td4, Qv_td5, Qvc_status_1, Qvc_status_2, Qvc_status_3, Qvc_status_4, Msnlgofs_tmp, Tvneutral, St_dgen.., Mdi_max, St_ngang,

Calculated in PWL:

Last valid value: Dfds[16] (once), Lurabs_f/Lurdif_f, Rt_bastatg_sa

Recurrence: 10 ms

except :

segment : Zw_md[NC_CYL_NR] , Ps, Ps_ist, Pspvdk, Rf, Mk_soll_s, Mkzyl_soll_s[NC_CYL_NR] [NC_NR_INJ_PULSES+1], Msagr, Msdk, Msdk_f, Mszyl, Mszyl_ges, Lurdif_f/Lurabs_f, Wesb1_h, Wesb1_hs, Wese2_hs, Wese_s, Wese2_h[NC_CYL_NR], Mkzyl_soll_hs[NC_CYL_NR] [NC_NR_INJ_PULSES], Mk_soll_h[NC_CYL_NR] [NC_NR_INJ_PULSES], F_rkkorwl_hs, F_rkkorwl_s, Msnlgofs_tmp, Wese3_h[NC_CYL_NR], Amo_05, Amo_10, Amo_15, Amo_20, Amo_30, Amo_40, Dla_soll_puls[NC_CBK_EX_NR], Td_wese[NC_CYL_NR][2], Td_wese23min_hs[NC_CYL_NR], Wesb2max_h, Wesb4_s[NC_CYL_NR], Wese3_hs[NC_CYL_NR], Injekt_hub_h, Injekt_hub_hs, Injekt_hub_s, Stat_fofumx_anf, Stat_lvs, Stat_zyl_aus, Wesbkh_s[NC_CYL_NR], B_aekh_akt, St_aekh, St_aekh_ae, St_injad_quit, St_wesb_s, Wesb_s[NC_CYL_NR], Zw_offkorrvr[NC_CYL_NR], Rk_kor_kva, St_prailpulskor, St_rk_kva, F_tikorzsk[NC_CYL_NR], Espr_mod_soll[NC_CYL_NR], Wes_soll_h[NC_CYL_NR][NC_PLS_NR], Zr_lvs_ll_reakt

5 ms : Exwink_soll, Tv_vvtlr

20 ms: Wdk_diag, F_lradap1, F_lradap2

100 ms : Nsl_koor , I_craweng , I_gen , St_dgen0 , St_gen , St_ldstgen, Tv_kft, Ba_subsoll, Ba_subist, Ibsderrs1, Ibsderrs2, Ibspreco2, Ktupcsctr, Ptc_pwr, St_dibs0, U_batt, I_batt, Pm_klemmstat, Stat_sv_reg1, Stat_sv_reg2, T2histshort, T3histshort, T4histshort, St_pmdiag0, St_pmdiag1, Rt_bastatg_sa, Krn_test, U_gen, Mrnn_test_dk, Igrinfo[30], Leminfo[40], Msainfo[50], Prnn_test, Prnn_test_agr,St_entlast_loc, St_igroutnv, St_eisydiag1, St_eisydiag2, St_imdgen0, F_atlad, St_hhs, Klann_test1, Klann_test2, I_gentest, U_gentest, St_gentest, Eco_jobstat1, Eco_result1, St_ecostat2, Pmbackup[7], St_nmax_ba, St_vbrzyl_aus, St_atlsvc, St_atlsvc_pvdk, St_atlsvc_start, Zr_lvs_0, Zr_lvs_1, Zr_lvs_2, Zr_lvs_3, Zr_lvssekt_0, , Zr_lvssekt_1, Zr_lvssekt_2 , Zr_lvssekt_3, Zr_lvssekt_4, Zr_lvssekt_5, Zr_lvssekt_6, Zr_lvssekt_7, Zr_lvssekt_8, Zr_lvszyl_0, Zr_lvszyl_2,Zr_lvszyl_3, Zr_lvszyl_4, Zr_lvszyl_5, St_sondenheiz, Atlsvc_dpvdk1, Atlsvc_dpvdk2, Atlsvc_dpvdk3, St_atldiag2_out, Klann_mw1, Klann_mw2, St_mdbanl_pwgll

200 ms : Newp_soll , St_egs1, Md_max_klima, ba_wm_ist, Tabg_r10, Tabg_r9, Tabg_vk1, Tabg_vk2, Tw_m1, Tw_m2, Tabg_demax1, Tabg_demax2, Tabg_demaxav, Tabg_ikat_av, Tabg_nk1, Tabg_nk2, Tabg_nk_av, Tabg_vk_av, St_wm1, Tw_nk1, Tw_nk2, Tabg_desul, Tabg_inok1, Tabg_inok2, Tabg_nnok1, Tabg_nnok2, Tabg_nok_av, Tabg_nok_av1, Tabg_nok_av2, Tabg_nokmax, Tabg_nokmax1, Tabg_nokmax2, Tabg_nokmin, T_ikat1_stat, T_ikat2_stat, Tumg_b1, Tumg_b2, Tumg_b3, Tumg_b4, Tumg_b5, Tget_b1, Tget_b2, Tget_b3, Tget_b4, Tget_b5, Toel_b1, Toel_b2, Toel_b3, Toel_b4, Toel_b5, Tmot_b1, , Tmot_b2, Tmot_b3, Tmot_b4, Tmot_b5, T_ikat1, T_ikat2, F_bsmdres, Q_abgnox, Tw_noxsens

1000 ms : Bosbtvfbk , Bosrlsm , Bosstate , Dfmonitor , Nslb , Id_bosmg , Id_bosmgt , Id_bosmsg , Nelueft_wm , Oz_kvbog Oz_kvbsm_ul Oz_lp , Oz_lv , Oz_oelzeit, Oz_permbog, Oz_permex, Oz_permlow, Oz_permoff, Oz_rwkvb, Oz_rwperm, St_ozbg0 , Toel , Toel_oz , Tumg_abst , Tumg_battab , Zrbosmld , Absch_korr, Bosfxid2, Bosid2, D_soc, Oz_nivakt, Oz_nivkrzt, Oz_nivlangt, Oz_reset, Bosjziel, Bosmziel, Bosprog2, Bosres, Bosrw2, Oz_permakt, Oz_tempakt, Bszsi, Igenk, Dzw_annm, Ibhwversi, Ibswbase, Ibswchang, Snibs, Zbibs, Oz_lf1t, Oz_lf2t, Bosconf, Bostoken, Id_bosrtak, St_ubw, Oz_krz2cnt, Oz_krzcnt, Oz_krzkor2k, Oz_krzor, Oz_lgmwcnt, Oz_nivkor, Oz_oricnt, Oz_vormw, Oz_vormwcn, Qv_nv_ezm, Qv_nv_start, Qv_nv_zh, Qv_out_1, Qv_out_2, Qv_out_3, Qv_out_4, Qv_out_5, Qv_out_m, Qv_quali_1, Qv_quali_2, Qv_quali_3, Qv_quali_4, Qv_quali_5, Qv_quali_m, Qv_status, Qv_td1, Qv_td2, Qv_td3, Qv_td4, Qv_td5, Qvc_status_1, Qvc_status_2, Qvc_status_3, Qvc_status_4, Oz_oelkm, St_dgen.., Genitest_tol, Geniutest_ab, Geniutest_err, Genutest_tol, Qv_h2o, Qv_h2oquali, Qv_h2ostatus, St_qvc1

background : Rf_vl, F_wnwkh, F_wnwkhk
once per driving cycle : Dfds[16]

triggered by event : St_sleep

Activation: calculation activation is defined in BMW code

with special values at power latch phase*:*

0 : St_mdinfo_ges, St_ldm_kupp, F_wnwkh, Md_rad_ist, Md_rad_max, Md_rad_min, St_atlout, Dzwo_lam_kor, Dzwot, I_batt, Mdi_ml_sollvb, Mdi_zw_sollvb, Mdk_wunsch_filt, Tvwg, Tvwg2, St_ar1, St_rail, St_vsa, St_vse, Tvulv, Mdk_wunsch_ipm, Mdk_max, F_wnwkh, I_ges, Eta_labas, Eta_labas_soll, Eta_lambda_1, Eta_lambda_2,Dzw_annm, St_entlast_loc, St_gl_adapt, Bostoken, Id_bosrtak,F_zw_hsplit, Mdi_reib, Rk_kor_h, Dla_soll_puls[NC_CBK_EX_NR], Md_can_dmee, St_prestoppmsa, Espr_mod_soll[NC_CYL_NR], Wes_soll_h[NC_CYL_NR][NC_PLS_NR], Md_na_ges_f

1: Eisyagr_korfak_b F_rkkorwl_hs, F_rkkorwl_s, La_sollreg[NC_CBK_EX_NR], F_bsmdres

63: I_ha

100%: Eta_md_uesp, Eta_mdkh

9000: Nmax_ba

last calculated value : Amo_05, Amo_10, Amo_15, Amo_20, Amo_30, Amo_40, St_deavns2, Tw_nk1, Tw_nk2, Pssol, Mdi_max_I1, Krn_test, Pspdvk_soll, Mrnn_test_dk, Oz_lf1t, Oz_lf2t, St_imdgen0, Rfv_vns, Atlr, Atlvst, F_atlad, Pldr_soll, Klann_test1, Klann_test2, Oz_krz2cnt, Oz_krzcnt, Oz_krzkor2k, Oz_krzkor, Oz_lgmwcnt , Oz_nivkor, Oz_oricnt, Oz_vormw, Oz_vormwcnt, Qv_nv_ezm, Qv_nv_ezm, Qv_nv_start, Qv_nv_zh, Qv_out_1, Qv_out_2, Qv_out_3, Qv_out_4, Qv_out_5, Qv_out_m, Qv_quali_1, Qv_quali_2, Qv_quali_3, Qv_quali_4, Qv_quali_5, Qv_quali_m, Qv_status, Qv_td1, Qv_td2, Qv_td3, Qv_td4, Qv_td5, Qvc_status_1, Qvc_status_2, Qvc_status_3, Qvc_status_4, St_ngang0, Stmsa, Stmsaaa, Stmsaav, Stmsaev, T_ikat1_stat, T_ikat2_stat, F_lradap1, F_lradap2, F_sicher_zw, Msnlgofs_tmp, Stmsaaa, Stmsaea, Tvneutral, St_blshub, Oz_oelkm, St_mdbafak2, St_wese, St_blsdisa2, Drf_spuel, I_gentest, U_gentest, St_dgen..., St_gentest, St_testpoelsys, St_testpoelsys2, Eco_jobstat1, Eco_result1, Genitest_tol, Genitest_bol, Md_reib_sa, Qv_h2o, Qv_h2oquali, Qv_h2ostatus, St_ecostat2, St_qvc1, St_klann_ad1, St_klann_ad2, Msa_arravrs[60] ,Msa_indexrs, Msastz, Msastzmsa, St_msabs, St_nmax_ba, St_vbrzyl_aus, T_ikat1, T_ikat2, Dzw_agr_hs, Prnn_test, Prnn_test_agr, Md_res_atlsvc, St_atlsvc, St_atlsvc_pvdk, St_atlsvc_start, St_sondenheiz, Zw_offkorrvr[NC_CYL_NR], Rk_kor_kva, St_prailpulskor, St_rk_kva, St_atldiag2_out, F_tikorzsk[NC_CYL_NR], Mdi_max, Rk_korres_hs, Rk_korres_s, Klann_mw1, Klann_mw2, St_mdbanl_pwgll, St_ngang, Md_gennm_na, Zr_lvs_ll_reakt

last value : St_wm1, St_ewp, St_lklps1, Ba_ist, Wdk_diag, Bosconf, Bosmanip, St_hhs, St_msaa, St_msaanz, St_ubw, Stat_zyl_aus

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
LV_N_MAX_CLC_MDBAPRIO_10MS	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call mdbaprio_10ms					
LV_N_MAX_CLC_AUSY_TURB_SEG	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call ausy_turb_seg					
LV_N_MAX_CLC_WESE_SEG	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call wese_seg					
LV_N_MAX_CLC_MDRK_SEG	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call mdrk_seg					
LV_N_MAX_CLC_KLANN_20MS	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call klann_20ms					
LV_N_MAX_CLC_EISY_VERD_SEG	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call eisy_verd_seg					
LV_N_MAX_CLC_MDBAFAK_10MS	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call mdbafak_10ms					
LV_N_MAX_CLC_MDBANL_10MS	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call mdanl_10ms					
LV_N_MAX_CLC_ULVREG_10MS	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call ulvreg_10ms					
LV_N_MAX_CLC_RAILKO_10MS	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call railko_10ms					
LV_N_MAX_CLC_ULVOUT_10MS	O/V	0...1H	0...1	1	[-]
Flag indicating runtime reduction on function call ulvout_10ms					

Input data:

N	LV_VAR_TCHA		

General information:

This module is a customer request out of the BMW-SV Software Layer, where the scheduling of BMW tasks is defined. For detailled information on the scheduling of BMW tasks please refer to the corresponding layer specification.

To prevent the CPU from overload the scheduling of some BMW tasks may be halved via engine speed thresholds or switched of via LV_VAR_TCHA. In case engine speed exceeds the corresponding threshold only each 2nd function call will be conducted.

Application conditions:

Initialisation: all = 0 at reset

Recurrence: seg and 10ms, 20ms

Activation: always

Deactivation: -

Formula section:

Function call MDBAPRIO_10MS:

If N > C_N_MAX_CLC_MDBAPRIO_10MS

```
Then    LV_N_MAX_CLC_MDBAPRIO_10MS = 1
Endif

If      N < C_N_MAX_CLC_MDBAPRIO_10MS - C_N_HYS_CLC_MDBAPRIO_10MS
Then    LV_N_MAX_CLC_MDBAPRIO_10MS = 0
Endif


If      LV_N_MAX_CLC_MDBAPRIO_10MS = 1
Then    only each 2nd function call for MDBAPRIO_10MS will be conducted
Else    each function call for MDBAPRIO_10MS will be conducted
Endif
```

Function call WESE_SEG:

```
If      N > C_N_MAX_CLC_WESE_SEG
Then    LV_N_MAX_CLC_WESE_SEG = 1
Endif

If      N < C_N_MAX_CLC_WESE_SEG - C_N_HYS_CLC_WESE_SEG
Then    LV_N_MAX_CLC_WESE_SEG = 0
Endif


If      LV_N_MAX_CLC_WESE_SEG = 1
Then    only each 2nd function call for WESE_SEG will be conducted
Else    each function call for WESE_SEG will be conducted
Endif
```

Function call MDRK_SEG:

```
If      N > C_N_MAX_CLC_MDRK_SEG
Then    LV_N_MAX_CLC_MDRK_SEG = 1
Endif

If      N < C_N_MAX_CLC_MDRK_SEG - C_N_HYS_CLC_MDRK_SEG
Then    LV_N_MAX_CLC_MDRK_SEG = 0
Endif


If      LV_N_MAX_CLC_MDRK_SEG = 1
Then    only each 2nd function call for MDRK_SEG will be conducted
Else    each function call for MDRK_SEG will be conducted
Endif
```

Function call KLANN_20MS:

```
If      N > C_N_MAX_CLC_KLANN_20MS
Then    LV_N_MAX_CLC_KLANN_20MS = 1
Endif

If      N < C_N_MAX_CLC_KLANN_20MS - C_N_HYS_CLC_KLANN_20MS
```

```
Then    LV_N_MAX_CLC_KLANN_20MS = 0
Endif


If      LV_N_MAX_CLC_KLANN_20MS = 1
Then    only each 2nd function call for KLANN_20MS will be conducted
Else    each function call for KLANN_20MS will be conducted
Endif
```

Function call EISY_VERD_SEG:

```
If      N > C_N_MAX_CLC_EISY_VERD_SEG
Then    LV_N_MAX_CLC_EISY_VERD_SEG = 1
Endif

If      N < C_N_MAX_CLC_EISY_VERD_SEG - C_N_HYS_CLC_EISY_VERD_SEG
Then    LV_N_MAX_CLC_EISY_VERD_SEG = 0
Endif


If      LV_N_MAX_CLC_EISY_VERD_SEG = 1
Then    only each 2nd function call for EISY_VERD_SEG will be conducted
Else    each function call for EISY_VERD_SEG will be conducted
Endif
```

Function call MDBAFAK_10MS:

```
If      N > C_N_MAX_CLC_MDBAFAK_10MS
Then    LV_N_MAX_CLC_MDBAFAK_10MS = 1
Endif

If      N < C_N_MAX_CLC_MDBAFAK_10MS - C_N_HYS_CLC_MDBAFAK_10MS
Then    LV_N_MAX_CLC_MDBAFAK_10MS = 0
Endif


If      LV_N_MAX_CLC_MDBAFAK_10MS = 1
Then    only each 2nd function call for MDBAFAK_10MS will be conducted
Else    each function call for MDBAFAK_10MS will be conducted
Endif
```

Function call MDBANL_10MS:

```
If      N > C_N_MAX_CLC_MDBANL_10MS
Then    LV_N_MAX_CLC_MDBANL_10MS = 1
Endif

If      N < C_N_MAX_CLC_MDBANL_10MS - C_N_HYS_CLC_MDBANL_10MS
Then    LV_N_MAX_CLC_MDBANL_10MS = 0
Else    do nothing
Endif
```

```
If      LV_N_MAX_CLC_MDBANL_10MS = 1
Then    only each 2nd function call for MDBANL_10MS will be conducted
Else    each function call for MDBANL_10MS will be conducted
Endif
```

Function call RAILKO_10MS:

```
If      N > C_N_MAX_CLC_RAILKO_10MS
Then    LV_N_MAX_CLC_RAILKO_10MS = 1
Endif

If      N < C_N_MAX_CLC_RAILKO_10MS - C_N_HYS_CLC_RAILKO_10MS
Then    LV_N_MAX_CLC_RAILKO_10MS = 0
Else    do nothing
Endif


If      LV_N_MAX_CLC_RAILKO_10MS = 1
Then    only each 2nd function call for RAILKO_10MS will be conducted
Else    each function call for RAILKO_10MS will be conducted
Endif
```

Function call AUSY_TURB_SEG

```
If      N > C_N_MAX_CLC_AUSY_TURB_SEG
Then    LV_N_MAX_CLC_AUSY_TURB_SEG = 1
Endif

If      N < C_N_MAX_CLC_AUSY_TURB_SEG - C_N_HYS_CLC_AUSY_TURB_SEG
Then    LV_N_MAX_CLC_AUSY_TURB_SEG = 0
Endif


If      LV_N_MAX_CLC_AUSY_TURB_SEG = 1
Then    only each 2nd function call for AUSY_TURB_SEG will be conducted
Else    each function call for AUSY_TURB_SEG will be conducted
```

Functions, called depending on LV_VAR_TCHA:

```
If(1)   LV_VAR_TCHA = 1
Then(1)
        atlstat_10ms,
        atlad_100ms,
        atldiag_100ms,
        atldiag_10ms,
        atlgls_10ms,
        atlob_20ms,
        atlob_50ms,
        atlout_10ms,
        atlpump_10ms,
```

```
    atlreg_10ms,
    atlstat_10ms,
    blsatl_10ms,
    blsatl_20ms,
    blsatl_50ms,
    eisy_diag_nl100ms,
    mdrfmxatl_50ms           will be conducted

    Function call ULVOUT_10MS:

    If   N > C_N_MAX_CLC_ULVOUT_10MS
    Then        LV_N_MAX_CLC_ULVOUT_10MS = 1
    Endif

    If   N < C_N_MAX_CLC_ULVOUT_10MS - C_N_HYS_CLC_ULVOUT_10MS
    Then        LV_N_MAX_CLC_ULVOUT_10MS = 0
    Endif


    If   LV_N_MAX_CLC_ULVOUT_10MS = 1
    Then        only each 2nd function call for ULVOUT_10MS will be conducted
    Else        each function call for ULVOUT_10MS will be conducted
    Endif

    Function call ULVREG_10MS:

    If   N > C_N_MAX_CLC_ULVREG_10MS
    Then        LV_N_MAX_CLC_ULVREG_10MS = 1
    Endif

    If   N < C_N_MAX_CLC_ULVREG_10MS - C_N_HYS_CLC_ULVREG_10MS
    Then        LV_N_MAX_CLC_ULVREG_10MS = 0
    Endif


    If   LV_N_MAX_CLC_ULVREG_10MS = 1
    Then        only each 2nd function call for ULVREG_10MS will be conducted
    Else        each function call for ULVREG_10MS will be conducted
    Endif

Endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_CLC_MDBAPRIO_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call mdbaprio_10ms (default: 400rpm)					
C_N_HYS_CLC_MDBAPRIO_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call mdbaprio_10ms (default: 200 rpm)					
C_N_MAX_CLC_AUSY_TURB_SEG	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call ausy_turb_seg (default: 4000 rpm)					
C_N_HYS_CLC_AUSY_TURB_SEG	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call ausy_turb_seg (default: 200 rpm)					
C_N_MAX_CLC_WESE_SEG	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call wese_seg (default: 4000 rpm)					
C_N_HYS_CLC_WESE_SEG	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call wese_seg (default: 200 rpm)					
C_N_MAX_CLC_MDRK_SEG	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call mdrk_seg (default: 4000 rpm)					
C_N_HYS_CLC_MDRK_SEG	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call mdrk_seg (default: 200 rpm)					
C_N_MAX_CLC_KLANN_20MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call klann_20ms (default: 4000 rpm)					
C_N_HYS_CLC_KLANN_20MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call klann_20ms (default: 200 rpm)					
C_N_MAX_CLC_EISY_VERD_SEG	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call eisy_verd_seg (default: 4000 rpm)					
C_N_HYS_CLC_EISY_VERD_SEG	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call eisy_verd_seg (default: 200 rpm)					
C_N_MAX_CLC_MDBAFAK_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call mdbafak_10ms (default: 4000 rpm)					
C_N_HYS_CLC_MDBAFAK_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call mdbafak_10ms (default: 200 rpm)					
C_N_MAX_CLC_MDBANL_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call mdbanl_10ms (default: 4000 rpm)					
C_N_HYS_CLC_MDBANL_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call mdbanl_10ms (default: 200 rpm)					
C_N_MAX_CLC_ULVREG_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call ulvreg_10ms (default: 4000 rpm)					
C_N_HYS_CLC_ULVREG_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call ulvreg_10ms (default: 200 rpm)					
C_N_MAX_CLC_RAILKO_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call railko_10ms (default: 4000 rpm)					
C_N_HYS_CLC_RAILKO_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call railko_10ms (default: 200 rpm)					
C_N_MAX_CLC_ULVOUT_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed threshold for runtime reduction on function call ulvout_10ms (default: 4000 rpm)					
C_N_HYS_CLC_ULVOUT_10MS	1	0...1FE0H	0...8160	1	[rpm]
Engine speed hysteresis for runtime reduction on function call ulvout_10ms (default: 200 rpm)					

36.14 OBJ File Reference List for SW release 80B3800S

FUNCTION DESCRIPTION:

General information:

This document is used to handle the configuration mangement of desired implementation of the customer OBJ files. For each SW release a fitting list of OBJ file references has to be created. The list is sorted in alphabetic order.

Description:

- atlad
- atldiag
- atlgls
- atlob
- atlout
- atlpump
- atlreg
- atlstat
- atmnn
- atmph
- atmph_n53
- atmph_svc
- ausy_ak
- ausy_kat
- ausy_turb
- bgcreng
- bgkuppl
- bgllgen
- bls
- blsagr
- blsatl
- blsdisa
- blsdk
- blshub
- blstempk
- bs
- co2mng
- deavanos
- dgen
- dibs
- dqlt
- eavanos
- eavnskt
- eisy
- eisy_diag
- eisy_dk
- eisy_ev
- eisy_llk
- eisy_mr
- eisy_psreg
- eisy_sr
- eisy_verd
- eisyagr
- gen_fasta
- igr
- igrdiag
- igrsbcalc
- imdgen
- inv_ipol
- ipmna_dynkor
- ipmna_genkor
- ipmna_genstoff
- ipmna_loadresp
- ipmna_md2ierr
- ipmna_statmbilanz
- ipmna_statmkor
- kf_ipol
- kl_ipol
- krann
- kt_ibs
- lamko
- layer
- ldstgen
- lemdiag
- llpmctrl
- llranh
- lolimot
- mdanman
- mdar
- mdbafak
- mdbanl
- mdbaprio
- mdbaschalt
- mddmk
- mdfw
- mdfzdyn
- mdipmfw
- mdist
- mdla
- mdmax
- mdna
- mdres
- mdreskrll
- mdrfmxatl
- mdrk
- mlp
- mslam
- ozbg
- ozgg
- ozleit
- oznivdl
- ozperm
- ozrestkm
- oztemp
- ozvisc
- pgenllred
- pgentred
- pid_regler
- pmbattmng
- pmbattregen
- pmcbsbatt
- pmdiag
- pmmsa
- pmmsadiag
- pmprsvcalc
- pmsoccalc
- pmsocfit

pmsockor
pmsocstart
pmspgw
pmubnkor
pmukomp
pmverbentl
pmverbmng
poel_reg
railko
restgas
services_msd80
ss_msv
sst
ulvout
ulvreg
wes
wese
wm_nl
wm_start
wmewapu
wmkft
wmprio
wmschalt
wmzusta
zwaus
zwgrund
zwopt

No version numbers of OBJ-Files are available !

36.15 Carrierbyte bitmasking

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_aekh_akt	O/V	0...1H	0...1	1	[-]
Flag indicating " BMW seitige Aussetzererkennung im Katheizbetrieb aktiv"					
B_ar_akt	O/V	0...1H	0...1	1	[-]
Flag indicating "Anti jerk active"					
B_atlsvc	O/V	0...1H	0...1	1	[-]
Diagnosefunktion der Abgasturbolader aktiv					
B_atlsvc_khm	O/V	0...1H	0...1	1	[-]
Anforderung Katheizmanager während der Turboladerdiagnose					
B_bns	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Bordnetzstabilität"					
B_bosmsgc	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung für Sendung der BOS Meldung"					
B_bs1	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung bauteilschutz aktiv 1"					
B_bs2	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung bauteilschutz aktiv 2"					
B_ccbtnach	O/V	0...1H	0...1	1	[-]
Flag indicating "Batterie nachladen"					
B_ccbttdfk	O/V	0...1H	0...1	1	[-]
Flag indicating "Batterie prüfen"					
B_ccbttknt	O/V	0...1H	0...1	1	[-]
Flag indicating "Batteriekontakt /-anschlüsse prüfen!"					
B_ccpmerr	O/V	0...1H	0...1	1	[-]
Flag indicating "Batterieüberwachung ausgefallen"					
B_ccprio	O/V	0...1H	0...1	1	[-]
Flag indicating "Komfortverbraucher eingeschränk"					
B_ccruhver	O/V	0...1H	0...1	1	[-]
Flag indicating "Erhöhte Batterieentladung im Stand"					
B_cdxenonr	O/V	0...1H	0...1	1	[-]
Xenonverbau-Codierdaten lesen					
B_codierpm	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Sendeanforderung Anfrage Codierung PWM"					
B_dash	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Dashpozänderungsbegrenzung aktiv"					
B_disa1_auf	O/V	0...1H	0...1	1	[-]
Anforderung Disa 1 öffnen aus BMW-Modul					
B_disa2_auf	O/V	0...1H	0...1	1	[-]
Anforderung Disa 2 öffnen aus BMW-Modul					
B_disa_stopp	O/V	0...1H	0...1	1	[-]
Flag indicating "DISA Ventil schliessen"					
B_ecobusy	O/V/S	0...1H	0...1	1	[-]
Ecos Ruhestrommessung busy					
B_esp_h1	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Einspritzung homogen Bank 1"					
B_esp_h2	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Einspritzung homogen Bank 2"					
B_espr_start	O/V	0...1H	0...1	1	[-]
Flag indicating "Einspritzung wird ausgeführt"					
B_ext_zylstop	O/V	0...1H	0...1	1	[-]
Bedingung CILC und CYBL stoppen					
B_gangnull	O/V	0...1H	0...1	1	[-]
Bedingung MSA bis zum nächsten Zündungswechsel deaktivieren					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_genoff	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Generator ausgeschaltet"					
B_nggelernt	O/V	0...1H	0...1	1	[-]
Flag indicating "Nullgangposition gelernt"					
B_ngimlf	O/V	0...1H	0...1	1	[-]
Flag indicating "Nullgangsignal im Lernfenster"					
B_hd_start	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Einspritzung wird ausgeführt"					
B_hmm_akt	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung homogen mager aktiv"					
B_hom_akt	O/V	0...1H	0...1	1	[-]
Flag indicating "Homogenbetriebberechnung aktiv"					
B_hstoech	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Homogenstoechiometrisch angefordert"					
B_kl61	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Klemme 61"					
B_kupp	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Kupplung"					
B_la_stopp1	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Lambdaregler Bank 1 stopp"					
B_la_stopp2	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Lambdaregler Bank 2 stopp"					
B_lklps	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Luftklappensteuerung Zu"					
B_lklps1	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Luftklappensteuerung down Zu"					
B_llrein	O/V	0...1H	0...1	1	[-]
Logical variable for coordination engine start torque and minimum torque and clutch					
B_lradap1_mn	O/V	0...1H	0...1	1	[-]
Gemischadaption am min. Anschlag, Bank1					
B_lradap1_mx	O/V	0...1H	0...1	1	[-]
Gemischadaption am max. Anschlag, Bank1					
B_lradap2_mn	O/V	0...1H	0...1	1	[-]
Gemischadaption am min. Anschlag, Bank2					
B_lradap2_mx	O/V	0...1H	0...1	1	[-]
Gemischadaption am max. Anschlag, Bank2					
B_lroff	O/V	0...1H	0...1	1	[-]
Flag indicating "Drehzahlschwelle für LR-Funktion Generator 1aktiv"					
B_lsd	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Lastschlagdämpfungsänderungsbegrenzung aktiv"					
B_hlsh1_off	O/V	0...1H	0...1	1	[-]
Bedingung Sondenheizung LSU Nachkat Bank 1 ausschalten					
B_hlsh2_off	O/V	0...1H	0...1	1	[-]
Bedingung Sondenheizung LSU Nachkat Bank 2 ausschalten					
B_hlsu1_off	O/V	0...1H	0...1	1	[-]
Bedingung Sondenheizung LSU Vorkat Bank 1 ausschalten					
B_hlsu2_off	O/V	0...1H	0...1	1	[-]
Bedingung Sondenheizung LSU Vorkat Bank 2 ausschalten					
B_hnox_off	O/V	0...1H	0...1	1	[-]
Bedingung Heizung Nox-Sensor ausschalten					
B_maregdk_ad	O/V	0...1H	0...1	1	[-]
Adaption Massenstromregler auf DK erstmalig erfolgt					
B_mdimin_sa	O/V	0...1H	0...1	1	[-]
Bedingung minimal mögliches Moment eingestellt für Freigabe Schubabschalten					
B_mol_sf	O/V	0...1H	0...1	1	[-]
Flag indicating "Bed. Ölzustands-Sensorfehler"					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_mol_vb	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Ölverbrauchswarnung"					
B_mol_vl	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Ölverbrauchswarnung"					
B_mrmslam	O/V	0...1H	0...1	1	[-]
Bedingung MS-Abgleich soll über Lambda passieren					
B_msaakt	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung MSA aktiv"					
B_msaaktber	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung MSA aktiv und bereit"					
B_msaanz1	O/V	0...1H	0...1	1	[-]
Flag indicating " MSA-Anzeige bei Abschaltverhinderer"					
B_msaanz2	O/V	0...1H	0...1	1	[-]
Flag indicating " MSA-Anzeige bei Einschaltverhinderer"					
B_msaccid2	O/V	0...1H	0...1	1	[-]
Bedingung MSA deaktiviert					
B_msaerr	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Fehler MSA"					
B_msaprestopp	O/V	0...1H	0...1	1	[-]
Preliminary MSA engine stop request					
B_msastart	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Startanforderung MSA"					
B_msastartf	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Startfreigabe MSA"					
B_msastarts	O/V	0...1H	0...1	1	[-]
Motor wegen MSA gestartet					
B_msastopp	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Stoppanforderung MSA"					
B_msastops	O/V	0...1H	0...1	1	[-]
Flag indicating "MSA-Stoppstatus für Powermanagement"					
B_mslam1_diag	O/V	0...1H	0...1	1	[-]
not used flag					
B_mslam2_diag	O/V	0...1H	0...1	1	[-]
not used flag					
B_nesps_akt	O/V	0...1H	0...1	1	[-]
Flag indicating "Nacheinspritzung Schicht wird ausgeführt"					
B_nesps_ena	O/V	0...1H	0...1	1	[-]
Flag indicating "Nacheinspritzung Schicht möglich"					
B_nmax_ba_cc	O/V	0...1H	0...1	1	[-]
Bedingung CC Meldung durch Drehzahlbegrenzung aus MDBA					
B_notafu	O/V	0...1H	0...1	1	[-]
Getriebe Not-Anfahrunterstützung					
B_nplmslam2_mn	O/V	0...1H	0...1	1	[-]
not used flag					
B_npmlslam2_mx	O/V	0...1H	0...1	1	[-]
not used flag					
B_nplmslam_mn	O/V	0...1H	0...1	1	[-]
not used flag					
B_nplmslam_mx	O/V	0...1H	0...1	1	[-]
not used flag					
B_nsub	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Solldrehzahlanhebung bei niedriger Ladebilanz"					
B_ntlkws	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Notlauf Kühlkreislauf"					
B_oeltemp	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung gültiger Öltemp Messwert"					
B_oelverb	O/V	0...1H	0...1	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Flag indicating "Bedingung für ausgelöste Ölverbrauchswarnung"					
B_oelverl	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung für ausgelöste Ölverlustswarnung"					
B_oztemic	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung QLT-Temic-verbaut"					
B_pmbsdanf	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung POWERMANAGMENT BATTERIESPANNUNG Sendeanforderung"					
B_pmvsdanf	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung POWERMANAGMENT VERBRAUCHERSTEUERUNG Sendeanforderung"					
B_poel_gelb	O/V	0...1H	0...1	1	[-]
Bedingung Öldruckkontrolllampe gelb (CC ID 427)					
B_poel_rot	O/V	0...1H	0...1	1	[-]
Bedingung Öldruckkontrollleuchte rot (CC ID 212)					
B_prailpulskor_ee	O/V	0...1H	0...1	1	[-]
Druckpulsationskorrektur für homogene Einfacheinspitzung ein-/ausgeschaltet					
B_prailpulskor_me	O/V	0...1H	0...1	1	[-]
Druckpulsationskorrektur für homogene Mehrfacheinspitzung ein-/ausgeschaltet					
B_prsoll_h	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Raildruckkennfeld"					
B_psr_ad	O/V	0...1H	0...1	1	[-]
Adaption Druckregler auf Hub erstmalig erfolgt					
B_psragr_ad	O/V	0...1H	0...1	1	[-]
Adaption AGR					
B_qvch2o	O/V	0...1H	0...1	1	[-]
Bit für zu hohen Wasserverlus					
B_rkkor_kva	O/V	0...1H	0...1	1	[-]
Anforderung Verbrauchsberechnung mit korrigierter Kraftstoffmasse					
B_sch_akt	O/V	0...1H	0...1	1	[-]
Flag indicating "Schichtberechnung aktiv"					
B_schicht	O/V	0...1H	0...1	1	[-]
Flag indicating "Motor läuft im Schichtbetrieb"					
B_sleepwait	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung einschlafen verhindern"					
B_spa_cist	O/V	0...1H	0...1	1	[-]
Zurückgemeltdete Codierung SPA					
B_temxon	O/V	0...1H	0...1	1	[-]
Flag indicating "Begrenzung der Einspritzzeit"					
B_tev_stopp	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Stop Tev"					
B_tmmi_warn1	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Warnlampe Stufe 1 (gelb) an MMI"					
B_tmmi_warn2	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Warnlampe Stufe 2 (rot) an MMI"					
B_tmmi_warn3	O/V	0...1H	0...1	1	[-]
Flag indicating "Ansteuerung der MMI-Meldung "bitte hochschalten""					
B_vb_sa	O/V	0...1H	0...1	1	[-]
Flag indicating "Schubabschalten verboten wg. niedriger KatTemp."					
B_vbkhm_notl	O/V	0...1H	0...1	1	[-]
Bedingung Verbot Katheizen					
B_ums_vbsa	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Schubabschaltung verboten"					
B_verb_h	O/V	0...1H	0...1	1	[-]
Flag indicating "Verbot Homogen"					
B_verb_off	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung elektrische Verbraucher aus"					
B_verb_s	O/V	0...1H	0...1	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Flag indicating "Verbot Schicht"					
B_vsa_an	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Vanos Auslass im Anschlag"					
B_vsaadp	O/V	0...1H	0...1	1	[-]
Flag indicating "Anschlagadaption Auslassspreizung variable NWS"					
B_vlschalt	O/V	0...1H	0...1	1	[-]
Flag indicating "Lader Status"					
B_vse_an	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Vanos Einlass im Anschlag"					
B_vseadp	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Vanos Einlass im Anschlag"					
B_vsean_loc	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Vanos Einlass im Anschlag beim letzten abstellen"					
B_wdkdiag_ugd	O/V	0...1H	0...1	1	[-]
Bedingung Berechnung DKWinkel aus Signal des HF-Sensors ungedrosselt					
B_wese2h_abs	O/V	0...1H	0...1	1	[-]
Winkel für 2. Einspritzung Homogen aus Wese2_h[Anz_Zyl] benutzen					
B_zw_dynman	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Dynamik"					
B_zwdyn	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung Dynamik für Modul zwgrund"					
B_gdst	O/V	0...1H	0...1	1	[-]
Flag indicating " Schnelle Rampe fuer Vollhubeinstellung "					
B_esp_hs1	O/V	0...1H	0...1	1	[-]
Injection homogeneous-stratiefied mode bank 1					
B_esp_hs2	O/V	0...1H	0...1	1	[-]
Injection homogeneous-stratiefied mode bank 2					
B_hschalt_komb	O/V	0...1H	0...1	1	[-]
Upshift request to instrument cluster (CC message)					
B_ldm_akt04	O/V	0...1H	0...1	1	[-]
LDM active					
B_ldm_off	O/V	0...1H	0...1	1	[-]
Shut off of LDM by ECU					
B_rschalt_komb	O/V	0...1H	0...1	1	[-]
Downshift request to instrument cluster (CC message)					
B_vb_sa_nk	O/V	0...1H	0...1	1	[-]
Prohibition of clinder cut off due to high underfloor temperature					
B_prailsw	O/V	0...1H	0...1	1	[-]
External fuel pressure setpoint request					
B_zwmin_ext	O/V	0...1H	0...1	1	[-]
Bedingung minimaler Zündwinkel von extern					
St_abgasklappe	O/V	0...FFH	0...255	1	[-]
Statuswort für Abgasklappe					
St_agr	O	0...FFH	0...255	1	[-]
Carrierbyte for status exhaust gas recirculation					
St_anf_prail	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status pressure request					
St_anman1	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of anman					
St_atldiag2_in	O/V	0...FFH	0...255	1	[-]
Statusbyte Interface Eingang					
St_auss	O/V	0...FFH	0...255	1	[-]
Status Aussetzerschwellen					
St_ba	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of operating mode					
St_bgkuppl	O/V	0...FFH	0...255	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Carrierbyte for status of clutch					
St_bls	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of brake light switch					
St_blsdisa1	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of ???					
St_btvns	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of VANOS					
St_ckl15	O/V/S	0...FFH	0...255	1	[-]
carrierbyte for status of terminal signals via CAN					
St_clc	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of combustion mode					
St_cybl	O/V	0...FFFFH	0...65535	1	[-]
Carrierword for Gleichstellung					
St_cybl2	O/V	0...FFH	0...255	1	[-]
Statuswort für Bits					
St_desul	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of desulfuration					
St_devxdetec	O/V/S	0...FFH	0...255	1	[-]
Ergebnis BSD-Lernfunktion					
St_dk_diag	O/V/S	0...FFH	0...255	1	[-]
carrierbyte for status of TPS diagnosis					
St_dkg	O/V	0...FFH	0...255	1	[-]
Anforderungsmodus					
St_dkheiz	O/V	0...FFH	0...255	1	[-]
Statuswort für Drosselklappenheizung					
St_dkratio	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of TPS ???					
St_dtev	O/V	0...FFH	0...255	1	[-]
Status TEV-Diagnose					
St_dyndzm	O/V	0...FFH	0...255	1	[-]
Statuswort					
St_ecostat1	O/V	0...FFH	0...255	1	[-]
Ecos Statusbyte 1					
St_egs	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of AT (automatic transmission)					
St_ekp	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of electronic fuel pump					
St_elu	O/V/S	0...FFH	0...255	1	[-]
carrierbyte for status of electrical cooling fan					
St_fdc	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of driving dynamics control units					
St_fgr	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of cruise control					
St_fi_nsobd	O/V	0...FFH	0...255	1	[-]
Container für Bits					
St_gen_ext	O	0...FFH	0...255	1	[-]
Carrierbyte for status external adjustment generator					
St_genallg	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of general alternator input values					
St_generator	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of alternator					
St_genll	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of 'LR' inputs (infos from alternator)					
St_gentester	O/V	0...FFH	0...255	1	[-]
Statusbyte Generatortest					
St_getrdaten	O/V	0...FFH	0...255	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Status Getriebedaten					
St_getriebe	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of gearbox					
St_gglsu2	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of WRAF sensors					
St_igrin	O/V	0...FFH	0...255	1	[-]
Carrierbyte for St_igrin					
St_igrinnv	O/V/S	0...FFH	0...255	1	[-]
Carrierbyte " St_igrinnv"					
St_katueb	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of catalyst monitoring					
St_kl15	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of ignition key					
St_klopfen	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of knock					
St_kr	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of knock control					
St_kr1	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of knock control (1)					
St_ksdiag	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status "Testeranforderung für Fuel-System-Diagnose"					
St_lam_ad_1	O/V	0...FFH	0...255	1	[-]
Status Lambdaadaption Bank1					
St_lam_ad_2	O/V	0...FFH	0...255	1	[-]
Status Lambdaadaption Bank2					
St_lamson	O/V	0...FFH	0...255	1	[-]
Statuswort Lambdasonden					
St_ll	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of idle speed					
St_llr	O/V	0...FFFFH	0...65535	1	[-]
carrierwords for status of idle speed control					
St_loadresp	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of loadresp					
St_lvs_in	O/V	0...FFH	0...255	1	[-]
Statuswort für LVS Eingangsgrößen					
St_lvs_in_nv	O/V/S	0...FFH	0...255	1	[-]
Statuswort für LVS Eingangsgrößen nichtflüchtig					
St_mdbaprio_in	O/V	0...FFH	0...255	1	[-]
Statusbyte Bit-Layereingänge Mdbaprio					
St_mkwchn	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of `master crankshaft¿					
St_motmsa	O/V	0...FFH	0...255	1	[-]
Statuswort Motorzustand					
St_motzustd	O/V	0...FFFFH	0...65535	1	[-]
carrierword for status of engine state					
St_msae	O/V	0...FFFFH	0...65535	1	[-]
carrierword for msae					
St_msae2	O/V	0...FFH	0...255	1	[-]
Statusword msae 2					
St_msaklima	O/V	0...FFH	0...255	1	[-]
carrierbyte MSA-Beeinflussung with KLIMA					
St_msasiko	O/V	0...FFH	0...255	1	[-]
Statuswort MSA-Plausi					
St_mdba_in	O/V	0...FFH	0...255	1	[-]
Statuswort für Eingang MDBA					
St_oz	O/V	0...FFH	0...255	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
carrierbyte for status of OEZS					
St_oz_i	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of OEZS					
St_pmcbsin	O/V	0...FFH	0...255	1	[-]
Carrierbyte for ???					
St_pmi	O/V/S	0...FFH	0...255	1	[-]
carrierbyte for status of inputs for low cost power management					
St_pmi_nv	O/S	0...FFH	0...255	1	[-]
Carrierbyte for status of "Eingänge LCP non volatile"					
St_poel	O/V	0...FFH	0...255	1	[-]
carrierbyte for oil switch					
St_poelreg1	O/V	0...FFH	0...255	1	[-]
Status Öldruckregelung 1					
St_pruef	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of lambda request for plausibility test					
St_pwf	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status of power fail					
St_sawe	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of fuel cut-off / restart fuel feed					
St_sk	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status of HOM_AFS					
St_slp	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of secondary air pump					
St_sls	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of secondary air system					
St_spa2	O/V/S	0...FFH	0...255	1	[-]
Status SPA					
St_start	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status of start					
St_szboost	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status of dwell time prolongation					
St_tank	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status tank					
St_tevanst	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of canister purge control					
St_tevein	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of canister purge valve					
St_ti	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of injection					
St_tp	O	0...FFH	0...255	1	[-]
Statuswort					
St_tum	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status of tum					
St_vbrzyl_ein	O/V	0...FFFFH	0...65535	1	[-]
Status Zylindergleichstellung					
St_wdkdiag	O/V	0...FFH	0...255	1	[-]
Carrierbyte for status "Status 3. Poti"					
St_wm	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of WM-coordinator input					
St_wuerg	O/V	0...FFH	0...255	1	[-]
Statuswort für Abwürgerkennung					
St_zwbts	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of catalyst overheating prevention					
St_zwgemad	O/V	0...FFH	0...255	1	[-]
Statuswort für Bits					
St_zwktibs	O/V	0...FFH	0...255	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
carrierbyte for status of IBS communication					
St_zylab	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of cylinder shut-off					
Status_kwnot	O/V	0...FFH	0...255	1	[-]
carrierbyte for status of crankshaft limp-home					

Input data:

B_abgasklappe	B_abgewuergt	B_agr_ktrl	B_atlsvc_if
B_auss_b1	B_cdigronw	B_cdxenonw	B_crashgen
B_dc_new	B_desu_fett	B_desu_puls	B_dev0detec
B_dev1detec	B_dev2detec	B_dev3detec	B_dev4detec
B_dev5detec	B_dev6detec	B_dev7detec	B_disa_akt
B_dkheiz_ende	B_dkratio_h	B_dscakt	B_dzm_dyn
B_ecojob1	B_ecojob2	B_fagurt	B_fe
B_fetrawedeak	B_fgr_akt	B_fi_nsobd1	B_fi_nsobd2
B_ftauf1	B_gang_rueck	B_gangwechsel_gs	B_gen_ext
B_gentestanf	B_gl_ad	B_gl_adder	B_gl_adz
B_gl_adz_offset1	B_gl_adz_offset2		
B_gl_hz	B_gl_wg1	B_gl_wg2	B_gl_zsk
B_gl_zswz	B_injad_anf	B_jumpstart	B_kath
B_keinhs_gs	B_kupp1	B_lamson_ok1	B_lamson_ok2
B_lamzylact_1	B_lamzylact_2	B_ldm_ena	B_ldm_nofil
B_lklps_kl1	B_lklps_kl2	B_lklps_ln	B_lklps_ln1
B_lrad_akt1	B_lrad_akt2	B_lrad_deakt1	B_lrad_deakt2
B_lurzylact	B_lvs_neustart	B_mhauf1	B_motlaeuft
B_msacanerr	B_msadeakt	B_msadxcav	B_msahfkreset
B_msaklimaav	B_msadltgpd	B_msadltgtd	B_msaklimaea
B_msaled	B_msalederr	B_msastopp_siko	B_msastopt
B_msasw	B_msataster	B_msavadapt	B_msaverh
B_nglern	B_nosa	B_nsadap_anf	B_nsdiag_anf
		B_pmrestore	B_poel
B_poelsoltst	B_rgnkat_hla1	B_safast	B_schalt_dkg
B_schalt_ldm	B_schlok	B_sk_homla1	B_spa_csoll
B_sta	B_start	B_taleer	B_td_og
B_td_ogok	B_teblg	B_testpoelsys	B_toel
B_eol_tev			
B_tra	B_tstkvs	B_warml_zykl	B_we
B_wst	B_zrlvs_clr	B_zwgemad_1	B_zwgemad_2
ECU_STATE	KNK_CTL_DIS	LV_AD_CLR_RON	LV_AFL_CLC
LV_ALTER_BSD_PROT_2	LV_ALTER_ERR_EL	LV_ALTER_ERR_MEC	LV_ALTER_ERR_TEMP
LV_ALTER_IF_ACT	LV_ALTER_RD_TBL_6	LV_AT	LV_BRAKE_DET
LV_CAM_SP_CH	LV_CAM_SP_EX_EXT_ADJ	LV_CAM_SP_IN_EXT_ADJ	LV_CS_CUS
LV_CWP_BLOCK_DEAC	LV_DRI	LV_ERR_BSD	LV_FAN_VAR_AD
LV_GS_ACT	LV_IGK	LV_INH_AFL	LV_INH_DPS_REG_CPSDIAG
LV_INH_LSCL[NC_CBK_EX_NR]	LV_INH_S	LV_KEY_AUX	LV_KNK_CTL_ENA
LV_KNK_TRA_N	LV_LAMB_CH	LV_LOAD_RESP_ALTER_CND_1	LV_LOAD_RESP_ALTER_THD_ACT_SP
LV_LOAD_RESP_ALTER_THD_ACT_SP	LV_LTG_HDLP_L_ON	LV_LTG_INL_ON	LV_MFF_S_POST_CH
LV_N_MON_CWP_DEAC	LV_NT_AFS_REQ	LV_REQ_HEAT	LV_REQ_ISC
LV_RGN_NT_REQ	LV_RLY_ST_CAN	LV_S_CLC	LV_SENS_BAT_SMT_DET
LV_SO2P_ACT	LV_SOF_SWI	LV_TI_CH	LV_TQ_CRU_ACT
LV_TQ_IGA_ACT	LV_TQ_MIN_CLU	NC_CBK_EX_NR	St_aekh
St_ar1	St_atldiag2_out	St_atloutif_out	St_atlstat
St_ba_agf	St_blshub	St_bns	St_deavns2
St_dgen0	St_dibs0	St_disa_bmw	St_dps
St_ecostat2	St_eisy_hfm	St_eisyad_read	St_eisydiag1

St_eisydiag2	St_gl_adapt	St_hhs	St_igroutnv
St_imdgen0	St_kath_ena	St_ldstgen	St_llranh0
St_mdar0	St_mdar1	St_mdbafak	St_mdbanl2
St_mdbanl_pwgll		St_mdk	St_mini
St_msaa	St_msaanz	St_msabs	St_ngang0
St_nmax_ba	St_oz_o	St_ozbg0	St_pmdiag0
St_pmdiag1	St_pmprsvcalc1	St_poelreg2	St_pr
St_prestoppmsa	St_qvc1	St_rail	St_sa
St_sleep	St_sondenheiz	St_statmkor1	St_ti_b
St_tiausgang	St_ugenkor0	St_vbrzyl_aus	St_vsa
St_vse	St_wm1	St_zwbts1	St_zwgrund
Stat_sv_reg1	Stat_sv_reg2	STATE_CH	STATE_CLU_AMT
STATE_ETC_LIH	STATE_TEMP_GB	STATE_TPS_DIAG	Stpcos

36.15.1 Carrierbytes St_.. which are defined as SV outputs

FUNCTION DESCRIPTION:

General information:

Adaptation to BMW environment

Application conditions:

Initialisation at reset or at exit ECU_STATE "PWL":

all variables St_.. are initialised with 0 except:

St_abgasklappe = 1
St_blsdisa1 = last calculated value
St_devxdetec = last valid value
St_dk_diag = first valid value
St_dkheiz = 1
St_elu = first value
St_elu(0) = last stored value
St_fdc = first valid value
St_fdc(0) = first measured value
St_getriebe = first valid value
St_getriebe(0) = first measured value

```
If      LV_VAR_BN = 0
then    St_kl15(0) = 1
else    St_kl15(0) = 0
endif
```

St_pmi_nv(0) = last stored value
St_pmi_nv(1) = last stored value
St_pmi_nv = last value from nonvolatile memory
St_spa2 = last value from nonvolatile memory
St_lvs_in_nv = last value from nonvolatile memory

Recurrence:

seg:	St_zylab, (if LV_ST_END = 1), St_szboost (if LV_ES = 0) Status_kwnot, St_ti
5ms:	St_dk_diag, St_klopfen, St_dkratio
10 ms:	St_zylab. (if LV_ST_END = 0), St_szboost (if LV_ES = 1) St_anman1, St_ba, St_bgkuppl, St_btvns, St_clc, St_dtev, St_egs, St_ekp, St_fdc, St_fgr, St_generator, St_getriebe, St_katueb, St_kr, St_kr1, St_ll, **St_llr (wird beim ersten LL-Regler ein gesetzt und nicht mehr geändert)**, St_loadresp, St_motzustd, St_slp, St_sls, St_start, St_tevein, St_tevanst, St_pmi, St_pmi_nv, St_agr, St_sk, St_anf_prail, St_cybl, St_blsdisa1 , St_poelreg1, St_msae, St_dkg, St_auss, St_lamson, St_zwgemad, St_abgasklappe, St_dyndzm, St_getrdaten, St_spa2, St_fi_nsobd, St_motmsa, St_wuerg, St_msae2, St_mdba_in, St_msasiko, St_mdbaprio_in
20 ms:	St_desul, St_kl15, St_sawe, St_zwktibs (via BSD call)
100 ms:	St_atldiag2_in, St_bls, St_ckl15, St_elu, St_genallg, St_genll, St_gglsu2, St_gen_ext, St_tank, St_ksdiag, St_pruef, St_tum, St_pmi_nv, St_tum, St_oz_i, St_entlast_loc, St_igrin, St_igrinnv, St_poel, St_gentester, St_ecostat1, St_dkheiz, St_vbrcyl_ein, St_msaklima, St_lam_ad_1, St_lam_ad_2, St_lvs_in, St_lvs_in_nv, St_cybl2
200 ms:	St_zwbts, HKWBits2_1000ms, St_wm, St_tp,
1000 ms:	St_oz
Once at ECU init:	St_devxdetec, St_pwf
BSD:	St_zwktibs

Activation: after reset

During power latch phase: all variables are zero, except:

St_atldiag2_in = last calculated value
St_abgasklappe = last calculated value
St_blsdisa1 = last calculated value
St_ckl15 is calculated in the PWL phase
St_ckl15(0) = received value
St_ckl15(1) = received value

St_devxdetec = last valid value
St_dk_diag = last calculated value
St_dkheiz = last calculated value
St_ecostat1 = last calculated value
St_egs = last value
St_egs(0) = last valid value
St_egs(1) = last valid value
St_egs(2) = last valid value
St_ekp(3) = last valid value
St_elu = last calculated value
St_fdc = last valid value
St_fdc(0) = last measured value
St_genallg = last value
St_genallg(0) = last value
St_genallg(2) = last value
St_genallg(3) = last value
St_genallg(4) = last value
St_generator = last calculated value
St_generator(0) = last valid value
St_generator(2) = last valid value
St_generator(3) = last valid value
St_generator(4) = last valid value
St_generator(5) = last valid value
St_generator(6) = last calculated value
St_genll = last value
St_genll(0) = last value
St_genll(2) = last value
St_getriebe = last valid value
St_getriebe(0) = last valid value
St_gglsu2(0)= 1
St_gglsu2(1)= 1
St_katueb(0)= 1
St_katueb(1)= 1
St_loadresp = last value
St_loadresp(0) = last value
St_mdba_in = last calculated value
St_msae = last value
St_oz(0) = last calculated value
St_oz(1) = last calculated value
St_oz_i(0) = last value
St_oz_i(1) = last value
St_oz_i(4) = last calculated value
St_pmi(0) is calculated in the PWL phase
St_pmi(1) is calculated in the PWL phase
St_pmi(2) is calculated in the PWL phase
St_pmi(3) is calculated in the PWL phase
St_pmi(4) is calculated in the PWL phase
St_poel = last value
St_slp(0) = last valid value
St_slp(2) = last valid value
St_sls(0) = last value

St_start = last calculated value
St_zylab(0) is calculated in the PWL phase
St_zylab(1) is calculated in the PWL phase
St_zwktibs(0) is calculated in the PWL phase
St_tum(0) = last calculated value
St_pmi_nv is calculated in the PWL phase
St_wm = last value
St_tum = last calculated value
St_gen_ext is calculated in the PWL phase
St_sk = last value
St_tank = last calculated value
St_wdkdiag = last calculated value
St_igrin is calculated in the PWL phase
St_gentester= last calculated value
St_zwgemad = last calculated value
St_szboost is calculated in the PWL phase
St_vbrcyl_ein = last calculated value
St_msa_klima = last calculated value
St_spa2 = last calculated value
St_fi_nsobd = last calculated value
St_wuerg = last calculated value
St_lvs_in = last calculated value
St_msae2 = last calculated value
St_mdbaprio_in = last calculated value
St_lvs_in_nv= last calculated value
St_cybl2 = last calculated value

Deactivation: after power latch phase

Formula section:

As the applied bitmasks are automatically generated by a script file, there is no separate arrangement stated in this specification.

36.15.2 Carrierbytes B_.. which are defined as BMW outputs

FUNCTION DESCRIPTION:

General information:

Adaptation to BMW environment

Application conditions:

Initialisation at reset or at exit power latch phase:

All values B_.. are initialised with 0, except for:

B_bns = last stored value
B_dash = first measured value
B_lsd = first measured value
B_vsa_an = last calculated value
B_vse_an = last calculated value
B_lklps = last valid value
B_lklps1 = last valid value
B_disa1_auf = 1
B_disa2_auf = 1
B_maregdk_ad = last calculated value
B_mrmslam = first valid value
B_diagigr = last stored value
B_ecobusy = from nonvoltile memory
B_spa_cist from nonvoltile memory
B_nmax_ba_cc = 1
B_psr_ad = last calculated value
B_psragr_ad= last calculated value

Recurrence:

10 ms:	B_ar_akt, B_bns, B_dash, B_disa_stopp, B_esp_h1, B_esp_h2, B_espr_start, B_prailsw, B_hd_start, B_hmm_akt, B_hom_akt, B_hstoech, B_kupp, B_la_stopp1, B_la_stopp2, B_lroff, B_lsd, B_nesps_akt, B_nesps_ena, B_prsoll_h, B_sch_akt, B_schicht, B_temxon, B_tev_stopp, B_ulv, B_verb_h, B_verb_s, B_vsa_an, B_vsaadp, B_vse_an, B_vseadp, B_zw_dynman, B_zwdyn, St_fup_bmw, B_gdst, B_esp_hs1, B_esp_hs2, B_hschalt_komb, B_ldm_akt04, B_ldm_off, B_rschalt_komb, B_prailsw, B_disa1_auf , B_disa2_auf, B_msaakt, B_msaaktber, B_msaerr, B_msastart, B_msastartf, B_msastopp, B_msastops, B_gangnull, B_poel_gelb, B_poel_rot, B_vlschalt, B_mdimin_sa, B_zwmin_ext, B_msaprestopp, B_spa_cist, B_msastarts, B_nmax_ba_cc, B_notafu,B_msaccid2, B_vbkhm_notl, B_msaanz1, B_msaanz2, B_nggelernt, B_ngimlf
20 ms:	B_wdkdiag_ugd
100 ms:	B_atlsvc_khm, ccbtnach, B_ccbttdfk, B_ccbttknt, B_ccpmerr, B_ccprio, B_ccruhver, B_codierpm, B_diagigr, B_genoff, B_kl61, B_pmbsdanf, B_pmvsdanf, B_ums_vbsa, B_verb_off, B_maregdk_ad, B_ecobusy, B_lradap1_mn, B_lradap1_mx, B_lradap2_mn, B_lradap2_mx, B_ext_zylstop, B_psr_ad , B_psragr_ad, B_hlsh1_off, B_hlsh2_off,B_hlsu1_off, B_hlsu2_off, B_hnox_off, B_atlsvc
200 ms:	B_bs1, B_bs2, B_ewp_swsp, B_lklps, B_lklps1,B_ntlkws, B_tmmi_warn1, B_tmmi_warn2, B_tmmi_warn3, B_vb_sa_nk

1000 ms: B_bosmsgc, B_mol_sf, B_mol_vb, B_mol_vl, B_nsub, B_oeltemp, B_oelverb, B_oelverl, B_oztemic, B_vb_sa, B_qvch2o

by event: B_sleepwait

seg: B_mrmslam, B_wese2h_abs, B_aekh, B_prailpulskor_ee, B_prailpulskor_me, B_rkkor_kva

ini2: B_vsean_loc

Activation: after reset

Deactivation: During power latch phase: all variables B_.. are 0, except for:
B_atlsvc = last calculated value
B_atlsvc_khm = last calculated value
B_bns is calculated in the PWL phase
B_bosmsgc = last calculated value
B_ccbtnach is calculated in the PWL phase
B_ccbttdfk is calculated in the PWL phase
B_ccbttknt is calculated in the PWL phase
B_ccpmerr is calculated in the PWL phase
B_ccprio is calculated in the PWL phase
B_ccruhver is calculated in the PWL phase
B_codierpm is calculated in the PWL phase
B_dash = last valid value
B_ext_zylstop= last valid value
B_ewp_swsp= last value
B_genoff = last valid value
B_kl61 = last calculated value
B_lklps is calculated in the PWL phase
B_lklps1 is calculated in the PWL phase
B_lroff = last value
B_lsd = last valid value
B_mol_sf = last calculated value
B_mol_vb = last calculated value
B_mol_vl = last calculated value
B_msaakt = last valid value
B_msaaktber= last valid value
B_msaerr = last valid value
B_msastart = last valid value
B_msastartf = last valid value
B_msastopp = last valid value
B_nggelernt = last calculated value
B_ngimlf = last calculated value
B_nsub = last calculated value
B_ntlkws = last calculated value
B_oeltemp = last calculated value
B_oelverb = last calculated value
B_oelverl = last calculated value
B_oztemic = last calculated value
B_pmbsdanf is calculated in the PWL phase
B_pmvsdanf is calculated in the PWL phase

B_sleepwait is calculated in the PWL phase
B_verb_off = last calculated value
B_vsa_an = last calculated value
B_vsaadp = last calculated value
B_vse_an = last calculated value
B_vseadp = last calculated value
B_disa1_auf = last calculated value
B_disa2_auf = last calculated value
B_maregdk_ad = last calculated value
B_mrmslam = last valid value
B_poelanz = last calculated value
B_diagigr is calculated in the PWL phase
B_gangnull = last calculated value
B_poel_gelb = last calculated value
B_poel_rot = last calculated value
B_vlschalt = last calculated value
B_qvch2o = last calculated value
B_spa_cist = last calculated value
B_lradap1_mn = last calculated value
B_lradap1_mx = last calculated value
B_lradap2_mn = last calculated value
B_lradap2_mx = last calculated value
B_msastarts = last valid value
B_nmaxc_ba_cc = last calculated value
B_psr_ad = last calculated value
B_psragr_ad = last calculated value
B_hlsh1_off = last valid value
B_hlsh2_off = last valid value
B_hlsu1_off = last valid value
B_hlsu2_off = last valid value
B_hnox_off = last valid value
B_msaccid2= last calculated value
B_vbkhm_notl= last calculated value
B_prailpulskor_ee = last calculated value
B_prailpulskor_me = last calculated value
B_rkkor_kva = last calculated value
B_msastops = last valid value

Formula section:

As the applied bitmasks are automatically generated by a script file, there is no separate arrangement stated in this specification.

37 Heating, Ventilation and Air Conditioning

Table of Contents

Index

37.1 HVAC - Requirements to Infrastructure interface

Export actions:

ACTION_INFR_SetACCOUT(IN <lv_accout_rly>**)**
This action sets the output pin for the "ACCOUT Switch"
ACTION_INFR_SetCRCVHEAT(IN <lv_rly_crcv_heat>**)**
This action sets the output pin for the "CRCV_HEAT Switch"
ACTION_INFR_GetElDiagACCOUT(OUT < Cdn_diag_rly_accout >, **OUT** < Err_diag_rly_accout >**)**
This action reads the failure and condition information for a symptom of the accout switch.
ACTION_INFR_GetElDiagCRCVHEAT(OUT<Cdn_diag_rly_crcv_heat >, **OUT**<Err_diag_rly_crcv_heat>**)**
This action reads the failure and condition information for a symptom of the crcv_heat switch.

Description for actions:

ACTION_INFR_SetACCOUT(IN < lv_accout_rly >**)**					
This action sets the output pin for the "ACCOUT Switch"					
Name	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
lv_accout_rly	IN	0H 1H	ACC-relay deactivated ACC-relay activated	1	-
Set ACC-relay					

ACTION_INFR_SetCRCVHEAT(IN < lv_rly_crcv_heat >**)**					
This action sets the output pin for the "CRCV_HEAT Switch"					
Name	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
lv_rly_crcv_heat	IN	0H 1H	CRCV-HEAT relay deactivated CRCV-HEAT relay activated	1	-
Set CRCV-HEAT					

ACTION_INFR_GetElDiagACCOUT(OUT <Cdn_diag_rly_accout >, **OUT** < Err_diag_rly_accout >**)**					
This action reads the failure and condition information for a symptom of the accout switch. When calling this action the information inside the infrastructure is reset after returning the OUT parameters.					
Name	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cdn_diag_rly_accout	OUT	0...7H	0...7	1	-
Diagnosis condition for symptoms of ACCOUT: bit 0: Set, if diagnosis condition for symptom SCP (SYM_0) is fulfilled bit 1: Set, if diagnosis condition for symptom SCG (SYM_1) is fulfilled bit 2: Set, if diagnosis condition for symptom OC (SYM_2) is fulfilled (all combinations are possible)					
Err_diag_rly_accout	OUT	0...7H	0...7	1	-
Raw value of error symptom: bit 0: Set, if raw value of error symptom SCP (SYM_0) is set bit 1: Set, if raw value of error symptom SCG (SYM_1) is set bit 2: Set, if raw value of error symptom OC (SYM_2) is set (all combinations are possible)					
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

<table>
<tr><td colspan="6">ACTION_INFR_GetElDiagCRCVHEAT(OUT<Cdn_diag_rly_crcv_heat >, OUT<Err_diag_rly_crcv_heat>)</td></tr>
<tr><td colspan="6">This action reads the failure and condition information for a symptom of the crcv_heat switch.</td></tr>
<tr><th>Name</th><th>Type</th><th>Hex. Limits</th><th>Phys. Limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>Cdn_diag_rly_crcv_heat</td><td>OUT</td><td>0...7H</td><td>0...7</td><td>1</td><td>-</td></tr>
<tr><td colspan="6">Diagnosis condition for symptoms of ACCOUT:
bit 0: Set, if diagnosis condition for symptom SCP (SYM_0) is fulfilled
bit 1: Set, if diagnosis condition for symptom SCG (SYM_1) is fulfilled
bit 2: Set, if diagnosis condition for symptom OC (SYM_2) is fulfilled
(all combinations are possible)</td></tr>
<tr><td>Err_diag_rly_crcv_heat</td><td>OUT</td><td>0...7H</td><td>0...7</td><td>1</td><td>-</td></tr>
<tr><td colspan="6">Raw value of error symptom:
bit 0: Set, if raw value of error symptom SCP (SYM_0) is set
bit 1: Set, if raw value of error symptom SCG (SYM_1) is set
bit 2: Set, if raw value of error symptom OC (SYM_2) is set
(all combinations are possible)</td></tr>
<tr><td colspan="6">Raw value of error symptom.
The relevant bit is set, if the error has been detected.</td></tr>
</table>

FUNCTION DESCRIPTION:

General information:

ACTION_INFR_GetElDiagACCOUT()

ACTION_INFR_GetElDiagACCOUT() returns the result of the electrical diagnosis of the ACCOUT Switch.
The device readout is performed autonomous by the infrastructure each 10ms.
The error informations of every symptom is gathered in the infrastructure (or-ed symptom) until the application reads out the information by calling ACTION_INFR_GetElDiagACCOUT().
After having read out the information by calling ACTION_INFR_GetElDiagACCOUT(), the data inside the infrastructure is reset. Resetting of Cdn_diag_rly_accout avoids ambiguous results in case of too short calling recurrence of ACTION_INFR_GetElDiagACCOUT(). Reset Cdn_diag_rly_accout indicates that the gathering of the information is not completely finished.

ACTION_INFR_GetElDiagCRCVHEAT()

ACTION_INFR_GetElDiagCRCVHEAT() returns the result of the electrical diagnosis of the CRCV_HEAT Switch.
The device readout is performed autonomous by the infrastructure each 10ms.
The error informations of every symptom is gathered in the infrastructure (or-ed symptom) until the application reads out the information by calling ACTION_INFR_GetElDiagCRCVHEAT ().
After having read out the information by calling ACTION_INFR_GetElDiagCRCVHEAT (), the data inside the infrastructure is reset. Resetting of Cdn_diag_rly_crcv_heat avoids ambiguous results in case of too short calling recurrence of ACTION_INFR_GetElDiagCRCVHEAT (). Reset Cdn_diag_rly_crcv_heat indicates that the gathering of the information is not completely finished.

Requirements for ACTION_INFR_SetACCOUT:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
lv_accout_rly	Not relevant	Not relevant	1	Not relevant	Not relevant

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_SetCRCVHEAT:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
lv_rly_crcv_heat	Not relevant	Not relevant	1	Not relevant	Not relevant

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_GetElDiagACCOUT:

Data acquisition					
Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment
Cdn_diag_rly_accout	-	-	<bit coded>	Cdn_diag_rly_accout	Diagnosis condition for each symptom bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
Err_diag_rly_accout	-	-	<bit coded>	Err_diag_rly_accout	Bitcoded result of each symptom bit 0: raw value of error symptom SCP (SYM_0) bit 1: raw value of error symptom SCG (SYM_1) bit 2: raw value of error symptom OC (SYM_2) The relevant bit is set, if the raw error has been detected.

Diagnosis: ACTION_INFR_GetElDiagACCOUT returns the electric diagnosis for the ACCOUT switch.

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_GetElDiagCRCVHEAT:

Data acquisition					
Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment
Cdn_diag_rly_crcv_heat	-	-	<bit coded>	Cdn_diag_rly_crcv_heat	Diagnosis condition for each symptom bit 0: diagnosis condition for symptom SCP (SYM_0) bit 1: diagnosis condition for symptom SCG (SYM_1) bit 2: diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
Err_diag_rly_crcv_heat	-	-	<bit coded>	Err_diag_rly_crcv_heat	Bitcoded result of each symptom bit 0: raw value of error symptom SCP (SYM_0) bit 1: raw value of error symptom SCG (SYM_1) bit 2: raw value of error symptom OC (SYM_2) The relevant bit is set, if the raw error has been detected.

Diagnosis: ACTION_INFR_GetElDiagCRCVHEAT returns the electric diagnosis for the CRCV_HEAT switch.

Coincidence requirements: no coincidence requirements to other events

37.2 AGGR HVAC adaptation

37.2.1 Outputs for BMW functions which are defined as HVAC exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Mdi_res_ac	O/V	D8F0...2710H	-1000...1000	0.1	Nm
torque reserve for switching on air conditioning compressor					
Mdi_res_heiz	O/V	D8F0...2710H	-1000...1000	0.1	Nm
Torque reserve for switching on additional heating by ACC					
B_korel	O/V	0...1H	0...1	1	-
condition air compressor					
Md_na_ac	O/V	8000H...7FFFH	-3276.8 ...3276.7	0.1	Nm
torque losses air conditioning compressor					

Input data:

TQ_ADD_ACC	TQ_ADD_HEAT_ACC	LV_ACCIN	ECU_STATE
TQ_LOSS_ACC			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: 0 at reset

Recurrence : 10 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

Mdi_res_ac = TQ_ADD_ACC

Mdi_res_heiz = TQ_ADD_HEAT_ACC

```
if      ECU_STATE = 4Hex     "power latch"
then    B_korel = 0
        Md_na_ac  =  0
else    B_korel = LV_ACCIN
        Md_na_ac  =  TQ_LOSS_ACC
endif
```

37.3 Blocking of the air conditioning compressor

37.3.1 Enabling the activation of the ACC-relay

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ACCOUT_RLY_ENA	O/V	0...1H	0...1	1	[-]
LV indicating that the conditions for activating the ACC-relay are fulfilled					

Input data:

LV_ES	LV_ST	VS	PV_AV
LV_ACCIN	T_AST	LV_ERR_BN_ACC	LV_ERR_TOUT_ICL_3
LV_ERR_CAN_BOFF			

FUNCTION DESCRIPTION:

General information:

The activation of the ACC-relay is controlled by the logical value LV_ACCOUT_RLY which is handled by this module. The A/C-ECU requests to switch the ACC-relay by LV_ACCIN. Under certain conditions the activation will be suppressed (LV_ACCOUT_RLY_ENA = 0). These conditions are
- engine stopped or start
- during time after transition from start
- during time after detecting high torque request when vehicle speed is low
- during time after last ACC-relay switch off
- CAN-Timeout (CAN11H) INSTR3-message: LV_ERR_TOUT_ICL_3
- CAN-Timeout (BN2000) IHKA-message: LV_ERR_BN_ACC
- CAN-Busoff

Activation : at every engine operating state

Deactivation : -

Initialization : LV_ACCOUT_RLY_ENA = 0 at reset

Update rate : 10 ms

Formula section:

a) If LV_ES = 1 or LV_ST = 1, then LV_ACCOUT_RLY_ENA = 0

b) If T_AST $\leq$ C_ACCIN_DLY_4 then LV_ACCOUT_RLY_ENA = 0

c) **If** PV_AV becomes $\geq$ C_PVS_FL_ACCIN when VS $\leq$ C_VS_MAX_ACCIN

then LV_ACCOUT_RLY_ENA = 0 for the time C_ACCIN_DLY_1 **and**

a timer (C_ACCIN_DLY_1_LOCK_FL) is started;

a renewed triggering of ACC-suppression due to condition c) is not possible:

- during the time C_ACCIN_DLY_1_LOCK_FL

- if time C_ACCIN_DLY_1_LOCK_FL has passed but the state FL has not been exited since last triggering due to c)

d) If time after last ACC-relay switch off ≤ C_ACCIN_DLY_5
then LV_ACCOUT_RLY_ENA = 0

This condition d) is not valid, if a blocking due to condition c) ends.

h) If LV_ERR_BN_ACC = 1 then LV_ACCOUT_RLY_ENA = 0

i) If LV_ERR_TOUT_ICL_3 = 1 then LV_ACCOUT_RLY_ENA = 0

j) If LV_ERR_CAN_BOFF = 1 then LV_ACCOUT_RLY_ENA = 0

In all other cases : LV_ACCOUT_RLY_ENA = LV_ACCIN

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_MAX_ACCIN	1	0...FFH	0...255	1	[km/h]
Speed condition for ACC suppression at high load					
C_ACCIN_DLY_1	1	1...FFFFH	0.01...655.35	0.01	[s]
Delay for ACC suppression in transition after full load					
C_ACCIN_DLY_1_LOCK_FL	1	1...FFFFH	0.01...655.35	0.01	[s]
Blocking time for renewed triggering of the ACC suppression at high load					
C_ACCIN_DLY_4	1	1...FFFFH	0.1...6553.5	0.1	[s]
Delay in transition from start					
C_PVS_FL_ACCIN	1	0...FFH	0...99.60937	0.390625	[%]
PV-threshold for ACC suppression					
C_ACCIN_DLY_5	1	1...FFFFH	0.01...655.35	0.01	[s]
Delay before renewed reactivation of the ACC					

37.3.2 Turn-On-/Turn-Off-Delay ACC-relay

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ACCOUT_RLY	O/V	0...1H	0...1	1	[-]
Air conditioning compressor clutch signal is enable					
LV_TOUT_ACC	O/V	0...1H	0...1	1	[-]
logical variable to start deactivation of torque reserve and condition for switch off TQ_LOSS_ACC calculation					
T_ACC	O/V	0...FFFFH	0...655.35	0.01	[s]
Timer for duration of transient TQ_LOSS_ACC correction					
T_ACC_DLY	O/V	0...FFH	0...2.55	0.01	[s]
delay time for start of TQ_LOSS_ACC-calculation & de-/activation of the ACC-relay					

Input data:

LV_ACCOUT_RLY_ENA	LV_SUPP_ACCIN_DIAGCPS	LV_ACCOUT_RLY_EXT_ADJ
LV_ACT_ACCOUT_RLY_EXT_ADJ	N_32	LV_VAR_RLY_ACCOUT

Import actions:

ACTION_INFR_SetACCOUT(IN <lv_accout_rly>**)**
This action sets the output pin for the "ACCOUT Switch"

Note: The imported actions are defined in the HVAC – IRS (infrastructure requirement specification).

FUNCTION DESCRIPTION:

General information:

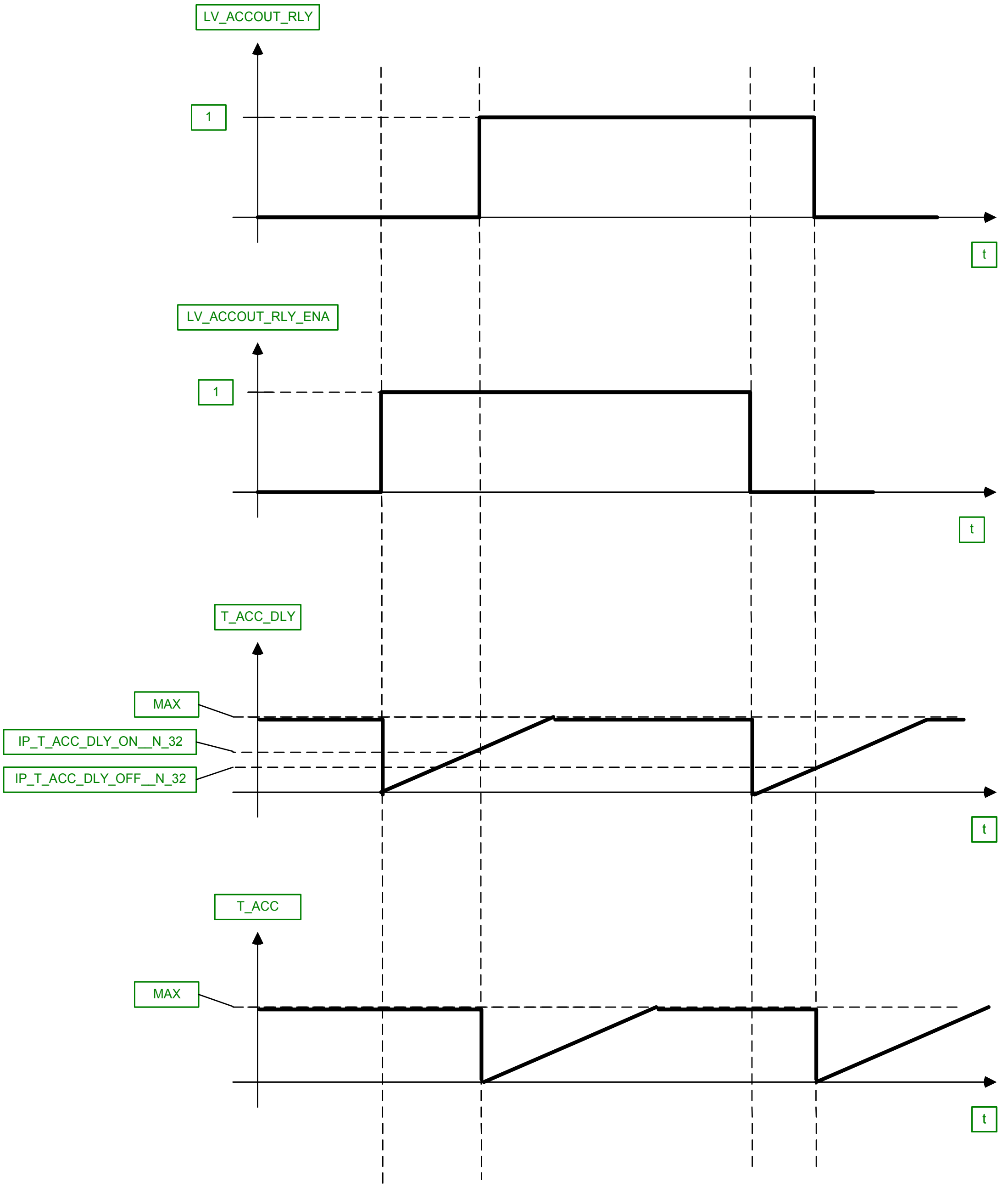
If the enabling conditions for activating the ACC-relay are fulfilled (LV_ACCOUT_RLY_ENA = 0→1) then the ACC-relay is activated (LV_ACCOUT_RLY = 0→1) via a turn-on-delay. The same way the ACC-relay is deactivated (LV_ACCOUT_RLY = 1→0) via a turn-off-delay if LV_ACCOUT_RLY_ENA = 1→0.

If LV_SUPP_ACCIN_DIAGCPS = 1 due to functional check CPS then the state of LV_ACCOUT_RLY is freezed.

For test purpose at assembling line or workshop the ACC-relay can be switched by a serial communication tool, if available (external adjustment). These switchings are done without the turn-on-delay, so the engine speed will show a clear reaction.

Activation :	at every engine operating state
Deactivation :	-
Initialization :	LV_ACCOUT_RLY = 0 at reset LV_TOUT_ACC = 1 T_ACC = FFFFH T_ACC_DLY = FFH
Update rate :	10 ms

Signal flow diagram:



Valid, if external adjustment is not active

Formula section:

Timer T_ACC_DLY :

```
IF        LV_ACCOUT_RLY_ENA $_{N}$ <> LV_ACCOUT_RLY_ENA $_{N-1}$
THEN      T_ACC_DLY = 0
ELSE      IF     T_ACC_DLY < FFH
          THEN   T_ACC_DLY = T_ACC_DLY + 10 ms
          ENDIF
ENDIF
```

Turn-on-delay:

```
IF        LV_ACCOUT_RLY_ENA = 1          AND
          T_ACC_DLY = IP_T_ACC_DLY_ON__N_32
THEN      LV_ACCOUT_RLY = 1
ENDIF
```

Turn-off-delay:

```
IF        LV_ACCOUT_RLY_ENA = 0          AND
          T_ACC_DLY = IP_T_ACC_DLY_OFF__N_32
THEN      LV_ACCOUT_RLY = 0
ENDIF
```

T_ACC for ramping torque loss:

```
IF        LV_ACCOUT_RLY$_{N}$ <> LV_ACCOUT_RLY $_{N-1}$
THEN      T_ACC = 0
          LV_TOUT_ACC =0
ELSE      IF     T_ACC < FFFFH
          THEN   T_ACC = T_ACC + 10ms
          ENDIF
          IF     T_ACC < C_T_TOUT_ACC
          THEN   LV_TOUT_ACC = 0
          ELSE   LV_TOUT_ACC = 1
          ENDIF
ENDIF
```

Freezing the state of LV_ACCOUT_RLY due to activation of DIAGCPS:

```
If        LV_SUPP_ACCIN_DIAGCPS =  1,
  then    the status of   LV_ACCOUT_RLY  is freezed
          until    LV_SUPP_ACCIN_DIAGCPS =  0
```

LV_ACCOUT_RLY during external adjustment:

```
If        LV_ACCOUT_RLY_EXT_ADJ = 1
  then    LV_ACCOUT_RLY  =  LV_ACT_ACCOUT_RLY_EXT_ADJ
  else    LV_ACCOUT_RLY determined as described before   (external adjustment passive)
endif
```

```
If LV_VAR_RLY_ACCOUT = 1
  then
  After LV_ACCOUT_RLY calculation,
  ACTION_INFR_SetACCOUT(IN < LV_ACCOUT_RLY >)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_T_ACC_DLY_ON__N_32	3	0...FFH	0...2.55	0.01	[s]
LDPM_N_32__T_ACC	3	0...FFH	0...8160	32	[rpm]
Turn-On-Delay for start of T_ACC depending on N_32					
IP_T_ACC_DLY_OFF__N_32	3	0...FFH	0...2.55	0.01	[s]
LDPM_N_32__T_ACC	3	0...FFH	0...8160	32	[rpm]
Turn-Off-Delay for start of T_ACC depending on N_32					
C_T_TOUT_ACC	1	0...FFFFH	0...655.35	0.01	[s]
calibration constant to start deactivation of ACC torque reserve					

37.4 Crankcase ventilation heater relay control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_RLY_CRCV_HEAT	V/O	0...1H	0...1	1	[-]
Crankcase ventilation heater relay (1:activated, 0:deactivated)					

Input data:

LV_RLY_CRCV_HEAT_EXT_ADJ	LV_ACT_RLY_CRCV_HEAT_EXT_ADJ	LV_ERR_RLY_CRCV_HEAT	LV_IGK
T_AST	TIA		

Import actions:

ACTION_INFR_SetCRCVHEAT(IN <lv_rly_crcv_heat>**)**
This action sets the output pin for the “CRCV_HEAT Switch”

Note: The imported actions are defined in the HVAC – IRS (infrastructure requirement specification).

General information:

To prevent crankcase ventilation from fouling under cold conditions it is necessary to heat the ventilation rail. For the power stage of this heating an external relay plus heating foil is introduced. The control stage is realised by the engine management system using the below statemachine.

Application conditions:

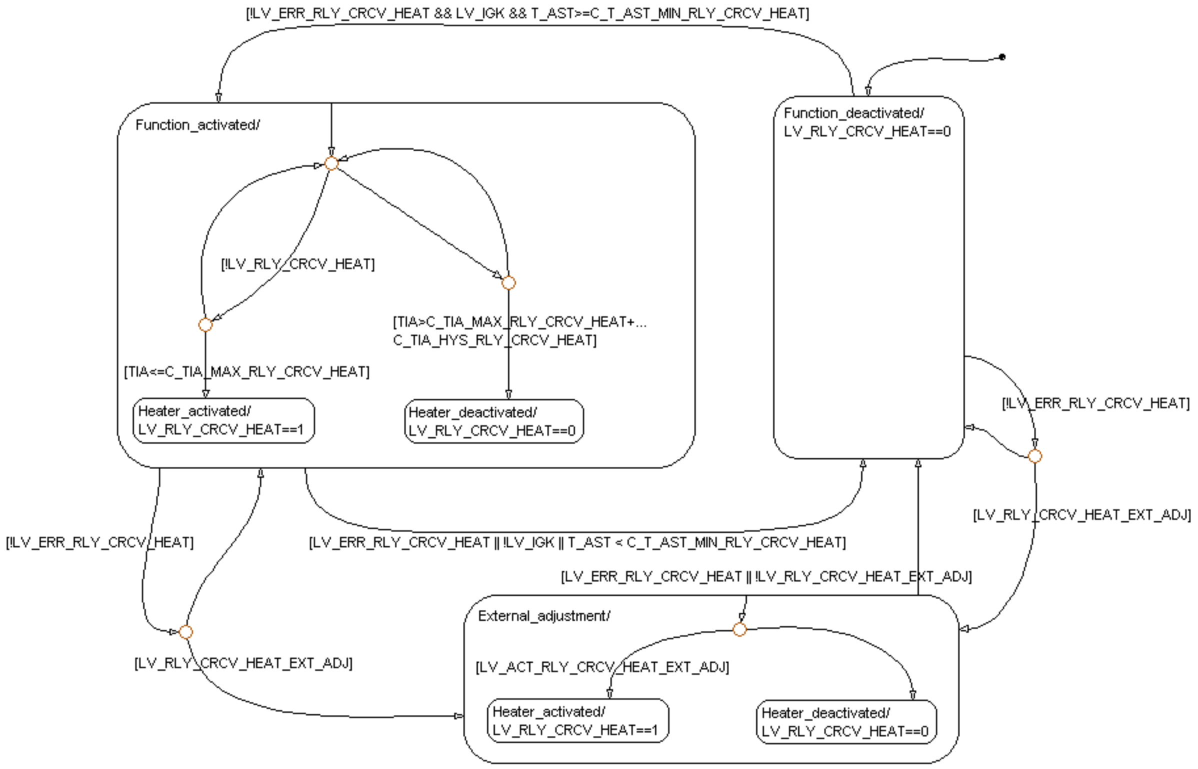
Initialisation: at reset and deactivation all var. are initialised with 0

Recurrence: 1000ms

Activation: *see signal flow diagram*

Deactivation: *see signal flow diagram*

Signal flow diagram:



Formula section:

(redundant information to signal flow diagram – only for a better understanding)

External adjustment:

If(1) LV_ERR_RLY_CRCV_HEAT = 0

Then(1) **If(2)** LV_RLY_CRCV_HEAT_EXT_ADJ = 1

Then(2)

If(3) LV_ACT_RLY_CRCV_HEAT_EXT_ADJ = 1

Then(3) LV_RLY_CRCV_HEAT = 1

Else(3) LV_RLY_CRCV_HEAT = 0

Else(2) either stay in the actual operation mode or (if external adjustment was activated before) firstly deactivate function and check if it has to be reactivated again

Else(1) Function is deactivated

ACTION_INFR_SetCRCVHEAT(IN < LV_RLY_CRCV_HEAT >**)**

"Normal" operation:

```
If(1)       LV_ERR_RLY_CRCV_HEAT = 0      and
            LV_IGK = 1                    and
            T_AST >= C_T_AST_MIN_RLY_CRCV_HEAT
Then(1)
    If(2)       LV_RLY_CRCV_HEAT = 0
    Then(2)
        If(3)       TIA <= C_TIA_MAX_RLY_CRCV_HEAT
        Then(3)     LV_RLY_CRCV_HEAT = 1
    Else(2)
        If(4)       TIA > C_TIA_MAX_RLY_CRCV_HEAT +
                              C_TIA_HYS_RLY_CRCV_HEAT
        Then(4)     LV_RLY_CRCV_HEAT = 0
Else(1)     Function is deactivated
ACTION_INFR_SetCRCVHEAT(IN < LV_RLY_CRCV_HEAT >)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_AST_MIN_RLY_CRCV_HEAT	1	0...FFFFH	0...6553.5	0.1	[s]
Minimal time after start for CRCV_HEAT activation (default: 60s)					
C_TIA_MAX_RLY_CRCV_HEAT	1	0...FEH	-48...142.5	0.75	[°C]
Maximal intake air temperature for CRCV_HEAT activation (default: 0°C)					
C_TIA_HYS_RLY_CRCV_HEAT	1	0...FEH	-48...142.5	0.75	[°C]
intake air temperature hysteresis between activation and deactivation of CRCV_HEAT (default: 3°C)					

37.5 Torque Loss and Reserve for A/C Compressor

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
FAC_SP_N_PSTE	O/V	0... FFH	0... 1.9921875	7.8125e-3	[-]
Factor of engine speed relative to setpoint					
LV_REQ_HEAT_ACC_ENA	O/V	0... 1H	0... 1	1	[-]
LV indicating that the conditions for additional heating are fulfilled					
LV_TOUT_REQ_HEAT_ACC	O/V	0... 1H	0... 1	1	[-]
logical variable to start deactivation of torque reserve TQ_ADD_HEAT_ACC					
T_REQ_HEAT_ACC	O/V	0... FFFFFFFFH	0... 42949672.95	0.01	[s]
Timer for duration of torque reserve TQ_ADD_HEAT_ACC calculation					
TQ_ADD_ACC	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for switching on air conditioning compressor					
TQ_ADD_HEAT_ACC	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for switching on additional heating by ACC					
TQ_LOSS_ACC	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque losses air conditioning compressor					
TQ_LOSS_ACC_INP	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
ACC torque loss due to engine-speed and ACC torque demand					

Input Data:

LV_ACCOUT_RLY	LV_ACCOUT_RLY_ENA	LV_REQ_HEAT	LV_TOUT_ACC
N_32	TAM	TCO	TQ_ACCIN_CAN
TQ_DIF_IS_AD_ACC	T_ACC	N	N_SP_IS

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CRLC_TQ_LOSS_ACC_INP	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlation Factor for tQ_ACCIN_CAN calculation					
C_N_SP_IS_PSTE	1	0... FFH	0... 8160	32	[rpm]
Idle speed setpoint power steering, only for FAC_SP_N_PSTE calculation					
C_T_REQ_HEAT_ACC	1	0... FFFFFFFFH	0... 42949672.95	0.01	[s]
time for duration of torque reserve TQ_ADD_HEAT_ACC calculation					
C_TAM_MIN_REQ_HEAT_ACC	1	0... FEH	-48... 142.5	0.75	[°C]
TAM threshold for enabling additional heat request					
C_TCO_MIN_REQ_HEAT_ACC	1	0... FEH	-48... 142.5	0.75	[°C]
TCO threshold for enabling additional heat request					
C_TQ_ADD_ACC_LGRD	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limitation gradient for TQ_ADD_ACC decrease after deactivation					
C_TQ_ADD_ACC_LGRD_ON	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limitation gradient for TQ_ADD_ACC increase at activation					
C_TQ_ADD_HEAT_ACC_LGRD	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limitation gradient for TQ_ADD_HEAT_ACC decrease after deactivation					

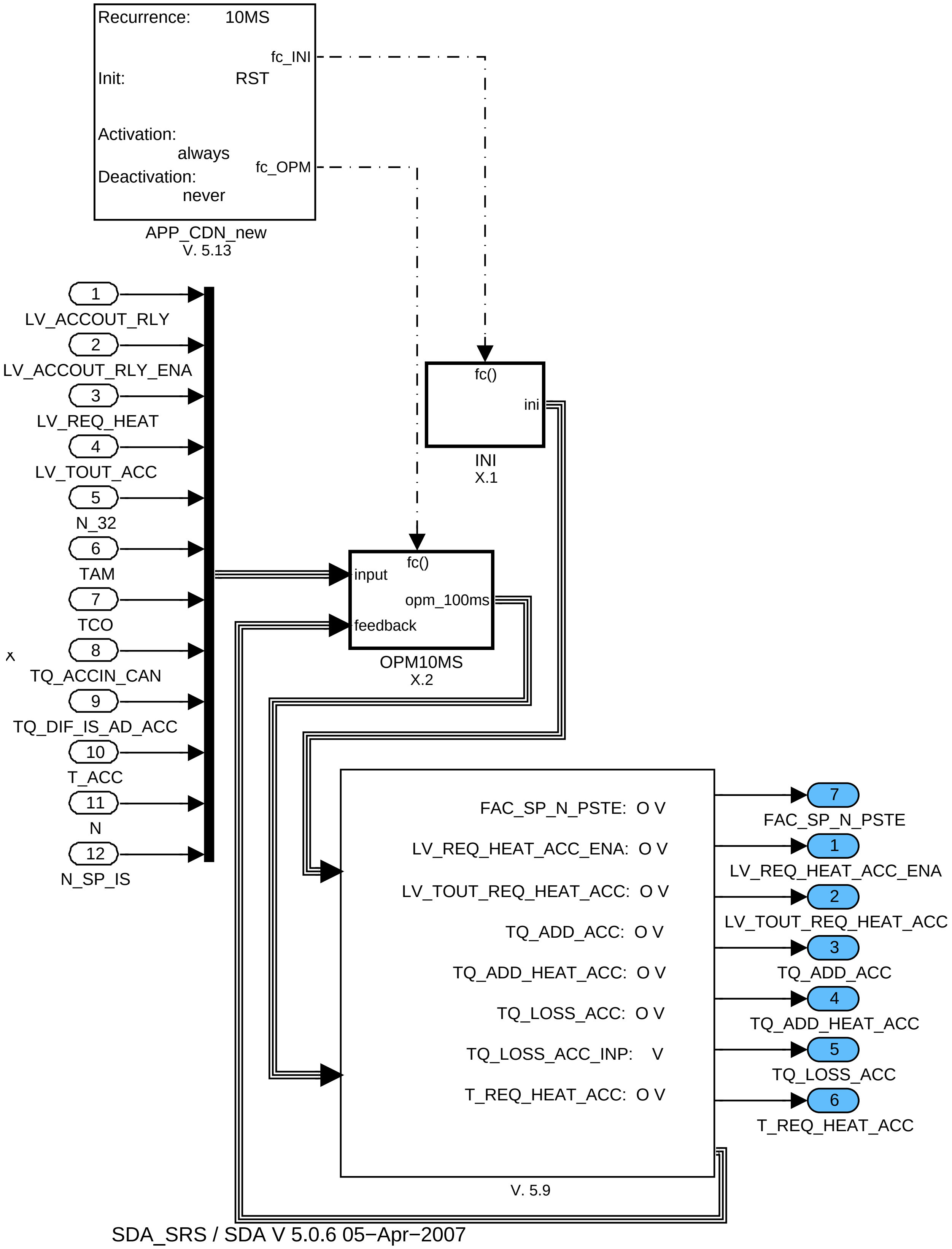
C_TQ_LOSS_MAX_ACC	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Maximum TQ_LOSS_ACC					
IP_FAC_TQ_LOSS_ACC_OFF	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDPM_T_ACC__FAC_TQ_LOSS_ACC	8	0... FFFFH	0... 655.35	0.01	[s]
TQ_LOSS_ACC weighting factor at ACC deactivation					
IP_FAC_TQ_LOSS_ACC_ON	8*8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDPM_T_ACC__FAC_TQ_LOSS_ACC_ON	8	0... FFFFH	0... 655.35	0.01	[s]
LDP_FAC_SP_N_PSTE_IP_FAC_TQ	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
TQ_LOSS_ACC weighting factor for A/C compressor clutch time behavior at activation					
IP_TQ_ADD_ACC	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32__TQ_ADD_ACC	6	0... FFH	0... 8160	32	[rpm]
LDP_TQ_LOSS_ACC_INP__TQ_ADD_ACC	6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for switching on air conditioning compressor depending on expected A/C torque loss					
IP_TQ_ADD_HEAT_ACC	6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32__TQ_ADD_HEAT_ACC	6	0... FFH	0... 8160	32	[rpm]
Torque reserve for switching on additional heating					
IP_TQ_LOSS_ACC_INP	6*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32__TQ_LOSS_ACC	6	0... FFH	0... 8160	32	[rpm]
LDP_TQ_ACCIN_CAN__TQ_LOSS_ACC	6	0... FFH	0... 127.5	0.5	[Nm]
ACC torque loss due to engine-speed and ACC torque demand					
LC_FAC_SP_N_PSTE_SWI	1	0... 1H	0... 1	1	[-]
LC to switch between use of N_SP_IS or C_N_SP_IS_PSTE(1:N_SP_IS)					

General Information

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.0.6 05−Apr−2007

Figure 1:

37.5.1 Initialisation

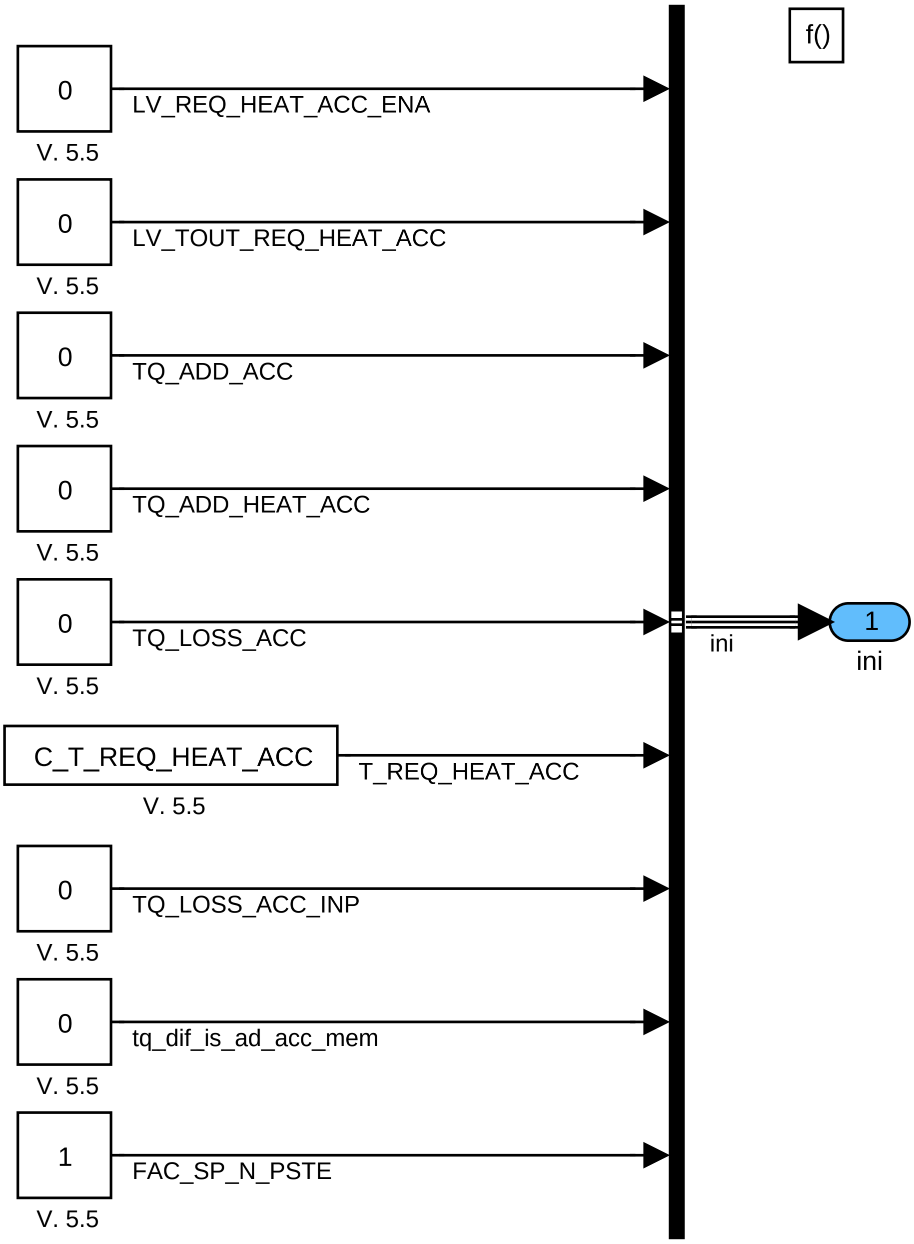


Figure 2:

37.5.2 Calculation of torque loss and reserve for Air conditioning compressor:

1.Determination of torque loss ACC
2.Determination of torque reserve for ACC
2.1 Determination of torque reserve ACC
2.2 Detemination of REQ_HEAT by ACC

37.5.2.1 Determination of Torque Loss ACC

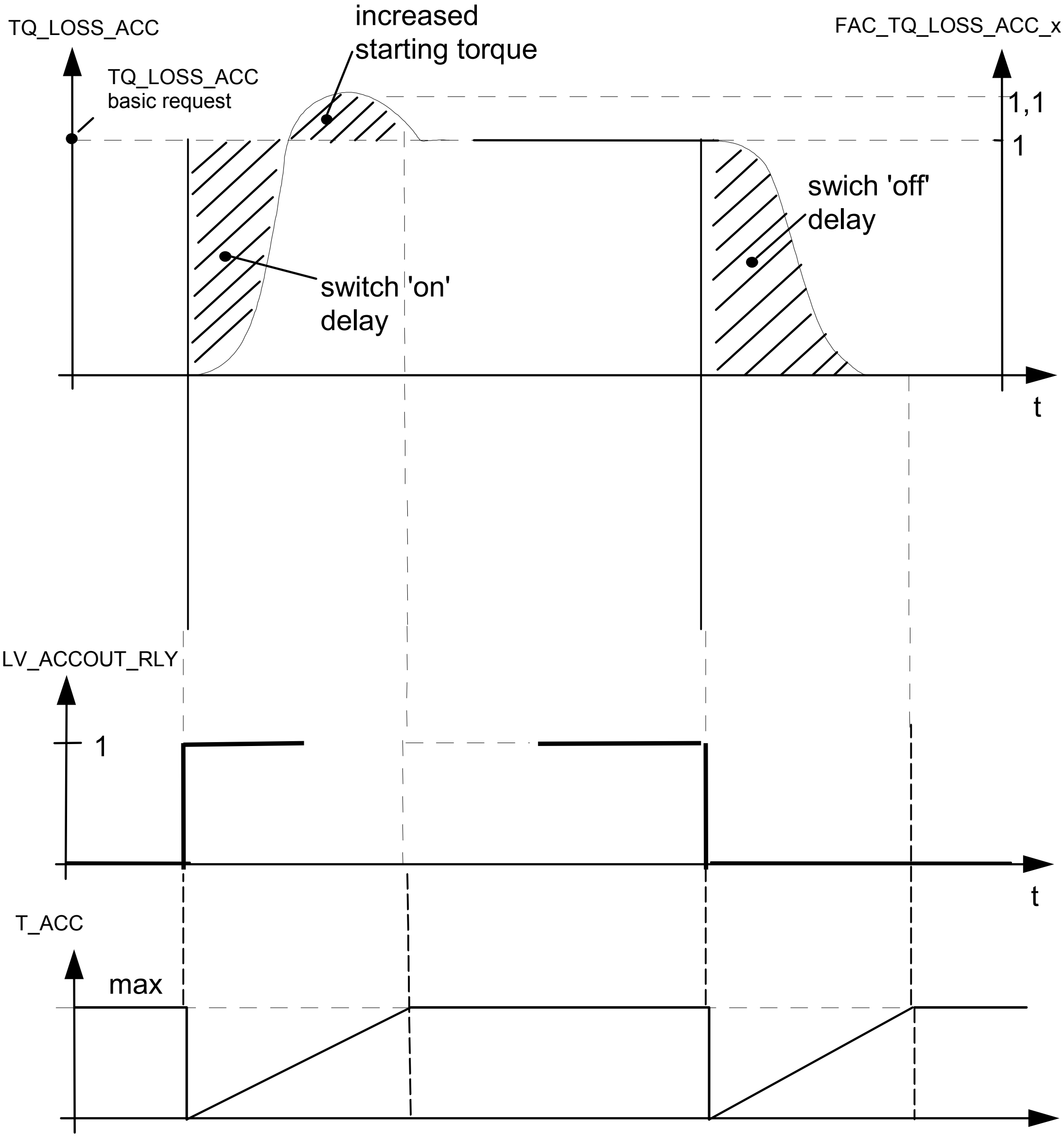
FUNCTION DESCRIPTION:

general specification

The calculation of TQ_LOSS_ACC starts already at transition LV_ACCOUT_RLY_ENA = 0 --> 1. The adaptation value TQ_DIF_IS_AD_ACC, determined in the module "idle speed adaptation", is added to the A/C compressor torque TQ_LOSS_ACC_INP which depends on ACC-torque from CAN and on engine speed (N_32).

The A/C-compressor is switched ON and OFF using an electro-magnetic clutch. If the clutch is closing the starting torque overshoots the torque calculated for steady state conditions (basic request) due to the inertia of compressor, which has to be accelerated. This overshoot is taken into account by the weighting factor IP_FAC_TQ_LOSS_ACC_ON depending on time T_ACC and by relative engine speed due to dynamic torque loss depending on target compressor speed (engine speed). The transient correction for clutch switching ON is always active as long as LV_ACCOUT_RLY_ENA = 1. So IP_FAC_TQ_LOSS_ACC_ON has to be 1 at steady state conditions are reached, at latest at T_ACC = 2.55s. At opening clutch the time delay until A/C compressor torque loss has reached 0 Nm is pre-controlled by IP_FAC_TQ_LOSS_ACC_OFF depending on T_ACC. The transient correction for clutch switching OFF is always active as long as LV_ACCOUT_RLY_ENA = 0. So IP_FAC_TQ_LOSS_ACC_OFF has to be 0 at latest at T_ACC = 2.55s.

Signal flow diagram:


TQ_LOSS_ACC
increased
starting torque
FAC_TQ_LOSS_ACC_x
TQ_LOSS_ACC
basic request
1,1
1
swich 'off'
delay
switch 'on'
delay
t
LV_ACCOUT_RLY
1
t
T_ACC
max

37.5.2.1.1 Calculation of ACC torque loss due to engine speed and ACC torque demand

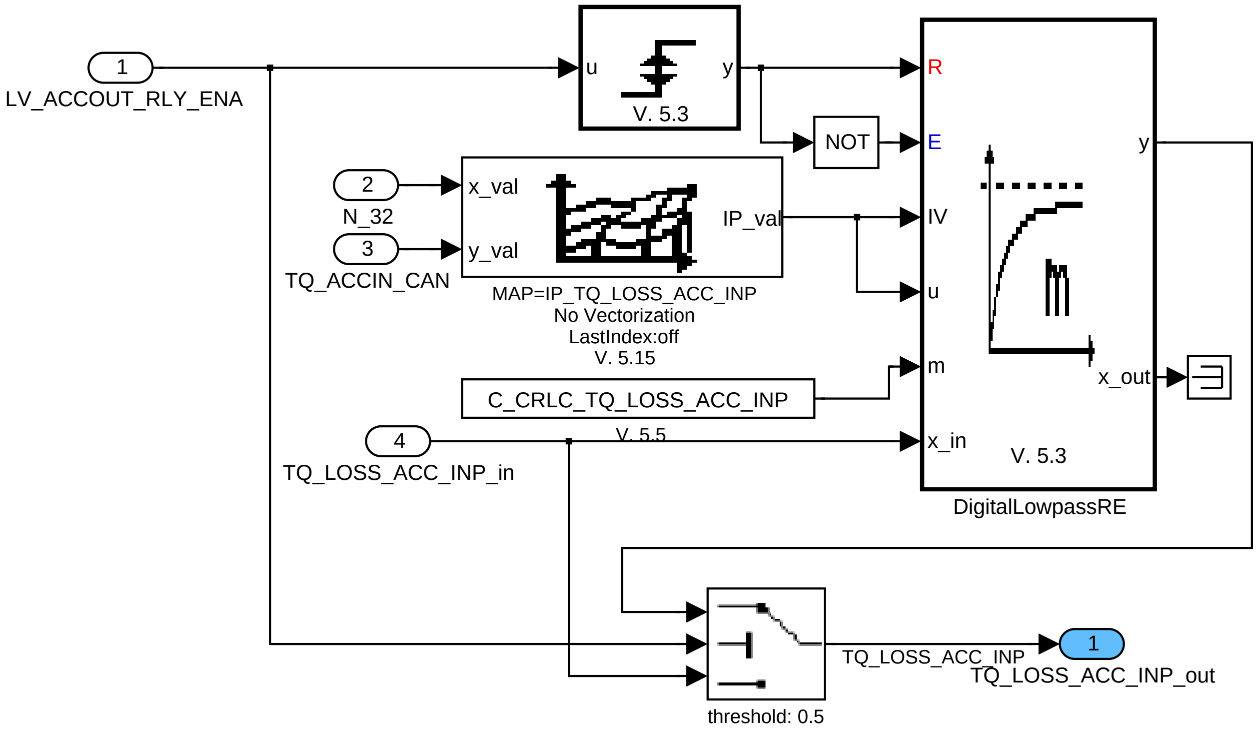


Figure 3:

37.5.2.1.2 Calculation of torque loss of air conditioning compressor

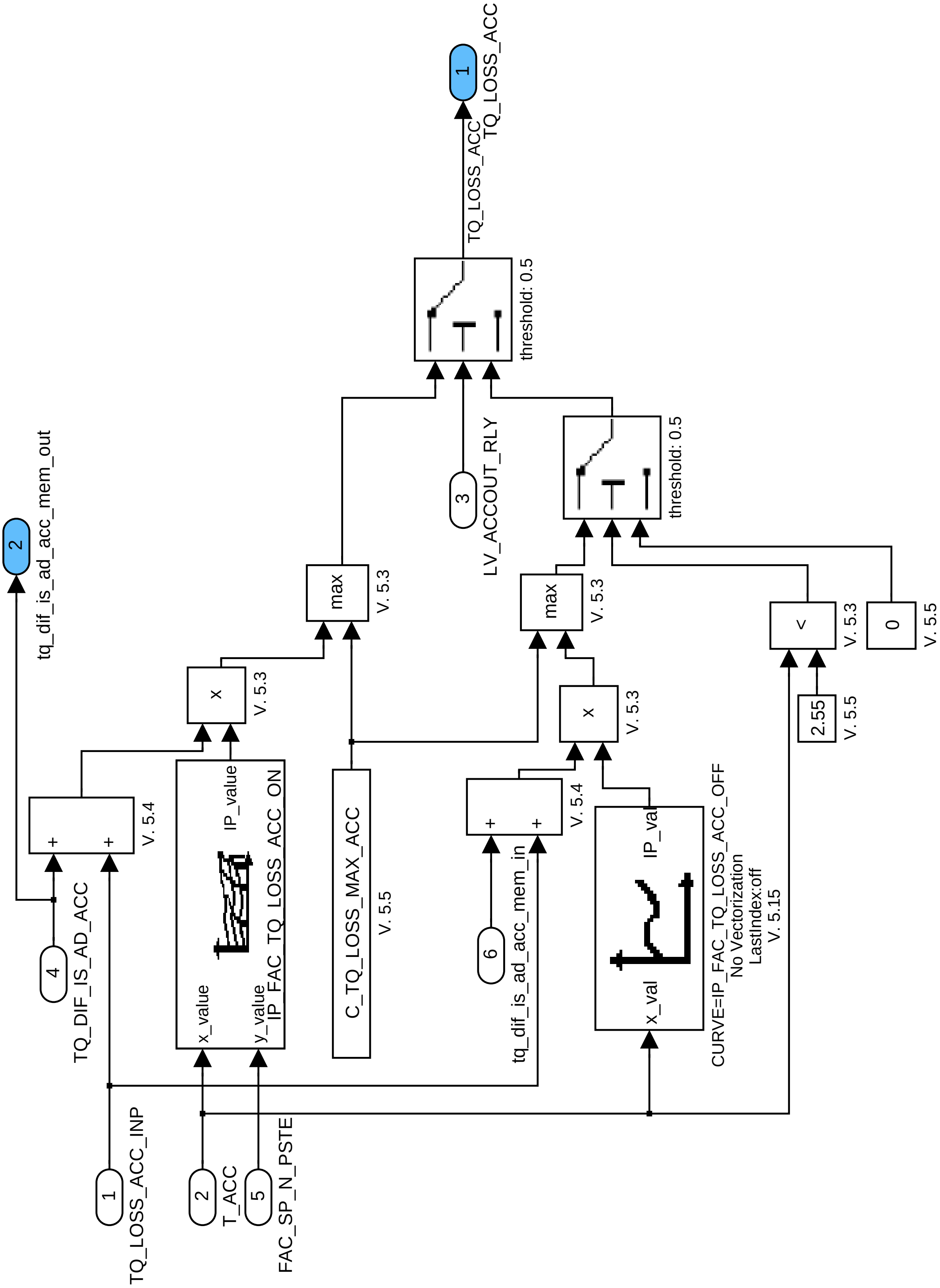


Figure 4:

37.5.2.1.3 Calculation of factor of engine speed relative to setpoint

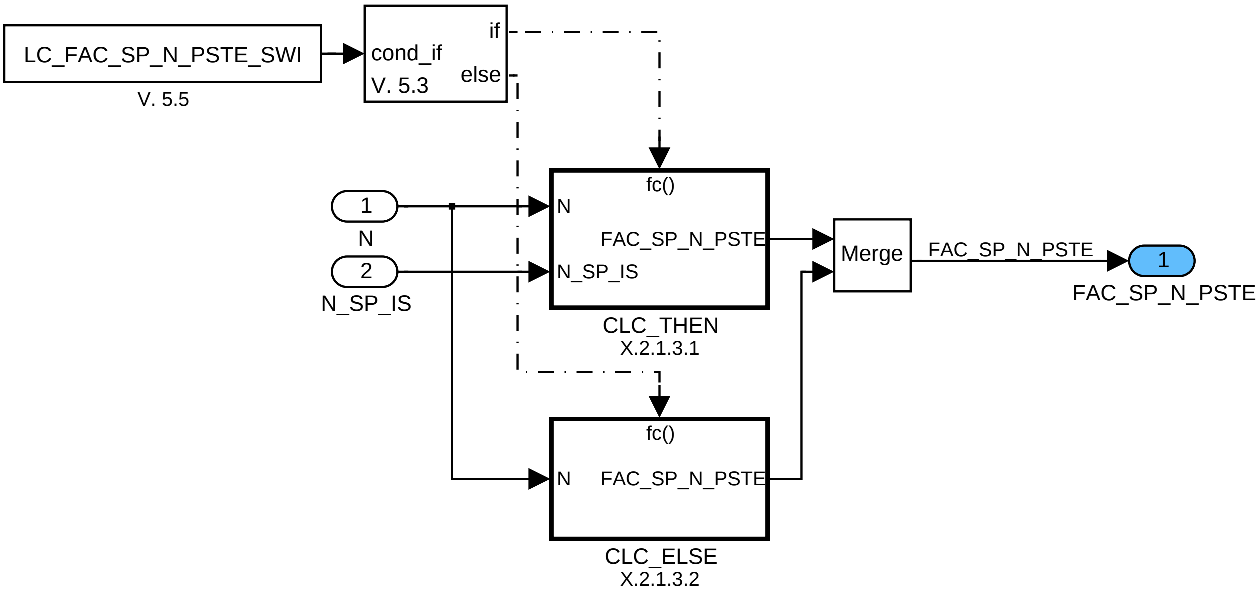


Figure 5:

37.5.2.1.3.1 Calculate FAC_N_SP_PSTE based on idle speed setpoint

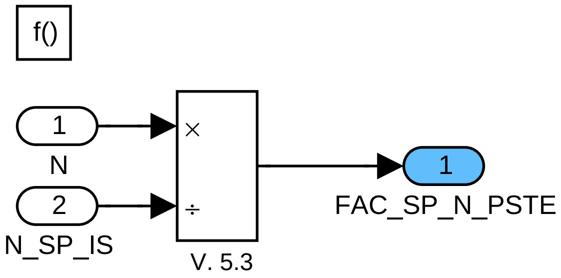


Figure 6:

37.5.2.1.3.2 Calculate FAC_N_SP_PSTE based on constant for idle speed setpoint

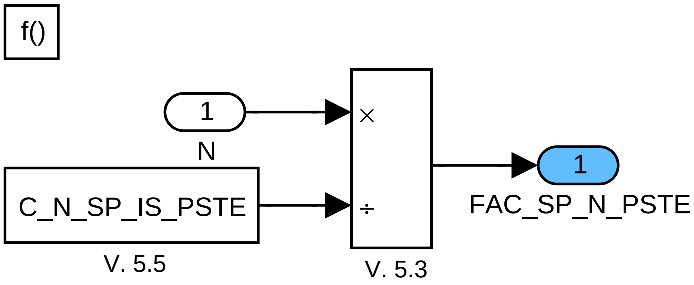
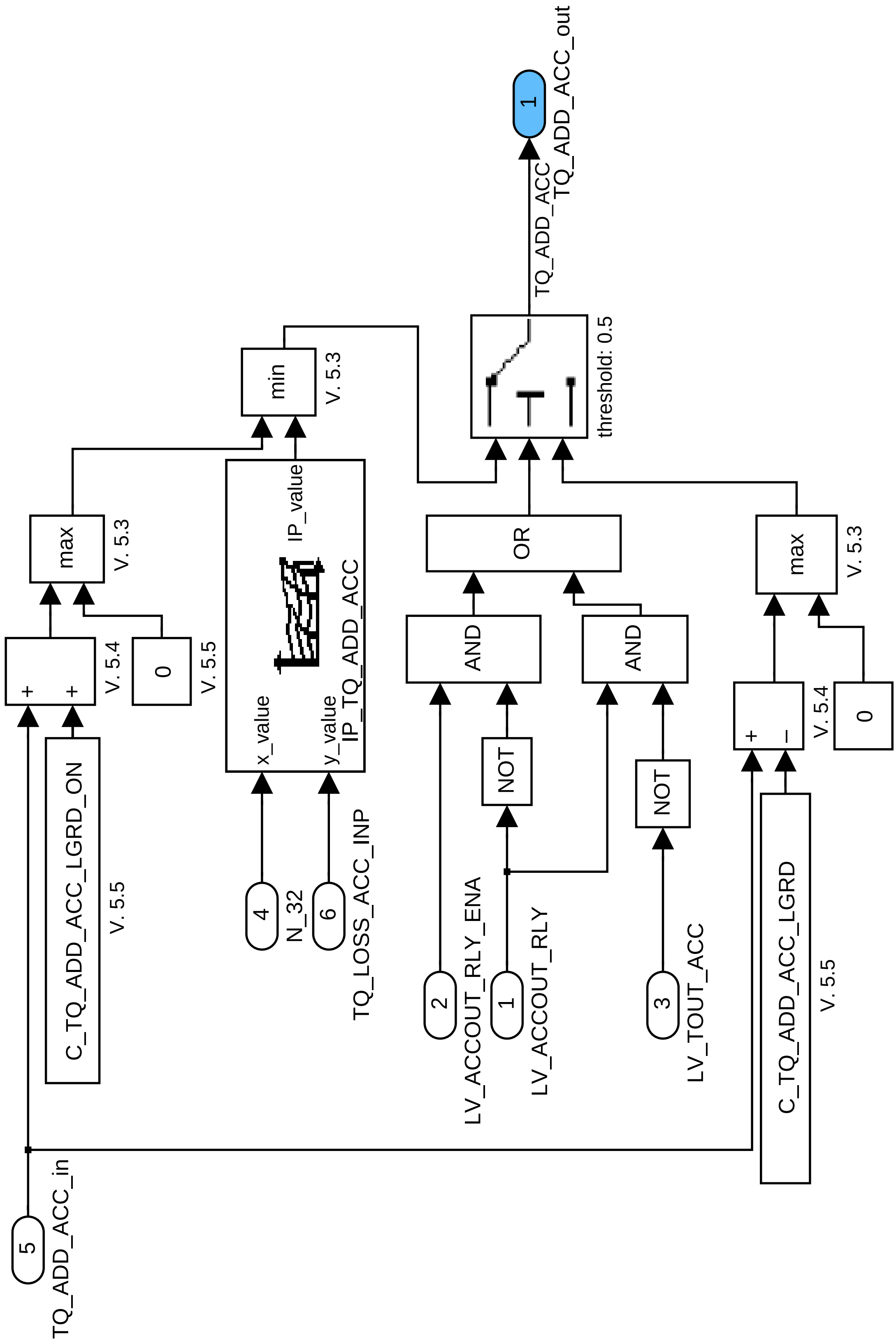


Figure 7:

37.5.2.2 Determination of Torque reserve for ACC

Torque reserve ACC
FUNCTION DESCRIPTION:
General information:
To guarantee idle quality during A/C compressor clutch switching ON a positive torque reserve can be requested. To avoid fast spark advance/retard rampsare used for activation/deactivation of TQ_ADD_ACC.



5
TQ_ADD_ACC_in
C_TQ_ADD_ACC_LGRD_ON
V. 5.5
+
+
V. 5.4
0
V. 5.5
max
V. 5.3
min
V. 5.3
x_value
y_value
IP_value
IP_TQ_ADD_ACC
V. 5.5
4
N_32
6
TQ_LOSS_ACC_INP
2
LV_ACCOUT_RLY_ENA
1
LV_ACCOUT_RLY
NOT
AND
AND
NOT
3
LV_TOUT_ACC
OR
threshold: 0.5
TQ_ADD_ACC
1
TQ_ADD_ACC_out
max
V. 5.3
+
–
V. 5.4
0
V. 5.5
C_TQ_ADD_ACC_LGRD
V. 5.5

Figure 8:

37.5.2.3 Determination of Torque reserve REQ_HEAT by ACC

Torque reserve REQ_HEAT by ACC
FUNCTION DESCRIPTION:
General information:
To guarantee idle quality during additional heating being present (required by INSTR3-message/CAN11H or IHKA/BN2000) a positive torque reserve can be requested.

At transition LV_REQ_HEAT_ACC_ENA = 0 --> 1 the timer T_REQ_HEAT_ACC is set to zero and starts to count until its maximum value is reached. The timer s maximum value is hold until transition LV_REQ_HEAT_ACC_ENA = 1 --> 0. Then the timer is set to zero and and starts again until its maximum value.

37.5.2.3.1 Calculation of LV_REQ_HEAT_ACC_ENA

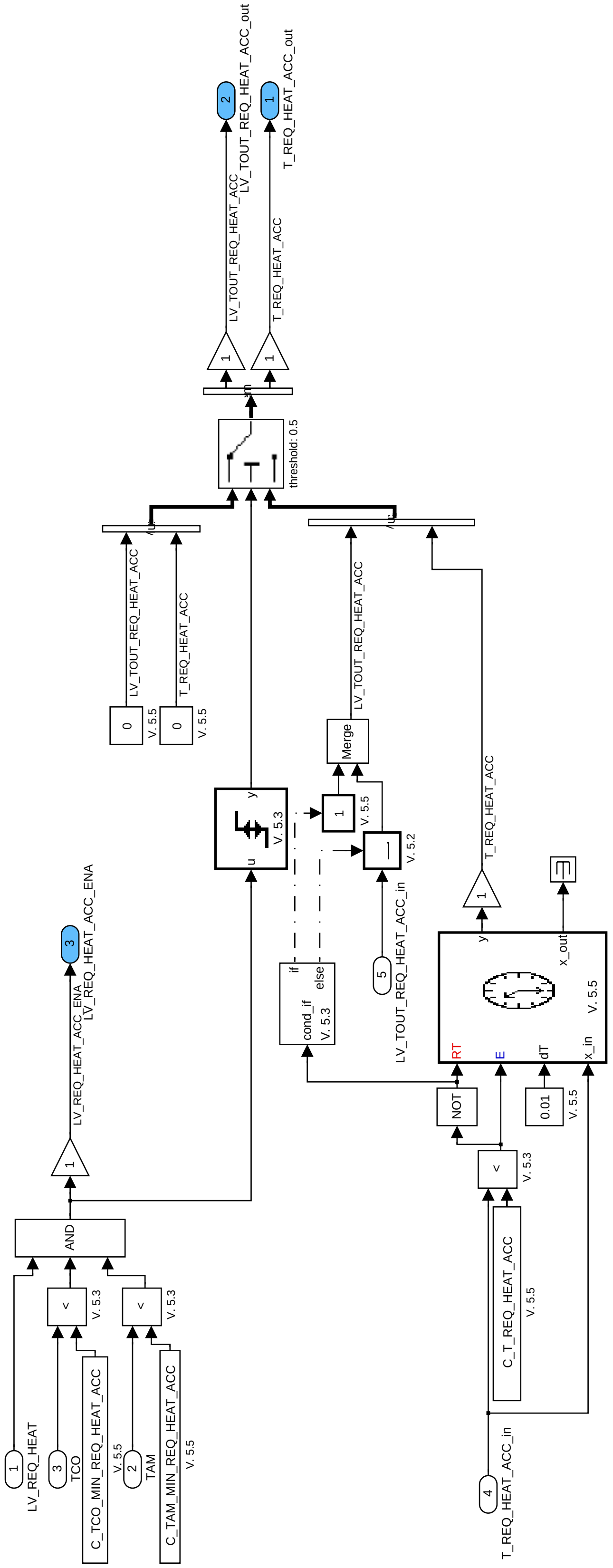


Figure 9:

37.5.2.3.2 Calculation of toeque reserve for switching on additional heating by ACC

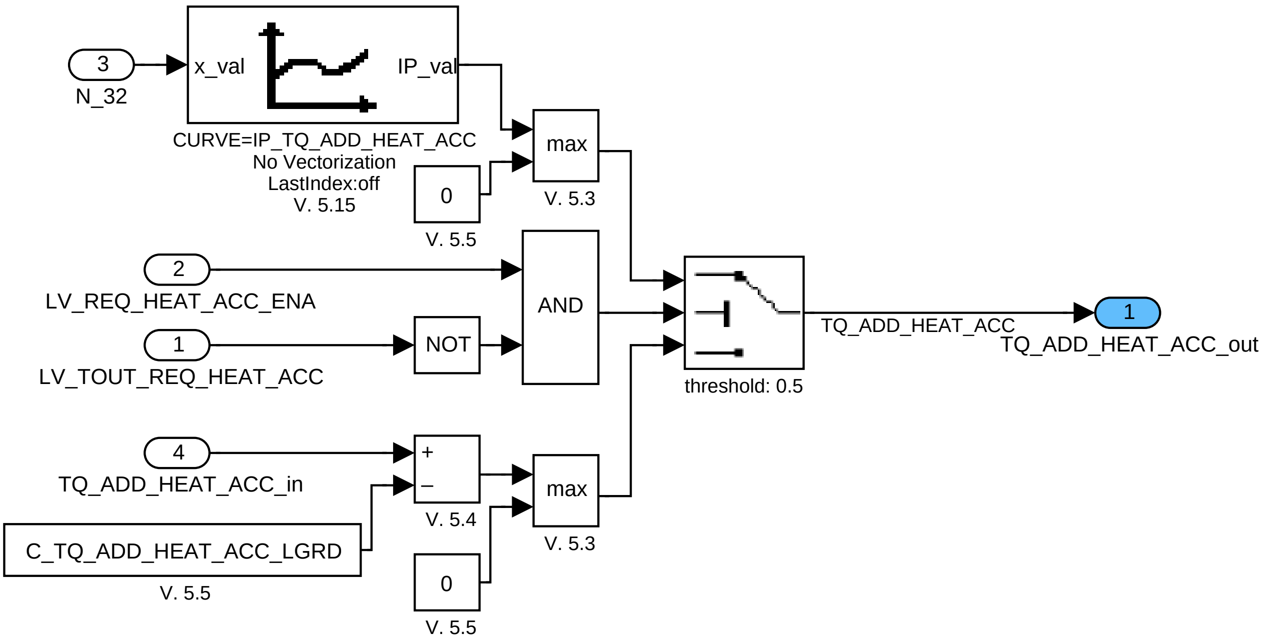


Figure 10:

38 Ignition realisation

Chapter Ignition realisation	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 6167 of 9643

Table of Contents

Index

38.1.1 Introduction

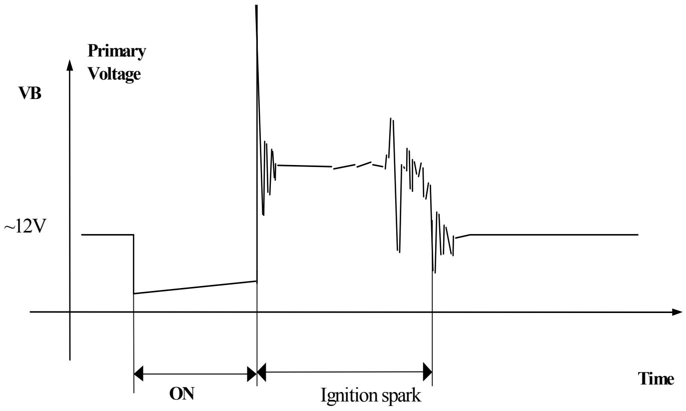
This document is the Aggregate Integration Document of the IGRE Aggregate. Its aim is to describe the integration constraints of this aggregate with the others. It aims with interfaces (imported and exported), architecture, and configuration. This document covers all versions of the IGRE aggregate.

38.1.2 Purpose

The IGRE Aggregate defines the calculation of the dwell time, the ignition realization and the ignition diagnosis functions.

Basically the ignition command function has to fulfill two main tasks; to ensure that there is enough energy in the ignition coil and to start the combustion at the requested ignition angle. Knowing the desired ignition advance, engine speed and several other parameters, this aggregate determines the correct timing for switching ON and OFF the power stage in order to bring the ignition coil to the necessary primary current value. The time during ON state (load phase) of the switch is often improperly called Dwell time (with reference to the dwell ratio in breaker systems).

At the end of this Dwell time the ignition stage is switched OFF and the ignition spark is created. The OFF event corresponds to the advance ignition angle (IGA_IGC).



The ignition command functions and the linked diagnostic functions are regrouped in the ignition realisation aggregate.

38.1.3 Function Description

This function sets the angular position of the dwell time turn on in order to have the time to set up the necessary current in the ignition coil.

The strategy is based on a priority of the ignition angle. If dwell time priority is requested the minimum dwell time is equal to maximum dwell time.

The strategy respects ignition coils with dual and single outputs. This function could be used independently of the number of cylinders (x) and the geometry of the crankshaft target wheel.

All of the functions described in this section operate over the entire engine speed range.

- Realization of minimum 1 ignition spark per cycle
- Realization of a number of multiple spark
- Physical limits of the ignition angle +60°CRK / -35,625°CRK

- In case of an acceleration of ±10 000 RPM/s the ignition spark output tolerance depending on engine speed.

Input data:

NC_IGN_DIAG_TYP	NC_TD_LIM_MIN	NC_TD_LIM_MAX_H	NC_MPL_T_MAX
NC_MAX_IGN_MPL_NR	NC_IGC_DLY	NC_MPL_IGN_H_RNG_CRK_MAX	NC_IGC_CONF
NC_TD_AD_TYP	NC_INI_CTR_DEAC	NC_IGBT_CUT_OFF_T	NC_T_MIN_SCP
NC_CFB_DIAG_SCP_CHN	NC_TD_AD_CHN_L	NC_TD_LIM_MAX_L	NC_TD_AD_CHN_H

General information :

The following describes the general rules for determination of the configuration data

38.1.4 Global configuration data

Data	Value

38.1.5 Local configuration data

Here are listed the configuration data, which are used only in the IGRE aggregate.

Data	Value
NC_IGN_DIAG_TYP	ATM46: Diag with ATM46 or ATIC71 circuit ATIC29: : Diag with ATIC29 circuit SHUNT: Diag with SHUNT (not supported yet)
NC_TD_LIM_MIN	TD_MIN> = NC_TD_LIM_MIN Typical value for debug = 1 ms
NC_TD_LIM_MAX_H	High maximum value for dwell time Typical value for debug = 20 ms
NC_TD_LIM_MAX_L	Low maximal value for dwell time Typical value for debug = 10 ms
NC_MAX_IGN_MPL_NR	Maximum number of multi spark Typical value for debug = 7
NC_IGC_DLY	Number of recurrence after engine start, which had to pass by to activate the diagnosis Typical value =16
NC_MPL_T_MAX	Absolute duration of multispark Typical value 60 ms
NC_MPL_IGN_H_RNG_CRK_MAX	Maximum ignition angle after TDC to start TD_MPL (normal value 12 to 18 °CRK)

Data	Value
NC_IGC_CONF	Ignition mode : half or full static Typical value = Full
NC_TD_AD_TYP	Type of adaptive dwell time control: NONE or FLAG or (CURR not supported today)
NC_INI_CTR_DEAC	Time delay after IGBT response before switching OFF the IGBT – typical value 600µs
NC_IGBT_CUT_OFF_T	Inhibition of "IGBT protection cut off function" (ATM46 ot ATIC71 usage without shunt) – typical value 5ms
NC_T_MIN_SCP	Minimum time to detect SCB with current feedback flag (ATIC71 usage) – typical value 1.5ms
NC_CFB_DIAG_SCP_CHN	Selection of the channel number to be used to do the SCB diagnosis whith current feedback flag
NC_TD_AD_CHN_L	Selection of channel number where current feedback flag is read to do dwell time adaptation
NC_TD_AD_CHN_H	Selection of channel number where current feedback flag is read to do dwell time adaptation in satrt phase

Note: care has to be taken with the setting of NC_TD_LIM_MAX in order to avoid Dwell Time / coil charging recovery:

- For half-static coil configuration, Dwell Time has to be shorter than 360°CRK (= delay time between 2 coil charging request).
  This means, for a maximal engine speed of eg. 7500rpm, Dwell Time < 8ms
- For full-static coil, Dwell Time has to be shorter than 720°CRK
  This means, for a maximal engine speed of eg. 7500rpm, Dwell Time < 16ms

NC_TD_LIM_MAX should be tuned in consequence to respect coil charging time no recovery and coil capacity

Formula Section:

NC_IGN_DIAG_TYP = "ATM46"

NC_TD_LIM_MIN = 1ms

NC_TD_LIM_MAX_L = 10ms

NC_TD_LIM_MAX_H = 20ms

NC_MAX_IGN_MPL_NR = 20

NC_IGC_DLY = 16

NC_MPL_T_MAX = 60ms

NC_MPL_IGN_H_RNG_CRK_MAX = 180°CRK

NC_IGC_CONF = FULL

NC_TD_AD_TYP = "Flag"

NC_INI_CTR_DEAC = 852µs

NC_IGBT_CUT_OFF_T = 5ms

NC_T_MIN_SCP = 1.5ms

NC_CFB_DIAG_SCP_CHN = 1

NC_TD_AD_CHN_L = 0

NC_TD_AD_CHN_H = 1

38.2 Spark Advance and Dwell Output

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
TD_FAC_MAX	O/V	1...FFH	1/128...255/128	1/128	-
Maximal dwell time control					
TD_FAC_MIN	O/V	1...FFH	1/128...255/128	1/128	-
Minimal dwell time control					

Input data:

LV_ ERR_CAM	VB	LV_ST	N_32
LDPM_N_32_1_IGRE	NC_CYL_NR		

Import actions:

ACTION_INFR_SetIgnCtl(IN <>)

FUNCTION DESCRIPTION:

This function sets the angular position of the dwell time turn on in order to have the time to set up the necessary current in the ignition coil.

The strategy is based on a priority of the ignition angle. If dwell time priority is requested the minimum dwell time is equal to maximum dwell time.

The strategy respects ignition coils with dual and single outputs. This function could be used independently of the number of cylinders (n) and the geometry of the crankshaft target wheel.

This function could be used for homogeneous combustion mode and for stratified combustion mode. The calculation of the spark advance is in both cases the same.

All of the functions described in this section operate over the entire engine speed range.

This specification includes two mode following the IGRE version: PI or PI + DI

38.2.1 Minimum / Maximum dwell time calculation

FUNCTION DESCRIPTION:

The two values are calculated independently of all environment factors. The aim is to have a larger window at start and low engine speed to obtain a better flexibility to apply the requested ignition angle.

These 2 values are expressed in term of min/max factor to be applied to the nominal dwell time and transferred to the Basic SW

The cylinder individual nominal Dwell Time to be applied requested by ASW function is defined in TD_IGC[x] taking into account the combustion mode switch

Application conditions:

Initialisation: TD_FAC_MIN = TD_FAC_MAX = 1 *at reset*

Recurrence: 10 ms

Formula section:

```
If      LV_ST = 1
Then (Start mode)
    If      VB > C_TD_VB_MAX
    Then    TD_FAC_MAX = 1
            TD_FAC_MIN  = Min[ C_TD_FAC_ST_MIN , 1]
    Else    TD_FAC_MAX  = Max[ C_TD_FAC_ST_MAX , 1]
            TD_FAC_MIN  = Min[ C_TD_FAC_ST_MIN , 1]
Else (run mode)
    If      VB > C_TD_VB_MAX
    Then    TD_FAC_MAX = 1
            TD_FAC_MIN  = Min[ IP_TD_FAC_MIN , 1]
    Else    TD_FAC_MAX  = Max[ IP_TD_FAC_MAX , 1]
            TD_FAC_MIN  = Min[ IP_TD_FAC_MIN , 1]
Endif
```

38.2.2 Half static of full static

Recurrence: engine stop to engine run

Formula section:

Following NC_IGC_CONF = Half or full, the ignition is half static controlled or full static controlled

ACTION_INFR_SetIgnCtl(NC_IGC_CONF)

38.2.3 Limp home without cam signal

Recurrence: engine stop to engine run

Formula section:

In case of CAM failure, the cylinder in combustion cannot be unequivocally identified. Engine phasing might be wrong. In such a case, a limp home mode function on Ignition is provided in order to have the possibility to drive the engine. Nevertheless, in this case, since the right cylinder phasing is not known, Ignition coil have to be driven in Half-Static mode to be sure to execute at least an ignition on the cylinder in combustion.

Decision to use such Limp Home functionality will depend on the setting of LC_IGC_LIH_CONF. If LC_IGC_LIH_CONF = 0, no Limp Home is provided in CAM failure and Ignition is stopped. If LC_IGC_LIH_CONF = 1, then Half-Static Ignition mode is done in case of CAM failure

If LV_ERR_CAM is active, the ignition coil control will be executed in half-static ignition-coil control mode if selected by the settings of LC_IGC_LIH_CONF

In such a case (*half-static mode activation in CAM failure*) the same Ignition and Dwell Time will be applied to the 2 Crank synchronous cylinders (*eg. on a 4 cyl engine, cyl. 1 and 4 and cyl. 2 and 3 will be fed with the same Ignition and Dwell Time*)

The limp home function could be applied only if the number of cylinders is even (2, 4, 6, ...)

```
If      NC_IGC_CONF = full
Then    If     LV_ERR_CAM = 1   And   LC_IGC_LIH_CONF = 1
        Then   ACTION_INFR_SetIgnCtl(HALF_STATIC)
        Else   ACTION_INFR_SetIgnCtl(FULL_STATIC)
        EndIf
Else    ACTION_INFR_SetIgnCtl(HALF_STATIC)
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_TD_ FAC_ST_ MIN	1	1...FFH	1/128...255/128	1/128	-
Factor for minimum dwell time					
C_TD_ FAC_ST_ MAX	1	1...FFH	1/128...255/128	1/128	-
Factor for maximum dwell time					
C_TD_VB_MAX	1	0…FFH	0...26	0,102	V
Maximum battery voltage to activate MPL (typical value : 16V)					
IP_TD_FAC_MAX	8	1...FF	1/128...255/128	1/128	-
LDPM_N_32_1_IGRE	8	0...FF	0...8160	32	rpm
Max dwell time factor					
IP_TD_FAC_MIN	8	1...FF	1/128...255/128	1/128	-
LDPM_N_32_1_IGRE	8	0...FF	0...8160	32	rpm
Min dwell time factor					
LC_IGC_LIH_CONF	1	0...1	0...1	1	-
Configuration Choice for Half static Mode in case of CAM limp home: 0 = no Limp Home - 1 = Half Static					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_IGC_CONF	1	1...2	1...2	1	-
Half static or full static (Typical value for debug = full static)					

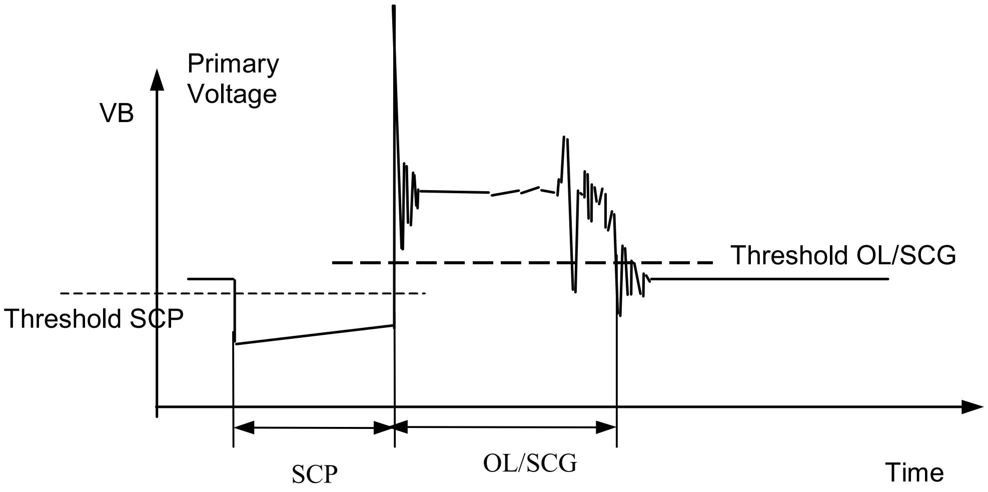
38.3 Ignition output diagnosis (for ATM 46 / Combined “ASIC”)

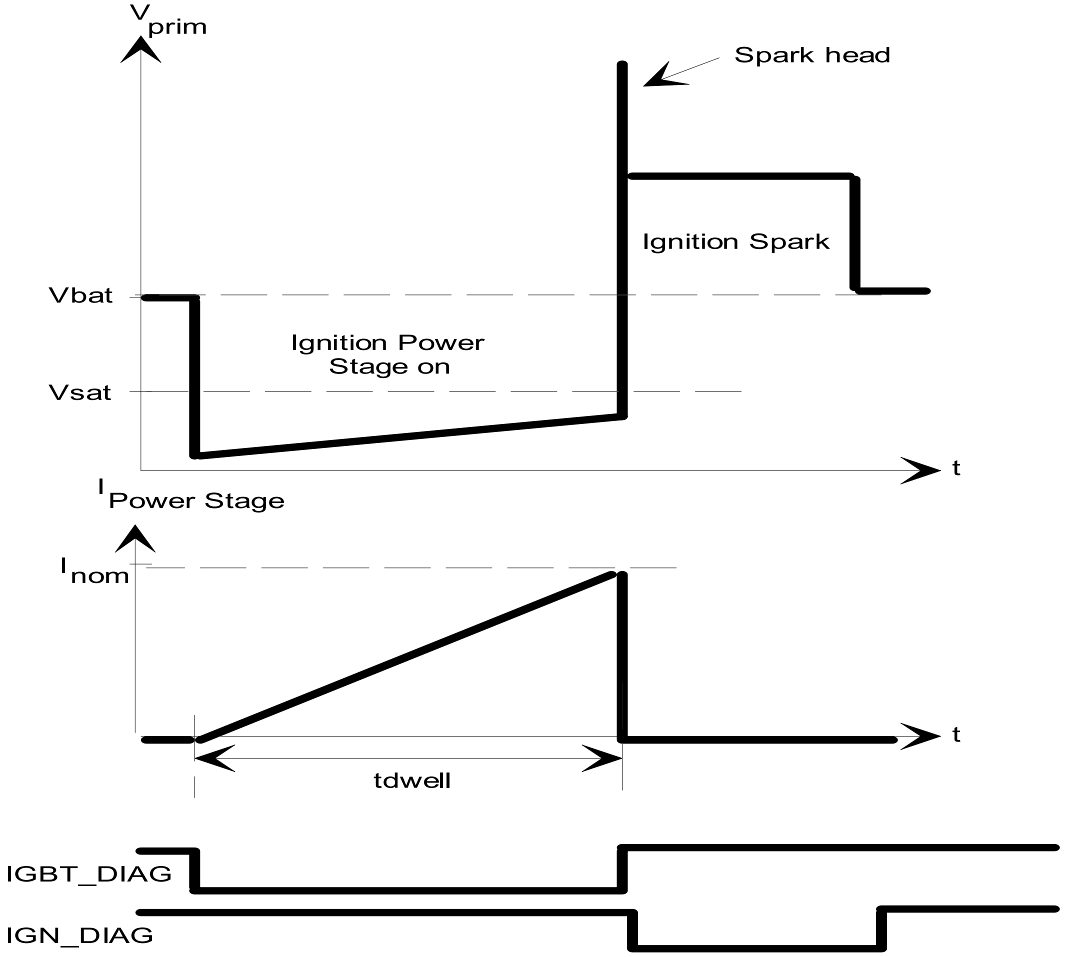
Each ATM46 could analyze 4 ignition coils. If diagnostics for more the than 4 coils are necessary, a parallel ATM46 has to be used.

The diagnosis is based on ignition coil primary over voltage measurement.

The input signal of the ATM 46 is the IGBT collector voltage.

In case of an ignition system with single output coils, the function does not work in camshaft limp home mode.





38.3.1 IGBT Protection (with “ATM46” / Combined “ASIC”)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SCP_IGC[NC_CYL_NR]	O/V	0...1H	0...1	1	-
Diagnostic of primary short circuit to battery (cylinder selected)					

Input data:

LV_SYN_ENG	NC_CYL_NR	SEG_NR	LV_CDN_DIAG_IGC_SCP[NC_CYL_NR]

Import actions:

ACTION_INFR_GetIgnScpDiag(IN<>, OUT<>)

FUNCTION DESCRIPTION:

The aim of the present function is to protect the ignition power stage against short circuit to battery (SCP).

The purpose of ignition power stage diagnostic is to detect a short circuit to battery voltage condition (SCP). At each Dwell Time On event, the ignition power stage is switched on. In normal condition, the current increase in the ignition power stage is limited by the inductance of the ignition coil solenoid.
In case of a short circuit to battery, the coil external device does not limit the current in the ignition power stage. In this situation the current is only limited by the current capability of the ignition power stage. This power dissipation would destroy the power stage. From the Dwell Time ON event in SCP condition, the IGBT’s can withstand this overload current only for a very short time (*750µs for IGBT without shunt*). So, it is necessary to switch off this IGBT transistor for safety and protection reasons by software (*IGBT deactivation by switching off the charging command in SW*) in case of an overload condition and after a certain time delay has elapsed (NC_INI_CTR_DEAC). This time delay is necessary in order to filter parasitic spikes. On the other hand this time delay must not exceed the maximum rating of the IGBT’s.

For ignition power stage diagnostic, the voltage across the ignition power stage is compared to a fixed threshold voltage. This threshold voltage is chosen to be higher than the maximum saturation voltage of the ignition power stage during a dwell time in normal operation. If there is no short to battery voltage condition, the voltage across the ignition power stage drops below this fixed threshold voltage at each dwell time. In case of a short to battery voltage failure, this voltage is not (or only for a short time) passed. A micro controller monitoring the ignition power stage diagnostic output is able to detect this fault.

This ignition power stage diagnostic circuit is designed for an engine management system including four ignition power stages. This four diagnostic information can be evaluated at one output by having a logic OR function of them.

After each deactivation from this protection function, the cut off IGBT will be reactivated again in the next ignition cycle

The function could be used for ignition systems with single-output coils and double-output coils. The IGBT protection function is cylinder selectively deactivated in case of single-output coil and linked dual cylinders for ½ static coils.

Application conditions:

Activation: *LV_SYN_ENG = 1*

Formula section:

Initialization: at reset
LV_SCP_IGC[x] = 0

Update rate: segment

System description:

The ATM46 is used to generate a signal evaluated by the SW for overload condition detection.

To perform the IGBT protection and SCP diagnostic, the BSW provides one functional service (BSW driver – one driver for one ATM46 protection and SCP diagnostic line)

As only one IGBT diagnosis output signal from the “ATM 46” exists (all channels are linked by a wired “OR”) and only one BSW functional service (*this BSW functional service used is described in the following paragraph*) is used in connection to this diagnosis output signal line, the function is limited to non-overlapping dwell time pulses. In case of overlapping Dwell Time, the last TD ON event that occurs will re-trigger the call of this function and thus be considered for diagnosis purpose.

Formula section:

ACTION_INFR_GetIgnScpDiag(SEG_NR -2,LV_SCP_IGC[SEG_NR-2])

38.3.1.1 Ignition Actuator Tests Diagnosis

Formula section:

Initialization: LV_SCP_IGC[x] = 0 at reset

System description:

For ignition coil actuator tests purpose, the SCP diagnostic function has to be enabled and activated upon the tests requests

The same case and comments (see before) apply here also.

Formula section:

```
For     x = 0 to NC_CYL_NR – 1
        If      LV_CDN_DIAG_IGC_SCP[x] = 1
        Then    ACTION_INFR_GetIgnScpDiag(SEG_NR -2,LV_SCP_IGC[SEG_NR-2])

        EndIf
EndFor
```

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_INI_CTR_DEAC	1	0...FFFFH	0...262,14	0,004	ms
Time delay after IGBT threshold response before switching OFF the IGBT - typical value 600 us					
NC_IGBT_CUT_OFF_T	1	0...FFFFH	0...262,14	0,004	ms
Inhibition of “IGBT protection cut off function”					
NC_T_MIN_SCP		0...FFFFH	0...262.14	0.004ms	ms
Minimum time to detect SCB with current feedback flag					

38.3.1.2 SCG (Burn Time measurement) - Acquisition of the burn time

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SCG_IGC[NC_CYL_NR]	O/V	0...1H	0...1	1	-
Diagnostic of primary short circuit to ground (cylinder selected)					
V_DUR_IGC[NC_CYL_NR]	O/V	0...FFFFH	0...262,14	0,004	ms
Burn time duration for diagnosis validation					

Input data:

LV_SYN_ENG	N_32	MFF_SP_MV	NC_CYL_NR

Import actions:

ACTION_INFR_GetIgnScgDiag (IN <>,IN<>,OUT<>,OUT<>)

Description:

This ignition diagnosis function evaluates the primary over voltage duration **V_DUR_IGC[x]** provided after the ignition coil cut off.

The aim of this spark burning diagnostic function is to measure the spark duration by detecting the presence of a spark on an ignition plug. The over-voltage at the primary coil of an ignition solenoid is equal to the voltage across the spark gap of the ignition plug divided by the transmission ratio of the ignition solenoid and superimposed on the battery voltage. The time where the voltage at the primary coil of the ignition solenoid is higher than a battery voltage dependent threshold voltage (V_{BD_TH}) gives the spark duration.

In the moment, when the ignition power stage (IGBT) is switched OFF *(ie. at TD OFF event occurrence)*, the voltage V_C at the collector of the IGBT rises, due to self-induction, and ignition occurs. As long as the ignition spark is burning, the voltage is well exceeding the battery voltage.

By means of a hardware defined threshold (V_{BD_TH}; typical value: V_{Bat} + 3V) the burn condition is detected and a matching signal is generated, *eg. Here by the combined ASIC ATM46*. This signal is used to measure the overall burning duration.

The Ignition Burning diagnostic signal IGN_DIAG is generated by this combined ASIC. This time for the burning spark IGN_DIAG is calculated via the duration of the diagnostic signal staying above the threshold of V_{BD_TH}.

This ATM46 ignition diagnostic circuit is designed for an engine management system including up to 4 ignition coils/solenoids. This four diagnostic information are then evaluated on one output by applying a logic NOR function on them. The output signal IGN_DIAG of the ignition diagnostic circuit is processed by the µP.

Due to combustion chamber turbulence the spark of an ignition plug is not stable. The result is an unstable primary voltage at the ignition solenoid. Therefore a filter in the output signal IGN_DIAG is necessary to prevent the µP from high frequency inputs.

The ignition coil primary over voltage on the ignition output is roughly equal to the voltage across the spark gap divided by the ignition coil transformation ratio, superimposed on the battery voltage.

The Burn Time measurement is used today only for Spark / Combustion Diagnosis Purpose. It is a rough measurement of the effective Burn Time.

Please note that the SCG detection is equivalent to the detection of an Open Load.

Application conditions:

Activation: LV_SYN_ENG = 1

Initialisation: LV_SCG_IGC[x] = 0

at reset

V_DUR_IGC[x] = 0

Recurrence: segment

Formula section:

ACTION_INFR_GetIgnScgDiag(SEG_NR-2,ID_V_DUR_IGC_MIN,
V_DUR_IGC[SEG_NR-2], LV_SCG_IGC[SEG_NR-2])

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
ID_V_DUR_IGC_MIN	8x6	0...FFFFH	0...262,14	0,004	ms
LDP_N_32_ID_V_DUR_IGC_MIN	8	0...FFH	0...8160	32	rpm
LDP_MFF_SP_MV_ID_V_DUR_IGC_MIN	6	0...FFFFH	0...1389	0.0211948	mg/stk
Minimum duration of the Burntime to detect ignition errors					

38.4 Dwell time adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TD_AD[NC_CYL_NR]	O/V	8000...7FFFH	-131.072...131.068	0.004	ms
Dwell time adaptative					
TD_AD_H[NC_CYL_NR]	V/S	8000...7FFFH	-131.072...131.068	0.004	ms
Dwell time adaptative during high mode current					
TD_AD_L[NC_CYL_NR]	V/S	8000...7FFFH	-131.072...131.068	0.004	ms
Dwell time adaptative in low mode current					
LV_TD_AD_ACQ_VLD	V	0...1	0...1	1	-
Flag indicating validity of current measurement					
T_SUM_TD_AD[NC_CYL_NR]	V	0...FFFFH	0...262.14	0.004	ms
Sum of added time on TD_AD without reaching current threshold					
LV_TD_AD_SWI	V	0...1H	0...1	1	-
Flag indicating switch in between low and high mode currents					
CTR_TD_AD_SWI	V	0...FFH	0...255	1	-
Counter used for the switch in between low and high mode currents					

Input data:

LV_INH_TD_AD	LV_SCG_IGC[NC_CYL_NR]	LV_SCP_IGC[NC_CYL_NR]	LV_ERR_CAM
T_THD_IGCFB	SEG_NR	LV_TD_AD_H	N_32
TD_IGC[NC_CYL_NR]	NC_CYL_NR		

Export actions:

ACTION_IGRE_*GetIgnDwellAdNvmy*(IN <PRM_Cyl>, OUT <PRM_Td_ad_l>, OUT <PRM_Td_ad_h>)
This action allows getting the values of dwell time adaptation stored in NVMY.
ACTION_IGRE_*RstIgnDwellAdNvmy*(IN <PRM_Cyl>)
This action allows resetting the values of dwell time adaptation.

Description for actions:

ACTION_IGRE_*GetIgnDwellAdNvmy*(IN < PRM_Cyl>, OUT < PRM_Td_ad_l>, OUT < PRM_Td_ad_h>)

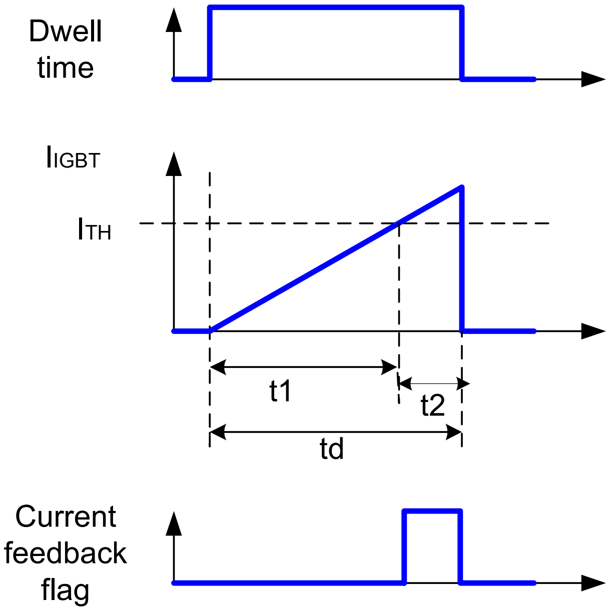
This action allows getting the values of dwell time adaptation stored in NVMY.
For specified cylinder Cyl, the dwell time adaptative values TD_AD_L and TD_AD_H will be read.
If cylinder number received is not in the range of the engine cylinder number, the maximum value will be put in output for TD_AD_L and TD_AD_H.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
PRM_Cyl	IN	0...7	0...7	1	-
Cylinder number on which we want to get the dwell time adaptative values					
PRM_Td_ad_l	OUT	8000...7FFFH	-131.072...131.068	0.004	ms
Dwell time adaptative for low current mode					
PRM_Td_ad_h	OUT	8000...7FFFH	-131.072...131.068	0.004	ms
Dwell time adaptative for high current mode					

ACTION_IGRE_*RstIgnDwellAdNvmy* (IN <PRM_Cyl>)					
This action allows resetting the values of dwell time adaptation stored in NVMY. This action will reset for the specified cylinder the adadaptive values TD_AD_L and TD_AD_H to **0 ms**.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_Cyl	IN	0...7	0...7	1	-
Cylinder number on which we want to reset the dwell time adaptative values					

FUNCTION DESCRIPTION:

This function should be used with ATIC71 component. Through the 2 current feedback flags of the component it is possible to measure the time spent above a given current threshold during the main dwell pulse.



Two different current thresholds can be used depending on which engine condition we are.

The current threshold is fixed by hardware component at the current target the coils have to reach. If the threshold is not reached we increase the dwell time. In the other case we will decrease the dwell time by the time t2(time measured above the current threshold).

Please note that in this whole specification x corresponds to SEG_NR-2.

Application conditions:

Initialisation:

in case of EEPROM fault or new ECU: All TD_AD[x] = 0 ms

At reset :

```
if    LC_TD_AD_CLR = 1
then        for x = 0 to NC_CYL_NR-1
```

```
                            TD_AD_H[x] = 0
                            TD_AD_L[x] = 0
                        Endfor
        Else            TD_AD_H[x] = saved value
                        TD_AD_L[x] = saved value
        endif
```

After power latch : all TD_AD_H[x] and all TD_AD_L[x] are stored into the non-volatile memory

At reset: T_SUM_TD_AD[x] = 0ms

TD_AD[x] = TD_AD_H[x] for all cylinders

Recurrence: segment

```
    if N_32 > C_ N_32_MIN_TD_AD
    Then recurrence = 1 out of NC_CYL_NR-1 segment
    EndIf
```

Activation: at the 3rd segment occurence

Deactivation:

Formula section:

Detection of current threshold switch

```
If          $LV\_TD\_AD\_H_n$ != $LV\_TD\_AD\_H_{n-1}$
Then        for x = 0 to NC_CYL_NR-1
                T_SUM_TD_AD[x] = 0
            Endfor
            CTR_TD_AD_SWI = NC_CYL_NR + 1
            LV_TD_AD_SWI = 1
Else
    If      CTR_TD_AD_SWI = 0
    Then    nop           // non operating: no action
    Elseif  CTR_TD_AD_SWI = 1
    Then    CTR_TD_AD_SWI = CTR_TD_AD_SWI - 1
            LV_TD_AD_SWI = 0
    Else    CTR_TD_AD_SWI = CTR_TD_AD_SWI - 1
    Endif
Endif

If  CTR_TD_AD_SWI = NC_CYL_NR
    For x = 0 to NC_CYL_NR-1
        Then If     LV_TD_AD_H = 1
             Then   TD_AD[x] = TD_AD_H[x]
             Else   TD_AD[x] = TD_AD_L[x]
             Endif
        End if
    endfor
endif
```

Check of the validity of current feedback flag result

```
If          T_THD_IGCFB          ≥     C_T_THD_IGCFB_MAX
    Or      (LV_SCG_IGC[x]       ==    1 And TD_IGC[X] > C_TD_IGC_OL_MIN)
    Or      LV_SCP_IGC[x]        ==    1
    Or      LV_ERR_CAM           ==    1
    Or      LV_INH_TD_AD         ==    1
    Or      LV_TD_AD_SWI         ==    1
Then        LV_TD_AD_ACQ_VLD     ==    0
Else        LV_TD_AD_ACQ_VLD     ==    1
Endif
```

Calculation of TD_AD

```
If    LV_TD_AD_ACQ_VLD = 1
Then
    if      T_THD_IGCFB=0
    Then    If      T_SUM_TD_AD[x]$_n$ <  C_T_SUM_TD_AD_MAX
            Then    TD_AD[x]$_n$ = TD_AD[x]$_{n-1}$ + C_TD_AD_INC
                    T_SUM_TD_AD[x]$_n$ = T_SUM_TD_AD[x]$_{n-1}$ + C_TD_AD_INC
            Endif
    ElseIf  T_THD_IGCFB > C_TD_AD_THD
    Then            TD_AD[x]$_n$ = TD_AD[x]$_{n-1}$ – C_FAC_DEC_TD_AD * T_THD_IGCFB
                    T_SUM_TD_AD[x]$_n$ = 0
    Else    T_SUM_TD_AD[x]$_n$ = 0
    Endif

    Limitation to maximum and minimum values authorized for dwell time adaptation
    If              TD_AD[x] > C_TD_AD_MAX
    Then            TD_AD[x] = C_TD_AD_MAX
    Endif
    If              TD_AD[x] < C_TD_AD_MIN
    Then            TD_AD[x] = C_TD_AD_MIN
    Endif

    If      LV_TD_AD_H = 1
    Then    TD_AD_H[x] = TD_AD[x]
    Else    TD_AD_L[x] = TD_AD[x]
    Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TD_AD_MIN	1	8000...7FFFH	-131.072...131.068	0.004	ms
Minimum value for dwell time adaptative					
C_TD_AD_MAX	1	8000...7FFFH	-131.072...131.068	0.004	ms
Maximum value for dwell time adaptative					
C_TD_AD_INC	1	0...FFH	0...1.02	0.004	ms
Increment on dwell time in case current threshold is not reached					
C_TD_AD_THD	1	0...FFFFH	0...262.14	0.004	ms
Definition of threshold to enable dwell adaptation					
C_FAC_DEC_TD_AD	1	0...FFH	0...0.996	3.9e-3	-
Factor applied on T_THD_IGCFB to calculate time to be removed to TD_AD					
C_T_THD_IGCFB_MAX	1	0...FFFFH	0...262.14	0.004	ms
Maximum measured time for valid sample					
C_T_SUM_TD_AD_MAX	1	0...FFFFH	0...262.14	0.004	ms
Maximum time allowed to be added in TD_AD without reaching target threshold					
LC_TD_AD_CLR	1	0...1H	0...1	1	-
Logical value enabling reset of variables TD_AD[x]					
C_ N_32_MIN_TD_AD	1	0...FFH	0...8160	32	rpm
Engine speed threshold at which the TD_AD calculation change its recurrence					
C_TD_IGC_OL_MIN	1	0...FFFFH	0...262.14	0.004	ms
Minimum value at which we inhibit the adaptation of dwell time in case of open load					

38.5 Dwell time adaptation (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_THD_IGCFB	V/O	0...FFFFH	0...262.14	0.004	[ms]
Time measurement between current threshold and end of dwell					
TD_CLC	V/O	0...FFFFH	0...262.14	0.004	[ms]
Dwell time calculated by BSW					
LV_TD_AD_H	V/O	0...1H	0...1	1	[-]
Flag indicating dwell time adaptation is done in start					
LV_INH_TD_AD	V/O	0...1H	0...1	1	[-]
Flag indicating that dwell time adaptative function is inhibited					

Input data:

LV_ST	TCO	OPM_AV	

Import actions:

ACTION_INFR_GetIgnDwellAd(IN, IN, OUT, OUT)
ACTION_INFR_SetIgnConfigDiag(IN)

38.5.1 Channel configuration for diagnosis

38.5.2 FUNCTION DESCRIPTION:

In case of dwell time adaptation by current feedback flag, the SCB diagnosis is also done by current feedback flag.

It is necessary in this case to select on which channel will be done the SCB diagnosis.

Application conditions:

Initialisation:

Activation: at engine stop to engine run

Deactivation:

Formula section:

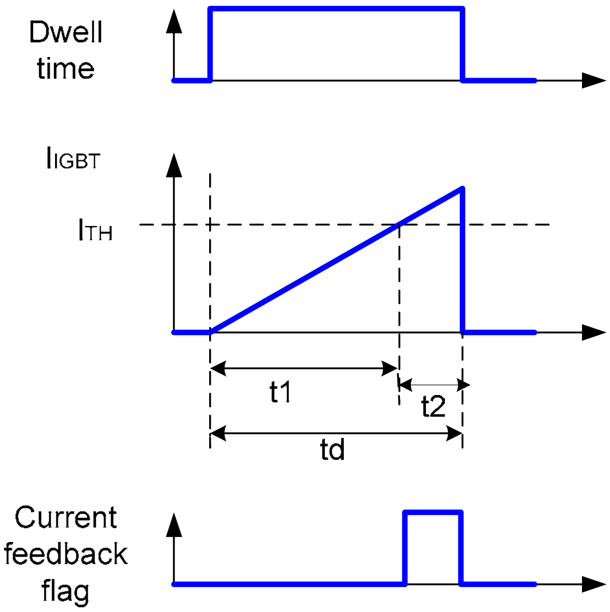
Channel configuration for SCB diagnosis with current feedback flag

ACTION_INFR_SetIgnConfigDiag(NC_CFB_DIAG_SCP_CHN)

38.5.3 Reading of current feedback flag for dwell time adaptation

38.5.4 FUNCTION DESCRIPTION:

This function should be used with ATIC71 component. Through the 2 current feedback flags of the component it is possible to measure the time spent above a given current threshold during the main dwell pulse.



Two different current thresholds can be used depending on which engine condition we are.

The current threshold is fixed by hardware component at the current target the coils have to reach. If the threshold is not reached we increase the dwell time. In the other case we will decrease the dwell time by the time t2(time measured above the current threshold).

Application conditions:

Initialisation: at reset LV_INH_TD_AD = 0

Recurrence: segment

Activation: at the 3rd segment occurence

Deactivation:

Formula section:

Reading of the result of current feedback flag

```
If        (LV_ST = 1) or TCO < C_TCO_TD_AD_SWI
Then
          LV_TD_AD_H = 1
          ACTION_INFR_GetIgnDwellAd(NC_TD_AD_CHN_H,SEG_NR-2,T_THD_IGCFB, TD_CLC)

ElseIf    OPM_AV = 1H or  OPM_AV = 3H    (S or AFL = HOM_S mode)

Then

          ACTION_INFR_GetIgnDwellAd(NC_TD_AD_CHN_H,SEG_NR-2,T_THD_IGCFB, TD_CLC)
          LV_TD_AD_H = 1

Else      LV_TD_AD_H = 0
          ACTION_INFR_GetIgnDwellAd(NC_TD_AD_CHN_L,SEG_NR-2, T_THD_IGCFB, TD_CLC)
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TCO_TD_AD_SWI	1	0...FEH	-48...142.5	0.75	[°C]
TCO threshold to enable switch from channel NC_TD_AD_CHN_ST to NC_TD_AD_CHN					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CFB_DIAG_SCP_CHN	1	0...1	0...1	1	
Selection of the channel number to be used to do the SCB diagnosis whith current feedback flag 0 : CURR_TH1 1 : CURR_TH2					
NC_TD_AD_CHN_L	1	0...1	0...1	1	
Selction of channel number where current feedback flag is read to do dwell time adaptation 0 : CURR_TH1 1 : CURR_TH2					
NC_TD_AD_CHN_H	1	0...1	0...1	1	
Selction of channel number where current feedback flag is read to do dwell time adaptation during start phase 0 : CURR_TH1 1 : CURR_TH2					

38.6 IGRE – Requirements to infrastucture interface

Input data:

NC_MPL_T_MAX	NC_MPL_IGN_H_RNG_CRK_MAX		

Detailed input data definition:

Name	Hex. limits	Phys. limits	Resol.	Unit
NC_MPL_T_MAX	0...FFFFH	0...262.14	0.004	ms
Maximum time duration of multiple spark (normal value 60ms)				
NC_MPL_IGN_H_RNG_CRK_MAX	0...12C0H	0...300	0.0625	°CRK
Maximum ignition angle after TDC to start TD_MPL (normal value 12 to 18°CRK)				

Export actions:

ACTION_INFR_SetIgnEnable (IN <Cyl>, IN <Inh_igc>)
Enables (inh_igc=1) or disables (inh_igc=0) the ignition for cylinder Cyl.
ACTION_INFR_SetIgnAngle (IN <Cyl>, IN <Iga_igc_h_rng>)
Sets the ignition angle (high range) to apply for cylinder Cyl
ACTION_INFR_SetIgnDwell (IN <Cyl>, IN<Td_fac_min>, IN <Td_igc>, IN <Td_fac_max>)
Sets the dwell time duration and the factors for maximal and minimum dwell time for cylinder Cyl
ACTION_INFR_SetIgcDwellTest (IN <Cyl>, IN <Td_igc>)
Sets the dwell time duration for cylinder Cyl and activate coil charging accordingly
ACTION_INFR_SetIgnMpl (IN <Cyl>, IN <Td_mpl1>, IN <Td_mpl>, IN <Td_mpl1_dly>, IN <Td_mpl_dly>, IN <Ign_mpl_nr>)
Sets dwell times, dwell time interruptions and number of ignition sparks of multiple sparks
ACTION_INFR_SetIgnCtl (IN <Ign_ctl_mod>)
Sets the ignition in full static or half static mode

Description for actions:

ACTION_INFR_ SetIgnEnable (IN <Cyl>, IN <Inh_igc>)					
Enables (inh_igc=1) or disables (inh_igc=0) the ignition for cylinder Cyl.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cyl	IN	0 ... 7	0 ... 7	1	-
Cylinder number affected by the inhibition of ignition system					
Inh_igc	IN	0 ... 1	0 ... 1	1	1
Flag indicating inhibition of ignition system					

ACTION_INFR_SetIgnAngle (IN <Cyl>, IN <Iga_igc_h_rng>)					
Sets the ignition angle (high range) to apply for cylinder Cyl					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cyl	IN	0 ... 7	0 ... 7	1	-
Cylinder number on which the ignition angle is applied					
Iga_igc_h_rng	IN	FA60..5A0h	-90..+90	0.0625	°CRK
Ignition angle (high range) to be applied on cylinder Cyl					

ACTION_INFR_SetIgnDwell (IN <Cyl>, In <Td_fac_min>, IN <Td_igc>, IN <Td_fac_max>)					
Sets the dwell time duration and the factors for maximal and minimum dwell time					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cyl	IN	0 ... 7	0 ... 7	1	-
Cylinder number on which the dwell time is set					
Td_fac_min	IN	1 ... FF	1/128 ... 255/128	1/128	-
Factor to calculate minimum dwell time					
Td_igc	IN	0 ... FFFF	0 ... 262.14	0.004	ms
Dwell time duration					
Td_fac_max	IN	1 ... FF	1/128 ... 255/128	1/128	-
Factor to calculate maximum dwell time					

ACTION_INFR_SetIgcDwellTest (IN <Cyl>, IN <Td_igc>)					
Sets the dwell time duration for cylinder Cyl and activate coil charging accordingly					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cyl	IN	0 ... 7	0 ... 7	1	-
Cylinder number on which the dwell time is set					
Td_igc	IN	0 ... FFFF	0 ... 262.14	0.004	ms
Dwell time duration					

ACTION_INFR_SetIgnMpl (IN <Cyl>, IN <Td_mpl1>, IN <Td_mpl>, IN <Td_mpl1_dly>, IN <Td_mpl_dly>, IN <Ign_mpl_nr>)					
Sets dwell times, dwell time interruptions and number of ignition sparks of multiple sparks					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cyl	IN	0 ... 7	0 ... 7	1	-
Cylinder number on which the dwell time is set					
Td_mpl1	IN	0 ... FFFF	0 ... 262.14	0.004	ms
Dwell time for 1st MPL (first spark after main spark) spark					
Td_mpl	IN	0 ... FFFF	0 ... 262.14	0.004	ms
Dwell time for 2nd and others multiple spark					
Td_mpl1_dly	IN	0 ... FFFF	0 ... 262.14	0.004	ms
Dwell time interruption between main spark and 1st MPL spark					
Td_mpl_dly	IN	0 ... FFFF	0 ... 262.14	0.004	ms
Dwell time interruption between MPL sparks (except between main and 1st MPL spark)					
Ign_mpl_nr	IN	0 ... FF	0 ... 255	1	-
Number of ignition sparks					

ACTION_INFR_SetIgnCtl (IN <Ign_ctl_mod>)					
Sets the ignition in full static or half static mode					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Ign_ctl_mod	IN	1 ... 2	1 HALF_STATIC 2 FULL_STATIC		
Control mode for ignition are : HALF_STATIC or FULL_STATIC					

FUNCTION DESCRIPTION:

General information:

Requirements for ACTION_INFR_SetIgnEnable:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cyl	1	-	1		Index range: 0 ... NC_CYL_NR-1
Inh_igc	1	-	1		0: ignition inhibited 1 : ignition enabled

Coincidence requirements:

The decision to inhibit an ignition stage is taken for each output before the turn on at TD_ON[x]. This means that no Disable / cut off can be set on a coil while charging.

Management of Ignition Cut-Off: Enable / Disable mode of Ignition Power stages

For normal Dwell Time and coil charging management:

a. No Ignition cut is allowed on a currently charging coil, *ie.* no Disable Command of a power stage is considered after the TD_ON event
   Nevertheless, if the Disable request still be present, it will be applied on the next coming to fire cylinder
b. Enable command request coming to a power stage in Disable mode will be only performed if TD_IGC[x] time to IGA_IGC point can be respected/realised.
   In other case, this Enable request will be performed only on the next engine cycle (720°CRK or 360°CRK depending of the Ignition Coil mode - full or half static) if it still be present
c. No Re-Enable command can - neither will - be done on a power stage which is yet already in Enable mode

In Multiple Spark phase:

d. As for the nominal phase, no Ignition-Cut can be allowed on a currently charging coil even in Multiple Spark phase, *ie.* no Disable Command of power stage is considered after during the TD_MPL charge: if one TD_MPL is on the way, no Ignition cut is possible.
   See dwell comments
   The Disable request could be done/happened anywhere but it will only be executed, can only occur and be taken into account only at the end of the charging of this TD_MPL

Requirements for ACTION INFR_SetIgnAngle:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cyl	1	-	1		Index range: 0 ... NC_CYL_NR-1
Iga_igc_h_rng	0.0625°CRK		0.0625°CRK		

Diagnosis:

Coincidence requirements:

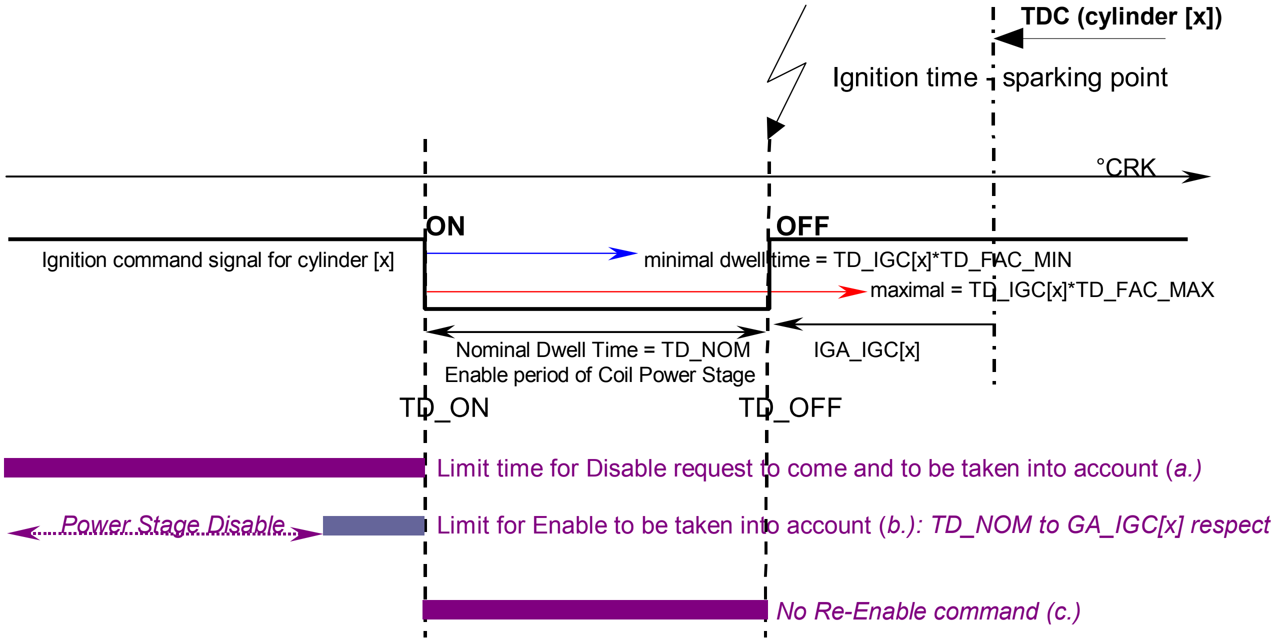
Requirements for ACTION INFR_SetIgnDwell:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cyl	1	-	1		Index range: 0 ... NC_CYL_NR-1
Td_fac_min			1/128		
Td_igc			0.004ms		
Td_fac_max			1/128		

Diagnosis:

Coincidence requirements:

The normal request dwell time to be programmed and applied for charging the ignition coil is and has to be TD_IGC[x]
The coil starts charging with the target of achieving TD_IGC[x] at the IGA_IGC[x] angle. An update occuring for the Ignition Spark Angle will be taken into account as long as TD_MIN is not reached.
This Ignition Spark Angle change obtained when reaching TD_MIN will be taken into account and respected by a corresponding change of the coil charging time (the Dwell Time).
But, such a change of the Dwell Time to be thus and then applied can only be processed within the limits defined by TD MIN and TD MAX limits excursion range (*see drawing here after*)
For the definition and calculation of those TD MIN and TD MAX:



Note: IGA_IGC[x] is a signed value. Negative angles are for sparks generated after TDC; positive angles are for sparks generated before TDC.

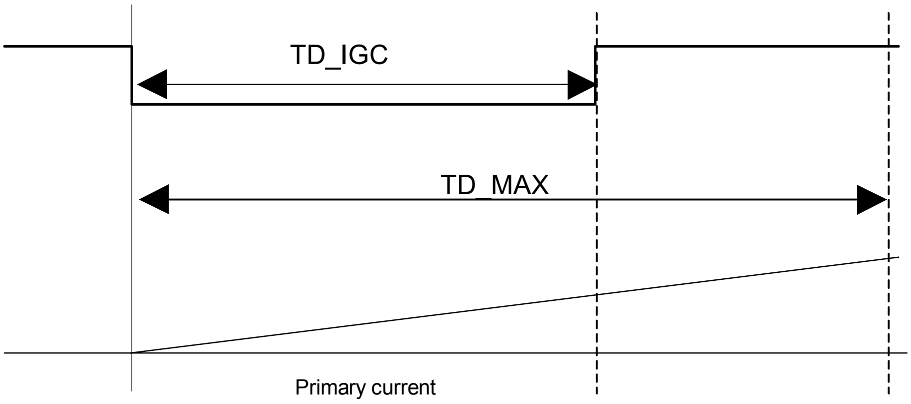
Calculations after starting the dwell time output at TD_ON

For the ignition time there are three main requests for the dwell time output.

1. A minimum dwell time TD_IGC[x] * TD_FAC_MIN has to be guaranteed to ensure ignition.
2. The dwell time has to be limited by a maximum dwell time which is defined as a TD_IGC[x] * TD_FAC_MAX.

   To prevent all cases of big negative engine speed gradients which could destroy the ignition system, by too long conduction between conduction angle calculation and spark, a timer limits the resulting conduction time (ignition power stage protection against high current).

   The ignition output is turned off immediately at TD_IGC[x] * TD_FAC_MAX (IGA_IGC[x] could not be respected).
3. In these limits (TD_IGC[x] * TD_FAC_MIN, TD_IGC[x] * TD_FAC_MAX) we have an absolute ignition angle priority.



Requirements for ACTION_INFR_SetIgcDwellTest:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cyl	1	-	1		Index range: 0 ... NC_CYL_NR-1
Td_igc			0.004ms		

Diagnosis:

Coincidence requirements:

The normal request dwell time to be programmed and applied for charging the ignition coil is and has to be TD_IGC[x]
On Actuator test request activation, the coil starts charging with the target of achieving TD_IGC[x]

The coils start charging upon actuator test request pattern with the dwell time defined.

The pulse is generated once and immediately after calling the action. The Dwell time could not be updated during the pulse generation, excepting if it is set to 0. in this case, the current pulse should be stopped immediately.

Detection of SCP failure must be enabled

In case of overlapping dwell time, diagnosis information can be false then in case of dwell time overlap there is a risk to destroy the IGBT.

Requirements for ACTION_INFR_SetIgnCtl:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Ign_ctl_mod					

Diagnosis:

Coincidence requirements:

This action musn't be performed after the synchronization.

Requirements for actions linked to multiple spark mode:

Requirements for ACTION_INFR_SetIgnMpl

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cyl	1	-	1		Index range: 0 ... NC_CYL_NR-1
Td_mpl1			0.004ms		
Td_mpl			0.004ms		
Td_mpl1_dly			0.004ms		
Td_mpl_dly			0.004ms		
Ign_mpl_nr			1		

Diagnosis:

Coincidence requirements:

The parameter Cyl determines the cylinder number on which we set the following attributes.

The values TD_MPL1 and TD_MPL determines the conduction time of the ignition coil: TD_MPL1 for the 1st MPL spark (1st spark after main spark) and TD_MPL for the other sparks (2nd MPL and following sparks)

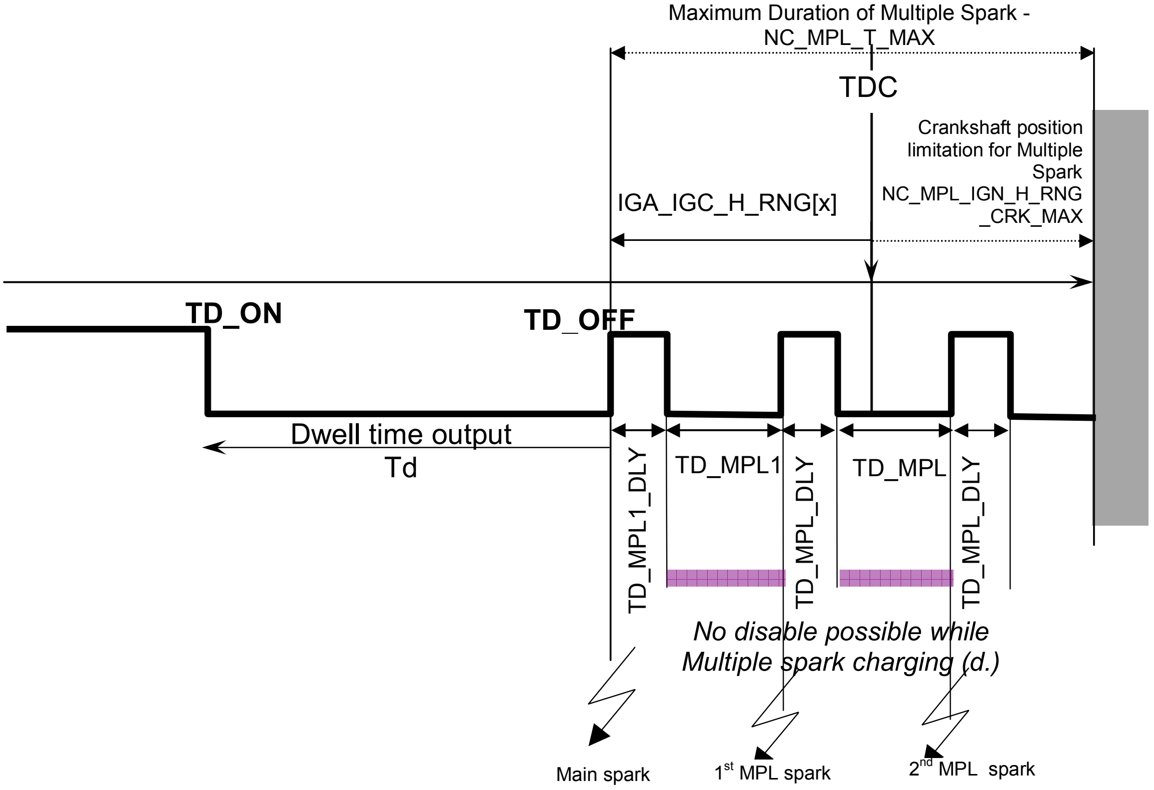
The time delays TD_MPL1_DLY and TD_MPL_DLY specifies a time-out when the ignition coil is turned on for the next spark: TD_MPL1_DLY is the time between main spark and 1st MPL; TD_MPL_DLY is the time between other sparks.

The number of successive sparks is defined as IGN_MPL_NR and checked at every TD_OFF.

The absolute duration of multiple sparks is limited to a maximum time defined as NC_MPL_T_MAX.

The maximum ignition angle to start a TD_MPL is defined as NC_MPL_IGN_H_RNG_CRK_MAX after TDC.

Description:



Requirements description for overlapping prevention in case of multiple spark activation:

Overlapping in between the different cylinders affected to the same HW component could lead to false diagnosis detection.

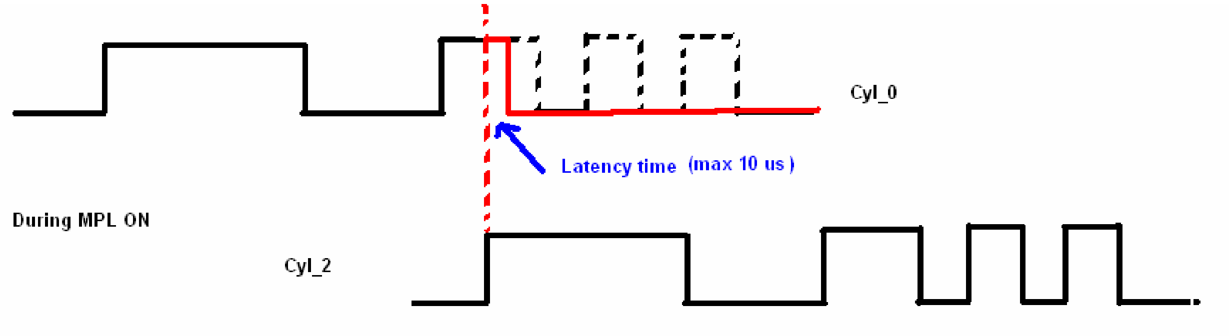
In order to avoid this behavior the following implementation should be made inside the software.

At each dwell ON event, the curent cylinder read the state of the others cylinders in the same IC component.
If the state of others cylinders are in :

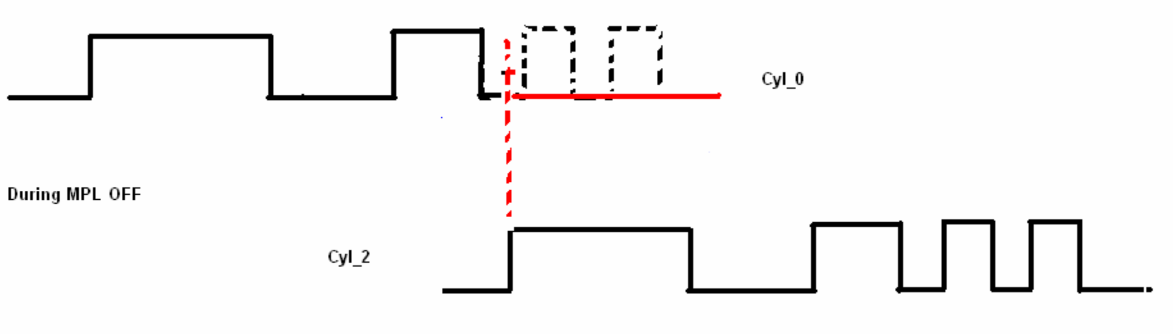
1. **Multiple spark mode** : multiple sparks on these cylinders will be switched off as soon as possible

The 2 following drawings describes the 2 cases which could occur :

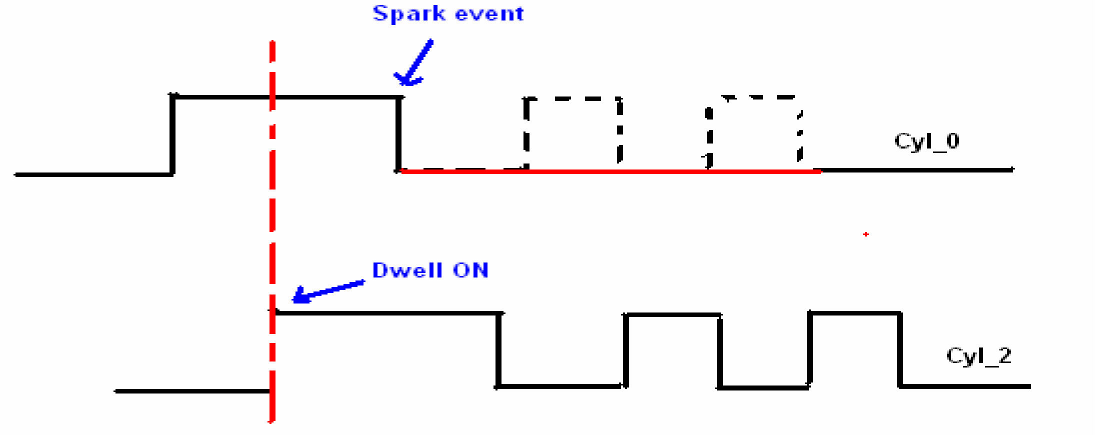
- Cut-off requested during MPL_ON event of cylinder (one multiple spark is currently in charge)
  In this case the latency time to cut-off the MPL in progress will be 10µs in worst case.



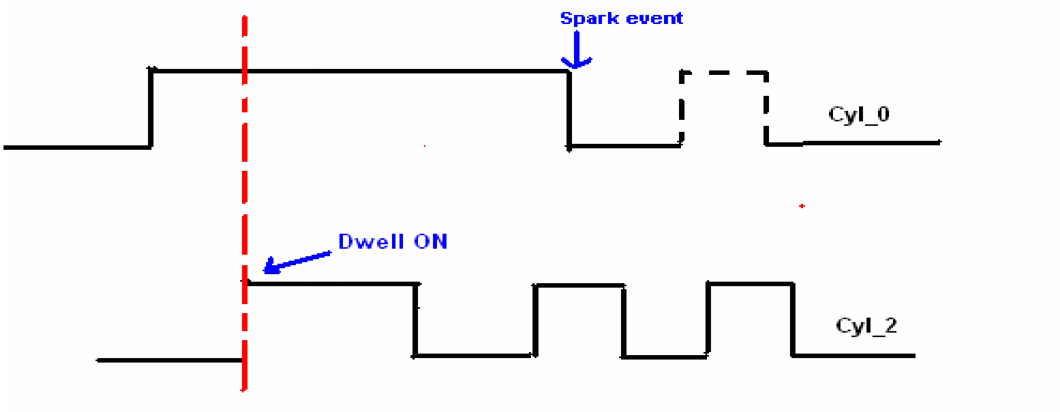
- Cut-off requested during MPL_OFF event of cylinder (in between 2 multiple sparks)
  In this case the following MPL won't be applied at all.



2. Main dwell : the multiple sparks of those cylinders will be cut-off to prevent overlapping.



Note : This implementation only prevents overlapping in between multiple sparks and main spark and in between multiple sparks. Prevention in between main sparks is not addressed because it is not realistic unless at very low battery voltage at high engine speed. (these cases are considered as a limp home mode)
In addition case decribed below could not happen and thus will not be covered by this implementation:


Spark event
Cyl_0
Dwell ON
Cyl_2

38.7 IGRE – Requirements to infrastucture interface

Input data:

NC_INI_CTR_DEAC	NC_T_MIN_SCP		

Detailed input data definition:

Name	Hex. limits	Phys. limits	Resol.	Unit
NC_INI_CTR_DEAC	0...FFFFH	0...262.14	0.004ms	ms
Time delay after IGBT response before switching OFF the IGBT – typical value 600µs				
NC_T_MIN_SCP	0...FFFFH	0...262.14	0.004ms	ms
Minimum time to detect SCB with current feedback flag				

Export actions:

ACTION_INFR_GetIgnDwellAd(IN <Chan>, IN <Cyl>, OUT <T_flag>, OUT <Td>)
Gets the duration of the flag (time in between current threshold and end of the main pulse) and the effective main pulse applied
ACTION_INFR_GetIgnScpDiag(IN <Cyl>, OUT<Diag_scp>)
Gets the diagnosis output for SCP detection with usage of ATM46 or ATIC71
ACTION_INFR_GetIgnScgDiag(IN <Cyl>, IN <Min_burn_time>, OUT <V_dur_igc>, OUT<Diag_scg>)
Gets the diagnosis output for SCG or OL detection and the burn time duration with usage of primary voltage feedback
ACTION_INFR_SetIgnConfigDiag(IN <Chan>)
Sets the channel used for SCP detection

Description for actions:

ACTION_INFR_GetIgnDwellAd(IN <Chan>, IN <Cyl>, OUT <T_flag>, OUT <Td>)					
Gets the duration of the flag (time in between current threshold and end of main pulse) and the effective main pulse applied					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Chan	IN	0 ... 1	0 ... 1	1	-
Channel number on which we want to obtain the time measurement for dwell time control 0: CURR_TH1 1: CURR_TH2					
Cyl	IN	0 ... 7	0 ... 7	1	-
Cylinder number on which we want to obtain the time measurement for dwell time control					
T_flag	OUT	0 ... FFFF	0 ... 262.14	0.004	-
Integrated time spent above current threshold during the last main dwell pulse (accuracy of 50µs due to SW interrupt latency)					
Td	OUT	0 ... FFFF	0 ... 262.14	0.004	-
Duration of main pulse which has been applied					

ACTION_INFR_GetIgnScpDiag(IN <Cyl>,OUT<Diag_scp>)

Gets the diagnosis output for SCP detection with usage of ATM46 or ATIC71

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Cyl	IN	0 ... 7	0 ... 7	1	-
Cylinder number on which the diagnosis for SCP is get					
Diag_scp	OUT	0 ... 1	0 ... 1	1	-
Diagnostic of primary short circuit to battery 0 : no SCP detected from last invocation 1 : SCP detected from last invocation					

ACTION_INFR_GetIgnScgDiag(IN <Cyl>, IN <min_burn_time>, OUT <V_dur_igc>, OUT<Diag_scg>)

Gets the diagnosis output for SCG or OL detection and the burn time duration with usage of primary voltage feedback

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Cyl	IN	0 ... 7	0 ... 7	1	-
Cylinder number on which the diagnosis for SCG or OL is get					
Min_burn_time	IN	0 ... FFFF	0 ... 262.14	0.004	ms
Minimum duration for burn time to detect ignition errors					
V_dur_igc	OUT	0 ... FFFF	0 ... 262.14	0.004	ms
Burn time duration for diagnosis validation					
Diag_scg	OUT	0 ... 1	0 ... 1	1	-
Diagnostic for primary short circuit to ground or open load 0 : no SCG detected based on last burntime measured on this cylinder 1 : SCG detected based on last burntime measured on this cylinder					

ACTION_INFR_SetIgnConfigDiag(IN <Chan>)

Sets the channel used for SCP detection

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Chan	IN	0...1	0 ... 1	1	-
Channel number on which the SCP diagnosis will be done 0 : CURR_TH1 1 : CURR_TH2					

FUNCTION DESCRIPTION:

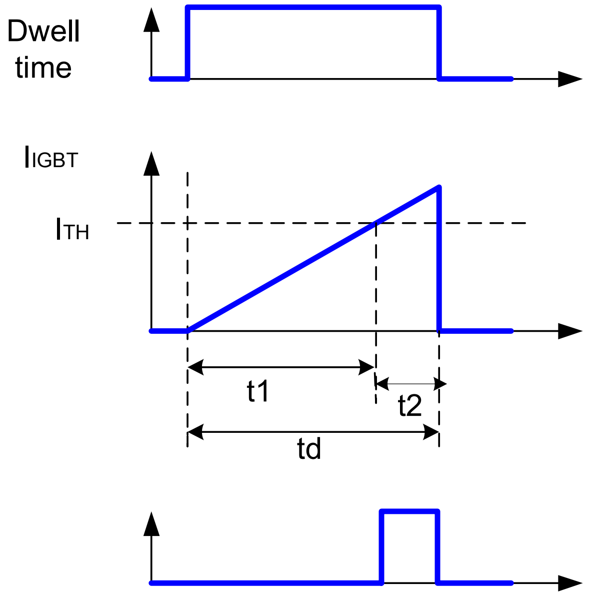
General information:

Requirements for ACTION_INFR_GetIgnDwellAd:

Data acquisition

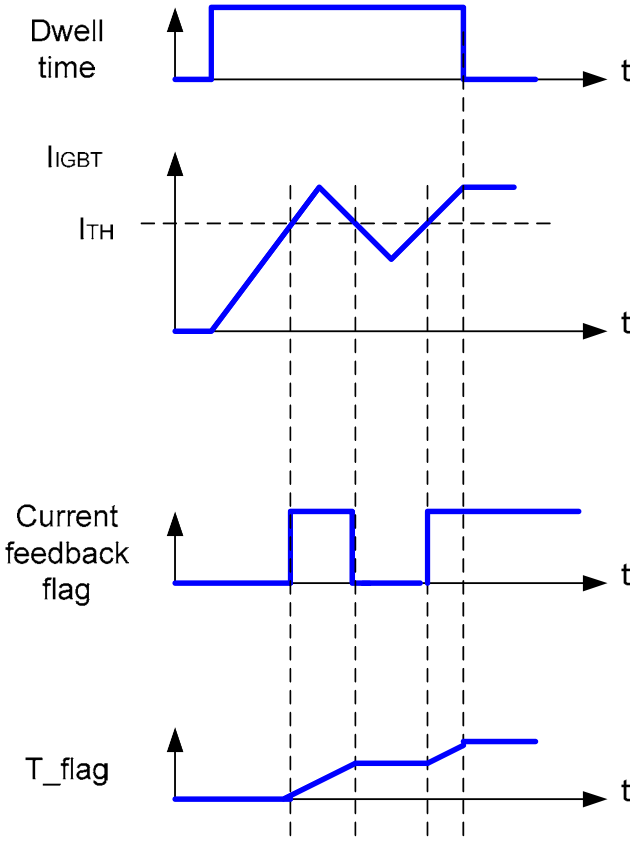
Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment
Chan	1	-	1		
Cyl	1	-	1		Index range: 0 ... NC_CYL_NR-1
T_flag	1	-	0.004ms		Accuracy of 50µs due to SW interrupt latency
Td	1	-	0.004ms		

Coincidence requirements:



T_flag corresponds to the value t2 of the above graph.
Td corresponds to td.

In case of oscillation of I_{IGBT} around I_{th}, T_flag is the result of the time spent above the threshold I_{th}



Note that: * in case of overlapping of dwell time periods T_flag value cannot be guaranty due to HW design and limitation.

* If Dwell Time periods are overlapping, only the most recent one that has the latest started a coil charging (*last and most recent TD ON event, whatever it is*) will trigger the activation of this integration time function.

With overlapping Dwell Time periods, there is thus a risk to get a wrong T_flag value.

Dwell Times overlapping conditions may occur, in normal running, with too long settings of Dwell time in half and full static ignition modes.

Dwell Times overlapping conditions will occur, in Ignition Limp Home running (*ie. ½ static mode usage for coils charging*).

This interface will return maximum value for T_flag in case :

- SCP has been detected during main pulse
- No dwell has still been performed
- Channel requested is not activated

When Applicative software has read the T_flag value, this value is reseted to maximum value by BSW. This functionning allows safety mode in case applicative software reads two times the same T_flag value without acquisition of new current feedback flag value by BSW.

Requirements for ACTION_INFR_GetIgnScpDiag:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cyl	1	-	1		Index range: 0 ... NC_CYL_NR-1
Diag_scp			1		

Diagnosis:

Coincidence requirements:

SCP diagnosis by primary current feedback flag

1. A Dwell Time Pulse is considered as a main or a multiple dwell pulse. Thus there is no distinction in the treatment of “Dwell Time On Events” from main and multiple pulses.
2. Consequently, there is no distinction in the treatment of “Spark Events” from main and/or multiple Dwell Time pulses.
3. A “Dwell Time On Event” defines the start time of coil charging. The resolution and precision of this time is defined by the corresponding ignition channel
4. A “Spark Event” or “Dwell Time Off Event” defines the end time of coil charging. The resolution and precision of this time is defined by the corresponding ignition channel.

Each time a “Dwell Time On Event” occurs, the time during which the IGBT_DIAG Pin indicates a SCP condition has to be integrated within a time window (*defined below*). If this integrated time inside this window exceeds NC_INI_CTR_DEAC, the cylinder that has produced the most recent Dwell Time On event must be switch off as fast as possible for the current ignition cycle. The time window begins at any “Dwell Time On Event” and ends if either one of the following criteria is met:

cr.a) another “Dwell Time On Event” from any Cylinder occurs. This restarts the time window and thus the integration of the time for which IGBT_DIAG indicates SCP conditions

cr.b) a “Spark Event” of the same Cylinder that started charging the coil occurs. “Spark Events” of any other Cylinder are ignored.

cr.c) the integrated time for which the IGBT_DIAG pin indicates a SCP condition exceeds NC_INI_CTR_DEAC

cr.d) the time period back to the most recent “Dwell Time On Event” exceeds NC_IGBT_CUT_OFF_T (primary voltage feedback) or NC_T_MIN_SCP (primary current feedback).

Note that: * in case of overlapping of dwell time periods no short circuit detection can be guaranty due to HW design and limitation.

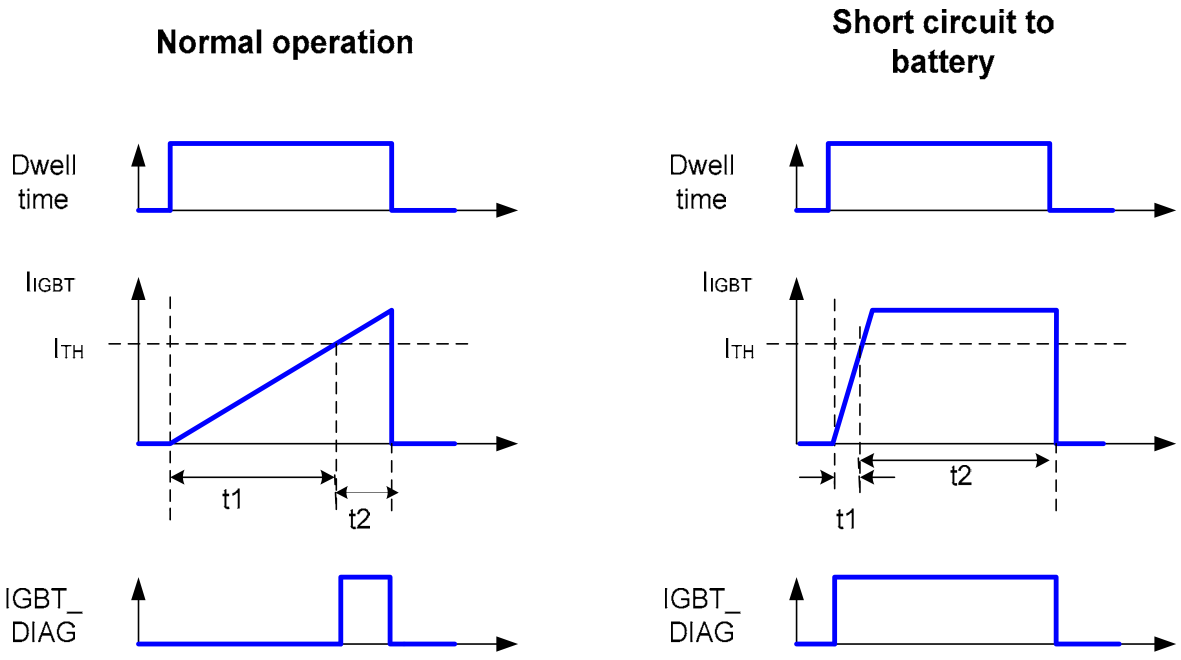
* If Dwell Time periods are overlapping, only the most recent one that has the latest started a coil charging (*last and most recent TD ON event, whatever it is*) will trigger the activation of this protection (BSW service) function. Hence, the last occurring TD ON event will define the coil to be considered by this function and thus diagnosed and protected (cut off if needed, when SCP failure appears)

With overlapping Dwell Time periods, there is thus a risk to do not diagnose (and switch off) the correct IGBT in real failure conditions

Dwell Times overlapping conditions may occur, in normal running, with too long settings of Dwell time in half and full static ignition modes.

Dwell Times overlapping conditions will occur, in Ignition Limp Home running (*ie. ½ static mode usage for coils charging*).

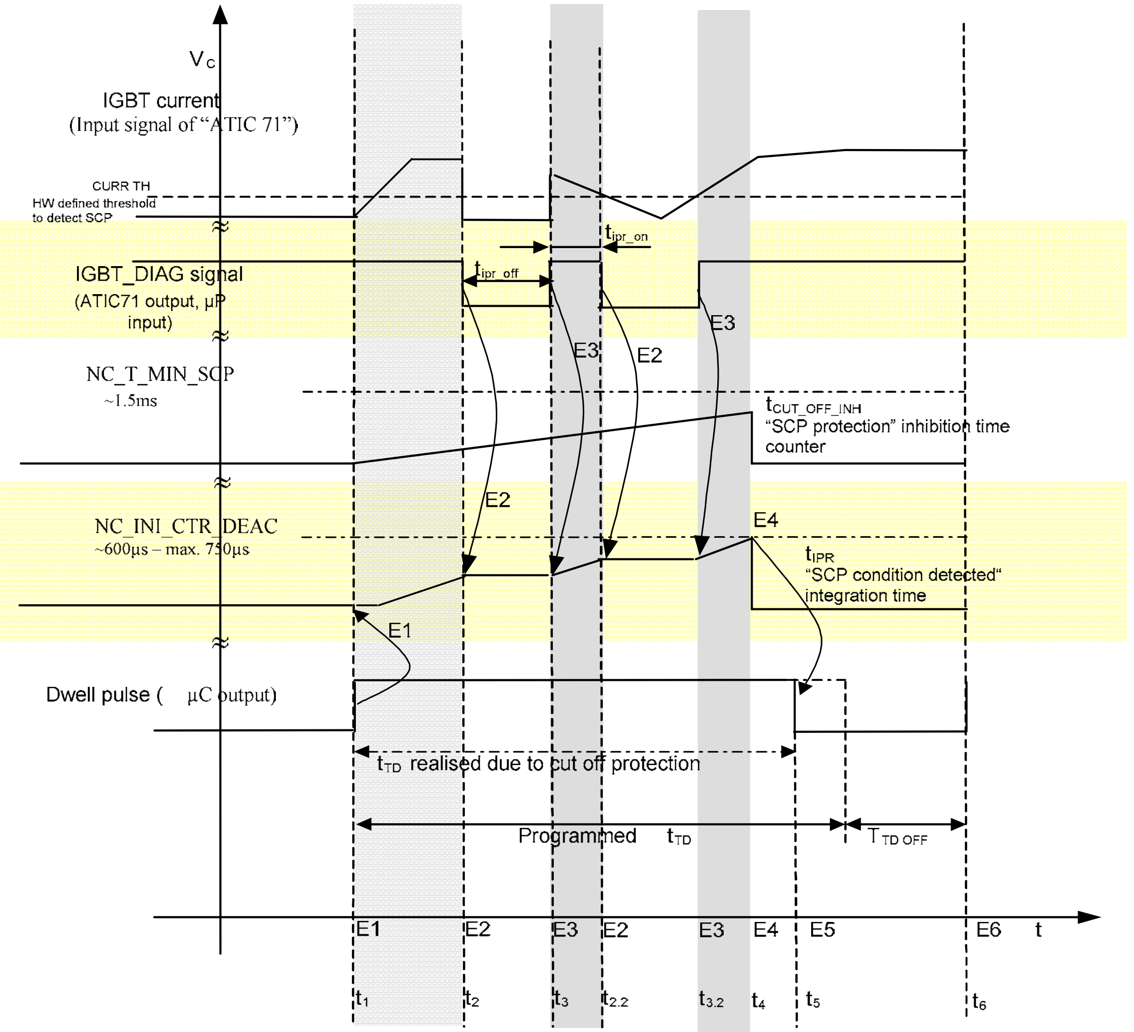
One ATM 46/ATIC71 could handle only 4 ignition coils. If there are more than 4 ignition coils 2 ASIC have to be used. To reduce the possibility of overlapping we have to use them in a crossed order.



Each time a “Dwell Time On Event” occurs, the time during which the IGBT_DIAG(resp. DWELL_DIAG) Pin is high has to be integrated. If this integrated time inside the main dwell pulse exceeds NC_INI_CTR_DEAC, an interruption is generated.

Within this interrupt routine t1+t2 is compared to NC_T_MIN_SCP.

If t1+t2 < NC_T_MIN_SCP the failure is detected and the cylinder that has produced the most recent Dwell Time On event must be switch off as fast as possible for the current ignition cycle.



A dwell pulse is considered as a main or multiple dwell pulse, no distinctions are done as both can be critics in case of SCP failure.

When the IGBT current is higher than the threshold (CURR_TH), it indicates a SCP condition on IGBT. In this case, the IGBT_DIAG signal from the ASIC is high (in SCP level).

a) Events and Actions definition

E1: Dwell Time On Event – Start the charging of the coil
When this event occurs at t_1:
- Clear the SCP pulse periods measured up to now (*ie. the "SCP condition detected" integration time*)
– start / enable to integrate the time for which the IGBT_DIAG Pin indicates a SCP condition

E2: IGBT output without SCP condition (t_2) or transition to No SCP ($t_{2.2}$ - *SCP disappears*)
When E2 occurs, the *"SCP condition detected" integration time* is stopped (*no or no more SCP conditions seen*)

E3: IGBT output in SCP conditions (t_3) or transition to SCP ($t_{3.2}$ - *SCP occurs again*)
When E3 occurs, the *"SCP condition detected" integration time* is started or re-started (*SCP conditions seen or re-seen again*)

E4: Cut Event from the IGBT SCP Protection function
E4 will not occur in normal condition when the coil encounters no SCP conditions (no or too short E3 is met, only E2)

t_{IPR} designates the *“SCP condition detected“ integration time* acquired (*see after*)
t_{IPR} is permanently checked when this protection function is active

IGBT Cut Off in case of SCP detection validated:

If t_{IPR} > NC_INI_CTR_DEAC

And $(t_4 - t_1)$ < NC_T_MIN_SCP

Then Switch off the ignition channel that has produced the most recent Dwell Time On Event (*most recent or latest E1 event*) as fast as possible

Dwell Time applied in this case will be “t_{TD} realized due to cut off protection” function and not the “Programmed t_{TD}”

Else No action on ignition channel

EndIf

E5: Spark Event at the cutting of Ignition Coil charging
E5 occurs at t_5. E5 / t_5 can be either:
- the normal “*Programmed t_{TD}*” if this SCP protection function has not been reached (no E4) or
- the Coil charging Time corresponding to *“t_{TD} realized due to cut off protection”* function if this SCP protection function has been reached (at E4)

When E5 occurs: the *“SCP condition detected“ integration time* (t_{IPR}) is stopped

E6: Identical to Dwell On Event E1 for the next following coil coming to charging mode on the same ASIC diagnostic line

b) Times definition

t_{ipr_on} = time period that indicates a SCP conditions (*ie. I_{IGBT} > CURR_TH, the IGBT_DIAG signal from the ASIC indicates the SCP*). During this time, the *“SCP condition detected” integration time* – t_{IPR} – is enabled.

t_{ipr_off} = time period where no SCP conditions exist (*ie. I_{IGBT} < CURR_TH*). This can happen also in between IGBT SCP diagnosis pulses detection

t_{TD} = time period of any dwell pulse. It correspond normally to the programmed Dwell Time period for charging the Ignition coil. Here, it can be shortened (*cut off of the Coil Charging*) by the IGBT SCP protection function.

$t_{TD\ OFF}$ =time period between any successive dwell pulses

$t_{CUT\_OFF\_INH}$ = For the Ignition Coil under SCP Protection diagnostic, the “Cut Off” functionality is deactivated after a time longer than
$t_{CUT\_OFF\_INH}$ > NC_T_MIN_SCP
If such a time out occurs, the dwell time is not interrupted cause it corresponds to case where IGBT handles over a situation with a high gradient of battery tension and not a short circuit to battery

t_{IPR} = *“SCP condition detected“ integration time* acquired
This Pulse Measurement Method has to be active – which is equivalent to: the IGBT_DIAG input to the µP at SCP level has to be measured – between t_0 and either criteria cr.a), cr.b), cr.c) or cr.d) is met **whichever** comes first.

Warning: due different HW and/or µP architecture and implementation constraint limitations

- Every event E1 to E6 must be processed within a Maximum tolerable time delay smaller than 50µs
- The Dwell Time and Dwell Charging period succession should be limited to:
  100µs ≤ t_{TD} ≤ 60ms and 100µs ≤ $t_{TD\ OFF}$
- Switch off the ignition channel that has produced the most recent Dwell Time On Event should only occur within 50µs after SCP detection confirmation
- IGBT_DIAG pulses shorter than 50µs have a risk of not being processed

Requirements for ACTION_INFR_GetIgnScgDiag:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cyl	1	-	1		Index range: 0 ... NC_CYL_NR-1
Min_burn_time			0.004ms		
V_dur_igc			0.004ms		
Diag_scg			1		

Diagnosis:

Min burn time (V_DUR_IGC) is set to 0 at reset.

So we have open load detection on the first segments until the burn time has been measured and read from applicative software.

The diagnosis is done by comparing V_dur_igc to Min_burn_time. If V_dur_igc < Min_burn_time then the failure is detected.

In case of SCP detection (IGBT cut-off) the burn time measurement must be disabled for the concerned cylinder. The output V_DUR_IGC will be thus in this case the value measured at the previous cycle.

Coincidence requirements:

For the Spark Burn Time Duration measurement, the functionality is:

- To ensure a correct measurement in case of spark blow out, only the Active Level of Burn Time measurement signal from the ATM46 (IGN_DIAG) is integrated in the time window defined by two consecutive main spark events on the same ASIC
- Active Level of Burn Time measurement signal can be “High” or “Low” depending on HW configuration
- Minimum length of Active Level pulses to be detected is 50µs and Minimum inter-space between Active Level pulses to be detected is 50µs. Consequently, IGN_DIAG pulses shorter than 50µs have a risk of not being processed.

Sparking events from multiple spark pulses should not be taken into account for burn time measurement (see timing diagram in case of multiple spark).

At the end of the time measurement window started by cylinder[x], the measured burning duration is read out and written into the variable V_DUR_IGC[x] for the corresponding logical cylinder [x]. The time for Burn Time measurement is then cleared and prepared for the next measurement.

Consider that only one IGN_DIAG signal exists (*this IGN_DIAG signal represents a wired “OR” of all the presents coils channels*) which will limit the measurement range.

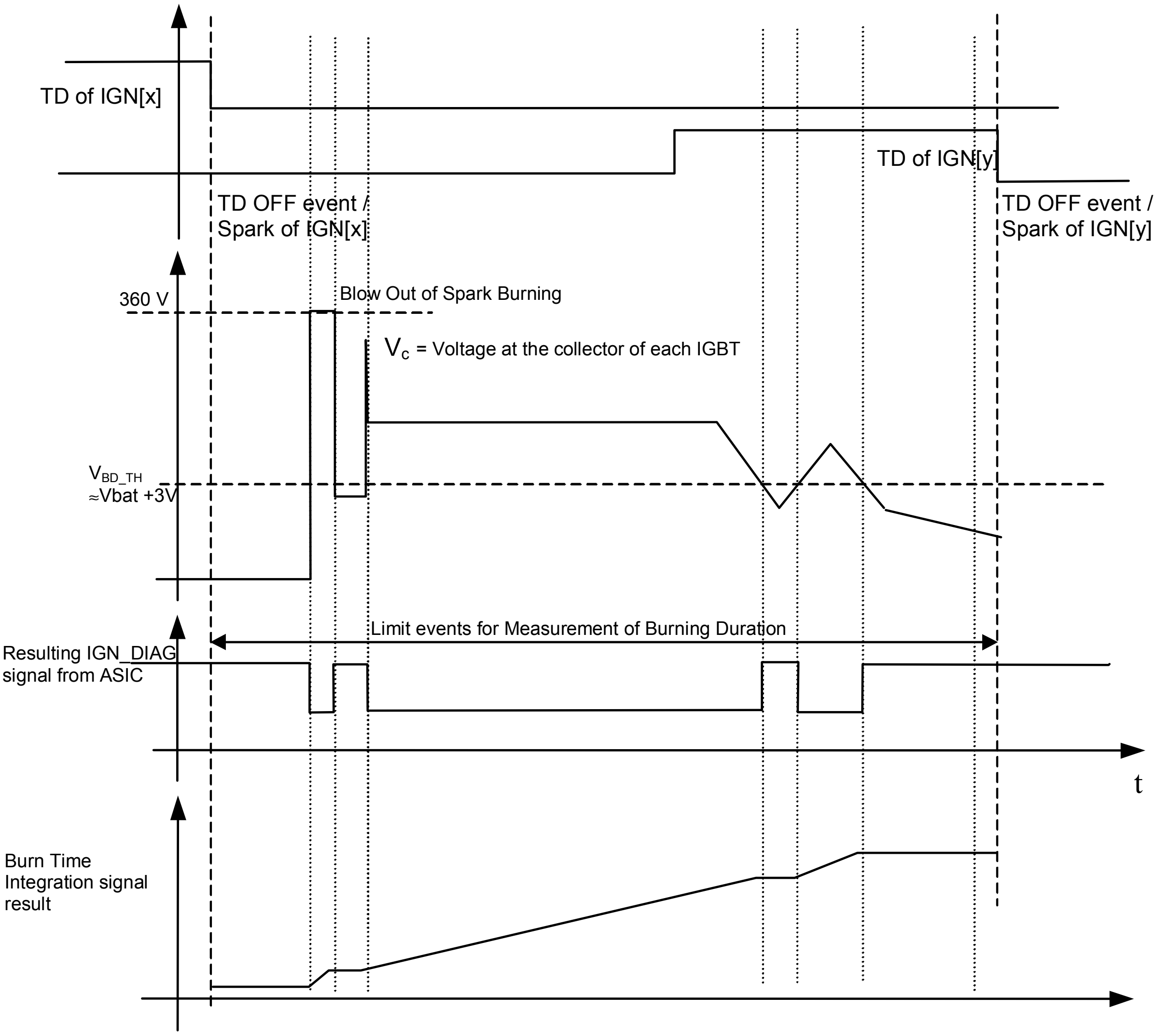
In Limp Home and ½ static modes, 2 main spark events – on one ATM46 diagnostic signal line – occur at the same time. This means that the time windows for measurement of (*at least*) two cylinders overlap. In such cases, due to the “OR” wiring on the signal information,

only the cylinder on which the most recent main spark event (*TD OFF*) is produced will define and will be identified for the time window measurement.
Therefore, the reported burning duration in the interface variable V_DUR_IGC[x] may not correspond to right real burning cylinder [x] in case of Limp Home mode and ½ static coils.

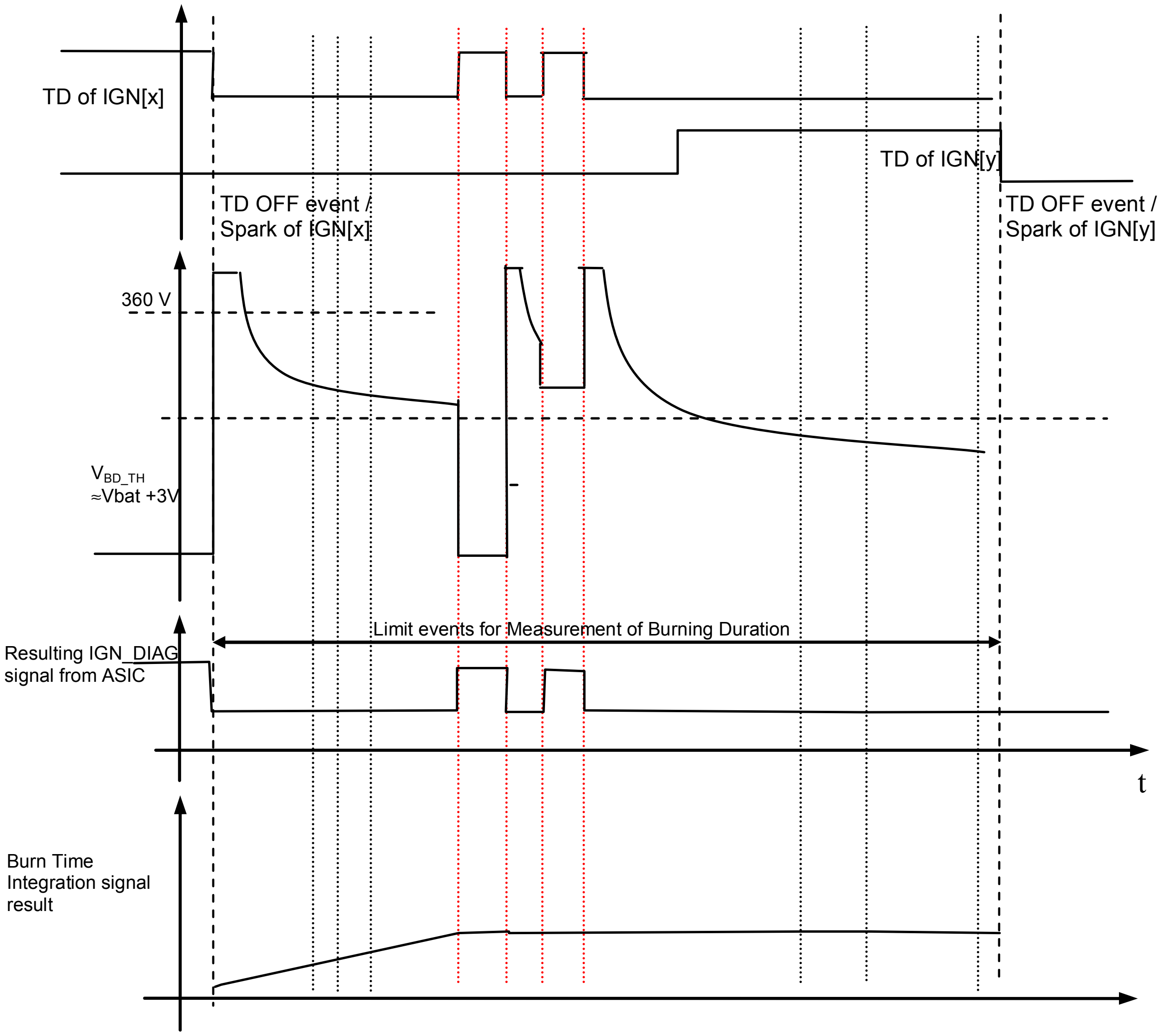
Each ATM46 could analyze 4 ignition coils. If diagnostics for more the than 4 coils are necessary, a parallel ATM46 have to be used.

Timing diagram:

In case of no multiple spark



In case of multiple spark



Requirements for ACTION INFR_SetIgnConfigDiag:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Chan	1	-	1		

Diagnosis:

Coincidence requirements:

This action is sent to configure the channel on which will be done the SCP diagnosis.This action must be sent before the first dwell has occured in order to start the diagnosis on the first dwell generated.

38.8 AGGR IGRE adaptation

38.8.1 Reset of New Ignition Angle Range to old in Integration process

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_AV_MV	O/V	0...FFH	-35.625...60	0.375	°CRK
Mean value of actual ignition angle					
IGA_IGC[NC_CYL_NR]	O/V	0...FFH	-35.625...60	0.375	°CRK
Applied Ignition angle					
IGA_AV[NC_CYL_NR]	O/V	0...FFH	-35.625...60	0.375	°CRK
Ignition Angle Applied on CYL					

Input data:

IGA_IGC_H_RNG[NC_CYL_NR]	IGA_AV_MV_H_RNG	LV_ERR_IGC_SCP[NC_CYL_NR]	LV_ERR_IGC_SCG[NC_CYL_NR]

Information:

Function is applied for Integration of IGRE3.0.0 to keep variables in old ignition angle range. With update to IGRE3.1.0, only new variables will be used. This, except in a few modules that will not be touched due to affection of different projects

Formula section:

IGA_AV_MV = IGA_AV_MV_H_RNG

IGA_IGC[NC_CYL_NR] = IGA_IGC_H_RNG[NC_CYL_NR]

IGA_AV[NC_CYL_NR] = IGA_IGC[NC_CYL_NR]

38.9 Dwell time control (open loop)

38.9.1 Dwell time calculation

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
TD	O/V	0...FFFFH	0....262,14	0,004	ms
Dwell time					

Input data:

TD_FAC	VB	N_32	LV_HOM_RUN
LV_TD_AD_H	C_TD_S_AS	IP_TD_S	

FUNCTION DESCRIPTION:

The dwell time raw value calculation is based on the simulation done by the HW-group. At any time an adaptation or correction factor has to be multiplied or added in this chapter.

The dwell time open-loop control value is stored in terms of a battery voltage and engine speed related characteristics.

Application conditions:

Initialisation:	TD = C_TD_INI		*at reset*
Recurrence:	10 ms		
Activation:	*at every engine state*	**if**	LV_HOM_RUN = 1

Formula section

```
If LV_TD_AD_H == 1
Then TD = IP_TD_S(VB, N_32) * TD_FAC + C_TD_S_AS
Else TD = IP_TD(VB, N_32) * TD_FAC + C_TD_AS
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
IP_TD	8*8	0...FFFFH	0...262,14	0,004	ms
LDPM_VB_1_IGRE	8	0…FFH	0….26	0,102	V
LDPM_N_32_1_IGRE	8	0…FFH	0….8160	32	rpm
Dwell time correction versus battery voltage and engine speed					
C_TD_AS	1	F800...7F0H	-8,192..8,128	0,004	ms
Dwell time offset usual value: 0 ms					
C_TD_INI	1	0...FFFFH	0...262,14	0,004	ms
initial value for dwell time control (typ. Value 3ms for standart coils and <2ms for pencil coils)					

38.10 Dwell time factor calculation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TD_FAC	V/O	0...FFH	0...1.99218	0.0078125	[-]
Basic dwell time factor					
TD_FAC_TCO	V	1...FFH	0.00781... 1.99218	0.0078125	[-]
Adjustment factor with TCO dependence					

Input data:

TCO			

FUNCTION DESCRIPTION:

General information:

The output data could be only TD_FAC. All other factors should be only visible values.

The dwell time factor regroups all adjustment factors as coil temperature factor or a boost factor for start. Here the boost factor is not taken into account as we use for MSD8X project an adaptation of Dwell Time at hight current target at start.

Initialisation : at reset

TD_FAC = 1

TD_FAC_TCO = 1

Recurrence: 10 ms

Formula section:

TD_FAC_TCO = IP_TD_FAC_TCO

TD_FAC = TD_FAC_TCO

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
IP_TD_FAC_TCO	8	1...FFH	0.00781... 1.99218	0.0078125	[-]
LDPM_TCO_TD_FAC	8	0...FEH	-48...142.5	0.75	[°C]
Adjustment Factor for dwell time					

38.11 Dwell time period for stratified mode (open loop)

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
TD_S	O/V	0...FFFFH	0.....262,14	0,004	ms
Dwell time for stratified mode					

Input data:

VB	N_32	LV_S_RUN	TD_FAC
LDPM_VB_1_IGRE	LDPM_N_32_1_IGRE		

FUNCTION DESCRIPTION:

The following function is similar to the dwell time calculation for homogeneous combustion. Only the calibration data should be adapted to stratified combustion.

The dwell time value calculation is based on the simulation done by the HW-group. The dwell time open-loop control value is stored in terms of a battery voltage-related characteristic.

Application conditions:

Initialisation: TD_S = C_TD_S_INI *at reset*

Recurrence: 10 ms

Activation: LV_S_RUN = 1

Deactivation: LV_S_RUN = 0

Formula section:

TD_S = IP_TD_S(VB,N_32) * TD_FAC + C_TD_S_AS

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TD_S	8*8	0...FFFFH	0...262,14	0,004	ms
LDPM_VB_1_IGRE	8	0…FFH	0….26	0,102	V
LDPM_N_32_1_IGRE	8	0…FFH	0….8160	32	rpm
Dwell time correction versus battery voltage and engine speed for stratified mode					
C_TD_S_INI	2	0...FFFF H	0...262.14	0.004	ms
Init. value for dwell time control (same for homogeneous and stratified mode)					
C_TD_S_AS	2	F800...7F0H	-8.192...8.128	0.004	ms
Application intervention for stratified mode (typ. Value : 0 ms)					

38.12 TD and IGA switches

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_IGC_H_RNG[NC_CYL_NR]	O/V	FA60h..5A0h	-90..+90	0.0625	°CRK
Ignition angle (high range) at cylinder x					
TD_IGC[NC_CYL_NR]	O/V	0..FFFFH	0...262,14	0,004	ms
Dwell time duration					
TD_LIM_MAX	O/V	0..FFFFH	0...262,14	0,004	ms
Maximum dwell time allowed					
TD_FAC_MAX_TMP	-	1...FFH	1/128...255/128	1/128	-
Maximal dwell time control					
TD_FAC_MIN_TMP	-	1...FFH	1/128...255/128	1/128	-
Minimal dwell time control					
TD_IGC_ACT_MAX	-	0..FFFFH	0...262,14	0,004	ms
Maximal Dwell time duration					
TD_IGC_ACT_MIN	-	0..FFFFH	0...262,14	0,004	ms
Minimal Dwell time duration					

Input data:

IGA_AV_H_RNG[NC_CYL_NR]	TD	INJ_MOD[NC_CYL_NR]	TD_FAC_MAX
INJ_MOD_SP[NC_CYL_NR]	TD_S	TD_FAC_MIN	NC_CYL_NR
LV_ES	LV_IGK	LV_IGC_EXT_ADJ[NC_CYL_NR]	IGC_EXT_ADJ[NC_CYL_NR]
LV_FIRST_VLD_TOOTH	LV_INH_IGC[NC_CYL_NR]	NC_TD_LIM_MIN	NC_TD_LIM_MAX_L
TD_AD[NC_CYL_NR]	VB	NC_INJ_MOD_MASK_1	NC_INJ_MOD_HOM
NC_TD_LIM_MAX_H	LV_ST_END	IGA_IGC_H_RNG_ACT[NC_CYL_NR]	TD_IGC_ACT[NC_CYL_NR]
LV_IGN_INJ_SYN_DEAC			

Imported actions:

ACTION_INFR_SetIgnDwell (IN <>, IN <>, IN <>, IN <>)

ACTION_INFR_SetIgnAngle (IN <>, IN <>)

ACTION_INFR_SetIgcDwellTest (IN <>, IN <>)

ACTION_INJR_IgnUpdate (IN <>, IN <>)

Exported action:

ACTION_IGRE_SendTrigForIgnUpd (IN <PRM_STATE_IGN_UPD_ENA>)

This action allows INJR to trig IGRE for updating its ignition parameter to its driver according to the designating cylinders specified by action parameter.

Description for action:

ACTION_IGRE_SendTrigForIgnUpd (IN <PRM_STATE_IGN_UPD_ENA>)

This action allows INJR to trig IGRE for updating its ignition parameter to its driver according to the designating cylinders specified by action parameter.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
PRM_STATE_IGN_UPD_ENA	IN	0...FF	0...255	1	-
Bitmap to enable/disable ignition update calculations.					

Formula section For ACTION_IGRE_SendTrigForIgnUpd:

```
TD_IGC_ACT_MAX = 0 ms
TD_IGC_ACT_MIN = 262.14 ms

For x = 0 to NC_CYL_NR - 1
    TD_IGC_ACT_MAX = max (TD_IGC_ACT[x], TD_IGC_ACT_MAX)
    TD_IGC_ACT_MIN = min (TD_IGC_ACT[x], TD_IGC_ACT_MIN)
Endfor

TD_ FAC_MAX_TMP = min (TD_LIM_MAX / TD_IGC_ACT_MAX, TD_FAC_MAX)
TD_ FAC_MIN_TMP = max (NC_TD_LIM_MIN / TD_IGC_ACT_MIN, TD_FAC_MIN)

TD_IGC[x] = TD_IGC_ACT[x]
IGA_IGC_H_RNG[x] = IGA_IGC_H_RNG_ACT[x]

For x = 0 to NC_CYL_NR – 1
    Set ignition parameters if newly calculated:
    If bit x of PRM_STATE_IGN_UPD_ENA == 1
    Then
            ACTION_INFR_SetIgnDwell (x, TD_FAC_MIN_TMP, TD_IGC[x],
                    TD_FAC_MAX_TMP)
            ACTION_INFR_SetIgnAngle (x, IGA_IGC_H_RNG[x])
    EndIf
Endfor
```

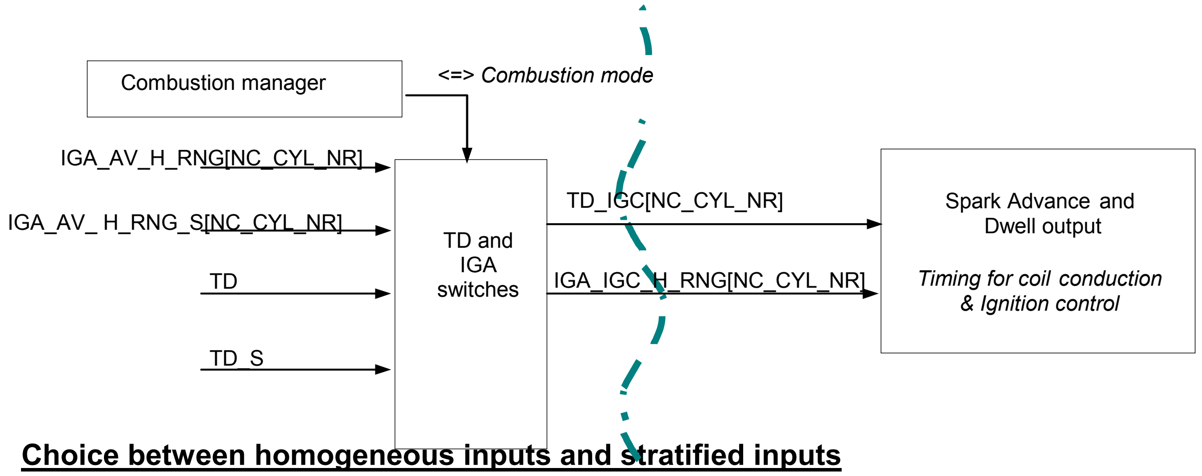
FUNCTION DESCRIPTION:

This module should cover all necessary switches between the different dwell times and ignition angles to ensure two outputs, TD_IGC[NC_CYL_NR] and IGA_IGC_H_RNG[NC_CYL_NR].

Initialisation: at reset

```
TD_IGC[NC_CYL_NR] = TD + TD_AD[NC_CYL_NR]
IGA_IGC_H_RNG[NC_CYL_NR] = IGA_AV_H_RNG[NC_CYL_NR]

For x = 0 to NC_CYL_NR - 1
    ACTION_INFR_SetIgnDwell(x,TD_FAC_MIN,TD_IGC[x],TD_FAC_MAX)
    ACTION_INFR_SetIgnAngle(x,IGA_IGC_H_RNG[x])
Endfor
```

Signal flow diagram:



Choice between homogeneous inputs and stratified inputs

The switch between homogeneous and stratified combustion mode could not be realised between TD_ON[NC_CYL_NR] and TD_OFF[NC_CYL_NR].

Switching is only possible if there's no action on the cylinder.

The switch has to be synchronised with the injection switch. The combustion manager managed by INJR does this synchronisation. The switch has to work per cylinder. Some inputs on this signal flow diagram are in fact use by the synchronization module of INJR.

38.12.1 Update of Ignition angle if homogeneous mode is selected

Application conditions:

Activation: LV_ST_END = 1: Engine synchronized

Recurrence: 10ms

Formula section:

```
For x = 0 to NC_CYL_NR - 1
    If    INJ_MOD[x] == 'SNGH' And INJ_MOD_SP[x] == 'SNGH'
          Or    [ INJ_MOD[x] == 'MPLH' And INJ_MOD_SP[x] == 'MPLH' And LV_IGN_INJ_SYN_DEAC = 1 ]
          Or    [ INJ_MOD[x] == 'MPLH+PLS3' And INJ_MOD_SP[x] == 'MPLH+PLS3' And LV_IGN_INJ_SYN_DEAC = 1 ]
    Then  IGA_IGC_H_RNG[x] = IGA_AV_H_RNG[x]
          ACTION_INJR_IgnUpdate (x, IGA_IGC_H_RNG[x])
          ACTION_INFR_SetIgnAngle (x, IGA_IGC_H_RNG_ACT[x])
    Endif
Endfor
```

38.12.2 TD_LIM_MAX calculation

Application conditions:

Activation: Engine synchronized

Recurrence: 10ms

Initialisation: at reset

Formula section:

```
If      VB < C_VB_TD_MAX_LIM_L
Then    TD_LIM_MAX = NC_TD_LIM_MAX_H
Endif

If      VB > C_VB_TD_MAX_LIM_H
Then    TD_LIM_MAX = NC_TD_LIM_MAX_L
Endif
```

In every case: TD_IGC[x] * TD_FAC_MAX <= TD_LIM_MAX

TD_IGC[x] * TD_FAC_MIN >= NC_TD_LIM_MIN

NC_TD_LIM_MIN is defined for dwell time protection in case of advance ignition changes during the dwell time application. If the dwell time is equal to this minimal limit value, the advance ignition could be not guaranteed

NC_TD_LIM_MAX is defined for the maximum dwell time in case of advance ignition changes during the dwell time application. If the dwell time is equal to this maximal limit value, the advance ignition could be not guaranteed

Note: In any case, TD_FAC_MIN has to be smaller than 1 and TD_FAC_MAX greater than one to be sure to respect the requested dwell and avoid inconsistencies
care has to be taken with the setting of NC_TD_LIM_MAX_L and NC_TD_LIM_MAX_H in order to avoid Dwell Time / coil charging recovery:
Dwell time can start at the earliest at 288°CRK before the TDC.
NC_TD_LIM_MAX_L and NC_TD_LIM_MAX_H should be tuned in consequence to respect coil charging time no recovery and coil capacity

*TD MAX = min(TD_IGC[x] * TD_FAC_MAX , TD_LIM_MAX)*

*TD MIN = max(TD_IGC[x] * TD_FAC_MIN, NC_TD_LIM_MIN)*

38.12.3 Dwell time switch Ignition Actuator Tests Diagnosis

Application conditions:

Activation: [LV_ES = 1 *// Actuator tests activation*
and LV_IGK = 1
and LV_FIRST_VLD_TOOTH = 0
and LV_INH_IGC[x] = 0
and at least one LV_IGC _EXT_ADJ[x] = 1]

Deactivation: [LV_ES = 0 *// Actuator tests deactivation*
or LV_IGK = 0
or LV_FIRST_VLD_TOOTH = 1
or LV_INH_IGC[x] = 1
or at least one LV_IGC_EXT_ADJ[x] = 0]

Recurrence: On transition of IGC_EXT_ADJ[x] from 0 to 1

System description:

For ignition coil actuator tests purpose, the Dwell Time has to be applied and transmitted to BSW via action. A specific service is used for those tests purposes.

Formula section:

```
If      (INJ_MOD[x] & NC_INJ_MOD_MASK_1) != NC_INJ_MOD_HOM // Stratified mode for cyl. _x

Then          TD_IGC[x]     = TD_S
Else          TD_IGC[x]     = TD
Endif

For           x = 0 to NC_CYL_NR – 1
              If        LV_IGC _EXT_ADJ[x] = 1
              Then      On transition of IGC _EXT_ADJ[x] from 0 to 1
                                  And LV_FIRST_VLD_TOOTH = 0
                        ACTION_INFR_SetIgcDwellTest(x,TD_IGC[x])
              ElseIf    LV_FIRST_VLD_TOOTH = 1
                        ACTION_INFR_SetIgcDwellTest(x,0)        // immediate cut of tests on engine running
              EndIf
EndFor
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VB_TD_MAX_LIM_H	1	0...FFH	0...26	0.102	V
High battery voltage threshold above which we apply NC_TD_LIM_MAX_H for maximum dwell time					
C_VB_TD_MAX_LIM_L	1	0...FFH	0...26	0.102	V
Low battery voltage threshold below which we apply NC_TD_LIM_MAX_L for maximum dwell time					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_TD_LIM_MIN	1	0...FFFFH	0...262,14	0.004	ms
NC_TD_LIM_MIN (Typical value 1 ms)					
NC_TD_LIM_MAX_L	1	0...FFFFH	0...262,14	0.004	ms
NC_TD_LIM_MAX_L - Typical value 10 ms					
NC_TD_LIM_MAX_H	1	0...FFFFH	0...262,14	0.004	ms
NC_TD_LIM_MAX_H - Typical value 20 ms					

38.13 Ignition with multiple spark

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TD_MPL_1	V/O	0...FFFFH	0...262,14	0,004	ms
Dwell time for 1st multiple spark (e.g. 1st spark after main spark)					
TD_MPL	V/O	0...FFFFH	0...262,14	0,004	ms
Dwell time for multiple spark (for 2nd and others MPL sparks)					
TD_MPL_1_DLY	V/O	0...FFFFH	0...262,14	0,004	ms
Dwell time interruption between main spark and 1st MPL spark					
TD_MPL_DLY	V/O	0...FFFFH	0...262,14	0,004	ms
Dwell time interruption between MPL sparks (except between main and 1st MPL spark)					
IGN_MPL_NR[NC_CYL_NR]	V/O	0...FFH	0…255	1	-
Number of ignition sparks					
T_CTR_TD_MPL	V	0...FFFFH	0...6553,5	0,1	s
Maintenance Time of Multiple Spark activation before Out Case					

Input data:

VB	TCO	LV_PUC	N_32
LV_IS	LV_ST	LV_ES	LV_FL
LDPM_VB_1_IGRE	LDPM_N_32_1_IGRE	LV_PL	LV_PU
LV_TD_MPL_SPC_IGN_ACT	#IF(NC_INJ_CONF=HPDI	NC_INJ_MOD_MASK_1	NC_INJ_MOD_HOM
LV_INH_TD_MPL	INJ_MOD[NC_CYL_NR]	NC_CYL_NR	NR_MPL_CYL_IGN [NC_CYL_NR]
INJ_MOD_MPL	#ENDIF	TEMP_MDL_IGC	

Import actions:

ACTION_ INFR_SetIgnMpl (IN <> , IN <> , IN <>, IN <>, IN <>, IN <>)

38.13.1 Calculation of parameters for multiple spark

FUNCTION DESCRIPTION:

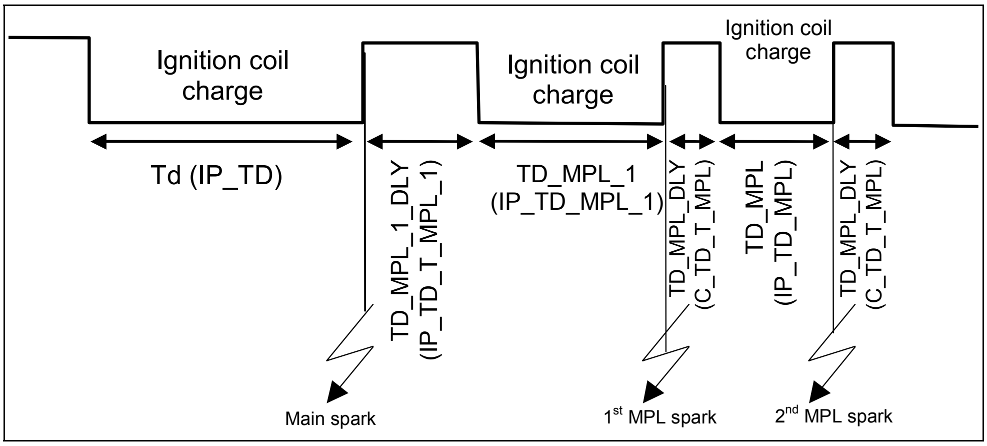
With this function several subsequent sparks could be created. The delay time to create a spark and the time to reload the coil are defined in tables depending on battery voltage and on engine speed.

At start and for the following application conditions, the ignition is controlled by means of multiple sparks.

After an adjustable combustion period, the ignition coil is loaded with a specific dwell time to create successive sparks. The number of successive sparks is defined as IGN_MPL_NR.

The multiple sparks ignition can also be activated by the input flag LV_TD_MPL_SPC_IGN_ACT. In this case, the 1st MPL spark (e.g. the first spark after the main spark) is controlled by specific calibration data IP_TD_MPL_1 and IP_TD_T_MPL_1 (see the picture below for more explanation).

If LV_TD_MPL_SPC_IGN_ACT is inactive, the 1st MPL spark takes the same parameters than the other MPL sparks.



The MPL Sparks are maintained during a time (C_T_CTR_TD_MPL_AST or C_T_CTR_TD_MPL) after the end of the activation condition, before exiting this MPL procedure (*Application Multiple Spark Maintenance for a time before Exit Case*)

We use an estimated coil temperature to limit the number of multiple sparks and then protect ignition coil.

We also increase the number of multiple sparks in case of misfiring.

Priority is given to the protection of ignition coil, according to the estimated temperature coil compared to a maximum value: C_TEMP_MDL_IGC_MAX.

Application conditions:

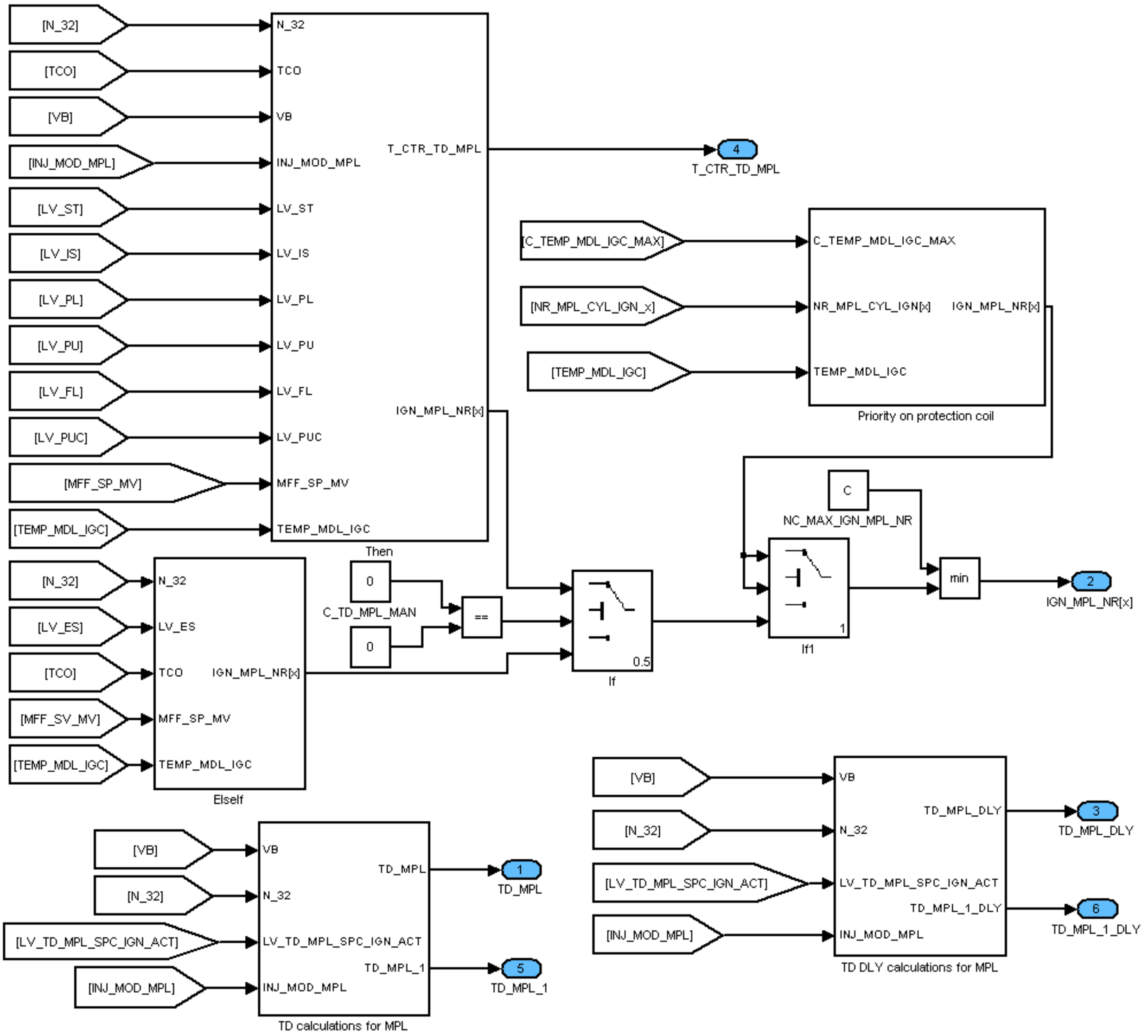
Initialisation: IGN_MPL_NR = 0
TD_MPL = 0
TD_MPL_1 = 0
TD_MPL_DLY = 0
TD_MPL_1_DLY = 0
T_CTR_TD_MPL = 0 *At reset and in deactivation condition*

Recurrence: 10 ms

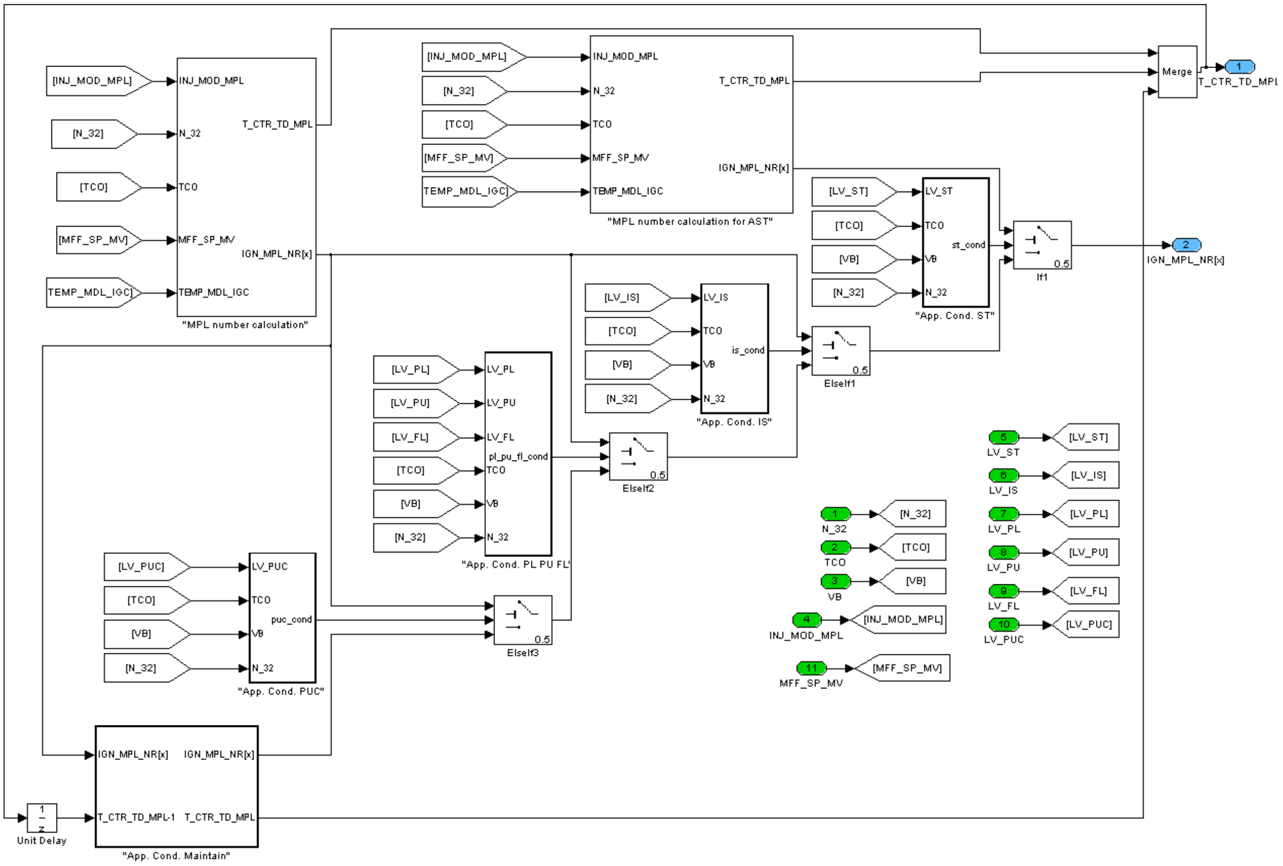
Activation: LV_INH_TD_MPL = 0

Deactivation: LV_INH_TD_MPL = 1

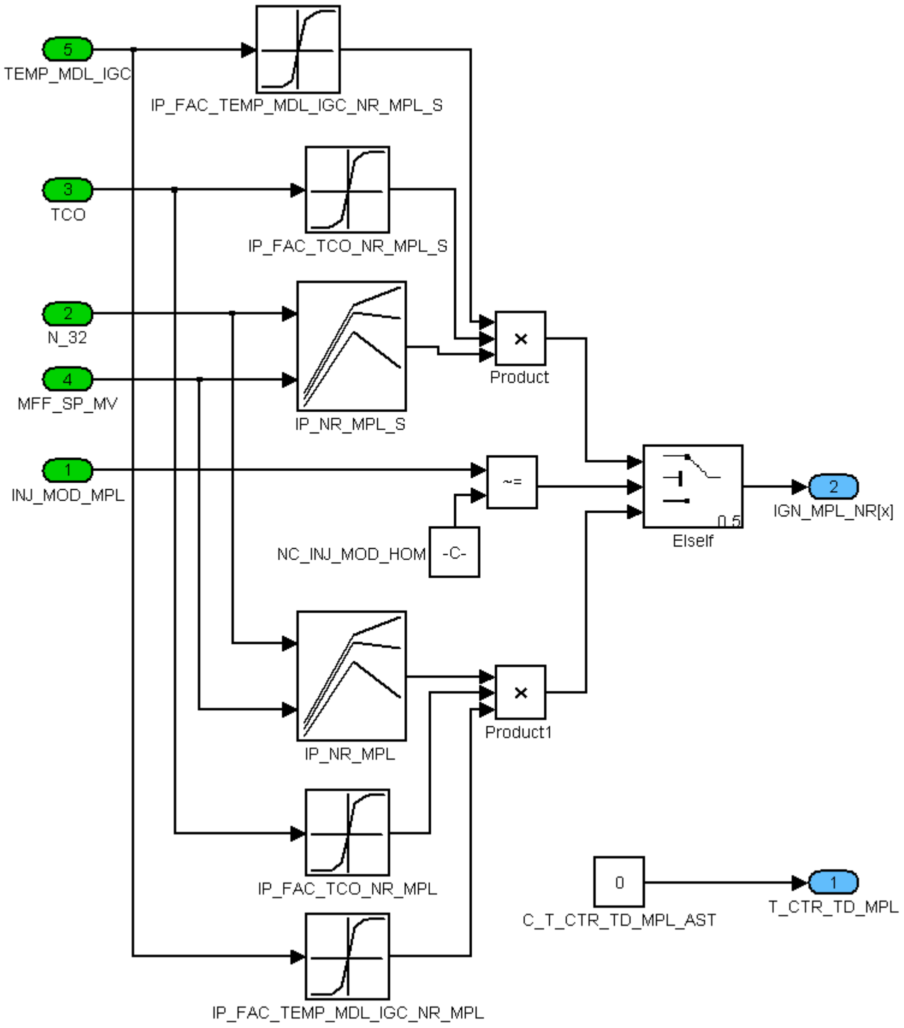
Formula section:



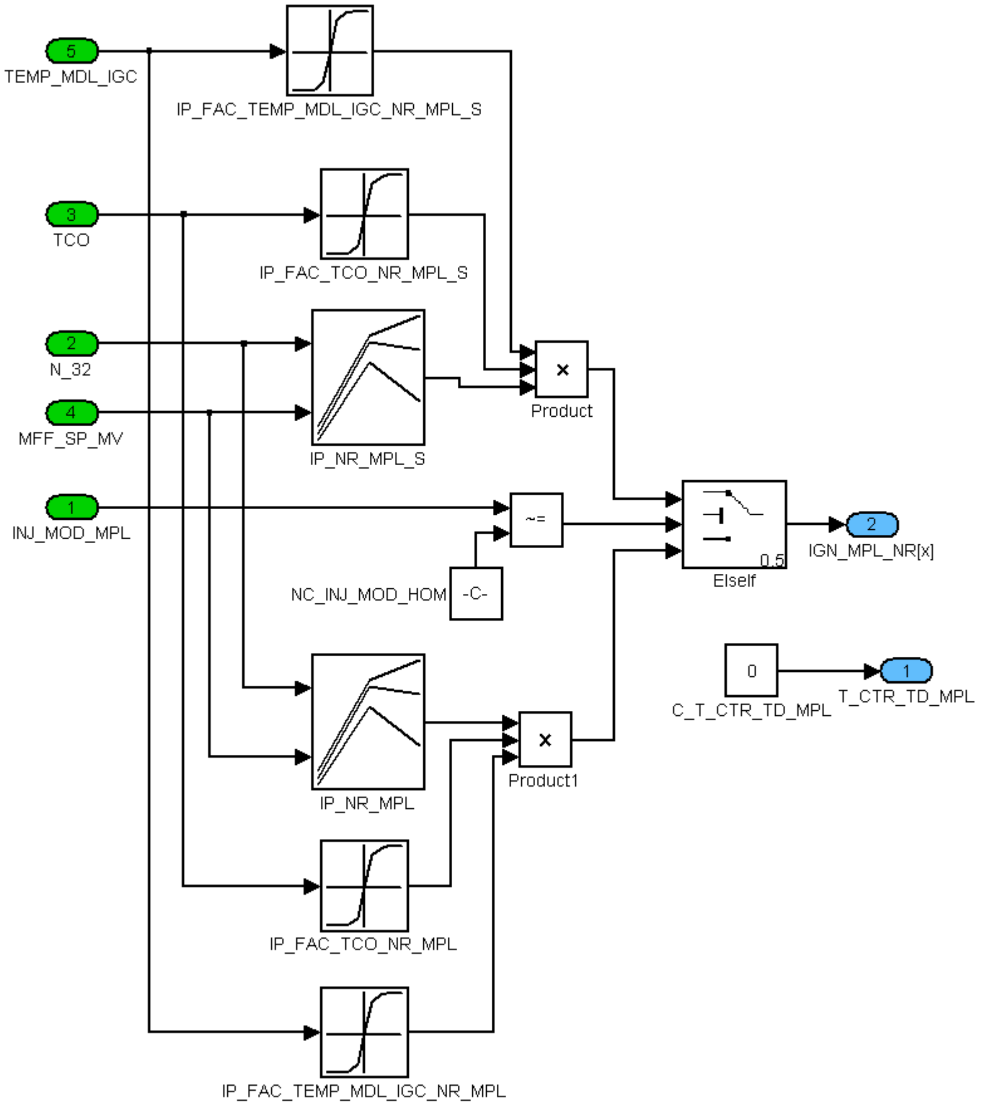
IGRE_TD_MPL



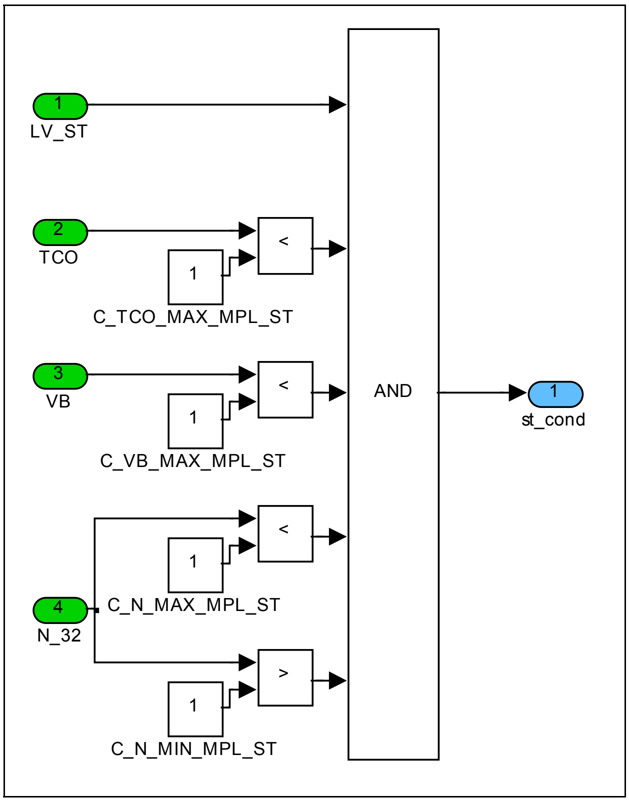
iGRE_TD_MPL / Then



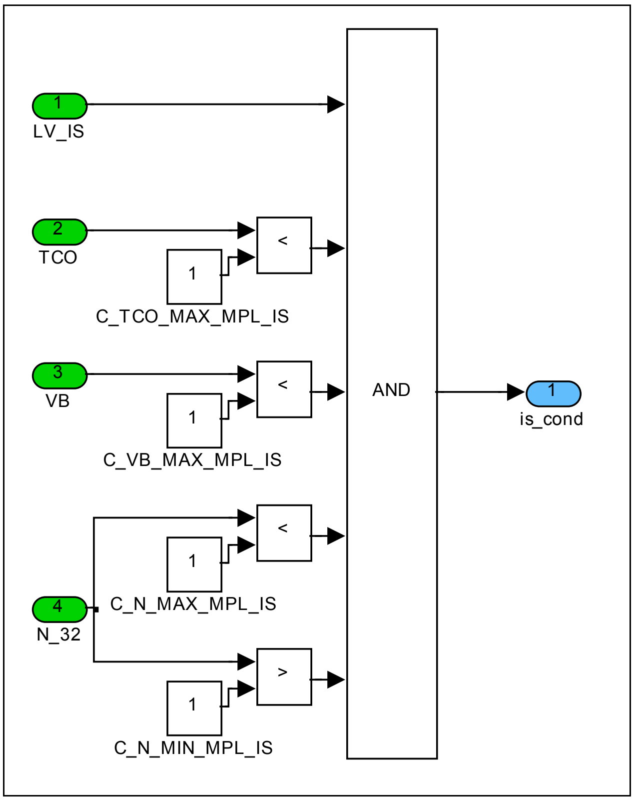
IGRE_TD_MPL / Then / MPL number calculation for AST



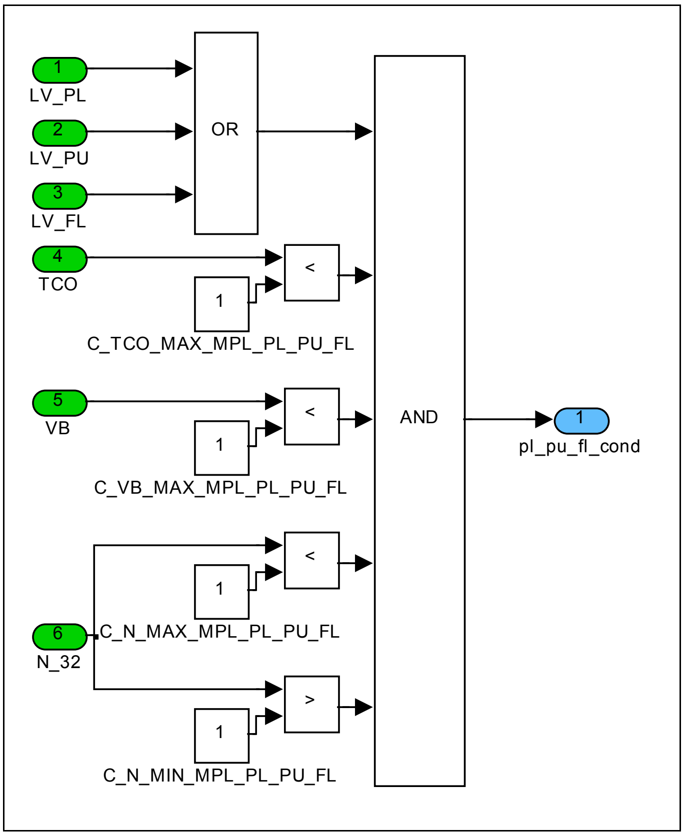
IGRE_TD_MPL / Then / MPL number calculation



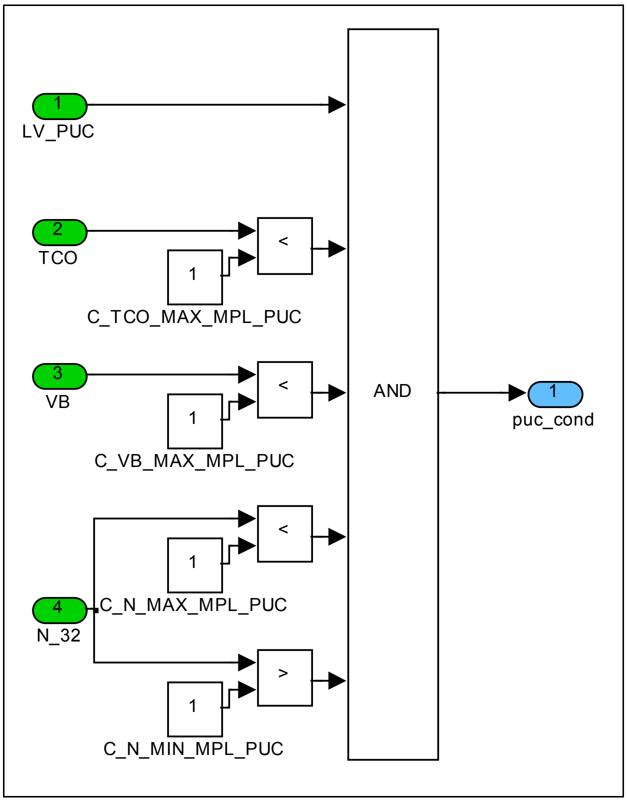
IGRE_TD_MPL / Then / App. Cond. ST



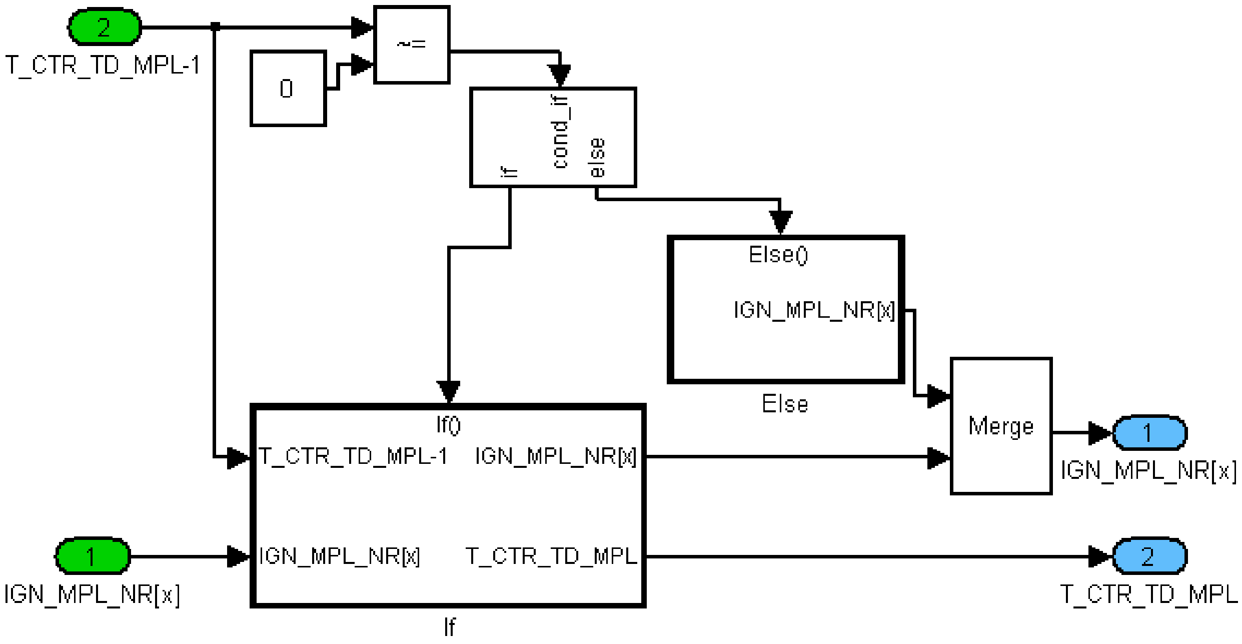
IGRE_TD_MPL / Then / App. Cond. IS



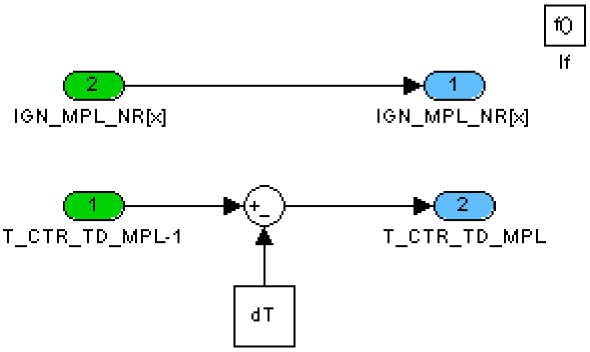
IGRE_TD_MPL / Then / App. Cond. PL PU FL



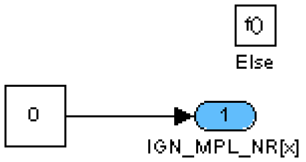
IGRE_TD_MPL / Then / App. Cond. PUC



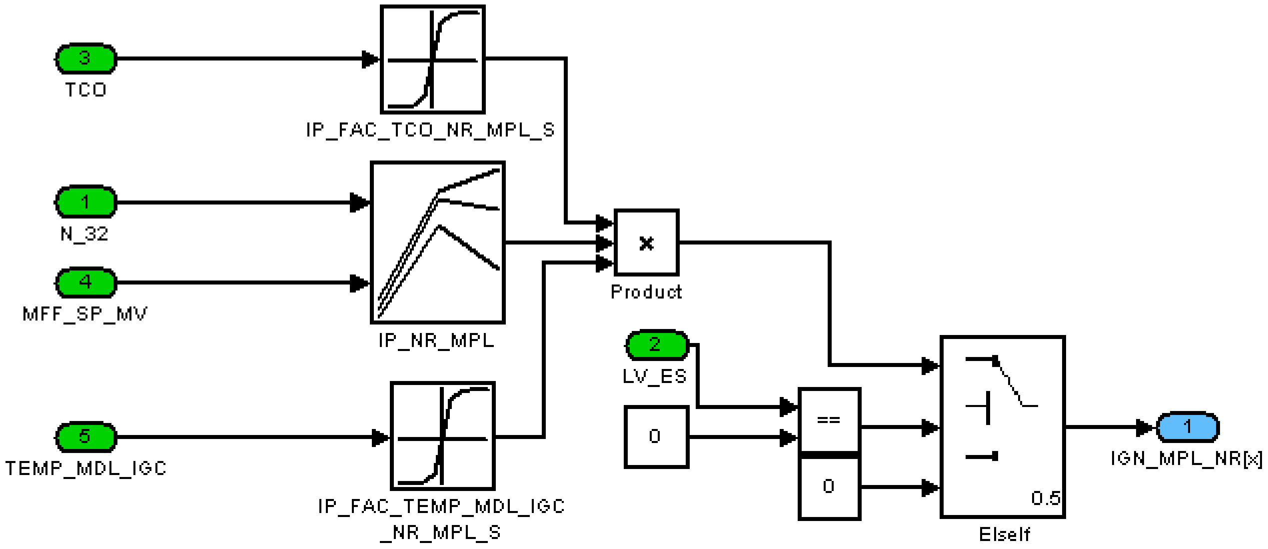
IGRE_TD_MPL / Then /App. Cond. Maintain



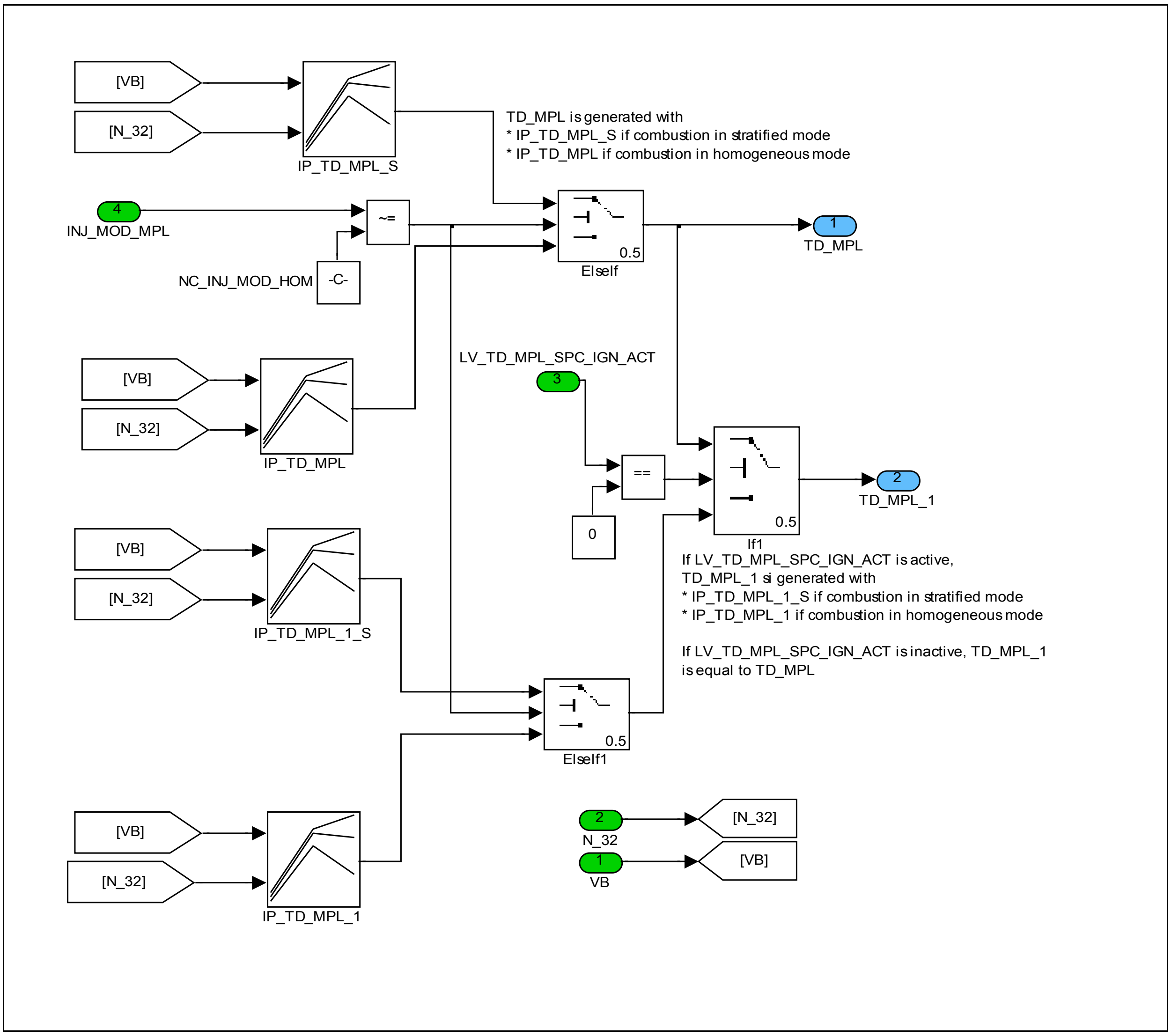
IGRE_TD_MPL / Then /App. Cond. Maintain / If



IGRE_TD_MPL / Then /App. Cond. Maintain / Else



IGRE_TD_MPL / ElseIf

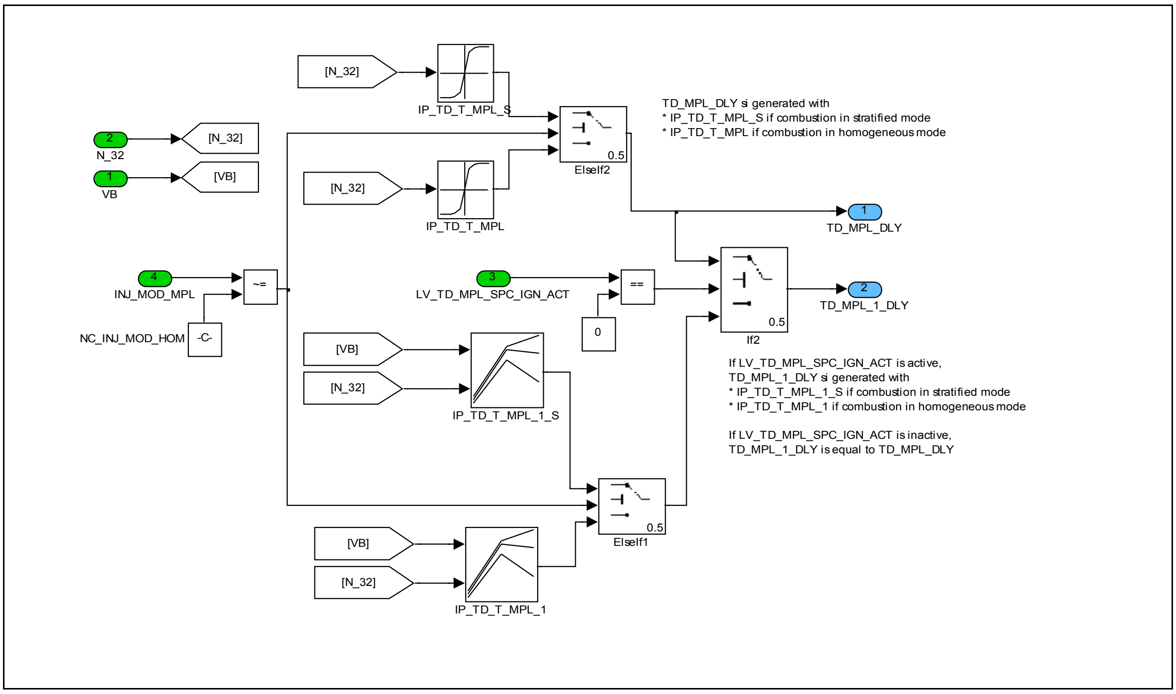


TD_MPL is generated with
* IP_TD_MPL_S if combustion in stratified mode
* IP_TD_MPL if combustion in homogeneous mode

If LV_TD_MPL_SPC_IGN_ACT is active,
TD_MPL_1 si generated with
* IP_TD_MPL_1_S if combustion in stratified mode
* IP_TD_MPL_1 if combustion in homogeneous mode

If LV_TD_MPL_SPC_IGN_ACT is inactive, TD_MPL_1
is equal to TD_MPL

IGRE_TD_MPL / TD calculations for MPL

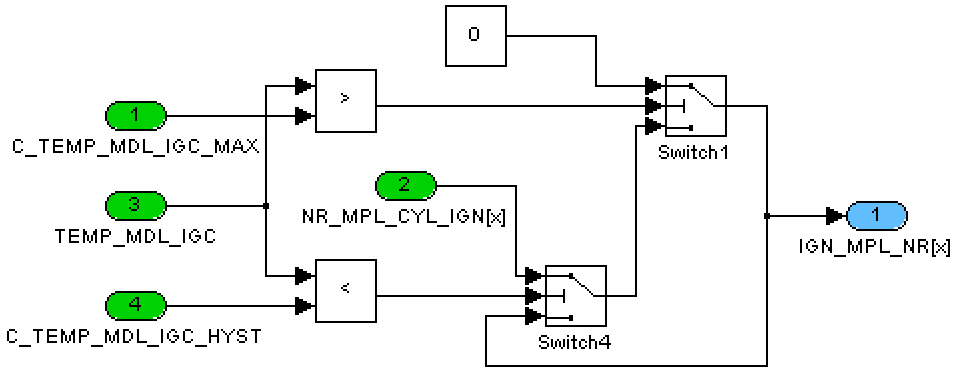


TD_MPL_DLY si generated with
* IP_TD_T_MPL_S if combustion in stratified mode
* IP_TD_T_MPL if combustion in homogeneous mode

If LV_TD_MPL_SPC_IGN_ACT is active,
TD_MPL_1_DLY si generated with
* IP_TD_T_MPL_1_S if combustion in stratified mode
* IP_TD_T_MPL_1 if combustion in homogeneous mode

If LV_TD_MPL_SPC_IGN_ACT is inactive,
TD_MPL_1_DLY is equal to TD_MPL_DLY

IGRE_TD_MPL / TD DLY calculations for MPL



Priority on protection coil

38.13.2 Settings of multiple spark parameters

Application conditions:

Initialisation: *At reset*

ACTION_INFR_SetIgnMpl (x, TD_MPL_1, TD_MPL, TD_MPL_1_DLY, TD_MPL_DLY, IGN_MPL_NR[x])

Recurrence: 10 ms

Formula section:

For x = 0 **to** NC_CYL_NR – 1

ACTION_INFR_SetIgnMpl (x, TD_MPL_1, TD_MPL, TD_MPL_1_DLY, TD_MPL_DLY, IGN_MPL_NR[x])

EndFor

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_TCO_MAX_MPL_ST	1	0...FEH	-48...142.5	0.75	°C
Max. Coolant Temp. to have multiple spark at start					
C_TCO_MAX_MPL_IS	1	0...FEH	-48...142.5	0.75	°C
Max. Coolant Temp. to have multiple spark at Idle speed					
C_TCO_MAX_MPL_PL_PU_FL	1	0...FEH	-48...142.5	0.75	°C
Max. Coolant Temp. to have multiple spark at Run					
C_TCO_MAX_MPL_PUC	1	0...FEH	-48...142.5	0.75	°C
Max. Coolant Temp. To have multiple spark at fuel cut off					
C_VB_MAX_MPL_ST	1	0...FFH	0...26	0.102	V
Max. Battery voltage to have multiple spark (requested value 16V) at start					
C_VB_MAX_MPL_IS	1	0...FFH	0...26	0.102	V
Max. Battery voltage to have multiple spark (requested value 16V) at idle speed					
C_VB_MAX_MPL_PL_PU_FL	1	0...FFH	0...26	0.102	V
Max. Battery voltage to have multiple spark (requested value 16V) at Run					
C_VB_MAX_MPL_PUC	1	0...FFH	0...26	0.102	V
Max. Battery voltage to have multiple spark (requested value 16V) at fuel cut off					
C_N_MIN_MPL_ST	1	0...FFH	0...8160	32	rpm
Engine speed min for ignition with multiple spark at Start					
C_N_MAX_MPL_ST	1	0...FFH	0...8160	32	rpm
Engine speed max for ignition with multiple spark at Start					
C_N_MIN_MPL_IS	1	0...FFH	0...8160	32	rpm
Engine speed min for ignition with multiple spark at Idle					
C_N_MAX_MPL_IS	1	0...FFH	0...8160	32	rpm
Engine speed max for ignition with multiple spark at idle					
C_N_MIN_MPL_PL_PU_FL	1	0...FFH	0...8160	32	rpm
Engine speed min for ignition with multiple spark at Run					
C_N_MAX_MPL_PL_PU_FL	1	0...FFH	0...8160	32	rpm
Engine speed max for ignition with multiple spark at Run					
C_N_MIN_MPL_PUC	1	0...FFH	0...8160	32	rpm
Engine speed min for ignition with multiple spark at fuel cut off					
C_N_MAX_MPL_PUC	1	0...FFH	0...8160	32	rpm
Engine speed max for ignition with multiple spark at Fuel cut off					
IP_TD_T_MPL	8	0...FFFFH	0...262,14	0,004	ms
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
Minimum time between two ignition (normal value 150µs) in homogeneous mode					
IP_TD_T_MPL_S	8	0...FFFFH	0...262,14	0,004	ms
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
Minimum time between two ignition (normal value 150µs) in stratified mode					
IP_TD_MPL_1	8*8	0...FFFFH	0...262,14	0,004	ms
LDPM_VB_1_IGRE	8	0...FFH	0...26	0.102	V
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
Dwell time for 1st multiple spark in homogeneous mode confirmed					
IP_TD_MPL	8	0...FFFFH	0...262,14	0,004	ms
LDPM_VB_1_IGRE	8	0...FFH	0...26	0.102	V
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
Dwell time for multiple spark in homogeneous mode confirmed					
IP_TD_T_MPL_1	8*8	0...FFFFH	0...262,14	0,004	ms
LDPM_VB_1_IGRE	8	0...FFH	0...26	0.102	V
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
Time delay between mains spark and start of ign coil drive for 1st MPL spark (homogeneous mode confirmed)					
IP_TD_MPL_S	8	0...FFFFH	0...262,14	0,004	ms
LDPM_VB_1_IGRE	8	0...FFH	0...26	0.102	V
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
Dwell time for multiple spark in stratified mode confirmed					
IP_TD_MPL_1_S	8*8	0...FFFFH	0...262,14	0,004	ms
LDPM_VB_1_IGRE	8	0...FFH	0...26	0.102	V
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
Dwell time for 1st multiple spark in stratified mode confirmed					
IP_TD_T_MPL_1_S	8*8	0...FFFFH	0...262,14	0,004	ms
LDPM_VB_1_IGRE	8	0...FFH	0...26	0.102	V
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
Time delay between main spark and start of ign coil drive for 1st MPL spark (stratified mode confirmed)					

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
IP_NR_MPL	8*6	0…FFH	0…255	1	-
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	rpm
LDPM_MFF_SP_MV_1_IGRE	6	0...FFFFH	0...1389	0.0211948	mg/stk
Number of successive ignition sparks depending on engine speed and mass fuel in homogeneous mode					
IP_NR_MPL_S	8*6	0…FFH	0…255	1	-
LDPM_N_32_1_IGRE	8	0...FFH	0...8160	32	Rpm
LDPM_MFF_SP_MV_1_IGRE	6	0...FFFFH	0...1389	0.0211948	mg/stk
Number of successive ignition sparks depending on engine speed and mass fuel in stratified mode					
C_TD_MPL_MAN	1	0...1H	0....1	1	-
Manual activation of the MPL					
C_T_CTR_TD_MPL_AST	1	0...FFFFH	0...655,35	0,01	s
Maintenance Time of Multiple Spark activation before Out Case and for the After Start phase					
C_T_CTR_TD_MPL	1	0...FFFFH	0...655,35	0,01	s
Maintenance Time of Multiple Spark activation before Out Case					
IP_FAC_TCO_NR_MPL	8	0...50H	0...8	0.1	-
LDPM_TCO_1_IGRE	8	0...FEH	-48...142.5	0.75	°C
TCO factor used to calculate multiple spark number in homogeneous mode					
IP_FAC_TCO_NR_MPL_S	8	0...50H	0...8	0.1	-
LDPM_TCO_1_IGRE	8	0...FEH	-48...142.5	0.75	°C
TCO factor used to calculate multiple spark number in stratified mode					
IP_FAC_TEMP_MDL_IGC_NR_MPL	8	0...50H	0...8	0.1	-
LDPM_TEMP_MDL_IGC_1_IGRE	8	0...FFH	-40...215	1	°C
estimated coil temperature factor used to calculate multiple spark number in homogeneous mode					
IP_FAC_TEMP_MDL_IGC_NR_MPL_S	8	0...50H	0...8	0.1	-
LDPM_TEMP_MDL_IGC_1_IGRE	8	0...FFH	-40...215	1	°C
estimated coil temperature factor used to calculate multiple spark number in stratified mode					
C_TEMP_MDL_IGC_MAX	8	0...FFH	-40...215	1	°C
estimated coil temperature maximum threshold					
C_TEMP_MDL_IGC_HYS	1	0...FFH	-40...215	1	°C
Estimated coil temperature minimum threshold					

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_MAX_IGN_MPL_NR	1	0...FFH	0...255	1	-
Maximum multiple spark - Typical value for debug:7					
NC_MPL_T_MAX	1	0...FFFFH	0...262,14	0.004	ms
Maximum time duration of Multiple Spark (Normal value 60 ms)					
NC_MPL_IGN_H_RNG_CRK_MAX	1	0...12C0H	0…300	0.0625	°CRK
Maximum ignition angle after TDC to start TD_MPL (normal value 12 to 18 °CRK)					

38.14 Ignition with multiple spark (Appl. Inc)

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
EGY_COLD_IGC	O/V	0... FFFFH	0... 6553.5	0.1	[W]
Lost energy in the ignition coils because of cooling					
EGY_HEAT_IGC	O/V	0... FFFFH	0... 6553.5	0.1	[W]
Ignition energy of the ignition coils					
EGY_HEAT_IGC_HOM	O/V	0... FFFFH	0... 6553.5	0.1	[W]
Ignition energy of the ignition coils in homogeneous mode					
EGY_HEAT_IGC_S	O/V	0... FFFFH	0... 6553.5	0.1	[W]
Ignition energy of the ignition coils in stratified mode					
INJ_MOD_MPL	O/V	0... FFH	0... 255	1	[-]
Variable indicating if combustion mode is considered as homogeneous or stratified					
LV_INH_TD_MPL	O/V	0... 1H	0... 1	1	[-]
Inhibition for multiple spark					
LV_TD_MPL_SPC_IGN_ACT	O/V	0... 1H	0... 1	1	[-]
Activation of the 2° ignition event using modified extended MPL					
NR_MPL_CYL_IGN [NC_CYL_NR]	O/V	0... FFH	0... 255	1	[-]
Number of MPL for selected cylinder by BMW layer word label in homogeneous mode					
TEMP_COOL	V	0... FFH	-40... 215	1	[°C]
Cooling temperature of ignition coils					
TEMP_MDL_IGC	O/V/S	0... FFH	-40... 215	1	[°C]
Estimated coil temperature					

Input Data:

IGN_MPL_NR [NC_CYL_NR]	N_32	LV_TD_AD_H	TOIL
TCO	TAM	TECU	STATE_MPL_CYL_IGN
INJ_MOD [NC_CYL_NR]	NC_CYL_NR		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ALFA_A_IGC	1	0... 3E8H	0... 1	1e-3	[W/K]
Product of heat-transfer-coefficient and area of the ignition coils					
C_CP_M_IGC	1	0... 2710H	0... 1000	0.1	[J/K]
Product of heat-capacity and mass of ignition coils					
C_N_MAX_MPL_SPC_IGN_ACT	1	0... FFH	0... 8160	32	[rpm]
Maximum engine speed for 2° ignition event activation					
C_TCO_STE	1	0... FEH	-48... 142.5	0.75	[°C]
Minimum threshold for calculation of temperature model					
C_TECU_MAX_IGN_MPL	1	0... FEH	-48... 142.5	0.75	[°C]
TECU threshold to disable multiple spark activation					
C_TEMP_COOL_INI	1	0... FFH	-40... 215	1	[°C]
Initialisation of ignition coils cooling temperature					
C_TEMP_GRD	1	0... 320H	0... 8	0.01	[-]
Temperature gradient in integrator-block					
C_TEMP_MDL_IGC_HYS_MAX	1	0... FFH	-40... 215	1	[°C]
Estimated coil temperature maximum threshold					
C_TEMP_MDL_IGC_HYS_MIN	1	0... FFH	-40... 215	1	[°C]
Estimated coil temperature minimum threshold					
C_TEMP_MDL_IGC_INI	1	0... FFH	-40... 215	1	[°C]

initialization-value for temperature model					
C_TEMP_MDL_IGC_INT_MAX	1	0... FFH	-40... 215	1	[°C]
Integrator maximum limit of coil model-temperature					
C_TEMP_MDL_IGC_INT_MIN	1	0... FFH	-40... 215	1	[°C]
Integrator minimum limit of coil model-temperature					
C_TOIL_IGC_MAX	1	0... C8H	-40... 160	1	[°C]
Maximum threshold for oil temperature					
C_TOIL_IGC_MIN	1	0... C8H	-40... 160	1	[°C]
Threshold for TOIL					
ID_NR_IGN_MPL	8	0... FFH	0... 255	1	[-]
LDPM_N_32_1_IGRE	8	0... FFH	0... 8160	32	[rpm]
Number of successive ignition sparks in homogeneous mode if misfiring is detected					
ID_NR_IGN_MPL_S	8	0... FFH	0... 255	1	[-]
LDPM_N_32_1_IGRE	8	0... FFH	0... 8160	32	[rpm]
Number of successive ignition sparks in stratified mode if misfiring is detected					
IP_EFF_IGC	6	0... 7FH	0... 1.984375	0.015625	[-]
LDP_TEMP_MDL_IGC_IP_EFF_IGC	6	0... FFH	-40... 215	1	[°C]
Efficiency of ignition coils depending on model temperature					
IP_EGY_IGC_HOM	8*8	0... FFFFH	0... 6553.5	0.1	[W]
LDPM_IGN_MPL_NR_IP_EGY_IGC	8	0... FFH	0... 255	1	[-]
LDPM_N_32	8	0... FFH	0... 8160	32	[rpm]
ignition energy in homogeneous mode					
IP_EGY_IGC_S	8*8	0... FFFFH	0... 6553.5	0.1	[W]
LDPM_IGN_MPL_NR_IP_EGY_IGC	8	0... FFH	0... 255	1	[-]
LDPM_N_32	8	0... FFH	0... 8160	32	[rpm]
ignition energy in stratified mode					
IP_TAM_COR	6	0... FE00H	-48... 142.5	2.92969e-3	[°C]
LDP_TAM_IP_TAM_COR	6	0... FEH	-48... 142.5	0.75	[°C]
Temperature correction due to ambient temperature influence					
LC_TD_MPL_SPC_IGN_ACT	1	0... 1H	0... 1	1	[-]
Calibration for activation control of 2° ignition event using modified extended MPL to switch off					
LC_TEMP_IGC_VAR	1	0... 1H	0... 1	1	[-]
Switch between TOIL and TCO for TEMP_COOL					

Configuration Data:

Name	Mode	Hex.Limits	Phys.Limits	Resol.	Unit
NC_INJ_MOD_HOM	1	0... FFH	0... 255	1	[-]
NC_INJ_MOD_HOM					
NC_INJ_MOD_MASK_1	1	0... FFH	0... 255	1	[-]
Mask					
NC_INJ_MOD_S	1	0... FFH	0... 255	1	[-]
Constant defined to indicate stratified mode					

General Information

Application Conditions

Initialization: RST, NVMINI, NVMSTO, NVMRES
Recurrence: 100MS if alwayselse 10MS
Activation: 100MS: always
10MS: always
Deactivation: never
Additionally Activation of one event deactivates others

Function description

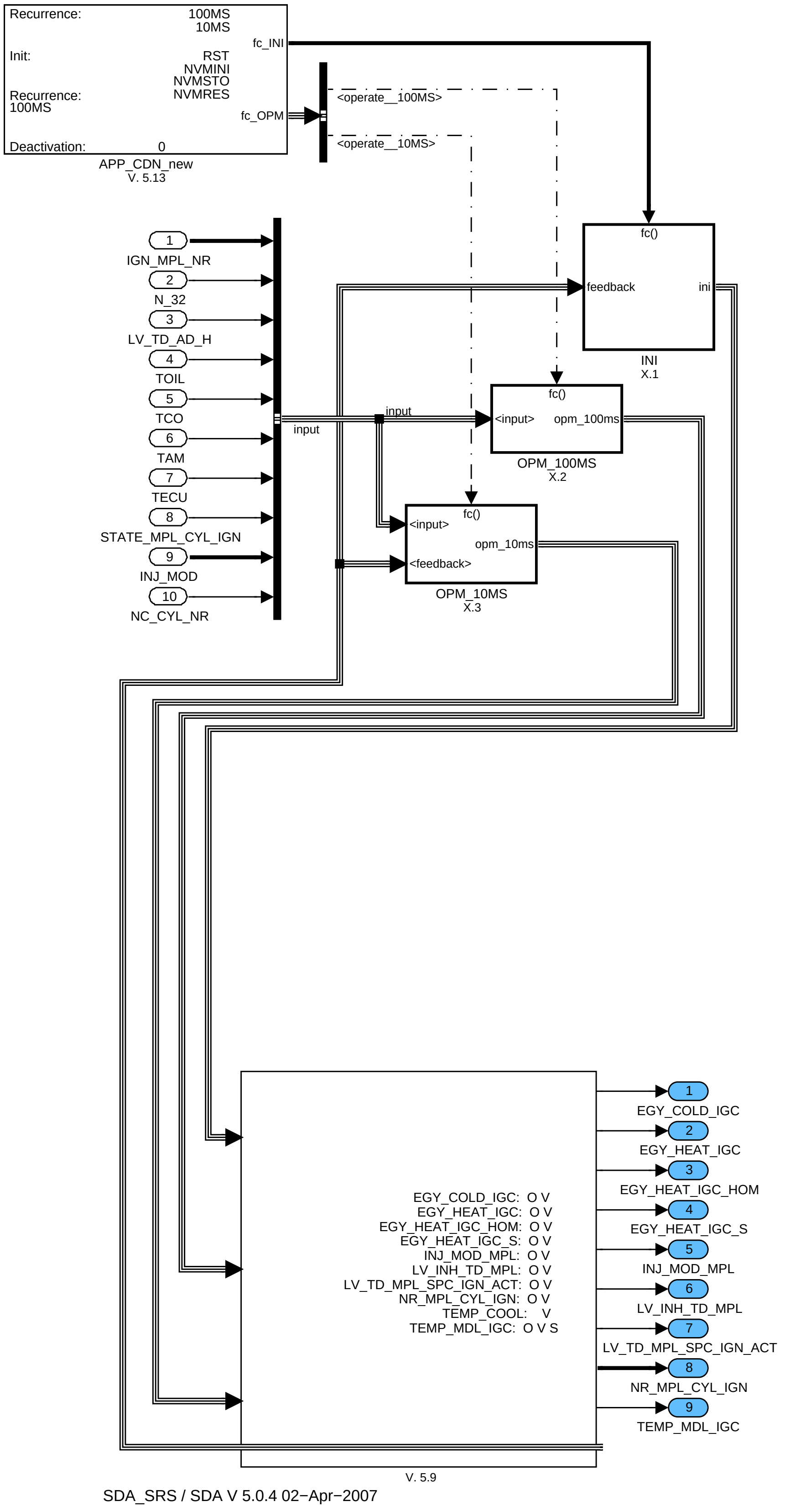


Figure 1:
Path: IGRE_M602J

38.14.1 Initialization

38.14.1.1 Initialization of TEMP_MDL_IGC in non-volatile-memory

The last calculated value of TEMP_MDL_IGC shall be stored.

38.14.1.1.1 Initialization of TEMP_MDL_IGC

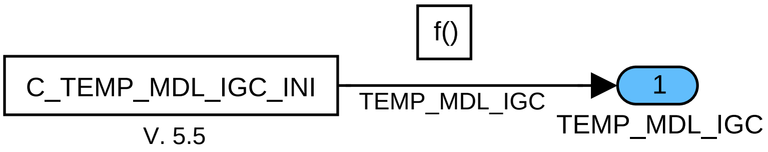


Figure 2:
Path: IGRE_M602J/INI/NVMY/INI_NVMY

38.14.1.1.2 Initialization of TEMP_MDL_IGC out of NVMY

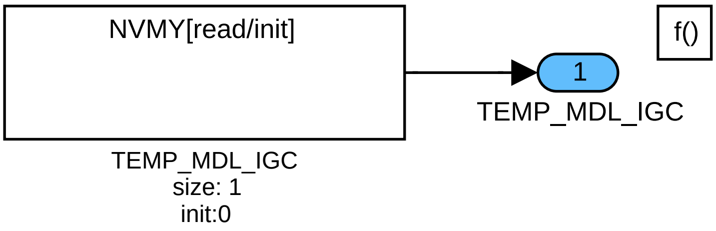


Figure 3:
Path: IGRE_M602J/INI/NVMY/RD_NVMY

38.14.1.1.3 Storing of TEMP_MDL_IGC in NVMY

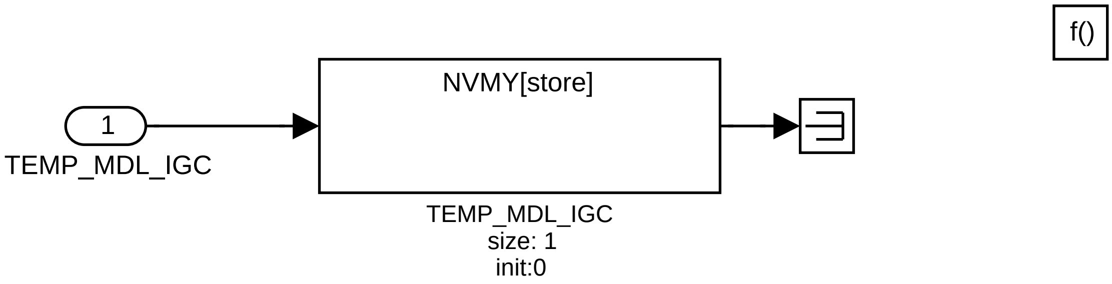


Figure 4:
Path: IGRE_M602J/INI/NVMY/SAVE_NVMY

38.14.1.2 Initialization values

Initialization

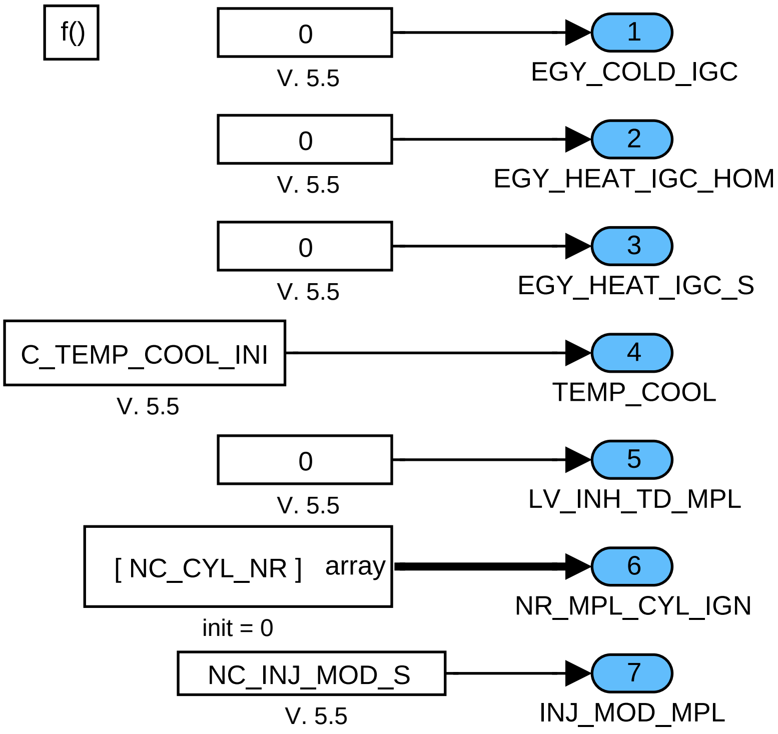


Figure 5:
Path: IGRE_M602J/INI/RST

38.14.2 Recurrence 100ms

38.14.2.1 Calculation of cooling temperature for temperature model (recurrence 100ms)

The cooling temperature is calculated as the sum of the influence of the ambient temperature and TOIL or TCO. The customer wants the LC to choose between TOIL and TCO because he is not sure which one is correct. In a future version the selection will be removed.

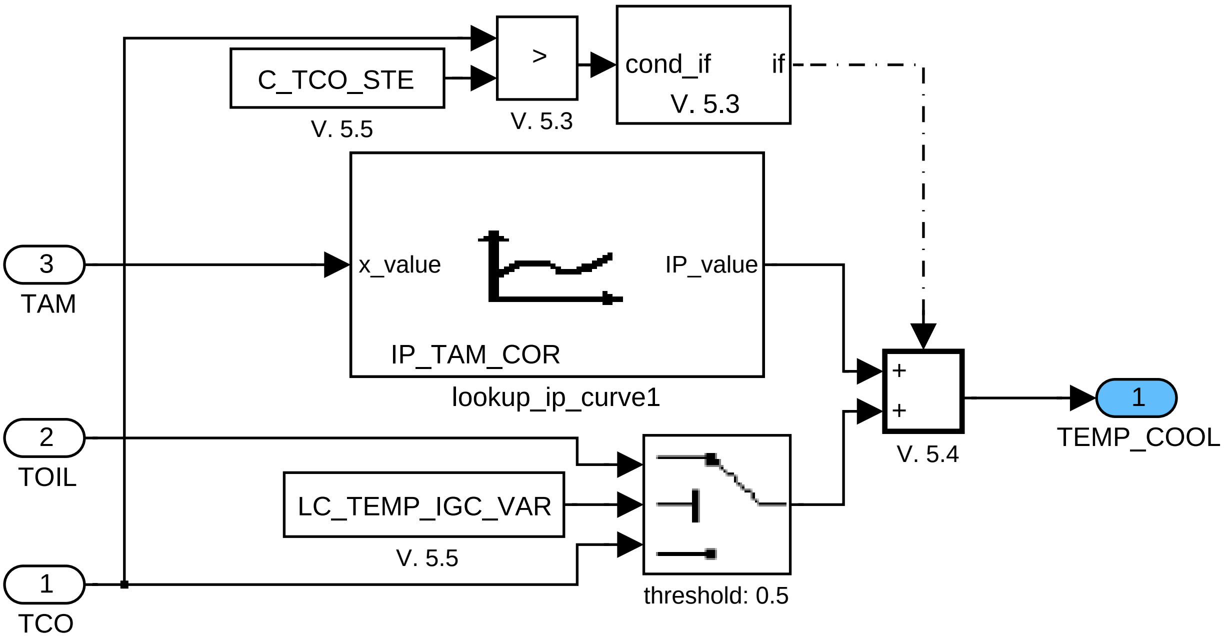


Figure 6:
Path: IGRE_M602J/OPM_100MS/TEMP

38.14.3 Recurrence 10ms

38.14.3.1 Temperature Model for Ignition Coils (recurrence 10ms)

The function shall only be calculated when the activation-condition TCO>C_TCO_STE is fulfilled.

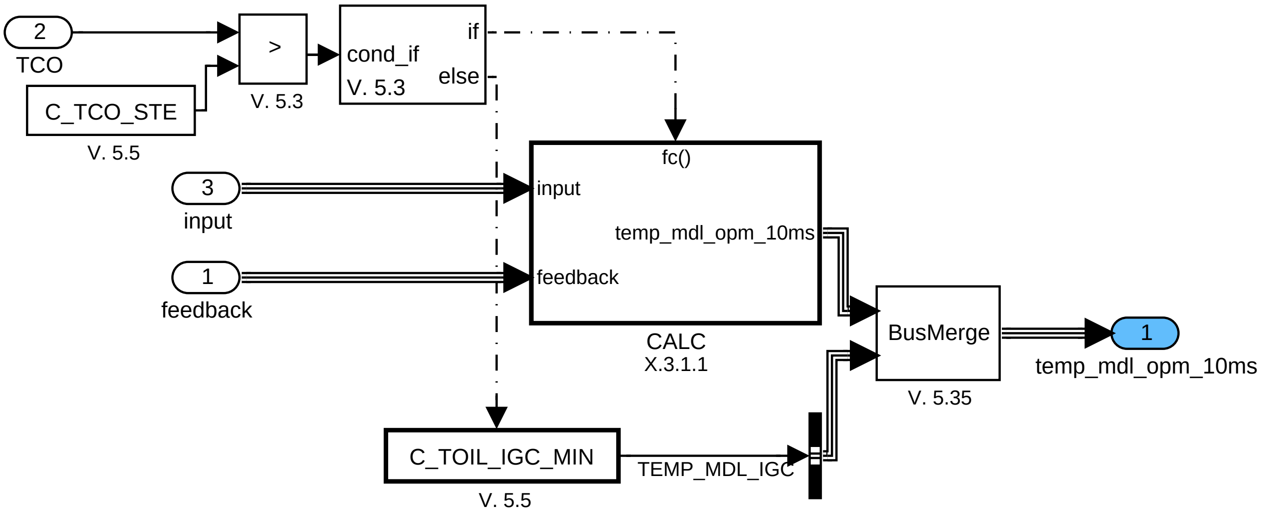


Figure 7:
Path: IGRE_M602J/OPM_10MS/CALC_TEMP_MDL

38.14.3.1.1 Calculation overview for Temperature-Model of Ignition-Coils

No Calculation

38.14.3.1.1.1 Energy of ignition-coils in homogeneous and stratified mode

As a fundamental of the ignition-coil model the heat-energy of the ignition-coils is calculated. It can differentiate between the homogenous and stratified mode.

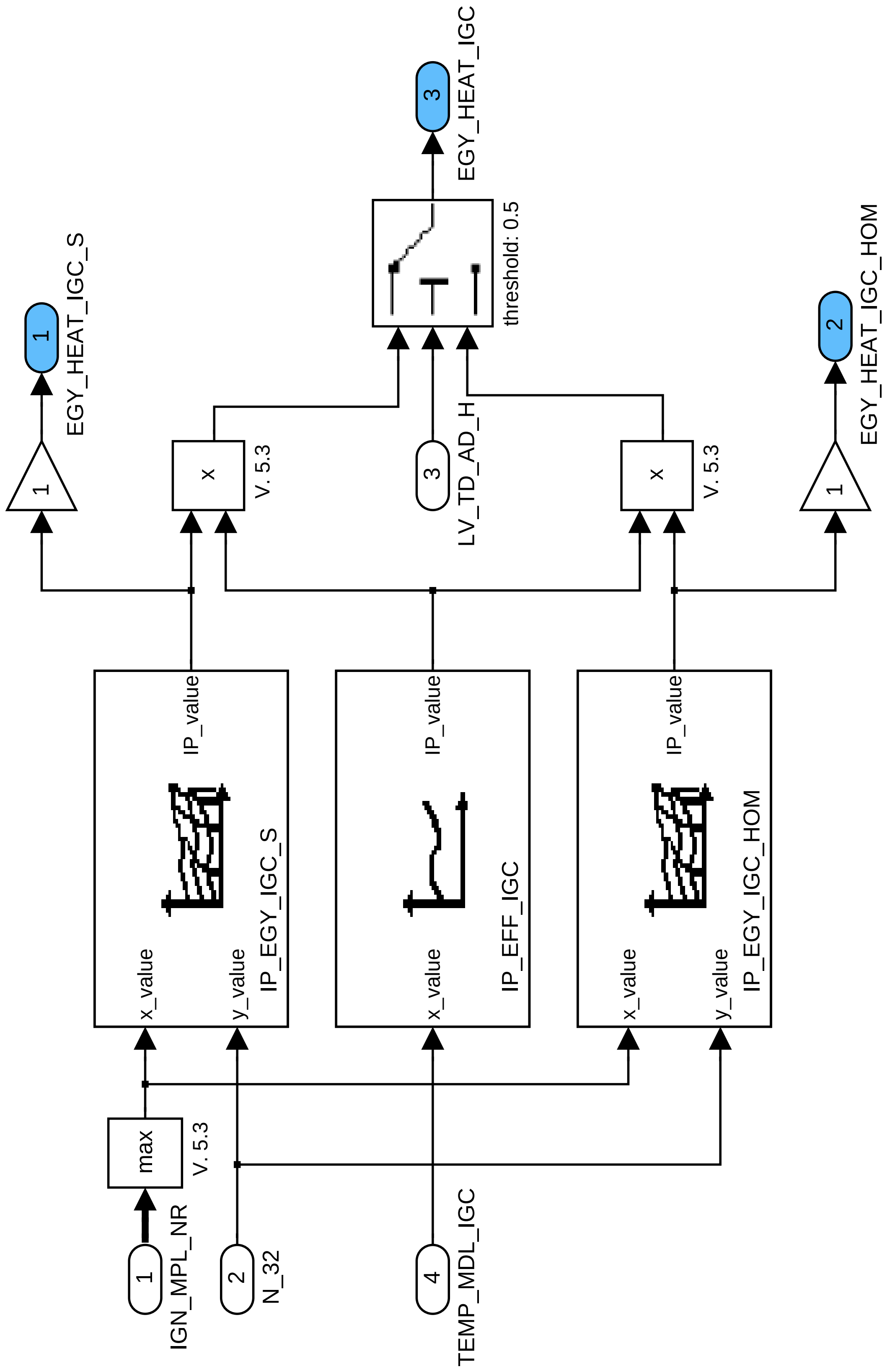


Figure 8:
Path: IGRE_M602J/OPM_10MS/CALC_TEMP_MDL/CALC/EGY_HEAT

38.14.3.1.1.2 Calculation of model-temperature

The model temperature of the ignition coils gets calculated in this function. Based on the heat-energy, the cooling temperature, and the last calculated model temperature the cooling-energy and the new model temperature get calculated. C_ALFA_A_IGC is the product of the heat-transfer-coefficient and the area of the coils and C_CP_M_IGC the product of the heat-capacity and the mass of the coils. The integrator-block generates the temperature.

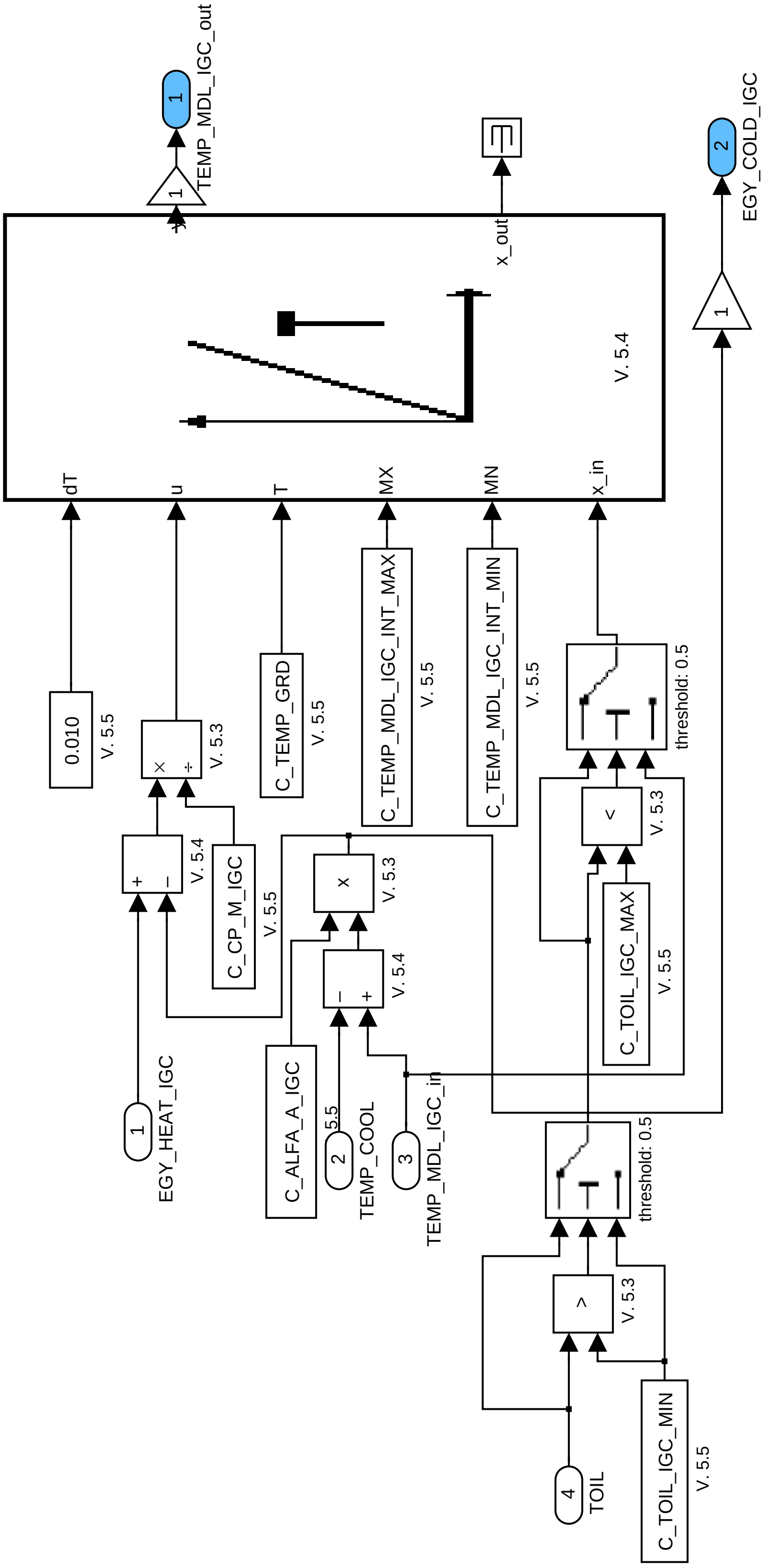


Figure 9:
Path: IGRE_M602J/OPM_10MS/CALC_TEMP_MDL/CALC/TEMP_IGC

38.14.3.2 Deactivation of multiple spark on ECU temperature threshold

The purpose of this function is to deactivate the multiple sparks above an ECU temperature threshold in order to decrease heat dissipation due to multiple sparks.

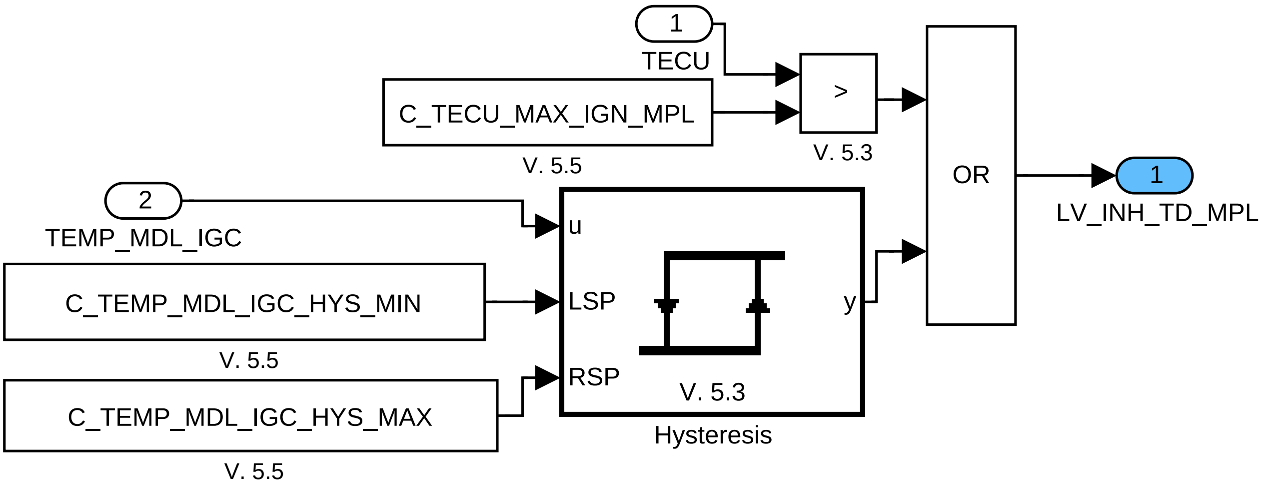


Figure 10:
Path: IGRE_M602J/OPM_10MS/TEMP_MPL_DEAC

38.14.3.3 Extended Multiple Spark function Activation

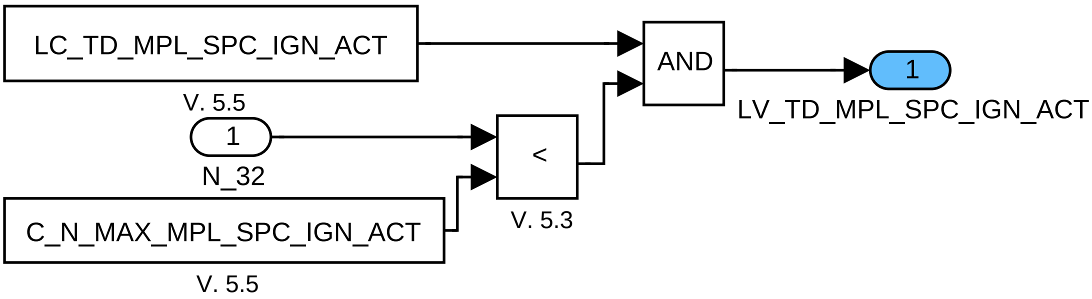


Figure 11:
Path: IGRE_M602J/OPM_10MS/MPL_SPC_IGN_ACT

38.14.3.4 Individual raise of Multiple Spark in case of misfiring

The purpose of this function is to set a calibrated number of multiple spark on a specific cylinder in case of misfiring.
The selection of cylinder is done by a word: STATE_MPL_CYL_IGN. Each bit codes a cylinder, the first cylinder is code by the lowest bit. If the bit is set to 1, there is misfiring and we set the number of MPL for this cylinder to a calibrated value. If it's set to 0, the number of MPL in this specifications will be set to 0 and will be replace in the generic specifications by an appropriate value based on a interpolation table depending on engine speed and battery voltage.

38.14.3.4.1 Set INJ_MOD_MPL to NC_INJ_MOD_S

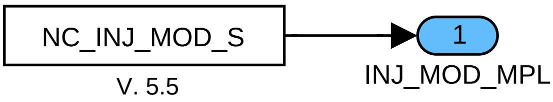


Figure 12:
Path: IGRE_M602J/OPM_10MS/INJ_MOD_MPL/SET_NC_INJ_MOD_S

38.14.3.4.2 FOR-Loop for calculation for every cylinder

The calculation shall be done for every cylinder.

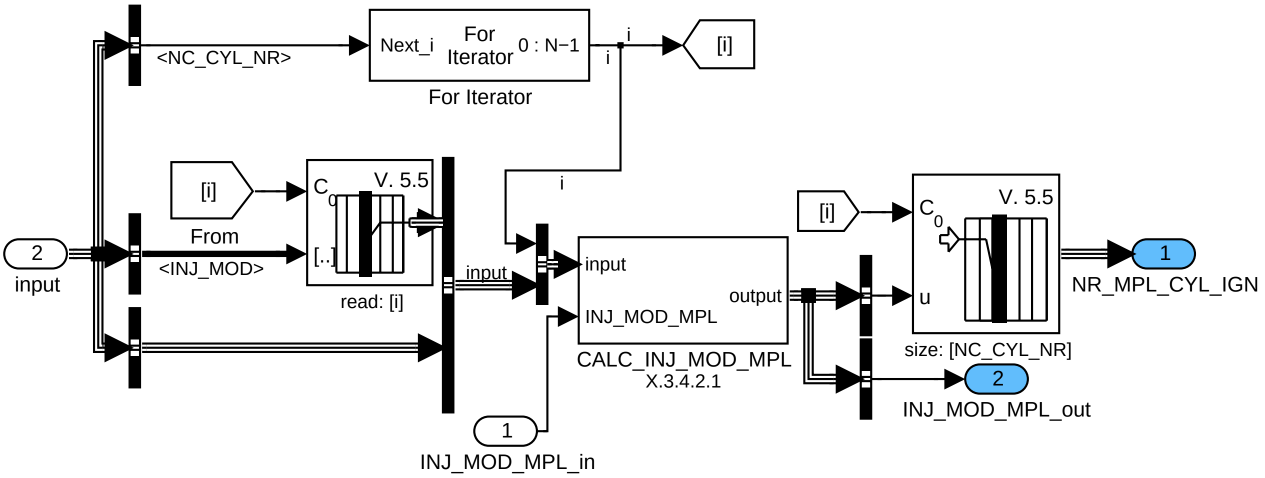


Figure 13:
Path: IGRE_M602J/OPM_10MS/INJ_MOD_MPL/FOR_INJ_MOD_MPL

38.14.3.4.2.1 Calculation-Subsystem in FOR-Loop

38.14.3.4.2.1.1 Calculation of Injection-Mode and Number of Multiple sparks

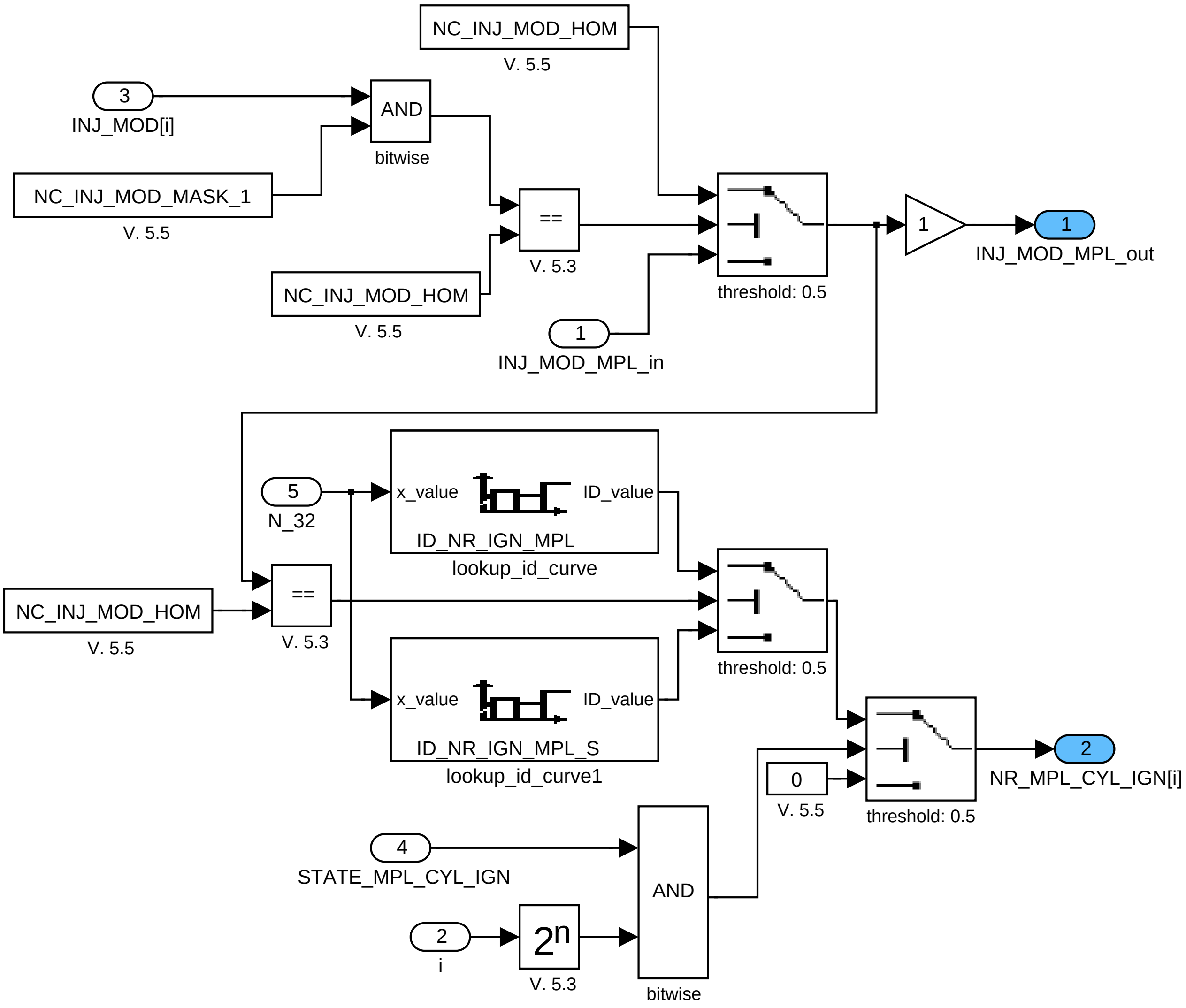


Figure 14:
Path: IGRE_M602J/OPM_10MS/INJ_MOD_MPL/FOR_INJ_MOD_MPL/CALC_INJ_MOD_MPL/CALC

38.15 Ignition activation control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
INH_IGC	V	0...FFH	0...255	1	[-]
Inhibition of selected cylinder					
CTR_CYCNR_IGA_ES	V	0...FH	0...15	1	[-]
Counter for number of cycles for which ignition is maintained after detection of LV_IGK= 0 or LV_STST_STOP_REQ= 1					

Input data:

LV_INH_IGC[NC_CYL_NR]	INH_IGC_MIS_GEN	LV_SYN_ENG	LV_LOCK_IMOB
NC_CYL_NR	LV_IGN_INJ_LOCK_REQ	LV_FIRST_VLD_TOOTH	LV_IGK
LV_STST_STOP_REQ	LV_ST_END		

Import actions:

ACTION_INFR_SetIgnEnable(IN <>, IN <>)

38.15.1 Normal Mode Activation

FUNCTION DESCRIPTION:

As there could be different possibilities to deactivate the ignition system an **OR-Junction** is done in this chapter. The output of this or-junction has to be INH_IGC.

The Inputs can come from the ignition diagnosis, from the injection chapter, from the Misfire Generator or from any other function requesting ignition cut-off.

Application conditions:

Initialisation:	at reset

Recurrence:	if LV_ST_END = 1and LV_IGN_INJ_LOCK_REQ = 0:
	segment synchronous (every TDC)
	if LV_ST_END = 0 or LV_IGN_INJ_LOCK_REQ = 1:
	10 ms

Activation:	LV_SYN_ENG = 1
Deactivation:	LV_SYN_ENG = 0

Formula section:

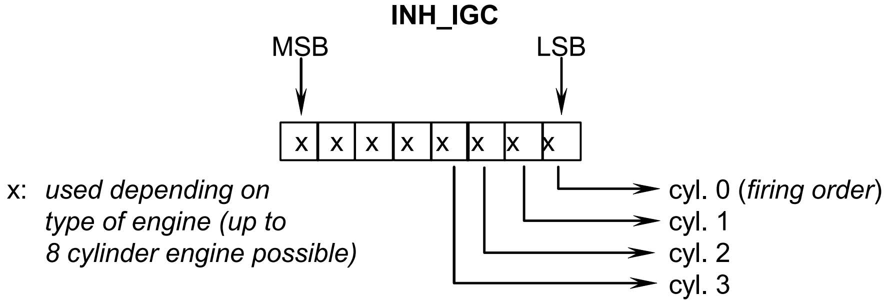
Output driver deactivation

The decision to inhibit an ignition stage is taken for each output before the turn on at TD_ON[x]. This means that no Disable / cut-off can be set on a coil while charging

INH_IGC is the bit wise OR of all inputs requesting an ignition cut-off.

LV_INH_IGC[x] sets the INH_IGC for the corresponding cylinder to 1.

When one bit within INH_IGC is set to 1, the corresponding ignition output is performed as soon as possible. The ignition output remains inhibited while the INH_IGC is unchanged.



- *Calculation of CTR_CYCNR_IGA_ES*

```
IF (LV_IGK$_n$ = 0 and LV_IGK$_{n-1}$ = 1)
   or (LV_STST_STOP_REQ$_n$ = 1 and LV_STST_STOP_REQ$_{n-1}$ = 0)

THEN CTR_CYCNR_IGA_ES$_n$ = C_CYCNR_IGA_ES

ELSE  IF (LV_IGK$_n$ = 0 and LV_IGK$_{n-1}$ = 0)
         or (LV_STST_STOP_REQ$_n$ = 1 and LV_STST_STOP_REQ$_{n-1}$ = 1)

      THEN  CTR_CYCNR_IGA_ES$_n$ = CTR_CYCNR_IGA_ES$_{n-1}$ -1

      ELSE  CTR_CYCNR_IGA_ES$_n$ = CTR_CYCNR_IGA_ES$_{n-1}$

      ENDIF

ENDIF
```

- *Inhibition for ignition system of cylinder x*

```
IF LV_LOCK_IMOB = 1
   or   LV_IGN_INJ_LOCK_REQ = 1
   or   LV_FIRST_VLD_TOOTH = 0
   or   (CTR_CYCNR_IGA_ES = 0 and LV_IGK = 0)
   or   (CTR_CYCNR_IGA_ES = 0 and LV_STST_STOP_REQ = 1)

THEN  INH_IGC = $2^{NC\_CYL\_NR} - 1$   ( ignitions on all the cylinders are inhibited )
ELSE
      FOR x = 0 to NC_CYL_NR-1
            IF LV_INH_IGC[x] = 1
            or bit x of INH_IGC_MIS_GEN = 1
                  THEN the corresponding bit / Cylinder x of INH_IGC is set to 1
                  ELSE  the corresponding bit / Cylinder x of INH_IGC is set to 0
            ENDIF
      ENDFOR
ENDIF

FOR   x = 0 to NC_CYL_NR-1
      ACTION_INFR_SetIgnEnable(x, not(corresponding bit x of INH_IGC))
ENDFOR
```

Note:

1. For DI engine it is strongly recommended to enable Ignition only after Validated synchronisation, *ie.* activation of this function on LV_SYN_VLD = 1 only instead of LV_SYN_ENG
2. No deactivation is possible on a charging coil, *ie.* after TD_ON[x] event: if Dwell Time has started on a stage, this charging will be elapsed to its end
   Refer to module 200X for more details

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CYCNR_IGA_ES	1	0...FH	0...15	1	[-]
Number of cycles for which ignition is maintained after detection of LV_IGK = 0					

38.16 Ignition diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CDN_DIAG_IGC_SCP[NC_CYL_NR]	V/O	0H 1H	Passive Active	1	-
Ignition diagnosis condition for short circuit plus					
LV_CDN_DIAG_IGC_SCG[NC_CYL_NR]	V	0H 1H	Passive Active	1	-
Ignition diagnosis condition for short circuit to ground					
LV_CDN_DIAG_DUR_IGC_MPL	V	0H 1H	Passive Active	1	-
Diagnosis condition bit for open load on all cylinders					
ERR_SYM_IGC_SCP[NC_CYL_NR]	V	0H 1H	NO_SYM SYM_0	1	-
Symptom of the ignition coil failure primary short circuit plus					
ERR_SYM_IGC_SCG[NC_CYL_NR]	V	0H 2H	NO_SYM SYM_1	1	-
Symptom of the ignition coil failure primary short circuit ground					
ERR_SYM_DUR_IGC_MPL	V	0H 8H	NO_SYM SYM_3	1	-
Symptom of the ignition coil failure : open load on all cylinders					
LV_ERR_IGC_SCP[NC_CYL_NR]	V/O	0H 1H	Passive Active	1	-
Ignition coil failure short circuit plus					
LV_ERR_IGC_SCG[NC_CYL_NR]	V/O	0H 1H	Passive Active	1	-
ignition coil failure primary short circuit ground					
LV_ERR_DUR_IGC_MPL	V/O	0H 1H	Passive Active	1	-
Ignition coil failure : open load on all cylinders					
LV_END_DIAG_IGC_SCP[NC_CYL_NR]	V	0H 1H	Passive Active	1	-
Result of ignition diagnosis for SCP					
LV_END_DIAG_IGC_SCG[NC_CYL_NR]	V	0H 1H	Passive Active	1	-
Result of ignition diagnosis for SCG					
LV_END_DIAG_DUR_IGC_MPL	V	0H 1H	Passive Active	1	-
Result of ignition diagnosis : open load on all cylinders					
IGC_DIAG_MIS	O/V	0...FFH	0...255	1	-
Non filtered error information for misfire detection					
LV_INH_IGC[NC_CYL_NR]	O/V	0...1H	0...1	1	-
Inhibition for ignition system of cylinder x					
LV_ERR_IGC	O/V	0...1H	0...1	1	-
OR of every failure every 100 ms					
LV_INH_INJ_DIAG_IGC	O/V	0...1H	0...1	1	-
Flag to inhibit injection on all cylinders due to lost ignition coil supply					

Input data:

V_DUR_IGC[NC_CYL_NR]	LV_ST	LV_CDN_VB_OBD1	TCO
LV_SCG_IGC[NC_CYL_NR]	LV_ES	LV_ERR_CAM	LV_IGK
LV_SCP_IGC[NC_CYL_NR]	N_32	LV_ES	LV_IGC_x_EXT_ADJ

IGC_x_EXT_ADJ	LV_INH_DIAG_DUR_IGC_MPL	LV_INH_DIAG_IGC_SCG	LV_INH_DIAG_IGC_SCP
TD_AD[NC_CYL_NR]	TD_IGC[NC_CYL_NR]	NC_CYL_NR	

FUNCTION DESCRIPTION:

General information:

The purpose of this diagnosis function is to detect all major failures, which can happen between ECU output and ignition coils.

This function is adaptable on every engine but the type of spark plugs and ignition coils must be taken into account.

The diagnosis is performed separate for each ignition coil. The feedback signal from the specific ignition coil (following the firing order) is evaluated by the microcontroller.

38.16.1 Short circuit to plus: SCP

Description:

In some hardware driver the ignition power stage is not protected against short circuit to ground and they are not able to withstand an overload current for a long time so it is necessary to switch off the power stage by software.

For ignition coil actuator tests purpose, the SCP diagnostic function has to be enabled, activated and monitored upon the tests requests

The same case and comments (see before) apply here also.

The Ignition output diagnosis function detects short circuit to battery and sets LV_IGC_SCP[x]. It cuts off the ignition driver after a delay time which is defined in the IGBT Protection function. In the ignition diagnosis function the error is debounced.

Application conditions:

1. **Normal Engine Running mode**

Initialization: all ERRM output data according Filter type (see Applic. Inc.), all other output data 0 at reset

Recurrence: **If** NC_IGN_DIAG_TYP =ATM46 every segment

If NC_IGN_DIAG_TYP =ATIC29 every 100 ms

Activation: **If** LV_INH_IGC[x] = 0

And LV_ERR_CAM = 0

And N_32 > C_N_32_MIN_SCP_IGC_DIAG

And LV_INH_DIAG_IGC_SCP = 0

And **if** NC_IGN_DIAG_TYP =ATIC29
Failure information for current cylinder
from ATIC29 is valid

```
                    Endif
        Then        LV_CDN_DIAG_IGC_SCP[x] = 1
        Else        LV_CDN_DIAG_IGC_SCP[x] = 0
        Endif
```

2. **Ignition Actuator Tests mode**

Initialisation: all output variables of the function have to be set to 0 at reset ECU *(except saved error flag)*

Recurrence: **if** NC_IGN_DIAG_TYP =ATM46
On transition of IGC_x_EXT_ADJ from 1 to 0
for reading the SCP failure status

Activation:

```
For         x = 0 to NC_CYL_NR – 1
    If          LV_INH_IGC[x] = 0
        And     LV_ERR_CAM = 0
        [And    LV_ES       = 1          // Actuator tests activation
        And     LV_IGK      = 1
        And     LV_IGC_x_EXT_ADJ = 1]
    Then        LV_CDN_DIAG_IGC_SCP[x] = 1
    Else        LV_CDN_DIAG_IGC_SCP[x] = 0
    Endif
EndFor
```

Formula section:

```
If      LV_CDN_DIAG_IGC_SCP =1
Then    If      LV_SCP_IGC[x] = 1
        Then    ERR_SYM_IGC_SCP[x] = SYM_0
        Else    ERR_SYM_IGC_SCP[x] = NO_SYM
Else    No action on the antibounce counter
Endif
```

Debounce:

Anti - bounce counter increment: according ABC type

Maximum value of anti - bounce counter: according ABC type

```
If      LV_CDN_DIAG_IGC_SCP[x] =1
Then    If      ERR_SYM_IGC_SCP[x] = SYM_0
        Then the antibounce counter is started according ABC type
        Else The antibounce counter is decremented according ABC type
        Endif
Endif
```

Ignition cut off conditions:

If	C_ABC_MAX_IGC_SCP delay is reached
Then	LV_INH_IGC[x] = 1
Endif	

Ignition restart after cut off:

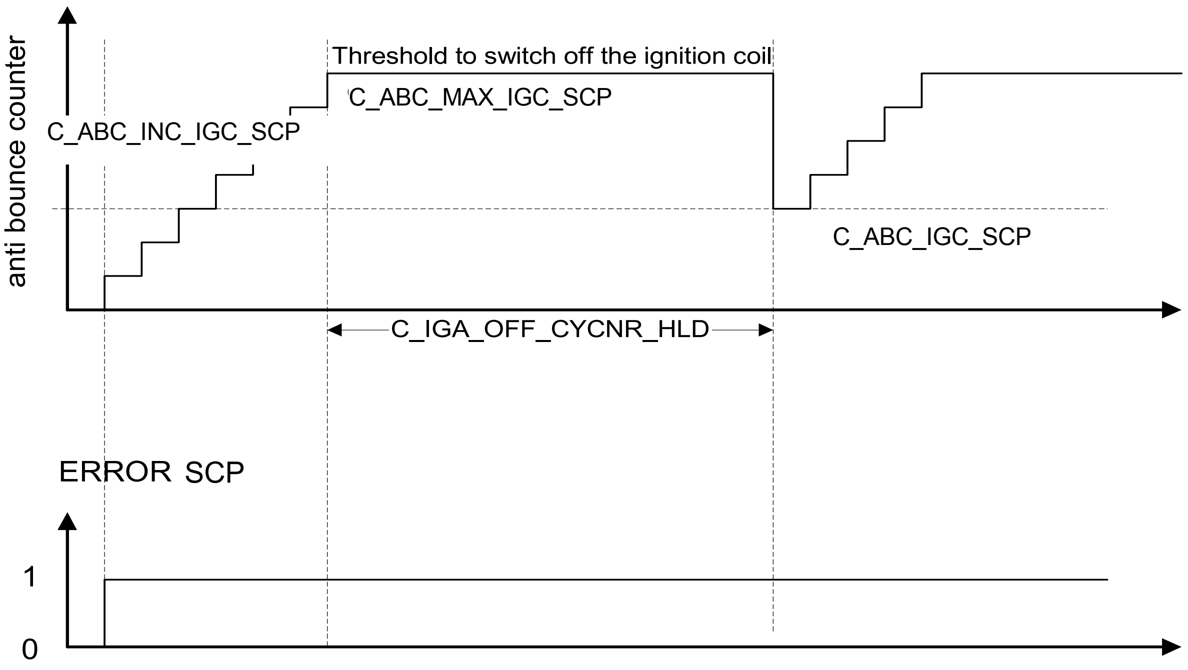
To regenerate the power stage the diagnosis will be switched off (LV_INH_IGC[x] = 1) for C_IGA_OFF_CYCNR_HLD recurrence number (see graphic, failure reaction in case of error).

After the refreshing cycle the debounce counter is set to C_ABC_IGC_SCP (see graphic below) and the ignition will be switched on to test the operating stage again. In case of error it will be debounced again.

This procedure is executed C_SUM_IGC_SCP times before the coil is absolutely switched off, until reset for ATM46 not for ATIC29

In next engine cycle the error is kept present. To take sure that the vehicle is able to restart the ignition will be switched on, till the diagnosis conditions are fulfilled and the diagnosis restarts.

Failure reaction in case of error



Application hint:

Some driver as ATM46 does not cut itself in case of SCP failure. For better Driver protection The counter threshold C_ABC_IGC_SCP mustn’t exceed 4 cycles For the same reason the counter C_ABC_MAX_IGC_SCP must be set below 12 cycles.
The function can be disabled by the ‘enable byte’ C_ESB_IGN_CUT_DIAG.

38.16.2 SCG diagnosis

Description

The diagnosis is based on SCG information from the ignition output diagnosis function.

Please note that this diagnosis is valid also for Open Load detection.

Application conditions:

Initialisation: all ERRM output data according Filter type (see Applic. Inc.), all other output data 0 at reset

Recurrence: *if NC_IGN_DIAG_TYP =ATM46 every segment*

if NC_IGN_DIAG_TYP =ATIC29 every 100 ms

Activation:

```
If      LV_ES = 0
And     LV_ST = 0
And     N_32 > C_N_32_MIN_OL_SCG_IGC_DIAG
And     TCO > C_TCO_MIN_IGC_DIAG
And     LV_INH_IGC[x] = 0
And     LV_ERR_CAM = 0
And     No low voltage battery (LV_CDN_VB_OBD1 = 1)
And     Counter Number of recurrence > NC_IGC_DLY
And     LV_INH_DIAG_IGC_SCG = 0
And     if      NC_IGN_DIAG_TYP =ATIC29
                Failure information for current cylinder
                from ATIC29 is valid
        endif
Then    LV_CDN_DIAG_IGC_SCG[x] =1
Else    LV_CDN_DIAG_IGC_SCG[x] = 0
Endif
```

Formula section:

```
If      LV_CDN_DIAG_IGC_SCG[x] =1
Then    If          LV_SCG_IGC[x] = 1
        Then        ERR_SYM_IGC_SCG[x] = SYM_1
        Else        ERR_SYM_IGC_SCG[x] = NO_SYM
        Endif
Else    No action on the antibounce counter
Endif
```

Debounce:

Anti - bounce counter increment: C_ABC_INC_IGC_SCG

Maximum value of anti - bounce counter: C_ABC_MAX_IGC_SCG

```
If      LV_CDN_DIAG_IGC_SCG[x] =1
  Then  If    ERR_SYM_IGC_SCG[x] = SYM_1
```

```
            Then the antibouce SCG counter is started according ABC type
            Else The antibounce SCG counter is decremented according ABC type
            Endif
 Endif
```

38.16.3 Loss of power supply for all ignition coils

Description:

If all flags V_DUR_IGC[x] are equal to 0 at the same time it is assumed that the power supply for all ignition coils is lost or damaged. in this case LV_ERR_DUR_IGC_MPL is set.

Application conditions:

Initialisation: according ABC type

LV_INH_INJ_DIAG_IGC = 0 at LV_IGK 0 -->1 or reset or clearing FMY

Recurrence: every segment

Activation:

```
    If          LV_ES = 0
          And   LV_ST = 0
          And   LV_IGK = 1
          And   LV_INH_DIAG_DUR_IGC_MPL = 0
    Then        LV_CDN_DIAG_DUR_IGC_MPL = 1
    Else        LV_CDN_DIAG_DUR_IGC_MPL = 0
    Endif
```

Formula section:

```
If      LV_CDN_DIAG_DUR_IGC_MPL =1
Then    If      all V_DUR_IGC[x] = 0 ms                 // for all cylinders

        Then    ERR_SYM_DUR_IGC_MPL =  SYM_3
                If      LV_ERR_DUR_IGC_MPL = 1
                Then    LV_INH_INJ_DIAG_IGC = 1
                Endif
        Else    ERR_SYM_DUR_IGC_MPL = NO_SYM
                LV_INH_INJ_DIAG_IGC = 0
        Endif
Else    No action on the antibounce counter
Endif
```

LV_ERR_DUR_IGC_MPL and LV_END_DIAG_DUR_IGC_MPL are calculated by error management.

38.16.4 Or of ignition failure

Initialisation:	output variable has to be set to "0" at reset
Recurrence:	**If** NC_IGN_DIAG_TYP =ATM46 every segment
	If NC_IGN_DIAG_TYP =ATIC29 every 100 ms

Activation:

Formula section:

LV_ERR_IGC = LV_ERR_IGC_SCP[x] **OR** LV_ERR_IGC_SCG[x]

38.16.5 Unfiltered ignition error information for misfire detection

Application conditions:

Initialisation:	IGC_DIAG_MIS = 0 at reset ECU
Recurrence:	**If** NC_IGN_DIAG_TYP =ATM46 every segment
	If NC_IGN_DIAG_TYP =ATIC29 every 100 ms

Formula section:

```
If      LV_SCP_IGC[x]     = 1
Or      LV_SCG_IGC[x]     = 1
Then    set the bit for the actual cylinder in IGC_DIAG_MIS = 1
Else    reset the bit for the actual cylinder in IGC_DIAG_MIS = 0
Endif
```

Description:

The error information of the ignition system should not be filtered. The different error detections are linked by a wired "OR". As soon as an ignition error is detected the information should be available as output cylinder individually in the byte IGC_DIAG_MIS.

IGC_DIAG_MIS

Bit Low to high	4 cylinder	5 cylinder	6 cylinder	8 cylinder
0	Cyl 0	Cyl 0	Cyl 0	Cyl 0
1	Cyl 1	Cyl 1	Cyl 1	Cyl 1
2	Cyl 2	Cyl 2	Cyl 2	Cyl 2
3	Cyl 3	Cyl 3	Cyl 3	Cyl 3
4		Cyl 4	Cyl 4	Cyl 4
5			Cyl 5	Cyl 5
6				Cyl 6
7				Cyl 7

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_SUM_IGC_SCP	1	0...FFH	0...255	1	-
Number of cycles before switching off the coil until reset Typical value for debug = 255					
C_ESB_IGN_CUT_DIAG	1	0...01H	0...1	1	-
C_ESB_IGN_CUT_DIAG = 1 will inhibit the function Typical value for debug = 0					
C_TCO_MIN_IGC_DIAG	1	0..FEH	-48..142.5	0.75	°C
Min Temperature for condition diagnosis OL & SCG Typical value for debug = 20 ° (emission cycle starting)					
C_N_32_MIN_SCP_IGC_DIAG	1	0..FFH	0..8160	32	Rpm
Min engine speed for condition diagnosis SCP Typical value for debug = 500 rpm					
C_N_32_MIN_OL_SCG_IGC_DIAG	1	0..FFH	0..8160	32	Rpm
Min engine speed for condition diagnosis OL/SCG Typical value for debug = 500 rpm					
C_IGA_OFF_CYCNR_HLD	1	0...FFH	0...255	1	-
Duration for regeneration cycle Recurrence number; coil has been switched off Typical value for debug = 16					
C_ABC_IGC_SCP	1	0...FFH	0...255	1	-
Value of anti-bounce counter after regeneration cycle Typical value for debug = 128					

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_IGC_DLY	1	0...FFH	0...255	1	-
Number of recurrence after engine start, which had to pass by to activate the diagnosis value =16					
NC_IGN_DIAG_TYP	1	0..FF	0..255	1	-
ATM46 = 01H ATIC29 = 02H SHUNT (not supported today) Typical value for debug = ATIC29					

Note: typical value for debug is proposed for default value of the Calibration

38.17 Application incidences for ignition diagnosis

38.17.1 IGBT protection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
INH_IV_IGC	V/O	0...FFH	0...255	1	[-]
Injection cut off					

Input data:

LV_INH_IGC[NC_CYL_NR]	LV_INH_INJ_DIAG_IGC		

Description:

To protect the catalyst in case of ignition driver cut off it is necessary to shut off the injection of the concerning cylinder as well.

Recurrence: Segment Synchronous

Formula section:

Shut off injection sequence of IGBT protection:

```
if      LV_INH_INJ_DIAG_IGC = 1
then    set all bits  in  INH_IV_IGC
else    reset all bits  in  INH_IV_IGC
        for x = 0 to NC_CYL_NR-1
            if     LV_INH_IGC[x] = 1
            then   set bit    x  in  INH_IV_IGC
            else   reset bit  x  in  INH_IV_IGC
            endif
        endfor
endif
```

38.17.2 Inhibition flags for ignition diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_IGC_SCP	V/O	0...1H	0...1	1	[-]
Inhibition condition for the short circuit plus for ignition diagnosis					
LV_INH_DIAG_IGC_SCG	V/O	0...1H	0...1	1	[-]
Inhibition condition for the short circuit to ground for ignition diagnosis					
LV_INH_DIAG_IGC_OL	V/O	0...1H	0...1	1	[-]
Inhibition condition for open load for ignition diagnosis					
LV_INH_DIAG_DUR_IGC_MPL	V/O	0...1H	0...1	1	[-]
Inhibition condition for diagnosis of open load on all cylinders					

Input data:

LV_IGK	LV_ERR_TCO	LV_ERR_CRK	LV_CDN_VB_OBD2

Description:

Application conditions:

Initialisation: At reset event :

LV_INH_DIAG_IGC_SCP = 0

LV_INH_DIAG_IGC_SCG = 0

LV_INH_DIAG_IGC_OL = 0

LV_INH_DIAG_DUR_IGC_MPL =0

Recurrence: 10 ms

Activation:

Deactivation:

Formula section:

```
if      LV_IGK = 1          and
        LV_ERR_TCO = 0      and
        LV_ERR_CRK = 0

then    LV_INH_DIAG_IGC_SCP = 0
else    LV_INH_DIAG_IGC_SCP = 1
endif

if      LV_CDN_VB_OBD2 = 1       and
```

```
        LV_ERR_TCO = 0          and
        LV_ERR_CRK = 0
then    LV_INH_DIAG_IGC_OL = 0
else    LV_INH_DIAG_IGC_OL = 1
endif
LV_INH_DIAG_IGC_SCG = LV_INH_DIAG_IGC_OL
```

38.17.3 Configuration for diagnostic symptoms

Diagnosis	Symptom	SYM	ABC type
Ignition diagnosis for SCP	SCP	*SYM_0*	STD_INI
LV_ERR_IGC_SCP			

Diagnosis	Symptom	SYM	ABC type
Ignition diagnosis for OL and SCG	OL	*SYM_2*	STD_INI
	SCG	*SYM_1*	
LV_ERR_IGC_OL LV_ERR_IGC_SCG			

Diagnosis	Symptom	SYM	ABC type
Ignition diagnosis for OL on all cylinders	MPL_OL	*SYM_3*	STD_INI + DEC_CAL
LV_ERR_DUR_IGC_MPL			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_IGC_SCP	0	0...FFH	0...255	1	[-]
Anti ¿ bounce counter increment for the ignition system diagnosis SCPTypical value for debug = 5					
C_ABC_INC_IGC_OL	0	0...FFH	0...255	1	[-]
Anti ¿ bounce counter increment for the ignition system diagnosis OL Typical value for debug = 5					
C_ABC_INC_IGC_SCG	0	0...FFH	0...255	1	[-]
Anti bounce counter increment for the ignition system diagnosis SCG Typical value for debug = 5					
C_ABC_INC_DUR_IGC_MPL	1	0...FFH	0...255	1	[-]
Anti ¿ bounce counter increment for the ignition system diagnosis multiple open loadTypical value for debug = 5					
C_ABC_DEC_DUR_IGC_MPL	1	0...FFH	0...255	1	[-]
Anti ¿ bounce counter decrement for the ignition system diagnosis multiple open loadTypical value for debug = 5					
C_ABC_MAX_IGC_SCP	0	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter for the ignition system diagnosis to switch off the coilTypical value for debug = 255					
C_ABC_MAX_IGC_OL	1	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter for the ignition system diagnosis OLTypical value for debug = 255					
C_ABC_MAX_IGC_SCG	1	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter for the ignition system diagnosis SCGTypical value for debug = 255					
C_ABC_MAX_DUR_IGC_MPL	1	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter for the ignition system diagnosis multiple open loadTypical value for debug = 255					

38.18 Calibration hints for function IGRE – Dwell time period for stratified mode (open loop) - 601L

The same process and tuning as for the Homogenous mode has to / can be followed just considering here the specific needs from the S mode: specific spark energy to be delivered by the coil in order to insure a good burning in S mode.

Consequently, please refer to the module Q018 for this calibration process.

For the stratified mode, it could be interesting to estimate the combustion regularity of the engine, based on an acquisition of the pressure during each cycle, or a bench torquemeter. Compare to homogeneous mode, the energy has to be carefully checked for S needs and also the perfect ignition synchronisation has to be adapted to have a good combustion.

Generally:

IP_TD_S	Dwell time value versus battery voltage and engine speed for stratified mode Normally bigger than in HOM mode ~5 to 4ms decreasing with Engine Speed and Load increase
C_TD_S_INI	Initial value for dwell time control Normally, same value chosen for homogeneous and stratified mode ~5ms
C_TD_S_AS	Application intervention for stratified mode Only for the Calibration phase for testing and validation on engine set to 0 ms

38.19 Calibration hints for function IGRE – Dwell time factor calculation (Template)

To improve Ignition capability of the mixture, especially during Start phase, a specific boost factor will / can be applied to the Dwell Time charging of the coil in order to increase the energy provided in the combustion chamber by the spark.

Care has to be taken when setting such a boost factor on dwell time not to over-stress the coil (*coil internal temperature increase by energy dissipation*) due to the increased dwell time resulting from this boost factor application.

General recommendation from previous experiences and coil / engine test leads to have:

IP_TD_ST_FAC	Factor for dwell time boost at start ~0,5 to 0,25 decreasing when engine speed increase

This file (6025) is a Customer version. For any other boost factor on Dwell Time that could be applied by project choice, please refer to the customer-related file and settings.

38.20 Calibration hints for function IGRE – Dwell time control (open loop)

To ensure the correct ignition of the spark plug on an engine, it is necessary to calibrate different system parameters. This calibration will depend of engine type (direct Injection, homogeneous injection, number of cylinders, capacity).

Specific tests are done, for each application, in order to check if the calibrated dwell times are suitable to the requirements decided by the customer. The tests must consider the conditions for the worst case coil. This also leads to verify if the nominal mapping is coherent when applied on the engine and SOP components, in real running conditions.

Moreover, tests are achieved to show if there is a need of HT or spark duration improvement for transients or cold conditions or stratified mode. One possibility is then to use for example the Multiple Spark Strategy.

The definition and settings of Dwell time control in open loop (6002) is generally performed on a coil test bench in order to simplify the process and to not block an engine or a car for these test cases.

Ignition requirements for the Engine

Depending on the engine (displacement, number of cylinder), injection (classic, direct), combustion (homogeneous or stratified), the ignition requirements for the engine will be specific. Main parameters to ensure the ignition are the secondary performances of the coil: the high tension and energy values necessary for the engine combustion must be inquire by the engine carmaker.

Before calibrate the dwell, although the choice of the coil is not our responsibility, we have to check globally that the coil is adapted to get the performance with correct system condition (primary current not too high).

The customer, responsible of the coil choice and engine design has to provide a target for the spark energy to reach or equivalently for the 1ary current at end of charge time. The Dwell Time setting will then be mapped accordingly to this target taking care to not over-stress the coil (*coil internal temperature increase by energy dissipation*). Description to be given for priority for technical requirements is established, as following, by the customer:

1) high voltage (HT)
2) spark duration
3) spark current: in peak (for a good engine cycles regularity)
4) spark energy...

These data are the base of the primary current definition made by the coil supplier.

Parts

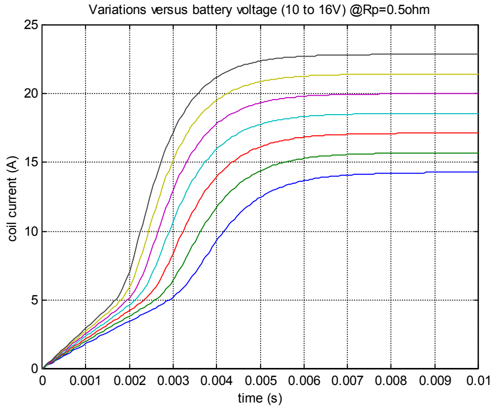
Due to the tolerance of the coils, it is necessary to have several sample coils with different characteristics (Minimum coil, Nominal coil, Maximum coil: in term of inductance).

As the primary current will depend on the coil's dispersion, as far as possible, use nominal coils to do the tests. In other case it is necessary to know the properties of the coil tested (nearest of mini or maxi) to adapt the calibration.

Standard Characterisation

Ignition coil is a transformer where energy stocked during the primary charge is transferred to the secondary and made the high tension necessary for the spark plug. The energy is given by: $E_{tot} = E_{stock} + E_{dis}$ with E_{stock}: stocked energy in the coil

E_{dis}: dispersed energy



- During the primary current charge, energy is stocked. In the “full saturation zone”, stocked energy in the coil is flat because the most influence parameter is the resistance of the coil and inductance L is very low. So in this zone, the energy will be dissipated with heating proportionally to the square of the current. The working energy at the secondary will not increase.
- The saturation’s zone does not allow to stock energy: it is important to check the primary current operating zone not to risk an overheating of coil due to a maximal dissipation (not adapted dwell -> coil damaged), this zone has to be avoided.

The primary current in the coil depends of the voltage supply applied on the +: it will be necessary to compensate the supply (VB) variation in dwell mapping.

A thermocouple is stick on the surface of the coil to control the temperature. For coils with IGBT inside, a thermocouple has to be stick on the radiator.

To simulate the burn time, Zener diode and 1MO load are connected to the secondary of the coil. Only the coil is tested without ECU control: this characterisation is done on the ignition test bench with specifics loads at ambient temperature.

Remarks: it is important to use adapted wire for connecting. For the primary side coil use the appropriate connector with wire with sufficient cross-section (1.5mm² to 2mm²) to limit the electrical leakage. For the secondary side, use specific high-tension wire (no resistive).

if the coil includes the electronic module (IGBT...), tests will be done with it.

Those tests authorise:

- To have an over view of the performance of the coil (linear zone and saturation zone for the primary current, high tension, energy)
- Preliminary analyse for the calibration
- To compare easily the different parameters in same condition with other coils

Endurance Test

The coil is set in a box: 3mm diameter steel balls cover completely the coil. This box is connected to the ground. All external connection with the coil must be isolated of the steel balls. This test is hard to detect a problem of electrical insulation on a coil.

- Power supply: 14V
- Pulse generator: Square signal amplitude 5V
- Frequency 100Hz = 6000 rpm for 4cyl engine in semi-static ignition
- Frequency 50Hz = 6000 rpm for 4cyl engine: full static ignition
- Impulsion width tuned to have 7.5A pick current primary
- No resistive cable for secondary connection

With an X radiation analysis, impacts zone due to the stress are explored to determine the cause (plastic isolation, pick-up...)

Hot Test Sequence

The coil has to be connected to the ignition test bench.

Dwell calibration have to be done with temperature conditions near of heat engine condition (90°C, 100°C to define). For this, before the beginning of the first point, the coil have to be conditioned one hour in the test chamber at the required temperature. For coil with electronic driver inside, short circuit current of the coil (typically at the end of charge) must be conform to the current maximum specified for the driver. Power dissipation and temperature of the driver has to be check.

Depending on the coil type, but especially for double output coil with driver inside, the coil must operate one after one to have similar sequence in comparison with engine operating and to have a warm up more representative. If there is no driver inside the coil, calibration can be done only on one output.

Operating frequency (engine speed) has more impact for the coil temperature that the battery voltage supplier.

So, begin with the lower voltage and frequency and next increase the frequency by 2. The voltage must be measure directly on the + of the coil to have the real voltage and no influence of the electric leakage. After the last speed, increase the voltage by 2 volts steps.

For 24V battery, only test engine speed corresponding to the start-up and idle speed: in fact this calibration point is really important in case of start-up with additional battery.

Check that the primary current is not in full saturation (energy performance at secondary will be not improved and will risk to overheat the coil): the ideal area is in the first zone where the inductance is quasi constant (for a standard coil without hard permanent magnet) and the self has low dispersion. In term of energy the efficiency will be better in the linear area and globally the Energy = g * I. In the linear area, the dissipated temperature is lower. 15 minutes are necessary between each point to stabilise the temperature.

Measurement for each point (VB, N):
- Dwell time tuned (ms)
- Primary current at the end of the dwell (A)
- Burn time (ms)
- Secondary current (mA)
- Secondary voltage (V)
- Energy at secondary (mJ)
- Temperature at the surface of the coil (°C)
- Temperature of the radiator driver if the driver is inside the coil (°C).

Warm up test sequence

With the programmable test chamber, perform a heating sequence representative to the engine warm up (from –30°C to 90°C: ~4°C/min) in driving condition. This test will be done for “cold conditions voltage (11V)” and with an intermediary engine speed of 3000rpm.

Measure the primary current each minute and check it is not in full saturation: a record can be done with oscilloscope to see the current variation pick.

Standard Dwell calibration (VB, Engine speed)

Primary current on the coil is mainly dependants of 3 parameters:
- Battery voltage power source
- Time conduction (dwell)

- Coil’s temperature: this temperature depends on the engine speed (or frequency), ambient temperature of the coil (typically engine temperature) and dwell applied. With the temperature increasing, the primary current decreases (internal resistor increases): so, the performance of the coil will decrease with heating environment. Generally, limit operational temperature for the coil is around 130°C -140°C. As the actual strategies do not yet include compensation for coil's temperature, it is important to not calibrate a dwell for which the temperature on the coil will be too high: it will be a possible cause of failures on the coil. For cold conditions, primary current can be more important but operating point with primary current in fully saturation must be avoid.
  Remark: for the pencils coils or plug top modules which are inside spark-plug recess, the temperature of the coil will be superior compare to a coil only fixed externally.

IP_TD	Dwell time calibration versus battery voltage and engine speed As far as possible, it would be better to begin calibration with precalibrations (coil manufacturer or previous application using similar coil) Care has to be taken regarding the dissipation constraint on the power transistor: short circuit current of the coil (typically at the end of charge) must be conformed to the maximum current specified for the driver. Before the calibration the maximum current supported by the driver has to be know to adapt the dwell and have a short circuit primary current not too high! ~3 to 5 ms decreasing with Engine Speed and Load increase
LDPM_VB_1_IGRE	- 6V: discharged battery - 8V: low voltage - 10V to 11V: cold conditions (winter...) with engine accessories active - 12V: engine accessories active. - 14V: NOMINAL battery voltage in normal driving conditions - 16V: high voltage - 18V or 24V: in case of assistant start-up with 24V truck battery Theses ordinates could be adapted depending on specification. LDPM's have to be reserved for the behaviour in starting phase where jumps to low VB values occur
LDPM_N_32_1_IGRE	The “Volumetric Efficiency” mapping represents the image of the engine torque ramp change and can be use to select engine speed ordinates for the dwell calibration: - Engine start = 400 or 500 rpm - Idle speed (hot conditions) = 700 or 750rpm - Idle Speed (cold conditions) = 1200 or 1300rpm - Engine speed for partial load = 6 points by step (500rpm or 750rpm) until the maximum - Maximum Engine speed Select the most appropriate ordinates (between linear zone) and adapt the number of ordinates for the dimension of the dwell mapping (generally 10 points for the engine speed). Some LDPM's can be reserve for a better mapping of the Start phase. If S mode is also possible a finer grid of specific LDPM can be allocated to this S mode
C_TD_AS	Initial value for dwell time control Normally, same value chosen for homogeneous and stratified mode ~5ms
C_TD_INI	Application intervention in Homogenous mode Only for the Calibration phase for testing and validation on engine set to 0 ms

Engine Change Validation

The dwell calibration will be done normally on chassis dyno and exceptionally precalibration can be done on engine dyno: as the wiring harness for coil connection (battery, command, ground) between the 2 methods is not exactly identical, a complete checking has to be done with the vehicle in case of precalibration on chassis dyno.

Warning: For most of the applications, the customer takes the decision to adapt an ignition coil (and coil supplier) to its engine. This means that any modification of engine design or electrical components leads to a check of the calibrations of the dwell control.

38.21 Calibration hints for function IGRE – Ignition with multiple spark - 6003

In order to improve the combustion capability in some specific phases, it can be useful to use the Multiple Spark Strategy (MPL). MPL will result in:
- Increase of the spark energy provided to the mixture
- Artificial increase of the spark duration in the chamber

Those 2 effect leads to a better ignitability and burn of the air/fuel mixture.

It is recommended to use at least the MPL in:
- Start phase because of the strong variation occurring there
- Idle Speed for more engine stability
- Stratified mode (if applied) to be sure that at least 1 spark event will reach the fuel cloud

Depending on specific engine need and behaviour, those tuning and recommendations can be adapted following engine tests, cold and hot trip, etc…

Generally, from calibration settings, MPL are not dangerous for the coils. Nevertheless, care has to be taken when employing MPL to not over-stress the coil (*coil internal temperature increase by energy dissipation*) due to the increased energy stored in the coil.

The values here proposed have to be refined by measurements on engine, with the target of:
- Engine stability
- burn time increase by 2ary current measurement

C_TCO_MAX_MPL_ST	Max. Coolant Temp. to have MPL at start 142,5°C (full range application in Start)
C_TCO_MAX_MPL_IS	Max. Coolant Temp. to have MPL at Idle speed 142,5°C (full range application in Idle)
C_TCO_MAX_MPL_PL_PU_FL	Max. Coolant Temp. to have MPL at Run -48°C (not used)
C_TCO_MAX_MPL_PUC	Max. Coolant Temp. To have MPL at fuel cut off -48°C (not used)
C_VB_MAX_MPL_ST	Max. Battery voltage to have MPL at start requested value: 16V (full range application in Start)
C_VB_MAX_MPL_IS	Max. Battery voltage to have MPL at idle speed requested value: 16V (full range application in Idle)
C_VB_MAX_MPL_PL_PU_FL	Max. Battery voltage to have MPL at Run 0V (not used)
C_VB_MAX_MPL_PUC	Max. Battery voltage to have MPL at fuel cut off 0V (not used)
C_N_MIN_MPL_ST	Engine speed min for ignition with MPL at Start ~300rpm (electrical starter speed)
C_N_MAX_MPL_ST	Engine speed max for ignition with MPL at Start ~900rpm (max engine speed during flare up)
C_N_MIN_MPL_IS	Engine speed min for ignition with MPL at Idle ~500rpm (cf. setting for minimal engine speed of ISC)
C_N_MAX_MPL_IS	Engine speed max for ignition with MPL at idle ~900rpm (cf. setting for maximal engine speed of ISC)
C_N_MIN_MPL_PL_PU_FL	Engine speed min for ignition with MPL at Run 8160rpm (not used)
C_N_MAX_MPL_PL_PU_FL	Engine speed max for ignition with MPL at Run 0rpm (not used)
C_N_MIN_MPL_PUC	Engine speed min for ignition with MPL at fuel cut off 8160rpm (not used)
C_N_MAX_MPL_PUC	Engine speed max for ignition with MPL at Fuel cut off 0rpm (not used)
IP_TD_T_MPL	Minimum time between two ignition in homogeneous mode decreasing with N increase normal value 110µs - 150µs up to 180µs max.
IP_TD_T_MPL_S	Minimum time between two ignition in stratified mode decreasing with N increase normal value 110µs - 150µs up to 180µs max.

IP_TD_T_MPL_1	Minimum time in between main spark and first MPL decreasing with N increase and increasing with VB increase in H mode ~2 to 10 ms
IP_TD_T_MPL_1_S	Minimum time in between main spark and first MPL decreasing with N increase and increasing with VB increase in S mode ~2 to 10 ms
IP_TD_MPL	Dwell time for MPL in homogeneous mode run ~2 to 0,25ms (~1/10 to 1/15 of nominal TD) decreasing with VB and N increase
IP_TD_MPL_S	Dwell time for MPL in stratified mode run ~2 to 0,25ms (~1/10 to 1/15 of nominal TD) decreasing with VB and N increase
IP_TD_MPL_1	Dwell time for 1st MPL in H mode decreasing with N and VB increase ~10 to 1 ms
IP_TD_MPL_1_S	Dwell time for 1st MPL in S mode decreasing with N and VB increase ~10 to 1 ms
IP_NR_MPL	Number of successive MPL depending on N, MFF_SP_MV in H mode 7 to 3 decreasing with engine speed and TCO increase
IP_NR_MPL_S	Number of successive MPL depending on N, MFF_SP_MV in stratified mode 7 to 3 decreasing with engine speed and TCO increase
IP_FAC_TCO_NR_MPL	TCO factor used in H mode to calculate MPL number
IP_FAC_TCO_NR_MPL_S	TCO factor used in S mode to calculate MPL number
LDP_TCO_IP_NR_MPL	Refer to the setting for Ignition Correction versus temperature in IGSP
C_TD_MPL_MAN	Manual activation of the MPL 0 used only for test and calibration phase
C_T_CTR_TD_MPL_AST	Maintenance Time of MPL activation before Out Case and for the After Start phase only 0ms (not used)
C_T_CTR_TD_MPL	Maintenance Time of MPL activation before Out Case 0ms (not used)

38.22 Calibration hints for function IGRE – Dwell time adaptation – 202C

The primary current slope in fact is not only depending on the battery voltage, but also on many other parameter (e.g. coil inductance, coil and wiring harness resistance, coil temperature, etc.). Most of the parameter cannot be measured on the system, some have manufacturing tolerances and some also vary because of ageing.
This means that the programmed dwell-time table only contains typical values. Therefore the ignition break current can vary in the range of 20..30%. This causes unwanted effects on the system, such as low spark energy or high stress for the ignition power stage.
To solve this, an adaptive dwell-time control can be achieved by using the flag present ion ATIC71 in current controlled mode.

To perform an adaptive dwell-time control, one time is necessary:

- the time **t** during which primary current is above the 100% threshold

For the next ignition a correction can be done.
If **t** = 0 ms => the target threshold has not been reached and the dwell time has to be increased by C_TD_AD_INC

If **t** > C_TD_AD_THD => the target has been reached and we are above the dead zone (C_TD_AD_THD). In this case the dwell time will be reduced at the next ignition by **t** * C_FAC_DEC_TD_AD.

C_TD_AD_MIN	Minimum value learned for dwell time adaptative -3ms
C_TD_AD_MAX	Maximum value learned for dwell time adaptative 3ms
C_TD_AD_INC	Increment on dwell time in case current threshold is not reached 0.052ms
C_TD_AD_THD	Definition of threshold to enable dwell adaptation 0.1ms (defines a dead zone where no action is done)
C_FAC_DEC_TD_AD	Factor applied on T_THD_IGCFB to calculate time to be removed to TD_AD 0.98 (should be near to 1)
C_T_THD_IGCFB_MAX	Maximum measured time for valid sample 3ms
C_T_SUM_TD_AD_MAX	Maximum time allowed to be added in TD_AD without reaching target threshold 3ms
LC_TD_AD_CLR	Logical value enabling reset of variables concerning adaptive values 0 (not used)

38.23 Calibration hints for function IGRE – Dwell time adaptation (Appl.Inc) – 2052

C_TCO_AD_SWI	TCO threshold to enable switch from channel NC_TD_AD_CHN_H to NC_TD_AD_CHN_L -48°C (not used)

38.24 Calibration hints for function IGRE – Spark Advance and Dwell Output

Measurements and tests on an engine define settings of the related calibration values for the Spark Advance management in BSW (200X). The target is to have safe combustion (no misfire) and no coil over-stress with the selected parameters. Here are given normal values applied that should be refined on each engine following tests for combustion stability during transient phases (acceleration and deceleration).

C_TD_ FAC_ST_ MIN	Factor for minimum dwell time in Start phase = 0,5
C_TD_ FAC_ST_ MAX	Factor for maximum dwell time in Start phase = 1,5
C_TD_VB_MAX	Maximum battery voltage to activate MPL = 16V
IP_TD_FAC_MAX	Max dwell time factor in normal running The generic strategy is based on a priority of the ignition angle. If dwell time priority is requested the minimum dwell time has to be equal to the maximum dwell time. The min and Max factor defined here and applied on Dwell Time represent the excursion range window enabled for Ignition Change while the coil is charging To be adapted to coil capability: avoid coil over heating = 1,25 to 1,1 decreasing with engine speed increase
IP_TD_FAC_MIN	Min dwell time factor in normal running To be adapted to coil capability and engine need: avoid misfiring event = 0,7 to 0,8 increasing with engine speed increase
LC_IGC_LIH_CONF	Configuration Choice for Half static Mode in case of CAM limp home (0 = no Limp Home - 1 = Half Static) If CAM LH is enabled on the engine (*cf. ENSD settings: no CAM signal -> engine starting and running on CRK only*) Then the ignition can / have to be turned to 1/2 static mode (=1) to support the case of a false phased engine from start and insure thus at least a spark on the real burning cylinder *Here, this setting can be change to =0 if the engine can only run without CAM and not start without CAM* Else (no CAM LH mode enabled on the project), no LH mode on coil driving is needed (=0) = 1 generic recommendation

38.25 Calibration hints for function IGRE – IGRE Configuration Data

This file (100E) is a Customer version. It has thus to be adapted to the concerned engine configuration (*HW module for coil control and diagnostic + combustion mode + CAM LH settings*) and to project specific needs or recommendations from the customer.

Here are listed the configuration data, which are used only in the IGRE aggregate. Given there are the typical values widely applied and recommended from previous experiences and tests:

NC_IGN_DIAG_TYP	Engine HW dependent: IGRE configuration depending on HW module for coil control and diagnostic used. Please refer to the coil driver mounted on engine 01 = ATM46 02 = ATIC29
NC_TD_LIM_MIN	Minimal value of Dwell Time at least to insure a spark, avoid Misfire = 1 ms
NC_TD_LIM_MAX	Maximal value of Dwell Time. No risk of coil over-stresses. Refer to coil supplier recommendation and information in 100E = 16ms (max)
NC_MAX_IGN_MPL_NR	Maximum number of multiple spark = 7
NC_IGC_DLY	Number of recurrence after engine start, which had to pass by to activate the diagnosis =16
NC_MPL_T_MAX	Absolute maximal duration of multiple spark =60ms
NC_MPL_IGN_CRK_MAX	Maximum ignition angle after TDC where Multiple spark has to be stopped =12 to 18°CRK
NC_IGC_CONF	Engine HW dependant: Ignition mode half or full static. Refer to the coils driver used on the engine and/or engine ECU. 0 = 1/2 Static Coils 1 = Full Static Coils

38.26 Calibration hints for function IGRE

Ignition output diagnosis (for ATM 46)

Settings and Calibration for the Ignition Driver output diagnosis (200Y01) in ATM46 usage

Coil Driver protection

NC_INI_CTR_DEAC	Time delay after IGBT threshold response before switching OFF the IGBT In case of a short circuit to battery, the power dissipation could destroy the power stage = ATM46. From the Dwell Time ON event in Short Circuit to Battery, the IGBT's can withstand this overload current only for a very short time (750µs max. for IGBT without shunt). So, it is necessary to switch off this IGBT transistor for safety and protection reasons after a time delay has elapsed (NC_INI_CTR_DEAC). This time delay is necessary in order to filter parasitic spikes. = 600µs - 750µs max. From ATM46 capability, to not burn the driver
NC_IGBT_CUT_OFF_T	Inhibition of "IGBT protection cut off function" To avoid false diagnostic and cut off, the survey of the power stage has to be limited to ~ the Dwell Time setting ~3 to 5ms

Burn Time measurement

ID_V_DUR_IGC_MIN	Minimum duration of the Burn time to detect sparking failure To be measured on the 2ary wire of the coils on an engine Depend on each combustion chambers burning capability and behaviour (type, geometry, valves, tumble, swirl, etc…) *If measured Burn time is smaller than the map then combustion failure is set. The information is mainly used by Misfire function for cross diagnostic* ~0,5 to 0,1ms decreasing with engine speed increase
LDP_N_32_ID_V_DUR_IGC_MIN LDP_MAF_ID_V_DUR_IGC_MIN	For the LDP settings, refer to the one used for Dwell Time and / or the one from IGSP (volumetric efficiency influence)

Ignition diagnosis - A016

The values of the ABC have to be adapted according to the engine HW (*coil driver device and coils*) capability to withstand a failure.

The Short to Battery failure - most dangerous and damaging one for the driver - has to be set faster than the others

Open Circuit and Short to Ground are only interesting has feedback information to Misfire function for cross diagnosis to alert that spark is no more created (they are generally out of concern regarding coil or coil driver protections). The calibration of the related ABC have thus to be linked also to the Misfire target.

Settings and Calibration for the Ignition diagnosis (A016)

C_ABC_IGC_SCP	Value of anti-bounce counter after regeneration cycle (*cf. after*) = 128 (1/2 of the C_ABC_MAX_IGC_SCP)
C_IGA_OFF_CYCNR_HLD	Number of engine Cycle for the regeneration tentative of the coil = number of engine cycle where the coil is switched off after a confirmed SCP failure, before attempting to set it back to work = 16 (coil cooling time)
C_SUM_IGC_SCP	Number of regeneration cycles (*cf. before*) applied before switching off definitively the coil until next ECU reset = 255 set to max.
C_ABC_INC_IGC_SCP	Anti – bounce counter increment for the ignition system diagnosis SCP = 5 (quicker than OL and SCG)
C_ABC_INC_IGC_OL	Anti – bounce counter increment for the ignition system diagnosis OL = 4
C_ABC_INC_IGC_SCG	Anti – bounce counter increment for the ignition system diagnosis SCG = 4
C_ABC_MAX_IGC_SCP	Maximum value of the ABC for the ignition system diagnosis SCP. Will switch off the coil and launch the regeneration cycle (*cf. before*) = 255
C_ABC_MAX_IGC_OL	Maximum value of the ABC for the ignition system diagnosis OL = 255
C_ABC_MAX_IGC_SCG	Maximum value of the ABC for the ignition system diagnosis SCG = 255
C_ESB_IGN_CUT_DIAG	C_ESB_IGN_CUT_DIAG = 1 will inhibit the Cut Off protection function = 0 (function enabled is recommended)
C_TCO_MIN_IGC_DIAG	Min Temperature for condition diagnosis OL & SCG = 20°C (emission cycle starting)
C_N_32_MIN_SCP_IGC_DIAG	Min engine speed for condition diagnosis SCP = 512rpm = exist of Start phase
C_N_32_MIN_OL_SCG_IGC_DIAG	Min engine speed for condition diagnosis OL/SCG = 512rpm = exist of Start phase

38.27 Cus adap module: IGRE

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Fu_time_zyl[NC_CYL_NR]	O/V	0...FFFFH	0...255.99609	3.9063e-3	[ms]
zylinderindividuelle Funkenbrenndauer					
Stat_fofumx_ist	O/V	0...FFFFH	0...65535	1	[-]
Umsetzung maximale Folgefunken					
STATE_MPL_CYL_IGN	O/V	0...FFFFH	0...65535	1	[-]
request of maximal number of follow ignition pulses (bit coded cylinder individuel)					

Input data:

ECU_STATE	LV_ST_END	NC_CYL_NR	Stat_fofumx_anf
V_DUR_IGC[NC_CYL_NR]			

38.27.1 Outputs for BMW functions

FUNCTION DESCRIPTION:

Adaptations to BMW environment.

Application conditions:

Initialisation at reset or at exit PWL:	0
Recurrence:	segment
Activation:	at every engine state
Deactivation:	---

Formula section:

Stat_fofumx_ist = 0
Fu_time_zyl[NC_CYL_NR] = V_DUR_IGC[NC_CYL_NR]

38.27.2 Outputs for SV aggregates

FUNCTION DESCRIPTION:

Adaptations to BMW environment.

Application conditions:

Initialisation: 0 at reset

Recurrence:
if LV_ST_END **=** 0 or ECU_STATE = "PWL"
then recurrence is 10ms
else recurrence is segment synchron

Activation: at every engine state

Deactivation: ----

Formula section:

STATE_MPL_CYL_IGN = Stat_fofumx_anf

39 Ignition angle setpoint

Table of Contents

Index

39.1 Ignition Angle General

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA[NC_CYL_NR]	O/V	0...FFH	-35.625...60	0.375	°CRK
Basic ignition angle application correction and knock control included					
IGA_AV_MV_H_RNG	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Mean value of actual ignition angle in wide range					
IGA_AV_MV_CAN_H_RNG	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Mean value of actual ignition angle in wide range					
IGA_AV_MV_CBK_H_RNG[NC_CBK_EX_NR]	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Mean value of actual ignition angle bank selective in wide range					
IGA_AV_H_RNG[NC_CYL_NR]	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Homogeneus ignition angle in wide range applied on cylinder x					
IGA_MIN_CBK_H_RNG[NC_CBK_EX_NR]	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Minimum ignition angle in wide range bank selective					
IGA_WOUT_KNK[NC_CYL_NR]	O/V	0...FFH	-35.625...60	0.375	°CRK
Actual ignition angle (incl. application corrections without knock interception)					
LV_IGA_GRD_ACT	O/V	0...1H	0...1	1	-
Flag for ignition angle gradient limitation out of start active					
IGA_AV_H_RNG_S[NC_CYL_NR]	O/V	FA60...5A0H	-90...90	0.0625	°CRK
homogeneus ignition angle in high range applied on cylinder x					
IGA_SP_MAX_1_CBK_H_RNG[NC_CBK_EX_NR]	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Maximum ignition angle depending on torque intervention in wide range bank selective					
IGA_SP_MAX_1_CAN_H_RNG	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Maximum ignition angle depending on torque intervention in wide range					
IGA_AV_H_RNG_1[NC_CYL_NR]	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Intermediate ignition angle in wide range till gradient limitation after start is terminated					
IGA_MIN_H_RNG	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Minimum ignition angle in wide range					
IGA_AV_H_RNG_HOMS[NC_CYL_NR]	O/V	FA60...5A0H	-90...90	0.0625	°CRK
Homogeneus ignition angle in wide range applied on cylinder x					

Input data:

IGA_ST	IGA_DIF_MIN_H_RNG	IGA_ADD_CMB_CTL[NC_CYL_NR]	IGA_ADJ_KNK[NC_CYL_NR]
IGA_IS_TQ_KNK	IGA_SP_CBK_H_RNG[NC_CBK_EX_NR]	IGA_TRA_KNK	LV_ES
LV_N_MAX_ETC_LIH	LV_ST	N	TCO
IGA_REF_COR_CBK[NC_CBK_EX_NR]	IGA_BAS_COR_CBK[NC_CBK_EX_NR]	IGA_AV_H_RNG_S_1[NC_CYL_NR]	IGA_AV_H_RNG_HOMS_1[NC_CYL_NR]
IGA_SP_CBK_CAN_H_RNG[NC_CBK_EX_NR]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_IGA_AS	1	80...7FH	-48...47.625	0.375	°CRK
Spark retard by application system global					
C_IGA_AS_CYL[NC_CYL_NR]	1	80...7FH	-48...47.625	0.375	°CRK
Spark retard by application system cylinderindividual					
C_IGA_INI_H_RNG	1	FA60...5A0H	-90...90	0.0625	°CRK
Init value for ignition angle in wide range					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_IGA_MAN_H_RNG	1	FA60...5A0H	-90...90	0.0625	°CRK
Absolute spark retard input by application system in wide range					
C_IGA_MIN_ETC_LIH	1	0...FFH	-35.625...60	0.375	°CRK
Absolute possible spark retard at ETC LIH					
C_N_MAX_IGA_10MS_CLC	1	0...1FE0H	0...8.16E+3	1	rpm
Max. engine speed threshold above which the 10ms tasks are deactivated					
LC_IGA_MAN	1	0...1H	0...1	1	-
Switch for absolute spark retard input by application system					
LC_IGA_ST_ENA	1	0...1H	0...1	1	-
Logical constant for separated start ignition angle					
IP_IGA_LGRD_AST	8	0...FFH	0...95.625	0.375	°CRK
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	°C
Gradient limitation IGA after start phase					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CBK_EX_NR	1	1...4H	1...4	1	-
Number of exhaust cylinder banks					
NC_CYL_NR	1	1...8H	1...8	1	-
count of cylinder					

39.1.1 IGSP_M6001

General information:

This module coordinates different ignition angle requests and calculates the ignition angles IGA_AV_H_RNG[x] and IGA_AV_H_RNG_S[x].

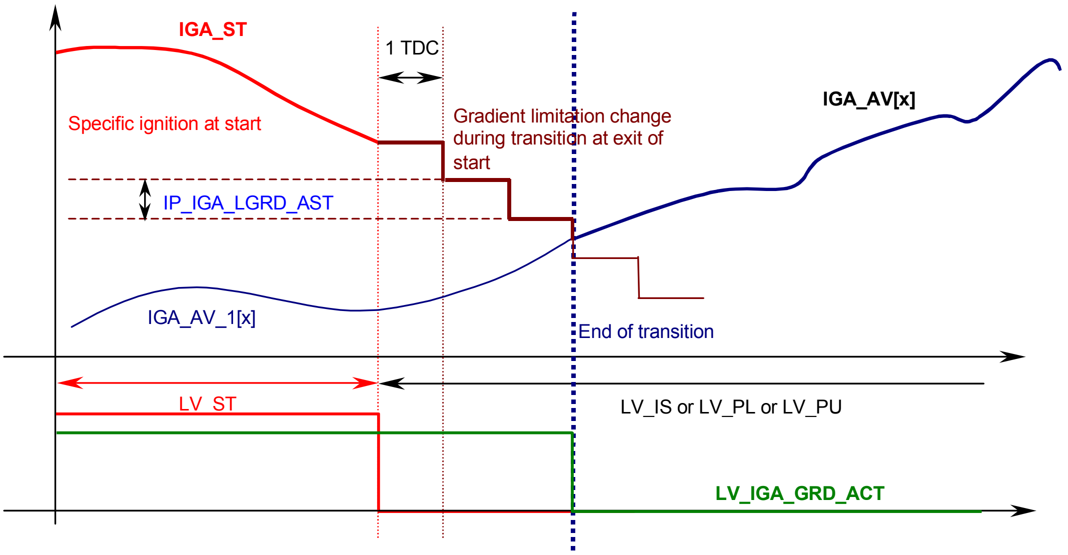
IGA_SP is the output from the torque management functions to realize the torque setpoint. IGA_SP is limited to IGA_MIN the minimum ignition angle which can be applied to the engine.

IGA_BAS_COR is the corrected basic igniton angle which is realized normally if no other requests like knock control or torque requests (via IGA_SP) requires a retarded ignition angle.

The reference ignition angle IGA_REF_COR is a theoretical ignition angle at maximum engine torque. It is the reference value to all ignition angles to calculate the ignition angle differences (IGA_DIF_xxx) for the calculation of EFF_IGA_xxx values.

According to application choice (through logical application incidence flag LC_IGA_ST_ENA) the ignition angle during start phase is not managed through the torque but by the specific value IGA_ST. In this case, the corresponding ignition efficiencies are set to 1. At the end of start phase the transition from IGA_ST to the set point ignition delivered by torque calculation IGA_AV_1[x] is approached through a gradient limitation (the ignition gradient step is calibrateable).

This module delivers the homogeneus ignition angle IGA_AV_H_RNG[x] and the stratified ignition angle IGA_AV_H_RNG_S[x] for the MSD70 project in IGRE. For the MSV70, the stratified ignition angle is a copy of the homogeneus one. Therefore, this HOM / S switch is also visible in MSV70, but effectiv.



Picture: example for transition after start with IGA gradient limitation

Application conditions:

Initialisation:	IGA_AV_H_RNG[x] = C_IGA_INI_H_RNG at reset
	IGA_MIN = -90 °crk at reset
	LV_IGA_GRD_ACT = 0 at engine stalling
Recurrence:	every 10 ms and every TDC
Activation:	at every engine operating state
Deactivation:	-

Application Condition

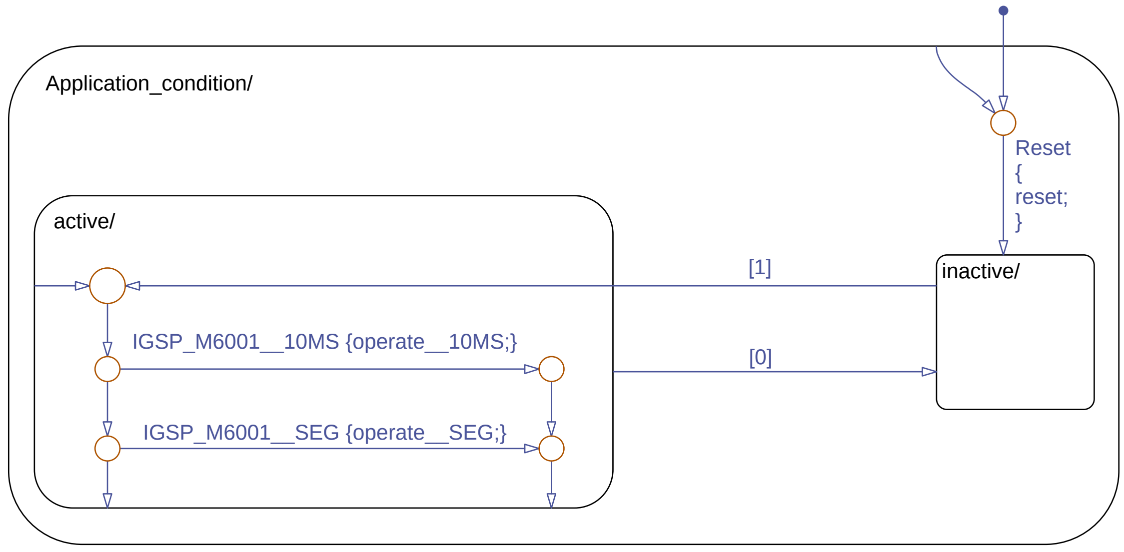


Figure 1 IGSP_M6001/ APP_CDN/ APPCND

Function Description

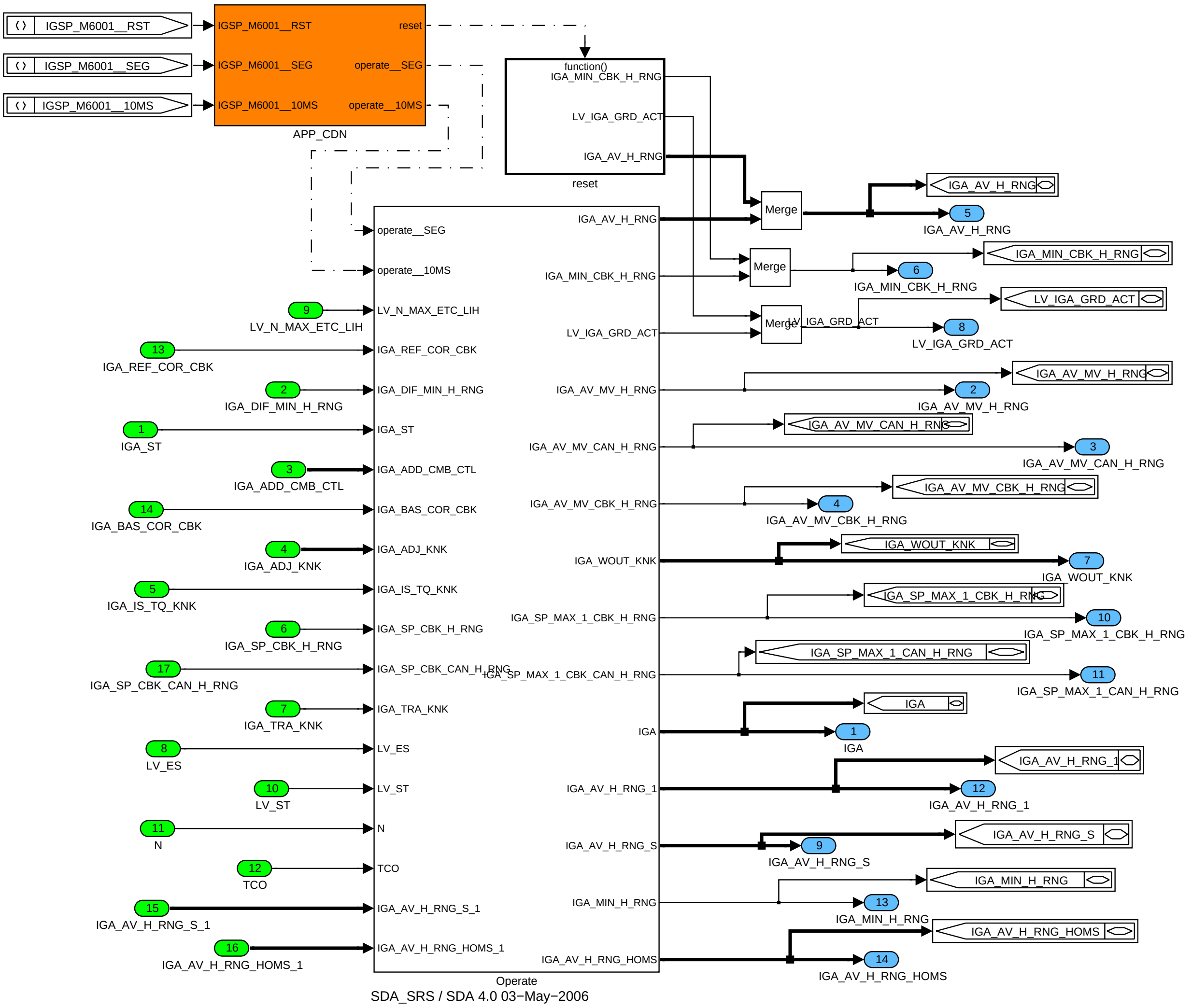


Figure 2 IGSP_M6001

39.1.1.1 SUBFUNCTION: Operate

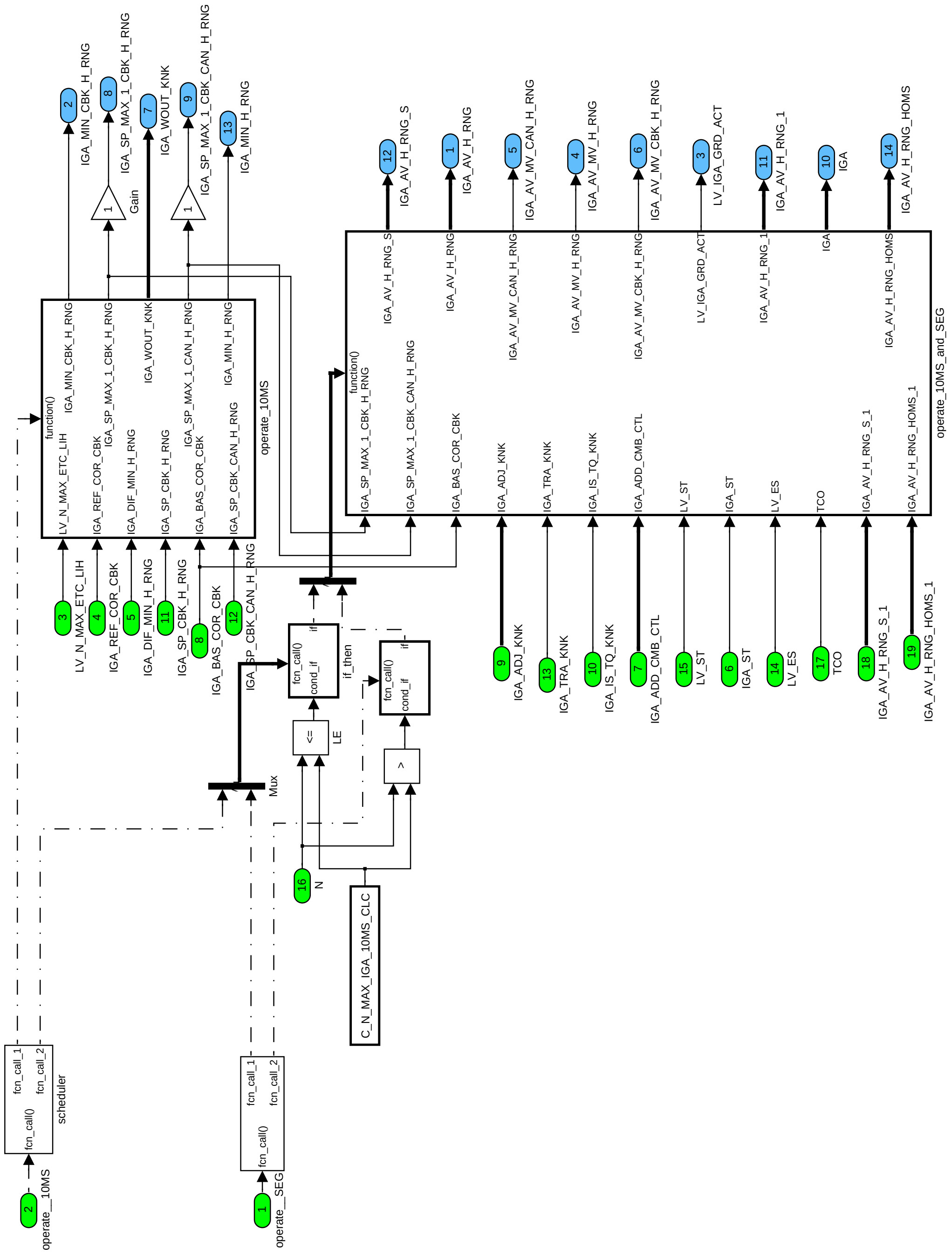


Figure 3 IGSP_M6001/ Operate

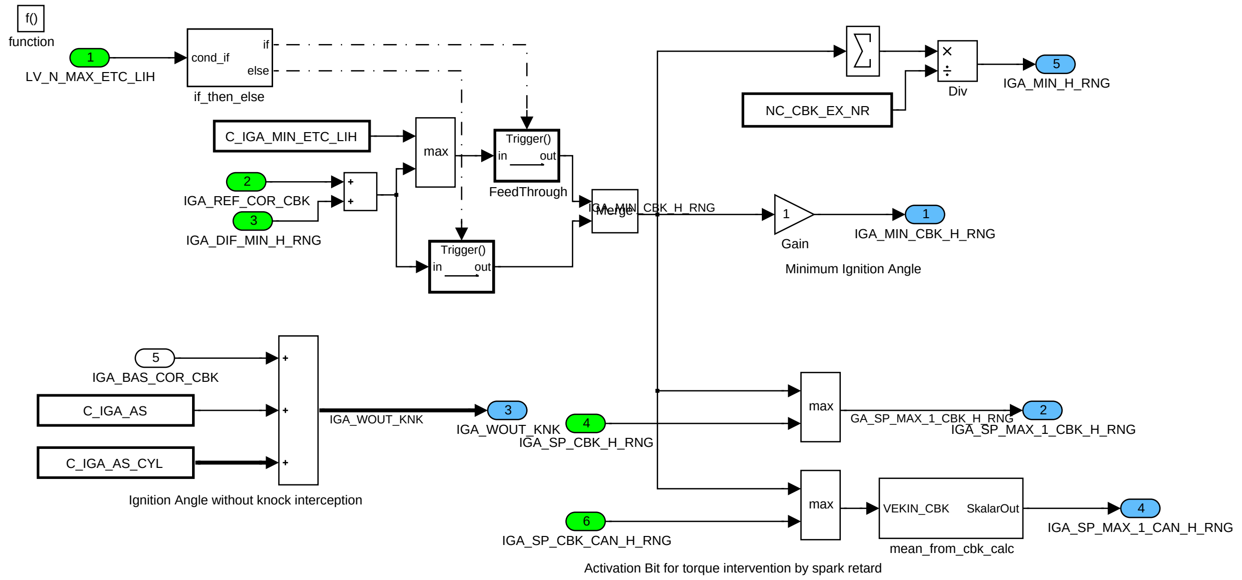


Figure 4 IGSP_M6001/ Operate/ operate_10MS

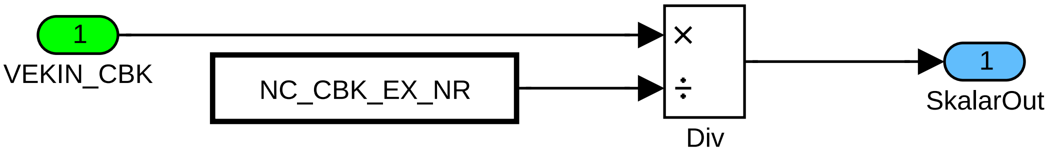


Figure 5 IGSP_M6001/ Operate/ operate_10MS/ mean_from_cbk_calc

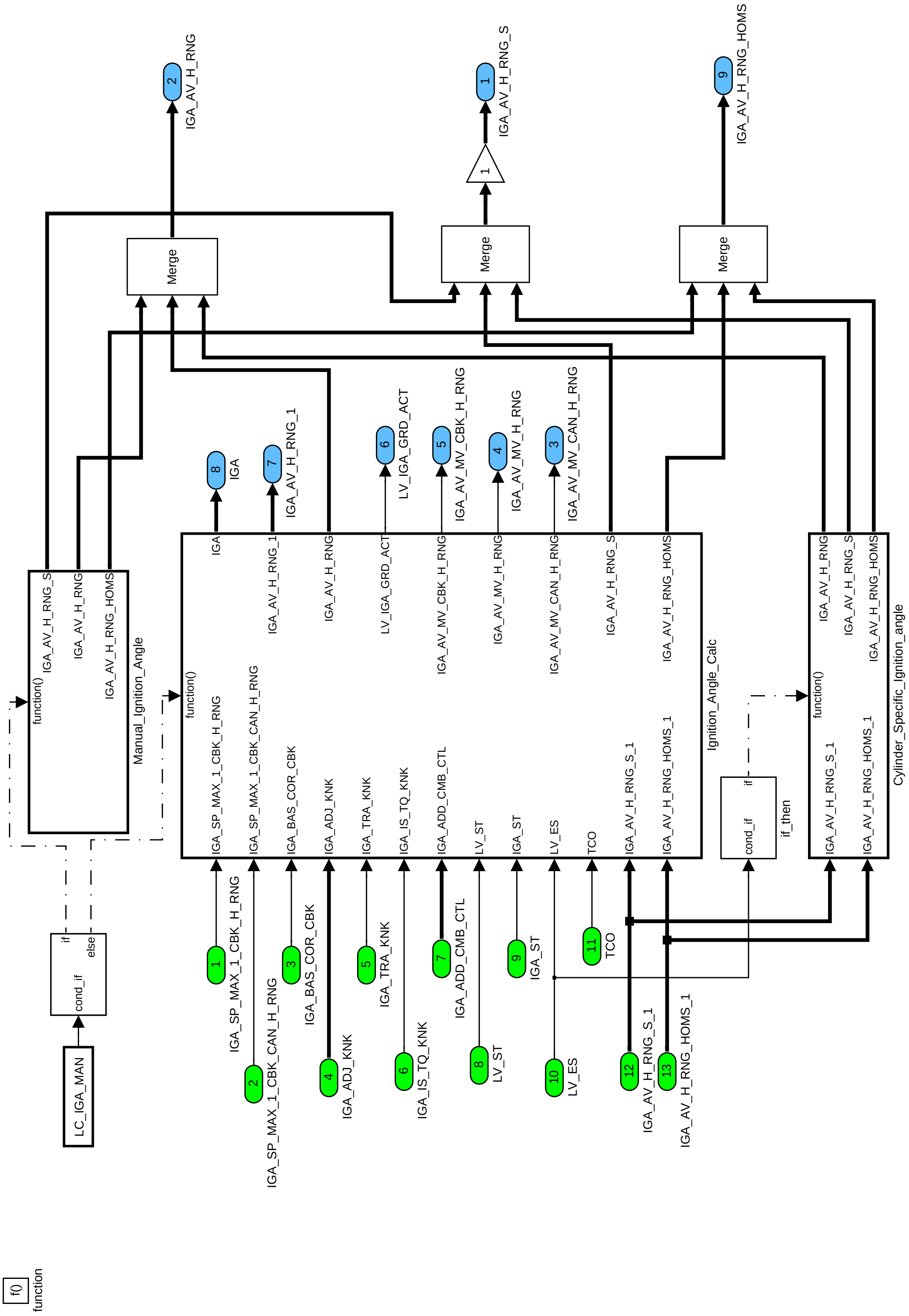


Figure 6 IGSP_M6001/ Operate/ operate_10MS_and_SEG

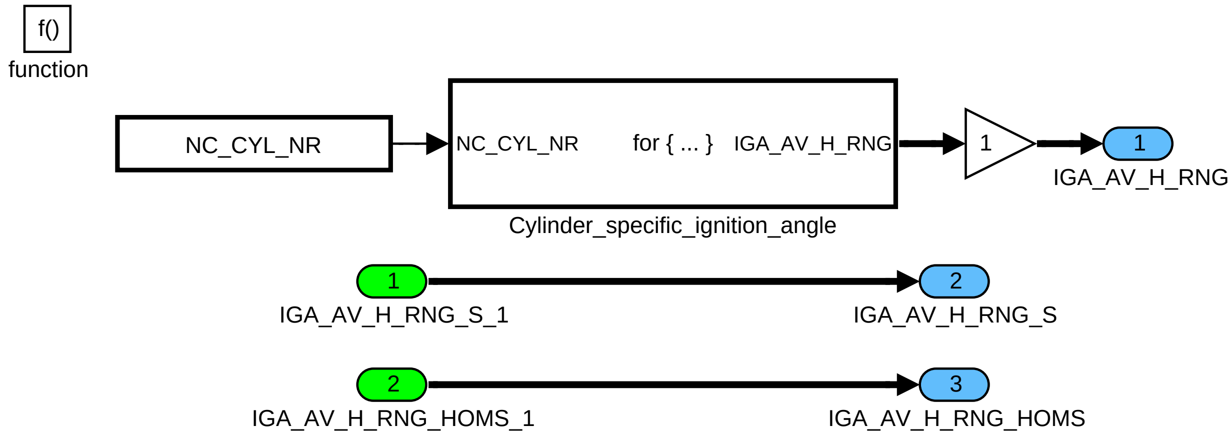


Figure 7 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Cylinder_Specific_Ignition_angle

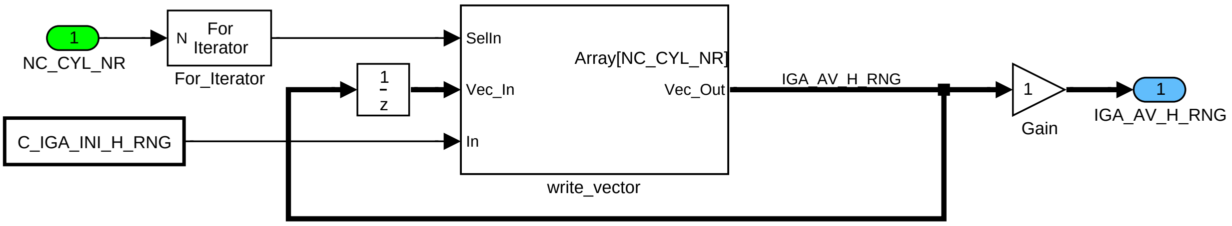


Figure 8 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Cylinder_Specific_Ignition_angle/ Cylinder_specific_ignition_angle

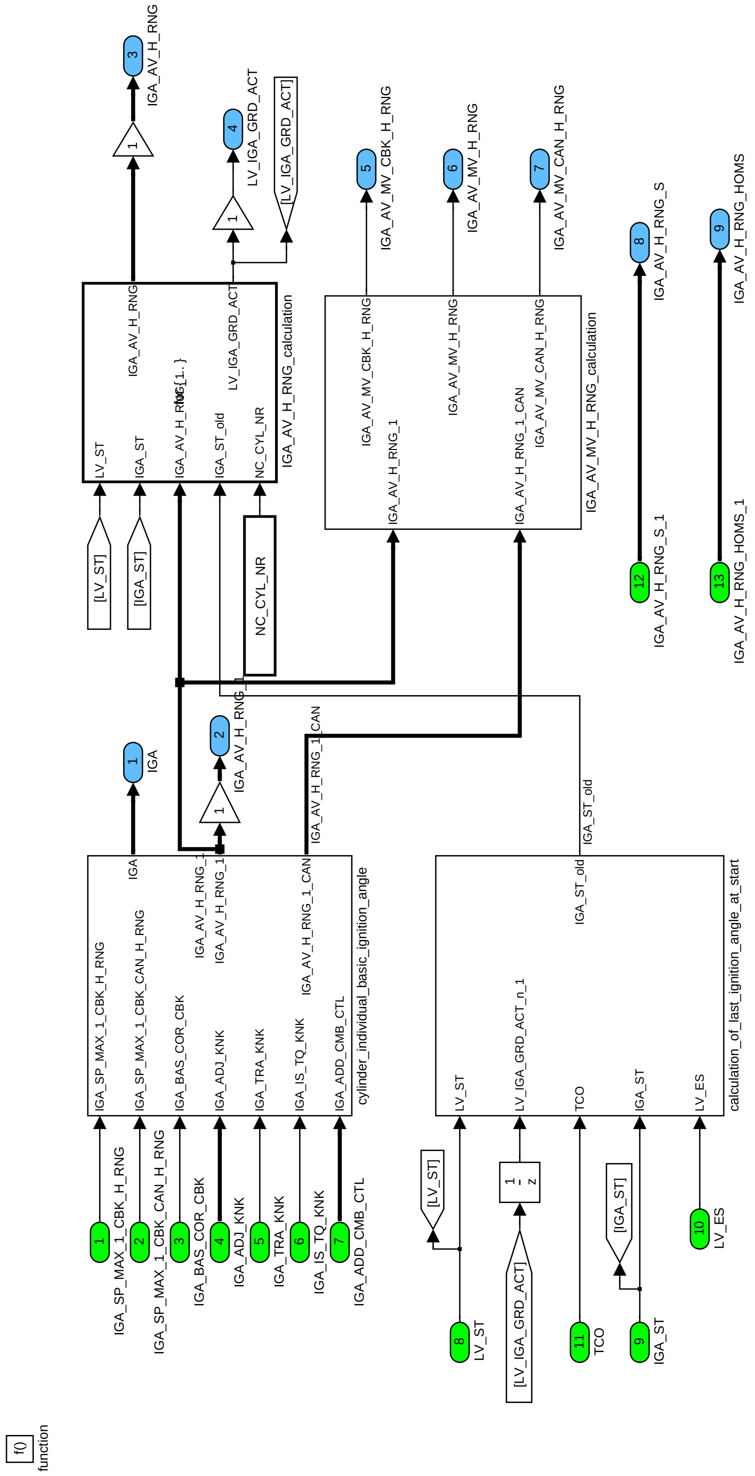


Figure 9 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc

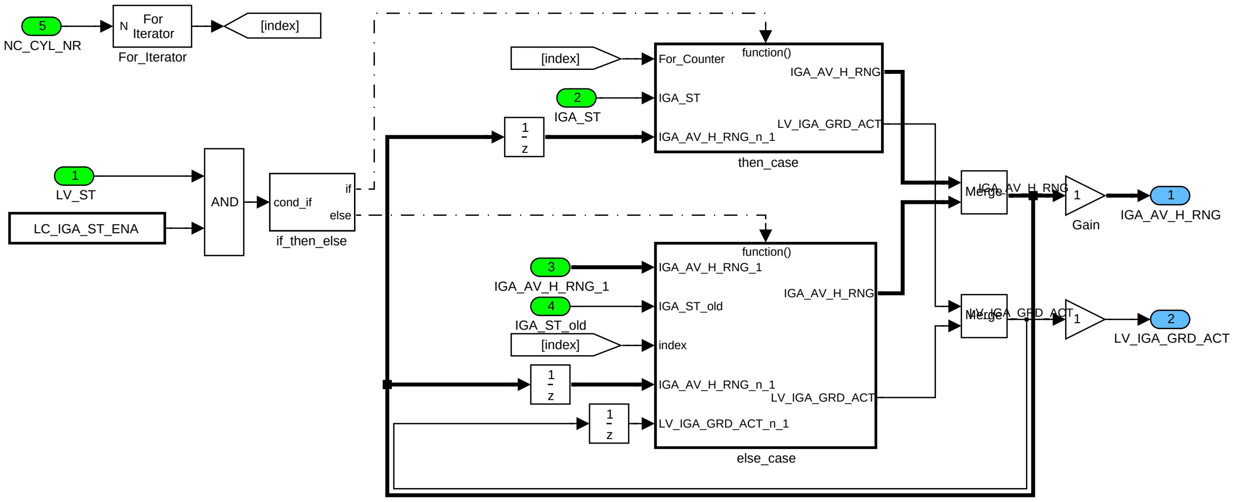


Figure 10 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ IGA_AV_H_RNG_calculation

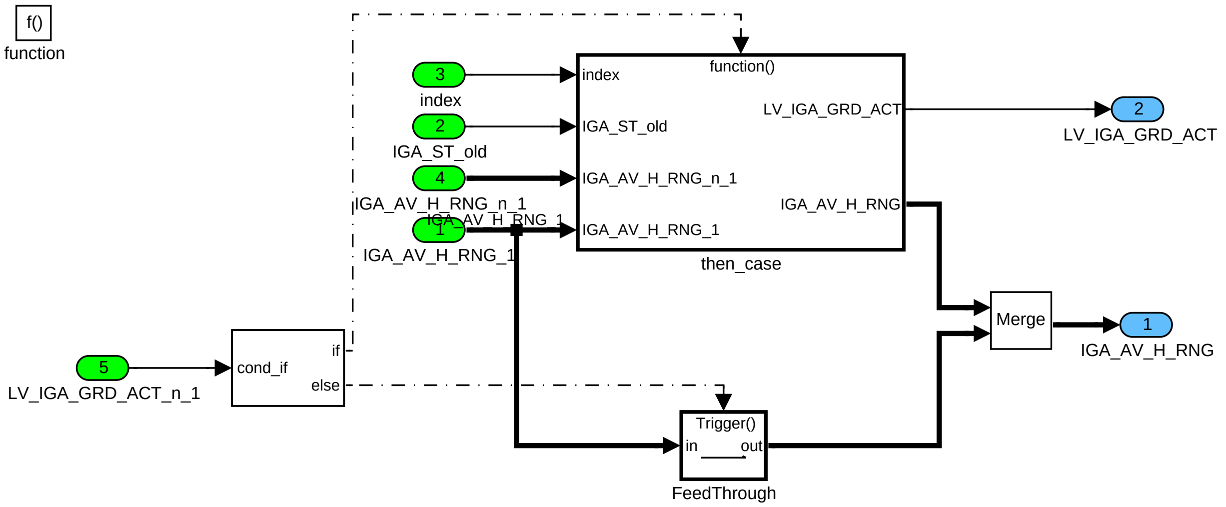


Figure 11 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ IGA_AV_H_RNG_calculation/ else_case

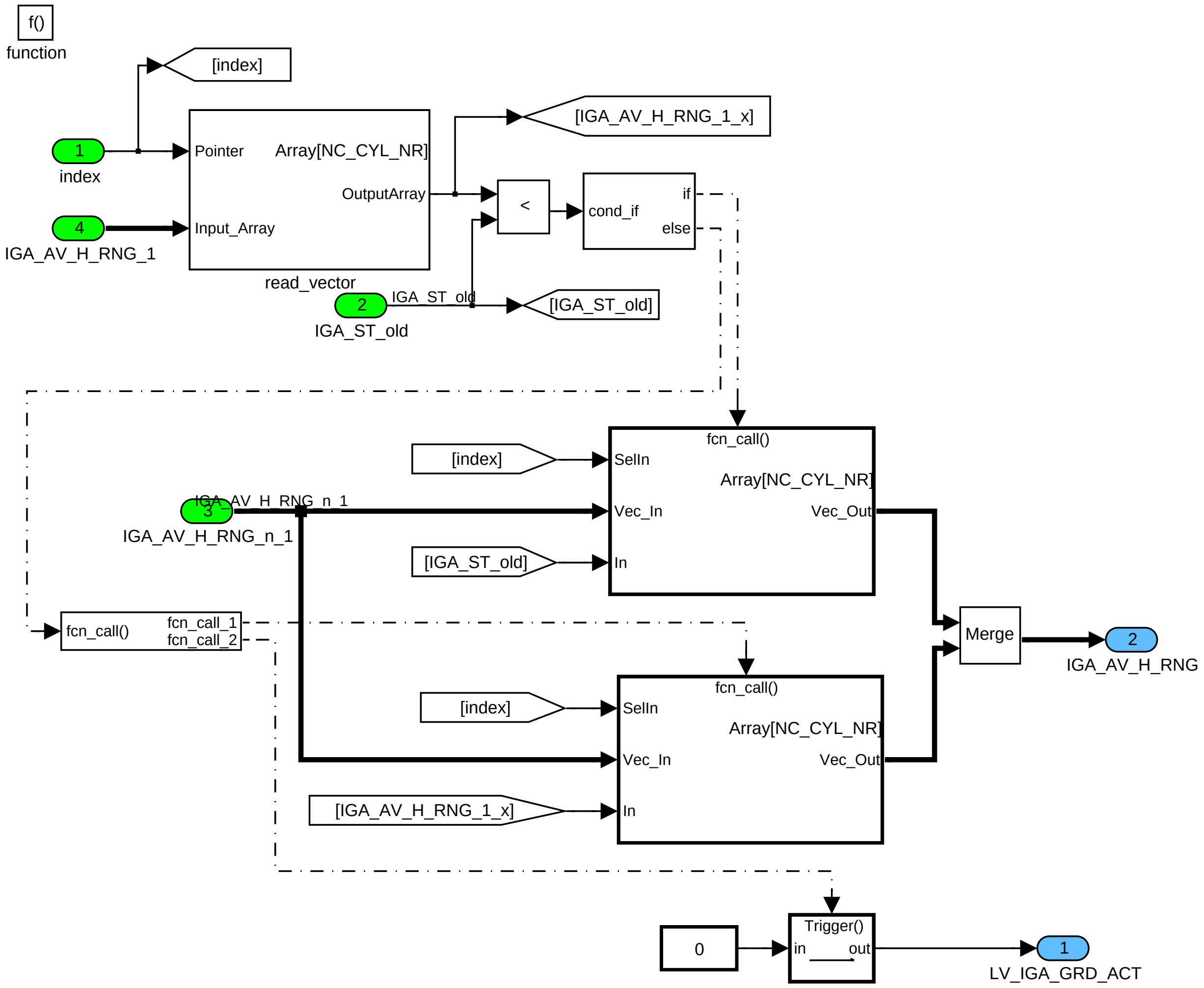


Figure 12 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ IGA_AV_H_RNG_calculation/ else_case/ then_case

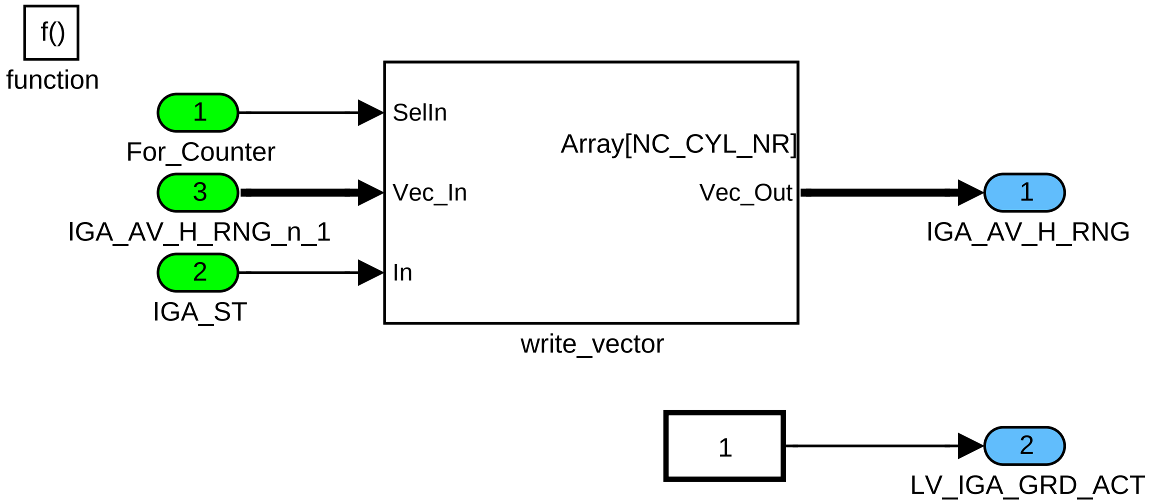


Figure 13 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ IGA_AV_H_RNG_calculation/ then_case

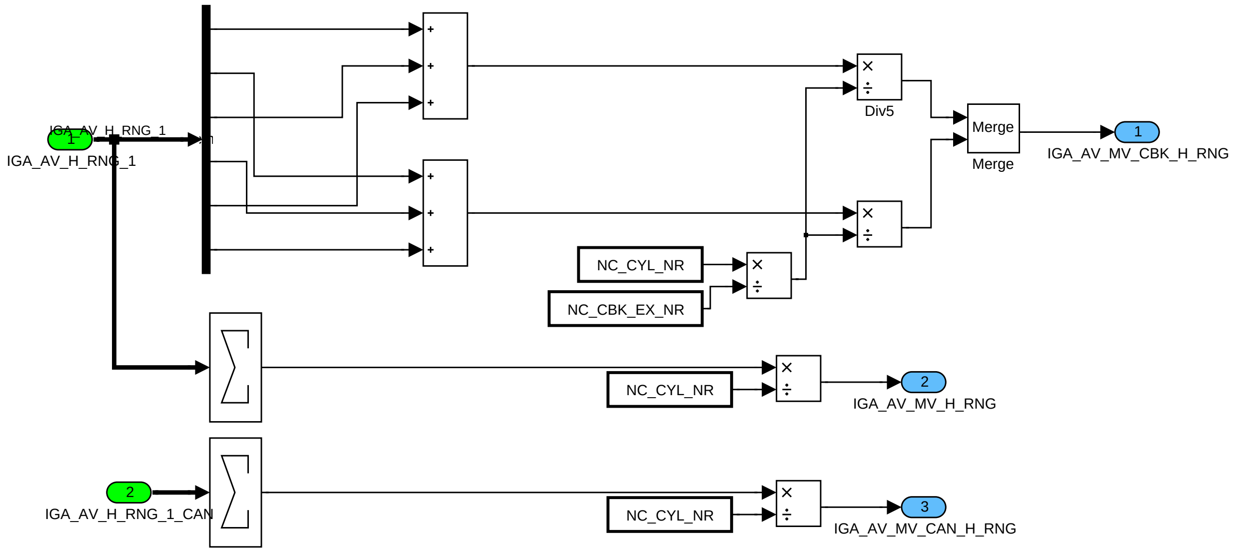


Figure 14 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ IGA_AV_MV_H_RNG_calculation

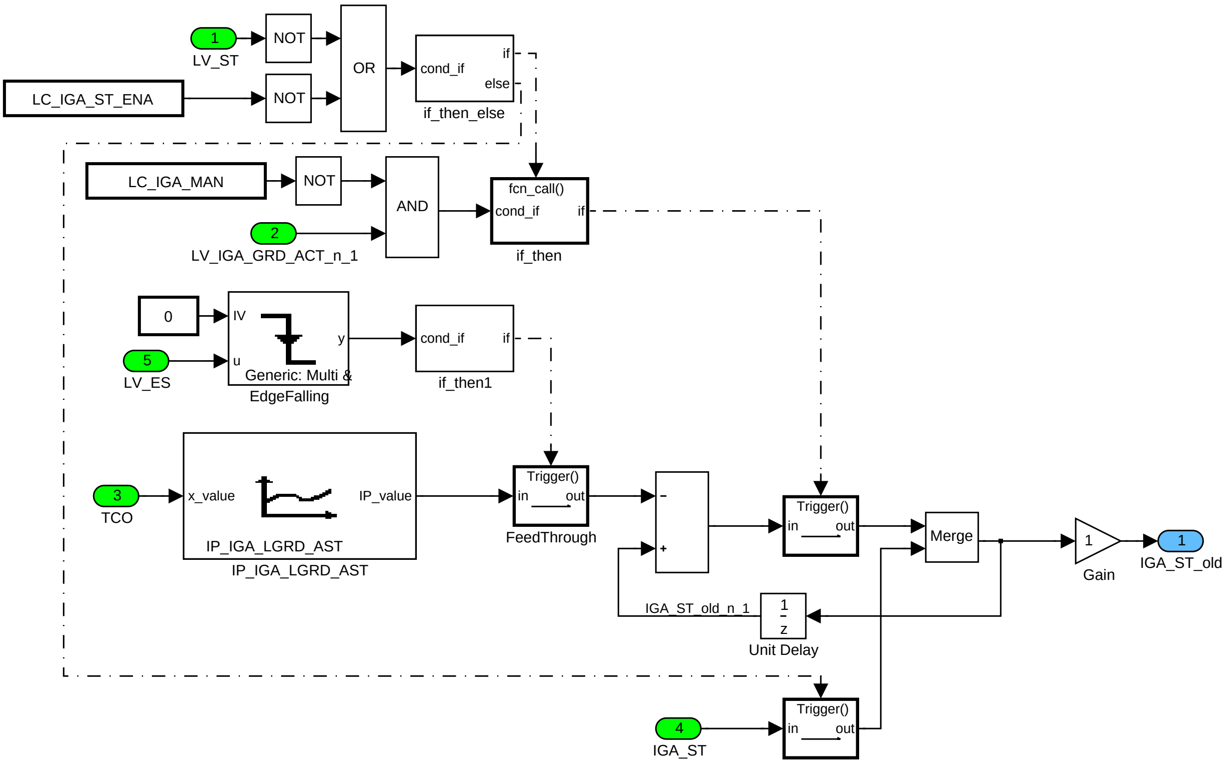


Figure 15 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ calculation_of_last_ignition_angle_at_start

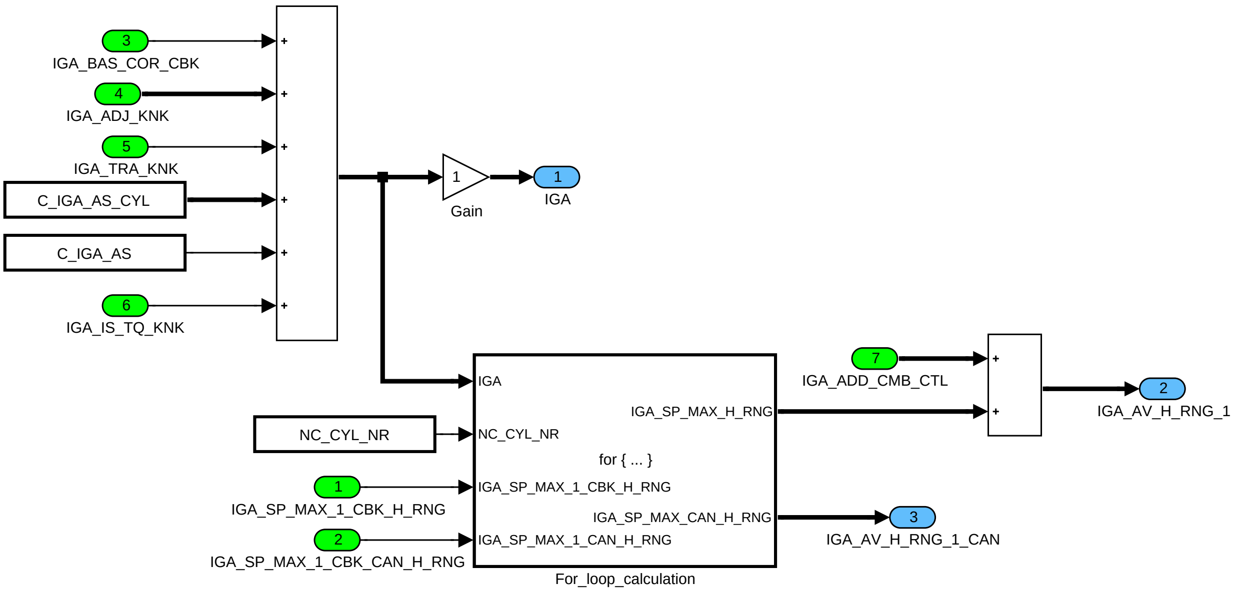


Figure 16 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ cylinder_individual_basic_ignition_angle

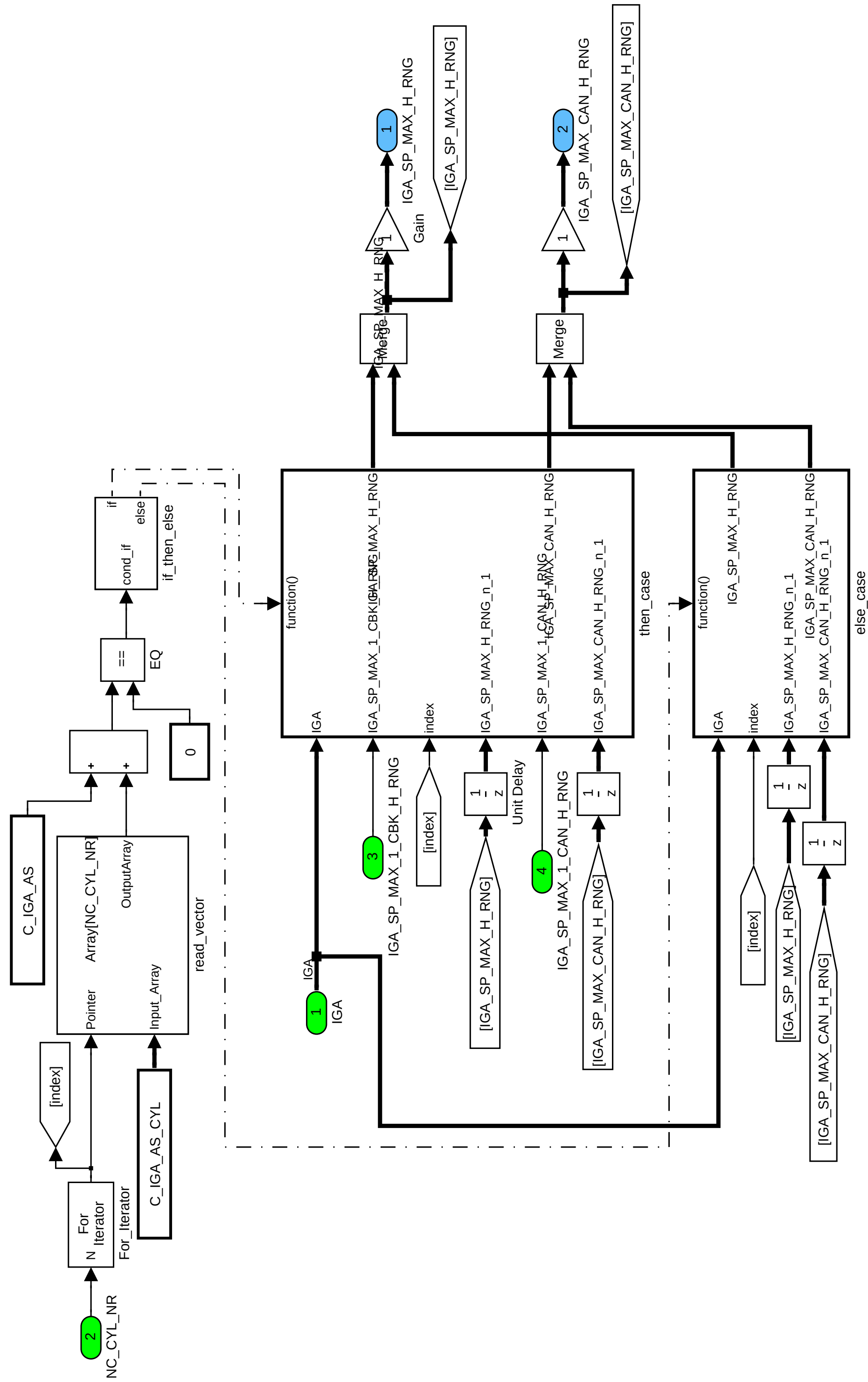


Figure 17 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ cylinder_individual_basic_ignition_angle/ For_loop_calculation

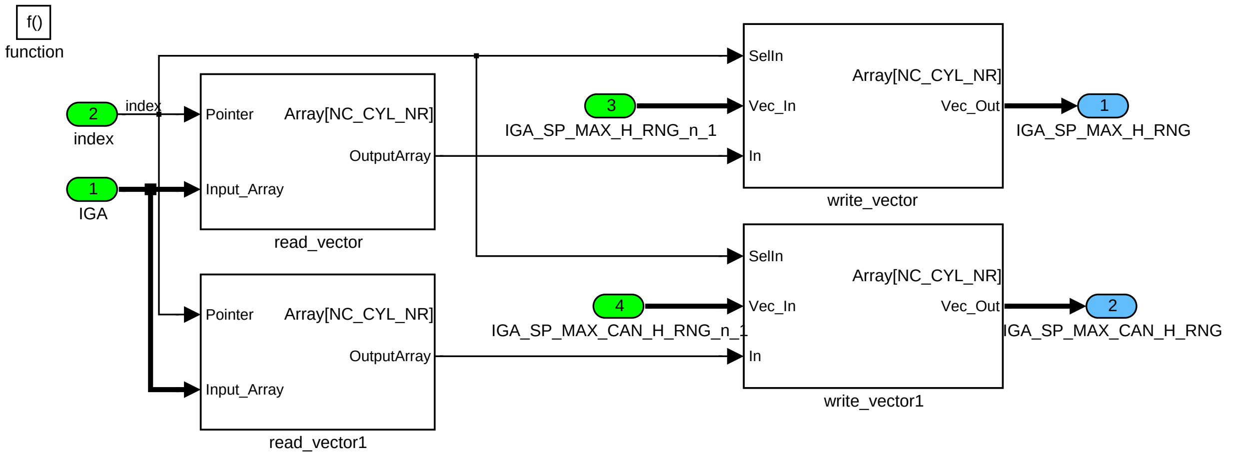


Figure 18 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ cylinder_individual_basic_ignition_angle/ For_loop_calculation/ else_case

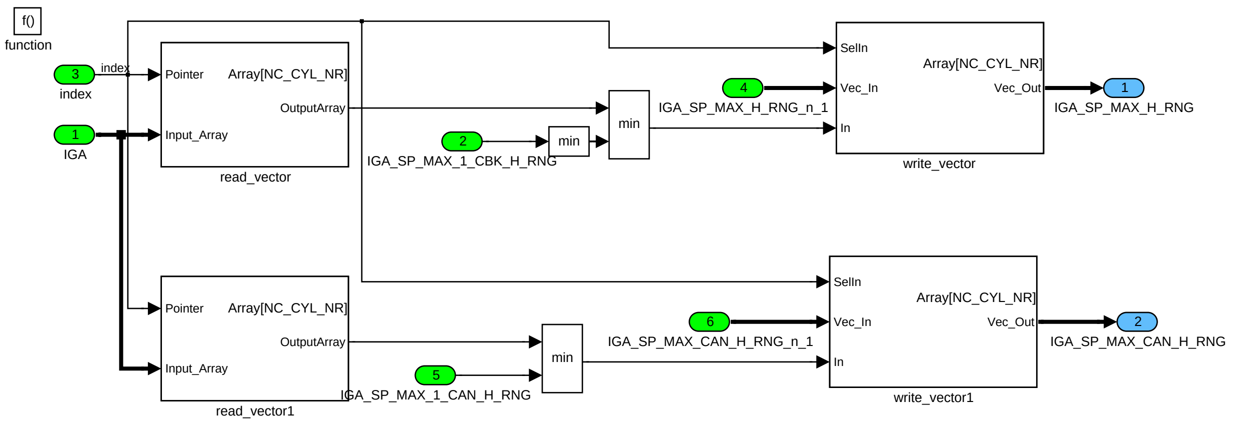


Figure 19 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Ignition_Angle_Calc/ cylinder_individual_basic_ignition_angle/ For_loop_calculation/ then_case

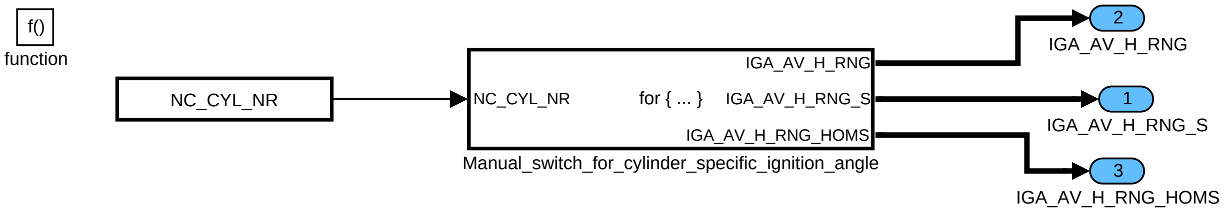


Figure 20 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Manual_Ignition_Angle

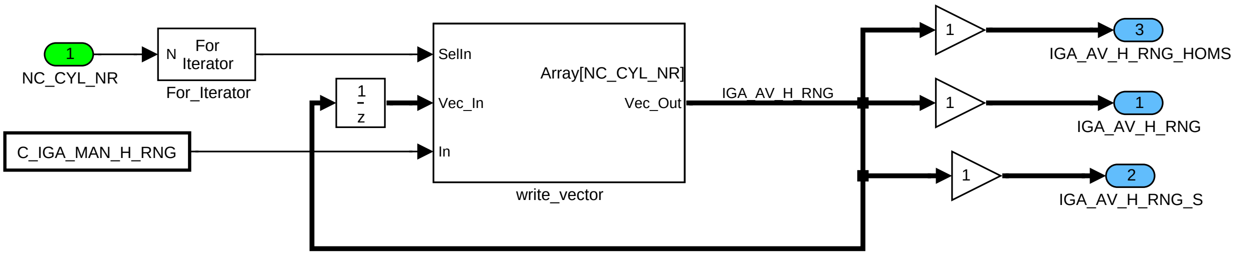


Figure 21 IGSP_M6001/ Operate/ operate_10MS_and_SEG/ Manual_Ignition_Angle/ Manual_switch_for_cylinder_specific_ignition_angle

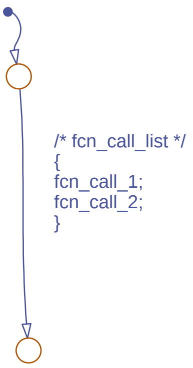


Figure 22 IGSP_M6001/ Operate/ scheduler2/ module_scheduler

39.1.1.2 SUBFUNCTION: reset

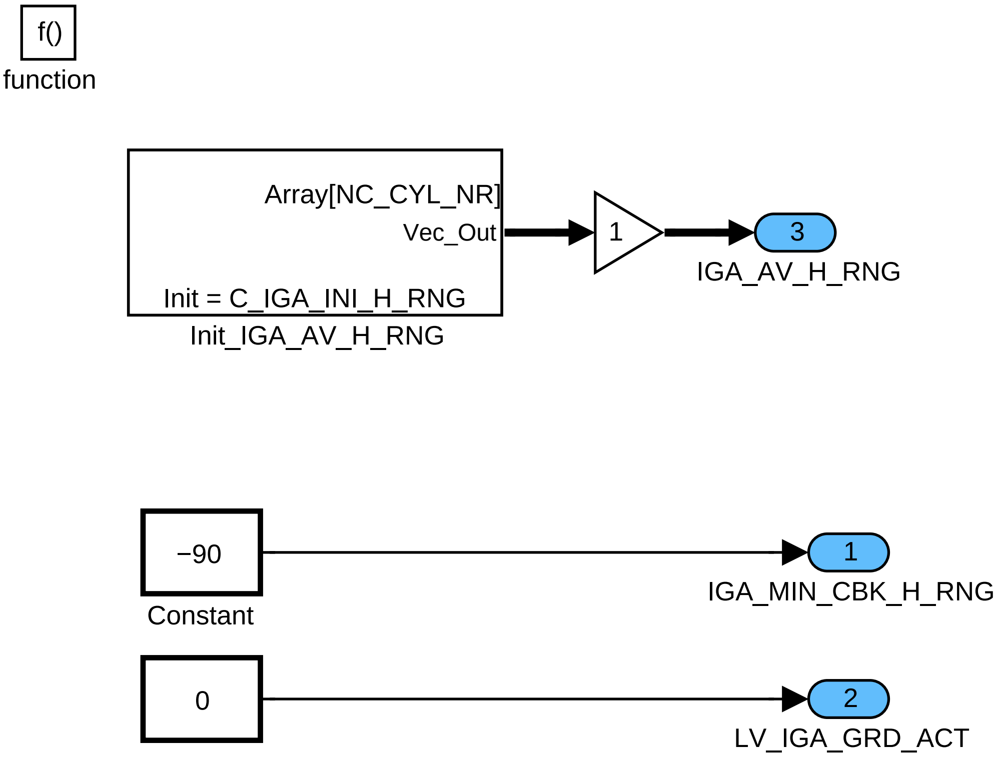


Figure 23 IGSP_M6001/ reset

39.2 Basic ignition angle at start (IGA_ST)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_ST	O/V	0...FFH	-35.625...60	0.375	°CRK
Basic ignition angle at start					
LV_IGA_ST_PSN_ENG_COR_STST	V	0...1H	0...1	1	-
IGA_ST correction for MSA start is active					
LV_ANG_PSN_ENG_REL_TMP	V	0...1H	0...1	1	-
The first ITDC is reached in case of MSA start					

Input data:

LV_IGA_ST_OPM_SEL	AMP	CYC_ST	LV_ST
N	TCO	TIA	LV_ES
LV_SYN_ENG	LV_STST_ST_REQ	PSN_ENG_REL	PSN_ENG_REL_ST

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ANG_PSN_ENG_REL_STST	1	0...780H	0...720	0.375	°CRK
Relative engine position threshold for deactivation of additive ignition angle correction (default 120 °CRK) [Attention: PSN_ENG_REL = 180...0,375°CRK]					
IP_IGA_ST_AMP_COR_OPM_1	6	0...FFH	-48...47.625	0.375	°CRK
LDPM_AMP_1_IGA_ST	6	0...FFFFH	0...5.434E+3	0.08291752	hPa
Ambient pressure correction of the basic ignition angle at start at operation mode 1					
IP_IGA_ST_AMP_COR_OPM_2	6	0...FFH	-48...47.625	0.375	°CRK
LDPM_AMP_1_IGA_ST	6	0...FFFFH	0...5.434E+3	0.08291752	hPa
Ambient pressure correction of the basic ignition angle at start at operation mode 2					
IP_IGA_ST_PSN_ENG_COR_STST	6	0...FFH	-48...47.625	0.375	°CRK
LDP_ANG_PSN_ENG_REL_ST	6	0...780H	0...720	0.375	°CRK
Relative engine stop position correction additive to IGA_ST for the first ignition at MSA start					
IP_IGA_ST_TIA_COR_OPM_1	6	0...FFH	-48...47.625	0.375	°CRK
LDPM_TIA_1_IGA_ST	6	0...FEH	-48...142.5	0.75	°C
Ambient pressure correction of the basic ignition angle at start at operation mode 1					
IP_IGA_ST_TIA_COR_OPM_2	6	0...FFH	-48...47.625	0.375	°CRK
LDPM_TIA_1_IGA_ST	6	0...FEH	-48...142.5	0.75	°C
Ambient pressure correction of the basic ignition angle at start at operation mode 2					
IP_IGA_ST_BAS_OPM_1	6x8	0...FFH	-35.625...60	0.375	°CRK
LDPM_N_1_IGA_ST	6	0...1FE0H	0...8.16E+3	1	rpm
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	°C
Basic ignition angle at start at operation mode 1					
IP_IGA_ST_BAS_OPM_2	6x8	0...FFH	-35.625...60	0.375	°CRK
LDPM_N_1_IGA_ST	6	0...1FE0H	0...8.16E+3	1	rpm
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	°C
Basic ignition angle at start at operation mode 2					
IP_IGA_ST_CYC_ST_COR_OPM_1	6x8	0...FFH	-48...47.625	0.375	°CRK
LDPM_CYC_1_IGA_ST	6	0...FFH	0...255	1	-
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	°C
Correction of the basic ignition angle at start dependen on the start cycle counter at operation mode 1					
IP_IGA_ST_CYC_ST_COR_OPM_2	6x8	0...FFH	-48...47.625	0.375	°CRK
LDPM_CYC_1_IGA_ST	6	0...FFH	0...255	1	-
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	°C
Correction of the basic ignition angle at start dependen on the start cycle counter at operation mode 2					

39.2.1 IGSP_M6004

FUNCTION DESCRIPTION:

General information:

The ignition angle at start is basically dependent on the engine speed N and coolant temperature TCO determined by the map IP_IGA_ST_BAS. Additionally a map dependent on the ambient pressure AMP corrects this basic ignition angle. To avoid backfiring due to the fast increase of the combustion chamber temperature a second correction dependent on the cycle counter CYC_ST and the coolant temperature is added. A further map consideres influences of the inrake air temperature TIA.

The switch LV_IGA_ST_OPM_SEL is used to determine if either the basic ignition angle at start for operation mode 1 or the basic ignition angle at start for operation mode 2 is calculated.

In case of MSA start (engine start stop automatic) there is an additive correction dependent on the relative engine position PSN_ENG_REL_ST for the first ignition.

Application conditions :

Activation : LV_ST = 1 or (LV_ES = 1 and LV_SYN_ENG = 1)

Deactivation : LV_ST = 0 and (LV_ES = 0 or LV_SYN_ENG = 0)

Update rate : every 10 ms and every TDC

Initialisation: at reset : *IGA_ST = 0*
LV_IGA_ST_PSN_ENG_COR_STST = 0
LV_PSN_ENG_REL_TMP = 0

<u>Remark</u>: if LV_IGA_ST_OPM_SEL = 1 only the upper part has to be executed (ΣIP....OPM_2), else only the lower part has to be executed (ΣIP....OPM_1).

Application Condition

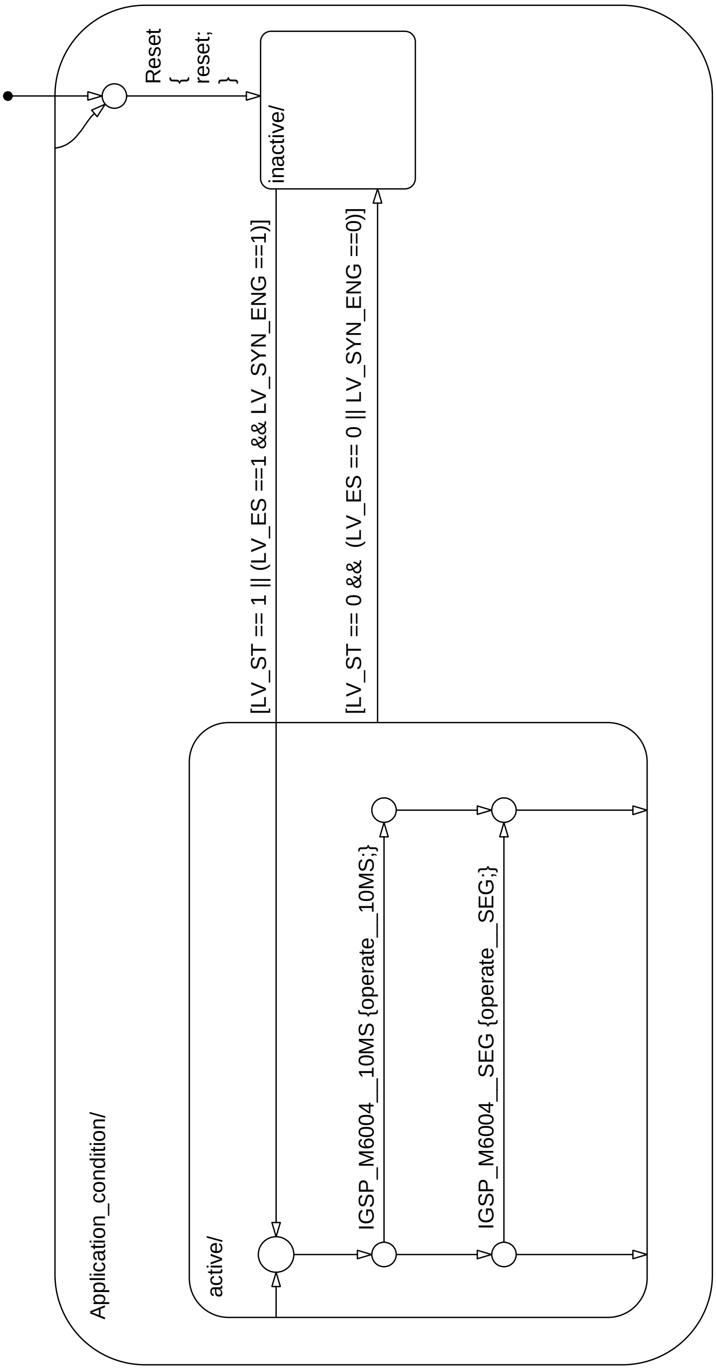
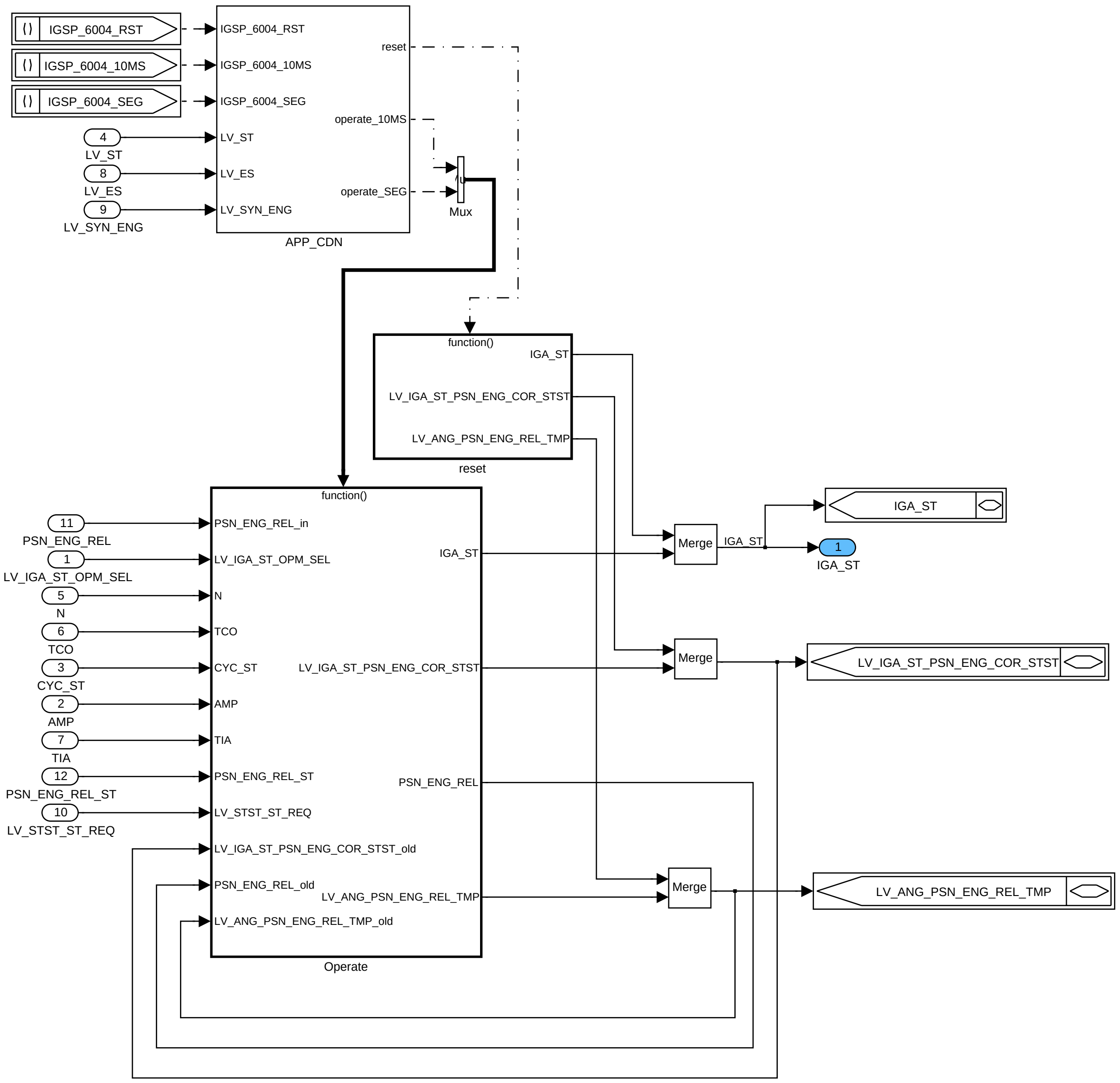


Figure 24 IGSP_M6004/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 25−Apr−2005

Figure 25 IGSP_M6004

39.2.1.1 Reset

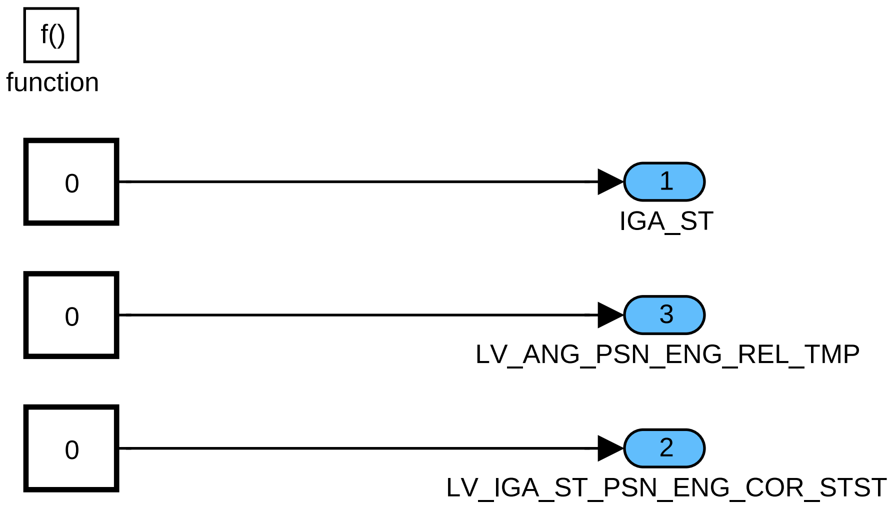


Figure 26 IGSP_M6004/ reset

39.2.1.2 Overview of IGA_ST calculation

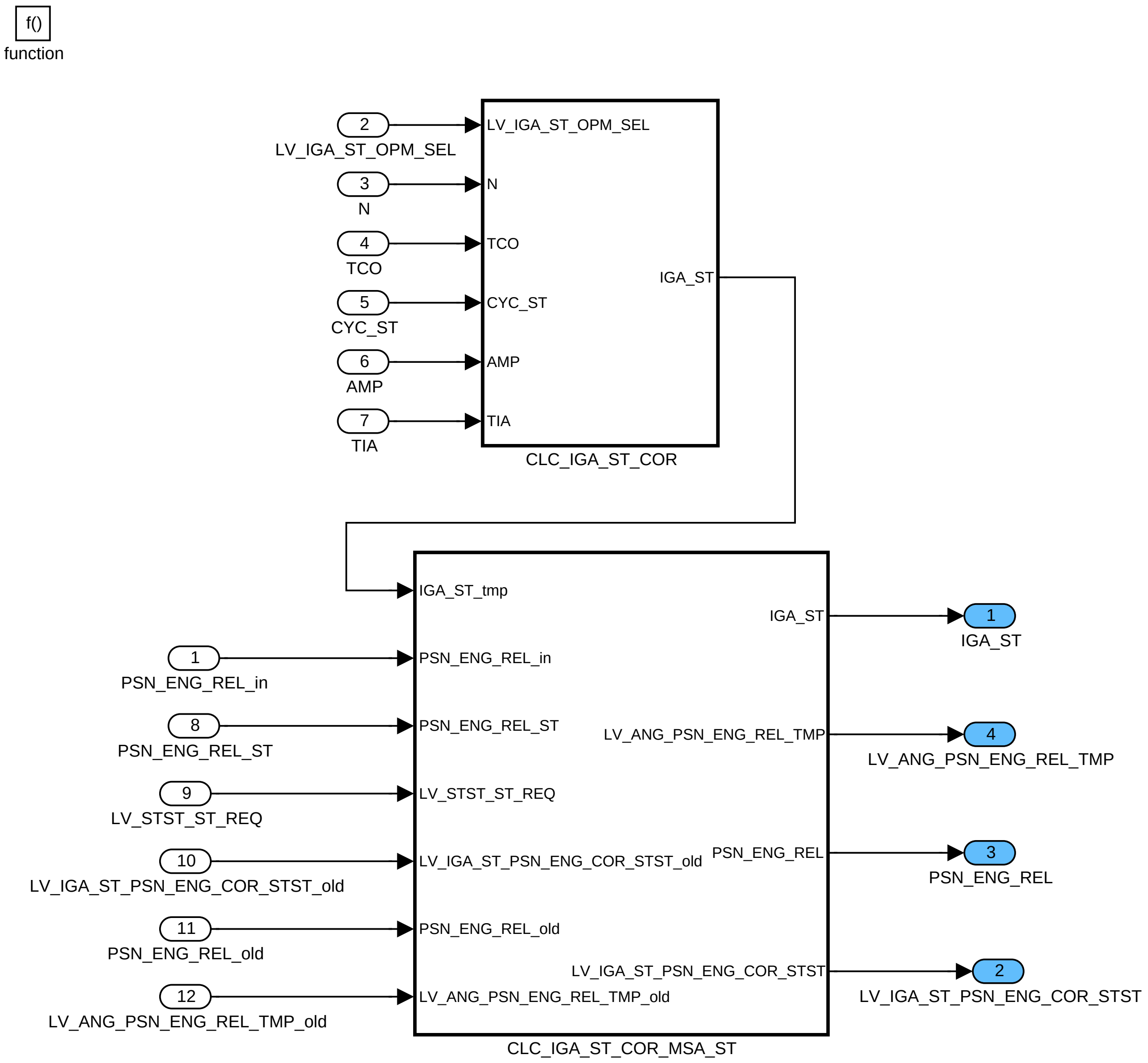


Figure 27 IGSP_M6004/ Operate

Calculation of standard IGA_ST correction

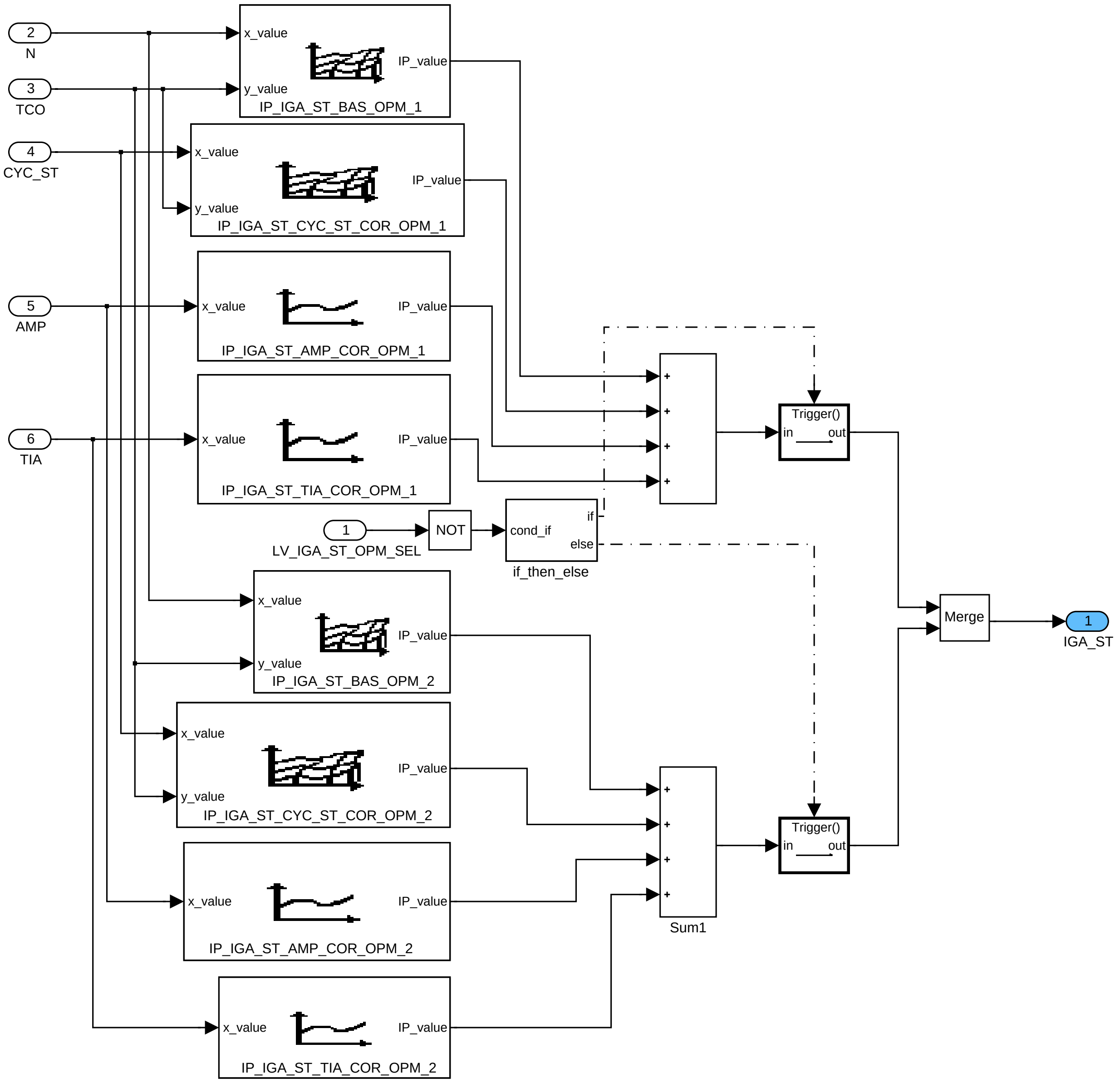


Figure 28 IGSP_M6004/ Operate/ CLC_IGA_ST_COR

Calculation of IGA_ST correction in case of MSA start (engine start stop automatic)

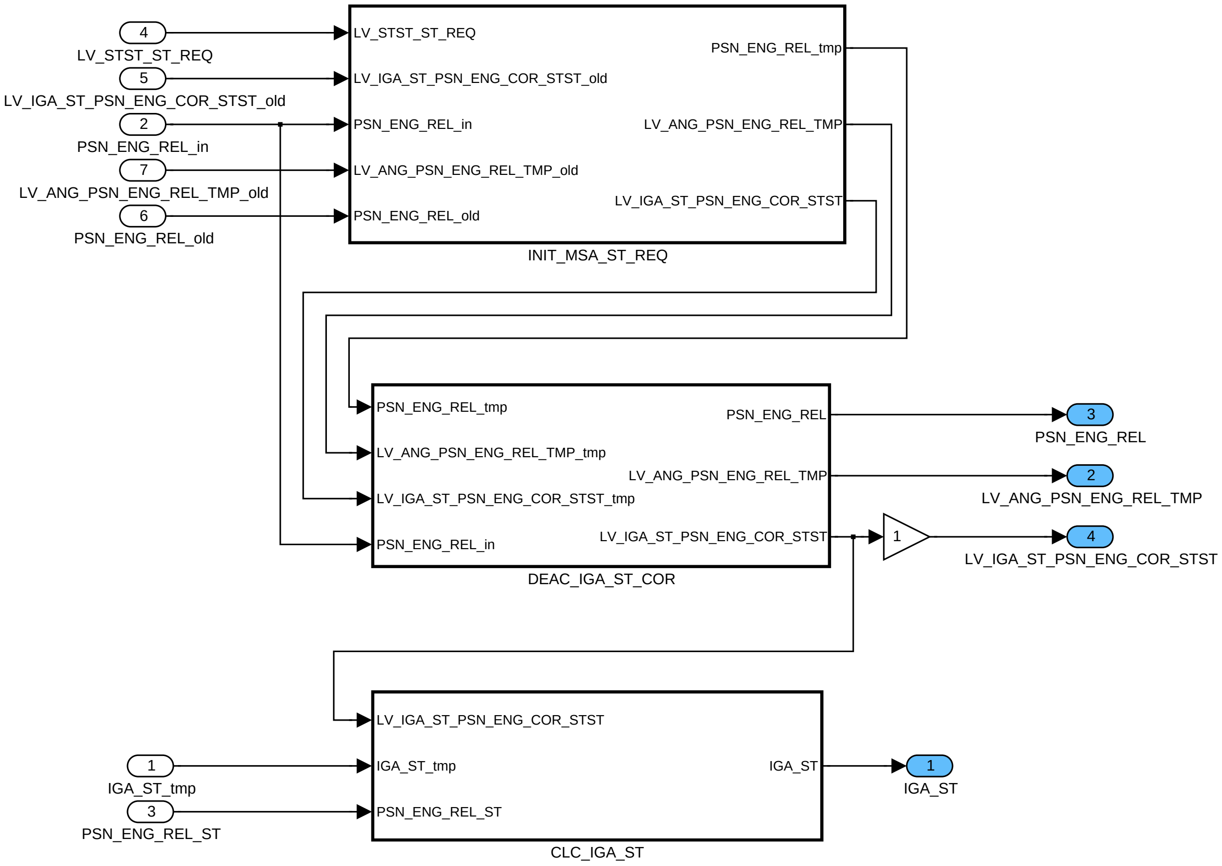


Figure 29 IGSP_M6004/ Operate/ CLC_IGA_ST_COR_MSA_ST

Initialisation at MSA start request

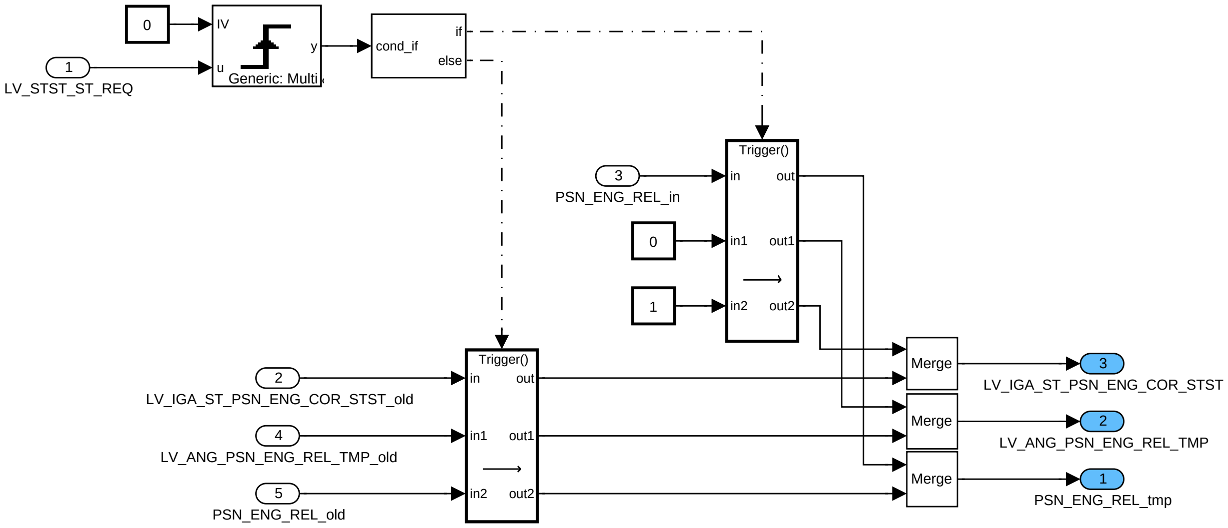


Figure 30 IGSP_M6004/ Operate/ CLC_IGA_ST_COR_MSA_ST/ INIT_MSA_ST_REQ

Deactivation of IGA_ST correction

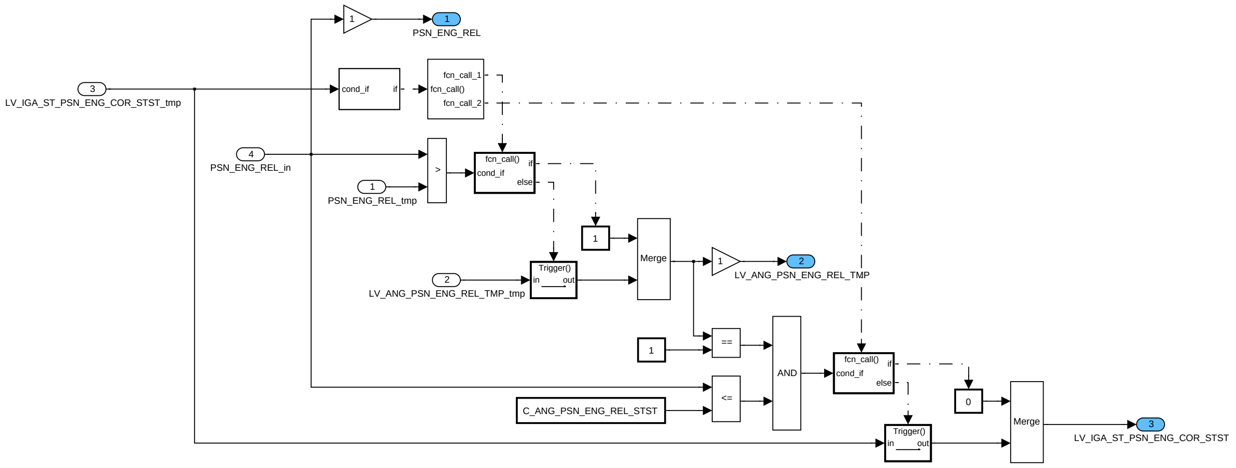


Figure 31 IGSP_M6004/ Operate/ CLC_IGA_ST_COR_MSA_ST/ DEAC_IGA_ST_COR

Calculation of corrected IGA_ST

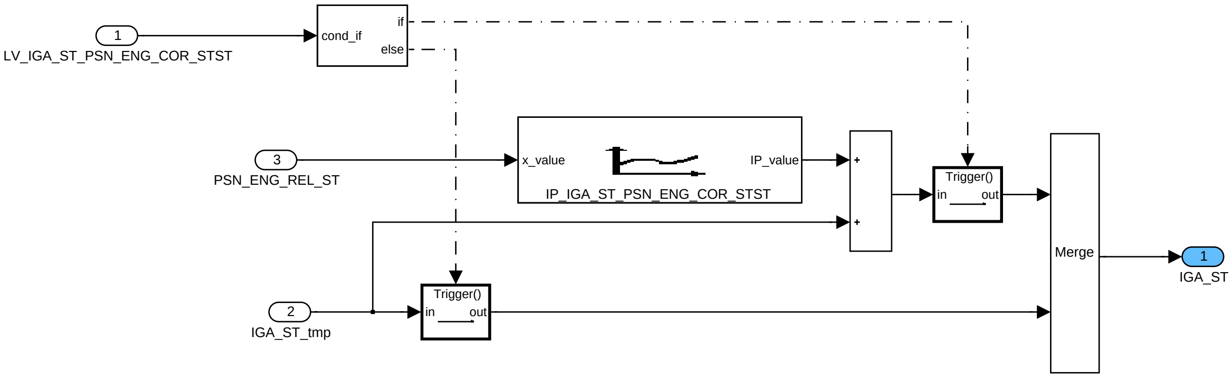


Figure 32 IGSP_M6004/ Operate/ CLC_IGA_ST_COR_MSA_ST/ CLC_IGA_ST

39.3 Minimum Ignition Angle

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_DIF_MIN_H_RNG	V/O	0...B40H	0...-180	-0.0625	[°CRK]
Difference of minimum ignition angle to reference ignition angle in wide range					
IGA_DIF_MIN_DELTA_H_RNG	V	0...B40H	0...-180	-0.0625	[°CRK]
Extra retard on IGA_DIF_MIN in case of torque intervention in wide range					
LV_IGA_DIF_MIN_DELTA_ACT	V	0...1H	0...1	1	[-]
Flag to enable and declare the acting of the extra retard on IGA_DIF_MIN					
IGA_DIF_MIN_AST_H_RNG	V	FA60...5A0H	-90...90	0.0625	[°CRK]
Intermediate variable: correction on IGA_DIF_MIN for combustion cycles after start management in wide range					
IGA_MIN_BAS_1_H_RNG	V	FA60...5A0H	-90...90	0.0625	[°CRK]
Intermediate variable: basic minimum ignition angle in wide range					
IGA_MIN_BAS_PU_H_RNG	V	FA60...5A0H	-90...90	0.0625	[°CRK]
Intermediate variable: PU minimum ignition angle in wide range					
IGA_MIN_BAS_EXT_H_RNG	V	FA60...5A0H	-90...90	0.0625	[°CRK]
Intermediate variable: basic minimum ignition angle in wide range					
IGA_MIN_BAS_PUC_H_RNG	V	FA60...5A0H	-90...90	0.0625	[°CRK]
Intermediate variable: basic minimum ignition angle in wide range					
IGA_MIN_BAS_INT_H_RNG	V	FA60...5A0H	-90...90	0.0625	[°CRK]
Intermediate variable: basic minimum ignition angle in wide range					
IGA_MIN_BAS_H_RNG	V	FA60...5A0H	-90...90	0.0625	[°CRK]
basic minimum ignition angle in wide range					
IGA_MIN_BAS_2_H_RNG	V	FA60...5A0H	-90...90	0.0625	[°CRK]
basic minimum ignition angle including cat heating in wide range					
IGA_MIN_BAS_OPM_H_RNG	V	0...B40H	-90...90	0.0625	[°CRK]
basic minimum ignition angle depening on engine operation mode					
IGA_MIN_BAS_2_OPM_1_H_RNG	V	0...B40H	-90...90	0.0625	[°CRK]
basic minimum ignition angle high range regarding cat heating and OPM 1					
IGA_MIN_BAS_2_OPM_2_H_RNG	V	0...B40H	-90...90	0.0625	[°CRK]
basic minimum ignition angle high range regarding cat heating and OPM 2					

Input data:

LV_HOM_RUN	MAF_HB	N_32	LV_IGA_MIN_PUC
IGA_DIF_SP_H_RNG	TCO	LV_AST_END	CYC_CAST
LV_AT	AMP	FAC_IGA_OPM_SEL	~~TQ_ADD_CH~~
OPM_AV	OPM_REQ	LV_IGA_MIN_EXT	IGA_REF_COR
	N_DIF	~~LV_CH_TQ_ADD~~	~~LV_CH_TQ_ADD_IS~~
		LV_TQ_IGA_ACT	LV_HOM_ACT

FUNCTION DESCRIPTION:

The minimum ignition angle is the minimum allowable ignition angle which is possible to apply to the engine. The basic minimum ignition angle determined through a calibration map is corrected by a temperature dependent additive term. Furthermore there are additive influences by cat heating and engine operating mode like throttled or unthrottled.In some special engine running states, the minimal ignition angle calibrated in this function can be limited by external request to avoid too strong spark retards during those states.Hence, in all practical cases IGA_DIF_MIN_H_RNG is a negative value which is the maximum allowable spark retard determined by a calibration map. It includes the coolant temperature correction (*for engine protection, pollution condition, etc…*).

39.3.1 IGA_DIF_MIN_H_RNG calculation with extra spark retard for torque intervention and after start management

General information:

In case of a strong fast torque intervention requested (*ie. if IGA_DIF_SP_H_RNG hence requested by torque intervention becomes lower than IGA_DIF_MIN_0_H_RNG*) then a specific over-retard (*IGA_DIF_MIN_DELTA_H_RNG*) is authorised on minimum ignition angle for a while. It permits to manage a strongest response from the engine to a torque reduction request. IGA_DIF_MIN_0_*H_RNG* is just an intermediate variable introduced here for clarification of explanations. IGA_DIF_MIN_0_*H_RNG* is just the result of the interpolation map without other affects. This over-retard is only allowed for a given time (*C_IGA_DIF_MIN_DELTA_TEMP_ENA*). In case of a rising edge of the flag for enabling extra-retard (LV_IGA_DIF_MIN_DELTA_ACT= 0->1) the active counter is reseted to C_IGA_DIF_MIN_DELTA_TEMP_ENA. If torque intervention by spark retard is active as well as homogeneous ignition angle and injection parameters are applied to the engine (LV_TQ_IGA_ACT=1 and LV_HOM_ACT=1) the active counter is decremented to 0, otherwise it is incremented back to C_IGA_DIF_MIN_DELTA_TEMP_ENA. The counter for extra-retard is set to avoid exhaust gas over-heating and thus protect the exhaust branch. At the end of this counter, this over-retard on IGA_DIF_MIN_H_RNG goes back up to 0 with a limited gradient (*C_IGA_DIF_MIN_DELTA_LGRD*_H_RNG) for its come-back law. This limited gradient for come-back is necessary to avoid the changes on ignition angle applied can be felt on-board the car. At the end of this gradient limited come-back, such an other kind of extra retard is inhibited during a while (*C_IGA_DIF_MIN_DELTA_TEMP_ENA*) just to be sure to avoid any problems of exhaust gas over-heating. This functionality aims to deal with thermal exchange effects. So, the timer are in time units (*sec.*) and the calibration map here defined for extra retard is related to LDP of temperature.

Application conditions:

Activation:	LV_HOM_RUN	= 1	
Deactivation:	LV_HOM_RUN	= 0	
Initialisation:	IGA_DIF_MIN_H_RNG	= 0° CRK	at reset
	IGA_DIF_MIN_0_H_RNG	= 0 °CRK	
	IGA_DIF_MIN_AST_H_RNG	= 0 °CRK	
	IGA_DIF_MIN_DELTA_H_RNG	= 0 °CRK	
	IGA_MIN_BAS_x_H_RNG	= -35.625°CRK	
Update Rate:	whole functionality	=> 10 ms	
	IP_IGA_DIF_MIN_TCO_H_RNG	=> 1000 ms	

Formula section:

Condition to enable the extra-spark retard:

If LV_IGA_DIF_MIN_DELTA_ACT = 0
And When IGA_DIF_SP_H_RNG becomes lower than IGA_DIF_MIN_0_*H_RNG*$_{n-1}$
Then LV_IGA_DIF_MIN_DELTA_ACT = 1

```
    And            IGA_DIF_MIN_DELTA_H_RNG = IGA_DIF_MIN_DELTA_1_H_RNG

Elself             LV_IGA_DIF_MIN_DELTA_ACT = 1
    And When       IGA_DIF_SP_H_RNG becomes lower than IGA_DIF_MIN_0_H_RNG$_{n-1}$
Then               IGA_DIF_MIN_DELTA_H_RNG = IGA_DIF_MIN_DELTA_1_H_RNG

Else               if    IGA_DIF_SP_H_RNG is greater than IGA_DIF_MIN_0_H_RNG$_{n-1}$
                   And   LV_IGA_DIF_MIN_DELTA_ACT = 0 or 1
                         (Whatever should be the state of LV_IGA_DIF_MIN_DELTA_ACT)

Then               LV_IGA_DIF_MIN_DELTA_ACT      remains in its previous state
    And            IGA_DIF_MIN_DELTA_H_RNG = 0
EndIf
```

IGA_DIF_MIN_DELTA_1_H_RNG and IGA_DIF_MIN_0_H_RNG are only an intermediate variable just used for calculation and clarification

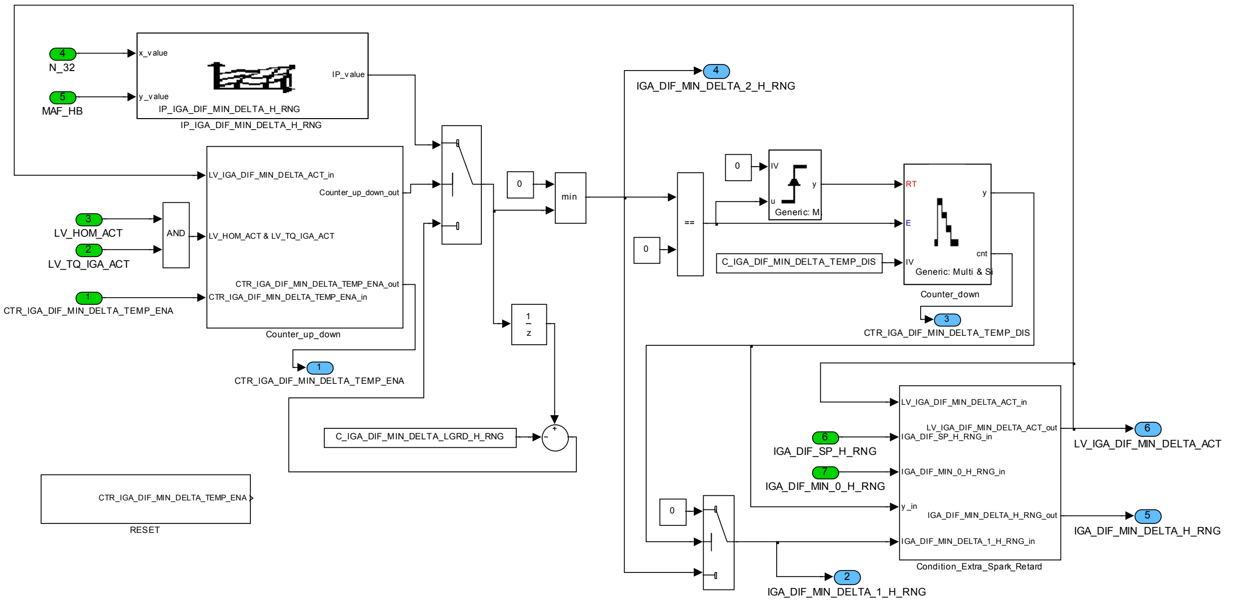
Management of extra spark retard:

```
IF        (LV_IGA_DIF_MIN_DELTA_ACT$_n$ = 1 and
              LV_IGA_DIF_MIN_DELTA_ACT$_{n-1}$ = 0)
THEN      CTR_IGA_DIF_MIN_DELTA_TEMP_ENA$_n$ =
              C_IGA_DIF_MIN_DELTA_TEMP_ENA
                              the timer for extra-retard authorised is reseted
ENDIF


IF        (LV_HOM_ACT = 1 and LV_TQ_IGA_ACT = 1)
THEN      CTR_IGA_DIF_MIN_DELTA_TEMP_ENA$_n$ =
              CTR_IGA_DIF_MIN_DELTA_TEMP_ENA$_{n-1}$ - 1
                              the timer for extra-retard authorised is decremented
          IGA_DIF_MIN_DELTA_1_H_RNG =
              IP_IGA_DIF_MIN_DELTA_H_RNG (N_32,MAF_HB)
ELSE      CTR_IGA_DIF_MIN_DELTA_TEMP_ENA$_n$ =
              CTR_IGA_DIF_MIN_DELTA_TEMP_ENA$_{n-1}$ + 1
                              the timer for extra-retard authorised is incremented
          IGA_DIF_MIN_DELTA_1_H_RNG =
              IP_IGA_DIF_MIN_DELTA_H_RNG (N_32,MAF_HB)
ENDIF
IF        CTR_IGA_DIF_MIN_DELTA_TEMP_ENA$_n$ = 0
                              at the end of timer for extra-retard authorized
THEN      IGA_DIF_MIN_DELTA_2_H_RNG$_n$ =
              Min[(IGA_DIF_MIN_DELTA_H_RNG$_{n-1}$ –
                      C_IGA_DIF_MIN_DELTA_LGRD_H_RNG), 0]
          IGA_DIF_MIN_DELTA_1_H_RNG$_n$ = IGA_DIF_MIN_DELTA_2_H_RNG$_n$
                              Gradient limited come-back of the over-retard bordered to 0
ENDIF
```

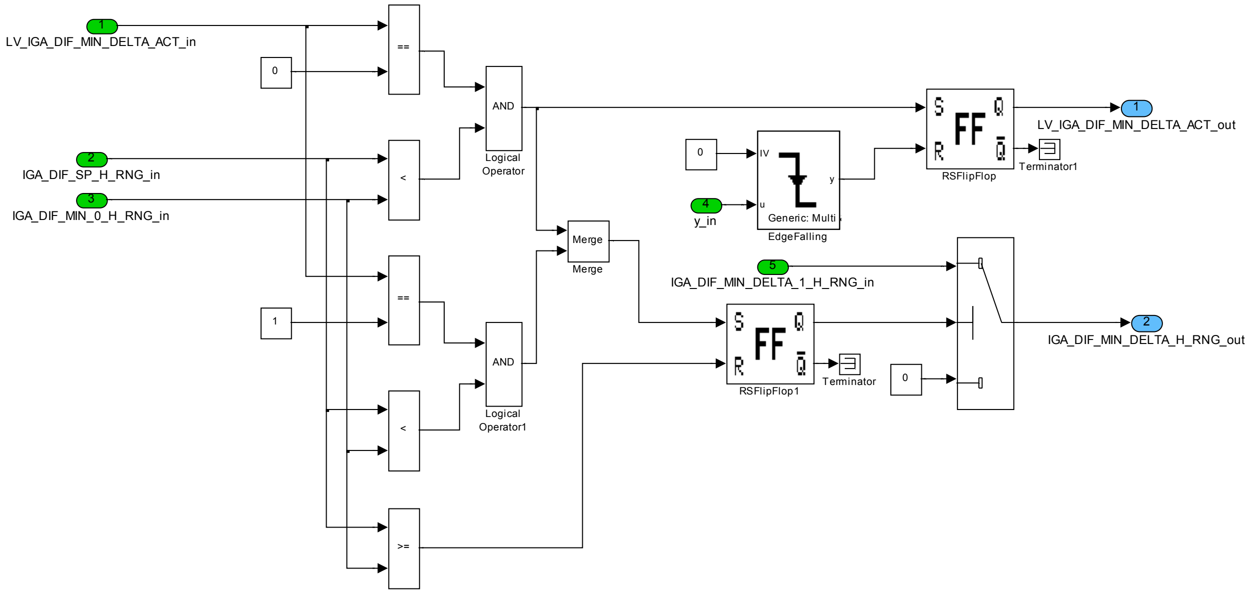
```
IF      IGA_DIF_MIN_DELTA_2_H_RNG = 0
                        at the end of the gradient limited come-back to 0
THEN    CTR_IGA_DIF_MIN_DELTA_TEMP_DIS$_n$ =
            C_IGA_DIF_MIN_DELTA_TEMP_DIS

        IF      CTR_IGA_DIF_MIN_DELTA_TEMP_DIS$_n$ > 0
        THEN    IGA_DIF_MIN_DELTA_1_H_RNG$_n$ = 0
                CTR_IGA_DIF_MIN_DELTA_TEMP_DIS$_n$ =
                    CTR_IGA_DIF_MIN_DELTA_TEMP_DIS$_{n-1}$ -1
                        Inhibition of another over-retard at the end of a first one
                        during the lasting of the timer C_IGA_DIF_MIN_DELTA_TEMP_DIS
        ELSE
                IF      (CTR_IGA_DIF_MIN_DELTA_TEMP_DIS$_n$ = 0 and
                            (CTR_IGA_DIF_MIN_DELTA_TEMP_DIS$_{n-1}$ = 1)
                THEN    LV_IGA_DIF_MIN_DELTA_ACT = 0
                        At the end of the inhibition time, another extra retard is
                        allowed again
                ENDIF
        ENDIF
ENDIF
```

Note: CTR_IGA_DIF_MIN_DELTA_TEMP_ENA, IGA_DIF_MIN_DELTA_2_H_RNG and CTR_IGA_DIF_MIN_DELTA_TEMP_DIS are only intermediate variables just used for calculation and clarification

Management of extra spark retard

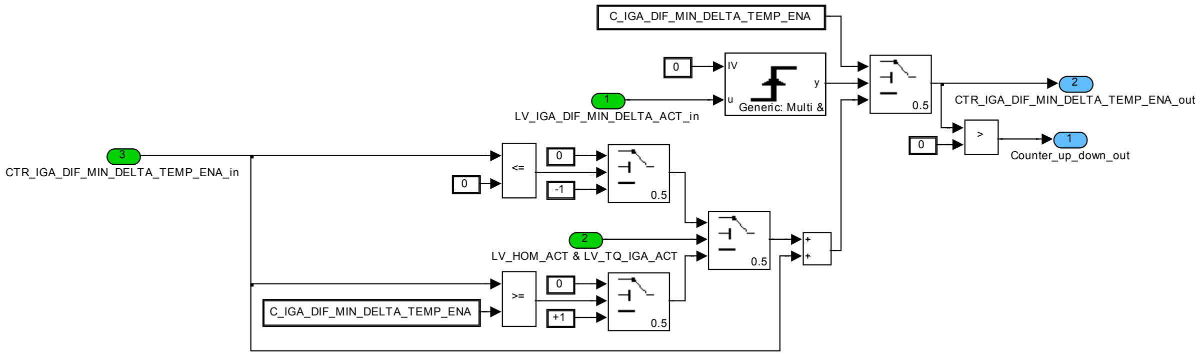


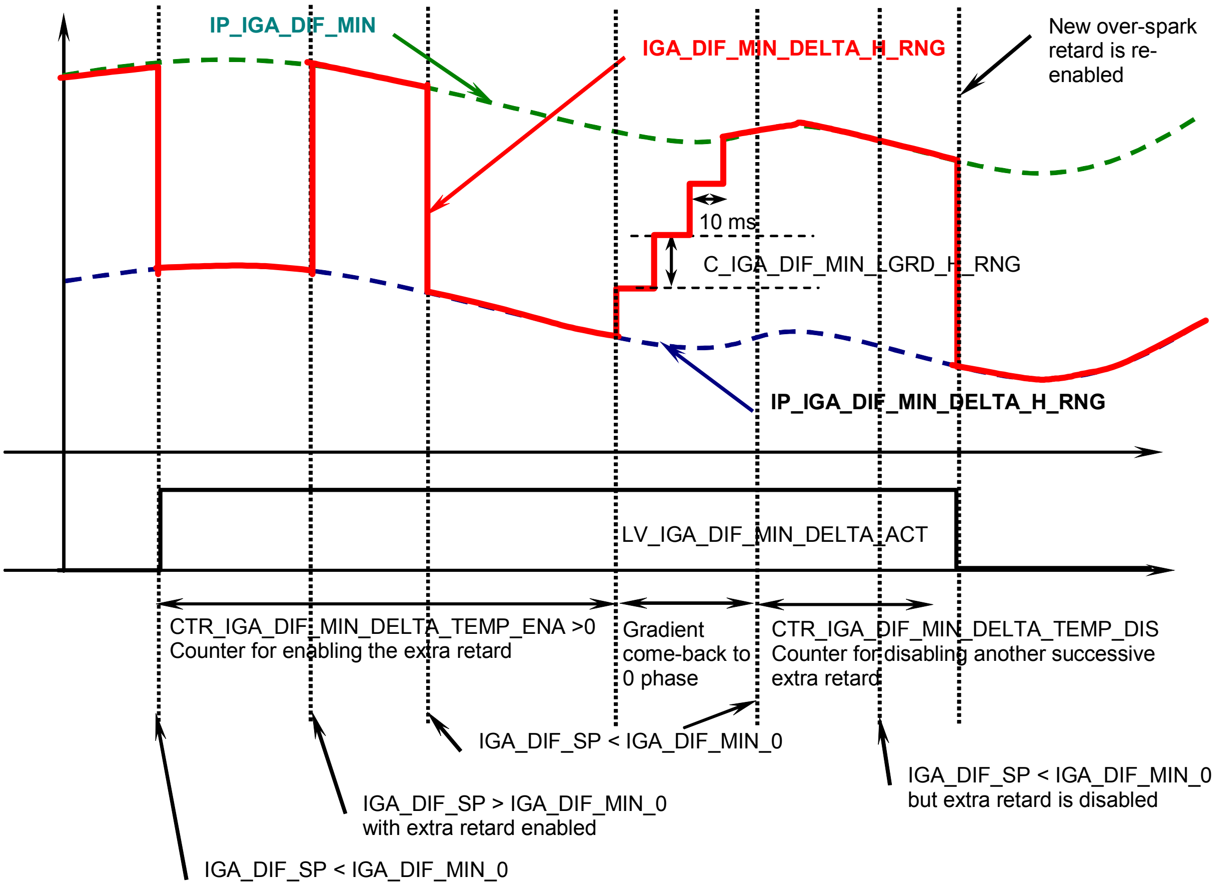
Condition_Extra_Spark_Retard



general specification

Counter_up_down





After start correction on IGA_DIF_MIN_H_RNG:every 10ms ***if NLC_USE_IGA_MIN_AST = 1***

The following calculations are only performed in homogenous mode during the after start phase (LV_AST_END = 0) and only if considered by project management choice (if NLC_USE_IGA_MIN_AST = 1). In other case, they can be switched off. The update rate is 10ms.

```
If      LV_AT = 1   and   LV_AST_END = 0   and   NLC_USE_IGA_MIN_AST = 1
Then    IGA_DIF_MIN_AST_H_RNG =      (IP_IGA_DIF_MIN_CYC_AT         +
                                 IP_IGA_DIF_MIN_CYC_AMP_AT_H_RNG         *
                                 IP_FAC_IGA_DIF_MIN_AMP) *
                                 IP_FAC_IGA_DIF_AST

elseif  LV_AT = 0   and   LV_AST_END = 0   and   NLC_USE_IGA_MIN_AST = 1
then    IGA_DIF_MIN_AST_H_RNG =      (IP_IGA_DIF_MIN_CYC_MT   +
                                  IP_IGA_DIF_MIN_CYC_AMP_MT_H_RNG        *
                                 IP_FAC_IGA_DIF_MIN_AMP) *
                                 IP_FAC_IGA_DIF_AST

ElseIf  IGA_DIF_MIN_AST_H_RNG = 0                 the computation can be
switched off
EndIf
```

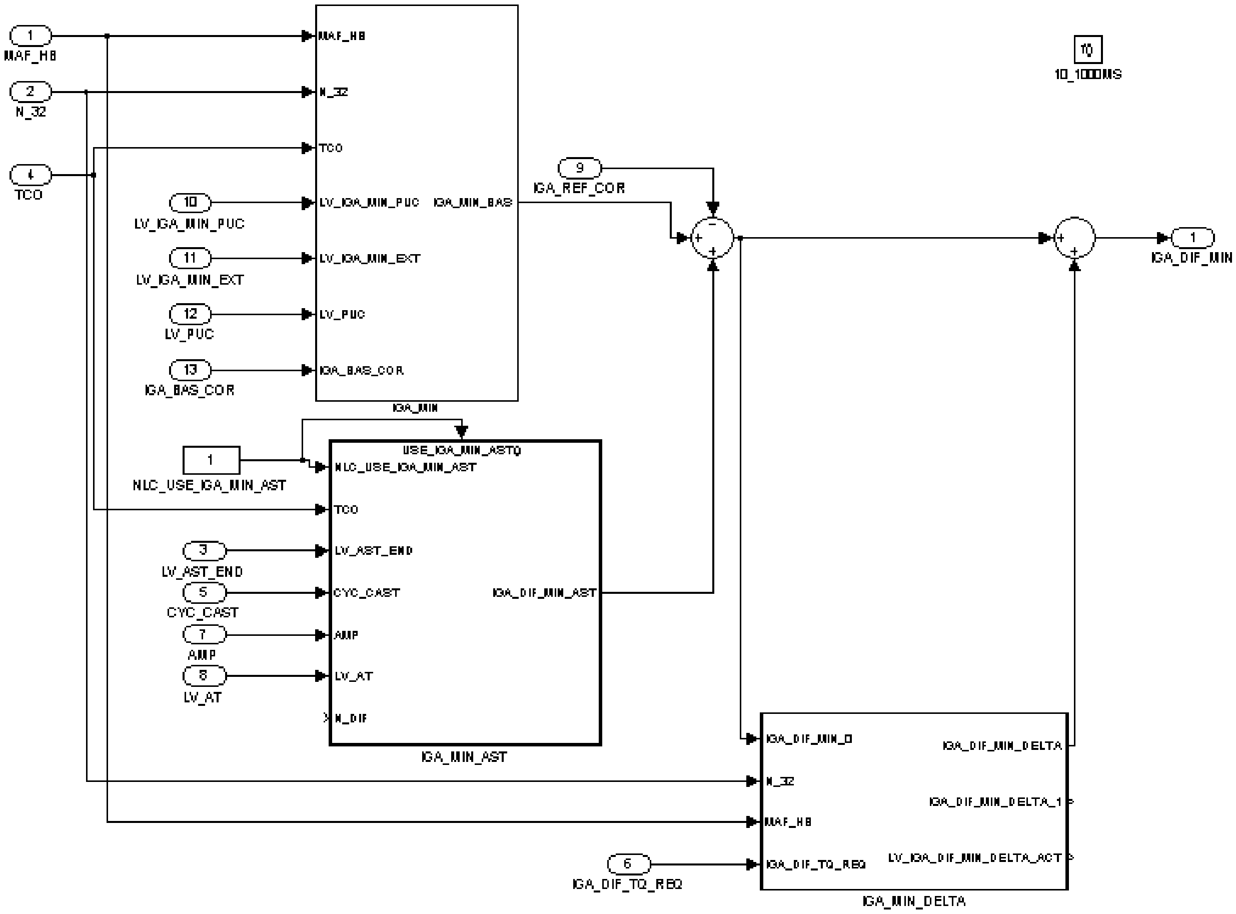
39.3.2 Limitation of IGA_DIF_MIN_H_RNG for after start management

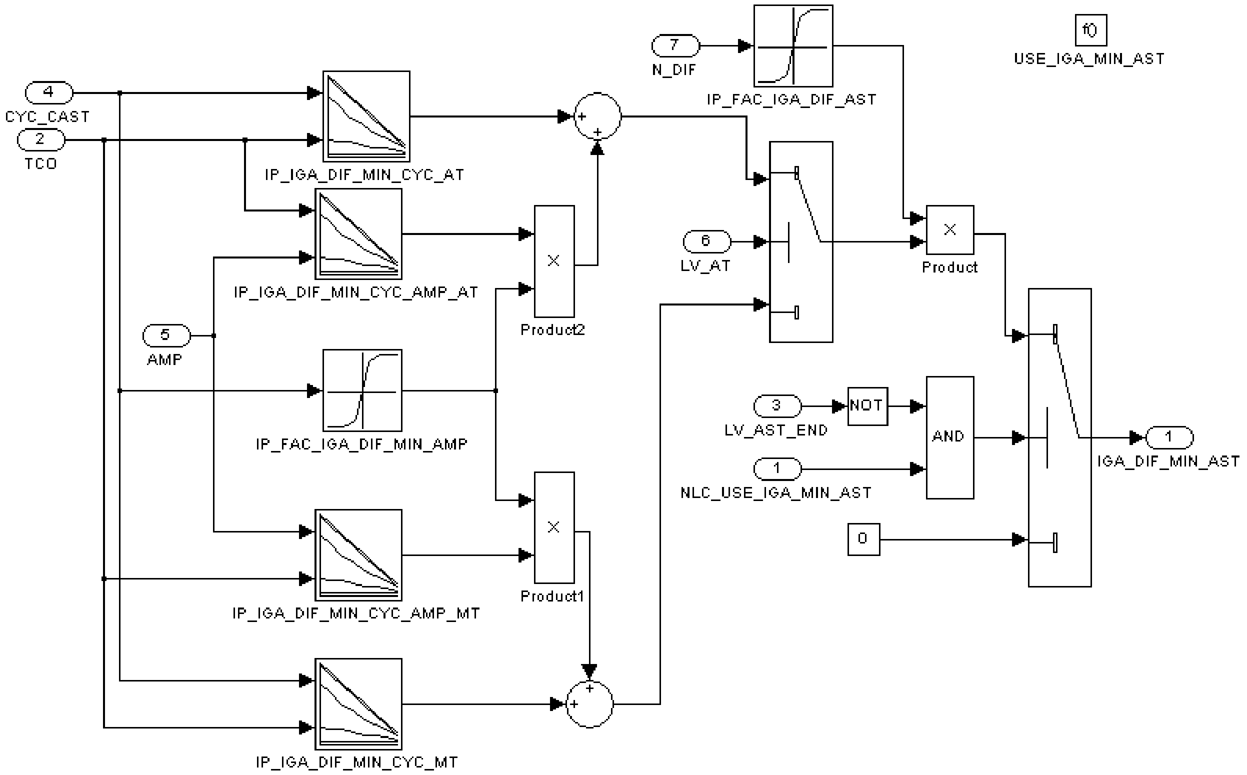
General information:

After start phase it is possible to include a specific correction depending on engine cycles after start and engine coolant temperature to the minimum ignition angle calculated from basic maps to improve the efficiency of catalyst heating.

Note: *All this following part is only computed – and integrated in the SW –* ***if NLC_USE_IGA_MIN_AST = 1***

Signal flow diagram:

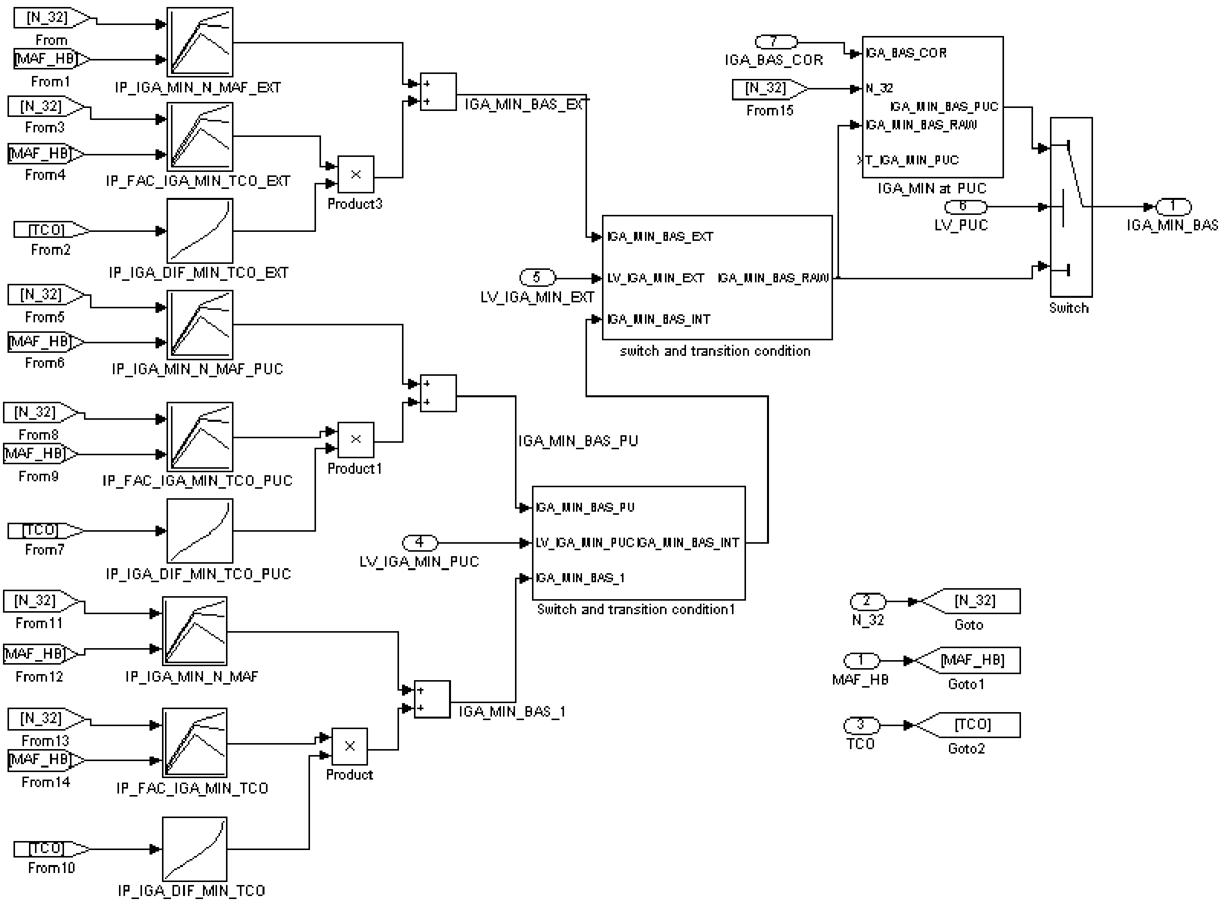




7
N_DIF
IP_FAC_IGA_DIF_AST
f()
USE_IGA_MIN_AST
4
CYC_CAST
2
TCO
IP_IGA_DIF_MIN_CYC_AT
IP_IGA_DIF_MIN_CYC_AMP_AT
6
LV_AT
Product
Product2
5
AMP
IP_FAC_IGA_DIF_MIN_AMP
3
LV_AST_END
NOT
AND
1
NLC_USE_IGA_MIN_AST
1
IGA_DIF_MIN_AST
0
Product1
IP_IGA_DIF_MIN_CYC_AMP_MT
IP_IGA_DIF_MIN_CYC_MT

39.3.3 IGA_MIN switch conditions

Signal flow diagram:



Additional maps are introduced to consider the influence of cat heating and throttled engine mode. IP_IGA_MIN_CH_COR and IP_IGA_MIN_OPM_COR are additive corrections on IGA_MIN_BAS_H_RNG. Each one is depending on N_32 and MAF_HB. The combined sum results in IGA_MIN_BAS_OPM.

The fading of IP_IGA_MIN_CH_COR in depence of operation mode is managed by function function OPM_SWI.

Formula description:

IF OPM_REQ_CUS = 66 **or** OPM_REQ_CUS = 82

THEN

IGA_MIN_BAS_2_H_RNG = IGA_MIN_BAS_H_RNG + IP_IGA_MIN_CH_COR_H_RNG

ELSE

IGA_MIN_BAS_2_H_RNG = IGA_MIN_BAS_H_RNG ~~+ IP_IGA_MIN_CH_COR_H_RNG~~

ENDIF

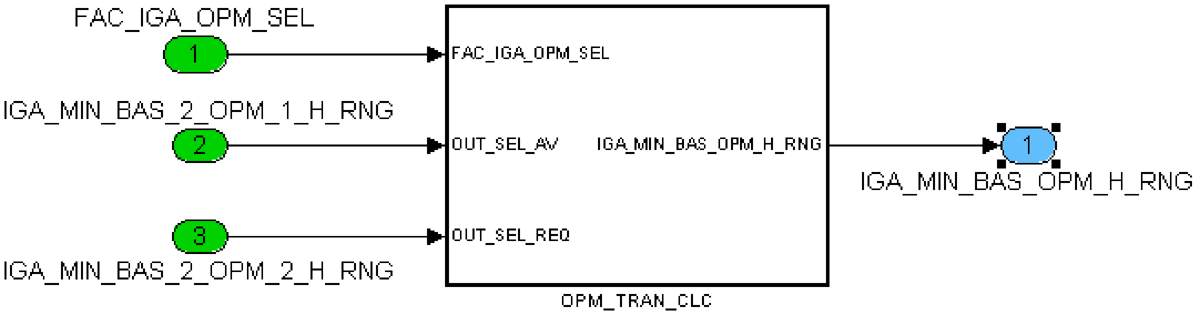
IGA_MIN_BAS_2_OPM_1_H_RNG = IGA_MIN_BAS_2_H_RNG

IGA_MIN_BAS_2_OPM_2_H_RNG = IGA_MIN_BAS_2_H_RNG + IP_IGA_MIN_OPM_COR_H_RNG

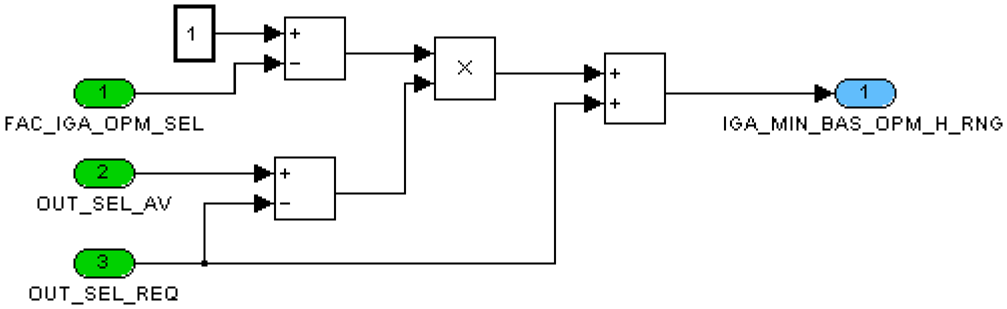
IGA_MIN_BAS_OPM_H_RNG = Function call OPM_SWI (described in chapter 1.5)

39.3.4 Operation mode manager (function OPM_SWI)

There are two input values IGA_MIN_BAS_2_OPM_1_H_RNG, IGA_MIN_BAS_2_OPM_2_H_RNG. Each input value belongs to a special operation mode OPM_1, OPM_2. Moreover the value OPM_SEL_AV indicates the active and OPM_SEL_REQ indicates the requested operation mode. The aim of the operation manager is to create an output value depending on the active operation mode. If there is a switch from one mode to another a smooth changeover from one input value to the other in possible by using the interpolation factor FAC_IGA_OPM_SEL .



Operation mode manager



OPM_TRAN_CLC

To prevent a jump in the IGA_MIN a transition is necessary at switching to an other IGA_MIN_BAS_x_H_RNG with the change limitation of C_IGA_DIF_MIN_BAS_LGRD_H_RNG.

LV_IGA_MIN_PUC == from 0 to 1 **and** LV_IGA_MIN_EXT == 0:

IF IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - IGA_MIN_BAS_PU_H_RNG$_n$ > 0

Then IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - C_IGA_DIF_MIN_BAS_LGRD_H_RNG

```
    until |IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - IGA_MIN_BAS_PU_H_RNG$_n$| <= C_IGA_DIF_MIN_BAS_LGRD_H_RNG
        Then IGA_MIN_BAS_INT_H_RNG$_n$ = IGA_MIN_BAS_PU_H_RNG
else
    IF IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - IGA_MIN_BAS_PU_H_RNG$_n$ < 0
    Then IGA_MIN_BAS_INT_H_RNG$_{n-1}$ + C_IGA_DIF_MIN_BAS_LGRD_H_RNG
        until |IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - IGA_MIN_BAS_PU_H_RNG$_n$| <= C_IGA_DIF_MIN_BAS_LGRD_H_RNG
            Then IGA_MIN_BAS_INT_H_RNG$_n$ = IGA_MIN_BAS_PU_H_RNG
```

LV_IGA_MIN_PUC == from 1 to 0 **and** LV_IGA_MIN_EXT == 0:

```
IF IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - IGA_MIN_BAS_1_H_RNG$_n$ > 0
Then IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - C_IGA_DIF_MIN_BAS_LGRD_H_RNG
    until |IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - IGA_MIN_BAS_1_H_RNG$_n$| <= C_IGA_DIF_MIN_BAS_LGRD_H_RNG
        Then IGA_MIN_BAS_INT_H_RNG$_n$ = IGA_MIN_BAS_1_H_RNG
else
    IF IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - IGA_MIN_BAS_1_H_RNG$_n$ < 0
    Then IGA_MIN_BAS_INT_H_RNG$_{n-1}$ + C_IGA_DIF_MIN_BAS_LGRD_H_RNG
        until |IGA_MIN_BAS_INT_H_RNG$_{n-1}$ - IGA_MIN_BAS_1_H_RNG$_n$|<= C_IGA_DIF_MIN_BAS_LGRD_H_RNG
            Then IGA_MIN_BAS_INT_H_RNG$_n$ = IGA_MIN_BAS_1_H_RNG
```

LV_IGA_MIN_EXT == from 0 to 1:

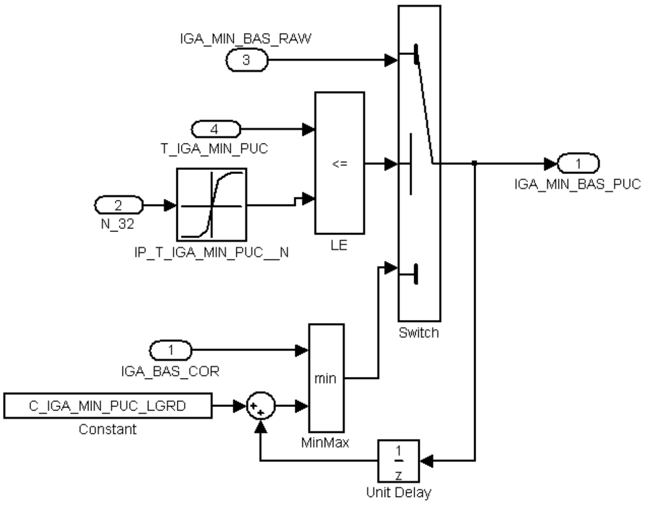
```
IF IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - IGA_MIN_BAS_EXT_H_RNG$_n$ > 0
Then IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - C_IGA_DIF_MIN_BAS_LGRD_H_RNG
    until |IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - IGA_MIN_BAS_EXT_H_RNG$_n$| ≤ C_IGA_DIF_MIN_BAS_LGRD_H_RNG
        Then IGA_MIN_BAS_RAW_H_RNG$_n$ = IGA_MIN_BAS_EXT_H_RNG
else
  IF IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - IGA_MIN_BAS_EXT_H_RNG$_n$ < 0
  Then IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ + C_IGA_DIF_MIN_BAS_LGRD_H_RNG
  until |IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - IGA_MIN_BAS_EXT_H_RNG$_n$| ≤ C_IGA_DIF_MIN_BAS_LGRD_H_RNG
        Then IGA_MIN_BAS_RAW_H_RNG$_n$ = IGA_MIN_BAS_EXT_H_RNG
```

LV_IGA_MIN_EXT == from 1 to 0:

```
IF   IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - IGA_MIN_BAS_INT_H_RNG$_{n}$ > 0
Then       IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - C_IGA_DIF_MIN_BAS_LGRD_H_RNG
    until      |IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - IGA_MIN_BAS_INT_H_RNG$_{n}$| ≤ C_IGA_DIF_MIN_BAS_LGRD_H_RNG
        Then IGA_MIN_BAS_RAW_H_RNG$_{n}$ = IGA_MIN_BAS_INT_H_RNG
else
  IF IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - IGA_MIN_BAS_INT_H_RNG$_{n}$ < 0
  Then    IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ + C_IGA_DIF_MIN_BAS_LGRD_H_RNG
     until |IGA_MIN_BAS_RAW_H_RNG$_{n-1}$ - IGA_MIN_BAS_INT_H_RNG$_{n}$| ≤ C_IGA_DIF_MIN_BAS_LGRD_H_RNG
          Then IGA_MIN_BAS_RAW_H_RNG$_{n}$ = IGA_MIN_BAS_INT_H_RNG
```

LV_IGA_MIN_EXT = = 1:

```
IF    LV_IGA_MIN_PUC = = 0
Then        IGA_MIN_BAS_INT_H_RNG = IGA_MIN_BAS_1_H_RNG
else        IGA_MIN_BAS_INT_H_RNG = IGA_MIN_BAS_PU_H_RNG
endif
```



```
IF   LV_PUC$_{n}$ = 1  AND  LV_PUC$_{n-1}$ = 0  AND  LC_INH_IGA_MIN_PUC = 0
     Then start timer T_IGA_MIN_PUC incrementation
IF   T_IGA_MIN_PUC < = IP_T_IGA_MIN_PUC
     then   IGA_MIN_BAS_PUC_H_RNG = IGA_MIN_BAS_RAW_H_RNG
     else  IGA_MIN_BAS_PUC_H_RNG$_{n}$        = min((IGA_MIN_BAS_PUC_H_RNG$_{n-1}$ +
                                               C_IGA_MIN_PUC_LGRD_H_RNG;
                                                          IGA_BAS_COR)
endif
```

```
IF    LV_PUC == 0
Then        IGA_MIN_BAS_H_RNG = IGA_MIN_BAS_RAW_H_RNG
else        IGA_MIN_BAS_H_RNG = IGA_MIN_BAS_PUC_H_RNG
```

endif

Maximum possible spark retard: every 10ms

The two following calculations are always performed (in homogenous mode) at an update rate of 10ms.

IGA_DIF_MIN_0_*H_RNG* = -1* IGA_REF_COR + IGA_MIN_BAS_OPM_H_RNG+ IGA_DIF_MIN_AST_H_RNG*after start correction*

IGA_DIF_MIN_0_H_RNG is just the standard result of the interpolation maps
IGA_DIF_MIN_H_RNG = (IGA_DIF_MIN_0_*H_RNG* + IGA_DIF_MIN_DELTA_H_RNG)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DELTA_TEMP_DIS	1	0...FFH	0...2.55	0.01	[s]
Disable time of extra spark retard on IGA_DIF_MIN after end of gradient return					
C_IGA_DIF_MIN_DELTA_LGRD_H_RNG	1	0...B40H	0...-180	-0.0625	[°CRK]
Step gradient limitation after extra spark retard on IGA_DIF_MIN					
C_IGA_DIF_MIN_DELTA_TEMP_DIS	1	0...FFH	0...2.55	0.01	[s]
Disable time of extra spark retard on IGA_DIF_MIN after end of gradient return					
C_IGA_DIF_MIN_DELTA_TEMP_ENA	1	0...FFH	0...2.55	0.01	[s]
Maintaining time of extra retard on IGA_DIF_MIN					
IP_IGA_DIF_MIN_TCO	8	0...FFH	-48...47.625	0.375	[°CRK]
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	[°C]
TCO correction for minimum ignition angle					
IP_IGA_DIF_MIN_CYC_MT	8*8	0...FFH	-48...47.625	0.375	[°CRK]
LDPM_CYC_CAST_IP_IGA_DIF_MIN	8	0...FFFFH	0...65535	1	[-]
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	[°C]
Correction on minimum ignition angle for combustion cycles after start for manual tranmission					
IP_IGA_DIF_MIN_DELTA_H_RNG	8*6	0...B40H	0...-180	-0.0625	[°CRK]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]
Extra spark retard on minimum ignition angle in case of torque intervention					
IP_IGA_DIF_MIN_CYC_AT	8*8	0...FFH	-48...47.625	0.375	[°CRK]
LDPM_CYC_CAST_IP_IGA_DIF_MIN	8	0...FFFFH	0...65535	1	[-]
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	[°C]
Correction on minimum ignition angle for combustion cycles after start for automatic transmission					
IP_IGA_DIF_MIN_CYC_AMP_MT_H_RNG	6*8	0...B40H	-90...90	0.0625	[°CRK]
LDP_AMP_IP_IGA_DIF_MIN_CYC_AMP	6	0...FFFFH	0...5434	0.0829175	[hPa]
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	[°C]
Altitude correction on minimum ignition angle for after start for manual transmission					
IP_IGA_DIF_MIN_CYC_AMP_AT_H_RNG	6*8	0...B40H	-90...90	0.0625	[°CRK]
LDP_AMP_IP_IGA_DIF_MIN_CYC_AMP	6	0...FFFFH	0...5434	0.0829175	[hPa]
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	[°C]
Altitude correction on minimum ignition angle for after start for automatic transmission					
IP_IGA_MIN_N_MAF_H_RNG	8*6	0...B40H	-90...90	0.0625	[°CRK]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]
Basic minimum ignition angle difference value					
IP_IGA_MIN_N_MAF_PU_H_RNG	8*6	0...B40H	-90...90	0.0625	[°CRK]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]
Basic minimum ignition angle difference value					
IP_IGA_MIN_N_MAF_EXT_H_RNG	8*6	0...B40H	-90...90	0.0625	[°CRK]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]
Basic minimum ignition angle difference value					
IP_IGA_DIF_MIN_TCO_EXT_H_RNG	8	0...B40H	-90...90	0.0625	[°CRK]
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	[°C]
TCO correction for minimum ignition angle					
IP_IGA_DIF_MIN_TCO_PU_H_RNG	8	0...B40H	-90...90	0.0625	[°CRK]
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	[°C]
TCO correction for minimum ignition angle					
IP_IGA_DIF_MIN_TCO_H_RNG	8	0...B40H	-90...90	0.0625	[°CRK]
LDPM_TCO_1_IGSP	8	0...FEH	-48...142.5	0.75	[°C]
TCO correction for minimum ignition angle					
IP_FAC_IGA_MIN_TCO	8*6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]
scaling TCO correction for minimum ignition angle					
IP_FAC_IGA_MIN_TCO_PU	8*6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]

scaling TCO correction for minimum ignition angle					
IP_FAC_IGA_MIN_TCO_EXT	8*6	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]
scaling TCO correction for minimum ignition angle					
IP_FAC_IGA_DIF_AST	8	0...FFH	0...1.99218	0.0078125	[-]
LDPM_N_DIF_IGSP	8	8000...7FFFH	-32768...32767	1	[rpm]
scaling N_DIF correction for minimum ignition angle ast					
C_IGA_DIF_MIN_BAS_LGRD_H_RNG	1	0...B40H	0...180	0.0625	[°CRK]
Step gradient limitation					
C_IGA_MIN_PUC_LGRD_H_RNG	1	0...B40H	0...180	0.0625	[°CRK]
Step gradient limitation					
IP_T_IGA_MIN_PUC	8	0...FFH	0...2.55	0.01	[s]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
timer inside puc					
IP_FAC_IGA_DIF_MIN_AMP	8	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_CYC_CAST_IP_IGA_DIF_MIN	8	0...FFFFH	0...65535	1	[-]
scaling AMPcorrection for minimum ignition angle AST					
IP_IGA_MIN_CH_COR_H_RNG	8*6	0...B40H	-90...90	0.0625	[°CRK]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]
Offset on basic minimum ignition angle caused by cat heating					
IP_IGA_MIN_OPM_COR_H_RNG	8*6	0...B40H	-90...90	0.0625	[°CRK]
LDPM_N_32_2_IGSP	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_HB_2_IGSP	6	0...FFH	0...1389	5.4470588	[mg/stk]
Offset on basic minimum ignition angle caused by engine throttled mode					
LC_INH_IGA_MIN_PUC	1	0...1H	0...1	1	[-]
Logical constant for inhibiting ramping of IGA_MIN_BAS_PUC_H_RNG to IGA_BAS_COR in case of LV_PUC=1					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_USE_IGA_MIN_AST	1	0...1H	0...1	1	[-]
Compiler switch for specific minimum IGA after start phase enabled					

39.4 Customer adaptation module: AGGR IGSP

39.4.1 Outputs for BMW functions which are defined as IGSP exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Zw_ks	V/O	FE0C...258H	-50...60	0.1	[°CRK]
Grundzündwinkel bei KR-Sicherheitsspätverstellung					
Zw_min	V/O	FE0C...258H	-50...60	0.1	[°CRK]
minimum ignition angle					
Zw_ve[NC_CYL_NR]	V/O	FE0C...258H	-50...60	0.1	[°CRK]
basic ignition angle, application correction and knock control included					
Zw_md[NC_CYL_NR]	V/O	FE0C...258H	-50...60	0.1	[°CRK]
ignition angle after torque intervention					
Zw_out_mw	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[°CRK]
Ignition angle applied on the respective cylinder (0 to 5)					

Input data:

IGA_IGC_H_RNG[NC_CYL_NR]	IGA_IGC_0_5_H_RNG	IGA[NC_CYL_NR]	
IGA_MIN_H_RNG	IGA_ADJ_MAX_KNK		

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit power latch phase:

Zw_ks = IGA_ADJ_MAX_KNK
Zw_min = 0°crk
Zw_ve[NC_CYL_NR] = IGA[NC_CYL_NR]
Zw_md[NC_CYL_NR] = 0°crk
Zw_out_mw = IGA_IGC_0_5_H_RNG *`/* IGA_IGC_0_5_H_RNG is initialized with C_IGA_INI_H_RNG */`*

Recurrences: segment: Zw_ks, Zw_min, Zw_ve[NC_CYL_NR], Zw_md[NC_CYL_NR], Zw_out_mw,

Activation: every engine state, except power latch phase

Deactivation: at power latch phase

Values at deactivation: 0, except:

Zw_out_mw = IGA_IGC_0_5_H_RNG

Formula section:

Remark: all formulas are valid in a **physical** meaning

Zw_ks = IGA_ADJ_MAX_KNK
Zw_min = IGA_MIN_H_RNG
Zw_ve[NC_CYL_NR] = IGA[NC_CYL_NR]
Zw_md[NC_CYL_NR] = IGA_IGC_H_RNG[NC_CYL_NR]
Zw_out_mw = IGA_IGC_0_5_H_RNG

39.4.2 Outputs for SV aggregates

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_AV_H_RNG_S_1[NC_CYL_NR]	V/O	FA60...5A0H	-90...90	0.0625	[°CRK]
Ignition angle applied on cylinder CYL in stratified combustion					
IGA_BAS_COR	V/O	0...FFH	-35.625...60	0.375	[°CRK]
corrected basic ignition angle					
IGA_BAS_COR_CBK[NC_CBK_EX_NR]	V/O	0...FFH	-35.625...60	0.375	[°CRK]
corrected basic ignition angle, bank selective					
LV_IGA_MIN_EXT	V/O	0...1H	0...1	1	[-]
Flag to enable external IGA_MIN					
LV_IGA_ST_OPM_SEL	V/O	0...1H	0...1	1	[-]
condition for IGA_ST operation mode, 1 = OPM_1, 0 = OPM_2					
IGA_ADD_CMB_CTL[NC_CYL_NR]	V/O	80...7FH	-48...47.625	0.375	[°CRK]
ignition angle correction due to combustion control function					
IGA_REF_COR	V/O	0...FFH	-35.625...60	0.375	[°CRK]
Corrected reference ignition angle					
IGA_REF_COR_CBK[NC_CBK_EX_NR]	V/O	0...FFH	-35.625...60	0.375	[°CRK]
corrected reference ignition angle bank selective					
IGA_REF_EGR_HOM_COR	O/V	80...7FH	-48...47.625	0.375	[°CRK]
additive reference ignition angle correction due to EGR influence in HOM					
IGA_REF_EGR_HOMS_COR	O/V	80...7FH	-48...47.625	0.375	[°CRK]
additive reference ignition angle correction due to EGR influence in HOMS					
IGA_REF_LAMB_COR	V/O	80...7FH	-48...47.625	0.375	[°CRK]
Additive lambda correction of IGA_REF					
IGA_REF_TEMP_COR	V/O	80...7FH	-48...47.625	0.375	[°CRK]
Additive temperature correction of IGA_REF					
IGA_AV_H_RNG_HOMS_1[NC_CYL_NR]	V/O	FA60...5A0H	-90...90	0.0625	[°CRK]
HOMS ignition angle in wide range applied on cylinder x					
FAC_IGA_OPM_SEL	V/O	0...FFH	0...0.99609	3.9063e-3	[-]
Ignition interpolation factor for operation switch manager					

Input data:

Zw_soll_s[NC_CYL_NR]	Zw_opt1	B_gdst	F_zw_hsplit
Zw_opt2	Zw_soll_hs[NC_CYL_NR]	Zw_grund1	Zw_grund2
NC_CBK_EX_NR	NC_CYL_NR	B_zwmin_ext	LV_S_CLC
Dzw_agr_hs	Dzw_agr_kor	Zw_offkorrvr[NC_CYL_NR]	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit power latch phase :

all values 0 phys

Except first calculation after init of BMW-values:

IGA_REF_COR_CBK[NC_CBK_EX_NR], IGA_REF_COR,
IGA_BAS_COR_CBK[NC_CBK_EX_NR], IGA_BAS_COR

Recurrence : segment: IGA_AV_H_RNG_S_1[NC_CYL_NR],
IGA_AV_H_RNG_HOMS_1[NC_CYL_NR],
IGA_ADD_CMB_CTL[NC_CLY_NR]

10ms : FAC_IGA_OPM_SEL, IGA_BAS_COR, IGA_REF_COR, IGA_REF_LAMB_COR, IGA_REF_TEMP_COR, LV_IGA_ST_OPM_SEL IGA_IGC_HOM_S[NC_CYL_NR], IGA_REF_COR_1, IGA_REF_COR_2, IGA_BAS_COR_1, IGA_BAS_COR_2, LV_IGA_MIN_EXT, IGA_REF_EGR_HOM_COR, IGA_REF_EGR_HOMS_COR

Activation: at every engine state

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
If    LV_S_CLC = 1
then
        IGA_AV_H_RNG_S_1[NC_CYL_NR] = Zw_soll_s[NC_CYL_NR]
        IGA_AV_H_RNG_HOMS_1 [NC_CYL_NR] = Zw_soll_hs[NC_CYL_NR]
        IGA_REF_EGR_HOMS_COR =  Dzw_agr_hs
endif
FAC_IGA_OPM_SEL       = F_zw_hsplit
IGA_BAS_COR_CBK[1]    = Zw_grund1
IGA_BAS_COR_CBK[2]    = Zw_grund2
IGA_BAS_COR           = 0.5 * ( IGA_BAS_COR_CBK[1] + IGA_BAS_COR_CBK[2] )
LV_IGA_ST_OPM_SEL=          B_gdst
IGA_ADD_CMB_CTL[NC_CLY_NR]  =  Zw_offkorrvr[NC_CYL_NR]
IGA_REF_COR_CBK[1] = Zw_opt1
IGA_REF_COR_CBK[2] = Zw_opt2
IGA_REF_COR           = 0.5 * ( IGA_REF_COR_CBK[1] +   IGA_REF_COR_CBK[2] )
IGA_REF_EGR_HOM_COR =  Dzw_agr_kor
IGA_REF_LAMB_COR                    = Dzwo_lam_kor
IGA_REF_TEMP_COR                    = Dzwot
LV_IGA_MIN_EXT                 = B_zwmin_ext
```

40 Immobilizer

Table of Contents

Index

40.1 Immobilizer signal diagnosis (EWS4)

General information:

IMOB diagnosis is performed to provide error-symptom-information by setting of LV_ERR_IMOB_i and entry in error memory in case of ECU-blocking/error detection.

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_ERR_IMOB_0	V/O	0...1H	0...1	1	[-]
Present failure in IMOB manipulation after filtering					
LV_END_DIAG_IMOB_0	V/O	0...1H	0...1	1	[-]
Diagnostic performed at least one time					
LV_CDN_DIAG_IMOB_0	V/O	0...1H	0...1	1	[-]
Diagnostic condition to start symptom detection (set to 1 when condition is fulfilled)					
ERR_SYM_IMOB_0	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
For each symptom : status of failure (set to 1 when failure symptom detected)					
LV_ERR_IMOB_1	V/O	0...1H	0...1	1	[-]
Present failure in IMOB CAS BUS after filtering					
LV_END_DIAG_IMOB_1	V/O	0...1H	0...1	1	[-]
Diagnostic performed at least one time					
LV_CDN_DIAG_IMOB_1	V/O	0...1H	0...1	1	[-]
Diagnostic condition to start symptom detection (set to 1 when condition is fulfilled)					
ERR_SYM_IMOB_1	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
For each symptom : status of failure (set to 1 when failure symptom detected)					
LV_ERR_IMOB_2	V/O	0...1H	0...1	1	[-]
Present failure in IMOB data storage after filtering					
LV_END_DIAG_IMOB_2	V/O	0...1H	0...1	1	[-]
Diagnostic performed at least one time					
LV_CDN_DIAG_IMOB_2	V/O	0...1H	0...1	1	[-]
Diagnostic condition to start symptom detection (set to 1 when condition is fulfilled)					
ERR_SYM_IMOB_2	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
For each symptom : status of failure (set to 1 when failure symptom detected)					
LV_ERR_IMOB_3	V/O	0...1H	0...1	1	[-]
Present failure in IMOB CAN BUS after filtering					
LV_END_DIAG_IMOB_3	V/O	0...1H	0...1	1	[-]
Diagnostic performed at least one time					
LV_CDN_DIAG_IMOB_3	V/O	0...1H	0...1	1	[-]
Diagnostic condition to start symptom detection (set to 1 when condition is fulfilled)					
ERR_SYM_IMOB_3	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
For each symptom : status of failure (set to 1 when failure symptom detected)					
CDN_DIAG_IMOB_1	V/O	0...FFH	0...255	1	[-]
Condition bit coded for each sym					
CDN_DIAG_IMOB_3	V/O	0...FFH	0...255	1	[-]
Condition bit coded for each sym					

Defined in non-public specification:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit

LV_LOCK_IMOB	V/O	0...1H	0...1	1	[-]
LV indicates locking of IGN and INJ by IMOB					
INH_IV_IMOB	V/O	0...FFH	0...255	1	[-]
Locking request for INJ by IMOB					

Export actions:

ACTION_IMOB_DIAG_ERR_1(CDN_IMOB, SYM_IMOB)
ACTION_IMOB_DIAG_ERR_3(CDN_IMOB, SYM_IMOB)

Input data:

LV_IGK	STATE_IMOB_0_ERR	STATE_IMOB_2_ERR	LV_VAR_BN
STATE_CAN_DNM_D	PWL_LOCK_CDN		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_IMOB_0	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_IMOB_0	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					
C_ABC_DEC_IMOB_0	1	0...FFH	0...255	1	[-]
Antibounce counter decrement					
C_ABC_INC_IMOB_1	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_IMOB_1	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					
C_ABC_INC_IMOB_2	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_IMOB_2	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					
C_ABC_INC_IMOB_3	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_IMOB_3	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					
C_ABC_DEC_IMOB_1	1	0...FFH	0...255	1	[-]
Antibounce counter decrement					
C_ABC_DEC_IMOB_3	1	0...FFH	0...255	1	[-]
Antibounce counter decrement					

FUNCTION DESCRIPTION:

This diagnosis calculates the error LV_ERR_IMOB_0

Imob: Manipulation

Application conditions:

Initialisation:

```
LV_CDN_DIAG_IMOB_0  = STD_INI
LV_END_DIAG_IMOB_0  = STD_INI
LV_ERR_IMOB_0       = STD_INI
ERR_SYM_IMOB_0      = STD_INI
```

when LV_IGK changes from 0→1 or at reset:

Recurrence: 100ms

Activation/ Deactivation:

```
IF      LV_IGK = 1 then
THEN    LV_CDN_DIAG_IMOB_0 = 1
ELSE    LV_CDN_DIAG_IMOB_0 = 0
ENDIF
```

Formula section:

Symptoms calculation:

Error state of STATE_IMOB_0_ERR equal to ERR_SYM

Error set after debounce.

End of diag calculation see ERRM

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Configuration for diagnostic symptoms:

Diagnostic IMOB_0	Symptom description	Symptom	Filter type
manipulation	not used	*SYM_0*	STD_INI
	no secrete key	*SYM_1*	
	not used	*SYM_2*	
	response does not match	*SYM_3*	

FUNCTION DESCRIPTION:

This diagnosis calculates the error LV_ERR_IMOB_2

Imob: Data storage

Application conditions:

Initialization:

```
LV_CDN_DIAG_IMOB_2  = STD_INI
LV_END_DIAG_IMOB_2  = STD_INI
LV_ERR_IMOB_2       = STD_INI
ERR_SYM_IMOB_2      = STD_INI
```

when LV_IGK changes from 0→1 or at reset:

Recurrence: 100ms

Activation/ Deactivation:

```
IF      LV_IGK = 1 then
THEN    LV_CDN_DIAG_IMOB_2 = 1
ELSE    LV_CDN_DIAG_IMOB_2 = 0
ENDIF
```

Formula section:

Symptoms calculation:

Error state of STATE_IMOB_2_ERR equal to ERR_SYM

Error set after debounce.

End of diag calculation see ERRM

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Configuration for diagnostic symptoms:

Diagnostic IMOB_2	Symptom description	Symptom	Filter type
Data storage	No storage for secrete key avl.	*SYM_0*	STD_INI
	read/write FSC (nvm error)	*SYM_1*	
	Read error secrete key (2of3)	*SYM_2*	
	Spare for checksum error (later)	*SYM_3*	

Exported Interface: ACTION_IMOB_DIAG_ERR_1(CDN_IMOB, SYM_IMOB)

FUNCTION DESCRIPTION:

This diagnosis calculates the error LV_ERR_IMOB_1

Imob: KBUS Diagnosis

Note: Diagnosis is **not** initialized at KL15 0->1 because diagnosis is already active at ecu_state = wakeup

Application conditions:

Initialization:

all at 0 only at reset:
LV_CDN_DIAG_IMOB_1 = STD = 0
LV_END_DIAG_IMOB_1 = STD = 0
LV_ERR_IMOB_1 = STD = 0
ERR_SYM_IMOB_1 = STD = 0

Activation/ Deactivation:

LV_VAR_BN = 0 **or** (STATE_CAN_DNM_D > 0 and PWL_LOCK_CDN[BN2000] = 1)

Recurrence: at call ACTION_IMOB_DIAG_ERR_1(CDN_IMOB, SYM_IMOB)

Symptoms calculation:

Error state of ERRM: ERR_SYM equal to SYM_IMOB

```
If      SYM_IMOB = END_DIAG
Then    LV_END_DIAG_IMOB_1 = 1 (set immediatly)
Else
    If   SYM_IMOB > 0
    Then     LV_CDN_DIAG_IMOB_1 = 1
         ERR_SYM_IMOB_1= SYM_IMOB
         LV_ERR_IMOB_1 = 1                       (after debounce)
    Else     If    (CDN_IMOB & ERR_SYM_IMOB_1) = true        or
                   ERR_SYM_IMOB_1 = NO_SYM
             Then  LV_CDN_DIAG_IMOB_1 = 1
                   healing only of cdn fits to err_sym once
                   ERR_SYM_IMOB_1= NO_SYM
                   LV_ERR_IMOB_1 = 0                 (after debounce)
             Else  LV_CDN_DIAG_IMOB_1 = 0      (freeze healing)
             Endif
    Endif
```

Endif

Error state of ERRM: ERR_SYM = SYM_IMOB

CDN_DIAG_IMOB_1 = CDN_IMOB

Error set after debounce.

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Configuration for diagnostic symptoms:

Diagnostic IMOB_1	Symptom description	Symptom	Filter type
Kbus	Open line	*SYM_0*	STD
	Frame error (type_ews/st_ews)	*SYM_1*	
	Time out data	*SYM_2*	
	CRC error	*SYM_3*	

Exported Interface: ACTION_IMOB_DIAG_ERR_3(CDN_IMOB, SYM_IMOB)

FUNCTION DESCRIPTION:

This diagnosis calculates the error LV_ERR_IMOB_3

Imob: CAN Diagnosis

Note: Diagnosis is **not** initialized at KL15 0->1 because diagnosis is already active at ecu_state = wakeup

Application conditions:

Initialisation:

all at 0 only at reset:
LV_CDN_DIAG_IMOB_3 = STD = 0
LV_END_DIAG_IMOB_3 = STD = 0
LV_ERR_IMOB_3 = STD = 0
ERR_SYM_IMOB_3 = STD = 0

Activation/ Deactivation:

LV_VAR_BN = 0 **or** (STATE_CAN_DNM_D > 0 and PWL_LOCK_CDN[BN2000] = 1)

Recurrence: at call ACTION_IMOB_DIAG_ERR_3(CDN_IMOB, SYM_IMOB)

Formula section:

Symptoms calculation:

Error state of ERRM: ERR_SYM equal to SYM_IMOB

```
If      SYM_IMOB = END_DIAG
Then    LV_END_DIAG_IMOB_3 = 1 (set immediatly)
Else
        If    SYM_IMOB > 0
        Then        LV_CDN_DIAG_IMOB_3 = 1
              ERR_SYM_IMOB_3 = SYM_IMOB
              LV_ERR_IMOB_3 = 1                          (after debounce)
        Else        If     (CDN_IMOB & ERR_SYM_IMOB_3) = true          or
                           ERR_SYM_IMOB_3 = NO_SYM
                    Then   LV_CDN_DIAG_IMOB_3 = 1
                           healing only of cdn fits to err_sym once
                           ERR_SYM_IMOB_3= NO_SYM
                           LV_ERR_IMOB_3 = 0                 (after debounce)
                    Else   LV_CDN_DIAG_IMOB_3 = 0        (freeze healing)
                    Endif
        Endif
```

Endif

Error state of ERRM: ERR_SYM = SYM_IMOB

CDN_DIAG_IMOB_3 = CDN_IMOB

Error set after debounce.

End of diag calculation:

Symptoms are **unchanged** if condition for doing symptom detection is not fulfilled.

Configuration for diagnostic symptoms:

Diagnostic IMOB_3	Symptom description	Symptom	Filter type
CAN	Not used	*SYM_0*	STD
	Frame error (type_ews/st_ews)	*SYM_1*	
	Time out data	*SYM_2*	
	Not used	*SYM_3*	

41 Injection realisation

Chapter Injection realisation	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 6350 of 9643

Index

INJR Configuration Data

The following document describes the general rules for definition of the configuration data for INJR Aggregate.

General information:

The following document describes the general rules for definition of the configuration data for INJR Aggregate

41.1.1 Global configuration data

Here are listed the configuration data, which are not only used in the modules of the INJR aggregate.

Data	Value
NC_INJ_CONF	3
NC_INJ_CONF_GDI	3
NC_NR_SYM_IV	0

41.1.2 Local configuration data

The configuration data are listed here which are used only in the INJR aggregate.

Data	Value
NC_INJ_INH_SWI_IV_SHIFT_NR	6
NC_CRK_INJ_REF_TDC	-360 °CRK
NC_CRK_INJ_BAS_REF	180 °CRK
NC_IDX_CYL_PBK_IV_REF[NC_CYL_NR]	[0 1 0 1 0 1]
NC_NR_IV_PLS	3
NC_NR_PLS_INJ_PAC_0	3
NC_NR_PLS_INJ_PAC_1	1
NC_NR_PLS_INJ_PAC_2	0
NC_NR_PLS_INJ_PAC_3	0
NC_NR_PLS_INJ_PAC_4	0
NC_NR_PLS_INJ_PAC_5	0
NC_NR_PLS_INJ_PAC_6	0
NC_NR_PLS_INJ_PAC_7	0
NC_PBK_IV_NR	2
NC_NR_DCDC_INJ	1
NC_USE_TI_AS_CBK	0
NC_N_MAX_IV_TEST_MOD	0 rpm
NC_CTR_MAX_IV_TEST_MOD	5
NC_TI_TEST_PLS_PER	1ms
NC_IV_CRASH	0x3 (00000011 binary)
NC_TI_CRASH	1000 ms
NC_CRK_INJ_ANG_IMDT	720 °CRK
NC_T_MIN_OFS_INJ_CYL	0.400 ms
NC_T_MIN_OFS_INJ_PLS	0.240 ms
NC_T_MES_DLY_BEG	6 µs
NC_T_MES_DLY_END	12 µs
NC_CRK_INJ_RNG	240 °CRK
NC_CHA_DIAG_IV_MIN	300 µAs
NC_CHA_DIAG_IV_MAX	1200 µAs
NC_V_DIAG_IV_MIN	50 V
NC_V_DIAG_IV_MAX	210 V
NC_V_DIAG_IV_MIN_SCG	4 V

Data	Value
NC_V_DIAG_IV_MIN_SCB	16 V
NC_T_DIAG_IV_OCC	100 µs
NC_STATE_STST_ENA	0
NC_IDX_CYL_HPP_REF	[0 0 0 0 0 0]
NC_NR_IDX_TI_PRM_H	142
NC_NR_IDX_TI_PRM_L	84
NC_NR_IDX_MFF_INJ	8
NC_NR_IDX_T_DLY_INJ	8
NC_NR_IDX_FUP_INJ	6
NC_NR_IDX_TEMP_INJ	5
NC_NR_IDX_RATIO_T_INJ	12
NC_NR_IDX_MFF_2_INJ	16
NC_NR_IDX_PRS_DEC_INJ	12
NC_T_DCHA_DLY_MIN	0.045 ms

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_INJ_CONF_GDI	1	0H 1H 3H	MPI HPDI PIEZO	1	[-]
Injection mode for gasoline direct injection					
NC_INJ_CONF	1	0H 1H 3H	MPI HPDI PIEZO	1	[-]
Injection Mode					
NC_INJ_INH_SWI_IV_SHIFT_NR	1	3...10H	3...16	1	[-]
Constant which defines how many bits are used for the shut off sequence					
NC_CRK_INJ_REF_TDC	1	FC40...0H	-360...0	0.375	[°CRK]
(NC_CRK_INJ_REF_TDC + NC_CRK_INJ_BAS_REF) defines the zero position of the co-ordinate system of the injection system with reference to the ignition TDC of logical cylinder 0.					
NC_CRK_INJ_BAS_REF	1	0...780H	0...720	0.375	[°CRK]
End of phasing base reference angle related to NC_CRK_INJ_REF_TDC.					
NC_NR_IV_PLS	1	0...FH	1...16	1	[-]
Number of pulses per cylinder and working cycle (720°CRK)					
NC_NR_PLS_INJ_PAC_0	1	0...10H	0...16	1	[-]
Number of injection pulses combined in packet number 0					
NC_NR_PLS_INJ_PAC_1	1	0...10H	0...16	1	[-]
Number of injection pulses combined in packet number 1					
NC_NR_PLS_INJ_PAC_2	1	0...10H	0...16	1	[-]
Number of injection pulses combined in packet number 2					
NC_NR_PLS_INJ_PAC_3	1	0...10H	0...16	1	[-]
Number of injection pulses combined in packet number 3					
NC_NR_PLS_INJ_PAC_4	1	0...10H	0...16	1	[-]
Number of injection pulses combined in packet number 4					
NC_NR_PLS_INJ_PAC_5	1	0...10H	0...16	1	[-]
Number of injection pulses combined in packet number 5					
NC_NR_PLS_INJ_PAC_6	1	0...10H	0...16	1	[-]
Number of injection pulses combined in packet number 6					
NC_NR_PLS_INJ_PAC_7	1	0...10H	0...16	1	[-]
Number of injection pulses combined in packet number 7					
NC_PBK_IV_NR	1	1...8H	1...8	1	[-]
Number of power stage banks for the injection valves					
NC_NR_DCDC_INJ	1	1...8H	1...8	1	[-]
Number of DC/DC converters for the injection power stages					
NC_NR_SYM_IV	1	1...FH	1...15	1	[-]
Available symptoms for injection valve diagnostic					
NC_USE_TI_AS_CBK	1	0...1H	0...1	1	[-]
Compiler switch to enable external TI adjustment caused by application system					
NC_N_MAX_IV_TEST_MOD	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed up to an actuator test is possible					
NC_CTR_MAX_IV_TEST_MOD	1	0...FFH	0...255	1	[-]
Max. number of actuator test pulses before an engine run event has to be occurred to start the functionality again					
NC_TI_TEST_PLS_PER	1	0...FFFFH	0...65.535	0.001	[ms]
Injection valve open time for actuator test					
NC_IV_CRASH	1	0...FFH	0...255	1	[-]
Injection valve opening pattern					
NC_TI_CRASH	1	0...FFFFH	0...655350	10	[ms]
Injection time during vehicle crash condition is true					
NC_CRK_INJ_ANG_IMDT	1	0...780H	0...720	0.375	[°CRK]
Injection angle to perform an injection pulse immediate					
NC_T_MIN_OFS_INJ_CYL	1	0...FFFFH	0...65.535	0.001	[ms]
Minimum off time between two injection on corresponding cylinders					
NC_T_MIN_OFS_INJ_PLS	1	0...FFFFH	0...65.535	0.001	[ms]

Minimum off time between two injection pulses					
NC_T_MES_DLY_BEG	1	0...11H	1...18	1	[µs]
Time delay at begin of charge/discharge phase					
NC_T_MES_DLY_END	1	0...11H	1...18	1	[µs]
Time delay at end of charge/discharge phase					
NC_CRK_INJ_RNG	1	0...780H	0...720	0.375	[°CRK]
Definition of maximum injection range, depending on power stages configuration for injection valves.					
NC_CHA_DIAG_IV_MIN	1	0...3FFH	0...2272.6968	2.2216	[µAs]
ATIC63 diagnosis threshold: minimum valid charge					
NC_CHA_DIAG_IV_MAX	1	0...3FFH	0...2272.6968	2.2216	[µAs]
ATIC63 diagnosis threshold: maximum valid charge					
NC_V_DIAG_IV_MIN	1	0...7FFFH	0...639.98046	0.0195313	[V]
ATIC63 diagnosis threshold: minimum voltage at opened injector					
NC_V_DIAG_IV_MAX	1	0...7FFFH	0...639.98046	0.0195313	[V]
ATIC63 diagnosis threshold: maximum voltage at opened injector					
NC_V_DIAG_IV_MIN_SCG	1	0...7FFFH	0...639.98046	0.0195313	[V]
ATIC63 diagnosis threshold: minimum injector voltage for diagnosis of a short circuit to ground error					
NC_V_DIAG_IV_MIN_SCB	1	0...7FFFH	0...639.98046	0.0195313	[V]
ATIC63 diagnosis threshold: minimum injector voltage for diagnosis of a short circuit to battery error					
NC_T_DIAG_IV_OCC	1	0...FFFFH	0...65.535	0.001	[ms]
ATIC63 diagnosis threshold: Maximum time till the injector discharge is started and the negative current is detected, after EOI was reached.					
NC_IDX_CYL_PBK_IV_REF[NC_CYL_NR]	1	0...7H	0...7	1	[-]
Array to assign each injector (cylinder) to the corresponding injector power bank					
NC_STATE_STST_ENA	1	0...1H	0...1	1	[-]
Switch to indicate engine stop start automatic enabled (= 1) or disabled (= 0)					
NC_IDX_CYL_HPP_REF[NC_CYL_NR]	1	0...7H	0...7	1	[-]
Array to assign each injector (cylinder) to the corresponding high pressure bank					
NC_NR_IDX_TI_PRM_H	1	0...FFH	0...255	1	[-]
Number of individually calibratable injection timing parameters for high injector needle lift.					
NC_NR_IDX_TI_PRM_L	1	0...FFH	0...255	1	[-]
Number of individually calibratable injection timing parameters for low injector needle lift.					
NC_NR_IDX_MFF_INJ	1	0...FFH	0...255	1	[-]
Number of axis points for MFF axis of fuel rail pressure pulsation correction function.					
NC_NR_IDX_T_DLY_INJ	1	0...FFH	0...255	1	[-]
Number of axis points for pulse delay time axis of fuel rail pressure pulsation correction function.					
NC_NR_IDX_FUP_INJ	1	0...FFH	0...255	1	[-]
Number of axis points for FUP axis of fuel rail pressure pulsation correction function.					
NC_NR_IDX_TEMP_INJ	1	0...FFH	0...255	1	[-]
Number of axis points for TEMP axis of fuel rail pressure pulsation correction function.					
NC_NR_IDX_RATIO_T_INJ	1	0...FFH	0...255	1	[-]
Number of axis points for time ratio axis of fuel rail pressure pulsation correction function.					
NC_NR_IDX_MFF_2_INJ	1	0...FFH	0...255	1	[-]
Number of axis points for MFF axis of MFF TI map.					
NC_NR_IDX_PRS_DEC_INJ	1	0...FFH	0...255	1	[-]
Number of axis points for pressure difference axis of MFF TI map.					
NC_T_DCHA_DLY_MIN	1	0...FFFFH	0...65.535	0.001	[ms]
Minimum applicable time for injector discharge delay that is greater than zero.					

41.2 INJR - Requirements to Infrastructure

Only SV internal

41.3 Final Injection Timing

Overview:

Calculation of the cylinder individual injection times for homogeneous mode:

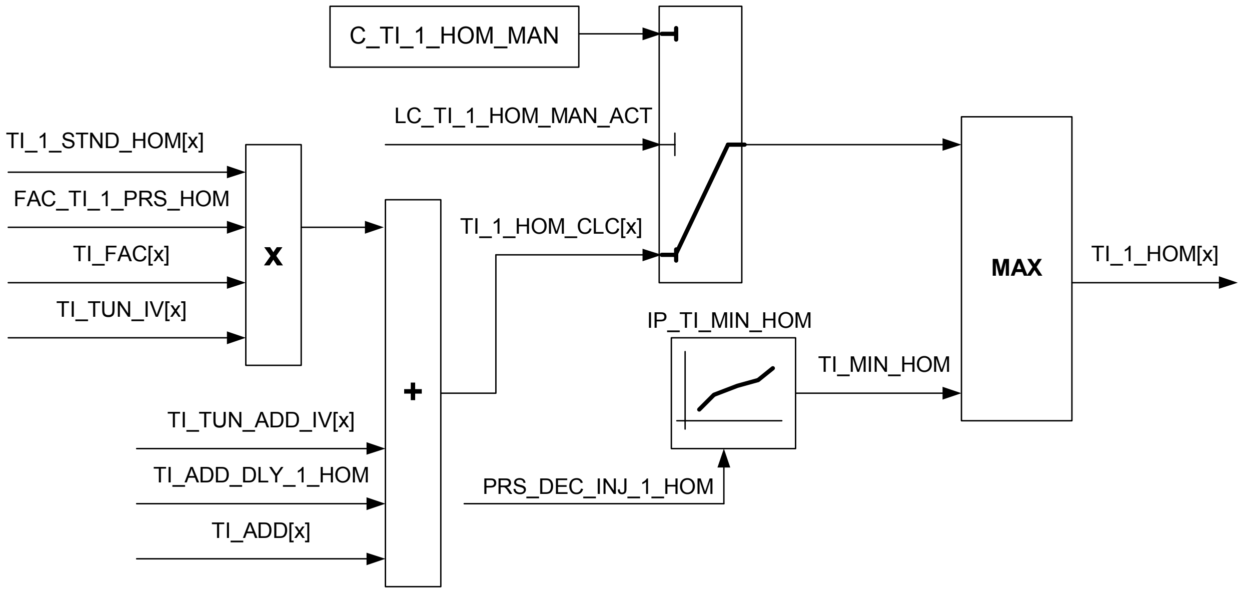
First pulse:

This module calculates the cylinder individual injection times TI_1_HOM[x], which are applied at the injectors.

The injection time can be set manually. If the switch LC_TI_1_HOM_MAN_ACT is set to one, the constant C_TI_1_HOM_MAN is applied.

If homogeneous lean mode is active (STATE_CMB_CTL = HOM_AFL) and TI_1_HOM[x] is smaller than (TI_MIN + C_TI_MIN_OFS), the combustion mode changes to homogeneous mode. This is indicated if the flag LV_AUTH_TI_MIN_AFL is set to zero. Switching back to homogeneous lean mode is allowed at the earliest if the delay time C_T_AUTH_TI_MIN_AFL has elapsed. Then the flag LV_AUTH_TI_MIN_AFL is set back to zero.

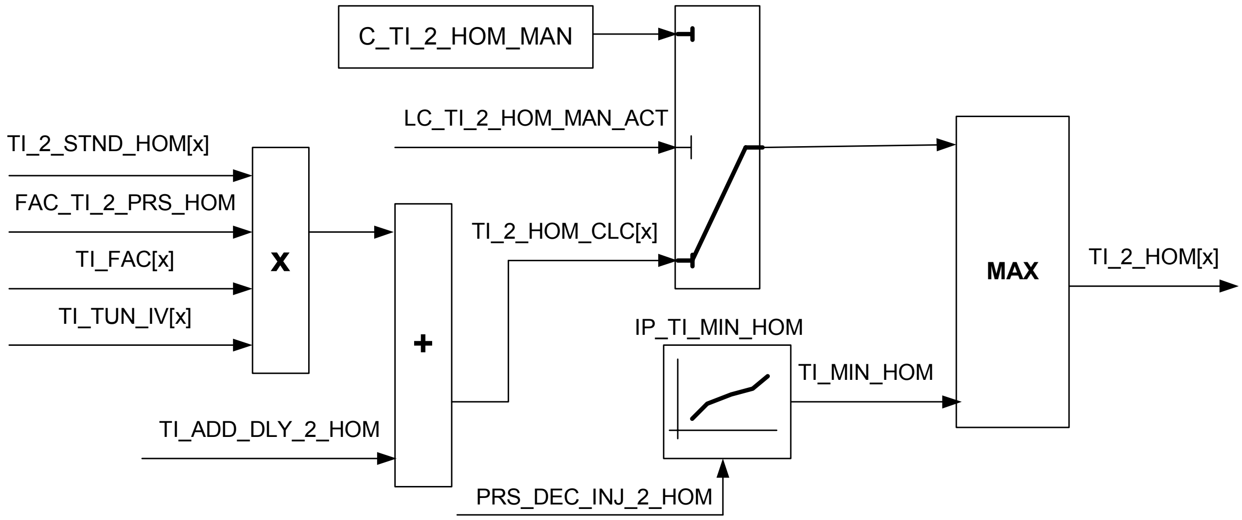
Overview for calculation of the first pulse in homogenous mode:



Second pulse:

If double injection is active (STATE_INJ_MOD_HOM_REQ = MPLH), then TI_2_HOM[x] is calculated. In order to avoid linearity problems, the injection time is limited to TI_MIN_HOM. This is indicated by the flag LV_TI_2_HOM_MIN = 1.

Overview for calculation of the second pulse in homogenous mode:

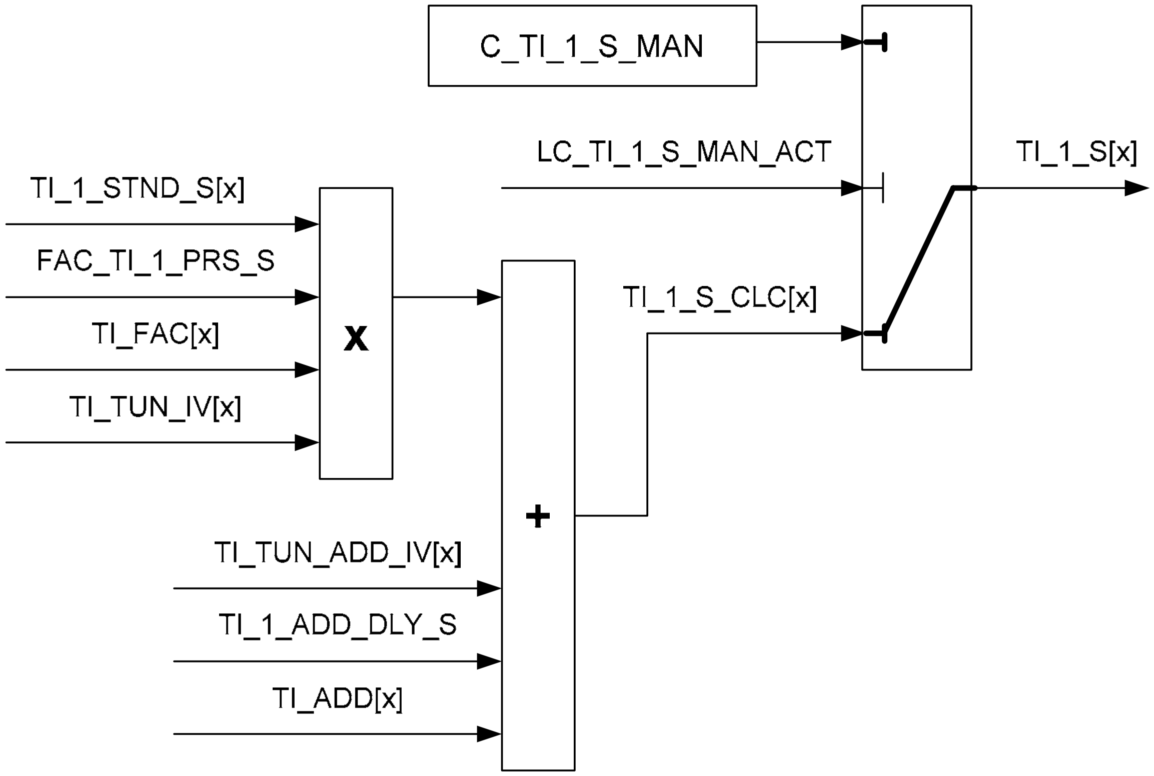


Calculation of the cylinder individual injection times for stratified mode:

First pulse:
The injection time TI_1_S[x] can be set manually. If the switch LC_TI_1_S_MAN_ACT is set to one, the constant C_TI_1_S_MAN is applied.

Then the combustion mode changes to homogeneous mode (indicated by LV_AUTH_TI_MIN_S = 0). The switch back to stratified mode is possible if TI_1_S[x] bigger than (TI_MIN + C_TI_MIN_OFS + C_TI_MIN_HYS_S).

Overview for calculation of the first pulse in stratified mode:

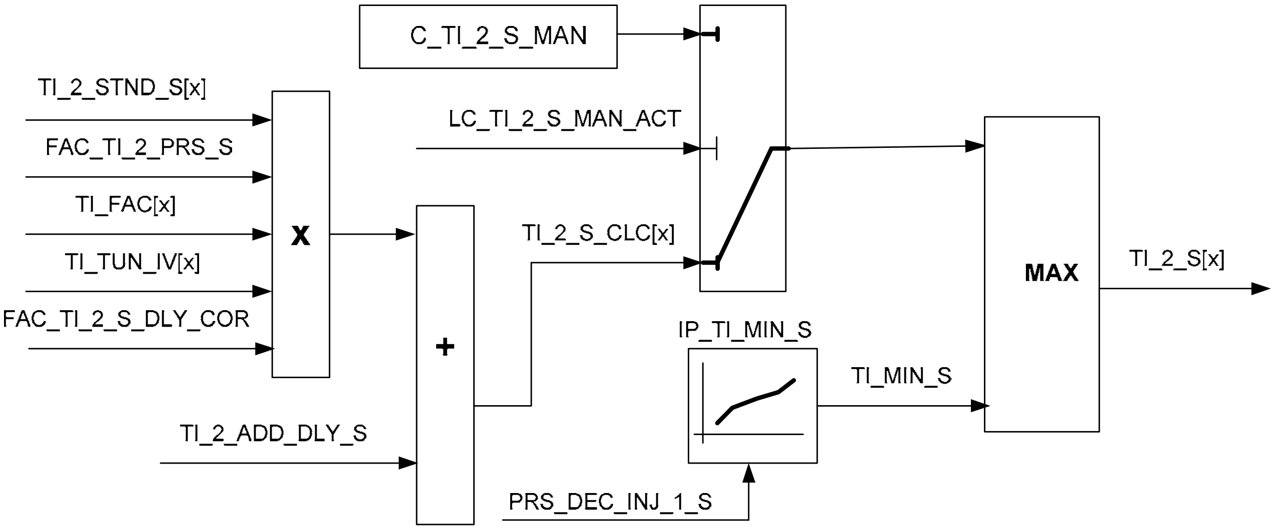


Second pulse:

If double injection is active (STATE_INJ_MOD_S_REQ = MPLS), then TI_2_S[x] is calculated.

The injection time TI_2_S[x] can be set manually. If the switch LC_TI_2_S_MAN_ACT is set to one, the constant C_TI_2_S_MAN is applied.

For stratified multi injection mode the TI_ADD[x] is divided up proportionately.



Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CRK_DIF_SOI_IGN	O	0...F00H	0...1440	0.375	[°CRK]
Output variable for LACO					
FAC_PRS_COR_DAMP_1_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to delay time and fuel pressure; first pulse homogeneous mode					
FAC_PRS_COR_DAMP_2_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to delay time and fuel pressure; second pulse homogeneous mode					
FAC_PRS_COR_DAMP_3_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to delay time and fuel pressure; third pulse homogeneous mode					
FAC_PRS_COR_MFF_1_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to previous injected fuel mass; first pulse homogeneous mode					
FAC_PRS_COR_MFF_2_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to previous injected fuel mass; second pulse homogeneous mode					
FAC_PRS_COR_MFF_3_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to previous injected fuel mass; third pulse homogeneous mode					
FAC_PRS_COR_OSC_1_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to delay time and injection time; first pulse homogeneous mode					
FAC_PRS_COR_OSC_2_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to delay time and injection time; second pulse homogeneous mode					
FAC_PRS_COR_OSC_3_HOM	V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
Pressure pulsation correction factor due to delay time and injection time; third pulse homogeneous mode					
FAC_TI_2_S_COR[NC_CBK_HPP_NR]	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Correction factor for mass flow difference at double injection, caused by duration of the second pulse.					
FAC_TI_2_S_DLY_COR[NC_CBK_HPP_NR]	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Correction factor for mass flow difference at double injection					
FAC_TI_3_S_COR[NC_CBK_HPP_NR]	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Correction factor for mass flow difference at multiple injection, caused by duration of the third pulse.					
FAC_TI_3_S_DLY_COR[NC_CBK_HPP_NR]	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Correction factor for mass flow difference at multiple injection; 3. pulse strat. mode					
FAC_TI_PRS_COR_1	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Pressure pulsation correction factor for first pulse					
FAC_TI_PRS_COR_1_HOM[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Pressure pulsation correction factor for first pulse homogeneous mode					
FAC_TI_PRS_COR_2	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Pressure pulsation correction factor for the second pulse					
FAC_TI_PRS_COR_2_HOM[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Pressure pulsation correction factor for second pulse homogeneous mode					
FAC_TI_PRS_COR_2_S[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Pressure pulsation correction factor for second pulse stratified mode					
FAC_TI_PRS_COR_3	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Pressure pulsation correction factor for the third pulse					
FAC_TI_PRS_COR_3_HOM[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Pressure pulsation correction factor for third pulse homogeneous mode					
FAC_TI_PRS_COR_3_S[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Pressure pulsation correction factor for third pulse stratified mode					
IDX_TI_1_HOM_CLC[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for first pulse HOM					
IDX_TI_1_HOM_MIN	O/V	0...FFFFH	0...65535	1	[-]
Minimum couting factor index for first pulse HOM.					
IDX_TI_1_S_CLC[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for first pulse S					

IDX_TI_1_S_MIN	O/V	0...FFFFH	0...65535	1	[-]
Minimum couting factor index for first pulse S.					
IDX_TI_1_STND_HOM[NC_CYL_NR]	V	0...FFFFH	0...65535	1	[-]
Couting factor index for first pulse HOM ¿ intermediate value					
IDX_TI_2_HOM_CLC[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for second pulse HOM					
IDX_TI_2_HOM_MIN	O/V	0...FFFFH	0...65535	1	[-]
Minimum couting factor index for second pulse HOM.					
IDX_TI_2_S_CLC[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for second pulse S					
IDX_TI_2_S_MIN	O/V	0...FFFFH	0...65535	1	[-]
Minimum couting factor index for second pulse S.					
IDX_TI_2_STND_HOM[NC_CYL_NR]	V	0...FFFFH	0...65535	1	[-]
Couting factor index for second pulse HOM ¿ intermediate value					
IDX_TI_3_HOM_CLC[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for third pulse HOM					
IDX_TI_3_HOM_MIN	O/V	0...FFFFH	0...65535	1	[-]
Minimum couting factor index for third pulse HOM					
IDX_TI_3_S_CLC[NC_CYL_NR]	O/V	0...FFFFH	0...65535	1	[-]
Couting factor index for third pulse S					
IDX_TI_3_S_MIN	O/V	0...FFFFH	0...65535	1	[-]
Minimum couting factor index for third pulse S.					
IDX_TI_3_STND_HOM[NC_CYL_NR]	V	0...FFFFH	0...65535	1	[-]
Couting factor index for third pulse HOM ¿ intermediate value					
IDX_TI_TMP	-	0...FFFFH	0...65535	1	[-]
Array index for injection timing selection; temporary value.					
LV_AUTH_TI_MIN_AFL	O/V	0...1H	0...1	1	[-]
Logical variable to authorize homogeneous lean mode if injection time is not to less					
LV_AUTH_TI_MIN_S	O/V	0...1H	0...1	1	[-]
Logical variable to authorize stratified mode if injection time is not to less					
LV_TI_1_HOM_MIN	O/V	0...1H	0...1	1	[-]
Flag to indicate that minimum injection time is reached for homogeneous mode, first pulse					
LV_TI_1_S_MIN	O/V	0...1H	0...1	1	[-]
Flag to indicate that minimum injection time is reached for stratified mode, first pulse					
LV_TI_2_HOM_MIN	O/V	0...1H	0...1	1	[-]
Flag to indicate that minimum injection time is reached for homogeneous mode, second pulse					
LV_TI_2_S_MIN	O/V	0...1H	0...1	1	[-]
Flag to indicate that minimum injection time is reached for stratified mode, second pulse					
LV_TI_3_HOM_MIN	O/V	0...1H	0...1	1	[-]
Flag to indicate that minimum injection time is reached for homogeneous mode, third pulse					
LV_TI_3_S_MIN	O/V	0...1H	0...1	1	[-]
Flag to indicate that minimum injection time is reached for stratified mode, third pulse					
MFF_SP_1_HOM_COR_INJ[NC_CYL_NR]	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual mass fuel flow set-point for homogeneous mode corrected by piezo energy carryover, first pulse					
MFF_SP_1_S_COR_INJ[NC_CYL_NR]	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual mass fuel flow set-point for stratified mode corrected by piezo energy carryover, first pulse					
MFF_SP_2_HOM_COR_INJ[NC_CYL_NR]	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual mass fuel flow set-point for homogeneous mode corrected by piezo energy carryover, second pulse					
MFF_SP_2_S_COR_INJ[NC_CYL_NR]	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual mass fuel flow set-point for stratified mode corrected by piezo energy carryover, second pulse					
MFF_SP_3_HOM_COR_INJ[NC_CYL_NR]	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual mass fuel flow set-point for homogeneous mode corrected by piezo energy carryover, third pulse					
MFF_SP_3_S_COR_INJ[NC_CYL_NR]	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Cylinder individual mass fuel flow set-point for stratified mode corrected by piezo energy carryover, third pulse					
RATIO_T_PER_PRS_COR_1_HOM	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Ratio of offset time to periode time of fuel rail pressure pulsation; homogeneous first pulse					

RATIO_T_PER_PRS_COR_2_HOM	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Ratio of offset time to periode time of fuel rail pressure pulsation; homogeneous second pulse					
RATIO_T_PER_PRS_COR_3_HOM	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Ratio of offset time to periode time of fuel rail pressure pulsation; homogeneous third pulse					
RATIO_TI_1_HOM	V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Ratio of injection time to periode time of fuel rail pressure pulsation; homogeneous first pulse					
RATIO_TI_2_HOM	V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Ratio of injection time to periode time of fuel rail pressure pulsation; homogeneous second pulse					
RATIO_TI_3_HOM	V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Ratio of injection time to periode time of fuel rail pressure pulsation; homogeneous third pulse					
T_AUTH_TI_MIN_AFL	-	0...FFFFH	0.01...655.36	0.01	[s]
Delay time after AFL mode was inhibited due to injection time limitation					
T_CHA_PER_HOM_1[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Injector charge periode; 1st pulse HOM.					
T_CHA_PER_HOM_2[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Injector charge periode; 2nd pulse HOM.					
T_CHA_PER_HOM_3[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Injector charge periode; 3rd pulse HOM.					
T_CHA_PER_POST	O/V	0...12CH	0...0.3	0.001	[ms]
Injector charge periode for post pulse.					
T_CHA_PER_S_1[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Injector charge periode; 1st pulse S.					
T_CHA_PER_S_2[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Injector charge periode; 2nd pulse S.					
T_CHA_PER_S_3[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Injector charge periode; 3rd pulse S.					
T_DCHA_DLY_HOM_1[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Discharge pause time between first and second discharge phase; 1st pulse HOM.					
T_DCHA_DLY_HOM_2[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Discharge pause time between first and second discharge phase; 2nd pulse HOM.					
T_DCHA_DLY_HOM_3[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Discharge pause time between first and second discharge phase; 3rd pulse HOM.					
T_DCHA_DLY_S_1[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Discharge pause time between first and second discharge phase; 1st pulse S.					
T_DCHA_DLY_S_2[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Discharge pause time between first and second discharge phase; 2nd pulse S.					
T_DCHA_DLY_S_3[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Discharge pause time between first and second discharge phase; 3rd pulse S.					
T_DCHA_PER_HOM_1_1[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode; 1st pulse HOM.					
T_DCHA_PER_HOM_1_2[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Second injector discharge periode; 1st pulse HOM.					
T_DCHA_PER_HOM_2_1[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode; 2nd pulse HOM.					
T_DCHA_PER_HOM_2_2[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Second injector discharge periode; 2nd pulse HOM.					
T_DCHA_PER_HOM_3_1[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode; 3rd pulse HOM.					
T_DCHA_PER_HOM_3_2[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Second injector discharge periode; 3rd pulse HOM.					
T_DCHA_PER_POST	O/V	0...12CH	0...0.3	0.001	[ms]
Injector discharge periode for post pulse.					
T_DCHA_PER_S_1_1[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode; 1st pulse S.					
T_DCHA_PER_S_1_2[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Second injector discharge periode; 1st pulse S.					
T_DCHA_PER_S_2_1[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode; 2nd pulse S.					
T_DCHA_PER_S_2_2[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Second injector discharge periode; 2nd pulse S.					

T_DCHA_PER_S_3_1[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode; 3rd pulse S.					
T_DCHA_PER_S_3_2[NC_CYL_NR]	O/V	0...12CH	0...0.3	0.001	[ms]
Second injector discharge periode; 3rd pulse S.					
T_DLY_1_1_HOM	V	0...FFFFH	0...65.535	0.001	[ms]
Delay time between the first pulses of two consecutive cylinders at homogeneous single injection					
T_DLY_1_2_HOM	V	0...FFFFH	0...65.535	0.001	[ms]
Delay time between the first and second pulse at homogeneous multi injection					
T_DLY_2_3_HOM	V	0...FFFFH	0...65.535	0.001	[ms]
Delay time between the second and third pulse at homogeneous multi injection					
T_PER_PRS_COR_1_HOM	V	0...FFFFH	0...65.535	0.001	[ms]
Periode time of fuel rail pressure pulsation; homogeneous first pulse					
T_PER_PRS_COR_2_HOM	V	0...FFFFH	0...65.535	0.001	[ms]
Periode time of fuel rail pressure pulsation; homogeneous second pulse					
T_PER_PRS_COR_3_HOM	V	0...FFFFH	0...65.535	0.001	[ms]
Periode time of fuel rail pressure pulsation; homogeneous third pulse					
TEMP_PRS_COR_1_HOM	V	0...FEH	-48...142.5	0.75	[°C]
Fuel temperature for pressure pulsations correction; first pulse homogeneous mode					
TEMP_PRS_COR_2_HOM	V	0...FEH	-48...142.5	0.75	[°C]
Fuel temperature for pressure pulsations correction; second pulse homogeneous mode					
TI_1_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, homogeneous mode, first pulse (intermediate value)					
TI_1_HOM_CLC[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, homogeneous mode, first pulse (intermediate value)					
TI_1_HOM_CLC_MAX	O/V	0...FFFFH	0...65.535	0.001	[ms]
Maximum injection time of first pulse, homogeneous mode					
TI_1_HOM_CLC_MIN	V	0...FFFFH	0...65.535	0.001	[ms]
Minimum injection time of first pulse, homogeneous mode					
TI_1_S[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, stratified mode, fist pulse (intermediate value)					
TI_1_S_CLC[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, stratified mode, fist pulse (intermediate value)					
TI_1_S_CLC_MIN	V	0...FFFFH	0...65.535	0.001	[ms]
Minimum injection time of first pulse, stratified mode					
TI_2_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, homogeneous mode, second pulse (intermediate value)					
TI_2_HOM_CLC[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, homogeneous mode, second pulse (intermediate value)					
TI_2_HOM_CLC_MAX	O/V	0...FFFFH	0...65.535	0.001	[ms]
Maximum injection time of second pulse, homogeneous mode					
TI_2_HOM_CLC_MIN	V	0...FFFFH	0...65.535	0.001	[ms]
Minimum injection time of second pulse, homogeneous mode					
TI_2_S[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, stratified mode, second pulse (intermediate value)					
TI_2_S_CLC[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, stratified mode, second pulse (intermediate value)					
TI_2_S_CLC_MIN	V	0...FFFFH	0...65.535	0.001	[ms]
Minimum injection time of second pulse, stratified mode					
TI_3_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, homogeneous mode, third pulse					
TI_3_HOM_CLC[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, homogeneous mode, third pulse (intermediate value)					
TI_3_HOM_CLC_MAX	O/V	0...FFFFH	0...65.535	0.001	[ms]
Maximum injection time of third pulse, homogeneous mode					
TI_3_HOM_CLC_MIN	V	0...FFFFH	0...65.535	0.001	[ms]
Minimum injection time of third pulse, homogeneous mode					
TI_3_S[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, stratified mode, third pulse					
TI_3_S_CLC[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, stratified mode, third pulse (intermediate value)					

TI_3_S_CLC_MIN	V	0...FFFFH	0...65.535	0.001	[ms]
Minimum injection time of third pulse, stratified mode					
TI_MIN_HOM	O/V	0...FFFFH	0...65.535	0.001	[ms]
Minimum injection time limitation for homogeneous mode					

Input data:

TI_TUN_IV[NC_CYL_NR]	TI_FAC[NC_CYL_NR]	MFF_SP_1_HOM[NC_CYL_NR]	LV_AFL_CLC
CRK_DIF_SOI_IGN_S	NC_CBK_HPP_NR	MFF_SP_2_HOM[NC_CYL_NR]	FAC_TI_COR_IV_EGY_RNG_L[NC_CBK_HPP_NR]
PRS_DEC_INJ_1_HOM[NC_CYL_NR]	PRS_DEC_INJ_1_S[NC_CYL_NR]	TI_TUN_ADD_IV[NC_CYL_NR]	LV_S_CLC
STATE_INJ_MOD_S_REQ	LF_PRS_COR_HPP_ENA	FUP_H_INJ[NC_CBK_HPP_NR]	TI_2_STND_S[NC_CYL_NR]
TI_ADD[NC_CYL_NR]	MFF_ADD_REAC[NC_CYL_NR]	CTR_CYL_NR_ST_CLC_INJR	STATE_CMB_CTL
CRK_DIF_SOI_IGN_HOM			LV_ST_END
MFF_SP_1_S[NC_CYL_NR]	FAC_MFF_DIF_HOM[NC_CYL_NR]	T_DLY_1_2_S_EXT[NC_CYL_NR]	STATE_INJ_MOD_REQ
T_DLY_2_3_S_EXT[NC_CYL_NR]	MFF_SP_2_S[NC_CYL_NR]	TI_3_STND_S[NC_CYL_NR]	NC_IDX_CYL_HPP_REF[NC_CYL_NR]
MFF_SP_3_HOM[NC_CYL_NR]	STATE_INJ_MOD_HOMS_REQ	FAC_TI_1_PRS_CYL_HOM[NC_CYL_NR]	FAC_TI_2_PRS_CYL_HOM[NC_CYL_NR]
FAC_TI_3_PRS_CYL_HOM[NC_CYL_NR]	FAC_TI_1_PRS_CYL_S[NC_CYL_NR]	FAC_TI_2_PRS_CYL_S[NC_CYL_NR]	FAC_TI_3_PRS_CYL_S[NC_CYL_NR]
PRS_DEC_INJ_2_HOM[NC_CYL_NR]	PRS_DEC_INJ_3_HOM[NC_CYL_NR]	PRS_DEC_INJ_2_S[NC_CYL_NR]	PRS_DEC_INJ_3_S[NC_CYL_NR]
	LV_EGY_RNG_IV_PLS_2_HOM[NC_CYL_NR]	LV_EGY_RNG_IV_PLS_3_HOM[NC_CYL_NR]	LV_EGY_RNG_IV_PLS_1_S[NC_CYL_NR]
LV_EGY_RNG_IV_PLS_2_S[NC_CYL_NR]	LV_EGY_RNG_IV_PLS_3_S[NC_CYL_NR]	NR_CYL_CLC_RED_INJ	INJ_MOD_SP_HOM[NC_CYL_NR]
MFF_SP_3_S[NC_CYL_NR]	FAC_MFF_DIF_S[NC_CYL_NR]	LV_EGY_RNG_IV_PLS_1_HOM[NC_CYL_NR]	IDX_PRS_COR_CYL_CLC_S[NC_CBK_HPP_NR]
INJ_MOD_SP_S[NC_CYL_NR]	INJ_MOD_SP_HOMS[NC_CYL_NR]	FAC_MFF_TFU	FAC_N
IDX_PRS_COR_CYL_CLC_MPLH[NC_CBK_HPP_NR]	NC_T_DCHA_DLY_MIN	TFU_INJ	TEMP_CAPA_IV_MV
TCO	SOI_1_HOM_EXT[NC_CYL_NR]	EOI_2_HOM_EXT[NC_CYL_NR]	EOI_3_HOM_EXT[NC_CYL_NR]
TI_2_PRS_HOM_SP[NC_CYL_NR]	TI_3_PRS_HOM_SP[NC_CYL_NR]	LV_PRS_COR_SNGH_ENA	LV_PRS_COR_MPLH_ENA
NC_INJ_MOD_MASK_1	NC_INJ_MOD_HOM	NC_NR_IDX_TI_PRM_H	NC_NR_IDX_TI_PRM_L
NC_NR_IDX_MFF_INJ	NC_NR_IDX_T_DLY_INJ	NC_NR_IDX_FUP_INJ	NC_NR_IDX_TEMP_INJ
NC_NR_IDX_RATIO_T_INJ	NC_NR_IDX_MFF_2_INJ	NC_NR_IDX_PRS_DEC_INJ	LV_IV_POST_EGY_RNG

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TI_MIN_HOM	6	0...FFFFH	0...65.535	0.001	[ms]
LDPM_PRS_DEC_INJ_1_INJR	6	0...FFFFH	0...347776	5.3067216	[hPa]
Minimum injection time limitation for homogeneous mode					
IP_MFF_TI_EGY_H	NC_NR_IDX_MFF_2_INJ*NC_NR_IDX_PRS_DEC_INJ	0...FFFFH	0...65.535	0.001	[ms]
LDPM_MFF_SP_2_INJR	NC_NR_IDX_MFF_2_INJ	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_PRS_DEC_INJ_3_INJR	NC_NR_IDX_PRS_DEC_INJ	0...FFFFH	0...347776	5.3067216	[hPa]
Injector characteristic at different injector pressure differences, for high energy range					
IP_MFF_TI_EGY_L	NC_NR_IDX_MFF_2_INJ*NC_NR_IDX_PRS_DEC_INJ	0...FFFFH	0...65.535	0.001	[ms]
LDPM_MFF_SP_2_INJR	NC_NR_IDX_MFF_2_INJ	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_PRS_DEC_INJ_3_INJR	NC_NR_IDX_PRS_DEC_INJ	0...FFFFH	0...347776	5.3067216	[hPa]
Injector characteristic at different injector pressure differences, for low energy range					
IP_MFF_IDX_TI_EGY_H	16*12	0...FFFFH	0...65535	1	[-]
LDPM_MFF_SP_1_INJR	16	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_PRS_DEC_INJ_2_INJR	12	0...FFFFH	0...347776	5.3067216	[hPa]
Injector characteristic at different injector pressure differences, for high energy range					
IP_MFF_IDX_TI_EGY_L	16*12	0...FFFFH	0...65535	1	[-]
LDPM_MFF_SP_1_INJR	16	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDPM_PRS_DEC_INJ_2_INJR	12	0...FFFFH	0...347776	5.3067216	[hPa]
Injector characteristic at different injector pressure differences, for low energy range					
IP_FAC_TI_2_S_DLY_COR	16*12	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_T_DLY_1_2_S_IP_FAC_TI_2_S	16	0...FFFFH	0...65.535	0.001	[ms]
LDP_MFF_SP_IP_FAC_TI_2_S	12	0...FFFFH	0...1389	0.0211948	[mg/stk]
This calibration data is for correction of the mass flow difference between the mass flow sum of two pulses in comparison with the same requested mass for one pulse. The difference is caused by physical effects of the piezo injector.					
IP_FAC_TI_2_S_COR	8*8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_TI_2_S_IP_FAC_TI_2_S_COR	8	0...FFFFH	0...65.535	0.001	[ms]
LDP_T_DLY_1_2_S_IP_FAC_COR_2_S	8	0...FFFFH	0...65.535	0.001	[ms]
This calibration data is for correction of the mass flow difference between the mass flow sum of two pulses in comparison with the same requested mass for one pulse. This correction compensate different second injection duration.					
IP_FAC_TI_3_S_DLY_COR	8*8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_T_DLY_2_3_S_IP_FAC_TI_3_S	8	0...FFFFH	0...65.535	0.001	[ms]

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LDP_MFF_SP_IP_FAC_TI_3_S	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
This calibration data is for correction of the mass flow difference between the mass flow sum of two pulses in comparison with the same requested mass for one pulse. The difference is caused by physical effects of the piezo injector.					
IP_FAC_TI_3_S_COR	8*8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_TI_3_S_IP_FAC_TI_3_S_COR	8	0...FFFFH	0...65.535	0.001	[ms]
LDP_T_DLY_2_3_S_IP_FAC_COR_3_S	8	0...FFFFH	0...65.535	0.001	[ms]
This calibration data is for correction of the mass flow difference between the mass flow sum of two pulses in comparison with the same requested mass for one pulse. This correction compensate different third injection duration.					
C_TI_1_HOM_MAN	1	0...FFFFH	0...65.535	0.001	[ms]
Manual setpoint for injection time in homogeneous mode, 1. pulse					
C_TI_2_HOM_MAN	1	0...FFFFH	0...65.535	0.001	[ms]
Manual setpoint for injection time in homogeneous mode, 2. pulse					
C_TI_3_HOM_MAN	1	0...FFFFH	0...65.535	0.001	[ms]
Manual setpoint for injection time in homogeneous mode, 3. pulse					
LC_TI_1_HOM_MAN_ACT	1	0...1H	0...1	1	[-]
Manual switch for manual setpoint for injection time in homogeneous mode					
LC_TI_2_HOM_MAN_ACT	1	0...1H	0...1	1	[-]
Manual switch for manual setpoint for injection time in homogeneous mode, 2. pulse					
LC_TI_3_HOM_MAN_ACT	1	0...1H	0...1	1	[-]
Manual switch for manual setpoint for injection time in homogeneous mode, 3. pulse					
C_TI_MIN_OFS	1	0...FFH	0...1.02	0.004	[ms]
Offset to TI_MIN, (injection time limitation at AFL mode and stratified mode)					
C_T_AUTH_TI_MIN_AFL	1	0...FFFFH	0.01...655.36	0.01	[s]
Delay time after AFL mode was inhibited due to injection time limitation					
C_TI_MIN_HYS_S	1	0...FFH	0...1.02	0.004	[ms]
Hysteresis to TI_MIN, (injection time limitation at stratified mode)					
C_TI_1_S_MAN	1	0...FFFFH	0...65.535	0.001	[ms]
Manual setpoint for injection time in stratified mode, 1. pulse					
LC_TI_1_S_MAN_ACT	1	0...1H	0...1	1	[-]
Switch for manual setpoint for injection time in startified mode, 1. pulse					
C_TI_2_S_MAN	1	0...FFFFH	0...65.535	0.001	[ms]
Manual setpoint for injection time in stratified mode, 2. pulse					
LC_TI_2_S_MAN_ACT	1	0...1H	0...1	1	[-]
switch for manual setpoint for injection time in stratified mode, 2. pulse					
C_TI_3_S_MAN	1	0...FFFFH	0...65.535	0.001	[ms]
Manual setpoint for injection time in stratified mode, 3. pulse					
LC_TI_3_S_MAN_ACT	1	0...1H	0...1	1	[-]
switch for manual setpoint for injection time in stratified mode, 3. pulse					
LC_PRS_COR_MPLS_ENA	1	0...1H	0...1	1	[-]
Switch to enable correction of rail pressure pulsations in multiple stratified mode					
LC_PRS_COR_SNGH_ENA	1	0...1H	0...1	1	[-]
Switch to enable correction of rail pressure pulsations in homogeneous single injection mode					
LC_PRS_COR_MPLH_ENA	1	0...1H	0...1	1	[-]
Switch to enable correction of rail pressure pulsations in homogeneous multi injection mode					
LC_PRS_COR_SOI_DLY_ENA	1	0...1H	0...1	1	[-]
Switch to enable delaytime calculation from start of injection in homogeneous mode					
C_PRS_COR_TEMP_SEL	1	0...8H	0...8	1	[-]
Constant to select temperature for pressure pulsation correction (0=TFU_INJ; 1=TEMP_CAPA_IV_MV; 2=TCO)					
C_IDX_TI_MIN_H	1	0...FFFFH	0...65535	1	[-]
Minimum counting factor index for high needle lift calculations.					
C_IDX_TI_MIN_L	1	0...FFFFH	0...65535	1	[-]
Minimum counting factor index for low needle lift calculations.					
C_TI_OFS_H	1	8000...7FFFH	-32.768...32.767	0.001	[ms]
Injection time offset for high needle lift pulses that exceed special timing parameter calculation.					
C_TI_OFS_L	1	8000...7FFFH	-32.768...32.767	0.001	[ms]
Injection time offset for low needle lift pulses that exceed special timing parameter calculation.					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H]	1	0...12CH	0...0.3	0.001	[ms]
Injector charge periode for high injector needle lift; depending on counting factor index.					
C_T_CHA_PER_L[NC_NR_IDX_TI_PRM_L]	1	0...12CH	0...0.3	0.001	[ms]
Injector charge periode for low injector needle lift; depending on counting factor index.					
C_T_DCHA_PER_1_H[NC_NR_IDX_TI_PRM_H]	1	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode for high injector needle lift; depending on counting factor index.					
C_T_DCHA_PER_1_L[NC_NR_IDX_TI_PRM_L]	1	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode for low injector needle lift; depending on counting factor index.					
C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H]	1	0...FFFFH	0...65.535	0.001	[ms]
Injector discharge pause periode for high injector needle lift; depending on counting factor index.					
C_T_DCHA_DLY_L[NC_NR_IDX_TI_PRM_L]	1	0...FFFFH	0...65.535	0.001	[ms]
Injector discharge pause periode for low injector needle lift; depending on counting factor index.					
C_TI_H[NC_NR_IDX_TI_PRM_H]	1	0...FFFFH	0...65.535	0.001	[ms]
Injection time for high injector needle lift; depending on counting factor index.					
C_TI_L[NC_NR_IDX_TI_PRM_L]	1	0...FFFFH	0...65.535	0.001	[ms]
Injection time for low injector needle lift; depending on counting factor index.					
IP_FAC_PRS_COR_MFF_1_HOM	NC_NR_IDX_MFF_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_MFF_1_INJR	NC_NR_IDX_MFF_INJ	0...FFFFH	0...1389	0.0211948	[mg/stk]
Pressure pulsation correction factor due to previous injected fuel mass; homogeneous first pulse					
IP_FAC_PRS_COR_MFF_2_HOM	NC_NR_IDX_MFF_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_MFF_1_INJR	NC_NR_IDX_MFF_INJ	0...FFFFH	0...1389	0.0211948	[mg/stk]
Pressure pulsation correction factor due to previous injected fuel mass; homogeneous second pulse					
IP_FAC_PRS_COR_MFF_3_HOM	NC_NR_IDX_MFF_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_MFF_1_INJR	NC_NR_IDX_MFF_INJ	0...FFFFH	0...1389	0.0211948	[mg/stk]
Pressure pulsation correction factor due to previous injected fuel mass; homogeneous third pulse					
IP_FAC_PRS_COR_DAMP_1_HOM	NC_NR_IDX_T_DLY_INJ*NC_NR_IDX_FUP_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_T_DLY_1_INJR	NC_NR_IDX_T_DLY_INJ	0...FFFFH	0...65.535	0.001	[ms]
LDPM_FUP_H_2_INJR	NC_NR_IDX_FUP_INJ	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure pulsation correction factor due to delay time and fuel pressure; homogeneous first pulse					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_PRS_COR_DAMP_2_HOM	NC_NR_IDX_T_DLY_INJ*NC_NR_IDX_FUP_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_T_DLY_1_INJR	NC_NR_IDX_T_DLY_INJ	0...FFFFH	0...65.535	0.001	[ms]
LDPM_FUP_H_2_INJR	NC_NR_IDX_FUP_INJ	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure pulsation correction factor due to delay time and fuel pressure; homogeneous second pulse					
IP_FAC_PRS_COR_DAMP_3_HOM	NC_NR_IDX_T_DLY_INJ*NC_NR_IDX_FUP_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_T_DLY_1_INJR	NC_NR_IDX_T_DLY_INJ	0...FFFFH	0...65.535	0.001	[ms]
LDPM_FUP_H_2_INJR	NC_NR_IDX_FUP_INJ	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure pulsation correction factor due to delay time and fuel pressure; homogeneous third pulse					
ID_T_PER_PRS_COR_1_HOM	5	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_NR_OSC_1_INJR	5	0...9H	0...9	1	[-]
Pressure pulsation correction factor due increasing delay time; homogeneous first pulse					
ID_T_PER_PRS_COR_2_HOM	5	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_NR_OSC_1_INJR	5	0...9H	0...9	1	[-]
Pressure pulsation correction factor due increasing delay time; homogeneous second pulse					
ID_T_PER_PRS_COR_3_HOM	5	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_NR_OSC_1_INJR	5	0...9H	0...9	1	[-]
Pressure pulsation correction factor due increasing delay time; homogeneous third pulse					
IP_T_PER_PRS_COR_1_HOM	NC_NR_IDX_TEMP_INJ*NC_NR_IDX_FUP_INJ	0...FFFFH	0...65.535	0.001	[ms]
LDPM_TEMP_PRS_COR_1_INJR	NC_NR_IDX_TEMP_INJ	0...FEH	-48...142.5	0.75	[°C]
LDPM_FUP_H_2_INJR	NC_NR_IDX_FUP_INJ	0...FFFFH	0...347776	5.3067216	[hPa]
Periode time of fuel rail pressure pulsation; homogeneous first pulse					
IP_T_PER_PRS_COR_2_HOM	NC_NR_IDX_TEMP_INJ*NC_NR_IDX_FUP_INJ	0...FFFFH	0...65.535	0.001	[ms]

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LDPM_TEMP_PRS_COR_1_INJR	NC_NR_IDX_TEMP_INJ	0...FEH	-48...142.5	0.75	[°C]
LDPM_FUP_H_2_INJR	NC_NR_IDX_FUP_INJ	0...FFFFH	0...347776	5.3067216	[hPa]
Periode time of fuel rail pressure pulsation; homogeneous second pulse					
IP_T_PER_PRS_COR_3_HOM	NC_NR_IDX_TEMP_INJ*NC_NR_IDX_FUP_INJ	0...FFFFH	0...65.535	0.001	[ms]
LDPM_TEMP_PRS_COR_1_INJR	NC_NR_IDX_TEMP_INJ	0...FEH	-48...142.5	0.75	[°C]
LDPM_FUP_H_2_INJR	NC_NR_IDX_FUP_INJ	0...FFFFH	0...347776	5.3067216	[hPa]
Periode time of fuel rail pressure pulsation; homogeneous third pulse					
IP_FAC_PRS_COR_OSC_1_HOM	NC_NR_IDX_RATIO_T_INJ*NC_NR_IDX_RATIO_T_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_RATIO_T_PER_1_INJR	NC_NR_IDX_RATIO_T_INJ	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_RATIO_TI_1_INJR	NC_NR_IDX_RATIO_T_INJ	0...FFFFH	0...15.99975	0.2441e-3	[-]
Pressure pulsation correction factor due to delay time and injection time; homogeneous first pulse					
IP_FAC_PRS_COR_OSC_2_HOM	NC_NR_IDX_RATIO_T_INJ*NC_NR_IDX_RATIO_T_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_RATIO_T_PER_1_INJR	NC_NR_IDX_RATIO_T_INJ	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_RATIO_TI_1_INJR	NC_NR_IDX_RATIO_T_INJ	0...FFFFH	0...15.99975	0.2441e-3	[-]
Pressure pulsation correction factor due to delay time and injection time; homogeneous second pulse					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_PRS_COR_OSC_3_HOM	NC_NR_IDX_RATIO_T_INJ*NC_NR_IDX_RATIO_T_INJ	0...FFFFH	-1...0.99996	0.0305e-3	[-]
LDPM_RATIO_T_PER_1_INJR	NC_NR_IDX_RATIO_T_INJ	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_RATIO_TI_1_INJR	NC_NR_IDX_RATIO_T_INJ	0...FFFFH	0...15.99975	0.2441e-3	[-]
Pressure pulsation correction factor due to delay time and injection time; homogeneous third pulse					

FUNCTION DESCRIPTION:

41.3.1 Segment synchronous tasks

Application conditions:

Activation: every engine state

Deactivation: -

Initialization: at reset: TI_3_HOM_CLC_MIN = 65.535 ms
TI_2_HOM_CLC_MIN = 65.535 ms
TI_1_HOM_CLC_MIN = 65.535 ms

Initialize all NC_CYL_NR elements of following variables:

T_CHA_PER_HOM_1[ ] =
C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
T_CHA_PER_HOM_2[ ] =
C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
T_CHA_PER_HOM_3[ ] =
C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
T_CHA_PER_S_1[ ] =
C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
T_CHA_PER_S_2[ ] =
C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
T_CHA_PER_S_3[ ] =
C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]

T_DCHA_PER_HOM_1_2[ ] = T_CHA_PER_HOM_1[ ]
T_DCHA_PER_HOM_2_2[ ] = T_CHA_PER_HOM_2[ ]
T_DCHA_PER_HOM_3_2[ ] = T_CHA_PER_HOM_3[ ]
T_DCHA_PER_S_1_2[ ] = T_CHA_PER_S_1[ ]
T_DCHA_PER_S_2_2[ ] = T_CHA_PER_S_2[ ]
T_DCHA_PER_S_3_2[ ] = T_CHA_PER_S_3[ ]

T_DCHA_PER_HOM_1_1[ ] = 0

T_DCHA_PER_HOM_2_1[ ] = 0
T_DCHA_PER_HOM_3_1[ ] = 0
T_DCHA_PER_S_1_1[ ] = 0
T_DCHA_PER_S_2_1[ ] = 0
T_DCHA_PER_S_3_1[ ] = 0

T_DCHA_DLY_HOM_1[ ] = 0
T_DCHA_DLY_HOM_2[ ] = 0
T_DCHA_DLY_HOM_3[ ] = 0
T_DCHA_DLY_S_1[ ] = 0
T_DCHA_DLY_S_2[ ] = 0
T_DCHA_DLY_S_3[ ] = 0

Recurrence: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

<u>Formula section:</u>

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
                x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
    <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

41.3.1.1 Calculation of the cylinder individual injection times for homogeneous mode

41.3.1.1.1 Calculation of the minimal applied injection time

TI_MIN_HOM is used to for deactivation of air fuel lean mode.

```
(1)IF LV_AFL_CLC = 1
(1)THEN
    TI_MIN_HOM = IP_TI_MIN_HOM
    ... (Input data for IP_TI_MIN_HOM is
                PRS_DEC_INJ_1_HOM[IDX_PRS_COR_CYL_CLC_MPLH[0]])
(1)ENDIF
```

41.3.1.1.2 First pulse (main pulse)

Correction of fuel pressure pulsations for homogeneous single injection mode:

```
(1)IF STATE_INJ_MOD_HOM_REQ = "SNGH"
     AND LC_PRS_COR_SNGH_ENA = 1
     AND LV_PRS_COR_SNGH_ENA = 1
(1)THEN
     (1a)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
        (1a)IF Bit m of LF_PRS_COR_HPP_ENA = 1
        (1a)THEN
           Correction due to MFF_SP_1_HOM:
           FAC_PRS_COR_MFF_1_HOM = IP_FAC_PRS_COR_MFF_1_HOM(
             Input: (MFF_SP_1_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] +
             MFF_ADD_REAC[IDX_PRS_COR_CYL_CLC_MPLH[m]]) * (1+
             FAC_MFF_DIF_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]) * FAC_MFF_TFU)
           Correction of damping depending on FUP and delay time between pulses:
           (2)IF LC_PRS_COR_SOI_DLY_ENA = 1
           (2)THEN
             T_DLY_1_1_HOM = (720°/(NC_CYL_NR/NC_CBK_HPP_NR)) / FAC_N
           (2)ELSE
             T_DLY_1_1_HOM = (720°/(NC_CYL_NR/NC_CBK_HPP_NR)) / FAC_N
                             - TI_1_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                             - T_DCHA_DLY_HOM_1[IDX_PRS_COR_CYL_CLC_MPLH[m]]
             Note: Previous calculated TI_1_HOM is used to calculate delay time between
                   pulses
           (2)ENDIF
           FAC_PRS_COR_DAMP_1_HOM = IP_FAC_PRS_COR_DAMP_1_HOM(
             Inputs: T_DLY_1_1_HOM; FUP_H_INJ[m])
           Calculate period time of pressure waves:
           (3)IF C_PRS_COR_TEMP_SEL = 0
         (3)THEN
           TEMP_PRS_COR_1_HOM = TFU_INJ
         (3)ELSE
           (4) IF C_PRS_COR_TEMP_SEL = 1
           (4)THEN
             TEMP_PRS_COR_1_HOM =
                             TEMP_CAPA_IV_MV
           (4)ELSE
             TEMP_PRS_COR_1_HOM = TCO
           (4)ENDIF
         (3)ENDIF
         T_PER_PRS_COR_1_HOM = IP_T_PER_PRS_COR_1_HOM(
             Inputs: TEMP_PRS_COR_1_HOM; FUP_H_INJ[m])
         Calculate Time offset within one wave:
```

```
        i = 0 // i indicates the number of while loops
        (1)WHILE T_DLY_1_1_HOM > T_PER_PRS_COR_1_HOM DO
            T_DLY_1_1_HOM = T_DLY_1_1_HOM
                - (T_PER_PRS_COR_1_HOM * ID_T_PER_PRS_COR_1_HOM(Input: i))
            Note: In case of i > 9 use 9 as input for ID map to prevent overflow
            i = i + 1
            Note: To prevent an endless loop, break the while-loop if i reaches 50. In that
                case, set T_DLY_1_1_HOM to 0!
        (1)ENDWHILE
        Calculate normalized Time offset within one wave:
        RATIO_T_PER_PRS_COR_1_HOM =
            T_DLY_1_1_HOM/(T_PER_PRS_COR_1_HOM
                                * ID_T_PER_PRS_COR_1_HOM(Input: i))
        RATIO_TI_1_HOM = (TI_1_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                + T_DCHA_DLY_HOM_1[IDX_PRS_COR_CYL_CLC_MPLH[m]])
            / (T_PER_PRS_COR_1_HOM * ID_T_PER_PRS_COR_1_HOM(Input: i))
        Calculate correction due to wave position and injection time:
        FAC_PRS_COR_OSC_1_HOM = IP_FAC_PRS_COR_OSC_1_HOM(
            Inputs: RATIO_T_PER_PRS_COR_1_HOM; RATIO_TI_1_HOM)
            Note: Previous calculated TI_1_HOM is used to calculate factor
        Calculate final correction factor:
        FAC_TI_PRS_COR_1_HOM[m] = 1 + (FAC_PRS_COR_MFF_1_HOM
                * FAC_PRS_COR_DAMP_1_HOM * FAC_PRS_COR_OSC_1_HOM)
      (1a)ENDIF
    (1a)ENDFOR
(1)ELSE
    (1b)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
        FAC_TI_PRS_COR_1_HOM[m] = 1
    (1b)ENDFOR
(1)ENDIF
```

Calculation of injection time for homogeneous first pulse:

```
(1)FOR cylinder_start to cylinder_stop DO
MFF_SP_1_HOM_COR_INJ[x] = (MFF_SP_1_HOM[x] + MFF_ADD_REAC[x]) * (1+
        FAC_MFF_DIF_HOM[x]) * FAC_MFF_TFU

(2)IF    LV_EGY_RNG_IV_PLS_1_HOM[x] = 1
(2)THEN
    IDX_TI_1_STND_HOM[x] = IP_MFF_IDX_TI_EGY_H(MFF_SP_1_HOM_COR_INJ[x];
                        PRS_DEC_INJ_1_HOM[x])
(2)ELSE
```

```
        IDX_TI_1_STND_HOM[x] = IP_MFF_IDX_TI_EGY_L(MFF_SP_1_HOM_COR_INJ[x];
                                PRS_DEC_INJ_1_HOM[x])
                                * FAC_TI_COR_IV_EGY_RNG_L[NC_IDX_CYL_HPP_REF[x]]
    (2)ENDIF

    IDX_TI_1_HOM_CLC[x] =          IDX_TI_1_STND_HOM[x]
                              * FAC_TI_1_PRS_CYL_HOM[x]
                              * TI_FAC[x]
                              * TI_TUN_IV[x]
                              * FAC_TI_PRS_COR_1_HOM[NC_IDX_CYL_HPP_REF[x]]
                              + (TI_TUN_ADD_IV[x] / 0.001ms)
                              + (TI_ADD[x] / 0.001ms)

    Calculate_Pulse_timing_parameters(IN <LV_EGY_RNG_IV_PLS_1_HOM[x]>,
        IN <IDX_TI_1_HOM_CLC[x]>, OUT <T_CHA_PER_HOM_1[x]>,
        OUT <T_DCHA_PER_HOM_1_1[x]>, OUT <T_DCHA_PER_HOM_1_2[x]>,
        OUT <T_DCHA_DLY_HOM_1[x]>, OUT <TI_1_HOM_CLC[x]>)


(1) ENDFOR
TI_1_HOM_CLC_MIN  = MIN(TI_1_HOM_CLC[ ]) // Minimum of all NC_CYL_NR cylinders
TI_1_HOM_CLC_MAX  = MAX(TI_1_HOM_CLC[ ]) // Maximum of all NC_CYL_NR cylinders
IDX_TI_1_HOM_MIN = MIN(IDX_TI_1_HOM_CLC) // Minimum of all NC_CYL_NR cylinders
(4) IF LC_TI_1_HOM_MAN_ACT = 1
(4) THEN
    (4)FOR cylinder_start to cylinder_stop DO
        TI_1_HOM[x]  = C_TI_1_HOM_MAN
        T_CHA_PER_HOM_1[x] = C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
        T_DCHA_PER_HOM_1_1[x] =
                    C_T_DCHA_PER_1_H[NC_NR_IDX_TI_PRM_H – 1]
        T_DCHA_DLY_HOM_1[x] = C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H – 1]
        T_DCHA_PER_HOM_1_2[x] =
                    T_CHA_PER_HOM_1[x] – T_DCHA_PER_HOM_1_1[x]
    (4)ENDFOR
    LV_TI_1_HOM_MIN = 0
(4) ELSE
    (5)FOR cylinder_start to cylinder_stop DO
        TI_1_HOM[x] = TI_1_HOM_CLC[x]
    (5)ENDFOR
    (5a)IF IDX_TI_1_HOM_MIN < C_IDX_TI_MIN_H
    (5a)THEN
        LV_TI_1_HOM_MIN = 1
```

```
    (5a)ELSE
            LV_TI_1_HOM_MIN = 0
    (5a)ENDIF
(4) ENDIF
```

Note: For calibration purposes the injection time TI_1_HOM[x] can be set manually to the value C_TI_1_HOM_MAN

41.3.1.1.3 Second pulse

Correction of fuel pressure pulsations for homogeneous multi injection mode:

```
(1)IF (STATE_INJ_MOD_HOM_REQ = "MPLH"
        OR STATE_INJ_MOD_HOM_REQ = "MPLH+PLS3")
    AND LC_PRS_COR_MPLH_ENA = 1
    AND LV_PRS_COR_MPLH_ENA = 1
(1)THEN
    (2a)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
        (2a)IF Bit m of LF_PRS_COR_HPP_ENA = 1
        (2a)THEN
            Correction due to MFF_SP_1_HOM:
            FAC_PRS_COR_MFF_2_HOM = IP_FAC_PRS_COR_MFF_2_HOM(
                Input: MFF_SP_1_HOM_COR_INJ[IDX_PRS_COR_CYL_CLC_MPLH[m]])
            Correction of damping depending on FUP and delay time between pulses:
            (2)IF LC_PRS_COR_SOI_DLY_ENA = 1
            (2)THEN
                T_DLY_1_2_HOM =
                    (SOI_1_HOM_EXT[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                        - (EOI_2_HOM_EXT[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                        + TI_2_PRS_HOM_SP[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                                    * FAC_N)) / FAC_N
            (2)ELSE
                T_DLY_1_2_HOM =
                    (SOI_1_HOM_EXT[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                    - (TI_1_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                    + T_DCHA_DLY_HOM_1[IDX_PRS_COR_CYL_CLC_MPLH[m]])
                        * FAC_N
                        - (EOI_2_HOM_EXT[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                        + TI_2_PRS_HOM_SP[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                                    * FAC_N)) / FAC_N
            (2)ENDIF
            FAC_PRS_COR_DAMP_2_HOM = IP_FAC_PRS_COR_DAMP_2_HOM(
                Inputs: T_DLY_1_2_HOM; FUP_H_INJ[m])
            Calculate period time of pressure waves:
```

```
            (3)IF C_PRS_COR_TEMP_SEL = 0
            (3)THEN
                TEMP_PRS_COR_2_HOM = TFU_INJ
            (3)ELSE
                (4) IF C_PRS_COR_TEMP_SEL = 1
                (4)THEN
                    TEMP_PRS_COR_2_HOM =
                            TEMP_CAPA_IV_MV
                (4)ELSE
                    TEMP_PRS_COR_2_HOM = TCO
                (4)ENDIF
            (3)ENDIF
            T_PER_PRS_COR_2_HOM = IP_T_PER_PRS_COR_2_HOM(
                Inputs: TEMP_PRS_COR_2_HOM; FUP_H_INJ[m])
            Calculate Time offset within one wave:
            i = 0 // i indicates the number of while loops
            (1)WHILE T_DLY_1_2_HOM > T_PER_PRS_COR_2_HOM DO
                T_DLY_1_2_HOM = T_DLY_1_2_HOM
                  - (T_PER_PRS_COR_2_HOM
                       * ID_T_PER_PRS_COR_2_HOM(Input: i))
                Note: In case of i > 4 use 4 as input for ID map to prevent overflow
                i = i + 1
                Note: To prevent an endless loop, break the while-loop if i reaches 50.
                In that case, set T_DLY_1_2_HOM to 0!
            (1)ENDWHILE
            Calculate normalized Time offset within one wave:
            RATIO_T_PER_PRS_COR_2_HOM =
                T_DLY_1_2_HOM/
                (T_PER_PRS_COR_2_HOM * ID_T_PER_PRS_COR_2_HOM(Input: i))
            RATIO_TI_2_HOM =
              TI_2_PRS_HOM_SP[IDX_PRS_COR_CYL_CLC_MPLH[m]]/
              (T_PER_PRS_COR_2_HOM * ID_T_PER_PRS_COR_2_HOM(Input: i))
            Calculate correction due to wave position and injection time:
            FAC_PRS_COR_OSC_2_HOM = IP_FAC_PRS_COR_OSC_2_HOM(
              Inputs: RATIO_T_PER_PRS_COR_2_HOM; RATIO_TI_2_HOM)
            Calculate final correction factor:
            FAC_TI_PRS_COR_2_HOM[m] = 1 + (FAC_PRS_COR_MFF_2_HOM
                * FAC_PRS_COR_DAMP_2_HOM * FAC_PRS_COR_OSC_2_HOM)
        (2a)ENDIF
    (2a)ENDFOR
(1)ELSE
    (2b)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
```

```
                FAC_TI_PRS_COR_2_HOM[m] = 1
        (2b)ENDFOR
    (1)ENDIF

    Calculation of injection time for homogeneous second pulse:
    (3)FOR cylinder_start to cylinder_stop DO
    (1)IF INJ_MOD_SP_HOM[x] = "SNGH"  (if single injection is requested)
    (1) THEN
              TI_2_HOM[x] = 0
              IDX_TI_2_HOM_CLC[x] = 0
    (1) ELSE (if multiple injection is requested)
    MFF_SP_2_HOM_COR_INJ[x] = MFF_SP_2_HOM[x] *(1+ FAC_MFF_DIF_HOM[x])
                                                     * FAC_MFF_TFU
    (4) IF   LV_EGY_RNG_IV_PLS_2_HOM[x] = 1
    (4) THEN
         IDX_TI_2_STND_HOM[x] = IP_MFF_IDX_TI_EGY_H(MFF_SP_2_HOM_COR_INJ[x];
                                                    PRS_DEC_INJ_2_HOM[x])
    (4) ELSE
         IDX_TI_2_STND_HOM[x] = IP_MFF_IDX_TI_EGY_L(MFF_SP_2_HOM_COR_INJ[x];
                                                    PRS_DEC_INJ_2_HOM[x])
                               * FAC_TI_COR_IV_EGY_RNG_L[NC_IDX_CYL_HPP_REF[x]]
    (4) ENDIF

    IDX_TI_2_HOM_CLC[x] =          IDX_TI_2_STND_HOM[x]
                             * FAC_TI_2_PRS_CYL_HOM[x]
                             * FAC_TI_PRS_COR_2_HOM[NC_IDX_CYL_HPP_REF[x]]
                             * TI_FAC[x]
                             * TI_TUN_IV[x]
    Calculate_Pulse_timing_parameters(IN <LV_EGY_RNG_IV_PLS_2_HOM[x]>,
         IN <IDX_TI_2_HOM_CLC[x]>, OUT <T_CHA_PER_HOM_2[x]>,
         OUT <T_DCHA_PER_HOM_2_1[x]>, OUT <T_DCHA_PER_HOM_2_2[x]>,
         OUT <T_DCHA_DLY_HOM_2[x]>, OUT <TI_2_HOM_CLC[x]>)


    (7) IF LC_TI_2_HOM_MAN_ACT = 1
    (7) THEN
         TI_2_HOM[x]  = C_TI_2_HOM_MAN
         T_CHA_PER_HOM_2[x] = C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
         T_DCHA_PER_HOM_2_1[x] = C_T_DCHA_PER_1_H[NC_NR_IDX_TI_PRM_H – 1]
         T_DCHA_DLY_HOM_2[x] = C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H – 1]
         T_DCHA_PER_HOM_2_2[x] = T_CHA_PER_HOM_2[x] – T_DCHA_PER_HOM_2_1[x]
    (7) ELSE
         TI_2_HOM[x] = TI_2_HOM_CLC[x]
```

```
(7) ENDIF
```

Note: For calibration purposes the injection time TI_2_HOM[x] can be set manually to the value C_TI_2_HOM_MAN

```
(1) ENDIF
(3)ENDFOR
IDX_TI_2_HOM_MIN = MIN(IDX_TI_2_HOM_CLC) // Minimum of all NC_CYL_NR cylinders
(8)IF IDX_TI_2_HOM_MIN < C_IDX_TI_MIN_H
   AND (STATE_INJ_MOD_HOM_REQ = "MPLH"
        OR STATE_INJ_MOD_HOM_REQ = "MPLH+PLS3")
(8)THEN
   LV_TI_2_HOM_MIN = 1
(8)ELSE
   LV_TI_2_HOM_MIN = 0
(8)ENDIF
TI_2_HOM_CLC_MIN  = MIN(TI_2_HOM_CLC[ ]) // Minimum of all NC_CYL_NR cylinders
TI_2_HOM_CLC_MAX  = MAX(TI_2_HOM_CLC[ ]) // Maximum of all NC_CYL_NR cylinders
```

41.3.1.1.4 Third pulse

Correction of fuel pressure pulsations for homogeneous multi injection mode:

```
(1)IF STATE_INJ_MOD_HOM_REQ = "MPLH+PLS3"
   AND LC_PRS_COR_MPLH_ENA = 1
   AND LV_PRS_COR_MPLH_ENA = 1
(1)THEN
   (3a)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
      (3a)IF Bit m of LF_PRS_COR_HPP_ENA = 1
      (3a)THEN
         Correction due to MFF_SP_2_HOM:
         FAC_PRS_COR_MFF_3_HOM = IP_FAC_PRS_COR_MFF_3_HOM(
                Input: MFF_SP_2_HOM_COR_INJ[IDX_PRS_COR_CYL_CLC_MPLH[m]])
         Correction of damping depending on FUP and delay time between pulses:
         (2)IF LC_PRS_COR_SOI_DLY_ENA = 1
         (2)THEN
                T_DLY_2_3_HOM =
                   (EOI_2_HOM_EXT[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                   + (TI_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                   + T_DCHA_DLY_HOM_2[IDX_PRS_COR_CYL_CLC_MPLH[m]])
                          * FAC_N
                   - (EOI_3_HOM_EXT[IDX_PRS_COR_CYL_CLC_MPLH[m]]
```

```
                                + TI_3_PRS_HOM_SP[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                                                    * FAC_N)) / FAC_N

                (2)ELSE

                     T_DLY_2_3_HOM =
                       (EOI_2_HOM_EXT[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                       - (EOI_3_HOM_EXT[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                       + TI_3_PRS_HOM_SP[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                                                    * FAC_N)) / FAC_N
                (2)ENDIF

                FAC_PRS_COR_DAMP_3_HOM = IP_FAC_PRS_COR_DAMP_3_HOM(
                  Inputs: T_DLY_2_3_HOM; FUP_H_INJ[m])

                Calculate period time of pressure waves:

                T_PER_PRS_COR_3_HOM = IP_T_PER_PRS_COR_3_HOM(
                  Inputs: TEMP_PRS_COR_2_HOM; FUP_H_INJ[m])

                Calculate Time offset within one wave:

                i = 0 // i indicates the number of while loops
                (1)WHILE T_DLY_2_3_HOM > T_PER_PRS_COR_3_HOM DO

                  T_DLY_2_3_HOM = T_DLY_2_3_HOM
                       - (T_PER_PRS_COR_3_HOM * ID_T_PER_PRS_COR_3_HOM(Input: i))
                       Note: In case of i > 4 use 4 as input for ID map to prevent overflow

                       i = i + 1

                       Note: To prevent an endless loop, break the while-loop if i reaches 50.
                       In that case, set T_DLY_2_3_HOM to 0!

                (1)ENDWHILE

                Calculate normalized Time offset within one wave:

                RATIO_T_PER_PRS_COR_3_HOM =
                  T_DLY_2_3_HOM/(T_PER_PRS_COR_3_HOM
                                                    * ID_T_PER_PRS_COR_3_HOM(Input: i))

                RATIO_TI_3_HOM = TI_3_PRS_HOM_SP[IDX_PRS_COR_CYL_CLC_MPLH[m]]/
                  (T_PER_PRS_COR_3_HOM * ID_T_PER_PRS_COR_3_HOM(Input: i))

                Calculate correction due to wave position and injection time:

                FAC_PRS_COR_OSC_3_HOM = IP_FAC_PRS_COR_OSC_3_HOM(
                     Inputs: RATIO_T_PER_PRS_COR_3_HOM; RATIO_TI_3_HOM)

                Calculate final correction factor:

                FAC_TI_PRS_COR_3_HOM[m] = 1 + (FAC_PRS_COR_MFF_3_HOM
                       * FAC_PRS_COR_DAMP_3_HOM * FAC_PRS_COR_OSC_3_HOM)
              (3a)ENDIF

            (3a)ENDFOR

         (1)ELSE

            (3b)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO

                  FAC_TI_PRS_COR_3_HOM[m] = 1

            (3b)ENDFOR
```

```
(1)ENDIF

(3)FOR cylinder_start to cylinder_stop DO
(1)IF INJ_MOD_SP_HOM[x] ≠ "MPLH+PLS3"    (if third injection is not requested)
(1) THEN
            TI_3_HOM[x] = 0
            IDX_TI_3_HOM_CLC[x] = 0
(1) ELSE (if third injection is requested)
MFF_SP_3_HOM_COR_INJ[x] = MFF_SP_3_HOM[x] *(1+ FAC_MFF_DIF_HOM[x])
                                                      * FAC_MFF_TFU
(4) IF   LV_EGY_RNG_IV_PLS_3_HOM[x] = 1
(4) THEN
      IDX_TI_3_STND_HOM[x] = IP_MFF_IDX_TI_EGY_H(MFF_SP_3_HOM_COR_INJ[x];
                                             PRS_DEC_INJ_3_HOM[x])
(4) ELSE
      IDX_TI_3_STND_HOM[x] = IP_MFF_IDX_TI_EGY_L(MFF_SP_3_HOM_COR_INJ[x];
                                             PRS_DEC_INJ_3_HOM[x])
                         * FAC_TI_COR_IV_EGY_RNG_L[NC_IDX_CYL_HPP_REF[x]]
(4) ENDIF

IDX_TI_3_HOM_CLC[x] =           IDX_TI_3_STND_HOM[x]
                    * FAC_TI_3_PRS_CYL_HOM[x]
                    * FAC_TI_PRS_COR_3_HOM[NC_IDX_CYL_HPP_REF[x]]
                    * TI_FAC[x]
                    * TI_TUN_IV[x]


Calculate_Pulse_timing_parameters(IN <LV_EGY_RNG_IV_PLS_3_HOM[x]>,
      IN <IDX_TI_3_HOM_CLC[x]>, OUT <T_CHA_PER_HOM_3[x]>,
      OUT <T_DCHA_PER_HOM_3_1[x]>, OUT <T_DCHA_PER_HOM_3_2[x]>,
      OUT <T_DCHA_DLY_HOM_3[x]>, OUT <TI_3_HOM_CLC[x]>)


(7) IF LC_TI_3_HOM_MAN_ACT = 1
(7) THEN
      TI_3_HOM[x]  = C_TI_3_HOM_MAN
      T_CHA_PER_HOM_3[x] = C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
      T_DCHA_PER_HOM_3_1[x] = C_T_DCHA_PER_1_H[NC_NR_IDX_TI_PRM_H – 1]
      T_DCHA_DLY_HOM_3[x] = C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H – 1]
      T_DCHA_PER_HOM_3_2[x] = T_CHA_PER_HOM_3[x] – T_DCHA_PER_HOM_3_1[x]
(7) ELSE
      TI_3_HOM[x] = TI_3_HOM_CLC[x]
```

```
(7) ENDIF
```

Note: For calibration purposes the injection time TI_3_HOM[x] can be set manually to the value C_TI_3_HOM_MAN

```
(1) ENDIF
(3)ENDFOR
IDX_TI_3_HOM_MIN = MIN(IDX_TI_3_HOM_CLC) // Minimum of all NC_CYL_NR cylinders
(8)IF IDX_TI_3_HOM_MIN < C_IDX_TI_MIN_H
     AND STATE_INJ_MOD_HOM_REQ = "MPLH+PLS3"
(8)THEN
     LV_TI_3_HOM_MIN = 1
(8)ELSE
     LV_TI_3_HOM_MIN = 0
(8)ENDIF
TI_3_HOM_CLC_MIN  = MIN(TI_3_HOM_CLC[ ]) // Minimum of all NC_CYL_NR cylinders
TI_3_HOM_CLC_MAX  = MAX(TI_3_HOM_CLC[ ]) // Maximum of all NC_CYL_NR cylinders
```

41.3.1.2 Calculation of the cylinder individual injection times for stratified mode and homogeneous-stratified mode

```
(1) IF LV_S_CLC = 1            (Note: this part is only calculated, if stratified mode or
    homogeneous-stratified mode is enabled)

(1) THEN
```

41.3.1.2.1 First pulse

```
  (1)FOR cylinder_start to cylinder_stop DO

  MFF_SP_1_S_COR_INJ[x] = MFF_SP_1_S[x]*(1 + FAC_MFF_DIF_S[x]) * FAC_MFF_TFU

  (1a) IF LV_EGY_RNG_IV_PLS_1_S[x] = 1
  (1a) THEN (high injector needle lift is selected for the first pulse)

           IDX_TI_1_S_CLC[x] = IP_MFF_IDX_TI_EGY_H(MFF_SP_1_S_COR_INJ[x];
                                         PRS_DEC_INJ_1_S[x])
                               * TI_FAC[x]
                               * TI_TUN_IV[x]
                               * FAC_TI_1_PRS_CYL_S[x]
                               + (TI_TUN_ADD_IV[x] / 0.001ms)
                               + (TI_ADD[x] / 0.001ms)
  (1a) ELSE (low injector needle lift is selected for the first pulse)

           IDX_TI_1_S_CLC[x] = IP_MFF_IDX_TI_EGY_L(MFF_SP_1_S_COR_INJ[x];
                                         PRS_DEC_INJ_1_S[x])
                          * FAC_TI_COR_IV_EGY_RNG_L[NC_IDX_CYL_HPP_REF[x]]
                               * TI_FAC[x]
                               * TI_TUN_IV[x]
                               * FAC_TI_1_PRS_CYL_S[x]
                               + (TI_TUN_ADD_IV[x] / 0.001ms)
                               + (TI_ADD[x] / 0.001ms)
  (1a) ENDIF

  Calculate_Pulse_timing_parameters(IN <LV_EGY_RNG_IV_PLS_1_S[x]>,
     IN <IDX_TI_1_S_CLC[x]>, OUT <T_CHA_PER_S_1[x]>,
     OUT <T_DCHA_PER_S_1_1[x]>, OUT <T_DCHA_PER_S_1_2[x]>,
     OUT <T_DCHA_DLY_S_1[x]>, OUT <TI_1_S_CLC[x]>)


  (1)ENDFOR

  TI_1_S_CLC_MIN = MIN(TI_1_S_CLC[ ]) // Minimum of all NC_CYL_NR cylinders

  IDX_TI_1_S_MIN = MIN(IDX_TI_1_S_CLC) // Minimum of all NC_CYL_NR cylinders

  (2) IF

  LC_TI_1_S_MAN_ACT = 1

  (2) THEN

     (2)FOR cylinder_start to cylinder_stop DO

           TI_1_S[x]  = C_TI_1_S_MAN
           T_CHA_PER_S_1[x] = C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
           T_DCHA_PER_S_1_1[x] =
                   C_T_DCHA_PER_1_H[NC_NR_IDX_TI_PRM_H – 1]
```

```
            T_DCHA_DLY_S_1[x] = C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H – 1]
            T_DCHA_PER_S_1_2[x] =
                    T_CHA_PER_S_1[x] – T_DCHA_PER_S_1_1[x]
        (2)ENDFOR
        LV_AUTH_TI_MIN_S = 1
        LV_TI_1_S_MIN = 0
    (2) ELSE
        (3)FOR cylinder_start to cylinder_stop DO
            TI_1_S[x] = TI_1_S_CLC[x]
        (3)ENDFOR
        LV_AUTH_TI_MIN_S = 1
        (4)IF IDX_TI_1_S_MIN < C_IDX_TI_MIN_H
        (4)THEN
            LV_TI_1_S_MIN = 1
        (4)ELSE
            LV_TI_1_S_MIN = 0
        (4)ENDIF
    (2) ENDIF
```

Note: For calibration purposes the injection time TI_1_S[x] can be set manually to the value C_TI_1_S_MAN

41.3.1.2.2 Second pulse

```
(6) IF (STATE_INJ_MOD_REQ = 'HOMS' OR STATE_INJ_MOD_REQ = 'HOMS+PLS3')
    OR LC_PRS_COR_MPLS_ENA = 0
(6) THEN
    (4)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
        FAC_TI_2_S_DLY_COR[m] = 1
        FAC_TI_2_S_COR[m] = 1
        FAC_TI_PRS_COR_2_S[m] = 1
    (4)ENDFOR
(6) ELSE
    (4a)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
        FAC_TI_2_S_DLY_COR[m] = IP_FAC_TI_2_S_DLY_COR (
          Input: T_DLY_1_2_S_EXT[IDX_PRS_COR_CYL_CLC_S[m]],
                 MFF_SP_1_S[IDX_PRS_COR_CYL_CLC_S[m]])

        FAC_TI_2_S_COR[m] = IP_FAC_TI_2_S_COR (
          Input: TI_2_STND_S[IDX_PRS_COR_CYL_CLC_S[m]];
                 T_DLY_1_2_S_EXT[IDX_PRS_COR_CYL_CLC_S[m]])
        FAC_TI_PRS_COR_2_S[m] = FAC_TI_2_S_DLY_COR[m]
                                    * FAC_TI_2_S_COR[m]
```

This correction factor is for correction of the mass flow difference between the mass flow sum of two pulses in comparison with the same requested mass for one pulse. The difference is caused by physical effects of the piezo injector.

```
        (4a)ENDFOR
    (6) ENDIF


    (5)FOR cylinder_start to cylinder_stop DO
    (5)IF INJ_MOD_SP_S[x] = "SNGS"
        AND INJ_MOD_SP_HOMS[x] = "HOMS-SNG"
    (5)THEN
        TI_2_S[x] = 0
        IDX_TI_2_S_CLC[x] = 0
    (5)ELSE
    MFF_SP_2_S_COR_INJ[x] = MFF_SP_2_S[x]*(1 + FAC_MFF_DIF_S[x]) * FAC_MFF_TFU
    (2a) IF LV_EGY_RNG_IV_PLS_2_S[x] = 1
    (2a) THEN (high injector needle lift is selected for the second pulse)
                IDX_TI_2_S_CLC[x] = IP_MFF_IDX_TI_EGY_H(MFF_SP_2_S_COR_INJ[x];
                                        PRS_DEC_INJ_2_S[x])
                              * TI_FAC[x]
                              * TI_TUN_IV[x]
                              * FAC_TI_2_PRS_CYL_S[x]
                              * FAC_TI_PRS_COR_2_S[NC_IDX_CYL_HPP_REF[x]]
    (2a) ELSE (low injector needle lift is selected for the second pulse)
                IDX_TI_2_S_CLC[x] = IP_MFF_IDX_TI_EGY_L(MFF_SP_2_S_COR_INJ[x];
                                        PRS_DEC_INJ_2_S[x])
                         * FAC_TI_COR_IV_EGY_RNG_L[NC_IDX_CYL_HPP_REF[x]]
                              * TI_FAC[x]
                              * TI_TUN_IV[x]
                              * FAC_TI_2_PRS_CYL_S[x]
                              * FAC_TI_PRS_COR_2_S[NC_IDX_CYL_HPP_REF[x]]
    (2a) ENDIF


    Calculate_Pulse_timing_parameters(IN <LV_EGY_RNG_IV_PLS_2_S[x]>,
        IN <IDX_TI_2_S_CLC[x]>, OUT <T_CHA_PER_S_2[x]>,
        OUT <T_DCHA_PER_S_2_1[x]>, OUT <T_DCHA_PER_S_2_2[x]>,
        OUT <T_DCHA_DLY_S_2[x]>, OUT <TI_2_S_CLC[x]>)


    (7) IF
    LC_TI_2_S_MAN_ACT = 1
    (7) THEN
        TI_2_S[x]  = C_TI_2_S_MAN
        T_CHA_PER_S_2[x] = C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
        T_DCHA_PER_S_2_1[x] = C_T_DCHA_PER_1_H[NC_NR_IDX_TI_PRM_H – 1]
```

```
        T_DCHA_DLY_S_2[x] = C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H – 1]
        T_DCHA_PER_S_2_2[x] = T_CHA_PER_S_2[x] – T_DCHA_PER_S_2_1[x]
    (7) ELSE
        TI_2_S[x] = TI_2_S_CLC[x]
    (7) ENDIF
    Note:  For calibration purposes the injection time TI_2_S[x] can be set manually to the value
           C_TI_2_S_MAN
    (5) ENDIF
    (5)ENDFOR
    IDX_TI_2_S_MIN = MIN(IDX_TI_2_S_CLC) // Minimum of all NC_CYL_NR cylinders
    (4)IF IDX_TI_2_S_MIN < C_IDX_TI_MIN_H
        AND (STATE_INJ_MOD_S_REQ = "MPLS"
                OR STATE_INJ_MOD_S_REQ = "MPLS+PLS3")
    (4)THEN
        LV_TI_2_S_MIN = 1
    (4)ELSE
        LV_TI_2_S_MIN = 0
    (4)ENDIF
    TI_2_S_CLC_MIN = MIN (TI_2_S_CLC[x]) // Minimum of all NC_CYL_NR cylinders
```

41.3.1.2.3 Third pulse

```
        (10) IF STATE_INJ_MOD_REQ = 'HOMS+PLS3'
                OR LC_PRS_COR_MPLS_ENA = 0
        (10) THEN
                (6)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
                        FAC_TI_3_S_DLY_COR[m] = 1
                        FAC_TI_3_S_COR[m] = 1
                        FAC_TI_PRS_COR_3_S[m] = 1
                (6)ENDFOR
        (10) ELSE
                (6a)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
                     FAC_TI_3_S_DLY_COR[m] = IP_FAC_TI_3_S_DLY_COR
                                                             (Input:
    T_DLY_2_3_S_EXT[IDX_PRS_COR_CYL_CLC_S[m]],
                        MFF_SP_2_S[IDX_PRS_COR_CYL_CLC_S[m]])

                    FAC_TI_3_S_COR[m] = IP_FAC_TI_3_S_COR
                        (Input: TI_3_STND_S[IDX_PRS_COR_CYL_CLC_S[m]];
                        T_DLY_2_3_S_EXT[IDX_PRS_COR_CYL_CLC_S[m]])
                    FAC_TI_PRS_COR_3_S[m] = FAC_TI_3_S_DLY_COR[m]
```

```
                                                * FAC_TI_3_S_COR[m]
```

This correction factor is for correction of the mass flow difference between the mass flow sum of two pulses in comparison with the same requested mass for one pulse. The difference is caused by physical effects of the piezo injector.

```
    (6a)ENDFOR
(10) ENDIF

(8)FOR cylinder_start to cylinder_stop DO
(9)IF INJ_MOD_SP_S[x] = "MPLS+PLS3"
        OR INJ_MOD_SP_HOMS[x] = "HOMS+PLS3"
(9)THEN
MFF_SP_3_S_COR_INJ[x] = MFF_SP_3_S[x]*(1 + FAC_MFF_DIF_S[x])
                                                * FAC_MFF_TFU
(3a) IF LV_EGY_RNG_IV_PLS_3_S[x] = 1
(3a) THEN (high injector needle lift is selected for the third pulse)
    IDX_TI_3_S_CLC[x] = IP_MFF_IDX_TI_EGY_H(MFF_SP_3_S_COR_INJ[x];
                                        PRS_DEC_INJ_3_S[x])
                        * TI_FAC[x]
                        * TI_TUN_IV[x]
                        * FAC_TI_3_PRS_CYL_S[x]
                        * FAC_TI_PRS_COR_3_S[NC_IDX_CYL_HPP_REF[x]]
(3a) ELSE (low injector needle lift is selected for the third pulse)
        IDX_TI_3_S_CLC[x] = IP_MFF_IDX_TI_EGY_L(MFF_SP_3_S_COR_INJ[x];
                                        PRS_DEC_INJ_3_S[x])
                    * FAC_TI_COR_IV_EGY_RNG_L[NC_IDX_CYL_HPP_REF[x]]
                        * TI_FAC[x]
                        * TI_TUN_IV[x]
                        * FAC_TI_3_PRS_CYL_S[x]
                        * FAC_TI_PRS_COR_3_S[NC_IDX_CYL_HPP_REF[x]]
(3a) ENDIF


Calculate_Pulse_timing_parameters(IN <LV_EGY_RNG_IV_PLS_3_S[x]>,
   IN <IDX_TI_3_S_CLC[x]>, OUT <T_CHA_PER_S_3[x]>,
   OUT <T_DCHA_PER_S_3_1[x]>, OUT <T_DCHA_PER_S_3_2[x]>,
   OUT <T_DCHA_DLY_S_3[x]>, OUT <TI_3_S_CLC[x]>)


(11) IF LC_TI_3_S_MAN_ACT = 1
(11) THEN
        TI_3_S[x]  = C_TI_3_S_MAN
        T_CHA_PER_S_3[x] = C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
        T_DCHA_PER_S_3_1[x] = C_T_DCHA_PER_1_H[NC_NR_IDX_TI_PRM_H – 1]
        T_DCHA_DLY_S_3[x] = C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H – 1]
        T_DCHA_PER_S_3_2[x] = T_CHA_PER_S_3[x] – T_DCHA_PER_S_3_1[x]
(11) ELSE
```

```
                TI_3_S[x] = TI_3_S_CLC[x]
            (11) ENDIF
            Note: For calibration purposes the injection time TI_3_S[x] can be set manually to the
                  value C_TI_3_S_MAN

        (9) ELSE (No third pulse in stratified mode or homogeneous-stratified mode is required)
            TI_3_S[x]  = 0
            IDX_TI_3_S_CLC[x] = 0
        (9) ENDIF
        (8)ENDFOR
        IDX_TI_3_S_MIN = MIN(IDX_TI_3_S_CLC) // Minimum of all NC_CYL_NR cylinders
        (4)IF IDX_TI_3_S_MIN < C_IDX_TI_MIN_H
            AND STATE_INJ_MOD_S_REQ = "MPLS+PLS3"
        (4)THEN
            LV_TI_3_S_MIN = 1
        (4)ELSE
            LV_TI_3_S_MIN = 0
        (4)ENDIF
        TI_3_S_CLC_MIN = MIN (TI_3_S_CLC[x]) // Minimum of all NC_CYL_NR cylinders

    (1) ELSE – LV_S_CLC
            (12)FOR cylinder_start to cylinder_stop DO
                TI_1_S[x]  = 0, TI_2_S[x]  = 0, TI_3_S[x] = 0
            (12)ENDFOR
            LV_AUTH_TI_MIN_S = 0
            LV_TI_1_S_MIN = 0, LV_TI_2_S_MIN = 0, LV_TI_3_S_MIN = 0
    (1) ENDIF
```

This section calculates the charging and discharging time for the post injection pulse.

```
(2)IF LV_IV_POST_EGY_RNG = 1
(2)THEN
        T_CHA_PER_POST = C_T_CHA_PER_H[NC_NR_IDX_TI_PRM_H – 1]
(2)ELSE
        T_CHA_PER_POST = C_T_CHA_PER_L[NC_NR_IDX_TI_PRM_L – 1]
(2)ENDIF
T_DCHA_PER_POST = T_CHA_PER_POST
```

41.3.2 Calculate interface to monitoring unit:

```
   (1)IF STATE_INJ_MOD_REQ BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_HOM
   (1)THEN
        FAC_TI_PRS_COR_1 = MEAN(FAC_TI_PRS_COR_1_HOM[ ])
                                             // mean value of all elements
        FAC_TI_PRS_COR_2 = MEAN(FAC_TI_PRS_COR_2_HOM[ ])
                                             // mean value of all elements
        FAC_TI_PRS_COR_3 = MEAN(FAC_TI_PRS_COR_3_HOM[ ])
                                             // mean value of all elements
(1)ELSE
        FAC_TI_PRS_COR_1 = 1
        FAC_TI_PRS_COR_2 = MEAN(FAC_TI_PRS_COR_2_S[ ])
                                             // mean value of all elements
        FAC_TI_PRS_COR_3 = MEAN(FAC_TI_PRS_COR_3_S[ ])
                                             // mean value of all elements
(1)ENDIF
```

41.3.2.1 Subroutines

This chapter describes the sub functions which are called from the main sections.

41.3.2.1.1 Calculate Pulse timing parameters

This section calculates the pulse timing parameter (Charge Time, TI, Discharge Time 1, Discharge Time 2 and Discharge Delay) for the given counting factor index and needle lift.

Input parameters: PRM_LV_EGY_RNG_IV, PRM_IDX_TI_CLC

Output parameters: PRM_T_CHA_PER, PRM_T_DCHA_PER_1, PRM_T_DCHA_PER_2, PRM_T_DCHA_DLY, PRM_TI

Parameter description:

PRM_LV_EGY_RNG_IV	IN	0...1H	0...1	1	[-]
Injector needle lift.					
PRM_IDX_TI_CLC	IN	0...FFFFH	0...65535	1	[-]
Counting factor index					
PRM_T_CHA_PER	OUT	0...12CH	0...0.3	0.001	[ms]
Injector charge periode					
PRM_T_DCHA_PER_1	OUT	0...12CH	0...0.3	0.001	[ms]
First injector discharge periode					
PRM_T_DCHA_PER_2	OUT	0...12CH	0...0.3	0.001	[ms]
Second injector discharge periode					
PRM_T_DCHA_DLY	OUT	0...FFFFH	0...65.535	0.001	[ms]
Delay time beweend first and second injector discharge phase					
PRM_TI	OUT	0...FFFFH	0...65.535	0.001	[ms]
Injection time					

Calculate_Pulse_timing_parameters(IN <PRM_LV_EGY_RNG_IV>, IN <PRM_IDX_TI_CLC>, OUT <PRM_T_CHA_PER>, OUT <PRM_T_DCHA_PER_1>, OUT <PRM_T_DCHA_PER_2>, OUT <PRM_T_DCHA_DLY>, OUT <PRM_TI>)

```
(1)IF PRM_LV_EGY_RNG_IV = 0
     AND PRM_IDX_TI_CLC >= C_IDX_TI_MIN_L
(1)THEN – low injector needle lift with special calculation

     IDX_TI_TMP = MIN( (PRM_IDX_TI_CLC – C_IDX_TI_MIN_L) ;
                                             (NC_NR_IDX_TI_PRM_L – 1) )

     PRM_T_CHA_PER = C_T_CHA_PER_L[IDX_TI_TMP]
     PRM_T_DCHA_DLY = C_T_DCHA_DLY_L[IDX_TI_TMP]
     (2a)IF PRM_T_DCHA_DLY < ~~0.03ms~~ NC_T_DCHA_DLY_MIN
     (2a)THEN

        PRM_T_DCHA_PER_1 = 0
        PRM_T_DCHA_DLY = 0

     (2a)ELSE

        PRM_T_DCHA_PER_1 = C_T_DCHA_PER_1_L[IDX_TI_TMP]

     (2a)ENDIF

     (2)IF IDX_TI_TMP >= NC_NR_IDX_TI_PRM_L – 1
     (2)THEN

        PRM_TI = PRM_IDX_TI_CLC * 0.001ms + C_TI_OFS_L

     (2)ELSE
```

```
            PRM_TI = C_TI_L[IDX_TI_TMP]

        (2)ENDIF

    (1)ELSE – high injector needle lift or low lift identical to high lift

        IDX_TI_TMP = MIN( (PRM_IDX_TI_CLC – C_IDX_TI_MIN_H) ;
                                                   (NC_NR_IDX_TI_PRM_H – 1) )
                                          // limit calculation result to zero

        PRM_T_CHA_PER = C_T_CHA_PER_H[IDX_TI_TMP]
        PRM_T_DCHA_DLY = C_T_DCHA_DLY_H[IDX_TI_TMP]
        (3a)IF PRM_T_DCHA_DLY < ~~0.03ms~~ NC_T_DCHA_DLY_MIN
        (3a)THEN

            PRM_T_DCHA_PER_1 = 0
            PRM_T_DCHA_DLY = 0

        (3a)ELSE

            PRM_T_DCHA_PER_1 = C_T_DCHA_PER_1_H[IDX_TI_TMP]

        (3a)ENDIF

        (3)IF IDX_TI_TMP >= NC_NR_IDX_TI_PRM_H – 1
        (3)THEN

            PRM_TI = PRM_IDX_TI_CLC * 0.001ms + C_TI_OFS_H

        (3)ELSE

            PRM_TI = C_TI_H[IDX_TI_TMP]

        (3)ENDIF

    (1)ENDIF

    PRM_T_DCHA_PER_2 = PRM_T_CHA_PER – PRM_T_DCHA_PER_1

End of Calculate_Pulse_timing_parameters
```

41.3.3 Time synchronous tasks

Application conditions:

Activation: every engine state

Deactivation: -

Initialization: at reset: LV_AUTH_TI_MIN_AFL = 1

Recurrence: 10 ms

Calculation of LV_AUTH_TI_MIN_AFL:

If homogeneous lean mode is active (STATE_CMB_CTL = HOM_AFL) and the electrical injection time is smaller than (TI_MIN_HOM + C_TI_MIN_OFS), the combustion mode changes to homogeneous stochiometric mode. Switching back to homogeneous lean mode is allowed at the earliest if the delay time C_ T_AUTH_TI_MIN_AFL has elapsed.

```
(1) IF
LV_AFL_CLC = 0
(1) THEN
LV_AUTH_TI_MIN_AFL = 1
T_AUTH_TI_MIN_AFL = 0
(1) ELSE
  (2) IF
  LV_AUTH_TI_MIN_AFL = 1
  AND
  STATE_CMB_CTL = 'HOM_AFL'       (homogeneous lean mode)
  AND
  TI_1_HOM_CLC_MIN < (TI_MIN_HOM + C_TI_MIN_OFS)
  (2) THEN
  LV_AUTH_TI_MIN_AFL = 0
  T_AUTH_TI_MIN_AFL = C_ T_AUTH_TI_MIN_AFL
  (2) ELSE
    (3) IF
    T_AUTH_TI_MIN_AFL (n) > 0.01s (0H)
    (3) THEN
    T_AUTH_TI_MIN_AFL (n+1) = T_AUTH_TI_MIN_AFL (n) – 0.01 [s]
    (3) ELSE
    LV_AUTH_TI_MIN_AFL = 1
    (3) ENDIF
```

```
        (2) ENDIF
    (1) ENDIF
```

Output variable for LACO:

```
(4) IF (STATE_INJ_MOD_REQ = 'SNGH') OR (STATE_INJ_MOD_REQ = 'MPLH')
        OR (STATE_INJ_MOD_REQ = 'MPLH+PLS3')
(4) THEN
CRK_DIF_SOI_IGN = CRK_DIF_SOI_IGN_HOM
 (4) ELSE
CRK_DIF_SOI_IGN = CRK_DIF_SOI_IGN_S
(4) ENDIF
```

Application hint:

The input value TI_FAC[x] and TI_ADD[x] are the output value of the module “Coordination of Injection Time Correction Factors for Cylinder Balancing ”. If the Cylinder Balancing Function is not used than TI_FAC[x] and TI_ADD[x] have to be initialized in an “Initialization Module” as follows:

TI_FAC[x] =1

TI_ADD [x] = 0

41.4 Transfer to I/O SW for Piezo injectors

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
NR_CYL_INJ_BAS	O/V	0...7H	0...7	1	[-]
Number of the current cylinder, which has reached its CRK_INJ_BAS[x]-Event					
NR_CYL_INJ_BAS_PREV	V/O	0...7H	0...7	1	[-]
Number of the previous cylinder, which had reached its CRK_INJ_BAS[x]-Event					
INJ_MOD[NC_CYL_NR]	O/V	1H	SNGS	1	[-]
		2H	MPLS		
		3H	MPLS+PLS3		
		4H	PRE_INJ		
		6H	CRASH		
		7H	TEST_PULSE		
		21H	SNGS+MPLP		
		22H	MPLS+MPLP		
		41H	HOMS-SNG		
		42H	HOMS		
		43H	HOMS+PLS3		
		62H	HOMS+MPLP		
		80H	DISABLE		
		81H	SNGH		
		82H	MPLH		
		83H	MPLH+PLS3		
Cylinder individual injection mode					
LV_STATE_PREV_IV[NC_CYL_NR]	V/O	0...1H	0...1	1	[-]
Cylinder individuel flag which indicates whether last injection was activated or deactivated					
PREV_STATE_IV	V/O	0...FFH	0...255	1	[-]
Bit coded byte which indicates whether a cylinder was deactivated or not					
INH_IV_DYN	V/O	0...FFH	0...255	1	[-]
Shut off pattern for dynamic shut off (fixed cylinder allocation)					
LV_INH_INJ[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Flag that indicates if cylinder shut off is active or not					
LV_INH_INJ_OLD	-	0...1H	0...1	1	[-]
Flag that indicates if cylinder shut off was active or not. It contains the information from previous calculation run.					
CTR_PSN_INH_IV_DYN	-	0...FH	0...15	1	[-]
Position counter for dynamic cylinder shut off sequence calculation					
NR_CYL_INH_IV_DYN	-	0...7H	0...7	1	[-]
Destination logical cylinder number for dynamic fuel shut off					
NR_CYL_OFS_INH_IV_DYN	V	0...7H	0...7	1	[-]
Cylinder offset for destination of current cylinder number for dynamic fuel shut off					
TI_ADD_PULSE_MIN	V/O	0...FFFFH	0...65.535	0.001	[ms]
Minimum injection time for additive injection pulse					
FAC_ADD_PULSE	V/O	0...FFH	0...1.99218	0.0078125	[-]
Weighting factor for injection time update at transient conditions (additive pulse)					
LV_ST_INJ	V/O	0...1H	0...1	1	[-]
Flag which indicates that the injection has started after engine synchronization					
LV_INJ_MPLP_CYL[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Logical variable for enabling post injection, cylinder individual					
LV_IV_MES_VLD	V/O	0...1H	0...1	1	[-]
The flag indicates if the measurements for charge and voltage of the piezo injector is valid					
SOI_1_MES[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
Actual performed SOI of the first injection pulse, estimated					
SOI_2_MES[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
Actual performed SOI of the second injection pulse, estimated					
SOI_POST_MES[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
Actual performed SOI of the post injection pulse, estimated					
EOI_1_MES[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
Actual performed EOI of the first injection pulse, estimated					
EOI_2_MES[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
Actual performed EOI of the second injection pulse, estimated					
EOI_POST_MES[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
Actual performed EOI of the post injection pulse, estimated					
TI_1_MES[NC_CYL_NR]	V/O	0...FFFFH	0...65.535	0.001	[ms]
Actual performed cylinder individual injection time, first pulse					
TI_2_MES[NC_CYL_NR]	V/O	0...FFFFH	0...65.535	0.001	[ms]
Actual performed cylinder individual injection time, second pulse or additional pulse					
TI_POST_MES[NC_CYL_NR]	V/O	0...FFFFH	0...65.535	0.001	[ms]
Actual performed cylinder individual injection time, post injection pulse					
CHA_IV_1_MES[NC_CYL_NR]	V/O	0...3FFH	0...2272.6968	2.2216	[µAs]
Actual measured charge value of a carried out first pulse on the piezo injector, cylinder individual					
CHA_IV_2_MES[NC_CYL_NR]	V/O	0...3FFH	0...2272.6968	2.2216	[µAs]
Actual measured charge value of a carried out second pulse on the piezo injector, cylinder individual					
CHA_IV_POST_MES[NC_CYL_NR]	V/O	0...3FFH	0...2272.6968	2.2216	[µAs]
Actual measured charge value of a carried out post injection pulse on the piezo injector, cylinder individual					
V_IV_1_MES[NC_CYL_NR]	V/O	0...7FFFH	0...639.98046	0.0195313	[V]
Actual measured voltage of a carried out first pulse on the piezo injector, cylinder individual					
V_IV_2_MES[NC_CYL_NR]	V/O	0...7FFFH	0...639.98046	0.0195313	[V]
Actual measured voltage of a carried out second pulse on the piezo injector, cylinder individual					
V_IV_POST_MES[NC_CYL_NR]	V/O	0...7FFFH	0...639.98046	0.0195313	[V]
Actual measured voltage of a carried out post pulse on the piezo injector, cylinder individual					
CHA_IV_MES_BEG_CHA	-	0...3FFH	0...2272.6968	2.2216	[µAs]
Reference charge value before start of charge phase					
CHA_IV_MES_END_CHA	-	0...3FFH	0...2272.6968	2.2216	[µAs]
Remaining charge value after charge phase					
INH_IV_MIS_GEN_ACK	V/O	0...FFH	0...255	1	[-]
Shut off pattern for shut off caused by misfire generator					
CTR_INH_IV_MIS_GEN[NC_CYL_NR]	V/O	0...FFH	0...255	1	[-]
Counter of number of cycles where shut off caused by misfire generator was active					
CTR_TEST_MOD_IV[NC_CYL_NR]	V/O	0...FFH	0...255	1	[-]
Counter of number of injection valve actuator test cycles, cylinder individual					
STATE_INJ_CRASH_ACT	V	0...FFH	0...255	1	[-]
Injection driver state during vehicle crash request					
T_INJ_CRASH_ACT	V	0...FFFFH	0...655350	10	[ms]
Injection valve opening time during vehicle crash					
EGY_IV_1_CLC[NC_CYL_NR]	V/O	0...FFFFH	0...255	3.8911e-3	[mJ]
Energy inside injector calculated from measurement data for the first pulse					
EGY_IV_2_CLC[NC_CYL_NR]	V	0...FFFFH	0...255	3.8911e-3	[mJ]
Energy inside injector calculated from measurement data for the second pulse					
EGY_IV_POST_CLC[NC_CYL_NR]	V	0...FFFFH	0...255	3.8911e-3	[mJ]
Energy inside injector calculated from measurement data for the post injection pulse					
CAPA_IV_1_CLC[NC_CYL_NR]	V	0...FFFFH	0...27.96	0.4266e-3	[µF]
Capacity of the piezo injector for the first pulse, calculated from measurement data					
CAPA_IV_2_CLC[NC_CYL_NR]	V	0...FFFFH	0...27.96	0.4266e-3	[µF]
Capacity of the piezo injector for the second pulse, calculated from measurement data					
CAPA_IV_POST_CLC[NC_CYL_NR]	V	0...FFFFH	0...27.96	0.4266e-3	[µF]
Capacity of the piezo injector for the post injection pulse, calculated from measurement data					
EOI_INJ_UPD_PSN	-	0...780H	0...720	0.375	[°CRK]
End of injection position for autonomous generated update pulses					
STATE_ERR_IV_CYL_RAW[NC_CYL_NR]	V/O	0...FFH	0...255	1	[-]
Detailed error information of injection valve, cylinder individual I/O-SW raw value					

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CDN_DIAG_IV_RAW[NC_CYL_NR]	V/O	0...FFH	0...255	1	[-]
Detailed error condition information of injection valve, cylinder individual I/O-SW returned raw value					
STATE_ERR_PBK_IV_RAW[NC_PBK_IV_NR]	V/O	0...FFH	0...255	1	[-]
Detailed error information of injection valve power stages, ATIC bank individual I/O-SW raw value					
CDN_DIAG_PBK_IV_RAW[NC_PBK_IV_NR]	V/O	0...FFH	0...255	1	[-]
Detailed error condition information of injection valve power stages, ATIC bank individual I/O-SW returned raw value					
STATE_ERR_PBK_IV_ST_RAW[NC_PBK_IV_NR]	V/O	0...FFH	0...255	1	[-]
Detailed error information of injection valve power stages during startup phase, ATIC bank individual I/O-SW raw value					
LV_STATE_PREV_PBK_IV[NC_PBK_IV_NR]	V	0...1H	0...1	1	[-]
Injection valve power stage bank individual flag which indicates if the power stage bank was activated (=1) or deactivated during the last injection cycles					
CTR_STATE_PREV_PBK_IV[NC_PBK_IV_NR]	V	0...8H	0...8	1	[-]
Number of activated IV on the selected power stage bank during last engine cycle (720°CRK)					
VBOOST_PBK_IV_MES[NC_NR_DCDC_INJ]	V	0...7FFFH	0...639.98046	0.0195313	[V]
Actual measured voltage of the DCDC converter for the injection valve power stage banks					
STATE_PBK_IV_INI	O/V	0H 1H 2H 3H	INIT_PENDING REINIT_PENDING INIT_ACTIVE INIT_FINISHED	1	[-]
State variable indicating the current status of initialization of injection valve power stage banks					
LV_TI_3_HOM_ACT	-	0...1H	0...1	1	[-]
Logical value indicating active homogeneous injection for third pulse					
STATE_INJ_DR	V	0H 1H 2H 3H 4H 5H	DISABLED INI CONF WAIT WAIT2RUN RUN	1	[-]
Actual state of injection driver					
LV_INJ_REST_ENA	V	0...1H	0...1	1	[-]
Flag to indicate a possible restart of injection during running engine					
NR_PBK_IV_INI	V	0...8H	0...8	1	[-]
Number of initialized injection valve power banks					
LV_INJ_MOD_UPD	V	0...1H	0...1	1	[-]
Flag to indicate that the possibility of an injection mode data update (1) or mode switch (0) has to be checked					
IGA_TMP_SYN	-	FA60...5A0H	-90...90	0.0625	[°CRK]
Temporary ignition angle for ignition/injection synchronization calculations					
TD_TMP_SYN	-	0...FFFFH	0...262.14	0.004	[ms]
Temporary ignition dwell time for ignition/injection synchronization calculations					
SOI_TMP_SYN	-	0...780H	0...720	0.375	[°CRK]
Temporary start of injection angle for ignition/injection synchronization calculations					
SOI_TMP_SYN_PREV	-	0...780H	0...720	0.375	[°CRK]
Temporary start of injection angle for ignition/injection synchronization calculations (previous value)					
SOI_EOI_ACT_UPD[NC_CYL_NR]	V	0...780H	0...720	0.375	[°CRK]
Start or end of injection angle for injection mode data update calculations					
TI_ACT_UPD[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Injection time for injection mode data update calculations					
LV_SOI_ACT_UPD[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Flag to indicate the use of start of injection angle for injection mode update calculations					
SOI_EOI_ACT_SWI[NC_CYL_NR]	V	0...780H	0...720	0.375	[°CRK]
Start or end of injection angle for injection mode switch calculations					

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
TI_ACT_SWI[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Injection time for injection mode switch calculations					
LV_SOI_ACT_SWI[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Flag to indicate the use of start of injection angle for injection mode switch calculations					
LV_IGN_INJ_UPD_WIN	V	0...1H	0...1	1	[-]
Flag to indicate that an injection mode switch or update is possible					
LV_INH_INJ_UPD_MOD_UPD[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Flag to indicate an inhibitation of update mode update					
CRK_PSN_INJ_BAS[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
Engine position relative to INJ_BAS angle					
CRK_WIN_SEL_IGN_INJ[NC_CYL_NR]	O/V	0H 1H 2H 3H 4H	NO_CLC NO_WIN ADC_WIN RTD_WIN UPD_OLD_STATE	1	[-]
Selected injection window for update/mode switch					
CRK_WIN_SEL_TMP	-	0H 1H 2H 3H 4H	NO_CLC NO_WIN ADC_WIN RTD_WIN UPD_OLD_STATE	1	[-]
Selected injection window for update/mode switch; temporary value					
CRK_WIN_SEL_PREV	-	0H 1H 2H 3H 4H	NO_CLC NO_WIN ADC_WIN RTD_WIN UPD_OLD_STATE	1	[-]
Previous selected injection window for update/mode switch; temporary value					
STATE_UPD_IGN[NC_CYL_NR]	V	0H 1H 2H	NONE TRIG_IT TRIG_EXT	1	[-]
State of ignition update for ignition/injection synchronization calculations					
STATE_IGN_UPD_ENA	O/V	0...FFH	0...255	1	[-]
Bitmask to enable/disable ignition update calculations					
LV_EGY_RNG_IV_PLS_1_ACT[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Flag for the injector energy range of the first pulse (0 = low range, 1 = high range); for injection update calculations					
LV_INJ_UPD_CLC_ENA	O/V	0...1H	0...1	1	[-]
Flag to enable injection update calculations					
STATE_INJ_UPD_TRM	O/V	0...FFH	0...255	1	[-]
State, which indicates that at least one data set per cylinder was sent to I/O-SW (bit x of assigned cylinder =1)					
INJ_UPD_ACK[NC_CYL_NR]	V	0...FFH	0...255	1	[-]
Actual applied update mode; returned by I/O software					
IGA_IGC_H_RNG_ACT[NC_CYL_NR]	O/V	FA60...5A0H	-90...90	0.0625	[°CRK]
Ignition angle (high range) currently applied by I/O-SW, cylinder individual					
TD_IGC_ACT[NC_CYL_NR]	O/V	0...FFFFH	0...262.14	0.004	[ms]
Dwell time currently applied by I/O-SW, cylinder individual					
CRK_PSN_ENG_IGN_INJ_UPD[NC_CYL_NR]	V	0...780H	0...720	0.375	[°CRK]
Expected engine update position at start of transmitting ignition and injection parameter to I/O-SW					
CTR_INJ_SWI_ACK_ERR	V	0...FFFFH	0...65535	1	[-]
Counter for injection update errors					
LV_STATE_PBK_IV_INI_ACT	V	0...1H	0...1	1	[-]
Flag to indicate an active injection driver initialization phase					
LV_EGY_RNG_IV_PLS_1_ACK[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
Cylinder individual flag to indicate that the first injection pulse was executed with high injector needle lift (1 = high).					
TI_1_ADD_DLY_HOM	V	8000...7FFFH	-32768...32767	1	[µs]
Injector dead time correction first pulse, homogeneous mode					
LV_INJ_DEAC_BACK_ENG	V	0...1H	0...1	1	[-]
Flag to indicate deactivated injection due to backward rotation of engine					
TI_1_HOM_ACT[NC_CYL_NR]	V	0...FFFFH	0...65.535	0.001	[ms]
Injection time for first pulse HOM that was latetst transferred to basic software.					
IDX_TI_1_HOM_ACT[NC_CYL_NR]	V	0...FFFFH	0...65535	1	[-]
Counting factor index for first pulse HOM that was used for calculation of latetst transferred injection time.					

Input data:

NC_CRK_INJ_BAS_REF	NC_CTR_MAX_IV_TEST_MOD	LV_TI_EXT_ADJ[NC_CYL_NR]	
LV_INJ_CRASH_ACT	NC_CYL_NR	TI_EXT_ADJ[NC_CYL_NR]	TI_1_S[NC_CYL_NR]
SOI_LIM	INH_INJ		TI_2_S[NC_CYL_NR]
EOI_LIM_HOM	LV_POST_INJ_ACT	SOI_POST_INJ[NC_CYL_NR]	PRS_DEC_INJ_1_HOM[NC_CYL_NR]
EOI_LIM_POST	SOI_LIM_POST	TI_POST_INJ[NC_CYL_NR]	EOI_1_S[NC_CYL_NR]
TI_1_HOM[NC_CYL_NR]	STATE_INH_IV_DYN	NC_CBK_HPP_NR	NR_CYL_IV_TEST_MOD
SOI_1_HOM[NC_CYL_NR]	N_32	IDX_PRS_COR_CYL_CLC_MPLH[NC_CBK_HPP_NR]	SUM_INH_INJ
TI_2_HOM[NC_CYL_NR]	EGY_STEP_INJ_CHA_GRD[NC_CYL_NR]	EGY_STEP_INJ_DCHA_GRD[NC_CYL_NR]	INH_IV
EOI_2_HOM[NC_CYL_NR]	LV_ES	NC_INJ_INH_SWI_IV_SHIFT_NR	PSN_ENG
LV_ADD_PULSE_ENA	LV_ST_END	T_DLY_1_2_S_EXT[NC_CYL_NR]	INH_IV_MIS_GEN
~~T_CHA_PER_HOM[NC_CYL_NR]~~	T_CHA_PER_S_1[NC_CYL_NR]	T_CHA_PER_S_2[NC_CYL_NR]	T_CHA_PER_POST
~~T_DCHA_PER_HOM[NC_CYL_NR]~~	~~T_DCHA_PER_S_1[NC_CYL_NR]~~	~~T_DCHA_PER_S_2[NC_CYL_NR]~~	T_DCHA_PER_POST
	SOI_1_HOMS[NC_CYL_NR]	EOI_2_HOMS[NC_CYL_NR]	CDN_DIAG_IV_REQ[NC_CYL_NR]
NC_TI_CRASH	LV_ST_INJ_REQ	NC_PBK_IV_NR	TCO
T_CHA_PER_HOM_2[NC_CYL_NR]	~~T_DCHA_PER_HOM_2[NC_CYL_NR]~~	SEG_NR	LV_MC_SOPC_INH_DI
LV_ERR_TMP_MU_MC	LV_DIAG_END_RLY_MAIN_DLY	LV_RLY_MAIN_DLY_ERR	T_CHA_PER_HOM_1[NC_CYL_NR]
TI_3_HOM[NC_CYL_NR]	EOI_3_HOM[NC_CYL_NR]	LV_SYN_ENG	NC_NR_DCDC_INJ
T_CHA_PER_HOM_3[NC_CYL_NR]	~~T_DCHA_PER_HOM_3[NC_CYL_NR]~~	LV_INJ_OFF_TMR_INJ_ENA	T_CHA_PER_S_3[NC_CYL_NR]
~~T_DCHA_PER_S_3[NC_CYL_NR]~~	T_DLY_2_3_S_EXT[NC_CYL_NR]	EOI_3_HOMS[NC_CYL_NR]	TI_3_S[NC_CYL_NR]
LV_IGK	INJ_MOD_SP[NC_CYL_NR]	FAC_CHA_IV_MES_PBK[NC_PBK_IV_NR]	FAC_V_IV_MES_PBK[NC_PBK_IV_NR]
NC_IDX_CYL_PBK_IV_REF[NC_CYL_NR]	LV_IV_TEST_MOD_AUTH	LV_EGY_RNG_IV_PLS_1_HOM[NC_CYL_NR]	CRK_INJ_BAS[NC_CYL_NR]
FAC_N	IGA_AV_H_RNG_HOMS[NC_CYL_NR]	IGA_AV_H_RNG_S[NC_CYL_NR]	IGA_AV_H_RNG[NC_CYL_NR]
TD	TD_S	TD_AD[NC_CYL_NR]	TD_LIM_MAX
NC_TD_LIM_MIN	INJ_MOD_SP_HOM[NC_CYL_NR]	INJ_MOD_SP_S[NC_CYL_NR]	INJ_MOD_SP_HOMS[NC_CYL_NR]
NC_N_MAX_IV_TEST_MOD	NC_TI_TEST_PLS_PER	NC_IV_CRASH	STATE_INJ_UPD_ENA

CRK_PSN_STAT_IGN_UPD_ST	CRK_PSN_STAT_IGN_UPD_END	CRK_PSN_STAT_WIN_ST	CRK_PSN_STAT_WIN_END
LV_IGA_AND_INJ_SWI_HOMS	NC_STATE_STST_ENA	LV_INJ_PLS_UPD_MPLH_DEAC	T_DCHA_PER_HOM_1_1[NC_CYL_NR]
T_DCHA_PER_HOM_1_2[NC_CYL_NR]	T_DCHA_PER_HOM_2_1[NC_CYL_NR]	T_DCHA_PER_HOM_2_2[NC_CYL_NR]	T_DCHA_PER_HOM_3_1[NC_CYL_NR]
T_DCHA_PER_HOM_3_2[NC_CYL_NR]	T_DCHA_PER_S_1_1[NC_CYL_NR]	T_DCHA_PER_S_1_2[NC_CYL_NR]	T_DCHA_PER_S_2_1[NC_CYL_NR]
T_DCHA_PER_S_2_2[NC_CYL_NR]	T_DCHA_PER_S_3_1[NC_CYL_NR]	T_DCHA_PER_S_3_2[NC_CYL_NR]	T_DCHA_DLY_HOM_1[NC_CYL_NR]
T_DCHA_DLY_HOM_2[NC_CYL_NR]	T_DCHA_DLY_HOM_3[NC_CYL_NR]	T_DCHA_DLY_S_1[NC_CYL_NR]	T_DCHA_DLY_S_2[NC_CYL_NR]
T_DCHA_DLY_S_3[NC_CYL_NR]	IDX_TI_1_HOM_CLC[NC_CYL_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TI_ADD_PULSE_MIN	6	0...FFFFH	0...65.535	0.001	[ms]
LDPM_PRS_DEC_INJ_1_INJR	6	0...FFFFH	0...347776	5.3067216	[hPa]
Minimum injection time for additive injection pulse, pressure dependent					
IP_FAC_ADD_PULSE	7*7	0...FFH	0...1.99218	0.0078125	[-]
LDP_N_32_IP_FAC_ADD_PULSE	7	0...FFH	0...8160	32	[rpm]
LDP_TCO_IP_FAC_ADD_PULSE	7	0...FEH	-48...142.5	0.75	[°C]
Weighting factor for additive pulses in case of injection time update at transient conditions					
C_SOI_EXT_ADJ	1	0...780H	0...720	0.375	[°CRK]
Start of injection for external adjustment (service tool intervention)					
C_NR_CYL_OFS_INH_IV_DYN	1	0...7H	0...7	1	[-]
Cylinder offset for destination of current cylinder number for dynamic fuel shut off					
LC_INJ_PLS_UPD_MOD	1	0H 1H	IMMEDIATE AUTONOMOUS	1	[-]
Logical constant for selection of injection pulse update mode					
IP_EOI_INJ_UPD_PSN	6	0...780H	0...720	0.375	[°CRK]
LDP_N_32_IP_EOI_INJ_UPD_PSN	6	0...FFH	0...8160	32	[rpm]
End of injection position for autonomous generated update pulses					
ID_NR_PBK_IV_PTR	NC_CYL_NR	0...7H	0...7	1	[-]
LDP_NR_CYL_ID_NR_PBK_IV_PTR	NC_CYL_NR	0...7H	0...7	1	[-]
Pointer map from logical cylinder number to assigned power stage bank					
C_DUCY_SP_BOOST_PBK_IV	1	0...4000H	0...100	6.1035e-3	[%]
Basic operating setpoint for the DCDC converter of the injection valve power stages					
LC_IGN_UPD_STAT_ENA	1	0...1H	0...1	1	[-]
Flag to enable static update calculation of ignition angle (at ignition/injection synchronization)					
C_T_OFS_UPD_IGN_INJ_SYN	1	0...FFFFH	0...65.535	0.001	[ms]
Task run time offset for a synchronized update of ignition and injection parameters to I/O-SW					
LC_IV_ADD_PLS_SNGH_ENA	1	0...1H	0...1	1	[-]
Switch to enable makeup pulse functionality at homogeneous single injection					
LC_IV_ADD_PLS_MPLH_ENA	1	0...1H	0...1	1	[-]
Switch to enable pulse update functionality at homogeneous multiple injection					
C_SOI_MIN_UPD_PLS_SNGH	1	0...780H	0...720	0.375	[°CRK]
Minimum start (limit) of injection for makeup pulse at homogeneous single injection mode					
LC_INJ_UPD_OLD_MOD_ENA	1	0...1H	0...1	1	[-]
Flag to enable injection data update of old injection mode in case of mode switch failed					
IP_TI_ADD_DLY_HPDI_EGY_H	6	0...FFFFH	-32768...32767	1	[µs]
LDPM_PRS_DEC_INJ_1_INJR	6	0...FFFFH	0...347776	5.3067216	[hPa]
Injector dead time correction for high energy range of the injector					
IP_TI_ADD_DLY_HPDI_EGY_L	6	0...FFFFH	-32768...32767	1	[µs]
LDPM_PRS_DEC_INJ_1_INJR	6	0...FFFFH	0...347776	5.3067216	[hPa]
Injector dead time correction for low energy range of the injector					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_INJ_MOD_DI	1	0...FFH	0...255	1	[-]
Constant defined to indicate that injection is disabled					
NC_INJ_MOD_HOM	1	0...FFH	0...255	1	[-]
Constant defined to indicate homogeneous mode					
NC_INJ_MOD_S	1	0...FFH	0...255	1	[-]
Constant defined to indicate stratified mode					
NC_INJ_MOD_HOMS	1	0...FFH	0...255	1	[-]
Constant defined to indicate homogeneous stratified mode					
NC_INJ_MOD_SINGLE	1	0...FFH	0...255	1	[-]
Constant defined to indicate single injection mode					
NC_INJ_MOD_MULTI	1	0...FFH	0...255	1	[-]
Constant defined to indicate multi injection mode					

NC_INJ_MOD_MULTI_PLS3	1	0...FFH	0...255	1	[-]
Constant defined to indicate multi injection mode with three pulses					
NC_INJ_MOD_TEST_PLS	1	0...FFH	0...255	1	[-]
Constant defined to indicate injection mode for actuator tests by diagnostic tester					
NC_INJ_MOD_MASK_1	1	0...FFH	0...255	1	[-]
Mask for injection mode determination					
NC_INJ_MOD_MASK_2	1	0...FFH	0...255	1	[-]
Mask for injection pulse type determination					

Imported actions:

ACTION_INFR_**IniInjDriver** ( )
Initialization of the injection driver system.
ACTION_INFR_**SetInjPulseEnable** (IN <Cyl>, IN <Pls_nr>, IN <Lv_pls>)
Enables (Lv_pls=1) or disables (Lv_pls=0) the injection pulse Pls_nr for cylinder Cyl.
ACTION_INFR_**SetInjPulseType**(IN <Cyl>, IN <Pls_nr>, IN <Pls_type>)
Sets the type for the parameterization of the Pulse Pls_nr and Cylinder Cyl by the parameter Pls_type.
ACTION_INFR_**SetInjPulsePosition** (IN <Cyl>, IN <Pls_nr>, IN <Pls_position>)
Sets the position of the Pulse Pls_nr and Cylinder Cyl; value is interpreted according to ACTION_INFR_SetInjPulseType().
ACTION_INFR_**SetInjPulseDuration** (IN <Cyl>, IN <Pls_nr>, IN <Pls_duration>)
Sets the pulse duration for the Pulse Pls_nr and Cylinder Cyl; value is interpreted according to ACTION_INFR_SetInjPulseType().
ACTION_INFR_**SetInjPlsUpdMinPer**(IN <Cyl>, IN <Ti_min>)
Sets the minimum injection time for self-created update pulses of the selected cylinder Cyl.
ACTION_INFR_**SetInjPulseUpdFac**(IN <Cyl>, IN <Pls_nr>, IN <Fac_add_pls>)
In case of an update of the pulse parameters for Cyl and Pls_nr the factor, Fac_add_pls is used.
ACTION_INFR_**ReqInjUpdate** (IN <Cyl>, IN <Mode>, OUT <Acknowledge>)
Update the pulse parameters for cylinder Cyl. Depending on type, a running injection is changed or not. The output parameter Acknowledge reflects if the update mode is fulfilled (=1) or will be fulfilled (=0) later on.
ACTION_INFR_**SetInjPulseUpdMode** (IN <Cyl>, IN <Pls_nr>, IN <Upd_mode>)
Sets the update mode Upd_mode of pulse Pls_nr on cylinder Cyl.
ACTION_INFR_**SetInjPlsUpdOffset**(IN <Cyl>, IN <Upd_offset>)
Sets the injector dead time correction TI_ADD_DLY for self created update pulses of cylinder Cyl.
ACTION_INFR_**SetInjSOILim**(IN <Cyl>, IN <Packet_nr>, IN <Soi_lim>)
ACTION_INFR_SetInjSOILim() and ACTION_INFR_SetInjEOILim() sets the injection window for injection pulse packet Packet_nr and Cylinder Cyl.
ACTION_INFR_**SetInjEOILim**(IN <Cyl>, IN <Packet_nr>, IN <Eoi_lim>)
ACTION_INFR_SetInjSOILim() and ACTION_INFR_SetInjEOILim() sets the injection window for injection pulse packet Packet_nr and Cylinder Cyl.
ACTION_INFR_**ReqInjInhibit** (IN <Cyl>, IN <Lv_inh_inj>)
Enables (Lv_inh_inj=0) or disables (Lv_inh_inj=1) all pulses of the injection for cylinder Cyl.
ACTION_INFR_**SetInjMode** (IN <Injection_mode>)
This action sets the injection mode (DISABLE, SEQUENTIAL, STATIC, PREINJECTION, CRASH).
ACTION_INFR_**GetInjCylAv** (OUT <Cyl_av>)
Returns the actual cylinder Cyl_av of the trigger INJBAS at CRK_INJ_BAS[Cyl_av].
ACTION_INFR_**GetInjPrevState** (IN <Cyl>, OUT <Lv_prev_state_iv>)
Returns the previous state of the injection for cylinder Cyl.
ACTION_INFR_**SetInjBaseAngle** (IN <Crk_inj_bas_ref>)
Sets the crankshaft position, where the Infrastructure generates the trigger INJBAS at the end of the injection of each cylinder. Crk_inj_bas_ref and NC_CRK_INJ_REF_TDC are used to set CRK_INJ_BAS[x].
ACTION_INFR_**SetInjChargeCur** (IN <Cyl>, IN <Ducy>)
Sets the charge current as duty cycle Ducy of Cylinder Cyl
ACTION_INFR_**SetInjDischargeCur** (IN <Cyl>, IN <Ducy>)
Sets the discharge current as duty cycle Ducy of Cylinder Cyl
ACTION_INFR_**SetInjChargePer** (IN <Cyl>, IN <Pls_nr>, IN <Period>)
Sets the charge duration of pulse Pls_nr for cylinder Cyl
ACTION_INFR_SetInjDischargePer2(IN <Cyl>, IN <Pls_nr>, IN <Period>)
Sets the second discharge duration of pulse Pls_nr for cylinder Cyl
ACTION_INFR_SetInjDischargePer1(IN <Cyl>, IN <Pls_nr>, IN <Period>)
Sets the first discharge duration of pulse Pls_nr for cylinder Cyl
ACTION_INFR_SetInjDischargeDly(IN <Cyl>, IN <Pls_nr>, IN <Period>)
Sets the discharge delay time between the first and the second discharge phase of pulse Pls_nr for cylinder Cyl
ACTION_INFR_**GetInjVMesVld** (IN <Cyl>, OUT <Lv_Inj_Mes_Vld>)
Returns a flag that indicates if the measurements for charge and voltage on injector Cyl are valid
ACTION_INFR_**GetInjVCha** (IN <Cyl>, IN <Pls_nr>, IN <Mes_ref>, OUT <Charge>)
Returns the measured charge of a carried out pulse Pls_nr on injector Cyl at reference point Mes_ref

ACTION_INFR_**GetInjVInj** (IN <Cyl>, IN <Pls_nr>, IN <Mes_ref>, OUT <Voltage>)
Returns the measured voltage of a carried out pulse Pls_nr on injector Cyl at reference point Mes_ref
ACTION_INFR_**GetInjVTrim** (IN <Cyl>, OUT <Charge>)
Returns the measured charge of the trim pulse related to cylinder Cyl

ACTION_INFR_**GetInjMesPulsePer** (IN <Cyl>, IN <Pls_nr>, OUT <Period>)
Returns the injection period of a carried out pulse Pls_nr on injector Cyl
ACTION_INFR_**GetInjMesSoi** (IN <Cyl>, IN <Pls_nr>, OUT <Pls_position>)
Returns the start of injection position (SOI) of a carried out pulse Pls_nr on injector Cyl
ACTION_INFR_**GetInjMesEoi** (IN <Cyl>, IN <Pls_nr>, OUT <Pls_position>)
Returns the end of injection position (EOI) of a carried out pulse Pls_nr on injector Cyl
ACTION_INFR_**ReqInjPulse** (IN <Cyl>, IN <Pulse_per>)
Requests static injection test pulses with a period Pulse_per on cylinder Cyl.
ACTION_INFR_**GetElDiagIVCyl** (IN <Cyl>, INOUT <Cdn_diag>, OUT <Err_diag>)
Returns the electrical diagnosis information Err_diag for cylinder Cyl under consideration of diagnosis condition Cdn_diag.
ACTION_INFR_**SetInjDcdc**(IN <Dcdc>, IN <Ducy>)
Sets the duty cycle for the DC/DC converter used by power stages.
ACTION_INFR_**GetInjVDcdc**(IN <Dcdc>, OUT <Voltage_dcdc>)
Returns the output voltage of the DC/DC converter used by power stages.
ACTION_INFR_**GetInjIniErr**(IN <Bank>, OUT <Err_ini>)
Returns error information in Err_ini after startup-check of the selected power stage bank.
ACTION_INFR_**GetInjOnlineErr** (IN <Bank>, OUT <Err_online>)
Returns error information in Err_online about the online state of the selected power stage Bank.
ACTION_INJR_**SetStateInjUpdEna**(IN <Cyl>, IN <Value>)
This action sets the bit x of the variable STATE_INJ_UPD_ENA to the input Value
ACTION_IGRE_**SendTrigForIgnUpd**(IN <PRM_STATE_IGN_UPD_ENA>)
Trig IGRE to update ignition parameter on designated cylinders which are specified by the parameter.

Exported actions:

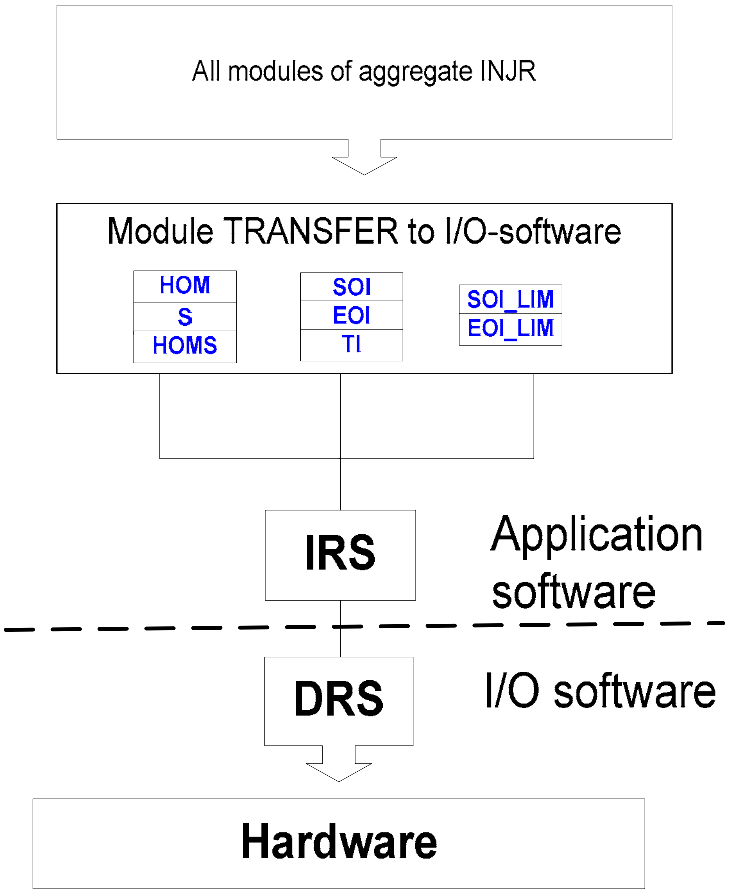
ACTION_INJR_IgnUpdate(IN <Cyl>, IN <Iga_Igc_H_Rng>)
This action sets the cylinder individual ignition angle for mode switch and data update of <Cyl> to <Iga_Igc_H_Rng>..

FUNCTION DESCRIPTION:

Overview:

The purpose of this module is the transfer of injection related data to IO-Software under real time conditions. The most important thing is to maintain the data consistency. The figure shows the general flow of data:

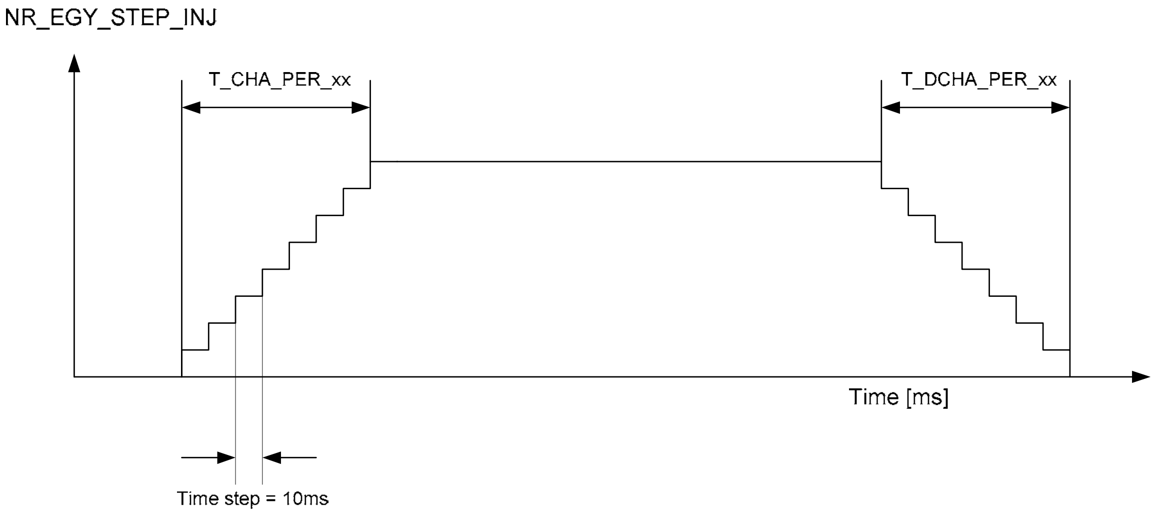
Overview: Interface ASW to I/O-software



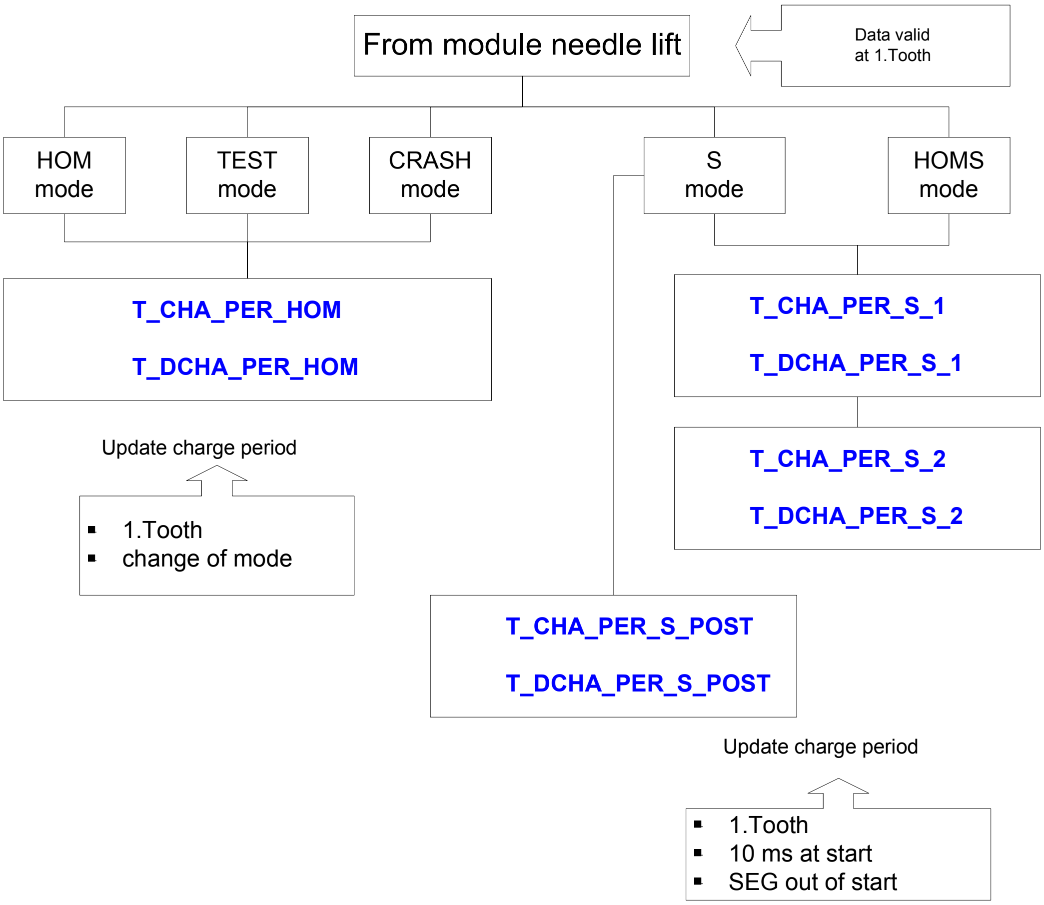
Control of piezo injector needle lift

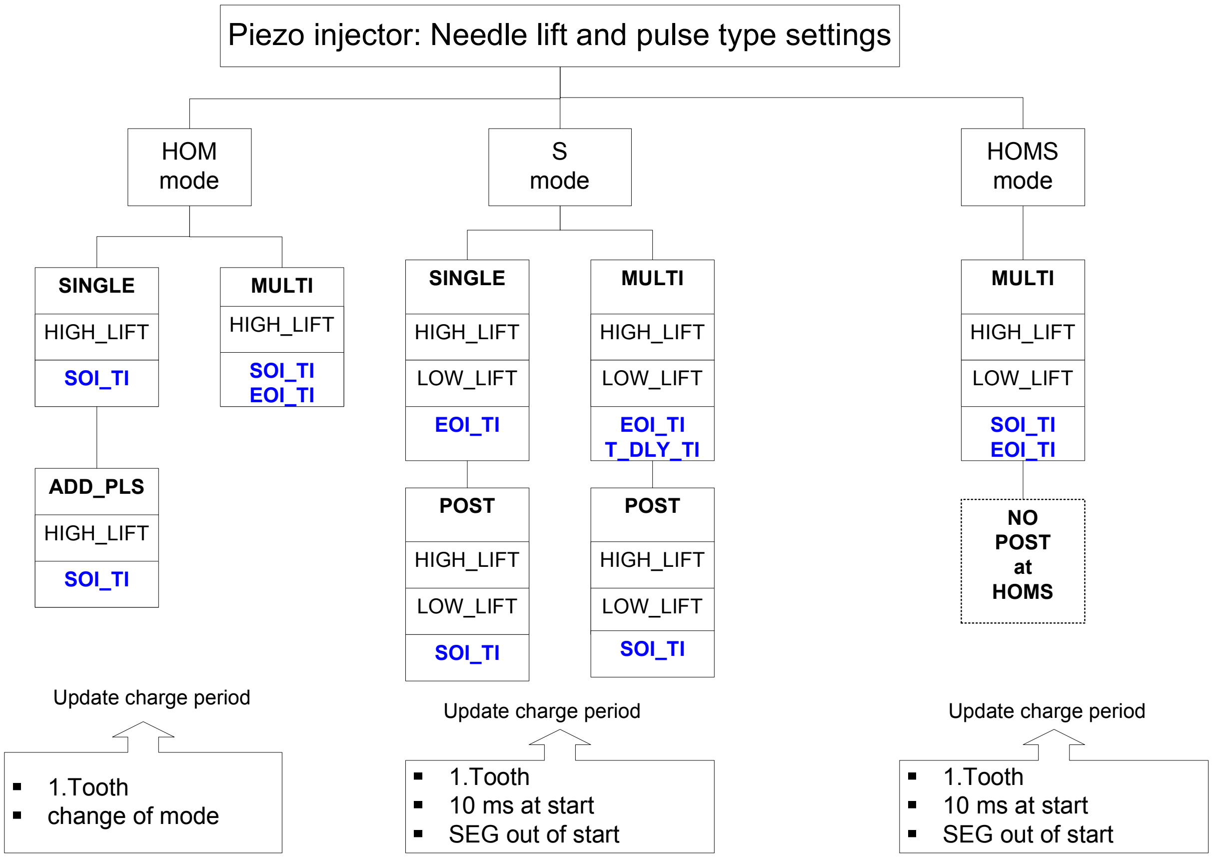
To control the needle lift of the piezo injector the energy loaded into the actuator is set by the number of energy packets. In the actual version are two possible setting: High lift and low lift. The data is an input data from module needle lift. The variables are T_CHA_PER_xxx and T_DCHA_PER_xxx, which describe the charge and discharge time.

Setting of energy packets for piezo injector



For different injection modes, the energy packets can be different. The following picture shows the overview of the different setting and the update timing conditions.



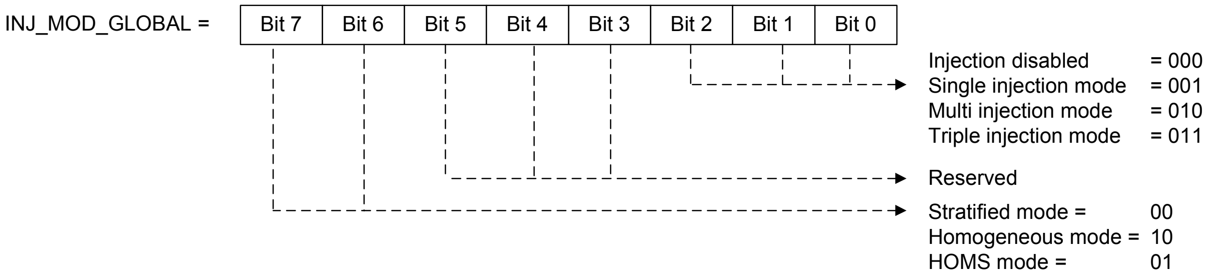


Formula section

41.4.1 Definitions bit masks for injection mode

Byte definition

INJ_MOD[x], INJ_MOD_SP[x]



Example: homogenous multi injection mode (MPLH)

INJ_MOD_GLOBAL =								
	1	0	0	0	0	0	1	0

Definition of NC constants

NC_INJ_MOD_DI = 0x00H
NC_INJ_MOD_S = 0x00H
NC_INJ_MOD_HOM = 0x80H
NC_INJ_MOD_HOMS = 0x40H
NC_INJ_MOD_SINGLE = 0x01H

NC_INJ_MOD_MULTI = 0x02H
NC_INJ_MOD_MULTI_PLS3 = 0x03H
NC_INJ_MOD_TEST_PLS = 0x07H
NC_INJ_MOD_MASK_1 = 0xC0H
NC_INJ_MOD_MASK_2 = 0x07H

41.4.2 Reset tasks

This chapter describes the procedures required at reset for an accurate engine start.

Application conditions:

Activation: at reset
Deactivation: -
Initialization: -
Recurrence: once at reset

Note! This task has to be called first in calculation order of reset tasks.

Formula Section:

Set the crankshaft position where the Infrastructure generates the trigger INJBAS at the end of the injection of each cylinder.

```
ACTION_INFR_SetInjBaseAngle (IN Crk_inj_bas_ref = NC_CRK_INJ_BAS_REF)
(1) FOR d = 0 TO (NC_NR_DCDC_INJ - 1) DO
41.4.2.1    ACTION_INFR_SetInjDcdc(IN d, IN Ducy = C_DUCY_SP_BOOST_PBK_IV)
(1) ENDFOR

CTR_INJ_SWI_ACK_ERR = 0
STATE_INJ_UPD_TRM = 0
LV_INJ_UPD_CLC_ENA = 1

(1) FOR x = 0 TO (NC_CYL_NR –1) DO:
      INJ_MOD[x] = "DISABLE"
      SOI_1_MES[x] = 720° CRK
      SOI_2_MES[x] = 720° CRK
      SOI_POST_MES[x] = 720° CRK
(1) ENDFOR
```

Do not start injection before first valid tooth event was reached. Injection driver is locked

```
LV_ST_INJ = 1
STATE_INJ_DR = "DISABLED"
STATE_PBK_IV_INI = "INIT_PENDING"
LV_INJ_REST_ENA = 0
LV_STATE_PBK_IV_INI_ACT = 0
```

41.4.3 Power stage initialization

This chapter describes the procedures to initialize the power stage driver.

Application conditions:

Activation: at every engine state

Deactivation: -

Initialization: -

Recurrence: every 10ms

Formula Section:

Initialization of injection IO-driver

```
(1)IF (STATE_PBK_IV_INI = "INIT_PENDING")
    AND
    (LV_MC_SOPC_INH_DI = 0)
    AND
    (LV_ERR_TMP_MU_MC = 0)
    AND
    (LV_DIAG_END_RLY_MAIN_DLY = 1)
    AND
    (LV_RLY_MAIN_DLY_ERR = 0)
    AND
    (LV_INJ_OFF_TMR_INJ_ENA = 1)
    AND
    (LV_STATE_PBK_IV_INI_ACT = 0)
(1)THEN
    STATE_PBK_IV_INI = "INIT_ACTIVE"
    NR_PBK_IV_INI = 0
    LV_STATE_PBK_IV_INI_ACT = 1
    ACTION_INFR_IniInjDriver ()
(1)ENDIF
```

Re-initialization of injection IO-driver

```
(2)IF (STATE_PBK_IV_INI = "REINIT_PENDING")
    AND
    (LV_IGK = 1)
    AND
    (LV_INJ_OFF_TMR_INJ_ENA= 1)
    AND
    (LV_STATE_PBK_IV_INI_ACT = 0)
(2)THEN
    STATE_PBK_IV_INI = "INIT_ACTIVE"
    NR_PBK_IV_INI = 0
    LV_STATE_PBK_IV_INI_ACT = 1
    ACTION_INFR_IniInjDriver ()
```

```
(2)ENDIF
```

Set flag to indicate that a proper start was performed

```
(3)IF LV_ST_END = 1
(3)THEN
        LV_INJ_REST_ENA = 1
(3)ENDIF
```

41.4.4 End of injection driver initialization

Application conditions:

Activation:	every INJEOD-event (EndOfDriverInitialization)
Deactivation:	-
Initialization:	-
Recurrence:	INJEOD-event (EndOfDriverInitialization)

Formula Section:

Note! p represents the power-stage bank which reached the INJEOD-event.

```
NR_PBK_IV_INI = NR_PBK_IV_INI + 1
ACTION_INFR_GetInjIniErr (IN p, OUT STATE_ERR_PBK_IV_ST_RAW[p] = Err_ini)

(1)IF NR_PBK_IV_INI >= NC_PBK_IV_NR
(1)THEN – every injection valve powerbank was initialized
  LV_STATE_PBK_IV_INI_ACT = 0
  (1a)IF STATE_PBK_IV_INI = "INIT_ACTIVE"
  (1a)THEN
      STATE_PBK_IV_INI = "INIT_FINISHED"
        (2)IF STATE_INJ_DR = "DISABLED" AND LV_INJ_REST_ENA = 1
        (2)THEN
             Calculate section 41.4.6 (Start initialization)
        (2)ENDIF
  (1a)ENDIF
(1)ENDIF
```

41.4.5 First valid tooth tasks

Application conditions:

Activation: at first valid tooth

Deactivation: -

Initialization: -

Recurrence: once at first valid tooth

This chapter describes the procedures required at first valid tooth for an accurate engine start.

Formula Section:

Calculate section 41.4.6 (Start initialization)

LV_ST_INJ = 0

41.4.6 Start initialization

Note: calculate this section only if called by another section

41.4.6.1 Mode independent

```
(1) FOR d = 0 TO (NC_NR_DCDC_INJ - 1) DO
41.4.6.2    ACTION_INFR_SetInjDcdc(IN d, IN Ducy = C_DUCY_SP_BOOST_PBK_IV)
(1) ENDFOR
INH_IV_DYN = 0
NR_CYL_OFS_INH_IV_DYN = C_NR_CYL_OFS_INH_IV_DYN
STATE_INJ_UPD_TRM = 0 (all bits)
LV_INJ_UPD_CLC_ENA = 1
INH_IV_MIS_GEN_ACK = 0
Do the following calculations for all cylinders
(2) FOR x = 0 TO (NC_CYL_NR –1) DO:
    ACTION_INJR_SetStateInjUpdEna(IN Cyl = x, State_Inj_Upd_Ena = 0)
    CTR_INH_IV_MIS_GEN[x] = 0

(2) ENDFOR (all cylinders)
NR_CYL_INJ_BAS = 0
NR_CYL_INJ_BAS_PREV = NC_CYL_NR -1
```

41.4.7 Crank synchronous tasks

Application conditions:

Activation: every engine state
Deactivation: -
Initialization: -
Recurrence: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

Formula Section:

```
(1)IF LV_INJ_UPD_CLC_ENA = 1
(1)THEN
    STATE_IGN_UPD_ENA = 0 (all bits)
```

Minimum injection time for additional pulse:
In case of single injection and homogeneous mode, a second injection pulse can be applied by the I/O-software at transient engine operation (as a result of an injection update). For this pulse, a minimum injection time TI_ADD_PULSE_MIN is defined. Each of the applied IP_TI_ADD_PULSE_MIN data should be greater than the smallest calculated TI_MIN.

```
    TI_ADD_PULSE_MIN = 0
    FOR m = 0 TO NC_CBK_HPP_NR - 1 DO
        TI_ADD_PULSE_MIN = MAX(TI_ADD_PULSE_MIN;
            IP_TI_ADD_PULSE_MIN (Input:
            PRS_DEC_INJ_1_HOM[IDX_PRS_COR_CLY_CLC_MPLH[m]]) )
    ENDFOR
    FAC_ADD_PULSE = IP_FAC_ADD_PULSE (Input: N_32, TCO)

(1)FOR x = 0 TO (NC_CYL_NR-1) DO
```

Note: In this specification the LSB of a bit field (e. g. STATE_INJ_UPD_ENA) is revered to as Bit 0! That means that the first loop of the for loop (x = 0) reveres to the LSBs, the next (x = 1) to the next and so on.

Cylinder shut off transfer data:

```
    Disable_task_interruption
    LV_INH_INJ[x] = ((Bit x of INH_INJ) OR (Bit x of INH_IV_MIS_GEN_ACK)    )    … logical
    ACTION_INFR_ReqInjInhibit (IN Cyl = x, IN Lv_inh_inj = LV_INH_INJ[x])
    Enable_task_interruption
```

Set injection mode set-point:

```
    (2)IF Bit x of STATE_INJ_UPD_TRM = 0
    (2)THEN – no valid data set within I/O software
```

```
        (3)IF Bit x of STATE_INJ_UPD_ENA = 1
        (3)THEN
                INJ_MOD[x] = INJ_MOD_SP[x]
                Calculate section 41.4.20.1 (Configure injection)
                Calculate section 41.4.20.2 (Update injection data)
                ACTION_INJR_SetStateInjUpdEna(IN Cyl = x, State_Inj_Upd_Ena = 0)
                ACTION_INFR_ReqInjUpd(IN Cyl = x, IN Mode = Immediate, OUT INJ_UPD_ACK[x] = Acknowledge)
                Calculate section 41.4.20.3 (Update ignition data)
                Bit x of STATE_IGN_UPD_ENA = 1
                CRK_WIN_SEL_IGN_INJ[x] = "NO_CLC"
                Bit x of STATE_INJ_UPD_TRM = 1
        (3)ENDIF

    (2)ELSE – valid data set within I/O software
        Calculate section 41.4.20.4 (Check engine position)
            (4)IF INJ_MOD[x] = INJ_MOD_SP[x]
        (4)THEN (no mode switch requested)
                (5)IF CRK_WIN_SEL_IGN_INJ[x] = "ADC_WIN" OR
                          CRK_WIN_SEL_IGN_INJ[x] = "RTD_WIN"
                (5)THEN (update possible)
                    (6)IF Bit x of STATE_INJ_UPD_ENA = 1
                    (6)THEN
                          Calculate section 41.4.20.2 (Update injection data)
                          ACTION_INJR_SetStateInjUpdEna(IN Cyl = x, State_Inj_Upd_Ena = 0)
                          ACTION_INFR_ReqInjUpd(IN Cyl = x, IN Mode = Immediate, OUT INJ_UPD_ACK[x] = Acknowledge)
                    (6)ENDIF
                (5)ENDIF
            (4)ELSE (mode switch requested)
                (7)IF CRK_WIN_SEL_IGN_INJ[x] = "ADC_WIN" OR
                          CRK_WIN_SEL_IGN_INJ[x] = "RTD_WIN"
                (7)THEN (mode switch possible)
                    INJ_MOD[x] = INJ_MOD_SP[x]
                    Calculate section 41.4.20.1 (Configure injection)
                    Calculate section 41.4.20.2 (Update injection data)
```

```
        ACTION_INJR_SetStateInjUpdEna(IN Cyl = x, State_Inj_Upd_Ena = 0)
        ACTION_INFR_ReqInjUpd(IN Cyl = x, IN Mode = Cycle, OUT INJ_UPD_ACK[x] = Acknowledge)
        (8)IF INJ_UPD_ACK[x] != "Cycle"
        (8)THEN
            Increment CTR_INJ_SWI_ACK_ERR by 1
        (8)ENDIF
    (7)ELSE (mode switching failed)
        Check if old injection mode can be updated
        LV_INJ_MOD_UPD = 1
        Calculate section 41.4.20.5 (Check variable window(input: LV_INJ_MOD_UPD; output: LV_IGN_INJ_UPD_WIN))
        (9)IF LC_INJ_UPD_OLD_MOD_ENA = 1
            AND LV_IGN_INJ_UPD_WIN = 1
            AND LV_INH_INJ_UPD_MOD_UPD[x] = 0
            AND ((INJ_MOD[x] = INJ_MOD_SP_HOM[x]) OR
                     (INJ_MOD[x] = INJ_MOD_SP_S[x]
                         AND LV_IGA_AND_INJ_SWI_HOMS = 0) OR
                     (INJ_MOD[x] = INJ_MOD_SP_HOMS[x]
                         AND LV_IGA_AND_INJ_SWI_HOMS = 1))
            AND NOT (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_S AND
                         INJ_MOD_SP[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_HOMS)
            AND NOT (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_HOMS AND
                     INJ_MOD_SP[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_S)
        (9)THEN
            Calculate section 41.4.20.2 (Update injection data)
```

```
                  ACTION_INFR_ReqInjUpd(IN Cyl = x, IN Mode = Immediate,
      OUT INJ_UPD_ACK[x] = Acknowledge)
                  CRK_WIN_SEL_IGN_INJ[x] = "UPD_OLD_STATE"
                  STATE_UPD_IGN[x] = "TRIG_IT"
              (9)ENDIF
          (7)ENDIF
      (4)ENDIF – mode switch or data update
    (10)IF STATE_UPD_IGN[x] = "TRIG_IT"
    (10)THEN
          Bit x of STATE_IGN_UPD_ENA = 1
          Calculate section 41.4.20.3 (Update ignition data)
    (10)ENDIF
  (2)ENDIF – data set within I/O software
(1)ENDFOR
ACTION_IGRE_SendTrigForIgnUpd(IN
PRM_STATE_IGN_UPD_ENA = STATE_IGN_UPD_ENA)
(1)ENDIF – LV_INJ_UPD_CLC_ENA


Get actual measured voltage of the DCDC converter
(2)FOR d = 0 TO (NC_NR_DCDC_INJ - 1) DO
 ACTION_INFR_GetInjVDcdc(IN d, OUT VBOOST_PBK_IV_MES[d] = Voltage_dcdc)
(2)ENDFOR
```

41.4.8 INJBAS triggered tasks

Application conditions:

Activation: every INJBAS event

Deactivation: -

Initialization: -

Recurrence: INJBAS synchronous

Formula Section:

Note! x represents the current assigned active cylinder cyl_av, which is used for subsequent data calculations.

41.4.8.1 Get actual and previous cylinder number of CRK_INJ_BAS

The Parameter Cyl_av is delivered from the I/O SW and indicates the logical cylinder number for the actual INJBAS event.

```
ACTION_INFR_GetInjCylAv (OUT x =Cyl_av)
NR_CYL_INJ_BAS_PREV= NR_CYL_INJ_BAS
NR_CYL_INJ_BAS= x
```

41.4.8.2 Transfer data

```
CTR_PSN_INH_IV_DYN = (STATE_INH_IV_DYN / $2^{16}$) AND 0x0F            ... bitwise
NR_CYL_INH_IV_DYN = (NC_CYL_NR + x - NR_CYL_OFS_INH_IV_DYN)
                              MODULO NC_CYL_NR
(1) IF (STATE_INH_IV_DYN AND $2^{CTR\_PSN\_INH\_IV\_DYN}$) $\neq$ 0       ... bitwise
(1) THEN
    Cylinder will be switched off:
    INH_IV_DYN = INH_IV_DYN OR $2^{NR\_CYL\_INH\_IV\_DYN}$                ... bitwise
(1) ELSE
    Cylinder will be switched on:
    INH_IV_DYN = INH_IV_DYN AND NOT($2^{NR\_CYL\_INH\_IV\_DYN}$)          ... bitwise
(1) ENDIF
CTR_PSN_INH_IV_DYN = CTR_PSN_INH_IV_DYN + 1
(2) IF CTR_PSN_INH_IV_DYN $\geq$ NC_INJ_INH_SWI_IV_SHIFT_NR
(2) THEN
    CTR_PSN_INH_IV_DYN = 0
(2) ENDIF
STATE_INH_IV_DYN = (STATE_INH_IV_DYN AND 0x0000FFFF)                   ... bitwise
                   OR                                                 ... bitwise
                   (CTR_PSN_INH_IV_DYN * $2^{16}$)                     ... bitwise
INH_INJ = INH_IV OR INH_IV_DYN                                        ... bitwise
LV_INH_INJ_OLD = LV_INH_INJ[NR_CYL_INH_IV_DYN]
LV_INH_INJ[NR_CYL_INH_IV_DYN] = (Bit NR_CYL_INH_IV_DYN of INH_INJ
                                 OR
                       Bit NR_CYL_INH_IV_DYN of INH_IV_MIS_GEN)
```

Note! Pay attention on data consistency of INH_INJ, INH_IV_DYN and LV_INH_INJ[x].

SUM_INH_INJ = The sum of the digits of INH_INJ

```
(3) IF (Bit NR_CYL_INH_IV_DYN of INH_IV_MIS_GEN) = 0
(3) THEN
INH_IV_MIS_GEN_ACK = INH_IV_MIS_GEN_ACK AND NOT ($2^{NR\_CYL\_INH\_IV\_DYN}$)
                                                        ... bitwise
CTR_INH_IV_MIS_GEN[NR_CYL_INH_IV_DYN] = 0
(3) ELSE
INH_IV_MIS_GEN_ACK = INH_IV_MIS_GEN_ACK OR ($2^{NR\_CYL\_INH\_IV\_DYN}$)        ... bitwise
CTR_INH_IV_MIS_GEN[NR_CYL_INH_IV_DYN] =
CTR_INH_IV_MIS_GEN[NR_CYL_INH_IV_DYN] + 1
(3) ENDIF
```

Please Note! STATE_INH_IV_DYN is an output data of cylinder shut off specification. The content of the data will be changed in this interrupt procedure. Pay attention on data consistency, even at cylinder shut off module

```
(1) IF LV_INH_INJ[NR_CYL_INH_IV_DYN] $\neq$ LV_INH_INJ_OLD
(1) THEN
    ACTION_INFR_ReqInjInhibit (IN Cyl = NR_CYL_INH_IV_DYN,
    IN Lv_inh_inj = LV_INH_INJ[NR_CYL_INH_IV_DYN])
(1) ENDIF
```

41.4.8.3 Post injection

```
(1) IF (LV_INJ_MPLP_CYL[x] ≠ LV_POST_INJ_ACT)
      AND                                                          ...logical
      ((INJ_MOD[x] AND NC_INJ_MOD_MASK_1) = NC_INJ_MOD_S)          ...bitwise
      AND                                                          ...logical
      ((INJ_MOD[x] AND NC_INJ_MOD_MASK_2) ≠ NC_INJ_MOD_MULTI_PLS3) ...bitwise
(1) THEN (stratified mode)
    LV_INJ_MPLP_CYL[x] = LV_POST_INJ_ACT
    (2) IF LV_INJ_MPLP_CYL[x] = 1
    (2) THEN
    Activate post injection:
    ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 3, IN Pls_type = SOI_TI)
    ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 3, IN Upd_mode  = NONE)
    ACTION_INFR_SetInjSOILim(IN Cyl = x, IN Packet_nr = 1, IN Soi_lim = SOI_LIM_POST)
    ACTION_INFR_SetInjEOILim(IN Cyl = x, IN Packet_nr = 1, IN Eoi_lim = EOI_LIM_POST)
    ACTION_INFR_SetInjPulsePosition( IN Cyl = x, IN Pls_nr = 3,
                                     IN Pls_position = SOI_POST_INJ[x])
    ACTION_INFR_SetInjPulseDuration( IN Cyl = x, IN Pls_nr = 3,
                                     IN Pls_duration = TI_POST_INJ[x])
    ACTION_INFR_SetInjPulseEnable(  IN Cyl  = x, IN Pls_nr = 3, IN Lv_pls = 1)
    (2) ELSE
    Deactivate post injection:
    SOI_POST_MES[x] = 720° CRK
    EOI_POST_MES[x] = 0° CRK
    TI_POST_MES[x] = 0 ms
    ACTION_INFR_SetInjPulseEnable(IN Cyl  = x, IN Pls_nr = 3, IN Lv_pls = 0)
    (2) ENDIF
(1) ELSE (not stratified)
    (3) IF (LV_INJ_MPLP_CYL[x] = 1)
        AND
        ((INJ_MOD[x] AND NC_INJ_MOD_MASK_1) ≠ NC_INJ_MOD_S))       ...bitwise
    (3) THEN Deactivate post injection
        LV_INJ_MPLP_CYL[x] = 0
        SOI_POST_MES[x] = 720° CRK
        EOI_POST_MES[x] = 0° CRK
        TI_POST_MES[x] = 0 ms
        ACTION_INFR_SetInjPulseEnable(IN Cyl  = x, IN Pls_nr = 3, IN Lv_pls = 0)
     (3) ENDIF
```

(1) ENDIF

41.4.8.4 Get previous state from I/O-software

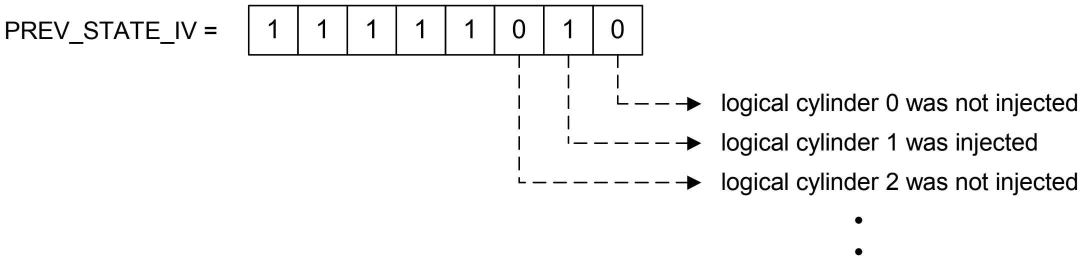
```
ACTION_INFR_GetInjPrevState (IN Cyl = x,
                             OUT LV_STATE_PREV_IV[x] = Lv_prev_state_iv)
(1) IF LV_STATE_PREV_IV[x] = 1
(1) THEN (flag fuel injection in the previous cycle)
    set bit x of PREV_STATE_IV
    CTR_STATE_PREV_PBK_IV[ID_NR_PBK_IV_PTR(Input: x)] =
          CTR_STATE_PREV_PBK_IV[ID_NR_PBK_IV_PTR(Input: x)] + 1
(1) ELSE (flag no fuel injection in the previous cycle)
    clear bit x of PREV_STATE_IV
(1) ENDIF
```

Note! x represents the current assigned active cylinder cyl_av

Pay attention to data consistency of PREV_STATE_IV and LV_STATE_PREV_IV[x]

Lv_prev_state_iv = 1 .... logical cylinder x was injected
Lv_prev_state_iv = 0 .... logical cylinder x was not injected

Note! Pay attention on data consistency of PREV_STATE_IV and LV_STATE_PREV_IV[x].



41.4.8.5 Transfer of measured injection related data from I/O-SW

After an injection was performed, the (cylinder individual) SOI, EOI and TI of that injection are indicated at the output interface as:
Note! x represents the current assigned active cylinder cyl_av

SOI_1_MES[x]	... actual performed SOI of the first injection pulse
SOI_2_MES[x]	... actual performed SOI of the second injection pulse or of the additional pulse
SOI_POST_MES[x]	... actual performed SOI of the third injection pulse
EOI_1_MES[x]	... actual performed EOI of the first injection pulse
EOI_2_MES[x]	... actual performed EOI of the second injection pulse or of the additional pulse
EOI_POST_MES[x]	... actual performed EOI of the third injection pulse
TI_1_MES[x]	... actual performed injection time of the first injection pulse
TI_2_MES[x]	... actual performed injection time of the second injection pulse or of the additional pulse
TI_POST_MES[x]	... actual performed injection time of the third injection pulse

```
ACTION_INFR_GetInjMesPulsePer (IN Cyl= x, IN Pls_nr= 0, OUT TI_1_MES[x]= Period)
ACTION_INFR_GetInjMesPulsePer (IN Cyl= x, IN Pls_nr= 1, OUT TI_2_MES[x]= Period)
ACTION_INFR_GetInjMesPulsePer (IN Cyl= x, IN Pls_nr= 2, OUT TI_POST_MES[x]= Period)
(1) IF TI_1_MES[x] > 0ms
(1) THEN
  ACTION_INFR_GetInjMesSoi (IN Cyl= x, IN Pls_nr= 0, OUT SOI_1_MES[x]= Pls_position)
  ACTION_INFR_GetInjMesEoi (IN Cyl= x, IN Pls_nr= 0, OUT EOI_1_MES[x]= Pls_position)
(1) ELSE
    SOI_1_MES[x] = 720°CRK
    EOI_1_MES[x] = 0°CRK
(1) ENDIF
(2) IF TI_2_MES[x] > 0ms
(2) THEN
  ACTION_INFR_GetInjMesSoi (IN Cyl= x, IN Pls_nr= 1, OUT SOI_2_MES[x]= Pls_position)
  ACTION_INFR_GetInjMesEoi (IN Cyl= x, IN Pls_nr= 1, OUT EOI_2_MES[x]= Pls_position)
(2) ELSE
    SOI_2_MES[x] = 720°CRK
    EOI_2_MES[x] = 0°CRK
(2) ENDIF
(3) IF TI_POST_MES[x] > 0ms
(3) THEN
  LV_TI_3_HOM_ACT = 1
  ACTION_INFR_GetInjMesSoi (IN Cyl= x, IN Pls_nr= 2, OUT SOI_POST_MES[x]= Pls_position)
  ACTION_INFR_GetInjMesEoi (IN Cyl= x, IN Pls_nr= 2, OUT EOI_POST_MES[x]= Pls_position)
(3) ELSE
    ACTION_INFR_GetInjMesPulsePer(IN Cyl=x, IN Pls_nr= 3,
                                  OUT TI_POST_MES[x]=Period)
```

```
    (4) IF TI_POST_MES[x] > 0ms
    (4) THEN
         LV_TI_3_HOM_ACT = 0
        ACTION_INFR_GetInjMesSoi (IN Cyl= x, IN Pls_nr= 3,
                                             OUT SOI_POST_MES[x]= Pls_position)
        ACTION_INFR_GetInjMesEoi (IN Cyl= x, IN Pls_nr= 3,
                                             OUT EOI_POST_MES[x]= Pls_position)
    (4) ELSE
        SOI_POST_MES[x] = 720°CRK
        EOI_POST_MES[x] = 0°CRK
    (4) ENDIF
  (3) ENDIF


 (4) IF (STATE_ERR_PBK_IV_ST_RAW[ID_NR_PBK_IV_PTR(Input: x)] AND 0x10) ≠ 0
                                                                  ...bitwise
 (4) THEN
           LV_IV_MES_VLD = 0
 (4) ELSE
           Get the information if measurements of charge and voltage of the current cylinder are valid:
           ACTION_INFR_GetInjVMesVld (IN Cyl= x,
                                         OUT LV_IV_MES_VLD= Lv_Inj_Mes_Vld)
 (4) ENDIF

 (5) IF LV_IV_MES_VLD = 1 THEN
          Get measurements of charge and voltage for all pulses from I/O-SW
          ACTION_INFR_GetInjVTrim (IN Cyl= x,
                                  OUT CHA_IV_MES_BEG_CHA = Charge)
 (5) ENDIF

 (6) IF (TI_1_MES [x] > 0) AND (LV_IV_MES_VLD = 1) THEN
           ACTION_INFR_GetInjVCha (IN Cyl= x, IN Pls_nr= 0,
                                   IN Mes_ref= END_CHA,
                                   OUT Charge= CHA_IV_MES_END_CHA

           CHA_IV_1_MES [x] = (CHA_IV_MES_END_CHA
                                             - CHA_IV_MES_BEG_CHA)
                                                           *
 FAC_CHA_IV_MES_PBK[NC_IDX_CYL_PBK_IV_REF[x]]

           ACTION_INFR_GetInjVInj (IN Cyl= x, IN Pls_nr= 0,
                                 IN Mes_ref= END_CHA,
                                 OUT V_IV_1_MES [x] = Voltage)
                      V_IV_1_MES[x] = V_IV_1_MES[x]
                           * FAC_V_IV_MES_PBK[NC_IDX_CYL_PBK_IV_REF[x]]
```

```
                calculate energy and capacity out of measured data

                EGY_IV_1_CLC [x] =  0.5*CHA_IV_1_MES[x]*V_IV_1_MES[x]
                CAPA_IV_1_CLC [x] = CHA_IV_1_MES[x]/V_IV_1_MES[x]

  (6) ELSE

                CHA_IV_1_MES [x] = 0
                 V_IV_1_MES [x] = 0
                EGY_IV_1_CLC [x] = 0
                CAPA_IV_1_CLC [x] = 0
  (6) ENDIF

  (7) IF TI_2_MES [x] > 0 AND (LV_IV_MES_VLD = 1) THEN

                ACTION_INFR_GetInjVCha (IN Cyl= x, IN Pls_nr= 1,
                                              IN Mes_ref= END_CHA,
                                              OUT Charge= CHA_IV_MES_END_CHA

                CHA_IV_2_MES [x] = (CHA_IV_MES_END_CHA
                                                        - CHA_IV_MES_BEG_CHA)
                                                                            *
FAC_CHA_IV_MES_PBK[NC_IDX_CYL_PBK_IV_REF[x]]

                ACTION_INFR_GetInjVInj (IN Cyl= x, IN Pls_nr= 1,
                                            IN Mes_ref= END_CHA,
                                            OUT V_IV_2_MES [x] = Voltage)
                        V_IV_2_MES[x] = V_IV_2_MES[x]
                                             * FAC_V_IV_MES_PBK[NC_IDX_CYL_PBK_IV_REF[x]]

                calculate energy and capacity out of measured data

                EGY_IV_2_CLC [x] =  0.5*CHA_IV_2_MES[x]*V_IV_2_MES[x]
                CAPA_IV_2_CLC [x] = CHA_IV_2_MES[x]/V_IV_2_MES[x]


  (7) ELSE

                CHA_IV_2_MES [x] = 0
                V_IV_2_MES [x] = 0
                EGY_IV_2_CLC [x] = 0
                CAPA_IV_2_CLC [x] = 0
  (7) ENDIF


  (8) IF TI_POST_MES [x] > 0 AND (LV_IV_MES_VLD = 1) THEN

      (9) IF LV_TI_3_HOM_ACT = 1

                ACTION_INFR_GetInjVCha (IN Cyl= x, IN Pls_nr= 2,
                                              IN Mes_ref= END_CHA,
                                              OUT Charge= CHA_IV_MES_END_CHA

                CHA_IV_POST_MES [x] = (CHA_IV_MES_END_CHA
                                                            - CHA_IV_MES_BEG_CHA)
                                                                            *
FAC_CHA_IV_MES_PBK[NC_IDX_CYL_PBK_IV_REF[x]]
```

```
            ACTION_INFR_GetInjVInj (IN Cyl= x, IN Pls_nr= 2,
                                    IN Mes_ref= END_CHA,
                                    OUT V_IV_POST_MES [x] = Voltage)
                                V_IV_POST_MES[x] = V_IV_POST_MES[x]
                                          * FAC_V_IV_MES_PBK[NC_IDX_CYL_PBK_IV_REF[x]]

        (9) ELSE
            ACTION_INFR_GetInjVCha (IN Cyl= x, IN Pls_nr= 3,
                                    IN Mes_ref= END_CHA,
                                    OUT Charge= CHA_IV_MES_END_CHA

            CHA_IV_POST_MES [x] = (CHA_IV_MES_END_CHA
                                                   - CHA_IV_MES_BEG_CHA)
                                                                *
    FAC_CHA_IV_MES_PBK[NC_IDX_CYL_PBK_IV_REF[x]]

            ACTION_INFR_GetInjVInj (IN Cyl= x, IN Pls_nr= 3,
                                    IN Mes_ref= END_CHA,
                                    OUT V_IV_ POST _MES [x] = Voltage)
                                V_IV_POST_MES[x] = V_IV_POST_MES[x]
                                          * FAC_V_IV_MES_PBK[NC_IDX_CYL_PBK_IV_REF[x]]

        (9)ENDIF

            calculate energy and capacity out of measured data

            EGY_IV_POST_CLC [x] = 0.5*CHA_IV_POST_MES[x]*V_IV_POST_MES[x]
            CAPA_IV_POST_CLC [x] = CHA_IV_POST_MES[x]/V_IV_POST_MES[x]

    (8) ELSE

            CHA_IV_POST_MES [x] = 0
            V_IV_POST_MES [x] = 0
            EGY_IV_POST_CLC [x] = 0
            CAPA_IV_POST_CLC [x] = 0
    (8) ENDIF
```

41.4.8.6 Transfer of injection valve diagnosis I/O-SW raw values

```
        ACTION_INFR_GetElDiagIVCyl(
                    IN Cyl = x,
                    INOUT CDN_DIAG_IV_RAW[x] = Cdn_diag = CDN_DIAG_IV_REQ[x],
                    OUT STATE_ERR_IV_CYL_RAW[x] = Err_diag)
```

41.4.9 Synchronized start

Application conditions:

Activation: at every engine state

Deactivation: -

Initialization: -

Recurrence: 10ms

Note! This task has to be called separately in calculation order after IGRE.

Formula Section:

```
(1)IF (LV_ST_INJ = 0 AND LV_ST_INJ_REQ = 1) OR
      (STATE_INJ_DR = "DISABLED"
        AND LV_INJ_REST_ENA = 1
        AND LV_INJ_DEAC_BACK_ENG = 0)
    (2) IF (STATE_INJ_UPD_TRM = $2^{(NC\_CYL\_NR)} - 1$)
          AND (STATE_INJ_CRASH_ACT = 00H)
          AND (LV_SYN_ENG = 1)
          AND (STATE_PBK_IV_INI = "INIT_FINISHED")
            Check if at least one data set was sent to I/O-SW before starting the injection.
    (2)THEN
        ACTION_INFR_SetInjMode (IN Injection_mode= DISABLE)
        ACTION_INFR_SetInjMode (IN Injection_mode= SEQUENTIAL)
        STATE_INJ_DR = "RUN"
        Please note, that this step was necessary to prevent a mode switch from IV test mode to sequential
        injection mode directly.

        LV_ST_INJ = 1

    (2) ENDIF

(1)ENDIF
```

```
# IF NC_STATE_STST_ENA = 1
# THEN
```

41.4.10 ECU-state-transition from synchronous to asynchronous state

Application conditions:

Activation: at ECU-state-transition from SYN_ENG_IGK_ON to ENG_RUN (SYN_ON to ASYN_ON)

Deactivation: -

Initialization: -

Recurrence: every ECU-state-transition from SYN_ENG_IGK_ON to ENG_RUN

(SYN_ON to ASYN_ON)

Formula Section:

```
LV_INJ_DEAC_BACK_ENG = 1
ACTION_INFR_SetInjMode (IN Injection_mode= DISABLE)
STATE_INJ_DR = "DISABLED"
```

41.4.11 ECU-state-transition from asynchronous to synchronous state

Application conditions:

Activation: at ECU-state-transition from ENG_RUN to SYN_ENG_IGK_ON (ASYN_ON to SYN_ON)

Deactivation: -

Initialization: -

Recurrence: every ECU-state-transition from ENG_RUN to SYN_ENG_IGK_ON (ASYN_ON to SYN_ON)

Formula Section:

```
IF LV_INJ_DEAC_BACK_ENG = 1
THEN
        Calculate section 41.4.6 (Start initialization)
    LV_ST_INJ = 0
    LV_INJ_DEAC_BACK_ENG = 0
ENDIF
# ENDIF
```

41.4.12 Ignition key OFF

Application conditions:

Activation: at transition of LV_IGK from 1 -> 0

Deactivation: -

Initialization: -

Recurrence: every transition of LV_IGK from 1 -> 0

Formula Section:

STATE_PBK_IV_INI = "REINIT_PENDING"

LV_INJ_DEAC_BACK_ENG = 0

41.4.13 Ignition key on

Application conditions:

Activation: at ignition key on event (LV_IGK = 0 -> 1)

Deactivation: -

Initialization: -

Recurrence: once at every ignition key on event

This chapter describes the procedures required at ignition key on to ensure an accurate engine restart at running engine.

Note: This section has to be calculated before all other time and segment synchronous sections which are dependent on LV_IGK!

Formula Section:

```
(1)IF (STATE_PBK_IV_INI = "REINIT_PENDING")
(1)THEN
        Calculate section 41.4.15 (Disable injection driver functionality)
(1)ENDIF
```

41.4.14 Engine stop

Application conditions:

Activation: at transition of LV_ES from 0 -> 1

Deactivation: -

Initialization: -

Recurrence: every transition of LV_ES from 0 -> 1

Formula Section:

Calculate section 41.4.15 (Disable injection driver functionality)

```
LV_INJ_REST_ENA = 0
LV_INJ_DEAC_BACK_ENG = 0
```

41.4.15 Disable injection driver functionality

Note: Calculate this section only if called by another section

Formula Section:

This chapter describes the engine stop procedure as required for this specification.
It is called either at engine run to engine stop event or at ignition key off event

```
(1) FOR x = 0 TO (NC_CYL_NR –1) DO:
        INJ_MOD[x] = "DISABLE"
(1) ENDFOR
LV_ST_INJ = 1
LV_INJ_UPD_CLC_ENA = 0
(1) IF STATE_INJ_CRASH_ACT = 00H
(1) THEN
      ACTION_INFR_SetInjMode (IN Injection_mode= DISABLE)
      STATE_INJ_DR = "DISABLED"
(1) ENDIF
 PREV_STATE_IV = 0      (No Injection is performed by I/O-SW from this time)
```

Do the following calculations for all cylinders

```
(2) FOR x = 0 TO (NC_CYL_NR –1) DO:
        ACTION_INJR_SetStateInjUpdEna(IN Cyl = x, State_Inj_Upd_Ena = 0)
        LV_STATE_PREV_IV[x] = 0
        LV_INJ_MPLP_CYL[x] = 0
        SOI_1_MES[x] = 720° CRK
        SOI_2_MES[x] = 720° CRK
        SOI_POST_MES[x] = 720° CRK
        EOI_1_MES[x] = 0° CRK
        EOI_2_MES[x] = 0° CRK
        EOI_POST_MES[x] = 0° CRK
        TI_1_MES[x] = 0 ms
        TI_2_MES[x] = 0 ms
        TI_POST_MES[x] = 0 ms
```

```
        CHA_IV_1_MES [x] = 0
        V_IV_1_MES [x] = 0
        EGY_IV_1_CLC [x] = 0
        CAPA_IV_1_CLC [x] = 0

        CHA_IV_2_MES [x] = 0
        V_IV_2_MES [x] = 0
        EGY_IV_2_CLC [x] = 0
        CAPA_IV_2_CLC [x] = 0

        CHA_IV_POST_MES [x] = 0
        V_IV_POST_MES [x] = 0
        EGY_IV_POST_CLC [x] = 0
        CAPA_IV_POST_CLC [x] = 0
(2) ENDFOR
```

Do the following calculations for all power stage banks

```
(3) FOR p = 0 TO (NC_PBK_IV_NR -1) DO:
        STATE_ERR_PBK_IV_RAW[p] = 0
        CDN_DIAG_PBK_IV_RAW[p] = 0
        CTR_STATE_PREV_PBK_IV[p] = 0
        LV_STATE_PREV_PBK_IV[p] = 0
(3) ENDFOR
```

41.4.16 Injection valve power stage diagnosis

Application conditions:

Activation:	every engine state
Deactivation:	-
Initialization:	-
Recurrence:	segment synchronous

Formula Section:

This chapter describes the procedure to get injection power stage diagnosis information from I/O-SW.

```
(1) IF (STATE_PBK_IV_INI = "INIT_FINISHED")
        AND
        (LV_ST_INJ = 1)
        AND
        (LV_INJ_UPD_CLC_ENA = 1 AND STATE_INJ_UPD_TRM = (2^NC_CYL_NR) -1)
(1) THEN
   (2) IF (SEG_NR = 0)
   (2) THEN
      (1) FOR p = 0 TO (NC_PBK_IV_NR -1) DO:
        CDN_DIAG_PBK_IV_RAW[p] = 0x07
        Disable_task_interruption
        ACTION_INFR_GetInjOnlineErr (IN Bank = p,
                                OUT STATE_ERR_PBK_IV_RAW[p] = Err_online)
        (3) IF CTR_STATE_PREV_PBK_IV[p] > 0
        (3) THEN
              CTR_STATE_PREV_PBK_IV[p] = 0
              LV_STATE_PREV_PBK_IV[p] = 1
              CDN_DIAG_PBK_IV_RAW[p] = CDN_DIAG_PBK_IV_RAW[p] OR 0x08 ...bitwise
        (3) ELSE
              LV_STATE_PREV_PBK_IV[p] = 0
        (3) ENDIF
        Enable_task_interruption
      (1) ENDFOR
   (2) ENDIF
(1) ELSE
   (2) FOR p = 0 TO (NC_PBK_IV_NR -1) DO:
        STATE_ERR_PBK_IV_RAW[p] = 0
        CDN_DIAG_PBK_IV_RAW[p] = 0
```

```
            CTR_STATE_PREV_PBK_IV[p] = 0
            LV_STATE_PREV_PBK_IV[p] = 0
        (2) ENDFOR
    (1) ENDIF
```

41.4.17 Injector test mode

This chapter describes the procedure for actuator tests by diagnostic tester.

Application conditions:

Activation:	LV_ES = 1
Deactivation:	-
Initialization:	at EXIT_ST event: CTR_TEST_MOD_IV[NC_CYL_NR] = 0
Recurrence:	1 s

Formula Section:

```
(1) IF     (LV_IV_TEST_MOD_AUTH =1)
    AND    (N_32 <= NC_N_MAX_IV_TEST_MOD)
    AND    (CTR_TEST_MOD_IV[NR_CYL_IV_TEST_MOD] < NC_CTR_MAX_IV_TEST_MOD)
    AND    ((INJ_MOD[NR_CYL_IV_TEST_MOD] AND NC_INJ_MOD_MASK_2) =
           NC_INJ_MOD_DI)                                        ... bitwise
(1) THEN
    CTR_TEST_MOD_IV[NR_CYL_IV_TEST_MOD] =
    CTR_TEST_MOD_IV[NR_CYL_IV_TEST_MOD] + 1

    ACTION_INFR_SetInjMode (IN Injection_mode= STATIC)
    STATE_INJ_DR = "RUN"
    ACTION_INFR_SetInjChargePer (IN Cyl = NR_CYL_IV_TEST_MOD, IN Pls_nr = 0,
                    IN Period = (T_CHA_PER_HOM_1[NR_CYL_IV_TEST_MOD]))
    ACTION_INFR_SetInjDischargePer2 (IN Cyl = NR_CYL_IV_TEST_MOD,
           IN Pls_nr = 0, IN Period = T_CHA_PER_HOM_1[NR_CYL_IV_TEST_MOD])
    ACTION_INFR_SetInjDischargePer1 (IN Cyl = NR_CYL_IV_TEST_MOD,
           IN Pls_nr = 0, IN Period = 0)
    ACTION_INFR_SetInjDischargeDly (IN Cyl = NR_CYL_IV_TEST_MOD,
           IN Pls_nr = 0, IN Period = 0)
    ACTION_INFR_ReqInjUpdate (IN Cyl = x, IN Mode = IMMEDIATE)
    ACTION_INFR_ReqInjPulse (IN Cyl= NR_CYL_IV_TEST_MOD,
                             IN Pulse_per= NC_TI_TEST_PLS_PER)
(1) ENDIF
```

41.4.18 Crash signal reaction of injection driver

Application conditions:

Activation: every engine state

Deactivation: -

Initialization: at reset: T_INJ_CRASH_ACT = 0
STATE_INJ_CRASH_ACT = 00H

Recurrence: 10 ms

This chapter describes the behavior of the injection driver during vehicle crash. Under crash conditions, all selected injectors (NC_IV_CRASH) are open for a predefined period, to drain off the fuel from the rail into the cylinder. This will reduce the danger of fire or explosion, if the rail will be damaged.

Definition of STATE_INJ_CRASH_ACT:

00H = no request received;
The injection driver works under normal conditions.

01H = vehicle crash reaction request received by LV_INJ_CRASH_ACT = 1;
The injection driver will be shut down.

02H = injection driver crash reaction acknowledged;
The selected injectors are opened for NC_TI_CRASH period.

03H = injection driver crash reaction finished;
All working injectors are closed. The driver is under hold conditions.

Formula Section:

```
(1) IF (STATE_INJ_CRASH_ACT = 01H) AND (STATE_PBK_IV_INI = "INIT_FINISHED")
(1) THEN
    ACTION_INFR_SetInjMode (IN Injection_mode= STATIC)
    STATE_INJ_DR = "RUN"
    (1) FOR x = 0 TO (NC_CYL_NR –1) DO:
        ACTION_INFR_SetInjChargePer     (IN Cyl = x, IN Pls_nr = 0,
                                         IN Period = T_CHA_PER_HOM_1[x])
        ACTION_INFR_SetInjDischargePer2 (IN Cyl = x, IN Pls_nr = 0,
                                         IN Period = T_CHA_PER_HOM_1[x])
        ACTION_INFR_SetInjDischargePer1 (IN Cyl = x, IN Pls_nr = 0, IN Period = 0)
        ACTION_INFR_SetInjDischargeDly (IN Cyl = x, IN Pls_nr = 0, IN Period = 0)
        (2) IF bit x of NC_IV_CRASH = 1
        (2) THEN
            ACTION_INFR_ReqInjUpdate (IN Cyl = x, IN Mode = IMMEDIATE)
            ACTION_INFR_ReqInjPulse (IN Cyl= x, IN Pulse_per = MIN(470 ms, NC_TI_CRASH))
        (2) ENDIF
    (1) ENDFOR
    STATE_INJ_CRASH_ACT = 02H
(1) ENDIF
```

```
(3) IF (LV_INJ_CRASH_ACT = 1) AND (STATE_INJ_CRASH_ACT = 00H)
(3) THEN
    Disable_task_interruption
    STATE_INJ_CRASH_ACT = 01H
    (2) FOR x = 0 TO (NC_CYL_NR –1) DO:
        INJ_MOD[x] = "DISABLE"
    (2) ENDFOR
    LV_ST_INJ = 1
    Enable_task_interruption
    ACTION_INFR_SetInjMode (IN Injection_mode= DISABLE)
    STATE_INJ_DR = "DISABLED"
(3) ENDIF
(4) IF (STATE_INJ_CRASH_ACT = 02H)          AND (T_INJ_CRASH_ACT >= NC_TI_CRASH)
(4) THEN
    STATE_INJ_CRASH_ACT = 03H
    ACTION_INFR_SetInjMode (IN Injection_mode= DISABLE)
    STATE_INJ_DR = "DISABLED"
(4) ENDIF
(5) IF STATE_INJ_CRASH_ACT = 02H
(5) THEN
    T_INJ_CRASH_ACT = T_INJ_CRASH_ACT + 10 ms
    (6) IF (T_INJ_CRASH_ACT MODULO 500ms) = 0
    (6) THEN
        (3) FOR x = 0 TO (NC_CYL_NR –1) DO:
            (7) IF bit x of NC_IV_CRASH = 1
            (7) THEN
                ACTION_INFR_ReqInjPulse (IN Cyl= x, IN Pulse_per =
                                    MIN( 480 ms, (NC_TI_CRASH  - (T_INJ_CRASH_ACT - 10 ms))))
            (7) ENDIF
        (3) ENDFOR
    (6) ENDIF
(5) ENDIF
```

41.4.19 Description of exported ACTIONS

41.4.19.1 Description of ACTION_INJR_IgnUpdate

ACTION_INJR_IgnUpdate(IN <Cyl>, IN <Iga_Igc_H_Rng>)					
This action sets the cylinder individual ignition angle for mode switch and data update of <Cyl> to <Iga_Igc_H_Rng>..					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cyl	IN	0 ... 7	0 ... 7	1	1
This action affects the injection mode of cylinder Cyl.					
Iga_Igc_H_Rng	IN	FA60 ... 5A0H	-90 ... 90	0.0625	[°CRK]
This parameter defines the new ignition angle for homogeneous single injection mode.					

Formula Section:

IGA_IGC_H_RNG_ACT[Cyl] = Iga_Igc_H_Rng

STATE_UPD_IGN[Cyl] = "TRIG_EXT"

41.4.20 Subroutines

This section defines the subroutines. The following chapters are only calculated if called by another section like a function call. The input parameters are mentioned at the function calls.

41.4.20.1 Configure Injection

Note: Calculate this section only if called by another section.

```
TI_ACT_SWI[x] = 0
TI_ACT_UPD[x] = 0

Disable post pulse
LV_INJ_MPLP_CYL[x] = 0 (post injection disabled)
ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 3, IN Lv_pls = 0)

Homogeneous combustion mode:
(1)IF INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_HOM
(1)THEN – homogeneous combustion mode
  (2)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2) ≠ NC_INJ_MOD_SINGLE
  (2)THEN - homogeneous mode multi injection
       Settings for first pulse
       ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 0, IN Pls_type = SOI_TI)
       ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 0,
                                                      IN Upd_mode = NONE)
            ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 0, IN Lv_pls = 1)
       Settings for second pulse
       ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 1, IN Pls_type = EOI_TI)
       (3)IF LC_IV_ADD_PLS_MPLH_ENA = 1
       (3)THEN
         ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 1
                                                      IN Upd_mode = UPDATE)
            (3)ELSE
         ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 1
                                                      IN Upd_mode = NONE)
            (3)ENDIF
            ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 1, IN Lv_pls = 1)
       Settings for third pulse
       (7)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2) = NC_INJ_MOD_MULTI_PLS3
       (7)THEN
         ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 2, IN Pls_type = EOI_TI)
         ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 2
```

```
                                                        IN Upd_mode = NONE)
                  ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 2, IN Lv_pls =
  1)
      (7)ELSE
            ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 2, IN Lv_pls = 0)
      (7)ENDIF
  (2)ELSE - homogeneous mode single injection
      Settings for first pulse
      ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 0, IN Pls_type = SOI_TI)
      ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 0,
                                                        IN Upd_mode = NONE)
      Enable first pulse
      ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 0, IN Lv_pls = 1)
      Disable third pulse
      ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 2, IN Lv_pls = 0)
      (9)IF LC_IV_ADD_PLS_SNGH_ENA = 1
      (9)THEN
            Settings makeup pulse
            ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 1,
                                                        IN Pls_type = EOI_TI)
            (10)IF (LC_INJ_PLS_UPD_MOD = AUTONOMOUS)
            (10)THEN
                  ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 1,
                                                   IN Upd_mode = AUTONOMOUS)
            (10)ELSE
                  ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 1,
                                                   IN Upd_mode = IMMEDIATE)
            (10)ENDIF
            Enable second (makeup) pulse
            ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 1, IN Lv_pls = 1)
      (9)ELSE
            Disable second (makeup) pulse
            ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 1, IN Lv_pls = 0)
      (9)ENDIF
  (2)ENDIF
(1)ENDIF (HOM mode)
```

```
(11)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1) = NC_INJ_MOD_S
(11)THEN - stratified combustion mode
    (12)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2) ≠ NC_INJ_MOD_SINGLE
    (12)THEN – stratified multi injection
          Settings for first pulse
          ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 0, IN Pls_type = EOI_TI)
          ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 0,
                                                                IN Upd_mode = NONE)
          Settings for second pulse
          ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 1, IN Pls_type = T_TI)
          ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 1,
                                                                IN Upd_mode = NONE)
          Enable first pulse
          ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 0, IN Lv_pls = 1)
          Enable second pulse
          ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 1, IN Lv_pls = 1)
          (13)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2)

                              = NC_INJ_MOD_MULTI_PLS3
          (13)THEN – stratified triple injection
               Settings for third pulse
               ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 2, IN Pls_type = T_TI)
               ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 2,
                                                                IN Upd_mode = NONE)
               Enable third pulse
               ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 2, IN Lv_pls = 1)
          (13)ELSE
               Disable third pulse
               ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 2, IN Lv_pls = 0)
          (13)ENDIF
    (12)ELSE - Stratified mode single injection
        Settings for first pulse
        ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 0, IN Pls_type = EOI_TI)
        ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 0, IN Upd_mode = NONE)
        Enable first pulse
        ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 0, IN Lv_pls = 1)
        Disable second pulse
```

```
        ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 1, IN Lv_pls = 0)
        Disable third pulse
        ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 2, IN Lv_pls = 0)
    (12)ENDIF
(11)ENDIF (end stratified mode)
```

Homogeneous-stratified combustion mode

```
(13)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1) = NC_INJ_MOD_HOMS
(13)THEN – homs mode
    (14)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2) ≠ NC_INJ_MOD_SINGLE
    (14)THEN – homs multi injection
        Settings for first pulse
        ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 0, IN Pls_type = SOI_TI)
        ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 0,
                                       IN Upd_mode = NONE)
        Settings for second pulse
        ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 1, IN Pls_type = EOI_TI)
        ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 1;
                                       IN Upd_mode = NONE)
        Enable first pulse
        ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 0, IN Lv_pls = 1)
        Enable second pulse
        ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 1, IN Lv_pls = 1)
        (15)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2) =
                                       NC_INJ_MOD_MULTI_PLS3
        (15)THEN – homs triple injection
            Settings for third pulse
            ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 2,
                                        IN Pls_type = EOI_TI)
            ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 2,
                                           IN Upd_mode = NONE)
            Enable third pulse
            ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 2, IN Lv_pls = 1)
        (15)ELSE
            Disable third pulse
            ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 2, IN Lv_pls = 0)
        (15)ENDIF
    (14)ELSE – homs single injection
        Settings for first pulse
```

```
        ACTION_INFR_SetInjPulseType(IN Cyl = x, IN Pls_nr = 0, IN Pls_type = SOI_TI)
        ACTION_INFR_SetInjPulseUpdMode(IN Cyl = x, IN Pls_nr = 0,
                                                    IN Upd_mode = NONE)
        Enable first pulse
        ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 0, IN Lv_pls = 1)
        Disable second pulse
        ACTION_INFR_SetInjPulseEnable(IN Cyl = x, IN Pls_nr = 1, IN Lv_pls = 0)
        Disable third pulse
        ACTION_INFR_SetInjPulseEnable (IN Cyl = x, IN Pls_nr = 2, IN Lv_pls = 0)
    (14)ENDIF
(13)ENDIF (end HOMS mode)
```

41.4.20.2 Update injection data

Note: Calculate this section only if called by another section.

```
ACTION_INFR_SetInjChargeCur(IN Cyl = x, IN Ducy = EGY_STEP_INJ_CHA_GRD[x])
ACTION_INFR_SetInjDischargeCur(IN Cyl= x, IN Ducy= EGY_STEP_INJ_DCHA_GRD[x])
(1)IF LV_INJ_MPLP_CYL[x] = 1 (post injection enabled)
(1)THEN
    ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 3,
                                IN Period = T_CHA_PER_POST)
    ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 3,
                                IN Period = T_DCHA_PER_POST)
    ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 3, IN Period = 0)
    ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 3, IN Period = 0)
    ACTION_INFR_SetInjSOILim(IN Cyl= x, IN Packet_nr= 1, IN Soi_lim= SOI_LIM_POST)
    ACTION_INFR_SetInjEOILim(IN Cyl= x, IN Packet_nr= 1, IN Eoi_lim= EOI_LIM_POST)
    ACTION_INFR_SetInjPulsePosition(IN Cyl=x, IN Pls_nr= 3,
                                IN Pls_position=
                                SOI_POST_INJ[x])
    ACTION_INFR_SetInjPulseDuration(IN Cyl=x, IN Pls_nr= 3,
                                IN Pls_duration= TI_POST_INJ[x])
(1)ENDIF

Homogeneous combustion mode
(1)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1) = NC_INJ_MOD_HOM
(1)THEN - Homogeneous mode
    (2)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2) ≠ NC_INJ_MOD_SINGLE
    (2)THEN – homogeneous mode multi injection
        (2a)IF LC_IV_ADD_PLS_MPLH_ENA
        (2a)THEN
            ACTION_INFR_SetInjPulseUpdFac(IN Cyl= x, IN Pls_nr= 1,
                                IN Fac_add_pls = FAC_ADD_PULSE)
            ACTION_INFR_SetInjPlsUpdMinPer(IN Cyl= x, IN Ti_min = TI_ADD_PULSE_MIN)
        (2a)ENDIF
        ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 1,
                                IN Period = T_CHA_PER_HOM_2[x])
        ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 1,
                                IN Period = T_DCHA_PER_HOM_2_2[x])
        ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 1,
                                IN Period = T_DCHA_PER_HOM_2_1[x])
```

```
ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 1,
                               IN Period = T_DCHA_DLY_HOM_2[x])
ACTION_INFR_SetInjPulsePosition(IN Cyl= x, IN Pls_nr= 1,
                                IN Pls_position= EOI_2_HOM[x])
ACTION_INFR_SetInjPulseDuration(IN Cyl= x, IN Pls_nr= 1,
                                IN Pls_duration = TI_2_HOM[x])
(3)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2)
                              = NC_INJ_MOD_MULTI_PLS3
(3)THEN – homogeneous mode triple injection
    ACTION_INFR_SetInjChargePer (IN Cyl = x, IN Pls_nr = 2,
                                 IN Period = T_CHA_PER_HOM_3[x])
    ACTION_INFR_SetInjDischargePer2 (IN Cyl = x, IN Pls_nr = 2,
                                     IN Period = T_DCHA_PER_HOM_3_2[x])
ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 2,
                                IN Period = T_DCHA_PER_HOM_3_1[x])
    ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 2,
                                   IN Period = T_DCHA_DLY_HOM_3[x])
    ACTION_INFR_SetInjPulsePosition (IN Cyl=x, IN Pls_nr= 2,
                                     IN Pls_position= EOI_3_HOM[x])
    ACTION_INFR_SetInjPulseDuration (IN Cyl=x, IN Pls_nr= 2,
                                     IN Pls_duration= TI_3_HOM[x])
(3)ENDIF
SOI_EOI_ACT_UPD[x] = EOI_2_HOM[x]
        LV_SOI_ACT_UPD[x] = 0
        TI_ACT_UPD[x] = TI_2_HOM[x]
(2)ELSE – homogeneous mode single injection
        Handling of makeup pulse
        (4)IF LC_IV_ADD_PLS_SNGH_ENA = 1
        (4)THEN
                ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 1,
                                            IN Period = T_CHA_PER_HOM_1[x])
                ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 1,
                                                IN Period = T_CHA_PER_HOM_1[x])
ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 1, IN Period = 0)
    ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 1, IN Period = 0)
    (5)IF (LC_INJ_PLS_UPD_MOD = AUTONOMOUS)
    (5)THEN
        EOI_INJ_UPD_PSN = IP_EOI_INJ_UPD_PSN(N_32)
        ACTION_INFR_SetInjPulsePosition(IN Cyl=x, IN Pls_nr= 1,
                                IN Pls_position = EOI_INJ_UPD_PSN)
    (5)ELSE
        ACTION_INFR_SetInjPulsePosition(IN Cyl=x, IN Pls_nr= 1,
```

```
                    IN Pls_position = MAX(C_SOI_MIN_UPD_PLS_SNGH; EOI_LIM_HOM))
        (5)ENDIF
        ACTION_INFR_SetInjPulseUpdFac(IN Cyl= x, IN Pls_nr= 1,
                                                  IN Fac_add_pls = FAC_ADD_PULSE)
        ACTION_INFR_SetInjPlsUpdMinPer(IN Cyl= x, IN Ti_min = TI_ADD_PULSE_MIN)
        (5a)IF LV_EGY_RNG_IV_PLS_1_HOM[x] = 1
        (5a)THEN
            TI_1_ADD_DLY_HOM =
    IP_TI_ADD_DLY_HPDI_EGY_H(PRS_DEC_INJ_1_HOM[x])
        (5a)ELSE
            TI_1_ADD_DLY_HOM =
    IP_TI_ADD_DLY_HPDI_EGY_L(PRS_DEC_INJ_1_HOM[x])
        (5a)ENDIF
        ACTION_INFR_SetInjPlsUpdOffset(IN Cyl= x, IN Upd_offset = TI_1_ADD_DLY_HOM)
        SOI_EOI_ACT_UPD[x] = MAX(C_SOI_MIN_UPD_PLS_SNGH; EOI_LIM_HOM)
        LV_SOI_ACT_UPD[x] = 1
            (4)ELSE
        SOI_EOI_ACT_UPD[x] = SOI_1_HOM[x]
        LV_SOI_ACT_UPD[x] = 1
            (4)ENDIF
    (2)ENDIF – homogeneous single injection
    Calculations for all homogeneous modes
ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 0,
                                            IN Period = T_CHA_PER_HOM_1[x])
ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 0,
                                          IN Period = T_DCHA_PER_HOM_1_2[x])
ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 0,
                                          IN Period = T_DCHA_PER_HOM_1_1[x])
ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 0,
                                          IN Period = T_DCHA_DLY_HOM_1[x])
LV_EGY_RNG_IV_PLS_1_ACT[x] = LV_EGY_RNG_IV_PLS_1_HOM[x]
(2a)IF CRK_WIN_SEL_IGN_INJ[x] != "RTD_WIN"
(2a)THEN
            LV_EGY_RNG_IV_PLS_1_ACK[x] = LV_EGY_RNG_IV_PLS_1_ACT[x]
(2a)ENDIF
Service tool intervention
(6)IF LV_TI_EXT_ADJ[x] = 0
(6)THEN – no service tool intervention
```

```
        ACTION_INFR_SetInjPulsePosition (IN Cyl=x, IN Pls_nr= 0,
                                         IN Pls_position= SOI_1_HOM[x])
        ACTION_INFR_SetInjPulseDuration (IN Cyl=x, IN Pls_nr= 0,
                                         IN Pls_duration= TI_1_HOM[x])
            SOI_EOI_ACT_SWI[x] = SOI_1_HOM[x]
            LV_SOI_ACT_SWI[x] = 1
            TI_1_HOM_ACT[x] = TI_1_HOM[x]
            IDX_TI_1_HOM_ACT[x] = IDX_TI_1_HOM_CLC[x]
    (6)ELSE - service tool intervention
        ACTION_INFR_SetInjPulsePosition (IN Cyl= x, IN Pls_nr= 0,
                                         IN Pls_position= C_SOI_EXT_ADJ)
        ACTION_INFR_SetInjPulseDuration (IN Cyl= x, IN Pls_nr= 0,
                                         IN Pls_duration= TI_EXT_ADJ[x])
            SOI_EOI_ACT_SWI[x] = C_SOI_EXT_ADJ
            LV_SOI_ACT_SWI[x] = 1
            TI_1_HOM_ACT[x] = TI_EXT_ADJ[x]
            IDX_TI_1_HOM_ACT[x] = IDX_TI_1_HOM_CLC[x]
    (6)ENDIF
    ACTION_INFR_SetInjSOILim (IN Cyl= x, IN Packet_nr= 0, IN Soi_lim= SOI_LIM)
    ACTION_INFR_SetInjEOILim (IN Cyl= x, IN Packet_nr= 0, IN Eoi_lim= EOI_LIM_HOM)
(1)ENDIF (HOM mode)

Stratified combustion mode
(9)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1) = NC_INJ_MOD_S
(9)THEN
    Set limits
    ACTION_INFR_SetInjSOILim(IN Cyl= x, IN Packet_nr= 0,IN Soi_lim= SOI_LIM)
    ACTION_INFR_SetInjEOILim(IN Cyl= x, IN Packet_nr= 0, IN Eoi_lim= 0° CRK)
    (10)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2) ≠ NC_INJ_MOD_SINGLE
    (10)THEN – stratified multi injeciton
        Settings for second pulse:
        Set injector needle lift by charge and discharge time
        ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 1,
                                    IN Period = T_CHA_PER_S_2[x])
        ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 1,
                                        IN Period = T_DCHA_PER_S_2_2[x])
        ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 1,
                                        IN Period = T_DCHA_PER_S_2_1[x])
        ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 1,
            IN Period = T_DCHA_DLY_S_2[x])
        Set time and phasing
```

```
                ACTION_INFR_SetInjPulsePosition(IN Cyl= x, IN Pls_nr= 1,
                                                IN Pls_position= T_DLY_1_2_S_EXT[x])
                ACTION_INFR_SetInjPulseDuration(IN Cyl= x, IN Pls_nr= 1,
                                                IN Pls_duration= TI_2_S[x])
                (11)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2)
                                                = NC_INJ_MOD_MULTI_PLS3
                (11)THEN – stratified triple injection
                    ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 2,
                                                IN Period = T_CHA_PER_S_3[x])
                    ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 2,
                                                IN Period = T_DCHA_PER_S_3_2[x])
                          ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 2,
                                                IN Period = T_DCHA_PER_S_3_1[x])
                          ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 2,

                     IN Period = T_DCHA_DLY_S_3[x])
                    ACTION_INFR_SetInjPulsePosition(IN Cyl=x, IN Pls_nr= 2,
                                                IN Pls_position= T_DLY_2_3_S_EXT[x])
                    ACTION_INFR_SetInjPulseDuration(IN Cyl=x, IN Pls_nr= 2,
                                                IN Pls_duration= TI_3_S[x])
                (11)ENDIF
            (10)ENDIF
            Stratified mode single injection
            Settings for first pulse Set injector needle lift by charge and discharge time
            ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 0,
                                                IN Period = T_CHA_PER_S_1[x])
            ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 0,
                                                IN Period = T_DCHA_PER_S_1_2[x])
            ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 0,
                                                IN Period = T_DCHA_PER_S_1_1[x])
            ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 0,

                          IN Period = T_DCHA_DLY_S_1[x])
            LV_EGY_RNG_IV_PLS_1_ACT[x] = LV_EGY_RNG_IV_PLS_1_S[x]
            (10a)IF CRK_WIN_SEL_IGN_INJ[x] != "RTD_WIN"
            (10a)THEN
                    LV_EGY_RNG_IV_PLS_1_ACK[x] = LV_EGY_RNG_IV_PLS_1_ACT[x]
            (10a)ENDIF
            Set time and phasing
            ACTION_INFR_SetInjPulsePosition(IN Cyl= x, IN Pls_nr= 0, IN Pls_position= EOI_1_S[x])
            ACTION_INFR_SetInjPulseDuration(IN Cyl= x, IN Pls_nr= 0, IN Pls_duration= TI_1_S[x])
            SOI_EOI_ACT_SWI[x] = EOI_1_S[x]
```

```
   LV_SOI_ACT_SWI[x] = 0
   TI_ACT_SWI[x] = TI_1_S[x]

   SOI_EOI_ACT_UPD[x] = EOI_1_S[x]
   LV_SOI_ACT_UPD[x] = 0
   TI_ACT_UPD[x] = TI_1_S[x]

(9)ENDIF (S mode)

Homogeneous Stratified mode

(11)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1) = NC_INJ_MOD_HOMS
(11)THEN

      Set limits

      ACTION_INFR_SetInjSOILim(IN Cyl= x, IN Packet_nr= 0, IN Soi_lim= SOI_LIM)

      ACTION_INFR_SetInjEOILim(IN Cyl= x, IN Packet_nr= 0, IN Eoi_lim= 0° CRK)

      (12)IF (INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_2) ≠ NC_INJ_MOD_SINGLE
      (12)THEN - Homogeneous Stratified mode multi injection

         Settings for second pulse:
         Set injector needle lift by charge and discharge time

         ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 1,
                                                  IN Period = T_CHA_PER_S_2[x])

         ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 1,
                                              IN Period = T_DCHA_PER_S_2_2[x])

         ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 1,
                                              IN Period = T_DCHA_PER_S_2_1[x])

         ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 1,

                     IN Period = T_DCHA_DLY_S_2[x])

         Set time and phasing

         ACTION_INFR_SetInjPulsePosition(IN Cyl= x, IN Pls_nr= 1,
                                                   IN Pls_position= EOI_2_HOMS[x])

         ACTION_INFR_SetInjPulseDuration(IN Cyl= x, IN Pls_nr= 1,
                                                        IN Pls_duration= TI_2_S[x])

         (13)IF (INJ_MOD[x] BiswiseAND NC_INJ_MOD_MASK_2)
                                                 = NC_INJ_MOD_MULTI_PLS3
         (13)THEN - Homogeneous Stratified mode triple injection

            ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 2,
                                                     IN Period = T_CHA_PER_S_3[x])

            ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 2,
                                                 IN Period = T_DCHA_PER_S_3_2[x])

                 ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 2,
                                              IN Period = T_DCHA_PER_S_3_1[x])

                 ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 2,

            IN Period = T_DCHA_DLY_S_3[x])
```

```
                ACTION_INFR_SetInjPulsePosition(IN Cyl=x, IN Pls_nr= 2,
                                                    IN Pls_position= EOI_3_HOMS[x])
                ACTION_INFR_SetInjPulseDuration(IN Cyl=x, IN Pls_nr= 2,
                                                    IN Pls_duration= TI_3_S[x])
            (13)ENDIF
            SOI_EOI_ACT_UPD[x] = EOI_2_HOMS[x]
            LV_SOI_ACT_UPD[x] = 0
            TI_ACT_UPD[x] = TI_2_S[x]
        (12)ELSE – homogeneous stratified mode single injection
            SOI_EOI_ACT_UPD[x] = SOI_1_HOMS[x]
            LV_SOI_ACT_UPD[x] = 1
        (12)ENDIF
        Homogeneous Stratified mode single injection and multi injection
        Settings for first pulse Set injector needle lift by charge and discharge time
        ACTION_INFR_SetInjChargePer(IN Cyl = x, IN Pls_nr = 0,
                                        IN Period = T_CHA_PER_S_1[x])
        ACTION_INFR_SetInjDischargePer2(IN Cyl = x, IN Pls_nr = 0,
                                        IN Period = T_DCHA_PER_S_1_2[x])
        ACTION_INFR_SetInjDischargePer1(IN Cyl = x, IN Pls_nr = 0,
                                        IN Period = T_DCHA_PER_S_1_1[x])
        ACTION_INFR_SetInjDischargeDly(IN Cyl = x, IN Pls_nr = 0,

                                        IN Period = T_DCHA_DLY_S_1[x])
        LV_EGY_RNG_IV_PLS_1_ACT[x] = LV_EGY_RNG_IV_PLS_1_S[x]
        (10a)IF CRK_WIN_SEL_IGN_INJ[x] != "RTD_WIN"
        (10a)THEN
            LV_EGY_RNG_IV_PLS_1_ACK[x] = LV_EGY_RNG_IV_PLS_1_ACT[x]
        (10a)ENDIF
        Set time and phasing
        ACTION_INFR_SetInjPulsePosition(IN Cyl= x, IN Pls_nr= 0,
                                        IN Pls_position= SOI_1_HOMS[x])

        ACTION_INFR_SetInjPulseDuration (IN Cyl= x, IN Pls_nr= 0, IN Pls_duration= TI_1_S[x])
        SOI_EOI_ACT_SWI[x] = SOI_1_HOMS[x]
        LV_SOI_ACT_SWI[x] = 1
    (11)ENDIF (end HOMS mode)
```

41.4.20.3 Update ignition data

Note: Calculate this section only if called by another section.

```
(1)IF INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_S
(1)THEN – stratified combustion mode
    IGA_IGC_H_RNG_ACT[x] = IGA_AV_H_RNG_S[x]
    TD_TMP_SYN = TD_S
(1)ELSE
    (2)IF INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_HOMS
    (2)THEN – homogeneous-stratified combustion mode
        IGA_IGC_H_RNG_ACT[x] = IGA_AV_H_RNG_HOMS[x]
        TD_TMP_SYN = TD_S
    (2)ELSE – homogeneous combustion mode
        IGA_IGC_H_RNG_ACT[x] = IGA_AV_H_RNG[x]
        TD_TMP_SYN = TD
    (2)ENDIF
(1)ENDIF
TD_IGC_ACT[x] = MIN(TD_LIM_MAX; MAX(NC_TD_LIM_MIN; TD_TMP_SYN + TD_AD[x]))
```

41.4.20.4 Check engine position

Note: Calculate this section only if called by another section.

This section calculates the actual engine position of the given cylinder x and the possibility of injection mode switch or data update.

LV_INH_INJ_UPD_MOD_UPD[x] = 0

Set acknowledged injector needle lift of first injection pulse to value transmitted to BSW:
LV_EGY_RNG_IV_PLS_1_ACK[x] = LV_EGY_RNG_IV_PLS_1_ACT[x]

```
CRK_PSN_INJ_BAS[x] = ((720° - PSN_ENG - CRK_INJ_BAS[0]) + (x*720°/NC_CYL_NR) +720°) MODULO 720°

(1)IF LC_IGN_UPD_STAT_ENA = 0 OR
        Bit x of STATE_INJ_UPD_ENA = 1
(1)THEN
        CRK_PSN_ENG_IGN_INJ_UPD[x] = CRK_PSN_INJ_BAS[x] –
                                     C_T_OFS_UPD_IGN_INJ_SYN * FAC_N

    Note: In case of C_T_OFS_UPD_IGN_INJ_SYN * FAC_N > CRK_PSN_INJ_BAS[x], limit
    CRK_PSN_ENG_IGN_INJ_UPD[x] to zero!

    CRK_WIN_SEL_TMP = "NO_WIN"
    CRK_WIN_SEL_PREV = CRK_WIN_SEL_IGN_INJ[x]
    (2)IF CRK_PSN_INJ_BAS[x] < CRK_PSN_STAT_WIN_ST
    (2)THEN
            CRK_WIN_SEL_TMP = "RTD_WIN"
        (2)ELSE
            (3)IF CRK_PSN_INJ_BAS[x] > CRK_PSN_STAT_WIN_END
            (3)THEN
                CRK_WIN_SEL_TMP = "ADC_WIN"
            (3)ELSE
                (3a)IF INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_HOM
                (3a)THEN
                    (4)IF INJ_MOD[x] = "SNGH"
                    (4)THEN
                        (4a)IF LV_ADD_PULSE_ENA = 0
                                OR LV_EGY_RNG_IV_PLS_1_HOM[x] != LV_EGY_RNG_IV_PLS_1_ACT[x]
                                OR LC_IV_ADD_PLS_SNGH_ENA = 0
                                OR    (TI_1_HOM[x] > TI_1_HOM_ACT[x]
                    AND IDX_TI_1_HOM_CLC[x] < IDX_TI_1_HOM_ACT[x])
                                OR    (TI_1_HOM[x] < TI_1_HOM_ACT[x]
                    AND IDX_TI_1_HOM_CLC[x] > IDX_TI_1_HOM_ACT[x])
                        (4a)THEN
```

```
                LV_INH_INJ_UPD_MOD_UPD[x] = 1
            (4a)ENDIF
        (4)ELSE
            (5)IF LV_EGY_RNG_IV_PLS_1_HOM[x] != LV_EGY_RNG_IV_PLS_1_ACT[x]
                OR (LV_EGY_RNG_IV_PLS_1_HOM[x] = 0 AND
                        LV_EGY_RNG_IV_PLS_2_HOM[x] = 1)
                OR LC_IV_ADD_PLS_MPLH_ENA = 0
                OR LV_INJ_PLS_UPD_MPLH_DEAC = 1
                OR    (TI_1_HOM[x] > TI_1_HOM_ACT[x]
        AND IDX_TI_1_HOM_CLC[x] < IDX_TI_1_HOM_ACT[x])
                OR    (TI_1_HOM[x] < TI_1_HOM_ACT[x]
        AND IDX_TI_1_HOM_CLC[x] > IDX_TI_1_HOM_ACT[x])
            (5)THEN
                LV_INH_INJ_UPD_MOD_UPD[x] = 1
            (5)ENDIF
        (4)ENDIF
    (3a)ENDIF
    (3b)IF (INJ_MOD[x] = "HOMS" OR INJ_MOD[x] = "HOMS+PLS3")
            AND LV_EGY_RNG_IV_PLS_1_S[x] != LV_EGY_RNG_IV_PLS_1_ACT[x]
    (3b)THEN
                LV_INH_INJ_UPD_MOD_UPD[x] = 1
    (3b)ENDIF
    (6)IF INJ_MOD[x] != INJ_MOD_SP[x]
        OR LV_INH_INJ_UPD_MOD_UPD[x] = 1
    (6)THEN – check window for switching to INJ_MOD_SP[x]
        LV_INJ_MOD_UPD = 0
    (6)ELSE - check window for update of INJ_MOD[x]
        LV_INJ_MOD_UPD = 1
    (6)ENDIF
    Calculate section 41.4.20.5 (check variable window (input: LV_INJ_MOD_UPD; output:
                                    LV_IGN_INJ_UPD_WIN))
    (7)IF LV_IGN_INJ_UPD_WIN = 1
    (7)THEN
        CRK_WIN_SEL_TMP = "ADC_WIN"
    (7)ENDIF
  (3)ENDIF
(2)ENDIF
CRK_WIN_SEL_IGN_INJ[x] = CRK_WIN_SEL_TMP
```

```
        (8)IF LC_IGN_UPD_STAT_ENA = 0
        (8)THEN

                (9)IF CRK_WIN_SEL_TMP = "ADC_WIN"
                        OR (CRK_WIN_SEL_TMP = "NO_WIN"
                AND CRK_WIN_SEL_PREV = "RTD_WIN")
                (9)THEN

                        STATE_UPD_IGN[x] = "TRIG_IT"

                (9)ELSE

                        STATE_UPD_IGN[x] = "NONE"

                (9)ENDIF

        (8)ELSE

                (10)IF INJ_MOD[x] != INJ_MOD_SP[x]
                (10)THEN

            (11)IF CRK_WIN_SEL_TMP = "ADC_WIN"
                                        OR (CRK_WIN_SEL_TMP != "RTD_WIN"
                                                AND CRK_PSN_INJ_BAS[x] <
CRK_PSN_STAT_IGN_UPD_ST
                                                AND CRK_PSN_INJ_BAS[x] >
CRK_PSN_STAT_IGN_UPD_END)
                        (11)THEN

                                STATE_UPD_IGN[x] = "TRIG_IT"

                        (11)ELSE

                                STATE_UPD_IGN[x] = "NONE"

                        (11)ENDIF

                (10)ELSE

                        (12)IF CRK_PSN_INJ_BAS[x] < CRK_PSN_STAT_IGN_UPD_ST
                                AND CRK_PSN_INJ_BAS[x] >
CRK_PSN_STAT_IGN_UPD_END
                        (12)THEN

                                STATE_UPD_IGN[x] = "TRIG_IT"

                        (12)ELSE

                                STATE_UPD_IGN[x] = "NONE"

                        (12)ENDIF

                (10)ENDIF

        (8)ENDIF

(1)ELSE

        CRK_WIN_SEL_IGN_INJ[x] = "NO_CLC"

        (13)IF CRK_PSN_INJ_BAS[x] < CRK_PSN_STAT_IGN_UPD_ST
                        AND CRK_PSN_INJ_BAS[x] > CRK_PSN_STAT_IGN_UPD_END
        (13)THEN

                STATE_UPD_IGN[x] = "TRIG_IT"
```

```
        (13)ELSE
                STATE_UPD_IGN[x] = "NONE"
        (13)ENDIF
(1)ENDIF
```

41.4.20.5 Check variable window

Note: Calculate this section only if called by another section.
Note: This section works like a function with LV_INJ_MOD_UPD as input and LV_IGN_INJ_UPD_WIN as output.

This section calculates, if an injection mode switch (LV_INJ_MOD_UPD = 0) or an injection data update at no mode switch (LV_INJ_MOD_UPD = 1) is possible. Therefore, the engine position must be within the "variable window". This is indicated by LV_IGN_INJ_UPD_WIN = 1.

```
LV_IGN_INJ_UPD_WIN = 0
(1)IF CRK_PSN_ENG_IGN_INJ_UPD[x] > (IGA_IGC_H_RNG_ACT[x] – CRK_INJ_BAS[0] + TD_IGC_ACT[x] * FAC_N)
(1)THEN
        (2)IF LV_INJ_MOD_UPD = 1
        (2)THEN – calculate temporary values of previous SOI for data update
                (3)IF LV_SOI_ACT_UPD[x] = 1
                (3)THEN
                        SOI_TMP_SYN_PREV = SOI_EOI_ACT_UPD[x]
                (3)ELSE
                        SOI_TMP_SYN_PREV = SOI_EOI_ACT_UPD[x] + TI_ACT_UPD[x] * FAC_N
                (3)ENDIF
        (2)ELSE – calculate temporary values of previous SOI for injection mode switch
                (4)IF LV_SOI_ACT_SWI[x] = 1
                (4)THEN
                        SOI_TMP_SYN_PREV = SOI_EOI_ACT_SWI[x]
                (4)ELSE
                        SOI_TMP_SYN_PREV = SOI_EOI_ACT_SWI[x] + TI_ACT_SWI[x] * FAC_N
                (4)ENDIF
        (2)ENDIF
        (5)IF CRK_PSN_ENG_IGN_INJ_UPD[x] > SOI_TMP_SYN_PREV
        (5)THEN
                Calculate section 41.4.20.6 (Determine new ignition and injection values for variable window)
                (6)IF CRK_PSN_ENG_IGN_INJ_UPD[x] >
                    MAX(IGA_TMP_SYN – CRK_INJ_BAS[0] + (TD_TMP_SYN + TD_AD[x]) * FAC_N;
                        SOI_TMP_SYN)
                (6)THEN – switch or update to new ignition and injection data also possible
                        LV_IGN_INJ_UPD_WIN = 1
                (6)ENDIF
        (5)ENDIF
```

(1)**ENDIF**

41.4.20.6 Determine new ignition and injection values for variable window

Note: Calculate this section only if called by another section.

This section calculates the temporary values of ignition angle, dwell time and start of injection for either mode switch (LV_INJ_MOD_UPD = 0) or data update in case of no mode switch (LV_INJ_MOD_UPD = 1).

```
(1)IF LV_INJ_MOD_UPD = 1
(1)THEN – calculate temporary values for data update
    (2)IF INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_S
    (2)THEN – stratified combustion mode
        IGA_TMP_SYN = IGA_AV_H_RNG_S[x]
        TD_TMP_SYN = TD_S
        SOI_TMP_SYN = EOI_1_S[x] + TI_1_S[x] * FAC_N
    (2)ELSE
        (3)IF INJ_MOD[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_HOMS
        (3)THEN – homogeneous-stratified combustion mode
            IGA_TMP_SYN = IGA_AV_H_RNG_HOMS[x]
            TD_TMP_SYN = TD_S
            (4)IF INJ_MOD[x] = "HOMS_SNG"
            (4)THEN
                SOI_TMP_SYN = SOI_1_HOMS[x]
            (4)ELSE
                SOI_TMP_SYN = EOI_2_HOMS[x] + TI_2_S[x] * FAC_N
            (4)ENDIF
        (3)ELSE – homogeneous combustion mode
            IGA_TMP_SYN = IGA_AV_H_RNG[x]
            TD_TMP_SYN = TD
            (5)IF INJ_MOD[x] = "SNGH"
            (5)THEN
                (5a)IF LC_IV_ADD_PLS_SNGH_ENA = 1
                (5a)THEN
                    SOI_TMP_SYN = MAX(C_SOI_MIN_UPD_PLS_SNGH; EOI_LIM_HOM)
                (5a)ELSE
                    SOI_TMP_SYN = SOI_1_HOM[x]
                (5a)ENDIF
            (5)ELSE
                SOI_TMP_SYN = EOI_2_HOM[x] + TI_2_HOM[x] * FAC_N
            (5)ENDIF
        (3)ENDIF
```

```
        (2)ENDIF
(1)ELSE – calculate temporary values for mode switch
        (6)IF INJ_MOD_SP[x] BitwiseAND NC_INJ_MOD_MASK_1 = NC_INJ_MOD_S
        (6)THEN – stratified combustion mode
                IGA_TMP_SYN = IGA_AV_H_RNG_S[x]
                TD_TMP_SYN = TD_S
                SOI_TMP_SYN = EOI_1_S[x] + TI_1_S[x] * FAC_N
        (6)ELSE
                (7)IF INJ_MOD_SP[x] BitwiseAND NC_INJ_MOD_MASK_1 =
NC_INJ_MOD_HOMS
                (7)THEN – homogeneous-stratified combustion mode
                        IGA_TMP_SYN = IGA_AV_H_RNG_HOMS[x]
                        TD_TMP_SYN = TD_S
                        SOI_TMP_SYN = SOI_1_HOMS[x]
                (7)ELSE – homogeneous combustion mode
                        IGA_TMP_SYN = IGA_AV_H_RNG[x]
                        TD_TMP_SYN = TD
                        SOI_TMP_SYN = SOI_1_HOM[x]
                (7)ENDIF
        (6)ENDIF
(1)ENDIF
```

41.5 Transfer to basic software (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_V_IV_MES_PBK[NC_PBK_IV_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Factor to correct the measured injector voltage due to end of line trimming					
FAC_CHA_IV_MES_PBK[NC_PBK_IV_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Factor to correct the measured injector charge due to end of line trimming					

Input data:

C_FAC_V_IV_MES_PBK_0_0	C_FAC_V_IV_MES_PBK_0_1	C_FAC_V_IV_MES_PBK_0_CPL	C_FAC_V_IV_MES_PBK_1_0
C_FAC_V_IV_MES_PBK_1_1	C_FAC_V_IV_MES_PBK_1_CPL	C_FAC_CHA_IV_MES_PBK_0_0	C_FAC_CHA_IV_MES_PBK_0_1
C_FAC_CHA_IV_MES_PBK_0_CPL	C_FAC_CHA_IV_MES_PBK_1_0	C_FAC_CHA_IV_MES_PBK_1_1	C_FAC_CHA_IV_MES_PBK_1_CPL
NC_PBK_IV_NR			

FUNCTION DESCRIPTION:

This function reads the ECU end of line trimming values for piezo injector energy measurement. A check against the complementary values is done as well as a plausibility check.

General information:

The pizo injector energy measurement consists of a voltage measurement and a charge measurement. Both kinds of measurement have to be calibrated at the end of production line (EOL trimming) to improve measurement accuracy. The results of the EOL trimming are one correction factor for voltage measurement and one correciton factor for charge measurement for each injection driver ATIC.

Signal flow diagram:

-

Description:

If the functionality is activated by LC, each values are checked against their complementary values which are also stored within the ECU at end of produciton line. After that, a plausibility check is done by using an upper and a lower threshold. If the complementary check or the plausibility check fails or if the functionality is deactivated by LC, the trimming factors are set to 1.

Application conditions:

Initialisation: -

Recurrence: only once at reset

Activation: every engine state

Deactivation: -

Formula section:

```
(1) IF LC_FAC_IV_MES_PBK_ENA = 1
(1) THEN

        Injector voltage correction of first power bank:
        Check the ECU end of line trimming values against their binary complement and select a
        valid value or set the correction factor to one in case of invalid trimming values.
        (2) IF (C_FAC_V_IV_MES_PBK_0_0  BitwiseXOR  C_FAC_V_IV_MES_PBK_0_CPL
                                                                   = FFFFH)
        (2) THEN
                FAC_V_IV_MES_PBK[0] = C_FAC_V_IV_MES_PBK_0_0
        (2) ELSE
                (3) IF (C_FAC_V_IV_MES_PBK_0_1  BitwiseXOR
                                                C_FAC_V_IV_MES_PBK_0_CPL = FFFFH)
                (3) THEN
                        FAC_V_IV_MES_PBK[0] = C_FAC_V_IV_MES_PBK_0_1
                (3) ELSE
                        FAC_V_IV_MES_PBK[0] = 1
                (3) ENDIF
        (2) ENDIF

        Check the calculated correction factor for plausibility
        (4) IF (FAC_V_IV_MES_PBK[0] > C_FAC_V_IV_MES_PBK_MAX_PLAUS  OR
                (FAC_V_IV_MES_PBK[0] < C_FAC_V_IV_MES_PBK_MIN_PLAUS)
        (4) THEN
                FAC_V_IV_MES_PBK[0] = 1
        (4) ENDIF

        Injector charge correction of first power bank:
        Check the ECU end of line trimming values against their binary complement and select a
        valid value or set the correction factor to one in case of invalid trimming values.
        (5) IF (C_FAC_CHA_IV_MES_PBK_0_0  BitwiseXOR
                                          C_FAC_CHA_IV_MES_PBK_0_CPL = FFFFH)
        (5) THEN
                FAC_CHA_IV_MES_PBK[0] = C_FAC_CHA_IV_MES_PBK_0_0
        (5) ELSE
                (6) IF (C_FAC_CHA_IV_MES_PBK_0_1  BitwiseXOR
                                          C_FAC_CHA_IV_MES_PBK_0_CPL = FFFFH)
                (6) THEN
                        FAC_CHA_IV_MES_PBK[0] = C_FAC_CHA_IV_MES_PBK_0_1
                (6) ELSE
                        FAC_CHA_IV_MES_PBK[0] = 1
                (6) ENDIF
        (5) ENDIF

        Check the calculated correction factor for plausibility
        (7) IF (FAC_CHA_IV_MES_PBK[0] > C_FAC_CHA_IV_MES_PBK_MAX_PLAUS  OR
```

```
        (FAC_CHA_IV_MES_PBK[0] < C_FAC_CHA_IV_MES_PBK_MIN_PLAUS)
(7) THEN
        FAC_CHA_IV_MES_PBK[0] = 1
(7) ENDIF
```

Injector voltage correction of second power bank:

Check the ECU end of line trimming values against their binary complement and select a valid value or set the correction factor to one in case of invalid trimming values.

```
(8) IF (C_FAC_V_IV_MES_PBK_1_0  BitwiseXOR  C_FAC_V_IV_MES_PBK_1_CPL
                                                        = FFFFH)
(8) THEN
        FAC_V_IV_MES_PBK[1] = C_FAC_V_IV_MES_PBK_1_0
(8) ELSE
        (9) IF (C_FAC_V_IV_MES_PBK_1_1  BitwiseXOR
                                    C_FAC_V_IV_MES_PBK_1_CPL = FFFFH)
        (9) THEN
                FAC_V_IV_MES_PBK[1] = C_FAC_V_IV_MES_PBK_1_1
        (9) ELSE
                FAC_V_IV_MES_PBK[1] = 1
        (9) ENDIF
(8) ENDIF
```

Check the calculated correction factor for plausibility

```
(10) IF (FAC_V_IV_MES_PBK[1] > C_FAC_V_IV_MES_PBK_MAX_PLAUS  OR
        (FAC_V_IV_MES_PBK[1] < C_FAC_V_IV_MES_PBK_MIN_PLAUS)
(10) THEN
        FAC_V_IV_MES_PBK[1] = 1
(10) ENDIF
```

Injector charge correction of second power bank:

Check the ECU end of line trimming values against their binary complement and select a valid value or set the correction factor to one in case of invalid trimming values.

```
(11) IF (C_FAC_CHA_IV_MES_PBK_1_0  BitwiseXOR
                                C_FAC_CHA_IV_MES_PBK_1_CPL = FFFFH)
(11) THEN
        FAC_CHA_IV_MES_PBK[1] = C_FAC_CHA_IV_MES_PBK_1_0
(11) ELSE
        (12) IF (C_FAC_CHA_IV_MES_PBK_1_1  BitwiseXOR
                                C_FAC_CHA_IV_MES_PBK_1_CPL = FFFFH)
        (12) THEN
                FAC_CHA_IV_MES_PBK[1] = C_FAC_CHA_IV_MES_PBK_1_1
        (12) ELSE
                FAC_CHA_IV_MES_PBK[1] = 1
        (12) ENDIF
(11) ENDIF
```

Check the calculated correction factor for plausibility

```
(13) IF (FAC_CHA_IV_MES_PBK[1] > C_FAC_CHA_IV_MES_PBK_MAX_PLAUS  OR
        (FAC_CHA_IV_MES_PBK[1] < C_FAC_CHA_IV_MES_PBK_MIN_PLAUS)
(13) THEN
        FAC_CHA_IV_MES_PBK[1] = 1
```

```
        (13) ENDIF

(1) ELSE
        (1) FOR  i = 0  TO  NC_PBK_IV_NR – 1
                FAC_V_IV_MES_PBK[i] = 1
                FAC_CHA_IV_MES_PBK[i] = 1
        (1) ENDFOR
(1) ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_FAC_IV_MES_PBK_ENA	1	0...1H	0...1	1	[-]
Switch to enable the use of ECU end of line trimming values for injector voltage and charge measurement					
C_FAC_V_IV_MES_PBK_MAX_PLAUS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum value for injector voltage correction plausibility check due to end of line trimming					
C_FAC_V_IV_MES_PBK_MIN_PLAUS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Minimum value for injector voltage correction plausibility check due to end of line trimming					
C_FAC_CHA_IV_MES_PBK_MAX_PLAUS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Maximum value for injector charge correction plausibility check due to end of line trimming					
C_FAC_CHA_IV_MES_PBK_MIN_PLAUS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Minimum value for injector charge correction plausibility check due to end of line trimming					

41.6 Injection phase for homogeneous mode, 1. pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EOI_1_HOM	V/O	0...780H	0...720	0.375	[°CRK]
End of injection for homogeneous mode, 1.pulse - non cylinder individual HPDI-Version					
SOI_1_HOM[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
Start of injection for homogeneous mode, 1.pulse - cylinder individual HPDI-Version					
EOI_LIM_HOM	V/O	0...780H	0...720	0.375	[°CRK]
Latest possible EOI at homogeneouse mode					
SOI_LIM	V/O	0...780H	0...720	0.375	[°CRK]
Earliest possible SOI					
LV_CRK_DIF_SOI_EOI_LIM	V/O	0...1H	0...1	1	[-]
Difference between SOI and EOI is beyond the maximum, the pulse will be limited					
CRK_DIF_SOI_IGN_HOM	V/O	0...F00H	0...1440	0.375	[°CRK]
Current start of injection phase related to ignition TDC					
CRK_INJ_BAS[NC_CYL_NR]	V/O	F100...F00H	-1440...1440	0.375	[°CRK]
Cylinder individual zero position for phasing related to ignition TDC					
LV_EOI_LIM_HOM	V	0...1H	0...1	1	[-]
At least one injection was repositioned to EOI_LIM_HOM					
SOI_1_HOM_CLC[NC_CYL_NR]	V	0...780H	0...720	0.375	[°CRK]
Calculated start of injection for homogeneous mode, 1.pulse - cylinder individual HPDI-Version					
SOI_1_HOM_MAX	-	0...780H	0...720	0.375	[°CRK]
Maximum start of injection angle for homogeneous mode, 1.pulse					

Input data:

CAM_SHIFT_EX	CTR_CYL_NR_ST_CLC_INJR	EOI_2_HOM_EXT[NC_CYL_NR]	FAC_N
FUP_H_INJ[NC_CBK_HPP_NR]	INJ_MOD_HOM_REQ	LV_EOI_2_DELTA_HOM_CUS	LV_EOI_ST_ENA
LV_ST_END	NC_CBK_HPP_NR	NC_CRK_INJ_BAS_REF	NC_CRK_INJ_REF_TDC
NC_CRK_INJ_RNG	NC_CYL_NR	NC_INJ_MOD_SINGLE	NC_T_MIN_OFS_INJ_CYL
NR_CYL_CLC_RED_INJ	SOI_1_HOM_EXT[NC_CYL_NR]	T_DCHA_DLY_HOM_1[NC_CYL_NR]	T_DCHA_PER_HOM_1_1[NC_CYL_NR]
TI_1_HOM[NC_CYL_NR]	TI_1_HOM_CLC_MAX		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_EOI_LIM_HOM_FUP_COR	8	0...780H	-360...360	0.375	[°CRK]
LDPM_FUP_H_1_INJR	8	0...FFFFH	0...347776	5.3067216	[hPa]
FUP correction of EOI_LIM_HOM					
C_EOI_LIM_HOM	1	0...780H	0...720	0.375	[°CRK]
Latest possible EOI at homogeneous mode					
C_EOI_LIM_HOM_ST	1	0...780H	0...720	0.375	[°CRK]
Latest possible EOI at homogeneous mode at start					
C_EOI_LIM_HOM_MULTI_COR	1	FC40...3C0H	-360...360	0.375	[°CRK]
Correction value for calculation of the latest possible EOI, if homogeneous multi injection mode is active					
C_SOI_LIM	1	0...780H	0...720	0.375	[°CRK]
Earliest possible SOI					
LC_SOI_1_HOM_MAN	1	0...1H	0...1	1	[-]

Switch to activate the manual setpoint for homogeneous SOI					
C_SOI_1_HOM_MAN	1	0...780H	0...720	0.375	[°CRK]
Manual setpoint for homogeneous SOI					
LC_SOI_1_HOM_ST_MAN	1	0...1H	0...1	1	[-]
Switch to activate the manual setpoint for homogeneous SOI during start					
C_SOI_1_HOM_ST_MAN	1	0...780H	0...720	0.375	[°CRK]
Manual setpoint for homogeneous SOI during start					
C_EOI_LIM_HOM_CH_COR	1	FC40...3C0H	-360...360	0.375	[°CRK]
Additive correction on EOI_LIM_HOM duing catalyst heating					
LC_EOI_DLY_AS_ENA	1	0...1H	0...1	1	[-]
Switch to enable EOI calibration of by second discharge phase.					

FUNCTION DESCRIPTION:

41.6.1 Definition of the injection range

Application conditions:

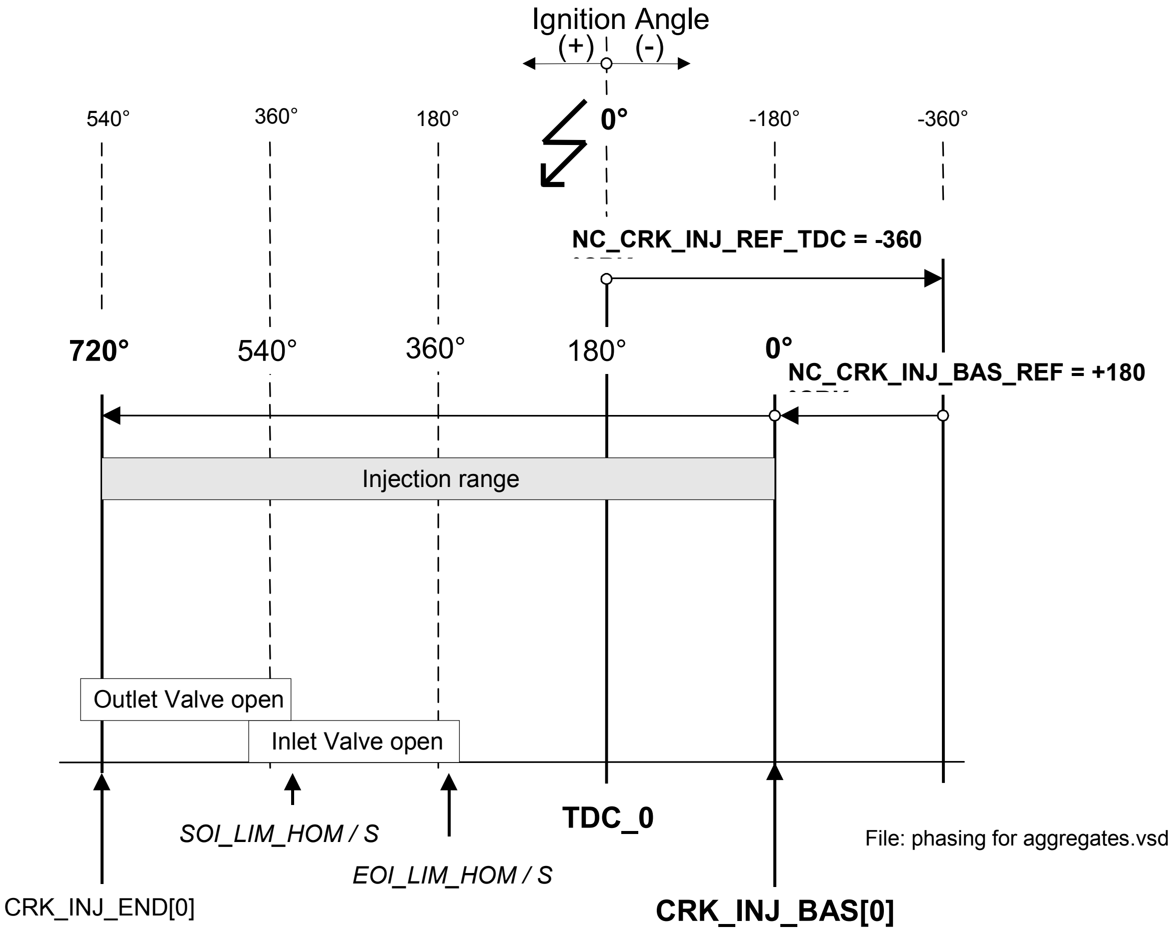
Activation:	at reset
Deactivation:	-
Initialization:	-
Recurrence:	once at reset

The fuel can be injected over a range of 720 °CRK. The cylinder individual zero position of this range is defined via the constants NC_CRK_INJ_REF_TDC and NC_CRK_INJ_BAS_REF with reference to the ignition TDC of cylinder 0, see figure below.

Formula section:

Related to tdc cylinder 0:

```
FOR x = 0 TO (NC_CYL_NR-1) DO:
    CRK_INJ_BAS[x] = NC_CRK_INJ_REF_TDC + NC_CRK_INJ_BAS_REF - x * (720 / NC_CYL_NR)
ENDFOR
```


Ignition Angle
(+)
(-)
540°
360°
180°
0°
-180°
-360°
NC_CRK_INJ_REF_TDC = -360
720°
540°
360°
180°
0°
NC_CRK_INJ_BAS_REF = +180
Injection range
Outlet Valve open
Inlet Valve open
SOI_LIM_HOM / S
EOI_LIM_HOM / S
TDC_0
File: phasing for aggregates.vsd
CRK_INJ_END[0]
CRK_INJ_BAS[0]

Definition of the non calibrateable constants:

NC_CRK_INJ_REF_TDC = -360°CRK

NC_CRK_INJ_BAS = 180 °CRK

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
            x = CTR_CYL_NR_ST_CLC_INJR + i
            (1a)IF x  >= NC_CYL_NR
            (1a)THEN
                    x = x – NC_CYL_NR
            (1a)ENDIF
            <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
     <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

41.6.2 Injection phasing at start

Application conditions:

Activation: LV_ST_END = 0

Deactivation: -

Initialization: -

Recurrence: 10ms

Formula section:

All calculations are related to CRK_INJ_BAS[NC_CYL_NR].

```
EOI_LIM_HOM = 0
(1) IF (single injection)
INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE (if single injection is requested)
(1) THEN
    FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
            EOI_LIM_HOM = MAX(EOI_LIM_HOM;
                   C_EOI_LIM_HOM_ST + IP_EOI_LIM_HOM_FUP_COR (Input: FUP_H_INJ[m]) )
    ENDFOR
(1) ELSE (multiple injection)
    For double injection the minimum injection angle will be limited by the following calculation:
    FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
            EOI_LIM_HOM = MAX(EOI_LIM_HOM;
                   C_EOI_LIM_HOM_ST + IP_EOI_LIM_HOM_FUP_COR (Input: FUP_H_INJ[m])
                   + C_EOI_LIM_HOM_MULTI_COR )
    ENDFOR
(1) ENDIF
```

```
SOI_LIM = C_SOI_LIM + CAM_SHIFT_EX
```

Check if the difference between SOI and EOI is beyond the maximum.

```
(2) IF
    (TI_1_HOM_CLC_MAX * FAC_N + EOI_LIM_HOM) > SOI_LIM
(2) THEN
    LV_CRK_DIF_SOI_EOI_LIM = 1          (TI is beyond the maximum)
(2) ELSE
    LV_CRK_DIF_SOI_EOI_LIM = 0          (TI is below the maximum)
(2) ENDIF
```

Calculation of SOI:

```
(3) IF
LC_SOI_1_HOM_ST_MAN = 1
(3) THEN
   (1) FOR x = 0 TO (NC_CYL_NR -1) DO:
      SOI_1_HOM_CLC[x] = C_SOI_1_HOM_ST_MAN
   (1) ENDFOR
(3) ELSE
   (4) IF LV_EOI_ST_ENA = 1
   (4) THEN
     (4a)IF LC_EOI_DLY_AS_ENA = 1
     (4a)THEN
        (2a) FOR x = 0 TO (NC_CYL_NR -1) DO:
             SOI_1_HOM_CLC[x] =
             (TI_1_HOM[x] + T_DCHA_PER_HOM_1_1[x] + T_DCHA_DLY_HOM_1[x])
                                   * FAC_N + EOI_2_HOM_EXT[x]
        (2a) ENDFOR
     (4a)ELSE
        (2) FOR x = 0 TO (NC_CYL_NR -1) DO:
             SOI_1_HOM_CLC[x] = TI_1_HOM[x] * FAC_N + EOI_2_HOM_EXT[x]
        (2) ENDFOR
     (4a)ENDIF
   (4) ELSE
     (3) FOR x = 0 TO (NC_CYL_NR -1) DO:
        SOI_1_HOM_CLC[x] = SOI_1_HOM_EXT[x]
     (3) ENDFOR
   (4) ENDIF
(3) ENDIF
```

Limitation to the end of injection:

```
SOI_1_HOM_MAX = 0
(4) FOR x = 0 TO (NC_CYL_NR -1) DO:
   (5) IF
   SOI_1_HOM_CLC[x] - TI_1_HOM[x] * FAC_N < EOI_LIM_HOM
   (5) THEN
   SOI_1_HOM[x] = EOI_LIM_HOM + TI_1_HOM[x] * FAC_N
```

```
        LV_ EOI_LIM_HOM = 1        At least one injection was repositioned to EOI_LIM_HOM
        (5) ELSE
        SOI_1_HOM[x] = SOI_1_HOM_CLC[x]
        LV_ EOI_LIM_HOM = 0
        (5) ENDIF
        Limit SOI_1_HOM to SOI_LIM:
        SOI_1_HOM[x] = MIN(SOI_1_HOM[x]; SOI_LIM)
        SOI_1_HOM_MAX = MAX(SOI_1_HOM_MAX, SOI_1_HOM[x])
    (4) ENDFOR

    (6) IF (single injection)
    INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE (if single injection is requested)
    (6) THEN
    EOI_LIM_HOM = MAX(EOI_LIM_HOM, (SOI_1_HOM_MAX
                             - (NC_CRK_INJ_RNG - NC_T_MIN_OFS_INJ_CYL *FAC_N)))
    (6) ENDIF

    EOI_1_HOM = 720°CRK
    (5) FOR x = 0 TO (NC_CYL_NR -1) DO:
        EOI_1_HOM = MIN(EOI_1_HOM, (SOI_1_HOM[x] - TI_1_HOM[x] * FAC_N))
    (5) ENDFOR
```

Output variable for LACO:

CRK_DIF_SOI_IGN_HOM = SOI_1_HOM[0] + CRK_INJ_BAS [0]

41.6.3 Injection phasing out of start

Application conditions:

Activation:	LV_ST_END = 1
Deactivation:	-
Initialization:	-
Recurrence:	segment synchronous

Formula section:

Out of start injection phasing is defined via SOI.

All calculations are related to CRK_INJ_BAS[NC_CYL_NR].

```
EOI_LIM_HOM = 0
(1) IF (single injection)
INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE (if single injection is requested)
(1) THEN
      FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
            EOI_LIM_HOM = MAX(EOI_LIM_HOM;
                  C_EOI_LIM_HOM + IP_EOI_LIM_HOM_FUP_COR (Input: FUP_H_INJ[m]) )
      ENDFOR
(1) ELSE (multiple injection)
      (1a) IF LV_EOI_2_DELTA_HOM_CUS = 1
      (1a)THEN
            FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
                  Minimum injection angle during catalyst heating:
                  EOI_LIM_HOM = MAX(EOI_LIM_HOM;
                        C_EOI_LIM_HOM
                        + IP_EOI_LIM_HOM_FUP_COR (Input: FUP_H)
                        + C_EOI_LIM_HOM_CH_COR)
            ENDFOR
      (1a) ELSE
            For double injection the minimum injection angle will be limited by the following calculation
            FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
                  EOI_LIM_HOM = MAX(EOI_LIM_HOM;
                        C_EOI_LIM_HOM + IP_EOI_LIM_HOM_FUP_COR (Input:
                        FUP_H_INJ[m])
                        + C_EOI_LIM_HOM_MULTI_COR )
```

```
                    ENDFOR
            (1a) ENDIF

    (1) ENDIF

    SOI_LIM = C_SOI_LIM + CAM_SHIFT_EX
    Check it the difference between SOI and EOI is beyond the maximum.
    (2) IF
            (TI_1_HOM_CLC_MAX * FAC_N + EOI_LIM_HOM) > SOI_LIM
    (2) THEN
            LV_CRK_DIF_SOI_EOI_LIM = 1              (TI is beyond the maximum)
    (2) ELSE
            LV_CRK_DIF_SOI_EOI_LIM = 0              (TI is below the maximum)
    (2) ENDIF


    Calculation of SOI:
    (3) IF
    LC_SOI_1_HOM_MAN = 1
    (3) THEN
        (1)FOR cylinder_start to cylinder_stop DO
            SOI_1_HOM_CLC[x] = C_SOI_1_HOM_MAN
        (1) ENDFOR
    (3) ELSE
        (2)FOR cylinder_start to cylinder_stop DO
                Dynamical limitation of SOI under consideration of  NC_CRK_INJ_RNG:
                (4) IF SOI_1_HOM_EXT[x] < (EOI_1_HOM + (NC_CRK_INJ_RNG -
                                                  NC_T_MIN_OFS_INJ_CYL *FAC_N))
                (4) THEN
                        SOI_1_HOM_CLC[x] = SOI_1_HOM_EXT[x]
                (4) ELSE
                        SOI_1_HOM_CLC[x] = EOI_1_HOM + (NC_CRK_INJ_RNG -
                                                  NC_T_MIN_OFS_INJ_CYL *FAC_N)
                (4) ENDIF
        (2) ENDFOR
     (3) ENDIF


    Limitation to the end of injection:
    SOI_1_HOM_MAX = 0
```

```
(3)FOR cylinder_start to cylinder_stop DO
    (5) IF
    SOI_1_HOM_CLC[x] - TI_1_HOM[x] * FAC_N < EOI_LIM_HOM
    (5) THEN
    SOI_1_HOM[x] = EOI_LIM_HOM + TI_1_HOM[x] * FAC_N
    LV_ EOI_LIM_HOM = 1          At least one injection was repositioned to EOI_LIM_HOM
    (5) ELSE
    SOI_1_HOM[x] = SOI_1_HOM_CLC[x]
    LV_ EOI_LIM_HOM = 0
    (5) ENDIF
    Limit SOI_1_HOM to SOI_LIM:
    SOI_1_HOM[x] = MIN(SOI_1_HOM[x]; SOI_LIM)
    SOI_1_HOM_MAX = MAX(SOI_1_HOM_MAX, SOI_1_HOM[x])
(3) ENDFOR
(6) IF (single injection)
INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE (if single injection is requested)
(6) THEN
EOI_LIM_HOM = MAX(EOI_LIM_HOM, (SOI_1_HOM_MAX
                              - (NC_CRK_INJ_RNG - NC_T_MIN_OFS_INJ_CYL *FAC_N)))
(6) ENDIF

EOI_1_HOM = 720°CRK
(4)FOR cylinder_start to cylinder_stop DO
     EOI_1_HOM = MIN(EOI_1_HOM, (SOI_1_HOM[x] - TI_1_HOM[x] * FAC_N))
(4) ENDFOR
```

Output variable for LACO (10 ms synchronous):

CRK_DIF_SOI_IGN_HOM = SOI_1_HOM[0] + CRK_INJ_BAS[0]

41.7 Injection phase for multiple homogeneous mode, 2. pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EOI_2_HOM[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of injection of the second injection for hom. combustion mode (intermediate value)					
SOI_2_HOM	V	0...780H	0...720	0.375	[°CRK]
Start of injection of the second injection for homogeneous combustion					
EOI_1_SOI_2_GAP_HOM	V	0...780H	0...720	0.375	[°CRK]
Gap between the two injections at cylinder 0					
SOI_2_HOM_INTER[NC_CYL_NR]	-	0...780H	0...720	0.375	[°CRK]
Calibrated start of injection of the second injection for homogeneous combustion mode					
EOI_2_HOM_INTER[NC_CYL_NR]	-	0...780H	0...720	0.375	[°CRK]
end of injection of the second injection for hom. combustion mode (intermediate value)					
SOI_2_HOM_TOL	-	0...780H	0...720	0.375	[°CRK]
Top limit for the start of the second injection					

Input data:

CTR_CYL_NR_ST_CLC_INJR	EOI_1_HOM	EOI_2_HOM_EXT[NC_CYL_NR]	FAC_N
INJ_MOD_SP_HOM[NC_CYL_NR]	LC_EOI_DLY_AS_ENA	LV_ST_END	NC_CYL_NR
NR_CYL_CLC_RED_INJ	SOI_2_MAX_HOM_EXT	T_DCHA_DLY_HOM_1[NC_CYL_NR]	T_DCHA_DLY_HOM_2[NC_CYL_NR]
T_DCHA_PER_HOM_2_1[NC_CYL_NR]	TI_2_HOM[NC_CYL_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DELTA_T_MIN_DUI	1	0...FFFFH	0...65.535	0.001	[ms]
Least time between two injections					
C_EOI_2_HOM_MAN	1	0...780H	0...720	0.375	[°CRK]
Manual setpoint for EOI					
LC_EOI_2_HOM_MAN	1	0...1H	0...1	1	[-]
switch for using the manual setpoit for EOI					

FUNCTION DESCRIPTION:

For the definition of the injection range see module “Injection phase for homogeneous mode, 1. pulse”.

The injection phase for the second pulse is defined via EOI.

All calculations are related to CRK_INJ_BAS[NC_CYL_NR].

Application conditions:

Activation:	every engine state	
Deactivation:	-	
Initialization:	-	
Recurrence:	LV_ST_END = 0:	10 ms
	LV_ST_END = 1:	segment synchronous

Formula section:

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
              x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
   <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

Start of the 2nd injection has to be earliest at SOI_2_MAX_HOM_EXT and after the end of the 1st pulse plus a defined delay time C_DELTA_T_MIN_DUI.

```
SOI_2_HOM_TOL = MIN(EOI_1_HOM – (C_DELTA_T_MIN_DUI
                                + T_DCHA_DLY_HOM_1[CTR_CYL_NR_ST_CLC_INJR])
                                * FAC_N; SOI_2_MAX_HOM_EXT)
SOI_2_HOM = 0
```

```
FOR cylinder_start to cylinder_stop DO
(1)IF INJ_MOD_SP_HOM[x] ≠ "SNGH"
(1)THEN
     (2) IF LC_EOI_2_HOM_MAN = 1
     (2) THEN
          EOI_2_HOM_INTER[x] = C_EOI_2_HOM_MAN
     (2) ELSE
          EOI_2_HOM_INTER[x] = EOI_2_HOM_EXT[x]
     (2) ENDIF
     (2a)IF LC_EOI_DLY_AS_ENA = 1
     (2a)THEN
          EOI_2_HOM_INTER[x] = EOI_2_HOM_INTER[x]
               + (T_DCHA_PER_HOM_2_1[x] + T_DCHA_DLY_HOM_2[x]) * FAC_N
     (2a)ENDIF
     SOI_2_HOM_INTER[x]  =    EOI_2_HOM_INTER[x] + TI_2_HOM[x] * FAC_N
     (3) IF SOI_2_HOM_INTER[x] ≥ SOI_2_HOM_TOL
     (3) THEN
          EOI_2_HOM[x] = EOI_2_HOM_INTER[x] - (SOI_2_HOM_INTER[x] -SOI_2_HOM_TOL)
          SOI_2_HOM = MAX(SOI_2_HOM; SOI_2_HOM_TOL)
     (3) ELSE
          EOI_2_HOM[x] = EOI_2_HOM_INTER[x]
          SOI_2_HOM = MAX(SOI_2_HOM; SOI_2_HOM_INTER[x])
     (3) ENDIF

(1)  ELSE
       EOI_2_HOM[x] = 0
(1)  ENDIF
ENDFOR
```

Gap between the two injections:

```
(4)IF SOI_2_HOM ≠ 0
(4)THEN
     EOI_1_SOI_2_GAP_HOM = EOI_1_HOM – SOI_2_HOM
(4)ELSE
     EOI_1_SOI_2_GAP_HOM = 0
```

(4)ENDIF

41.8 Injection phase for multiple homogeneous mode, 3. pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EOI_3_HOM[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
End of injection of the third injection for hom. combustion mode					
SOI_3_HOM_MAX	V	0...780H	0...720	0.375	[°CRK]
Maximum start of injection angle of the third injection for homogeneous combustion mode					
EOI_2_SOI_3_GAP_HOM	V	0...780H	0...720	0.375	[°CRK]
Gap between the second and the third injection at cylinder 0					
SOI_3_HOM_INTER[NC_CYL_NR]	-	0...780H	0...720	0.375	[°CRK]
Calibrated start of injection of the third injection for homogeneous combustion mode					
EOI_3_HOM_INTER[NC_CYL_NR]	-	0...780H	0...720	0.375	[°CRK]
End of injection of the third injection for hom. combustion mode (intermediate value)					
SOI_3_HOM_TOL	-	0...780H	0...720	0.375	[°CRK]
Top limit for the start of the third injection					

Input data:

C_DELTA_T_MIN_DUI	CTR_CYL_NR_ST_CLC_INJR	EOI_2_HOM[NC_CYL_NR]	EOI_3_HOM_EXT[NC_CYL_NR]
FAC_N	IDX_PRS_COR_CYL_CLC_MPLH[NC_CBK_HPP_NR]	INJ_MOD_SP_HOM[NC_CYL_NR]	LC_EOI_DLY_AS_ENA
LV_ST_END	NC_CBK_HPP_NR	NC_CYL_NR	NR_CYL_CLC_RED_INJ
T_DCHA_DLY_HOM_2[NC_CYL_NR]	T_DCHA_DLY_HOM_3[NC_CYL_NR]	T_DCHA_PER_HOM_3_1[NC_CYL_NR]	TI_3_HOM[NC_CYL_NR]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EOI_3_HOM_MAN	1	0...780H	0...720	0.375	[°CRK]
Manual setpoint for EOI of third pulse, homogeneous combustion mode					
LC_EOI_3_HOM_MAN	1	0...1H	0...1	1	[-]
Switch for enabling the manual setpoint for EOI of third pulse, homogeneous combustion mode					

FUNCTION DESCRIPTION:

For the definition of the injection range see module “Injection phase for homogeneous mode, 1. pulse”.

The injection phase for the second pulse is defined via EOI.

All calculations are related to CRK_INJ_BAS[NC_CYL_NR].

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
             x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
    <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

Application conditions:

Activation: every engine state

Deactivation: -

Initialization: -

Recurrence:
LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

Formula section:

```
SOI_3_HOM_MAX = 0°CRK
FOR cylinder_start to cylinder_stop DO
  (1)IF INJ_MOD_SP_HOM[x] = "MPLH+PLS3"
  (1)THEN
    SOI_3_HOM_TOL = EOI_2_HOM[x] – (C_DELTA_T_MIN_DUI
                                        + T_DCHA_DLY_HOM_2[x])* FAC_N
    (2) IF LC_EOI_3_HOM_MAN = 1
    (2) THEN
       EOI_3_HOM_INTER[x] = C_EOI_3_HOM_MAN
```

```
        (2) ELSE
            EOI_3_HOM_INTER[x] = EOI_3_HOM_EXT[x]
        (2) ENDIF
        (2a)IF LC_EOI_DLY_AS_ENA = 1
        (2a)THEN
                EOI_3_HOM_INTER[x] = EOI_3_HOM_INTER[x]
                        + (T_DCHA_PER_HOM_3_1[x] + T_DCHA_DLY_HOM_3[x]) * FAC_N
        (2a)ENDIF
        SOI_3_HOM_INTER[x] = EOI_3_HOM_INTER[x] + TI_3_HOM[x] * FAC_N
        (3) IF SOI_3_HOM_INTER[x] > SOI_3_HOM_TOL
        (3) THEN
            EOI_3_HOM[x] = EOI_3_HOM_INTER[x] - (SOI_3_HOM_INTER[x] - SOI_3_HOM_TOL)
            SOI_3_HOM_MAX = MAX(SOI_3_HOM_MAX, SOI_3_HOM_TOL)
        (3) ELSE
            EOI_3_HOM[x] = EOI_3_HOM_INTER[x]
            SOI_3_HOM_MAX = MAX(SOI_3_HOM_MAX, SOI_3_HOM_INTER[x])
        (3) ENDIF
      (1)ELSE
        EOI_3_HOM[x] = 0
      (1)ENDIF
    ENDFOR
```

Gap between the second and the third injection:

```
    (4)IF SOI_3_HOM_MAX ≠ 0
    (4)THEN
        FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
            (5)IF INJ_MOD_SP_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] =
                                                            "MPLH+PLS3"
            (5)THEN
                EOI_2_SOI_3_GAP_HOM = EOI_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                        - (EOI_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]
                           + TI_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] *FAC_N)
                Break the for-loop after condition is fulfilled one time
            (5)ENDIF
        ENDFOR
    (4)ELSE
        EOI_2_SOI_3_GAP_HOM = 0
    (4)ENDIF
```

41.9 Injection phase for homogeneous-stratified mode

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SOI_1_HOMS[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
Start of the first injection for homogeneous stratified mode					
EOI_2_HOMS[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of the second injection for homogeneous stratified mode					
EOI_3_HOMS[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of the third injection for homogeneous stratified mode					

Input data:

CTR_CYL_NR_ST_CLC_INJR	EOI_1_HOMS_EXT[NC_CYL_NR]	EOI_2_HOMS_EXT[NC_CYL_NR]	EOI_3_HOMS_EXT[NC_CYL_NR]
FAC_N	INJ_MOD_SP_HOMS[NC_CYL_NR]	LC_EOI_DLY_AS_ENA	LV_S_CLC
LV_ST_END	NC_CYL_NR	NC_T_MIN_OFS_INJ_PLS	NR_CYL_CLC_RED_INJ
SOI_1_HOMS_EXT[NC_CYL_NR]	T_DCHA_DLY_S_1[NC_CYL_NR]	T_DCHA_DLY_S_2[NC_CYL_NR]	T_DCHA_DLY_S_3[NC_CYL_NR]
T_DCHA_PER_S_1_1[NC_CYL_NR]	T_DCHA_PER_S_2_1[NC_CYL_NR]	T_DCHA_PER_S_3_1[NC_CYL_NR]	T_DLY_2_3_MIN_HOMS[NC_CYL_NR]
TI_1_S[NC_CYL_NR]	TI_2_S[NC_CYL_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_SOI_1_HOMS_MAN	1	0...1H	0...1	1	[-]
Switch to activate the manual setpoint for SOI_1, homogeneous stratified mode					
LC_EOI_2_HOMS_MAN	1	0...1H	0...1	1	[-]
Switch to activate the manual setpoint for EOI_2, homogeneous stratified mode					
LC_EOI_3_HOMS_MAN	1	0...1H	0...1	1	[-]
Switch to activate the manual setpoint for EOI_3, homogeneous stratified mode					
C_SOI_1_HOMS_MAN	1	0...780H	0...720	0.375	[°CRK]
Manual setpoint for SOI of the first pulse during homogeneous stratified mode					
C_EOI_2_HOMS_MAN	1	0...780H	0...720	0.375	[°CRK]
Manual setpoint for EOI of the second pulse during homogeneous stratified mode					
C_EOI_3_HOMS_MAN	1	0...780H	0...720	0.375	[°CRK]
Manual setpoint for EOI of the third pulse during homogeneous stratified mode					

FUNCTION DESCRIPTION:

Calculation of injection phasing for homogeneous-stratified mode.

Application conditions:

Activation:	LV_S_CLC = 1	
Deactivation:	LV_S_CLC = 0	
Initialization at Reset:	-	
Recurrence:	LV_ST_END = 0:	10 ms
	LV_ST_END = 1:	segment synchronous

Formula section:

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
                x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
    <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

```
(0)FOR cylinder_start to cylinder_stop DO
```

Single injection

```
(0)IF INJ_MOD_SP_HOMS[x] = "HOMS-SNG"
(0)THEN
    (1) IF LC_SOI_1_HOMS_MAN = 1
    (1) THEN
            SOI_1_HOMS[x] = C_SOI_1_HOMS_MAN
    (1) ELSE
            (2)IF LC_EOI_DLY_AS_ENA = 1
            (2)THEN
                    SOI_1_HOMS[x] = EOI_1_HOMS_EXT[x]
                            + (T_DCHA_PER_S_1_1[x] + T_DCHA_DLY_S_1[x] + TI_1_S[x])
                            * FAC_N
            (2)ELSE
                    SOI_1_HOMS[x] = EOI_1_HOMS_EXT[x] + TI_1_S[x]*FAC_N
            (2)ENDIF
    (1) ENDIF
    EOI_2_HOMS[x] = 0
    EOI_3_HOMS[x] = 0
(0) ENDIF
```

Double injection

```
(0)IF INJ_MOD_SP_HOMS[x] = "HOMS"
(0)THEN
    (1) IF LC_SOI_1_HOMS_MAN = 1
```

```
    (1) THEN
            SOI_1_HOMS[x] = C_SOI_1_HOMS_MAN
    (1) ELSE
            SOI_1_HOMS[x] = SOI_1_HOMS_EXT[x]
    (1) ENDIF
    (2) IF LC_EOI_2_HOMS_MAN = 1
    (2) THEN
            EOI_2_HOMS[x] = C_EOI_2_HOMS_MAN
    (2) ELSE
            EOI_2_HOMS[x] = EOI_2_HOMS_EXT[x]
    (2) ENDIF
    (3)IF LC_EOI_DLY_AS_ENA = 1
    (3)THEN
            EOI_2_HOMS[x] = EOI_2_HOMS[x]
                    + (T_DCHA_PER_S_2_1[x] + T_DCHA_DLY_S_2[x]) * FAC_N
    (3)ENDIF
    EOI_3_HOMS[x] = 0
(0) ENDIF
```

Triple injection

```
(0)IF INJ_MOD_SP_HOMS[x] = "HOMS+PLS3"
(0)THEN
    (1) IF LC_SOI_1_HOMS_MAN = 1
    (1) THEN
            SOI_1_HOMS[x] = C_SOI_1_HOMS_MAN
    (1) ELSE
            SOI_1_HOMS[x] = SOI_1_HOMS_EXT[x]
    (1) ENDIF
    (2) IF LC_EOI_2_HOMS_MAN = 1
    (2) THEN
            EOI_2_HOMS[x] = C_EOI_2_HOMS_MAN
    (2) ELSE
            EOI_2_HOMS[x] = EOI_2_HOMS_EXT[x]
    (2) ENDIF
    (3)IF LC_EOI_DLY_AS_ENA = 1
    (3)THEN
            EOI_2_HOMS[x] = EOI_2_HOMS[x]
                    + (T_DCHA_PER_S_2_1[x] + T_DCHA_DLY_S_2[x]) * FAC_N
    (3)ENDIF
```

```
        (1) IF LC_EOI_3_HOMS_MAN = 1
        (1) THEN
                EOI_3_HOMS[x] = C_EOI_3_HOMS_MAN
        (1) ELSE
                EOI_3_HOMS[x] = EOI_3_HOMS_EXT[x]
        (1) ENDIF
        (2)IF LC_EOI_DLY_AS_ENA = 1
        (2)THEN
                EOI_3_HOMS[x] = EOI_3_HOMS[x]
                        + (T_DCHA_PER_S_3_1[x] + T_DCHA_DLY_S_3[x]) * FAC_N
        (2)ENDIF
    (0) ENDIF
    (0)ENDFOR
```

Check plausibility of calculated injection angles. If necessary, move the first pulse:

```
(5)FOR cylinder_start to cylinder_stop DO
    (6)IF INJ_MOD_SP_HOMS[x] = "HOMS+PLS3"
    (6) THEN
        (7) IF (EOI_3_HOMS[x]
                    + (TI_3_S[x] + T_DYL_2_3_MIN_HOMS[x] + T_DCHA_DLY_S_2[x]) *FAC_N)
              >  EOI_2_HOMS[x]
        (7) THEN
            EOI_2_HOMS[x] = EOI_3_HOMS[x]
                    + (TI_3_S[x] + T_DYL_2_3_MIN_HOMS[x] + T_DCHA_DLY_S_2[x]) * FAC_N
        (7) ENDIF
        (7) IF (EOI_2_HOMS[x]
                    + (TI_2_S[x] + NC_T_MIN_OFS_INJ_PLS + T_DCHA_DLY_S_1[x]) * FAC_N)
             > (SOI_1_HOMS[x] - TI_1_S[x] *FAC_N)
        (7) THEN
            SOI_1_HOMS[x] = EOI_2_HOMS[x]
                    + (TI_2_S[x] + TI_1_S[x] + NC_T_MIN_OFS_INJ_PLS + T_DCHA_DLY_S_1[x])
                    * FAC_N
        (7) ENDIF
    (6) ENDIF
    (6)IF INJ_MOD_SP_HOMS[x] = "HOMS"
    (6) THEN
        (7) IF (EOI_2_HOMS[x]
            + (TI_2_S[x] + T_DYL_2_3_MIN_HOMS[x] + T_DCHA_DLY_S_1[x]) * FAC_N)
              >  (SOI_1_HOMS[x] - TI_1_S[x] *FAC_N)
        (7) THEN
```

```
            SOI_1_HOMS[x] = EOI_2_HOMS[x]
            + (TI_2_S[x] + TI_1_S[x] + T_DYL_2_3_MIN_HOMS[x] + T_DCHA_DLY_S_1[x])
            * FAC_N
        (7) ENDIF
    (6) ENDIF
(5) ENDFOR
```

41.10 Injection phase for stratified mode, 1. Pulse (main pulse)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EOI_1_S[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of the first injection for stratified mode					
SOI_1_S[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
Start of the first injection phase for stratified mode					
CRK_DIF_SOI_IGN_S	O	0...F00H	0...1440	0.375	[°CRK]
Output variable for LACO					

Input data:

CRK_INJ_BAS[NC_CYL_NR]	CTR_CYL_NR_ST_CLC_INJR	EOI_1_S_EXT[NC_CYL_NR]	FAC_N
INJ_MOD_SP_S[NC_CYL_NR]	LC_EOI_DLY_AS_ENA	LV_S_CLC	LV_ST_END
NC_CYL_NR	NR_CYL_CLC_RED_INJ	SOI_2_S[NC_CYL_NR]	T_DCHA_DLY_S_1[NC_CYL_NR]
T_DCHA_PER_S_1_1[NC_CYL_NR]	T_DLY_1_2_S_EXT[NC_CYL_NR]	TI_1_S[NC_CYL_NR]	T_SOI_2_EOI_S[NC_CYL_NR]
EOI_2_S[NC_CYL_NR]	EOI_3_S[NC_CYL_NR]		

FUNCTION DESCRIPTION:

Application conditions:

Activation: LV_S_CLC = 1

Deactivation: -

Initialization at Reset: -

Recurrence: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment sychronous

Formula section:

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
              x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
```

```
    <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

```
FOR cylinder_start to cylinder_stop DO
(1) IF ~~INJ_MOD_SP_S[x] ≠ "SNGS"~~  INJ_MOD_SP_S[x] = "MPLS+PLS3"
(1) THEN
    ~~EOI_1_S[x] = SOI_2_S[x] + T_DLY_1_2_S_EXT[x] * FAC_N~~
    EOI_1_S[x] = EOI_3_S[x] + (T_SOI_2_EOI_S[x] + T_DLY_1_2_S_EXT[x]) * FAC_N
(1) ELSE
      (1a)IF INJ_MOD_SP_S[x] = "MPLS"
      (1a)THEN
                EOI_1_S[x] = EOI_2_S[x]
                          + (T_SOI_2_EOI_S[x] + T_DLY_1_2_S_EXT[x]) * FAC_N
      (1a)ELSE
                (2) IF LC_EOI_1_S_MAN = 1
                (2) THEN
                        EOI_1_S[x] = C_EOI_1_S_MAN
                (2) ELSE
                        EOI_1_S[x] = EOI_1_S_EXT[x]
                (2) ENDIF
                (3)IF LC_EOI_DLY_AS_ENA = 1
                (3)THEN
                        EOI_1_S[x] = EOI_1_S[x]
                                + (T_DCHA_PER_S_1_1[x] + T_DCHA_DLY_S_1[x]) * FAC_N
                (3)ENDIF
      (1a)ENDIF
(1) ENDIF
SOI_1_S[x] = EOI_1_S[x] + TI_1_S[x] * FAC_N
ENDFOR
```

Output variable for LACO (10 ms synchronous):

```
CRK_DIF_SOI_IGN_S = SOI_1_S[CTR_CYL_NR_ST_CLC_INJR] + CRK_INJ_BAS[0]
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EOI_1_S_MAN	1	0...780H	0...720	0.375	[°CRK]
Global application assistance for the second injection					
LC_EOI_1_S_MAN	1	0...1H	0...1	1	[-]
Global application assistance switch passive/active for the second injection					

41.11 Injection phase for stratified mode, 2. Pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EOI_2_S[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of the second injection for stratified mode					
~~SOI_2_S[NC_CYL_NR]~~	~~O/V~~	~~0...780H~~	~~0...720~~	~~0.375~~	~~[°CRK]~~
~~Start of the second injection for stratified mode~~					
T_SOI_2_EOI_S[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Time between start of second injection and end of last injection in stratified mode.					

Input data:

CTR_CYL_NR_ST_CLC_INJR	EOI_2_S_EXT[NC_CYL_NR]	FAC_N	INJ_MOD_SP_S[NC_CYL_NR]
LC_EOI_DLY_AS_ENA	LV_S_CLC	LV_ST_END	NC_CYL_NR
NR_CYL_CLC_RED_INJ	~~SOI_3_S[NC_CYL_NR]~~	T_DCHA_DLY_S_2[NC_CYL_NR]	T_DCHA_PER_S_2_1[NC_CYL_NR]
T_DLY_2_3_S_EXT[NC_CYL_NR]	TI_2_S[NC_CYL_NR]	EOI_3_S[NC_CYL_NR]	T_SOI_3_EOI_S[NC_CYL_NR]

FUNCTION DESCRIPTION:

Application conditions:

Activation: LV_S_CLC = 1

Deactivation: LV_S_CLC = 0

Initialization at Reset: -

Recurrence: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment sychronous

Formula section:

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
                x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
    <loop content>
```

```
ENDFOR
Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

FOR cylinder_start to cylinder_stop DO
(1)IF INJ_MOD_SP_S[x] = "MPLS+PLS3"
(1)THEN
    ~~EOI_2_S[x] = SOI_3_S[x] + T_DLY_2_3_S_EXT[x]* FAC_N~~
    ~~SOI_2_S[x] = EOI_2_S[x] + TI_2_S[x]*FAC_N~~

    EOI_2_S[x] = EOI_3_S[x] + (T_SOI_3_EOI_S[x] + T_DLY_2_3_S_EXT[x]) * FAC_N
    T_SOI_2_EOI_S[x] = T_SOI_3_EOI_S[x] + T_DLY_2_3_S_EXT[x] + TI_2_S[x]
(1)ELSE
    (2) IF INJ_MOD_SP_S[x] = "MPLS"
    (2) THEN
            (3) IF LC_EOI_2_S_MAN = 1
            (3) THEN
                    EOI_2_S[x] = C_EOI_2_S_MAN
            (3) ELSE
                    EOI_2_S[x] = EOI_2_S_EXT[x]
            (3) ENDIF
            (4)IF LC_EOI_DLY_AS_ENA = 1
            (4)THEN
                    ~~EOI_2_S[x] = EOI_2_S[x]~~
                        ~~+ (T_DCHA_PER_S_2_1[x] + T_DCHA_DLY_S_2[x]) * FAC_N~~

                    T_SOI_2_EOI_S[x] = T_DCHA_PER_S_2_1[x] + T_DCHA_DLY_S_2[x]
                                                    + TI_2_S[x]
            (4)ELSE
                    T_SOI_2_EOI_S[x] = TI_2_S[x]
            (4)ENDIF
            ~~SOI_2_S[x] = EOI_2_S[x] + TI_2_S[x]*FAC_N~~
    (2) ELSE
            ~~SOI_2_S[x] = 0~~
            EOI_2_S[x] = 0
            T_SOI_2_EOI_S[x] = 0
    (2) ENDIF
(1) ENDIF
ENDFOR
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EOI_2_S_MAN	1	0...780H	0...720	0.375	[°CRK]
Global application assistance for the second injection					
LC_EOI_2_S_MAN	1	0...1H	0...1	1	[-]
Global application assistance switch passive/active for the second injection					

41.12 Injection phase for stratified mode, 3. Pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EOI_3_S[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of the third injection for stratified mode					
~~SOI_3_S[NC_CYL_NR]~~	~~O/V~~	~~0...780H~~	~~0...720~~	~~0.375~~	~~[°CRK]~~
~~Start of the third injection for stratified mode~~					
T_SOI_3_EOI_S[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Time between start and end of injection of third pulse in stratified mode.					

Input data:

CTR_CYL_NR_ST_CLC_INJR	EOI_3_S_EXT[NC_CYL_NR]	~~FAC_N~~	INJ_MOD_SP_S[NC_CYL_NR]
LC_EOI_DLY_AS_ENA	LV_S_CLC	LV_ST_END	NC_CYL_NR
NR_CYL_CLC_RED_INJ	T_DCHA_DLY_S_3[NC_CYL_NR]	T_DCHA_PER_S_3_1[NC_CYL_NR]	TI_3_S[NC_CYL_NR]

FUNCTION DESCRIPTION:

Application conditions:

Activation: LV_S_CLC = 1

Deactivation: LV_S_CLC = 0

Initialization at Reset: -

Recurrence: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment sychronous

Formula section:

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
                x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
    <loop content>
```

```
ENDFOR
Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

FOR cylinder_start to cylinder_stop DO
    (1)IF INJ_MOD_SP_S[x] = "MPLS+PLS3"
    (1)THEN
      (2) IF LC_EOI_3_S_MAN = 1
      (2) THEN
            EOI_3_S[x] = C_EOI_3_S_MAN
      (2) ELSE
            EOI_3_S[x] = EOI_3_S_EXT[x]
      (2) ENDIF
      (3)IF LC_EOI_DLY_AS_ENA = 1
      (3)THEN
            ~~EOI_3_S[x] = EOI_3_S[x]~~
                    ~~+ (T_DCHA_PER_S_3_1[x] + T_DCHA_DLY_S_3[x]) * FAC_N~~
            T_SOI_3_EOI_S[x] = T_DCHA_PER_S_3_1[x] + T_DCHA_DLY_S_3[x]
                                       + TI_3_S[x]
      (3)ELSE
            T_SOI_3_EOI_S[x] = TI_3_S[x]
      (3)ENDIF
      ~~SOI_3_S[x] = EOI_3_S[x] + TI_3_S[x] * FAC_N~~
    (1)ELSE
      EOI_3_S[x] = 0
      ~~SOI_3_S[x] = 0~~
      T_SOI_3_EOI_S[x] = 0
    (1)ENDIF
ENDFOR
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EOI_3_S_MAN	1	0...780H	0...720	0.375	[°CRK]
Global application assistance for the third injection					
LC_EOI_3_S_MAN	1	0...1H	0...1	1	[-]
Global application assistance switch passive/active for the third injection					

41.13 Injection phase (application incidences)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CAM_SHIFT_EX	V/O	FC40...3C0H	-360...360	0.375	[°CRK]
Shift of the exhaust camshaft relative to passiv position					
CAM_SHIFT_IN	V/O	FC40...3C0H	-360...360	0.375	[°CRK]
Shift of the inlet camshaft relative to passiv position					
SOI_1_HOM_EXT[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
Start of injection for homogeneous mode, 1.pulse - base value from customer layer					
SOI_2_MAX_HOM_EXT	O/V	0...780H	0...720	0.375	[°CRK]
Earliest start of injection for the second pulse in homogeneous injection mode					
EOI_2_HOM_EXT[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
End of injection for homogeneous mode, 2.pulse – base value from customer layer					
EOI_3_HOM_EXT[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
End of injection for homogeneous mode, 3.pulse - base value from customer layer					
EOI_1_S_EXT[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
End of injection for stratified mode, 1.pulse - base value from customer layer					
EOI_2_S_EXT[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
End of injection for stratified mode, 2.pulse - base value from customer layer					
EOI_3_S_EXT[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of injection for stratified mode, 3.pulse - base value from customer layer					
SOI_1_HOMS_EXT[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
Start of injection for homogeneous stratified mode, 1.pulse - base value from customer layer					
EOI_1_HOMS_EXT[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of injection for homogeneous stratified mode, 1. pulse - base value from customer layer					
EOI_2_HOMS_EXT[NC_CYL_NR]	V/O	0...780H	0...720	0.375	[°CRK]
End of injection for homogeneous stratified mode, 2.pulse - base value from customer layer					
EOI_3_HOMS_EXT[NC_CYL_NR]	O/V	0...780H	0...720	0.375	[°CRK]
End of injection for homogeneous stratified mode, 3.pulse - base value from customer layer					
T_DLY_1_2_S_EXT[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Interval between first and second pulse of stratified injection ¿ base value from MSD70 layer					
T_DLY_2_3_S_EXT[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Interval between second and third pulse of stratified injection; base value from MSD70 layer					
T_DLY_2_3_MIN_HOMS[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Minimum time delay between second and third pulse for homogeneous-stratified mode					
LV_INJ_PLS_UPD_MPLH_DEAC	O/V	0...1H	0...1	1	[-]
Flag to deactivate pulse updates in multiple homogeneous mode					

Input data:

LC_EOI_DLY_AS_ENA	IGA_AV_H_RNG[NC_CYL_NR]	NC_CYL_NR	LV_ST_END
CRK_INJ_BAS[NC_CYL_NR]	EOI_3_S_CUS[NC_CYL_NR]	EOI_2_HOM_CUS[NC_CYL_NR]	SOI_1_HOMS_CUS[NC_CYL_NR]
SOI_1_HOM_CUS[NC_CYL_NR]	LV_EOI_2_DELTA_HOM_CUS	EOI_3_HOM_CUS[NC_CYL_NR]	EOI_2_HOMS_CUS[NC_CYL_NR]
EOI_3_HOMS_CUS[NC_CYL_NR]	T_DCHA_PER_S_1_1[NC_CYL_NR]	T_DCHA_PER_S_2_1[NC_CYL_NR]	T_DLY_1_2_S_CUS[NC_CYL_NR]
T_DLY_2_3_S_CUS[NC_CYL_NR]	T_DLY_2_3_MIN_HOMS_CUS[NC_CYL_NR]	SOI_2_MAX_CUS	LF_SOI_S_ENA
FAC_N	SOI_S_CUS[NC_CYL_NR]	TI_1_S[NC_CYL_NR]	TI_2_S[NC_CYL_NR]
TI_3_S[NC_CYL_NR]	LV_S_CLC	CTR_CYL_NR_ST_CLC_INJR	NR_CYL_CLC_RED_INJ
INJ_MOD_SP_S[NC_CYL_NR]	INJ_MOD_SP_HOMS[NC_CYL_NR]	INJ_MOD_SP_HOM[NC_CYL_NR]	STATE_INJ_MOD_HOM_REQ

T_DCHA_DLY_S_1[NC_CYL_NR]	T_DCHA_DLY_S_2[NC_CYL_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_T_DLY_1_2_S_EXT_MAN_ENA	1	0...1H	0...1	1	[-]
Switch to enable manual setpoint for time interval between first and second pulse of stratified injection					
LC_T_DLY_2_3_S_EXT_MAN_ENA	1	0...1H	0...1	1	[-]
Switch to enable manual setpoint for time interval between second and third pulse of stratified injection					
C_T_DLY_1_2_S_EXT_MAN	1	0...FFFFH	0...65.535	0.001	[ms]
Manual setpoint for time interval between first and second pulse of stratified injection					
C_T_DLY_2_3_S_EXT_MAN	1	0...FFFFH	0...65.535	0.001	[ms]
Manual setpoint for time interval between second and third pulse of stratified injection					

FUNCTION DESCRIPTION:

In this part, the conversion of injection phasing from customer nomenclature to SV nomenclature is done.

For calculation order purposes, the specification is divided in two parts. The part that calculates the delay times for stratified combustion mode must be calculated before the task "Final injection timing", the part that calculates the rest of the phasing must be calculated after.

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
             x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
   <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

41.13.1 Calculation of pulse delay times for stratified combustion mode

Note: This task must be calculated **before** the task "Final injection timing"!

Application conditions:

Activation: every engine state
Deactivation: -
Initialization: -
Recurrence: LV_ST_END = 0: 10ms
LV_ST_END = 1: segment synchronous

Formula section:

Deactivate pulse update during double injection in HOM mode with ignition related injection:

```
IF LV_EOI_2_DELTA_HOM_CUS = 1
    AND STATE_INJ_MOD_HOM_REQ ≠ MPLH+PLS3
THEN
    LV_INJ_PLS_UPD_MPLH_DEAC = 1
ELSE
    LV_INJ_PLS_UPD_MPLH_DEAC = 0
ENDIF
(1)FOR cylinder_start to cylinder_stop DO
    SOI_1_HOM_EXT[x] = SOI_1_HOM_CUS[x] – CRK_INJ_BAS[0]
    (1) IF LV_EOI_2_DELTA_HOM_CUS = 1
            AND INJ_MOD_SP_HOM[x] ≠ MPLH+PLS3
    (1) THEN
            EOI_2_HOM_EXT[x] = EOI_2_HOM_CUS[x] + (IGA_AV_H_RNG[x] -
                                    CRK_INJ_BAS[0])
    (1) ELSE
            EOI_2_HOM_EXT[x] = EOI_2_HOM_CUS[x]  - CRK_INJ_BAS[0]
    (1) ENDIF
    (1a) IF LV_EOI_2_DELTA_HOM_CUS = 1
            AND INJ_MOD_SP_HOM[x] = MPLH+PLS3
    (1a) THEN
            EOI_3_HOM_EXT[x] = EOI_3_HOM_CUS[x] + (IGA_AV_H_RNG[x] -
                                    CRK_INJ_BAS[0])
    (1a) ELSE
            EOI_3_HOM_EXT[x] = EOI_3_HOM_CUS[x]  - CRK_INJ_BAS[0]
    (1a) ENDIF
```

```
        (1)IF LV_S_CLC = 1
        (1)THEN
                Calculation of pulse delay times:
                (2)IF INJ_MOD_SP_S[x] = "MPLS+PLS3" (triple injection)
                (2)THEN
                        T_DLY_1_2_S_EXT[x] =  T_DLY_1_2_S_CUS[x]
                        T_DLY_2_3_S_EXT[x] =  T_DLY_2_3_S_CUS[x]
                (2)ENDIF
                (3)IF INJ_MOD_SP_S[x] = "MPLS" (double injection)
                (3)THEN
                        T_DLY_1_2_S_EXT[x] =  T_DLY_2_3_S_CUS[x]
                (3)ENDIF
                Manual setting of pulse delay times:
                (4)IF LC_T_DLY_1_2_S_EXT_MAN_ENA = 1
                (4)THEN
                        T_DLY_1_2_S_EXT[x] = C_T_DLY_1_2_S_EXT_MAN
                (4)ENDIF
                (5)IF LC_T_DLY_2_3_S_EXT_MAN_ENA = 1
                (5)THEN
                        T_DLY_2_3_S_EXT[x] = C_T_DLY_2_3_S_EXT_MAN
                (5)ENDIF
        (1)ENDIF
    (1)ENDFOR
```

41.13.2 Calculation of injection phasing

Note: This task must be calculated **after** the task "Final injection timing"!

Application conditions:

Activation: every engine state

Deactivation: -

Initialization: -

Recurrence: LV_ST_END = 0: 10ms

LV_ST_END = 1: segment synchronous

Formula section:

```
CAM_SHIFT_EX = 0
CAM_SHIFT_IN = 0
SOI_2_MAX_HOM_EXT = SOI_2_MAX_CUS – CRK_INJ_BAS[0]

(1)FOR cylinder_start to cylinder_stop DO

    (0)IF LV_S_CLC = 1
    (0)THEN

        Mapping for stratified combustion mode
        (1)IF INJ_MOD_SP_S[x] = "MPLS+PLS3" (triple injection)
        (1) THEN
            (1a)IF Bit x of LF_SOI_S_ENA = 0
            (1a)THEN

                EOI_3_S_EXT[x] = EOI_3_S_CUS[x] – CRK_INJ_BAS[0]

            (1a)ELSE

                EOI_3_S_EXT[x] = SOI_S_CUS[x] – CRK_INJ_BAS[0]
                                                     – (TI_3_S[x] * FAC_N)

            (1a)ENDIF

            (1b)IF LC_EOI_DLY_AS_ENA = 1
            (1b)THEN – transfer pulse delay times to BSW convention

                T_DLY_1_2_S_EXT[x] =  T_DLY_1_2_S_EXT[x]
                        + T_DCHA_PER_S_1_1[x] + T_DCHA_DLY_S_1[x]

                T_DLY_2_3_S_EXT[x] =  T_DLY_2_3_S_EXT[x]
                        + T_DCHA_PER_S_2_1[x] + T_DCHA_DLY_S_2[x]

            (1b)ENDIF

        (1) ENDIF

        (2)IF INJ_MOD_SP_S[x] = "MPLS" (double injection)
        (2) THEN
            (2a)IF Bit x of LF_SOI_S_ENA = 0
            (2a)THEN

                EOI_2_S_EXT[x] = EOI_3_S_CUS[x] – CRK_INJ_BAS[0]

            (2a)ELSE

                EOI_2_S_EXT[x] = SOI_S_CUS[x] – CRK_INJ_BAS[0]
                                                     – (TI_2_S[x] * FAC_N)

            (2a)ENDIF

            (2b)IF LC_EOI_DLY_AS_ENA = 1
            (2b)THEN – transfer pulse delay times to BSW convention

                T_DLY_1_2_S_EXT[x] =  T_DLY_1_2_S_EXT[x]
                        + T_DCHA_PER_S_1_1[x] + T_DCHA_DLY_S_1[x]

            (2b)ENDIF
```

```
            (2) ENDIF

            (3)IF INJ_MOD_SP_S[x] = "SNGS" (single injection)
            (3) THEN
                (2a)IF Bit x of LF_SOI_S_ENA = 0
                (2a)THEN

                    EOI_1_S_EXT[x] = EOI_3_S_CUS[x] – CRK_INJ_BAS[0]

                (2a)ELSE

                    EOI_1_S_EXT[x] = SOI_S_CUS[x] – CRK_INJ_BAS[0]
                                                        – (TI_1_S[x] * FAC_N)

                (2a)ENDIF

            (3) ENDIF

            Mapping for homogeneous stratified combustion mode
            (1)IF INJ_MOD_SP_HOMS[x] = "HOMS+PLS3" (triple injection)
                EOI_3_HOMS_EXT[x] = EOI_3_HOMS_CUS[x] – CRK_INJ_BAS[0]
                EOI_2_HOMS_EXT[x] = EOI_2_HOMS_CUS[x] – CRK_INJ_BAS[0]
                SOI_1_HOMS_EXT[x] = SOI_1_HOMS_CUS[x] – CRK_INJ_BAS[0]
            (1) ENDIF

            (2)IF INJ_MOD_SP_HOMS[x] = "HOMS" (double injection)
                EOI_2_HOMS_EXT[x] = EOI_3_HOMS_CUS[x] – CRK_INJ_BAS[0]
                SOI_1_HOMS_EXT[x] = SOI_1_HOMS_CUS[x] – CRK_INJ_BAS[0]
            (2) ENDIF

            (3)IF INJ_MOD_SP_HOMS[x] = "HOMS-SNG" (single injection)
                EOI_1_HOMS_EXT[x] = EOI_3_HOMS_CUS[x] – CRK_INJ_BAS[0]
            (3) ENDIF
            T_DLY_2_3_MIN_HOMS[x] = T_DLY_2_3_MIN_HOMS_CUS[x]

        (0)ENDIF – LV_S_CLC

    (1) ENDFOR
```

41.14 Injection Pressure Correction – Homogeneous Mode, 1. Pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TI_1_PRS_CYL_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
correction factor to compensate the influence of the cylinder counter pressure, 1. pulse homogeneous					
PRS_DEC_INJ_1_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure difference at the injector, 1. pulse homogeneous mode					
FAC_TI_1_PRS_CYL_HOM_TMP	-	0...FFFFH	0...15.99975	0.2441e-3	[-]
correction factor to compensate the influence of the cylinder counter pressure, 1. pulse homogeneous; temporary value					
PRS_DEC_INJ_1_HOM_TMP	-	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure difference at the injector, 1. pulse homogeneous mode; temporary value					

Input data:

LV_EGY_RNG_IV_PLS_1_HOM[NC_CYL_NR]	IDX_PRS_COR_CYL_CLC_MPLH[NC_CBK_HPP_NR]	MAP_INJ[NC_CBK_HPP_NR]	LV_ST_END
FUP_H_INJ[NC_CBK_HPP_NR]	NC_CBK_HPP_NR	NC_CYL_NR	NC_IDX_CYL_HPP_REF[NC_CYL_NR]

FUNCTION DESCRIPTION:

General information:

The module calculates a correction factor to compensate the influence of the cylinder counter pressure at the injector (here: MAP) and the influence of the fuel temperature on the injected fuel mass. Also, the pressure difference at the injector is calculated.

Application conditions:

Activation: every engine state

Deactivation: -

Initialisation: at reset: FAC_TI_1_PRS_CYL_HOM[ ] = 1 // all NC_CYL_NR elements

Update rate: LV_ST_END = 0: 10 ms

LV_ST_END = 1: segment synchronous

Formula section:

```
FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
    The pressure difference at the injector is calculated from fuel rail pressure and the manifold pressure.

    PRS_DEC_INJ_1_HOM_TMP = FUP_H_INJ[m] – MAP_INJ[m]

    The correction factor FAC_TI_1_PRS_CYL_HOM is used to compensate the influence of the cylinder counter pressure and the fuel tempreature on the injector needle lift. In case of the first pulse in homogeneous mode, the cylinder couter pressure equals MAP.
    A separate map is used for low energy pulses.

    IF LV_EGY_RNG_IV_PLS_1_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = 1
    THEN
        FAC_TI_1_PRS_CYL_HOM_TMP = IP_FAC_TI_PRS_CYL(MAP_INJ[m])
    ELSE
        FAC_TI_1_PRS_CYL_HOM_TMP = IP_FAC_TI_PRS_CYL_L(MAP_INJ[m])
    ENDIF

    Calculate cylinder individual injection pressure correction values:
    FOR x = 0 TO NC_CYL_NR – 1 DO
        IF NC_IDX_CYL_HPP_REF[x] = m
        THEN
            PRS_DEC_INJ_1_HOM[x] = PRS_DEC_INJ_1_HOM_TMP
            FAC_TI_1_PRS_CYL_HOM[x] = FAC_TI_1_PRS_CYL_HOM_TMP
        ENDIF
    ENDFOR
ENDFOR
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_TI_PRS_CYL	12	0...FFFFH	0...15.99975	0.2441e-3	[-]
LDP_PRS_INJ_IP_FAC_TI_PRS_CYL	12	0...FFFFH	0...347776	5.3067216	[hPa]
correction factor to compesate the influence of the cylinder couter pressure					
IP_FAC_TI_PRS_CYL_L	12	0...FFFFH	0...15.99975	0.2441e-3	[-]
LDP_PRS_INJ_IP_FAC_TI_PRS_CYL	12	0...FFFFH	0...347776	5.3067216	[hPa]
correction factor to compesate the influence of the cylinder couter pressure for low injector needle lift.					

41.15 Injection Pressure Correction for Double Injection – Homogeneous Mode, 2. Pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TI_2_PRS_CYL_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Correction factor to compensate the influence of the cylinder counter pressure, 2. Pulse homogeneous					
PRS_INJ_MV_2_HOM	V	0...FFFFH	0...347776	5.3067216	[hPa]
mean cylinder pressure during injection, 2. Pulse					
CRK_INJ_2_HOM	V	0...780H	0...720	0.375	[°CRK]
rough value for injection angle (EOI-SOI), 2. pulse					
PRS_DEC_INJ_2_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure difference at the injector, 2. pulse homogeneous mode					
TI_2_PRS_HOM_SP[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Injection time setpoint for calculation of injection pressure correction at homogeneous mode, second pulse					
FAC_TI_2_PRS_CYL_HOM_TMP	-	0...FFFFH	0...15.99975	0.2441e-3	[-]
Correction factor to compensate the influence of the cylinder counter pressure, 2. Pulse homogeneous; temporary value					
PRS_DEC_INJ_2_HOM_TMP	-	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure difference at the injector, 2. pulse; temporary value					
TI_2_PRS_HOM_SP_TMP	-	0...FFFFH	0...65.535	0.001	[ms]
Injection time setpoint for calculation of injection pressure correction at homogeneous mode, second pulse; temporary value					
IDX_TI_2_PRS_HOM_SP_TMP	-	0...FFFFH	0...65535	1	[-]
Temporary counting factor index for pressure correction calculations; second pulse HOM.					

Input data:

	C_IDX_TI_MIN_H	C_IDX_TI_MIN_L	C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H]
C_T_DCHA_DLY_L[NC_NR_IDX_TI_PRM_L]	C_TI_H[NC_NR_IDX_TI_PRM_H]	C_TI_L[NC_NR_IDX_TI_PRM_L]	C_TI_OFS_H
C_TI_OFS_L	EOI_2_HOM[NC_CYL_NR]	FAC_MFF_DIF_HOM[NC_CYL_NR]	FAC_MFF_TFU
FAC_N	FAC_TI_COR_IV_EGY_RNG_L[NC_CBK_HPP_NR]	FUP_H_INJ[NC_CBK_HPP_NR]	IDX_PRS_COR_CYL_CLC_MPLH[NC_CBK_HPP_NR]
INJ_MOD_SP_HOM[NC_CYL_NR]	IP_FAC_TI_PRS_CYL	IP_FAC_TI_PRS_CYL_L	IP_MFF_IDX_TI_EGY_H
IP_MFF_IDX_TI_EGY_L	~~IP_MFF_TI_EGY_H~~	~~IP_MFF_TI_EGY_L~~	LF_PRS_COR_HPP_ENA
LV_EGY_RNG_IV_PLS_2_HOM[NC_CYL_NR]	LV_ST_END	MAP_INJ[NC_CBK_HPP_NR]	MFF_SP_2_HOM[NC_CYL_NR]
NC_CBK_HPP_NR	NC_CYL_NR	NC_IDX_CYL_HPP_REF[NC_CYL_NR]	NC_NR_IDX_TI_PRM_H
NC_NR_IDX_TI_PRM_L			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_PRS_INC_CMP	12*16	0...FFFFH	0...31.99951	0.4883e-3	[-]
LDP_EOI_IP_PRS_INC_CMP	12	0...780H	0...720	0.375	[°CRK]
LDP_CRK_INJ_IP_PRS_INC_CMP	16	0...780H	0...720	0.375	[°CRK]
Mean pressure increase in cylinder due to compression					

Overview

The module calculates a correction factor to compensate the influence of the cylinder counter pressure and the influence of the fuel temperature on the injected fuel mass.

Note for Software: IDX_TI_TMP_PRS is a temporary calculation value with same resolution and range as IDX_TI_2_PRS_HOM_SP_TMP.

Application Condition

Activation:	every engine state
Deactivation:	-
Initialization:	-
Update rate:	LV_ST_END = 0: 10 ms LV_ST_END = 1: every TDC

Function Description

```
(1)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO

   (1)IF INJ_MOD_SP_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = "MPLH" OR
         INJ_MOD_SP_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = "MPLH+PLS3"
   (1) THEN

    Check, if fuel bank m has to be calculated:
    (2)IF Bit m of LF_PRS_COR_HPP_ENA = 1
    (2)THEN

      Calculation of an injection time for pressure correction:
      (3)IF LV_EGY_RNG_IV_PLS_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = 1
      (3)THEN
        IDX_TI_2_PRS_HOM_SP_TMP = IP_MFF_IDX_TI_EGY_H(
             (MFF_SP_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]*(1 +
             FAC_MFF_DIF_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]])
             *FAC_MFF_TFU);
             PRS_DEC_INJ_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1))
      (3)ELSE
        IDX_TI_2_PRS_HOM_SP_TMP = IP_MFF_IDX_TI_EGY_L(
             (MFF_SP_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]*(1 +
             FAC_MFF_DIF_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]])
             *FAC_MFF_TFU);
             PRS_DEC_INJ_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1))
             * FAC_TI_COR_IV_EGY_RNG_L[m]

      (3)ENDIF
```

```
IDX_TI_2_PRS_HOM_SP_TMP = IDX_TI_2_PRS_HOM_SP_TMP
          * FAC_TI_2_PRS_CYL_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1)

(3)IF LV_EGY_RNG_IV_PLS_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = 0
   AND IDX_TI_2_PRS_HOM_SP_TMP >= C_IDX_TI_MIN_L
(3)THEN – low injector needle lift with special calculation

   IDX_TI_TMP_PRS = MIN( (IDX_TI_2_PRS_HOM_SP_TMP – C_IDX_TI_MIN_L)
                                             ; (NC_NR_IDX_TI_PRM_L – 1) )

   (3a)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_L – 1
   (3a)THEN

          TI_2_PRS_HOM_SP_TMP =
                 IDX_TI_2_PRS_HOM_SP_TMP * 0.001ms
                 + C_TI_OFS_L
                 + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]

   (3a)ELSE

          TI_2_PRS_HOM_SP_TMP = C_TI_L[IDX_TI_TMP_PRS]
                        + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]

   (3a)ENDIF

(3)ELSE – high injector needle lift or low lift identical to high lift

   IDX_TI_TMP_PRS =
          MIN( (IDX_TI_2_PRS_HOM_SP_TMP – C_IDX_TI_MIN_H) ;
                                             (NC_NR_IDX_TI_PRM_H – 1) )
                                             // limit calculation result to zero

   (3b)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_H – 1
   (3b)THEN

          TI_2_PRS_HOM_SP_TMP =
                 IDX_TI_2_PRS_HOM_SP_TMP * 0.001ms
                 + C_TI_OFS_H
                 + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]

   (3b)ELSE

          TI_2_PRS_HOM_SP_TMP = C_TI_H[IDX_TI_TMP_PRS]
                 + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]

   (3b)ENDIF

(3)ENDIF
```

Calculation of the counter pressure within the cylinder:

A rough injection duration angle CRK_INJ_2_HOM is estimated from TI_2_PRS_HOM_SP_TMP, FAC_TI_2_PRS_CYL_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1) and engine speed (FAC_N):

From the manifold pressure MAP, the end of injection EOI_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] and the rough injection angle a mean counter pressure within the cylinder during the injection is calculated.

The pressure difference at the injector is calculated from fuel rail pressure and the mean counter pressure.

The total correction factor results from the map IP_FAC_TI_PRS_CYL.
For low injector needle lift, a separate map IP_FAC_TI_PRS_CYL_L is used.

```
        CRK_INJ_2_HOM = TI_2_PRS_HOM_SP_TMP * FAC_N
           ~~* FAC_TI_2_PRS_CYL_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1)~~

        PRS_INJ_MV_2_HOM = MAP_INJ[m] * IP_PRS_INC_CMP
        (Input data for IP_PRS_INC_CMP is
        EOI_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] and CRK_INJ_2_HOM)

        PRS_DEC_INJ_2_HOM_TMP = FUP_H_INJ[m] - PRS_INJ_MV_2_HOM

        (3a)IF LV_EGY_RNG_IV_PLS_2_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = 1
        (3a)THEN
           FAC_TI_2_PRS_CYL_HOM_TMP =
                                 IP_FAC_TI_PRS_CYL(PRS_INJ_MV_2_HOM)
        (3a)ELSE
           FAC_TI_2_PRS_CYL_HOM_TMP =
                                 IP_FAC_TI_PRS_CYL_L(PRS_INJ_MV_2_HOM)
        (3a)ENDIF

        (2)FOR x = 0 TO NC_CYL_NR – 1 DO
           (4)IF NC_IDX_CYL_HPP_REF[x] = m
           (4)THEN
                 PRS_DEC_INJ_2_HOM[x] = PRS_DEC_INJ_2_HOM_TMP
                 FAC_TI_2_PRS_CYL_HOM[x] = FAC_TI_2_PRS_CYL_HOM_TMP
                 TI_2_PRS_HOM_SP[x] = TI_2_PRS_HOM_SP_TMP
           (4)ENDIF
        (2)ENDFOR
      (2)ENDIF

   (1)ELSE
      (3)FOR x = 0 TO NC_CYL_NR – 1 DO
         (5)IF NC_IDX_CYL_HPP_REF[x] = m
         (5)THEN
                 FAC_TI_2_PRS_CYL_HOM[x] = 1.0
                 PRS_DEC_INJ_2_HOM[x] = FUP_H_INJ[m]
         (5)ENDIF
      (3)ENDFOR
   (1)ENDIF
```

(1)ENDFOR

41.16 Injection Pressure Correction for Double Injection – Homogeneous Mode, 3. Pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TI_3_PRS_CYL_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Correction factor to compensate the influence of the cylinder counter pressure, 3. Pulse homogeneous					
PRS_INJ_MV_3_HOM	V	0...FFFFH	0...347776	5.3067216	[hPa]
mean cylinder pressure during injection, 3. Pulse					
CRK_INJ_3_HOM	V	0...780H	0...720	0.375	[°CRK]
rough value for injection angle (EOI-SOI), 3. pulse					
PRS_DEC_INJ_3_HOM[NC_CYL_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure difference at the injector, 3. pulse homogeneous mode					
TI_3_PRS_HOM_SP[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Injection time setpoint for calculation of injection pressure correction at homogeneous mode, third pulse					
FAC_TI_3_PRS_CYL_HOM_TMP	-	0...FFFFH	0...15.99975	0.2441e-3	[-]
Correction factor to compensate the influence of the cylinder counter pressure, 3. Pulse homogeneous; temporary value					
PRS_DEC_INJ_3_HOM_TMP	-	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure difference at the injector, 3. pulse homogeneous mode; temporary value					
TI_3_PRS_HOM_SP_TMP	-	0...FFFFH	0...65.535	0.001	[ms]
Injection time setpoint for calculation of injection pressure correction at homogeneous mode, third pulse; temporary value					
IDX_TI_3_PRS_HOM_SP_TMP	-	0...FFFFH	0...65535	1	[-]
Temporary counting factor index for pressure correction calculations; third pulse HOM.					

Input data:

C_IDX_TI_MIN_H	C_IDX_TI_MIN_L	C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H]	C_T_DCHA_DLY_L[NC_NR_IDX_TI_PRM_L]
C_TI_H[NC_NR_IDX_TI_PRM_H]	C_TI_L[NC_NR_IDX_TI_PRM_L]	C_TI_OFS_H	C_TI_OFS_L
EOI_3_HOM[NC_CYL_NR]	FAC_MFF_DIF_HOM[NC_CYL_NR]	FAC_MFF_TFU	FAC_N
FAC_TI_COR_IV_EGY_RNG_L[NC_CBK_HPP_NR]	FUP_H_INJ[NC_CBK_HPP_NR]	IDX_PRS_COR_CYL_CLC_MPLH[NC_CBK_HPP_NR]	INJ_MOD_SP_HOM[NC_CYL_NR]
IP_FAC_TI_PRS_CYL	IP_FAC_TI_PRS_CYL_L	IP_MFF_IDX_TI_EGY_H	IP_MFF_IDX_TI_EGY_L
~~IP_MFF_TI_EGY_H~~	~~IP_MFF_TI_EGY_L~~	IP_PRS_INC_CMP	LF_PRS_COR_HPP_ENA
LV_EGY_RNG_IV_PLS_3_HOM[NC_CYL_NR]	LV_ST_END	MAP_INJ[NC_CBK_HPP_NR]	MFF_SP_3_HOM[NC_CYL_NR]
NC_CBK_HPP_NR	NC_CYL_NR	NC_IDX_CYL_HPP_REF[NC_CYL_NR]	NC_NR_IDX_TI_PRM_H
NC_NR_IDX_TI_PRM_L			

Overview

The module calculates a correction factor to compensate the influence of the cylinder counter pressure and the influence of the fuel temperature on the injection time..

Note for Software: IDX_TI_TMP_PRS is a temporary calculation value with same resolution and range as IDX_TI_3_PRS_HOM_SP_TMP.

Application Condition

Activation: every engine state

Deactivation: -

Initialisation: -

Update rate: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

Function Description

```
(1)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO

   (1) IF (INJ_MOD_SP_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = "MPLH+PLS3")
   (1) THEN

      Check, if fuel bank m has to be calculated:
      (2)IF Bit m of LF_PRS_COR_HPP_ENA = 1
      (2)THEN

         Calculation of an injection time for pressure correction:
         (3)IF LV_EGY_RNG_IV_PLS_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = 1
         (3)THEN
           IDX_TI_3_PRS_HOM_SP_TMP = IP_MFF_IDX_TI_EGY_H(
              (MFF_SP_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]*(1 +
              FAC_MFF_DIF_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]])*FAC_MFF_TFU);
                     PRS_DEC_INJ_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1))
         (3)ELSE
           IDX_TI_3_PRS_HOM_SP_TMP = IP_MFF_IDX_TI_EGY_L(
              (MFF_SP_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]]*(1 +
              FAC_MFF_DIF_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]])*FAC_MFF_TFU);
                     PRS_DEC_INJ_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1))
                                      * FAC_TI_COR_IV_EGY_RNG_L[m]
         (3)ENDIF


           IDX_TI_3_PRS_HOM_SP_TMP = IDX_TI_3_PRS_HOM_SP_TMP
                     * FAC_TI_3_PRS_CYL_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1)

           (4)IF LV_EGY_RNG_IV_PLS_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = 0
              AND IDX_TI_3_PRS_HOM_SP_TMP >= C_IDX_TI_MIN_L
           (4)THEN – low injector needle lift with special calculation

              IDX_TI_TMP_PRS = MIN( (IDX_TI_3_PRS_HOM_SP_TMP – C_IDX_TI_MIN_L)
                                                ; (NC_NR_IDX_TI_PRM_L – 1) )

              (4a)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_L – 1
              (4a)THEN

                     TI_3_PRS_HOM_SP_TMP =
                            IDX_TI_3_PRS_HOM_SP_TMP * 0.001ms
                            + C_TI_OFS_L
                            + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]

              (4a)ELSE
```

```
            TI_3_PRS_HOM_SP_TMP = C_TI_L[IDX_TI_TMP_PRS]
                        + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]

        (4a)ENDIF

    (4)ELSE – high injector needle lift or low lift identical to high lift

        IDX_TI_TMP_PRS =
            MIN( (IDX_TI_3_PRS_HOM_SP_TMP – C_IDX_TI_MIN_H) ;
                                        (NC_NR_IDX_TI_PRM_H – 1) )
                                        // limit calculation result to zero

        (4b)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_H – 1
        (4b)THEN

            TI_3_PRS_HOM_SP_TMP =
                IDX_TI_3_PRS_HOM_SP_TMP * 0.001ms
                + C_TI_OFS_H
                + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]

        (4b)ELSE

            TI_3_PRS_HOM_SP_TMP = C_TI_H[IDX_TI_TMP_PRS]
                + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]

        (4b)ENDIF

    (4)ENDIF
```

<u>Calculation of the counter pressure within the cylinder:</u>
A rough injection duration angle CRK_INJ_3_HOM is estimated from TI_3_PRS_HOM_SP_TMP, FAC_TI_3_PRS_CYL_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1) and engine speed (FAC_N):

From the manifold pressure MAP, the end of injection EOI_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] and the rough injection angle a mean counter pressure within the cylinder during the injection is calculated.

The pressure difference at the injector is calculated from fuel rail pressure and the mean counter pressure.

The total correction factor results from the map IP_FAC_TI_PRS_CYL.
For low injector needle lift, a separate map IP_FAC_TI_PRS_CYL_L is used.

CRK_INJ_3_HOM = TI_3_PRS_HOM_SP_TMP * FAC_N
~~* FAC_TI_3_PRS_CYL_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]](n-1)~~

PRS_INJ_MV_3_HOM = MAP_INJ[m] * IP_PRS_INC_CMP
(Input data for IP_PRS_INC_CMP is
EOI_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] and CRK_INJ_3_HOM)

PRS_DEC_INJ_3_HOM_TMP = FUP_H_INJ[m] - PRS_INJ_MV_3_HOM

```
(3a)IF LV_EGY_RNG_IV_PLS_3_HOM[IDX_PRS_COR_CYL_CLC_MPLH[m]] = 1
(3a)THEN

    FAC_TI_3_PRS_CYL_HOM_TMP =
                    IP_FAC_TI_PRS_CYL(PRS_INJ_MV_3_HOM)
```

```
        (3a)ELSE
            FAC_TI_3_PRS_CYL_HOM_TMP =
                                        IP_FAC_TI_PRS_CYL_L(PRS_INJ_MV_3_HOM)
        (3a)ENDIF

        (2)FOR x = 0 TO NC_CYL_NR - 1 DO
            (4)IF NC_IDX_CYL_HPP_REF[x] = m
            (4)THEN
                    PRS_DEC_INJ_3_HOM[x] = PRS_DEC_INJ_3_HOM_TMP
                    FAC_TI_3_PRS_CYL_HOM[x] = FAC_TI_3_PRS_CYL_HOM_TMP
                    TI_3_PRS_HOM_SP[x] = TI_3_PRS_HOM_SP_TMP
            (4)ENDIF
        (2)ENDFOR
      (2)ENDIF

    (1)ELSE
        (3)FOR x = 0 TO NC_CYL_NR – 1 DO
            (5)IF NC_IDX_CYL_HPP_REF[x] = m
            (5)THEN
                    FAC_TI_3_PRS_CYL_HOM[x] = 1.0
                    PRS_DEC_INJ_3_HOM[x] = FUP_H_INJ[m]
            (5)ENDIF
        (3)ENDFOR
    (1)ENDIF
  (1)ENDFOR
```

41.17 Injection Pressure Correction for Double Injection – Stratified Mode, 1. Pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TI_1_PRS_CYL_S[NC_CYL_NR]	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
factor for correction of fuel injection time due to an influence of cylinder counter pressure, strat. mode 1. pulse					
FAC_TI_1_PRS_CYL_S_TMP	-	0...FFFFH	0...15.99975	0.2441e-3	[-]
factor for correction of fuel injection time due to an influence of cylinder counter pressure, strat. mode 1. pulse; temporary value					
PRS_INJ_MV_1_S	V	0...FFFFH	0...347776	5.3067216	[hPa]
mean cylinder pressure during injection, stratified mode 1. pulse					
CRK_INJ_1_S	V	0...780H	0...720	0.375	[°CRK]
rough value for injection angle (EOI-SOI), stratified mode 1. pulse					
PRS_DEC_INJ_1_S[NC_CYL_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
pressure decrease at the injector, stratified mode 1. pulse					
PRS_DEC_INJ_1_S_TMP	-	0...FFFFH	0...347776	5.3067216	[hPa]
pressure decrease at the injector, stratified mode 1. pulse; temporary value					
FUP_PRS_COR[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel pressure for calculation of the pressure correction at stratified mode					
TI_1_PRS_S_SP_TMP	-	0...FFFFH	0...65.535	0.001	[ms]
Injection time setpoint for calculation of injection pressure correction, stratified mode, 1. pulse; temporary value					
IDX_TI_1_PRS_S_SP_TMP	-	0...FFFFH	0...65535	1	[-]
Temporary counting factor index for pressure correction calculations; first pulse S.					

Input data:

C_IDX_TI_MIN_H	C_IDX_TI_MIN_L	C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H]	C_T_DCHA_DLY_L[NC_NR_IDX_TI_PRM_L]
C_TI_H[NC_NR_IDX_TI_PRM_H]	C_TI_L[NC_NR_IDX_TI_PRM_L]	C_TI_OFS_H	C_TI_OFS_L
EOI_1_S[NC_CYL_NR]	FAC_MFF_DIF_S[NC_CYL_NR]	FAC_MFF_TFU	FAC_N
FAC_TI_COR_IV_EGY_RNG_L[NC_CBK_HPP_NR]	FUP_H_INJ[NC_CBK_HPP_NR]	FUP_H_SP_S	FUP_H_SP_S_INJ[NC_CBK_HPP_NR]
IDX_PRS_COR_CYL_CLC_S[NC_CBK_HPP_NR]	IP_FAC_TI_PRS_CYL	IP_FAC_TI_PRS_CYL_L	IP_MFF_IDX_TI_EGY_H
IP_MFF_IDX_TI_EGY_L	~~IP_MFF_TI_EGY_H~~	~~IP_MFF_TI_EGY_L~~	LF_PRS_COR_HPP_ENA
LV_EGY_RNG_IV_PLS_1_S[NC_CYL_NR]	LV_IGA_AND_INJ_SWI	LV_S_CLC	LV_ST_END
MAP_INJ[NC_CBK_HPP_NR]	MFF_SP_1_S[NC_CYL_NR]	NC_CBK_HPP_NR	NC_CYL_NR
NC_IDX_CYL_HPP_REF[NC_CYL_NR]	NC_NR_IDX_TI_PRM_H	NC_NR_IDX_TI_PRM_L	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_PRS_INC_CMP_S	12*16	0...FFFFH	0...31.99951	0.4883e-3	[-]
LDP_EOI_2_S_0_IP_PRS_INC_CMP_S	12	0...780H	0...720	0.375	[°CRK]
LDP_CRK_INJ_2_S_IP_PRS_INC_CMP	16	0...780H	0...720	0.375	[°CRK]
Mean pressure increase in cylinder due to compression					

FUNCTION DESCRIPTION:

General information:

The module calculates a correction factor to compensate the influence of the cylinder counter pressure and the influence of the fuel temperature on the injected fuel mass.

Note for Software: IDX_TI_TMP_PRS is a temporary calculation value with same resolution and range as IDX_TI_1_PRS_S_SP_TMP.

Application conditions:

Activation: every engine state

Deactivation: -

Initialisation: -

Update rate: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

Formula section:

```
(1)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO

    (1) IF
    LV_S_CLC = 1
    (1) THEN
      Check, if fuel bank m has to be calculated:
      (2)IF Bit m of LF_PRS_COR_HPP_ENA = 1
      (2)THEN
          The injection time TI_1_PRS_S_SP_TMP is necessary to calculate CRK_INJ_1_S.
          Calculation of TI_1_PRS_S_SP_TMP:
          (3)IF LV_EGY_RNG_IV_PLS_1_S[IDX_PRS_COR_CYL_CLC_S[m]] = 1
          (3)THEN
              IDX_TI_1_PRS_S_SP_TMP = IP_MFF_IDX_TI_EGY_H(
                  (MFF_SP_1_S[IDX_PRS_COR_CYL_CLC_S[m]]*(1 +
                  FAC_MFF_DIF_S[IDX_PRS_COR_CYL_CLC_S[m]])*FAC_MFF_TFU);
                              PRS_DEC_INJ_1_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1))
          (3)ELSE
              IDX_TI_1_PRS_S_SP_TMP = IP_MFF_IDX_TI_EGY_L(
                  (MFF_SP_1_S[IDX_PRS_COR_CYL_CLC_S[m]]*(1 +
                  FAC_MFF_DIF_S[IDX_PRS_COR_CYL_CLC_S[m]])*FAC_MFF_TFU);
                              PRS_DEC_INJ_1_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1))
                                          * FAC_TI_COR_IV_EGY_RNG_L[m]
          (3)ENDIF

          IDX_TI_1_PRS_S_SP_TMP = IDX_TI_1_PRS_S_SP_TMP
```

```
        * FAC_TI_1_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1)

(3)IF LV_EGY_RNG_IV_PLS_1_S[IDX_PRS_COR_CYL_CLC_S[m]] = 0
   AND IDX_TI_1_PRS_S_SP_TMP >= C_IDX_TI_MIN_L
(3)THEN – low injector needle lift with special calculation

   IDX_TI_TMP_PRS = MIN( (IDX_TI_1_PRS_S_SP_TMP – C_IDX_TI_MIN_L)
                                                  ; (NC_NR_IDX_TI_PRM_L – 1) )

   (3a)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_L – 1
   (3a)THEN

           TI_1_PRS_S_SP_TMP =
                   IDX_TI_1_PRS_S_SP_TMP * 0.001ms
                   + C_TI_OFS_L
                   + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]

   (3a)ELSE

           TI_1_PRS_S_SP_TMP = C_TI_L[IDX_TI_TMP_PRS]
                           + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]

   (3a)ENDIF

(3)ELSE – high injector needle lift or low lift identical to high lift

   IDX_TI_TMP_PRS =
           MIN( (IDX_TI_1_PRS_S_SP_TMP – C_IDX_TI_MIN_H) ;
                                                  (NC_NR_IDX_TI_PRM_H – 1) )
                                                  // limit calculation result to zero

   (3b)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_H – 1
   (3b)THEN

           TI_1_PRS_S_SP_TMP =
                   IDX_TI_1_PRS_S_SP_TMP * 0.001ms
                   + C_TI_OFS_H
                   + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]

   (3b)ELSE

           TI_1_PRS_S_SP_TMP = C_TI_H[IDX_TI_TMP_PRS]
                   + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]

   (3b)ENDIF
(3)ENDIF
```

A rough injection duration angle CRK_INJ_1_S is estimated from TI_1_PRS_S_SP_TMP, FAC_TI_1_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1) and engine speed:

From the manifold pressure MAP_INJ[m], the end of injection EOI_1_S[IDX_PRS_COR_CYL_CLC_S[m]] and the rough injection angle a mean counter pressure within the cylinder during the injection is calculated.

CRK_INJ_1_S = TI_1_PRS_S_SP_TMP * FAC_N
~~* FAC_TI_1_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1)~~

PRS_INJ_MV_1_S = MAP_INJ[m] * IP_PRS_INC_CMP_S
(EOI_1_S[IDX_PRS_COR_CYL_CLC_S[m]]; CRK_INJ_1_S)

The pressure difference at the injector is calculated from fuel rail pressure and the mean counter pressure.

If the engine is not running in stratified mode (LV_IGA_AND_INJ_SWI = 1), for a correct calculation of the pressure correction the FUP_H_SP_S_INJ[m] has to be used instead of the actual FUP_H_INJ[m], because the actual FUP_H_INJ[m] is tuned for a different combustion mode than stratified.

```
(4)IF LV_IGA_AND_INJ_SWI = 1
(4)THEN
    FUP_PRS_COR[m] = FUP_H_SP_S_INJ[m]
(4)ELSE
    FUP_PRS_COR[m] = FUP_H_INJ[m]
(4)ENDIF
PRS_DEC_INJ_1_S_TMP = FUP_PRS_COR[m] - PRS_INJ_MV_1_S
```

The total correction factor results from the map IP_FAC_TI_PRS_CYL.
For low injector needle lift, a separate map IP_FAC_TI_PRS_CYL_L is used.

```
    (4a)IF LV_EGY_RNG_IV_PLS_1_S[IDX_PRS_COR_CYL_CLC_S[m]] = 1
    (4a)THEN
        FAC_TI_1_PRS_CYL_S_TMP = IP_FAC_TI_PRS_CYL(PRS_INJ_MV_1_S)
    (4a)ELSE
        FAC_TI_1_PRS_CYL_S_TMP = IP_FAC_TI_PRS_CYL_L(PRS_INJ_MV_1_S)
    (4a)ENDIF

    (2)FOR x = 0 TO NC_CYL_NR – 1 DO
        (5)IF NC_IDX_CYL_HPP_REF[x] = m
        (5)THEN
            PRS_DEC_INJ_1_S[x] = PRS_DEC_INJ_1_S_TMP
            FAC_TI_1_PRS_CYL_S[x] = FAC_TI_1_PRS_CYL_S_TMP
        (5)ENDIF
    (2)ENDFOR
  (2)ENDIF
(1) ELSE
  (3)FOR x = 0 TO NC_CYL_NR – 1 DO
    (5)IF NC_IDX_CYL_HPP_REF[x] = m
    (5)THEN
        PRS_DEC_INJ_1_S[x] = FUP_H_SP_S_INJ[m]
    (5)ENDIF
  (3)ENDFOR
```

```
    (1) ENDIF
(1)ENDFOR
```

41.18 Injection Pressure Correction – Stratified Mode, 2. Pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TI_2_PRS_CYL_S[NC_CYL_NR]	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
factor for correction of fuel injection time due to an influence of cylinder counter pressure, strat. mode 2. pulse					
PRS_DEC_INJ_2_S[NC_CYL_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure decrease at the injector, stratified mode 2. pulse					
PRS_INJ_MV_2_S	V	0...FFFFH	0...347776	5.3067216	[hPa]
mean cylinder pressure during injection, stratified mode 2. pulse					
CRK_INJ_2_S	V	0...780H	0...720	0.375	[°CRK]
rough value for injection angle (EOI-SOI), stratified mode 1. pulse					
TI_2_STND_S[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Mapped Injection time, stratified mode, 2. pulse					
FAC_TI_2_PRS_CYL_S_TMP	-	0...FFFFH	0...15.99975	0.2441e-3	[-]
Factor for correction of fuel injection time due to an influence of cylinder counter pressure, strat. mode 2. pulse; temporary value					
PRS_DEC_INJ_2_S_TMP	-	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure decrease at the injector, stratified mode 2. pulse; temporary value					
TI_2_PRS_S_SP_TMP	-	0...FFFFH	0...65.535	0.001	[ms]
Injection time setpoint for calculation of injection pressure correction, stratified mode, 2. pulse; temporary value					
IDX_TI_2_PRS_S_SP_TMP	-	0...FFFFH	0...65535	1	[-]
Temporary counting factor index for pressure correction calculations; second pulse S.					

Input data:

C_IDX_TI_MIN_H	C_IDX_TI_MIN_L	C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H]	C_T_DCHA_DLY_L[NC_NR_IDX_TI_PRM_L]
C_TI_H[NC_NR_IDX_TI_PRM_H]	C_TI_L[NC_NR_IDX_TI_PRM_L]	C_TI_OFS_H	C_TI_OFS_L
EOI_2_S[NC_CYL_NR]	FAC_MFF_DIF_S[NC_CYL_NR]	FAC_MFF_TFU	FAC_N
FAC_TI_COR_IV_EGY_RNG_L[NC_CBK_HPP_NR]	FUP_H_SP_S_INJ[NC_CBK_HPP_NR]	FUP_PRS_COR[NC_CBK_HPP_NR]	IDX_PRS_COR_CYL_CLC_S[NC_CBK_HPP_NR]
INJ_MOD_SP_HOMS[NC_CYL_NR]	INJ_MOD_SP_S[NC_CYL_NR]	IP_FAC_TI_PRS_CYL	IP_FAC_TI_PRS_CYL_L
IP_MFF_IDX_TI_EGY_H	IP_MFF_IDX_TI_EGY_L	~~IP_MFF_TI_EGY_H~~	~~IP_MFF_TI_EGY_L~~
IP_PRS_INC_CMP_S	LF_PRS_COR_HPP_ENA	LV_EGY_RNG_IV_PLS_2_S[NC_CYL_NR]	LV_S_CLC
LV_ST_END	MAP_INJ[NC_CBK_HPP_NR]	MFF_SP_2_S[NC_CYL_NR]	NC_CBK_HPP_NR
NC_CYL_NR	NC_IDX_CYL_HPP_REF[NC_CYL_NR]	NC_INJ_MOD_MULTI	NC_NR_IDX_TI_PRM_H
NC_NR_IDX_TI_PRM_L			

FUNCTION DESCRIPTION:

General information:

The module calculates a correction factor to compensate the influence of the cylinder counter pressure and the influence of the fuel temperature on the injected fuel mass.

Note for Software: IDX_TI_TMP_PRS is a temporary calculation value with same resolution and range as IDX_TI_2_PRS_S_SP_TMP.

Application conditions:

Activation: every engine state

Deactivation: -

Initialisation: -

Update rate: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

Formula section:

```
(1)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
   (1)IF LV_S_CLC = 1 AND
         (INJ_MOD_SP_S[IDX_PRS_COR_CYL_CLC_S[m]] = "MPLS" OR
         INJ_MOD_SP_S[IDX_PRS_COR_CYL_CLC_S[m]] = "MPLS+PLS3" OR
         INJ_MOD_SP_HOMS[IDX_PRS_COR_CYL_CLC_S[m]] = "HOMS" OR
         INJ_MOD_SP_HOMS[IDX_PRS_COR_CYL_CLC_S[m]] = "HOMS+PLS3")
   (1)THEN
      Check, if fuel bank m has to be calculated:
      (2)IF Bit m of LF_PRS_COR_HPP_ENA = 1
      (2)THEN
         The injection time TI_2_PRS_S_SP_TMP is necessary to calculate CRK_INJ_2_S.
         Calculation of TI_2_PRS_S_SP_TMP:
         (3)IF LV_EGY_RNG_IV_PLS_2_S[IDX_PRS_COR_CYL_CLC_S[m]] = 1
         (3)THEN
            IDX_TI_2_PRS_S_SP_TMP = IP_MFF_IDX_TI_EGY_L(
                 (MFF_SP_2_S[IDX_PRS_COR_CYL_CLC_S[m]]*(1 +
                 FAC_MFF_DIF_S[IDX_PRS_COR_CYL_CLC_S[m]])*FAC_MFF_TFU);
                        PRS_DEC_INJ_2_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1))
         (3)ELSE
            IDX_TI_2_PRS_S_SP_TMP = IP_MFF_IDX_TI_EGY_L(
                    (MFF_SP_2_S[IDX_PRS_COR_CYL_CLC_S[m]]*(1 +
                    FAC_MFF_DIF_S[IDX_PRS_COR_CYL_CLC_S[m]])*FAC_MFF_TFU);
                        PRS_DEC_INJ_2_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1))
                    * FAC_TI_COR_IV_EGY_RNG_L[m]
         (3)ENDIF
         IDX_TI_2_PRS_S_SP_TMP = IDX_TI_2_PRS_S_SP_TMP
                    * FAC_TI_2_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1)
         (3)IF LV_EGY_RNG_IV_PLS_2_S[IDX_PRS_COR_CYL_CLC_S[m]] = 0
            AND IDX_TI_2_PRS_S_SP_TMP >= C_IDX_TI_MIN_L
```

```
(3)THEN – low injector needle lift with special calculation
    IDX_TI_TMP_PRS = MIN( (IDX_TI_2_PRS_S_SP_TMP – C_IDX_TI_MIN_L)
                                                    ; (NC_NR_IDX_TI_PRM_L – 1) )
    (3a)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_L – 1
    (3a)THEN
            TI_2_PRS_S_SP_TMP =
                    IDX_TI_2_PRS_S_SP_TMP * 0.001ms
                    + C_TI_OFS_L
                    + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]
    (3a)ELSE
            TI_2_PRS_S_SP_TMP = C_TI_L[IDX_TI_TMP_PRS]
                            + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]
    (3a)ENDIF
(3)ELSE – high injector needle lift or low lift identical to high lift
    IDX_TI_TMP_PRS =
            MIN( (IDX_TI_2_PRS_S_SP_TMP – C_IDX_TI_MIN_H) ;
                                                    (NC_NR_IDX_TI_PRM_H – 1) )
                                                    // limit calculation result to zero
    (3b)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_H – 1
    (3b)THEN
            TI_2_PRS_S_SP_TMP =
                    IDX_TI_2_PRS_S_SP_TMP * 0.001ms
                    + C_TI_OFS_H
                    + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]
    (3b)ELSE
            TI_2_PRS_S_SP_TMP = C_TI_H[IDX_TI_TMP_PRS]
                    + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]
    (3b)ENDIF
(3)ENDIF
```

A rough injection duration angle CRK_INJ_2_S is estimated from TI_2_PRS_S_SP_TMP, FAC_TI_2_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1) and engine speed:

From the manifold pressure MAP_INJ[m], the end of injection EOI_2_S[IDX_PRS_COR_CYL_CLC_S[m]] and the rough injection angle a mean counter pressure within the cylinder during the injection is calculated.

The pressure difference at the injector is calculated from fuel rail pressure and the mean counter pressure.

The total correction factor results from the map IP_FAC_TI_PRS_CYL.
For low injector needle lift, a separate map IP_FAC_TI_PRS_CYL_L is used.

CRK_INJ_2_S = TI_2_PRS_S_SP_TMP * FAC_N
~~* FAC_TI_2_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1)~~

```
            PRS_INJ_MV_2_S = MAP_INJ[m] * IP_PRS_INC_CMP_S
            (Input data for IP_PRS_INC_CMP_S is EOI_2_S[IDX_PRS_COR_CYL_CLC_S[m]]
            and CRK_INJ_2_S)

            PRS_DEC_INJ_2_S_TMP = FUP_PRS_COR[m] - PRS_INJ_MV_2_S
            (3a)IF LV_EGY_RNG_IV_PLS_2_S[IDX_PRS_COR_CYL_CLC_S[m]] = 1
            (3a)THEN
              FAC_TI_2_PRS_CYL_S_TMP = IP_FAC_TI_PRS_CYL(PRS_INJ_MV_2_S)
            (3a)ELSE
              FAC_TI_2_PRS_CYL_S_TMP = IP_FAC_TI_PRS_CYL_L(PRS_INJ_MV_2_S)
            (3a)ENDIF

            (2)FOR x = 0 TO NC_CYL_NR – 1 DO
              (4)IF NC_IDX_CYL_HPP_REF[x] = m
              (4)THEN
                   PRS_DEC_INJ_2_S[x] = PRS_DEC_INJ_2_S_TMP
                   FAC_TI_2_PRS_CYL_S[x] = FAC_TI_2_PRS_CYL_S_TMP
                   TI_2_STND_S[x] = TI_2_PRS_S_SP_TMP
              (4)ENDIF
            (2)ENDFOR
         (2)ENDIF
      (1)ELSE
         (3)FOR x = 0 TO NC_CYL_NR – 1 DO
            (5)IF NC_IDX_CYL_HPP_REF[x] = m
            (5)THEN
              PRS_DEC_INJ_2_S[x] = FUP_H_SP_S_INJ[m]
            (5)ENDIF
         (3)ENDFOR
      (1)ENDIF
   (1)ENDFOR
```

41.19 Injection Pressure Correction – Stratified Mode, 3. Pulse

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TI_3_PRS_CYL_S[NC_CYL_NR]	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
factor for correction of fuel injection time due to an influence of cylinder counter pressure, strat. mode 3. pulse					
PRS_DEC_INJ_3_S[NC_CYL_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure decrease at the injector, stratified mode 3. pulse					
PRS_INJ_MV_3_S	V	0...FFFFH	0...347776	5.3067216	[hPa]
Cylinder pressure (mean value) during injection, stratified mode 3. pulse					
CRK_INJ_3_S	V	0...780H	0...720	0.375	[°CRK]
Rough value for injection angle (SOI-EOI), stratified mode 3. pulse					
TI_3_STND_S[NC_CYL_NR]	O/V	0...FFFFH	0...65.535	0.001	[ms]
Mapped Injection time, stratified mode, 3. pulse					
FAC_TI_3_PRS_CYL_S_TMP	-	0...FFFFH	0...15.99975	0.2441e-3	[-]
factor for correction of fuel injection time due to an influence of cylinder counter pressure, strat. mode 3. pulse; temporary value					
PRS_DEC_INJ_3_S_TMP	-	0...FFFFH	0...347776	5.3067216	[hPa]
Pressure decrease at the injector, stratified mode 3. pulse; temporary value					
TI_3_PRS_S_SP_TMP	-	0...FFFFH	0...65.535	0.001	[ms]
Injection time setpoint for calculation of injection pressure correction, stratified mode, 3. pulse; temporary value					
IDX_TI_3_PRS_S_SP_TMP	-	0...FFFFH	0...65535	1	[-]
Temporary counting factor index for pressure correction calculations; third pulse S.					

Input data:

C_IDX_TI_MIN_H	C_IDX_TI_MIN_L	C_T_DCHA_DLY_H[NC_NR_IDX_TI_PRM_H]	C_T_DCHA_DLY_L[NC_NR_IDX_TI_PRM_L]
C_TI_H[NC_NR_IDX_TI_PRM_H]	C_TI_L[NC_NR_IDX_TI_PRM_L]	C_TI_OFS_H	C_TI_OFS_L
EOI_3_S[NC_CYL_NR]	FAC_MFF_DIF_S[NC_CYL_NR]	FAC_MFF_TFU	FAC_N
FAC_TI_COR_IV_EGY_RNG_L[NC_CBK_HPP_NR]	FUP_H_SP_S_INJ[NC_CBK_HPP_NR]	FUP_PRS_COR[NC_CBK_HPP_NR]	IDX_PRS_COR_CYL_CLC_S[NC_CBK_HPP_NR]
INJ_MOD_SP_HOMS[NC_CYL_NR]	INJ_MOD_SP_S[NC_CYL_NR]	IP_FAC_TI_PRS_CYL	IP_FAC_TI_PRS_CYL_L
IP_MFF_IDX_TI_EGY_H	IP_MFF_IDX_TI_EGY_L	~~IP_MFF_TI_EGY_H~~	~~IP_MFF_TI_EGY_L~~
IP_PRS_INC_CMP_S	LF_PRS_COR_HPP_ENA	LV_EGY_RNG_IV_PLS_3_S[NC_CYL_NR]	LV_S_CLC
LV_ST_END	MAP_INJ[NC_CBK_HPP_NR]	MFF_SP_3_S[NC_CYL_NR]	NC_CBK_HPP_NR
NC_CYL_NR	NC_IDX_CYL_HPP_REF[NC_CYL_NR]	NC_NR_IDX_TI_PRM_H	NC_NR_IDX_TI_PRM_L

FUNCTION DESCRIPTION:

General information:

The module calculates a correction factor to compensate the influence of the cylinder counter pressure and the influence of the fuel temperature on the injected fuel mass.

Note for Software: IDX_TI_TMP_PRS is a temporary calculation value with same resolution and range as IDX_TI_3_PRS_S_SP_TMP.

Application conditions:

Activation: every engine state

Deactivation: -

Initialisation: -

Update rate: LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

Formula section:

```
(1)FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
  (1)IF (LV_S_CLC = 1) AND
       (INJ_MOD_SP_S[IDX_PRS_COR_CYL_CLC_S[m]] = "MPLS+PLS3" OR
       INJ_MOD_SP_HOMS[IDX_PRS_COR_CYL_CLC_S[m]] = "HOMS+PLS3")
  (1)THEN
    Check, if fuel bank m has to be calculated:
    (2)IF Bit m of LF_PRS_COR_HPP_ENA = 1
    (2)THEN
      The injection time TI_3_PRS_S_SP_TMP is necessary to calculate CRK_INJ_3_S.
      Calculation of TI_3_PRS_S_SP_TMP:
      (3)IF LV_EGY_RNG_IV_PLS_3_S[IDX_PRS_COR_CYL_CLC_S[m]] = 1
      (3)THEN
        IDX_TI_3_PRS_S_SP_TMP = IP_MFF_IDX_TI_EGY_H(
            (MFF_SP_3_S[IDX_PRS_COR_CYL_CLC_S[m]]*(1 +
            FAC_MFF_DIF_S[IDX_PRS_COR_CYL_CLC_S[m]])*FAC_MFF_TFU);
                   PRS_DEC_INJ_3_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1))
      (3)ELSE
        IDX_TI_3_PRS_S_SP_TMP = IP_MFF_IDX_TI_EGY_L(
              (MFF_SP_3_S[IDX_PRS_COR_CYL_CLC_S[m]]*(1 +
              FAC_MFF_DIF_S[IDX_PRS_COR_CYL_CLC_S[m]])*FAC_MFF_TFU);
                   PRS_DEC_INJ_3_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1))
              * FAC_TI_COR_IV_EGY_RNG_L[m]
      (3)ENDIF

      IDX_TI_3_PRS_S_SP_TMP = IDX_TI_3_PRS_S_SP_TMP
              * FAC_TI_3_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1)
      (3)IF LV_EGY_RNG_IV_PLS_3_S[IDX_PRS_COR_CYL_CLC_S[m]] = 0
        AND IDX_TI_3_PRS_S_SP_TMP >= C_IDX_TI_MIN_L
```

```
(3)THEN – low injector needle lift with special calculation
    IDX_TI_TMP_PRS = MIN( (IDX_TI_3_PRS_S_SP_TMP – C_IDX_TI_MIN_L)
                                                ; (NC_NR_IDX_TI_PRM_L – 1) )
    (3a)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_L – 1
    (3a)THEN
            TI_3_PRS_S_SP_TMP =
                    IDX_TI_3_PRS_S_SP_TMP * 0.001ms
                    + C_TI_OFS_L
                    + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]
    (3a)ELSE
            TI_3_PRS_S_SP_TMP = C_TI_L[IDX_TI_TMP_PRS]
                            + C_T_DCHA_DLY_L[IDX_TI_TMP_PRS]
    (3a)ENDIF
(3)ELSE – high injector needle lift or low lift identical to high lift
    IDX_TI_TMP_PRS =
            MIN( (IDX_TI_3_PRS_S_SP_TMP – C_IDX_TI_MIN_H) ;
                                                (NC_NR_IDX_TI_PRM_H – 1) )
                                                // limit calculation result to zero
    (3b)IF IDX_TI_TMP_PRS >= NC_NR_IDX_TI_PRM_H – 1
    (3b)THEN
            TI_3_PRS_S_SP_TMP =
                    IDX_TI_3_PRS_S_SP_TMP * 0.001ms
                    + C_TI_OFS_H
                    + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]
    (3b)ELSE
            TI_3_PRS_S_SP_TMP = C_TI_H[IDX_TI_TMP_PRS]
                    + C_T_DCHA_DLY_H[IDX_TI_TMP_PRS]
    (3b)ENDIF
(3)ENDIF
```

A rough injection duration angle CRK_INJ_3_S is estimated from TI_3_PRS_S_SP_TMP, FAC_TI_3_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1) and engine speed:
Note: TI_3_PRS_S_SP_TMP is calculated with the pressure correction information of the previous calculated injector pressure difference PRS_DEC_INJ_3_S[IDX_PRS_COR_CYL_CLC_S[m]].

From the manifold pressure MAP_INJ[m], the end of injection EOI_3_S[IDX_PRS_COR_CYL_CLC_S[m]] and the rough injection angle a mean counter pressure within the cylinder during the injection is calculated.

The pressure difference at the injector is calculated from fuel rail pressure and the mean counter pressure.

The total correction factor results from the map IP_FAC_TI_PRS_CYL.
For low injector needle lift, a separate map IP_FAC_TI_PRS_CYL_L is used.

```
            CRK_INJ_3_S = TI_3_PRS_S_SP_TMP * FAC_N
                          ~~* FAC_TI_3_PRS_CYL_S[IDX_PRS_COR_CYL_CLC_S[m]](n-1)~~

            PRS_INJ_MV_3_S = MAP_INJ[m] * IP_PRS_INC_CMP_S
            (Input data for IP_PRS_INC_CMP_S is EOI_3_S[IDX_PRS_COR_CYL_CLC_S[m]]
            and CRK_INJ_3_S)


            PRS_DEC_INJ_3_S_TMP = FUP_PRS_COR[m] - PRS_INJ_MV_3_S

            (3a)IF LV_EGY_RNG_IV_PLS_3_S[IDX_PRS_COR_CYL_CLC_S[m]] = 1
            (3a)THEN
                FAC_TI_3_PRS_CYL_S_TMP = IP_FAC_TI_PRS_CYL (PRS_INJ_MV_3_S)
            (3a)ELSE
                FAC_TI_3_PRS_CYL_S_TMP = IP_FAC_TI_PRS_CYL_L(PRS_INJ_MV_3_S)
            (3a)ENDIF


            (2)FOR x = 0 TO NC_CYL_NR – 1 DO
                (4)IF NC_IDX_CYL_HPP_REF[x] = m
                (4)THEN
                        PRS_DEC_INJ_3_S[x] = PRS_DEC_INJ_3_S_TMP
                        FAC_TI_3_PRS_CYL_S[x] = FAC_TI_3_PRS_CYL_S_TMP
                        TI_3_STND_S[x] = TI_3_PRS_S_SP_TMP
                (4)ENDIF
            (2)ENDFOR
        (2)ENDIF

    (1)ELSE
        (3)FOR x = 0 TO NC_CYL_NR – 1 DO
            (5)IF NC_IDX_CYL_HPP_REF[x] = m
            (5)THEN
                PRS_DEC_INJ_3_S[x] = FUP_H_SP_S_INJ[m]
            (5)ENDIF
        (3)ENDFOR
    (1)ENDIF
(1)ENDFOR
```

41.20 Fuel temperature correction

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_MFF_TFU	O/V	0...1FFFH	0...1.99975	0.2441e-3	[-]
Factor to correct fuel mass due to the fuel temperature in rail.					
FAC_TI_TFU	O/V	0...1FFFH	0...1.99975586	2.44141E-4	-
factor for correction of fuel injection time due to a change of fuel temperature variatio					

Input data:

TFU_INJ

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_MFF_TEMP_COR	12	0...1FFFH	0...1.99975	0.2441e-3	[-]
LDP_TFU_IP_FAC_MFF_TEMP_COR	12	0...FEH	-48...142.5	0.75	[°C]
factor for correction of fuel injection due to the fuel temperature					

41.20.1 General Information

Density fluctuations of fuel due to temperature variation can be compensated with a correction factor on the fuel mass setpoint.

Application Condition

Following initilisation needs to be done at reset:

FAC_TI_TFU = 1;

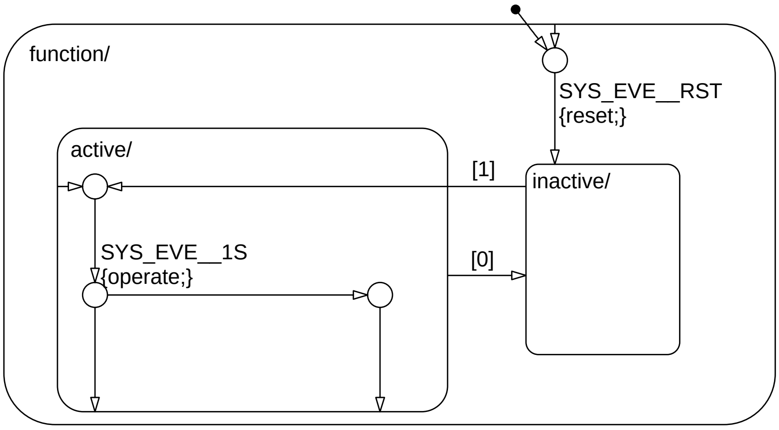


Figure 1 INJR_ISPCLTFU0/ APP_CDN/ Chart

Function Description

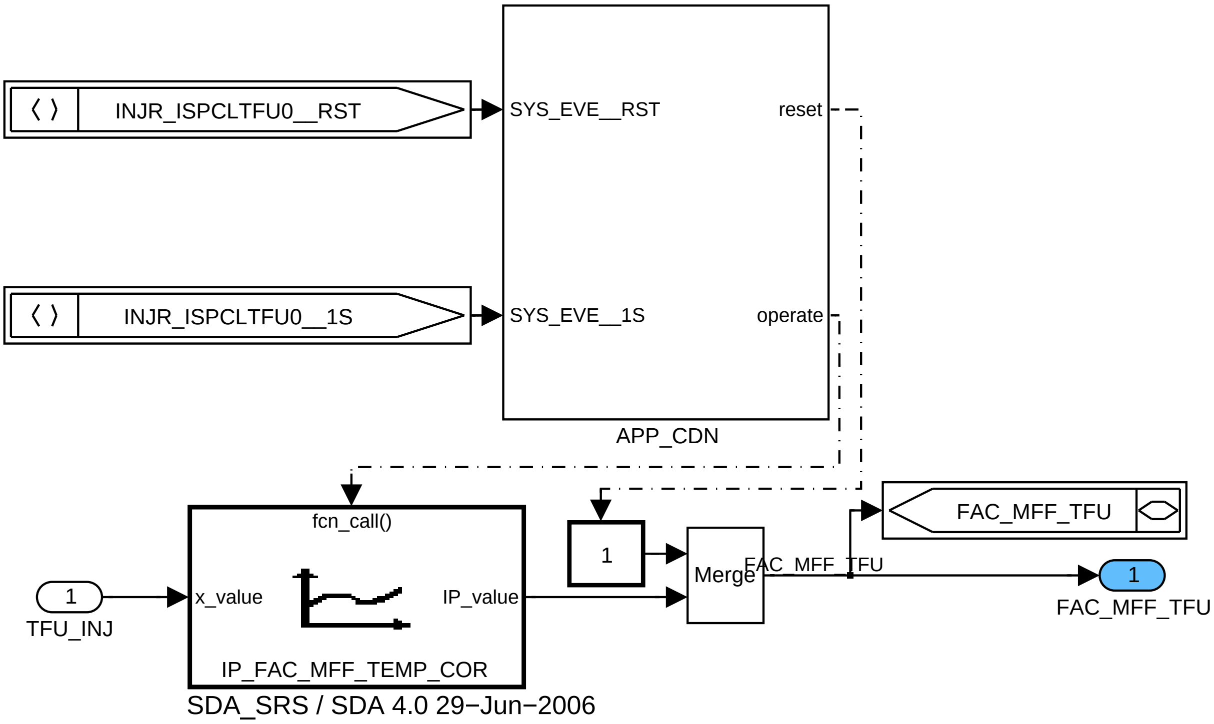


Figure 2 INJR_ISPCLTFU0

41.21 Compensation of temperature influence on the injector needle lift

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EGY_ADD_TEMP	O/V	8000...7FFFH	-127.501946 ... 127.498054	0.00389105	mJ
Additive piezo energy correction depending on the injector temperature					

Input data:

TFU_INJ	TEMP_CAPA_IV[NC_CYL_NR]	NC_CYL_NR	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TEMP_CAPA_IV_ENA	1	0...1H	0...1	1	-
Switch to enable useage of piezo stack temperature					
IP_EGY_ADD_TEMP	8	0...FFFFH	-127.501946 ... 127.498054	0.00389105	mJ
LDP_TFU_INJ_IP_EGY_ADD_TEMP	8	0...FEH	-48...142.5	0.75	°C
Additive piezo energy correction depending on the injector temperature					

41.21.1 General information

The needle lift of injection valves with piezo actuators depends on the injector's temperature. In order to compensate this effect and to keep the needle lift constant an additive correction of the piezo stack energy has to be applied.

Application Condition

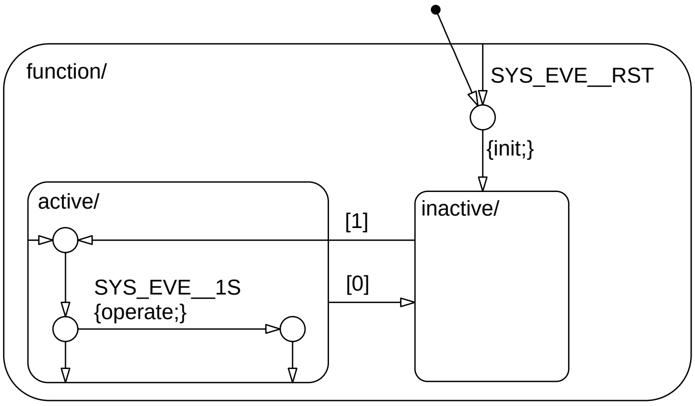


Figure 3 INJR_ISPCLTECOR0/ APP_CDN/ Chart

Function Description

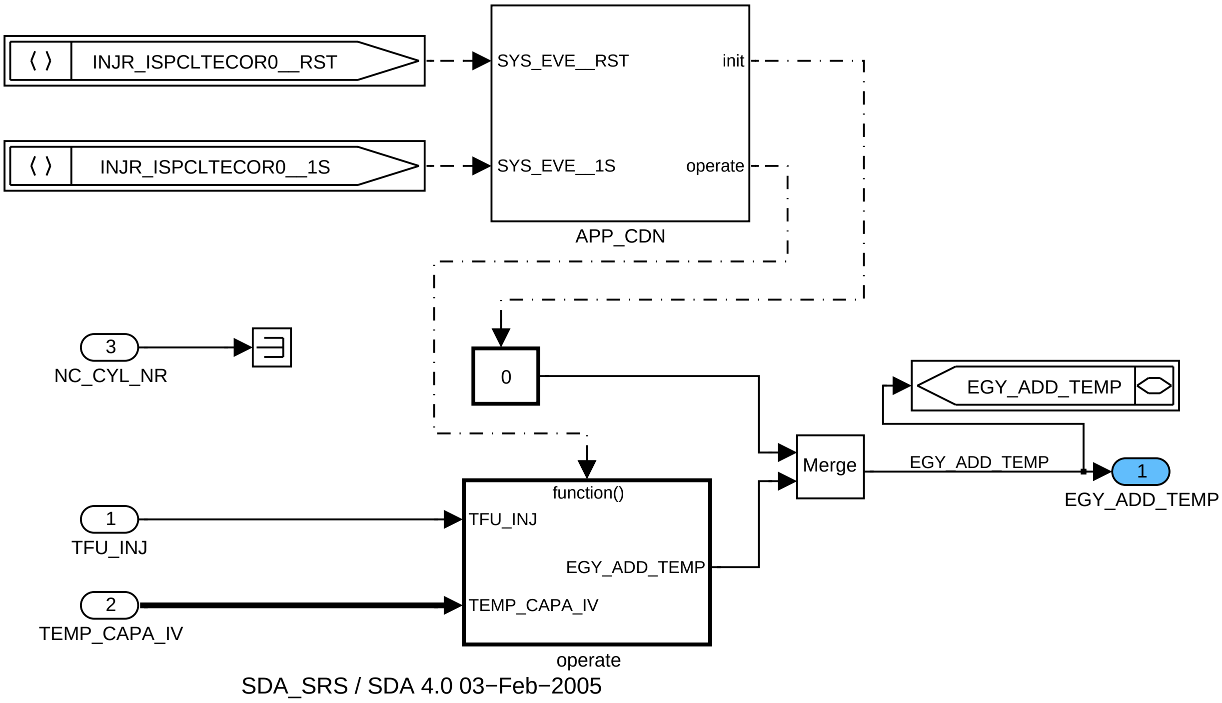


Figure 4 INJR_ISPCLTECOR0

41.21.1.1 SUBFUNCTION: operate

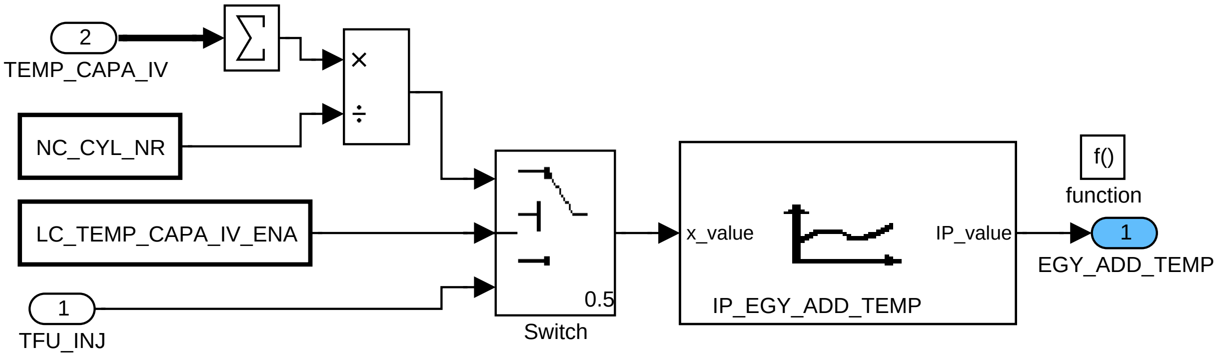


Figure 5 INJR_ISPCLTECOR0/ operate

41.22 Compensation of fuel pressure influence on the injector needle lift

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EGY_ADD_FUP[NC_CBK_HPP_NR]	O/V	8000...7FFFH	-127.501946 ... 127.498054	0.00389105	mJ
Additive piezo energy correction depending on the fuel pressure					

Input data:

LV_ST_END	FUP_H_INJ[NC_CBK_HPP_NR]	NC_CBK_HPP_NR	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_EGY_ADD_FUP	12	0...FFFFH	-127.501946 ... 127.498054	0.00389105	mJ
LDP_FUP_IP_EGY_ADD_FUP	12	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Additive piezo energy correction depending on the fuel pressure					

41.22.1 General information

The needle lift of injection valves with piezo actuators depends on the fuel pressure. In order to compensate this effect and to keep the needle lift constant an additive correction of the piezo stack energy has to be applied.

Application Condition

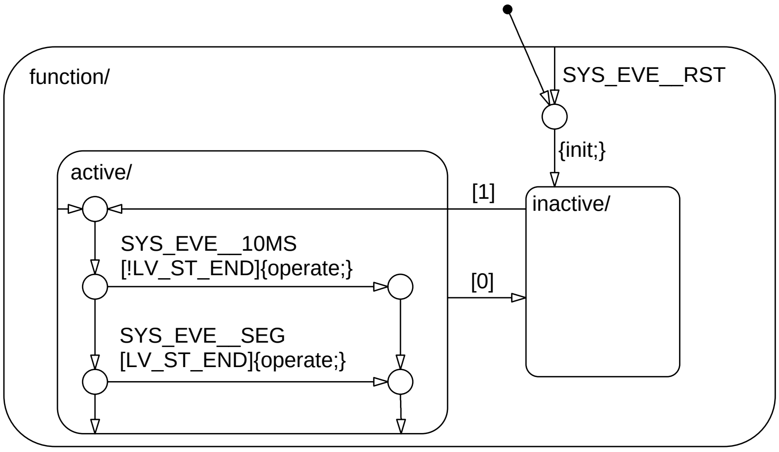


Figure 6 INJR_ISPCLFUCOR0/ APP_CDN/ Chart

Function Description

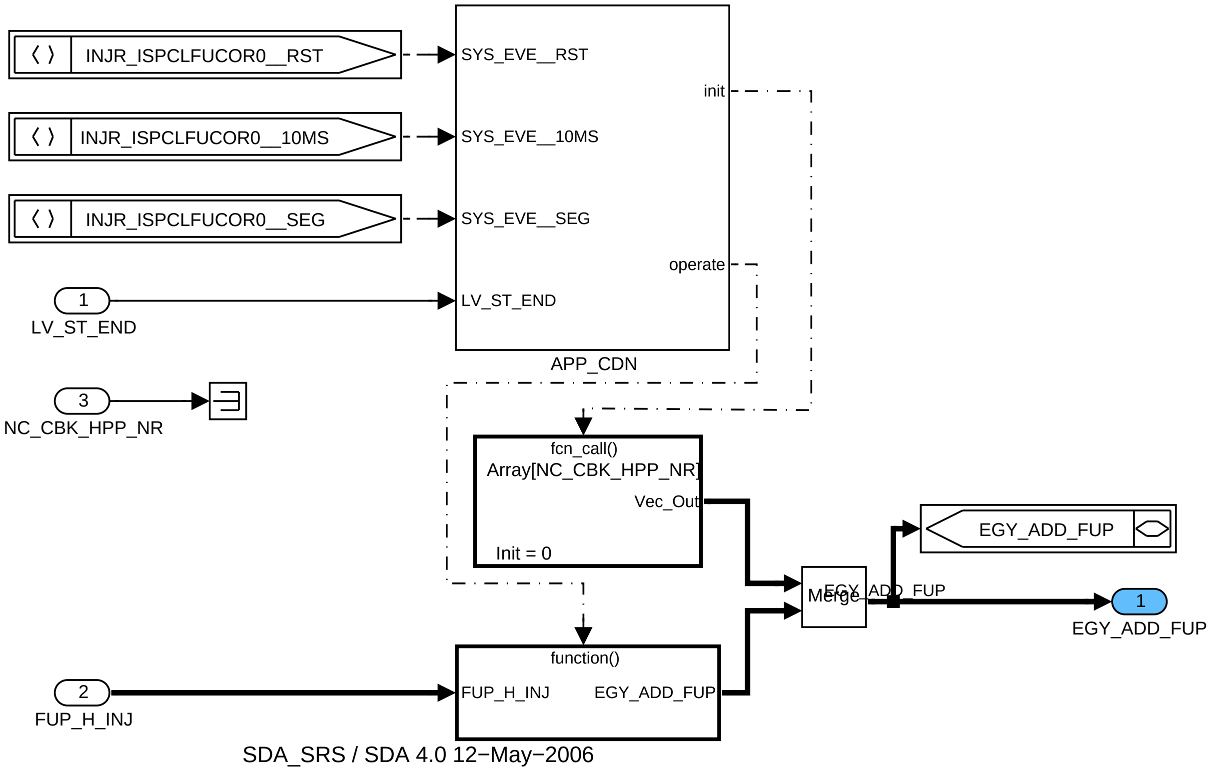


Figure 7 INJR_ISPCLFUCOR0

41.22.1.1 SUBFUNCTION: Subsystem

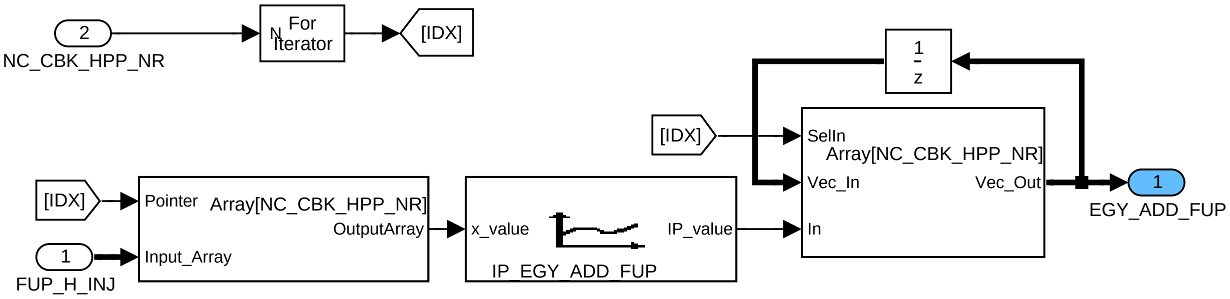


Figure 8 INJR_ISPCLFUCOR0/ Subsystem/ For_Iterator

41.23 Compensation of the injection duty cycle on the injector needle lift

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EGY_ADD_DUCY	O/V	8000...7FFFH	-127.501946 ... 127.498054	0.00389105	mJ
Additive piezo energy correction depending on the injection duty cycle					
DUCY_TI_N	V	0...4000H	0...100	0.00610352	%
Injection duty cycle					

Input data:

LV_ST_END	TI_1_MES[NC_CYL_NR]	TI_2_MES[NC_CYL_NR]	TI_POST_MES[NC_CYL_NR]
FAC_N	N_32		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_EGY_ADD_DUCY_ENA	1	0...1H	0...1	1	-
Flag to activate energy correction accounting for injection duty cycle					
IP_EGY_ADD_DUCY	7x7	0...FFFFH	-127.501946 ... 127.498054	0.00389105	mJ
LDP_DUCY_IP_EGY_ADD_DUCY	7	0...4000H	0...100	0.00610352	%
LDP_N_IP_EGY_ADD_DUCY	7	0...FFH	0...8.16E+3	32	rpm
Additive piezo energy correction depending on the injection duty cycle					

41.23.1 General information

The needle lift of injection valves with piezo actuators depends on the injection duty cycle. In order to compensate this effect and to keep the needle lift constant an additive correction of the piezo stack energy has to be applied.

Application Condition

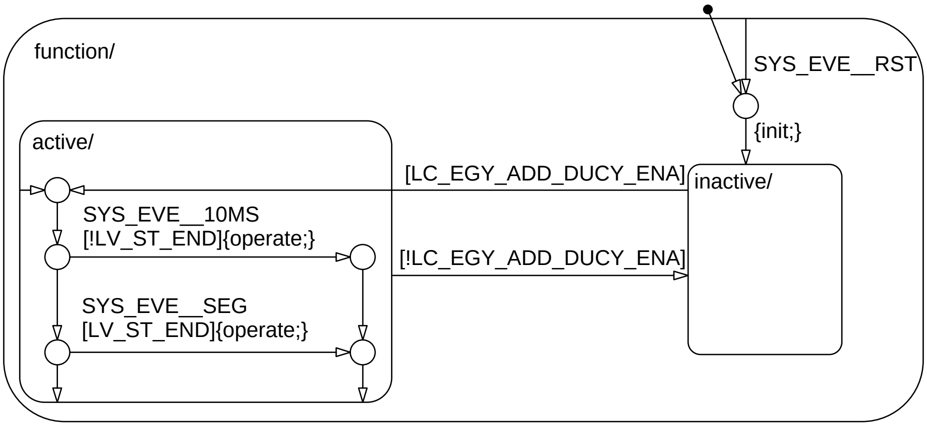


Figure 9 INJR_ISPCLDUCY0/ APP_CDN/ Chart

Function Description

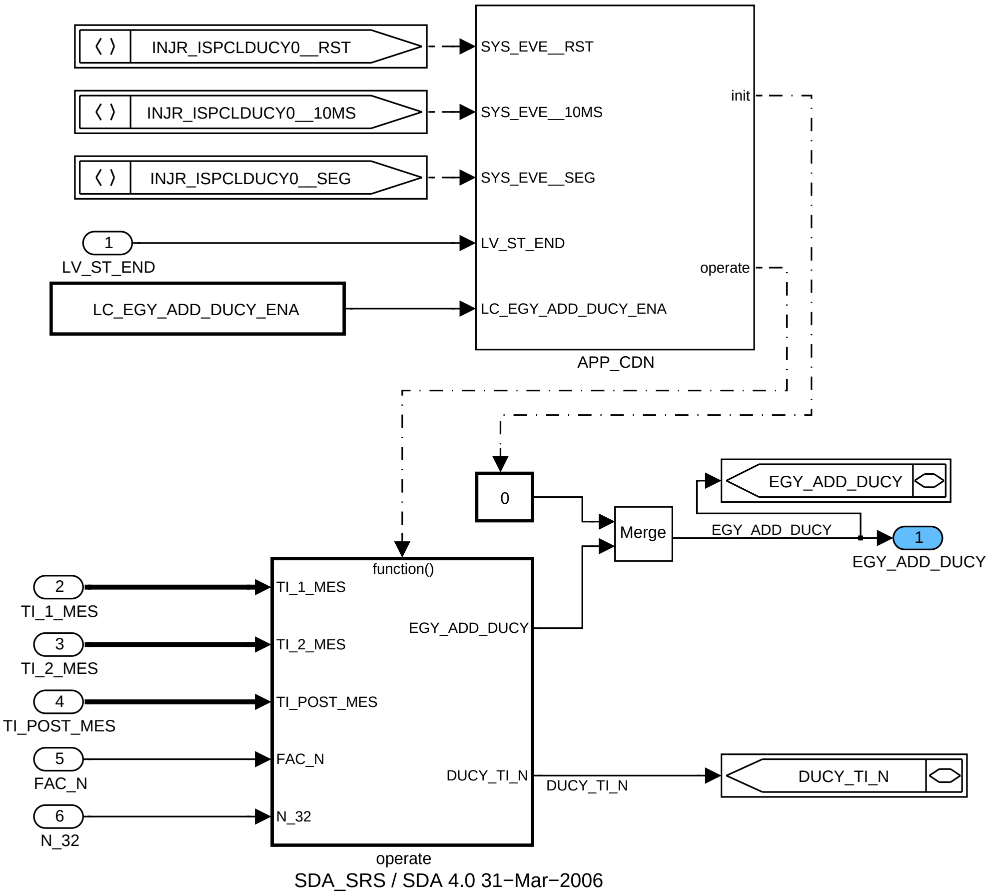


Figure 10 INJR_ISPCLDUCY0

41.23.1.1 SUBFUNCTION: operate

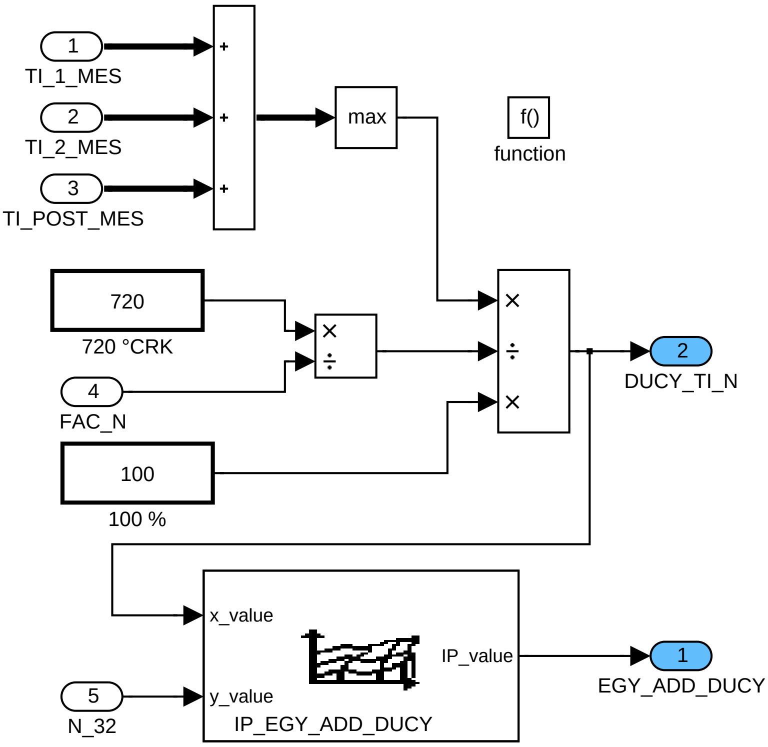


Figure 11 INJR_ISPCLDUCY0/ operate

41.24 Injection time correction by the application system

Overview

For development, the application tool can perform a cylinder specific injection time correction. Additionally – for service tool tests – an exhaust bank dependent TI correction can be applied also.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TI_TUN_IV[NC_CYL_NR]	V/O	0...FFFFH	0...15.99975	2.44E-04	[-]
Cylinder selective Injection Time Correction by the Application System - multiplicative value					
TI_TUN_ADD_IV[NC_CYL_NR]	V/O	8000...7FFFH	-131.072...131.068	0.004	[ms]
Cylinder selective Injection Time Correction by the Application System - additive value					
TI_AS	V/O	0...FFFFH	0...15.99975	2.44E-04	[-]
Global Injection Time Correction by the Application System - multiplicative					
#if (NC_USE_TI_AS_CBK = 1)					
TI_AS_CBK_EX[NC_CBK_EX_NR]	V	0...FFFFH	0...15.99975	2.44E-04	[-]
Exhaust bank selective injection time correction caused by service tool intervention - multiplicative value					
TI_ADD_AS_CBK_EX[NC_CBK_EX_NR]	V	8000...7FFFH	-131.072...131.068	0.004	[ms]
Exhaust bank selective injection time correction caused by service tool intervention - additive value					
#endif					

Input data:

LV_FAC_TI_EXT_ADJ	FAC_TI_EXT_ADJ	NC_CYL_NR	NC_CBK_EX_NR
NC_USE_TI_AS_CBK			

Calibration data:

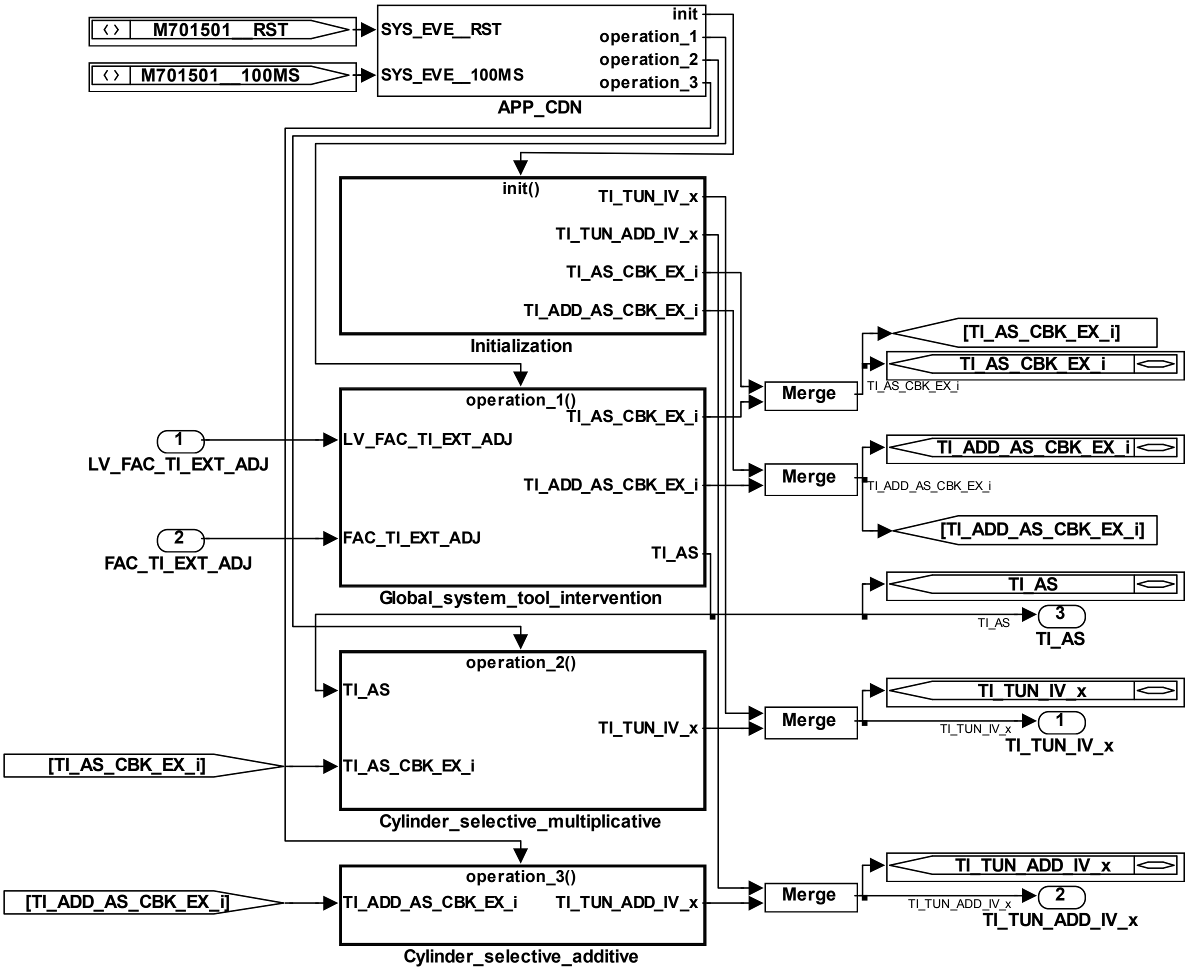
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TI_AS_GLOBAL	1	0...FFFFH	0...15.99975	2.44E-04	[-]
Global injection time correction					
C_TI_AS[NC_CYL_NR]	1	0...FFFFH	0...15.99975	2.44E-04	[-]
Injection time factor (cylinder selective) to correct cylinder injection value					
C_TI_ADD_AS_GLOBAL	1	8000...7FFFH	-131.072...131.068	0.004	[ms]
Global Injection Time Correction - additive value					
C_TI_ADD_AS[NC_CYL_NR]	1	8000...7FFFH	-131.072...131.068	0.004	[ms]
Cylinder selective Injection Time Correction - additive value					
#if (NC_USE_TI_AS_CBK = 1)					
LC_TI_AS_CBK_UPD_DIS	1	0...1H	0...1	1	[-]
Switch to disable TI_AS_CBK_EX and TI_ADD_AS_CBK_EX data update, for customer delivery it has to be 0					
C_TI_AS_CBK_EX[NC_CBK_EX_NR]	1	0...FFFFH	0...15.99975	2.44E-04	[-]
Injection time factor (exhaust bank selective) to correct cylinder injection value					
C_TI_ADD_AS_CBK_EX[NC_CBK_EX_NR]	1	8000...7FFFH	-131.072...131.068	0.004	[ms]
Exhaust bank selective Injection Time Correction - additive value					
#endif					

FUNCTION DESCRIPTION:

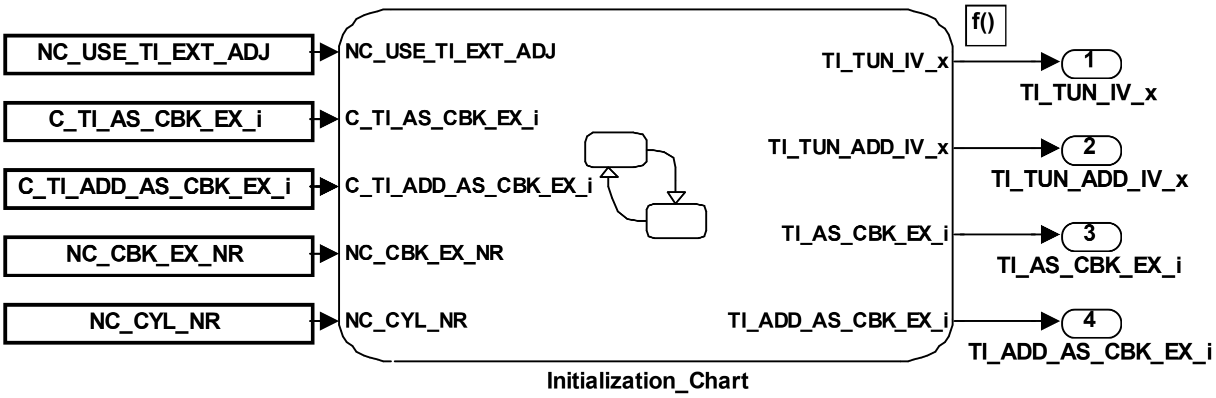
Application condition:

Activation:	every engine state
Deactivation:	-
Initialization:	at reset
Recurrence:	100ms

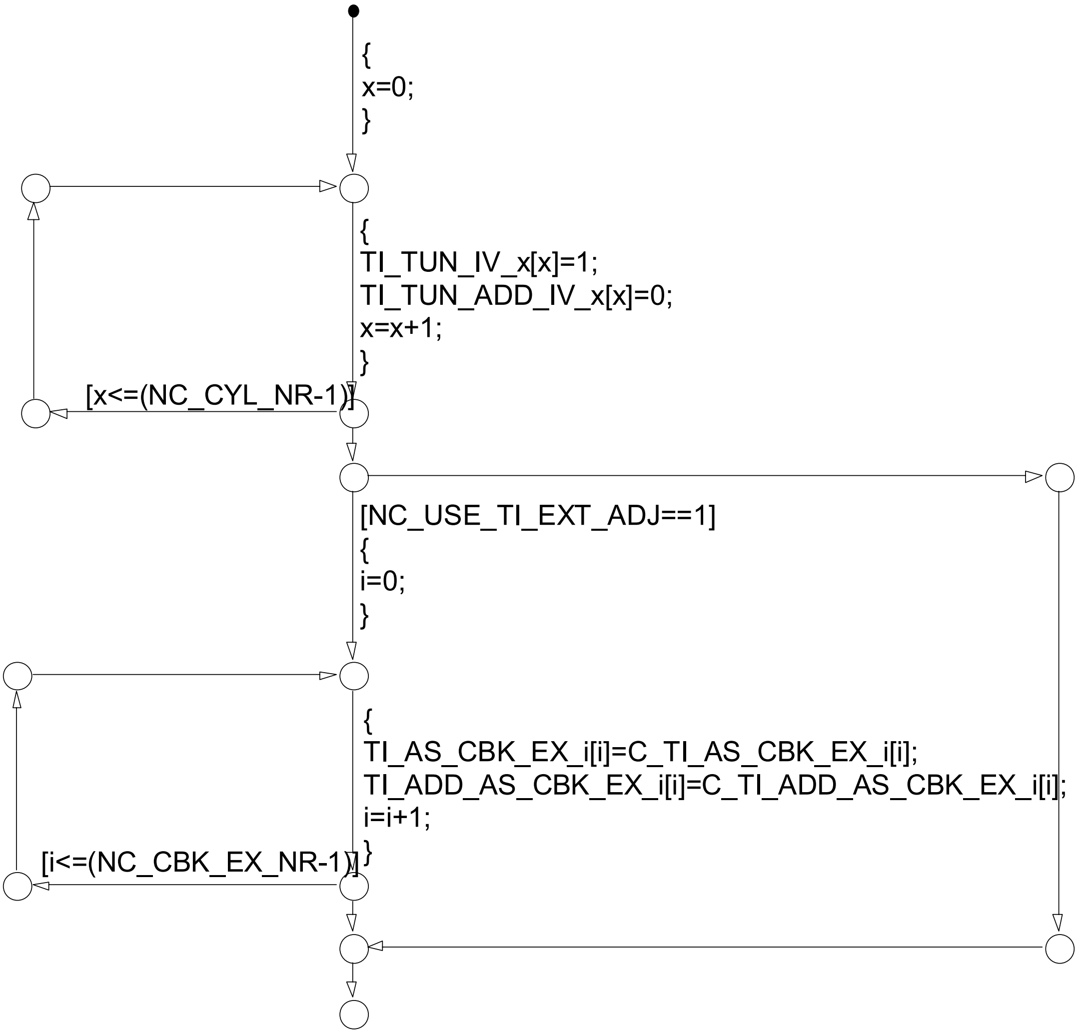
Main System:



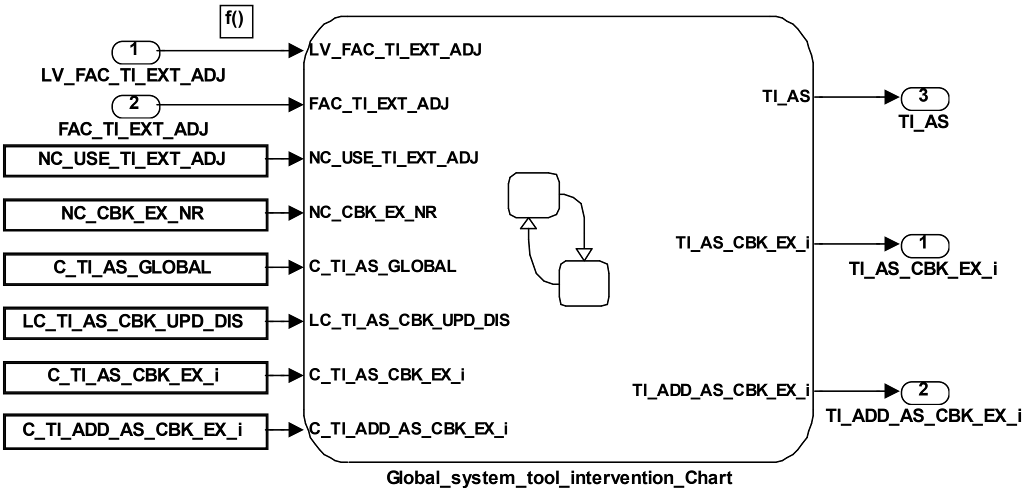
Initialization:



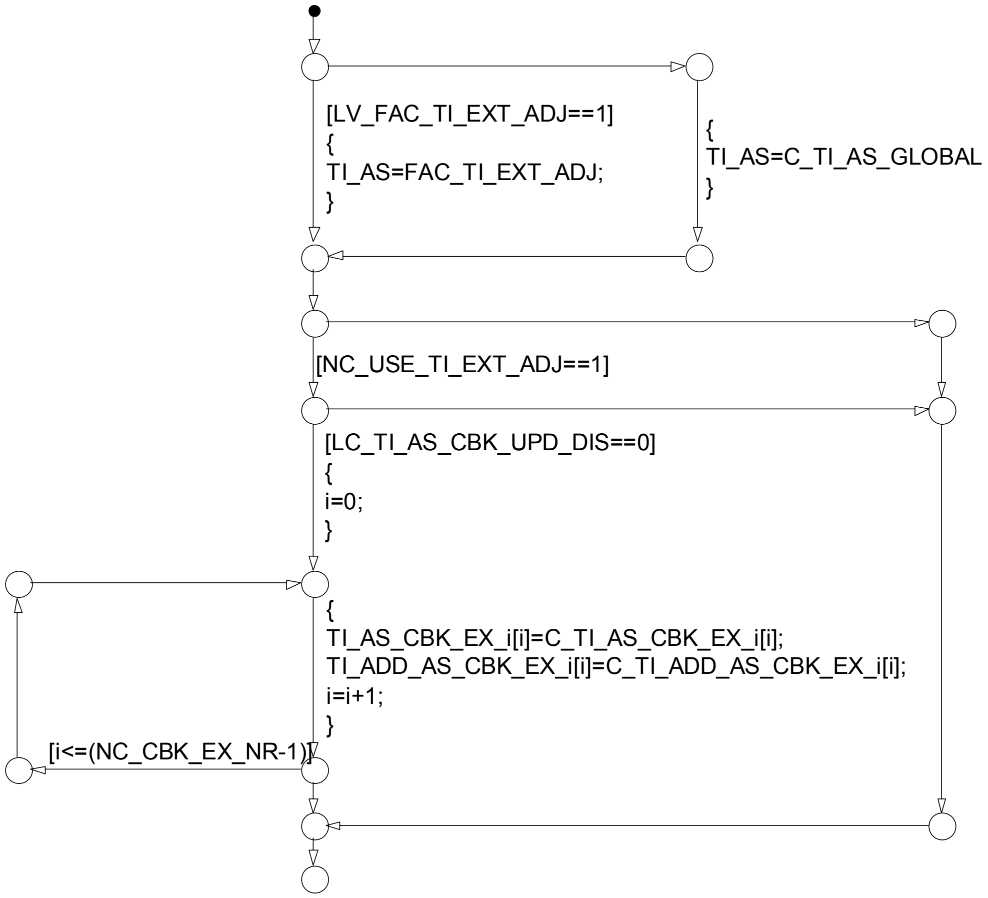
Initialization_Chart:


```
{
x=0;
}

{
TI_TUN_IV_x[x]=1;
TI_TUN_ADD_IV_x[x]=0;
x=x+1;
}

[x<=(NC_CYL_NR-1)]

[NC_USE_TI_EXT_ADJ==1]
{
i=0;
}

{
TI_AS_CBK_EX_i[i]=C_TI_AS_CBK_EX_i[i];
TI_ADD_AS_CBK_EX_i[i]=C_TI_ADD_AS_CBK_EX_i[i];
i=i+1;
}

[i<=(NC_CBK_EX_NR-1)]
```

Global_system_tool_intervention:

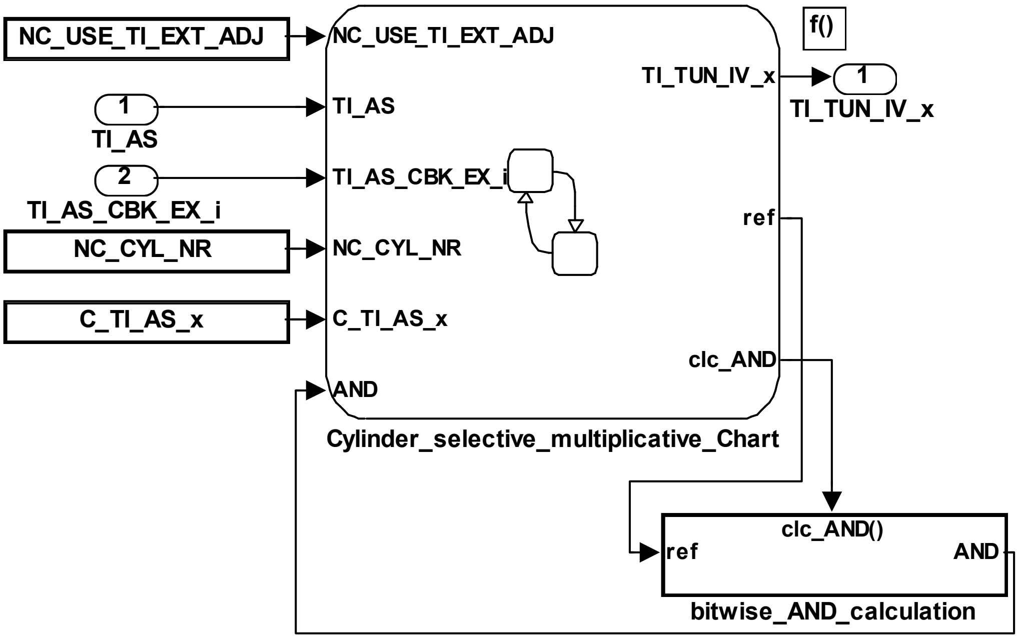


Global_system_tool_intervention_Chart:

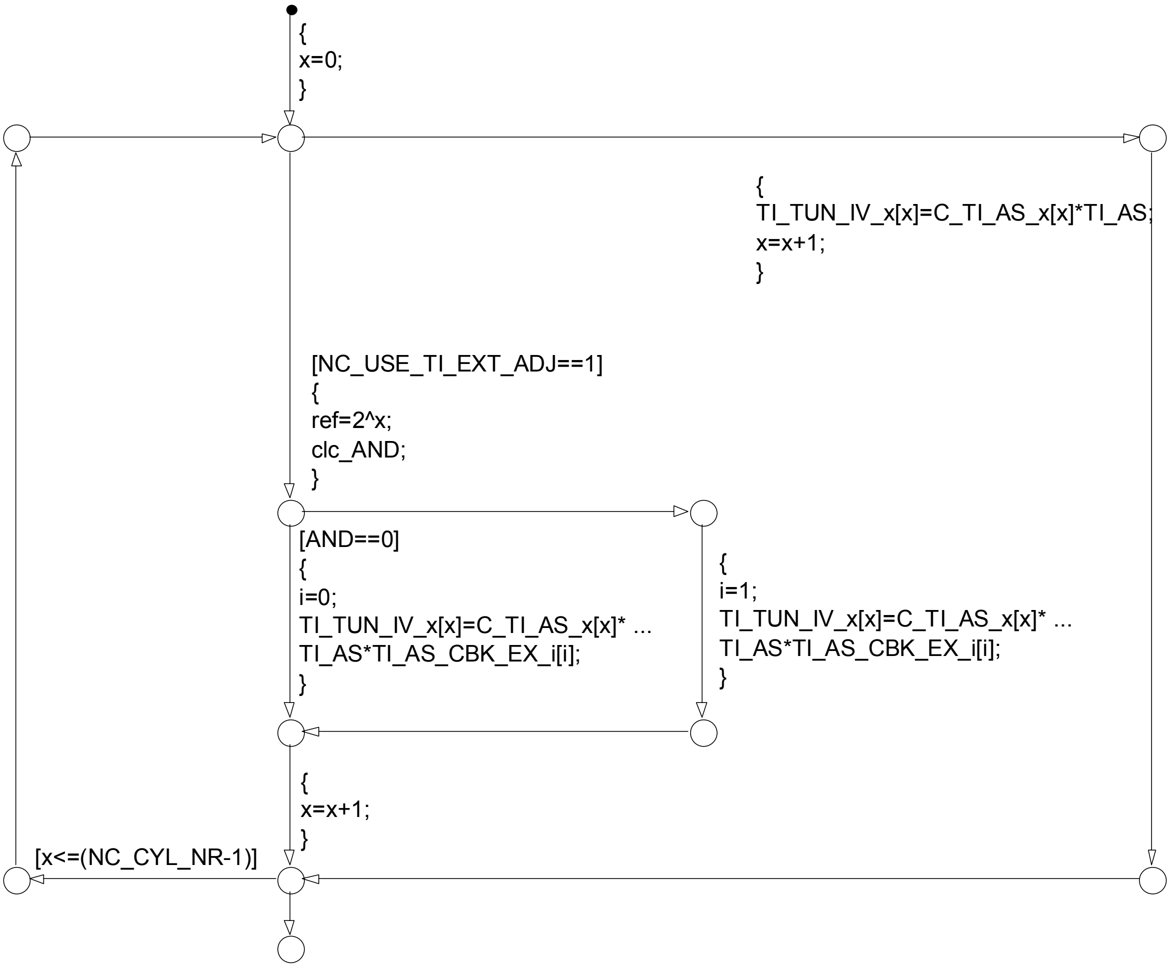


```
[LV_FAC_TI_EXT_ADJ==1]
{
TI_AS=FAC_TI_EXT_ADJ;
}

{
TI_AS=C_TI_AS_GLOBAL
}

[NC_USE_TI_EXT_ADJ==1]

[LC_TI_AS_CBK_UPD_DIS==0]
{
i=0;
}

{
TI_AS_CBK_EX_i[i]=C_TI_AS_CBK_EX_i[i];
TI_ADD_AS_CBK_EX_i[i]=C_TI_ADD_AS_CBK_EX_i[i];
i=i+1;
}

[i<=(NC_CBK_EX_NR-1)]
```

Cylinder_selective_multiplative:

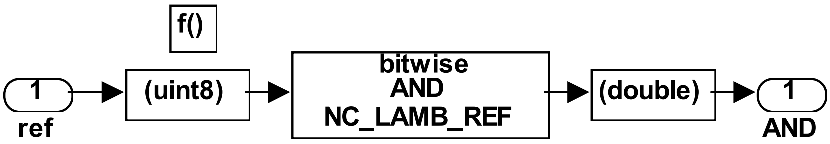


Cylinder_selective_multiplative_Chart:

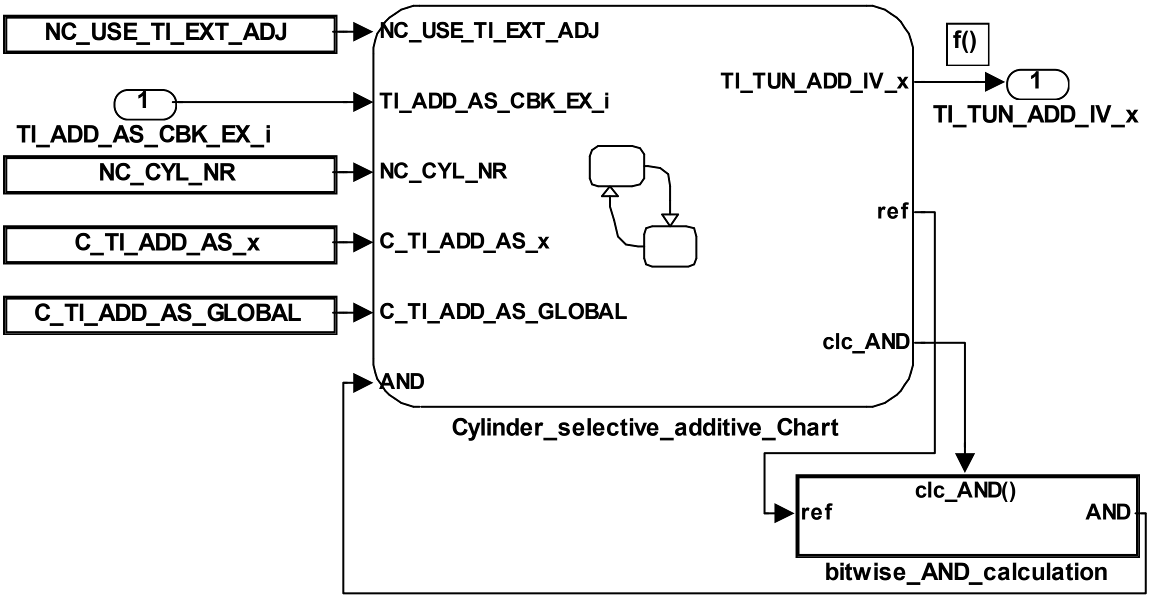


```
{
x=0;
}

{
TI_TUN_IV_x[x]=C_TI_AS_x[x]*TI_AS;
x=x+1;
}

[NC_USE_TI_EXT_ADJ==1]
{
ref=2^x;
clc_AND;
}

[AND==0]
{
i=0;
TI_TUN_IV_x[x]=C_TI_AS_x[x]* ...
TI_AS*TI_AS_CBK_EX_i[i];
}

{
i=1;
TI_TUN_IV_x[x]=C_TI_AS_x[x]* ...
TI_AS*TI_AS_CBK_EX_i[i];
}

{
x=x+1;
}

[x<=(NC_CYL_NR-1)]
```

bitwise_AND_calculation:



Cylinder_selective_additive:



Cylinder_selective_additive_Chart:

```
{
x=0;
}
```

```
[NC_USE_TI_EXT_ADJ==1]
{
ref=2^x;
clc_AND;
}
```

```
{
TI_TUN_ADD_IV_x[x]=C_TI_ADD_AS_x[x]+C_TI_ADD_AS_GLOBAL;
x=x+1;
}
```

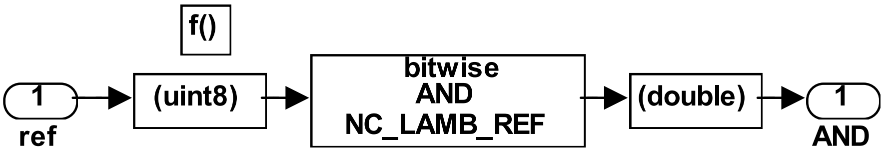
```
[AND==0]
{
i=0;
TI_TUN_ADD_IV_x[x]=C_TI_ADD_AS_x[x] ...
+C_TI_ADD_AS_GLOBAL ...
+TI_ADD_AS_CBK_EX_i[i];
}
```

```
{
i=1;
TI_TUN_ADD_IV_x[x]=C_TI_ADD_AS_x[x] ...
+C_TI_ADD_AS_GLOBAL ...
+TI_ADD_AS_CBK_EX_i[i];
}
```

```
{
x=x+1;
}
```

```
[x<=(NC_CYL_NR-1)]
```

bitwise_AND_calculation:



Formula section:

41.24.1 Global application system/service tool intervention

```
(1) IF      LV_FAC_TI_EXT_ADJ = 1          ... Service tool intervention active
(1) THEN    TI_AS = FAC_TI_EXT_ADJ
(1) ELSE    TI_AS = C_TI_AS_GLOBAL          ... Application system intervention active
(1) ENDIF
#if (NC_USE_TI_AS_CBK = 1)
(2) IF      LC_TI_AS_CBK_UPD_DIS = 0       ... If switch is not set, update adaptation factors.
                                            If not, a data update from service tool is necessary
(2) THEN
    (3) FOR i = 1 TO NC_CBK_EX_NR DO:
        TI_AS_CBK_EX[i] = C_TI_AS_CBK_EX[i]
        TI_ADD_AS_CBK_EX[i] = C_TI_ADD_AS_CBK_EX[i]
    (3) ENDFOR
(2) ENDIF
#endif
```

41.24.2 Cylinder selective injection time correction - multiplicative

```
(4) FOR x = 0 TO (NC_CYL_NR-1) DO:
   #if (NC_USE_TI_AS_CBK = 1)
         i = {exhaust bank number which corresponds to cylinder x }
         TI_TUN_IV[x] = C_TI_AS[x]        ... Cylinder specific application system intervention
                    * TI_AS               ... Global application system/service tool
                    * TI_AS_CBK_EX[i]     ... Used and updated by service tool
   #else
         TI_TUN_IV[x] = C_TI_AS[x]        ... Cylinder specific application system intervention
                    * TI_AS               ... Global application system/service tool
   #endif
(4) ENDFOR
```

41.24.3 Cylinder selective injection time correction – additive

```
(5) FOR x = 0 TO (NC_CYL_NR-1) DO:
   #if (NC_USE_TI_AS_CBK = 1)
         i = {exhaust bank number which corresponds to cylinder x }
         TI_TUN_ADD_IV[x] = C_TI_ADD_AS[x]    ... Cylinder specific application system intervention
                    + C_TI_ADD_AS_GLOBAL      ... Global application system
                    + TI_ADD_AS_CBK_EX[i]     ... Used and updated by service tool
   #else
         TI_TUN_ADD_IV[x] = C_TI_ADD_AS[x]    ... Cylinder specific application system intervention
                    + C_TI_ADD_AS_GLOBAL      ... Global application system
   #endif
(5) ENDFOR
```

41.25 Piezo capacity temperature determination

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CAPA_IV_DIF_NOM[NC_CYL_NR]	O/V/S	8000...7FFFH	-13.9802133 ... 13.9797867	4.26642E-4	µF
Capacity offset of the piezo injector to the nominal value					
LV_TEMP_CAPA_IV_EOL	O/V/S	0...1H	0...1	1	-
Flag indicating valid capacatiy-temperature adaptation at end-of-line					
TEMP_CAPA_IV[NC_CYL_NR]	O/V	0...FEH	-48...142.5	0.75	°C
Temperature of the piezo stack derived from its capacity					
CAPA_IV_1_FIL[NC_CYL_NR]	O/V	0...FFFFH	0...27.96	4.26642E-4	µF
Capacity of the piezo injector for the first pulse, filtered value					
LV_T_ES_CAPA_TEMP_VLD	O/V	0...1H	0...1	1	-
Engine stop time is valid for calculation of capacity offset					
TEMP_CAPA_IV_MV	O/V/S	0...FEH	-48...142.5	0.75	°C
Temperature of the piezo stacks derived from their capacities, mean value					
TEMP_CAPA_IV_SUB	V	0...FEH	-48...142.5	0.75	°C
Temperature of the piezo stacks, substitutional value					
LV_TEMP_CAPA_IV_CLC_ENA[NC_CYL_NR]	V	0...1H	0...1	1	-
Cylinder individual flag to enable calculation of piezo temperature out of its capacity value					

Input data:

LV_ST_END	SEG_CTR	T_ES	LV_IV_MES_VLD
CAPA_IV_1_CLC[NC_CYL_NR]	NR_CYL_INJ_BAS	NC_CYL_NR	TCO
FUP_H_INJ[NC_CBK_HPP_NR]	LV_T_ES_NOT_PLAUS	LV_EGY_RNG_IV_PLS_1_ACK[NC_CYL_NR]	TI_1_MES[NC_CYL_NR]
EGY_IV_1_CLC[NC_CYL_NR]	LV_CH	LV_SYN_ENG	NC_CBK_HPP_NR
NC_IDX_CYL_HPP_REF[NC_CYL_NR]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CAPA_IV_DIF_NOM_LIM	1	0...FFFFH	0...27.96	4.26642E-4	µF
Limit of the capacity offset of the piezo injector to the nominal value					
C_FAC_FIL_CAPA_IV	1	0...FFH	0...0.99609375	0.00390625	-
PT1-Filter constant for piezo capacity					
C_SEG_CTR_CAPA_TEMP	1	0...FFFFH	0...6.5535E+4	1	-
Number of segments for capacity offset calculation					
C_TEMP_CAPA_INC	1	0...2000H	0...1.5	1.83105E-4	°C
Temperature increment					
C_TEMP_CAPA_IV_EOL	1	0...1H	0...1	1	-
Flag for capacatiy-temperature adaptation at end-of-line, manual setting					
C_TI_MIN_CAPA_FIL	1	0...FFFFH	0...65.535	0.001	ms
Minimum injection time limit for calculation of filtered capacity					
C_T_ES_CAPA_TEMP	1	0...FFFFH	0...6.5535E+4	1	min
Minimum engine of duration for capacity offset calculation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_CAPA_IV_DIF_NOM_INH_CH_ENA	1	0...1H	0...1	1	-
Logical constant to enable inhibition of capacatiy offset determination for catalyst heating					
LC_TEMP_CAPA_IV_CLC_ENA	1	0...1H	0...1	1	-
Flag to enable calculation of piezo temperature					
LC_TEMP_CAPA_IV_EOL_MAN	1	0...1H	0...1	1	-
Logical constant to enable manual setting of end-of-line bit for capacity temperature adaptation					
IP_CAPA_COR_EGY_IV	6	0...FFFFH	-13.9802133 ... 13.9797867	4.26642E-4	µF
LDP_EGY_IP_CAPA_COR_EGY_IV	6	0...FFFFH	0...255	0.00389105	mJ
Capacity correction depending on the piezo injector´s energy					
IP_FAC_T_ES_TEMP_CAPA_IV_2	8	0...FFH	0...1.9921875	0.0078125	-
LDPM_T_ES_1_INJR	8	0...FFFFH	0...6.5535E+4	1	min
2. factor for piezo stack temperature substitution accounting for engine off time					
IP_CAPA_IV_NOM	8x6	0...FFFFH	0...27.96	4.26642E-4	µF
LDP_TCO_IP_CAPA_IV_NOM	8	0...FEH	-48...142.5	0.75	°C
LDPM_FUP_2_INJR	6	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Nominal capacity of the piezo injector depending on temperature and fuel pressure					
IP_TEMP_CAPA_IV_NOM	8x6	0...FEH	-48...142.5	0.75	°C
LDP_CAPA_IP_TEMP_CAPA_IV_NOM	8	0...FFFFH	0...27.96	4.26642E-4	µF
LDPM_FUP_2_INJR	6	0...FFFFH	0...3.47776E+5	5.3067216	hPa
Nominal temperature-capacity relationship of the piezo injector depending on fuel pressure					

41.25.1 General information

The large signal capacity of the piezo stack rises with temperature. This dependency can be used to determine the piezo temperature out of its capacity. Since the capacity is also influenced by forces acting on the piezo stack the fuel pressure has to be taken into account. This dependency is described by the map IP_TEMP_CAPA_IV_NOM supplying a temperature for a certain nominal (mean) capacity and fuel pressure level. The map IP_CAPA_IV_NOM describes the inverse relationship supplying a capacity for a given temperature and fuel pressure. The look-up curve IP_CAPA_COR_EGY_IV accounts for the dependency of the capacity on the energy stored in the piezo stack and provides an additive correction for the capacity value.

The capacity value deviates from one stack to another due to manufacturing variations, aging effects etc.. Therefore the offset of each injector to the nominal capacity has to be determined at a reference temperature.(Fig. 1)

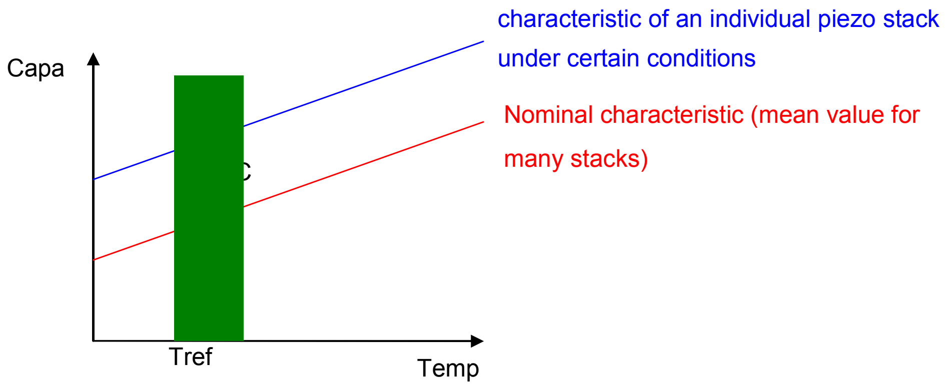
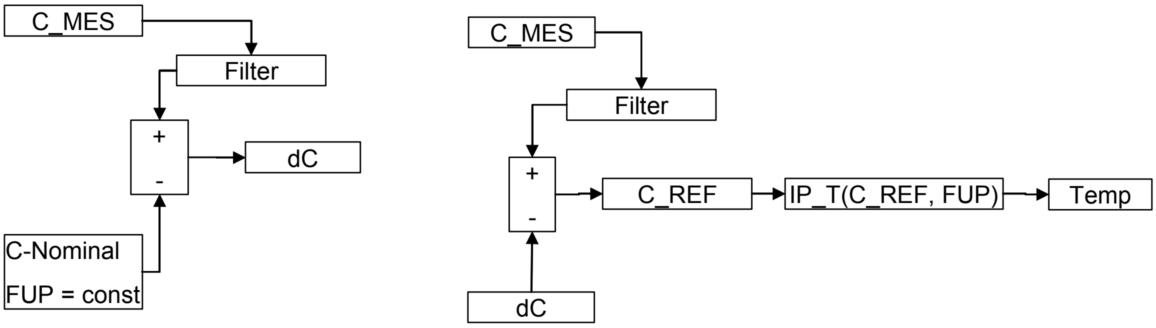


Fig. 1: capacity-temperature relationship for piezo stacks

The capacity offset of each injector is stored in the nonvolatile memory (NVMY). Once the capacity offset is available the individual capacity can be shifted to the nominal characteristic allowing for a determination of the piezo temperature (Fig. 2).



Determination of capacity offset at T_{ref} Determination of piezo stack temperature

Fig. 2: Functional overview

Application Condition

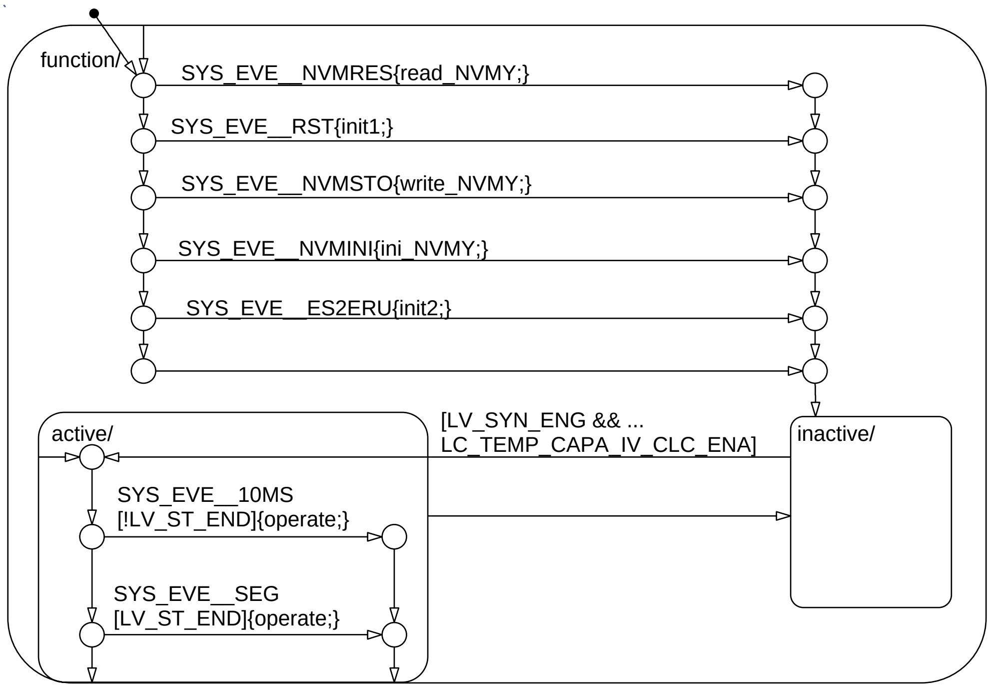
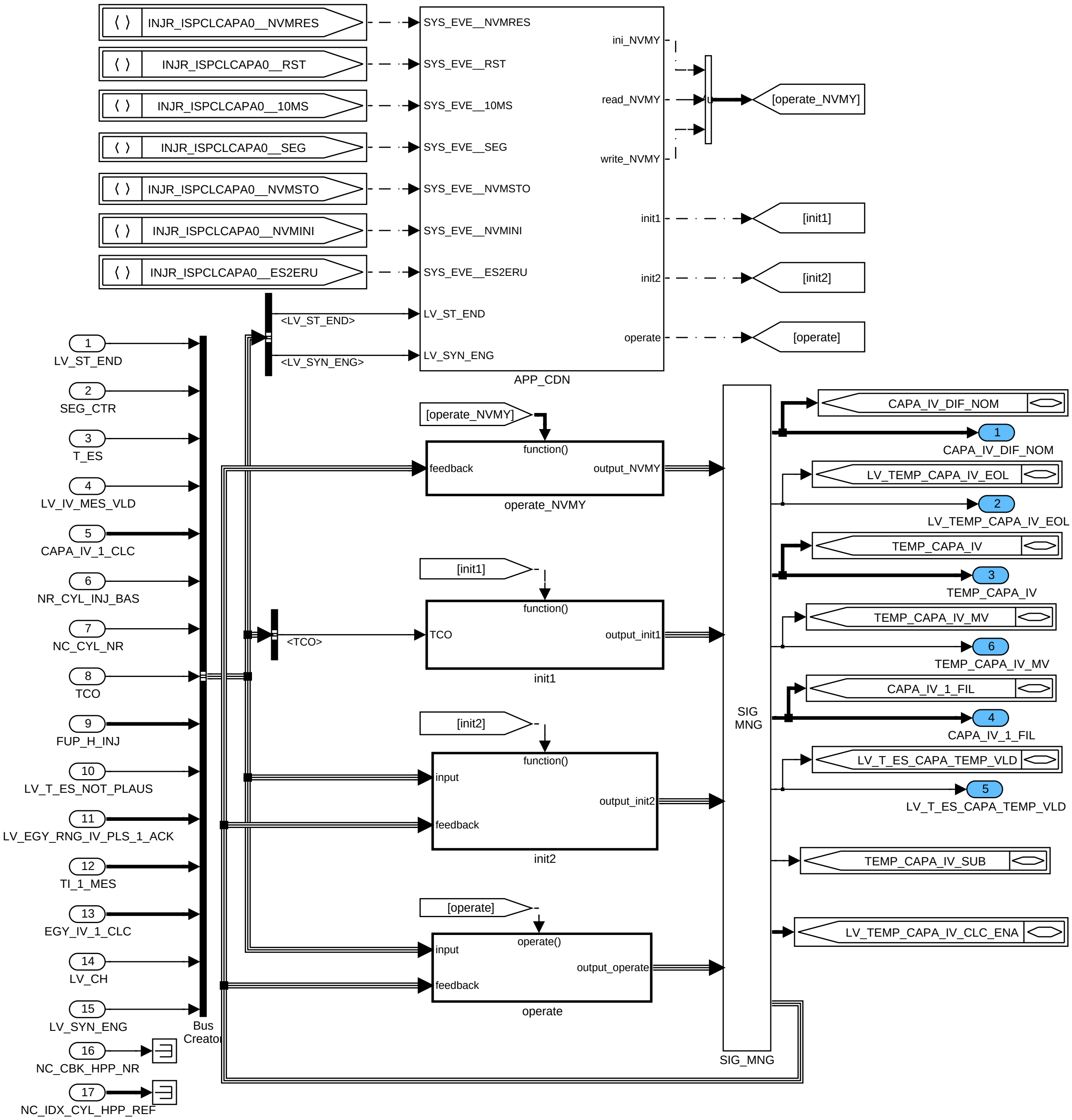


Figure 12 INJR_ISPCLCAPA0/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 24−Oct−2006

Figure 13 INJR_ISPCLCAPA0

41.25.1.1 Handling of nonvolatile memory

The capacity offset of each injector is stored. It is read and written to the NVMY before/after each complete driving cycle.

Initialization of nonvolatile memory

Reading of nonvolatile memory

Writing of nonvolatile memory

41.25.1.2 Initialization

The piezo stack temperature is initialized with the coolant temperature at ECU-reset.

41.25.1.3 Plausibility check of engine off time and calculation of substitute value for TEMP_CAPA_IV

A new adaptation value of the capacity offset is calculated if the engine offtime is plausible and greater than a calibratable constant. If both conditions are true the bit LV_T_ES_CAPA_TEMP_VLD is set. For the first engine start at end-of-line the capacity offset adaptation is always done. At the first engine start LV_TEMP_CAPA_IV_EOL is zero. It is set to one if the capacity offset is adapted and will be stored in the nonvolatile memory. Furthermore an substitute value for the piezo-capacity temperature is determined via engine off-time, coolant-temperature and the mean value of the piezo-temperatures of the last driving cycle.

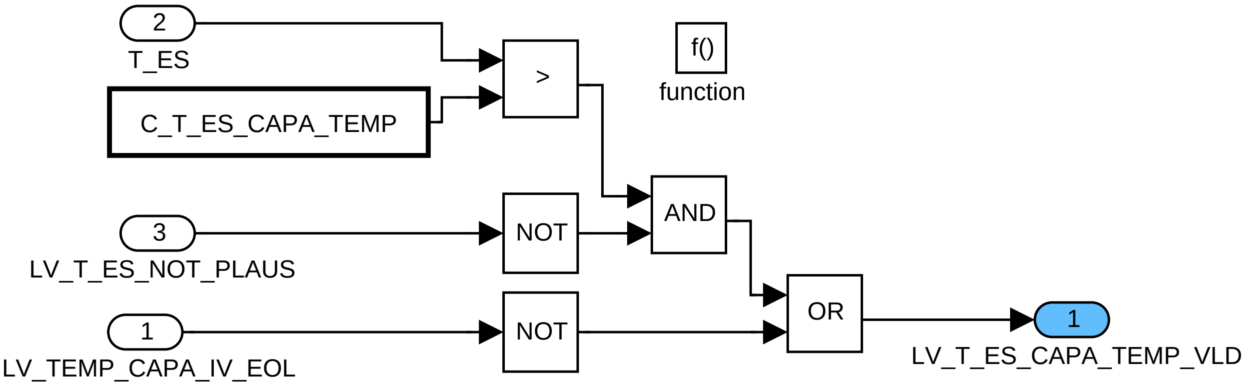


Figure 14 INJR_ISPCLCAPA0/ init2/ Plausiblity_Check

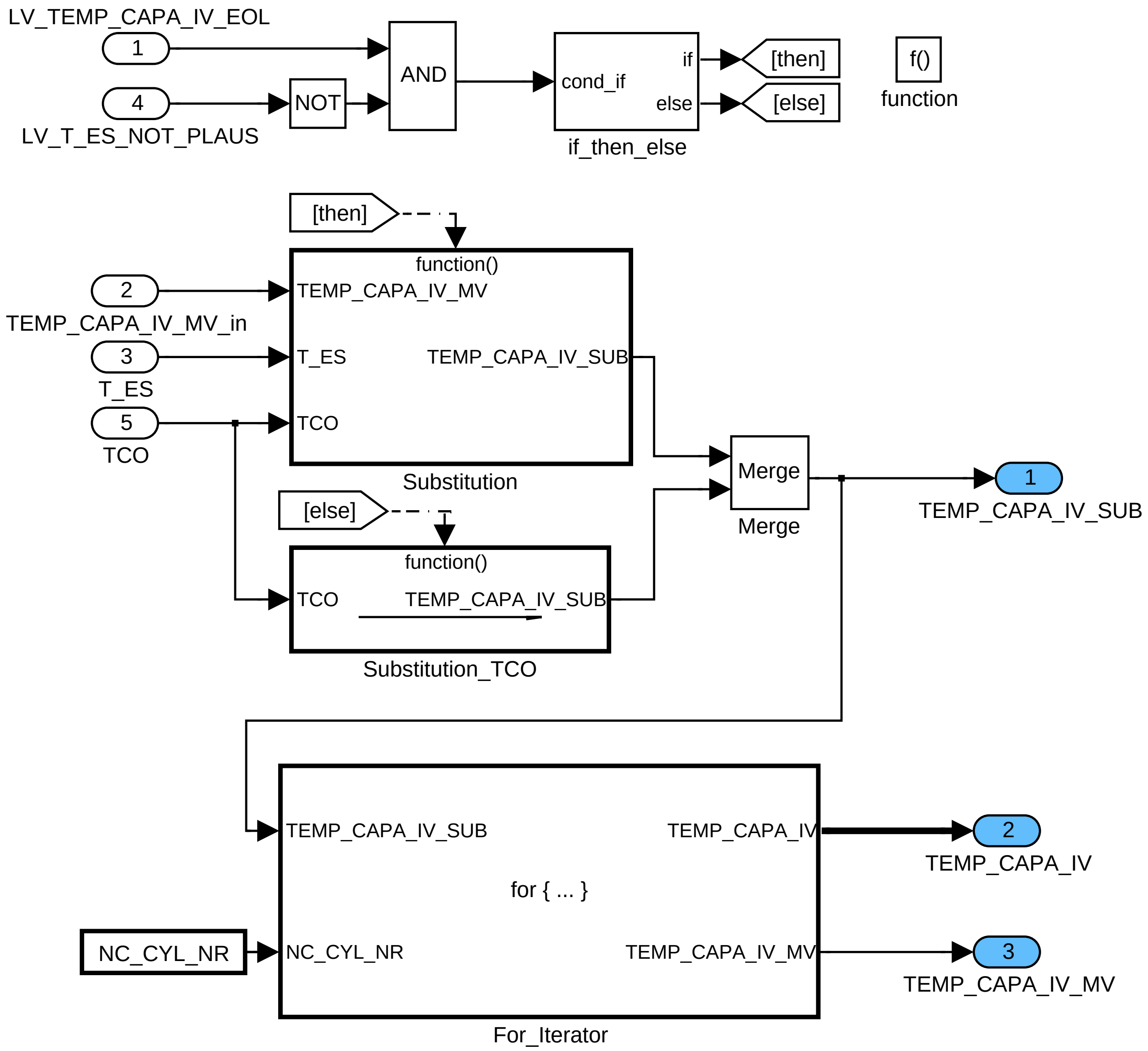


Figure 15 INJR_ISPCLCAPA0/ init2/ TEMP_CAPA_Sunstitution

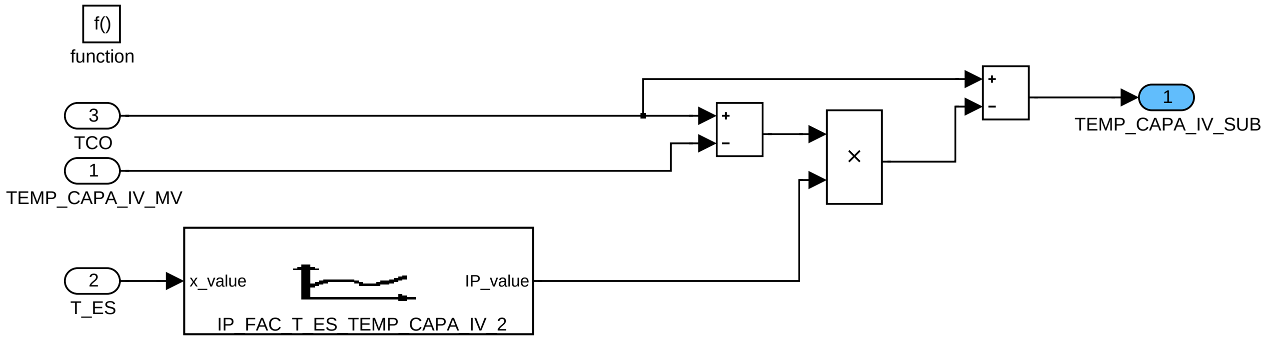


Figure 16 INJR_ISPCLCAPA0/ init2/ TEMP_CAPA_Sunstitution/ Substitution

41.25.1.4 Calculation of filtered capacity, capacity offset and stack temperature

Filtering of piezo capacity

The large signal capacity is calculated out of measured charge and voltage value. Since the signal is noisy it has to be low pass filtered. The filter constant can be applied via C_FAC_FIL_CAPA_IV. The filtered capacity is used for the further calculations. At first entry the filtered capacity is set to the measured capacity value via reset (R) and intial value (IV) input ports. The filter is enabled if the measured voltage and charge values are valid (LV_IV_MES_VLD), if the measured injection time is greater than a calibratable constant and if high energy range is used (LV_EGY_RNG_IV_PLS_1_ACT).

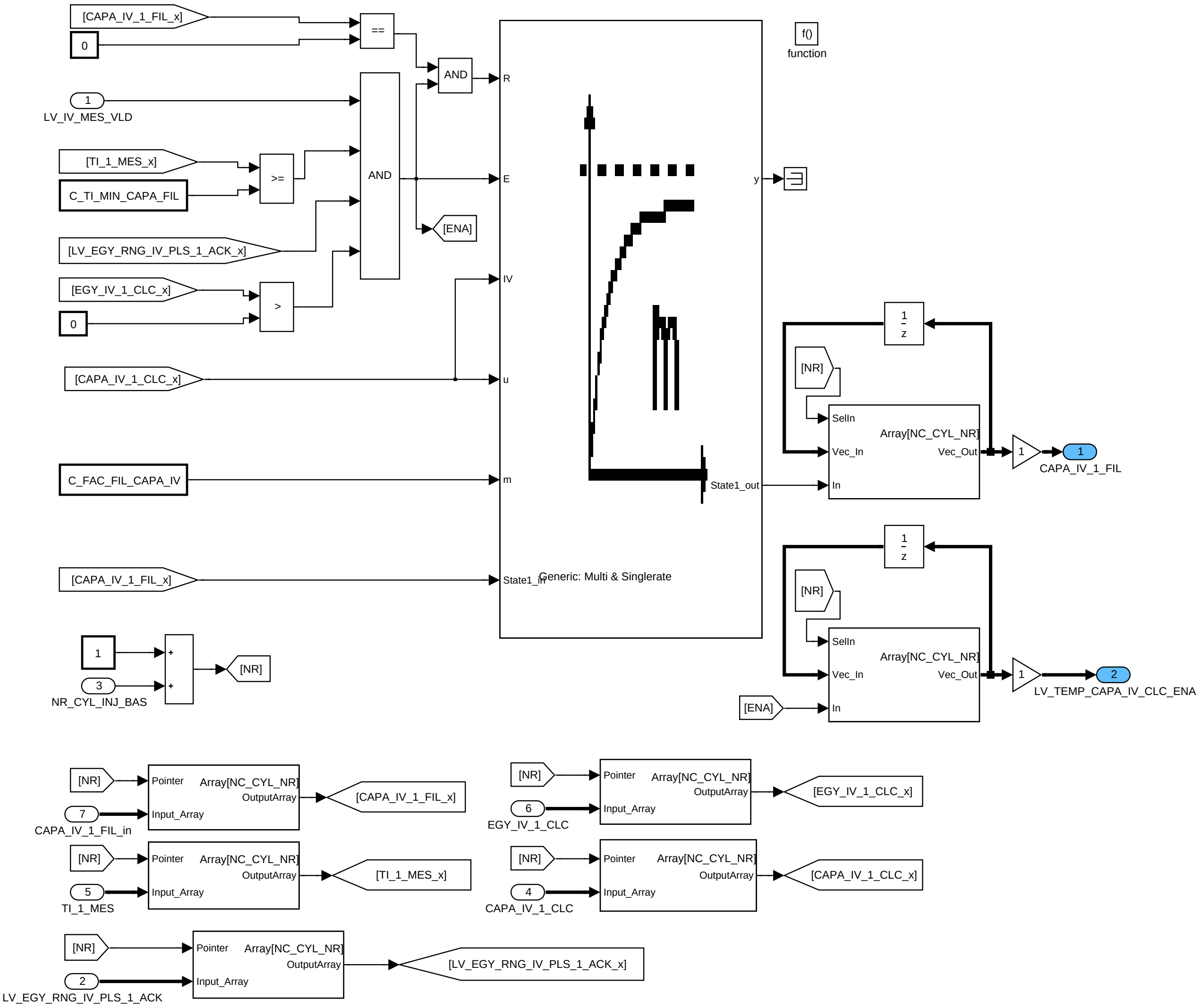


Figure 17 INJR_ISPCLCAPA0/ operate/ Capa_Filter

Calculation of capacity offset

The capacity offset of each injector is stored in the nonvolatile memory. A new adaptation value is calculated if the engine offtime T_ES is plausible and greater than a calibratable constant (LV_T_ES_CAPA_TEMP_VLD = TRUE). Then the coolant temperature TCO is used as the reference temperature. The calculation of the offset value is triggered as soon as the segment counter exceeds a calibratable level. The capacity offset is limited to C_CAPA_IV_DIF_NOM_LIM. The calculation of capacity offset can be inhibited if catalyst heating is active. For the first engine start the end-of-line flag LV_TEMP_CAPA_IV_EOL is set if adaptation was done.

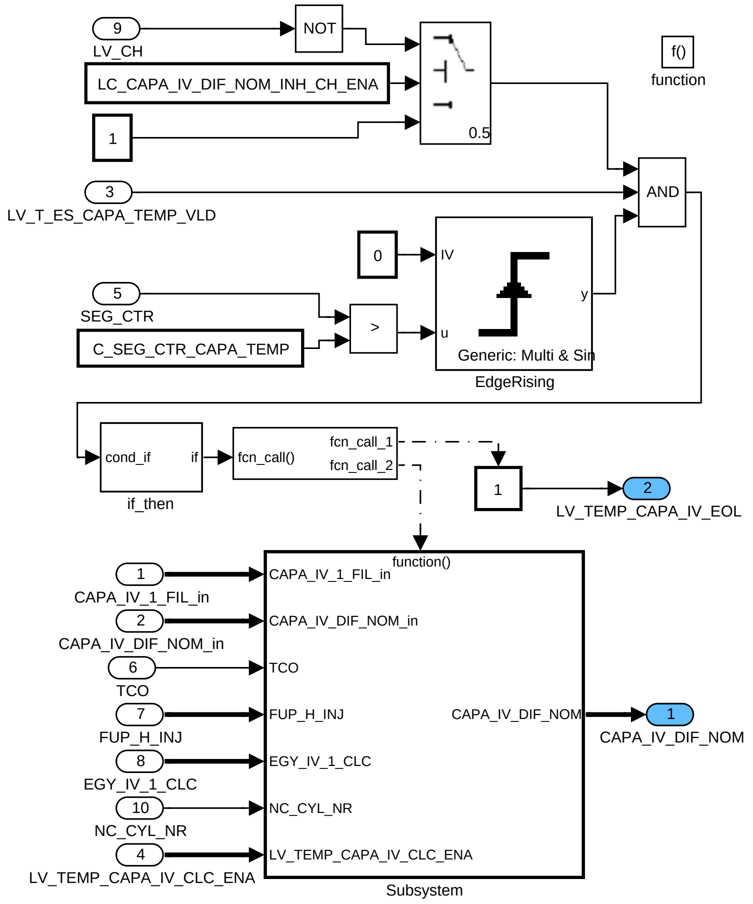


Figure 18 INJR_ISPCLCAPA0/ operate/ Capacity_Offset

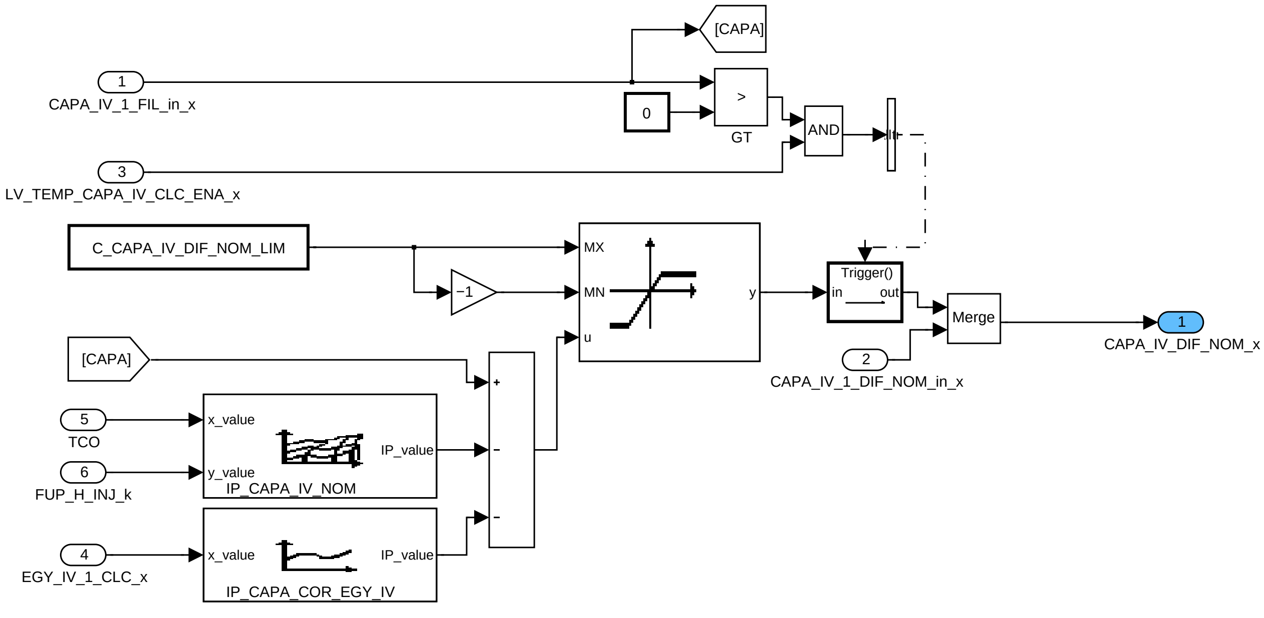


Figure 19 INJR_ISPCLCAPA0/ operate/ Capacity_Offset/ Subsystem/ For_Iterator/ Sub_System

Calculation of piezo stack temperature

While the segment counter is lower than a calibratable constant the temperature of the piezo stacks is set to the substitute value TEMP_CAPA_IV_SUB. When the SEG_CTR exceed this threshold the temperature is extracted from the map IP_TEMP_CAPA_IV_NOM accounting for each stack's individual capacity offset to the nominal value.

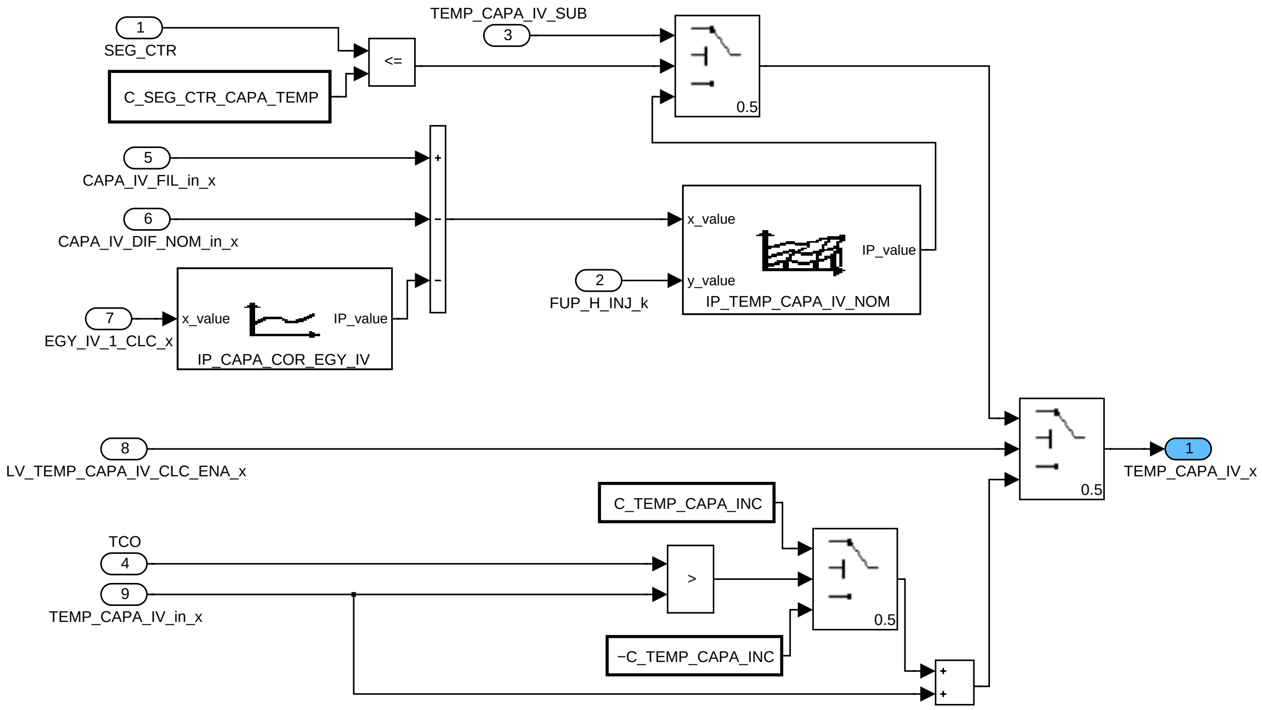


Figure 20 INJR_ISPCLCAPA0/ operate/ Calc_of_Temp/ For_Iterator2/ Subsystem

Manual setting for end of line flag

The end-of-line fag LV_TEMP_CAPA_IV_EOL can be set manually via LC_TEMP_CAPA_IV_EOL_MAN and C_TEMP_CAPA_IV_EOL.

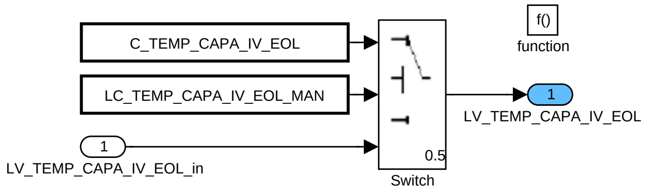


Figure 21 INJR_ISPCLCAPA0/ operate/ Manual_setting_for_EOL

41.25.1.5 SUBFUNCTION: SIG_MNG

41.26 Injector needle lift adjustment

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_IV_EGY_RNG_1	V/O	0...1H	0...1	1	[-]
Flag for the injector energy range of the first pulse (0 = low range, 1 = high range)					
LV_IV_EGY_RNG_2	V/O	0...1H	0...1	1	[-]
Flag for the injector energy range of the second pulse (0 = low range, 1 = high range)					
LV_IV_EGY_RNG_3	V/O	0...1H	0...1	1	[-]
Flag for the injector energy range of the third pulse (0 = low range, 1 = high range)					
EGY_ADD_AD[NC_CYL_NR]	V/O	E000...2000H	-50...50	6.1035e-3	[%]
Adaptive piezo energy correction depending on CYBL + CILC corrections					
LV_EGY_ADD_AD_LIM[NC_CYL_NR]	V/O	0...1H	0...1	1	[-]
Adaptive piezo energy correction is at limit, cylinder individual					
LV_EGY_ADD_AD_LIM_TOL[NC_CYL_NR]	V/O	0...1H	0...1	1	[-]
Adaptive piezo energy correction is at top limit, cylinder individual					
EGY_STEP_INJ_CHA_GRD_LIM_NOT[NC_CYL_NR]	V	0...4000H	0...100	6.1035e-3	[%]
Unlimited piezo injector energy charge gradient for a single step					
EGY_STEP_INJ_CHA_GRD_DIF_HOM[NC_CYL_NR]	V	E000...2000H	-50...50	6.1035e-3	[%]
Piezo injector energy charge gradient abundance for a single step - homogenous operation					
FAC_EGY_DIF_HOM[NC_CYL_NR]	V	0...4000H	-50...50	6.1035e-3	[%]
Energy-level-difference factor - homogenous operation					
FAC_MFF_DIF_HOM[NC_CYL_NR]	V/O	0...FFFFH	-1...0.99996	0.0305e-3	[-]
MFF-difference factor - homogenous operation					
EGY_STEP_INJ_CHA_GRD_DIF_S[NC_CYL_NR]	V	E000...2000H	-50...50	6.1035e-3	[%]
Piezo injector energy charge gradient abundance for a single step - stratified operation					
FAC_EGY_DIF_S[NC_CYL_NR]	V	0...4000H	-50...50	6.1035e-3	[%]
Energy-level-difference factor - stratified operation					
FAC_MFF_DIF_S[NC_CYL_NR]	O/V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
MFF-difference factor - stratified operation					
~~TI_1_STND_S_TMP[NC_CYL_NR]~~	~~-~~	~~0...FFFFH~~	~~0...65.535~~	~~0.001~~	~~[ms]~~
~~Mapped Injection time, stratified mode, 1. pulse for temporary values~~					
~~TI_2_STND_S_TMP[NC_CYL_NR]~~	~~-~~	~~0...FFFFH~~	~~0...65.535~~	~~0.001~~	~~[ms]~~
~~Mapped Injection time, stratified mode, 2. pulse for temporary values~~					
~~TI_3_STND_S_TMP[NC_CYL_NR]~~	-	~~0...FFFFH~~	~~0...65.535~~	~~0.001~~	~~[ms]~~
~~Mapped Injection time, stratified mode, 3. pulse for temporary values~~					
IDX_TI_1_STND_S_TMP[NC_CYL_NR]	-	0...FFFFH	0...65535	1	[-]
Temporary counting factor index for first pulse S.					
IDX_TI_2_STND_S_TMP[NC_CYL_NR]	-	0...FFFFH	0...65535	1	[-]
Temporary counting factor index for second pulse S.					
IDX_TI_3_STND_S_TMP[NC_CYL_NR]	-	0...FFFFH	0...65535	1	[-]
Temporary counting factor index for third pulse S.					
~~TI_THD_IV_EGY_RNG~~	~~V~~	~~0...FFFFH~~	~~0...65.535~~	~~0.001~~	~~[ms]~~
~~Injection time threshold for the selection of the injector energy range~~					
IDX_TI_THD_IV_EGY_RNG	V	0...FFFFH	0...65535	1	[-]
Counting factor index threshold for the selection of the injector energy range in stratified mode.					
FAC_TI_COR_IV_EGY_RNG_L[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Multiplicative injection time correction factor for corrections in case of injector low lift					
FAC_MFF_DIF_MV	O/V	0...FFFFH	-1...0.99996	0.0305e-3	[-]
MFF-difference factor - mean value					
LV_EGY_RNG_IV_PLS_1_S[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Flag for the injector energy range of the first pulse stratified mode (0 = low range, 1 = high range)					
LV_EGY_RNG_IV_PLS_2_S[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Flag for the injector energy range of the second pulse stratified mode (0 = low range, 1 = high range)					
LV_EGY_RNG_IV_PLS_3_S[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Flag for the injector energy range of the third pulse stratified mode (0 = low range, 1 = high range)					
LV_EGY_RNG_IV_PLS_1_HOM[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Flag for the injector energy range of the first pulse homogeneous mode (0 = low range, 1 = high range)					
LV_EGY_RNG_IV_PLS_2_HOM[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Flag for the injector energy range of the second pulse homogeneous mode (0 = low range, 1 = high range)					
LV_EGY_RNG_IV_PLS_3_HOM[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Flag for the injector energy range of the third pulse homogeneous mode (0 = low range, 1 = high range)					
LF_EGY_RNG_IV_1_HOM_REQ	V	0...FFH	0...255	1	[-]
Bit coded injector needle lift request for each cylinder; first pulse homogeneous mode					
LF_EGY_RNG_IV_2_HOM_REQ	V	0...FFH	0...255	1	[-]
Bit coded injector needle lift request for each cylinder; second pulse homogeneous mode					
LF_EGY_RNG_IV_3_HOM_REQ	V	0...FFH	0...255	1	[-]
Bit coded injector needle lift request for each cylinder; third pulse homogeneous mode					
LF_EGY_RNG_IV_1_S_REQ	V	0...FFH	0...255	1	[-]
Bit coded injector needle lift request for each cylinder; first pulse stratified mode					
LF_EGY_RNG_IV_2_S_REQ	V	0...FFH	0...255	1	[-]
Bit coded injector needle lift request for each cylinder; second pulse stratified mode					
LF_EGY_RNG_IV_3_S_REQ	V	0...FFH	0...255	1	[-]
Bit coded injector needle lift request for each cylinder; third pulse stratified mode					
MFF_THD_EGY_RNG_IV[NC_CBK_HPP_NR]	V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass threshold for the selection of the injector energy range depending on fuel pressure					

Input data:

EGY_LEVEL_IV_BAL[NC_CYL_NR]	MFF_SP_1_S[NC_CYL_NR]	MFF_SP_3_HOM[NC_CYL_NR]	N_32
CTR_CYL_NR_ST_CLC_INJR	MFF_SP_2_S[NC_CYL_NR]	TI_FAC[NC_CYL_NR]	LV_S_CLC
TFU_INJ	FUP_H_INJ[NC_CBK_HPP_NR]	TI_TUN_IV[NC_CYL_NR]	LV_ST_END

NC_IDX_CYL_HPP_REF[NC_CYL_NR]	NC_CYL_NR	STATE_INJ_MOD_HOM_REQ
		LV_IV_POST_EGY_RNG
NC_INJ_MOD_HOM	NC_INJ_MOD_MASK_1	NC_INJ_MOD_MULTI

EGY_STEP_INJ_CHA_GRD_BAS[NC_CYL_NR]	STATE_INJ_MOD_REQ
IP_EGY_LEVEL_IV_BAL_CONV	EGY_LEVEL_IV_BAL_CONV[NC_CYL_NR]
LV_EGY_STEP_INJ_CHA_LIM[NC_CYL_NR]	EGY_STEP_INJ_CHA_LIM_DIF[NC_CYL_NR]
	MFF_SP_2_HOM[NC_CYL_NR]

MFF_SP_MV	IP_FAC_EGY_LEVEL_IV_BAL_COR	STATE_INJ_MOD_S_REQ	MFF_SP_3_S[NC_CYL_NR]
NC_CBK_HPP_NR	STATE_INJ_MOD_HOMS_REQ	~~IP_MFF_TI_EGY_H~~	~~IP_MFF_TI_EGY_L~~
PRS_DEC_INJ_1_S[NC_CYL_NR]	PRS_DEC_INJ_2_S[NC_CYL_NR]	PRS_DEC_INJ_3_S[NC_CYL_NR]	FAC_TI_1_PRS_CYL_S[NC_CYL_NR]
FAC_TI_2_PRS_CYL_S[NC_CYL_NR]	FAC_TI_3_PRS_CYL_S[NC_CYL_NR]		
LFT_L_IV_HOM_REQ	LFT_L_IV_S_REQ	NR_CYL_CLC_RED_INJ	INJ_MOD_SP_HOM[NC_CYL_NR]
INJ_MOD_SP_S[NC_CYL_NR]	INJ_MOD_SP_HOMS[NC_CYL_NR]	IP_MFF_IDX_TI_EGY_H	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EGY_STEP_INJ_CHA_GRD_MAX_HOM	1	0...4000H	0...100	6.1035e-3	[%]
Upper threshold of energy charge gradient for a single step - homogenous operation					
C_EGY_STEP_INJ_CHA_GRD_MIN_HOM	1	0...4000H	0...100	6.1035e-3	[%]
Lower threshold of energy charge gradient for a single step - homogenous operation					
C_EGY_STEP_INJ_CHA_GRD_MAX_S	1	0...4000H	0...100	6.1035e-3	[%]
Upper threshold of energy charge gradient for a single step - stratified operation					
C_EGY_STEP_INJ_CHA_GRD_MIN_S	1	0...4000H	0...100	6.1035e-3	[%]
Lower threshold of energy charge gradient for a single step - stratified operation					
LC_EGY_RNG_IV_CYL_IND_DI	1	0...1H	0...1	1	[-]
Switch to disable cylinder individual injector needle lift					
~~ID_TI_THD_IV_EGY_RNG~~	~~3~~	~~0...FFFFH~~	~~0...65.535~~	~~0.001~~	~~[ms]~~
~~LDP_N_32_TI_THD_IV_EGY_RNG~~	~~3~~	~~0...FFH~~	~~0...8160~~	~~32~~	~~[rpm]~~
~~Injection time threshold for the selection of the injector energy range depending on engine speed~~					
ID_IDX_TI_THD_IV_EGY_RNG	3	0...FFFFH	0...65535	1	[-]
LDP_N_32_TI_THD_IV_EGY_RNG	3	0...FFH	0...8160	32	[rpm]
Counting factor index threshold for the selection of the injector energy range depending on engine speed.					
~~C_TI_HYS_IV_EGY_RNG~~	~~1~~	~~0...FFFFH~~	~~0...65.535~~	~~0.001~~	~~[ms]~~
~~Injection time hysteresis for the selection of the injector energy range~~					
C_IDX_TI_HYS_IV_EGY_RNG	1	0...FFFFH	0...65535	1	[-]
Counting factor index hysteresis for the selection of the injector energy range.					
IP_MFF_THD_IV_EGY_RNG	4	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_FUP_MFF_THD_IV_EGY_RNG	4	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel mass threshold for the selection of the injector energy range depending on FUP					
C_MFF_HYS_IV_EGY_RNG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass hysteresis for the selection of the injector energy range					
IP_FAC_TI_COR_IV_EGY_RNG_L	12	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_FUP_FAC_TI_COR_IV_EGY_RNG_L	12	0...FFFFH	0...347776	5.3067216	[hPa]
Multiplicative injection time correction factor for corrections in case of injector low lift depending on FUP					

DESCRIPTION:

General information:

This functionality switches between two different base energy ranges for PIEZO injectors. Starting with these base ranges, specific energy levels are calculated for each cylinder depending on the cylinder balancing injection valve energy level.

Application Conditions:

Initialization: at reset and at first valid tooth:

```
LF_EGY_RNG_IV_1_HOM_REQ = (2^NC_CYL_NR) - 1
LF_EGY_RNG_IV_2_HOM_REQ = (2^NC_CYL_NR) - 1
LF_EGY_RNG_IV_3_HOM_REQ = (2^NC_CYL_NR) - 1
LF_EGY_RNG_IV_1_S_REQ = (2^NC_CYL_NR) - 1
LF_EGY_RNG_IV_2_S_REQ = (2^NC_CYL_NR) - 1
LF_EGY_RNG_IV_3_S_REQ = (2^NC_CYL_NR) - 1
LV_IV_EGY_RNG_1 = 1
LV_IV_EGY_RNG_2 = 1
LV_IV_EGY_RNG_3 = 1
LV_EGY_RNG_IV_PLS_1_S[ ] = 1                 // all NC_CYL_NR elements
LV_EGY_RNG_IV_PLS_2_S[ ] = 1                 // all NC_CYL_NR elements
LV_EGY_RNG_IV_PLS_3_S[ ] = 1                 // all NC_CYL_NR elements
LV_EGY_RNG_IV_PLS_1_HOM[ ] = 1               // all NC_CYL_NR elements
LV_EGY_RNG_IV_PLS_2_HOM[ ] = 1               // all NC_CYL_NR elements
LV_EGY_RNG_IV_PLS_3_HOM[ ] = 1               // all NC_CYL_NR elements

EGY_ADD_AD[x] = EGY_LEVEL_IV_BAL[x]

EGY_STEP_INJ_CHA_GRD_DIF_HOM[x] = 0
FAC_EGY_DIF_HOM[x] = 0
FAC_MFF_DIF_HOM[x] = 0
EGY_STEP_INJ_CHA_GRD_DIF_S[x] = 0
FAC_EGY_DIF_S[x] = 0
FAC_MFF_DIF_S[x] = 0
FAC_MFF_DIF_MV = 0

FAC_TI_COR_IV_EGY_RNG_L = 1.0
```

Recurrence:

LV_ST_END = 0: 10ms
LV_ST_END = 1: segment synchronous

Activation: every engine state

Deactivation: -

Formula section:

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
            x = CTR_CYL_NR_ST_CLC_INJR + i
            (1a)IF x  >= NC_CYL_NR
            (1a)THEN
                    x = x – NC_CYL_NR
            (1a)ENDIF
            <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
     <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

41.26.1 Cylinder individual adjustment of the energy level

This function calculates cylinder individual charge and discharge gradients depending on the cylinder balancing injection valve energy level. Should the unlimited gradient value run out of the lower/upper limits, the gradient variable shall be limited to the maximum or minimum threshold – dependently on engine operation mode (homogenous or stratified) - and the energy-level-difference factor shall be converted in the MFF-difference factor.

```
if(1)  {(STATE_INJ_MOD_REQ & NC_INJ_MOD_MASK_1) = NC_INJ_MOD_HOM}
       [Bit-wise AND- operator]
then
    for(2)  cylinder_start to cylinder_stop DO
            EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] =
                        EGY_STEP_INJ_CHA_GRD_BAS[x] + EGY_LEVEL_IV_BAL[x]
            if(3)   EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] >
                                C_EGY_STEP_INJ_CHA_GRD_MAX_HOM
            then
                    LV_EGY_ADD_AD_LIM[x] = 1
                    LV_EGY_ADD_AD_LIM_TOL[x] = 1
                    EGY_STEP_INJ_CHA_GRD_DIF_HOM[x] =
                                EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] –
                                C_EGY_STEP_INJ_CHA_GRD_MAX_HOM
                    if(4)   LV_EGY_STEP_INJ_CHA_LIM[x] = 1
```

```
        then
            FAC_EGY_DIF_HOM[x] =
                                  (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                               (EGY_STEP_INJ_CHA_GRD_DIF_HOM[x] +
                                          EGY_STEP_INJ_CHA_LIM_DIF[x])/
                                          (100% + EGY_LEVEL_IV_BAL[x])/
                        IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
        else(4)
            FAC_EGY_DIF_HOM[x] =
                                  (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                                EGY_STEP_INJ_CHA_GRD_DIF_HOM[x]/
                                          (100% + EGY_LEVEL_IV_BAL[x])/
                        IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
        endif(4)
        FAC_MFF_DIF_HOM[x] = inverse(IP_EGY_LEVEL_IV_BAL_CONV
                                                 (FAC_EGY_DIF_HOM[x]))

        EGY_ADD_AD[x] = C_EGY_STEP_INJ_CHA_GRD_MAX_HOM –
                                          EGY_STEP_INJ_CHA_GRD_BAS[x]
    else(3)
        if(4)   EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] <
                                  C_EGY_STEP_INJ_CHA_GRD_MIN_HOM
        then
            LV_EGY_ADD_AD_LIM[x] = 1
            EGY_STEP_INJ_CHA_GRD_DIF_HOM[x] =
                                  EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] -
                                  C_EGY_STEP_INJ_CHA_GRD_MIN_HOM
            if(5)   LV_EGY_STEP_INJ_CHA_LIM[x] = 1
            then
                FAC_EGY_DIF_HOM[x] =
                                  (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                               (EGY_STEP_INJ_CHA_GRD_DIF_HOM[x] +
                                          EGY_STEP_INJ_CHA_LIM_DIF[x])/
                                          (100% + EGY_LEVEL_IV_BAL[x])/
                        IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
            else(5)
                FAC_EGY_DIF_HOM[x] =
```

```
                        (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                        EGY_STEP_INJ_CHA_GRD_DIF_HOM[x]/
                        (100% + EGY_LEVEL_IV_BAL[x])/
                        IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
                endif(5)
                FAC_MFF_DIF_HOM[x] =
                        inverse(IP_EGY_LEVEL_IV_BAL_CONV
                        (FAC_EGY_DIF_HOM[x]))
                EGY_ADD_AD[x] = C_EGY_STEP_INJ_CHA_GRD_MIN_HOM –
                        EGY_STEP_INJ_CHA_GRD_BAS[x]
            else(4)
                if(5)   LV_EGY_STEP_INJ_CHA_LIM[x] = 1
                then
                    FAC_EGY_DIF_HOM[x] =
                        (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                        EGY_STEP_INJ_CHA_LIM_DIF[x]/
                        (100% + EGY_LEVEL_IV_BAL[x])/
                        IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
                    FAC_MFF_DIF_HOM[x] =
                        inverse(IP_EGY_LEVEL_IV_BAL_CONV
                        (FAC_EGY_DIF_HOM[x]))

                else(5)
                    FAC_EGY_DIF_HOM[x] = 0
                    FAC_MFF_DIF_HOM[x] = 0
                endif(5)
                EGY_STEP_INJ_CHA_GRD_DIF_HOM[x] = 0
                EGY_ADD_AD[x] = EGY_LEVEL_IV_BAL[x]
                LV_EGY_ADD_AD_LIM[x] = 0
                LV_EGY_ADD_AD_LIM_TOL[x] = 0
            endif(4)
        endif(3)
    endfor(2)

    calculation of mean value of FAC_MFF_DIF_HOM[ ] (mean value over all cylinders):
    FAC_MFF_DIF_MV = mean(FAC_MFF_DIF_HOM[ ])

else(1)
    for(2)  cylinder_start to cylinder_stop DO
        EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] =
                EGY_STEP_INJ_CHA_GRD_BAS[x] + EGY_LEVEL_IV_BAL[x]
```

```
if(3)   EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] >
                                C_EGY_STEP_INJ_CHA_GRD_MAX_S
then
        LV_EGY_ADD_AD_LIM[x] = 1
        LV_EGY_ADD_AD_LIM_TOL[x] = 1
        EGY_STEP_INJ_CHA_GRD_DIF_S[x] =
                                EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] –
                                C_EGY_STEP_INJ_CHA_GRD_MAX_S
        if(4)   LV_EGY_STEP_INJ_CHA_LIM[x] = 1
        then
                FAC_EGY_DIF_S[x] = (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                                (EGY_STEP_INJ_CHA_GRD_DIF_S[x] +
                                EGY_STEP_INJ_CHA_LIM_DIF[x])/
                                (100% + EGY_LEVEL_IV_BAL[x])/
                                IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
        else(4)
                FAC_EGY_DIF_S[x] = (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                                EGY_STEP_INJ_CHA_GRD_DIF_S[x]/
                                (100% + EGY_LEVEL_IV_BAL[x])/
                                IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
        endif(4)
        FAC_MFF_DIF_S[x] = inverse(IP_EGY_LEVEL_IV_BAL_CONV
                                (FAC_EGY_DIF_S[x]))
        EGY_ADD_AD[x] = C_EGY_STEP_INJ_CHA_GRD_MAX_S -
                                EGY_STEP_INJ_CHA_GRD_BAS[x]
else(3)
        if(4)   EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] <
                                C_EGY_STEP_INJ_CHA_GRD_MIN_S
        then
                LV_EGY_ADD_AD_LIM[x] = 1
                EGY_STEP_INJ_CHA_GRD_DIF_S[x] =
                                EGY_STEP_INJ_CHA_GRD_LIM_NOT[x] -
                                C_EGY_STEP_INJ_CHA_GRD_MIN_S
                if(5)   LV_EGY_STEP_INJ_CHA_LIM[x] = 1
                then
                        FAC_EGY_DIF_S[x] =
                                (100% + EGY_LEVEL_IV_BAL_CONV[x])*
```

```
                                                (EGY_STEP_INJ_CHA_GRD_DIF_S[x] +
                                                    EGY_STEP_INJ_CHA_LIM_DIF[x])/
                                                    (100% + EGY_LEVEL_IV_BAL[x])/
                                          IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
                    else$^{(5)}$
                        FAC_EGY_DIF_S[x] =
                                                (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                                                    EGY_STEP_INJ_CHA_GRD_DIF_S[x]/
                                                    (100% + EGY_LEVEL_IV_BAL[x])/
                                          IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
                    endif$^{(5)}$
                    FAC_MFF_DIF_S[x] = inverse(IP_EGY_LEVEL_IV_BAL_CONV
                                                            (FAC_EGY_DIF_S[x]))

                EGY_ADD_AD[x] = C_EGY_STEP_INJ_CHA_GRD_MIN_S -
                                                    EGY_STEP_INJ_CHA_GRD_BAS[x]
                else$^{(4)}$
                    if$^{(5)}$  LV_EGY_STEP_INJ_CHA_LIM[x] = 1
                    then
                        FAC_EGY_DIF_S[x] =
                                                (100% + EGY_LEVEL_IV_BAL_CONV[x])*
                                                    EGY_STEP_INJ_CHA_LIM_DIF[x]/
                                                    (100% + EGY_LEVEL_IV_BAL[x])/
                                          IP_FAC_EGY_LEVEL_IV_BAL_COR(MFF_SP_MV)
                        FAC_MFF_DIF_S[x] =
                                                inverse(IP_EGY_LEVEL_IV_BAL_CONV
                                                            (FAC_EGY_DIF_S[x]))
                    else$^{(5)}$
                        FAC_EGY_DIF_S[x] = 0
                        FAC_MFF_DIF_S[x] = 0
                    endif$^{(5)}$
                    EGY_STEP_INJ_CHA_GRD_DIF_S[x] = 0
                    EGY_ADD_AD[x] = EGY_LEVEL_IV_BAL[x]
                    LV_EGY_ADD_AD_LIM[x] = 0
                    LV_EGY_ADD_AD_LIM_TOL[x] = 0
                endif$^{(4)}$
            endif$^{(3)}$

        endfor$^{(2)}$
```

```
        calculation of mean value of FAC_MFF_DIF_S[ ] (mean value over all cylinders):
        FAC_MFF_DIF_MV = mean(FAC_MFF_DIF_S[ ])

    endif(1)
```

41.26.2 Selection of injector energy range

41.26.2.1 Basis calculations

The minimum fuel mass setpoint for stratified mode is determined from the pulses with a fuel mass setpoint greater than zero. The corresponding minimum injection time is calculated and if the injection time is greater than a threshold or homogeneous mode is selected, high energy range will be selected for the injector, pulse individual.

The energy range can also be clipped to low range by dynamic injector needle lift selection (bit field LFT_L_IV_S_REQ).

Important note: For better readability of the specification, the LSB of the bit field LFT_L_IV_S_REQ, which corresponds to the physically first injection pulse, is referred to as bit 1 (**not** bit 0 as it would be in SW). The next bit would be bit 2 etc.

41.26.2.2 Main calculations for stratified combustion mode:

```
FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
    FAC_TI_COR_IV_EGY_RNG_L[m] =
                IP_FAC_TI_COR_IV_EGY_RNG_L (Input: FUP_H_INJ[m])
ENDFOR
if(1) LV_S_CLC = 1          Check if stratified mode has to be calculated
then
    TI_THD_IV_EGY_RNG = ID_TI_THD_IV_EGY_RNG (Input: N_32)   [struck through]
    IDX_TI_THD_IV_EGY_RNG = ID_IDX_TI_THD_IV_EGY_RNG (Input: N_32)
    (1)FOR cylinder_start to cylinder_stop DO
        check, if third pulse stratified mode has to be calculated
        if(2)   INJ_MOD_SP_S[x] = "MPLS+PLS3" OR
                INJ_MOD_SP_HOMS[x] = "HOMS+PLS3"
        then    [Third pulse has to be calculated]
                calculate section 41.26.2.3 (third pulse stratified)
        endif(2)
        check, if second pulse stratified mode has to be calculated
        if(3)   INJ_MOD_SP_S[x] = "MPLS" OR
                INJ_MOD_SP_S[x] = "MPLS+PLS3" OR
                INJ_MOD_SP_HOMS[x] = "HOMS" OR
                INJ_MOD_SP_HOMS[x] = "HOMS+PLS3"
        then
                calculate section 41.26.2.4 (second pulse stratified)
        endif(3)
        always calculate first pulse stratified mode, if LV_S_CLC = 1
        calculate section 41.26.2.5 (first pulse stratified)
    (1)ENDFOR
endif(1)
```

end of section 41.26.2.2

41.26.2.3 Calculate energy selection parameters for the third injection pulse stratified mode:

Note: Calculate this section only if called by another section

```
IDX_TI_3_STND_S_TMP[x] = IP_MFF_IDX_TI_EGY_H (
                    MFF_SP_3_S[x]*(1 + FAC_MFF_DIF_S[x]); PRS_DEC_INJ_3_S[x])
                    * TI_FAC[x] * TI_TUN_IV[x] * FAC_TI_3_PRS_CYL_S[x]


if(2)   (IDX_TI_3_STND_S_TMP[x] < IDX_TI_THD_IV_EGY_RNG) or
        Bit 3 of LFT_L_IV_S_REQ = 1
then
        LV_EGY_RNG_IV_PLS_3_S[x] = 0
        Bit x of LF_EGY_RNG_IV_3_S_REQ = 0
        [Select low energy range]

else(2)
        if(3)   IDX_TI_3_STND_S_TMP[x] > (IDX_TI_THD_IV_EGY_RNG
                                            + C_IDX_TI_HYS_IV_EGY_RNG)
        then
                LV_EGY_RNG_IV_PLS_3_S[x] = 1
                Bit x of LF_EGY_RNG_IV_3_S_REQ = 1
                [Select high energy range]

        else(3)
                [no operation: energy range unchanged]
        endif(3)
endif(2)
```

end of section 41.26.2.3

41.26.2.4 Calculate energy selection parameters for the second injection pulse stratified mode:

Note: Calculate this section only if called by another section

```
IDX_TI_2_STND_S_TMP[x] = IP_MFF_IDX_TI_EGY_H (
                MFF_SP_2_S[x]*(1 + FAC_MFF_DIF_S[x]); PRS_DEC_INJ_2_S[x])
                * TI_FAC[x] * TI_TUN_IV[x] * FAC_TI_2_PRS_CYL_S[x]


if(2)   (IDX_TI_2_STND_S_TMP[x] < IDX_TI_THD_IV_EGY_RNG) or
        Bit 2 of LFT_L_IV_S_REQ = 1
then
        LV_EGY_RNG_IV_PLS_2_S[x] = 0
        Bit x of LF_EGY_RNG_IV_2_S_REQ = 0
        [Select low energy range]

else(2)
        if(3)   IDX_TI_2_STND_S_TMP[x] > (IDX_TI_THD_IV_EGY_RNG
                                          + C_ IDX_TI_HYS_IV_EGY_RNG)
        then
                LV_EGY_RNG_IV_PLS_2_S[x] = 1
                Bit x of LF_EGY_RNG_IV_2_S_REQ = 1
                [Select high energy range]

        else(3)
                [no operation: energy range unchanged]
        endif(3)
endif(2)
```

end of section 41.26.2.4

41.26.2.5 Calculate energy selection parameters for the first injection pulse stratified mode:

Note: Calculate this section only if called by another section

```
IDX_TI_1_STND_S_TMP[x] = IP_MFF_IDX_TI_EGY_H (
                MFF_SP_1_S[x]*(1 + FAC_MFF_DIF_S[x]); PRS_DEC_INJ_1_S[x])
                * TI_FAC[x] * TI_TUN_IV[x] * FAC_TI_1_PRS_CYL_S[x]


if(2)   (IDX_TI_1_STND_S_TMP[x] < IDX_TI_THD_IV_EGY_RNG) or
        Bit 1 of LFT_L_IV_S_REQ = 1
then
        LV_EGY_RNG_IV_PLS_1_S[x] = 0
        Bit x of LF_EGY_RNG_IV_1_S_REQ = 0
        [Select low energy range]

else(2)
        if(3)   IDX_TI_1_STND_S_TMP[x] > (IDX_TI_THD_IV_EGY_RNG
                                          + C_ IDX_TI_HYS_IV_EGY_RNG)
        then
                LV_EGY_RNG_IV_PLS_1_S[x] = 1
                Bit x of LF_EGY_RNG_IV_1_S_REQ = 1
                [Select high energy range]

        else(3)
                [no operation: energy range unchanged]
        endif(3)
endif(2)
```

end of section 41.26.2.5

41.26.2.6 Energy range calculations for multiple homogeneous mode (Main calculations)

If multiple homogenous injection mode is selected, the second/third fuel mass setpoint will be checked against a threshold. If the fuel mass is below the threshold, the second/third injection pulse will be executed with low energy range.

The energy range can also be clipped to low range by dynamic injector needle lift selection (bit field LFT_L_IV_HOM_REQ).

Important note: For better readability of the specification, the LSB of the bit field LFT_L_IV_HOM_REQ, which corresponds to the physically first injection pulse, is referred to as bit 1 (**not** bit 0 as it would be in SW). The next bit would be bit 2 etc.

Note: For improvement of runtime, calculate the map IP_MFF_THD_IV_EGY_RNG only once per recurrence for every fuel pressure value!

```
FOR m = 0 TO NC_CBK_HPP_NR – 1 DO
    MFF_THD_EGY_RNG_IV[m] = IP_MFF_THD_IV_EGY_RNG(Input: FUP_H_INJ[m])
ENDFOR
(1)FOR cylinder_start to cylinder_stop DO
    check, if second or second and third pulse homogeneous mode has to be calculated
    if(1)   INJ_MOD_SP_HOM[x] = 'MPLH' OR
            INJ_MOD_SP_HOM[x] = 'MPLH+PLS3'
            [Homogeneous multiple injection mode requested]
    then
            calculate section 41.26.2.7 (second pulse homogeneous)

            check, if third pulse homogeneous mode has to be calculated
            if(2)   INJ_MOD_SP_HOM[x] = 'MPLH+PLS3'
                    [Homogeneous triple injection mode requested]
            then
                    calculate section 41.26.2.8 (third pulse homogeneous)
            endif(2)

    endif(1)

    always calculate parameters for the first pulse homogeneous mode
    calculate section 41.26.2.9 (first pulse homogeneous)
(1)ENDFOR
end of section 41.26.2.6
```

41.26.2.7 Calculate energy selection parameters for the second injection pulse at homogeneous mode based on fuel mass threshold:

Note: Calculate this section only if called by another section

```
if$^{(2)}$   (MFF_SP_2_HOM[x] < MFF_THD_EGY_RNG_IV[NC_IDX_CYL_HPP_REF[x]])
        or Bit 2 of LFT_L_IV_HOM_REQ = 1
then
        LV_EGY_RNG_IV_PLS_2_HOM[x] = 0
        Bit x of LF_EGY_RNG_IV_2_HOM_REQ = 0
        [Select low energy range]

else$^{(2)}$
        if$^{(3)}$   MFF_SP_2_HOM[x] >
                (MFF_THD_EGY_RNG_IV[NC_IDX_CYL_HPP_REF[x]]
                                                + C_MFF_HYS_IV_EGY_RNG)
        then
                LV_EGY_RNG_IV_PLS_2_HOM[x] = 1
                Bit x of LF_EGY_RNG_IV_2_HOM_REQ = 1
                [Select high energy range]

        else$^{(3)}$
                [no operation: energy range unchanged]
        endif$^{(3)}$
endif$^{(2)}$
```

end of section 41.26.2.7

41.26.2.8 Calculate energy selection parameters for the third injection pulse at homogeneous mode based on fuel mass threshold:

Note: Calculate this section only if called by another section

```
if(2)   (MFF_SP_3_HOM[x] < MFF_THD_EGY_RNG_IV[NC_IDX_CYL_HPP_REF[x]])
        or Bit 3 of LFT_L_IV_HOM_REQ = 1
then
        LV_EGY_RNG_IV_PLS_3_HOM[x] = 0
        Bit x of LF_EGY_RNG_IV_3_HOM_REQ = 0
        [Select low energy range]

else(2)
        if(3)   MFF_SP_3_HOM[x] >
                (MFF_THD_EGY_RNG_IV[NC_IDX_CYL_HPP_REF[x]]
                                                + C_MFF_HYS_IV_EGY_RNG)
        then
                LV_EGY_RNG_IV_PLS_3_HOM[x] = 1
                Bit x of LF_EGY_RNG_IV_3_HOM_REQ = 1
                [Select high energy range]

        else(3)
                [no operation: energy range unchanged]
        endif(3)
endif(2)
```

end of section 41.26.2.8

41.26.2.9 Calculate energy selection parameters for the first injection pulse at homogeneous mode:

Note: Calculate this section only if called by another section

```
if(2)   (MFF_SP_1_HOM[x] < MFF_THD_EGY_RNG_IV[NC_IDX_CYL_HPP_REF[x]])
        or Bit 1 of LFT_L_IV_HOM_REQ = 1
then
        LV_EGY_RNG_IV_PLS_1_HOM[x] = 0
        Bit x of LF_EGY_RNG_IV_1_HOM_REQ = 0
        [Select low energy range]

else(2)
        if(3)   MFF_SP_1_HOM[x] >
                (MFF_THD_EGY_RNG_IV[NC_IDX_CYL_HPP_REF[x]]
                                        + C_MFF_HYS_IV_EGY_RNG)
        then
                LV_EGY_RNG_IV_PLS_1_HOM[x] = 1
                Bit x of LF_EGY_RNG_IV_1_HOM_REQ = 1
                [Select high energy range]

        else(3)
                [no operation: energy range unchanged]
        endif(3)
endif(2)
```

end of section 41.26.2.9

41.26.2.10 Calculate global energy selection parameters for monitoring unit:

```
(1)IF LC_EGY_RNG_IV_CYL_IND_DI = 1
(1)THEN
    (2)IF LV_S_CLC = 1
    (2)THEN
        (3)IF LF_EGY_RNG_IV_1_S_REQ = (2^NC_CYL_NR) – 1
        (3)THEN
            (1)FOR cylinder_start to cylinder_stop DO
                LV_EGY_RNG_IV_PLS_1_S[x] = 1
            (1)ENDFOR
        (3)ELSE
            (2)FOR cylinder_start to cylinder_stop DO
                LV_EGY_RNG_IV_PLS_1_S[x] = 0
            (2)ENDFOR
        (3)ENDIF
        (4)IF (STATE_INJ_MOD_S_REQ = "MPLS" OR
            STATE_INJ_MOD_S_REQ = "MPLS+PLS3" OR
            STATE_INJ_MOD_HOMS_REQ = "HOMS" OR
            STATE_INJ_MOD_HOMS_REQ = "HOMS+PLS3")
        (4)THEN
            (5)IF LF_EGY_RNG_IV_2_S_REQ = (2^NC_CYL_NR) – 1
            (5)THEN
                (3)FOR cylinder_start to cylinder_stop DO
                    LV_EGY_RNG_IV_PLS_2_S[x] = 1
                (3)ENDFOR
            (5)ELSE
                (4)FOR cylinder_start to cylinder_stop DO
                    LV_EGY_RNG_IV_PLS_2_S[x] = 0
                (4)ENDFOR
            (5)ENDIF
        (4)ENDIF
        (6)IF (STATE_INJ_MOD_S_REQ = "MPLS+PLS3" OR
            STATE_INJ_MOD_HOMS_REQ = "HOMS+PLS3")
        (6)THEN
            (7)IF LF_EGY_RNG_IV_3_S_REQ = (2^NC_CYL_NR) – 1
            (7)THEN
                (5)FOR cylinder_start to cylinder_stop DO
                    LV_EGY_RNG_IV_PLS_3_S[x] = 1
```

```
                    (5)ENDFOR
                (7)ELSE
                    (6)FOR cylinder_start to cylinder_stop DO
                        LV_EGY_RNG_IV_PLS_3_S[x] = 0
                    (6)ENDFOR
                (7)ENDIF
            (6)ENDIF
        (2)ENDIF – LV_S_CLC
        (8)IF LF_EGY_RNG_IV_1_HOM_REQ = (2^NC_CYL_NR) – 1
        (8)THEN
            (7)FOR cylinder_start to cylinder_stop DO
                LV_EGY_RNG_IV_PLS_1_HOM[x] = 1
            (7)ENDFOR
        (8)ELSE
            (8)FOR cylinder_start to cylinder_stop DO
                LV_EGY_RNG_IV_PLS_1_HOM[x] = 0
            (8)ENDFOR
        (8)ENDIF
        (9)IF (STATE_INJ_MOD_HOM_REQ = "MPLH" OR
            STATE_INJ_MOD_HOM_REQ = "MPLH+PLS3")
        (9)THEN
            (10)IF LF_EGY_RNG_IV_2_HOM_REQ = (2^NC_CYL_NR) – 1
            (10)THEN
                (9)FOR cylinder_start to cylinder_stop DO
                    LV_EGY_RNG_IV_PLS_2_HOM[x] = 1
                (9)ENDFOR
            (10)ELSE
                (10)FOR cylinder_start to cylinder_stop DO
                    LV_EGY_RNG_IV_PLS_2_HOM[x] = 0
                (10)ENDFOR
            (10)ENDIF
        (9)ENDIF
        (11)IF STATE_INJ_MOD_HOM_REQ = "MPLH+PLS3"
        (11)THEN
            (12)IF LF_EGY_RNG_IV_3_HOM_REQ = (2^NC_CYL_NR) – 1
            (12)THEN
                (11)FOR cylinder_start to cylinder_stop DO
                    LV_EGY_RNG_IV_PLS_3_HOM[x] = 1
```

```
                (11)ENDFOR
            (12)ELSE
                (12)FOR cylinder_start to cylinder_stop DO
                    LV_EGY_RNG_IV_PLS_3_HOM[x] = 0
                (12)ENDFOR
            (12)ENDIF
        (11)ENDIF
    (1)ENDIF - LC_EGY_RNG_IV_CYL_IND_DI
```

To keep the interface to the monitoring unit stable, LV_IV_EGY_RNG_1, LV_IV_EGY_RNG_2 and LV_IV_EGY_RNG_3 have to be set to the values of the homogeneous path in case of requested homogeneous injection or to the values of the stratified path else.

```
if(1)   {(STATE_INJ_MOD_REQ & NC_INJ_MOD_MASK_1) = NC_INJ_MOD_HOM}
then
        global requested injection mode is homogeneous mode; use homogeneous
        values
        LV_IV_EGY_RNG_1 =
                LV_EGY_RNG_IV_PLS_1_HOM[CTR_CYL_NR_ST_CLC_INJR]
        LV_IV_EGY_RNG_2 =
                LV_EGY_RNG_IV_PLS_2_HOM[CTR_CYL_NR_ST_CLC_INJR]
        LV_IV_EGY_RNG_3 =
                LV_EGY_RNG_IV_PLS_3_HOM[CTR_CYL_NR_ST_CLC_INJR]
else
        global requested injection mode is homs or stratified mode; use stratified values
        LV_IV_EGY_RNG_1 =
                LV_EGY_RNG_IV_PLS_1_S[CTR_CYL_NR_ST_CLC_INJR]
        LV_IV_EGY_RNG_2 =
                LV_EGY_RNG_IV_PLS_2_S[CTR_CYL_NR_ST_CLC_INJR]
        LV_IV_EGY_RNG_3 =
                LV_EGY_RNG_IV_PLS_3_S[CTR_CYL_NR_ST_CLC_INJR]
endif(1)
```

41.27 Injector needle lift control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_EGY_PWM_AD[NC_CYL_NR]	O/V/S	0...FFFFH	0...1.99996948	3.05176E-5	%/mJ
Conversion factor: EGY to % PWM for piezo injector					
MIN_INJ_EGY_LEVEL	O/V	E000...2000H	-50...50	0.00610352	%
Minimum correction value of injector energy level (calibrateable)					
MAX_INJ_EGY_LEVEL	O/V	E000...2000H	-50...50	0.00610352	%
Maximum correction value of injector energy level (calibrateable)					
NR_EGY_STEP_INJ_L	O/V	2...1FH	2...31	1	-
Number of charge steps for low piezo energy range					
NR_EGY_STEP_INJ_H	O/V	2...1FH	2...31	1	-
Number of charge steps for high piezo energy range					
EGY_STEP_INJ_CHA_GRD_BAS[NC_CYL_NR]	O/V	0...4000H	0...100	0.00610352	%
Piezo injector base energy charge gradient for a single step, without adaptation corrections.					
EGY_STEP_INJ_DCHA_GRD[NC_CYL_NR]	O/V	0...4000H	0...100	0.00610352	%
Piezo injector energy discharge gradient for a single step					
EGY_STEP_INJ_CHA_GRD[NC_CYL_NR]	O/V	0...4000H	0...100	0.00610352	%
Piezo injector energy charge gradient for a single step					
EGY_STEP_INJ_CHA_LIM_DIF[NC_CYL_NR]	O/V	0...4000H	0...100	0.00610352	%
Difference of piezo stack energy, which is over the voltage limit					
LV_EGY_STEP_INJ_CHA_LIM[NC_CYL_NR]	O/V	0...1H	0...1	1	-
Piezo stack energy is out of limit (over-voltage protection)					
EGY_MAX_V_LIM[NC_CYL_NR]	V	0...FFFFH	0...255	0.00389105	mJ
Maximum limit of energy after voltage limiter					
LV_EGY_PWM_LIM[NC_CYL_NR]	V	0...1H	0...1	1	-
Piezo stack energy demand over 100% PWM					
NR_EGY_CTL_CYL_CLC	V	0...7H	0...7	1	-
Cylinder for that the injector needle lift control is calculated					
LV_EGY_AD_ENA	V	0...1H	0...1	1	-
Piezo energy adaption is enabled and active, if set					
LV_V_LIM_ENA	V	0...1H	0...1	1	-
Piezo voltage limitation is enabled and active, if set					
EGY_STEP_INJ_CHA_GRD_OFS	V	E000...2000H	-50...50	0.00610352	%
Piezo injector energy charge and discharge gradient offset					
EGY_SP[NC_CYL_NR]	V	0...FFFFH	0...255	0.00389105	mJ
Energy setpoint for piezo injector energy controller after voltage limiter					
TEMP_SEL_EGY_CTL	V	0...FEH	-48...142.5	0.75	°C
Selected temperature for engergy controller functions.					
EGY_SP_NOT_LIM[NC_CYL_NR]	V	0...FFFFH	0...255	0.00389105	mJ
Unlimited energy setpoint for piezo injector energy controller					
FAC_EGY_COR_FIL_TMP[NC_CYL_NR]	-	8000...7FFFH	-1...0.99996948	3.05176E-5	%/mJ
Temporary correction factor after PT1 - filter, cylinder individual					
FAC_EGY_COR_I_TMP[NC_CYL_NR]	-	8000...7FFFH	-1...0.99996948	3.05176E-5	%/mJ
Temporary correction factor before Integrator, cylinder individual					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_EGY_PWM_AD_INI[NC_CYL_NR]	-	0...FFFFH	0...1.99996948	3.05176E-5	%/mJ
Conversion factor: EGY to % PWM for piezo injector at RESET					
V_INJ_IV_ERR	-	8000...7FFFH	-640 ... 639.980469	0.01953125	V
Control error of voltage dependent energy limiter					

Input data:

EGY_ADD_AD[NC_CYL_NR]	EGY_ADD_FUP[NC_CBK_HPP_NR]	EGY_ADD_TEMP	N
MFF_SP_MV	TFU_INJ	LV_EGY_RNG_IV_PLS_1_ACK[NC_CYL_NR]	LV_IV_MES_VLD
EGY_IV_1_CLC[NC_CYL_NR]	V_IV_1_MES[NC_CYL_NR]	TI_1_MES[NC_CYL_NR]	STATE_DIAG_IV
LV_EGY_ADD_AD_LIM[NC_CYL_NR]	NR_CYL_INJ_BAS	NC_CYL_NR	EGY_ADD_DUCY
EGY_ADD_COR_EXT[NC_CYL_NR]	LV_ES	EGY_SP_IV_EXT[NC_CYL_NR]	FUP_H_INJ[NC_CBK_HPP_NR]
LV_EGY_ADD_AD_LIM_TOL[NC_CYL_NR]	SEG_CTR	NC_CBK_HPP_NR	NC_IDX_CYL_HPP_REF[NC_CYL_NR]
TEMP_CAPA_IV_MV	TCO	TECU	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EGY_BAS_INJ_SP_H	1	0...FFFFH	0...255	0.00389105	mJ
Energy base value for piezo injector at high needle lift					
C_EGY_CTL_ENA_THD_MAX	1	0...FFFFH	0...255	0.00389105	mJ
Upper energy threshold to activate controller					
C_EGY_CTL_ENA_THD_MIN	1	0...FFFFH	0...255	0.00389105	mJ
Lower energy threshold to activate controller					
C_EGY_CTL_I_GAIN	1	0...FFFFH	0...1.99996948	3.05176E-5	%/mJ
Integral gain for piezo energy controller					
C_EGY_CTL_P_GAIN	1	0...FFFFH	0...1.99996948	3.05176E-5	%/mJ
proportional gain for energy controller					
C_EGY_INJ_MAX_LIM	1	0...FFFFH	0...255	0.00389105	mJ
Maximum value for voltage dependent injector energy limiter					
C_EGY_STEP_INJ_CHA_GRD[NC_CYL_NR]	1	0...4000H	0...100	0.00610352	%
Piezo injector energy charge and discharge gradient for a single step					
C_FAC_COR_EGY_STEP_INJ_DCHA_GRD	1	0...FFFFH	0...1.99996948	3.05176E-5	-
Multiplicative correction factor for cylinder balancing influence on discharge gradient calculation					
C_FAC_EGY_PWM_AD_MAX	1	0...FFFFH	0...1.99996948	3.05176E-5	%/mJ
Maximal value of conversion factot EGY to % PWM					
C_FAC_EGY_PWM_AD_MIN	1	0...FFFFH	0...1.99996948	3.05176E-5	%/mJ
Minimal value of conversion factot EGY to % PWM					
C_FAC_EGY_PWM_MAN[NC_CYL_NR]	1	0...FFFFH	0...1.99996948	3.05176E-5	%/mJ
Manually set conversion factor: EGY to % PWM for piezo injector					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_LPF_EGY_PWM_AD	1	0...FFH	0...0.99609375	0.00390625	-
Factor for PT1 - Filter for calculation of FAC_EGY_PWM_AD					
C_MAX_INJ_EGY_LEVEL_COR	1	E000...2000H	-50...50	0.00610352	%
Maximum correction value of injector energy level					
C_MIN_INJ_EGY_LEVEL_COR	1	E000...2000H	-50...50	0.00610352	%
Minimum correction value of injector energy level					
C_NR_EGY_STEP_INJ_H	1	2...1FH	2...31	1	-
Number of charge steps for a high piezo energy range					
C_NR_EGY_STEP_INJ_L	1	2...1FH	2...31	1	-
Number of charge steps for a low piezo energy range					
C_PWM_LIM_V_MAX_I_GAIN	1	0...FFFFH	0...1.99996948	3.05176E-5	mJ/V
Integral gain for PWM limiting related to stack voltage					
C_SEG_CTR_EGY_AD_ENA	1	0...FFFFH	0...6.5535E+4	1	-
Number of segments for activation of piezo energy adaptation					
C_TECU_MIN_EGY_CTL_ENA	1	0...FEH	-48...142.5	0.75	°C
Minimum ECU temperature to activate piezo injector energy controller.					
C_TEMP_MIN_EGY_CTL_ENA	1	0...FEH	-48...142.5	0.75	°C
Minimum selected temperature to activate piezo injector energy controller.					
C_TEMP_SEL_EGY_CTL	1	0...FFH	0...255	1	-
Switch to select temperature for energy controller functions; 0: TEMP_CAPA_IV_MV, 1: TFU_INJ, 2: TCO;					
C_TI_EGY_CTL_CLC_MIN	1	0...FFFFH	0...65.535	0.001	ms
Minimum injection time limit for calculation of energy control					
LC_EGY_ADD_AD_LIM_ENA	1	0...1H	0...1	1	-
Enable blocking of adaption if additive corrections (EGY_ADD_AD) are limited					
LC_EGY_SP_IV_EXT_ENA	1	0...1H	0...1	1	-
Logical constant to enable external energy setpoint for injector energy controller					
LC_EGY_STEP_CTL_DYN_ENA	1	0...1H	0...1	1	-
Enable complete piezo energy controller					
LC_ENA_ADAP_EGY_CTL	1	0...1H	0...1	1	-
Enable new adaptation of piezo energy controller values after engine start					
IP_V_IV_MAX	6	0...7FFFH	0...639.980469	0.01953125	V
LDP_TFU_INJ_IP_V_IV_MAX	6	0...FEH	-48...142.5	0.75	°C
Maximum allowed voltage on piezo stack before limiting					
IP_EGY_STEP_INJ_CHA_GRD_OFS	6x4	0...4000H	-50...50	0.00610352	%
LDP_FUP_IP_EGY_STEP_INJ_CHA_GRD	6	0...FFFFH	0...3.47776E+5	5.3067216	hPa
LDP_TFU_INJ_IP_EGY_STEP_INJ_CHA	4	0...FEH	-48...142.5	0.75	°C
Piezo injector energy charge and discharge gradient offset for a single step					

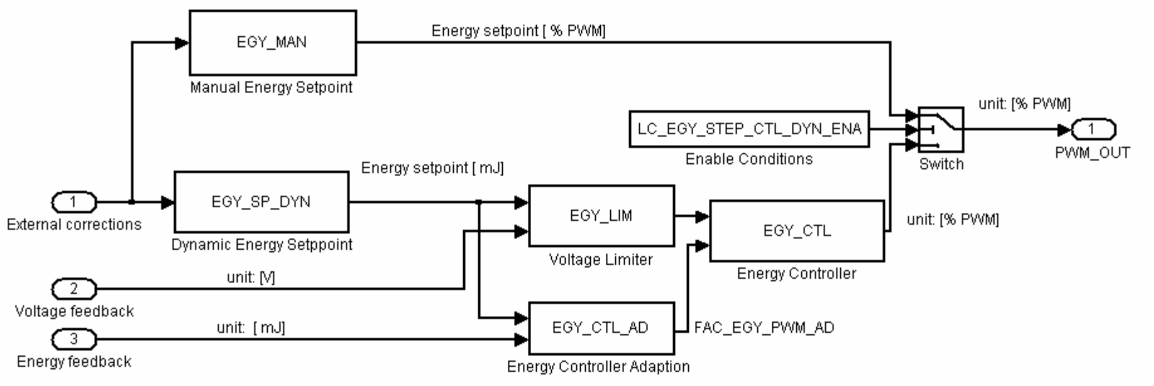
Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_EGY_RNG_H	1	0...1H	0...1	1	-
Symbolic constant for usage of high energy range of PIEZO injection valve					

Exported actions:

ACTION_INJR_**SetFacEgyPwmAd** (IN <Cyl>, IN <Fac>)
Sets the cylinder individual conversion factor FAC_EGY_PWM_AD of cylinder <Cyl> to value <Fac>.

41.27.1 Controller concept overview



There are several modes to operate the piezo energy controller unit.

If **LC_EGY_STEP_CTL_DYN_ENA** is FALSE, the manual set point calculation with direct feed through to the PWM output is active.

Otherwise the dynamic set point calculation and voltage limiter are activated.

For the voltage limiter are proper feedback conditions necessary.

The adaption of the conversion factor FAC_EGY_PWM_AD is calculated, if switched ON only, and if proper feedback is granted.

Value of NC_EGY_RNG_H = 1.

41.27.1.1 Description for ACTION_INJR_SetFacEgyPwmAd

ACTION_INJR_SetFacEgyPwmAd (IN <Cyl>, IN <Fac>)					
Sets the cylinder individual conversion factor FAC_EGY_PWM_AD of cylinder <Cyl> to value <Fac>.					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
Cyl	IN	0...7 H	0...7	1	1
This action affects the conversion factor FAC_EGY_PWM_AD of cylinder Cyl.					
Fac	IN	0...FFFFH	0...1.99996948	3.05176 E-5	%/mJ
This parameter defines the new value of FAC_EGY_PWM_AD.					

Formula Section for ACTION_INJR_SetFacEgyPwmAd:

FAC_EGY_PWM_AD[Cyl] = Fac

Application Condition

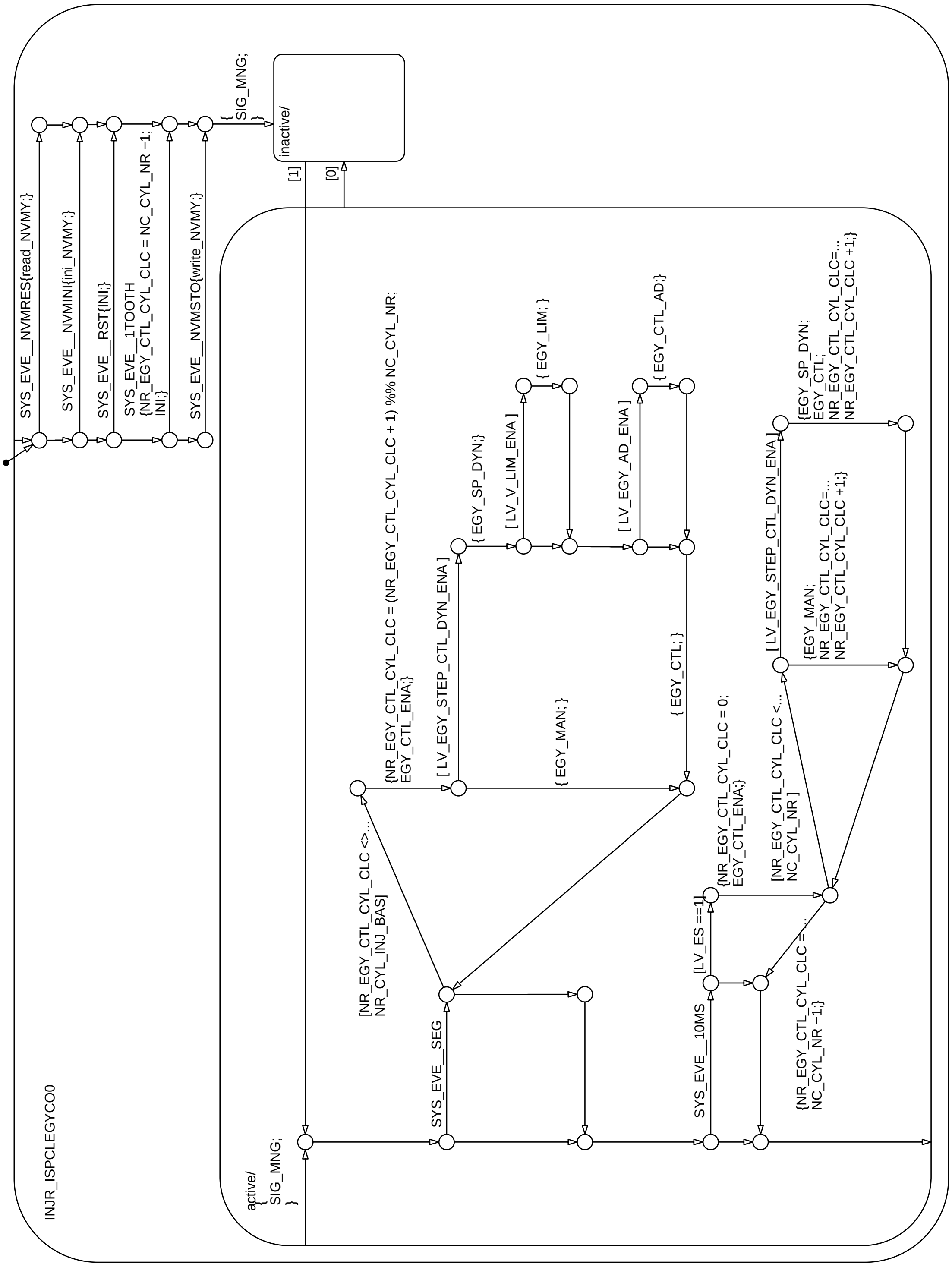
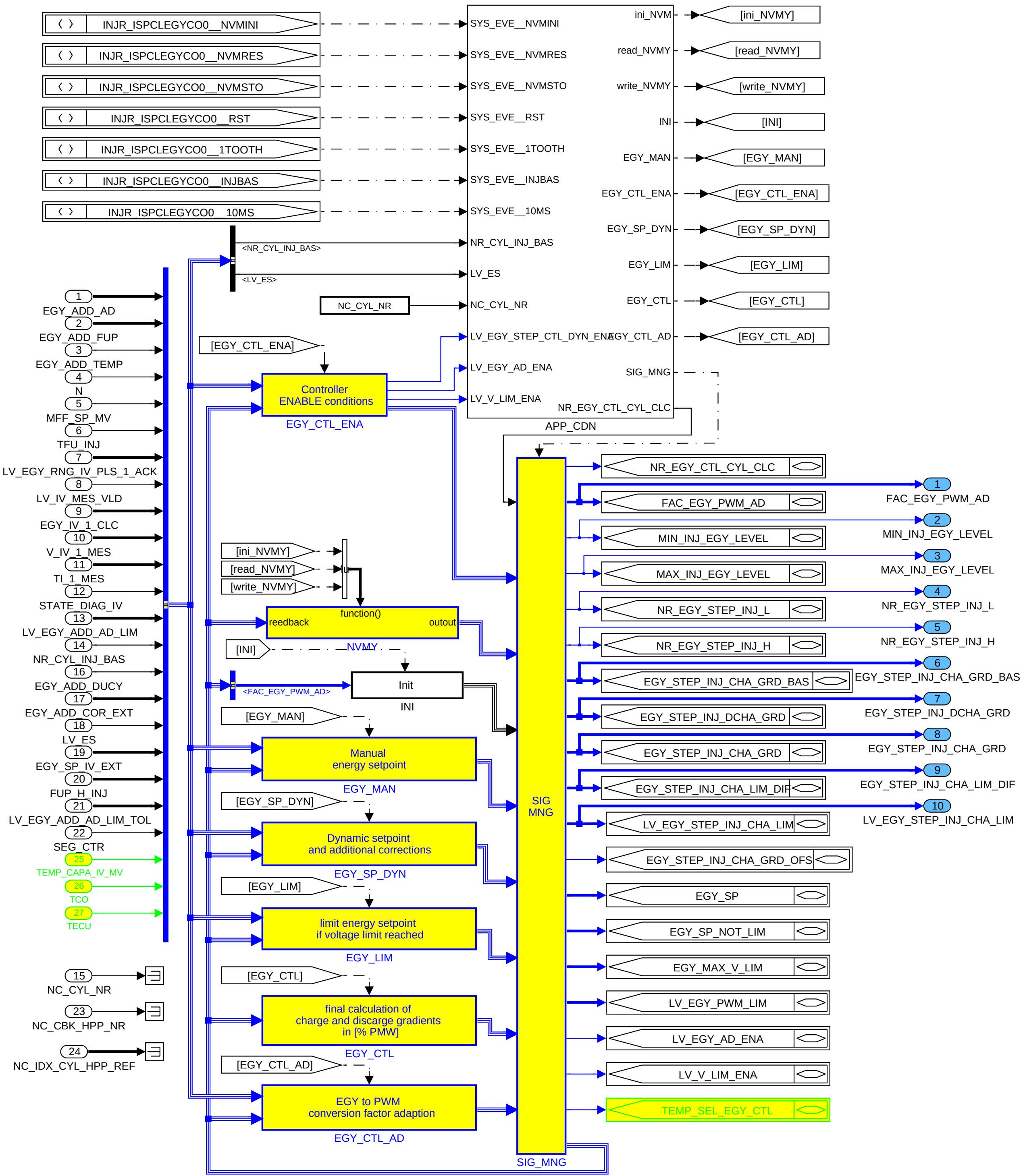


Figure 22 INJR_ISPCLEGYCO0/ APP_CDN/ Chart



SDA_SRS / SDA 4.0 17–Jan–2007

Figure 23 INJR_ISPCLEGYCO0

41.27.1.2 Handling of nonvolatile memory

The conversion factor FAC_EGY_PWM_AD is stored in the nonvolatile memory. It is read and written to the NVMY before/after each complete driving cycle. The NVMY is initialized with C_FAC_EGY_PWM_MAN.

41.27.1.3 Initialization

Do all initializations at RESET

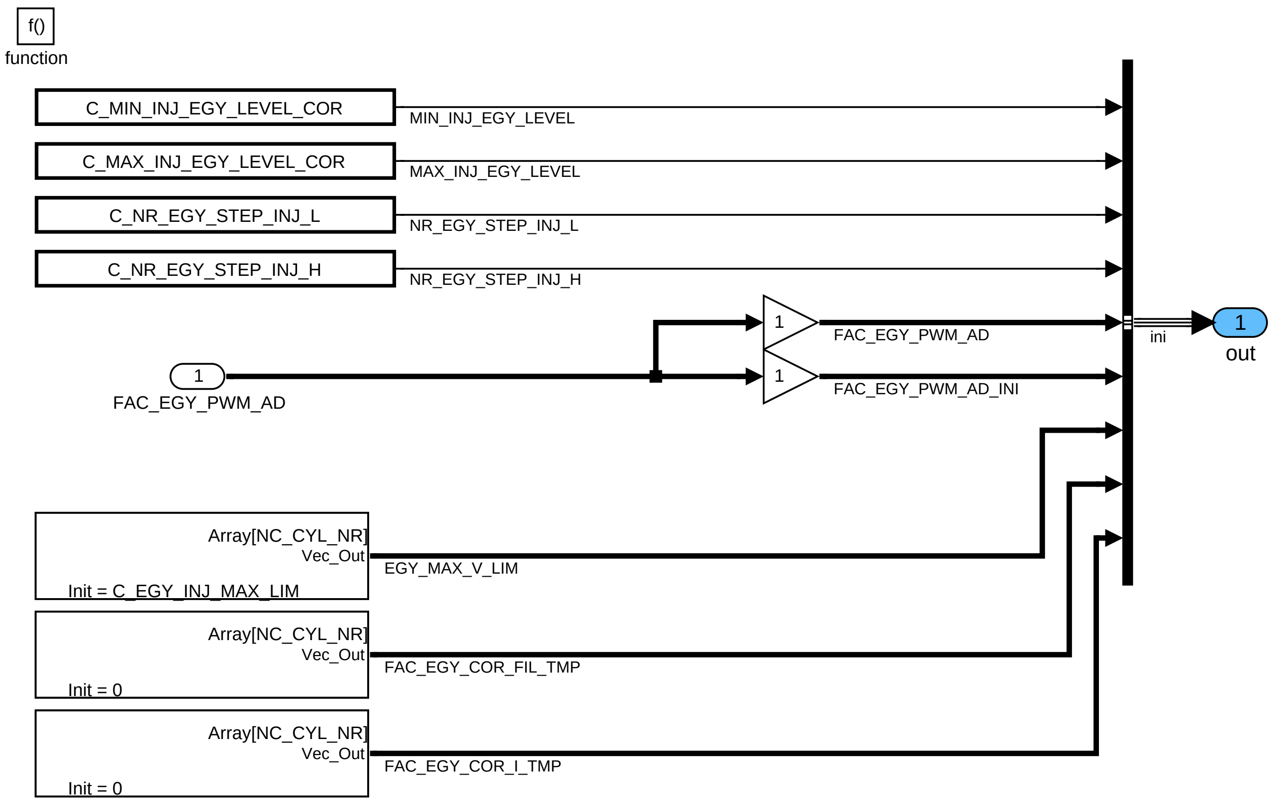


Figure 24 INJR_ISPCLEGYCO0/ INI

41.27.1.4 SUBFUNCTION: EGY_MAN

<u>Manual set point calculation and direct PWM signal output</u>

If dynamic control is not enabled the manual set point is written directly to PWM outputs

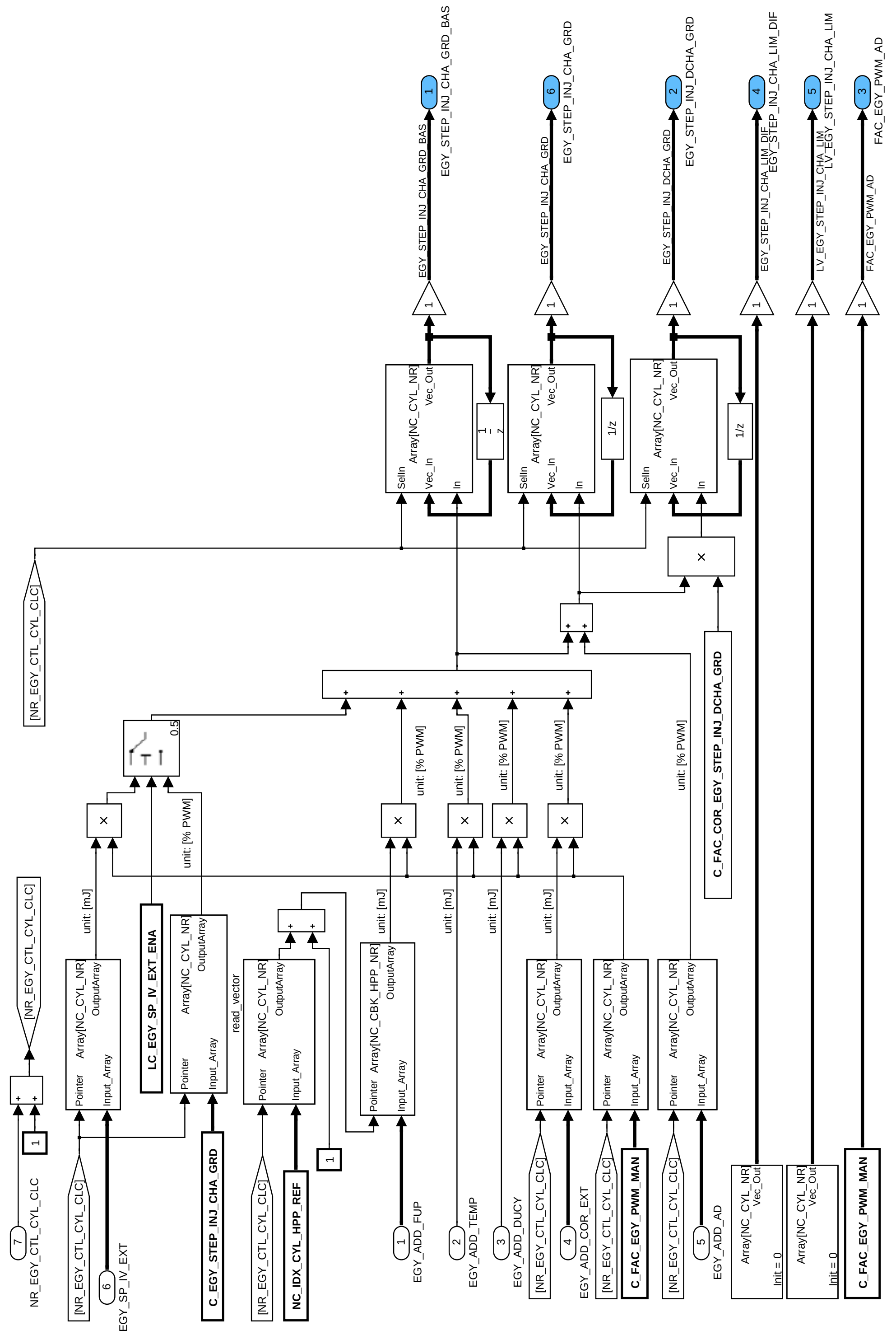


Figure 25 INJR_ISPCLEGYCO0/ EGY_MAN/ EGY_SP_MAN

41.27.1.5 SUBFUNCTION: EGY_CTL_ENA

ENABLE conditions for energy controller and controller gain adaption

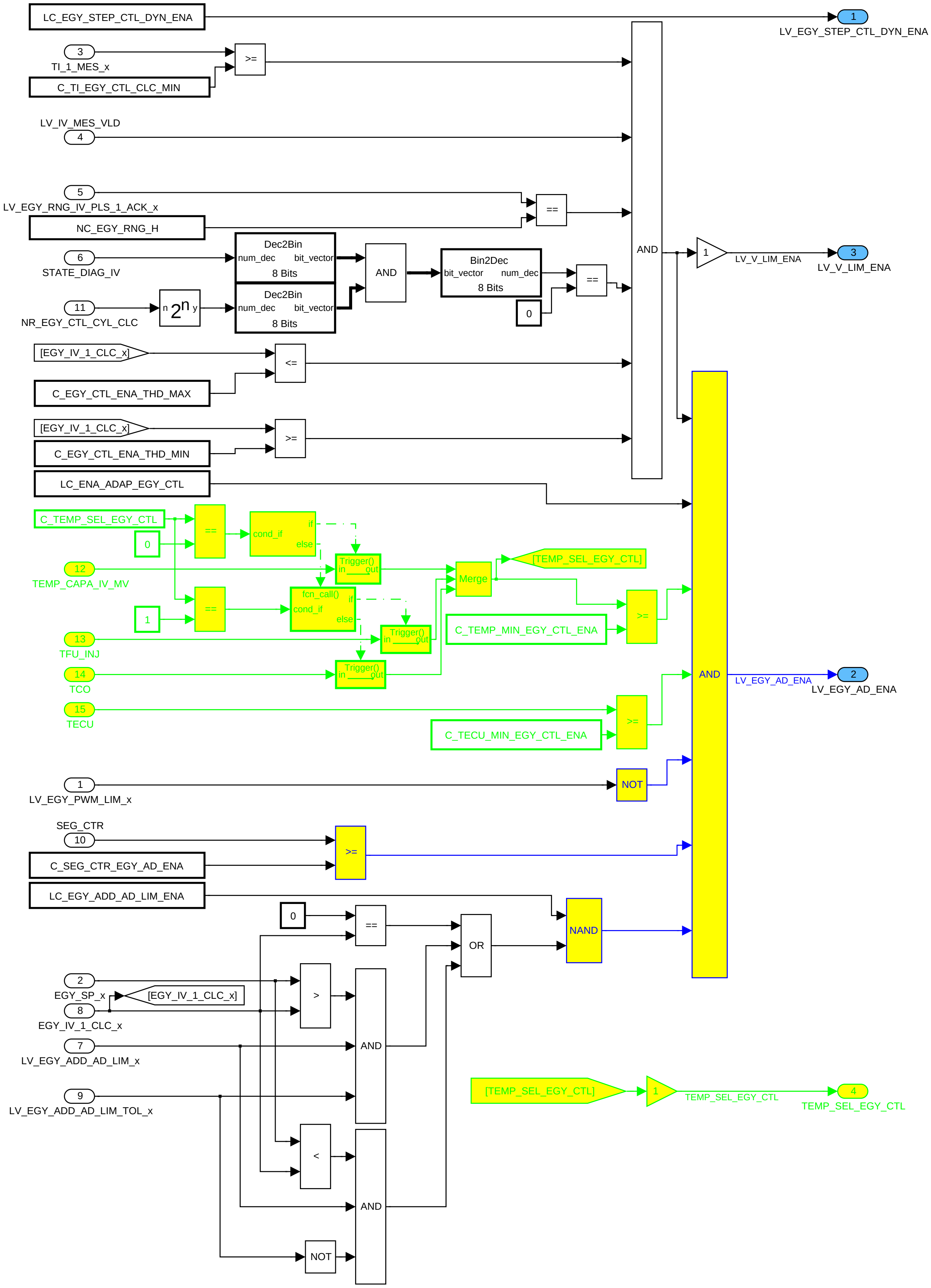


Figure 26 INJR_ISPCLEGYCO0/ EGY_CTL_ENA/ EGY_CTL_ENA_x

41.27.1.6 SUBFUNCTION: EGY_SP_DYN

Dynamic Energy set point calculation with additional correction

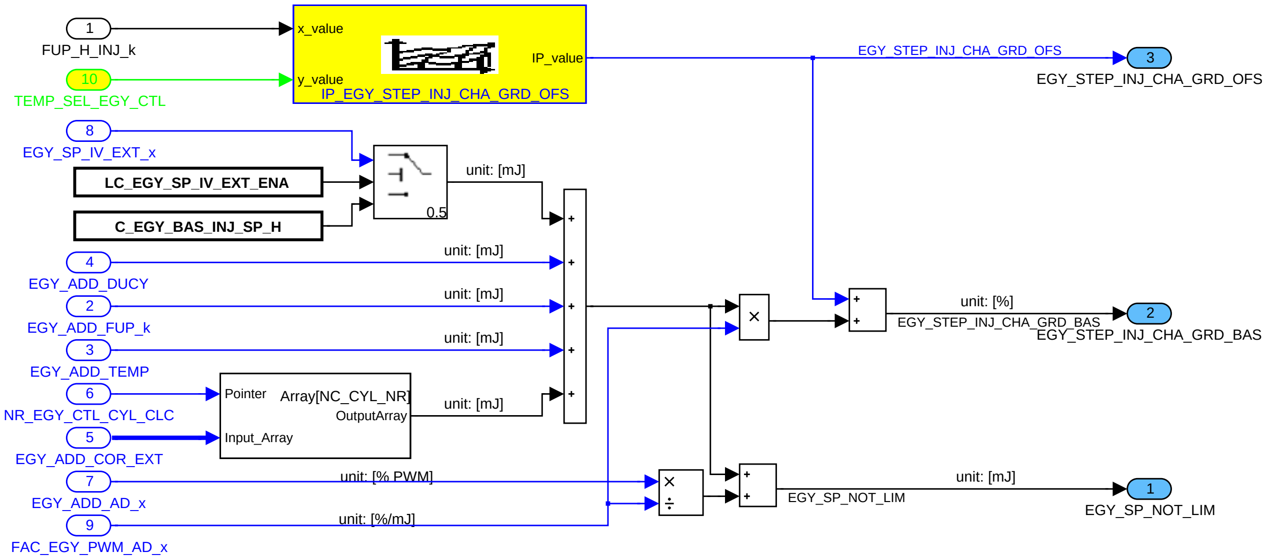


Figure 27 INJR_ISPCLEGYCO0/ EGY_SP_DYN/ EGY_SP_DYN_x

41.27.1.7 SUBFUNCTION: EGY_LIM

Limitation of energy set point if stack voltage limit is reached

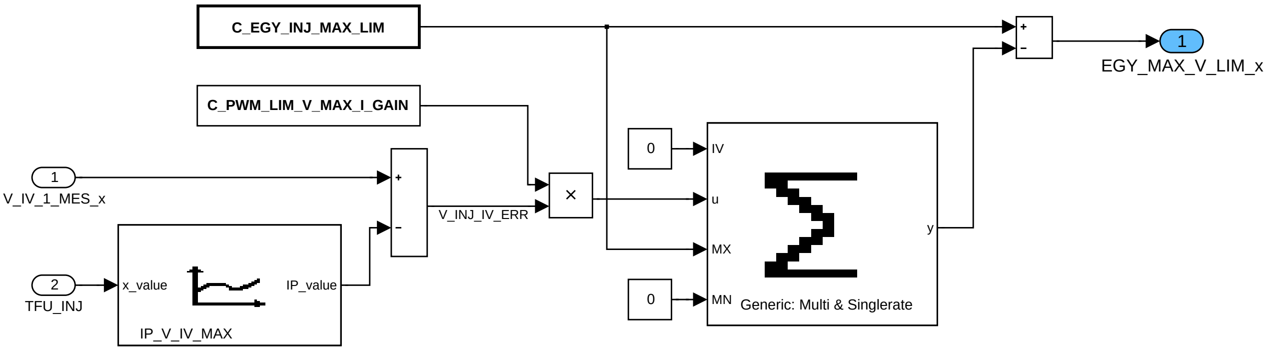


Figure 28 INJR_ISPCLEGYCO0/ EGY_LIM/ EGY_LIM_x

41.27.1.8 SUBFUNCTION: EGY_CTL

Energy controller for direct calculation of PWM signal out of energy set point

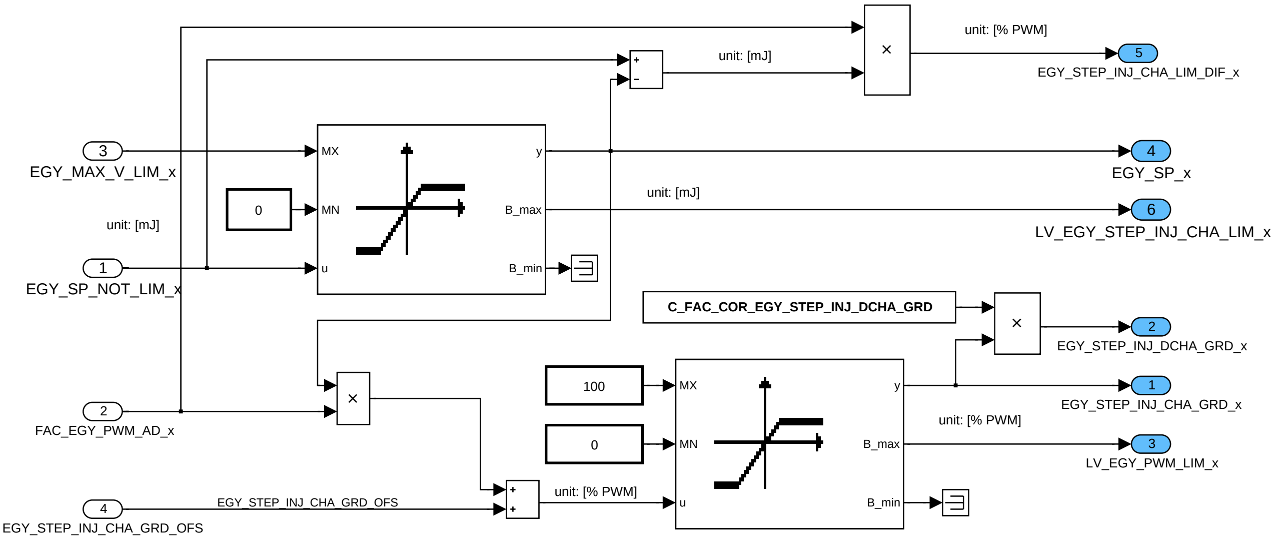


Figure 29 INJR_ISPCLEGYCO0/ EGY_CTL/ EGY_CTL_x

41.27.1.9 SUBFUNCTION: EGY_CTL_AD

Adaption of energy to PWM conversion factor FAC_EGY_PWM_AD

At RESET the factor is set to the stored value from the non volatile memory or to '1' if the stored value is not valid.

At Engine switch off the last learned value is stored to the NVMY.

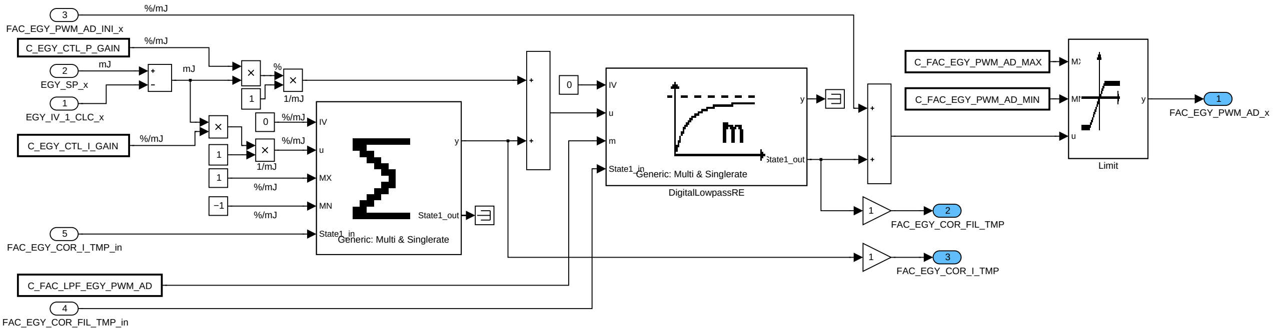


Figure 30 INJR_ISPCLEGYCO0/ EGY_CTL_AD/ EGY_CTL_AD_x

41.28 Injector needle lift control (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EGY_ADD_COR_EXT[NC_CYL_NR]	O/V	8000...7FFFH	-127.501946 ... 127.498054	0.00389105	mJ
External additive piezo energy correction					
EGY_SP_IV_EXT[NC_CYL_NR]	O/V	0...FFFFH	0...255	0.00389105	mJ
External injection valve EGY setpoint value for energy controller					
DIST_IV_CHG[NC_CYL_NR]	O/V/S	0...FFFFH	0...5.2428E+5	8	km
Cylinder individual distance accumulation [8 km] at injector change					

Input data:

LV_ST_END	N	MFF_SP_MV	STATE_INJ_MOD_REQ
NC_CYL_NR	DIST_KWP	EGY_SP_IV_EXT_ADJ[NC_CYL_NR]	STATE_IV_CHG
EGY_LAM_ADJ_COR_LAM_AD_CUS[NC_CYL_NR]	LF_ERR_PLAUS_IV_EGY_CAL		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EGY_SP_IV_EXT_ADJ[NC_CYL_NR]	1	0...FFFFH	0...255	0.00389105	mJ
Default value for external energy setpoint for piezo injector energy controller calculation					
LC_EGY_SP_IV_EXT_ADJ_ENA	1	0...1H	0...1	1	-
Logical constant to enable external energy setpoint calculation for piezo injector energy controller					
IP_EGY_ADD_IV_AGI_DIST	16	0...FFFFH	-127.501946 ... 127.498054	0.00389105	mJ
LDPM_DIST_IP_EGY_ADD_IV_AGI	16	0...FFFFH	0...5.2428E+5	8	km
Additive piezo energy depending on accumulated driving distance for compensation of aging effects					
IP_EGY_ADD_COR_EXT_HOM	8x8	0...FFFFH	-127.501946 ... 127.498054	0.00389105	mJ
LDPM_N_IP_EGY_ADD_COR_EXT	8	0...1FE0H	0...8.16E+3	1	rpm
LDPM_MFF_SP_IP_EGY_ADD_COR_EXT	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Additive piezo injector energy correction depending on engine speed and load, HOM mode					
IP_EGY_ADD_COR_EXT_HOMS	8x8	0...FFFFH	-127.501946 ... 127.498054	0.00389105	mJ
LDPM_N_IP_EGY_ADD_COR_EXT	8	0...1FE0H	0...8.16E+3	1	rpm
LDPM_MFF_SP_IP_EGY_ADD_COR_EXT	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Additive piezo injector energy correction depending on engine speed and load, HOMS mode					
IP_EGY_ADD_COR_EXT_S	8x8	0...FFFFH	-127.501946 ... 127.498054	0.00389105	mJ
LDPM_N_IP_EGY_ADD_COR_EXT	8	0...1FE0H	0...8.16E+3	1	rpm
LDPM_MFF_SP_IP_EGY_ADD_COR_EXT	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Additive piezo injector energy correction depending on engine speed and load, S mode					

41.28.1 General information

This module generates an external cylinder individual set point for the piezo energy controller. EGY_SP_IV_EXT. Furthermore an additive piezo energy correction is supplied. It is taken into account for the energy set point calculation.

Application Condition

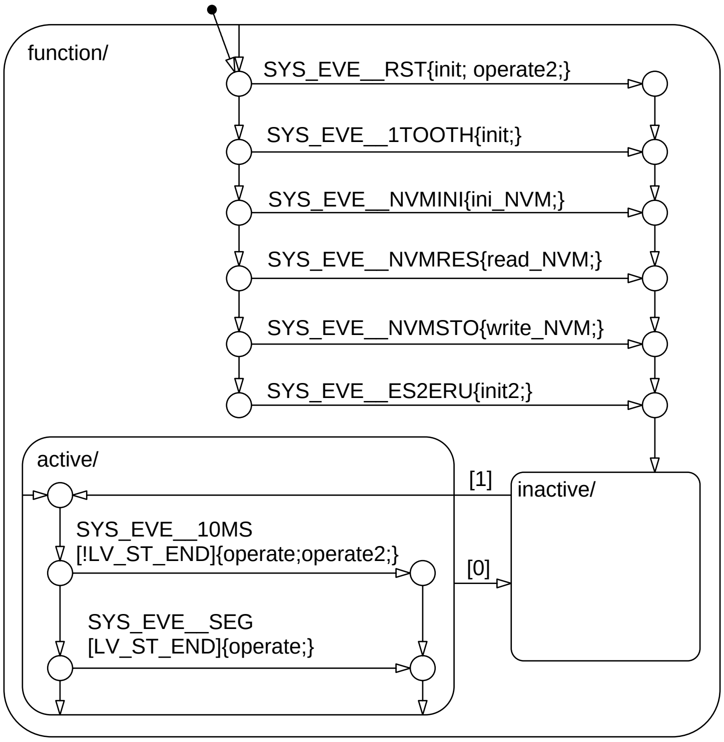


Figure 31 INJR_ISPCLEGYAI0/ APP_CDN/ Chart

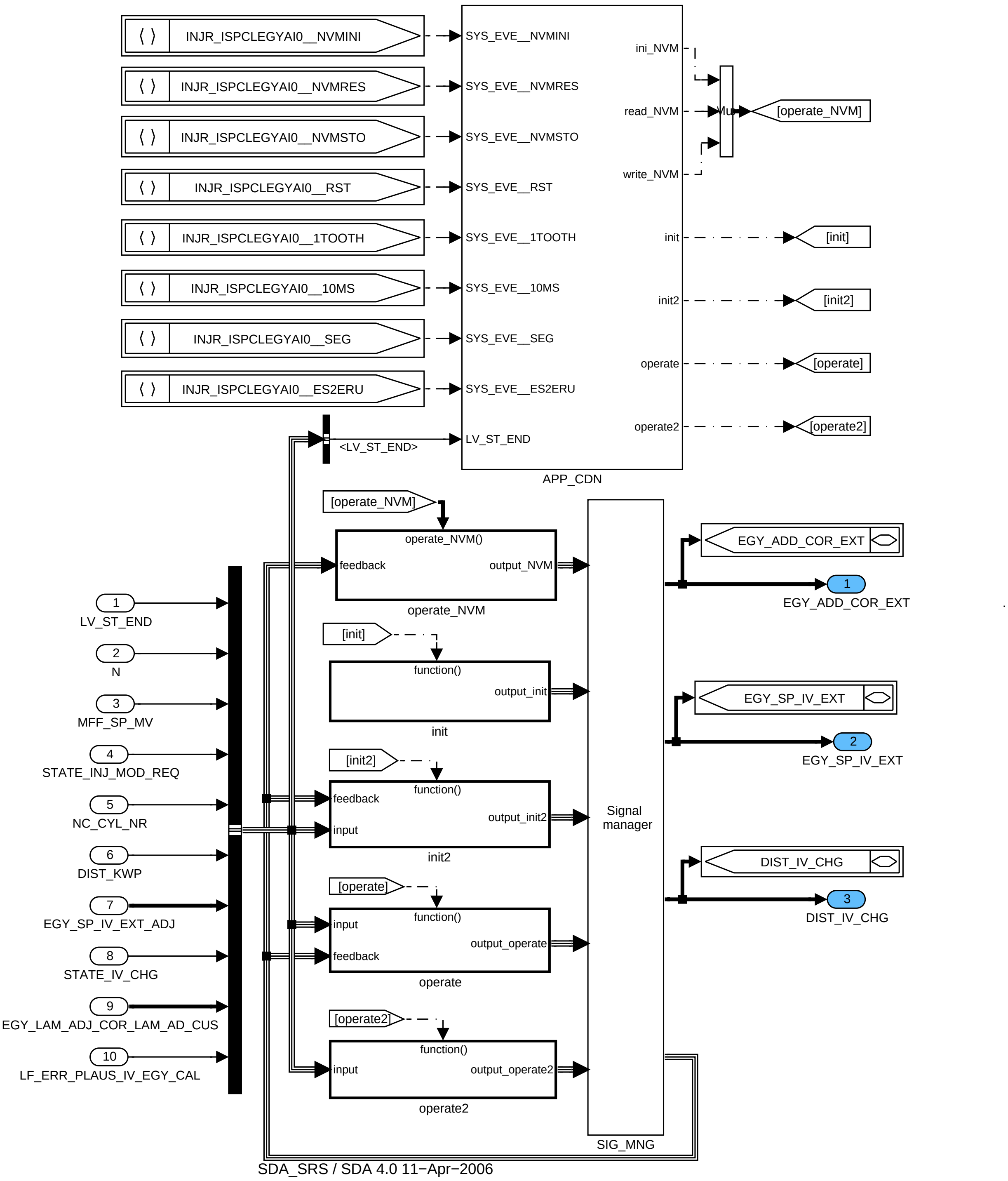


Figure 32 INJR_ISPCLEGYAI0

41.28.1.1 Nonvolatile memory operations

The accumulated driving distance at injector change DIST_IV_CHG is stored in the nonvolatile memory (NVM). It is read from and written to the NVM before/after each complete driving cycle.

Please note: DIST_IV_CHG has to be allocated in that part of the NVM-RAM which is not erased during reflashing of ECU.

41.28.1.2 Initialization at reset and first valid tooth

At reset and fist valid tooth the additive energy correction is set to 0.

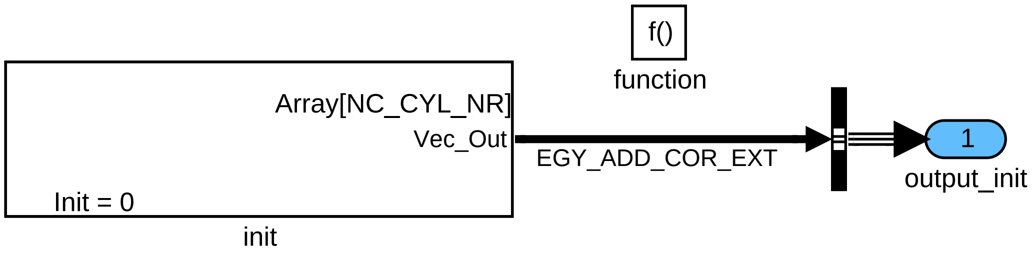


Figure 33 INJR_ISPCLEGYAI0/ init

41.28.1.3 Calculation at engine stop to engine run

At ES2RUN it is determined whether an injection valve was changed: STATE_IV_CHG is provided by the keyword-protocol KWP and supplies a bit coded information: If the bit on position x is set the injector was changed. DIST_IV_CHG is the cylinder individual driving distance when the injector was changed. It is stored in the nonvolatile memory.

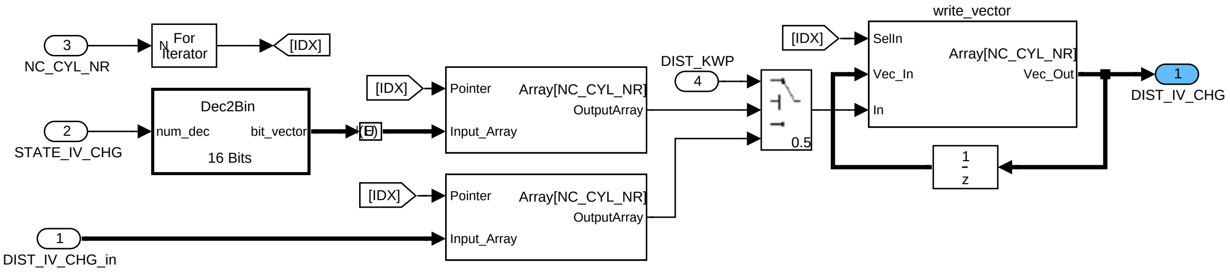


Figure 34 INJR_ISPCLEGYAI0/ init2/ ForIterator

41.28.1.4 Calculation of additive energy correction

An additive piezo energy correction depending on engine speed and load is supplied. Depending on the requested combustion modes (homogeneous, homogeneous-stratified and stratified) different maps are used. Furthermore an additive correction accounting for aging effects of the piezo injection valve is supplied cylinder individually. The accumulated driving distance is used as an indicator for aging. It is given by the difference of the actual driving distance and the driving distance at the last injector change. EGY_LAM_ADJ_COR_LAM_AD_CUS accounts for the long term lambda adaptation.

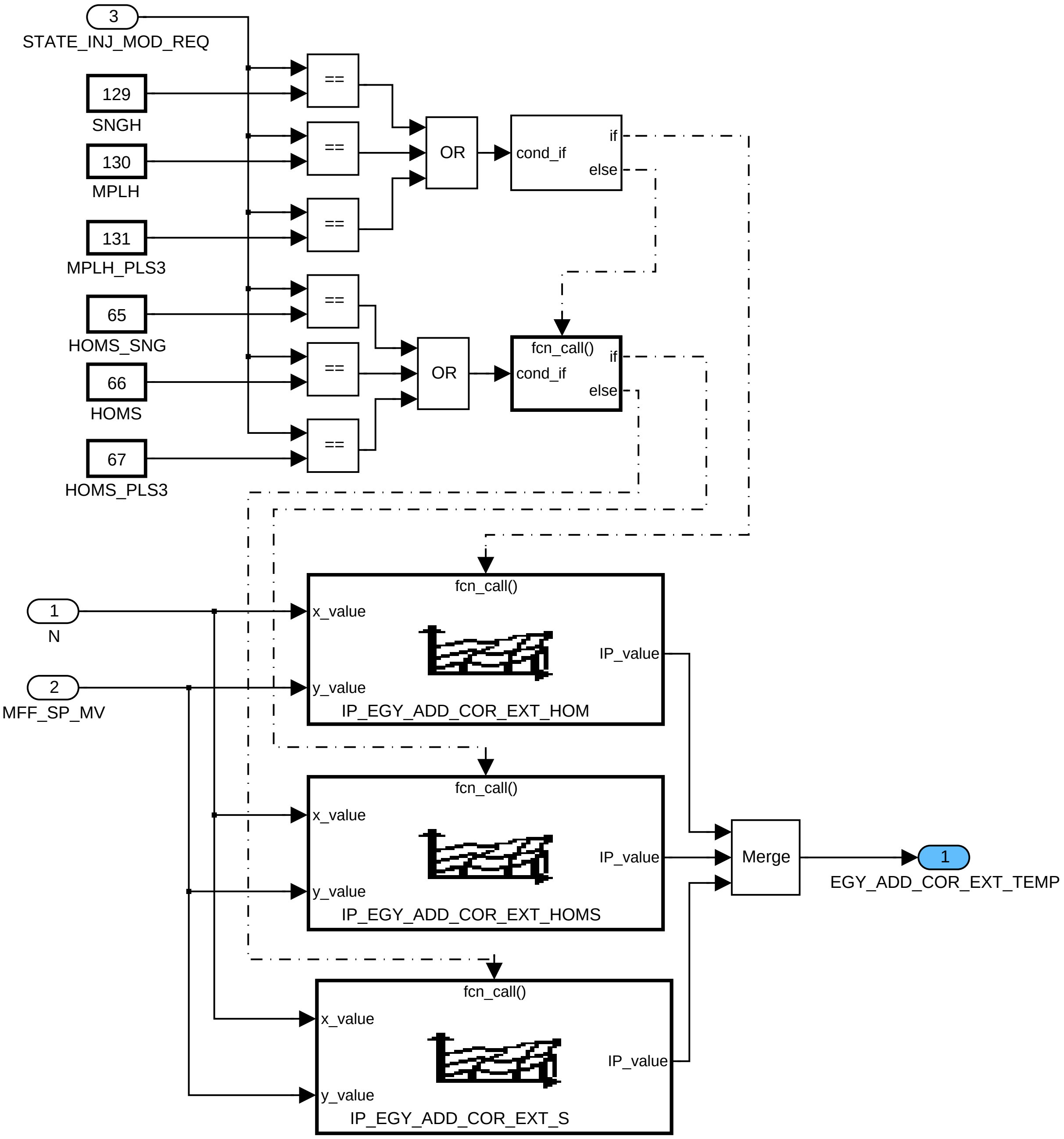


Figure 35 INJR_ISPCLEGYAI0/ operate/ CLC_EGY_ADD_COR_EXT

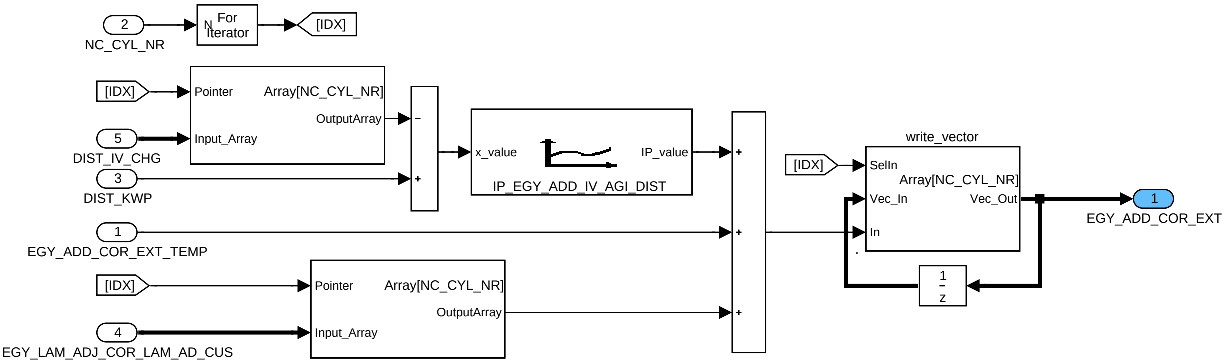


Figure 36 INJR_ISPCLEGYAI0/ operate/ ForIterator

41.28.1.5 External energy setpoint for energy controller

This functionality supplies a cylinder individual injection valve energy set point for the energy controller. It is set to EGY_SP_IV_EXT_ADJ given by the KWP if the cylinder individual values are plausible (bit x is zero in bit-mask LF_ERR_PLAUS_IV_EGY_CAL for logical cylinder x) and if LC_EGY_SP_IV_EXT_ADJ_ENA is set. Otherwise cylinder individual values can be supplied manually via the calibration constant C_EGY_SP_IV_EXT_ADJ.

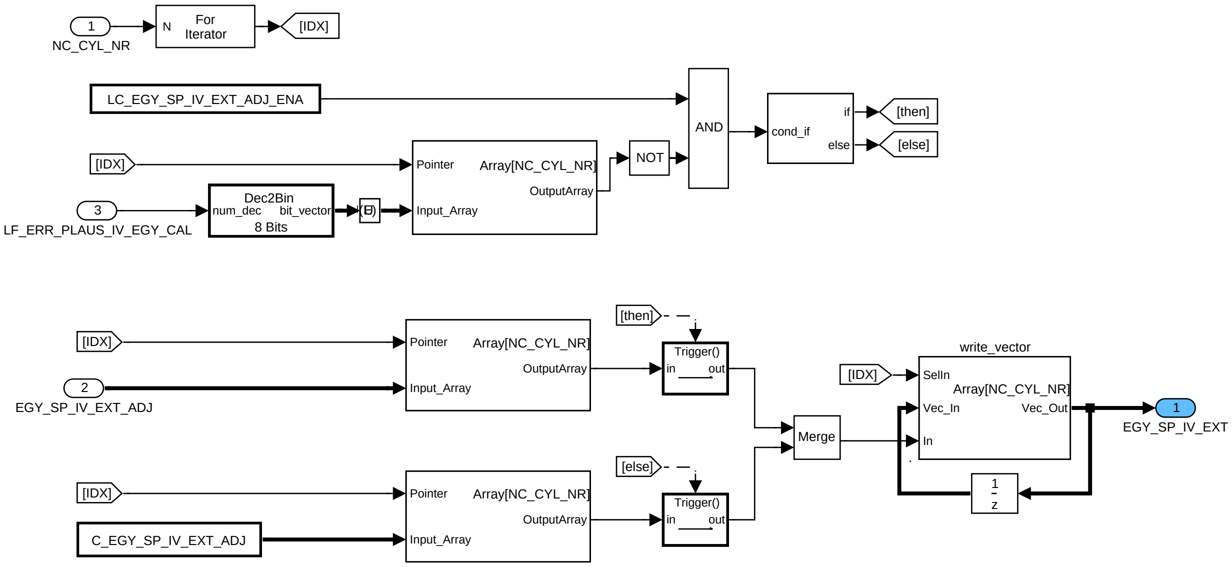


Figure 37 INJR_ISPCLEGYAI0/ operate2/ Calc_of_EGY_SP/ For_Iterator

41.29 Cylinder Shut Off

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
INH_INJ	V/O	0...FFH	0...255	1	[-]
Final cylinder shut off pattern (fixed cylinder allocation)					
LV_FCUT_IND	V/O	0...1H	0...1	1	[-]
At least one cylinder is shut off					
LV_INJ_CUT	V/O	0...1H	0...1	1	[-]
All cylinders shut-off					
LV_SCC[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag that indicates static single cylinder shut off, exhaust cylinder bank individual					
INH_IV	V/O	0...FFH	0...255	1	[-]
Shut off pattern for static cylinder shut off (fixed cylinder allocation)					
INH_SWI_IV	V/O	0...FFFFH	0...65535	1	[-]
Shut off pattern for dynamic cylinder shut off					
SUM_INH_IV	V/O	0...8H	0...8	1	[-]
Sum of INH_IV					
SUM_INH_IV_CBK[NC_CBK_EX_NR]	V/O	0...8H	0...8	1	[-]
Sum of those bits within INH_IV which are allocated to exhaust cylinder bank i					
SUM_INH_IV_DYN	V/O	0...8H	0...8	1	[-]
Sum of INH_IV_DYN					
SUM_INH_INJ	V/O	0...8H	0...8	1	[-]
Sum of INH_INJ					
NR_PAT	V/O	0...FFH	0...255	1	[-]
Index corresponding to the highest shut off level (fuel shut off with pattern)					
NR_PAT_OLD	-	0...FFH	0...255	1	[-]
Index corresponding to the highest shut off level (fuel shut off with pattern) from previous calculation run					
STATE_INH_IV_DYN	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
State of the dynamic cylinder shut off request					
INH_IV_IGK	V	0...FFH	0...255	1	[-]
Shut off pattern of IGK					
INH_IV_PUC	V	0...FFH	0...255	1	[-]
Shut off pattern of pull fuel shut off					
INH_IV_MON	V	0...FFH	0...255	1	[-]
Shut off pattern of shut off request by monitoring concept					
INH_IV_N_MAX_REQ_FCUT	V	0...FFH	0...255	1	[-]
Shut off pattern of LV_N_MAX_REQ_FCUT (engine speed limitation)					

Input data:

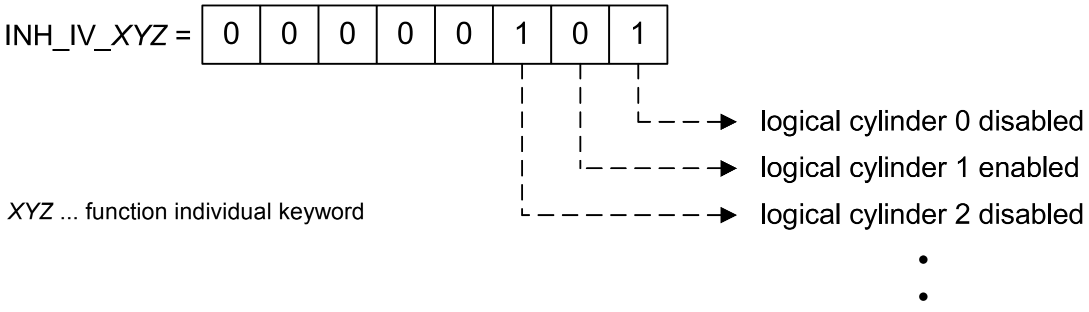
INH_IV_MIS	INH_IV_EXT	INH_IV_IGC	NC_CBK_EX_NR
NR_PAT_SEL_CYL	NR_PAT_SCC	LV_N_MAX_REQ_FCUT	LV_PUC
LV_OFF_IV_MON	LV_SEL_CYL	LV_IGK	INH_IV_DYN
LV_ST_END	NC_LAMB_REF	NC_INJ_INH_SWI_IV_SHIFT_NR	NC_CYL_NR
LV_ERR_TMP_MU_MC	LV_INJ_PUC_ENA	LF_IV_INH_PUC_EXT	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_INH_SWI_IV	17	0...FFFFH	0...65535	1	[-]
LDP_NR_PAT_ID_INH_SWI_IV	17	0...FFH	0...255	1	[-]
Cylinder shut off pattern for dynamic cylinder shut off					
LC_SWI_MOD_INH_IV_DYN	1	0...1H	0...1	1	[-]
Mode switch between different dynamic fuel shut off algorithm					

41.29.1 General information

Static fuel shut off:
In several engine operating- and system states and in case of malfunctions of the injection or ignition system the fuel injection must be disabled at individual cylinders. All the shut off requirements are coordinated, hence these requirements have to have the same structure as described below:



Dynamic fuel shut off:
The information which cylinder has to be shut off static, is coded within a byte (with the length of 8 bit) at the cylinder corresponding position.
In some engine operating states, e. g. fast torque reduction, restart fuel feed or pull fuel cut off, the injection has to be disabled for a certain subsequent numbers of injections. The fuel cut off starts immediately at the next cylinder following an order defined in a pattern. That means, different to the cylinder individual static fuel shut off, there is no predefined and fixed association between cylinders and the shut off sequence.
Injections, which are already started, will not be stopped.

FUNCTION DESCRIPTION:

Application conditions:

Activation:	every engine state
Deactivation:	-
Initialization:	-
Reccurence:	if LV_ST_END = 0: 10 ms if LV_ST_END = 1: segment synchronous

41.29.2 Static fuel shut off

41.29.2.1 Ignition Key

IF LV_IGK = 0

THEN all cylinders are shut off

INH_IV_IGK =	1	1	1	1	1	1	1	1

ELSE no cylinder is shut off

INH_IV_IGK =	0	0	0	0	0	0	0	0

ENDIF

41.29.2.2 Ignition System

INH_IV_IGC (Input pattern)

41.29.2.3 Monitoring Concept

IF (LV_OFF_IV_MON = 1) OR (LV_ERR_TMP_MU_MC = 1)

THEN all cylinders are shut off

INH_IV_MON =	1	1	1	1	1	1	1	1

ELSE no cylinder is shut off

INH_IV_MON =	0	0	0	0	0	0	0	0

ENDIF

41.29.2.4 Misfire Detection

INH_IV_MIS (Input pattern)

41.29.2.5 Pull Fuel Cut Off

To guarantee a moderate transition from trailing throttle PU to trailing throttle fuel cut off, the transition is handled by the module “Sequential fuel cut off and restart fuel feed”, see chapter Cylinder Individual Fuel Shut Off with Pattern. After the transition is finished (indicated by the flag LV_SEL_CYL) all cylinders are shut off by INH_IV_PUC.

```
IF      LV_PUC = 1
        AND
        LV_SEL_CYL = 0

THEN
        IF LV_INJ_PUC_ENA = 1
        THEN
                INH_IV_PUC = LF_IV_INH_PUC_EXT
        ELSE

                all cylinders are shut off
                INH_IV_PUC = | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
        ENDIF

ELSE    no cylinder is shut off
        INH_IV_PUC = | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
ENDIF
```

41.29.2.6 Engine speed limitation

```
IF      LV_N_MAX_REQ_FCUT = 1

THEN    all cylinders are shut off
        INH_IV_N_MAX_REQ_FCUT = | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

ELSE    no cylinder is shut off
        INH_IV_N_MAX_REQ_FCUT = | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
ENDIF
```

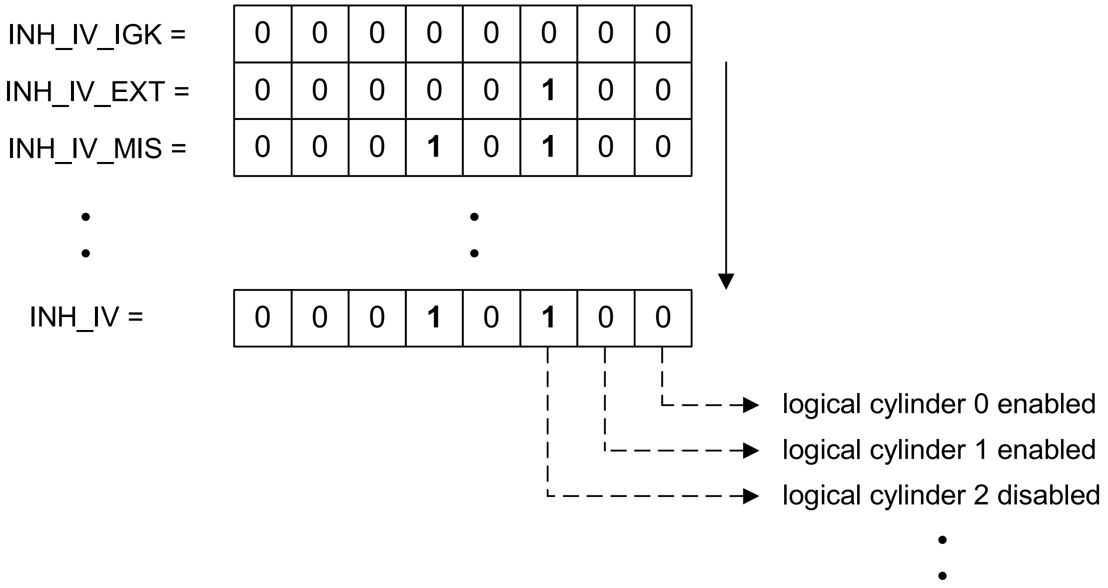
41.29.2.7 Shut Off Pattern from Application Incidences

INH_IV_EXT (Input pattern)

Additional shut off requirements are coordinated within the module “Cylinder Shut Off (Application Incidences)”.

41.29.2.8 Coordination of the Static Fuel Shut Off Requirements

The output pattern INH_IV is the bitwise OR of all pattern, linked with a mask $((2^{NC\_CYL\_NR})-1)$. As an example see below:



41.29.3 Dynamic fuel shut off

In some engine operating states, e. g. fast torque reduction, restart fuel feed or pull fuel cut off, the injection has to be disabled for a certain subsequent numbers of injections. The fuel cut off starts immediately at the next cylinder following an order defined in a pattern. That means, different to the cylinder individual static fuel shut off, there is no predefined and fixed association between cylinders and the shut off sequence.

Injections, which are already started, are not stopped.

The map ID_INH_SWI_IV contains (2 * NC_CYL_NR + 1) different pattern with calibrate able shut off sequences. Every pattern is allocated to an index NR_PAT (mapping points).

There are two requirements for “Dynamic fuel shut off” to this module:

NR_PAT_SEL_CYL is an index defined by the module “Sequential fuel cut off and restart fuel feed”.

NR_PAT_SCC is an index defined by the module “Torque based pattern calculation”.

NR_PAT is the index corresponding to the highest shut off level (number of cylinders disabled) of these both requirements. A certain pattern is repeated until this index changes.

NR_PAT_OLD = NR_PAT

NR_PAT = MAX (NR_PAT_SEL_CYL, NR_PAT_SCC)

INH_SWI_IV = ID_INH_SWI_IV

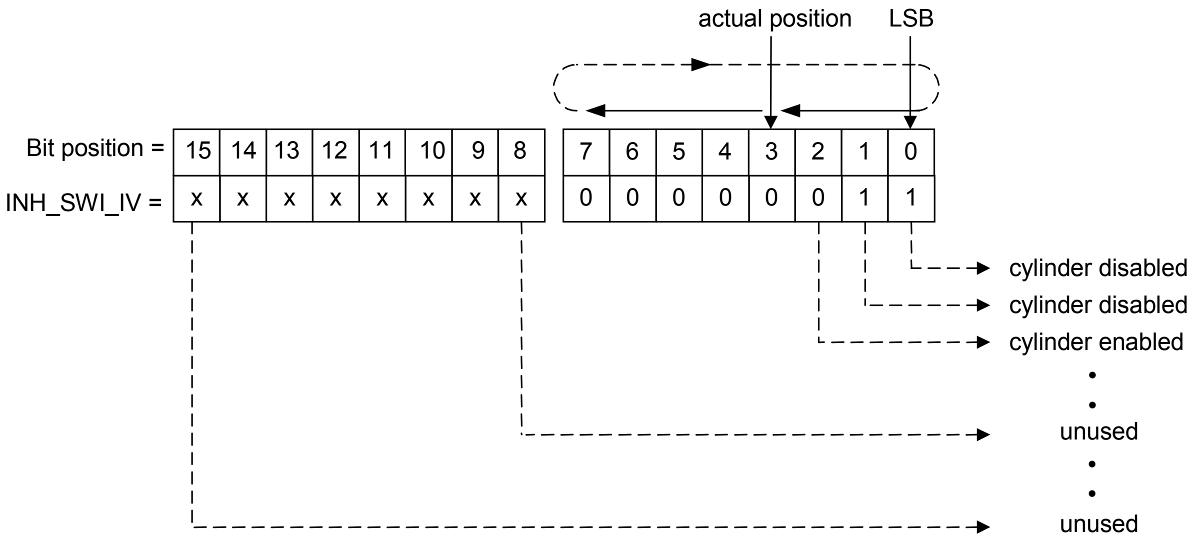
For example a map ID_INH_SWI_IV for an 8-cylinder engine with different shut off sequences (8 next injections):

NR_PAT (Index)	Shut off sequence
0	xxxxxxxx 00000000
1	xxxxxxxx 00000001
2	xxxxxxxx 00000011
3	xxxxxxxx 00000111
4	xxxxxxxx 00001111
5	xxxxxxxx 00011111
6	xxxxxxxx 00111111
7	xxxxxxxx 01111111
8	xxxxxxxx 11111111
The following sequences can be used for reactivation of cylinders	
9	xxxxxxxx 11111111
10	xxxxxxxx 01111111
11	xxxxxxxx 00111111
12	xxxxxxxx 00011111
13	xxxxxxxx 00001111
14	Xxxxxxxx 00000111
15	xxxxxxxx 00000011
16	xxxxxxxx 00000001

Note:

To guarantee that the algorithm works correct, the table ID_INH_SWI_IV has to be applied as follows: The higher the index NR_PAT, the higher the number of cylinders deactivated.

INH_SWI_IV is defined as follows:



(e.g. NC_INJ_INH_SWI_IV_SHIFT_NR = 8, actual bit position = 4):

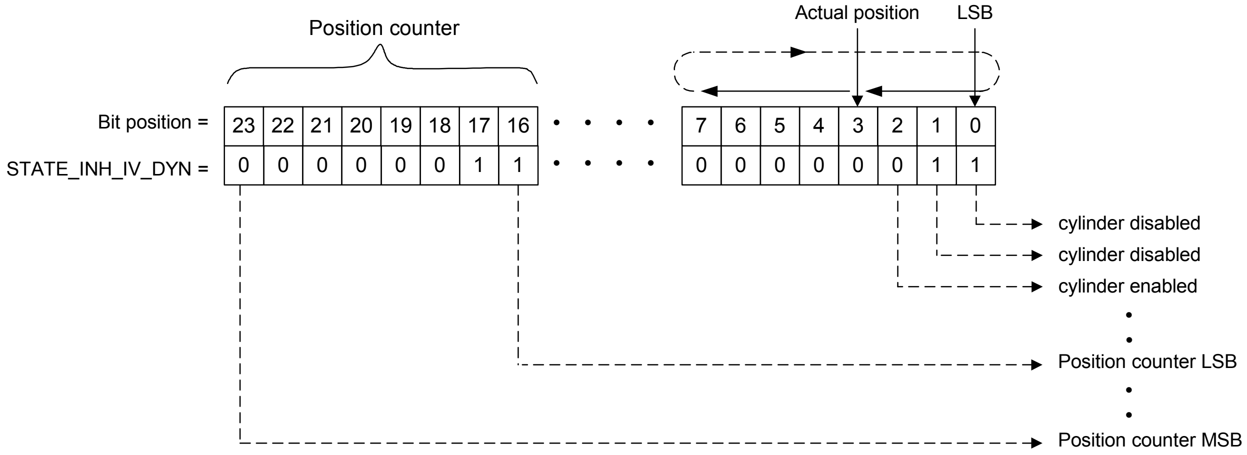
If the bit at a bit position = 1 → cylinder disabled
If the bit at a bit position = 0 → cylinder enabled

The length of the pattern is 16 bit. NC_INJ_INH_SWI_IV_SHIFT_NR defines how many bits are used for the shut off sequence.

NC_CYL_NR < = NC_INJ_INH_SWI_IV_SHIFT_NR < = 16

Calculation of STATE_INH_IV_DYN:

Definition of STATE_INH_IV_DYN:



**Note! The position counter data field inside STATE_INH_IV_DYN will be updated at TRIG_EOI_LIM[x].
It contains the 'Actual position' inside the shut off pattern sequence.**

Select between two different algorithm for STATE_INH_IV_DYN calculation:

(1) IF LC_SWI_MOD_INH_IV_DYN = 0

(1) THEN Classic mode: If NR_PAT increases, then STATE_INH_IV_DYN starts always with a new fuel cut off sequence and 'Actual position' will be set to zero. Otherwise the ‘Actual position’ stays unchanged and a new shut off pattern – or the old one – is used based on INH_SWI_IV – see picture above.

(2) IF NR_PAT > NR_PAT_OLD

(2) THEN

Start with a new fuel cut off sequence at the first bit position (LSB). Use as sequence the current content of INH_SWI_IV.
STATE_INH_IV_DYN = INH_SWI_IV

(2) ELSE

Continue with the fuel cut off sequence at 'Actual position' and change the shut off pattern depending on INH_SWI_IV:

```
STATE_INH_IV_DYN = STATE_INH_IV_DYN AND 0xFFFF0000        ...bitwise
                   OR                                     ...bitwise
                   INH_SWI_IV AND 0x0000FFFF              ...bitwise
```

Note! Pay attention of data consistency of STATE_INH_IV_DYN.

(2) ENDIF

(1) ELSE Modern mode: Only if NR_PAT was zero and increases, then STATE_INH_IV_DYN starts with a new fuel cut off sequence and 'Actual position' will be set to zero. Otherwise the ‘Actual position’ stays unchanged and a new shut off pattern – or the old one – is used based on INH_SWI_IV – see picture above.

(3) IF (NR_PAT_OLD = 0) **AND** (NR_PAT > NR_PAT_OLD)

(3) THEN

Start with a new fuel cut off sequence at the first bit position (LSB). Use as sequence the current content of INH_SWI_IV.
STATE_INH_IV_DYN = INH_SWI_IV

(3) ELSE

Continue with the fuel cut off sequence at 'Actual position' and change the shut off pattern depending on INH_SWI_IV:

```
STATE_INH_IV_DYN = STATE_INH_IV_DYN AND 0xFFFF0000        ...bitwise
                   OR                                     ...bitwise
                   INH_SWI_IV AND 0x0000FFFF              ...bitwise
```

Note! Pay attention of data consistency of STATE_INH_IV_DYN.

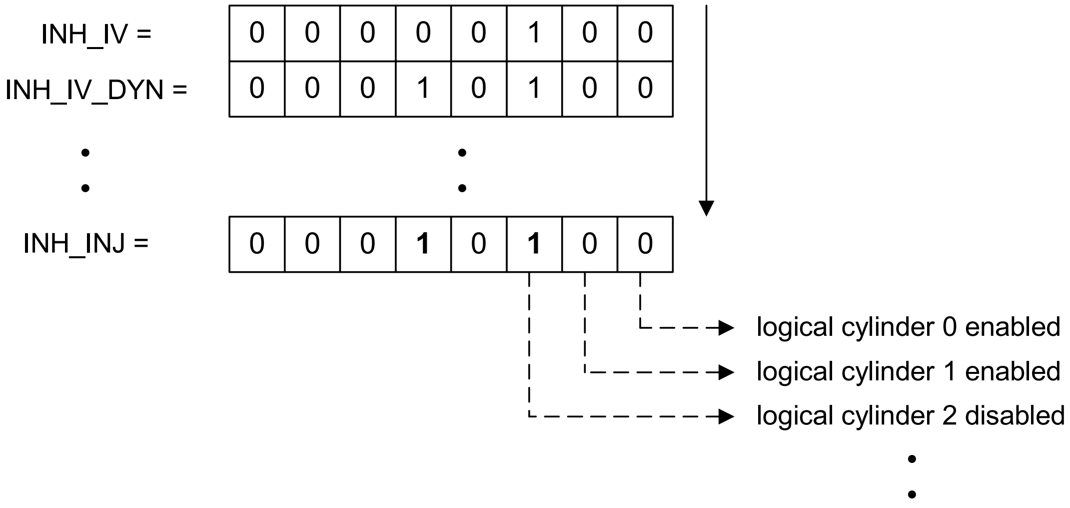
(3) ENDIF

(1) ENDIF

41.29.4 Coordination of shut off information for basic software (INH_INJ)

The output pattern INH_INJ is the bitwise OR of the pattern INH_IV and INH_IV_DYN, linked with a mask $((2^{NC\_CYL\_NR}) -1)$.

As an example see below:



Note! Pay attention of data consistency of INH_IV.

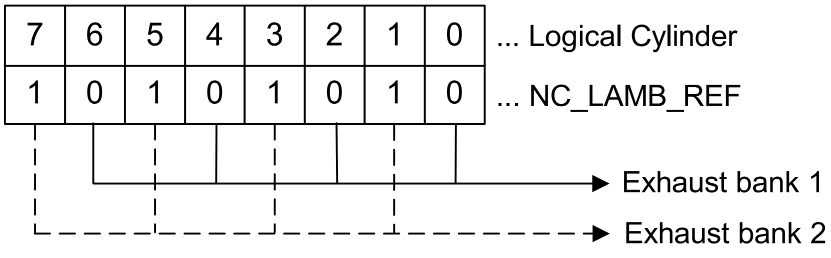
41.29.5 Shut Off Status Information

SUM_INH_IV	.... sum of INH_IV
SUM_INH_IV_CBK_i	.... sum of those bits within INH_IV which are allocated to exhaust cylinderbank i, with i = 1 for exhaust cylinderbank 1 and i = 2 for exhaust cylinderbank 2.
SUM_INH_IV_DYN	..... sum of INH_IV_DYN
SUM_INH_INJ	..... sum of INH_INJ

Note! Pay attention on data consistency of SUM_INH_INJ and SUM_INH_IV_DYN

Note:

The allocation between physical cylinders and exhaust cylinderbank 1 and 2 is defined by the pattern NC_LAMB_REF. For example an 8 cylinder engine:



```
(1) IF   SUM_INH_INJ ≠ 0
(1) THEN
        LV_FCUT_IND = 1            (at least one cylinder is shut off)
    (2) IF      SUM_INH_INJ ≥ NC_CYL_NR
    (2) THEN                       (all cylinders are shut off)
            LV_INJ_CUT = 1
            (3) FOR i = 1 TO NC_CBK_EX_NR DO:
                LV_SCC[i] = 1        (i = 1 and 2, for exhaust cylinder bank 1 and 2)
            (3) ENDFOR
    (2) ELSE
            LV_INJ_CUT = 0
            (4) FOR i = 1 TO NC_CBK_EX_NR DO:
                (5) IF SUM_INH_IV_CBK[i] ≠ 0
                (5) THEN
                LV_SCC[i] = 1
                (5) ELSE
                LV_SCC[i] = 0
                (5) ENDIF
            (4) ENDFOR
    (2) ENDIF
(1) ELSE
        LV_FCUT_IND = 0
        LV_INJ_CUT = 0
        (6) FOR i = 1 TO NC_CBK_EX_NR DO:
            LV_SCC[i] = 0
        (6) ENDFOR
(1) ENDIF
```

41.30 Cylinder Shut Off (Application Incidences)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
INH_IV_EXT	V/O	0...FFH	0...255	1	[-]
Shut off pattern for cylinder shut off (cylinder allocated)					
INH_IV_SWI_MAN	V	0...FFH	0...255	1	[-]
Manual shut off request					
INH_IV_IGN_INJ_LOCK_REQ	V	0...FFH	0...255	1	[-]
Shut off request from ENSD due to backwards rotation detection					
INH_IV_CLK_DRIFT	O/V	0...FFH	0...255	1	[-]
Shut off request from Clock Drift Diagnosis					

Input data:

LV_ST_END	INH_IV_IMOB	LV_SCC[NC_CBK_EX_NR]	INH_IV_DIAG_ERR
LV_IGN_INJ_LOCK_REQ	INH_IV_KWP	INH_IV_KNK	INH_IV_MIS
INH_IV_FTL_MIN	LV_ERR_CLK_DRIFT_PBK_IV[NC_CYL_NR]	INH_IV_CUS	INH_IV_STST
INH_PBK_IV_DIAG_ERR	NC_STATE_STST_ENA		

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_INH_IV_SWI_MAN	1	0...FFH	0...255	1	[-]
Application constant for manual shut off request					
LC_INH_IV_DIAG_ERR_IMDT_ENA	1	0...1H	0...1	1	[-]
Manual switch to activate cylinder shut off immediate after debouncing of an IV error without accordance to MISF					

41.30.1 Cylinder Individual Static Fuel Shut Off

Application conditions:

Activation:	every engine state
Deactivation:	-
Initialization:	at reset all 0

```
# IF NC_STATE_STST_ENA = 1
# THEN
at ECU_STATE transition SYN_ENG_IGK_ON to RUN_ENG
LV_INJ_LOCK_REQ_TMP = 1
at ECU_STATE transition RUN_ENG to ENG_STOP
LV_INJ_LOCK_REQ_TMP = 0
# ENDIF
```

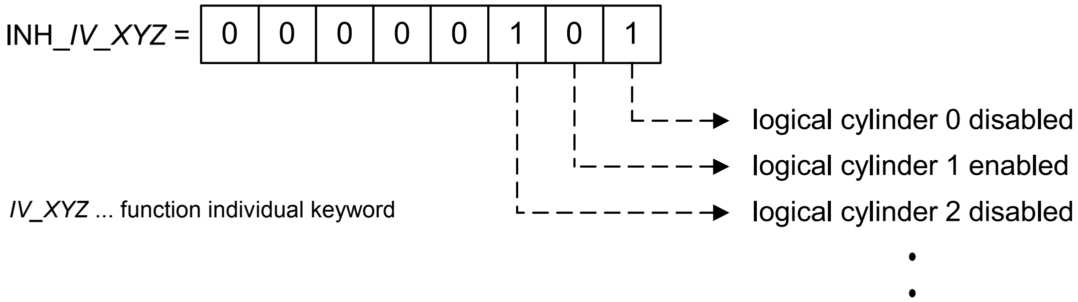
Reccurence: if LV_ST_END = 0: 10 ms

if LV_ST_END = 1: segment synchronous

Note: The module has to be calculated before the module “Cylinder Shut Off”.

Formula section:

41.30.1.1 General



In several engine operating- and system states and in case of malfunctions the fuel injection must be disabled at individual cylinders. All the shut off requirements are coordinated, hence these requirements have to have the same structure as described below:

The information which cylinder has to be shut off static, is coded within a byte (with the length of 8 bit) at the cylinder corresponding position.

41.30.1.2 Static Fuel Shut Off Requirements

41.30.1.2.1 Inhibit by manual switch

INH_IV_SWI_MAN (Input pattern) = C_INH_IV_SWI_MAN

41.30.1.2.2 INH_IV_IMOB

41.30.1.2.3 Shut off request from immobilizer (cylinder allocated):

41.30.1.2.4 INH_IV_IMOB (Input pattern)

41.30.1.2.5 Inhibition of the injection by ENSD

```
IF      LV_IGN_INJ_LOCK_REQ = 1          or
        (LC_INJ_LOCK_REQ_TMP = 1         and
        LV_INJ_LOCK_REQ_TMP = 1)

THEN
    INH_IV_IGN_INJ_LOCK_REQ = $2^{NC\_CYL\_NR} - 1$ (Input pattern)
ELSE
    INH_IV_IGN_INJ_LOCK_REQ = 0x00 (Input pattern)
ENDIF
```

41.30.1.2.6 Inhibition of the injection by injection valve diagnosis error

```
IF (LC_INH_IV_DIAG_ERR_IMDT_ENA = 1)
   OR                                                   ... logical
   ((INH_IV_DIAG_ERR AND INH_IV_MIS) ≠ 0)               ... bitwise
THEN
     INH_IV_DIAG_ERR (Input pattern)
ELSE
     0x00 (Input pattern)
ENDIF
```

41.30.1.2.7 Inhibition of the injection by pre-ignition knock control

Cylinder individual shut off in case of knock pre-ignition	INH_IV_KNK (Input pattern)

41.30.1.2.8 FTL_MIN diagnosis

Cylinder individual shut off in case of less fuel	INH_IV_FTL_MIN

41.30.1.2.9 Inhibition of the injection by customer

Cylinder individual shut off by customer request	INH_IV_CUS

41.30.1.2.10 Workshop tester

Cylinder individual shut off by KWP-job, sent from tester:	INH_IV_KWP

41.30.1.2.11 Inhibition due to Clock Drift in ATIC88 or over voltage in injector

```
For i = 0 to NC_CYL_NR - 1
{
Bit i of INH_IV_CLK_DRIFT = LV_ERR_CLK_DRIFT_PBK_IV[i]
}
```

41.30.1.2.12 Inhibition of the injection for engine shut down due to stop request

INH_IV_STST (Input pattern)

41.30.1.2.13 Inhibition of the injection due to injector power bank errors

Cylinder individual shut off in case of injector power bank errors

INH_PBK_IV_DIAG_ERR

41.30.1.3 Coordination of the Static Fuel Shut Off Requirements

The output pattern INH_IV_EXT is the bitwise OR of all patterns INH_IV_XYZ (and INH_PBK_IV_DIAG_ERR). As an example see below:

INH_IV_XYZ =	0	0	0	0	0	0	0	0
INH_IV_XYZ =	0	0	0	0	0	**1**	0	0
INH_IV_XYZ =	0	0	0	**1**	0	**1**	0	0
:					:			
INH_IV_EXT =	0	0	0	**1**	0	**1**	0	0

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INJ_LOCK_REQ_TMP	1	0...1H	0...1	1	[-]
Switch for lock of injection until complete engine stop					

41.31 INJR application incidences

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_N	O/V	0...1FE0H	0...48.96	0.006	[°CRK/ms]
Engine speed factor					
LV_FAC_TI_EXT_ADJ	O	0...1H	0...1	1	[-]
Flag for external injection time adjustment factor active					
N_FAST_INJ	O	0...1FE0H	0...8160	1	[rpm]
Engine Speed- Resolution 1 rpm					
CTR_CYL_NR_ST_CLC_INJR	O/V	0...7H	0...7	1	[-]
Number of first cylinder in calculation order					
CTR_CYL_NR_STOP_CLC_INJR	V	0...7H	0...7	1	[-]
Number of last cylinder in calculation order					
CTR_CBK_IN_NR_ST_CLC_INJR	O	0...1H	1...2	1	[-]
Start number of intake bank for calculations					
CTR_CBK_IN_NR_STOP_CLC_INJR	O	0...1H	1...2	1	[-]
Stop number of intake bank for calculations					
CTR_CBK_EX_NR_ST_CLC_INJR	O	0...1H	1...2	1	[-]
Start number of exhaust bank for calculations					
CTR_CBK_EX_NR_STOP_CLC_INJR	O	0...1H	1...2	1	[-]
Stop number of exhaust bank for calculations					
FAC_TI_EXT_ADJ	O	0...FFH	0...1.99218	0.0078125	[-]
external injection time adjustment factor					
LV_TI_EXT_ADJ[NC_CYL_NR]	O	0...1H	0...1	1	[-]
Flag for external injection time adjustment					
TI_EXT_ADJ[NC_CYL_NR]	O	0...FFFFH	0...65.535	0.001	[ms]
External injection time adjustment					
MFF_ADD_REAC[NC_CYL_NR]	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Additive fuel amount at reactivation of fuel feed					
LV_IV_TEST_MOD_AUTH	V/O	0...1H	0...1	1	[-]
Flag to authorize the injection valve actuator test					
NR_CYL_IV_TEST_MOD	V/O	0...7H	0...7	1	[-]
Logical cylinder number of that cylinder for which the actuator test is in progress					
LV_MFF_SP_HOM_VLD	V	0...1H	0...1	1	[-]
Flag which indicates a valid fuel amount of at least one cylinder					
LV_ST_INJ_REQ	V/O	0...1H	0...1	1	[-]
Start request flag for the injection driver					
LV_INJ_CRASH_ACT	O	0...1H	0...1	1	[-]
Flag to authorize the crash signal reaction of the injection driver					
CTR_TEST_MOD_IV_EXT[NC_CYL_NR]	O	0...FFH	0...255	1	[-]
Counter of number of injection valve actuator test cycles for external use, cylinder individual					
TFU_INJ	V/O	0...FEH	-48...142.5	0.75	[°C]
Fuel temperature for injection system					
STATE_INJ_MOD_REQ	O/V	1H 2H 3H 41H 42H 43H 80H 81H 82H 83H	SNGS MPLS MPLS+PLS3 HOMS-SNG HOMS HOMS+PLS3 DISABLE SNGH MPLH MPLH+PLS3	1	[-]
Injection mode request for all cylinders for requested combustion mode					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_INJ_MOD_HOM_REQ	O/V	1H	SNGS	1	[-]
		2H	MPLS		
		3H	MPLS+PLS3		
		41H	HOMS-SNG		
		42H	HOMS		
		43H	HOMS+PLS3		
		80H	DISABLE		
		81H	SNGH		
		82H	MPLH		
		83H	MPLH+PLS3		
Injection mode request for all cylinders for homogeneous combustion mode					
STATE_INJ_MOD_S_REQ	O/V	1H	SNGS	1	[-]
		2H	MPLS		
		3H	MPLS+PLS3		
		41H	HOMS-SNG		
		42H	HOMS		
		43H	HOMS+PLS3		
		80H	DISABLE		
		81H	SNGH		
		82H	MPLH		
		83H	MPLH+PLS3		
Injection mode request for all cylinders for stratified combustion mode					
STATE_INJ_MOD_HOMS_REQ	O/V	1H	SNGS	1	[-]
		2H	MPLS		
		3H	MPLS+PLS3		
		41H	HOMS-SNG		
		42H	HOMS		
		43H	HOMS+PLS3		
		80H	DISABLE		
		81H	SNGH		
		82H	MPLH		
		83H	MPLH+PLS3		
Injection mode request for all cylinders for homogeneous-stratified combustion mode					
LV_N_TOOTH_END_ACT	V/O	0...1H	0...1	1	[-]
Flag for indication that engine speed is over N_TOOTH_END					
LFT_L_IV_HOM_REQ	O/V	0...FFH	0...255	1	[-]
Dynamic injector needle lift selection request for homogeneous combustion mode - bit coded; LSB represents pulse 1, etc.					
LFT_L_IV_S_REQ	O/V	0...FFH	0...255	1	[-]
Dynamic injector needle lift selection request for stratified combustion mode - bit coded; LSB represents pulse 1, etc.					
LFT_L_IV_HOMS_REQ	V	0...FFH	0...255	1	[-]
Dynamic injector needle lift selection request for homs combustion mode - bit coded; LSB represents pulse 1, etc.					
NR_CYL_CLC_RED_INJ	O/V	1...8H	1...8	1	[-]
Number of calculated cylinders per segment in case of runtime reduction of the aggregate INJR					
INJ_MOD_GLOBAL	O/V	1H	SNGS	1	[-]
		2H	MPLS		
		3H	MPLS+PLS3		
		4H	PRE_INJ		
		6H	CRASH		
		7H	TEST_PULSE		
		21H	SNGS+MPLP		
		22H	MPLS+MPLP		
		41H	HOMS-SNG		
		42H	HOMS		
		43H	HOMS+PLS3		
		62H	HOMS+MPLP		
		80H	DISABLE		
		81H	SNGH		
		82H	MPLH		
		83H	MPLH+PLS3		
Global injection mode for all cylinders					
STATE_INJ_UPD_ENA	O/V	0...FFH	0...255	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Bitmask to enable/disable injection update calculations					
INJ_MOD_SP[NC_CYL_NR]	O/V	1H 2H 3H 4H 6H 7H 21H 22H 41H 42H 43H 62H 80H 81H 82H 83H	SNGS MPLS MPLS+PLS3 PRE_INJ CRASH TEST_PULSE SNGS+MPLP MPLS+MPLP HOMS-SNG HOMS HOMS+PLS3 HOMS+MPLP DISABLE SNGH MPLH MPLH+PLS3	1	[-]
Injection mode set point, cylinder individual					
INJ_MOD_SP_HOM[NC_CYL_NR]	O/V	1H 2H 3H 4H 6H 7H 21H 22H 41H 42H 43H 62H 80H 81H 82H 83H	SNGS MPLS MPLS+PLS3 PRE_INJ CRASH TEST_PULSE SNGS+MPLP MPLS+MPLP HOMS-SNG HOMS HOMS+PLS3 HOMS+MPLP DISABLE SNGH MPLH MPLH+PLS3	1	[-]
Injection mode set point for homogeneous combustion mode, cylinder individual					
INJ_MOD_SP_S[NC_CYL_NR]	O/V	1H 2H 3H 4H 6H 7H 21H 22H 41H 42H 43H 62H 80H 81H 82H 83H	SNGS MPLS MPLS+PLS3 PRE_INJ CRASH TEST_PULSE SNGS+MPLP MPLS+MPLP HOMS-SNG HOMS HOMS+PLS3 HOMS+MPLP DISABLE SNGH MPLH MPLH+PLS3	1	[-]
Injection mode set point for stratifed combustion mode, cylinder individual					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
INJ_MOD_SP_HOMS[NC_CYL_NR]	O/V	1H 2H 3H 4H 6H 7H 21H 22H 41H 42H 43H 62H 80H 81H 82H 83H	SNGS MPLS MPLS+PLS3 PRE_INJ CRASH TEST_PULSE SNGS+MPLP MPLS+MPLP HOMS-SNG HOMS HOMS+PLS3 HOMS+MPLP DISABLE SNGH MPLH MPLH+PLS3	1	[-]
Injection mode set point for homogeneous-stratified combustion mode, cylinder individual					
IDX_PRS_COR_CYL_CLC_MPLH[NC_CBK_HPP_NR]	O/V	0...7H	0...7	1	[-]
Index to assign the logical cylinder for calculation of pressure correction for homogeneous multiple mode					
IDX_PRS_COR_CYL_CLC_S[NC_CBK_HPP_NR]	O/V	0...7H	0...7	1	[-]
Index to assign the logical cylinder for calculation of pressure correction for stratified mode					
STATE_INJ_MOD_SWI_ACT	O/V	0...FFH	0...255	1	[-]
State indicates a currently ongoing injection mode change, if not equal to zero					
CRK_PSN_STAT_WIN_ST	O/V	0...780H	0...720	0.375	[°CRK]
Start position of static injection update window					
CRK_PSN_STAT_WIN_END	O/V	0...780H	0...720	0.375	[°CRK]
End position of static injection update window					
CRK_PSN_STAT_IGN_UPD_ST	O/V	0...780H	0...720	0.375	[°CRK]
Start of static ignition update window for ignition/injeciton synchronization					
CRK_PSN_STAT_IGN_UPD_END	O/V	0...780H	0...720	0.375	[°CRK]
End of static ignition update window for ignition/injeciton synchronization					
LV_PRS_COR_SNGH_ENA	O/V	0...1H	0...1	1	[-]
Switch to activate pressure pulsation correction for the first pulse homogeneous mode					
LV_PRS_COR_MPLH_ENA	O/V	0...1H	0...1	1	[-]
Switch to activate pressure pulsation correction for homogeneous multi injection mode					
LV_IDX_CYL_CLC_MPLH_VLD	-	0...1H	0...1	1	[-]
Flag to indicate, that at least one calculated cylinder requests triple injection					
FUP_H_INJ[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel pressure (high range) for calculation of injection parameters					
MAP_INJ[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold pressure for calculation of injection parameters					
FUP_H_SP_S_INJ[NC_CBK_HPP_NR]	O/V	0...FFFFH	0...347776	5.3067216	[hPa]
Fuel pressure setpoint for stratified mode (high range) for calculation of injection parameters					
LF_IV_INH_PUC_EXT	O/V	0...FFH	0...255	1	[-]
External injection inhibit pattern, which can be applied during PUC-phases					
LV_INJ_PUC_ENA	O/V	0...1H	0...1	1	[-]
Flag to set inhibit pattern for PUC to external pattern in case of PUC-phases					
LF_PRS_COR_HPP_ENA	O/V	0...FFH	0...255	1	[-]
Logical field to enable injection pressure correction for each fuel bank; bit coded					
CTR_SEG_EOC_INJ[NC_PBK_IV_NR]	V	0...FFH	0...255	1	[-]
Counter that counts the number of segment triggers between two injection end of cycle events.					
CTR_SEG_EOC_INJ_ERR[NC_PBK_IV_NR]	V	0...FFH	0...255	1	[-]
Counter that counts the number of injection driver locks.					
LV_ERR_SEG_EOC_INJ	O/V	0...1H	0...1	1	[-]
Injectiion driver lock error detected.					
LV_CDN_DIAG_SEG_EOC_INJ	O/V	0...1H	0...1	1	[-]
Condition flags for injectiion driver lock error detection.					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_SEG_EOC_INJ	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom for injection driver lock diagnosis.					

Input data:

INJ_MOD_S_REQ	NC_CYL_NR	FUP_H	PREV_STATE_IV
NC_CBK_IN_NR	NC_CBK_EX_NR	CTR_CBK_IN_NR_ST_CLC	CTR_CYL_NR_ST_CLC
N_TOOTH_CUS	C_N_TOOTH_END	CTR_CBK_IN_NR_STOP_CLC	CTR_CYL_NR_STOP_CLC
INJ_MOD_HOM_REQ	LV_ST_END	CTR_CBK_EX_NR_ST_CLC	CTR_CBK_EX_NR_STOP_CLC
N	STATE_EFP_CRASH_CAN	MFF_SP_1_HOM[NC_CYL_NR]	TFU_IV
NR_INJ_PLS_HOM_REQ	NC_INJ_MOD_SINGLE	LV_IGA_AND_INJ_SWI_HOMS	LV_IGA_AND_INJ_SWI
Mk_soll_h[NC_CYL_NR][NC_NR_IV_PLS]	LV_ST_INJ_AUTH	LV_ST_INJ	NR_INJ_PLS_S_REQ
NC_NR_IV_PLS	LFT_L_IV_REQ_HOM_CUS	LFT_L_IV_REQ_S_CUS	LFT_L_IV_REQ_HOMS_CUS
LV_PRS_COR_SNGH_ENA_CUS	LV_LAM_AD_INJ_CUS_ACK[NC_CBK_EX_NR]	MAP	MAF_CYL
TCYL_MDL_CUS	INJ_MOD[NC_CYL_NR]	SEG_NR	LV_PRS_COR_MPLH_ENA_CUS
LV_S_CLC	NC_PBK_IV_NR	NC_IDX_CYL_PBK_IV_REF[NC_CYL_NR]	STATE_PBK_IV_INI
STATE_ERR_PBK_IV_ST_RAW[NC_PBK_IV_NR]	STATE_INJ_DR	STATE_INJ_CRASH_ACT	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_MFF_ADD_REAC	8*8	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_MFF_SP_HOM_IP_MFF_ADD_REAC	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_IP_MFF_ADD_REAC	8	0...1FE0H	0...8160	1	[rpm]
Additive fuel amount at reactivation of fuel feed					
LC_INJ_MOD_SP_MAN_ENA	1	0...1H	0...1	1	[-]
Logical constant to enable manual injection mode selection					
C_INJ_MOD_SP_MAN	1	1H 2H 3H 41H 42H 43H 80H 81H 82H 83H	SNGS MPLS MPLS+PLS3 HOMS-SNG HOMS HOMS+PLS3 DISABLE SNGH MPLH MPLH+PLS3	1	[-]
Injection mode set-point for manual selection					
C_CRK_PSN_STAT_WIN_ST	1	0...780H	0...720	0.375	[°CRK]
Start position of static injection update window					
C_CRK_PSN_STAT_IGN_UPD_ST	1	0...780H	0...720	0.375	[°CRK]
Start of static ignition update window for ignition/injeciton synchronization					
C_CRK_PSN_STAT_IGN_UPD_END	1	0...780H	0...720	0.375	[°CRK]
End of static ignition update window for ignition/injeciton synchronization					
C_NR_CYL_OFS_CLC_RED_INJ	1	0...7H	0...7	1	[-]
Segment offset to calculated cylinders for INJR runtime optimization strategy					
LC_LOAD_CPU_RED_INJ	1	0...1H	0...1	1	[-]
Switch to enable INJR runtime optimization strategy					
LC_MFF_ADD_REAC_ENA	1	0...1H	0...1	1	[-]
Switch to enable wallfilm correction on fuel mass					
LC_MAP_PHY_CLC_ENA	1	0...1H	0...1	1	[-]
Switch to enable physical model for cylinder pressure calculation.					
C_FAC_MAP_TCYL_MDL	1	0...FFFFH	0...6553.5	0.1	[N/(kg*K*m²)]
Constant term of ideal gas law to calculate physical model of cylinder pressure.					
ID_NR_CYL_CLC_RED_INJ	4	1...8H	1...8	1	[-]
LDP_N_ID_NR_CYL_CLC_RED_INJ	4	0...1FE0H	0...8160	1	[rpm]
Number of calculated cylinders per segment in case of INJR runtime optimization					
C_CTR_SEG_EOC_INJ_MAX_PLAUS	1	0...FFH	0...255	1	[-]
Maximum plausible number of segment triggers between two injection end of cycle events on one injection bank.					

Exported actions:

ACTION_INJR_SetStateInjUpdEna(IN <Cyl>, IN <State_Inj_Upd_Ena>)
This action sets the cylinder individual bit <Cyl> of STATE_INJ_UPD_ENA to <State_Inj_Upd_Ena>.

Imported actions:

ACTION_INFR_SetInjMode (IN <Injection_mode>)
This action sets the injection mode (DISABLE, SEQUENTIAL, STATIC, PREINJECTION, CRASH).
ACTION_ERRM_NoFilterSymptom(IN<IDX_DIAG>, IN< LV_CDN_DIAG >, IN< ERR_SYM >, IN< LV_ERR_SET >, IN<LV_ERR_RST>, IN< LV_END_DIAG >, OUT< LV_ERR>)
ACTION_ERRM_NoFilterReset(IN<IDX_DIAG>, OUT< LV_ERR>)

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_IDX_DIAG_SEG_EOC_INJ	1	1...FFFFH	1...65535	1	[-]
Index of diagnostic failure SEG_EOC_INJ					

FUNCTION DESCRIPTION:

General information:

This specification is for project specific adaptations.

41.31.1 Runtime reduction

Description:

This chapter describes runtime reduction steps. First step is to reduce cylinder dependent calculations from all cylinders to one cylinder, dependent on the engine speed. An additional step is to reduce bank dependent calculations to that intake or exhaust bank, which is allocated to the related cylinder by NC_IN_REF and NC_LAMB_REF. The control variables for cylinder and bank dependent calculations will be imported via CTR_CYL_NR_xxx and CTR_CBK_xxx data.

Application conditions :

Activation :	every engine state
Deactivation :	-
Initialization:	-
Recurrence :	LV_ST_END = 0: 10 ms LV_ST_END = 1: segment sychronous

Formula Section :

For loop for runtime optimization:

In this specification the for loop

```
FOR i = 0, i < NR_CYL_CLC_RED_INJ DO
        x = CTR_CYL_NR_ST_CLC_INJR + i
        (1a)IF x  >= NC_CYL_NR
        (1a)THEN
              x = x – NC_CYL_NR
        (1a)ENDIF
        <loop content>
ENDFOR
```

is substituted by

```
FOR cylinder_start to cylinder_stop DO
    <loop content>
ENDFOR
```

Note: In the for loops, x will be used as loop variable to indicate the cylinder x!

Determine cylinder and bank numbers for follow up calculations

```
CTR_CYL_NR_ST_CLC_INJR          = CTR_CYL_NR_ST_CLC
CTR_CYL_NR_STOP_CLC_INJR        = CTR_CYL_NR_STOP_CLC
CTR_CBK_IN_NR_ST_CLC_INJR       = CTR_CBK_IN_NR_ST_CLC
CTR_CBK_IN_NR_STOP_CLC_INJR     = CTR_CBK_IN_NR_STOP_CLC
CTR_CBK_EX_NR_ST_CLC_INJR       = CTR_CBK_EX_NR_ST_CLC
CTR_CBK_EX_NR_STOP_CLC_INJR     = CTR_CBK_EX_NR_STOP_CLC
```

Note: NR_CYL_CLC_RED_INJ has to be greater than zero and less or equal than NC_CYL_NR;

```
(1)IF LC_LOAD_CPU_RED_INJ = 0
    OR CTR_CYL_NR_ST_CLC_INJR ≠ 0
    OR CTR_CYL_NR_STOP_CLC_INJR ≠ (NC_CYL_NR – 1)
    OR LV_ST_END = 0
    OR SUM(LV_LAM_AD_INJ_CUS_ACK[ ]) ≠ 0     // sum over all elements; at least one
                                               exhaust-bank requests MFMA
(1)THEN
    NR_CYL_CLC_RED_INJ = CTR_CYL_NR_STOP_CLC_INJR
                                  - CTR_CYL_NR_ST_CLC_INJR + 1
(1)ELSE
  (2)IF SEG_NR < C_NR_CYL_OFS_CLC_RED_INJ
  (2)THEN
    CTR_CYL_NR_STOP_CLC_INJR = NC_CYL_NR + SEG_NR
        - C_NR_CYL_OFS_CLC_RED_INJ
  (2)ELSE
    CTR_CYL_NR_STOP_CLC_INJR = SEG_NR - C_NR_CYL_OFS_CLC_RED_INJ
  (2)ENDIF
  NR_CYL_CLC_RED_INJ = ID_NR_CYL_CLC_RED_INJ(Input: N)
  (3)IF CTR_CYL_NR_STOP_CLC_INJR < (NR_CYL_CLC_RED_INJ – 1)
  (3)THEN
          CTR_CYL_NR_ST_CLC_INJR = NC_CYL_NR
              + CTR_CYL_NR_STOP_CLC_INJR - NR_CYL_CLC_RED_INJ + 1
    (3)ELSE
```

```
    CTR_CYL_NR_ST_CLC_INJR = CTR_CYL_NR_STOP_CLC_INJR
        - NR_CYL_CLC_RED_INJ + 1
  (3)ENDIF
(1)ENDIF
```

41.31.2 General tasks

Application conditions :

Activation : every engine state

Deactivation : -

Initialization: at reset : FAC_TI_EXT_ADJ = 1
LV_TI_EXT_ADJ[NC_CYL_NR] = 0
TI_EXT_ADJ[NC_CYL_NR] = 0

Recurrence : LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

Formula Section :

To prevent N_FAST_INJ from signal jittering a hysteresis for the calculation is introduced:

```
(1) IF (N > = C_N_TOOTH_END)
(1) THEN
    LV_N_TOOTH_END_ACT = 1
(1) ELSE
    (2) IF (N <  C_N_TOOTH_END - 100)
    (2) THEN
            LV_N_TOOTH_END_ACT = 0
    (2) ENDIF
(1) ENDIF
(3) IF (LV_N_TOOTH_END_ACT = 1)
(3) THEN
    N_FAST_INJ = N
(3) ELSE
    N_FAST_INJ = N_TOOTH_CUS
(3) ENDIF

(4) IF (LV_ST_INJ = 0)
(4) THEN
   LV_MFF_SP_HOM_VLD = 0
   (1) FOR x = 0 TO (NC_CYL_NR -1)
            (2) IF ((Mk_soll_h[x][0] > 0) OR (Mk_soll_h[x][1] > 0) OR (Mk_soll_h[x][2] > 0))
            (2) THEN
                    LV_MFF_SP_HOM_VLD = 1
            (2) ENDIF
   (1) END FOR
   (5) IF (LV_MFF_SP_HOM_VLD = 1) AND (LV_ST_INJ_AUTH = 1)
   (5) THEN
            LV_ST_INJ_REQ = 1
   (5) ELSE
            LV_ST_INJ_REQ = 0
   (5) ENDIF
(4) ELSE
            LV_ST_INJ_REQ = 0
```

```
(4) ENDIF

TFU_INJ = TFU_IV
FAC_N = N_FAST_INJ  * 360 / (1000 * 60)
LV_FAC_TI_EXT_ADJ = 0
FAC_TI_EXT_ADJ = 1
LV_PRS_COR_SNGH_ENA = LV_PRS_COR_SNGH_ENA_CUS
LV_PRS_COR_MPLH_ENA = LV_PRS_COR_MPLH_ENA_CUS
(2)FOR cylinder_start to cylinder_stop DO
    LV_TI_EXT_ADJ[x] = 0
    TI_EXT_ADJ[x]= 0
(2) END FOR
FUP_H_INJ[0] = FUP_H
FUP_H_SP_S_INJ[0] = FUP_H


(6)IF LC_MAP_PHY_CLC_ENA = 1
(6)THEN – physical calculation of cylinder pressure at inlet closing
    The cylinder pressure is calculated by use of the ideal gas law:
            $p = m * T * R_S/V$
    $R_S/V$ as well as the unit conversions (1hPa = 100N/m² and 1rpm = 30stk/h) is included in
    C_FAC_MAP_TCYL_MDL (C_FAC_MAP_TCYL_MDL = $R_S/(V*3000)$).
    MAP_INJ[0] = MAF_CYL * TCYL_MDL_CUS * C_FAC_MAP_TCYL_MDL / N
(6)ELSE
    MAP_INJ[0] = MAP

(6)ENDIF



LV_INJ_PUC_ENA = 0
LF_IV_INH_PUC_EXT = 255 // all bits must be set to 1
```

41.31.3 Compensation of missing residual-gas amount after PUC

Description:

This chapter describes the compensation of missing residual-gas amount at restart of fuel feed. This is caused by a inhibit of injection in the past. At restart of fuel feed for the first combustion residual-gas amount is missing. Therfore the calculated fuel amount is wrong. This functionality should compensate this effect.

Application conditions :

Activation :	LV_ST_END = 0
Deactivation :	-
Initialization:	-
Recurrence :	10 ms

Formula Section :

```
FOR x = 0 TO NC_CYL_NR-1
    MFF_ADD_REAC[x] = 0
END FOR
```

Application conditions :

Activation :	LV_ST_END = 1
Deactivation :	-
Initialization:	-
Recurrence :	segment sychronous

Formula Section :

```
(1)IF LC_MFF_ADD_REAC_ENA = 1
(1)THEN
    FOR cylinder_start to cylinder_stop DO
        IF (Bit-Nr x of PREV_STATE_IV = 0)
        THEN
            MFF_ADD_REAC[x] = IP_MFF_ADD_REAC
            Input for IP_MFF_ADD_REAC is (MFF_SP_1_HOM[x], N)
        ELSE
            MFF_ADD_REAC[x] = 0
        ENDIF
    END FOR
(1)ENDIF
```

41.31.4 Injector test mode

Application conditions:

Activation: LV_ES = 1

Deactivation: -

Initialization: -

Recurrence: 1 s

This chapter describes the necessary interfaces to the procedure for actuator tests by diagnostic tester.

41.31.4.1 Formula Section:

```
LV_IV_TEST_MOD_AUTH = 0
NR_CYL_IV_TEST_MOD = 0
(1) FOR x = 0 TO (NC_CYL_NR -1)
    CTR_TEST_MOD_IV_EXT[x] = CTR_TEST_MOD_IV[x]
(1) END FOR
```

Please note! This for-loop should be realized in SW with a #define. CTR_TEST_MOD_IV_EXT[NC_CYL_NR] has to be realized also as a #define for CTR_TEST_MOD_IV to preserve resources.

41.31.5 Crash signal reaction of injection driver

Application conditions:

Activation: every engine state

Deactivation: -

Initialization: -

Recurrence: 10 ms

This chapter describes the necessary interfaces to customer signals to react on a vehicle crash.

41.31.5.1 Formula Section:

```
(1) IF STATE_EFP_CRASH_CAN = 2H
(1) THEN
    LV_INJ_CRASH_ACT = 1
(1) ELSE
    LV_INJ_CRASH_ACT = 0
(1) ENDIF
```

41.31.6 Select injection mode STATE_INJ_MOD_REQ and copy information of dynamic needle lift selection from customer nomenclature to SV nomenclature

Application conditions :

Activation : every engine state

Deactivation : -

Initialization: at reset : STATE_INJ_MOD_REQ = 'DISABLE'
STATE_INJ_MOD_HOM_REQ = 'SNGH'
STATE_INJ_MOD_S_REQ = 'SNGS'
STATE_INJ_MOD_HOMS_REQ = 'HOMS-SNG'
LFT_L_IV_HOM_REQ = 0
LFT_L_IV_S_REQ = 0
INJ_MOD_GLOBAL = "DISABLE"
STATE_INJ_MOD_SWI_ACT = 0

at engine run to engine stop event: INJ_MOD_GLOBAL = "DISABLE"
STATE_INJ_MOD_SWI_ACT = 0

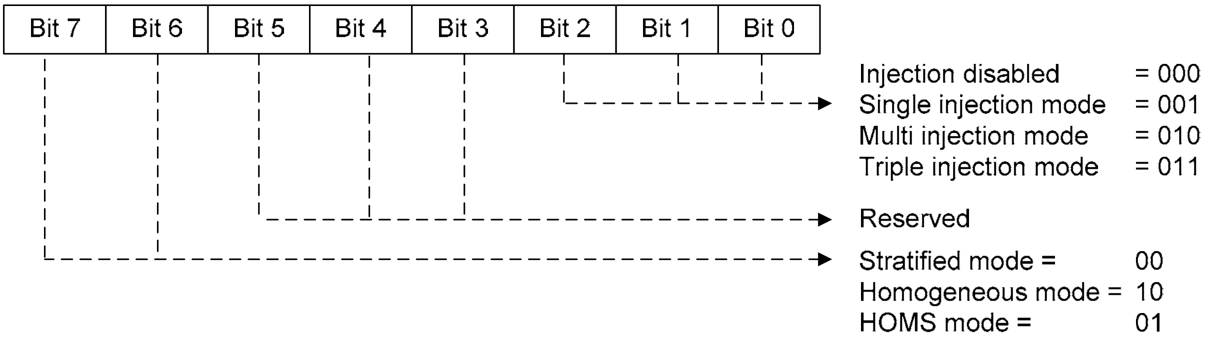
Recurrence : LV_ST_END = 0: 10 ms
LV_ST_END = 1: segment synchronous

Description:

This section determines the actual set injection mode request.

Byte definition

Definitions bit masks for injection modes: STATE_INJ_MOD_REQ, STATE_INJ_MOD_HOM_REQ, STATE_INJ_MOD_S_REQ and STATE_INJ_MOD_HOMS_REQ



Example: homogenous multi injection mode (MPLH)

1	0	0	0	0	0	1	0

The mapping of the information about injector needle lift selection is done from a bit field as input to bit field as output. Here, the conversion from customer pulse numeration to SV numeration is done.

Important note: For better readability of the specification, the LSB of each bit field, which corresponds to the physically first injection pulse, is referred to as bit 1 (**not** bit 0 as it would be in SW). The next bit would be bit 2 etc.

Definition of NC constants

NC_INJ_MOD_SINGLE = 0x01H

Formula Section :

```
(1) IF (LC_INJ_MOD_SP_MAN_ENA = 1)
(1) THEN
    (2) IF (C_INJ_MOD_SP_MAN = 'SNGH') OR (C_INJ_MOD_SP_MAN = 'DISABLE')
    (2) THEN
        STATE_INJ_MOD_REQ = 'SNGH'
        STATE_INJ_MOD_HOM_REQ = 'SNGH'
        STATE_INJ_MOD_S_REQ = 'SNGS'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS-SNG'
    (2) ENDIF

    (3) IF (C_INJ_MOD_SP_MAN = 'MPLH')
    (3) THEN
        STATE_INJ_MOD_REQ = 'MPLH'
        STATE_INJ_MOD_HOM_REQ = 'MPLH'
        STATE_INJ_MOD_S_REQ = 'SNGS'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS-SNG'
    (3) ENDIF

    (4) IF (C_INJ_MOD_SP_MAN = 'MPLH+PLS3')
    (4) THEN
        STATE_INJ_MOD_REQ = 'MPLH+PLS3'
        STATE_INJ_MOD_HOM_REQ = 'MPLH+PLS3'
        STATE_INJ_MOD_S_REQ = 'SNGS'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS-SNG'
    (4) ENDIF

    (5) IF (C_INJ_MOD_SP_MAN = 'SNGS')
    (5) THEN
        STATE_INJ_MOD_REQ = 'SNGS'
        STATE_INJ_MOD_HOM_REQ = 'SNGH'
        STATE_INJ_MOD_S_REQ = 'SNGS'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS-SNG'
    (5) ENDIF

    (6) IF (C_INJ_MOD_SP_MAN = 'MPLS')
    (6) THEN
        STATE_INJ_MOD_REQ = 'MPLS'
        STATE_INJ_MOD_HOM_REQ = 'SNGH'
        STATE_INJ_MOD_S_REQ = 'MPLS'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS'
```

```
    (6) ENDIF

    (6a) IF (C_INJ_MOD_SP_MAN = 'MPLS+PLS3')
    (6a) THEN
        STATE_INJ_MOD_REQ = 'MPLS+PLS3'
        STATE_INJ_MOD_HOM_REQ = 'SNGH'
        STATE_INJ_MOD_S_REQ = 'MPLS+PLS3'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS+PLS3'
    (6a) ENDIF

    (6b) IF (C_INJ_MOD_SP_MAN = 'HOMS-SNG')
    (6b) THEN
        STATE_INJ_MOD_REQ = 'HOMS-SNG'
        STATE_INJ_MOD_HOM_REQ = 'SNGH'
        STATE_INJ_MOD_S_REQ = 'SNGS'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS-SNG'
    (6b) ENDIF


    (7) IF (C_INJ_MOD_SP_MAN = 'HOMS')
    (7) THEN
        STATE_INJ_MOD_REQ = 'HOMS'
        STATE_INJ_MOD_HOM_REQ = 'SNGH'
        STATE_INJ_MOD_S_REQ = 'MPLS'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS'
    (7) ENDIF

    (7a) IF (C_INJ_MOD_SP_MAN = 'HOMS+PLS3')
    (7a) THEN
        STATE_INJ_MOD_REQ = 'HOMS+PLS3'
        STATE_INJ_MOD_HOM_REQ = 'SNGH'
        STATE_INJ_MOD_S_REQ = 'MPLS+PLS3'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS+PLS3'
    (7a) ENDIF

    LFT_L_IV_HOM_REQ = 0
    LFT_L_IV_S_REQ = 0

(1) ELSE

    (8) IF (INJ_MOD_HOM_REQ = NC_INJ_MOD_SINGLE )
    (8) THEN
        STATE_INJ_MOD_HOM_REQ = 'SNGH'
        Bit 1 of LFT_L_IV_HOM_REQ = Bit 1 of LFT_L_IV_REQ_HOM_CUS
        Bit 2 of LFT_L_IV_HOM_REQ = Bit 3 of LFT_L_IV_HOM_REQ = 0
    (8) ELSE
        (9) IF (NR_INJ_PLS_HOM_REQ = 3)
        (9) THEN
            STATE_INJ_MOD_HOM_REQ = 'MPLH+PLS3'
            Bit 1 of LFT_L_IV_HOM_REQ = Bit 1 of LFT_L_IV_REQ_HOM_CUS
            Bit 2 of LFT_L_IV_HOM_REQ = Bit 2 of LFT_L_IV_REQ_HOM_CUS
            Bit 3 of LFT_L_IV_HOM_REQ = Bit 3 of LFT_L_IV_REQ_HOM_CUS
```

```
        (9) ELSE
            STATE_INJ_MOD_HOM_REQ = 'MPLH'
            Bit 1 of LFT_L_IV_HOM_REQ = Bit 1 of LFT_L_IV_REQ_HOM_CUS
            Bit 2 of LFT_L_IV_HOM_REQ = Bit 2 of LFT_L_IV_REQ_HOM_CUS
            Bit 3 of LFT_L_IV_HOM_REQ = 0
        (9) ENDIF
    (8) ENDIF

    (10) IF (INJ_MOD_S_REQ = NC_INJ_MOD_SINGLE )
    (10) THEN
        STATE_INJ_MOD_S_REQ = 'SNGS'
        STATE_INJ_MOD_HOMS_REQ = 'HOMS-SNG'
        Bit 1 of LFT_L_IV_S_REQ = Bit 3 of LFT_L_IV_REQ_S_CUS
        Bit 2 of LFT_L_IV_S_REQ = Bit 3 of LFT_L_IV_S_REQ = 0
        Bit 1 of LFT_L_IV_HOMS_REQ = Bit 3 of LFT_L_IV_REQ_HOMS_CUS
        Bit 2 of LFT_L_IV_HOMS_REQ = Bit 3 of LFT_L_IV_HOMS_REQ = 0

    (10) ELSE
        (10a) IF (NR_INJ_PLS_S_REQ = 3)
        (10a) THEN
                STATE_INJ_MOD_S_REQ = 'MPLS+PLS3'
                STATE_INJ_MOD_HOMS_REQ = 'HOMS+PLS3'
                Bit 1 of LFT_L_IV_S_REQ = Bit 1 of LFT_L_IV_REQ_S_CUS
                Bit 2 of LFT_L_IV_S_REQ = Bit 2 of LFT_L_IV_REQ_S_CUS
                Bit 3 of LFT_L_IV_S_REQ = Bit 3 of LFT_L_IV_REQ_S_CUS
                Bit 1 of LFT_L_IV_HOMS_REQ = Bit 1 of LFT_L_IV_REQ_HOMS_CUS
                Bit 2 of LFT_L_IV_HOMS_REQ = Bit 2 of LFT_L_IV_REQ_HOMS_CUS
                Bit 3 of LFT_L_IV_HOMS_REQ = Bit 3 of LFT_L_IV_REQ_HOMS_CUS
        (10a) ELSE
                STATE_INJ_MOD_S_REQ = 'MPLS'
                STATE_INJ_MOD_HOMS_REQ = 'HOMS'
                Bit 1 of LFT_L_IV_S_REQ = Bit 2of LFT_L_IV_REQ_S_CUS
                Bit 2 of LFT_L_IV_S_REQ = Bit 3 of LFT_L_IV_REQ_S_CUS
                Bit 3 of LFT_L_IV_S_REQ = 0
                Bit 1 of LFT_L_IV_HOMS_REQ = Bit 1 of LFT_L_IV_REQ_HOMS_CUS
                Bit 2 of LFT_L_IV_HOMS_REQ = Bit 3 of LFT_L_IV_REQ_HOMS_CUS
                Bit 3 of LFT_L_IV_HOMS_REQ = 0
        (10a) ENDIF
    (10) ENDIF

    (11) IF (LV_IGA_AND_INJ_SWI = 1)
    (11) THEN (HOM)
        STATE_INJ_MOD_REQ = STATE_INJ_MOD_HOM_REQ
    (11) ELSE (S or HOMS)

      (12) IF (LV_IGA_AND_INJ_SWI_HOMS = 1)
      (12) THEN
          STATE_INJ_MOD_REQ = STATE_INJ_MOD_HOMS_REQ
      (12) ELSE
          STATE_INJ_MOD_REQ = STATE_INJ_MOD_S_REQ
      (12) ENDIF

    (11) ENDIF
```

```
        (13) IF (LV_IGA_AND_INJ_SWI_HOMS = 1)
        (13) THEN
               LFT_L_IV_S_REQ = LFT_L_IV_HOMS_REQ
        (13) ENDIF

    (1) ENDIF
```

Calculation of injection mode setpoints:

```
(1)FOR cylinder_start to cylinder_stop DO
        Bit x of STATE_INJ_UPD_ENA = 1
        (1)IF LV_S_CLC = 1
        (1)THEN
                INJ_MOD_SP_S[x] = STATE_INJ_MOD_S_REQ
                INJ_MOD_SP_HOMS[x] = STATE_INJ_MOD_HOMS_REQ
        (1)ELSE
                INJ_MOD_SP_S[x] = "DISABLE"
                INJ_MOD_SP_HOMS[x] = "DISABLE"
        (1)ENDIF
(1)ENDFOR
```

```
Calculate indices for pressure correction:
IDX_PRS_COR_CYL_CLC_S[0] = CTR_CYL_NR_ST_CLC_INJR
LF_PRS_COR_HPP_ENA = 1 // set bit 0 to 1 to indicate that fuel bank 1 has to be calculated
```

```
(1)IF SUM(LV_LAM_AD_INJ_CUS_ACK[ ]) ≠ 0   // sum over all elements; at least one
                                                   exhaust-bank requests MFMA
   AND STATE_INJ_MOD_REQ = "MPLH+PLS3"
(1)THEN – Minimum fuel mass adaptation is requested and at least 1 cylinder requests triple
          injection
        LV_IDX_CYL_CLC_MPLH_VLD = 0  // this local flag indicates, that at least one cylinder
                                             within the for-loop requests triple injection
        (2)FOR cylinder_start to cylinder_stop DO
                (2)IF MFF_SP_3_HOM[x] > 0
                (2)THEN – cylinder x requests triple injection
                        INJ_MOD_SP_HOM[x] = "MPLH+PLS3"
                        INJ_MOD_SP[x] = "MPLH+PLS3"
                        IDX_PRS_COR_CYL_CLC_MPLH[0] = x
                        LV_IDX_CYL_CLC_MPLH_VLD = 1
                (2)ELSE
                        (3)IF MFF_SP_2_HOM[x] = 0
```

```
                    (3)THEN – cylinder x requests single injection
                        INJ_MOD_SP_HOM[x] = "SNGH"
                        INJ_MOD_SP[x] = "SNGH"
                    (3)ELSE - cylinder x requests double injection (invalid for MFMA)
                        INJ_MOD_SP_HOM[x] = "MPLH+PLS3"
                        INJ_MOD_SP[x] = "MPLH+PLS3"
                        (4)IF LV_IDX_CYL_CLC_MPLH_VLD = 0
                        (4)THEN
                            IDX_PRS_COR_CYL_CLC_MPLH[0] = x
                        (4)ENDIF
                    (3)ENDIF
                (2)ENDIF
            (2)ENDFOR
        (1)ELSE
            (3)FOR cylinder_start to cylinder_stop DO
                INJ_MOD_SP[x] = STATE_INJ_MOD_REQ
                INJ_MOD_SP_HOM[x] = STATE_INJ_MOD_HOM_REQ
            (3)ENDFOR
            IDX_PRS_COR_CYL_CLC_MPLH[0] = CTR_CYL_NR_ST_CLC_INJR
        (1)ENDIF
```

Calculate variables used by other functionalities (stable interface):

```
INJ_MOD_GLOBAL = STATE_INJ_MOD_REQ
(1)FOR cylinder_start to cylinder_stop DO
    (1)IF INJ_MOD_SP[x] ≠ INJ_MOD[x]
    (1)THEN
        Bit x of STATE_INJ_MOD_SWI_ACT = 1
    (1)ELSE
        Bit x of STATE_INJ_MOD_SWI_ACT = 0
    (1)ENDIF
(1)ENDFOR
```

41.31.7 Calculation of injection fixed update window and ignition update window

Application conditions :

Activation : every engine state

Deactivation : -

Initialization: at reset : **calculate Formula Section**

Recurrence : 1s

Description:

This section calculates the crank windows for ignition and injection update. The initialization routine at reset is the same as the time synchronous routine (1s)!

Formula Section :

```
CRK_PSN_STAT_WIN_ST = C_CRK_PSN_STAT_WIN_ST
(1)IF CRK_PSN_STAT_WIN_ST >= ((720°/NC_CYL_NR)*1.1)
(1)THEN
    CRK_PSN_STAT_WIN_ST = ((720°/NC_CYL_NR)*1.1)
    CRK_PSN_STAT_WIN_END = 720°
(1)ELSE
    CRK_PSN_STAT_WIN_END = 720° - ((720°/NC_CYL_NR)*1.1
                                            - CRK_PSN_STAT_WIN_ST)
(1)ENDIF
CRK_PSN_STAT_IGN_UPD_ST = C_CRK_PSN_STAT_IGN_UPD_ST
CRK_PSN_STAT_IGN_UPD_END = C_CRK_PSN_STAT_IGN_UPD_END
```

41.31.8 Description of exported actions

41.31.8.1 Description of ACTION_INJR_SetStateInjUpdEna

ACTION_INJR_SetStateInjUpdEna(IN <Cyl>, IN <State_Inj_Upd_Ena>)

This action sets the cylinder individual bit <Cyl> of STATE_INJ_UPD_ENA to <State_Inj_Upd_Ena>.

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Cyl	IN	0 ... 7	0 ... 7	1	-
Number of logical cylinder.					
State_Inj_Upd_Ena	IN	0 ... 1	0 ... 1	1	-
Logical value for setting or resetting the corresponding cylinder value.					

Formula Section:

Bit Cyl of STATE_INJ_UPD_ENA = State_Inj_Upd_Ena

Note: The LSB of STATE_INJ_UPD_ENA is referred to as Bit 0 (zero based indication)!

41.31.9 Detection of locked injection driver, fault reaction and diagnosis

This function detects a locked injection driver and generates a resync to restart the driver. Additionally, a failure memory entry is done.

41.31.9.1 Detection of locked injection driver – segment synchronous task

Application conditions :

Activation: every engine state

Deactivation: -

Initialization: at reset: CTR_SEG_EOC_INJ[ ] = 0 // all NC_PBK_IV_NR elements
CTR_SEG_EOC_INJ_ERR[ ] = 0
// all NC_PBK_IV_NR elements
ACTION_ERRM_NoFilterReset(
IN<NC_IDX_DIAG_SEG_EOC_INJ>,
OUT<LV_ERR_SEG_EOC_INJ>)

at engine stop to engine run and at clearing of error memory:
CTR_SEG_EOC_INJ[ ] = 0 // all NC_PBK_IV_NR elements
CTR_SEG_EOC_INJ_ERR[ ] = 0
// all NC_PBK_IV_NR elements

Recurrence: segment synchronous

Description:

This function shall detect a locked injection driver on one or more injection banks indiviually. The detection criteria are missing end of cycle (EOC) events. Theoretically, only a limited number of segments may occure between two EOC events on one injection bank. If more than this amount of segments occure before the next EOC event, the conclusion is that EOC events are missing. To get the injection drivers alive again, it is set to mode DISABLED and then to SEQUENTIAL again to trigger a resynchronization. Also, a failure memory entry is done.

Configuration for diagnostic symptoms:

Diagnostic SEG_EOC_INJ	Symptom description	Symptom	Filter type
Injection dirver lock detection diagnosis	Injection driver lock detection on injection bank 0 (physical bank 1)	*SYM_0*	NO
	Injection driver lock detection on injection bank 1 (physical bank 2)	*SYM_1*	
	not used	*SYM_2*	
	not used	*SYM_3*	

Formula Section:

Reset condition flag and error symptom flag for diagnosis function:

LV_CDN_DIAG_SEG_EOC_INJ = 0
ERR_SYM_SEG_EOC_INJ = NO_SYM

```
(0)IF C_CTR_SEG_EOC_INJ_MAX_PLAUS > 0
      // Function is disabled in case of C_CTR_SEG_EOC_INJ_MAX_PLAUS = 0
(0)THEN
  Segment counter since last EOC event:
  (1)FOR i = 0 TO NC_PBK_IV_NR – 1 DO
    (2)IF STATE_PBK_IV_INI = "INIT_FINISHED"    // injection banks initialized ...
      AND (STATE_ERR_PBK_IV_ST_RAW[i] = 0       // ... without errors or ...
             OR STATE_ERR_PBK_IV_ST_RAW[i] = 10hex)
                                                // ...  only with trim pulse error
      AND STATE_INJ_DR = "RUN"                  // injection driver is not in disable mode
      AND STATE_INJ_CRASH_ACT = 00hex           // injection driver is not in static mode
                                                           due to crash reaction
      // injection driver is not in static mode due to test mode
                 (LV_IV_TEST_MOD_AUTH is always zero => test mode is nerver active)
    (2)THEN
      Increment CTR_SEG_EOC_INJ[i] by 1
      Set diagnosis condition flag to indicate that diagnosis is active:
      LV_CDN_DIAG_SEG_EOC_INJ = 1
    (2)ELSE
      CTR_SEG_EOC_INJ[i] = 0
      Reset diagnosis condition flag to indicate that diagnosis is inactive:
      LV_CDN_DIAG_SEG_EOC_INJ = 0
    (2)ENDIF
    Check segment counter for driver lock detection threshhold:
    (3)IF CTR_SEG_EOC_INJ[i] >= C_CTR_SEG_EOC_INJ_MAX_PLAUS
    (3)THEN
      Fault reaction -> Set injection driver to DISABLED and then to SEQUENTIAL again:
      ACTION_INFR_SetInjMode(IN Injection_mode= DISABLE)
      ACTION_INFR_SetInjMode(IN Injection_mode= SEQUENTIAL)
      Increment CTR_SEG_EOC_INJ_ERR[i] by 1
      Reset segment counter:
      CTR_SEG_EOC_INJ[ ] = 0 // all NC_PBK_IV_NR elements
      Set diagnosis error symptom flag to indicate that an error was detected:
      (3a)IF i = 0
      (3a)THEN // error on bank 0
        ERR_SYM_SEG_EOC_INJ = SYM_0
      (3a)ELSE // error on bank 1
        ERR_SYM_SEG_EOC_INJ = SYM_1
      (3a)ENDIF
    (3)ELSE
```

```
      ERR_SYM_SEG_EOC_INJ = NO_SYM
    (3)ENDIF
    Diagnosis and error management handling:
    (4)IF LV_CDN_DIAG_SEG_EOC_INJ = 1
    (4)THEN // diagnosis is active
      (5)IF ERR_SYM_SEG_EOC_INJ ≠ NO_SYM
      (5)THEN // an error was detected
        ACTION_ERRM_NoFilterSymptom( IN<NC_IDX_DIAG_SEG_EOC_INJ>,
            IN<LV_CDN_DIAG_SEG_EOC_INJ>, IN<ERR_SYM_SEG_EOC_INJ>,
            IN<lv_err_set = 1>, IN< lv_err_reset = 0>,
            IN<lv_end_diag = 1>, OUT<LV_ERR_SEG_EOC_INJ> )
      (5)ELSE // no error was detected
        ACTION_ERRM_NoFilterSymptom( IN<NC_IDX_DIAG_SEG_EOC_INJ>,
            IN<LV_CDN_DIAG_SEG_EOC_INJ>, IN <ERR_SYM_SEG_EOC_INJ>,
            IN<lv_err_set = 0>, IN< lv_err_reset = 1>,
            IN<lv_end_diag = 1>, OUT<LV_ERR_SEG_EOC_INJ> )
      (5)ENDIF
   (4)ENDIF
  (1)ENDFOR
(0)ELSE
  // Diagnosis is manually disabled -> set end diag flag
  ACTION_ERRM_NoFilterSymptom( IN<NC_IDX_DIAG_SEG_EOC_INJ>,
            IN<lv_cdn_diag = 1>, IN <ERR_SYM_SEG_EOC_INJ>,
            IN<lv_err_set = 0>, IN< lv_err_reset = 1>,
            IN<lv_end_diag = 1>, OUT<LV_ERR_SEG_EOC_INJ> )
(0)ENDIF
```

41.31.9.2 Detection of locked injection driver – end of cycle synchronous task

Application conditions :

Activation:	every engine state
Deactivation:	-
Initialization:	-
Recurrence:	INJ EOC synchronous (every end of cycle trigger)

Description:

At an EOC event, the segment counter for the bank that reached the EOC shall be reset to zero to indicate that the driver bank is still alive.

Formula Section:

Note: x represents the cylinder number that reached its EOC. The EOC trigger provides the cylinder number x that reached its EOC.

Reset segment counter for the injection bank that reached an EOC event:

CTR_SEG_EOC_INJ[NC_IDX_CYL_PBK_IV_REF[x]] = 0

41.32 Injection valves diagnosis

Overview

The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements. STATE_ERR_IV is the general error state after de-bouncing for limp-home reactions.

All symptoms of the current error code are handled by anti-bouncing.
We assume that only one symptom of an error code can be active at the same time.

The error detection is effected via the ECU hardware.

Please Note:
A valid diagnosis of all failures can only be detected when the injector was operated at least one time since the last changes and the filter times are observed.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_DIAG_IV	V/O	0...FFH	0...255	1	[-]
Error pattern without debouncing for all injectors					
LV_ERR_IV[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Error flag for IVx derived from STATE_ERR_IV (only for display)					
STATE_ERR_IV	V/O	0...FFH	0...255	1	[-]
Error pattern after debouncing for all injectors					
LV_CDN_DIAG_IV[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Diagnosis condition for injector [x]					
LV_END_DIAG_IV[NC_CYL_NR]	V	0...1H	0...1	1	[-]
End of diagnosis for injector [x]					
ERR_DIAG_IV[NC_CYL_NR]	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for injector [x]					
ERR_SYM_IV[NC_CYL_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom for injector [x] filtered with CDN_DIAG_IV [NC_CYL_NR]					
CDN_DIAG_IV[NC_CYL_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Diagnosis condition for each symptom of injector [x]					
CDN_DIAG_IV_REQ[NC_CYL_NR]	V/O	0...FFH	0...255	1	[-]
Diagnosis condition request to I/O-SW for each symptom of injector [x]					
LV_ERR_IV_SC[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Error flag for indication of a detailed short circuit error information (only for display)					
LV_CDN_DIAG_IV_SC[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Short circuit detailed error diagnosis condition for injector [x]					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_IV_SC[NC_CYL_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Short circuit detailed error symptom for injector [x]					
ERR_SYM_IV_SC_TMP[NC_CYL_NR]	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Short circuit detailed error symptom for temporary calculations					
LV_END_DIAG_IV_SC[NC_CYL_NR]	V	0...1H	0...1	1	[-]
End of short circuit detailed error diagnosis for injector [x]					
STATE_ERR_IV_CYL[NC_CYL_NR]	V	0...FFH	0...255	1	[-]
Detailed error information of injection valve, cylinder individual					
STATE_ERR_IV_CYL_TMP[NC_CYL_NR]	-	0...FFH	0...255	1	[-]
Temporary detailed error information of injection valve, cylinder individual					
NR_CYL_DIAG_IV	V	0...7H	0...7	1	[-]
Number of the logical cylinder, which has done its injection valve diagnosis cycle					
NR_CYL_INJ_BAS_DIAG_IV_TMP	-	0...7H	0...7	1	[-]
Temporary number of the current cylinder, which has reached its CRK_INJ_BAS[x]-Event					
NR_CYL_INJ_BAS_PREV_DIAG_IV_TMP	-	0...7H	0...7	1	[-]
Temporary number of the previous cylinder, which had reached its CRK_INJ_BAS[x]-Event					

Input data:

LV_IGK	LV_INH_DIAG_IV	N_32	NC_CYL_NR
LV_CDN_VB_OBD1	CDN_DIAG_IV_RAW[NC_CYL_NR]	LV_FL	LV_ES
STATE_ERR_IV_CYL_RAW[NC_CYL_NR]	NR_CYL_INJ_BAS	NR_CYL_INJ_BAS_PREV	CTR_ABC_IV[NC_CYL_NR]
TI_1_MES[NC_CYL_NR]			

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>,OUT<LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result
ACTION_ERRM_NoFilterSymptom(IN< XX >, IN< lv_cdn_diag_XX >, IN< err_sym_XX >, IN< lv_err_set_XX >, IN< lv_err_reset_XX >, IN< lv_end_diag_XX >, OUT< LV_ERR_XX >)
This action computes the elementary treatment case no filtering is used

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_IV	1	0...FFH	0...255	1	[-]
Anti-bounce increment value					
C_ABC_MAX_IV	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value					
C_N_CDN_DIAG_IV_FL_THD	1	0...FFH	0...8160	32	[rpm]
Threshold disable diagnosis at full load					
C_TI_THD_INH_DIAG_IV	1	0...FFFFH	0...65.535	0.001	[ms]
Injection time threshold below the IV diagnosis will be inhibited					
C_INH_DIAG_IV_SWI_MAN	1	0...FFH	0...255	1	[-]
Manual switch pattern for cylinder individual inhibition of injection valve diagnosis (bit x = 1, diagnosis of cylinder x will be inhibited)					

Configuration for diagnostic symptoms:

Diagnosis	Symptom	Nr	ABC type
Injection valves diagnosis	*Open circuit (OC)*	*0*	MPL_STD_INI
	Short circuit (SC)	*1*	
	High side to low side short circuit (HL)	*2*	
	Open circuit at charged injector (OCC)	*3*	
IV[x]			

Diagnosis	Symptom	Nr	ABC type
Injection valves diagnosis short circuit detailed	*High side to ground short circuit (HG)*	*0*	NO_FIL
	Low side to battery short circuit (LB)	*1*	
	High side to battery short circuit (HB)	*2*	
	Low side to ground short circuit (LG)	*3*	
IV_SC[x]			

FUNCTION DESCRIPTION:

Application conditions:

Activation :	LV_IGK = 1
Deactivation:	at transition of LV_IGK 1 -> 0 DO: LV_CDN_DIAG_IV[NC_CYL_NR] = 0 LV_CDN_DIAG_IV_SC[NC_CYL_NR] = 0 CDN_DIAG_IV[x] = 0 CDN_DIAG_IV_REQ[x] = 0
Initialization:	at reset , at transition of LV_IGK 0 -> 1 and at global clearing of failure memory DO: STATE_DIAG_IV = 0 STATE_ERR_IV = 0 STATE_ERR_IV_CYL[NC_CYL_NR] = 0 LV_ERR_IV_SC[NC_CYL_NR] = 0 ERR_SYM_IV_SC[NC_CYL_NR] = NO_SYM LV_CDN_DIAG_IV_SC[NC_CYL_NR] = 0 LV_END_DIAG_IV_SC[NC_CYL_NR] = 0 all other output data according filter type by ERRM
Recurrence:	segment synchronous

Formula section:

41.32.1 Electrical Injection Valve Diagnosis – selection of the current cylinder number

Determine the previous cylinder:

NR_CYL_INJ_BAS_PREV_DIAG_IV_TMP = NR_CYL_INJ_BAS_PREV

Determine the current cylinder:

NR_CYL_INJ_BAS_DIAG_IV_TMP = NR_CYL_INJ_BAS

Check cylinder numbers under consideration of re-entrance capability:

```
(1) IF NR_CYL_INJ_BAS_ PREV_DIAG_IV_TMP ≠ NR_CYL_INJ_BAS_PREV
(1) THEN
    NR_CYL_INJ_BAS_PREV_DIAG_IV_TMP = NR_CYL_INJ_BAS_PREV
    NR_CYL_INJ_BAS_DIAG_IV_TMP = NR_CYL_INJ_BAS
(1) ENDIF
```

41.32.2 Electrical Injection Valve Diagnosis – diagnosis routine call

```
(1) IF NR_CYL_DIAG_IV = NR_CYL_INJ_BAS_DIAG_IV_TMP
(1) THEN
    Nothing to do! The last diagnosis cycle is equal to the current.
(1) ELSE
    Diagnosis Cycle has to be done:
    (1) IF NR_CYL_DIAG_IV ≠ NR_CYL_INJ_BAS_PREV_DIAG_IV_TMP
    (1) THEN
        Diagnosis Cycle has to be done for the previous cylinder, because its missing:
        EXECUTE "41.32.3 Common Diagnosis Operations" for cylinder x =
                                      NR_CYL_INJ_BAS_PREV_DIAG_IV_TMP
    (1) ENDIF
    Diagnosis Cycle has to be done for the current cylinder:
    EXECUTE "41.32.3 Common Diagnosis Operations" for cylinder x =
                                      NR_CYL_INJ_BAS_DIAG_IV_TMP
    Update cylinder number of the done diagnosis cycle:
    NR_CYL_DIAG_IV = NR_CYL_INJ_BAS_DIAG_IV_TMP
(1) ENDIF
```

41.32.3 Common Diagnosis Operations

This section works like a subroutine call in C-code programs
Note! x represents the current assigned active cylinder number from calling section.

```
(1) IF   (LV_INH_DIAG_IV = 0)
         AND
         (LV_CDN_VB_OBD1 = 1)
         AND
         (LV_ES = 0)
         AND
         (LV_FL = 0)
         AND
         (TI_1_MES[x] >= C_TI_THD_INH_DIAG_IV)
         AND
         (N_32 < C_N_CDN_DIAG_IV_FL_THD)
         AND
         (Bit x of C_INH_DIAG_IV_SWI_MAN = 0)
```

```
(1) THEN
        Valid diagnosis conditions:
        CDN_DIAG_IV_REQ[x] = 0xFF (All errors should be detected)
        CDN_DIAG_IV[x] = CDN_DIAG_IV_RAW[x]
        STATE_ERR_IV_CYL_TMP[x] = STATE_ERR_IV_CYL_RAW[x] AND CDN_DIAG_IV[x]
                                                                  ... bitwise
        STATE_ERR_IV_CYL[x] = (STATE_ERR_IV_CYL[x]
                               OR STATE_ERR_IV_CYL_TMP[x])        ... bitwise
                               AND 0xFF                           ... bitwise
        CDN_DIAG_IV[x] = (CDN_DIAG_IV[x] AND 0x03)                ... bitwise
        (5) IF (CDN_DIAG_IV[x] = SYM_1)      (SC-Error detectable)
              OR
              (CDN_DIAG_IV[x] = (SYM_1 OR SYM_0))                 ... bitwise
        (5) THEN
          CDN_DIAG_IV[x] = CDN_DIAG_IV[x] OR SYM_2                ... bitwise
        (5) ENDIF
        (6) IF (CDN_DIAG_IV[x] = SYM_0)      (OC-Error detectable)
              OR
              (CDN_DIAG_IV[x] = (SYM_1 OR SYM_0))                 ... bitwise
        (6) THEN
          CDN_DIAG_IV[x] = CDN_DIAG_IV[x] OR SYM_3                ... bitwise
        (6) ENDIF
(1) ELSE
        CDN_DIAG_IV[x] = 0
        CDN_DIAG_IV_REQ[x] = 0
        STATE_ERR_IV_CYL_TMP[x] = 0
(1) ENDIF
```

Error Symptom calculation (raw value from I/O SW)

```
ERR_DIAG_IV[x] = 0
(7) IF      (STATE_ERR_IV_CYL_TMP[x] AND 0xFF) ≠ 0
(7) THEN
    (8) IF (STATE_ERR_IV_CYL[x] AND 0x7E) ≠ 0
    (8) THEN
        (9) IF (STATE_ERR_IV_CYL[x] AND 0x40) ≠ 0
        (9) THEN
            ERR_DIAG_IV[x] = SYM_2      HL-Error detected
        (9) ELSE
            ERR_DIAG_IV[x] = SYM_1      SC-Error detected
```

```
            (9) ENDIF
        (8) ELSE
            (10) IF (STATE_ERR_IV_CYL[x] AND 0x80) ≠ 0
            (10) THEN
                ERR_DIAG_IV[x] = SYM_3       OCC-Error detected
            (10) ELSE
                ERR_DIAG_IV[x] = SYM_0       OC-Error detected
            (10) ENDIF
        (8) ENDIF
    (7) ENDIF
```

ACTION_ERRM_FilterMulticondition (IV[x], CDN_DIAG_IV[x], ERR_DIAG_IV[x], LV_END_DIAG_IV[x], LV_ERR_IV[x], LV_CDN_DIAG_IV[x], ERR_SYM_IV[x])

Note: For failure debouncing and error management treatment the multicondition debounce algorithm (part of Error Management AGGR) is used with the parameters CDN_DIAG_IV[x] and ERR_DIAG_IV[x].

This algorithm determines:

ERR_SYM_IV [x] (= raw value ERR_DIAG_IV [x] filtered with CDN_DIAG_IV [x])

and

LV_ERR_IV [x] (Error flag for debounced error of Injector [x])

and

LV_CDN_DIAG_IV [x]

and

LV_END_DIAG_IV [x]

Calculation of STATE_DIAG_IV:

```
(11) IF      ERR_SYM_IV [x] <> 0
(11) THEN    Bit x of STATE_DIAG_IV = 1
(11) ELSE    Bit x of STATE_DIAG_IV = 0
(11) ENDIF
```

Calculation of STATE_ERR_IV:

Bit x of STATE_ERR_IV = LV_ERR_IV [x]

Calculation of detailed SC error information:

```
(12) IF LV_ERR_IV[x] 0 → 1
(12) THEN
    (13) IF Bit HG of STATE_ERR_IV_CYL[x] is set
    (13) THEN
        ERR_SYM_IV_SC_TMP = SYM_0 (HG-Error detected)
    (13) ELSE
        (14) IF Bit LB of STATE_ERR_IV_CYL[x] is set
        (14) THEN
            ERR_SYM_IV_SC_TMP = SYM_1 (LB-Error detected)
        (14) ELSE
            (15) IF Bit HB of STATE_ERR_IV_CYL[x] is set
            (15) THEN
                ERR_SYM_IV_SC_TMP = SYM_2 (HB-Error detected)
            (15) ELSE
                (16) IF Bit LG of STATE_ERR_IV_CYL[x] is set
                (16) THEN
                    ERR_SYM_IV_SC_TMP = SYM_3 (LG-Error detected)
                (16) ELSE
                    ERR_SYM_IV_SC_TMP = NO_SYM (Not a detailed SC-Error was detected
                                                or HL-Error was detected)
                (16) ENDIF
            (15) ENDIF
        (14) ENDIF
    (13) ENDIF
    ERR_SYM_IV_SC[x] = ERR_SYM_IV_SC_TMP
    (17) IF ERR_SYM_IV_SC_TMP = NO_SYM
    (17) THEN
        ACTION_ERRM_NoFilterSymptom( IN< IV_SC >,
        IN< LV_CDN_DIAG_IV[x] >, IN <ERR_SYM_IV_SC[x] >,
        IN< lv_err_set_IV_SC = 0>, IN< lv_err_reset_IV_SC = 1>,
        IN< LV_END_DIAG_IV[x] >, OUT< LV_ERR_IV_SC> )
    (17) ELSE
        ACTION_ERRM_NoFilterSymptom( IN< IV_SC >,
        IN< LV_CDN_DIAG_IV[x] >, IN <ERR_SYM_IV_SC[x] >,
        IN< lv_err_set_IV_SC = 1>, IN< lv_err_reset_IV_SC = 0>,
        IN< LV_END_DIAG_IV[x] >, OUT< LV_ERR_IV_SC> )
    (17) ENDIF
```

```
(12) ENDIF

(18) IF LV_ERR_IV[x] 1 → 0
(18) THEN
    ERR_SYM_IV_SC[x] = NO_SYM
    ACTION_ERRM_NoFilterSymptom( IN< IV_SC >,
    IN< LV_CDN_DIAG_IV[x] >, IN <ERR_SYM_IV_SC[x] >,
    IN< lv_err_set_IV_SC = 0>, IN< lv_err_reset_IV_SC = 1>,
    IN< LV_END_DIAG_IV[x] >, OUT< LV_ERR_IV_SC> )
(18) ENDIF

(19) IF LV_ERR_IV[x] stays unchanged
(19) THEN
    ACTION_ERRM_NoFilterSymptom( IN< IV_SC >,
    IN< LV_CDN_DIAG_IV[x] >, IN <ERR_SYM_IV_SC[x] >,
    IN< lv_err_set_IV_SC = 0>, IN< lv_err_reset_IV_SC = 0>,
    IN< LV_END_DIAG_IV[x] >, OUT< LV_ERR_IV_SC> )
(19) ENDIF
```

```
(18) IF CTR_ABC_IV[x] = 0
(18) THEN
        STATE_ERR_IV_CYL[x] = 0
(18) ENDIF
```

Please Note! Synchronization of LV_CDN_DIAG_IV_SC[x] with LV_CDN_DIAG_IV[x] and LV_END_DIAG_IV_SC[x] with LV_END_DIAG_IV[x] by parameter passing to ACTION_ERRM_NoFilterSymptom.

41.33 Injection valve diagnosis for ATIC 63 PIEZO injection driver (Application Incidences)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_IV	V/O	0...1H	0...1	1	[-]
Inhibition condition for the IV diagnostic					
LV_ERR_IV_LST_CYC	V/O/S	0...1H	0...1	1	[-]
Flag that indicates if a debounced injector error is present at the end of the driving cycle					
INH_IV_DIAG_ERR	V/O	0...FFH	0...255	1	[-]
Cylinder shut off request from injection valve diagnosis (cylinder allocated)					
INJ_DR_CYL_REF[NC_CYL_NR]	O/V	0...FFH	0...255	1	[-]
Assignment between logical cylinders and injector drivers					

Input data:

STATE_ERR_IV	LV_IGK	TCO	ERR_SYM_IV[NC_CYL_NR]
NC_CYL_NR	NC_IDX_CYL_PBK_IV_REF[NC_CYL_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_TCO_THD_INH_DIAG_IV	1	0...FEH	-48...142.5	0.75	[°C]
TCO threshold below which the injector diagnosis is inhibited					
LC_INH_IV_DIAG_ERR_ENA	1	0...1H	0...1	1	[-]
Manual switch for activation of INH_IV_DIAG_ERR					

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_IV deactivates (freezes) the power stage diagnostic of all injection valves.

The injector diagnosis has to be deactivated in case of:

- TCO is below the threshold C_TCO_THD_INH_DIAG_IV

If a de-bounced injector error is present at the end of the driving cycle (LV_ERR_IV = 1), the flag LV_ERR_IV_LST_CYC is set and is stored in the permanent RAM.

41.33.1 Calculation of INJ_DR_CYL_REF[NC_CYL_NR]

Application conditions:

Activation:	every engine state
Deactivation:	-
Initialisation:	-

Recurrence: only once at reset

Formula section:

The variable INJ_DR_CYL_REF[x] represents an inhibit pattern where all cylinders on the same bank as cylinder x are inhibited (corresponding bits are set to 1) due to injection valve errors.

Calculation of INJ_DR_CYL_REF[NC_CYL_NR]:
Note: m reveres to the injection power stage bank within this loop

```
FOR x = 0 TO NC_CYL_NR – 1 DO
    INJ_DR_CYL_REF[x] = 0
    m = NC_IDX_CYL_PBK_IV_REF[x]
    FOR y = 0 TO NC_CYL_NR – 1 DO
        IF NC_IDX_CYL_PBK_IV_REF[y] = m  // cylinder y is on bank m
        THEN
            Bit y of INJ_DR_CYL_REF[x] = 1
        ENDIF
    ENDFOR
ENDFOR
```

41.33.2 Calculation of LV_INH_DIAG_IV

Application conditions:

Activation: every engine state

Deactivation: -

Initialisation: at reset: LV_INH_DIAG_IV = 1

Recurrence: 100 ms and 'at first valid tooth'

Formula section:

```
(1) IF
TCO < C_TCO_THD_INH_DIAG_IV
(1) THEN
LV_INH_DIAG_IV = 1
(1) ELSE
LV_INH_DIAG_IV = 0
(1) ENDIF
```

41.33.3 Calculation of LV_ERR_IV_LST_CYC

Application conditions:

Activation: LV_IGK = 1 → 0

Deactivation: -

Initialisation: -

Recurrence: only once at LV_IGK = 1 → 0

Formula section:

(1) IF

STATE_ERR_IV ≠ 0

(1) THEN

LV_ERR_IV_LST_CYC = 1

(1) ELSE

LV_ERR_IV_LST_CYC = 0

(1) ENDIF

Note: In case of "checksum error" after power latch or reset the flag LV_ERR_IV_LST_CYC has to be set to zero.

41.33.4 Shut off request from injection valve diagnosis

General information:

In this chapter the diagnosis information for ATIC 63 PIEZO driver will be analysed. In the case of a short circuit error (SC) except the case of a high side to low side short circuit error (HL) all the injectors, which are connected to the same injection driver have to be switched off to prevent wrong fuel mass on not affected injection valves. In the case of an OC or HL error only the injector with a diagnosed error has to be switched off.

Application conditions:

Activation: LV_IGK = 1

Deactivation: -

Initialisation: at reset, at transition of LV_IGK 0 -> 1 and at global clearing of failure memory DO:
INH_IV_DIAG_ERR = 0

Recurrence: segment synchronous

Note: The chapter 41.33.4 Shut off request from injection valve diagnosis has to be calculated after the module "Injection valve diagnosis".

Formula section:

```
(1) IF (STATE_ERR_IV ≠ 0) AND (LC_INH_IV_DIAG_ERR_ENA = 1)
(1) THEN
      INH_IV_DIAG_ERR = STATE_ERR_IV
      (Switch off all injectors with an error symptom)
      (1) FOR x = 0 TO (NC_CYL_NR –1 ) DO:
        (2) IF (Bit x of STATE_ERR_IV = 1) AND (ERR_SYM_IV[x] = 'SC-Error')
              (Check if a short circuit error (without HL-error) was detected)
        (2) THEN
              (Switch off all the injectors which are connected to the same driver additionally)
              INH_IV_DIAG_ERR =          INH_IV_DIAG_ERR
                                    OR                                    ...bitwise
                                         INJ_DR_CYL_REF[x]
        (2) END IF
      (1) END FOR
(1) ELSE
      INH_IV_DIAG_ERR = 0
(1) ENDIF
```

41.34 Injection valves power stage diagnosis

Overview

The purpose of the injection valve power stage diagnosis is to detect electrical faults of used ATIC drivers. STATE_ERR_PBK_IV is the general error state after de-bouncing for limp-home reactions.

All symptoms of the current error code are handled by anti-bouncing.
We assume that only one symptom of an error code can be active at the same time.

The error detection is effected via the ECU hardware.

Please Note:
A valid diagnosis of all failures can only be detected when the injector was operated at least one time since the last changes and the filter times are observed.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_DIAG_PBK_IV	V/O	0...FFH	0...255	1	[-]
Error pattern without debouncing for all injection valve power stages					
LV_ERR_PBK_IV[NC_PBK_IV_NR]	V	0...1H	0...1	1	[-]
Error flag for injection valve power stage bank errors derived from STATE_ERR_PBK_IV (only for display)					
STATE_ERR_PBK_IV	V/O	0...FFH	0...255	1	[-]
Error pattern after debouncing for all injection valve power stages					
LV_CDN_DIAG_PBK_IV[NC_PBK_IV_NR]	V	0...1H	0...1	1	[-]
Diagnosis condition for injection valve power stage bank					
LV_END_DIAG_PBK_IV[NC_PBK_IV_NR]	V	0...1H	0...1	1	[-]
End of diagnosis for injection valve power stage bank					
ERR_DIAG_PBK_IV[NC_PBK_IV_NR]	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for injection valve power stage bank					
ERR_SYM_PBK_IV[NC_PBK_IV_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom for injection valve power stage bank filtered with CDN_DIAG_PBK_IV					
CDN_DIAG_PBK_IV[NC_PBK_IV_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Diagnosis condition for each error symptom of injection valve power stage bank					
STATE_ERR_SYM_PBK_IV[NC_PBK_IV_NR]	V/O	0...FFH	0...255	1	[-]
Detailed error symptom information of injection valve power stages, ATIC bank individual					
STATE_ERR_SYM_PBK_IV_ST[NC_PBK_IV_NR]	V/O	0...FFH	0...255	1	[-]
Detailed error symptom information of injection valve power stages during startup phase, ATIC bank individual					

Input data:

LV_ES	STATE_ERR_PBK_IV_RAW[NC_PBK_IV_NR]	CDN_DIAG_PBK_IV_RAW[NC_PBK_IV_NR]	STATE_ERR_PBK_IV_ST_RAW[NC_PBK_IV_NR]
STATE_PBK_IV_INI	LV_INH_DIAG_PBK_IV	NC_PBK_IV_NR	LV_IGK
SEG_NR			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_PBK_IV	1	0...FFH	0...255	1	[-]
Anti-bounce increment value for injection valve power stage diagnosis					
C_ABC_MAX_PBK_IV	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value for injection valve power stage diagnosis					
C_STATE_ERR_SYM_PBK_IV_INH	1	0...FFH	0...255	1	[-]
Manual inhibition pattern for injection valve power stage online diagnosis symptom (bit x = 1, diagnosis symptom will be inhibited)					
C_STATE_ERR_SYM_PBK_IV_ST_INH	1	0...FFH	0...255	1	[-]
Manual inhibition pattern for injection valve power stage startup diagnosis symptom (bit x = 1, diagnosis symptom will be inhibited)					

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>,OUT<LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result

Configuration for diagnostic symptoms:

Diagnosis	Symptom	Nr	ABC type
Injection valves power stage diagnosis	*Power stage startup error*	*0*	MPL_STD_INI
	Power stage online error	*1*	
	Power stage discharge error	*2*	
PBK_IV[p]			

FUNCTION DESCRIPTION:

Application conditions:

Activation : (LV_IGK = 1) **AND** (STATE_PBK_IV_INI = "INIT_FINISHED")

Deactivation: at transition of LV_IGK 1 -> 0 DO:
LV_CDN_DIAG_PBK_IV[NC_PBK_IV_NR] = 0
CDN_DIAG_PBK_IV[NC_PBK_IV_NR] = NO_SYM

Initialization: at reset , at transition of LV_IGK 0 -> 1 and at global clearing of failure memory DO:
STATE_DIAG_PBK_IV = 0
STATE_ERR_PBK_IV = 0
STATE_ERR_SYM_PBK_IV[NC_PBK_IV_NR] = 0
STATE_ERR_SYM_PBK_IV_ST[NC_PBK_IV_NR] = 0
all other output data according filter type by ERRM

Recurrence: LV_ES = 0: segment synchronous
LV_ES = 1: 10 ms

Formula section:

41.34.1 Common injection valve power stage diagnosis operations

```
(1) FOR p = 0 TO (NC_PBK_IV_NR -1) DO:
Error Condition calculation:
(1) IF   (LV_INH_DIAG_PBK_IV = 0)
         AND
         (LV_ES = 0)
         AND
         (SEG_NR = 0)
(1) THEN
    CDN_DIAG_PBK_IV[p] = NO_SYM
    (2) IF   (CDN_DIAG_PBK_IV_RAW[p] AND 0x08) = 0x08                  ...bitwise
    (2) THEN
             CDN_DIAG_PBK_IV[p] = SYM_2
    (2) ENDIF
    (3) IF   (CDN_DIAG_PBK_IV_RAW[p] AND 0x07) ≠ 0                     ...bitwise
    (3) THEN
             CDN_DIAG_PBK_IV[p] = CDN_DIAG_PBK_IV[p] OR SYM_1          ...bitwise
    (3) ENDIF
    STATE_ERR_SYM_PBK_IV[p] = STATE_ERR_PBK_IV_RAW[p]
                        AND CDN_DIAG_PBK_IV_RAW[p]                     ...bitwise
                        AND NOT(C_STATE_ERR_SYM_PBK_IV_INH)            ...bitwise
```

```
(1) ELSE
        CDN_DIAG_PBK_IV[p] = NO_SYM
(1) ENDIF
CDN_DIAG_PBK_IV[p] = CDN_DIAG_PBK_IV[p] OR SYM_0                              ...bitwise
STATE_ERR_SYM_PBK_IV_ST[p] = STATE_ERR_PBK_IV_ST_RAW[p]
                  AND NOT(C_STATE_ERR_SYM_PBK_IV_ST_INH)                      ...bitwise
```

Error Symptom calculation:

```
ERR_DIAG_PBK_IV[p] = NO_SYM
(4) IF STATE_ERR_SYM_PBK_IV_ST[p] ≠ 0
(4) THEN
    ERR_DIAG_PBK_IV[x] = SYM_0
    'Power stage startup error' detected
(4) ENDIF
(5) IF (STATE_ERR_SYM_PBK_IV[p] AND 0x08) = 0x08                             ...bitwise
(5) THEN
    ERR_DIAG_PBK_IV[x] = ERR_DIAG_PBK_IV[x] OR SYM_2                          ...bitwise
    'Power stage discharge error' detected
(5) ENDIF
(6) IF (STATE_ERR_SYM_PBK_IV[p] AND 0x07) ≠ 0                                ...bitwise
(6) THEN
    ERR_DIAG_PBK_IV[x] = ERR_DIAG_PBK_IV[x] OR SYM_1                          ...bitwise
    'Power stage online error' detected
(6) ENDIF
```

Error Symptom debouncing:

ACTION_ERRM_FilterMulticondition (PBK_IV[p], CDN_DIAG_PBK_IV[p], ERR_DIAG_PBK_IV[p], LV_END_DIAG_PBK_IV[p], LV_ERR_PBK_IV[p], LV_CDN_DIAG_PBK_IV[p], ERR_SYM_PBK_IV[p])

Note: For failure debouncing and error management treatment the multicondition debounce algorithm (part of Error Management AGGR) is used with the parameters CDN_DIAG_PBK_IV[p] and ERR_DIAG_PBK_IV[p].

This algorithm determines:

ERR_SYM_PBK_IV[p] *(= raw value ERR_DIAG_PBK_IV[p] filtered with CDN_DIAG_PBK_IV[x])*

and

LV_ERR_PBK_IV[p] *(Error flag for debounced error of injection valve power stage [p])*

and

LV_CDN_DIAG_PBK_IV[p]

and

```
LV_END_DIAG_PBK_IV[p]

        Calculation of STATE_DIAG_PBK_IV:
        (7) IF        ERR_SYM_PBK_IV[p] <> 0
        (7) THEN      Bit p of STATE_DIAG_PBK_IV = 1
        (7) ELSE      Bit p of STATE_DIAG_PBK_IV = 0
        (7) ENDIF

        Calculation of STATE_ERR_PBK_IV:
        Bit p of STATE_ERR_PBK_IV = LV_ERR_PBK_IV[p]

    (1) ENDFOR
```

41.35 Injection valve power stage diagnosis for PIEZO injection driver (Application Incidences)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_PBK_IV	V/O	0...1H	0...1	1	[-]
Inhibition condition for the injection valve power stage diagnosis					
LV_ERR_PBK_IV_LST_CYC	V/O/S	0...1H	0...1	1	[-]
Flag that indicates if a debounced injection valve power stage error is present at the end of the driving cycle					
INH_PBK_IV_DIAG_ERR	O/V	0...FFH	0...255	1	[-]
Cylinder shut off request from injection valve power stage diagnosis (cylinder allocated)					
INJ_DR_PBK_IV_REF[NC_PBK_IV_NR]	O/V	0...FFH	0...255	1	[-]
Assignment between logical cylinders and injection power banks (power bank individual)					

Input data:

LV_IGK	STATE_ERR_PBK_IV	STATE_PBK_IV_INI	ERR_SYM_PBK_IV[NC_PBK_IV_NR]
STATE_ERR_SYM_PBK_IV_ST[NC_PBK_IV_NR]	NC_PBK_IV_NR	NC_CYL_NR	NC_IDX_CYL_PBK_IV_REF[NC_CYL_NR]

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
LC_DIAG_INH_PBK_IV_ERR_ENA	1	0...1H	0...1	1	[-]
Flag to enable inhibition of cylinders due to injection power bank errors					
LC_DIAG_INH_PBK_IV_ERR_ST_ENA	1	0...1H	0...1	1	[-]
Flag to enable inhibition of cylinders due to injection power bank start up errors					

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_PBK_IV deactivates (freezes) the power stage online diagnostic of all injection valve drivers.

If a de-bounced injector error is present at the end of the driving cycle (LV_ERR_PBK_IV = 1), the flag LV_ERR_PBK_IV_LST_CYC is set and is stored in the non-volatile memory.

The variable INJ_DR_PBK_IV_REF[NC_PBK_IV_NR] represents an inhibit pattern, which inhibits all cylinders assigned to the injection power bank revered to by the array index.

41.35.1 Calculation of INJ_DR_PBK_IV_REF[NC_PBK_IV_NR]

Application conditions:

Activation:	every engine state
Deactivation:	-
Initialisation:	-

Recurrence: only once at reset

Formula section:

The variable INJ_DR_PBK_IV_REF[p] represents an inhibit pattern where all cylinders on bank m are inhibited (corresponding bits are set to 1) due to injection power bank errors.

Initial value:

```
FOR p = 0 TO NC_PBK_IV_NR – 1 DO
    INJ_DR_PBK_IV_REF[p] = 0
ENDFOR
```

Calculation of INJ_DR_PBK_IV_REF[NC_PBK_IV_NR]:

```
FOR x = 0 TO NC_CYL_NR – 1 DO
    Bit x of INJ_DR_PBK_IV_REF[NC_IDX_CYL_PBK_IV_REF[x]] = 1
ENDFOR
```

41.35.2 Calculation of LV_INH_DIAG_PBK_IV

Application conditions:

Activation: (LV_IGK = 1) AND (STATE_PBK_IV_INI = "INIT_FINISHED")

Deactivation: -

Initialisation: at reset, at transition of LV_IGK 0 -> 1 DO:
LV_INH_DIAG_PBK_IV = 1
INH_PBK_IV_DIAG_ERR = 0

Recurrence: segment synchronous

Formula section:

```
LV_INH_DIAG_PBK_IV = 0

(1)IF STATE_ERR_PBK_IV ≠ 0
(1)THEN
    (1)FOR p = 0 TO NC_PBK_IV_NR – 1 DO
          (2)IF (Bit p of STATE_ERR_PBK_IV = 1)
             AND (ERR_SYM_PBK_IV[p] BitwiseAND SYM_0 = SYM_0)
             AND (STATE_ERR_SYM_PBK_IV_ST[p] ≠ 0x10) // no trim pulse error
             AND (LC_DIAG_INH_PBK_IV_ERR_ST_ENA = 1)
          (2)THEN
```

```
                INH_PBK_IV_DIAG_ERR = INH_PBK_IV_DIAG_ERR BitwiseOR
                                      INJ_DR_PBK_IV_REF[p]
            (2)ENDIF
            (3)IF (Bit p of STATE_ERR_PBK_IV = 1)
               AND (ERR_SYM_PBK_IV[p] ≠ SYM_0)
               AND (LC_DIAG_INH_PBK_IV_ERR_ENA = 1)
            (3)THEN
                INH_PBK_IV_DIAG_ERR = INH_PBK_IV_DIAG_ERR BitwiseOR
                                      INJ_DR_PBK_IV_REF[p]
            (3)ENDIF
        (1)ENDFOR
    (1)ELSE
        INH_PBK_IV_DIAG_ERR = 0
    (1)ENDIF
```

41.35.3 Calculation of LV_ERR_PBK_IV_LST_CYC

Application conditions:

Activation:	LV_IGK = 1 → 0
Deactivation:	-
Initialisation:	-
Recurrence:	only once at LV_IGK = 1 → 0

Formula section:

```
(1) IF
STATE_ERR_PBK_IV ≠ 0
(1) THEN
LV_ERR_PBK_IV_LST_CYC = 1
(1) ELSE
LV_ERR_PBK_IV_LST_CYC = 0
(1) ENDIF
```

Note: In case of “checksum error” after power latch or reset the flag LV_ERR_PBK_IV_LST_CYC has to be set to zero.

41.36 Injector classification programming diag

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_PLAUS_IV_CAL	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom PLAUS_IV_CAL					
LF_ERR_PLAUS_IV_EGY_CAL	O/V	0...FFFFH	0...65535	1	[-]
cylinder individuell bit coded not plausible injector valve coding values "Energie" (1 - not plausible, 0 - plausible)					
LF_ERR_PLAUS_IV_MFF_CAL	O/V	0...FFFFH	0...65535	1	[-]
cylinder individuell bit coded not plausible injector valve coding values "Kleinmenge" (1 - not plausible, 0 - plausible)					
LV_CDN_DIAG_PLAUS_IV_CAL	V	0...1H	0...1	1	[-]
Diagnosis condition PLAUS_IV_CAL					
LV_END_DIAG_PLAUS_IV_CAL	V	0...1H	0...1	1	[-]
End of Diagnosis PLAUS_IV_CAL					
LV_ERR_PLAUS_IV_CAL	O/V	0...1H	0...1	1	[-]
Flag that indicates at least one injector coding is not plausible					

Input data:

EGY_SP_IV_EXT_ADJ[NC_CYL_NR]	LV_IGK	LV_ST_END	MFF_ABSV_IV_EXT_ADJ[NC_CYL_NR]
N	NC_CYL_NR		

Import actions:

ACTION_ERRM_GetLvErr(IN<IDX_DIAG>, OUT<LV_ERR>)
Action that returns the status of the debounced failure
ACTION_ERRM_FilterSymptom(IN<IDX_DIAG>, IN<LV_CDN_DIAG>, IN<ERR_SYM>, IN<ABC_INC>, IN<ABC_DEC>, IN<ABC_MAX>, OUT<LV_ERR>)
This action computes the elementary anti-bounce filter

41.36.1 Checking plausiblity of injector programming value "Kleinmenge"

FUNCTION DESCRIPTION:

General information:

Goal of this chapter is to check the plausiblity of injector programming value "Kleinmenge".

Application conditions:

Initialisation at reset:	first calculation
Recurrence:	1 s

Activation: at every engine operating state

Deactivation: -

Formula section:

```
For i = 0 to (NC_CYL_NR -1)
do
        if MFF_ABSV_IV_EXT_ADJ[i] < C_MFF_ABS_IV_CAL_MIN
          or
          MFF_ABSV_IV_EXT_ADJ[i] > C_MFF_ABS_IV_CAL_MAX
        then
                set bit i of LF_ERR_PLAUS_IV_MFF_CAL to "1"
        else
                reset bit i of LF_ERR_PLAUS_IV_MFF_CAL to "0"
        endif
endfor
```

41.36.2 Checking plausiblity of injector programming value "Energie"

FUNCTION DESCRIPTION:

General information:

Goal of this chapter is to check the plausiblity of injector programming value "Energie".

Application conditions:

Initialisation at reset: first calculation

Recurrence: 1 s

Activation: at every engine operating state

Deactivation: -

Formula section:

```
For i = 0 to (NC_CYL_NR -1)
do
        if  EGY_SP_IV_EXT_ADJ[i] < C_EGY_SP_IV_CAL_MIN
           or
           EGY_SP_IV_EXT_ADJ[i] > C_EGY_SP_IV_CAL_MAX
        then
                set bit i of LF_ERR_PLAUS_IV_EGY_CAL to "1"
        else
                reset bit i of LF_ERR_PLAUS_IV_EGY_CAL to "0"
        endif
endfor
```

41.36.3 Error detection for injector programming plausibility

FUNCTION DESCRIPTION:

General information:

The flag LV_ERR_PLAUS_IV_CAL indicates an at least one not plausible injector programming values. That occurs if injector programming was not done by testerservice or at least one written value by testerservice is out of allowed range.

Description:

Error-symptoms are defined to this diagnosis function as following :

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 |  |  |  └─>-                                    (= SYM_0)
 |  |  └────>-                                    (= SYM_1)
 |  └────────> unplausible Energie                (= SYM_2)
 └───────────> unplausible Kleinmenge             (= SYM_3)
```

Application conditions:

Initialisation: all output data according ABC configuration "**STD_INI**"

// LV_IGK 0->1 ***or*** *reset* ***or*** *at clearing error memory*

This action initialized the diagnostic result according the filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_GS>, OUT<LV_ERR_GS>)

```
Activation:      if      LV_IGK = 1 and LV_ST_END = 0
                 then    LV_CDN_DIAG_PLAUS_IV_CAL = 1
                 endif
Deactivation:    if      LV_IGK = 0  or  LV_ST_END = 1
                 then    LV_CDN_DIAG_PLAUS_IV_CAL = 0
                 endif
Recurrence:      10ms
```

Formula section:

```
if(1)       LF_ERR_PLAUS_IV_EGY_CAL =  0
      and   LF_ERR_PLAUS_IV_MFF_CAL  = 0
then(1)     ERR_SYM_PLAUS_IV_CAL =  NO_SYM
 else(1)
      if(2) N  >=  C_N_MIN_PLAUS_IV_VAL
            if(3)   LF_ERR_PLAUS_IV_EGY_CAL <> 0
            then(3) ERR_SYM_PLAUS_IV_CAL =  SYM_2
            endif(3)
            if(3)   LF_ERR_PLAUS_IV_MFF_CAL <> 0
```

```
            then(3) ERR_SYM_PLAUS_IV_CAL =  SYM_3
            endif(3)
      endif(2)
endif(1)
```

For failure and error management treatment the anti-bounce mechanism is called with the parameters LV_CDN_DIAG_PLAUS_IV_CAL and ERR_SYM_PLAUS_IV_CAL.

```
ACTION_ERRM_FilterSymptom
    ( IN< GS >,
      IN< LV_CDN_DIAG_GS >,
      IN< ERR_SYM_GS >,
      IN< C_ABC_INC_GS >,
      IN< 1 >,
      IN< C_ABC_MAX_GS >,
      OUT< LV_ERR_GS >
    )
```

This algorithm determines LV_ERR_PLAUS_IV_CAL and LV_END_DIAG_PLAUS_IV_CAL and delivers the result to Error Management.

Configuration for diagnostic symptoms:

Diagnostic GS	Symptom description	Symptom	Filter type
Injector programming plausibility diagnosis	-	*SYM_0*	STD_INI
	-	*SYM_1*	
	"Energie"-programming value for at least ore cylinder is not plausible	*SYM_2*	
	"Kleinmengen"-programming value for at least ore cylinder is not plausible	*SYM_3*	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_PLAUS_IV_CAL	1	0...FFH	0...255	1	[-]
Debounce counter increment injection valve programming plausibility diagnosis (default: 1)					
C_ABC_MAX_PLAUS_IV_CAL	1	1...FFH	1...255	1	[-]
Debounce counter maximum value injection valve programming plausibility diagnosis (default: 1)					
C_EGY_SP_IV_CAL_MAX	1	0...FFFFH	0...255	3.8911e-3	[mJ]
maximal value for plausible injectoe programming value "Energie""					
C_EGY_SP_IV_CAL_MIN	1	0...FFFFH	0...255	3.8911e-3	[mJ]
minimal value for plausible injector programming value "Energie"					
C_MFF_ABS_IV_CAL_MAX	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
maximal value for plausible injector programming value "Kleinmenge"					
C_MFF_ABS_IV_CAL_MIN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
minimal value for plausible injector programming value "Kleinmenge"					
C_N_MIN_PLAUS_IV_VAL	1	0...1FE0H	0...8160	1	[rpm]
minimum of engine speed for injec tion valve programming plausibility diagnosis					

41.37 Customer adaptation module: AGGR INJR

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_zylab1	O/V	0...1H	0...1	1	[-]
Logical value for single cylinder cut off bank 1					
B_zylab2	O/V	0...1H	0...1	1	[-]
Logical value for single cylinder cut off bank 2					
EOI_2_HOM_CUS[NC_CYL_NR]	O	E200...5A00H	-180...540	0.0234375	[°CRK]
End of the second injection for homogeneus mode, absolute value					
EOI_3_HOMS_CUS[NC_CYL_NR]	O	E200...5A00H	-180...540	0.0234375	[°CRK]
End of the third injection for homogeneous stratified mode					
EOI_3_S_CUS[NC_CYL_NR]	O	E200...5A00H	-180...540	0.0234375	[°CRK]
End of the third injection for stratified mode					
Espr_mod[NC_CYL_NR]	O/V	0...FFH	0...255	1	[-]
Aktueller Einspritzmodus					
Eta_zyl_aus	O/V	0...7FFFH	0...99.99694	3.0518e-3	[%]
Actual efficiency fuel cut-off					
FAC_EOI_OPM_SEL	O/V	0...FFH	0...0.99609	3.9062e-3	[-]
injection phasing correction interpolation factor for operation switch manager					
INH_IV_CUS	O	0...FFH	0...255	1	[-]
shut off pattern requested by customer					
LF_SOI_S_ENA	O/V	0...FFH	0...255	1	[-]
Bitfield to enable SOI-based stratified injection; cylinder individual (bit 0 stands for cyl 0 ...); 0 = use EOI, 1 = use SOI					
LFT_L_IV_REQ_HOM_CUS	O	0...FFH	0...255	1	[-]
Dynamic injector needle lift selection request by customer for homogeneous combustion mode					
LFT_L_IV_REQ_HOMS_CUS	O	0...FFH	0...255	1	[-]
Dynamic injector needle lift selection request by customer for homogeneous-stratified combustion mode					
LFT_L_IV_REQ_S_CUS	O	0...FFH	0...255	1	[-]
Dynamic injector needle lift selection request by customer for stratified combustion mode					
LV_EOI_2_DELTA_HOM_CUS	O/V	0...1H	0...1	1	[-]
Switch between EOI_2_HOM_CUS (LV... = 0) or EOI_2_DELTA_HOM_CUS (LV = 1) for phasing of second hom. Injection pulse					
LV_HOMS_ACT	O/V	0...1H	0...1	1	[-]
Indicates if HOMS is active					
LV_IGN_INJ_SYN_DEAC	O/V	0...1H	0...1	1	[-]
Flag if there is no synchronisation needed between ignition and injection					
LV_LAM_AD_INJ_CUS_ACK[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Customer acknowlegde flag for injection mode control handshake					
LV_MPLH_ACT	O/V	0...1H	0...1	1	[-]
Indicates if homogeneous mode with split injection is active					
LV_MPLP_ACT	O/V	0...1H	0...1	1	[-]
Indicates if stratified mode with post injection is active					
LV_MPLP_ENA	O/V	0...1H	0...1	1	[-]
Enable post injection					
LV_PRS_COR_MPLH_ENA_CUS	O/V	0...1H	0...1	1	[-]
Switch to enable/disable fuel rail pressure pulsation correction for homogeneous multi injection; customer provided					
LV_PRS_COR_SNGH_ENA_CUS	O/V	0...1H	0...1	1	[-]
Switch to enable/disable fuel rail pressure pulsation correction for homogeneous single injection; customer provided					
LV_TI_LIM_ACT	O/V	0...1H	0...1	1	[-]
1: injection time limited by cycle time (720°crk)					
SOI_1_HOM_CUS[NC_CYL_NR]	O	E200...5A00H	-180...540	0.0234375	[°CRK]
Start of the first injection for homogeneous mode					
SOI_1_HOMS_CUS[NC_CYL_NR]	O	E200...5A00H	-180...540	0.0234375	[°CRK]

Start of the first injection for homogeneous stratified mode					
SOI_2_MAX_CUS	O	E200...5A00H	-180...540	0.0234375	[°CRK]
Maximum start of the second injection					
SOI_3_S_CUS[NC_CYL_NR]	O	E200...5A00H	-180...540	0.0234375	[°CRK]
Start of the post injection during stratified mode					
SOI_MPLP_CUS[NC_CYL_NR]	O	E200...5A00H	-180...540	0.0234375	[°CRK]
Start of the post injection during stratified mode					
SOI_S_CUS[NC_CYL_NR]	O	E200...5A00H	-180...540	0.0234375	[°CRK]
Start of injeciton for last stratified pulse					
St_injad_anf[NC_CBK_EX_NR]	O/V	0...FFFFH	0...65535	1	[-]
state of adaptions request for "Kleinstmengen"					
T_DLY_1_2_S_CUS[NC_CYL_NR]	O	0...FFFFH	0...262.14	0.004	[ms]
Time delay between first and second pulse for stratified mode					
T_DLY_2_3_MIN_HOMS_CUS[NC_CYL_NR]	O	0...FFFFH	0...262.14	0.004	[ms]
Minimum time delay between second and third pulse for homogeneous-stratified mode					
T_DLY_2_3_S_CUS[NC_CYL_NR]	O	0...FFFFH	0...262.14	0.004	[ms]
Time delay between second and third pulse for stratified mode					
TCYL_MDL_CUS	O/V	8000...7FFFH	-2048... 2047.9375	0.0625	[K]
Modelled cylinder air temperature - customer provided.					
Ti_1_homext	O/V	0...FFFFH	0...838.848	0.0128	[ms]
effektive Einspritzzeit Bank 1					
Ti_2_homext	O/V	0...FFFFH	0...838.848	0.0128	[ms]
effektive Einspritzzeit Bank 2					
Wese	O/V	0...FFFFH	0...1535.97656	0.0234375	[°CRK]
Angle of end of injection					
CTR_SEG_IGN_INJ_SYN_DEAC	V	0...FFH	0...255	1	[-]
Segment counter until deactivation of ignition injection synchronization in HOM mode.					

Input data:

B_nesps_ena	B_prailpulskor_ee	B_prailpulskor_me	B_temxon
B_wese2h_abs	Baw_ist	CTR_CYL_NR_ST_CLC	Dwese2_h[NC_CYL_NR]
		ECU_STATE	EFF_SCC_AV
EOI_1_HOM[NC_CYL_NR]	F_vwi_ba	INJ_MOD[NC_CYL_NR]	Injekt_hub_h
Injekt_hub_hs	Injekt_hub_s	LC_MFF_MPG_NEW	LF_MFF_HOM
LV_S_CLC	LV_SCC[NC_CBK_EX_NR]	LV_ST_END	NC_CBK_EX_NR
NC_CYL_NR	NC_LAMB_REF	NC_PLS_NR	St_injad_quit
St_wesb_s	Stat_zyl_aus	STATE_LAM_AD_INJ_ACT [NC_CBK_EX_NR][1][1]	Td_wese[NC_CYL_NR][2]
Td_wese23min_hs[NC_CYL_NR]	TI_1_HOM[NC_CYL_NR]	TI_2_HOM[NC_CYL_NR]	TI_3_HOM[NC_CYL_NR]
Tevt_mdl	Wes_soll_h[NC_CYL_NR][NC_PLS_NR]	Wesb_s[NC_CYL_NR]	Wesb1_h[NC_CYL_NR]
Wesb1_hs[NC_CYL_NR]	Wesb2max_h	Wesbkh_s[NC_CYL_NR]	Wese_s[NC_CYL_NR]
Wese2_h[NC_CYL_NR]	Wese2_hs[NC_CYL_NR]	Wese3_h[NC_CYL_NR]	Wese3_hs[NC_CYL_NR]

41.37.1 Outputs for BMW functions which are defined as INJR exported data

FUNCTION DESCRIPTION:

General information:

Adaptation to BMW environment.

Application conditions:

Initalisation at reset or at exit power latch phase (BMW-variables):

0 all, except: Eta_zyl_aus = 100 %

Recurrences:	segment:	B_zylab1/2, Espr_mod[NC_CYL_NR], Wese
	10 ms:	Eta_zyl_aus, Ti_1_homext, Ti_2_homext
	100ms:	St_injad_anf

Activation: every engine state

Deactivation: ---

Formula section:

```
B_zylab1      = LV_SCC_1
B_zylab2      = LV_SCC_2

If    ECU_STATE = "PWL"

Then

    Espr_mod[NC_CYL_NR] = 0
    Wese            = 0
    Eta_zyl_aus     = 0
    Ti_1_homext     = 0
    Ti_2_homext     = 0
    St_injad_anf [NC_CBK_EX_NR] = 0

Elseif

    Espr_mod[NC_CYL_NR] = INJ_MOD[NC_CYL_NR]
    Wese       = EOI_1_HOM
    St_injad_anf [NC_CBK_EX_NR] = STATE_LAM_AD_INJ_ACT[NC_CBK_EX_NR]

    Eta_zyl_aus     = EFF_SCC_AV * 100

    IF NC_CBK_EX_NR > 1                    (please note physical meaning)

    THEN

            Ti_1_homext = 0

            Ti_2_homext = 0

      FOR x = 0 TO (NC_CYL_NR – 1)

            IF (NC_LAMB_REF AND $2^x$) ≠ 0          (bitwise)

            THEN

                    Bank 2:
                    Ti_2_homext += TI_1_HOM[x] + TI_2_HOM[x] + TI_3_HOM[x]

            ELSE

                    Bank 1:
                    Ti_1_homext += TI_1_HOM[x] + TI_2_HOM[x] + TI_3_HOM[x]

            ENDIF

      ENDFOR
```

```
            Ti_1_homext = Ti_1_homext / (NC_CYL_NR / NC_CBK_EX_NR)
            Ti_2_homext = Ti_2_homext / (NC_CYL_NR / NC_CBK_EX_NR)
    ELSE
        Ti_1_homext = 0
        FOR x = 0 TO (NC_CYL_NR – 1)
            Ti_1_homext += TI_1_HOM[x] + TI_2_HOM[x] + TI_3_HOM[x]
        ENDFOR
        Ti_1_homext = Ti_1_homext / NC_CYL_NR
    ENDIF
Endif
```

41.37.2 Outputs for SV aggregates

FUNCTION DESCRIPTION:

General information:

Adaptation to BMW environment.

Application conditions:

Initialisation(Siemens-Variables): 0, except LV_EOI_2_DELTA_HOM_CUS = 1

// SOI_3_S_CUS is not valid any more for post injection. It is set to a default value to keep the interface to aggr. EXTC stable. Post injection has to be disabled for the function sample!
SOI_3_S_CUS[NC_CYL_NR] = 140 (each element)

Recurrence : 10 ms: LV_TI_LIM_ACT, FAC_EOI_OPM_SEL, LV_MPLP_ENA, LV_MPLH_ACT, LV_MPLP_ACT, LV_HOMS_ACT

20ms: LV_LAM_AD_INJ_CUS_ACK[NC_CBK_EX_NR]

if LV_ST_END **=** 0 or at power latch phase
then recurrence of the following vaules is 10ms
SOI_1_HOM_CUS[NC_CYL_NR],
EOI_3_S_CUS[NC_CYL_NR],
T_DLY_1_2_S_CUS[NC_CYL_NR],
T_DLY_2_3_S_CUS[NC_CYL_NR],
SOI_MPLP_CUS[NC_CYL_NR],
SOI_1_HOMS_CUS[NC_CYL_NR],
EOI_2_HOMS_CUS[NC_CYL_NR],
EOI_3_HOMS_CUS[NC_CYL_NR],
T_DLY_2_3_MIN_HOMS_CUS[NC_CYL_NR],
EOI_2_HOM_CUS[NC_CYL_NR],
EOI_3_HOM_CUS[NC_CYL_NR],
SOI_2_MAX_CUS,
LV_EOI_2_DELTA_HOM_CUS,

```
                    LF_SOI_S_ENA,
                    SOI_S_CUS[NC_CYL_NR],
                    INH_IV_CUS,
                    LFT_L_IV_REQ_S_CUS,
                    LFT_L_IV_REQ_HOM_CUS,
                    LFT_L_IV_REQ_HOMS_CUS,
                    LV_PRS_COR_SNGH_ENA_CUS,
                    LV_PRS_COR_MPLH_ENA_CUS
                    LV_IGN_INJ_SYN_DEAC

              else  recurrence of the following vaules is segment synchron
                    SOI_1_HOM_CUS[NC_CYL_NR],
                    EOI_3_S_CUS[NC_CYL_NR],
                    T_DLY_1_2_S_CUS[NC_CYL_NR],
                    T_DLY_2_3_S_CUS[NC_CYL_NR],
                        SOI_MPLP_CUS[NC_CYL_NR],
                        SOI_1_HOMS_CUS[NC_CYL_NR],
                        EOI_2_HOMS_CUS[NC_CYL_NR],
                    EOI_3_HOMS_CUS[NC_CYL_NR],
                    T_DLY_2_3_MIN_HOMS_CUS[NC_CYL_NR],
                        EOI_2_HOM_CUS[NC_CYL_NR],
                        EOI_3_HOM_CUS[NC_CYL_NR],
                    SOI_2_MAX_CUS,
                        LV_EOI_2_DELTA_HOM_CUS,
                    LF_SOI_S_ENA,
                        SOI_S_CUS[NC_CYL_NR],
                    INH_IV_CUS,
                    LFT_L_IV_REQ_S_CUS,
                    LFT_L_IV_REQ_HOM_CUS,
                    LFT_L_IV_REQ_HOMS_CUS,
                    LV_PRS_COR_SNGH_ENA_CUS,
                    LV_PRS_COR_MPLH_ENA_CUS
                    LV_IGN_INJ_SYN_DEAC

          100ms:    TCYL_MDL_CUS
```

Activation: at every engine state

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

If LV_S_CLC = 1

then

EOI_3_S_CUS[NC_CYL_NR] = Wese_s[NC_CYL_NR]
T_DLY_ 1_2_S_CUS[NC_CYL_NR] = Td_wese[NC_CYL_NR][1]
T_DLY_2_3_S_CUS[NC_CYL_NR] = Td_wese[NC_CYL_NR][2]
SOI_MPLP_CUS[NC_CYL_NR] = Wesbkh_s[NC_CYL_NR]
SOI_1_HOMS_CUS[NC_CYL_NR] = Wesb1_hs[NC_CYL_NR]
EOI_2_HOMS_CUS[NC_CYL_NR] = Wese2_hs[NC_CYL_NR]
EOI_3_HOMS_CUS[NC_CYL_NR] = Wese3_hs[NC_CYL_NR]
T_DLY_2_3_MIN_HOMS_CUS[NC_CYL_NR] = Td_wese23min_hs[NC_CYL_NR]

```
        LF_SOI_S_ENA                                    = St_wesb_s
        SOI_S_CUS[NC_CYL_NR]                            = Wesb_s[NC_CYL_NR]
endif

INH_IV_CUS                              = Stat_zyl_aus
FAC_EOI_OPM_SEL                         = F_vwi_ba
LFT_L_IV_REQ_S_CUS                      = Injekt_hub_s
LFT_L_IV_REQ_HOM_CUS            = Injekt_hub_h
LFT_L_IV_REQ_HOMS_CUS           = Injekt_hub_hs

If      Bit 0 of St_injad_quit is 1
Then    LV_LAM_AD_INJ_CUS_ACK[1] = 1
Else    LV_LAM_AD_INJ_CUS_ACK[1] = 0
Endif
If      Bit 1 of St_injad_quit is 1
Then    LV_LAM_AD_INJ_CUS_ACK[2] = 1
Else    LV_LAM_AD_INJ_CUS_ACK[2] = 0
endif

LV_MPLP_ENA             = B_nesps_ena
If    LC_MFF_MPG_NEW = 1
Then
    IF (Bit 3 of LF_MFF_HOM is set) or (Bit 4 of LF_MFF_HOM is set)
      //Note: zero based indexing is used for LF_MFF_HOM
    THEN
        LV_IGN_INJ_SYN_DEAC = 0
        CTR_SEG_IGN_INJ_SYN_DEAC = NC_CYL_NR
    ELSE
        IF CTR_SEG_IGN_INJ_SYN_DEAC > 0
        THEN
                Decrement CTR_SEG_IGN_INJ_SYN_DEAC by 1
        ENDIF
        IF CTR_SEG_IGN_INJ_SYN_DEAC = 0
        THEN
                LV_IGN_INJ_SYN_DEAC = 1
        ENDIF
    ENDIF
    For x = 0 to NC_CYL_NR-1 do
        SOI_1_HOM_CUS[x] = Wes_soll_h[x][0]
        If Bit 1 of LF_MFF_HOM = 1
                        //Note: zero based indexing is used for LF_MFF_HOM
        Then
```

```
            EOI_2_HOM_CUS[x] = Wes_soll_h[x][1]
        Else
            EOI_2_HOM_CUS[x] = Wes_soll_h[x][2]
        Endif

        EOI_3_HOM_CUS[x] = Wes_soll_h[x][2]

    Endfor

Else - LC_MFF_MPG_NEW = 0

    IF B_wese2h_abs = 0
    THEN

        LV_IGN_INJ_SYN_DEAC = 0
        CTR_SEG_IGN_INJ_SYN_DEAC = NC_CYL_NR

    ELSE

        IF CTR_SEG_IGN_INJ_SYN_DEAC > 0
        THEN

            Decrement CTR_SEG_IGN_INJ_SYN_DEAC by 1

        ENDIF

        IF CTR_SEG_IGN_INJ_SYN_DEAC = 0
        THEN

            LV_IGN_INJ_SYN_DEAC = 1

        ENDIF

    ENDIF

    For x = 0 to NC_CYL_NR-1 do

        SOI_1_HOM_CUS[x] = Wesb1_h[x]

        If B_wese2h_abs = 0
        Then

            EOI_2_HOM_CUS[x] = Dwese2_h[x]

        Else

            EOI_2_HOM_CUS[x] = Wese2_h[x]

        Endif

        EOI_3_HOM_CUS[x] = Wese3_h[x]

    Endfor

Endif

If    ECU_STATE = "PWL"

Then

    LV_TI_LIM_ACT                    = 0
    LV_EOI_2_DELTA_HOM_CUS           = 1
    EOI_3_HOM_CUS[NC_CYL_NR]         = 0
    SOI_2_MAX_CUS                    = 0
```

```
Elseif

    LV_TI_LIM_ACT                        =  B_temxon
    If   LC_MFF_MPG_NEW = 1
    Then

       LV_EOI_2_DELTA_HOM_CUS =
                              (Bit 3 of LF_MFF_HOM) OR (Bit 4 of LF_MFF_HOM)
                              //Note: zero based indexing is used for LF_MFF_HOM
    Else

       LV_EOI_2_DELTA_HOM_CUS = NOT(B_wese2h_abs)
    Endif

    SOI_2_MAX_CUS                        = Wesb2max_h
Endif

If    ( Baw_ist and 0xff ) = 66 or ( Baw_ist and 0xff ) = 226 or ( Baw_ist and 0xff ) = 82
Then
    LV_MPLH_ACT = 1
Elseif
    LV_MPLH_ACT = 0
Endif

If    ( Baw_ist and 0xff ) = 65
Then
    LV_MPLP_ACT = 1
Elseif
    LV_MPLP_ACT = 0
Endif

If    ( Baw_ist and 0xff ) = 19
Then
    LV_HOMS_ACT = 1
Elseif
    LV_HOMS_ACT = 0
Endif

LV_PRS_COR_MPLH_ENA_CUS = B_prailpulskor_me
LV_PRS_COR_SNGH_ENA_CUS = B_prailpulskor_ee
```

```
/* with 100ms recurrence */
TCYL_MDL_CUS = Tevt_mdl + 273.15  /* Attention: transfer of °C to K */
```

42 Intake System

Chapter Intake System	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 6683 of 9643

Table of Contents

Index

42.1 Acquisition of mass air flow

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_SUM	V/O	0...FFFFFFFFH	0...67108863.98437	0.015625	[kg/h]
Accumulated mass air flow measurements for segment number (2n), n = 1,2,3,...					
MAF_SUM_1	V/O	0...FFFFFFFFH	0...67108863.98437	0.015625	[kg/h]
Accumulated mass air flow measurements for segment number (2n-1), n = 1,2,3,...					
MAF_CTR	V/O	0...FFFFH	0...65535	1	[-]
accumulated mass air flow measurements for segment number (2n), n = 1,2,3,...					
MAF_CTR_1	V/O	0...FFFFH	0...65535	1	[-]
accumulated mass air flow measurements for segment number (2n-1), n = 1,2,3,...					
LV_MAF_SEG	V/O	0...1H	0...1	1	[-]
indicates if segment number is even or odd in order to choose MAF_SUM / MAF_C...					
V_MAF	V/O	0...FFH	0...4.98046	0.0195313	[V]
Mass air flow sensor raw acquisition.					

Input data:

LV_VAR_MAF	LV_VAR_TCHA		

Import action:

ACTION_INFR_GetVmafSensor(OUT <V_maf**>)**
This action returns the digitized voltage values from the analog MAF sensor

Note: The imported action is defined in 'INSY - Requirements to infrastructure interface'

FUNCTION DESCRIPTION:

General information:

After having read out one buffer - MAF_SUM or MAF_SUM_1 - the application software has to clear this buffer as well as the corresponding counter - MAF_CTR or MAF_CTR_1.

Application conditions:

Initialisation:

After HW reset, the buffers - MAF_SUM and MAF_SUM_1 - and the corresponding counters - MAF_CTR and MAF_CTR_1 - have to be cleared and the pointer ID_MAF_PTR has to be initialized with the address of ID_MAF_TAB.

Recurrence: 1ms

Activation: LV_VAR_MAF = 1 **and** LV_VAR_TCHA = 1

Deactivation: LV_VAR_MAF = 0 **or** LV_VAR_TCHA = 0

Description:

The raw value for MAF is measured by continous conversion every 1 ms. The raw values (8 bit) are converted by a table ID_MAF_TAB (size: 256 bytes) into values with the unit [kg/h] and summed up in one of the two buffers, MAF_SUM and MAF_SUM_1. The number of values is counted in MAF_CTR or MAF_CTR_1.

A two-buffer-system is used to avoid incorrect MAF calculation. The buffers alternate with each change of segment, i. e. LV_MAF_SEG is toggled each segment from the BSW always on fixed tooth event before setting crankshaft trigger.

The conversion of the mass air flow voltage V_MAF into the physical meanig of a mass air flow with the unit kg/h is done by using the map ID_MAF_TAB. Basicly the table represents a one dimensional map with the dependency V_MAF. V_MAF is splitted into 256 steps with the resolution of 0.0195. That mean, if V_MAP=0.468V (=24 steps) the corresponding mass air flow will be find in ID_MAF_TAB at the break point number 24. The indexing of this map has to be understood in the following way:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	32	33	...	...												
													...	...	...	

...

...

Formula section:

ACTION_INFR_GetVmafSensor(OUT <V_maf**>)**

$$\mathrm{MAF\_SUM} = \sum_{i=1}^{\mathrm{MAF\_CTR}} maf_i,\ maf_i\text{: MAF acquis., converted by the table ID\_MAF\_TAB in kg/h}$$

and

$$\mathrm{MAF\_SUM\_1} = \sum_{i=1}^{\mathrm{MAF\_CTR\_1}} maf_i,\ maf_i\text{: MAF acquis., converted by the table ID\_MAF\_TAB in kg/h.}$$

Calibration data:

Name	**Dim**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
ID_MAF_TAB	16*16	0...FFFFH	0...1023.98437	0.015625	[kg/h]
LDP_ID_MAF_TAB_1	16	0...FFH	0...255	1	[-]
LDP_ID_MAF_TAB_2	16	0...FFH	0...255	1	[-]
Conversion table for MAF value					

42.2 Mass air flow acquisiton for freq. MAF sensor

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_FRQ_KGH_BAS[NC_MAF_NR]	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
basic value of the air mass flow (freq. MAF)					
T_PER_MAF_FRQ[NC_MAF_NR]	V/O	0...FFFFH	0...65535	1	[µs]
arithmetric average of the period time					
LV_CTR_T_PER_MAF_FRQ_RNG[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
failure (range) of the mean period time for frequency MAF sensor					
LV_CTR_T_PER_MAF_FRQ_GRD[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
failure (gradient) of the mean period time for frequency MAF sensor					
LV_CTR_T_PER_MAF_FRQ_EL[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
failure (electrical) of the mean period time for frequency MAF sensor					
LV_CTR_T_PER_MAF_FRQ_SENS[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
failure (sensor) of the mean period time for frequency MAF sensor					
T_PER_MAF_FRQ_BAS[NC_MAF_NR]	V/O	0...FFFFH	0...65535	1	[µs]
basic arithmetric average of the period time					
MAF_FRQ_KGH_MES[NC_MAF_NR]	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
measured mass air flow					
LV_MAF_FRQ_SEG[NC_MAF_NR]	V	0...1H	0...1	1	[-]
indicates if segment number is even or odd to choose the right buffer					
MAF_FRQ_KGH_SUM_0[NC_MAF_NR]	V	0...FFFFFFFFH	0... 134217727.96875	0.03125	[kg/h]
accumulated mass air flow measured every sampling pulse time					
MAF_FRQ_KGH_SUM_1[NC_MAF_NR]	V	0...FFFFFFFFH	0... 134217727.96875	0.03125	[kg/h]
accumulated mass air flow measured every sampling pulse time					
CTR_MAF_FRQ_KGH_SUM_0[NC_MAF_NR]	V	0...FFFFH	0...65535	1	[-]
number of accumulated mass air flows during one segment					
CTR_MAF_FRQ_KGH_SUM_1[NC_MAF_NR]	V	0...FFFFH	0...65535	1	[-]
number of accumulated mass air flows during one segment					
MAF_FRQ_KGH_SUM[NC_MAF_NR]	V	0...FFFFFFFFH	0... 134217727.96875	0.03125	[kg/h]
accumulated mass air flow					
CTR_MAF_FRQ_KGH_SUM[NC_MAF_NR]	V/O	0...FFFFH	0...65535	1	[-]
number of valid accumulated mass air flows during one segment					
CTR_T_PER_MAF_FRQ_RNG_0[NC_MAF_NR]	V	0...FFFFH	0...65535	1	[-]
number of accumulated range failures during one segment for buffer 0					
CTR_T_PER_MAF_FRQ_RNG_1[NC_MAF_NR]	V	0...FFFFH	0...65535	1	[-]
number of accumulated range failures during one segment for buffer 1					
CTR_T_PER_MAF_FRQ_RNG[NC_MAF_NR]	V/O	0...FFFFH	0...65535	1	[-]
number of accumulated range failures during one segment					
CTR_T_PER_MAF_FRQ_GRD_0[NC_MAF_NR]	V	0...FFFFH	0...65535	1	[-]
number of accumulated gradient failures during one segment for buffer 0					
CTR_T_PER_MAF_FRQ_GRD_1[NC_MAF_NR]	V	0...FFFFH	0...65535	1	[-]

number of accumulated gradient failures during one segment for buffer 1					
CTR_T_PER_MAF_FRQ_GRD[NC_MAF_NR]	V/O	0...FFFFH	0...65535	1	[-]
number of accumulated gradient failures during one segment					
CTR_PER_MAF_FRQ_EL[NC_MAF_NR]	V	0...FFH	0...255	1	[-]
number of times without a flank					

Input data:

LV_ES	LV_VAR_MAF_LEARNT	LV_VAR_MAF	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_MAF_FRQ_TAB[NC_MAF_NR]	256	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDP_T_PER_MAF_FRQ_OFS	256	0...FFFFH	0...65535	1	[µs]
conversion table for the frequency MAF sensor					
C_T_PER_MAX_FRQ_OFS	1	0...FFFFH	0...65535	1	[µs]
offset of the period time for conversion table					
C_T_PER_MAF_FRQ_MAX_DIAG	1	0...FFFFH	0...65535	1	[µs]
maximum threshold for range check of T_PER_MAF_FRQ_SUM_j					
C_T_PER_MAF_FRQ_MIN_DIAG	1	0...FFFFH	0...65535	1	[µs]
minimum threshold for range check of T_PER_MAF_FRQ_SUM_j					
C_T_PER_MAF_FRQ_GRD_DIAG	1	0...FFFFH	0...65535	1	[µs]
maximum threshold for gradient check of T_PER_MAF_FRQ_SUM					
C_T_PER_MAF_FRQ_SENS	1	0...FFFFH	0...65535	1	[µs]
minimum threshold for range check of T_PER_MAF_FRQ_SENS					
C_CTR_PER_MAF_FRQ_EL	1	0...FFH	0...255	1	[-]
Maximum threshold for electrical diagnosis					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_MAF_NR	1	0...FFH	0...255	1	[-]
number of MAF- sensors					

Import actions:

ACTION_INFR_GetResultTimeStamp (OUT <CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR]>, OUT <T_PER_MAF_FRQ_SUM[NC_MAF_NR]>)

FUNCTION DESCRIPTION:

General information:

Calculation of basic air mass flow for a frequency MAF sensor

The determination of basic air mass flow from the input signal to the ECU in general is done in 3 steps.

First step:

Using these informations the mean value of time periods between falling edges during the last sampling time is calculated:

T_PER_MAF_FRQ[NC_MAF_NR] = T_PER_MAF_FRQ_SUM[NC_MAF_NR] / CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR]

Second step:

The mean value of time periods (T_PER_MAF_FRQ[NC_MAF_NR]) is converted into air mass flows (MAF_FRQ_KGH_BAS[NC_MAF_NR]) with help of a map (ID_MAF_FRQ_TAB[NC_MAF_NR]).

Third step:

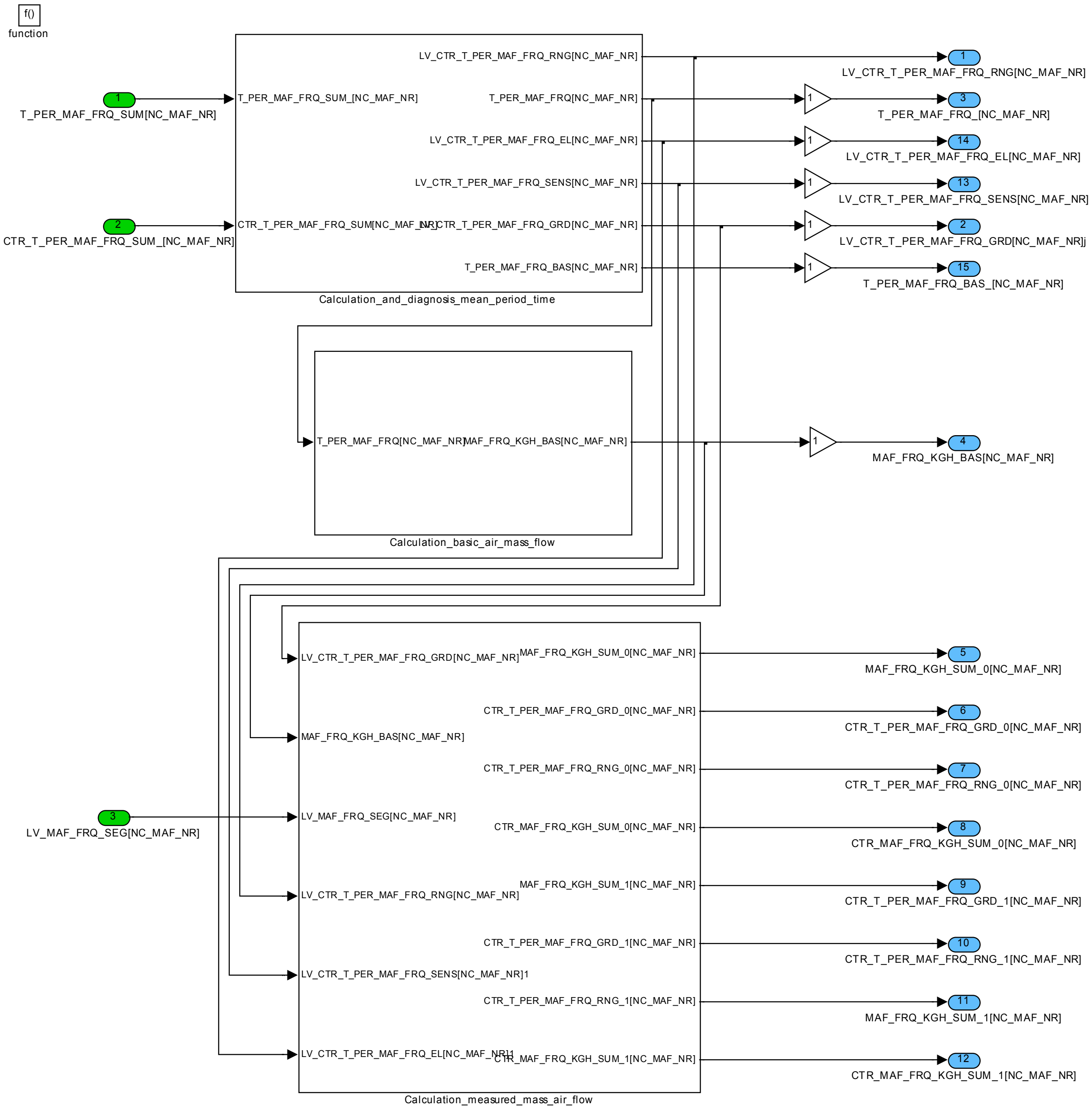
During a segment time the air mass flows of each millisecond (MAF_FRQ_KGH_BAS[NC_MAF_NR]) are summed up (MAF_FRQ_KGH_SUM[NC_MAF_NR]) in order to calculate the mean value of air mass flow:

MAF_FRQ_KGH_MES[NC_MAF_NR] = MAF_FRQ_KGH_SUM[NC_MAF_NR] / CTR_MAF_FRQ_KGH_SUM[NC_MAF_NR]

Description:

This is a global overview about all input and output signals of this function. Details are shown in the following chapters.

42.2.1 Overview



The function to do all calculations is splitted into three parts:

- calculation of mean period time
- calculation of basic mass air flow
- calculation of mean mass air flow during one segment

Besides the number of MAF sensors is fixed.

42.2.1.1 Calculation and Diagnosis of the mean period time

Application conditions:

Initialisation: -

Recurrence: 1 ms

Activation: **If** (LV_VAR_MAF_LEARNT = 0 **and** LV_VAR_MAF = 0) **or**
(LV_VAR_MAF_LEARNT = 0 **and** LV_VAR_MAF = 1) **or**
(LV_VAR_MAF_LEARNT = 1 **and** LV_VAR_MAF = 1)

Deactivation: **If** (LV_VAR_MAF_LEARNT = 1 **and** LV_VAR_MAF = 0)

FUNCTION DESCRIPTION:

The logical value LV_CTR_T_PER_MAF_FRQ_RNG[NC_MAF_NR] will be set, if the mean period time is not within a maximum and minimum threshold.

Additionally the logical value LV_CTR_T_PER_MAF_FRQ_GRD[NC_MAF_NR] will be set, if the gradient of the mean period time is higher than C_T_PER_MAF_FRQ_GRD_DIAG.

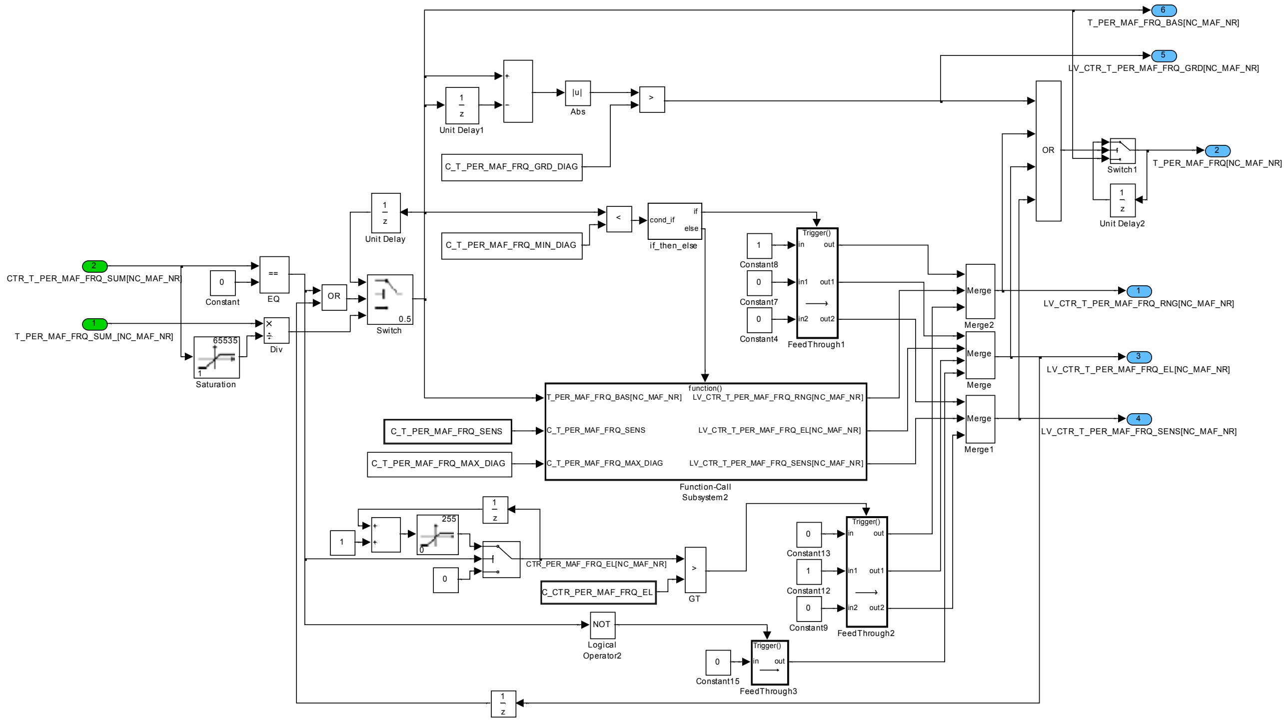
If one of the error bits LV_CTR_T_PER_MAF_FRQ_EL/SENS/RNG/GRD[NC_MAF_NR] is set, the old period time is used.

If one of the error bits LV_CTR_T_PER_MAF_FRQ_EL/SENS/RNG is set, then the other error bits has to be set to zero.

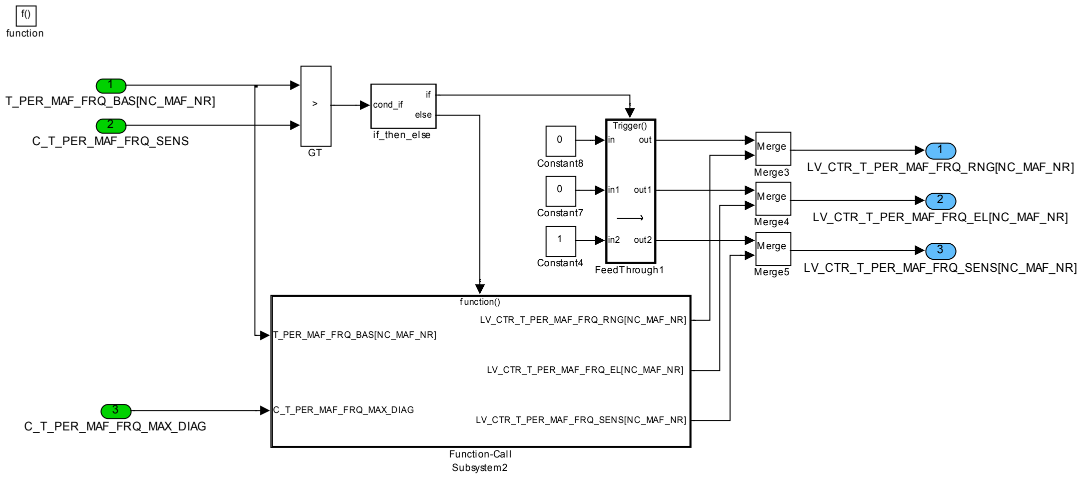
For setting of LV_CTR_T_PER_MAF_FRQ_EL[NC_MAF_NR] a counter is started everytime CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR] = 0, if the counter is over a calibratable threshold LV_CTR_T_PER_MAF_FRQ_EL[NC_MAF_NR] is set.

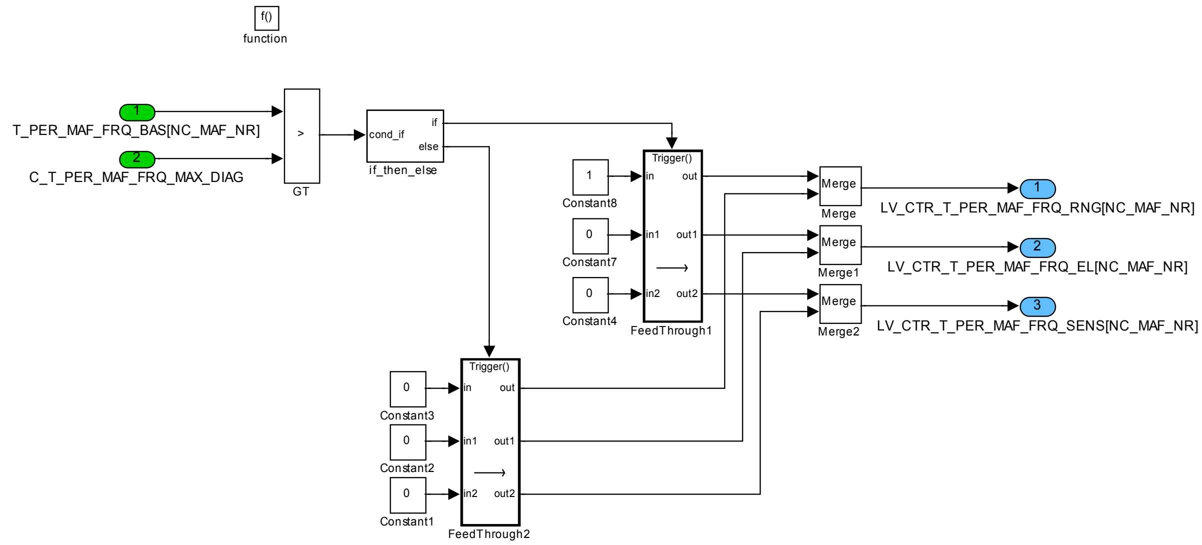
If LV_CTR_T_PER_MAF_FRQ_EL[NC_MAF_NR] = 1 the first value T_PER_MAF_FRQ_SUM[NC_MAF_NR] != 0 delievered by infrastructure is not used.

Signal flow diagram:



In the following 2 signal flow diagrams, the function call function call subsystem2 is shown:





42.2.1.2 Basic air mass flow calculation for frequency MAF sensor

Application conditions:

Initialisation: -

Recurrence: 1 ms

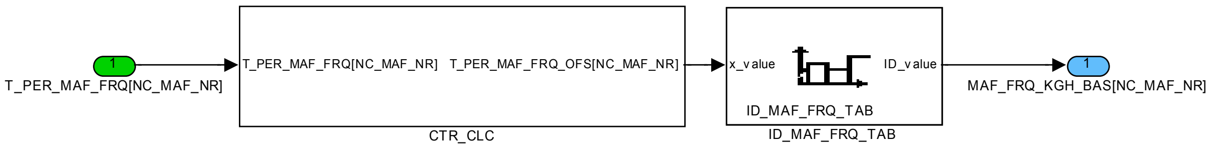
Activation: **If** (LV_VAR_MAF_LEARNT = 0 **and** LV_VAR_MAF = 0) **or**
(LV_VAR_MAF_LEARNT = 0 **and** LV_VAR_MAF = 1) **or**
(LV_VAR_MAF_LEARNT = 1 **and** LV_VAR_MAF = 1)

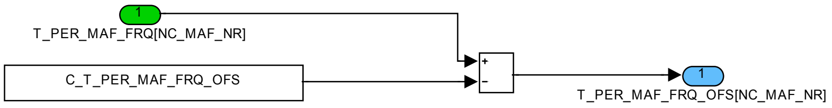
Deactivation: **If** (LV_VAR_MAF_LEARNT = 1 **and** LV_VAR_MAF = 0)

FUNCTION DESCRIPTION:

The limited mean value of the period time T_PER_MAF_FRQ[NC_MAF_NR] are converted by a table ID_MAF_FRQ_TAB[NC_MAF_NR] into values with the unit [kg/h].

Signal flow diagram:





42.2.1.3 Calculation of measured air mass flow buffer

Application conditions:

Initialisation: -

Recurrence: 1 ms

Activation: **If** (LV_VAR_MAF_LEARNT = 0 **and** LV_VAR_MAF = 0) **or**
(LV_VAR_MAF_LEARNT = 0 **and** LV_VAR_MAF = 1) **or**
(LV_VAR_MAF_LEARNT = 1 **and** LV_VAR_MAF = 1)

Deactivation: **If** (LV_VAR_MAF_LEARNT = 1 **and** LV_VAR_MAF = 0)

FUNCTION DESCRIPTION:

The calculation of measured air mass flow consists of two different parts.

Accumulation of air mass flow during one segment:

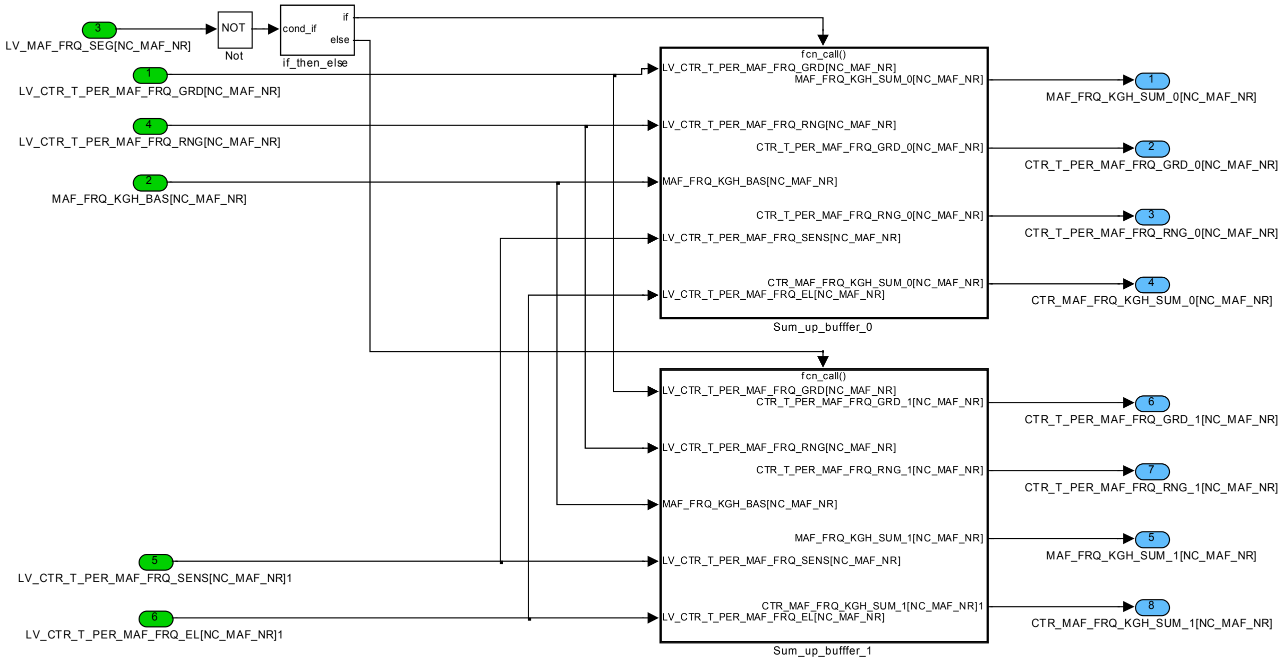
The air mass flows MAF_FRQ_KGH_BAS[NC_MAF_NR] are summed up in alternating buffers, if there is no failure. The number of summed up values is counted in

CTR_MAF_FRQ_KGH_SUM_0[NC_MAF_NR] or CTR_MAF_FRQ_KGH_SUM_1[NC_MAF_NR].

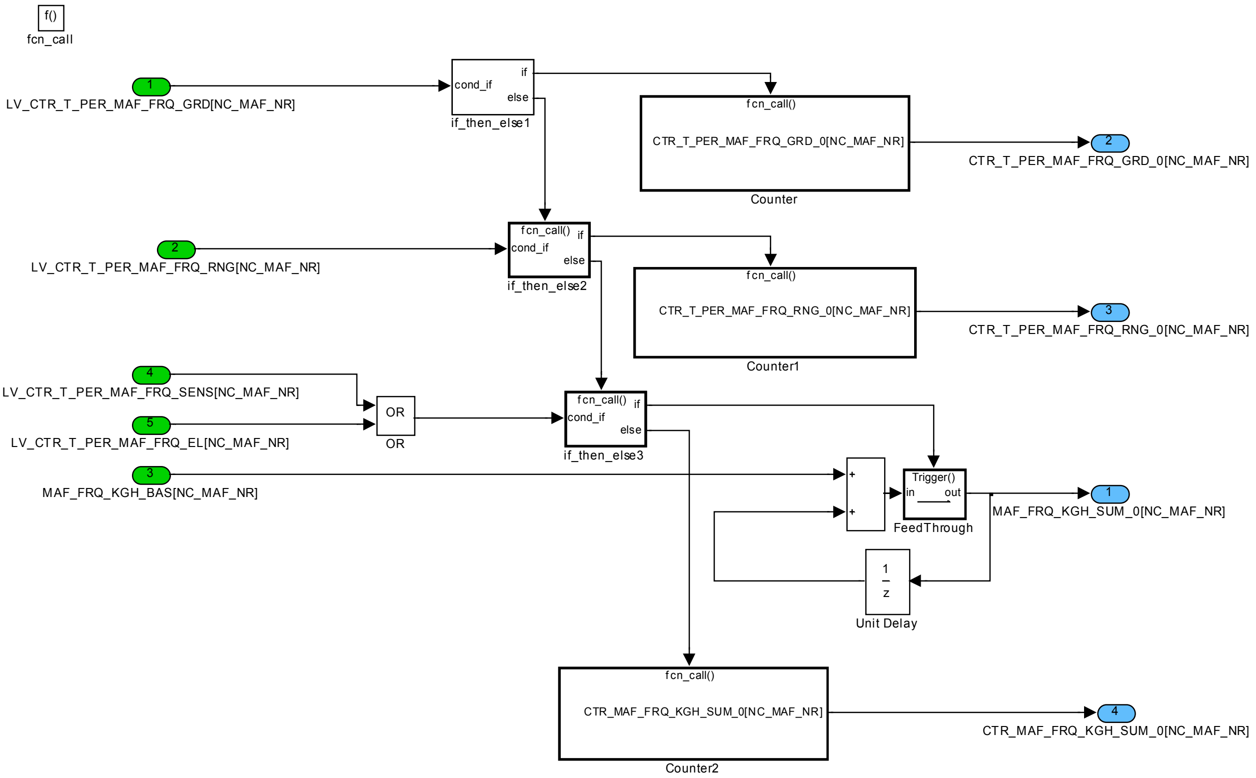
Depending on the logical value LV_MAF_FRQ_SEG one of the two buffers and counters is selected for writing whereas the content of the other buffer and counter is used for calculation.

After having read out one buffer, it needs to be cleared as well as the corresponding counter. Such a two-buffer-system is used to avoid incorrect mass air flow calculation. The buffer alternate at each change of segment.

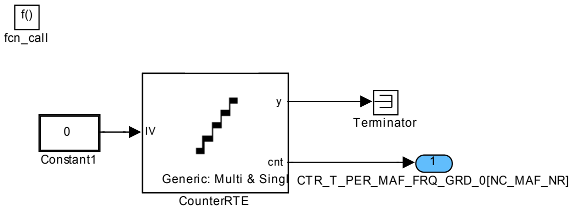
Signal flow diagram:

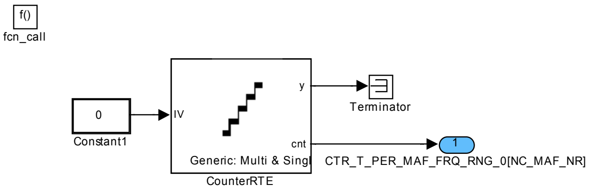


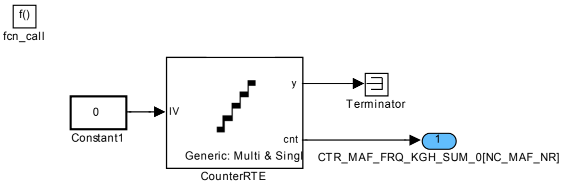
Sum up buffer:



Description of the defined counter:







The calculation above is defined only for the buffer 0, the calculation of the necessary variables for buffer 1 has to be done analogous.

42.2.1.4 Calculation of measured mean value of air mass flow

Application conditions:

Initialisation: at LV_ES 0->1, *set MAF_FRQ_KGH_MES[NC_MAF_NR] = 0*

Recurrence: every segment

Activation: **If** LV_ES = 0 **and**
(LV_VAR_MAF_LEARNT = 0 **and** LV_VAR_MAF = 0) **or**
(LV_VAR_MAF_LEARNT = 0 **and** LV_VAR_MAF = 1) **or**
(LV_VAR_MAF_LEARNT = 1 **and** LV_VAR_MAF = 1)

Deactivation: **If** (LV_VAR_MAF_LEARNT = 1 **and** LV_VAR_MAF = 0)

FUNCTION DESCRIPTION:

Selection of the sum up buffer:

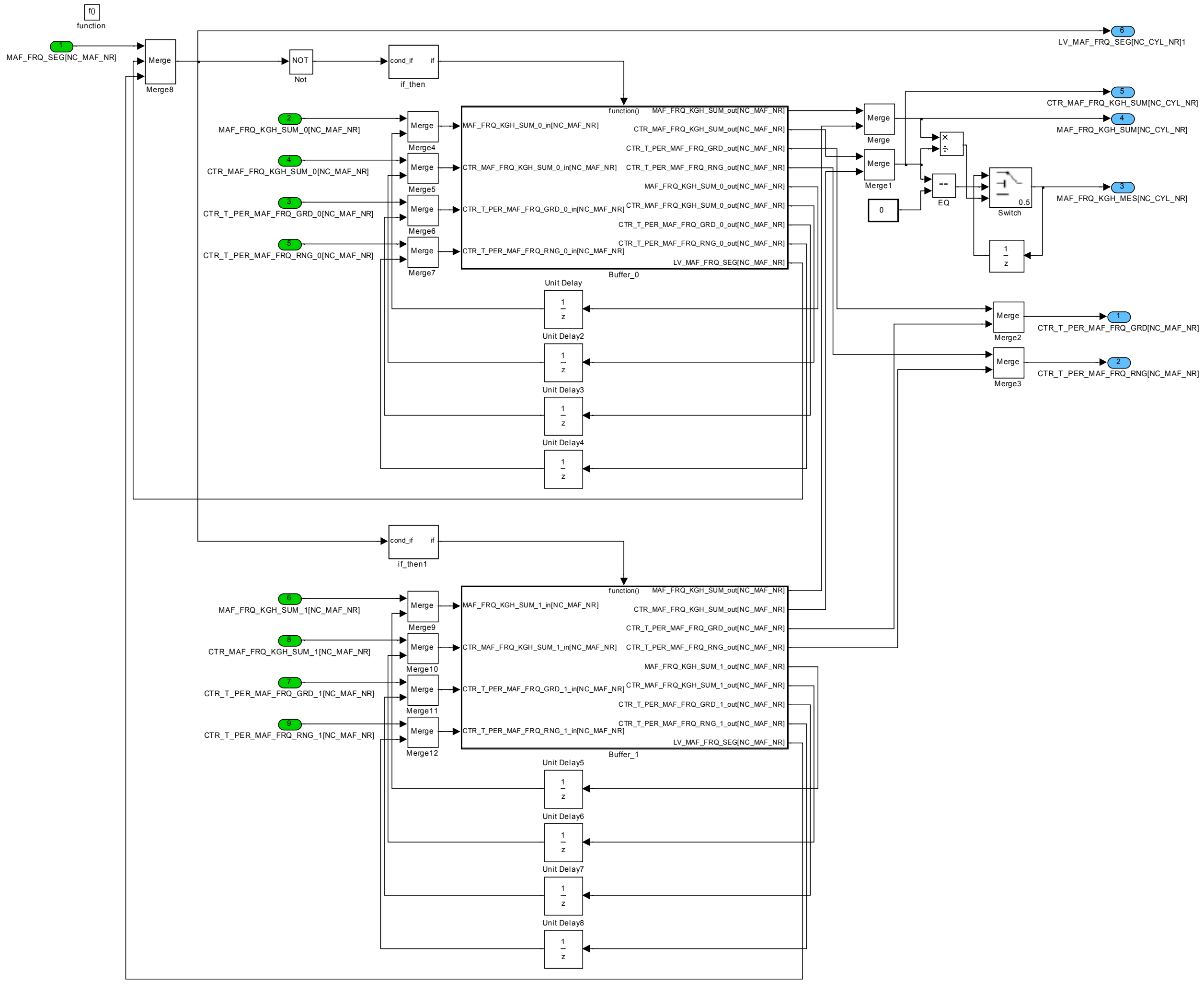
LV_MAF_FRQ_SEG[NC_MAF_NR] = 1 - LV_MAF_FRQ_SEG[NC_MAF_NR]

switching LV_MAF_FRQ_SEG[NC_MAF_NR] once per segment

mean value of the mass air flow during a segment:

MAF_FRQ_KGH_MES[NC_MAF_NR] = MAF_FRQ_KGH_SUM[NC_MAF_NR] / CTR_ MAF_FRQ_KGH_SUM[NC_MAF_NR]

Signal flow diagram:



The calculation above is defined only for the buffer 0, the calculation of the necessary variables for buffer 1 has to be done analogous.

42.3 Acquisition of manifold air pressure (MAP_MES)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAP_SUM	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Acquisition sum, buffer					
MAP_SUM_1	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Acquisition sum, buffer1					
MAP_CTR	V/O	0...FFFFH	0...65535	1	[-]
Number of MAP samples in buffer					
MAP_CTR_1	V/O	0...FFFFH	0...65535	1	[-]
Number of MAP samples in buffer1					
PUT_SUM	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Acquisition sum pressure up throttle, buffer					
PUT_SUM_1	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
Acquisition sum pressure up throttle, buffer1					
PUT_CTR	V/O	0...FFFFH	0...65535	1	[-]
Number of PUT samples in buffer					
PUT_CTR_1	V/O	0...FFFFH	0...65535	1	[-]
Number of PUT samples in buffer1					

Input data:

MAP_BAS	PUT_BAS	LV_VAR_TCHA	LV_MAP_SEG

FUNCTION DESCRIPTION:

The raw value (voltage) for MAP_BAS (PUT_BAS) is measured every 1 ms. The values (10 bit) are summed up in one of the two buffers, MAP_SUM or MAP_SUM_1 (PUT_SUM or PUT_SUM_1). The number of values is counted in MAP_CTR or MAP_CTR_1 (PUT_CTR or PUT_CTR_1).

A two-buffer-system is used to avoid incorrect MAP_MES_BAS calculation.

Calculation Description:

If (LV_MAP_SEG == 0) then

$$\text{MAP\_SUM} = \sum_{i=1}^{\text{MAP\_CTR}} \text{MAP\_BAS}_i$$

and in case the alternate buffer is active for writing (LV_MAP_SEG == 1)

$$\text{MAP\_SUM\_1} = \sum_{i=1}^{\text{MAP\_CTR\_1}} \text{MAP\_BAS}_i$$

MAP_BAS_i: MAP acquisition, raw values from the ad_buffer

additional variables for LV_VAR_TCHA = 1 (turbo)

If (LV_MAP_SEG == 0) then

$$\text{PUT\_SUM} = \sum_{i=1}^{\text{PUT\_CTR}} \text{PUT\_BAS}_i$$

and in case the alternate buffer is active for writing (LV_MAP_SEG == 1)

$$\text{PUT\_SUM\_1} = \sum_{i=1}^{\text{PUT\_CTR\_1}} \text{PUT\_BAS}_i$$

PUT_BAS_i: PUT acquisition, raw values from the ad_buffer

42.4 INSY - Requirements Infrastructure Interface

Export actions:

ACTION_INFR_GetResultTimeStampMaf(OUT <CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR]>, OUT <T_PER_MAF_FRQ_SUM[NC_MAF_NR]>)
This action returns the number of falling edges during one sampling time (1 ms) and the time over all detected signal periods of mass air flow data acquisition
ACTION_INFR_GetVmafSensor(OUT <v_maf**>)**
This action returns the digitized voltage values from the analog MAF sensor

Description for actions:

ACTION_INFR_GetResultTimeStampMaf(OUT <CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR]>, OUT <T_PER_MAF_FRQ_SUM[NC_MAF_NR]>)					
This action returns the number of falling edges during one sampling time (1 ms) and the time over all detected signal periods of mass air flow data acquisition. The counting is done autonoumous by the Infrastructure. Default SYNCHRONIZATION option is ASYNCHRONOUS					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR]	OUT	0 ... FFFFH	0 ... 65535	1	-
This parameter returns number of falling edges during one sampling time (1 ms)					
T_PER_MAF_FRQ_SUM[NC_MAF_NR]	OUT	0 ... FFFFH	0 ... 65535	1	us
This parameter returns the time over all detected signal periods					

ACTION_INFR_GetVmafSensor(OUT <v_maf**>)**					
This action returns the digitized voltage values from the analog MAF sensor					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
v_maf	OUT	0 ... FFH	0 ... 4.98046	0.0195313	V
This parameter returns voltage values from the analog MAF sensor					

FUNCTION DESCRIPTION:

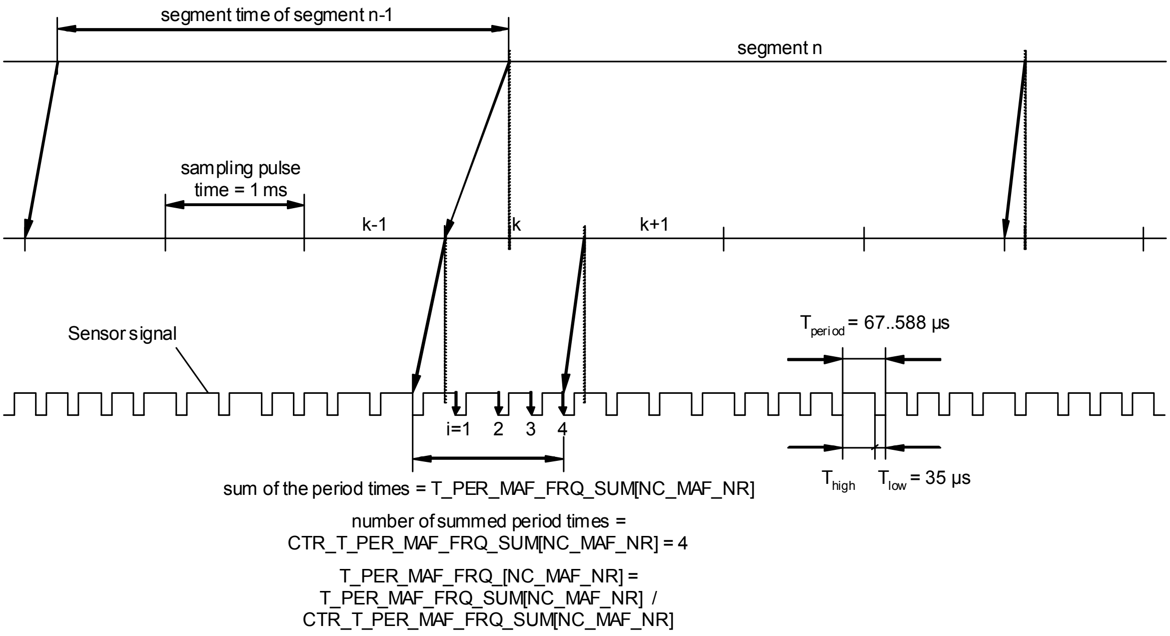
General information:

Those Actions reads out the dedicated information about the measured pulse input of the air mass flow sensor.

ACTION_INFR_GetResultTimeStampMaf

A BIOS driver called detects falling edges of the input signal. During one sampling time (1ms)

- the number of falling edges is counted (variable CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR] for further calculations in this specification)
- and the time over all detected signal periods is measured (variable T_PER_MAF_FRQ_SUM[NC_MAF_NR])



ACTION_INFR_GetVmafSensor

This action is used for the acquisition of the digitized voltage values from the analog MAF sensor connected on ECU pin 1-25 (N54 only). The content of the standard AD converter queue is transfered to the application software.

- The AD conversion is performed autonomously by the infrastructure, the returned values have not be older than 1 ms.
- The voltage values are gathered in the infrastructure until the application SW reads out the information by calling the action, old values are replaced by new ones.

Requirements for ACTION_INFR_GetResultTimeStampMaf:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR]			1	Not relevant	the number of falling edges is counted
T_PER_MAF_FRQ_SUM[NC_MAF_NR]			1us	Not relevant	time over all detected signal periods.

Diagnosis: A diagnosis is performed based on the returned signal elsewhere

Coincidence requirements: When calling the Action, the returned voltage is not older than 1ms.

further requirement due to timeout behaviour:

In case of that no falling edges are detected the parameter are delieverd as follows:
CTR_T_PER_MAF_FRQ_SUM[NC_MAF_NR] = 0

T_PER_MAF_FRQ_SUM[NC_MAF_NR] = 0

Hint for use in Application SW:
If this state is active during a sufficient long time this can be used to diagose a timeout. Due to overrun problems which can occur after a long time without any flank the first new delivered signals after a timeout should not be considered for further evaluations in the application SW.

Requirements for ACTION_INFR_GetVmafSensor:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
v_maf	-	-	0.0195313 V	not relevant	-

Diagnosis: No electrical diagnosis is done here.

Coincidence requirements:

42.5 AGGR INSY adaptation

42.5.1 Outputs for SV aggregates

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
AMP	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Ambient pressure (measured or adapted)					
AMP_AD	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Ambient pressure (adapted)					
AMP_MDL	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Ambient pressure (modelled)					
MAF_INT_PUC	V/O	0...FFFFH	0...2912.66666	0.0444444	[g]
air mass flow integral during pull cut off phase					
MAF_INT_PUE	V/O	0...FFFFH	0...2912.66666	0.0444444	[g]
air mass flow integral out of pull cut off phase					
LV_ERR_RATIO_CHK	O	0...1H	0...1	1	[-]
Boolean for actual value MAF or TPS ratio check is present (yet availiable only for interface)					
LV_ERR_MAP	V/O	0...1H	0...1	1	[-]
MAP sensor error					

Input data:

AMP_MES	MAF_INT_PUC_ACT	MAF_INT_PUC_NOT_ACT	LV_ERR_MAP_DIP_SENS
LV_ERR_MAP_DIP_PLAUS			

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: AMP = AMP_AD = AMP_MDL = AMP_MES
MAF_INT_PUC_ACT, MAF_INT_PUC_NOT_ACT 0 g at reset
Recurrence: AMP, AMP_AD, AMP_MDL 100ms
MAF_INT_PUC ; MAF_INT_PUE 20 ms, as input data (sequencing: input before output)
LV_ERR_RATIO_CHK, LV_ERR_MAP: 10ms

Activation: at every engine state

Deactivation: -

Formula section:

AMP = AMP_MES

AMP_AD = AMP_MES

AMP_MDL = AMP_MES

MAF_INT_PUC = MAF_INT_PUC_ACT

MAF_INT_PUE = MAF_INT_PUC_NOT_ACT

LV_ERR_RATIO_CHK = 0

```
IF      LV_ERR_MAP_DIP_SENS = 1     OR    LV_ERR_MAP_DIP_PLAUS = 1
THEN    LV_ERR_MAP = 1
ELSE    LV_ERR_MAP = 0
ENDIF
```

42.6 Common Air-Mass Flow Variables

Input data:

LV_MAF_CTL	LV_MAP_CTL	LV_MAF_CONF	LV_MAP_CONF
MAF_MDL_MV	MAF_KGH_MES	MAF_KGH_FG_PRED	MAF
LV_CLC_2SEG	LV_CLC_2SEG_ENA		

Application conditions:

Initialisation: all outputs phys. 0 at reset

Recurrence: segment synchronous

Activation: LV_CLC_2SEG = 0

Deactivation: LV_CLC_2SEG = 1

42.6.1 Air-mass flow MAF_KGH

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_KGH	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Air-mass flow per segment in kg/h					

FUNCTION DESCRIPTION:

General information:

Depending on system configuration the air-mass flow MAF_KGH is calculated. If no air-mass sensor is available or if the sensor diagnosis rejects the measured value (e.g. due to pulsation), a substitution value is assigned to MAF_KGH.

Formula section:

```
if ( LV_MAF_CONF = 1 )    then
    if ( LV_MAF_CTL = 1 )   then
        MAF_KGH = MAF_KGH_MES
    else
        if ( LV_MAP_CONF = 1 )then
            if ( LV_MAP_CTL = 1 )   then
                MAF_KGH =  MAF_KGH_FG_PRED
            else
                MAF_KGH = MAF_MDL_MV
            endif
        else
            MAF_KGH = MAF_MDL_MV
        endif
    endif
else
    MAF_KGH =  MAF_KGH_FG_PRED
endif
```

42.6.2 Air-mass flow difference between two segments

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_DIF	V/O	8000...7FFFH	-694.51059... 694.48940	2.12E-02	[mg/stk]
Mass air flow difference per segment					

FUNCTION DESCRIPTION:

General information:

This function calculates the difference air-mass flow between two consecutive segments MAF_DIF.

Formula section:

```
if  LV_CLC_2SEG_ENA = 0
 then    MAF_DIF  =  $MAF_n - MAF_{n-1}$
 else    MAF_DIF  =  $\frac{1}{2} * (MAF_n - MAF_{n-1})$
endif
```

42.6.3 Air mass difference for rapidly falling load (MAF_MMV)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_MMV	V/O	0...FFFFH	0...1389	2.12E-02	[mg/stk]
MAF moving mean value					

FUNCTION DESCRIPTION:

General information:

In the case of rapidly falling load, the transition to (LV_PUC) can be accelerated in order to avoid an engine speed run-up, which is undesirable.

To trigger this function (see the following chapters: ignition, ignition angle correction for trailing throttle) some load conditions have to be fulfilled. For this reason, the value MAF_MMV is calculated every segment from the corrected air mass MAF.

Formula section:

```
if  MAF_DIF > C_MAF_MMV_HYS          ( MAF_MMV for increasing air mass )
  then    MAF_MMV = MAF
  else                               ( MAF_MMV for decreasing air mass )
    if      LV_CLC_2SEG_ENA = 0
      then  MAF_MMV =  MAF_MMV + (MAF - MAF_MMV) * C_MAF_MMV_DIF_CRLC
      else  MAF_MMV =  MAF_MMV + (MAF - MAF_MMV) * C_MAF_MMV_DIF_CRLC * 2
    endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MAF_MMV_DIF_CRLC	1	0...FFH	0...0.99609	3.91E-03	[-]
MAF moving mean value filtering factor					
C_MAF_MMV_HYS	1	0...7FFFH	0...694.47880	2.12E-02	[mg/stk]
Threshold for detecting raising MAF_MMV					

42.7 Pressure Variables (MAP-Sensor, PUT-Sensor)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAP_DIP_MES_BAS	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold pressure per segment measured (differential pressure)					
MAP_DIP_MES_BAS_2SEG	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold pressure per second segment measured (differential pressure)					
MAP_DIP_MES_2SEG	O/V	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold differential pressure mean value over two segments, model value in case of sensor error					
V_MAP	V/O	0...400H	0...5	4.8828e-3	[V]
Voltage of the intake manifold pressure sensor (for diagnosis)					
PUT_MES_BAS	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold pressure up throttle per segment measured					
PUT_MES_BAS_2SEG	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold pressure up throttle per second segment measured					
V_PUT	V/O	0...400H	0...5	4.8828e-3	[V]
Voltage of the intake manifold pressure sensor up throttle					
LV_MAP_SEG	V/O	0...1H	0...1	1	[-]
Event segment, used to switch buffer					
MAP_MES	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold pressure per segment measured					
MAP_MES_BAS	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold pressure per segment measured (raw value)					
MAP_MES_BAS_2SEG	V	0...FFFFH	0...5434	0.0829175	[hPa]
Intake manifold pressure per second segment measured					
MAP_SAE	O/V	0...FFH	0...255	1	[kPA]
Intake manifold pressure raw value for OBD					

Input data:

MAP	MAP_SUM_1	MAP_CTR	MAP_CTR_1
PUT_CTR	LV_ERR_MAP_DIP_SENS	LV_ES	N_32
LV_VAR_TCHA	AMP	PUT_SUM	PUT_SUM_1
	PUT_CTR_1	PUT	LV_ERR_PUT_EL

FUNCTION DESCRIPTION:

General information:

The raw value (voltage) MAP_BAS (PUT_BAS) is measured every 1 ms. The values (10 bit) are summed up in two alternating buffers, MAP_SUM or MAP_SUM_1 (PUT_SUM or PUT_SUM_1). The number of the values are counted in comparable buffers MAP_CTR or MAP_CTR_1 (PUT_CTR or PUT_CTR_1). This mechanism is necessary to synchronize the measurement and the calculation of MAP_DIP_MES_BAS resp. PUT_MES_BAS (build mean value with a standardized range)

Depending on the logical variable LV_MAP_SEG one of the two buffers — MAP_SUM or MAP_SUM_1 (PUT_SUM or PUT_SUM_1) — and the respective counter — MAP_CTR or MAP_CTR_1 (PUT_CTR or PUT_CTR_1) — are selected for writing, the contents of the other buffer (counter) is used for calculations and diagnosis (after this buffer and the corresponding counter have been read they are cleared).

To avoid pressure sensor diagnosis errors during initialization of the ECU, the first measured value is mirrored from buffer MAP_SUM into MAP_SUM_1 resp. PUT_SUM into PUT_SUM_1 (also the counters). This ensures, that the sensor diagnosis gets reasonable values and doesn’t detect a bad sensor due to zero values in the buffer.

*MAP_DIP/PUT*_MES_BAS_2SEG is introduced to get a smooth signal of MAP_DIP_MES_BAS (filtering of possibly occurring pulsation of pressure).

Signal flow diagram:

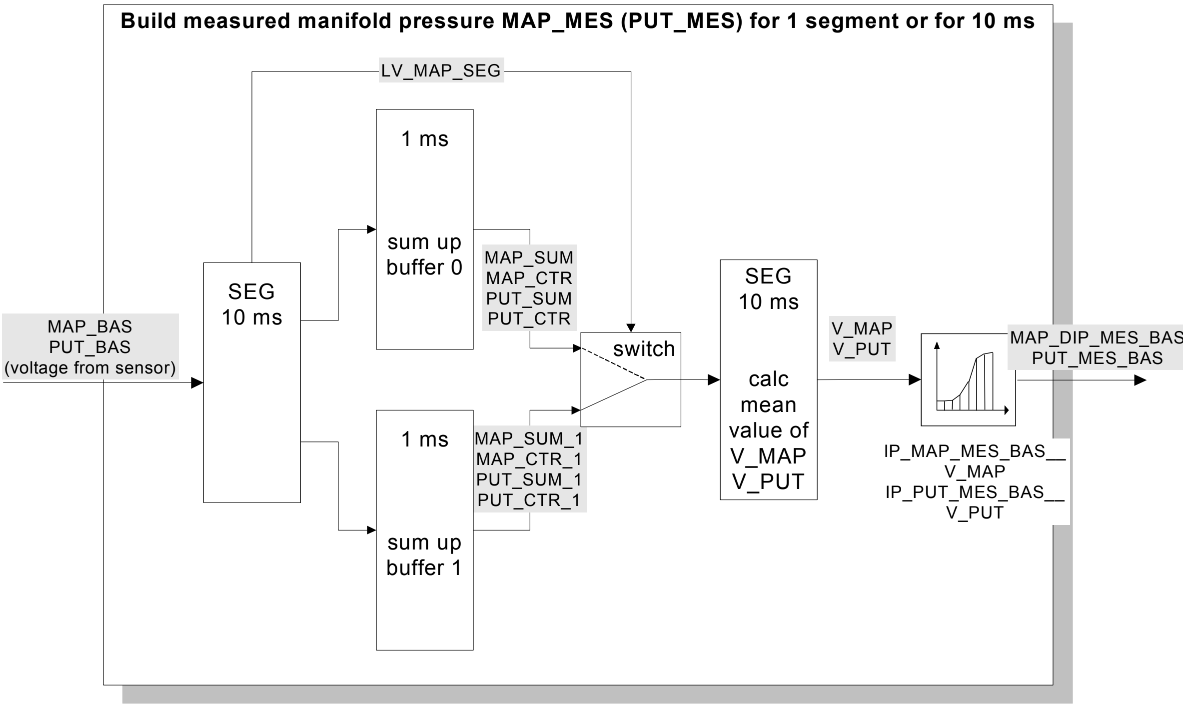


Figure 1: Signal flow during acquisition of manifold pressure MAP_DIP_MES_BAS, PUT_MES_BAS

Description:

The application SW toggles LV_MAP_SEG either at a period of 10 ms (N_32 < C_N_MAP_PUT_CLC_SWI) or at each segment (N_32 >= C_N_MAP_PUT_CLC_SWI).

Because there are used two different MAP-sensors for pressure acquisition (relativ and absolute sensors) it is necessary to calculate the absolute value as input variable for the intake manifold modell.

Application conditions:

Initialisation: all: 0
except:
MAP_MES_BAS = AMP *// AMP must be initialized before*
MAP_MES_BAS_2SEG = AMP *// AMP must be initialized before*
MAP_MES = AMP *// AMP must be initialized before*

Recurrence: at N_32 < C_N_MAP_PUT_CLC_SWI all: 10ms
at N_32 >= C_N_MAP_PUT_CLC_SWI all: segment

Activation: at every engine state

Deactivation: -

Formula section:

Calculation of V_MAP:

```
If      LV_MAP_SEG = 1          and
        MAP_CTR > 0
Then    $V\_MAP = \frac{MAP\_SUM}{MAP\_CTR}$ - V_MAP_AD
Else
        If      LV_MAP_SEG = 0          and
                MAP_CTR_1 > 0
        Then    $V\_MAP = \frac{MAP\_SUM\_1}{MAP\_CTR\_1}$ - V_MAP_AD
        Endif
Endif
```

Calculation of V_MAP_1:

```
If      MAP_CTR = 0             and
        MAP_CTR_1 = 0
Then    $V\_MAP\_1_n = V\_MAP\_1_{n-1}$
Else
    If          V_MAP > 0
    Then        $V\_MAP\_1 = \frac{(MAP\_SUM + MAP\_SUM\_1)}{(MAP\_CTR + MAP\_CTR\_1)}$ - V_MAP_AD
    Else        V_MAP_1 = 0
    Endif
Endif
```

If the selected counter is zero V_MAP(_1) and MAP(_DIP)_MES_BAS(_2SEG) remain unchanged.

Calculation of MAP_MES_BAS(_2SEG) and MAP_DIP_MES_BAS(_2SEG):

V_MAP (unit: V) has to be converted in MAP(_DIP)_MES_BAS(_2SEG) (unit: hPa)

```
If LC_MAP_ABSV_REL_SWI = 1
then    (N54)
        MAP_MES_BAS = IP_MAP_MES_BAS(V_MAP)
        MAP_MES_BAS_2SEG = IP_MAP_MES_BAS(V_MAP_1)
        MAP_DIP_MES_BAS = AMP - MAP_MES_BAS
```

```
                MAP_DIP_MES_BAS_2SEG = AMP - MAP_MES_BAS_2SEG
else            (N53)
                MAP_DIP_MES_BAS = IP_MAP_MES_BAS(V_MAP)
                MAP_DIP_MES_BAS_2SEG = IP_MAP_MES_BAS(V_MAP_1)
                MAP_MES_BAS = AMP - MAP_DIP_MES_BAS
                MAP_MES_BAS_2SEG = AMP - MAP_DIP_MES_BAS_2SEG
endif
If              LV_ERR_MAP_DIP_SENS = 0
then            MAP_MES = MAP_MES_BAS
                MAP_DIP_MES_2SEG = MAP_DIP_MES_BAS_2SEG
else            MAP_MES = MAP     (model value)
                MAP_DIP_MES_2SEG = AMP – MAP (model value)
endif
```

<u>Calculation of OBD relevant MAP_SAE:</u>

```
MAP_SAE = IP_MAP_MES_BAS(V_MAP)          //Convertion must be done in physical way
```

Additional variables for LV_VAR_TCHA = 1 (turbo)

<u>Calculation of V_PUT:</u>

```
If              LV_MAP_SEG = 1            and
                PUT_CTR > 0
Then            $V\_PUT = \frac{PUT\_SUM}{PUT\_CTR}$
Else
                If      LV_MAP_SEG = 0            and
                        PUT_CTR_1 > 0
                Then    $V\_PUT = \frac{PUT\_SUM\_1}{PUT\_CTR\_1}$
                Endif
Endif
```

If the selected counter is zero V_PUT and PUT_MES_BAS stay unchanged.

Calculation of PUT_MES_BAS:

V_PUT (unit: V) has to be converted in PUT_MES_BAS (unit: hPa)

```
If      LV_ERR_PUT_EL = 0
Then            if      LC_PUT_ABSV_REL_SWI =1
                then    PUT_MES_BAS = IP_PUT_MES_BAS
                else    PUT_MES_BAS = AMP - IP_PUT_MES_BAS
                endif

Else    PUT_MES_BAS = PUT               (calculated from Model)
Endif
```

Calculation of V_PUT_1:

```
If      PUT_CTR = 0             and
        PUT_CTR_1 = 0
Then    $V\_PUT\_1_n = V\_PUT\_1_{n-1}$
Else
    If      V_PUT > 0
    Then    $V\_PUT\_1 = \frac{(PUT\_SUM + PUT\_SUM\_1)}{(PUT\_CTR + PUT\_CTR\_1)}$
    Else    V_PUT_1 = 0
    Endif
Endif
```

Calculation of PUT_MES_BAS_2SEG:

V_PUT_1 (unit: V) has to be converted into PUT_MES_BAS_2SEG (unit: hPa)

```
If      LV_ERR_PUT_EL = 0
Then    if      LC_PUT_ABSV_REL_SWI = 1
        then    PUT_MES_BAS_2SEG = IP_PUT_MES_BAS
        else    PUT_MES_BAS_2SEG = AMP - IP_PUT_MES_BAS
        endif

Else    PUT_MES_BAS_2SEG = PUT          (calc. from Model)
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_MAP_MES_BAS	2	0...FFFFH	0...5434	0.0829175	[hPa]
LDP_V_MAP__MAP_MES_BAS	2	0...400H	0...5	4.8828e-3	[V]
Conversion characteristic for MAP_DIP_MES_BAS					
IP_PUT_MES_BAS	2	0...FFFFH	0...5434	0.0829175	[hPa]
LDP_V_PUT__PUT_MES_BAS	2	0...400H	0...5	4.8828e-3	[V]
Conversion characteristic for PUT_MES_BAS					
C_N_MAP_PUT_CLC_SWI	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for switching between time and segment synchronous calculation					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_MAP_ABSV_REL_SWI	-	0...1H	0...1	1	[-]
Switch for Calculation MAP_DIP_MES_BAS with measured absolute (1) or relative pressure (0)					
LC_PUT_ABSV_REL_SWI	-	0...1H	0...1	1	[-]
Switch for Calculation PUT_MES_BAS with measured absolute (1) or relative pressure (0)					

42.7.1 Adaptation of sensor characteristic

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
V_MAP_AD	V/O/S	FE00...200H	-2.5...2.5	4.8828e-3	[V]
Adapted voltage offset of manifold pressure sensor					
TEMP_MAP_DIP_MDL	V	0...FEH	-48...142.5	0.75	[°C]
Estimated temperature of manifold pressure sensor					
LV_T_DLY_MAP_MES_AD	V	0...1H	0...1	1	[-]
Logical value delay time for manifold evacuation is over					
V_MAP_DIF	V/O	FE00...200H	-2.5...2.5	4.8828e-3	[V]
Deviation to nominal output at differential pressure 0					
V_MAP_DIF_VLD	V/O	FE00...200H	-2.5...2.5	4.8828e-3	[V]
Allowed sensor output deviation at differential pressure 0					

Input data:

TIA	TPS_AV	TCO	TCO_EX
VS_FIL	TAM	ECU_STATE	

FUNCTION DESCRIPTION:

General information:

If MAP is evaluated from a differential pressure sensor (MAP_DIP), the sensor characteristic can be adapted due to the known differential pressure (0 hPa) when the engine doesn't run.

Application conditions:

Initialisation:

```
V_MAP_AD from nonvolatile memory   or   = 0 V in case of memory error or
when the NVMY is reset
CTR_NR_V_MAP_ACQ = 0                      ┐
LV_V_MAP_ACQ = 0                          |
T_DLY_MAP_MES_AD = 0                      |
LV_T_DLY_MAP_MES_AD = 0                   |
TEMP_MAP_DIP_MDL = TCO                    |-  at Activation
V_MAP_SUM = 0 V                           |
V_MAP_DIF = 0 V                           |
V_MAP_DIF_VLD = C_V_MAP_SP_DIF_MAX        ┘
```

Recurrence: 10 ms
except TEMP_MAP_DIP_MDL :   500 ms

Activation:

```
ECU_STATE = PWL                      and
LC_MAP_ABSV_REL_SWI = 0              and    //only for DIP-Sensor
TPS_AV ≥ C_TPS_V_MAP_AD_MIN          and    //the manifold can be evacuated
VS_FIL = 0
```

Deactivation: !Activation

Formula section:

Estimation of MAP_DIP sensor temperature
Done only as long as LV_V_MAP_ACQ = 0 , every 500 ms

$$\begin{aligned}\mathrm{TEMP\_MAP\_DIP\_MDL} = {} & (((((\mathrm{TCO} - \mathrm{TCO\_EX}) * \mathrm{C\_FAC\_TEMP\_MAP\_DIP\_TCO\_EX} \\ & + \mathrm{TCO\_EX}) - \mathrm{TIA}) * \mathrm{C\_FAC\_TEMP\_MAP\_DIP\_TIA} + \mathrm{TIA}) \\ & - \mathrm{TAM}) * \mathrm{C\_FAC\_TEMP\_MAP\_DIP\_TAM} + \mathrm{TAM}\end{aligned}$$

The following formulas are processed each recurrence in the same sequence as described here.

Delay collection of measurements until the manifold is evacuated
Done only as long as LV_T_DLY_MAP_MES_AD = 0

```
If    T_DLY_MAP_MES_AD < C_T_DLY_MAP_MES_AD
  then      T_DLY_MAP_MES_AD++                    //incremented by 10 ms
  else      LV_T_DLY_MAP_MES_AD = 1
endif
```

Evaluation of a mean sensor output and of deviation to nominal output at differential pressure 0
Done only as long as LV_V_MAP_ACQ = 0 **and** LV_T_DLY_MAP_MES_AD = 1

```
If    CTR_NR_V_MAP_ACQ < C_NR_V_MAP_ACQ
  then      CTR_NR_V_MAP_ACQ++
            $V\_MAP\_SUM_{n} = V\_MAP\_SUM_{n-1} + V\_MAP$
  else      $V\_MAP\_MV = V\_MAP\_SUM_{n-1} / C\_NR\_V\_MAP\_ACQ$
            V_MAP_DIF  =  V_MAP_MV - C_V_MAP_SP_AD
            LV_V_MAP_ACQ  =  1
endif
```

Evaluation of allowed deviation and of the factor for the adaptation
Done only if LV_V_MAP_ACQ = 1

```
If    TEMP_MAP_DIP_MDL ≥ C_TEMP_MAP_DIP_MDL_FAC_MIN              or
      TEMP_MAP_DIP_MDL ≤ C_TEMP_MAP_DIP_MDL_FAC_MAX
  then        V_MAP_DIF_VLD  =  C_V_MAP_SP_DIF_MAX
              FAC_V_MAP_AD  =  C_FAC_V_MAP_AD
  else        V_MAP_DIF_VLD  =  C_V_MAP_SP_DIF_MAX *
                                C_FAC_TEMP_V_MAP_SP_DIF_MAX
              FAC_V_MAP_AD  =  C_FAC_V_MAP_AD * C_FAC_TEMP_V_MAP_AD
endif
```

<u>Calculation of the sensor characteristic adaptation value (voltage offset)</u>

Done only if LV_V_MAP_ACQ = 1

```
If    | V_MAP_DIF |  ≤  V_MAP_DIF_VLD
  then        V_MAP_AD  =  V_MAP_AD(old) + V_MAP_DIF * FAC_V_MAP_AD
              LV_V_MAP_ACQ  =  0                        //enables next adaptation loop
              CTR_NR_V_MAP_ACQ  =  0
              V_MAP_SUM  =  0

      if    V_MAP_AD > C_V_MAP_AD_MAX
        then          V_MAP_AD  =  C_V_MAP_AD_MAX
        else
              if    V_MAP_AD  <  -1 * C_V_MAP_AD_MAX
                then  V_MAP_AD  =  -1 * C_V_MAP_AD_MAX
              endif
      endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TPS_V_MAP_AD_MIN	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Minimum TPS to enable V_MAP adaptation					
C_FAC_TEMP_MAP_DIP_TCO_EX	1	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for TCO_EX correction for MAP_DIP temperature model					
C_FAC_TEMP_MAP_DIP_TIA	1	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for TIA correction for MAP_DIP temperature model					
C_FAC_TEMP_MAP_DIP_TAM	1	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for TAM correction for MAP_DIP temperature model					
C_T_DLY_MAP_MES_AD	1	0...3FFH	0...10.23	0.01	[s]
Delay time after LV_ES detection before starting adaptation of sensor characteristic					
C_NR_V_MAP_ACQ	1	0...FFH	0...255	1	[-]
Number of acquisitions to build the mean value V_MAP_MV					
C_V_MAP_SP_AD	1	0...200H	0...2.5	4.8828e-3	[V]
Setpoint of MAP sensor output at stopped engine					
C_TEMP_MAP_DIP_MDL_FAC_MIN	1	0...FEH	-48...142.5	0.75	[°C]
Lower TEMP_MAP_DIP_MDL temperature threshold to switch adaptation/diagnosis values					
C_TEMP_MAP_DIP_MDL_FAC_MAX	1	0...FEH	-48...142.5	0.75	[°C]
Upper TEMP_MAP_DIP_MDL temperature threshold to switch adaptation/diagnosis values					
C_V_MAP_SP_DIF_MAX	1	FE00...200H	-2.5...2.5	4.8828e-3	[V]
Allowed sensor output deviation at stopped engine at regular temperature condition					
C_FAC_TEMP_V_MAP_SP_DIF_MAX	1	0...FFH	0...1.99218	0.0078125	[-]
Factor on allowed sensor output deviation at stopped engine at irregular temperature condition					
C_FAC_V_MAP_AD	1	0...FFH	0...0.99609	3.9063e-3	[-]
Part of sensor output deviation at stopped engine used for sensor characteristic adaptation					
C_FAC_TEMP_V_MAP_AD	1	0...FFH	0...0.99609	3.9063e-3	[-]
Factor on adaptation part at irregular temperature condition					
C_V_MAP_AD_MAX	1	0...200H	0...2.5	4.8828e-3	[V]
Maximum absolute value of adapted voltage offset of manifold pressure sensor					

42.8 Sensor specific Air-Mass Flow Variables

42.8.1 Air-mass flow acquisition MAF_KGH_MES

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_KGH_MES	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Corrected mass air flow, measured per segment					
MAF_KGH_MES_BAS	O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Raw air-mass flow measured (without correction, to be used for diagnosis only)					
MAF_KGH_MES_TCHA[NC_MAF_SENS_CONF]	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Corrected mass air flow up turbo charger, measured per segment					
MAF_KGH_MES_BAS_TCHA[NC_MAF_SENS_CONF]	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Raw air-mass flow up turbo chager measured (without correction, to be used for diagnosis only)					
MAF_MES	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Air-mass flow per segement					
LV_MAF_PULS	V/O	0...1H	0...1	1	[-]
Boolean for pulsation present					

Input data:

N	NC_MAF_FAC_CYL	MAF	LV_VAR_TCHA
PQ_SP	MAF_FRQ_KGH_MES[NC_MAF_NR]	TIA	LV_MAF_SEG
MAF_SUM	MAF_SUM_1	MAF_CTR	MAF_CTR_1
TIA_TCHA	LV_IGK	STATE_ENG	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_MAF_KGH_COR	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_MAF_KGH_MES_BAS	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDP_TIA_IP_FAC_MAF_KGH_COR	6	0...FEH	-48...142.5	0.75	[°C]
Correction for MAF_KGH_MES depending on raw signal and on intake air temperature					
IP_FAC_MAF_KGH_COR_FRQ	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_MAF_KGH_MES_BAS_TCHA_0	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDP_TIA_TCHA_IP_MAF_KGH_COR_FRQ	6	0...FEH	-48...142.5	0.75	[°C]
Correction for MAF_KGH_MES_TCHA[0] depending on raw signal and on intake air temperature					
IP_FAC_MAF_KGH_COR_AN	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_MAF_KGH_MES_BAS_TCHA_1	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDP_TIA_TCHA_IP_MAF_KGH_COR_AN	6	0...FEH	-48...142.5	0.75	[°C]
Correction for MAF_KGH_MES_TCHA[1] depending on raw signal and on intake air temperature					
IP_PQ_SP_PULS	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_N__PQ_SP_PULS	6	0...1FE0H	0...8160	1	[rpm]
PQ-threshold for LV_MAF_PULS					
C_PQ_SP_HYS_PULS	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Hysteresis for PQ-threshold for LV_MAF_PULS					

FUNCTION DESCRIPTION:

General information:

For non turbo variant (N53) a frequency MAF sensor is used. For turbo variant (N54) a frequency MAF sensor (index [0]) and an analog MAF sensor (index [1]) is used.

Analog sensor: Depending on the logical value LV_MAF_SEG one of the two buffers MAF_SUM or MAF_SUM_1 and the respective counter MAF_CTR or MAF_CTR_1 is selected for writing, the contents of the other buffer (counter) is used for diagnosis and calculations. After having read out one buffer MAF_SUM or MAF_SUM_1 the buffer has to be cleared as well as the corresponding counter MAF_CTR or MAF_CTR_1.

Analog and frequency sensor: A correction is possible in order to take into consideration deviations of the measured mass air flow due to different installations of the air mass meter. This correction factor is dependant from MAF_KGH_MES_BAS itself and of the air temperature.

Application conditions:

Initialisation: all 0 at reset **or** LV_IGK 0->1

Recurrence: at STATE_ENG: ST, IS, PL, PU and PUC segment synchronous

at STATE_ENG: ES 1s

Activation: at all STATE_ENG

Deactivation: -

Formula section:

```
If      LV_VAR_TCHA = 0
then    // evaluation of frequency MAF sensor (N53)
        MAF_KGH_MES_BAS = MAF_FRQ_KGH_MES[0]
        MAF_KGH_MES = MAF_KGH_MES_BAS *
                IP_FAC_MAF_KGH_COR(MAF_KGH_MES_BAS, TIA)
else    // evaluation of frequency and analog MAF sensor  (N54)
        MAF_KGH_MES_BAS = (MAF * N) / NC_MAF_FAC_CYL
        MAF_KGH_MES = MAF_KGH_MES_BAS
        MAF_KGH_MES_BAS_TCHA[0] = MAF_FRQ_KGH_MES[0]
        MAF_KGH_MES_TCHA[0] = MAF_KGH_MES_BAS_TCHA[0] *
                IP_FAC_MAF_KGH_COR_FRQ(MAF_KGH_MES_BAS_TCHA[0], TIA_TCHA)
        If          LV_MAF_SEG = 1
        then        MAF_KGH_MES_BAS_TCHA[1] = MAF_SUM / MAF_CTR
        else        MAF_KGH_MES_BAS_TCHA[1] = MAF_SUM_1 / MAF_CTR_1
        endif
        MAF_KGH_MES_TCHA[1] = MAF_KGH_MES_BAS_TCHA[1] *
                IP_FAC_MAF_KGH_COR_AN(MAF_KGH_MES_BAS_TCHA[1], TIA_TCHA)
endif
```

42.8.2 Air-mass Flow Acquisition MAF_MES

FUNCTION DESCRIPTION:

General information:

The air-mass MAF_MES in mg/stroke is calculated from the measured air-mass flow MAF_KGH_MES and the engine speed N. The non calibratable constant NC_MAF_FAC_CYL is inversly proportional to the number of cylinders, so the output MAF_MES depends on the number of cylinders.

Application conditions:

Initialisation: all 0 at reset **or** LV_IGK 0->1

Recurrence: at STATE_ENG: ST, IS, PL, PU and PUC segment synchronous

at STATE_ENG: ES 1s

Activation: at all STATE_ENG

Deactivation: -

Formula section:

```
If      N = 0
Then    MAF_MES = 0
Else    $MAF\_MES = \frac{MAF\_KGH\_MES}{N} * NC\_MAF\_FAC\_CYL$
Endif
```

42.8.3 Fade out of Mass Airflow Sensor

FUNCTION DESCRIPTION:

General information:

It is possible, that in some operating points of an engine the MAF sensor is unable to deliver the correct MAF value. Those operating points are displayed by LV_MAF_PULS = 1. This state will result in switching off the MAF feedback loop of the intake manifold model, so the load determination will be done like "α-/N-control".

Application conditions:

Initialisation: all 0 at reset **or** LV_IGK 0->1

Recurrence: at STATE_ENG: ST, IS, PL, PU and PUC segment synchronous

at STATE_ENG: ES 1s

Activation: at all STATE_ENG

Deactivation: -

Formula section:

```
If      PQ_SP > IP_PQ_SP_PULS
Then    LV_MAF_PULS = 1
Else    If      PQ_SP < IP_PQ_SP_PULS - C_PQ_SP_HYS_PULS
        Then    LV_MAF_PULS = 0
        Else    LV_MAF_PULS remains unchanged
        Endif
Endif
```

42.9 Ambient Pressure variables

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_AMP	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
factor of the ambient pressure related to the reference ambient pressure					
AMP_MES	O/V	0...FFFFH	0...5434	0.0829175	[hPa]
valid ambient pressure					
LV_AMP_GRD_MAX	V	0...1H	0...1	1	[-]
ambient pressure maximum gradient exceeded					
AMP_RAW	V	0...FFFFH	0...5434	0.0829175	[hPa]
ambient pressure raw value					
AMP_COR	-	0...FFFFH	0...5434	0.0829175	[hPa]
Corrected ambient pressure					
AMP_RAW_NEW	-	0...FFFFH	0...5434	0.0829175	[hPa]
Interim value for calculation of ambiant pressure raw value					
AMP_RAW_FRZ	-	0...FFFFH	0...5434	0.0829175	[hPa]
Interim value for calculation of AMP_RAW substitute value					
SUM_AMP_RAW	-	0...FFFFFFFFH	0...356128058	0.0829175	[hPa]
summarized ambient pressure raw value					
T_DLY_AMP_SUB	V	0...FFH	0...25500	100	[ms]
Delay time before AMP_RAW value substitution					
CTR_SAMPLES	-	0...FFH	0...255	1	[-]
counter for average determination					

Input data:

V_AMP	VS	LV_ERR_AMP	C_V_AMP_MAX_DIAG
C_V_AMP_MIN_DIAG	LV_ERR_AMP_PLAUS	LV_ERR_AMP_PLAUS_CUS	LV_ERR_PUT_EL
LV_ERR_PUT_PLAUS	PUT_MES_BAS	MAF_KGH_MES	LV_VAR_TCHA
STATE_ENG			

FUNCTION DESCRIPTION:

Application conditions:

Activation : at every engine operating state
Deactivation : -
Initialization : At Reset:

```
If(1)   V_AMP < C_V_AMP_MIN_DIAG     or
        V_AMP >  C_V_AMP_MAX_DIAG
Then(1)         AMP_MES = C_AMP_REF
                FAC_AMP = 1(phys)
                LV_AMP_GRD_MAX = 1
Else(1) If(2)         LV_ERR_AMP = 0
        Then(2)       AMP_RAW = IP_AMP_RAW__V_AMP
        Else(2)       AMP_ RAW = C_AMP_REF
        Endif(2)
```

```
            AMP_COR  =  AMP_RAW + IP_AMP_RAW_COR__VS
            AMP_MES = AMP_COR        // Calculated with first valid measurement
                                                                  value V_AMP //
            FAC_AMP =   AMP_MES / C_AMP_REF  // C_FAC_AMP_MIN  ≤
                                              FAC_AMP  ≤  C_FAC_AMP_MAX //

            LV_AMP_GRD_MAX = 0
        Endif(1)

        AMP_RAW_FRZ = C_AMP_REF
```

Update rate: 100 ms

Formula section:

```
If(1)     LV_ERR_AMP = 0                     and
          LV_ERR_AMP_PLAUS = 0               and
          LV_ERR_AMP_PLAUS_CUS = 0
Then(1)   If(2)         V_AMP < C_V_AMP_MIN_DIAG     or
                        V_AMP > C_V_AMP_MAX_DIAG
          Then(2)       AMP_RAW(n) = AMP_RAW(n-1)  // V_AMP out of range, keep last valid
                                                                              value //
          Else(2)       AMP_RAW_NEW = IP_AMP_RAW__V_AMP
                        If(3)     LV_AMP_GRD_MAX = 0                    and
                                  AMP_RAW_NEW – AMP_RAW > C_AMP_HYS
                        Then(3)   LV_AMP_GRD_MAX = 1
                                  $AMP\_RAW_{(n)} = AMP\_RAW_{(n-1)}$   // Keep old value //
                        Else(3)   AMP_RAW = AMP_RAW_NEW
                                  LV_AMP_GRD_MAX = 0
                        Endif(3)

Else(1)   LV_AMP_GRD_MAX = 1
          If(2)         LV_VAR_TCHA = 0  // Qustion if there is a turbocharger and thus
                                                                  also a PUT sensor //
          Then(2)       AMP_RAW = C_AMP_REF
          Else(2)   If(3)     LV_ERR_PUT_EL = 0                               and
```

```
                                LV_ERR_PUT_PLAUS = 0                          and
                     Then(3)    If(4)      MAF_KGH_MES <
                                                 C_MAF_THD _AMP_SUB            and
                                           STATE_ENG = 2                        or
                                           STATE_ENG = 4                        or
                                           STATE_ENG = 5
                                Then(4)  T_DLY_AMP_SUB ++          // timer starts counting,
                                                                      no overflow allowed //
                                           If(5)       T_DLY_AMP_SUB >
                                                       C_T_DLY_AMP_SUB
                                           Then(5) AMP_RAW = PUT_MES_BAS
                                                   AMP_RAW_FRZ =  PUT_MES_BAS
                                                    // Freeze last PUT_MES_BAS value //
                                           Else(5)  AMP_RAW = AMP_RAW_FRZ   // Use
last  AMP_RAW_FRZ (= PUT_MES_BAS)  value at T_DLY_ERR_AMP > C_T_DLY_AMP_SUB conditions  //
                                          Endif(5)
                                 Else(4)  AMP_RAW = AMP_RAW_FRZ
                                          T_DLY_AMP_SUB = 0
                                Endif(4)
                     Else(3)    AMP_RAW = C_AMP_REF
                    Endif(3)
             Endif(2)
Endif(1)



SUM_AMP_RAW = AMP_RAW + SUM_AMP_RAW          // summation of actual and preceding
                                                                amp_raw values   //

If(1)        CTR_SAMPLES < 9
Then(1)      CTR_SAMPLES ++
Else(1)      AMP_RAW_NEW = SUM_AMP_RAW / 10
             AMP_COR  =  AMP_RAW_NEW + IP_AMP_RAW_COR__VS
                 // Compensation of the back pressure of the vehicle, due to the vehicle speed //

             AMP_MES  =  AMP_MES + (AMP_COR - AMP_MES) * C_AMP_CRLC
                // Suppression of the signal noise //
```

$$\text{FAC\_AMP} = \frac{\text{AMP\_MES}}{\text{C\_AMP\_REF}}$$

// Additionally the factor FAC_AMP is calculated which is used in several

functions and limited by C_FAC_AMP_MAX and C_FAC_AMP_MIN //

C_FAC_AMP_MIN ≤ FAC_AMP ≤ C_FAC_AMP_MAX

SUM_AMP_RAW = 0 *// Clear counter //*

CTR_SAMPLES = 0 *// Clear counter //*

Endif(1)

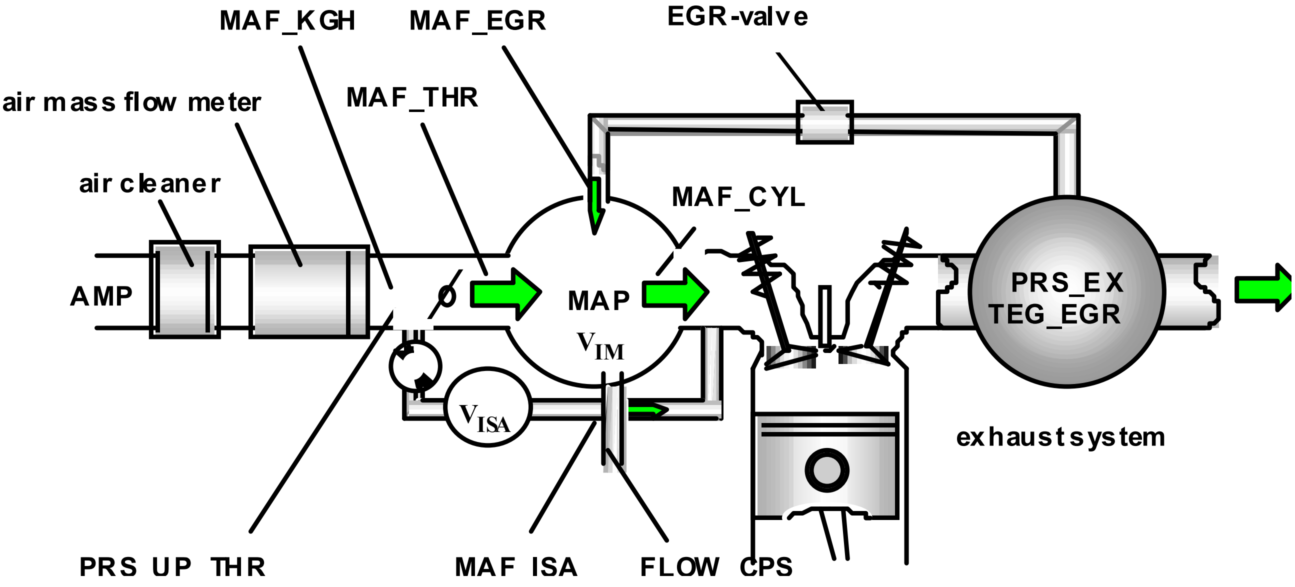
Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_AMP_RAW__V_AMP	8	0...FFFFH	0...5434	0.0829175	[hPa]
LDP_V_AMP__AMP_RAW	8	0...3FFH	0...5	4.8876e-3	[V]
correlation table for the linearization of the ambient pressure value acquisition with motorola sensor					
IP_AMP_RAW_COR__VS	8	0...FFFFH	0...5434	0.0829175	[hPa]
LDP_VS__AMP_RAW_COR	8	0...FFH	0...255	1	[km/h]
correlation table for the linearization of the ambient pressure with the vehicle speed					
C_AMP_HYS	1	0...FFFFH	0...5434	0.0829175	[hPa]
maximum value between two acquired ambient pressure values					
C_AMP_CRLC	1	0...FFH	0...0.99609	3.9063e-3	[-]
correlation value for the compensation of the noise signal of the ambient pressure signal					
C_AMP_REF	1	0...FFFFH	0...5434	0.0829175	[hPa]
reference value of the ambient pressure					
C_FAC_AMP_MIN	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
BOT for FAC_AMP					
C_FAC_AMP_MAX	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
TOP for FAC_AMP					
C_MAF_THD_AMP_SUB	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
MAF threshold for AMP_RAW value substitution					
C_T_DLY_AMP_SUB	1	0...FFH	0...25500	100	[ms]
Delay time threshold for AMP_RAW value substitution					

42.10 Intake Manifold Model with Turbulence System and EGR

General information:

The intake manifold model calculates the air mass flow at throttle, at idle speed actuator valve, the air mass flow into the cylinder and the pressure in the intake manifold. In case of exhaust gas recirculation in addition the residual gas partial pressure and the fresh gas partial pressure are calculated. The calculation is done every segment due to charge determination for injection and ignition. Therefore it is necessary to scan the throttle- and idle speed actuator valve position also every segment.



Signal flow diagram:

Application conditions:

Recurrence:	segment synchronous
Activation:	at LV_IGK = 1
Deactivation:	-

42.10.1 Calculation of the manifold pressure MAP

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PQ_EGR	V/O	0000...FFFFH	0...0.999985	1.52e-5	-
pressure quotient at the EGR - valve					

Input data:

MAP	PRS_EX_EGR		

Application conditions:

Initialisation: PQ_EGR =0

Formula section:

$PQ\_EGR_N = MAP_N / PRS\_EX\_EGR_N$

42.10.2 Calculation of MAP_FG

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_FG_CYL	V/O	0000...FFFFH	0...2047.96875	0.03125	kg/h
fresh air mass flow in the cylinder					

Input data:

MAF_CYL	MAF_EGR		

Application conditions:

Initialisation: all 0 at Reset

Formula section:

the air entering in the cylinder presently is given by :

MAF_FG_CYL = MAF_CYL - MAF_EGR

42.10.3 Calculation of maximum possible mass air flow MAF_MAX_COR and the ratio between actual and maximum mass air flow FAC_MAF_MAX

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_MAF_MAX	V/O	0H...FFFFH	0...2	3.052e-5	-
ratio between MAF and MAF_MAX_COR					

Input data:

MAF_SP_TQI	MAF_MAX_COR		

Application conditions:

Initialisation:

Formula section:

FAC_MAF_MAX = $MAF\_SP\_TQI_N$ / $MAF\_MAX\_COR_N$

42.11 Application Inc. Intake Manifold Model (MAF-System)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MAF_CTL	V/0	0...1H	0...1	1	-
1: MAF controlled system					

Input data:

LV_MAF_PULS	LV_ERR_MAF	LV_ERR_LOAD_TPS_PLAUS	ERR_SYM_LOAD_TPS_PLAUS

Application conditions:

Initialisation: LV_MAF_CTL = 1

Recurrence: 10 ms

Activation: every engine state

Deactivation: -

Formula section:

```
if                        LV_ERR_MAF = 1
        or                LV_MAF_PULS = 1
        or                [ LV_ERR_LOAD_TPS_PLAUS = 1 and
                          ERR_SYM_LOAD_TPS_PLAUS = SYM_0 ]
then    α-/N-control:     LV_MAF_CTL = 0
else    MAF-control:      LV_MAF_CTL = 1
endif
```

42.12 Pressure decrease in the air cleaner

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
PUT	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Pressure upstream throttle					
PUT_MAX	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Maximum possible pressure upstream throttle					

Input data:

AMP_MDL	AMP_AD	MAF_CYL	LV_CLC_2SEG
AMP_MES	PUT_MES_BAS]	LV_VAR_TCHA	LV_ERR_PUT_EL
LV_ES		LV_ERR_PUT_PLAUS	

FUNCTION DESCRIPTION:

General information:

With increasing air mass flow through the air cleaner there is a drop of pressure in the air cleaner in comparison to the ambient pressure.

Application conditions:

Initialisation: PUT = AMP_MES

PUT_MAX = AMP_MES at ECU reset

Activation: LV_CLC_2SEG = 0

Deactivation: LV_CLC_2SEG = 1

No pressure decrease will be calculated, if all elements of IP_AMP_DEC are 0.

Recurrence: LV_ES = 10 ms

LV_ES = 0 segment synchronous

Formula section:

```
If      LV_VAR_TCHA = 0
Then    PUT = AMP_MDL - IP_AMP_DEC
        PUT_MAX = AMP_AD - IP_AMP_DEC
Else    If      LV_ERR_PUT_EL = 1  or
              ( LV_ERR_PUT_PLAUS = 1 and
                LC_ENA_ PUT_SUB_PLAUS_ERR = 1 )

        Then    PUT = AMP_MDL - IP_AMP_DEC
                PUT_MAX = AMP_AD - IP_AMP_DEC
```

```
        Else    PUT = PUT_MES_BAS
                PUT_MAX = AMP_AD - IP_AMP_DEC
        Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_AMP_DEC	4	0...FFFFH	0...5434	8.29E-02	[hPa]
LDP_MAF_CYL__AMP_DEC	4	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Pressure decrease through the air cleaner					
LC_ENA_PUT_SUB_PLAUS_ERR	1	0...1H	0...1	1	[-]
Switch to enable PUT substitution during PUT_PLAUS_ERR					

42.13 Temperature correction of Volumetric Efficiency

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EFF_VOL_TEMP_COR	V	0...FFFFH	0...1.99996	3.05E-05	[-]
Volumetric efficiency correction factor due to temperature effects					
EFF_VOL_TEMP_COR_MMV	V/O	0...FFFFH	0...1.99996	3.05E-05	[-]
Filtered volumetric efficiency correction factor due to temperature effects					

Input data:

MAF	N_32	TCO	TIA_IM
	LV_CLC_2SEG	LV_CLC_2SEG_ENA	

General information:

Target of the temperature correction function is, to compensate the temperature influences onto the volumetric efficiency calculation. The air-mass flow into the cylinder is the product of the volume flow and the density of the air $\dot{m} = A \cdot w \cdot \rho(T) = \dot{V} \cdot \rho(T)$, where the density depends on the air-temperature. At low temperatures the air-density is much higher than at standard temperatures, therefore the air flowing into the cylinder at the same manifold pressure will increase. The basic volumetric efficiency usually is calibrated at standard temperatures of about 25 °C for intake air and about 90 °C for coolant temperature. To take into account the different density at operating points with other temperatures, the volumetric efficiency then has to be corrected.

Especially at speed-density systems, this temperature correction is essential, because the density of the air and therefore the air-mass flow strongly depend on the air temperature. This means all deviations from the temperatures at which the volumetric efficiency was calibrated will lead to a wrong air-mass flow value into the cylinder (e.g. wrong MAF value at cold starts without temperature corrections for a MAP system).

At systems equipped with an HFM sensor, the temperature correction is not as essential as for speed-density systems. Nevertheless, it is used to consider the heating of the air due to heat transfer from the warm intake duct to the air.

The compensation function is capable of taking into account deviations of coolant temperature and intake air temperature separately.

$$EFF\_VOL\_COR = \frac{273 + C\_TCYL\_STND}{273 + C\_TCYL\_STND + IP\_EFF\_TCO\_FAC \cdot (TCO\_C\_TCO\_STND) + IP\_EFF\_TIA\_IM\_FAC \cdot (TIA\_IM - C\_TIA\_STND)}$$

Signal flow diagram:

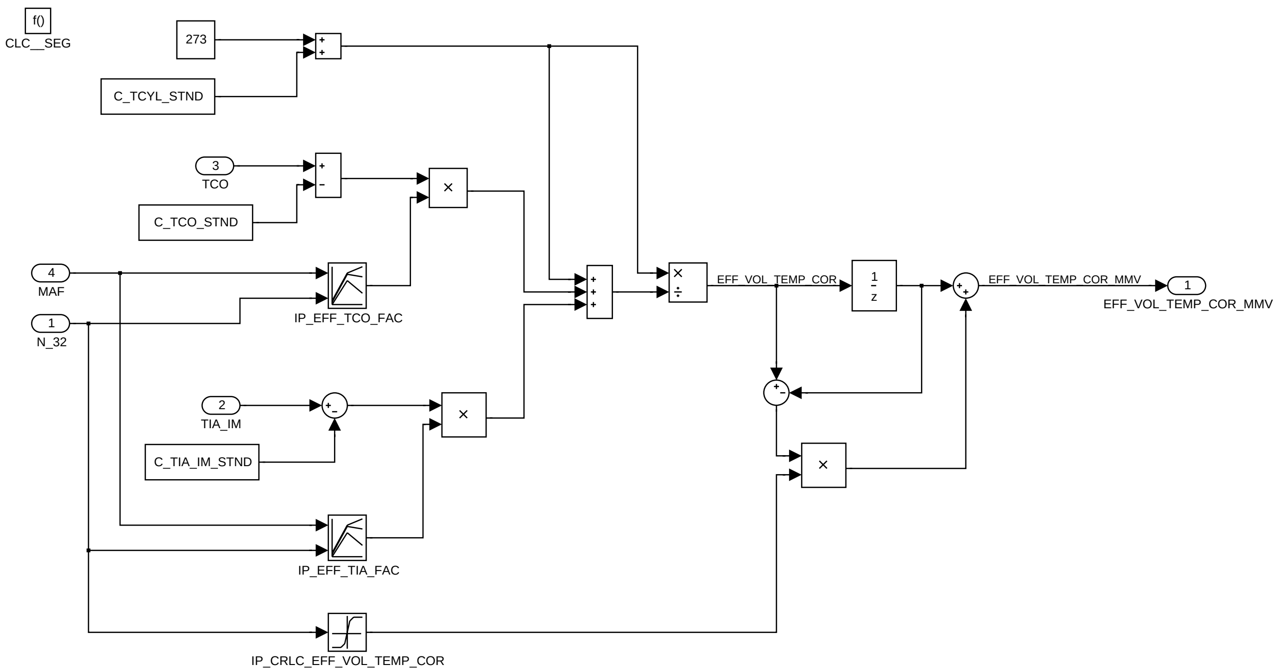


Figure 1: Calculation algorithm for temperature correction

Description:

The temperature correction factor EFF_VOL_TEMP_COR is the ratio between air temperature in the cylinder for standard conditions TCYL compared to air temperature in the cylinder for current TCO, TIA_IM and operating point. TCO and TIA_IM are compared to standard conditions and weighted by operating point.

Application conditions:

Initialisation: *at reset and engine running to stopped engine*

EFF_VOL_TEMP_COR = 1

Remark: The typical range of EFF_VOL_TEMP_COR is 0,8 to 1,5.

Activation: LV_CLC_2SEG = 0

Deactivation: LV_CLC_2SEG = 1

Recurrence: segment synchronous

Formula section:

$$\mathrm{EFF\_VOL\_TEMP\_COR} = [\,273 + \mathrm{C\_TCYL\_STND}\,] \;/\; [\,273 + \mathrm{C\_TCYL\_STND} + \mathrm{IP\_EFF\_TCO\_FAC} * (\mathrm{TCO} - \mathrm{C\_TCO\_STND}) + \mathrm{IP\_EFF\_TIA\_IM\_FAC} * (\mathrm{TIA\_IM} - \mathrm{C\_TIA\_IM\_STND})\,]$$

Changes in the operating point cause thermal inertia effects, which have to be taken into account. Filtering EFF_VOL_TEMP_COR does this.

At first calculation EFF_VOL_TEMP_COR_MMV is = EFF_VOL_TEMP_COR.

```
if LV_CLC_2SEG_ENA = 0
    then  $\mathrm{EFF\_VOL\_TEMP\_COR\_MMV}_n = \mathrm{EFF\_VOL\_TEMP\_COR\_MMV}_{n-1} + (\mathrm{EFF\_VOL\_TEMP\_COR} - \mathrm{EFF\_VOL\_TEMP\_COR\_MMV}_{n-1}) * \mathrm{IP\_CRLC\_EFF\_VOL\_TEMP\_COR}$
    else  $\mathrm{EFF\_VOL\_TEMP\_COR\_MMV}_n = \mathrm{EFF\_VOL\_TEMP\_COR\_MMV}_{n-1} + (\mathrm{EFF\_VOL\_TEMP\_COR} - \mathrm{EFF\_VOL\_TEMP\_COR\_MMV}_{n-1}) * 2 * \mathrm{IP\_CRLC\_EFF\_VOL\_TEMP\_COR}$
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_CRLC_EFF_VOL_TEMP_COR	6	0...FFH	0...0.99609	3.91E-03	[-]
LDP_N_32_IP_CRLC_EFF_VOL_TEMP	6	0...FFH	0...8160	32	[rpm]
Filter constant for volumetric efficiency correction					
IP_EFF_TCO_FAC__N_32__MAF	8*8	0...FFH	0...0.99609	3.91E-03	[-]
LDPM_N_32__EFF_VOL_COR	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF__EFF_VOL_COR	8	0...FFFFH	0...1389	2.12E-02	[mg/stk]
Volumetric efficiency weighting factor versus cooling temperature					
IP_EFF_TIA_IM_FAC__N_32__MAF	8*8	0...FFH	0...0.99609	3.91E-03	[-]
LDPM_N_32__EFF_VOL_COR	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF__EFF_VOL_COR	8	0...FFFFH	0...1389	2.12E-02	[mg/stk]
Volumetric efficiency weighting factor versus intake manifold gas temperature					
C_TCYL_STND	1	0...FEH	-48...142.5	0.75	[°C]
Standard temperature in the cylinder					
C_TCO_STND	1	0...FEH	-48...142.5	0.75	[°C]
Standard coolant temperature					
C_TIA_IM_STND	1	0...FEH	-48...142.5	0.75	[°C]
Standard intake air temperature					

42.14 Ambient pressure sensor diagnosis (AMP)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_AMP	V/O	0...1H	0...1	1	[-]
AMP error present					
LV_CDN_DIAG_AMP	V/O	0...1H	0...1	1	[-]
Diagnosis condition AMP					
ERR_SYM_AMP	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom AMP					
LV_END_DIAG_AMP	V/O	0...1H	0...1	1	[-]
End of Diagnosis AMP					

Input data:

V_AMP	LV_IGK	LV_ES	LV_ST

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis shall be to detect electrical faults as defined by OBD I requirements.
The signal of the altitude pressure sensor on the A/D-input of the microcontroller is checked.

Description:

Error-symtoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Signal Line Short to Battery Voltage or Open load   (= SYM_0)
 │ │ └───> Signal Line Short to Ground                         (= SYM_1)
 │ └─────> -                                                   (= SYM_2)
 └───────> -                                                   (= SYM_3)
```

Application conditions:

Initialization: all 0 at LV_IGK 0->1 **or** reset

Recurrence: 100ms

```
Activation  If    LV_IGK = 1           and
                  LV_ES = 0            and
                  LV_ST = 0
            Then  LV_CDN_DIAG_AMP = 1
            Else  LV_CDN_DIAG_AMP = 0
            Endif
```

Formula section:

Short circuit to VBatt or open load

```
If      V_AMP  >  C_V_AMP_MAX_DIAG
Then    ERR_SYM_AMP = SYM_0
```

Short circuit to ground

```
ElseIf  V_AMP  <  C_V_AMP_MIN_DIAG
Then    ERR_SYM_AMP = SYM_1
Else    ERR_SYM_AMP = NO_SYM
Endif
```

Calculation of present error and end of diagnosis:

LV_ERR_AMP and LV_END_DIAG_AMP is calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_AMP_MAX_DIAG	1	0...3FFH	0...4.99511	4.88E-03	[V]
Maximum diagnostic value for the ambient pressure value					
C_V_AMP_MIN_DIAG	1	0...3FFH	0...4.99511	4.88E-03	[V]
Minimum diagnostic value for the ambient pressure value					
C_ABC_INC_AMP	1	0...FFH	0...255	1	[-]
Anti - bounce counter increment					
C_ABC_MAX_AMP	1	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter					

42.15 Manifold differential pressure sensor diagnosis (MAP_DIP)

42.15.1 Electrical diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_MAP_DIP_SENS	O/V	0...1H	0...1	1	[-]
MAP_DIP sensor error present					
LV_CDN_DIAG_MAP_DIP_SENS	O/V	0...1H	0...1	1	[-]
Diagnosis condition MAP_DIP sensor					
ERR_SYM_MAP_DIP_SENS	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom MAP_DIP sensor					
LV_END_DIAG_MAP_DIP_SENS	O/V	0...1H	0...1	1	[-]
End of Diagnosis MAP_DIP sensor					

Input data:

V_MAP	LV_IGK	LV_ES	LV_ST

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis shall be to detect electrical faults as defined by OBD I requirements.
The signal of the manifold differential pressure sensor on the A/D-input of the microcontroller is checked.

Description:

Error-symtoms are defined to this diagnosis function as following :

```
|_|_|_|_|
 | | | └─→ Signal Line Short to Battery Voltage or Open load    (= SYM_0)
 | | └───→ Signal Line Short to Ground                         (= SYM_1)
 | └─────→ -                                                   (= SYM_2)
 └───────→ -                                                   (= SYM_3)
```

Application conditions:

Initialization: according ABC type **STD_INI** (all 0 at LV_IGK 0→1 or at reset)

Recurrence: 100ms

```
Activation    If     LV_IGK = 1    and
                     LV_ES = 0     and
                     LV_ST = 0
              Then   LV_CDN_DIAG_MAP_DIP_SENS = 1
              Else   LV_CDN_DIAG_MAP_DIP_SENS = 0
              Endif
```

Formula section:

Short circuit to VBatt or open load

```
If        V_MAP  >  C_V_MAP_BAS_MAX_DIAG
Then      ERR_SYM_MAP_DIP_SENS  =  SYM_0
```

Short circuit to ground

```
Elself    V_MAP  <  C_V_MAP_BAS_MIN_DIAG
Then      ERR_SYM_MAP_DIP_SENS  =  SYM_1
Else      ERR_SYM_MAP_DIP_SENS  =  NO_SYM
Endif
```

Calculation of present error and end of diagnosis:

LV_ERR_MAP_DIP_SENS and LV_END_DIAG_MAP_DIP_SENS is calculated by error management.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_MAP_BAS_MAX_DIAG	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Maximum diagnostic value for the manifold differential pressure value					
C_V_MAP_BAS_MIN_DIAG	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Minimum diagnostic value for the manifold differential pressure value					
C_ABC_INC_MAP_DIP_SENS	1	0...FFH	0...255	1	[-]
Anti - bounce counter increment					
C_ABC_MAX_MAP_DIP_SENS	1	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter					

42.15.2 Plausibility diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_MAP_DIP_SHIFT	O	0...1H	0...1	1	[-]
MAP_DIP sensor error present					
LV_CDN_DIAG_MAP_DIP_SHIFT	O	0...1H	0...1	1	[-]
Diagnosis condition MAP_DIP sensor					
ERR_SYM_MAP_DIP_SHIFT	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom MAP_DIP sensor					
LV_END_DIAG_MAP_DIP_SHIFT	O	0...1H	0...1	1	[-]
End of Diagnosis MAP_DIP sensor					

Input data:

ECU_STATE	V_MAP_DIF	V_MAP_DIF_VLD	TPS_AV
C_TPS_V_MAP_AD_MIN	LV_ERR_TPS	LV_ERR_TPS_AD	LV_ERR_TPS_AD_BOL
LV_ERR_TPS_ST_CHK_2	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MAP_TPS_PLAUS	

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis shall be to detect an unplausible shift of the sensor characteristic. The deviation and the maximum allowed deviation from the nominal curve are determined by the module "Pressure Variables (MAP-Sensor) / Adaptation of sensor characteristic".

Description:

Error-symtoms are defined to this diagnosis function as following :

```
□□□□
 | | | └→ -                                       (= SYM_0)
 | | └───→ -                                      (= SYM_1)
 | └─────→ -                                      (= SYM_2)
 └───────→ signal offset out of limit             (= SYM_3)
```

Application conditions:

Initialization: according ABC type **STD** (init. from nonvolatile error memory)

Recurrence: 100ms

```
Activation    If    ECU_STATE                   = PWL    and
                    TPS_AV ≥ C_TPS_V_MAP_AD_MIN          and
                    LV_ERR_TPS                  = 0      and
                    LV_ERR_MAP_DIP_SENS         = 0      and
                    LV_ERR_TPS_AD               = 0      and
                    LV_ERR_TPS_AD_BOL           = 0      and
                    LV_ERR_TPS_ST_CHK_2         = 0      and
                    LV_ERR_LOAD_TPS_PLAUS       = 0      and
                    LV_ERR_MAP_TPS_PLAUS        = 0

              Then  LV_CDN_DIAG_MAP_DIP_SHIFT = 1
              Else  LV_CDN_DIAG_MAP_DIP_SHIFT = 0
              Endif
```

Formula section:

```
If    | V_MAP_DIF |  >  V_MAP_DIF_VLD
  then      ERR_SYM_MAP_DIP_SHIFT  =  SYM_3
  else      ERR_SYM_MAP_DIP_SHIFT  =  NO_SYM
endif
```

Calculation of present error and end of diagnosis:
LV_ERR_MAP_DIP_SHIFT and LV_END_DIAG_MAP_DIP_SHIFT is calculated by error management.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_MAP_DIP_SHIFT	1	0...FFH	0...255	1	[-]
Anti - bounce counter increment					
C_ABC_MAX_MAP_DIP_SHIFT	1	1...FFH	1...255	1	[-]
Maximum value of the anti - bounce counter					

42.16 Mass air flow sensor diagnosis (V_MAF)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_MAF_AN	V/O	0...1H	0...1	1	[-]
MAF error detected					
LV_CDN_DIAG_MAF_AN	V/O	0...1H	0...1	1	[-]
Diagnosis condition MAF diagnosis					
ERR_SYM_MAF_AN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom MAF					
LV_END_DIAG_MAF_AN	V	0...1H	0...1	1	[-]
End of Diagnosis MAF					
V_MAF_DIAG	-	0...FFH	0...4.98046	0.0195313	[V]
Filtered V_MAF value for diagnosis					
V_MAF_DIAG_L	-	0...FFH	0...4.98046	0.0195313	[V]
Calculated V_MAF value for readiness					
V_MAF_DIAG_H	-	0...FFH	0...4.98046	0.0195313	[V]
Calculated V_MAF value for readiness					
LV_MAF_READY_L	-	0...1H	0...1	1	[-]
Flag that indicates that readyness L is reached					
LV_MAF_READY_H	-	0...1H	0...1	1	[-]
Flag that indicates that readyness H is reached					

Input data:

ANG_EXC_VVL	LV_ST_END	LV_INH_DIAG_MAF	LV_VAR_MAF
LV_PUC	V_MAF	N_32	LV_VAR_TCHA
LV_IGK	TPS_AV		

FUNCTION DESCRIPTION:

General information:

Diagnosis of analog input signal in the A/D-Input from the Microprocessor. The diagnosis is only performed if the variant is learnt.

Description

To avoid the readiness before the diagnosis is able to detect a symptom the diagnosis is delayed until V_MAF_DIAG_L / H are reaching the thresholds for SCP / SCG_OC detection.
In some engine operating states the measured air flow is under the threshold for SCG / OC detection, thus the diagnosis result SCG / OC is only validated if the engine conditions can ensure a minimum air flow.

The following error symptoms can be detected:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Vbatt SCP                (= SYM_0)
 │ │ └───> Short circuit to Ground or Open load      (= SYM_1)
 │ └─────> -                                         (= SYM_2)
 └───────> -                                         (= SYM_3)
```

Application conditions:

Initialisation: According the ABC configuration **STD_INI**

(0 at LV_IGK 0->1 or reset)

[V_MAF_DIAG / _L / _H = C_V_MAF_DIAG_INI ;

LV_MAF_READY_L / H = 0

at clearing error management **or** LV_IGK 0->1 **or** reset]

Recurrence: 20ms

Activation: LV_VAR_MAF = 1 **and** LV_VAR_TCHA = 1

Deactivation:

```
If      (LV_VAR_MAF = 0 or LV_VAR_TCHA = 0)        and
        LV_ST_END = 1
Then    LV_END_DIAG_MAF_AN = 1                 // set for readiness
Endif
```

Formula section:

Calculation of V_MAF_DIAG:

```
If      LV_ST_END = 1
Then    V_MAF_DIAG = V_MAF_DIAG + ID_CRLC_V_MAF_DIAG__N_32 *
                                          (V_MAF – V_MAF_DIAG)
Else    V_MAF_DIAG = V_MAF_DIAG      //freeze
Endif
```

Calculation of delay to activate the diagnosis:

```
If      LV_MAF_READY_L = 0      or
        LV_MAF_READY_H = 0
Then
    If          LV_ST_END = 1
    Then        V_MAF_DIAG_L = V_MAF_DIAG_L + ID_CRLC_V_MAF_DIAG__N_32 *
                                          (0h – V_MAF_DIAG_L)
                V_MAF_DIAG_H = V_MAF_DIAG_H + ID_CRLC_V_MAF_DIAG__N_32 *
                                          (FFh – V_MAF_DIAG_H)
    Else        V_MAF_DIAG_H / L = V_MAF_DIAG_H / L        // freeze
```

```
        If              V_MAF_DIAG_L < C_V_MAF_MIN_DIAG
        Then            LV_MAF_READY_L = 1
        If              V_MAF_DIAG_H > C_V_MAF_MAX_DIAG
        Then            LV_MAF_READY_H = 1
        Endif
    Else        V_MAF_DIAG_H / L = V_MAF_DIAG_H / L        // freeze
    Endif
```

Calculation of error symptoms:

```
    If(1)       LV_IGK              = 1         and
                LV_ST_END           = 1         and
                LV_INH_DIAG_MAF     = 0         and
                LV_MAF_READY_H = 1              and
                LV_MAF_READY_L = 1
    Then(1)     LV_CDN_DIAG_MAF_AN = 1
        If(2)           V_MAF_DIAG > C_V_MAF_MAX_DIAG
        Then(2)         ERR_SYM_MAF_AN = SYM_0
        Else(2) If(3)   V_MAF_DIAG < C_V_MAF_MIN_DIAG
                Then(3)
                        If(4)       LV_PUC = 0                          and
                                    N_32 > C_N_MIN_MAF_DIAG             and
                                    TPS_AV > C_TPS_MIN_MAF_DIAG         and
                                    ANG_EXC_VVL > C_ANG_EXC_VVL_MIN_MAF_DIAG
                        Then(4)     ERR_SYM_MAF_AN = SYM_1
                        Else(4)     LV_CDN_DIAG_MAF_AN = 0
                        Endif(4)
                Else(3)  ERR_SYM_MAF_AN = NO_SYM
                Endif(3)
        Endif(2)
    Else(1)     LV_CDN_DIAG_MAF_AN = 0
    Endif(1)
```

Remark:

LV_END_DIAG_MAF_AN and LV_ERR_MAF_AN are calculated by error management.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MIN_MAF_DIAG	1	0...FFH	0...8160	32	[rpm]
Engine speed condition					
C_TPS_MIN_MAF_DIAG	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Throttle position threshold for detection of mass air flow signal line shorted to ground or broken					
C_V_MAF_MIN_DIAG	1	0...FFH	0...4.98046	0.0195313	[V]
Threshold value V_MAF to detect Short circuit in signal wire to earth or wire break					
C_V_MAF_MAX_DIAG	1	0...FFH	0...4.98046	0.0195313	[V]
Threshold value V_MAF to detect Short circuit in signal wire to VB					
C_V_MAF_DIAG_INI	1	0...FFH	0...4.98046	0.0195313	[V]
Initial value for V_MAF_DIAG calculation					
C_ANG_EXC_VVL_MIN_MAF_DIAG	1	0...238EH	0...200	0.0219732	[°]
Threshold value ANG_EXC_VVL to detect Short circuit in signal wire to earth or wire break					
C_ABC_INC_MAF_AN	1	0...FFH	0...255	1	[-]
Anti-bounce increment value for analogue MAF sensor					
C_ABC_MAX_MAF_AN	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value for analogue MAF sensor					
ID_CRLC_V_MAF_DIAG__N_32	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_N_32__CRLC_V_MAF_DIAG	4	0...FFH	0...8160	32	[rpm]
Filtering factor for MAF diagnosis					

42.17 Diagnosis of Frequential Mass Airflow Sensor

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_MAF_FRQ_RNG[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
error flag for range check of T_PER_MAF_FRQ_BAS[NC_MAF_NR]					
ERR_SYM_MAF_FRQ_RNG[NC_MAF_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
error symptoms of diagnostic instance MAF_FRQ_RNG[NC_MAF_NR]					
LV_CDN_DIAG_MAF_FRQ_RNG[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
Condition flag for diagnostic instance MAF_FRQ_RNG[NC_MAF_NR]					
LV_END_DIAG_MAF_FRQ_RNG[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
end flag for diagnostic instance MAF_FRQ_RNG[NC_MAF_NR]					
LV_ERR_MAF_FRQ_GRD[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
error flag for gradient check of T_PER_MAF_FRQ_BAS[NC_MAF_NR]					
ERR_SYM_MAF_FRQ_GRD[NC_MAF_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
error symptoms of diagnostic instance MAF_FRQ_GRD[NC_MAF_NR]					
LV_CDN_DIAG_MAF_FRQ_GRD[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
Condition flag for diagnostic instance MAF_FRQ_GRD[NC_MAF_NR]					
LV_END_DIAG_MAF_FRQ_GRD[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
end flag for diagnostic instance MAF_FRQ_GRD_j					
LV_ERR_MAF	V/O	0...1H	0...1	1	[-]
MAF sensor error detected					
ERR_SYM_MAF	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom MAF					
LV_CDN_DIAG_MAF	V/O	0...1H	0...1	1	[-]
Diagnosis condition MAF diagnosis					
LV_END_DIAG_MAF	V/O	0...1H	0...1	1	[-]
End of Diagnosis MAF					
LV_ERR_MAF_FRQ_EL[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
error flag for MAF electrical diagnosis					
ERR_SYM_MAF_FRQ_EL[NC_MAF_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
error symptoms of diagnostic instance MAF_FRQ_EL[NC_MAF_NR]					
LV_CDN_DIAG_MAF_FRQ_EL[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
Condition flag for diagnostic instance MAF_FRQ_EL[NC_MAF_NR]					
LV_END_DIAG_MAF_FRQ_EL[NC_MAF_NR]	V/O	0...1H	0...1	1	[-]
end flag for diagnostic instance MAF_FRQ_EL[NC_MAF_NR]					
CTR_TCC_T_PER_MAF_FRQ_EL[NC_MAF_NR]	V/O	0...FFH	0...255	1	[-]
TCC counter for diagnositic instance MAF_FRQ_EL[NC_MAF_NR]					

Input data:

LV_IGK	LV_CDN_VB_OBD1	LV_ST_END	CTR_MAF_FRQ_KGH_SUM[NC_MAF_NR]
CTR_T_PER_MAF_FRQ_RNG[NC_MAF_NR]	CTR_T_PER_MAF_FRQ_GRD[NC_MAF_NR]	LV_VAR_MAF	NC_MAF_NR
LV_CTR_T_PER_MAF_FRQ_SENS[NC_MAF_NR]	LV_CTR_T_PER_MAF_FRQ_EL[NC_MAF_NR]	N	FAC_MAF_REL
LV_CDN_DIAG_MAF_AN	LV_ERR_MAF_AN		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_RATIO_T_PER_MAF_FRQ_RNG_DIAG	1	0...7FFFH	0...99.99694	3.0518e-3	[%]
maximum acceptable percentage of out of range T_PER_MAF_FRQ_SUM[NC_MAF_NR] per ms					
C_RATIO_T_PER_MAF_FRQ_GRD_DIAG	1	0...7FFFH	0...99.99694	3.0518e-3	[%]
maximum acceptable percentage of exceeding gradients for T_PER_MAF_FRQ_SUM_j per ms					
C_ABC_INC_MAF_FRQ_RNG[NC_MAF_NR]	1	0...FFH	0...255	1	[-]
anti bounce counter increment for diagnosis instance MAF_FRQ_RNG[NC_MAF_NR]					
C_ABC_MAX_MAF_FRQ_RNG[NC_MAF_NR]	1	1...FFH	1...255	1	[-]
anti bounce counter maximum value for diagnosis instance MAF_FRQ_RNG[NC_MAF_NR]					
C_ABC_INC_MAF_FRQ_GRD[NC_MAF_NR]	1	0...FFH	0...255	1	[-]
anti bounce counter increment for diagnosis instance MAF_FRQ_GRD_j					
C_ABC_MAX_MAF_FRQ_GRD[NC_MAF_NR]	1	1...FFH	1...255	1	[-]
anti bounce counter maximum value for diagnosis instance MAF_FRQ_GRD_j					
C_ABC_INC_MAF_FRQ_EL[NC_MAF_NR]	1	0...FFH	0...255	1	[-]
anti bounce counter increment for diagnosis instance MAF_FRQ_EL[NC_MAF_NR]					
C_ABC_MAX_MAF_FRQ_EL[NC_MAF_NR]	1	1...FFH	1...255	1	[-]
anti bounce counter maximum value for diagnosis instance MAF_FRQ_EL[NC_MAF_NR]					
C_N_THD_MAF_FRQ_RNG_MIN	1	0...1FE0H	0...8160	1	[rpm]
min. N threshold for enable the MAF_FRQ_RNG diagnosis					
C_FAC_MAF_REL_THD_MAF_RNG_MIN	1	0...BB8H	0...300	0.1	[%]
min. relative cylinder filling for enable the MAF_FRQ_RNG diagnosis					
C_CTR_TCC_T_PER_MAF_FRQ_EL	1	0...FFH	0...255	1	[-]
threshold for TCC counter for diagnositic instance MAF_FRQ_EL[NC_MAF_NR]					
C_ABC_INC_MAF	1	0...FFH	0...255	1	[-]
Anti-bounce increment value					
C_ABC_MAX_MAF	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value					

FUNCTION DESCRIPTION:

The whole electrical diagnostic feature for frequential MAF sensor consists of four parts:

The error detection itself is done within the signal acquisition. Every 1 ms the mean period time per ms of mass air flow sensor (T_PER_MAF_FRQ_BAS[NC_MAF_NR]) is evaluated. Whenever an error is detected (per ms) the corresponding counter (CTR_T_PER_MAF_FRQ_RNG[NC_MAF_NR] or CTR_T_PER_MAF_FRQ_GRD[NC_MAF_NR]) gets incremented and this value itself is neglected.

Here in the diagnostic function these summed up error counters as well as the summed up counter for valid values per segment (CTR_MAF_FRQ_KGH_SUM[NC_MAF_NR]) are evaluated at the end of each segment.

42.17.1 Frequential MAF Sensor: Period Range Check per ms

General information:

The diagnosis described in this section is performed for each MAF sensor separately [NC_MAF_NR].

The purpose is to check the range of the mean period time per ms of mass air flow sensor (T_PER_MAF_FRQ_BAS[NC_MAF_NR]). The diagnostic instance consists of the following symptom:

- SYM_0: ‘MAF-sensor-[NC_MAF_NR]: mean period time per ms exceeding limit’

  MAF_FRQ_RNG[NC_MAF_NR]

Description:

Activation

```
If     LV_VAR_MAF = 1
then if     LV_IGK = 1                                   and
            LV_CDN_VB_OBD1 = 1                           and
            LV_ST_END = 1                                and
            N > C_N_THD_MAF_FRQ_RNG_MIN                  and
            FAC_MAF_REL > C_FAC_MAF_REL_THD_MAF_RNG_MIN
    Then    LV_CDN_DIAG_MAF_FRQ_RNG[NC_MAF_NR] = 1
    Else    LV_CDN_DIAG_MAF_FRQ_RNG[NC_MAF_NR] = 0
    endif
else        LV_END_DIAG_MAF_FRQ_RNG[NC_MAF_NR] = 1
            LV_CDN_DIAG_MAF_FRQ_RNG[NC_MAF_NR] = 0
Endif
```

Initialization: STD_INI

Recurrence: every segment

Formula section:

```
If     LV_CDN_DIAG_MAF_FRQ_RNG[NC_MAF_NR] = 1

then if
            [CTR_T_PER_MAF_FRQ_RNG[NC_MAF_NR] /
            (CTR_MAF_FRQ_KGH_SUM[NC_MAF_NR] +
            CTR_T_PER_MAF_FRQ_RNG[NC_MAF_NR] +
            CTR_T_PER_MAF_FRQ_GRD[NC_MAF_NR])]
                  > C_RATIO_T_PER_MAF_FRQ_RNG_DIAG          AND
```

```
            LV_CTR_T_PER_MAF_FRQ_SENS[NC_MAF_NR] = 0    AND
            LV_CTR_T_PER_MAF_FRQ_EL[NC_MAF_NR] = 0
    then    ERR_SYM_MAF_FRQ_RNG[NC_MAF_NR] = SYM_0
    else    ERR_SYM_MAF_FRQ_RNG[NC_MAF_NR] = NO_SYM
    endif
endif
```

Calculation end of diagnosis

LV_END_DIAG_MAF_FRQ_RNG[NC_MAF_NR] is calculated by error management if diagnosis is active.

42.17.2 Frequential MAF Sensor: Gradient Check per ms

General information:

The diagnosis described in this section is performed for each MAF sensor separately (j = 0 ... NC_MAF_NR – 1).

Purpose is to check the gradient of consecutively measured mean period times per ms (absolute value) of mass air flow sensor (T_PER_MAF_FRQ_BAS[NC_MAF_NR]). This allows to detect a completely unplausible increase or decrease of it caused by a loose contact. The diagnostic instance consists of the following error symptom:

- SYM_0: ‘MAF-sensor-[NC_MAF_NR]: unplausible gradient of mean period time per ms’

MAF_FRQ_GRD[NC_MAF_NR]

Description:

```
Activation  If      LV_VAR_MAF = 1
            then    if      LV_IGK = 1                  and
                            LV_CDN_VB_OBD1 = 1          and
                            LV_ST_END = 1
                    Then    LV_CDN_DIAG_MAF_FRQ_GRD[NC_MAF_NR] = 1
                    Else    LV_CDN_DIAG_MAF_FRQ_GRD[NC_MAF_NR] = 0
                    endif
            else    LV_END_DIAG_MAF_FRQ_GRD[NC_MAF_NR] = 1
                    LV_CDN_DIAG_MAF_FRQ_GRD[NC_MAF_NR] = 0
            Endif
```

Initialization: STD_INI

Recurrence: every segment

Formula section:

```
If    LV_CDN_DIAG_MAF_FRQ_GRD[NC_MAF_NR] = 1

then if
      [CTR_T_PER_MAF_FRQ_GRD[NC_MAF_NR] /
 (CTR_MAF_FRQ_KGH_SUM[NC_MAF_NR] + CTR_T_PER_MAF_FRQ_RNG[NC_MAF_NR] +
      CTR_T_PER_MAF_FRQ_GRD[NC_MAF_NR])]
      > C_RATIO_T_PER_MAF_FRQ_GRD_DIAG                    AND
      LV_CTR_T_PER_MAF_FRQ_SENS[NC_MAF_NR] = 0            AND
      LV_CTR_T_PER_MAF_FRQ_EL[NC_MAF_NR] = 0
      then   ERR_SYM_MAF_FRQ_GRD[NC_MAF_NR] = SYM_0
      else   ERR_SYM_MAF_FRQ_GRD[NC_MAF_NR] = NO_SYM
      endif
endif
```

Calculation end of diagnosis

LV_END_DIAG_MAF_FRQ_GRD[NC_MAF_NR] is calculated by error management if diagnosis is active.

42.17.3 MAF electrical diagnosis

General information:

The diagnosis described in this section is performed for each MAF sensor separately (j = 0 ... NC_MAF_NR – 1).

Purpose is to recognize a problem due to an electrical error (SCG, SCB of open line) of the mass air flow sensor (within the sensor on the wiring harness) or a self diagnosis problem is recognized. The diagnostic instance consists of the following error symptom:

- SYM_0: ‘MAF-sensor-[NC_MAF_NR]: electrical fault
- SYM_1: 'MAF-sensor-[NC_MAF_NR]: sensor error recognized

```
Activation   If      LV_VAR_MAF = 1
             then    if     LV_IGK = 1               and
                            LV_CDN_VB_OBD1 = 1       and
                            LV_ST_END = 1
                     Then   LV_CDN_DIAG_MAF_FRQ_EL[NC_MAF_NR] = 1
                     Else   LV_CDN_DIAG_MAF_FRQ_EL[NC_MAF_NR] = 0
                     endif
             else    LV_END_DIAG_MAF_FRQ_EL[NC_MAF_NR] = 1
```

```
                    LV_CDN_DIAG_MAF_FRQ_EL[NC_MAF_NR] = 0
          Endif
```

Initialization: STD_INI

Recurrence: every segment

Formula section:

```
If      LV_CDN_DIAG_MAF_FRQ_EL[NC_MAF_NR] = 1
then if     LV_CTR_T_PER_MAF_FRQ_EL[NC_MAF_NR] = 1
    then    CTR_TCC_T_PER_MAF_FRQ_EL[NC_MAF_NR] ++
    Else    CTR_TCC_T_PER_MAF_FRQ_EL[NC_MAF_NR] = 0
    Endif
    If      CTR_TCC_T_PER_MAF_FRQ_EL[NC_MAF_NR] >
            C_CTR_TCC_T_PER_MAF_FRQ_EL
    then    ERR_SYM_MAF_FRQ_EL[NC_MAF_NR] = SYM_0
    else if         LV_CTR_T_PER_MAF_FRQ_SENS[NC_MAF_NR] = 1
            then    ERR_SYM_MAF_FRQ_EL[NC_MAF_NR] = SYM_1
            else    ERR_SYM_MAF_FRQ_EL[NC_MAF_NR] = NO_SYM
            endif
    endif
endif
```

Calculation end of diagnosis

LV_END_DIAG_MAF_FRQ_EL[NC_MAF_NR] is calculated by error management if diagnosis is active.

42.17.4 Global error instance

Activation

```
If      LV_VAR_MAF = 1
then    If      LV_ST_END = 1               and
                LV_IGK = 1                  and
                LV_CDN_VB_OBD1 = 1
        then    LV_CDN_DIAG_MAF = 1
        else    LV_CDN_DIAG_MAF = 0
        endif
else    LV_END_DIAG_MAF = 1
        LV_CDN_DIAG_MAF = 0
endif
```

Initialization: STD_INI

Recurrence: every segment

Formula section:

```
If      LV_CDN_DIAG_MAF = 1          or
        LV_CDN_DIAG_MAF_AN = 1
Then    If    LV_ERR_MAF_FRQ_RNG[NC_MAF_NR] = 1
        Then  ERR_SYM_MAF = SYM_0
              LV_ERR_MAF = 1
        Else  If    LV_ERR_MAF_FRQ_GRD[NC_MAF_NR] = 1
              Then  ERR_SYM_MAF = SYM_1
                    LV_ERR_MAF = 1
              Else  If    LV_ERR_MAF_FRQ_EL[NC_MAF_NR] = 1
                    Then  ERR_SYM_MAF = SYM_2
                          LV_ERR_MAF = 1
                    Else  If    LV_ERR_MAF_AN = 1
                          Then  ERR_SYM_MAF = SYM_3
                                LV_ERR_MAF = 1
                          Else  ERR_SYM_MAF = NO_SYM
                                LV_ERR_MAF = 0
                          Endif
                    Endif
              Endif
        Endif
Endif
```

Calculation end of diagnosis

LV_END_DIAG_MAF is calculated by error management if diagnosis is active.

42.18 Mass air flow diagnosis (Applic. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MAF_BLS_DIAG	O/V	0...1H	0...1	1	-
MAF error and brake is active					
LV_INH_DIAG_MAF	V/O	0...1H	0...1	1	[-]
Inhibition of MAF diagnosis					

Input data:

LV_ERR_MAF	LV_BRAKE_DET	LV_ERR_BLS_PLAUS	LV_VAR_MAF

General information:

LV_MAF_BLS_DIAG is used by engine speed limit coordination and torque request at clutch.

Application conditions:

Initialisation: 0 at reset

Recurrence: 10ms

Activation: LV_VAR_MAF = 1

Deactivation: LV_VAR_MAF = 0

Formula section:

```
LV_INH_DIAG_MAF = 0

If      LV_ERR_MAF = 1              and
        [ LV_BRAKE_DET = 1          or
        LV_ERR_BLS_PLAUS = 1]
Then    LV_MAF_BLS_DIAG = 1
Else    LV_MAF_BLS_DIAG = 0
Endif
```

42.18.1 Interface for Rate – Based – Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_MAF	V/O	0...7H	0...7	1	[-]
Interface of MAF monitor with the Rate-Based Monitoring statistics Bit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1) Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1) Bit 2: individual RBM conditions of the monitor were encountered within this DC (bit 2 = 1)					

Input data:

LV_DC	LV_END_DIAG_MAF		

FUNCTION DESCRIPTION:

General information:

With this module the interface between the MAF monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_MAF data.

Within STATE_RBM_MAF, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for MAF diagnosis)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_MAF = 0

on failure memory reset :

bit 1 of STATE_RBM_MAF = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

Bit 0:

```
If    bit 0 of STATE_RBM_MAF = 0
then
      if      LV_END_DIAG_MAF = 1
      then    bit 0 of STATE_RBM_MAF = 1
      endif
endif
```

Bit 1:

bit 1 of STATE_RBM_MAF = 0

Bit 2:

bit 2 of STATE_RBM_MAF = 1

42.19 Ambient pressure plausibility diagnosis (LV_ERR_AMP_PLAUS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_AMP_PLAUS	V/O	0...1H	0...1	1	[-]
Boolean for detected error Ambient pressure plausibility diagnosis					
LV_CDN_DIAG_AMP_PLAUS	V	0...1H	0...1	1	[-]
Status of diagnosis flag for Ambient pressure plausibility					
ERR_SYM_AMP_PLAUS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom Ambient pressure plausibility					
LV_END_DIAG_AMP_PLAUS	V	0...1H	0...1	1	[-]
End of diagnosis flag for Ambient pressure plausibility diagnosis					
AMP_MMV_PLAUS	V	0...FFFFH	0...5434	0.0829175	[hPa]
Mean moving value of AMP_MES					

Input data:

LV_IGK	AMP_MES	LV_INH_DIAG_AMP_PLAUS	VS

FUNCTION DESCRIPTION:

General information:

Due to CARB requirements the diagnosis detects unplausible ambient pressures which cannot be detected by electrical diagnosis.

The diagnosis is based gradient monitoring of AMP_MES.

Description:

AMP_MMV_PLAUS is the mean-moving value of AMP_MES and represents the maximum possible change of AMP if the car is driving uphill or downhill.

If AMP_MES is below or upper AMP_MMV_PLAUS -/+ calibratable offset C_AMP_PLAUS_DIF_ERR the sensor is detected as not plausible and ERR_SYM_AMP_PLAUS is set.

Application conditions:

Initialisation: according ABC – type "**MEM**" *(all 0 at LV_IGK 0->1, error is set irreversible)*

AMP_MMV_PLAUS = AMP_MES at LV_IGK 0->1 **or** Reset

Recurrence: 1s

Activation: LV_IGK = 1 **and** LV_INH_DIAG_AMP_PLAUS = 0

Deactivation: LV_IGK = 0 **or** LV_INH_DIAG_AMP_PLAUS = 1
(set LV_CDN_DIAG_AMP_PLAUS = 0 at deactivation)

Formula section:

Continuous gradient monitoring:

```
LV_CDN_DIAG_AMP_PLAUS = 1
AMP_MMV_PLAUS(n) = AMP_MMV_PLAUS(n-1)
                   + IP_CLRC_AMP_PLAUS * (AMP_MES – AMP_MMV_PLAUS(n-1))
If      AMP_MES > AMP_MMV_PLAUS + C_AMP_PLAUS_DIF_ERR
Then    ERR_SYM_AMP_PLAUS = SYM_0
Elseif  AMP_MES < AMP_MMV_PLAUS - C_AMP_PLAUS_DIF_ERR
Then    ERR_SYM_AMP_PLAUS = SYM_1
Else    ERR_SYM_AMP_PLAUS = NO_SYM
Endif
```

LV_ERR_AMP_PLAUS, LV_END_DIAG_AMP_PLAUS is calculated by error management.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_AMP_PLAUS	1	0...FFH	0...255	1	[-]
Anti bounce increment AMP_PLAUS					
C_ABC_MAX_AMP_PLAUS	1	1...FFH	1...255	1	[-]
Anti bounce maximum AMP_PLAUS					
C_AMP_PLAUS_DIF_ERR	1	0...FFFFH	0...5434	0.0829175	[hPa]
Maximum allowed offset to detect a unplausible AMP_MES					
IP_CLRC_AMP_PLAUS	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_VS_IP_CLRC_AMP_PLAUS	4	0...FFH	0...255	1	[km/h]
Filter constant for mean moving calculation of AMP_MES					

42.20 Application incidences for ambient plausibility diagnosis

42.20.1 Inhibition of diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_AMP_PLAUS	V/O	0...1H	0...1	1	[-]
Inhibition of AMP_PLAUS diagnosis					

Input data:

LV_ERR_AMP	LV_ERR_VS	LV_IGK	

Description:

LV_INH_DIAG_AMP_PLAUS is set to 1 if AMP_PLAUS diagnosis is inhibited due to a present OBD1 error.

Application conditions:

Initialisation: 0 at reset

Recurrence: 1s

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

```
If      LV_ERR_VS = 1     or
        LV_ERR_AMP = 1
Then    LV_INH_DIAG_AMP_PLAUS = 1
Else    LV_INH_DIAG_AMP_PLAUS = 0
Endif
```

42.20.2 Interface for Rate-based-monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_AMP_PLAUS	V/O	0...7H	0...7	1	[-]
Interface of AMP_PLAUS monitor with the Rate-Based Monitoring statistics					

Input data:

LV_DC	CTR_ERR_DYN_NR	LV_END_DIAG_AMP_PLAUS	LV_INH_DIAG_AMP_PLAUS

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

General information:

With this module the interface between the AMP_PLAUS plausibility monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_AMP_PLAUS data.

Within STATE_RBM_AMP_PLAUS, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)
  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)
  (not valid for AMP_PLAUS diagnosis)

Application conditions:

Initialisation :

at ECU reset :

bit 0, bit 1 and bit 2 of STATE_RBM_AMP_PLAUS = 0

at LV_DC 0 → 1 transition :

bit 0 and bit 1of STATE_RBM_AMP_PLAUS = 0

bit 2 of STATE_RBM_AMP_PLAUS = 1

on failure memory reset :

bit 1 of STATE_RBM_AMP_PLAUS = 0

Recurrence: 1 s

Activation: LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_AMP	LV_ERR_VS		

```
If(1)       { CPU optimization at LV_DC 0 → 1 transition }
            CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
            While bit 1 of STATE_RBM_AMP_PLAUS = 0 do
                  with each XX failure of the above list :
                  ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                  SYNCHRONIZATION<CALL>)
                  If(2)         XX has a pending status
                  Then(2)
                                bit 1 of STATE_RBM_AMP_PLAUS = 1
                  Endif(2)
            Endwhile
Else(1)
            { the dynamic failure memory is empty }
            No action
Endif(1)
```

Every 1 s :

```
If   bit 0 of STATE_RBM_AMP_PLAUS = 0
Then
          If     LV_END_DIAG_AMP_PLAUS = 1
          Then   bit 0 of STATE_RBM_AMP_PLAUS = 1
          Endif
Endif

If   bit 1 of STATE_RBM_AMP_PLAUS = 0
Then
          If     LV_INH_DIAG_AMP_PLAUS = 1
          Then   bit 1 of STATE_RBM_AMP_PLAUS = 1
          Endif
Endif
```

42.21 Customer adaptation module : AGGR INSY

42.21.1 Outputs for BMW functions which are defined as INSY exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Ftbr	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
Filtered volumetric efficiency correction factor due to temperature effects					
Mshfm_ist	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
measured mass air flow					
Mshfm2_ist	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Massenstrom vom HFM Bank 2					
Mslui	V/O	0...FFFFH	0...29.12667	0.4444e-3	[kg]
measured mass air flow integral					
Pvagr	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
egr pressur					

Input data:

EFF_VOL_TEMP_COR_MMV	MAF_KGH_MES	MAF_KGH	PRS_EX_EGR
MAF_KGH_MES_TCHA[NC_MAF_SENS_CONF]	NC_MAF_SENS_CONF		

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: at reset: Ftbr = 1.0 , all others =0
Recurrence : Ftbr : 100 ms

Mshfm_ist, Mshfm2_ist: segment

Mslui : 200 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

Ftbr = EFF_VOL_TEMP_COR_MMV

If LC_MAF_TCHA_MES = 0

Then

Mshfm_ist = MAF_KGH_MES

Mshfm2_ist = 0

Else

Mshfm_ist = MAF_KGH_MES_TCHA[0] * C_FAC_MAF_TCHA_0
+ MAF_KGH_MES_TCHA[1] * C_FAC_MAF_TCHA_1

Mshfm2_ist = MAF_KGH_MES_TCHA[1]

Endif

Mslui = Mslui_{n-1} + MAF_KGH * **Error!** after reaching the maximum value 29.13 kg the calculation of Mslui is stopped.

Pvagr = PRS_EX_EGR

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_MAF_TCHA_MES	1	0...1H	0...1	1	[-]
Switch for calculation of Mshfm_ist/m by model (0) or by sensor (1), only for N54					
C_FAC_MAF_TCHA	1	0...FFFFH	0...3.99993	0.061e-3	[-]
Factor for conversion of mass air flow from one to two banks					
C_FAC_MAF_TCHA_0	1	0...FFFFH	0...3.99993	0.061e-3	[-]
Factor for conversion of mass air flow from bank one					
C_FAC_MAF_TCHA_1	1	0...FFFFH	0...3.99993	0.061e-3	[-]
Factor for conversion of mass air flow from bank two					

42.21.2 Outputs for BMW functions which are not defined as INSY exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Dpsr	V/O	8000...7FFFH	-1280... 1279.96093	0.0390625	[hPa]
differential manifold pressure					
Dpsr_seg	V/O	8000...7FFFH	-1280... 1279.96093	0.0390625	[hPa]
differential manifold pressure, calculated segment synchronous					
Fho	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
altitude factor					
Pvdkds	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
pressure upstream throttle					
Pvdkds_t100	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
pressure upstream throttle, 100ms					
Pu	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
ambient pressure					
Ps_abs	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
Absolute Pressure of manifold					
Ps_abs_t100	V/O	0...FFFFH	0...2559.96093	0.0390625	[hPa]
Absolute Pressure of manifold, 100ms					
B_disa_akt	V/O	0...1H	0...1	1	[-]
Bedingung DISA aktiv					

Input data:

FAC_AMP	PUT_MES_BAS	PUT	LV_ES
AMP_MES	N	MAP_DIP_MES_BAS_2SEG	ECU_STATE
MAP_DIP_MES_BAS	LV_VAR_TCHA	LV_VIM_RLS	MAP_MES_BAS

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: Dpsr: 0 hPa
Dpsr_seg: 0 hPa
Ps_abs : 0 hPa
Ps_abs_t100: 0 hPa
Fho: 1.0
Pvdkds: AMP_MES
Pvdkds_t100: AMP_MES
Pu: AMP_MES
B_disa_akt: 0

Recurrence : segment: Dpsr_seg
100ms: Fho Remark: the update rate of FAC_AMP is 1000 ms only
100 ms: Pu Remark: the update rate of AMP_MES is 1000 ms only
10ms : B_disa_akt, Ps_abs_t100, Pvdkds_t100

```
if      LV_VAR_TCHA = 0
then    Pvdkds:       100 ms
else    Pvdkds:       segment
endif

if      LV_ES = 1

then    Dpsr: 100ms, Ps_abs: 10ms

elseif  Dpsr, Ps_abs: segment
```

Activation: every engine state

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

Pvdkds = PUT
Pvdkds_t100 = PUT_MES_BAS
Pu = AMP_MES
B_disa_akt = LV_VIM_RLS
Ps_abs_t100 = MAP_MES_BAS
Fho = FAC_AMP

```
if   ECU_STATE= "PWL"
then      Dpsr_seg      = 0
          Dpsr          = 0

else      Dpsr_seg      = MAP_DIP_MES_BAS
          Dpsr          = MAP_DIP_MES_BAS_2SEG
          Ps_abs        = MAP_MES_BAS
endif
```

42.21.3 Outputs for SV aggregates

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EGR_RATIO	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Air mass flow at EGR valve / air mass flow to cylinder					
MAF_EGR	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Air mass flow at EGR valve					
MAF_KGH_FG_PRED	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Predicted fresh gas out of the manifold					
MAF_MDL_MV	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Model air mass flow mean value [kg/h]					
MAF_CYL	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Air mass flow to cylinder [kg/h]					
MAF_CYL_STK	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Air mass flow to cylinder [mg/stk]					
MAF_SCAV_EXT	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Delta RF					
MAP	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
Manifold pressure					
PQ	V/O	0...FFFFH	0...0.99998	0.0153e-3	[-]
Pressure quotient at the throttle					
PQ_SP	V/O	0...FFFFH	0...0.99998	0.0153e-3	[-]
PQ-setpoint					
MAF	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
calculated mass air flow					
FAC_MAF_REL	V/O	0...BB8H	0...300	0.1	[%]
relative filling					
MAF_MAX_COR	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Maximum available fresh air going into the cylinder					
ISAPWM	O	0...FFFFH	0...99.99847	1.5259e-3	[%]
idle speed actuator duty cycle, linearised					
ISAPWM_MMV	O	0...FFFFH	0...99.99847	1.5259e-3	[%]
idle speed actuator duty cycle moving mean value					
ISAPWM_ISA	O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Idle speed actuator duty cycle					
MAF_SP	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
MAF setpoint output for inverse air path					
MAP_SP	V/O	0...FFFFH	0...5434	0.0829175	[hPa]
mainfold air pressure setpoint					
PQ_EGR_SP	V/O	0...FFFFH	0...0.99998	0.0153e-3	[-]
pressure quotient between exhaused branch and MAP					
MAP_DRV1	V/O	8000...7FFFH	-82.91752... 82.91499	2.5304e-3	[hPa/ms]
First derivative of intake manifold pressure					

MAF_THR	V/O	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Air mass flow at the throttle					
MAF_EGR_NEUT_GAS	V/O	0...FFFFH	0...1389	0.0211948	[mg/stk]
Neutral Gas component of Exhaust Gas Recirculation					
FAC_MAF_AD_THR_KWP	V/O	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
Testerwert NN DK-Adaption					
FAC_MAF_AD_EGR_KWP	V/O	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Testerwert NN PR-Adaption					
LV_TPS_SUB_DIAG_NOT_VLD	V/O	0...1H	0...1	1	[-]
Bedingung Berechnung DKWinkel aus Signal des HF-Sensors ungedrosselt					
LV_VIM_INH_CUS	V/O	0...1H	0...1	1	[-]
Enable bit for VIM					
LV_VIM_1_CUS	V/O	0...1H	0...1	1	[-]
Setpoint disa flap 1 specified by customer					
LV_VIM_2_CUS	V/O	0...1H	0...1	1	[-]
Request from BMW for open variable intake manifold 2					
PRS_IM_CTL_I	O/V	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Saugrohrdruckregler I-Anteil					

Input data:

Agr_rate	Msagr	Msdk	Msdk_f
Mszyl_ges	Ps		Mszyl
Rf	Rf_vl	Pspvdk	Rf_soll
Pspvdk_soll	Pssol	PRS_EX	PRS_EX
	Rf_ig_ist	Mrnn_test_dk	Prnn_test
NC_MAF_FAC_CYL	B_disa_stopp	B_disa1_auf	B_disa2_auf
LV_ES	B_wdkdiag_ugd	Drf_spuel	Psreg_i

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset: 0, except PQ_EGR_SP= 0.99998, PRS_IM_CTL_I=1

Recurrence:

```
if LV_ES = 1 then   10ms
Endif

If LV_ES= 0
Then
10ms:  PQ_SP, LV_TPS_SUB_DIAG_NOT_VLD,
                MAF_EGR_NEUT_GAS,
       FAC_MAF_AD_THR_KWP,  FAC_MAF_AD_EGR_KWP,
       LV_VIM_INH_CUS,   LV_VIM_1_CUS, LV_VIM_2_CUS ,
       MAF_SCAV_EXT, PRS_IM_CTL_I

segment: other
       (Hint: the update rate of the input data can be slower )
  endif
```

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

EGR_RATIO = Agr_rate

MAF_EGR = Msagr

MAF_EGR_NEUT_GAS = Rf_ig_ist * C_MAF_REF

MAF_KGH_FG_PRED = Msdk

MAF_MDL_MV = Msdk_f

MAF_CYL = Mszyl

MAF_CYL_STK = Mszyl_ges * NC_MAF_FAC_CYL / N

MAP = Ps

PQ = Pspvdk

PQ_SP = Pspvdk_soll

MAF = Rf * C_MAF_REF

MAF_SP = Rf_soll * C_MAF_REF

FAC_MAF_REL = Rf

MAF_MAX_COR = Rf_vl * C_MAF_REF

ISAPWM = 0 (init., never change)

ISAPWM_MMV = 0 (init., never change)

ISAPWM_ISA = 0 (init., never change)

MAP_SP = Pssol

MAP_DRV1 = $(MAP_N – MAP_{N - 1})$ / T_SEG_AV recurrence: segment

PQ_EGR_SP = MAP_SP / PRS_EX

MAF_THR = Msdk

LV_TPS_SUB_DIAG_NOT_VLD = B_wdkdiag_ugd

FAC_MAF_AD_THR_KWP = Mrnn_test_dk

FAC_ MAF_AD_EGR_KWP = Prnn_test

LV_VIM_INH_CUS = B_disa_stopp

LV_VIM_1_CUS = B_disa1_auf

LV_VIM_2_CUS = B_disa2_auf

MAF_SCAV_EXT = Drf_spuel * C_MAF_REF

PRS_IM_CTL_I = Psreg_i

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MAF_REF	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Reference air mass flow for the calculation of MAF and MAF_MAX_COR					

43 Intersystem communication

Table of Contents

Index

43.1 CAN messages BN2000

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TQ_DCC_INC_REQ	V/O	0...1H	0...1	1	[-]
LV indicating torque increse required by DCC					
LV_DCC_LIH_CAN	V/O	0...1H	0...1	1	[-]
LV indicating DCC intervention limp home					
LV_DCC_OFF_ACK	V/O	0...1H	0...1	1	[-]
Acknowledge bit for switch-off DCC					
LV_DCC_PUC_INH	V/O	0...1H	0...1	1	[-]
inhibitation of PUC required by DCC					
TCC_CAN_TQ_DCC	V/O	0...EH	0...14	1	[-]
free running test cycle counter TQ_DCC (alive counter)					
CKS_CAN_TQ_DCC	V/O	0...FFH	0...255	1	[-]
Checksum TQ_DCC from CAN					
TQ_DCC_FAST_BN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
Torque target (fast) for DCC via CAN interface					
TQ_DCC_SLOW_BN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
Torque target (slow) for DCC via CAN interface					
STATE_DCC_INTV	V/O	0H 1H 2H 3H	NO_INTERVENTION TORQUE_INCREASE INVALID_SIGNAL INVALID_SIGNAL	1	[-]
status torque target DCC via CAN interface					
STATE_DCC_PUC_INH	V/O	0H 1H 3H	NO_REQUEST INHIBIT_PUC INVALID_SIGNAL	1	[-]
status request PUC-inhibitation via CAN interface - DCC					
STATE_DCC_OFF_REQ	V/O	0H 1H 2H 3H	RESERVED SYS_OFF SYS_OK INVALID_SIGNAL	1	[-]
status switch-off DCC					
STATE_DCC	V/O	0H 1H 2H 3H 4H 5H 6H 7H	INACTIVE STAND_BY NORMAL RESERVE BRAKE_ACTIVE DRIVER_ACTIVE RESERVE INVALID_SIGNAL	1	[-]
Status DCC					
STATE_DCC_CTL	V/O	0...7H	0...7	1	[-]
Status DCC control					
TCC_CAN_ARS	V/O	0...EH	0...14	1	[-]
free running test cycle counter ARS (alive counter)					

TQ_LOSS_ARS_AV_CAN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
torque loss actual value ARS from CAN					
TQ_LOSS_ARS_SP_CAN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
torque loss setpoint ARS from CAN					
STATE_ARS_CAN	V/O	0H 1H 3H	INACTIVE ACTIVE INVALID_SIGNAL	1	[-]
status anti roll stabilization via CAN					
LV_ERR_TQ_LOSS_ARS_AV_CAN	V/O	0...1H	0...1	1	[-]
interface bit for TQ_LOSS_ARS calculation in case of error via CAN					
LV_ERR_TQ_LOSS_ARS_SP_CAN	V/O	0...1H	0...1	1	[-]
interface bit for TQ_LOSS_ARS calculation in case of error via CAN					
TCC_CAN_TQ_PSTE_2	V/O	0...EH	0...14	1	[-]
free running test cycle counter TQ_PSTE_2 (alive counter)					
CKS_CAN_TQ_PSTE_2	V/O	0...FFH	0...255	1	[-]
Checksum TQ_PSTE_2 from CAN					
STATE_PSTE_2_INTV	V/O	0H 1H 2H 3H	NO_INTV TQ_INC TQ_DEC INVALID_SIGNAL	1	[-]
status torque target PSTE_2 via CAN interface					
TQ_LOSS_PSTE_2_AV_CAN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
torque loss actual value PSTE_2 from CAN					
TQ_LOSS_PSTE_2_SP_CAN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
torque loss setpoint PSTE_2 from CAN					
LV_PSTE_2_ERR	V/O	0...1H	0...1	1	[-]
LV indicating AFS error (sent by AFS via CAN)					
VEH_KEY_NR	V/O	0...FH	0...15	1	[-]
vehicle key number					
LV_RLY_ST_CAN	V/O	0...1H	0...1	1	[-]
status starter relay					
LV_KEY_AUX	V/O	0...1H	0...1	1	[-]
status Kl. R					
STATE_IGK_CAN	V/O	0H 1H 2H	IGK_OFF IGK_ON INVALID_SIGNAL	1	[-]
status ignition key via CAN					
STATE_IGK_HW	V/O	0...3H	0...3	1	[-]
status ignition key (HW) via CAN					
TCC_CAN_CAS	V/O	0...EH	0...14	1	[-]
free running test cycle counter CAS (alive counter)					
CKS_CAN_CAS	V/O	0...FH	0...15	1	[-]
check sum CAS from CAN					
CUR_SC_MAX_CAN	V/O/S	0...FFH	0...2550	10	[A]
Maximum short-current (vehicle-specific)					
CUR_SC_MIN_CAN	V/O/S	0...FFH	0...2550	10	[A]
Minimum short-current (vehicle-specific)					
CUR_WKU_CAN	V/O/S	0...FFH	0...0.255	0.001	[A]
Quiescent-current (vehicle-specific)					
IDX_BAT_CAN	V/O/S	0...FFH	0...255	1	[-]
implemented battery					

STATE_POW_CLAS_VEH	V/O	0...3H	0...3	1	[-]
status vehicle power class via CAN					
STATE_VEH_MOD	V/O	0...FH	0...15	1	[-]
vehicle mode					
LV_VAR_VEH_MOD	V/O	0...1H	0...1	1	[-]
auxiliary bit for activation of BN-Diag					
TCC_CAN_VEH_MOD	V/O	0...EH	0...14	1	[-]
free running test cycle counter VEH_MOD (alive counter)					
CKS_CAN_VEH_MOD	V/O	0...FH	0...15	1	[-]
check sum VEH_MOD from CAN					
TQ_TCS_FAST_BN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
torque target (fast) due to TCS via CAN					
TQ_TCS_SLOW_BN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
torque target (slow) due to TCS via CAN					
LV_PLAUS_ASR_CTL	V/O	0...1H	0...1	1	[-]
logical variable for ASR intervention is active					
LV_PLAUS_ESP_CTL	V/O	0...1H	0...1	1	[-]
logical variable for DSC intervention is active					
LV_TQ_MSR_REQ	V/O	0...1H	0...1	1	[-]
logical variable MSR intervention required					
LV_TQ_ASR_REQ	V/O	0...1H	0...1	1	[-]
logical variable ASR intervention required					
LV_TCS_LIH_CAN	V/O	0...1H	0...1	1	[-]
logical variable traction control system limp home					
STATE_TCS_CAN	V/O	0H 1H 2H 4H 6H 7H	VALID PASSIVE DEFECT TRACTION_MODE UNDERVOLTAGE INVALID_SIGNAL	1	[-]
status traction control system via CAN					
STATE_TCS_INTV	V/O	0H 1H 2H 3H	NO_INTV MSR_INTV ASR_INTV INTERFACE_LOCKED	1	[-]
status traction control system via CAN					
STATE_TCS_CTL	V/O	0H 1H 2H 4H 8H 10H 20H 40H FEH FFH	NO_CONTROL ABS_CONTROL ASR_CONTROL TCS_CONTROL HBA_CONTROL MSR_CONTROL EBV_CONTROL DYNO_ACTIVE MULTI_CONTROL SIGNAL_INVALID	1	[-]
status closed-loop-control via CAN					

STATE_TCS_DECE	V/O	0H 1H 15H 2H 3H	NO_INTV DCC_INTV_OK INVALID_SIGNAL EMF_INTV_OK HBA_INTV_OK	1	[-]
status deceleration (actual value) via CAN					
STATE_BRAKE_PRS	V/O	0...3H	0...3	1	[-]
status brake pressure via CAN					
BRAKE_PRS	V/O	0...FFH	0...255	1	[bar]
Brake pressure					
CKS_CAN_TQ_TCS	V/O	0...FFH	0...255	1	[-]
check sum traction control system TQ_TCS from CAN					
TCC_CAN_TQ_TCS	V/O	0...EH	0...14	1	[-]
free running test cycle counter TQ_TCS (alive counter)					
WHEEL	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed signal from CAN for cruise control					
VS_CAN	V/O	0...FFH	0...255	1	[km/h]
Vehicle speed signal from CAN					
AC_VEH_LGT_TCS	V/O	8010...7FF0H	-51.175...51.175	0.0015625	[m/s²]
Vehicle acceleration longitudinal					
AC_VEH_TRV_TCS	V/O	8010...7FF0H	-51.175...51.175	0.0015625	[m/s²]
Vehicle acceleration transversal					
TQ_GS_FAST_BN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
torque target (fast) for GS via CAN interface					
TQ_GS_SLOW_BN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
torque target (slow) for GS via CAN interface					
TQ_CONV_CAN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
converter torque from ETCU					
GEAR_INFO	O/V	0...FH	0...15	1	[-]
gear information from EGS-ECU					
STATE_CC	V/O	0H 1H 2H 3H	OPEN CONTROLLED CLOSED NOT_DEFINED	1	[-]
State of converter clutch (CC)					
CKS_CAN_TQ_ETCU	V/O	0...FFH	0...255	1	[-]
check sum TQ_ETCU from CAN					
TCC_CAN_TQ_ETCU	V/O	0...EH	0...14	1	[-]
free running test cycle counter TQ_ETCU (alive counter)					
STATE_ETCU_OBD	V/O	0H 2H 4H 6H 8H AH CH EH FH	MIL_OFF MIL_OFF MIL_ON MIL_ON MIL_FLL MIL_FLL IDLE INI_VALUE INVALID_SIGNAL	1	[-]
status OBD gearbox					
STATE_ETCU_PROG_INFO	V/O	0...FH	0…15	1	[-]
State mode ETCU					

STATE_ETCU_OBD_ERR	V/O	0H 1H 2H	NO_OBD_ERR OR OBD_ERROR INVALID_SIGNAL	1	[-]
state OBD error					
STATE_ETCU_INTV	V/O	0H 1H 2H 3H	NO_INTV TQ_INC TQ_DEC LOCKED	1	[-]
status torque target ETCU system via CAN					
LV_ETCU_LIH_CAN	V/O	0...1H	0...1	1	[-]
logical variable ETCU gear shift intevention limp home					
LV_TQ_GS_DEC_REQ	O/V	0...1H	0...1	1	[-]
LV indicating TQ decrease requested by transmission					
LV_TQ_GS_INC_REQ	O/V	0...1H	0...1	1	[-]
LV indicating TQ increase requested by transmission					
STATE_TEMP_GB	V/O	0H 1H 2H 3H	NO_REQUEST ACTIVE_STEP_1 ACTIVE_STEP_2 INVALID_SIGNAL	1	[-]
State overtemperature gearbox (AT) - request for cooling (overheating protection) via ECT/ECF					
TEMP_GB	V/O	0...FFH	-40...215	1	[°C]
Temperature gearbox (AT) – request for cooling (overheating protection) via ECF					
STATE_ST_TQ_LIM_GS	V/O	0H 1H 3H	NO_INTV INTV INVALID_SIGNAL	1	[-]
State starting torque limitation					
STATE_MOD_GB	V/O	0H 1H 2H 3H 7H	SSC_SBC_OFF SSC_ON SBC_ON SSC_SBC_ON INVALID_SIGNAL	1	[-]
State mode gearbox : SSC (Sleeping Starter Clutch), SBC (Stand By Control)					
FAC_GB_GAIN	V/O	0...FFFFH	0...8191.875	0.125	[1/m]
gearbox-drive-train gain					
N_GB	V/O	0...FFFEH	0...8191.75	0.125	[rpm]
Input shaft speed turbine/gearbox (AT)					
TQ_AMT_FAST_BN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
Torque target (fast) for AMT via CAN interface					
TQ_AMT_SLOW_BN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
Torque target (slow) for AMT via CAN interface					
CKS_CAN_TQ_AMT	V/O	0...FFH	0...255	1	[-]
check sum TQ_AMT from CAN					
TCC_CAN_TQ_AMT	V/O	0...EH	0...14	1	[-]
free running test cycle counter TQ_AMT (alive counter)					

STATE_AMT_OBD	V/O	0H 2H 4H 6H 8H AH CH EH FH	MIL_OFF MIL_OFF MIL_ON MIL_ON MIL_FLL MIL_FLL IDLE INI_VALUE INVALID_SIGNAL	1	[-]
Status OBD gearbox					
STATE_AMT_OBD_ERR	V/O	0H 1H 2H	NO_OBD_ERROR OBD_ERROR INVALID_SIGNAL	1	[-]
Status OBD-relevant error - AMT					
STATE_AMT_INTV	V/O	0H 1H 2H 3H	NO_INTV TQ_INC TQ_DEC LOCKED	1	[-]
status torque target AMT via CAN					
STATE_AMT	V/O	0H 1H 2H 3H 7H	NO_INTERV TORQUE_DEC N_REGUL TORQUE_INC INVALID_SIGNAL	1	[-]
Status shift-process AMT via CAN					
STATE_ST_TQ_LIM_AMT	V/O	0H 1H 3H	NO_INTV INTV INVALID_SIGNAL	1	[-]
State starting torque limitation AMT					
STATE_CLU_AMT	V/O	0H 1H 2H 3H 7H	CLU_OPEN CREEPING DRIVE_OFF CLU_CLOSED INVALID_SIGNAL	1	[-]
Status shift-process AMT via CAN					
LV_AMT_LIH_CAN	V/O	0...1H	0...1	1	[-]
Logical variable AMT intervention limp home					
LV_TQ_AMT_DEC_REQ	V/O	0...1H	0...1	1	[-]
Logical variable indicating torque decrease due to AMT intervention					
LV_TQ_AMT_INC_REQ	V/O	0...1H	0...1	1	[-]
Logical variable indicating torque increase due to AMT intervention					
LV_AMT_ES	V/O	0...1H	0...1	1	[-]
LV indicating engine stop via CAN					
N_MAX_AMT_CAN	V/O	0...1FE0H	0...8160	1	[rpm]
Engine speed limit AMT via CAN					
LV_GS	O/V	0...1H	0...1	1	[-]
LV indicating active gear shift (get from transmission)					
LV_PUMP_AMT_ON	V/O	0...1H	0...1	1	[-]
LV indicating status of hydraulic pump - AMT					
STATE_EXT_POW_CNS	O/V	0...C8H	0...100	0.5	[%]
State power consumption (Sonderverbraucher)					
TQ_ACCIN_CAN	V/O	0...FFH	0...127.5	0.5	[Nm]
ACC-torque from CAN					
N_ECF	V/O	0...FH	0...15	1	[-]
Setpoint step of electric cooling fan speed					
LV_REQ_TCO_L	V/O	0...1H	0...1	1	[-]
LV requirement for low coolant temperature					

LV_REQ_HEAT	V/O	0...1H	0...1	1	[-]
LV increased heater power request					
LV_ACIN	V/O	0...1H	0...1	1	[-]
logical variable ACC readiness					
LV_ACCIN	V/O	0...1H	0...1	1	[-]
logical variable ACC request					
STATE_ACIN_CAN	V/O	0...7H	0...7	1	[-]
state air-conditioning request signal CAN					
HEAT_REQ_PERC	V/O	0...C8H	0...100	0.5	[%]
heat-flow request air-conditioning					
LV_ECRAS_CTL_CAN_1	V/O	0...1H	0...1	1	[-]
ECRAS control via CAN step 1					
LV_ECRAS_CTL_CAN_2	V/O	0...1H	0...1	1	[-]
ECRAS control via CAN step 2					
TAM	V/O	0...FEH	-48...142.5	0.75	[°C]
Ambient temperature					
LV_TAM_CAN_ERR	V/O	0...1H	0...1	1	[-]
TAM-failure present					
CTR_KM_BN	O/V	0...FFFFFFH	0...16777215	1	[km]
vehicle kilometer BN reading from Kombi - for OBD					
CTR_KM_CAN	V/O/S	0...FFFFH	0...655350	10	[km]
vehicle kilometer reading from Kombi					
CTR_FTL_NOT_VLD	V	0...FFH	0...255	1	[-]
Counter for messages with invalid FTL-value					
CTR_FTL_RI_NOT_VLD	V	0...FFH	0...255	1	[-]
Counter for messages with invalid FTL_RI-value					
CTR_FTL_LE_NOT_VLD	V	0...FFH	0...255	1	[-]
Counter for messages with invalid FTL_LE-value					
LV_FTL_CAN_ERR	V/O	0...1H	0...1	1	[-]
Logical value erroneous fuel tank level signal					
LV_FTL_TOT_CAN_ERR	V/O	0...1H	0...1	1	[-]
logical value erroneous fuel tank level signal total FTL					
LV_FTL_RI_CAN_ERR	V/O	0...1H	0...1	1	[-]
Logical value erroneous fuel tank level (right) signal					
FTL	V/O	0...7FH	0...127	1	[l]
Fuel tank level					
FTL_LE	V/O	0...7FH	0...127	1	[l]
Fuel tank level left					
FTL_RI	V/O	0...7FH	0...127	1	[l]
Fuel tank level right					
LV_FTL_OBD_INH_L	V/O	0...1H	0...1	1	[-]
logical variable fuel reserve					
STATE_WAL_CAN	V	0H 1H 2H 3H	OFF YELLOW INVALID_SIGNAL INVALID_SIGNAL	1	[-]
logical variable fuel reserve					
TCC_CAN_ICL	V/O	0...EH	0...14	1	[-]
free running test cycle counter instrument cluster (alive counter)					
T_CTR_REL_CAN	V	0...FFFFFFFFH	0...4294967295	1	[s]
relative time counter (reset at Kl. 30 OFF), E65					
T_CAN	V/O	0...FFFFH	0...65535	1	[d]
day counter absolute (counts the days at sufficient mains voltage)					
VS_ICL_DISP	V/O	0...FFEH	0...409.4	0.1	[km/h]
displayed vehicle speed (speedometer)					
STATE_VS_ICL_DISP	V/O	0H 1H 3H	KM/H MPH INVALID_SIGNAL	1	[-]
status scaling of speedometer (displayed vehicle speed)					

STATE_IF_ICL_BN_MSW	V/O	0...7H	0...7	1	[-]
status interface ICL/cruise control					
idbosrst	O/V	0...FFH	0...255	1	[-]
ID2_BOS_resetting					
idfbosrt_w	O/V	0...FFFFH	0...65535	1	[-]
ID_function_BOS_resetting					
vbost	O/V	0...FFH	0...255	1	[-]
availibility_BOS_resetting					
zrbosr	O/V	0...FFH	0...255	1	[-]
counter resetting BOS resetting					
T_CLK_ICL_DISP_1	V/O	0...FFH	0...255	1	[-]
Display of date (day)					
T_CLK_ICL_DISP_2	V/O	0...FFH	0...255	1	[-]
Display of date (month)					
T_CLK_ICL_DISP_3	V/O	0...FFFFH	0...65535	1	[-]
Display of date (year)					
QOIL_DS_RST_CAN_1_5	V/O	8000...7FFFH	-327670...327680	10	[km]
Restlaufleistung_BOS_Rückstellung_1					
QOIL_DS_RST_CAN_2_1	V/O	0...FFH	0...255	1	[-]
ID2_BOS_resetting_2					
QOIL_DS_RST_CAN_2_2	V/O	0...FFFFH	0...65535	1	[-]
ID_function_BOS_resetting_2					
QOIL_DS_RST_CAN_2_3	V/O	0...FFH	0...255000	1000	[km]
Prognose_Intervall_Weg_BOS_Rückstellung_2					
QOIL_DS_RST_CAN_2_4	V/O	0...FFH	0...255	1	[-]
Zieltermin_Monat_BOS_Rückstellung_2					
QOIL_DS_RST_CAN_2_5	V/O	0...FFH	0...255	1	[-]
Zieltermin_Jahr_BOS_Rückstellung_2					
QOIL_DS_RST_CAN_2_6	V/O	0...FFH	0...255	1	[-]
Prognose_Intervall_Zeit_BOS_Rückstellung_2					
STATE_DI_PUC	V/O	1H 0H 3H	DISABLE_PUC NO_REQUEST INVALID_SIGNAL	1	[-]
state disable pull cut off					
TQ_WHEEL_LDM_BN	O/V	8000...7FFFH	-32768...32767	1	[Nm]
Torque request by ldm					
LV_TQ_WHEEL_LDM_BN_ERR	O/V	0...1H	0...1	1	[-]
LV indicating torque request by ldm invalid					
STATE_SP_DYN_WHEEL	V/O	0...FFH	0...255	1	[-]
state follow dynamic					
STATE_LDM	O/V	0...FH	0...15	1	[-]
state ldm					
STATE_LDM_INTV	O/V	0...3H	0...3	1	[-]
state ldm					
TCC_CAN_LDM	V/O	0...FH	0...15	1	[-]
free running test cycle counter TQ_DCC (alive counter)					
CKS_CAN_LDM	V/O	0...FFH	0...255	1	[-]
Checksum LDM from CAN					
STATE_CUS_LDM_CAN	V/O	0...FFFFH	0...65535	1	[-]
auxiliary state for LDM test required by customer					
LV_TQ_WHEEL_LDM_REQ	O/V	0...1H	0...1	1	[-]
LDM requested via CAN					
LV_LDM_LIH_CAN	O/V	0...1H	0...1	1	[-]
LDM LIH via CAN					
STATE_SWI_OFF_VB	V/O	0...3H	0...3	1	[-]
state switch off battery voltage (kl30)					
T_CTR_SWI_OFF_VB	V/O	0...FFH	0...255	1	[s]
shutdown counter battery-main-switch					

T_REL_CAN	O/V	0...FFFFFFFFH	0...4294967295	1	[min]
Relative time counter					
T_REL_CAN_2	O/V	0...FFFFFFFFH	0...4294967295	1	[s]
Relative time counter, resolution in seconds					
V_GEN_TAR	V/O	0...FFH	10.6...16.975	0.025	[V]
target generator voltage					
FAC_IS_INC_REQ	V/O	0...FFH	0...255	1	[-]
Request IS-increase					
ANG_PSTE	V/O	8000...7FFFH	-1439.98976...1439.94581	0.043945	[°STW]
Steering wheel sensor angle					
VEL_ANG_PSTE	V/O	8000...7FFFH	-1439.98976...1439.94581	0.043945	[°STW/s]
Steering wheel sensor velocity					
STATE_SENS_ANG_PSTE	V/O	0...7H	0...7	1	[-]
Status of steering wheel sensor					
CKS_CAN_MSW	V/O	0...FFH	0...255	1	[-]
Checksum MSW - SZL					
TCC_CAN_MSW	V/O	0...EH	0...14	1	[-]
Alive counter MSW - SZL					
STATE_BN_MSW	V/O	0H 1H 10H 2H 4H 40H 41H 42H 44H 48H 50H 7FH 8H	NO_ACTION TIP_PRE TIP_UP_DOWN PRESS_PRE TIP_POST TIP_AXIAL TIP_AXIAL TIP_AXIAL TIP_POST PRESS_POST TIP_UP_DOWN INVALID_SIGNAL PRESS_POST	1	[-]
Status MSW BN (CAN)					
STATE_GAP_MSW	-	0...3H	0...3	1	[-]
Status MSW BN (CAN)					
LV_ERR_ANG_PSTE_CAN	V/O	0...1H	0...1	1	[-]
Interface bit for TQ_LOSS_PSTE calculation in case of error via CAN					
STATE_GEAR_REV_CAN	V/O	0...1H	0...1	1	[-]
State reverse gear: 0 inactive, 1 active					
LV_LTG_HDLP_L_ON	V/O	0...1H	0...1	1	[-]
State low beam: 0 off, 1 on					
LV_LTG_INL_ON	V/O	0...1H	0...1	1	[-]
State interior lighting: 0 off, 1 on					
LV_STATE_TRL	V/O	0...1H	0...1	1	[-]
State trailer: 0 off, 1 on					
STATE_DIAG_EFP	V/O	0...FFH	0...255	1	[-]
Diagnosis state of the CAN-EFP					
LV_EFP_ON_EXT_ADJ_CAN	V/O	0...1H	0...1	1	[-]
Switch ON of EFP via CAN (tester diagnosis)					
LV_EFP_OFF_EXT_ADJ_CAN	V/O	0...1H	0...1	1	[-]
Switch OFF of EFP via CAN (tester diagnosis)					
STATE_EFP_CAN	V/O	0...3H	0...3	1	[-]
EFP - state for EOL and ASA diagnosis					
V_ENG_EFP	V/O	0...FFH	5...30.5	0.1	[V]
Engine voltage of the EFP					
CUR_ENG_EFP	V/O	0...FFH	0...25.5	0.1	[A]
Engine current of the EFP					
N_EFP_AV	V/O	0...FFH	0...12750	50	[rpm]

Engine speed of the EFP					
VB_EFP	V/O	0...FFH	5...30.5	0.1	[V]
VB measured by EFP					
STATE_EFP_CTL	O/V	0...3H	0...3	1	[-]
State which EFP-control is active					
STATE_EFP_CTL_ECU	O/V	0...FFH	0...255	1	[-]
State EFP-control CAN					
NR_VS_CRU_IF	V/O	0...FH	0...15	1	[-]
Current vehicle speed step via CAN					
VS_STEP_IF_1	V/O	0...FFEH	0...409.4	0.1	[km/h]
Vehicle speed step from CAN					
VS_STEP_IF_2	V/O	0...FFEH	0...409.4	0.1	[km/h]
Vehicle speed step from CAN					
VS_STEP_IF_3	V/O	0...FFEH	0...409.4	0.1	[km/h]
Vehicle speed step from CAN					
LV_MSG_PROG_STEP_CRU	V/O	0...1H	0...1	1	[-]
Message of "Programmierung Stufentempomat" received					
FAC_POW_MNG_VST_CNS[15]	V/O/S	0...FFH	0...255	1	[-]
Powermanagement Standverbraucher					
N_ECF_CAN	O/V	0...FFH	0...99.60937	0.390625	[%]
relative desired ECF speed from ext. ECU (AFS, EPS)					
ECF_REQ_EXT	V	0...FFH	0...255	1	[-]
Ext. ECU (AFS, EPS) requiring ECF speed					
CTL_SHIFT_LOCK_CAN	V/O	0...3H	0...3	1	[-]
Control shift-lock					
LV_LEVEL_IS	V/O	0...1H	0...1	1	[-]
Status switch level rpm					
N_CAN	-	0...7F80H	0...8160	0.25	[rpm]
engine speed (converted for BN 2000)					
PV_MAX_CAN	-	0...FEH	0...99.21875	0.390625	[%]
maximum selection pedal value (converted for BN 2000)					
PV_LDM_REQ_CAN	-	0...1000H	0...100	0.0244141	[%]
pedal value requested from LDM (converted for BN 2000)					
TCC_CAN_ECU	V	0...EH	0...14	1	[-]
free running test cycle counter ECU1 (alive counter)					
TCC_CAN_ECU1	V	0...EH	0...14	1	[-]
free running test cycle counter ECU1 (alive counter)					
TCC_CAN_ECU2	V	0...EH	0...14	1	[-]
free running test cycle counter ECU2 (alive counter)					
TCC_CAN_ECU3	V	0...EH	0...14	1	[-]
free running test cycle counter ECU3 (alive counter)					
TQ_MAF	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
actual engine torque - spare, positive					
TQ_EMS_BN	V	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
actual engine torque - driver’s choice (converted for BN)					
TOIL_CAN	-	0...FFH	-48...207	1	[°C]
Oil temperature (converted for BN 2000)					
N_MAX_THD_1_CAN	-	0...FFH	0...12750	50	[rpm]
Actual engine speed limit ¿ orange engine speed area (converted for CAN)					
N_MAX_THD_2_CAN	-	0...FEH	0...12700	50	rpm
Actual engine speed limit – red engine speed area (converted for CAN)					
AMP_CAN	-	0...FEH	598...1106	2	[hPa]
AMP (converted for CAN)					
STATE_CRU_BN	V	0...7H	0...7	1	[-]
Status of Cruise Control for BN-CAN					
LV_POW_CLAS_VEH_CAN_RCV	V/O	0...1H	0...1	1	[-]
Status vehicle power class received via CAN					

CC_ID	V/O	0...FFFFH	0...65535	1	[-]
Checkcontrol ID					
CYC_CTR_CC	O	0...FFH	0...255	1	[-]
Checkcontrol counter					
LV_CC_ID_XX	-	0...1H	0...1	1	[-]
LV indicating if the sending conditions of CC-messages are true due to present (1) or healed (0) error					
LV_SEND_CC_ID_XX	-	0...1H	0...1	1	[-]
LV indicating the CC-message which conditions for sending are fulfilled					
STATE_CC_KEY	V	0...FH	0...15	1	[-]
CC-message transmission-condition ID xx					
CKS_CLC_TQ_DCC	V/O	0...FFH	0...255	1	[-]
Check sum TQ_DCC					
CKS_CLC_CAS	V/O	0...FH	0...15	1	[-]
Check sum CAS					
CKS_CLC_TQ_TCS	V/O	0...FFH	0...255	1	[-]
Check sum TQ_TCS					
CKS_CLC_TQ_ETCU	V/O	0...FFH	0...255	1	[-]
Check sum TQ_ETCU					
CKS_CLC_TQ_AMT	V/O	0...FFH	0...255	1	[-]
Check sum TQ_AMT					
CKS_CLC_TQ_PSTE_2	V/O	0...FFH	0...255	1	[-]
Check sum TQ_PSTE_2					
CKS_CLC_VEH_MOD	V/O	0...FFH	0...255	1	[-]
Check sum VEH_MOD (CAS)					
CKS_CLC_TQ_WHEEL_1	V/O	0...FFH	0...255	1	[-]
Check sum TQ_WHEEL_1					
CKS_CLC_TQ_WHEEL_2	V/O	0...FFH	0...255	1	[-]
Check sum TQ_WHEEL_2					
CYC_CTR_CC_SAVE	O	0...FFH	0...255	1	[-]
Checkcontrol counter buffer					
CKS_CLC_MSW	V/O	0...FFH	0...255	1	[-]
Check sum CRU (SZL)					
CKS_CLC_ECU1	V/O	0...FFH	0...255	1	[-]
Check sum ECU 1					
CKS_CLC_ECU2	V/O	0...FFH	0...255	1	[-]
Check sum ECU 2					
CKS_CLC_ECU3	V/O	0...FFH	0...255	1	[-]
Check sum ECU 3					
CKS_CLC_LDM	V/O	0...FFH	0...255	1	[-]
Check sum LDM					
LV_MIL_FLL	V/O	0...1H	0...1	1	[-]
bit for MIL flash light					
CTR_TOUT_N_ECF_CAN	O/V	0...FFH	0...255	1	[-]
Counter for timeout-detection ext. ECF-request					
LV_LDM_CAN_INI	O/V	0...1H	0...1	1	[-]
Ini-values sent via CAN					
LV_MIL_CAN	V/O	0...1H	0...1	1	[-]
Boolean indicating MIL					
LV_WAL_1_CAN	V/O	0...1H	0...1	1	[-]
Boolean indicating WAL_1					
CTR_DLY_MSG_OIL	V	0...FFH	0...255	1	[-]
CTR for delay of oil-message					

LV_KEY_VLD	O/V	0...1H	0...1	1	[-]
Valid key recognized - CAS					
LV_KEY_VLD_MSG_FAILED	O/V	0...1H	0...1	1	[-]
Valid key message failed - CAS					
IMOB_RCV_RESP_CAN_x	O	…	…	.	-
CAS-response					
CTR_IMOB_RESP_RCV_CAN	O	0…FFH	0…255	1	-
CAS-response					
CKS_CAN_TQ_PSTE_3	V/O	0...FFH	0...255	1	[-]
Check sum TQ_PSTE_3					
TCC_CAN_TQ_PSTE_3	V	0...EH	0...14	1	[-]
free running test cycle counter PSTE_3 (alive counter)					
STATE_PSTE_3_SRC	V/O	0H 1H 2H 3H 4H	NORMAL_STEERING AFS EPS RESERVED INVALID_SIGNAL	1	[-]
Source of torque request due to steering					
LV_PSTE_3_INTV	V/O	0...1H	0...1	1	[-]
PSTE 3 (EHB3) torque intervention active					
TQ_LOSS_PSTE_3_SP_CAN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
setpoint torque value for steering					
TQ_LOSS_PSTE_3_AV_CAN	V/O	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
actual torque value for steering					
STATE_PSTE_3_INTV	V/O	0H 1H 2H 3H	RESERVED NO_INTERVENTION INTERVENTION_ACTIVE INVALID_SIGNAL	1	[-]
state of PSTE 3 (EHB3) torque intervention					
STATE_PSTE_3_ERR	V/O	0H 1H 2H 3H 4H 5H 6H 7H	NO_ERROR NO_SENSOR SENSOR_ERROR SENSOR_AND_MODEL_ERROR SENSOR_ERROR_TEMP SENSOR_AND_MODEL_ERROR_TEMP NOT_USED SIGNAL_INVALID	1	[-]
Error state of PSTE 3 (EHB3)					
LV_TQ_PSTE_3_CAN_ENA	V/O	0...1H	0...1	1	[-]
PSTE 3 (EHB3) torque request via CAN enabled					
LV_TQ_PSTE_3_CAN_LIH	V/O	0...1H	0...1	1	[-]
PSTE 3 (EHB3) torque request via CAN in limp home					
LV_TQ_PSTE_3_CAN_DI	V/O	0...1H	0...1	1	[-]
PSTE 3 (EHB3) torque request via CAN disabled					
CKS_CLC_TQ_PSTE_3	V/O	0...FFH	0...255	1	[-]
Check sum EHB3 (PSTE 3)					
STATE_VEH_CNS[10]	O/V	0...FFH	0...255	1	[-]

State vehicle consumer (Verbraucherstatus)					
STATE_VEH_CNS_FCT	O/V	0...3H	0...3	1	[-]
State functionality of vehicle consumer					
T_ERR_N_ENG	V	0...FFH	0...2.55	0.01	[s]
Timer for sending RPM_ENG_ERR true or false without debounced CRK error					
STATE_CC_TEST_BENCH_OBD_1	-	0...FFH	0...255	1	[-]
Bit coded readiness 1 for CC test bench					
STATE_CC_TEST_BENCH_OBD_2	-	0...FFH	0...255	1	[-]
Bit coded readiness 2 for CC test bench					
LV_CC_TEST_BENCH_ACT	V	0...1H	0...1	1	[-]
Checkcontrol test bench active					
LV_CC_ID_BENCH	-	0...1H	0...1	1	[-]
Bit indicating if the sending condition of CC test message true					
LV_SEND_CC_ID_BENCH	-	0...1H	0...1	1	[-]
Bit indicating sending of CC test message					
ipm_typ_fp	O	0...FFH	0...255	1	[-]
IPM-Fahrertyp zur Beeinflussung der Pedalauswertung					
dm_ab_fws	O	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Fahrwiderstand (Moment zur Bergkompensation)					
ipm_typ_mdkw	O	0...FFH	0...255	1	[-]
IPM-Fahrertyp zur Beeinflussung der Soll-Momentenbestimmung					
CKS_CLC_ETCU_3	V/O	0...FFH	0...255	1	[-]
Checksum ETCU_3					
TCC_CAN_ETCU_3	V/O	0...FH	0...15	1	[-]
free running test cycle counter ETCU_3 (alive counter)					
CKS_CAN_ETCU_3	V/O	0...FFH	0...255	1	[-]
check sum ETCU_3 from CAN					
TQ_MAX_ACC_CAN	V	0...FFH	0...127.5	0.5	[Nm]
air-conditioning-compressor limited torque after switch on					
STATE_ERR_SYM_TAM_CAN	V/O	0H 1H 2H 4H	NO_SYM SCP SCG OL	1	[-]
Error symptom for TAM sensor get via CAN					
STATE_ERR_SYM_FTL_LE_CAN	V/O	0H 1H 2H 4H	NO_SYM SCP SCG OL	1	[-]
Error symptom for left fuel tank level sensor get via CAN					
STATE_ERR_SYM_FTL_RI_CAN	V/O	0H 1H 2H 4H	NO_SYM SCP SCG OL	1	[-]
Error symptom for right fuel tank level sensor get via CAN					
STATE_PBR	V/O	0...FH	0...15	1	[-]
State of the parking brake (get from CAN)					
STATE_HLD_PBR	V/O	0...3H	0...3	1	[-]
State of holding vehicle with parking brake (get from CAN)					
LV_CFT_MOD_PBR	V/O	0...1H	0...1	1	[-]
Bit indicating comfort mode of parking brake (get from CAN)					
CKS_CLC_REQ_PBR	V/O	0...FFH	0...255	1	[-]
Checksum REQ_PBR					
CKS_CAN_REQ_PBR	V/O	0...FFH	0...255	1	[-]
checksum REQ_PBR from CAN					
TCC_CAN_REQ_PBR	V/O	0...FH	0...15	1	[-]
free running test cycle counter REQ_PBR (alive counter)					
CKS_CLC_PBR	V/O	0...FFH	0...255	1	[-]
Checksum PBR					
CKS_CAN_PBR	V/O	0...FFH	0...255	1	[-]
checksum PBR from CAN					
TCC_CAN_PBR	V/O	0...FH	0...15	1	[-]

free running test cycle counter PBR (alive counter)					
LV_HOOD_OPEN	V/O	0...1H	0...1	1	[-]
Engine hood open					
LV_DRIV_DOOR_OPEN	V/O	0...1H	0...1	1	[-]
Driver door open					
LV_STATE_ERR_CAN_RCV	-	0...1H	0...1	1	[-]
OBD sensor diagnosis message received cyclic					
LV_STATE_ERR_CAN_ACT	-	0...1H	0...1	1	[-]
OBD sensor diagnosis message received at least once					
LV_STATE_ERR_CAN_TOUT	V/O	0...1H	0...1	1	[-]
Timeout detected of service message OBD diagnosis					
TCC_CAN_GS_IDC	V	0...EH	0...14	1	[-]
free running test cycle counter					
CTR_STATE_ERR_CAN	-	0...FFH	0...255	1	[-]
Timeout counter for service message OBD diagnosis					
LV_ERR_VS_CAN	O/V	0...1H	0...1	1	[-]
Logical variable for VS_CAN error					
LV_FTL_LE_CAN_ERR	O/V	0...1H	0...1	1	[-]
Logical value erroneous fuel tank level (left) signal					
TQI_MAF	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Indicated engine torque after charging interventions (converted for CAN)					
LV_CS_2	O/V	0...1H	0...1	1	[-]
Clutch switch 2 information					
STATE_EFP_CRASH_CAN	O/V	0...3H	0...3	1	[-]
Request for switch off fuel pump after crash					
N_ECF_CAN_2	V	0...FFH	0...99.60937	0.390625	[%]
relative desired ECF speed from ext. ECU (AFS, EPS)					
LV_TCS_CTL_ACT	O/V	0...1H	0...1	1	[-]
Bit indicating active chassis control functions					
LV_MIL_REQ_ETCU	O/V	0...1H	0...1	1	[-]
Bit indicating MIL request from transmission control unit					
CKS_CLC_ETCU	O/V	0...FFH	0...255	1	[-]
Checksum ETCU					
CKS_CAN_ETCU	O/V	0...FFH	0...255	1	[-]
checksum ETCU from CAN					
TCC_CAN_ETCU	O/V	0...FH	0...15	1	[-]
free running test cycle counter ETCU (alive counter)					
CKS_CLC_TQ_TCT	O/V	0...FFH	0...255	1	[-]
Checksum TQ_TCT					
CKS_CAN_TQ_TCT	O/V	0...FFH	0...255	1	[-]
checksum TQ_TCT from CAN					
TCC_CAN_TQ_TCT	O/V	0...FH	0...15	1	[-]
free running test cycle counter TQ_TCT (alive counter)					
LV_TCT_LIH_CAN	O/V	0...1H	0...1	1	[-]
logical variable TCT gear shift intevention limp home					

<table>
<tr><td>LV_TCT_ES</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">LV indicating ES requested by DKG</td></tr>
<tr><td>LV_N_SP_TCT_REQ</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">LV indicating N-regultation requested by DKG</td></tr>
<tr><td>STATE_TCT_INTV</td><td>O/V</td><td>0H
2H
3H
4H
6H
CH
DH
FH</td><td>NO_INTV
CREEP
TCT_INTV_DEC
TCT_INTV_INC
N_REGULATION
TQ_DEC
TCT_ES
INVALID</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">status torque target TCT system via CAN</td></tr>
<tr><td>TQ_TCT_CAN</td><td>O/V</td><td>8000...7FFFH</td><td>-1024...1023.96875</td><td>0.03125</td><td>[Nm]</td></tr>
<tr><td colspan="6">torque clutch for TCT via CAN interface</td></tr>
<tr><td>N_SP_TCT</td><td>O/V</td><td>0...1FE0H</td><td>0...8160</td><td>1</td><td>[rpm]</td></tr>
<tr><td colspan="6">Target N during TCT-intervention (N-regulation active)</td></tr>
<tr><td>LV_N_SP_TCT</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">LV indicating valid Target N during TCT-intervention (N-regulation active)</td></tr>
<tr><td>N_SP_IS_TCT_CAN</td><td>O/V</td><td>0...1FE0H</td><td>0...8160</td><td>1</td><td>[rpm]</td></tr>
<tr><td colspan="6">Target N_SP_IS requested during TCT-intervention</td></tr>
<tr><td>STATE_ETCU_CLU</td><td>O/V</td><td>0H
1H
2H
3H
4H
5H
6H
7H</td><td>CLU_OPEN
CREEPING
DRIVE_OFF
CLU_CLOSED
CTL_SLIP
DRIVE_DYNAMIC
UNUSED
INVALID_SIGNAL</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">State gearbox for starting-clutch</td></tr>
<tr><td>STATE_MAX_AC_ST</td><td>O/V</td><td>0H
1H
2H
3H</td><td>NO_RACE_ST
RACE_ST_ACT
RACE_ST_PREP
INVALID_SIGNAL</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Status of race start (maximum acceleration at start)</td></tr>
<tr><td>OBD_TAM</td><td>O/V</td><td>0...FFH</td><td>-40...215</td><td>1</td><td>[°C]</td></tr>
<tr><td colspan="6">Ambient air temperature SAE J1979</td></tr>
<tr><td>LV_OBD_TAM_RCV</td><td>V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">ambient temperature sensor data for OBD received</td></tr>
<tr><td>LV_T_REL_CAN_REG</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">T_REL_CAN receiving of INSTR2 message</td></tr>
<tr><td>LV_INH_GP_SUP</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Bit indicating inhibition of shift up for gearbox protection (info get from ETCU)</td></tr>
<tr><td>CKS_CLC_GS_IDC</td><td>O/V</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Checksum GS_IDC (gear shift indication)</td></tr>
<tr><td>LV_GS_IDC_LDM</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Request gear shift indication for LDM (gear shitt down)</td></tr>
<tr><td>STATE_ACK_IGK_OFF</td><td>O/V</td><td>0...3H</td><td>0...3</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Ignition off request from CAS acknowledged</td></tr>
<tr><td>STATE_DHL_CTL</td><td>O/V</td><td>0...FH</td><td>0...15</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">status of hill descent control function</td></tr>
<tr><td>TCC_CAN_DISP_ECU</td><td>V</td><td>0...EH</td><td>0...14</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">free running test cycle counter</td></tr>
<tr><td>N_GB_OUT</td><td>O/V</td><td>0...FFFFH</td><td>-32768...32767</td><td>1</td><td>[rpm]</td></tr>
<tr><td colspan="6">Output speed drive train gearbox</td></tr>
<tr><td>LV_N_DISP_DYN</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
</table>

Bit indicating reqeust to calculate dynamisation of N for display in ICL (Kombi)					
STATE_ETCU_SPT_SWI	O/V	0...3H	0...3	1	[-]
Status if gearbox has built in sport switch					
LV_ETCU_SPT_SWI	O/V	0...1H	0...1	1	[-]
Sport mode set in gearbox					
T_INH_ACK_IGK_OFF	V	0...FFH	0...2.55	0.01	[s]
Timer for recognition if ignition off acknowledge is allowed					
LV_CTOP	O/V	0...1H	0...1	1	[-]
State convertible top: 0 = closed, 1 = open					
STATE_PBR_ACT	O/V	0...FH	0...15	1	[-]
Status actuator EMF (electromechanic park brake PBR)					
STATE_PBR_ACT_QLY	O/V	0...FH	0...15	1	[-]
Qualifier Status actuator EMF (electromechanic park brake PBR)					
STATE_TQ_WHEEL_PBR_QLY	O/V	0...FH	0...15	1	[-]
Qualifier Status actuator EMF (electromechanic park brake PBR)					
TCC_CAN_TQ_PBR	V	0...EH	0...14	1	[-]
Alive counter TQ_PBR					
CKS_CAN_TQ_PBR	O/V	0...FFH	0...255	1	[-]
Checksum AC_WHEEL_PBR					
CKS_CLC_TQ_PBR	O/V	0...FFH	0...255	1	[-]
Check sum AC_WHEEL_PBR					
LV_REQ_ACK_IGK_OFF	V	0...1H	0...1	1	[-]
intermediate state for CAN message control of ACK_IGK_OFF					
LV_REQ_PWL_DMTL_SND	O/V	0...1H	0...1	1	[-]
Bit indicating power latch time for DMTL requested via CAN					
VS_MAX_LIH_TCT	O/V	0...7FFFH	0...327.67	0.01	[km/h]
Vehicle speed limitation from TCT gear box in case of limp home					
LV_CC_ID_TRA	-	0...1H	0...1	1	[-]
Bit indicating if the sending condition of CC TRA message true					
LV_SEND_CC_ID_TRA	-	0...1H	0...1	1	[-]
Bit indicating sending of CC TRA message					
AC_WHEEL_PBR	O/V	10...FFF0H	-51.175...51.175	0.0015625	[m/s²]
request deceleration wheel torque EMF electromechanic park brake (PBR)					
LV_OBD_SCAN_REQ	V	0...1H	0...1	1	[-]
Bit indicating that OBD tester is connected and has send request once					

Export actions:

ACTION_COMM_SEND_CHAL_CAN_TRIG()

Input data:

N_32	LV_ES	VS	LV_VAR_VEH
CONF_SOF_SWI	LV_ERR_CAN_BOFF	LV_ERR_LDM_INH_MON	LV_ERR_BN_EFP
TIA	DIST	LV_IGK	LV_ERR_BN_T_CLK
LV_VAR_BN_EFP	LV_ERR_BN_GEAR_REV	LV_ERR_BN_TRL	LV_ERR_BN_LTG_HDLP_L
TQ_AV	SF_TQD	LV_ERR_BLS_PLAUS	LV_IM_BTS
LV_IM_BLS	N_SP_IS	TQ_MAX_CLU	TQ_LOSS
TQI_REF_IGA_MIN_LAMB	LV_ERR_TCO	TCO	C_T_DMTL_MAX
TOIL	LV_ST	PV_MAX	LV_LDM_OFF_ECU
EFF_LAMB_BAS_COR	AMP	TQI_EMS	TQI_SP_CAN
LV_ERR_BN_TQ_ETCU	ERR_SYM_BN_TQ_ETCU	LV_ERR_AMP	LV_WUP
CTR_CDN_OBD_RBM	LV_ERR_BN_TQ_PSTE_3	EFF_IGA_BAS_COR	LV_TEMP_ENG_WARN_3
STATE_COC	LV_ERR_CRK	VFF_EFP	STATE_CUR_ENG_CNS
CUR_GEN	STATE_ALTER	st_ldstgen	POW_REL_ALTER_CLC
CONF_SWI_EFP_OUT	LV_LIH_COC_ON	N_MAX_THD_1	N_MAX_THD_2
LV_ST_END	ECU_STATE	STATE_TQ_LDM	STATE_TQ_DCC
STATE_CRU_CAN	LV_AT	LV_VAR_DCC	LV_VAR_ASR
LV_VAR_ARS	TQI_REF_MAX	LV_VAR_AMT	LV_WUP_CAN
NR_VS_CRU	TQ_MAX_ACC	LV_DCC_OFF_ECU	TQ_ASR_FAST_DEC_BN
	LV_ERR_BN_TQ_DCC	ERR_SYM_BN_TQ_DCC	LV_ERR_BN_ARS
ERR_SYM_BN_ARS	LV_TQ_MAX_ARS	LV_ERR_BN_TQ_TCS	ERR_SYM_BN_TQ_TCS
LV_KD	VS_SP_DRIV_CRU_CAN	LV_CRU_MAIN_SWI	LV_ACT_N_SP_IS_EXT_ADJ
LV_STEP_ON_ICL	LV_PROG_STEP_1	VS_STEP_CAN_1	VS_STEP_CAN_2
VS_STEP_CAN_3		LV_IM_CS_PN	STATE_DIAG_GS
LV_REQ_PWL_DMTL	TQI_DCC_FAST_INC	STATE_CRU	LV_CAN_SND_MSG_PWR_MNG_1
LV_CAN_SND_MSG_PWR_MNG_0	LV_CHK_FUC_OPEN_CAN	N_ST_POW_MOD	TQ_ECU_ETCU
PWM_VALUE_HIV_PERC	idbosmsg	idfbosmg_w	DIST_RESI_OIL
DIST_RESI_OIL_KM	STATE_OIL_AVL	zrbosmld	LV_PUC
LV_IS	POW_CTL_PRI_PEAK_RED_CFT	STATE_N_MAX_THD_LIH	N_REL_CWP
stpcos	rqpcos	POW_CTL_PRI_PEAK_RED	VB_POW_MNG
POW_CTL_PARK_CNS	selspcos	CUR_RNG_CTL	POW_CTL_PWR_CNS_1
POW_CTL_PWR_CNS_2	TQ_WHEEL	ulev	STATE_ENGG_POS
STATE_TQ_WHEEL	TQ_MAX_WHEEL	PV_AV	STATE_CUT_OFF_DT
	STATE_INI_DT	TQ_MIN_WHEEL_L	TQ_MAX_WHEEL
STATE_OIL_REQ	STATE_LOIL	LV_POW_CLAS_VEH_CAN_REQ	LV_WAL_ST
LV_POIL_SWI	LV_MIS_STATE_A	LV_ERR_QOIL_SENS	LV_CHK_FUC_OPEN_CAN
B_ccprio	LV_CRU_OFF_IRR	LV_MIL	LV_OIL_CNS_WARN_1
LV_OIL_CNS_WARN_2	LV_WAL_1	LV_TEMP_ENG_WARN_1	LV_TEMP_ENG_WARN_2
LV_WAL_ST	STATE_MIL	B_ccruhver	LV_FUC_CAN
B_ccbtnach	B_ccpmerr	LV_ERR_BLS_PLAUS	B_ccbttdfk
LV_ERR_TAM_PLAUS	QOIL_DS_CAN_2_1	QOIL_DS_CAN_2_2	QOIL_DS_CAN_2_3
QOIL_DS_CAN_2_4	QOIL_DS_CAN_2_5	QOIL_DS_CAN_2_6	LV_DC_RBM
LV_ACT_MIL_EXT_ADJ	LV_MIL_EXT_ADJ	LV_ACT_WAL_1_EXT_ADJ	LV_WAL_1_EXT_ADJ
STATE_MIL_ON_DIS_EXT_REQ	LV_INH_STST_CDN	LV_GS_UP	LV_GS_DOWN
EFPPWM_CAN	LV_ERR_BN_ETCU_2	EFF_IGA_AV_CAN	EFF_SCC_AV_CAN
EFF_LAMB_AV	LV_CRU_ACT_INH	LV_CRU_DISP_HUD	IMOB_TRM_CHAL_x

STATE_READY_OBD_1	STATE_READY_OBD_2	CTR_ERR_DYN_NR	OPM_AV
LC_ETCU_FRF	FCO	Md_rad_ksoll	OBD_TPS_1
LOAD_CLC	LV_ERR_BN_ETCU_3	Id_bosrtak	LV_ERR_STST
LV_ACCOUT_RLY	Qv_out_m	Qv_quali_m	Qv_td1
Qvc_status_3	Qvc_status_4	LV_ERR_BN_REQ_PBR	LV_ERR_BN_PBR
LV_ERR_BN_CDN_DOOR	LV_ERR_FTL_LE_CAN	LV_ERR_FTL_RI_CAN	ERR_SYM_FTL_LE_CAN
ERR_SYM_FTL_RI_CAN	LV_ERR_TOUT_ICL_7	GS_IDC_DISP	GS_IDC_GEAR
B_poel_gelb	Bostoken	B_notafu	LV_MIL_ACT_REQ
LV_VAR_ECRAS_UP	LV_VAR_ECRAS_DOWN	LV_ERR_BN_TQ_TCT	LV_VAR_ETCU_3
LV_VAR_ETCU	LV_VAR_TCT	LV_VS_RUN	ERR_SYM_BN_TQ_TCT
LV_ERR_BN_DHL_CTL	N_DISP_DYN	LV_VAR_STST	LV_STST_ST_REQ_CUS
LC_IGK_OFF_ACK	LV_ERR_VS	LV_ERR_BN_STAT_TCT	LV_VAR_PBR
LV_ERR_BN_TQ_PBR	LV_VAR_TQ_PBR	LV_VAR_TCT	LV_ERR_ACK_IGK_OFF
LV_ERR_EGY_MIN_2	LV_OBD_SCAN_REQ	ECU_STATE	T_WAKE_UP_ON
T_PWL			

General information:

The basis of this CAN communication is the specification for "BN 2000" integration revision 6.3.0 (E65) and 4.1.0 (E60). All messages are transmitted as a complete frame, that means every label which is not defined will be transmitted with its "invalid" value (not initialization value!!).
The network management is done by the standard core, so it isn´t described in that document.

The allocation of the messages to the two CAN- controller follows the BMW- Excel- Sheet “Verteilung_CAN_Kanäle_2CANController.xls” from 25-06-2003. The messages which are according to this sheet transmitted in channel 0 ... 13, Basic 1 and Basic 2 are sent via CAN-controller 1. The messages in channel 14 ... 27, Basic 3 and Basic 4 are sent via CAN-controller 2.

43.1.1 Input messages overview

message / *identifier*	**BN2000 definition**	**ECU definition**
ACC-Modul	**adaptive cruise control**	
DREHMOMENT_ANF_ACC	ST_TORQ_CLCTR_ACC	STATE_DCC_CTL
	ST_ACC_MOD_DME	STATE_DCC
	ST_ACC_SWO_SYS_DME	STATE_DCC_OFF_REQ
	CHKSM_TORQ_ACC	CKS_CAN_TQ_DCC
	ALIV_TORQ_ACC	TCC_CAN_TQ_DCC
	TORQ_TAR_ACC	TQ_DCC_FAST_BN
	TORQ_TAR_ADJR_POS_ACC	TQ_DCC_SLOW_BN
	ST_TORQ_TAR_ACC	STATE_DCC_INTV
	RQ_PREV_INFS_ACC	STATE_DCC_PUC_INH
ARS-Modul	**anti roll stabilization**	**ARS (anti roll stabilization)**
STAT_ARS	ALIV_COU_ARS	TCC_CAN_ARS
	ST_CLCTR_ARS	STATE_ARS_CAN
	TORQ_TAR_ARS	TQ_LOSS_ARS_SP_CAN
	TORQ_AVL_ARS	TQ_LOSS_ARS_AV_CAN
AFS-Modul	**active front steering**	**PSTE_2 (power steering 2)**
DREHMOMENT_ANF_AFS	ALIV_COU_AFS	TCC_CAN_TQ_PSTE_2
	CHKSM_TORQ_AFS	CKS_CAN_TQ_PSTE_2

	ST_TORQ_TAR_AFS	STATE_PSTE_2_INTV
	TORQ_TAR_AFS	TQ_LOSS_PSTE_2_SP_CAN
	TORQ_AVL_AFS	TQ_LOSS_PSTE_2_AV_CAN
	ST_RQ_TORQ_AFS_ERR	LV_PSTE_2_ERR
CAS	**car access system**	
KLEMMENSTATUS	ST_KL_R	LV_KEY_AUX
	ST_KL_15	STATE_IGK_CAN
	ST_KL_50	LV_RLY_ST_CAN
	NO_KEY	VEH_KEY_NR
	ST_STCD_PENG_MSA	LV_CS_2
	ST_KL_15_HW	STATE_IGK_HW (STATE_ACK_IGK_OFF)
	ALIV_KL	TCC_CAN_CAS
	CHKSM_KL	CKS_CAN_CAS
	ST_KEY_VLD	LV_KEY_VLD
CODIERUNG_PM	I_SC_MAX_VEH	CUR_SC_MAX_CAN
	I_SC_MIN_VEH	CUR_SC_MIN_CAN
	QC_VEH	CUR_WKU_CAN
	CLAS_BT	IDX_BAT_CAN
FAHRZEUGTYP	TYP_VEH	Not in use yet
	TYP_BODY	Not in use yet
	QUAN_CYL	Not in use yet
	QUAN_GR	Not in use yet
	TYP_ENG	Not in use yet
	TYP_STE	Not in use yet
	TYP_GRB	Not in use yet
	CLAS_PWR	STATE_POW_CLAS_VEH
	TYP_CNT	Not in use yet
	TYP_CAPA	Not in use yet
STAT_ZV_KLAPPEN	ST_DSW_DRD	LV_DRIV_DOOR_OPEN
	ST_CT_BON	LV_HOOD_OPEN
Response from CAS	See formula section	See formula section
SZM		
FAHRZEUGMODUS	MOD_VEH	STATE_VEH_MOD
	ALIV_MOD_VEH	TCC_CAN_VEH_MOD
	CHKSM_MOD_VEH	CKS_CAN_VEH_MOD
CTR_CRASH_SWO_EKP	CTR_CRASH_SWO_EKP	STATE_EFP_CRASH_CAN
DSC-Modul		**TCS (traction control system)**
DREHMOMENT_ANF_DSC	CHKSM_TORQ_DSC	CKS_CAN_TQ_TCS
	ALIV_TORQ_DSC	TCC_CAN_TQ_TCS
	TORQ_TAR_DSC	TQ_TCS_FAST_BN
	TORQ_TAR_ADJR_POS_DSC	TQ_TCS_SLOW_BN
	ST_TORQ_TAR_DSC	STATE_TCS_INTV (LV_TQ_ASR_REQ, LV_TQ_MSR_REQ, LV_TCS_LIH_CAN)
GESCHWINDIGKEIT	V_VEH	VS_CAN, WHEEL
	ACLN_VEH_LN_DSC	AC_VEH_LGT_TCS
	ACLN_VEH_ACRO_DSC	AC_VEH_TRV_TCS
GESCHWINDIGKEIT_RAD	V_WHL_FLH	not in use → CAN11H

	V_WHL_FRH	not in use → CAN11H
	V_WHL_RLH	not in use → CAN11H
	V_WHL_RRH	not in use → CAN11H
STAT_DSC	ST_CLCTR	STATE_TCS_CTL (LV_PLAUS_ASR_CTL, LV_PLAUS_ESP_CTL, LV_TCS_CTL_ACT)
	ST_DSC	STATE_TCS_CAN
	ST_DCRN_AVL	STATE_TCS_DECE
	ST_BRP	STATE_BRAKE_PRS
	BRP	BRAKE_PRS
EGS-Modul	**elektr. Getriebesteuerung**	**ETCU (electronic transmission control unit)**
DREHMOMENT_ANF_EGS	CHKSM_TORQ_EGS	CKS_CAN_TQ_ETCU
	ALIV_TORQ_EGS	TCC_CAN_TQ_ETCU
	TORQ_TAR_EGS	TQ_GS_FAST_ BN
	TORQ_TAR_ADJR_POS_EGS	TQ_GS_SLOW_ BN
	ST_TORQ_TAR_EGS	STATE_ETCU_INTV (LV_TQ_GS_INC_REQ, LV_TQ_GS_DEC_REQ, LV_ETCU_LIH_CAN)
	ST_OBD_GRB	STATE_ETCU_OBD
	ST_OBD_ERR_GRB	STATE_ETCU_OBD_ERR
	ST_LIM_STORQ	STATE_ST_TQ_LIM_GS
	TORQ_DRAW_GRB	TQ_CONV_CAN
	TEMP_GRB	TEMP_GB
	ST_OTMP_GRB	STATE_TEMP_GB
SSG-Modul	**sequentielles Schaltgetriebe**	**AMT (automated manual transmission)**
DREHMOMENT_ANF_SSG	CHKSM_TORQ_SSG	CKS_CAN_TQ_AMT
	ALIV_TORQ_SSG	TCC_CAN_TQ_AMT
	TORQ_TAR_SSG	TQ_AMT_FAST_ BN
	TORQ_TAR_ADJR_POS_SSG	TQ_AMT_SLOW_ BN
	ST_TORQ_TAR_SSG	STATE_AMT_INTV (LV_TQ_AMT_INC_REQ, LV_TQ_AMT_DEC_REQ, LV_AMT_LIH_CAN)
	ST_OBD_GRB_SSG	STATE_AMT_OBD
	ST_OBD_ERR_GRB_SSG	STATE_AMT_OBD_ERR
	ST_LIM_STORQ_SSG	STATE_ST_TQ_LIM_AMT
	ST_SHPS_SSG	STATE_AMT
	ST_CLT_SSG	STATE_CLU_AMT
	RQ_Motorstop	LV_AMT_ES
	LIM_RPM_ENG	N_MAX_AMT_CAN
	ST_GRLV_ACV	LV_GS
DKG-Modul (TCT)	**Doppelkupplungagetriebe**	**TCT (twin clutch transmission)**
DREHMOMENT_ANF_DKG	CHKSM_TORQ_DKG	CKS_CAN_TQ_TCT
	ALIV_TORQ_DKG	TCC_CAN_TQ_TCT
	TORQ_TAR_DKG	TQ_GS_FAST_BN
	TORQ_TAR_ADJR_POS_DKG	TQ_GS_SLOW_BN
	ST_MOD_DKG	STATE_TCT_INTV

		(LV_TQ_GS_INC_REQ, LV_TQ_GS_DEC_REQ, LV_TCT_LIH_CAN, LV_TCT_ES, LV_N_SP_TCT_REQ)
	TORQ_CLT_DKG	TQ_TCT_CAN
	RPM_TAR_ENG_DKG	N_SP_TCT (LV_N_SP_TCT)
ST_DKG	RQ_AUXWP	Not used
	ST_OTMP_DKG	STATE_TEMP_GB
	TEMP_DKG	TEMP_GB
Transmission units	**Getriebeinformationen**	**Transmission data**
GETRIEBEDATEN		
	ST_GR_GRB	GEAR_INFO
	GRDT_REIN	FAC_GB_GAIN
only EGS (LV_AT) ->	ST_CCLT	STATE_CC
only EGS (LV_AT) ->	ST_MOD_GRB	STATE_MOD_GB
only SSG (LV_VAR_AMT) ->	ST_HYPP_ACV	LV_PUMP_AMT_ON
only DKG or EGS (LV_VAR_TCT or LV_AT) ->	ST_GRLV_ACV	LV_GS
only DKG (LV_VAR_TCT) ->	ST_RSTA	STATE_MAX_AC_ST
GETRIEBEDATEN_2		
	RPM_GRB_TURB	N_GB
	ST_RCOG_FSHUP	LV_INH_GP_SUP
only DKG (LV_VAR_TCT) ->	RQ_BST_RPM_IDLG	N_SP_IS_TCT_CAN
only DKG (LV_VAR_TCT) ->	ST_STCLT	STATE_ETCU_CLU
	RPM_GRB_NEGL_2	N_GB_OUT
	RQ_GRB_DYNS_DISP_RPM_ENG	LV_N_DISP_DYN
	ST_GRB_SW	STATE_ETCU_SPT_SWI
GETRIEBEDATEN_3		
	TORQ_DVCH_RISE	dm_ab_fws
	IDX_ACPD	ipm_typ_fp
	IDX_TORQ_FIL	ipm_typ_mdkw
only DKG (LV_VAR_TCT) ->	RQ_MIL_GRB	LV_MIL_REQ_ETCU
	LIM_MAX_V_GRB_EMMOD	VS_MAX_LIH_TCT
ANZEIGE_GETRIEBEDATEN		
	CTR_DISP_PRG_GRB	STATE_ETCU_PROG_INFO
	CTR_DISP_SPMOD_GRB	LV_ETCU_SPT_SWI
IHKA		
WAERMESTROM_KLIMA	HTFL_RQ_AIC	HEAT_REQ_PERC
	HTFL_ST_AIC	not in use, value initialised FFh
	TORQ_TAR_ACCM	TQ_ACCIN_CAN
	CTR_EFAN_AIC	N_ECF
	CTR_AIC_SPFN	LV_REQ_HEAT , LV_REQ_TCO_L
	CTR_CLMP	LV_REQ_TCO_L
	CTR_AIC_RDI	STATE_ACIN_CAN
	ST_PWR_SPCOS	STATE_EXT_POW_CNS
	CTR_FAN_FLAP	LV_ECRAS_CTL_CAN_1/_2

Kombi		
A_TEMP_RELATIVZEIT	TEMP_EX	TAM
	T_SEC_COU_REL	T_CTR_REL_CAN (E65) bzw. T_REL_CAN (E60), T_REL_CAN_2
	T_DAY_COU_ABSL	T_CAN
BOS_RUECKSTELLUNG	ID2_BOS_RSTG	idbosrst
	ID_FN_BOS_RSTG	idfbosrt_w
	AVAI_BOS_RSTG	vbost
	COU_RSTG_BOS_RSTG	zrbosr
	RMMI_BOS_RSTG	QOIL_DS_RST_CAN_1_5
BOS_RUECKSTELLUNG_2	ID2_BOS_RSTG_2	QOIL_DS_RST_CAN_2_1
	ID_FN_BOS_RSTG_2	QOIL_DS_RST_CAN_2_2
	FRC_INTM_WAY_BOS_RSTG_2	QOIL_DS_RST_CAN_2_3
	TARD_MON_BOS_RSTG_2	QOIL_DS_RST_CAN_2_4
	TARD_YR_BOS_RSTG_2	QOIL_DS_RST_CAN_2_5
	FRC_INTM_T_BOS_RSTG_2	QOIL_DS_RST_CAN_2_6
KILOMETERSTAND	MILE_KM	CTR_KM_CAN
	FLLV_FUTA	FTL
	FLLV_FUTA_LH	FTL_LE
	FLLV_FUTA_RH	FTL_RI
	ST_FLLV_FUTA_SPAR	LV_FTL_OBD_INH_L
	RNG	not defined
STAT_KOMBI	V_VEH_SPDM	VS_ICL_DISP
	ST_UN_V_SPDM	STATE_VS_ICL_DISP
	ST_INTF_CCTRA_KI	STATE_IF_ICL_BN_MSW
	ALIV_COU_KI	TCC_CAN_ICL
	ST_WALI_ENG	STATE_WAL_CAN
UHRZEIT_DATUM	DISP_HR	Not in use yet
	DISP_MN	Not in use yet
	DISP_SEC	Not in use yet
	DISP_DATE_DAY	T_CLK_ICL_DISP_1
	DISP_DATE_WDAY	Not in use yet
	DISP_DATE_MON	T_CLK_ICL_DISP_2
	DISP_DATE_YR	T_CLK_ICL_DISP_3
	ST_DISP_CTI_DATE	Not in use yet
LDM	**Längsdynamikmodul**	
Anforderung Radmoment Antriebsstrang	Byte 7 + Byte 6	STATE_CUS_LDM_CAN
	FOLDYN_WMOM_PT_TAR	STATE_DI_PUC
	ST_DRASY_PT	STATE_LDM
	WMOM_PT_TAR	TQ_WHEEL_LDM_BN
	WMOM_PT_ENBL	STATE_LDM_INTV
	FOLDYN_WMOM_PT_TAR	STATE_SP_DYN_WHEEL
	ALIV_RQ_WMOM_PT	TCC_CAN_LDM
	CHKSM_RQ_WMOM_PT	CKS_CAN_LDM
	RQ_GSI_CLCTR_LN	LV_GS_IDC_LDM
Power-Modul		
POWERMGMT_	ST_BTSW	see **2.8 Power-Modul**

BATTERIESPANNUNG		
	ST_SWO_KL_30	STATE_SWI_OFF_VB
	SDWN_COU_BTSW	T_CTR_SWI_OFF_ VB
	T_SEC_COU_REL_ RED	T_REL_CAN (E65) T_REL_CAN_2 (E65)
POWERMGMT_ LADESPANNUNG	CTR_GEN_U_TAR	V_GEN_TAR
	RPM_IDLG_SPEC	FAC_IS_INC_REQ
	TEMP_BT	not defined
SZ-Lenksäule		
BEDIENUNG_TEMPOMAT	CHKSM_CCTR	CKS_CAN_MSW
	ALIV_CCTR	TCC_CAN_MSW
	OP_PUBU_CCTR_ACC	STATE_BN_MSW
	OP_GAPC_ACC	STATE_GAP_MSW
LENKRADWINKEL		
	STWA	ANG_PSTE
	STWA_V	VEL_ANG_PSTE
	STWA_ERR	STATE_SENS_ANG_PSTE
K_CAN: LM		
STAT_GANG_RUECKWAERTS	ST_GR_BAC	STATE_GEAR_REV_CAN
LAMPENZUSTAND	ST_DIPB	LV_LTG_HDLP_L_ON
CTR_LICHT	CTR_ITLI	LV_LTG_INL_ON
STAT_ANHAENGER	ST_TRAI	LV_STATE_TRL
STAT_VERDECK_CABRIO	ST_PO_FLDT_CAB	LV_CTOP
EKP		
STAT_EKP	ST_DIAG_FU_PP_EL	STATE_DIAG_EFP
	ST_FU_PP_EL	STATE_EFP_CAN
	U_ENG_FU_PP_EL	V_ENG_EFP
	I_ENG_FU_PP_EL	CUR_ENG_EFP
	RPM_AVL_FU_PP_EL	N_EFP_AV
	U_KL_30_FU_PP_EL	VB_EFP
	FLQUAN_TAR_FU_PP_EL	not defined yet
	SETQ_FU_PP_EL	not defined yet
	ST_FU_CLCTR_FU_PP_EL	STATE_EFP_CTL
K_CAN: CCC_GW, M_ASK		
PRGG_CCTR	IDX_NO_PRGG_CCTR	NR_VS_CRU_IF
	V_PRGG_CCTR_IDX_NO_1	VS_STEP_IF_1
	V_PRGG_CCTR_IDX_NO_2	VS_STEP_IF_2
	V_PRGG_CCTR_IDX_NO_3	VS_STEP_IF_3
K_CAN		
PM_STANDVERBRAUCHER	ID_FN_PWGM	FAC_POW_MNG_VST_CNS, formula section
	ST_PCOS_PWGM	FAC_POW_MNG_VST_CNS, formula section
VERBRAUCHERSTATUS	ID_FN_COS	STATE_VEH_CNS[10], see formula section
	ST_FN_COS	STATE_VEH_CNS_FCT, see formula section
ANFORDERUNG_ELUEFT		see formula section

ER		
OBD Sensor Diagnosestatus		see formula section
EHB3		
DREHMOMENT_ANF_STE	CHKSM_TORQ_STE	CKS_CAN_TQ_ PSTE_3
	ALIV_TORQ_STE	TCC_CAN_TQ_ PSTE_3
	SRC_RQ_TORQ_STE	STATE_PSTE_3_SCR
	TORQ_AVL_STE	TQ_LOSS_PSTE_3_AV_CAN
	TORQ_TAR_STE	TQ_LOSS_PSTE_3_SP_CAN
	ST_TORQ_TAR_STE	STATE_PSTE_3_INTV (LV_PSTE_3_INTV)
	ST_RQ_TORQ_STE_ERR	STATE_PSTE_3_ERR (LV_TQ_ PSTE_3_CAN_ENA LV_TQ_ PSTE_3_CAN_LIH LV_TQ_ PSTE_3_CAN_DI)
EMF (PBR)		
STELLANF_EMF	ST_HYD_RETA_2	STATE_HLD_PBR
	ST_CFFU_EMF_2	LV_CFT_MOD_PBR
STATUS_EMF	ST_EMF_LOCA_2	STATE_PBR
ST_RQ_EMF	ST_ACT_EMF	STATE_PBR_ACT
	QU_ST_ACT_EMF	STATE_PBR_ACT_QLY
	~~RQ_WMOM_EMF~~ RQ_DCRN_EMF	~~TQ_WHEEL_PBR~~ AC_WHEEL_PBR
	~~QU_RQ_WMOM_EMF~~ QU_RQ_DCRN	STATE_TQ_WHEEL_PBR_QLY
	ALIV_ST_RQ_EMF	TCC_CAN_TQ_PBR
	CHKSM_ST_RQ_EMF	CKS_CAN_TQ_PBR
	SAFG_RQ_WMOM_EMF	not used
DXC		
SOLL_MOM_ANF	ST_HDC_FN	STATE_DHL_CTL

43.1.2 Output messages overview

message / *identifier*	**BN2000 definition**	**ECU definition**
DME1		
ANFRAGE	ID2_INQY	see chapter 3
	ID_FN_INQY	see chapter 3
DISP_CC_MSG_EXT	ID2_CC_MESS_EXT	not defined
	NO_CC_MESS_EXT	not defined
	ST_CC_MESS_EXT	not defined
	TRANF_CC_MESS_EXT	not defined
	ST_IDC_CCLK_CC_MESS_EXT	not defined
	NO_FRM_CC_MESS_EXT	not defined
	QUAN_FRM_CC_MESS_EXT	not defined
	UTDT_CC_MESS	not defined
DISP_CC_MSG	ID2_CC_MESS_STD	01000000b
	NO_CC_MESS_STD	depending on dif. condition
	ST_CC_MESS_STD	depending on dif. condition
	TRANF_CC_MESS_STD	C_T_CYC_DISP_ICL_CAN
	ST_IDC_CCLK_CC_MESS_STD	C_ICL_DISP_FRQ

BOS_MELDUNG	ID2_BOS_MESS	idbosmsg
	ID_FN_BOS_MESS	idfbosmg_w
	RMMI_BOS	DIST_RESI_OIL
	ST_UN_BOS	DIST_RESI_OIL_KM
	AVAI_BOS_MESS	STATE_OIL_AVL
	COU_RSTG_BOS_MESS	zrbosmld
BOS_MELDUNG_2	ID2_BOS_MESS_2	QOIL_DS_CAN_2_1
	ID_FN_BOS_MESS_2	QOIL_DS_CAN_2_2
	FRC_INTM_WAY_BOS_MESS_2	QOIL_DS_CAN_2_3
	TARD_YR_BOS_MESS_2	QOIL_DS_CAN_2_4
	TARD_MON_BOS_MESS_2	QOIL_DS_CAN_2_5
	FRC_INTM_T_BOS_MESS_2	QOIL_DS_CAN_2_6
TORQUE_1	CHKSM_TORQ_1_DME	CKS_CLC_ECU1
	ALIV_TORQ_1_DME	TCC_CAN_ECU1
	TORQ_AVL	TQ_AV (converted for BN)
	ST_TORQ_AVL	SF_TQD, LV_DI_TQ_REQ_CAN_MPI_GDI
	TORQ_AVL_DMEE	TQ_ECU_ETCU
	ST_SW_CLT	LV_IM_CS_PN
	ST_DMEA_SWO	LV_DCC_OFF_ECU
	RCPT_ACC_SWO_SYS_DME	LV_DCC_OFF_ACK
	ST_RCPT_ENG_ACC	STATE_TQ_INTV_DCC
	ST_RCPT_ENG_ARS	STATE_TQ_INTV_ARS
	ST_RCPT_ENG_DSC	STATE_TQ_INTV_TCS
	ST_RCPT_ENG_EGS	STATE_TQ_INTV_ETCU, STATE_TQ_INTV_AMT, STATE_TQ_INTV_TCT
	ST_OBD_CTFN_GRB	STATE_DIAG_GS
	ST_CT_BRPD_DME	BLS-/BTS-conditions
TORQUE_2	CHKSM_TORQ_2_DME	CKS_CLC_ECU2
	ALIV_TORQ_2_DME	TCC_CAN_ECU2
	ST_SW_LEV_RPM	N_SP_IS-conditions
	ST_INFS	LV_PUC
	TORQ_AVL_MIN	TQ_LOSS (converted for BN)
	TORQ_AVL_MAX	TQ_MAX_CLU (converted for BN)
	TORQ_AVL_SPAR_NEG	TQ_REF_IGA_MIN_LAMB (converted for BN)
	TORQ_AVL_SPAR_POS	TQ_MAF (converted for BN)
TORQUE_3	CHKSM_TORQ_3_DME	CKS_CLC_ECU3
	ALIV_TORQ_3_DME	TCC_CAN_ECU3
	TORQ_DVCH	TQ_EMS_BN
	ANG_ACPD	PV_MAX_CAN
	RPM_ENG	N_CAN
	RPM_ENG_ERR	LV_ERR_CRK
	ST_IDLG_ENG	! LV_IS
	ST_CLCTR_V	STATE_CRU_CAN, STATE_TQ_DCC, STATE_TQ_LDM

	RQAM_FU	VFF_EFP / EFPPWM_CAN
ENGINE_RPM	RPM_TEMP_DOM_1	N_MAX_THD_1_CAN
	RPM_TEMP_DOM_2	N_MAX_THD_2_CAN
ENGINE_1	TEMP_ENG	TCO (converted)
	TEMP_EOI	TOIL_CAN
	ALIV_COU_DME	TCC_CAN_ECU
	ST_ENG_RUN	LV_ES and LV_ST conditions
	ST_SW_WAUP	LV_WUP_CAN; LV_LIH_COC_ON
	AIP_ENG	AMP_CAN
	IJV_FU	FCO (like CAN11h)
	CTR_SLCK	CTL_SHIFT_LOCK_CAN
	ST_SW_CLT	LV_IM_CS_PN
	RDUC_DOCTR_RPM	STATE_N_MAX_THD_LIH
	RPM_IDLG_TAR	N_SP_IS
	ST_RBMCYC	LV_DC_RBM (see formula)
ENGINE_2	RQ_STASS	B_notafu
GESCHWINDIGKEIT_TEMPOMAT	V_CLCTR_CCTR	VS_SP_DRIV_CRU_CAN
	ST_V_CLCTR_CCTR	depending on conditions
	CTR_DISP_CCTR	LV_STEP_ON_ICL
	CTR_DISP_CCTR_HUD	LV_CRU_DISP_HUD
	RCPT_ACTVN_CCTR_DR	LV_CRU_ACT_INH
	IDX_NO	NR_VS_CRU
	V_CCTR_IDX_NO_1	VS_STEP_CAN_1
	V_CCTR_IDX_NO_2	VS_STEP_CAN_2
	V_CCTR_IDX_NO_3	VS_STEP_CAN_3
WAERMESTROM_MOTOR	HTFL_RQ_DME	FFh
	HTFL_ST_DME	N_REL_CWP
	ACCM_LIM_TORQ	TQ_MAX_ACC_CAN
	ST_CRAW_ENG	STATE_CUR_ENG_CNS
	ST_I_GEN	CUR_GEN
	ST_GEN	STATE_ALTER
	ST_LDST_GEN	st_ldstgen
	CTR_APE_WV	PWM_VALUE_HIV_PERC
	LDST_GEN	POW_REL_ALTER_CLC
POWERMANAGEMENT_BATTERIESPANNUNG *(in case of E60, LV_VAR_VEH = 0)*	T_SEC_COU_REL_RED	*invalid value: FFFFFFH*
	SDWN_COU_BTSW	*invalid value: FFH*
	ST_SWO_KL_30	*invalid value: 11b = 3H*
	ST_COS_SWO	*invalid value: 11b = 3H*
	ST_HT_RSCR	*invalid value: 11b = 3H*
	ST_PCOS	stpcos
	CTR_PRED_PRIO_CF	POW_CTL_PRI_PEAK_RED_CFT
	CTR_PRED_PRIO	POW_CTL_PRI_PEAK_RED
	ST_SW_DSTB	*invalid value: 11b = 3H*
	ST_BTSW	*invalid value: 11b = 3H*
	U_BT	VB_POW_MNG

	ACKM_RQ_SWO_KL_15	STATE_ACK_IGK_OFF
POWERMANAGEMENT_VERBRAUCHERSTEUERUNG *(in case of E60, LV_VAR_VEH = 0)*	CTR_PCOS	POW_CTL_PARK_CNS
	RQ_PCOS	rqpcos
	CTR_CBR	CUR_RNG_CTL
	CTR_PWR_COS	POW_CTL_PWR_CNS_1
	CTR_PWR_SPCOS	POW_CTL_PWR_CNS_2
	SLCTN_SPCOS	selspcos
	ULEV	ulev
Radmoment Antriebsstrang 1	ST_PENG	STATE_ENGG_POS
	ANG_ACPD_TAR	PV_LDM_REQ_CAN
	WMOM_PT_AVL	TQ_WHEEL
	ANG_ACPD_REAL	PV_AV
	ALIV_WMOM_PT_1	TCC_CAN_TQ_WHEEL_1
	CHKSM_WMOM_PT_1	CKS_CLC_TQ_WHEEL_1
Radmoment Antriebsstrang 2	WMOM_PT_MAX	TQ_MAX_WHEEL
	WMOM_PT_MIN_LOW	TQ_MIN_WHEEL_L
	WMOM_PT_MAX	TQ_MAX_WHEEL
	ST_PT_INIT	LV_LDM_OFF, LV_LDM_OFF_ECU
	ALIV_WMOM_PT_2	TCC_CAN_TQ_WHEEL_2
	CHKSM_WMOM_PT_2	CKS_CLC_TQ_WHEEL_2
Challenge to CAS	See formula section	See formula section
Status EKP	ST_FU_CLCTR_DME	STATE_EFP_CTL_ECU
OBD_DT_MOTOR	PO_THVA_ENG	OBD_TPS_1
	CALCVL_LD_ENG	LOAD_CLC
ST_BATT	CND_BT	Qv_out_m
	CND_BT_ACCY	Qv_quali_m
	CND_BT_CHNG	Qvc_status_4
	CND_BT_T	Qv_td
	ST_CND_BT	Qvc_status_3
DISP_GSI	ALIV_DISP_GSI	TCC_CAN_GS_IDC
	DISP_TARG	Spa_gang
	DISP_GSI	Spa_art
	CHKSM_ DISP_GSI	CKS_CLC_GS_IDC
DISP_ENGDAT	ALIV_DISP_ENGDAT	TCC_CAN_DISP_ECU
	DISP_RPM_ENG	N_DISP_DYN
DME2		

43.1.3 Enable conditions for BN 2000 diagnosis

General information:

The CAN diagnosis for BN input messages is only to be enabled if the variant has been recognized. In case of a modul (e.g. TCS) which is not a variant but implemented in every vehicle the diagnosis is always enabled.

If BN diagnosis for message XXX is enabled (conditions see table below) then the conditions for activating the BN diagnosis have to be checked: only if LV_CDN_DIAG_BN_XXX =1 then the BN diagnosis is to be activated. For activation conditions see Chapter "BN2000 diagnosis".

In the following the input messages, their keywords, the symptoms which have to be dectected (timeout, alive, checksum) and the enable conditions for each diagnosis are described.

The XXX in the Chapter "BN2000 diagnosis" have to be replaced by the respective keyword in the table below.

As soon as an TOUT/CKS/TCC-symptom is detected the last CAN-signals of the corresponding message are freezed until the LV_ERR_BN_xxx is debounced, no more new values are taken over. After the LV_ERR_BN_xxx is debounced the signals are set to default-values.

The diagnosis of the symptoms TOUT/CKS/TCC can be inhibited by setting the
corresponding calibration constant C_CTR_DIAG_TOUT/CKS/TCC_CAN_MAX_xxx to 0.
In this case the CAN-signals will be taken over.

Table of BN input messages

message / *identifier*	**Keyword for BN-diagnosis (XXX):**	**Enable conditions for BN-diagnosis:**	**Diagnosis of symptoms:**
ACC-Modul	**(Distance Cruise Control)**		
DREHMOMENT_ANF_ACC	**TQ_DCC**	LV_VAR_DCC = 1	TCC_CAN_**TQ_DCC**
			CKS_CAN_**TQ_DCC**
			Timeout **TQ_DCC**
ARS-Modul	**(Anti-roll-stabilisation)**		
STAT_ARS	**ARS**	LV_VAR_ARS = 1	TCC_CAN_**ARS**
			Timeout **ARS**
AFS-Modul	**(Active Front Steering)**		
DREHMOMENT_ANF_ACC	**TQ_PSTE_2**	LV_VAR_PSTE_2 = 1	TCC_CAN_**TQ_PSTE_2**
			CKS_CAN_**TQ_PSTE_2**
			Timeout **TQ_PSTE_2**
EHB3-Modul	**(Electric Power Steering)**		

DREHMOMENT_ANF_STE	**TQ_PSTE_3**	LV_VAR_PSTE_3 = 1	TCC_CAN_**TQ_PSTE_3**
			CKS_CAN_**TQ_PSTE_3**
			Timeout **TQ_PSTE_3**
CAS	**(Car-access-system)**		
KLEMMENSTATUS	**CAS**	always enabled	TCC_CAN_**CAS**
			CKS_CAN_**CAS**
			Timeout **CAS**
FAHRZEUGMODUS	**VEH_MOD**	LV_VAR_VEH_MOD = 1	TCC_CAN_**VEH_MOD**
			CKS_CAN_**VEH_MOD**
			Timeout **VEH_MOD**
STAT_ZV_KLAPPEN	**CDN_DOOR**	always enabled	Timeout **CDN_DOOR**
DSC-Modul	**(Traction-control-system)**		
DREHMOMENT_ANF_DSC	**TQ_TCS**	always enabled	TCC_CAN_**TQ_TCS**
			CKS_CAN_**TQ_TCS**
			Timeout **TQ_TCS**
GESCHWINDIGKEIT	**VS_TCS**	always enabled	Timeout **VS_TCS**
GESCHWINDIGKEIT_RAD		**not in use**	**-------------------------**
STAT_DSC	**TCS**	always enabled	Timeout **TCS**
EGS-Modul			
DREHMOMENT_ANF_EGS	**TQ_ETCU**	LV_AT = 1	TCC_CAN_**TQ_ETCU**
			CKS_CAN_**TQ_ECTU**
			Timeout **TQ_ETCU**
SSG-Modul			
DREHMOMENT_ANF_SSG	**TQ_AMT**	LV_VAR_AMT = 1	TCC_CAN_**TQ_AMT**
			CKS_CAN_**TQ_AMT**
			Timeout **TQ_AMT**
DKG-Modul			
DREHMOMENT_ANF_DKG	**TQ_TCT**	LV_VAR_TCT = 1	TCC_CAN_**TQ_TCT**
			CKS_CAN_**TQ_TCT**
			Timeout **TQ_TCT**
ST_DKG	**STAT_TCT**	LV_VAR_TCT = 1	Timeout **STAT_TCT**
Transmission units			
GETRIEBEDATEN	**ETCU**	LV_VAR_ETCU = 1	TCC_CAN_**ETCU**
			CKS_CAN_**ETCU**
			Timeout **ETCU**
GETRIEBEDATEN_2	**ETCU_2**	LV_AT = 1 **or** LV_VAR_TCT = 1	Timeout **ETCU_2**
GETRIEBEDATEN_3	**ETCU_3**	LV_VAR_ETCU_3	TCC_CAN_**ETCU_3**
			CKS_CAN_**ECTU_3**
			Timeout **ETCU_3**
ANZEIGE_GETRIEBEDATEN	**ETCU_DISP**	LV_AT = 1 **or** LV_VAR_TCT = 1	Timeout **ETCU_DISP**
IHKA			
WAERMESTROM_KLIMA	**ACC**	LV_VAR_ACIN = 1	Timeout **ACC**
Kombi			

A_TEMP_RELATIVZEIT	**T_ICL**	LV_VAR_ICL = 1	Timeout **T_ICL**
BOS_RUECKSTELLUNG			
KILOMETERSTAND	**KM_ICL**	LV_VAR_ICL = 1	Timeout **KM_ICL**
STAT_KOMBI	**ICL**	LV_VAR_ICL = 1	TCC_CAN_**ICL**
			Timeout **ICL**
UHRZEIT_DATUM	**T_CLK**	LV_VAR_ICL = 1	Timeout **T_CLK**
Power-Modul			
POWERMGMT_ BATTERIESPANNUNG	**POW_VB**	LV_VAR_VEH = 1	Timeout **POW_VB**
POWERMGMT_ LADESPANNUNG	**POW_GEN**	LV_VAR_VEH = 1	Timeout **POW_GEN**
SZ-Lenksäule			
BEDIENUNG_TEMPOMAT	**MSW**	LV_VAR_BN_MSW = 1	TCC_CAN_**MSW**
			CKS_CAN_**MSW**
			Timeout **MSW**
LENKRADWINKEL	**ANG_PSTE**	LV_VAR_PSTE = 1	Timeout **ANG_PSTE**
SZM			
Steuerung Crashabschaltung EKP	EFP_CRASH	LV_VAR_EFP_CRASH = 1	Timeout **EFP_CRASH**
EKP			
STAT_EKP	**EFP**	LV_VAR_BN_EFP = 1	Timeout **EFP**
K_CAN: LM			
STAT_GANG_RUECKWAERTS	**GEAR_REV**	LV_VAR_BN_GEAR_REV = 1	Timeout **GEAR_REV**
LAMPENZUSTAND	**LTG_HDLP_L**	LV_VAR_BN_LTG_HDLP_L = 1	Timeout **LTG_HDLP_L**
STAT_ANHAENGER	**TRL**	LV_VAR_BN_TRL = 1	Timeout **TRL**
LDM			
Anforderung Radmoment Antriebsstrang	**LDM**	LV_VAR_BN_LDM = 1	TCC_CAN_**LDM**
			CKS_CAN_**LDM**
			Timeout **LDM**
EMF-Modul	**(Electric mechanical park brake)**		
STELLANF_EMF	**REQ_PBR**	LV_VAR_PBR = 1	TCC_CAN_**REQ_PBR**
			CKS_CAN_**REQ_PBR**
			Timeout **REQ_PBR**
STATUS_EMF	**PBR**	LV_VAR_PBR = 1	TCC_CAN_**PBR**
			CKS_CAN_**PBR**
			Timeout **PBR**
ST_RQ_EMF	**TQ_PBR**	LV_VAR_TQ_PBR = 1	Timeout **TQ_PBR** TCC_CAN_**TQ_PBR** CKS_CAN_**TQ_PBR**
DXC-Modul	**(4 wheel drive ECU)**		
SOLL_MOM_ANF	**DHL_CTL**	LV_VAR_4WD = 1	Timeout DHL_CTL

43.1.4 Input messages

General information:

The handling of default-values in case of a CAN-error is as follows: if one of the symptoms (timeout, alive or checksum) is detected the last received value should be freezed until the error is debounced. After debounce the respective default-value should be used until rebounce.

43.1.4.1 ACC-Modul

Formula section:

DREHMOMENT_ANF_ACC:

```
IF          LV_ERR_BN_TQ_DCC       = 0              AND
            LV_ERR_CAN_BOFF        = 0
THEN        // checksum_torque_ACC:
CKS_CAN_TQ_DCC      =       CHKSM_TORQ_ACC

// alive_torque_ACC:
TCC_CAN_TQ_DCC      =       ALIV_TORQ_ACC

// Torque_target _ACC:
TQ_DCC_FAST_BN      =       TORQ_TAR_ACC


// Torque_target_adjustment-range_posistive _ACC:
TQ_DCC_SLOW_BN      =       TORQ_TAR_ADJR_POS_ACC

// state ACC-intervention:
if          ST_TORQ_TAR_ACC        = 00H
then                STATE_DCC_INTV          = NO_INTERVENTION
            LV_TQ_DCC_INC_REQ      = 0
elseif              ST_TORQ_TAR_ACC         = 01H
        then        STATE_DCC_INTV          = TORQUE_INCREASE
                            LV_TQ_DCC_INC_REQ       = 1
                else        LV_TQ_DCC_INC_REQ       = 0
                endif
elseif              ST_TORQ_TAR_ACC         = 02H           or
                    ST_TORQ_TAR_DCC         = 03H
        then        STATE_DCC_INTV          = INVALID_SIGNAL
                            LV_DCC_LIH_CAN          = 1
                else        LV_DCC_LIH_CAN          = 0
endif

// request_prevention_over-run-fuel-shutoff:
IF          STATE_DCC_PUC_INH      =        01H
THEN                LV_DCC_PUC_INH          =       1
ELSE                LV_DCC_PUC_INH          =       0
ENDIF

STATE_DCC_PUC_INH = RQ_PREV_INFS_ACC
```

```
// status_ACC_switch-off_system_DME:
IF        STATE_DCC_OFF_REQ    =    01H
THEN          LV_DCC_OFF_ACK        =    1
ELSE          LV_DCC_OFF_ACK        =    0
ENDIF

STATE_DCC_OFF_REQ = ST_ACC_SWO_SYS_DME


// status_ACC_mode_DME:
IF        ST_ACC_MOD_DME       =    00H
THEN          STATE_DCC             =    INACTIV
ELSEIF    ST_ACC_MOD_DME       =    01H
THEN          STATE_DCC             =    STAND_BY
ELSEIF    ST_ACC_MOD_DME       =    02H
THEN          STATE_DCC             =    NORMAL
ELSEIF    ST_ACC_MOD_DME       =    03H
THEN          STATE_DCC             =    RESERVE
ELSEIF    ST_ACC_MOD_DME       =    04H
THEN          STATE_DCC             =    BRAKE_ACTIVE
ELSEIF    ST_ACC_MOD_DME       =    05H
THEN          STATE_DCC             =    DRIVER_ACTIVE
ELSEIF    ST_ACC_MOD_DME       =    06H
THEN          STATE_DCC             =    RESERVE
ELSEIF    ST_ACC_MOD_DME       =    07H
THEN          STATE_DCC             =    INVALID_SIGNAL
ENDIF

// status_torque_control_ACC:
STATE_DCC_CTL                 =  ST_TORQ_CLCTR_ACC

ELSE      // checksum_torque_ACC:
CKS_CAN_TQ_DCC      =      FFH


// alive_torque_ACC:
TCC_CAN_TQ_DCC      =      FH


// Torque_target _ACC:
TQ_DCC_FAST_BN      =      8000H


// Torque_target_adjustment-range_posistive _ACC:
TQ_DCC_SLOW_BN      =      8000H

// state ACC-intervention:
STATE_DCC_INTV                =      3H
LV_TQ_DCC_INC_REQ   =      0
LV_DCC_LIH_CAN                =      1
```

```
// request_prevention_over-run-fuel-shutoff:
STATE_DCC_PUC_INH = 0H
     LV_DCC_PUC_INH          =     0


// status_ACC_switch-off_system_DME:
STATE_DCC_OFF_REQ =          03H
LV_DCC_OFF_ACK               =     0


// status_ACC_mode_DME:
STATE_DCC                    =     0H

// status_torque_control_ACC:
STATE_DCC_CTL                =     7H     // invalid signal

ENDIF
```

43.1.4.2 ARS-Modul

Formula section:

STAT_ARS

```
IF          LV_ERR_BN_ARS = 0        AND
            LV_ERR_CAN_BOFF = 0
THEN        // free running test cycle counter anti roll stabilization :
TCC_CAN_ARS   =      ALIV_COU_ARS


// status anti roll stabilization :
if    ST_CLCTR_ARS      =      00b
then        STATE_ARS_CAN    =      INACTIVE
elseif      ST_CLCTR_ARS     =      01b
then        STATE_ARS_CAN    =      ACTIVE
else        STATE_ARS_CAN    =      INVALID_SIGNAL
endif


// target value torque loss ARS :
TQ_LOSS_ARS_SP_CAN           =      TORQ_TAR_ARS (converted from BN 2000)

IF          TORQ_TAR_ARS     = FFH
THEN              LV_ERR_TQ_LOSS_ARS_SP_CAN = 1
ELSE              LV_ERR_TQ_LOSS_ARS_SP_CAN = 0
ENDIF
```

```
// actual value torque loss ARS :
TQ_LOSS_ARS_AV_CAN        =        TORQ_AVL_ARS (converted from BN 2000)

IF    TORQ_AVL_ARS     = FFH
THEN        LV_ERR_TQ_LOSS_ARS_AV_CAN = 1
ELSE        LV_ERR_TQ_LOSS_ARS_AV_CAN = 0
ENDIF
ELSE        // free running test cycle counter anti roll stabilization :
TCC_CAN_ARS   =        FH

// status anti roll stabilization :
STATE_ARS_CAN          =        3H

// target value torque loss ARS :
TQ_LOSS_ARS_SP_CAN        =        8000H
LV_ERR_TQ_LOSS_ARS_SP_CAN = 1

// actual value torque loss ARS :
TQ_LOSS_ARS_AV_CAN        =        8000H
LV_ERR_TQ_LOSS_ARS_AV_CAN = 1
ENDIF
```

43.1.4.3 AFS-Modul

Formula section:

DREHMOMENT_ANF_AFS:

```
IF          LV_ERR_BN_TQ_PSTE_2 = 0        AND
            LV_ERR_CAN_BOFF = 0
THEN        // checksum_torque_AFS:

CKS_CAN_TQ_PSTE_2  = CHKSM_TORQ_AFS

// alive_torque_AFS:

TCC_CAN_TQ_PSTE_2  = ALIV_TORQ_AFS

// target value torque loss AFS :
```

```
TQ_LOSS_PSTE_2_SP_CAN = TORQ_TAR_AFS (converted from BN 2000)


// actual value torque loss AFS :
TQ_LOSS_PSTE_2_AV_CAN = TORQ_AVL_AFS (converted from BN 2000)


// state AFS-intervention:

if     ST_TORQ_TAR_AFS          = 00H
then          STATE_PSTE_2_INTV       = NO_INTERVENTION
elseif        ST_TORQ_TAR_AFS         = 01H
then          STATE_PSTE_2_INTV       = TORQUE_INCREASE
elseif        ST_TORQ_TAR_AFS         = 02H
then          STATE_PSTE_2_INTV       = TORQUE_DECREASE
elseif        ST_TORQ_TAR_AFS         = 03H
then          STATE_PSTE_2_INTV       = INVALID_SIGNAL
endif

if     ST_RQ_TORQ_AFS_ERR   = 02H  or
       ST_RQ_TORQ_AFS_ERR   = 04H
then          LV_PSTE_2_ERR = 1
else          LV_PSTE_2_ERR = 0
endif

ELSE          // checksum_torque_AFS:

CKS_CAN_TQ_PSTE_2  = FFH


// alive_torque_AFS:

TCC_CAN_TQ_PSTE_2  = FH


// target value torque loss AFS :
TQ_LOSS_PSTE_2_SP_CAN = 8000H


// actual value torque loss AFS :
TQ_LOSS_PSTE_2_AV_CAN = 8000H



// state AFS-intervention:
STATE_PSTE_2_INTV    = 3H
LV_PSTE_2_ERR = 1
ENDIF
```

43.1.4.4 CAS (car access system)

Formula section:

Part calculated with 10ms recurrence (at every engine state):

```
If      LV_STST_ST_REQ_CUS = 1
Then    T_INH_ACK_IGK_OFF = 0 s
Else    increment T_INH_ACK_IGK_OFF    (until max reached)
Endif
```

KLEMMENSTATUS:

```
IF      LV_ERR_BN_CAS = 0                 AND
        LV_ERR_CAN_BOFF = 0
THEN    // status_klemme_R:
        if      ST_KL_R            = 01H
        then    LV_KEY_AUX         = 1
        else    LV_KEY_AUX         = 0
        endif

        // status klemme 15 :
        if      ST_KL_15           = 00b
        then    STATE_IGK_CAN      = IGK_OFF
        elseif  ST_KL_15           = 01b
        then    STATE_IGK_CAN      = IGK_ON
        else    STATE_IGK_CAN      = INVALID_SIGNAL
        endif

        //status positive engagement clutch:
        If      ST_STCD_PENG_MSA = 1
        Then    LV_CS_2 = 1
        Else    LV_CS_2 = 0
        Endif

        // status klemme 15 HW :
        If      ST_KL_15_HW = 3H
        Then    STATE_IGK_HW = 2
```

```
Else    STATE_IGK_HW        = ST_KL_15_HW
Endif

If      STATE_IGK_HW 2->1
Then    LV_REQ_ACK_IGK_OFF =1
Else    If       STATE_IGK_HW != 1
        Then     LV_REQ_ACK_IGK_OFF =0
        Else // do nothing
        Endif
Endif

// status if IGK-off request from CAS can be acknowledged:
// hint:  LC_IGK_OFF_ACK active = 1
If      LC_IGK_OFF_ACK          = 1             and
        LV_REQ_ACK_IGK_OFF = 1                  and
        VS                      = 0 km/h        and
        LV_ERR_VS               = 0             and
        ECU_STATE               = ENG_STOP      and
        LV_STST_ST_REQ_CUS = 0                  and
        LV_OBD_SCAN_REQ = 0                     and     //see NOTE below
        T_INH_ACK_IGK_OFF >= C_T_MAX_INH_ACK_IGK_OFF
Then    STATE_ACK_IGK_OFF= 1
Else    STATE_ACK_IGK_OFF= 0
Endif
```

NOTE : Bit LV_OBD_SCAN_REQ is supplied via network layer. Purpose of this part is to make sure that Kl15 will be deactivated only by pressing START/STOP button once OBD scan tool has been connected to vehicle and one of the messages in table below was sent. Switching off KL15 by power managment is not possible in this state.

Requirements for LV_OBD_SCAN_REQ :

- is initialised with "0" at reset

- after disconnecting OBD scan tool Kl15 can be switched off by pressing START/STOP button.

- is set when OBD scan tool is connected to vehicle can and one of the messages listed in table has been sent successfully once.

CAN identifier (hex)	Description
7DF	CAN identifier for functionally addressed request messages sent by the external test equipment.

CAN identifier (hex)	Description
7E0	Physical request CAN identifier from the external test equipment to ECU #1
7E1	Physical request CAN identifier from the external test equipment to ECU #2
7E2	Physical request CAN identifier from the external test equipment to ECU #3
7E3	Physical request CAN identifier from the external test equipment to ECU #4
7E4	Physical request CAN identifier from the external test equipment to ECU #5
7E5	Physical request CAN identifier from the external test equipment to ECU #6
7E6	Physical request CAN identifier from the external test equipment to ECU #7
7E7	Physical request CAN identifier from the external test equipment to ECU #8

```
// status_klemme_50:
if          ST_KL_50          = 01H
then   LV_RLY_ST_CAN    = 1
else   LV_RLY_ST_CAN    = 0
endif

// key number :
VEH_KEY_NR       =      NO_KEY

// checksum_klemme CAS :
CKS_CAN_CAS      =      CHKSM_KL

// alive_klemme CAS :
TCC_CAN_CAS      =      ALIV_KL

// status_key_valid:
LV_KEY_VLD_MSG_FAILED = 0
If(1)          ST_KEY_VLD = 01H
Then(1)        LV_KEY_VLD = 1
Else(1)        if(2)          ST_KEY_VLD        = 00H
            Then(2)  LV_KEY_VLD         = 0
            Else(2)  LV_KEY_VLD         = 0
                           LV_KEY_VLD_MSG_FAILED = 1
            Endif(2)
Endif(1)
```

```
ELSE    // checksum_klemme CAS :
        CKS_CAN_CAS      =      FH
        // alive_klemme CAS :
        TCC_CAN_CAS      =      FH
        // status klemme 15 HW :
        STATE_IGK_HW     = 2H
        // status_key_valid:
        LV_KEY_VLD       = 0
        LV_KEY_VLD_MSG_FAILED = 1
        //status positive engagement clutch:
        LV_CS_2 = 0
        // status if IGK-off request from CAS can be acknowledged:
        STATE_ACK_IGK_OFF= 0
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MAX_INH_ACK_IGK_OFF	1	0...FFH	0...2.55	0.01	[s]
threshold to allow igk off acknowledge					
C_CC_ID_TRA	1	0...FFFFH	0...65535	1	[-]
Checkcontrol ID for TRA message					
C_STATE_CC_ID_TRA	1	0...FH	0...15	1	[-]
sending-condition for CC-message					

CODIERUNG_PM

current_short-circuit_maximum_vehicle:

CUR_SC_MAX_CAN = I_SC_MAX_VEH

current_short-circuit_minimum_vehicle:

CUR_SC_MIN_CAN = I_SC_MIN_VEH

quiescent-current_vehicle:

CUR_WKU_CAN = QC_VEH

class_battery:

IDX_BAT_CAN = CLAS_BT

Fahrzeugtyp

Class_power

STATE_POW_CLAS_VEH = CLAS_PWR

meesage Fahrzeugtyp is received :

LV_POW_CLAS_VEH_CAN_RCV = 1

STAT_ZV_KLAPPEN

```
IF          LV_ERR_BN_CDN_DOOR = 0      AND
            LV_ERR_CAN_BOFF = 0
THEN
            // status_door-switch_DRD
            If      ST_DSW_DRD = 00 h
            then    LV_DRIV_DOOR_OPEN = 0
            else    LV_DRIV_DOOR_OPEN = 1
            endif
            // status_contact_bonnet
            If      ST_CT_BON = 00 h
            then    LV_HOOD_OPEN = 0
            else    LV_HOOD_OPEN = 1
            endif
ELSE
            LV_DRIV_DOOR_OPEN = 1
```

```
        LV_HOOD_OPEN = 1
ENDIF
```

Response CAS (EWS):

Frame 1:

```
ID a → see LH EWS
IMOB_RCV_RESP_CAN_x

CTR_IMOB_RESP_RCV_CAN = CTR_IMOB_RESP_RCV_CAN or 0x1
```

Frame 2:

```
ID b → see LH EWS
IMOB_RCV_RESP_CAN_x

CTR_IMOB_RESP_RCV_CAN = CTR_IMOB_RESP_RCV_CAN or 0x2
```

43.1.4.5 SZM

FAHHRZEUGMODUS

```
IF      LV_ERR_BN_VEH_MOD = 0       AND
        LV_ERR_CAN_BOFF = 0
THEN    // mode_vehicle:
STATE_VEH_MOD       =     MOD_VEH

// checksum_mode_vehicle:
CKS_CAN_VEH_MOD     =     CHKSM_MOD_VEH

// alive_mode_vehicle:
TCC_CAN_VEH_MOD     =     ALIV_MOD_VEH

// enable conditions for BN-Diag "VEH_MOD":
IF   CONF_SOF_SWI    = 1
THEN    LV_VAR_VEH_MOD = 1
ELSE    LV_VAR_VEH_MOD = 0
ENDIF
ELSE    // mode_vehicle:
STATE_VEH_MOD       =     1

// checksum_mode_vehicle:
CKS_CAN_VEH_MOD     =     FFH

// alive_mode_vehicle:
```

```
TCC_CAN_VEH_MOD    =       FH

ENDIF


CTR_CRASH_SWO_EKP
control_crash_switch-off_EKP:
IF          LV_ERR_BN_EFP_CRASH = 0      and
            LV_ERR_CAN_BOFF = 0
THEN        IF      CTR_CR_SWO_EKP = AH
            THEN  STATE_EFP_CRASH_CAN = 2H
            ELSE  STATE_EFP_CRASH_CAN = 0H
            ENDIF
ELSE        STATE_EFP_CRASH_CAN = FFH
ENDIF
```

43.1.4.6 DSC-Modul

Formula section:

```
DREHMOMENT_ANF_DSC
IF          LV_ERR_BN_TQ_TCS = 0            AND
            LV_ERR_CAN_BOFF = 0
THEN        // checksum_torque_DSC:
     CKS_CAN_TQ_TCS          =       CHKSM_TORQ_DSC

     // free running test cycle counter TCS :

     TCC_CAN_TQ_TCS  =       ALIV_TORQ_DSC


     // Torque_target _DSC:
     TQ_TCS_FAST_BN  =       TORQ_TAR_DSC

     // Torque_target_adjustment-range_posistive _DSC:
     IF             LV_TQ_MSR_REQ = 1               AND
                    LC_MSR_BN = 1
     THEN  TQ_TCS_SLOW_BN = TQ_TCS_FAST_BN
     ELSE  TQ_TCS_SLOW_BN = TORQ_TAR_ADJR_POS_TCS
     ENDIF
```

```
// state TCS-intervention :
if   ST_TORQ_TAR_DSC       =      00b
then       STATE_TCS_INTV        =      NO_INTERVENTION
elseif           ST_TORQ_TAR_DSC       =      01b
           then  STATE_TCS_INTV        =      MSR_INTERVENTION
                       LV_TQ_MSR_REQ = 1
                 else  LV_TQ_MSR_REQ = 0
                 endif
elseif           ST_TORQ_TAR_DSC       =      10b
      then STATE_TCS_INTV        =      ASR_INTERVENTION
                       LV_TQ_ASR_REQ = 1
           else  LV_TQ_ASR_REQ = 0
           endif
elseif           ST_TORQ_TAR_DSC       =      11b
      then STATE_TCS_INTV        =      INTERFACE_LOCKED
                       LV_TCS_LIH_CAN = 1
                 else  LV_TCS_LIH_CAN = 0
                 endif
endif
ELSE       // checksum_torque_DSC:
           CKS_CAN_TQ_TCS        =      FFH

           // free running test cycle counter TCS :
           TCC_CAN_TQ_TCS =      FH


           // Torque_target _DSC:
           TQ_TCS_FAST_BN  =     8000H

           // Torque_target_adjustment-range_posistive _DSC:
           TQ_TCS_SLOW_BN =      8000H

           // state TCS-intervention :
           STATE_TCS_INTV        =      3H
           LV_TCS_LIH_CAN        =      1
           LV_TQ_MSR_REQ         =      0
           LV_TQ_ASR_REQ         =      0


ENDIF
```

Geschwindigkeit:

```
IF      LV_ERR_BN_VS_TCS = 0          AND
        LV_ERR_CAN_BOFF = 0
THEN    // speed_vehicle:
      VS_CAN     =    V_VEH      (converted from BN2000)
      WHEEL      =    V_VEH      (converted from BN2000)

      IF          V_VEH      =      FFFH
      THEN   LV_ERR_VS_CAN = 1
      ELSE   LV_ERR_VS_CAN = 0
      ENDIF

      // acceleration_vehicle_longitudinal_DSC:
      IF     ACLN_VEH_LN_DSC       =      800H   //Signal invalid
      THEN   AC_VEH_LGT_TCS        =      0 m/s²
      ELSE   AC_VEH_LGT_TCS  =     ACLN_VEH_LN_DSC
             (converted from BN2000)
      ENDIF


      // acceleration_vehicle_across_DSC:
      IF     ACLN_VEH_ACRO_DSC     =      800H   //Signal invalid
      THEN   AC_VEH_TRV_TCS        =      0 m/s²
      ELSE   AC_VEH_TRV_TCS  =     ACLN_VEH_ACRO_DSC
             (converted from BN2000)
      ENDIF

ELSE    // speed_vehicle:
        VS_CAN     =    FFH
        WHEEL      =    FFFFH
        LV_ERR_VS_CAN = 1

        // acceleration_vehicle_longitudinal_DSC:
        AC_VEH_LGT_TCS  =      0 m/s²

        // acceleration_vehicle_across_DSC:
```

```
              AC_VEH_TRV_TCS  =        0 m/s²

ENDIF
```

Geschwindigkeit_Rad:

This message will not be read into internal variables and evaluated as for rough road detection the function of CAN11h will be used.

STAT_DSC:

```
IF          LV_ERR_BN_TCS = 0         AND
            LV_ERR_CAN_BOFF = 0
THEN        // status close-loop-control via CAN:
if    ST_CLCTR              = 0000 0000b
then        STATE_TCS_CTL    = NO_CONTROL
elseif      ST_CLCTR         = 0000 0001b
then        STATE_TCS_CTL    = ABS_CONTROL
elseif      ST_CLCTR         = 0000 0010b
then        STATE_TCS_CTL    = ASR_CONTROL
elseif      ST_CLCTR         = 0000 0100b
then        STATE_TCS_CTL    = TCS_CONTROL
elseif      ST_CLCTR         = 0000 1000b
then        STATE_TCS_CTL    = HBA_CONTROL
elseif      ST_CLCTR         = 0001 0000b
then        STATE_TCS_CTL    = MSR_CONTROL
elseif      ST_CLCTR         = 0010 0000b
then        STATE_TCS_CTL    = EBV_CONTROL
elseif      ST_CLCTR         = 0100 0000b
then        STATE_TCS_CTL    = DYNO_ACTIVE
elseif      ST_CLCTR         = 1111 1111b
then        STATE_TCS_CTL    = SIGNAL_INVALID
else        STATE_TCS_CTL    = MULTI_CONTROL
endif

IF          ST_CLCTR != 1111 1111b   AND
            (ST_CLCTR[Bit 0] = 1   OR          //ABS_CONTROL
```

```
                ST_CLCTR[Bit 1] = 1   OR            //ASR_CONTROL
                ST_CLCTR[Bit 4] = 1)                //MSR_CONTROL
THEN        LV_PLAUS_ASR_CTL = 1
ELSE        LV_PLAUS_ASR_CTL = 0
ENDIF

IF          ST_CLCTR != 1111 1111b   AND
            ST_CLCTR[Bit 2] = 1          // DSC/ TCS_CONTROL
THEN        LV_PLAUS_ESP_CTL = 1
ELSE        LV_PLAUS_ESP_CTL = 0
ENDIF

// chassis control function active:
IF          ST_CLCTR != 1111 1111b   AND
            (ST_CLCTR[Bit 0] = 1   OR           //ABS_CONTROL
             ST_CLCTR[Bit 1] = 1   OR           //ASR_CONTROL
             ST_CLCTR[Bit 2] = 1   OR           //DSC_CONTROL
             ST_CLCTR[Bit 3] = 1)               //HBA_CONTROL
THEN        LV_TCS_CTL_ACT = 1
ELSE        LV_TCS_CTL_ACT = 0
ENDIF

// status traction control system via CAN :
if    ST_DSC            =       000b
then        STATE_TCS_CAN   =       VALID
elseif      ST_DSC          =       001b
then        STATE_TCS_CAN   =       PASSIV
elseif      ST_DSC          =       010b
then        STATE_TCS_CAN   =       DEFECTIVE
elseif      ST_DSC          =       100b
then        STATE_TCS_CAN   =       TRACTION_MODE
elseif      ST_DSC          =       110 b
then        STATE_TCS_CAN   =       UNDERVOLTAGE_TCS
elseif      ST_DSC          =       111b
then        STATE_TCS_CAN   =       INVALID_SIGNAL
endif
```

```
// ST_DSC - status deceleration actual-value:
if     ST_DCRN_AVL        =       0H
then           STATE_TCS_DECE =        NO_INTERVENTION
elseif         ST_DCRN_AVL    =        1H
then           STATE_TCS_DECE =        DCC_INTV_ACCEPTED
elseif         ST_DCRN_AVL    =        2H
then           STATE_TCS_DECE =        EMF_INTV_ACCEPTED
elseif         ST_DCRN_AVL    =        3H
then           STATE_TCS_DECE =        HBA_INTV_ACCEPTED
elseif         ST_DCRN_AVL    =        FH
then           STATE_TCS_DECE =        SIGNAL_INVALID
endif

// status brake-pressure:
STATE_BRAKE_PRS       =       ST_BRP

// brake-pressure:
BRAKE_PRS             =       BRP
ELSE
    // status close-loop-control via CAN:
    STATE_TCS_CTL     = 0H
    LV_PLAUS_ASR_CTL = 0
    LV_PLAUS_ESP_CTL = 0

    //status chassis control functions
    LV_TCS_CTL_ACT = 0

    // status traction control system via CAN :
    STATE_TCS_CAN     =       7H

    // ST_DSC - status deceleration actual-value:
    STATE_TCS_DECE  =       0H

    // status brake-pressure:
    STATE_BRAKE_PRS         =       2H
```

```
    // brake-pressure:
    BRAKE_PRS          =        FFH
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_MSR_BN	1	0...1H	0...1	1	[-]
LC for allowing MSR-intervention in case of TQ_TCS_FAST_BN <> TQ_TCS_SLOW_BN by DSC					

43.1.4.7 EGS-Modul

Formula section:

```
DREHMOMENT_ANF_EGS
IF(1)      LV_ERR_BN_TQ_ETCU = 0          AND
           LV_ERR_CAN_BOFF = 0
THEN(1)    // Checksum ETCU:
           CKS_CAN_TQ_ETCU        =     CHKSM_TORQ_EGS
           // Alive-Counter ETCU:
           TCC_CAN_TQ_ETCU        =     ALIV_TORQ_EGS
           // Torque_target _EGS:
           TQ_GS_FAST_BN    =     TORQ_TAR_EGS
           // Torque_target_adjustment-range_posistive _EGS:
           TQ_GS_SLOW_BN    =     TORQ_TAR_ADJR_POS_EGS
           // state ETCU-intervention :
           if     ST_TORQ_TAR_EGS       =     00H
           then   STATE_ETCU_INTV             =     NO_INTERVENTION
           elseif ST_TORQ_TAR_EGS       =     01H
           then   STATE_ETCU_INTV             =     TORQUE_INCREASE
                  LV_TQ_GS_INC_REQ = 1
           else   LV_TQ_GS_INC_REQ                  =                    0
           endif
           elseif ST_TORQ_TAR_EGS       =     02H
           then   STATE_ETCU_INTV             =     TORQUE_DECREASE
                  LV_TQ_GS_DEC_REQ = 1
           else   LV_TQ_GS_DEC_REQ = 0
           endif
           elseif ST_TORQ_TAR_EGS       =     03H
           then   STATE_ETCU_INTV             =     INTERFACE_LOCKED
                  LV_ETCU_LIH_CAN = 1
           else   LV_ETCU_LIH_CAN = 0
           endif
endif

// status_OBD_gearbox ST_OBD_GRB:
IF         ST_OBD_GRB       =     00H
THEN       STATE_ETCU_OBD   =     MIL_OFF
ELSEIF     ST_OBD_GRB       =     02H
THEN       STATE_ETCU_OBD   =     MIL_OFF
ELSEIF     ST_OBD_GRB       =     04H
THEN       STATE_ETCU_OBD   =     MIL_ON
ELSEIF     ST_OBD_GRB       =     06H
THEN       STATE_ETCU_OBD   =     MIL_ON
ELSEIF     ST_OBD_GRB       =     08H
THEN       STATE_ETCU_OBD   =     MIL_FLL
ELSEIF     ST_OBD_GRB       =     0AH
THEN       STATE_ETCU_OBD   =     MIL_FLL
ELSEIF     ST_OBD_GRB       =     0CH
THEN       STATE_ETCU_OBD   =     IDLE
ELSEIF     ST_OBD_GRB       =     0EH
```

```
THEN      STATE_ETCU_OBD =      INI_VALUE
ELSEIF    ST_OBD_GRB      =      0FH
THEN      STATE_ETCU_OBD =      INVALID_SIGNAL
ENDIF

// status_OBD_error_gearbox:
IF        ST_OBD_ERR_GRB          =     00H
THEN      STATE_ETCU_OBD_ERR      =     NO_OBD_ERROR
ELSEIF    ST_OBD_ERR_GRB          =     01H
THEN      STATE_ETCU_OBD_ERR      =     OBD_ERROR
ELSEIF    ST_OBD_ERR_GRB          =     02H
THEN      STATE_ETCU_OBD_ERR      =     INVALID_SIGNAL
ENDIF

// Status starting torque limitation:
IF        ST_LIM_STORQ            = 00H
THEN      STATE_ST_TQ_LIM_GS      = NO_INTERVENTION
ELSEIF    ST_LIM_STORQ            = 01H
THEN      STATE_ST_TQ_LIM_GS      = INTERVENTION
ELSEIF    STATE_ST_TQ_LIM_GS      = INVALID_SIGNAL
ENDIF

// Torque_draw gearbox:
if        TORQ_DRAW_GRB = FFH
then      TQ_CONV_CAN     = 0 Nm
else      TQ_CONV_CAN     = TORQ_DRAW_GRB
endif

// Temperature gearbox:
IF        LV_AT = 1
THEN      IF      TEMP_GRB   = FFH
          THEN TEMP_GB      = C_TEMP_GB_SUB
          ELSE  TEMP_GB      = TEMP_GRB
          ENDIF
ELSE      TEMP_GB      = -40 °C
ENDIF

// Status_overtemperature_gearbox:
IF        LV_AT = 1
THEN      IF            ST_OTMP_GRB        = 00b
          THEN          STATE_TEMP_GB      = NO_REQUEST
          ELSEIF        ST_OTMP_GRB        = 01b
          THEN          STATE_TEMP_GB      = ACTIVE_STEP_1
          ELSEIF        ST_OTMP_GRB        = 10b
          THEN          STATE_TEMP_GB      = ACTIVE_STEP_2
          ELSEIF        ST_OTMP_GRB        = 11b
          THEN          STATE_TEMP_GB      = INVALID_SIGNAL
          ENDIF
ELSE               STATE_TEMP_GB = NO_REQUEST
ENDIF

ELSE(1)   // Checksum ETCU:
```

```
CKS_CAN_TQ_ETCU      =       FFH

// Alive-Counter ETCU:
TCC_CAN_TQ_ETCU      =       FH

// Torque_target _EGS:
TQ_GS_FAST_BN        =       8000H

// Torque_target_adjustment-range_posistive _EGS:
TQ_GS_SLOW_BN        =       8000H

// state ETCU-intervention :
STATE_ETCU_INTV      =       3H
LV_ETCU_LIH_CAN      =       1
LV_TQ_GS_INC_REQ     =       0
LV_TQ_GS_DEC_REQ     =       0

// status_OBD_gearbox ST_OBD_GRB:
STATE_ETCU_OBD       =       EH

// status_OBD_error_gearbox:
STATE_ETCU_OBD_ERR           =       3H

// Status starting torque limitation:
STATE_ST_TQ_LIM_GS =  INVALID_SIGNAL  (3H)

// Torque_draw gearbox:
TQ_CONV_CAN   = 0 Nm

// Temperature gearbox:
TEMP_GB = 0H

// Status_overtemperature_gearbox:
STATE_TEMP_GB = 0H

ENDIF(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEMP_GB_SUB	1	0...FFH	-40...215	1	[°C]
Default temperature for automatic gearbox in case of error TEMP_GB					

43.1.4.8 SSG-Modul

Formula section:

```
DREHMOMENT_ANF_SSG
IF        LV_ERR_BN_TQ_AMT = 0          AND
          LV_ERR_CAN_BOFF = 0
```

```
THEN        // Checksum TQ_AMT:
CKS_CAN_TQ_AMT      =       CHKSM_TORQ_SSG

// Alive-Counter TQ_AMT:
TCC_CAN_TQ_AMT      =       ALIV_TORQ_SSG

// Torque_target _SSG:
TQ_AMT_FAST_BN      =       TORQ_TAR_SSG

// Torque_target_adjustment-range_positive:
TQ_AMT_SLOW_BN      =       TORQ_TAR_ADJR_POS_SSG

// state ETCU-intervention STATE_AMT_INTV:
if          ST_TORQ_TAR_SSG         = 00H
then                STATE_AMT_INTV          = NO_INTERVENTION
elseif              ST_TORQ_TAR_SSG         = 01H
      then  STATE_AMT_INTV  = TORQUE_INCREASE
            LV_TQ_AMT_INC_REQ = 1
      else  LV_TQ_AMT_INC_REQ = 0
      endif
elseif              ST_TORQ_TAR_SSG         = 02H
      then  STATE_AMT_INTV  = TORQUE_DECREASE
            LV_TQ_AMT_DEC_REQ = 1
   else     LV_TQ_AMT_DEC_REQ = 0
   endif
elseif              ST_TORQ_TAR_SSG         = 03H
      then  STATE_AMT_INTV  = INTERFACE_LOCKED
            LV_AMT_LIH_CAN = 1
     else   LV_AMT_LIH_CAN = 0
     endif
endif

// status_OBD_gearbox ST_OBD_GRB_SSG:
IF          ST_OBD_GRB_SSG          =       00H
THEN                STATE_AMT_OBD           =       MIL_OFF
ELSEIF      ST_OBD_GRB_SSG          =       02H
THEN                STATE_AMT_OBD           =       MIL_OFF
ELSEIF      ST_OBD_GRB_SSG          =       04H
THEN                STATE_AMT_OBD           =       MIL_ON
ELSEIF      ST_OBD_GRB_SSG          =       06H
THEN                STATE_AMT_OBD           =       MIL_ON
ELSEIF      ST_OBD_GRB_SSG          =       08H
THEN                STATE_AMT_OBD           =       MIL_FLL
ELSEIF      ST_OBD_GRB_SSG          =       0AH
THEN                STATE_AMT_OBD           =       MIL_FLL
ELSEIF      ST_OBD_GRB_SSG          =       0CH
THEN                STATE_AMT_OBD           =       IDLE
ELSEIF      ST_OBD_GRB_SSG          =       0EH
THEN                STATE_AMT_OBD           =       INI_VALUE
ELSEIF      ST_OBD_GRB_SSG          =       0FH
THEN                STATE_AMT_OBD           =       INVALID_SIGNAL
ENDIF
```

```
// status_OBD_error_gearbox ST_OBD_ERR_GRB_SSG:
IF        ST_OBD_ERR_GRB_SSG   = 00H
THEN            STATE_AMT_OBD_ERR     = NO_OBD_ERROR
ELSEIF    ST_OBD_ERR_GRB_SSG   = 01H
THEN            STATE_AMT_OBD_ERR     = OBD_ERROR
ELSE            STATE_AMT_OBD_ERR     = INVALID_SIGNAL
ENDIF

// Status starting torque limitation:
IF        ST_LIM_STORQ_SSG       = 00H
THEN            STATE_ST_TQ_LIM_AMT   = NO_INTERVENTION
ELSEIF    ST_LIM_STORQ_SSG       = 01H
THEN            STATE_ST_TQ_LIM_AMT   = INTERVENTION
ELSE            STATE_ST_TQ_LIM_AMT   = INVALID_SIGNAL
ENDIF

// Status shift-process SSG:
IF        ST_SHPS_SSG        = 00H
THEN            STATE_AMT            = NO_INTERVENTION
ELSEIF    ST_SHPS_SSG        = 01H
THEN            STATE_AMT            = TORQUE_DECREASE
ELSEIF    ST_SHPS_SSG        = 02H
THEN            STATE_AMT            = N_REGULATION
ELSEIF    ST_SHPS_SSG        = 03H
THEN            STATE_AMT            = TORQUE_INCREASE
ELSEIF    ST_SHPS_SSG        = 07H
THEN            STATE_AMT            = INVALID_SIGNAL
ENDIF

// Status clutch SSG:
IF        ST_CLT_SSG         = 00H
THEN            STATE_CLU_AMT    = CLU_OPEN
ELSEIF    ST_CLT_SSG         = 01H
THEN            STATE_CLU_AMT    = CREEPING
ELSEIF    ST_CLT_SSG         = 02H
THEN            STATE_CLU_AMT    = DRIVE_OFF
ELSEIF    ST_CLT_SSG         = 03H
THEN            STATE_CLU_AMT    = CLU_CLOSED
ELSEIF    ST_CLT_SSG         = 07H
THEN            STATE_CLU_AMT    = INVALID_SIGNAL
ENDIF

// Request engine-stop:
IF        RQ_MOTORSTOP = 01H
THEN            LV_AMT_ES = 1
ELSE            LV_AMT_ES = 0
ENDIF

// Limit_RPM_engine:
N_MAX_AMT_CAN        =        LIM_RPM_ENG
(converted from BN2000)
```

```
// Status gear-shift SSG:
IF          ST_GRLV_ACV = 01H
THEN               LV_GS = 1
ELSE               LV_GS = 0
ENDIF
ELSE        // Checksum TQ_AMT:
CKS_CAN_TQ_AMT     =      FFH

// Alive-Counter TQ_AMT:
TCC_CAN_TQ_AMT     =      FH

// Torque_target _SSG:
TQ_AMT_FAST_BN     =      8000H

// Torque_target_adjustment-range_positive:
TQ_AMT_SLOW_BN     =      8000H

// state ETCU-intervention STATE_AMT_INTV:
STATE_AMT_INTV       = 3H
LV_AMT_LIH_CAN       = 1
LV_TQ_AMT_INC_REQ  = 0
LV_TQ_AMT_DEC_REQ = 0

// status_OBD_gearbox ST_OBD_GRB_SSG:
STATE_AMT_OBD             =      EH

// status_OBD_error_gearbox ST_OBD_ERR_GRB_SSG:
STATE_AMT_OBD_ERR         = INVALID_SIGNAL (2)

// Status starting torque limitation:
STATE_ST_TQ_LIM_AMT       = INVALID_SIGNAL  (3H)

// Status shift-process SSG:
STATE_AMT            =  0

// Status clutch SSG:
STATE_CLU_AMT        = 0

// Request engine-stop:
LV_AMT_ES = 0

// Limit_RPM_engine:
N_MAX_AMT_CAN        =      0H

// Status gear-shift SSG:
LV_GS = 0
ENDIF
```

43.1.4.9 DKG-Modul (TCT)

DREHMOMENT_ANF_DKG (ID = B8h)

Application conditions:

Initialisation: all 0

Recurrence: message received with recurrence 10ms (tmin = 9ms, tmax = 11ms)

Activation: LV_VAR_TCT = 1

Formula section:

```
IF(1)     LV_ERR_BN_TQ_TCT = 0           AND
          LV_ERR_CAN_BOFF = 0
THEN(1)
          // Checksum TCT:
          CKS_CAN_TQ_TCT  =     CHKSM_TORQ_DKG
          // Alive-Counter TCT:
          TCC_CAN_TQ_TCT  =     ALIV_TORQ_DKG
          // Torque_target _TCT:
          TQ_GS_FAST_BN   =     TORQ_TAR_DKG               (converted)
          // Torque_target_adjustment-range_posistive _TCT:
          TQ_GS_SLOW_BN   =     TORQ_TAR_ADJR_POS_DKG   (converted)

          // state TCT-mode :
          LV_TQ_GS_INC_REQ = 0
          LV_TQ_GS_DEC_REQ = 0
          LV_TCT_LIH_CAN = 0
          LV_TCT_ES   = 0
          LV_N_SP_TCT_REQ = 0
          if      ST_MOD_DKG        =     00H
          then    STATE_TCT_INTV    =     NO_INTERVENTION
          elseif  ST_MOD_DKG        =     02H
          then    STATE_TCT_INTV    =     CREEP
          elseif  ST_MOD_DKG        =     03H
```

```
then    STATE_TCT_INTV          =       TCT_INTV_DEC
        LV_TQ_GS_DEC_REQ = 1
elseif  ST_MOD_DKG              =       04H
then    STATE_TCT_INTV          =       TCT_INTV_INC
        LV_TQ_GS_INC_REQ = 1
elseif  ST_MOD_DKG              =       06H
then    STATE_TCT_INTV          =       N_REGULATION
        LV_N_SP_TCT_REQ = 1
elseif  ST_MOD_DKG              =       0CH
then    STATE_TCT_INTV          =       TQ_DEC
        ~~LV_TQ_GS_DEC_REQ = 1~~
elseif  ST_MOD_DKG              =       0DH
then    STATE_TCT_INTV          =       TCT_ES
        LV_TCT_ES    = 1
elseif  ST_MOD_DKG              =       0FH
then    STATE_TCT_INTV          =       INVALID
        LV_TCT_LIH_CAN = 1
endif

// torque_clutch_DKG:
IF      TORQ_CLT_DKG     = FFFH
THEN    TQ_TCT_CAN = 0 Nm
ELSE    TQ_TCT_CAN = MAX (0 Nm; TORQ_CLT_DKG)    (converted)
ENDIF

//RPM_target_engine_DKG:
If      RPM_TAR_ENG_DKG != FFFH
Then    N_SP_TCT = RPM_TAR_ENG_DKG  (converted)

        LV_N_SP_TCT = LV_N_SP_TCT_REQ
Else    N_SP_TCT = 0
        LV_N_SP_TCT = 0
Endif

// status_OBD_gearbox ST_OBD_GRB:
STATE_ETCU_OBD =         EH
```

// status_OBD_error_gearbox:

STATE_ETCU_OBD_ERR = 3H

// Status starting torque limitation:

STATE_ST_TQ_LIM_GS = INVALID_SIGNAL (3H)

// Torque_draw gearbox:

TQ_CONV_CAN = 0 Nm

~~*// Temperature gearbox:*~~

~~TEMP_GB = 0H~~

~~*// Status_overtemperature_gearbox:*~~

~~STATE_TEMP_GB = 0H~~

ELSE(1)

// Checksum ETCU:

CKS_CAN_TQ_TCT = FFH

// Alive-Counter ETCU:

TCC_CAN_TQ_TCT = FH

// Torque_target _EGS:

TQ_GS_FAST_BN = 8000H

// Torque_target_adjustment-range_posistive _EGS:

TQ_GS_SLOW_BN = 8000H

// state ETCU-intervention :

STATE_TCT_INTV = FH
LV_TCT_LIH_CAN = 1
LV_TQ_GS_INC_REQ = 0
LV_TQ_GS_DEC_REQ = 0
LV_TCT_ES = 0
LV_N_SP_TCT_REQ = 0

```
// torque_clutch_DKG:
TQ_TCT_CAN          =       0H

//RPM_target_engine_DKG:
N_SP_TCT = 0H
LV_N_SP_TCT = 0

// status_OBD_gearbox ST_OBD_GRB:
STATE_ETCU_OBD =        EH

// status_OBD_error_gearbox:
STATE_ETCU_OBD_ERR   =        3H

// Status starting torque limitation:
STATE_ST_TQ_LIM_GS     =  INVALID_SIGNAL  (3H)

// Torque_draw gearbox:
TQ_CONV_CAN       = 0 Nm
```

~~// Temperature gearbox:~~

~~TEMP_GB   = 0H~~

~~// Status_overtemperature_gearbox:~~

~~STATE_TEMP_GB = 0H~~

ENDIF (1)

ST_DKG (Status DKG) (ID =37D h)

Application conditions:

Initialisation: at reset TEMP_GB = -40 °C, STATE_TEMP_GB = NO_REQUEST

Recurrence: 1 s (tmin=0.9s, tmax=1.1s)

Activation: LV_VAR_TCT = 1

Formula section:
:

```
IF      LV_ERR_CAN_BOFF = 0 AND
        LV_ERR_BN_STAT_TCT = 0
THEN
        // Temperature gearbox:
        IF    TEMP_DKG = FFH
        THEN  TEMP_GB    = C_TEMP_GB_SUB
        ELSE  TEMP_GB    = TEMP_DKG
        ENDIF

        // Status_overtemperature_gearbox:
        IF     ST_OTMP_DKG      = 00b
        THEN   STATE_TEMP_GB    = NO_REQUEST
        Elseif ST_OTMP_DKG      = 01b
        THEN   STATE_TEMP_GB    = ACTIVE_STEP_1
        Elseif ST_OTMP_DKG      = 10b
        THEN   STATE_TEMP_GB    = ACTIVE_STEP_2
        Elseif ST_OTMP_DKG      = 11b
        THEN   STATE_TEMP_GB    = INVALID_SIGNAL
        ENDIF

ELSE    setting default values:
        TEMP_GB          = -40 °C
        STATE_TEMP_GB    = NO_REQUEST
ENDIF
```

43.1.4.10 Transmission Data (from EGS, SSG or DKG)

GETRIEBEDATEN (ID = BAh)

Application conditions:

Initialisation: all 0

Recurrence: message received with recurrence 20ms (tmin = 19ms, tmax = 21ms)

Activation: LV_VAR_ETCU = 1

Formula section:

```
IF        LV_ERR_BN_ETCU = 0              AND
          LV_ERR_CAN_BOFF = 0
THEN      // checksum ETCU :
          CKS_CAN_ETCU = CHKSM_GRB
          // Alive counter ETCU:
          TCC_CAN_ETCU = ALIV_GRB

          // target gear information from ETCU :
          if           ST_GR_GRB = 0000b      (reserve)
                 or    ST_GR_GRB = 0001b      (Neutral)
                 or    ST_GR_GRB = 0011b      (Park)
                 or    ST_GR_GRB = 1111b      (invalid signal)
          then   GEAR_INFO  = 00H
          elseif ST_GR_GRB = 0101b      (1.gear)
          then   GEAR_INFO  = 01H
          elseif ST_GR_GRB = 0110b      (2.gear)
          then   GEAR_INFO  = 02H
          elseif ST_GR_GRB = 0111b      (3.gear)
          then   GEAR_INFO  = 03H
          elseif ST_GR_GRB = 1000b      (4.gear)
          then   GEAR_INFO  = 04H
          elseif ST_GR_GRB = 1001b      (5.gear)
          then   GEAR_INFO  = 05H
          elseif ST_GR_GRB = 1010b      (6.gear)
          then   GEAR_INFO  = 06H
          elseif ST_GR_GRB = 1011b      (7.gear)
          then   GEAR_INFO  = 07H
          elseif ST_GR_GRB = 0010b      (reverse gear)
          then   GEAR_INFO  = 0AH
          endif

          //gearbox-drive-train_reinforcement:
          If     GRDT_REIN  <     FFEH
          Then   FAC_GB_GAIN      = GRDT_REIN
          Else   FAC_GB_GAIN      = 00H
          Endif
```

```
//data sent only by EGS (LV_AT):
IF      LV_AT = 1
THEN
        // Status_torque-converter-lockup-clutch:
        IF          ST_CCLT      = 0H
        THEN        STATE_CC     = CC open
        ELSEIF      ST_CCLT      = 1H
        THEN        STATE_CC     = CC controlled
        ELSEIF      ST_CCLT      = 02H
        THEN        STATE_CC     = CC closed
        ELSEIF      ST_CCLT      = 03H
        THEN        STATE_CC     = Signal invalid
        ENDIF

        // Status_mode_gearbox:
        IF          ST_MOD_GRB        = 00H
        THEN        STATE_MOD_GB      = SSC_SBC_OFF
        ELSEIF      ST_MOD_GRB        = 01H
        THEN        STATE_MOD_GB      = SSC_ON
        ELSEIF      ST_MOD_GRB        = 02H
        THEN        STATE_MOD_GB      = SBC_ON
        ELSEIF      ST_MOD_GRB        = 03H
        THEN        STATE_MOD_GB      = SSC_SBC_ON
        ELSEIF      ST_MOD_GRB        = 07H
        THEN        STATE_MOD_GB      = INVALID SIGNAL
        ENDIF
ENDIF

//data sent only by SSG (LV_VAR_AMT):
IF      LV_VAR_AMT = 1
THEN
        // Status_hydraulic-pump_active ST_HYPP_ACV:
        IF          ST_HYPP_ACV       = 1
        THEN        LV_PUMP_AMT_ON = 1
        ELSE        LV_PUMP_AMT_ON = 0
        ENDIF
ENDIF

//data received from DKG or EGS (LV_VAR_TCT or LV_AT):
IF      LV_VAR_TCT = 1 or  LV_AT = 1
THEN
        // Status gear-shift:
        IF          ST_GRLV_ACV = 01H
        THEN        LV_GS = 1
        ELSE        LV_GS = 0
        ENDIF
ENDIF

//data sent only by DKG (LV_VAR_TCT):
IF      LV_VAR_TCT = 1
THEN
```

```
                // status race start:
                STATE_MAX_AC_ST = ST_RSTA
            ENDIF

ELSE
            // target gear information from ETCU :
            GEAR_INFO  = 00H

            //gearbox-drive-train_reinforcement:
            FAC_GB_GAIN         = 00H

            // Status_torque-converter-lockup-clutch:
            STATE_CC    = 3H

            // Status_mode_gearbox:
            STATE_MOD_GB      = 07H

            // Status_hydraulic-pump_active ST_HYPP_ACV:
            LV_PUMP_AMT_ON = 0

            IF      LV_VAR_TCT = 1  or  LV_AT = 1
            THEN
                // Status gear-shift EGS:
                LV_GS = 0
            ENDIF

            IF      LV_VAR_TCT = 1
            THEN
                // Status race start:
                STATE_MAX_AC_ST = 11b
            ENDIF

            // checksum ETCU :
            CKS_CAN_ETCU = FFH
            // Alive counter ETCU:
            TCC_CAN_ETCU = FH

ENDIF
```

GETRIEBEDATEN 2 (ID = 1A2h)

Application conditions:

Initialisation: all 0

Recurrence: message received with recurrence 20ms (tmin = 19ms, tmax = 21ms)

Activation: LV_VAR_TCT = 1 **or** LV_AT = 1

Formula section:

```
If      LV_ERR_CAN_BOFF   =  0  and
        LV_ERR_BN_ETCU_2  =  0
Then    //RPM_gearbox_turbine:
        N_GB  = RPM_GRB_TURB

        //data sent only by DKG (LV_VAR_TCT):
        IF     LV_VAR_TCT = 1
        THEN
               // request_boost_RPM_idling:
               if     RQ_BST_RPM_IDLG        ! =     FFH
               Then   N_SP_IS_TCT_CAN =      RQ_BST_RPM_IDLG
               Else   N_SP_IS_TCT_CAN =      0
               Endif
               // Status_starting_clutch:
               STATE_ETCU_CLU = ST_STCLT
        ENDIF

        //status if forced-shift up is active in actual gear shift mode:
        IF     ST_RCOG_FSHUP = 0
        THEN   LV_INH_GP_SUP = 0
        ELSE   LV_INH_GP_SUP = 1
        ENDIF

        //RPM_at_gearbox_output / drivetrain turbine:
        N_GB_OUT = RPM_GRB_NEGL_2           (converted)

        //request dynamisation of engine-rpm signal for display (kombi) :
        IF     RQ_GRB_DYNS_DISP_RPM_ENG = 01b
```

```
            THEN  LV_N_DISP_DYN = 1
            ELSE  LV_N_DISP_DYN = 0
            ENDIF

            //status of gearbox sport switch:
            STATE_ETCU_SPT_SWI = ST_GRB_SSW

Else
            // request_boost_RPM_idling:
            N_SP_IS_TCT_CAN = 0H

            //RPM_gearbox_turbine:
            N_GB  = 0
            // Status_starting_clutch:
            STATE_ETCU_CLU  = 0
            //status if forced-shift up is active in actual gear shift mode:
            LV_INH_GP_SUP = 1
            //RPM_at_gearbox_output / drivetrain turbine:
            N_GB_OUT = 0  rpm
            //request dynamisation of engine-rpm signal for display (kombi) :
            LV_N_DISP_DYN = 0
            //status of gearbox sport switch:
            STATE_ETCU_SPT_SWI = 3H
ENDIF
```

GETRIEBEDATEN 3 (ID = 3B1h):

Application conditions:

Initialisation: all 0

Recurrence: message received with recurrence 100ms (tmin = 90ms, tmax = 110ms) and event triggered (tmin=50ms)

Activation: LV_VAR_ETCU_3 = 1

Formula section:

```
If      LV_ERR_CAN_BOFF   = 1   or
        LV_ERR_BN_ETCU_3  = 1
Then    TCC_CAN_ETCU_3  =  FH
        CKS_CAN_ETCU_3  =  FFH
        ipm_typ_fp      =  0
        ipm_typ_mdkw    =  0
        dm_ab_fws       =  -32768 Nm
        VS_MAX_LIH_TCT  = 7FFFH
        //data sent only by DKG (LV_VAR_TCT):
        IF      LV_VAR_TCT = 1
        THEN
                LV_MIL_REQ_ETCU = 0
        ENDIF
Else    // Checksum ETCU:
        CKS_CAN_ETCU_3  =  CHKSM_GRB_3
        // Alive-Counter ETCU:
        TCC_CAN_ETCU_3  =  ALIV_GRB_3
        if      IDX_ACPD > 5H   // not defined and  invalid case
        then    ipm_typ_fp     =  0
        else    ipm_typ_fp     =  IDX_ACPD
        endif
        if      IDX_TORQ_FIL > 5H   // not defined and invalid case
        then    ipm_typ_mdkw = 0
        else    ipm_typ_mdkw = IDX_TORQ_FIL
        endif
        if      TORQ_DVCH_RISE = FFFH   // invalid case
        then    dm_ab_fws      =  -32768 Nm
        else    dm_ab_fws      =  TORQ_DVCH_RISE (converted from BN)
        endif
```

```
        //data sent only by DKG (LV_VAR_TCT):
        IF      LV_VAR_TCT = 1
        THEN
                // bit requesting Mil illumination by transmission control unit:
                if      RQ_MIL_GRB = 1H
                then    LV_MIL_REQ_ETCU = 0
                else    If      RQ_MIL_GRB = 2H
                        then    LV_MIL_REQ_ETCU = 1
                        endif
                endif
        ENDIF

        //limitation maximum speed gearbox limp home:
        if      LIM_MAX_V_GRB_EMMOD = 3FH   // invalid case
        then    VS_MAX_LIH_TCT  = 7FFFH
        else    VS_MAX_LIH_TCT  =  LIM_MAX_V_GRB_EMMOD (converted from BN)
        endif
Endif
```

ANZEIGE GETRIEBEDATEN (ID = 1D2h):

Application conditions:

Initialisation: all 0

Recurrence: message received with recurrence 200ms (tmin = 90ms, tmax = 110ms) and event triggered (tmin=50ms)

Activation: LV_AT = 1 **or** LV_VAR_TCT

Formula section:

```
If      LV_ERR_CAN_BOFF         =       1       or
        LV_ERR_BN_ETCU_DISP   = 1
Then    STATE_ETCU_PROG_INFO            = FH
        LV_ETCU_SPT_SWI = 0
Else    STATE_ETCU_PROG_INFO            = CTR_DISP_PRG_GRB
        //control display sport mode gearbox:
        IF      CTR_DISP_SPMOD_GRB = 1H
        Then    LV_ETCU_SPT_SWI = 1
```

```
                Else    LV_ETCU_SPT_SWI = 0
                Endif

    Endif
```

43.1.4.11 IHKA

Formula section:

```
    WAERMESTROM_KLIMA
    IF          LV_ERR_BN_ACC = 0           AND
                LV_ERR_CAN_BOFF = 0
    THEN        // torque target air-conditioning-compressor:

        if      TORQ_TAR_ACCM   <       C_TQ_ACCIN
        then    TQ_ACCIN _CAN   =       0
        else    TQ_ACCIN_CAN    =       TORQ_TAR_ACCM
        endif

        // logical variable ACC request LV_ACCIN :
        if      STATE_ACIN_CAN  >       0
        then    LV_ACCIN = 1
        else    LV_ACCIN = 0
        endif

        // control electric-fan air-conditioning:
        if      CTR_EFAN_AIC    =       ! 0FH
        then    N_ECF       =       CTR_EFAN_AIC
        else    N_ECF       =       0
        endif

        // control_air-conditioning_special-function / control_cooling-map-system:
        IF      CTR_AIC_SPFN    = 01H
        THEN    LV_REQ_HEAT     = 1
        ELSE    LV_REQ_HEAT     = 0
        ENDIF
```

```
IF      CTR_AIC_SPFN        = 02H           OR
        CTR_CLMP            = 01H
THEN    LV_REQ_TCO_L        = 1
ELSE    LV_REQ_TCO_L        = 0
ENDIF

// control air-conditioning readiness:
STATE_ACIN_CAN   =        CTR_AIC_RDI
IF              STATE_ACIN_CAN = 0
THEN    LV_ACIN = 0
ELSEIF          STATE_ACIN_CAN = 2H     OR
         STATE_ACIN_CAN = 4H
THEN    LV_ACIN = 1
ELSEIF          STATE_ACIN_CAN = 1H     AND
                LV_IS = 1
THEN    IF      LV_ACIN = 0
        THEN    LV_ACIN = 0
        ELSE    LV_ACIN = 1 until LV_IS <> 1
        ENDIF
ELSEIF          STATE_ACIN_CAN = 7H
THEN    freeze  value of LV_ACIN (until STATE_ACIN <> 7H)
ENDIF

//heat-flow_request_air-conditioning:
if      HTFL_RQ_AIC         =       FFh
then    HEAT_REQ_PERC       =       0
else    HEAT_REQ_PERC       =       HTFL_RQ_AIC
endif

//state_power_special-consumer:
if      ST_PWR_SPCOS        =       FFH
then    STATE_EXT_POW_CNS = 0
else    STATE_EXT_POW_CNS = ST_PWR_SPCOS
endif
```

```
          // control_fan_flap:
          LV_ECRAS_CTL_CAN_1 = Bit 0 of CTR_FAN_FLAP
          LV_ECRAS_CTL_CAN_2 = Bit 1 of CTR_FAN_FLAP

ELSE      // torque target air-conditioning-compressor:
              TQ_ACCIN _CAN    =    0H
              LV_ACCIN = 0

// control electric-fan air-conditioning:
N_ECF    =    0

// control_air-conditioning_special-function / control_cooling-map-system:
LV_REQ_HEAT   = 0
LV_REQ_TCO_L  = 0

// control air-conditioning readiness:
STATE_ACIN_CAN = 7H
LV_ACIN = 0

//heat-flow_request_air-conditioning:
HEAT_REQ_PERC       =    0

//state_power_special-consumer:
STATE_EXT_POW_CNS = 0

// control_fan_flap:
LV_ECRAS_CTL_CAN_1 = 1
LV_ECRAS_CTL_CAN_2 = 1
endif
```

<u>Not in use:</u>

The following messages are not in use by the IHKA. Therefore the values are initialised with FFh and the messages are not evaluated.

heat-flow_status_air-conditioning:

HTFL_ST_AIC = FFh

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TQ_ACCIN	1	0...FFH	0...127.5	0.5	[Nm]
Minimum torque ACCIN					

43.1.4.12 Kombi

Application conditions:

Initialisation: at reset: TAM = TIA

Formula section:

A_TEMP_RELATIVZEIT

Ambient temperature TAM:

```
If (1)          LC_TAM_MAN_AS   =   0
then (1)
if (2)                TEMP_EX =  FFH           OR  \
                      LV_ERR_BN_T_ICL = 1      OR   >     ('TAM-failure')
                      LV_ERR_CAN_BOFF = 1          /
then (2)
    if (3)   'TAM-failure' was not present the recurrence before
             then (3) $TAM_n$ = $TAM_{n-1}$                (last valid TAM remains)
                  LV_TAM_CAN_ERR =   1
             else (3)      TAM  = C_TAM_SUB           (TAM-failure present
                           LV_TAM_CAN_ERR =   1       since ECU initialisation)
             endif (3)
else (2)     LV_TAM_CAN_ERR = 0
             if (4)        LV_ERR_TAM_PLAUS = 1
             then (4)      TAM  = C_TAM_SUB
             else (4)      TAM   =     TEMP_EX                   (CAN)
             endif (4)
endif (2)
else (1)        TAM    =    C_TAM_MAN_AS     (application system)
endif (1)


IF           LV_ERR_BN_T_ICL = 0       AND
             LV_ERR_CAN_BOFF = 0
THEN         T_CTR_REL_CAN = T_SEC_COU_REL            // for E65 only
             T_CAN = T_DAY_COU_ABSL
```

```
ELSE      T_CTR_REL_CAN = T_REL_CAN  converted      // for E65 only
          T_CAN = FFFFH
ENDIF
```

BOS_RUECKSTELLUNG + BOS_RUECKSTELLUNG_2

//detection if messages were used and reset of idfbosrt_w and QOIL_DS_RST_CAN_2_2: this part has to be calculated every 100ms:

```
if        LV_ERR_CAN_BOFF = 0    and
          Id_bosrtak = idfbosrt_w and
          Id_bosrtak > 0
then      idfbosrt_w             = 0
          QOIL_DS_RST_CAN_2_2    = 0
endif
```

BOS_RUECKSTELLUNG

```
IF        LV_ERR_CAN_BOFF        = 0
THEN      if     new message received BOS_RUECKSTELLUNG
          then   idbosrst               = ID2_BOS_RSTG
                 idfbosrt_w             = ID_FN_BOS_RSTG
                 vbost                  = AVAI_BOS_RSTG
                 zrbosr                 = COU_RSTG_BOS_RSTG
                 QOIL_DS_RST_CAN_1_5    = RMMI_BOS_RSTG
          else   variables unchanged
          endif
ELSE      idbosrst               = FFH
          idfbosrt_w             = FFFFH
          vbost                  = FFH
          zrbosr                 = FFH
          QOIL_DS_RST_CAN_1_5    = 8000H
ENDIF
```

BOS_RUECKSTELLUNG_2

```
IF        LV_ERR_CAN_BOFF        = 0
THEN      if     new message received BOS_RUECKSTELLUNG_2
          then   QOIL_DS_RST_CAN_2_1    = ID2_BOS_RSTG_2
                 QOIL_DS_RST_CAN_2_2    = ID_FN_BOS_RSTG_2
```

```
                    QOIL_DS_RST_CAN_2_3   = FRC_INTM_WAY_BOS_RSTG_2
                    QOIL_DS_RST_CAN_2_4   = TARD_MON_BOS_RSTG_2
                    QOIL_DS_RST_CAN_2_5   = TARD_YR_BOS_RSTG_2
                    QOIL_DS_RST_CAN_2_6   = FRC_INTM_T_BOS_RSTG_2
            else    variables unchanged
            endif
ELSE        QOIL_DS_RST_CAN_2_1               = FFH
            QOIL_DS_RST_CAN_2_2               = FFFFH
            QOIL_DS_RST_CAN_2_3               = FFH
            QOIL_DS_RST_CAN_2_4               = FH
            QOIL_DS_RST_CAN_2_5               = FFH
            QOIL_DS_RST_CAN_2_6               = FFH
ENDIF
```

KILOMETERSTAND

```
IF          LV_ERR_BN_KM_ICL = 0            AND
            LV_ERR_CAN_BOFF = 0

THEN        // mileage_kilometre
if          MILE_KM = FFFFFFH
then        CTR_KM_CAN = DIST
elseif      MILE_KM < 655350 km (9FFF6H)
then        CTR_KM_CAN = MILE_KM
else        CTR_KM_CAN = 655340 km (FFFEH)
endif

// Kilometerstand – BN (1km-Auflösung)
if          MILE_KM < 16777215
then        CTR_KM_BN = MILE_KM
else        CTR_KM_BN = 16777214 (FFFFFFH)
endif

if   LV_IGK = 1
then
```

```
        // fluid-level_fuel-tank:
        if      FLLV_FUTA < 7FH
        then    FTL = FLLV_FUTA          and    LV_FTL_TOT_CAN_ERR = 0
                CTR_FTL_NOT_VLD = 0
        elseif  7FH ≤ FLLV_FUTA < FFH
        then    FTL = 126l               and    LV_FTL_TOT_CAN_ERR = 0
                CTR_FTL_NOT_VLD = 0
        elseif  FLLV_FUTA = FFH
        then    if      CTR_FTL_NOT_VLD >= C_CTR_FTL_NOT_VLD
                then    FTL = 0      and    LV_FTL_TOT_CAN_ERR  =  1
                else    CTR_FTL_NOT_VLD = ++
                        keep last value
                endif
        endif
    else        keep last value
    endif


    fluid-level_fuel-tank_left-hand:
    if    LV_IGK = 1
    then
        if      FLLV_FUTA_LH < 7FH
        then    FTL_LE = FLLV_FUTA_LH   and LV_FTL_LE_CAN_ERR = 0
                CTR_FTL_LE_NOT_VLD = 0
        elseif  7FH ≤ FLLV_FUTA_LE < FFH
        then    FTL_LE = 62l             and    LV_FTL_LE_CAN_ERR = 0
                CTR_FTL_LE_NOT_VLD = 0
        elseif  FLLV_FUTA_LE = FFH
        then    if      CTR_FTL_LE_NOT_VLD >= C_CTR_FTL_NOT_VLD
                then    FTL_LE = 0    and    LV_FTL_LE_CAN_ERR = 1
                else    CTR_FTL_LE_NOT_VLD = ++
                        keep last value
                endif
        endif
    else        keep last value
    endif


    fluid-level_fuel-tank_right-hand:
    if    LV_IGK = 1
    then
```

```
        if       FLLV_FUTA_RH < 7FH
        then     FTL_RI = FLLV_FUTA_RH   and    LV_FTL_RI_CAN_ERR = 0
                 CTR_FTL_RI_NOT_VLD = 0
        elseif   7FH ≤ FLLV_FUTA_RH < FFH
        then     FTL_RI = 62l            and    LV_FTL_RI_CAN_ERR = 0
                 CTR_FTL_RI_NOT_VLD = 0
        elseif   FLLV_FUTA_RH = FFH
        then     if      CTR_FTL_RI_NOT_VLD >= C_CTR_FTL_NOT_VLD
                 then    FTL_RI = 0    and    LV_FTL_RI_CAN_ERR = 1
                 else    CTR_FTL_RI_NOT_VLD = ++
                         keep last value
                 endif
        endif
    else         keep last value
    endif


    status fluid-level fuel-tank spare :
    if           ST_FLLV_FUTA_SPAR = 01b
                 or      ST_FLLV_FUTA_SPAR = 10b
                 or      ST_FLLV_FUTA_SPAR = 11b
    then         LV_FTL_OBD_INH_L = 1
    else         LV_FTL_OBD_INH_L = 0
    endif


    ELSE
        // mileage_kilometre
        CTR_KM_CAN = DIST


        // fluid-level_fuel-tank:
        FTL = 0              and    LV_FTL_CAN_ERR = 1
        LV_FTL_TOT_CAN_ERR = 1
        fluid-level_fuel-tank_left-hand:
        FTL_LE =  0          and    LV_FTL_LE_CAN_ERR = 1

        fluid-level_fuel-tank_right-hand:
        FTL_RI =  0          and    LV_FTL_RI_CAN_ERR = 1
```

```
        status fluid-level fuel-tank spare :
        LV_FTL_OBD_INH_L = 0
ENDIF


//setting of LV_FTL_LE_CAN_ERR in dependency of LV_ERR_xxx
If      LV_FTL_LE_CAN_ERR = 0  and
        [(LV_ERR_FTL_LE_CAN = 1  and  ERR_SYM_FTL_LE_CAN != SYM_3 and
          ERR_SYM_FTL_LE_CAN != NO_SYM)
        or (LV_ERR_TOUT_ICL_7 = 1 or LV_STATE_ERR_CAN_TOUT = 1 )]
Then    LV_FTL_LE_CAN_ERR = 1
Endif


//setting of LV_FTL_RI_CAN_ERR in dependency of LV_ERR_xxx
If      LV_FTL_RI_CAN_ERR = 0  and
        [(LV_ERR_FTL_RI_CAN = 1  and  ERR_SYM_FTL_RI_CAN != SYM_3 and
          ERR_SYM_FTL_RI_CAN != NO_SYM)
        or (LV_ERR_TOUT_ICL_7 = 1 or LV_STATE_ERR_CAN_TOUT = 1 )]
Then    LV_FTL_RI_CAN_ERR = 1
Endif


If      LV_FTL_TOT_CAN_ERR = 1      or
        LV_FTL_LE_CAN_ERR = 1       or
        LV_FTL_RI_CAN_ERR = 1
Then    LV_FTL_CAN_ERR = 1
Else    LV_FTL_CAN_ERR = 0
Endif
```

STAT_KOMBI

```
IF      LV_ERR_BN_ICL = 0           AND
        LV_ERR_CAN_BOFF = 0
THEN    // speed_vehicle_speedometer:
VS_ICL_DISP     =     V_VEH_SPDM

// status_unit-of-measurement_speed_speedometer:
STATE_VS_ICL_DISP     =     ST_UN_V_SPDM
```

```
// status_interface_cruise-control:
STATE_IF_ICL_BN_MSW        =       ST_INTF_CCTRA_KI

// alive_ICL:
TCC_CAN_ICL    =       ALIV_COU_KI

STATE_WAL_CAN        =       ST_WALI_ENG
ELSE       // speed_vehicle_speedometer:
VS_ICL_DISP      =       FFFH

// status_unit-of-measurement_speed_speedometer:
STATE_VS_ICL_DISP    =       3H

// status_interface_cruise-control:
STATE_IF_ICL_BN_MSW        =       1H

// alive_ICL:
TCC_CAN_ICL    =       FH

STATE_WAL_CAN        =       3H
```

UHRZEIT_DATUM

```
IF          LV_ERR_BN_T_CLK = 0              AND
            LV_ERR_CAN_BOFF = 0
THEN        // display day:
            IF     0 < DISP_DATE_DAY < 32
            THEN   T_CLK_ICL_DISP_1 = DISP_DATE_DAY
            ELSE   T_CLK_ICL_DISP_1 = 255
            ENDIF
            // display month:
            IF     0 < DISP_DATE_MON < 13
            THEN   T_CLK_ICL_DISP_2 = DISP_DATE_DAY
            ELSE   T_CLK_ICL_DISP_2 = 255
            ENDIF
            // display year:
```

```
        IF      0 < DISP_DATE_YR < 65535
        THEN    T_CLK_ICL_DISP_3 = DISP_DATE_YR
        ELSE    T_CLK_ICL_DISP_3 = 65535
        ENDIF

ELSE    T_CLK_ICL_DISP_1 = 255
        T_CLK_ICL_DISP_2 = 255
        T_CLK_ICL_DISP_3 = 65535
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TAM_MAN_AS	1	0...1H	0...1	1	[-]
Logical constant switch TAM from CAN or application system					
C_TAM_MAN_AS	1	0...FEH	-48...142.5	0.75	[°C]
Ambient temperature TAM by application system					
C_TAM_SUB	1	0...FEH	-48...142.5	0.75	[°C]
Substitute TAM-temperature in case of TAM-failure					
C_CTR_FTL_NOT_VLD	1	0...FFH	0...255	1	[-]
Counter threshold for messages with invalid FTL_XX-value					

43.1.4.13 LDM

Formula section:

Anforderung Radmoment Antriebsstrang:

```
IF      LV_ERR_BN_LDM          = 0          AND
        LV_ERR_CAN_BOFF        = 0

THEN    // checksum LDM:
        CKS_CAN_LDM      =      CHKSM_RQ_WMOM_PT
        // alive LDM:
        TCC_CAN_LDM      =      ALIV_RQ_WMOM_PT
        // follow-dynamic_wheel-moment_power-train_target:
        STATE_SP_DYN_WHEEL  =  FOLDYN_WMOM_PT_TAR  (Bit 0 ... Bit 3)
        // request_prevention_inertia-fuel-shutoff:
        STATE_DI_PUC     =      FOLDYN_WMOM_PT_TAR (Bit 0)
        // wheel-moment_power-train_target:
        If      WMOM_PT_TAR = FFFFH
        Then    LV_TQ_WHEEL_LDM_BN_ERR = 1
                TQ_WHEEL_LDM_BN       =       - 32000 Nm
```

```
          Else    LV_TQ_WHEEL_LDM_BN_ERR = 0
                  if      WMOM_PT_TAR  <= 32000 Nm
                  then    TQ_WHEEL_LDM_BN       =       WMOM_PT_TAR
                  else    TQ_WHEEL_LDM_BN       =       32000 Nm
                  endif
          Endif
          // status_driver-assistence-system_powertrain:
          STATE_LDM  =        ST_DRASY_PT
          // Wheel-moment power-train enable:
          STATE_LDM_INTV   =        WMOM_PT_ENBL
          If      STATE_LDM_INTV = 0
          Then    LV_TQ_WHEEL_LDM_REQ = 0
                  LV_LDM_LIH_CAN = 0
          Elseif  STATE_LDM_INTV = 1
          Then    LV_TQ_WHEEL_LDM_REQ = 1
                  LV_LDM_LIH_CAN = 0
          Else    LV_TQ_WHEEL_LDM_REQ = 0
                  LV_LDM_LIH_CAN = 1
          Endif

          //request gear shift indicator for LDM:
          If      RQ_GSI_CLCTR_LN = 1
          Then    LV_GS_IDC_LDM = 1
          Else    LV_GS_IDC_LDM = 0
          Endif

ELSE      // checksum LDM:
          CKS_CAN_LDM       =       FFh
          // alive LDM:
          TCC_CAN_LDM       =       Fh
          // follow-dynamic_wheel-moment_power-train_target:
          STATE_SP_DYN_WHEEL    =       Fh
          // wheel-moment_power-train_target:
          TQ_WHEEL_LDM_BN         =       -32000 Nm
          LV_TQ_WHEEL_LDM_BN_ERR = 0
          // Wheel-moment power-train enable:
```

```
STATE_LDM_INTV   =        3H
LV_TQ_WHEEL_LDM_REQ = 0
LV_LDM_LIH_CAN = 1
// request_prevention_inertia-fuel-shutoff:
STATE_DI_PUC       =        0H (nur Auswertung Bit 3)
// status_driver-assistence-system_powertrain:
STATE_LDM  =        FH
// auxiliary bytes for LDM test required by customer:
STATE_CUS_LDM_CAN =   FFFFh
//request gear shift indicator for LDM:
LV_GS_IDC_LDM = 0

ENDIF
```

43.1.4.14 Power-Modul

__Application conditions:__

Initialisation: V_GEN_TAR = C_V_SP_GEN (valid for BN2000 **and** CAN11H)

__Formula section:__

POWERMGMT_BATTERIESPANNUNG

status_battery-main-switch:

The signal cannot be evaluated as it is transmitted only after transition KL30 ON → OFF.

status_switch-off_klemme_30:

STATE_SWI_OFF_VB = ST_SWO_KL_30

The signal is only to be read into the internal variable STATE_SWI_OFF_VB but not evaluated.

shutdown counter battery-main-switch :

```
if      LV_IGK = 0 and VS = 0 and LV_ES = 1 (“engine stopped”)
then    if      SDWN_COU_BTSW = FFH
then    T_CTR_SWI_OFF_ VB = 254
else    T_CTR_SWI_OFF_ VB = SDWN_COU_BTSW
                if      T_CTR_SWI_OFF_VB ≤ C_T_CTR_SWI_OFF_VB_MIN  and
                        PWL is active (before NM)
```

```
then        "shut down operating system"
endif
endif
else        T_CTR_SWI_OFF_VB = 254
endif
```

time second counter relative Redundant :

```
IF          LV_ERR_CAN_BOFF = 0
THEN        IF              LV_VAR_VEH = 1              // E65
            THEN IF         LV_ERR_BN_POW_VB = 0
                 THEN       T_REL_CAN =  T_SEC_COU_REL_Redundant ( converted)
                            //from Powermodul, message
POWERMGMT_BATTERIESPANNUNG
                            T_REL_CAN_2 = T_SEC_COU_REL_Redundant
                            //setting bit indicating t_rel_can received:
                            LV_T_REL_CAN_REG = 1
                 ELSEIF           LV_ERR_BN_T_ICL = 0
                            THEN  T_REL_CAN = T_SEC_COU_REL (converted)
                                  T_REL_CAN_2 = T_SEC_COU_REL
                            ELSE  T_REL_CAN = 0
                                  T_REL_CAN_2 = 0
                 ENDIF
            ELSEIF          LV_VAR_VEH = 0              // E60
            THEN IF         LV_ERR_BN_T_ICL = 0
                 THEN       T_REL_CAN = T_SEC_COU_REL (converted)
                            T_REL_CAN_2 = T_SEC_COU_REL
                            // from Kombi, message A_TEMP_RELATIVZEIT
                            //setting bit indicating t_rel_can received:
                            LV_T_REL_CAN_REG = 1
                 ELSE       T_REL_CAN = 0
                            T_REL_CAN_2 = 0
                 ENDIF
            ENDIF
ELSE        T_REL_CAN = 0
            T_REL_CAN_2 = 0
ENDIF
```

POWERMGMT_LADESPANNUNG

```
IF        LV_ERR_BN_POW_GEN = 0       AND
          LV_ERR_CAN_BOFF = 0
THEN      // Control_generator_voltage_target:
if   CTR_GEN_U_TAR = FFH
then      V_GEN_TAR = C_V_SP_GEN
else              V_GEN_TAR = CTR_GEN_U_TAR ( converted)
endif

RPM_idling_specification:
if   RPM_IDLG_SPEC = 3H
then      FAC_IS_INC_REQ = 0
else      FAC_IS_INC_REQ = RPM_IDLG_SPEC
endif

ELSE      // Control_generator_voltage_target:
V_GEN_TAR = C_V_SP_GEN

RPM_idling_specification:
FAC_IS_INC_REQ = 0
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_SP_GEN	1	0...FFH	10.6...16.975	0.025	[V]
Generator voltage (substitute value in case of invalid signal from CAN)					
C_T_CTR_SWI_OFF_VB_MIN	1	0...FFH	0...255	1	[s]
Threshold shutdown counter battery-main-switch					

43.1.4.15 SZ-Lenksäule

Formula section:

LENKRADWINKEL (Sender SZL/E65, DSC/E60)

```
IF        ( LV_IGK = 0 AND LV_ES = 1 )                        OR
```

```
          (LV_IGK = 1 AND T_WAKE_UP_ON < C_T_DLY_ERR_PSTE)   OR
          (ECU_STATE = 'Wake_up' AND T_PWL < C_T_DLY_ERR_PSTE)
THEN      ANG_PSTE = 0°
          LV_ERR_ANG_PSTE_CAN = 0
          STATE_SENS_ANG_PSTE = 0H
          VEL_ANG_PSTE  = 0°/s
ELSE      IF          LV_ERR_BN_ANG_PSTE = 0 AND
                      LV_ERR_CAN_BOFF = 0
          THEN        IF     STWA_ERR = 1
                      THEN   ANG_PSTE = 0°
                             VEL_ANG_PSTE = 0°/s
                             LV_ERR_ANG_PSTE_CAN = 0
                             STATE_SENS_ANG_PSTE = 0H
                      ELSE   IF     STWA = 8000H  or  STWA_ERR >= 1
                             THEN   ANG_PSTE = 0
                                    LV_ERR_ANG_PSTE_CAN = 1
                                    VEL_ANG_PSTE = 0°/s
                             ELSE   ANG_PSTE = STWA
                                    LV_ERR_ANG_PSTE_CAN = 0
                                    VEL_ANG_PSTE = STWA_V
                             ENDIF
                      STATE_SENS_ANG_PSTE = STWA_ERR
                      ENDIF
            ELSE      ANG_PSTE = 0°
                      LV_ERR_ANG_PSTE_CAN = 1
                      STATE_SENS_ANG_PSTE = 3H
                      VEL_ANG_PSTE = 0°/s
            ENDIF
ENDIF
```

BEDIENUNG_TEMPOMAT

```
IF        LV_ERR_BN_MSW         = 0          AND
          LV_ERR_CAN_BOFF       = 0
THEN      CKS_CAN_MSW = CHKSM_CCTR
```

```
            TCC_CAN_MSW = ALIV_CCTR

IF          OP_PUBU_CCTR_ACC    = 00H
THEN        STATE_BN_MSW        = NO_ACTION
ELSEIF      OP_PUBU_CCTR_ACC    = 10H           OR
            OP_PUBU_CCTR_ACC    = 50H
THEN        STATE_BN_MSW        = TIP UP or DOWN            // OFF
ELSEIF      OP_PUBU_CCTR_ACC    = 04H           OR
            OP_PUBU_CCTR_ACC    = 44H
THEN        STATE_BN_MSW        = TIP_POST
ELSEIF      OP_PUBU_CCTR_ACC    = 08H
            OP_PUBU_CCTR_ACC    = 48H
THEN        STATE_BN_MSW        = PRESS_POST
ELSEIF      OP_PUBU_CCTR_ACC    = 40H           OR
            OP_PUBU_CCTR_ACC    = 41H           OR
            OP_PUBU_CCTR_ACC    = 42H
THEN        STATE_BN_MSW        = TIP_AXIAL
ELSEIF      OP_PUBU_CCTR_ACC    = 01H
THEN        STATE_BN_MSW        = TIP_PRE
ELSEIF      OP_PUBU_CCTR_ACC    = 02H
THEN        STATE_BN_MSW        = PRESS_PRE
ELSEIF      OP_PUBU_CCTR_ACC    = 7F
THEN        STATE_BN_MSW        = INVALID SIGNAL
ENDIF

STATE_GAP_MSW = OP_GAPC_ACC // this variable will only be used for the checksum-calculation of the message “Bedienung Tempomat”

ELSE        CKS_CAN_MSW     = FFH
            TCC_CAN_MSW     = FH
            STATE_BN_MSW    = 0H
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_ERR_PSTE	1	0...FFH	0...2.55	0.01	[s]
Time delay threshold to suppress error steering angle					

43.1.4.16 K_CAN: LM

Formula section:

STATUS_GANG_RÜCKWÄRTS

```
IF      LV_ERR_CAN_BOFF = 0    and
        LV_ERR_BN_GEAR_REV = 0
THEN    if      ST_GR_BAC = 2H
then    STATE_GEAR_REV_CAN = 1
else    STATE_GEAR_REV_CAN = 0
endif
ELSE    STATE_GEAR_REV_CAN = 0
ENDIF
```

LAMPENZUSTAND

```
IF      LV_ERR_CAN_BOFF = 0    and
        LV_ERR_BN_ LTG_HDLP_L = 0
THEN    if      ST_DIPB = 1
then    LV_LTG_HDLP_L_ON = 1
else    LV_LTG_HDLP_L_ON = 0
endif
ELSE    LV_LTG_HDLP_L_ON = 0
ENDIF
```

CTR_LICHT

```
IF      LV_ERR_CAN_BOFF = 0
THEN    if      CTR_ITLI = 1
then    LV_LTG_INL_ON = 1
else    LV_LTG_INL_ON = 0
endif
ELSE    LV_LTG_INL_ON = 0
ENDIF
```

STAT_ANHÄNGER

```
IF      LV_ERR_CAN_BOFF = 0   AND
        LV_ERR_BN_TRL = 0     AND
        ST_TRAI < 2
THEN    LV_STATE_TRL = ST_TRAI
ELSE    LV_STATE_TRL = 0
ENDIF
```

STAT_VERDECK_CABRIO* (27EH *)

General information:

The state of the cabrio top must be available in the ECU For handling of the sound flap.

Application conditions:

Initialisation: at reset LV_CTOP = 0

Recurrence: message received event triggered (tmin=100ms)

Activation: every engine state

Formula section:

```
IF      LV_ERR_CAN_BOFF = 0
THEN    setting LV_CTOP:
        If      ST_PO_FLDT_CAB = 0
                OR ST_PO_FLDT_CAB = 4
                OR ST_PO_FLDT_CAB = 7
        Then    LV_CTOP = 0
        Elseif  ST_PO_FLDT_CAB = 1
                OR  ST_PO_FLDT_CAB = 2
                OR  ST_PO_FLDT_CAB = 3
        Then    LV_CTOP = 1
        Endif
ELSE    setting default value:
        LV_CTOP = 0
ENDIF
```

43.1.4.17 EKP

General information:

In case of LV_VAR_BN_EFP = 1 (e.g. LEVII-vehicles) the EKP sends its data concerning the DME interface on the PT-CAN, message ID = 335h.

Application conditions:

Initialisation: at reset and LV_IGK = 1:

LV_EFP_ON_EXT_ADJ_CAN = 0H

LV_EFP_OFF_EXT_ADJ_CAN = 0H

STATE_DIAG_EFP = FFH

STATE_EFP_CAN = 0H

V_ENG_EFP = 5V

CUR_ENG_EFP = 0A

N_EFP_AV = 0 rpm

VB_EFP = 5V

STATE_EFP_CTL = 3H

Recurrence: 1 s **or** Event controlled

Activation: *LV_IGK = 1 and LV_VAR_BN_EFP = 1*

Formula section:

```
If   LV_ERR_BN_EFP = 0          AND
     LV_ERR_CAN_BOFF = 0
Then        STATE_DIAG_EFP   =      ST_DIAG_FU_PP_EL
            STATE_EFP_CAN    =      ST_FU_PP_EL
            V_ENG_EFP        =      U_ENG_FU_PP_EL
            CUR_ENG_EFP      =      I_ENG_FU_PP_EL
            N_EFP_AV         =      RPM_AVL_FU_PP_EL
            VB_EFP           =      U_KL_30_FU_PP_EL
            STATE_EFP_CTL    =      ST_FU_CLCTR_FU_PP_EL
Else        STATE_DIAG_EFP   =      FFH
            STATE_EFP_CAN    =      0H
            V_ENG_EFP        =      5V
            CUR_ENG_EFP      =      0A
            N_EFP_AV         =      0 rpm
            VB_EFP           =      5V
            STATE_EFP_CTL    =      3H
Endif

If   STATE_EFP_CAN = 1
     Then   LV_EFP_ON_EXT_ADJ_CAN         = 1
LV_EFP_OFF_EXT_ADJ_CAN   = 0
     Else if        STATE_EFP_CAN    = 2
            Then    LV_EFP_ON_EXT_ADJ_CAN         = 0
LV_EFP_OFF_EXT_ADJ_CAN   = 1
            Else    LV_EFP_ON_EXT_ADJ_CAN         = 0
LV_EFP_OFF_EXT_ADJ_CAN   = 0
Endif
Endif
```

43.1.4.18 K_CAN: CCC_GW, M_ASK

General information:

Application conditions:

Initialisation:

Recurrence: event controlled

Activation: LV_IGK = 1

Formula section:

Programmierung Stufentempomat:

handshake with stepped cruise-control function:

```
IF      message "Programmierung Stufentempomat" has been received
THEN    LV_MSG_PROG_STEP_CRU = 1
input for stepped cruise control-function
resetting of bit through stepped cruise control function
ENDIF
```

index_number_programming_cruise-control

NR_VS_CRU_IF = IDX_NO_PRGG_CCTR

speed_programming_cruise-control_index_number_1

VS_STEP_IF_1 = V_PRGG_CCTR_IDX_NO_1

speed_programming_cruise-control_index_number_2

VS_STEP_IF_2 = V_PRGG_CCTR_IDX_NO_2

speed_programming_cruise-control_index_number_3

VS_STEP_IF_3 = V_PRGG_CCTR_IDX_NO_3

43.1.4.19 K_CAN/PT-CAN

Application conditions:

Initialisation: at LV_IGK 0 --> 1:
FAC_POW_MNG_VST_CNS (index 0) = 255,
FAC_POW_MNG_VST_CNS (index i) = 0
at reset N_ECF_CAN = 0 %

N_ECF_CAN_2 = 0 %

ECF_REQ_EXT = 0

Activation: Powermanagement_Standverbraucher : LV_IGK = 1

Anforderung_E-Lüfter: bus_active

DIENSTENACHRICHTEN:

Powermanagement_Standverbraucher

```
IF      ID2 = 08H
THEN    “Dienstenachricht” Powermanagement_Standverbraucher
ENDIF

IF              ST_PCOS_PWMG = ON              // “Status Standverbraucher PM”
THEN    IF      FAC_POW_MNG_VST_CNS (index 0) = 255
        THEN    FAC_POW_MNG_VST_CNS (index 0) = SV-ID
        ELSE    IF      FAC_POW_MNG_VST_CNS (index i) = SV-ID
                THEN    no action              // “SV already in list”
                ENDIF
        ELSE    IF      FAC_POW_MNG_VST_CNS (index i) = 0
                THEN    FAC_POW_MNG_VST_CNS (index i) = SV-ID
                        // ”free position found”
                ENDIF
        ENDIF
ELSE    IF       ST_PCOS_PWMG = OFF            // “Status Standverbraucher PM”
        THEN    IF      FAC_POW_MNG_VST_CNS (index i) = SV-ID
                THEN    FAC_POW_MNG_VST_CNS (index i) = 0          //  ”SV  found, delete from list”
                ENDIF
                IF      FAC_POW_MNG_VST_CNS (index i) = 0
                THEN    FAC_POW_MNG_VST_CNS (index 0) = 255
                ENDIF
        ENDIF
ENDIF
```

Standverbraucher (SV)	**SV-ID (ID_FN_PWMG)**
Anhängerlicht	1
Bordmonitor (MMI)	2
Diebstahlwarnanlage	3
Licht	4

Standheizen	5
Standlüften	6
Getriebe	7
Restwärme	8
Telefon (ECE)	9
Telefon (USA)	10
Standklimatisierung	11
APU	12
Frontscheibenheizung	13

Verbraucherstatus

```
IF      ID2 = 6EH
THEN    "Dienstenachricht" Verbraucherstatus
ENDIF
```

STATE_VEH_CNS[10] = ID_FN_COS

STATE_VEH_CNS_FCT = ST_FN_COS

Bit 0	Bit 1	Status
0	0	Verbraucherfunktion aus
0	1	Verbraucherfunktion ein
1	0	Ungültiger Zustand
1	1	Signal ungültig

43.2 Byte	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Byte 1	V0	V0	V1	V1	V2	V2	V3	V3
Byte 2	V4	V4	V5	V5	V6	V6	V7	V7
Byte 3	V8	V8	V9	V9	V10	V10	V11	V11
Byte 4	V12	V12	V13	V13	V14	V14	V15	V15
Byte 5	V16	V16	V17	V17	V18	V18	V19	V19
Byte 6	V20	V20	V21	V21	V22	V22	V23	V23
Byte 7	V24	V24	V25	V25	V26	V26	V27	V27
Byte 8	V28	V28	V29	V29	V30	V30	V31	V31
Byte 9	V32	V32	V33	V33	V34	V34	V35	V35
Byte 10	V36	V36	V37	V37	V38	V38	V39	V39

Anforderung_E-Lüfter

```
IF          ID2_RQ_STG_EFAN = 15h
            // Dienstenachricht Anforderung E-Lüfter
THEN        LV_N_ECF_CAN_ACT = 1
            //Dienstenachricht E-Lüfter zum 1. Mal angefordert
            LV_N_ECF_CAN_RCV = 1
            //Dienstenachricht zyklisch erhalten (1s-Raster)
            IF  ((ID_FN_RQ_STG_EFAN != FFH) AND (RQ_STG_EFAN != FH))
            THEN N_ECF_CAN_2  = RQ_STG_EFAN //conversion from CAN
                 ECF_REQ_EXT = ID_FN_RQ_STG_EFAN
                 //anforderndes Steuergerät
                 IF     RQ_STG_EFAN = 0
                 THEN   LV_N_ECF_CAN_ACT = 0
                 //Anforderung E-Lüfter zurückgenommen
                 ENDIF
            ELSE If     (ID_FN_RQ_STG_EFAN != FFH)
                 Then   N_ECF_CAN_2  = C_N_ECF_CAN
                 Else   N_ECF_CAN_2  = 0
                 Endif
                 ECF_REQ_EXT = 0
            ENDIF
ENDIF
```

Monitoring of Anforderung_E-Lüfter (cyclic transmission, 1s):

```
IF          LV_N_ECF_CAN_ACT = 1
```

```
                // Dienstenachricht Anforderung E-Lüfter wird angefordert
THEN            IF      LV_N_ECF_CAN_RCV = 1
                //Dienstenachricht E-Lüfter wird zyklisch erhalten
                THEN    CTR_TOUT_N_ECF_CAN = 0
                        LV_N_ECF_CAN_RCV = 0
                        ELSE IF     CTR_TOUT_N_ECF_CAN>=C_
        CTR_TOUT_N_ECF_CAN
                                THEN N_ECF_CAN_2 = C_N_ECF_CAN
                                ECF_REQ_EXT = 0
                        ELSE    CTR_TOUT_N_ECF_CAN  ++
                        ENDIF
                ENDIF
ENDIF
```

//setting of N_ECF_CAN (add. check if component protection neccessary): This part has to be calculated with recurrence of 1s (independent on message received or not)

```
If          [LV_ERR_CAN_BOFF = 1  or  LV_ERR_BN_TQ_PSTE_3 = 1]   and
            [LV_VAR_ECRAS_UP = 1  or  LV_VAR_ECRAS_DOWN = 1]
Then        // setting of min value for component protection:
            N_ECF_CAN = MAX(N_ECF_CAN_2,C_N_ECF_CAN)
Else        // setting value like requested by CAN message (Anforderung E-Lüfter):
            N_ECF_CAN = N_ECF_CAN_2
Endif
```

OBD Sensor Diagnosestatus (condition based service)

Application conditions:

Initialisation: ODB_TAM = TAM (has to be initialised after TAM initialisation),
all other = 0

Recurrence: 10s

Activation: LV_IGK = 1

Formula section:

```
IF   ID2 = 8CH      //Dienstenachricht OBD Sensor Diagnosestatus
THEN
     LV_STATE_ERR_CAN_ACT = 1
```

```
//Dienstenachricht OBD Sensor Diagnosestatus zum 1. Mal angefordert
LV_STATE_ERR_CAN_RCV = 1
//Dienstenachricht zyklisch erhalten (10s-Raster)
If      ID_OBD_SEN = 0H
Then    // get symptom of TAM sensor
        if      ERR_OBD_SEN_1 = 1H
        then    // short circuit to ground
                STATE_ERR_SYM_TAM_CAN = SCG
        elseif  ERR_OBD_SEN_1 = 2H
        then    // short circuit to vbatt (plus) or open line (can't be distinguished)
                STATE_ERR_SYM_TAM_CAN = SCP
        else    STATE_ERR_SYM_TAM_CAN = NO_SYM
        endif
        // get raw value of TAM for OBD:
        OBD_TAM = RWDT_OBD_SEN_1
                  (converted.  -> OBD_TAM = [hex] / 2 – 40°C )
        // TAM value received:
        LV_OBD_TAM_RCV = 1

Elseif  ID_OBD_SEN = 1H
Then    // get symptom of FTL_RI sensor
        if      ERR_OBD_SEN_2 = 1H
        then    // short circuit to ground
                STATE_ERR_SYM_FTL_RI_CAN = SCG
        elseif  ERR_OBD_SEN_2 = 2H
        then    // short circuit to vbatt (plus) or open line (can't be distinguished)
                STATE_ERR_SYM_FTL_RI_CAN = SCP
        else    STATE_ERR_SYM_FTL_RI_CAN = NO_SYM
        endif
        // get symptom of FTL_LE sensor
        if      ERR_OBD_SEN_1 = 1H
        then    // short circuit to ground
                STATE_ERR_SYM_FTL_LE_CAN = SCG
        elseif  ERR_OBD_SEN_1 = 2H
        then    // short circuit to vbatt (plus) or open line (can't be distinguished)
                STATE_ERR_SYM_FTL_LE_CAN = SCP
```

```
            else    STATE_ERR_SYM_FTL_LE_CAN = NO_SYM
            endif
        Else    no changes
        Endif
ENDIF
```

Monitoring of Dienstenachricht OBD Diagnosestatus (cyclic transmission, 10s):

```
IF      LV_STATE_ERR_CAN_ACT = 1
        // Dienstenachricht OBD Diagnosestatus wird angefordert
THEN    IF      LV_STATE_ERR_CAN_RCV = 1
                //Dienstenachricht OBD Diagnosestatus wird zyklisch erhalten
        THEN    CTR_STATE_ERR_CAN = 0
                LV_STATE_ERR_CAN_RCV = 0
                LV_STATE_ERR_CAN_TOUT = 0
        ELSE    IF      CTR_STATE_ERR_CAN>=C_CTR_TOUT_STATE_ERR_CAN
                THEN    LV_STATE_ERR_CAN_TOUT = 1
                        STATE_ERR_SYM_FTL_LE_CAN = NO_SYM
                        STATE_ERR_SYM_FTL_RI_CAN = NO_SYM
                        STATE_ERR_SYM_TAM_CAN = NO_SYM
                ELSE    CTR_STATE_ERR_CAN  ++
                ENDIF
        ENDIF
ENDIF
```

Application conditions:

Recurrence: 1s

Activation: at every engine state

//setting of OBD_TAM if message not received yet (after reset a time gap up to 10s until first receive of message is possible). This part has to be calculated with recurrence of 1s (independent on message received or not):

```
If      LV_OBD_TAM_RCV = 0
Then    // setting of OBD_TAM = TAM for initialisation:
        OBD_TAM = TAM     (converted)
Else    do nothing
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_TOUT_N_ECF_CAN	1	0...FFH	0...255	1	[-]
Threshold for counter timeout-detection ext. ECF-request					
C_N_ECF_CAN	1	0...FFH	0...99.60937	0.390625	[%]
Default value for ECF-request					
C_CTR_TOUT_STATE_ERR_CAN	1	0...FFH	0...255	1	[-]
Threshold for counter timeout-detection of OBD diagnosis service message					

43.2.1.1 HB3 – DREHMOMENT_ANF_STE

Formula section:

DREHMOMENT_ANF_STE:

```
IF          LV_ERR_BN_TQ_PSTE_3 = 0       AND
            LV_ERR_CAN_BOFF = 0
THEN

// checksum_torque_STE:
CKS_CAN_TQ_ PSTE_3 = CHKSM_TORQ_STE

// alive_torque_STE:
TCC_CAN_TQ_ PSTE_3 = ALIV_TORQ_STE

// source torque_STE
if    SRC_RQ_TORQ_STE = 0H
then        STATE_PSTE_3_SRC = NORMAL_STEERING
elseif      SRC_RQ_TORQ_STE = 1H
then        STATE_PSTE_3_SRC = AFS
elseif      SRC_RQ_TORQ_STE = 2H
then        STATE_PSTE_3_SRC = EPS
elseif      SRC_RQ_TORQ_STE = 4H
then        STATE_PSTE_3_SRC = RESERVED
elseif      SRC_RQ_TORQ_STE = FH
then        STATE_PSTE_3_SRC = INVALID_SIGNAL
endif

// actual value torque loss STE :
if          TORQ_AVL_STE = FFH
then        TQ_LOSS_PSTE_3_AV_CAN = 8000H
else        TQ_LOSS_PSTE_3_AV_CAN = TORQ_AVL_STE (converted from BN 2000)
endif

// target value torque loss STE
if          TORQ_TAR_STE = FFH
then        TQ_LOSS_PSTE_3_SP_CAN = 8000H
else        TQ_LOSS_PSTE_3_SP_CAN = TORQ_TAR_STE (converted from BN 2000)
endif

// state STE-intervention:
```

```
STATE_PSTE_3_INTV = ST_TORQ_TAR_STE

if    STATE_PSTE_3_INTV = 02H
then        LV_PSTE_3_INTV            = 1
else        LV_PSTE_3_INTV            = 0
endif

// state torque_STE_error
STATE_PSTE_3_ERR = ST_RQ_TORQ_STE_ERR

if          STATE_PSTE_3_ERR = 00H
then        LV_TQ_PSTE_3_CAN_ENA = 1
else        LV_TQ_PSTE_3_CAN_ENA = 0
endif

if          STATE_PSTE_3_ERR = 01H or
            STATE_PSTE_3_ERR = 02H or
            STATE_PSTE_3_ERR = 04H
then        LV_TQ_PSTE_3_CAN_LIH = 1
else        LV_TQ_PSTE_3_CAN_LIH = 0
endif

if          STATE_PSTE_3_ERR = 03H or
            STATE_PSTE_3_ERR = 05H or
            STATE_PSTE_3_ERR = 06H or
            STATE_PSTE_3_ERR = 07H
then        LV_TQ_PSTE_3_CAN_DI = 1
else        LV_TQ_PSTE_3_CAN_DI = 0
endif


ELSE        // substitute values_STE:
// checksum_torque_STE
CKS_CAN_TQ_PSTE_3  = FFH

// alive_torque_STE:
TCC_CAN_TQ_PSTE_3  = FH

// source torque_STE
STATE_PSTE_3_SRC     = 4H

// actual value torque loss STE :
TQ_LOSS_PSTE_3_AV_CAN = 8000H

// target value torque loss STE :
TQ_LOSS_PSTE_3_SP_CAN = 8000H

// state STE-intervention:
STATE_PSTE_3_INTV    = 3H
LV_PSTE_3_INTV          = 0H

// state torque_STE_error
```

```
    STATE_PSTE_3_ERR            = 7H
    LV_TQ_PSTE_3_CAN_ENA        = 0H
    LV_TQ_PSTE_3_CAN_LIH        = 0H
    LV_TQ_PSTE_3_CAN_DI         = 1H

ENDIF
```

43.2.1.2 EMF Modul (PBR) electromechanic park brake

STELLANF_EMF

Formula section:

```
IF          LV_ERR_BN_REQ_PBR = 0                  AND
            LV_ERR_CAN_BOFF = 0
THEN
            // checksum:
            CKS_CAN_REQ_PBR = CHKSM_PRQ_EMF
            // alive counter:
            TCC_CAN_REQ_PBR = ALIV_PRQ_EMF
            // status indicating holding of vehicle with parking brake
            STATE_HLD_PBR = ST_HYD_RETA_2
            // comfort mode of parking brake
            If      ST_CFFU_EMF_2 < 2H
            Then    LV_CFT_MOD_PBR = ST_CFFU_EMF_2
            Else    LV_CFT_MOD_PBR = 0
            Endif
ELSE        // substitute values:

            // checksum
            CKS_CAN_REQ_PBR         = FFH
            // alive_counter:
            TCC_CAN_REQ_PBR         = FH
            // status indicating holding of vehicle with parking brake
            STATE_HLD_PBR = 3H
            // comfort mode of parking brake
            LV_CFT_MOD_PBR = 0
ENDIF
```

STATUS_EMF

Formula section:

```
IF      LV_ERR_BN_PBR = 0           AND
        LV_ERR_CAN_BOFF = 0
THEN
        // checksum:
        CKS_CAN_PBR      = CHKSM_ST_EMF
        // alive counter:
        TCC_CAN_PBR      = ALIV_COU_EMF_2
        //status of parking brake
        STATE_PBR = ST_EMF_LOCA_2

ELSE    // substitute values
        // checksum
        CKS_CAN_PBR      = FFH
        // alive_counter:
        TCC_CAN_PBR      = FH
        //status of parking brake
        STATE_PBR = FH
ENDIF
```

ST_RQ_EMF (Status Anforderung EMF)   (ID =1FDH)

Application conditions:

Initialisation: at reset all 0

Recurrence: 40 ms (tmin=36 ms, tmax=44 ms)

Activation: LV_VAR_TQ_PBR = 1

Formula section:

```
IF      LV_ERR_CAN_BOFF = 0 AND
        LV_ERR_BN_TQ_PBR = 0
THEN
        // Checksum:
        CKS_CAN_TQ_PBR              = CHKSM_ST_RQ_EMF
```

```
        // Alive-Counter:
        TCC_CAN_TQ_PBR                 = ALIV_ST_RQ_EMF
        // status actuator EMF:
        STATE_PBR_ACT                  = ST_ACT_EMF
        STATE_PBR_ACT_QLY              = QU_ST_ACT_EMF
        // request deceleration ~~wheel moment~~ EMF:
        ~~TQ_WHEEL_PBR                   = RQ_WMOM_EMF   (converted)~~
        AC_WHEEL_PBR                   = RQ_DCRN_EMF      (converted)
        STATE_TQ_WHEEL_PBR_QLY         = QU_RQ_DCRN    ~~QU_RQ_WMOM_EMF~~
ELSE    //setting default values:
        STATE_PBR_ACT                  = FH
        STATE_PBR_ACT_QLY              = FH
        ~~TQ_WHEEL_PBR                   = 0 Nm~~
        AC_WHEEL_PBR                   = 0 m/s²
        STATE_TQ_WHEEL_PBR_QLY         = FH
        TCC_CAN_TQ_PBR                 = FH
        CKS_CAN_TQ_PBR                 = FFH
ENDIF
```

43.2.1.3 DXC – SOLL_MOM_ANF (Status hill descent control – HDC)

Formula section:

```
IF      LV_ERR_BN_DHL_CTL = 0          AND
        LV_ERR_CAN_BOFF = 0
THEN
        //status of hill descent control:
        STATE_DHL_CTL = ST_HDC_FN
ELSE    //default value:
        STATE_DHL_CTL = Fh
ENDIF
```

43.2.2 Output messages of the DME to the CAN bus

Application conditions:

Initialisation:	N_MAX_THD_1_CAN = 0	at reset
	N_MAX_THD_2_CAN = 0	at reset
	STATE_TQ_INTV_DCC = 0H	
	LV_POW_CLAS_VEH_CAN_RCV = 0 at transition LV_IGK 1 --> 0	
	CTR_LDM_CAN_INI = 0	at reset
	LV_ENG_RUN_CAN = 0	at reset
	LV_LDM_CAN_INI = 1	at reset
	STATE_EFP_CTL_ECU = 03H	at reset
	T_ERR_N_ENG = 0 s	at reset

Recurrence: TORQUE_1 = 10 ms

TORQUE_2 = C_T_CYC_CAN_ECU_2

BOS_MELDUNG:

at transition LV_IGK and at signal change (time for debounce is 1 s) **or** at transition of Bostoken form 0 -> !0 (time for debounce is 1 s)

BOS_MELDUNG_2:

at transition LV_IGK and at signal change of either BOS_MELDUNG **or** BOS_MELDUNG_2 (time for debounce is 1 s) **or** at transition of Bostoken form 0 -> !0 (time for debounce is 1 s)

Status Kraftstoffregelung DME:

General: 10s

at transition LV_IGK and at signal change (time for debounce is 160ms);

at transition LV_IGK 0 → 1 the message has to be sent 3 times,

recurrency 160ms

ENGINE_1 = 100 ms

ENGINE_2: cyclic = 1s and at signal change (tmin = 50 ms)

ENGINE_RPM = 1 s

POWERMANAGEMENT_BATTERIESPANNUNG

always when LV_CAN_SND_MSG_PWR_MNG_1 is set

(recurrency of LV_CAN_SND_MSG_PWR_MNG_1 is 100ms)

reset LV_CAN_SND_MSG_PWR_MNG_1 after the message is sent

POWERMANAGEMENT_VERBRAUCHERSTEUERUNG

always when LV_CAN_SND_MSG_PWR_MNG_0 is set

(recurrency of LV_CAN_SND_MSG_PWR_MNG_0 is 100ms)

reset LV_CAN_SND_MSG_PWR_MNG_0 after the message is sent

GESCHWINDIGKEIT_TEMPOMAT

cyclic: 200ms; event triggered: min. time for sending 100ms

RADMOM_PT_1 and RADMOM_PT_2 = 20 ms

ELEKTRONISCHER MOTORÖLMESSSTAB

cyclic: 10 s, event triggered (at transition KL.R off --> on and at signal change): min. time for sending 100ms and on request

WAERMESTROM_MOTOR:

Cyclic: 1000ms, event-triggered, min. time 10 ms

OBD_DT_MOTOR: cyclic 1000ms

ST_BATT: event triggered at signal change (tmin = 200 ms) -> the message has to be sent 3 times with recurrence 160ms

DISP_GSI: - event triggered at signal change (tmin = 100ms)
-cyclic with recurrence 1s

DISP_ENGDAT: 100ms (tmin=90ms, tmax=110ms)

Activation: POWERMANAGEMENT_BATTERIESPANNUNG **AND**

POWERMANAGEMENT_VERBRAUCHERSTEUERUNG

only if LV_VAR_VEH = 0 (E60)

GESCHWINDIGKEIT_TEMPOMAT at Kl.15 on **and** LV_VAR_DCC = 0 **and** LV_VAR_BN_LDM = 0

RADMOM_PT_1 and RADMOM_PT_2 only if LV_VAR_BN_LDM = 1

ELEKTRONISCHER MOTORÖLMESSSTAB at bus_active (always)

Status Kraftstoffregelung DME: Kl. 15 on

OBD_DT_MOTOR: LV_IGK = 1 **and**
LC_ETCU_FRF = 0 **and**
(LV_AT = 1 **or** LV_VAR_TCT =1)

ST_BATT: LV_IGK = 1

DISP_GSI:(LV_AT = 0 **AND** LV_VAR_AMT = 0 **AND** LV_VAR_TCT = 0)

ENGINE_2: LV_IGK = 1

DISP_ENGDAT: C_N_DISP_DYN_ENA != 0

Deactivation: GESCHWINDIGKEIT_TEMPOMAT at Kl.15 off

ELEKTRONISCHER MOTORÖLMESSSTAB at Kl.15 off

Status Kraftstoffregelung DME: Kl. 15 off

DISP_ENGDAT: C_N_DISP_DYN_ENA = 0

Formula section:

43.2.2.1 DME1

43.2.2.2 Anfrage

requested by DME:

```
IF          B_codierpm = 1
THEN        "Anfrage: Codierung Powermanagement" is reqested by DME from CAS
            ID 395H                                        // (recurrency 5 sec)
ENDIF

IF          LV_POW_CLAS_VEH_CAN_REQ 0->1
THEN        "Anfrage: Fahrzeutyp" is reqested by DME from CAS
            ID 388H                                        // once per trigger
ENDIF
```

requested from DME by other STG:

```
IF          ID2_INQY     =       1

THEN        IF      ID_FB_INQY = 897 dez// please note : motorola-format

            THEN    "Anfrage: Elektronischer Ölmeßstab" is requested from DME
                    By STG (ID 580 + STG)
                    --> description of message see 3.1.2
            ELSE IF         ID_FN_INQY  =        910 dez
                            // please note that the positions of high byte
                            and low byte are changed (Intel-format)

                    THEN    "Anfrage: Startdrehzahl" is reqested from DME
                            by STG  (ID 580H + STG)
                            RPM_ENG_STA = N_ST_POW_MOD  (converted)
                    ENDIF
            ENDIF
ENDIF
```

43.2.2.3 Elektronischer Ölmeßstab

Please note: this message is sent:

- cyclic

- event-triggered
- on request (see 3.1.1)

```
If      LV_IGK = 0
Then    CTR_DLY_MSG_OIL = 0
Else    if      CTR_DLY_MSG_OIL < C_CTR_DLY_MSG_OIL
        Then    CTR_DLY_MSG_OIL ++
        Endif
Endif

If      CTR_DLY_MSG_OIL >= C_CTR_DLY_MSG_OIL  or
        LC_SND_MSG_OIL_KEY_AUX = 1
 Then   RECM_RFLV_EOI =  STATE_OIL_REQ
        If      STATE_LOIL = 0
        Then    FLLV_EOI = C_STATE_LOIL_SUB
        Else    FLLV_EOI = STATE_LOIL
        Endif
 Else   FLLV_EOI = C_STATE_LOIL_SUB
        RECM_RFLV_EOI =  0
 Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_SND_MSG_OIL_KEY_AUX	1	0...1H	0...1	1	-
LC for transmitting Layer-values independent of Kl.15					
C_CTR_DLY_MSG_OIL	1	0...FFH	0...255	1	-
counter for transmitting Layer-values after Kl.15 on					
C_STATE_LOIL_SUB	1	0...FFH	0...255	1	s
Default value for transmitting layer-values					

43.2.2.4 BOS_MELDUNG

ID2_BOS_MESS = idbosmsg

ID_FN_BOS_MESS = idfbosmg_w

RMMI_BOS = DIST_RESI_OIL

ST_UN_BOS = DIST_RESI_OIL_KM

AVAI_BOS_MESS = STATE_OIL_AVL

COU_RSTG_BOS_MESS = zrbosmld

43.2.2.5 BOS_MELDUNG_2

ID2_BOS_MESS_2 = QOIL_DS_CAN_2_1
ID_FN_BOS_MESS_2 = QOIL_DS_CAN_2_2
FRC_INTM_WAY_BOS_MESS_2 = QOIL_DS_CAN_2_3
TARD_YR_BOS_MESS_2 = QOIL_DS_CAN_2_4
TARD_MON_BOS_MESS_2 = QOIL_DS_CAN_2_5
FRC_INTM_T_BOS_MESS_2 = QOIL_DS_CAN_2_6

43.2.2.6 TORQUE_1

check control torque_1 DME **:**

CHKSM_TORQ_1_DME = CKS_CLC_ECU1

alive counter torque_1 DME **:**

ALIV_TORQ_1_DME = TCC_CAN_ECU1
TCC_CAN_ECU1$_{N}$ = TCC_CAN_ECU1$_{N-1}$ + 10 ms

torque actual-value:

TORQ_AVL = TQ_AV (converted for BN2000)

status torque actual-value:

```
If      SF_TQD = 3      or
        LV_DI_TQ_REQ_CAN_MPI_GDI = 1
Then    ST_TORQ_AVL = 3
Else    ST_TORQ_AVL =   SF_TQD
Endif
```

TORQ_AVL_DMEE = TQ_ECU_ETCU (converted for BN2000)

TQI_MAF:

```
IF      LC_TQI_MAF_MAN = 1
Then    TQI_MAF    =    TQI_SP_CAN
Else    TQI_MAF    =    TQI_REF * EFF_IGA_AV_CAN * EFF_SCC_AV_CAN *
                        EFF_LAMB_AV
Endif
```

calculation of TQ_MAF:

TQ_MAF = TQI_MAF + TQ_LOSS

Status_switch_clutch:

```
IF      LV_AT      =    1    OR    LV_VAR_AMT = 1
```

```
THEN    ST_SW_CLT       =   3H
ELSE    ST_SW_CLT       =   LV_IM_CS_PN
ENDIF
```

status DME/ACC switch-off:

```
IF      LV_DCC_OFF_ECU  =   0
THEN    ST_DMEA_SWO     =   0H
ELSE    ST_DMEA_SWO     =   1H
ENDIF
```

receipt_ACC_switch-off_system_DME:

```
IF      LV_DCC_OFF_ACK          =   0
THEN    RCPT_ACC_SWO_SYS_DME        =   0H
ELSE    RCPT_ACC_SWO_SYS_DME        =   1H
ENDIF
```

status_receipt_engine_ACC:

```
IF      LV_VAR_DCC              =   0
THEN    STATE_TQ_INTV_DCC       =   3H
ELSEIF  LV_ERR_BN_TQ_DCC        =   0
THEN    STATE_TQ_INTV_DCC       =   0H
ELSEIF  ERR_SYM_BN_TQ_DCC       =   SYM_1       OR
        ERR_SYM_BN_TQ_DCC       =   SYM_3
THEN    STATE_TQ_INTV_DCC       =   1H
ELSEIF  ERR_SYM_BN_TQ_DCC       =   SYM_2
THEN    STATE_TQ_INTV_DCC       =   2H
ENDIF
```

ST_RCPT_ENG_ACC = STATE_TQ_INTV_DCC

status_receipt_engine_ARS:

```
IF      LV_VAR_ARS              =   0
THEN    STATE_TQ_INTV_ARS       =   3H
ELSEIF  LV_ERR_BN_ARS           =   0
THEN    IF          LV_TQ_MAX_ARS           =   1
        THEN        STATE_TQ_INTV_ARS       =   1H
```

```
            ELSE        STATE_TQ_INTV_ARS       =       0H
            ENDIF
    ELSEIF  ERR_SYM_BN_ARS          =       SYM_1       OR
            ERR_SYM_BN_ARS          =       SYM_3       OR
            LV_TQ_MAX_ARS           =       1
    THEN    STATE_TQ_INTV_ARS       =       1H
    ELSEIF  LV_ERR_BN_ARS           =       1           AND
            ERR_SYM_BN_ARS          =       SYM_2
    THEN    STATE_TQ_INTV_ARS       =       2H
    ENDIF

    ST_RCPT_ENG_ARS = STATE_TQ_INTV_ARS
```

status_receipt_engine_DSC:

```
    IF      LV_VAR_ASR              =       0
    THEN    STATE_TQ_INTV_TCS       =       3H
    ELSEIF  LV_ERR_BN_TQ_TCS        =       0
    THEN    STATE_TQ_INTV_TCS       =       0H
    ELSEIF  ERR_SYM_BN_TQ_TCS       =       SYM_1       OR
            ERR_SYM_BN_TQ_TCS       =       SYM_3
    THEN    STATE_TQ_INTV_TCS       =       1H
    ELSEIF  ERR_SYM_BN_TQ_TCS       =       SYM_2       OR
    THEN    STATE_TQ_INTV_TCS       =       2H
    ENDIF

ST_RCPT_ENG_DSC = STATE_TQ_INTV_TCS
```

status_receipt_engine_EGS/SSG/TCT:

```
    IF      LV_AT       =       1
    THEN    ST_RCPT_ENG_EGS         =       STATE_TQ_INTV_ETCU
    ELSEIF  LV_VAR_AMT =        1
    THEN    ST_RCPT_ENG_EGS         =       STATE_TQ_INTV_AMT
    ELSEIF  LV_VAR_TCT =        1
    THEN    ST_RCPT_ENG_EGS         =       STATE_TQ_INTV_TCT
    ENDIF

    IF      LV_AT       =       0
    THEN    STATE_TQ_INTV_ETCU      =       3H
    ELSEIF  LV_ERR_BN_TQ_ETCU       =       0
```

```
THEN      STATE_TQ_INTV_ETCU        =    0H
ELSEIF    ERR_SYM_BN_TQ_ETCU        =    SYM_1     OR
          ERR_SYM_BN_TQ_ETCU        =    SYM_3
THEN      STATE_TQ_INTV_ETCU        =    1H
ELSEIF    LV_ERR_BN_TQ_ETCU         =    1         AND
          ERR_SYM_BN_TQ_ETCU        =    SYM_2     OR
THEN      STATE_TQ_INTV_ETCU        =    2H
ENDIF

IF        LV_VAR_AMT      =    0
THEN      STATE_TQ_INTV_AMT         =    3H
ELSEIF    STATE_ERR_AMT_CAN         =    0H
THEN      STATE_TQ_INTV_AMT         =    0H
ELSEIF    STATE_ERR_AMT_CAN         =    2H        OR
          STATE_ERR_AMT_CAN         =    3H        OR
THEN      STATE_TQ_INTV_AMT         =    1H
ELSEIF    STATE_ERR_AMT_CAN         =    1H
THEN      STATE_TQ_INTV_AMT         =    2H
ELSE      STATE_TQ_INTV_AMT         =    3H
ENDIF

IF        LV_VAR_TCT      =    0
THEN      STATE_TQ_INTV_TCT         =    3H
ELSEIF    LV_ERR_BN_TQ_TCT          =    0
THEN      STATE_TQ_INTV_TCT         =    0H
ELSEIF    ERR_SYM_BN_TQ_TCT         =    SYM_1     OR
          ERR_SYM_BN_TQ_TCT         =    SYM_3
THEN      STATE_TQ_INTV_TCT         =    1H
ELSEIF    LV_ERR_BN_TQ_TCT          =    1         AND
          ERR_SYM_BN_TQ_TCT         =    SYM_2
THEN      STATE_TQ_INTV_TCT         =    2H
ENDIF
```

status_OBD_control-function_gearbox ST_OBD_CTFN_GRB:
ST_OBD_CTFN_GRB = STATE_DIAG_GS

status_contact_brake-pedal_DME:

```
if        LV_ERR_BLS_PLAUS        =      1
then      ST_CT_BRPD_DME          =      111b
elseif          LV_IM_BTS    = 0    and    LV_IM_BLS    = 0
then            ST_CT_BRPD_DME             =       000b
elseif          LV_IM_BTS    = 1    and    LV_IM_BLS    = 0
then            ST_CT_BRPD_DME             =       010b
elseif          LV_IM_BTS    = 0    and    LV_IM_BLS    = 1
then            ST_CT_BRPD_DME             =       001b
else            ST_CT_BRPD_DME             =       011b
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TQI_MAF_MAN	1	0...1	0...1	1	[-]
Logical constant to switch between two paths of the TQI_MAF calculation					

43.2.2.7 TORQUE_2

checksum_torque_2_DME:

CHKSM_TORQ_2_DME = CKS_CLC_ECU2

status_over-run-fuel-shutoff:

```
IF          LV_PUC=0    and    LV_TQ_DCC_INC_REQ=1  and
                                   TQI_DCC_FAST_INC < TQI_REF_IGA_MIN_LAMB
Then   ST_INFS =02H
else   ST_INFS = LV_PUC
endif
```

alive counter torque_2 DME **:**

ALIV_TORQ_2_DME = TCC_CAN_ECU2

$TCC\_CAN\_ECU2_{N}$ = T $CC\_CAN\_ECU2_{N-1}$ + 10 ms

status_switch_level_RPM:

```
if        N_SP_IS            >      C_N_SP_IS_THD_CAN
then      ST_SW_LEV_RPM      =      00b
     LV_LEVEL_IS        =      0
else      ST_SW_LEV_RPM      =      01b
     LV_LEVEL_IS        =      1
endif
```

torque_actual-value_minimum:

TORQ_AVL_MIN = TQ_LOSS (converted for BN2000)

torque_actual-value_maximum:

TORQ_AVL_MAX = TQ_MAX_CLU (converted for BN2000) *torque_actual-value_spare_negative:*

TORQ_AVL_SPAR_NEG = TQ_REF_IGA_MIN_LAMB (converted for BN2000)

TQ_REF_IGA_MIN_LAMB = TQI_REF_IGA_MIN_LAMB + TQ_LOSS

torque_actual-value_spare_positive:

TORQ_AVL_SPAR_POS = TQ_MAF (converted for BN2000)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_CYC_CAN_ECU_2	1	0...FFH	10...2560	10	[ms]
Constant for cycle time of output CAN-message "TORQUE_2" (default = 10ms)					

43.2.2.8 TORQUE_3

checksum_torque_3_DME:

CHKSM_TORQ_3_DME = CKS_CLC_ECU3

alive counter torque_2 DME **:**

ALIV_TORQ_3_DME = TCC_CAN_ECU3

$TCC\_CAN\_ECU3_N$ = $TCC\_CAN\_ECU3_{N-1}$ + 10 ms

torque_driver's-choice:

TORQ_DVCH = TQ_EMS_BN (converted)

TQ_EMS_BN = TQ_LOSS + MIN [MAX(TQI_EMS;TQI_DCC_FAST_INC); TQI_REF_MAX*EFF_IGA_BAS_COR]

angle_accelerator-pedal:

PV_MAX_CAN = PV_MAX (converted)

```
If      PV_MAX_CAN  <   FFH
Then    ANG_ACPD    =   PV_MAX_CAN
Else    ANG_ACPD    =   FEH
Endif
```

```
// save "engine should run-status" without need of ENSD
If      LV_ST_END = 1
Then    LV_ENG_RUN_CAN = 1
        T_ERR_N_ENG = 0 s
Else    If      ECU_STATE = PWL
        Then    LV_ENG_RUN_CAN = 0
        Endif
Endif
```

RPM_engine and RPM_engine_error:

```
If      N > 0
Then    N_CAN = N    (converted)
        RPM_ENG_ERR = LV_ERR_CRK
Else    if      (LV_ENG_RUN_CAN = 1 AND LV_STALL=0)
        Then    if (T_ERR_N_ENG >= C_T_MAX_ERR_N_ENG) and LV_ERR_CRK=0
                then    LV_ENG_RUN_CAN = 0
                        T_ERR_N_ENG = 0 s
                        N_CAN = 0
                        RPM_ENG_ERR = 00H
                else    T_ERR_N_ENG = T_ERR_N_ENG + 0,01 s
                        N_CAN = FFFFH            // CRK error active but not debounced
                        RPM_ENG_ERR = 01H        // RPM_ENG=FFFH → RPM_ENG_ERR=1
                endif
        Else    N_CAN = 0
                RPM_ENG_ERR = LV_ERR_CRK
        Endif
Endif
RPM_ENG         =       N_CAN
```

status_idling_engine:

```
ST_IDLG_ENG     =       ! LV_IS
```

status closed-loop-control speed:

```
IF          STATE_CRU_CAN   = 7H                // DEC
THEN        STATE_CRU_BN    = 4H
ELSEIF      STATE_CRU_CAN   = 1H                // CONST/TIP
THEN        STATE_CRU_BN    = 5H
ELSEIF      STATE_CRU_CAN   = 3H                // RES
THEN        STATE_CRU_BN    = 6H
ELSEIF      STATE_CRU_CAN   = 5H                // SET_ACC
THEN        STATE_CRU_BN    = 7H
ELSEIF                  (STATE_CRU_CAN                          = 0H   AND
      //PASSIVE
            PV_AV        = 0%)
THEN        STATE_CRU_BN    = 8H
ELSEIF      (STATE_CRU_CAN  = 0H   AND          //PASSIVE
            PV_AV        <> 0%)
THEN        STATE_CRU_BN    = 9H
ENDIF
```

```
if          LV_MTC_CUR_OFF = 1
then        ST_CLCTR_V       = FH
elseif      LV_KD  =  1
then        ST_CLCTR_V       = 0BH
elseif      LV_VAR_DCC = 1
then        ST_CLCTR_V       = STATE_TQ_DCC
elseif      LV_VAR_BN_LDM = 1
then        ST_CLCTR_V       = STATE_TQ_LDM
else        ST_CLCTR_V       = STATE_CRU_BN
endif
```

Required_amount_of_fuel:

```
IF          CONF_SWI_EFP_OUT      =       !1
THEN        IF            LC_EFPPWM_CTL = 0
            THEN          RQAM_FU     =       VFF_EFP
            ELSE          RQAM_FU     =       EFPPWM_CAN
ENDIF
```

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_EFPPWM_CTL	1	0...1H	0...1	1	-
Switch between PT-CAN definition and EKP Ctl functionality					
C_T_MAX_ERR_N_ENG	1	0...FFH	0...2.55	0.01	[s]
Threshold for sending RPM_ENG_ERR true or false without debounced CRK error					

43.2.2.9 ENGINE_RPM

actual engine speed limit (orange area) depending on TCO:

```
IF          N_MAX_THD_1           >       N_MAX_THD_1_CAN
THEN        N_MAX_THD_1_CAN       =       N_MAX_THD_1
```

```
ENDIF
RPM_TEMP_DOM_1      =       N_MAX_THD_1_CAN

IF        LV_ERR_TCO          = 0
THEN      RPM_TEMP_DOM_1 = N_MAX_THD_1_CAN
ELSE      RPM_TEMP_DOM_1 = FFH
ENDIF
```

actual engine speed limit (red area) depending on TCO:

```
IF        N_MAX_THD_2              >       N_MAX_THD_2_CAN
THEN      N_MAX_THD_2_CAN          =       N_MAX_THD_2
ENDIF

IF        LV_ERR_TCO          = 0
THEN      RPM_TEMP_DOM_2 = N_MAX_THD_2_CAN
ELSE      RPM_TEMP_DOM_2 = FFH
ENDIF
```

43.2.2.10 ENGINE_1

temperature engine :

```
if        LV_ERR_TCO          = 1
then      TEMP_ENG            = FFH
else      TEMP_ENG            = TCO ( converted)
endif
```

alive counter DME :

ALIV_COU_DME_N = TCC_CAN_ECU

TCC_CAN_ECU_N = TCC_CAN_ECU_{N-1} + 200 ms

temperature engine oil :

TOIL_CAN = TOIL (converted)

```
if        LV_ERR_QOIL_SENS  = 1  and
          ( ERR_SYM_QOIL_SENS = SYM_0 or
            ERR_SYM_QOIL_SENS = SYM_2 )
then      TEMP_EOI   = FFH
else      TEMP_EOI   = TOIL_CAN
endif
```

status engine run :

```
if        LV_ES = 1
then      ST_ENG_RUN = 00b
elseif    LV_ST = 1
then      ST_ENG_RUN = 01b
elseif    LV_ES = 0
     and  LV_ST = 0
then      ST_ENG_RUN = 10b
endif
```

status switch warm-up ST_SW_WAUP :

```
if      LV_LIH_COC_ON       =   1
then    ST_SW_WAUP          =   10b
elseif  LV_WUP_CAN          =   0
then    ST_SW_WAUP          =   00b
else    ST_SW_WAUP          =   01b
endif
```

air-pressure engine AIP_ENG :

```
AMP_CAN         =       AMP_MES
if      LV_ERR_AMP      =       1
then    AIP_ENG         =       FFH
else    AIP_ENG         =       AMP_CAN
endif
```

Injection-volume_fuel:

```
IJV_FU    =     FCO   (like CAN11h)
```

Control_shift-lock:

```
IF      (LV_ACT_N_SP_IS_EXT_ADJ     = 1         AND
        LV_IS  = 1                              AND
        VS     = 0
THEN    CTL_SHIFT_LOCK_CAN = 10b            intervention
ELSE    CTL_SHIFT_LOCK_CAN = 01b            no intervention (default)
ENDIF
CTR_SLCK        =       CTL_SHIFT_LOCK_CAN
```

Active engine speed limit in case of limp home:

```
RDUC_DOCTR_RPM = STATE_N_MAX_THD_LIH
```

RPM_idling_target:

```
RPM_IDLG_TAR  =  N_SP_IS
```

Status_rbm-cycle:

```
IF      LV_DC_RBM = 1
THEN    ST_RBMCYC = 10b
ELSE    ST_RBMCYC = 00b
ENDIF
```

43.2.2.11 ENGINE_2 (Motordaten 2, ID=383h)

Request starting assistance:

```
If      B_notafu = 0
Then    RQ_STASS = 0
Else    RQ_STASS = 1
Endif
```

43.2.2.12 GESCHWINDIGKEIT_TEMPOMAT

speed_closed-loop-control_cruise-control:

V_CLCTR_CCTR = VS_SP_DRIV_CRU_CAN

status_speed_closed-loop-control_cruise-control ST_V_CLCTR_CCTR:

```
IF      LV_PROG_STEP_1  =     0
THEN    IF          LV_CRU_MAIN_SWI         = 0
THEN          ST_V_CLCTR_CCTR        = 00H          "Off"
ELSE          IF    STATE_CRU           =      "PASSIVE"
                    THEN  ST_V_CLCTR_CCTR =     01H    "inactive"
                    ELSE  ST_V_CLCTR_CCTR =     02H    "active"
                    ENDIF
        ENDIF
ELSE    ST_V_CLCTR_CCTR =     01H    "inactive"
ENDIF
```

control_display_cruise-control:

CTR_DISP_CCTR = LV_STEP_ON_ICL

control_display_cruise-control_HUD:

CTR_DISP_CCTR_HUD = LV_CRU_DISP_HUD

receipt_activation_cruise-control_DR:

RCPT_ACTVN_CCTR_DR = LV_CRU_ACT_INH

index_number:

IDX_NO = NR_VS_CRU

speed_cruise-control_index_number_i :

V_CCTR_IDX_NO_i = VS_STEP_CAN_i (i = 1,2,3)

43.2.2.13 WAERMESTROM_MOTOR

air-conditioning-compressor_limit_torque:

```
IF      LV_ACCOUT_RLY = 0
THEN    TQ_MAX_ACC_CAN = 0
ELSE    TQ_MAX_ACC_CAN = min(TQ_MAX_ACC, C_TQ_MAX_ACC)
ENDIF
```

ACCM_LIM_TORQ = TQ_MAX_ACC_CAN

heat-flow_request_DME:
HTFL_RQ_DME = FFh

heat-flow_status_DME:
HTFL_ST_DME = N_REL_CWP

status_current-raw_engine:
ST_CRAW_ENG = STATE_CUR_ENG_CNS

status_current_generator:
ST_I_GEN = CUR_GEN

status_generator:
ST_GEN = STATE_ALTER

status_load-state_generator:
ST_LDST_GEN = st_ldstgen

control_aperture_water-valve:
CTR_APE_WV = PWM_VALUE_HIV_PERC

load-state_generator:
LDST_GEN = POW_REL_ALTER_CLC

43.2.2.14 POWERMANAGEMENT_BATTERIESPANNUNG

(in case of E60, LV_VAR_VEH = 0)

T_SEC_COU_REL_RED = FFFFFFH
SDWN_COU_BTSW = FFH
ST_SWO_KL_30 = 3H
ST_COS_SWO = 3H
ST_HT_RSCR = 3H
ST_PCOS = stpcos
CTR_PRED_PRIO_CF = POW_CTL_PRI_PEAK_RED_CFT
CTR_PRED_PRIO = POW_CTL_PRI_PEAK_RED
ST_SW_DSTB = 3H
ST_BTSW = 3H
U_BT = VB_POW_MNG

```
If      LV_ERR_ACK_IGK_OFF = 1
```

```
then    ACKM_RQ_SWO_KL_15 = 0
else    ACKM_RQ_SWO_KL_15 = STATE_ACK_IGK_OFF
endif
```

43.2.2.15 POWERMANAGEMENT_ VERBRAUCHERSTEUERUNG

(in case of E60, LV_VAR_VEH = 0)

CTR_PCOS = POW_CTL_PARK_CNS
RQ_PCOS = rqpcos
CTR_CBR = CUR_RNG_CTL
CTR_PWR_COS = POW_CTL_PWR_CNS_1
CTR_PWR_SPCOS = POW_CTL_PWR_CNS_2
SLCTN_SPCOS = selspcos
ULEV = ulev

43.2.2.16 Radmoment Antriebsstrang 1

CHKSM_WMOM_PT_1 = CKS_CLC_TQ_WHEEL_1

ALIV_WMOM_PT_1 = TCC_CAN_TQ_WHEEL_1

$\text{TCC\_CAN\_TQ\_WHEEL\_1}_{N} = \text{TCC\_CAN\_TQ\_WHEEL\_1}_{N-1} + 20\ \text{ms}$

ST_PENG = STATE_ENGG_POS

WMOM_PT_AVL = TQ_WHEEL

```
IF      LV_TQ_WHEEL_LDM_BN_ERR = 1
THEN    PV_LDM_REQ_CAN = FFFH
ELSE    PV_LDM_REQ_CAN = PV_CRU (converted)
ENDIF
```

ANG_ACPD_TAR = PV_LDM_REQ_CAN

ANG_ACPD_REAL = PV_AV

```
IF      CTR_LDM_CAN_INI > = C_CTR_LDM_CAN_INI
Then    ST_INTF_PT = 0
        LV_LDM_CAN_INI = 0
        If      LV_LDM_OFF_ECU = 1       or
                LV_ERR_LDM_INH_MON = 1
        Then    Bit 1 of ST_INTF_PT is set to 1
        Else    Bit 1 of ST_INTF_PT is reset to 0
        Endif

        If      LV_LDM_OFF = 1
        Then    Bit 0 of ST_INTF_PT is set to 1
        Else    Bit 0 of ST_INTF_PT is reset to 0
        Endif
Else    Bit 0, Bit 1, Bit 2 and Bit 3 of ST_INTF_PT are set, Bit 4 is reset to 0
        // ST_INTF_PT = FH
        if      LV_IGK = 1
```

```
        then    CTR_LDM_CAN_INI ++
                //incremented with cyclic transmission of message
        endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_LDM_CAN_INI	1	0...FF	0...255	1	-
Counter for sending ini-value of ST_INTF_PT					
C_TQ_MAX_ACC	1	0...FFH	0...127.5	0.5	[Nm]
air-conditioning-compressor limit torque after switch on					

43.2.2.17 Radmoment Antriebsstrang 2

CHKSM_WMOM_PT_2 = CKS_CLC_TQ_WHEEL_2

ALIV_WMOM_PT_2 = TCC_CAN_TQ_WHEEL_2

$TCC\_CAN\_TQ\_WHEEL\_2_{N} = TCC\_CAN\_TQ\_WHEEL\_2_{N-1} + 20\ ms$

WMOM_PT_MAX = TQ_MAX_WHEEL

WMOM_PT_MIN_LOW = TQ_MIN_WHEEL_L

WMOM_PT_COODT_TAR = Md_rad_ksoll (converted to BN! see LH)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_SP_IS_THD_CAN	1	0...1FE0H	0...8160	1	[rpm]
Engine speed setpoint threshold for reconition of ST_SW_LEV_RPM condition					

43.2.2.18 Challenge to CAS (EWS):

The frame n+1 will only be send if the frame n has be confirmed by the can-driver

Frame 1:

ID c → see LH EWS

IMOB_TRM_CHAL_x

Frame 2:

ID d → see LH EWS

IMOB_TRM_CHAL_x

Frame 3:

ID e → see LH EWS

IMOB_TRM_CHAL_x

ACTION_COMM_SEND_CHAL_CAN_TRIG() :

ctr_imob_resp_rcv_can = 0 (delete receive-counter)

Start Transmission

43.2.2.19 Status Kraftstoffregelung DME

status_fuel_closed-loop-control_DME:

```
IF      LC_EFPPWM_CTL   =       0
THEN    STATE_EFP_CTL_ECU = 01H
ELSE    STATE_EFP_CTL_ECU = 02H
ENDIF
ST_FU_CLCTR_DME = STATE_EFP_CTL_ECU
```

43.2.2.20 OBD_DT_MOTOR (OBD Daten Motor)

Position of throttle valve (converted from 0...FFh -> 0...FEh):

```
if      OBD_TPS_1 > 7Fh
then    PO_THVA_ENG = OBD_TPS_1 - 1h    (shift for conversion)
else    PO_THVA_ENG = OBD_TPS_1
endif
```

Calculated value for enginge load (converted from 0...FFh -> 0...FEh):

```
if      LOAD_CLC > 7Fh
then    CALCVL_LD_ENG   = LOAD_CLC - 1h    (shift for conversion)
else    CALCVL_LD_ENG   = LOAD_CLC
endif
```

43.2.2.21 ST_BATT (Status Battterie – ID:38Bh)

Condition battery:

CND_BT = Qv_out_m // *converted* -> Ph = 2 * [hex] %

Condition battery accuracy:

CND_BT_ACCY = Qv_quali_m // *converted* -> Ph = 2 * [hex] %

Condition battery change:

CND_BT_CHNG = Qvc_status_4 // *converted* -> Ph = 2 * [hex] - 128

Condition battery time (resolution 4 h, Qv_td -> resolution 1h):

```
If      Qv_td1 / 4 < 255
Then    CND_BT_T = Qv_td1 / 4       //residue not used
Else    CND_BT_T = FEh
Endif
```

Status condition battery:

```
If      Qvc_status_3 < FH
Then    ST_CND_BT = Qvc_status_3
Else    ST_CND_BT = 1
Endif
```

43.2.2.22 DISP_GSI- (Anzeige Schalthinweis – ID:2F3h)

Formula section:

alive counter display gear shift indication :

ALIV_DISP_GSI = TCC_CAN_GS_IDC

$\text{TCC\_CAN\_GC\_IDC}_{n} = \text{TCC\_CAN\_GS\_IDC}_{n-1} + 1$

display target gear:

DISP_TARG = GS_IDC_GEAR (copy lowest 4 bit)

display gear shift indication :

DISP_GSI = GS_IDC_DISP (copy lowest 2 bit)

Checksum for gear shift indication :

CHKSM_DISP_GSI = CKS_CLC_GS_IDC

43.2.2.23 DISP_ENGDAT (Anzeige Motordaten – ID:175h)

alive counter display gear shift indication :

ALIV_DISP_ENGDAT = TCC_CAN_DISP_ECU

$\text{TCC\_CAN\_DISP\_ECU}_{n} = \text{TCC\_CAN\_DISP\_ECU}_{n-1} + 1$

display rpm of engine:

```
If      C_N_DISP_DYN_ENA = 1
Then    DISP_RPM_ENG   =   N_DISP_DYN
Else    DISP_RPM_ENG   =   N_CAN
Endif
```

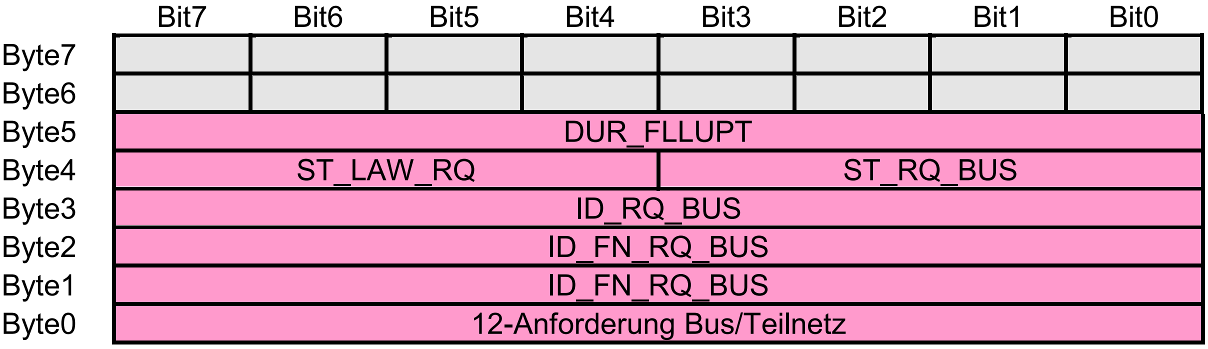
Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_DISP_DYN_ENA	1	0...2H	0...2	1	[-]
Condition for sending DISP_RPM_ENG on CAN (0...message not sent, 1...=N_DISP_DYN, 2...=N_CAN)					

43.2.2.24 Dienste - Anforderung Bus/Teilnetz (ID2 = 12/Ch)

General information:

This message is sent to guarantee a defined PWL time for DMTL-diagnosis. The maximum power latch time is requested only once per driving cycle. This Function is only necessary for vehicle platforms which have a switched Klemme 30 by CAS. Message Layout:



	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte7								
Byte6								
Byte5	DUR_FLLUPT							
Byte4	ST_LAW_RQ				ST_RQ_BUS			
Byte3	ID_RQ_BUS							
Byte2	ID_FN_RQ_BUS							
Byte1	ID_FN_RQ_BUS							
Byte0	12-Anforderung Bus/Teilnetz							

Application conditions:

Initialisation: LV_REQ_PWL_DMTL_SND = 0 at reset or LV_IGK 0->1

Recurrence: event triggered (tmin=100ms, sent 3 times with recurrence 160ms) -> see formula section

Activation: LC_ENA_REQ_PWL_DMTL_CAN = 1

Formula section:

```
If      LV_REQ_PWL_DMTL = 0 -> 1  and  LV_REQ_PWL_DMTL_SND = 0
Then    send message 3 times with recurrence 160ms (as described below)
        //ID_function_request_bus : tank leakage diagnosis:
        ID_FN_RQ_BUS      =  110h
        //request bus  "Klemme 30g":
        ID_RQ_BUS         =  50
```

```
        // status of request bus : activation:
        ST_RQ_BUS          =  0
        // request due to law:
        ST_LAW_RQ          =  1
        // requested powerledge time (converted and limited):
        DUR_FLLUPT         =  C_T_DMTL_MAX
        (DUR_FLLPT = 0 ... 254 min,  (PH) = (HEX)  [min])
        If      sending finished
        Then    LV_REQ_PWL_DMTL_SND = 1
        Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_ENA_REQ_PWL_DMTL_CAN	1	0...1H	0...1	1	[-]
Switch to enable requests to get additional powerledge time for DMTL via CAN (only relevant for E70)					

43.2.3 Checkcontrol- Messages

General Information:

Activation: LV_KEY_AUX = 1 **OR** LV_IGK = 1

Update rate: 100ms

Initialisation: at reset or at transition KL.15 0 --> 1 :

LV_CC_POIL_SWI_INH = 1
T_DLY_POIL_SWI_CAN = IP_T_DLY_POIL_SWI_CAN__TOIL
LV_CC_TEST_BENCH_ACT = 0

at reset:

CYC_CTR_CC = 0
LV_CC_ID_XX = 0
LV_SEND_CC_ID_XX = 0
LV_CC_ID_BENCH = 0
LV_SEND_CC_ID_BENCH = 0
CC_ID = 0

Formula section:

conditions for sending of CC-messages:

--> to be calculated only in 100ms recurrency:

```
if       (LV_ES = 0)
then     STATE_CC_KEY = 3  // Motor läuft
else     if     (LV_IGK = 1)
         then  STATE_CC_KEY = 2       // Kl. 15 an
         else  if    (LV_KEY_AUX = 1)
               then  STATE_CC_KEY = 1  // Kl. R an
               else  STATE_CC_KEY = 0
               endif
         endif
endif
```

// Auxiliary bit for suppressing EML during predrive-check:

--> to be calculated only in 500ms recurrency

```
if          LV_WAL_1_EXT_ADJ = 1
then        LV_WAL_1_CAN = LV_ACT_WAL_1_EXT_ADJ
elseif      (LV_WAL_ST = 1)                          //predrive check active
then        LV_WAL_1_CAN = 0;
else        LV_WAL_1_CAN = LV_WAL_1
endif

if          LV_MIL_EXT_ADJ = 1
then        LV_MIL_CAN = LV_ACT_MIL_EXT_ADJ
else        LV_MIL_CAN = LV_MIL
endif
```

// Auxiliary bit for MIL-Lamp only (or flash) CC-message:

--> to be calculated only in 500ms recurrency

```
if          STATE_MIL_ON_DIS_EXT_REQ = 02h or  LC_ENA_CC_MIL_1 = 0
            //MIL-OFF by coding or calibration
then        LV_MIL_CAN_1 = 0
elseif      LV_MIL_EXT_ADJ = 1  // external adjustment by tester
then        LV_MIL_CAN_1 = LV_ACT_MIL_EXT_ADJ
elseif      STATE_MIL = 2
    then    if      STATE_MIL $_{N-1}$ <> 2
            then    LV_MIL_CAN_1 = 0
            else    LV_MIL_CAN_1 =1
            endif
    else    if      [LV_MIL =1 and
                    ( LV_WAL_ST= 1 or  STATE_MIL = 1 or  LV_MIL_ACT_REQ=1)]
            then    LV_MIL_CAN_1 = 1
            else    LV_MIL_CAN_1 = 0
            endif
    endif
endif
```

~~// Auxiliary bit for MIL-dot-matrix graphic CC-message:~~

~~--> to be calculated only in 500ms recurrency~~

```
~~if          LV_MIL_EXT_ADJ = 1 and  LC_ENA_CC_MIL_2 = 1~~
~~then        LV_MIL_CAN_2 = LV_ACT_MIL_EXT_ADJ~~
~~elseif      STATE_MIL_ON_DIS_EXT_REQ = 02h~~
~~then        LV_MIL_CAN_2 = 0~~
~~elseif      (( LV_MIL  &&  [STATE_MIL = 1  or  LV_MIL_ACT_REQ=1]  &&~~
~~              LC_ENA_CC_MIL_2 = 1))~~
~~then        LV_MIL_CAN_2 = 1~~
~~else        LV_MIL_CAN_2 = 0~~
~~endif~~
```

//Auxiliary bit for FUC CC-message:

```
--> to be calculated only in  500ms recurrency
if          LV_FUC_CAN = 1                    or
            LV_CHK_FUC_OPEN_CAN = 1
then        LV_FUC_CAN_BN = 1
else        LV_FUC_CAN_BN = 0
endif
```

//Auxiliary bit for „BLS defect“ CC-message:

```
--> to be calculated only in  500ms recurrency
if          (VAR_VEH = PL2  &&
             LV_ERR_BLS_PLAUS = 1)
then        LV_BLS_PLAUS_CAN = 1
else        LV_BLS_PLAUS_CAN = 0
endif
```

// activation of Bench mode

```
--> to be calculated in 100ms recurrency:
if      (LC_CC_TEST_BENCH = 1)
then    STATE_CC_TEST_BENCH_OBD_1 = STATE_READY_OBD_1 nor
        C_STATE_CC_TEST_MASK_OBD_1 // (bitweise Verknüpfung)
        STATE_CC_TEST_BENCH_OBD_2 = STATE_READY_OBD_2 nor
        C_STATE_CC_TEST_MASK_OBD_2 // (bitweise Verknüpfung)
        if    [(C_STATE_CC_TEST_BENCH [Bit 0] = 1) and
              (CTR_ERR_DYN_NR >= C_CTR_ERR_DYN_CC_TEST_BENCH)] or
              [(C_STATE_CC_TEST_BENCH [Bit 1] = 1) and
              (OPM_AV >= C_STATE_OPM_MIN_CC_TEST_BENCH) and
              (OPM_AV <= C_STATE_OPM_MAX_CC_TEST_BENCH)]        or
              [(C_STATE_CC_TEST_BENCH [Bit 2] = 1) and
              (STATE_COC >= C_STATE_COC_MIN_CC_TEST_BENCH) and
              (STATE_COC <= C_STATE_COC_MAX_CC_TEST_BENCH)]  or
              [(C_STATE_CC_TEST_BENCH [Bit 3] = 1) and
              (CTR_CDN_OBD_RBM > CTR_CDN_OBD_RBMn-1)]          or
              [(C_STATE_CC_TEST_BENCH [Bit 4] = 1) and
              (LV_WUP = 0)]                                     or
              [(C_STATE_CC_TEST_BENCH [Bit 5] = 1) and
              (STATE_CC_TEST_BENCH_OBD_1 =
               C_STATE_CC_TEST_BENCH_OBD_1     or
               STATE_CC_TEST_BENCH_OBD_2 =
               C_STATE_CC_TEST_BENCH_OBD_2 )]
```

```
                then    LV_CC_TEST_BENCH_ACT = 1
                else    LV_CC_TEST_BENCH_ACT = 0

                endif
     else       LV_CC_TEST_BENCH_ACT = 0
     endif



     // Auxiliary bit and transmission for CC message „low oil pressure“:

     --> to be calculated only in  100ms recurrency:
       If       N_32  >      C_N_32_THD_CC_POIL_SWI
                if      T_DLY_POIL_SWI_CAN    >     0
                then    T_DLY_POIL_SWI_CAN    =     T_DLY_POIL_SWI_CAN - 100 ms
                else    LV_CC_POIL_SWI_INH    =     0
                endif
      else      LV_CC_POIL_SWI_INH = 1
      endif

     --> to be calculated only in  500ms recurrency:
      If        LV_CC_POIL_SWI_INH = 0              and
                LV_POIL_SWI       = 0               // oil-pressure low
      then      LV_POIL_SWI_CAN  = 1
      else      LV_POIL_SWI_CAN  = 0
      endif

     // Auxiliary bit and transmission for CC message „TRA-mode“:
     ➔  to be calculated only in  500ms recurrency:

      If        LV_ERR_EGY_MIN_2 = 1       // transportation mode
       then     LV_TRA_MODE_CAN = 1
      else      LV_TRA_MODE_CAN = 0
      endif


     // If an error occurs CC-message has to be activated
     // If an error already exists and a new error occurs the actual
     // counter has to be stopped and the new error has to be transmitted first
     // before the old error is transmitted again

     if    (CYC_CTR_CC % 5 = 0 )                // to be calculated only in  500ms recurrency
```

```
    // CC-message MIL-flash-light:
    if      (LV_MIL_CAN_1  != LV_MIL_CAN_1 $_{n-1}$
            && STATE_CC_KEY >= C_STATE_CC_ID_34)
    then    CYC_CTR_CC_SAVE = CYC_CTR_CC
            CYC_CTR_CC = 0
            LV_SEND_CC_ID_34 = true
            if      (LV_MIL_CAN_1 =  1 )
            then    LV_CC_ID_34 = true
            elseif  (LV_MIL_CAN_1 = 0 )
            then    LV_CC_ID_34 = false
            endif
            LV_EVE_CC_MSG = 1

   // CC-message Misfire B:
    elseif  (LV_MIS_STATE_A  != LV_MIS_STATE_A $_{n-1}$
            && STATE_CC_KEY >= C_STATE_CC_ID_33)
    then    CYC_CTR_CC_SAVE = CYC_CTR_CC
            CYC_CTR_CC = 5
            LV_SEND_CC_ID_33 = true
            if      (LV_MIS_STATE_A  = 1 )
            then    LV_CC_ID_33 = true
            elseif  (LV_MIS_STATE_A  = 0 )
            then    LV_CC_ID_33 = false
            endif
            LV_EVE_CC_MSG = 1


    // CC-message "Engine temperature (warning 1)"
    elseif  ( LV_TEMP_ENG_WARN_1 != LV_TEMP_ENG_WARN_1 $_{n-1}$
            && STATE_CC_KEY >= C_STATE_CC_ID_257)
             // only at transition
            // nur Übergang beachten
    then    CYC_CTR_CC_SAVE = CYC_CTR_CC
            CYC_CTR_CC = 10
            LV_SEND_CC_ID_257 = true
            if      (LV_TEMP_ENG_WARN_1 = 1 )
            then    LV_CC_ID_257 = true
            elseif  (LV_TEMP_ENG_WARN_1 = 0 )
```

```
        then    LV_CC_ID_257 = false
        endif
        LV_EVE_CC_MSG = 1

// CC_message "Engine temperature (warning 2)":
if      ( LV_ TEMP_ENG_WARN_2   != LV_TEMP_ENG_WARN_2 n-1
        && STATE_CC_KEY >= C_STATE_CC_ID_39)
        // only at transition
          // only at transition
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 15
        LV_SEND_CC_ID_39 = true
        if      (LV_ TEMP_ENG_WARN_2  = 1 )
        then    LV_CC_ID_39 = true
        elseif  (LV_ TEMP_ENG_WARN_2  = 0 )
        then    LV_CC_ID_39 = false
        endif
        LV_EVE_CC_MSG = 1

// CC_message „low oil pressure":
elseif ( LV_POIL_SWI_CAN != LV_POIL_SWI_CAN n-1
        && STATE_CC_KEY >= C_STATE_CC_ID_212)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 20
        LV_SEND_CC_ID_212 = true
        if      (LV_POIL_SWI_CAN  = 1)
        then    LV_CC_ID_212 = true
        else    LV_CC_ID_212 = false
        endif
        LV_EVE_CC_MSG = 1

// CC_message STATE_ALTER:
elseif ( STATE_ALTER   != STATE_ALTER n-1
        && STATE_CC_KEY >= C_STATE_CC_ID_213)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 25
```

```
        LV_SEND_CC_ID_213= true
        if      (STATE_ALTER = 02H)
        then    LV_CC_ID_213 = true
        elseif  (STATE_ALTER = ! 02H  )
        then    LV_CC_ID_213 = false
        endif
        LV_EVE_CC_MSG = 1

// CC_message CRU_OFF_IRR:
elseif (LV_CRU_OFF_IRR  != LV_CRU_OFF_IRR $_{n-1}$
        && STATE_CC_KEY >= C_STATE_CC_ID_26)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 30
        LV_SEND_CC_ID_26 = true
        if      (LV_CRU_OFF_IRR  = 1 )
                LV_CC_ID_26 = true
        elseif  (LV_CRU_OFF_IRR  = 0 )
        then    LV_CC_ID_26 = false
        endif
        LV_EVE_CC_MSG = 1

// CC_message „oil level below minimum“:
elseif (LV_OIL_CNS_WARN_2 != LV_OIL_CNS_WARN_2 $_{n-1}$
        && STATE_CC_KEY >= C_STATE_CC_ID_28)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 35
        LV_SEND_CC_ID_28 = true
        if      (LV_OIL_CNS_WARN_2 = 1 )
        then    LV_CC_ID_28 = true
        elseif  (LV_OIL_CNS_WARN_2 = 0 )
        then    LV_CC_ID_28 = false
        endif
        LV_EVE_CC_MSG = 1

// CC_message EML:
elseif ( LV_WAL_1_CAN  != LV_WAL_1_CAN $_{n-1}$
```

```
            && STATE_CC_KEY >= C_STATE_CC_ID_29)
then        CYC_CTR_CC_SAVE = CYC_CTR_CC
            CYC_CTR_CC = 40
            LV_SEND_CC_ID_29 = true
            if      ( LV_WAL_1_CAN  = 1 )
            then    LV_CC_ID_29 = true
            elseif  ( LV_WAL_1_CAN  = 0 )
            then    LV_CC_ID_29 = false
            endif
            LV_EVE_CC_MSG = 1
```

~~// CC_message MIL:~~

```
~~elseif (LV_MIL_CAN_2 != LV_MIL_CAN_2 n-1~~
~~            && STATE_CC_KEY >= C_STATE_CC_ID_31)~~
~~then        CYC_CTR_CC_SAVE = CYC_CTR_CC~~
~~            CYC_CTR_CC = 45~~
~~            LV_SEND_CC_ID_31 = true~~
~~            if      (LV_MIL_CAN_2 = 1 )~~
~~            then    LV_CC_ID_31 = true~~
~~            elseif  (LV_MIL_CAN_2  = 0 )~~
~~            then    LV_CC_ID_31 = false~~
~~            endif~~
~~            LV_EVE_CC_MSG = 1~~
```

// CC_message "transportation mode":

```
elseif (LV_TRA_MODE_CAN  != LV_TRA_MODE_CAN n-1
            && STATE_CC_KEY >= C_STATE_CC_ID_TRA)
then        CYC_CTR_CC_SAVE = CYC_CTR_CC
            CYC_CTR_CC = 45
            LV_SEND_CC_ID_TRA = true
            if      (LV_TRA_MODE_CAN = 1 )
            then    LV_CC_ID_TRA = true
            elseif  (LV_TRA_MODE_CAN = 0 )
            then    LV_CC_ID_TRA = false
            endif
            LV_EVE_CC_MSG = 1
```

```
// CC_message „Zwangsschaltung aktiv“:
elseif ( LV_LIH_COC_ON != LV_LIH_COC_ON $_{n-1}$
        && STATE_CC_KEY >= C_STATE_CC_ID_155)
then   CYC_CTR_CC_SAVE = CYC_CTR_CC
       CYC_CTR_CC = 50
       LV_SEND_CC_ID_155 = true
       if      ( LV_LIH_COC_ON = 1 )
       then    LV_CC_ID_155 = true
       elseif  ( LV_LIH_COC_ON = 0 )
       then    LV_CC_ID_155 = false
       endif
       LV_EVE_CC_MSG = 1

// CC_message „oil level at minimum“:
elseif (LV_OIL_CNS_WARN_1 != LV_OIL_CNS_WARN_1 $_{n-1}$
        && STATE_CC_KEY >= C_STATE_CC_ID_27)
then   CYC_CTR_CC_SAVE = CYC_CTR_CC
       CYC_CTR_CC = 55
       LV_SEND_CC_ID_27 = true
       if      (LV_OIL_CNS_WARN_1 = 1 )
       then    LV_CC_ID_27 = true
       elseif  (LV_OIL_CNS_WARN_1 = 0 )
       then    LV_CC_ID_27 = false
       endif
       LV_EVE_CC_MSG = 1

// CC_message „close filler cap“:
elseif ( LV_FUC_CAN_BN != LV_FUC_CAN_BN $_{n-1}$
        && STATE_CC_KEY >= C_STATE_CC_ID_32)
then   CYC_CTR_CC_SAVE = CYC_CTR_CC
       CYC_CTR_CC = 60
       LV_SEND_CC_ID_32 = true
       if      ( LV_FUC_CAN_BN = 1 )
       then    LV_CC_ID_32 = true
```

```
          elseif  ( LV_FUC_CAN_BN  = 0 )
          then    LV_CC_ID_32 = false
          endif
          LV_EVE_CC_MSG = 1

// CC  message „error oil level sensor“:
elseif  ( LV_ERR_QOIL_SENS  != LV_ERR_QOIL_SENS n-1
          && STATE_CC_KEY >= C_STATE_CC_ID_182)
then      CYC_CTR_CC_SAVE = CYC_CTR_CC
          CYC_CTR_CC = 65
          LV_SEND_CC_ID_182 = true
          if      ( LV_ERR_QOIL_SENS  = 1 )
          then    LV_CC_ID_182 = true
          elseif  ( LV_ERR_QOIL_SENS = 0 )
          then    LV_CC_ID_182 = false
          endif
          LV_EVE_CC_MSG = 1

// CC_message „reload 12V-battery“:
elseif  ( B_ccbtnach  != B_ccbtnach n-1
          && STATE_CC_KEY >= C_STATE_CC_ID_229)
then      CYC_CTR_CC_SAVE = CYC_CTR_CC
          CYC_CTR_CC = 70
          LV_SEND_CC_ID_229 = true
          if      ( B_ccbtnach  = 1 )
          then    LV_CC_ID_229 = true
          elseif  ( B_ccbtnach = 0 )
          then    LV_CC_ID_229 = false
          endif
          LV_EVE_CC_MSG = 1

// CC_message „no battery monitoring“:
elseif  ( B_ccpmerr  != B_ccpmerr n-1
          && STATE_CC_KEY >= C_STATE_CC_ID_247)
then      CYC_CTR_CC_SAVE = CYC_CTR_CC
          CYC_CTR_CC = 75
```

```
          LV_SEND_CC_ID_247 = true
          if      ( B_ccpmerr  = 1 )
          then    LV_CC_ID_247 = true
          elseif  ( B_ccpmerr = 0 )
          then    LV_CC_ID_247 = false
          endif
          LV_EVE_CC_MSG = 1

// CC  message „stand-by-current error“:
elseif ( B_ccruhver  != B_ccruhver n-1
          && STATE_CC_KEY >= C_STATE_CC_ID_220)
then      CYC_CTR_CC_SAVE = CYC_CTR_CC
          CYC_CTR_CC = 80
          LV_SEND_CC_ID_220 = true
          if      ( B_ccruhver  = 1 )
          then    LV_CC_ID_220 = true
          elseif  ( B_ccruhver = 0 )
          then    LV_CC_ID_220 = false
          endif
          LV_EVE_CC_MSG = 1

// CC_message „check battery“:
elseif ( B_ccbttdfk  != B_ccbttdfk n-1
          && STATE_CC_KEY >= C_STATE_CC_ID_304)
then      CYC_CTR_CC_SAVE = CYC_CTR_CC
          CYC_CTR_CC = 85
          LV_SEND_CC_ID_304 = true
          if      ( B_ccbttdfk  = 1 )
          then    LV_CC_ID_304 = true
          elseif  ( B_ccbttdfk = 0 )
          then    LV_CC_ID_304 = false
          endif
          LV_EVE_CC_MSG = 1

// CC_message „check battery contacts“:
elseif ( B_ccbttknt  != B_ccbttknt n-1
```

```
         && STATE_CC_KEY >= C_STATE_CC_ID_305)
then     CYC_CTR_CC_SAVE = CYC_CTR_CC
         CYC_CTR_CC = 90
         LV_SEND_CC_ID_305 = true
         if        ( B_ccbttknt  = 1 )
         then      LV_CC_ID_305 = true
         elseif    ( B_ccbttknt = 0 )
         then      LV_CC_ID_305 = false
         endif
         LV_EVE_CC_MSG = 1

// CC  message „consumer switch off“:
elseif ( B_ccprio  != B_ccprio $_{n-1}$
         && STATE_CC_KEY >= C_STATE_CC_ID_306)
then     CYC_CTR_CC_SAVE = CYC_CTR_CC
         CYC_CTR_CC = 95
         LV_SEND_CC_ID_306 = true
         if        ( B_ccprio  = 1 )
         then      LV_CC_ID_306 = true
         elseif    ( B_ccprio = 0 )
         then      LV_CC_ID_306 = false
         endif
         LV_EVE_CC_MSG = 1

// CC_message „brake light switch defect“:
elseif ( LV_BLS_PLAUS_CAN != LV_BLS_PLAUS_CAN $_{n-1}$
         && STATE_CC_KEY >= C_STATE_CC_ID_148)
then     CYC_CTR_CC_SAVE = CYC_CTR_CC
         CYC_CTR_CC = 100
         LV_SEND_CC_ID_148 = true
         if        ( LV_BLS_PLAUS_CAN = 1 )
         then      LV_CC_ID_148 = true
         elseif    ( LV_BLS_PLAUS_CAN = 0 )
         then      LV_CC_ID_148 = false
         endif
         LV_EVE_CC_MSG = 1
```

```
// CC_message „gear-shift up“:
elseif ( LV_GS_UP != LV_GS_UP n-1
          && STATE_CC_KEY >= C_STATE_CC_ID_276)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 105
        LV_SEND_CC_ID_276 = true
        if      ( LV_GS_UP = 1 )
        then    LV_CC_ID_276 = true
        elseif  ( LV_GS_UP = 0 )
        then    LV_CC_ID_276 = false
        endif
        LV_EVE_CC_MSG = 1

// CC_message „gear-shift down“:
elseif ( LV_GS_DOWN != LV_GS_DOWN n-1
          && STATE_CC_KEY >= C_STATE_CC_ID_278)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 110
        LV_SEND_CC_ID_278 = true
        if      ( LV_GS_DOWN = 1 )
        then    LV_CC_ID_278 = true
        elseif  ( LV_GS_DOWN = 0 )
        then    LV_CC_ID_278 = false
        endif
        LV_EVE_CC_MSG = 1

// CC_message „request for gear-shift up (temperature warning 3)“:
elseif ( LV_TEMP_ENG_WARN_3 != LV_TEMP_ENG_WARN_3 n-1
          && STATE_CC_KEY >= C_STATE_CC_ID_367)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 115
        LV_SEND_CC_ID_367 = true
        if      ( LV_TEMP_ENG_WARN_3 = 1 )
        then    LV_CC_ID_367 = true
        elseif  (LV_TEMP_ENG_WARN_3 = 0 )
```

```
        then    LV_CC_ID_367 = false
        endif
        LV_EVE_CC_MSG = 1

// CC  message "MSA failure":
elseif ( LV_ERR_STST != LV_ERR_STST$_{n-1}$
        && STATE_CC_KEY >= C_STATE_CC_ID_397
        && LV_VAR_STST = 1)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 120
        LV_SEND_CC_ID_397 = true
        if      (LV_ERR_STST = 1 )
        then    LV_CC_ID_397 = true
        elseif  (LV_ERR_STST = 0 )
        then    LV_CC_ID_397 = false
        endif
        LV_EVE_CC_MSG = 1

// CC  message "oil pressure lamp - yellow":
elseif ( B_poel_gelb != B_poel_gelb$_{n-1}$
        && STATE_CC_KEY >= C_STATE_CC_ID_427)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
        CYC_CTR_CC = 125
        LV_SEND_CC_ID_427 = true
        if      (B_poel_gelb = 1 )
        then    LV_CC_ID_427 = true
        elseif  (B_poel_gelb = 0 )
        then    LV_CC_ID_427 = false
        endif
        LV_EVE_CC_MSG = 1

// CC  message "condition MSA deactivated":
elseif ( LV_INH_STST_CDN != LV_INH_STST_CDN $_{n-1}$
        && STATE_CC_KEY >= C_STATE_CC_ID_450
        && LV_VAR_STST = 1)
then    CYC_CTR_CC_SAVE = CYC_CTR_CC
```

```
            CYC_CTR_CC = 130
            LV_SEND_CC_ID_450 = true
            if      (LV_INH_STST_CDN = 1 )
            then    LV_CC_ID_450 = true
            elseif  (LV_INH_STST_CDN = 0 )
            then    LV_CC_ID_450 = false
            endif
            LV_EVE_CC_MSG = 1

        // CC  testmessage (bench mode):
        elseif ( LV_CC_TEST_BENCH_ACT != LV_CC_TEST_BENCH_ACT n-1
            && STATE_CC_KEY >= C_STATE_CC_ID_BENCH)
        then CYC_CTR_CC_SAVE = CYC_CTR_CC
            CYC_CTR_CC = 135
            LV_SEND_CC_ID_BENCH = true
            if      (LV_CC_TEST_BENCH_ACT = 1 )
            then    LV_CC_ID_BENCH = true
            elseif  (LV_CC_TEST_BENCH_ACT = 0 )
            then    LV_CC_ID_BENCH = false
            endif
            LV_EVE_CC_MSG = 1
        endif
endif
```

```
  // If an error is present the corresponding CC-message is transmitted depending on the
  // counter value. Due to the 100ms-recurrency and the calculation-steps of 5 it is secured that
  // the messages are only transmitted with a recurrency of 500ms
  If    ( CYC_CTR_CC = 0 && LV_SEND_CC_ID_34 = True )
      then  CC_ID = 34
            if      ( STATE_MIL = 2 )           // MIL flash light requested

            then    LV_MIL_FLL = true           //Activation  MIL flash light
            else    LV_MIL_FLL = false          // MIL: no flash light
            endif
            ST_CC_MESS_STD = LV_CC_ID_34
```

```
elseif  ( CYC_CTR_CC = 5 && LV_SEND_CC_ID_33 = True )
then    CC_ID = 33
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_33

elseif  ( CYC_CTR_CC = 10 && LV_SEND_CC_ID_257 = True )
then    CC_ID = 257
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_257

elseif  ( CYC_CTR_CC = 15 && LV_SEND_CC_ID_39 = True )
then    CC_ID = 39
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_39

elseif  ( CYC_CTR_CC = 20 && LV_SEND_CC_ID_212 = True )
then    CC_ID = 212
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_212

elseif  ( CYC_CTR_CC = 25 && LV_SEND_CC_ID_213 = True )
then    CC_ID = 213
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_213

elseif  ( CYC_CTR_CC = 30 && LV_SEND_CC_ID_26 = True )
then    CC_ID = 26
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_26

elseif  ( CYC_CTR_CC = 35 && LV_SEND_CC_ID_28 = True )
then    CC_ID = 28
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_28
```

```
elseif  ( CYC_CTR_CC = 40 && LV_SEND_CC_ID_29 = True )
then    CC_ID = 29
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_29

~~elseif  ( CYC_CTR_CC = 45 && LV_SEND_CC_ID_31 = True )~~
~~then    CC_ID = 31~~
~~        LV_MIL_FLL = false~~
~~        ST_CC_MESS_STD = LV_CC_ID_31~~

elseif  ( CYC_CTR_CC == 45 && LV_SEND_CC_ID_TRA == True )
then    CC_ID = C_CC_ID_TRA
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_TRA

elseif  ( CYC_CTR_CC = 50 && LV_SEND_CC_ID_155 = True )
then    CC_ID = 155
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_155

elseif  ( CYC_CTR_CC = 55 && LV_SEND_CC_ID_27 = True )
then    CC_ID = 27
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_27

elseif  ( CYC_CTR_CC = 60 && LV_SEND_CC_ID_32 = True )
then    CC_ID = 32
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_32

elseif  ( CYC_CTR_CC = 65 && LV_SEND_CC_ID_182 = True )
then    CC_ID = 182
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_182

elseif  ( CYC_CTR_CC = 70 && LV_SEND_CC_ID_229 = True )
```

```
then    CC_ID = 229
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_229

elseif  ( CYC_CTR_CC = 75 && LV_SEND_CC_ID_247 = True )
then    CC_ID = 247
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_247

elseif  ( CYC_CTR_CC = 80 && LV_SEND_CC_ID_220 = True )
then    CC_ID = 220
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_220

elseif  ( CYC_CTR_CC = 85 && LV_SEND_CC_ID_304 = True )
then    CC_ID = 304
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_304

elseif  ( CYC_CTR_CC = 90 && LV_SEND_CC_ID_305 = True )
then    CC_ID = 305
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_305

elseif  ( CYC_CTR_CC = 95 && LV_SEND_CC_ID_306 = True )
then    CC_ID = 306
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_306

elseif  ( CYC_CTR_CC = 100 && LV_SEND_CC_ID_148 = True )
then    CC_ID = 148
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_148

elseif  ( CYC_CTR_CC = 105 && LV_SEND_CC_ID_276 = True )
then    CC_ID = 276
```

```
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_276

elseif  ( CYC_CTR_CC = 110 && LV_SEND_CC_ID_278 = True )
then    CC_ID = 278
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_278

elseif  ( CYC_CTR_CC = 115 && LV_SEND_CC_ID_367 = True )
then    CC_ID = 367
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_367

elseif  ( CYC_CTR_CC = 120 && LV_SEND_CC_ID_397 = True )
then    CC_ID = 397
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_397

elseif  ( CYC_CTR_CC = 125 && LV_SEND_CC_ID_427 = True )
then    CC_ID = 427
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_427

elseif  ( CYC_CTR_CC = 130 && LV_SEND_CC_ID_450 = True )
then    CC_ID = 450
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_450

elseif  ( CYC_CTR_CC = 135 && LV_SEND_CC_ID_BENCH = True )
then    CC_ID = C_CC_ID_TEST_BENCH
        LV_MIL_FLL = false
        ST_CC_MESS_STD = LV_CC_ID_BENCH

else    CC_ID = 0

endif
```

```
If      ( LV_SEND_CC_ID_26 = True || ..... || LV_SEND_CC_ID_450 ||
        LV_SEND_CC_ID_BENCH =True || LV_SEND_CC_ID_TRA == True )
Then    ID2_CC_MESS_STD = C_CC_MESS
        NO_CC_MESS_STD = CC_ID
        // ST_CC_MESS_STD see above
        TRANF_CC_MESS_STD = C_T_CYC_DISP_ICL_CAN
        If      LV_MIL_FLL = 1
                ST_IDC_CCLK_CC_MESS_STD = C_ICL_DISP_FRQ
        else    ST_IDC_CCLK_CC_MESS_STD = 00H
        endif
        CAN_SendMessage()
endif

if  LV_EVE_CC_MSG = 1
then      CYC_CTR_CC = CYC_CTR_CC_SAVE
          if    (CYC_CTR_CC>=5)
          then  CYC_CTR_CC = CYC_CTR_CC - 4
          else  CYC_CTR_CC = 136
          endif
else      if    CYC_CTR_CC < 140
          then  CYC_CTR_CC++
          else  CYC_CTR_CC = 0
          endif
endif
```

status indicate-direction frequency check-control message standard :

C_ICL_DISP_FRQ name

00H	"Kein Blinken"
01H	"Langsames Blinken"
02H	"Schnelles Blinken"
03H	"Signal ungültig"

// If an error is not present any longer, the transmission-request of the corresponding
// CC-message has to be reset to 0.

if LV_MIL_CAN_1 != LV_MIL_CAN_1 $_{n-1}$

```
                 && STATE_CC_KEY >= C_STATE_CC_ID_34
then             if     LV_MIL_CAN_1 n-1 = 1
                 then   LV_SEND_CC_ID_34     =     false
                 endif
                 LV_MIL_CAN_1 n-1 = LV_MIL_CAN_1
elseif           LV_MIS_STATE_A  != LV_MIS_STATE_A n-1
                 && STATE_CC_KEY >= C_STATE_CC_ID_33
then             if     LV_MIS_STATE_A n-1 = 1
                 then   LV_SEND_CC_ID_33     =     false
                 endif
                 LV_MIS_STATE_A n-1 = LV_MIS_STATE_A
elseif           LV_TEMP_ENG_WARN_1  != LV_TEMP_ENG_WARN_1 n-1
                 && STATE_CC_KEY >= C_STATE_CC_ID_257
then             if     LV_TEMP_ENG_WARN_1 n-1 = 1
                 then   LV_SEND_CC_ID_257    =     false
                 endif
                 LV_TEMP_ENG_WARN_1 n-1 = LV_TEMP_ENG_WARN_1
elseif           LV_TEMP_ENG_WARN_2  != LV_TEMP_ENG_WARN_2 n-1
                 && STATE_CC_KEY >= C_STATE_CC_ID_39
then             if     LV_TEMP_ENG_WARN_2 n-1 = 1
                 then   LV_SEND_CC_ID_39     =     false
                 endif
                 LV_TEMP_ENG_WARN_2 n-1 = LV_TEMP_ENG_WARN_2
elseif           LV_POIL_SWI_CAN  != LV_POIL_SWI_CAN n-1
                 && STATE_CC_KEY >= C_STATE_CC_ID_212
then             if     LV_POIL_SWI_CAN n-1 = 1
                 then   LV_SEND_CC_ID_212    =     false
                 endif
                 LV_POIL_SWI_CAN n-1 = LV_POIL_SWI_CAN
elseif           STATE_ALTER  != STATE_ALTER n-1
                 && STATE_CC_KEY >= C_STATE_CC_ID_213
then             if     STATE_ALTER n-1 = 1
                 then   LV_SEND_CC_ID_213    =     false
                 endif
                 STATE_ALTER n-1 = STATE_ALTER
elseif           LV_CRU_OFF_IRR != LV_CRU_OFF_IRR n-1
```

```
                    && STATE_CC_KEY >= C_STATE_CC_ID_26
then                if      LV_CRU_OFF_IRR n-1 = 1
                    then    LV_SEND_CC_ID_26        =       false
                    endif
                    LV_CRU_OFF_IRR n-1 = LV_CRU_OFF_IRR
elseif              LV_OIL_CNS_WARN_2 != LV_OIL_CNS_WARN_2 n-1
                    && STATE_CC_KEY >= C_STATE_CC_ID_28
then                if      LV_OIL_CNS_WARN_2 n-1 = 1
                    then    LV_SEND_CC_ID_28        =       false
                    endif
                    LV_OIL_CNS_WARN_2 n-1 = LV_OIL_CNS_WARN_2
elseif              LV_WAL_1_CAN  != LV_WAL_1_CAN n-1
                    && STATE_CC_KEY >= C_STATE_CC_ID_29
then                if      LV_WAL_1_CAN n-1 = 1
                    then    LV_SEND_CC_ID_29        =       false
                    endif
                    LV_WAL_1_CAN n-1 = LV_WAL_1_CAN
~~elseif              LV_MIL_CAN_2  != LV_MIL_CAN_2 n-1~~
                    ~~&& STATE_CC_KEY >= C_STATE_CC_ID_31~~
~~then                if      LV_MIL_CAN_2 n-1 = 1~~
                    ~~then    LV_SEND_CC_ID_31        =       false~~
                    ~~endif~~
                    ~~LV_MIL_CAN_2 n-1 = LV_MIL_CAN_2~~
elseif              LV_TRA_MODE_CAN != LV_TRA_MODE_CAN n-1
                    && STATE_CC_KEY >= C_STATE_CC_ID_TRA
then                if      LV_TRA_MODE_CAN n-1     =       1
                    then    LV_SEND_CC_ID_TRA       =       false
                    endif
                    LV_TRA_MODE_CAN n-1 = LV_TRA_MODE_CAN
elseif              LV_LIH_COC_ON != LV_LIH_COC_ON n-1
                    && STATE_CC_KEY >= C_STATE_CC_ID_155
then                if      LV_LIH_COC_ON n-1 = 1
                    then    LV_SEND_CC_ID_155       =       false
                    endif
                    LV_LIH_COC_ON n-1 = LV_LIH_COC_ON
elseif              LV_OIL_CNS_WARN_1 != LV_OIL_CNS_WARN_1 n-1
```

```
                && STATE_CC_KEY >= C_STATE_CC_ID_27
then            if      LV_OIL_CNS_WARN_1 $_{n-1}$ = 1
                then    LV_SEND_CC_ID_27        =       false
                endif
                LV_OIL_CNS_WARN_1 $_{n-1}$ = LV_OIL_CNS_WARN_1
elseif          LV_FUC_CAN_BN != LV_FUC_CAN_BN $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_32
then            if      LV_FUC_CAN_BN $_{n-1}$ = 1
                then    LV_SEND_CC_ID_32        =       false
                endif
                LV_FUC_CAN_BN $_{n-1}$ = LV_FUC_CAN_BN
elseif          LV_ERR_QOIL_SENS  != LV_ERR_QOIL_SENS $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_182
then            if      LV_ERR_QOIL_SENS $_{n-1}$ = 1
                then    LV_SEND_CC_ID_182       =       false
                endif
                LV_ERR_QOIL_SENS $_{n-1}$ = LV_ERR_QOIL_SENS
elseif          B_ccbtnach  != B_ccbtnach $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_229
then            if      B_ccbtnach $_{n-1}$ = 1
                then    LV_SEND_CC_ID_229       =       false
                endif
                B_ccbtnach $_{n-1}$ = B_ccbtnach
elseif          B_ccpmerr  != B_ccpmerr $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_247
then            if      B_ccpmerr $_{n-1}$ = 1
                then    LV_SEND_CC_ID_247       =       false
                endif
                B_ccpmerr $_{n-1}$ = B_ccpmerr
elseif          B_ccruhver  != B_ccruhver $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_220
then            if      B_ccruhver $_{n-1}$ = 1
                then    LV_SEND_CC_ID_220       =       false
                endif
                B_ccruhver $_{n-1}$ = B_ccruhver
elseif          B_ccbttdfk  != B_ccbttdfk $_{n-1}$
```

```
                && STATE_CC_KEY >= C_STATE_CC_ID_304
then            if      B_ccbttdfk $_{n-1}$ = 1
                then    LV_SEND_CC_ID_304      =      false
                endif
                B_ccbttdfk $_{n-1}$ = B_ccbttdfk
elseif          B_ccbttknt  != B_ccbttknt $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_305
then            if      B_ccbttknt $_{n-1}$ = 1
                then    LV_SEND_CC_ID_305      =      false
                endif
                B_ccbttknt $_{n-1}$ = B_ccbttknt
elseif          B_ccprio  != B_ccprio $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_306
then            if      B_ccprio $_{n-1}$ = 1
                then    LV_SEND_CC_ID_306      =      false
                endif
                B_ccprio $_{n-1}$ = B_ccprio
elseif          LV_BLS_PLAUS_CAN  != LV_BLS_PLAUS_CAN $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_148
then            if      LV_BLS_PLAUS_CAN $_{n-1}$ = 1
                then    LV_SEND_CC_ID_148      =      false
                endif
                LV_BLS_PLAUS_CAN $_{n-1}$ = LV_BLS_PLAUS_CAN
elseif          LV_GS_UP  != LV_GS_UP $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_276
then            if      LV_GS_UP $_{n-1}$ = 1
                then    LV_SEND_CC_ID_276      =      false
                endif
                LV_GS_UP $_{n-1}$ = LV_GS_UP
elseif          LV_GS_DOWN  != LV_GS_DOWN $_{n-1}$
                && STATE_CC_KEY >= C_STATE_CC_ID_278
then            if      LV_GS_DOWN $_{n-1}$ = 1
                then    LV_SEND_CC_ID_278      =      false
                endif
                LV_GS_DOWN $_{n-1}$ = LV_GS_DOWN
elseif          LV_TEMP_ENG_WARN_3  != LV_TEMP_ENG_WARN_3 $_{n-1}$
```

```
                  && STATE_CC_KEY >= C_STATE_CC_ID_367
then              if      LV_TEMP_ENG_WARN_3 $_{n-1}$ = 1
                  then    LV_SEND_CC_ID_367       =       false
                  endif
                  LV_TEMP_ENG_WARN_3 $_{n-1}$ = LV_TEMP_ENG_WARN_3
elseif            LV_ERR_STST  != LV_ERR_STST $_{n-1}$
                  && STATE_CC_KEY >= C_STATE_CC_ID_397
then              if      LV_ERR_STST $_{n-1}$ = 1
                  then    LV_SEND_CC_ID_397       =       false
                  endif
                  LV_ERR_STST $_{n-1}$ = LV_ERR_STST
elseif            B_poel_gelb != B_poel_gelb $_{n-1}$
                  && STATE_CC_KEY >= C_STATE_CC_ID_427
then              if      B_poel_gelb $_{n-1}$ = 1
                  then    LV_SEND_CC_ID_427       =       false
                  endif
                  B_poel_gelb $_{n-1}$ = B_poel_gelb
elseif            LV_INH_STST_CDN != LV_INH_STST_CDN $_{n-1}$
                  && STATE_CC_KEY >= C_STATE_CC_ID_450
then              if      LV_INH_STST_CDN $_{n-1}$ = 1
                  then    LV_SEND_CC_ID_450       =       false
                  endif
                  LV_INH_STST_CDN $_{n-1}$ = LV_INH_STST_CDN
elseif            LV_CC_TEST_BENCH_ACT  != LV_CC_TEST_BENCH_ACT $_{n-1}$
                  && STATE_CC_KEY >= C_STATE_CC_ID_BENCH
then              if      LV_CC_TEST_BENCH_ACT $_{n-1}$ = 1
                  then    LV_SEND_CC_ID_BENCH  =       false
                  endif
                  LV_CC_TEST_BENCH_ACT $_{n-1}$ = LV_CC_TEST_BENCH


endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CC_MESS	1	0...FFH	0...255	1	[-]
ID2 check-control message standard ID2_CC_MESS_STD (Standard CC-report: 40H)					
C_T_CYC_DISP_ICL_CAN	1	0...FH	0...15	1	[s]
recurrence for transmission of check control message (appl. suggestion: 10 s)					
C_ICL_DISP_FRQ	1	0...3H	0...3	1	[-]
calibratable constant for display messages on ICL					
C_N_32_THD_CC_POIL_SWI	1	0...FFH	0...8160	32	[rpm]
N_32-Threshold for activating CC-message LV_POIL_SWI					
IP_T_DLY_POIL_SWI_CAN__TOIL	6	0...FFFFH	0...6553.5	0.1	[s]
LDP_TOIL__T_DLY_POIL_SWI_CAN	6	0...C8H	-40...160	1	[°C]
Delay-time for first activation of CC-message LV_POIL_SWI					
LC_ENA_CC_MIL_1	1	0...1H	0...1	1	[-]
LC for enabling MIL_1 checkcontrol-message					
~~LC_ENA_CC_MIL_2~~	~~1~~	~~0...1H~~	~~0...1~~	~~1~~	~~[-]~~
~~LC for enabling MIL_2 checkcontrol-message: 1 (US); 0 (ECE)~~					
C_STATE_CC_ID_xx	1	0...FH	0...15	1	-
sending-condition for CC-message (e.g. LV_IGK, LV_KEY_AUX, etc.)					
LC_CC_TEST_BENCH	1	0...1H	0...1	1	[-]
LC for enabling CC test bench					
C_STATE_CC_TEST_BENCH	1	0...FFH	0...255	1	[-]
Bit coded trigger conditions for activating Checkcontrol test bench					
C_STATE_CC_TEST_MASK_OBD_1	1	0...FFH	0...255	1	[-]
Bit mask for readiness code completion status 1					
C_STATE_CC_TEST_MASK_OBD_2	1	0...FFH	0...255	1	[-]
Bit mask for readiness code completion status 2					
C_CTR_ERR_DYN_CC_TEST_BENCH	1	0...FFH	0...255	1	[-]
Threshold for number of failures stored in dynamic memory					
C_STATE_OPM_MAX_CC_TEST_BENCH	1	0...8H	0...8	1	[-]
maximum trigger threshold (OPM_AV) for CC bench mode					
C_STATE_OPM_MIN_CC_TEST_BENCH	1	0...8H	0...8	1	[-]
minimum trigger threshold (OPM_AV) for CC bench mode					
C_STATE_CC_TEST_BENCH_OBD_1	1	0...FFH	0...255	1	[-]
Bit coded readiness code status 1 threshold for CC bench mode					
C_STATE_CC_TEST_BENCH_OBD_2	1	0...FFH	0...255	1	[-]
Bit coded readiness code status 2 threshold for CC bench mode					
C_STATE_COC_MIN_CC_TEST_BENCH	1	0...FFH	0...255	1	[-]
minimum state of coolant circuit management for CC test bench					
C_STATE_COC_MAX_CC_TEST_BENCH	1	0...FFH	0...255	1	[-]
maximum state of coolant circuit management for CC test bench					
C_STATE_CC_ID_BENCH	1	0...FH	0...15	1	[-]
sending-condition for CC- test - message					
C_CC_ID_TEST_BENCH	1	0...FFFFH	0...65535	1	[-]
Checkcontrol ID for test message					

`C_STATE_CC_ID_xxx = 0, 1, 2 oder 3`

1 = CC ab Kl.R senden
2 = CC ab Kl.15 senden
3 = CC ab Motor läuft senden
...
FH = CC nie senden

43.2.4 Check Sum Calculation

FUNCTION DESCRIPTION:

The check sum calculation is additional to the "alive counter" a further possibility to supervise the data transmitted via CAN. From all safety relevant controllers, the ECU included, a check sum over all sent data including his own CAN-ID (11 bit) is calculated and sent together with the data frame. The receiver of the data frame can calculate the check sum over the received data and compare it with the received check sum. If both check sums are not the same a transmission error occured. For check sum calculation see newest BN 2000 specification.
The check sums CKS_CLC_TQ_DCC, CKS_CLC_CAS, CKS_CLC_TQ_TCS, CKS_CLC_TQ_ETCU, CKS_CLC_TQ_AMT, CKS_CLC_MSW, CKS_CLC_LDM, CKS_CLC_TQ_PSTE_3, CKS_CLC_PBR, CKS_CLC_ETCU, CKS_CLC_TQ_TCT CKS_CLC_TQ_PBR are calculated with received data (see description below).

CKS_CLC_ECU1, CKS_CLC_ECU2 and CKS_CLC_ECU3, CKS_CLC_TQ_WHEEL_1, CKS_CLC_TQ_WHEEL_2, CKS_CLC_GS_IDC are calculated with sent data from ECU.

Formula section:

Check sum ECU1

CKS_CLC_ECU1

Torque_1 – ID A8H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte7	ST_CT_BRPD_DME			ST_OBD_CTFN_GRB				
Byte6	ST_RCPT_ENG_EGS		ST_RCPT_ENG_DSC		ST_RCPT_ENG_ARS		ST_RCPT_ENG_ACC	
Byte5	RCPT_ACC_SWO_SYS_DME		ST_DMEA_SWO		ST_RTIR		ST_SW_CLT	
Byte4	TORQ_AVL_DMEE							
Byte3	TORQ_AVL_DMEE				ST_TORQ_AVL			
Byte2	TORQ_AVL							
Byte1	TORQ_AVL				ALIV_TORQ_1_DME			
Byte0	CHKSM_TORQ_1_DME							

Check sum ECU2

CKS_CLC_ECU2

Torque_2 – IDA9H

<table>
<tr><th></th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>Byte7</td><td colspan="8">TORQ_AVL_SPAR_POS</td></tr>
<tr><td>Byte6</td><td colspan="4">TORQ_AVL_SPAR_POS</td><td colspan="4">TORQ_AVL_SPAR_NEG</td></tr>
<tr><td>Byte5</td><td colspan="8">TORQ_AVL_SPAR_NEG</td></tr>
<tr><td>Byte4</td><td colspan="8">TORQ_AVL_MAX</td></tr>
<tr><td>Byte3</td><td colspan="4">TORQ_AVL_MAX</td><td colspan="4">TORQ_AVL_MIN</td></tr>
<tr><td>Byte2</td><td colspan="8">TORQ_AVL_MIN</td></tr>
<tr><td>Byte1</td><td colspan="2">ST_INFS</td><td colspan="2">ST_SW_LEV_RPM</td><td colspan="4">ALIV_TORQ_2_DME</td></tr>
<tr><td>Byte0</td><td colspan="8">CHKSM_TORQ_2_DME</td></tr>
</table>

Check sum ECU3

CKS_CLC_ECU3

Torque_3 – ID AAH

<table>
<tr><th></th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>Byte7</td><td colspan="8">RQAM_FU</td></tr>
<tr><td>Byte6</td><td colspan="4">ST_CLCTR_V</td><td colspan="2">ST_IDLG_ENG</td><td colspan="2">RPM_ENG_ERR</td></tr>
<tr><td>Byte5</td><td colspan="8">RPM_ENG</td></tr>
<tr><td>Byte4</td><td colspan="8">RPM_ENG</td></tr>
<tr><td>Byte3</td><td colspan="8">ANG_ACPD</td></tr>
<tr><td>Byte2</td><td colspan="8">TORQ_DVCH</td></tr>
<tr><td>Byte1</td><td colspan="4">TORQ_DVCH</td><td colspan="4">ALIV_TORQ_3_DME</td></tr>
<tr><td>Byte0</td><td colspan="8">CHKSM_TORQ_3_DME</td></tr>
</table>

Check sum TQ_WHEEL_1

CKS_CLC_TQ_WHEEL_1

RADMOM_PT_1 – ID B4H

<table>
<tr><th></th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>Byte7</td><td colspan="5">ST_INTF_PT</td><td colspan="3">ST_PENG</td></tr>
<tr><td>Byte6</td><td colspan="4">--</td><td colspan="4">ANG_ACPD_TAR</td></tr>
<tr><td>Byte5</td><td colspan="8">ANG_ACPD_TAR</td></tr>
<tr><td>Byte4</td><td colspan="8">WMOM_PT_AVL</td></tr>
<tr><td>Byte3</td><td colspan="8">WMOM_PT_AVL</td></tr>
<tr><td>Byte2</td><td colspan="8">ANG_ACPD_REAL</td></tr>
<tr><td>Byte1</td><td colspan="4">ANG_ACPD_REAL</td><td colspan="4">ALIV_WMOM_PT_1</td></tr>
</table>

Byte0	CHKSM_WMOM_PT_1

Check sum TQ_WHEEL_2

CKS_CLC_TQ_WHEEL_2

RADMOM_PT_2 – ID ACH

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte7	WMOM_PT_MAX							
Byte6	WMOM_PT_MAX							
Byte5	WMOM_PT_MIN_LOW							
Byte4	WMOM_PT_MIN_LOW							
Byte3	WMOM_COODT_TAR							
Byte2	WMOM_COODT_TAR							
Byte1					ALIV_WMOM_PT_2			
Byte0	CHKSM_WMOM_PT_2							

Check sum GS_IDC

CKS_CLC_GS_IDC

DISP_GSI – ID 2F3H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte2	CHKSM_ DISP_GSI							
Byte1							DISP_GSI	
Byte0	DISP_TARG				ALIV_DISP_GSI			

Check sum TQ_DCC

CKS_CLC_TQ_DCC

DREHMOMENT_ANF_ACC – ID B7H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte5	ST_TORQ_CTRL_ACC			ST_ACC_MOD_DME			ST_ACC_SWO_SYS_DME	
Byte4	RQ_PREV_INFS_ACC		ST_TORQ_TAR_ACC		TORQ_TAR_ADJR_POS_ACC			
Byte3	TORQ_TAR_ADJR_POS_ACC							
Byte2	TORQ_TAR_ACC							
Byte1	TORQ_TAR_ACC				ALIV_TORQ_ACC			
Byte0	CHKSM_TORQ_ACC							

Check sum TQ_PSTE_2 (AFS)

CKS_CLC_TQ_PSTE_2

DREHMOMENT_ANF_AFS – ID B9H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte5					ST_RQ_TORQ_AFS_ERR			
Byte4			ST_TORQ_TAR_AFS		TORQ_TAR_AFS			
Byte3	TORQ_TAR_AFS							
Byte2	TORQ_AVL_AFS							
Byte1	TORQ_AVL_AFS				ALIV_TORQ_AFS			
Byte0	CHKSM_TORQ_AFS							

Check sum CAS :

CKS_CLC_CAS

KLEMMENSTATUS – ID 130H

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 4	CHKSM_KL				ALIV_KL			
Byte 3	ST_KL15_HW							
Byte 2	ST_STCD_PENG_MSA							
Byte 1					NO_KEY			
Byte 0	ST_KEY_VLD		ST_KL_50		ST_KL_15		ST_KL_R	

Check sum TQ_TCS :

CKS_CLC_TQ_TCS

DREHMOMENT_ANF_DSC – ID B6H

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Byte 4	▨	▨	ST_TORQ_TAR_DSC		TORQ_TAR_ADJR_POS_DSC			
Byte 3	TORQ_TAR_ADJR_POS_DSC							
Byte 2	TORQ_TAR_DSC							
Byte 1	TORQ_TAR_DSC				ALIV_TORQ_DSC			
Byte 0	CHKSM_TORQ_DSC							

Check sum TQ_ETCU :

CKS_CLC_TQ_ETCU

DREHMOMENT_ANF_EGS – ID B5H

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 7	TEMP_GRB							
Byte 6	TORQ_DRAW_GRB							
Byte 5	ST_LIM_STORQ		ST_OBD_ERR_GRB		ST_OBD_GRB			
Byte 4	ST_OTMP_GRB		ST_TORQ_TAR_EGS		TORQ_TAR_ADJR_POS_EGS			
Byte 3	TORQ_TAR_ADJR_POS_EGS							
Byte 2	TORQ_TAR_EGS							
Byte 1	TORQ_TAR_EGS				ALIV_TORQ_EGS			
Byte 0	CHKSM_TORQ_EGS							

Check sum TQ_AMT :

CKS_CLC_TQ_AMT
DREHMOMENT_ANF_SSG – ID BDH

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte7	LIM_RPM_ENG							
Byte6	RQ_Motorstop		ST_CLT_SSG			ST_SHPS_SSG		
Byte5	ST_LIM_STORQ_SSG		ST_OBD_ERR_GRB_SSG		ST_OBD_GRB_SSG			
Byte4	▨	▨	ST_TORQ_TAR_SSG		TORQ_TAR_ADJR_POS_SSG			
Byte3	TORQ_TAR_ADJR_POS_SSG							
Byte2	TORQ_TAR_SSG							
Byte1	TORQ_TAR_SSG				ALIV_TORQ_SSG			
Byte0	CHKSM_TORQ_SSG							

Check sum MSW (SZL)

CKS_CLC_MSW

BEDIENUNG_TEMPOMAT – ID 194H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte3	▨	▨	▨	▨				
Byte2	OP_PUBU_CCTR_ACC							

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte1					ALIV_CCTR			
Byte0	CHKSM_CCTR							

Check sum VEH_MOD (CAS)

CKS_CLC_VEH_MOD

VEH_MOD - ID 315H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte1					MOD_VEH			
Byte0	ALIV_MOD_VEH				CHKSM_MOD_VEH			

Check sum LDM

CKS_CLC_LDM

ANF_RADMOM_PT – ID BFH

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte4	RQ_GSI_CLCTR_LN		WMOM_PT_ENBL		ST_DRASY_PT			
Byte3	WMOM_PT_TAR							
Byte2	WMOM_PT_TAR							
Byte1	FOLDYN_WMOM_PT_TAR				ALIV_RQ_WMOM_PT			
Byte0	CHKSM_RQ_WMOM_PT							

Check sum EHB3 (TQ_PSTE_3)

CKS_CLC_TQ_PSTE_3

DREHMOMENT_ANF_STE – ID B1H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte4				ST_RQ_TORQ_STE_ERR			ST_TORQ_TAR_STE	
Byte3	TORQ_TAR_STE							
Byte2	TORQ_AVL_STE							
Byte1	SRC_RQ_TORQ_STE				ALIV_TORQ_STE			
Byte0	CHKSM_TORQ_STE							

Check sum ETCU_3

CKS_CLC_ETCU_3

GETRIEBEDATEN_3 – ID 3B1H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte5	LIM_MAX_V_GRB_EMMOD							
Byte4	RQ_MIL_GRB		ST_MSA_GRB					
Byte3	IDX_TORQ_FIL				IDX_ACPD			
Byte2	TORQ_DVCH_RISE							
Byte1	TORQ_DVCH_RISE				ALIV_GRB_3			

Byte0	CHKSM_GRB_3							

Check sum REQ_PBR

CKS_CLC_REQ_PBR

STELLANF_EMF – ID 1A7H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte3	▨	▨			ST_CFFU_EMF_2		ST_HYD_RETA_2	
Byte2								
Byte1					ALIV_PRQ_EMF			
Byte0	CHKSM_PRQ_EMF							

Check sum PBR

CKS_CLC_PBR

STATUS_EMF – ID 201H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte3								
Byte2								
Byte1	ST_EMF_LOCA_2				ALIV_COU_EMF_2			
Byte0	CHKSM_ST_EMF							

Check sum ETCU

CKS_CLC_ETCU

GETRIEBEDATEN – ID BAH

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte7							ST_HYPP_ACV	
Byte6	▨	ST_MOD_GRB			ALIV_GRB			
Byte5	CHKSM_GRB							
Byte4								
Byte3								
Byte2					GRDT_REIN			
Byte1	GRDT_REIN							
Byte0	ST_CCLT		ST_GRLV_ACV		ST_GR_GRB			

Check sum TQ_TCT

CKS_CLC_TQ_TCT

DREHMOMENT_ANF_DKG – ID B8H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte7	RPM_TAR_ENG_DKG							
Byte6	RPM_TAR_ENG_DKG				ST_MOD_DKG			
Byte5	TORQ_CLT_DKG							
Byte4	TORQ_CLT_DKG				TORQ_TAR_ADJR_POS_DKG			
Byte3	TORQ_TAR_ADJR_POS_DKG							
Byte2	TORQ_TAR_DKG							
Byte1	TORQ_TAR_DKG				ALIV_TORQ_DKG			
Byte0	CHKSM_TORQ_DKG							

Check sum ST_RQ_EMF

CKS_CLC_TQ_PBR

ST_RQ_EMF – ID 1FDH

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte5	ST_ACT_EMF				QU_ST_ACT_EMF			
Byte4	SAFG_RQ_WMOM_EMF							
Byte3	~~RQ_WMOM_EMF~~ RQ_DCRN_EMF							
Byte2	~~RQ_WMOM_EMF~~ RQ_DCRN_EMF							
Byte1	~~QU_RQ_WMOM_EMF~~ QU_RQ_DCRN_EMF				ALIV_ST_RQ_EMF			
Byte0	CHKSM_ST_RQ_EMF							

43.3 BN2000 interface torque distribution

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_GS_FAST_INC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
fast torque increased intervention due to GS					
TQ_GS_SLOW_INC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
slow torque increased intervention due to GS					
TQ_GS_FAST_DEC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
fast torque decreased intervention due to GS					
TQ_GS_SLOW_DEC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
slow torque decreased intervention due to GS					
TQ_AMT_FAST_INC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
fast torque increased intervention due to AMT					
TQ_AMT_SLOW_INC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
slow torque increased intervention due to AMT					
TQ_AMT_FAST_DEC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
fast torque decreased intervention due to AMT					
TQ_AMT_SLOW_DEC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
slow torque decreased intervention due to AMT					
TQ_DCC_FAST_INC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
fast torque increased intervention due to DCC					
TQ_DCC_SLOW_INC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
slow torque increased intervention due to DCC					
TQ_MSR_FAST_INC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
fast torque increased intervention due to TCS					
TQ_MSR_SLOW_INC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
slow torque increased intervention due to TCS					
TQ_ASR_FAST_DEC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
fast torque decreased intervention due to TCS					
TQ_ASR_SLOW_DEC_BN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
slow torque decreased intervention due to TCS					
TQ_WHEEL_LDM_INC_DEC_BN	O/V	8000...7FFFH	-32768...32767	1	[Nm]
torque increased intervention due to LDM					
T_TQ_DCC_CS	V	0...FFH	0...2.55	0.01	[s]
Time of clutch switch intervention during DCC active					
LV_ERR_SYM_TQ_DCC_CS	O/V	0...1H	0...1	1	[-]
Logical bit for TQ_REQ_CAN error due to DCC torque intervention clutch switch					

Input data:

STATE_ETCU_INTV	LV_ETCU_LIH_CAN	TQ_GS_SLOW_BN	TQ_GS_FAST_BN
LV_TQ_GS_DEC_REQ	LV_TQ_GS_INC_REQ	LV_VAR_BN	LV_IGK
STATE_AMT_INTV	LV_TQ_DCC_INC_REQ	TQ_AMT_SLOW_BN	TQ_AMT_FAST_BN
LV_TQ_AMT_DEC_REQ	LV_TQ_AMT_INC_REQ	TQ_DCC_SLOW_BN	TQ_DCC_FAST_BN

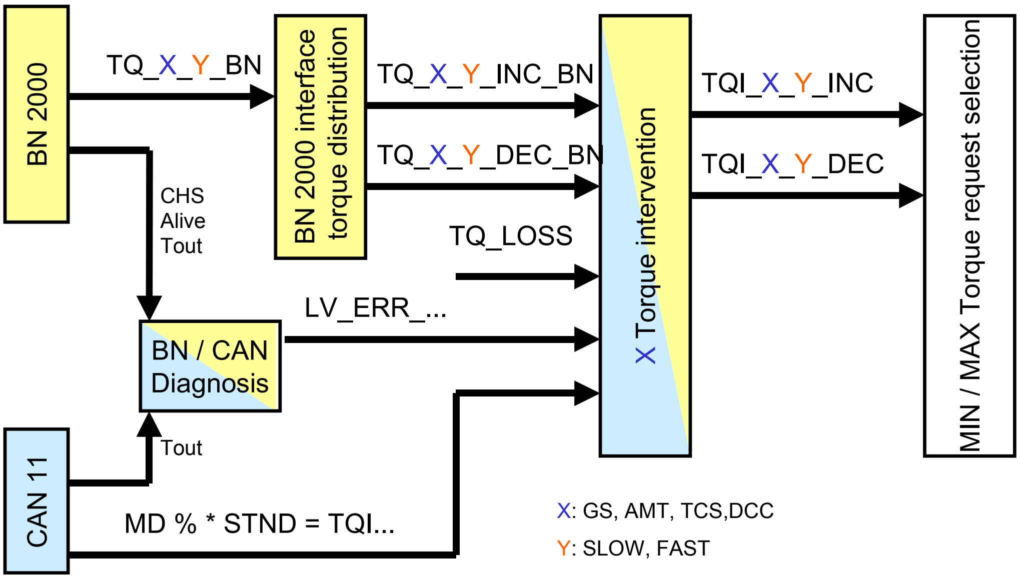
TQ_TCS_FAST_BN	TQ_TCS_SLOW_BN	LV_TQ_ASR_REQ	LV_TQ_MSR_REQ
STATE_TCS_INTV	STATE_DCC_INTV	LV_AMT_LIH_CAN	LV_DCC_LIH_CAN
LV_IM_CS_PN	STATE_DCC_CTL	N	LV_AT
LV_VAR_AMT	LV_VAR_BN_LDM	TQ_WHEEL_LDM_BN	LV_TQ_WHEEL_LDM_REQ
LV_TQ_WHEEL_LDM_BN_ERR	LV_VAR_TCT	STATE_TCT_INTV	LV_TCT_LIH_CAN

FUNCTION DESCRIPTION:

General information:

Some functions of BN 2000 contain a torque request. For the torque structure interfaces it's necessary to define the direction of torque request as increased or decreased. The BN delivers just one torque request and an additional direction information bit. This module combines the direction information bit of BN with the torque request and supported the torque structure interfaces.

Signal flow diagram:



Application conditions:

Initialisation: *TQ_**X**_FAST_DEC_BN = 1023.97Nm*

*TQ_**X**_SLOW_DEC_BN = 1023.97Nm*

*TQ_**X**_FAST_INC_BN = -1024 Nm*

*TQ_**X**_SLOW_INC_BN = -1024 Nm*

TQ_WHEEL_LDM_INC_DEC_BN = -32000 Nm

LV_RST_T_TQ_DCC_CS = 1

T_TQ_DCC_CS = 0

LV_ERR_SYM_TQ_DCC_CS = 0

Recurrence: 10ms

Activation: LV_VAR_BN = 1 and LV_IGK = 1

Description:

INDEX	Functionality				
Customer	EGS	DSC	SSG	ACC	LDM
Y	ETCU	TCS	AMT	DCC	LDM
X_INC	GS_INC	MSR	AMT_INC	DCC_INC	LDM_INC_ DEC
X_DEC	GS_DEC	ASR	AMT_DEC	-	-
Z	GS	TCS	AMT	DCC	LDM

(The DCC function delivers only an increased Torque request. By using this template the initialisation of LV_TQ_DCC_DEC_REQ with 0 is necessary.) Beside of this, it is necessary that under defined conditions the torque request of the DCC is controlled by the ECU.

Formula section:

Torque CAN selection for intervention :

```
If          STATE_Y_INTV  = NO_INTERVENTION (00H)
     or     LV_Y_LIH_CAN  = 1
     or     TQ_Z _FAST_BN = -1024 Nm
     or     TQ_Z _SLOW_BN = -1024 Nm
then        TQ_X _FAST_DEC_BN  = 1023.97 Nm
            TQ_X _SLOW_DEC_BN  = 1023.97 Nm
            TQ_X _FAST_INC_BN = -1024 Nm
            TQ_X _SLOW_INC_BN  = -1024 Nm
elseif      LV_TQ_X_INC_REQ              (X_INC-intervention)
            then   TQ_X _FAST_DEC_BN  = 1023.97 Nm
                   TQ_X _SLOW_DEC_BN  = 1023.97 Nm
                   TQ_X _FAST_INC_BN  = TQ_Z _FAST_BN (converted)
                   TQ_X _SLOW_INC_BN  = TQ_Z _SLOW_BN (converted)
            elseif LV_TQ_X_DEC_REQ              (X_DEC-intervention)
                   then   TQ_X _FAST_DEC_BN  = TQ_Z _FAST_BN (converted)
                          TQ_X _SLOW_DEC_BN  = TQ_Z _SLOW_BN (converted)
                          TQ_X _FAST_INC_BN  = -1024 Nm
                          TQ_X _SLOW_INC BN  = -1024 Nm
                   endif
            endif
endif
```

DCC Torque CAN selection for intervention

```
If(1)          STATE_DCC_INTV  = NO_INTERVENTION (00H)
      or       LV_DCC_LIH_CAN  = 1
      or       TQ_DCC _FAST_BN = -1024 Nm
      or       TQ_DCC _SLOW_BN = -1024 Nm
then(1)        TQ_DCC _FAST_DEC_BN  = 1023.97 Nm
               TQ_DCC_SLOW_DEC_BN  = 1023.97 Nm
               TQ_DCC_FAST_INC_BN = -1024 Nm
               TQ_DCC_SLOW_INC_BN  = -1024 Nm
               LV_RST_T_TQ_DCC_CS = 1
               LV_ERR_SYM_TQ_DCC_CS = 0
Else(1) if(2)  LV_TQ_DCC_INC_REQ = 1
        Then(2) If(3)            (LV_AT = 0 and LV_VAR_AMT = 0 and LV_VAR_TCT = 0)
                         and
                         LC_ENA_TQ_DCC_CS = 1
                Then(3)  if(4)   LV_IM_CS_PN = 1                  and
                                 (STATE_DCC_CTL = 0 H     or
                                 STATE_DCC_CTL = 2 H)             and
                                 LV_ERR_SYM_TQ_DCC_CS = 0
                         Then(4) TQ_DCC_FAST_INC_BN = TQ_DCC_FAST_INC_BN$_{(n-1)}$ +
                                 C_CRLC_TQ_DCC_CS_DEC * (IP_TQ_DCC_CS –
                                     TQ_DCC_FAST_INC_BN$_{(n-1)}$)
                                 TQ_DCC_SLOW_INC_BN = TQ_DCC_SLOW_INC_BN$_{(n-1)}$
                                 + C_CRLC_TQ_DCC_CS_DEC * (IP_TQ_DCC_CS –
                                     TQ_DCC_SLOW_INC_BN$_{(n-1)}$)
                                 If(4´) LV_RST_T_TQ_DCC_CS = 1
                                 Then(4´)      T_TQ_DCC_CS = 0
                                               LV_RST_T_TQ_DCC_CS = 0
                                 Else(4´)      T_TQ_DCC_CS = T_TQ_DCC_CS + 10 ms
                                     If(4´´) T_TQ_DCC_CS ≥  C_THD_T_TQ_DCC_CS
                                     Then(4´´) LV_ERR_SYM_TQ_DCC_CS = 1
                                     Else(4´´) LV_ERR_SYM_TQ_DCC_CS = 0
                                     Endif(4´´)
                                     Endif(4´)
                         Else(4) TQ_DCC_FAST_INC_BN =
```

```
                                TQ_DCC_FAST_INC_BN$_{(n-1)}$ +
                        C_CRLC_TQ_DCC_CS_INC * (TQ_DCC_FAST_BN$_{(n)}$ –
                                TQ_DCC_FAST_INC_BN$_{(n-1)}$)
                        TQ_DCC_SLOW_INC_BN=TQ_DCC_SLOW_INC_BN$_{(n-1)}$+
                        C_CRLC_TQ_DCC_CS_INC * (TQ_DCC_SLOW_BN$_{(n)}$ –
                                TQ_DCC_SLOW_INC_BN$_{(n-1)}$)
                        LV_RST_T_TQ_DCC_CS = 1
                    Endif(4)
            Else(3) TQ_DCC_FAST_INC_BN  = TQ_DCC_FAST_BN (converted)
                    TQ_DCC_SLOW_INC_BN  = TQ_DCC_SLOW_BN (converted)
                    TQ_DCC_FAST_DEC_BN  = 1023.97 Nm
                    TQ_DCC_SLOW_DEC_BN  = 1023.97 Nm
                    LV_ERR_SYM_TQ_DCC_CS = 0
            Endif(3)
    Then(2)     TQ_DCC_FAST_DEC_BN  = 1023.97 Nm
                TQ_DCC_SLOW_DEC_BN  = 1023.97 Nm
                TQ_DCC_FAST_INC_BN  = TQ_DCC_FAST_BN (converted)
                TQ_DCC_SLOW_INC_BN  = TQ_DCC_SLOW_BN (converted)
                LV_RST_T_TQ_DCC_CS = 1
                LV_ERR_SYM_TQ_DCC_CS = 0
    Endif(2)
Endif(1)
```

LDM Torque CAN selection for intervention

```
If(1)       LV_TQ_WHEEL_LDM_REQ = 1     and
            LV_TQ_WHEEL_LDM_BN_ERR = 0
then(1)     TQ_WHEEL_LDM_INC_DEC_BN = TQ_WHEEL_LDM_BN
Else(1)     TQ_WHEEL_LDM_INC_DEC_BN = -32000 Nm
Endif(1)
```

DKG Torque CAN selection for intervention :

```
If      STATE_TCT_INTV  = NO_INTERVENTION (00H)
   or   LV_TCT_LIH_CAN  = 1
   or   TQ_GS _FAST_BN = -1024 Nm
   or   TQ_GS_SLOW_BN = -1024 Nm
then    TQ_GS _FAST_DEC_BN  = 1023.97 Nm
        TQ_GS _SLOW_DEC_BN  = 1023.97 Nm
        TQ_GS _FAST_INC_BN = -1024 Nm
        TQ_GS _SLOW_INC_BN  = -1024 Nm
elseif  LV_TQ_GS_INC_REQ = 1
        then   TQ_GS_FAST_DEC_BN  = 1023.97 Nm
               TQ_GS_SLOW_DEC_BN  = 1023.97 Nm
               TQ_GS_FAST_INC_BN  = TQ_GS_FAST_BN (converted)
               TQ_GS_SLOW_INC_BN  = TQ_GS_SLOW_BN (converted)
        elseif LV_TQ_GS_DEC_REQ = 1
               then   TQ_GS_FAST_DEC_BN  = TQ_GS_FAST_BN (converted)
                      TQ_GS_SLOW_DEC_BN  = TQ_GS_SLOW_BN (converted)
                      TQ_GS_FAST_INC_BN  = -1024 Nm
                      TQ_GS_SLOW_INC BN  = -1024 Nm
               endif
        endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQ_DCC_CS	6	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_TQ_DCC_CS	6	0...1FE0H	0...8160	1	[rpm]
Intended DCC torque during clutch switch on					
C_THD_T_TQ_DCC_CS	1	0...FFH	0...2.55	0.01	[s]
Threshold for maximum allowed torque intervention concerning clutch switch					
C_CRLC_TQ_DCC_CS_DEC	1	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation constant for decrement TQ_DCC_CS intervention					
C_CRLC_TQ_DCC_CS_INC	1	0...FFH	0...0.99609	3.9063e-3	[-]
Correlation constant for increment TQ_DCC_CS intervention					
LC_ENA_TQ_DCC_CS	1	0...1H	0...1	1	[-]
Logical constant to set the TQ_DCC_CS intervention active					

43.4 11hex – Diagnosis Gateway

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STMIN_ECU	1	0...FFH	0...255	1	[ms]
STMIN for gateway from EGS to tester					

Description:

ECU is in case of 11hex gateway between EGS on PT-CAN and Tester on Lo-CAN (e.g reprogramming via tester). The table below describes the way of routing the named diagnosis messages. Routing by hardware means that the controller is used (no application software involved). Routing by software means that the related message is received and if necessary modified and than sent to designated receiver (e.g. tester).

Messages	ID	Routing by ..	Sent from..	Sent to..
Diagnosis-Messages	6F0-6FF	Hardware	Tester	EGS
Diagnosis-Messages	618	Software	EGS	Tester

Formula section:

Rule for software routing:

```
receive message (buffer byte 0-7)
If      byte 1 of received message == 30hex              (FlowControl)
Then    if      byte 3 of received message <  C_STMIN_ECU  (STMIN)
        Then    byte 3 = C_STMIN_ECU
        Endif
Endif
send message (buffer byte 0-7)
```

43.5 CAN Calibration Protocol (CCP)

The Software contains a CAN calibration protocol in accordance with the Spec

CAN-Calibration Protocol
Draft Version 2.1, 03-june-98

Remark: Online Data acquisition with CCP is also possible with connected ETK, but calibration with these ECUs is only possible via ETK.

43.6 Component Driver for BSD-components

43.6.1 BSD-Drivers

43.6.1.1 Device–detection algorithm

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SENS_BAT_SMT_DET	V/O/S	0...1H	0...1	1	[-]
IBS has been detected					
LF_BSD_CPT_AVL	V/O	0...FFH	0...255	1	[-]
Logical data field displaying the learnt and configured BSD devices					

Input data:

LV_IGK			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_BSD_CPT_DET	1	0...FFH	0...255	1	[-]
Number of required consecutive positive acknowledges to get a component learnt					
C_STATE_BSD_CPT_AVL	1	0...FFH	0...255	1	[-]
Actual used BSD devices (bit coded! 1 = used)					
C_STATE_BSD_CPT_DET	1	0...FFH	0...255	1	[-]
BSD devices, which are set by default (bit coded! 1 = not to be learnt)					

FUNCTION DESCRIPTION:

General information:

There is an algorithm installed to detect the single component devices connected to the ECU via BSD. At every ECU reset the learning algorithm is started and for each possible device it is checked whether the device is available or not. In case the consecutive positive acknowledges of the device reach the calibratabel threshold C_CTR_BSD_CPT_DET the corresponding device has been learnt. Once a device has been detected it is assumed that it is installed for ever and for this specific device the learning algorithm is deactivated. To realise this the flag indicating whether a device is learnt or not is stored in the NVMY. The learnt devices can solely be deleted via tester tool. They can be read via KWP job. Alternator, QLT and electronically controlled water pump are assumed to be there in every case, resulting in a fulltime activation of the corresponding device drivers. In addition each device can be set by a SW constant as to be available (C_STATE_BSD_CPT_AVL) and if the available devices should be leart or not (C_STATE_BSD_CPT_DET) (set as loc. True!)

Description:

The following bit-coding is applied to the calibration constants C_STATE_BSD_CPT_AVL, C_STATE_BSD_CPT_DET and the status byte LF_BSD_CPT_AVL (set as loc. True!)
The logical data field LF_BSD_CPT_AVL displays the configured and learnt BSD components. In case there is a mismatch between C_STATE_BSD_CPT_AVL and LF_BSD_CPT_AVL obviously a problem occurred in the BSD components learning session. The by the constant C_STATE_BSD_CPT_DET configured BSD components are directly set in the logical data field LF_BSD_CPT_AVL. The BSD components which have to be learnt are set after the learning process has been finished. In case the BSD component has been successfully learnt the corresponding bit is set to 1 in the logical data field LF_BSD_CPT_AVL. Otherwise the corresponding device bit stays 0.
.

Bit position	BSD device
0	SENS_BAT_SMT -- Intelligent battery sensor (IBS)
1	-- O3 Sensor
2	-- el. waterpump electronics
3	CWP -- electrical coolant water pump
4	QOIL – oil quality sensor (QLT)
5	-- alternator 2
6	ALTER – alternator
7	-- glow plug control unit

Application conditions:

Initialisation: all variables are initialised out of NVMY

Recurrence: once after LV_IGK = 1

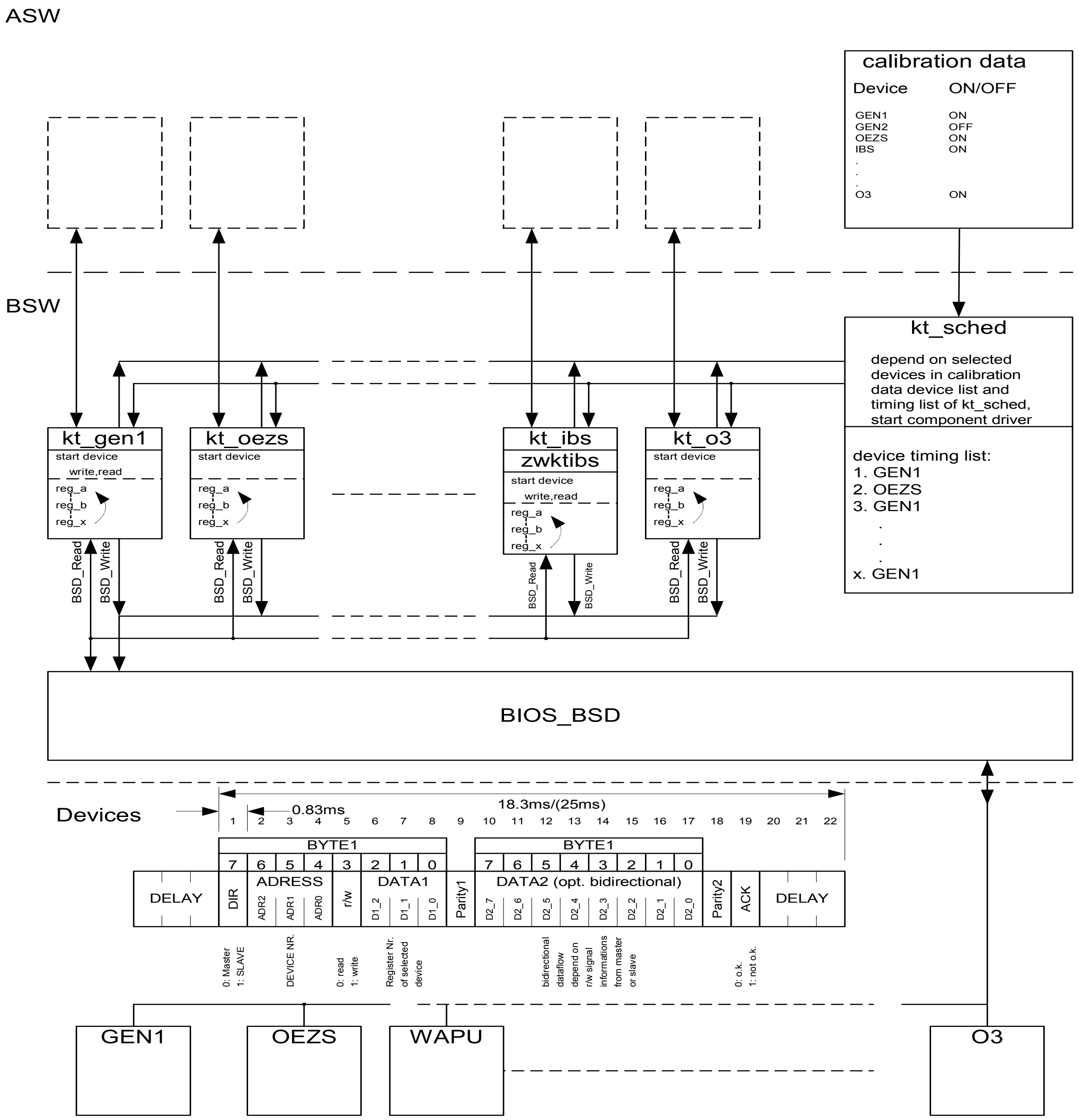
Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

```
If      KT_IBS_VAR = 1 (ASW, BSW Interface)
Then    LV_SENS_BAT_SMT_DET = 1
Else    LV_SENS_BAT_SMT_DET = 0
Endif

If      LV_SENS_BAT_SMT_DET = 0
Then    Device–detection algorithm for IBS is activated
Else    Device–detection algorithm for IBS is deactivated
Endif

If      LV_SENS_BAT_SMT_DET = 1            /*IBS learnt in this or in any former driving cycle*/
Then    Component driver for IBS is activated
Else    Component driver for IBS is deactivated
Endif
```

Signal flow diagram:



43.6.1.2 BSD-Timing coordinaton and BSD timing table

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_BSD_CPT_COM_OK	-	0...FFH	0...255	1	[-]
Number of "ok" communications between ECU and BSD devices per scheduling timing table list					
IDX_BSD_SDL_TBL	-	0...FFH	0...255	1	[-]
Index of the timing scheduler for the BSD devices					
LV_ALTER_COM_ACT	-	0...1H	0...1	1	[-]
Flag indication communication on BSD to alternator disturbed					
LV_CWP_COM_ACT	-	0...1H	0...1	1	[-]
Flag indication communication on BSD to coolant water pump disturbed					
LV_QOIL_COM_ACT	-	0...1H	0...1	1	[-]
Flag indication communication on BSD to oil quality sensor disturbed					
LV_SENS_BAT_SMT_COM_ACT	-	0...1H	0...1	1	[-]
Flag indication communication on BSD to intelligent battery sensor disturbed					
STATE_BSD_COM	-	0...FFH	0...255	1	[-]
Actual status of the communication between ECU and BSD devices					
STATE_BSD_CPT	-	0H 3H 4H 6H	SENS_BAT_SMT CWP QOIL ALTER	1	[-]
Actually communicating BSD device					

Input data:

C_CTR_MAX_ALTER_COM_STOP	C_CTR_MAX_ALTER_COM_STOP	C_CTR_MAX_CWP_COM_STOP	C_CTR_MAX_CWP_COM_STOP
C_CTR_MAX_QOIL_COM_STOP	C_CTR_MAX_QOIL_COM_STOP	C_CTR_MAX_SENS_BAT_SMT_COM_STOP	C_CTR_MAX_SENS_BAT_SMT_COM_STOP
ECU_STATE	LV_ERR_ALTER_COM	LV_ERR_CWP_COM	LV_ERR_QOIL_COM
LV_ERR_SENS_BAT_SMT_COM	STATE_BSD_MSG_COM_STOP		

Import actions:

ACTION_COMS_*SetBsdGlobalCom* (IN <CTR_BSD_CPT_COM_OK>)

ACTION_COMS_*SetBsdSingleCom (*IN <STATE_BSD_CPT>, IN <STATE_BSD_COM>)

Description for actions:

ACTION_COMS_*SetBsdGlobalCom* (<CTR_BSD_CPT_COM_OK>)					
Action which delivers at the end of the BSD timing table the status of the communication of all BSD devices of the list; number of successful communications for one scheduling timing list cycle					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
CTR_BSD_CPT_COM_OK	IN	0...FFH	0...255	1	-
Number of the 'ok' communications per total scheduling timing list cycle of the BSD devices					
ACTION_COMS_*SetBsdSingleCom* (<STATE_BSD_CPT>,<STATE_BSD_COM>)					
Action which delivers at the end of the BSD timing slot the status of the communication of the actual BSD device of the scheduling list					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
STATE_BSD_CPT	IN	0...FFH	0...255	1	-
BSD device, which communicates actually with the ECU via BSD					
STATE_BSD_COM	IN	0...FFH	0...255	1	-
Status of the communication to the BSD devices (faulty /o.k.)					

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_STATE_BSD_SDL_TBL	32	0...FFH	0...255	1	[-]
LDP_IDX_BSD_SDL_TBL	32	0...FFH	0...255	1	[-]
BSD component device scheduling timing table					

FUNCTION DESCRIPTION:

General information:

The BSD timing coordinator handles the scheduling of the device drivers. The 8 possible component devices (bit 0...3) can be configured in the table ID_STATE_BSD_SDL_TBL. In addition the configuration can be configered for 3 different states (Engine running (bit 4), power latch (bit 5), wake up (bit3)). The table configures the scheduling of the BSD devices. The end of the scheduling table is marked by FFH.

For each BSD device the status of the communication is checked. In case of no acknowledge of the device the flags LV_CWP_COM_ACT, LV_ALTER_COM_ACT, LV_QOIL_COM_ACT, LV_SENS_BAT_SMT_COM_ACT are set at the end of the timing slot and the action ACTION_COMS_*SetBsdSingleCom (*IN <STATE_BSD_CPT>, IN <STATE_BSD_COM>) is called.
The action feeds the diagnosis of the individual BSD devices in term of the detection of a communication error.

After the detection of faulty communication in a certain timing slot the next corresponding timing slot is served with a '22 synchronisation pulse phase' followed by a none acknoledge!

Unter certain circumstances the communication to a device can be stopped (LV_MSG_xxx_COM_STOP) although the time slot is still reserved for this device. The staus byte STATE_BSD_MSG_COM_STOP communicates the information via to the BSD timing coordinator to which BSD component a timout is applied. Background: it is assumed that after a disturbed communication to a device has been detected a 'break' in the communication could cure the problem.

Before the BSD device scheduling timing table is started again (IDX_BSD_SDL_TBL =0) the flags LV_xxx_COM_ACT have to be reset to 0. In case no acknowledge is set through the whole BSD timing table, the flags LV_xxx_COM_ACT are still =0 at the end of the timing table, so therefore the global BSD error can be fed with this information. For each positive acknowledge per timing slot the counter CTR_BSD_CPT_COM_OK is incremented by 1. CTR_BSD_CPT_COM_OK equal to 0 indicates a global BSD communication error. At the end of the list IDX_BSD_SDL_TBL(ID_STATE_BSD_SDL_TBL = FFH) the action ACTION_COMS_*SetBsdGlobalCom*(<CTR_BSD_CPT_COM_OK>) is called and the global BSD error can be debounced by the ERRM.

The communication to the devices is monitored by the BSW.

The error symptom detection is made out of basic software information BIOS_BSD_xx.
If a problem is detected the corresponding malfunction of the communication between the ECU and the BSD device is indicated by the corresponding flag LV_xxx_COM_ACT.

With xxx:

ALTER	– alternator
CWP	– electrical coolant water pump
QOIL	– oil quality sensor (QLT)
SENS_BAT_SMT	– intelligent battery sensor (IBS)

Description:

The driver of a certain device is only called if the device has been learnt. Some BSD device drivers are not called (writing only!) if the message to be written has not changed. So if it is not necessary for the device driver to send (e.g. if the values did not change), this is reported to the timing coordinator. The timing coordinator can then pass on this time slot to the next BSD device in the scheduling timing table.

Design of the scheduling timing table of the BSD devices:

ID_STATE_BSD_SDL_TBL

IDX_BSD_SDL_TBL	Bit 7 not used	Bit 6 not used	Bit 5 'wake up' ECU_STATE = 6H	Bit 4 'engine running' ECU_STATE = 0H v	Bit 3 'power latch' ECU_STATE = 4H	Bit 2	Bit 1	Bit 0	
0	0	0	0	1	0	1	1	0	e.g. Alternator ***SetBsdSingleCom***
1	0	0	1	1	0	0	0	0	e.g. Battery sensor ***SetBsdSingleCom***
2	0	0	1	1	1	0	1	1	e.g. Coolant water pump ***SetBsdSingleCom***
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	
17	1	1	1	1	1	1	1	1	FFH – End of table; goto begin of table ***SetBsdGlobalCom***
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	
30	0	0	0	0	0	0	0	0	
31	0	0	0	0	0	0	0	0	

The BSD device is coded in Bit 0...2.

Application conditions:

Initialisation:	0 at WakeUp	
	Recurrence:	20ms (optional 25ms in MS45)
Activation:	WakeUp	
Deactivation:	Sleeping	

Formula section:

Handling of the single /global BSD com error /counter CTR_BSD_CPT_COM_OK:

```
If      IDX_BSD_SDL_TBL < IDX_BSD_SDL_TBL(ID_STATE_BSD_SDL_TBL = FFH)
        //Pointer of the BSD timing coordinator is between the starting point of the scheduling timing table
          and the end of the Scheduling timing table//

Then
   If      LV_MSG_xxx_COM_STOP = 0 (STATE_BSD_MSG_COM_STOP)
   Then  If      ECU receives no positive acknowledge from slave device
         Then    LV_xxx_COM_ACT = 0                    and
                 Send a 22 sychronisation pulse phase
                 in next corresponding time slot followed by
                 LV_xxx_COM_ACT == 0
         Else    LV_xxx_COM_ACT = 1
         Endif
             If      LV_xxx_COM_ACT = 1
                     With: xxx – corresponding to the actual time slot device
             Then    CTR_BSD_CPT_COM_OK ++
             Endif
         // the end of the timing table is not reached-transmit single message error status//
         Call ACTION_COMS_SetBsdSingleCom (IN <STATE_BSD_CPT>, IN <STATE_BSD_COM>)
         IDX_BSD_SDL_TBL++
   Else  stop communication to device xxx in the corresponding time slot
         STATE_BSD_COM = FFH
         Call ACTION_COMS_SetBsdSingleCom (IN <STATE_BSD_CPT>, IN <STATE_BSD_COM>)
         IDX_BSD_SDL_TBL++

   Endif

        //Counter is reset to 0 at the end of the Scheduling timing table after the action has been called//
        // the end of the timing table is reached-transmit global BSD error status//
Else    Call        ACTION_COMS_SetBsdGlobalCom(IN          <CTR_BSD_CPT_COM_ACT>)
        CTR_BSD_CPT_COM_OK = 0          // reset communication counter//*
        IDX_BSD_SDL_TBL = 0             // re-started BSD timing table//*
        LV_CWP_COM_ACT = 0              // Table is re-started ; reset communication flags//*
        LV_ALTER_COM_ACT = 0
        LV_QOIL_COM_ACT = 0
        LV_SENS_BAT_SMT_COM_ACT = 0
Endif
```

Control of the communication after an communication error has been detected:

```
If      C_CTR_MAX_xxx_COM_STOP <> FFFFH             and
        LV_ERR_xxx_COM = 1
Then    stop communication with this device for this DC (LV_IGK 1-->0)
Else    communication with the corresponding device to be continued
```

With:
STATE_BSD_CPT = STATE_BSD_SDL_TBL(bit0...2)
STATE_BSD_COM = LV_xxx_COM_ACT
STATE_BSD_MSG_COM_STOP = LV_MSG_xxx_COM_STOP

STATE_BSD_CPT	STATE_BSD_COM
0H --SENS_BAT_SMT–intelligent battery sensor	1H = LV_SENS_BAT_SMT_COM_ACT=0
3H -- CWP– electrical coolant water pump	8H = LV_CWP_COM_ACT = 0
4H – QOIL – oil quality sensor (QLT)	10H = LV_QOIL_COM_ACT = 0
6H -- ALTER – alternator	40H = LV_ALTER_COM_ACT = 0

STATE_BSD_CPT	STATE_BSD_MSG_COM_STOP
0H --SENS_BAT_SMT–intelligent battery sensor	1H = LV_MSG_SENS_BAT_SMT_COM_STOP=1
3H -- CWP– electrical coolant water pump	8H = LV_MSG_CWP_COM_STOP = 1
4H – QOIL – oil quality sensor (QLT)	10H = LV_MSG_QOIL_COM_STOP = 1
6H -- ALTER – alternator	40H = LV_MSG_ALTER_COM_STOP = 1

43.6.1.3 Component driver

FUNCTION DESCRIPTION:

General information:

Basics that should be implemented in a device driver:

Description:

To be translated:

- Wenn Werte geschrieben werden sollen, soll abgeprüft werden ob der Wert gleich dem zuletzt gesendeten Wert ist. Ist dieses der Fall, kann die Funktion für diesen Buszyklus wieder verlassen werden.
  Diese Funktionalität dient dazu, die Busbelastung niedrig zu halten.
  Der nächste Teilnehmer laut Timing-Liste kann dann diesen Zeitslot nutzen.

  Nach dem Lernprozess wird das angeschlossene Gerät vom BSD-Timing-Koordinator als vorhanden markiert. Jetzt wird die eigentliche BSD-Kommunikation mit dem ausgewähltem Gerät gestartet. Dabei läuft innerhalb des Komponententreibers wiederum ein Schreib-Lese-Koordinator, welcher in seiner Abfolge frei definiert werden kann. Getaktet wird dieser Vorgang durch den Aufruf des BSD-Timing-Koordinators.

Application conditions:

Recurrence:	wird durch BSD-Timingcoordinator generiert
Activation:	WakeUp
Deactivation:	Sleeping

43.6.2 Component Driver for BSD-alternator

43.6.2.1 Identification of the controller type

FUNCTION DESCRIPTION:

This function allows to identify the controller type of the alternator that is installed and to choose the communication protocol via code word. Possible controller types:

General information:

BSD-controller II measures and limits the exciting current. Additional a detection of the chip temperature. It provides information about the actual alternator torque for a torque model. The communication of the BSD-controller type I is different to the BSD-controller type II. In the first phase BSD-controller type I or type II can be installed. At init., register 7 provides information about controller- and chip version and the nominal voltage. Address 6 communicates with type I and type II. Later on a mix of type I and type II assembly is impossible.

Description:

Detection of the controller type (Register 7, Bit 6/7):

Type	ALTER_COD_2
BSD-controller I	00
BSD-controller II	01

Choosing the communication protocol via code word:

Type	LC_ALTER_BSD_PROT_2
BSD-controller I	0
BSD-controller II	1

Combinations and possible BSD protocol:

ALTER_COD_2	LC_ALTER_BSD_PROT_2	LV_ALTER_BSD_PROT_2
00	0	0
00	1	0
01	0	0
01	1	1

Conditions to be choose by software switch:

sw-switch / codeword	condition	meaning	BMW naming
LC_ALTER_BSD_PROT_2	LV_ALTER_BSD_PTOT_2	BSD-prot 1 or 2	CW_GENPROT

43.6.2.2 BSD controller I

Id. for address 0x06 (BSD component driver controller typ I)

Part: electr. alternator, electronic: address: 6

Remark: address 6 must communicate with type I and type II (depending of the detection what type is installed).

Specification only for BSD-controller type I !!

Id	Parameter	LS Bit	Bit Anz.	w(rite) r(ead)	unit	function
0	V_ALTER_SP	0	6	r/w	V	Alternator voltage / without control XX00 0000 = 10,6V limited XX11 0110 = 16,0V XX11 1111 = 16,0V
	T_LOAD_RESP_ALTER_SP	6	2	r/w	s	Load Response (LR) 00XX XXXX = 0s 01XX XXXX = 3s 10XX XXXX = 6s 11XX XXXX = 9s
1	--	--	--		--	not used
2	POW_REL_ALTER	0	5	-/r	%	DF-monitor XXX0 0000 = 0% capacity XXX1 1111 = 96,875% capacity
	LV_ALTER_ERR_EL	5	1	-/r	-	electrical alternator error XX0X XXXX = no error XX1X XXXX = error present
	LV_ALTER_ERR_MEC	6	1	-/r		mec. Alternator error X0XX XXXX = no error X1XX XXXX = error present
	LV_ALTER_ERR_TEMP	7	1	-/r	-	High temperature protection 0XXX XXXX = no error 1XXX XXXX = error present
3	--	--	--		--	not used
4	--	--	--		--	not used
5	--	--	--		--	not used
6	ALTER_COD_0	5	3	-/r	-	manufacturer ID 000X XXXX = Bosch 001X XXXX = Valeo
	ALTER_COD_1	0	5	-/r	-	alternator type coding XXX0 0000
7	ALTER_COD_2	6	2	-/r	-	BSD-controller version 00XX XXXX = BSD1 01XX XXXX = BSD2
	V_ALTER_NOM	4	2	-/r	V	nominal voltage alternator controller XX01 XXXX = 14V
	ALTER_COD_3	0	4	-/r	-	Chip-version BSD controller (Index) XXXX 0000 = 0 XXXX 1111 = 15

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ALTER_COD_0	O/V	0...7H	0...7	1	[-]
manufacturer identifier					
ALTER_COD_1	O/V	0...1FH	0...31	1	[-]
type identifier alternator					
ALTER_COD_2	O/V	0...3H	0...3	1	[-]
controller version alternator					
ALTER_COD_3	O/V	0...FH	0...15	1	[-]
chip version alternator					
CUR_ALTER_EXCT	O/V	0...3EH	0...7.75	0.125	[A]
actual value exciting current alternator					
CUR_ALTER_EXCT_LIM	O/V	0...1FH	0...7.75	0.25	[A]
actual value exciting current limitation alternator					
LV_ALTER_BSD_PROT_2	O/V	0...1H	0...1	1	[-]
condition for BSD protocol controller type 2					
LV_ALTER_ERR_EL	O/V	0...1H	0...1	1	[-]
electrical fault alternator					
LV_ALTER_ERR_IF	O/V	0...1H	0...1	1	[-]
Alternator interface failure.					
LV_ALTER_ERR_MEC	O/V	0...1H	0...1	1	[-]
mechanical fault alternator					
LV_ALTER_ERR_TEMP	O/V	0...1H	0...1	1	[-]
(excess) temperature fault alternator					
LV_ALTER_EVE_RD	O/V	0...1H	0...1	1	[-]
condition: eo-read					
LV_ALTER_EVE_WR	O/V	0...1H	0...1	1	[-]
condition: eo-write					
LV_ALTER_IF_ACT	O/V	0...1H	0...1	1	[-]
condition alternator interface activ					
LV_LOAD_RESP_ALTER_CND_1	O/V	0...1H	0...1	1	[-]
alternator load response condition for register 1 activ					
LV_LOAD_RESP_ALTER_THD_ACT	O/V	0...1H	0...1	1	[-]
shut of the load response threshold for alternator (copy of SP)					
POW_REL_ALTER	O/V	0...1FH	0...96.875	3.125	[%]
Exitation of the alternator					
POW_REL_ALTER_L_RES	O/V	0H 1H 2H 3H	DF<50% 50%<DF<75% 75%<DF<100% 100%	1	[-]
DF rough value (occupancy factor) from alternator					
T_FIL_CUR_EXCT_ALTER	O/V	0...FFH	0...2.55	0.01	[s]
Calibratable filter time constant for exciting current of the alternator (Powermanagement)					
T_LOAD_RESP_ALTER	O/V	0...3H	0...15	0	[s]
actual value from alternator load response time					
TEMP_ALTER_MC	O/V	0...1FH	-32...216	8	[°C]
chip temperature alternator					
V_ALTER	O/V	0...3FH	10.6...16.9	0.1	[V]
received voltage alternator					
V_ALTER_NOM	O/V	0H 1H	14V 42V	1	[-]
Nominal voltage alternator / bord net version 14/42V					

Input data:

LV_LOAD_RESP_ALTER_THD_ACT_SP	POW_REL_ALTER_CLC	VB	

Formula section:

```
If      LC_ALTER_BSD_PROT_2  = 1     And
        ALTER_COD_2          = 1
Then    LV_ALTER_BSD_PROT_2  = 1
Else    LV_ALTER_BSD_PROT_2  = 0
Endif

IF      (BSD status is ok, BSD device send an answer to alternator)
Then    LV_ALTER_ERR_IF = 0
Else    LV_ALTER_ERR_IF = 1 (communication BSD device alternator was incorrect)
Endif
If      LV_ALTER_EVE_WR              Or
        LV_ALTER_EVE_RD      = 1
Then    T_FIL_CUR_EXCT_ALTER = C_T_FIL_CUR_EXCT_ALTER * 2
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_FIL_CUR_EXCT_ALTER	1	0...FFH	0...2.55	0.01	[s]
calibratible filter time for exciting current (alternator powermanagement)					
LC_ALTER_BSD_PROT_2	1	0...1H	0...1	1	[-]
logical constant: software swich to choose BSD protocol					

FUNCTION DESCRIPTION:

General information:

V_ALTER_SP will be limited to the upper threshold 16.0 V. After that it will be provided to the BSW for the BSD interface.
T_LOAD_RESP_ALTER will be converted to 3 s resolution, limited between 0 and 9 seconds and provided to the BSW for the BSD interface as well.
ALTER_COD_1 delivers the type of alternator
ALTER_COD_0 delivers the manufactory of alternator
This both information is necessary to identify the alternator. Depending on this information the power management is able to switch different calibrations.

Application conditions:

Activation: at every engine operating state

Deactivation: -

Initialization: all outputs 0 (LV_GEN_xx = FALSE)

ALTER_COD_1 = 1F H /*initalisation to max value*/
ALTER_COD_0 = 7 H /*initalisation to max value*/

Update-Rate: 50 ms

Formula section:

if no valid message was received during a timeout : T_LOAD_RESP_ALTER = 3 s
V_ALTER = 14.3 V

else : POW_REL_ALTER_CLC, T_LOAD_RESP_ALTER and V_ALTER are received via BSD interface.

The failure-bits LV_GEN_xx are received via BSD interface.

43.6.2.3 BSD controller II

Id. for address 0x06 (BSD component driver controller typ II)
Part: electr. alternator, electronic: address:6
Remark: Specification only for BSD-controller type II !!

Id	Parameter	LS Bit	Bit Anz.	w(rite) r(ead)	unit	function
0	V_ALTER_SP	0	6	r/w	V	Alternator voltage / without control XX00 0000 = 10,6V limited XX11 0110 = 16,0V XX11 1111 = 16,0V
1	LV_LOAD_RESP_ALTER_THD_ACT_SP	0	1	r/w	-	Shut of speed threshold (s.o.s.t.) for LR-function. at 0 more than approx. 3000 alternator revolutions 0 = s.o.s.t. akt. (def.) 1 = s.o.s.t. deactivated
	LV_LOAD_RESP_ALTER_CND_1	4	1	r/w	-	condition LR register 1 aktiv 0 = LR in reg. 0 act. (def.) 1 = LR in reg. 1 activ
	T_LOAD_RESP_ALTER_SP	5	3	r/w	s	LR- value 000X XXXX = 0s 001X XXXX = 0,2s 010X XXXX = 0,4s 011X XXXX = 3s 100X XXXX = 6s 101X XXXX = 9s 110X XXXX = 12s 111X XXXX = 15s
2	POW_REL_ALTER	0	5	r/-	%	DF-monitor XXX0 0000 = 0% capacity XXX1 1111 = 96,875% capacity
	LV_ALTER_ERR_EL	5	1	r/-	-	electrical alternator error XX0X XXXX = no error XX1X XXXX = error present
	LV_ALTER_ERR_MEC	6	1	r/-		mec. Alternator error X0XX XXXX = no error X1XX XXXX = error present
	LV_ALTER_ERR_TEMP	7	1	r/-	-	High temperature protection 0XXX XXXX = no error 1XXX XXXX = error present
3	CUR_ALTER_EXCT	0	6	r/-	A	read exciting current step 125mA XX00 0000 = 0A XX11 1110 = 7,75A XX11 1111 = 7,75A
	CUR_ALTER_EXCT_LIM_SP	1	5	-/w	A	exciting current limit. XX00 0000 = no limitation up to XX00 0010 = lim. to act. value up to XX00 0100 = lim. to 2A up to XX01 0000 = lim. to 2A up to XX01 0010 = lim. to 2,25A up to XX11 1110 = lim. to 7,75A

	POW_REL_ALTER_L_RES	6	2	r/-	%	DF-rough value 00XX XXXX = DF< 50% 01XX XXXX = 50% <DF< 75% 10XX XXXX = 75% <DF< 100% 11XX XXXX = 100%
4	--	--	--		--	--
5	TEMP_ALTER_MC	1	5	r/-	°C	Chiptemperature XX00 000(0) = < -32°C Up to XX10 111(0) = ≤ 152°C XX11 001(0) = > 152°C (Bit 0=1/0 depending on Gen type. no info. about temp.)
6	ALTER_COD_0	5	3	r/-	-	manufacturer ID 000X XXXX = Bosch 001X XXXX = Valeo
	ALTER_COD_1	0	5	r/-	-	alternator type coding XXX0 0000
7	ALTER_COD_2	6	2	r/-	-	BSD-controller version 00XX XXXX = BSD1 01XX XXXX = BSD2
	V_ALTER_NOM	4	2	r/-	V	nominal voltage alternator controller (bord net version 14/42V) XX00 XXXX = 14V XX01 XXXX = 42V
	ALTER_COD_3	0	4	r/-	-	Chip-version BSD controller (Index) XXXX 0000 = 0 XXXX 1111 = 15

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_ALTER_EVE_WR	V	0...FFH	0...255	1	[-]
eo-write counter					
CTR_T_RD_PER_ALTER_TBL_2	-	0...FFH	0...10.2	0.04	[s]
Time for reading register 2					
CTR_T_RD_PER_ALTER_TBL_5	-	0...FFH	0...10.2	0.04	[s]
Time for reading register 5					
CTR_T_WR_PER_ALTER_TBL_0	-	0...FFH	0...10.2	0.04	[s]
Time for write register 0					
CTR_T_WR_PER_ALTER_TBL_1	-	0...FFH	0...10.2	0.04	[s]
Time for write register 1					
LF_ALTER_EVE_WR	V	0...FFH	0...255	1	[-]
Logical field for eo-write operation					
LV_EXC_CUR_ALTER_EXCT_LIM_SP	-	0...1H	0...1	1	[-]
Indicates "B_chexc" condition					
LV_EXC_T_LOAD_RESP_ALTER_SP_0	-	0...1H	0...1	1	[-]
Indicates "B_chtlrgen" condition					
LV_EXC_V_ALTER_SP	-	0...1H	0...1	1	[-]
Indicates "B_chugen" condition					
LV_REQ_GEN_RD_REG2	-	0...1H	0...1	1	[-]
Request read register 2					
LV_REQ_GEN_RD_REG5	-	0...1H	0...1	1	[-]
Request read register 5					
LV_REQ_GEN_WR_REG0	-	0...1H	0...1	1	[-]
Request write register 0					
LV_REQ_GEN_WR_REG1	-	0...1H	0...1	1	[-]
Request write register 1					
LV_REQ_GEN_WR_REG3	-	0...1H	0...1	1	[-]
Request write register 3					
LV_STEP_BSD_ALTER	-	0...1H	0...1	1	[-]
Request action on "ablaufwr"					
NR_ALTER_EVE_WR	V	0...FFH	0...255	1	[-]
Number of eo-write event					
STATE_ALTER_VALUE_WR	V	0...FFH	0...255	1	[-]
State for selection of eo-write value					

I

Input data:

C_T_WR_PER_ALTER_TBL_0	C_T_WR_PER_ALTER_TBL_1	CUR_ALTER_EXCT_LIM	T_LOAD_RESP_ALTER
V_ALTER			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_NR_ALTER_EVE_WR	32	0...7H	0...7	1	-
LDP_IDX_LF_ALTER_EVE_WR	32	0...7FH	0...127	1	-
Map for selection of eo-write event					
ID_STATE_ALTER_VALUE_WR	8	0...3H	0...3	1	-
LDP_IDX_NR_ALTER_EVE_WR	8	0...7H	0...7	1	-
LDP_IDX_CTR_ALTER_EVE_WR	3	0...2H	0...2	1	-
Map for selection of eo-write value					
C_T_WR_PER_ALTER_TBL_0	1	0...FFH	0...10.2	0.04	[s]
calibratible time period for write register 0 (alternator powermanagement)					
C_T_WR_PER_ALTER_TBL_1	1	0...FFH	0...10.2	0.04	[s]
calibratible time period for write register 0 (alternator powermanagement)					
C_T_RD_PER_ALTER_TBL_2	1	0...FFH	0...10.2	0.04	[s]
calibratible time period for reading register 2 (alternator powermanagement)					
C_T_RD_PER_ALTER_TBL_5	1	0...FFH	0...10.2	0.04	[s]
calibratible time period for reading register 5 (alternator powermanagement)					

FUNCTION DESCRIPTION:

General information:

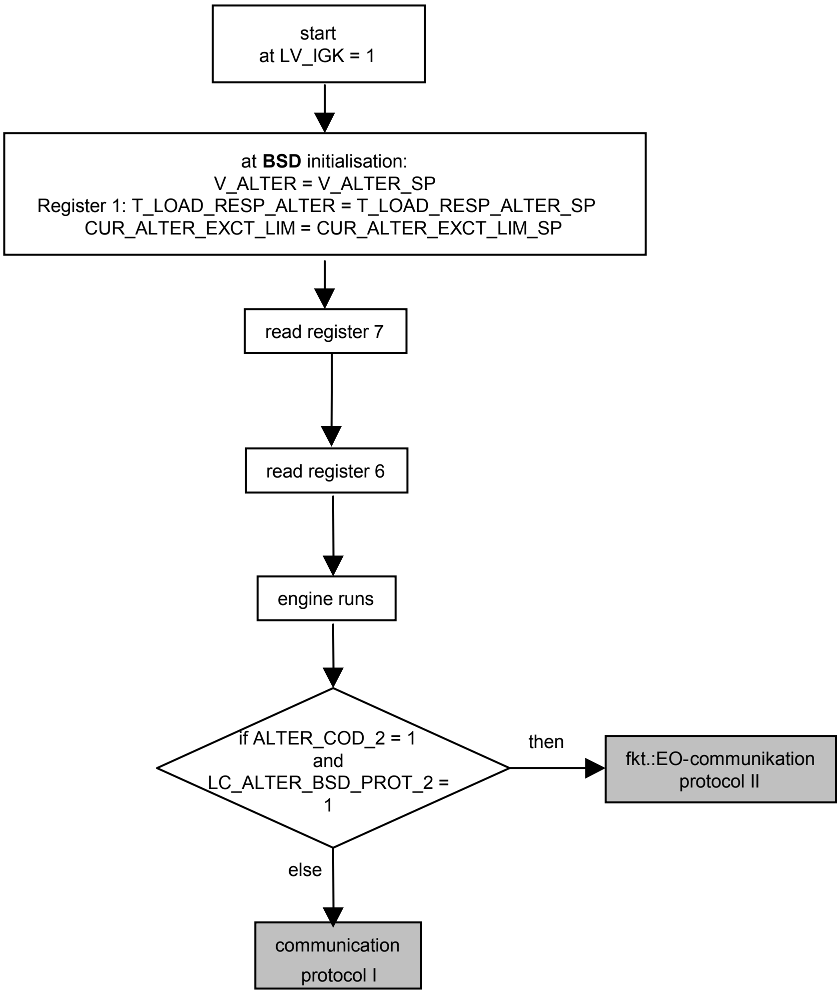
LF_ALTER_EVE_WR is a status flag, which is generated by comparison of the variables CUR_ALTER_EXCT_LIM, T_LOAD_RESP_ALTER, V_ALTER and the corresponding setpoints.
NR_ALTER_EVE_WR is the output of the map **ID_NR_ALTER_EVE_WR** and describes the number of the eo-write event. In case of equality of LF_ALTER_EVE_WR and the datapoint of ID_NR_ALTER_EVE_WR one of 8 possible write events is selected, otherwise the output of ID_NR_ALTER_EVE_WR is 0.
CTR_ALTER_EVE_WR is the counter for the eo-write event process.
STATE_ALTER_VALUE_WR is the output of the map **ID_STATE_ALTER_VALUE_WR** and selects dependant on CTR_ALTER_EVE_WR and NR_ALTER_EVE_WR the next value to write on the bus.

43.6.2.3.1 Additional information for the BSD controller II protocol.

Writing only after a value has changed (after events) therefore comparison of value and reference value (for example Tlrgen -> Tlrfgen).

register 1:
Tlrgen (LR-value) should be round down to the next value of tn. For more details see 1.2.3.5 table writing reg 1. When writing the LR-value then the bit B_lrreg1ak = 1.
Writing of the set points to the alternator register 0 only in a defined time schedule (depending on C_T_WR_PER_ALTER_TBL_0)
register 2:
Reading of the values only in a defined time schedule (depending on C_T_RD_PER_ALTER_TBL_2).
register 3:
Read of register 3 “as often as possible” without eo-communication (defined in the flow chart 1.2.3.2). Minimum every second time slot = read register 3. At successfully communication the variable lerrfgrenz is written.
register 5:
Reading of the values only in a defined time schedule (depending on C_T_RD_PER_ALTER_TBL_5)
register 6:
Reading once at start communication BSD
register 7:
Reading once at start communication BSD

General flowchart KT_Gen BSD controller II



43.6.2.3.2 Flow chart for function EO-communication BSD controller II

Superior function KT_Sched (time scheduler)

only for example, content of time slots not describe the software !!!

20 ms	20 ms	20 ms	20 ms	20 ms	20 ms	20 ms	20 ms	20 ms	20 ms	20 ms
alter- nator	battery sensor	electr. cwp	QLT	alter- nator	battery sensor	electr. cwp	QLT	alter- nator	battery sensor	....

By a time scheduler every 20 ms a new communication is triggered. During this time one register can be in progress (sending a protocol and call a driver). If KT_Gen is triggered, then the progress shown below is started.

Subordinated function EO communication KT_GEN

Application conditions:

Initialisation: only if LV_ALTER_BSD_PROT_2 = 1

Recurrence: 40 ms

Activation: at every engine state

Deactivation: -

Following flow charts are only a functional description of EO-communication BSD controller II. Due to the fact, that it is not possible to do action on BSD immediately for the requested device (for e.g. Generator), the requested jobs has to be stored in circular buffer (FIFO).

In case of GEN slot is carry out the jobs in the buffer will be executed according the FIFO principle.

In case of "V_ALTER_SP (U_gen)" goes to 10,6V, immediately write of REG 0 and delete all other Jobs in the queue.

In case of "wrablauf > 0", immediately execute function "ablaufwr" and delete all other Jobs in the queue. New entries can be done after the last requested job from "ablaufwr" was executed.

Was a deleting of the queue performed and there are still jobs for reading Reg. 2 or Reg. 5 in the buffer, the jobs are lost.
Due to the fact, there is no positive response for reading Reg.2 or Reg.5, therefore set the timers for Reg.2 and Reg.5 again.

The recurrence of calculation "Generator BSD Controller II function" (40ms) could be higher than the BSD-Driver execute the Generator components (best case 20ms, worst case more than 40ms).
To avoid multiple entries for the same job in the buffer (because there was no update yet), the job entry in the buffer is locked till this job was performed by the BSD Driver.

The locking of the Register will be indicated with following flags:

- LV_REQ_GEN_WR_REG0
- LV_REQ_GEN_WR_REG 1
- LV_REQ_GEN_WR_REG 3
- LV_REQ_GEN_RD_REG 2
- LV_REQ_GEN_RD_REG 5
- LV_STEP_BSD_ALTER

the corresponding flags will be set to "0" after job has been perfomed.

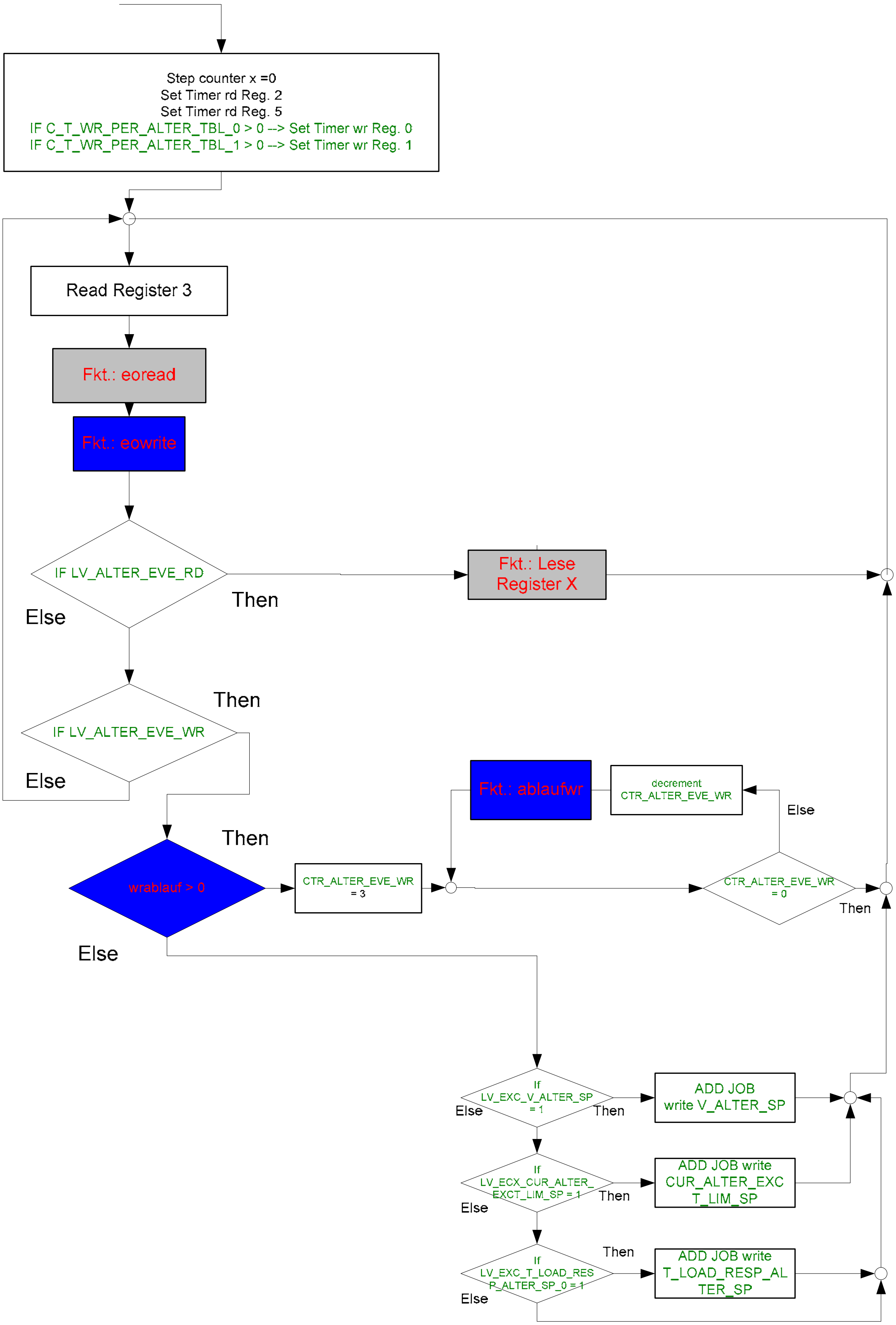
Following locks in the buffer could be possible:

wrablauf = 0
Lock requested writing Reg. 0, 1, 3 till this job is done.
wrablauf > 0
Immediately execute function "ablaufwr", writing all jobs that are defined in ID_STATE_ALTER_VALUE_WR and lock new entry in the queue.

Lock reading Reg. 2 or Reg. 5 till reading was successfully executed.

In case of C_T_WR_PER_ALTER_TBL_0 / 1 are set to "0" not time triggert write to REG 0 / 1 will be executed, only event triggert action can be perfomed.

Fkt.: EO-Kommunikation



43.6.2.3.3 Flow chart for the function eowrite

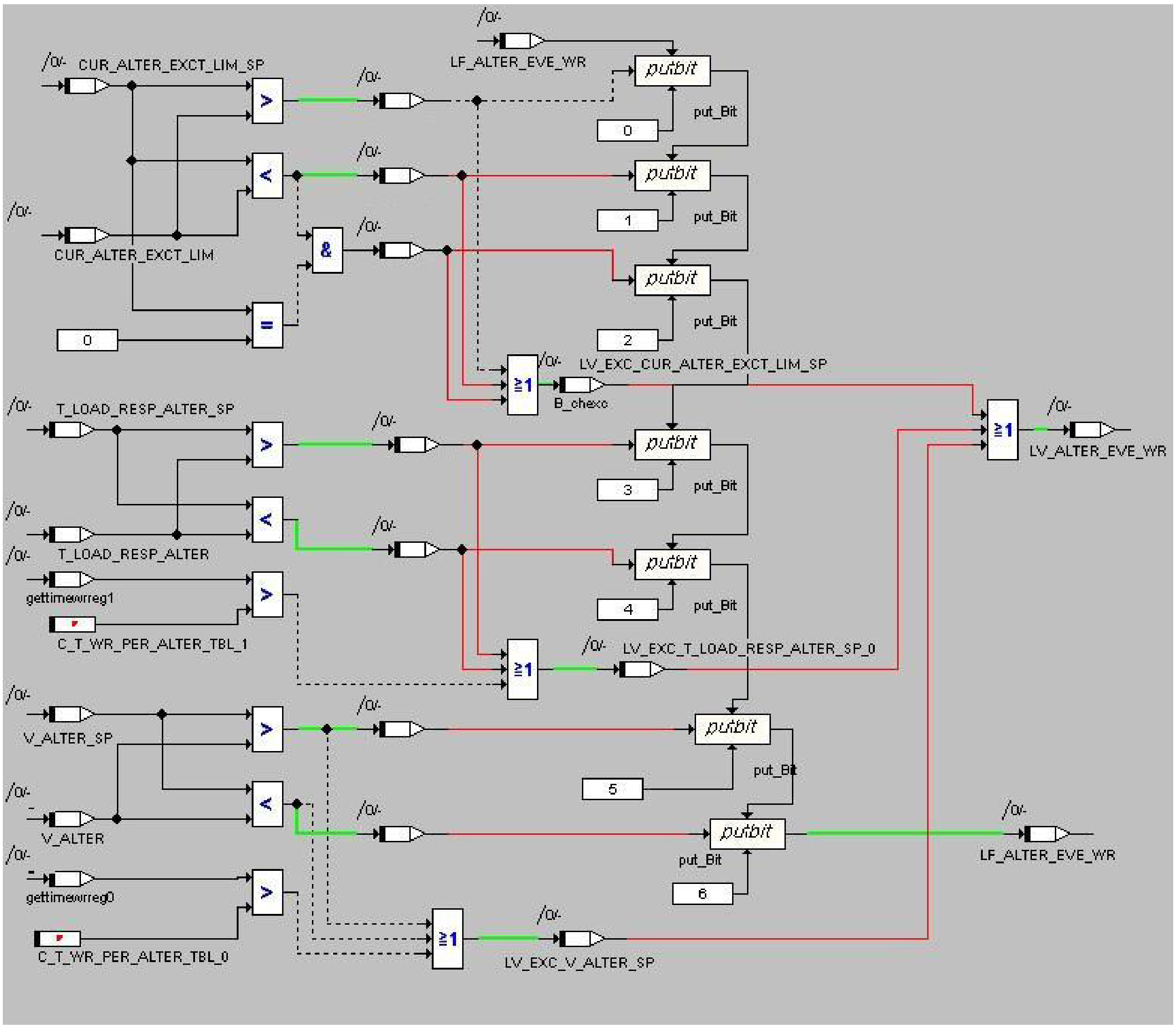
Application conditions:

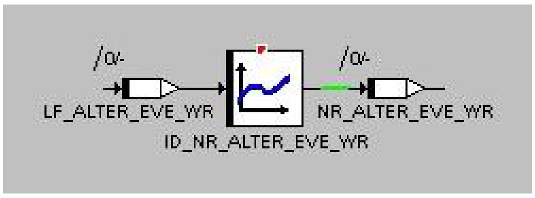
Initialisation: only if LV_ALTER_BSD_PROT_2 = 1

Recurrence: 40 ms

Activation: at every engine state

Deactivation: -





43.6.2.3.4 Flow chart for the function eoread

Application conditions:

Initialisation: only if LV_ALTER_BSD_PROT_2 = 1

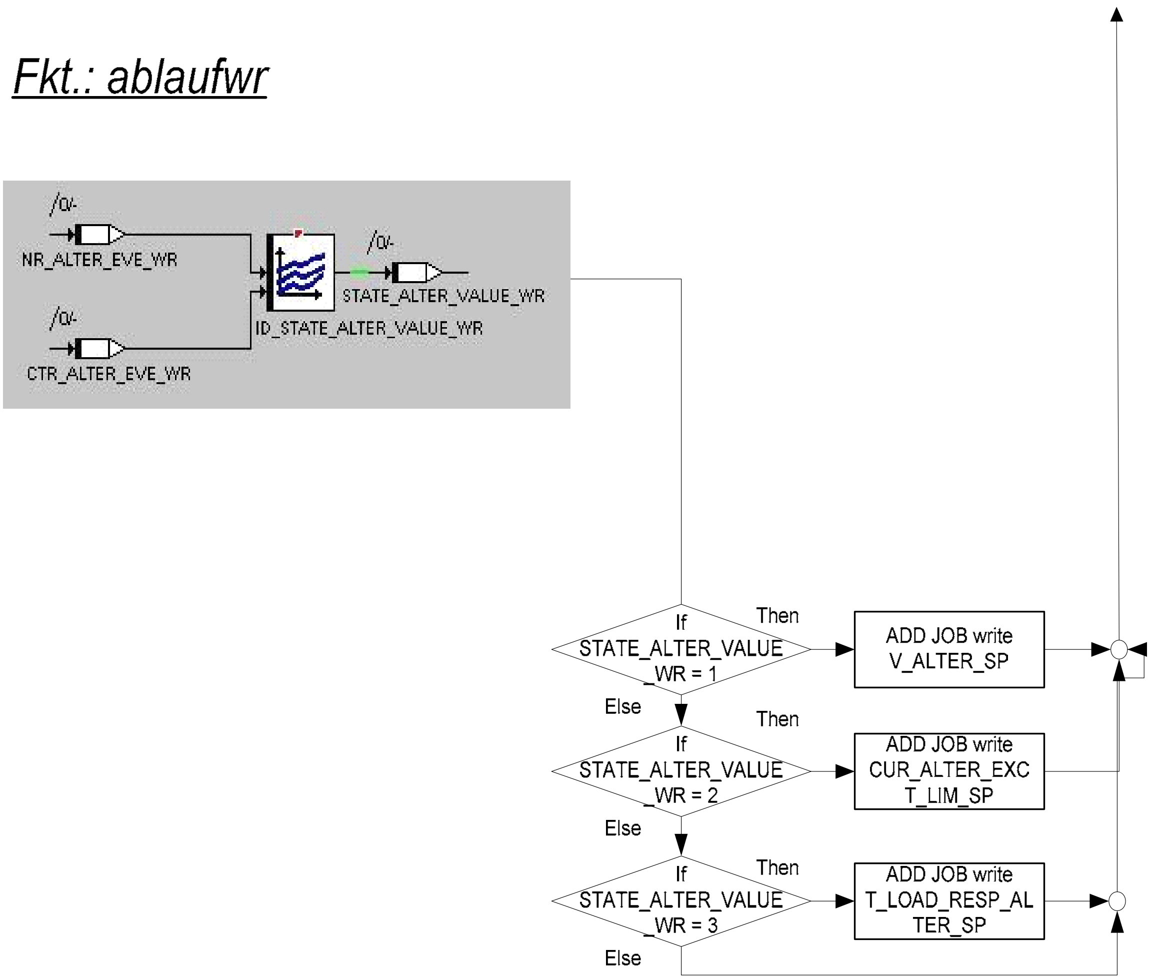
Recurrence: 40 ms

Activation: at every engine state

Deactivation: -

43.6.2.3.5 Flow charts for the functions ablaufwr / LR-write / read reg.x / calculation of the filter time

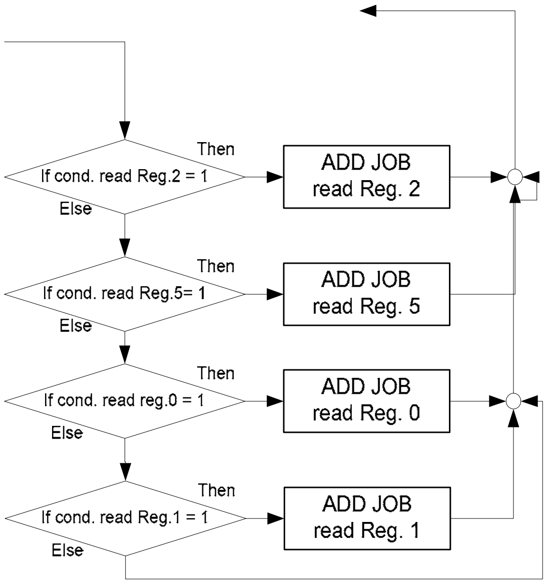
Fkt.: ablaufwr



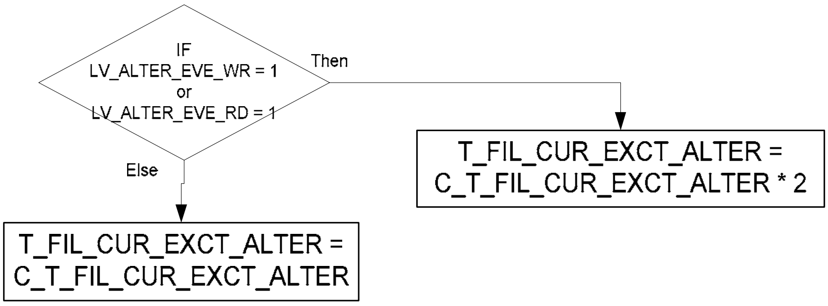
Fkt.: LR-write

The translation table for the fkt.: LR-write are shown in the AGGR-Spec PWSL!!

Fkt.: read register X



Fkt.: calculation of the filter time



43.6.2.4 State machine Component Driver for BSD-alternator Type I

FUNCTION DESCRIPTION:

General information:

Der Komponententreiber wird durch einen Zustandsautomaten der im Aufrufraster des BSD-Timing-Koordinator läuft dargestellt. Wie der **Error! Reference source not found.** zu entnehmen ist besteht der KT aus mehreren Zuständen die im folgenden näher beschrieben werden.

Signal flow diagram:

Abbildung 43.6-1: Statemachine KT-Treiber Generator

Description:

A) Allgemeiner Zustand:

In jedem Zustand des KT werden folgende Fehler geprüft:

Überschreitet der ***timeout counter for valid message*** die Schwelle **KT_GEN_VALID_MESSAGE_TIMEOUT** so wird die Größe **V_ALTER** auf den Applikationswert **UGENTYP** (13.8V entspricht den Notlaufwert des Generators) bzw. die Größe **T_LOAD_RESP_ALTER** auf den Applikatonswert **TLRDEFAULT** (0x1E entspricht den Notlaufwert des Generators) gesetzt ansonsten wird der ***timeout counter for valid message*** inkrementiert (bei gültigen Daten wird dieser durch die Call_Back-Function zurückgesetzt).

Anschließend wird geprüft ob das Generatorinterface aktiv ist d.h. Bedingung ***B_genactiv*** = True, wenn ja wird der ***timeout send counter*** und der ***timeout read counter*** inkrementiert (bei gültigen Daten wird dieser durch die jeweilige Call_Back-Function zurückgesetzt).

Desweiteren wird im jedem Zustand geprüft ob der ***timeout send counter*** die Schwelle **KT_GEN_SEND_TIMEOUT** bzw. der ***timeout read counter*** die Schwelle **KT_GEN_SEND_TIMEOUT** überschritten hat. Wurde einer dieser Schwellen überschritten geht der KT in den Zustand *kt_GenStatus = KT_GEN_ST_INI* über .

B) Zustand KT_GEN_ST_INI:

Zu Beginn wird der KT initialisiert d.h.
V_ALTER = 0;
T_LOAD_RESP_ALTER = 0;
LV_GEN_ERR_EL = FALSE;
LV_GEN_ERR_TEMP = FALSE;
LV_GEN_ERR_MEC = FALSE;
LV_GEN_IF_OFF = FALSE;
POW_REL_ALTER_CLC = 0;

und andere diverse Größen wie Fehlerzähler usw..

Nach der Initialisierung befindet sich der KT im Zustand *kt_GenStatus = KT_GEN_ST_INI*. In diesem Zustand wird zyklisch ein ***initialization timeout counter*** hochgezählt und dem Hardwaretreiber (HWT) Initialisierungsdaten übergeben und anschließend der HWT gestartet. Als Initialisierungsdaten werden dem HWT folgende Größen übergeben:

period length
low_time Sync pulse
low_time of sended LOW Bit
low_time of sended HIGH Bit
pointer to callback function for read register when result is ready
pointer to callback function for send register when quit from peripherie is OK
pointer to error handler

Wie oben zu sehen ist, werden dem HWT unter anderem zwei Pointer für die Call-Back-Funktion übergeben genaueres hierzu folgt etwas später.

Nun wird der ***initialization timeout counter*** inkrementiert. Anschließend wird geprüft ob der ***initialization timeout counter*** die Initialisierungsschwelle **KT_GEN_INI_TIMEOUT** überschritten hat, wenn ja geht der KT in dem Zustand *kt_GenStatus = KT_GEN_ST_IDLE* über und der ***initialization timeout counter*** wird zurückgesetzt.

Nachdem der HWT gestartet worden ist, geht der KT in dem Zustand *kt_GenStatus = KT_GEN_ST_WAIT* über.

C) Zustand KT_GEN_ST_WAIT:

In diesem Zustand wird gewartet bis der HWT den Status „work“ mitteilt. Hier muß durch den BSD-Timing-Koordinator angezeigt werden ob sich der Treiber noch im Lernmodus befindet . Ist dieser Lernprozess abgeschlossen wird der ***initialization timeout counter*** , ***timeout counter send message*** und der ***timeout counter read message*** zurückgesetzt und die Bedingung ***B_genactiv*** auf True gesetzt (generator interface is active). Anschließend geht der KT in den Status über welcher vom internen Schreib-Lese-Koordinator vorgegeben wird.

D) Zustand KT_GEN_ST_IDLE:

Befindet sich der KT im Zustand *kt_GenStatus = KT_GEN_ST_IDLE* wird dem HWT ein „ausschalten“ des generator interface mitgeteilt, d.h. die Kommunikation auf der BSD-Schnittstelle bzgl. Generator wird eingestellt. Anschließend wird gewartet bis der HWT ein generator interface off mitteilt.

Nachdem vom HWT ein generator interface off mitgeteilt wurde wird die Bedingung ***B_genactiv*** auf False gesetzt, die Bedingung ***LV_GEN_IF_OFF*** auf True gesetzt und in diesem Zustand verharrt. Die Bedingung ***LV_GEN_IF_OFF*** wird nur oberhalb einer Batterie Spannungsschwelle VB >=C_VB_MIN_POW_DIAG aktiviert.

E) Zustand KT_GEN_ST_WRITE_REG0:

Die Generatorsollspannung ***V_ALTER_SP*** zwischen den beiden Applikationswerten **UGMN** und **UGMX** begrenzt. Ist ***V_ALTER_SP*** kleiner gleich **UGMN** wird als Generatorsollspannung der Applikationswert **OUINIGEN** verwendet.

Nach einer Umquantisierung des Sollwerts Load- Response- Zeit Generator ***T_LOAD_RESP_ALTER_SP*** wird diese zwischen den beiden Applikationswerten **TLRMN** und **TLRMX** begrenzt.

Anschließend wird dem HWT über einen Funktionsaufruf die Generatorsollspannung ***V_ALTER_SP*** und der Sollwert der Load-Response-Zeit ***T_LOAD_RESP_ALTER_SP*** mitgeteilt. Wurden diese Werte vom HWT über die BSD-Schnittstelle gesendet, wird die Call-Back-Funktion *KT_gen_sendErgHandler* über einen Pointer aufgerufen.

Nachdem die Werte dem HWT übergeben worden sind, geht der KT in den nächsten angeforderten Zustand *kt_GenStatus* über.

F) Zustand KT_GEN_ST_READ_REG0:

Es werden HWT Daten von der BSD-Schnittstelle bzw. vom Generator durch einen Funktionsaufruf des HWT mit gültiger Generatoradressse angefordert.

Liegen Daten vom BSD-Generator dem HWT vor, so ruft der HWT durch einen Pointer die Call-Back-Funktion *KT_gen_readErgHandler* auf.
Nachdem die Werte durch den HWT übernommen worden sind, geht der KT in den nächsten angeforderten Zustand *kt_GenStatus* über.

G) Zustand KT_GEN_ST_READ_REG2:

Es werden HWT Daten von der BSD-Schnittstelle bzw. vom Generator durch einen Funktionsaufruf des HWT mit gültiger Generatoradressse angefordert.

Liegen Daten vom BSD-Generator dem HWT vor, so ruft der HWT durch einen Pointer die Call-Back-Funktion *KT_gen_readErgHandler* auf.

Nachdem die Werte durch den HWT übernommen worden sind, geht der KT in den nächsten angeforderten Zustand *kt_GenStatus* über.

H) Call-Back-Funktion KT_gen_sendErgHandler:

Wie bereits im Kapitel erwähnt, wird bei einer erfolgreichen Übertragung der Generatorsollspannung ***V_ALTER_SP*** bzw. des Sollwerts Load- Response- Zeit ***T_LOAD_RESP_ALTER_SP*** durch den HWT die Call-Back-Funktion *KT_gen_sendErgHandler* über einen Pointer durch den HWT aufgerufen.

In dieser Call-Back-Funktion wird der ***timeout counter for valid message*** sowie der ***timeout send counter*** zurückgesetzt.

I) Call-Back-Funktion KT_gen_readErgHandler:

Liegen Daten vom BSD-Generator dem HWT vor, so ruft der HWT durch einen Pointer die Call-Back-Funktion *KT_gen_readErgHandler* and the flag LV_ALTER_ERR_IF is reset.

In dieser Call-Back-Funktion wird zunächst der ***timeout counter for valid message*** sowie der ***timeout read counter*** zurückgesetzt. Anschließend werden die vom HWT übergebenen Daten ausgewertet bzw. nach einer entsprechenden Umquantisierung abgespeichert.

J) Fehler-Handler-Funktion KT_gen_errHandler:
Erkennt der HWT Fehler auf der BSD-Schnitstelle, geht der KT in den Zustand *kt_GenStatus = KT_GEN_ST_INI* über and the flag LV_ALTER_ERR_IF is set.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
S_RDGENTYP	1	0...FFH	0...255	1	[-]
Switch of generatur identify					
C_VB_MIN_BSD_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
Battery Voltage threshold of power management diagnosis					

43.6.3 Component Driver (Interface) for Intelligent Battery Sensor IBS - %ZWKTIBS 1.20

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
BSD_SENS_BAT_SMT_INFO	V/O	0...FFH	0...255	1	[-]
Content of register					
BSD_SENS_BAT_SMT_ADR	V/O	0...FFH	0...255	1	[-]
Actual register number					
LF_BSD_SENS_BAT_SMT_CTL	V/O	0...FFH	0...255	1	[-]
Communication coordination byte (logical field)					
LV_SENS_BAT_SMT_ACT	V/O	0...1H	0...1	1	[-]
Communication to IBS has been detected as alive					

Input data:

LV_ERR_BSD	LV_VAR_BN	LV_VAR_VEH	CTR_MSG_SENS_BAT_SMT_COM_DIAG
CTR_ABC_SENS_BAT_SMT_COM			

FUNCTION DESCRIPTION:

Beschreibung der Schnittstelle HW-Treiber – Komponententreiber für die Verbindung IBS – DME

General information:

Der intelligente Batteriesensor bietet eine Reihe unterschiedlicher Einsatzmöglichkeiten und wird darum auf längere Sicht Gegenstand der Entwicklung sein. Dabei ist abzusehen, dass bei den unterschiedlichen Nutzungen auch wechselnde Variablen, Parameter und sonstige Datenstrukturen von und zum Sensor übertragen werden müssen. Es ist daher sinnvoll, in der Kommunikation zwischen Host und Sensor eine invariante Schnittstelle zu definieren, die die variablen Schichten des Komponententreibers und der Anwendersoftware vom Hardwaretreiber trennt, der nur einmal implementiert wird.

Description:

A) Realisierung Schnittstelle

Zum Datenaustausch in beiden Richtungen werden 3 Variable vom Typ unsigned char verwendet.:
BSD_SENS_BAT_SMT_INFO, BSD_SENS_BAT_SMT_ADR und LF_BSD_SENS_BAT_SMT_CTL. Sie sind im Modul des Hardwaretreibers global vereinbart.

BSD_SENS_BAT_SMT_INFO überträgt den Inhalt eines der acht BSD Register.

BSD_SENS_BAT_SMT_ADR (0 .. 7) adressiert das gewünschte Register.

LF_BSD_SENS_BAT_SMT_CTL enthält 7 Flags, die als Anforderung, Quittung, Fehlermeldung etc. die Kommunikation zwischen Komponententreiber (KT) und Hardwaretreiber (HT) koordinieren.

Um eine möglichst schnelle Übertragung zu erreichen, müssen der KT und der HT nacheinander in dieser Reihenfolge von der gleichen zyklischen Task aufgerufen werden.

B) Flags zur Kommunikation Hardware-/ Komponententreiber, in LF_BSD_SENS_BAT_SMT_CTL

Name	***Bit***	***gesetzt durch***	***gelöscht durch***	***Bedeutung***
BSDHTOK LV_SENS_BAT_SMT_BSD_DR_OK	0	HT	HT	Hardware Treiber ist bereit zur Übertragung; Flag bleibt an solange kein Fehlerzustand
BSDWRITE **LV_SENS_BAT_SMT_BSD_WR**	1	KT	KT	Richtung, hi: Register im Sensor schreiben
BSDDO **LV_SENS_BAT_SMT_BSD_RUN**	2	KT	HT	Anforderung von Komponenten- an Hardwaretreiber zum Start einer Übertragung
BSDBSY LV_SENS_BAT_SMT_BSD_WORK	3	HT	HT	Übertragung aktiv, Hardwaretreiber busy; Flag wird z.Zt. vom KT nicht benutzt
BSDRXRDY **LV_SENS_BAT_SMT_BSD_TRM_END_AVL**	4	HT	KT	Übertragung beendet, eingelesenes Byte ist verfügbar; Flag von Handler () im HT gesetzt
BSDTXOK **LV_SENS_BAT_SMT_BSD_TRM_END_OK**	5	HT	KT	Übertragung beendet, Byte wurde erfolgreich gesendet; Flag von Handler () im HT gesetzt
BSDERR LV_SENS_BAT_SMT_BSD_ERR_IT	7	HT	HT	BSD Fehler

Nur die fett gedruckten Flags werden für die laufende Kommunikation zwischen Komponenten- und Hardwaretreiber benötigt.

C) Ablauf der Übertragung eines BSD Telegramms

1. Nach „KL15 an“ löscht die Hardwaretreiber- Initialisierung alle Flags in LF_BSD_SENS_BAT_SMT_CTL.

2. Der Hardwaretreiber baut die Verbindung zur Schnittstellen- Hardware auf. Nach erfolgreicher Verbindung und bei Funktionsbereitschaft setzt der HT das Flag LV_SENS_BAT_SMT_BSD_DR_OK.

3. Ist ausschließlich das Bit LV_SENS_BAT_SMT_BSD_DR_OK in LF_BSD_SENS_BAT_SMT_CTL gesetzt, interpretiert der KT dies als Startbedingung und setzt seine zyklische Kommunikation zum Sensor auf.

4. Der Komponententreiber prüft LV_SENS_BAT_SMT_BSD_DR_OK vor jeder Aktion. Ist der HT bereit, wird das gewünschte Register in BsdReg adressiert. Bei Schreiboperationen wird das Bit LV_SENS_BAT_SMT_BSD_WR und das Datenbyte BSD_SENS_BAT_SMT_INFO belegt. Der KT setzt das Bit LV_SENS_BAT_SMT_BSD_RUN.

5. Der Hardwaretreiber erkennt an LV_SENS_BAT_SMT_BSD_RUN die Anforderung zur Übertragung. Soll gesendet werden, übermittelt er das Datenbyte an die Hardware und startet die Sendung zum gewünschten Register, andernfalls veranlasst er den Empfang eines Bytes aus dem Sensor. Der HT löscht dann die Anforderung LV_SENS_BAT_SMT_BSD_RUN und setzt statt dessen das Flag LV_SENS_BAT_SMT_BSD_WORK.

6. Das Ende einer Übertragung wird je nach Implementierung durch schnelles Polling oder per Interrupt festgestellt, der HT enthält eine entsprechende Handler Routine. Diese löscht LV_SENS_BAT_SMT_BSD_WORK, kopiert nach einem Lesevorgang das eingelesene Byte nach BSD_SENS_BAT_SMT_INFO und setzt das Flag LV_SENS_BAT_SMT_BSD_TRM_END_AVL zum Zeichen, dass der Komponententreiber ein empfangenes Byte auslesen kann. Nach einem erfolgreichen Schreibvorgang wird nur das Flag LV_SENS_BAT_SMT_BSD_TRM_END_OK gesetzt.

7. Der Komponententreiber liest LV_SENS_BAT_SMT_BSD_TRM_END_AVL bzw. LV_SENS_BAT_SMT_BSD_TRM_END_OK und löscht diese Flags, nachdem er ggf. das eingelesene Byte übernommen hat. Nach einem erneuten Test auf LV_SENS_BAT_SMT_BSD_DR_OK kann die nächste Übertragung durch Setzen von LV_SENS_BAT_SMT_BSD_RUN gestartet werden, s. 4).

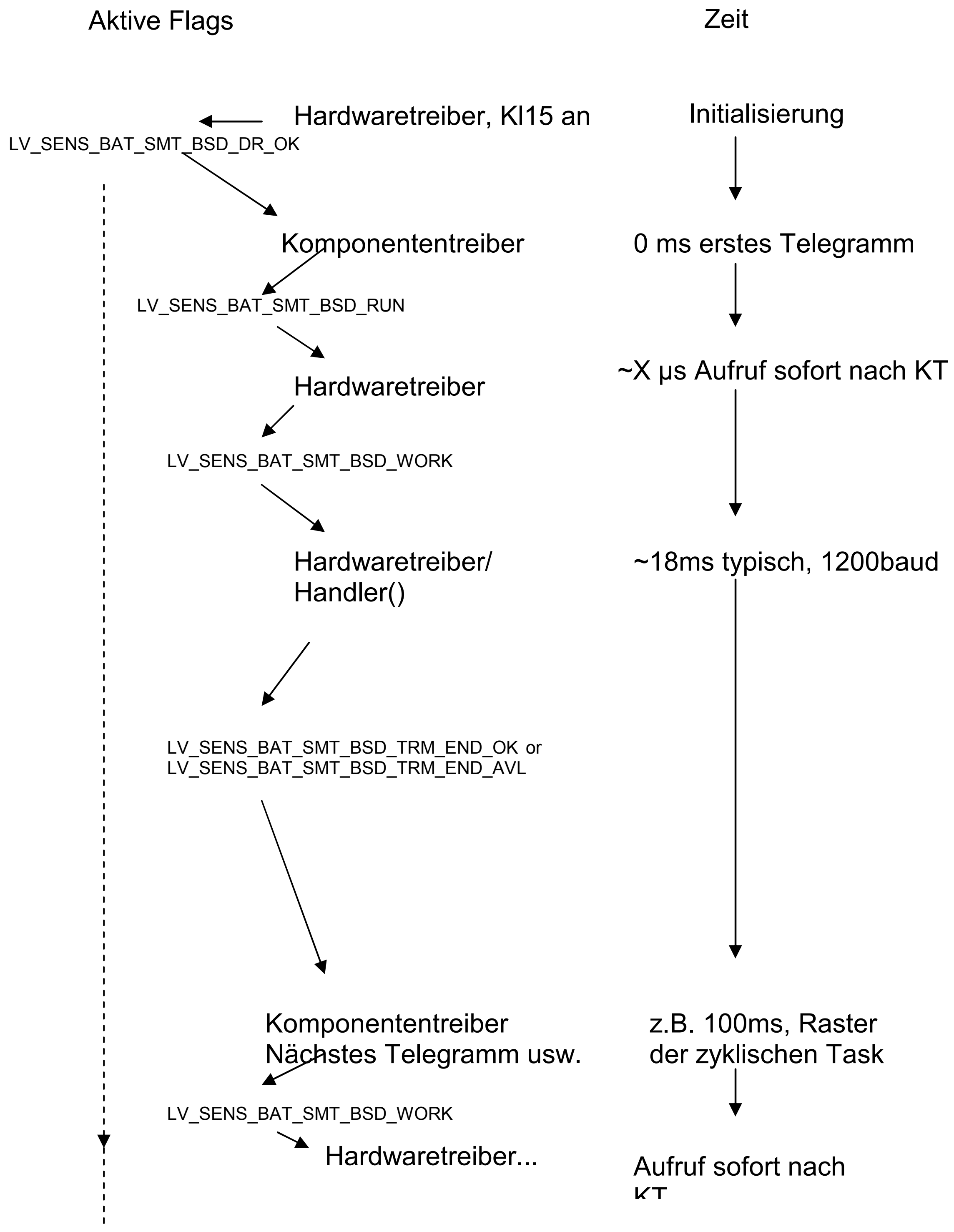
Bemerkungen:
Das Flag LV_SENS_BAT_SMT_BSD_WORK wird z.Zt. nur HT- intern verwendet, der KT greift nicht darauf zurück.
Bei interruptgesteuerten Handler Routinen muss die Möglichkeit einer Unterbrechung von HT oder KT durch den Interrupt berücksichtigt werden. Im Komponententreiber wird darum eine lokale Kopie von LF_BSD_SENS_BAT_SMT_CTL verwendet.

D) Fehlerbehandlung durch den HT

Wenn der Hardwaretreiber initial die Verbindung nicht herstellen kann oder wenn zwischenzeitlich die Verbindung abreisst, wird das Flag LV_SENS_BAT_SMT_BSD_DR_OK nicht gesetzt bzw. gelöscht.
In der Folge werden laufend neue Versuche zur Verbindung unternommen. Ist nach Ablauf einer entsprechenden Wartezeit Ibs_Ht_TimeOut keine Verbindung zustande gekommen, wird das Flag LV_SENS_BAT_SMT_BSD_ERR_IT gesetzt und alle anderen Flags gelöscht.
Der HT nimmt anschliessend einen Idle- Zustand ein, aus dem er nach der Zeit Ibs_Ht_IdleTime erwacht und wieder für Ibs_Ht_TimeOut weitere Initialisierungen versucht. Bei Erfolg kann wieder LV_SENS_BAT_SMT_BSD_DR_OK gesetzt und alle anderen Flags gelöscht werden, andernfalls bleibt die Fehlermeldung stehen.

Ibs_Ht_TimeOut:	20 Lese/ Schreibversuche
Ibs_Ht_IdleTime:	applizierbar, vorbelegt 100s

E) Schema Flagaustausch zwischen Komponenten- und Hardwaretreiber



Application conditions:

```
Activation:    if LC_POW_CORD_INH = 0   and
                  LV_VAR_BN = 1         and
                  LV_VAR_VEH = 0
Deactivation:     -
Update rate:      100ms
```

Formula section:

Setting of LV_SENS_BAT_SMT_ACT:

```
if      ReadyStatus from BSD-interface
then    LV_SENS_BAT_SMT_ACT = 1
endif
```

Re-setting of LV_SENS_BAT_SMT_ACT:

```
if      BSD-control-flag IBS_BSD = 1
then    LV_SENS_BAT_SMT_ACT = 0
endif
```

Control of error interface across the BMW Layer for the PWSL OBJ file

As soon as a certain number of communications between the IBS and the ECU were not succesful the bit LV_SENS_BAT_SMT_ERR_IT is set. This bit is part of the logical field LF_BSD_SENS_BAT_SMT_CTL (bit 7). This logical field is then copied across the BMW Layer and used in the PWSL functionality. In case of a global BSD error the bit is set as well. Due to the fact that the single BSD error is de-bounced with a anti bounce counter and the counter CTR_MSG_SENS_BAT_SMT_COM_DIAG is reset to 0 after each increment of the anti bounce counter the anti bounce counter has to be taken into account as well.

```
IF      LV_SENS_BAT_SMT_DET = 1                         and
        [ CTR_MSG_SENS_BAT_SMT_COM_DIAG ≥ 1        or
          CTR_ABC_SENS_BAT_SMT_COM ≥ 1             or
          LV_ERR_BSD = 1                              ]
THEN    LV_SENS_BAT_SMT_BSD_ERR_IT = 1
EISE    LV_SENS_BAT_SMT_BSD_ERR_IT = 0
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_POW_CORD_INH	1	0...1H	0...1	1	-
Configuration switch of alternator powermanagement					

43.6.4 Component Driver for electrical water pump

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_REL_CWP	V/O	0...FFH	0...255	1	[-]
Register 1 Bit 0 (ewpiwbsd) / actual rpm stage of the el. water pump					
V_CWP	V/O	0...FFH	0...25.5	0.1	[V]
Register 2 (ewpu) / Supply voltage at el. Water pump					
TEMP_EL_CWP	V/O	0...FFH	-50...205	1	[°C]
electronic temperature coolant water pump / Register 3 (ewpt)					
CUR_CNS_CWP	V/O	0...FFH	0...127.5	0.5	[A]
Register 4 (ewpi) / required current of the el. Water pump					
LV_N_MON_CWP_DEAC	V/O	0...1H	0...1	1	[-]
Register 6 Bit 1 (ewpnoff) / no CWP speed monitoring active					
LV_CWP_HW_LIH_IN_CHK	V/O	0...1H	0...1	1	[-]
Register 6 Bit 2 / Supply voltage to CWP (HW PIN) connected correctly - LIH stategy in case of BSD com. error					
LV_CWP_TEMP_HIGH	V/O	0...1H	0...1	1	[-]
Register 6 Bit 3 (ewpft) / over-temperature shut down					
LV_CWP_LOCK	V/O	0...1H	0...1	1	[-]
Register 6 Bit 4 (ewpblk1) / CWP blocked or requests a too high current					
LV_CWP_PRE_LOCK	V/O	0...1H	0...1	1	[-]
Register 6 Bit 5 (ewptlf) / CWP running dry					
LV_CWP_VCC_PLAUS	V/O	0...1H	0...1	1	[-]
Register 6 Bit 6 (ewpfu) / voltage supply not in the correct range					
LV_CWP_BLOCK_DEAC	V/O	0...1H	0...1	1	[-]
Register 6 Bit 7 (ewpdblk) / automatic de-blocking active					
LF_POW_CWP_COD	V/O	0...FFH	0...255	1	[-]
Register 5 Bit 0...4 / CWP power HW version (200W or 350/400W)					
LF_CWP_INFO_COD	V/O	0...FFH	0...255	1	[-]
Register 5 Bit 5...7 <<5 / Supplier information (Pierburg / SiemensVDO)					

Input data:

N_REL_CWP_SP			

FUNCTION DESCRIPTION:

General information:

Register 0 and 6 is continuously updated although the LH requires a demand oriented writing of the messages (e.g. N_REL_CWP_SP). This is due to the fact that the CWP SW expectes at least all 2,5secs an updated information otherwise LIH mode is activated by the CWP SW.

Description:

Please find in the following section the **r/w** operations for the registers of the CWP.

1.	Time slot for CWP	KT_WE_ST_**WRITE**_REG0	Speed Setpoint
2.	Time slot for CWP	KT_WE_ST_*READ*_REG5	Variant
3.	Time slot for CWP	KT_WE_ST_*READ*_REG2	Supply Voltage
4.	Time slot for CWP	KT_WE_ST_*READ*_REG1	Actual Speed
5.	Time slot for CWP	KT_WE_ST_*READ*_REG6	Error Section
6.	Time slot for CWP	KT_WE_ST_**WRITE**_REG0	Speed Setpoint
7.	Time slot for CWP	KT_WE_ST_*READ*_REG3	Temperature Electronics
8.	Time slot for CWP	KT_WE_ST_*READ*_REG2	Supply Voltage
9.	Time slot for CWP	KT_WE_ST_*READ*_REG1	Actual Speed
10.	Time slot for CWP	KT_WE_ST_*READ*_REG4	Current consumption
1.	.	.	
	.	.	
	.	.	

Identifikationsbeschreibung für Adresse 0x03 (Wasserpumpe Elektronik)
Bauteil: elektr. Wasserpumpe, Elektronik, Adresse: 3

Ident	Parameter	LS Bit	Bit Anz.	w(rite) r(ead)	Ein-heit	Funktion
0	WE_PDZS	0	8	-/w	[-]	Setpoint speed CWP ewpswbsd) 0...250 setpoint stages 251...254 power latch request 255 delete powerlatch reqest
						N_REL_CWP_SP (u8)
1	WE_PDZI	0	8	r/-	[-]	Actual speed CWP (ewpiwbsd)
						N_REL_CWP
2				r/-	[V]	Supply voltage CWP (ewpu)
						V_CWP (u8)
3				r/-	[°C]	Temperature of CWP electronics (ewpt)
						TEMP_EL_CWP (u8)
4				r/-	[A]	current consumption CWP
						CUR_CNS_CWP (u8)
5	WE_VAR	0	5	r/-	[-]	Mounted CWP type (0 = 200Watt or 1= 350/400Watt)
						LF_POW_CWP_COD
		5	3	r/-	[-]	Supplier of CWP (0 = Pierburg / 1 = SiemensVDO)
						LF_CWP_INFO_COD
6	WE_ERR2	1	1	r/-	-	rpm monitoring CWP deactivated
						LV_N_MON_CWP_DEAC (flag)

	WE_ERR3	2	1	r/-	-	Continuous monitoring of HW power supply of the CWP; used for LIH strategy in case of BSD communication faulty (CWP)
						LV_CWP_HW_LIH_IN_CHK (flag)
	WE_ERR4	3	1	r/-	-	Electronics CWP overtemperature (ewpft)
						LV_CWP_TEMP_HIGH (flag)
	WE_ERR5	4	1	r/-	-	CWP blocked / Current consumption too high (ewpblk1)
						LV_CWP_LOCK (flag)
	WE_ERR6	5	1	r/-	-	CWP running dry
						LV_CWP_PRE_LOCK (flag)
	WE_ERR7	6	1	r/-	-	Voltage supply too high(ewpfu)
						LV_CWP_VCC_PLAUS (flag)
	WE_ERR8	7	1	r/-	-	Automatic de-blocking active
						LV_CWP_BLOCK_DEAC (flag)
7	WE_RES	0	8	r/-	-	Reserved for supplier

43.6.5 Aquisition values for QLT – sensor

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_OZFBSD	V/O	0...1H	0...1	1	[-]
failure in oil-BSD-bevice					
oznivr	O/V	0...FFH	0...255	1	mm
signal oil level					
ozpermr_w	O/V	0...FFFFH	0...1024	1	-
signal oil quality					
OZTEMPR	V/O	0...FFH	0...255	1	[°C]
signal oil temperature					
OZSTATUS	V/O	0...FFH	0...255	1	[-]
status of oil sensor					
ozleitr_w	O/V	0...FFFFH	0...27	0.000412	-
ÖZS: oil sensor raw value of oil resitance					
oznivpr_w	O/V	0...FFFFH	0...65535	1	-
ÖZS: oil sensor raw value of oil level calculated out of permitivity					
Ozoelkm	O/V	0...FFFFH	0...655350	10	km
Ölkilometer					

FUNCTION DESCRIPTION:

General information:

The oil sensor (QLT = Quality Level Temperature) delivers information about the oil temperature, the oil level and the oil quality. This sensor is located in oil pan. The communication between this oil sensor and the engine management systen (ECU) is done via BSD-interface.

Description:

A) Describtion of communication of the BSD-Interface

On the SW-side in the ECU the recieved BSD-messages are treated with a "component driver".
5 Registers are relevant:
0 Oiltemperatur
1 Oilniveau
2,3 Permittivity
6 Status / Variant
this values are updated every 200ms

In case of a BSD-communication error the bit B_ozfbsd is set

B) Describtion of the registers

Ident	Parameter	LS Bit	Bit Anz.	w(rite) r(ead)	Ein-heit	Function
0	OZ_TEMP	0	8	-/r	°C	Öltemperatur (8Bit)
1	OZ_NIV	0	8	-/r	mm	Ölstand (8Bit)
2	OZ_PERML	0	8	-/r	--	Permittivität/ Ölqualität (LSB von 16 Bit)
3	OZ_PERMM	8	2	-/r	--	Permittivität/ Ölqualität (MSB von 16 Bit)
4	--	--	--	--	--	not used
5	--	--	--	--	--	not used
6	OZ_ERRx	0	6	-/r	--	Status- und Fehlerinformationen: xxxxx1 = Sensorfehler Permittivität Anzeigekriterium: OZ-PERMM/OZ-PERML ungültig
						Status- und Fehlerinformationen: xxxx1x = Sensorfehler Ölniveau Anzeigekriterium: OZ_NIV ungültig
						Status- und Fehlerinformationen: xxx1xx = Sensorfehler Öltemperatur Anzeigekriterium: OZ_TEMP ungültig
	OZ_VAR	6	2	-/r	--	Sensortyp / Variante
						00 = offen 01 = TEMIC (Temp/Ölniveau/Permittivitätsz.) 10 = offen 11 = offen
7	OZ_RES	0	8	-/r	--	01 = reserviert für Prüfzwecke Lieferant rest = offen

Signal Öltemperatur:

Label : OZ_TEMP
Skalierungsbereich: -55,2 bis +175,4 = 01 h bis FE h [°C]
Plausibler Bereich: -40°C bis +160°C ⇒ Messwertebereich = 200 K
Messbereichsverletzung: 00 h ⇒ Messbereichsunterschreitung
FF h ⇒ Messbereichsüberschreitung
Fehlerkennzeichnung: DATA2 = 4 h ⇒ Sensorfehler Öltemperaturmessung
Auflösung: 8 bit

Signal Ölstand:

Label :	OZ_NIV	
Skalierungsbereich:	0.bis 75 = 01 h .bis.FE h [mm]	
Plausibler Bereich:0 bis	75 mm	⇒ Messwertebereich = 75 mm
Messbereichsverletzung:	00 h	⇒ Messbereichsunterschreitung
	FF h	⇒ Messbereichsüberschreitung
Fehlerkennzeichnung:	DATA2 = 2 h	⇒ Sensorfehler Ölstandsmessung
Auflösung:	8 bit	

Signal Ölzustand (Permittivitätszahl ε_r):

Label :	OZ_PERMM / OZ_PERML	
Skalierungsbereich:	1 bis 6 = 0001 h bis 03FE h	
Plausibler Bereich:	1 bis 6	⇒ Messwertebereich = 5
Messbereichsverletzung:	0000 h	⇒ Messbereichsunterschreitung
	03FF h	⇒ Messbereichsüberschreitung
Fehlerkennzeichnung:	DATA2 = 1 h	⇒ Sensorfehler Permittivitätsmessung
Auflösung:	10 bit	

Application conditions:

Initialisation: *all 0 at reset*

Recurrence: *200ms*

Activation: *at every engine state*

43.7 CAN (11H) diagnosis

FUNCTION DESCRIPTION

General information:

The following CAN diagnosis are defined:

- Timeout diagnosis of CAN message **ASC1**
- Timeout diagnosis of CAN message **ASC3**
- Timeout diagnosis of CAN message **ASC4**
- Timeout diagnosis of CAN message **EGS1**
- Timeout/Alive/Checksum diagnosis of CAN message **EGS2**
- Timeout diagnosis of CAN message **INSTR2**
- Timeout diagnosis of CAN message **INSTR3**
- Timeout diagnosis of CAN message **INSTR7**
- Timeout diagnosis of CAN message **LWS1**
- Timeout diagnosis of CAN message **EKP**

- Timeout/Alive/Complement diagnosis of CAN message **SSG1**

For calculation of LV_END_DIAG_xxx, see “Calculation end of diagnosis” in chapter “Anti-bounce algorithm.

43.7.1 Timeout diagnosis of CAN message ASC1

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_ASR_1	V/O	0...1H	0...1	1	[-]
logical variable timeout error CAN message ASC1					
CYC_CTR_ASR_1_DIAG	V	0...FFH	0...255	1	[-]
Counter of not received ASC1 messages					
LV_CDN_DIAG_TOUT_ASR_1	V/O	0...1H	0...1	1	[-]
Diagnosis condition ASC1 messages					
ERR_SYM_TOUT_ASR_1	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom ASC1 messages					
LV_END_DIAG_TOUT_ASR_1	V/O	0...1H	0...1	1	[-]
End of diagnosis ASC1 messages					

Input data:

LV_CDN_VB_CAN_DIAG	LV_VAR_ASR	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_ASR_1_DIAG is always incremented If the CAN message ASC1 has not been received. The counter is set to 0 If a CAN message ASC1 has been received or If the counter has reached his maximum value C_ASR_1_CYCNR_MAX.
If this counter has reached the calibratable threshold C_ASR_1_CYCNR_MAX and there is no CAN message ASC1 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_ASR_1. If a CAN message ASC1 is received, the anti bounce counter is decremented.
If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_ASR_1 the timeout error CAN message ASC1 (LV_ERR_TOUT_ASR_1) is set.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> -                            (= SYM_0)
 │  │  └────> -                            (= SYM_1)
 │  └───────> Timeout signal ASC1          (= SYM_2)
 └──────────> -                            (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 ms

Activation: **If** LV_CDN_VB_CAN_DIAG = 1 **and**
LV_VAR_ASR = 1 **and**
LV_ERR_CAN_BOFF = 0
Then LV_CDN_DIAG_TOUT_ASR_1 = 1
Else LV_CDN_DIAG_TOUT_ASR_1 = 0

```
            Endif
```

Formula section:

Calculation of time-out counter

```
If          no CAN message ASC1 received
Then        $CYC\_CTR\_ASR\_1\_DIAG_{N} = CYC\_CTR\_ASR\_1\_DIAG_{N-1} + 1$
Else        CYC_CTR_ASR_1_DIAG   = 0     (reset)
EndIf
```

Detection of error symptom:

```
If(1)       CYC_CTR_ASR_1_DIAG = C_ASR_1_CYCNR_MAX
Then(1)     CYC_CTR_ASR_1_DIAG   = 0     (reset)
    If(2)       no CAN message ASC1 is received
    Then(2)     ERR_SYM_TOUT_ASR_1 = SYM_2
                Error symptom is detected, ABC counter starts to increment
                LV_ERR_TOUT_ASR_1     = 1     (after debounce)
    Else(2)     ERR_SYM_TOUT_ASR_1 = NO_SYM
                on Error symptom is detected, ABC counter starts to decrement
                LV_ERR_TOUT_ASR_1     = 0     (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ASR_1_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing ASC1 messages for error debounce counter increment					
C_ABC_INC_TOUT_ASR_1	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message ASC1					
C_ABC_MAX_TOUT_ASR_1	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message ASC1					

43.7.2 Timeout CAN message ASC3

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_ASR_3	V/O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message ASC3					
CYC_CTR_ASR_3_DIAG	V	0...FFH	0...255	1	[-]
Counter of not received ASC3messages					
LV_CDN_DIAG_TOUT_ASR_3	V/O	0...1H	0...1	1	[-]
Diagnosis condition ASC3messages					
ERR_SYM_TOUT_ASR_3	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom ASC3messages					
LV_END_DIAG_TOUT_ASR_3	V/O	0...1H	0...1	1	[-]
End of diagnosis ASC3messages					

Input data:

LV_CDN_VB_CAN_DIAG	LV_VAR_ASR_3	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_ASR_3_DIAG is always incremented If the CAN message ASC3 as not been received. The counter is set to 0 If a CAN message ASC3 has been received or If the counter has reached his maximum value C_ASR_3_CYCNR_MAX.
If this counter has reached the calibratable threshold C_ASR_3_CYCNR_MAX and there is no CAN message ASC3 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_ASR_3. If a CAN message ASC3 is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_ASR_3 the timeout error CAN message ASC3 (LV_ERR_TOUT_ASR_3) is set.

Error symptoms are defined for this diagnosis function as:

```
□□□□
| | | └─> -                        (= SYM_0)
| | └───> -                        (= SYM_1)
| └─────> Timeout signal ASC3      (= SYM_2)
└───────> -                        (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 ms

Activation: **If** LV_CDN_VB_CAN_DIAG = 1 **and**

LV_VAR_ASR_3 = 1 **and**

LV_ERR_CAN_BOFF = 0

Then LV_CDN_DIAG_TOUT_ASR_3 = 1

```
        Else    LV_CDN_DIAG_TOUT_ASR_3 = 0
        Endif
```

Formula section:

Calculation of time-out counter

```
If      no CAN message ASC3 received
Then    $CYC\_CTR\_ASR\_3\_DIAG_{N} = CYC\_CTR\_ASR\_3\_DIAG_{N-1} + 1$
Else    CYC_CTR_ASR_3_DIAG    = 0     (reset)
EndIf
```

Detection of error symptom

```
If(1)       CYC_CTR_ASR_3_DIAG = C_ASR_3_CYCNR_MAX
Then(1)     CYC_CTR_ASR_3_DIAG    = 0     (reset)
    If(2)       no CAN message ASC3 is received
    Then(2)     ERR_SYM_TOUT_ASR_3 = SYM_2
                Error symptom is detected, ABC counter starts to increment
                LV_ERR_TOUT_ASR_3      = 1     (after debounce)
    Else(2)     ERR_SYM_TOUT_ASR_3 = NO_SYM
                no Error symptom is detected, ABC counter starts to decrement
                LV_ERR_TOUT_ASR_3      = 0     (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ASR_3_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing ASC3 messages for error debounce counter increment					
C_ABC_INC_TOUT_ASR_3	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message ASC3					
C_ABC_MAX_TOUT_ASR_3	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message ASC3					

43.7.3 Timeout CAN message ASC4

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_ASR_4	V/O	0...1H	0...1	1	[-]
logical variable timeout error CAN message ASC4					
CYC_CTR_ASR_4_DIAG	V	0...FFH	0...255	1	[-]
counter of not received ASC4messages					
LV_CDN_DIAG_TOUT_ASR_4	V/O	0...1H	0...1	1	[-]
Diagnosis condition ASC4messages					
ERR_SYM_TOUT_ASR_4	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom ASC4messages					
LV_END_DIAG_TOUT_ASR_4	V/O	0...1H	0...1	1	[-]
End of diagnosis ASC4messages					

Input data:

LV_CDN_VB_CAN_DIAG	LV_VAR_ASR_4	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_ASR_4_DIAG is always incremented if the CAN message ASC4 as not been received. The counter is set to 0 if a CAN message ASC4 has been received or if the counter has reached its maximum value C_ASR_4_CYCNR_MAX.
If this counter has reached the calibratable threshold C_ASR_4_CYCNR_MAX and there is no CAN message ASC4 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_ASR_4. If a CAN message ASC4 is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_ASR_4 the timeout error CAN message ASC4 (LV_ERR_TOUT_ASR_4) is set.

Error symptoms are defined for this diagnosis function as:

```
☐☐☐☐
 │ │ │ └─> -                               (= SYM_0)
 │ │ └───> -                               (= SYM_1)
 │ └─────> Timeout signal ASC4             (= SYM_2)
 └───────> -                               (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 20 ms

Activation: **If** LV_CDN_VB_CAN_DIAG = 1 **and**

LV_VAR_ASR_4 = 1 **and**

LV_ERR_CAN_BOFF = 0

Then LV_CDN_DIAG_TOUT_ASR_4 = 1

```
                    Else    LV_CDN_DIAG_TOUT_ASR_4 = 0
                    Endif
```

Formula section:

Calculation of time-out counter

```
If          no CAN message ASC4 received
Then        $CYC\_CTR\_ASR\_4\_DIAG_{N} = CYC\_CTR\_ASR\_4\_DIAG_{N-1} + 1$
Else        CYC_CTR_ASR_4_DIAG    = 0     (reset)
EndIf
```

Detection of error symptom

```
If(1)       CYC_CTR_ASR_4_DIAG = C_ASR_4_CYCNR_MAX
Then(1)     CYC_CTR_ASR_4_DIAG    = 0     (reset)
    If(2)       no CAN message ASC4 is received
    Then(2)     ERR_SYM_TOUT_ASR_4 = SYM_2
                Error symptom is detected, ABC counter starts to increment
                LV_ERR_TOUT_ASR_4     = 1     (after debounce)
    Else(2)     ERR_SYM_TOUT_ASR_4 = NO_SYM
                no Error symptom is detected, ABC counter starts to decrement
                LV_ERR_TOUT_ASR_4     = 0     (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ASR_4_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing ASC4 messages for error debounce counter increment					
C_ABC_INC_TOUT_ASR_4	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message ASC4					
C_ABC_MAX_TOUT_ASR_4	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message ASC4					

43.7.4 Timeout CAN message EGS1

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_ETCU_1	V/O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message EGS1					
CYC_CTR_ETCU_1_DIAG	V	0...FFH	0...255	1	[-]
Counter of not received EGS1 messages					
LV_CDN_DIAG_TOUT_ETCU_1	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN message EGS1					
ERR_SYM_TOUT_ETCU_1	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CAN message EGS1					
LV_END_DIAG_TOUT_ETCU_1	V/O	0...1H	0...1	1	[-]
End of diagnosis CAN message EGS1					

Input data:

LV_CDN_VB_CAN_DIAG	LV_AT	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_ETCU_1_DIAG is always incremented If the CAN message EGS1 as not been received. The counter is set to 0 If a CAN message EGS1 has been received or If the counter has reached his maximum value C_ETCU_1_CYCNR_MAX.
If this counter has reached the calibratable threshold C_ETCU_1_CYCNR_MAX and there is no CAN message EGS1 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_ETCU_1. If a CAN message EGS1 is received, the anti bounce counter is decremented.
If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_ETCU_1 the timeout error CAN message EGS1 (LV_ERR_TOUT_ETCU_1) is set.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
 │ │ │ └─> -                              (= SYM_0)
 │ │ └───> -                              (= SYM_1)
 │ └─────> Timeout signal EGS1            (= SYM_2)
 └───────> -                              (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 ms

Activation: **If** LV_CDN_VB_CAN_DIAG = 1 **and**

LV_AT = 1 **and**

LV_ERR_CAN_BOFF = 0

Then LV_CDN_DIAG_TOUT_ETCU_1 = 1

Else LV_CDN_DIAG_TOUT_ETCU_1 = 0

```
Endif
If      LV_AT = 0   (AT not learnt if LV_IGK = 1 after 3s)
Then    LV_END_DIAG_TOUT_ETCU_1 = 1          (set for readiness)
Endif
```

Formula section:

Calculation of time-out counter

```
If          no CAN message EGS1 received
Then        $CYC\_CTR\_ETCU\_1\_DIAG_{N} = CYC\_CTR\_ETCU\_1\_DIAG_{N-1} + 1$
Else        CYC_CTR_ETCU_1_DIAG  = 0      (reset)
EndIf
```

Detection of error symptom:

```
If(1)       CYC_CTR_ETCU_1_DIAG = C_ETCU_1_CYCNR_MAX
Then(1)     CYC_CTR_ETCU_1_DIAG  = 0      (reset)
    If(2)       no CAN message EGS1 is received
    Then(2)     ERR_SYM_TOUT_ETCU_1 = SYM_2
                Error symptom is detected, ABC counter starts to increment
                LV_ERR_TOUT_ETCU_1    = 1      (after debounce)
    Else(2)     ERR_SYM_TOUT_ETCU_1 = NO_SYM
                no Error symptom is detected, ABC counter starts to decrement
                LV_ERR_TOUT_ETCU_1    = 0      (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ETCU_1_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing EGS1 messages for error debounce counter increment					
C_ABC_INC_TOUT_ETCU_1	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message EGS1					
C_ABC_MAX_TOUT_ETCU_1	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message EGS1					

43.7.5 Timeout CAN message EGS2

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_ETCU_2	V/O	0...1H	0...1	1	[-]
logical variable timeout error CAN message EGS2					
CYC_CTR_ETCU_2_DIAG	V	0...FFH	0...255	1	[-]
counter of not received EGS2 messages					
LV_CDN_DIAG_TOUT_ETCU_2	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN message EGS2					
ERR_SYM_TOUT_ETCU_2	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CAN message EGS2					
LV_END_DIAG_TOUT_ETCU_2	V/O	0...1H	0...1	1	[-]
End of diagnosis CAN message EGS2					
CYC_CTR_ETCU_2_TCC_DIAG	V	0...FFH	0...255	1	[-]
Counter for wrong Alive-Counter					
CYC_CTR_ETCU_2_CKS_DIAG	V	0...FFH	0...255	1	[-]
Counter for wrong checksum					

Input data:

LV_CDN_VB_CAN_DIAG	LV_AT	LV_ERR_CAN_BOFF	LV_IGK
TCC_CAN_ETCU_2	CKS_CAN_ETCU_2	CKS_CLC_ETCU_2	

Description:

The counter CYC_CTR_ETCU_2_DIAG is always incremented If the CAN message EGS2 has not been received. The counter is set to 0 If a CAN message EGS2 has been received or if the counter has reached his maximum value C_ETCU_2_CYCNR_MAX.
If this counter has reached the calibratable threshold C_ETCU_2_CYCNR_MAX and there is no CAN message EGS2 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_ETCU_2. If a CAN message EGS2 is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_ETCU_2 the timeout error CAN message EGS2 (LV_ERR_TOUT_ETCU_2) is set.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                      (= SYM_0)
 │ │ └───> Alive-Counter          (= SYM_1)
 │ └─────> Timeout signal EGS2    (= SYM_2)
 └───────> Checksum               (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 ms

Activation: **If** LV_CDN_VB_CAN_DIAG = 1 **and**

```
                LV_AT = 1                               and
                LV_ERR_CAN_BOFF = 0
        Then    LV_CDN_DIAG_TOUT_ETCU_2 = 1
        Else    LV_CDN_DIAG_TOUT_ETCU_2 = 0
        Endif
        If      LV_AT = 0   (AT not learnt if LV_IGK = 1 after 3s)
        Then    LV_END_DIAG_TOUT_ETCU_2 = 1          (set for readiness)
        Endif
```

Formula section:

Calculation of time-out counter

```
If          no CAN message EGS2 received
Then        $\mathrm{CYC\_CTR\_ETCU\_2\_DIAG}_{N} = \mathrm{CYC\_CTR\_ETCU\_2\_DIAG}_{N-1} + 1$
Else        CYC_CTR_ETCU_2_DIAG = 0     (reset)
EndIf
```

Detection of error symptom "timeout"

```
If(1)       CYC_CTR_ETCU_2_DIAG = C_ETCU_2_CYCNR_MAX
Then(1)     CYC_CTR_ETCU_2_DIAG = 0     (reset)
    If(2)       no CAN message EGS2 is received
    Then(2)     ERR_SYM_TOUT_ETCU_2 = SYM_2
                Error symptom is detected, ABC counter starts to increment
                LV_ERR_TOUT_ETCU_2 = 1      (after debounce)
    Else(2)     ERR_SYM_TOUT_ETCU_2 = NO_SYM
                no Error symptom is detected, ABC counter starts to decrement
                LV_ERR_TOUT_ETCU_2 = 0      (after rebounce)
    EndIf(2)
EndIf(1)
```

Detection of error symptom "Alivecounter"

```
If(1)  CYC_CTR_ETCU_2_DIAG = 0                 no detected timeout   and
       $\mathrm{CYC\_CTR\_ETCU\_2\_CKS\_DIAG}_{n} = 0$      no detected CKS error
Then(1)If(2) $\mathrm{TCC\_CAN\_ETCU\_2}_{n} = \mathrm{TCC\_CAN\_ETCU\_2}_{n-1}$
    Then(2) $\mathrm{CYC\_CTR\_ETCU\_2\_TCC\_DIAG}_{n} = \mathrm{CYC\_CTR\_ETCU\_2\_TCC\_DIAG}_{n-1} + 1$
        If(3)   CYC_CTR_ETCU_2_TCC_DIAG >=
                C_ETCU_2_TCC_CYCNR_MAX
```

```
                Then(3) ERR_SYM_TOUT_ETCU_2 = SYM_1
                        Error symptom “Alive-Counter ” is active, ABC counter starts to increment
                        LV_ERR_TOUT_ETCU_2    = 1    (after debounce)
               Else(3) ERR_SYM_TOUT_ETCU_2 = NO_SYM
                        no Error symptom “Alive-Counter” is passive, anti-bounce decrement
                        LV_ERR_TOUT_ETCU_2    = 0    (after rebounce)
               EndIf(3)
        Else(2) CYC_CTR_ETCU_2_TCC_DIAG = 0              reset, no error detection
        EndIf(2)
    Else(1)     no detection
    Endif(1)
```

Detection of error symptom “checksum is wrong”

```
    If(1) CYC_CTR_ETCU_2_DIAG  = 0              no detected timeout
    ThenIf(2) CKS_CAN_ETCU_2  ≠ CKS_CLC_ETCU_2 (sent ≠ calculated)
        Then(2) CYC_CTR_ETCU_2_CKS_DIAG$_{n}$ = CYC_CTR_ETCU_2_CKS_DIAG$_{n-1}$ +1
          If(3) CYC_CTR_ETCU_2_CKS_DIAG >= C_ETCU_2_CKS_CYCNR_MAX
          Then(3) ERR_SYM_TOUT_ETCU_2 = SYM_3
                  Error symptom “CKS” is active, ABC counter starts to increment
                 LV_ERR_TOUT_ETCU_2        = 1    (after debounce)
          Else(3) ERR_SYM_TOUT_ETCU_2 = NO_SYM
                  no Error symptom “CKS” is passive anti-bounce decrement
                  LV_ERR_TOUT_ETCU_2       = 0    (after rebounce)
          EndIf(3)
        Else(2) CYC_CTR_ETCU_2_CKS_DIAG = 0             reset, no error detection
        EndIf(2)
    Else(1)     no error detection
    EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ETCU_2_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing EGS2 messages for error debounce counter increment					
C_ABC_INC_TOUT_ETCU_2	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message EGS2					
C_ABC_MAX_TOUT_ETCU_2	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message EGS2					
C_ETCU_2_TCC_CYCNR_MAX	1	0...FFH	0...255	1	[-]
Maximum number of wrong alive counter events for error debounce counter increment					
C_ETCU_2_CKS_CYCNR_MAX	1	0...FFH	0...255	1	[-]
Maximum number of wrong complement messages for error debounce counter increment					

43.7.6 Timeout / Alive / Complement diagnosis of CAN message SSG1

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_AMT_1	V/O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message SSG1					
CYC_CTR_AMT_1_DIAG	V	0...FFH	0...255	1	[-]
Counter of not received SSG1 messages					
LV_CDN_DIAG_TOUT_AMT_1	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN message SSG1					
ERR_SYM_TOUT_AMT_1	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CAN message SSG1					
LV_END_DIAG_TOUT_AMT_1	V/O	0...1H	0...1	1	[-]
End of diagnosis CAN message SSG1					

Input data:

LV_CDN_VB_CAN_DIAG	LV_VAR_AMT	TQI_AMT_REQ_CAN	TQI_AMT_CPL_CAN
TCC_CAN_AMT_1	LV_ERR_CAN_BOFF	LV_IGK	

Description:

The counter CYC_CTR_AMT_1_DIAG is always incremented If the CAN message SSG1 as not been received. The counter is set to 0 If a CAN message SSG1 has been received or If the counter has reached his maximum value C_AMT_1_CYCNR_MAX.
If this counter has reached the calibratable threshold C_AMT_1_CYCNR_MAX and there is no CAN message SSG1 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_AMT_1. If a CAN message SSG1 is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_AMT_1 the timeout error CAN message SSG1 (LV_ERR_TOUT_AMT_1) is set.

Error symptoms are defined for this diagnosis function as:

```
□□□□
| | | └> -                                   (= SYM_0)
```

```
│ │ └───> Alive-Counter          (= SYM_1)
│ └─────> Timeout signal SSG1    (= SYM_2)
└───────> Complement             (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 ms

Activation:

```
If      LV_CDN_VB_CAN_DIAG = 1         and
        LV_VAR_AMT = 1                 and
        LV_ERR_CAN_BOFF = 0
Then    LV_CDN_DIAG_TOUT_AMT_1 = 1
Else    LV_CDN_DIAG_TOUT_AMT_1 = 0
Endif
If      LV_VAR_AMT = 0   (AMT not learnt if LV_IGK = 1 after 3s)
Then    LV_END_DIAG_TOUT_AMT_1 = 1  (set for readiness)
Endif
```

Formula section:

Calculation of time-out counter

```
If      no CAN message SSG1 received
Then    $CYC\_CTR\_AMT\_1\_DIAG_{N} = CYC\_CTR\_AMT\_1\_DIAG_{N-1} + 1$
Else    CYC_CTR_AMT_1_DIAG = 0   (reset)
EndIf
```

Detection of error symptom "timeout"

```
If(1)     CYC_CTR_AMT_1_DIAG = C_AMT_1_CYCNR_MAX
Then(1)   CYC_CTR_AMT_1_DIAG = 0   (reset)
    If(2)     no CAN message SSG1 is received
    Then(2)   ERR_SYM_TOUT_AMT_1 = SYM_2
              Error symptom "Timeout" is active, ABC counter starts to increment
              LV_ERR_TOUT_AMT_1 = 1   (after debounce)
    Else(2)   ERR_SYM_TOUT_AMT_1 = NO_SYM
              no Error symptom "Timeout" is passive, anti-bounce decrement
              LV_ERR_TOUT_AMT_1 = 0   (after rebounce)
    EndIf(2)
EndIf(1)
```

Detection of error symptom "Alivecounter"

```
If(1) CYC_CTR_AMT_1_DIAG    = 0               no detected timeout   and
      CTR_DIAG_CPL_CAN_AMT_1$_{n}$     = 0    no detected complement error
Then(1)If(2) TCC_CAN_AMT_1$_{n}$ = TCC_CAN_AMT_1$_{n-1}$
    Then(2) CTR_DIAG_TCC_CAN_AMT_1$_{n}$ = CTR_DIAG_TCC_CAN_AMT_1$_{n-1}$ +1
        If(3)                     CTR_DIAG_TCC_CAN_AMT_1                      >=
      C_CTR_DIAG_TCC_CAN_MAX_AMT_1
        Then(3) ERR_SYM_TOUT_AMT_1 = SYM_1
                Error symptom “Alive-Counter ” is active, ABC counter starts to increment
                LV_ERR_TOUT_AMT_1      = 1      (after debounce)
        Else(3) ERR_SYM_TOUT_AMT_1 = NO_SYM
                no Error symptom “Alive-Counter” is passive, anti-bounce decrement
                LV_ERR_TOUT_AMT_1      = 0      (after rebounce)
        EndIf(3)
    Else(2) CTR_DIAG_TCC_CAN_AMT_1 = 0                     reset, no error detection
    EndIf(2)
Else(1)     no detection
Endif(1)
```

Detection of error symptom “complement”

```
If(1) CYC_CTR_AMT_1_DIAG    = 0               no detected timeout
ThenIf(2) TQI_AMT_REQ_CAN + TCC_CAN_AMT_1 – TQI_AMT_CPL_CAN ≠ 0
Then(2) CTR_DIAG_CPL_CAN_AMT_1$_{n}$ = CTR_DIAG_CPL_CAN_AMT_1$_{n-1}$ +1
      If(3) CTR_DIAG_CPL_CAN_AMT_1  >= C_CTR_DIAG_CPL_CAN_MAX_AMT_1
      Then(3) ERR_SYM_TOUT_AMT_1 = SYM_3
              Error symptom “Complement” is active, ABC counter starts to increment
              LV_ERR_TOUT_AMT_1 = 1      (after debounce)
      Else(3) ERR_SYM_TOUT_AMT_1 = NO_SYM
              no Error symptom “Complement” is passive anti-bounce decrement
              LV_ERR_TOUT_AMT_1= 0      (after rebounce)
      EndIf(3)
    Else(2) CTR_DIAG_CPL_CAN_AMT_1 = 0                     reset, no error detection
    EndIf(2)
Else(1)     no error detection
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_AMT_1_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing SSG1 messages for error debounce counter increment					
C_ABC_INC_TOUT_AMT_1	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message SSG1					
C_ABC_MAX_TOUT_AMT_1	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message SSG1					
C_CTR_DIAG_TCC_CAN_MAX_AMT_1	1	0...FFH	0...255	1	[-]
Maximum number of wrong alive counter events for error debounce counter increment					
C_CTR_DIAG_CPL_CAN_MAX_AMT_1	1	0...FFH	0...255	1	[-]
Maximum number of wrong complement messages for error debounce counter increment					

43.7.7 Timeout CAN message INSTR2

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_ICL_2	V/O	0...1H	0...1	1	[-]
logical variable timeout error CAN message INSTR2					
CYC_CTR_ICL_2_DIAG	V	0...FFH	0...255	1	[-]
counter of not received INSTR2 messages					
LV_CDN_DIAG_TOUT_ICL_2	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN message INSTR2					
ERR_SYM_TOUT_ICL_2	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CAN message INSTR2					
LV_END_DIAG_TOUT_ICL_2	V/O	0...1H	0...1	1	[-]
End of diagnosis CAN message INSTR2					

Input data:

LV_CDN_VB_CAN_DIAG	LV_VAR_ICL	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_ICL_2_DIAG is always incremented If the CAN message INSTR2 has not been received. The counter is set to 0 If a CAN message INSTR2 has been received or If the counter has reached his maximum value C_ICL_2_CYCNR_MAX.
If this counter has reached the calibratable threshold C_ICL_2_CYCNR_MAX and there is no CAN message ASC1 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_ICL_2. If a CAN message INSTR2 is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_ICL_2 the timeout error CAN message INSTR2 (LV_ERR_TOUT_ICL_2) is set.

Error symptoms are defined for this diagnosis function as:


```
 │  │  │  └─> -                                   (= SYM_0)
 │  │  └────> -                                   (= SYM_1)
 │  └───────> Timeout signal INSTR2               (= SYM_2)
 └──────────> -                                   (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 200 ms

Activation:

```
If      LV_CDN_VB_CAN_DIAG =  1        and
        LV_VAR_ICL = 1                 and
        LV_ERR_CAN_BOFF = 0
Then    LV_CDN_DIAG_TOUT_ICL_1 = 1
Else    LV_CDN_DIAG_TOUT_ICL_1 = 0
Endif
```

Formula section:

Calculation of time-out counter

```
If      no CAN message INSTR2 received
Then    $CYC\_CTR\_ICL\_2\_DIAG_{N}$  = $CYC\_CTR\_ICL\_2\_DIAG_{N-1}$ +1
Else    CYC_CTR_ICL_2_DIAG    = 0     (reset)
EndIf
```

Detection of error symptom

```
If(1)       CYC_CTR_ICL_2_DIAG = C_ICL_2_CYCNR_MAX
Then(1)     CYC_CTR_ICL_2_DIAG     = 0     (reset)
    If(2)       no CAN message INSTR2 is received
    Then(2)     ERR_SYM_TOUT_ICL_2 = SYM_2
                Error symptom is detected, ABC counter starts to increment
                LV_ERR_TOUT_ICL_2       = 1     (after debounce)
    Else(2)     ERR_SYM_TOUT_ICL_2 = SYM_2
                no Error symptom is detected, ABC counter starts to decrement
                LV_ERR_TOUT_ICL_2       = 0     (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ICL_2_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing INSTR2 messages for error debounce counter increment					
C_ABC_INC_TOUT_ICL_2	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message INSTR2					
C_ABC_MAX_TOUT_ICL_2	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message INSTR2					

43.7.8 Timeout CAN message INSTR3

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_ICL_3	V/O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message INSTR3					
CYC_CTR_ICL_3_DIAG	V	0...FFH	0...255	1	[-]
Counter of not received INSTR3 messages					
LV_CDN_DIAG_TOUT_ICL_3	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN message INSTR3					
ERR_SYM_TOUT_ICL_3	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CAN message INSTR3					
LV_END_DIAG_TOUT_ICL_3	V/O	0...1H	0...1	1	[-]
End of diagnosis CAN message INSTR3					

Input data:

LV_CDN_VB_CAN_DIAG	LV_VAR_ICL	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_ICL_3_DIAG is always incremented If the CAN message INSTR3 has not been received. The counter is set to 0 If a CAN message INSTR3 has been received or If the counter has reached his maximum value C_ICL_3_CYCNR_MAX.
If this counter has reached the calibratable threshold C_ICL_3_CYCNR_MAX and there is no CAN message INSTR3 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_ICL_3. If a CAN message INSTR3 is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_ICL_3 the timeout error CAN message INSTR3 (LV_ERR_TOUT_ICL_3) is set.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                       (= SYM_0)
 │ │ └───> -                       (= SYM_1)
 │ └─────> Timeout signal INSTR3   (= SYM_2)
 └───────> -                       (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 200 ms

Activation:

```
If      LV_CDN_VB_CAN_DIAG =  1        and
        LV_VAR_ICL = 1                 and
        LV_ERR_CAN_BOFF = 0
Then    LV_CDN_DIAG_TOUT_ICL_3 = 1
Else    LV_CDN_DIAG_TOUT_ICL_3 = 0
Endif
```

Formula section:

Calculation of time-out counter

```
If      no CAN message INSTR3 received
Then    CYC_CTR_ICL_3_DIAG $_{N}$   = CYC_CTR_ICL_3_DIAG $_{N-1}$ +1
Else    CYC_CTR_ICL_3_DIAG      = 0     (reset)
EndIf
```

Detection of error symptom

```
If(1)       CYC_CTR_ICL_3_DIAG = C_ICL_3_CYCNR_MAX
Then(1)     CYC_CTR_ICL_3_DIAG      = 0     (reset)
    If(2)       no CAN message INSTR3 is received
    Then(2)     ERR_SYM_TOUT_ICL_3 = SYM_2
                Error symptom is detected, ABC counter starts to increment
                LV_ERR_TOUT_ICL_3       = 1     (after debounce)
    Else(2)     ERR_SYM_TOUT_ICL_3 = NO_SYM
                no Error symptom is detected, ABC counter starts to decrement
                LV_ERR_TOUT_ICL_3       = 0     (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ICL_3_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing INSTR3 messages for error debounce counter increment					
C_ABC_INC_TOUT_ICL_3	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message INSTR3					
C_ABC_MAX_TOUT_ICL_3	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message INSTR3					

43.7.9 Timeout CAN message INSTR7

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_ICL_7	V/O	0...1H	0...1	1	[-]
Logical variable timeout error CAN message INSTR7					
CYC_CTR_ICL_7_DIAG	V	0...FFH	0...255	1	[-]
Counter of not received INSTR7 messages					
LV_CDN_DIAG_TOUT_ICL_7	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN message INSTR7					
ERR_SYM_TOUT_ICL_7	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CAN message INSTR7					
LV_END_DIAG_TOUT_ICL_7	V/O	0...1H	0...1	1	[-]
End of diagnosis CAN message INSTR7					

Input data:

LV_CDN_VB_CAN_DIAG	LV_VAR_ICL	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_ICL_7_DIAG is always incremented If the CAN message INSTR7 has not been received. The counter is set to 0 If a CAN message INSTR7 has been received or If the counter has reached his maximum value C_ICL_7_CYCNR_MAX.
If this counter has reached the calibratable threshold C_ICL_7_CYCNR_MAX and there is no CAN message INSTR7 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_ICL_7. If a CAN message INSTR7 is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_ICL_7 the timeout error CAN message INSTR7 (LV_ERR_TOUT_ICL_7) is set.

Error symptoms are defined for this diagnosis function as:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 |  |  |  └─> -                          (= SYM_0)
 |  |  └────> -                          (= SYM_1)
 |  └───────> Timeout signal INSTR7      (= SYM_2)
 └──────────> -                          (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 s

Activation: **If** LV_CDN_VB_CAN_DIAG = 1 **and**
LV_IGK = 1 **and**
LV_VAR_ICL = 1 **and**

```
               LV_ERR_CAN_BOFF = 0
          Then LV_CDN_DIAG_TOUT_ICL_7 = 1
          Else LV_CDN_DIAG_TOUT_ICL_7 = 0
          Endif
```

Formula section:

Calculation of time-out counter

```
If        no CAN message INSTR7 received
Then      CYC_CTR_ICL_7_DIAG $_{N}$ = CYC_CTR_ICL_7_DIAG $_{N-1}$ +1
Else      CYC_CTR_ICL_7_DIAG    = 0     (reset)
EndIf
```

Detection of error symptom

```
If(1)      CYC_CTR_ICL_7_DIAG = C_ICL_7_CYCNR_MAX
Then(1)    CYC_CTR_ICL_7_DIAG    = 0     (reset)
    If(2)      no CAN message INSTR7 is received
    Then(2)    ERR_SYM_TOUT_ICL_7 = SYM_2
               Error symptom is detected, ABC counter starts to increment
               LV_ERR_TOUT_ICL_7     = 1     (after debounce)
    Else(2)    ERR_SYM_TOUT_ICL_7 = NO_SYM
               no Error symptom is detected, ABC counter starts to decrement
               LV_ERR_TOUT_ICL_7     = 0     (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ICL_7_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing INSTR7 messages for error debounce counter increment					
C_ABC_INC_TOUT_ICL_7	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message INSTR7					
C_ABC_MAX_TOUT_ICL_7	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message INSTR7					

43.7.10 Timeout CAN message LWS1

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_PSTE_1	V/O	0...1H	0...1	1	[-]
logical variable timeout error CAN message LWS1					
CYC_CTR_PSTE_1_DIAG	V	0...FFH	0...255	1	[-]
counter of not received LWS1 messages					
LV_CDN_DIAG_TOUT_PSTE_1	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN message LWS1					
ERR_SYM_TOUT_PSTE_1	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CAN message LWS1					
LV_END_DIAG_TOUT_PSTE_1	V/O	0...1H	0...1	1	[-]
End of diagnosis CAN message LWS1					

Input data:

LV_CDN_VB_CAN_DIAG	LV_VAR_PSTE	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_PSTE_1_DIAG is always incremented If the CAN message LWS1 as not been received. The counter is set to 0 If a CAN message LWS1 has been received or If the counter has reached his maximum value C_PSTE_1_CYCNR_MAX.
If this counter has reached the calibratable threshold C_PSTE_1_CYCNR_MAX and there is no CAN message LWS1 then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_PSTE_1. If a CAN message LWS1 is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_PSTE_1 the timeout error CAN message LWS1 (LV_ERR_TOUT_PSTE_1) is set.

Error symptoms are defined for this diagnosis function as:

```
□□□□
│ │ │ └─> -                      (= SYM_0)
│ │ └───> -                      (= SYM_1)
│ └─────> Timeout signal LWS1    (= SYM_2)
└───────> -                      (= SYM_3)
```

Application conditions:

Initialisation:	all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory		
Recurrence:	10 ms		
Activation:	**If**	LV_CDN_VB_CAN_DIAG = 1	**and**
		LV_VAR_PSTE = 1	**and**
		LV_ERR_CAN_BOFF = 0	

```
            Then    LV_CDN_DIAG_TOUT_PSTE_1 = 1
            Else    LV_CDN_DIAG_TOUT_PSTE_1 = 0
            Endif
```

Formula section:

Calculation of time-out counter

```
If      no CAN message LWS1 received
Then    $CYC\_CTR\_PSTE\_1\_DIAG_{N} = CYC\_CTR\_PSTE\_1\_DIAG_{N-1} + 1$
Else    CYC_CTR_PSTE_1_DIAG = 0    (reset)
EndIf
```

Detection of error symptom

```
If(1)     CYC_CTR_PSTE_1_DIAG = C_PSTE_1_CYCNR_MAX
Then(1)   CYC_CTR_PSTE_1_DIAG = 0    (reset)
    If(2)     no CAN message LWS1 is received
    Then(2)   ERR_SYM_TOUT_PSTE_1 = SYM_2
              Error symptom is detected, ABC counter starts to increment
              LV_ERR_TOUT_PSTE_1 = 1    (after debounce)
    Else(2)   ERR_SYM_TOUT_PSTE_1 = NO_SYM
              no Error symptom is detected, ABC counter starts to decrement
              LV_ERR_TOUT_PSTE_1 = 0    (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_PSTE_1_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing LWS1 messages for error debounce counter increment					
C_ABC_INC_TOUT_PSTE_1	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message EGS2					
C_ABC_MAX_TOUT_PSTE_1	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message LWS1					

43.7.11 Timeout CAN message EKP

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_EFP_CAN	V/O	0...1H	0...1	1	[-]
logical variable timeout error CAN message EKP					

CYC_CTR_EFP_CAN_DIAG	V	0...FFH	0...255	1	[-]
counter of not received EKP messages					
LV_CDN_DIAG_TOUT_EFP_CAN	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN message EKP					
ERR_SYM_TOUT_EFP_CAN	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom CAN message EKP					
LV_END_DIAG_TOUT_EFP_CAN	V/O	0...1H	0...1	1	[-]
End of diagnosis CAN message EKP					

Input data:

LV_CDN_VB_CAN_DIAG	CONF_SWI_EFP_OUT	LV_ERR_CAN_BOFF	LV_IGK

Description:

The counter CYC_CTR_EFP_CAN_DIAG is always incremented If the CAN message EKP has not been received. The counter is set to 0 If a CAN message EKP has been received or If the counter has reached his maximum value C_EFP_CAN_CYCNR_MAX.
If this counter has reached the calibratable threshold C_EFP_CAN_CYCNR_MAX and there is no CAN message EKP then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_EFP_CAN. If a CAN message EKP is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_EFP_CAN the timeout error CAN message EKP (LV_ERR_TOUT_EFP_CAN) is set.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                          (= SYM_0)
 │ │ └───> -                          (= SYM_1)
 │ └─────> Timeout signal EKP         (= SYM_2)
 └───────> -                          (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 ms

Activation:

```
If    LV_CDN_VB_CAN_DIAG =  1        and
      CONF_SWI_EFP_OUT = 2           and
      LV_ERR_CAN_BOFF = 0
Then  LV_CDN_DIAG_TOUT_EFP_CAN = 1
Else  LV_CDN_DIAG_TOUT_EFP_CAN = 0
Endif
```

Formula section:

Calculation of time-out counter

```
If          no CAN message EKP received
Then        CYC_CTR_EFP_CAN_DIAG $_N$      = CYC_CTR_EFP_CAN_DIAG $_{N-1}$ +1
Else        CYC_CTR_EFP_CAN_DIAG           = 0     (reset)
EndIf
```

Detection of error symptom

```
If(1)       CYC_CTR_EFP_CAN_DIAG = C_EFP_CAN_CYCNR_MAX
Then(1)     CYC_CTR_EFP_CAN_DIAG         = 0     (reset)
    If(2)       no CAN message LWS1 is received
    Then(2)     ERR_SYM_TOUT_EFP_CAN = SYM_2
                Error symptom is detected, ABC counter starts to increment
                LV_ERR_TOUT_EFP_CAN  = 1     (after debounce)
    Else(2)     ERR_SYM_TOUT_EFP_CAN = NO_SYM
                no Error symptom is detected, ABC counter starts to decrement
                LV_ERR_TOUT_EFP_CAN  = 0     (after rebounce)
    EndIf(2)
EndIf(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EFP_CAN_CYCNR_MAX	1	1...FFH	1...255	1	[-]
Maximum number of missing LWS1 messages for error debounce counter increment					
C_ABC_INC_TOUT_EFP_CAN	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of timeout CAN message EGS2					
C_ABC_MAX_TOUT_EFP_CAN	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of timeout CAN message LWS1					

43.8 BN2000 diagnosis

43.8.1 Table of BN2000 diagnosis

FUNCTION DESCRIPTION:

General information:

This chapter is for the definition of all ERRM relevant output – data using the general BN2000 diagnosis algoritm.

The CAN diagnosis for BN input messages are only enabled if the variant has been recognized. In case of a modul (e.g. Kombi) which is not a variant but implemented in every vehicle the diagnosis is always enabled.

43.8.1.1 Diagnosis ACC module (Distance cruise control)

43.8.1.1.1 DREHMOMENT_ANF_ACC (LV_ERR_BN_TQ_DCC)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TQ_DCC	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TQ_DCC	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TQ_DCC	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TQ_DCC	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_TQ_DCC	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_TQ_DCC	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_TQ_DCC	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_DCC			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_DCC = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> -                                       (= SYM_0)
 │  │  └────> Alive counter did not count             (= SYM_1)
 │  └───────> Timeout                                 (= SYM_2)
 └──────────> Sent checksum ≠ calculated checksum     (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TQ_DCC	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TQ_DCC	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TQ_DCC	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_TQ_DCC	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message TQ_DCC					
C_CTR_DIAG_CKS_MAX_TQ_DCC	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message TQ_DCC					
C_CTR_DIAG_TOUT_MAX_TQ_DCC	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TQ_DCC					

43.8.1.2 Diagnosis ARS module

43.8.1.2.1 STAT_ARS (LV_ERR_BN_ARS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_ARS	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_ARS	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_ARS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_ARS	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_ARS	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_TOUT_ARS	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_ARS			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_ARS = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
☐☐☐☐
 │ │ │ └─> -                                    (= SYM_0)
 │ │ └───> Alive counter did not count          (= SYM_1)
 │ └─────> Timeout                              (= SYM_2)
 └───────> -                                    (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_ARS	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_ARS	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_ARS	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_ARS	1	0...FFH	0...255	1	[-]
Maximum number of wrong ATCC_CANe counrt events before error detection of message ARS					
C_CTR_DIAG_TOUT_MAX_ARS	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ARS					

43.8.1.3 Diagnosis AFS module (Active front steering)

43.8.1.3.1 DREHMOMENT_ANF_AFS (LV_ERR_BN_TQ_PSTE_2)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TQ_PSTE_2	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TQ_PSTE_2	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TQ_PSTE_2	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TQ_PSTE_2	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_TQ_PSTE_2	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_TQ_PSTE_2	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_TQ_PSTE_2	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_PSTE_2			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_PSTE_2 = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
□□□□
 | | | └─> -                                  (= SYM_0)
 | | └───> Alive counter did not count        (= SYM_1)
 | └─────> Timeout                            (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TQ_PSTE_2	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TQ_PSTE_2	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TQ_PSTE_2	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_TQ_PSTE_2	1	0...FFH	0...255	1	[-]
Maximum number of wrong ATCC_CANe counrt events before error detection of message TQ_PSTE_2					
C_CTR_DIAG_CKS_MAX_TQ_PSTE_2	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message TQ_PSTE_2					
C_CTR_DIAG_TOUT_MAX_TQ_PSTE_2	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TQ_PSTE_2					

43.8.1.4 Diagnosis EHB module (Electro-hydraulic brake)

43.8.1.4.1 DREHMOMENT_ANF_STE (LV_ERR_BN_TQ_PSTE_3)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TQ_PSTE_3	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TQ_PSTE_3	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TQ_PSTE_3	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TQ_PSTE_3	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_TQ_PSTE_3	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_TQ_PSTE_3	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_TQ_PSTE_3	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_PSTE_3			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_PSTE_3 = 1

Description:

Error symptoms are defined for this diagnosis function as:


```
│ │ │ └─> -                                  (= SYM_0)
│ │ └───> Alive counter did not count        (= SYM_1)
│ └─────> Timeout                            (= SYM_2)
└───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TQ_PSTE_3	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TQ_PSTE_3	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TQ_PSTE_3	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_TQ_PSTE_3	1	0...FFH	0...255	1	[-]
Maximum number of wrong ATCC_CANe counrt events before error detection of message TQ_PSTE_3					
C_CTR_DIAG_CKS_MAX_TQ_PSTE_3	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message TQ_PSTE_3					
C_CTR_DIAG_TOUT_MAX_TQ_PSTE_3	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TQ_PSTE_3					

43.8.1.5 Diagnosis EMF module (Electro-mechanical parke brake)

43.8.1.5.1 STELLANF_EMF (LV_ERR_BN_REQ_PBR)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_REQ_PBR	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_REQ_PBR	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_REQ_PBR	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_REQ_PBR	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_REQ_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_REQ_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_REQ_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

TCC_CAN_REQ_PBR	CKS_CAN_REQ_PBR	CKS_CLC_REQ_PBR	LV_VAR_PBR

General information:

See general BN2000 diagnosis algorithm

Application conditions:

```
Activation:  LV_VAR_PBR = 1
             //set LV_END in case of not learnt variant:
             if       LV_VAR_PBR = 0
             then     LV_END_DIAG_BN_REQ_PBR = 1
             endif
```

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                  (= SYM_0)
 │ │ └───> Alive counter did not count        (= SYM_1)
 │ └─────> Timeout                            (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_REQ_PBR	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_REQ_PBR	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_REQ_PBR	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_REQ_PBR	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counter events before error detection of message REQ_PBR					
C_CTR_DIAG_CKS_MAX_REQ_PBR	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message REQ_PBR					
C_CTR_DIAG_TOUT_MAX_REQ_PBR	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message REQ_PBR					

43.8.1.5.2 STATUS_EMF (LV_ERR_BN_PBR)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_DIAG_CKS_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TCC_CAN_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_TOUT_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					
ERR_SYM_BN_PBR	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_CDN_DIAG_BN_PBR	O/V	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
LV_END_DIAG_BN_PBR	O/V	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
LV_ERR_BN_PBR	O/V	0...1H	0...1	1	[-]
Present error BN2000					

Input data:

TCC_CAN_PBR	CKS_CAN_PBR	CKS_CLC_PBR	LV_VAR_PBR

General information:

See general BN2000 diagnosis algorithm

Application conditions:

```
Activation:   LV_VAR_PBR = 1
              //set LV_END in case of not learnt variant:
              if      LV_VAR_PBR = 0
              then    LV_END_DIAG_BN_PBR = 1
              endif
```

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                    (= SYM_0)
 │ │ └───> Alive counter did not count          (= SYM_1)
 │ └─────> Timeout                              (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum  (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_PBR	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_PBR	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_PBR	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_PBR	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counter events before error detection of message PBR					
C_CTR_DIAG_CKS_MAX_PBR	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message PBR					
C_CTR_DIAG_TOUT_MAX_PBR	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message PBR					

43.8.1.5.3 ST_RQ_EMF (LV_ERR_BN_TQ_PBR)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_DIAG_CKS_TQ_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TCC_CAN_TQ_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_TOUT_TQ_PBR	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					
ERR_SYM_BN_TQ_PBR	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_CDN_DIAG_BN_TQ_PBR	O/V	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
LV_END_DIAG_BN_TQ_PBR	O/V	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
LV_ERR_BN_TQ_PBR	O/V	0...1H	0...1	1	[-]
Present error BN2000					

Input data:

CKS_CAN_TQ_PBR	CKS_CLC_TQ_PBR	LV_VAR_TQ_PBR	TCC_CAN_TQ_PBR

General information:

See general BN2000 diagnosis algorithm

Application conditions:

```
Activation:  LV_VAR_TQ_PBR = 1
             //set LV_END in case of not learnt variant:
             if      LV_VAR_TQ_PBR = 0
             then    LV_END_DIAG_BN_TQ_PBR = 1
             endif
```

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                   (= SYM_0)
 │ │ └───> Alive counter did not count         (= SYM_1)
 │ └─────> Timeout                             (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_BN_TQ_PBR	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_ABC_INC_BN_TQ_PBR	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TQ_PBR	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_CTR_DIAG_CKS_MAX_TQ_PBR	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message ETCU_3					
C_CTR_DIAG_TCC_CAN_MAX_TQ_PBR	1	0...FFH	0...255	1	[-]
Maximum number of wrong alive counter events for error debounce counter increment					
C_CTR_DIAG_TOUT_MAX_TQ_PBR	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ACC					

43.8.1.6 Diagnosis CAS (Car-access-system)

43.8.1.6.1 Klemmenstatus (LV_ERR_BN_CAS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_CAS	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_CAS	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_CAS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_CAS	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_CAS	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_CAS	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_CAS	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: in every engine operating state

Description:

Error symptoms are defined for this diagnosis function as:

```
|_|_|_|_|
 | | | └─> -                                 (= SYM_0)
 | | └───> Alive counter did not count       (= SYM_1)
 | └─────> Timeout                           (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_CAS	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_CAS	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_CAS	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_CAS	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message CAS					
C_CTR_DIAG_CKS_MAX_CAS	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message CAS					
C_CTR_DIAG_TOUT_MAX_CAS	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message CAS					

43.8.1.6.2 Fahrzeugmodus (LV_ERR_BN_VEH_MOD)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_VEH_MOD	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_VEH_MOD	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_VEH_MOD	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_VEH_MOD	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_VEH_MOD	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_VEH_MOD	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_VEH_MOD	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_VEH_MOD			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_VEH_MOD = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 | | | └─> -                                        (= SYM_0)
```

```
 │  │  └───> Alive counter did not count           (= SYM_1)
 │  └─────> Timeout                                 (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum     (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_VEH_MOD	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_VEH_MOD	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_VEH_MOD	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_VEH_MOD	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message VEH_MOD					
C_CTR_DIAG_CKS_MAX_VEH_MOD	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message VEH_MOD					
C_CTR_DIAG_TOUT_MAX_VEH_MOD	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message VEH_MOD					

43.8.1.6.3 STAT_ZV_KLAPPEN (LV_ERR_BN_CDN_DOOR)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_CDN_DOOR	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_CDN_DOOR	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_CDN_DOOR	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_CDN_DOOR	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_CDN_DOOR	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: in every engine operating state

Description:

Error symptoms are defined for this diagnosis function as:

```
☐☐☐☐
 │  │  │  └─> -                                     (= SYM_0)
 │  │  └───> -                                     (= SYM_1)
 │  └─────> Timeout                                (= SYM_2)
```

```
└────────> -                                        (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_CDN_DOOR	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_CDN_DOOR	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_CDN_DOOR	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_CDN_DOOR	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message CDN_DOOR					

43.8.1.7 Diagnosis DSC - modul (Traction control system)

43.8.1.7.1 DREHMOMENT_ANF_DSC (LV_ERR_BN_TQ_TCS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TQ_TCS	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TQ_TCS	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TQ_TCS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TQ_TCS	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_TQ_TCS	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_TQ_TCS	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_TQ_TCS	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: in every engine operating state

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                        (= SYM_0)
 │ │ └───> Alive counter did not count              (= SYM_1)
 │ └─────> Timeout                                  (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum      (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TQ_TCS	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TQ_TCS	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TQ_TCS	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_TQ_TCS	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message TQ_TCS					
C_CTR_DIAG_CKS_MAX_TQ_TCS	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message TQ_TCS					
C_CTR_DIAG_TOUT_MAX_TQ_TCS	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TQ_TCS					

43.8.1.7.2 GESCHWINDIGKEIT (LV_ERR_BN_VS_TCS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_VS_TCS	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_VS_TCS	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_VS_TCS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_VS_TCS	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_VS_TCS	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: in every engine operating state

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                   (= SYM_0)
 │ │ └───> -                                   (= SYM_1)
 │ └─────> Timeout                             (= SYM_2)
 └───────> -                                   (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_VS_TCS	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_VS_TCS	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_VS_TCS	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_VS_TCS	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message VS_TCS					

43.8.1.7.3 GESCHWINDIGKEIT_RAD

Diagnosis not requested

43.8.1.7.4 STAT_DSC (LV_ERR_BN_TCS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TCS	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TCS	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TCS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TCS	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_TCS	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: in every engine operating state

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                          (= SYM_0)
 │ │ └───> -                          (= SYM_1)
 │ └─────> Timeout                    (= SYM_2)
 └───────> -                          (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TCS	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TCS	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TCS	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_TCS	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TCS					

43.8.1.8 Diagnosis EGS (Electronic transmission control unit)

43.8.1.8.1 DREHMOMENT_ANF_EGS (LV_ERR_BN_TQ_ETCU)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TQ_ETCU	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TQ_ETCU	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TQ_ETCU	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TQ_ETCU	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_TQ_ETCU	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_TQ_ETCU	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_TQ_ETCU	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_AT	C_T_DET_VAR_CAN		

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_AT = 1

```
If      LV_AT = 0
(AT not learnt if C_T_DET_VAR_CAN elapsed after LV_IGK 0 → 1)
Then    LV_END_DIAG_BN_TQ_ETCU = 1 (set for readiness)
Endif
```

Description:

Error symptoms are defined for this diagnosis function as:

```
□□□□
 │ │ │ └─> -                                  (= SYM_0)
 │ │ └───> Alive counter did not count        (= SYM_1)
 │ └─────> Timeout                            (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TQ_ETCU	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TQ_ETCU	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TQ_ETCU	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_TQ_ETCU	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message TQ_ETCU					
C_CTR_DIAG_CKS_MAX_TQ_ETCU	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message TQ_ETCU					
C_CTR_DIAG_TOUT_MAX_TQ_ETCU	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TQ_ETCU					

43.8.1.9 Diagnosis SSG module (Automated manual transmission)

43.8.1.9.1 DREHMOMENT_ANF_SSG (LV_ERR_BN_TQ_AMT)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TQ_AMT	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TQ_AMT	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TQ_AMT	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TQ_AMT	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_TQ_AMT	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_TQ_AMT	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_TQ_AMT	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_AMT	C_T_DET_VAR_CAN		

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_AMT = 1

If LV_VAR_AMT = 0

(AMT not learnt if C_T_DET_VAR_CAN elapsed after LV_IGK 0 → 1)

Then LV_END_DIAG_BN_TQ_AMT = 1 (set for readiness)

Endif

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                     (= SYM_0)
 │ │ └───> Alive counter did not count           (= SYM_1)
 │ └─────> Timeout                               (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum   (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TQ_AMT	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TQ_AMT	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TQ_AMT	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_TQ_AMT	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message TQ_AMT					
C_CTR_DIAG_CKS_MAX_TQ_AMT	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message TQ_AMT					
C_CTR_DIAG_TOUT_MAX_TQ_AMT	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TQ_AMT					

43.8.1.10 Diagnosis DKG module (Twin clutch transmission)

43.8.1.10.1 DREHMOMENT_ANF_DKG (LV_ERR_BN_TQ_TCT)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TQ_TCT	O/V	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TQ_TCT	O/V	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TQ_TCT	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TQ_TCT	O/V	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_TQ_TCT	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_TQ_TCT	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_TQ_TCT	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_TCT	C_T_DET_VAR_CAN		

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_TCT = 1

If LV_VAR_TCT = 0

(TCT not learnt if C_T_DET_VAR_CAN elapsed after LV_IGK 0 → 1)

Then LV_END_DIAG_BN_TQ_TCT = 1 (set for readiness)

Endif

Description:

Error symptoms are defined for this diagnosis function as:

```
|_|_|_|_|
 | | | └─> -                                  (= SYM_0)
 | | └───> Alive counter did not count        (= SYM_1)
 | └─────> Timeout                            (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TQ_TCT	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TQ_TCT	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TQ_TCT	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_TQ_TCT	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message TQ_TCT					
C_CTR_DIAG_CKS_MAX_TQ_TCT	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message TQ_TCT					
C_CTR_DIAG_TOUT_MAX_TQ_TCT	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TQ_TCT					

43.8.1.10.2 ST_DKG (LV_ERR_BN_STAT_TCT)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_STAT_TCT	O/V	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_STAT_TCT	O/V	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_STAT_TCT	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_STAT_TCT	O/V	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_STAT_TCT	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_TCT	C_T_DET_VAR_CAN		

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_TCT = 1

If LV_VAR_TCT = 0

(TCT not learnt if C_T_DET_VAR_CAN elapsed after LV_IGK 0 → 1)

Then LV_END_DIAG_BN_STAT_TCT = 1 (set for readiness)

Endif

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                (= SYM_0)
 │ │ └───> -                                (= SYM_1)
 │ └─────> Timeout                          (= SYM_2)
 └───────> -                                (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_STAT_TCT	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_STAT_TCT	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_STAT_TCT	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_STAT_TCT	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ACC					

43.8.1.11 Transmission data (from transmisssion units)

43.8.1.11.1 GETRIEBEDATEN (LV_ERR_BN_ETCU)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_ETCU	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_ETCU	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_ETCU	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_ETCU	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_ETCU	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					
CTR_DIAG_TCC_CAN_ETCU	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_ETCU	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					

Input data:

LV_VAR_ETCU			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

```
Activation:   LV_VAR_ETCU = 1
              If      LV_VAR_ETCU = 0
              Then    LV_END_DIAG_BN_ETCU = 1       (set for readiness)
              Endif
```

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                   (= SYM_0)
 │ │ └───> Alive counter did not count         (= SYM_1)
 │ └─────> Timeout                             (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_ETCU	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_ETCU	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_ETCU	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_ETCU	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ETCU					
C_CTR_DIAG_TCC_CAN_MAX_ETCU	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message ETCU_3					
C_CTR_DIAG_CKS_MAX_ETCU	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message ETCU					

43.8.1.11.2 GETRIEBEDATEN_2 (LV_ERR_BN_ETCU_2)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_ETCU_2	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_ETCU_2	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_ETCU_2	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_ETCU_2	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_ETCU_2	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_AT	LV_VAR_TCT		

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_AT = 1 **or** LV_VAR_TCT = 1

If LV_AT = 0 **and** LV_VAR_TCT = 0

Then LV_END_DIAG_BN_ETCU_2 = 1 (set for readiness)

Endif

Description:

Error symptoms are defined for this diagnosis function as:

```
  └──┴──┴──┴──┘
   │  │  │  └─> -                                    (= SYM_0)
   │  │  └────> -                                    (= SYM_1)
   │  └───────> Timeout                              (= SYM_2)
   └──────────> -                                    (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_ETCU_2	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_ETCU_2	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_ETCU_2	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_ETCU_2	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ETCU_2					

43.8.1.11.3 GETRIEBEDATEN_3 (LV_ERR_BN_ETCU_3)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_ETCU_3	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_ETCU_3	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_ETCU_3	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_ETCU_3	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_ETCU_3	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_ETCU_3	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_ETCU_3	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_AT	CKS_CLC_ETCU_3	TCC_CAN_ETCU_3	CKS_CAN_ETCU_3
LV_VAR_ETCU_3			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

```
Activation:   LV_VAR_ETCU_3 = 1
              //set LV_END in case of not learnt variant:
              if        LV_VAR_ETCU_3 = 0
              then      LV_END_DIAG_BN_ETCU_3 = 1
              endif
```

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                 (= SYM_0)
 │ │ └───> Alive counter did not count       (= SYM_1)
 │ └─────> Timeout                           (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_ETCU_3	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_ETCU_3	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_ETCU_3	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_ETCU_3	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message ETCU_3					
C_CTR_DIAG_CKS_MAX_ETCU_3	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message ETCU_3					
C_CTR_DIAG_TOUT_MAX_ETCU_3	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ETCU_3					

43.8.1.11.4 ANZEIGE_GETRIEBEDATEN (LV_ERR_BN_ETCU_DISP)

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>LV_ERR_BN_ETCU_DISP</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Present error BN2000</td></tr>
<tr><td>LV_CDN_DIAG_BN_ETCU_DISP</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Diagnosis condition BN2000 diagnosis</td></tr>
<tr><td>ERR_SYM_BN_ETCU_DISP</td><td>V/O</td><td>0H
1H
2H
4H
8H</td><td>NO_SYM
SYM_0
SYM_1
SYM_2
SYM_3</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Detected symptom BN2000 diagnosis</td></tr>
<tr><td>LV_END_DIAG_BN_ETCU_DISP</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">End of diagnosis BN2000 diagnosis</td></tr>
<tr><td>CTR_DIAG_TOUT_ETCU_DISP</td><td>V</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Diagnosis counter for timeout counter error symptom detection</td></tr>
</table>

Input data:

LV_AT	LV_VAR_TCT		

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_AT = 1 **or** LV_VAR_TCT = 1

```
If      LV_AT = 0  and LV_VAR_TCT = 0
Then    LV_END_DIAG_BN_ETCU_DISP = 1        (set for readiness)
Endif
```

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                          (= SYM_0)
 │ │ └───> -                          (= SYM_1)
 │ └─────> Timeout                    (= SYM_2)
 └───────> -                          (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_ETCU_DISP	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_ETCU_DISP	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_ETCU_DISP	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_ETCU_DISP	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ETCU_DISP					

43.8.1.12 Diagnosis IHKA

43.8.1.12.1 WÄRMASTROM_KLIMA (LV_ERR_BN_ACC)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_ACC	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_ACC	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_ACC	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_ACC	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_ACC	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_ACIN			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_ACIN = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                        (= SYM_0)
 │ │ └───> -                        (= SYM_1)
 │ └─────> Timeout                  (= SYM_2)
 └───────> -                        (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_ACC	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_ACC	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_ACC	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_ACC	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ACC					

43.8.1.13 Diagnosis KOMBI

43.8.1.13.1 A_TEMP_RELATIVZEIT (LV_ERR_BN_T_ICL)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_T_ICL	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_T_ICL	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_T_ICL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_T_ICL	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_T_ICL	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_ICL			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_ICL = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
│ │ │ │ │
└─┴─┴─┴─┘
 │ │ │ └─> -                          (= SYM_0)
 │ │ └───> -                          (= SYM_1)
 │ └─────> Timeout                    (= SYM_2)
 └───────> -                          (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_T_ICL	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_T_ICL	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_T_ICL	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_T_ICL	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ICL					

43.8.1.13.2 KILOMETERSTAND (LV_ERR_BN_KM_ICL)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_KM_ICL	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_KM_ICL	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_KM_ICL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_KM_ICL	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_KM_ICL	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_ICL			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_ICL = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
|_|_|_|_|
 | | | └─> -                       (= SYM_0)
 | | └───> -                       (= SYM_1)
 | └─────> Timeout                 (= SYM_2)
 └───────> -                       (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_KM_ICL	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_KM_ICL	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_KM_ICL	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_KM_ICL	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message KM_ICL					

43.8.1.13.3 STAT_KOMBI (LV_ERR_BN_ICL)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_ICL	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_ICL	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_ICL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_ICL	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_ICL	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_TOUT_ICL	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_ICL			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_ICL = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 │  │  │  └─> -                              (= SYM_0)
 │  │  └────> Alive counter did not count    (= SYM_1)
 │  └───────> Timeout                        (= SYM_2)
```

```
└────────> -                                    (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_ICL	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_ICL	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_ICL	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_ICL	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message ICL					
C_CTR_DIAG_TOUT_MAX_ICL	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ICL					

43.8.1.13.4 UHRZEIT_DATUM (LV_ERR_BN_T_CLK)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_T_CLK	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_T_CLK	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_T_CLK	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_T_CLK	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_T_CLK	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_TOUT_T_CLK	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_ICL			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_ICL = 1

Description:

Error symptoms are defined for this diagnosis function as:

☐☐☐☐

```
 │  │  │  └─> -                                          (= SYM_0)
 │  │  └────> -                                          (= SYM_1)
 │  └───────> Timeout                                    (= SYM_2)
 └──────────> -                                          (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_T_CLK	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_T_CLK	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_T_CLK	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_T_CLK	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message T_CLK					
C_CTR_DIAG_TOUT_MAX_T_CLK	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message T_CLK					

43.8.1.14 Diagnosis Power modul

43.8.1.14.1 POWERMGMT_BATTERIESPANNUNG (LV_ERR_BN_POW_VB)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_POW_VB	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_POW_VB	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_POW_VB	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_POW_VB	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_POW_VB	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_VEH			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_VEH = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                   (= SYM_0)
 │ │ └───> -                                   (= SYM_1)
 │ └─────> Timeout                             (= SYM_2)
 └───────> -                                   (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_POW_VB	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_POW_VB	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_POW_VB	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_POW_VB	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message POW_VB					

43.8.1.14.2 POWERMGMT_LADESPANNUNG (LV_ERR_BN_POW_GEN)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_POW_GEN	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_POW_GEN	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_POW_GEN	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_POW_GEN	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_POW_GEN	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_VEH			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_VEH = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                   (= SYM_0)
 │ │ └───> -                                   (= SYM_1)
```

```
 |  └————> Timeout                                   (= SYM_2)
 └—————> -                                           (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_POW_GEN	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_POW_GEN	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_POW_GEN	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_POW_GEN	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message POW_GEN					

43.8.1.15 Diagnosis SZ Lenksäule

43.8.1.15.1 BEDIENUNG_TEMPOMAT (LV_ERR_BN_MSW)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_MSW	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_MSW	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_MSW	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_MSW	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_MSW	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_MSW	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_MSW	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_BN_MSW			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_BN_MSW = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
```

```
└─┴─┴─┴─┘
 │ │ │ └─> -                                    (= SYM_0)
 │ │ └───> Alive counter did not count          (= SYM_1)
 │ └─────> Timeout                              (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum  (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_MSW	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_MSW	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_MSW	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_MSW	1	0...FFH	0...255	1	[-]
Maximum number of wrong TCC_CAN counrt events before error detection of message MSW					
C_CTR_DIAG_CKS_MAX_MSW	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message MSW					
C_CTR_DIAG_TOUT_MAX_MSW	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message MSW					

43.8.1.15.2 LENKRADWINKEL (LV_ERR_BN_ANG_PSTE)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_ANG_PSTE	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_ANG_PSTE	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_ANG_PSTE	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_ANG_PSTE	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_ANG_PSTE	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_PSTE			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_PSTE = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
```

```
└─┴─┴─┴─┘
  │  │  │  └─> -                                  (= SYM_0)
  │  │  └────> -                                  (= SYM_1)
  │  └───────> Timeout                            (= SYM_2)
  └──────────> -                                  (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_ANG_PSTE	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_ANG_PSTE	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_ANG_PSTE	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_ANG_PSTE	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message ANG_PSTE					

43.8.1.16 Diagnosis EKP

43.8.1.16.1 STAT_EKP(LV_ERR_BN_EFP)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_EFP	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_EFP	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_EFP	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_EFP	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_EFP	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_BN_EFP			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_BN_EFP = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
```

```
 │  │  │  └─> -                                        (= SYM_0)
 │  │  └───> -                                        (= SYM_1)
 │  └─────> Timeout                                  (= SYM_2)
 └───────> -                                        (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_EFP	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_EFP	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_EFP	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_EFP	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message EFP					

43.8.1.17 Diagnosis K_CAN: LM

43.8.1.17.1 STAT_GANG_RUECKWAERTS (LV_ERR_BN_GEAR_REV)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_GEAR_REV	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_GEAR_REV	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_GEAR_REV	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_GEAR_REV	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_GEAR_REV	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_BN_GEAR_REV			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_BN_GEAR_REV = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
☐☐☐☐
 │  │  │  └─> -                                        (= SYM_0)
```

```
 │  │  └───> -                                       (= SYM_1)
 │  └─────> Timeout                                 (= SYM_2)
 └───────> -                                        (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_GEAR_REV	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_GEAR_REV	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_GEAR_REV	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_GEAR_REV	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message GEAR_REV					

43.8.1.17.2 LAMPENZUSTAND (LV_ERR_BN_LTG_HDLP_L)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_LTG_HDLP_L	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_LTG_HDLP_L	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_LTG_HDLP_L	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_LTG_HDLP_L	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_LTG_HDLP_L	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_BN_LTG_HDLP_L			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_BN_LTG_HDLP_L = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
☐☐☐☐
 │  │  │  └─> -                                     (= SYM_0)
 │  │  └───> -                                      (= SYM_1)
 │  └─────> Timeout                                 (= SYM_2)
```

```
 └──────────> -                                  (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_LTG_HDLP_L	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_LTG_HDLP_L	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_LTG_HDLP_L	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_LTG_HDLP_L	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message LTG_HDLP_L					

43.8.1.17.3 STAT_ANHAENGER (LV_ERR_BN_TRL)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_TRL	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_TRL	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_TRL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_TRL	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_TRL	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_BN_TRL			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_BN_TRL = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 │  │  │  └──> -                                  (= SYM_0)
 │  │  └─────> -                                  (= SYM_1)
 │  └────────> Timeout                            (= SYM_2)
 └──────────> -                                  (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_TRL	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_TRL	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_TRL	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_TRL	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message TRL					

43.8.1.18 Diagnosis LDM

43.8.1.18.1 ANFORDERUNG_RADM_ANTRIEBSTRANG (LV_ERR_BN_LDM)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_LDM	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_LDM	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_LDM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_LDM	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TCC_CAN_LDM	V	0...FFH	0...255	1	[-]
Diagnosis counter for alive-counter error symptom detection					
CTR_DIAG_CKS_LDM	V	0...FFH	0...255	1	[-]
Diagnosis counter for checksum counter error symptom detection					
CTR_DIAG_TOUT_LDM	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_BN_LDM			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_BN_LDM = 1

Description:

Error symptoms are defined for this diagnosis function as:

☐☐☐☐

| | | └> - (= SYM_0)

```
 │  │  └───> -                                           (= SYM_1)
 │  └─────> Timeout                                      (= SYM_2)
 └───────> -                                             (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_LDM	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_LDM	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_LDM	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TCC_CAN_MAX_LDM	1	0...FFH	0...255	1	[-]
Maximum number of wrong LDM_CAN counrt events before error detection of message LDM					
C_CTR_DIAG_CKS_MAX_LDM	1	0...FFH	0...255	1	[-]
Maximum number of wrong Checksum calculation events before error detection of message LDM					
C_CTR_DIAG_TOUT_MAX_LDM	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message LDM					

43.8.1.19 Diagnosis EKP-Crashabschaltung

43.8.1.19.1 Steuerung EKP-Crashabschaltung (LV_ERR_BN_EFP_CRASH)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_EFP_CRASH	V/O	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_EFP_CRASH	V/O	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_EFP_CRASH	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_EFP_CRASH	V/O	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_EFP_CRASH	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_EFP_CRASH			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

Activation: LV_VAR_EFP_CRASH = 1

Description:

Error symptoms are defined for this diagnosis function as:

```
 ┌──┬──┬──┬──┐
```

```
 └─┴─┴─┴─┘
  │  │  │  └─> -                              (= SYM_0)
  │  │  └────> -                              (= SYM_1)
  │  └───────> Timeout                        (= SYM_2)
  └──────────> -                              (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_EFP_CRASH	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_EFP_CRASH	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_EFP_CRASH	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_EFP_CRASH	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message EFP_CRASH					

43.8.1.20 Diagnosis DXC

43.8.1.20.1 SOLL_MOM_ANF (LV_ERR_BN_DHL_CTL)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BN_DHL_CTL	O/V	0...1H	0...1	1	[-]
Present error BN2000					
LV_CDN_DIAG_BN_DHL_CTL	O/V	0...1H	0...1	1	[-]
Diagnosis condition BN2000 diagnosis					
ERR_SYM_BN_DHL_CTL	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BN2000 diagnosis					
LV_END_DIAG_BN_DHL_CTL	O/V	0...1H	0...1	1	[-]
End of diagnosis BN2000 diagnosis					
CTR_DIAG_TOUT_DHL_CTL	V	0...FFH	0...255	1	[-]
Diagnosis counter for timeout counter error symptom detection					

Input data:

LV_VAR_4WD			

General information:

See general BN2000 diagnosis algorithm

Application conditions:

```
Activation:  LV_VAR_4WD = 1
             If    LV_VAR_4WD = 0
             Then  LV_END_DIAG_BN_DHL_CTL = 1  (set for readiness)
             Endif
```

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
 │ │ │ └─> -                                  (= SYM_0)
 │ │ └───> -                                  (= SYM_1)
 │ └─────> Timeout                            (= SYM_2)
 └───────> -                                  (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BN_DHL_CTL	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment BN2000 diagnosis					
C_ABC_MAX_BN_DHL_CTL	1	1...FFH	1...255	1	[-]
Anti-bounce counter maximum BN2000 diagnosis					
C_ABC_DEC_BN_DHL_CTL	1	0...FFH	0...255	1	[-]
Anti-bounce counter decrement BN2000 diagnosis					
C_CTR_DIAG_TOUT_MAX_DHL_CTL	1	0...FFH	0...255	1	[-]
Maximum number of missing messeges events before error detection of message DHL_CTL					

D

43.8.2 BN2000 diagnosis algorithm

FUNCTION DESCRIPTION:

General information:

All diagnosis are only activated if the variant BN2000 is valid (LV_VAR_BN = 1).
For the diagnosis each message is monitored.

XXX is the variable for each different input message (see “CAN messages BN2000, Enable condition for BN2000 diagnosis).

Description:

The following symptoms can be detected:

- Alive counter did not count
- Timeout of the message
- Sent checksum is not equal to the calculated checksum of the message

For detailed information of Alive-counter, checksum calculation and timeout see (see “CAN messages BN2000).

If symptom “Timeout” is detected (counter CTR_DIAG_TOUT_XXX ≠ 0), the remaining symptom detection is inhibited.

If C_CTR_DIAG_TOUT/CKS/TCC_CAN_MAX_XXX are calibrated to 0 then the diagnosis of the symptom TOUT/CKS/TCC_CAN is inhibited.

Error symptoms are defined for this diagnosis function as:

```
□□□□
 │ │ │ └─> -                                   (= SYM_0)
 │ │ └───> Alive counter did not count         (= SYM_1)
 │ └─────> Timeout                             (= SYM_2)
 └───────> Sent checksum ≠ calculated checksum (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: each message is diagnosed

Activation:

```
If        Diagnosis is enabled       (learnt/calibrated etc..., see "CAN messages
                                     BN2000, enable condition for BN2000 diagnosis).
Then If          LV_WAKE_UP                = 1          and
                 LV_CDN_VB_BN_DIAG         = 1          and
                 LV_ERR_CAN_BOFF           = 0          and
                 LV_VAR_BN                 = 1          and
     Then        LV_CDN_DIAG_BN_XXX        = 1
     Else        LV_CDN_DIAG_BN_XXX        = 0
     Endif
Else      LV_CDN_DIAG_BN_XXX    = 0
Endif
```

Deactivation:

```
If   XXX = TQ_DCC                    (Diagnosis of TQ_DCC message)
Then      If     LV_ERR_BN_XXX             = 1          and
                 ERR_SYM_BN_XXX            = SYM_1/SYM_3
          Then   LV_CDN_DIAG_BN_XXX        = 0    (for this DC!)
          Endif

Endif
```

Formula section:

Error detection alive counter:

```
If(1)       CTR_DIAG_TOUT_XXX = 0              and
            CTR_DIAG_CKS_xxx = 0               and
            C_CTR_DIAG_TCC_CAN_MAX_XXX ≠ 0
Then If(2)  $TCC\_CAN\_XXX_{n} = TCC\_CAN\_XXX_{n-1}$        or
                TCC_CAN_XXX = FH
    Then(2)     $CTR\_DIAG\_TCC\_CAN\_XXX_{n} = CTR\_DIAG\_TCC\_CAN\_XXX_{n-1} + 1$
        If(3)   CTR_DIAG_TCC_CAN_XXX >= C_CTR_DIAG_TCC_CAN_MAX_XXX
        Then(3)     Error symptom “Alive-counter” is active
                    ERR_SYM_BN_XXX = SYM_1
                    anti-bounce increment per C_ABC_INC_BN_XXX
                    LV_ERR_BN_XXX = 1          after debounce
        Endif(3)
    Else(2)     CTR_DIAG_TCC_CAN_XXX = 0
                Error symptom “Alive-counter” is passive
                ERR_SYM_BN_XXX = NO_SYM
                anti-bounce decrement per C_ABC_DEC_BN_XXX
                LV_ERR_BN_XXX = 0          after rebounce
    Endif(2)
Else(1)     No error detection
Endif(1)
```

Error detection “Checksum is wrong”:

```
If(1)       CTR_DIAG_TOUT_XXX = 0              and
            C_CTR_DIAG_CKS_MAX_XXX ≠ 0
Then  If(2)     CKS_CAN_XXX ≠ CKS_CLC_XXX  (sent ≠ calculated)
    Then(2)     $CTR\_DIAG\_CKS\_XXX_{n} = CTR\_DIAG\_CKS\_XXX_{n-1} + 1$
        If(3)       CTR_DIAG_CKS_XXX >= C_CTR_DIAG_CKS_MAX_XXX
        Then(3)     Error symptom “Calculated checksum ≠ sent checksum” is active
                    ERR_SYM_BN_XXX = SYM_3
                    anti-bounce increment per C_ABC_INC_BN_XXX
                    LV_ERR_BN_XXX = 1          after debounce
        Endif(3)
    Else(2)     CTR_DIAG_CKS_XXX = 0
                Error symptom “Calculated checksum ≠ sent checksum” is passive,
                ERR_SYM_BN_XXX = NO_SYM
```

```
                    anti-bounce decrement per C_ABC_DEC_BN_XXX
                    LV_ERR_BN_XXX = 0       after rebounce
        Endif(2)
Else(1)     No error detection
Endif(1)
```

Error detection “Timeout”:

```
If(1)       C_CTR_DIAG_TOUT_MAX_XXX ≠ 0
ThenIf(2)           no BN2000 message XXX is received
    Then(2)         CTR_DIAG_TOUT_XXX $_n$ = CTR_DIAG_TOUT_XXX $_{n-1}$ + 1
        If(3)           CTR_DIAG_TOUT_XXX >= C_CTR_DIAG_TOUT_MAX_XXX
        Then(3)         Error symptom “Timeout” is active
                        ERR_SYM_BN_XXX = SYM_2
                        anti-bounce increment per C_ABC_INC_BN_XXX
                        LV_ERR_BN_XXX = 1       after debounce
                        Else(3)
        Endif(3)
    Else(2)         CTR_DIAG_TOUT_XXX = 0
                    Error symptom “Timeout” is passive,
                    ERR_SYM_BN_XXX = NO_SYM
                    anti-bounce decrement per C_ABC_DEC_BN_XXX
                    LV_ERR_BN_XXX = 0       after rebounce
    Endif(2)
Else(1)     No error detection
Endif(1)
```

Calculation of the end of diagnosis:

For calculation of LV_END_DIAG_BN_XXX see “Anti-bounce algorithm, calculation of the end of diagnosis

Configuration of Anti-bounce

NC_ABC_CONF_FCT_DIAG_XX = 19H (Not saved, INI at LV_IGK, CTR_ABC_.. = 0 at INI, Calibratable decrement)

43.9 CAN diagnosis for NOX control (local CAN)

FUNCTION DESCRIPTION

General information:

The following CAN diagnosis are defined:

- Timeout CAN messages NOX

43.9.1 Timeout CAN messages NOX

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TOUT_NOX_SENS_i	V/O	0 ... 1H	0 ... 1	1	-
logical variable timeout error CAN message NOX_SENS_i (NOX_SENS – sensor bank i)					
LV_CDN_DIAG_TOUT_ NOX_SENS_i	V/O	0 ... 1H	0 ... 1	1	-
Diagnosis condition NOX_SENS_i					
ERR_SYM_TOUT_ NOX_SENS_i	V/O	0H 4H	NO_SYM SYM_3	1	-
Detected Symptom NOX_SENS_i					
LV_END_DIAG_TOUT_ NOX_SENS_i	V/O	0 ... 1H	0 ... 1	1	-
End of diagnosis NOX_SENS_i					
CYC_CTR_NOX_SENS_i_DIAG	V	0 ... FFH	0 ... 255	1	-
counter of not received NOX_SENS_i messages					

Input data:

LV_CDN_VB_CAN_DIAG	LV_IGK	LV_ERR_LOCAN_BOFF	

Description:

The counter CYC_CTR_NOX_SENS_i_DIAG are always incremented if the CAN messages NOX_SENS_i have not been received. The counter are set to 0 if a CAN message NOX_SENS_i has been received or if the counter has reached his maximum value C_NOX_SENS_CYCNR_MAX.
If these counters have reached the calibratable threshold C_NOX_SENS_CYCNR_MAX and there is no CAN message NOX_SENS_i then the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TOUT_NOX_SENS. If a CAN message NOX_SENS_i is received, the anti bounce counter is decremented. If the anti bounce counter has reached his maximum value C_ABC_MAX_TOUT_NOX_SENS the timeout error CAN message NOX_SENS (LV_ERR_TOUT_NOX_SENS_i) is set.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                         (= SYM_0)
 │ │ └───> -                         (= SYM_1)
 │ └─────> Timeout signal NOX_SENS_i (= SYM_2)
 └───────> -                         (= SYM_3)
```

Application conditions:

Initialisation: all 0 at Transition LV_IGK 0->1 or reset or at clearing error memory

Recurrence: 10 ms

Activation:

```
If      LV_CDN_VB_CAN_DIAG = 1        and
NOX-message has been received via LoCAN at least for once     and
LV_ERR_LOCAN_BOFF = 0
Then    LV_CDN_DIAG_TOUT_NOX_SENS_i = 1
Else    LV_CDN_DIAG_TOUT_NOX_SENS_i = 0
Endif
```

Formula section:

Calculation of time-out counter

```
If      no CAN message NOX_SENS_i received
Then    $CYC\_CTR\_NOX\_SENS\_i\_DIAG_{N} = CYC\_CTR\_NOX\_SENS\_i\_DIAG_{N-1} + 1$
Else    CYC_CTR_NOX_SENS_i_DIAG = 0     (reset)
EndIf
```

Detection of error symptom

```
If(1)       CYC_CTR_NOX_SENS_i_DIAG = C_NOX_SENS_CYCNR_MAX
Then(1)     CYC_CTR_NOX_SENS_i_DIAG = 0     (reset)
    If(2)       no CAN message NOX_SENS_i is received
    Then(2)     ERR_SYM_TOUT_NOX_SENS_i = SYM_2          Timeout is detected
                ABC counter starts to increment
                LV_ERR_TOUT_NOX_SENS_i = 1               (after debounce)
    Else(2)     ERR_SYM_TOUT_NOX_SENS_i = NO_SYM
                ABC counter starts to decrement
                LV_ERR_TOUT_NOX_SENS_i = 0               (after rebounce)
    EndIf(2)
EndIf(1)
```

Calculation end of diagnosis

LV_END_DIAG_TOUT_NOX_SENS_i is calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NOX_SENS_CYCNR_MAX	1	1 ... FFH	1 ... 255	1	-
maximum number of missing NOX_SENS messages for error debounce counter increment					
C_ABC_INC_TOUT_NOX_SENS	1	0 ... FFH	0 ... 255	1	-
anti bounce counter increment - diagnosis of timeout CAN message NOX_SENS					
C_ABC_MAX_TOUT_NOX_SENS	1	1 ... FFH	1 ... 255	1	-
anti bounce counter maximum value - diagnosis of timeout CAN message NOX_SENS					

43.10 CAN bus off diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_LOCAN_BOFF	V/O	0...1H	0...1	1	[-]
Present error LOCAN bus off					
LV_CDN_DIAG_LOCAN_BOFF	V/O	0...1H	0...1	1	[-]
Diagnosis condition LOCAN Boff diagnosis					
ERR_SYM_LOCAN_BOFF	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom Boff diagnosis LOCAN					
LV_END_DIAG_LOCAN_BOFF	V/O	0...1H	0...1	1	[-]
End of Diagnosis Boff diagnosis LOCAN					
LV_ERR_CAN_BOFF	V/O	0...1H	0...1	1	[-]
Present error CAN bus off					
LV_CDN_DIAG_CAN_BOFF	V/O	0...1H	0...1	1	[-]
Diagnosis condition CAN_BOFF					
ERR_SYM_CAN_BOFF	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom CAN_BOFF					
LV_END_DIAG_CAN_BOFF	V/O	0...1H	0...1	1	[-]
End of Diagnosis CAN_BOFF					

Input data:

LV_CDN_VB_CAN_DIAG			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_MAX_CAN_BOFF	1	1...FFH	1...255	1	[-]
Anti-Bounce counter maximum for CAN diagnosis "CAN Bus off detected by CAN"					
C_ABC_INC_CAN_BOFF	1	0...FFH	0...255	1	[-]
Anti-Bounce counter increment for CAN diagnosis "Can Bus off detected by CAN"					
C_ABC_INC_LOCAN_BOFF	1	0...FFH	0...255	1	[-]
Anti bounce counter increment - diagnosis of LOCAN bus on / off					
C_ABC_MAX_LOCAN_BOFF	1	1...FFH	1...255	1	[-]
Anti bounce counter maximum value - diagnosis of LOCAN bus on / off					

43.10.1 PT-CAN bus off diagnosis

FUNCTION DESCRIPTION

General information:

This diagnosis is working for BN2000 and CAN11H.

Description:

From CPU- CAN- controller a message is sent if CAN bus is on or off (internal). This diagnosis takes place every 10 ms at all engine operating states with ignition key on.
If CAN bus off is active the anti bounce counter is incremented by C_ABC_INC_CAN_BOFF.
If the anti bounce counter reaches the calibratable threshold C_ABC_MAX_CAN_BOFF then the error bit LV_ERR_CAN_BOFF is set to 1.

Error symptoms are defined for this diagnosis function as:

```
|_|_|_|_|
 |  |  |  └─> -                                     (= SYM_0)
 |  |  └───> -                                      (= SYM_1)
 |  └─────> CAN bus off          controller                (= SYM_2)
 └───────> -                                        (= SYM_3)
```

Application conditions:

Initialisation: STD_INI

Recurrence: 10 ms

```
Activation:   If     LV_CDN_VB_CAN_DIAG = 1
              Then   LV_CDN_DIAG_CAN_BOFF = 1
              Else   LV_CDN_DIAG_CAN_BOFF = 0
              Endif
```

Formula section:

Detection of error symptom

```
If      CAN bus off = active
Then    ERR_SYM_CAN_BOFF = SYM_2
        LV_ERR_CAN_BOFF = 1     (after debounce)
Else    ERR_SYM_ CAN_BOFF = NO_SYM
        LV_ERR_CAN_BOFF = 0     (after rebounce)
Endif
```

Calculation end of diagnosis

LV_END_DIAG_CAN_BOFF is calculated by error management

43.10.2 LOCAN bus off diagnosis

FUNCTION DESCRIPTION

General information:

This diagnosis is working for LOCAN applications like EFP – control or NOX - sensors.

Description:

From CPU hardware a message is sent if CAN bus is on or off (internal). This diagnosis takes place every 10 ms at all engine operating states with ignition key on.
If CAN bus off is active the anti bounce counter is incremented by C_ABC_INC_LOCAN_BOFF. If the anti bounce counter reaches the calibratable threshold C_ABC_MAX_LOCAN_BOFF then the error bit LV_ERR_LOCAN_BOFF is set to 1.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                        (= SYM_0)
 │ │ └───> -                        (= SYM_1)
 │ └─────> LOCAN bus off            (= SYM_2)
 └───────> LOCAN TxWARN or RxWARN   (= SYM_3)
```

Application conditions:

Initialisation: STD_INI

Recurrence: 10 ms

Activation:

```
If    LV_CDN_VB_CAN_DIAG = 1
Then  LV_CDN_DIAG_LOCAN_BOFF = 1
Else  LV_CDN_DIAG_LOCAN_BOFF = 0
Endif
```

Formula section:

Detection of error symptom

```
If      LOCAN bus off = active
Then    ERR_SYM_LOCAN_BOFF = SYM_2
        LV_ERR_LOCAN_BOFF = 1          (after debounce)
Else    ERR_SYM_ LOCAN_BOFF = NO_SYM
        LV_ERR_LOCAN_BOFF = 0          (after rebounce)
EndIf
```

Calculation end of diagnosis

LV_END_DIAG_LOCAN_BOFF is calculated by error management

43.11 BSD diagnosis

43.11.1 Global BSD Error – No communication to any of the BSD components

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_BSD	V/O	0...1H	0...1	1	[-]
logical variable for BSD error					
LV_CDN_DIAG_BSD	V/O	0...1H	0...1	1	[-]
Diagnosis condition BSD diagnosis					
ERR_SYM_BSD	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom BSD					
LV_END_DIAG_BSD	V/O	0...1H	0...1	1	[-]
End of BSD - Diagnosis					

Input data:

LV_CDN_VB_MIN_DIAG	LV_IGK		

Export actions:

ACTION_AGGR_*SetBsdGlobalCom* (IN <CTR_BSD_CPT_COM_OK>)
Action which delivers at the end of the BSD scheduling timing table the status of the communication of all BSD components of the list (counter = 0 ; no communication occured at all)

Description for actions:

ACTION_COMS_*SetBsdGlobalCom _COM* (<CTR_BSD_CPT_COM_OK>)

Action which delivers at the end of the BSD timing table the status of the communication of all BSD devices of the list; number of successful communications for one scheduling timing list cycle

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_BSD_CPT_COM_OK	IN	0...FFH	0...255	1	-
Number of the ‘ok’ communications per total scheduling timing list cycle of the BSD devices					

FUNCTION DESCRIPTION:

General information:

This diagnosis is to detect a malfunction in the BSD-line (**B**it-**S**erial-**D**ata transfer). There could be several components be connected to the BSD bus (alternator, intelligent battery sensor, oil quality sensor, electrical coolant water pump).
The diagnosis works with the help of basic software information. To detect an error symptom it is necessary to have a data tranfer before, thus the information is stored until the next recurreny of the diagnosis. The error symptoms are set exclusive per data transfer.
There is a certain scheduling defined to serve the several components via BSD bus. In total for one complete cycle (defined in the BSD scheduling timing table) several time slots have to be passed after a decision can be made wether the communication to at least one BSD component has successfully been passed. Is there no positve acknowegde from none of the BSD components, the counter CTR_BSD_CPT_COM_OK = 0 is indicating a global communication.
Each component is linked to certain time slots.
In general each component can be either configered by a calibration switch (‘must’ component; e.g. alternater, electrical coolant water pump, oil quality sensor) or can be learnt (intelligent battery sensor).

Description:

As soon as the communication via BSD has been started the ECU expects in all time slots correct messages (positive acknowledge). In case none of the messages have been received the global BSD error is set after debounce followed by inhibiting of the individual BSD communication errors.
After the error has been detected, the BSD timing coordinator is still serving the BSD components in the hope that the problem which disturbed the global communication disapears.

The error symptom detection is made out of basic software information BIOS_BSD_xx:
If one of the following error symptoms are detected:

- To less Bits were received (*BIOS_BSD_TOO_FEW_BITS_ERR* is set)
- To many Bits were received (*BIOS_BSD_TOO_MANY_BITS_ERR* is set)
- At least one data-bit was not inside the puls (*BIOS_BSD_BIT_LVL_ERR* is set)
- Receiver send “NAK” (*BIOS_BSD_ACK_ERR* is set)
- Receiver does not respond (*BIOS_BSD_SLAVE_TO_ERR* is set)
- One of the parity-bits in the data frame is wrong (*BIOS_BSD_PARITY_ERR* is set)
- Period of last measured bit was not o.k. (*BIOS_BSD_PER_ERR* is set)

then the error symptom “BSD signal error” is set” to active.

For error symptom calculation (*ERR_SYM_BSD*) the symptom position is:

```
☐☐☐☐
 | | | └─> -                       (= SYM_0)
 | | └───> -                       (= SYM_1)
 | └─────> BSD signal error        (= SYM_2)
```

```
└────────> -                              (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** at reset

Recurrence: Function called after each communication block (see BSD timing table)

Activation:

```
If      LV_CDN_VB_MIN_DIAG = 1 and
        Communication on BSD started
Then    LV_CDN_DIAG_BSD = 1
Else    LV_CDN_DIAG_BSD = 0
Endif
```

Error management:

```
If      LV_CDN_DIAG_BSD = 1
Then If     CTR_BSD_CPT_COM_OK = 0      //no BSD com occured at all conf. CPT
     Then   ERR_SYM_BSD = SYM_2         //Error symptom detected, ABC increments
     Else   ERR_SYM_BSD = NO_SYM        //No error symptom detected, ABC decrements
     Endif
Endif
```

Remark: LV_ERR_BSD and LV_END_DIAG_BSD are calculated by error management if diagnosis is active.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_BSD	1	0...FFH	0...255	1	[-]
anti bounce counter increment - diagnosis BSD					
C_ABC_MAX_BSD	1	1...FFH	1...255	1	[-]
anti bounce counter maximum value - diagnosis BSD					

43.11.2 Single BSD component errors (ALTER, QOIL, CWP, SENS_BAT_SMT)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ALTER_COM	V/O	0...1H	0...1	1	[-]
Logical variable for alternator BSD communication error					
LV_ERR_CWP_COM	V/O	0...1H	0...1	1	[-]
Logical variable for electrical coolant water pump BSD communication error					
LV_ERR_QOIL_COM	V/O	0...1H	0...1	1	[-]
Logical variable for oil quality sensor (QLT) BSD communication error					
LV_ERR_SENS_BAT_SMT_COM	V/O	0...1H	0...1	1	[-]
Logical variable for intelligent battery sensor (IBS) BSD communication error					
CTR_ABC_ALTER_COM	V	0...FFH	0...255	1	[-]
Anti-bounce counter for alternator BSD communication error					
CTR_ABC_CWP_COM	V	0...FFH	0...255	1	[-]
Anti-bounce counter for electrical coolant water pump BSD communication error					
CTR_ABC_QOIL_COM	V	0...FFH	0...255	1	[-]
Anti-bounce counter for oil quality sensor (QLT) BSD communication error					
CTR_ABC_SENS_BAT_SMT_COM	V/O	0...FFH	0...255	1	[-]
Anti-bounce counter for intelligent battery sensor (IBS) BSD communication error					
LV_CDN_DIAG_ALTER_COM	V/O	0...1H	0...1	1	[-]
Diagnosis condition for alternator BSD communication error					
LV_CDN_DIAG_CWP_COM	V/O	0...1H	0...1	1	[-]
Diagnosis condition for electrical coolant water pump BSD communication error					
LV_CDN_DIAG_QOIL_COM	V/O	0...1H	0...1	1	[-]
Diagnosis condition for oil quality sensor (QLT) BSD communication error					
LV_CDN_DIAG_SENS_BAT_SMT_COM	V/O	0...1H	0...1	1	[-]
Diagnosis condition for intelligent battery sensor (IBS) BSD communication error					
ERR_SYM_ALTER_COM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom for alternator BSD communication error					
ERR_SYM_CWP_COM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom for electrical coolant water pump BSD communication error					
ERR_SYM_QOIL_COM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom for oil quality sensor (QLT) BSD communication error					
ERR_SYM_SENS_BAT_SMT_COM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom for intelligent battery sensor (IBS) BSD communication error					
LV_END_DIAG_ALTER_COM	V/O	0...1H	0...1	1	[-]
End of diagnosis condition for alternator BSD communication error					
CTR_END_DIAG_ALTER_COM	V	0...FFH	0...255	1	[-]
Counter for End of diagnosis condition detection for alternator BSD communication error					
LV_END_DIAG_CWP_COM	V/O	0...1H	0...1	1	[-]
End of diagnosis condition for electrical coolant water pump BSD communication error					
CTR_END_DIAG_CWP_COM	V	0...FFH	0...255	1	[-]
Counter for End of diagnosis condition detection for electrical coolant water pump BSD communication error					

LV_END_DIAG_QOIL_COM	V/O	0...1H	0...1	1	[-]
End of diagnosis condition for oil quality sensor (QLT) BSD communication error					
CTR_END_DIAG_QOIL_COM	V	0...FFH	0...255	1	[-]
Counter for End of diagnosis condition detection for oil quality sensor (QLT) BSD communication error					
LV_END_DIAG_SENS_BAT_SMT_COM	V/O	0...1H	0...1	1	[-]
End of diagnosis condition for intelligent battery sensor (IBS) BSD communication error					
CTR_END_DIAG_SENS_BAT_SMT_COM	V	0...FFH	0...255	1	[-]
Counter for End of diagnosis condition detection for intelligent battery sensor (IBS) BSD communication error					
CTR_MSG_ALTER_COM_DIAG	V	0...FFH	0...255	1	[-]
Counter of faulty alternator transmitting slots					
CTR_MSG_CWP_COM_DIAG	V	0...FFH	0...255	1	[-]
Counter of faulty electrical coolant water pump transmitting slots					
CTR_MSG_QOIL_COM_DIAG	V	0...FFH	0...255	1	[-]
Counter of faulty oil quality sensor (QLT) transmitting slots					
CTR_MSG_SENS_BAT_SMT_COM_DIAG	V/O	0...FFH	0...255	1	[-]
Counter of faulty intelligent battery sensor (IBS) transmitting slots					
CTR_MSG_ALTER_COM_STOP	V/O	0...FFFFH	0...65535	1	[-]
Interrupt counter to stop the communictation to the alternator					
CTR_MSG_CWP_COM_STOP	V/O	0...FFFFH	0...65535	1	[-]
Interrupt counter to stop the communictation to the electrical coolant water pump					
CTR_MSG_QOIL_COM_STOP	V/O	0...FFFFH	0...65535	1	[-]
Interrupt counter to stop the communictation to the oil quality sensor (QLT)					
CTR_MSG_SENS_BAT_SMT_COM_STOP	V/O	0...FFFFH	0...65535	1	[-]
Interrupt counter to stop the communictation to the intelligent battery sensor (IBS)					
LV_INH_CWP_DIAG	V/O	0...1H	0...1	1	[-]
Flag to freeze the CWP diagnosis in case a non-acknowledge of the BSD component CWP occurred					
STATE_BSD_MSG_COM_STOP	V/O	0...FFH	0...255	1	[-]
Status of the communication STOP counter to the BSD devices					
LV_MSG_ALTER_COM_STOP	V	0...1H	0...1	1	[-]
Flag indication that the communication stop counter of the device alternator <>0					
LV_MSG_CWP_COM_STOP	V	0...1H	0...1	1	[-]
Flag indication that the communication stop counter of the device el. Coolant water pump <>0					
LV_MSG_QOIL_COM_STOP	V	0...1H	0...1	1	[-]
Flag indication that the communication stop counter of the device oil quality sensor <>0					
LV_MSG_SENS_BAT_SMT_COM_STOP	V	0...1H	0...1	1	[-]
Flag indication that the communication stop counter of the device intelligent battery sensor <>0					

Input data:

LV_CDN_VB_MIN_DIAG	LV_IGK		

Export actions:

ACTION_AGGR_*SetBsdSingleCom* (IN<STATE_BSD_CPT>,IN<STATE_BSD_COM>)
Action which delivers at the end of the BSD timing slot the status of the communication of the actual BSD device of the scheduling list

Description for actions:

ACTION_COMS_*SetBsdSingleCom* (<STATE_BSD_CPT>,<STATE_BSD_COM>)					
Action which delivers at the end of the BSD timing slot the status of the communication of the actual BSD device of the scheduling list					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
STATE_BSD_CPT	IN	0...FFH	0...255	1	-
BSD device, which communicates actually with the ECU via BSD					
STATE_BSD_COM	IN	0...FFH	0...255	1	-
Status of the communication to the BSD devices (faulty /o.k.)					

FUNCTION DESCRIPTION:

General information:

In addition to the global diagnosis of the BSD communication a selective diagnosis for each individual BSD component (alternator, intelligent battery sensor, oil quality sensor, electrical coolant water pump) is performed. There are several counters introduced to control the diagnosis. A counter (CTR_MSG_xxx_COM_DIAG) which monitors the transmitting slot of each individual component is incremented as soon as a faulty transmitting slot has been detected by the BSW (symptoms of the error please see 'Global BSD Error – No communication to any of the BSD components'). In case of a faulty transmitting slot (no positive acknowledge) has been detected the next corresponding transmitting slot is served by a '22 synchronisation pulse phase' and an incrementation of the counter CTR_MSG_xxx_COM_DIAG by 1+1 (1 for the missing acknowledge and 1 for the 22 synchronisation pulses).

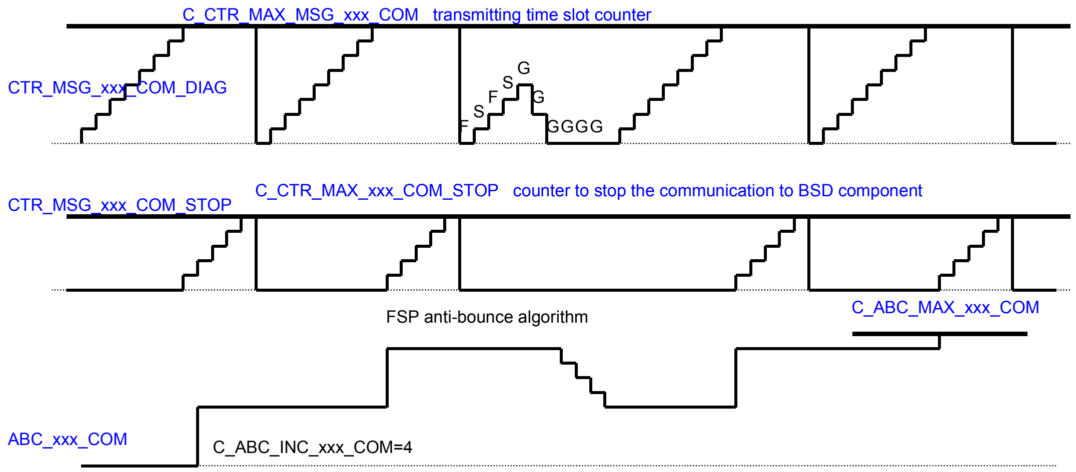
The '22 synchronisation pulse phase' is followed by new communication trial in the transmitting slot of the corresponding component determined by the timing coordinator. In case of a positive acknowledge the counter CTR_MSG_xxx_COM_DIAG is decremted by 2, otherwise it is proceeded as decribed above.

As soon as the counter CTR_MSG_xxx_COM_DIAG exceeds the threshold C_CTR_MAX_MSG_xxx_COM, a break for certain transmitting time slots can be applied by starting the counter CTR_MSG_xxx_COM_STOP (displayed in the flags LV_MSG_xxx_COM_STOP) in combination with the call of the anti-bounce algorithm of the error management and resetting the counter CTR_MSG_xxx_COM_DIAG after the counter CTR_MSG_xxx_COM_STOP has reached the threshold C_CTR_MAX_xxx_COM_STOP. The counter CTR_MSG_xxx_COM_STOP is incremented in each corresponding transmitting time slot until the threshold C_CTR_MAX_xxx_COM_STOP has been reached. During this period the timing coordinator does not serve the corresponding component. After the 'break' in the communication to the corresponding component has elapsed (C_CTR_MAX_xxx_COM_STOP reached) the next test cycle is started as described above. This occurs until the error has been debounced by the anti-bounce algorithm. After the debounce of the error the component is no longer served in this driving cycle by the timing coordinator.

If the counter threshold C_CTR_MAX_xxx_COM_STOP to stop the communication to the corresponding BSD component is calibrated to = 0, no 'break' of the communication occurs. Special treatment if the counter threshold is calibrated to C_CTR_MAX_xxx_COM_STOP = FFFFH. In that case no 'break' of the communication occurs and the component is served although the error has been detected.

With xxx:	ALTER	– alternator
	CWP	– electrical coolant water pump
	QOIL	– oil quality sensor (QLT)
	SENS_BAT_SMT	– intelligent battery sensor (IBS)

Description:



F -- faulty
G -- good
S -- 22 sync. pulse

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** at reset

Recurrence: Function called after each communication time slot (see BSD timing table)

Activation:

If LV_CDN_VB_MIN_DIAG = 1 **and**
Communication on BSD started **and**
LV_ERR_BSD = 0
Then LV_CDN_DIAG_xxx_COM = 1
Else LV_CDN_DIAG_xxx_COM = 0
Endif

Formula section:

```
If      STATE_BSD_CPT = 0H                        // time slot for SENS_BAT_SMT was active//
        CTR_END_DIAG_SENS_BAT_SMT_COM++           // END_DIAG in case no error//
Then
   If      C_CTR_MAX_SENS_BAT_SMT_COM_STOP<> FFFFH
   Then
      If      CTR_END_DIAG_SENS_BAT_SMT_COM =
              ((C_CTR_MAX_MSG_SENS_BAT_SMT_COM +
               C_CTR_MAX_SENS_BAT_SMT_COM_STOP) •
              (C_ABC_MAX_SENS_BAT_SMT_COM /
               C_ABC_INC_SENS_BAT_SMT_COM ))
      Then LV_END_DIAG_SENS_BAT_SMT_COM = 1
      Endif
   Else
      If      CTR_END_DIAG_SENS_BAT_SMT_COM =
              C_CTR_MAX_MSG_SENS_BAT_SMT_COM •
              (C_ABC_MAX_SENS_BAT_SMT_COM /
               C_ABC_INC_SENS_BAT_SMT_COM )
      Then LV_END_DIAG_SENS_BAT_SMT_COM = 1
      Endif
   Endif
   If      STATE_BSD_COM = FFH                    // time out for SENS_BAT_SMT is active//
   Then    CTR_MSG_SENS_BAT_SMT_COM_DIAG =
           CTR_MSG_SENS_BAT_SMT_COM_DIAG
           CTR_MSG_SENS_BAT_SMT_COM_STOP++
           //Communication passive during time slot to SENS_BAT_COM //
           LV_MSG_SENS_BAT_SMT_COM_STOP == 1
      If      CTR_MSG_SENS_BAT_SMT_COM_STOP =
                 C_CTR_MAX_SENS_BAT_SMT_COM_STOP
      Then    reset CTR_MSG_SENS_BAT_SMT_COM_DIAG == 0
              reset CTR_MSG_SENS_BAT_SMT_COM_STOP == 0
              reset LV_MSG_SENS_BAT_SMT_COM_STOP == 0
      Endif
   Else
      If      STATE_BSD_COM = 1H                  // no acknowledge from SENS_BAT_SMT//
      Then    CTR_MSG_SENS_BAT_SMT_COM_DIAG++
      If      CTR_MSG_SENS_BAT_SMT_COM_DIAG =
                 C_CTR_MAX_MSG_SENS_BAT_SMT_COM
      //Error symptom detected, ABC increments//
      Then    ERR_SYM_SENS_BAT_SMT_COM = SYM_2
              CTR_ABC_SENS_BAT_SMT_COM=
                 CTR_ABC_SENS_BAT_SMT_COM +
                    C_ABC_INC_SENS_BAT_SMT_COM
              Limitation to C_ABC_MAX_SENS_BAT_SMT_COM
         If      CTR_ABC_SENS_BAT_SMT_COM =
                 C_ABC_MAX_SENS_BAT_SMT_COM
         Then    LV_ERR_SENS_BAT_SMT_COM == 1
                 LV_END_DIAG_SENS_BAT_SMT_COM == 1
         Endif
         If      C_CTR_MAX_SENS_BAT_SMT_COM_STOP<> (FFFFH or 0H)
```

```
       Then CTR_MSG_SENS_BAT_SMT_COM_STOP++
            //Communication passive during time slot to SENS_BAT_COM //
            LV_MSG_SENS_BAT_SMT_COM_STOP == 1
        If   CTR_MSG_SENS_BAT_SMT_COM_STOP =
             C_CTR_MAX_SENS_BAT_SMT_COM_STOP
        Then reset CTR_MSG_SENS_BAT_SMT_COM_DIAG == 0
             reset CTR_MSG_SENS_BAT_SMT_COM_STOP == 0
             reset LV_MSG_SENS_BAT_SMT_COM_STOP == 0
        Endif
       Else If CTR_ABC_SENS_BAT_SMT_COM <
               C_ABC_MAX_SENS_BAT_SMT_COM
           Then reset CTR_MSG_SENS_BAT_SMT_COM_DIAG == 0
           Else
           Endif
       Endif
      Endif
     Else CTR_MSG_SENS_BAT_SMT_COM_DIAG =
             CTR_MSG_SENS_BAT_SMT_COM_DIAG – 2
      If    CTR_MSG_SENS_BAT_SMT_COM_DIAG = 0
      //Error symptom detected, ABC decrements//
      Then  ERR_SYM_SENS_BAT_SMT_COM = NO_SYM
            CTR_ABC_SENS_BAT_SMT_COM=
            CTR_ABC_SENS_BAT_SMT_COM – 1
       If    CTR_ABC_SENS_BAT_SMT_COM = 0
       Then  LV_ERR_SENS_BAT_SMT_COM == 0
       Endif
      Endif
     Endif
  Endif
Else If STATE_BSD_CPT = 3H                          // time slot for CWP was active//
        CTR_END_DIAG_CWP_COM++                      // END_DIAG in case no error//
    Then
      If    C_CTR_MAX_CWP_COM_STOP<> FFFFH
      Then
        If   CTR_END_DIAG_CWP_COM =
             ((C_CTR_MAX_MSG_CWP_COM +
              C_CTR_MAX_CWP_COM_STOP) •
             (C_ABC_MAX_CWP_COM /
              C_ABC_INC_CWP_COM ))
        Then LV_END_DIAG_CWP_COM = 1
        Endif
      Else
        If   CTR_END_DIAG_CWP_COM =
             C_CTR_MAX_MSG_CWP_COM •
             (C_ABC_MAX_CWP_COM /
              C_ABC_INC_CWP_COM )
        Then LV_END_DIAG_CWP_COM = 1
        Endif
      Endif
      If    STATE_BSD_COM = FFH                     // time out for CWP is active//
      Then  CTR_MSG_CWP_COM_DIAG =
            CTR_MSG_CWP_COM_DIAG
```

```
          LV_INH_CWP_DIAG = 1                  // time out CWP is active; no func. Diag.//
          CTR_MSG_CWP_COM_STOP++
          //Communication passive during time slot to CWP //
          LV_MSG_CWP_COM_STOP == 1
      If       CTR_MSG_CWP_COM_STOP =
               C_CTR_MAX_CWP_COM_STOP
      Then     reset CTR_MSG_CWP_COM_DIAG == 0
               reset CTR_MSG_CWP_COM_STOP == 0
               reset LV_MSG_CWP_COM_STOP == 0
      Endif
Else
    If       STATE_BSD_COM = 8H                 // no acknowledge from CWP//
    Then     CTR_MSG_CWP_COM_DIAG++
        If       CTR_MSG_CWP_COM_DIAG ≥
                 C_CTR_THD_MSG_CWP_COM_DIAG
        Then     LV_INH_CWP_DIAG = 1           //CWP com. BSD disturbed; no func. Diag.//
        Endif
        If       CTR_MSG_CWP_COM_DIAG =
                 C_CTR_MAX_MSG_CWP_COM
        //Error symptom detected, ABC increments//
        Then     ERR_SYM_CWP_COM = SYM_2
                 CTR_ABC_CWP_COM=
                   CTR_ABC_CWP_COM +
                     C_ABC_INC_CWP_COM
                 Limitation to C_ABC_MAX_CWP_COM
          If       CTR_ABC_CWP_COM =
                   C_ABC_MAX_CWP_COM
          Then     LV_ERR_CWP_COM == 1
                   LV_END_DIAG_CWP_COM == 1
          Endif
          If       C_CTR_MAX_CWP_COM_STOP<> (FFFFH or 0H)
          Then     CTR_MSG_CWP_COM_STOP++
                   //Communication passive during time slot to CWP//
                   LV_MSG_CWP_COM_STOP == 1
            If       CTR_MSG_CWP_COM_STOP =
                     C_CTR_MAX_CWP_COM_STOP
            Then     reset CTR_MSG_CWP_COM_DIAG == 0
                     reset CTR_MSG_CWP_COM_STOP == 0
                     reset LV_MSG_CWP_COM_STOP == 0
                     reset LV_INH_CWP_DIAG == 0  //reset com. disturbance bit.//
            Endif
          Else If    CTR_ABC_CWP_COM <
                     C_ABC_MAX_CWP_COM
                 Then reset CTR_MSG_CWP_COM_DIAG == 0
                 Else
                 Endif
          Endif
        Endif
    Else     CTR_MSG_CWP_COM_DIAG =
                       CTR_MSG_CWP_COM_DIAG – 2
        If     CTR_MSG_CWP_COM_DIAG <
                 C_CTR_THD_MSG_CWP_COM_DIAG
```

```
              Then  LV_INH_CWP_DIAG = 0            //reset com. disturbance bit.//
              Endif
              If    CTR_MSG_CWP_COM_DIAG = 0
              //Error symptom detected, ABC decrements//
              Then  ERR_SYM_CWP_COM = NO_SYM
                    CTR_ABC_CWP_COM=
                    CTR_ABC_CWP_COM – 1
                If      CTR_ABC_CWP_COM = 0
                Then    LV_ERR_CWP_COM == 0
                Endif
              Endif
            Endif
         Endif
        Else If  STATE_BSD_CPT = 4H                // time slot for QOIL was active//
                 CTR_END_DIAG_QOIL_COM++           // END_DIAG in case no error//
            Then
             If     C_CTR_MAX_QOIL_COM_STOP<> FFFFH
             Then
               If    CTR_END_DIAG_QOIL_COM =
                     ((C_CTR_MAX_MSG_QOIL_COM +
                      C_CTR_MAX_QOIL_COM_STOP) •
                     (C_ABC_MAX_QOIL_COM /
                      C_ABC_INC_QOIL_COM ))
               Then  LV_END_DIAG_QOIL_COM = 1
               Endif
             Else
               If    CTR_END_DIAG_QOIL_COM =
                     C_CTR_MAX_MSG_QOIL_COM •
                     (C_ABC_MAX_QOIL_COM /
                      C_ABC_INC_QOIL_COM )
               Then  LV_END_DIAG_QOIL_COM = 1
               Endif
             Endif
             If      STATE_BSD_COM = FFH           // time out for QOIL is active//
             Then    CTR_MSG_QOIL_COM_DIAG =
                     CTR_MSG_QOIL_COM_DIAG
                     CTR_MSG_QOIL_COM_STOP++
                     //Communication passive during time slot to QOIL //
                     LV_MSG_QOIL_COM_STOP == 1
               If          CTR_MSG_QOIL_COM_STOP =
                                C_CTR_MAX_QOIL_COM_STOP
               Then        reset CTR_MSG_QOIL_COM_DIAG == 0
                                reset CTR_MSG_QOIL_COM_STOP == 0
                                reset LV_MSG_QOIL_COM_STOP == 0
               Endif
             Else
                If     STATE_BSD_COM = 10H         // no acknowledge from QOIL//
                Then   CTR_MSG_QOIL_COM_DIAG++
                  If      CTR_MSG_QOIL_COM_DIAG =
                               C_CTR_MAX_MSG_QOIL_COM
                  //Error symptom detected, ABC increments//
                  Then    ERR_SYM_QOIL_COM = SYM_2
```

```
               CTR_ABC_QOIL_COM=
                CTR_ABC_QOIL_COM +
                 C_ABC_INC_QOIL_COM
                    Limitation to C_ABC_MAX_QOIL_COM
       If   CTR_ABC_QOIL_COM =
            C_ABC_MAX_QOIL_COM
       Then LV_ERR_QOIL_COM == 1
            LV_END_DIAG_QOIL_COM == 1
       Endif
       If   C_CTR_MAX_QOIL_COM_STOP<> (FFFFH or 0H)
       Then CTR_MSG_QOIL_STOP++
            //Communication passive during time slot to QOIL//
            LV_MSG_QOIL_COM_STOP == 1
        If    CTR_MSG_QOIL_STOP =
              C_CTR_MAX_QOIL_COM_STOP
        Then  reset CTR_MSG_QOIL_COM_DIAG == 0
              reset CTR_MSG_QOIL_COM_STOP == 0
              reset LV_MSG_QOIL_COM_STOP == 0
        Endif
       Else If   CTR_ABC_QOIL_COM <
                 C_ABC_MAX_QOIL_COM
            Then reset CTR_MSG_QOIL_COM_DIAG == 0
            Else
            Endif
       Endif
      Endif
    Else CTR_MSG_QOIL_COM_DIAG =
               CTR_MSG_QOIL_COM_DIAG – 2
      If   CTR_MSG_QOIL_COM_DIAG = 0
      //Error symptom detected, ABC decrements//
      Then ERR_SYM_QOIL_COM = NO_SYM
           CTR_ABC_QOIL_COM=
           CTR_ABC_QOIL_COM – 1
        If    CTR_ABC_QOIL_COM = 0
        Then  LV_ERR_QOIL_COM == 0
        Endif
      Endif
    Endif
 Endif
Else If   STATE_BSD_CPT = 6H        // time slot for ALTER was active//
          CTR_END_DIAG_ALTER_COM++     // END_DIAG in case no error//
    Then
      If   C_CTR_MAX_ALTER_COM_STOP<> FFFFH
      Then
        If    CTR_END_DIAG_ALTER_COM =
              ((C_CTR_MAX_MSG_ALTER_COM +
               C_CTR_MAX_ALTER_COM_STOP) •
              (C_ABC_MAX_ALTER_COM /
               C_ABC_INC_ALTER_COM ))
        Then  LV_END_DIAG_ALTER_COM = 1
        Endif
      Else
```

```
    If      CTR_END_DIAG_ALTER_COM =
            C_CTR_MAX_MSG_ALTER_COM •
            (C_ABC_MAX_ALTER_COM /
             C_ABC_INC_ALTER_COM )
    Then    LV_END_DIAG_ALTER_COM = 1
    Endif
Endif
If     STATE_BSD_COM = FFH    // time out for ALTER is active//
Then CTR_MSG_ALTER_COM_DIAG =
       CTR_MSG_ALTER_COM_DIAG
       CTR_MSG_ALTER_COM_STOP++
       //Communication passive during time slot to ALTER //
       LV_MSG_ ALTER _COM_STOP == 1
    If        CTR_MSG_ ALTER _COM_STOP =
              C_CTR_MAX_ ALTER _COM_STOP
    Then      reset CTR_MSG_ ALTER _COM_DIAG == 0
              reset CTR_MSG_ ALTER _COM_STOP == 0
              reset LV_MSG_ ALTER _COM_STOP == 0
    Endif
Else
  If       STATE_BSD_COM = 40H  // no acknowledge from ALTER//
  Then   CTR_MSG_ALTER_COM_DIAG++
        If       CTR_MSG_ALTER_COM_DIAG =
                 C_CTR_MAX_MSG_ALTER_COM
        //Error symptom detected, ABC increments//
        Then    ERR_SYM_ALTER_COM = SYM_2
                CTR_ABC_ALTER_COM=
                  CTR_ABC_ALTER_COM +
                    C_ABC_INC_ALTER_COM
                Limitation to C_ABC_MAX_ALTER_COM
          If      CTR_ABC_ALTER_COM =
                  C_ABC_MAX_ALTER_COM
          Then   LV_ERR_ALTER_COM == 1
                  LV_END_DIAG_ALTER_COM == 1
          Endif
          If      C_CTR_MAX_ALTER_COM_STOP<> (FFFFH or 0H)
          Then   CTR_MSG_ALTER_STOP++
                  //Communication passive during time slot to ALTER//
                  LV_MSG_ALTER_COM_STOP == 1
            If      CTR_MSG_ALTER_STOP =
                    C_CTR_MAX_ALTER_COM_STOP
            Then   reset CTR_MSG_ALTER_COM_DIAG == 0
                    reset CTR_MSG_ALTER_COM_STOP == 0
                    reset LV_MSG_ALTER_COM_STOP == 0
            Endif
          Else  If       CTR_ABC_ALTER_COM <
                         C_ABC_MAX_ALTER_COM
                Then   reset CTR_MSG_ALTER_COM_DIAG == 0
                Else
                Endif
          Endif
        Endif
```

```
                    Else  CTR_MSG_ALTER_COM_DIAG =
                              CTR_MSG_ALTER_COM_DIAG – 2
                       If     CTR_MSG_ALTER_COM_DIAG = 0
                       //Error symptom detected, ABC decrements//
                       Then ERR_SYM_ALTER_COM = NO_SYM
                              CTR_ABC_QOIL_COM=
                              CTR_ABC_QOIL_COM – 1
                          If      CTR_ABC_QOIL_COM = 0
                          Then    LV_ERR_QOIL_COM == 0
                          Endif

                       Endif
                    Endif
                 Endif
             Endif
         Endif
     Endif
```

With:
STATE_BSD_MSG_COM_STOP = LV_MSG_xxx_COM_STOP

STATE_BSD_CPT	STATE_BSD_MSG_COM_STOP
0H --SENS_BAT_SMT–intelligent battery sensor	1H = LV_MSG_SENS_BAT_SMT_COM_STOP=1
3H -- CWP– electrical coolant water pump	2H = LV_MSG_CWP_COM_STOP = 1
4H – QOIL – oil quality sensor (QLT)	4H = LV_MSG_QOIL_COM_STOP = 1
6H -- ALTER – alternator	8H = LV_MSG_ALTER_COM_STOP = 1

Calibration data:

<table>
<tr><th>Name</th><th>Dim</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>C_ABC_INC_ALTER_COM</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce counter increment - diagnosis BSD communication to alternater</td></tr>
<tr><td>C_ABC_MAX_ALTER_COM</td><td>1</td><td>1...FFH</td><td>1...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce counter maximum value - diagnosis BSD communication to alternater</td></tr>
<tr><td>C_ABC_INC_CWP_COM</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce counter increment - diagnosis BSD communication to electrical coolant water pump</td></tr>
<tr><td>C_ABC_MAX_CWP_COM</td><td>1</td><td>1...FFH</td><td>1...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce counter maximum value - diagnosis BSD communication to electrical coolant water pump</td></tr>
<tr><td>C_ABC_INC_QOIL_COM</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce counter increment - diagnosis BSD communication to oil quality sensor</td></tr>
<tr><td>C_ABC_MAX_QOIL_COM</td><td>1</td><td>1...FFH</td><td>1...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce counter maximum value - diagnosis BSD communication to oil quality sensor</td></tr>
<tr><td>C_ABC_INC_SENS_BAT_SMT_COM</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce counter increment - diagnosis BSD communication to intelligent battery sensor</td></tr>
<tr><td>C_ABC_MAX_SENS_BAT_SMT_COM</td><td>1</td><td>1...FFH</td><td>1...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Anti-bounce counter maximum value - diagnosis BSD communication to intelligent battery sensor</td></tr>
<tr><td>C_CTR_MAX_MSG_ALTER_COM</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Threshold for counter of faulty alternator transmitting slots</td></tr>
<tr><td>C_CTR_MAX_MSG_CWP_COM</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Threshold for counter of faulty electrical coolant water pump transmitting slots</td></tr>
<tr><td>C_CTR_MAX_MSG_QOIL_COM</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Threshold for counter of faulty oil quality sensor transmitting slots</td></tr>
<tr><td>C_CTR_MAX_MSG_SENS_BAT_SMT_COM</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Threshold for counter of faulty intelligent battery sensor transmitting slots</td></tr>
<tr><td>C_CTR_MAX_ALTER_COM_STOP</td><td>1</td><td>0...FFFFH</td><td>0...65535</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Time out threshold for counter of alternator transmitting slots</td></tr>
<tr><td>C_CTR_MAX_CWP_COM_STOP</td><td>1</td><td>0...FFFFH</td><td>0...65535</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Number of time out transmitting slots for electrical coolant water pump</td></tr>
<tr><td>C_CTR_MAX_QOIL_COM_STOP</td><td>1</td><td>0...FFFFH</td><td>0...65535</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Number of time out transmitting slots for oil quality sensor</td></tr>
<tr><td>C_CTR_MAX_SENS_BAT_SMT_COM_STOP</td><td>1</td><td>0...FFFFH</td><td>0...65535</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Number of time out transmitting slots for intelligent battery sensor</td></tr>
<tr><td>C_CTR_THD_MSG_CWP_COM_DIAG</td><td>1</td><td>0...FFH</td><td>0...255</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Threshold to disable the functional CWP diagnosis in case a problem with the interface BSD is present</td></tr>
</table>

43.12 Plausibility diagnosis torque request via CAN

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TQ_REQ_CAN	V/O	0...1H	0...1	1	[-]
Torque request via CAN not plausible					
ERR_SYM_TQ_REQ_CAN	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
detected symptom of TQ-request via CAN					
LV_CDN_DIAG_TQ_REQ_CAN	V/O	0...1H	0...1	1	[-]
condition of TQ-request via CAN					
LV_END_DIAG_TQ_REQ_CAN	V/O	0...1H	0...1	1	[-]
end of diagnosis of TQ-request via CAN					
CTR_ABC_TQ_REQ_CAN	V/O	0...FFH	0...255	1	[-]
anti-bounce counter of TQ-request via CAN					
STATE_TQ_CAN_PLAUS	V/O	0...FFH	0...255	1	[-]
State indicating which ECU is rejected due to TQ-plausibility-error via CAN					

Input data:

STATE_TQ_TCS_PLAUS	STATE_TQ_AMT_PLAUS	STATE_TQ_ETCU_PLAUS	STATE_TQ_DCC_PLAUS
STATE_TQ_ARS_PLAUS	STATE_TQ_PSTE_2_PLAUS	ECU_STATE	LV_ERR_SYM_TQ_DCC_CS
LV_CDN_VB_BN_DIAG	STATE_TQ_PSTE_3_PLAUS		

FUNCTION DESCRIPTION:

General information:

The torque functionality detects if the torque requests via CAN are not plausible.
As a consequence of that further torque requests of the respective control unit are rejected. The purpose of this diagnosis is to provide the information when a ECU is rejected by error-entry in the FMY. Through environmental data (STATE_TQ_CAN_PLAUS) the information is given which ECU has been rejected.
Additional there is the possibility to calibrate via LC_TQ_CAN_PLAUS which failure of the respective ECU lead to the error entry.

If a torque-plausibility-error is detected by the torque calculation (STATE_TQ_xxx_PLAUS =) the error symptom is set and the anti bounce counter is incremented by C_ABC_INC_TQ_REQ_CAN. If the plausibility error is not present any longer, the anti bounce counter is decremented.

Application conditions:

Initialisation: according ERRM initialization type **"STD_INI"**

//reset at LV_IGK 0->1 or reset

Recurrence: 10 ms

Activation:

```
If    LV_CDN_VB_BN_DIAG = 1          and
      LV_ES  = 0
Then  LV_CDN_DIAG_TQ_REQ_CAN = 1
```

Deactivation:

```
If    LV_CDN_VB_BN_DIAG = 0          or
      LV_ES  = 1                     or
      ECU_STATE = PWL
Then  LV_CDN_DIAG_TQ_REQ_CAN = 0
Endif
```

Formula section:

```
IF    Bit  x  of  STATE_TQ_xxx_PLAUS is set              and
      Bit  x  of  C_STATE_TQ_CAN_PLAUS is set            and
      Depending on LC_TQ_CAN_PLAUS_INFO
      // --> see below
```

Steuergerät	STATE_TQ_xxx_PLAUS	Bits [0...15]	Condition	C_STATE_TQ_CAN_PLAUS_ENA Bits [0...7]	Condition	LC_TQ_CAN_PLAUS_INFO	Action	STATE_TQ_CAN_PLAUS Bits [0...7]
DSC	STATE_TQ_TCS_PLAUS	2	&&	0	&&	x	=	Bit 0 (= dez 1)
SMG	STATE_TQ_AMT_PLAUS	3	&&	1	&&	x	=	Bit 1 (= dez 2)
EGS	STATE_TQ_ETCU_PLAUS	2,3	&&	2	&&	0	=	Bit 2 (= dez 4)
		2	&&		&&	1	=	Bits 2+0 (= dez 5)
		3	&&		&&	1	=	Bits 2+1 (= dez 6)
ACC	STATE_TQ_DCC_PLAUS	5	&&	3	&&	x	=	Bit 3 (= dez 8)
ARS	STATE_TQ_ARS_PLAUS	2,3	&&	4	&&	0	=	Bit 4 (= dez 16)
		2	&&		&&	1	=	Bits 4+0 (= dez 17)
		3	&&		&&	1	=	Bits 4+1 (= dez 18)
AFS	STATE_TQ_PSTE_2_PLAUS	0,1,2	&&	5	&&	0	=	Bit 5 (= dez 32)
		0	&&		&&	1	=	Bits 5+0 (= dez 33)
		1	&&		&&	1	=	Bits 5+1 (= dez 34)
		2	&&		&&	1	=	Bits 5+1+0 (= dez 35)
ACC-HS	LV_ERR_SYM_TQ_DCC_CS	-	&&	6	&&	x	=	Bit 6 (= dez 64)
EHB3	STATE_TQ_PSTE_3_PLAUS	0,1	&&	7	&&	0	=	Bit 7 (= dez 128)
		0	&&		&&	1	=	Bits 7 + 0 (= dez 129)
		1	&&		&&	1	=	Bits 7 + 1 (= dez 130)

```
THEN    Bit x of STATE_TQ_CAN_PLAUS is set
        ERR_SYM_TQ_REQ_CAN = SYM_3
        LV_ERR_TQ_REQ_CAN = 1          after debounce
ELSE    ERR_SYM_TQ_REQ_CAN = NO_SYM
        LV_ERR_TQ_REQ_CAN = 0          after rebounce
ENDIF
```

End of diagnosis calculation LV_END_DIAG_TQ_REQ_CAN:

→ see chapter “Anti-bounce Algorithm: Calculation of the end of diagnosis”

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_MAX_TQ_REQ_CAN	1	1...FFH	1...255	1	[-]
Anti-bounce max. value, TQ-plausibility via CAN					
C_ABC_INC_TQ_REQ_CAN	1	0...FFH	0...255	1	[-]
Anti bounce increment value, TQ-plausibility via CAN					
C_STATE_TQ_CAN_PLAUS_ENA	1	0...FFH	0...255	1	[-]
Mask for selective enabling of diagnoses					
LC_TQ_CAN_PLAUS_INFO	1	0...1H	0...1	1	[-]
LC for selective information of diagnosis					

44 Integrated Powertrain Management

Table of Contents

Index

44.1 LDM diagnosis (LV_ERR_LDM)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_LDM	V/O	0...1H	0...1	1	[-]
LDM error reversible/ irreversible					
LV_CDN_DIAG_LDM	V/O	0...1H	0...1	1	[-]
Status of diagnosis LDM error reversible/ irreversible					
ERR_SYM_LDM	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Detected symptoms LDM error reversible/ irreversible					
LV_END_DIAG_LDM	V/O	0...1H	0...1	1	[-]
End of diagnosis LDM error reversible/ irreversible					
LV_LDM_OFF_ECU	V/O	0...1H	0...1	1	[-]
Irreversible LDM error					
LV_LDM_OFF_ECU_1	V	0...1H	0...1	1	[-]
LDM Irreversible off					
LV_LDM_OFF_ECU_2	V	0...1H	0...1	1	[-]
LDM reversible off					
LV_LDM_ENA_PLAUS_ERR	V/O	0...1H	0...1	1	[-]
Conditions for enabling LDM not plausible					
LV_LDM_BRAKE_PLAUS_ERR	V/O	0...1H	0...1	1	[-]
Conditions for enabling LDM not plausible					
T_LDM_BRAKE_DET	V	0...FFH	0...2.55	0.01	[s]
Brake delay timer					

Input data:

LV_VAR_BN_LDM	LV_IGK	LV_ERR_BLS_PLAUS	LV_ERR_CS
STATE_LDM	STATE_LDM_INTV	LV_TQ_WHEEL_LDM_REQ	LV_ERR_BN_LDM
LV_BRAKE_DET	LV_ETCU_DISABLE_CAN	LV_ERR_CRK	LV_ERR_PVS
LV_TQI_REQ_CAN_INH	LV_LDM_CAN_INI	LV_DI_TQ_REQ_CAN_MPI_GDI	

FUNCTION DESCRIPTION:

General information:

At implausible LDM messages, e.g. timeout errors, or ECU internal errors a reversible or irreversible error is detected and reflect to the LDM Unit.

Description:

The following symptoms can be detected:

```
|_|_|_|_|
 | | | └─> LDM Brake plausibility error     (= SYM_0)
 | | └───> -                               (= SYM_1)
 | └─────> -                               (= SYM_2)
 └───────> LDM plausibility error          (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 ms

Activation:

```
If      LV_VAR_BN_LDM = 1          and
        LV_IGK= 1                  and
        LV_LDM_OFF_ECU_1 = 0       and
        LV_LDM_CAN_INI = 0
Then    LV_CDN_DIAG_LDM = 1
```

Deactivation:

```
If      LV_VAR_BN_LDM = 0      or
        LV_IGK = 0             or
        LV_LDM_OFF_ECU_1 = 1   or
        LV_LDM_CAN_INI = 1
Then    LV_CDN_DIAG_LDM = 0
Endif
```

Formula section:

Calculation of error condition “brake detection” for setting irrversible error

```
If      LV_BRAKE_DET = 1
Then    T_LDM_BRAKE_DET$_{n}$ = T_LDM_BRAKE_DET$_{n-1}$ + 10 ms
Else    T_LDM_BRAKE_DET$_{n}$ =0 ms
Endif
```

Calculation of error condition “brake detection ldm not plausible” for setting irrversible error

```
If      [ (T_LDM_BRAKE_DET > C_T_MAX_BRAKE_DET_LDM) and
        STATE_LDM_INTV == 01H ]
Then    LV_LDM_BRAKE_PLAUS_ERR = 1
Else    LV_LDM_BRAKE_PLAUS_ERR = 0
Endif
```

Calculation of error condition “enable conditions not plausible” for setting irrversible error

```
If      (STATE_LDM ! = 1or 2 or 3 or 4      and
        LV_TQ_WHEEL_LDM_REQ = 1)
Then    LV_LDM_ENA_PLAUS_ERR = 1
Else    LV_LDM_ENA_PLAUS_ERR = 0
Endif
```

Calculation of error

```
If      LV_LDM_BRAKE_PLAUS_ERR = 1
Then    ERR_SYM_LDM = SYM_0
        LV_ERR_LDM = 1
        // LV_ERR_LDM = 1 set after debounce by error management, set for this DC
ElseIf  LV_LDM_ENA_PLAUS_ERR = 1            and
        LC_LDM_ENA_PLAUS_ERR = 1
Then    ERR_SYM_LDM = SYM_3
        LV_ERR_LDM = 1
        // LV_ERR_LDM = 1 set after debounce by error management
Else    If    LV_ERR_LDM = 0                //not debounced yet
        Then  ERR_SYM_LDM = NO_SYM          //debounce counter reset to 0
        Else  ERR_SYM_LDM  = NO_SYM
              LV_ERR_LDM = 0                //after rebounce
        Endif
Endif
```

Calculation of LDM irrversible off

```
If      [ LV_ERR_BN_LDM  = 1 and            //ONLY ALIVE AND CHECKSUM ERROR
        ERR_SYM_BN_LDM  = SYM_1 or SYM_3 ]                    or
        LV_TQI_REQ_CAN_INH    =    1                          or
        LV_ERR_PVS       =    1                               or
        LV_ERR_LDM       =    1
Then    LV_LDM_OFF_ECU_1 = 1          //LDM  irreversible off
```

Calculation of reversible off

```
ElseIf  ( LV_ERR_BN_LDM  = 1 and  ERR_SYM_BN_LDM      = SYM_2 ) or
                                                  //only timeout
        LV_ETCU_DISABLE_CAN   = 1                             or
        LV_ERR_BLS_PLAUS      = 1                             or
        LV_ERR_CS             = 1                             or
        LV_ERR_CRK            = 1                             or
        LV_DI_TQ_REQ_CAN_MPI_GDI = 1

Then    LV_LDM_OFF_ECU_2      = 1     //LDM reversible LDM off
Else    LV_LDM_OFF_ECU_2      = 0
Endif
```

Calculation of LV_LDM_OFF_ECU

```
If      LV_LDM_OFF_ECU_1 = 1   or
        LV_LDM_OFF_ECU_2 = 1
Then    LV_LDM_OFF_ECU = 1
Else    LV_LDM_OFF_ECU = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_LDM	1	0...FFH	0...255	1	[-]
Increment counter					
C_ABC_MAX_LDM	1	1...FFH	1...255	1	[-]
Maximum counter					
C_T_MAX_BRAKE_DET_LDM	1	0...FFH	0...2.55	0.01	[s]
Maximum Brake delay timer					
LC_LDM_ENA_PLAUS_ERR	1	0...1H	0...1	1	-
LC for enabling diagnosis due to plausibility error					

44.2 Torque request for LDM

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LDM_DISABLE_CAN	V/O	0...1H	0...1	1	[-]
Logical variable for disabling LDM intervention due to CAN error					
LV_LDM_ENA	V/O	0...1H	0...1	1	[-]
Logical variable for increased torque intervention due to LDM					
STATE_TQ_LDM_PLAUS	V/O	0...FFH	0...255	1	[-]
Bitwise coded State for LDM intervention state					
TQ_WHEEL_LDM_REQ	V	8000...7FFFH	-32768...32767	1	[Nm]
Torque increase for torque intervention during LDM					
TQ_SP_WHEEL	O/V	8000...7FFFH	-32768...32767	1	[Nm]
wheel torque setpoint					
LV_LDM_ACT	V/O	0...1H	0...1	1	[-]
Logical variable for torque interventiondue to LDM					
LV_LDM_PUC_INH	V/O	0...1H	0...1	1	[-]
Logical variable for PUC-inhibit due to LDM					
LV_LDM_LIH	V/O	0...1H	0...1	1	[-]
Logical variable for torque intervention LIH due to LDM					
T_LDM_LIH_CTR	V	0...FFH	0...2.55	0.01	[s]
Time counter delay time					

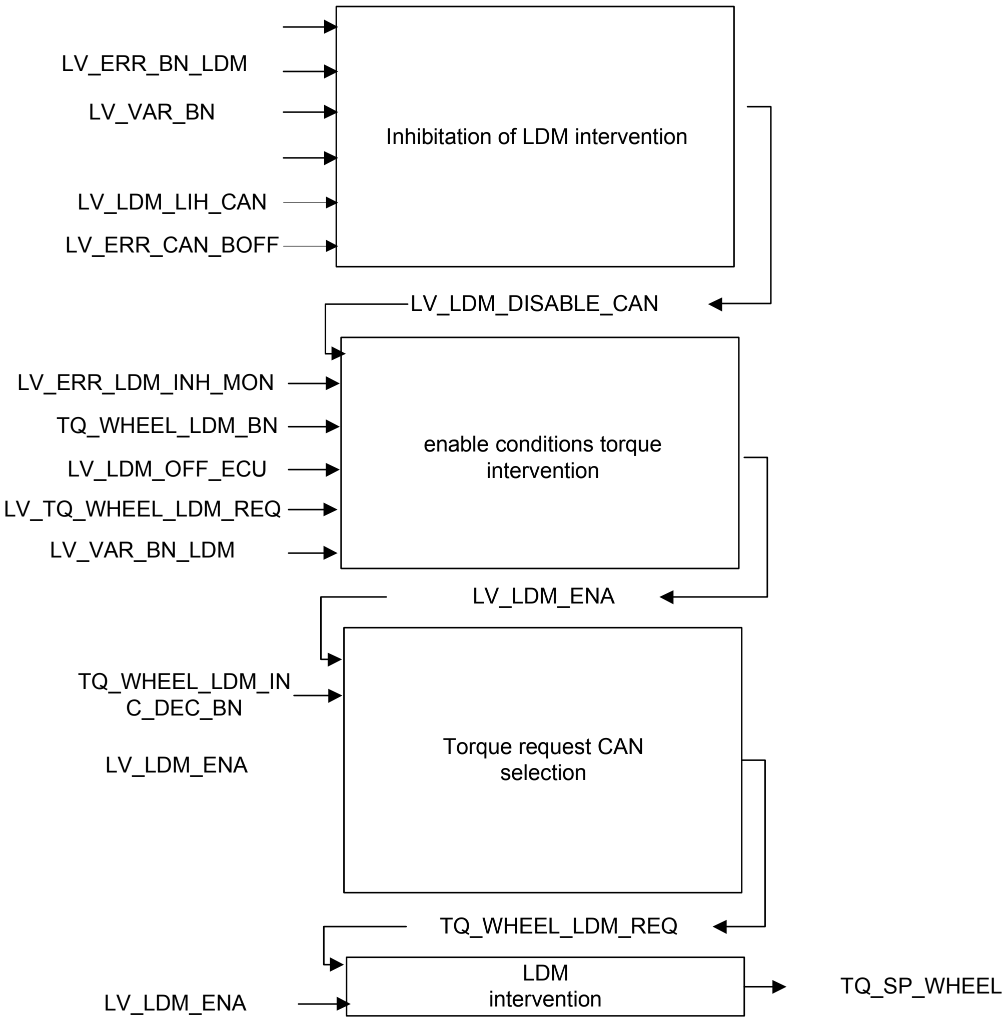
Input data:

LV_ERR_BN_LDM	LV_LDM_LIH_CAN	LV_CDN_VB_CAN_TQ_DIAG	LV_VAR_BN_LDM
TQ_WHEEL_LDM_BN	LV_IGK	LV_ERR_LDM_INH_MON	LV_ERR_CAN_BOFF
TQ_WHEEL_LDM_INC_DEC_BN	LV_LDM_OFF_ECU	STATE_DI_PUC	LV_TQ_WHEEL_LDM_REQ
LV_LDM_ENA_PLAUS_ERR	LV_LDM_BRAKE_PLAUS_ERR	LV_TQ_WHEEL_LDM_BN_ERR	LV_LDM_OFF_ECU
STATE_ENGG_POS			

General information:

The LDM unit delivers a torque request via BN. This module serves the enable conditions and the torque request which will be used in the torque request selection.

Signal flow diagram:



44.2.1 Enable conditions

44.2.1.1 Inhibitation LDM intervention

FUNCTION DESCRIPTION:

If the EMS detects an error on the CAN messages, the logical variable LV_LDM_DISABLE_CAN is set to 1.

Application conditions:

Initialisation: LV_LDM_DISABLE_CAN = 0 **at reset**

Recurrence: 10 ms

Activation: LV_IGK = 1 and LV_VAR_BN_LDM = 1

Deactivation: -

Formula section:

CAN enviroment LDM inhibition:

```
if      LV_ERR_BN_LDM            = 1        or
        LV_ERR_CAN_BOFF          = 1        or
        LV_LDM_LIH_CAN           = 1        or
        LV_CDN_VB_CAN_TQ_DIAG = 0

then        LV_LDM_DISABLE_CAN   = 1
else        LV_LDM_DISABLE_CAN   = 0
endif
```

44.2.1.2 enable conditions torque intervention

FUNCTION DESCRIPTION:

Depending on the configuration and plausible torque interfaces an increaced torque intervention is possible.

Application conditions:

Initialisation: LV_LDM_ENA = 0 **at reset**

Recurrence: 10 ms

Activation: LV_IGK = 1 and LV_VAR_BN_LDM =1

Formula section:

Torque increase intervention:

```
if      LV_LDM_OFF_ECU = 0                       and
        LV_TQ_WHEEL_LDM_REQ = 1                  and
        LV_LDM_DISABLE_CAN =  0                  and
        LV_TQ_WHEEL_LDM_BN_ERR = 0               and
        LV_ERR_LDM_INH_MON = 0                   and
        (STATE_ENGG_POS != 2H  or  LC_LDM_OFF_CLU = 0)
then    LV_LDM_ENA        = 1
else    LV_LDM_ENA        = 0
endif
```

STATE_TQ_LDM_PLAUS

```
if    LV_LDM_BRAKE_PLAUS_ERR = 1                              BIT [00] set
If    LV_LDM_ENA_PLAUS_ERR = 1                                BIT [01] set
if    LV_LDM_OFF_ECU = 1                                      BIT [03] set
if    LV_TQ_WHEEL_LDM_REQ = 1     and
      LV_TQ_WHEEL_LDM_BN_ERR = 1                              BIT [05] set
else                                                          BIT [04] set
endif
```

44.2.2 Torque request CAN selection

FUNCTION DESCRIPTION:

If the LDM intervention is enabled (LV_LDM_ENA = 1), the torque request is received from CAN and copied into TQ_SP_WHEEL.
After LV_LDM_ACT changes back to 0, TQ_SP_WHEEL becomes its minimum value again.

Application conditions:

Initialisation: at reset and LV_IGK off --> on

TQ_WHEEL_LDM_REQ = -32000 Nm

Recurrence: 10 ms

Activation: at LV_IGK = 1 and LV_VAR_BN_LDM = 1

Deactivation: -

Formula section:

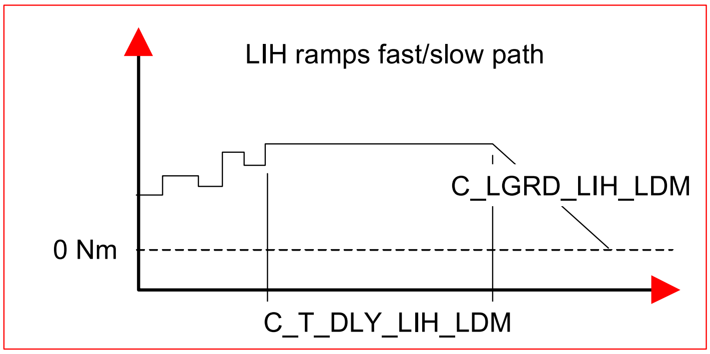
```
IF      LV_LDM_ENA = 1
Then    TQ_WHEEL_LDM_REQ = TQ_WHEEL_LDM_INC_DEC_BN
Else    TQ_WHEEL_LDM_REQ = -32000 Nm
Endif
```

44.2.2.1 LDM increased intervention and emergency operation

FUNCTION DESCRIPTION:

If during an active LDM intervention an LDM or an EMS error occurs, the current torque offset is continued for the time C_T_DLY_LIH_LDM and then subsequently decreased to TQ_SP_WHEEL = -32000 Nm using C_LGRD_LIH_LDM. The LDM unit itself controls the torque leading during active LV_TQ_WHEEL_LDM_REQ therefore you won't have a plausible Torque request without LV_LDM_ENA.

Signal flow diagram:



Application conditions:

Initialisation: at reset or at transition LV_IGK off -> on

LV_LDM_LIH = 0

LV_LDM_ACT = 0

TQ_SP_WHEEL = -32000 Nm

T_LDM_LIH_CTR = C_T_DLY_LIH_LDM

Recurrence: 10 ms

Activation: LV_IGK = 1 **and** LV_VAR_BN_LDM = 1

Formula section:

LDM_INC intervention and detection of LDM_INC limp home operation :

```
If(1)       LV_LDM_OFF_ECU = 1
then(1)
    if(2)       TQ_SP_WHEEL > -32000 Nm
    then(2)     LV_LDM_LIH = 1
        if(3)           T_LDM_LIH_CTR = 0                    (time delay)
        then(3)     $TQ\_SP\_WHEEL_{N} = TQ\_SP\_WHEEL_{N-1} - C\_LGRD\_LIH\_LDM$
                                            (till TQ_SP_WHEEL = -32000Nm )
        else(3)     $TQ\_SP\_WHEEL_{N} = TQ\_SP\_WHEEL_{N-1}$          (last valid value)
                    $T\_LDM\_LIH\_CTR_{N} = T\_LDM\_LIH\_CTR_{N-1} - 10\ ms$
                                    ( till T_LDM_LIH_CTR = C_T_DLY_LIH_LDM )
        endif(3)
    else(2)     LV_LDM_LIH = 0
                T_LDM_LIH_CTR = C_T_DLY_LIH_LDM
    endif(2)
else(1)         LV_LDM_LIH = 0
                T_LDM_LIH_CTR = C_T_DLY_LIH_LDM
                TQ_SP_WHEEL = TQ_WHEEL_LDM_REQ
endif(1)
```

LDM intervention active (LV_LDM_ACT) :

```
if          TQ_SP_WHEEL > -32000Nm
then        LV_LDM_ACT      = 1
else        LV_LDM_ACT      = 0
endif
```

LDM inhibitation of PUC active (LV_LDM_PUC_INH) :

```
if          (LV_LDM_ACT = 1   and
            STATE_DI_PUC = C_STATE_DI_PUC_ACT )
then        LV_LDM_PUC_INH  = 1
else        LV_LDM_PUC_INH  = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LGRD_LIH_LDM	1	0...7FFFH	0...1023.96875	0.03125	[Nm/10ms]
Limiting gradient for LDM emergency ramp function (default 0.6 Nm/10ms)					
C_T_DLY_LIH_LDM	1	0...FFH	0...2.55	0.01	[s]
Delay time for activation of LDM emergency ramp function (default 0s)					
C_STATE_DI_PUC_ACT	1	0...3H	0...3	1	[-]
State disable IS/PU/PUC engine states					
LC_LDM_OFF_CLU	1	0...1H	0...1	1	[-]
Switch for using LDM clutch info to deactivate LDM torque intervention					

45 Knock

Table of Contents

Index

45.1 BIOS Knock function

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
KNKS[NC_CYL_NR]	V/O	0...FFFFH	0...4.99992	7.63E-05	[V]
Knock noise raw signal for cylinder x					
KNKS_PRE[NC_CYL_NR]	V/O	0...FFFFH	0...4.99992	7.63E-05	[V]
Knock noise raw signal for cylinder x (first window after TDC)					
KNKS_REL_NL[NC_CYL_NR]	V/O	0...FFFFH	0...0.99998	1.53E-05	[-]
Relative knock value					

Input data:

KNKS_CMD_FIL[NC_CYL_NR]	KNKS_CMD_INT[NC_CYL_NR]	KNKS_CMD_GAIN[NC_CYL_NR]	KNKS_CMD_CONF[NC_CYL_NR]
KNKWB[NC_CYL_NR]	KNKWE[NC_CYL_NR]	NL[NC_CYL_NR]	KNKWB_PRE[NC_CYL_NR]
KNKWE_PRE[NC_CYL_NR]	KNKS_CMD_FIL_PRE[NC_CYL_NR]	KNKS_CMD_INT_PRE[NC_CYL_NR]	KNKS_CMD_GAIN_PRE[NC_CYL_NR]

General information:

Activation : at ENG_STOP to SYN_ENG_IGK_ON
or RUN_ENG to SYN_ENG_IGK_ON
T_KNK_ENABLE

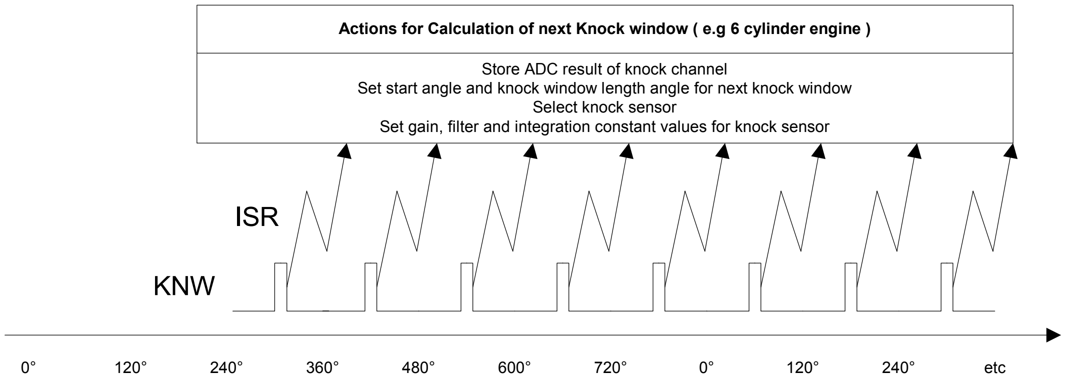
Deactivation : at SYN_ENG_IGK_ON to RUN_ENG (if supported by ECOP)
or SYN_ENG_IGK_ON to ENG_STOP
or SYN_ENG_IGK_OFF to PWL
T_KNK_DISABLE

Initialization : at reset or engine running to engine stop
KNKS_REL_NL[x] = C_KNKS_REL_INI (T_KNK_INIT)

The knock detection is performed for cylinder-individually defined measurement windows since knock-typical oscillations only occur for certain crankshaft angles. For the time of the knock window the knock signal is integrated. At the end of the knock window the integrated signal is sampled and converted to the digital value KNKSx.

KNKS_REL_NL[NC_CYL_NR] is the relative knocking value, it is the quotient of KNKS[NC_CYL_NR] and NL[NC_CYL_NR].

The knock window for cylinder x+1 is reprogrammed at the end of the knock window for cylinder x. The parameters KNKS_CMD_FIL[NC_CYL_NR], KNKS_CMD_FIL_PRE[NC_CYL_NR], KNKS_CMD_INT[NC_CYL_NR], KNKS_CMD_INT_PRE[NC_CYL_NR], KNKS_CMD_GAIN[NC_CYL_NR], KNKS_GAIN_PRE[NC_CYL_NR], KNKS_CMD_CONF[NC_CYL_NR], KNKWB KNKWB_PRE[NC_CYL_NR], KNKWE_PRE[NC_CYL_NR] and KNKWE have to be stable at this time. The knock channel adc is configured for triggering on the end of the knock window. The end of conversion generates an interrupt in which the parameters for the next knock window are set. The following picture illustrates this.



The knock function may be activated by T_KNK_INIT or T_KNK_ENABLE and deactivated by T_KNK_DISABLE.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_KNKS_REL_INI	1	0...FFFFH	0...0.99998	1.53E-05	[-]
ini value of relative knock signal					

45.2 KNCK - Requirements to infrastructure

Input data:

NC_KNKS_CYL_CONF_1	NC_KNKS_CYL_CONF_2	C_KNKS_CYL_CONF	

Export actions:

ACTION_INFR_*SetKnkWindow*(IN <Knkwb>, IN <Knkwd>, IN <Cyl>, IN <Win>)
Sets knock window for cylinder *cyl*
ACTION_INFR_*SetKnkCfg*(IN <Cmd_fil>, IN <Cmd_gain>, IN <Cmd_int>, IN <Cyl>, IN <Win>)
Sets knock pre-processor window, filter, gain and integrator factors for cylinder *cyl*
ACTION_INFR_*GetKnkDiag*(OUT <Diag>)
Gets the diagnosis information of knock pre-processor settings transfer
ACTION_INFR_*GetVKnks*(OUT <V_knks>, IN <Cyl>, IN <Win>)
Gets the formatted knock noise raw signal
ACTION_INFR_EnaKnks(IN <Enable>)
Enables the knock functionnality
ACTION_INFR_*InitKnKSensConf*()
This action Initialize theI/O driver configuration for different KNK Sensor to Cylinder configuration

Description for actions:

ACTION_INFR_*InitKnKSensConf*()

```
If      C_KNKS_CYL_CONF = 0
        Initialization of I/O driver configuration for configuration 1 fitting with NC_KNKS_CYL_CONF_1
Else
        Initialization of I/O driver configuration for configuration 2 fitting with NC_KNKS_CYL_CONF_2
EndIf
```

This action Initialize the I/O driver configuration at I/O driver initialisation depending on different "KNK Sensor to Cylinder configuration". This enables a dynamic run time configuration of KNCK.

This ACTION and implementation is not compatible to IRS guideline and aggregate implementation guideline. This solution is a temporary solution that will be managed in specific SW AGGR implementation file to isolate this solution.

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit

ACTION_INFR_*SetKnkWindow*((IN <Knkwb>, IN <Knkwd>, IN <Cyl>, IN <Win>)

This action sets the knock window for cylinder *Cyl*. The *Cyl* parameter is introduced to stay flexible regarding which cylinder to set. The knock window is defined by selected window, beginning (*Knkwb*) and duration (*Knkwd*) in degree crankshaft.

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit
Knkwb	IN	0...BBH	0...70.125	0.375	[°CRK]
Parameter for formatting of the knock window beginning					
Knkwd	IN	0...A0H	0...60	0.375	[°CRK]
Parameter for formatting of the knock window duration					
Cyl	IN	0...7H	0...7	1	[-]
Cylinder for which the knock window settings will apply					
Win	IN	0...1H	0...1	1	[-]
Window for which the knock window settings will apply					

ACTION_INFR_*SetKnkCfg*(IN <Cmd_fil>, IN <Cmd_gain>, IN <Cmd_int>, IN <Cyl>, IN <Win>)

This action passes the settings of the knock pre-processor for cylinder *Cyl*. The *Cyl* parameter is introduced to stay flexible regarding which cylinder to set. Physical values are mapped in the pre-processor and will be choosen corresponding to the transmitted hex. values. The correlation is given below in the action requirements. Since there is no linear dependency between the hex. values and the physical values of *Cmd_fil*, *Cmd_gain* and *Cmd_int*, no resolution is declared.
To check possible transmission errors ACTION_INFR_GetKnkDiag is introduced.

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit
Cmd_fil	IN	0...3FH	-	-	[-]
Filter centre frequency factor of the knock pre-processor					
Cmd_gain	IN	0...3FH	-	-	[-]
Gain of the knock pre-processor					
Cmd_int	IN	0...1FH	-	-	[-]
Integrator time constant of the knock pre-processor					
Cyl	IN	0...7H	0...7	1	[-]
Cylinder for which the transmitted factors will apply					
Win	IN	0...1H	0...1	1	[-]
Selected knock window					

ACTION_INFR_*GetKnkDiag*(OUT <Diag>)

This action checks if ACTION_INFR_SetKnockCfg was carried out successfully. In case of error, the parameter *Diag* is set to 1.

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit
Diag	OUT	0...1H	0 ... 1	1	[-]
Error flag for configuration settings of knock pre-processor					

ACTION_INFR_*GetVKnks*(OUT <V_knks>, IN <Cyl>, IN <Win>)

Gives back the formatted knock noise raw signal.

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit
V_knks	OUT	0...3FFH	0...4.995	5/1024	[V]
Formatted knock noise signal					
Cyl	OUT	0...7H	0...7	1	[-]
Cylinder for which the value was taken					
Win	IN	0...1H	0...1	1	[-]
Selected knock window					

ACTION_INFR_EnaKnks(IN <Enable>)

This action enables the knock strategie if Enable =1

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit
Enable	IN	0...1H	0 ... 1	1	[-]
Enable flag for knock strategie					

FUNCTION DESCRIPTION:

General information:

This strategy uses a knock signal pre-processor, which formats the knock sensor raw signal according the anti-knock strategy necessities. Figure 2 describes the functionality of this device.

The infrastructure is responsible for transmitting the programmable settings of the pre-processor calculated inside the knock aggregate and for synchronizing the internal tasks.

There are 2 configurations SENSORS-CYLINDERS possible: NC_KNKS_CYL_CONF_1 or_2 and the configuration of the engine is determined by the value C_KNKS_CYL_CONF.

Remark:

Two knock windows are created for each cylinder. The first of the two windows (Win=0) starts at least at the cylinder's TDC.
The 2nd window (Win=1) starts at least 2° after the TDC and ends at least 62° after the TDC.

The first window's length may exceed the 2° threshold, however it must not overlap the 2nd window. It has to be guaranteed by calibration that the gap between knock windows has a sufficient length.

If the beginning angle and length of the first window (Win=0) is set to 0,0, no window shall be generated and no parametrisation of the pre-processing device shall be performed.

Task synchronisation:

The task synchronisation of the anti knock strategy over °CRK is the following:

At each segment:
1. Ensure no error has occured for last transmission of the knock parameters
2. Get value of knock noise energy V_KNKS
3. Transmission of the knock window parameters for cylinder *cyl*
4. Transmission of the knock pre-processor parameters for cylinder *cyl*

At each window beginning:
5. Reset integrated noise signal

At each window end::
6. Store the value of the formatted (filtered end integrated) raw noise signal

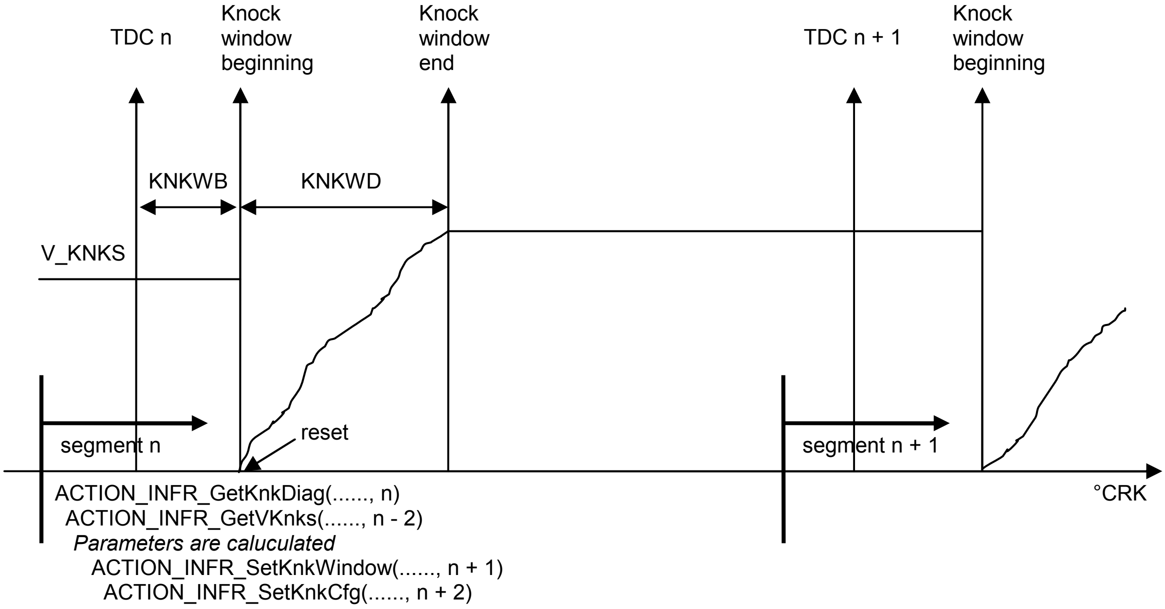


Figure 1: Synchronization of infrastructure internal tasks over °CRK

Pre-processor algorithm:

The knock pre-processor is used to provide a formatted representative of the knock sensor raw signal (see figure 2).

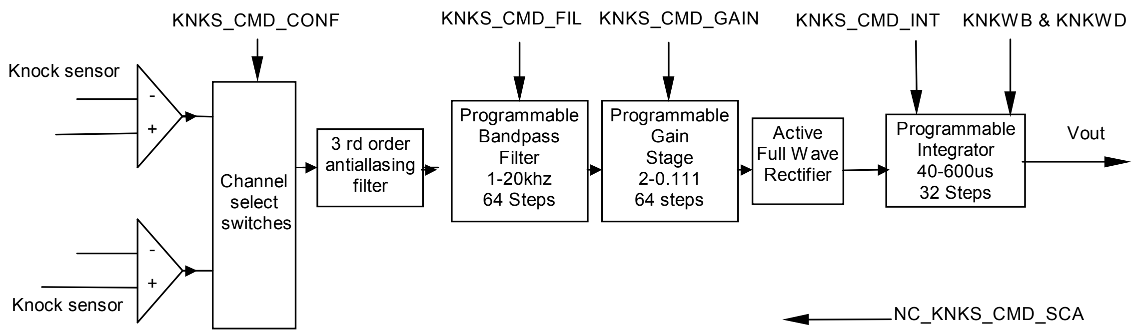


Figure 2: Simplified block of the pre-processor device

- For multi-sensor configuration: the pre-processor selects the sensors whose signal should be processed.
- A programmable band-pass filter processes the raw signal.
- A programmable gain adjusts stage is applied on the frequency band of knock
- An active full wave rectifier rectifies the signal
- After reset at window beginning, a programmable integrator integrates the signal during an angular window

The result is an analogue voltage whose output level is proportional to the engine noise. This value is called by ACTION_INFR_GetKnks.

Requirements for ACTION_INFR_SetKnkWindow:

Data acquisition

Parameter	Absolute precision	Relative precision	Resolution	Coincidence with parameter	Comment
Knkwb	+-0.375 °CRK		0.375 °CRK		TDC must be detected precisely enough
Knkwd	+-0.375 °CRK		0.375 °CRK		Gives duration of knock window with an absolute precision of 0.750°CRK
Cyl	1		1		
Win	1		1		Selected knock window, pre or normal one.

Diagnosis:	No diagnosis needed
Coincidence requirements:	It must be guaranteed, that the window configuration is transferred in time to the pre-processor for the given cylinder. For details see Figure 1.

Requirements for ACTION_INFR_SetKnkCfg:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cmd_fil		<4%	Relative resolution 1.04 to next value		**Parameter to set band pass filtering of knock raw noise** The relation between frequency freq(i) and the hexadecimal value i given by: freq(i) = freq(1) * 1.04 ^ (i-1) frequency range 1 ... 20kHz hex. range 0 ... 3FH
Cmd_gain		<9.5%	Relative resolution 0.948 to next value		**Programmable gain stage** The relation between gain and the hexadecimal value i given by: gain(i) = gain(1) * 0.948 ^ (i-1) gain range 2.00 ... 0.111 hex. range 0 ... 3FH
Cmd_int		<9%	6bit		**Programmable integrator** The relation between integration time ITC and the hexadecimal value i given by: ITC(i) = ITC(1) * 1.091 ^ (i-1) ITC range 40 ... 600µs hex. range 0 ... 3FH
Cyl	1		1		
Win	1		1		Selected knock window

Diagnosis: When setting the parameters of the pre-processor the received parameters are returned and compared with the originally send value. In case of error, ACTION_INFR_GetKnkDiag raises an error flag.

Coincidence requirements: see Figure 1.

Requirements for ACTION_INFR_GetKnkDiag:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Diag			1bit		

Diagnosis: no diagnosis done here

Coincidence requirements: When setting the pre-processor parameters using ACTION_INFR_SetKnockCfg the diagnosis result is stored. The diagnosis result of the last configuration setting is read out with this action.

Requirements for ACTION_INFR_GetVKnks:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
V_knks	50mV		10bit		
Cyl	1		1		
Win	1		1		Selected knock window

Diagnosis: no diagnosis done here

Coincidence requirements: The integrated signal value of the pre-processor must be read out and copied in time, see figure 1.

Requirements for ACTION_INFR_EnaKnks:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Enable	1		1		

Diagnosis: no diagnosis done here

Coincidence requirements: this action should be called to enable knock acquisition when engine is synchronised

45.3 Application incidences for knock control and measurement CAN-interface

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
KNK_CTL_DIS	V/O	0...1H	0...1	1	[-]
knock control disabled due to malfunctions					

Input data:

LV_ERR_CRK	LV_ERR_SPI_KNK	LV_ERR_KNKS[NC_NR_SENS_KNK]	IGA_ADJ_KNK[NC_CYL_NR]
N	LV_KNK	LV_IGK	LV_ES
IGA_KNK[NC_CYL_NR]	IGA[NC_CYL_NR]	NL[NC_CYL_NR]	KNKS[NC_CYL_NR]
CYL_ID_KNK	SEG_NR	MAF	

45.3.1 Inhibition of knock control

General information:

To activate the knock control, a macro function (KNK_CTL_DIS) is executed which checks if knock control is disabled due to malfunctions. If KNK_CTL_DIS = 0 the conditions are fulfilled and the knock control is enabled.

Application conditions:

Initialisation: at reset **or** LV_ES 0->1:

If LV_ERR_CRK = 0 **and** LV_ERR_KNKS[NC_NR_SENS_KNK] = 0 **and** LV_ERR_SPI_KNK = 0

Then KNK_CTL_DIS = 0

Recurrence: every segment

Activation: LV_ES = 0

Formula section:

```
If      LV_ERR_CRK = 1     or
        LV_ERR_KNKS[NC_NR_SENS_KNK] = 1  or
        LV_ERR_SPI_KNK = 1
Then    KNK_CTL_DIS = 1
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_IGA_LGRD_3	1	0...7FFFH	0...47.99853	1.46E-03	[°CRK]
gradient limitation when underspending knock load threshold					

45.3.2 Excursion Limit Analysis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_IGA_DIF_MAX_KNK[NC_CYL_NR]	V/S	0...FFH	0...255	1	[-]
Counter for number of occured excursion limit events (cylinder individual)					
N_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	V/S	0...FFH	0...8160	32	[rpm]
Freeze frame for engine speed when excursion limit event occurs					
MAF_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	V/S	0...FFH	0...1389	5.4470588	[mg/stk]
Freeze frame for mass air flow when excursion limit event occurs					
TCO_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	V/S	0...FEH	-48...142.5	0.75	[°C]
Freeze frame for coolant temperature when excursion limit event occurs					
TIA_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	V/S	0...FEH	-48...142.5	0.75	[°C]
Freeze frame for temperatur of injected air when excursion limit event occurs					
CYL_ID_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]	V/S	0...FFH	0...255	1	[-]
Freeze frame for knocking cylinder identification when excursion limit event occurs					

Input data:

NC_CYL_NR	LC_AD_CLR_RON	LV_IGA_DIF_MAX_KNK	CYL_ID_KNK
N_32	MAF_HB	TCO	TIA

FUNCTION DESCRIPTION:

General information:

'Control Excursion Limit' in the 'Knock Control' module is used to limit the difference of the ignition adjustment angle through knock control of the single cylinders to a certain value.
When a limitation on one cylinder is done, LV_IGA_DIF_MAX_KNK is set in order to give this information to the 'Excursion Limit Analysis' functionality.
Here the values for the concerning cylinder and actual operation point are stored.

Graphical interpretation:

- Cylinder individual counter CTR_IGA_DIF_MAX_KNK[NC_CYL_NR]

```
input  |_CTR_|
   --> |______|  (for CYL_ID_KNK = 0)
   --> |______|  (for CYL_ID_KNK = 1)
   --> |______|  (for CYL_ID_KNK = 2)
   --> |______|  (for CYL_ID_KNK = 3)
   --> |______|  (for CYL_ID_KNK = 4)
   --> |______|  (for CYL_ID_KNK = 5)
         :
       NC_CYL_NR rows for counting excursion limit events on each cylinder individually
```

- Freezed operation point data and id of concerning cylinder
  (shifting register with newest values on top position)
  N_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD],
  MAF_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD],
  TCO_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD],
  TIA_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD],
  CYL_ID_IGA_DIF_MAX_KNK[NC_NR_FRF_KNK_RTD]

```
input  |__N___|_MAF_|_TCO_|_TIA__|Cyl_ID_|
  -->  |______|______|______|______|______| ---┐ (newest dataset)
       |______|______|______|______|______| <-┘ (2nd recent dataset)
       |______|______|______|______|______| <-┘ (3rd recent dataset)
       |______|______|______|______|______| <-┘ (4th recent dataset)
       |______|______|______|______|______| <-┘ (5th recent dataset)
       |______|______|______|______|______| <-┘ (6th recent dataset)
       :
       NC_NR_FRF_KNK_RTD rows for data freeze at events on all cylinders
```

Application conditions:

Initialisation: CTR_IGA_DIF_MAX_KNK[0 to NC_CYL_NR - 1],
N_IGA_DIF_MAX_KNK[0 to NC_NR_FRF_KNK_RTD - 1],
MAF_IGA_DIF_MAX_KNK[0 to NC_NR_FRF_KNK_RTD - 1],
TCO_IGA_DIF_MAX_KNK[0 to NC_NR_FRF_KNK_RTD - 1],
TIA_IGA_DIF_MAX_KNK[0 to NC_NR_FRF_KNK_RTD - 1],
CYL_ID_IGA_DIF_MAX_KNK[0 to NC_NR_FRF_KNK_RTD - 1]
get values from E2PROM at reset

Recurrence: every segment

Activation: at every engine state except engine stop

Deactivation: at engine stop

Formula section:

Module has to be executed
- after calculation of 'Control Excursion Limit' module
- within the same SEG_NR.

fix number of 'freeze frames' with respect to storage space:
NC_NR_FRF_KNK_RTD = 60

cylinder individual counter for recognition of reaching excursion limit:

```
If    LV_IGA_DIF_MAX_KNK = 1 And $CTR\_IGA\_DIF\_MAX\_KNK[CYL\_ID\_KNK]_n < 255$
Then  $CTR\_IGA\_DIF\_MAX\_KNK[CYL\_ID\_KNK]_n = CTR\_IGA\_DIF\_MAX\_KNK[CYL\_ID\_KNK]_{n-1} + 1$
Endif
```

overwriting old values in 'freeze frames': data are shifted, latest values are stored

```
If      LV_IGA_DIF_MAX_KNK = 1
Then    For   i = NC_NR_FRF_KNK_RTD - 1 To 1 Step -1
                N_IGA_DIF_MAX_KNK[i] = N_IGA_DIF_MAX_KNK[i-1]
                MAF_IGA_DIF_MAX_KNK[i] = MAF_IGA_DIF_MAX_KNK[i-1]
                TCO_IGA_DIF_MAX_KNK[i] = TCO_IGA_DIF_MAX_KNK[i-1]
                TIA_IGA_DIF_MAX_KNK[i] = TIA_IGA_DIF_MAX_KNK[i-1]
                CYL_ID_IGA_DIF_MAX_KNK[i] = CYL_ID_IGA_DIF_MAX_KNK[i-1]
        Next i
        N_IGA_DIF_MAX_KNK[0] = N_32
        MAF_IGA_DIF_MAX_KNK[0] = MAF_HB
        TCO_IGA_DIF_MAX_KNK[0] = TCO
        TIA_IGA_DIF_MAX_KNK[0] = TIA
        CYL_ID_IGA_DIF_MAX_KNK[0] = CYL_ID_KNK
Endif
```

N.B.: i represents identification index of stored info data set
(shifting in 'For' loop has not to be executed, if NC_NR_FRF_KNK_RTD = 1)
(efficiency of CPU load could be optimized, if only 'filled' freeze frames are shifted)

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_FRF_KNK_RTD	1	0...FFH	1...256	1	[-]
Number of freeze frame datasets for storing operation point and cylinder number for excursion limit analysis					

45.3.3 Knock Control Measurement CAN-Interface

General information:

This chapter is used as interface between the variables on the local CAN bus of the knock control measurement with CAN-interface and the system variables.

The process is as followed:

a.) start the output of systemvalues

b.) output of the systemvalue

c.) stop the output of systemvalues

45.3.3.1 CAN Protocol

Description:

The following control units participate to CAN-communication:
DME: Engine management system
IAV: Measurement equipment from BMW

Baudrate: 500 kBaud

The IAV sends for starting its measurement values on CAN Messages the ID = **700H.**

The DME sends its measurement concerning the IAV interface on CAN Messages with the ID = **701H and 702H.**

The IAV sends for stopping its measurement values on CAN Messages the ID = **70AH.**

45.3.3.2 Send messages by DME ID 701H and ID702H

ID 701H

Byte No.	Signal Name Siemens	Byte Name Of Customer	Bit No.	Bit Name of Customer	Function Description
0	SEG_NR		0 - 7		segment number
1	CYL_ID_KNK		0 - 7		cylinder identification number
2	KNKS[NC_CYL_NR] (low Byte)		0 - 7		knock value
3	KNKS[NC_CYL_NR] (high Byte)		0 - 7		knock value
4	NL[NC_CYL_NR] (low Byte)		0 - 7		noise level
5	NL[NC_CYL_NR] (high Byte)		0 - 7		noise level
6	IGA[NC_CYL_NR]		0 - 7		ignition of cylinder x
7	IGA_KNK[NC_CYL_NR]		0 - 7		knock ignition value

ID 702H

Byte No.	Signal Name Siemens	Byte Name Of Customer	Bit No.	Bit Name of Customer	Function Description
0	SEG_NR		0 - 7		segment number
1	LV_KNK		0 - 7		state of knock control
2	IGA_ADJ_KNK[NC_CYL_NR]		0 - 7		jerk ignition of knock control
3	FF		0 - 7		dummy value fix
4	N (low Byte)		0 - 7		engine speed
5	N (high Byte)		0 - 7		engine speed
6	MAF (low Byte)		0 - 7		mass air flow
7	MAF (high Byte)		0 - 7		mass air flow

Application conditions:

Initialisation:

Recurrence: segment synchronous

Activation: LV_IGK = 1

The resolution and the Hex-limits for the values within the message is equal to the defined system quantity.

45.3.3.3 Received messages by the DME ID 700H and ID70AH

General information:

This message starts or stops the data transfer from the ECU to the IAV.

Application conditions:

Recurrence: 10 ms

Activation: LV_IGK = 1

ID 700H

Byte No.	Signal Name Siemens	Byte Name Of Customer	Bit No.	Bit Name of Customer	Function Description

ID 70AH

Byte No.	Signal Name Siemens	Byte Name Of Customer	Bit No.	Bit Name of Customer	Function Description

45.4 Knock Control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_KNK	O/V	0...FFH	0...1389	5.4470588	[mg/stk]
Mass air flow with hysteresis					
N_KNK	V	0...FFH	0...8160	32	[rpm]
Engine speed with hysteresis					
LV_KNK	O/V	0...1H	0...1	1	[-]
Logical variable for recognition of knock					
IGA_DEC_KNK	-	1...7FH	0.375...47.625	0.375	[°CRK/cyc]
Spark retard at recognised knocking					
IGA_ADJ_KNK[NC_CYL_NR]	O/V	0...80H	-48...0	0.375	[°CRK]
Total spark retard through knock control with adaptation (cylinder individual)					
IGA_KNK_BAS[NC_CYL_NR]	-	0...80H	-48...0	0.375	[°CRK]
Spark retard through knock control prior adaptation (cylinder individual)					
IGA_ADJ_MAX_KNK	V	0...FFH	-35.625...60	0.375	[°CRK]
Total spark retard through knock control including adaptation (cylinder individual)					
FAC_IGA_MAX_KNK	V	0...FFH	0...0.99609	3.9063e-3	[-]
Weighting factor for the max. spark retard against coolant					
LV_KNK_PAS_TRAN_ACT	-	0...1H	0...1	1	[-]
Transition for knock control passive to active					
LV_KNK_CTL_ENA	O/V	0...1H	0...1	1	[-]
Boolean for knock control enabled					
CYL_ID_KNK	O/V	0...FFH	0...255	1	[-]
Cylinder number for knock control					
KNK_THD[NC_CYL_NR]	V	0...FFH	0...4.98046	0.0195313	[V]
Knock Threshold cylinder x					
IGA_KNK[NC_CYL_NR]	O/V	0...80H	-48...0	0.375	[°CRK]
Spark retard due to knocking combustion					
KNK_AD_SF[NC_CYL_NR]	-	0...1H	0...1	1	[-]
Flag for transition from passive to active					
IGA_MV_KNK	V	0...80H	-48...0	0.375	[°CRK]
Mean value of actual retard adjustment without adaptation values					
IGA_MV_ADJ_KNK	O/V	0...80H	-48...0	0.375	[°CRK]
Meanvalue of actual retard adjustment with adaptation values					
IGA_MV_CBK_ADJ_KNK[NC_CBK_EX_NR]	O/V	0...80H	-48...0	0.375	[°CRK]
Meanvalue of actual retard adjustment with adaptation values bank selective					
LV_KNK_TRA_MAF	V	0...1H	0...1	1	[-]
Logical variable for Load Dynamic Function active or passive					
KNKWB[NC_CYL_NR]	O/V	0...FFH	0...95.625	0.375	[°CRK]
Beginning of Measurement Window Cyl.x					
KNKWB_PRE[NC_CYL_NR]	O/V	0...FFH	0...95.625	0.375	[°CRK]
Beginning of Measurement Window Cyl.x first window					
KNKWE[NC_CYL_NR]	O/V	0...FFH	0...95.625	0.375	[°CRK]
End of Measurement Window Cyl.x					
KNKWE_PRE[NC_CYL_NR]	O/V	0...FFH	0...95.625	0.375	[°CRK]
End of Measurement Window Cyl.x first window					
KNKS_CMD_CONF[NC_CYL_NR]	O	0...FFH	0...255	1	[-]
Configuration value for the analogue channel of knock sensor					
KNKS_CMD_FIL[NC_CYL_NR]	O/V	0...3FH	0...63	1	[-]
Knock Filter Frequency and Reference Filter Frequency Cyl.x					
KNKS_CMD_FIL_PRE[NC_CYL_NR]	O/V	0...3FH	0...63	1	[-]
Knock Filter Frequency and Reference Filter Frequency Cyl.x					
KNKS_CMD_INT[NC_CYL_NR]	O/V	0...1FH	0...31	1	[-]
Integration Time Constant Cyl.x					
KNKS_CMD_INT_PRE[NC_CYL_NR]	O/V	0...1FH	0...31	1	[-]
Integration Time Constant Cyl.x					

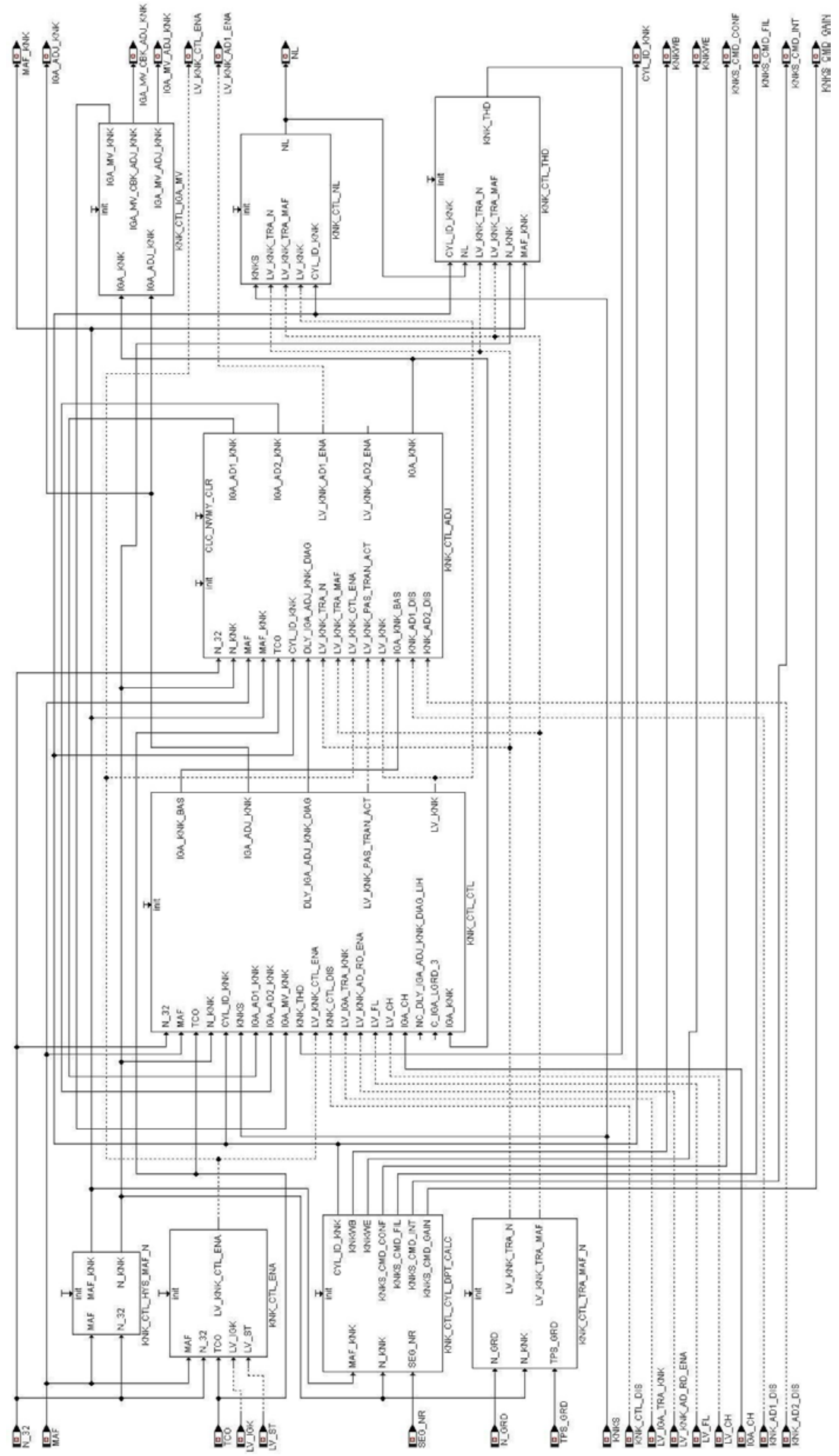
KNKS_CMD_GAIN[NC_CYL_NR]	O/V	0...3FH	0...63	1	[-]
Amplification precontrol Cyl.x					
KNKS_CMD_GAIN_PRE[NC_CYL_NR]	O/V	0...3FH	0...63	1	[-]
Amplification precontrol Cyl.x					
LV_KNK_TRA_N	V	0...1H	0...1	1	[-]
Logical variable for Engine Speed Dynamic Function active or passive					
NL[NC_CYL_NR]	O/V	0...FFFFH	0...4.99992	0.0763e-3	[V]
Floating mean value of the noise signal					
LV_CTR_KNK_CMB	V	0...1H	0...1	1	[-]
Logical variable to enable counter after combustion mode change					
CTR_KNK_CMB	V	0...FFH	0...255	1	[-]
Counter to disable KNK control after combustion mode switch					
STATE_CMB_CTL_TMP	V	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H	HOM_AFS AFS_TO_AFL HOM_AFL AFL_TO_AFS HOM_TO_HOMS HOMS BACKS S S_TO_HOMS BACKHOM	1	[-]
States of the combustion management					
STATE_CMB_CTL_KNK	V	0...8H	0...8	1	[-]
States of the combustion management to be saved for further use					
LV_IGA_DIF_MAX_KNK	O/V	0...1H	0...1	1	[-]
Indicator for excursion limit analysis (knock ignition angle of one cylinder is at maximum deviation limit from mean value)					
LV_KNK_CTL_PRE_ENA	O/V	0...1H	0...1	1	[-]
Boolean for knock control pre-ignition enabled					
LV_KNKWE_PRE_GAP	V	0...1H	0...1	1	[-]
Boolean for indication of pre-window restriction, caused by hardware required gap for save operation					
LV_KNK_PRE	V	0...1H	0...1	1	[-]
Logical variable for recognition of knock in first window					
LV_KNK_PRE_DET	V	0...1H	0...1	1	[-]
Logical variable for pre-ignition recognition by algorithm					
CYCNR_KNK_PRE_DET	V	0...FFFFH	0...65535	1	[-]
Cycle counter for observation period pre-ignition					
CTR_KNK_PRE_DET[NC_CYL_NR]	V	0...FFFFH	0...65535	1	[-]
Detection counter for evaluated knock and pre-ignition events					
CYCNR_INH_IV_KNK_PRE_DET[NC_CYL_NR]	V	0...FFFFH	0...65535	1	[-]
Cycle counter for duration of injection shut off by pre-ignition detection					
INH_IV_KNK	O/V	0...FFH	0...255	1	[-]
Ignition shut off pattern from knock control pre-ignition					
KNK_THD_PRE	V	0...FFH	0...4.998	0.0195313	[V]
Knock Threshold cylinder x in pre-window					
LV_KNK_CTL_PRE_INH	V	0...1H	0...1	1	[-]
Logical variable for complete inhibition of knock control pre-ignition					
LV_KNK_MAX	V	0...1H	0...1	1	[-]
Logical variable for recognition of knock above maximum threshold					
KNK_THD_MAX	V	0...FFH	0...4.998	0.0195313	[V]
Maximum knock threshold in main window					

Input data:

MAF	N_32	TCO	LV_IGK
LV_ST	IGA_CH	KNKS[NC_CYL_NR]	C_IGA_LGRD_3

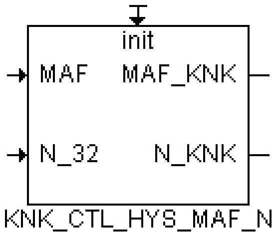
KNK_CTL_DIS	IGA_WOUT_KNK[NC_CYL_NR]	SEG_NR	LV_IGA_TRA_KNK
N_GRD	LV_FL	LV_CH	LV_HOM_ACT
OPM_AV	LV_ERR_KNKS[NC_NR_SENS_KNK]	KNKS_PRE[NC_CYL_NR]	MFF_SP_MV

45.4.1 Overview



45.4.2 Pre-processing

45.4.2.1 Hysteresis Function for MAF and N_32



General information:

Hysteresis functions for engine speed and air-mass changes are defined for the tables relevant for the knock control in order to avoid steps recognisable for the driver due to small changes of the operation point.

Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	at engine stop
Initialization:	MAF_KNK = N_KNK = 0 at reset and engine running to engine stop
Update Rate:	every segment

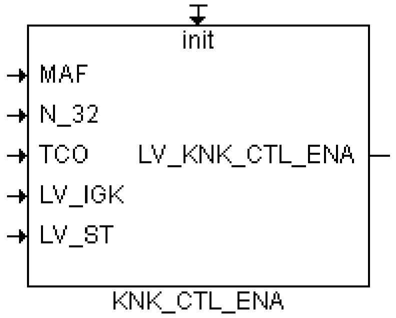
Formula section:

```
if    |(N_32 - N_KNK)|    >=    C_N_IGA_HYS_KNK
      N_KNK = N_32
else
      N_KNK remains unchanged
end if

if    |(MAF - MAF_KNK)|  >=    C_MAF_IGA_HYS_KNK
      MAF_KNK = MAF
else
      MAF_KNK remains unchanged
end if
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_IGA_HYS_KNK	1	0...FFH	0...8160	32	[rpm]
Speed Hysteresis for Knock Control					
C_MAF_IGA_HYS_KNK	1	0...FFH	0...1389	5.45E+00	[mg/stk]
Air-Mass Hysteresis for Knock Control					

45.4.2.2 Knock Control Enable



Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	at engine stop
Initialization:	LV_KNK_CTL_ENA = 0 at reset and engine running to engine stop LV_KNK_CTL_PRE_ENA = 0 at reset and engine running to engine stop LV_CTR_KNK_CMB = 0 reset and engine running to engine stop CTR_KNK_CMB = 0 reset and engine running to engine stop
Update Rate:	every segment

Formula section:

knock control enable:

```
if        (LV_ST = 1  or  LV_IGK = 0)
then      LV_KNK_CTL_ENA = 0
else
    if    (MAF_KNK  >= IP_MAF_MIN_KNK__N__TCO)
          LV_KNK_CTL_ENA = 1
    else
          LV_KNK_CTL_ENA = 0
    end if
end if
```

knock control pre-ignition enable:

```
if        LV_KNK_CTL_ENA = 1 and MAF_KNK >= IP_MAF_MIN_KNK_PRE
then      LV_KNK_CTL_PRE_ENA = 1
else      LV_KNK_CTL_PRE_ENA = 0
end if
```

inhibit knock control pre-ignition:

```
if        N_32 > C_N_KNK_CTL_PRE_INH
then      LV_KNK_CTL_PRE_INH = 1
else      LV_KNK_CTL_PRE_INH = 0
end if
```

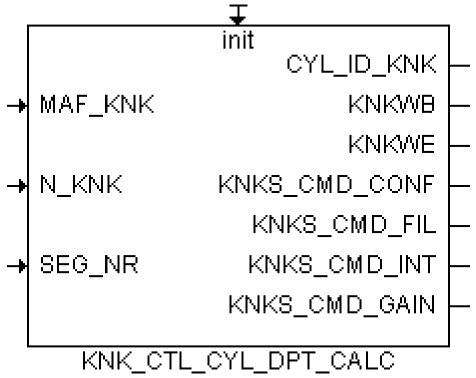
If the relevant air-mass is below the air-mass threshold (LV_KNK_CTL_ENA = 0), only the noise value calculation is executed.
An additional air-mass threshold exists for enabling cylinder cut off algorithm (LV_KNK_CTL_PRE_ENA).

A complete inhibition of pre-window calculation and consideration is forced (LV_KNK_CTL_PRE_INH = 1), when pre-window is initialized with 0 (see IRS); especially when engine speed is above a calibratable threshold.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_MAF_MIN_KNK__N__TCO	8*4	0...FFH	0...1389	5.45E+00	[mg/stk]
LDPM_N_3_9	8	0...FFH	0...8160	32	[rpm]
LDP_TCO__MAF_MIN_KNK	4	0...FEH	-48...142.5	0.75	[°C]
Air-mass Threshold Release KNK					
IP_MAF_MIN_KNK_PRE	8	0...FFH	0...1389	5.4470588	[mg/stk]
LDPM_N_3_9	8	0...FFH	0...8160	32	[rpm]
Air-mass Threshold Release KNK Pre-Ignition					
C_N_KNK_CTL_PRE_INH	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for complete inhibition of knock control pre-ignition					

45.4.2.3 Cylinder Dependent Calculations



45.4.2.3.1 Cylinder Number for Knock Control

Application conditions:

Activation: at every engine state except engine stop

Deactivation: at engine stop

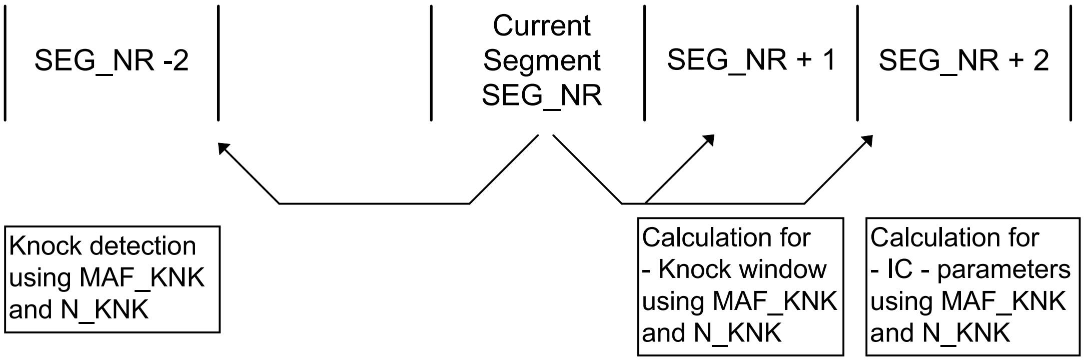
Initialization: CYL_ID_KNK = 0 at reset and engine running to engine stop

Update Rate: every segment

Formula section:

CYL_ID_KNK = SEG_NR - 2

Description:



45.4.2.3.2 Knock Measurement Window

General information:

The knock detection is performed for cylinder-individually defined measurement windows since knock-typical oscillations only occur for certain crankshaft angles. The measurement windows (crank angle of beginning and end) are specified as function of the engine speed and load.
If the estimated ‘End of the Knock Measurement Window’ is lower or equal compared to the ‘Beginning of the Knock measurement Window’, both values are initialized.

The crank angle for the beginning of the knock measurement window is taken for each cylinder from the respective table IP_KNKWB[x] according to the firing order.
The crank angle for the end of the knock measurement window is taken for each cylinder from the respective table IP_KNKWE[x] according to the firing order.

Apart from window settings for AFS (stoichiometric) separate windows are existing in operation mode AFL (lean) and S (stratified).

Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	at engine stop
Initialization:	at reset and engine running to engine stop
	KNKWB[0 to NC_CYL_NR] = NC_KNKWB_INI
	KNKWE[0 to NC_CYL_NR] = NC_KNKWE_INI
	KNKWB_PRE[0 to NC_CYL_NR] = NC_KNKWB_PRE_INI
	KNKWE_PRE[0 to NC_CYL_NR] = NC_KNKWE_PRE_INI
Update Rate:	every segment

Formula section:

Definition of cylinder dependent knock window ranges:

RNG_WB = (720° / NC_CYL_NR) - 12°
RNG_WB_HEX = HEX(RNG_WB / 0.375°)
RNG_WE = (720° / NC_CYL_NR) - 6°
RNG_WE_HEX = HEX(RNG_WE / 0.375°)

```
if      OPM_AV = "AFS"
then    KNKWB[x] = IP_KNKWB[x]
        KNKWE[x] = IP_KNKWE[x]
else    KNKWB[x] = IP_KNKWB_ AFL
        KNKWE[x] = IP_KNKWE_AFL
end if

if      (KNKWE[x] ≤ KNKWB[x])
then    KNKWB[x] = NC_KNKWB_INI
        KNKWE[x] = NC_KNKWE_INI
end if

if      LV_KNK_CTL_PRE_INH = 0
then    KNKWB_PRE[x] = IP_KNKWB_PRE[x]
        KNKWE_PRE[x] = IP_KNKWE_PRE[x]

        if      KNKWE_PRE[x] + IP_KNKWE_PRE_GAP >= KNKWB[x]
        then    KNKWE_PRE[x] = KNKWB[x] - IP_KNKWE_PRE_GAP
                LV_KNKWE_PRE_GAP = 1
        else    LV_KNKWE_PRE_GAP = 0
        end if

        if      (KNKWE_PRE[x] ≤ KNKWB_PRE[x])
        then    LV_KNK_CTL_PRE_INH = 1
        end if
endif

if      LV_KNK_CTL_PRE_INH = 1
then    KNKWB_PRE[x] = NC_KNKWB_PRE_INI
        KNKWE_PRE[x] = NC_KNKWE_PRE_INI
endif
```

call action interface for pre and normal window

ACTION_INFR_*SetKnkWindow*(IN <Knkwb>, IN <Knkwd>, IN <Cyl>, IN <Win>)
Sets knock window for cylinder *cyl*

N.B.: x represents current SEG_NR + 1

Important information for calibration:

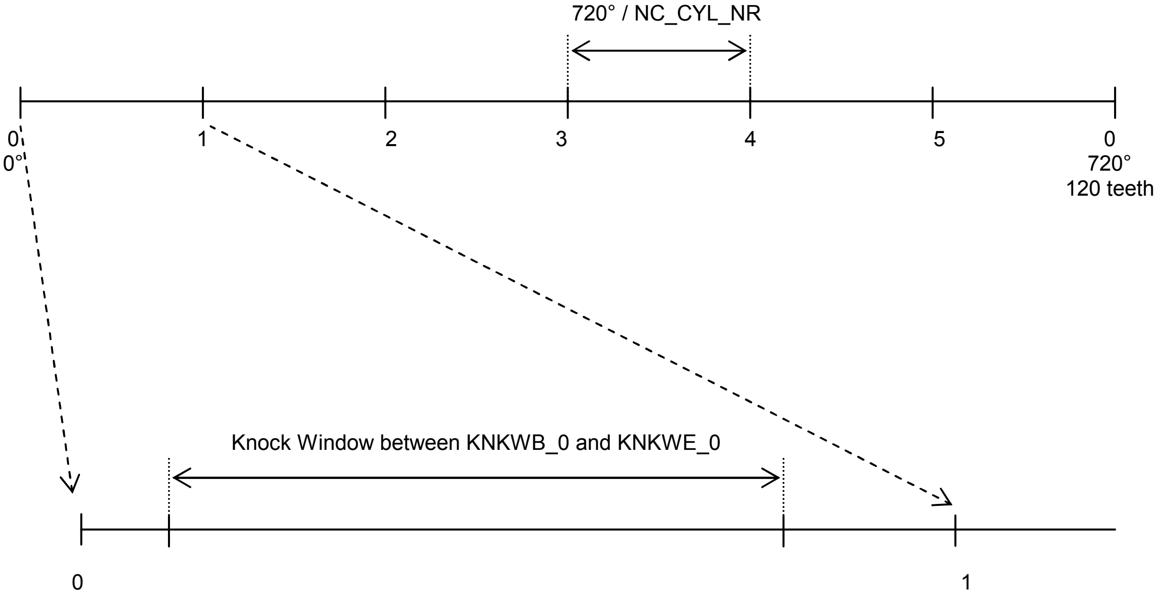
In order to have the possibility of two windows (pre- and main) in one segment, BSW is calculating with double number of segments with half of the original length. Every window begin (pre- or main) must lie within one of such doubled segments. The beginning of a segment for the main window is fixed in BSW to 2°CRK after TDC. Due to that the main knock window KNKWB[x] must be calibrated not before 2°CRK.

IP_KNKWE_PRE_GAP calibration must be done before KNKS_PRE signal parameter calibration (window, gain, filter frequency, integration time constant), in order to assure right calculated and transferred values from the knock IC and not to have different values because of new window length after changing this restriction. The duration of the gap between KNKWE_PRE and KNKWB is determined by the SPI bus communication. During the gap the integrator value of the pre-window has to be read and the ATIC62 device has to be programmed with the main window parameters (window, gain, filter frequency, integration time constant). The required duration has to be determined system individually by the BSW SW department.
Special restrictions must be considered, if distance between 0°CRK (minimum possible value for pre-window begin and end; also when set inactive a activation peak is remaining) and main window begin should be smaller than this determined gap or a smaller gap in general should be used. Then especially the same parametrisation for pre- and main window is required and no inhibition for pre-window calculation is allowed. Thus in BSW right parameters for main window are already set during pre-window parametrisation.

To have low CPU load, the knock pre-ignition window can be set to 0 by LV_KNK_CTL_PRE_INH = 1. This is done either above a calibratable engine speed threshold or at implausibility, when pre-window end is earlier than pre-window begin. KNKWB_PRE and KNKWE_PRE values are initialized with the NC values 0. Then no parametrisation and knock IC processing is executed (see IRS). Also the detection algorithm and KNK_THR_PRE aren't executed any longer.

Description:

(example for NC_CYL_NR=6; TDC offset = 0°CRK)



Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_KNKWB[NC_CYL_NR]	10*4	0...FFH	0...95.625	0.375	[°CRK]
LDPM_N_KNKWB	10	0...FFH	0...8160	32	[rpm]
LDPM_MAF_KNKWB	4	0...FFH	0...1389	5.4470588	[mg/stk]
Beginning of Measurement Window Cyl.x					
IP_KNKWE[NC_CYL_NR]	10*4	0...FFH	0...95.625	0.375	[°CRK]
LDPM_N_KNKWB	10	0...FFH	0...8160	32	[rpm]
LDPM_MAF_KNKWB	4	0...FFH	0...1389	5.4470588	[mg/stk]
End of Measurement Window Cyl.x					
IP_KNKWB_PRE[NC_CYL_NR]	10*4	0...FFH	0...95.625	0.375	[°CRK]
LDPM_N_KNKWB	10	0...FFH	0...8160	32	[rpm]
LDPM_MAF_KNKWB	4	0...FFH	0...1389	5.4470588	[mg/stk]
Beginning of Measurement Window Cyl.x previous					
IP_KNKWE_PRE[NC_CYL_NR]	10*4	0...FFH	0...95.625	0.375	[°CRK]
LDPM_N_KNKWB	10	0...FFH	0...8160	32	[rpm]
LDPM_MAF_KNKWB	4	0...FFH	0...1389	5.4470588	[mg/stk]
End of Measurement Window Cyl.x previous					
IP_KNKWE_PRE_GAP	4	0...FFH	0...95.625	0.375	[°CRK]
LDP_N_KNKWE_PRE_GAP	4	0...FFH	0...8160	32	[rpm]
Gap restricting Knock Pre-Window for save IC operation					
IP_KNKWB_AFL	10*4	0...FFH	0...95.625	0.375	[°CRK]
LDPM_N_KNKWB	10	0...FFH	0...8160	32	[rpm]
LDPM_MFF_1_9	4	0...FFFFH	0...1389	0.0211948	[mg/stk]
Beginning of Measurement Window for all Cyl. in AFL or S					
IP_KNKWE_AFL	10*4	0...FFH	0...95.625	0.375	[°CRK]
LDPM_N_KNKWB	10	0...FFH	0...8160	32	[rpm]
LDPM_MFF_1_9	4	0...FFFFH	0...1389	0.0211948	[mg/stk]
End of Measurement Window for all Cyl. in AFL or S					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_KNKWB_INI	1	0...FFH	0...95.625	0.375	[°CRK]
Beginning of Measurement Window Cyl.x					
NC_KNKWE_INI	1	0...FFH	0...95.625	0.375	[°CRK]
End of Measurement Window Cyl.x					
NC_KNKWB_PRE_INI	1	0...FFH	0...95.625	0.375	[°CRK]
Beginning of Measurement Pre-Window Cyl.x					
NC_KNKWE_PRE_INI	1	0...FFH	0...95.625	0.375	[°CRK]
End of Measurement Pre-Window Cyl.x					

45.4.2.3.3 Assignment of cylinder number to Knock sensor diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
KNKS_1_DIAG	O	0...FFH	0...255	1	[-]
Assignment of cylinder number to knock sensor 1 (diagnosis only)					
KNKS_2_DIAG	O	0...FFH	0...255	1	[-]
Assignment of cylinder number to knock sensor 2 (diagnosis only)					

Input data:

LV_ES	NC_KNKS_CONF		

FUNCTION DESCRIPTION:

The physically defined cylinder number is defined in firing oder 1 5 3 6 2 4 for a in-line 6 cyl. engine. There is the possibility to define especially for the knock sensor diagnosis which cylinder should be used for the knock sensor diagnosis via the help of the calibration constants C_KNKS_1/_2_DIAG. Only 'loud' cylinders should be used to detect a malfunktion of the Knock sensor signal.

The maximum possible number of cylinders is defined with NC_KNKS_CONF. Only cylinders can be used which are assigned for the knock detection of the individual knock sensor.

e.g. in-line 6 cyl. engine:

logically:	0	1	2	3	4	5	
physically:	1	5	3	6	2	4	
assignment KR	0	1	0	1	0	1	NC_KNKS_CONF = 2A hex
assignment KNKS1	1	0	1	0	1	0	C_KNK_1_DIAG = 15 hex
assignment KNKS2	0	1	0	1	0	1	C_KNK_2_DIAG = 2A hex

Application conditions:

Initialisation: all = 0 at reset and engine running to engine stop

Recurrence: segment synchronous

Activation: all engine states except LV_ES

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_KNKS_1_DIAG	1	0...FFH	0...255	1	[-]
Assignment of cylinder number to knock sensor 1 (diagnosis only)					
C_KNKS_2_DIAG	1	0...FFH	0...255	1	[-]
Assignment of cylinder number to knock sensor 2 (diagnosis only)					

45.4.2.3.4 Calibration of ATIC62 device

FUNCTION DESCRIPTION:

General information:

The knock control in the engine management works with the ATIC62 device. The output value of the integrator after the end of the measurement window is fed as Knock Value KNKS through an analogue channel to the control algorithm.
(Further information may be obtained in "Applications Guide to ATIC62 ".)

Apart from parameters for AFS (stoichiometric) separate sets of parameters are existing in operation mode AFL (lean) and S (stratified).

Application conditions:

Activation: at every engine state except engine stop

Deactivation: at engine stop

Initialization:
KNKS_CMD_INT[0 to NC_CYL_NR] = 0
KNKS_CMD_FIL[0 to NC_CYL_NR] = 0
KNKS_CMD_GAIN[0 to NC_CYL_NR] = 0

KNKS_CMD_INT_PRE[0 to NC_CYL_NR] = 0
KNKS_CMD_FIL_PRE[0 to NC_CYL_NR] = 0
KNKS_CMD_GAIN_PRE[0 to NC_CYL_NR] = 0

KNKS_CMD_CONF[0 to NC_CYL_NR] = value of configuration bit at position x in NC_KNKS_CONF

at reset and engine running to engine stop

Update Rate: every segment

Formula section:

If OPM_AV = "AFS"

Then KNKS_CMD_INT[x] = ID_ITC_KNK[x]
KNKS_CMD_FIL[x] = ID_FIL_FRQ_KNK[x]
KNKS_CMD_GAIN[x] = IP_GAIN_KNK[x]

KNKS_CMD_INT_PRE[x] = ID_ITC_KNK_PRE[x]
KNKS_CMD_FIL_PRE[x] = ID_FIL_FRQ_KNK_PRE[x]
KNKS_CMD_GAIN_PRE[x] = IP_GAIN_KNK_PRE[x]

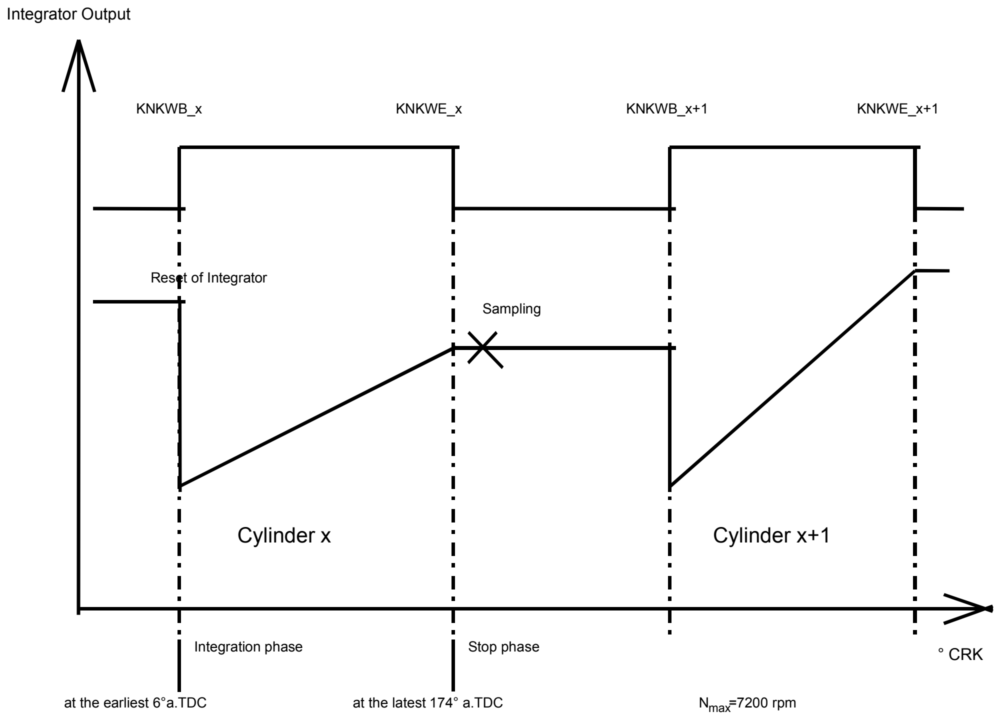
```
Else KNKS_CMD_INT[x]          =    C_ITC_KNK_AFL[x]
     KNKS_CMD_FIL[x]          =    C_FIL_FRQ_KNK_AFL[x]
     KNKS_CMD_GAIN[x]         =    IP_GAIN_KNK_AFL[x]

     KNKS_CMD_INT_PRE[x]      =    C_ITC_KNK_AFL[x]
     KNKS_CMD_FIL_PRE[x]      =    C_FIL_FRQ_KNK_AFL[x]
     KNKS_CMD_GAIN_PRE[x]     =    IP_GAIN_KNK_AFL[x]

Endif
```

call action interface for pre and normal window parameters

ACTION_INFR_*SetKnkCfg*(IN <Cmd_fil>, IN <Cmd_gain>, IN <Cmd_int>, IN <Cyl>, IN <Win>)
Sets knock pre-processor window, filter, gain and integrator factors for cylinder *cyl*

N.B.: x represents current SEG_NR + 2

Description (example for 4 cylinder engine):



Note: These calibration data have to be adjusted if the processor sampling rate is changed.

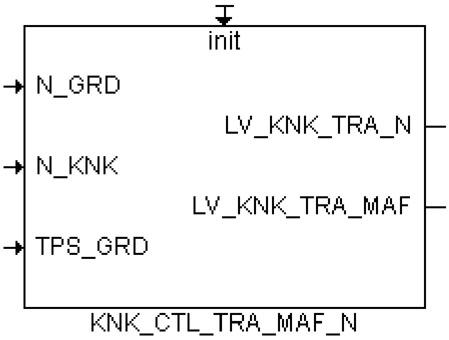
Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_FIL_FRQ_KNK[NC_CYL_NR]	4	0...3FH	0...63	1	[-]
LDPM_N_FIL_FRQ_KNK	4	0...FFH	0...8160	32	[rpm]
Knock Filter Frequency and Reference Filter Frequency Cyl.x					
ID_ITC_KNK[NC_CYL_NR]	16*4	0...1FH	0...31	1	[-]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
LDPM_MAF_1_9	4	0...FFH	0...1389	5.45E+00	[mg/stk]
Integration Time Constant Cyl.x					
IP_GAIN_KNK[NC_CYL_NR]	16*12	0...3FH	0...63	1	[-]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
LDPM_MAF_GAIN_KNK	12	0...FFH	0...1389	5.4470588	[mg/stk]
Amplification precontrol Cyl.x					
ID_FIL_FRQ_KNK_PRE[NC_CYL_NR]	4	0...3FH	0...63	1	[-]
LDPM_N_FIL_FRQ_KNK	4	0...FFH	0...8160	32	[rpm]
Knock Filter Frequency and Reference Filter Frequency Cyl.x previous					
ID_ITC_KNK_PRE [NC_CYL_NR]	16*4	0...1FH	0...31	1	[-]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
LDPM_MAF_1_9	4	0...FFH	0...1389	5.45E+00	[mg/stk]
Integration Time Constant Cyl.x previous					
IP_GAIN_KNK_PRE[NC_CYL_NR]	16*12	0...3FH	0...63	1	[-]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
LDPM_MAF_GAIN_KNK	12	0...FFH	0...1389	5.4470588	[mg/stk]
Amplification precontrol Cyl.x previous					
C_FIL_FRQ_KNK_AFL[NC_CYL_NR]	1	0...3FH	0...63	1	[-]
Knock Filter Frequency and Reference Filter Frequency Cyl.x in AFL or S					
C_ITC_KNK_AFL[NC_CYL_NR]	1	0...1FH	0...31	1	[-]
Integration Time Constant Cyl.x in AFL or S					
IP_GAIN_KNK_AFL[NC_CYL_NR]	16*4	0...3FH	0...63	1	[-]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
LDPM_MFF_1_9	4	0...FFFFH	0...1389	0.0211948	[mg/stk]
Amplification precontrol Cyl.x in AFL or S					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_KNKS_CONF	1	0...FFH	0...255	1	[-]
Configuration value of analogue channel of knock sensor for all cylinders (eg. 6 cylinder: 00101010b)					

45.4.2.4 Load and Speed Dynamics



45.4.2.4.1 Speed Dynamics Function

FUNCTION DESCRIPTION:

General information:

In order to keep track with the noise values for large engine speed gradients (LV_KNK_TRA_N = 1), the standard filtering constant is exchanged by a faster averaging constant C_NL_CRLC_TRA.
Care needs to be given to the absolute function evaluation, so that there is no difference in the treatment of positive and negative speed gradients.
Furthermore the values of Knock -Adaptation-circuits 1 + 2 are only read at active speed dynamics.

Application conditions:

Activation: at every engine state except engine stop

Deactivation: at engine stop

Initialization: LV_KNK_TRA_N = 0 at reset and engine running to engine stop

Update Rate: every segment

Formula section:

```
if    (|N_GRD| >= ID_N_GRD_MAX_KNK__N)

          LV_KNK_TRA_N = 1
else
          LV_KNK_TRA_N = 0
end if
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_N_GRD_MAX_KNK__N	8	0...7FH	0...4064	32	[rpm/s]
LDPM_N_IGA_AD	8	0...FFH	0...8160	32	[rpm]
Speed Gradient for dynamic identification					

45.4.2.4.2 Load Dynamics Function

FUNCTION DESCRIPTION:

General information:

In order to keep track with the noise values for large (positive) load gradients (LV_KNK_TRA_MAF = 1), the standard filtering constant is exchanged by a faster averaging constant C_NL_CRLC_TRA. After the triggering of the load dynamics the special filtering constant C_NL_CRLC_TRA is applied for the number of ID_CYCNR_FAC_TRA_KNK__N cycles. In addition, at active load dynamics function, the knock detection (KNK_THD[x]) is made less sensitive with the help of the factor C_FAC_TRA_KNK.
The values from the knock adaptation-circuits 1 + 2 are only read.

Application conditions:

Activation: at every engine state except engine stop

Deactivation: at engine stop

Initialization: LV_KNK_TRA_MAF = 0 at reset and engine running to engine stop

Update Rate: every segment

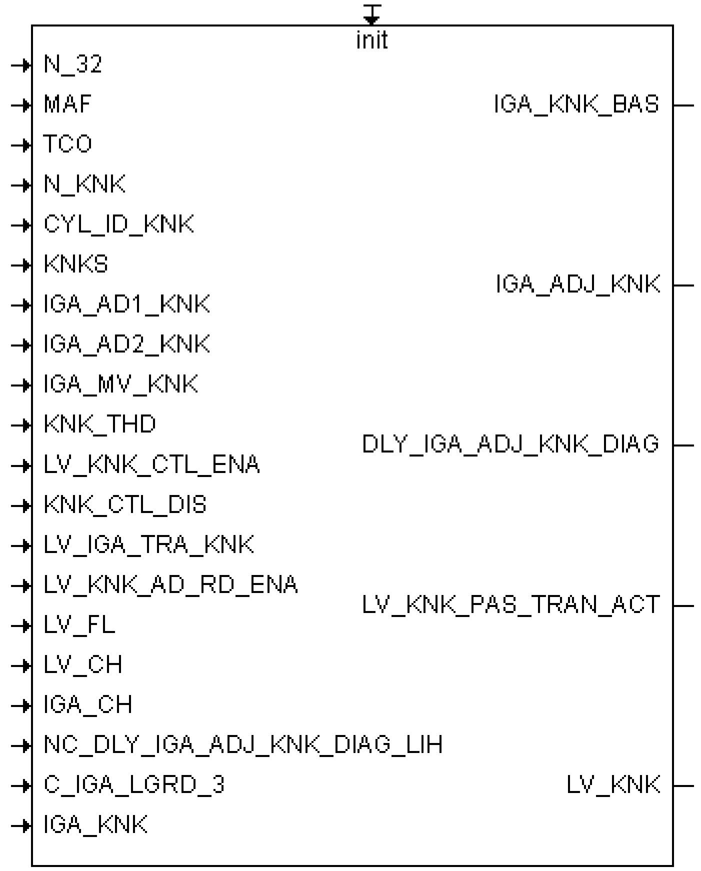
Formula section:

```
if      LV_IGA_TRA_KNK 0 -> 1

        LV_KNK_TRA_MAF = 1 for ID_CYCNR_FAC_TRA_KNK__N  cycles
else
            if      (ID_CYCNR_FAC_TRA_KNK__N has elapsed)
                    LV_KNK_TRA_MAF = 0
            end if
end if
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_CYCNR_FAC_TRA_KNK__N	4	0...FFH	0...255	1	[seg]
LDPM_N_6_4	4	0...FFH	0...8160	32	[rpm]
Cycle counter during which the Load Dynamic Function stays active after triggering					

45.4.3 Knock Control

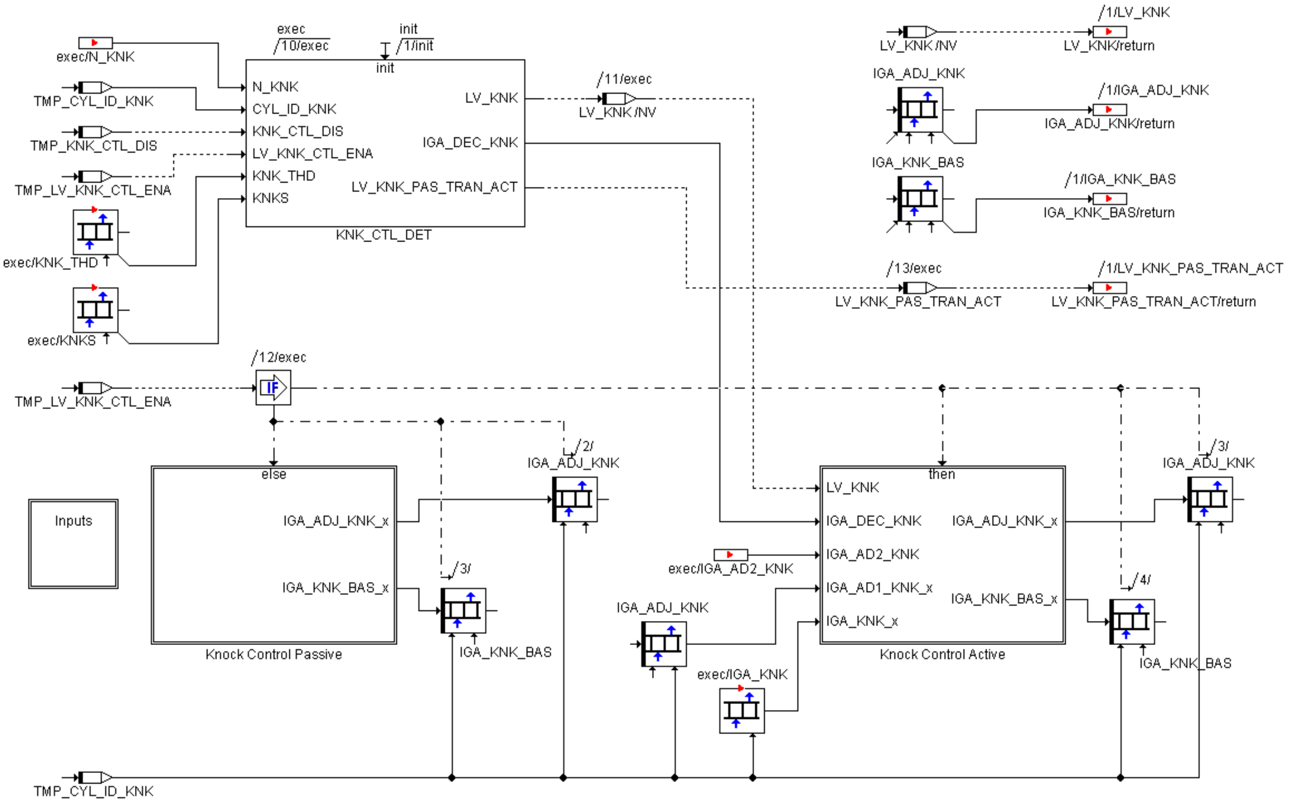


KNK_CTL_CTL

Overview:

```
→ Knock Control at change of operation mode
→ Knock Detection in separate window
→ Knock Detection
if    (LV_KNK_CTL_ENA = 1)
      → Knock Control active
            if    Combustion mode  = 'stratified'
                  → Calculation 'knock control in stratified mode'
            else
                  if
                  → Calculation of Maximum Retard Adjustment
                        if    (KNK_CTL_DIS = 1)
                              → Limp Home operation
                        else
                              if    (LV_KNK_CTL_PRE_ENA = 1)
                                    → Normal Operation Pre-Ignition
                              end if
                              → Normal operation
                              → Control excursion limit
                        end if
                  → Limitation of total ignition retard
                  else
```

```
                → Reset of Knock control
            end if
        endif
else
    → Knock control passive (Reset after abandoning the Knock-Operating-Limit)
end if
```



Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	at engine stop
Initialisation:	at reset and engine running to engine stop
	IGA_ADJ_KNK[NC_CYL_NR] = 0
	STATE_CMB_CTL_KNK = 0
Update Rate:	every segment

45.4.3.1 Knock Control at change of operation mode

Changing the operation mode causes some additional noise. To avoid faulty detection of knock, the knock control is cut off for a calibratable time. The noise level has to be shifted with a own correlation constant. This calculation has to be done before IGA_DEC_KNK is calculated.

Definition of temp variable for change between combustion states:

```
if          OPM_AV(n) != OPM_AV(n-1) // operation mode has changed
then
            if          OPM_AV = "AFS" to "AFL"
            then        STATE_CMB_CTL _TMP = "AFS_TO_AFL"       (1H)

            elseif      OPM_AV = "AFS" to "S"
            then        STATE_CMB_CTL _TMP = " HOM_TO_S "       (4H)

            elseif      OPM_AV = "S" to "AFL"
            then        STATE_CMB_CTL _TMP = "BACKHOM"          (8H)

            elseif      OPM_AV = "S" to "AFS"
            then        STATE_CMB_CTL _TMP = " S_TO_HOM "       (7H)

            elseif      OPM_AV = "AFL" to "AFS"
            then        STATE_CMB_CTL _TMP = "AFL_TO_AFS"       (3H)

            elseif      OPM_AV = "'AFL"  to "S"
            then        STATE_CMB_CTL _TMP = "BACKS"            (5H)
else
            if          OPM_AV = "AFS"
            then        STATE_CMB_CTL _TMP = "HOM_AFS"          (0H)

            elseif      OPM_AV = "AFL"
            then        STATE_CMB_CTL _TMP = "HOM_AFL"          (2H)

            elseif      OPM_AV = "S"
            then        STATE_CMB_CTL _TMP = "S"                           (6H)
endif


if          OPM_AV(n) != OPM_AV(n-1) // operation mode has changed
      then
            LV_CTR_KNK_CMB = 1
            CTR_KNK_CMB = 0
            STATE_CMB_CTL_KNK = STATE_CMB_CTL_TMP
endif

if    LV_CTR_KNK_CMB = 1 and CTR_KNK_CMB < ID_CTR_KNK_CMB
      then
            CTR_KNK_CMB ++
      else
            CTR_KNK_CMB = 0
            LV_CTR_KNK_CMB = 0
endif
```

ID_CTR_KNK_CMB is to be used in dividers of six (six segments equal one working cycle)

45.4.3.2 Knock Detection in separate window

FUNCTION DESCRIPTION:

General information:

Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	at engine stop
Initialisation:	at reset and engine running to engine stop LV_KNK_PRE=0 CYCNR_KNK_PRE_DET = 0 CTR_KNK_PRE_DET[0 to NC_CYL_NR] = 0 CYCNR_INH_IV_KNK_PRE_DET[0 to NC_CYL_NR] = 0 set all bits in INH_IV_KNK = 0
Update Rate:	every segment

Formula section:

```
if    (LV_KNK_CTL_PRE_ENA = 1)
      Knock Control Pre-Ignition Active:
      if     (KNK_CTL_DIS = 0)
             if     (NC_CYL_NR cycles has elapsed after Passive or Limp-Home)
                    LV_KNK_PAS_TRAN_ACT = 1
```

call action interface for acquisition of knks signal for pre-window

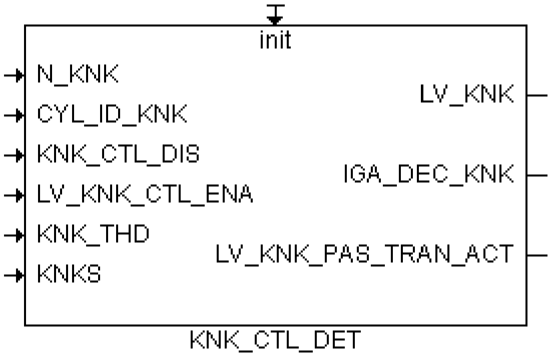
ACTION_INFR_*GetVKnks*(OUT <V_knks>, IN <Cyl>, IN <Win>)
Gets the formatted knock noise raw signal

```
                    if     LV_KNK_CTL_PRE_INH = 0 and
                           (KNKS_PRE[x] >= KNK_THD_PRE) and
                           LV_CTR_KNK_CMB=0 and
                           OPM_AV = "AFS" (no pre-window used in AFL or S)
                    then
                           LV_KNK_PRE = 1
                    else
                           LV_KNK_PRE = 0
                    end if
             else
                    LV_KNK_PAS_TRAN_ACT = 0
                    LV_KNK_PRE = 0
                    CYCNR_KNK_PRE_DET = 0 (reset of counter for observation period)
                    CTR_KNK_PRE_DET[x] = 0 (reset of detection counter)
                    CYCNR_INH_IV_KNK_PRE_DET[x] = 0 (reset of injection shut off counter)
                    set all bits in INH_IV_KNK = 0 (reset of pre-ignition inhibit bits)
             end if
      else
             LV_KNK_PAS_TRAN_ACT = 0
```

```
                LV_KNK_PRE = 0
                CYCNR_KNK_PRE_DET = 0 (reset of counter for observation period)
                CTR_KNK_PRE_DET[x] = 0 (reset of detection counter)
                CYCNR_INH_IV_KNK_PRE_DET[x] = 0 (reset of injection shut off counter)
                set all bits in INH_IV_KNK = 0 (reset of pre-ignition inhibit bits)
            end if
        else
            LV_KNK_PAS_TRAN_ACT = 0
            LV_KNK_PRE = 0
            CYCNR_KNK_PRE_DET = 0 (reset of counter for observation period)
            CTR_KNK_PRE_DET[x] = 0 (reset of detection counter)
            CYCNR_INH_IV_KNK_PRE_DET[x] = 0 (reset of injection shut off counter)
            set all bits in INH_IV_KNK = 0 (reset of pre-ignition inhibit bits)
        end if
```

N.B.: x represents current CYL_ID_KNK

45.4.3.3 Knock Detection



FUNCTION DESCRIPTION:

General information:

Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	at engine stop
Initialisation:	-
Update Rate:	every segment

Formula section:

```
        if    (LV_KNK_CTL_ENA = 1)
              Knock Control Active:
              if      (KNK_CTL_DIS = 0)
```

```
        if      (NC_CYL_NR cycles has elapsed after Passive or Limp-Home)
                LV_KNK_PAS_TRAN_ACT = 1
```

call action interface for acquisition of knks signal for normal window

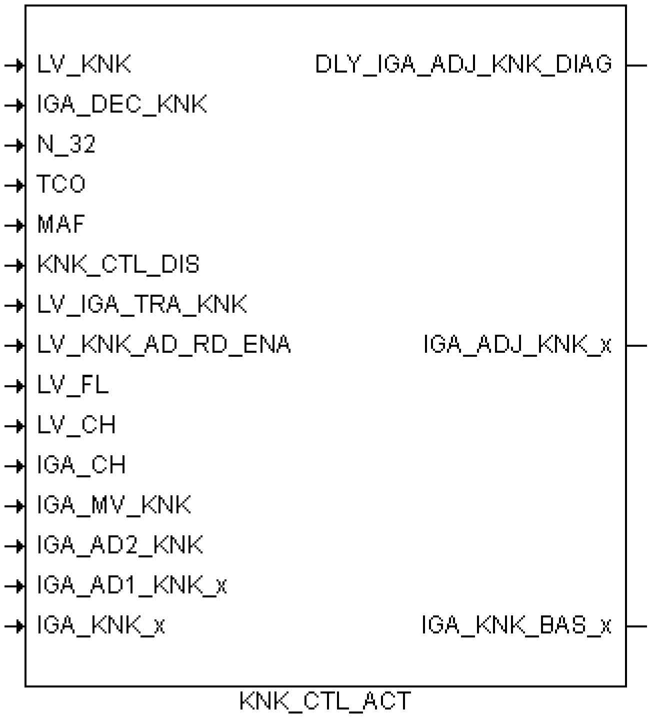
ACTION_INFR_*GetVKnks*(OUT <V_knks>, IN <Cyl>, IN <Win>)
Gets the formatted knock noise raw signal

```
                if      (KNKS[x] >= KNK_THD[x]) and LV_CTR_KNK_CMB = 0
                        LV_KNK = 1
                        if      OPM_AV = "AFS"
                                if      (KNKS[x] ≥ 2 * KNK_THD[x])
                                        Intensity 2:
                                        IGA_DEC_KNK = ID_IGA_DEC_KNK_2
                                else
                                        Intensity 1:
                                        IGA_DEC_KNK = ID_IGA_DEC_KNK_1
                                end if
                        else
                                if      (KNKS[x] ≥ 2 * KNK_THD[x])
                                        Intensity 2:
                                        IGA_DEC_KNK = ID_IGA_DEC_KNK_AFL_2
                                else
                                        Intensity 1:
                                        IGA_DEC_KNK = ID_IGA_DEC_KNK_AFL_1
                                end if
                        end if
                else
                        LV_KNK = 0
                end if
                if      (KNKS[x] >= KNK_THD_MAX) and LV_CTR_KNK_CMB = 0
                        LV_KNK_MAX = 1
                else
                        LV_KNK_MAX = 0
                end if
        else
                LV_KNK_PAS_TRAN_ACT = 0
                LV_KNK = 0
        end if
    else
        Limp-Home Intensity 1:
        IGA_DEC_KNK = ID_IGA_DEC_KNK_1
        LV_KNK_PAS_TRAN_ACT = 0
        LV_KNK = 0
    end if
else
    Knock Control Passive:
    LV_KNK_PAS_TRAN_ACT = 0
    LV_KNK = 0
end if
```

N.B.: x represents current CYL_ID_KNK

Different adjustment decrements are also produced due to the different knock intensities. Then it is possible to react to hard knock-strikes with a greater ignition - timing retardation.

45.4.3.4 Knock control active



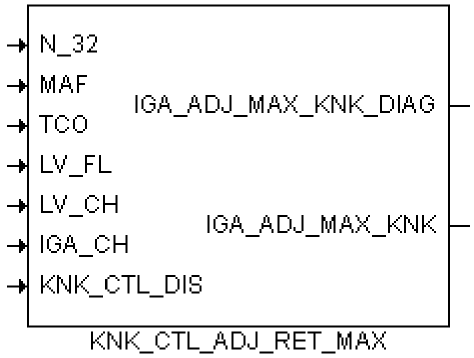
General information:

The spark retard is calculated cylinder-individual at every firing event. The spark retard due to the knock control system is determined by the spark retard due to knocking combustion (IGA_KNK[x]). The sum of these retard components is limited to a maximum spark retard.

Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	LV_ES = 1
Initialisation:	-
Update Rate:	every segment

45.4.3.4.1 Calculation of Maximum Retard Adjustment



FUNCTION DESCRIPTION:

General information:

This ignition angle is also executed if a knock control related malfunction is detected (KNK_CTL_DIS = 1).

Please note that this maximum spark retard is an absolute ignition angle referring to TDC.

Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	-
Initialisation:	-
Update Rate:	every segment

Formula section:

```
IGA_OFS_MAX_KNK      = IP_IGA_OFS_MAX_KNK(N_32, MAF)
FAC_IGA_MAX_KNK      = IP_FAC_IGA_MAX_KNK(TCO, TIA)
IF      LV_ERR_KNKS[NC_NR_SENS_KNK]  = 1   (or condition)
Then
        IGA_BAS_MAX_KNK      = IP_IGA_BAS_MAX_KNK_ERR(N_32, MAF)
Else
        IGA_BAS_MAX_KNK      = IP_IGA_BAS_MAX_KNK(N_32, MAF)
EndIf
IGA_ADJ_MAX_KNK = IGA_BAS_MAX_KNK + (IGA_OFS_MAX_KNK * FAC_IGA_MAX_KNK)
```

45.4.3.4.2 Normal Operation Pre-Ignition

FUNCTION DESCRIPTION:

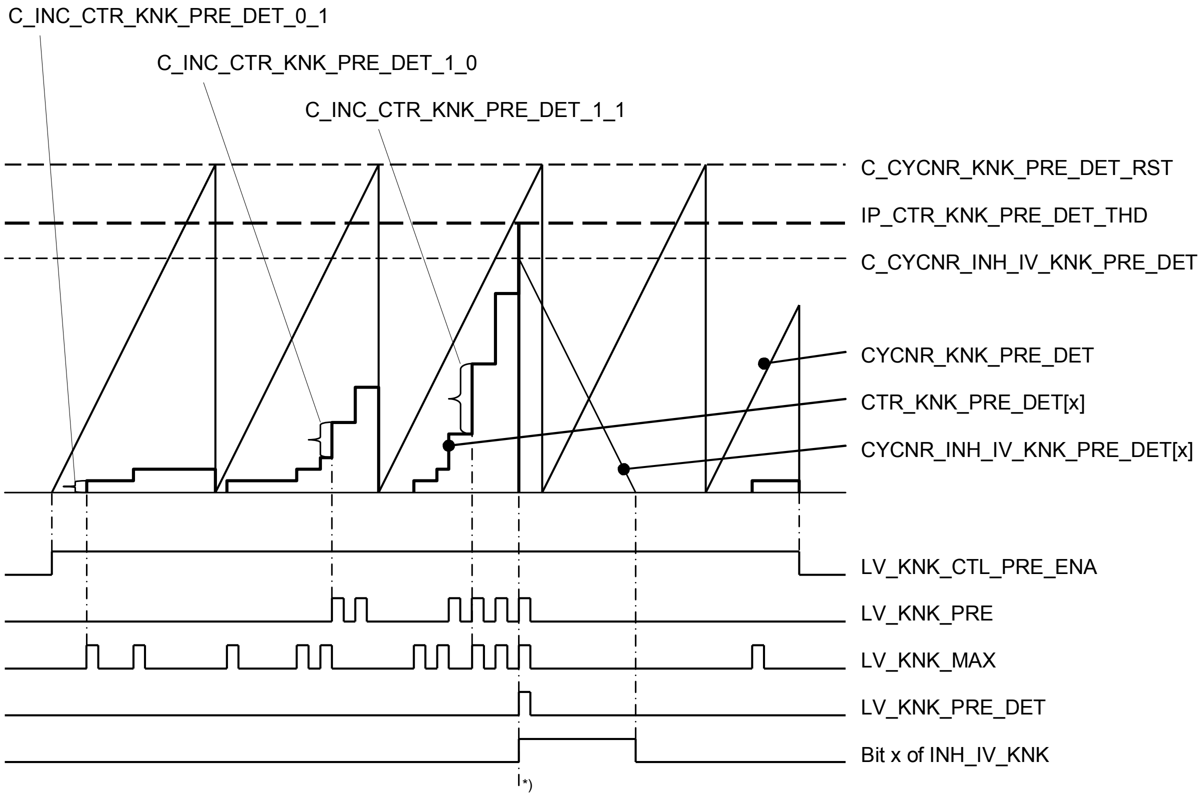
General information:

Surface ignition, before the regular ignition is triggered, is called pre-ignition. This kind of autoignition is caused by a hot spot, generated by prolonged, heavy spark knock. It is known as knock-induced pre-ignition. This is the most damaging side effect of spark knock.

Knock-induced pre-ignition is detected by following algorithm, that is observing normal knocking (normal knock window) and pre-ignition (pre-knock window). Both windows are evaluated to detect occuring pre-ignition events. If either many knock events are detected in the normal knock window or in the pre-knock window or in the normal knock window plus the pre-knock window during a certain number of firing events, autoignition is detected and the protection function of the engine is activated. Then the injection of the concerned cylinder is shut off for a calibratable number of firing events to prevent engine damage.

(No execution, if not (LV_KNK_CTL_PRE_ENA = 1))

Detection algorithm:



*) counter exceeds threshold within observation period --> cycle counter for injection inhibition starts

LV_KNK_PRE refers to ID_KNK_THD_MAX_PRE threshold,
LV_KNK_MAX refers to IP_KNK_THD_MAX threshold!

Formula section:

cycle counter for observation period
(update rate: every 720°)

```
CYCNR_KNK_PRE_DET = CYCNR_KNK_PRE_DET$_{n-1}$ + 1
```

cycle counter and detection counter reset when reaching cycle counter threshold
(update rate: every 720°)

```
If      CYCNR_KNK_PRE_DET > C_CYCNR_KNK_PRE_DET_RST
Then    CYCNR_KNK_PRE_DET = 0
        CTR_KNK_PRE_DET[0 to NC_CYL_NR] = 0
Endif
```

detection counter
(cylinder individually)

```
If      LV_KNK_PRE = 0 and LV_KNK_MAX = 1
Then    CTR_KNK_PRE_DET[x] = CTR_KNK_PRE_DET[x]$_{n-1}$ +
                                        C_INC_CTR_KNK_PRE_DET_0_1
Elseif  LV_KNK_PRE = 1 and LV_KNK_MAX = 0
Then    CTR_KNK_PRE_DET[x] = CTR_KNK_PRE_DET[x]$_{n-1}$ +
                                        C_INC_CTR_KNK_PRE_DET_1_0
Elseif  LV_KNK_PRE = 1 and LV_KNK_MAX = 1
Then    CTR_KNK_PRE_DET[x] = CTR_KNK_PRE_DET[x]$_{n-1}$ +
                                        C_INC_CTR_KNK_PRE_DET_1_1
Endif
```

knock pre-ignition detection and detection counter reset when reaching detection threshold
(cylinder individually)

```
If      CTR_KNK_PRE_DET[x] > IP_CTR_KNK_PRE_DET_THD
Then    LV_KNK_PRE_DET = 1
        CTR_KNK_PRE_DET[x] = 0
Else    LV_KNK_PRE_DET = 0
Endif
```

injection shut off for adjustable number of cycles
(cylinder individually)

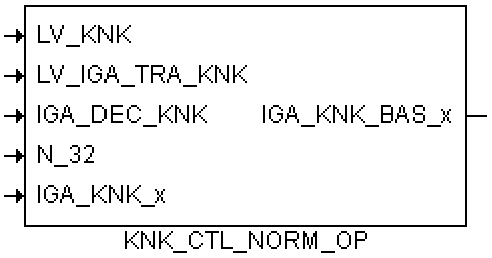
```
If      LV_KNK_PRE_DET = 1
Then    CYCNR_INH_IV_KNK_PRE_DET[x] = C_CYCNR_INH_IV_KNK_PRE_DET
Endif
```

```
If      CYCNR_INH_IV_KNK_PRE_DET[x] > 0
Then    CYCNR_INH_IV_KNK_PRE_DET[x] = CYCNR_INH_IV_KNK_PRE_DET[x]$_{n-1}$ - 1
        set bit x for corresponding cylinder in INH_IV_KNK = 1
Else    set bit x for corresponding cylinder in INH_IV_KNK = 0
Endif
```

this calculation has to be done before 'Cylinder Shut Off (Application Incidences)'

N.B.: x represents current CYL_ID_KNK

45.4.3.4.3 Normal Operation



FUNCTION DESCRIPTION:

General information:

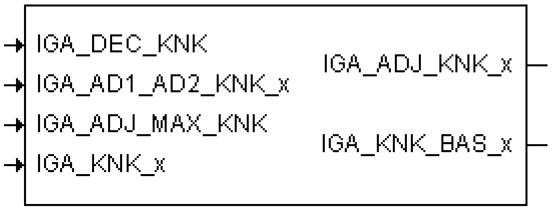
The spark retard is calculated cylinder-individual at every firing event. The spark retard due to the knock control system is determined by the spark retard due to knocking combustion (IGA_KNK[NC_CYL_NR]). The sum of these retard components is limited to a maximum spark retard.
In case of knock free operation (LV_KNK=0) and LV_KNK_CTL_ENA = 1 the Ignition retard due to knock control is re-set to zero (Controlled reset at Knock-free operation).
The controlled ignition-timing reset remains active until either the map ignition timing (IGA_KNK_BAS[x] = 0) is reached or until knocking occurs.

Formula section:

```
if        (LV_KNK = 1)
then      IGA_KNK_BAS[x] = IGA_KNK[x] - IGA_DEC_KNK
else      Controlled reset at Knock-free operation
    if    (LV_IGA_TRA_KNK = 0)
    then  IGA_KNK_BAS[x] = IGA_KNK[x] + IP_IGA_INC_KNK__N
    else  IGA_KNK_BAS[x] = IGA_KNK[x]
    end if
end if
```

N.B.: x represents current CYL_ID_KNK

45.4.3.4.4 Limp Home Operation



KNK_CTL_LIH

FUNCTION DESCRIPTION:

General information:

After an diagnosis error on the knock control system is detected (KNK_CTL_DIS = 1) the limp home spark retard level is achieved via spark retard Intensity 1 (IGA_DEC_KNK).

Formula section:

```
Check if limphome spark retard level is already reached
if    IGA_KNK[x] - IGA_DEC_KNK < IGA_ADJ_MAX_KNK - IGA_WOUT_KNK
limphome spark retard level reached: Limitation to max. absolute (refering to TDC) possible ignition retard
then
      IGA_KNK_BAS[x] = IGA_ADJ_MAX_KNK  – IGA_WOUT_KNK
Limphome spark retard level not yet reached
else
      IGA_KNK_BAS[x] = IGA_KNK[x] - IGA_DEC_KNK
end if
```

45.4.3.4.5 Control Exursion limit

FUNCTION DESCRIPTION:

General information:

The control excursion limit is used to limit the difference of the ignition adjustment angle through knock- control of the single cylinders to a certain value. All 720 °CRK the arithmetic mean value of the cylinder ignition adjustment angles IGA_KNK_BAS[NC_CYL_NR] is calculated:
All 720/Cyl. °CRK a test will be done, to see whether the ignition adjustment angle of the just calculated cylinder is more retarded or advanced than
$|$IGA_MV_KNK - C_IGA_ DIF_MIN_MAX_KNK$|$. In this case the knock control adjustment angle is limited to the control excursion limit.
In case that ignition angle retard reaches the maximum limit, LV_IGA_DIF_MAX_KNK is set in order to give this information to the 'Excursion Limit Analysis' functionality, where the values for concerning cylinder and actual operation point are stored.

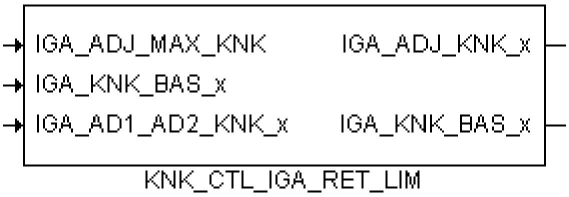
Formula section:

Case Differentiation:

```
Analysis trigger reset:
LV_IGA_DIF_MAX_KNK = 0
if          IGA_KNK_BAS[x]  >  IGA_MV_KNK + C_IGA_DIF_MIN_MAX_KNK
advance limitation:
then        IGA_KNK_BAS[x]  = IGA_MV_KNK + C_IGA_DIF_MIN_MAX_KNK
else
    if      IGA_KNK_BAS[x] < IGA_MV_KNK - C_IGA_DIF_MIN_MAX_KNK
    retard limitation:
    then    IGA_KNK_BAS[x] = IGA_MV_KNK - C_IGA_DIF_MIN_MAX_KNK
            Analysis trigger for cylinder individual counter, freeze of Cyl_ID_KNK and operation point:
            LV_IGA_DIF_MAX_KNK = 1
    endif
end if
```

N.B.: x represents current CYL_ID_KNK

45.4.3.4.6 Limitation of total ignition retard

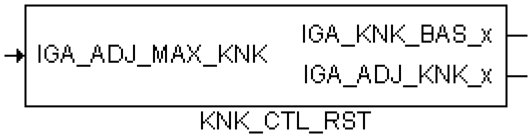


Formula section:

```
if    LV_HOM_ACT = 1
then
      check if limphome spark retard level is already reached
      if        IGA_KNK_BAS[x]  <  IGA_ADJ_MAX_KNK - IGA_WOUT_KNK
      limphome spark retard level reached: Limitation to max. absolute (refering to TDC) possible ignition retard
      then
                IGA_KNK_BAS[x] = IGA_ADJ_MAX_KNK – IGA_WOUT_KNK
      endif
endif
IGA_KNK[x] = IGA_KNK_BAS[x]
IGA_ADJ_KNK[x] = IGA_KNK_BAS[x]
```

N.B.: x represents current CYL_ID_KNK

45.4.3.4.7 Reset of Knock control

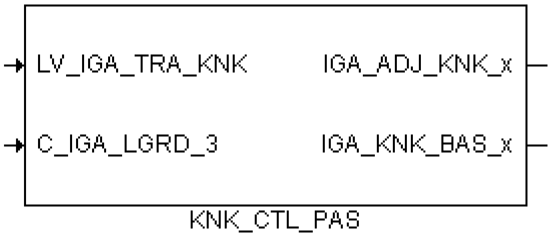


Formula section:

```
IGA_KNK_BAS[x]          = 0°CRK
IGA_ADJ_KNK[x]          = IGA_ADJ_MAX_KNK - IGA_WOUT_KNK
```

N.B.: x represents current CYL_ID_KNK

45.4.3.5 Knock control passiv (Reset after abandoning the Knock-Operating-Limit)



FUNCTION DESCRIPTION:

General information:

If the operating limit falls below (LV_KNK_CTL_ENA = 0), then the ignition adjustment angle IGA_KNK_BAS[x] is re-set to 0, with the change limitation C_IGA_LGRD_3 - however referring to 720° CRK -, in order to reach the map ignition timing.

Formula section:

```
if    (LV_IGA_TRA_KNK = 0)
then
      $IGA\_ADJ\_KNK[x] = IGA\_ADJ\_KNK[x]_{(N-1)} + C\_IGA\_LGRD\_3$
      $IGA\_KNK\_BAS[x] = 0$
else
      $IGA\_ADJ\_KNK[x] = IGA\_ADJ\_KNK[x]_{N-1)}$ (remains unchanged)
      $IGA\_KNK\_BAS[x] = 0$
end if
```

45.4.3.6 Knock Control in Stratified Mode

FUNCTION DESCRIPTION:

General information:

Knock detection and knock control is also active in stratified combustion mode. As ignition retard is not usable in stratified mode and may lead to problems at combustion mode changes, the spark retard variables IGA_ADJ_KNK and IGA_KNK_BAS are set to zero during stratifed. If stratified mode is left, the spark retard has to start at 0 °CRK when knock is recognized.

Formula section:

```
if    OPM_AV = 'S'        (stratified combustion, 1H)
then
```

call action interface for acquisition of knks signal for pre and normal window

ACTION_INFR_*GetVKnks*(OUT <V_knks>, IN <Cyl>, IN <Win>)
Gets the formatted knock noise raw signal

```
    IGA_ADJ_KNK[x]  = 0 °CRK
    IGA_KNK_BAS[x] = 0 °CRK
    if      (KNKSx >= KNK_THD[x])
    then    LV_KNK = 1
    else    LV_KNK = 0
    endif
    if      (KNKS_PRE[x] ≥ KNK_THD_PRE)
    then    LV_KNK_PRE = 1
    else    LV_KNK_PRE = 0
    end if
    CYCNR_KNK_PRE_DET = 0 (reset of counter for observation period)
    CTR_KNK_PRE_DET[x] = 0 (reset of detection counter)
    CYCNR_INH_IV_KNK_PRE_DET[x] = 0 (reset of injection shut off counter)
    set all bits in INH_IV_KNK = 0 (reset of pre-ignition inhibit bits)
else
    IGA_ADJ_KNK[x]  is calculated as described in the previous chapters
    IGA_KNK_BAS[x]  is calculated as described in the previous chapters
end if
```

Hints:

Due to former changes IGA_KNK[x] is now identical to IGA_KNK_BAS[x].

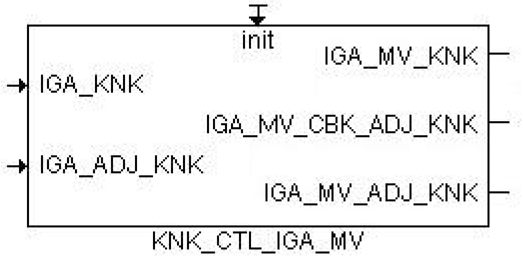
At transition from stratified combustion mode to normal operation mode of knock control, NC_CYL_NR cycles have to be elapsed for spark retard.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_IGA_DEC_KNK_1__N	3	1...7FH	0.375...47.625	0.375	[°CRK/cyc]
LDPM_N_4_9	3	0...FFH	0...8160	32	[rpm]
Spark retard at recognised knocking, Intensity 1					
ID_IGA_DEC_KNK_2__N	3	1...7FH	0.375...47.625	0.375	[°CRK/cyc]
LDPM_N_4_9	3	0...FFH	0...8160	32	[rpm]
Spark retard at recognised knocking, Intensity 2					
IP_IGA_BAS_MAX_KNK__N__MAF	8*8	0...FFH	-35.625...60	0.375	[°CRK]
LDPM_N_IGA_MAX_KNK	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_IGA_MAX_KNK	8	0...FFH	0...1389	5.4470588	[mg/stk]
Maximum value for spark retard					
IP_IGA_OFS_MAX_KNK__N__MAF	8*8	0...FFH	-48...47.625	0.375	[°CRK]
LDPM_N_IGA_MAX_KNK	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_IGA_MAX_KNK	8	0...FFH	0...1389	5.4470588	[mg/stk]
Maximum value for spark retard					
IP_FAC_IGA_MAX_KNK__TCO__TIA	4*4	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_TCO__FAC_IGA_TCO_KNK	4	0...FEH	-48...142.5	0.75	[°C]
LDP_TIA__FAC_IGA_TIA_KNK	4	0...FEH	-48...142.5	0.75	[°C]
Weighting factor for the max. spark retard against coolant					
IP_IGA_INC_KNK__N	4	1...7FFFH	0.00146... 47.99853	1.4648e-3	[°CRK/cyc]
LDPM_N_6_4	4	0...FFH	0...8160	32	[rpm]
Re-set Increment KNK					
C_IGA_DIF_MIN_MAX_KNK	1	0...80H	-48...0	0.375	[°CRK]
Control excursion limit valid for spark advance and spark retard					
ID_CTR_KNK_CMB	0	0...FFH	0...255	1	[-]
LDP_STATE_CMB_CTL_KNK	0	0H 1H 2H 3H 4H 5H 6H 7H 8H	HOM_AFS AFS_TO_AFL HOM_AFL AFL_TO_AFS HOM_TO_S BACKS S S_TO_HOM BACKHOM	1	[-]
Counter to disable KNK control after combustion mode switch					
IP_IGA_BAS_MAX_KNK_ERR__N__MAF	8*8	0...FFH	-35.625...60	0.375	[°CRK]
LDPM_N_IGA_MAX_KNK	8	0...FFH	0...8160	32	[rpm]
LDPM_MAF_IGA_MAX_KNK	8	0...FFH	0...1389	5.4470588	[mg/stk]
Maximum value for spark retard in case of KNK sensor error					
C_INC_CTR_KNK_PRE_DET_0_1	1	0...FFH	0...255	1	[-]
Increment for knock detection in normal knock window					
C_INC_CTR_KNK_PRE_DET_1_0	1	0...FFH	0...255	1	[-]
Increment for knock detection in knock pre-window					
C_INC_CTR_KNK_PRE_DET_1_1	1	0...FFH	0...255	1	[-]
Increment for knock detection in normal and knock pre-window					
C_CYCNR_KNK_PRE_DET_RST	1	0...FFFFH	0...65535	1	[-]
Reset threshold for observation period pre-ignition cycle counter					
IP_CTR_KNK_PRE_DET_THD	6	0...FFFFH	0...65535	1	[-]
LDP_N_CTR_KNK_PRE_DET	6	0...FFH	0...8160	32	[rpm]
Threshold for evaluated pre-ignition detection counter					
C_CYCNR_INH_IV_KNK_PRE_DET	1	0...FFFFH	0...65535	1	[-]
Cycle number of injection shut off duration by pre-ignition detection					
ID_IGA_DEC_KNK_AFL_1__N	3	1...7FH	0.375...47.625	0.375	[-]
LDPM_N_4_9	3	0...FFH	0...8160	32	[rpm]
Spark retard at recognised knocking, Intensity 1 in AFL or S					
ID_IGA_DEC_KNK_AFL_2__N	3	1...7FH	0.375...47.625	0.375	[-]
LDPM_N_4_9	3	0...FFH	0...8160	32	[rpm]
Spark retard at recognised knocking, Intensity 2 in AFL or S					

45.4.4 Post-processing

45.4.4.1 Meanvalues of knock-control ignition adjustment



Application conditions:

Activation: at every engine state except engine stop

Deactivation: -

Initialisation: at reset and engine running to engine stop

IGA_MV_KNK = IGA_MV_ADJ_KNK = 0
IGA_MV_CBK_ADJ_KNK[NC_CBK_EX_NR] = 0

Update Rate: every 720° CRK: IGA_MV_KNK, IGA_MV_ADJ_KNK

every segment: IGA_MV_CBK_ADJ_KNK[NC_CBK_EX_NR]

Formula section:

$$\mathbf{IGA\_MV\_KNK} = \frac{\sum_{i=1}^{NC\_CYL\_NR}(\mathbf{IGA\_KNK\_x})}{\mathbf{NC\_CYL\_NR}}$$

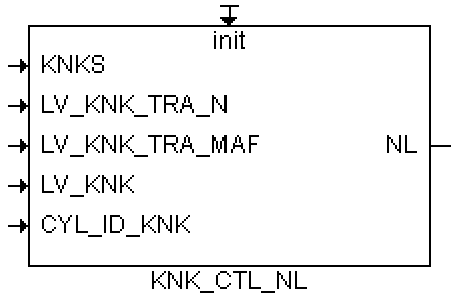
$$\mathbf{IGA\_MV\_ADJ\_KNK} = \frac{\sum_{i=1}^{NC\_CYL\_NR}(\mathbf{IGA\_ADJ\_KNK\_x})}{\mathbf{NC\_CYL\_NR}}$$

Calculation of IGA_MV_CBK_ADJ_KNK[NC_CBK_EX_NR]:

```
IF      NC_CBK_EX_NR > 1                    (please note physical meaning)
THEN
        FOR i = 1 TO NC_CBK_EX_NR           (please note physical meaning)
                IGA_MV_CBK_ADJ_KNK[i] = 0
        ENDFOR
        FOR x = 0 TO (NC_CYL_NR – 1)
```

```
                IF (NC_LAMB_REF AND $2^x$) ≠ 0          (bitwise)
                THEN
                        Bank 2:
                        IGA_MV_CBK_ADJ_KNK[2] += IGA_ADJ_KNK[x]
                ELSE
                        Bank 1:
                        IGA_MV_CBK_ADJ_KNK[1] += IGA_ADJ_KNK[x]
                ENDIF
        ENDFOR
        FOR i = 1 TO NC_CBK_EX_NR
                IGA_MV_CBK_ADJ_KNK[i] = IGA_MV_CBK_ADJ_KNK[i] /
                        (NC_CYL_NR / NC_CBK_EX_NR)
        ENDFOR
ELSE
        IGA_MV_CBK_ADJ_KNK[1] = 0
        FOR x = 0 TO (NC_CYL_NR – 1)
                IGA_MV_CBK_ADJ_KNK[1] += IGA_ADJ_KNK[x]
        ENDFOR
        IGA_MV_CBK_ADJ_KNK[1] = IGA_MV_CBK_ADJ_KNK[1] / NC_CYL_NR
ENDIF
```

45.4.4.2 Noise Value



FUNCTION DESCRIPTION:

The noise value is cylinder-individually calculated as sliding average from the knock values. The averaging constant is either taken from ID_NL_CRLC, if no knock control transient function is active (LV_KNK_TRA_N = 0 and LV_KNK_TRA_MAF = 0) or C_NL_CRLC_TRA, when a knock control transient function is active (LV_KNK_TRA_N = 1 or LV_KNK_TRA_MAF = 1).

Application conditions:

Activation: at every engine state except engine stop

Deactivation: at engine stop

Initialization: NL[0 to NC_CYL_NR] = C_KNK_THD_MAX

at reset or engine running to engine stop

Update Rate: every segment

Formula section:

```
if        (LV_KNK_TRA_N or LV_KNK_TRA_MAF)
 then     if      OPM_AV = "AFS"
          then    NL[x] =  NL[x] + (KNKS[x] – NL[x]) * C_NL_CRLC_TRA
          else    NL[x] = NL[x] + (KNKS[x] – NL[x]) * C_NL_CRLC_TRA_AFL
          end if

 elseif   LV_CTR_KNK_CMB = 1
 then     NL[x] = NL[x] + (KNKS[x] – NL[x]) * C_NL_CRLC_CMB

else      NL[x] =  NL[x] + (KNKS[x] – NL[x]) * ID_NL_CRLC
end if
```

NL[NC_CYL_NR] shows always the measured noise value from the ATIC62 device. The noise value is calculated always (independant from detection of knocking combustion).

In transient condition, if knocking combustions are detected (LV_KNK = 1) after knock-free operation, the first noise value is used weighted ((KNKS[x] / IP_FAC_THD_KNK) * C_FAC_KNK_TRA instead of simply use of KNKS[x]) to calculate the noise level, the second and the following are taken into consideration for normal transient calculation of noise level again.
Else, if KNKS[x] ≥ IP_FAC_NL_UPD_DEAC * KNK_THD[x] (signal near threshold) AND LV_KNK_PAS_TRAN_ACT = 1 (delay cycles elapsed after passive or limp-home) AND LV_CTR_KNK_CMB = 0 (delay counter elapsed after combustion mode switch), the first and the second noise value isn`t used to calculate the noise level (NL[NC_CYL_NR] remains unchanged), the third and the following are taken into consideration for calculation of noise level. NL[NC_CYL_NR] remains unchanged for this time.

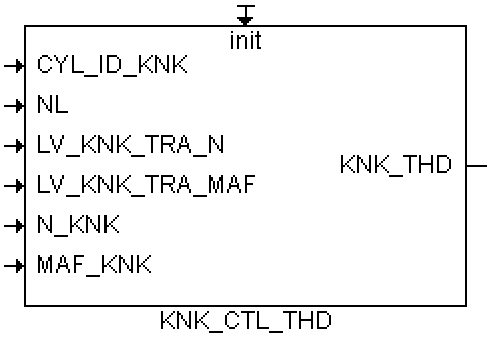
N.B.: x represents current CYL_ID_KNK

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_NL_CRLC	4	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_N_KNK_NL_CRLC	4	0...FFH	0...8160	32	[rpm]
Averaging Constant for Noise Value Calculation					
C_FAC_KNK_TRA	1	0...FFH	0...1.99218	0.0078125	[-]
Factor for weighted KNKS consideration at knocking in transient condition					
C_NL_CRLC_TRA	1	0...FFH	0...0.99609	3.9063e-3	[-]
Averaging Constant for Noise Value Calculation for Transient Case					
C_NL_CRLC_CMB	1	0...FFH	0...0.99609	3.9063e-3	[-]

Averaging Constant for Noise Value Calculation at HOM_AFS combustion					
IP_FAC_NL_UPD_DEAC	16	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
Factor on KNK_THD for comparison with KNKS to deactivate an update of NL values					
C_NL_CRLC_TRA_AFL	1	0...FFH	0...0.99609	3.9063e-3	[-]
Averaging Constant for Noise Value Calculation for Transient Case at AFL or S combustion					

45.4.4.3 Relative and Absolute Knock Threshold



FUNCTION DESCRIPTION:

General information:

The knock threshold (KNK_THD[NC_CYL_NR]) is the limit for the knock values (KNKSx), which are used for knock decision. If the knock values are greater or equal to the knock threshold, then knocking is recognised (LV_KNK = 1) and an ignition timing retardation follows. If the knock values are smaller, knocking is not detected (LV_KNK = 0) and the relevant KNK-ignition-angle is reset (→ Knock-free Operation after Knocking).

The knock threshold is calculated for each cylinder separately and depends on the cylinder-individual noise value. Calculation of the knock threshold occurs with help from the cylinder-individual knock factors IP_FAC_THD_KNK[NC_CYL_NR] and knock summand IP_ADD_KNK__N.

If the noise value is above the minimum threshold C_NL_MIN, the knock threshold is calculated on the basis of the cylinder-individually noise value NL[NC_CYL_NR]. If the noise value is below C_NL_MIN this value is used as noise value for calculation of the knock threshold. If the knock threshold is above ID_THD_MAX_KNK this value is used as knock threshold (Limitation). C_THD_MAX_KNK stays preserved as initialisation value for NL and NL_0_5 (in multiplexer specification).

N.B.: The actual knock value KNKS[x]$_n$ is not used in the calculation of the actual knock threshold KNK_THD[x]$_n$ but in the calculation of the next threshold KNK_THD[x]$_{n+1}$. So the decision knocking combustion yes/no is done with using the knock threshold KNKS[x]$_{n-1}$.

Application conditions:

Activation:	at every engine state except engine stop
Deactivation:	at engine stop
Initialization:	KNK_THD[0 to NC_CYL_NR] = KNK_THD_PRE = 5V at reset and engine running to engine stop
Update Rate:	every segment

Formula section:

The lower absolute threshold is one value for all cylinders and is represented by the constant C_NL_MIN.

```
If   NL[x] < C_NL_MIN
     NL_LIM = C_NL_MIN
else
     NL_LIM = NL[x]
end if

if   OPM_AV = "AFS"
     if      (LV_KNK_TRA_N = 0 and LV_KNK_TRA_MAF = 0)
             KNK_THD[x] = ( NL_LIM + IP_KNK_THD_ADD_N ) * IP_FAC_THD_KNK[x]
     else
             KNK_THD[x] = ( NL_LIM + IP_KNK_THD_ADD_N ) * IP_FAC_THD_KNK[x] *
                                  C_FAC_TRA_KNK
     end if
else
     if      (LV_KNK_TRA_N = 0 and LV_KNK_TRA_MAF = 0)
             KNK_THD)[x] = ( NL_LIM + IP_KNK_THD_ADD_N_AFL ) *
                                  IP_FAC_THD_KNK_AFL[x]
     else
             KNK_THD)[x] = ( NL_LIM + IP_KNK_THD_ADD_N_AFL ) *
                                  IP_FAC_THD_KNK_AFL[x] * C_FAC_TRA_KNK
     end if
end if
```

The knock threshold is generally limited to KNK_THD_MAX

```
KNK_THD_MAX = IP_KNK_THD_MAX
If  (KNK_THD[x] > KNK_THD_MAX)
    KNK_THD[x] = KNK_THD_MAX
end if

If  LV_KNK_CTL_PRE_INH = 0
    KNK_THD_PRE = ID_KNK_THD_MAX_PRE
end if
```

N.B.: x represents current CYL_ID_KNK

(*) The correction factor **C_FAC_TRA_KNK** is part of the KNK-load dynamics function and is effective after recognised load dynamics **ID_CYCNR_FAC_TRA_KNK__N** cycles.

The knock factor is the multiplicative part used for calculation of the knock threshold from the noise value.
The knock summand is an additive correction of the knock noise value. Thereby the knock threshold can be moved, dependent on the speed, in a positive or negative direction.

KNK_THD_MAX with respect to **IP_KNK_THD_MAX** is used as absolute threshold for detection algorithm in main window.
KNK_THD_PRE with respect to **ID_KNK_THD_MAX_PRE** is used as absolute threshold for detection algorithm in pre-window.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_THD_KNK[NC_CYL_NR]	16*4	0...FFH	0...7.96875	0.03125	[-]
LDPM_N_2_9c	16	0...FFH	0...8160	32	[rpm]
LDPM_MAF_1_9	4	0...FFH	0...1389	5.45E+00	[mg/stk]
Knock Factor Table Cyl. X					
C_FAC_TRA_KNK	1	0...FFH	0...7.96875	0.03125	[-]
correction factor for load dynamic active					
C_NL_MIN	1	0...FFH	0...4.998	1.95E-02	[V]
Minimum noise value					
IP_KNK_THD_ADD_N	16	0...FFH	-2.56...2.54	0.02	[V]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
Knock summand					
C_KNK_THD_MAX	1	0...FFH	0...4.998	0.0195313	[V]
Maximum knock threshold					
IP_KNK_THD_MAX	6*4	0...FFH	0...4.998	0.0195313	[V]
LDPM_N_KNK_THD_MAX	6	0...FFH	0...8160	32	[rpm]
LDPM_MAF_1_9	4	0...FFH	0...1389	5.4470588	[mg/stk]
Maximum knock threshold					
ID_KNK_THD_MAX_PRE	6	0...FFH	0...4.998	0.0195313	[V]
LDPM_N_KNK_THD_MAX	6	0...FFH	0...8160	32	[rpm]
Maximum knock threshold pre-window					
IP_FAC_THD_KNK_AFL[NC_CYL_NR]	16*4	0...FFH	0...7.96875	0.03125	[-]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
LDPM_MFF_1_9	4	0...FFFFH	0...1389	0.0211948	[mg/stk]
Knock Factor Table Cyl. X in AFL or S					
IP_KNK_THD_ADD_N_AFL	16	0...FFH	-2.56...2.54	0.02	[V]
LDPM_N_2_9	16	0...FFH	0...8160	32	[rpm]
Knock summand in AFL or S					

45.5 Knock sensor- diagnosis (KNKS_1/2)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CDN_DIAG_KNKS[NC_NR_SENS_KNK]	V/O	0...1H	0...1	1	[-]
Boolean for knock sensor 2 error currently active					
DYW_KNKS_DIAG[NC_NR_SENS_KNK]	V	0...3FFH	0...4.99511	4.8828e-3	[V]
Bandwidth knock signal to noise level					
CYCNR_DYW_KNKS_DIAG[NC_NR_SENS_KNK]	V	0...FFH	0...255	1	[seg]
Cycle counter for ´Master´-algorithm (knock sensor diagnosis)					
V_INT_KNKS_DIAG[NC_NR_SENS_KNK]	V	0...FFFFH	0...319.99511	4.8828e-3	[V]
Integrator value of bandwidth output voltage of the knock sensor					
CYCNR_INT_KNKS_DIAG[NC_NR_SENS_KNK]	V	0...FFH	0...255	1	[seg]
Cycle counter ´Slave´- algorithm (knock sensor diagnosis)					
LV_ERR_KNKS[NC_NR_SENS_KNK]	V/O	0...1H	0...1	1	[-]
Boolean for knock sensor 1 error currently active					
ERR_SYM_KNKS[NC_NR_SENS_KNK]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom KNKS					
LV_END_DIAG_KNKS[NC_NR_SENS_KNK]	V/O	0...1H	0...1	1	[-]
End of diagnosis KNKS					

Input data:

KNKS[NC_CYL_NR]	NL[NC_CYL_NR]	LV_PU	LV_PUC
MAF	N_32	LV_INH_DIAG_KNKS	LV_KNK_CTL_ENA
KNKS_1_DIAG	KNKS_2_DIAG	NC_KNKS_CONF	NC_NR_SENS_KNK
LV_DC	USE_SW_VER	MFF_SP_MV	

FUNCTION DESCRIPTION:

The array NC_CBK_EX_NR refers to sensor number (0 is KNKS_1; 1 is KNKS_2).

General information:

The task of the function "knock sensor diagnosis" lies in the input signal monitoring of the respective knock sensor.

The sensor signal is checked continuously regarding plausibility. This is done by observation of the signal noise value NL[NC_CYL_NR] ('Range Check') and with two algorithms (‘Master’- and ‘Slave’-Algorithm). The bandwidth (DYW_KNKS_DIAG) of the signal will be observed with these two algorithms. Both algorithms have to indicate the same system condition before the debounce counter is started.

The debounce counter will be called up every 720 °KW. The three test algorithms will be called up segment synchronously which will be done under consideration of the configuration values NC_KNKS_CONF (which knock sensor "hears" which cylinder), KNKS_1_DIAG and KNKS_2_DIAG (which cylinder will be taken for which knock sensor regarding diagnosis).

Range Check

In the range check a test will be done to find out if the output signal of the knock-IC is in the allowed working range between the calibratable minimum fault detection threshold and maximum fault detection threshold. A fault case is given if the minimum threshold is fallen-below or the maximum limit is exceeded. The ERR_SYM_KNKS is set when one (or more) of the monitored signals infringe a fault threshold.

‘Master’-Algorithm

In the master-algorithm the bandwidth of the signal (DYW_KNKS_DIAG = | NL[NC_CYL_NR] – KNKS[NC_CYL_NR] will be calculated for the concerned cylinder which is allocated to the concerned knock sensor regarding diagnosis purposes. The calculated bandwidth will be compared to the calibrated threshold value C_DYW_MIN_KNKS_DIAG.

At short circuit against ground or VB as well as a broken signal line the bandwidth of the signal does not exceed the threshold value. In this case the counter CYCNR_DYW_KNKS_DIAG will be incremented in each monitored segment.

As soon as the bandwidth of the sensor signal exceeds the value C_DYW_MIN_KNKS_DIAG again and LV_ERR_KNKS is not set the value will be reset again. In the case of LV_ERR_KNKS already set the value CYCNR_DYW_KNKS_DIAG will be decremented by 5 units.

If the counter reaches the value C_CYCNR_KNKS_DIAG a failure of the knock sensor will be set if the 'slave'-algorithm confirms this failure. When a failure has already debounced the SYM_3 is set as long as the counter of the master-algorithm is above 5. This is to delay the failure recovery.

The counter will be reset at transition LV_CDN_DIAG_KNKS = 0=>1.

‘Slave’-Algorithm

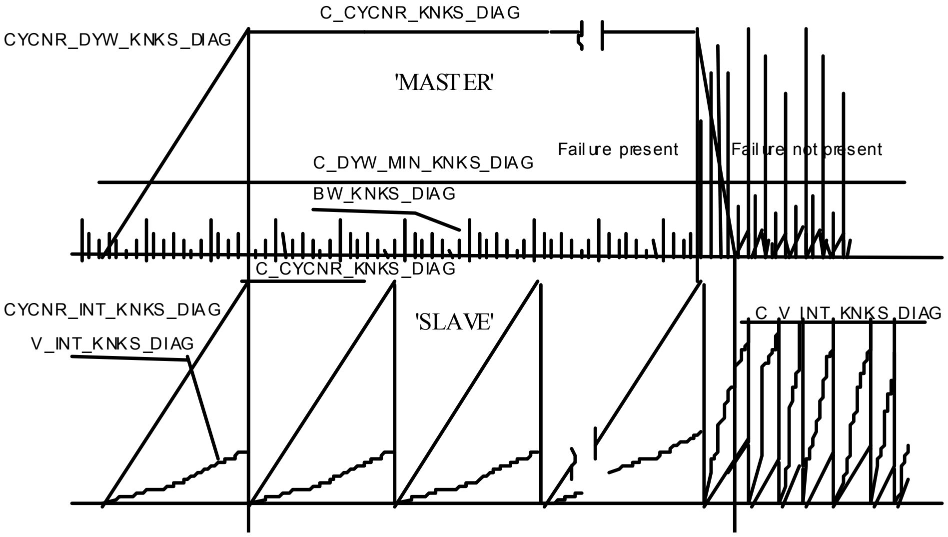
In the slave-algorithm the bandwidth of the sensor signal will also be used as an input signal. It will be recorded in its chronological process and will be integrated by means of an appropriate integration method. The calculated integral value V_INT_KNKS_DIAG will be compared to the threshold value C_V_INT_KNKS _DIAG.

As soon as the threshold value is reached or exceeded the counter CYCNR_INT_KNKS_DIAG will be reset. Otherwise it will be incremented by '1' in each monitored segment. When the counter reaches the value C_CYCNR_KNKS_DIAG the sensor failure will be confirmed through the slave-algorithm and value CYCNR_INT_KNKS_DIAG and value V_INT_KNKS_DIAG reset to 0.

The counter will be reset at transition LV_CDN_DIAG_KNKS = 0=>1.

If a sensor failure is recognized by both algorithms the debounce will get started.

Description:



Application conditions:

Initialisation: at reset **or** LV_IGK = 0->1 **or** clearing FMY:

all = 0, except:

LV_ERR_KNKS[NC_NR_SENS_KNK] **and**

CTR_ABC_KNKS[NC_NR_SENS_KNK] are initialised from FMY

(according STD filter type)

Recurrence: Signal range check and Master- Slave- Algorithm: segment

Error management: each 720 °KW

Activation:

If LV_KNK_CTL_ENA = 1 **and**
LV_PU = 0 **and**
LV_PUC = 0 **and**
[(USE_SW_VER = NC_MSV70 **and** MAF > C_MAF_MIN_KNKS_DIAG)
or
(USE_SW_VER = NC_MSD70 **and** MFF_SP_MV >
C_MFF_MIN_KNKS_DIAG)]
and
N_32> C_N_MIN_KNKS_DIAG **and**
LV_INH_DIAG_KNKS = 0
Then LV_CDN_DIAG_KNKS = 1
Transition LV_CDN_DIAG_KNKS =0-> 1
CYCNR_INT_KNKS_DIAG[NC_CBK_EX_NR] = 0;
CYCNR_DYW_KNKS_DIAG[NC_CBK_EX_NR] = 0;
V_INT_KNKS_DIAG[NC_CBK_EX_NR] = 0;
Else LV_CDN_DIAG_KNKS = 0

Formula section:

Master-Algorithm

$DYW\_KNKS\_DIAG[i] = |NL[x] - KNKS[x]|$

```
If      $DYW\_KNKS\_DIAG[i] > C\_DYW\_MIN\_KNKS\_DIAG$                    or
        $DYW\_KNKS\_DIAG[i]_{n-1} > C\_DYW\_MIN\_KNKS\_DIAG$
Then    If      $LV\_ERR\_KNKS[i] = 1$                   and
                $CYCNR\_DYW\_KNKS\_DIAG[i] > 5$
        Then    $CYCNR\_DYW\_KNKS\_DIAG[i]_n = CYCNR\_DYW\_KNKS\_DIAG[i]_{n-1} - 5$
        Else    $CYCNR\_DYW\_KNKS\_DIAG[i] = 0$
        Endif
Else    If      $CYCNR\_DYW\_KNKS\_DIAG[i] < C\_CYCNR\_KNKS\_DIAG$
        Then    $CYCNR\_DYW\_KNKS\_DIAG[i]_n = CYCNR\_DYW\_KNKS\_DIAG[i]_{n-1} + 1$
        Endif
Endif
```

Slave-Algorithm

$V\_INT\_KNKS\_DIAG[i] = V\_INT\_KNKS\_DIAG[i]_{n-1} + DYW\_KNKS\_DIAG[i]$

$CYCNR\_INT\_KNKS\_DIAG[i]_n = CYCNR\_INT\_KNKS\_DIAG[i]_{n-1} + 1$

```
If      $CYCNR\_INT\_KNKS\_DIAG[i]_n \geq C\_CYCNR\_KNKS\_DIAG$
Then    $CYCNR\_INT\_KNKS\_DIAG[i]_n = 0$
        $V\_INT\_KNKS\_DIAG[i]_n = 0$
Else    If      $V\_INT\_KNKS\_DIAG[i]_{n-1} \geq C\_V\_INT\_KNKS\_DIAG$
        Then    $CYCNR\_INT\_KNKS\_DIAG[i]_n = 0$
                $V\_INT\_KNKS\_DIAG[i]_n = 0$
        Endif
Endif
```

Signal range check and anti bounce

```
If      NL[x] > C_NL_MAX_DIAG                          /*at least one signal is above threshold*/
Then    ERR_SYM_KNKS[i] = SYM_0                        /*signal too high*/
Else    If      NL[x] < C_NL_MIN_DIAG                  /*at least one signal is below  threshold*/
        Then    ERR_SYM_KNKS[i] = SYM_1                /*signal too low*/
        Else    If      ( CYCNR_DYW_KNKS_DIAG[i] ≥ C_CYCNR_KNKS_DIAG and
                        CYCNR_INT_KNKS_DIAG[i] ≥ C_CYCNR_KNKS_DIAG ) or

                        ( CYCNR_DYW_KNKS_DIAG[i] ≥ 5 and
                        LV_ERR_KNKS[i] = 1)
                Then    ERR_SYM_KNKS[i] = SYM_3        /*signal implausible*/
                Else    ERR_SYM_KNKS[i] = NO_SYM       /*signal ok*/
                Endif
        Endif
Endif
```

The bit **LV_ERR_KNKS** will be set after debounce through the error management.
The bit **LV_END_DIAG_KNKS** will be set through the error management when after the last start of the diagnosis (normally it will be done only at the beginning of a Driving Cycle) the error debounce or the failure recovery would have been taken place.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> Noise level above valid range          (= SYM_0)
 │  │  └────> Noise level below valid range          (= SYM_1)
 │  └───────> -                                      (= SYM_2)
 └──────────> Knock sensor signal not plausible      (= SYM_3)
                  (Master-Slave)
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_MAF_MIN_KNKS_DIAG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower engine load treshold for knock sensor diagnosis					
C_N_MIN_KNKS_DIAG	1	0...FFH	0...8160	32	[rpm]
Lower engine speed treshold for knock sensor diagnosis					
C_NL_MIN_DIAG	1	0...FFH	0...4.998	0.0195313	[V]
Minimum noise level threshold					
C_NL_MAX_DIAG	1	0...FFH	0...4.998	0.0195313	[V]
Maximum noise level threshold					
C_DYW_MIN_KNKS_DIAG	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Bandwidth threshold to detect the error via the `Master Algorithm¿					
C_V_INT_KNKS_DIAG	1	0...FFFFH	0...319.99511	4.8828e-3	[V]
Integrator voltage threshold to detect the error via the `Slave Algorithm¿					
C_CYCNR_KNKS_DIAG	1	0...FFH	0...255	1	[seg]
Number of cycles (f(KNKS_DIAG[i])) for the Master-Slave knock sensor diagnosis					
C_ABC_INC_KNKS	1	0...FFH	0...255	1	[-]
Anti-bounce increment value					
C_ABC_MAX_KNKS	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value					
C_MFF_MIN_KNKS_DIAG	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Lower engine load treshold for knock sensor diagnosis					

45.6 Knock sensor diagnosis (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_KNKS	V/O	0...1H	0...1	1	[-]
Flag to inhibit the knock sensor diagnosis					

Input data:

LV_ERR_SPI_KNK			

FUNCTION DESCRIPTION:

General information:

This application incidences give the possibility to define customer specific conditions for inhibiting the diagnosis of knock sensors.

Application conditions:

Initialisation: LV_INH_DIAG_KNKS = 1 at Reset

Recurrence: like diagnosis

Activation: LV_IGK = 1

Formula section:

```
If      LV_ERR_SPI_KNK = 1
Then    LV_INH_DIAG_KNKS = 1
Else    LV_INH_DIAG_KNKS = 0
Endif
```

45.7 Pre-ignition diagnosis via knock control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_KNK_PRE[NC_CYL_NR]	O/V	0...1H	0...1	1	[-]
Error caused by knock control pre-ignition cylinder shut off					
LV_END_DIAG_KNK_PRE[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Diagnosis knock control pre-ignition cylinder shut off done at least one time					
LV_CDN_DIAG_KNK_PRE[NC_CYL_NR]	V	0...1H	0...1	1	[-]
Diagnosis condition knock control pre-ignition cylinder shut off					
ERR_SYM_KNK_PRE[NC_CYL_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom knock control pre-ignition cylinder shut off					

Input data:

SEG_NR	INH_IV_KNK	LV_KNK_CTL_PRE_ENA	

FUNCTION DESCRIPTION:

General information:

Workshops must be able to get information about cylinder shut off caused by pre-ignition detection e.g. in case of engine damages.

Description:

For error management failures on single cylinders should be set in case of cylinder shut off by pre-ignition detection algorithm within knock control.
An ABC counter starts incrementing segment synchronous. In case if a calibrataible number of counted segments is reached, a fault is detected by the error management.

Application conditions:

Initialisation: according filter type 'STD_INI'

Recurrence: segment synchronous (analogous knock control)
every 720° CRK for the individual cylinder – update for cylinder x
only in the corresponding segement, where injection shut off bit could change

Activation: **If** LV_KNK_CTL_PRE_ENA = 1
Then LV_CDN_DIAG_KNK_PRE[x] = 1

Deactivation: **If** LV_KNK_CTL_PRE_ENA = 0
Then LV_CDN_DIAG_KNK_PRE[x] = 0

Formula section:

```
If      bit x for corresponding cylinder in INH_IV_KNK = 1
Then    ERR_SYM_KNK_PRE[x]  = SYM_3
Else    ERR_SYM_KNK_PRE[x]  = NO_SYM
Endif
```

The bits LV_ERR_xx and LV_END_DIAG_xx are set by the error management after debounce.

N.B.: x represents the SEG_NR, where the corresponding cylinder can have a change in position x of INH_IV_KNK

Failure reaction:

handled by INH_IV_KNK

Configuration for diagnostic symptoms:

For each cylinder a separate fault location is used

symptoms for cylinder[NC_CYL_NR]

Diagnostic KNK_PRE	Symptom description	Symptom	Filter type
knock control pre-ignition cylinder shut off *cylinder[NC_CYL_NR]*		-	STD_INI
		-	
		-	
	knock control pre-ignition cylinder shut off	*SYM_3*	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_KNK_PRE	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_KNK_PRE	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					

45.8 Customer adaptation module: AGGR KNOCK

45.8.1 Outputs for BMW functions which are defined as KNOCK exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_clradkr	O/V	0...1H	0...1	1	[-]
Reset of the stored knock adaption value by service tool					
B_klopfen	O/V	0...1H	0...1	1	[-]
Knock event					
B_krakt	O/V	0...1H	0...1	1	[-]
Knock control activation (=0 inactive; 1=active)					
B_krdws	O/V	0...1H	0...1	1	[-]
Max sprark retard for knock control limb home value active (=0 inactive; 1=active)					
B_krndy	O/V	0...1H	0...1	1	[-]
Condition engine speed dynamic for knkock control					
B_warml_zykl	O/V	0...1H	0...1	1	[-]
Warmlaufzyklus abgeschlossen					
B_zwkraa	O/V	0...1H	0...1	1	[-]
Spark retard caused by torque request activ (0=active; 1=incative)					
Dzw_kr[NC_CYL_NR]	O/V	FE0C...258H	-50...60	0.1	[°CRK]
spark retard due to knock control					
Dzw_mkr	O/V	FE0C...258H	-50...60	0.1	[°CRK]
Spark retard by knock control without adaptation (average value)					
IGA_IS_TQ_KNK	O/V	0...80H	-48...0	0.375	[°CRK]
Torgue reserve for knockcontrol in idle speed					
IGA_MV_ADJ_KNK_CUS	O/V	0...80H	-48...0	0.375	[°CRK]
Mean spark retard of knock control incl. knock- and fuel quality adaptation by customer					
LV_AD_CLR_RON	V	0...1H	0...1	1	[-]
toggle bit for reseting of RON-adaption value					
LV_IGA_TRA_KNK	O/V	0...1H	0...1	1	[-]
Flag for instationary correction active					
St_knk	O/V	0...FFH	0...255	1	[-]
Injection shut off by knock pre-ignition					
Zyl_akt	O/V	0...FFH	0...255	1	[-]
Spark retard by knock					

Input data:

B_zwdyn	CYL_ID_KNK	Dzw_annm	Dzw_krkorll
ECU_STATE	IGA_BAS_COR	IGA_IGC_H_RNG[NC_CYL_NR]	IGA_KNK[NC_CYL_NR]
IGA_MV_KNK	INH_IV_KNK	KNK_CTL_DIS	LC_AD_CLR_RON
LV_KNK	LV_KNK_CTL_ENA	LV_KNK_TRA_N	LV_TQ_IGA_ACT
LV_WUP_CYC	NC_CYL_NR	OPM_AV	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_IGA_DELTA_KNK_AD_ENA	1	0...FFH	-48...47.625	0.375	[°CRK]
delta ignition angle for activation of knock adaptation					
LC_AD_IS_AFL_ACT	1	0...1H	0...1	1	[-]
KRAAN-adaption in engine operation mode "AFL" (0 - disable, 1 - enable)					

FUNCTION DESCRIPTION:

Adaption to BMW environment.

Remark: The really possible resolution and limits of the output are different to the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit power latch phase:

all output variables are initialized with 0hex

Recurrence: 10ms: B_krakt, B_krdws, B_krndy, B_warml_zykl, B_zwkraa, St_knk

every segment: B_klopfen, Zyl_akt, Dzw_mkr
the whole if-clause arround the calculation of
Dzw_kr[NC_CYL_NR]

Activation: at every engine state

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
if      ( LC_AD_CLR_RON 0 -> 1  or   LC_AD_CLR 0 -> 1
        or   tester request "alle Adaptionswerte löschen"
        or tester request "RON adaption löschen")
then    LV_AD_CLR_RON_n = 1
endif
B_clradkr                  = LV_AD_CLR_RON
B_warml_zykl               = LV_WUP_CYC
If   ECU_STATE = "PWL"
Then
    Dzw_mkr                = 0
    Zyl_akt                = 0
    Dzw_kr[NC_CYL_NR]      = 0
    B_krdws                = 0
    B_krndy                = 0
    B_zwkraa               = 0
    B_klopfen              = 0
    St_knk                 = 0
Elseif
    Dzw_mkr                =     IGA_MV_KNK
    if (0) CYL_ID_KNK = 0
    then (0)          Dzw_kr[1] = IGA_KNK[1]
    else (0)          if  (1) CYL_ID_KNK = 1
          then (1)    Dzw_kr[2] = IGA_KNK[2]
          else (1)    if  (2)CYL_ID_KNK = 2
                      then (2) Dzw_kr[3] = IGA_KNK[3]
                      else (2)    if  (3) CYL_ID_KNK = 3
                                  then (3)  Dzw_kr[4] = IGA_KNK[4]
```

```
                                        else (3)   if  (4) CYL_ID_KNK = 4
                                                   then (4)  Dzw_kr[5] = IGA_KNK[5]
                                                   else (4)   Dzw_kr[6] = IGA_KNK[6]
                                                   endif (4)
                                        endif (3)
                               endif (2)
                      endif (1)
                 endif (0)

                 B_krakt            =   LV_KNK_CTL_ENA
                 B_krdws            =   KNK_CTL_DIS
                 B_krndy            =   LV_KNK_TRA_N
                 Zyl_akt            =   CYL_ID_KNK
                 B_klopfen          =   LV_KNK
                 St_knk             =   INH_IV_KNK
                 If       LV_TQ_IGA_ACT=0
                             or    ( OPM_AV = "AFS"
                            and  (               ((IGA_BAS_COR – C_IGA_DELTA_KNK_AD_ENA)
                                            ≤ (IGA_IGC_H_RNG[0] – IGA_KNK[0]))
                                            or  ((IGA_BAS_COR – C_IGA_DELTA_KNK_AD_ENA)
                                            ≤ (IGA_IGC_H_RNG[1] – IGA_KNK[1]))
                                   ...
                                   or  ((IGA_BAS_COR – C_IGA_DELTA_KNK_AD_ENA)
                                            ≤ (IGA_IGC_H_RNG[NC_CYL_NR-1]- IGA_KNK[NC_CYL_NR-1])
                                   )
                          )
                   or     ( OPM_AV = "AFL"  and LC_AD_IS_AFL_ACT=1 )
                 Then   B_zwkraa = 1
                 Else   B_zwkraa = 0
                 Endif

             Endif
```

45.8.2 Outputs for SV aggregates

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Recurrence: 10 ms
Activation: every engine state

Formula section:

LV_IGA_TRA_KNK = B_zwdyn

IGA_IS_TQ_KNK = Dzw_krkorll

IGA_MV_ADJ_KNK_CUS = Dzw_annm

46 Lambda control

Table of Contents

Index

46.1 LACO General

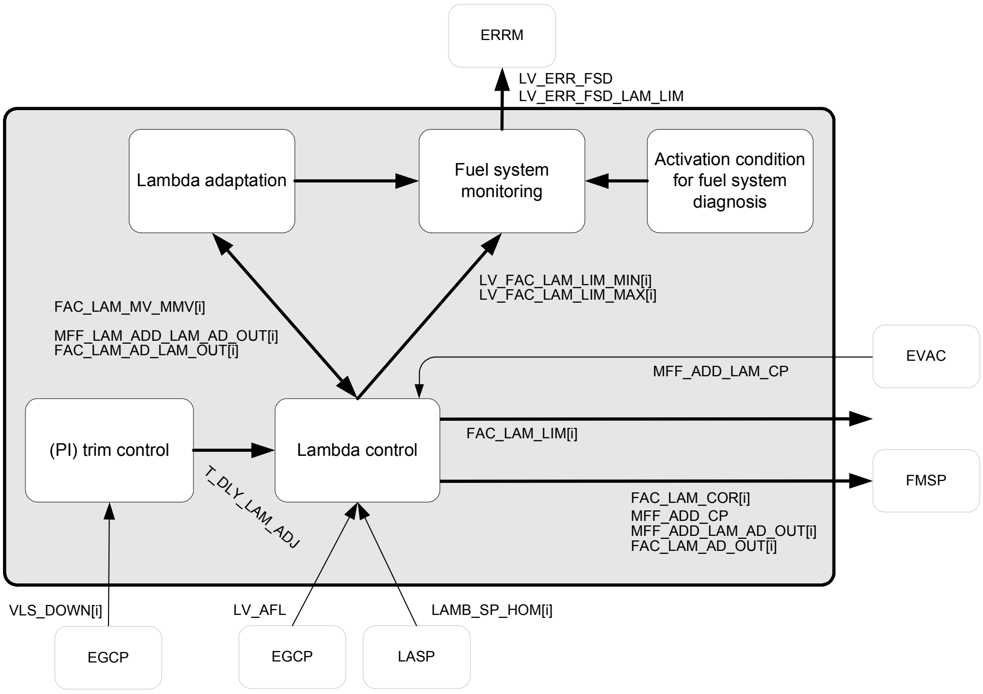
General information:

The Aggregate LACO fulfils the following tasks:
- Lambda control based von upstream oxygen sensor signal
- Trim control based on downstream oxygen sensor signal
- Lambda adaptation
- Fuel system monitoring

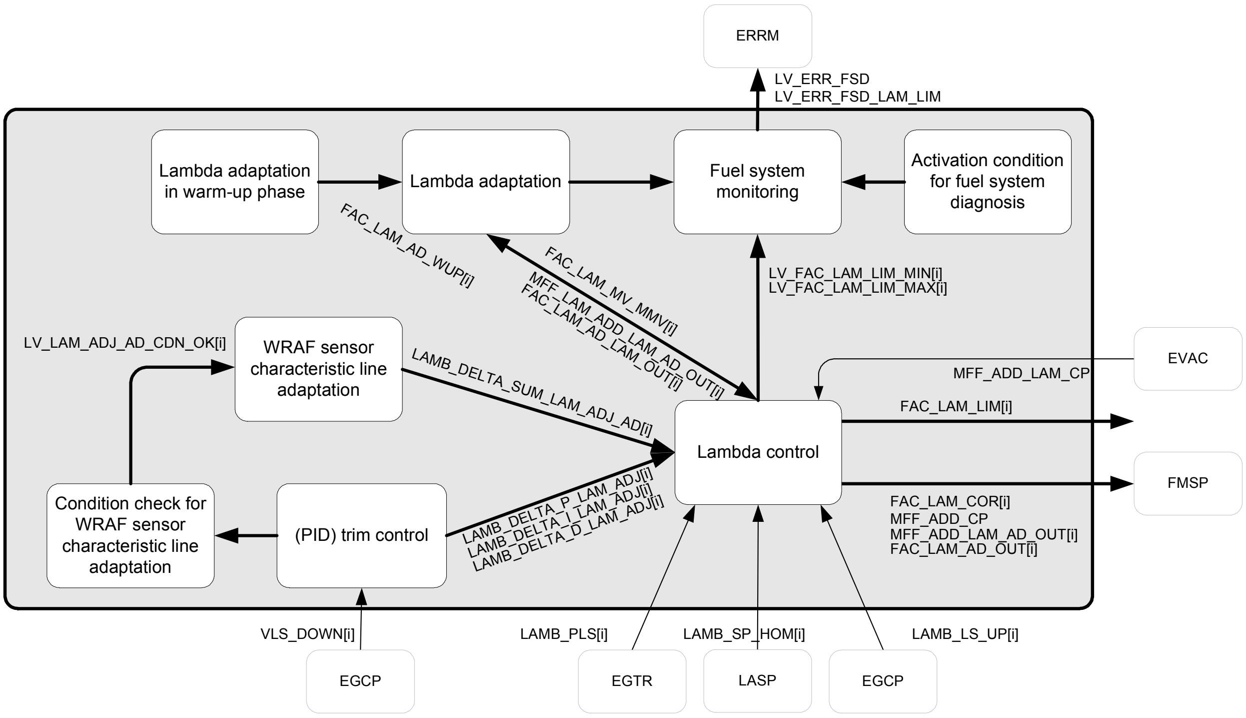
The aggregate is available in 2 variants:
- Variant 1 : bin/bin
- Variant 2 : lin/bin

Architecture Overview:

Variant 1:



Variant 2:



Functional Overview:

46.1.1 Lambda control

The Lambda Control Correction adjusts the fuel-air mixture to the desired ratio. It controls the factor FAC_LAM_COR which is adjusted to produce a leaner or a richer mixture according to the oxygen sensor information. The FAC_LAM_COR forms a multiplicative factor in the fuel mass set point calculation thus exerting direct influence on the injected quantity.

46.1.2 Lambda control conditions (Variant 2: Lin/Bin only)

This function manages the stop modes of the linear lambda controller, defines non stationary conditions, calculates the crankshaft difference between start of fuel injection and the position where the sensor detects the exhaust gas, detects whether lambda closed loop control was active at least once during the actual driving cycle and evaluates the fuel system status.

46.1.3 Lambda adaptation

In order to compensate serial production tolerances of components, an adaptive correction is calculated (one additive and two multiplicative) based on the filtered lambda controller output. Depending on load and engine speed three different adaptation fields (one offset and two factor areas) are observed. The additive and multiplicative adaptation corrections are used for calculating the injection time for all engine operating states, except at ‘engine stop’ and ‘engine start’. Lambda adaptation learning, for precision reason, needs to be performed at lambda equal 1 conditions. The function itself is activated by the corresponding LV_LAM_AD_ACT[i].

46.1.4 Lambda adaptation in warm-up phase (Variant 2: Lin/Bin only)

The Lambda adaptation in warm-up phase enhances the Lambda adaptation functionality to obtain a more precise correction at low temperatures (< 50°C). The adaptive correction is applied to compensate serial production tolerances of components at low.

TCO dependent correction factors (%) are learned in five temperature ranges. The correction factors are derived from the Lambda controller output. They are learned in an adaptation field specified by MFF_SP[i] and N_32. They are, however, to correct the injection time for all engine states. Learning of these factors requires that the Lambda controller is in the state 'ON'.

46.1.5 Dynamic fuel trim

The trim-controller causes a better observance of the catalyst window during all the life of the vehicle. The control value is determined depending on the trim control sensor signal (downstream binary signal). The difference of the sensor voltage from set point value is the basic characteristic. The set point value can be applied depending on the operating point, so that the dynamic lambda can be adjusted according to the operating range. For Variant 1 : bin/bin a PI controller is used while for Variant 2 : lin/bin an additional D share increases the control dynamics (PID controller). Variant 1 : bin/bin furthermore includes an adaptation functionality at the end of a driving cycle in order to compensate ageing effects of the upstream O2-sensor. This functionality is covered for Variant 2 : lin/bin by the module "*WRAF sensor characteristic line adaptation with trim control*" described hereafter.

46.1.6 WRAF sensor characteristic line adaptation with trim control (Variant 2: Lin/Bin only)

This function adapts the WRAF sensor characterisic line by observing the I share of the trim control function. A basic (load depending) lambda shift of the WRAF sensor can be corrected by means of an additional offset. This offset can change due to ageing of the sensor.

46.1.7 Valid cond. check for WRAF sensor char. line adap. with trim control (Variant 2: Lin/Bin only)

This function checks several conditions that must be fulfilled in order to start the WRAF sensor characteristic line adaption. These conditions shall define a stable state of the downstream sensor signal used by trim control.

46.1.8 Mean value calculation for limited dynamics

This function calculates only differences between base values and their moving means values. These differences or deltas are to be used within the function that needs limited dynamic conditions. The evaluation of the limited dynamic bit is part of the function that need this flag. So the mean value calculation is concentrated in this function.

46.1.9 Fuel system monitoring

The objective of the fuel system diagnosis is to monitor the lambda control output and the lambda adaptation values in various areas. It should also cover the PCV monitoring where the lambda control output is considered in the idle range.

Breaking the adaptation and lambda controller limits for a long time, which may have been caused by failures in the fuel or intake system will involve emission rise and therefore shall be diagnosed by fuel system diagnosis.

46.1.10 Activation of fuel system monitoring

This function checks all necessary activation conditions in order to activate fuel system diagnosis. High rate of evaporated fuel during rich warm-up phase under cold start conditions could lead to wrong failure detection and therefore a special inhibition flag for that case is generated.

46.2 LACO Configuration Data

General information :

The following describes the general rules for determination of the configuration data

46.2.1 Local configuration data

Here are listed the configuration data, which are used only in the EGCP aggregate.

Data	Value
NC_ERR_SYM_FSD_CONF	1

To be defined by project with information from customer.

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_ERR_SYM_FSD_CONF	1	1...2H	1...2	1	[-]
Number of failure locations in FSD					

46.3 AGGR LACO adaptation

46.3.1 Outputs for SV aggregates, SV internally

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MFF_AD_END	V/O	0...1H	0...1	1	[-]
logical value indicating temporary end of lambda adaptation					
LV_MFF_AD_CDN	V/O	0...1H	0...1	1	[-]
flag for time scheduler indicating good conditions for lambda adaptation					
LV_LSCL_i	V/O	0...1H	0...1	1	[-]
Activation flag of lambda control					
MFF_AD_FAC_MMV_REL_i	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Relative multiplicative adaptive factor					
MFF_AD_ADD_MMV_REL_QUO_i	V/O	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
relative additive adaptive value quotient					
TI_LAM_COR_i	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Controller output signal in consideration of the canister purge function and lambda adaptation					
LAMB_DIF_i	V/O	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
Controller difference (richness)					
LV_LAM_COR_LIM_INTR_i	V/O	0...01H	0...1	1	-
the sum output signal is limited under non-stationary operating conditions (threshold exceeded)					
LV_LAM_STOP_i	V/O	0 ... 01H	0 ... 1	1	-
activation signal for the controller stop-mode					
LAM_MV_i	V/O	8000...7FFFH	-50...49.9992	0.0015	%
sum of the controller output from the I-and I^2-share					
LAM_MV_LPF_CP_i	V/O	8000...7FFFH	-50...49.9992	0.0015	%
mean value of the controller output, used by canister purge					
LAMB_SP_FIL_i	V/O	0...7FFFH	0...32	0.98E-3	-
output signal of the lambda-sensor model					
LAMB_SP_FIL_S_CP_i	V/O	0...7FFFH	0...32	0.98E-3	-
output signal of the lambda-sensor model in stratified mode, used by canister purge					
T_MFF_AD_MIN	V/O	0...FFFFH	0...6553.5	0.1	s
Time indicating the minimum time for next requested lambda adaptation					
VLS_CYL_x	V/O	0...3FFH	0...4.99	4.88E-3	V
upstream oxygen sensor acquisition (cylinder individual)					
VLS_CYL_TRIG_x	V/O	0...FFH	0...255	1	-
cylinder selective update counter for trigger event					
LV_LAMB_OHP[NC_CBK_EX_NR]	O/V	0…1H	0…1	1	-
Flag indicating enrichment for overheating prevention					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_LAM_MV	O/V	8000...7FFFH	-50...49.998474	1.5259E-3	%
mean value of lambda controller output bank 1 and bank 2					
DELTA_LAMB_I_H_RES_i	V/O	F800...0800H	-0.125...0.125	6.106E-5	-
high word of λ-shift from I-share with high resolution					
FAC_LAM_PCTL_CUS[NC_CBK_EX_NR]	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Customer lambda pre-control value					
LV_ERR_DET_FSD[NC_CBK_EX_NR]	O	0...1H	0...1	1	[-]
Error detection flag FSD diagnosis					
FAC_LAM_AD_BAL[NC_CBK_EX_NR]	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lambda adaption output to be used in the MFF correction					
LV_LAM_AD_EXT	V/O	0...1H	0...1	1	[-]
Logical variable indicating that Lambda Adaptation is carried out outside from LACO					
LV_FAC_L_RNG_LIM_MIN_EXT_LAM_AD[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag indicating Lambda Adaptation external to LACO is under its min. limit threshold					
LV_FAC_L_RNG_LIM_MAX_EXT_LAM_AD[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
Flag indicating Lambda Adaptation external to LACO is over its max. limit threshold					
FAC_LAM_LIM_FIL[NC_CBK_EX_NR]	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Filtered Lambda Controller Output					
OBD_LAM_AD[NC_CBK_EX_NR]	V/O	0...FFH	-100...99.21875	0.78125	[%]
Lambda adaptation factor SAE J1979					

Input data:

LV_LAM_AD_END	LV_LAM_AD_CDN	LV_LAM_LSCL[NC_CBK_EX_NR]	LAMB_SP_FIL_S[NC_CBK_EX_NR]
LAMB_SP_FIL_S[NC_CBK_EX_NR]	NC_CBK_EX_NR	ERR_SYM_FSD[NC_CBK_EX_NR]	FAC_LAM_LIM[NC_CBK_EX_NR]
T_PRI_TOT_LAM_AD	LAMB_SP_HOM[NC_CBK_EX_NR]	LAMB_SP_DELTA_LAM[NC_CBK_EX_NR]	LV_LAM_STOP[NC_CBK_EX_NR]

FAC_DIF_LAM_IN[NC_CBK_EX_NR]	LAMB_SP_FIL_HOM[NC_CBK_EX_NR]	CTR_VLS_UP_CYL_SEL_TRIG[NC_CYL_NR]	FAC_DIF_LAM_IN[NC_CBK_EX_NR]
FAC_LAM_MV_MMV_CP[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_NOT_STAT_CDN[NC_CBK_EX_NR]	LV_LAM_STOP[NC_CBK_EX_NR]	FAC_LAM_MV[NC_CBK_EX_NR]
	VLS_UP_CYL_SEL[NC_CYL_NR]	LV_LAMB_COP[NC_CBK_EX_NR]	LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]
FAC_LAM_AD_OUT[NC_CBK_EX_NR]	FAC_LAM_AD_CUS[NC_CBK_EX_NR]	FAC_MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	FAC_LAM_ADJ_COR_LAM_AD_CUS[NC_CBK_EX_NR]
LV_FAC_L_RNG_LIM_MIN_LAM_AD[NC_CBK_EX_NR]	LV_FAC_L_RNG_LIM_MAX_LAM_AD[NC_CBK_EX_NR]	LV_FAC_LIM_MIN_LAM_AD_CUS[NC_CBK_EX_NR]	LV_FAC_LIM_MAX_LAM_AD_CUS[NC_CBK_EX_NR]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_LAM_AD_EXT	1	0...1H	0...1	1	[-]
Logical switch between LACO internal (=0) and external (=1) Lambda Adaptation source					
LC_FAC_LAM_LIM_FIL_ENA	1	0...1H	0...1	1	[-]
Logical switch to enable filtering of the lambda controller output					
IP_CRLC_FAC_LAM_LIM_FIL	8*8	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_N_IP_CRLC_FAC_LAM_LIM_FIL	8	0...1FE0H	0...8160	1	[rpm]
LDP_MAF_IP_CRLC_FAC_LAM_LIM_FIL	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Correlation constant for liltering the lambda controller output					

FUNCTION DESCRIPTION:

Adaptation to aggregate environment.

i = 1 ... NC_CBK_EX_NR

Application conditions:

Initialisation: 0

Recurrence : same recurrence as corresponding input data

Activation: *every engine operating state*

Formula section:

LV_MFF_AD_END = LV_LAM_AD_END

LV_MFF_AD_CDN = LV_LAM_AD_CDN

LV_LSCL_i = LV_LAM_LSCL[i]

TI_LAM_COR_i = FAC_LAM_LIM[i]

LAMB_DIF_i = FAC_DIF_LAM_IN[NC_CBK_EX_NR]

LV_LAM_COR_LIM_INTR_i = LV_FAC_LAM_LIM_NOT_STAT_CDN[NC_CBK_EX_NR]

LV_LAM_STOP_i = LV_LAM_STOP[NC_CBK_EX_NR]

LAM_MV_i = FAC_LAM_MV[NC_CBK_EX_NR]

LAM_MV_LPF_CP_i = FAC_LAM_MV_MMV_CP[NC_CBK_EX_NR]

LAMB_SP_FIL_i = LAMB_SP_FIL_HOM[NC_CBK_EX_NR]

LAMB_SP_FIL_S_CP_i = LAMB_SP_FIL_S[NC_CBK_EX_NR]

T_MFF_AD_MIN = T_PRI_TOT_LAM_AD
VLS_CYL_x = VLS_UP_CYL_SEL[x]
VLS_CYL_TRIG_x = CTR_VLS_UP_CYL_SEL_TRIG[x]
LV_LAMB_OHP[i] = LV_LAMB_COP[i]
FAC_LAM_MV = (FAC_LAM_MV[0] + FAC_LAM_MV[1]) / 2
DELTA_LAMB_I_H_RES_i = LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]
FAC_LAM_PCTL_CUS[i] = (LAMB_SP_DELTA_LAM[i]/LAMB_SP_HOM[i]) * 100%

Note: Interface to BMW Lambda Adaptation (KLANN)

LV_LAM_AD_EXT = LC_LAM_AD_EXT

```
If LV_LAM_AD_EXT = 1
Then
    FAC_LAM_AD_BAL[i]=
            FAC_LAM_AD_CUS[i] + FAC_LAM_ADJ_COR_LAM_AD_CUS[i]
    OBD_LAM_AD[i] = FAC_LAM_AD_CUS[i]  %please mind the variable definition differences
    LV_FAC_L_RNG_LIM_MIN_EXT_LAM_AD[i] = LV_FAC_LIM_MIN_LAM_AD_CUS[i]
    LV_FAC_L_RNG_LIM_MAX_EXT_LAM_AD[i] = LV_FAC_LIM_MAX_LAM_AD_CUS[i]
Else
    FAC_LAM_AD_BAL[i] =
            FAC_LAM_AD_OUT[i] + FAC_LAM_ADJ_COR_LAM_AD_CUS[i]
    OBD_LAM_AD[i] = FAC_MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]
    %please mind the variable definition differences
    LV_FAC_L_RNG_LIM_MIN_EXT_LAM_AD[i] = LV_FAC_L_RNG_LIM_MIN_LAM_AD[i]
    LV_FAC_L_RNG_LIM_MAX_EXT_LAM_AD[i]= LV_FAC_L_RNG_LIM_MAX_LAM_AD[i]
Endif
```

```
If      ERR_SYM_FSD_i ≠ 0
Then    LV_ERR_DET_FSD[i] = 1
Else    LV_ERR_DET_FSD[i] = 0
Endif
```

Note: Filtering the lambda controller output (requested for the N54 engine)

```
If LC_FAC_LAM_LIM_FIL_ENA = 1
Then
    $FAC\_LAM\_LIM\_FIL[i]_k = FAC\_LAM\_LIM\_FIL[i]_{k-1}*(1-IP\_CRLC\_FAC\_LAM\_LIM\_FIL) + FAC\_LAM\_LIM[i]*IP\_CRLC\_FAC\_LAM\_LIM\_FIL$
Else
    FAC_LAM_LIM_FIL[i] = FAC_LAM_LIM[i]
```

Endif

46.3.2 Activation flag for HOM-mode due to difference in lambda values in stratified

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CYL_BAL_LAM_AD_REQ_CUS	O/V	0...1H	0...1	1	[-]
Request bit for fast adaption due to lamb shift in stratified mode or FSD error					
LV_LAMB_DIF_AFL_ACT	V	0...1H	0...1	1	[-]
Steady state condition for lamb signal is reached					
T_LAMB_DIF_AFL_STEADY[NC_CBK_EX_NR]	V	0...9F6H	0...255	0.1	[s]
Timer indicating how long la_abgas is within boundaries					
T_LAMB_ADD_AFL	V	0...FFFFH	0...6553.5	0.1	[s]
Timer indicating how long LV_CYL_BAL_LAM_AD_REQ_CUS is already requested to deactivate					
LV_CYL_BAL_LAM_AD_REQ_CUS_FSD	V	0...1H	0...1	1	[-]
Request bit for fast lamb adaption due to FSD error					
LV_CYL_BAL_LAM_AD_REQ_CUS_LAM	V	0...1H	0...1	1	[-]
Request bit for fast lamb adaption due to lamb shift in stratified mode					

Input data:

LAMB_SP_S_EXT		LV_S_ACT	LAMB_LS_UP_MV
T_DLY_SOI_LSL_POS[NC_CBK_EX_NR]	T1_LSL_UP[NC_CBK_EX_NR]	MFF_SP[NC_CBK_EX_NR]	LV_IGK
NC_IDX_DIAG_FSD	STATE_MIL	TCO	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_DIF_AFL_STEADY	1	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambda deviation allowed to define lambda as steady state					
C_T_DIF_AFL_STEADY	1	0...9F6H	0...255	0.1	[s]
Timer adjustment to monitor steady state of la_abgas					
C_T_MAX_LAMB_ADD_AFL	1	0...FFFFH	0...6553.5	0.1	[s]
Timer adjustment to end LV_CYL_BAL_LAM_AD_REQ_CUS request					
C_N_BOL_LAMB_DIF_ACT	1	0...FFH	0...8160	32	[rpm]
minimum engine speed for hom switching					
C_N_TOL_LAMB_DIF_ACT	1	0...FFH	0...8160	32	[rpm]
maximum engine speed for hom switching					
ID_MFF_SP_BOL_LAMB_DIF_ACT	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_32_ID_MFF_SP_BOL_LAMB_DIF	8	0...FFH	0...8160	32	[rpm]
minimum fuel mass set point for hom switching					
ID_MFF_SP_TOL_LAMB_DIF_ACT	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_32_ID_MFF_SP_TOL_LAMB_DIF	8	0...FFH	0...8160	32	[rpm]
maximum fuel mass set point for hom switching					
IP_LAMB_DIF_THD	8	0...7FFFH	0...31.99902	0.9766e-3	[-]
LDP_LAMB_LS_UP_MV	8	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda deviation in stratified mode causing hom request					
C_T_MAX_LAMB_ADD_AFL_FSD	1	0...FFFFH	0...6553.5	0.1	[s]
Timer adjustment to end LV_CYL_BAL_LAM_AD_REQ_CUS request due to FSD error					
C_TCO_LAMB_DIF_ACT	1	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature activation condition for HOM request					
LC_CYL_BAL_LAM_AD_REQ_FSD	1	0...1H	0...1	1	[-]
Activation Flag to Request Lambda Adaption due to FSD Error					

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <NC_IDX_DIAG_FSD[i]>, OUT<PendingStatus>)

ACTION_ERRM_GetLvEndDiag (IN<NC_IDX_DIAG_FSD[i]>, OUT<EndFlag>)

FUNCTION DESCRIPTION:

In case of drift between lamb setpoint and lamb sensor signal in stratified mode after a long time of not adapting, hom mode is requested and canister purge is inhibited.

i = 1 ... NC_CBK_EX_NR

n: current calculation cycle; n-1: calculation cycle before current one

Application conditions:

Initialisation: LV_IGK 0 -> 1: all variables set to zero

Recurrence : 100ms

Activation: LV_DC = 1

Formula section:

```
% Setting LV_CYL_BAL_LAM_AD_REQ_CUS = 1:
If(0)          LV_S_ACT = 1                                                and
               LV_CYL_BAL_LAM_AD_REQ_CUS = 0

% Observation of steady state condition for lamb signal:
If(1a)         |LAMB_SP_S_EXT(n)-LAMB_SP_S_EXT(n-1)|<
                                                        C_LAMB_DIF_AFL_STEADY
Then(1a)       increment T_LAMB_DIF_AFL_STEADY[i] cyl. bank selective
          If(2a)     (T_LAMB_DIF_AFL_STEADY[1] >= T_DLY_SOI_LSL_POS[1] +
                                T1_LSL_UP[1] + C_T_DIF_AFL_STEADY      and
                     T_LAMB_DIF_AFL_STEADY[2] >= T_DLY_SOI_LSL_POS[2] +
                                T1_LSL_UP[2] + C_T_DIF_AFL_STEADY        )
          Then(2a)   LV_LAMB_DIF_AFL_ACT = 1
          Else(2a)   LV_LAMB_DIF_AFL_ACT = 0
          Endif(2a)
Else(1a)       T_LAMB_DIF_AFL_STEADY[i]    = 0
               LV_LAMB_DIF_AFL_ACT = 0
Endif(1a)

% Setting of HOM-Request flag:
If(1b)         |LAMB_SP_S_EXT – LAMB_LS_UP_MV| > IP_LAMB_DIF_THD and
               N_32 > C_N_BOL_LAMB_DIF_ACT
                     and
               N_32 < C_N_TOL_LAMB_DIF_ACT                                 and
               (MFF_SP[1] > ID_MFF_SP_BOL_LAMB_DIF_ACT                     or
               MFF_SP[2] > ID_MFF_SP_BOL_LAMB_DIF_ACT            )         and
               (MFF_SP[1] < ID_MFF_SP_TOL_LAMB_DIF_ACT
                     or
               MFF_SP[2] < ID_MFF_SP_TOL_LAMB_DIF_ACT            )         and
               LV_LAMB_DIF_AFL_ACT = 1                                     and
               TCO >= C_TCO_LAMB_DIF_ACT
Then(1b)       LV_CYL_BAL_LAM_AD_REQ_CUS_LAM = 1
Else(1b)       LV_CYL_BAL_LAM_AD_REQ_CUS_LAM = 0
Endif(1b)
Endif(0)  %end of LV_S_ACT condition
```

```
% FSD pending status condition for HOM-request:
If(1c)          LV_CYL_BAL_LAM_AD_REQ_CUS = 0                         and
                LC_CYL_BAL_LAM_AD_REQ_FSD = 1
Then(1c)
        For(2c) i=1 to NC_CBK_EX_NR
        ACTION_ERRM_CheckPendingStatus( <NC_IDX_DIAG_FSD[i]>, <PendingStatus> )
        ACTION_ERRM_GetLvEndDiag( <NC_IDX_DIAG_FSD[i]>, <EndFlag> )
                if(3c) (PendingStatus==1or STATE_MIL==1) and EndFlag==0
                then(3c)        LV_CYL_BAL_LAM_AD_REQ_CUS_FSD = 1
                else(3c)        LV_CYL_BAL_LAM_AD_REQ_CUS_FSD = 0
                endif(3c)
        endfor(2c)
EndIf(2c)

% General Decision about HOM-Request Flag:
If(1d)          LV_CYL_BAL_LAM_AD_REQ_CUS = 0
        If(2d)          LV_CYL_BAL_LAM_AD_REQ_CUS_LAM = 1             or
                        LV_CYL_BAL_LAM_AD_REQ_CUS_FSD = 1
        Then(2d)        LV_CYL_BAL_LAM_AD_REQ_CUS = 1
        Else(2d)        do nothing
        Endif(2d)
EndIf(2d)

% ReSetting LV_CYL_BAL_LAM_AD_REQ_CUS = 0:
If(1e)          LV_CYL_BAL_LAM_AD_REQ_CUS = 1
Then(1e)        increment T_LAMB_ADD_AFL
        If(2e)          (T_LAMB_ADD_AFL  >= C_T_MAX_LAMB_ADD_AFL      and
                        LV_CYL_BAL_LAM_AD_REQ_CUS_FSD = 0             and
                        LV_CYL_BAL_LAM_AD_REQ_CUS_LAM = 1)            or
                        (T_LAMB_ADD_AFL  >= C_T_MAX_LAMB_ADD_AFL_FSD  and
                        LV_CYL_BAL_LAM_AD_REQ_CUS_FSD = 1             and
                        LV_CYL_BAL_LAM_AD_REQ_CUS_LAM = 0)            or
        Then(2e)        LV_CYL_BAL_LAM_AD_REQ_CUS = 0
        Endif(2e)
Else(1e)        T_LAMB_ADD_AFL  = 0
Endif(1e)
```

46.4 Lambda control conditions for linear lambda control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CRK_DIF_SOI_LSL_POS[NC_CBK_EX_NR]	O/V	0...10EH	0...1.62E+3	6	°CRK
crank shaft angle difference between the start of injection and the end of the exhaust cycle					
LV_LAM_STOP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Activation signal for the lambda controller stop mode					
LV_LAM_NOT_STAT_CDN	O/V	0...1H	0...1	1	-
flag for stop mode activation of lambda controller by wall film					
LV_LAM_STOP_SHO_PER[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
activation signal for the controller stop mode while active offset adjustment					
STATE_LS[NC_CBK_EX_NR]	O/V	1H 2H 4H 8H 10H	OL_CDN CL OL_INTR OL_ERR CL_ERR	1	-
Fuel system status					
T_TOUT_FAC_LAM_IN_CHK[NC_CBK_EX_NR]	O/V	0...FFH	0...5.1	0.02	s
Timer for controller difference consideration to reactivate the lambda controller after stop mode					
LV_LAM_ACT_DC[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
closed loop had been enabled for at least one time					
MAF_INT_LAM_REAC[NC_CBK_EX_NR]	V	0...FFFFH	0...1.82042E+3	0.02777783	g
air mass flow for lambda controller reactivation after stop mode					
T_LAM_NOT_STAT_CDN	V	0...FFFFH	0...1.3107E+3	0.02	s
time counter for lambda controller reactivation after wall film					
T_LAM_REAC[NC_CBK_EX_NR]	V	0...FFH	0...5.1	0.02	s
time counter for lambda controller reactivation after stop mode					

Input data:

LV_LAM_LSCL[NC_CBK_EX_NR]	FAC_DIF_LAM_IN[NC_CBK_EX_NR]	LV_LAM_STOP_REQ[NC_CBK_EX_NR]	LV_STATE_LS_OPL_ERR[NC_CBK_EX_NR]
FAC_NEUT_NOT_STAT_CDN	LV_MAF_INT_MIN_LAM_REAC_SWI	LV_LAM_STOP_SHO_PER_REQ[NC_CBK_EX_NR]	CAM_EX[NC_NR_CBK_IVVT]
CRK_DIF_SOI_IGN	C_CAM_OP_EX	LV_FCUT_IND	LV_PUC
LV_ST_END	MAF_CYL	NC_CBK_EX_NR	NC_NR_CBK_IVVT
TCO	LV_LSL_UP_SHO_PER_REQ[NC_CBK_EX_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_DIF_LAM_IN_MIN_LAM_REAC	1	0...7FFFH	0...0.99996948	3.05176E-5	-
Threshold for controller difference to reactivate lambda controller after stop mode					
C_MAF_INT_MIN_LAM_REAC	1	0...FFFFH	0...1.82042E+3	0.02777783	g
threshold for the air mass flow integral for reactivation of the lambda controller					
C_MAF_INT_MIN_LAM_REAC_OPT	1	0...FFFFH	0...1.82042E+3	0.02777783	g
optional threshold for the air mass flow integral for reactivation of the lambda controller					
C_T_MIN_LAM_REAC	1	0...FFH	0...5.1	0.02	s
time delay for lambda controller reactivation after stop mode					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MIN_TOUT_FAC_LAM_IN_CHK	1	0...FFH	0...5.1	0.02	s
Time-out threshold for controller difference consideration to reactivate the lambda controller after stop mode					
IP_FAC_NEUT_MAX_NOT_STAT_CDN	6	0...FFFFH	-1...0.99996948	3.05176E-5	-
LDPM_TCO_1_LACO	6	0...FEH	-48...142.5	0.75	°C
maximum threshold for detection of non-stationary conditions for lambda controller					
IP_FAC_NEUT_MIN_NOT_STAT_CDN	6	0...FFFFH	-1...0.99996948	3.05176E-5	-
LDPM_TCO_1_LACO	6	0...FEH	-48...142.5	0.75	°C
minimum threshold for detection of non-stationary conditions for lambda controller					
IP_T_LAM_NOT_STAT_CDN	6	0...FFFFH	0...1.3107E+3	0.02	s
LDPM_TCO_1_LACO	6	0...FEH	-48...142.5	0.75	°C
time threshold to reactivate the lambda controller after wall film					

46.4.1 General information

This function

- manages the stop modes of the linear lambda controller,
- defines non stationary conditions,
- calculates the crankshaft difference between start of fuel injection and the position where the sensor detects the exhaust gas,
- detects whether lambda closed loop control was active at least once during the actual driving cycle and
- evaluates the fuel system status.

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

Application Condition

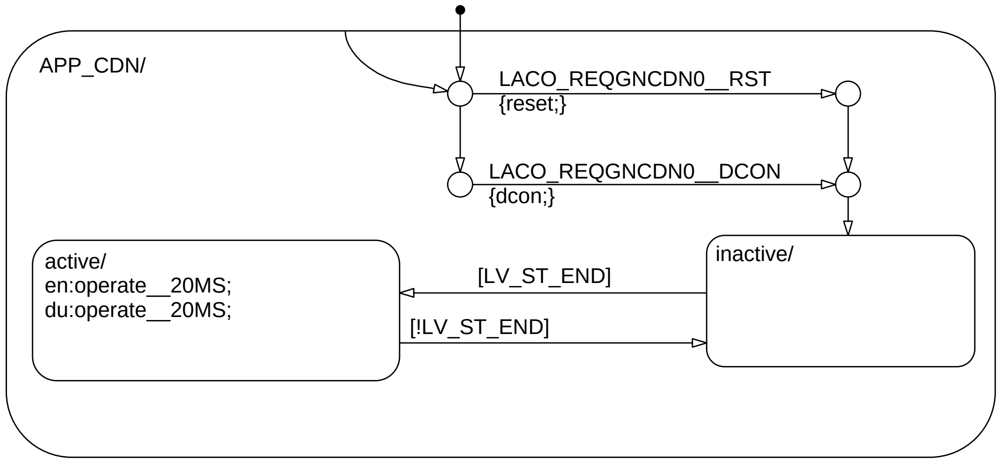


Figure 1 LACO_REQGNCDN0/ APP_CDN/ Chart

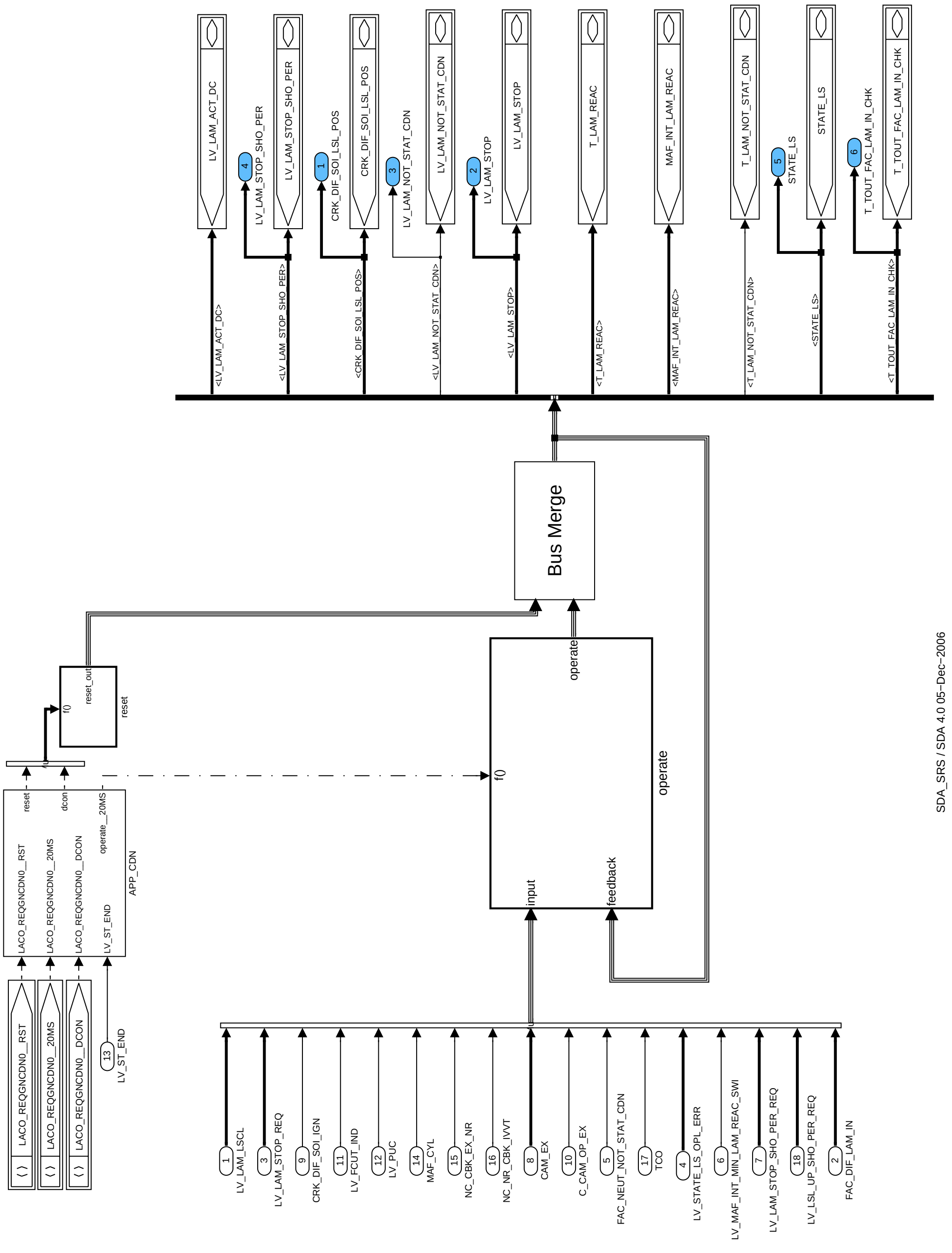


Figure 2 LACO_REQGNCDN0

46.4.1.1 Formula section

Exhaust bank specific functionality

Calculation of LV_LAM_STOP[i], LV_LAM_STOP_SHO_PER[i], CRK_DIF_SOI_LSL_POS[i], LV_LAM_LSCL[i], LV_LAM_ACT_DC[i], STATE_LS[i] and CTR_RAF_CHG[i]

Calculation of LV_LAM_STOP[i] and LV_LAM_STOP_SHO_PER[i]

The lambda-controller is in the stop-mode (LV_LAM_STOP[i] = 1) if one of the conditions

a) the offset-adjustment of the lambda-sensor is active

(LV_LSL_OFS_ADJ_ACT[i] = 1)

b) Switching of the pump current measurement range (gain-switching) of the lambda-sensor is currently taking place (LV_VLS_UP_INIT[i] = 1)

c) Ip gain-switch of the lambda-sensor is not yet confirmed by the hardware (LV_SWI_GAIN_LSL_IF[i] ≠ STATE_LSL_IF_SPI_RD[i] (*bit* SWI_GAIN))

d) OBDI signal failed (LV_LSL_DEAC[i] = 1)

e) one cylinder at least is cut off (LV_FCUT_IND = 1)

f) the engine state LV_PUC is active

g) lambda controller stop mode request (LV_LAM_STOP_REQ[i] = 1)

is fulfilled.

Additionally LV_LAM_STOP_SHO_PER[i] is set to 1 in case that condition a), b), c) or d) is true (short period stop mode).

LV_LAM_STOP_SHO_PER[i] is set to 0 if condition e), f) or g) is true (nominal stop mode). These latter conditions have the higher priority if a) or b) are fulfilled too.

If condition a) and b) are not fulfilled anymore the timer T_LAM_REAC[i] is to start. If this timer exceeds a calibration threshold, LV_LAM_STOP[i] and LV_LAM_STOP_SHO_PER[i] are set both to 0.

If condition e), f) and g) are not fulfilled anymore the integral MAF_INT_LAM_REAC[i] is calculated. If this integral exceeds a calibration threshold and the controller difference FAC_DIF_LAM_IN[i] is below a threshold while a time out threshold is not exceeded, LV_LAM_STOP[i] is set to 0.

For the calcuation of the air mass flow integral the unit conversion:

[g] = [kg/h] * [ms] * [(g*h) / (kg*ms)] / 3600

has to be considered.

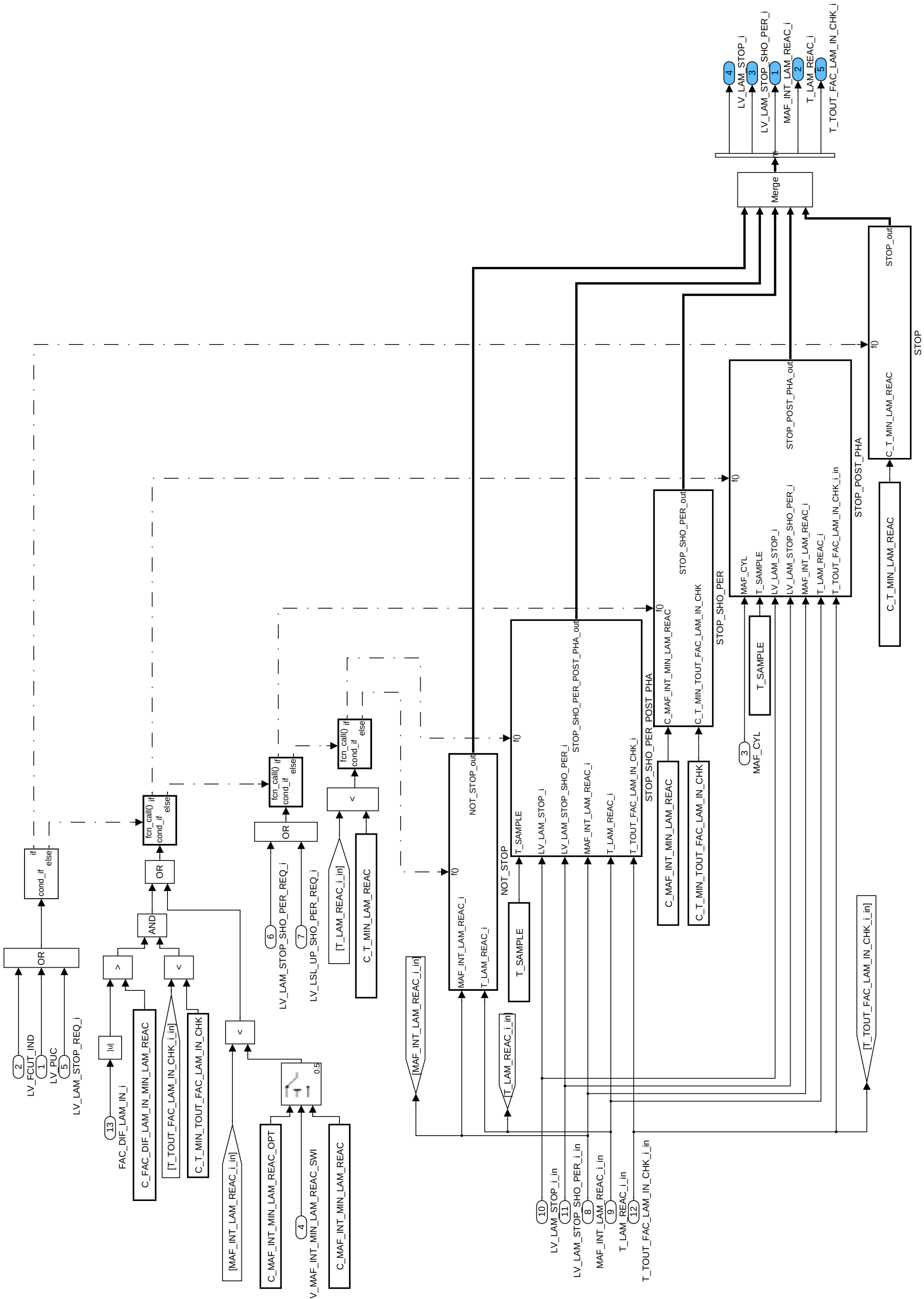


Figure 3 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ MNG_LAM_STOP

No stop mode

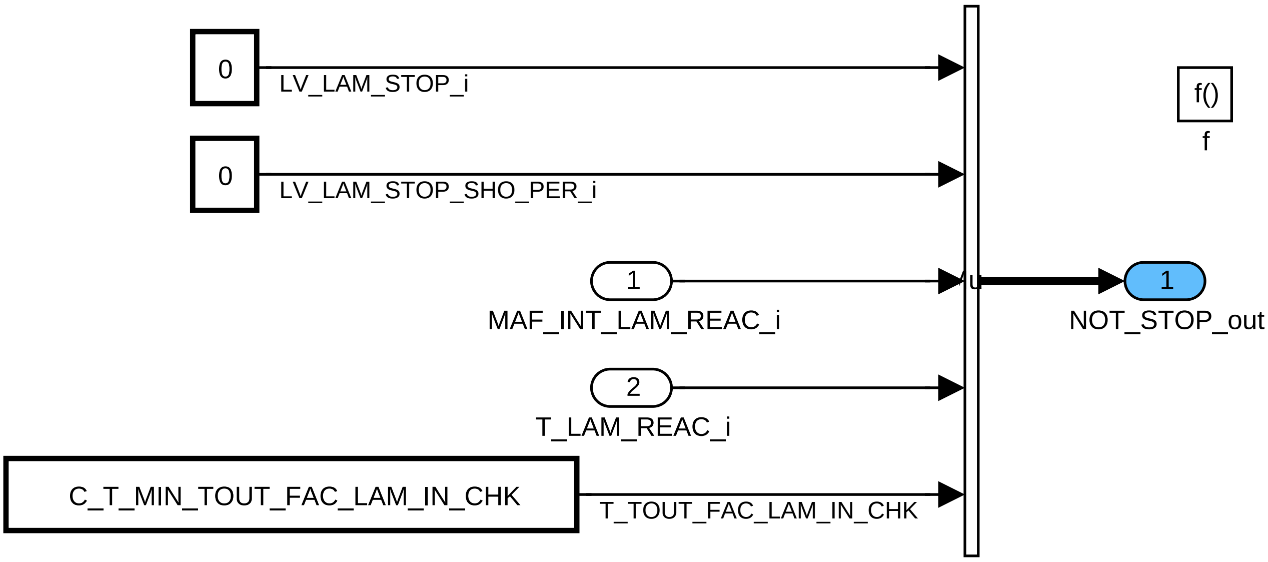


Figure 4 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ MNG_LAM_STOP/ NOT_STOP

Nominal stop mode

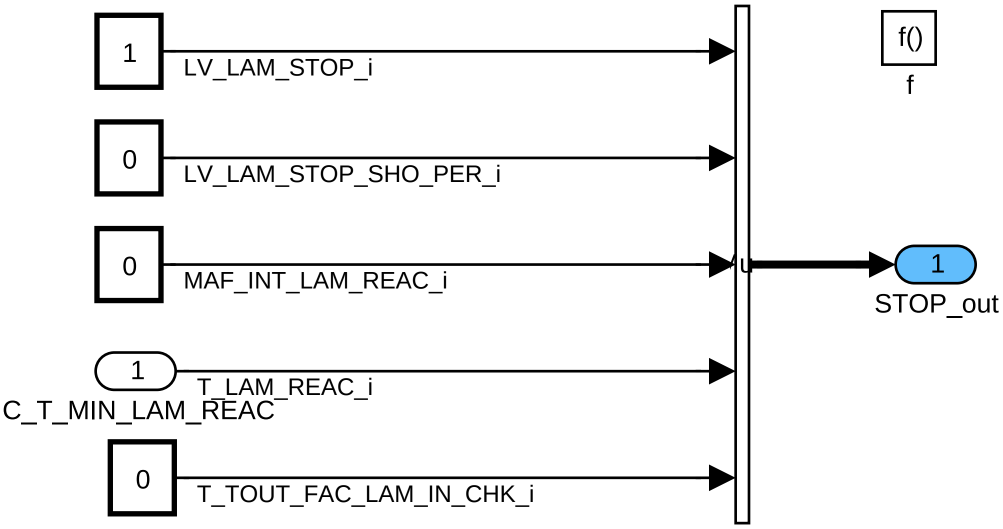


Figure 5 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ MNG_LAM_STOP/ STOP

Post phase of nominal stop mode

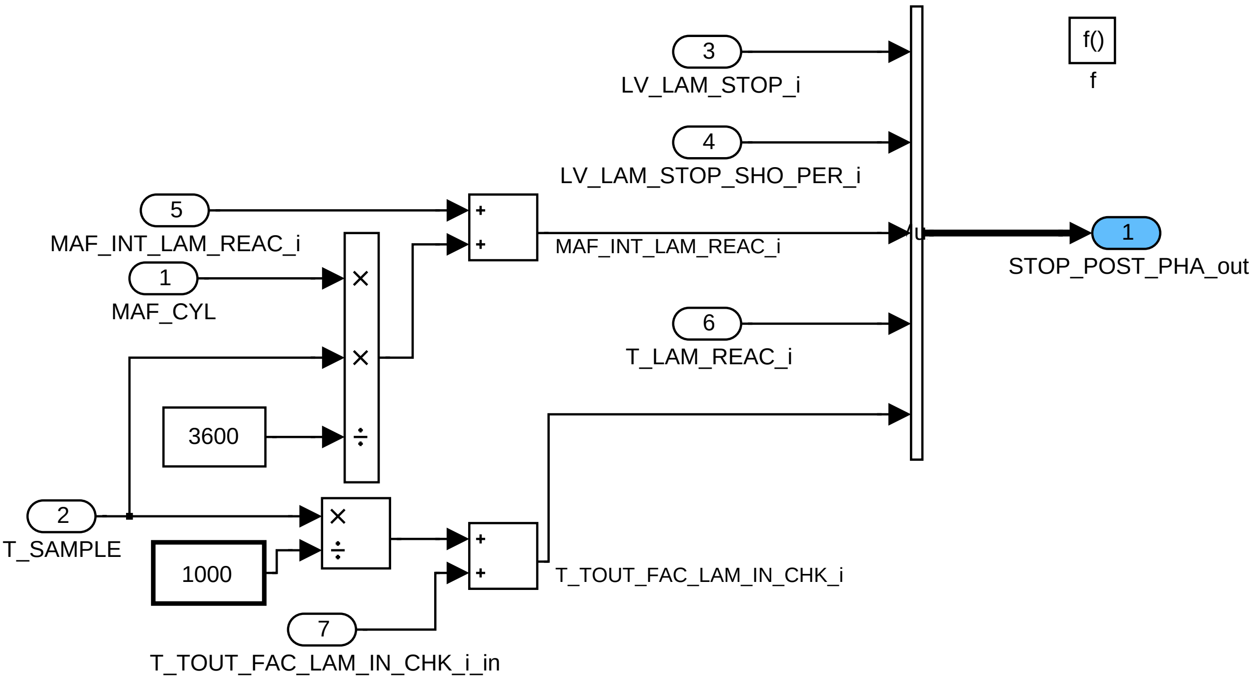


Figure 6 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ MNG_LAM_STOP/ STOP_POST_PHA

Short period stop mode

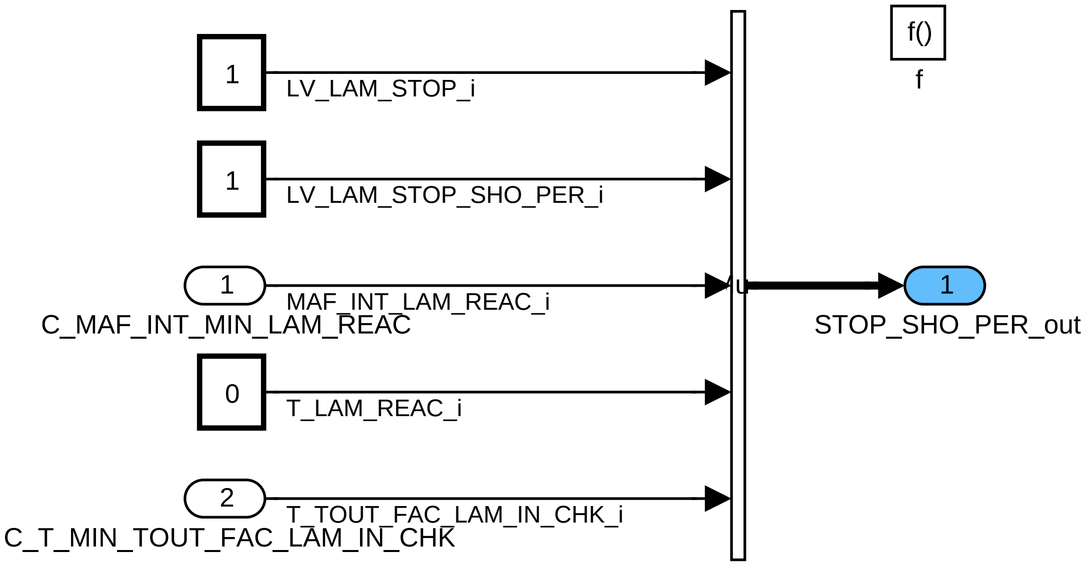


Figure 7 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ MNG_LAM_STOP/ STOP_SHO_PER

Post phase of short period stop mode

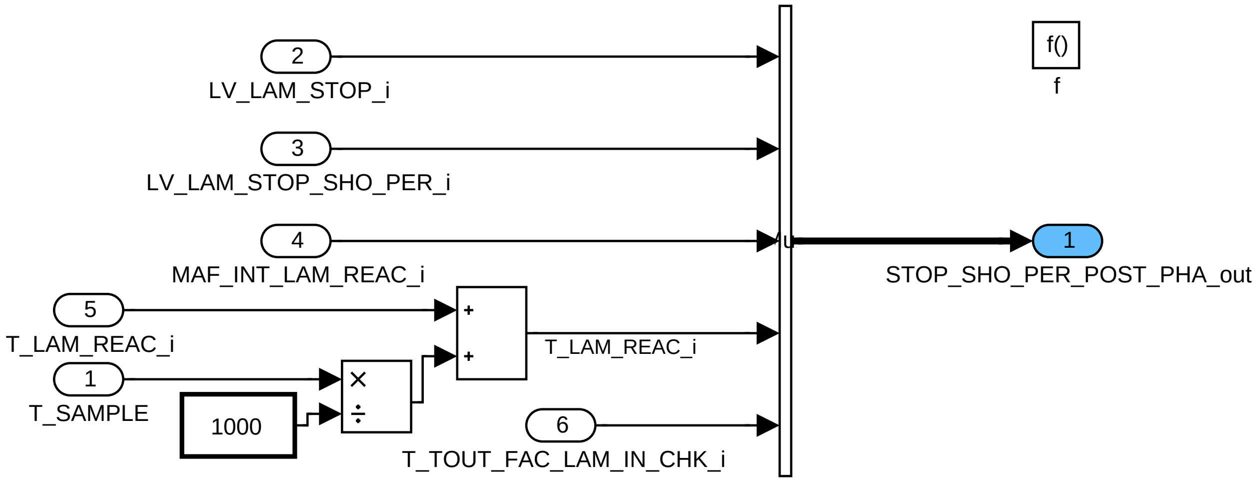


Figure 8 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ MNG_LAM_STOP/ STOP_SHO_PER_POST_PHA

Calculation of CRK_DIF_SOI_LSL_POS[i]

The value of CRK_DIF_SOI_LSL_POS[i] corresponds to the crankshaft angle difference between the start of injection and the end of the exhaust cycle of the engine. This value is required for the calculation of the gas delay needed for the linear lambda controller. The distance between start of injection and top dead center where the injected fuel is ignited is defined by CRK_DIF_SOI_IGN (see injection phase for homogeneous and stratified mode). The distance from the mentioned top dead center to the point where the exhaust valves close is defined by 360°CRK plus current position of exhaust camshaft (CAM_EX[i]; is always negative) minus half of the opening period. So the opening timing edge of the exhaust valve is met.

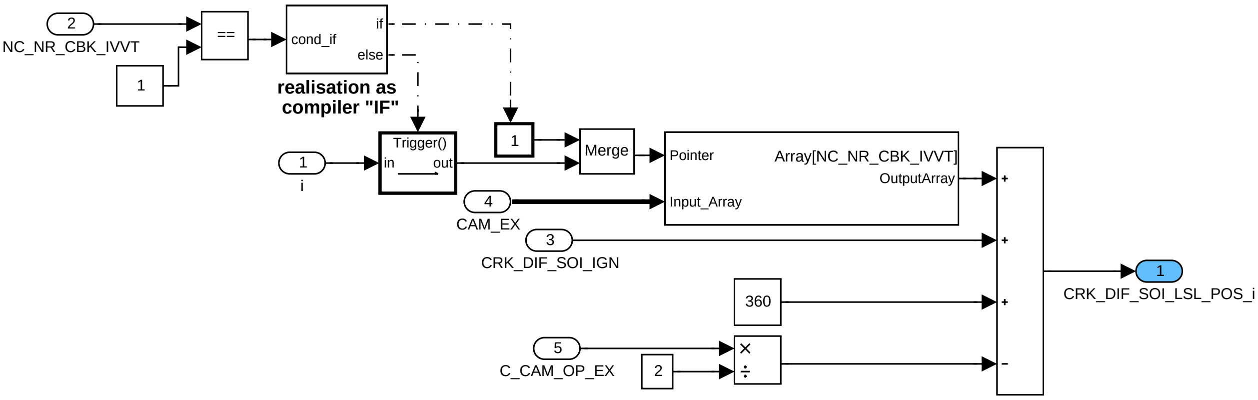


Figure 9 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ CLC_CRK_DIF_SOI_LSL_POS

Calculation of LV_LAM_ACT_DC[i]

The flag LV_LAM_ACT_DC[i] shows that lambda closed loop control was active at least once during the actual driving cycle.

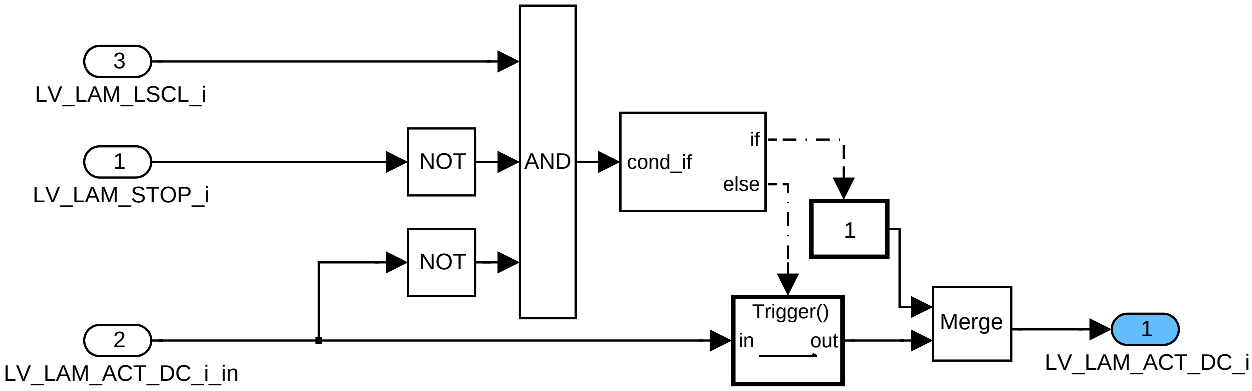
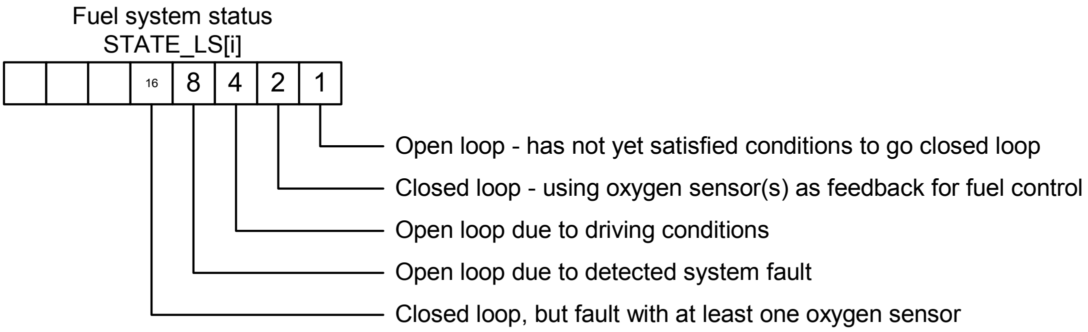


Figure 10 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ DET_LAM_ACT_DC

Evaluation of fuel system status STATE_LS[i]

The fuel system status byte indicated the status (open / closed loop) of the cylinder bank. It belongs to the freeze frame requested by law SAE J1979. State 16 is not supported.



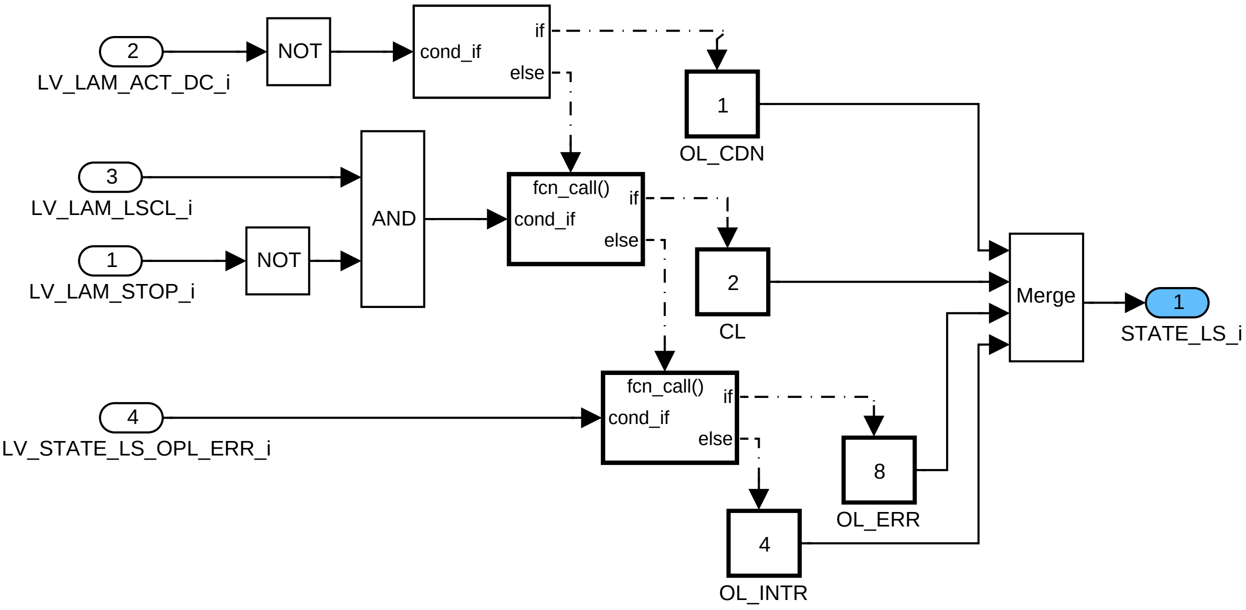


Figure 11 LACO_REQGNCDN0/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ CLC_STATE_LS

Definition of non stationary conditions

To avoid big controller excursion under non stationary conditions like wall film the output limitation of the lambda controller can be restricted. Therefore the flag LV_LAM_NOT_STAT_CDN is set if the neutral value FAC_NEUT_NOT_STAT_CDN exceeds the calibration minimum or maximum. The limitation itself (reaction on this flag) is executed in the module linear lambda control.

If the sum of the wall film is lower than this calibration threshold a counter is decreased. If the counter is 0 then the lambda-controller works without the limitation again.

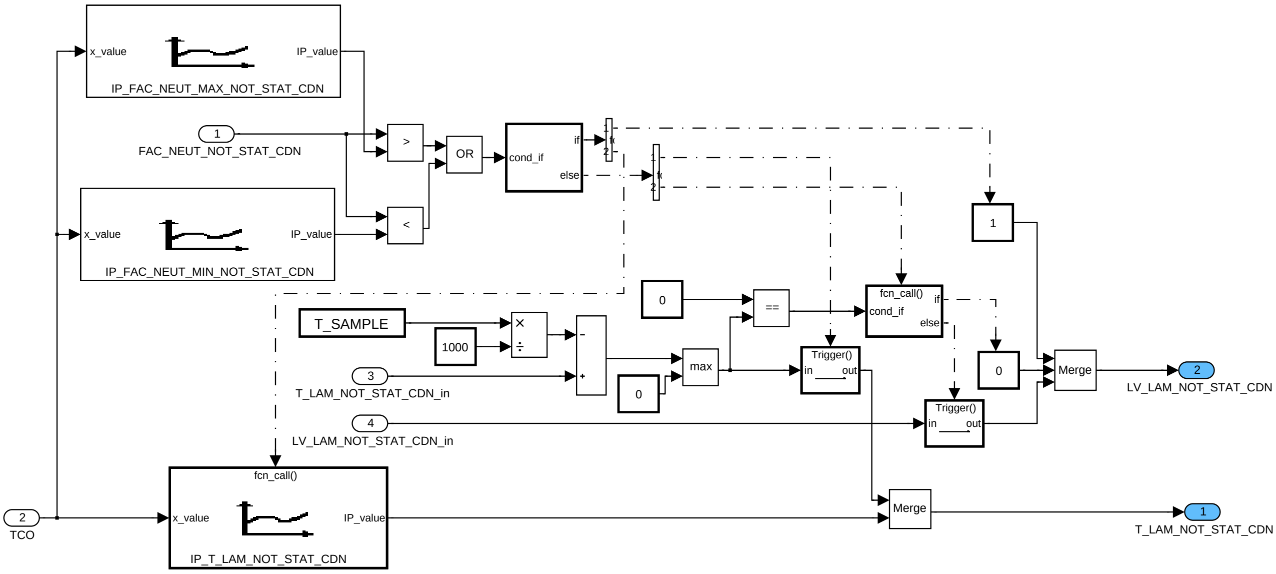


Figure 12 LACO_REQGNCDN0/ operate/ CLC_LAM_NOT_STAT_CDN/ LAM_NOT_STAT_CDN

46.4.1.2 Initialization

STATE_LS[i] is set to "OL_CDN".

T_LAM_REAC[i] is set to C_T_MIN_LAM_REAC.

MAF_INT_LAM_REAC[i] is set to C_MAF_INT_MIN_LAM_REAC.

T_TOUT_FAC_LAM_IN_CHK[i is set to C_T_MIN_TOUT_FAC_LAM_IN_CHK.]

All other output variables are set to 0.

46.5 Lambda control for WRAF sensor

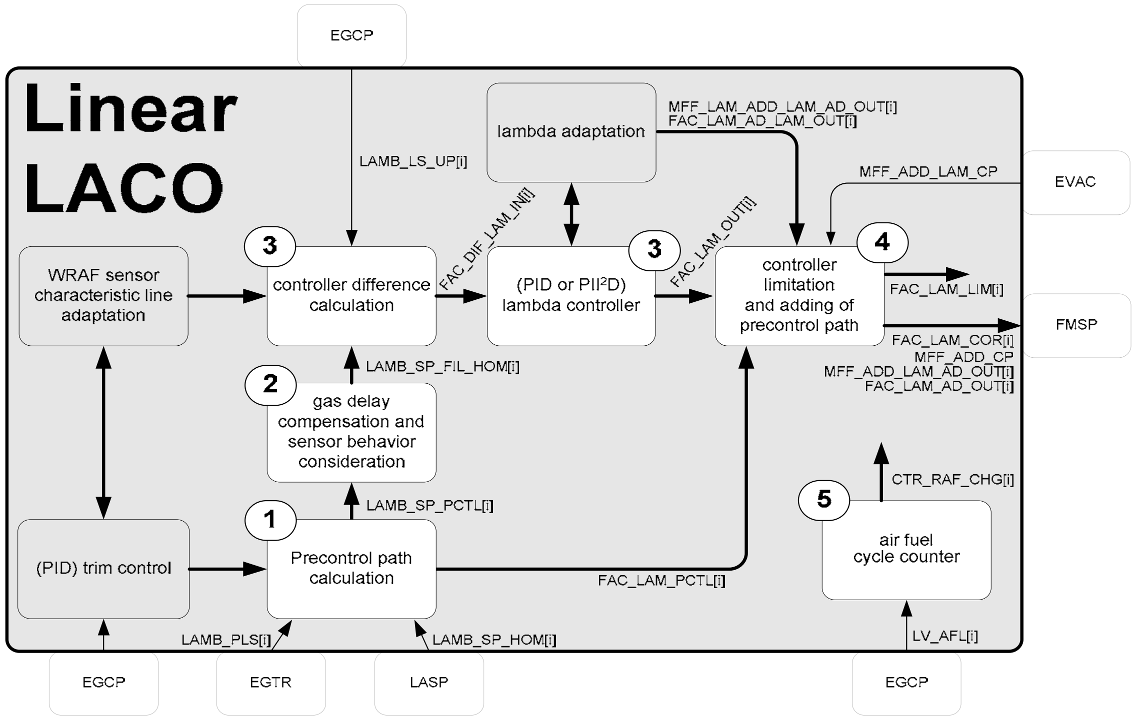
Overview

The linear lambda controller adjusts the air fuel ratio depending on the engine operating state. The controller is designed as a 4th order controller containing P, I, I^2 and D share. Beyond this also a PID controller can be calibrated. The measuring signal of the air fuel ratio is generated by a continuous (WRAF) O_2 sensor. An interface to lambda adaptation, canister purge function and evaporative system monitoring function permits the adjustment of the lambda controller output signal and internal state variables. Furthermore mean values of the output signal are calculated and provided to other functions.

This module consists of 5 sub-functionalities:

1. Precontrol path calculation (including set point determination)
2. Exhaust gas delay compensation and sensor behavior consideration
3. Closed loop control; PII^2D or PID controller including controller difference calculation (main functionality)
4. Controller output limitation and precontrol path correction consideration
5. Calculation of air fuel ratio changes counter (auxiliary function used by other modules)

The figure below gives an overview about the single sub-functions of this module.



Lambda control functions in the LACO context.

LACO_REQGNLACO0 used as model name defines unambiguously this whole module.

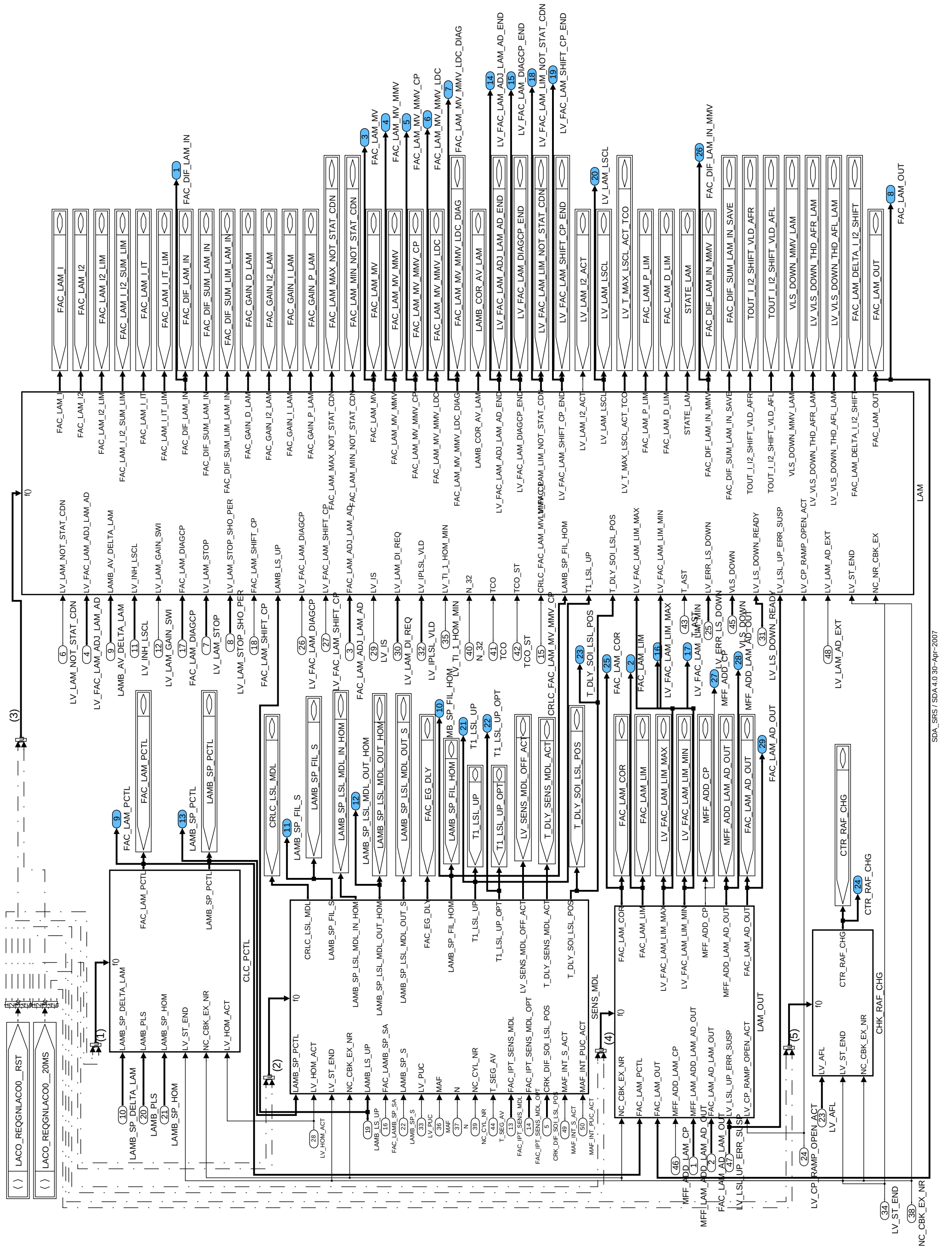


Figure 13 LACO_REQGNLACO0

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_DIF_LAM_IN[NC_CBK_EX_NR]	O/V	8000...7FFFH	-1...0.999969	3.05176E-5	-
controller difference (richness)					
FAC_LAM_LIM[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
limited value of lambda controller output					
FAC_LAM_MV[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
mean value of lambda controller output					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_LAM_MV_MMV[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
FAC_LAM_MV[i] average value					
FAC_LAM_MV_MMV_CP[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
mean value of lambda controller output, used by canister purge					
FAC_LAM_MV_MMV_LDC[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
moving mean value of the controller output, used for limited dynamics calculation					
FAC_LAM_MV_MMV_LDC_DIAG[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
Mean value of the controller output, used by limited dynamics for cat efficiency diagnosis					
FAC_LAM_OUT[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
lambda controller output					
FAC_LAM_PCTL[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
pre control path correction					
LAMB_SP_FIL_HOM[NC_CBK_EX_NR]	O/V	0...7FFFH	0...1.999939	6.10352E-5	-
output signal of the WRAF sensor model for homogeneous mode					
LAMB_SP_FIL_S[NC_CBK_EX_NR]	O/V	0...7FFFH	0...31.9990234	9.76563E-4	-
output signal of the WRAF sensor model for stratified mode					
LAMB_SP_LSL_MDL_OUT_HOM[NC_CBK_EX_NR]	O/V	0...FFFFH	0...3.99993896	6.10352E-5	-
output signal of the WRAF sensor model in homogeneous mode					
LAMB_SP_PCTL[NC_CBK_EX_NR]	O/V	0...7FFFH	0...1.999939	6.10352E-5	-
pre control lambda set point					
LV_FAC_LAM_ADJ_LAM_AD_END[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
lambda shift required by lambda adaptation was carried out					
LV_FAC_LAM_DIAGCP_END[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
controller shift caused by EVAP diagnosis was carried out					
LV_FAC_LAM_LIM_MAX[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
upper limit of lambda controller output reached					
LV_FAC_LAM_LIM_MIN[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
lower limit of lambda controller output reached					
LV_FAC_LAM_LIM_NOT_STAT_CDN[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
limitation under non stationary operating conditions (threshold exceeded)					
LV_FAC_LAM_SHIFT_CP_END[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Controller shift caused by canister purge was carried out					
LV_LAM_LSCL[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
activation flag of lambda control					
T1_LSL_UP[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1.99996948	3.05176E-5	s
first order time lag of the WRAF sensor model					
T1_LSL_UP_OPT[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1.99996948	3.05176E-5	s
first order time lag of the WRAF sensor model (optional)					
T_DLY_SOI_LSL_POS[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1.023984	1.5625E-5	s
delay between start of injection and WRAF sensor signal location					
CTR_RAF_CHG[NC_CBK_EX_NR]	O/V	0...FFH	0...255	1	-
counter of air fuel ratio changes					
FAC_LAM_COR[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
limited lambda controller output plus pre control correction					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_DIF_LAM_IN_MMV[NC_CBK_EX_NR]	O/V	80000000...7FFFFFFFH	-1...1	4.6566E-10	-
controller difference (richness) moving mean value					
MFF_ADD_CP	O/V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Canister Purge Fuel Mass correction corresponding to Lambda Factor Shift					
MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	O/V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
fuel mass set point offset, output from lambda adaptation					
FAC_LAM_AD_OUT[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
fuel mass set point factor, output from lambda adaptation					
VLS_DOWN_MMV_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...4.99992371	7.62939E-5	V
moving mean value of downstream sensor voltage					
LV_T_MAX_LSCL_ACT_TCO[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
lambda controller activation by timer condition instead of TCO condition					
LAMB_COR_AV_LAM[NC_CBK_EX_NR]	V	0...7FFFH	0...1.999939	6.10352E-5	-
corrected signal of WRAF sensor; actual value for lambda controller					
FAC_LAM_MAX_NOT_STAT_CDN[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limit in case of limitation under non stationary conditions					
FAC_LAM_MIN_NOT_STAT_CDN[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limit in case of limitation under non stationary conditions					
FAC_GAIN_I_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...49.9992371	7.62939E-4	1/s
gain of lambda controller I share					
FAC_EG_DLY[NC_CBK_EX_NR]	V	0...FFH	0...31.875	0.125	-
weighting factor for the calculation of the exhaust gas delay					
FAC_GAIN_P_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...15.999756	2.44141E-4	-
gain of lambda controller P share					
FAC_GAIN_D_LAM[NC_CBK_EX_NR]	V	0...7FFFH	0...0.03999878	1.2207E-6	s
gain of lambda controller D share					
FAC_GAIN_I2_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...24.9996185	3.8147E-4	1/s²
gain of lambda controller I^2 share					
FAC_DIF_SUM_LAM_IN[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-6.5536E+4 ... 6.5536E+4	3.05176E-5	-
unlimited summation of controller difference					
FAC_DIF_SUM_LIM_LAM_IN[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-6.5536E+4 ... 6.5536E+4	3.05176E-5	-
limited summation of controller difference					
FAC_LAM_I_IT_LIM[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-50...50	2.32831E-8	%
limited internal I share used for PID control only					
FAC_LAM_I2_LIM[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
limited I^2 share output signal					
FAC_LAM_I_I2_SUM_LIM[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
limited I share + I^2 share output signal					
FAC_LAM_I[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
unlimited I share output signal					
LV_VLS_DOWN_THD_AFL_LAM[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that downstream sensor signal is below calibration threshold (used for lambda control)					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_LAM_I2[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
unlimited I^2 share output signal					
FAC_LAM_I_IT[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-50...50	2.32831E-8	%
unlimited internal I share used for PID control only					
FAC_LAM_P_LIM[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
limited P share output signal					
FAC_LAM_D_LIM[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
limited D share output signal					
STATE_LAM[NC_CBK_EX_NR]	V	0H 1H 2H 3H	OFF ON STOP_SHO_PER STOP	1	-
lambda controller state					
FAC_DIF_SUM_LAM_IN_SAVE[NC_CBK_EX_NR]	V	80000000...7FFFFFFFH	-6.5536E+4 ... 6.5536E+4	3.05176E-5	-
stored value of controller difference summation					
TOUT_I_I2_SHIFT_VLD_AFR[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
time out for I and I^2 share shift based on rich downstream signal					
FAC_LAM_DELTA_I_I2_SHIFT[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
I and I^2 share shift based on downstream signal evaluation					
CRLC_LSL_MDL[NC_CBK_EX_NR]	V	0...FFFFH	0...0.99998474	1.52588E-5	-
correlation constant for lambda sensor model					
TOUT_I_I2_SHIFT_VLD_AFL[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
time out for I and I^2 share shift based on lean downstream signal					
LAMB_SP_LSL_MDL_IN_HOM[NC_CBK_EX_NR]	V	0...FFFFH	0...3.99993896	6.10352E-5	-
lambda controller set point including forced stimulation and the influence of secondary air					
LAMB_SP_LSL_MDL_OUT_S[NC_CBK_EX_NR]	V	0...7FFFH	0...31.9990234	9.76563E-4	-
output signal of the WRAF sensor model in stratified mode					
LV_VLS_DOWN_THD_AFR_LAM[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that downstream sensor signal is above calibration threshold (used for lambda control)					
LV_SENS_MDL_OFF_ACT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Flag indicating that lambda sensor model switch-off is active.					
T_DLY_SENS_MDL_ACT[NC_CBK_EX_NR]	V	0...FFH	0...5.1	0.02	s
Delay timer for reactivation of linear lamdba sensor model					
LV_LAM_I2_ACT	V	0...1H	0...1	1	-
flag to switch between PII^2D and PID lambda controller					
FAC_1_FAC_T_DLY_EG[NC_CBK_EX_NR]	-	0...400H	0...1	9.76563E-4	-
internal factor of Pade filter calculation					
FAC_1_T_DLY_EG[NC_CBK_EX_NR]	-	0...FFFFH	0...63.9990234	9.76563E-4	-
internal factor of Pade filter calculation					
FAC_2_FAC_T_DLY_EG[NC_CBK_EX_NR]	-	F800...800H	-2...2	9.76563E-4	-
internal factor of Pade filter calculation					
FAC_2_T_DLY_EG[NC_CBK_EX_NR]	-	0...FFFFFH	0...1.024E+3	9.76563E-4	-
internal factor of Pade filter calculation					

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>FAC_ADD_LAM_I2[NC_CBK_EX_NR]</td><td>-</td><td>8000...7FFFH</td><td>-50...49.9984741</td><td>0.00152588</td><td>%</td></tr>
<tr><td colspan="6">part that is added to the I^2 share every sample step</td></tr>
<tr><td>FAC_DIF_LAM_IN_OLD[NC_CBK_EX_NR]</td><td>-</td><td>8000...7FFFH</td><td>-1...0.999969</td><td>3.05176E-5</td><td>-</td></tr>
<tr><td colspan="6">controller difference (richness) (N-1)</td></tr>
<tr><td>FAC_LAM_D[NC_CBK_EX_NR]</td><td>-</td><td>8000...7FFFH</td><td>-50...49.9984741</td><td>0.00152588</td><td>%</td></tr>
<tr><td colspan="6">unlimited D share output signal</td></tr>
<tr><td>FAC_LAM_I_I2_SUM_MAX[NC_CBK_EX_NR]</td><td>-</td><td>8000...7FFFH</td><td>-50...49.9984741</td><td>0.00152588</td><td>%</td></tr>
<tr><td colspan="6">upper limitation for I + I^2 share</td></tr>
<tr><td>FAC_LAM_I_I2_SUM_MIN[NC_CBK_EX_NR]</td><td>-</td><td>8000...7FFFH</td><td>-50...49.9984741</td><td>0.00152588</td><td>%</td></tr>
<tr><td colspan="6">lower limitation for I share + I^2 share</td></tr>
<tr><td>FAC_LAM_OUT_MAX[NC_CBK_EX_NR]</td><td>-</td><td>8000...7FFFH</td><td>-50...49.9984741</td><td>0.00152588</td><td>%</td></tr>
<tr><td colspan="6">upper limit for lambda controller output</td></tr>
<tr><td>FAC_LAM_OUT_MIN[NC_CBK_EX_NR]</td><td>-</td><td>8000...7FFFH</td><td>-50...49.9984741</td><td>0.00152588</td><td>%</td></tr>
<tr><td colspan="6">bottom limit for lambda controller output</td></tr>
<tr><td>FAC_LAM_P[NC_CBK_EX_NR]</td><td>-</td><td>8000...7FFFH</td><td>-50...49.9984741</td><td>0.00152588</td><td>%</td></tr>
<tr><td colspan="6">unlimited P share output signal</td></tr>
<tr><td>LAMB_DIF_1_EG_DLY[NC_CBK_EX_NR]</td><td>-</td><td>E0000000...20000000H</td><td>-16...16</td><td>2.98023E-8</td><td>-</td></tr>
<tr><td colspan="6">internal lambda difference of Pade filter calculation</td></tr>
<tr><td>LAMB_DIF_1_FAC_EG_DLY[NC_CBK_EX_NR]</td><td>-</td><td>C0000000...40000000H</td><td>-32...32</td><td>2.98023E-8</td><td>-</td></tr>
<tr><td colspan="6">internal lambda difference of Pade filter calculation</td></tr>
<tr><td>LAMB_DIF_2_EG_DLY[NC_CBK_EX_NR]</td><td>-</td><td>F0000000...10000000H</td><td>-8...8</td><td>2.98023E-8</td><td>-</td></tr>
<tr><td colspan="6">internal lambda difference of Pade filter calculation</td></tr>
<tr><td>LAMB_DIF_2_FAC_EG_DLY[NC_CBK_EX_NR]</td><td>-</td><td>C0000000...40000000H</td><td>-32...32</td><td>2.98023E-8</td><td>-</td></tr>
<tr><td colspan="6">internal lambda difference of Pade filter calculation</td></tr>
<tr><td>LAMB_LSL_MDL_INI</td><td>-</td><td>0...FFFFH</td><td>0...3.99993896</td><td>6.10352E-5</td><td>-</td></tr>
<tr><td colspan="6">Theoretical lambda set point for initialization of linear lambda sensor model</td></tr>
<tr><td>LAMB_LSL_MDL_INP[NC_CBK_EX_NR]</td><td>-</td><td>0...FFFFFFFFH</td><td>0...32</td><td>7.45058E-9</td><td>-</td></tr>
<tr><td colspan="6">input variable for lambda sensor model</td></tr>
<tr><td>LAMB_LSL_MDL_OUT[NC_CBK_EX_NR]</td><td>-</td><td>0...FFFFFFFFH</td><td>0...32</td><td>7.45058E-9</td><td>-</td></tr>
<tr><td colspan="6">output variable for lambda sensor model</td></tr>
<tr><td>LAMB_LSL_MDL_OUT_HOM_OLD_1[NC_CBK_EX_NR]</td><td>-</td><td>0...FFFFH</td><td>0...3.99993896</td><td>6.10352E-5</td><td>-</td></tr>
<tr><td colspan="6">internal output variable (N-1) for lambda sensor model for homogeneous mode</td></tr>
<tr><td>LAMB_LSL_MDL_OUT_HOM_OLD_2[NC_CBK_EX_NR]</td><td>-</td><td>0...FFFFH</td><td>0...3.99993896</td><td>6.10352E-5</td><td>-</td></tr>
<tr><td colspan="6">internal output variable (N-2) for lambda sensor model for homogeneous mode</td></tr>
<tr><td>LAMB_LSL_MDL_OUT_S_OLD_1[NC_CBK_EX_NR]</td><td>-</td><td>0...7FFFH</td><td>0...31.9990234</td><td>9.76563E-4</td><td>-</td></tr>
<tr><td colspan="6">internal output variable (N-1) for lambda sensor model for stratified mode</td></tr>
<tr><td>LAMB_LSL_MDL_OUT_S_OLD_2[NC_CBK_EX_NR]</td><td>-</td><td>0...7FFFH</td><td>0...31.9990234</td><td>9.76563E-4</td><td>-</td></tr>
<tr><td colspan="6">internal output variable (N-2) for lambda sensor model for stratified mode</td></tr>
<tr><td>LAMB_SP_FIL_IT[NC_CBK_EX_NR]</td><td>-</td><td>0...40000000H</td><td>0...32</td><td>2.98023E-8</td><td>-</td></tr>
<tr><td colspan="6">internal output signal of the WRAF sensor model</td></tr>
</table>

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_SP_FIL_IT_OLD[NC_CBK_EX_NR]	-	0...40000000H	0...32	2.98023E-8	-
internal output signal (N-1) of the WRAF sensor model					
LV_AFL_OLD[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
Auxiliary flag for edge detection of LV_AFL[i]					
LV_LAM_NOT_STAT_CDN_OLD[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
auxiliary flag for edge detection of LV_LAM_NOT_STAT_CDN					
LV_SENS_MDL_HOM_ACT_OLD[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
Auxiliary flag for edge detection of LV_HOM_ACT					
LV_VLS_DOWN_THD_AFL_LAM_OLD[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
auxiliary flag used for downstream sensor signal evaluation					
LV_VLS_DOWN_THD_AFR_LAM_OLD[NC_CBK_EX_NR]	-	0...1H	0...1	1	-
auxiliary flag used for downstream sensor signal evaluation					
T_LSCL_ACT_TCO[NC_CBK_EX_NR]	-	0...FFFFH	0...6.5535E+4	1	s
timer for lambda controller activation					

Input data:

MFF_LAM_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	FAC_LAM_AD_LAM_OUT[NC_CBK_EX_NR]	FAC_LAM_ADJ_LAM_AD[NC_CBK_EX_NR]	LV_FAC_LAM_ADJ_LAM_AD[NC_CBK_EX_NR]
CRK_DIF_SOI_LSL_POS[NC_CBK_EX_NR]	LV_LAM_NOT_STAT_CDN	LV_LAM_STOP[NC_CBK_EX_NR]	LV_LAM_STOP_SHO_PER[NC_CBK_EX_NR]
LAMB_AV_DELTA_LAM[NC_CBK_EX_NR]	LAMB_SP_DELTA_LAM[NC_CBK_EX_NR]	LV_INH_LSCL[NC_CBK_EX_NR]	LV_LAM_GAIN_SWI
FAC_IPT_SENS_MDL[NC_CBK_EX_NR]	FAC_IPT_SENS_MDL_OPT[NC_CBK_EX_NR]	CRLC_FAC_LAM_MV_MMV_CP	FAC_LAMB_SP_SA
FAC_LAM_DIAGCP[NC_CBK_EX_NR]	FAC_LAM_SHIFT_CP	LAMB_LS_UP[NC_CBK_EX_NR]	LAMB_PLS[NC_CBK_EX_NR]
LAMB_SP_HOM[NC_CBK_EX_NR]	LAMB_SP_S	LV_AFL[NC_CBK_EX_NR]	LV_CP_RAMP_OPEN_ACT
LV_ERR_LS_DOWN[NC_CBK_EX_NR]	LV_FAC_LAM_DIAGCP	LV_FAC_LAM_SHIFT_CP	LV_HOM_ACT
LV_IS	LV_LAM_DI_REQ[NC_CBK_EX_NR]	LV_LS_DOWN_READY[NC_CBK_EX_NR]	LV_IPLSL_VLD[NC_CBK_EX_NR]
LV_PUC	LV_ST_END	LV_TI_1_HOM_MIN	MAF
N	NC_CBK_EX_NR	NC_CYL_NR	N_32
TCO	TCO_ST	T_AST	T_SEG_AV
VLS_DOWN[NC_CBK_EX_NR]	MFF_ADD_LAM_CP	LV_LSL_UP_ERR_SUSP[NC_CBK_EX_NR]	LV_LAM_AD_EXT
MAF_INT_S_ACT	MAF_INT_PUC_ACT		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_FAC_DIF_LAM_IN_MMV	1	0...FFH	0...0.99609375	0.00390625	-
correlation constant for the calculation of FAC_DIF_LAM_IN_MMV[i]					
C_CRLC_FAC_LAM_MV	1	0...FFH	0...0.99609375	0.00390625	-
correlation constant for calculation of the controller output mean value FAC_LAM_MV[i]					
C_CRLC_FAC_LAM_MV_MMV	1	0...FFH	0...0.99609375	0.00390625	-
correlation constant for the calculation of the filtered controller output mean value FAC_LAM_MV_MMV[i]					
C_CRLC_FAC_LAM_MV_MMV_LDC	1	0...FFH	0...0.99609375	0.00390625	-
correlation constant for the calculation of moving mean value for limited dynamics detection					

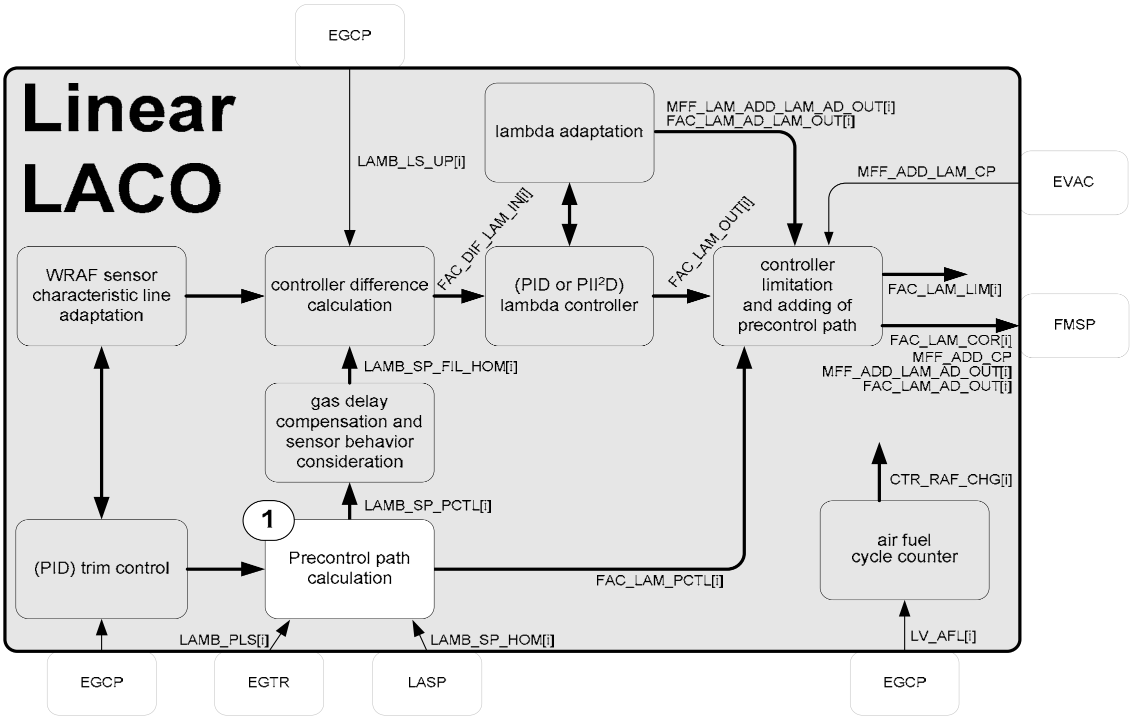
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_FAC_LAM_MV_MMV_LDC_DIAG	1	0...FFH	0...0.99609375	0.00390625	-
correlation constant for mean value calculation of lambda controller output, used by limited dynamics for cat efficiency diagnosis					
C_CRLC_VLS_DOWN_MMV_LAM	1	0...FFH	0...0.99609375	0.00390625	-
correlation constant for the calculation of VLS_DOWN_MMV_LAM[i]					
C_FAC_LAM_DELTA_LSCL_OFF	1	0...7FFFH	0...49.9984741	0.00152588	%
output signal delta to ramp the output signal to zero in case of deactivated lambda controller					
C_FAC_LAM_D_MAX_LAM	1	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limitation of the lambda controller D share					
C_FAC_LAM_D_MIN_LAM	1	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limitation of the lambda controller D share					
C_FAC_LAM_I_I2_SUM_MAX_LAM	1	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limitation of lambda controller I share + I^2 share					
C_FAC_LAM_I_I2_SUM_MAX_LAM_CP	1	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limitation of lambda controller I share + I^2 share during active canister purge					
C_FAC_LAM_I_I2_SUM_MIN_LAM	1	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limitation of lambda controller I share + I^2 share					
C_FAC_LAM_I_I2_SUM_MIN_LAM_CP	1	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limitation of lambda controller I share + I^2 share during active canister purge					
C_FAC_LAM_OUT_MAX	1	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limitation of the lambda controller output					
C_FAC_LAM_OUT_MAX_CP	1	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limitation of the lambda controller output during canister purge ramp open phase					
C_FAC_LAM_OUT_MAX_ERR_SUSP	1	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limitation of the lambda controller output during oxygen sensor error suspicion					
C_FAC_LAM_OUT_MIN	1	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limitation of the lambda controller output					
C_FAC_LAM_OUT_MIN_CP	1	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limitation of the lambda controller output during canister purge ramp open phase					
C_FAC_LAM_OUT_MIN_ERR_SUSP	1	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limitation of the lambda controller output during oxygen sensor error suspicion					
C_FAC_LAM_P_MAX_LAM	1	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limitation of lambda controller P share					
C_FAC_LAM_P_MAX_LAM_CP	1	8000...7FFFH	-50...49.9984741	0.00152588	%
upper limitation of lambda controller P share during active canister purge					
C_FAC_LAM_P_MIN_LAM	1	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limitation of lambda controller P share					
C_FAC_LAM_P_MIN_LAM_CP	1	8000...7FFFH	-50...49.9984741	0.00152588	%
bottom limitation of lambda controller P share during active canister purge					
C_LAMB_SENS_MDL_OFF_PUC	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Lambda signal threshold during PUC to switch-off lambda sensor model					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MIN_LAM_LSCL	1	0...FFH	0...8.16E+3	32	rpm
minimum engine speed for lambda controller activation					
C_TOUT_I_I2_SHIFT_VLD	1	0...FFFFH	0...1.3107E+3	0.02	s
time out for I and I^2 share shift based on downstream signal evaluation					
C_T_DLY_SENS_MDL_ACT	1	0...FFH	0...5.1	0.02	s
Time delay for reactivation of linear lamdba sensor model					
C_VLS_DOWN_THD_AFL_LAM	1	0...FFFFH	0...4.99992371	7.62939E-5	V
lean threshold of downstream sensor signal for I and I^2 share correction					
C_VLS_DOWN_THD_AFR_LAM	1	0...FFFFH	0...4.99992371	7.62939E-5	V
rich threshold of downstream sensor signal for I and I^2 share correction					
LC_LAM_I2_ACT	1	0...1H	0...1	1	-
calibration flag to switch between PII^2D and PID lambda controller					
ID_TCO_MIN_LAM_LSCL	6	0...FEH	-48...142.5	0.75	°C
LDPM_TCO_ST_1_LACO	6	0...FEH	-48...142.5	0.75	°C
minimum TCO for lambda controller activation					
ID_TCO_MIN_LAM_LSCL_IS	6	0...FEH	-48...142.5	0.75	°C
LDPM_TCO_ST_1_LACO	6	0...FEH	-48...142.5	0.75	°C
minimum TCO for lambda controller activation at idle speed					
ID_T_AST_MIN_LSCL_ACT	6	0...FFFFH	0...6.5535E+3	0.1	s
LDPM_TCO_ST_1_LACO	6	0...FEH	-48...142.5	0.75	°C
minimum time after start that must be elapsed before activating lambda controller					
ID_T_MAX_LSCL_ACT_TCO	6	1...FFFFH	1...6.5535E+4	1	s
LDPM_TCO_ST_1_LACO	6	0...FEH	-48...142.5	0.75	°C
maximum time value up to that the controller can be activated by TCO condition					
IP_FAC_LAM_MAX_NOT_STAT_CDN	8	0...FFFFH	-50...49.9984741	0.00152588	%
LDPM_FAC_LAM_MV_1_LACO	8	0...FFFFH	-50...49.998474	0.0015259	%
upper limit of lambda controller output under non stationary operating conditions					
IP_FAC_LAM_MIN_NOT_STAT_CDN	8	0...FFFFH	-50...49.9984741	0.00152588	%
LDPM_FAC_LAM_MV_1_LACO	8	0...FFFFH	-50...49.998474	0.0015259	%
bottom limit of lambda controller output under non stationary operating conditions					
IP_LAMB_SP_THE_PUC	4	0...FFFFH	0...3.99993896	6.10352E-5	-
LDP_MAF_INT_PUC_IP_LAMB_SP_THE	4	0...FFFFH	0...2.91267E+3	0.04444444	g
theoretical lambda set point during PUC					
IP_LAMB_SP_THE_S_HOM	4	0...FFFFH	0...3.99993896	6.10352E-5	-
LDP_MAF_INT_S_IP_LAMB_SP_THE	4	0...FFFFH	0...2.91267E+3	0.04444444	g
Theoretical lambda set point for switching from S to HOM					
IP_FAC_EG_DLY[NC_CBK_EX_NR]	8x8	0...FFH	0...31.875	0.125	-
LDPM_N_1_LACO	8	0...1FE0H	0...8.16E+3	1	rpm
LDPM_MAF_1_LACO	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
weighting factor for the calculation of the exhaust gas delay					
IP_FAC_EG_DLY_FAC_IPT[NC_CBK_EX_NR]	8x8	0...FFH	0...31.875	0.125	-
LDPM_N_1_LACO	8	0...1FE0H	0...8.16E+3	1	rpm
LDPM_MAF_1_LACO	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
weighting factor for the calculation of the exhaust gas delay for interpolation with FAC_IPT_SENS_MDL					
IP_FAC_GAIN_D_LAM	8x8	0...7FFFH	0...0.03999878	1.2207E-6	s
LDPM_T_DLY_SOI_LSL_POS_1_LACO	8	0...FFFFH	0...1.023984	1.5625E-5	s
LDPM_T1_LSL_UP_1_LACO	8	0...FFFFH	0...1.99997	3.0518E-5	s
gain of lambda controller D share					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_GAIN_D_LAM_GAIN_SWI	4x4	0...7FFFH	0...0.03999878	1.2207E-6	s
LDPM_T_DLY_SOI_LSL_POS_2_LACO	4	0...FFFFH	0...1.023984	1.5625E-5	s
LDPM_T1_LSL_UP_2_LACO	4	0...FFFFH	0...1.99997	3.0518E-5	s
gain of lambda controller D share in case of LV_LAM_GAIN_SWI is set					
IP_FAC_GAIN_I2_LAM	8x8	0...FFFFH	0...24.9996185	3.8147E-4	1/s²
LDPM_T_DLY_SOI_LSL_POS_1_LACO	8	0...FFFFH	0...1.023984	1.5625E-5	s
LDPM_T1_LSL_UP_1_LACO	8	0...FFFFH	0...1.99997	3.0518E-5	s
gain of lambda controller I^2 share					
IP_FAC_GAIN_I2_LAM_GAIN_SWI	4x4	0...FFFFH	0...24.9996185	3.8147E-4	1/s²
LDPM_T_DLY_SOI_LSL_POS_2_LACO	4	0...FFFFH	0...1.023984	1.5625E-5	s
LDPM_T1_LSL_UP_2_LACO	4	0...FFFFH	0...1.99997	3.0518E-5	s
gain of lambda controller I^2 share in case of LV_LAM_GAIN_SWI is set					
IP_FAC_GAIN_I_LAM	8x8	D...FFFFH	0.00991821 ... 49.9992371	7.62939E-4	1/s
LDPM_T_DLY_SOI_LSL_POS_1_LACO	8	0...FFFFH	0...1.023984	1.5625E-5	s
LDPM_T1_LSL_UP_1_LACO	8	0...FFFFH	0...1.99997	3.0518E-5	s
gain of lambda controller I share					
IP_FAC_GAIN_I_LAM_GAIN_SWI	4x4	D...FFFFH	0.00991821 ... 49.9992371	7.62939E-4	1/s
LDPM_T_DLY_SOI_LSL_POS_2_LACO	4	0...FFFFH	0...1.023984	1.5625E-5	s
LDPM_T1_LSL_UP_2_LACO	4	0...FFFFH	0...1.99997	3.0518E-5	s
gain of lambda controller I share in case of LV_LAM_GAIN_SWI is set					
IP_FAC_GAIN_P_LAM	8x8	0...FFFFH	0...15.999756	2.44141E-4	-
LDPM_T_DLY_SOI_LSL_POS_1_LACO	8	0...FFFFH	0...1.023984	1.5625E-5	s
LDPM_T1_LSL_UP_1_LACO	8	0...FFFFH	0...1.99997	3.0518E-5	s
gain of lambda controller P share					
IP_FAC_GAIN_P_LAM_GAIN_SWI	4x4	0...FFFFH	0...15.999756	2.44141E-4	-
LDPM_T_DLY_SOI_LSL_POS_2_LACO	4	0...FFFFH	0...1.023984	1.5625E-5	s
LDPM_T1_LSL_UP_2_LACO	4	0...FFFFH	0...1.99997	3.0518E-5	s
gain of lambda controller P share in case of LV_LAM_GAIN_SWI is set					
IP_T1_LSL_UP[NC_CBK_EX_NR]	8x8	0...FFFFH	0...1.99996948	3.05176E-5	s
LDPM_N_1_LACO	8	0...1FE0H	0...8.16E+3	1	rpm
LDPM_MAF_1_LACO	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
first order time lag of the WRAF sensor model					
IP_T1_LSL_UP_FAC_IPT[NC_CBK_EX_NR]	8x8	0...FFFFH	0...1.99996948	3.05176E-5	s
LDPM_N_1_LACO	8	0...1FE0H	0...8.16E+3	1	rpm
LDPM_MAF_1_LACO	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
first order time lag of the WRAF sensor model for interpolation with FAC_IPT_SENS_MDL					

46.5.1 Precontrol path calculation

For fast reaction on lambda set point changes a feed forward control (precontrol path) is included. Therefore in a first step the precontrol lambda set point LAMB_SP_PCTL[i] also used by the controller is determined.



Sub-function in the whole module context

Additional remark:

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

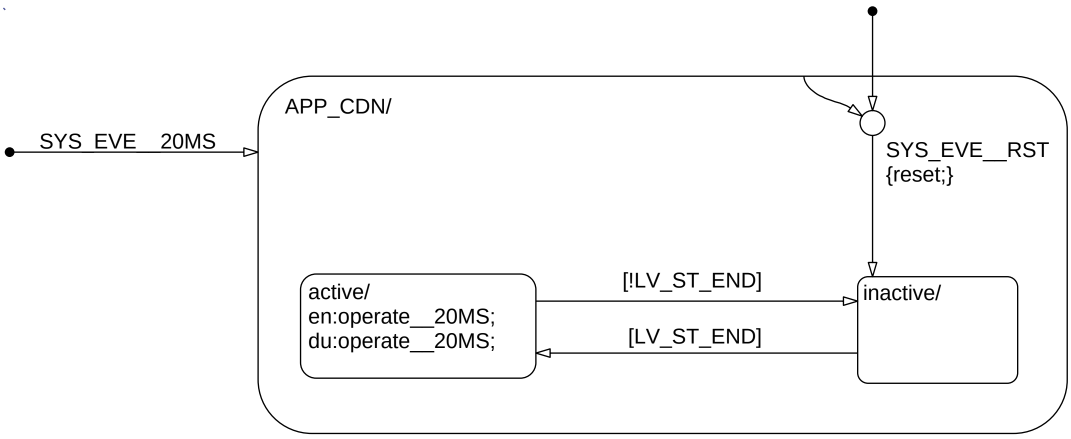


Figure 14 LACO_REQGNLACO0/ CLC_PCTL/ APP_CDN/ Chart

Function Description

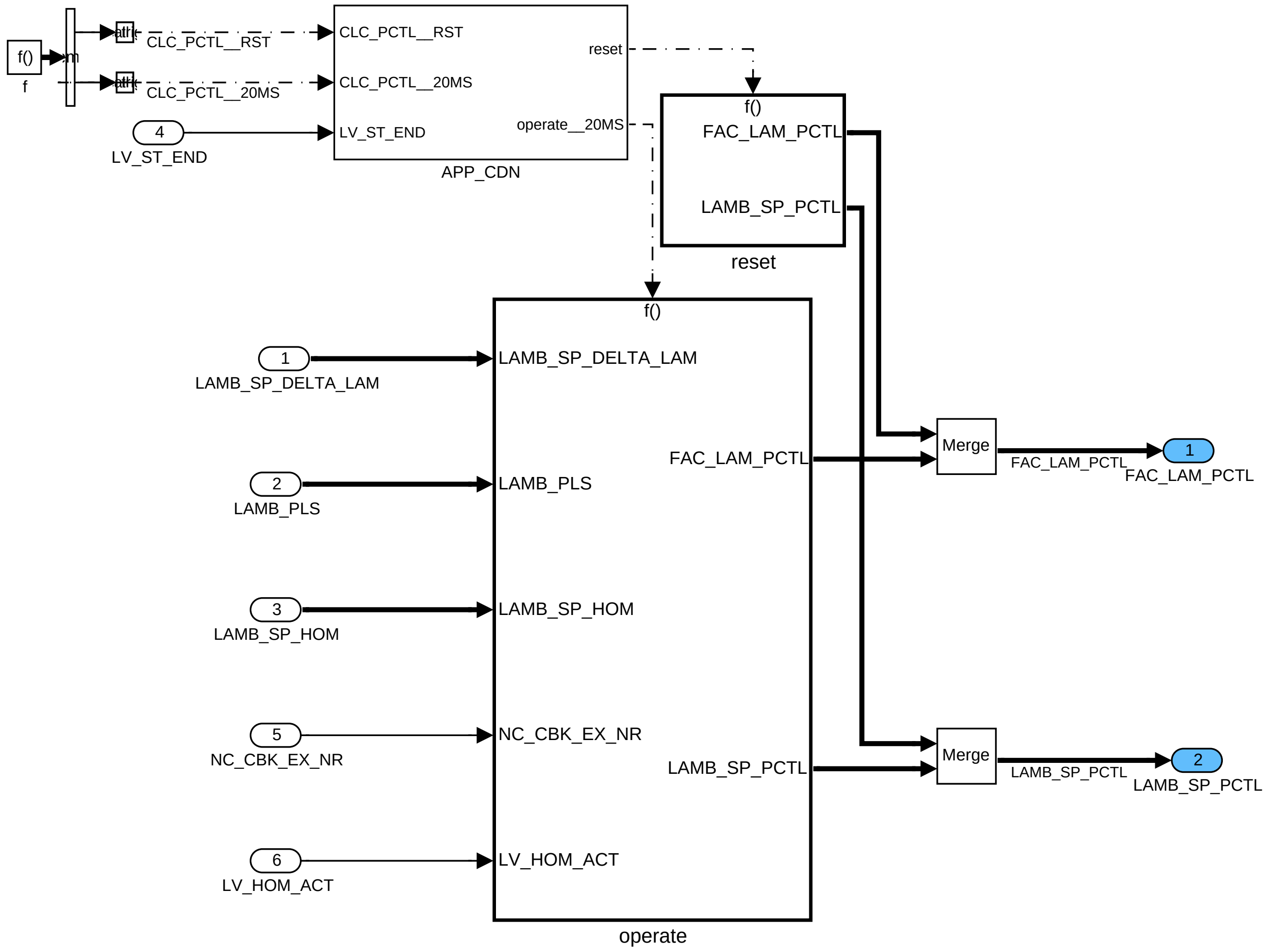


Figure 15 LACO_REQGNLACO0/ CLC_PCTL

46.5.1.1 Initialization

FAC_LAM_PCTL[i] is set to 0.

LAMB_SP_PCTL[i] is set to 1.

46.5.1.2 Formula section

Exhaust bank specific functionality

Calculation of LAMB_SP_PCTL[i] and FAC_LAM_PCTL[i]

In case of homogeneous mode the precontrol lambda set point is the static set point LAMB_SP_HOM[i] added up by the forced stimulation output LAMB_PLS[i]. LAMB_SP_DELTA_LAM[i] is subtracted to consider trim control lambda intervention. This latter variable is assigned in the application incidences.

The precontrol correction FAC_LAM_PCTL[i] is calculated by inverting LAMB_SP_PCTL[i].

In case of stratified mode (LV_HOM_ACT[i] = 0) both outputs are initialized.

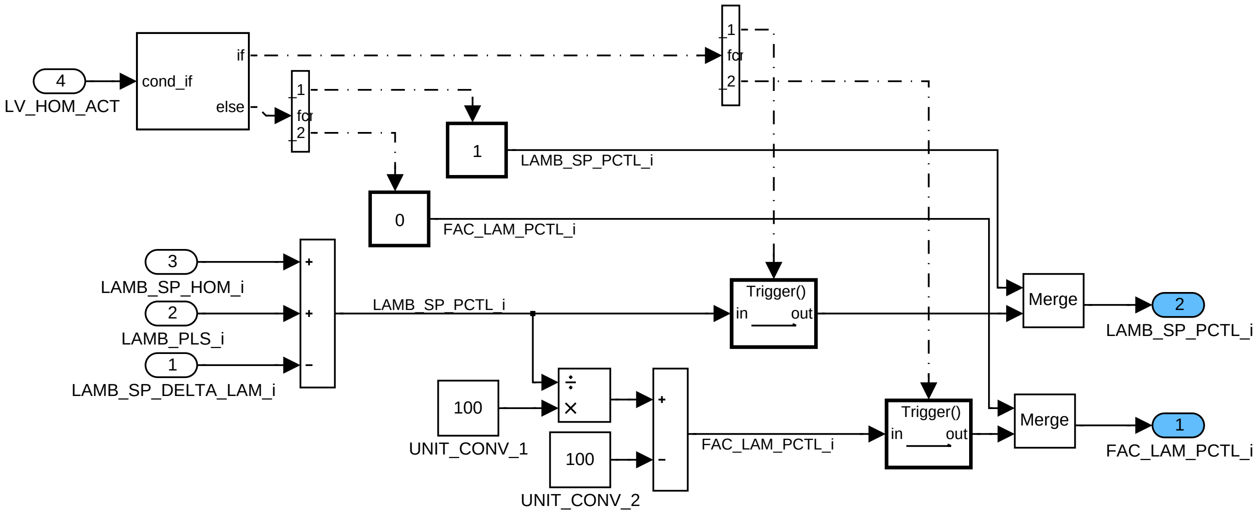
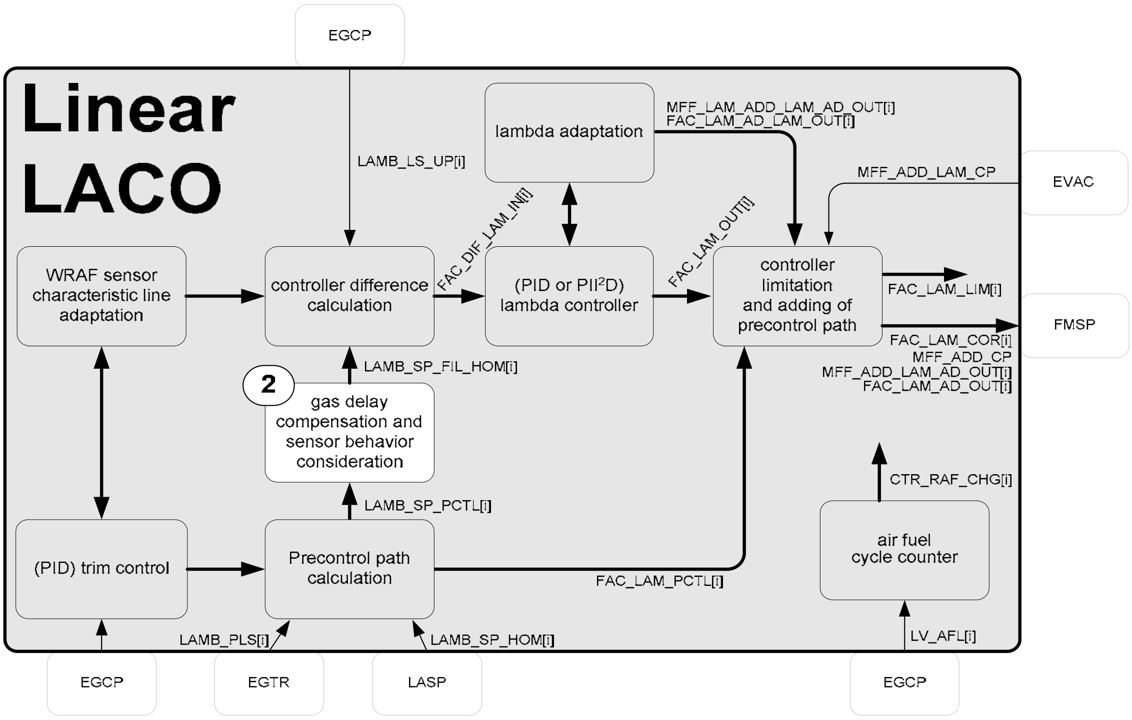


Figure 16 LACO_REQGNLACO0/ CLC_PCTL/ operate/ CBK_EX_MNG/ CBK_EX_SPC

46.5.2 Exhaust gas delay compensation and sensor behavior consideration

The lambda set point for homogeneous mode includes the signal of the forced lambda stimulation, the trim control intervention and eventually an influence of secondary air. In stratified mode the lambda set point is calculated out of the fuel mass flow set point. Both set points are corrected taking into account the exhaust gas delay and the sensor behavior. When changing the injected fuel mass the result of the air fuel ratio due to this change is "seen" by the WRAF sensor some time later. The consideration of this delay and the WRAF sensor rise time is the main content of this function.

The result is a filtered lambda set point for homogeneous mode LAMB_SP_FIL_HOM[i] and for stratified mode LAMB_SP_FIL_S[i]. This latter variable is calculated for other functions like canister purge; it is not used by LACO.



Sub-function in the whole module context

Additional remark:

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

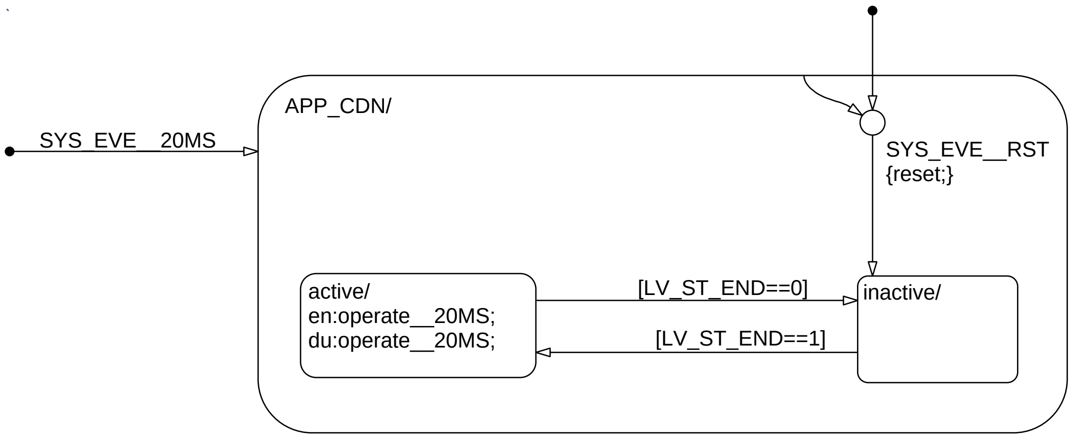


Figure 17 LACO_REQGNLACO0/ SENS_MDL/ APP_CDN/ Chart

Function Description

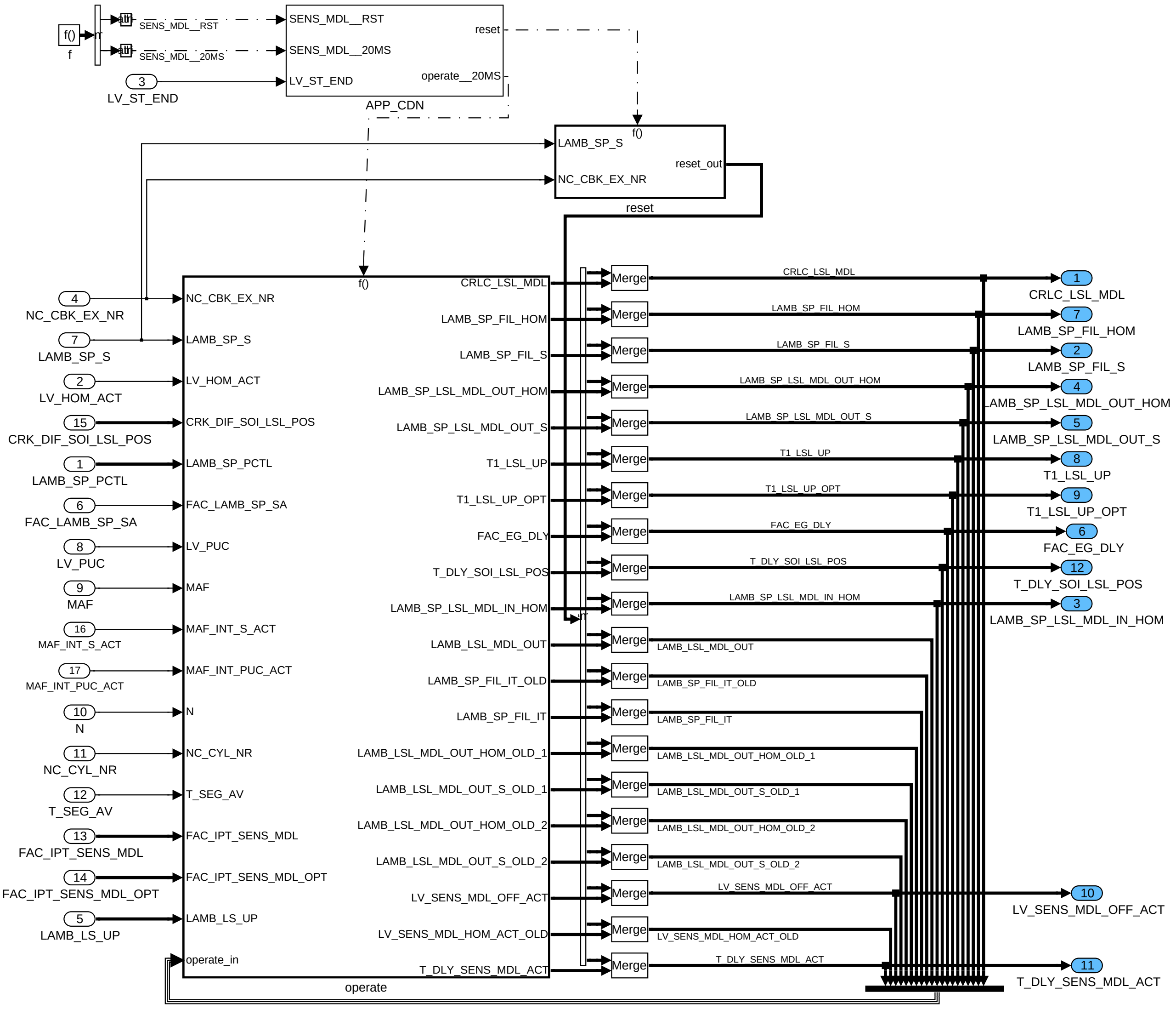


Figure 18 LACO_REQGNLACO0/ SENS_MDL

46.5.2.1 Initialization

All lambda variables are set to 1 except LAMB_SP_FIL_S[i] which is set to LAMB_SP_S.

All other output variables are set to 0.

46.5.2.2 Formula section

Exhaust bank specific functionality

Lambda set point; T1 determination; sensor and exhaust gas delay model

Lambda set point as model input (LAMB_SP_LSL_MDL_IN_HOM[i])

In case of homogeneous mode the input to the model LAMB_LSL_MDL_INP[i] (internal variable) is given by the variable LAMB_SP_LSL_MDL_IN_HOM[i] and in case of stratified mode by LAMB_SP_S.

The lambda set point LAMB_SP_LSL_MDL_IN_HOM[i] in homogeneous mode is being based on the precontrol lambda set point LAMB_SP_PCTL[i] and the secondary air factor

FAC_LAMB_SP_SA. If the secondary air function is not active or no secondary air function is included this factor is set to 1.

When lambda sensor signal LAMB_LS_UP exceeds during PUC the threshold C_LAMB_SENS_MDL_OFF_PUC or when LV_HOM_ACT switches from 0 to 1, the flag LV_SENS_MDL_OFF_ACT[i] is set in order to deactivate and initialize the sensor model.

Based on LV_SENS_MDL_OFF_ACT[i]=1 the 1st order filter and Padé filter are initialized either with IP_LAMB_SP_THE_PUC or IP_LAMB_SP_THE_S_HOM depending on LV_HOM_ACT.

Once set to 1 the flag LV_SENS_MDL_OFF_ACT is set to 0 after a tunable time delay outside PUC. The time delay is given by C_T_DLY_SENS_MDL_ACT[i] and should be >=40ms (= 2 sample steps) in order to initialize the Padé filter well. In order to avoid double initialization when LV_HOM_ACT switches just after exit from PUC the delay time threshold can be used with increased values (e.g. 200ms).

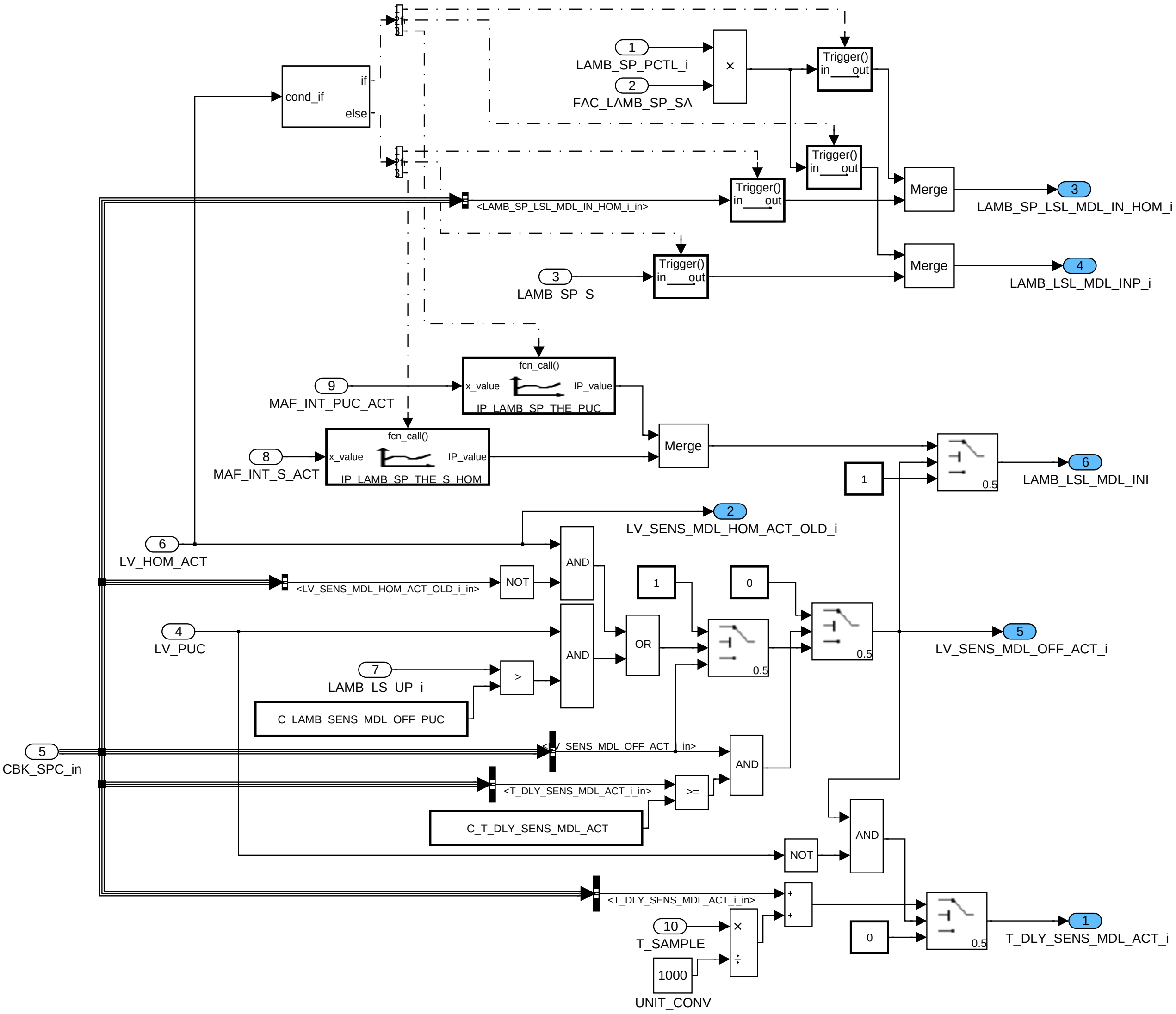


Figure 19 LACO_REQGNLACO0/ SENS_MDL/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ CLC__LSL_MDL_INP

Consideration of dynamic sensor behavior; T1 determination (T1_LSL_UP[i])

The dynamic behavior of the WRAF sensor is approximated by a first order lag system (low pass filter). The time lag (T1_LSL_UP[i]) of the lambda sensor model is calibrated N and MAF dependent. FAC_IPT_SENS_MDL[i] interpolates between the map IP_T1_LSL_UP[i] and the map IP_T1_LSL_UP_FAC_IPT[i] to take into account difference in the sensor behavior.

Furthermore the correlation constant CRLC_LSL_MDL[i] is calculated from T1_LSL_UP[i] taking into account the sample rate of the function.

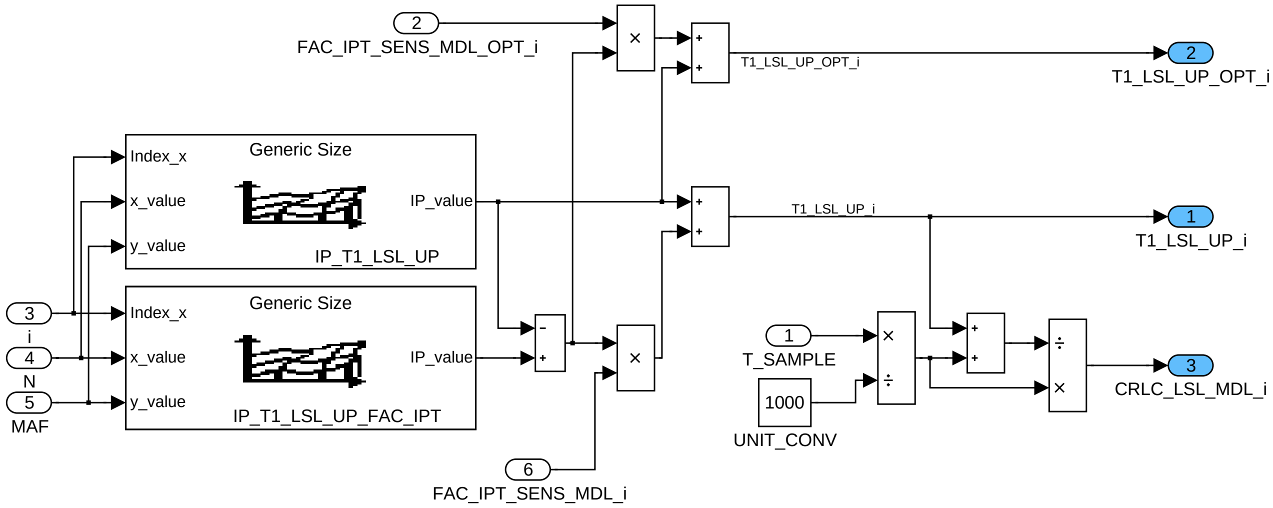


Figure 20 LACO_REQGNLACO0/ SENS_MDL/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ CLC__T1_LSL_UP

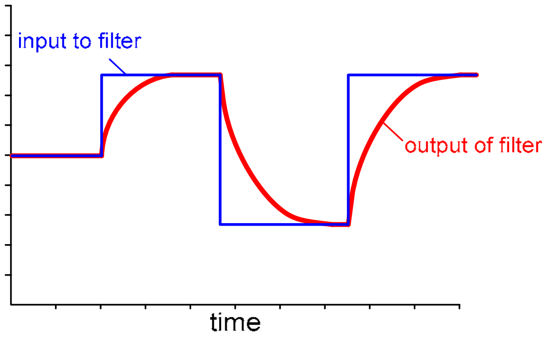
Filtering of lambda set point model input (LAMB_SP_LSL_MDL_OUT_HOM/S[i])

LAMB_LSL_MDL_INP[i] is low pass filtered (moving mean value) by means of the correlation constant CRLC_LSL_MDL[i]. The result is written to the variables LAMB_LSL_MDL_OUT[i] (internal variable), LAMB_SP_LSL_MDL_OUT_S[i] and LAMB_SP_LSL_MDL_OUT_HOM[i] with the corresponding resolution and physical range.

Based on LV_SENS_MDL_OFF_ACT[i] = 1 the 1st order filter is initialized with LAMB_LSL_MDL_INI.

The output LAMB_LSL_MDL_OUT[i] is copied to both variables LAMB_SP_LSL_MDL_OUT_HOM[i] and LAMB_SP_LSL_MDL_OUT_S[i]. If the range of LAMB_SP_LSL_MDL_OUT_HOM[i] is exceeded then the output is limited to the upper range. The defined range is sufficient for homogeneous mode. LAMB_SP_LSL_MDL_OUT_S[i] always covers the whole range but with a lower resolution. Both variables are calculated in stratified as well as in homogeneous mode.

The principle is shown here below.



Sensor behavior consideration (1^{st} order time lag)

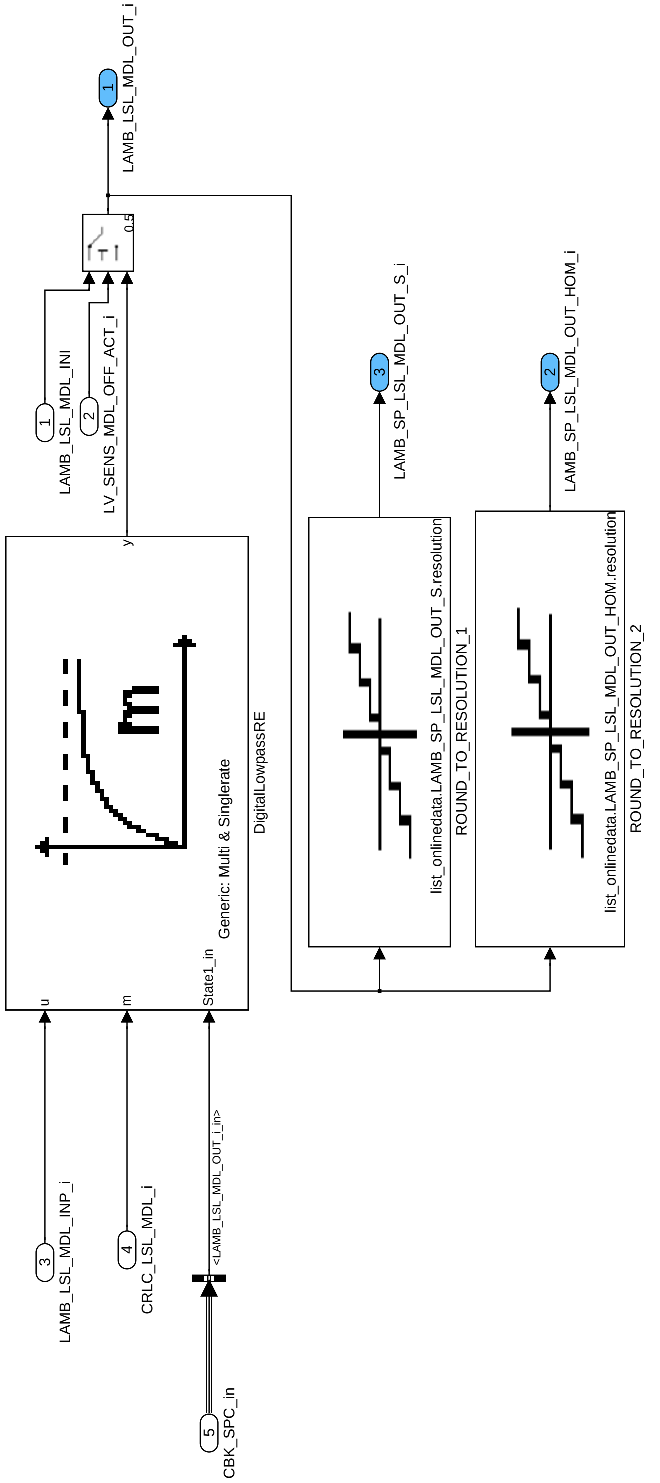


Figure 21 LACO_REQGNLACO0/ SENS_MDL/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ FIL__LAMB_LSL_MDL_INP

Calculation of gas delay time (T_DLY_SOI_LSL_POS[i])

The gas delay T_DLY_SOI_LSL_POS[i] corresponds to the time from fuel injection up to the measurement of the air fuel ratio by the WRAF sensor. IP_FAC_EG_DLY[i] is a tuning factor for the sensor position and is determined together with the T1 maps by means of a system identification. FAC_IPT_SENS_MDL[i] interpolates between the map IP_FAC_EG_DLY[i] and the map IP_FAC_EG_DLY_FAC_IPT[i] to take into account difference in the gas delay time.

The gas delay is calculated with the following formula:

$$\text{T\_DLY\_SOI\_LSL\_POS[i]} = \left[\frac{\text{CRK\_DIF\_SOI\_LSL\_POS[i]}}{180°\text{CRK}} \bullet \frac{\text{NC\_CYL\_NR}}{4} + \text{FAC\_EG\_DLY[i]}\right] \bullet \text{T\_SEG\_AV}.$$

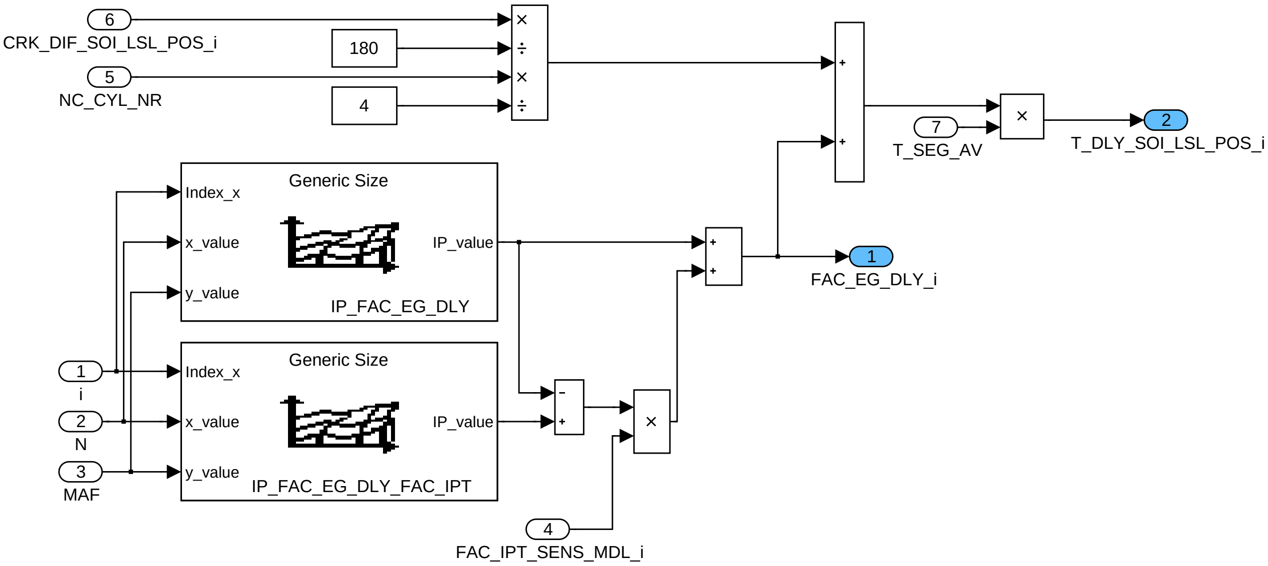
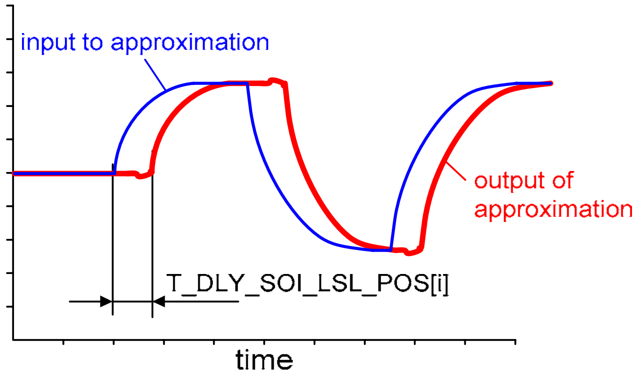


Figure 22 LACO_REQGNLACO0/ SENS_MDL/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ CLC__T_DLY_SOI_LSL_POS

Consideration of the gas delay

By means of a Padé approximation the delayed lambda set point LAMB_SP_FIL_HOM[i] (for homogeneous mode) and LAMB_SP_FIL_S[i] (for stratified mode) is calculated. All other variables which are output of these sub-systems are only necessary to realize the functionality with the desired accuracy.

The behavior of the delay is described by an all-pass filter (Padé approximation of 2nd order). Its principle is shown here below.



Gas delay approximation (2nd order Padé approximation)

The following formula describes the Padé approximation in its mathematical representation:

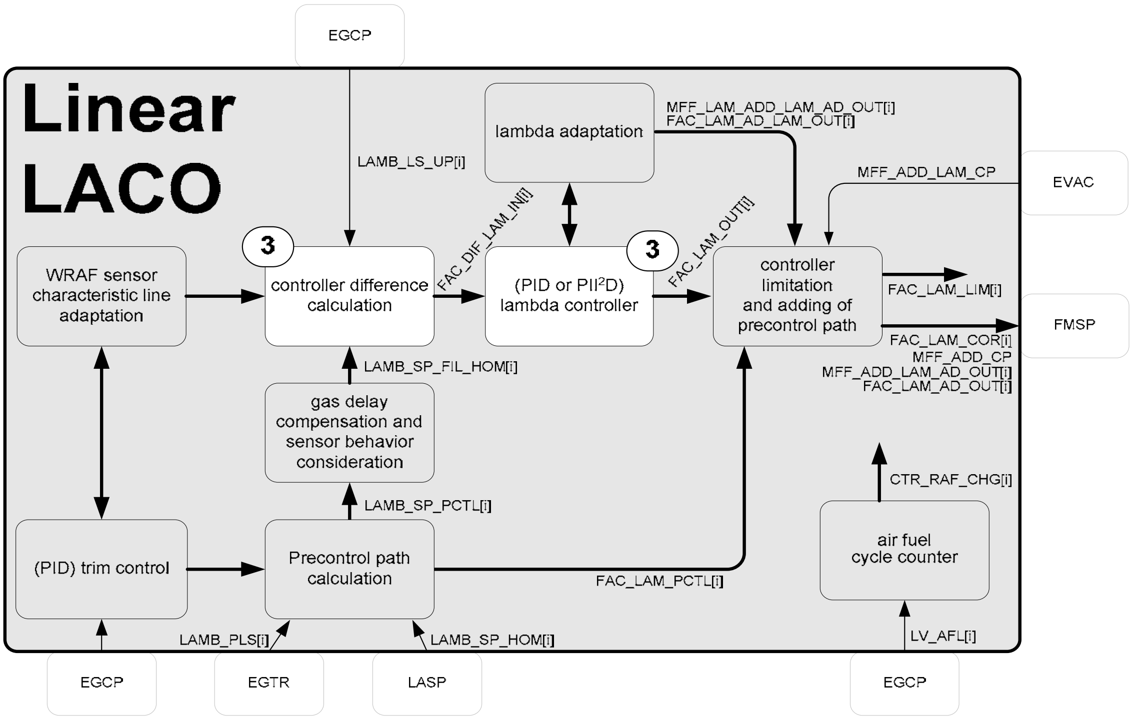
$$f_1 = \frac{\text{T\_DLY\_SOI\_LSL\_POS[i]}}{\text{T\_SAMPLE}},$$

$$f_2 = \frac{f_1^2}{3},$$

$$out_N = \frac{\left(1 - f_1 + f_2\right) \bullet \left(in_N - out_{N-2}\right) + 2 \bullet \left(1 - f_2\right) \bullet \left(in_{N-1} - out_{N-1}\right)}{1 + f_1 + f_2} + in_{N-2}.$$

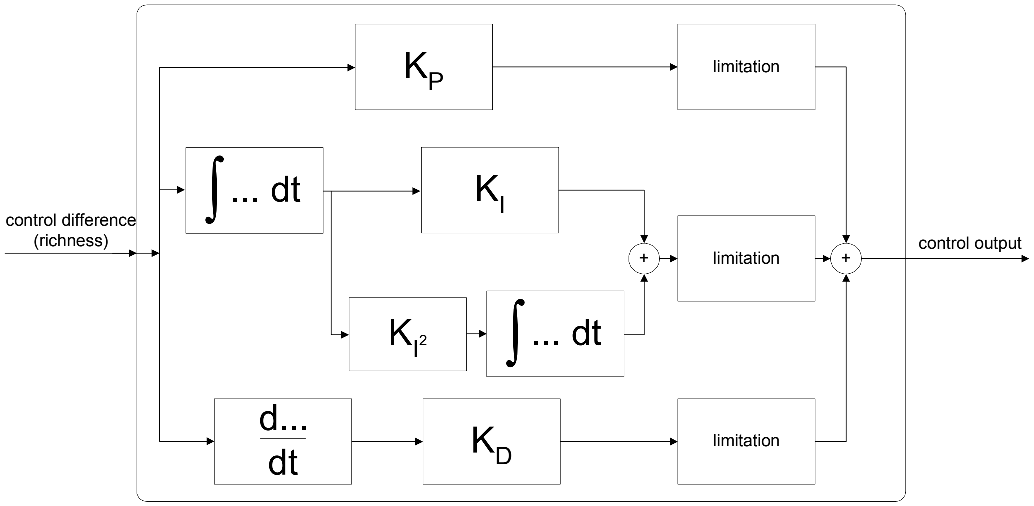
46.5.3 Linear lambda controller

The linear lambda controller can be used as PII^2D or PID controller. By means of the calibration flag LC_LAM_I2_ACT can be switched between both.

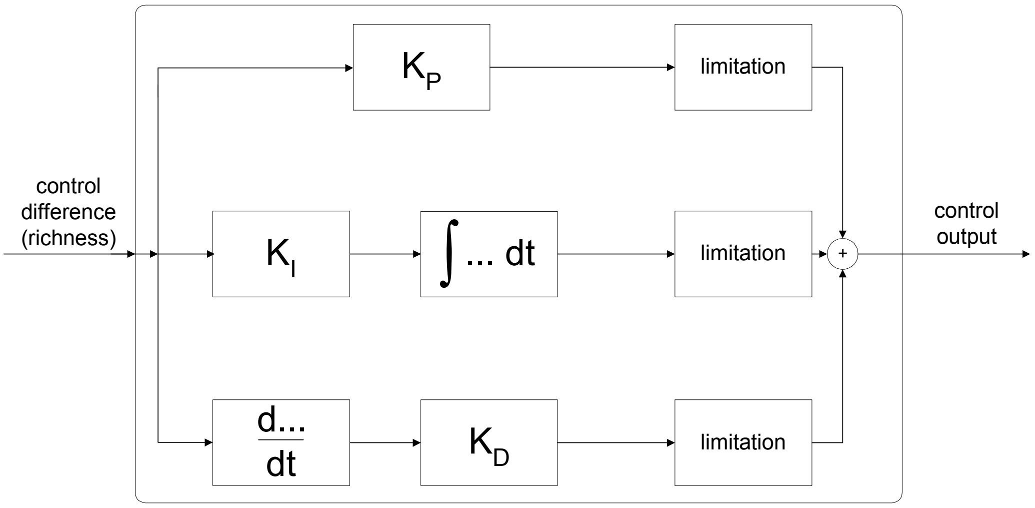


Sub-function in the whole module context

The structure of both controller (PII^2D and PID) is shown hereafter:



Structure of PII^2D controller



Structure of PID controller

Additional remark:

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

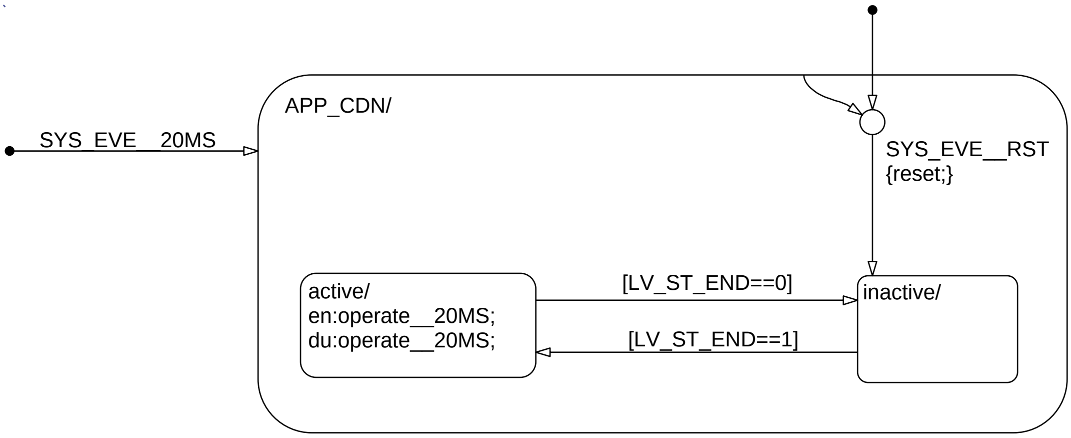


Figure 23 LACO_REQGNLACO0/ LAM/ APP_CDN/ Chart

Function Description

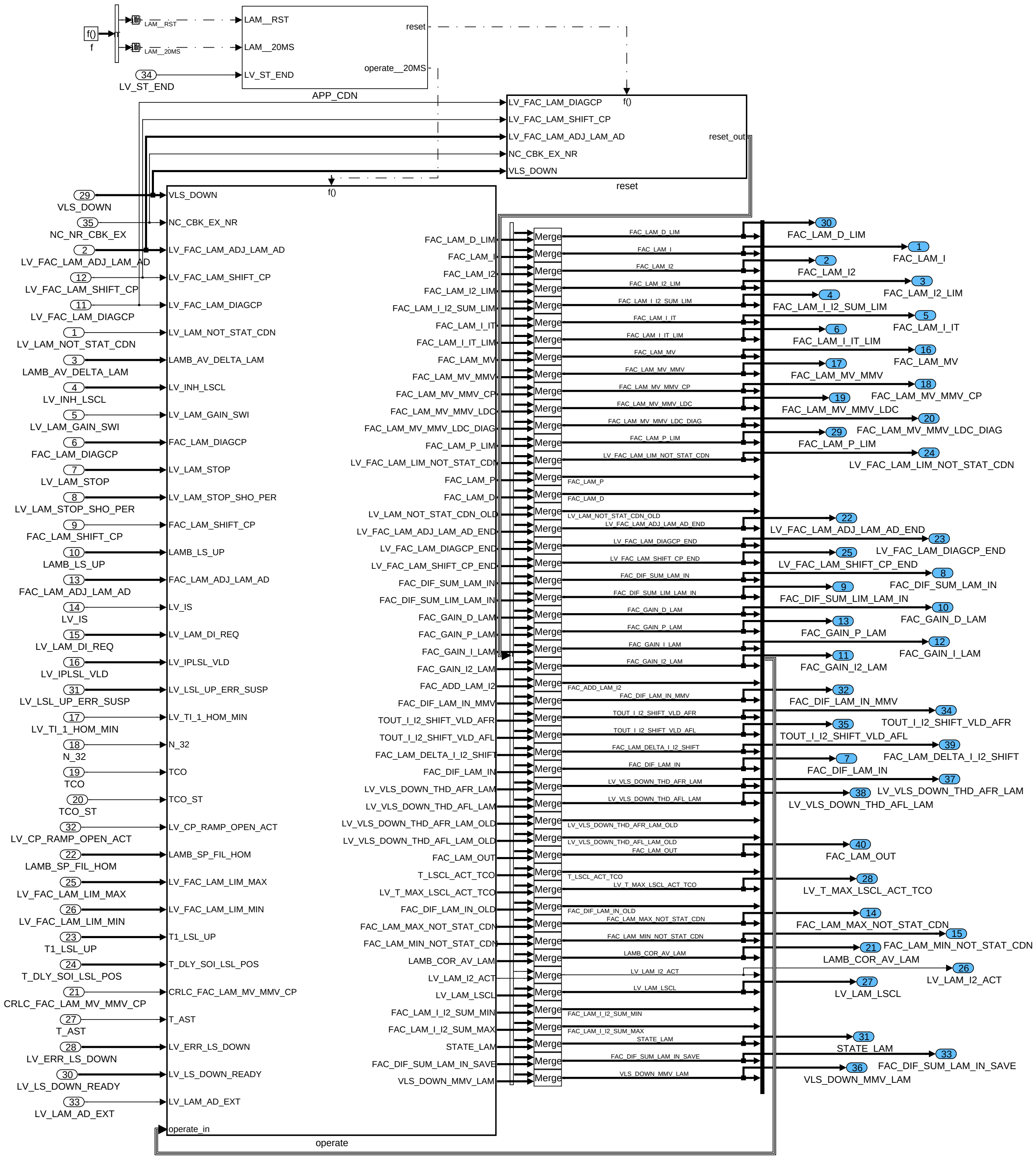


Figure 24 LACO_REQGNLACO0/ LAM

46.5.3.1 Initialization

LAMB_COR_AV_LAM[i] is set to 1.

FAC_LAM_MAX_NOT_STAT_CDN[i] and FAC_LAM_I_I2_SUM_MAX[i] are both set to their physical maximum.

FAC_LAM_MIN_NOT_STAT_CDN[i] and FAC_LAM_I_I2_SUM_MIN[i] are both set to their physical minimum.

LV_LAM_I2_ACT is set to LC_LAM_I2_ACT.

VLS_DOWN_MMV_LAM[i] is set to VLS_DOWN[i].

LV_FAC_LAM_ADJ_LAM_AD_END[i] is set to LV_FAC_LAM_ADJ_LAM_AD[i].

LV_FAC_LAM_SHIFT_CP_END[i] is set to LV_FAC_LAM_SHIFT_CP.

LV_FAC_LAM_DIAGCP_END[i] is set to LV_FAC_LAM_DIAGCP.

All other output variables are set to 0.

46.5.3.2 Formula section

Exhaust bank specific functionality

State machine of lambda controller (STATE_LAM[i])

The state machine of lambda controller distinguishes between 4 different states:

0:OFF

1:ON

2:STOP_SHO_PER

3:STOP

If LV_LAM_LSCL[i] is 0 then state "0:OFF" is applied. Otherwise three different states are to distinguish.

State "1:ON" is applied when the controller is not stopped by the flag LV_LAM_STOP[i]. In that case the controller is in normal operation mode.

If LV_LAM_STOP[i] is set to 1 either state "2:STOP_SHO_PER" or state "3:STOP" is applied depending on LV_LAM_STOP_SHO_PER[i].

If a transition to state "0:OFF" from any other state occurs the variable FAC_LAM_OUT[i] is set to FAC_LAM_I_I2_SUM_LIM[i]. Otherwise FAC_LAM_OUT[i] remains unchanged.

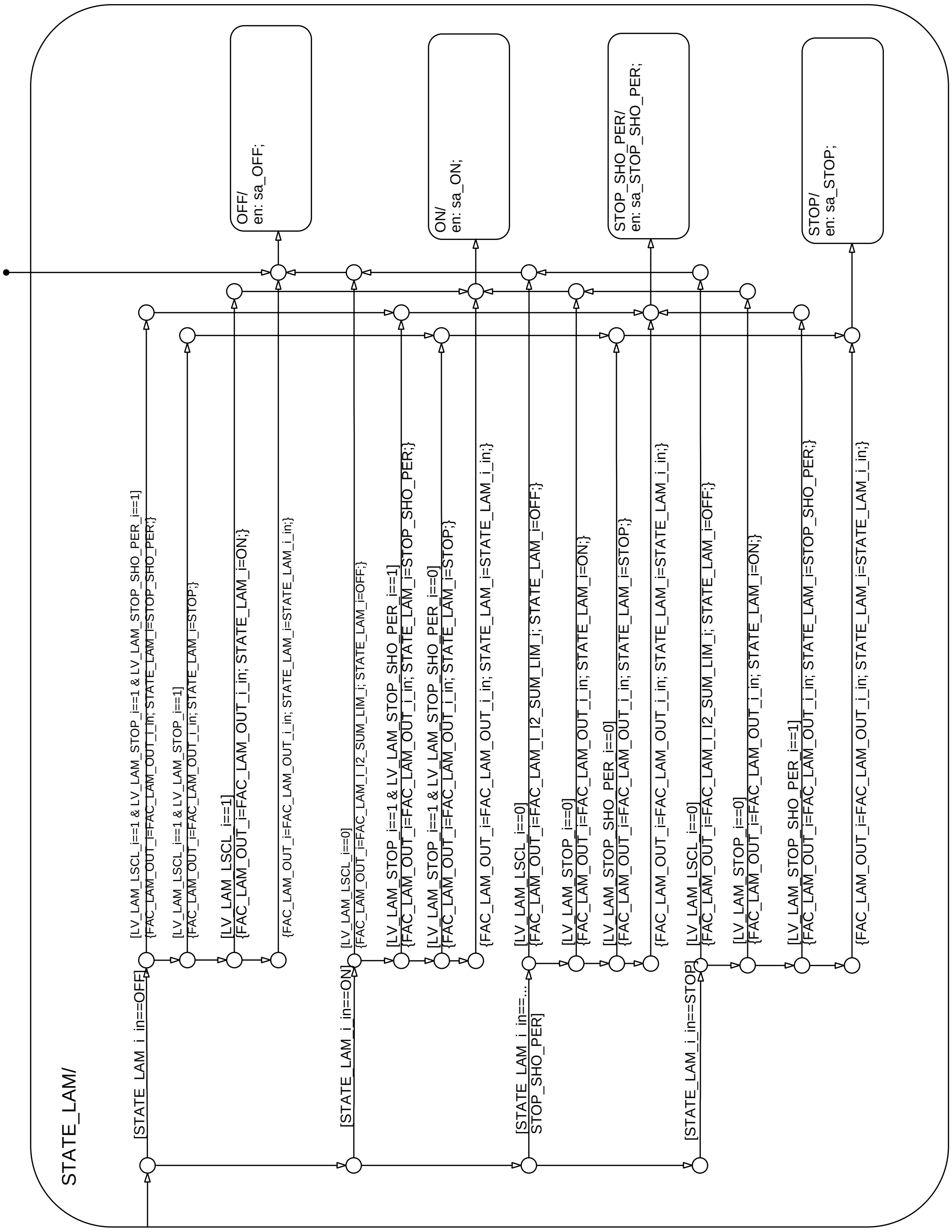


Figure 25 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM

Controller activation / deactivation (LV_LAM_LSCL[i])

Independent of the lambda controller state (STATE_LAM[i]) the evaluation of controller activation / deactivation is calculated.

The controller is activated by the flag LV_LAM_LSCL[i] = 1. This flag is set if the following conditions are fulfilled:

- inhibit flag not set (LV_INH_LSCL[i] = 0)
- no request from OBD function to inhibit lambda controller (LV_LAM_DI_REQ[i] = 0)

- WRAF sensor operability recognised (LV_IPLSL_VLD[i] = 1). In case the operability is removed due to an suspicion of an oxygen sensor error, the Lambda controller shall remain closed loop until the error is confirmed.

- engine speed above threshold (N_32 > C_N_MIN_LAM_LSCL)

- time after start threshold expired (T_AST > ID_T_AST_MIN_LSCL_ACT)

- coolant temperature exceeds threshold
  (in idle speed TCO > ID_TCO_MIN_LAM_LSCL_IS
  otherwise TCO > ID_TCO_MIN_LAM_LSCL)
  or the time ID_T_MAX_LSCL_ACT_TCO is expired

LV_T_MAX_LSCL_ACT_TCO[i] is permanently set to 1 if the controller was activated by the time condition instead of activation by the corresponding TCO condition. Otherwise LV_T_MAX_LSCL_ACT_TCO[i] is set to 0.

If one of the conditions above is not fulfilled the flag LV_LAM_LSCL[i] is set to 0 and the controller is not activated.

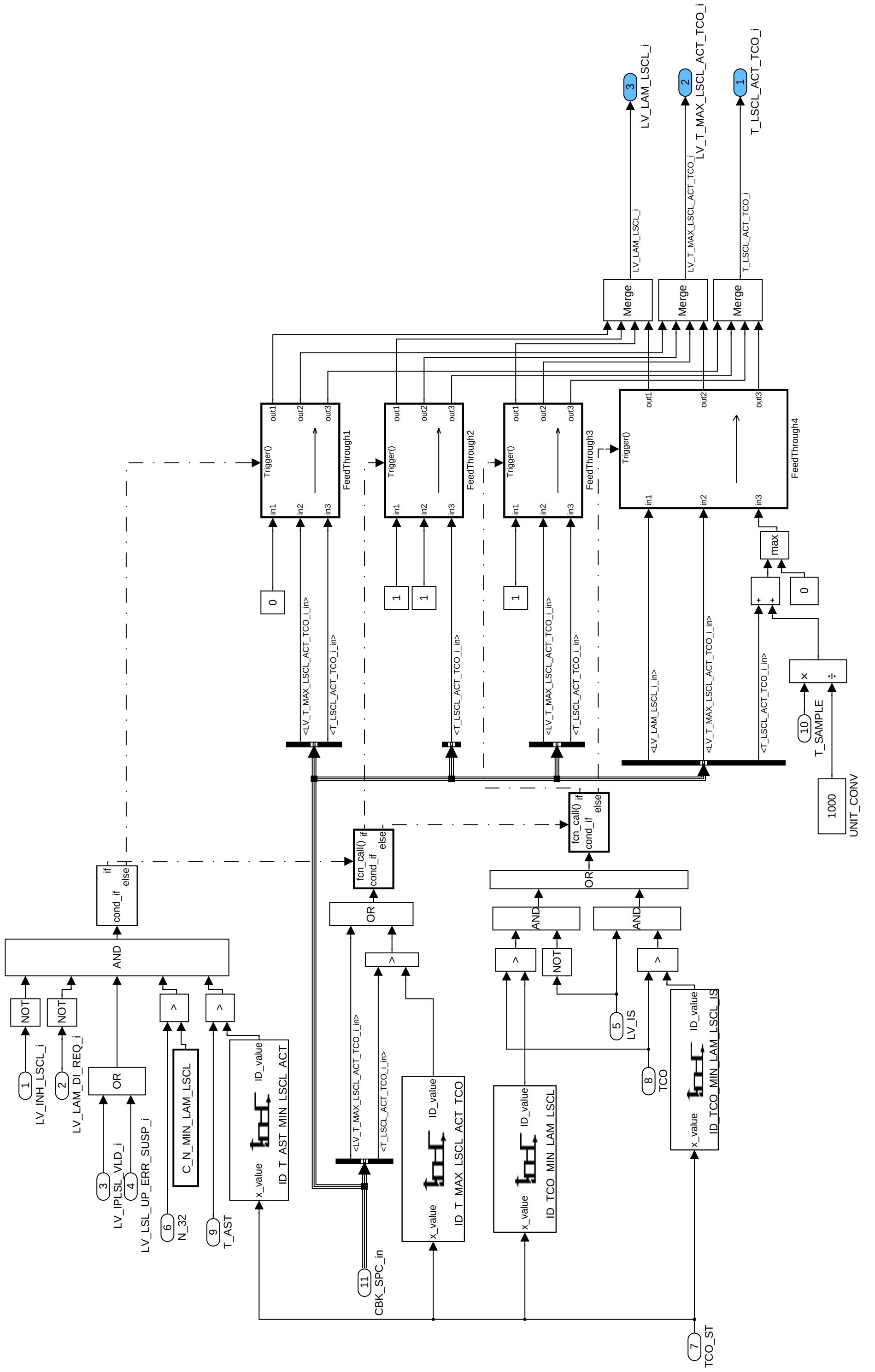


Figure 26 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ ACT_LAM

STATE_LAM: “0:OFF”

FAC_LAM_OUT[i] is set to the current offset (I + I^2 share) when entering this state and then ramped down to zero (for realization see state machine).

Reset of all internal state variables and output values

LV_FAC_LAM_ADJ_LAM_AD_END[i] is set to LV_FAC_LAM_ADJ_LAM_AD[i].

LV_FAC_LAM_SHIFT_CP_END[i] is set to LV_FAC_LAM_SHIFT_CP.

LV_FAC_LAM_DIAGCP_END[i] is set to LV_FAC_LAM_DIAGCP.

Most of all other internal state variables and output values are set to 0 except the ones which are output of the sub-systems described hereafter.

Ramping down of lambda controller output signal (FAC_LAM_OUT[i])

Depending on the sign of the lambda controller output signal FAC_LAM_OUT[i] this latter variable is increased or decreased by C_FAC_LAM_DELTA_LSCL_OFF in order to reach zero (no controller intervention).

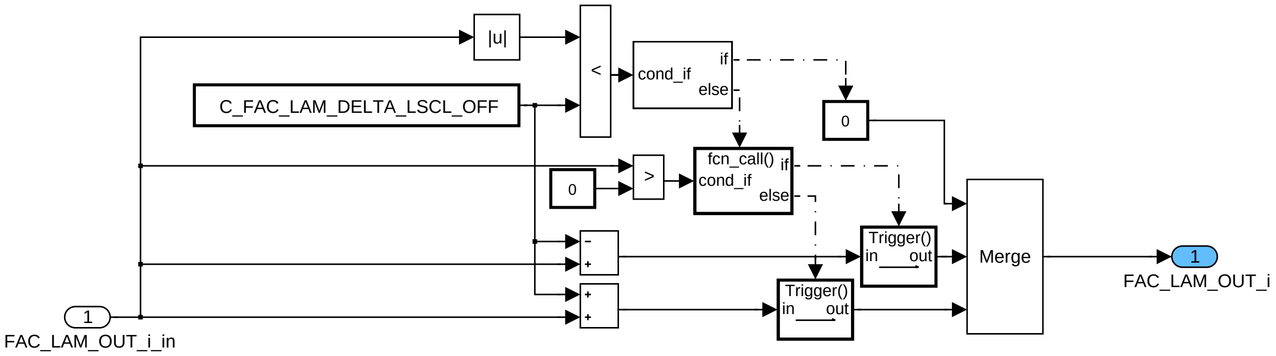


Figure 27 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__OFF/ RAMP_DOWN__FAC_LAM_OUT

Switch between PID or PII2D controller (LV_LAM_I2_ACT)

Depending on LC_LAM_I2_ACT it can be chosen whether I2 share calculation is active (LC_LAM_I2_ACT = 1 -> PII^2D) or not (LC_LAM_I2_ACT = 0 -> PID). This calibration flag is only taken into account in state "OFF".

Figure 28 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__OFF/ SET__LV_LAM_I2_ACT

Calculation of controller difference as richness (FAC_DIF_LAM_IN[i])

The controller difference FAC_DIF_LAM_IN[i] is calculated as difference of the inverted lambda set point

1 / LAMB_SP_FIL_HOM[i]

and the inverted measured lambda signal corrected by the trim control influence

1 / (LAMB_LS_UP[i] + LAMB_AV_DELTA_LAM[i]).

Therefore FAC_DIF_LAM_IN[i] represents a richness. LAMB_AV_DELTA_LAM[i] is assigned in the application incidences.

A moving mean value of the controller difference FAC_DIF_LAM_IN_MMV[i] is calculated in order to realise a shift of I and I^2 share based on downstream signal evaluation. This is described later on.

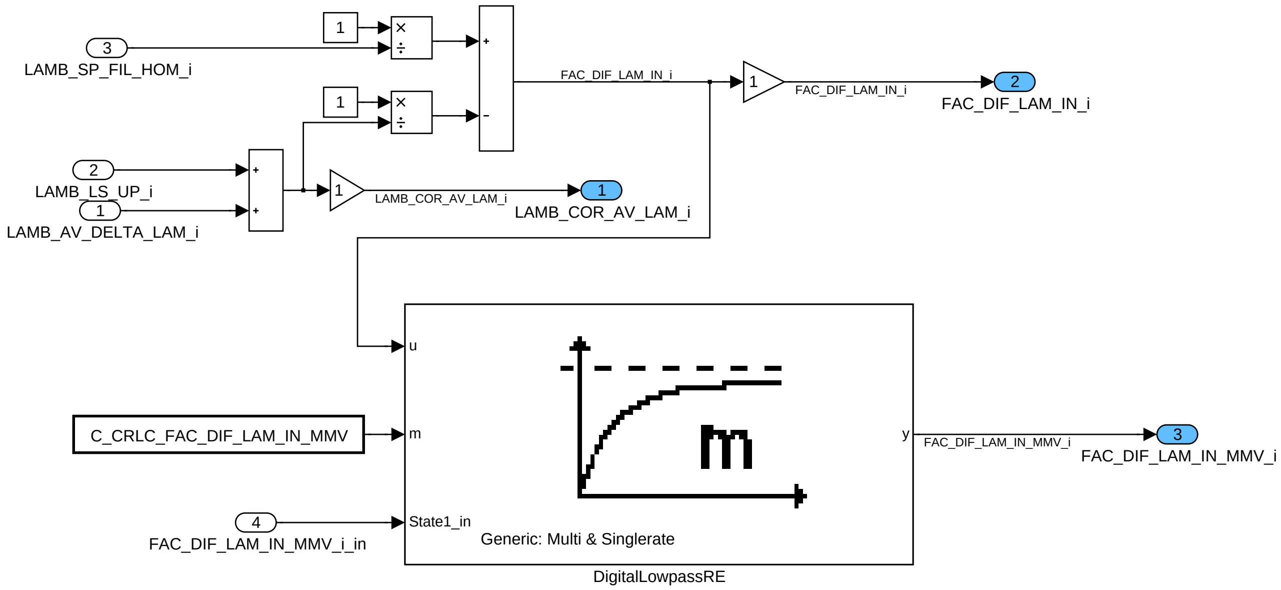


Figure 29 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__OFF/ CLC__FAC_DIF_LAM

Evaluation of downstream sensor voltage (LV_VLS_DOWN_THD_AFL/AFR_LAM[i])

A moving mean value of the downstream sensor signal voltage VLS_DOWN_LAM_MMV[i] is calculated. Based on this moving mean value a lean mixture (100% O_2 loading of the catalyst) or rich mixture (0% O_2 loading of the catalyst) is detected.

The respective flags LV_VLS_DOWN_THD_AFL_LAM[i] or LV_VLS_DOWN_THD_AFR_LAM[i] are set to 1.

The "OLD" flags LV_VLS_DOWN_THD_AFL_LAM_OLD[i] and LV_VLS_DOWN_THD_AFR_LAM_OLD[i] are representing the information of the before mentioned flags one sample step afore. In case of downstream sensor error LV_ERR_DOWN[i] or when the sensor is not operable (LV_LS_DOWN_READY[i] = 0) the "OLD" flags are set to the information of the flags of the current sample step.

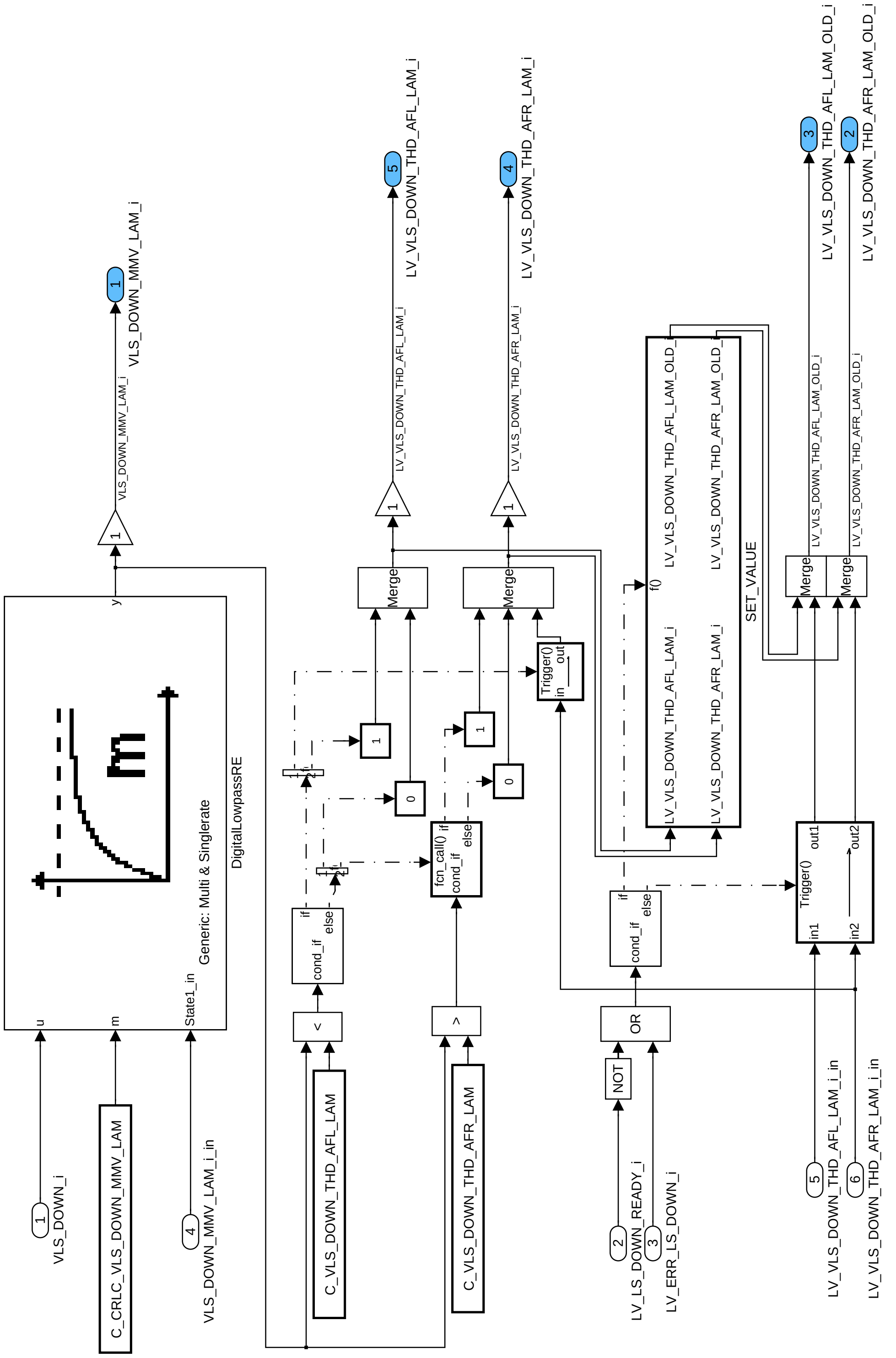


Figure 30 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__OFF/ DET__VLS_DOWN_AFL_AFR

STATE_LAM: "1:ON"

This is the normal operating mode of the lambda controller.

Calculation of controller difference as richness (FAC_DIF_LAM_IN[i])

For details see state "0:OFF".

Evaluation of downstream sensor voltage (LV_VLS_DOWN_THD_AFL/AFR_LAM[i])

For details see state "0:OFF".

Shift and initialization of lambda controller

Output shift of the lambda controller by lambda adaptation

A step change in the output signal of the lambda adaptation requires an inverse step change in the lambda controller output. The control of the output shift takes place in the lambda adaptation function and is indicated by the flags LV_FAC_LAM_ADJ_LAM_AD[i]. The shift of the controller output effects additionally a change of the initial values of the I and I^2 share and in the moving mean values of the lambda controller output.

In case the flag LV_LAM_AD_EXT is set this shift is suppressed.

The lambda controller is shifted if the condition LV_FAC_LAM_ADJ_LAM_AD[i] = 1 is valid. This variable is to be reset if the lambda controller shift was carried out. That requires a confirmation of the executed controller shift. The confirmation is done by setting the variable LV_FAC_LAM_ADJ_LAM_AD_END[i] to 1.

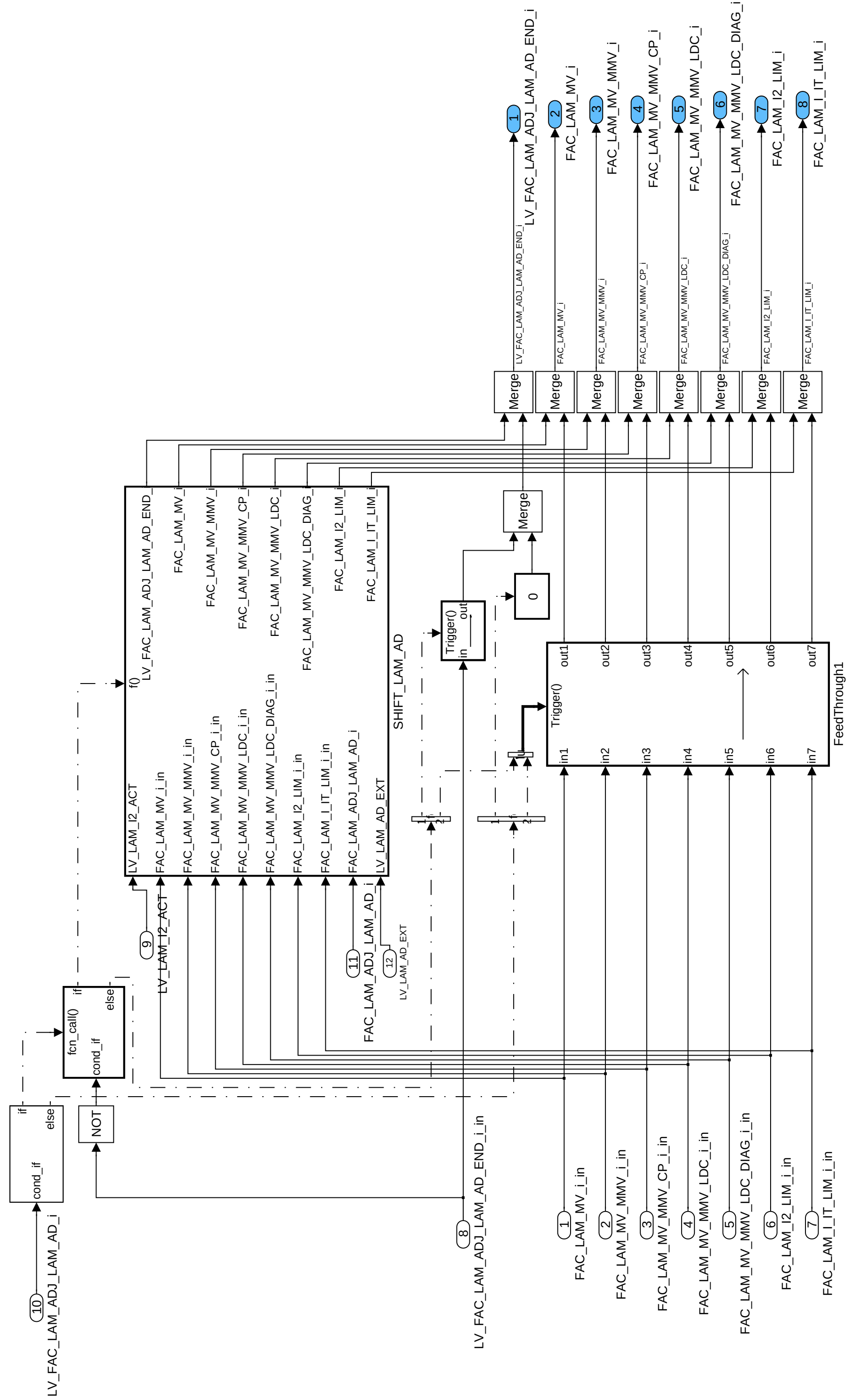


Figure 31 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__FAC_LAM__SHIFT/ MNG__SHIFT_LAM_AD

Execution of shift (SHIFT_LAM_AD)

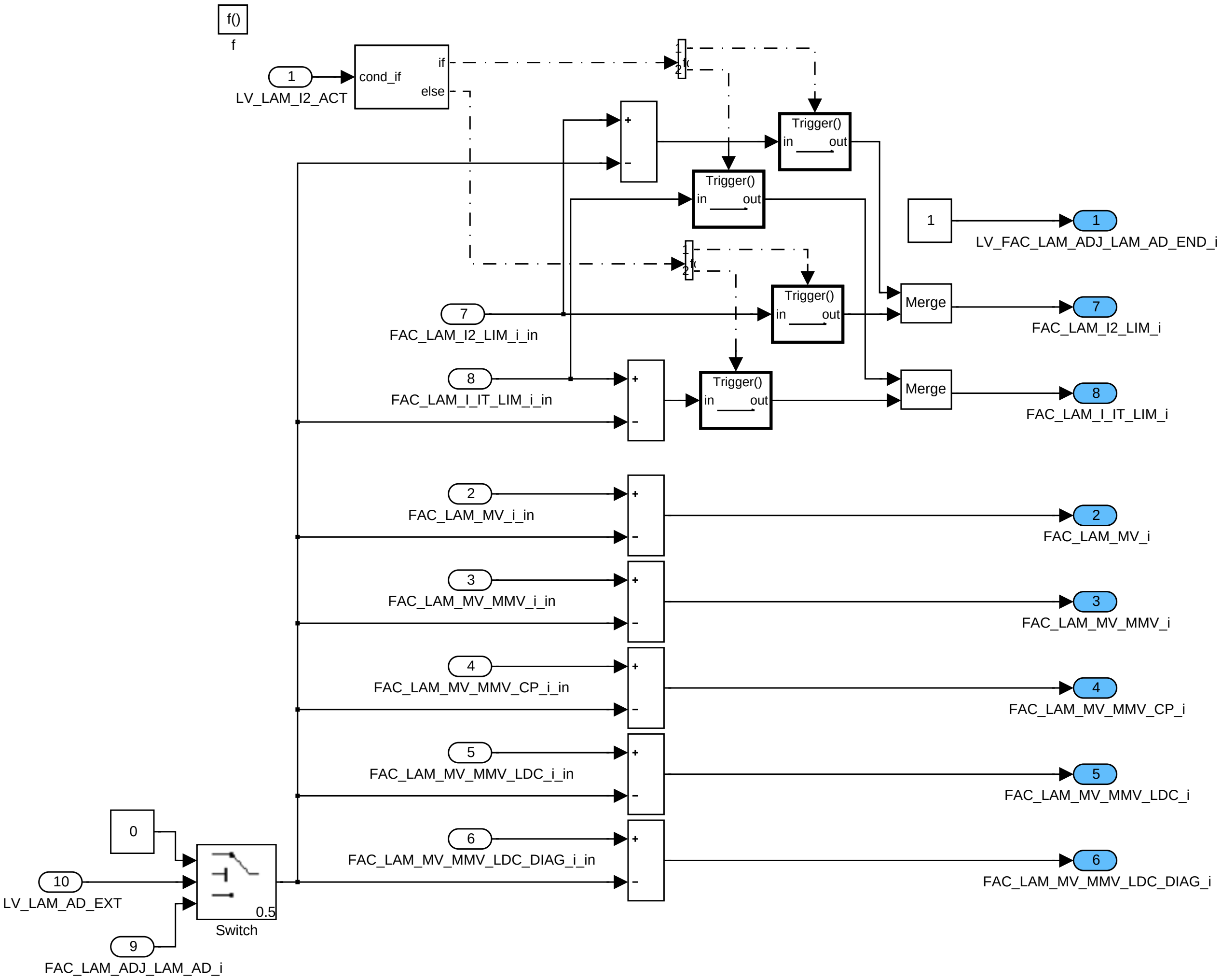


Figure 32 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__FAC_LAM__SHIFT/ MNG__SHIFT_LAM_AD/ SHIFT_LAM_AD

Output shift of lambda controller by canister purge function

A step change in the output signal of the canister purge function requires an stepped change in the lambda controller output. The control of the output shift takes place in the corresponding function and is indicated by the flag LV_FAC_LAM_SHIFT_CP. The shift of the controller output effects additionally a change of the initial values of the I and I^2 share and in the moving mean values of the lambda controller output.

If LV_FAC_LAM_SHIFT_CP is set to 1 the output shift must be carried out for both banks. This variable is to be reset if the lambda controller shift was carried out. That requires a confirmation of the executed controller shift. The confirmation is done by setting the variable LV_FAC_LAM_SHIFT_CP_END[i] to 1.

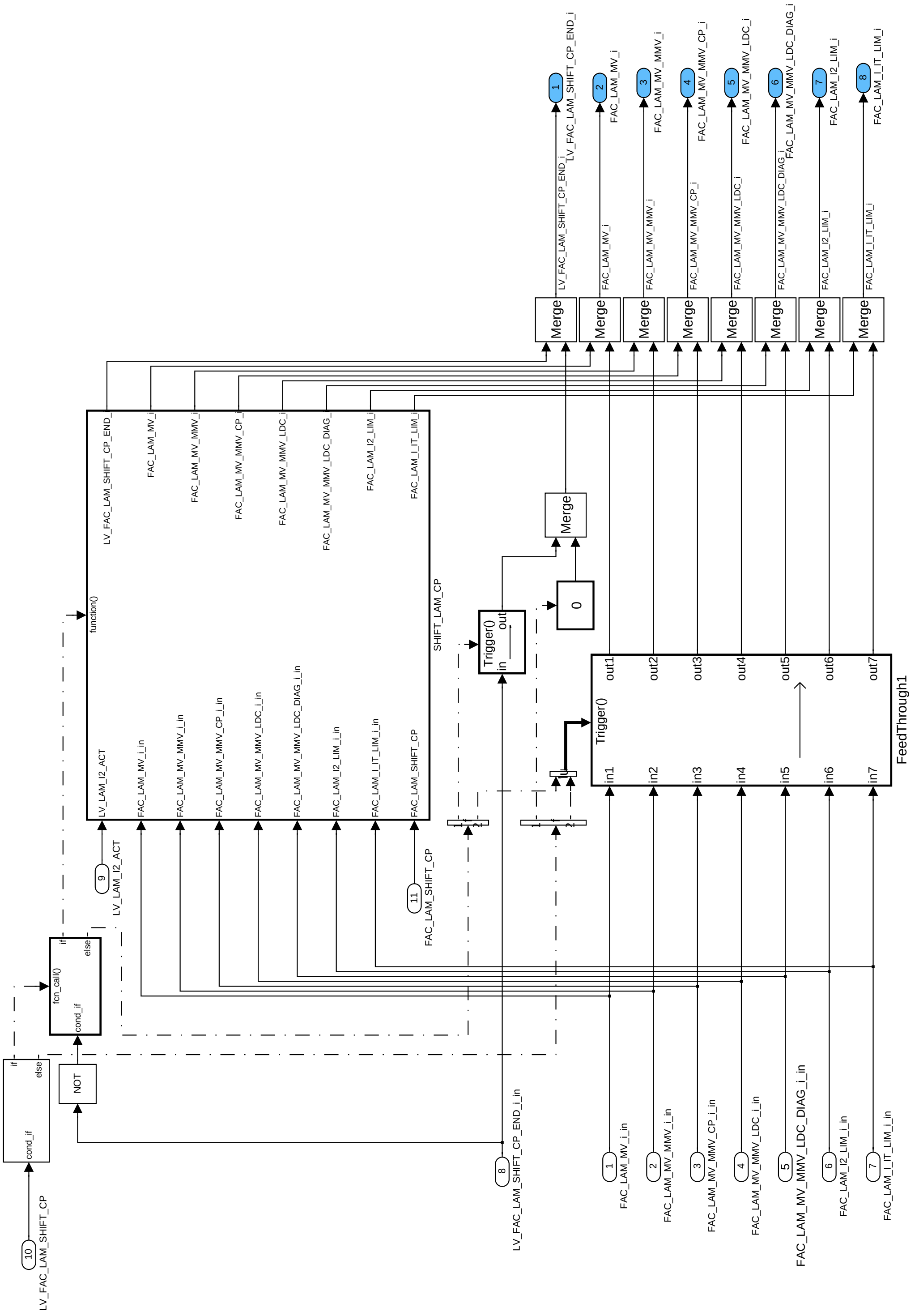


Figure 33 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__FAC_LAM__SHIFT/ MNG__SHIFT_LAM_CP

Execution of shift (SHIFT_LAM_CP)

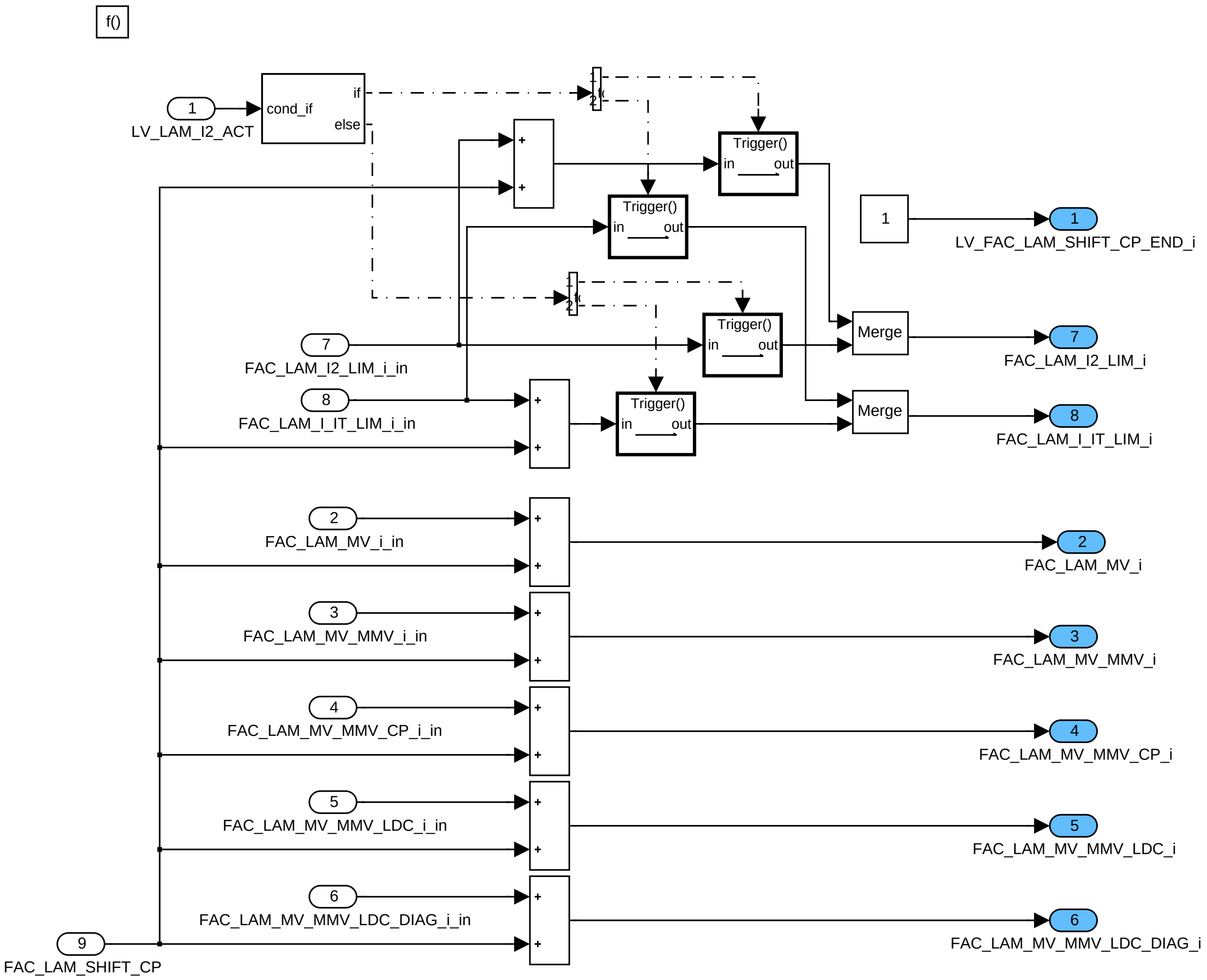


Figure 34 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__FAC_LAM__SHIFT/ MNG__SHIFT_LAM_CP/ SHIFT_LAM_CP

Output initialization of lambda controller by evaporative system monitoring function

The EVAP system monitoring function can require an initialization of the lambda controller output. The control of the output initialization takes place in the corresponding function and is indicated by the flag LV_FAC_LAM_DIAGCP. The initialization of the controller output effects additionally a change of the initial values of I and I^2 share and in the moving mean values of the lambda controller output.

If LV_FAC_LAM_DIAGCP is set to 1 the output initialization is carried out for both banks. This variable is to be reset if the lambda controller initialization was carried out. That requires a confirmation of the executed initialization. The confirmation is done by setting the variable LV_FAC_LAM_DIAGCP_END[i] to 1.

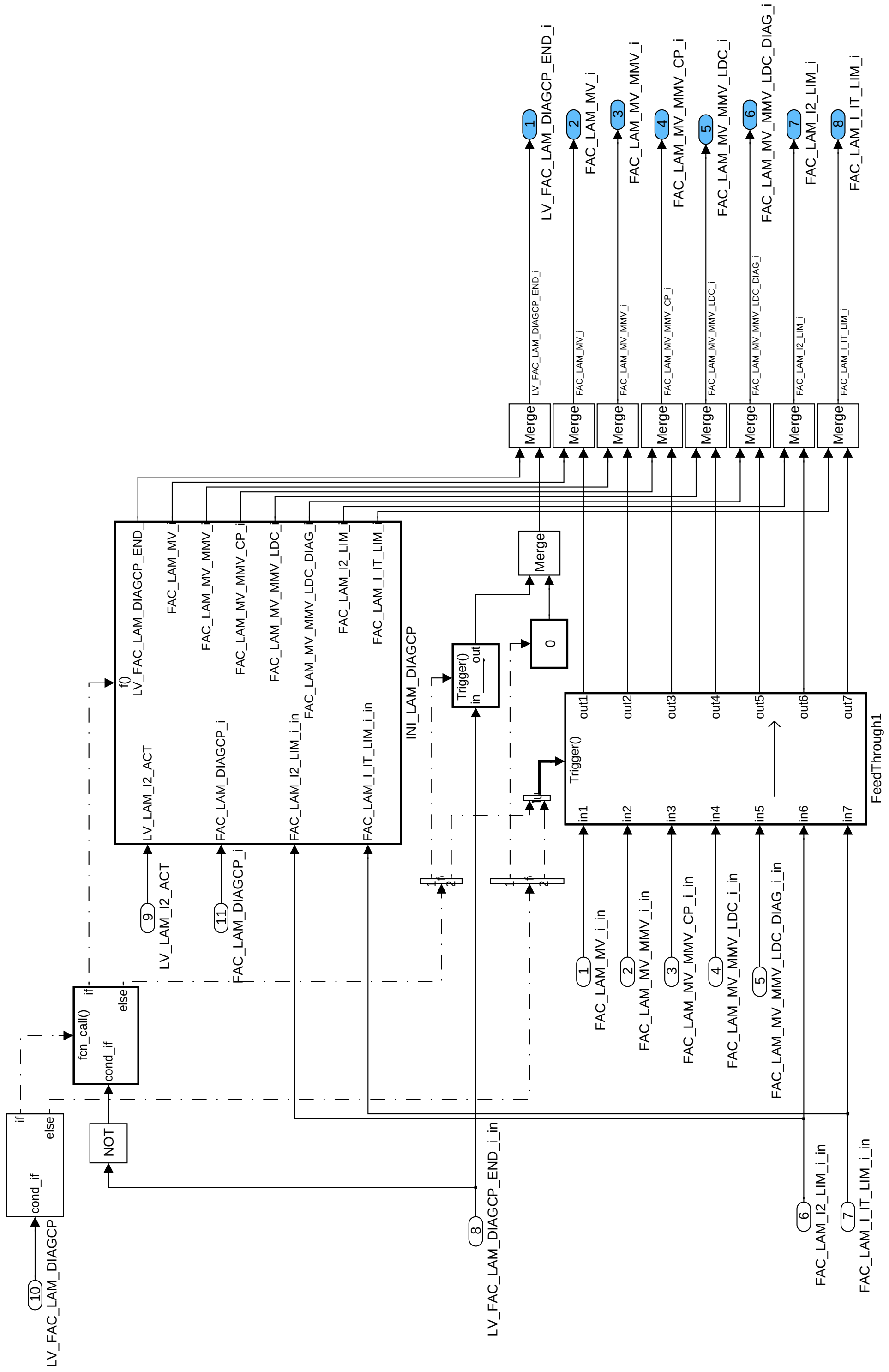


Figure 35 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__FAC_LAM__SHIFT/ MNG__INI_LAM_DIAGCP

Execution of initialization (INI_LAM_DIAGCP)

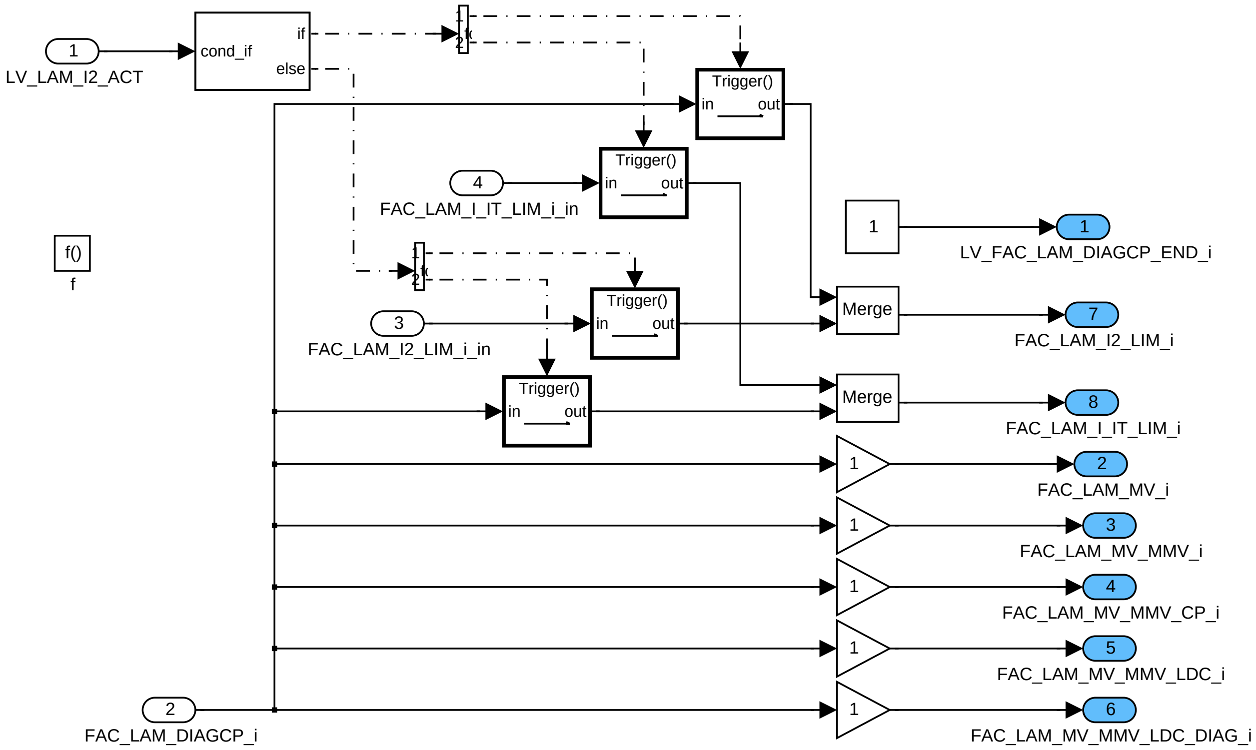


Figure 36 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__FAC_LAM__SHIFT/ MNG__INI_LAM_DIAGCP/ INI_LAM_DIAGCP

Determination of controller parameters (FAC_GAIN_P/I/I2/D_LAM[i])

Switching of controller parameters depending on specific engine operating states e.g. idle speed (to be defined in the application incidences) is controlled by the value of LV_LAM_GAIN_SWI. If this flag is set to 1 a special parameter set is applied.

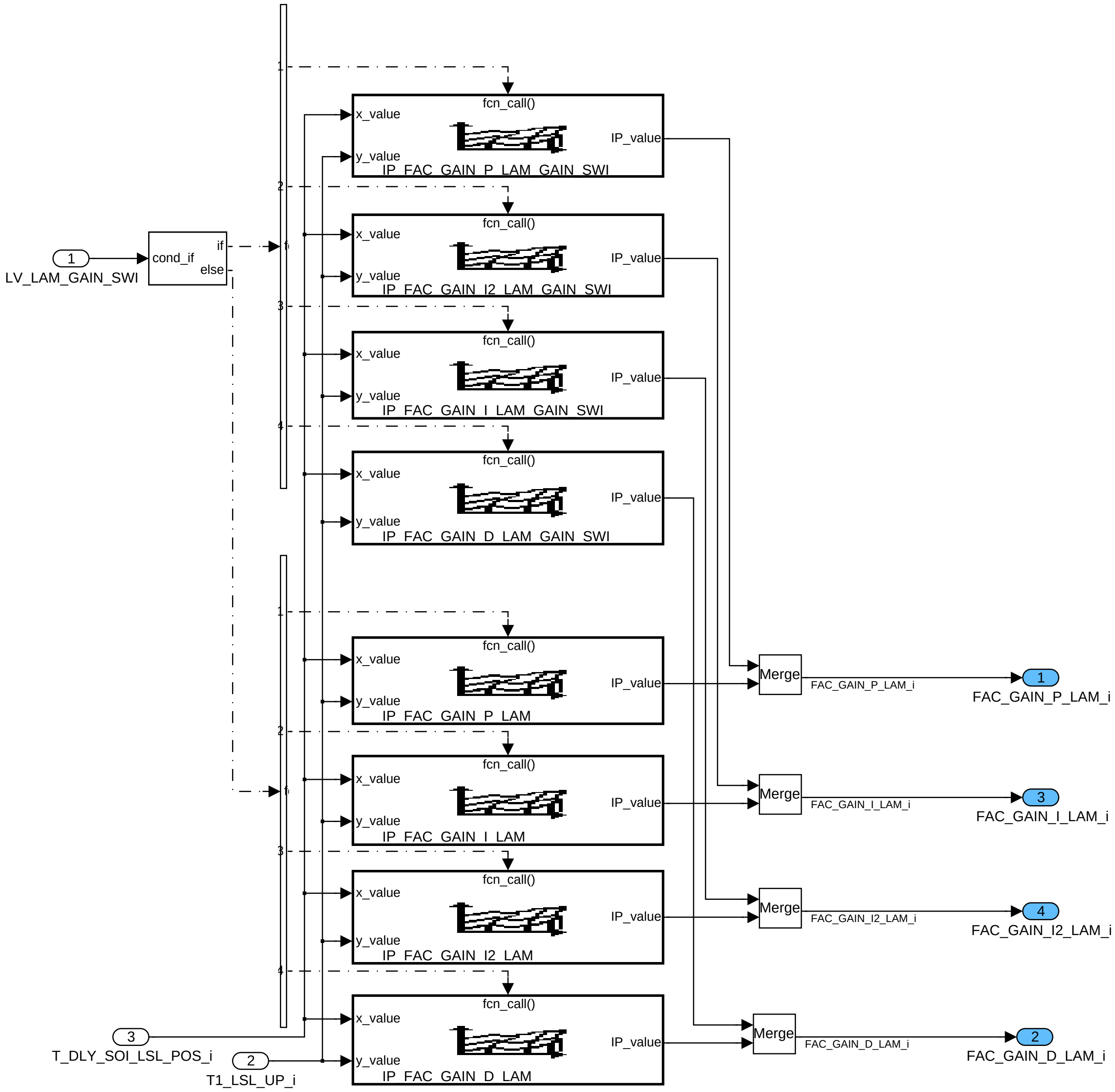


Figure 37 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ DET__CTL_GAIN

P share calculation (FAC_LAM_P_LIM[i])

The P share is calculated by multiplying the controller difference FAC_DIF_LAM_IN[i] with the P share gain FAC_GAIN_P_LAM[i]. Then limitation depending on active ramp open of canister purge is applied. The limited P share is given by the variable FAC_LAM_P_LIM[i].

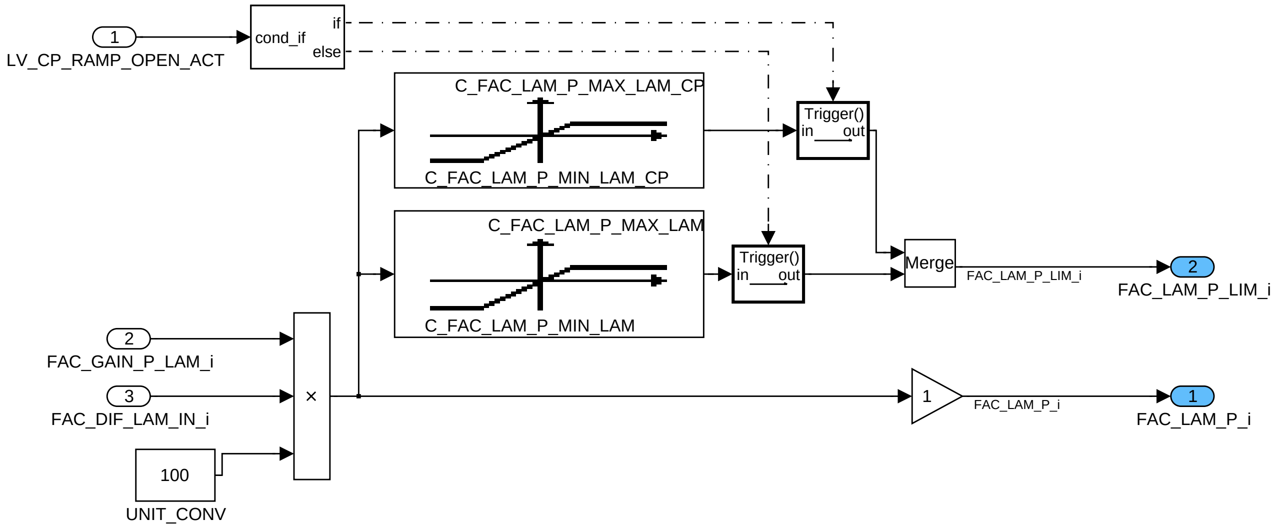


Figure 38 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ CLC__FAC_LAM_P

I share calculation

Depending on the flag LV_LAM_I2_ACT the lambda controller is calculated as PII^{2}D controller or PID controller.

I shares calculation with active I2 share

Summation of controller differences; only with active I2 share calculation (FAC_DIF_SUM_LAM_IN[i])

The controller differences FAC_DIF_LAM_IN[i] are summed up with the variable FAC_DIF_LAM_IN[i]. The summation is not applied in case of:

- controller limitation on positive side (LV_FAC_LAM_LIM_MAX[i] = 1) and positive FAC_DIF_LAM_IN[i],
- controller limitation on negative side (LV_FAC_LAM_LIM_MIN[i] = 1) and negative FAC_DIF_LAM_IN[i] or
- injected fuel mass is at its minimum value (LV_TI_1_HOM_MIN = 1) and negative FAC_DIF_LAM_IN[i].

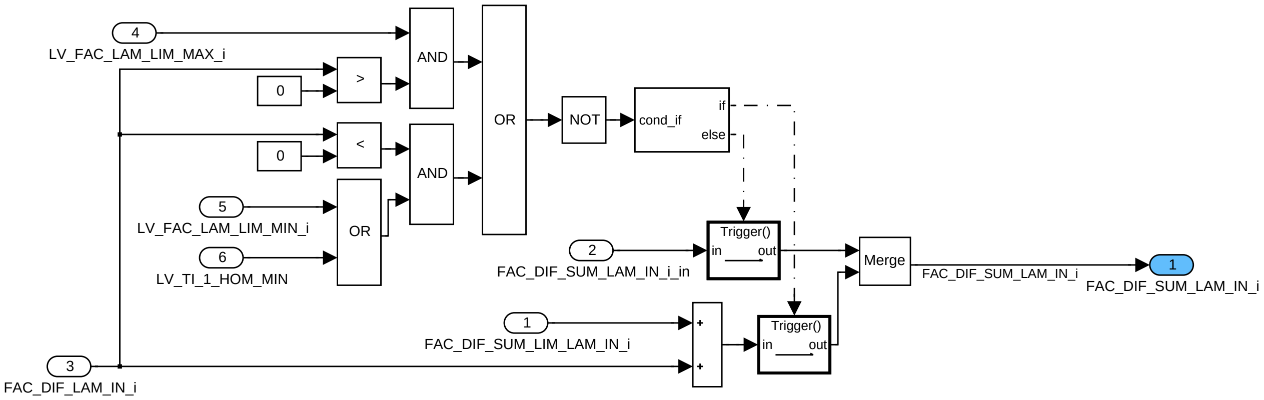


Figure 39 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_DIF_SUM_LAM_IN

Determination of I and I2 share shift based on downstream signal evaluation (only with active I2 share)

By means of the lambda controller I^2 share an oxygen balancing for the catalyst is applied. This balancing is only valid if the oxygen storage capacity (OSC) of the catalyst is between 0 and 100%. By observing the downstream signal these conditions can be evaluated. When catalyst OSC reaches its limits the integration of the control differences should be stopped. In order to influence the control behavior as less as possible the control differences sum is stored when OSC reaches the limits and applied later on when the disturbance is rejected. As this "manipulation" of the control differences sum leads to a shift of the I share the I^2 share is shifted by the same amount in the opposite direction. So the controller output is neutral.

Shift due to lean mixture detection (FAC_LAM_DELTA_I_I2_SHIFT[i])

In case of rising edge of LV_VLS_DOWN_THD_AFL_LAM[i] (OSC reaches 100%) the summarized controller differences FAC_DIF_SUM_LAM_IN[i] are stored in the variable FAC_DIF_SUM_LAM_IN_SAVE[i]. Furthermore the time out counter TOUT_I_I2_SHIFT_VLD_AFL[i] is initialized by the calibration value C_TOUT_I_I2_SHIFT_VLD.

If the controller compensates a disturbance (when FAC_DIF_LAM_IN_MMV[i] becomes 0) before the timer reaches zero, both, I share and I^2 share are shifted in opposite directions by the same amount. The shift of the I share is realized by setting the summarized controller differences to the saved value. The amount of the shift for the I^2 share FAC_LAM_DELTA_I_I2_SHIFT[i] is calculated from the difference between saved value and current summarized controller differences (of course before setting this latter variable to the saved value) multiplied with the I share gain and the sample time. The shift of the I^2 share itself is realized where the I^2 share is calculated by adding always FAC_LAM_DELTA_I_I2_SHIFT[i] which normally is 0 when no shift should be applied.

When the timer TOUT_I_I2_SHIFT_VLD_AFL[i] reaches 0 before the disturbance is compensated no shift is applied anymore because a later zero crossing of FAC_DIF_LAM_IN_MMV[i] cannot be surely assigned to the disturbance.

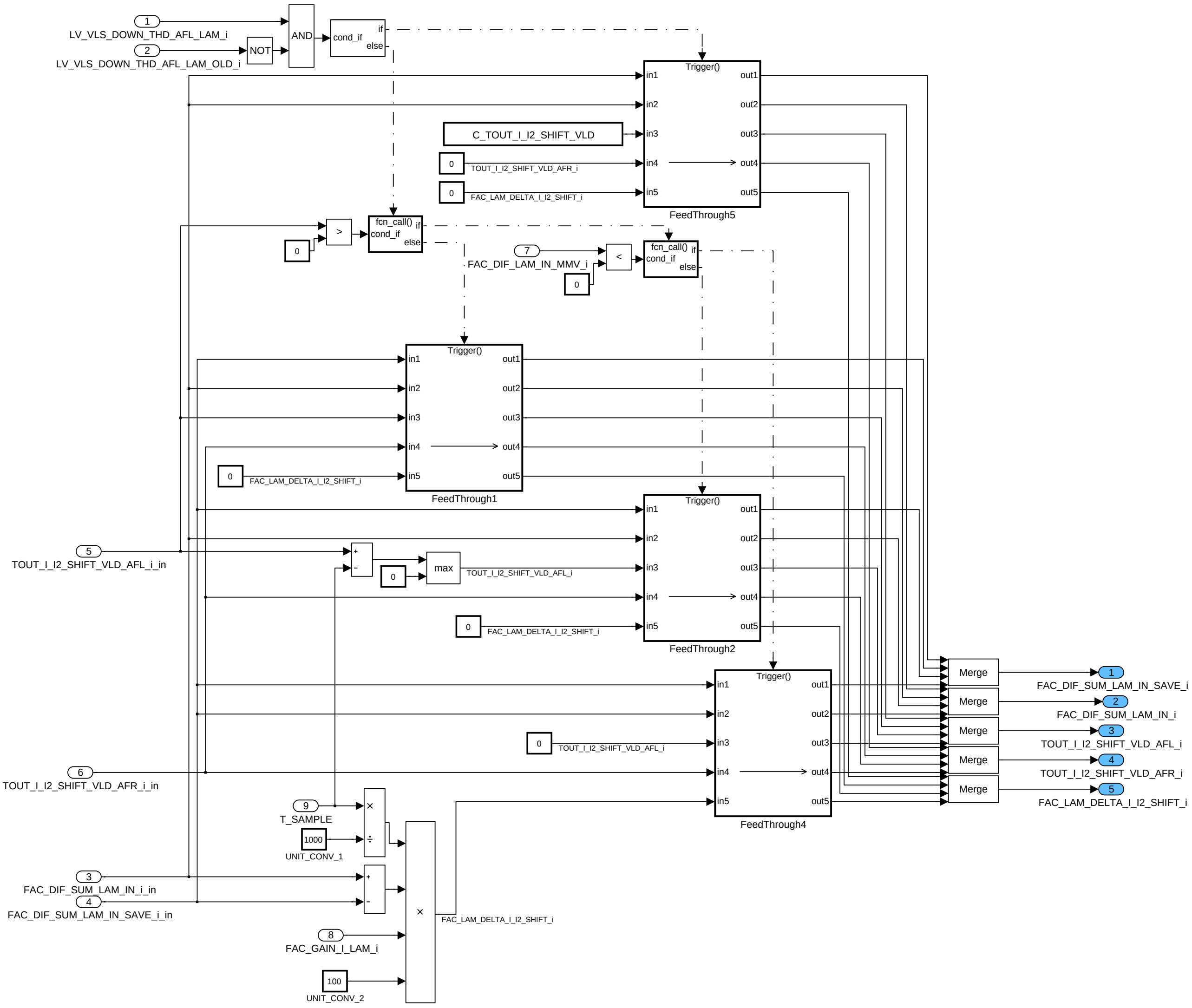


Figure 40 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_LAM_I__FAC_LAM_I2/ DET__FAC_LAM_I_FAC_LAM_I2_SHIFT/ FAC_LAM_I_FAC_LAM_I2_SHIFT_AFL

Shift due to rich mixture detection (FAC_LAM_DELTA_I_I2_SHIFT[i])

The realization for the opposite case (OSC reaches 0% abd rich mixture detection) is exactly the same as for the shift due to lean mixture detection. The summarized controller differences are stored in case of rising edge of LV_VLS_DOWN_THD_AFR_LAM[i] and the timer TOUT_I_I2_SHIFT_VLD_AFR[i] is used instead. The resulting shift of I and I^2 share is inverted.

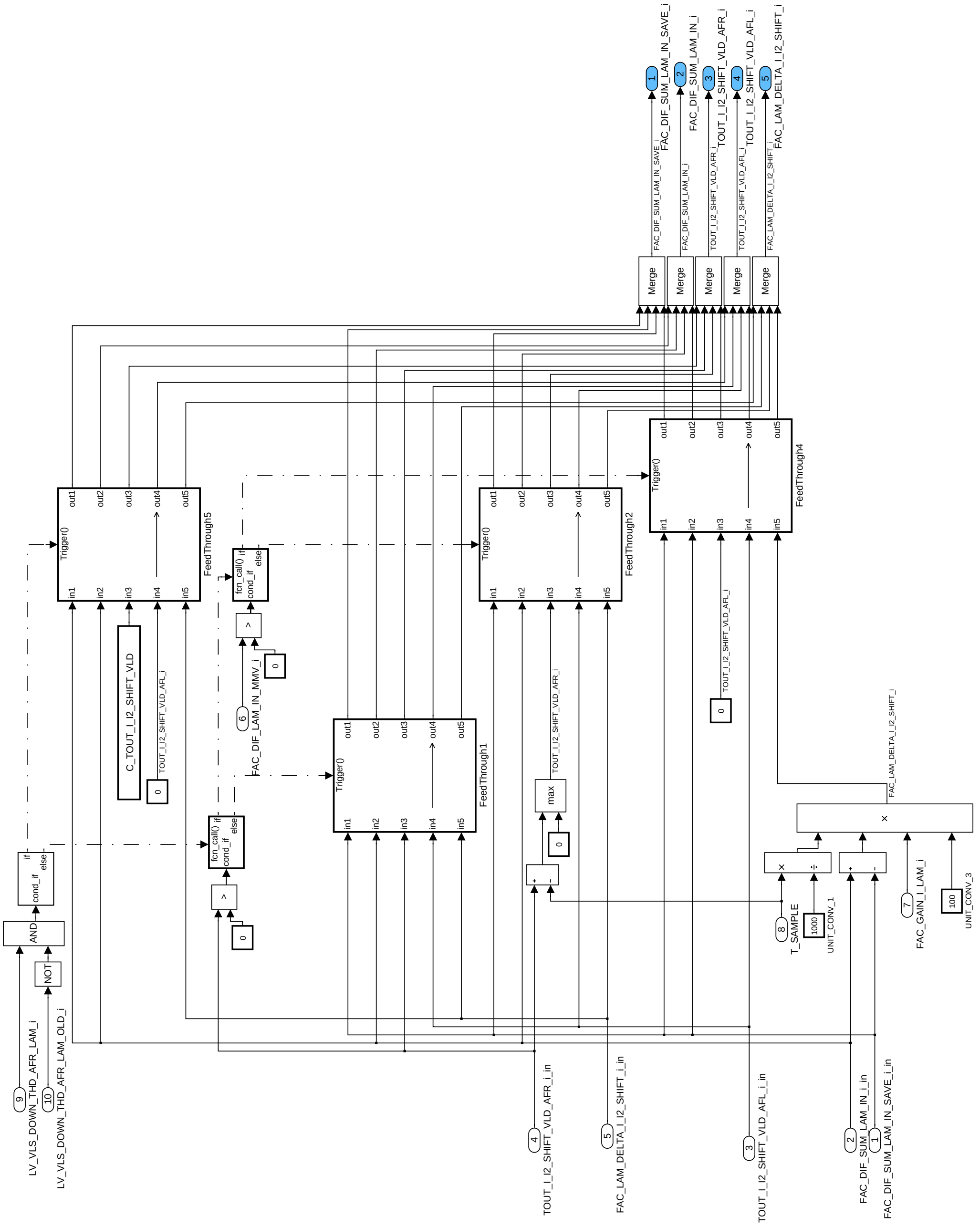


Figure 41 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_LAM_I__FAC_LAM_I2/ DET__FAC_LAM_I_FAC_LAM_I2_SHIFT/ FAC_LAM_I_FAC_LAM_I2_SHIFT_AFR

<u>I share calculation and anti wind up limitation (only with active I2 share calculation)</u>

The I share FAC_LAM_I[i] is calculated by multiplying the summarized controller differences with the I share gain and with the sample time. If FAC_LAM_I[i] reaches its physical limits it will be automatically limited but the summarized controller differences must be adapted to this limitation (anti wind up limitation).

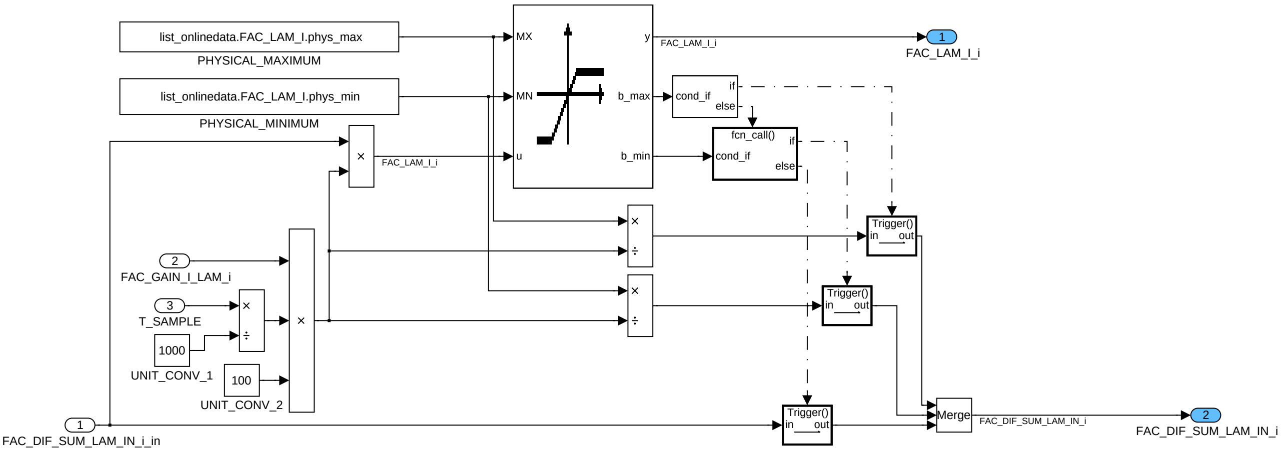


Figure 42 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_LAM_I__FAC_LAM_I2/ LIM__FAC_DIF_SUM_LAM_IN

I2 share calculation (FAC_LAM_I2[i])

The I^2 share calculation is not executed in case of:

- controller limitation on positive side (LV_FAC_LAM_LIM_MAX[i] = 1) and positive FAC_DIF_SUM_LAM_IN[i],
- controller limitation on negative side (LV_FAC_LAM_LIM_MIN[i] = 1) and negative FAC_DIF_SUM_LAM_IN[i] or
- injected fuel mass is at its minimum value (LV_TI_1_HOM_MIN = 1) and negative FAC_DIF_SUM_LAM_IN[i].

FAC_LAM_I2[i] remain unchanged in that case. Otherwise the calculation is active.

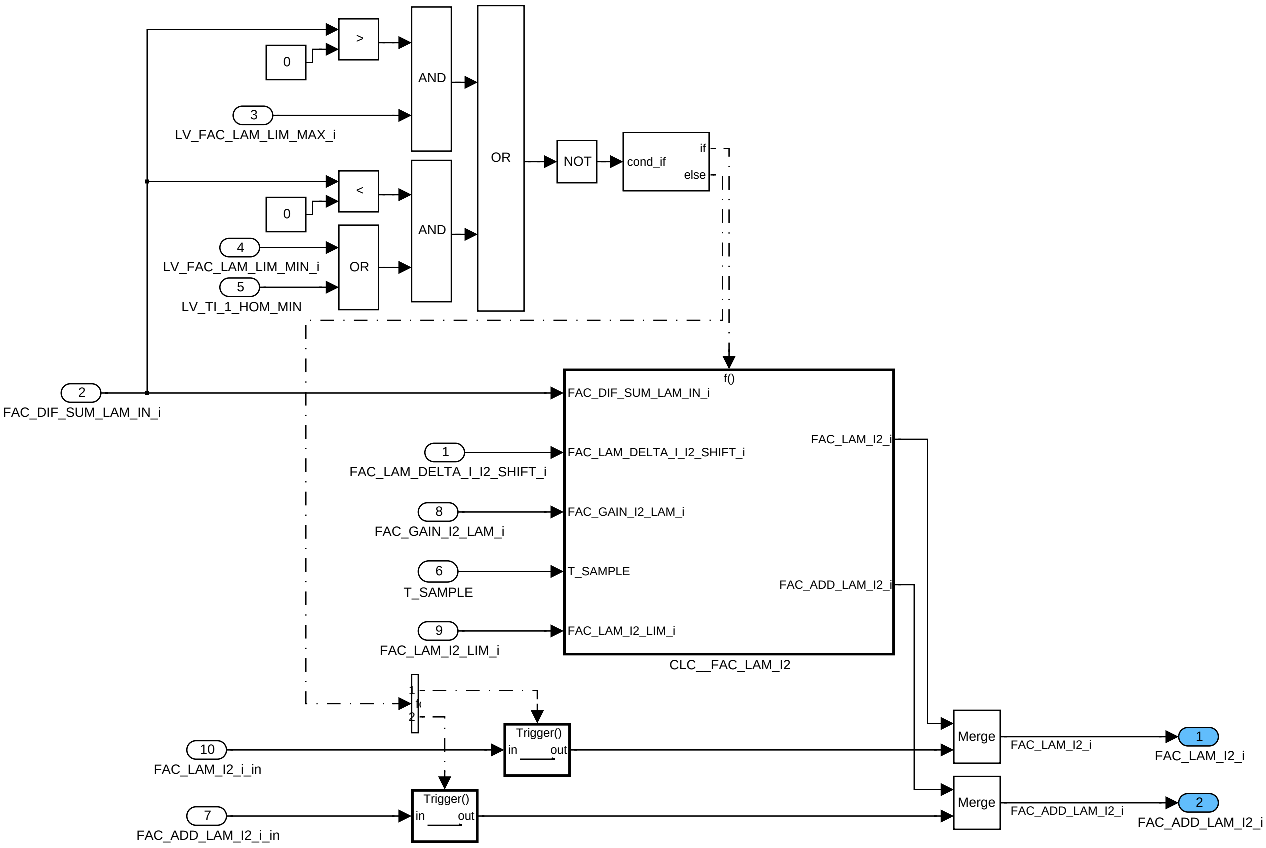


Figure 43 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_LAM_I__FAC_LAM_I2/ MNG__CLC__FAC_LAM_I2

Calculation of FAC_LAM_I2[i]

FAC_LAM_I2[i] is calculated by integrating the summarized controller differences FAC_DIF_SUM_LAM_IN[i] using the gain for the I^2 share FAC_GAIN_I2_LAM[i]. The sample time is considered twice (T_SAMPLE^2) as for the summarized controller differences no sample time was considered yet.

If FAC_DIF_SUM_LAM_IN[i] is not equal to 0 but the numerical result for FAC_ADD_LAM_I2[i] is 0 the resolution of this variable is added or subtracted depending on the sign of FAC_DIF_SUM_LAM_IN[i]. This prevents the I^2 share from being stopped although FAC_DIF_SUM_LAM_IN[i] is not 0 and is only needed because of fixed point arithmetic realization. FAC_ADD_LAM_I2[i] represents the amount that is added each sample step to the I^2 share. Furthermore a shift (if FAC_LAM_DELTA_I_I2_SHIFT[i] is not 0) is applied.

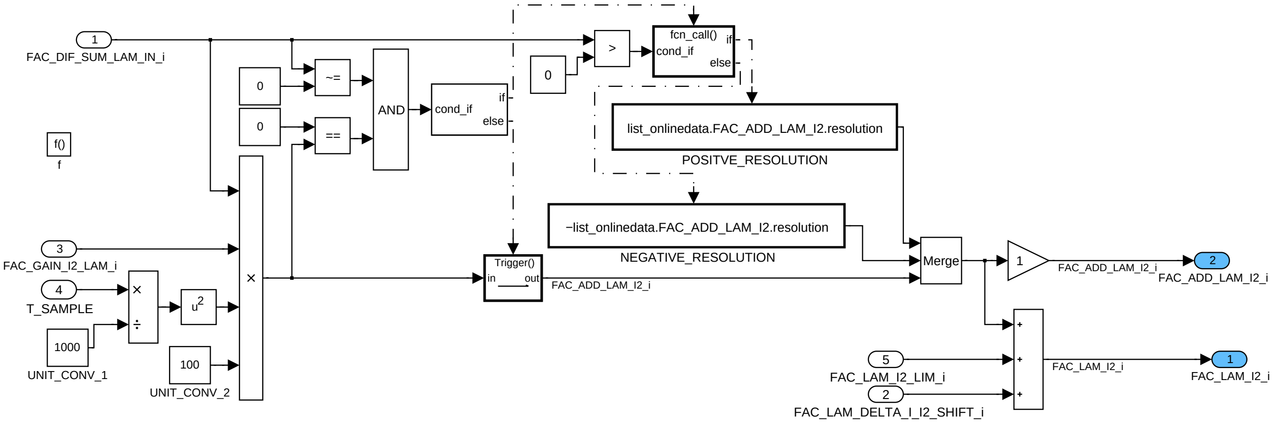


Figure 44 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_LAM_I__FAC_LAM_I2/ MNG__CLC__FAC_LAM_I2/ CLC__FAC_LAM_I2

<u>I share calculation without active I2 share (FAC_LAM_I[i])</u>

The I share is not calculated in case of:

- controller limitation on positive side (LV_FAC_LAM_LIM_MAX[i] = 1) and positive FAC_DIF_LAM_IN[i],
- controller limitation on negative side (LV_FAC_LAM_LIM_MIN[i] = 1) and negative FAC_DIF_LAM_IN[i] or
- injected fuel mass is at its minimum value (LV_TI_1_HOM_MIN = 1) and negative FAC_DIF_LAM_IN[i].

Otherwise FAC_LAM_I[i] is calculated.

The I^2 share FAC_LAM_I2[i] is always set to 0.

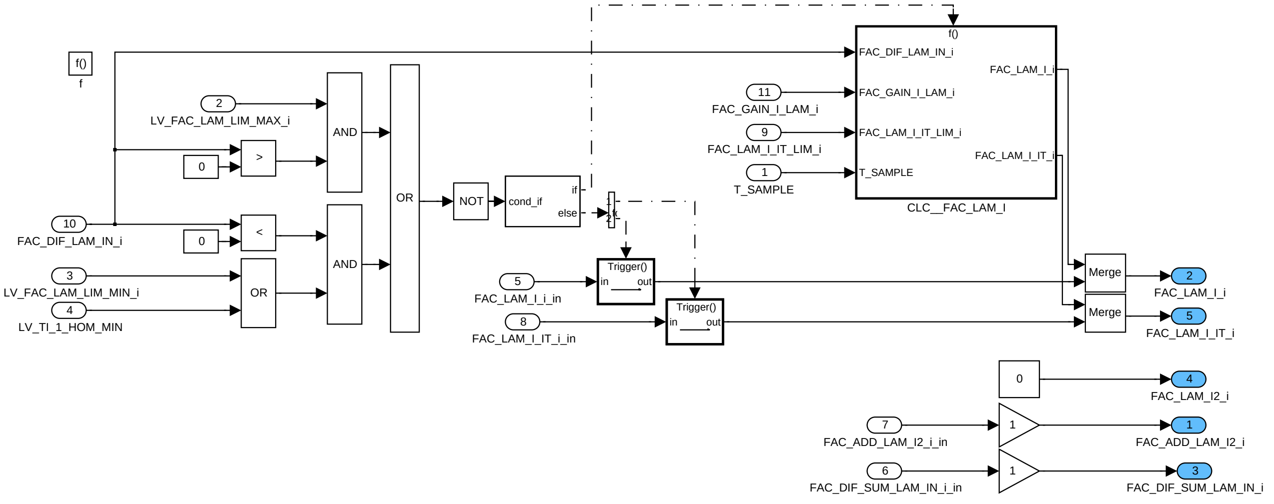


Figure 45 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_LAM_I_WOUT_I2

<u>Calculation of FAC_LAM_I[i]</u>

The I share is calculated with an internal variable FAC_LAM_I_IT[i] that has a higher resolution. FAC_LAM_I[i] is calculated by rounding FAC_LAM_I_IT[i] off to the resolution of FAC_LAM_I[i].

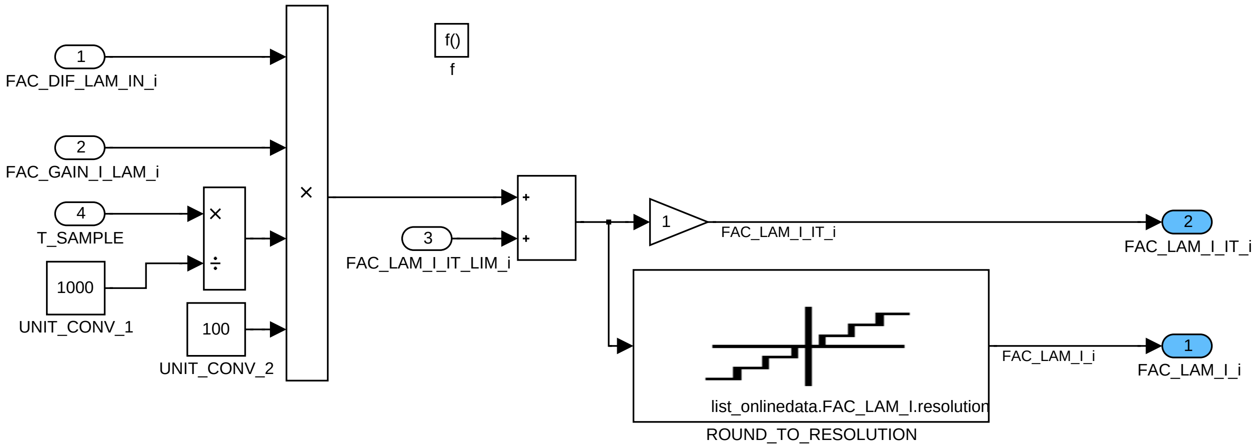


Figure 46 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ CLC__FAC_LAM_I_WOUT_I2/ CLC__FAC_LAM_I

Summation and limitation of I and I2 share (FAC_LAM_I_I2_SUM_LIM[i])

The I share and I^2 share are summarised in the variable FAC_LAM_I_I2_SUM_LIM[i]. Furthermore the sum of both can be limited (during canister purge ramp open with different thresholds).

In case of limitation 2 cases must be distinguished:

1. LV_LAM_I2_ACT = 1:

-> FAC_DIF_SUM_LIM_LAM_IN[i] = 0

-> FAC_LAM_I2_LIM[i] = FAC_LAM_I_I2_SUM_LIM[i]

2. LV_LAM_I2_ACT = 0:

-> FAC_LAM_I_IT_LIM_LIM[i] = FAC_LAM_I_I2_SUM_LIM[i]

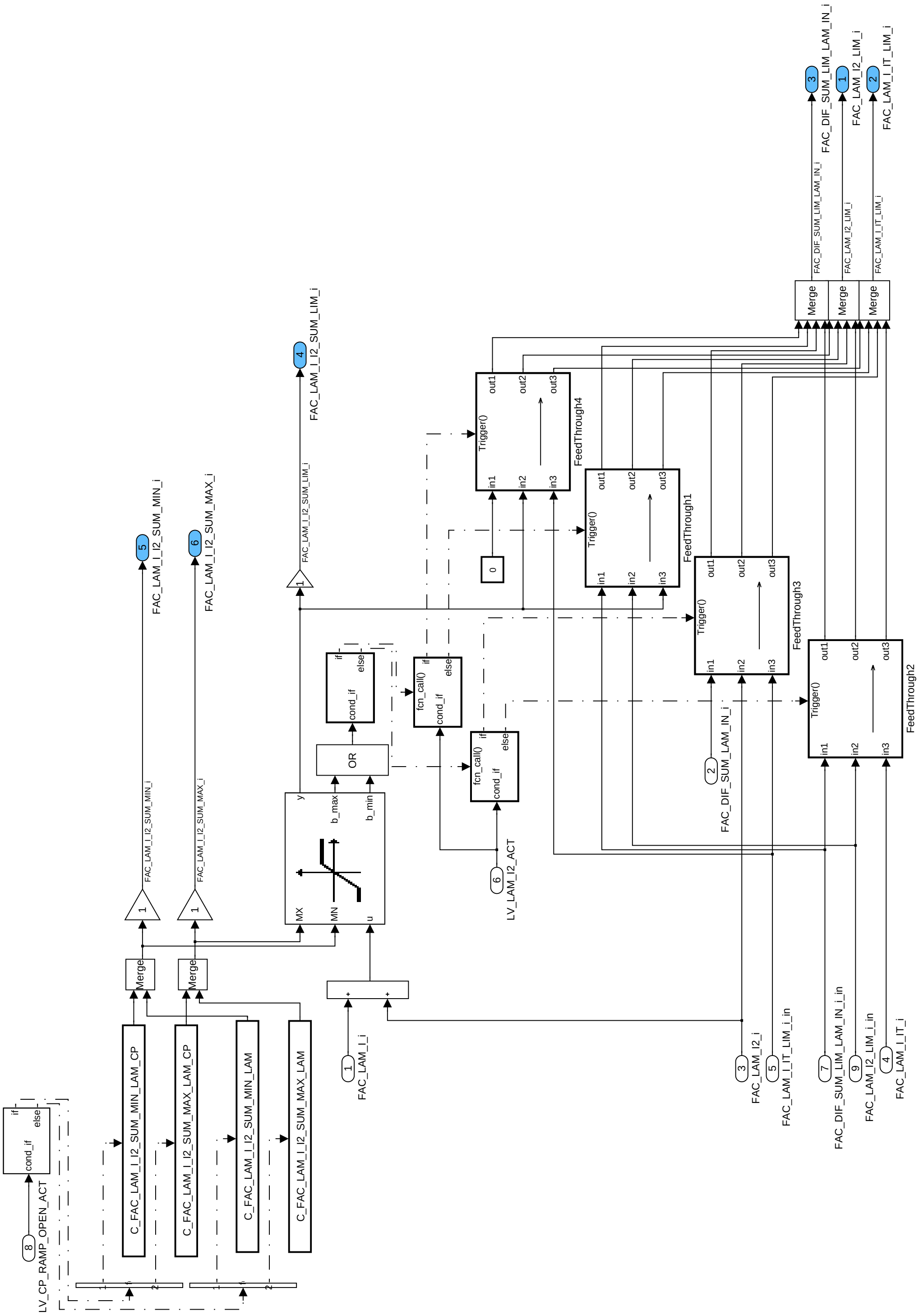


Figure 47 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__CLC__FAC_LAM_I__FAC_LAM_I2/ SUM__FAC_LAM_I__FAC_LAM_I2

D share calculation (FAC_LAM_D_LIM[i])

The D share is calculated by dividing the difference of FAC_DIF_LAM_IN[i]$_N$ and FAC_DIF_LAM_IN[i]$_{N-1}$ by T_SAMPLE and multiplying the result with the gain for the D share (FAC_GAIN_D_LAM[i]). Then limitation is applied and the limited D share is stored in the variable FAC_LAM_D_LIM[i].

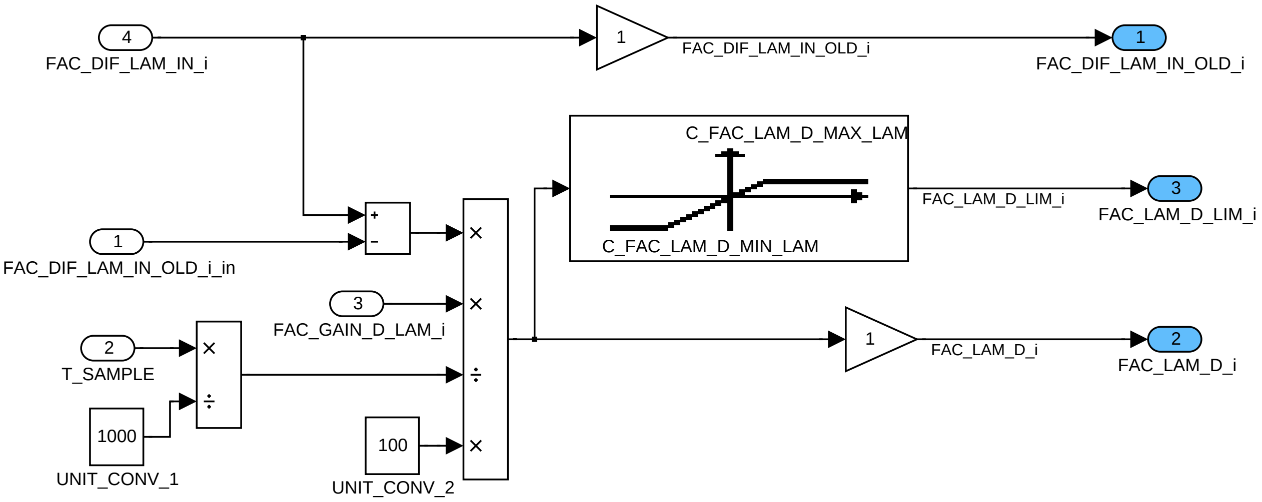


Figure 48 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ CLC__FAC_LAM_D

Calculation of controller output signal (FAC_LAM_OUT[i])

The limited sum output of the I and I^2 share FAC_LAM_I_I2_SUM_LIM[i] together with the limited P and D share (FAC_LAM_P_LIM[i] and FAC_LAM_D_LIM[i]) are summarised in the variable FAC_LAM_OUT[i].

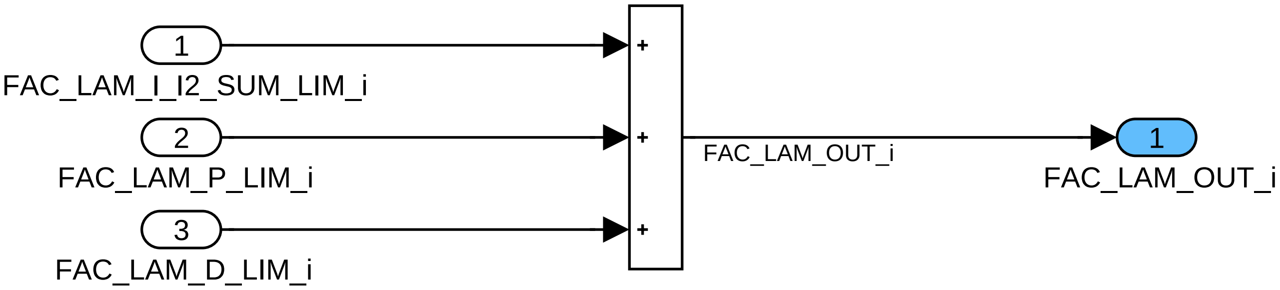


Figure 49 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ CLC__FAC_LAM_OUT

Calculation of moving mean values (FAC_LAM_MV[i] et al.)

For other functions like lambda adaptation, catalyst efficiency diagnosis and canister purge moving mean values of the remaining controller offset need to be calculated.

First FAC_LAM_MV[i] is calculated as filtered I (+I^2) share FAC_LAM_I_I2_SUM_LIM. Then FAC_LAM_MV[i] is input to all other moving mean value calculations.

The moving mean values can be shifted, initialized or limited by other functionalities as described before in "shift and initialization of lambda controller" and afterwards in "limitation under non stationary conditions".

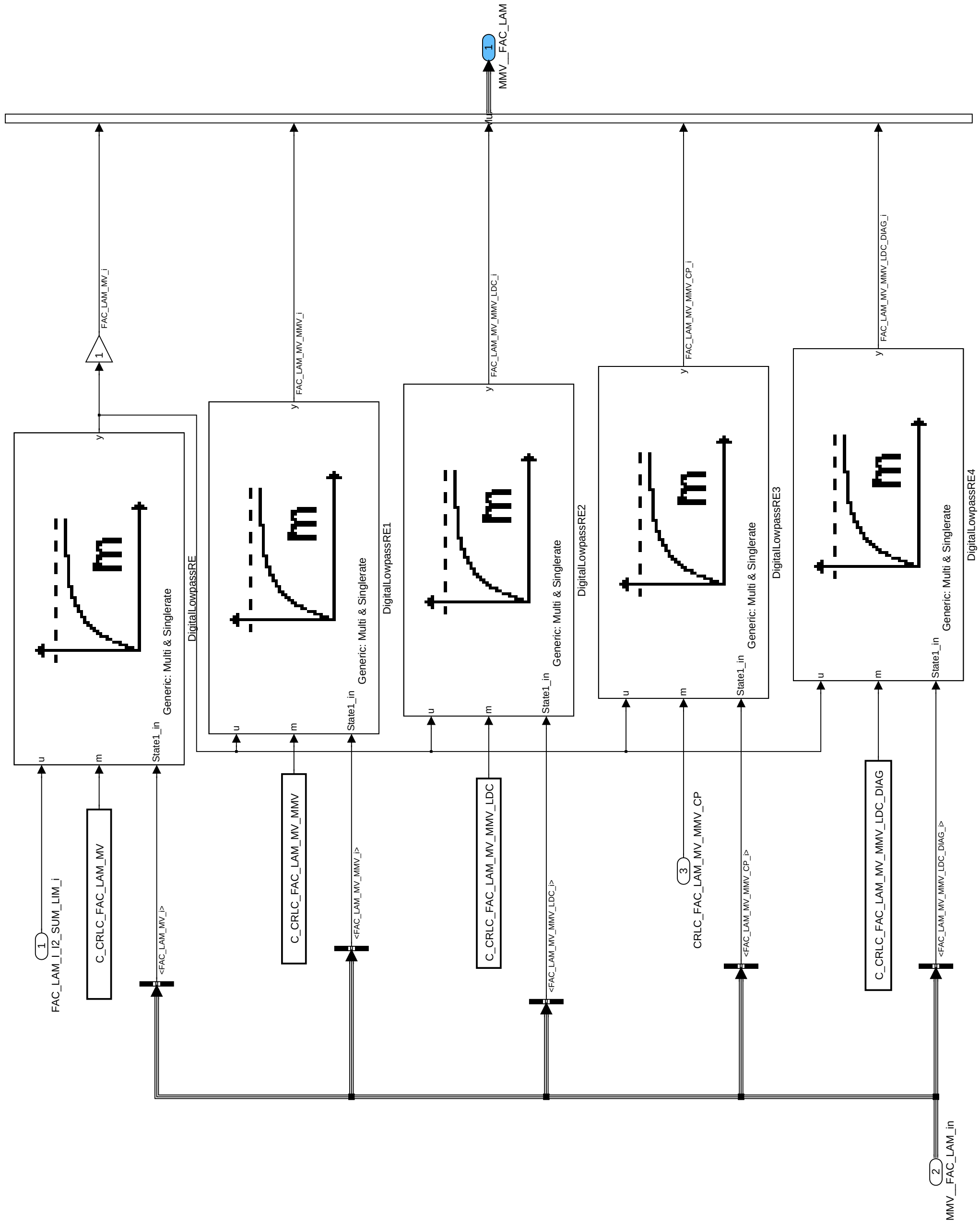


Figure 50 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ CLC__FAC_LAM_MV

<u>Limitation under non stationary conditions</u>

Under non stationary operating conditions (LV_LAM_NOT_STAT_CDN is set to 1), the value of the controller output shall be kept inside a tolerable area. In case of exceeding the calibrated threshold the lambda controller output FAC_LAM_OUT[i] must be limited and the initial values for integrators and low pass filters must be set to the limited output.

If the controller was stopped by LV_LAM_STOP[i] and released later while non stationary conditions are still valid the following cases must be distinguished:

a) LV_LAM_STOP_SHO_PER[i] = 1, then

- no reset of LV_FAC_LAM_LIM_NOT_STAT_CDN[i]
- no renewed determination of thresholds

b) LV_LAM_STOP_SHO_PER[i] = 0, then

- reset of LV_FAC_LAM_LIM_NOT_STAT_CDN[i]
- new determination of thresholds

If the lambda controller had been deactivated (LV_LAM_LSCL[i] = 0) and was activated later while non stationary conditions are still valid then the thresholds must be re-calculated.

Storage of limitation values for non stationary conditions (FAC_LAM_MIN/MAX_NOT_STAT_CDN[i])

The threshold values are given by the maps IP_FAC_LAM_MIN_NOT_STAT_CDN and IP_FAC_LAM_MAX_NOT_STAT_CDN and are evaluated when LV_LAM_NOT_STAT_CDN switches from 0 to 1.

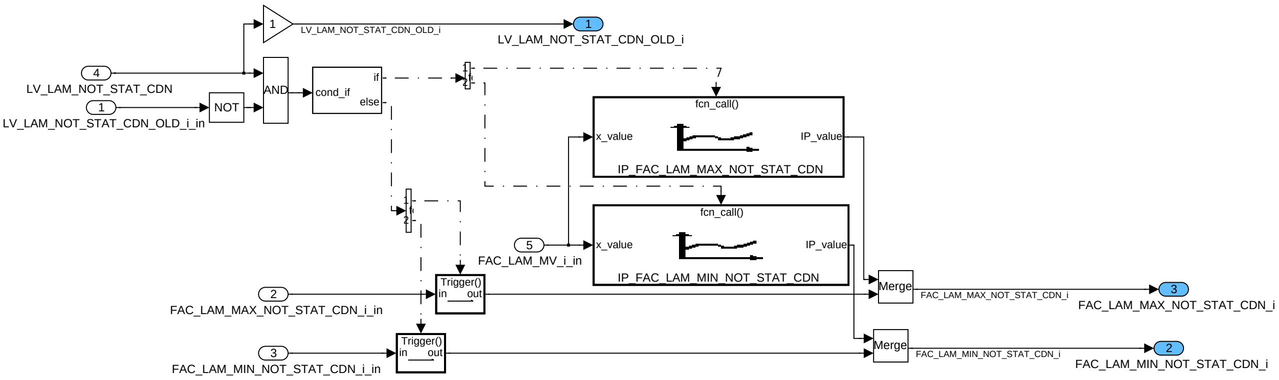


Figure 51 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__LIM_NOT_STAT_CDN/ CLC__LIM_NOT_STAT_CDN

Limitation of controller output FAC_LAM_OUT[i] and moving mean values; indicated by LV_FAC_LAM_LIM_NOT_STAT_CDN[i]

In case of exceeding the thresholds while non stationary conditions are valid the flag LV_FAC_LAM_LIM_NOT_STAT_CDN[i] is set to 1. This flag is reset if the non stationary conditions are not fulfilled anymore or the limitation is not valid anymore.

FAC_LAM_OUT[i] is limited to the threshold values. If the I share FAC_LAM_I_I2_SUM_LAM[i] exceeds the thresholds too all moving mean values are set to the limitation value.

In case LV_LAM_I2_ACT is set FAC_LAM_I2_LIM[i] is set to the limitation value and FAC_DIF_SUM_LIM_LAM_IN[i] is set to 0 (in order to reset the I share and apply the threshold value to the I^2 share).

In case LV_LAM_I2_ACT is not set FAC_LAM_I_IT_LIM[i] is limited.

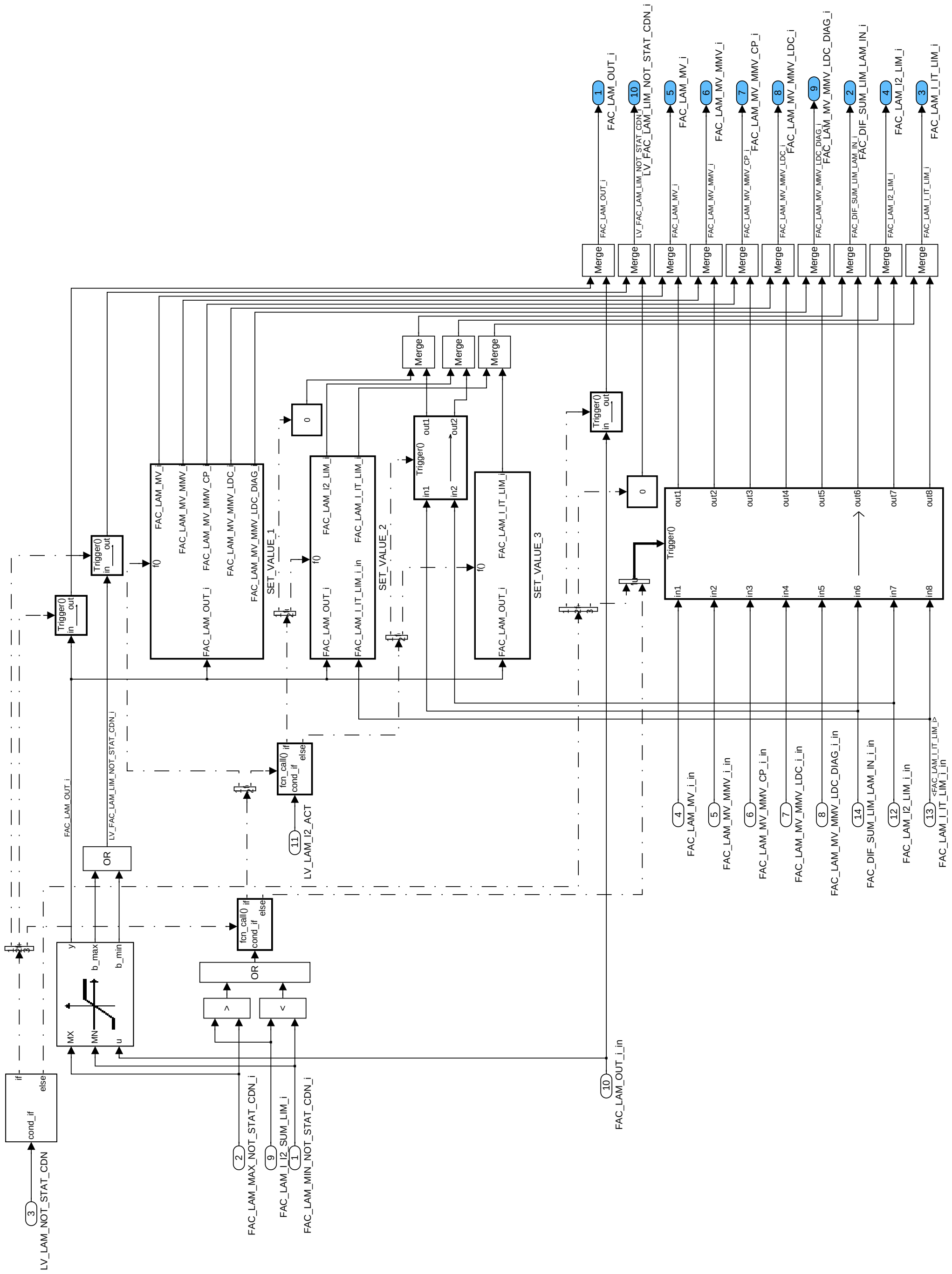


Figure 52 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__ON/ MNG__LIM_NOT_STAT_CDN/ LIM__FAC_LAM_OUT__NOT_STAT_CDN

STATE_LAM: “2:STOP_SHO_PER”

In state "2:STOP_SHO_PER " the D share is set to 0 and P, I and I^2 share are frozen. This state is for very short term controller stops.

Calculation of controller difference as richness (FAC_DIF_LAM_IN[i])

For details see state "0:OFF".

Evaluation of downstream sensor voltage (LV_VLS_DOWN_THD_AFL/AFR_LAM[i])

For details see state "0:OFF".

Reset D share (FAC_LAM_D_LIM[i])

The D share and its limited value are both set to 0.

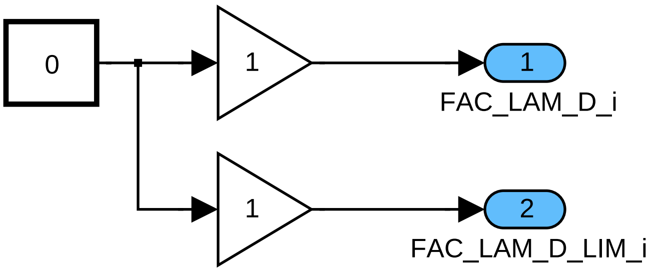


Figure 53 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__STOP_SHO_PER/ SET__FAC_LAM_D

Reset limitation flag for non-stationary conditions

If during short term controller stops the non-stationary conditions are no longer fulfilled (LV_LAM_NOT_STAT_CDN switches to 0) the flag indicating the limitation LV_FAC_LAM_LIM_NOT_STAT_CDN[i] is reset.

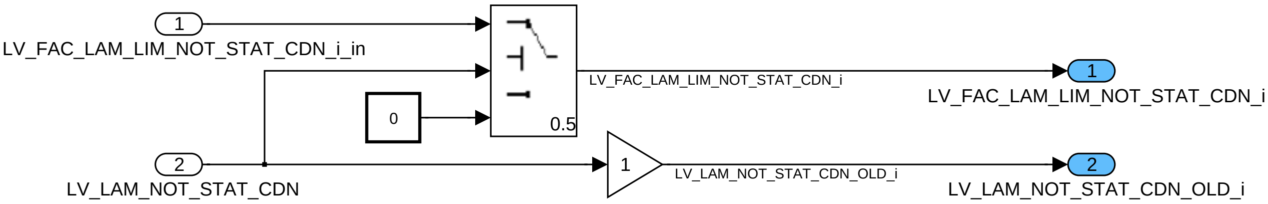


Figure 54 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__STOP_SHO_PER/ RST__LIM_NOT_STAT_CDN

Calculation of controller output signal (FAC_LAM_OUT[i])

FAC_LAM_OUT[i] is the sum of a frozen P and I share. In order to ensure limitation under non stationary conditions (which might be set in state “1:ON”) the flag indicating this limitation LV_FAC_LAM_LIM_NOT_STAT_CDN[i] is evaluated and – if set – the limitation is applied.

In case the request for limitation under non stationary conditions (LV_LAM_NOT_STAT_CDN) is reset and set again during state "2:STOP_SHO_PER" no new limitation is applied.

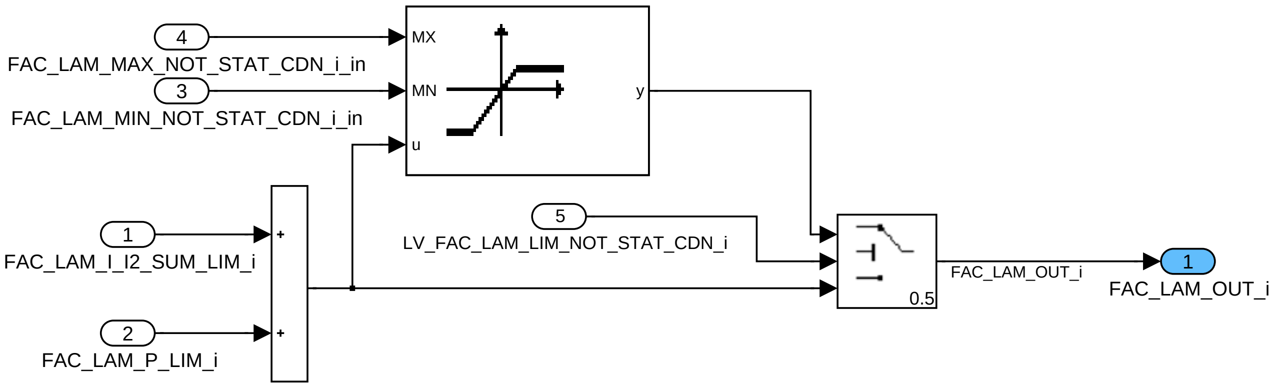


Figure 55 LACO_REQGNLACO0/ LAM/ operate/ CBK_EX_MNG/ CBK_EX_SPC/ STATE_LAM__STOP_SHO_PER/ CLC__FAC_LAM_OUT

STATE_LAM: “3:_STOP”

FAC_LAM_OUT[i] is set to 0.

Reset of all internal state variables and output values

For details see state "0:OFF".

Calculation of controller difference as richness (FAC_DIF_LAM_IN[i])

For details see state "0:OFF".

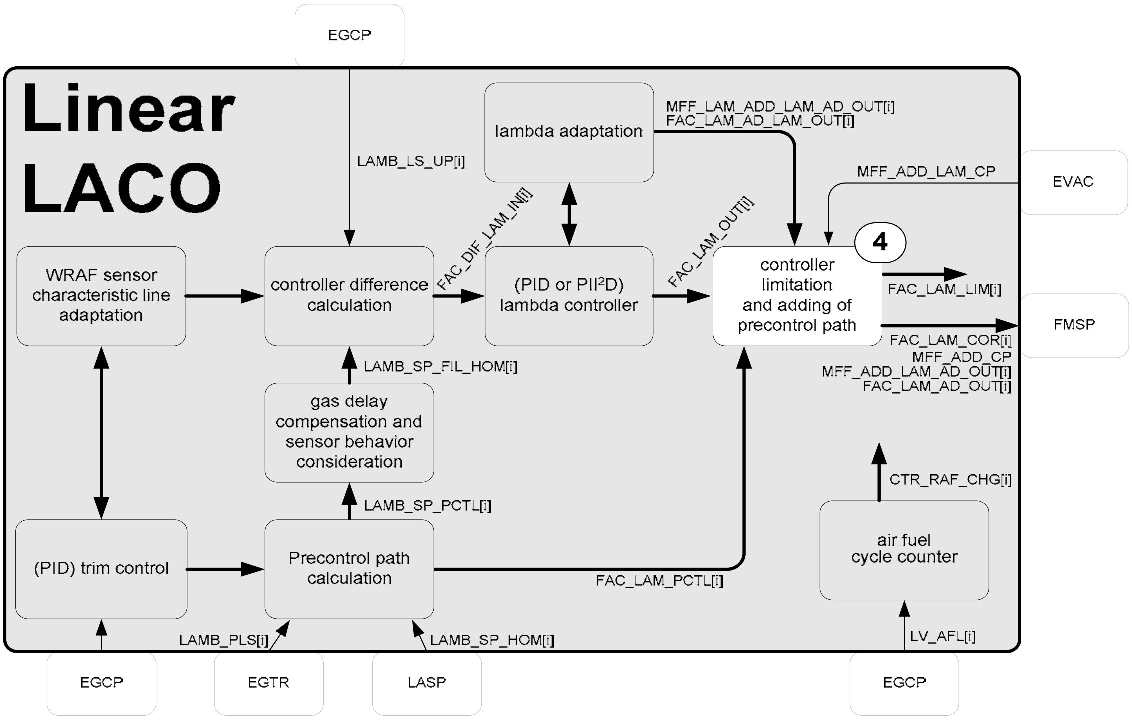
Evaluation of downstream sensor voltage (LV_VLS_DOWN_THD_AFL/AFR_LAM[i])

For details see state "0:OFF".

46.5.4 Controller output limitation and precontrol path correction consideration

The overall lambda control output FAC_LAM_COR[i] is computed taking into account also the precontrol intervention. This value is the correction for the fuel mass flow set point.

In order to synchronize output values of lambda controller, lambda adaptation and canister purge function some variables are fed through inside this function.



Sub-function in the whole module context

Additional remark:

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

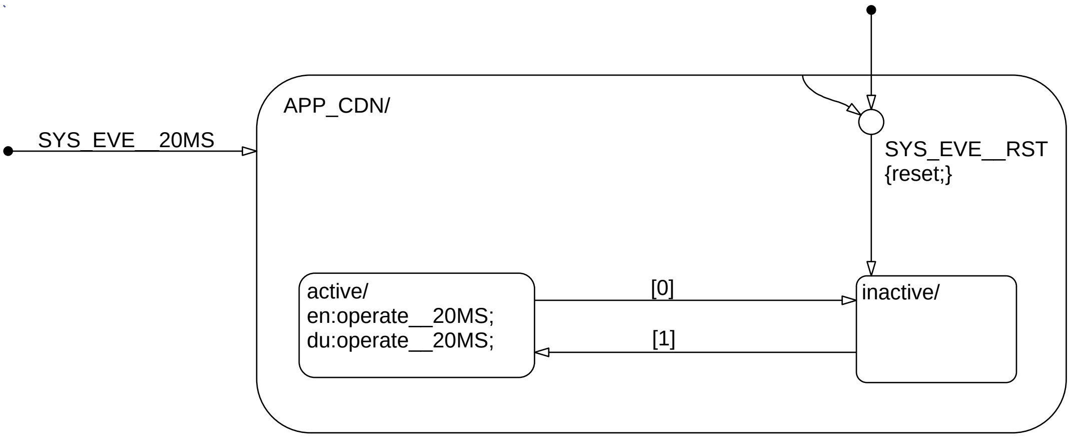


Figure 56 LACO_REQGNLACO0/ LAM_OUT/ APP_CDN/ Chart

Function Description

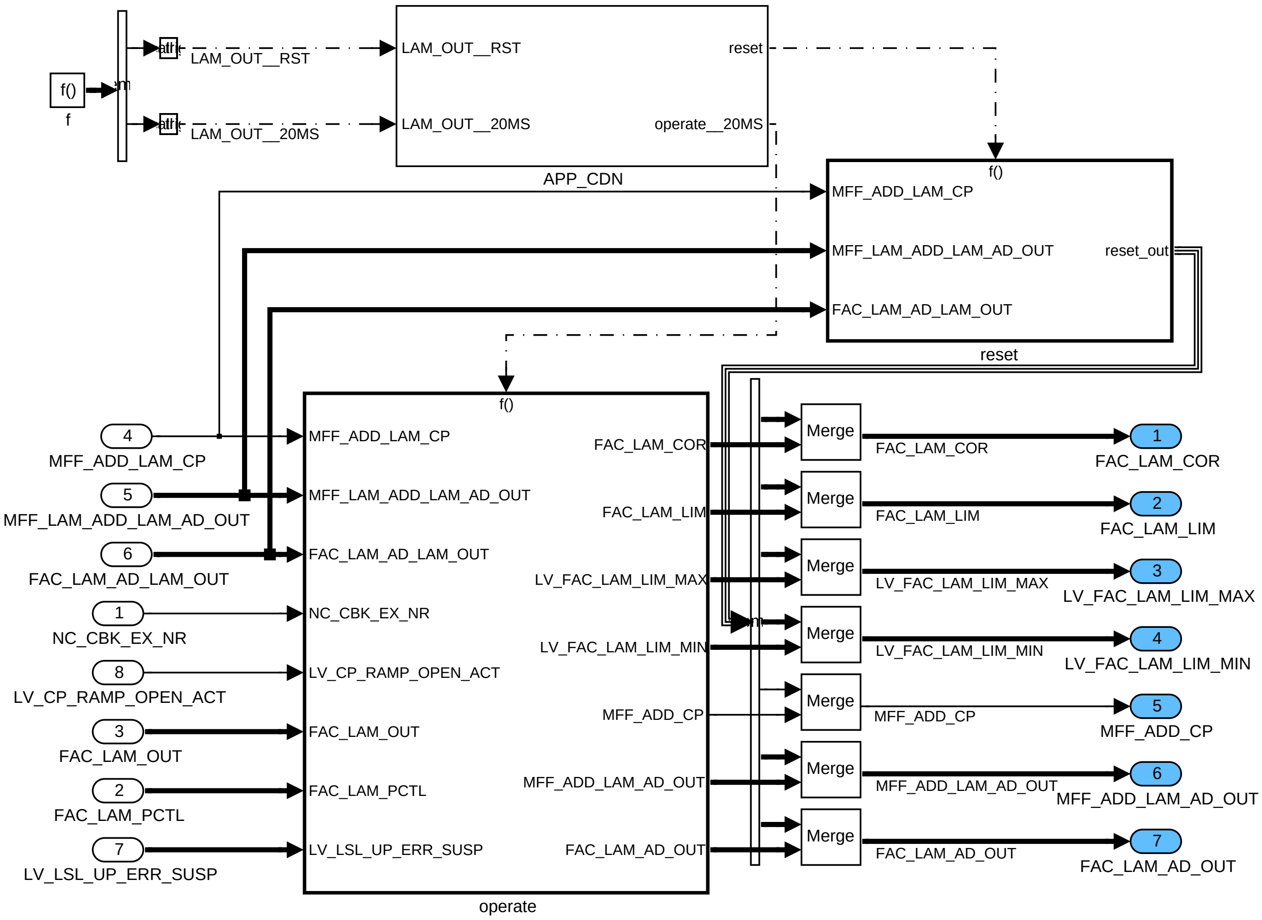


Figure 57 LACO_REQGNLACO0/ LAM_OUT

46.5.4.1 Initialization

MFF_ADD_CP is set to MFF_LAM_ADD_CP.

MFF_ADD_LAM_AD_OUT[i] is set to MFF_LAM_ADD_LAM_AD_OUT[i].

FAC_LAM_AD_OUT[i] is set to FAC_LAM_AD_LAM_OUT[i].

All other output variables are set to 0.

46.5.4.2 Formula section

Feed through of canister purge ouput varaible (MFF_ADD_CP)

The canister purge output variables (additive fuel mass flow) is written to the variable that is finally influencing the fuel mass set point.

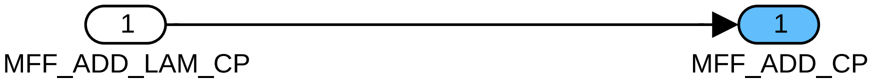


Figure 58 LACO_REQGNLACO0/ LAM_OUT/ operate/ CLC_MFF_ADD_CP

Exhaust bank specific functionality

Calculation of output variables (FAC_LAM_COR[i], MFF_ADD_LAM_AD_OUT[i], FAC_LAM_AD_OUT[i])

FAC_LAM_LIM[i] includes the limited output signal of the lambda controller. The limitation thresholds depend on whether the ramp open mode during canister purge is active or whether there is a suspicion of an oxygen sensor error. The limited output and the precontrol path correction together are summarised in the variable FAC_LAM_COR[i].

The lambda adaptation output variables (additive fuel mass flow and corrective factor) are written to the variables that are finally influencing the fuel mass set point.

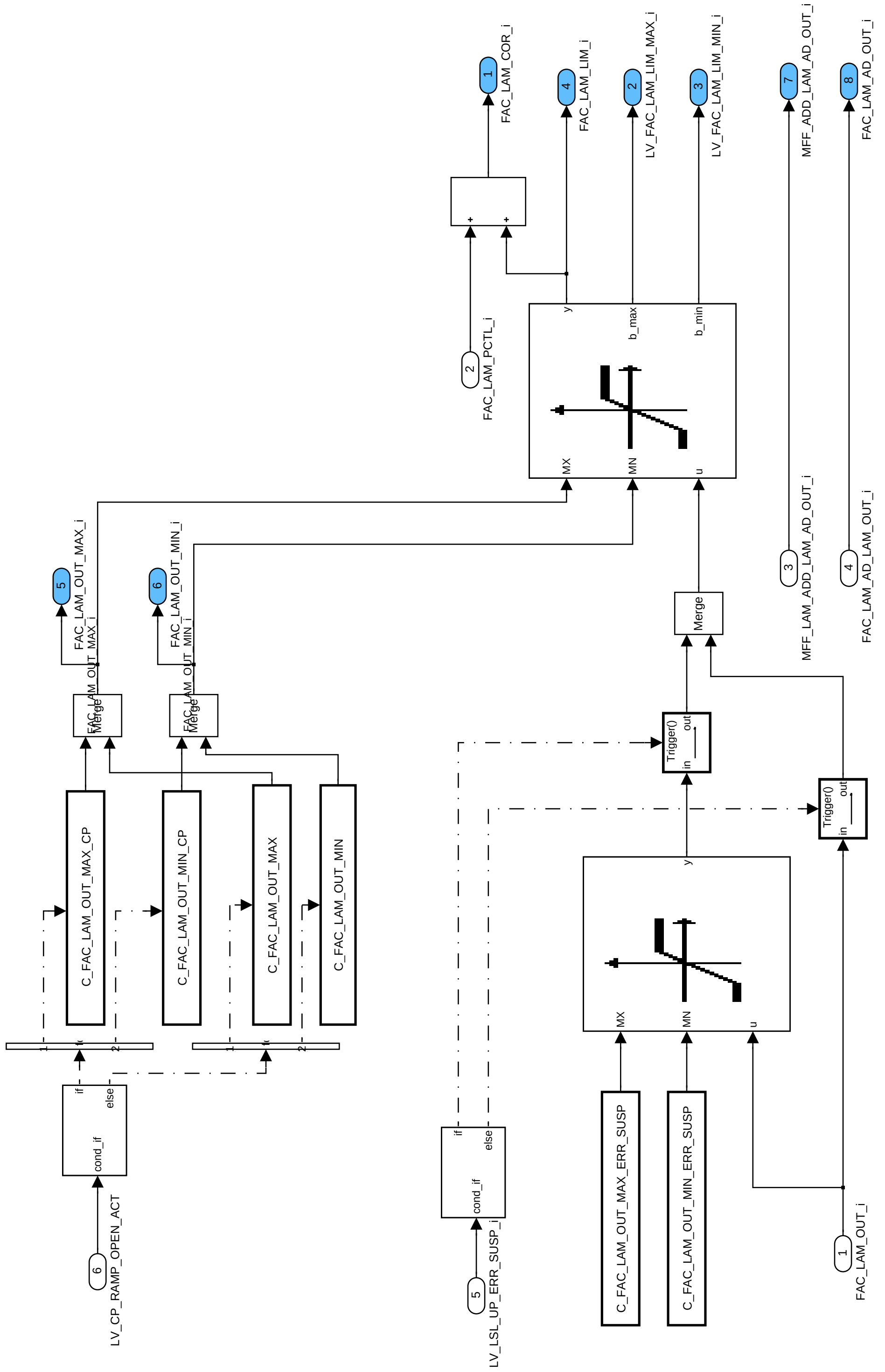
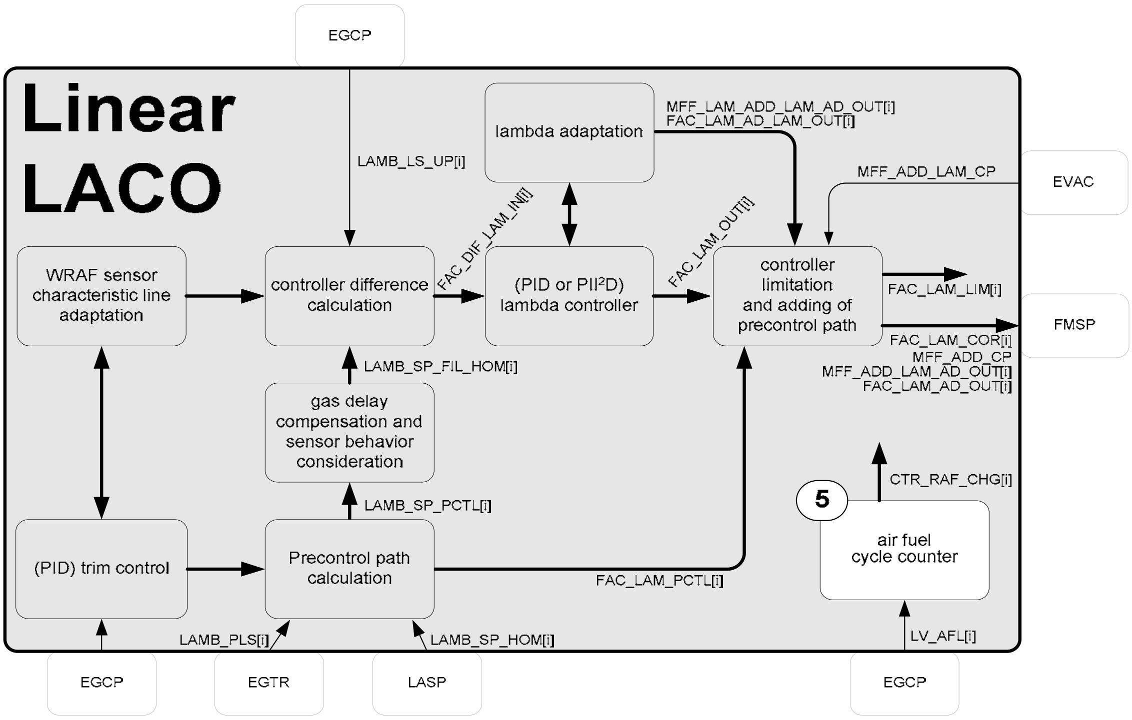


Figure 59 LACO_REQGNLACO0/ LAM_OUT/ operate/ CBK_EX_MNG/ CBK_EX_SPC

46.5.5 Calculation of air fuel ratio changes counter

In order to detect air fuel ratio changes in other functions a counter is provided that is incremented each time the air fuel ratio changes from lean to rich or vice versa.



Sub-function in the whole module context

Additional remark:

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

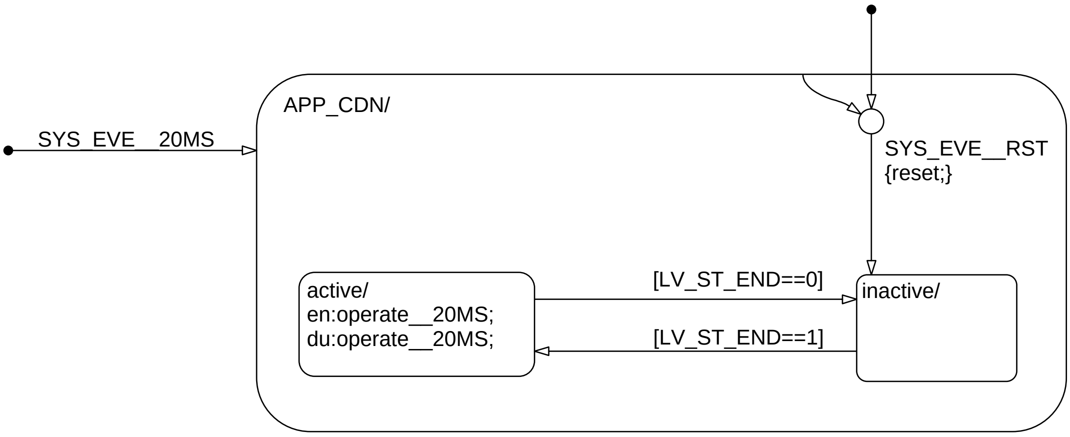


Figure 60 LACO_REQGNLACO0/ CHK_RAF_CHG/ APP_CDN/ Chart

Function Description

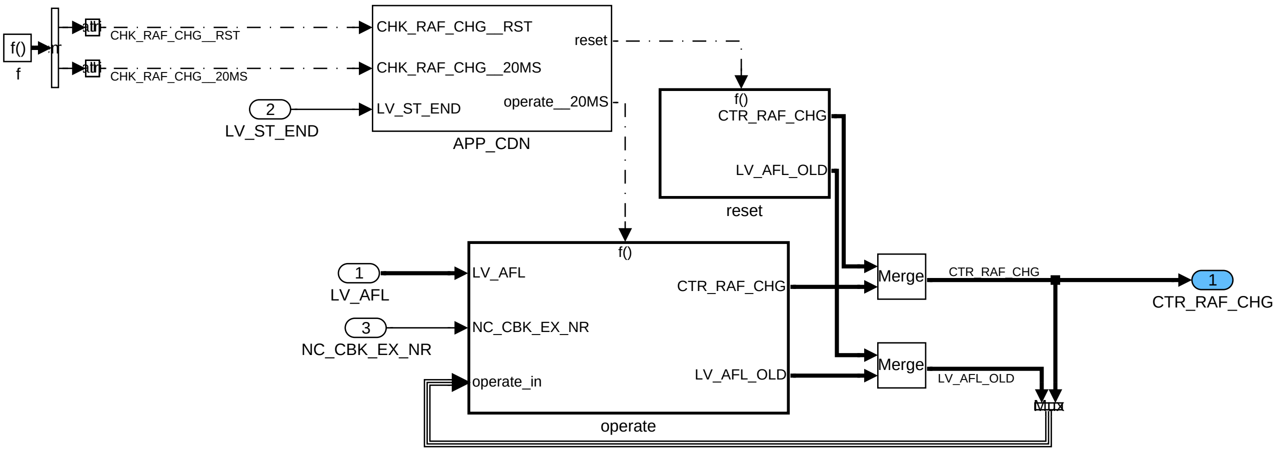


Figure 61 LACO_REQGNLACO0/ CHK_RAF_CHG

46.5.5.1 Initialization

All output variables are set to 0.

46.5.5.2 Formula section

Exhaust bank specific functionality

Calculation of CTR_RAF_CHG[i]

At each change of LV_AFL[i] the counter of air fuel ratio changes CTR_RAF_CHG[i] has to be incremented. The counter CTR_RAF_CHG[i] can overflow.

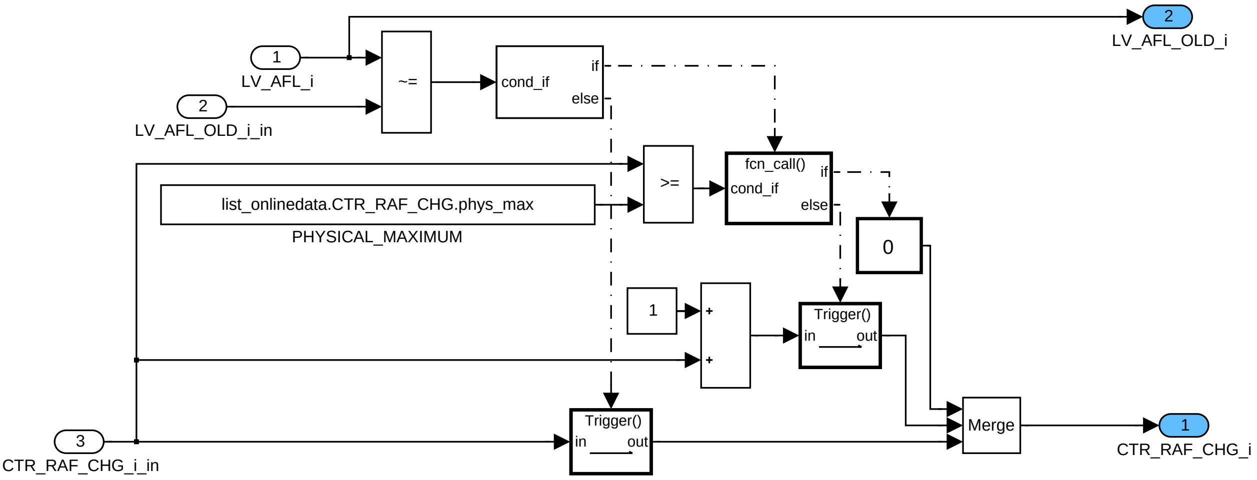


Figure 62 LACO_REQGNLACO0/ CHK_RAF_CHG/ operate/ CBK_EX_MNG/ CBK_EX_SPC

46.6 Application incidences for linear lambda control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_LSCL[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
inhibit signal for the controller deactivation					
LV_LAM_STOP_REQ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Request to stop the lambda controller					
LV_LAM_GAIN_SWI	O/V	0...1H	0...1	1	[-]
lambda controller gain switch for idle speed					
LAMB_SP_DELTA_LAM[NC_CBK_EX_NR]	O/V	F800...800H	-0.125...0.125	0.061e-3	[-]
lambda shift applied on lambda set point					
LAMB_AV_DELTA_LAM[NC_CBK_EX_NR]	O/V	F800...800H	-0.125...0.125	0.061e-3	[-]
lambda shift applied on measured lambda signal					
LV_STATE_LS_OPL_ERR[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
flag summarising errors that leads to open loop control					
FAC_NEUT_NOT_STAT_CDN	O/V	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
value for detection of non-stationary conditions for lambda controller limitation					
N_OFS_LAM_NOT_STAT	V	E020...1FE0H	-8160...8160	1	[rpm]
engine speed offset for detection of non stationary condition					
MAF_OFS_LAM_NOT_STAT	V	8000...7FFFH	-694.51059... 694.48940	0.0211948	[mg/stk]
air mass flow offset for detection of non stationary condition					
MAF_INT_LAM_NOT_STAT	V	0...FFFFH	0...1820.42	0.0277778	[g]
air mass flow integral for detection of non stationary condition					
MAF_INT_S_ACT	O/V	0...FFFFH	0...2912.66666	0.0444444	[g]
air mass flow integral for during the S or HOMS operation modi					
LV_LAMB_SP_AFR_THD[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag indicating that lambda setpoint is too rich and therefore the lambda controller is inhibited.					
LV_LAMB_SP_AFL_THD[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Flag indicating that lambda setpoint is too lean and therefore the lambda controller is inhibited.					
LAMB_DIF_AV_CTL_ST_OUT[NC_CBK_EX_NR]	V	F800...800H	-0.125...0.125	0.061e-3	[-]
Lambda offset in case of first lambda ctl start in DC					
LAMB_DIF_AV_CTL_ST_IN[NC_CBK_EX_NR]	V	F800...800H	-0.125...0.125	0.061e-3	[-]
start lambda difference for lambda ctl.					
LV_MAF_INT_MIN_LAM_REAC_SWI	O/V	0...1H	0...1	1	[-]
flag to switch limit of MAF_INT_MIN_LAM_REAC					
LV_LAM_STOP_SHO_PER_REQ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Request to trigger the short period stop mode of the Lambda controller					
FAC_TQ_REQ_OFS_LAM_NOT_STAT	V	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
actual torque value offset for detection of non-stationary condition					
FAC_IPT_SENS_MDL[NC_CBK_EX_NR]	O/V	0...400H	0...1	0.9766e-3	[-]
Interpolation factor for gas delay and sensor behavior model					
FAC_IPT_SENS_MDL_OPT[NC_CBK_EX_NR]	O/V	0...400H	0...1	0.9766e-3	[-]
Interpolation factor for sensor behavior model (optional)					

Input data:

LAMB_DELTA_LAM_ADJ[NC_CBK_EX_NR]	LV_INH_LAM_KWP	DELTA_LAMB_SP_O2L_MDL[NC_CBK_EX_NR]	LAMB_SP[NC_CBK_EX_NR]
LV_ACT_SA_EOL	LV_ERR_CHG_LS_UP	LV_SA_END	N
LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_ERR_LSL_UP_IF[NC_CBK_EX_NR]	LV_ERR_IGC	LV_HOM_ACT

LV_INH_LSCL_CUS[NC_CBK_EX_NR]	LV_LAMB_COP[NC_CBK_EX_NR]	LV_MIS_STATE_A	MAF
LV_REQ_ISC	MAF_CYL	LV_ST_END	MAF_DELTA_LDC
FAC_TQ_REQ_DELTA_LDC	LAMB_DELTA_SUM_LAM_ADJ_AD[NC_CBK_EX_NR]	N_DELTA_LDC	LAMB_LS_UP[NC_CBK_EX_NR]
LV_IS	LAMB_SP_HOM[NC_CBK_EX_NR]	STATE_LS[NC_CBK_EX_NR]	WGPWM[NC_CBK_EX_NR]
LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	FAC_DIAG_DYN_LSL_UP[NC_CBK_EX_NR]	LV_LAM_AD_INJ_ACT

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAM_MAF_MIN_HYS	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Hysteresis for minimum MAF - threshold for lambda control					
C_LAMB_SP_AFR_THD	1	0...400H	0...1	0.9766e-3	[-]
rich air-fuel mixture setpoint threshold					
C_LAMB_SP_AFL_THD	1	400...1000H	1...4	0.9766e-3	[-]
lean air-fuel mixture setpoint threshold					
C_LAMB_SP_AFR_THD_HYS	1	0...400H	0...1	0.9766e-3	[-]
rich air-fuel mixture setpoint threshold hysteresis					
C_LAMB_SP_AFL_THD_HYS	1	0...400H	0...1	0.9766e-3	[-]
lean air-fuel mixture setpoint threshold hysteresis					
IP_LAM_MAF_MIN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_LAM_MAF_MIN	6	0...1FE0H	0...8160	1	[rpm]
Minimum MAF - threshold for lambda control					
LC_HOM_ACT_COMP	1	0...1H	0...1	1	[-]
switching variable for the open loop/closed loop operation in stratified mode					
LC_LAM_GAIN_SWI	1	0...1H	0...1	1	[-]
flag indicating whether lambda controller parameter switch is allowed					
LC_LAM_OFF_SAWUP	1	0...1H	0...1	1	[-]
switching variable for the open loop/closed loop operation at secondary air					
LC_LAMB_COP_OFF	1	0...1H	0...1	1	[-]
switch to disable the cat-heating-"lambda control deactivation condition"					
LC_FAC_IPT_SENS_MDL_OPT_ACT	1	0...1H	0...1	1	[-]
switch to enable calculation of aged lin. O2 sensor signal correction (for monitoring only					
C_N_DYW_LAM_NOT_STAT	1	0...1FE0H	0...8160	1	[rpm]
engine speed window for detection of non stationary condition					
C_MAF_DYW_LAM_NOT_STAT	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
air mass flow window for detection of non stationary condition					
C_MAF_INT_LAM_NOT_STAT	1	0...FFFFH	0...1820.41666	0.0277778	[g]
air mass flow integral for detection of non stationary condition					
C_LAMB_DIF_AV_CTL_ST_MAX	1	F800...800H	-0.125...0.125	0.061e-3	[-]
max. allowable lambda deviation at initialisation of lambda difference for the lambda controller					
C_LAMB_DIF_AV_CTL_ST_MAX_IS	1	F800...800H	-0.125...0.125	0.061e-3	[-]
max. allowable lambda deviation at initialisation of lambda difference for the lambda controller in IS					
C_LAMB_DIF_AV_CTL_ST_MIN	1	F800...800H	-0.125...0.125	0.061e-3	[-]
min. allowable lambda deviation at initialisation of lambda difference for the lambda controller					
C_LAMB_DIF_AV_CTL_ST_MIN_IS	1	F800...800H	-0.125...0.125	0.061e-3	[-]
min. allowable lambda deviation at initialisation of lambda difference for the lambda controller in IS					
C_LGRD_LAMB_DIF_AV_CTL_ST	1	F800...800H	-0.125...0.125	0.061e-3	[-]
limitation gradient of lambda deviation at initialisation of lambda difference for the lambda controller					
C_FAC_TQ_REQ_DYW_LAM_NOT_STAT	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
torque scaling factor window for detection for non stationary condition					
ID_FAC_IPT_SENS_MDL	6	0...400H	0...1	0.9766e-3	[-]
LDP_WGPWM_ID_FAC_IPT_SENS_MDL	6	0...FFFFH	0...99.99847	1.5259e-3	[%]
Interpolation factor for waste gate position					
IP_FAC_IPT_SENS_MDL_OPT	4	0...400H	0...1	0.9766e-3	[-]

LDP_FAC_DYN_IP_FAC_IPT_SENS_MDL	4	0...400H	0...1	0.9766e-3	[-]
Interpolation factor for WRAF sensor dynamic					
LC_LAM_GAIN_SWI_LAM_AD_INJ	1	0...1H	0...1	1	[-]
Flag indicating whether lambda controller parameter switch is allowed for minimum fuel mass adaptation					

FUNCTION DESCRIPTION:

General information:

This module defines the project specific activation conditions and necessary output variables for the linear lambda controller.
The calculation shall be done for all exhaust cylinder banks.

For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2,

otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation: at ECU reset:

LV_INH_LSCL[i] = 1
LV_LAM_GAIN_SWI = LC_LAM_GAIN_SWI
LAMB_SP_DELTA_LAM[i] = 0
LAMB_AV_DELTA_LAM[i] = 0
LV_LAM_STOP_REQ[i] = 0
LV_STATE_LS_OPL_ERR[i] = 0
LV_LAMB_SP_AFR_THD[i] = 0
LV_LAMB_SP_AFL_THD[i] = 0
LV_LAM_MAF_MIN = 0

MFF_ADD_WF = 0
N_OFS_LAM_NOT_STAT = 0
MAF_OFS_LAM_NOT_STAT = 0
MAF_INT_LAM_NOT_STAT = 0
LAMB_DIF_AV_CTL_ST_OUT[i] = 0
LV_CLC_LAMB_DIF_AV_ST = 1
LV_LAM_STOP_SHO_PER_REQ[i] = 0
FAC_TQ_REQ_OFS_LAM_NOT_STAT = 0
MAF_INT_S_ACT = 0

At LV_IGK = 0 -> 1

LAMB_DIF_AV_CTL_ST_OUT[i] = 0
LV_CLC_LAMB_DIF_AV_ST = 1
LV_MAF_INT_MIN_LAM_REAC_SWI = 0

Recurrence: T_SAMPLE = 20 ms

Activation: LV_ST_END = 1

Deactivation: -

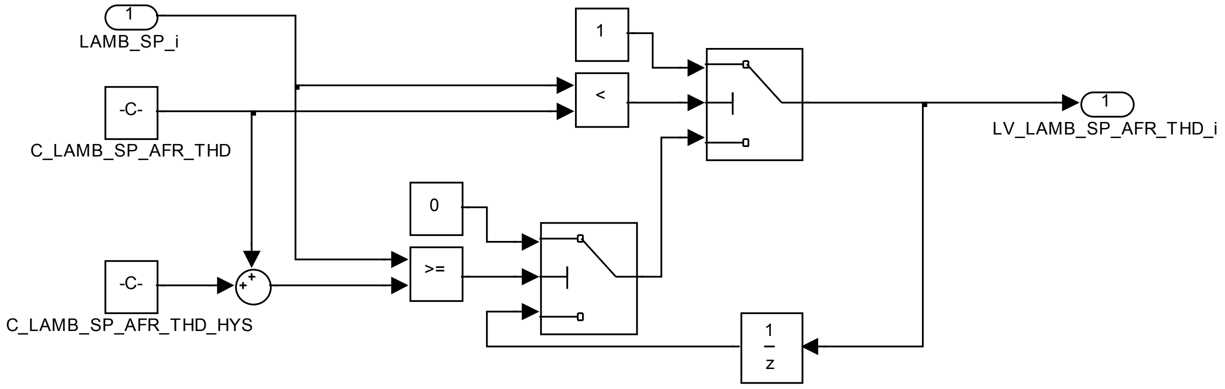
Formula section:

46.6.1 Calculation of LV_INH_LSCL[i]

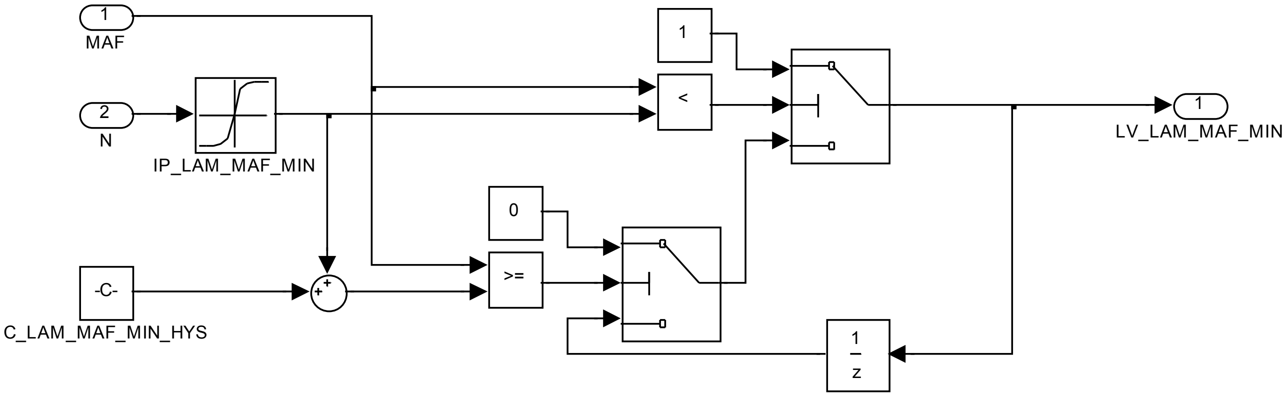
46.6.1.1 Inhibition of the lambda feedback control circuit

```
If(1)          LAMB_SP[i] < C_LAMB_SP_AFR_THD
        % lambda setpoint value is less than the rich air-fuel mixture threshold
Then(1)        LV_LAMB_SP_AFR_THD[i] = 1
Else(1)
        If(2)    LAMB_SP[i] ≥ C_LAMB_SP_AFR_THD + C_LAMB_SP_AFR_THD_HYS
        Then(2)  LV_LAMB_SP_AFR_THD[i]= 0
        Endif(2)
Endif(1)
If(1)          LAMB_SP[i] > C_LAMB_SP_AFL_THD  lambda setpoint value is greater than
                                               the lean air-fuel mixture threshold
Then(1)        LV_LAMB_SP_AFL_THD[i]  =  1
Else(1)
        If(2)    LAMB_SP[i] ≤ C_LAMB_SP_AFL_THD - C_LAMB_SP_AFL_THD_HYS
        Then(2)  LV_LAMB_SP_AFL_THD[i]  =  0
        Endif(2)
Endif(1)
```

(signal flow is analogous LV_LAMB_SP_AFR_THD[i] , see above)



```
If(1)       MAF < IP_LAM_MAF_MIN
Then(1)     LV_LAM_MAF_MIN = 1
Else(1)
    If(2)   MAF ≥ IP_LAM_MAF_MIN + C_LAM_MAF_MIN_HYS
    Then(2) LV_LAM_MAF_MIN = 0
    Endif(2)
Endif(1)
```



46.6.1.2 Inhibition due to present error

```
If      LV_ERR_IGC              = 1   or
        LV_ERR_LSL_UP_IF[i]     = 1   or
        LV_ERR_OFS_LSL_UP[i]    = 1   or
        LV_ERR_CTL_LSL_UP[i]    = 1   or
        LV_ERR_LS_UP[i]         = 1   or
        LV_ERR_CHG_LS_UP        = 1   or
        LV_MIS_STATE_A          = 1   or
Then    LV_STATE_LS_OPL_ERR[i]  = 1
Else    LV_STATE_LS_OPL_ERR[i]  = 0
Endif
```

46.6.1.3 Summary of all inhibition conditions for activation of lambda controller

```
If      LV_STATE_LS_OPL_ERR[i]        = 1   or    (present error)
        LV_LAMB_SP_AFR_THD[i]         = 1   or    (lambda setpoint to rich)
        LV_LAMB_SP_AFL_THD[i]         = 1         or    (lambda setpoint to lean)
        LV_LAM_MAF_MIN                = 1   or    (minimum maf not fulfilled)
        ( LC_LAMB_COP_OFF             = 1   and
        LV_LAMB_COP[i]                = 1)  or    (catalyst overheat prevention)
        LV_ACT_SA_EOL                 = 1   or    (EOL test SA is active)
        (LC_LAM_OFF_SAWUP=1and
        LV_SA_END                     = 0)  or    (secondary air is active)
        LV_INH_LAM_KWP                = 1         (disable Controller via Tester)
Then    LV_INH_LSCL[i] =  1
Else    LV_INH_LSCL[i] =  0
Endif
```

46.6.2 Calculation of LV_LAM_STOP_REQ[i]

The request to stop the lambda-controller (LV_LAM_STOP_REQ[i] = 1), if one of the conditions

a) stratified charge mode (LV_HOM_ACT=0) **and** LC_HOM_ACT_COMP = 0 **or**

b) secondary air is active (LV_SA_END = 0) **and** LC_LAM_OFF_SAWUP = 1

is fulfilled.

```
If      (LV_HOM_ACT            = 0     and
        LC_HOM_ACT_COMP        = 0)    or
        LV_INH_LSCL_CUS[i]     = 1     or
        LV_MIS_STATE_A         = 1     or
        (LV_SA_END             = 0     and
        LC_LAM_OFF_SAWUP       = 1)
Then    LV_LAM_STOP_REQ[i] = 1
Else    LV_LAM_STOP_REQ[i]= 0
Endif
```

46.6.3 Setting of LV_LAM_GAIN_SWI

The gain of the lambda controller is different between idle speed and part load. Therefore, the gain is switched in case of active idle speed controller.

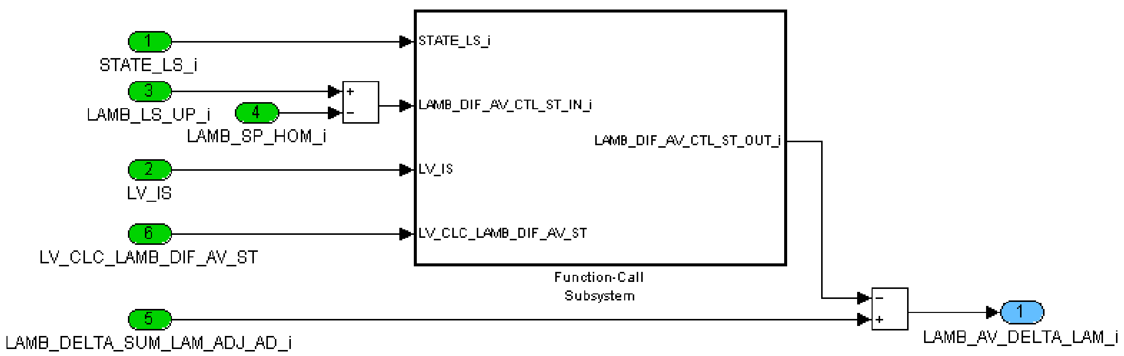
```
If      (LV_REQ_ISC = 1 and LC_LAM_GAIN_SWI = 1)
        or (LV_LAM_AD_INJ_ACT = 1 and LC_LAM_GAIN_SWI_LAM_AD_INJ = 1)
Then    LV_LAM_GAIN_SWI = 1
Else    LV_LAM_GAIN_SWI = 0
Endif
```

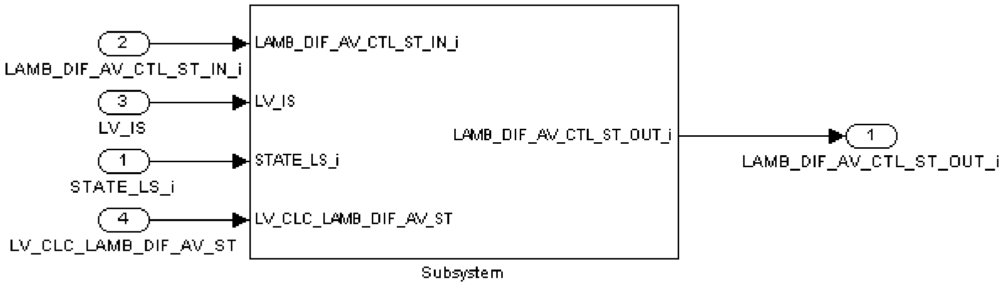
46.6.4 Assignment of LAMB_SP_DELTA_LAM[i]

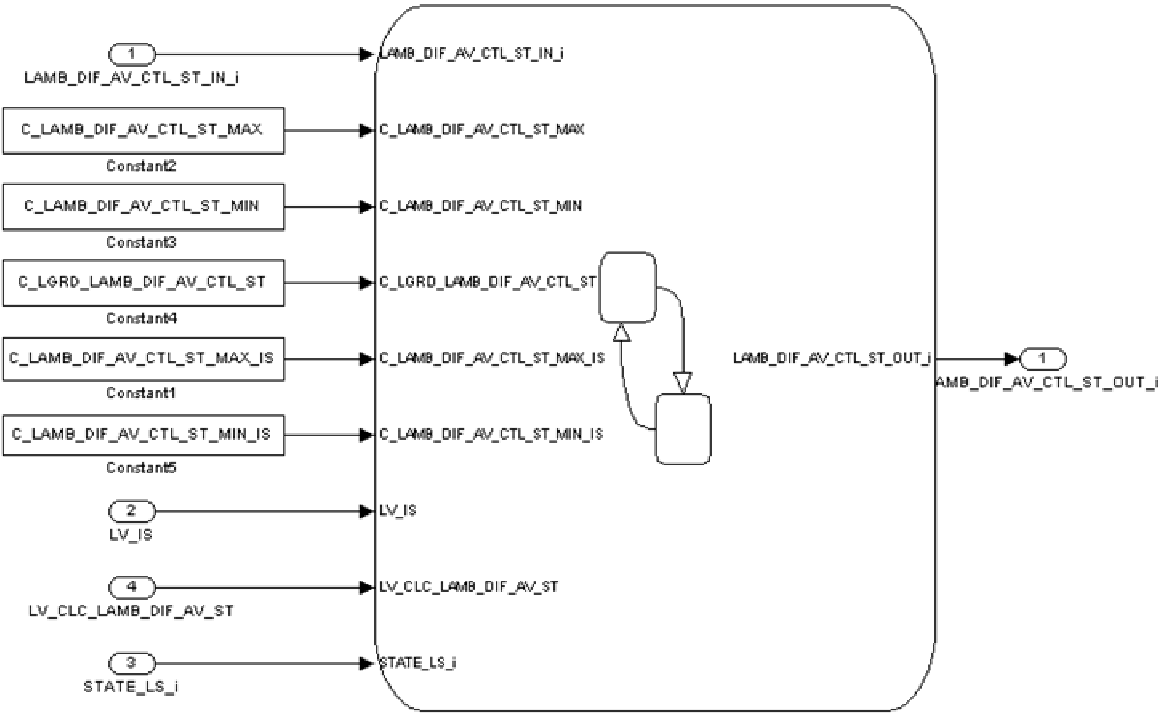
The output of the trim controller (P and I share) is distributed as follows as corrective intervention on the lambda set point and on the measured lambda signal:

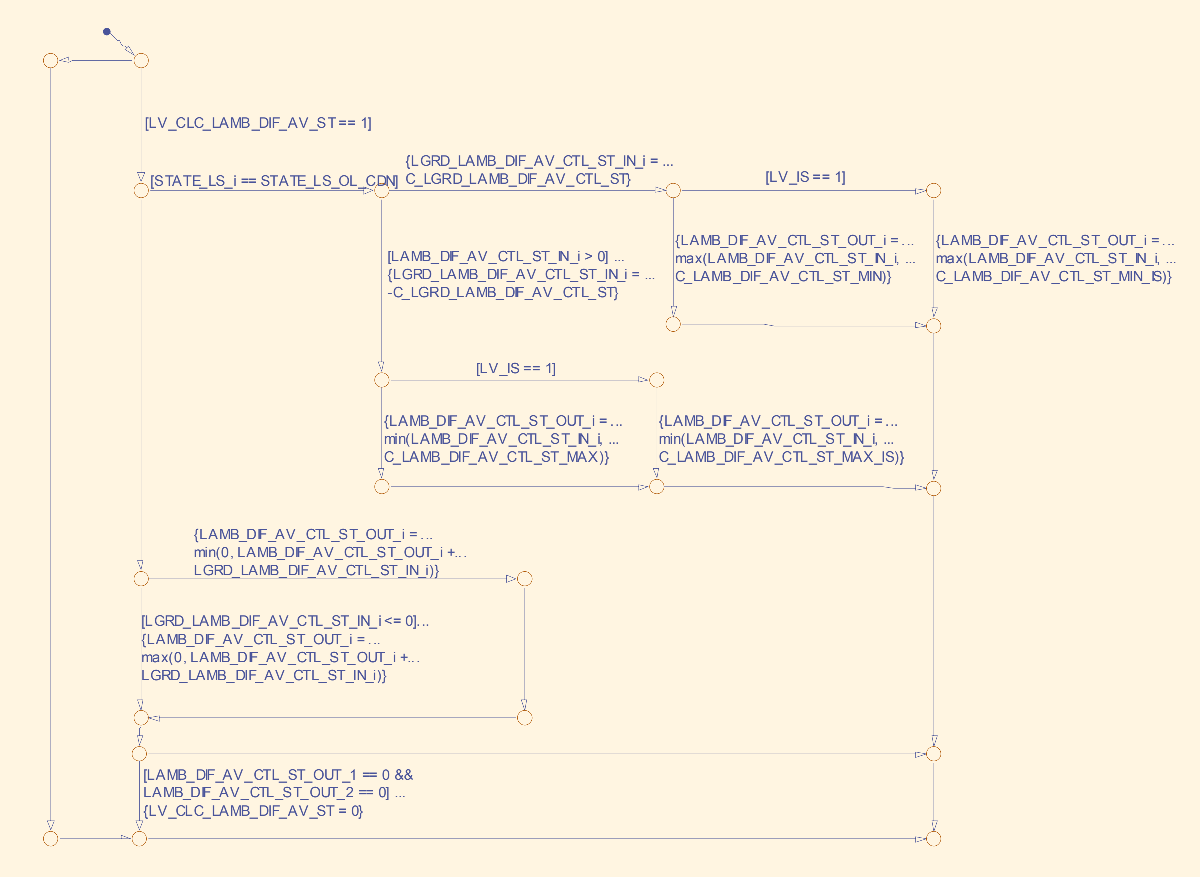
LAMB_SP_DELTA_LAM[i] = LAMB_DELTA_LAM_ADJ[i]
+ DELTA_LAMB_SP_O2L_MDL[i]

46.6.5 Assignement of LAMB_AV_DELTA_LAM[i]









46.6.6 Detection of non-stationary condition

A limited dynamic condition based on N, MAF and PV_AV is detected when the variation of those variables exceed the calibrateable threshold.

```
IF (ABS(N_DELTA_LDC) > C_N_DYW_LAM_NOT_STAT)
THEN
   IF (ABS(N_DELTA_LDC - N_OFS_LAM_NOT_STAT) > C_N_DYW_LAM_NOT_STAT)
   THEN
      N_OFS_LAM_NOT_STAT = N_DELTA_LDC
      MAF_INT_LAM_NOT_STAT = 0
   ENDIF
ELSE
   N_OFS_LAM_NOT_STAT = 0
ENDIF

IF (ABS(MAF_DELTA_LDC) > C_MAF_DYW_LAM_NOT_STAT)
THEN
   IF (ABS(MAF_DELTA_LDC - MAF_OFS_LAM_NOT_STAT) >
       C_MAF_DYW_LAM_NOT_STAT)
   THEN
      MAF_OFS_LAM_NOT_STAT = MAF_DELTA_LDC
      MAF_INT_LAM_NOT_STAT = 0
   ENDIF
ELSE
   MAF_OFS_LAM_NOT_STAT = 0
ENDIF

IF (ABS(FAC_TQ_REQ_DELTA_LDC) > C_FAC_TQ_REQ_DYW_LAM_NOT_STAT)
THEN
   IF (ABS(FAC_TQ_REQ_DELTA_LDC - FAC_TQ_REQ_OFS_LAM_NOT_STAT) >
       C_FAC_TQ_REQ_DYW_LAM_NOT_STAT)
   THEN
      FAC_TQ_REQ_OFS_LAM_NOT_STAT = FAC_TQ_REQ_DELTA_LDC
      MAF_INT_LAM_NOT_STAT = 0
   ENDIF
ELSE
   FAC_TQ_REQ_OFS_LAM_NOT_STAT = 0
ENDIF

IF (MAF_INT_LAM_NOT_STAT < C_MAF_INT_LAM_NOT_STAT)
THEN  % non stationary condition detected
   MAF_INT_LAM_NOT_STAT [g] = MAF_INT_LAM_NOT_STAT [g] +
                              MAF_CYL [kg/h] * T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]
   FAC_NEUT_NOT_STAT_CDN= 7FFFh
ELSE  % stationary condition
   FAC_NEUT_NOT_STAT_CDN= 0h
ENDIF
```

46.6.7 Assignment of interpolation factor

```
If LC_FAC_IPT_SENS_MDL_OPT_ACT = 1
then
        FAC_IPT_SENS_MDL[i] = 0
        FAC_IPT_SENS_MDL_OPT[i] = IP_FAC_IPT_SENS_MDL_OPT(FAC_DIAG_DYN_LSL_UP[i])
else
        FAC_IPT_SENS_MDL[i] = ID_FAC_IPT_SENS_MDL(WGPWM[i])
        FAC_IPT_SENS_MDL_OPT[i] = 0
endif
```

46.6.8 Calculation of air massflow integral during S/HOMS operation mode

```
If LV_HOM_ACT = 1 -> 0
Then  MAF_INT_S_ACT = 0
Elseif  LV_HOM_ACT = 0
Then MAF_INT_S_ACT = MAF_INT_S_ACT + MAF_CYL [kg/h] * T_SAMPLE [ms] * 1/3600
        [(g*h) / (kg*ms)]
Elseif LV_HOM_ACT = 1
Then  hold MAF_INT_S_ACT
Endif
```

46.7 Trim control for linear lambda control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_DELTA_LAM_ADJ[NC_CBK_EX_NR]	O/V	F800...800H	-0.125...0.125	6.10352E-5	-
total output (P share + I share) from trim control					
LV_LAM_ADJ_I_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
activation flag for calculation of trim controller I share					
VLS_DIF_LAM_ADJ[NC_CBK_EX_NR]	O/V	FC00...3FFH	-5...4.99511719	0.00488281	V
difference between set point and actual downstream LS signal					
LV_LAM_ADJ_P_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
activation flag for calculation of trim controller P share					
VLS_SP_LAM_ADJ[NC_CBK_EX_NR]	O/V	0...3FFH	0...4.995117	0.00488281	V
trim control set point					
LV_LAM_ADJ_AD_END[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating that trim control I share shift was executed					
LAMB_DELTA_I_MMV_LAM_ADJ[NC_CBK_EX_NR]	O/V	8000...7FFFH	-0.125 ... 0.12499619	3.8147E-6	-
I share moving mean value from trim control					
LAMB_DELTA_P_LAM_ADJ[NC_CBK_EX_NR]	O/V	F800...800H	-0.125...0.125	6.10352E-5	-
P share of from trim control					
LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]	O/V	F800...800H	-0.125...0.125	6.10352E-5	-
I share from trim control					
LV_LAM_ADJ_D_ACT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
activation flag for calculation of trim controller D share					
MAF_INT_LAM_ADJ_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral for trim control activation					
MAF_INT_MIN_LAM_ADJ_P_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...1.45633E+3	0.02222222	g
minimum air mass flow integral for trim control P share activation					
MAF_INT_MIN_LAM_ADJ_I_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...1.45633E+3	0.02222222	g
minimum air mass flow integral for trim control I share activation					
MAF_INT_MIN_LAM_ADJ_D_ACT[NC_CBK_EX_NR]	V	0...FFFFH	0...1.45633E+3	0.02222222	g
minimum air mass flow integral for trim control D share activation					
VLS_DIF_MMV_LAM_ADJ[NC_CBK_EX_NR]	V	8000...7FFFH	-5...4.99984741	1.52588E-4	V
moving mean value of difference between set point and actual downstream LS signal					
VLS_DIF_DELTA_LAM_ADJ[NC_CBK_EX_NR]	V	8000...7FFFH	-5...4.99984741	1.52588E-4	V
difference between downstream LS signal and moving mean value					
LAMB_DELTA_D_LAM_ADJ[NC_CBK_EX_NR]	V	F800...800H	-0.125...0.125	6.10352E-5	-
D share from trim control					
LAMB_DELTA_I_SAVE_LAM_ADJ[NC_CBK_EX_NR]	V	8000...7FFFH	-0.125 ... 0.12499619	3.8147E-6	-
saved I share from trim control (saved in case that catalyst diagnosis switched on)					
LV_LAM_ADJ_ACT[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
activation flag of dynamic fuel trim					

Input data:

LV_LAM_STOP[NC_CBK_EX_NR]	LV_LAM_STOP_SHO_PER[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MAX[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MIN[NC_CBK_EX_NR]
LV_LAM_LSCL[NC_CBK_EX_NR]	LV_INH_LAM_ADJ[NC_CBK_EX_NR]	LV_LAMB_DELTA_I_LAM_ADJ_DEAC[NC_CBK_EX_NR]	LV_LDC_LAM_ADJ[NC_CBK_EX_NR]
VLS_AV_LAM_ADJ[NC_CBK_EX_NR]	LV_LAM_ADJ_ACT_FAST_LAM_LSCL[NC_CBK_EX_NR]	LV_LAM_ADJ_AD_REQ[NC_CBK_EX_NR]	LAMB_DELTA_DIF_I_LAM_ADJ_AD[NC_CBK_EX_NR]
EFF_CAT_DIAG[NC_CBK_EX_NR]	LAMB_SP_HOM[NC_CBK_EX_NR]	LV_CAT_PURGE_ACT[NC_CBK_EX_NR]	LV_LAMB_OHP[NC_CBK_EX_NR]
LV_LAMB_PLS_ACT[NC_CBK_EX_NR]	LV_ST_END	LV_LAM_ADJ_CAT_DIAG[NC_CBK_EX_NR]	LV_LAM_ADJ_REQ_DYN_LSL_UP[NC_CBK_EX_NR]
LV_LS_DOWN_READY[NC_CBK_EX_NR]	MAF_CYL	MAF_HB	NC_CBK_EX_NR
N_32	TCO	TEMP_CAT_DYN_MDL[NC_CBK_EX_NR]	T_AST

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_LAMB_DELTA_I_MMV_LAM_ADJ	1	0...FFFFH	0...0.99998474	1.52588E-5	-
correlation constant for trim control I share moving mean value calculation for adaptation value					
C_EFF_CAT_DIAG_MAX_LAM_ADJ_ACT	1	0...FFH	0...1.9921875	0.0078125	-
catalyst diagnosis value threshold for trim control activation					
C_LAMB_DELTA_I_MAX_LAM_ADJ	1	F800...800H	-0.125...0.125	6.10352E-5	-
upper limit of trim control I share					
C_LAMB_DELTA_I_MIN_LAM_ADJ	1	F800...800H	-0.125...0.125	6.10352E-5	-
bottom limit of trim control I share					
C_LAMB_SP_MAX_LAM_ADJ_ACT	1	0...7FFFH	0...1.999939	6.10352E-5	-
maximum lambda set point for trim control activation					
C_LAMB_SP_MIN_LAM_ADJ_ACT	1	0...7FFFH	0...1.999939	6.10352E-5	-
minimum lambda set point for trim control activation					
C_MAF_INT_LAM_ADJ_D_ACT_FAST[NC_CBK_EX_NR]	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for fast trim control D share reactivation					
C_MAF_INT_LAM_ADJ_D_ACT_PURGE[NC_CBK_EX_NR]	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for trim control D share reactivation after cat purge					
C_MAF_INT_LAM_ADJ_D_ACT_SLOW	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for slow trim control D share reactivation					
C_MAF_INT_LAM_ADJ_I_ACT_FAST[NC_CBK_EX_NR]	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for fast trim control I share reactivation					
C_MAF_INT_LAM_ADJ_I_ACT_PURGE[NC_CBK_EX_NR]	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for trim control I share reactivation after cat purge					
C_MAF_INT_LAM_ADJ_I_ACT_SLOW	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for slow trim control I share reactivation					
C_MAF_INT_LAM_ADJ_P_ACT_FAST[NC_CBK_EX_NR]	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for fast trim control P share reactivation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MAF_INT_LAM_ADJ_P_ACT_PURGE[NC_CBK_EX_NR]	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for trim control P share reactivation after cat purge					
C_MAF_INT_LAM_ADJ_P_ACT_SLOW	1	0...FFFFH	0...1.45633E+3	0.02222222	g
air mass flow integral threshold for slow trim control P share reactivation					
C_MAF_MAX_LAM_ADJ_D_ACT	1	0...FFH	0...1.389E+3	5.44705882	mg/stk
maximum air mass flow threshold for trim control D share calculation					
C_MAF_MAX_LAM_ADJ_P_ACT	1	0...FFH	0...1.389E+3	5.44705882	mg/stk
maximum air mass flow threshold for trim control P share calculation					
C_MAF_MIN_LAM_ADJ_D_ACT	1	0...FFH	0...1.389E+3	5.44705882	mg/stk
minimum air mass flow threshold for trim control D share calculation					
C_MAF_MIN_LAM_ADJ_P_ACT	1	0...FFH	0...1.389E+3	5.44705882	mg/stk
minimum air mass flow threshold for trim control P share calculation					
C_N_MAX_LAM_ADJ_D_ACT	1	0...FFH	0...8.16E+3	32	rpm
maximum engine speed threshold for trim control D share calculation					
C_N_MAX_LAM_ADJ_P_ACT	1	0...FFH	0...8.16E+3	32	rpm
maximum engine speed threshold for trim control P share calculation					
C_N_MIN_LAM_ADJ_D_ACT	1	0...FFH	0...8.16E+3	32	rpm
minimum engine speed threshold for trim control D share calculation					
C_N_MIN_LAM_ADJ_P_ACT	1	0...FFH	0...8.16E+3	32	rpm
minimum engine speed threshold for trim control P share calculation					
C_TCO_LAM_ADJ_P_ACT_HST	1	0...FEH	-48...142.5	0.75	°C
TCO threshold for hot start detection to immediately active trim control P share					
C_TEMP_CAT_MIN_LAM_ADJ_ACT	1	0...7FFFH	-273.15 ... 1.77479E+3	0.0625	°C
minimum cat temperature threshold for trim control activation					
C_T_AST_LAM_ADJ_P_ACT_HST	1	0...FFFFH	0...6.5535E+3	0.1	s
T_AST threshold for hot start detection to immediately active trim control P share					
LC_LAMB_PLS_NOT_CHK_LAM_ADJ	1	0...1H	0...1	1	-
calibration flag to choose whether I share is calculated when forced lambda stimulation is not active					
LC_LAM_ADJ_I_SAVE_CAT_DIAG	1	0...1H	0...1	1	-
flag indicating that I share after cat diagnosis is set to the value before cat diagnosis					
LC_LAM_ADJ_LDC_NOT_CHK	1	0...1H	0...1	1	-
calibration flag to choose whether I share is calculated when limited dynamics are not fulfilled					
IP_CRLC_LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]	6	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_MAF_CYL_1_LACO	6	0...FFFFH	0...2.04797E+3	0.03125	kg/h
air mass flow dependent weighting coefficient for trim control I share calculation					
IP_FAC_ADD_LAMB_DELTA_D	8	0...FFH	0...0.99609375	0.00390625	-
LDPM_EFF_CAT_DIAG_1_LACO	8	0...FFH	0...1.9921875	0.0078125	-
additional weighting coefficient for trim control D share calculation					
IP_FAC_ADD_LAMB_DELTA_P	8	0...FFH	0...0.99609375	0.00390625	-
LDPM_EFF_CAT_DIAG_1_LACO	8	0...FFH	0...1.9921875	0.0078125	-
additional weighting coefficient for trim control P share calculation					
IP_FAC_LAMB_DELTA_D_LAM_ADJ[NC_CBK_EX_NR]	6	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_MAF_CYL_1_LACO	6	0...FFFFH	0...2.04797E+3	0.03125	kg/h
air mass flow dependent weighting coefficient for trim control D share calculation					
IP_FAC_LAMB_DELTA_P_LAM_ADJ[NC_CBK_EX_NR]	6	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_MAF_CYL_1_LACO	6	0...FFFFH	0...2.04797E+3	0.03125	kg/h
air mass flow dependent weighting coefficient for trim control P share calculation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_LAMB_DELTA_D_LAM_ADJ	16	0...1000H	-0.125...0.125	6.10352E-5	-
LDP_VLS_DIF_DELTA_IP_LAMB_D	16	0...7FFH	-5...4.99511719	0.00488281	V
characteristic line of trim control D share					
IP_LAMB_DELTA_I_LAM_ADJ	16	0...FFFFH	-0.125 ... 0.12499619	3.8147E-6	-
LDPM_VLS_DIF_LAM_ADJ_1_LACO	16	0...7FFH	-5...4.99511719	0.00488281	V
characteristic line of trim control I share					
IP_LAMB_DELTA_P_LAM_ADJ	16	0...1000H	-0.125...0.125	6.10352E-5	-
LDPM_VLS_DIF_LAM_ADJ_1_LACO	16	0...7FFH	-5...4.99511719	0.00488281	V
characteristic line of trim control P share					
IP_LAMB_DELTA_P_LAM_ADJ_DIAG	16	0...1000H	-0.125...0.125	6.10352E-5	-
LDPM_VLS_DIF_LAM_ADJ_2_LACO	16	0...7FFH	-5...4.995117	0.0048828	V
characteristic line of trim control P share in case of active cat diagnosis					
IP_MAF_MAX_LAM_ADJ_I_ACT	6	0...FFH	0...1.389E+3	5.44705882	mg/stk
LDPM_N_32_2_LACO	6	0...FFH	0...8.16E+3	32	rpm
maximum air mass flow threshold for trim control I share calculation					
IP_MAF_MIN_LAM_ADJ_I_ACT	6	0...FFH	0...1.389E+3	5.44705882	mg/stk
LDPM_N_32_2_LACO	6	0...FFH	0...8.16E+3	32	rpm
minimum air mass flow threshold for trim control I share calculation					
IP_VLS_SP_DELTA_LAM_ADJ	8	0...7FFH	-5...4.99511719	0.00488281	V
LDPM_EFF_CAT_DIAG_1_LACO	8	0...FFH	0...1.9921875	0.0078125	-
Additional offset of setpoint for trim control at not active catalyst diagnosis to take into account the catalyst aging					
IP_CRLC_VLS_DIF_MMV_LAM_ADJ	6x8	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_MAF_CYL_1_LACO	6	0...FFFFH	0...2.04797E+3	0.03125	kg/h
LDPM_EFF_CAT_DIAG_1_LACO	8	0...FFH	0...1.9921875	0.0078125	-
correlation constant to calculate moving mean value of trim controller difference					
IP_LAMB_DELTA_I_LAM_ADJ_DIAG	16x8	0...FFFFH	-0.125 ... 0.12499619	3.8147E-6	-
LDPM_VLS_DIF_LAM_ADJ_2_LACO	16	0...7FFH	-5...4.995117	0.0048828	V
LDPM_EFF_CAT_DIAG_1_LACO	8	0...FFH	0...1.9921875	0.0078125	-
characteristic line of trim control I share in case of active diagnosis function					
IP_VLS_SP_LAM_ADJ[NC_CBK_EX_NR]	8x8	0...3FFH	0...4.995117	0.00488281	V
LDP_MAF_HB_IP_VLS_SP_LAM_ADJ	8	0...FFH	0...1.389E+3	5.44705882	mg/stk
LDP_N_32_IP_VLS_SP_LAM_ADJ	8	0...FFH	0...8.16E+3	32	rpm
sensor voltage set point for trim control					

46.7.1 LACO_REQGNTRIM0

The trim controller causes a shift of the WRAF sensor signal (upstream of the catalyst) or/and a shift of the lambda set point (in opposite direction). The output components (P share, I share, the sum of both or the adaptation value) of the trim controller can be linked in the application incidence for the lambda controller to corresponding inputs of the lambda controller. A compensation of ageing effects of the WRAF sensor as well as a better observance of the catalyst window is reached because of these shifts.

The control value is composed of P, I and D shares. All are determined depending on the trim control sensor signal (binary signal). The difference of the sensor voltage from set point value is the basic characteristic. The set point value can be applied depending on the operating point, so that the dynamic lambda can be adjusted according to the operating range.

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

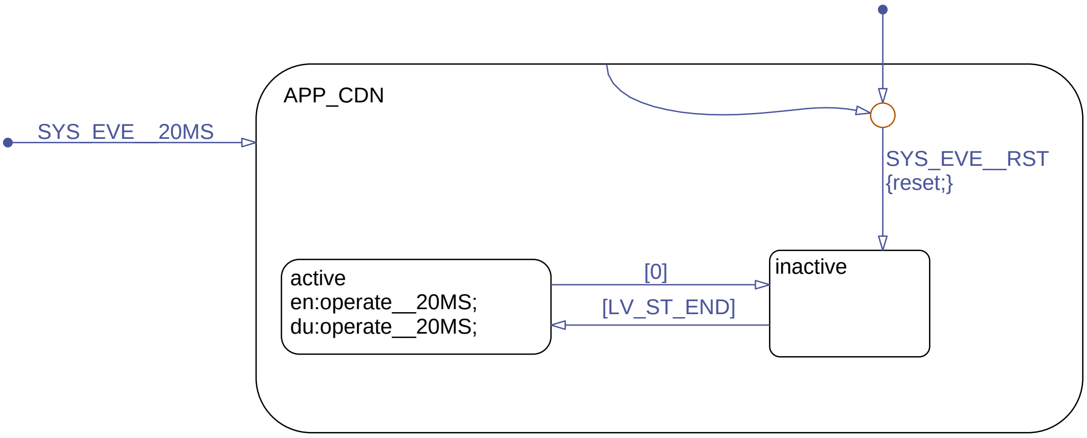


Figure 63 LACO_REQGNTRIM0/ APP_CDN/ Chart

Function Description

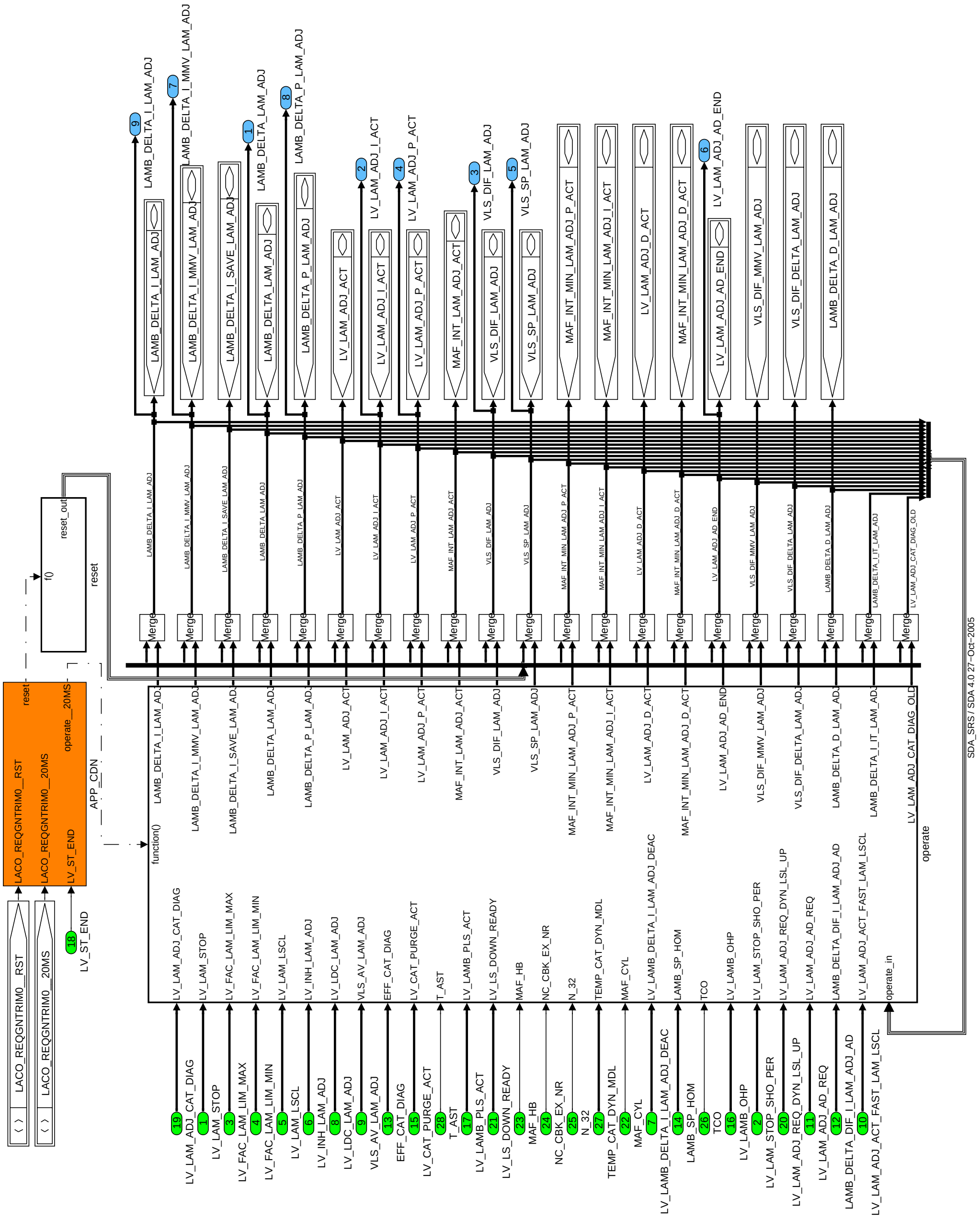


Figure 64 LACO_REQGNTRIM0

46.7.1.1 SUBFUNCTION: operate

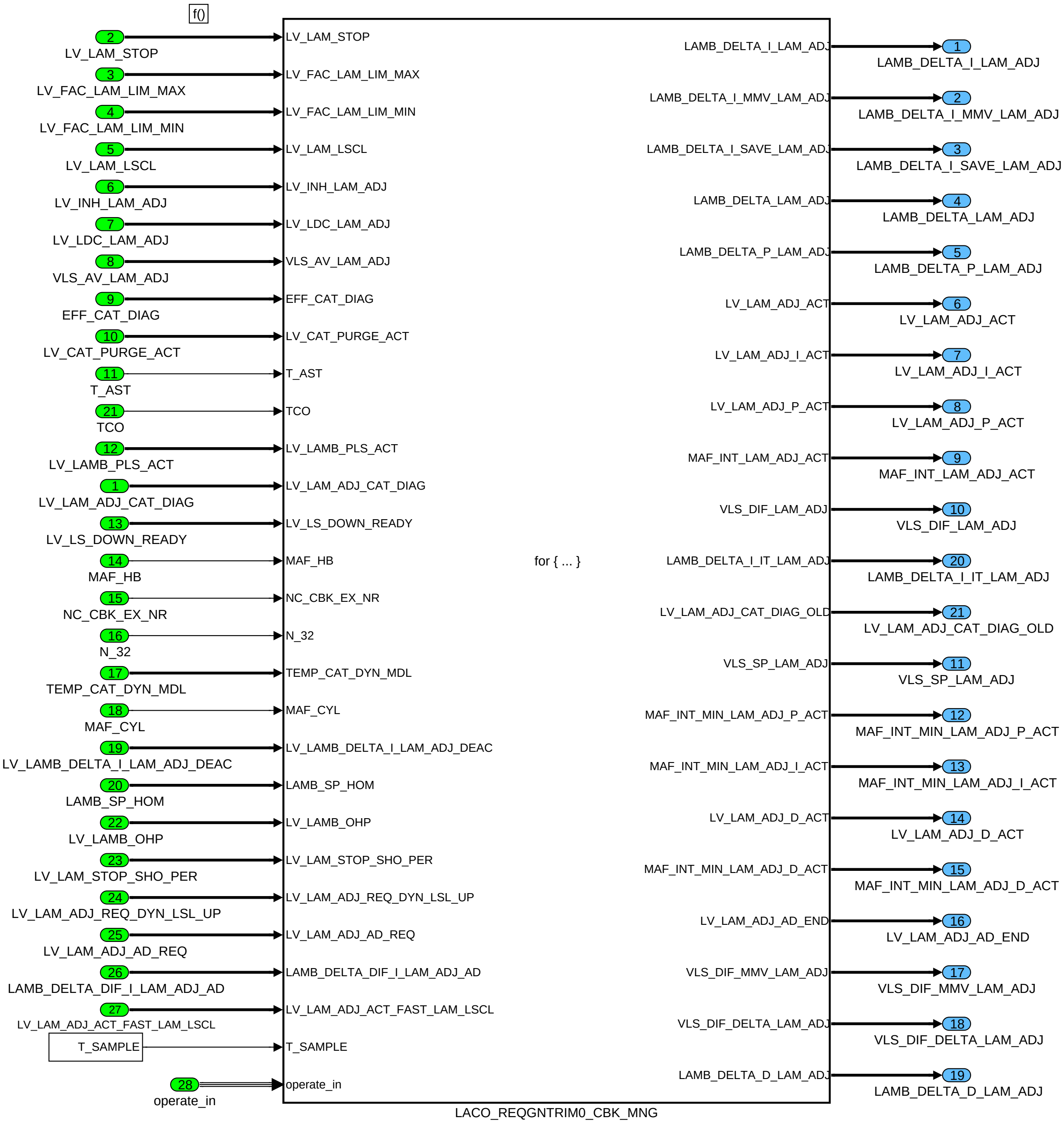


Figure 65 LACO_REQGNTRIM0/ operate

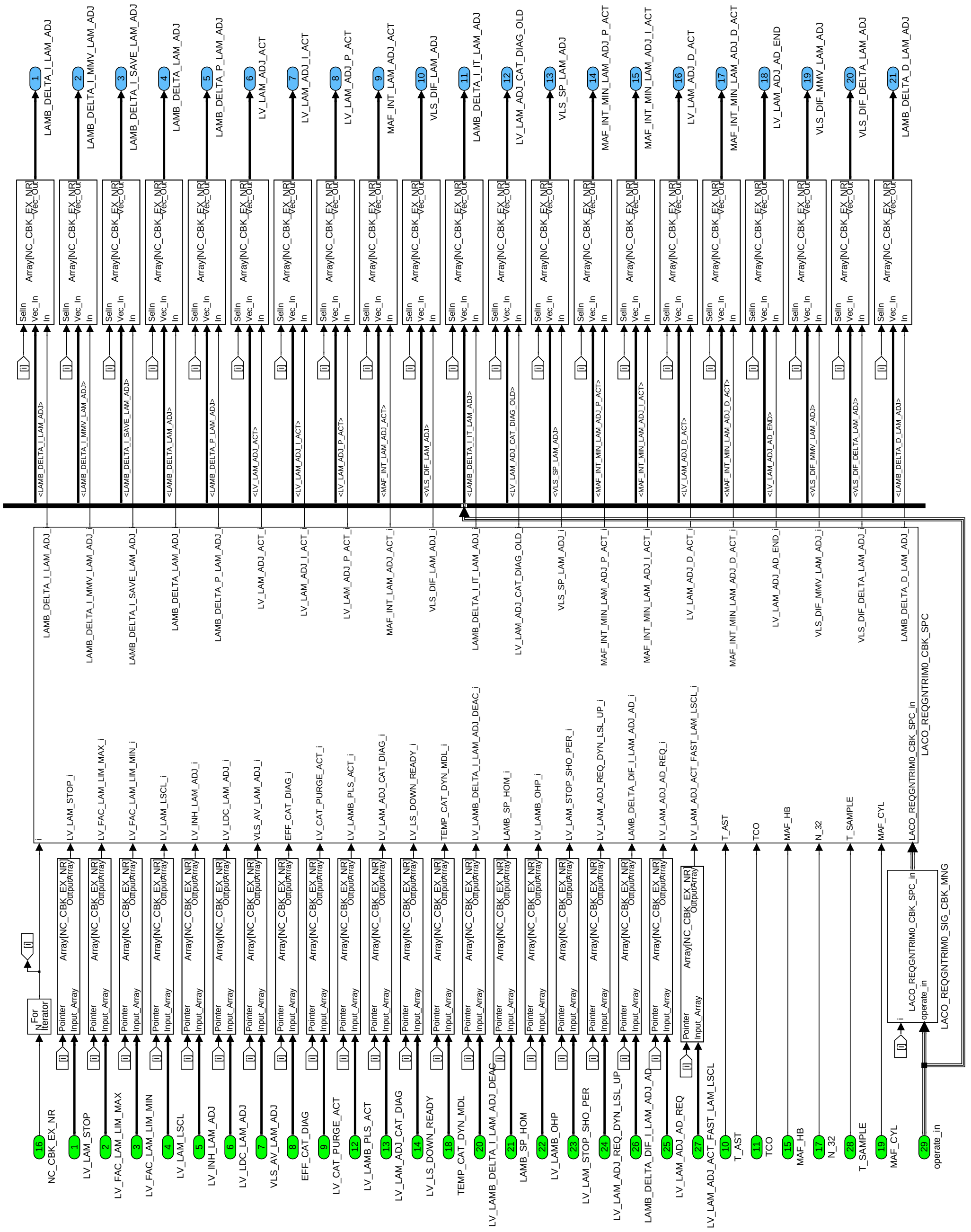


Figure 66 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG

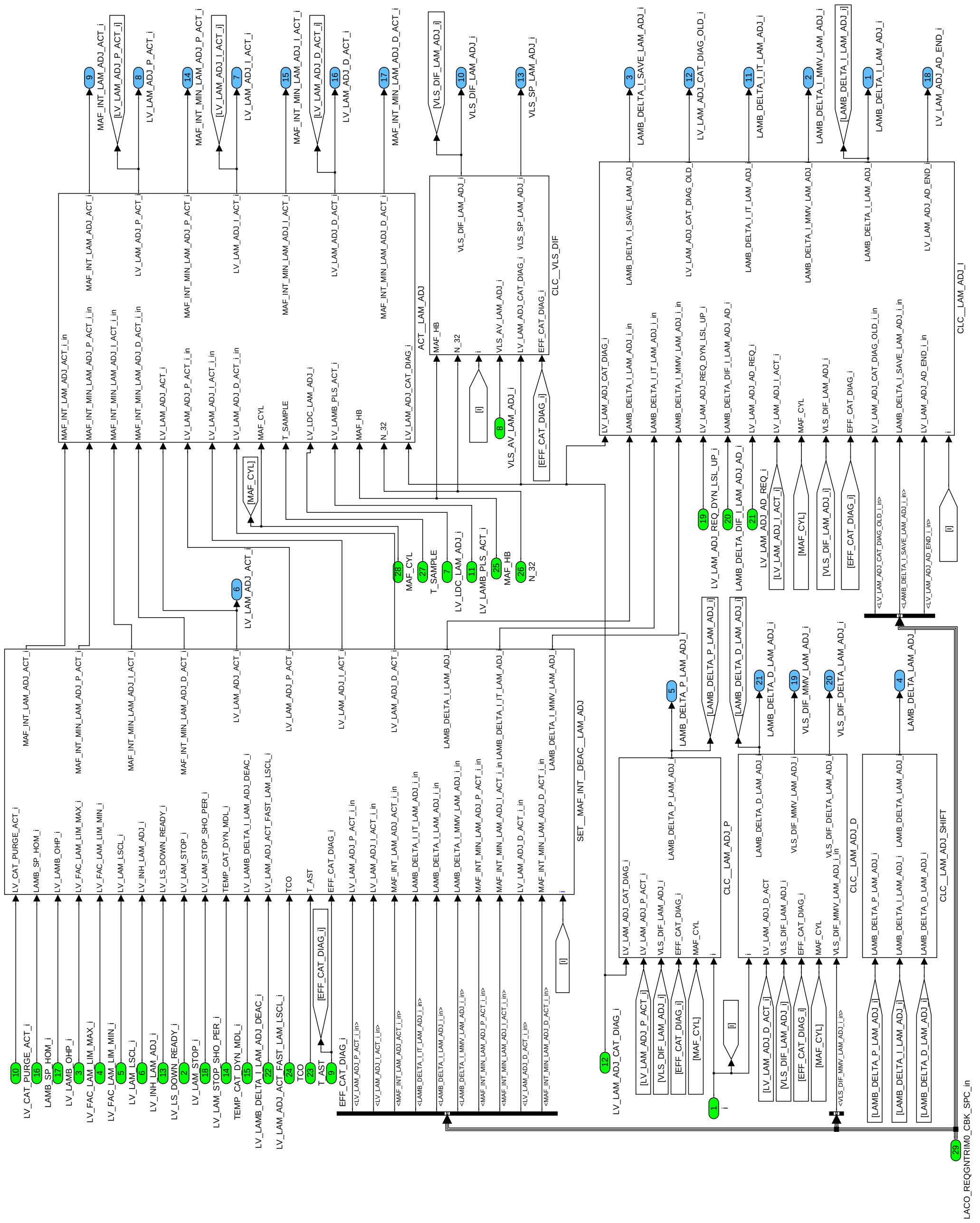


Figure 67 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC

Activation / deactivation of trim control (part I)

The trim controller is activated when the following conditions are fulfilled:

- no general inhibition
- the trim sensor is ready for operation
- the catalyst enrichment function is not active
- the lambda controller must be active
- the lambda controller must not be in short term in stop mode
- no output limitation of the lambda controller is active

- the catalyst is at operating temperature
- the catalyst efficiency is in the valid area
- the lambda set point must lie within calibration range
- no overheating protection with rich lambda is active

Before verifying the condition for P, I and/or D share activation, an air mass flow integral must exceed the threshold applied thresholds. These thresholds can be calibrated independently for P share, I share and D share activation. Depending on the deactivation reason different air mass flow integral thresholds can be applied. In case of hot start indicated by T_AST below limit and TCO above threshold the P share can be activated immediately if no other above mentioned activation condition is not fulfilled. The activation is realised by setting the MAF integral threshold for the P share to 0.

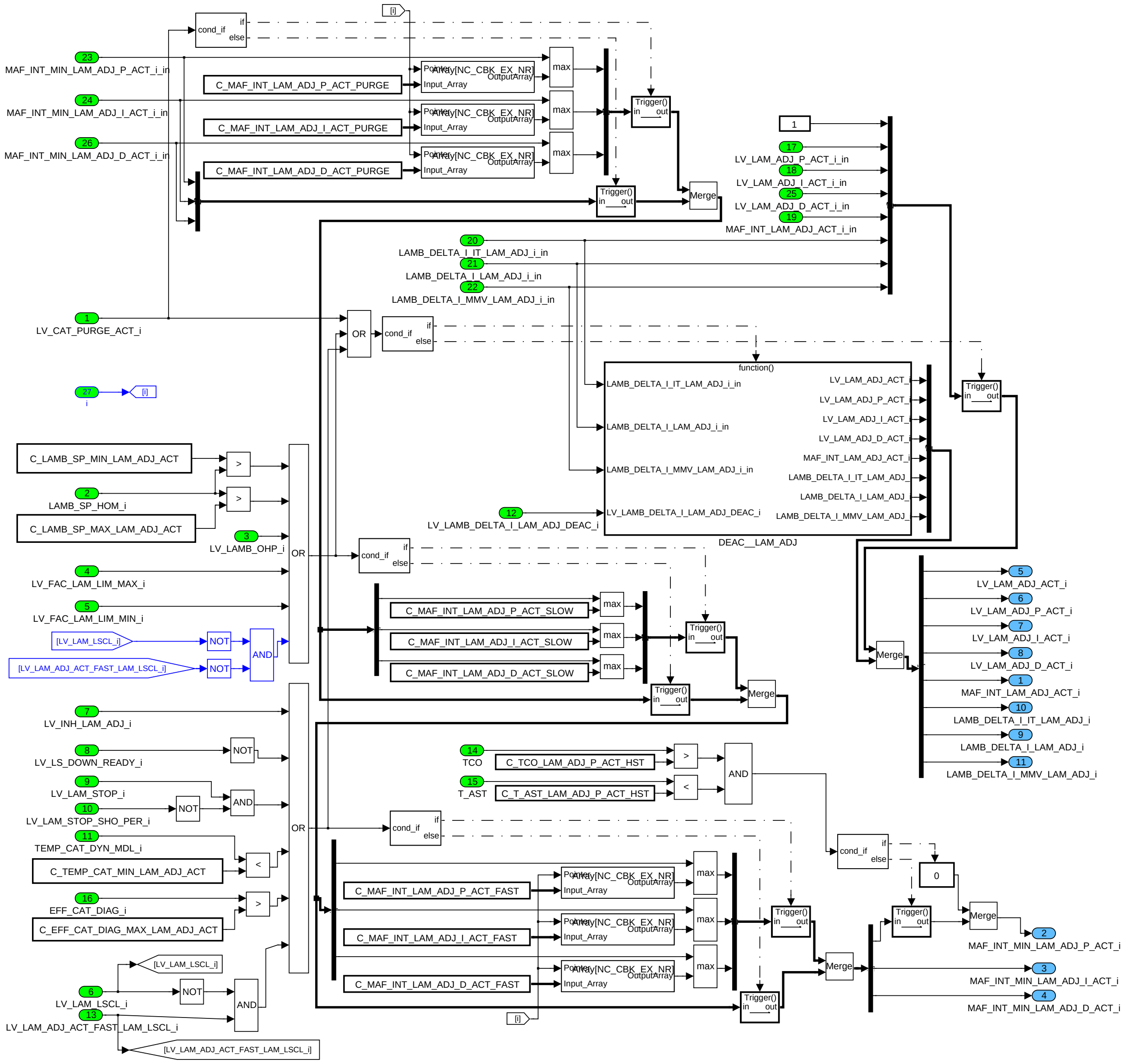


Figure 68 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ SET__MAF_INT__DEAC__LAM_ADJ

Deactivate trim controller

When deactivating the trim controller the P share and D share will get 0 and the I share remain fix (by deactivating the calualtion). If LV_LAMB_DELTA_I_LAM_ADJ_DEAC[i] was set to 0 (in the application incidences) then the I share is set to 0.

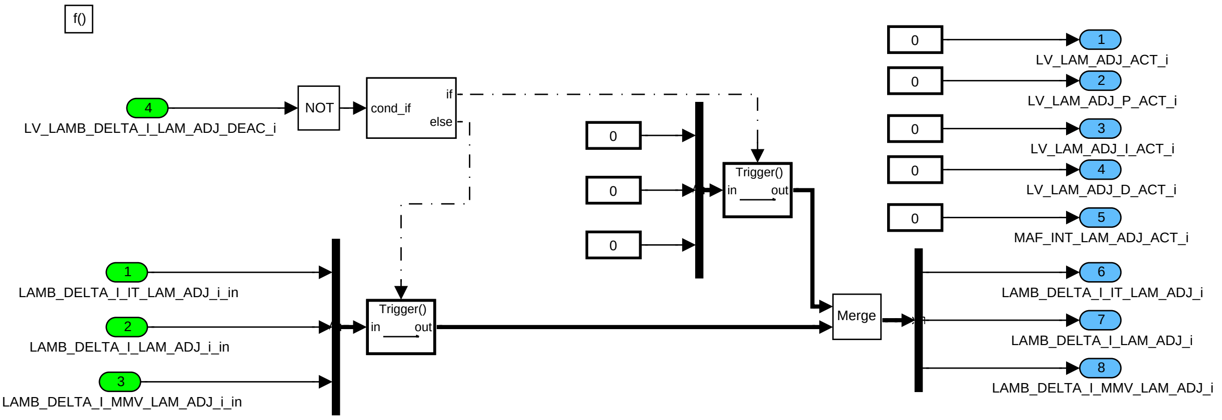


Figure 69 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ SET__MAF_INT__DEAC__LAM_ADJ/ DEAC__LAM_ADJ

Activation / deactivation of trim control (part II)

When all conditions to activate the trim controller are fulfilled an air mass integral is started and the conditions for P, I and D share are checked separately.

For the calcuation of the air mass flow integral the unit conversion:

[g] = [kg/h] * [ms] * [(g*h) / (kg*ms)] / 3600

has to be considered.

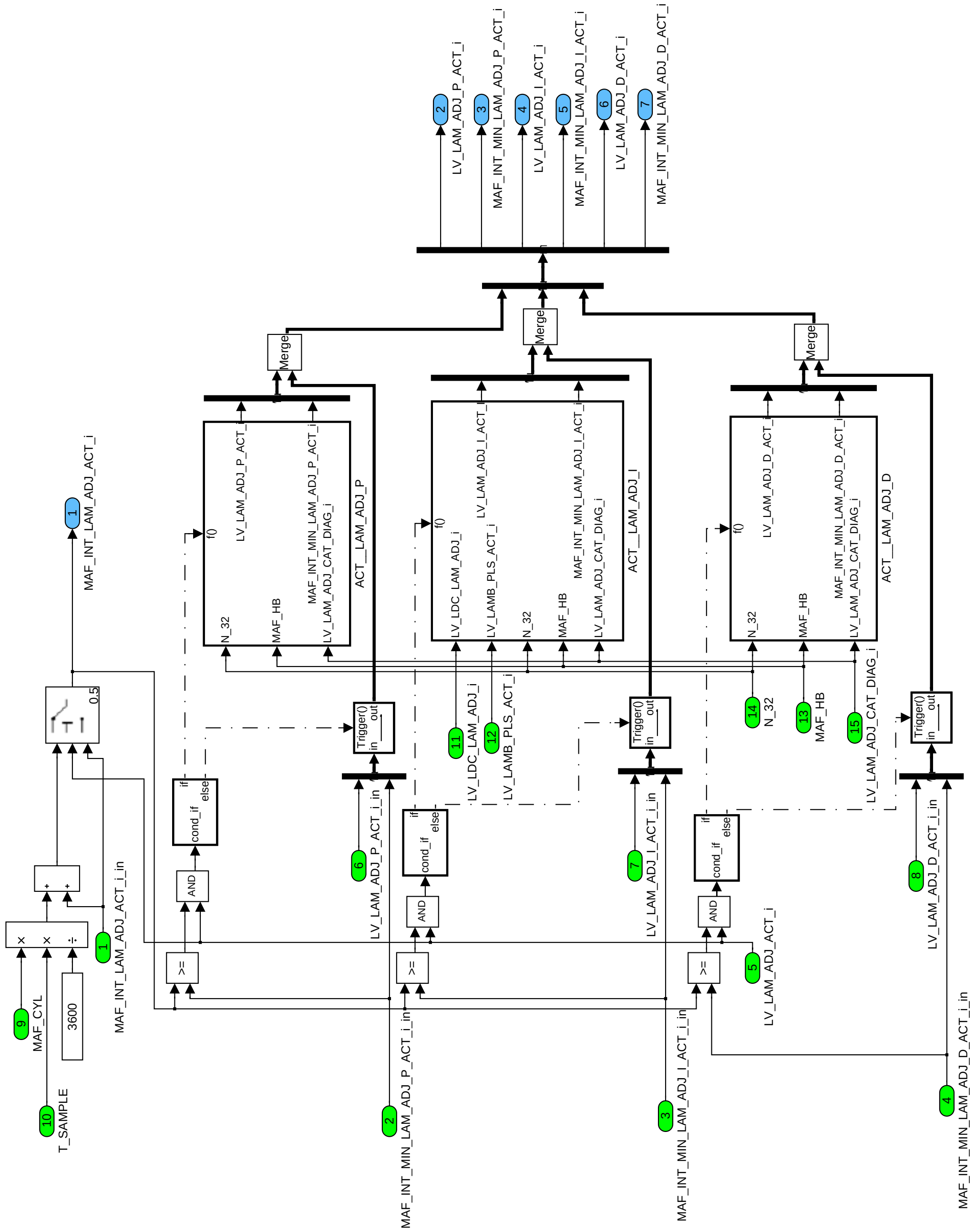


Figure 70 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ ACT__LAM_ADJ

Check D share activation conditions

The D share calculation gets active (indicated by LV_LAM_ADJ_D_ACT[i] – after the D share MAF integral threshold is exceeded - only in applied N and MAF areas and when the catalyst efficiency diagnosis is not active.

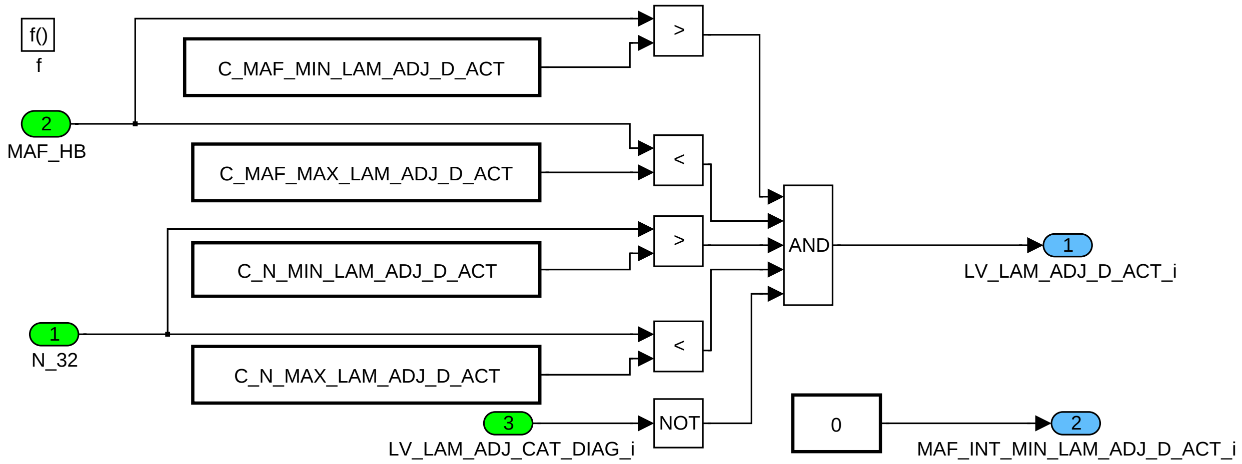


Figure 71 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ ACT__LAM_ADJ/ ACT__LAM_ADJ_D

Check I share activation conditions

The I share calculation gets active (indicated by LV_LAM_ADJ_I_ACT[i]) – after the I share MAF integral threshold is exceeded - in applied MAF areas when the limited dynamics conditions are fulfilled and when the forced lambda stimulation is active. Both latter conditions can be inhibited by means of calibration flags. During active cat efficiency diagnosis the I share is always active after exceeding the MAF integral threshold.

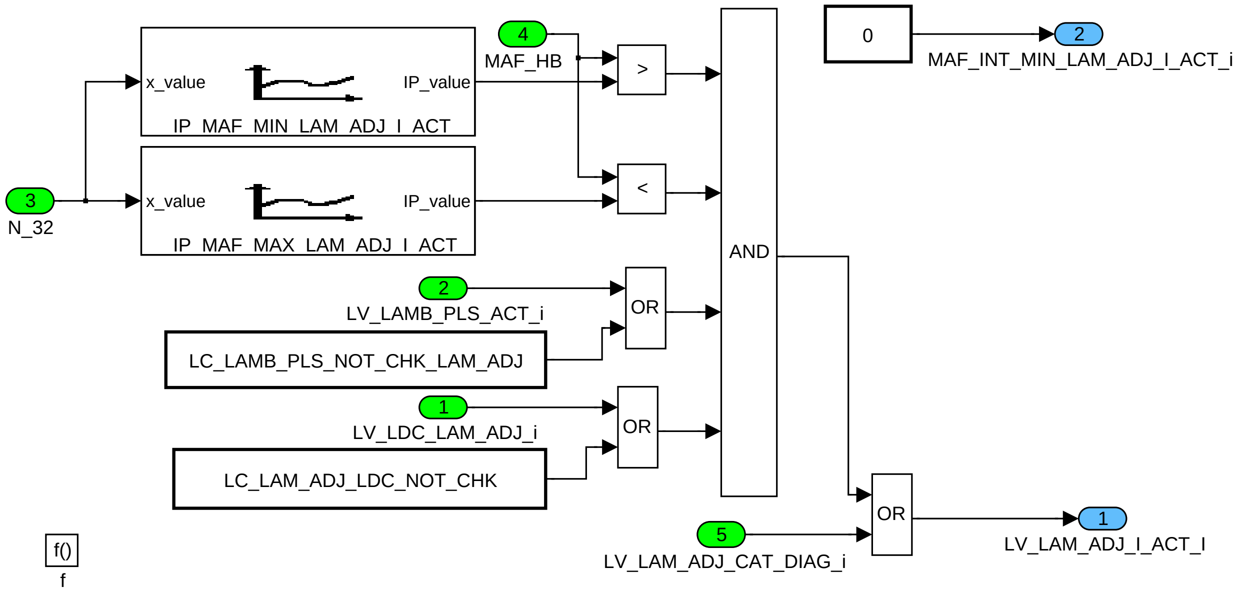


Figure 72 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ ACT__LAM_ADJ/ ACT__LAM_ADJ_I

Check P share activation conditions

The P share calculation gets active (indicated by LV_LAM_ADJ_P_ACT[i] – after the P share MAF integral threshold is exceeded - only in applied N and MAF areas or when catalyst efficiency diagnosis is active.

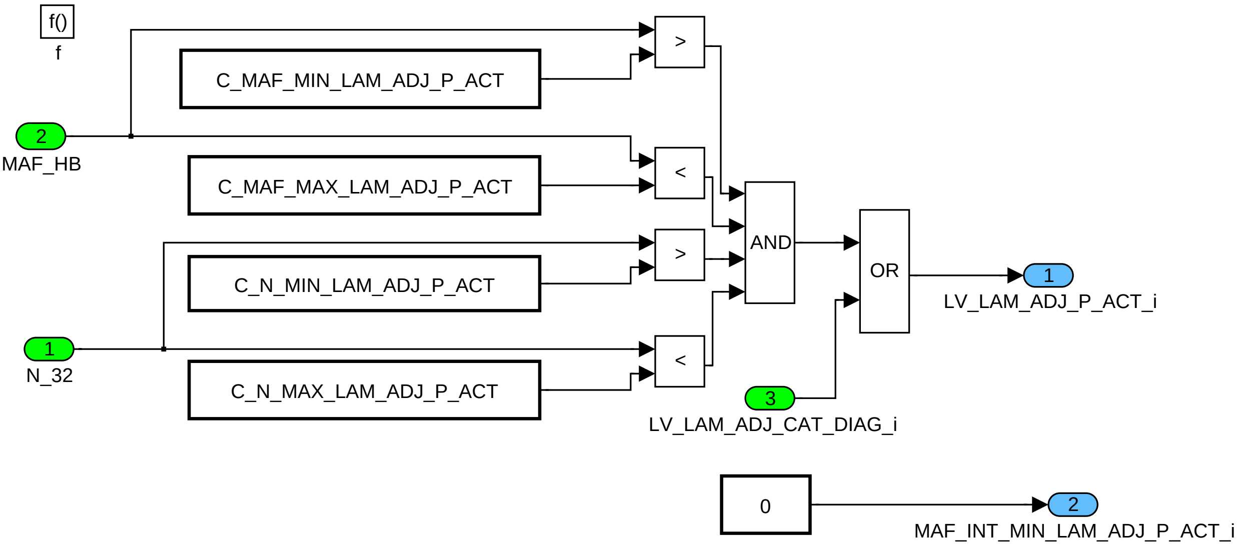


Figure 73 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ ACT__LAM_ADJ/ ACT__LAM_ADJ_P

Calculation of controller difference

The controller deviation is defined by the difference of the set point of the sensor output voltage IP_VLS_SP_LAM_ADJ[i] and the actual value VLS_AV_LAM_ADJ[i]. The setpoint can be shifted by IP_VLS_SP_DELTA_LAM_ADJ depending on catalyst ageing to take into account stringent NOx increase with catalyst ageing.

VLS_AV_LAM_ADJ[i] is calculated in the application incidences and can be directly the measured trim sensor signal or the filtered and shifted signal in case of Catalyst diagnosis (depending on project requirements). The actual value can also inlude an external adjustment.

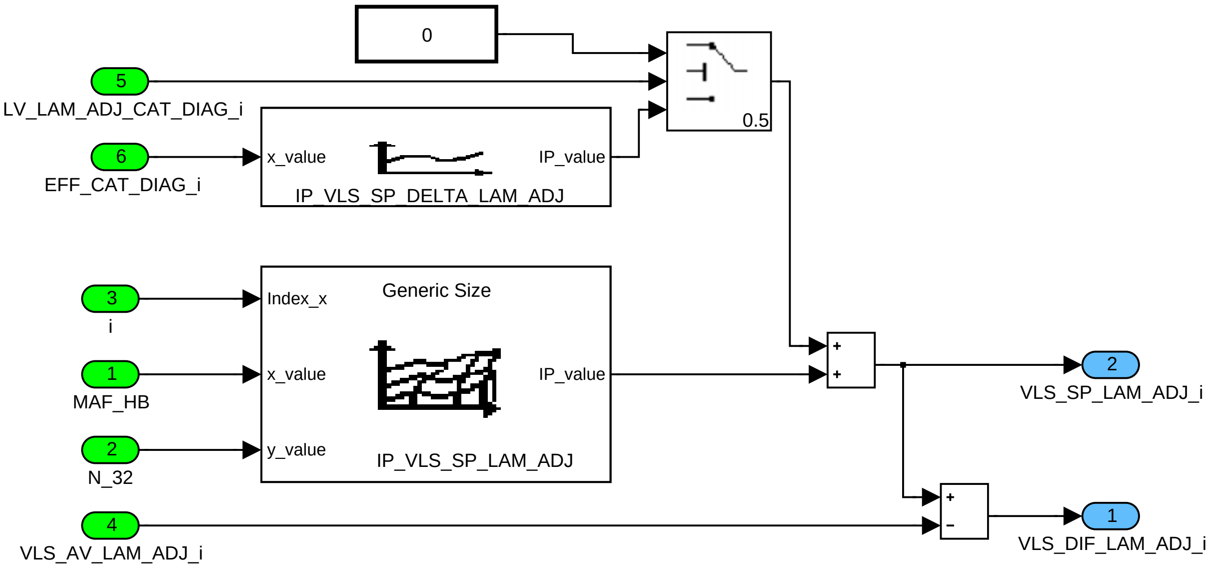


Figure 74 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ CLC__VLS_DIF

D share calculation

The D share is calculated when LV_LAM_ADJ_D_ACT[i] is 1. The D share is calculated by evaluating the voltage difference between the control difference and its moving mean value. The filter parameter for moving mean value is MAF_CYL and EFF_CAT_DIAG depending to take into account longer needed duration of D-share at low MAF_CYL and shorter due to OSC deterioring due to catalyst ageing This voltage difference is then input to a characteristic line IP_LAMB_DELTA_D_LAM_ADJ that is weighted by MAF_CYL and EFF_CAT_DIAG[i].

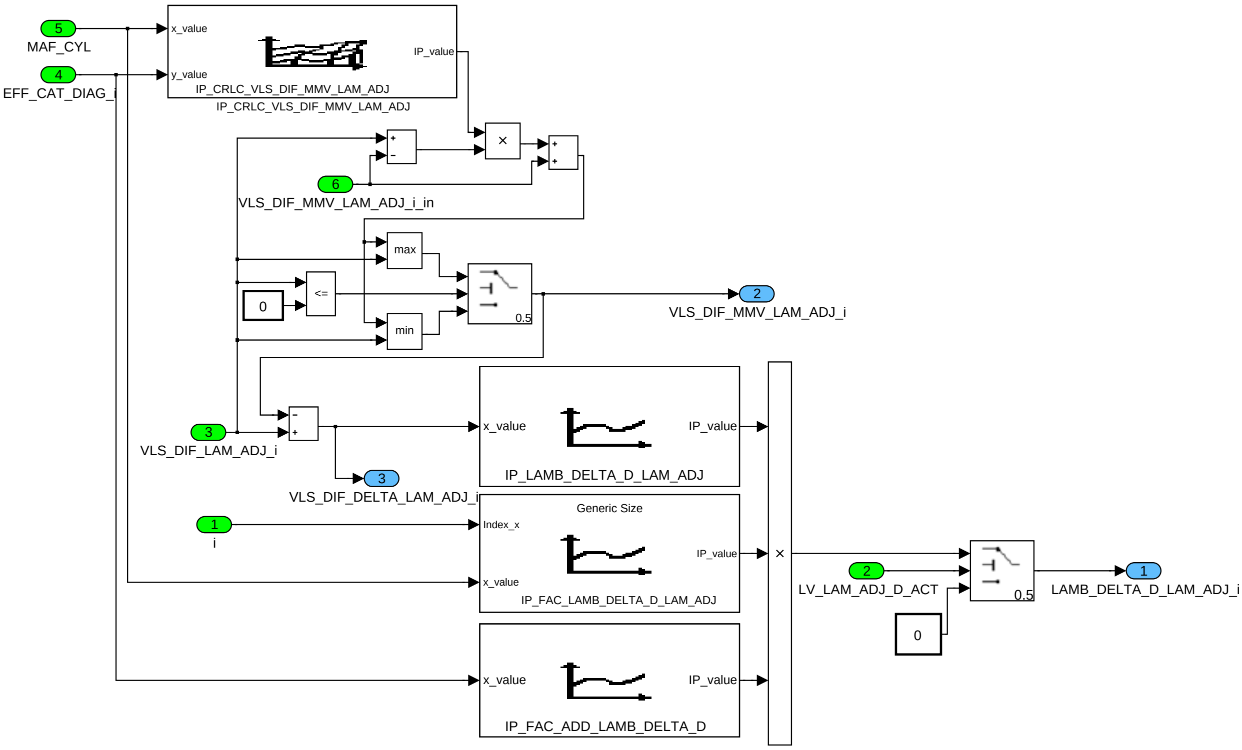


Figure 75 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ CLC__LAM_ADJ_D

I share calculation management

The I share itself is calculated internally with a higher resolution than the output variable LAMB_DELTA_I_LAM_ADJ[i]. The variable LAMB_DELTA_I_IT_LAM_ADJ[i] is used. A limitation of calibration thresholds (C_LAMB_DELTA_I_MIN_LAM_ADJ and C_LAMB_DELTA_I_MAX_LAM_ADJ) is applied at the end of the calculation. Both variables are initialised with the adaptation value LAMB_DELTA_AD_LAM_ADJ[i].

The I share is only calculated when LV_LAM_ADJ_I_ACT[i] is 1. The calculation is the summation of IP_LAMB_DELTA_I_LAM_ADJ * IP_CRLC_LAMB_DELTA_I_LAM_ADJ. This latter map is for MAF_CYL weighting. If the I share is not calculated it remains fix.

A moving mean value of the I share LAMB_DELTA_I_MMV_LAM_ADJ[i] (needed for WRAF sensor characterisitc line adaptation) is calculated.

In case of active catalyst diagnosis and sensor dynamic diagnosis own parameters are applied (IP_LAMB_DELTA_I_LAM_ADJ_DIAG multiplied by IP_CRLC_LAMB_DELTA_I_LAM_ADJ) and the moving mean value of the I share is not calculated. When cat diagnosis gets active the I share is stored. It is written back to the corresponding variable at the end of the diagnosis only in case that LC_LAM_ADJ_I_SAVE_CAT_DIAG is set to 1.

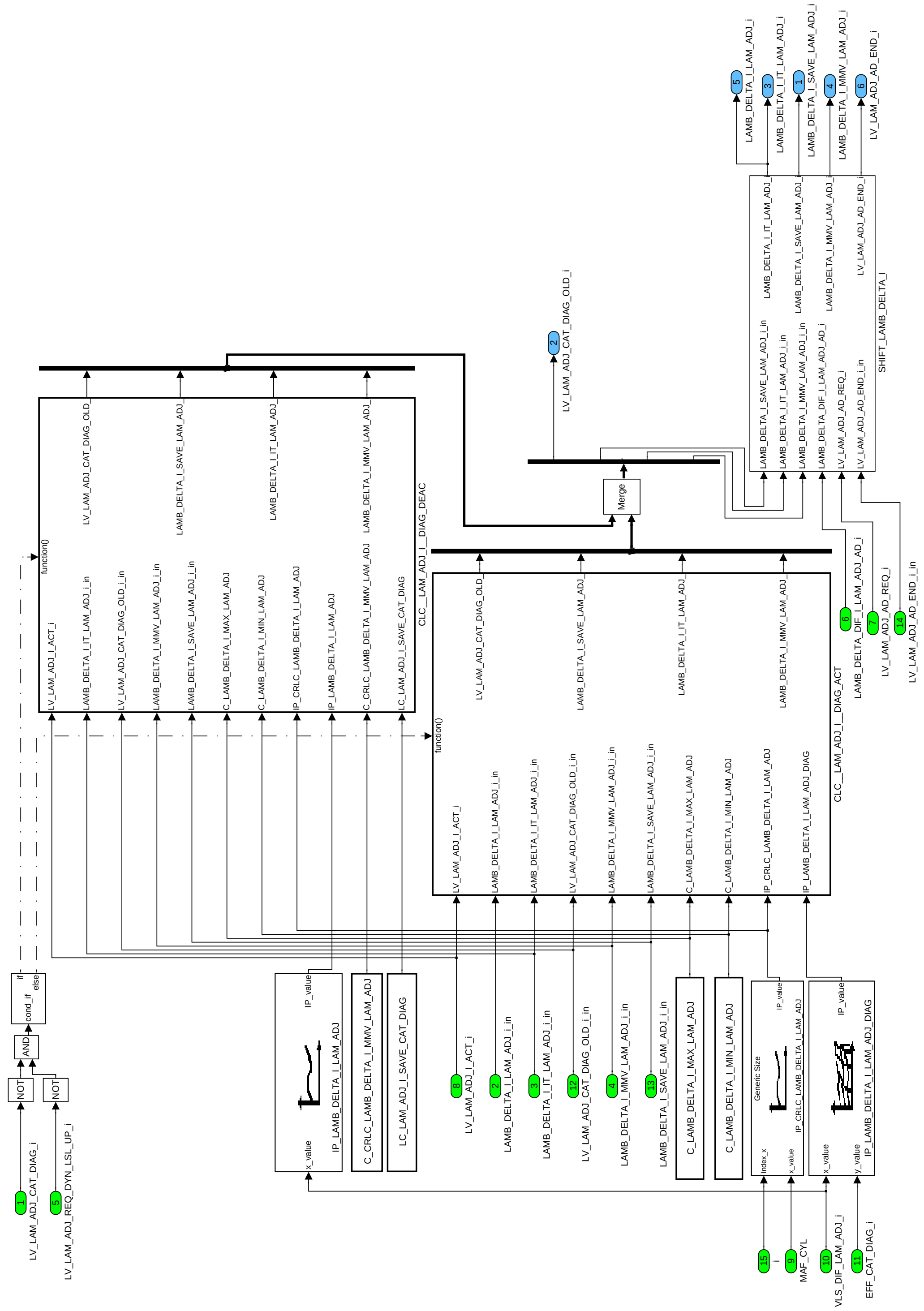


Figure 76 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ CLC__LAM_ADJ_I

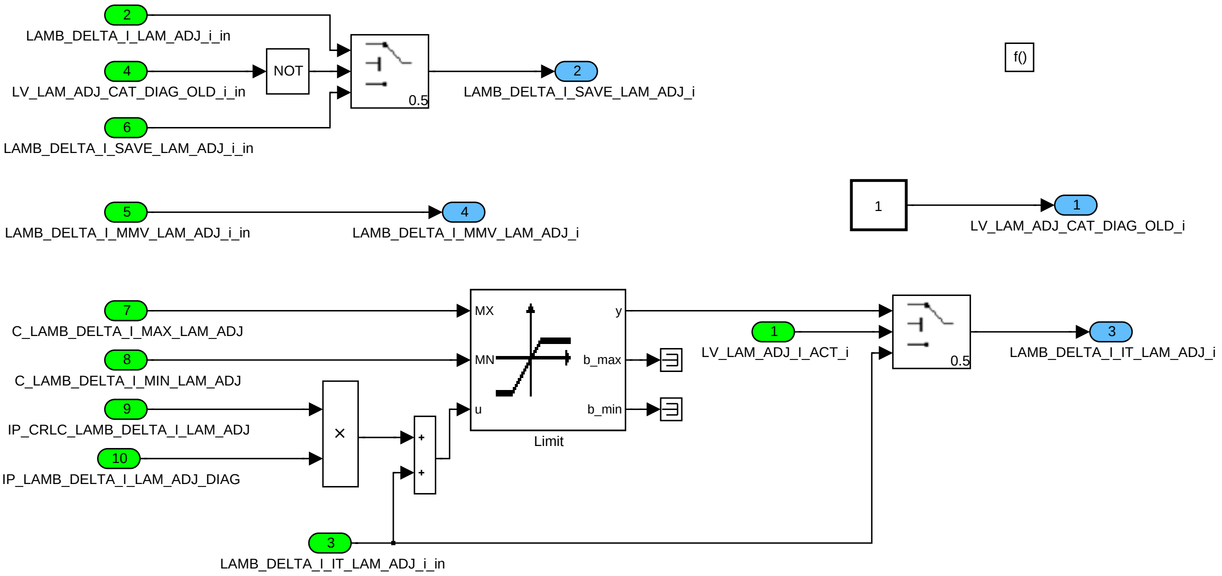


Figure 77 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ CLC__LAM_ADJ_I/ CLC__LAM_ADJ_I__DIAG_ACT

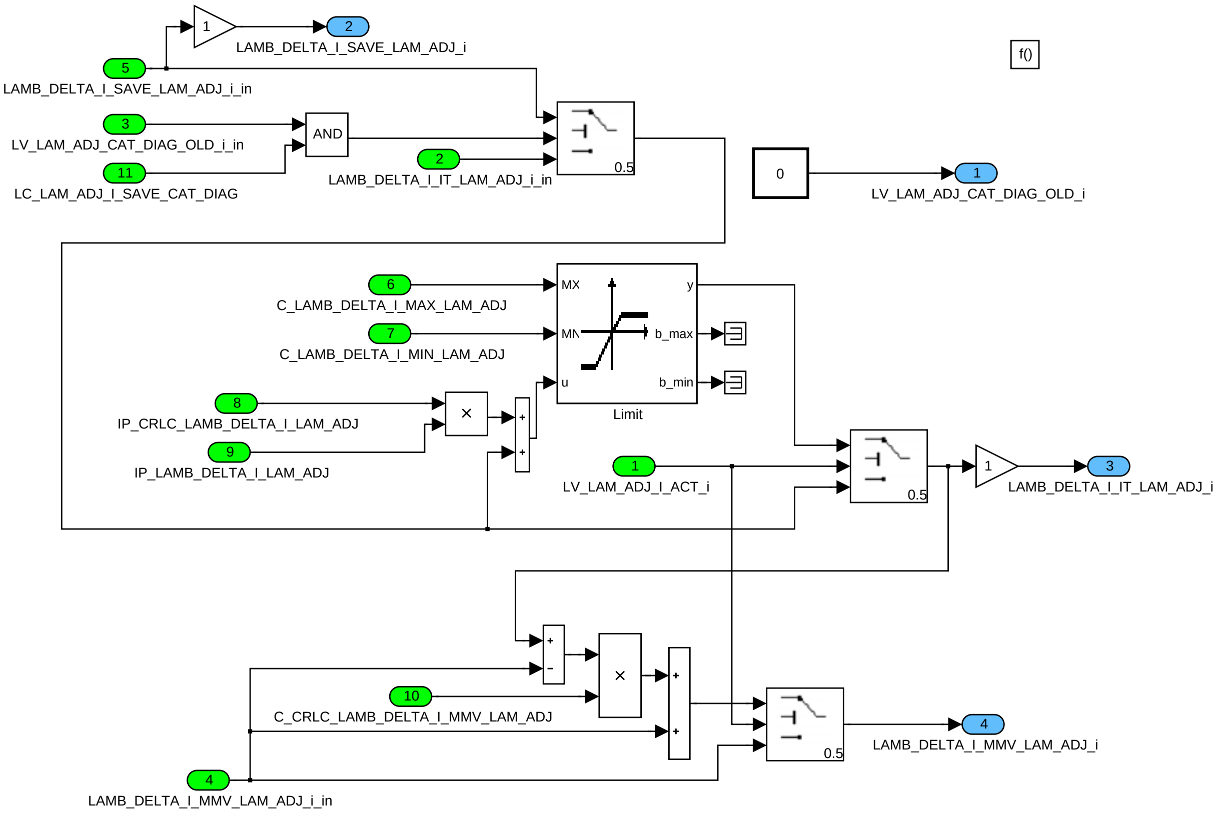


Figure 78 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ CLC__LAM_ADJ_I/ CLC__LAM_ADJ_I__DIAG_DEAC

Shift of I share

When requested by the function “WRAF sensor characterisitc line adaptation” a shift of all I share components is applied. The adaptation value in the mentioned function executes an opposite shift.

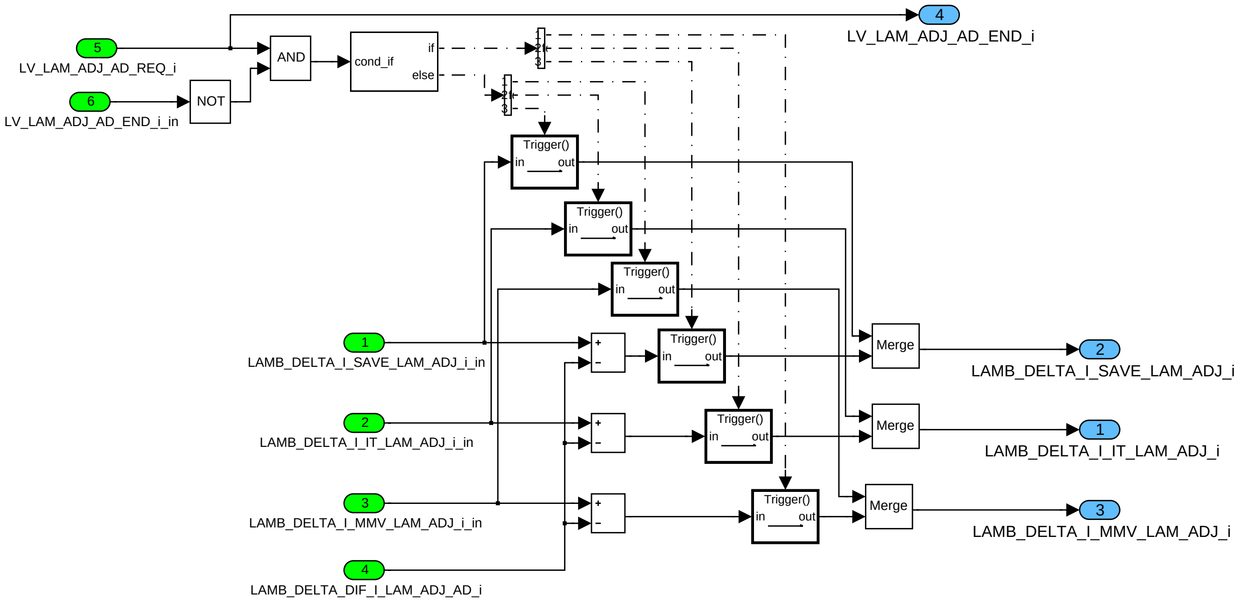


Figure 79 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ CLC__LAM_ADJ_I/ SHIFT_LAMB_DELTA_I

P share calculation

The P share is calculated when LV_LAM_ADJ_P_ACT[i] is 1. Otherwise the P share is 0. In case of active catalyst diagnosis an own P share gain characteristic line is used.

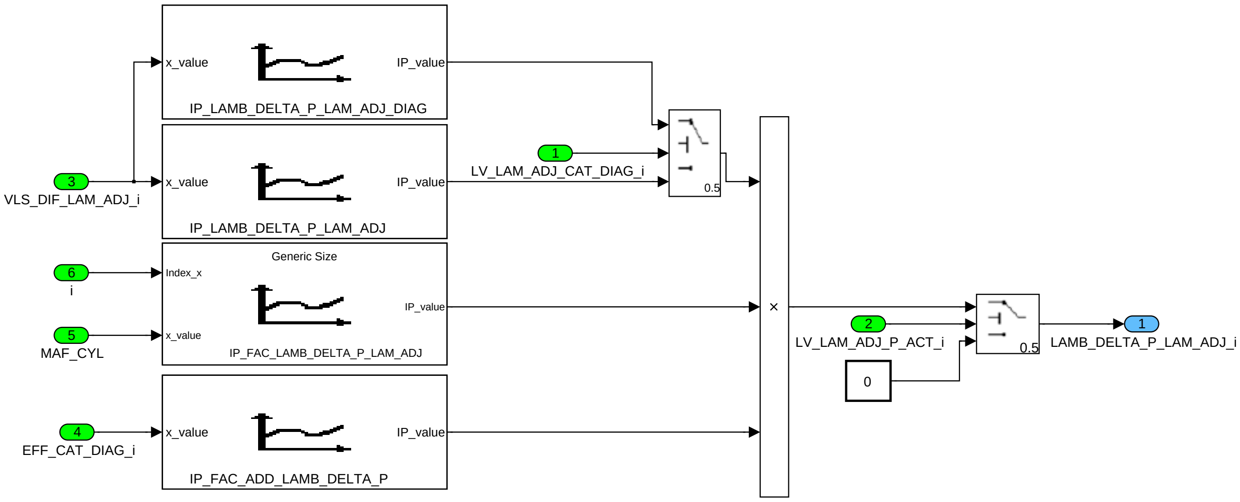


Figure 80 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ CLC__LAM_ADJ_P

Calculation of total lambda shift

The total lambda shift LAMB_DELTA_LAM_ADJ[i] is calculated by adding P share, I share and D share.

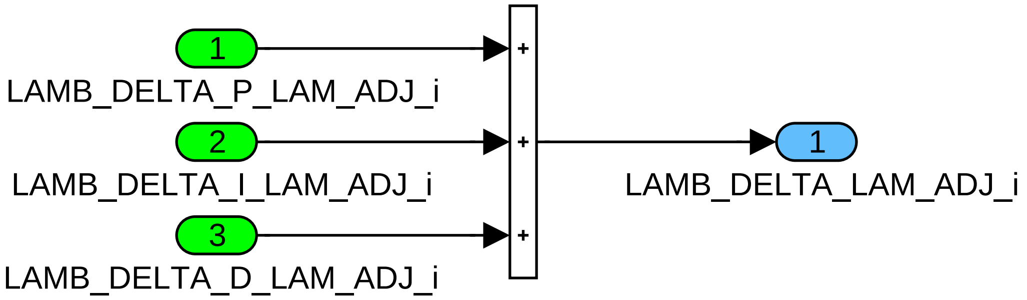


Figure 81 LACO_REQGNTRIM0/ operate/ LACO_REQGNTRIM0_CBK_MNG/ LACO_REQGNTRIM0_CBK_SPC/ CLC__LAM_ADJ_SHIFT

46.7.1.2 SUBFUNCTION: reset

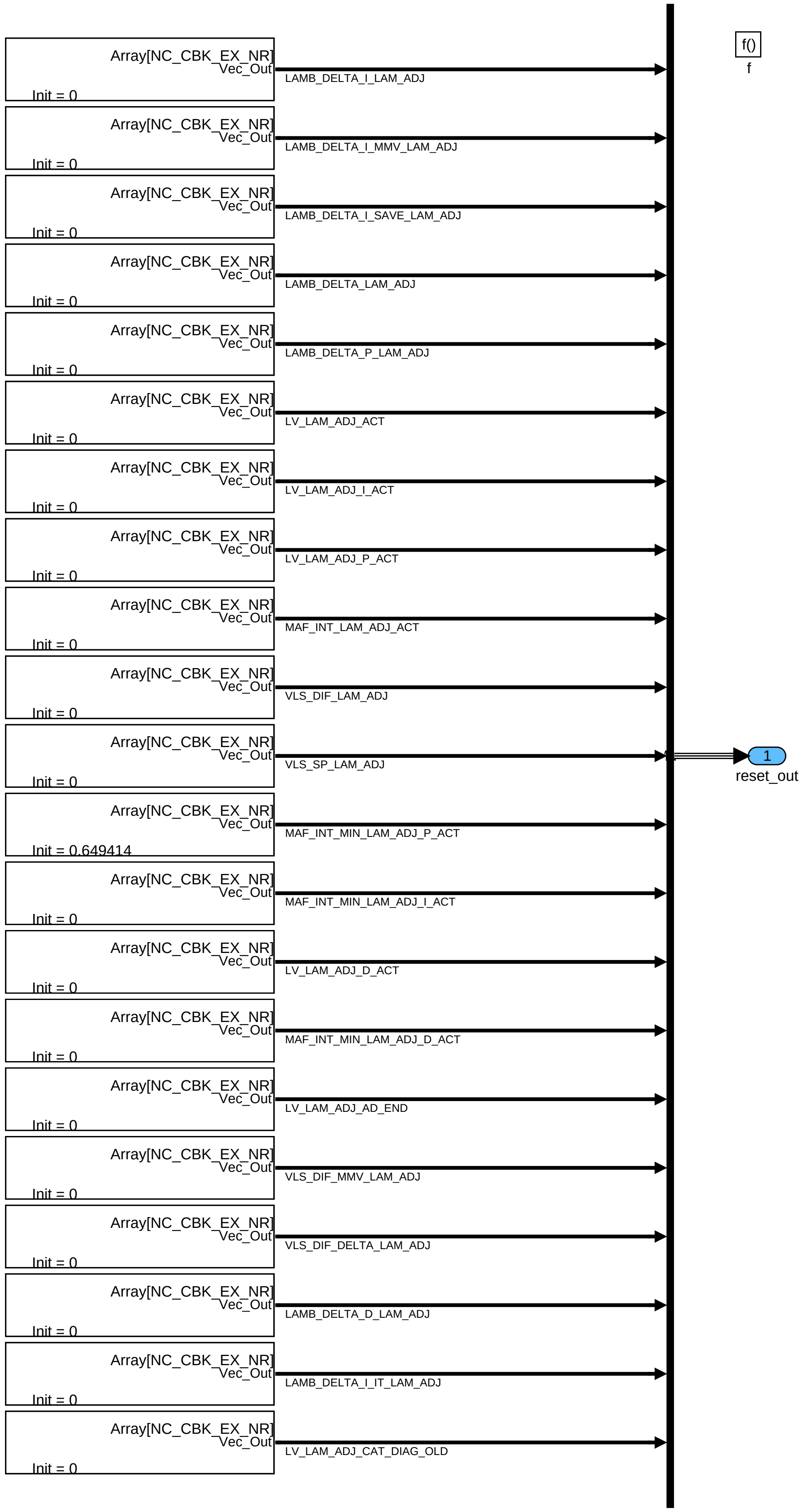

f()
f
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LAMB_DELTA_I_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LAMB_DELTA_I_MMV_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LAMB_DELTA_I_SAVE_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LAMB_DELTA_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LAMB_DELTA_P_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_LAM_ADJ_ACT
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_LAM_ADJ_I_ACT
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_LAM_ADJ_P_ACT
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
MAF_INT_LAM_ADJ_ACT
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
VLS_DIF_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
VLS_SP_LAM_ADJ
1
reset_out
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0.649414
MAF_INT_MIN_LAM_ADJ_P_ACT
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
MAF_INT_MIN_LAM_ADJ_I_ACT
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_LAM_ADJ_D_ACT
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
MAF_INT_MIN_LAM_ADJ_D_ACT
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_LAM_ADJ_AD_END
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
VLS_DIF_MMV_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
VLS_DIF_DELTA_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LAMB_DELTA_D_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LAMB_DELTA_I_IT_LAM_ADJ
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_LAM_ADJ_CAT_DIAG_OLD

Figure 82 LACO_REQGNTRIM0/ reset

46.8 WRAF sensor characteristic line adaption with trim control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_DELTA_AD_LAM_ADJ[NC_CBK_EX_NR]	O/V/S	F800...800H	-0.125...0.125	6.10352E-5	-
adaptation value for WRAF sensor characteritic line shift					
LAMB_DELTA_SUM_LAM_ADJ_AD[NC_CBK_EX_NR]	O/V	F800...800H	-0.125...0.125	6.10352E-5	-
total characteristic line shift of WRAF sensor					
LAMB_DELTA_DIF_I_LAM_ADJ_AD[NC_CBK_EX_NR]	O/V	F800...800H	-0.125...0.125	6.10352E-5	-
trim control I share shift in case of new adaptation value for WRAF sensor characteristic line shift					
LV_LAM_ADJ_AD_REQ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
request for trim control I share shift due to new adaptation value learning					
LAMB_DELTA_BAS_LAM_ADJ_AD[NC_CBK_EX_NR]	V	F800...800H	-0.125...0.125	6.10352E-5	-
basic characteristic line shift of WRAF sensor					
CTR_RAF_CHG_LAM_ADJ[NC_CBK_EX_NR]	V	0...FFH	0...255	1	-
counter for lean <-> rich air fuel ratio changes					

Input data:

LV_ST_END	NC_CBK_EX_NR	LV_IS	MAF
N_32	CTR_RAF_CHG[NC_CBK_EX_NR]	LV_LAM_ADJ_AD_END[NC_CBK_EX_NR]	LAMB_DELTA_I_MMV_LAM_ADJ[NC_CBK_EX_NR]
LV_LAM_ADJ_AD_CDN_OK[NC_CBK_EX_NR]	LV_LAMB_DELTA_AD_LAM_ADJ_CLC[NC_CBK_EX_NR]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_LAMB_DELTA_AD_LAM_ADJ	1	0...FFFFH	0...0.99998474	1.52588E-5	-
correlation constant of trim control adaptation value calculation					
C_LAMB_DELTA_BAS_LAM_ADJ_IS[NC_CBK_EX_NR]	1	F800...800H	-0.125...0.125	6.10352E-5	-
basic characteristic line shift of WRAF sensor in case of idle speed					
C_LAMB_DELTA_MAX_LAM_ADJ_AD	1	F800...800H	-0.125...0.125	6.10352E-5	-
maximum value for WRAF sensor characteristic line shift offset					
C_LAMB_DELTA_MIN_LAM_ADJ_AD	1	F800...800H	-0.125...0.125	6.10352E-5	-
minimum value for WRAF sensor characteristic line shift offset					
C_NR_RAF_CHG_WAIT_LAM_ADJ	1	1...FFH	1...255	1	-
number of lean <-> rich air fuel ratio changes between two adpatation cycles					
IP_LAMB_DELTA_BAS_LAM_ADJ[NC_CBK_EX_NR]	8x8	0...1000H	-0.125...0.125	6.10352E-5	-
LDP_MAF_IP_LAMB_DELTA_BAS	8	0...FFFFH	0...1.389E+3	0.021194781	mg/stk
LDP_N_32_IP_LAMB_DELTA_BAS	8	0...FFH	0...8.16E+3	32	rpm
basic characteric line shift of WRAF sensor					

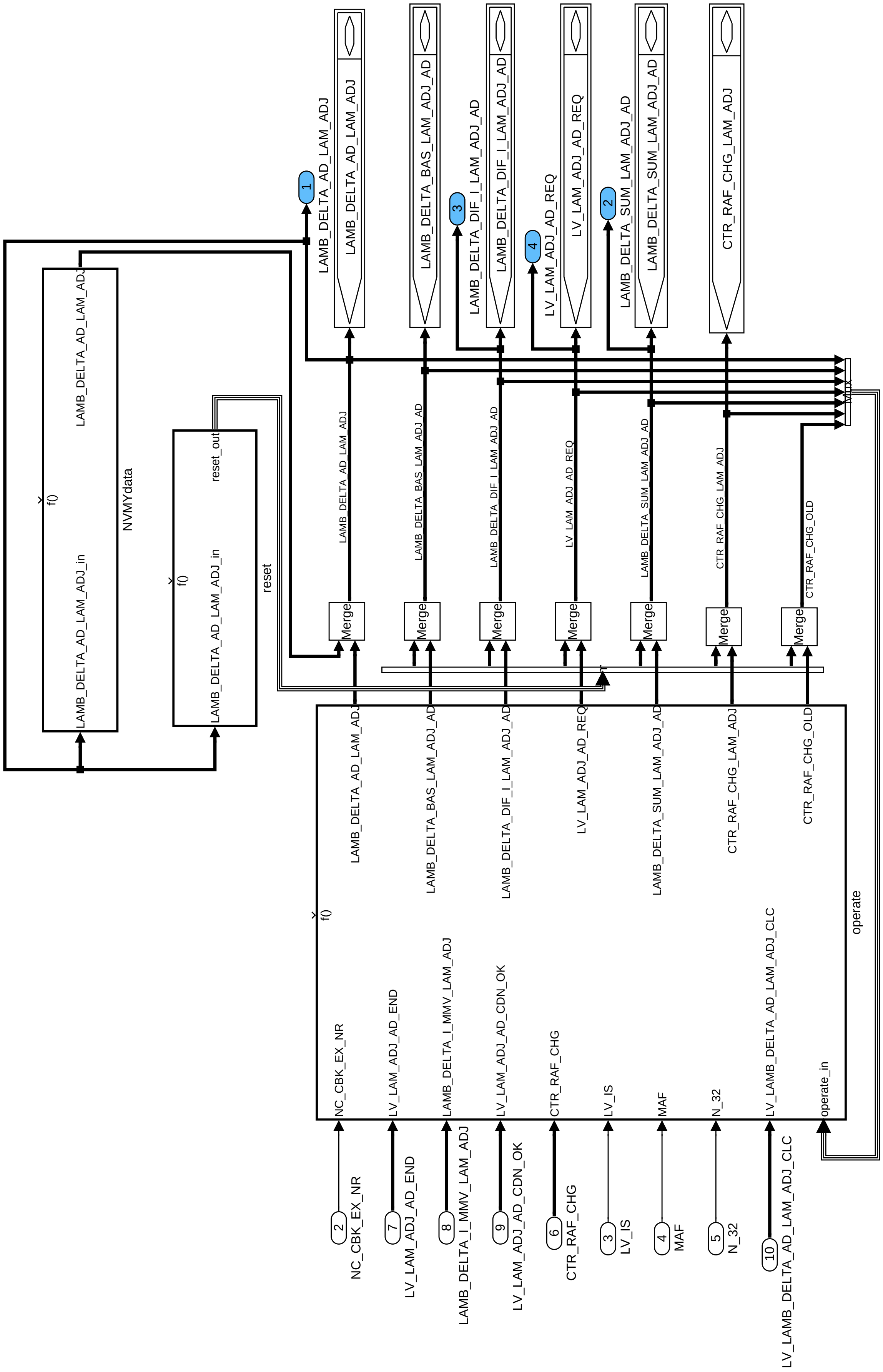
46.8.1 LACO_REQGNTRMAD0: General information

This function adapts the WRAF sensor characterisic line by observing the I share of the trim control function. A basic (load depending) lambda shift of the WRAF sensor can be corrected by means of an additional offset. This offset can change due to ageing of the sensor.

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.



SDA_SRS / SDA 3.1 02−Dec−2003

Chapter	Baseline		Include File
Lambda control	4DC3940S		30707J01.00B
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

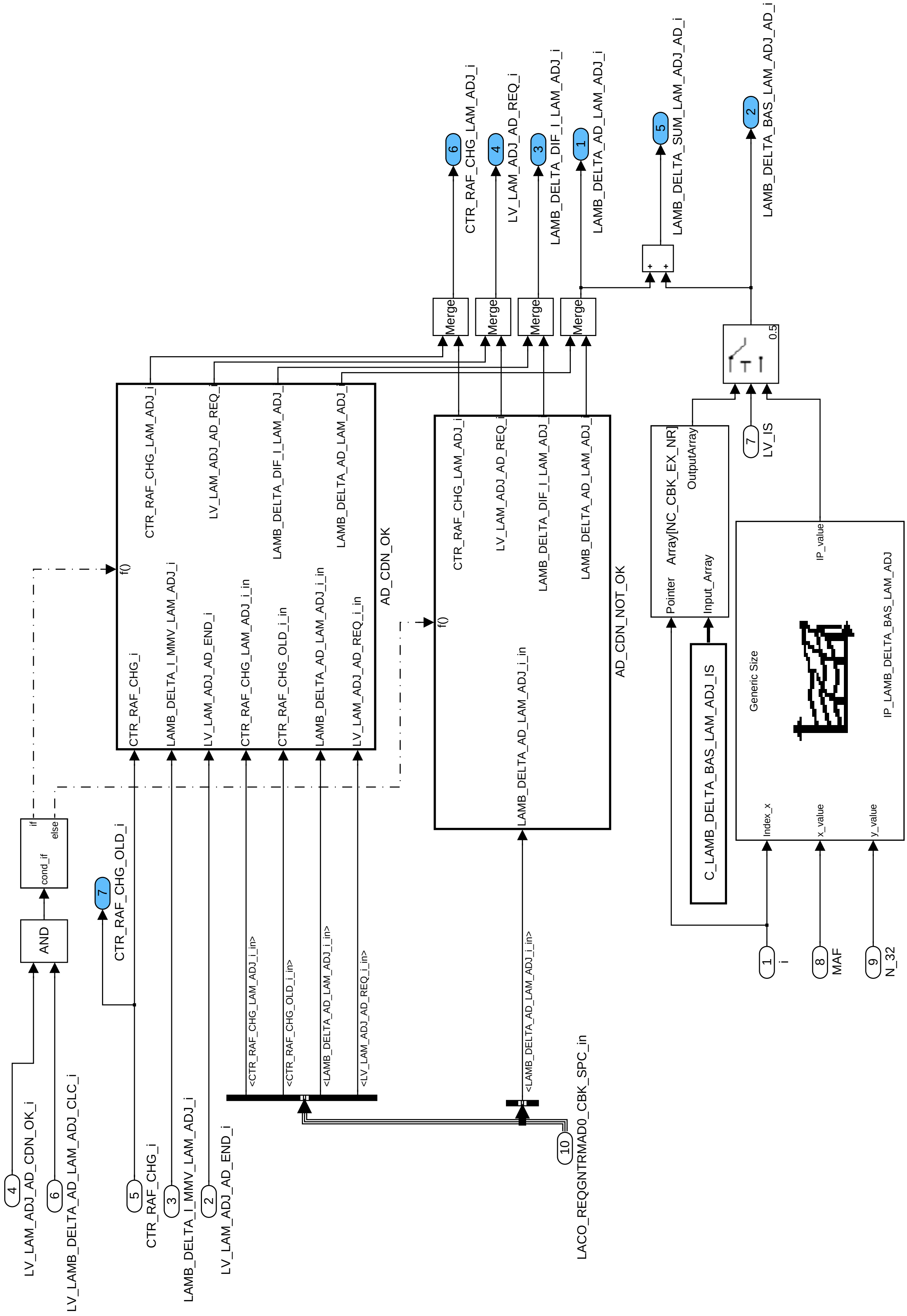
Pages
7336 of 9643

46.8.1.1 SUBFUNCTION: OPERATE

Basic WRAF sensor characteristic line shift

By means of IP_LAMB_DELTA_BAS_LAM_ADJ[i] the basic sensor characteristic line shift is defined MAF and N dependent. In idle speed the calibration data C_LAMB_DELTA_BAS_LAM_ADJ_IS[i] is applied. Together with the adaptive correction LAMB_DELTA_AD_LAM_ADJ[i] (stored as non volatile data) the total shift LAMB_DELTA_SUM_LAM_ADJ_AD[i] is defined.

When the conditions for activating the adaptation are fulfilled the sub-system “AD_CDN_OK” is executed. Otherwise the sub-system “AD_CDN_NOT_OK” is called where no action is executed.



Chapter Lambda control	Baseline 4DC3940S	Include File 30707J01.00B

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Continental

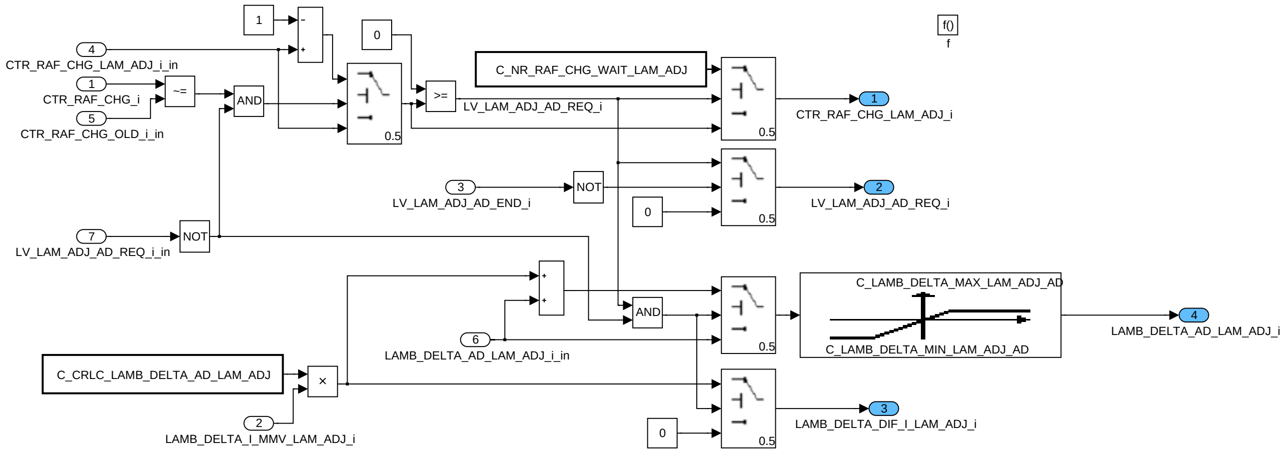
Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

LACO_REQGNTRMAD0/operate/LACO_REQGNTRMAD0_CBK_MNG/LACO_REQGNTRMAD0_CBK_SPC

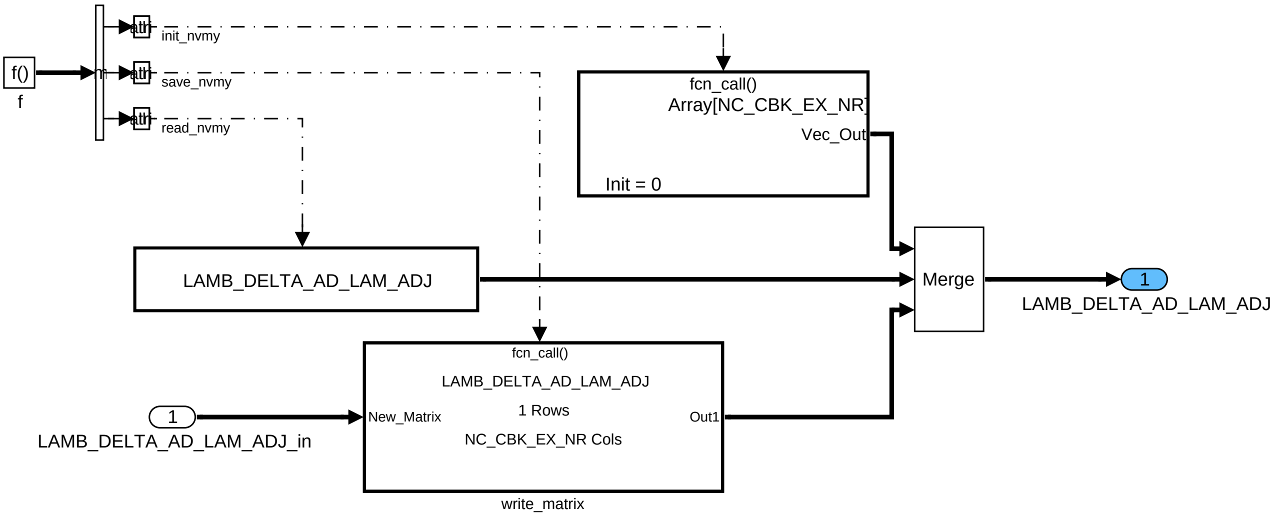
Adaptation of WRAF sensor characteristic line offset

The function waits a certain number of lean – rich air fuel ratio changes (C_NR_RAF_CHG_WAIT_LAM_ADJ) between 2 adaptation cycles. Then the current I share of the trim controller is shifted step by step into the adaptation value. Furthermore this latter value is limited on a calibration maximum and minimum.



LACO_REQGNTRMAD0/operate/LACO_REQGNTRMAD0_CBK_MNG/LACO_REQGNTRMAD0_CBK_SPC/AD_CDN_OK

46.8.1.2 Management of non volatile stored data



LACO_REQGNTRMAD0/NVMYdata

46.9 Valid condition check for WRAF sensor characterisitic line adaption with trim control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAM_ADJ_AD_CDN_OK[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating good conditions for WRAF sensor characteristic line adaptation					
MAF_INT_LAM_ADJ_CDN[NC_CBK_EX_NR]	V	0...FFFFH	0...1.82042E+3	0.0277778	g
air mass flow integral for condition detection of WRAF sensor characterisitc line adaptation					

Input data:

LV_ST_END	NC_CBK_EX_NR	LV_LAM_ADJ_CAT_DIAG[NC_CBK_EX_NR]	LV_LAMB_PLS_ACT[NC_CBK_EX_NR]
LV_CP_CLOSE_ACT	MAF_HB	MAF_CYL	N_32
LV_LDC_LAM_ADJ[NC_CBK_EX_NR]	LV_LAM_ADJ_I_ACT[NC_CBK_EX_NR]	LAMB_DELTA_P_LAM_ADJ[NC_CBK_EX_NR]	LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]
LAMB_DELTA_I_MMV_LAM_ADJ[NC_CBK_EX_NR]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_DELTA_DIF_I_MMV_LAM_ADJ	1	F800...800H	-0.125...0.125	6.10352E-5	-
difference between I share and moving mean value to allow WRAF sensor characteristic line adaptation					
C_LAMB_DELTA_P_MAX_LAM_ADJ_CDN	1	F800...800H	-0.125...0.125	6.10352E-5	-
maximum trim control P share for WRAF sensor characteristric line adaptation					
C_LAMB_DELTA_P_MIN_LAM_ADJ_CDN	1	F800...800H	-0.125...0.125	6.10352E-5	-
minimum trim control P share for WRAF sensor characteristric line adaptation					
C_MAF_INT_THD_LAM_ADJ_CDN	1	0...FFFFH	0...1.82042E+3	0.0277778	g
air mass flow integral threshold for stationary condition definition for WRAF sensor characteristic line adaptation					
C_MAF_MAX_LAM_ADJ_CDN	1	0...FFH	0...1.389E+3	5.44706	mg/stk
maximum air mass flow threshold for WRAF sensor characteristic line adaptation					
C_MAF_MIN_LAM_ADJ_CDN	1	0...FFH	0...1.389E+3	5.44706	mg/stk
minimum air mass flow threshold for WRAF sensor characteristic line adaptation					
C_N_MAX_LAM_ADJ_CDN	1	0...FFH	0...8.16E+3	32	rpm
maximum engine speed threshold for WRAF sensor characteristic line adaptation					
C_N_MIN_LAM_ADJ_CDN	1	0...FFH	0...8.16E+3	32	rpm
minimum engine speed threshold for WRAF sensor characteristic line adaptation					

46.9.1 LACO_REQGNTRCDN0

This function checks several conditions that must be fulfilled in order to start the WRAF sensor characteristic line adaption. These conditions shall define a stable state of the downstream sensor signal used by trim control.

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Function Description

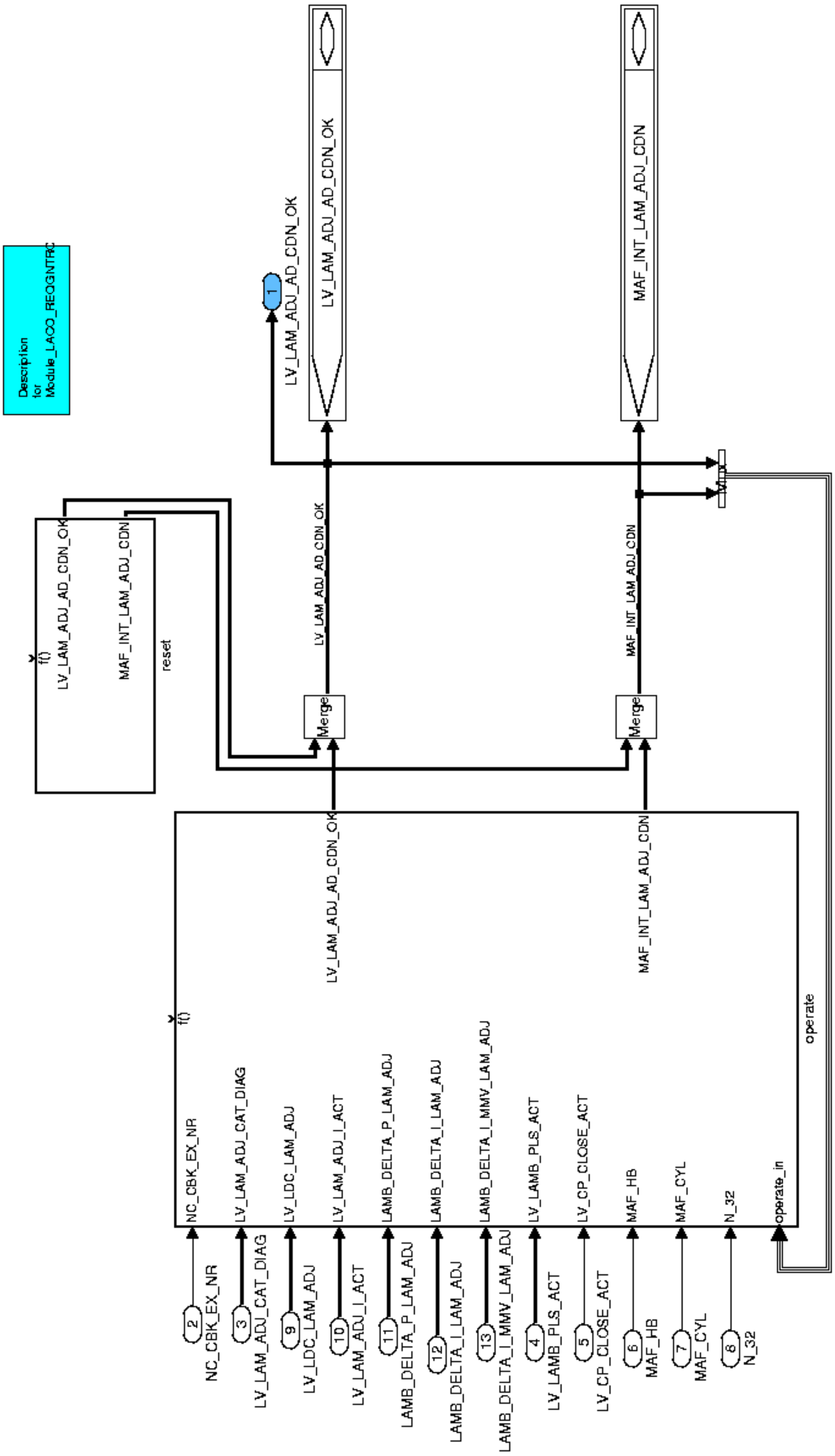


Figure 83 LACO_REQGNTRCDN0

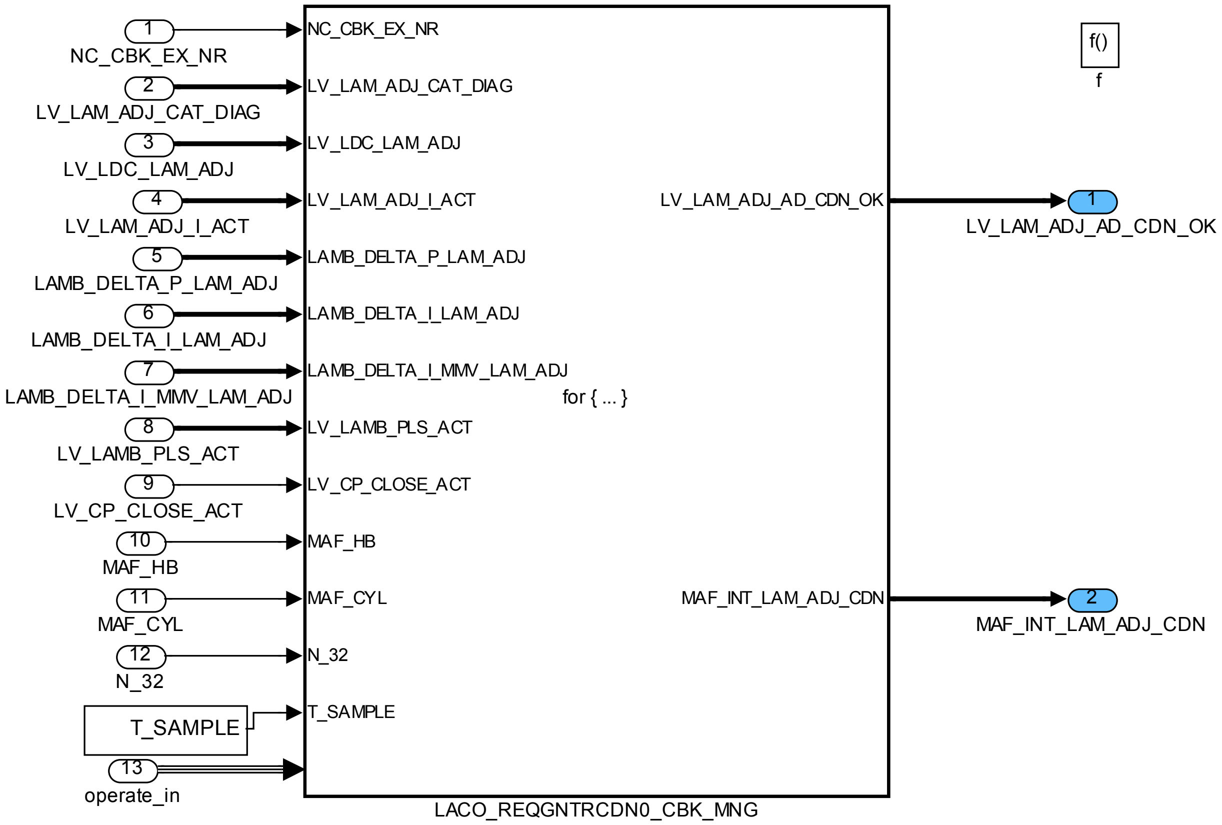


Figure 84 LACO_REQGNTRCDN0/ operate

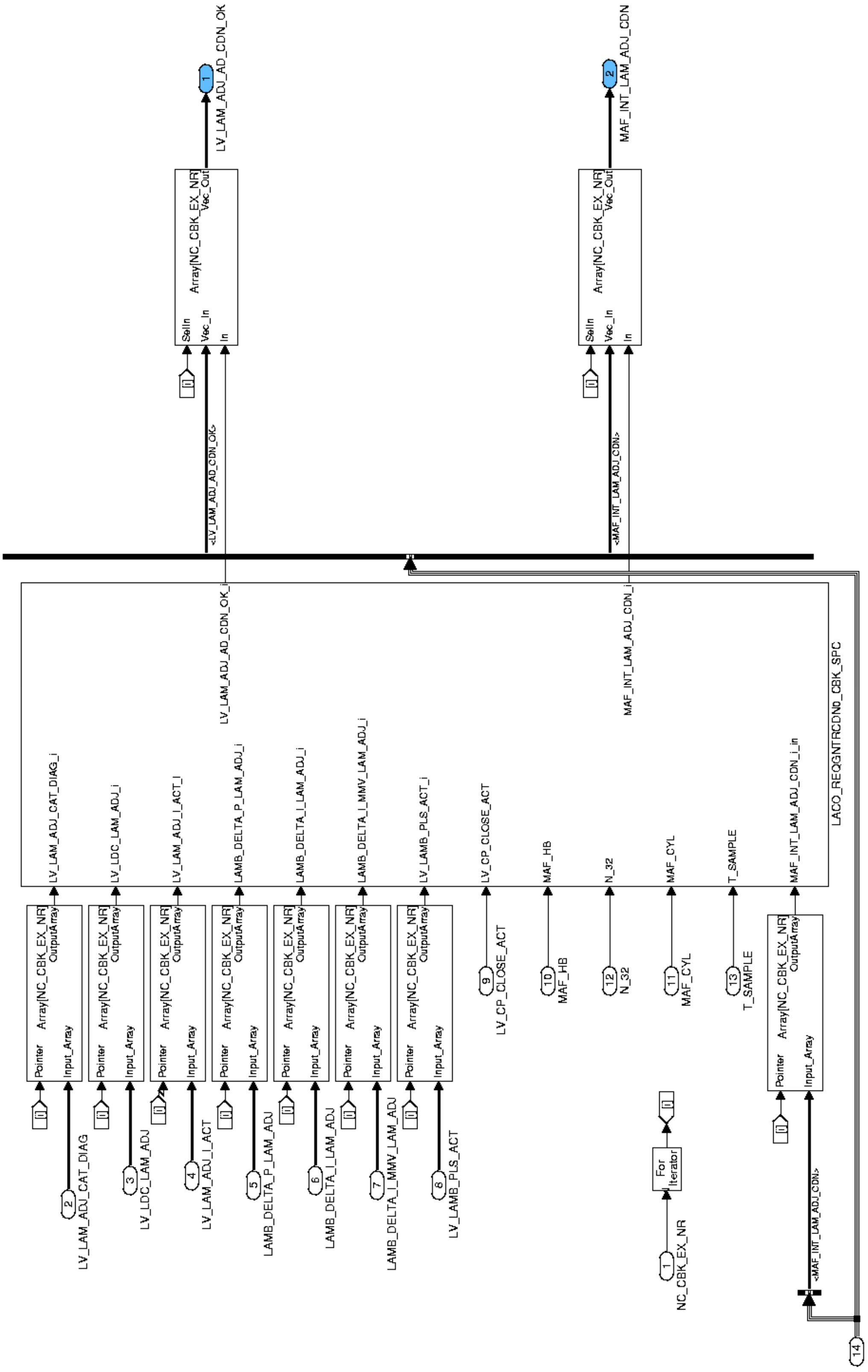


Figure 85 LACO_REQGNTRCDN0/ operate/ LACO_REQGNTRCDN0_CBK_MNG

Check conditions

If all the following conditions are fulfilled an air mass flow integral is calulated:

- Catalyst efficiency diangnosis not active

- limited dynamics (LV_LDC_LAM_ADJ[i]) are fulfilled
- I share of trim control is active
- forced lambda stimulation is active
- canister purge valve is closed
- air mass flow is in a valid range
- engine speed is in a valid range
- P share of trim contol must be very low (absolute value)
- I share of trim control must be stable
  (difference between I share and moving mean value must be small)

When the air mass flow integral exceeds a calibration threshold the flag LV_LAM_ADJ_AD_CDN_OK[i] indicating that the adaptation can be started is set.

For the calcuation of the air mass flow integral the unit conversion:

[g] = [kg/h] * [ms] * [(g*h) / (kg*ms)] / 3600

has to be considered.

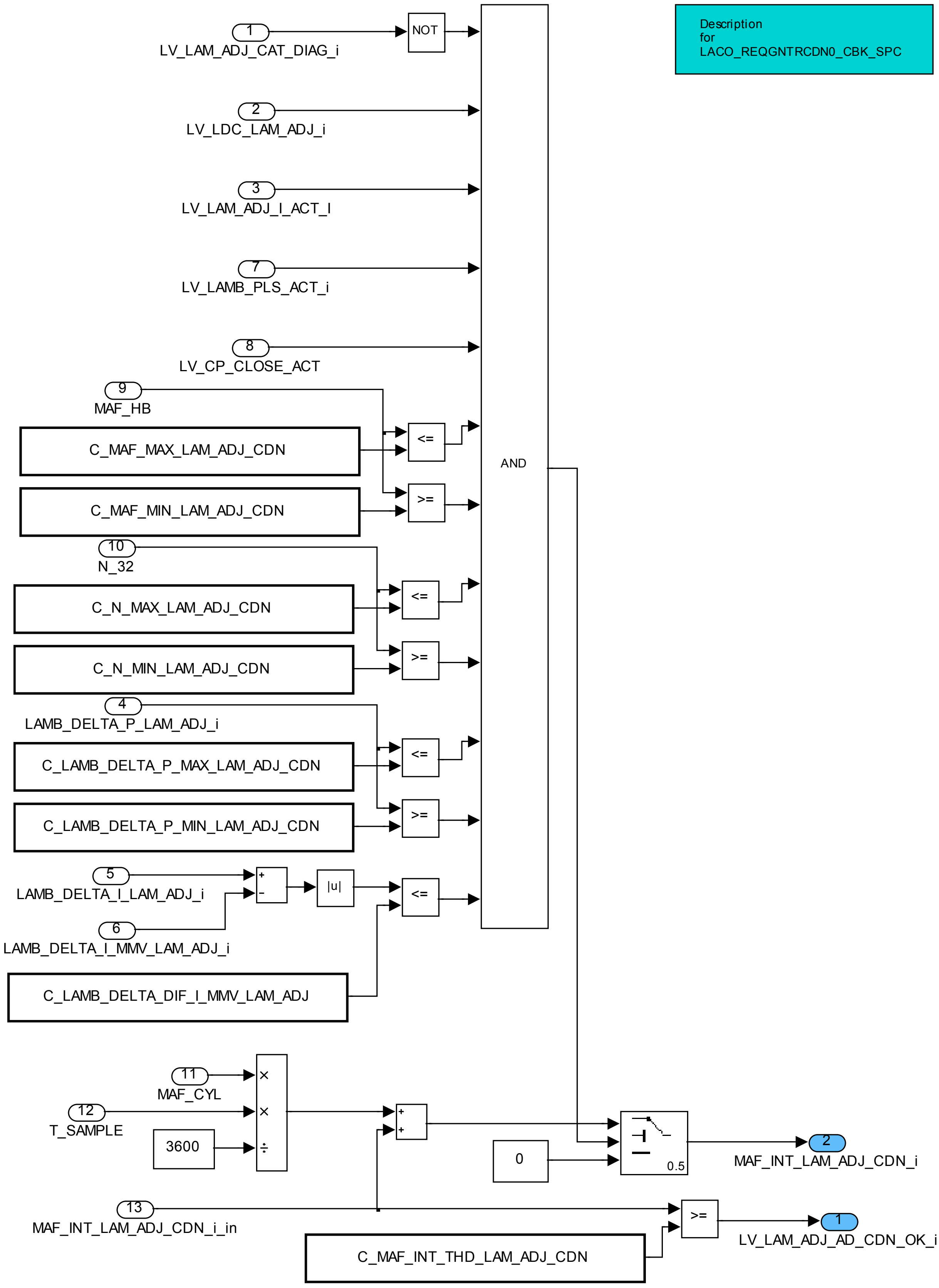


Figure 86 LACO_REQGNTRCDN0/ operate/ LACO_REQGNTRCDN0_CBK_MNG/ LACO_REQGNTRCDN0_CBK_SPC

46.10 Application incidences for trim control for linear lambda control

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>LV_INH_LAM_ADJ[NC_CBK_EX_NR]</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">inhibition flag for trim control</td></tr>
<tr><td>LV_LAMB_DELTA_I_LAM_ADJ_DEAC[NC_CBK_EX_NR]</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">flag indicating that I share is fixed when deactivating lambda controller</td></tr>
<tr><td>LV_LAMB_DELTA_AD_LAM_ADJ_CLC[NC_CBK_EX_NR]</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">flag indicating that adaptation value is calculated at start of power latch phase</td></tr>
<tr><td>VLS_AV_LAM_ADJ[NC_CBK_EX_NR]</td><td>O/V</td><td>0...3FFH</td><td>0...4.99511</td><td>4.8828e-3</td><td>[V]</td></tr>
<tr><td colspan="6">sensor voltage actual value for trim control</td></tr>
<tr><td>LV_LDC_LAM_ADJ[NC_CBK_EX_NR]</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">limited dynamic conditions for trim control</td></tr>
<tr><td>VLS_DELTA_LAM_ADJ_CAT_DIAG[NC_CBK_EX_NR]</td><td>O/V</td><td>0...3FFH</td><td>0...4.99511</td><td>4.8828e-3</td><td>[V]</td></tr>
<tr><td colspan="6">Sensor voltage set point shift in case of active cat efficiency diagnosis</td></tr>
<tr><td>LV_LAMB_DELTA_AD_LAM_ADJ_EXT</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">flag to set trim control adaptation value to calibration value (external adjustment)</td></tr>
<tr><td>LV_LAM_ADJ_ACT_FAST_LAM_LSCL[NC_CBK_EX_NR]</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">activation flag for fast activation of dynamic fuel trim</td></tr>
</table>

Input data:

LV_ERR_CAM	LV_ERR_CHG_LS_DOWN	LV_ERR_IVVT	LV_ERR_CRK
LV_ERR_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_TOOTH_OFF_IN[NC_NR_CAM_CBK]	LV_ERR_TPS	LV_ERR_CHG_LS_DOWN
LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	VLS_DOWN[NC_CBK_EX_NR]	LV_ERR_FSD[NC_CBK_EX_NR]	T_AST
LV_VAR_TCHA	LV_LAM_ADJ_CAT_DIAG[NC_CBK_EX_NR]	LV_LAM_ADJ_ACT[NC_CBK_EX_NR]	MAF_KGH
LV_LDC_DLY[NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_ERR_TOOTH_OFF_EX[NC_NR_CAM_CBK]	LV_ERR_TCO
LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_FSD_H_RNG[NC_CBK_EX_NR]
LV_ERR_MAP_TPS_PLAUS	LV_LAM_AD_INJ_ACT	VLS_DELTA_LAM_ADJ_NS_SHIFT[NC_CBK_EX_NR]	LV_LAM_ADJ_NS_SHIFT_DIAG[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

There are many errors which make it necessary to stop the trim control when they occure.
The calculation shall be done for all exhaust cylinder banks.

For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then
i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2,

otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Application conditions:

Initialisation: at ECU reset:

LV_INH_LAM_ADJ[i] = 1
LV_LAMB_DELTA_AD_LAM_ADJ_CLC[i] = 1
LV_LAMB_DELTA_I_LAM_ADJ_DEAC[i] = 1
VLS_AV_LAM_ADJ[i] = 0.649414 V
LV_LDC_LAM_ADJ[i] = 0
LV_LAMB_DELTA_AD_LAM_ADJ_EXT = 0

Recurrence: T_SAMPLE = 20 ms

Activation: in all engine operating states

Deactivation: -

Formula section:

46.10.1 Inhibit switch

```
If      LV_ERR_TPS = 1
   or   LV_ERR_CRK = 1
   or   LV_ERR_CAM = 1
   or   LV_ERR_LS_DOWN[i] = 1
   or   LV_ERR_IVVT =  1
   or   LV_ERR_CHG_LS_DOWN=  1
   or   LV_ERR_FSD_LAM_LIM[i] =  1
   or   LV_ERR_FSD_H_RNG[i] =  1
   or   LV_ERR_FSD[i] =  1
   or   LV_ERR_TCO  =  1
   or   T_AST < ID_T_DLY_LAM_AD[i]
   or   LV_ERR_DELTA_I_LAM[i] = 1
   or   LV_ERR_VLS_DOWN_DIF[i] = 1
   or   LV_ERR_TTIP_MES_LSH_UP[i] = 1
   or   LV_ERR_TOOTH_OFF_EX_1 = 1
   or   LV_ERR_TOOTH_OFF_IN_1 = 1
   or   LV_ERR_MAP_TPS_PLAUS = 1
   or   (LV_LAM_AD_INJ_ACT = 1 & LC_LAM_ADJ_INH_ENA_LAM_AD_INJ = 1)

Then    LV_INH_LAM_ADJ[i]          =  1
Else    LV_INH_LAM_ADJ[i]          =  0
Endif
LV_LAMB_DELTA_AD_LAM_ADJ_CLC[i]      =  NOT   (LV_INH_LAM_ADJ[i])
LV_LAMB_DELTA_I_LAM_ADJ_DEAC[i]      =  NOT   (LV_INH_LAM_ADJ[i])
```

46.10.2 Trim control input

```
If(1)        LV_LAM_ADJ_ACT[i]  =  0
Then(1)      VLS_AV_LAM_ADJ[i]  =  VLS_DOWN[i]
Else(1)
   If(2)     LV_LAM_ADJ_CAT_DIAG[i] = 0
   Then(2)
       If(3) LV_LAM_ADJ_NS_SHIFT_DIAG[i] = 1
       Then(3)
             VLS_AV_LAM_ADJ[i]  =  VLS_AV_LAM_ADJ[i] +
                                   (VLS_DOWN[i]
                                   - VLS_AV_LAM_ADJ[i] +
                                   VLS_DELTA_LAM_ADJ_NS_SHIFT[i])
                                   * ID_CRLC_VLS_DOWN_NS_SHIFT(MAF_KGH)
       Else(3)
             VLS_AV_LAM_ADJ[i]  =  VLS_AV_LAM_ADJ[i] + (VLS_DOWN[i]
                                   - VLS_AV_LAM_ADJ[i])
                                   * ID_CRLC_VLS_DOWN_T_DLY
       Endif(3)
   Else(2)
             VLS_DELTA_LAM_ADJ_CAT_DIAG[i]  =
                                   IP_VLS_DELTA_LAM_ADJ_CAT_DIAG
             VLS_AV_LAM_ADJ[i]   =  VLS_AV_LAM_ADJ[i] +
                                   (VLS_DOWN[i] - VLS_AV_LAM_ADJ[i] +
                                   VLS_DELTA_LAM_ADJ_CAT_DIAG[i]  )
                                   * IP_CRLC_VLS_LAM_ADJ_CAT_DIAG
   Endif(2)
Endif(1)
```

SW hint: Definitions of "VLS" variables are not the same!

// VLS_AV_LAM_ADJ[i] must be calculated SW internally with high resolution (32 bit) and must convergate. VLS_AV_LAM_ADJ[i] must be initialised with VLS_DOWN[i] at function activation//

Adaptation for Exhaust Gas Configuration using 2 linear and 1 binary O2 sensors

```
If  LC_VAR_EX_CUS = 1
Then  VLS_AV_LAM_ADJ[2] = VLS_AV_LAM_ADJ[1]
Endif
```

Interface for external adjustment of adaptation value for the λ-shift:

LV_LAMB_DELTA_AD_LAM_ADJ_EXT= LC_LAMB_DELTA_AD_LAM_ADJ_EXT

46.10.3 Determination of limited dynamic conditions for the trim controller

LV_LDC_LAM_ADJ[i] = LV_LDC_DLY[i]

46.10.4 Assignment of LV_LAM_ADJ_ACT_FAST_LAM_LSCL[i]

LV_LAM_ADJ_ACT_FAST_LAM_LSCL[i] = LC_LAM_ADJ_ACT_FAST_LAM_LSCL

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_CRLC_VLS_DOWN_T_DLY	1	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_MAF_KGH__CRLC_VLS_DOWN	4	0...FFFFH	0...2047.96875	0.03125	[kg/h]
correlation constant downstream signal					
LC_LAMB_DELTA_AD_LAM_ADJ_EXT	1	0...1H	0...1	1	[-]
LC for external adjustment of adaptation value for the LAMBDA-shift					
IP_VLS_DELTA_LAM_ADJ_CAT_DIAG	1	0...3FFH	0...4.99511	4.8828e-3	[V]
LDPM_EFF_CAT_DIAG_1_LACO	8	0...FFH	0...1.99218	0.0078125	[-]
sensor voltage set point shift in case of active cat efficiency diagnosis					
IP_CRLC_VLS_LAM_ADJ_CAT_DIAG	1	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_T_SUM_AFL_AFR_IP_CRLC_VLS	6	0...FFFFH	0...655.35	0.01	[s]
correlation constant for filtering downstream signal for catalyst efficiency diagnosis					
ID_T_DLY_LAM_AD[NC_CBK_EX_NR]	1	0...FFFFH	0...6553.5	0.1	[s]
LDP_TCO_ST_1_EGCP	6	0...FEH	-48...142.5	0.75	[°C]
activation of trim controller after calibratable time					
LC_VAR_EX_CUS	1	0...1H	0...1	1	[-]
Switch to an exhaust gas configuration with 2 linear and 1 binary O2 sensor (LC = 1)					
LC_LAM_ADJ_ACT_FAST_LAM_LSCL	1	0...1H	0...1	1	[-]
Switch to activate fast activation of dynamic fuel trim					
LC_LAM_ADJ_INH_ENA_LAM_AD_INJ	1	0...1H	0...1	1	[-]
Switch to enable inhibtation of trim controller during MFMA operation					
ID_CRLC_VLS_DOWN_NS_SHIFT	4	0...FFH	0...0.99609	3.9063e-3	[-]
LDPM_MAF_KGH__CRLC_VLS_DOWN	4	0...FFFFH	0...2047.96875	0.03125	[kg/h]
correlation constant downstream signal for NOx sensor shift diagnosis					

46.11 Detection of limited dynamic condition for dynamic fuel trim

Output data:

Name	V/S	Hex. limits	Phys. limits	Resol.	Unit
LV_LDC_DLY[NC_CBK_EX_NR]	V/O	0...1H	0...1	1	[-]
status flag indicated the limited dynamic conditions is fullfiled					
LV_N_LDC_DLY	V	0...1H	0...1	1	[-]
Status limited dynamic N					
LV_MAF_LDC_DLY	V	0...1H	0...1	1	[-]
Status limited dynamic MAF					
LV_FAC_TQ_REQ_LDC_DLY	V	0...1H	0...1	1	[-]
Status limited dynamic of the requested torque factor					
LV_LAM_MV_LDC_DLY[NC_CBK_EX_NR]	V	0...1H	0...1	1	[-]
Status limited dynamic FAC_LAM_MV[NC_CBK_EX_NR]					
N_MMV_DLY	V/O	0...1FE0H	0...8160	1	[rpm]
floating mean value for limited engine speed					
MAF_MMV_DLY	V/O	0...FFFFH	0...1389	2.12E-02	[mg/stk]
floating mean value for limited MAF					
FAC_LAM_MV_MMV_DLY[NC_CBK_EX_NR]	V/O	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
floating mean value for limited lambda deviation					
MAF_INT_LDC_DLY[NC_CBK_EX_NR]	V/O	0...FFFFH	0...1820.42	0.0277778	[g]
integral of mass air flow since limited dynamic conditions fulfilled					
FAC_TQ_REQ_GRD	V	8000...7FFFH	-1...0.99996	0.0305e-3	[-]
Gradient of the requested factor torque					
FAC_TQ_REQ_GRD_SUM_DLY	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Summation of FAC_TQ_REQ_GRD to determine if a dynamic condition is present					
T_FAC_TQ_REQ_GRD_SUM_DLY	V	0...FFH	0...5.1	0.02	[s]
Timer to decrement FAC_TQ_REQ_GRD_SUM_DLY					

Input data:

N	MAF	FAC_TQ_REQ	FAC_LAM_MV[NC_CBK_EX_NR]
LV_IGK	LV_ST_END	MAF_KGH	LV_VAR_LSH_UP
NC_FAC_MAF_INT_20			

FUNCTION DESCRIPTION:

General information:

For a two bank exhaust gas configuration the calculation of the limited dynamic condition LV_LDC_DLY[i] is calculated for each bank separately. To differentiate variables and calibration data the index i is used.

i = 1, for cylinder bank 1
i = 2, for cylinder bank 2

The limited dynamic conditions LV_LDC_DLY[i] for dynamic fuel trim is detected if engine speed, mass air flow, FAC_LAM_MV[i] and the factor torque request gradient are within the correspondent dynamic window. Once this condition is met, an integral of the air flow is incremented until the calibrated threshold exceeds. After this point the flag LV_LDC_DLY[i] is set to indicate the trim controller, that the engine is running under limited dynamic conditions.

Application conditions:

Initialisation: *at reset:*
LV_LDC_DLY[i] = 0
MAF_INT_LDC_DLY[i] = 0

Recurrence: 20ms

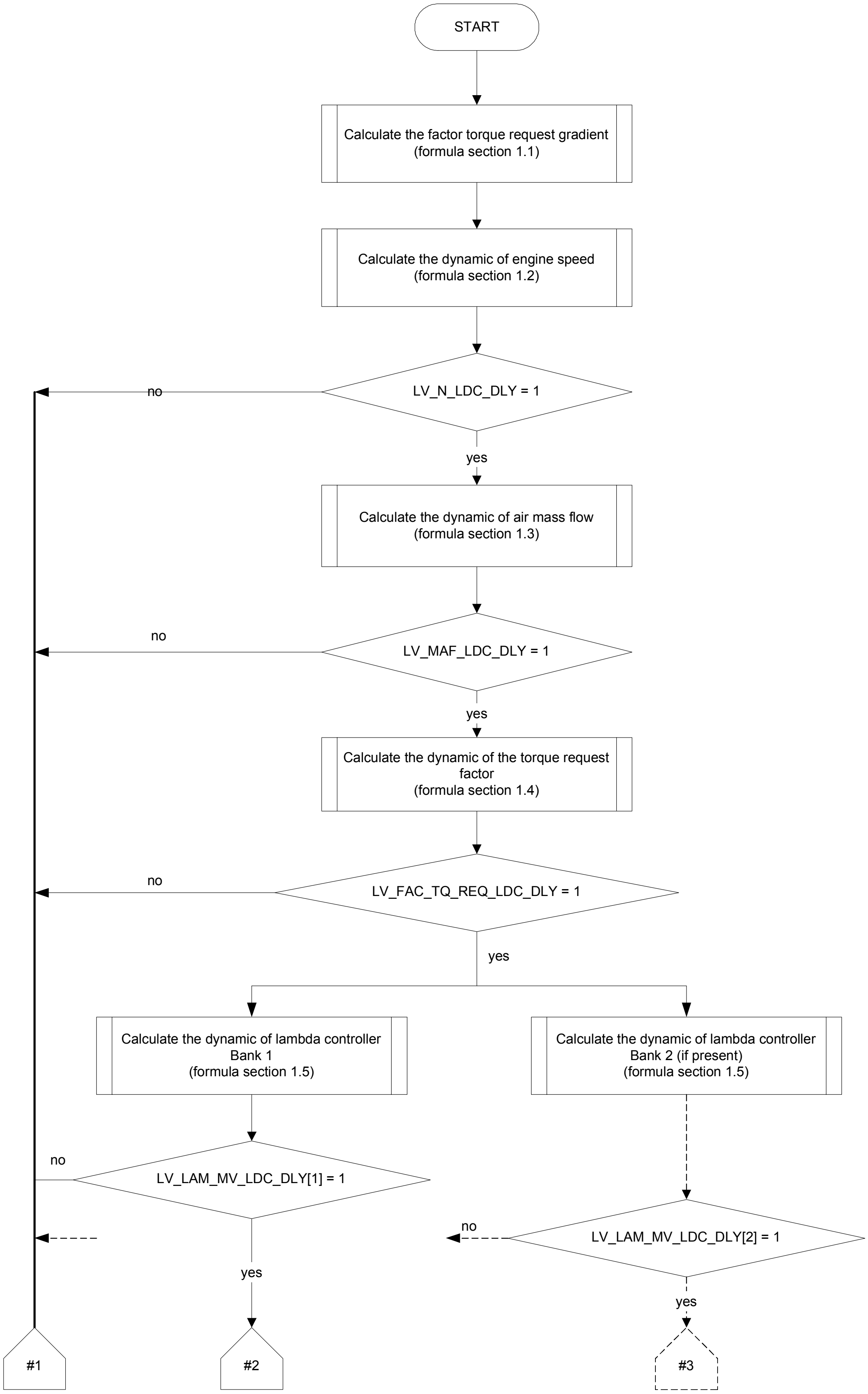
Activation: LV_VAR_LSH_UP = 1 **and** LV_ST_END = 1 **and** LV_IGK = 1

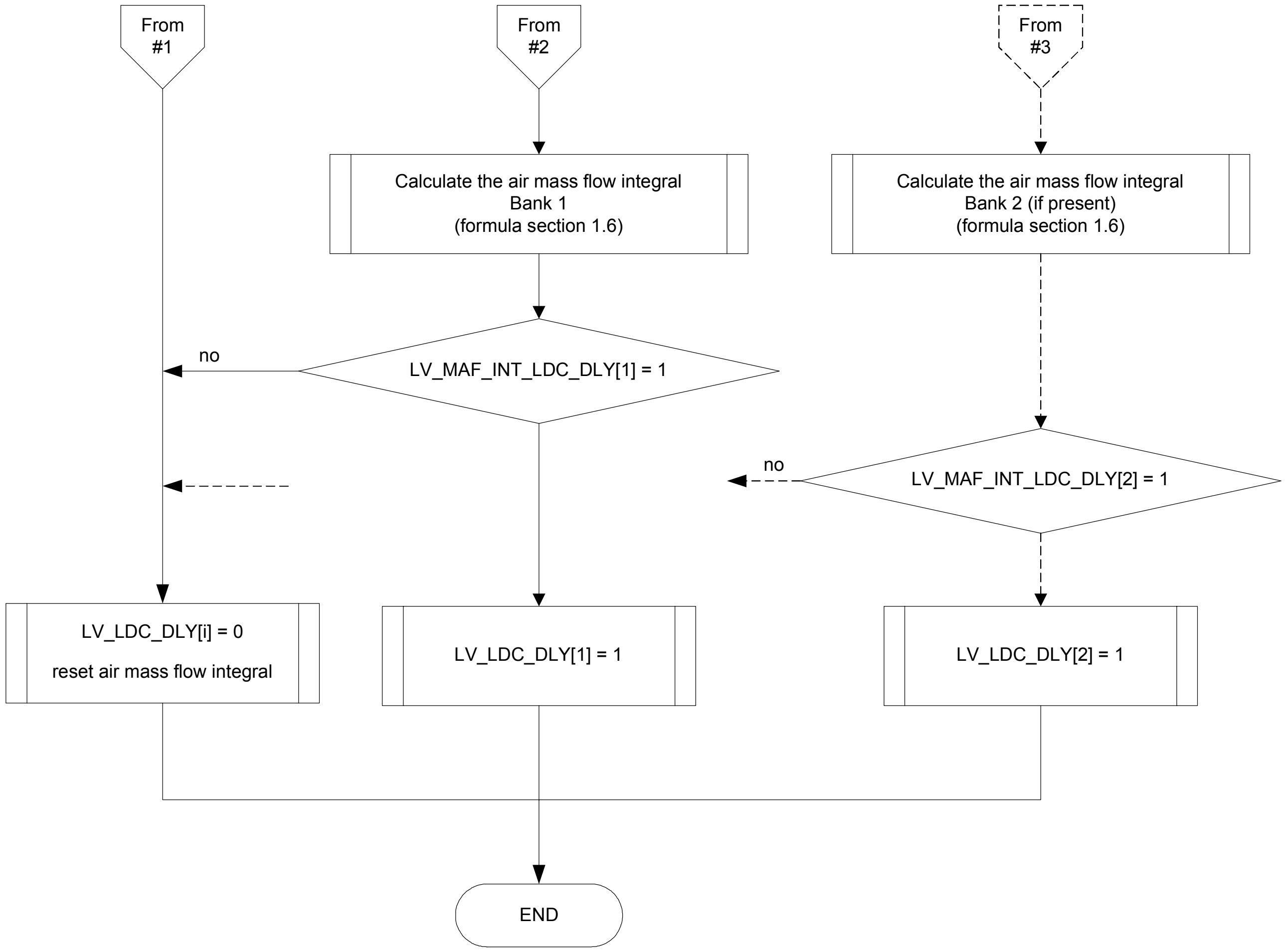
Deactivation: LV_VAR_LSH_UP = 0 **or** LV_ST_END = 0 **or** LV_IGK = 0

Initialisation at deactivation:
LV_LDC_DLY[i] = 0
LV_N_LDC_DLY = 0
LV_MAF_LDC_DLY = 0
MAF_INT_LDC_DLY[i] = 0
N_MMV_DLY = N
MAF_MMV_DLY = MAF
FAC_LAM_MV_MMV_DLY[i] = FAC_LAM_MV[i]
FAC_TQ_REQ_GRD_SUM_DLY = 0

All ring buffer registers shall be initialized with the same value, namely the value of FAC_TQ_REQ at the moment of activation.

Signal flow diagram:





Formula section:

46.11.1 Calculation of the factor torque request gradient FAC_TQ_REQ_GRD

At every recurrence the variable FAC_TQ_REQ shall be written into a ring buffer containing the last four FAC_TQ_REQ values. The gradient of FAC_TQ_REQ (FAC_TQ_REQ_GRD) is calculated based on the FAC_TQ_REQ difference between the values contained in the first and last position of the ring buffer.

46.11.2 Limited engine speed (N) dynamics

The floating mean value N_MMV_DLY is computed using the averaging constant C_N_CRLC_DLY. The limited dynamics condition only exist while the engine speed N lies within the dynamics window around N_MMV_DLY.
If the above limited dynamics condition is violated, the floating mean value N_MMV_DLY is set to the current engine speed in order to reach the limited dynamics condition faster:

N_MMV_DLY = N_MMV_DLY * (1 - C_N_CRLC_DLY) + C_N_CRLC_DLY * N

```
IF | N - N_MMV_DLY | < C_N_DYW_DLY
  THEN    LV_N_LDC_DLY = 1
  ELSE    LV_N_LDC_DLY = 0
```

```
        N_MMV_DLY = N
ENDIF
```

46.11.3 Limited mass air flow (MAF) dynamics

The floating mean value MAF_MMV_DLY is computed using the averaging constant C_MAF_CRLC_DLY. The limited dynamics condition only exist while the air mass MAF stays within the dynamics window around MAF_MMV_DLY.
If the above limited dynamics condition is violated, the floating mean value MAF_MMV_DLY is set to the current air-mass value in order to reach the limited dynamics condition faster:

```
IF LV_N_LDC_DLY = 1
THEN
    MAF_MMV_DLY = MAF_MMV_DLY * (1 - C_MAF_CRLC_DLY) +
                            C_MAF_CRLC_DLY * MAF

    IF | MAF - MAF_MMV_DLY | < C_MAF_DYW_DLY
      THEN        LV_MAF_LDC_DLY = 1
      ELSE        LV_MAF_LDC_DLY = 0
                  MAF_MMV_DLY = MAF
    ENDIF
ELSE
    MAF_MMV_DLY = MAF
    LV_MAF_LDC_DLY = 0
    LV_LDC_DLY[i] = 0
    MAF_INT_LDC_DLY[i] =  0
ENDIF
```

46.11.4 Limited requested torque factor gradient (FAC_TQ_REQ_GRD) dynamics

The FAC_TQ_REQ_GRD reflects the dynamic of the actual driver's demand prior to the corrections and is calculated with the recurence T_SAMPLE in [ms].

If C_FAC_TQ_REQ_GRD_SUM_MIN is larger than C_FAC_TQ_REQ_GRD_SUM_MAX the FAC_TQ_REQ gradient summation FAC_TQ_REQ_GRD_SUM_LDC_DLY will be no condition to determine the limited dynamic bit LV_LDC_DLY[i].

```
IF(3) LV_MAF_LDC_DLY = 1

THEN(3)

    IF(3a)   |FAC_TQ_REQ_GRD| ≥ C_FAC_TQ_REQ_GRD_MIN_LDC_DLY
    THEN(3a)       FAC_TQ_REQ_GRD_SUM_DLY$_{n+1}$ = FAC_TQ_REQ_GRD_SUM_DLY$_{n}$
                                                   +  |FAC_TQ_REQ_GRD|
                % Incrementation of FAC_TQ_REQ_GRD_SUM_DLY if FAC_TQ_REQ_GRD exceeds a threshold
    ENDIF(3a)

    T_FAC_TQ_REQ_GRD_SUM_DLY$_{n}$  =
                                T_FAC_TQ_REQ_GRD_SUM_DLY$_{n-1}$ + 1

    IF(3b)  T_FAC_TQ_REQ_GRD_SUM_DLY ≥
                            C_T_FAC_TQ_REQ_GRD_SUM_DLY
    THEN(3b)
```

```
            FAC_TQ_REQ_GRD_SUM_DLY$_n$   =   FAC_TQ_REQ_GRD_SUM_DLY$_{n-1}$ –
                                                 C_FAC_TQ_REQ_GRD_DEC_DLY
              % The decrementation by C_FAC_TQ_REQ_GRD_DEC_DLY is done every
              % C_T_FAC_TQ_REQ_GRD_SUM_DLY intervalls

            T_FAC_TQ_REQ_GRD_SUM_DLY  =  0
        ENDIF(3b)

        IF(3c)  C_FAC_TQ_REQ_GRD_SUM_MIN_DLY <
                                        C_FAC_TQ_REQ_GRD_SUM_MAX_DLY
        THEN(3c)     % FAC_TQ_REQ_GRD_SUM_DLY is condition for limited dynamic bit
            IF(4)    LV_FAC_TQ_REQ_LDC_DLY = 0
            THEN(4)          %  limited dynamic bit deactive, activation possible
                IF(5) FAC_TQ_REQ_GRD_SUM_DLY ≤
                                        C_FAC_TQ_REQ_GRD_SUM_MIN_DLY
                THEN(5)       % activation of limited dynamic bit
                    LV_FAC_TQ_REQ_LDC_DLY   =   1
                ENDIF(5)
            ELSE(4)          % limited dynamic bit active, deactivation possible

                IF(6) FAC_TQ_REQ_GRD_SUM_DLY >
                                          C_FAC_TQ_REQ_GRD_SUM_MAX_DLY
                THEN(6)       % deactivation of limited dynamic bit
                    LV_FAC_TQ_REQ_LDC_DLY  =   0
                ENDIF(6)
            ENDIF(4)
        ELSE(3c)
            LV_FAC_TQ_REQ_LDC_DLY  =   1
        ENDIF(3c)
    ELSE(3)
        FAC_TQ_REQ_GRD_SUM_DLY     =  0
        T_FAC_TQ_REQ_GRD_SUM_DLY =  0
        LV_FAC_TQ_REQ_LDC_DLY  =   0
        LV_LDC_DLY[i] = 0
        MAF_INT_LDC_DLY[i] =  0
    ENDIF(3)
```

46.11.5 Limited mean lambda controller output (FAC_LAM_MV[i]) dynamics

The floating mean value of the mean lambda controller output FAC_LAM_MV_MMV_DLY[i] is computed by using the averaging constant C_FAC_LAM_MV_CRLC_DLY. The limited dynamics condition is present while FAC_LAM_MV[i] stays within the dynamics window around FAC_LAM_MV_MMV_DLY[i]. If this condition violated, the correspondent flag is reset indicating a dynamic condition is present and the floating mean value FAC_LAM_MV_MMV_DLY[i] is set to the current value in order to reach the limited dynamics condition faster.

```
IF LV_FAC_TQ_REQ_LDC_DLY   =   1
THEN
    FAC_LAM_MV_MMV_DLY[i]  =
        FAC_LAM_MV_MMV_DLY[i] * (1 - C_LAM_MV_CRLC_DLY) +
                                  C_LAM_MV_CRLC_DLY * FAC_LAM_MV[i]

    IF | FAC_LAM_MV[i] - FAC_LAM_MV_MMV_DLY[i] | < C_LAM_MV_DYW_DLY

    THEN  LV_LAM_MV_LDC_DLY[i] = 1
```

```
        ELSE   LV_LAM_MV_LDC_DLY[i] = 0
               FAC_LAM_MV_MMV_DLY[i]  = FAC_LAM_MV[i]
        ENDIF
    ELSE
        FAC_LAM_MV_MMV_DLY[i]  = FAC_LAM_MV[i]
        LV_LAM_MV_LDC_DLY[i] = 0
        LV_LDC_DLY[i] = 0
        MAF_INT_LDC_DLY[i] =  0
    ENDIF
```

46.11.6 Calculation of mass air flow integral

```
    IF LV_LAM_MV_LDC_DLY[i] = 1
    THEN
        IF     $MAF\_INT\_LDC\_DLY[i] \leq C\_MAF\_INT\_MIN\_LDC\_DLY$
        THEN
               LV_LDC_DLY[i]   =  0
               $MAF\_INT\_LDC\_DLY[i]_n = MAF\_INT\_LDC\_DLY[i]_{n-1} +$
                                         $MAF\_KGH * NC\_FAC\_MAF\_INT\_20$
        ELSE
               LV_LDC_DLY[i]   =  1
        ENDIF
    ELSE
        LV_LDC_DLY[i]   =  0
        MAF_INT_LDC_DLY[i] =  0
    ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_CRLC_DLY	1	0...FFFFH	0...0.99998	1.53E-05	[-]
correlation constant floating mean value calculation engine speed					
C_MAF_CRLC_DLY	1	0...FFFFH	0...0.99998	1.53E-05	[-]
correlation constant floating mean value calculation MAF					
C_LAM_MV_CRLC_DLY	1	0...FFFFH	0...0.99998	1.53E-05	[-]
correlation constant floating mean value calculation lambda					
C_N_DYW_DLY	1	0...1FE0H	0...8160	1	[rpm]
threshold limited dynamic engine speed					
C_MAF_DYW_DLY	1	0...FFFFH	0...1389	2.12E-02	[mg/stk]
threshold limited dynamic engine MAF					
C_LAM_MV_DYW_DLY	1	0...FFFFH	0...99.99847	1.53E-03	[%]
threshold limited dynamic lambda					
C_FAC_TQ_REQ_GRD_DEC_DLY	1	0...FFFFH	0...1.999969	3.05176E-05	[-]
decrement for calculation PV_AV gradient					
C_FAC_TQ_REQ_GRD_MIN_LDC_DLY	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
minimum value of FAC_TQ_REQ_GRD to increment FAC_TQ_REQ_GRD_SUM_DLY					
C_FAC_TQ_REQ_GRD_SUM_MAX_DLY	1	0...FFFFH	0...1.999969	3.05176E-05	[-]
threshold to reset the limited dynamic flag LV_LDC_DLY[i] due to present high FAC_TQ_REQ_GRD condition					
C_FAC_TQ_REQ_GRD_SUM_MIN_DLY	1	0...FFFFH	0...1.999969	3.05176E-05	[-]
threshold to set the limited dynamic flag LV_LDC_DLY[i] when no high FAC_TQ_REQ_GRD condition is met					
C_T_FAC_TQ_REQ_GRD_SUM_DLY	1	0...FFH	0...5.1	0.02	[s]
time periods for calculation limited dynamic FAC_TQ_REQ_GRD					
C_MAF_INT_MIN_LDC_DLY	1	0...FFFFH	0...1820.42	2.78E-02	[g]
MAF integral after setting limited dynamic conditions LV_LDC_DLYi before starting the monitoring cycle					

46.12 Lambda Adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_LAM_ADJ_LAM_AD[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
output value from lambda adaptation for the lambda controller shift					
FAC_LAM_AD_LAM_OUT[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
fuel mass set point factor, output from lambda adaptation, input into lambda controller					
FAC_MFF_ADD_FAC_LAM_AD[NC_CBK_EX_NR]	O/V/S	8000...7FFFH	-100...99.996948	0.00305176	%
lambda adaptation correction for scan tool (factor and relative offset)					
FAC_MFF_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	O/V	8000...7FFFH	-100...99.996948	0.00305176	%
relative lambda adaptation offset quotient					
LV_FAC_H_RNG_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating that lambda adaptation in factor learning upper field is active					
LV_FAC_H_RNG_LIM_MAX_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating top limitation of lambda adaptation factor of upper area					
LV_FAC_H_RNG_LIM_MIN_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating bottom limitation of lambda adaptation factor of upper area					
LV_FAC_LAM_ADJ_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag to request lambda controller shift					
LV_FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating that lambda adaptation in factor learning lower field is active					
LV_FAC_L_RNG_LIM_MAX_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating top limitation of lambda adaptation factor of lower area					
LV_FAC_L_RNG_LIM_MIN_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating bottom limitation of lambda adaptation factor of lower area					
LV_LAM_AD_AFS_REQ	O/V	0...1H	0...1	1	-
request flag for combustion manager to force lambda eq. 1 conditions for lambda adaptation					
LV_LAM_AD_CDN	O/V	0...1H	0...1	1	-
flag for time scheduler indicating good conditions for lambda adaptation					
LV_LAM_AD_END	O/V	0...1H	0...1	1	-
logical value indicating temporary end of lambda adaptation					
LV_MFF_ADD_LIM_MAX_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating top limitation of lambda adaptation offset					
LV_MFF_ADD_LIM_MIN_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating bottom limitation of lambda adaptation offset					
LV_MFF_ADD_RNG_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag indicating that lambda adaptation in offset learning field is active					
MFF_LAM_ADD_LAM_AD_OUT[NC_CBK_EX_NR]	O/V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
fuel mass set point offset, output from lambda adaptation, input into lambda controller					
T_PRI_TOT_LAM_AD	O/V	0...FFFFH	0...6.5535E+3	0.1	s
minimum priority time of all exhaust cylinder banks for next requested lambda adaptation					
FAC_LAM_AD[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
fuel mass set point factor of lambda adaptation, interpolation of both factor areas (high and low field)					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_LAM_AD[NC_CBK_EX_NR]	O/V	0H 1H 2H 3H 4H 5H 6H 7H	INIT WAIT CDN_FAC_L CDN_FAC_H CDN_ADD ADAPT_FAC_L ADAPT_FAC_H ADAPT_ADD	1	-
state of lambda adaptation					
FAC_H_RNG_LAM_AD[NC_CBK_EX_NR]	O/V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
fuel mass set point factor of high field, stored value of lambda adaptation					
FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	O/V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
fuel mass set point factor of low field, stored value of lambda adaptation					
LV_LAM_AD_STOP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
logical variable indicating stop of the lambda adaptation cycle					
MFF_ADD_LAM_AD[NC_CBK_EX_NR]	O/V/S	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
fuel mass set point offset, stored value of lambda adaptation					
FAC_DELTA_L_RNG_LAM_AD[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
difference of lambda adaptation factor of low field to previous adaptation					
LV_LAM_AD_END_CBK[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
logical value indicating temporary end of lambda adaptation (bank specific flag)					
LV_FAC_LAM_AD_SHIFT_END[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Shift of Lambda adaptation was carried out					
MFF_DELTA_ADD_LAM_AD[NC_CBK_EX_NR]	V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
difference of lambda adaptation offset to previous adaptation					
T_WAIT_LAM_AD[NC_CBK_EX_NR]	V	0...FFH	0...5.1	0.02	s
waiting time between two adaptations					
FAC_DELTA_H_RNG_LAM_AD[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
difference of lambda adaptation factor of high field to previous adaptation					
LV_FAC_L_RNG_LAM_AD_INI[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Flag indicating that low range Lambda adaptation is initialized with value of high range					
MFF_DELTA_ADD_LAM_AD_H_RES[NC_CBK_EX_NR]	V	8000...7FFFH	-10.85 ... 10.8496689	3.31116E-4	mg/stk
difference integral with high resolution of lambda adaptation offset					
T_PRI_LAM_AD[NC_CBK_EX_NR]	V	0...FFFFH	0...6.5535E+3	0.1	s
priority time for next requested lambda adaptation					
LV_LAM_AD_STOP_CBK_EX	V	0...1H	0...1	1	-
logical variable indicating stop of at least one bank of the lambda adaptation cycle					
LV_FAC_H_RNG_LAM_AD_INI[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Flag indicating that high range Lambda adaptation is initialized with value of low range					

Input data:

C_TCO_MIN_LAM_AD	CRLC_LAM_AD[NC_CBK_EX_NR]	LV_LAM_AD_ACT[NC_CBK_EX_NR]	LV_LAM_AD_DEAC_ERR[NC_CBK_EX_NR]
T_WAIT_MAX_LAM_AD[NC_CBK_EX_NR]	MFF_SP_AD[NC_CBK_EX_NR]	LV_LAM_AD_CDN_ADD[NC_CBK_EX_NR]	LV_LAM_AD_CDN_L_RNG[NC_CBK_EX_NR]
LV_LAM_AD_CDN_H_RNG[NC_CBK_EX_NR]	FAC_LAM_AD_SHIFT[NC_CBK_EX_NR]	LV_FAC_LAM_AD_SHIFT[NC_CBK_EX_NR]	LV_LAM_STOP[NC_CBK_EX_NR]
FAC_LAM_MV_MMV[NC_CBK_EX_NR]	LV_FAC_LAM_ADJ_LAM_AD_END[NC_CBK_EX_NR]	LV_IGK	LV_LAM_AD_ENA

LV_ST_END	MFF_SP[NC_CBK_EX_NR]	NC_CBK_EX_NR	N_32
TCO	LV_FAC_LAM_ADJ_LAM_AD_WUP[NC_CBK_EX_NR]	FAC_LAM_ADJ_LAM_AD_WUP[NC_CBK_EX_NR]	FAC_LAM_AD_WUP[NC_CBK_EX_NR]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_MAX_H_RNG_LAM_AD	1	8000...7FFFH	-50...49.9984741	0.00152588	%
maximum value of upper area lambda adaptation factor					
C_FAC_MAX_L_RNG_LAM_AD	1	8000...7FFFH	-50...49.9984741	0.00152588	%
maximum value of lower area lambda adaptation factor					
C_FAC_MIN_H_RNG_LAM_AD	1	8000...7FFFH	-50...49.9984741	0.00152588	%
minimum value of upper area lambda adaptation factor					
C_FAC_MIN_L_RNG_LAM_AD	1	8000...7FFFH	-50...49.9984741	0.00152588	%
minimum value of lower area lambda adaptation factor					
C_MFF_MAX_ADD_RNG_LAM_AD	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
maximum value of lambda adaptation offset					
C_MFF_MIN_ADD_RNG_LAM_AD	1	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
minimum value of lambda adaptation offset					
C_MFF_SP_TOL_ADD_RNG_LAM_AD	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
maximum fuel mass set point for offset lambda adaptation					
C_N_BOL_FAC_H_RNG_LAM_AD	1	0...FFH	0...8.16E+3	32	rpm
minimum engine speed for upper area factor lambda adaptation					
C_N_BOL_FAC_L_RNG_LAM_AD	1	0...FFH	0...8.16E+3	32	rpm
minimum engine speed for lower area factor lambda adaptation					
C_N_TOL_ADD_RNG_LAM_AD	1	0...FFH	0...8.16E+3	32	rpm
maximum engine speed for offset lambda adaptation					
C_N_TOL_FAC_H_RNG_LAM_AD	1	0...FFH	0...8.16E+3	32	rpm
maximum engine speed for upper area factor lambda adaptation					
C_N_TOL_FAC_L_RNG_LAM_AD	1	0...FFH	0...8.16E+3	32	rpm
maximum engine speed for lower area factor lambda adaptation					
C_T_PRI_MAX_LAM_AD	1	0...FFFFH	0...6.5535E+3	0.1	s
maximum priority time to set temporary end of adaptation flag					
LC_FAC_H_RNG_AFS_REQ_LAM_AD	1	0...1H	0...1	1	-
logical calibration to allow forced lambda eq. 1 conditions for factor adaptation at upper area					
LC_FAC_LAM_AD_INI_ACT	1	0...1H	0...1	1	-
logical calibration to allow initialization of adaptation factor from other range					
LC_FAC_L_RNG_AFS_REQ_LAM_AD	1	0...1H	0...1	1	-
logical calibration to allow forced lambda eq. 1 conditions for factor adaptation at upper area					
LC_MFF_ADD_RNG_AFS_REQ_LAM_AD	1	0...1H	0...1	1	-
logical calibration to allow forced lambda eq. 1 conditions for offset adaptation					
ID_MFF_SP_BOL_FAC_H_RNG_LAM_AD	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDP_N_32_ID_MFF_SP_BOL_H_RNG	8	0...FFH	0...8.16E+3	32	rpm
minimum fuel mass set point for upper area factor lambda adaptation					
ID_MFF_SP_BOL_FAC_L_RNG_LAM_AD	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDP_N_32_ID_MFF_SP_BOL_L_RNG	8	0...FFH	0...8.16E+3	32	rpm
minimum fuel mass set point for lower area factor lambda adaptation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_MFF_SP_TOL_FAC_H_RNG_LAM_AD	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDP_N_32_ID_MFF_SP_TOL_H_RNG	8	0...FFH	0...8.16E+3	32	rpm
maximum fuel mass set point for upper area factor lambda adaptation					
ID_MFF_SP_TOL_FAC_L_RNG_LAM_AD	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDP_N_32_ID_MFF_SP_TOL_L_RNG	8	0...FFH	0...8.16E+3	32	rpm
maximum fuel mass set point for lower area factor lambda adaptation					
ID_T_PRI_FAC_H_RNG_LAM_AD	8	0...FFFFH	0...6.5535E+3	0.1	s
LDP_FAC_DELTA_H_RNG_LAM_AD_PRI	8	0...FFFFH	-50...49.998474	0.0015259	%
priority time for next requested lambda adaptation according to factor difference or upper area					
ID_T_PRI_FAC_L_RNG_LAM_AD	8	0...FFFFH	0...6.5535E+3	0.1	s
LDP_FAC_DELTA_L_RNG_LAM_AD_PRI	8	0...FFFFH	-50...49.998474	0.0015259	%
priority time for next requested lambda adaptation according to factor difference or lower area					
ID_T_PRI_MFF_ADD_RNG_LAM_AD	8	0...FFFFH	0...6.5535E+3	0.1	s
LDP_MFF_DELTA_ADD_LAM_AD_T_PRI	8	0...FFFFH	-694.510597 ... 694.489403	0.021195	mg/stk
priority time for next requested lambda adaptation according to offset difference					
IP_FAC_N_FAC_LAM_AD	6	0...80H	0...1	0.0078125	-
LDPM_N_32_4_LACO	6	0...FFH	0...8.16E+3	32	rpm
factor on lambda adaptation factor depending on engine speed					
IP_FAC_N_MFF_ADD_LAM_AD	6	0...80H	0...1	0.0078125	-
LDPM_N_32_4_LACO	6	0...FFH	0...8.16E+3	32	rpm
factor on lambda adaptation offset depending on engine speed					
IP_MFF_SP_MAX_FAC_L_RNG_LAM_AD	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDPM_N_32_5_LACO	8	0...FFH	0...8.16E+3	32	rpm
fuel mass set point threshold for consideration of only lower field lambda adaptation factor					
IP_MFF_SP_MIN_FAC_H_RNG_LAM_AD	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDPM_N_32_5_LACO	8	0...FFH	0...8.16E+3	32	rpm
fuel mass set point threshold for consideration of only upper field lambda adaptation factor					
IP_FAC_WUP_FAC_LAM_AD	5x5	0...80H	0...1	0.0078125	-
LDPM_TCO_2_LACO	5	0...FEH	-48...142.5	0.75	°C
LDP_FAC_LAM_AD_IP_FAC_WUP_FAC	5	0...FFFFH	-50...49.9984741	0.00152588	%
factor on lambda adaptation factor during warm up					
IP_FAC_WUP_MFF_ADD_LAM_AD	5x5	0...80H	0...1	0.0078125	-
LDPM_TCO_2_LACO	5	0...FEH	-48...142.5	0.75	°C
LDP_MFF_ADD_LAM_AD_IP_FAC_WUP	5	0...FFFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
factor on lambda adaptation offset during warm up					

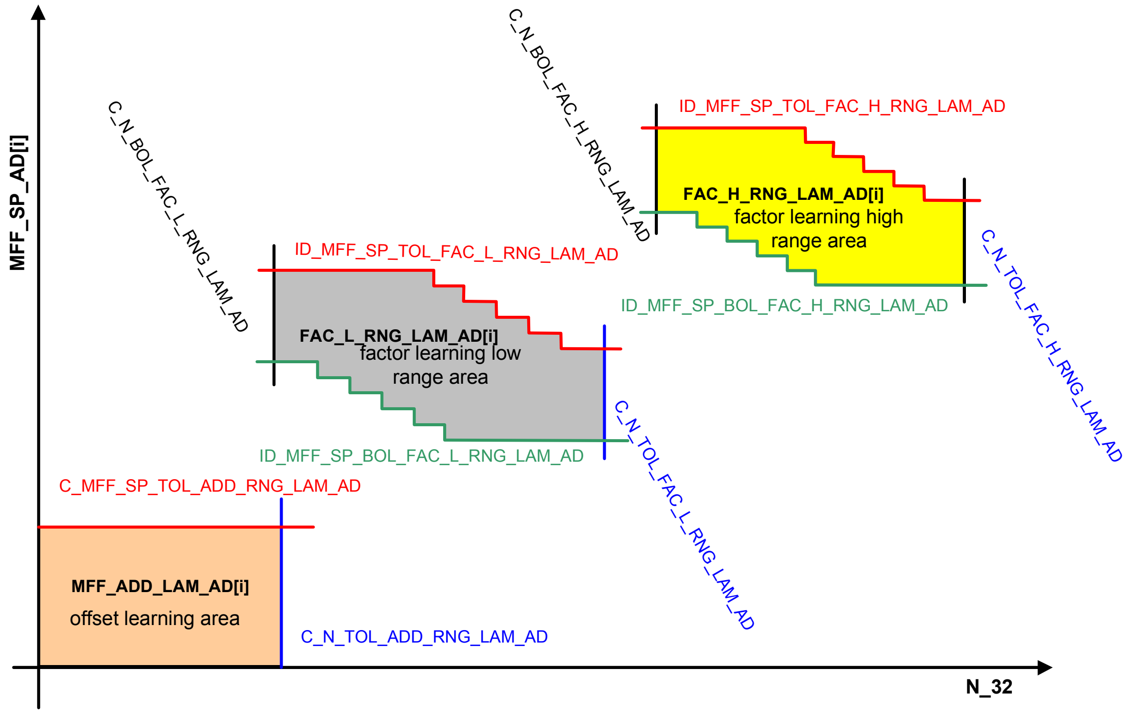
46.12.1 General Information

In order to compensate serial production tolerances of components, an adaptive correction is calculated (one additive and two multiplicative) based on the filtered lambda controller output.

The additive and multiplicative adaptation corrections are used for calculating the injection time for all engine operating states, except at ‘engine stop’ and ‘engine start’.

Lambda adaptation, for precision reason, needs to be performed at lambda equal 1 conditions. The function itself is activated by the corresponding LV_LAM_AD_ACT[i].

Depending on load and engine speed three different adaptation fields (one offset and two factor areas) are observed.



Adaptation field diagram

According to project philosophy homogeneous mode and suitable conditions can be forced by setting:

- LC_MFF_ADD_RNG_AFS_REQ_LAM_AD,
- LC_FAC_L_RNG_AFS_REQ_LAM_AD and / or
- LC_FAC_H_RNG_AFS_REQ_LAM_AD.

Furthermore this function is used by the lambda adaptation in warm-up phase to provide an unique interface to lambda controller and fuel mass set point calculation.

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

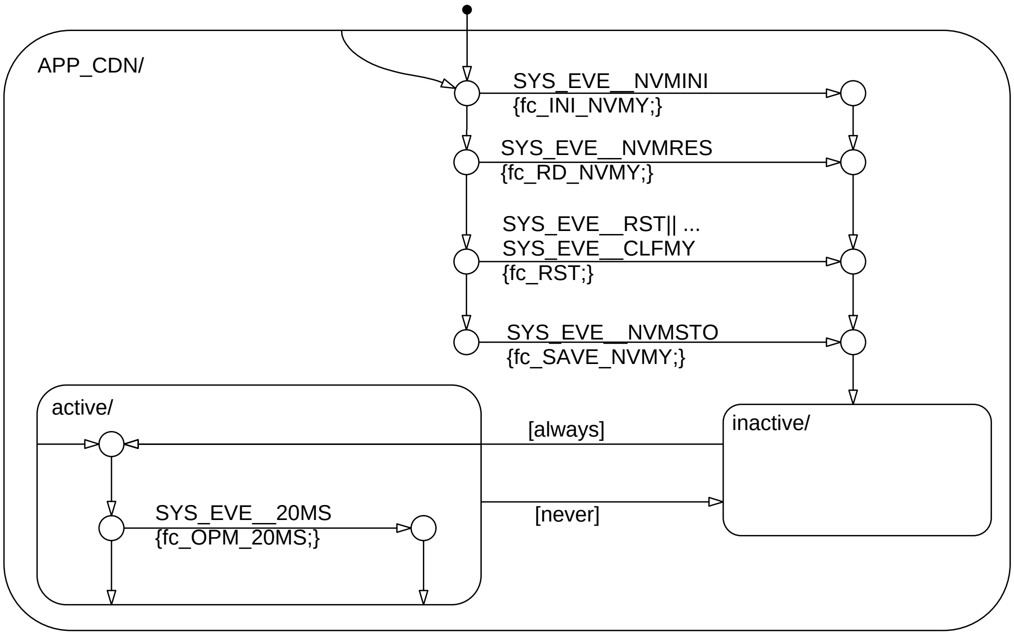


Figure 87 LACO_ISPCLADA0/ APP_CDN/ Chart

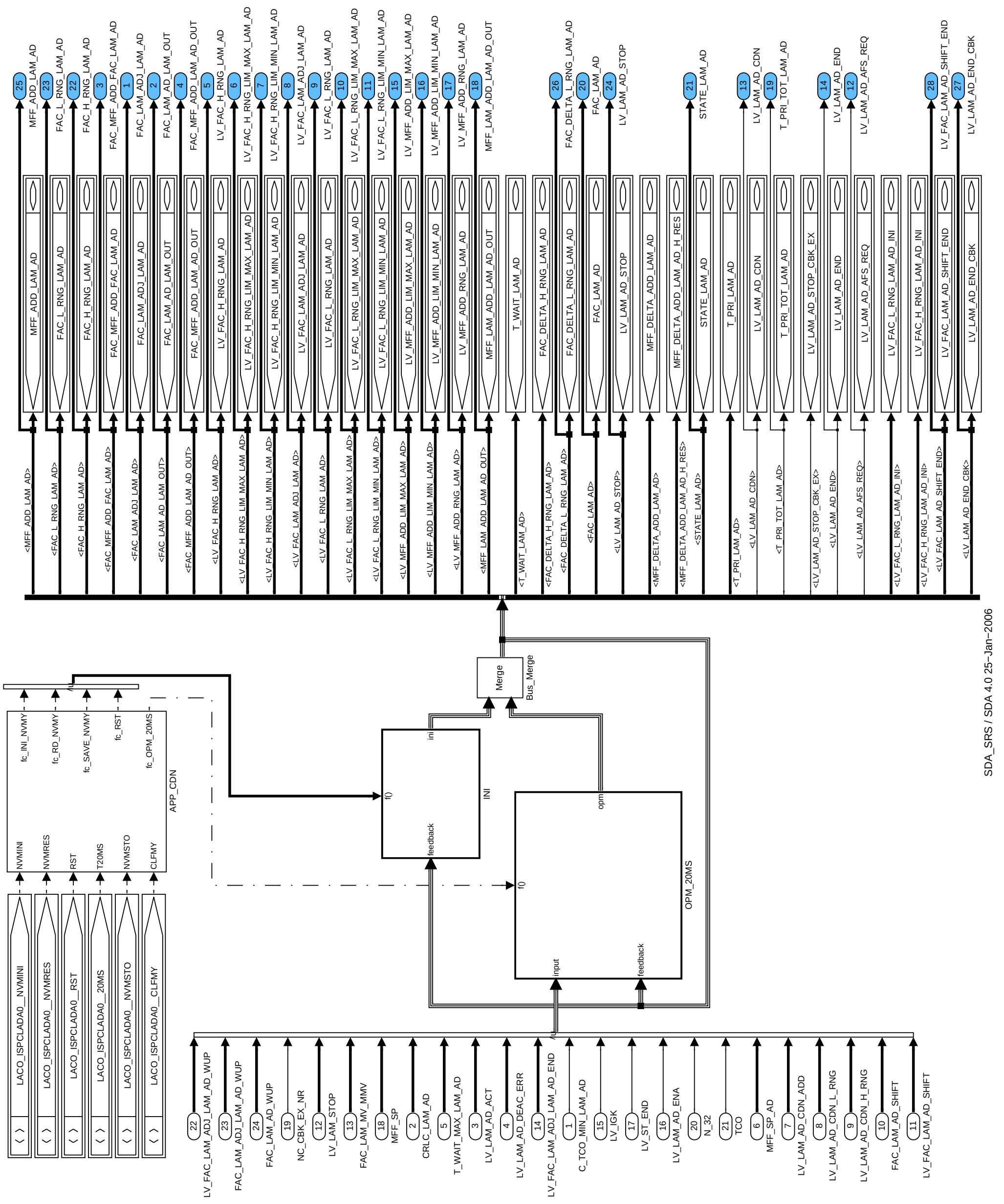


Figure 88 LACO_ISPCLADA0

46.12.1.1 Initialization and Non Volatile Memory Data Handling

ECU Reset

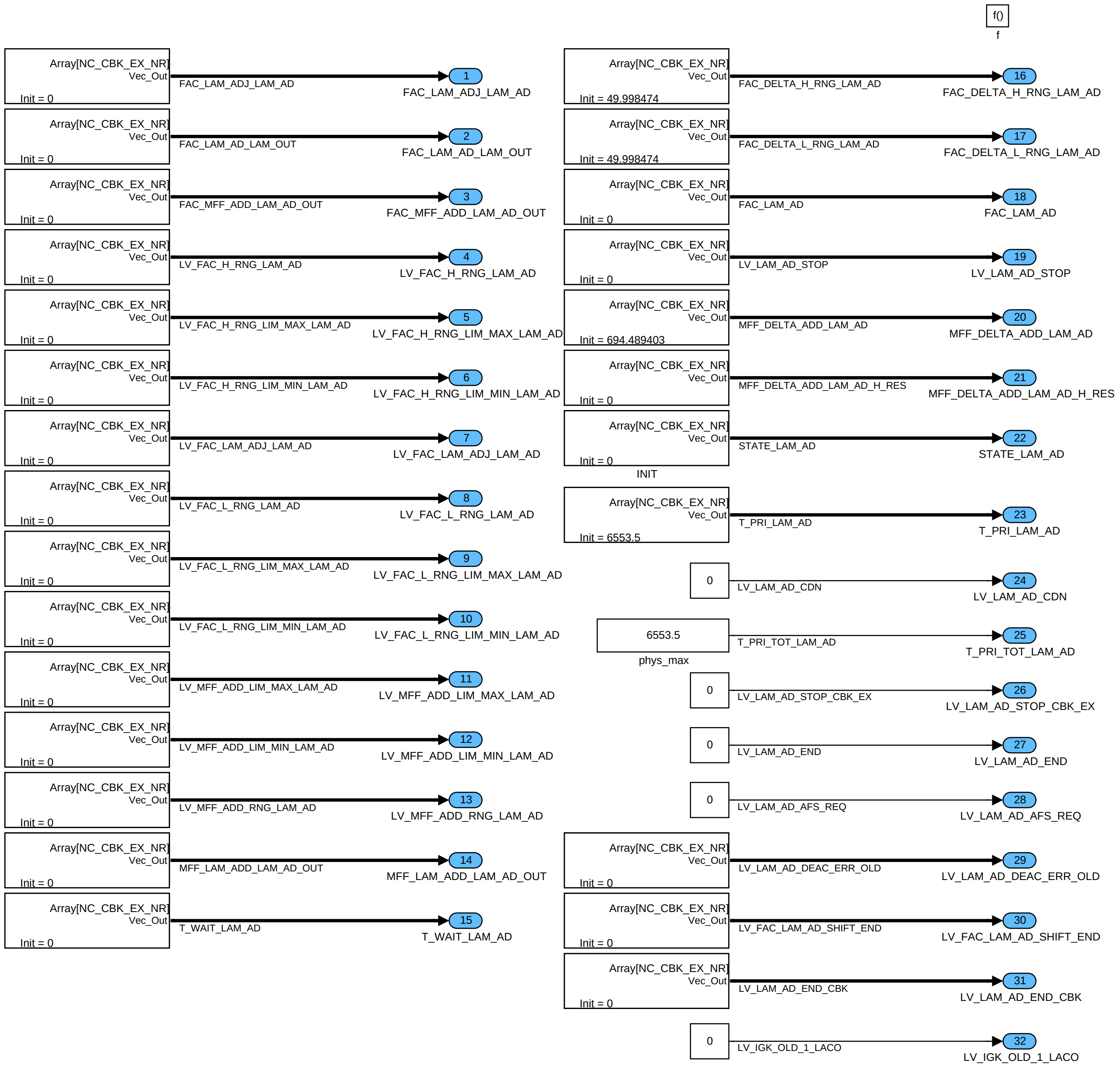


Figure 89 LACO_ISPCLADA0/ INI/ RST

Non-Volatile Memory Data

MFF_ADD_LAM_AD[i], FAC_L_RNG_LAM_AD[i], FAC_H_RNG_LAM_AD[i] and FAC_MFF_ADD_FAC_LAM_AD[i] are stored in the NVMY (at the end of PWL phase) and read out at ECU reset.

If the ECU is brand new or in case of EEPROM error these data are set to the default value 0.

Furthermore the variables MFF_ADD_LAM_AD_TMP[i], FAC_L_RNG_LAM_AD_TMP[i] and FAC_H_RNG_LAM_AD_TMP[i] are used to store the initial values of MFF_ADD_LAM_AD[i], FAC_L_RNG_LAM_AD[i] and FAC_H_RNG_LAM_AD[i] during the whole driving cycle for potential use.

If the low range or the high adaptation value is zero indicating that the adaptation value was not yet learned since the last reset of the NVMY data, LV_FAC_L_RNG_LAM_AD_INI[i] respectively LV_FAC_H_RNG_LAM_AD_INI[i] are set to one.

46.12.1.2 Formula Section

The main functionality is executed by means of a FOR loop over the number of exhaust banks. Only the determination of the priority information for the priority manager between canister purge and lambda adaptation is independent of the number of exhaust banks and executed after the FOR loop.

For Loop over Number of Exhaust Banks

Initialization of FOR Loop

Set Variables

Before executing the for loop variables are initialized.

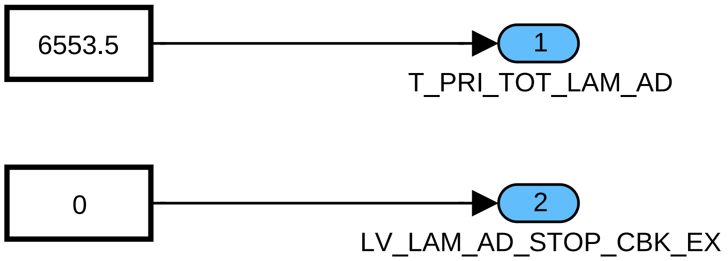


Figure 90 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ FB_INI_FLP/ INI_FLP

State Machine Management and State Independent Calculations

Manage Lambda Adaptation State Machine

Determine Calibration Values Used inside State Machine

The calibration data used to determine the current adaptation field are evaluated.

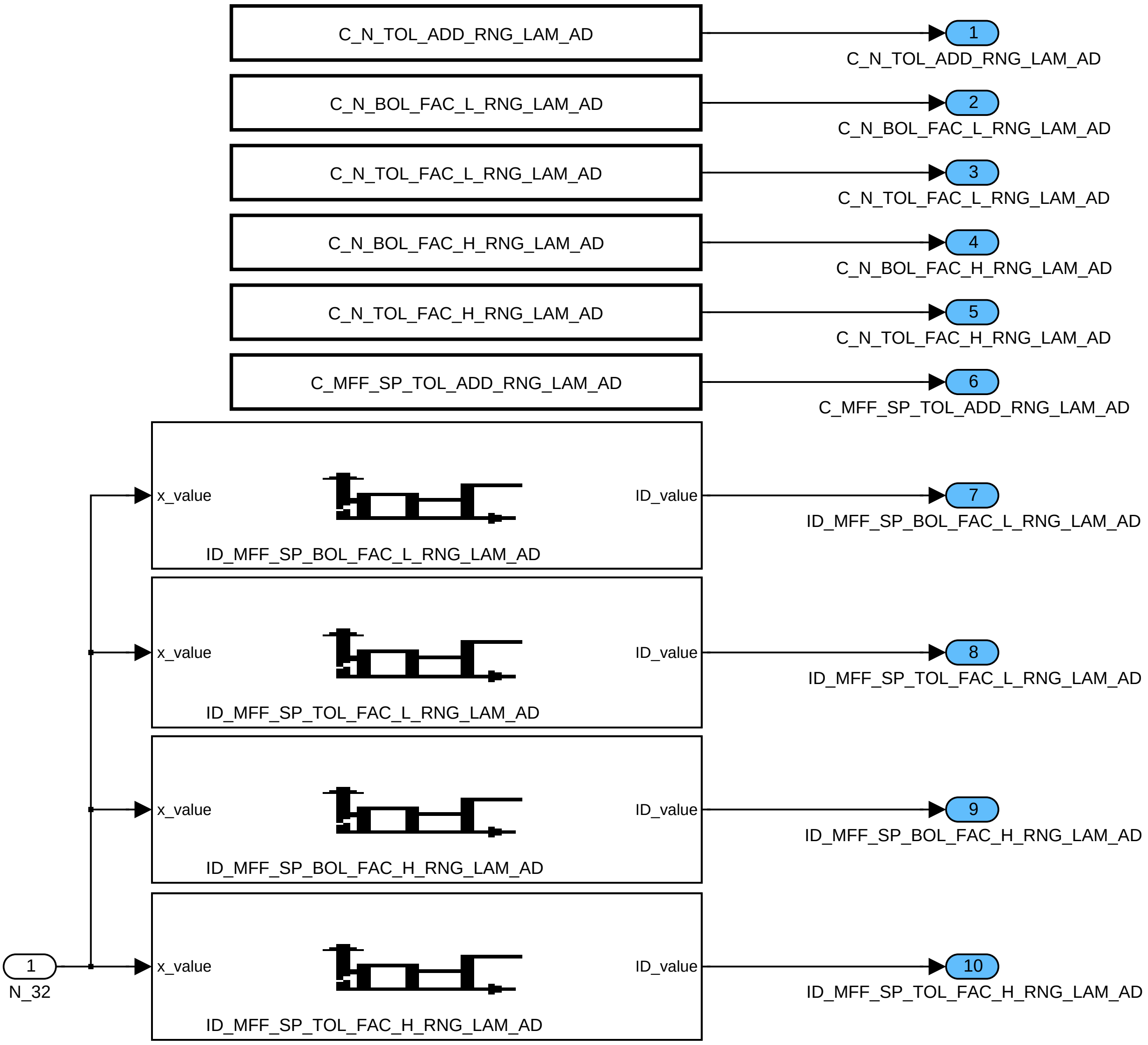


Figure 91 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ DET_STATE_LAM_AD/ SET_CAL_VALUE

State Machine

If the lambda adaptation is not active the state INIT is applied.

If the engine is not running in one of the adaptation fields the state WAIT is applied.

Depending on fuel mass flow and engine speed limits the respective states are entered.

If the respective condition bit LV_LAM_AD_CDN_ADD; LV_LAM_AD_CDN_L_RNG or LV_LAM_AD_CDN_H_RNG is set, the adaptation state ADAPT_ADD, ADAPT_FAC_L or ADAPT_FAC_H is applied. Otherwise one of the condition states is applied (CDN_ADD, CDN_FAC_L or CDN_FAC_H).

Furthermore at each transition to one of the adaptation sates the counter CTR_RAF_CHG_LAM_AD[i] is set to 0.

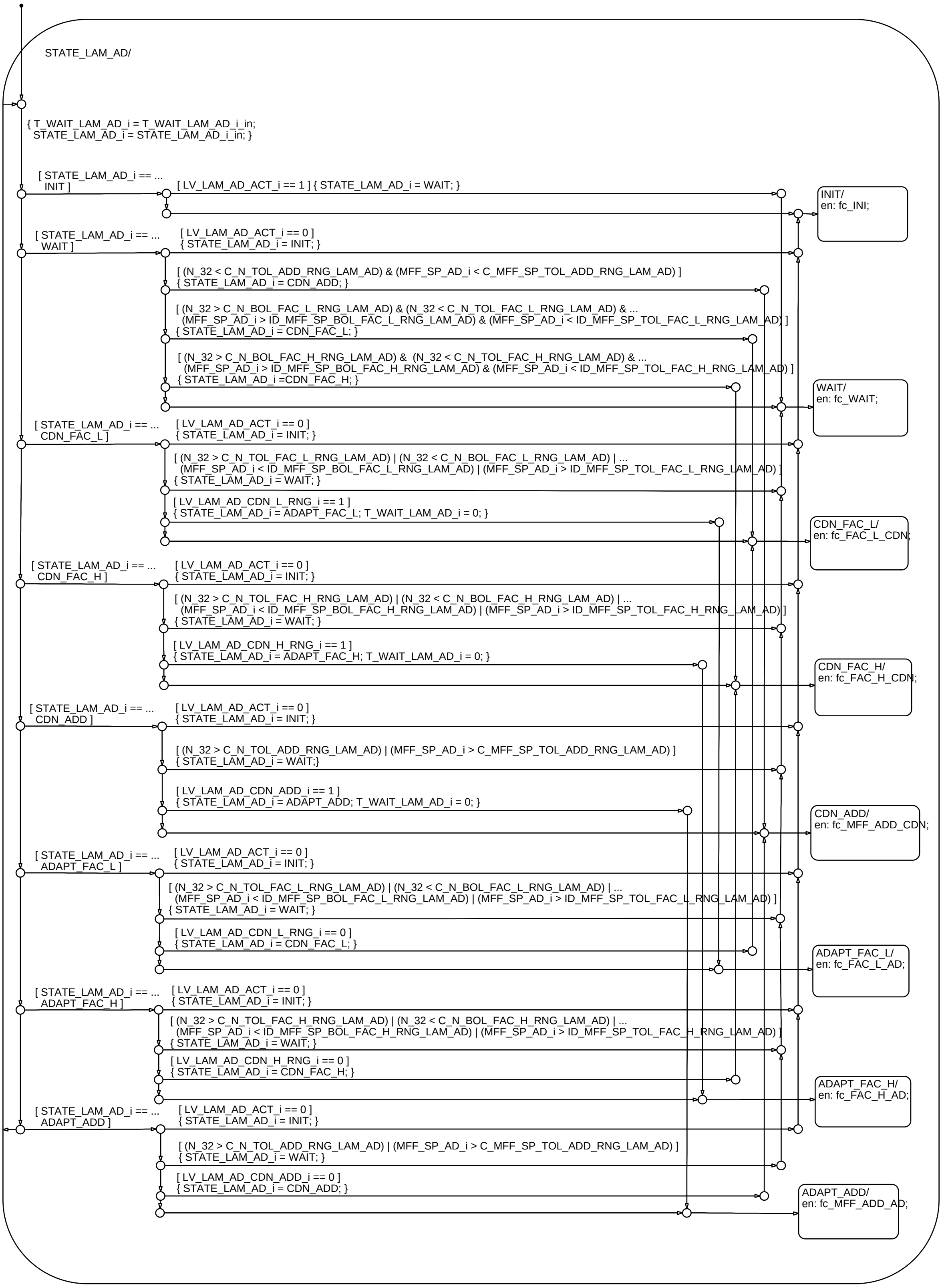


Figure 92 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ DET_STATE_LAM_AD/ STATE_LAM_AD/ STATE_LAM_AD

State Actions

The following flags are set in any state.

LV_LAM_AD_CDN is indicating that the conditions for adaptation are fulfilled (used by priority manager between canister purge and lambda adaptation).

LV_LAM_AD_END sets the temporary end of lambda adaptation (to indicate to the priority manager that the priority for lambda adaptation is low).

LV_LAM_AD_AFS_REQ (if set to 1) requests stoichiometric mixture.

LV_MFF_ADD_RNG_LAM_AD[i], LV_FAC_L_RNG_LAM_AD[i] and LV_FAC_H_RNG_LAM_AD[i] are indicating that the lambda adaptation is active and running in the respective adaptation field.

State “INIT”

Set Variables

The general flags are set to 0. All other variables remain unchanged.

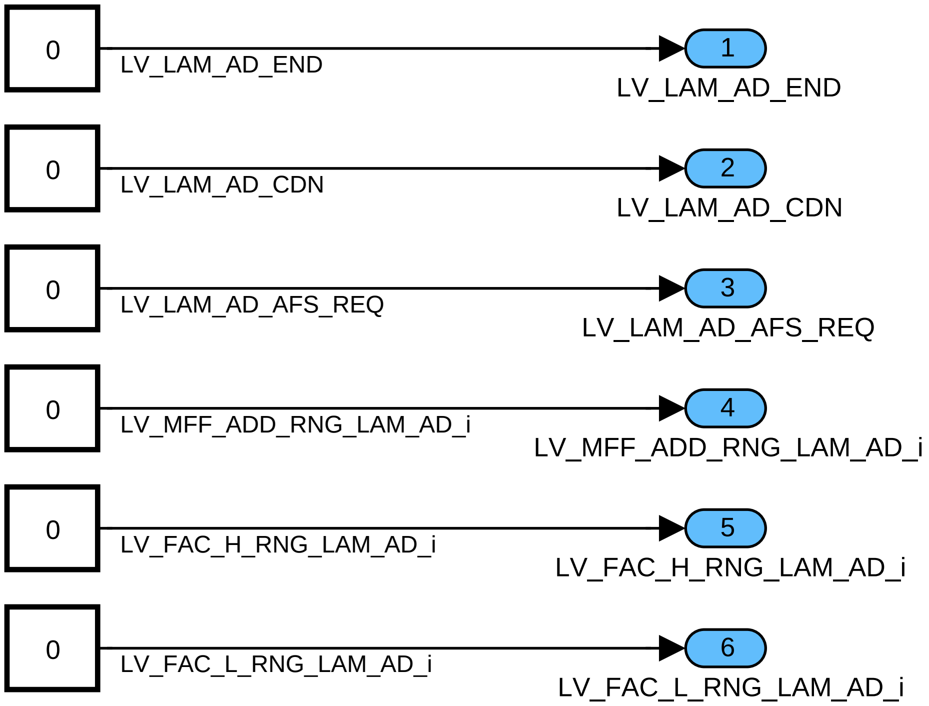


Figure 93 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ INI/ SET_VALUE

State “WAIT”

Set Variables

The general flags are set to 0. The time T_PRI_LAM_AD[i] is set to the maximum value of the 3 adaptation fields. All other variables remain unchanged.

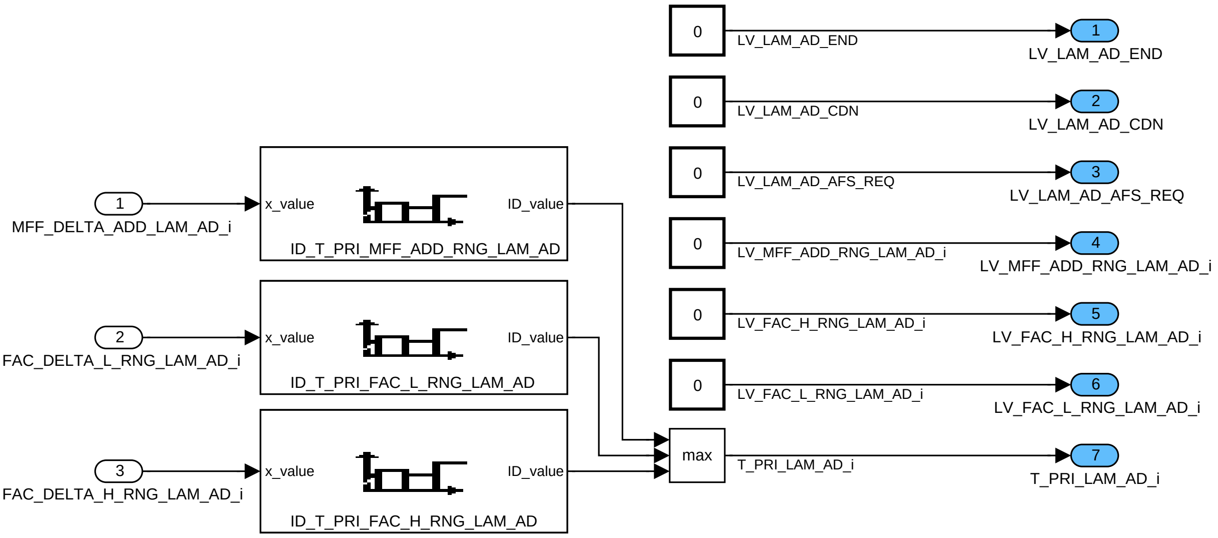


Figure 94 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ WAIT/ SET_VALUE

State “CDN_ADD”

Set Variables

LV_LAM_AD_CDN is set to 1 as the load conditions for adaptation are fulfilled.

If LV_LAM_AD_ENA is set by the priority manager and stoichiometry request is allowed for offset adaptation by means of the respective calibration flag LV_LAM_AD_AFS_REQ is set to 1.

T_PRI_LAM_AD[i] is set depending on the last learned value in this adaptation field.

The other general flags are set to 0 and all other variables remain unchanged.

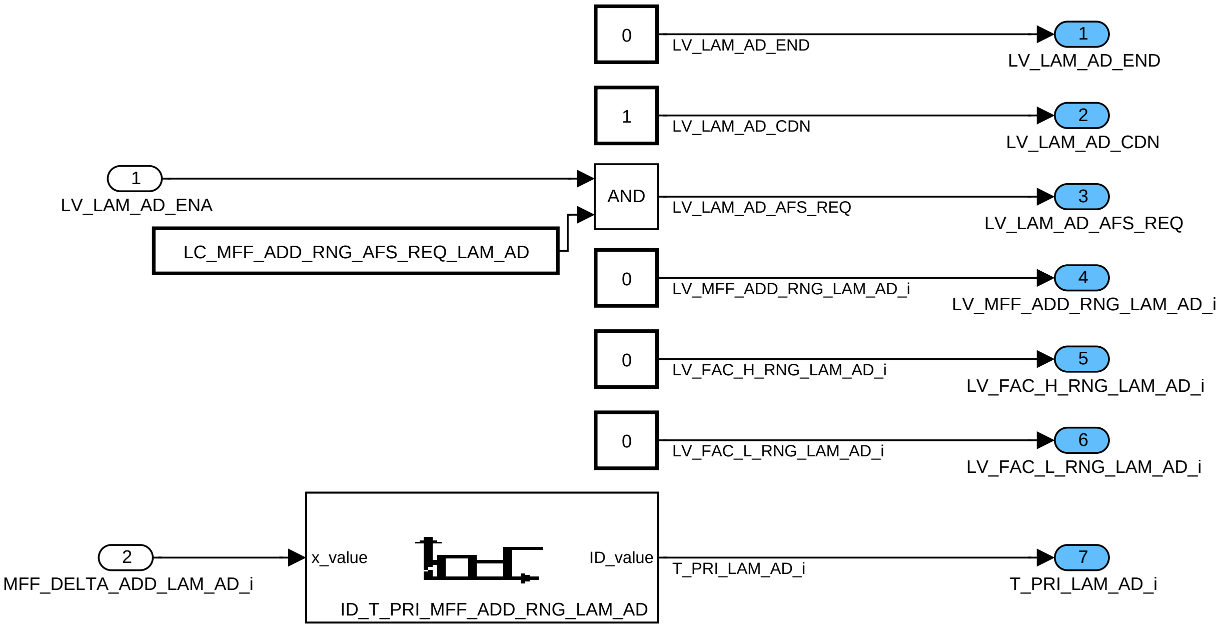


Figure 95 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ MFF_ADD_CDN/ SET_VALUE

State “ADAPT_ADD”

Set Variables

LV_MFF_ADD_RNG_LAM_AD[i] is set to 1 as adaptation is active in the respective adaptation field.

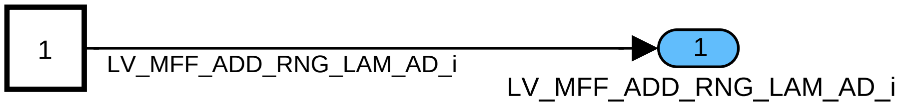


Figure 96 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ MFF_ADD_AD/ SET_VALUE

Check Stopped Lambda Adaptation Cycle

Lambda adaptation cycle shall not be stopped (LV_LAM_AD_STOP[i] must be 0).

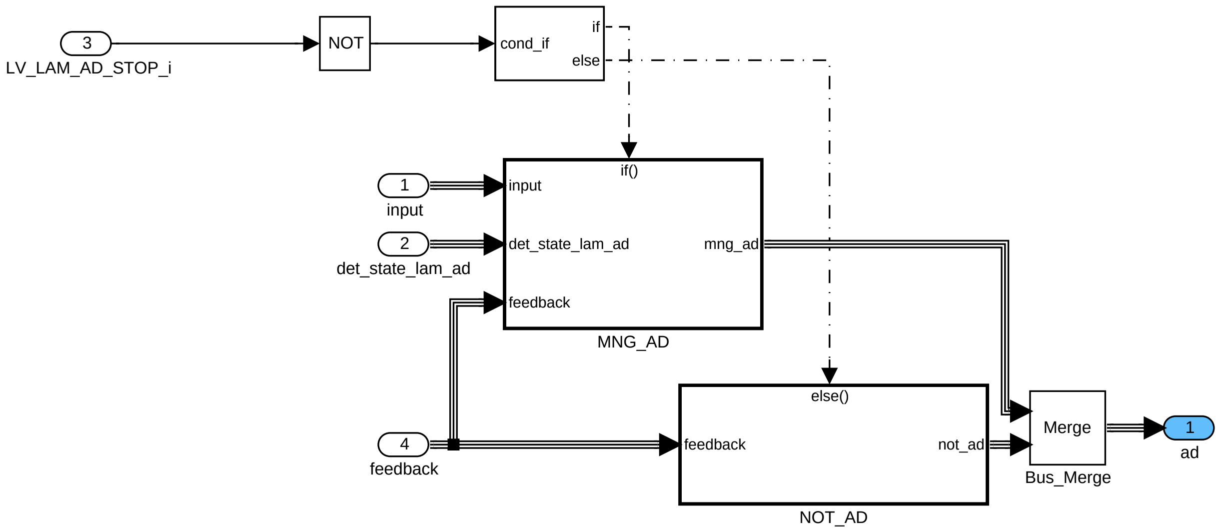


Figure 97 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ MFF_ADD_AD/ AD

Wait for Timer Threshold Reached

Increment Counter

Wait timer is incremented by T_SAMPLE.

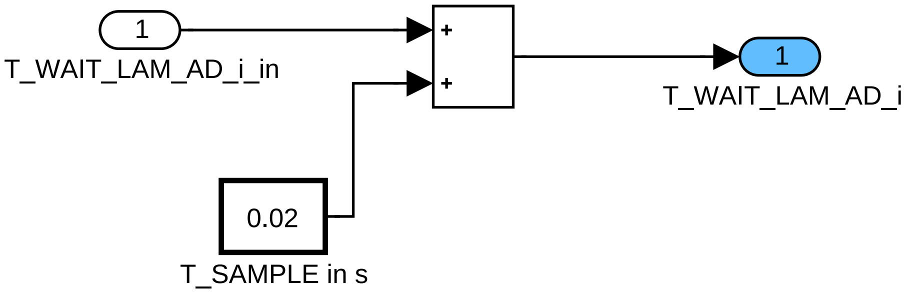


Figure 98 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ MFF_ADD_AD/ AD/ MNG_AD/ INC_CTR

Check Wait Timer Threshold

New adaptation value is calculated if wait timer reaches threshold.

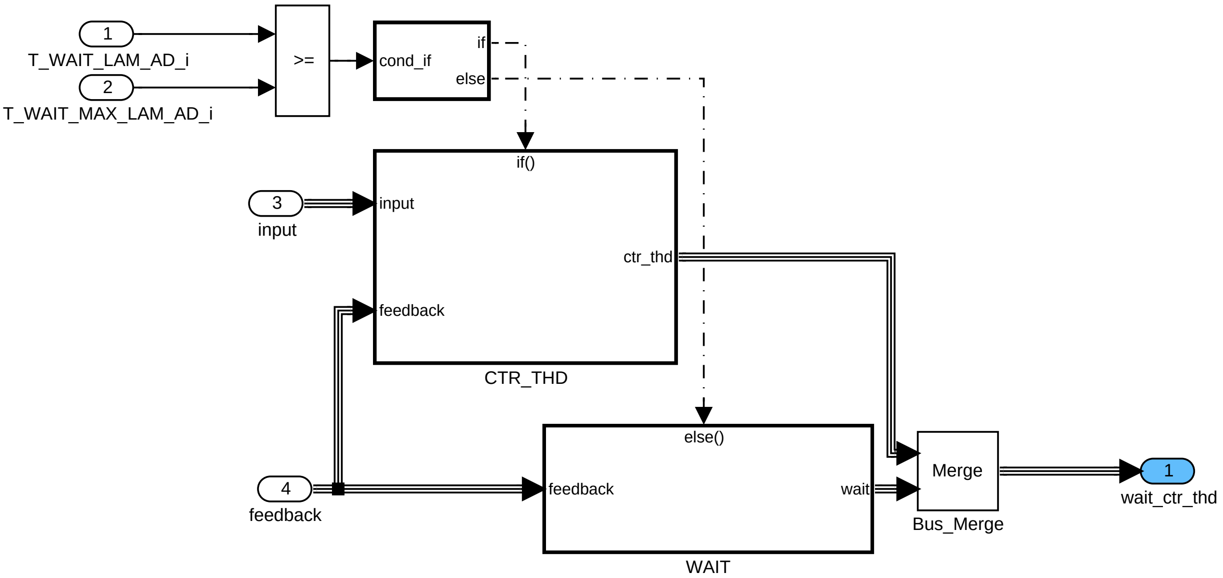


Figure 99 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ MFF_ADD_AD/ AD/ MNG_AD/ WAIT_CTR_THD

Wait Timer threshold reached

Calculation of lambda controller shift

In order to apply the lambda controller shift with the same amount as the lambda adaptation shift FAC_LAM_ADJ_LAM_AD[i] is calculated based on the lambda controller output and a correlation constant (defined in application incidences) The respective lambda adaptation shift is calculated (with high resolution by the internal variable MFF_DELTA_ADD_LAM_AD_H_RES[i]) by multiplying FAC_LAM_ADJ_LAM_AD[i] with the fuel mass flow set point. The result is assigned to MFF_DELTA_ADD_LAM_AD[i] which is the lambda adaptation shift that is finally applied.

The variable MFF_DELTA_ADD_LAM_AD[i] will be 0 as long as the absolute value of MFF_DELTA_ADD_LAM_AD_H_RES[i] is below the resolution of MFF_DELTA_ADD_LAM_AD[i]

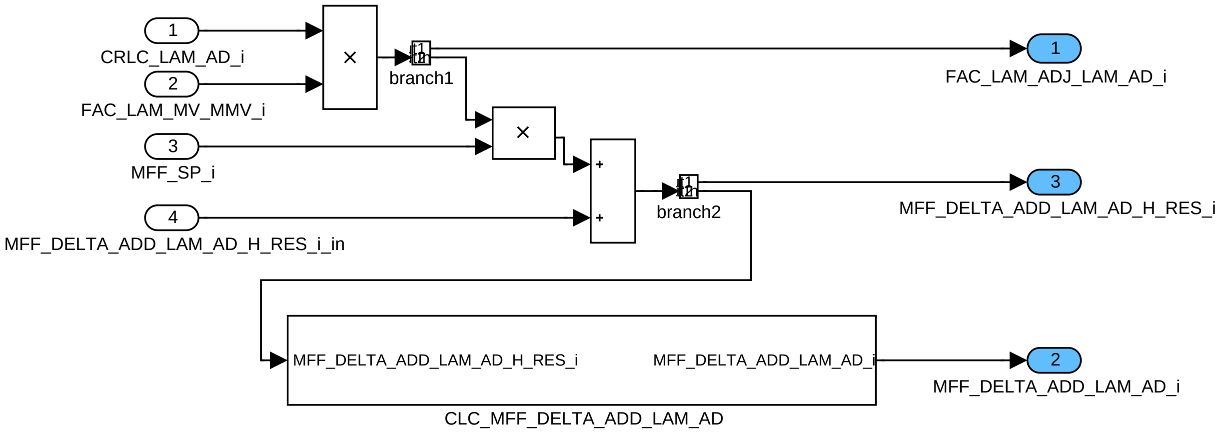


Figure 100 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ MFF_ADD_AD/ AD/ MNG_AD/ WAIT_CTR_THD/ CTR_THD/ CLC_LAM_SHIFT

Wait for determination of new adaptation value

As long as MFF_DELTA_ADD_LAM_AD[i] is 0 no new adaptation value is taken into account.

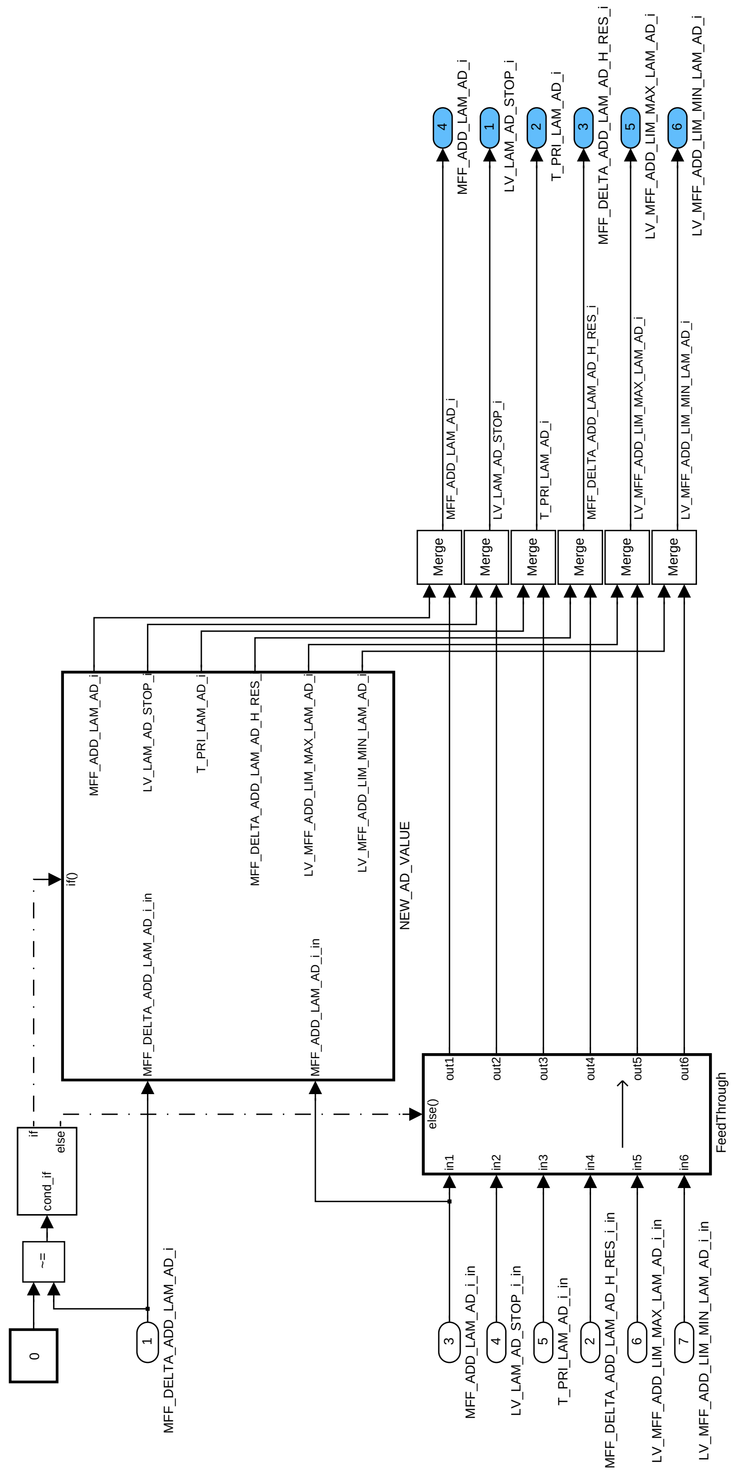


Figure 101 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ MFF_ADD_AD/ AD/ MNG_AD/ WAIT_CTR_THD/ CTR_THD/ WAIT_NEW_AD_VALUE

Determine new adaptation value

The new delta is added to the adaptation value and limited on upper and lower thresholds. If these thresholds are exceeded the respective flags are set to 1.

A new priority time is evaluated based on the delta.

LV_LAM_AD_STOP[i] is set to 1 to indicate that a lambda controller shift can take place.

MFF_DELTA_ADD_LAM_AD_H_RES[i] is set to 0 for next learning sequence.

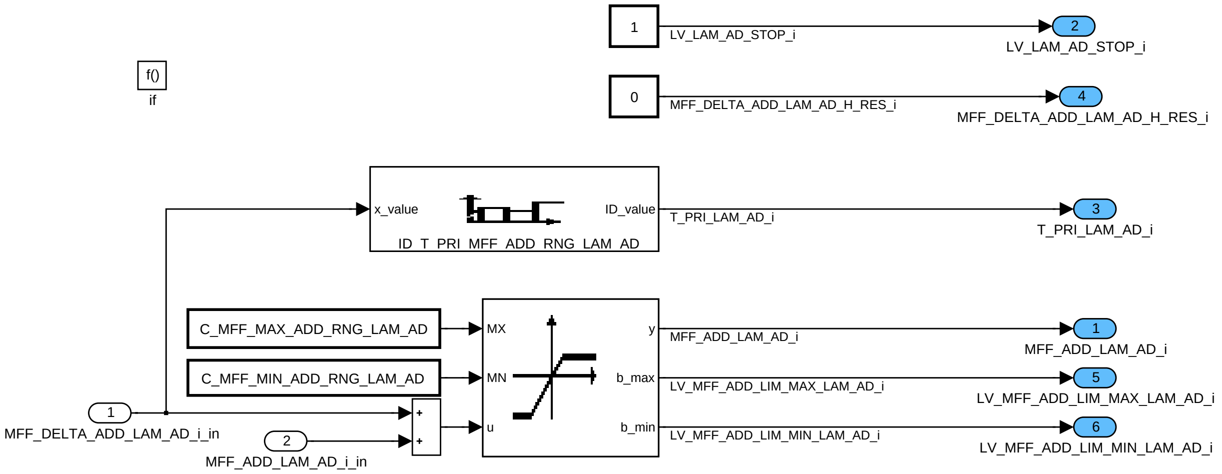


Figure 102 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ MFF_ADD_AD/ AD/ MNG_AD/ WAIT_CTR_THD/ CTR_THD/ WAIT_NEW_AD_VALUE/ NEW_AD_VALUE

State “CDN_FAC_L”

Set Variables

LV_LAM_AD_CDN is set to 1 as the load conditions for adaptation are fulfilled.

If LV_LAM_AD_ENA is set by the priority manager and stoichiometry request is allowed for lower factor field adaptation by means of the respective calibration flag LV_LAM_AD_AFS_REQ is set to 1.

T_PRI_LAM_AD[i] is set depending on the last learned value in this adaptation field.

The other general flags are set to 0 and all other variables remain unchanged.

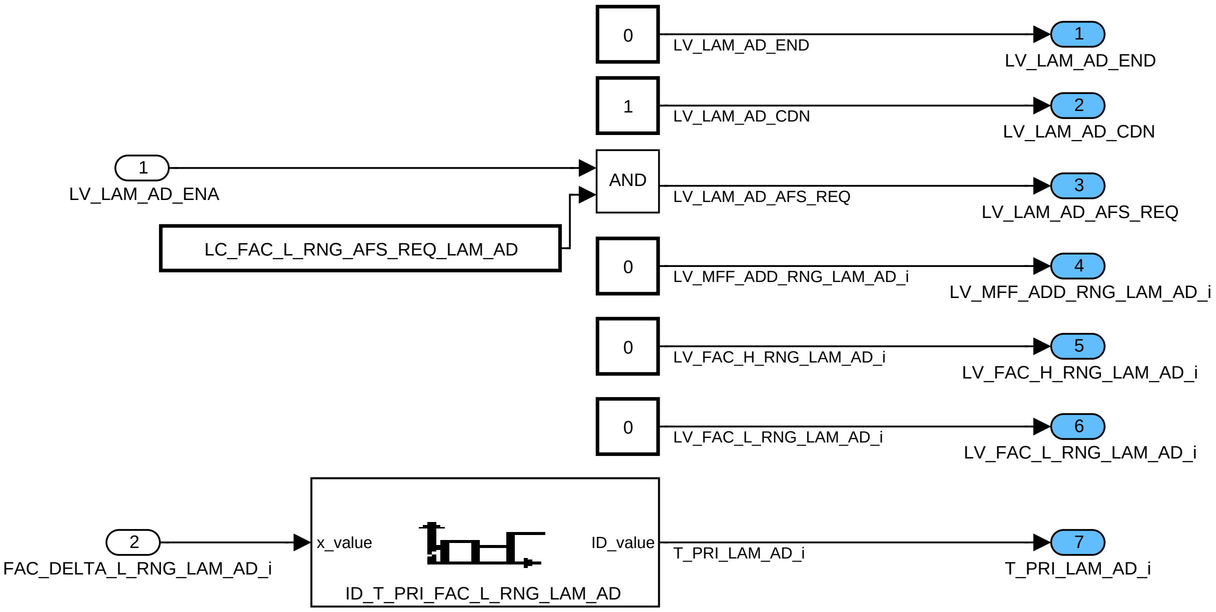


Figure 103 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_L_CDN/ SET_VALUE

State “ADAPT_FAC_L”

Set Variables

LV_FAC_L_RNG_LAM_AD[i] is set to 1 as adaptation is active in the respective adaptation field.

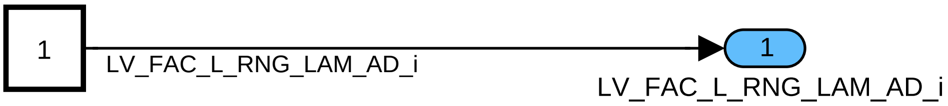


Figure 104 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_L_AD/ SET_VALUE

Check Stopped Lambda Adaptation Cycle

Lambda adaptation cycle shall not be stopped (LV_LAM_AD_STOP[i] must be 0).

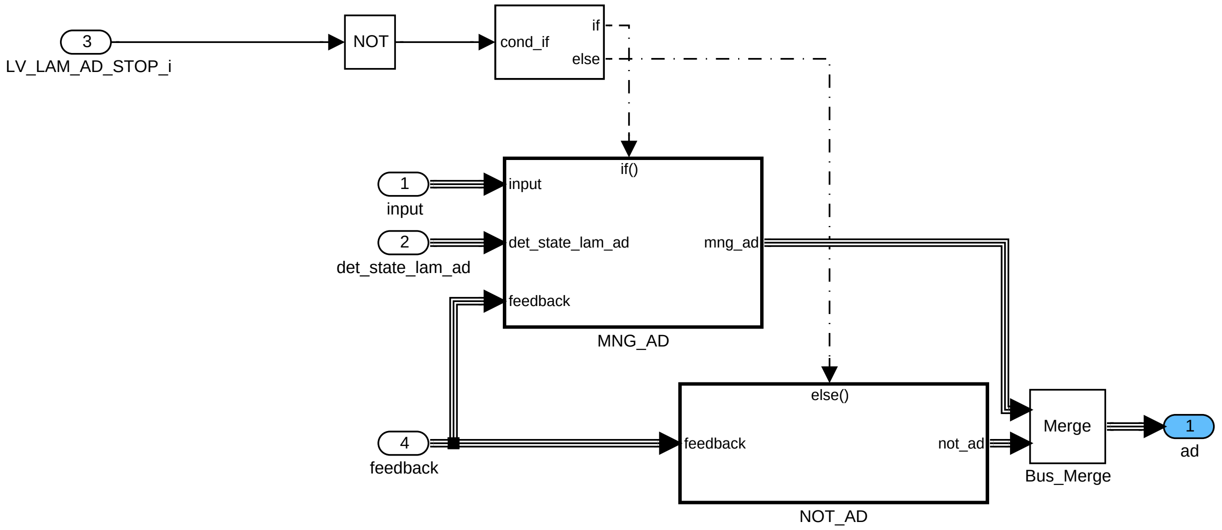


Figure 105 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_L_AD/ AD

Wait for Timer Threshold Reached

Increment Counter

Wait timer is incremented by T_SAMPLE.

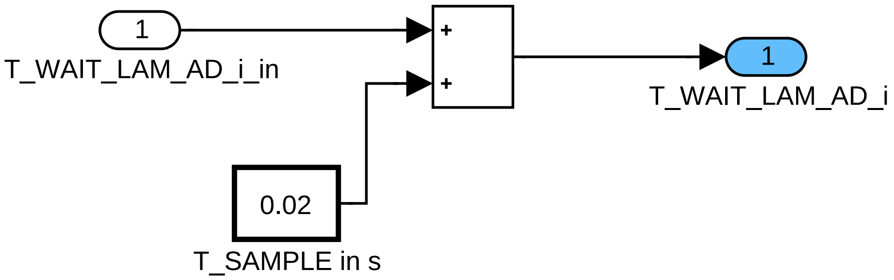


Figure 106 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_L_AD/ AD/ MNG_AD/ INC_CTR

Check Wait Timer Threshold

New adaptation value is calculated if wait timer reaches threshold.

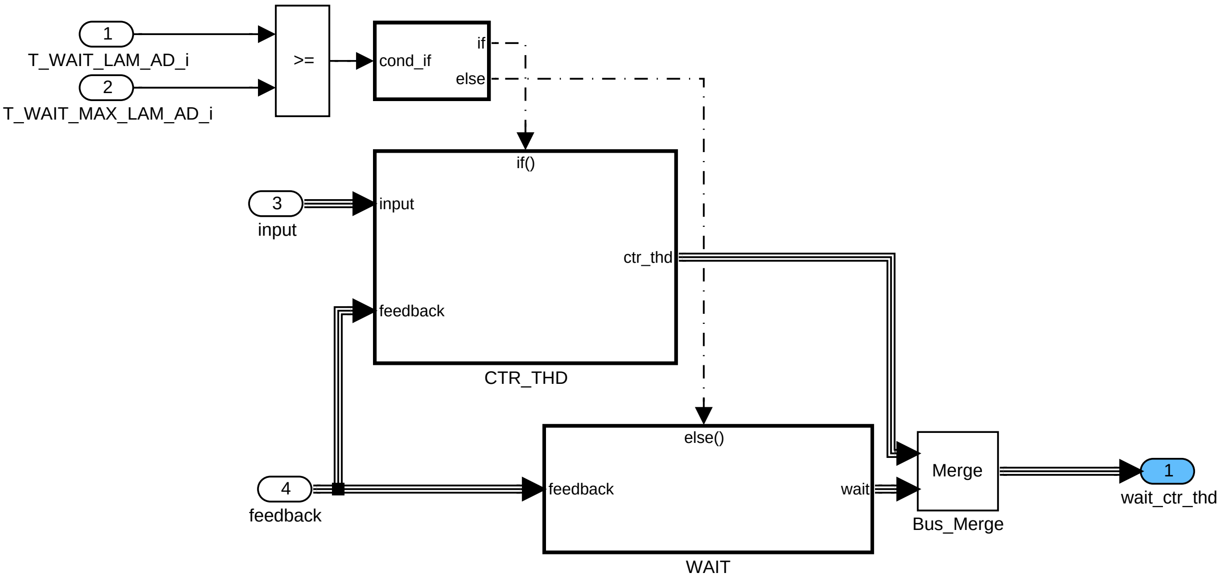


Figure 107 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_L_AD/ AD/ MNG_AD/ WAIT_CTR_THD

Wait Timer Threshold Reached

Calculation of Lambda Controller Shift

The new delta for the adaptation value is calculated based on the lambda controller output and a correlation constant (defined in application incidences). This value is the same as for the shift applied to the lambda controller.

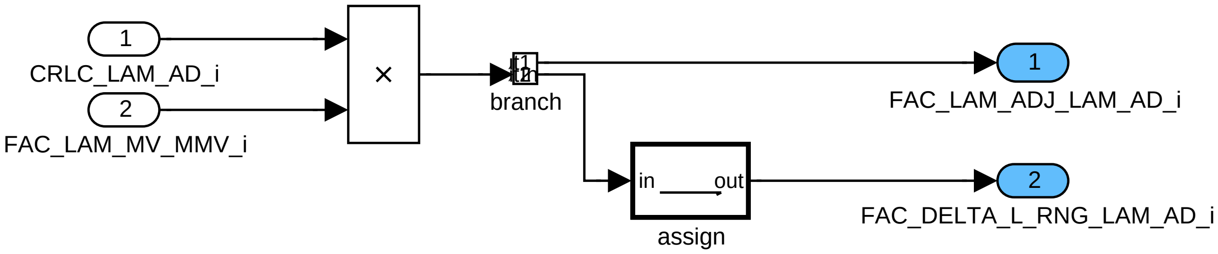


Figure 108 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_L_AD/ AD/ MNG_AD/ WAIT_CTR_THD/ CTR_THD/ CLC_LAM_SHIFT

Determine New Adaptation Value

The new delta is added to the adaptation value and limited on upper and lower thresholds. If these thresholds are exceeded the respective flags are set to 1.

In case the high range adaptation value was not yet learned since the last reset of the NVMY data (indicated by LV_FAC_H_RNG_LAM_AD_INI[i] = 1), the low range value is written to the high range value.

A new priority time is evaluated based on the delta.

LV_LAM_AD_STOP[i] is set to 1 to indicate that a lambda controller shift can take place.

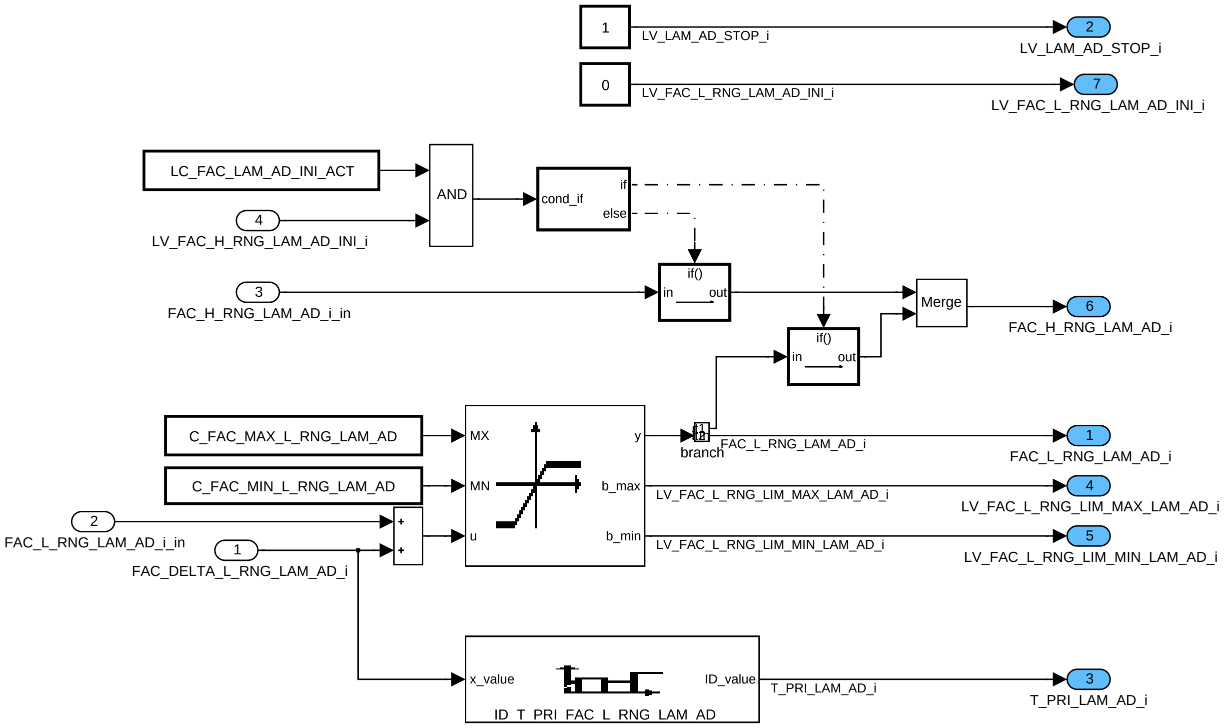


Figure 109 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_L_AD/ AD/ MNG_AD/ WAIT_CTR_THD/ CTR_THD/ NEW_AD_VALUE

State “CDN_FAC_H”

Set Variables

LV_LAM_AD_CDN is set to 1 as the load conditions for adaptation are fulfilled.

If LV_LAM_AD_ENA is set by the priority manager and stoichiometry request is allowed for higher factor field adaptation by means of the respective calibration flag LV_LAM_AD_AFS_REQ is set to 1.

T_PRI_LAM_AD[i] is set depending on the last learned value in this adaptation field.

The other general flags are set to 0 and all other variables remain unchanged.

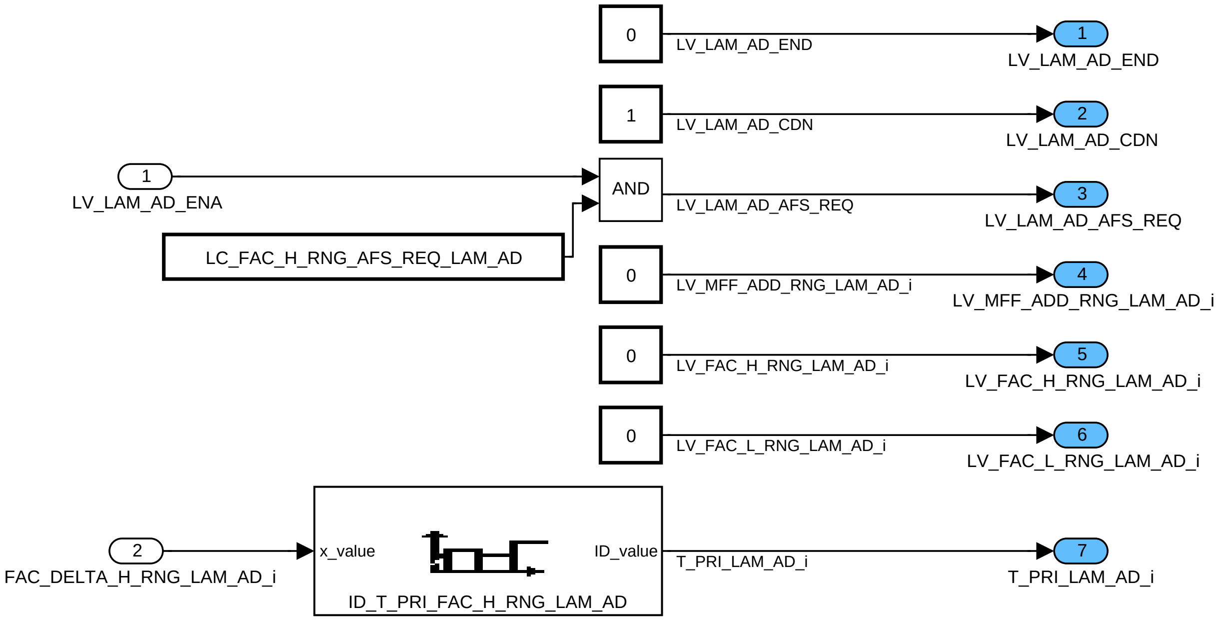


Figure 110 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_H_CDN/ SET_VALUE

State “ADAPT_FAC_H”

Set Variables

LV_FAC_H_RNG_LAM_AD[i] is set to 1 as adaptation is active in the respective adaptation field.

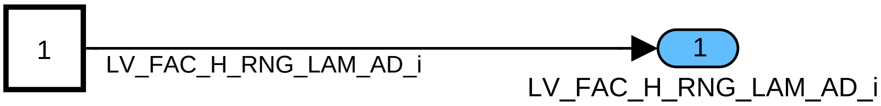


Figure 111 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_H_AD/ SET_VALUE

Check Stopped Lambda Adaptation Cycle

Lambda adaptation cycle shall not be stopped (LV_LAM_AD_STOP[i] must be 0).

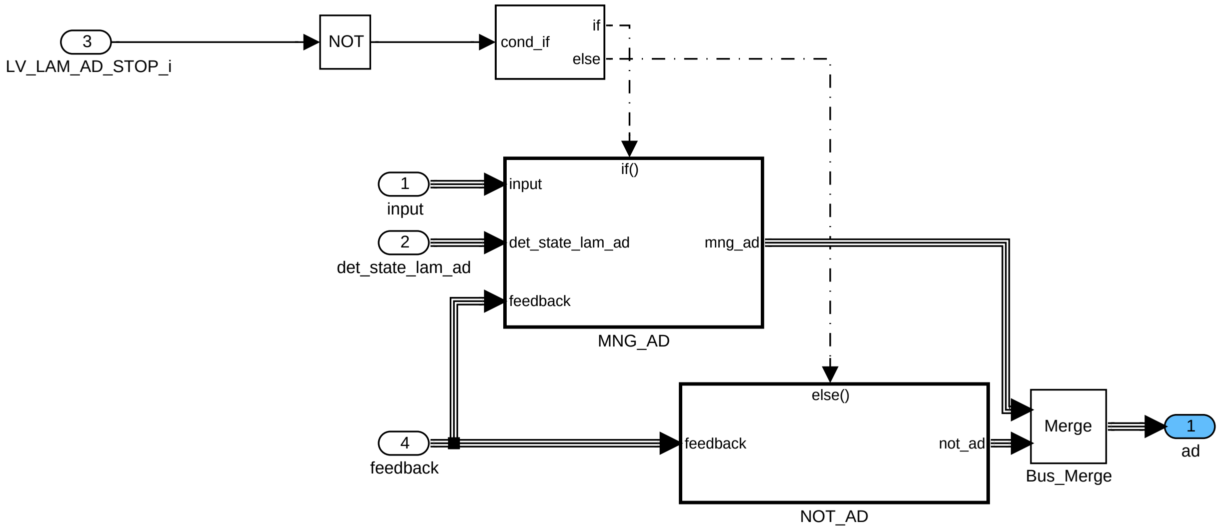


Figure 112 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_H_AD/ AD

Wait for Timer Threshold Reached

Increment Counter

Wait timer is incremented by T_SAMPLE.

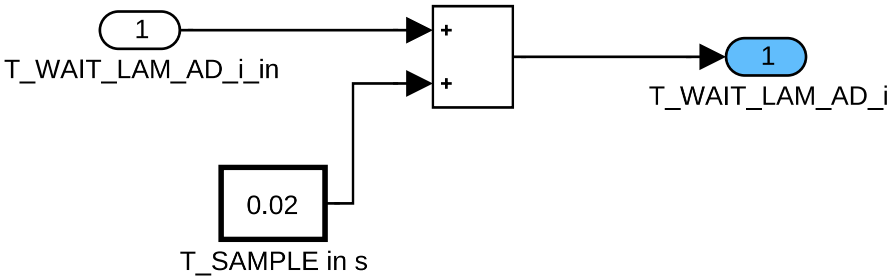


Figure 113 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_H_AD/ AD/ MNG_AD/ INC_CTR

Check Wait Timer Threshold

New adaptation value is calculated if wait timer reaches threshold.

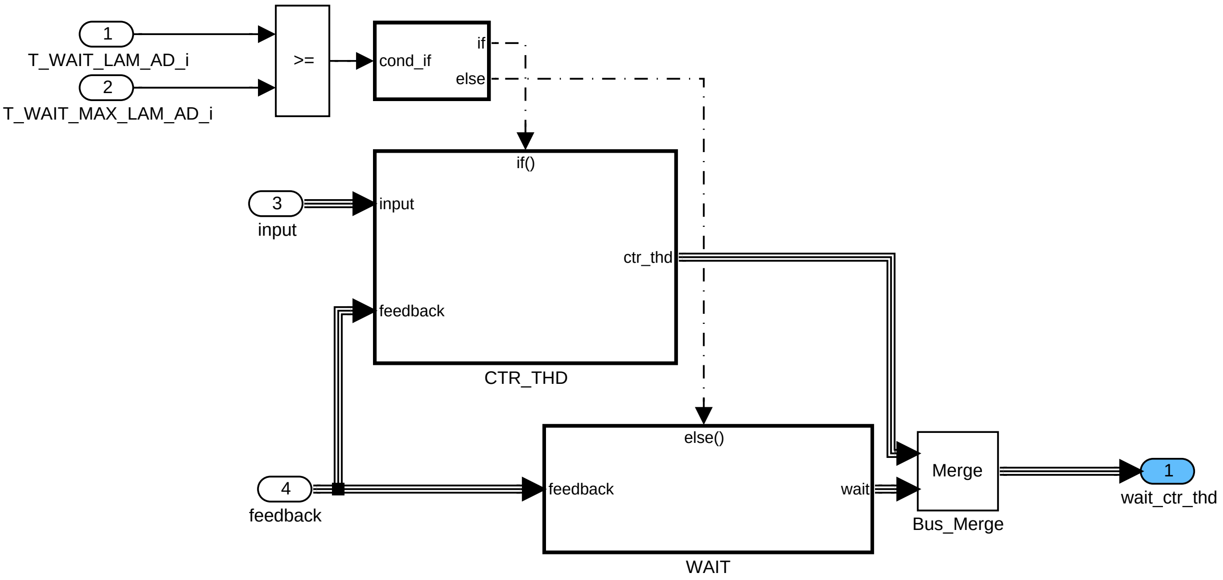


Figure 114 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_H_AD/ AD/ MNG_AD/ WAIT_CTR_THD

Wait Timer Threshold Reached

Calculation of Lambda Controller Shift

The new delta for the adaptation value is calculated based on the lambda controller output and a correlation constant (defined in application incidences). This value is the same as for the shift applied to the lambda controller.

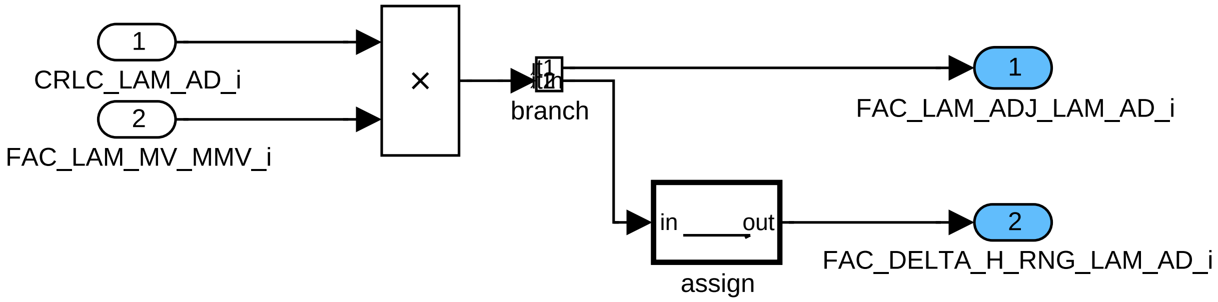


Figure 115 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_H_AD/ AD/ MNG_AD/ WAIT_CTR_THD/ CTR_THD/ CLC_LAM_SHIFT

Determine New Adaptation Value

The new delta is added to the adaptation value and limited on upper and lower thresholds. If these thresholds are exceeded the respective flags are set to 1.

In case the low range adaptation value was not yet learned since the last reset of the NVMY data (indicated by LV_FAC_L_RNG_LAM_AD_INI[i] = 1), the high range value is written to the low range value.

A new priority time is evaluated based on the delta.

LV_LAM_AD_STOP[i] is set to 1 to indicate that a lambda controller shift can take place.

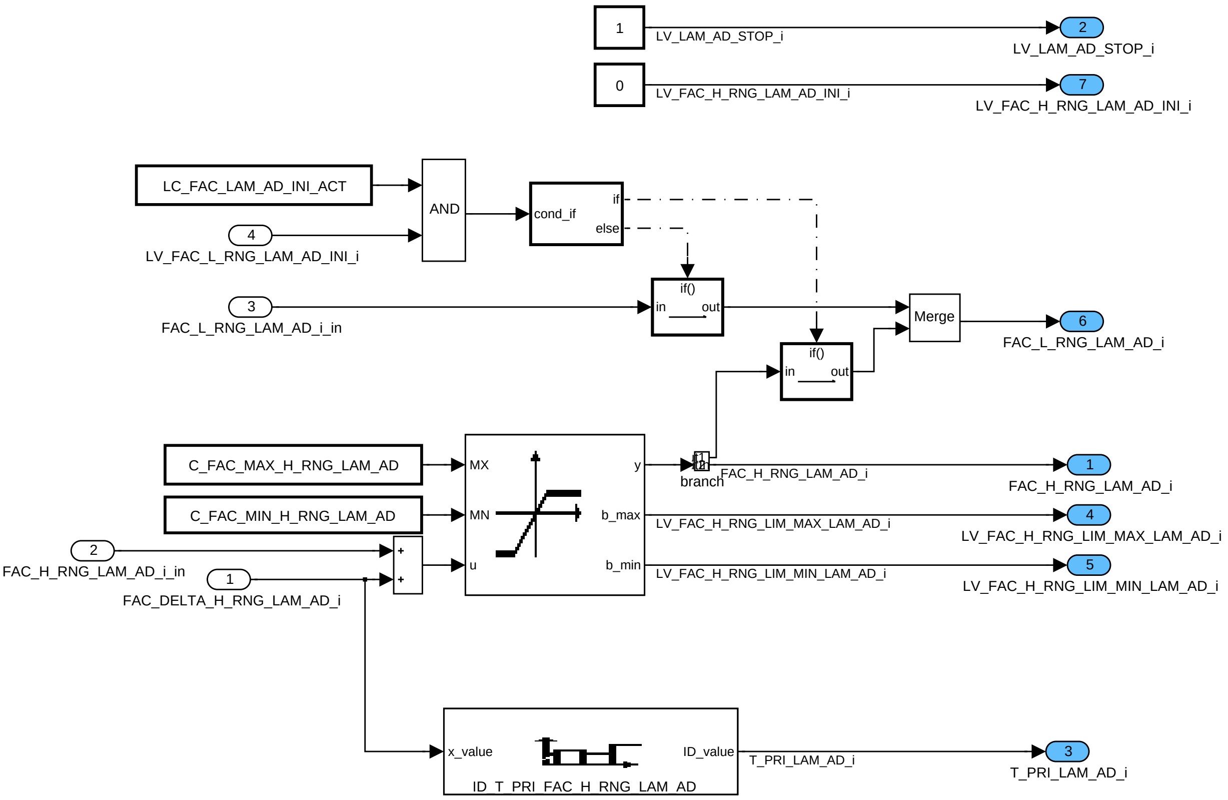


Figure 116 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ STATE_ACTION/ FAC_H_AD/ AD/ MNG_AD/ WAIT_CTR_THD/ CTR_THD/ NEW_AD_VALUE

State Machine Independent Calculations

Re-Writing of Adaptation Correction at transition of ignition key

Re-writing of the adaptation values if required if not both adaptation factors were learned in the past.

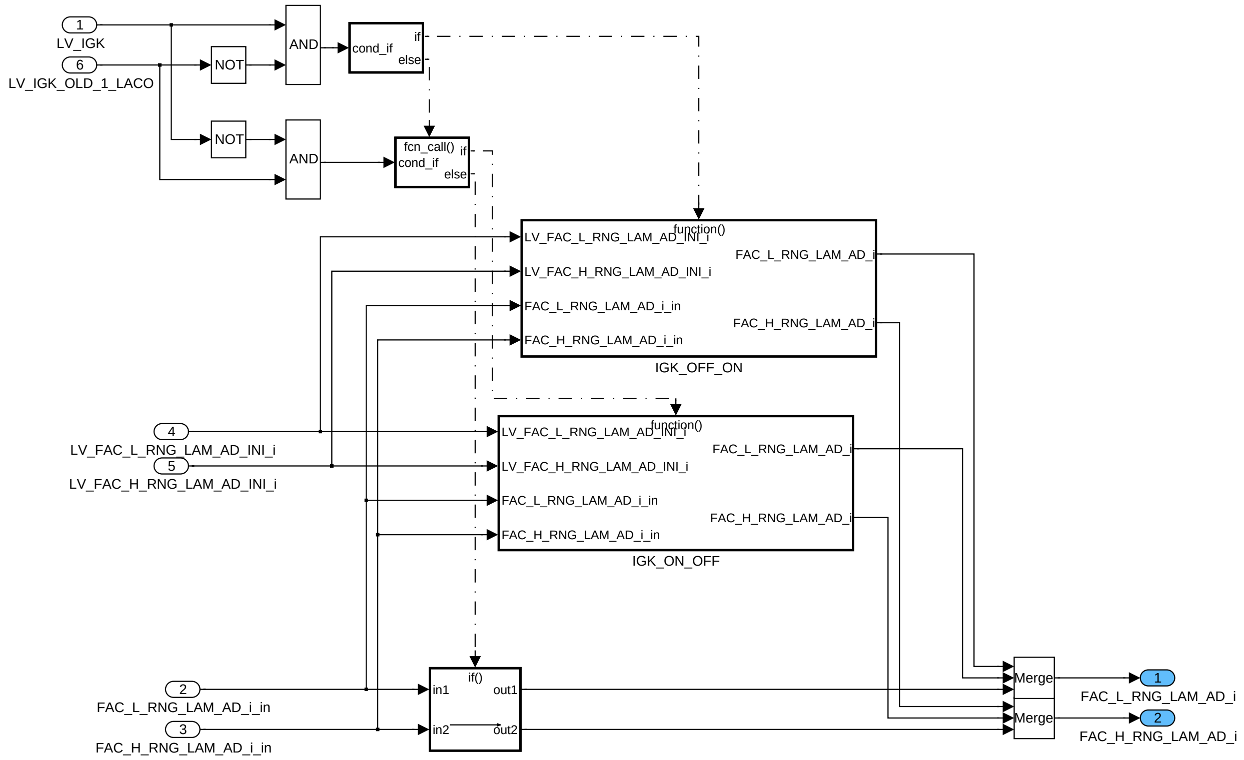


Figure 117 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ IGK_LAM_AD

Ignition key transition off to on

If only one of the initialization flags is set, then the respective adaptation value is reinitialized with the adaptation value of the other adaptation range.

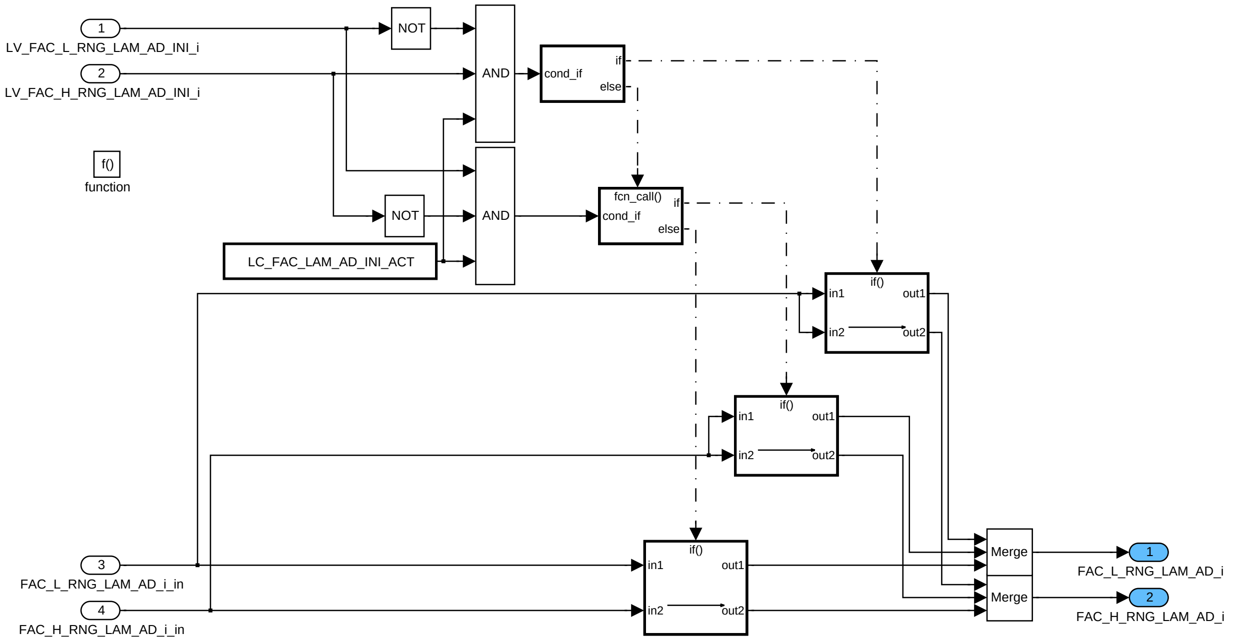


Figure 118 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ IGK_LAM_AD/ IGK_OFF_ON

Ignition key transition on to off

If the flag indicating that no adaptation was done in the relevant adaptation range, the adaptation values shall be reset to zero. This is required to set LV_FAC_L_RNG_LAM_AD_INI[i] respectively LV_FAC_H_RNG_LAM_AD_INI[i] in the next driving cycle.

If by any chance a adaptation of excately zero was learned, a adaptation value of 1 hex shall be written to the NVMY in order to avoid any false setting of LV_FAC_L_RNG_LAM_AD_INI[i] respectively LV_FAC_H_RNG_LAM_AD_INI[i] in the next driving cycle.

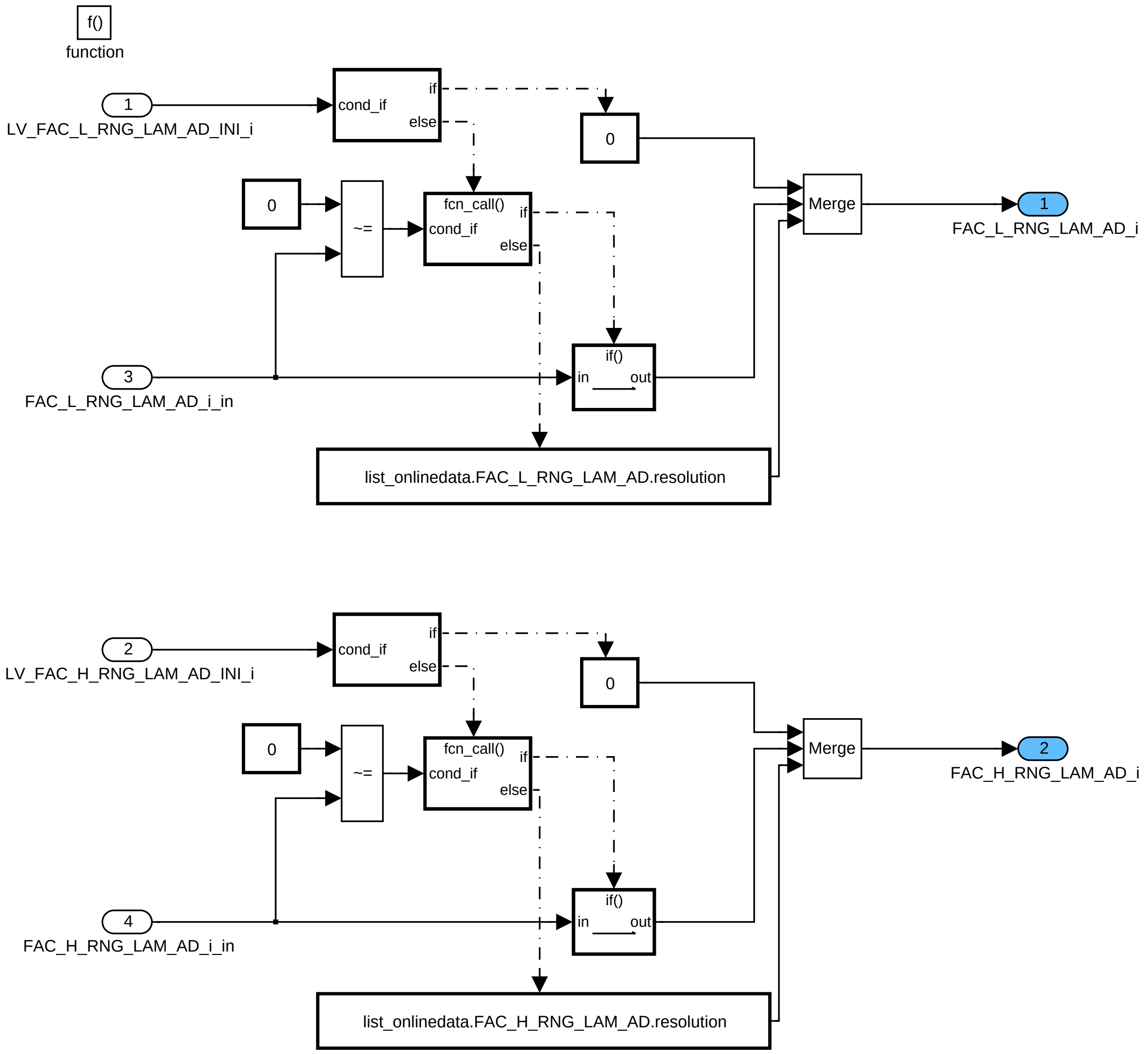


Figure 119 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ IGK_LAM_AD/ IGK_ON_OFF

Re-Writing of Adaptation Correction with Non Volatile Stored Values

In case of lambda adaptation relevant errors (see application incidences) the adaptation correction values MFF_ADD_LAM_AD[i], FAC_L_RNG_LAM_AD[i] and FAC_H_RNG_LAM_AD[i] must be rewritten with the values stored at the end of the last driving cycle. These values are available with the variables MFF_ADD_LAM_AD_TMP[i], FAC_L_RNG_LAM_AD_TMP[i] and FAC_H_RNG_LAM_AD_TMP[i] written to at the beginning of the driving cycle with the stored values of the last driving cycle.

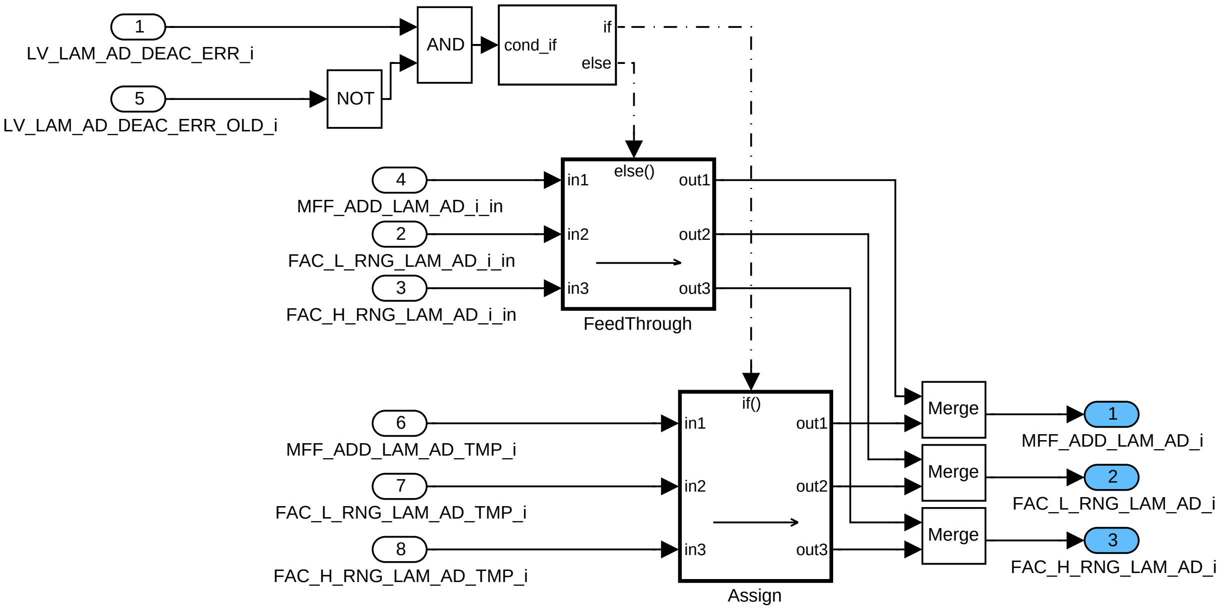


Figure 120 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ ERR_LAM_AD

Shift of adaptation values by external function

External functions might require an inverse step change in the Lambda adaptation value. The control of the value shift takes place in the external function and is indicated by the flags LV_FAC_LAM_AD_SHIFT[i].

The Lambda adaptation is shifted if the condition LV_FAC_LAM_AD_SHIFT[i] = 1 is valid. This variable is to be reset if the Lambda adaptation shift was carried out. That requires a confirmation of the executed controller shift. The confirmation is done by setting the variable LV_FAC_LAM_AD_SHIFT_END[i] to 1.

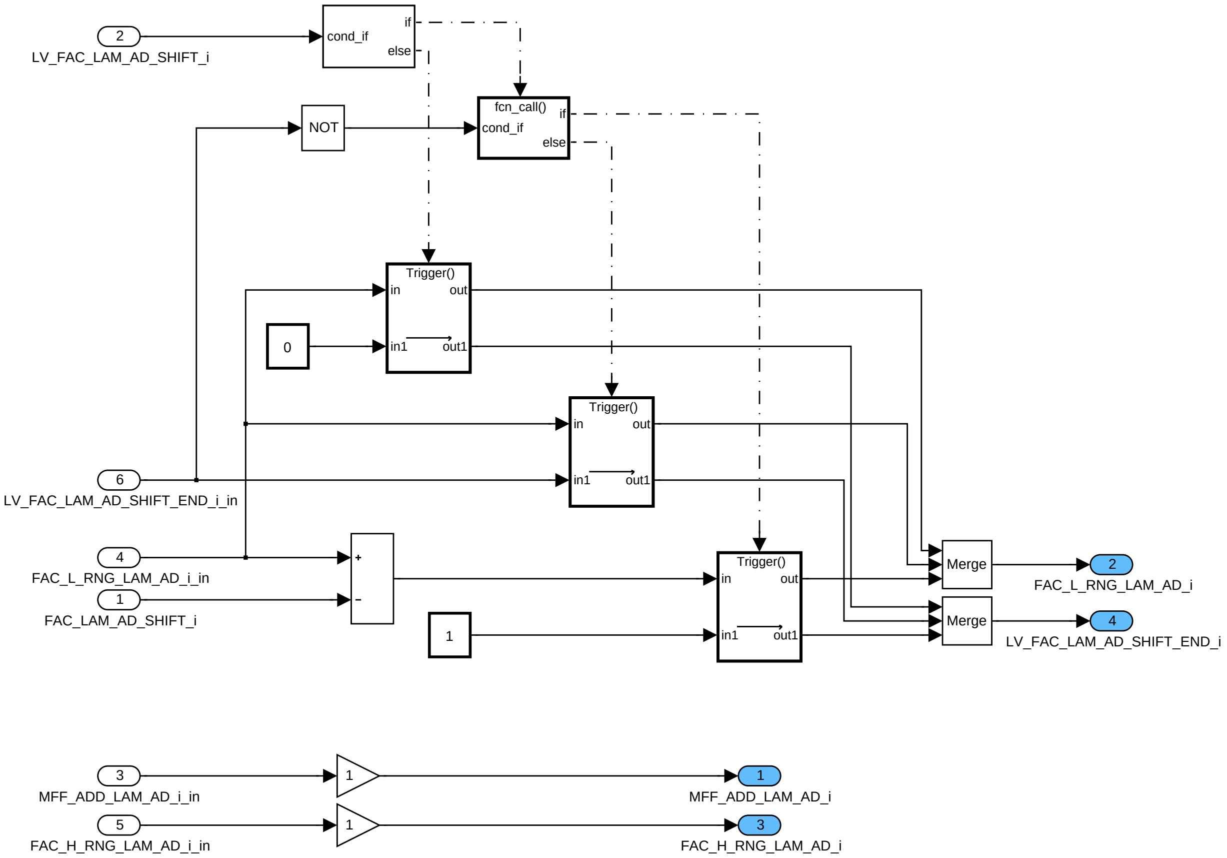


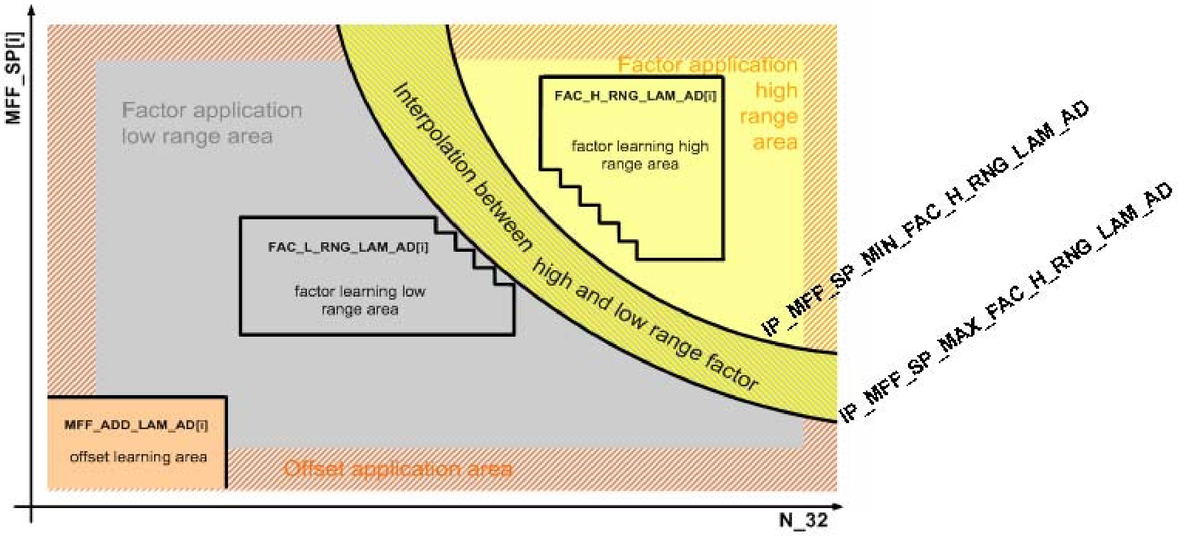
Figure 121 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ SHIFT_LAM_AD

Determination of Interpolated Lambda Adaptation Factor

In order to meet the physical behaviour of engine components (especially the air mass flow sensor), the actual multiplicative factor is calculated out of the two determined factors by interpolation at every state. Interpolation is applied between the borders

- IP_MFF_SP_MAX_FAC_H_RNG_LAM_AD (lower border) and
- IP_MFF_SP_MIN_FAC_H_RNG_LAM_AD (upper border)

as shown in the figure.



Interpolation of adaptation factors

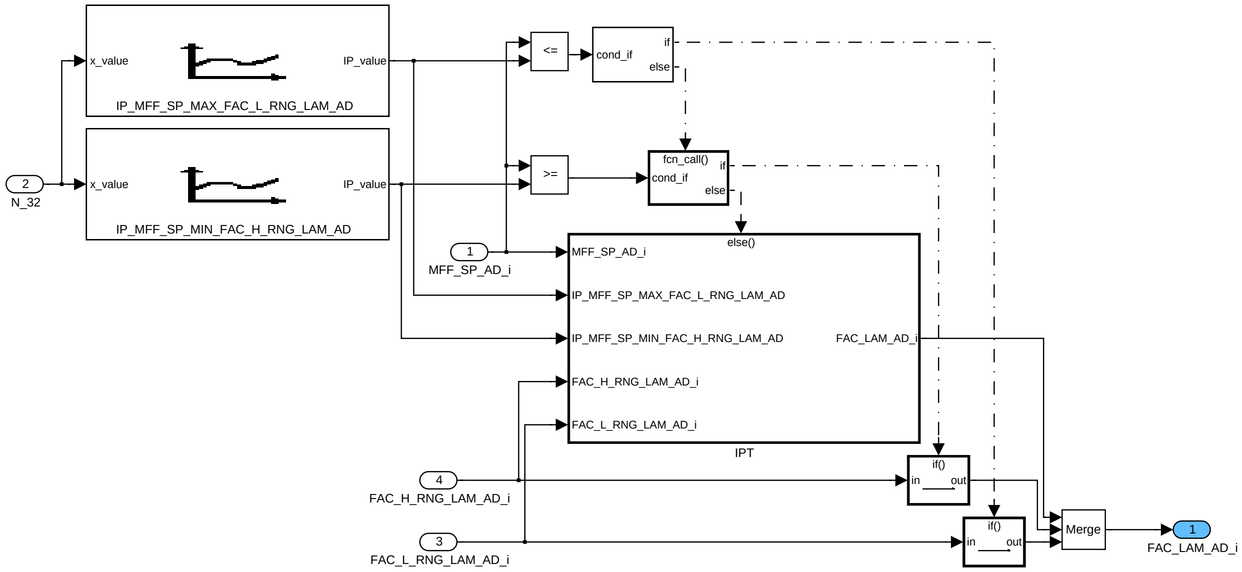


Figure 122 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ CLC_FAC_LAM_AD

Calculation of Output Signals for Fuel Mass Flow Set Point Correction

In order to avoid mixture problems when the engine has not reached its service temperature (warm up), lambda adaptation factors and their corresponding relative intermediate results can be decreased versus TCO and their own value (which is applied when the engine has its service temperature).

This is performed by weighting the adaptive factors as long as the lambda adaptation is disabled due to TCO ≤ C_TCO_MIN_LAM_AD (see: "application incidences").

The quotient of the additive relative factor is determined. The values MFF_LAM_ADD_LAM_AD_OUT[i] and MFF_SP[i] should be conditioned so, that FAC_MFF_ADD_LAM_AD_OUT[i] does not exceed the physical range of +/-100 %!

The relative lambda adaptation FAC_MFF_ADD_FAC_LAM_AD[i] as long term adaptation is calculated out of both adaptation factors.

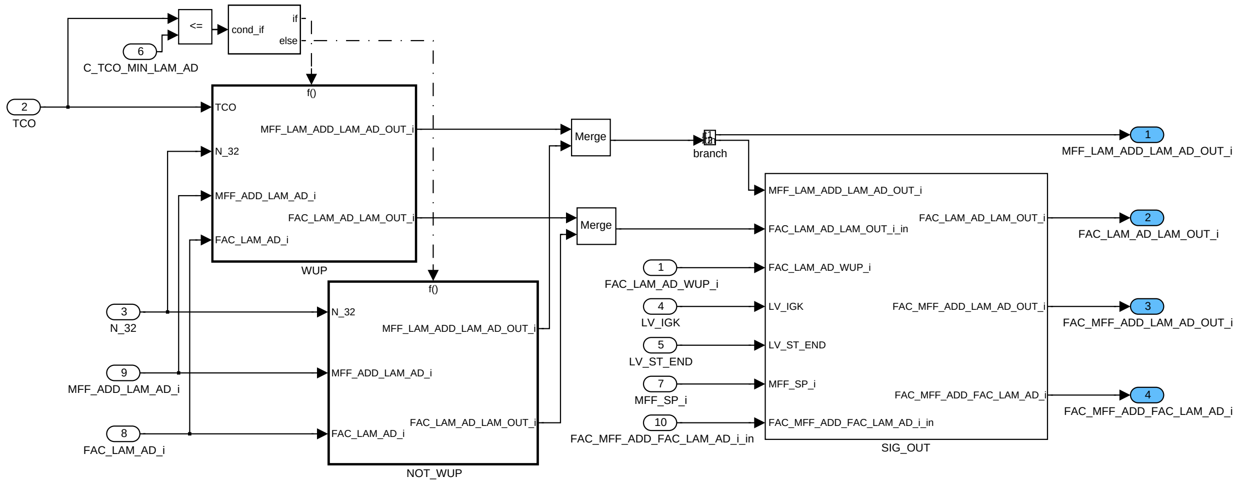


Figure 123 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ CLC_SIG_OUT

Apply Weighting Factors

A factor depending on engine speed can be applied if the lambda adaptation correction must not be applied by 100% in the whole engine speed range.

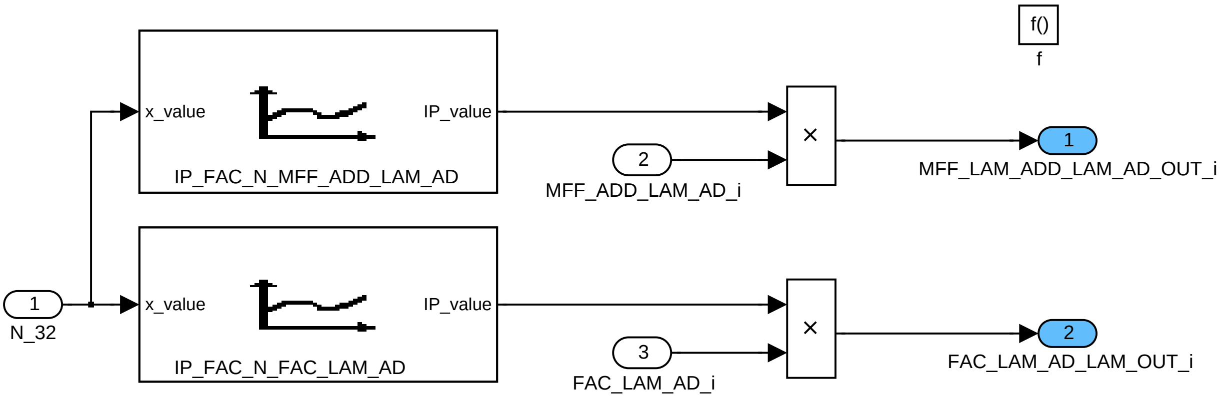


Figure 124 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ CLC_SIG_OUT/ NOT_WUP

Apply Weighting Factors during Warm Up Phase

During warm-up an additional factor depending on TCO and the lambda adaptation correction itself is applied.

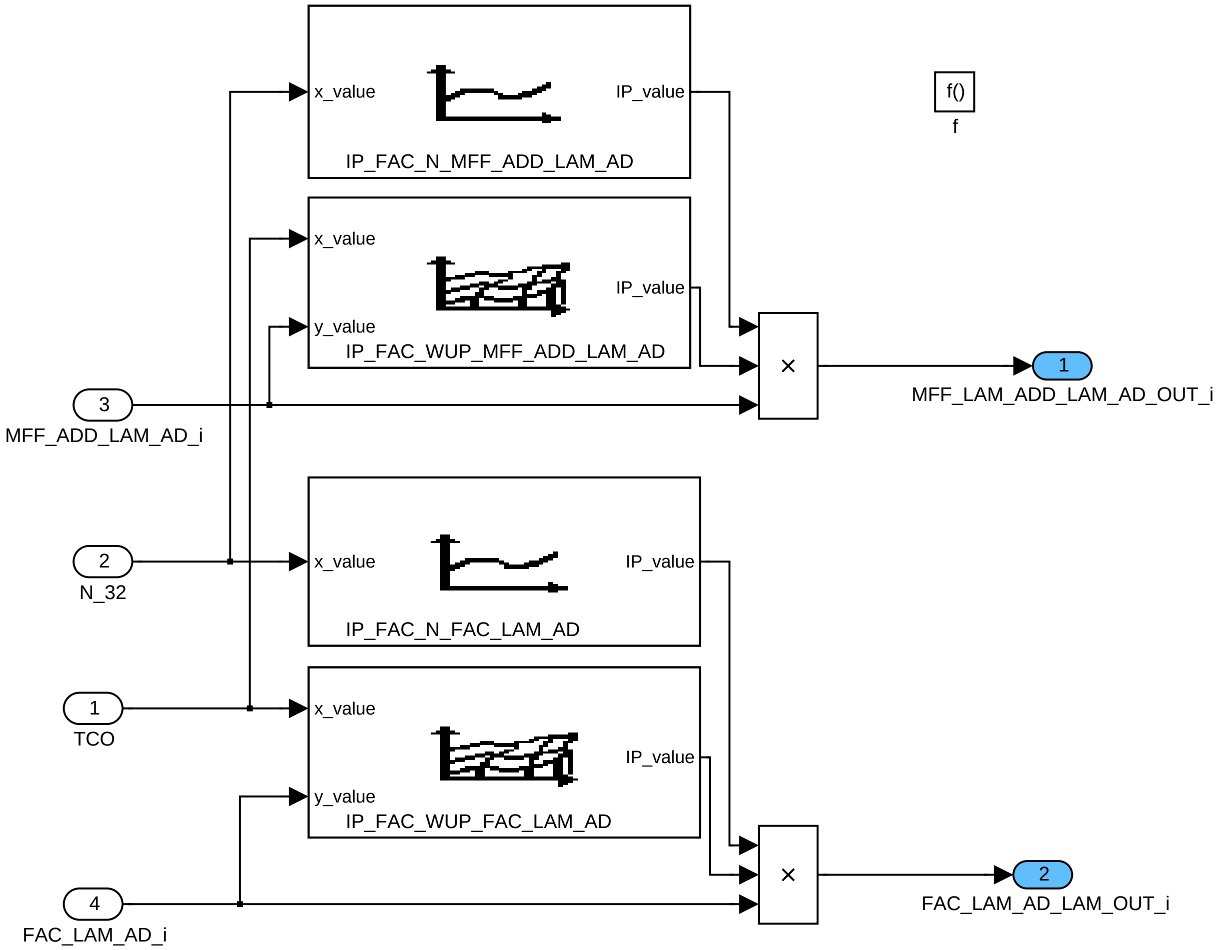


Figure 125 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ CLC_SIG_OUT/ WUP

Final Output

FAC_MFF_ADD_LAM_AD_OUT[i] is the additive correction related to the fuel mass flow set point in order to have the same definition as for the multiplicative correction. When the engine is running this value is added to the multiplicative correction; so with the variable FAC_MFF_ADD_FAC_LAM_AD[i] the whole lambda adaptation correction is expressed.

If the lambda adaptation in warm-up phase (external module) is active the respective output of this function FAC_LAM_AD_WUP[i] is not zero and added here to the over all lambda adaptation multiplicative correction.

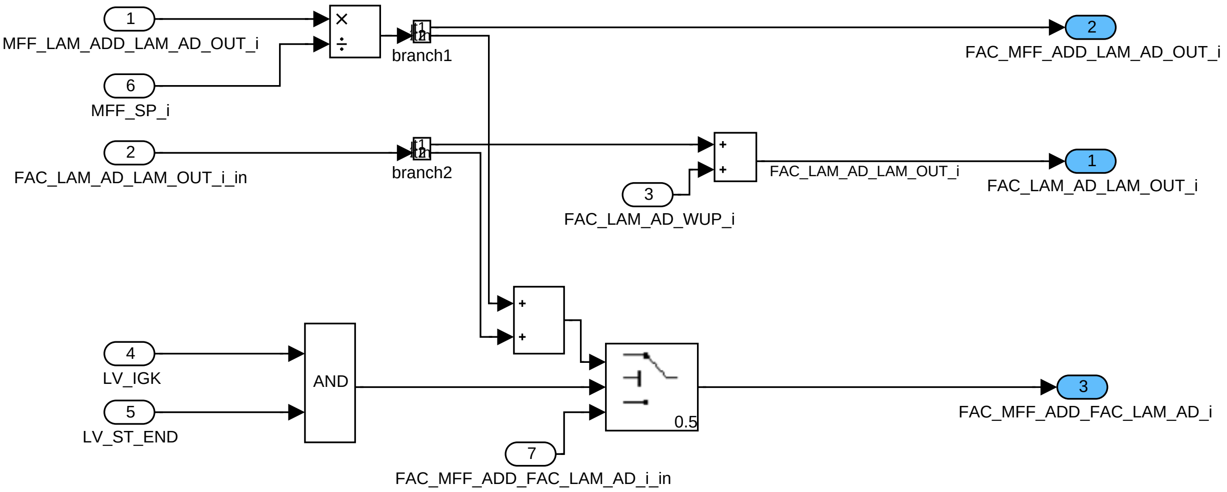


Figure 126 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ CLC_SIG_OUT/ SIG_OUT

Interface to Lambda Controller

Interfacing Lambda Adaptation (this function)

A shift of the lambda controller output after the calculation of a valid adaptation value by FAC_LAM_ADJ_LAM_AD[i] is required in order to avoid rapid changes in the air fuel ratio caused by the lambda adaptation. FAC_LAM_ADJ_LAM_AD[i] is calculated in the states "ADAPT_ADD", "ADAPT_FAC_L" and "ADAPT_FAC_H".

The shift of the lambda controller output is not carried out, if the lambda controller is stopped. To avoid a loss of adaptation values in that case, the lambda controller shall confirm the successful lambda controller shift.

If the lambda controller does not confirm the controller shift by FAC_LAM_ADJ_LAM_AD[i] by setting LV_FAC_LAM_ADJ_LAM_AD_END[i] to 1, the lambda adaptation stops further calculations (LV_LAM_AD_STOP[i] is set to 1) until the controller shift is confirmed. FAC_LAM_ADJ_LAM_AD[i] shall be set to 0 and LV_FAC_LAM_ADJ_LAM_AD[i] and LV_LAM_AD_STOP[i] are reset, if at a stopped lambda adaptation the deactivating conditions are fulfilled.

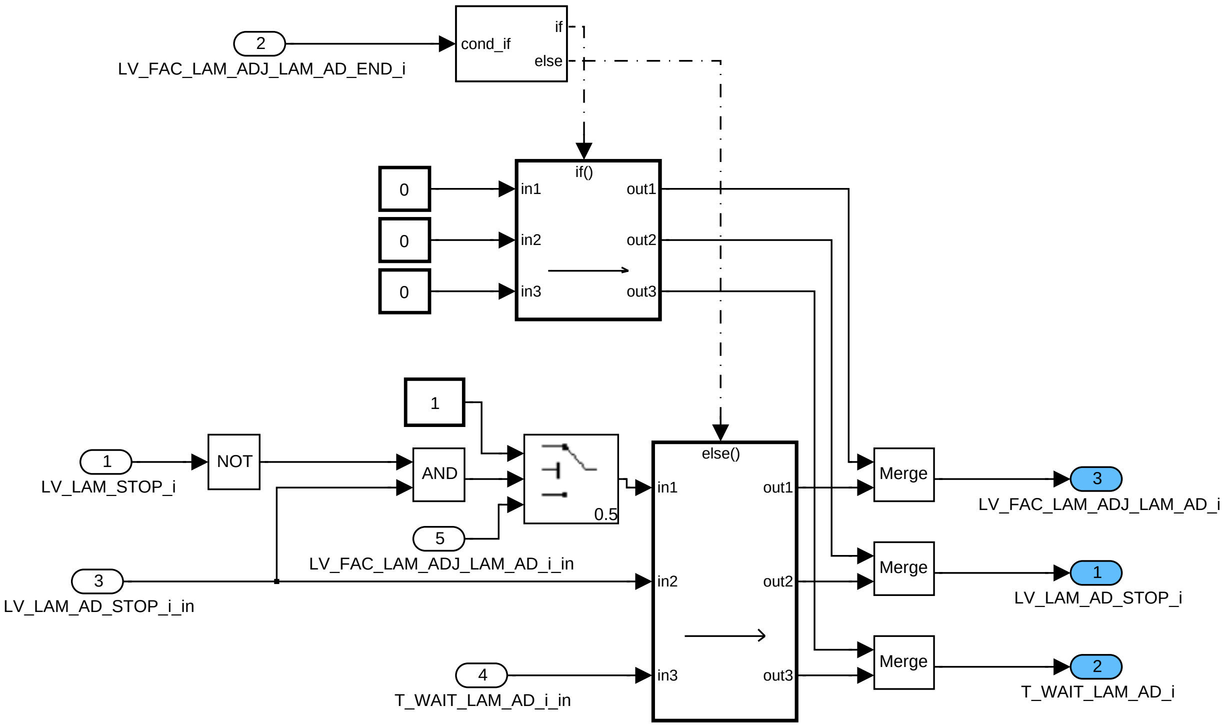


Figure 127 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ LAM_IF/ LAM_AD

Interfacing Lambda Adaptation in Warm-up phase (external module)

If TCO is below C_TCO_MIN_LAM_AD, any required shift by the warm-up lambda adaptation is taken into account. Above this threshold lambda adaptation is active and any shift of this function is taken into account. As both adaptation function cannot be active at the same time the request flag LV_FAC_LAM_ADJ_LAM_AD[i] is used for both.

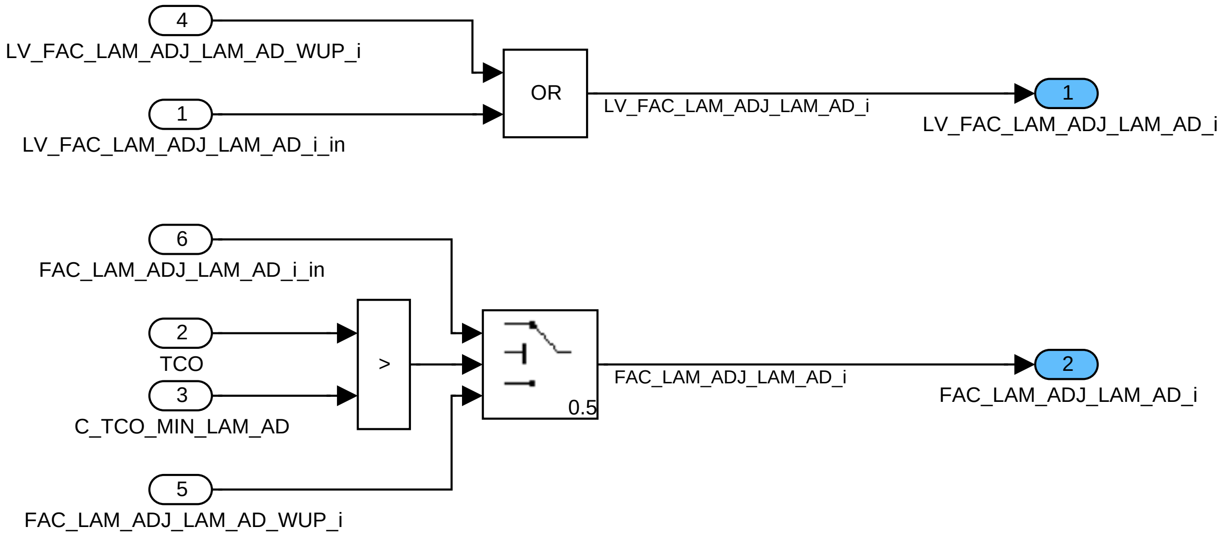


Figure 128 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ LAM_IF/ LAM_AD_WUP

Check of Requirements for Time Scheduler

In order to observe the priority of further lambda adaptations, the differences between the old and the new adapted factors are observed. According to the observed deviations the new maximum time for next adaptation is calculated regarding the present conditions.

The output value for the time scheduler T_PRI_TOT_LAM_AD is not bank selective and represents the minimum time of all cylinder banks. In case of single exhaust cylinder bank T_PRI_TOT_LAM_AD is equal to T_PRI_LAM_AD[i] of this single bank.

If at least for one exhaust bank LV_LAM_AD_STOP[i] is set then the auxiliary flag LV_LAM_AD_STOP_CBK_EX is set to 1.

If the Lambda adaptation priority of any bank is below a limit (that means that the priority time is above a calibration threshold), the bank selective temporary Lambda adaptation end flag is set. When T_PRI_LAM_AD[i] exceeds C_T_PRI_MAX_LAM_AD the flag LV_LAM_AD_END_CBK[i] is set.

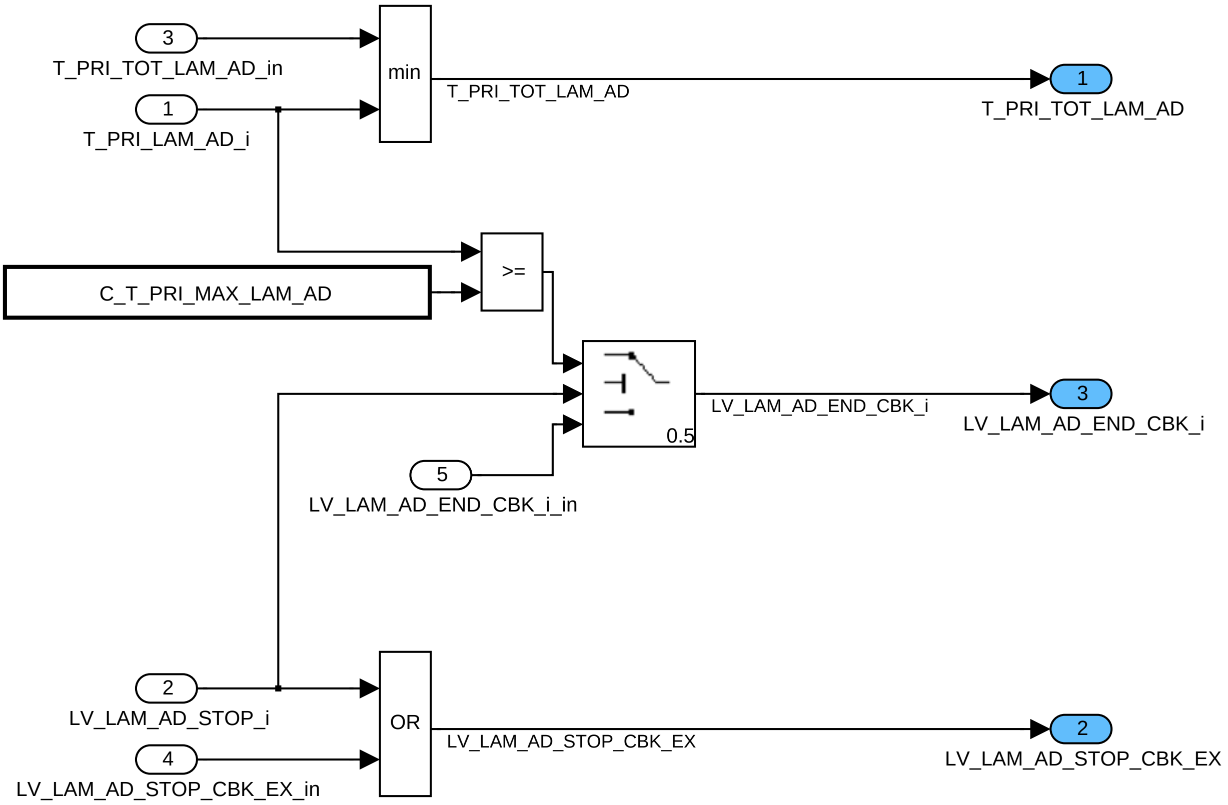


Figure 129 LACO_ISPCLADA0/ OPM_20MS/ CBK_MNG/ CBK_FLP/ CLC_CMN/ T_PRI_MNG

Not exhaust bank specific calculations

Set Lambda Adaptation End Flag

If the total lambda adaptation priority of both exhaust banks is below a limit (that means that the priority time is above a calibration threshold), the temporary lambda adaptation end flag is set. When T_PRI_TOT_LAM_AD exceeds C_T_PRI_MAX_LAM_AD the flag LV_LAM_AD_END is set.

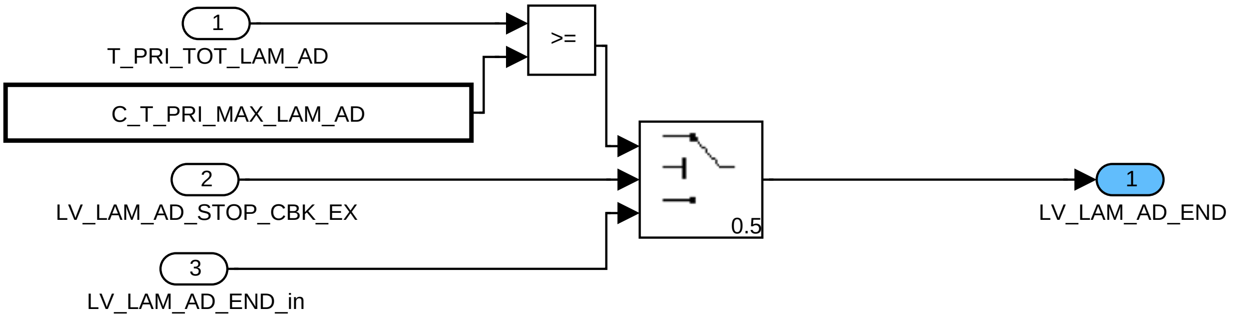


Figure 130 LACO_ISPCLADA0/ OPM_20MS/ NOT_CBK_SPC/ SET_LAM_AD_END

46.13 Lambda adaptation in warm up phase

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_LAM_AD_WUP[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
fuel mass set point factor of lambda adaptation in warm-up phase					
FAC_LAM_ADJ_LAM_AD_WUP[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
output value from lambda adaptation in warm-up phase for the lambda controller shift					
LV_FAC_LAM_ADJ_LAM_AD_WUP[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
flag to request lambda controller shift from Lambda adaptation in warm-up phase					
FAC_LAM_DIF_LAM_AD_WUP_MV[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
average value of Lambda controller output for Lambda adaptation in warm-up phase					
CRLC_FAC_LAM_TCO_X[NC_CBK_EX_NR]	V	0...80H	0...1	0.0078125	-
correlation factor for updating FAC_LAM_TCO_X					
CTR_TCO_MIN_INTER[NC_CBK_EX_NR]	V	0...FFH	0...255	1	-
counter for detection of TCO_MIN_INTER_LAM_AD_WUP					
T_LAM_AD_WUP_TRA_PHA[NC_CBK_EX_NR]	V	0...FFFFH	0...6.5535E+3	0.1	s
timer indicating transient phase of Lambda adaptation in warm-up phase					
LV_CTR_MAX_TCO_MIN_INTER[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that counter based criteria for transient phase has expired					
LV_T_MAX_LAM_AD_WUP_TRA_PHA[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that time dependent criteria for transient phase has expired					
T_LAM_AD_WUP_LDC[NC_CBK_EX_NR]	V	0...FFFFH	0...6.5535E+3	0.1	s
timer to have valid limited dynamic phase for Lambda adaptation in warm-up phase (partial load)					
TCO_MIN_INTER_LAM_AD_WUP[NC_CBK_EX_NR]	V	0...FEH	-48...142.5	0.75	°C
minimum TCO of current driving cycles where learning conditions for Lambda adaptation in warm-up phase are fulfilled					
LV_TCO_MIN_INTER_LAM_AD_WUP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that learning conditions for Lambda adaptation value in warm-up phase are fulfilled					
LV_LOAD_VLD[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that conditions for valid learning are fulfilled					
LV_TCO_MIN_LAM_AD_WUP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that Lambda adaptation value in warm-up phase at TCO_MIN_LAM_AD_WUP was updated					
LV_TCO_A_LAM_AD_WUP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that Lambda adaptation value in warm-up phase at C_TCO_A_LAM_AD_WUP was updated					
LV_TCO_B_LAM_AD_WUP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that Lambda adaptation value in warm-up phase at C_TCO_B_LAM_AD_WUP was updated					
LV_TCO_C_LAM_AD_WUP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that Lambda adaptation value in warm-up phase at C_TCO_C_LAM_AD_WUP was updated					
LV_TCO_D_LAM_AD_WUP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that Lambda adaptation value in warm-up phase at C_TCO_D_LAM_AD_WUP was updated					
LV_TCO_E_LAM_AD_WUP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that Lambda adaptation value in warm-up phase at C_TCO_E_LAM_AD_WUP was updated					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_LAM_AD_WUP_SHIFT[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
shift value of lambda adaptation in warm-up phase after 1st learning					
FAC_BEG_L_RNG_LAM_AD[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
Lambda adaptation value in low range after reset and after auto shift					
FAC_LAM_ADD_CHG_FQ_DET[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.9984741	0.00152588	%
additive shift of adaptation values if change in fuel quality is detected					
LV_LAM_AD_WUP_CHG_FQ_DET[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
flag indicating that change in fuel quality was detected in this driving cycle					
TCO_MIN_LAM_AD_WUP[NC_CBK_EX_NR]	V/S	0...FEH	-48...142.5	0.75	°C
minimum TCO of all driving cycles where learning conditions for Lambda adaptation in warm-up phase (partial load) are fulfilled					
FAC_LAM_TCO_MIN[NC_CBK_EX_NR]	V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
Lambda controller output at TCO_MIN_LAM_AD_WUP					
TCO_A_LAM_AD_WUP[NC_CBK_EX_NR]	V/S	0...FEH	-48...142.5	0.75	°C
actual coolant temperature in range around point A at which adaptation value was learnt					
FAC_LAM_TCO_A[NC_CBK_EX_NR]	V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
Lambda controller output at C_TCO_A_LAM_AD_WUP					
TCO_B_LAM_AD_WUP[NC_CBK_EX_NR]	V/S	0...FEH	-48...142.5	0.75	°C
actual coolant temperature in range around point B at which adaptation value was learnt					
FAC_LAM_TCO_B[NC_CBK_EX_NR]	V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
Lambda controller output at C_TCO_B_LAM_AD_WUP					
TCO_C_LAM_AD_WUP[NC_CBK_EX_NR]	V/S	0...FEH	-48...142.5	0.75	°C
actual coolant temperature in range around point C at which adaptation value was learnt					
FAC_LAM_TCO_C[NC_CBK_EX_NR]	V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
Lambda controller output at C_TCO_C_LAM_AD_WUP					
TCO_D_LAM_AD_WUP[NC_CBK_EX_NR]	V/S	0...FEH	-48...142.5	0.75	°C
actual coolant temperature in range around point D at which adaptation value was learnt					
FAC_LAM_TCO_D[NC_CBK_EX_NR]	V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
Lambda controller output at C_TCO_D_LAM_AD_WUP					
TCO_E_LAM_AD_WUP[NC_CBK_EX_NR]	V/S	0...FEH	-48...142.5	0.75	°C
actual coolant temperature in range around point E at which adaptation value was learnt					
FAC_LAM_TCO_E[NC_CBK_EX_NR]	V/S	8000...7FFFH	-50...49.9984741	0.00152588	%
Lambda controller output at C_TCO_E_LAM_AD_WUP					
LV_LAM_AD_WUP_STOP[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
logical variable indicating need of Lambda controller shift to mirror shift in Lambda adaptation in warm-up phase					

Input data:

LV_LAM_AD_DEAC_ERR[NC_CBK_EX_NR]	LV_LAM_AD_WUP_ACT[NC_CBK_EX_NR]	LV_LDC_LAM_AD[NC_CBK_EX_NR]	LV_LAM_STOP[NC_CBK_EX_NR]
STATE_LS[NC_CBK_EX_NR]	LV_FAC_LAM_ADJ_LAM_AD_END[NC_CBK_EX_NR]	LV_LAM_LSCL[NC_CBK_EX_NR]	FAC_LAM_LIM[NC_CBK_EX_NR]
NC_CBK_EX_NR	TCO	TCO_ST	N_32
MFF_SP[NC_CBK_EX_NR]	FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	FAC_DELTA_L_RNG_LAM_AD[NC_CBK_EX_NR]	MFF_ADD_LAM_AD[NC_CBK_EX_NR]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_FAC_DIF_LAM_AD_WUP_MV	1	0...80H	0...1	0.0078125	-
correlation constant for calculation of the controller deviation in warm-up phase					
C_CRLC_FAC_LAM_TCO_X	1	0...80H	0...1	0.0078125	-
correlation constant for updating FAC_LAM_TCO_X					
C_CRLC_FAC_LAM_TCO_X_FIRST_DC	1	0...80H	0...1	0.0078125	-
correlation constant for updating FAC_LAM_TCO_X in first driving cycle					
C_CRLC_GLOBAL_FAC_LAM_AD_WUP	1	0...80H	0...1	0.0078125	-
correlation constant for calculation of adaptation value in warm-up phase					
C_CTR_TCO_MIN_INTER	1	0...1FFFH	0...8.191E+3	1	-
counter threshold for detecting minimum temperature for valid learning of adapatation value in warm-up phase for current driving cycle					
C_FAC_LAM_AD_WUP_DELTA	1	0...7FFFH	0...49.9984741	0.00152588	%
delta of applied adaptation value in warm-up phase to ramp the signal to new value					
C_FAC_LAM_AD_WUP_MAX_INTER	1	8000...7FFFH	-50...49.9984741	0.00152588	%
maximum value for adaptation value in warm-up phase while in transient phase before valid learning is possible					
C_FAC_LAM_AD_WUP_MAX_LSCL_OFF	1	8000...7FFFH	-50...49.9984741	0.00152588	%
maximum value for adaptation value in warm-up phase while Lambda controller is off					
C_FAC_LAM_AD_WUP_MIN_GAP	1	0...7FFFH	0...49.9984741	0.00152588	%
minimum (absolute) gap between old and new adaptation value in warm-up phase necessary to update adaptation values					
C_FAC_LAM_AD_WUP_MIN_INTER	1	8000...7FFFH	-50...49.9984741	0.00152588	%
minimum value for adaptation value in warm-up phase while in transient phase before valid learning is possible					
C_FAC_LAM_AD_WUP_MIN_LSCL_OFF	1	8000...7FFFH	-50...49.9984741	0.00152588	%
minimum value for adaptation value in warm-up phase while Lambda controller is off					
C_MFF_SP_TOL_LAM_AD_WUP	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
maximum fuel mass set point for lambda adaptation in warm-up phase (partial speed)					
C_N_BOL_LAM_AD_WUP	1	0...FFH	0...8.16E+3	32	rpm
minimum engine speed for lambda adaptation in warm-up phase (partial load)					
C_N_TOL_LAM_AD_WUP	1	0...FFH	0...8.16E+3	32	rpm
maximum engine speed for lambda adaptation in warm-up phase (partial load)					
C_TCO_A_LAM_AD_WUP	1	0...FEH	-48...142.5	0.75	°C
coolant temperature of point A for Lambda adaptation in warm-up phase					
C_TCO_B_LAM_AD_WUP	1	0...FEH	-48...142.5	0.75	°C
coolant temperature of point B for Lambda adaptation in warm-up phase					
C_TCO_C_LAM_AD_WUP	1	0...FEH	-48...142.5	0.75	°C
coolant temperature of point C for Lambda adaptation in warm-up phase					
C_TCO_D_LAM_AD_WUP	1	0...FEH	-48...142.5	0.75	°C
coolant temperature of point D for Lambda adaptation in warm-up phase					
C_TCO_E_LAM_AD_WUP	1	0...FEH	-48...142.5	0.75	°C
coolant temperature of point E for Lambda adaptation in warm-up phase					
C_TCO_F_LAM_AD_WUP	1	0...FEH	-48...142.5	0.75	°C
coolant temperature of point F for Lambda adaptation in warm-up phase					
C_TCO_LAM_AD_WUP_RNG	1	0...FE00H	0...190.5	0.00292969	°C
temperature window for learning of adaptations value in warm-up phase					
C_TCO_RNG_WR_DOWN_LAM_AD_WUP	1	0...FE00H	0...190.5	0.00292969	°C
temperature window in which first learnt adaptation value can be written down to next lower point					
C_THD_DIF_L_RNG_LAM_AD	1	8000...7FFFH	-50...49.9984741	0.00152588	%
threshold value of change in low range adaptation factor to force auto shift					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_THD_FAC_DELTA_L_RNG	1	8000...7FFFH	-50...49.9984741	0.00152588	%
threshold of lambda adaptation difference in low range to allow auto shift					
C_T_MAX_LAM_AD_WUP_LDC	1	0...FFFFH	0...6.5535E+3	0.1	s
threshold for elasped time with limited dynamics for learning of adaptation value in warm-up phase under load					
C_T_MAX_LAM_AD_WUP_TRA_PHA	1	0...FFFFH	0...6.5535E+3	0.1	s
threshold for timer indicating transient phase of Lambda adaptation in warm-up phase					
LC_FAC_RAW_LAM_AD_WUP_LIM_ACT	1	0...1H	0...1	1	-
logical calibration to force FAC_LAM_AD_WUP to FAC_LAM_TCO_MIN at temperature below TCO_MIN					
LC_FAC_WR_DOWN_LAM_AD_WUP	1	0...1H	0...1	1	-
First learnt adaptation values shall be written down to next lower interpolation					
LC_LAM_AD_WUP_CLR_INH	1	0...1H	0...1	1	-
Inhibit initialization of NVMY data (for application purposes only)					
LC_LAM_AD_WUP_SHIFT_ACT	1	0...1H	0...1	1	-
logical calibration to enable auto shift in Lambda adaptation in warm-up phase in all driving cycles					
ID_FAC_THD_CHG_FQ_DET	6	0...FFFFH	-50...49.9984741	0.00152588	%
LDPM_TCO_3_LACO	6	0...FEH	-48...142.5	0.75	°C
threshold of difference in Lambda adaptation to detect change in fuel quality during warm-up phase					
ID_MFF_SP_BOL_LAM_AD_WUP	4	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDP_N_32_ID_MFF_BOL_LAM_AD_WUP	4	0...FFH	0...8.16E+3	32	rpm
minimum fuel mass set point for lambda adaptation in warm-up phase (partial load)					
IP_FAC_CHG_FQ_DET_SCA	6	0...80H	0...1	0.0078125	-
LDPM_TCO_3_LACO	6	0...FEH	-48...142.5	0.75	°C
scaling factor to shift adaptation values in warm-up phase after detection of change in fuel quality					
IP_FAC_LAM_AD_WUP_MAX	6	0...FFFFH	-50...49.9984741	0.00152588	%
LDPM_TCO_3_LACO	6	0...FEH	-48...142.5	0.75	°C
maximum value for adaptation value in warm-up phase					
IP_FAC_LAM_AD_WUP_MIN	6	0...FFFFH	-50...49.9984741	0.00152588	%
LDPM_TCO_3_LACO	6	0...FEH	-48...142.5	0.75	°C
minimum value for adaptation value in warm-up phase					
IP_FAC_WUP_FAC_LAM_AD_WUP	5x5	0...80H	0...1	0.0078125	-
LDPM_TCO_2_LACO	5	0...FEH	-48...142.5	0.75	°C
LDPM_FAC_LAM_AD_1_LACO	5	0...FFFFH	-50...49.9984741	0.00152588	%
copy of IP_FAC_WUP_FAC_LAM_AD used im Lambda adaptation in warm-up phase					

46.13.1 LACO_ISPCLADAW0

The Lambda adaptation in warm-up phase enhances the Lambda adaptation functionality to obtain a more precise correction at low temperatures (< 50°C). The adaptive correction is applied to compensate serial production tolerances of components at low.

TCO dependent correction factors (%/°C) are learned in five temperature ranges. The correction factors are derived from the Lambda controller output. They are learned in an adaptation field specified by MFF_SP[i] and N_32. They are, however, to correct the injection time for all engine states. Learning of these factors requires that the Lambda controller is in the state 'ON'.

The recurrence time is defined by T_SAMPLE in [ms].

NC_CBK_EX_NR defines the number of exhaust banks. For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

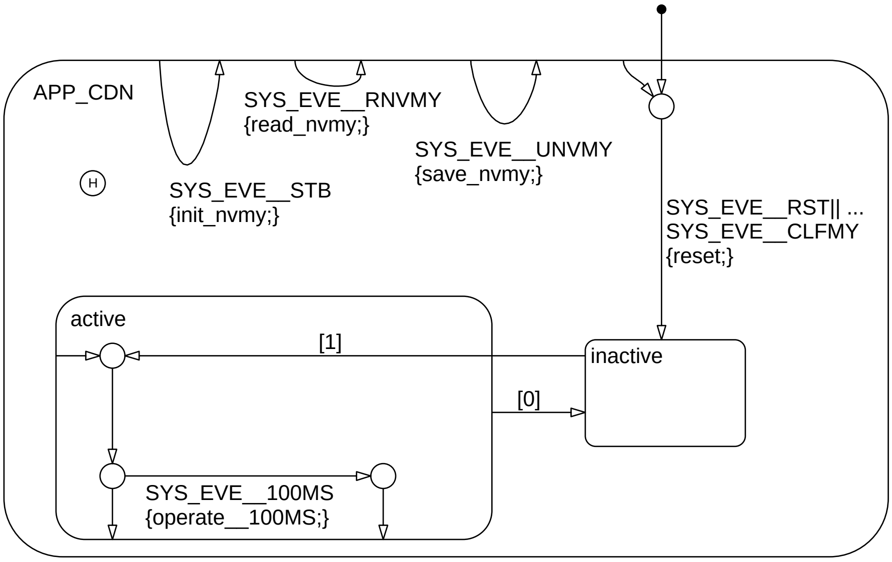


Figure 131 LACO_ISPCLADAW0/ APP_CDN/ Chart

Function Description

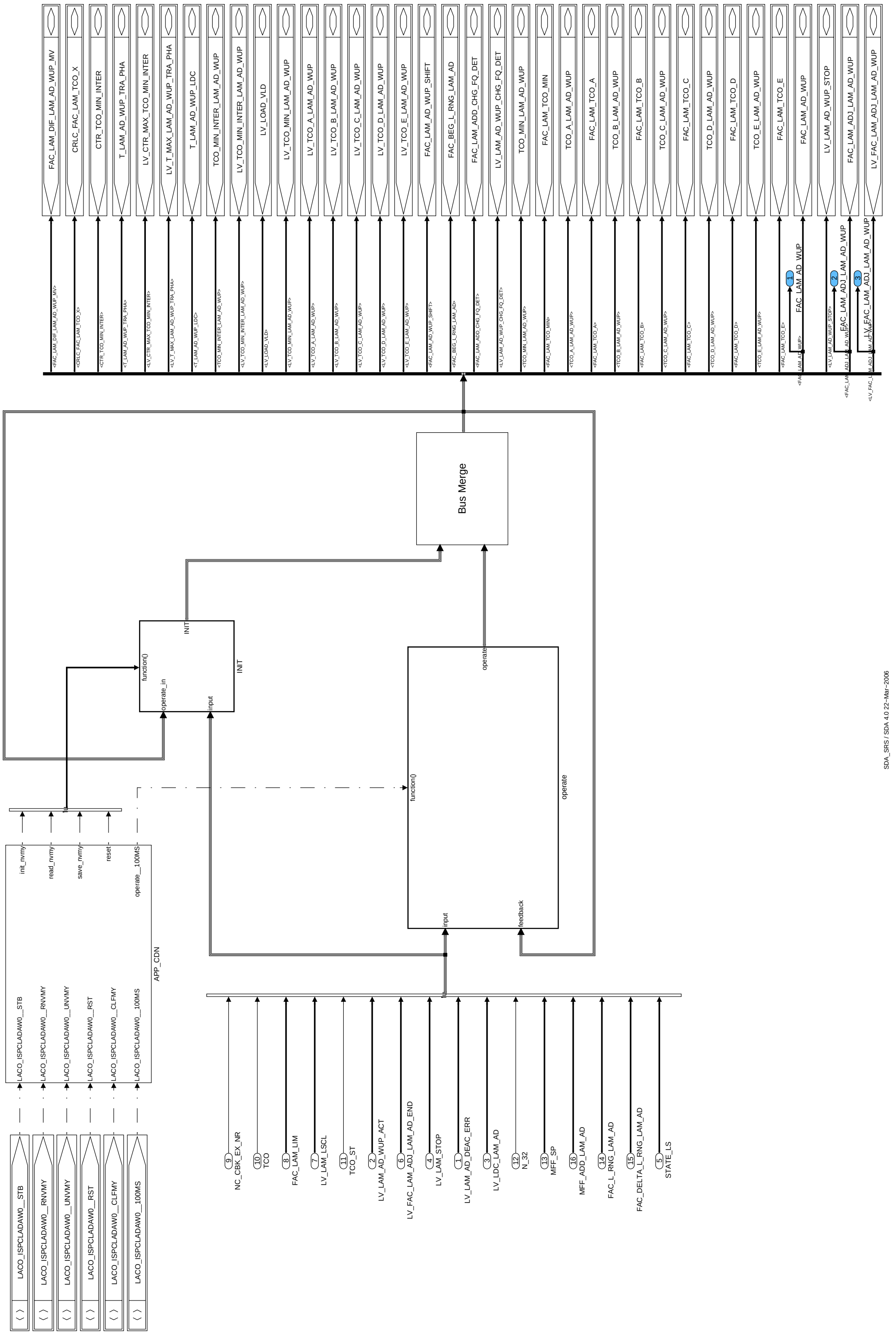


Figure 132 LACO_ISPCLADAW0

46.13.1.1 SUBFUNCTION: operate

Calculation of the uncorrected Lambda controller output

To obtain the actual load on the Lambda controller, the controller output without adaptation is calculated. This is the sum of FAC_LAM_LIM[i] and applied adaptive correction in the warm-up phase. The actual load is then filtered to have a smoother signal. In case of a not yet activated Lambda controller the value is set to zero. This is necessary in order to facilitate the detection of end of the transient phase after the start of the Lambda controller.

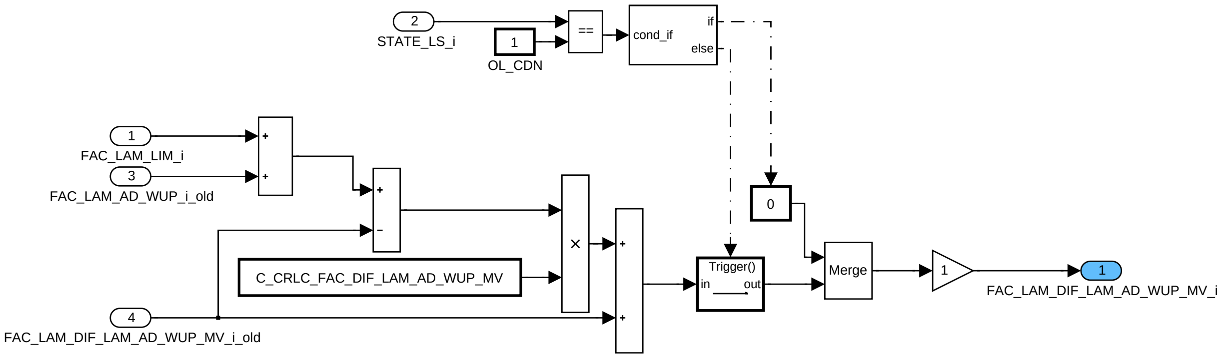


Figure 133 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ FAC_LAM_DIF_LAM_AD_WUP_MV

Correlation constant for updating of adaptation factors

In the update process of the adaptation factor a correlation constant is taken into account. Typically this value will be less then one to reduce the impact of outlier.

In case of a brand new ECU the cold adaptation values need to be shifted, if they are learned before the warm adaptation values are learned. The shift value amounts to the adaptation factor in the low range area. During this shift process no correlation constant is considered. Hence, in case of a brand new ECU the correlation constant is set to 1. Otherwise the auto shift value will not correspond to the corresponding value learned in the cold adaptation.

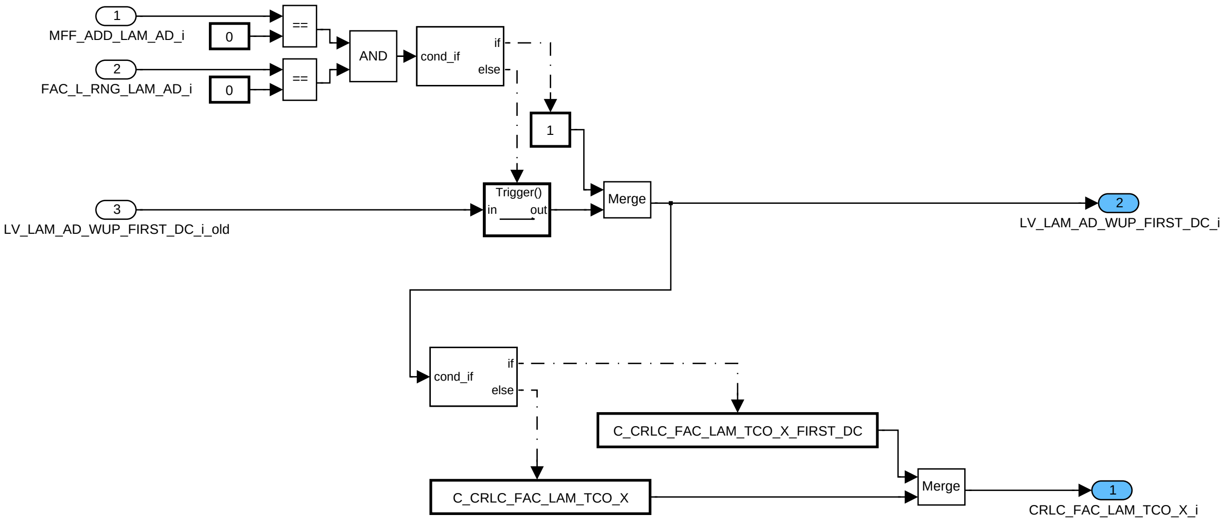


Figure 134 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ CLC_CRLC_LAM_TCO_X

Determination of valid partial load phase

To enable learning of the adaptation factor the conditions for valid learning have be fulfilled for at least the calibratable time C_T_MAX_LAM_AD_WUP_LDC.

The conditions are:

- Limited dynamics of the Lambda adaptation are fulfilled.
- Lambda controller is active.
- Lambda controller is not in the stop mode.
- N_32 is below a threshold, but above another threshold.
- MFF_SP[i] is below a threshold, but above another threshold.

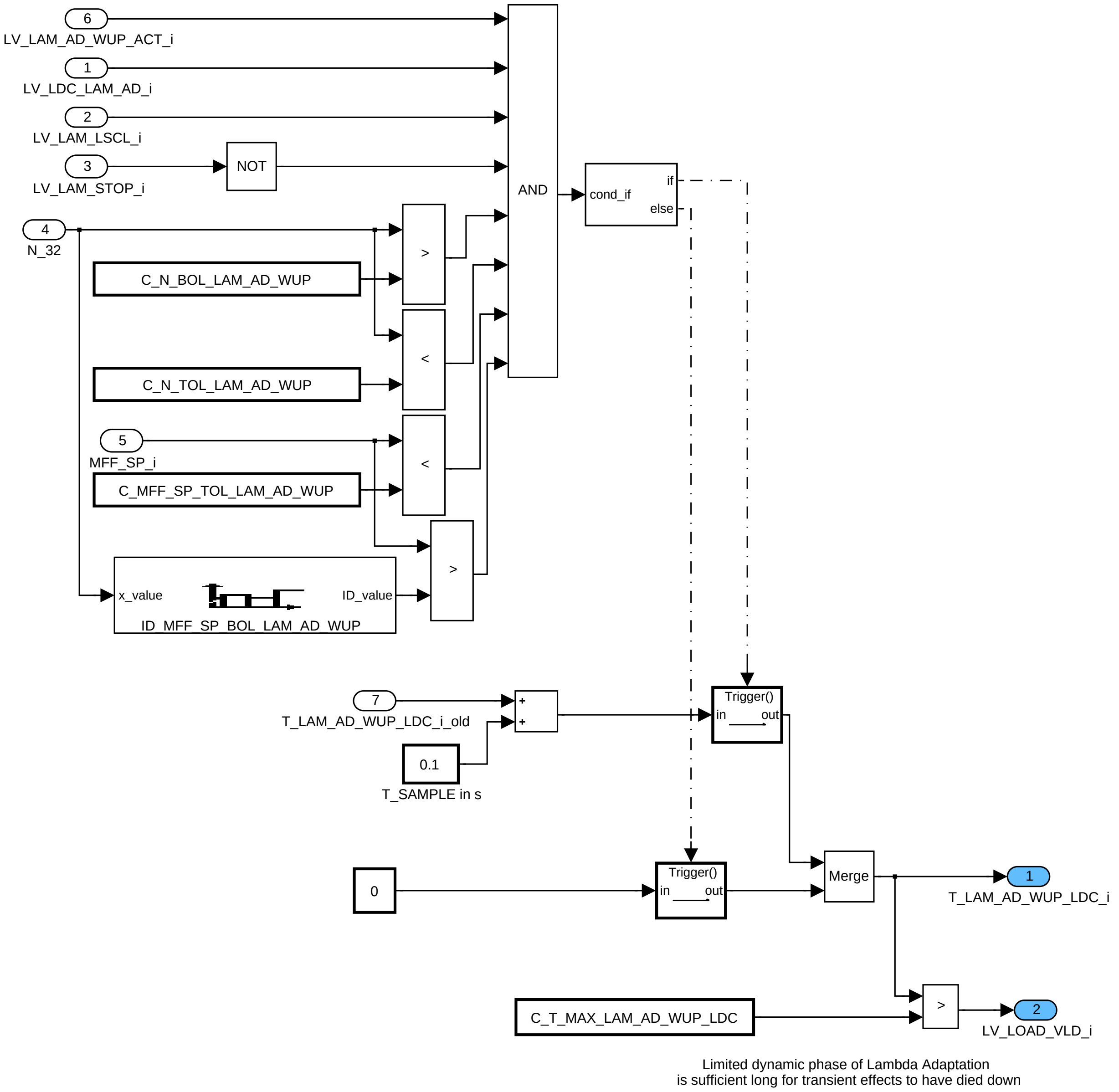


Figure 135 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_MIN_INTER_LAM_AD_WUP/ CLC_T_LAM_AD_WUP_LDC

Determination of minimum temperature valid for learning in current driving cycle

To enable learning in the current driving cycle for the first time the following conditions must be satisfied:

- The counter as well as the timer criteria are fulfilled, thus is transient phase is over.

- The global activation conditions are fulfilled.

- Limited dynamics are given, the Lambda controller is in state on and engine operating point is within the required mass air flow and speed range (see also definition of LV_LOAD_VLD[i]).

- The coolant temperature is within the temperature range of the lowest and highest interpolation points.

If above conditions are fulfilled, then the flag indicating the detection of TCO_MIN_INTER is set.

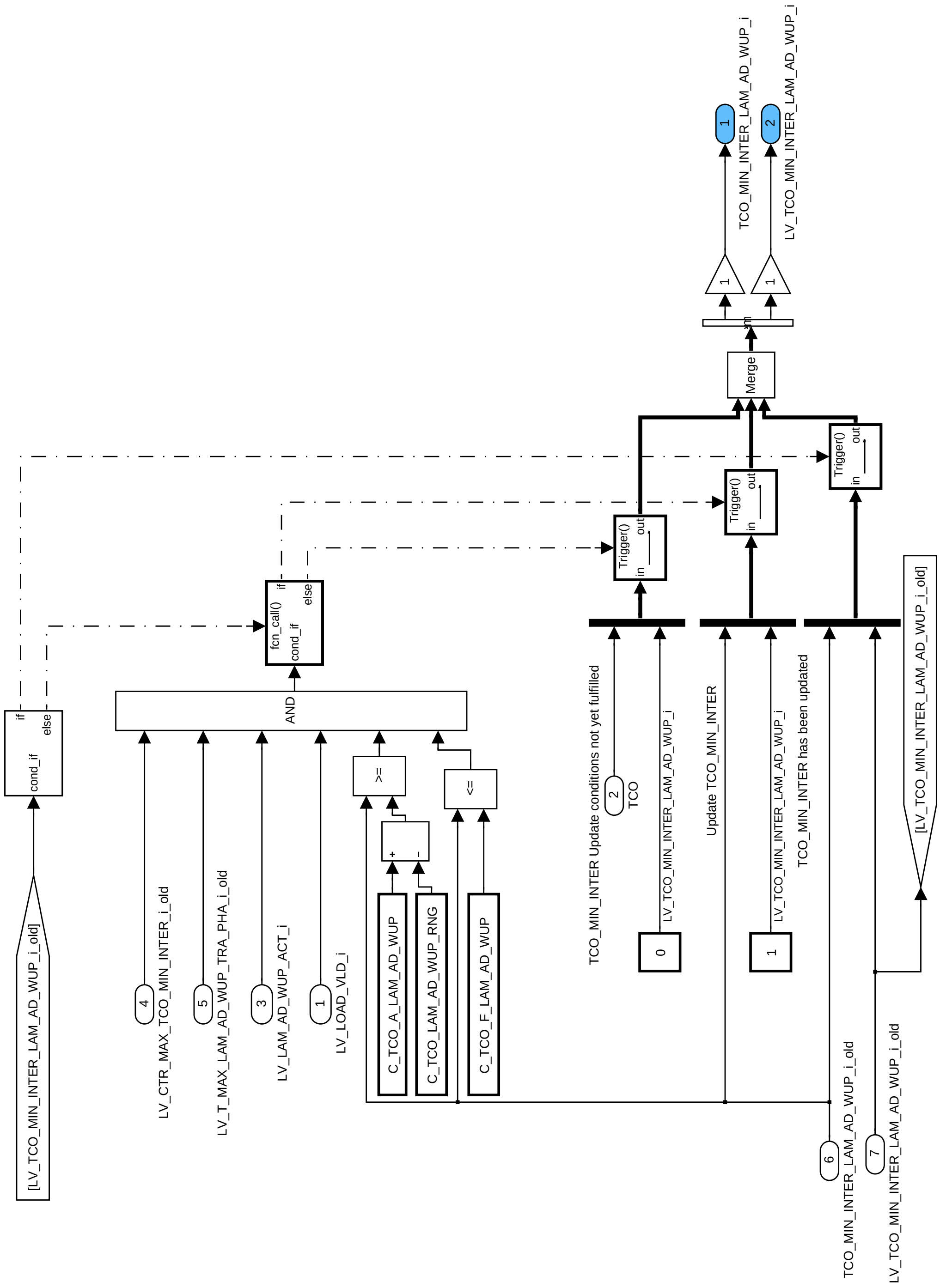


Figure 136 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_MIN_INTER_LAM_AD_WUP/ CLC_TCO_MIN_INTER_LAM_AD_WUP

Determination of end of transient phase

Two different criteria can be used to determine the end of the transient phase:

Timer criteria:

When the Lambda controller enters the state ON a timer is incremented. If this timer exceeds a threshold, it is assumed that the transient is over. This criterion is very simple, but not very accurate.

Counter criteria:

The counter is incremented when the uncorrected Lambda controller moves towards the zero percent line or stays unchanged. In case it moves away from the zero percent line, the counter is decremented. The counter cannot be decremented below zero. If the counter exceeds a certain threshold, the end of the transient phase is detected. This criterion was found to be very accurate.

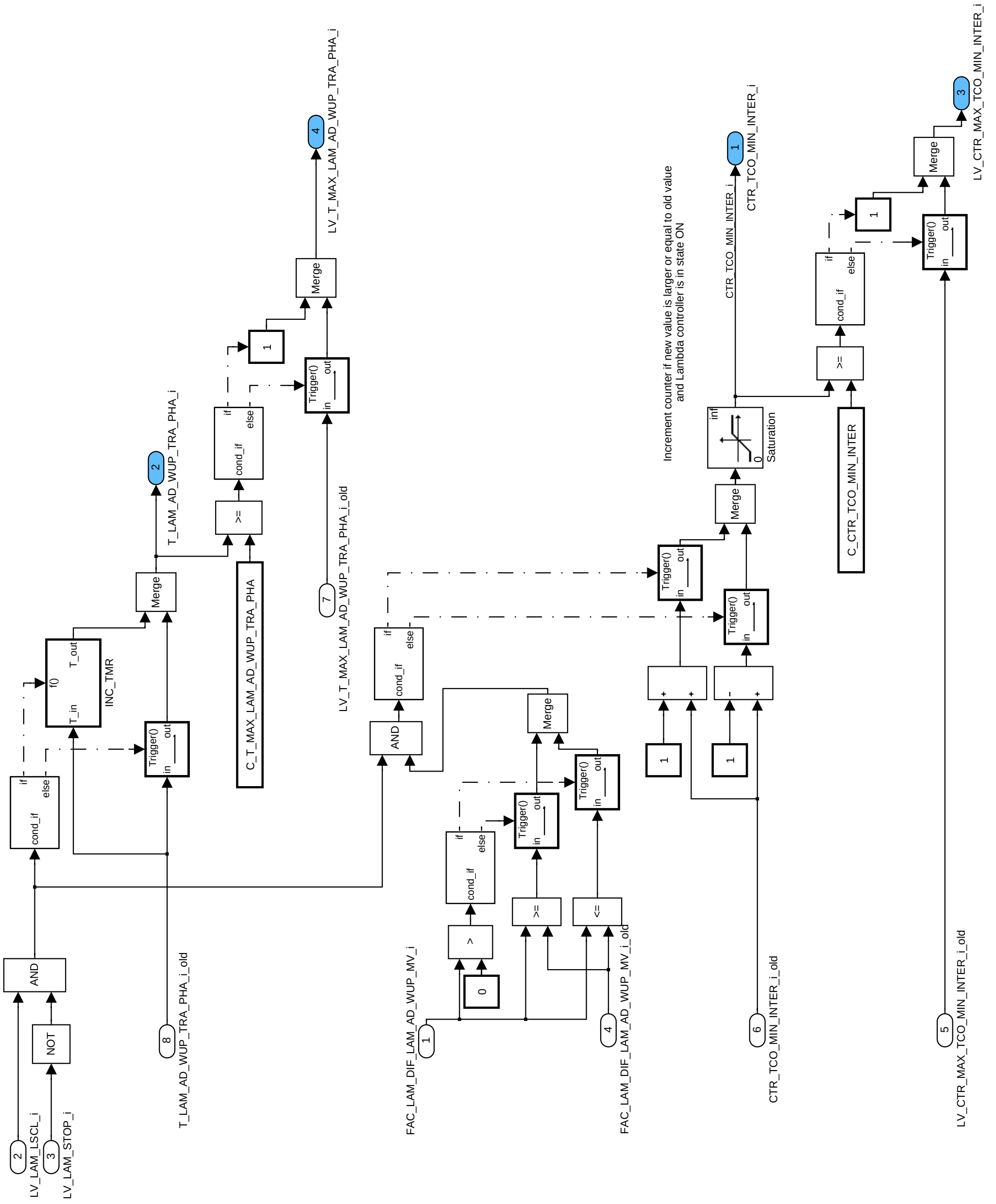


Figure 137 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_MIN_INTER_LAM_AD_WUP/ CLC_CTR_TCO_MIN_INTER

Auto shift functionality – correction of adaptation value with warm adaptation value

In each driving cycle it is checked whether the fuel quality has changed to the worse. This check is done once per driving cycle at the time when the conditions for learning are fulfilled for the first time. A change of fuel quality is detected if the uncorrected Lambda controller output has shifted by at least a calibratable threshold towards lean. The threshold is specified in a map with TCO as an input. This allows taking into account the TCO dependent deterioration effect of poor fuel quality.

Once fuel of low quality is detected, a flag indicating that detection was done in the current driving cycle is set to one. Furthermore, the difference in the uncorrected Lambda controller output (which led to this detection) is normalized. The normalization is done using a TCO dependent map. The same map is used later on to shift all adaptation values. Therefore, the map has to represent the qualitative course of the additional enrichment demand caused by the poor fuel quality.

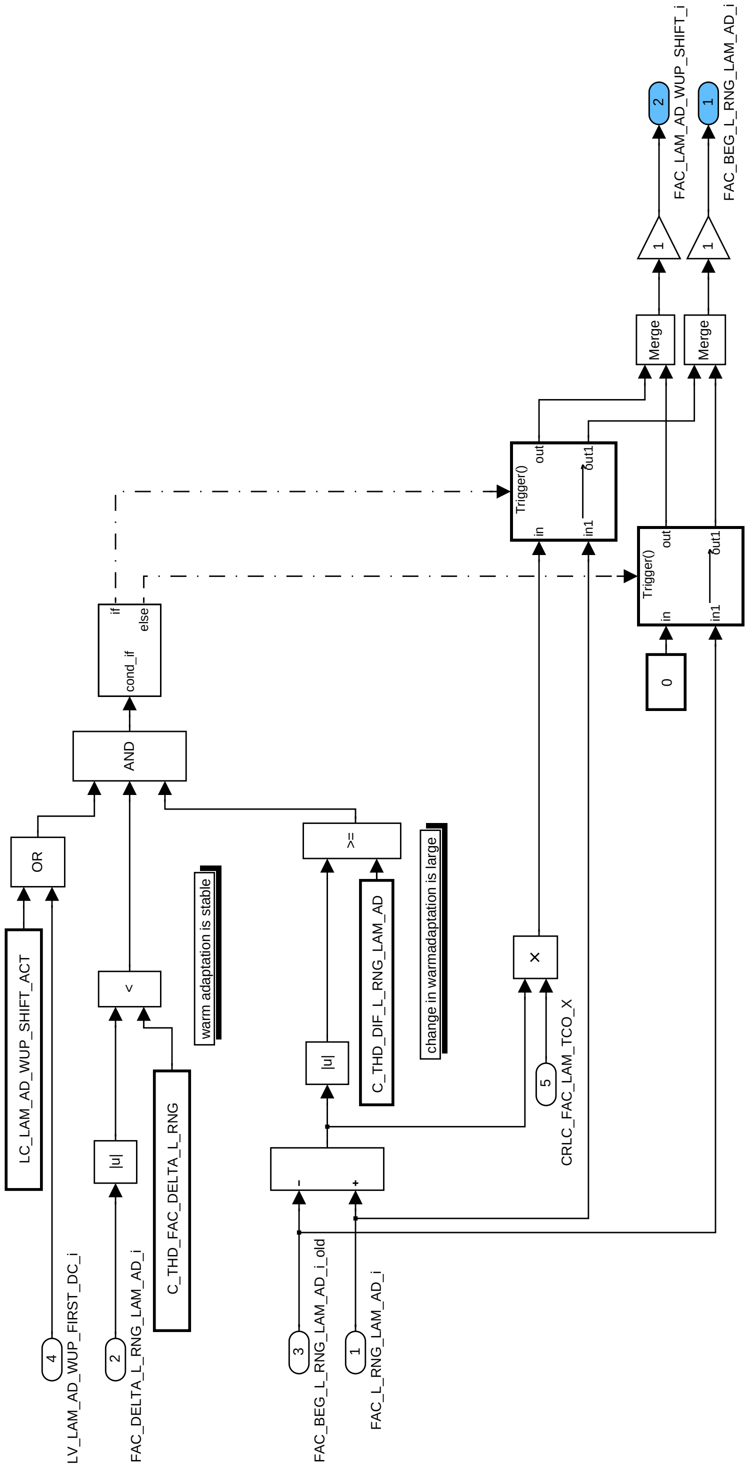


Figure 138 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ CLC_FAC_LAM_AD_WUP_SHIFT

Detection of change in fuel quality

In each driving cycle it is checked whether the fuel quality has changed to the worse. This check is done once per driving cycle at the time when the conditions for learning are fulfilled for the first time. A change of fuel quality is detected if the uncorrected Lambda controller output has shifted by at least a calibratable threshold towards lean. The threshold is specified in a map with TCO as an input. This allows taking into account the TCO dependent deterioration effect of poor fuel quality.

Once fuel of low quality is detected, a flag indicating that detection was done in the current driving cycle is set to one. Furthermore, the difference in the uncorrected Lambda controller output (which led to this detection) is normalized. The normalization is done using a TCO dependent map. The same map is used later on to shift all adaptation values. Therefore, the map has to represent the qualitative course of the additional enrichment demand caused by the poor fuel quality.

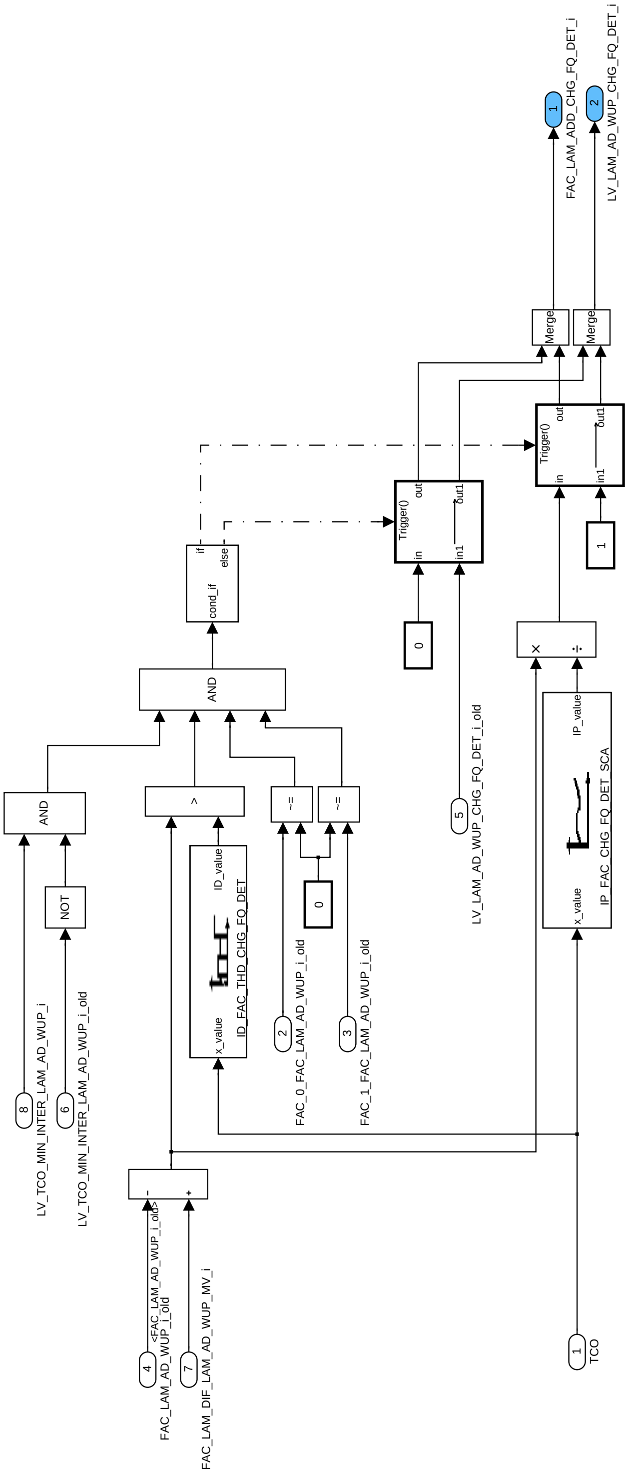


Figure 139 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ CLC_CHG_FQ_DET

<u>Adaptive factor at lowest temperature over all driving cycles</u>

Beside the interpolation points A to F with fixed temperature ranges, a moving interplation point is used. This interpolation point also represents the lowest temperature over all driving cycles at which the learning conditions were fulfilled.

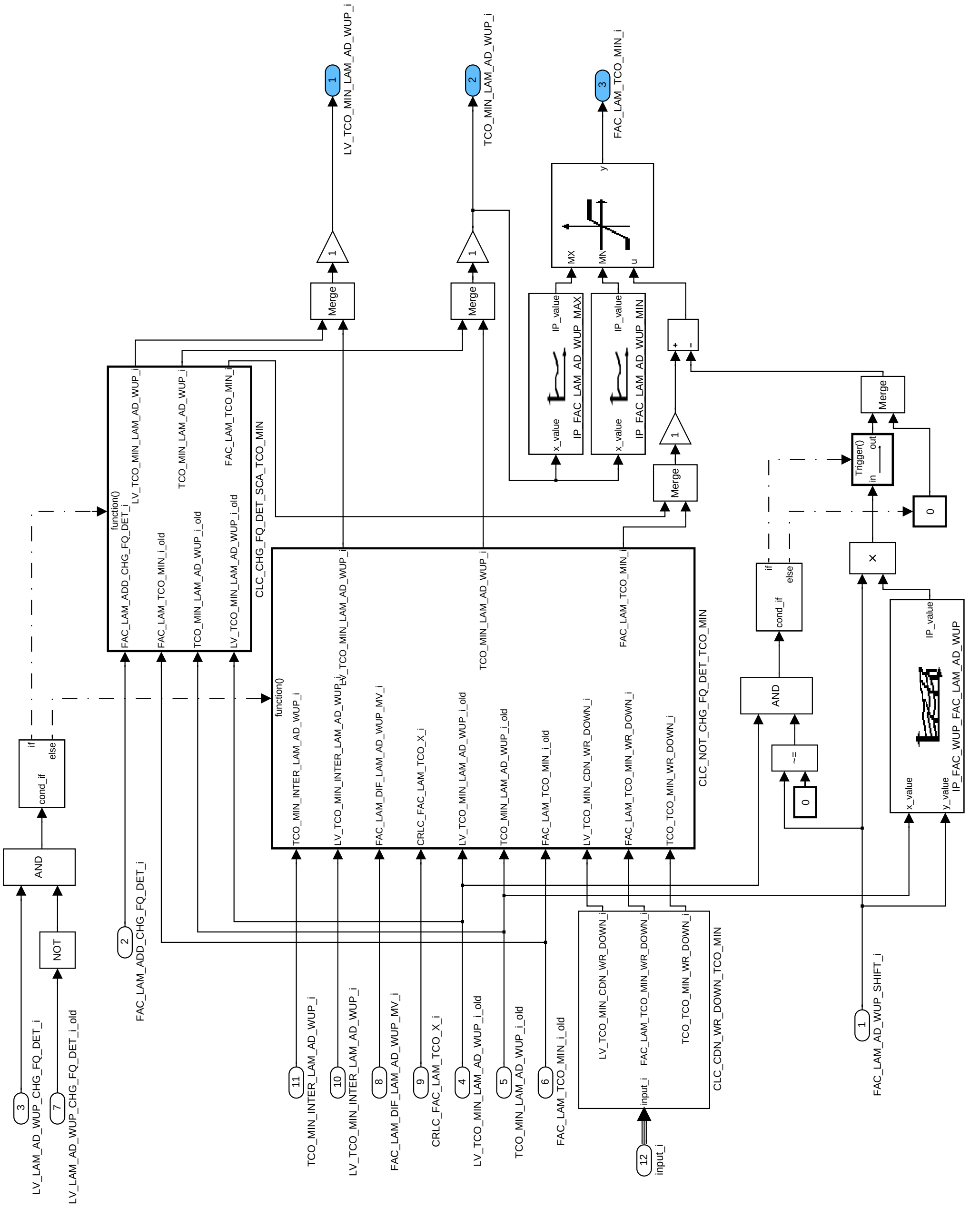


Figure 140 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_MIN_LAM_AD_WUP

Determination of adaptation value during detection of change in fuel quality

First it is checked whether the adaptation factor is different from zero to confirm, that it has been updated at some point in the past. If this is not the case no shift is realized at this particular interpolation point.

For shifting the normalized shift value is multiplied by a TCO dependent weighting factor and then added to the old adaptation factor. The weighting factor takes into account that the negative effects of the low quality fuel disappear with rising temperature. Hence, the weighting factor should be around 1 for the lowest interpolation point and at around zero for the highest temperature.

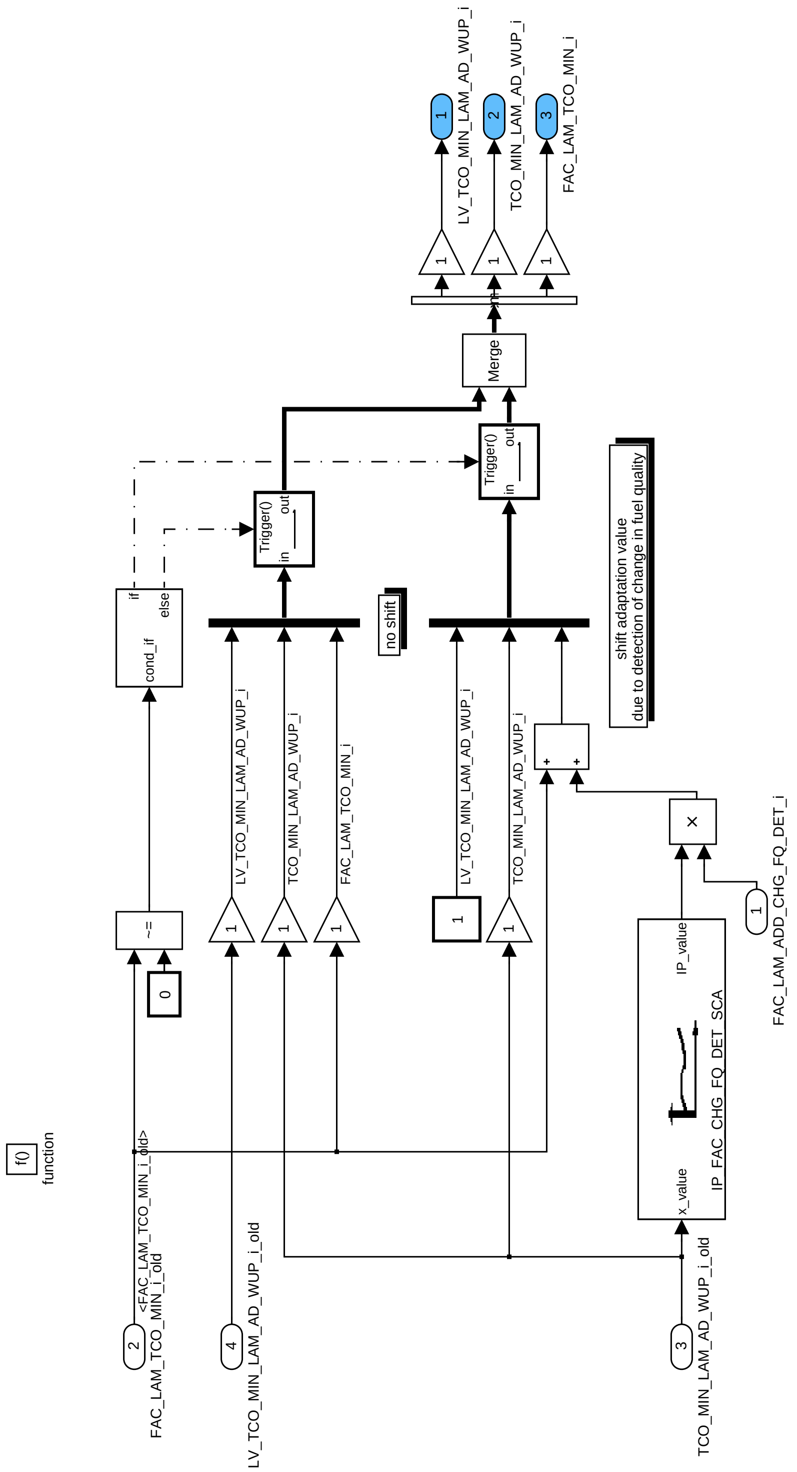


Figure 141 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_MIN_LAM_AD_WUP/ CLC_CHG_FQ_DET_SCA_TCO_MIN

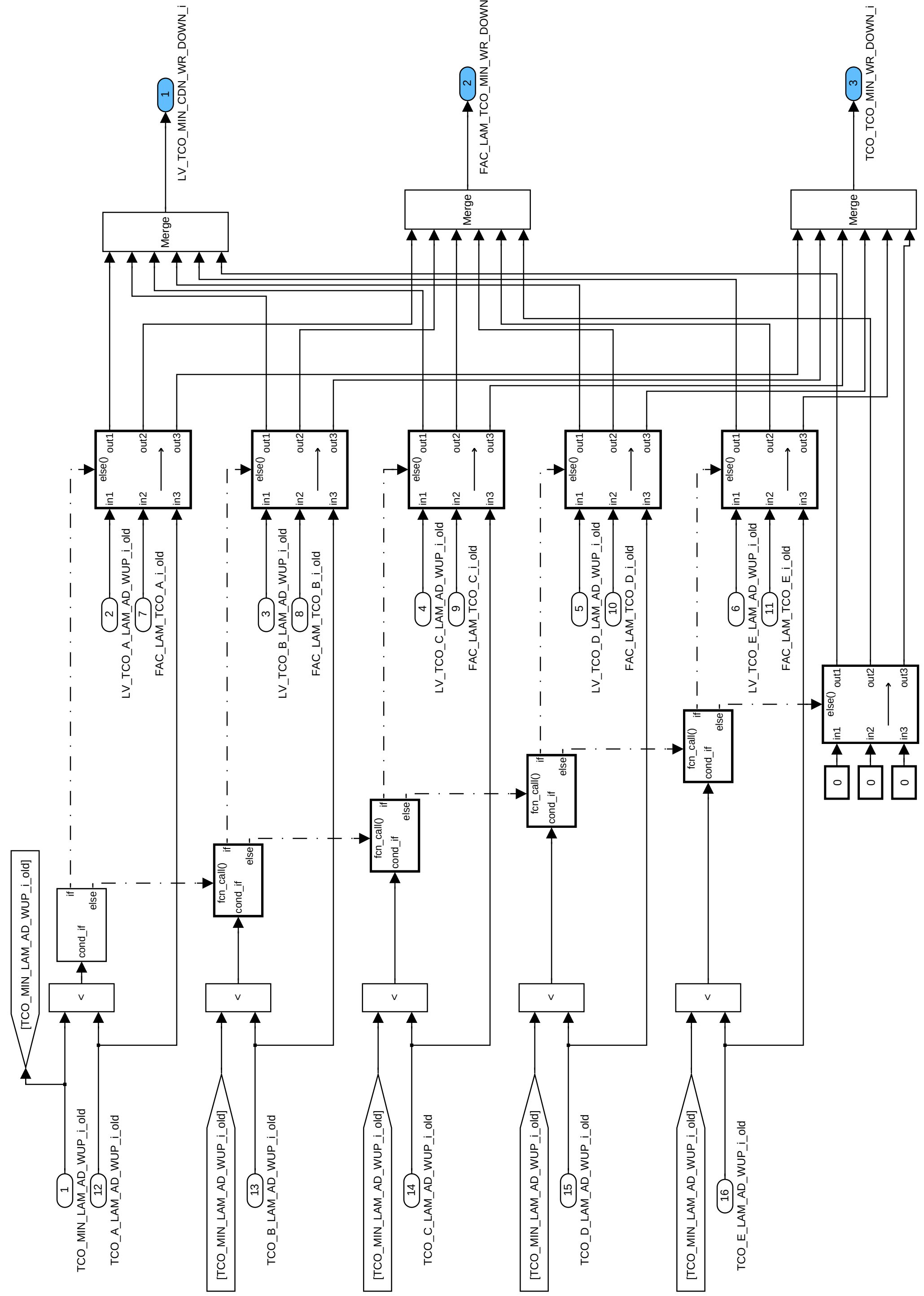


Figure 142 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_MIN_LAM_AD_WUP/ CLC_CDN_WR_DOWN_TCO_MIN/ CLC

<u>Determination of minimum temperature valid for learning of all driving cycles</u>

TCO_MIN is updated with TCO_MIN_INTER if the value for the current driving cycle (TCO_MIN_INTER) is smaller than the previously stored value of TCO_MIN. During update the correlation constant is considered.

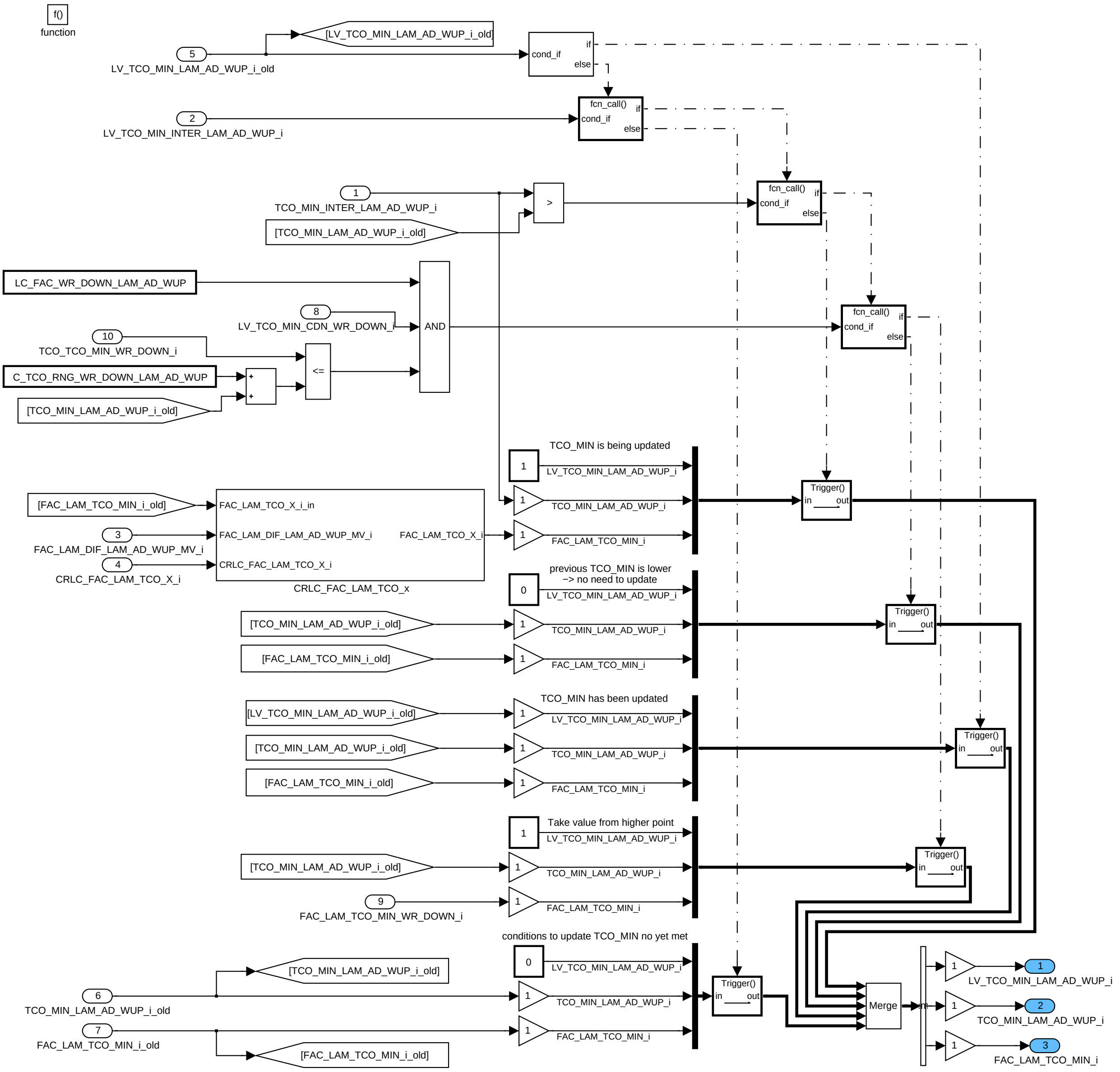


Figure 143 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_MIN_LAM_AD_WUP/ CLC_NOT_CHG_FQ_DET_TCO_MIN

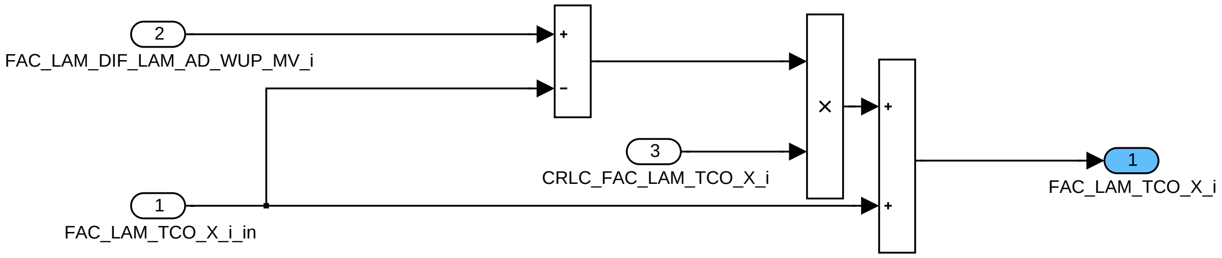


Figure 144 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_MIN_LAM_AD_WUP/ CLC_NOT_CHG_FQ_DET_TCO_MIN/ CRLC_FAC_LAM_TCO_x

Determination of new adaptation value at point A to E

If a change in fuel quality is detected the normal update process of the adaptation factor is inhibited. The auto shift value is always zero with the exception of the calculation recurrence during which the shift is executed. Furthermore, if the adaptation factor was not updated in this driving cycle (LV_TCO_X_LAM_AD_WUP[i]), then the auto shift correction is not applied.

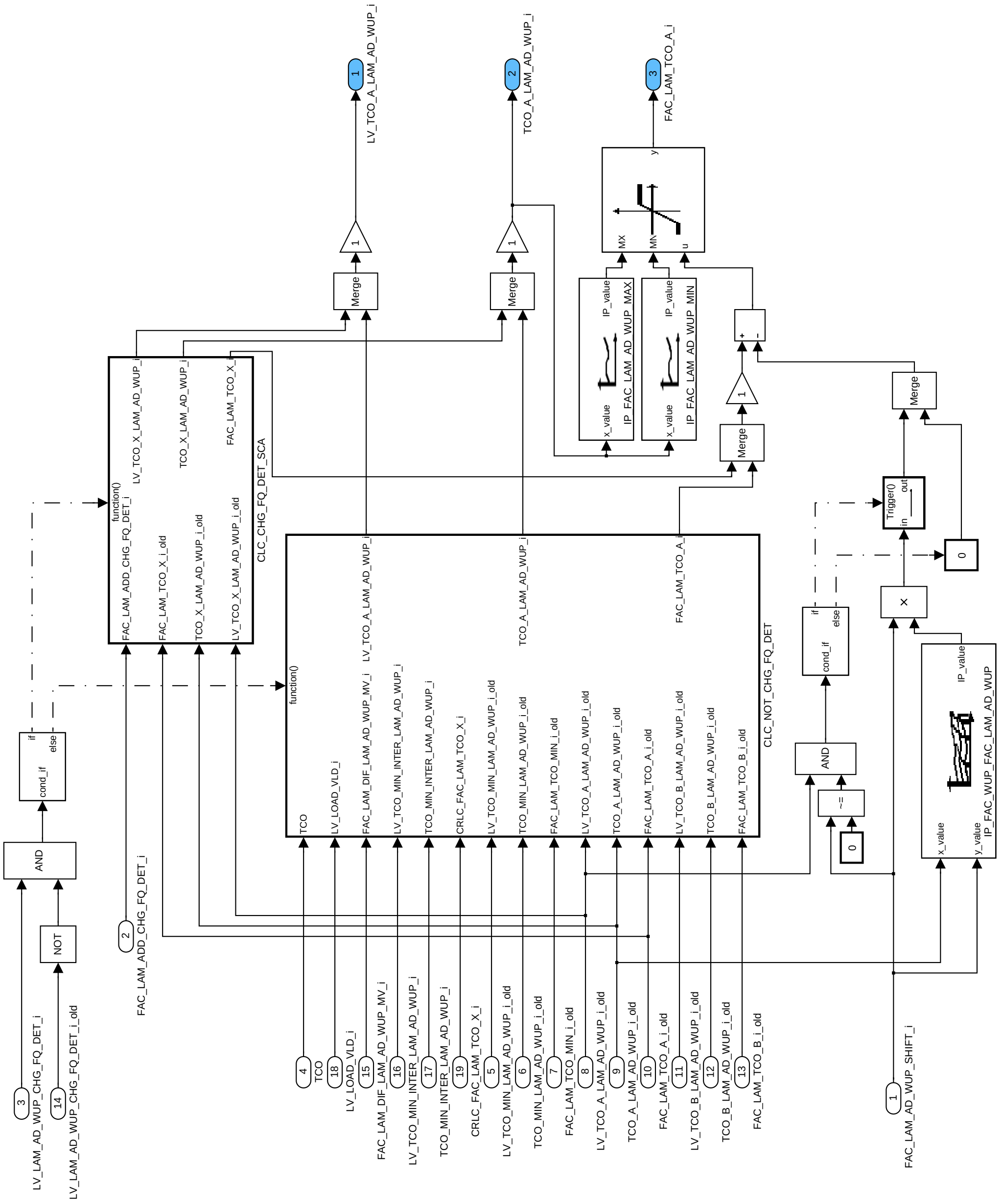


Figure 145 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_A

Determination of adaptation value during detection of change in fuel quality

First it is checked whether the adaptation factor is different from zero to confirm, that it has been updated at some point in the past. If this is not the case no shift is realized at this particular interpolation point.

For shifting the normalized shift value is multiplied by a TCO dependent weighting factor and then added to the old adaptation factor. The weighting factor takes into account that the negative effects of the low quality fuel disappear with rising temperature. Hence, the weighting factor should be around 1 for the lowest interpolation point and at around zero for the highest temperature.

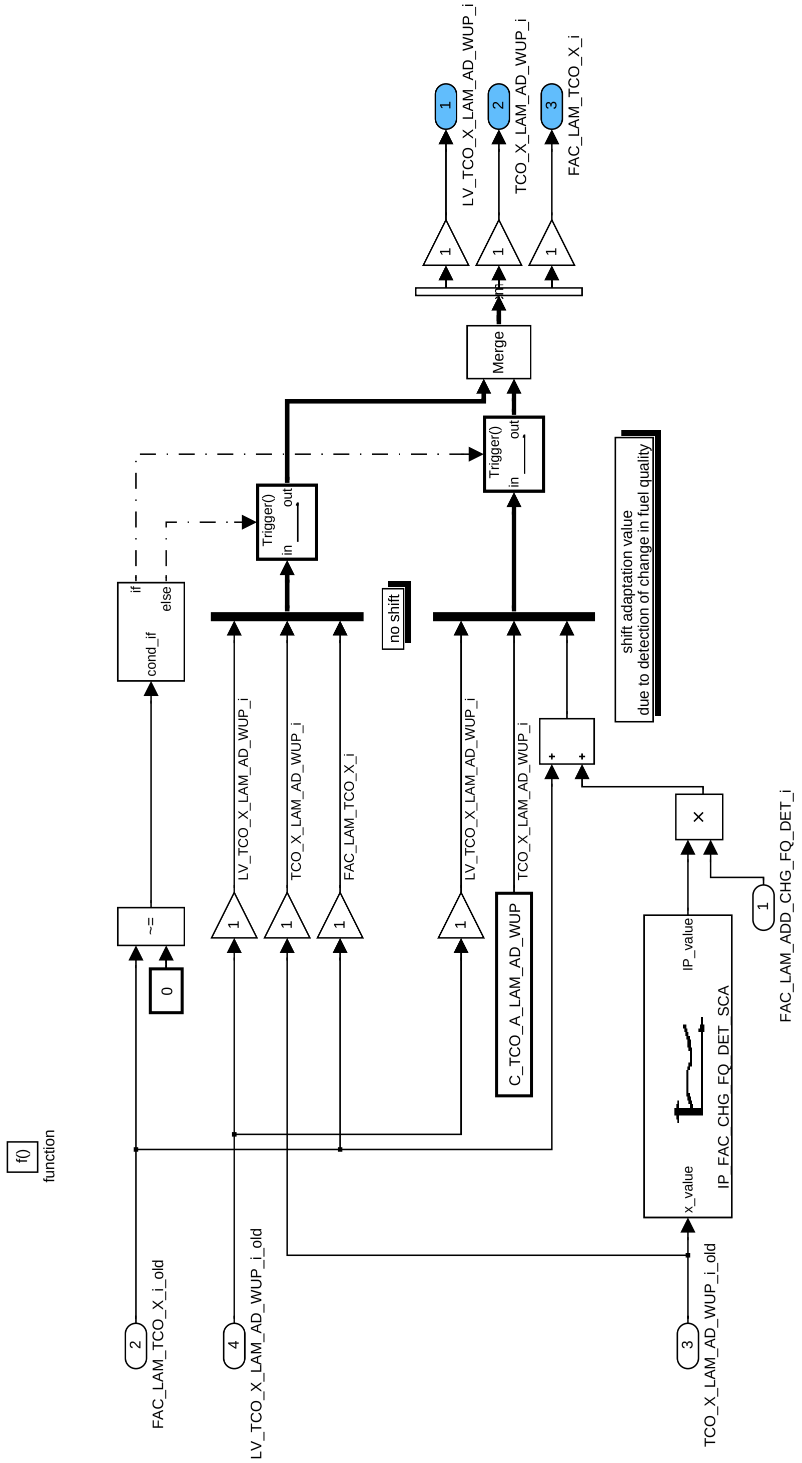


Figure 146 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_A/ CLC_CHG_FQ_DET_SCA

Determination of adaptation value during normal operation

Four different states shall be distinguished:

1.) Conditions to update are not yet fulfilled. Indicator flag remains zero and the factor and the temperature at the interpolation point are not updated.

2.) The adaptation factor the corresponding temperature are updated once the following conditions are met:
- TCO is within the temperature window
- End of transient phase has been detected, thus TCO_MIN_INTER[i] is set
- The conditions for valid learning are fulfilled sufficiently long (LV_LOAD_VLD[i])
- The difference to the previously stored value is sufficiently large

If all above conditions are met, the adaptation factor is updated with the uncorrected Lambda controller output (in this process a correlation factor is considered) and the corresponding temperature is updated with the current TCO. Additionally the indicator flag is set to one.

3.) If TCO_MIN[i] is updated (the lowest ever learnt temperature is reduced) and it is larger than the temperature of the interpolation point, then the adaptation value is updated with factor belonging to TCO_MIN[i]. The corresponding temperature is not updated in this state. This update procedure can be restricted in such that TCO_MIN[i] needs to be within temperature of point of A and point B plus size of the temperature window.

4.) Update was already done in this driving cycle: No actions are undertaken.

The calculation at point B to E proceed analogous. At interpolation point F the correction factor is fixed at zero, in order to fade out the correction smoothly.

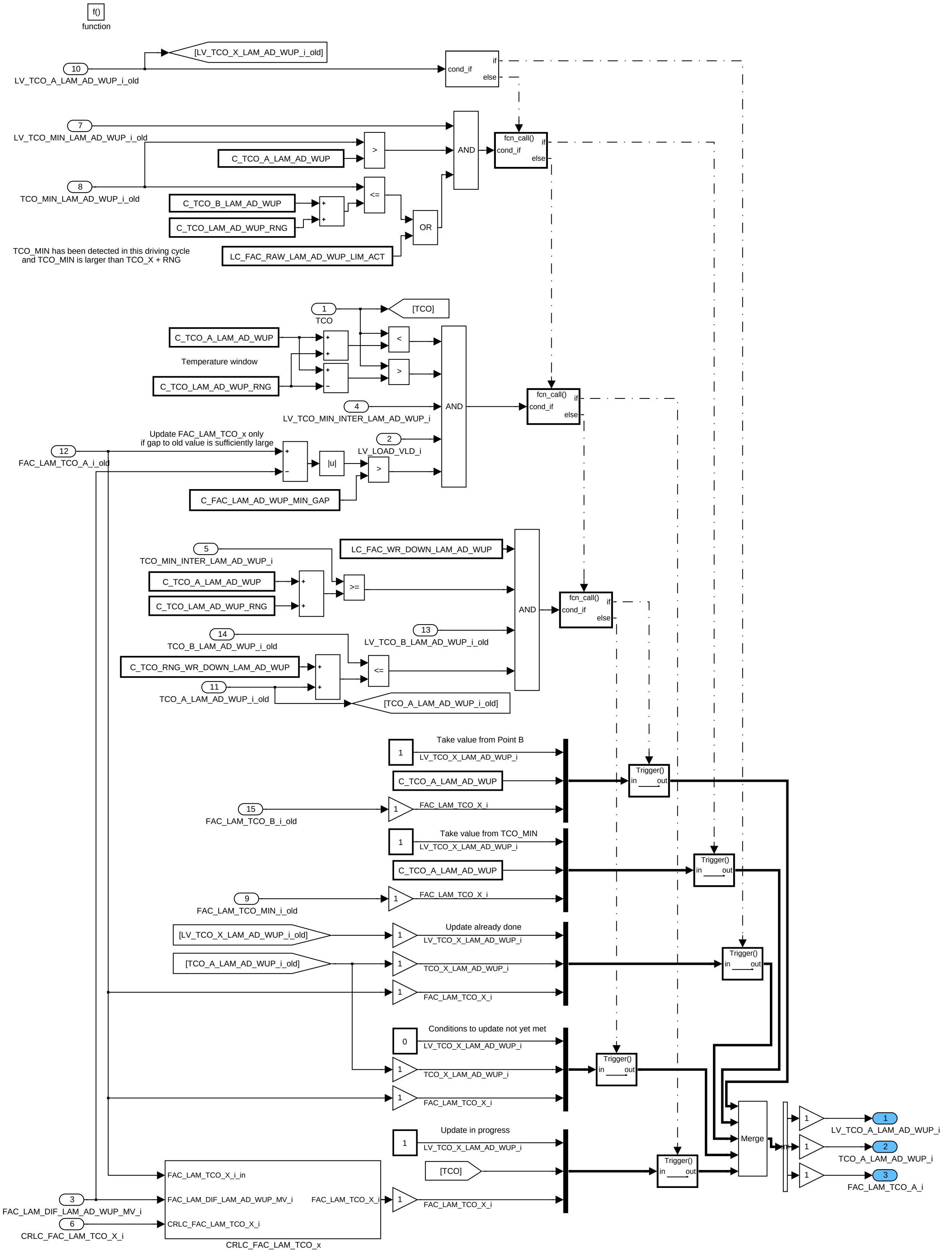


Figure 147 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_A/ CLC_NOT_CHG_FQ_DET

Determination of new adaptation value at point B

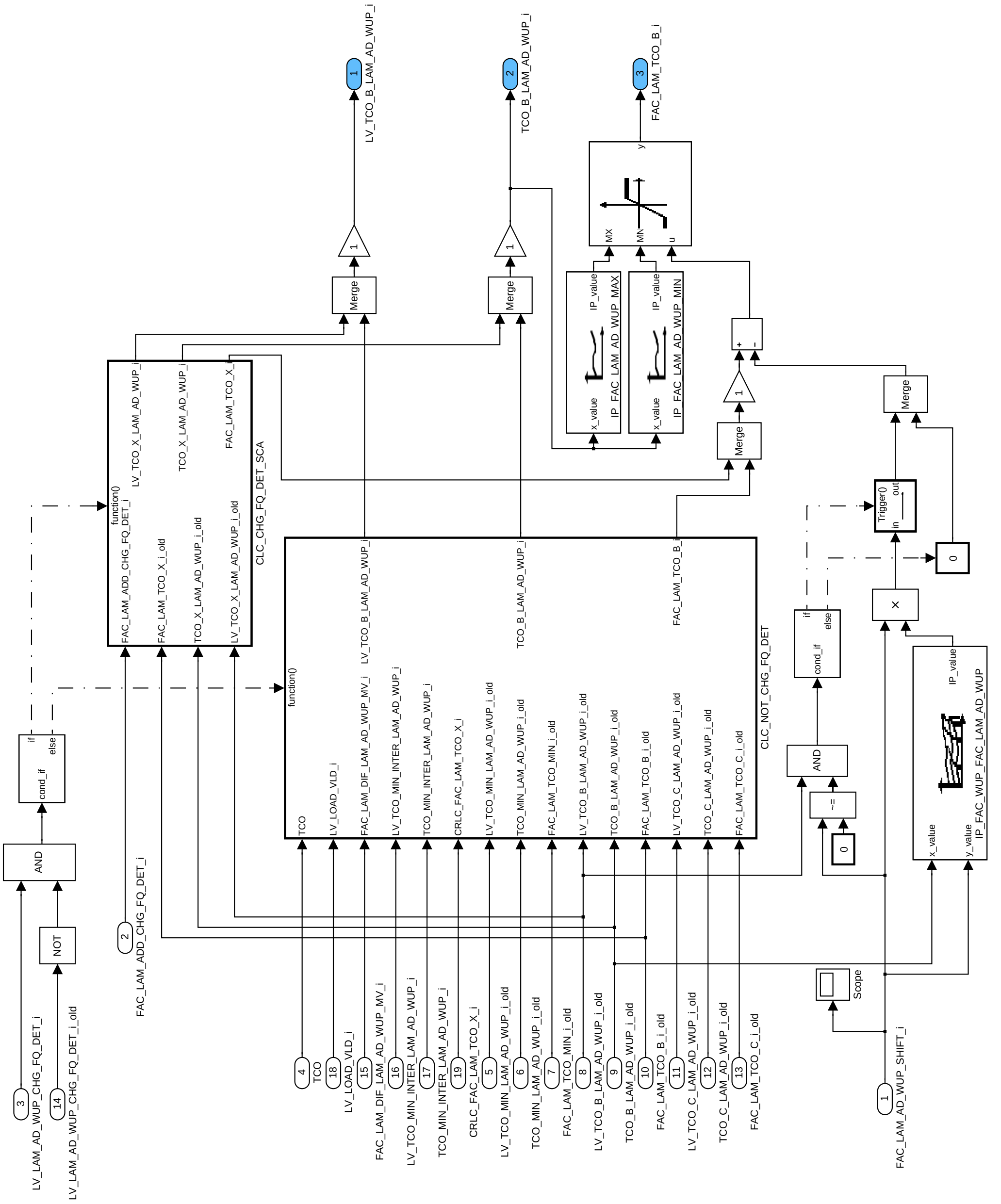


Figure 148 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_B

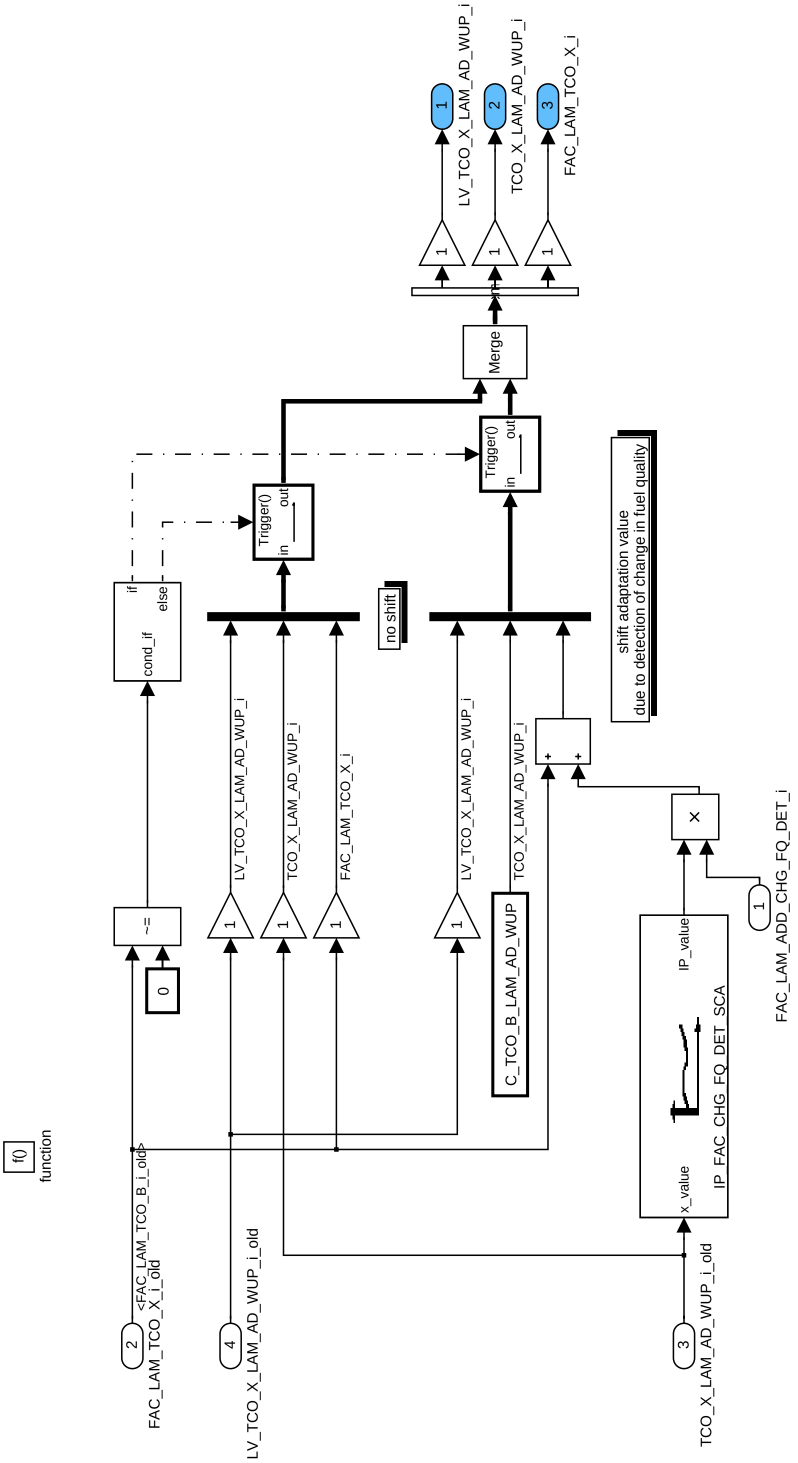


Figure 149 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_B/ CLC_CHG_FQ_DET_SCA

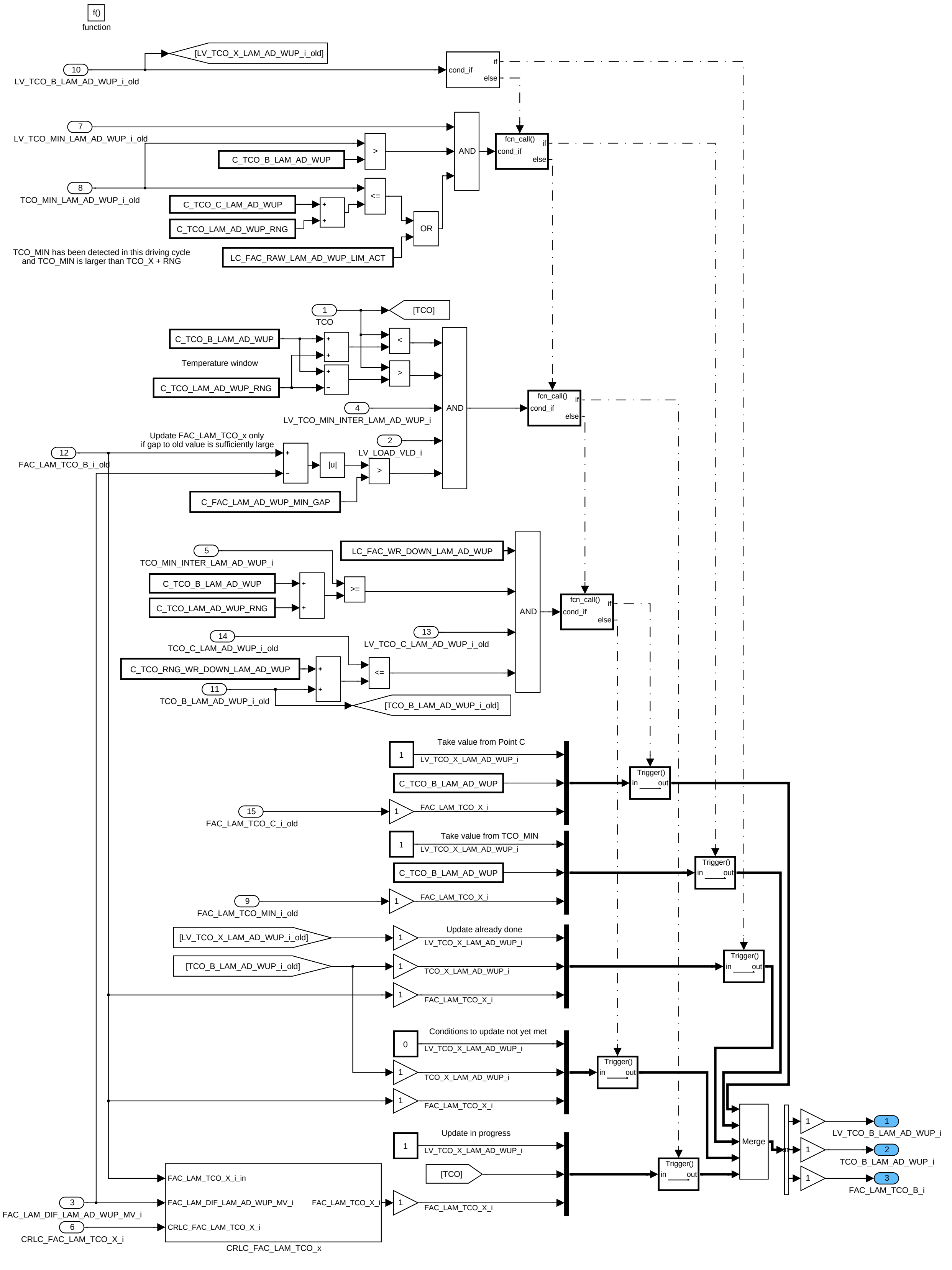


Figure 150 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_B/ CLC_NOT_CHG_FQ_DET

Determination of new adaptation value at point C

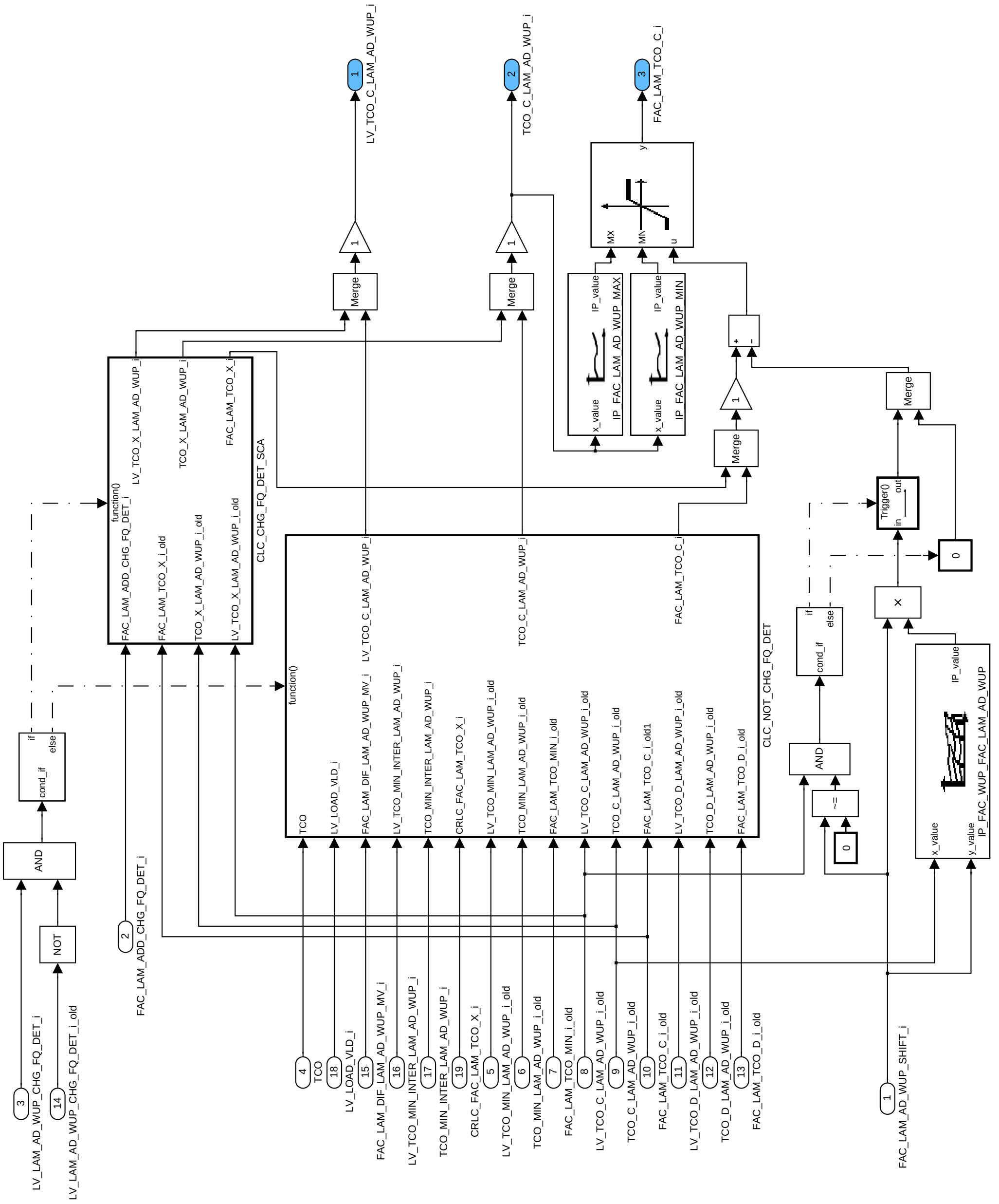


Figure 151 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_C

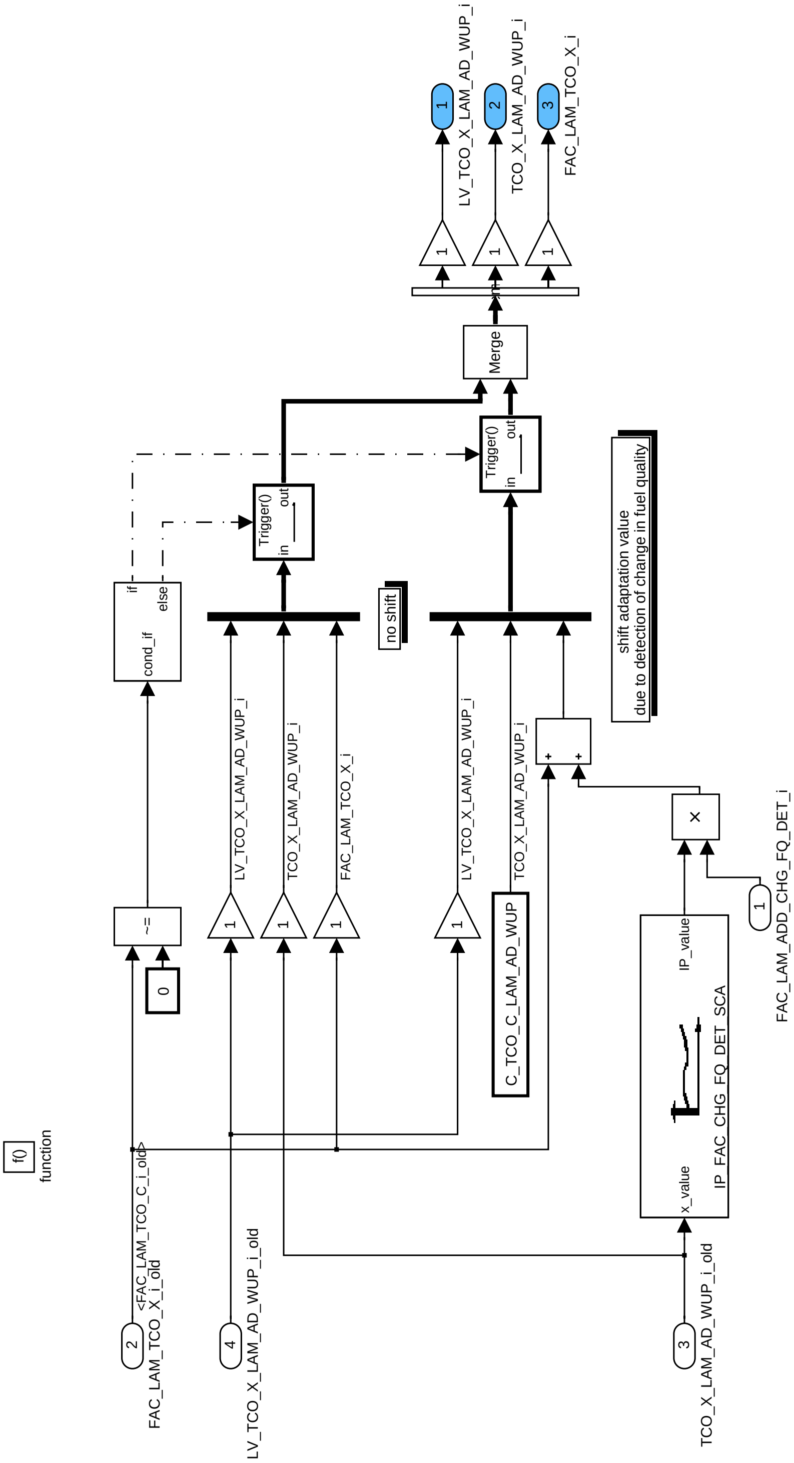


Figure 152 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_C/ CLC_CHG_FQ_DET_SCA

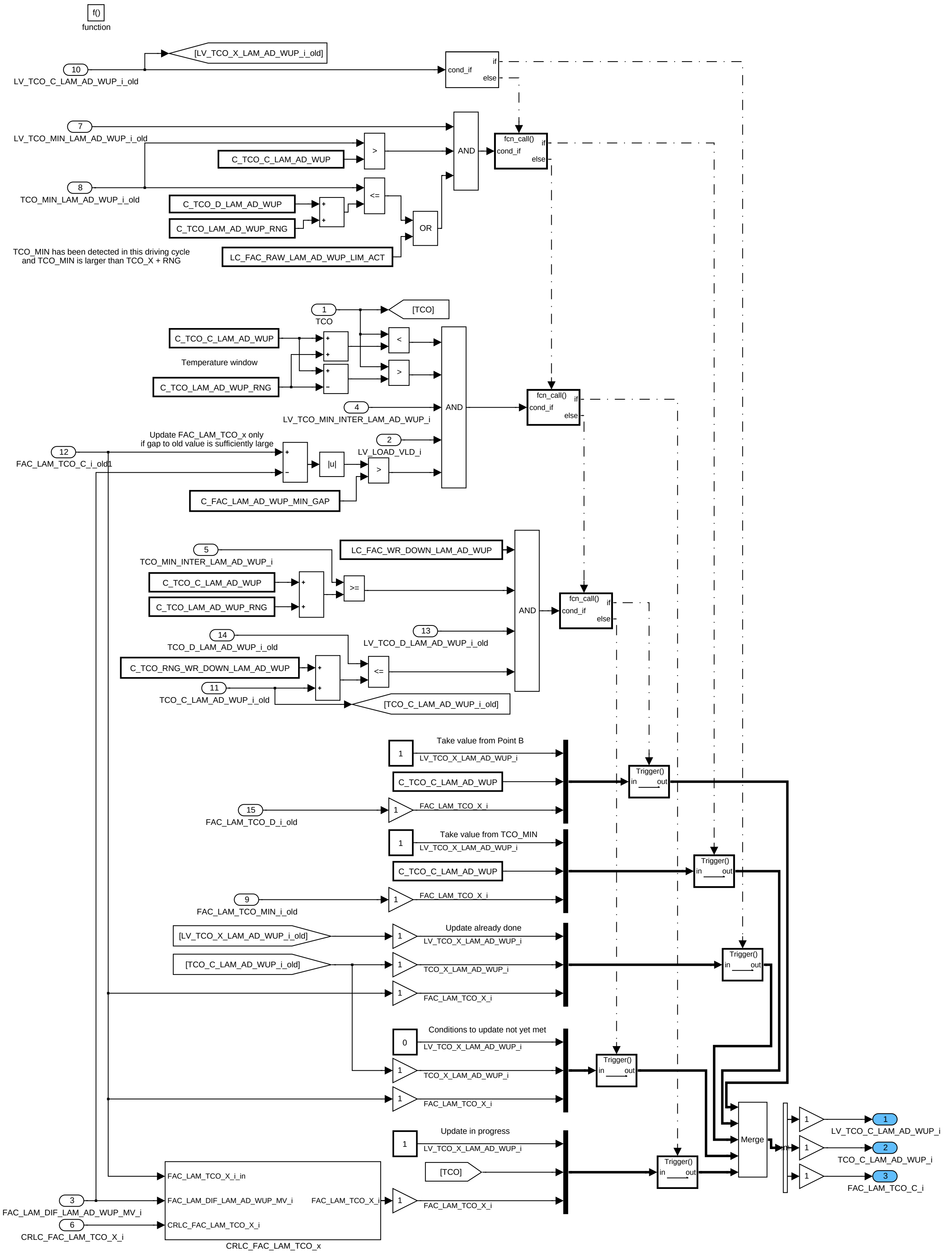


Figure 153 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_C/ CLC_NOT_CHG_FQ_DET

Determination of new adaptation value at point D

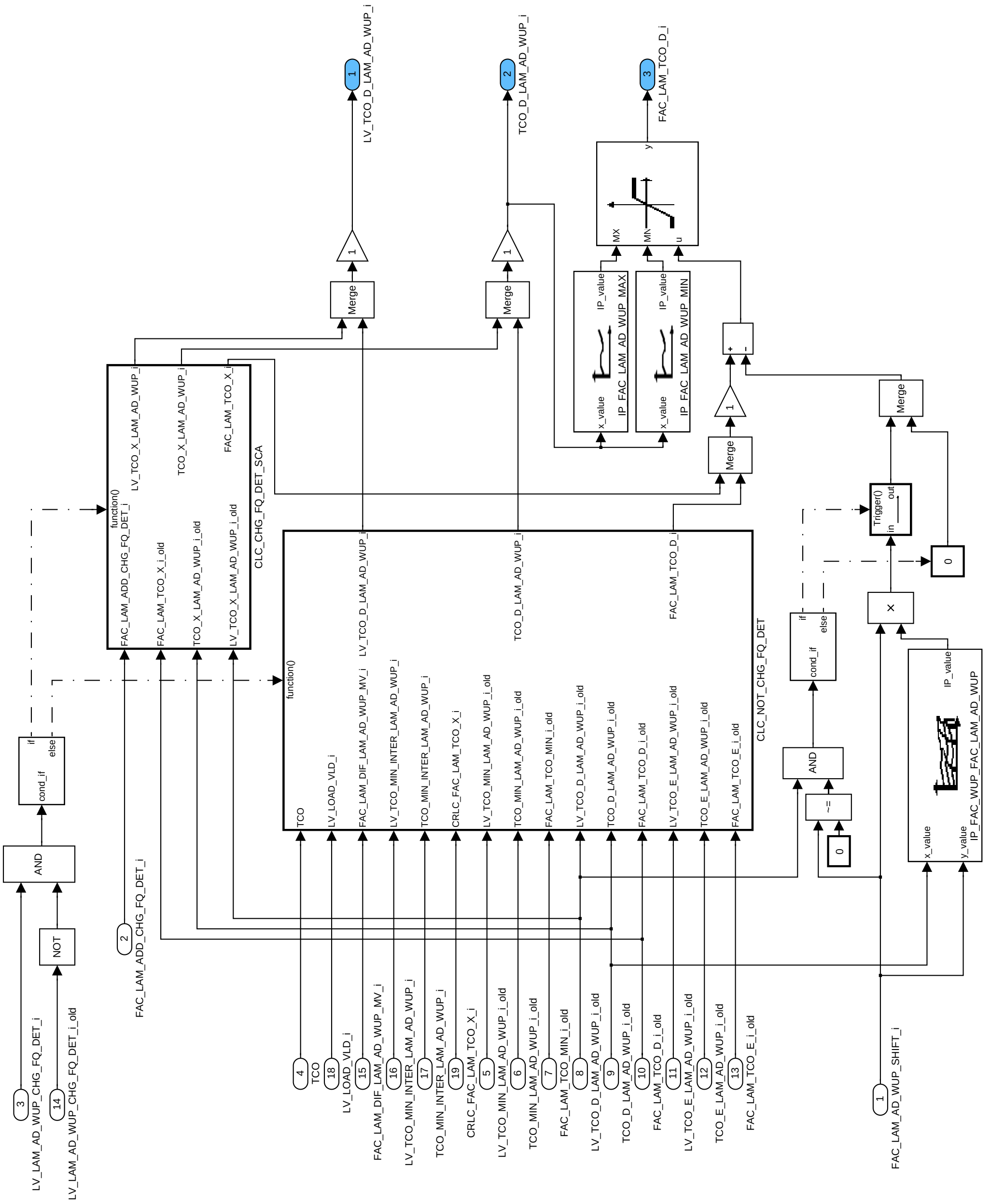


Figure 154 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_D

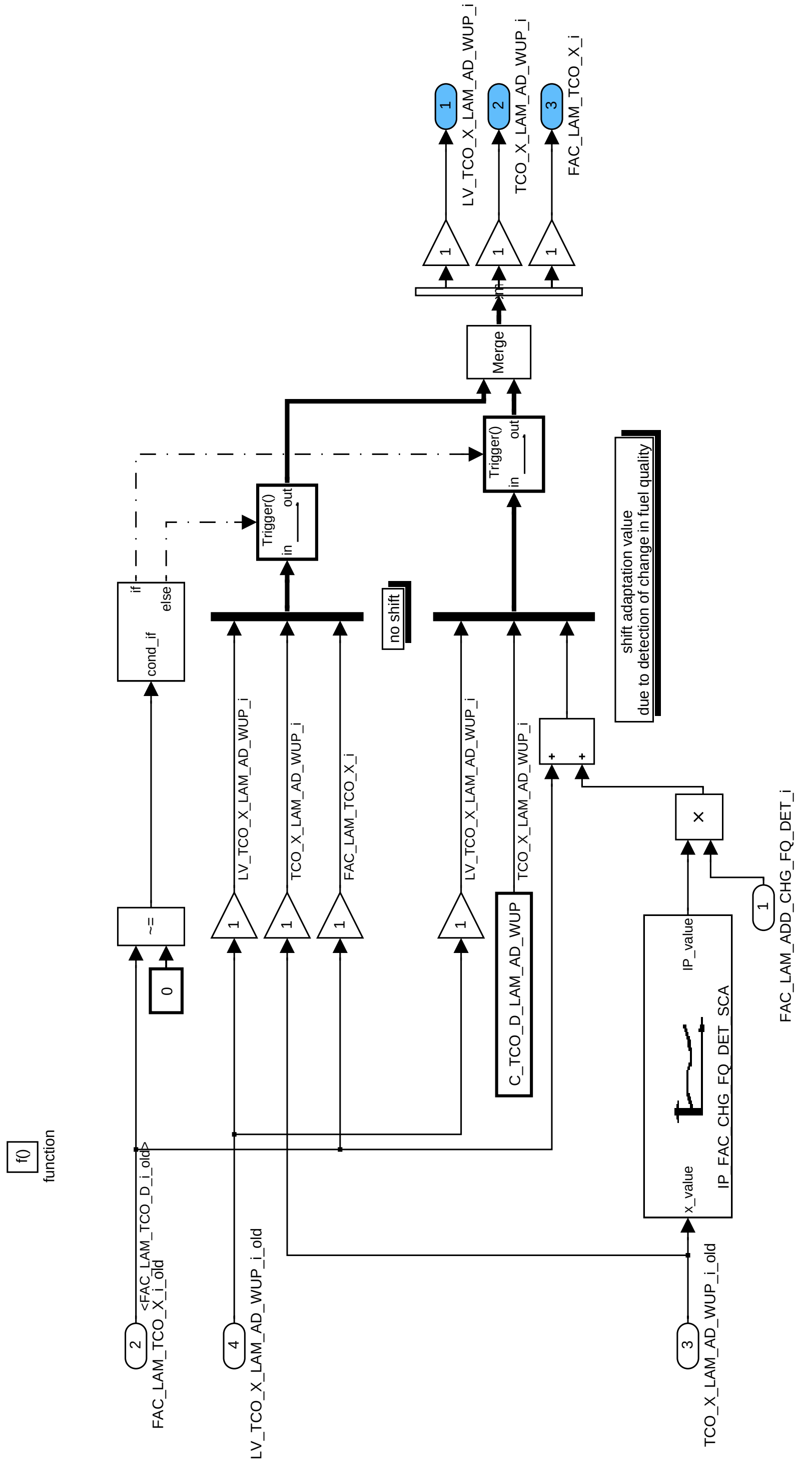


Figure 155 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_D/ CLC_CHG_FQ_DET_SCA

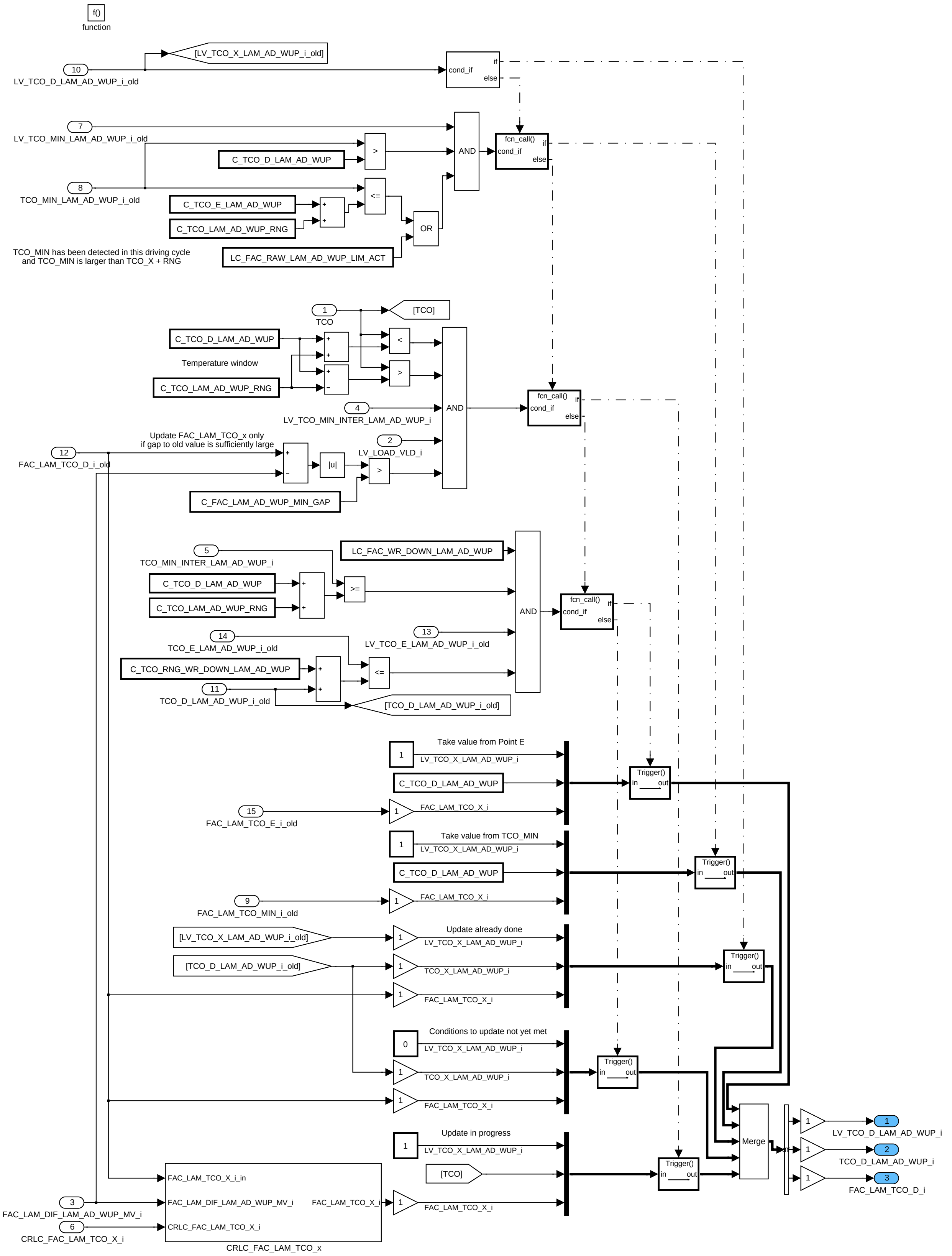


Figure 156 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_D/ CLC_NOT_CHG_FQ_DET

Determination of new adaptation value at point E

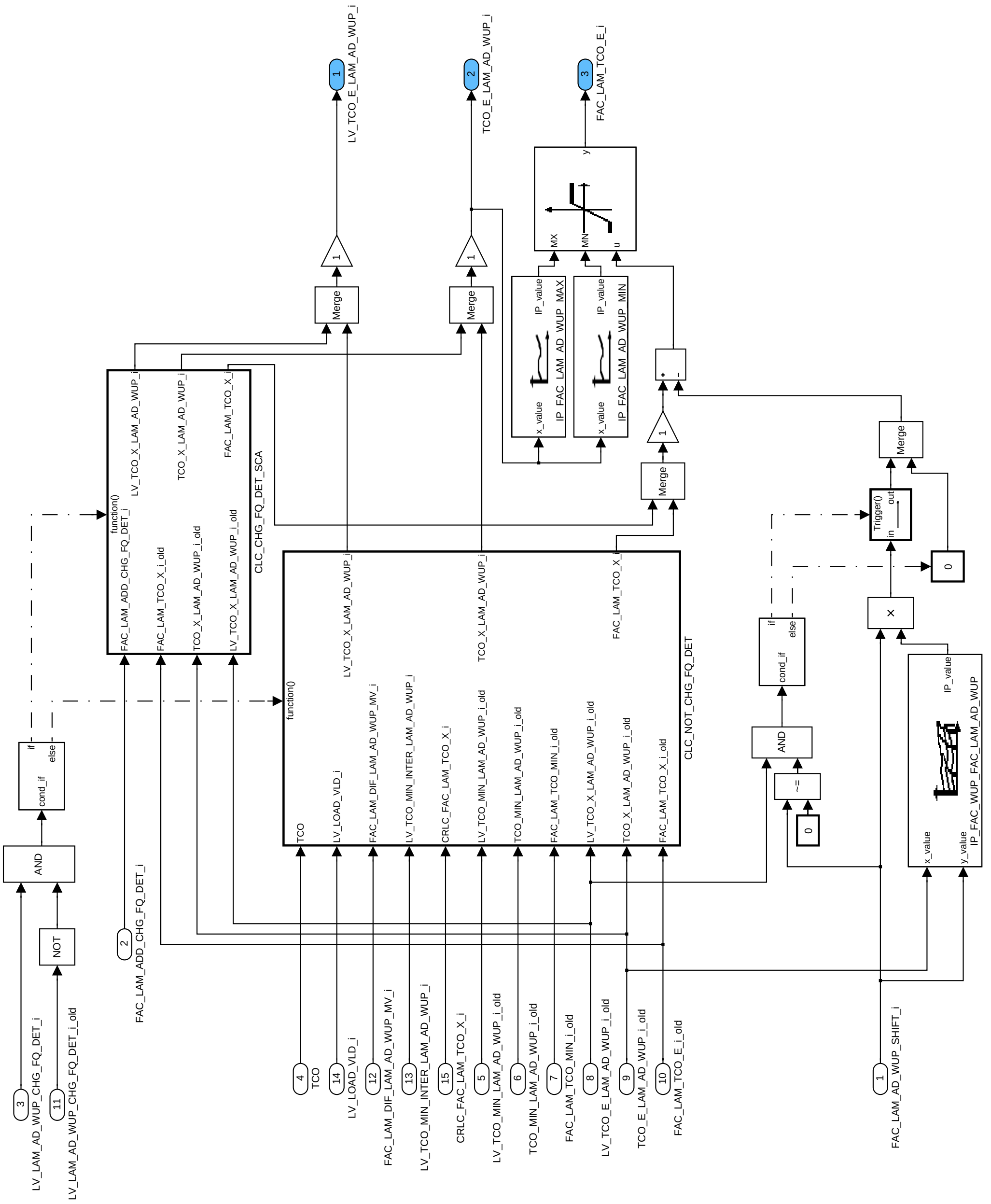


Figure 157 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_E

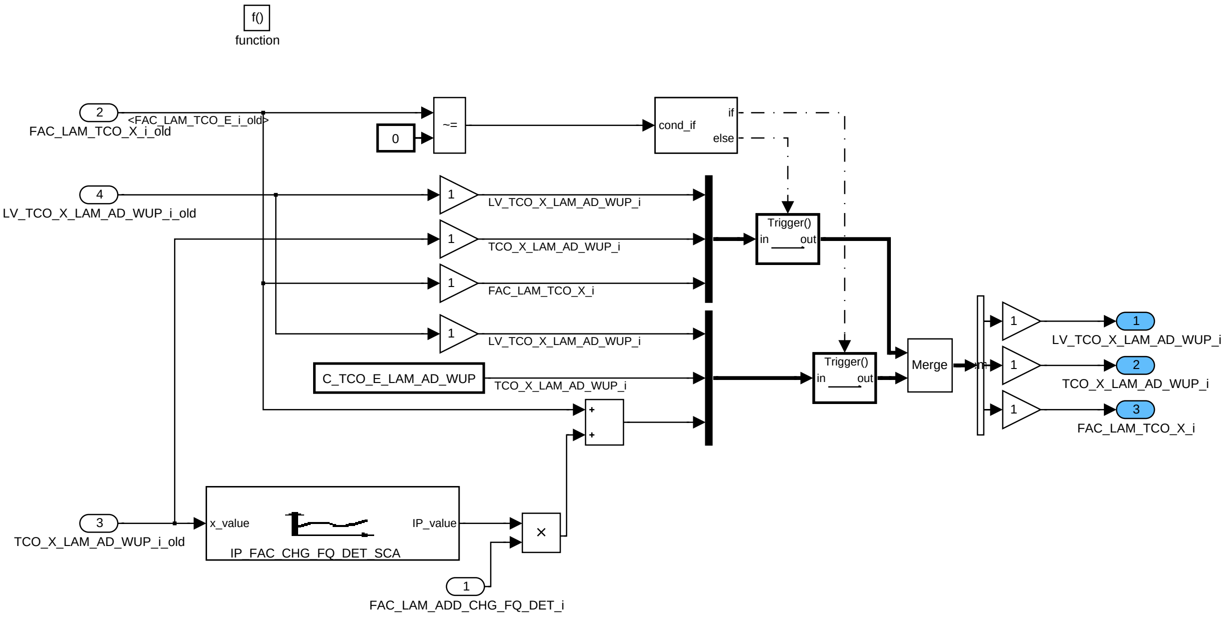


Figure 158 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_E/ CLC_CHG_FQ_DET_SCA

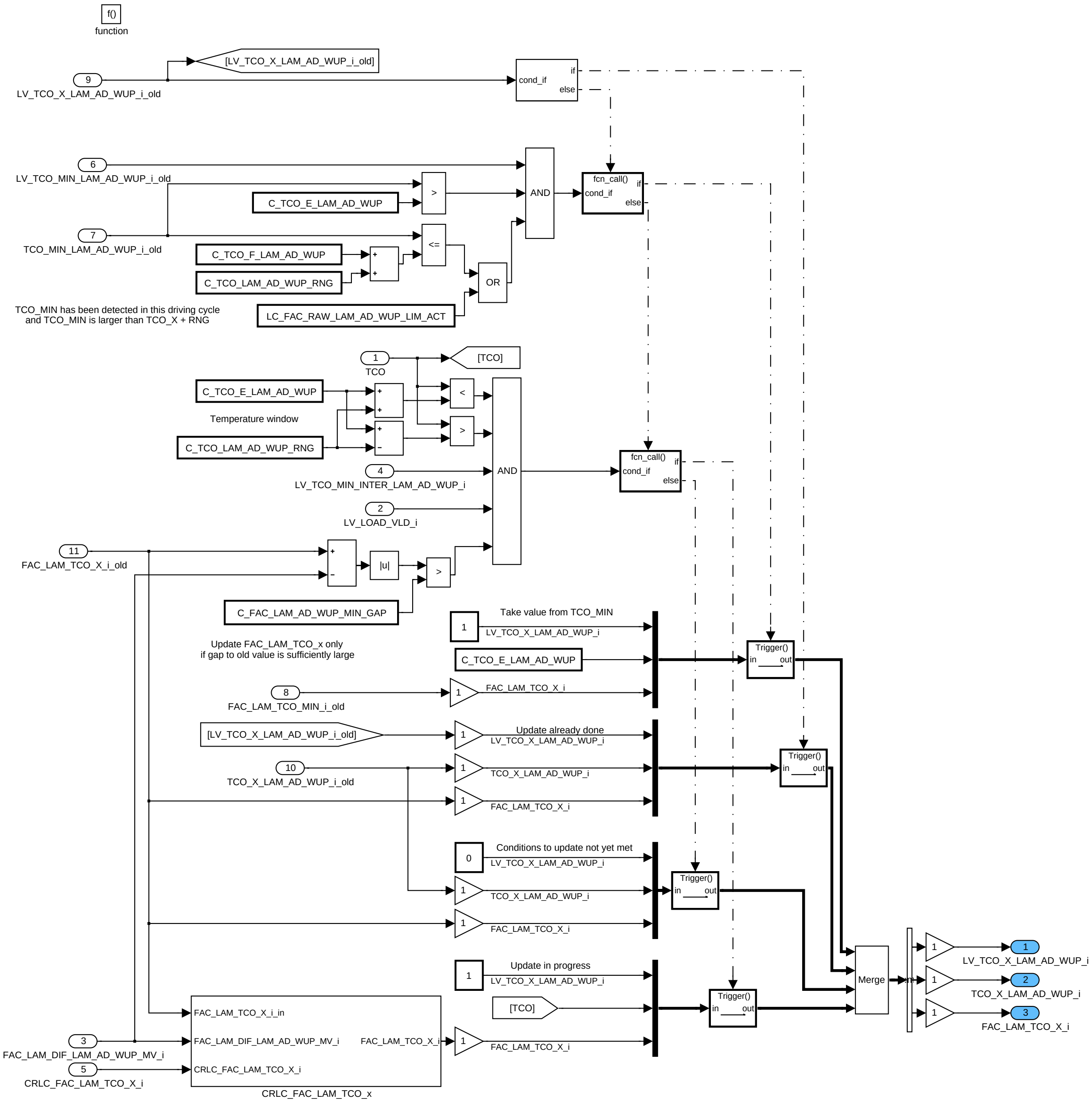


Figure 159 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_E/ CLC_NOT_CHG_FQ_DET

Set variables at point F

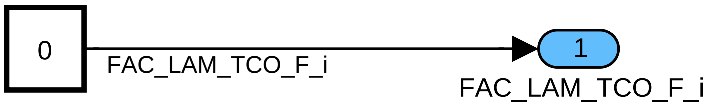


Figure 160 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ TCO_LAM_AD_WUP/ CLC_FAC_LAM_TCO_F

Parameters for slope equation

Here the parameters which are required to solve the slope equation are arranged.

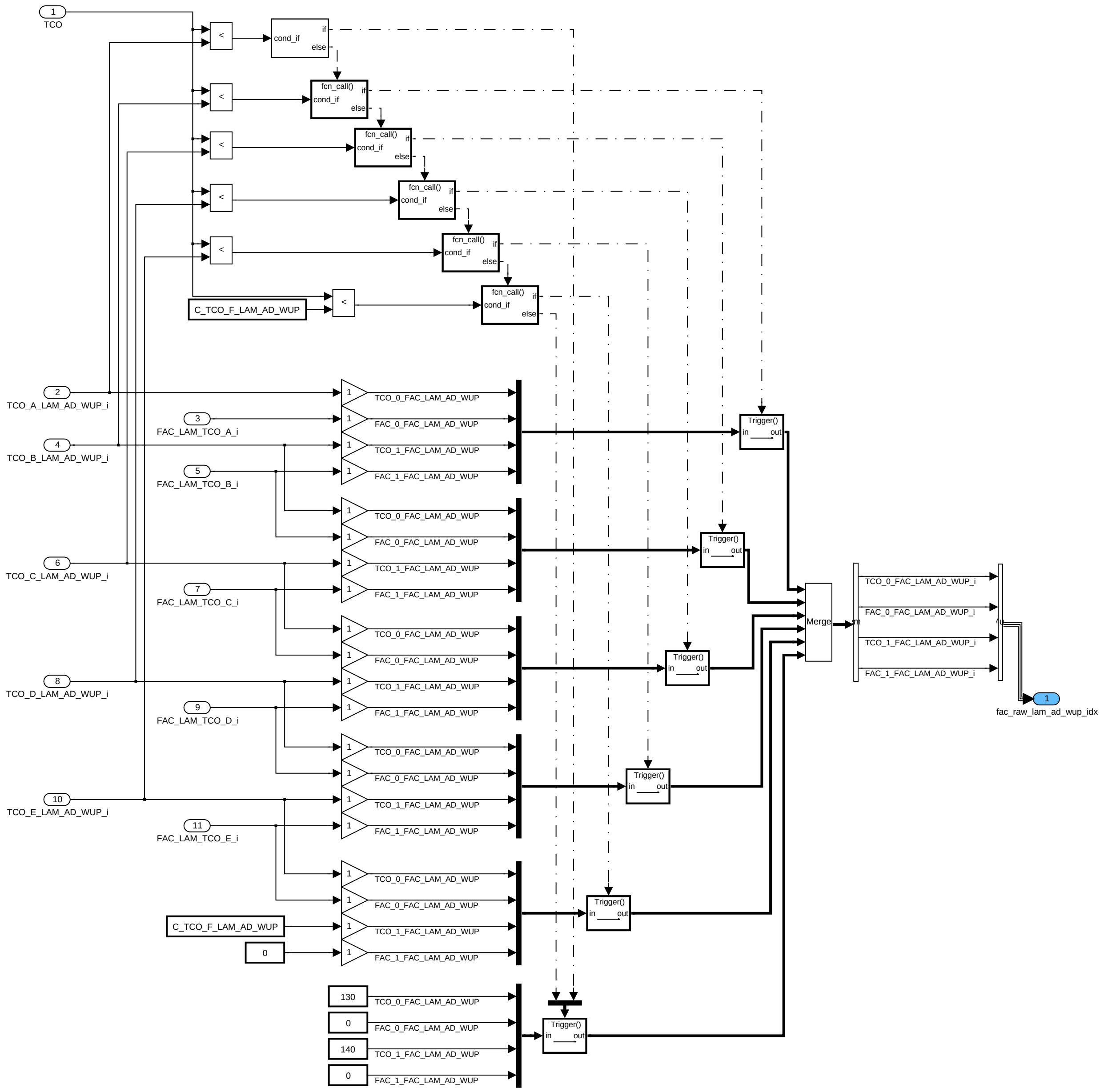


Figure 161 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ CLC_FAC_LAM_AD_WUP/ FAC_RAW_LAM_AD_WUP_IDX

Calculation of the Lambda adaptation factor in the warm-up phase

The respective adaptation factor in warm-up phase is derived from slope equation with TCO as an input. Five different cases for the calculation of FAC_RAW_LAM_AD_WUP must be distinguished:

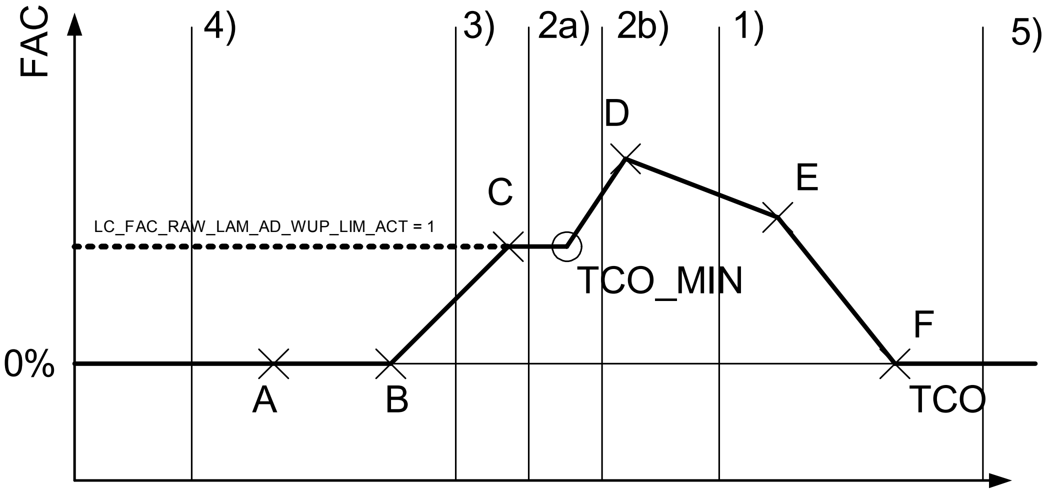
1.) The momentary TCO is between two interpolation points with TCO_MIN *smaller* than the lower interpolation point: Solving the slope equation with the two interpolation points and TCO as an input will yield FAC_RAW_LAM_AD_WUP.

2.) The momentary TCO is between two interpolation points with TCO_MIN *larger* than the lower interpolation point: Between the lower interpolation point and TCO_MIN, the output is fixed to FAC_LAM_TCO_MIN. Between TCO_MIN and the upper interpolation point, the slope equation made up from TCO_MIN and the upper interpolation point shall be used to calculate FAC_RAW_LAM_AD_WUP.

3.) TCO is smaller than TCO_MIN and TCO_MIN is larger than the next higher interpolation point. If LC_FAC_RAW_LAM_AD_WUP_LIM_ACT is set to one, the output value is fixed to FAC_LAM_TCO_MIN; otherwise it shall be obtained by solving the slope equation.

4.) TCO is smaller than temperature at point A:
If LC_FAC_RAW_LAM_AD_WUP_LIM_ACT is set to one, the output value is fixed to FAC_LAM_TCO_MIN, otherwise it is zero.

5.) TCO is larger than temperature of point F: the output is zero.



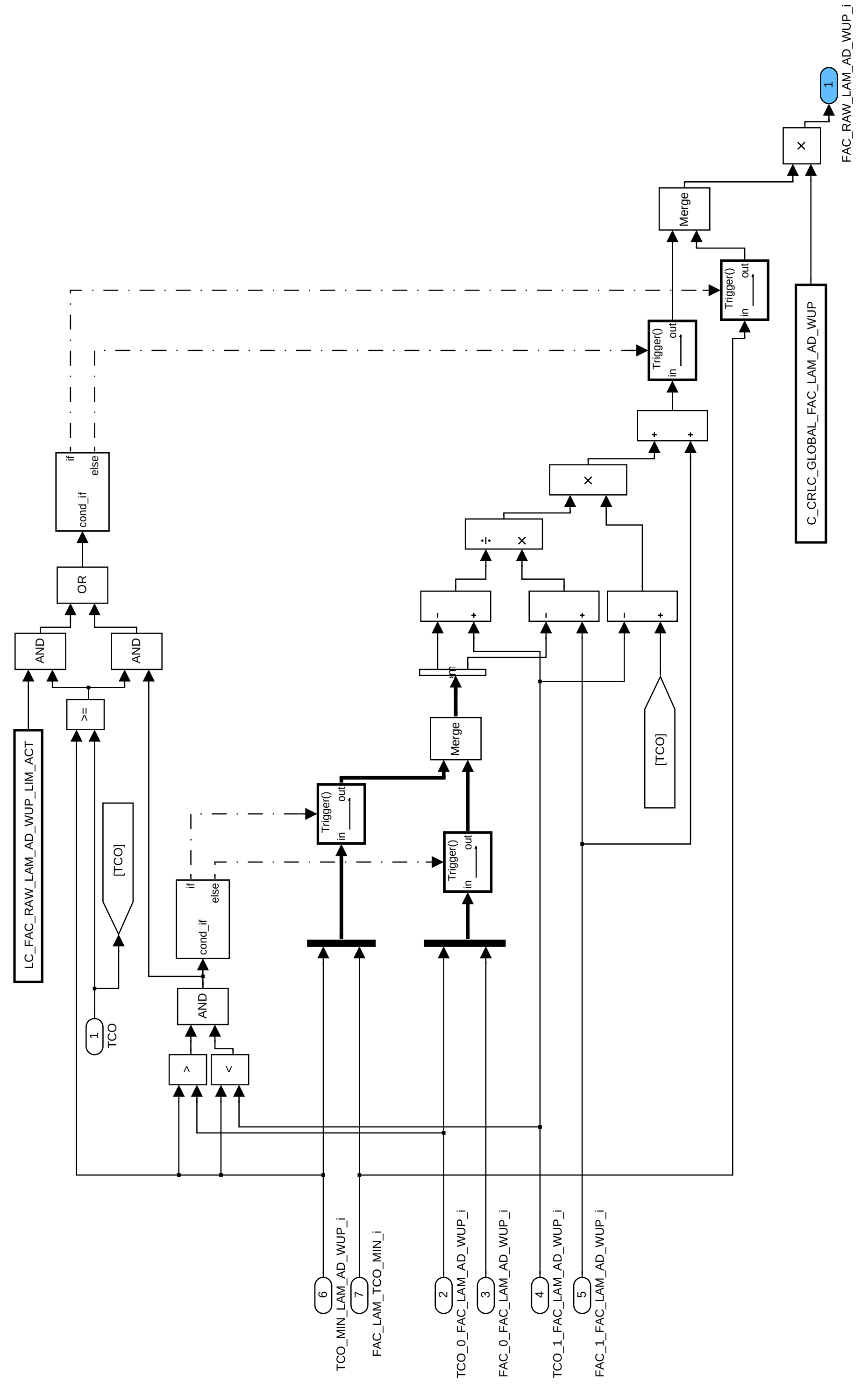


Figure 162 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ CLC_FAC_LAM_AD_WUP/ FAC_RAW_LAM_AD_WUP

Limitation of adaptation factor

Three different limitations of the Lambda adaptation factor in warm-up phase can be applied. If the conditions to go closed loop Lambda control are not yet fulfilled (STATE_LS == 0) C_FAC_LAM_AD_WUP_MAX/MIN_LSCL_OFF is applied. In the transient phase until TCO_MIN_INTER is detected C_FAC_LAM_AD_WUP_MAX/MIN_INTER is the relevant limitation. After detection of TCO_MIN_INTER the limitation are specified by IP_FAC_LAM_AD_WUP_MAX/MIN.

With C_FAC_LAM_AD_WUP_DELTA the shifts in the Lambda adaptation factor in warm-up phase can be smoothed by specifying the maximum step value. It has been observed that for large changes in the adaptation value that Lambda controller output is not reduced as fast as the adaptation value increases (due to transient effects). This causes unwanted deviation of the Lambda value, which can be avoided by specifying a smooth transition between different values of the adaptation factor.

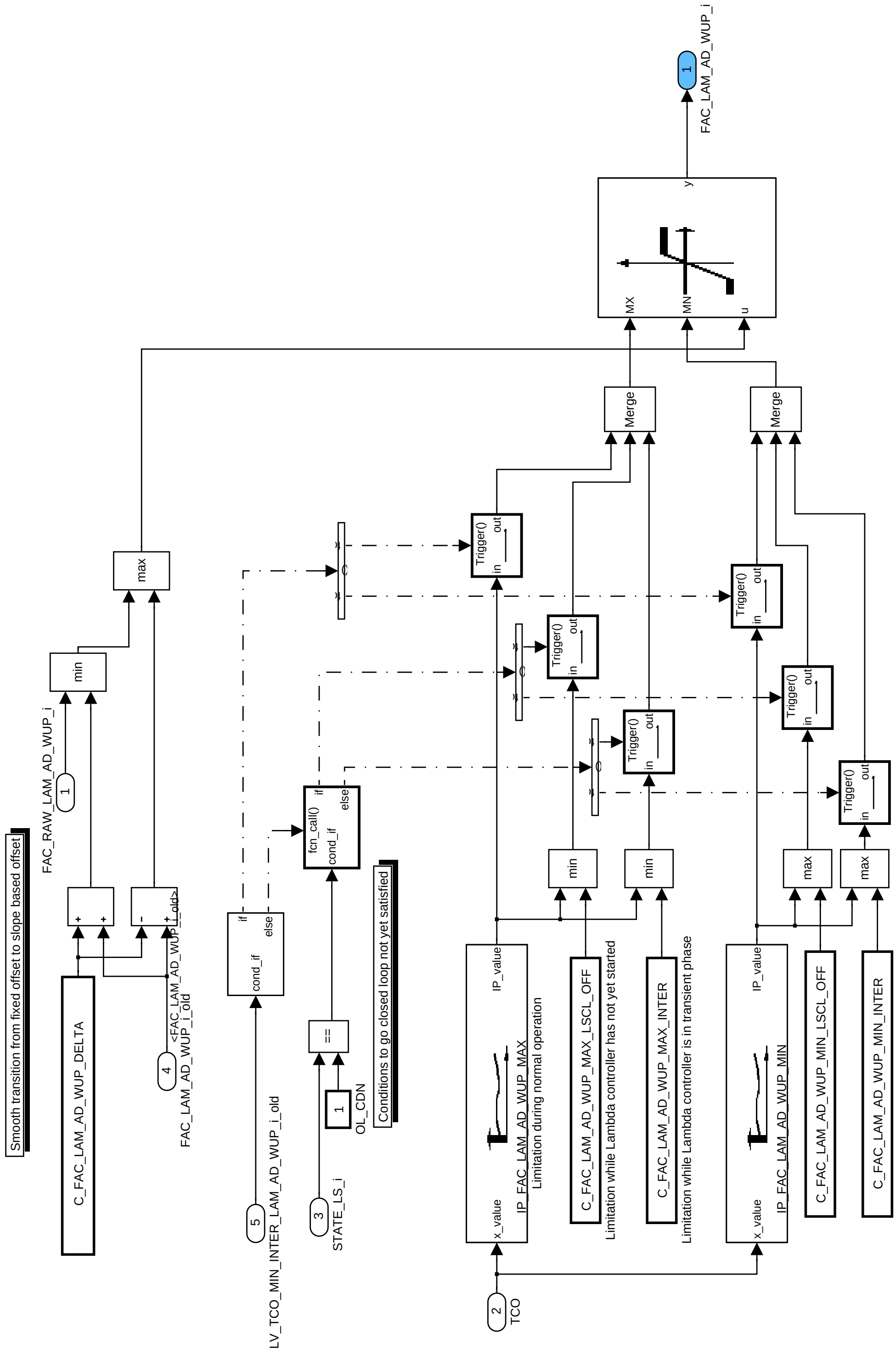


Figure 163 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ CLC_FAC_LAM_AD_WUP/ FAC_LAM_AD_WUP_LIM

Shift of the Lambda controller

A shift of the Lambda controller output is required if the value of FAC_LAM_AD_WUP[i] has changed since the last calculation recurrence in order to avoid large Lambda controller excursions.

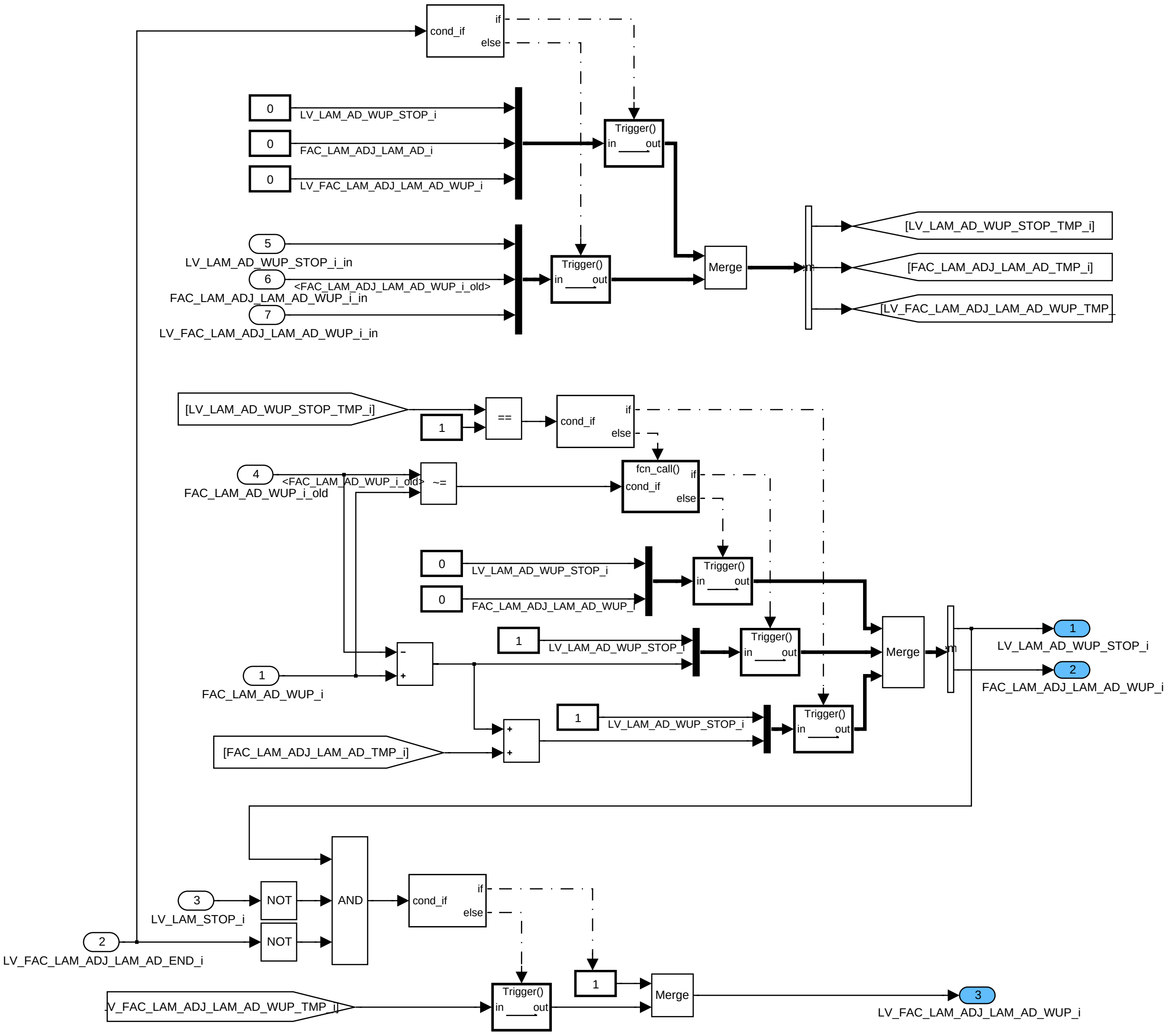


Figure 164 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ CLC_FAC_LAM_AD_WUP/ CLC_FAC_LAM_ADJ_LAM_AD_WUP

<u>Rewriting of adaptation correction in warm-up phase with non volatile stored values</u>

In case of Lambda adaptation relevant errors (see application incidences) the values of the data to be stored in the non-volatile memory must be rewritten with the values stored at the end of the last driving cycle.

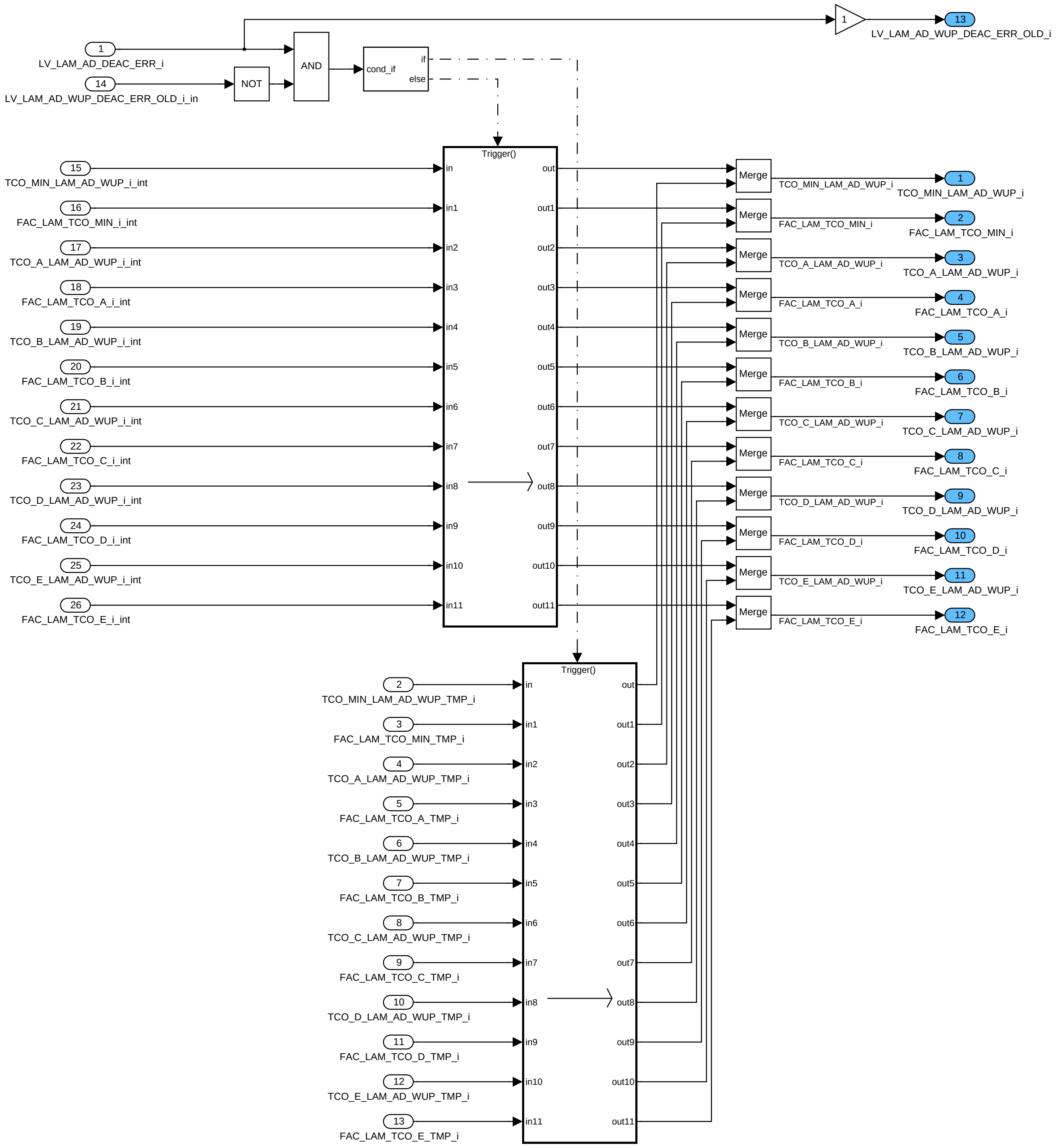


Figure 165 LACO_ISPCLADAW0/ operate/ FAC_LAM_AD_WUP_CBK_MNG/ FAC_LAM_AD_WUP_CBK_SPC/ ERR_LAM_AD_WUP

46.13.1.2 Initialization and Non Volatile Memory Data Handling

Non-Volatile Memory Data

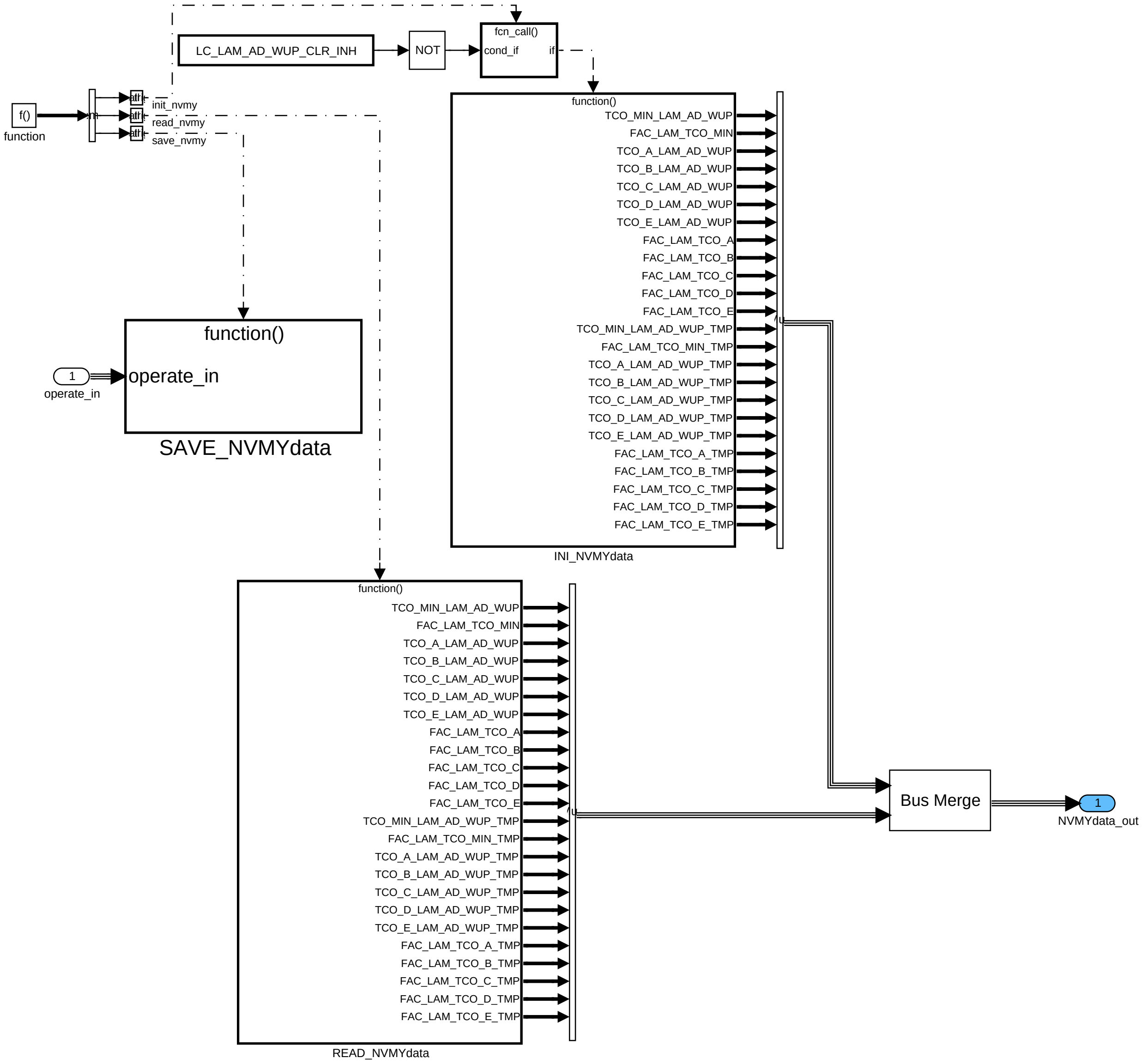


Figure 166 LACO_ISPCLADAW0/ INIT/ NVMYdata

Initialization of NVMY Data

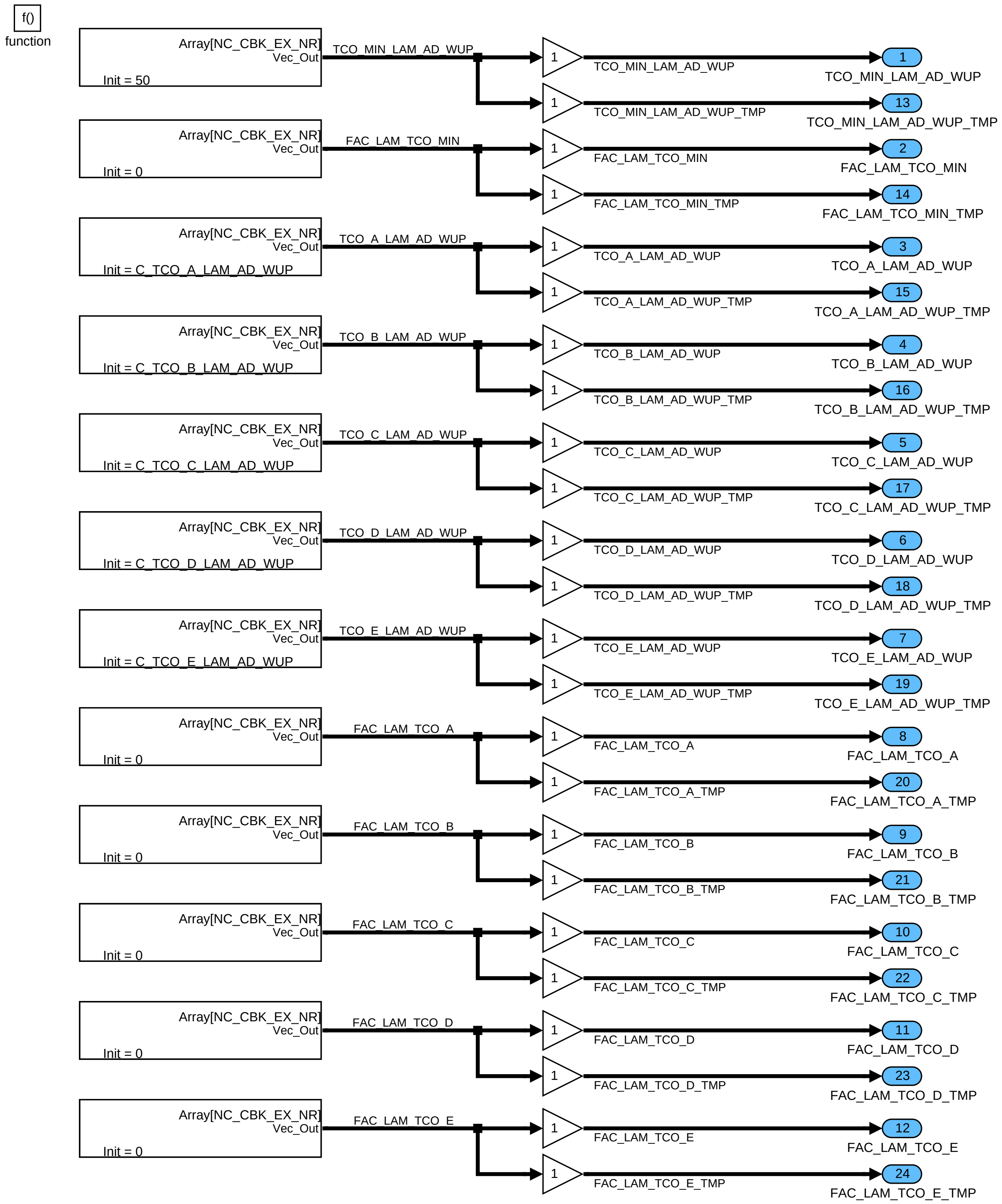


Figure 167 LACO_ISPCLADAW0/ INIT/ NVMYdata/ INI_NVMYdata

Read NVMY Data

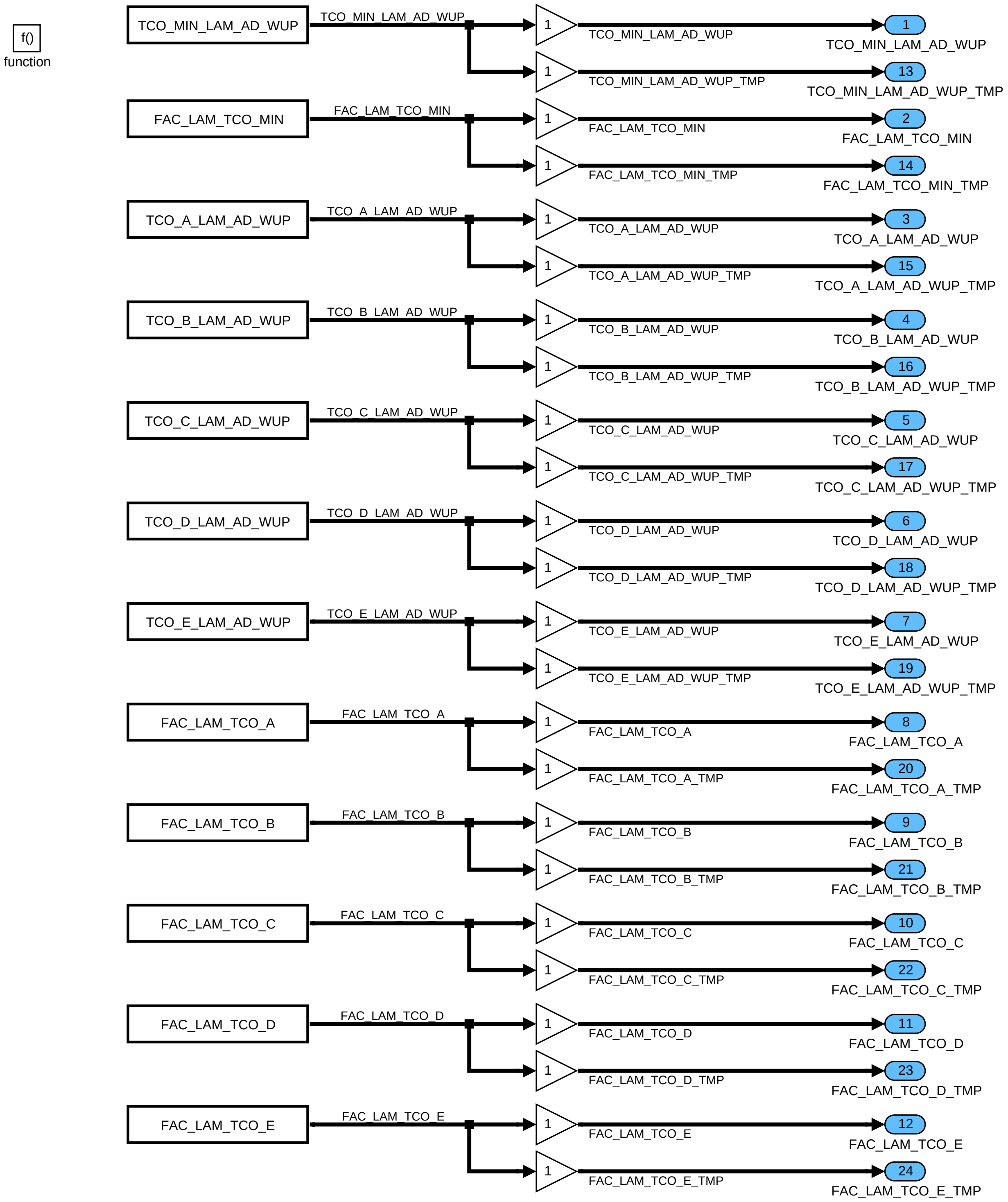
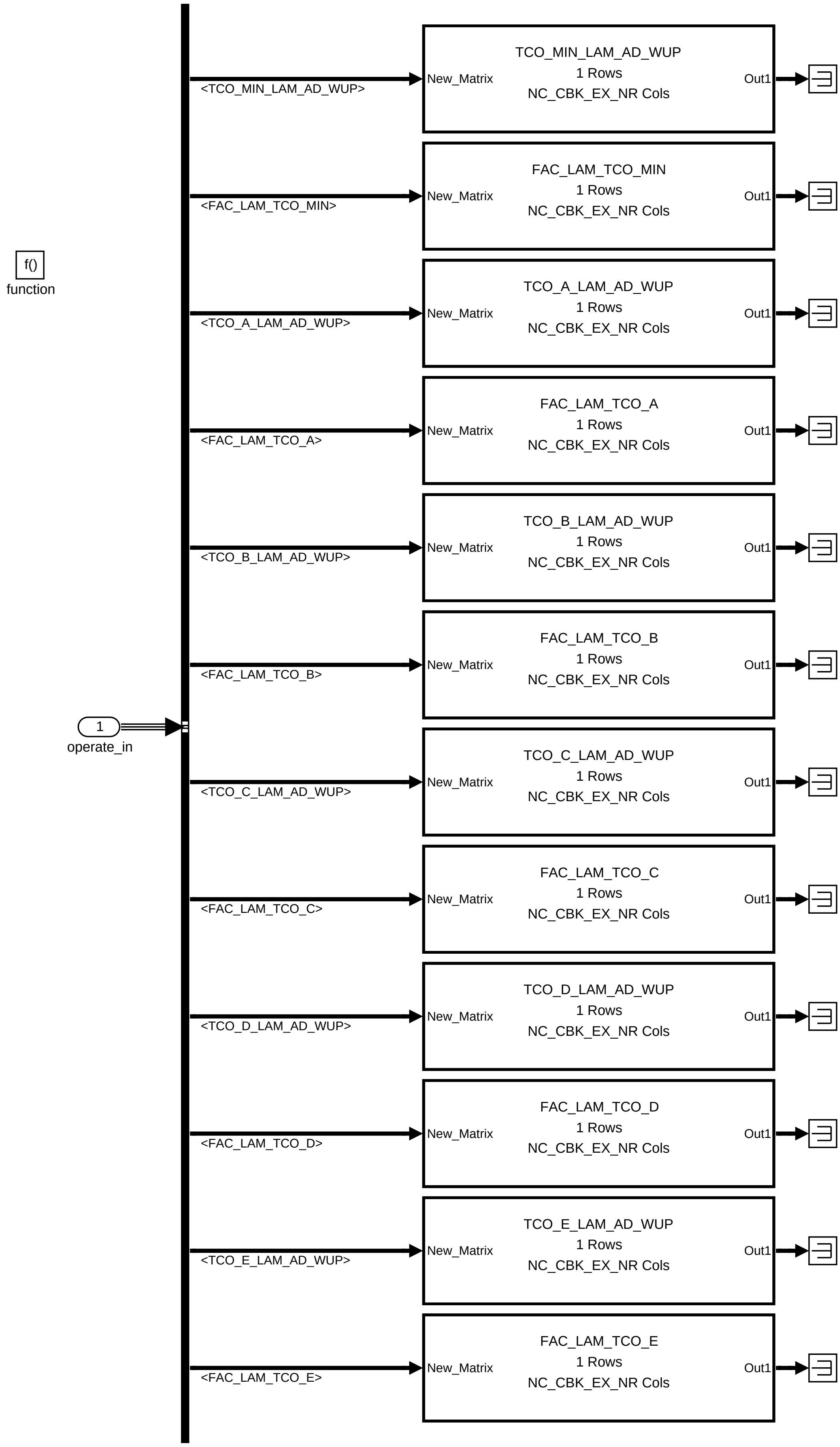


Figure 168 LACO_ISPCLADAW0/ INIT/ NVMYdata/ READ_NVMYdata

Store NVMY



Chapter
Lambda control

Baseline
4DC3940S

Include File
30707Z03.00A

Date
Department
Sign

Designed by
2008-07-01

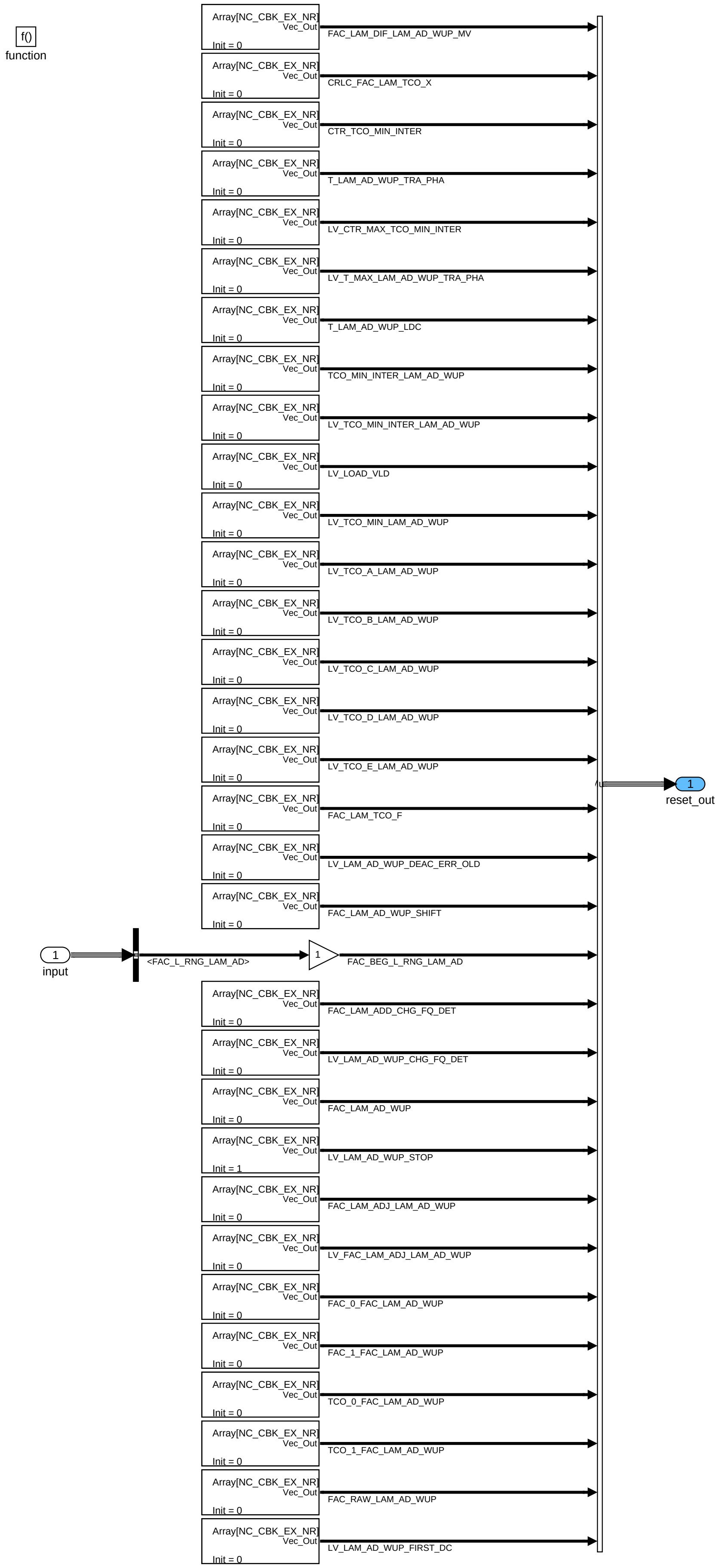
Released by
2008-07-01

Designation
Engine Management System MSD80 6 Cyl

Document Key
E002-190.49.02 SPE 000 48.0

Figure 169 LACO_ISPCLADAW0/ INIT/ NVMYdata/ SAVE_NVMYdata

ECU Reset



Chapter	Baseline	Include File
Lambda control	4DC3940S	30707Z03.00A

	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		

Designation
Engine Management System MSD80 6 Cyl

Figure 170 LACO_ISPCLADAW0/ INIT/ reset

46.14 Mean value calculation for limited dynamics

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_LAM_MV_DELTA_LDC[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.9984741	0.00152588	%
difference between FAC_LAM_MV[i] and moving mean value of FAC_LAM_MV[i] for limited dynamics detection					
MAF_DELTA_LDC	O/V	8000...7FFFH	-694.510597 ... 694.489403	0.02119478	mg/stk
difference between MAF and moving mean value of MAF for limited dynamics detection					
N_DELTA_LDC	O/V	E020...1FE0H	-8.16E+3 ... 8.16E+3	1	rpm
difference between N and moving mean value of N for limited dynamics detection					
FAC_TQ_REQ_DELTA_LDC	O/V	8000...7FFFH	-1...0.99996948	3.05176E-5	-
difference between torque scaling factor and moving mean value thereof for limited dynamics calculation					
MAF_MMV_LDC	V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
moving mean value of MAF for limited dynamics detection					
N_MMV_LDC	V	0...1FE0H	0...8.16E+3	1	rpm
moving mean value of N for limited dynamics detection					
FAC_TQ_REQ_MMV_LDC	V	0...FFFFH	0...1.99996948	3.05176E-5	-
moving mean value of torque scaling factor for limited dynamics detection					

Input data:

FAC_LAM_MV[NC_CBK_EX_NR]	FAC_LAM_MV_MMV_LDC[NC_CBK_EX_NR]	LV_ST_END	MAF
N	NC_CBK_EX_NR	FAC_TQ_REQ	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_FAC_TQ_REQ_MMV_LDC	1	0...FFFFH	0...0.99998474	1.52588E-5	-
correlation constant for moving mean value calculation of torque scaling factor for limited dynamics					
C_CRLC_MAF_MMV_LDC	1	0...FFFFH	0...0.99998474	1.52588E-5	-
correlation constant for moving mean value calculation of MAF for limited dynamics					
C_CRLC_N_MMV_LDC	1	0...FFFFH	0...0.99998474	1.52588E-5	-
correlation constant for moving mean value calculation of N for limited dynamics					

46.14.1 LACO_ISPCLMVLDC0

This function calculates only differences between base values and their moving means values. These differences or deltas are to be used within the function that needs limited dynamic conditions. The evaluation of the limited dynamic bit is part of the function that need this flag. So the mean value calculation is concentrated in this function.

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension "_i" is used in the model instead of "[i]" as found in the textual description.

Application Condition

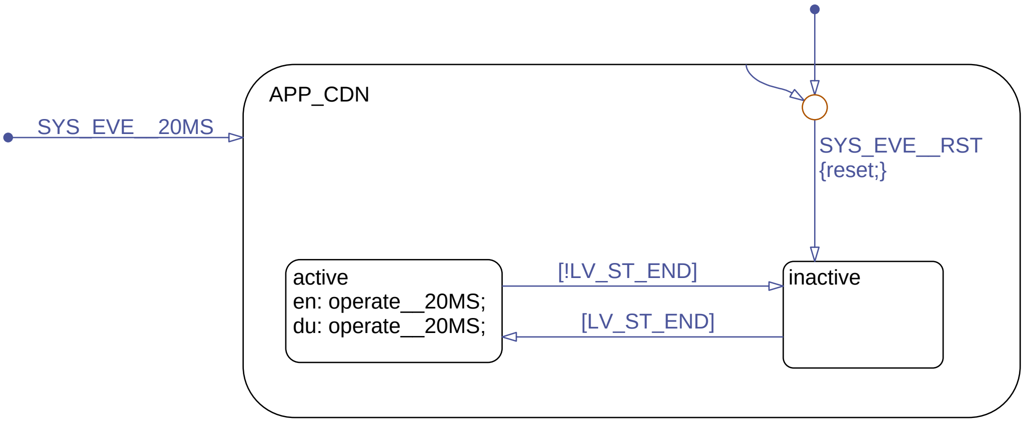
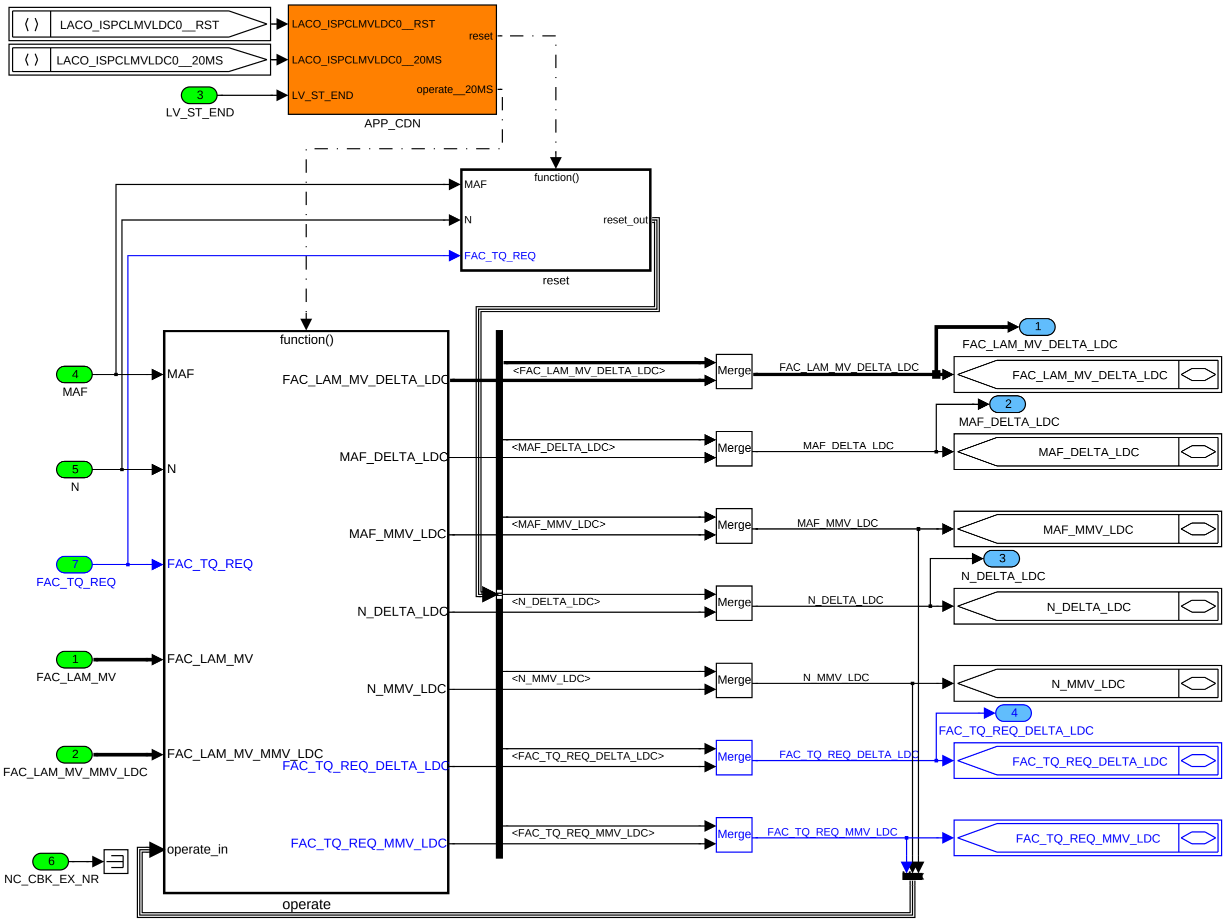


Figure 171 LACO_ISPCLMVLDC0/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 20−May−2005

Figure 172 LACO_ISPCLMVLDC0

46.14.1.1 SUBFUNCTION: reset

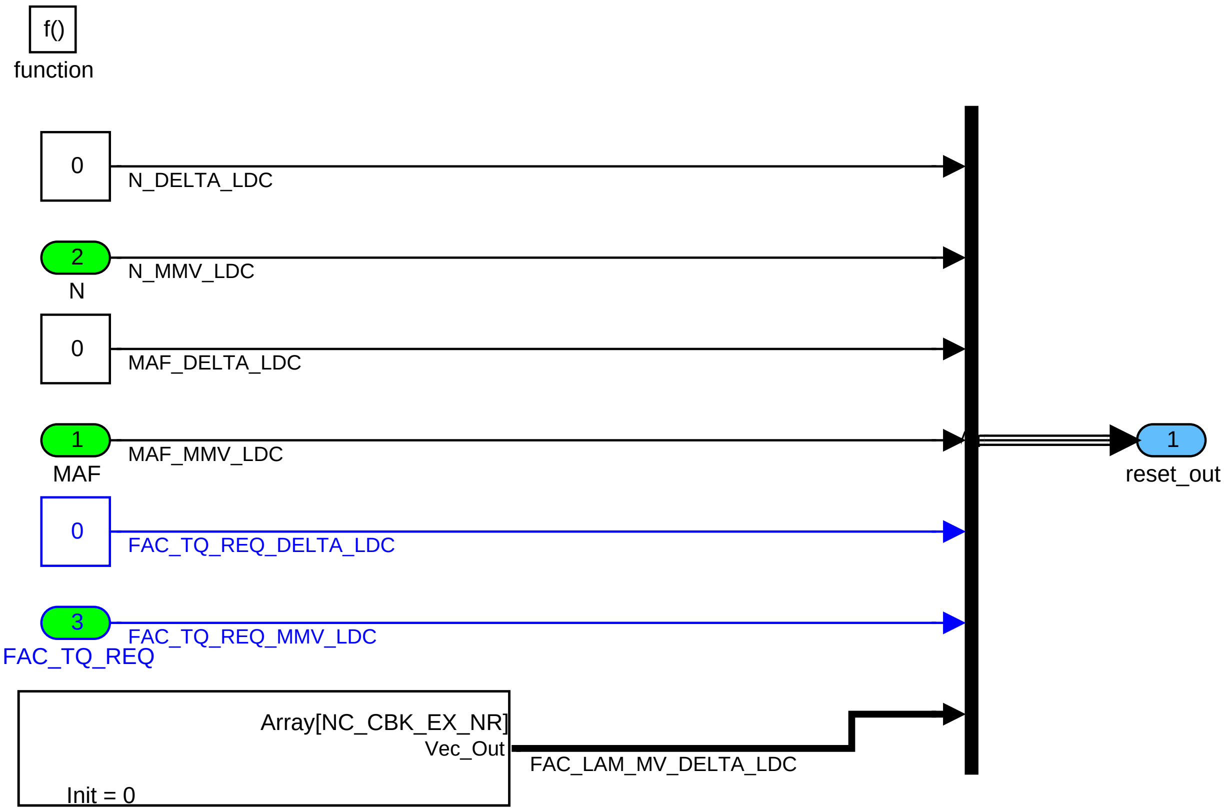


Figure 173 LACO_ISPCLMVLDC0/ reset

46.14.1.2 operate

This function calculates for the detection of the limited dynamics the moving mean values of engine speed (N and N_MMV_LDC), air mass flow (MAF and MAF_MMV_LDC) and degree of activation of the accelerator pedal (PV and PV_AV_MMV_LDC). Furthermore the difference between these input values and the moving mean values is calculated.

For the lambda control output only the difference is evaluated because the moving mean value calculation is located in the lambda control module itself because of easier realisation of shift operations of the moving mean value (FAC_LAM_MV_MMV_LDC[i]).

Remark: The difference calculation of lambda control output is a vector operation.

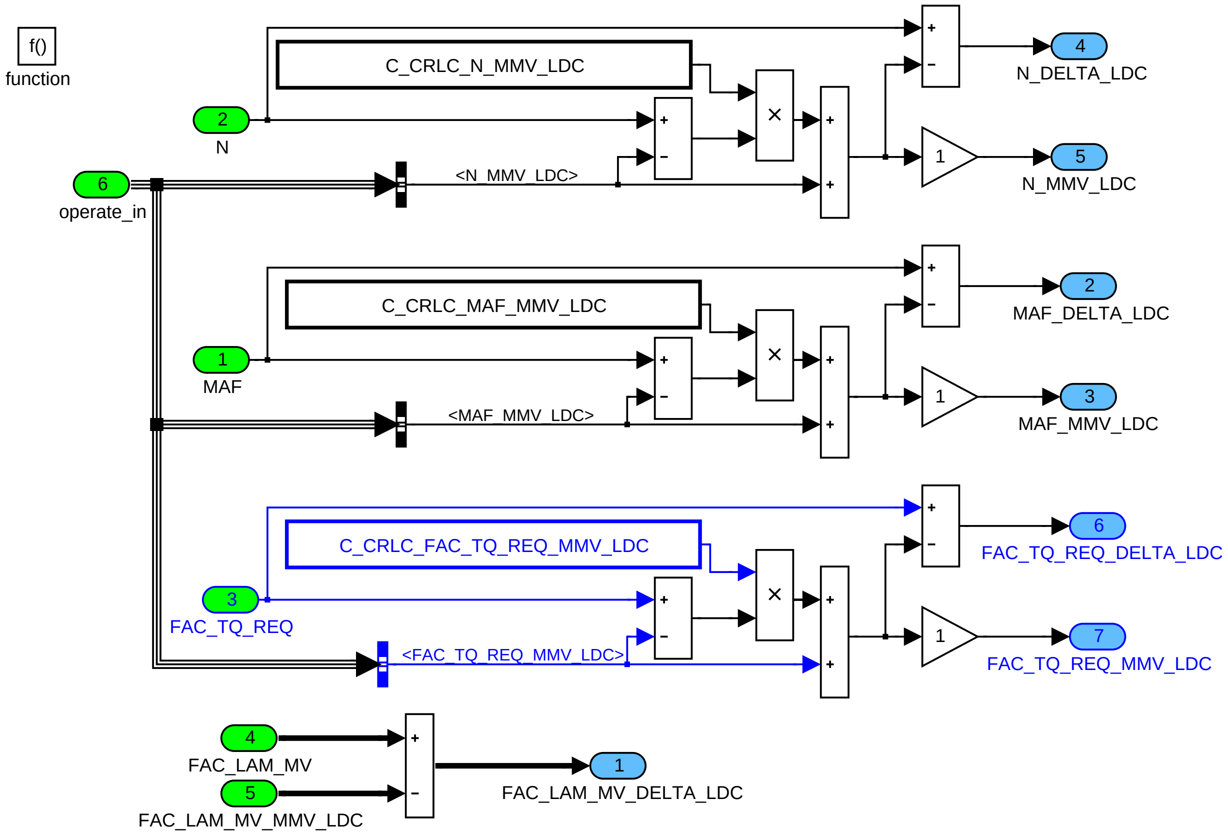


Figure 174 LACO_ISPCLMVLDC0/ operate

46.15 Application incidence for the Lambda Adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAM_AD_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
flag indicating that lambda adaptation can be activated					
LV_LAM_AD_WUP_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
flag indicating that lambda adaptation in warm-up phase can be activated					
LV_LAM_AD_ACT_ERR[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
flag indicating that lambda adaptation can be activated because of no error detected					
LV_LAM_AD_DEAC_ERR[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
flag indicating that lambda adaptation is deactivated due to an error					
CRLC_LAM_AD[NC_CBK_EX_NR]	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
correlation constant for calculation of adaptive values					
LV_LDC_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
limited dynamic conditions for lambda adaptation					
N_OFS_LDC_LAM_AD[NC_CBK_EX_NR]	V	E020...1FE0H	-8160...8160	1	[rpm]
engine speed offset for limited dynamics calculation					
MAF_OFS_LDC_LAM_AD[NC_CBK_EX_NR]	V	8000...7FFFH	-694.51059...694.48940	0.0211948	[mg/stk]
air mass flow offset for limited dynamics calculation					
FAC_LAM_MV_OFS_LDC_LAM_AD[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
filtered lambda controller output offset for limited dynamics calculation					
MAF_INT_LDC_LAM_AD[NC_CBK_EX_NR]	V	0...FFFFH	0...1820.42	0.0277778	[g]
air mass flow integral during limited dynamics not fulfilled					
LV_LAM_AD_CDN_H_RNG[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
condition bit for activation of lambda adaptation ADAPT_FAC_H					
LV_LAM_AD_CDN_L_RNG[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
condition bit for activation of lambda adaptation ADAPT_FAC_L					
LV_LAM_AD_CDN_ADD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
condition bit for activation of lambda adaptation ADAPT_ADD					
T_WAIT_MAX_LAM_AD[NC_CBK_EX_NR]	-	0...FFH	0...5.1	0.02	[s]
threshold of waiting time between two adaptations					
MFF_SP_AD[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Mass fuel flow setpoint for lambda adaptation - bank selective					
LV_FAC_LAM_AD_SHIFT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag to request shift of Lambda adaptation					
FAC_LAM_AD_SHIFT[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Output value for shift of Lambda adaptation					

Input data:

LV_ERR_EL_CPS	LV_LAM_LSCL[NC_CBK_EX_NR]	FAC_LAM_MV_DELTA_LDC[NC_CBK_EX_NR]	LV_ERR_MTC_CTL_2
LV_ERR_DIAGCPS	LV_LAM_STOP[NC_CBK_EX_NR]	LV_ERR_TPS_1[NC_ETC_NR]	LV_ERR_MTC_CTL_3
NC_CYL_NR	MAF_CYL	LV_ERR_TPS_2[NC_ETC_NR]	LV_ERR_MTC_DR
AMP	MAF_DELTA_LDC	LV_ERR_LOAD_TPS_PLAUS	STATE_LAM_AD[NC_CBK_EX_NR]
LV_ERR_MAF	MAF_KGH	LV_LAM_LSCL[NC_CBK_EX_NR]	

LV_ERR_TCO_EL	N_DELTA_LDC	LV_LAM_STOP[NC_CBK_EX_NR]	LV_HOM_ACT
LV_ERR_TCO_GRD	LV_ERR_TCO_PLAUS	NC_CBK_EX_NR	MFF_SP_S_SWI_HOM
LV_ERR_IGC	TCO	TIA[NC_SENS_NR_TIA]	TAM
LV_ERR_TCO_STUCK	FUP[NC_CBK_HPP_NR]	LAMB_SP_HOM[NC_CBK_EX_NR]	LV_ERR_MEC_IVVT_IN[NC_NR_CBK_IVVT]
LV_ERR_TCO_STUCK_RNG	LV_ERR_AMP	LV_FAC_H_RNG_LAM_AD[NC_CBK_EX_NR]	LV_ERR_MEC_IVVT_EX[NC_NR_CBK_IVVT]
LV_ERR_VLS_DOWN_DIF[NC_CBK_EX_NR]	LV_ERR_AMP_PLAUS	LV_FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	LV_ERR_SLV_IVVT_EX[NC_NR_CBK_IVVT]
LV_ERR_DELTA_I_LAM[NC_CBK_EX_NR]	LV_ERR_TPS_RATIO[NC_ETC_NR]	LV_MFF_ADD_RNG_LAM_AD[NC_CBK_EX_NR]	LV_ERR_SLV_IVVT_IN[NC_NR_CBK_IVVT]
LV_ERR_VCV	LV_ERR_TPS_ST_CHK_2	LV_ERR_TPS_MAF_1[NC_ETC_NR]	LV_ERR_TPS_MAF_2[NC_ETC_NR]
LV_ERR_VCV_PLAUS	LV_ERR_FUP[NC_CBK_HPP_NR]	LV_ERR_FUP_MFP_PLAUS	LV_LAM_AD_ENA
LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_TPS_AD[NC_ETC_NR]	MFF_ADD_LAM_AD[NC_CBK_EX_NR]	WGPWM[NC_NR_TCHA]
LV_CP_RAMP_OPEN_ACT	LV_LAM_LIM_LAM_AD[NC_CBK_EX_NR]	FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	LV_ERR_FUP_EFP
LV_LAM_AD_EXT	LV_ERR_MAP_TPS_PLAUS	LV_INH_FSD_STOP_OIL	LV_ERR_LS_UP[NC_CBK_EX_NR]

FAC_LAM_AD_CUS_SHIFT[NC_CBK_EX_NR]	LV_FAC_LAM_AD_CUS_SHIFT[NC_CBK_EX_NR]	MFF_SP_HOM_BAS_MV	LV_LAM_AD_INJ_ACT

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_AMP_MIN_LAM_AD	1	0...FFFFH	0...5434	0.0829175	[hPa]
Minimun ambient pressure to activate lambda adaptation					
C_TAM_MIN_LAM_AD	1	0...FEH	-48...142.5	0.75	[°C]
Minimum ambient temperature for activation of lambda adaptation					
C_TCO_MIN_LAM_AD	1	0...FEH	-48...142.5	0.75	[°C]
minimum coolant temperature for adaptive learning					
C_TCO_MIN_LAM_AD_WUP	1	0...FEH	-48...142.5	0.75	[°C]
minimum coolant temperature for Lambda adaptation in warm-up phase					
C_TCO_MAX_LAM_AD_WUP	1	0...FEH	-48...142.5	0.75	[°C]
maximum coolant temperature for Lambda adaptation in warm-up phase					
C_TIA_MAX_LAM_AD	1	0...FEH	-48...142.5	0.75	[°C]
maximum air temperature for adaptive learning					
C_MAF_KGH_MAX_LAM_AD	1	0...FFFFH	0...1023.98437	0.015625	[kg/h]
maximum mass air flow for adaptive learning					
C_MAF_KGH_MIN_LAM_AD_WUP	1	0...FFFFH	0...1023.98437	0.015625	[kg/h]
minimum mass air flow for adaptive learning during warm-up phase					
C_MAF_KGH_MAX_LAM_AD_WUP	1	0...FFFFH	0...1023.98437	0.015625	[kg/h]
maximum mass air flow for adaptive learning during warm-up phase					
C_FUP_MIN_LAM_AD	1	0...FFFFH	0...347776	5.3067216	[hPa]
Minimun fuel pressure mean value to activate lambda adaptation					
C_FUP_MAX_LAM_AD	1	0...FFFFH	0...347776	5.3067216	[hPa]
Minimun fuel pressure mean value to activate lambda adaptation					
C_CRLC_LAM_AD_L	1	0...FFH	0...0.99609	3.9063e-3	[-]
correlation constant for lambda adaptation at low area					
C_CRLC_LAM_AD_H	1	0...FFH	0...0.99609	3.9063e-3	[-]
correlation constant for lambda adaptation at high area					
C_CRLC_LAM_AD_ADD	1	0...FFH	0...0.99609	3.9063e-3	[-]
correlation constant for lambda adaptation at offset learning area					
C_CRLC_LAM_AD_L_LAM_LIM	1	0...FFH	0...0.99609	3.9063e-3	[-]
correlation constant for lambda adaptation at low area when forced modus is required					
C_CRLC_LAM_AD_H_LAM_LIM	1	0...FFH	0...0.99609	3.9063e-3	[-]
correlation constant for lambda adaptation at high area when forced modus is required					
C_CRLC_LAM_AD_ADD_LAM_LIM	1	0...FFH	0...0.99609	3.9063e-3	[-]
correlation constant for lambda adaptation at offset learning area when forced modus is required					
C_N_DYW_LDC_LAM_AD	1	0...1FE0H	0...8160	1	[rpm]
engine speed window for limited dynamic conditions trim control					
C_MAF_DYW_LDC_LAM_AD	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
air mass flow window for limited dynamic conditions lambda adaptation					
C_FAC_LAM_MV_DYW_LDC_LAM_AD	1	0...7FFFH	0...49.99847	1.5259e-3	[%]
lambda control output window for limited dynamic conditions lambda adaptation					
C_MAF_INT_LDC_LAM_AD	1	0...FFFFH	0...1820.42	0.0277778	[g]
air mass flow integral for duration of violation of limited dynamic conditions lambda adaptation					
C_LAMB_SP_AD_MIN_H	1	0...7FFFH	0...1.99993	0.061e-3	[-]
minimum lambda setpoint to allow lambda adaptation at high range operating point					
C_LAMB_SP_AD_MAX_H	1	0...7FFFH	0...1.99993	0.061e-3	[-]
maximum lambda setpoint to allow lambda adaptation at high range operating point					
C_LAMB_SP_AD_MIN_L	1	0...7FFFH	0...1.99993	0.061e-3	[-]
minimum lambda setpoint to allow lambda adaptation at low range operating point					
C_LAMB_SP_AD_MAX_L	1	0...7FFFH	0...1.99993	0.061e-3	[-]
maximum lambda setpoint to allow lambda adaptation at low range operating point					
C_LAMB_SP_AD_MIN_ADD	1	0...7FFFH	0...1.99993	0.061e-3	[-]
minimum lambda setpoint to allow lambda adaptation at offset learning area					
C_LAMB_SP_AD_MAX_ADD	1	0...7FFFH	0...1.99993	0.061e-3	[-]
maximum lambda setpoint to allow lambda adaptation at offset learning area					

C_LAMB_SP_AD_MIN_WUP_H	1	0...7FFFH	0...1.99993	0.061e-3	[-]
minimum lambda setpoint to allow lambda adaptation at high range operating point during warm-up phase					
C_LAMB_SP_AD_MAX_WUP_H	1	0...7FFFH	0...1.99993	0.061e-3	[-]
maximum lambda setpoint to allow lambda adaptation at high range operating point during warm-up phase					
C_LAMB_SP_AD_MIN_WUP_L	1	0...7FFFH	0...1.99993	0.061e-3	[-]
minimum lambda setpoint to allow lambda adaptation at low range operating point during warm-up phase					
C_LAMB_SP_AD_MAX_WUP_L	1	0...7FFFH	0...1.99993	0.061e-3	[-]
maximum lambda setpoint to allow lambda adaptation at low range operating point during warm-up phase					
C_LAMB_SP_AD_MIN_WUP_ADD	1	0...7FFFH	0...1.99993	0.061e-3	[-]
minimum lambda setpoint to allow lambda adaptation at offset learning area during warm-up phase					
C_LAMB_SP_AD_MAX_WUP_ADD	1	0...7FFFH	0...1.99993	0.061e-3	[-]
maximum lambda setpoint to allow lambda adaptation at offset learning area during warm-up phase					
LC_LAM_AD_WUP_DEAC	1	0...1H	0...1	1	[-]
logical calibration to disable Lambda adaptation in warm-up pahse					
LC_LAM_AD_WUP_ADD_RNG_CDN	1	0...1H	0...1	1	[-]
Logical switch to require previous adaptation in ADD Range					
LC_LAM_AD_WUP_L_RNG_CDN	1	0...1H	0...1	1	[-]
Logical switch to require previous adaptation in FAC_L Range					
C_T_WAIT_MAX_LAM_AD	1	0...FFH	0...5.1	0.02	[s]
threshold of waiting time between two adaptations					
C_T_WAIT_MAX_LAM_AD_H_RNG	1	0...FFH	0...5.1	0.02	[s]
threshold of waiting time between two adaptations in high adaptation range					
C_T_WAIT_MAX_LAM_AD_L_RNG	1	0...FFH	0...5.1	0.02	[s]
threshold of waiting time between two adaptations in low adaptation range					
C_T_WAIT_MAX_LAM_AD_ADD	1	0...FFH	0...5.1	0.02	[s]
threshold of waiting time between two adaptations at offset learning area					
C_WGPWM_MIN_LAM_AD	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Minimum waste gate signal for lambda adaptation at active driving cycle					
LC_WGPWM_INH_LAM_AD_ENA	1	0...1H	0...1	1	[-]
Logical constant to enable inhibition of the lambda adaptation below a wastegate PWM value					
LC_LAM_AD_INH_STOP_OIL	1	0...1H	0...1	1	[-]
Logical constant to inibit the lambda adaptation when LV_INH_FSD_STOP_OIL is set.					
LC_LAM_AD_WUP_INH_STOP_OIL	1	0...1H	0...1	1	[-]
Logical constant to inibit the lambda adaptation WUP when LV_INH_FSD_STOP_OIL is set.					
C_T_AST_BOL_INH_LAM_AD	1	0...FFFFH	0...6553.5	0.1	[s]
Bottom of boundary of time window to inhibit lambda adaption					
C_T_AST_TOL_INH_LAM_AD	1	0...FFFFH	0...6553.5	0.1	[s]
Top of boundary of time window to inhibit lambda adaption					
C_TCO_MAX_T_AST_INH_LAM_AD	1	0...FEH	-48...142.5	0.75	[°C]
Max. TCO to allow time window to inhibit lambda adaption					

FUNCTION DESCRIPTION:

General information:

The calculation shall be done for all exhaust cylinder banks. For instance, if two separate catalyst systems are concerned (NC_CBK_EX_NR = 2) then

i = 1, for exhaust cylinder bank 1
i = 2, for exhaust cylinder bank 2,
otherwise (NC_CBK_EX_NR = 1)
i = 1, for single exhaust cylinder bank.

Description:

The activation conditions of the lambda adaptation depends on:

- coolant temperature thresholds,
- temperature of intake air into manifold threshold,
- air mass flow threshold,
- secondary air function including diagnosis finished
- lambda setpoint thresholds
- and several error flags that must be zero

If all activation conditions are fulfilled, during the warm-up phase LV_LAM_AD_WUP_ACT[i] is set and during normal warm conditions LV_LAM_AD_ACT[i] is set to 1. That means that the lambda adaptation can get active if no other inhibition defined in the lambda adaptation module is set.

The detection of the limited dynamics is calculated based on the current engine speed, air mass flow and filtered lambda control output. The input to this function is the difference between each of the mentioned base values and its moving mean values. If one of those differences exceeds a calibrated threshold, an air mass flow integral is started to be calculated until a calibrated MAF threshold is reached. During the calculation the limited dynamics are defined to be not fulfilled (LV_LDC_LAM_AD[i] = 0). In order to reset the difference between base value and moving mean value after each threshold exceeding, an offset variable is introduced (see formula section).

Furthermore the correlation constant for calculation of adaptive values and the distance between two adaptations are determined dependent on the different adaptation ranges.

Application conditions:

Initialisation: *at ECU reset:*

LV_LAM_AD_ACT[i]	*= 0*
LV_LAM_AD_WUP_ACT[i]	*= 0*
LV_LAM_AD_ACT_ERR[i]	*= 0*
CRLC_LAM_AD[i]	*= 0*
T_WAIT_MAX_LAM_AD[i]	*= 0*
LV_LDC_LAM_AD[i]	*= 0*
N_OFS_LDC_LAM_AD[i]	*= 0 rpm*
MAF_OFS_LDC_LAM_AD[i]	*= 0 mg/stk*
FAC_LAM_MV_OFS_LDC_LAM_AD[i]	*= 0 %*
MAF_INT_LDC_LAM_AD[i]	*= 0 g*
LV_LAM_AD_DEAC_ERR[i]	*= 0*
LV_LAM_AD_CDN_H_RNG[i]	*= 0*
LV_LAM_AD_CDN_L_RNG[i]	*= 0*
LV_LAM_AD_CDN_ADD[i]	*= 0*
MFF_SP_AD[i]	*= MFF_SP_HOM_BAS_MV*

At transition LV_IGK = 1 -> 0:

LV_LAM_AD_ACT[i] = 0

LV_LAM_AD_WUP_ACT[i] = 0

Recurrence: T_SAMPLE = 20 ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

Determination of activation conditions for lambda adaptation:

```
    If
    LV_ERR_EL_CPS = 0                and
    LV_ERR_DIAGCPS = 0               and
    LV_ERR_TCO_EL = 0                and
    LV_ERR_TCO_GRD = 0               and
    LV_ERR_TCO_PLAUS = 0             and
    LV_ERR_TCO_STUCK = 0             and
    LV_ERR_MAF = 0                   and
    LV_ERR_LOAD_TPS_PLAUS = 0        and
    LV_ERR_TPS_1 = 0                 and
    LV_ERR_TPS_2 = 0                 and
    LV_ERR_TPS_RATIO = 0             and
    LV_ERR_TPS_AD = 0                and
    LV_ERR_TPS_MAF_1 = 0             and
    LV_ERR_TPS_MAF_2 = 0             and
    LV_ERR_TPS_ST_CHK_2 = 0          and
    LV_ERR_MAP_TPS_PLAUS = 0         and
    LV_ERR_MEC_IVVT_IN = 0           and
    LV_ERR_MEC_IVVT_EX = 0           and
    LV_ERR_SLV_IVVT_IN = 0           and
    LV_ERR_SLV_IVVT_EX = 0           and
    LV_ERR_IGC                       and
    LV_ERR_AMP = 0                   and
    LV_ERR_AMP_PLAUS = 0             and
    LV_ERR_VCV = 0                   and
    LV_ERR_VCV_PLAUS = 0             and
    LV_ERR_FUP = 0                   and
    LV_ERR_FUP_EFP = 0               and
    LV_ERR_FUP_MFP_PLAUS = 0         and
    LV_ERR_MTC_CTL_2 = 0             and
    LV_ERR_MTC_CTL_3 = 0             and
    LV_ERR_MTC_CTL_DR = 0            and
    LV_ERR_DELTA_I_LAM[i] = 0        and
    LV_ERR_VLS_DOWN_DIF[i] = 0       and
    LV_ERR_TCO_STUCK_RNG = 0         and
    LV_ERR_TTIP_MES_LSH_UP_1 = 0 and
    LV_ERR_TTIP_MES_LSH_UP_2 = 0

then      LV_LAM_AD_ACT_ERR[i]  =  0
else      LV_LAM_AD_ACT_ERR[i]  =  1
endif
```

```
% reset of adaption values in case of error:
LV_LAM_AD_DEAC_ERR[i]  =  LV_LAM_AD_ACT_ERR[i] OR LV_ERR_LS_UP[i]

If   LC_LAM_AD_WUP_DEAC = 1                             or
     LV_LAM_AD_INJ_ACT = 1                              or
     LV_LAM_AD_ACT[i]  =  1                             or
     LV_CP_RAMP_OPEN_ACT  =  1                          or
     LV_LAM_AD_ACT_ERR[i] = 1                           or
     TCO < C_TCO_MIN_LAM_AD_WUP                         or
     TCO > C_TCO_MAX_LAM_AD_WUP                         or
     TIA > C_TIA_MAX_LAM_AD                             or
     MAF_KGH < C_MAF_KGH_MIN_LAM_AD_WUP                 or
     MAF_KGH > C_MAF_KGH_MAX_LAM_AD_WUP                         or
     TAM < C_TAM_MIN_LAM_AD                             or
     AMP < C_AMP_MIN_LAM_AD                             or
     FUP < C_FUP_MIN_LAM_AD                             or
     FUP > C_FUP_MAX_LAM_AD                             or
      (MFF_ADD_LAM_AD[i] = 0  &
       LC_LAM_AD_WUP_ADD_RNG_CDN = 1)                   or
      (FAC_L_RNG_LAM_AD[i] = 0  &
        LC_LAM_AD_WUP_L_RNG_CDN = 1)                    or
     (LV_INH_FSD_STOP_OIL=1 & LC_LAM_AD_WUP_INH_STOP_OIL = 1)

then        LV_LAM_AD_WUP_ACT[i]  =  0
else        LV_LAM_AD_WUP_ACT[i]  =  1
endif

If   LV_LAM_AD_ACT_ERR[i] = 1                                         or
     LV_LAM_AD_WUP_ACT[i]  =  1                                       or
     LV_LAM_AD_INJ_ACT = 1                                            or
     TCO < C_TCO_MIN_LAM_AD                                           or
     TIA > C_TIA_MAX_LAM_AD                                           or
     MAF_KGH > C_MAF_KGH_MAX_LAM_AD                                   or
     TAM < C_TAM_MIN_LAM_AD                                           or
     AMP < C_AMP_MIN_LAM_AD                                           or
     FUP < C_FUP_MIN_LAM_AD                                           or
```

```
        FUP > C_FUP_MAX_LAM_AD                                          or
        ( LV_FAC_H_RNG_LAM_AD[i] = 1  and
        C_LAMB_SP_AD_MIN_H > LAMB_SP_HOM[i] > C_LAMB_SP_AD_MAX_H )      or
        ( LV_FAC_L_RNG_LAM_AD[i] = 1  and
        C_LAMB_SP_AD_MIN_L > LAMB_SP_HOM[i] > C_LAMB_SP_AD_MAX_L )      or
        ( LV_MFF_ADD_RNG_LAM_AD[i] = 1 and
        C_LAMB_SP_AD_MIN_ADD > LAMB_SP_HOM[i] > C_LAMB_SP_AD_MAX_ADD ) or
        (WGPWM[i] ≤ C_WGPWM_MIN_LAM_AD and  LV_LAM_CYL_ACT = 1 and
    LC_WPPWM_INH_LAM_AD_ENA = 1)                                        or
        (LV_INH_FSD_STOP_OIL=1 & LC_LAM_AD_INH_STOP_OIL = 1)            or
        ((C_T_AST_BOL_INH_LAM_AD < T_AST < C_T_AST_TOL_INH_LAM_AD)
        & TCO < C_TCO_MAX_T_AST_INH_LAM_AD)
    then        LV_LAM_AD_ACT[i]  =  0
    else        LV_LAM_AD_ACT[i]  =  1
    endif


    if  LV_LAM_AD_ENA = 1          and
        LV_LAM_LSCL[i] = 1                  and
        LV_LAM_STOP[i] = 0         and
        LV_LDC_LAM_AD[i] = 1
    then
        LV_LAM_AD_CDN_L_RNG[i] =1
        if      LV_LAM_AD_EXT = 0
        then    LV_LAM_AD_CDN_H_RNG[i] = 1
                LV_LAM_AD_CDN_ADD[i] = 1
        else    LV_LAM_AD_CDN_H_RNG[i] = 0
                LV_LAM_AD_CDN_ADD[i] = 0
        end
    else
        LV_LAM_AD_CDN_H_RNG[i] = 0
        LV_LAM_AD_CDN_L_RNG[i] = 0
        LV_LAM_AD_CDN_ADD[i] = 0
    endif
```

Determination of mass fuel flow setpoint for lambda adaptation:

```
    If   LV_HOM_ACT = 1
    Then        MFF_SP_AD[i] = MFF_SP_HOM_BAS_MV
```

```
              % set value of MFF_SP_HOM_BAS_MV to both banks.
Else          MFF_SP_AD[i] = MFF_SP_S_SWI_HOM
              % set value of MFF_SP_S_SWI_HOM to both banks.
Endif
```

Detection of limited dynamic conditions for lambda adaptation:

```
If   (ABS(N_DELTA_LDC) > C_N_DYW_LDC_LAM_AD)
  then
     If      (ABS(N_DELTA_LDC - N_OFS_LDC_LAM_AD[i]) > C_N_DYW_LDC_LAM_AD)
       then   N_OFS_LDC_LAM_AD[i]  =  N_DELTA_LDC
              MAF_INT_LDC_LAM_AD[i]  =  0
     endif
  else        N_OFS_LDC_LAM_AD[i]  =  0
endif


If   (ABS(MAF_DELTA_LDC) > C_MAF_DYW_LDC_LAM_AD)
  then
     If      (ABS(MAF_DELTA_LDC - MAF_OFS_LDC_LAM_AD[i]) >
                                                C_MAF_DYW_LDC_LAM_AD)
       then   MAF_OFS_LDC_LAM_AD[i]  =  MAF_DELTA_LDC
              MAF_INT_LDC_LAM_AD[i]  =  0
     endif
  else        MAF_OFS_LDC_LAM_AD[i]  =  0
endif


If   (ABS(FAC_LAM_MV_DELTA_LDC[i]) > C_FAC_LAM_MV_DYW_LDC_LAM_AD)
  then
     If      (ABS(FAC_LAM_MV_DELTA_LDC[i] - FAC_LAM_MV_OFS_LDC_LAM_AD[i]) >
                                          C_FAC_LAM_MV_DYW_LDC_LAM_AD)
       then  FAC_LAM_MV_OFS_LDC_LAM_AD[i]  =  FAC_LAM_MV_DELTA_LDC[i]
             MAF_INT_LDC_LAM_AD[i]  =  0
     endif
  else       FAC_LAM_MV_OFS_LDC_LAM_AD[i]  =  0
endif


If   (MAF_INT_LDC_LAM_AD[i] < C_MAF_INT_LDC_LAM_AD)
  then        MAF_INT_LDC_LAM_AD[i] [g] = MAF_INT_LDC_LAM_AD[i] [g] +
                          MAF_CYL [kg/h] * T_SAMPLE [ms] * 1/3600 [(g*h) / (kg*ms)]

              LV_LDC_LAM_AD[i]  =  0

  else        LV_LDC_LAM_AD[i]  =  1
endif
```

Determination of correlation constant for lambda adaptation and of distance between two adaptation calculations:

Note: Waiting time between two adaptations and correlation constant are solely dependent on the different adaptation ranges.

```
If      STATE_LAM_AD[i] = "ADAPT_ADD"
Then    T_WAIT_MAX_LAM_AD[i]  =  C_T_WAIT_MAX_LAM_AD_ADD
            If LV_LAM_LIM_LAM_AD[i] = 0
            Then CRLC_LAM_AD[i]  =  C_CRLC_LAM_AD_ADD
            Else  CRLC_LAM_AD[i]  =  C_CRLC_LAM_AD_ADD_LAM_LIM
            Endif
Else    If      STATE_LAM_AD[i] = "ADAPT_FAC_L"
        Then    T_WAIT_MAX_LAM_AD[i]  =  C_T_WAIT_MAX_LAM_AD_L_RNG
                If LV_LAM_LIM_LAM_AD[i] = 0
                Then CRLC_LAM_AD[i]  =  C_CRLC_LAM_AD_L
                Else  CRLC_LAM_AD[i]  =  C_CRLC_LAM_AD_L_LAM_LIM
                Endif
        Else    If      STATE_LAM_AD[i] = "ADAPT_FAC_H"
                Then    T_WAIT_MAX_LAM_AD[i]  = C_T_WAIT_MAX_LAM_AD_H_RNG
                        If LV_LAM_LIM_LAM_AD[i] = 0
                        Then CRLC_LAM_AD[i]  =  C_CRLC_LAM_AD_H
                        Else  CRLC_LAM_AD[i]  =  C_CRLC_LAM_AD_H_LAM_LIM
                        Endif
                Else    T_WAIT_MAX_LAM_AD[i]  =  C_T_WAIT_MAX_LAM_AD
                        If LV_LAM_LIM_LAM_AD[i] = 0
                        Then CRLC_LAM_AD[i]  =  C_CRLC_LAM_AD_H
                        Else  CRLC_LAM_AD[i]  =  C_CRLC_LAM_AD_H_LAM_LIM
                        Endif
                Endif
        Endif
Endif
```

Assignment of Lambda adaptation shift values

FAC_LAM_AD_SHIFT[i] = FAC_LAM_AD_CUS_SHIFT[i]

LV_FAC_LAM_AD_SHIFT[i] = LV_FAC_LAM_AD_CUS_SHIFT[i]

46.16 Activation conditions for fuel system diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_FSD_ACT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Global activation flag for fuel system diagnosis					
LV_INH_FSD_STOP_OIL	O/V	0...1H	0...1	1	-
Inhibition flag for fuel system diagnosis due to oil dilution under cold start conditions					
CTR_STOP_FSD	O/V/S	0...FFH	0...255	1	-
Counter value as a measure of the oil dilution under cold start conditions					
DLY_DEC_FSD_STOP_OIL	V	0...FFH	0...255	1	s
Time delay for decreasing the debounce counter for oil dilution detection in fuel system diagnosis					
T_DEC_FSD_STOP_OIL	V	0...FFH	0...255	1	s
Timer for decreasing the debounce counter for oil dilution detection in fuel system diagnosis					
LV_FSD_STOP_OIL_ES_OLD	-	0...1H	0...1	1	-
Value of last recurrence of LV_ES					

Input data:

LV_INH_DIAG_FSD[NC_CBK_EX_NR]	LV_INH_DIAG_FSD_CP[NC_CBK_EX_NR]	LV_LAM_LSCL[NC_CBK_EX_NR]	TCO_ST
NC_CBK_EX_NR	AMP	MAF_HB	N_32
TCO	TIA_IM	TOIL	LV_ES
TAM			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_AMP_MIN_FSD	1	0...FFFFH	0...5.434E+3	0.08291752	hPa
Minimum ambient pressure threshold for fuel system diagnosis					
C_CTR_MAX_FSD_STOP_OIL	1	0...FFH	0...255	1	-
Maximum value for the debounce counter for oil dilution detection in fuel system diagnosis					
C_MAF_MIN_FSD	1	0...FFH	0...1.389E+3	5.44705882	mg/stk
Mass air flow threshold for fuel system diagnosis					
C_N_MIN_FSD	1	0...FFH	0...8.16E+3	32	rpm
Engine speed threshold for fuel system diagnosis					
C_TAM_MIN_FSD	1	0...FEH	-48...142.5	0.75	°C
Minimum ambient temperature threshold for fuel system diagnosis					
C_TCO_MIN_FSD	1	0...FEH	-48...142.5	0.75	°C
Minimum threshold of TCO for fuel system diagnosis					
C_TIA_MIN_FSD	1	0...FEH	-48...142.5	0.75	°C
Minimum intake air temperature threshold for fuel system diagnosis					
C_TOIL_MIN_DEC_FSD_STOP_OIL	1	0...C8H	-40...160	1	°C
Minimum oil temperature threshold to activate timer to debounce oil dilution detection in fuel system diagnosis					
ID_CTR_INC_FSD_STOP_OIL	4	0...FFH	0...255	1	-
LDP_TCO_ST_ID_CTR_INC_FSD_STOP	4	0...FEH	-48...142.5	0.75	°C
Increasing value for the debounce counter for oil dilution detection in fuel system diagnosis					
ID_DLY_DEC_FSD_STOP_OIL	4	0...FFH	0...255	1	s
LDP_TOIL_ID_DLY_DEC_FSD_OIL	4	0...C8H	-40...160	1	°C
Time delay for decreasing the debounce counter for oil dilution detection in fuel system diagnosis					

46.16.1 General information

This function checks all necessary activation conditions in order to activate fuel system diagnosis. High rate of evaporated fuel during rich warm-up phase under cold start conditions could lead to wrong failure detection and therefore a special inhibition flag for that case is generated.

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

Application Condition

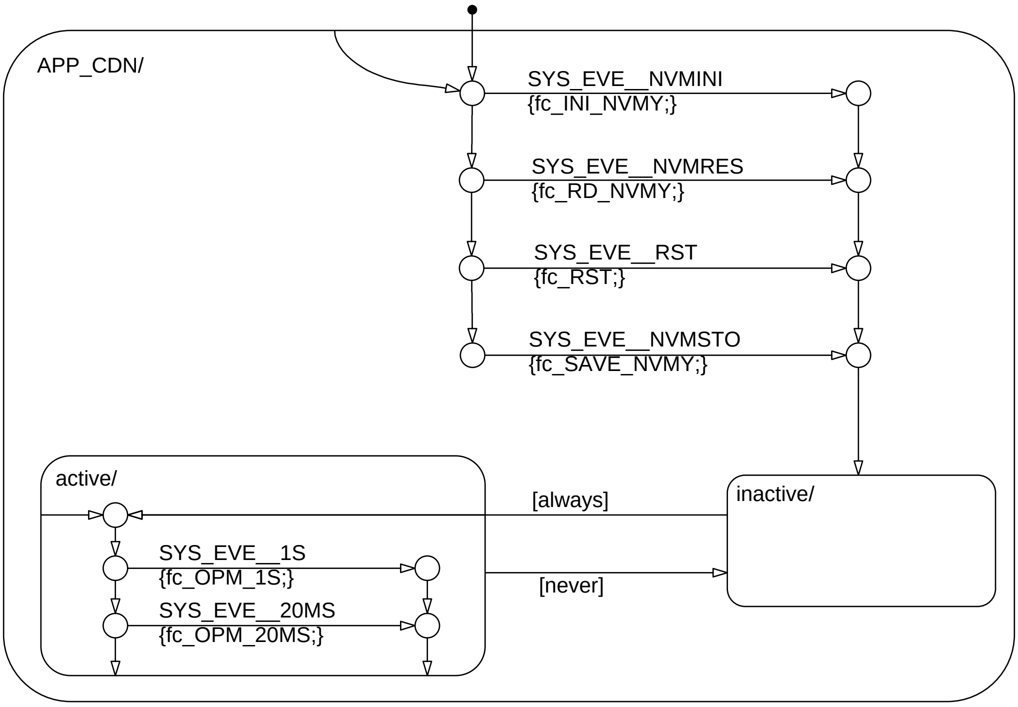


Figure 175 LACO_FCTDGFSDA0/ APP_CDN/ Chart

Function Description

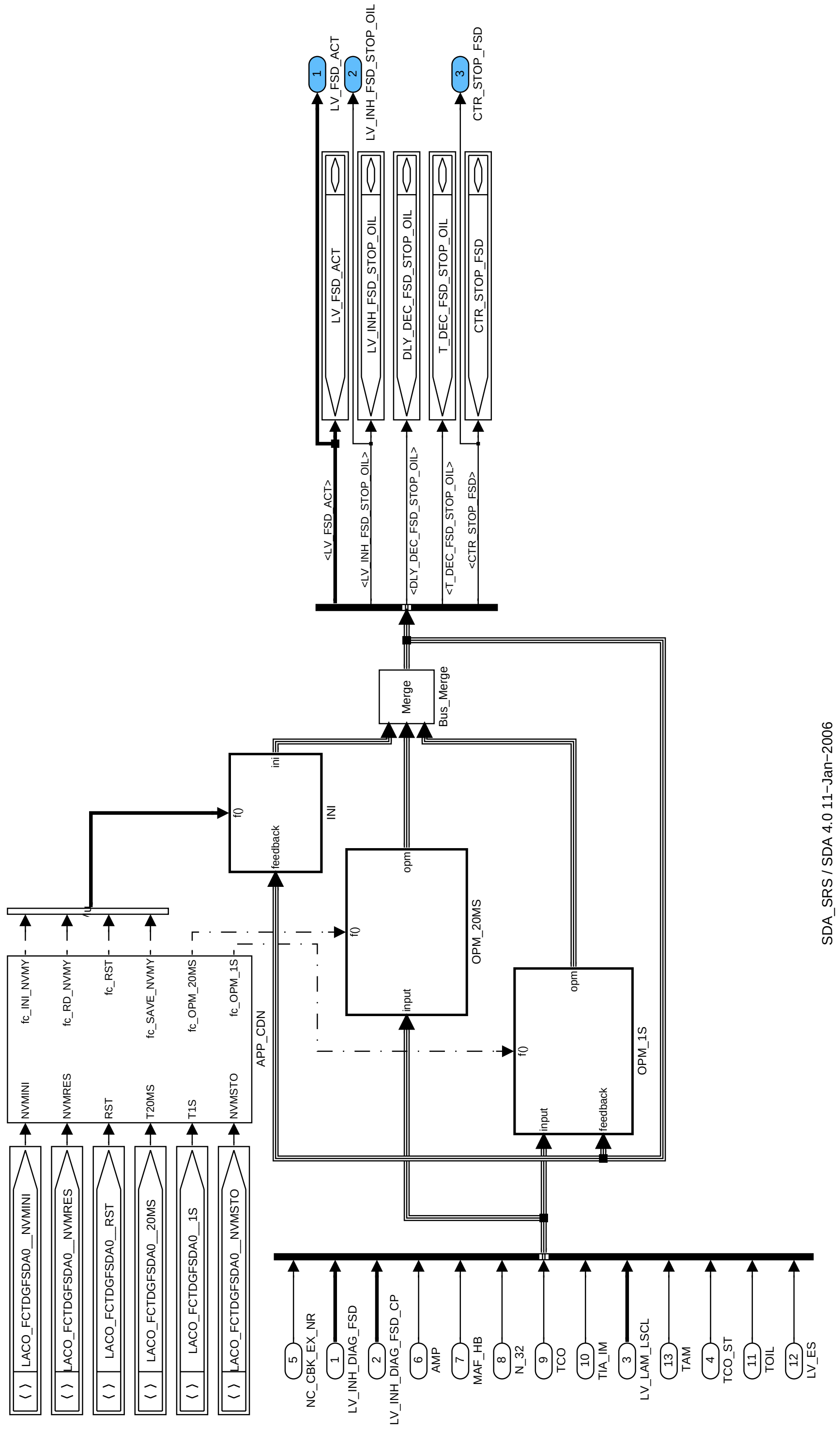


Figure 176 LACO_FCTDGFSDA0

46.16.1.1 Initialization

Initialization at ECU reset

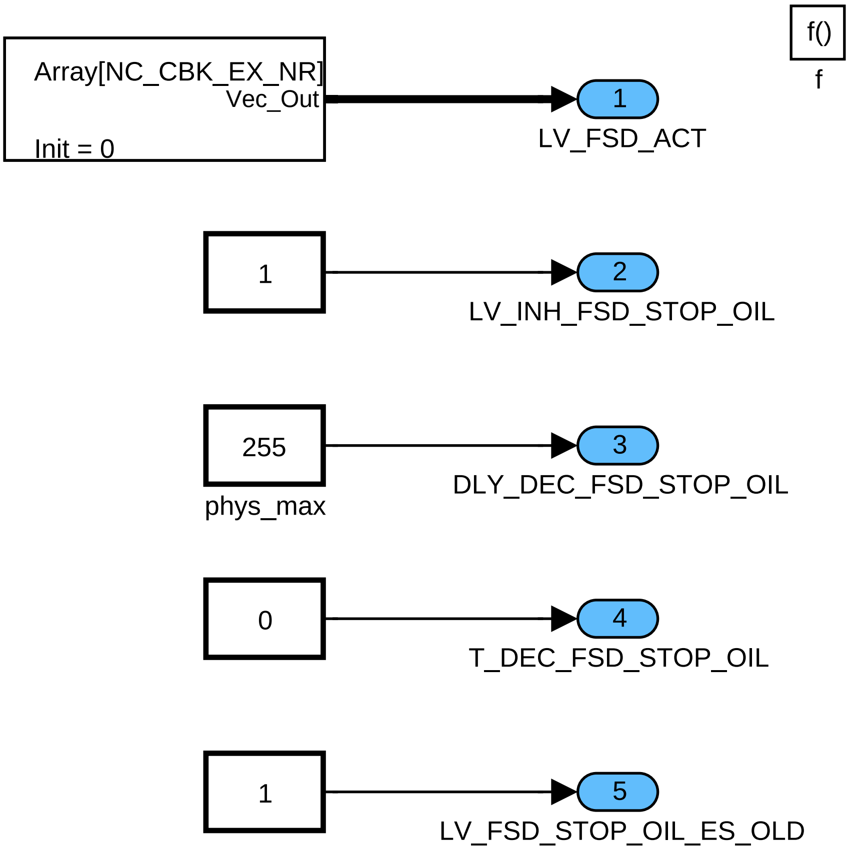


Figure 177 LACO_FCTDGFSDA0/ INI/ RST

Initialize NVMY data

CTR_STOP_FSD is set to 0 in case of failure in the non-volatile memory data storage.

Read NVMY data

CTR_STOP_FSD is restored from the non-volatile memory data storage after reset.

Store NVMY data

CTR_STOP_FSD is saved in the non-volatile memory data storage during PWL phase.

46.16.1.2 Formula section for 20ms task

Exhaust bank specific functionality

Activation of fuel system diagnosis

Calculation of LV_FSD_ACT[i]

In order to activate fuel system diagnosis no inhibition must be present, lambda controller must be active and ambient pressure, air mass flow, engine speed, coolant temperature and intake air temperature must be above their thresholds.

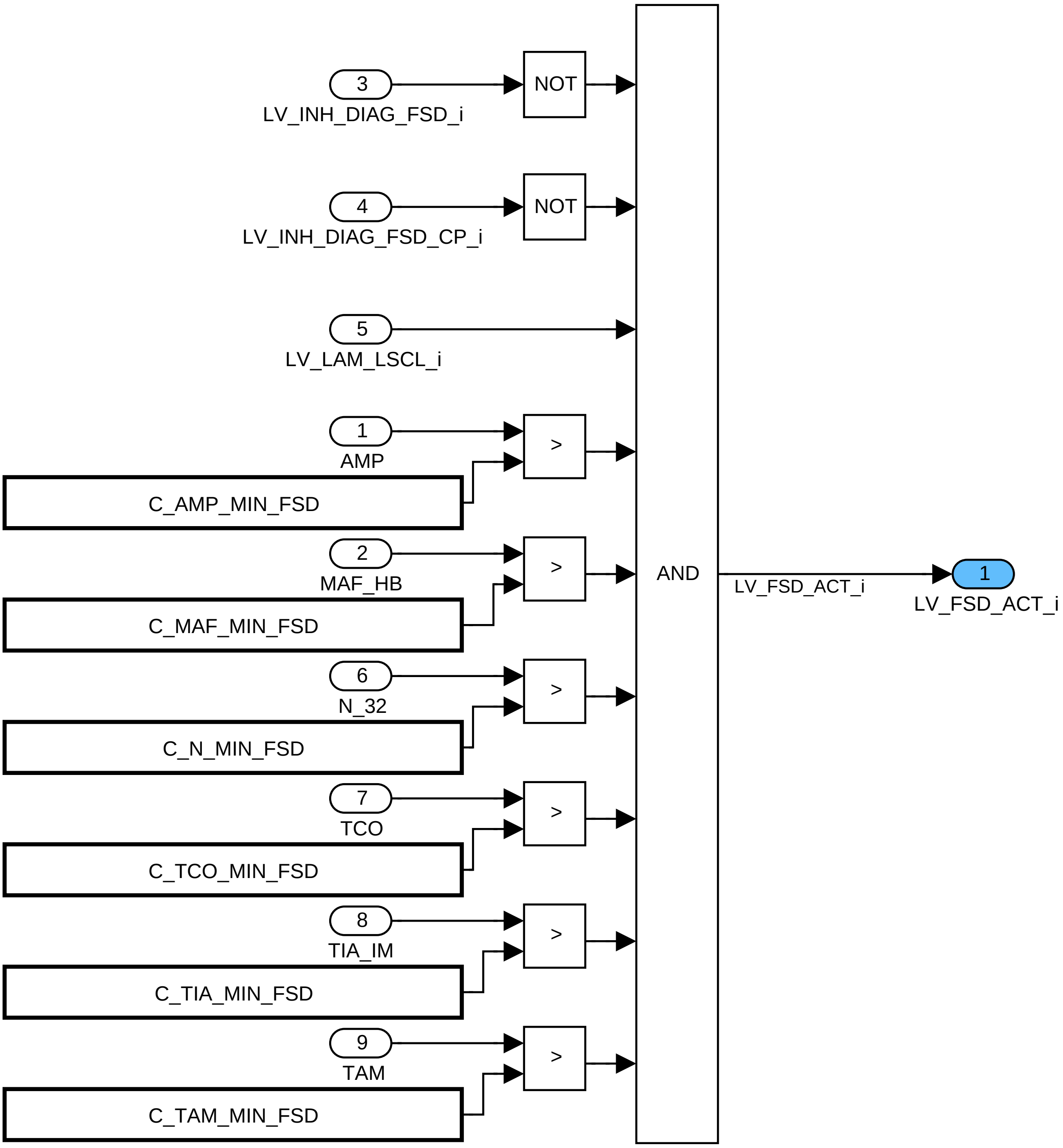


Figure 178 LACO_FCTDGFSDA0/ OPM_20MS/ OPM/ FSD_ACT/ FSD_ACT

46.16.1.3 Formula section for 1s task

For high rate of evaporated fuel (diluted in the engine oil) during rich warm-up phase under cold start conditions lambda controller and adaptation may reach the negative limits relevant for FSD (MIN limits). In order to prevent an error detection in that case the flag LV_INH_FSD_STOP_OIL used by FSD is set. The counter CTR_STOP_FSD is a measure for the amount of diluted fuel in the engine oil.

Increment of CTR_STOP_FSD

At each transition from engine stop to engine run (LV_ES switches from 1 to 0) the counter CTR_CTOP_FSD is incremented by the TCO_ST depending map value ID_CTR_INC_FSD_STOP_OIL.

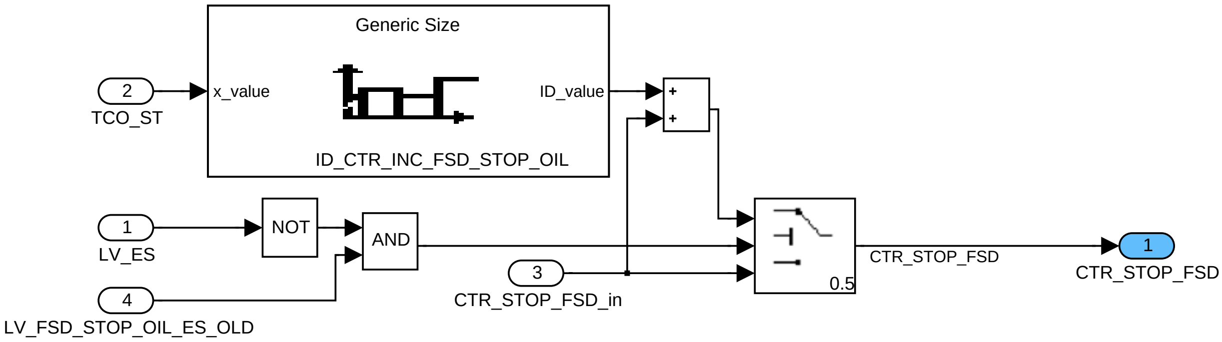


Figure 179 LACO_FCTDGFSDA0/ OPM_1S/ INC_CTR

Decrement CTR_STOP_FSD

ID_DLY_DEC_FSD_STOP_OIL defines the time to elapse at a certain TOIL range for decrementing CTR_STOP_FSD by 1. The TOIL range is defined by 2 adjacent map break points. At engine stop and as long as the oil temperature is below a threshold this timer is set to 0 in order to not run the increment functionality. The timer is reset to 0 as soon the threshold is reached.

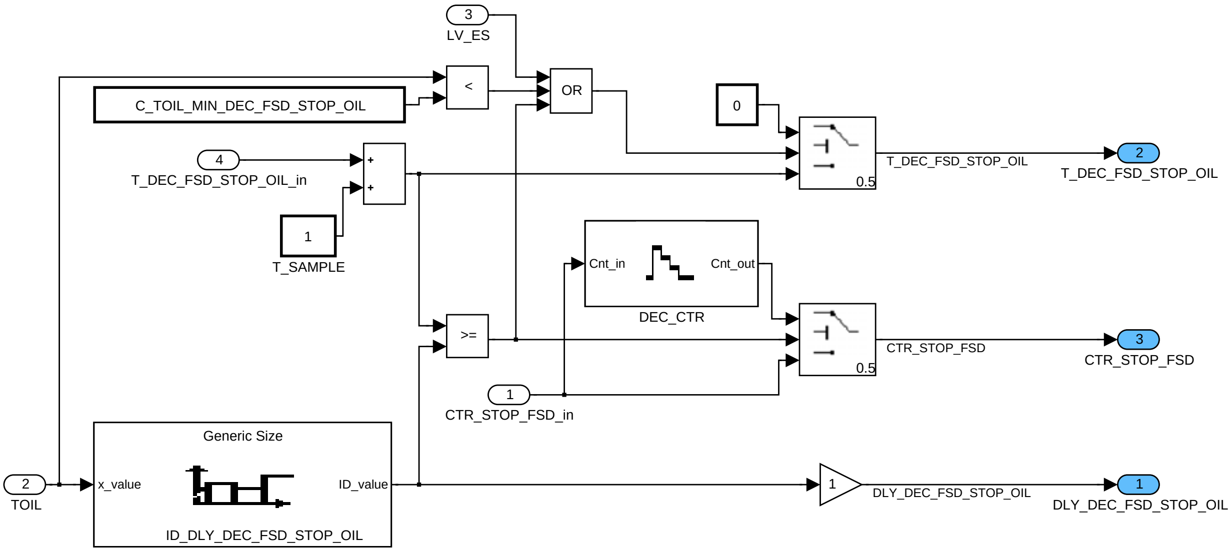


Figure 180 LACO_FCTDGFSDA0/ OPM_1S/ DEC_CTR

Inhibition of fuel system diagnosis

The flag LV_INH_FSD_STOP_OIL is set if the counter CTR_STOP_FSD reaches a calibration threshold.

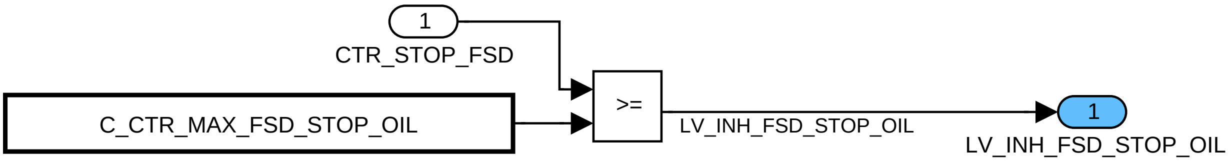


Figure 181 LACO_FCTDGFSDA0/ OPM_1S/ INH_FSD

46.17 Fuel system diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_FSD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Failure of lambda adaptation monitoring - relevant to malfunction indication light					
LV_ERR_FSD_LAM_LIM[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Failure for lambda controller in dead stop - relevant to malfunction indication light					
LV_ERR_FSD_H_RNG[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Failure in upper multiplicative adaptation learning area					
LV_END_DIAG_WIN_FSD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
End of diagnosis cycle for similar conditions window in additive adaptation learning area					
LV_END_DIAG_WIN_FSD_LAM_LIM[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
End of diagnosis cycle for similar conditions window in case of lambda control in dead stop					
LV_LAM_LIM_LAM_AD[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Request for forced lambda adaptation					
LV_LAM_ORNG_LAM_AD_REQ[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Flag indicating that request timer for forced lambda adaptation expired (lambda controller out of range for defined time)					
LV_CDN_DIAG_FSD_H_RNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Diagnosis conditions in upper multiplicative adaptation learning area					
LV_END_DIAG_FSD_H_RNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
End of first diagnosis cycle in upper multiplicative adaptation learning area					
T_SUM_END_DIAG_WIN_FSD_ADD[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Counter for the diagnosis window - additive adaptation learning area					
T_LAM_AD_ACT_FSD[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Timer indicating the forced lambda adaptation is running					
CTR_LAM_LIM_LAM_AD_REQ[NC_CBK_EX_NR]	V	0...FFFFH	0...6.5535E+4	1	-
Counter for activation of forced lambda adaptation					
ERR_SYM_FSD[NC_CBK_EX_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected failure of each symptom in case for lambda adaptation monitoring - relevant to MIL					
T_SUM_END_DIAG_WIN_FSD[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Counter for the diagnosis window - relevant to MIL					
T_SUM_END_DIAG_WIN_FSD_H_RNG[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Counter for the diagnosis window - upper multiplicative adaptation learning area					
T_SUM_END_DIAG_WIN_FSD_LAM[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Counter for the diagnosis window - lambda control in dead stop					
T_SUM_MAX_FSD[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Total duration of LV_MFF_ADD_LIM_MAX_LAM_AD[i] = 1 or LV_FAC_L_RNG_LIM_MAX_LAM_AD[i] = 1					
T_SUM_MAX_FSD_H_RNG[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Total duration of LV_FAC_H_RNG_LIM_MAX_LAM_AD[i] = 1					
T_SUM_MAX_FSD_LAM_LIM[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Total duration of LV_FAC_LAM_LIM_MAX[i] = 1					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
T_SUM_MIN_FSD[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Total duration of LV_MFF_ADD_LIM_MIN_LAM_AD[i] = 1 or LV_FAC_L_RNG_LIM_MIN_LAM_AD[i] = 1					
T_SUM_MIN_FSD_H_RNG[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Total duration of LV_FAC_H_RNG_LIM_MIN_LAM_AD[i] = 1					
T_SUM_MIN_FSD_LAM_LIM[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Total duration of LV_FAC_LAM_LIM_MIN[i] = 1					
T_SUM_RST_FSD[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Timer to reset the counter of total duration in MIL - area					
LV_CDN_DIAG_FSD[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Diagnosis conditions for lambda adaptation monitoring					
T_SUM_RST_FSD_H_RNG[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Timer to reset the counter of total duration in H_RNG - area					
T_SUM_RST_FSD_LAM_LIM[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Timer to reset the counter of total duration for lambda control in dead stop					
T_SUM_END_DIAG_WIN_FSD_FAC_L[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Counter for the diagnosis window - lower multiplicative adaptation learning area					
LV_END_DIAG_FSD[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
End of first diagnosis cycle in case of lambda adaptation monitoring					
ERR_SYM_FSD_LAM_LIM[NC_CBK_EX_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected failure of each symptom in case of lambda control in dead stop - relevant to MIL					
LV_CDN_DIAG_FSD_LAM_LIM[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Diagnosis conditions in the area where the lambda control output concerned - relevant to MIL					
LV_END_DIAG_FSD_LAM_LIM[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
End of first diagnosis cycle in the area where the lambda control output concerned					
ERR_SYM_FSD_H_RNG[NC_CBK_EX_NR]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Detected failure of each symptom in upper multiplicative adaptation learning area					

Input data:

LV_FAC_L_RNG_LIM_MAX_FSD[NC_CBK_EX_NR]	LV_FAC_L_RNG_LIM_MIN_FSD[NC_CBK_EX_NR]	LV_FSD_ACT[NC_CBK_EX_NR]	LV_INH_FSD_STOP_OIL
FAC_LAM_MV_MMV[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MAX[NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MIN[NC_CBK_EX_NR]	LV_FAC_H_RNG_LAM_AD[NC_CBK_EX_NR]
LV_FAC_H_RNG_LIM_MAX_LAM_AD[NC_CBK_EX_NR]	LV_FAC_H_RNG_LIM_MIN_LAM_AD[NC_CBK_EX_NR]	LV_FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	LV_MFF_ADD_LIM_MAX_FSD[NC_CBK_EX_NR]
LV_MFF_ADD_LIM_MIN_FSD[NC_CBK_EX_NR]	LV_MFF_ADD_RNG_LAM_AD[NC_CBK_EX_NR]	STATE_LAM_AD[NC_CBK_EX_NR]	NC_CBK_EX_NR
NC_IDX_DIAG_FSD[NC_CBK_EX_NR]	NC_IDX_DIAG_FSD_LAM_LIM[NC_CBK_EX_NR]	NC_IDX_DIAG_FSD_H_RNG[NC_CBK_EX_NR]	LV_STALL
TCO			

Calibration data:

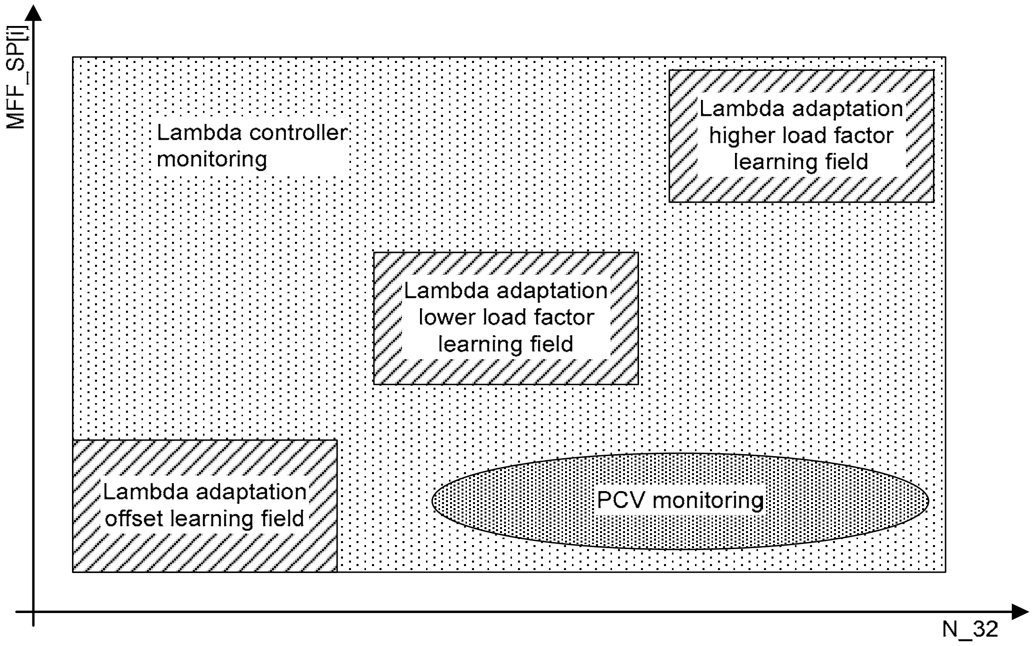
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_T_SUM_WIN_FSD_RST	1	0...FFH	0...0.99609375	0.00390625	-
Minimal fraction of counter threshold that must be spent in ADD as well as L_RNG for error healing					
C_CTR_DEC_LAM_LIM_LAM_AD_REQ	1	0...FFFFH	0...6.5535E+4	1	-
Counter decrement for activation of forced lambda adaptation					
C_CTR_INC_LAM_LIM_LAM_AD_REQ	1	0...FFFFH	0...6.5535E+4	1	-
Counter increment for activation of forced lambda adaptation					
C_CTR_MAX_LAM_LIM_LAM_AD_REQ	1	0...FFFFH	0...6.5535E+4	1	-
Counter threshold for activation of forced lambda adaptation					
C_T_LAM_AD_ACT_MIN_FSD	1	1...FFFFH	0.02...1.3107E+3	0.02	s
Minimum threshold of active forced lambda adaptation to reset the request flag					
C_T_SUM_MAX_THD_FSD	1	1...FFFFH	0.02...1.3107E+3	0.02	s
Maximum value of counter for ADD- , FAC_L- and FAC_H adaptation area (upper limit)					
C_T_SUM_MAX_THD_FSD_LAM_LIM	1	1...FFFFH	0.02...1.3107E+3	0.02	s
Maximum value of counter in case of lambda control - dead stop (upper limit)					
C_T_SUM_MIN_THD_FSD	1	1...FFFFH	0.02...1.3107E+3	0.02	s
Maximum value of counter for ADD- , FAC_L- and FAC_H adaptation area (lower limit)					
C_T_SUM_MIN_THD_FSD_LAM_LIM	1	1...FFFFH	0.02...1.3107E+3	0.02	s
Maximum value of counter in case of lambda control - dead stop (lower limit)					
C_T_SUM_RST_FSD	1	1...FFFFH	0.02...1.3107E+3	0.02	s
Calibration threshold to reset anti-bounce timers					
IP_FAC_LAM_OUT_MAX_FSD	6	0...FFFFH	-50...49.9984741	0.00152588	%
LDPM_TCO_4_LACO	6	0...FEH	-48...142.5	0.75	°C
The upper limit of lambda controller output to request forced lambda adaptation					
IP_FAC_LAM_OUT_MIN_FSD	6	0...FFFFH	-50...49.9984741	0.00152588	%
LDPM_TCO_4_LACO	6	0...FEH	-48...142.5	0.75	°C
The lower limit of lambda controller output to request forced lambda adaptation					

Import actions:

ACTION_ERRM_NoFilterReset(IN <IDX_DIAG>, OUT <LV_ERR>)
This action resets data filter in case of no filter usage
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure
ACTION_ERRM_NoFilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <LV_ERR_SET>, IN <LV_ERR_RST>, IN <LV_END_DIAG>, OUT <LV_ERR>)
This action returns the result on symptoms detected at each diagnostic recurrence, when no filter is used

46.17.1 General information

The objective of the fuel system diagnosis is to monitor the lambda control output and the lambda adaptation values in various areas. It should also cover the PCV monitoring where the lambda control output is considered in the idle range.



Monitoring area of fuel system diagnosis

Breaking the adaptation and lambda controller limits for a long time, which may have been caused by failures in the fuel or intake system will involve emission rise and therefore shall be diagnosed by fuel system diagnosis.

FSD errors are located in three failure memories with two and four symptoms each:

ERR_SYM_FSD[i]:

NO_SYM: no symptom
SYM_0: maximum limit of additive adaptation value reached
SYM_1: minimum limit of additive adaptation value reached
SYM_2: maximum limit of multiplicative adaptation value (lower area) reached
SYM_3: minimum limit of multiplicative adaptation value (lower area) reached

ERR_SYM_FSD_H_RNG[i]:

NO_SYM: no symptom
SYM_0: maximum limit of multiplicative adaptation value (upper area) reached
SYM_1: minimum limit of multiplicative adaptation value (upper area) reached

ERR_SYM_FSD_LAM_LIM[i]:

NO_SYM: no symptom
SYM_0: lambda control in dead stop (upper limit)
SYM_1: lambda control in dead stop (lower limit)

NC_CBK_EX_NR defines the number of exhaust banks.

For vector elements the variable extension “_i” is used in the model instead of “[i]” as found in the textual description.

Application Condition

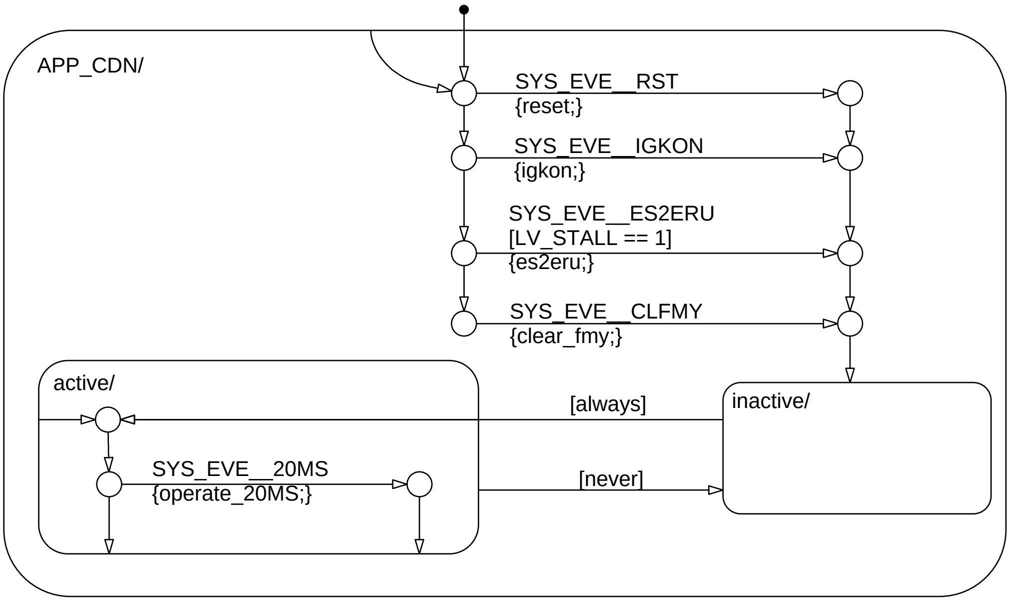
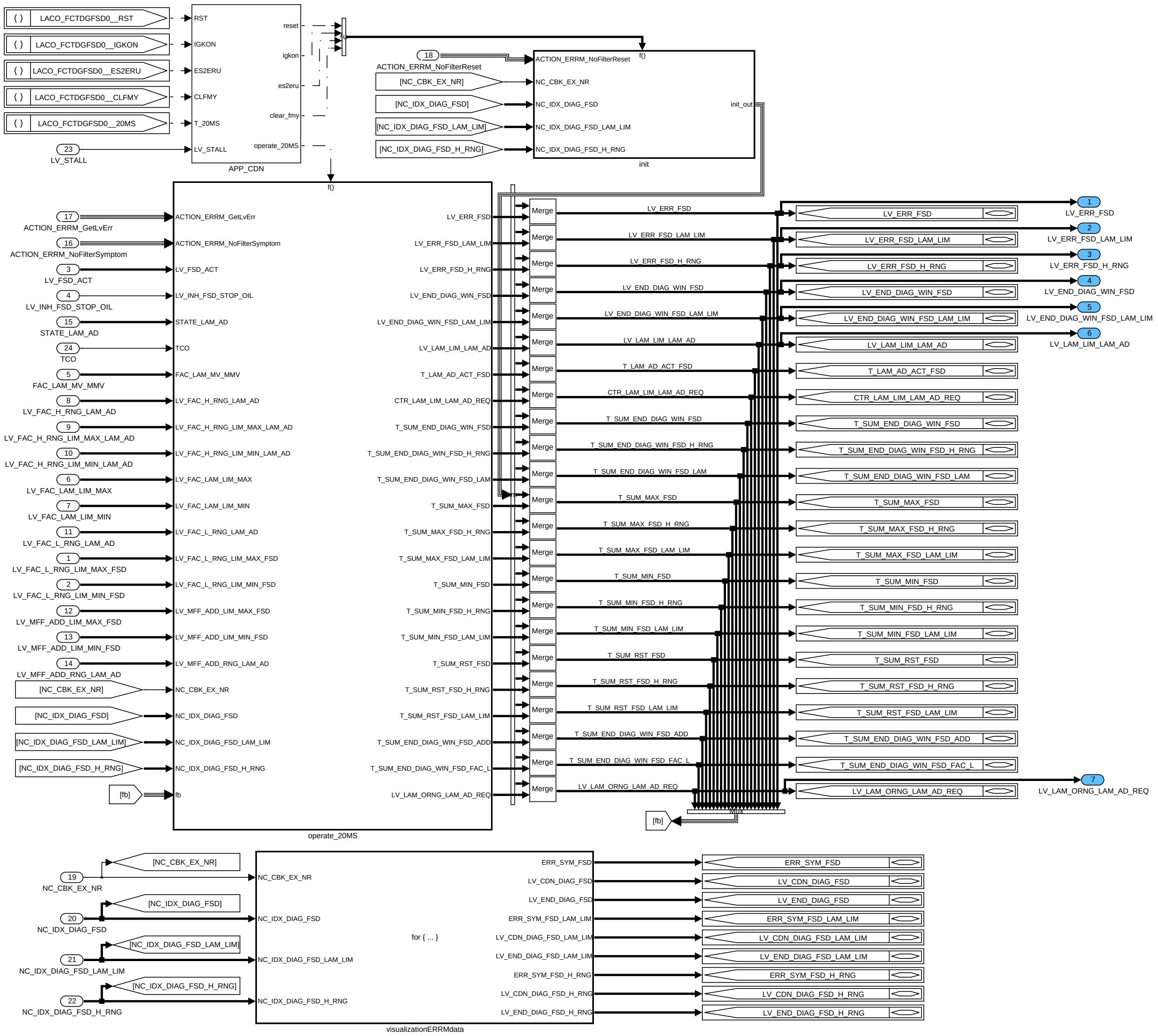


Figure 182 LACO_FCTDGFSD0/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 17−May−2007

Figure 183 LACO_FCTDGFSD0

46.17.1.1 Initialization

At ECU reset, IGKON, ES2ERU while LV_STALL is set (in order to cover engine stalling) and at clearance of failure memory all variables are initialized with 0.

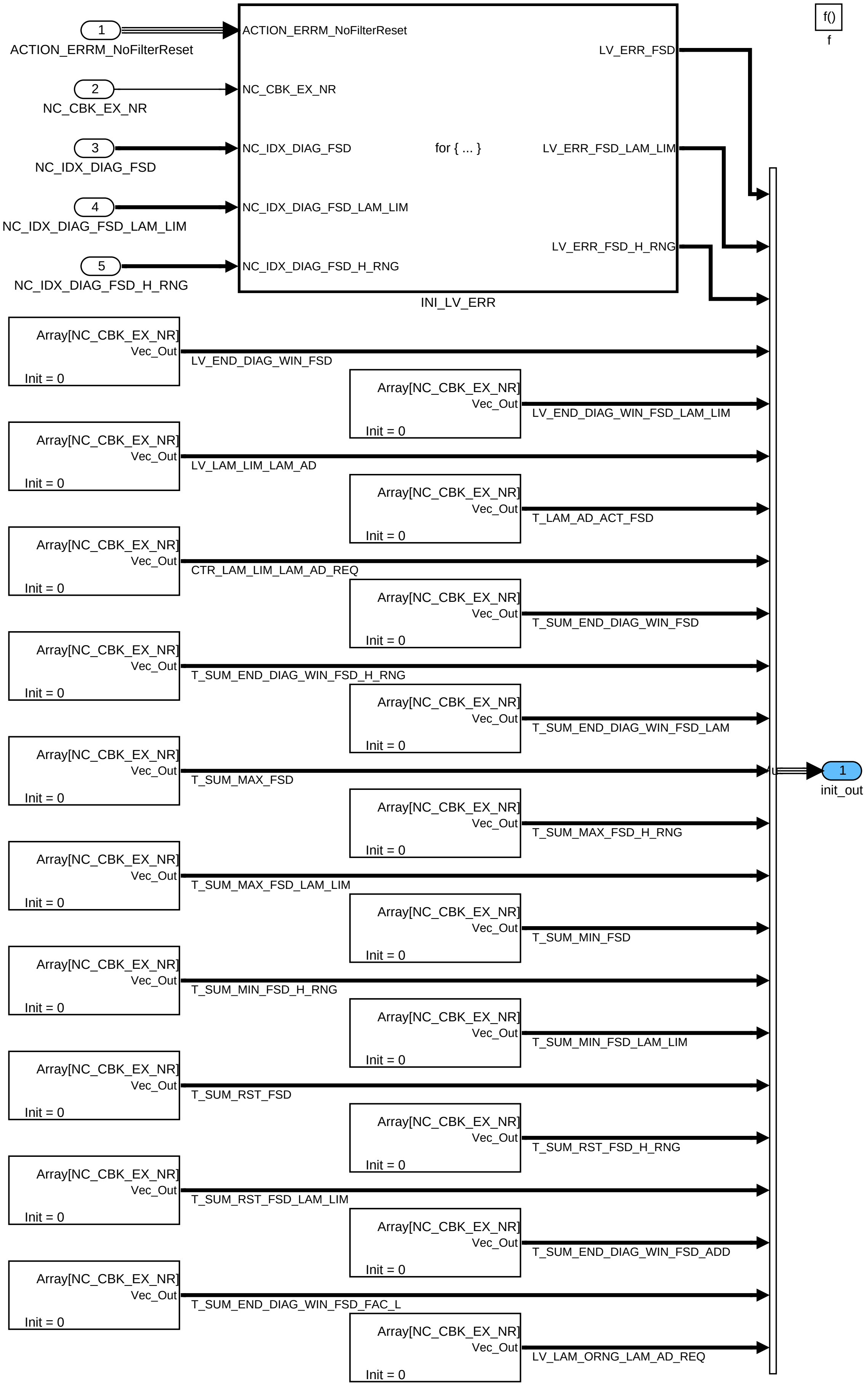

1
ACTION_ERRM_NoFilterReset
2
NC_CBK_EX_NR
3
NC_IDX_DIAG_FSD
4
NC_IDX_DIAG_FSD_LAM_LIM
5
NC_IDX_DIAG_FSD_H_RNG
ACTION_ERRM_NoFilterReset
NC_CBK_EX_NR
NC_IDX_DIAG_FSD
NC_IDX_DIAG_FSD_LAM_LIM
NC_IDX_DIAG_FSD_H_RNG
for { ... }
LV_ERR_FSD
LV_ERR_FSD_LAM_LIM
LV_ERR_FSD_H_RNG
INI_LV_ERR
f()
f
Array[NC_CBK_EX_NR]
Vec_Out
Init = 0
LV_END_DIAG_WIN_FSD
LV_END_DIAG_WIN_FSD_LAM_LIM
LV_LAM_LIM_LAM_AD
T_LAM_AD_ACT_FSD
CTR_LAM_LIM_LAM_AD_REQ
T_SUM_END_DIAG_WIN_FSD
T_SUM_END_DIAG_WIN_FSD_H_RNG
T_SUM_END_DIAG_WIN_FSD_LAM
T_SUM_MAX_FSD
T_SUM_MAX_FSD_H_RNG
T_SUM_MAX_FSD_LAM_LIM
T_SUM_MIN_FSD
T_SUM_MIN_FSD_H_RNG
T_SUM_MIN_FSD_LAM_LIM
T_SUM_RST_FSD
T_SUM_RST_FSD_H_RNG
T_SUM_RST_FSD_LAM_LIM
T_SUM_END_DIAG_WIN_FSD_ADD
T_SUM_END_DIAG_WIN_FSD_FAC_L
LV_LAM_ORNG_LAM_AD_REQ
1
init_out

Figure 184 LACO_FCTDGFSD0/ init

Initialization of error flags

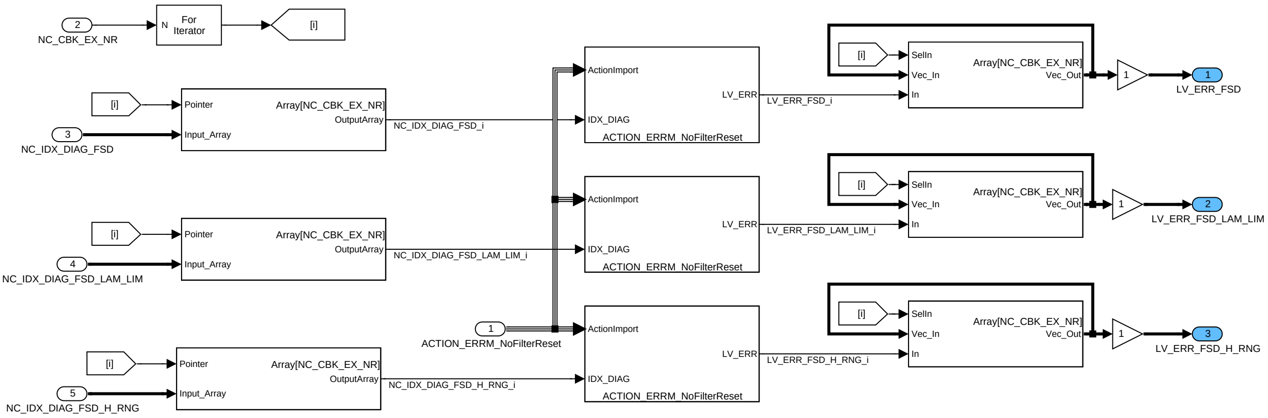


Figure 185 LACO_FCTDGFSD0/ init/ INI_LV_ERR

46.17.1.2 Formula section

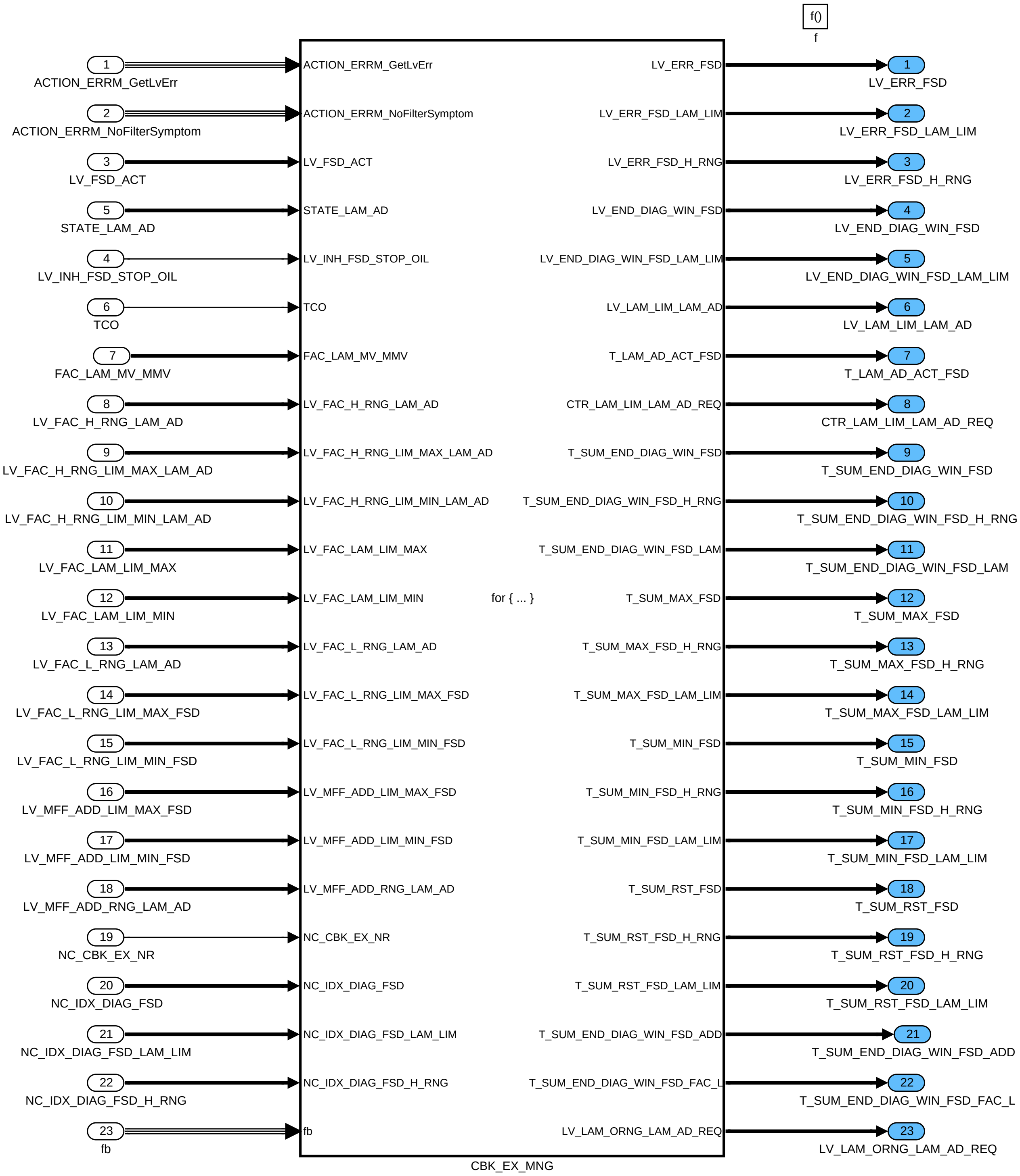


Figure 186 LACO_FCTDGFSD0/ operate_20MS

Exhaust bank specific functionality

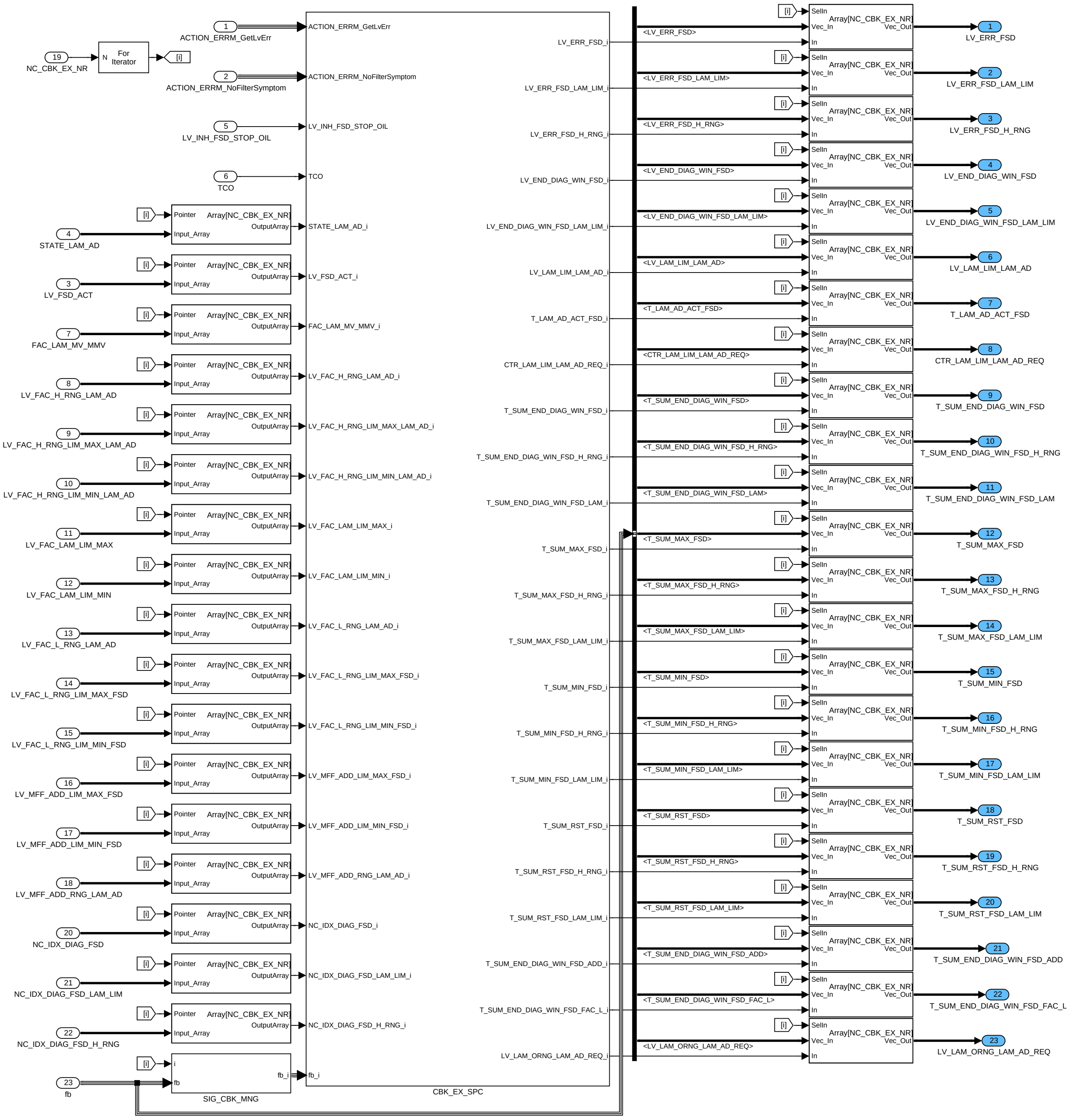


Figure 187 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG

Overview about the diagnostics

The diagnosis function is split into 4 blocks.

1. By means of an interface functionality to the lambda adaptation the forced lambda adaptation can be requested by the FSD. This functionality runs independent of the rest.

2. The diagnosis conditions for all failure location are evaluated separately from each other.

3. The main block is the evaluation of the error symptom and the management of the end of diagnosis information. The decision about present errors is also part of this block.

4. The interface to the error management ERRM is handled in this block. My means of ACTION calls the relevant information is transmitted to the ERRM.

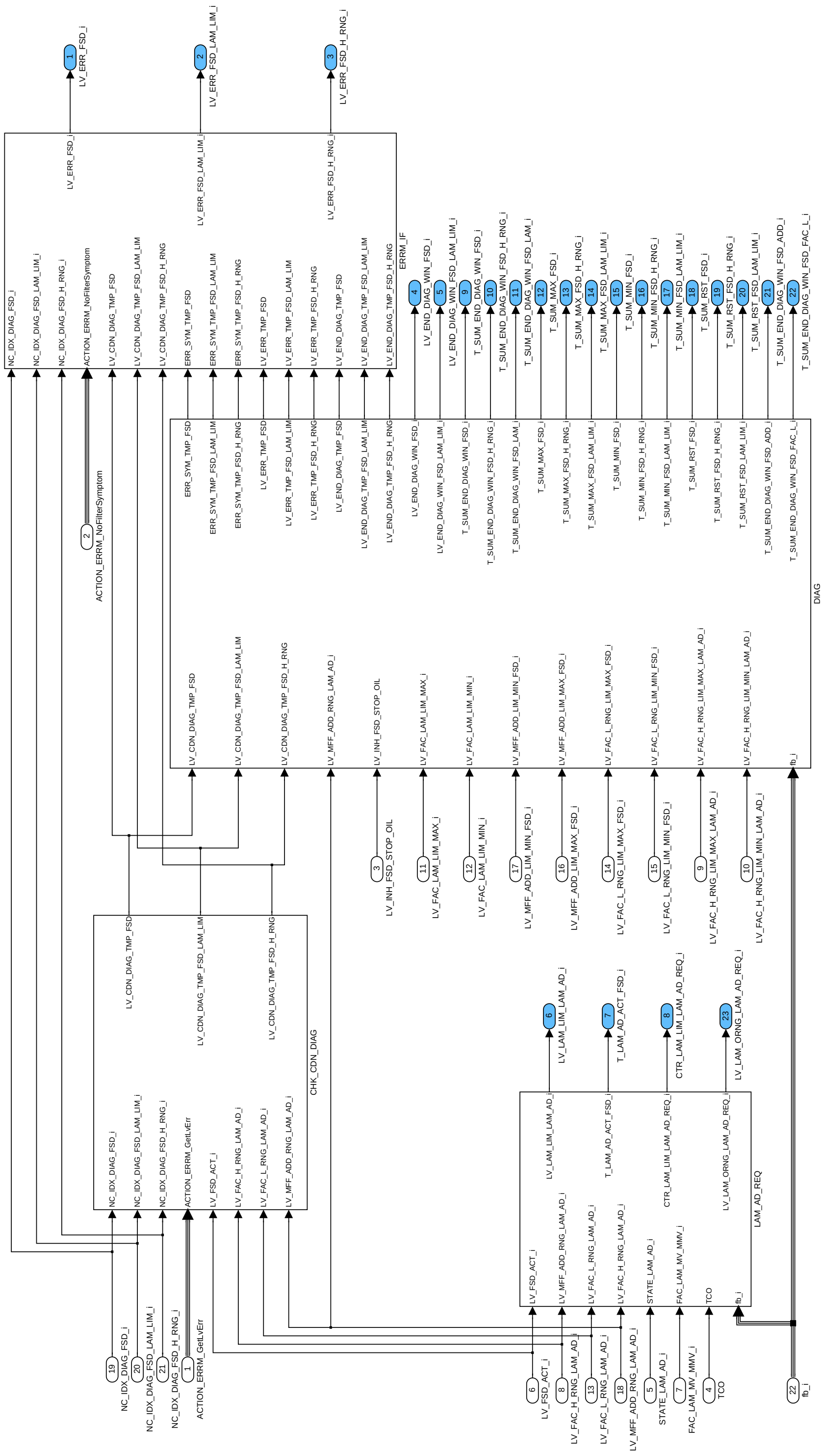


Figure 188 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC

Interface to lambda adaptation

When the low-pass filtered lambda controller output exceeds a minimum or maximum calibration threshold the lambda adaptation shall be forced by the FSD to get active and adapt the lambda controller deviation in order to confirm to lambda excursion. The request for this forced lambda adaptation is controlled by an anti-bounce mechanism.

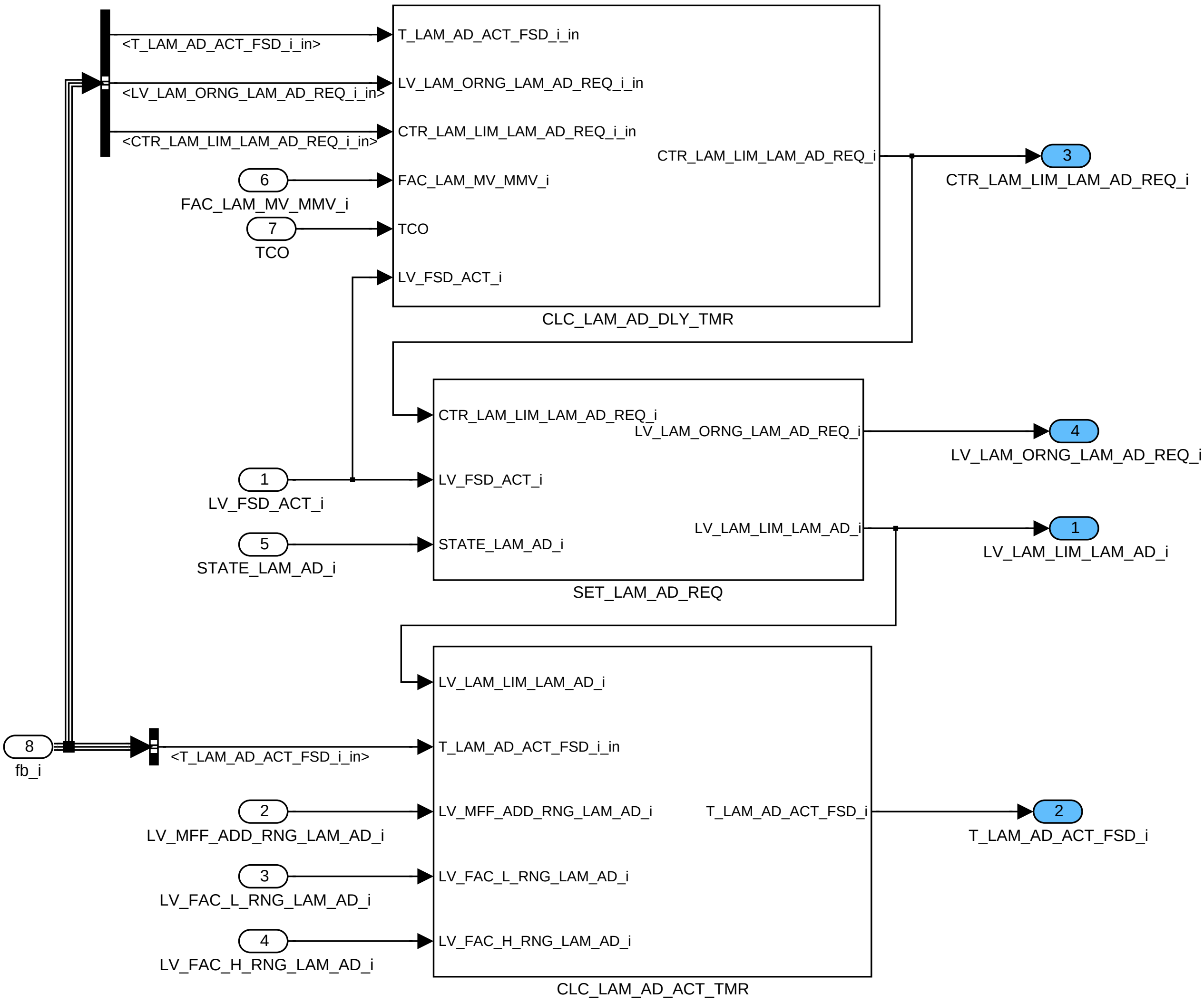


Figure 189 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ LAM_AD_REQ

Request counter

The request counter CTR_LAM_LIM_LAM_AD_REQ[i] is reset to 0 as soon as the activation timer T_LAM_AD_ACT_FSD[i] expires.

As long as

- the request counter is not yet expired (indicated by the flag LV_LAM_ORNG_LAM_AD_REQ[i]),
- fuel system diagnosis is active (LV_FSD_ACT[i] = 1) and
- lambda controller output is not zero

the counter is either incremented or decremented depending on whether lambda controller output is inside or outside the permitted range. The incrementation / decrementation speed is tunable.

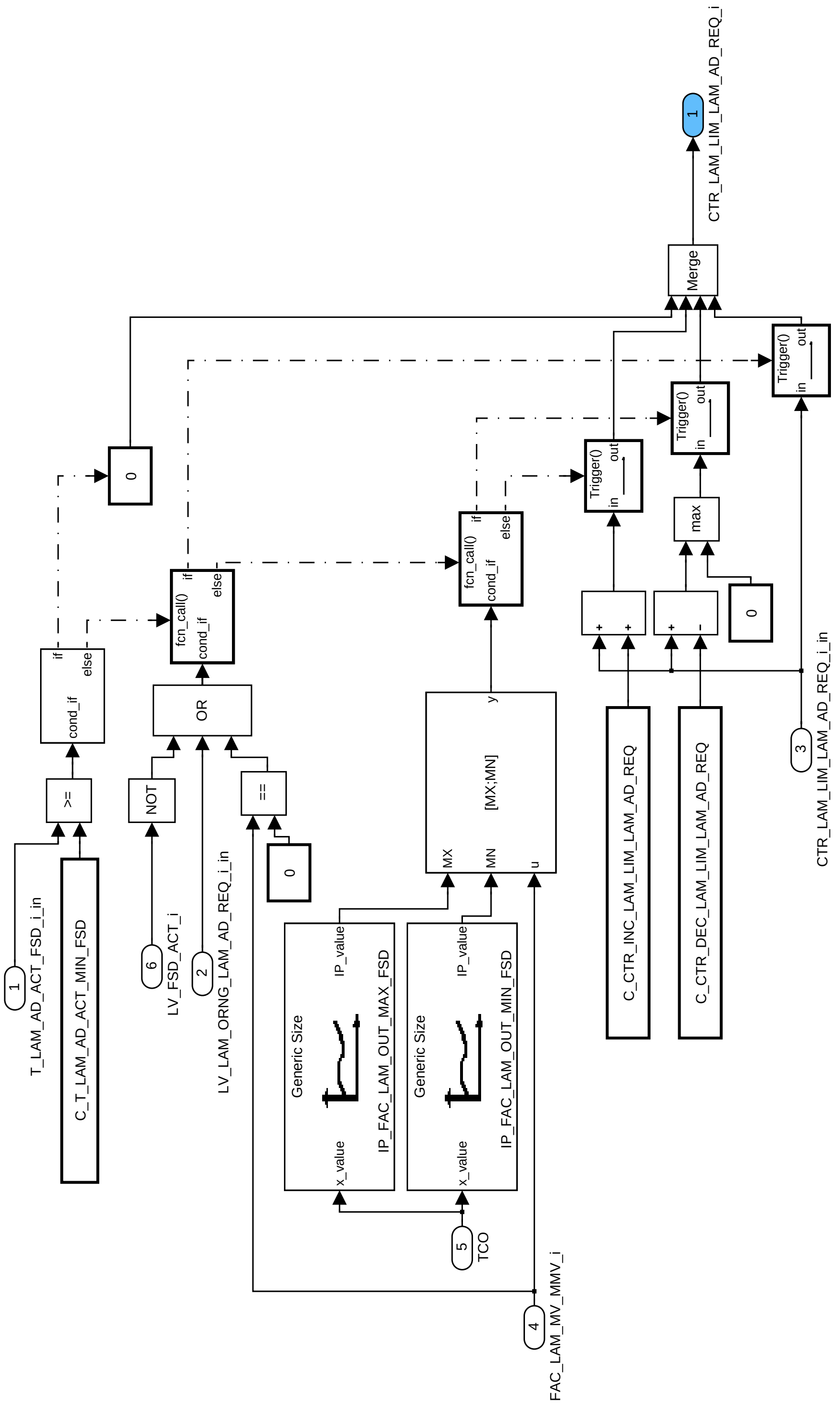


Figure 190 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ LAM_AD_REQ/ CLC_LAM_AD_DLY_TMR

Request flag / Lamda controller out of range flag

If the request counter is expired, fuel system diagnosis is active and the lambda adaptation is inside one of its learning fields (either state CDN_FAC_L, CDN_FAC_H, CDN_ADD, ADAPT_FAC_L, ADAPT_FAC_H or ADAPT_ADD indicated by STATE_LAM_AD[i] >= CDN_FAC_L), the flag LV_LAM_LIM_LAM_AD[i] that requests the forced lambda adaptation is set to 1.

As soon as request counter reaches the calibration threshold the flag LV_LAM_ORNG_LAM_AD_REQ[i] indicating that the request timer expired is set.

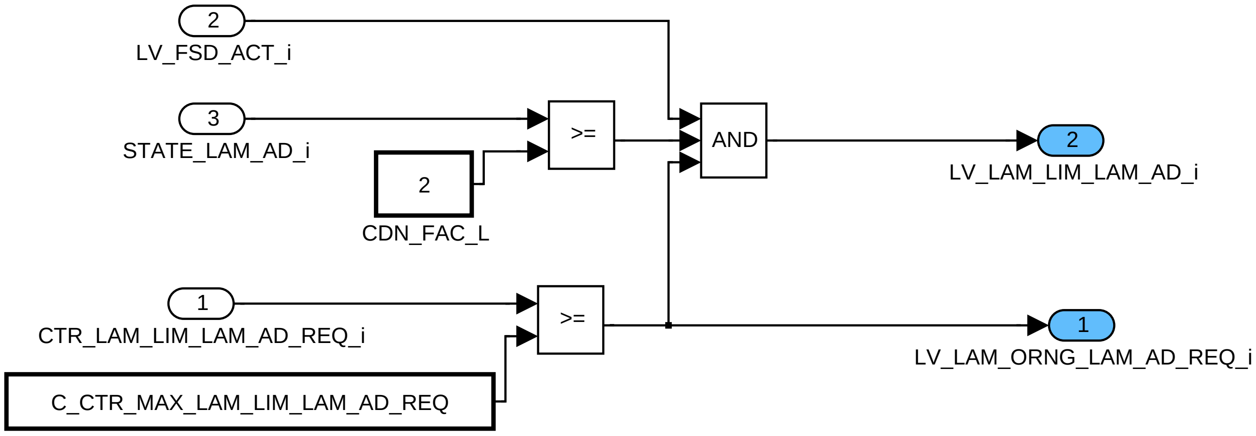


Figure 191 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ LAM_AD_REQ/ SET_LAM_AD_REQ

Activation timer

The activation timer T_LAM_AD_ACT_FSD[i] is incremented as long as the request flag for forced lambda adaptation is set and lambda adaptation is active. As soon as the activation timer expires it is reset to 0.

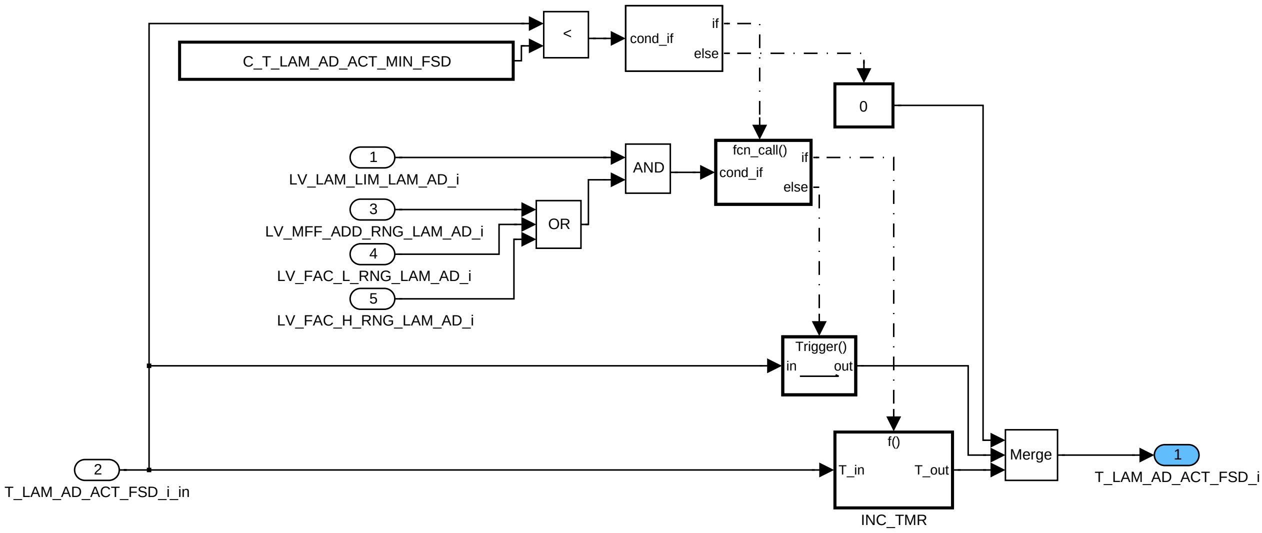


Figure 192 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ LAM_AD_REQ/ CLC_LAM_AD_ACT_TMR

Increment timer

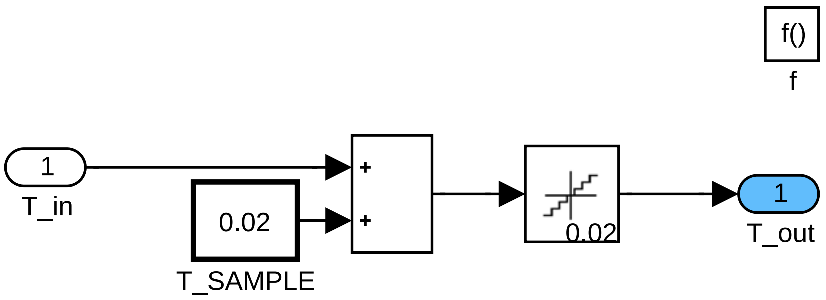


Figure 193 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ LAM_AD_REQ/ CLC_LAM_AD_ACT_TMR/ INC_TMR

Check diagnostic conditions

For each failure location the diagnostic condition is evaluated separately.

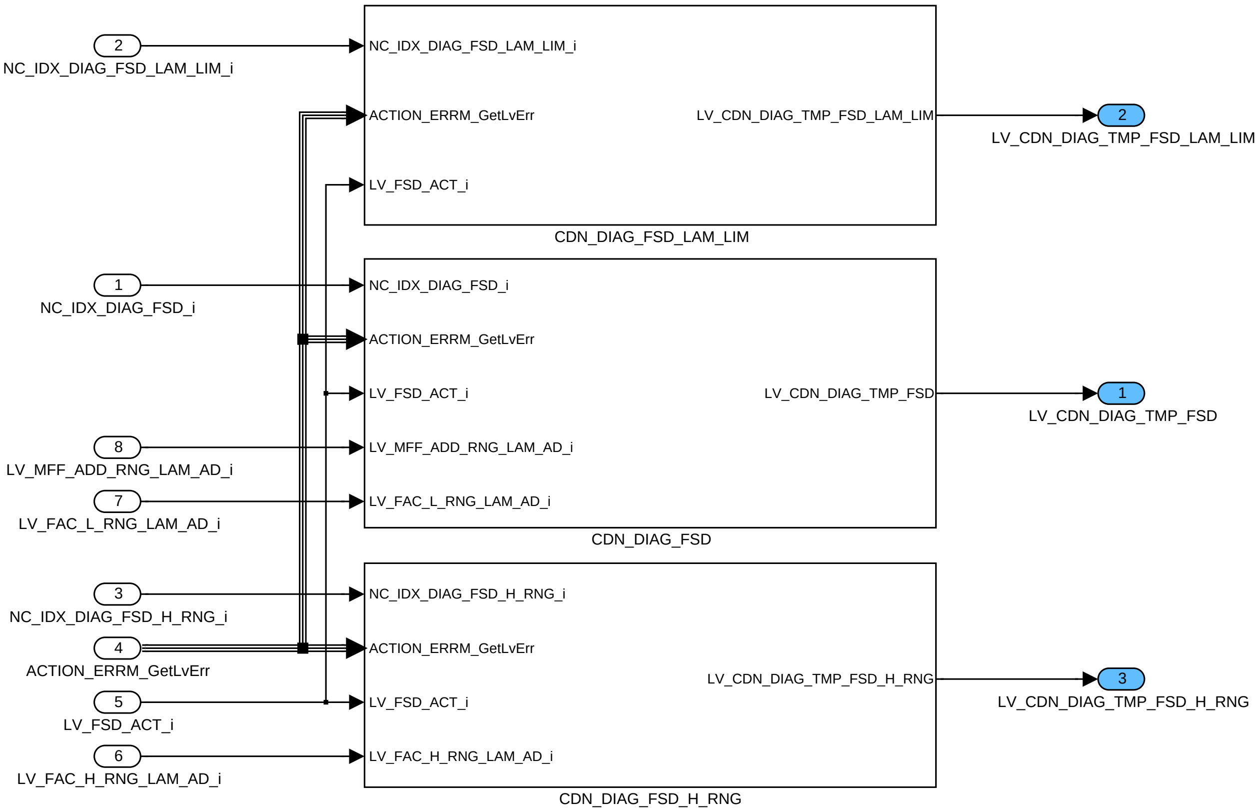


Figure 194 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ CHK_CDN_DIAG

Check conditions for lambda adaptation monitoring

Conditions are fulfilled when no error is present, the global activation conditions are fulfilled and the lambda adaptation is either in the offset or lower load factor learning field.

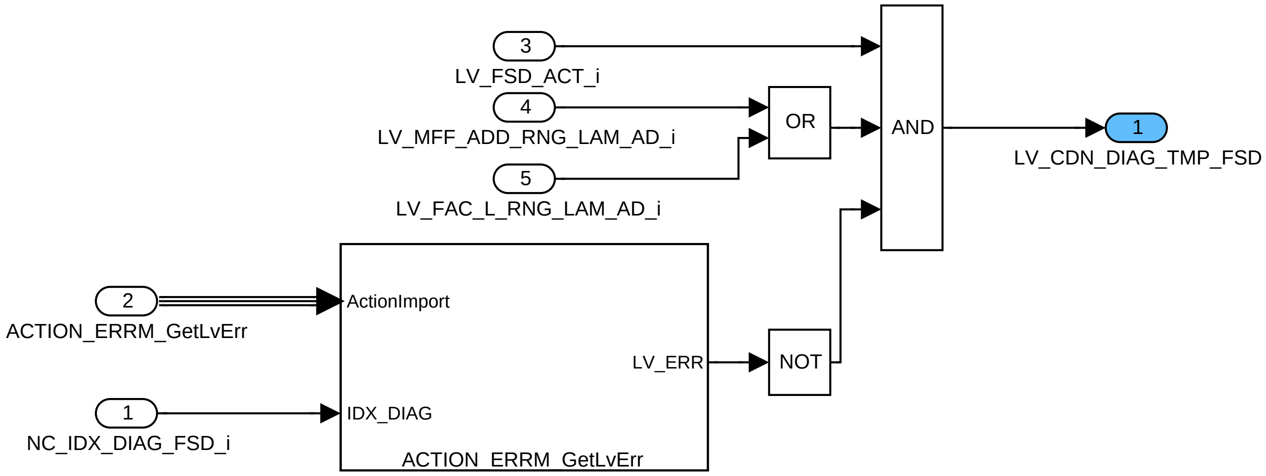


Figure 195 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ CHK_CDN_DIAG/ CDN_DIAG_FSD

Check conditions for lambda controller monitoring

Conditions are fulfilled when no error is present and the global activation conditions are fulfilled. The lambda controller monitoring is executed in the whole operating range.

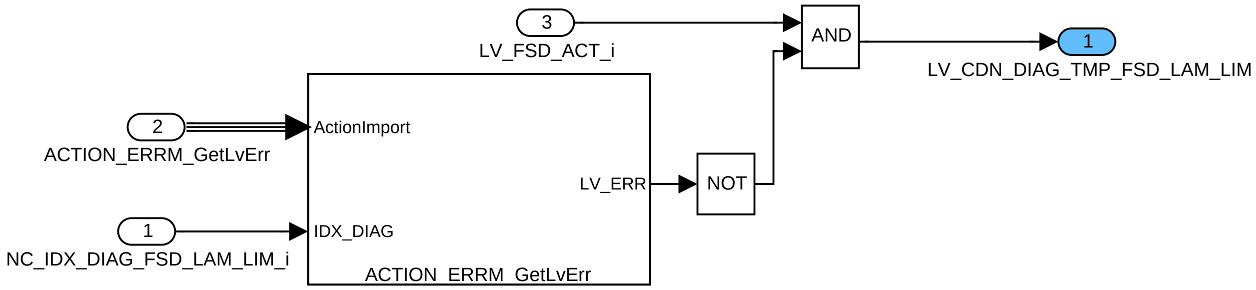


Figure 196 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ CHK_CDN_DIAG/ CDN_DIAG_FSD_LAM_LIM

Check conditions for lambda adaptation high load monitoring

Conditions are fulfilled when no error is present, the global activation conditions are fulfilled and the lambda adaptation is in the higher load factor learning field.

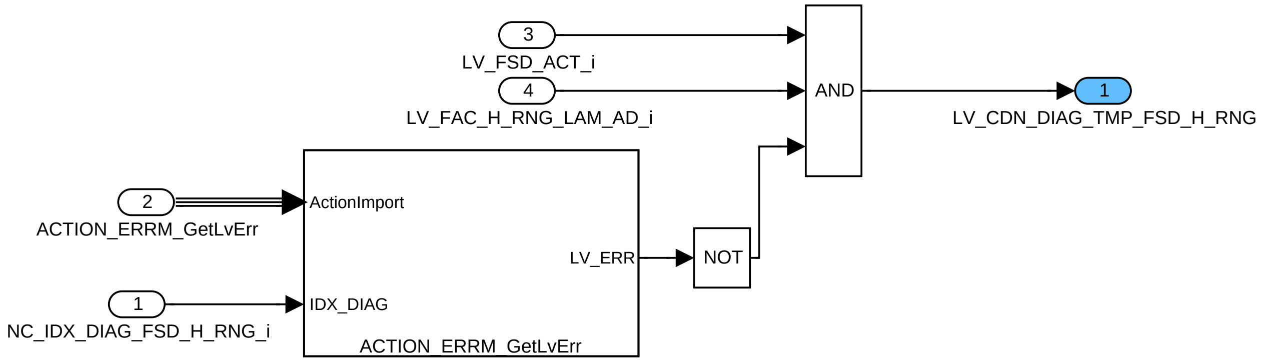


Figure 197 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ CHK_CDN_DIAG/ CDN_DIAG_FSD_H_RNG

Error detection and end of diagnosis management

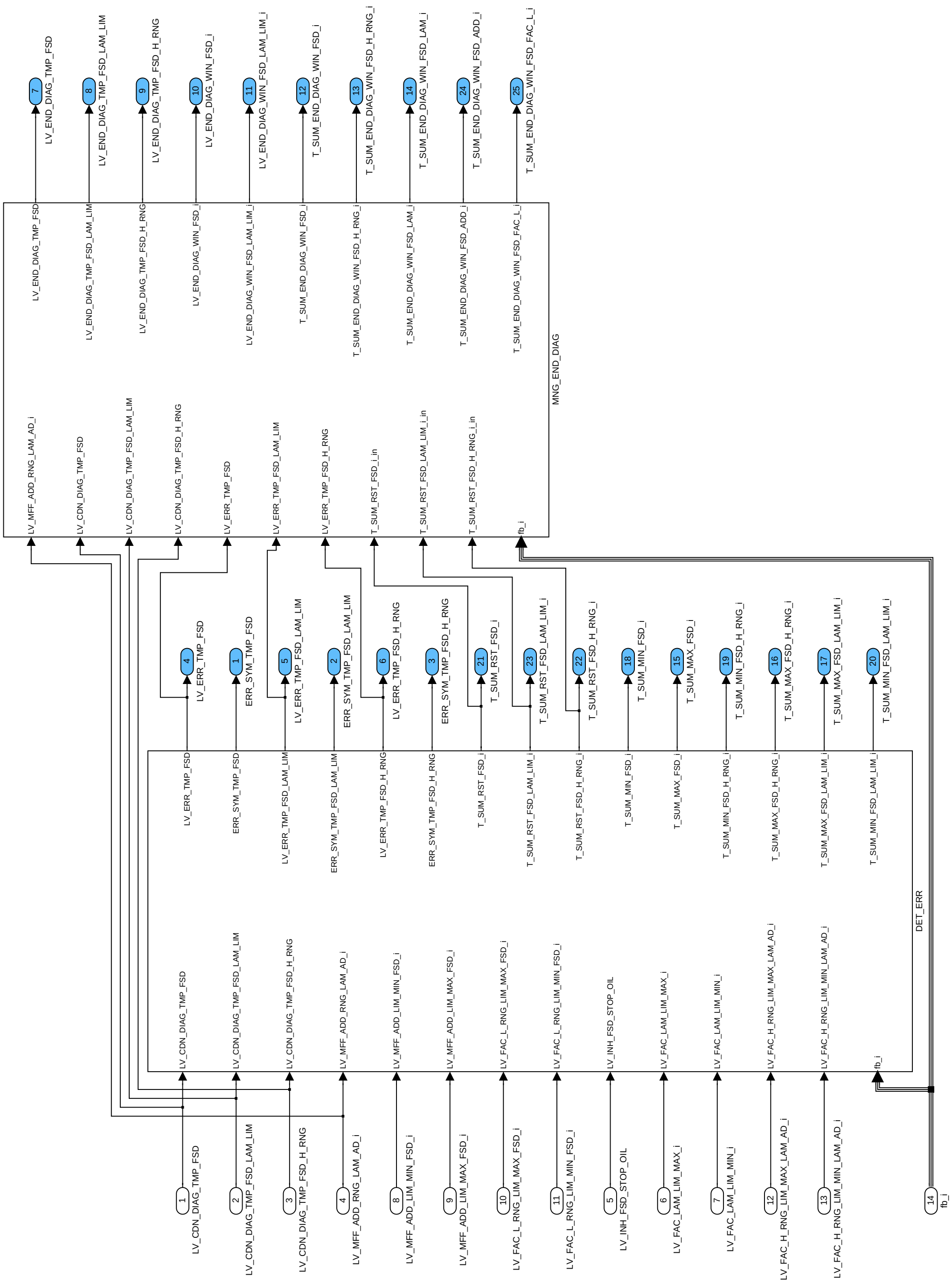


Figure 198 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG

Error symptom detection

For each failure location the symptom detection is executed separately.

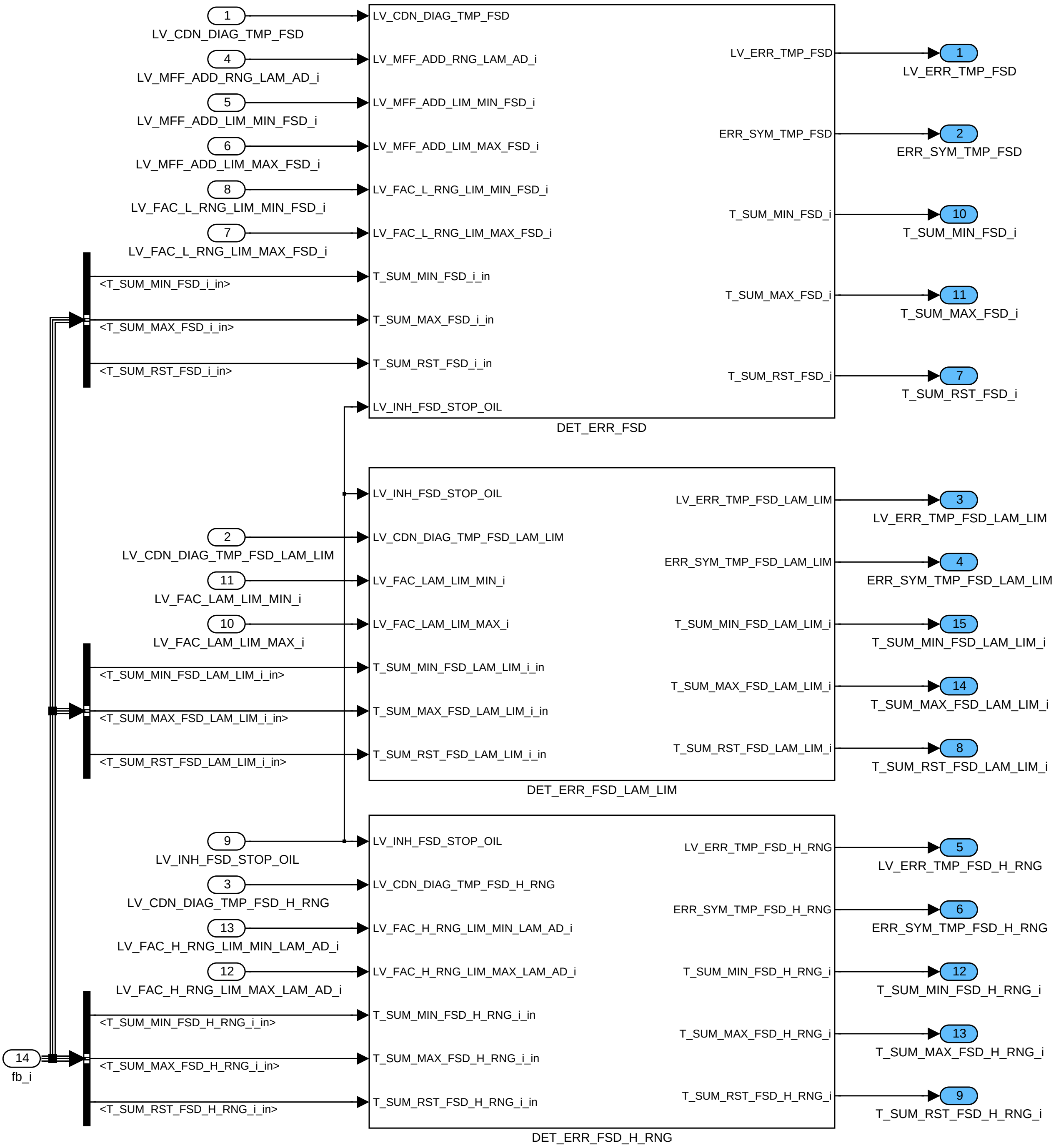


Figure 199 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR

Error symptom detection for lambda adaptation monitoring

In case the conditions are not fulfilled all timers remain unchanged and the temporary error flag and symptom have no relevance.

Depending on whether lambda adaptation is in the offset or lower load factor learning field different but similar functionalities are executed.

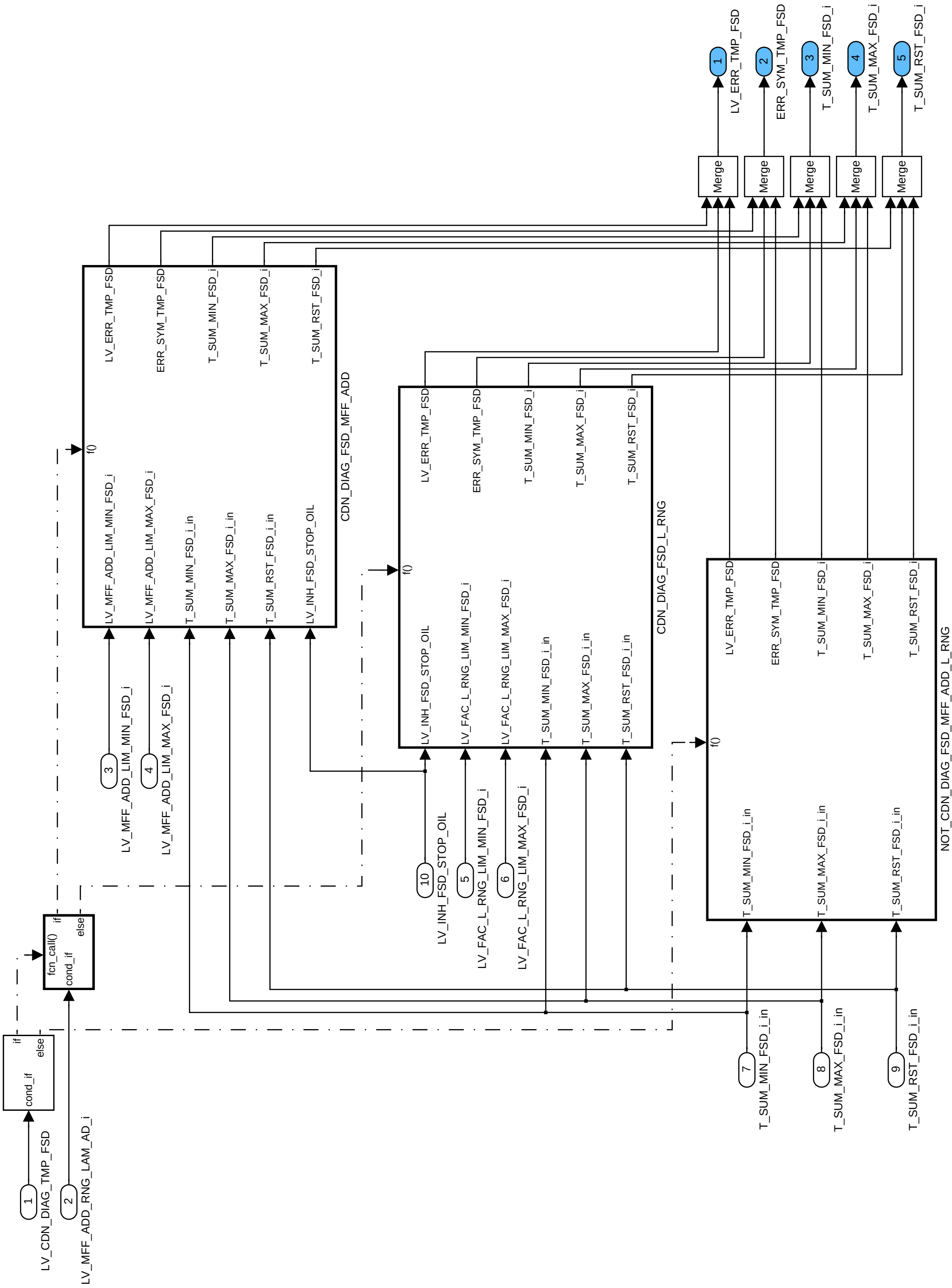


Figure 200 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD

Conditions for lambda adaptation offset monitoring fulfilled

Between 3 different cases must be distinguished:

1. Lambda adaptation factor for lower load is at its minimum threshold (indicated by the flag LV_ MFF_ADD _LIM_MIN_LAM_AD[i]) and no inhibition due to oil dilution under cold start conditions.

2. Lambda adaptation factor for lower load is at its maximum threshold (indicated by the flag LV_MFF_ADD_LIM_MAX_LAM_AD[i]).

3. No limitation of the lambda adaptation factor for lower load.

Independent of these cases the anti-bounce timers used to finally set the error are evaluated afterwards.

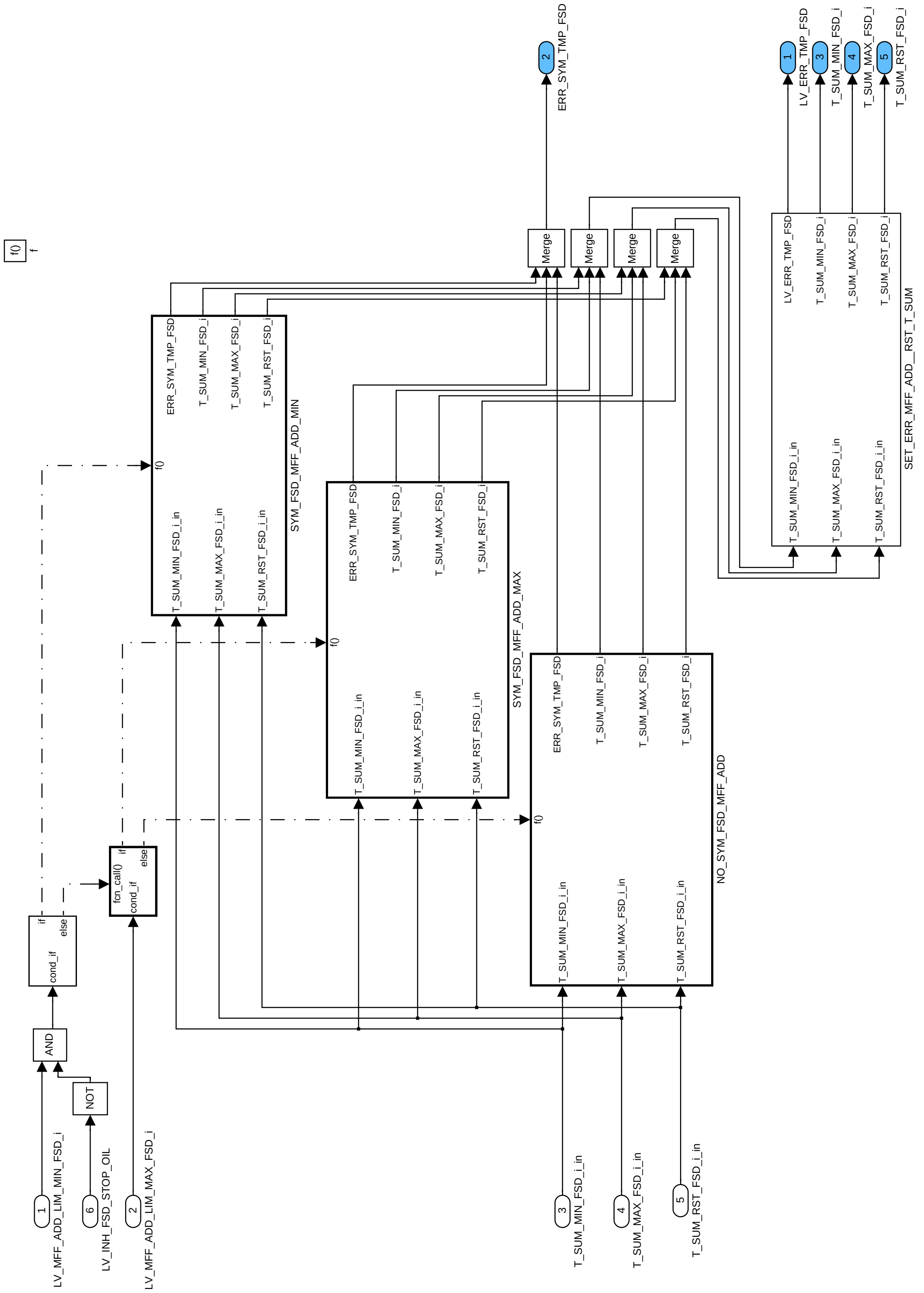


Figure 201 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD

Set error symptom for lambda adaptation offset at the minimum

The error symptom is set to the respective value (SYM_1). The timer indicating that the minimum threshold is reached and the reset timer are both increased.

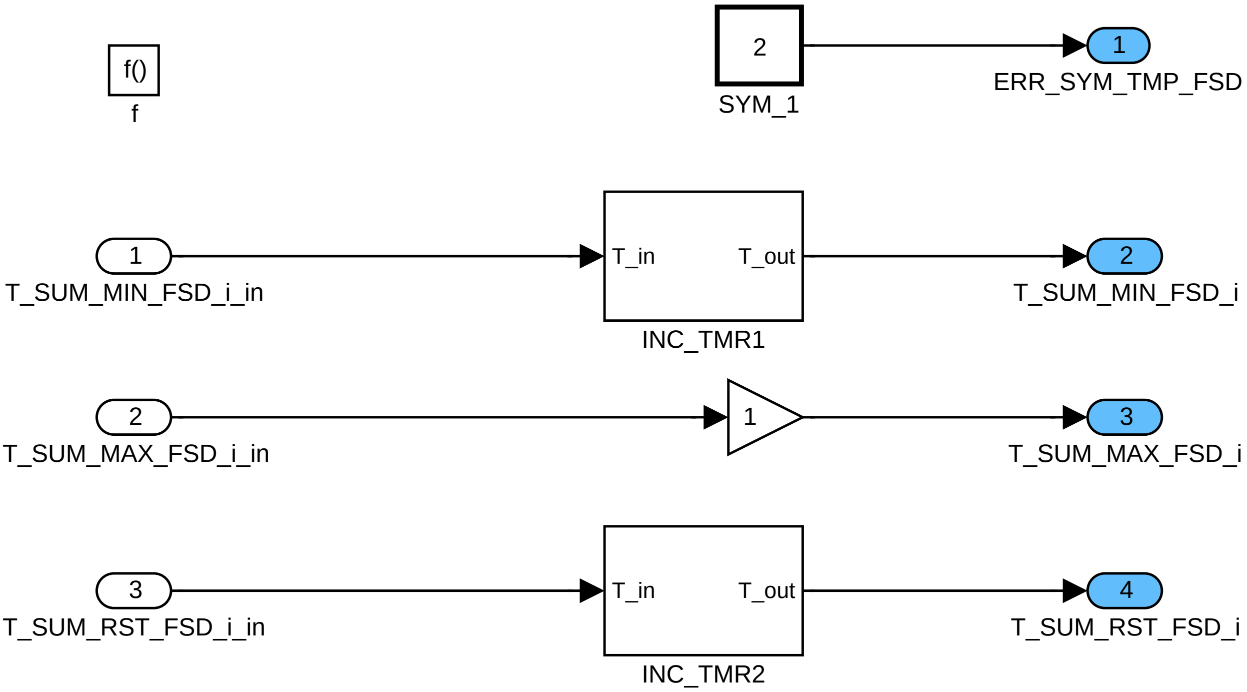


Figure 202 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ SYM_FSD_MFF_ADD_MIN

Increment timer

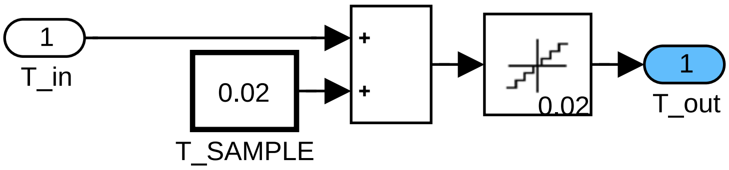


Figure 203 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ SYM_FSD_MFF_ADD_MIN/ INC_TMR1

Increment timer

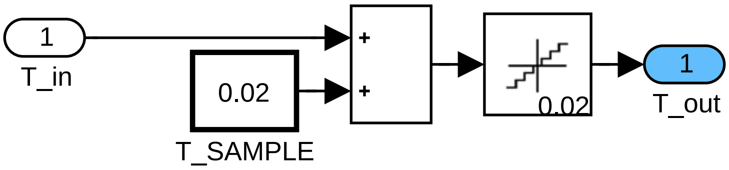


Figure 204 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ SYM_FSD_MFF_ADD_MIN/ INC_TMR2

Set error symptom for lambda adaptation offset at the maximum

The error symptom is set to the respective value (SYM_0). The timer indicating that the maximum threshold is reached and the reset timer are both increased.

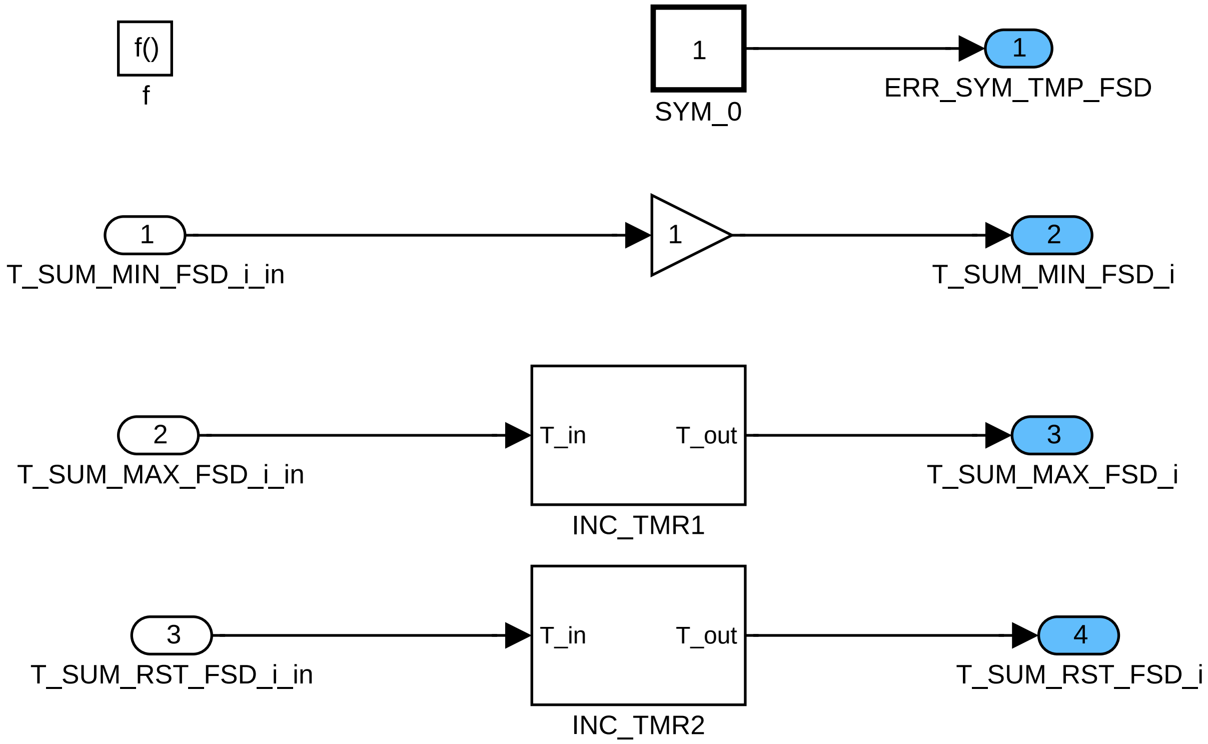


Figure 205 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ SYM_FSD_MFF_ADD_MAX

Increment timer

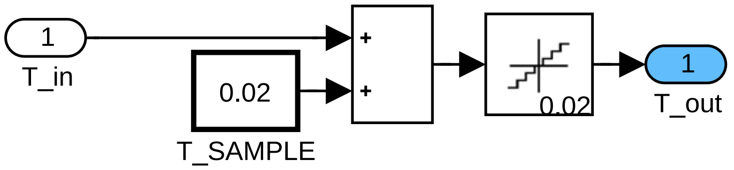


Figure 206 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ SYM_FSD_MFF_ADD_MAX/ INC_TMR1

Increment timer

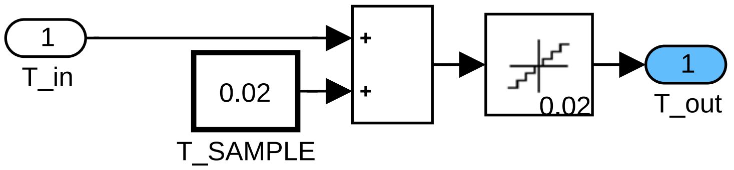


Figure 207 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ SYM_FSD_MFF_ADD_MAX/ INC_TMR2

No error symptom for lambda adaptation offset

The error symptom is set to NO_SYM and the reset timer is increased in case it is already different from 0. In that case the error symptom was at least one time different from NO_SYM.

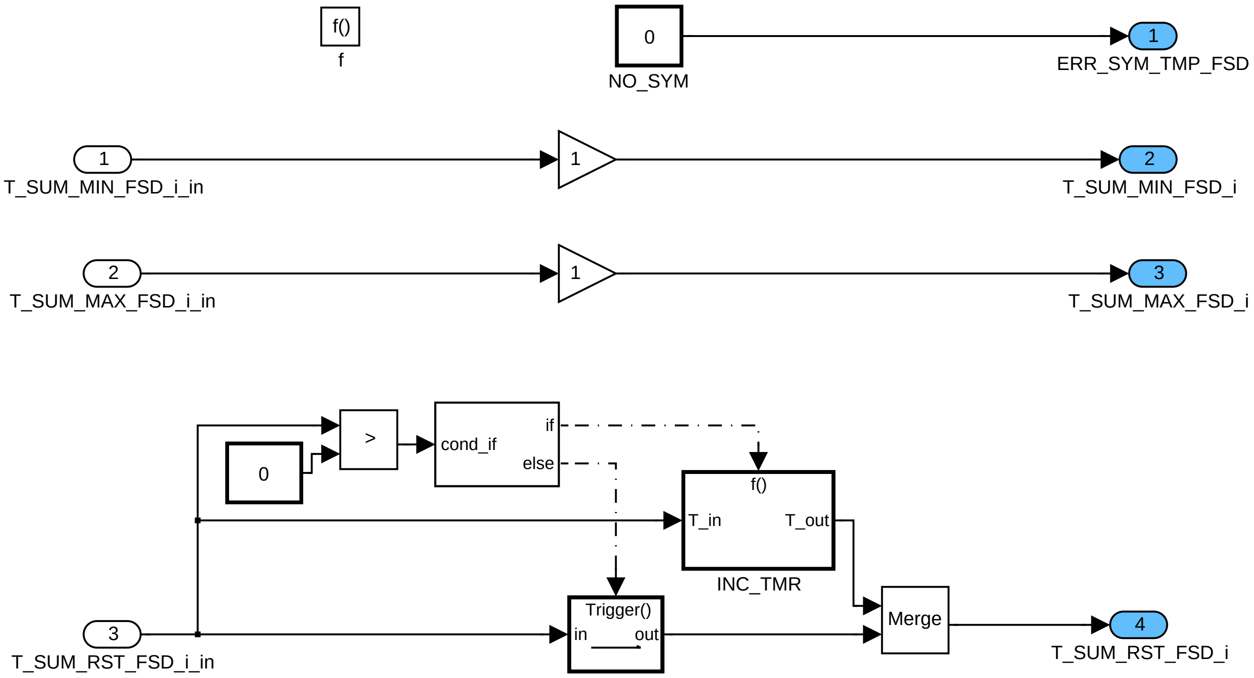


Figure 208 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ NO_SYM_FSD_MFF_ADD

Increment timer

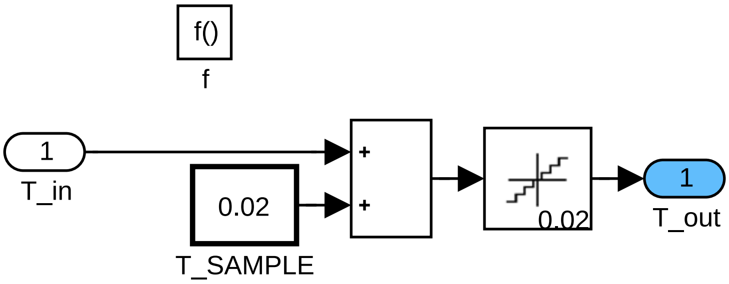


Figure 209 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ NO_SYM_FSD_MFF_ADD/ INC_TMR

Set error flag for lambda adaptation offset monitoring

The temporary error flag is set if the lambda adaptation factor was either to long at its maximum or at its minimum.

When the reset timer reaches the threshold C_T_SUM_RST_FSD all timers are reset to 0.

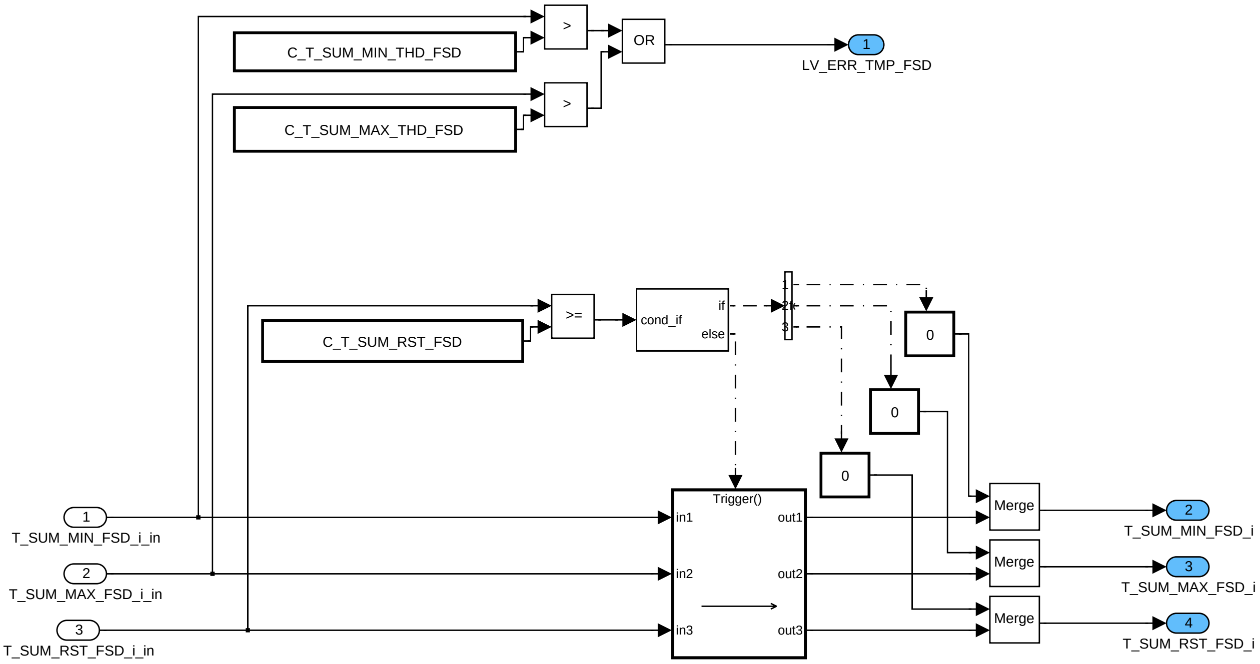


Figure 210 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_MFF_ADD/ SET_ERR_MFF_ADD__RST_T_SUM

Conditions for lambda adaptation low range monitoring fulfilled

Between 3 different cases must be distinguished:

1. Lambda adaptation factor for lower load is at its minimum threshold (indicated by the flag LV_FAC_L_RNG_LIM_MIN_FSD[i] which is assigned in the application incidences) and no inhibition due to oil dilution under cold start conditions.

2. Lambda adaptation factor for lower load is at its maximum threshold (indicated by the flag LV_FAC_L_RNG_LIM_MAX_FSD[i] which is assigned in the application incidences).

3. No limitation of the lambda adaptation factor for lower load.

Independent of these cases the anti-bounce timers used to finally set the error are evaluated afterwards.

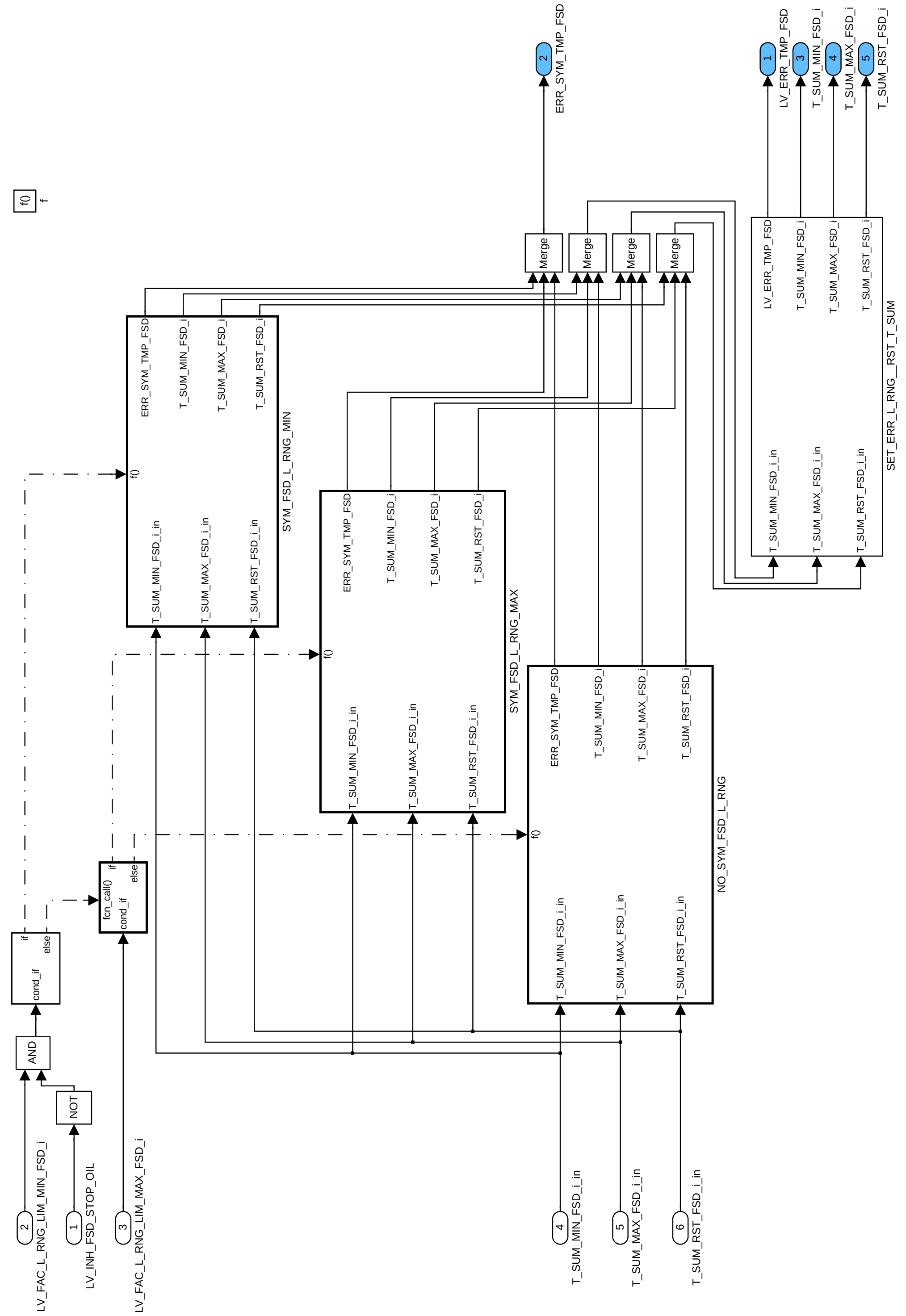


Figure 211 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG

<u>Set error symptom for lambda adaptation low range at the minimum</u>

The error symptom is set to the respective value (SYM_3). The timer indicating that the minimum threshold is reached and the reset timer are both increased.

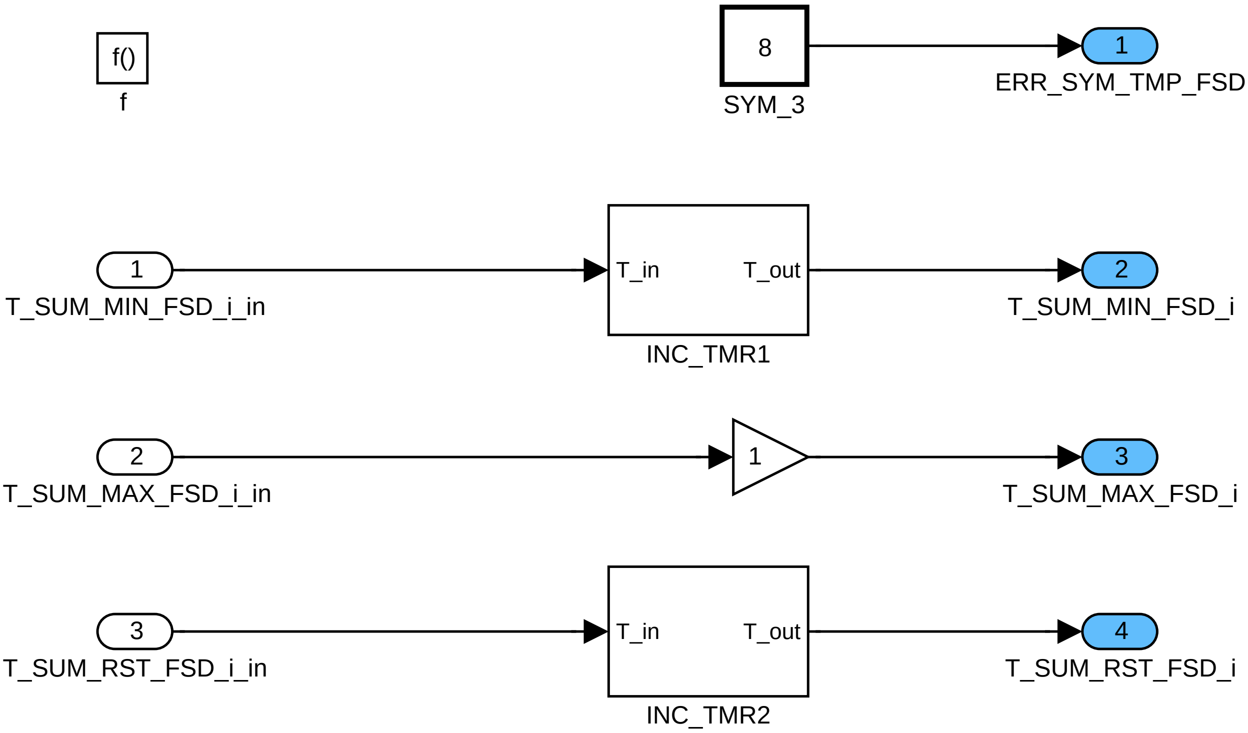


Figure 212 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ SYM_FSD_L_RNG_MIN

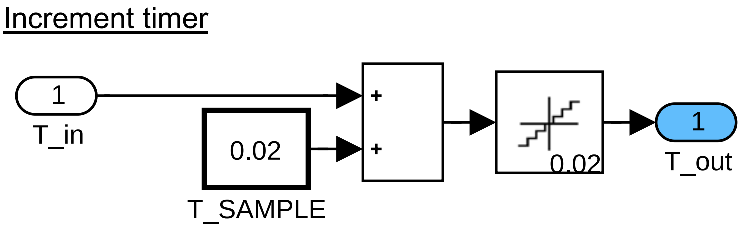


Figure 213 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ SYM_FSD_L_RNG_MIN/ INC_TMR1

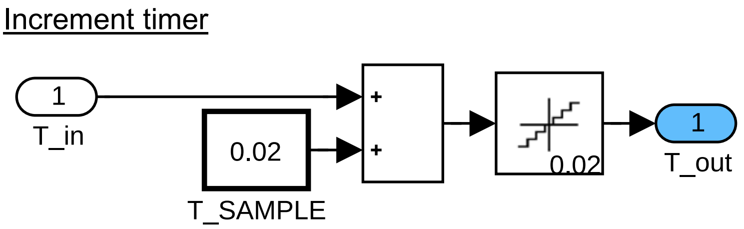


Figure 214 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ SYM_FSD_L_RNG_MIN/ INC_TMR2

Set error symptom for lambda adaptation low range at the maximum

The error symptom is set to the respective value (SYM_2). The timer indicating that the maximum threshold is reached and the reset timer are both increased.

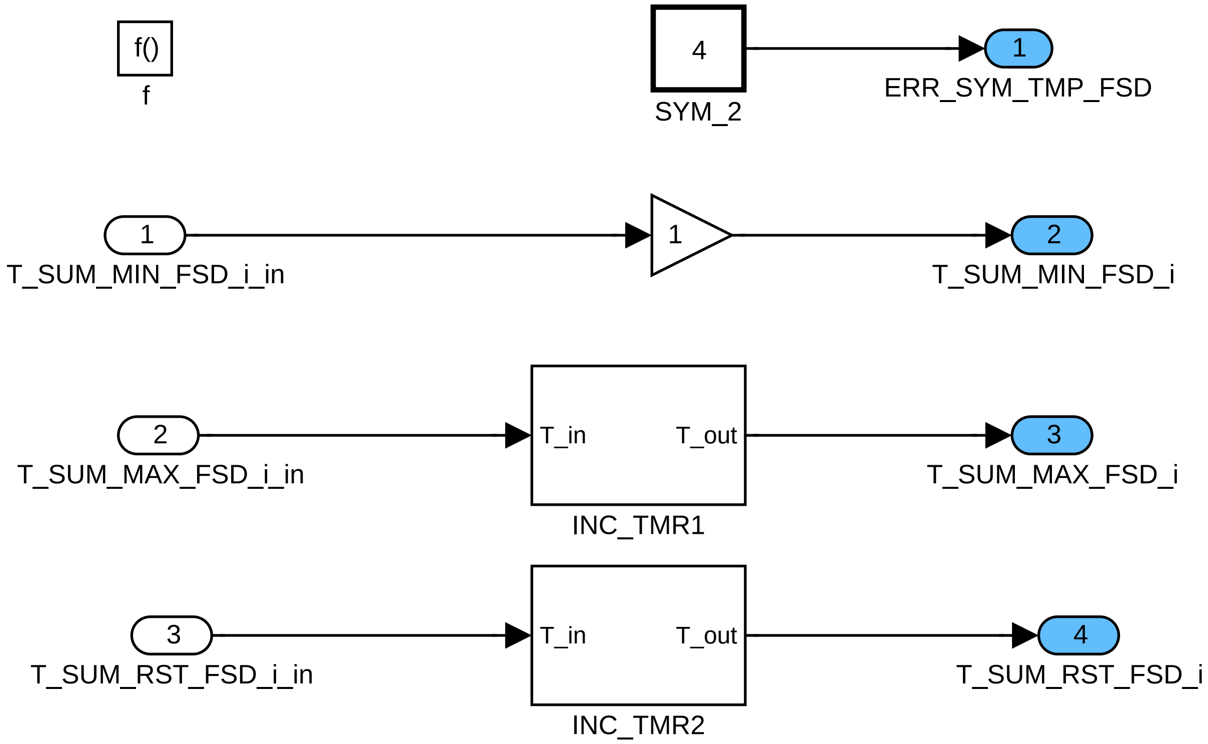


Figure 215 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ SYM_FSD_L_RNG_MAX

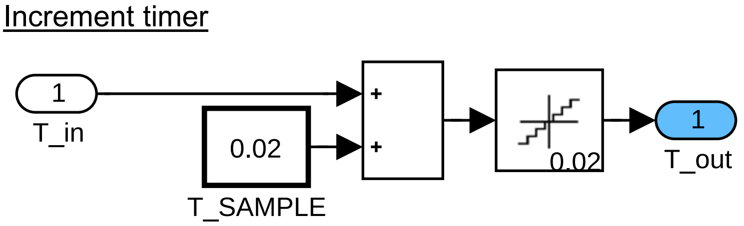


Figure 216 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ SYM_FSD_L_RNG_MAX/ INC_TMR1

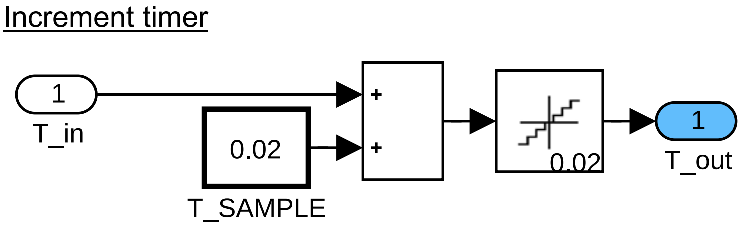


Figure 217 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ SYM_FSD_L_RNG_MAX/ INC_TMR2

No error symptom for lambda adaptation low range

The error symptom is set to NO_SYM and the reset timer is increased in case it is already different from 0. In that case the error symptom was at least one time different from NO_SYM.

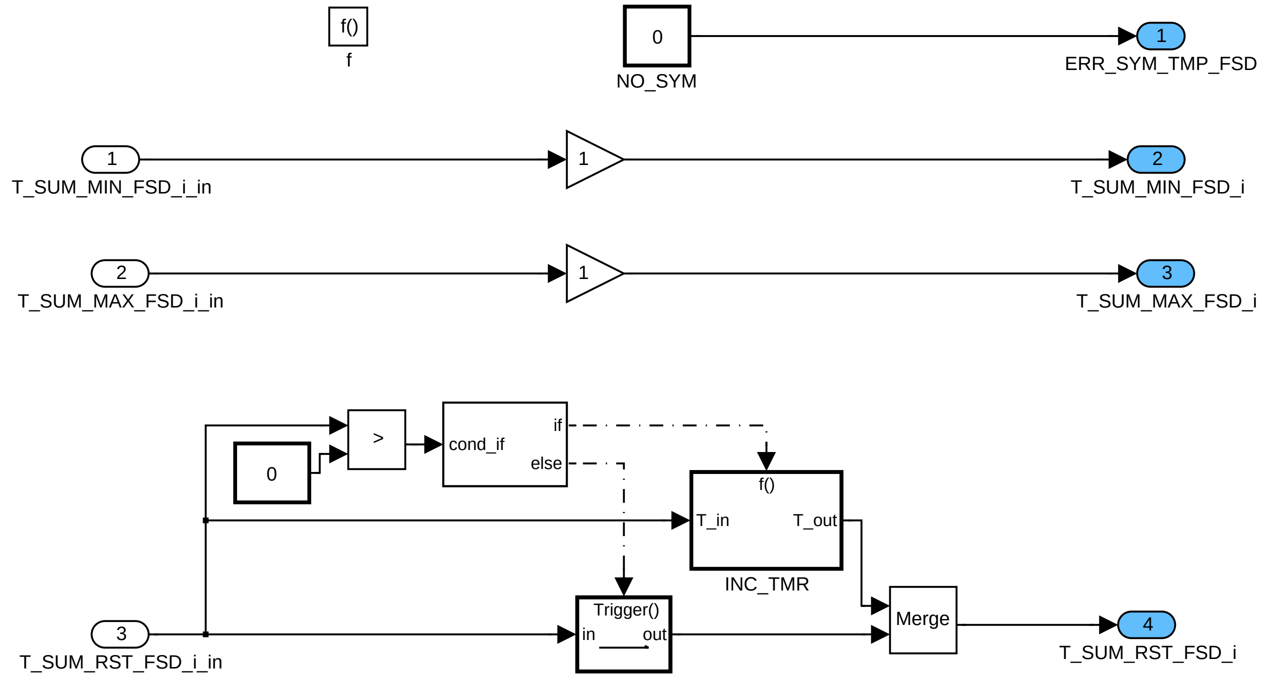


Figure 218 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ NO_SYM_FSD_L_RNG

Increment timer

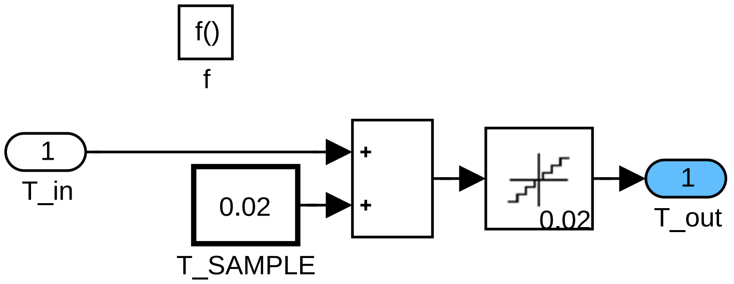


Figure 219 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ NO_SYM_FSD_L_RNG/ INC_TMR

Set error flag for lambda adaptation low range monitoring

The temporary error flag is set if the lambda adaptation factor was either to long at its maximum or at its minimum.

When the reset timer reaches the threshold C_T_SUM_RST_FSD all timers are reset to 0.

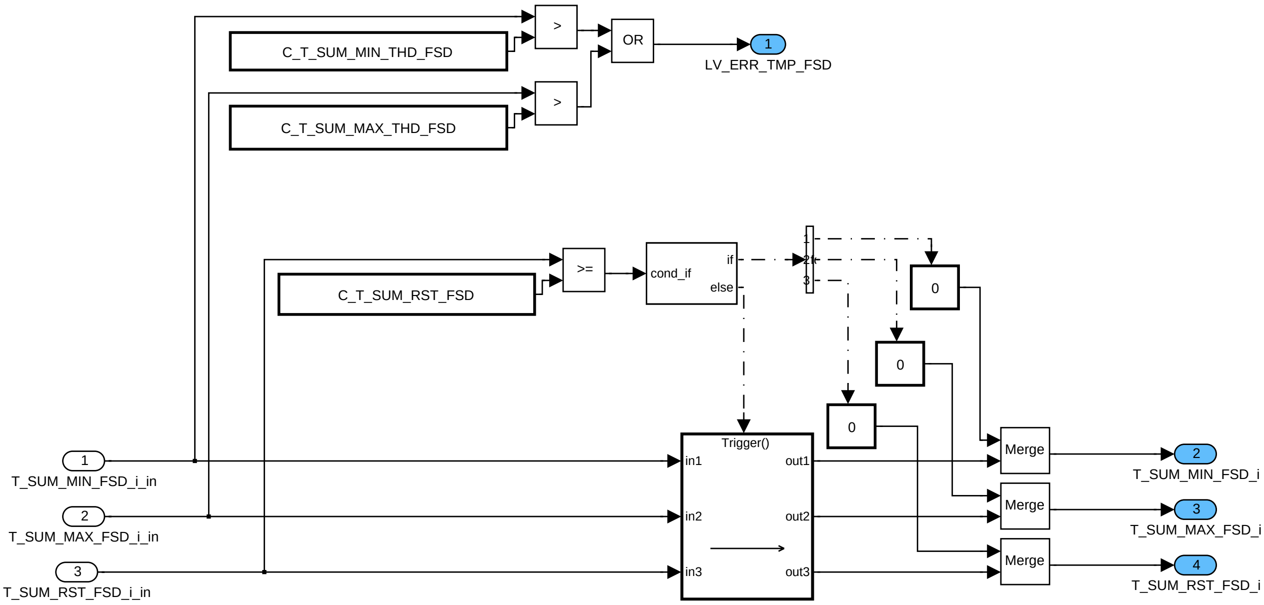


Figure 220 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ CDN_DIAG_FSD_L_RNG/ SET_ERR_L_RNG__RST_T_SUM

Conditions for lambda adaptation monitoring not fulfilled

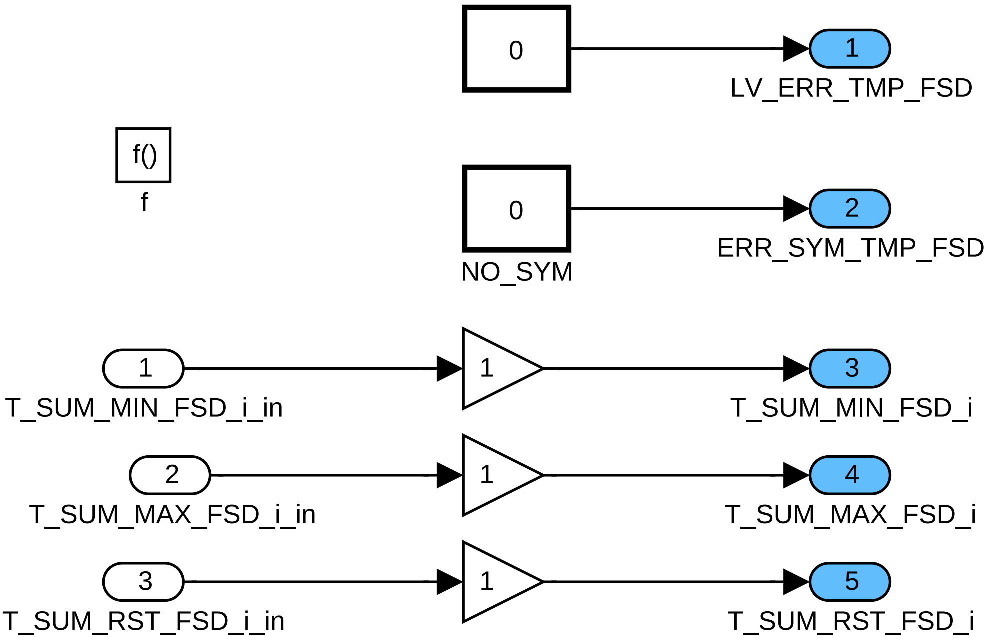


Figure 221 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD/ NOT_CDN_DIAG_FSD_MFF_ADD_L_RNG

Error symptom detection for lambda controller monitoring

In case the conditions are not fulfilled all timers remain unchanged and the temporary error flag and symptom have no relevance.

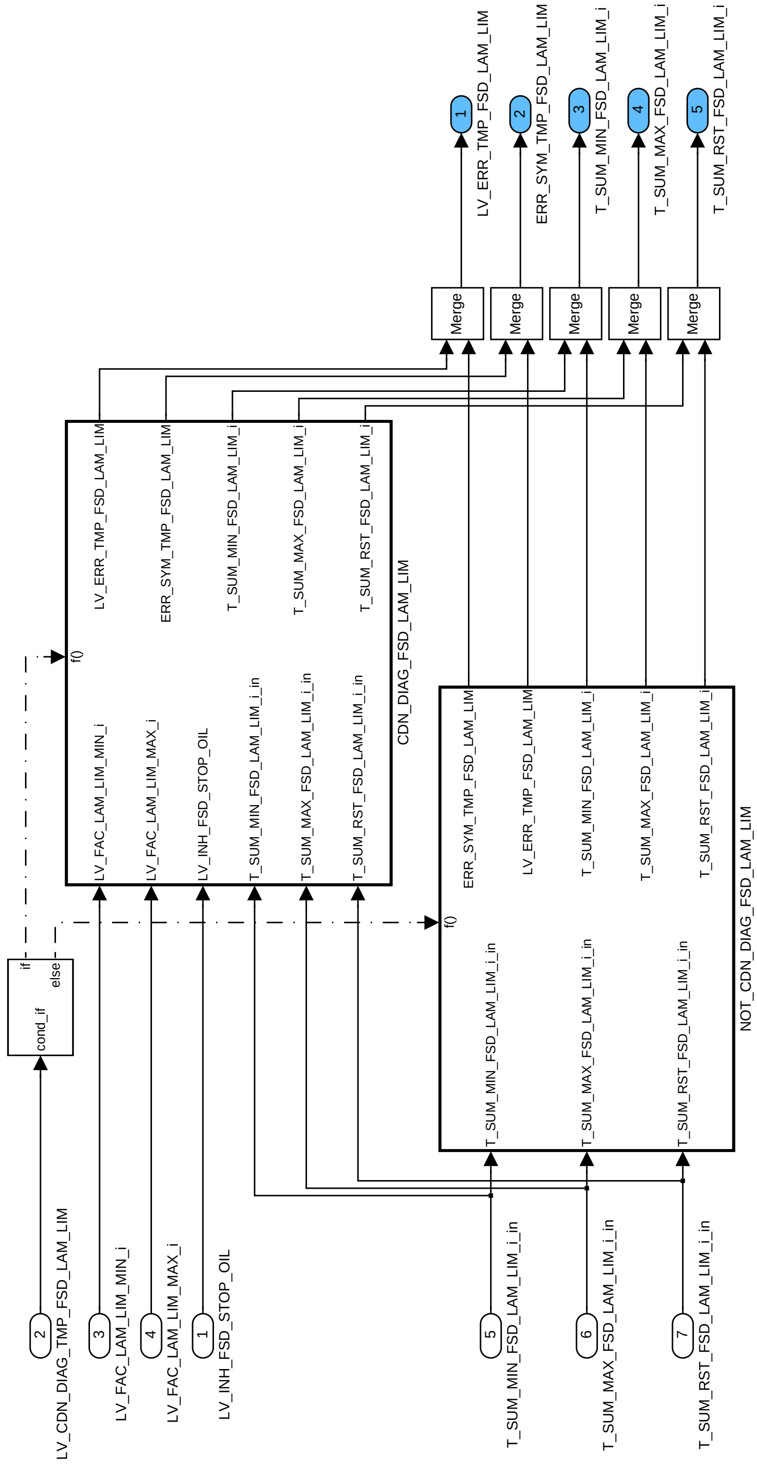


Figure 222 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM

Conditions for lambda controller monitoring fulfilled

Between 3 different cases must be distinguished:

1. Lambda controller is at its minimum threshold (indicated by the flag LV_FAC_LAM_LIM_MIN[i]) and no inhibition due to oil dilution under cold start conditions.

2. Lambda controller is at its maximum threshold (indicated by the flag LV_FAC_LAM_LIM_MAX[i]).

3. No limitation of the lambda controller.

Independent of these cases the anti-bounce timers used to finally set the error are evaluated afterwards.

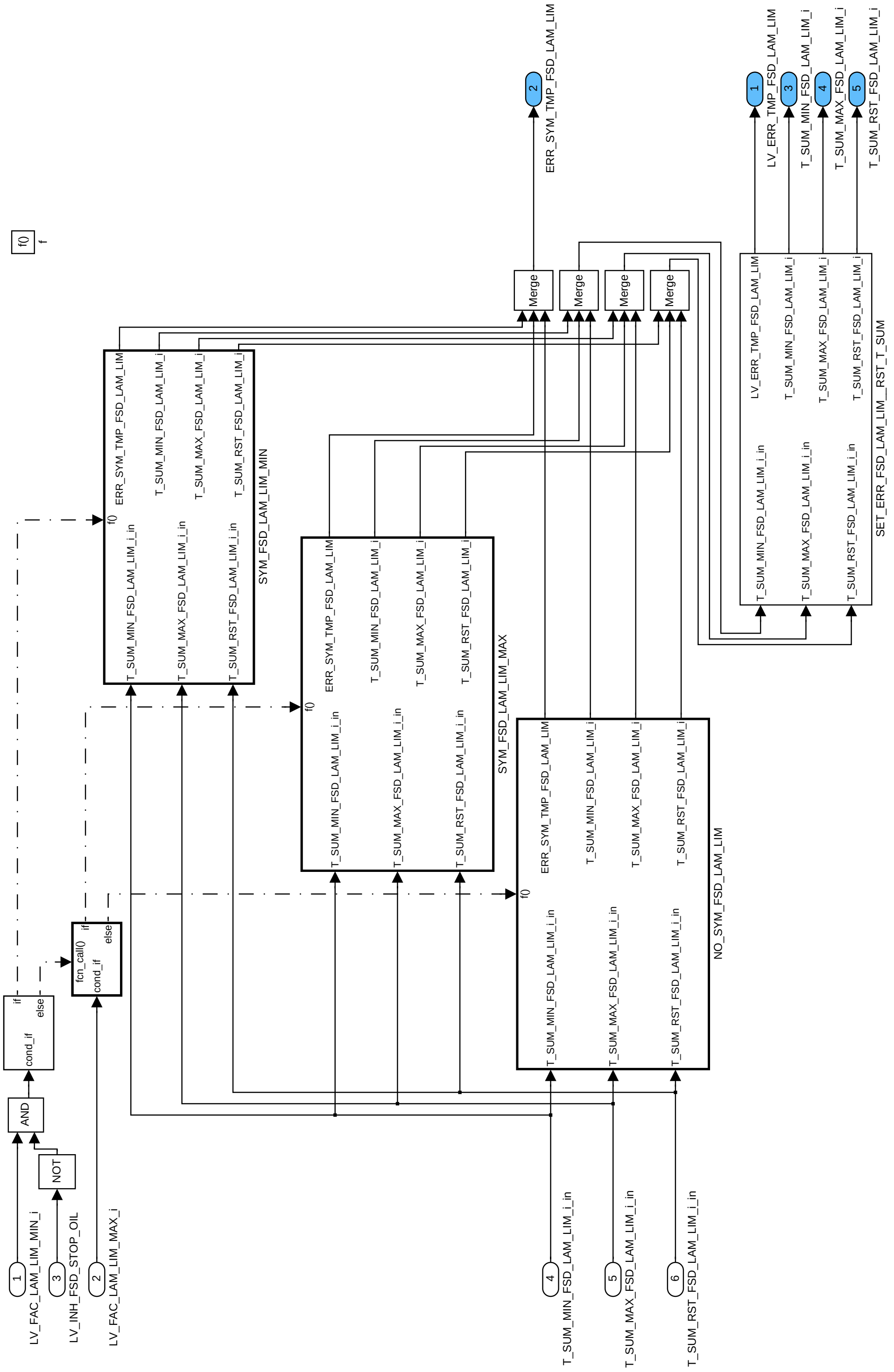


Figure 223 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM

Set error symptom for lambda controller at the minimum

The error symptom is set to the respective value (SYM_1). The timer indicating that the minimum threshold is reached and the reset timer are both increased.

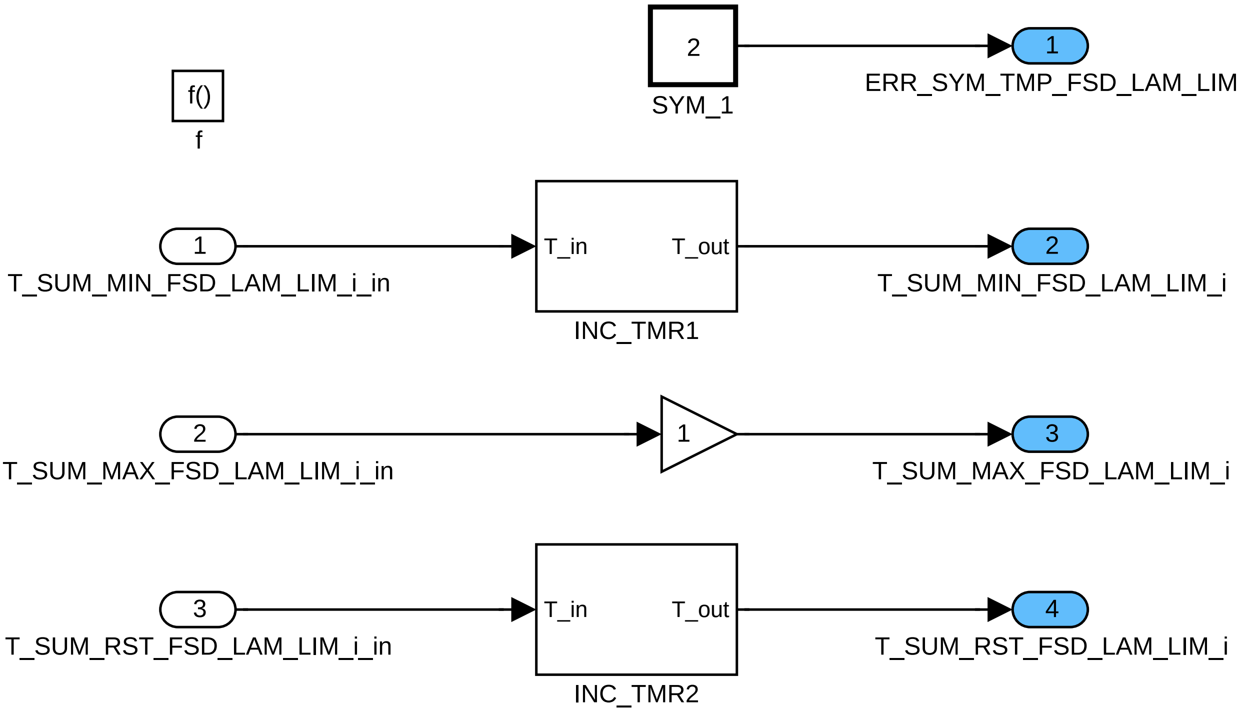


Figure 224 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ SYM_FSD_LAM_LIM_MIN

Increment timer

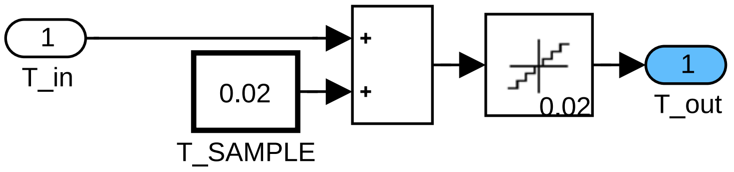


Figure 225 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ SYM_FSD_LAM_LIM_MIN/ INC_TMR1

Increment timer

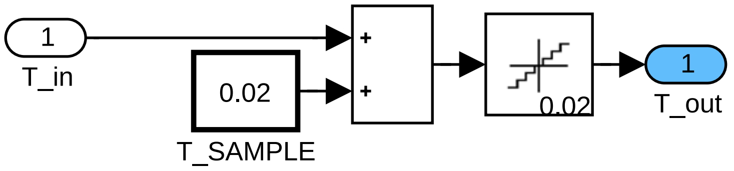


Figure 226 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ SYM_FSD_LAM_LIM_MIN/ INC_TMR2

general specification

Set error symptom for lambda controller at the maximum

The error symptom is set to the respective value (SYM_0). The timer indicating that the maximum threshold is reached and the reset timer are both increased.

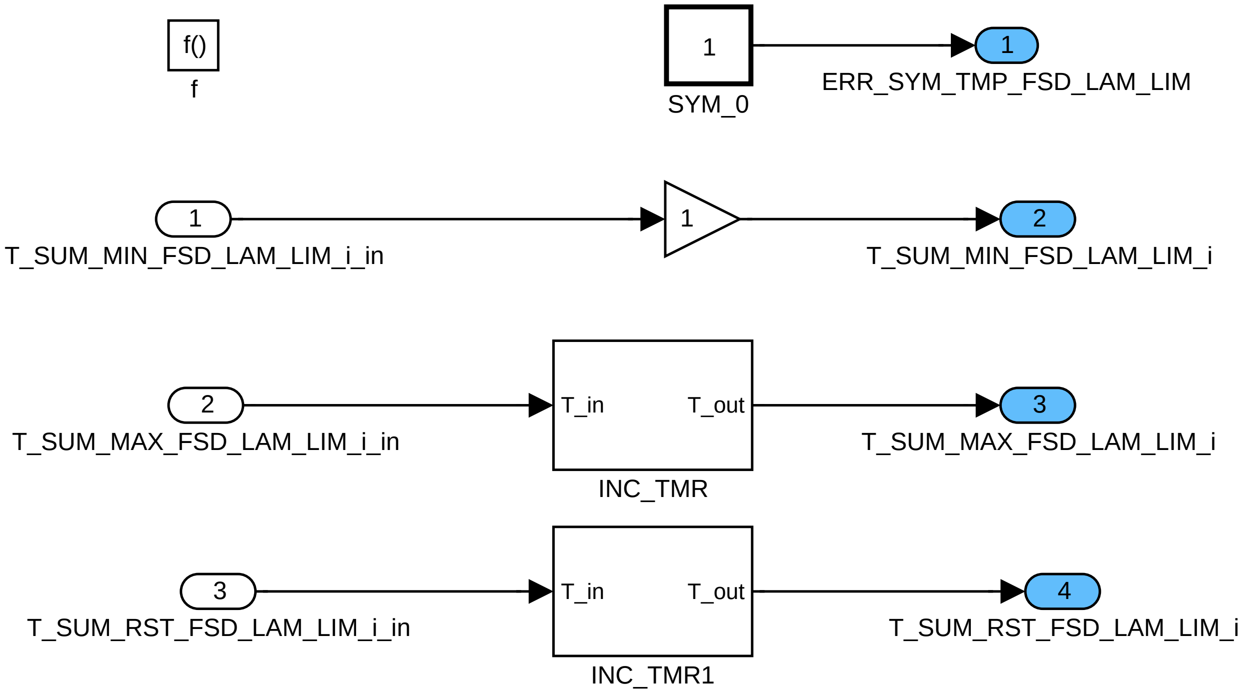


Figure 227 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ SYM_FSD_LAM_LIM_MAX

Increment timer

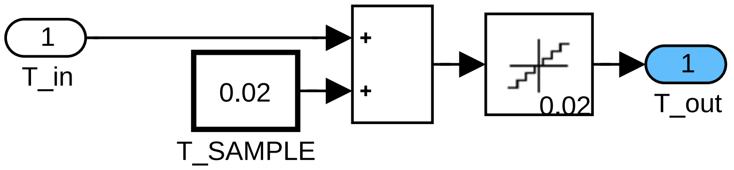


Figure 228 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ SYM_FSD_LAM_LIM_MAX/ INC_TMR

Increment timer

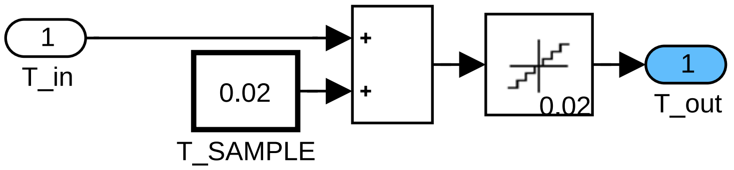


Figure 229 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ SYM_FSD_LAM_LIM_MAX/ INC_TMR1

No error symptom for lambda controller monitoring

The error symptom is set to NO_SYM and the reset timer is increased in case it is already different from 0. In that case the error symptom was at least one time different from NO_SYM.

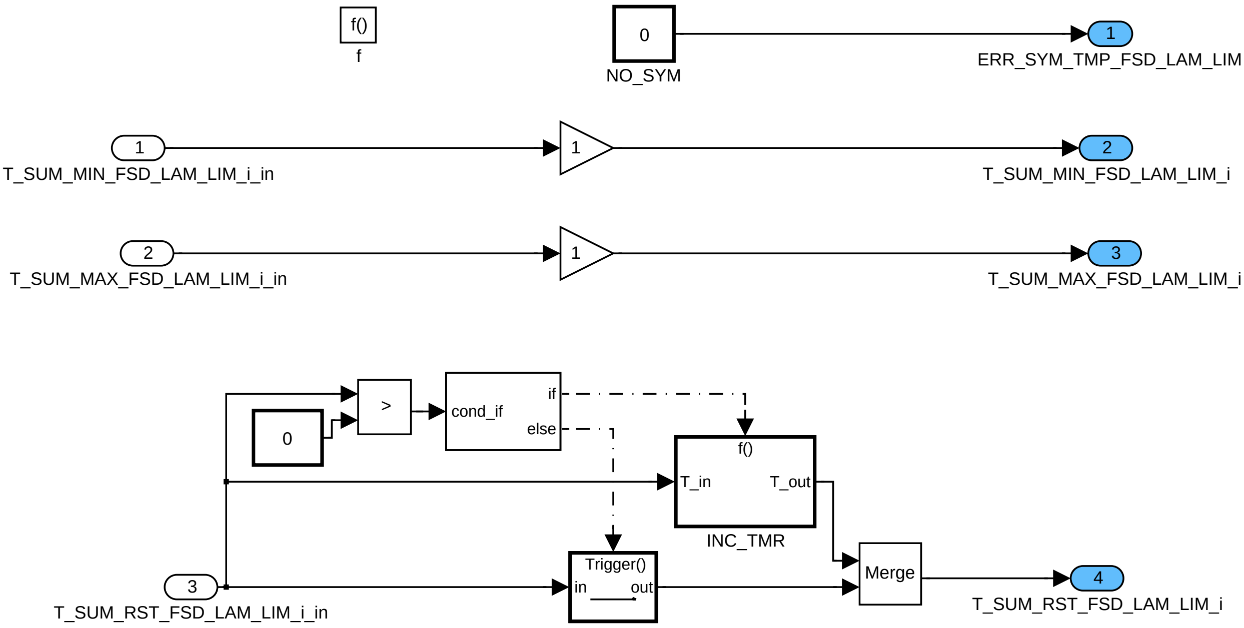


Figure 230 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ NO_SYM_FSD_LAM_LIM

Increment timer

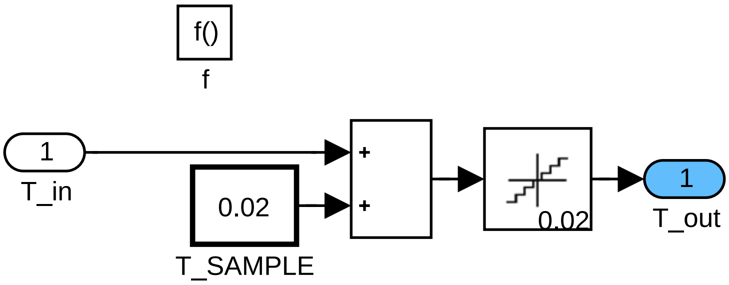


Figure 231 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ NO_SYM_FSD_LAM_LIM/ INC_TMR

Set error flag for lambda controller monitoring

The temporary error flag is set if the lambda controller was either to long at its maximum or at its minimum.

When the reset timer reaches the threshold C_T_SUM_RST_FSD all timers are reset to 0.

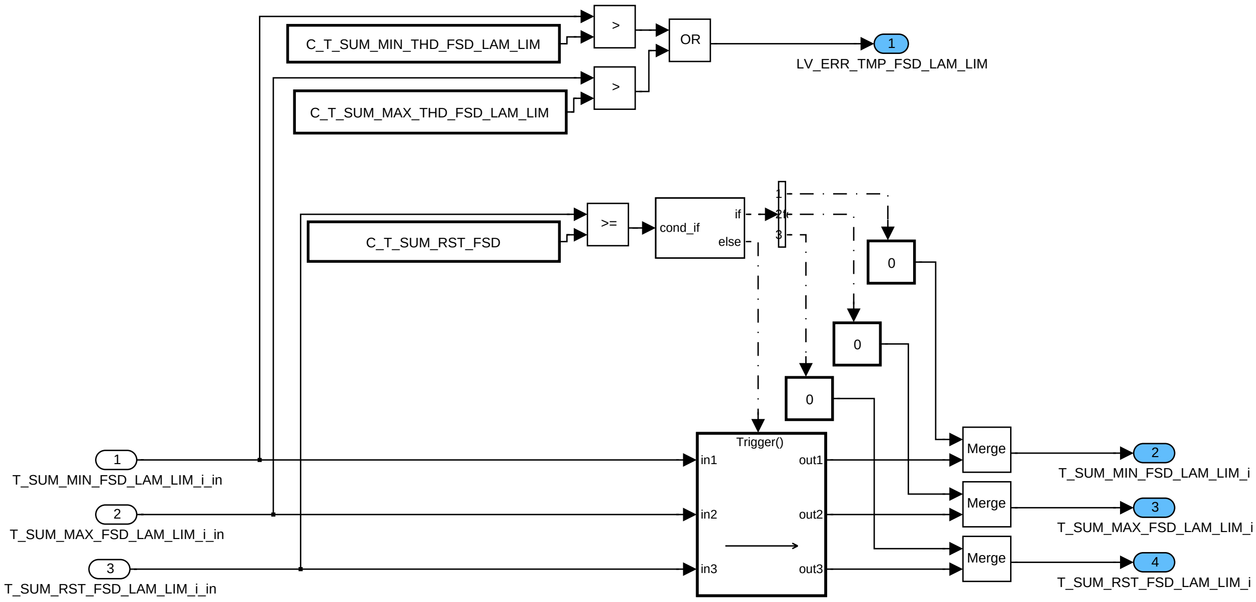


Figure 232 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ CDN_DIAG_FSD_LAM_LIM/ SET_ERR_FSD_LAM_LIM__RST_T_SUM

Conditions for lambda controller monitoring not fulfilled

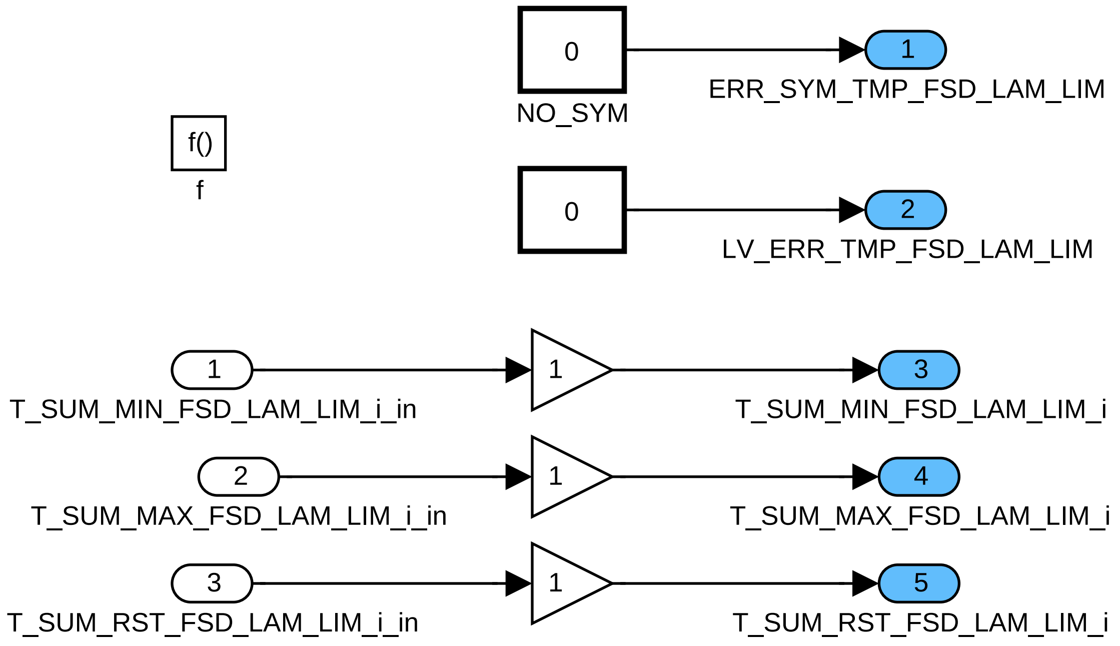


Figure 233 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_LAM_LIM/ NOT_CDN_DIAG_FSD_LAM_LIM

Error symptom detection for lambda adaptation high range monitoring

In case the conditions are not fulfilled all timers remain unchanged and the temporary error flag and symptom have no relevance.

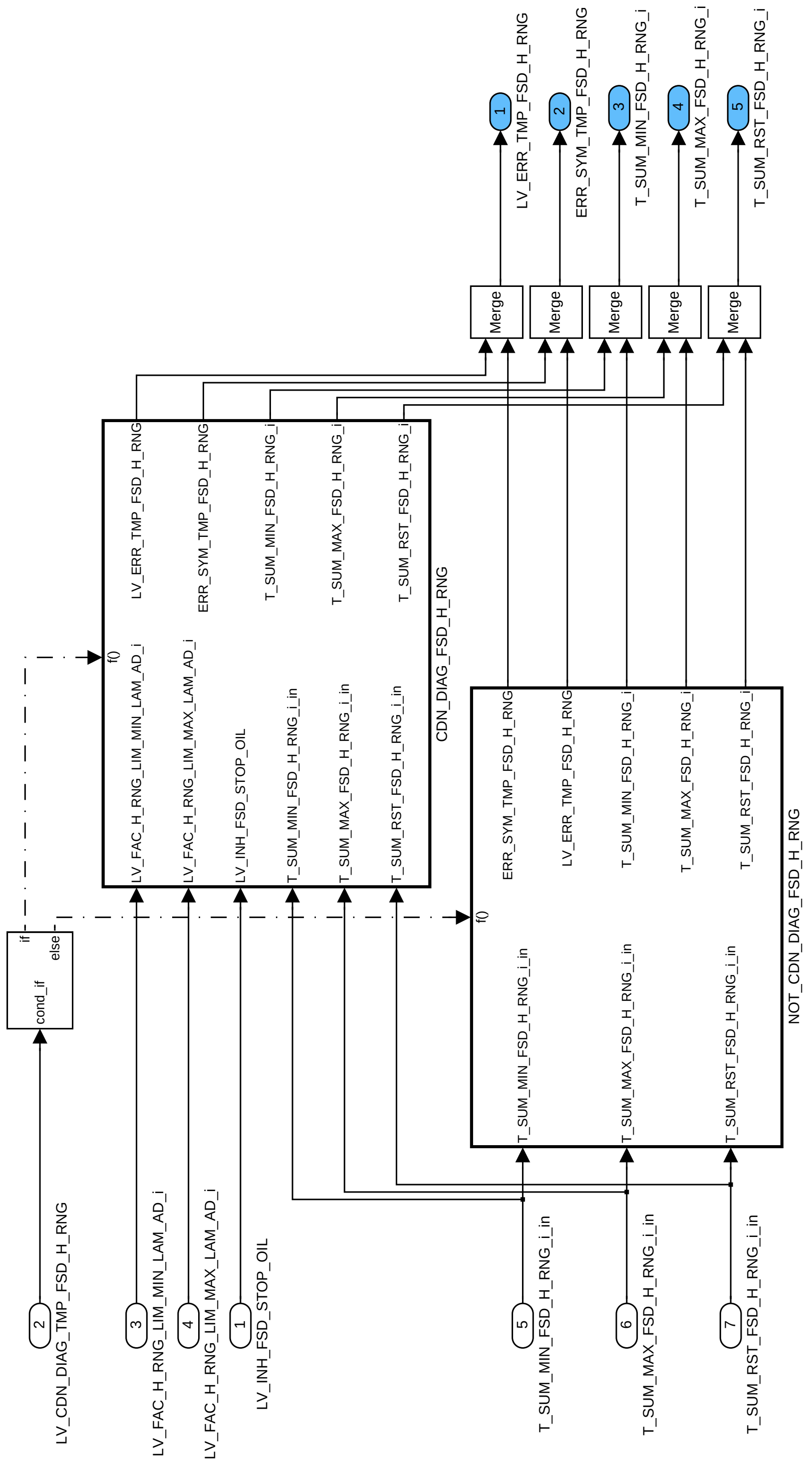


Figure 234 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG

Conditions for lambda adaptation high range monitoring fulfilled

Between 3 different cases must be distinguished:

1. Lambda adaptation factor for higher load is at its minimum threshold (indicated by the flag LV_FAC_H_RNG_LIM_MIN_LAM_AD[i]) and no inhibition due to oil dilution under cold start conditions.

2. Lambda adaptation factor for higher load is at its maximum threshold (indicated by the flag LV_FAC_H_RNG_LIM_MAX_LAM_AD[i]).

3. No limitation of the lambda adaptation factor for higher load.

Independent of these cases the anti-bounce timers used to finally set the error are evaluated afterwards.

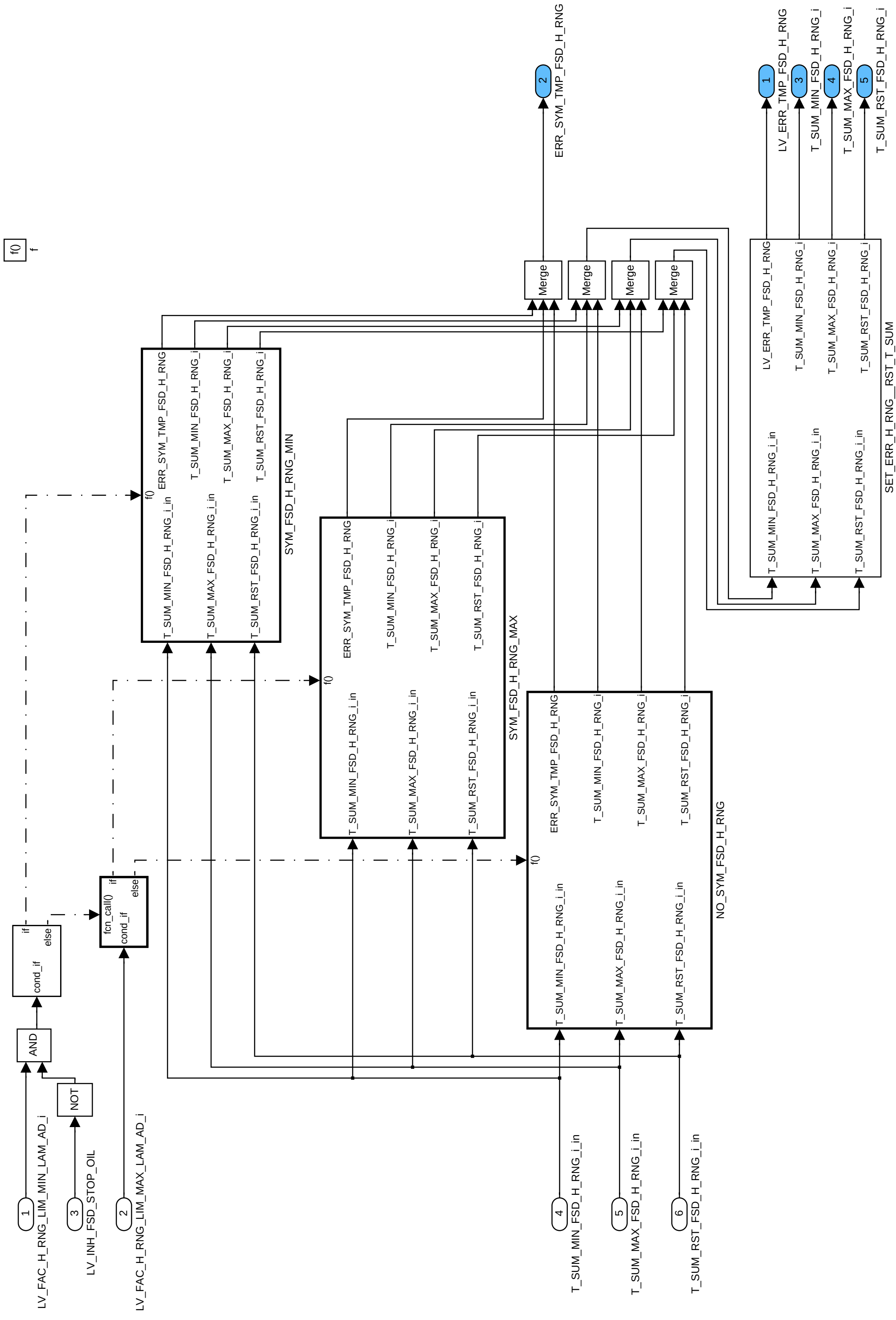


Figure 235 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG

Set error symptom for lambda adaptation high range at the minimum

The error symptom is set to the respective value (SYM_1). The timer indicating that the minimum threshold is reached and the reset timer are both increased.

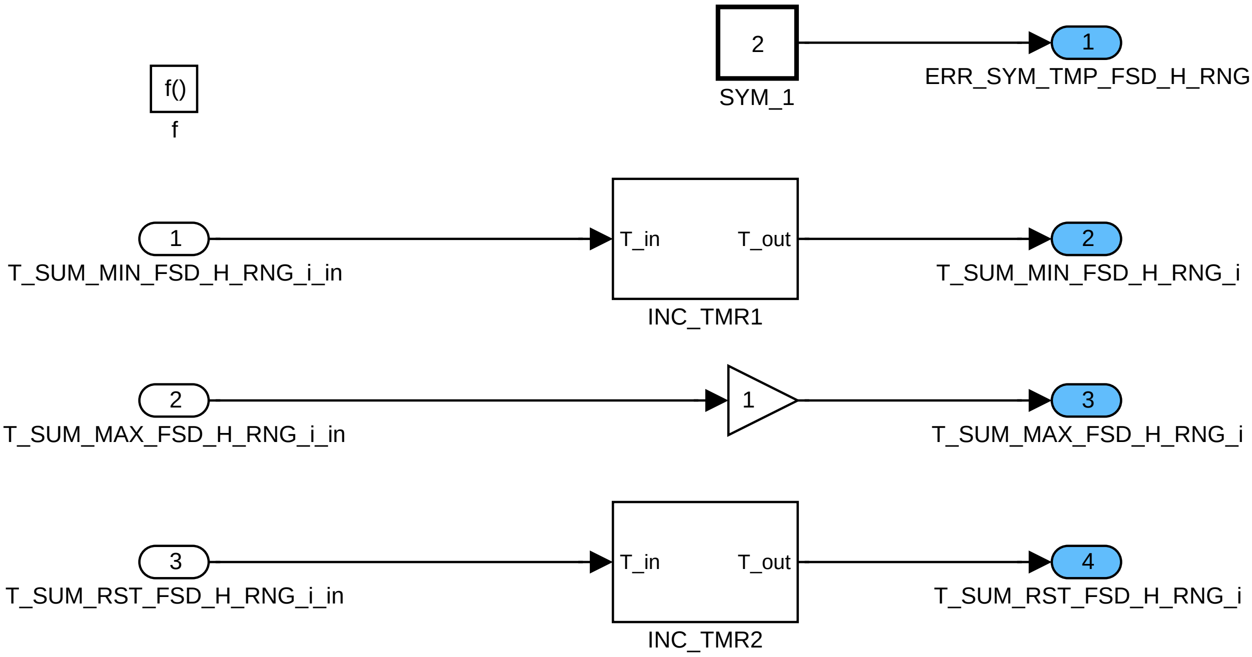


Figure 236 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ SYM_FSD_H_RNG_MIN

Increment timer

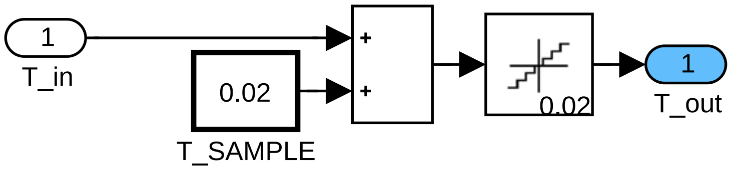


Figure 237 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ SYM_FSD_H_RNG_MIN/ INC_TMR1

Increment timer

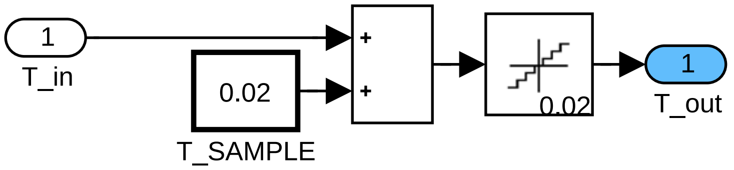


Figure 238 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ SYM_FSD_H_RNG_MIN/ INC_TMR2

Set error symptom for lambda adaptation high range at the maximum

The error symptom is set to the respective value (SYM_0). The timer indicating that the maximum threshold is reached and the reset timer are both increased.

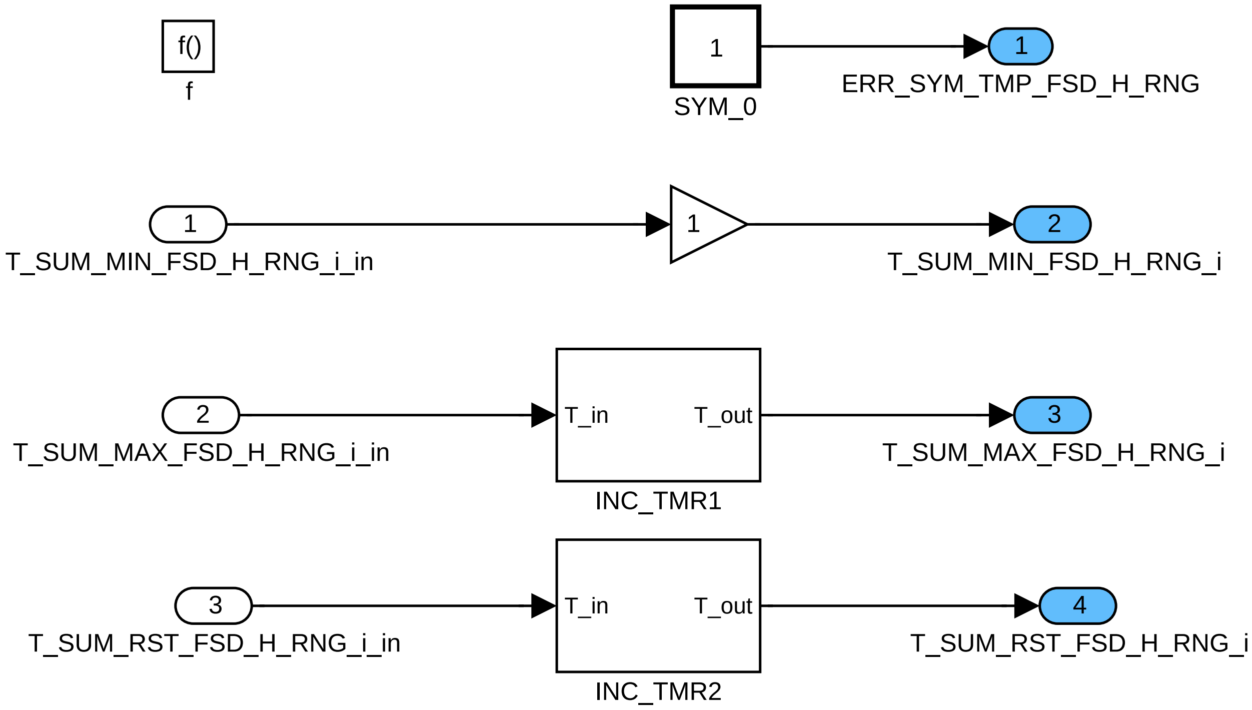


Figure 239 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ SYM_FSD_H_RNG_MAX

Increment timer

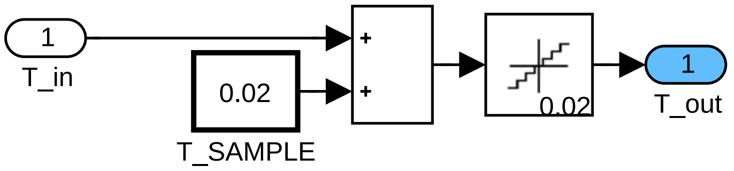


Figure 240 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ SYM_FSD_H_RNG_MAX/ INC_TMR1

Increment timer

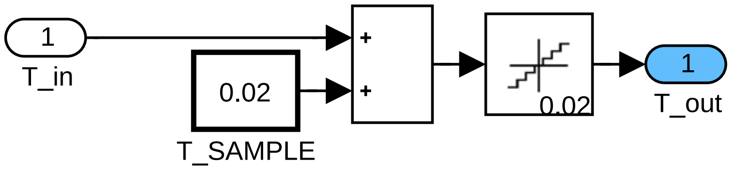


Figure 241 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ SYM_FSD_H_RNG_MAX/ INC_TMR2

No error symptom for lambda adaptation high range

The error symptom is set to NO_SYM and the reset timer is increased in case it is already different from 0. In that case the error symptom was at least one time different from NO_SYM.

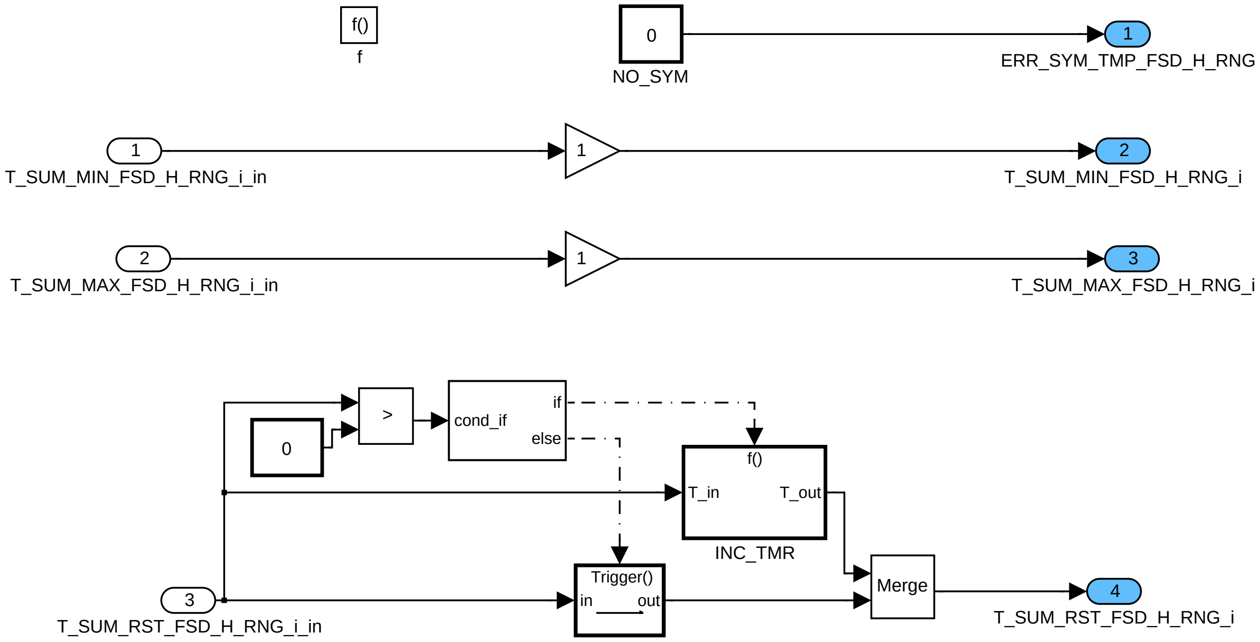


Figure 242 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ NO_SYM_FSD_H_RNG

Increment timer

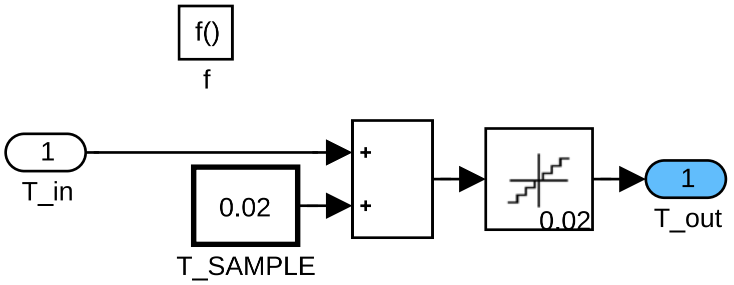


Figure 243 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ NO_SYM_FSD_H_RNG/ INC_TMR

Set error flag for lambda adaptation high range monitoring

The temporary error flag is set if the lambda adaptation factor was either to long at its maximum or at its minimum.

When the reset timer reaches the threshold C_T_SUM_RST_FSD all timers are reset to 0.

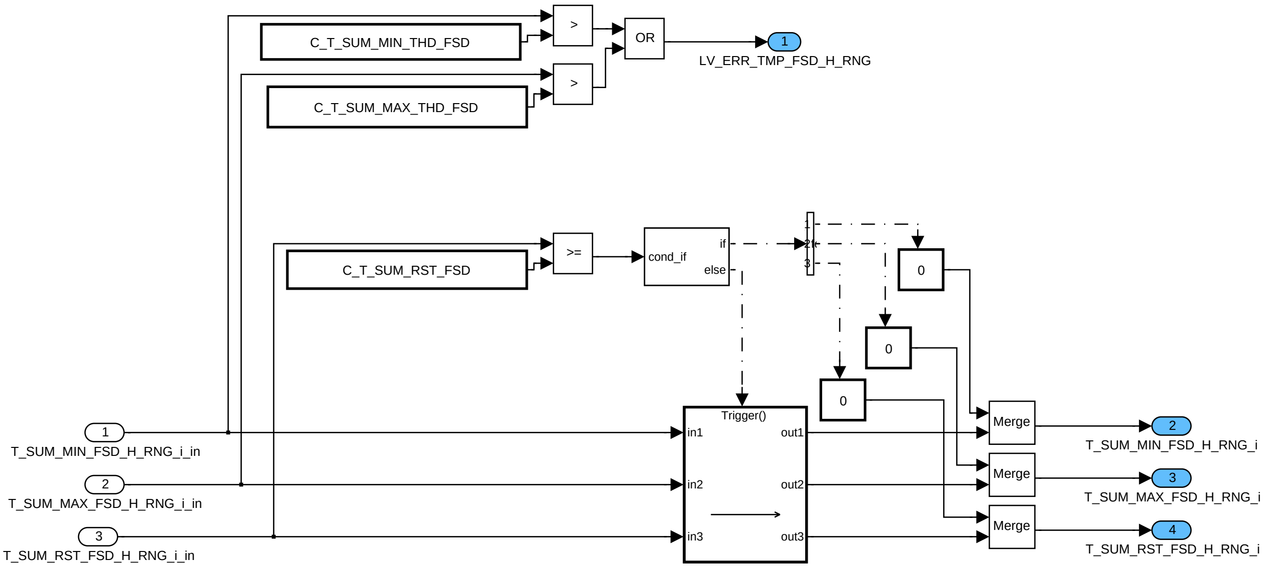


Figure 244 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ CDN_DIAG_FSD_H_RNG/ SET_ERR_H_RNG__RST_T_SUM

Conditions for lambda adaptation high range monitoring not fulfilled

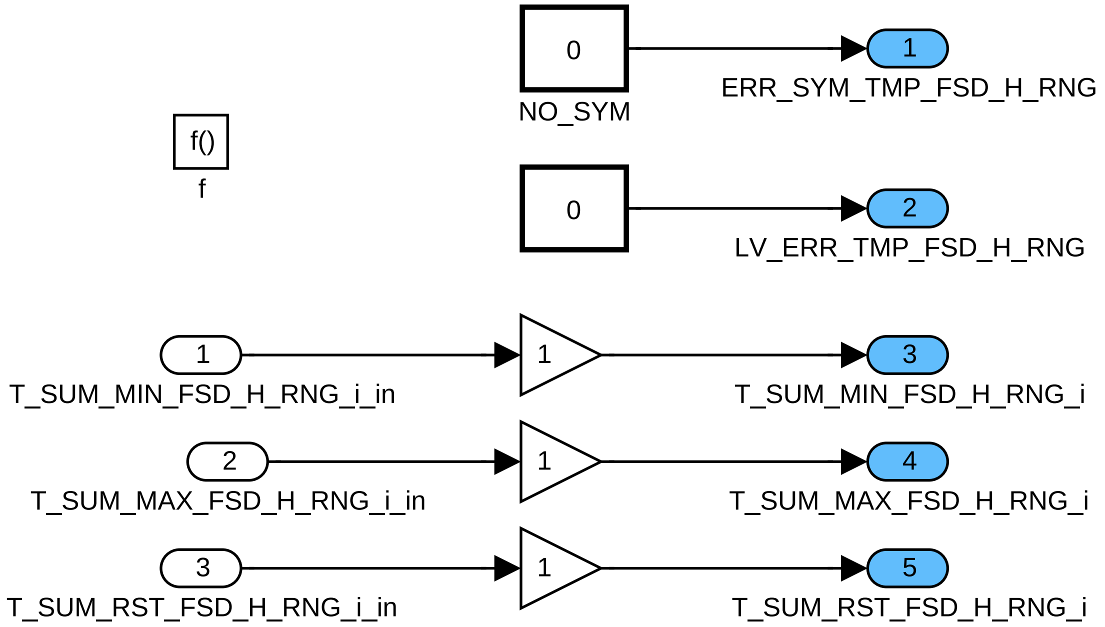


Figure 245 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ DET_ERR/ DET_ERR_FSD_H_RNG/ NOT_CDN_DIAG_FSD_H_RNG

Management of end of diagnosis flags

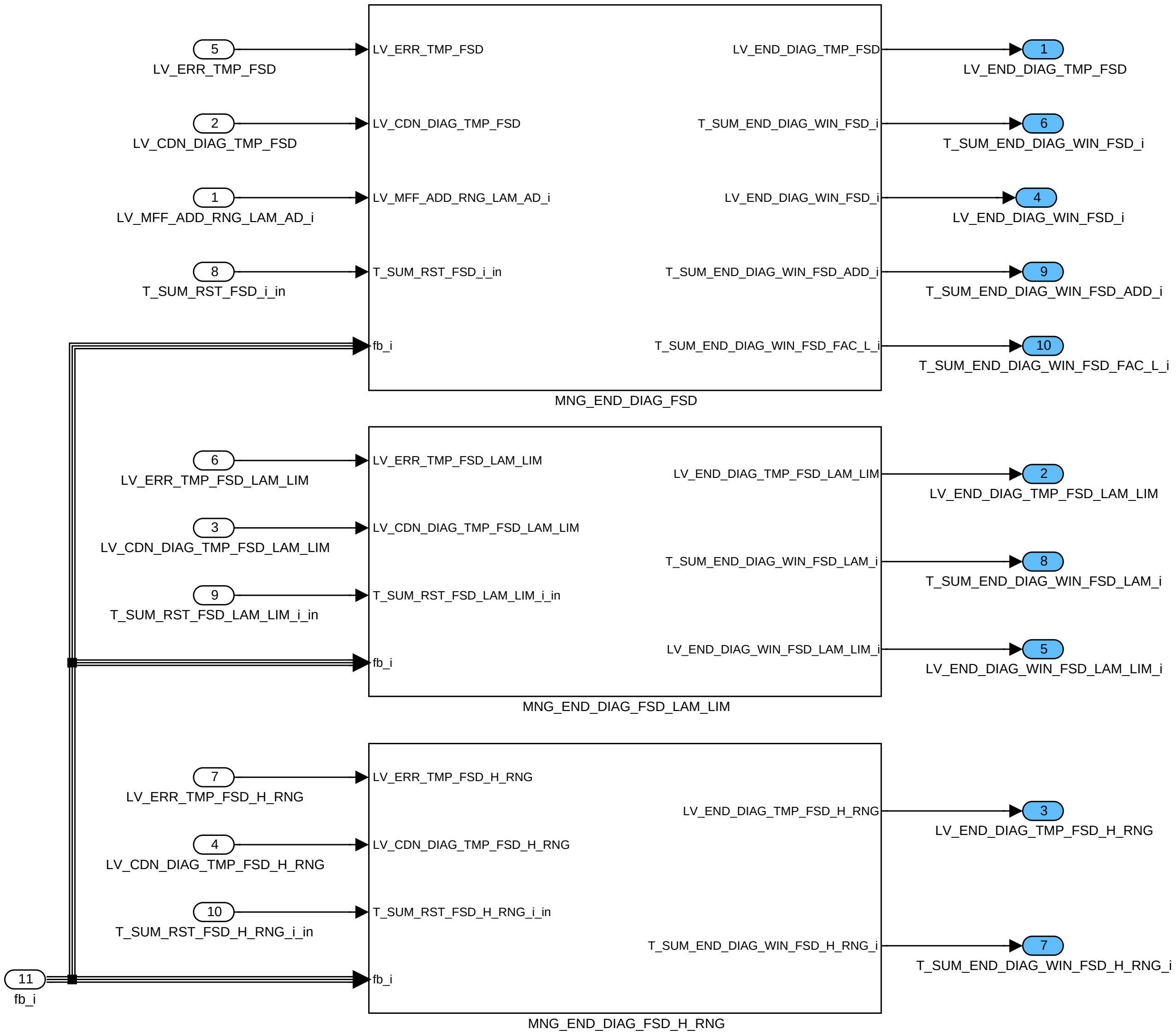


Figure 246 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG

Manage end flags for lambda adaptation monitoring

First the end flag timers are calculated and then the conditions to set the end flags are evaluated.

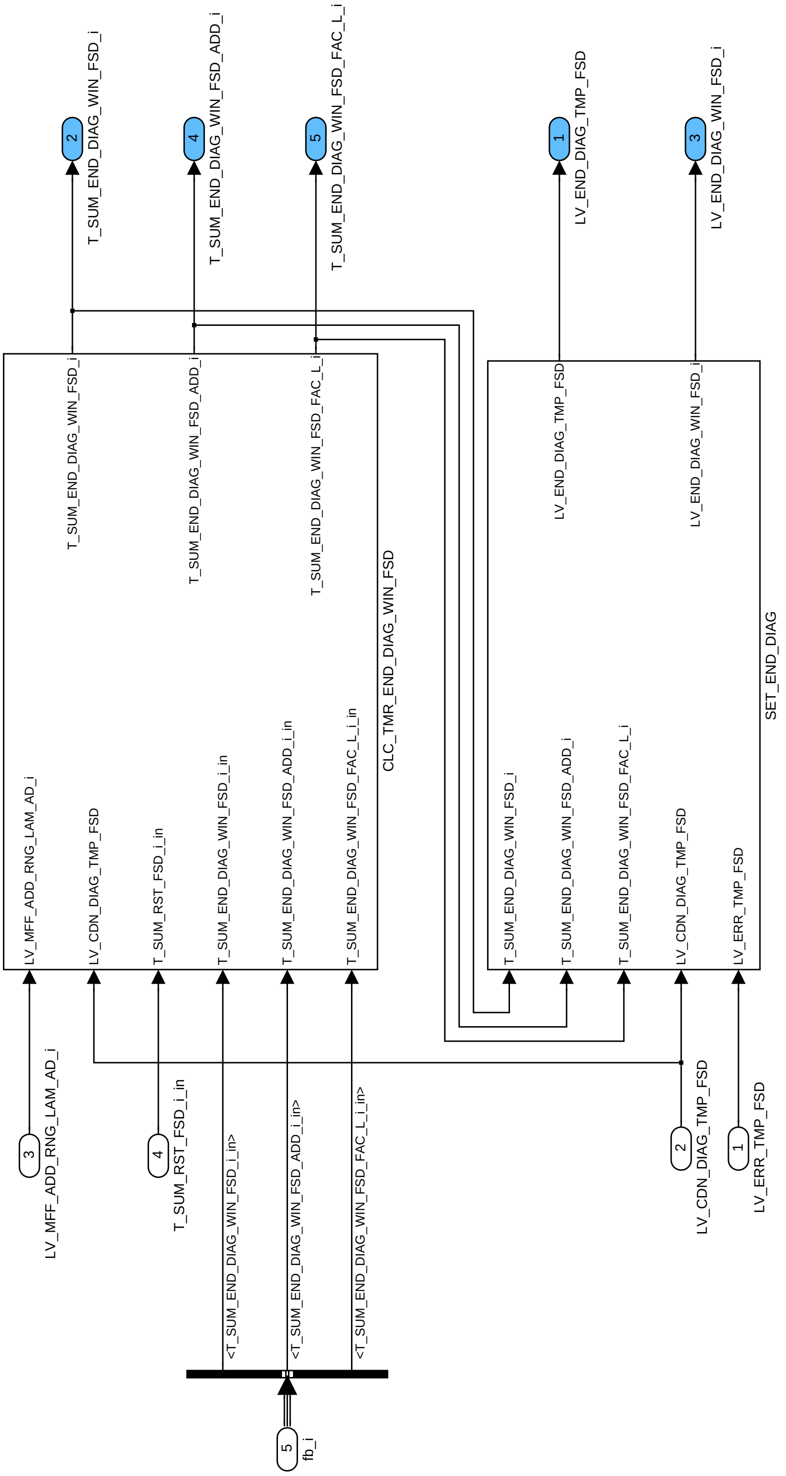


Figure 247 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD

Calculate end of diagnosis timers for lambda adaptation monitoring

If the conditions are not fulfilled all timers T_SUM_END_DIAG_WIN_FSD[i], T_SUM_END_DIAG_WIN_FSD_ADD[i] and T_SUM_END_DIAG_WIN_FSD_FAC_L[i] remain unchanged.

The end timers are reset (to one sample step) either if the calibration threshold is reached by T_SUM_END_DIAG_WIN_FSD[i] and both timers T_SUM_END_DIAG_WIN_FSD_ADD[i] and T_SUM_END_DIAG_WIN_FAC_L[i] were above the threshold defined by C_T_SUM_RST_FSD and C_CRLC_T_SUM_WIN_FSD_RST or when the reset timer was started to be incremented.

In all other cases the timers are incremented as described in the corresponding sub-system.

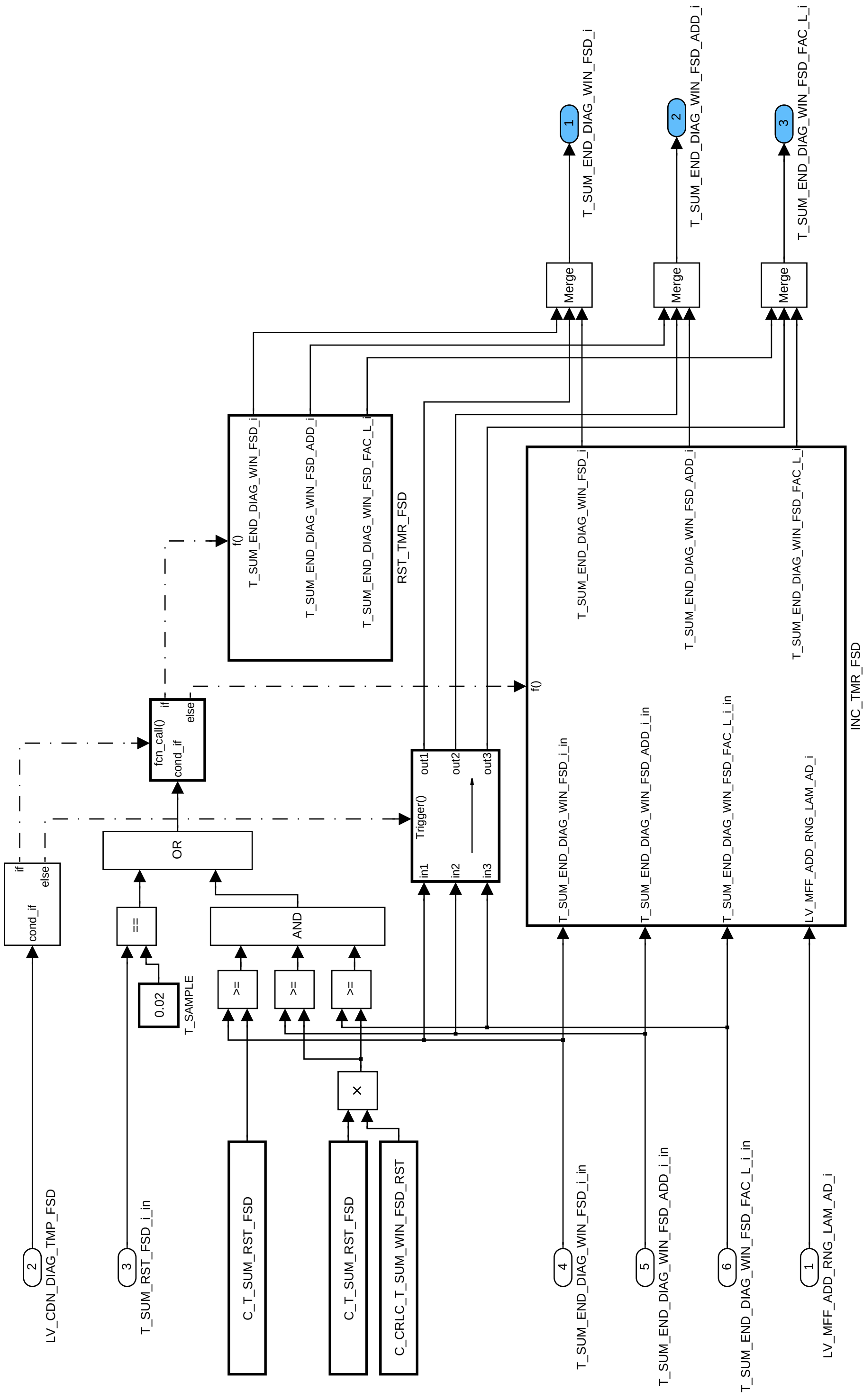


Figure 248 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD/ CLC_TMR_END_DIAG_WIN_FSD

Reset end of diagnosis timers for lambda adaptation monitoring

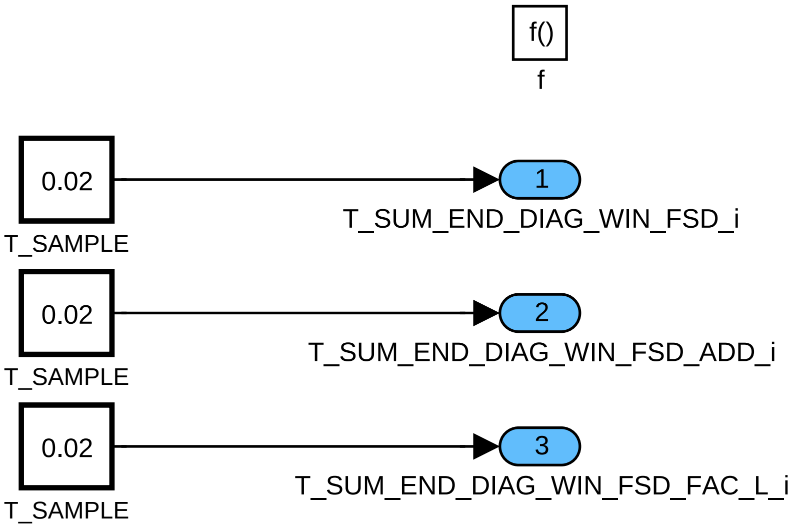


Figure 249 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD/ CLC_TMR_END_DIAG_WIN_FSD/ RST_TMR_FSD

Increment end of diagnosis timers for lambda adaptation monitoring

T_SUM_END_DIAG_WIN_FSD[i] is incremented without restrictions.

T_SUM_END_DIAG_WIN_FSD_ADD[i] is incremented when the lambda adaptation is in the offset learning field.

T_SUM_END_DIAG_WIN_FSD_FAC_L[i] is incremented when the lambda adaptation is in the lower load factor learning field.

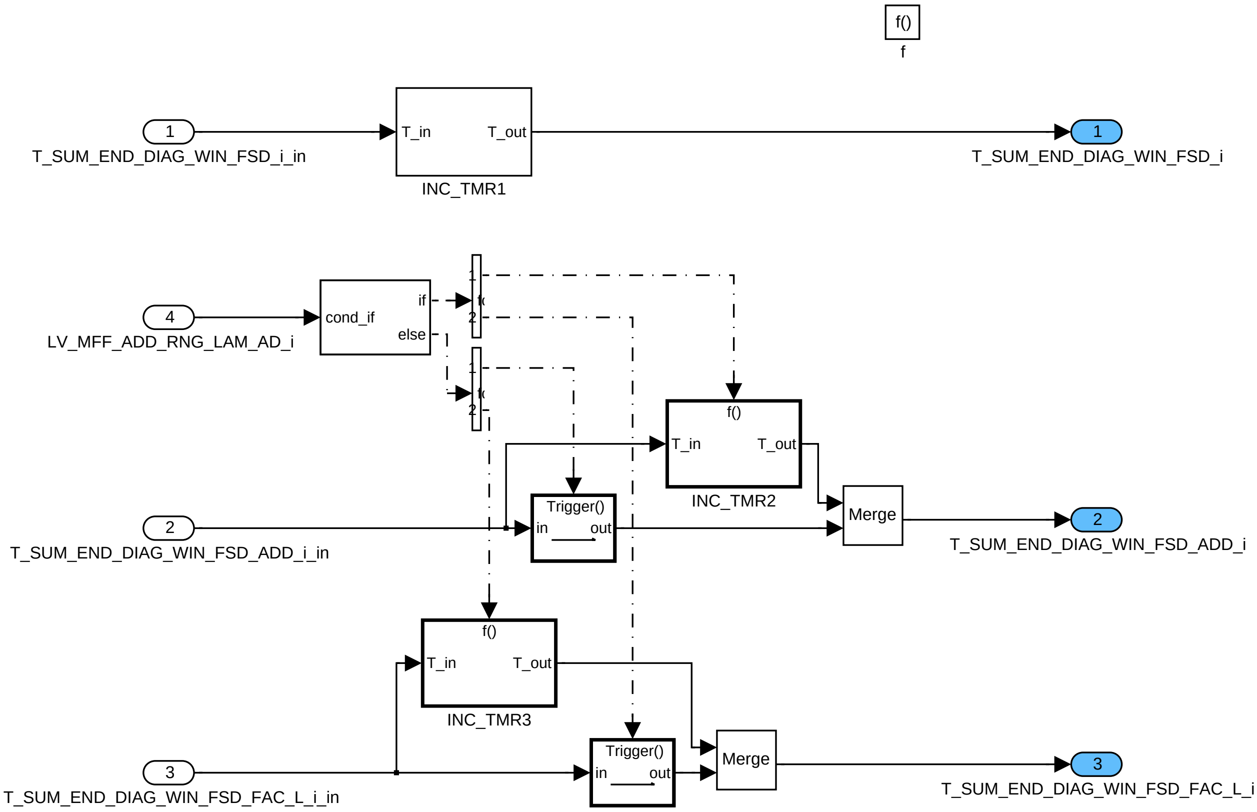


Figure 250 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD/ CLC_TMR_END_DIAG_WIN_FSD/ INC_TMR_FSD

Increment timer

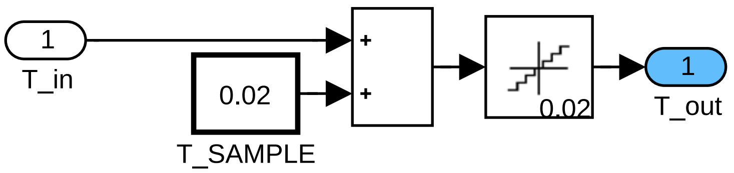


Figure 251 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD/ CLC_TMR_END_DIAG_WIN_FSD/ INC_TMR_FSD/ INC_TMR1

Increment timer

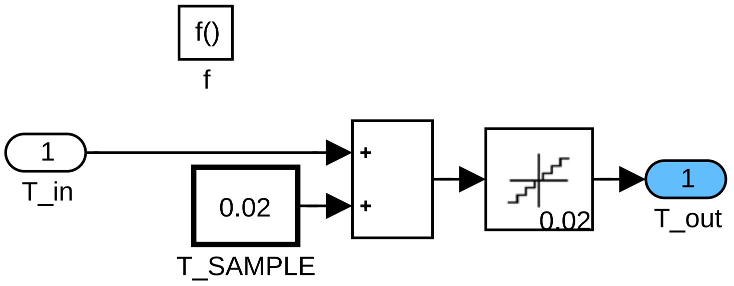


Figure 252 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD/ CLC_TMR_END_DIAG_WIN_FSD/ INC_TMR_FSD/ INC_TMR2

Increment timer

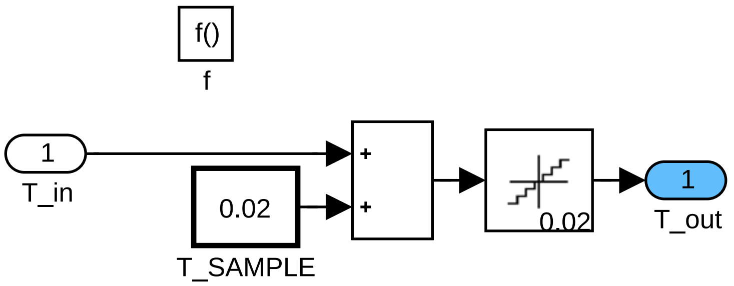


Figure 253 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD/ CLC_TMR_END_DIAG_WIN_FSD/ INC_TMR_FSD/ INC_TMR3

Set end of diagnosis for lambda adaptation monitoring

The condition to set the temporary end flag LV_END_DIAG_TMP_FSD are: Diagnostic condition must be fulfilled and either an error was detected (temporary end flag LV_ERR_TMP_FSD is 1) or the timer T_SUM_END_DIAG_WIN_FSD[i] has reached the calibration threshold.

If LV_END_DIAG_TMP_FSD is set once, the corresponding error flag of the error management remain set for the rest of the driving cycle.

In order to set the end flag for similar conditions LV_END_DIAG_WIN_FSD[i] the diagnostic conditions must be fulfilled too and either the temporary error flag was set or the timer T_SUM_END_DIAG_WIN_FSD[i] has reached the calibration threshold and both timers T_SUM_END_DIAG_WIN_FSD_ADD[i] and T_SUM_END_DIAG_WIN_FAC_L[i] were above the threshold defined by C_T_SUM_RST_FSD and C_CRLC_T_SUM_WIN_FSD_RST.

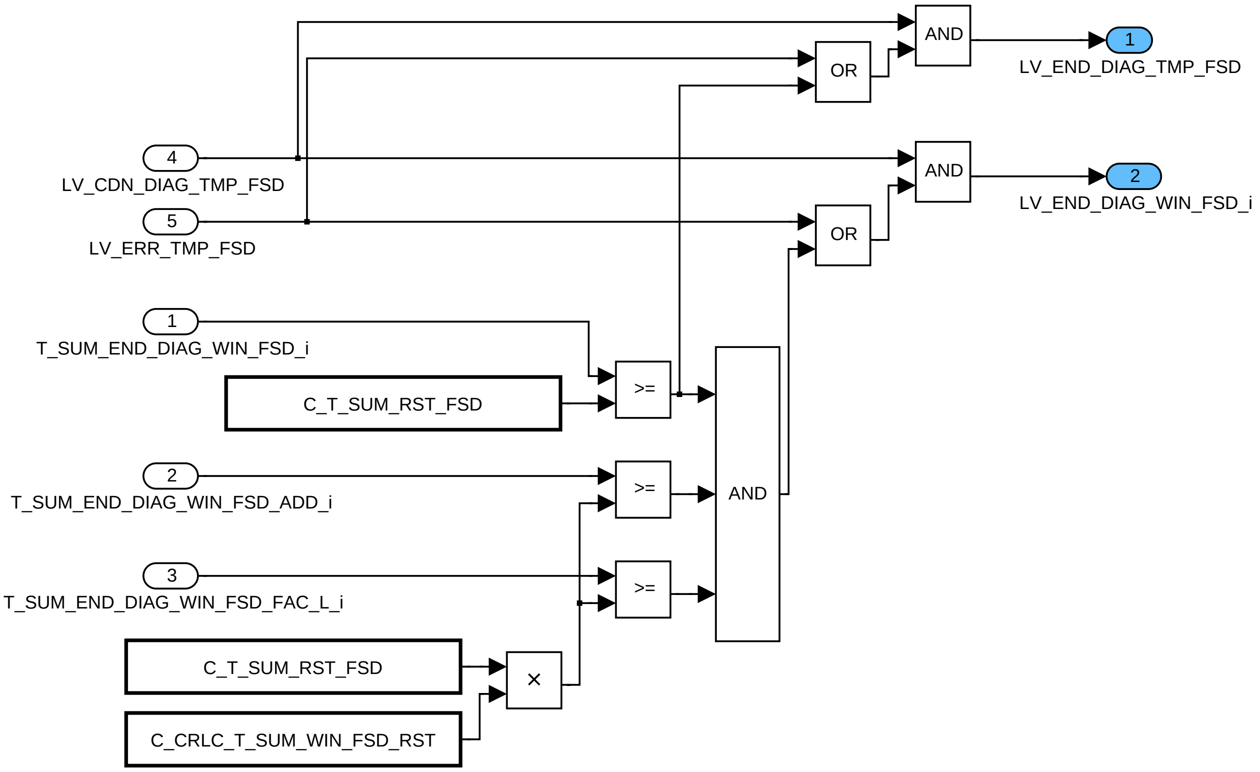


Figure 254 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD/ SET_END_DIAG

Manage end flags for lambda controller monitoring

First the end flag timers are calculated and then the conditions to set the end flags are evaluated.

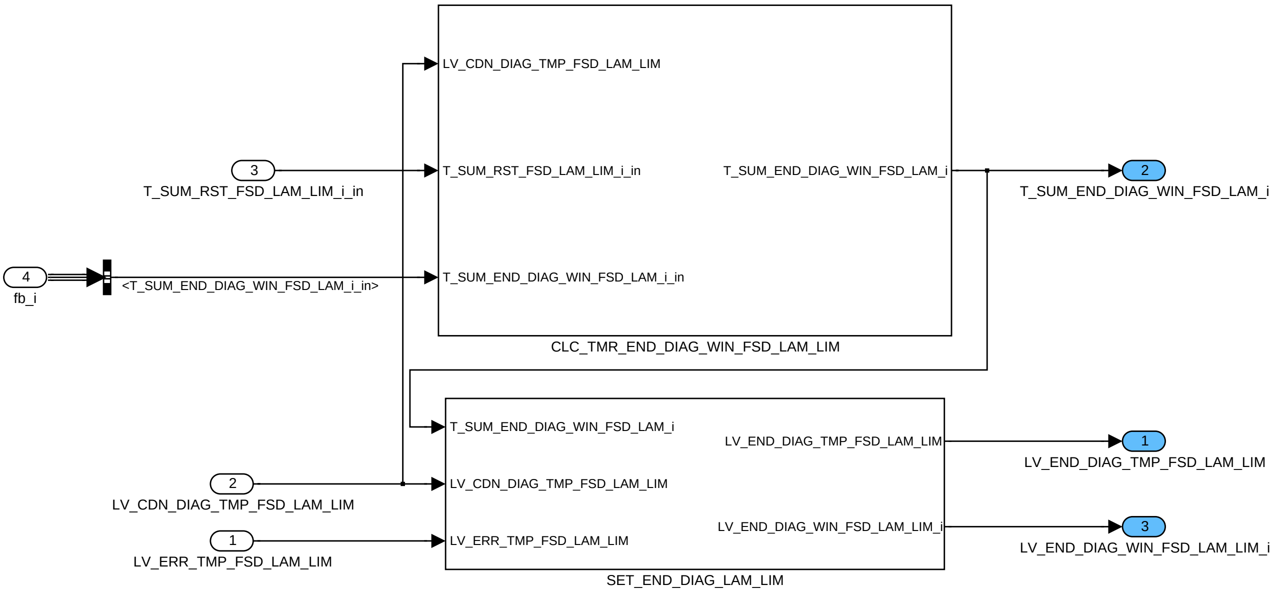


Figure 255 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD_LAM_LIM

Calculate end of diagnosis timers for lambda controller monitoring

If the conditions are not fulfilled the timer T_SUM_END_DIAG_WIN_FSD_LAM_LIM[i] remains unchanged.

The end timer is reset (to one sample step) either if the calibration threshold is reached or when the reset timer was started to be incremented.

In all other cases the timer is incremented.

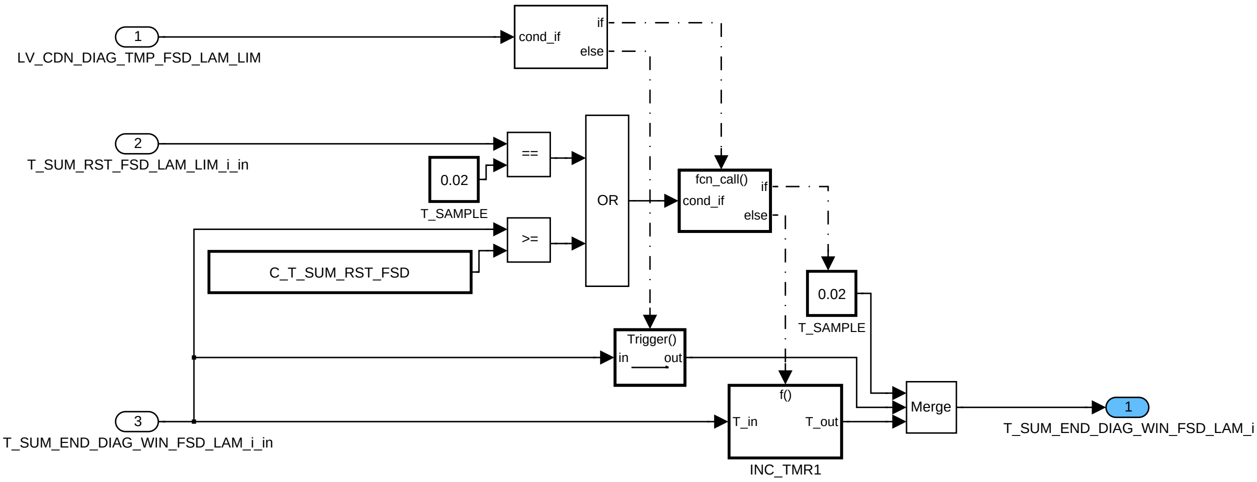


Figure 256 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD_LAM_LIM/ CLC_TMR_END_DIAG_WIN_FSD_LAM_LIM

Increment timer

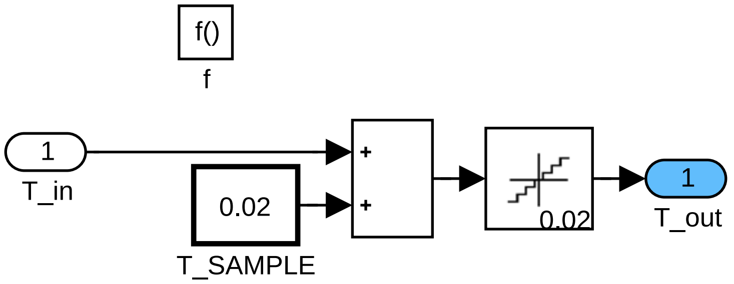


Figure 257 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD_LAM_LIM/ CLC_TMR_END_DIAG_WIN_FSD_LAM_LIM/ INC_TMR1

Set end of diagnosis for lambda controller monitoring

The temporary end flag LV_END_DIAG_TMP_FSD_LAM_LIM and the end flag for similar conditions LV_END_DIAG_WIN_FSD_LAM_LIM[i] are set when the same conditions are fulfilled. If LV_END_DIAG_TMP_FSD_LAM_LIM is set once, the corresponding error flag of the error management remain set for the rest of the driving cycle. LV_END_DIAG_WIN_FSD_LAM_LIM[i] is only set for one recurrence.

The condition to set the end flags are: Diagnostic condition must be fulfilled and either an error was detected (temporary end flag LV_ERR_TMP_FSD_LAM_LIM is 1) or the timer T_SUM_END_DIAG_WIN_FSD_LAM_LIM[i] has reached the calibration threshold.

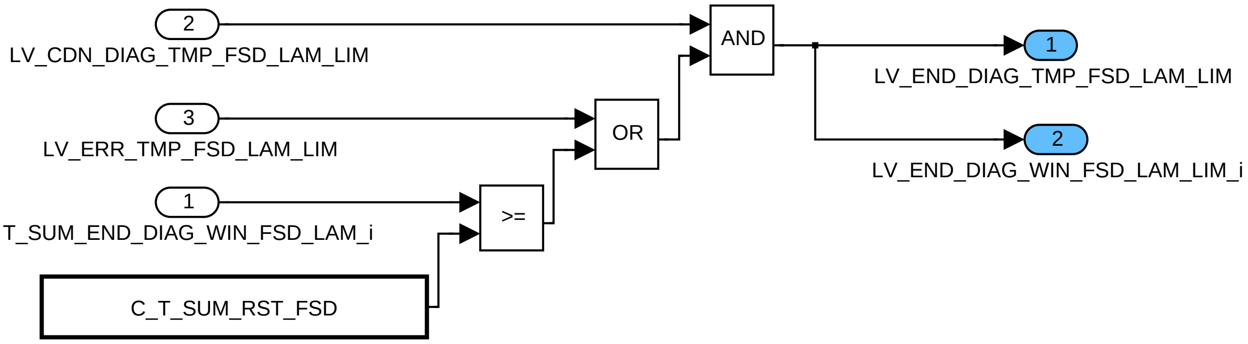


Figure 258 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD_LAM_LIM/ SET_END_DIAG_LAM_LIM

Manage end flags for lambda adaptation high range monitoring

First the end flag timers are calculated and then the conditions to set the end flags are evaluated.

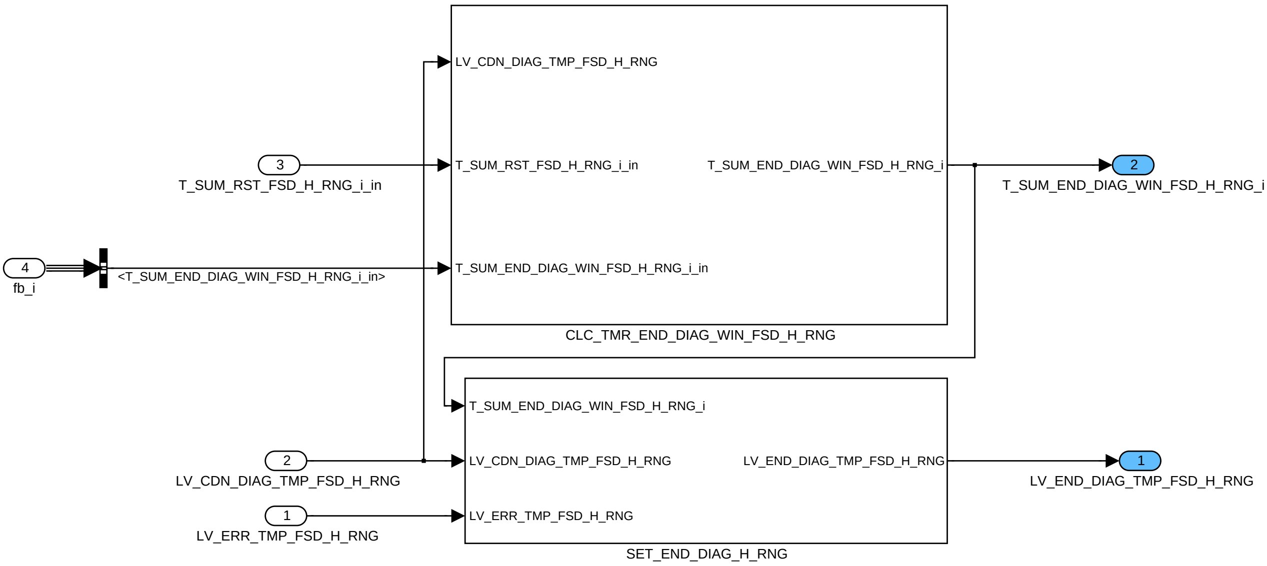


Figure 259 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD_H_RNG

Calculate end of diagnosis timers for lambda adaptation high range monitoring

If the conditions are not fulfilled the timer T_SUM_END_DIAG_WIN_FSD_H_RNG[i] remains unchanged.

The end timer is reset (to one sample step) either if the calibration threshold is reached or when the reset timer was started to be incremented.

In all other cases the timer is incremented.

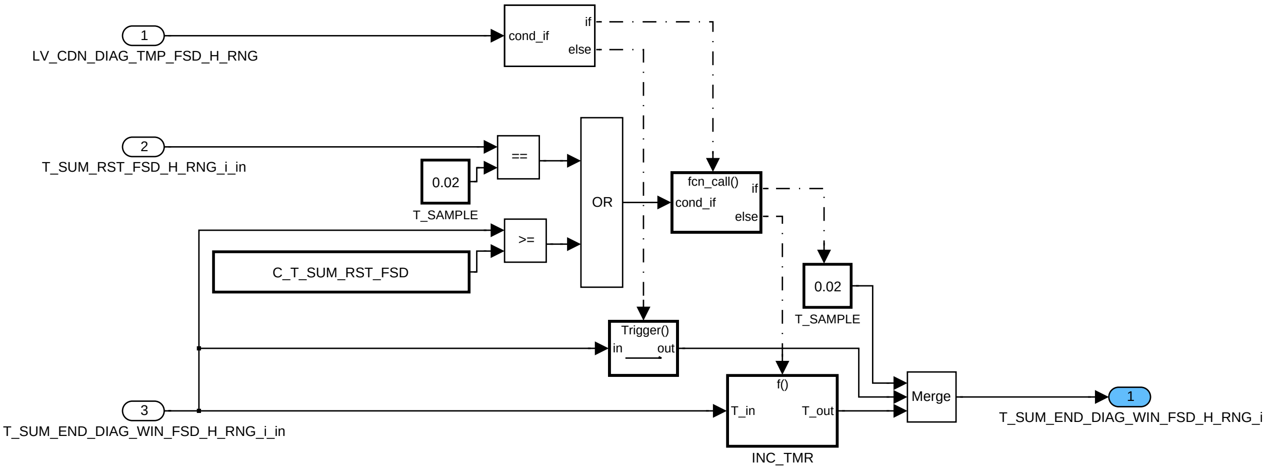


Figure 260 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD_H_RNG/ CLC_TMR_END_DIAG_WIN_FSD_H_RNG

Increment timer

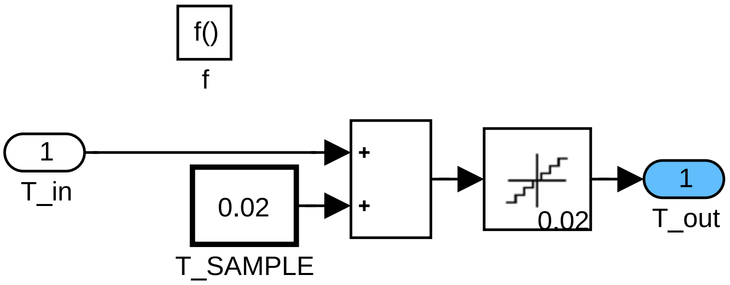


Figure 261 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD_H_RNG/ CLC_TMR_END_DIAG_WIN_FSD_H_RNG/ INC_TMR

Set end of diagnosis for lambda adaptation high range monitoring

The condition to set the temporary end flag LV_END_DIAG_TMP_FSD_H_RNG are: Diagnostic condition must be fulfilled and either an error was detected (temporary end flag LV_ERR_TMP_FSD_H_RNG is 1) or the timer T_SUM_END_DIAG_WIN_FSD_H_RNG[i] has reached the calibration threshold.

If LV_END_DIAG_TMP_FSD_LAM_LIM is set once, the corresponding error flag of the error management remain set for the rest of the driving cycle.

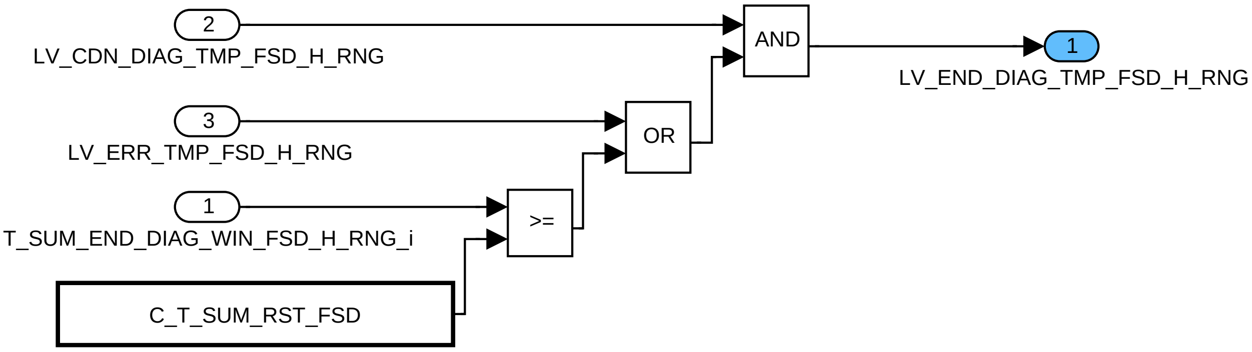


Figure 262 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ DIAG/ MNG_END_DIAG/ MNG_END_DIAG_FSD_H_RNG/ SET_END_DIAG_H_RNG

Interface to error management

For each failure location a separate ACTION is called. By means of the temporary condition and error flags and the temporary error symptom all relevant information is transmitted to the error management.

The output of each ACTION is the error information for each failure location.

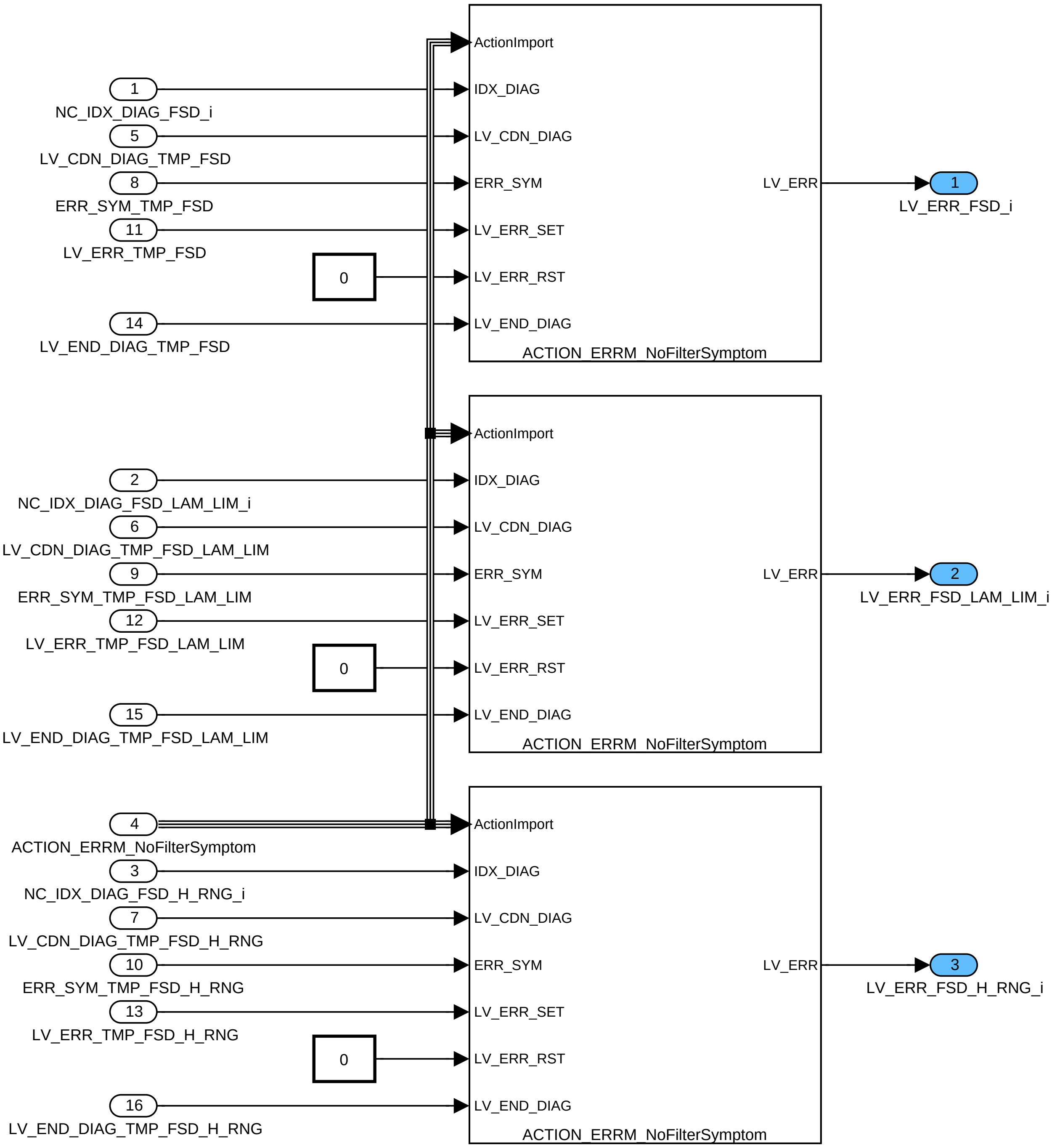


Figure 263 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ CBK_EX_SPC/ ERRM_IF

Read bank specific output values of last recurrence

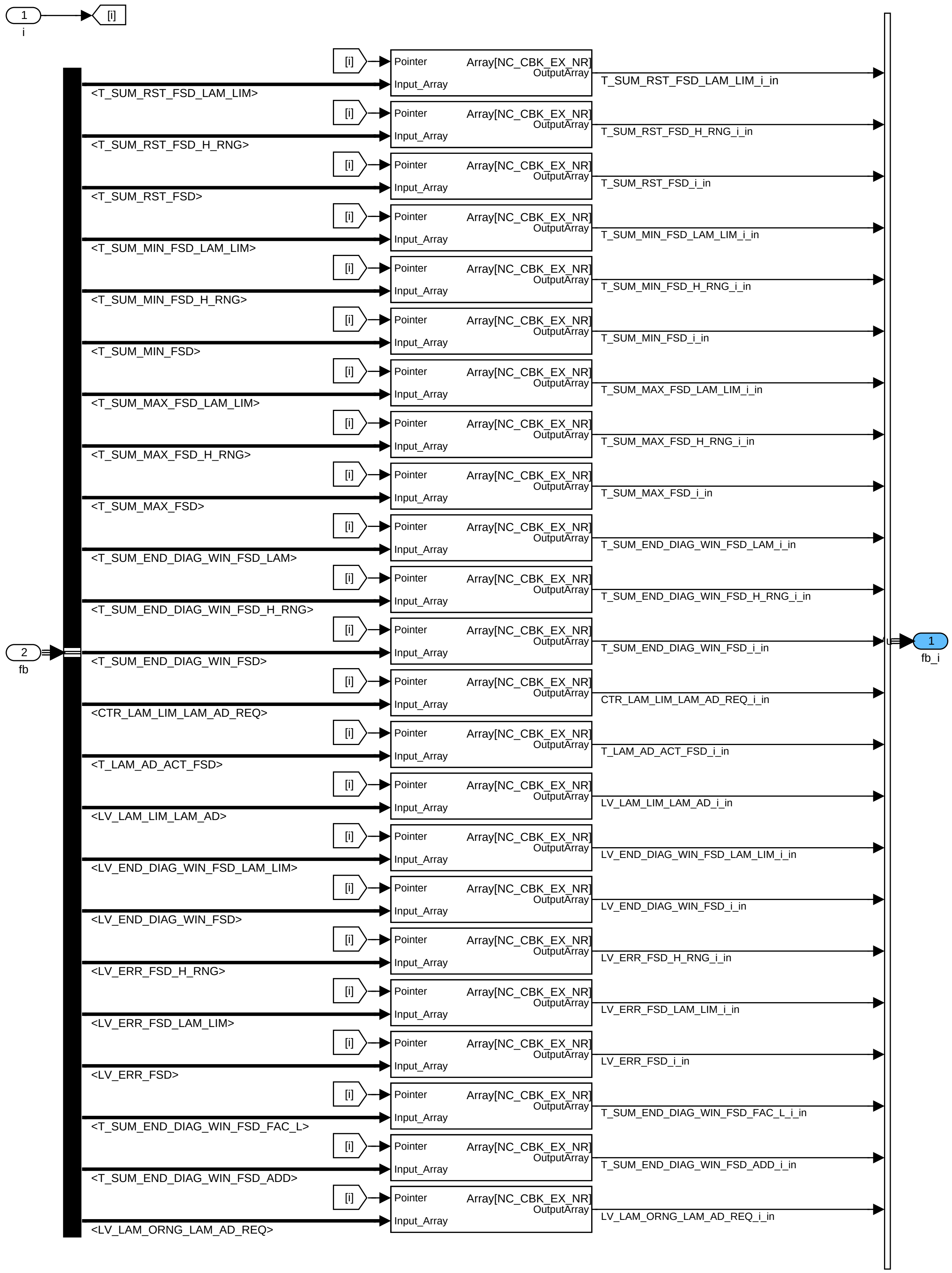


Figure 264 LACO_FCTDGFSD0/ operate_20MS/ CBK_EX_MNG/ SIG_CBK_MNG

46.17.1.3 Visualization of ERRM data

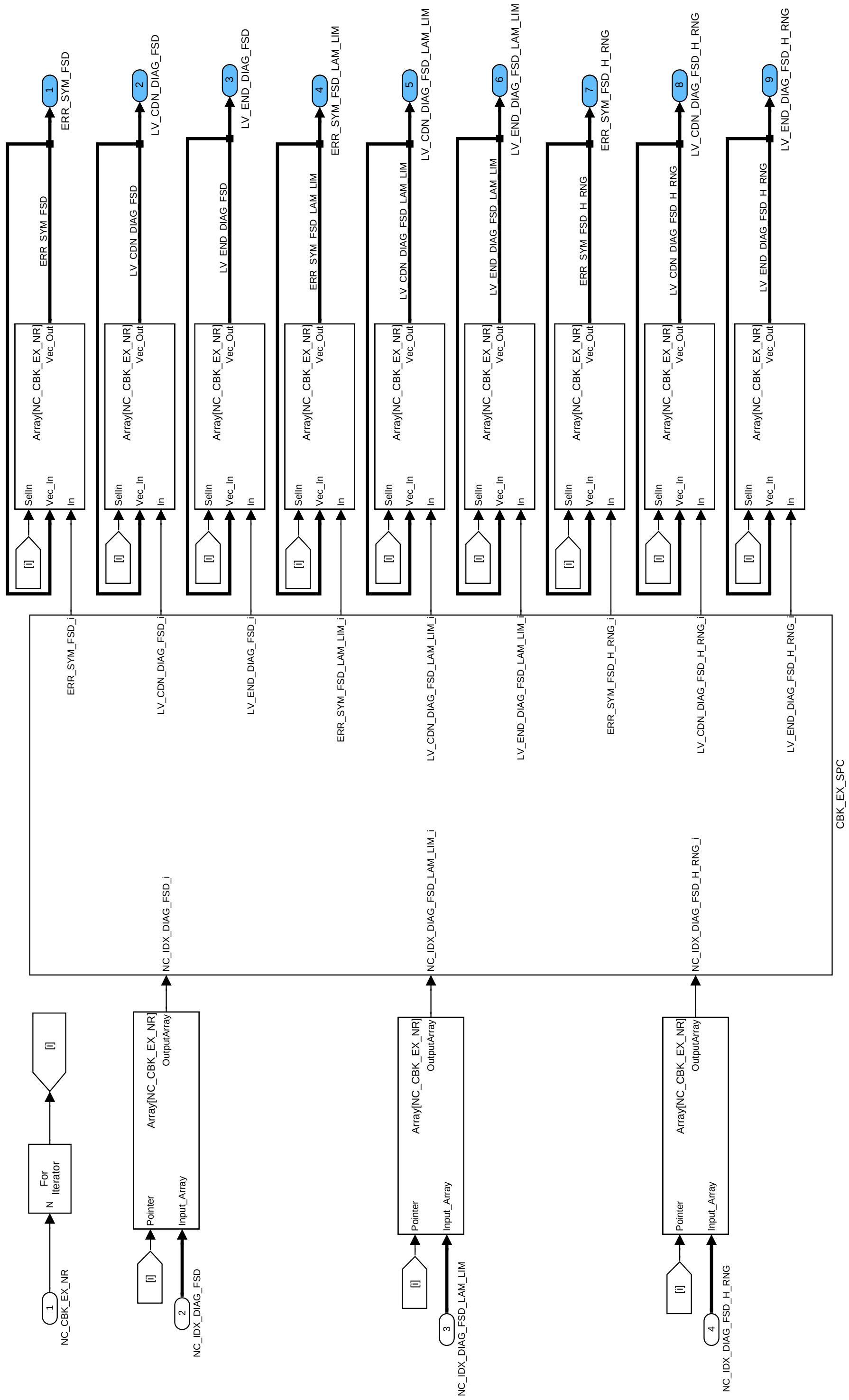


Figure 265 LACO_FCTDGFSD0/ visualizationERRMdata

Pointer on ERRM data

The error management variables, that are only visible (no output), are visualized by means of a pointer on the respective entry in the ERRM array structure.

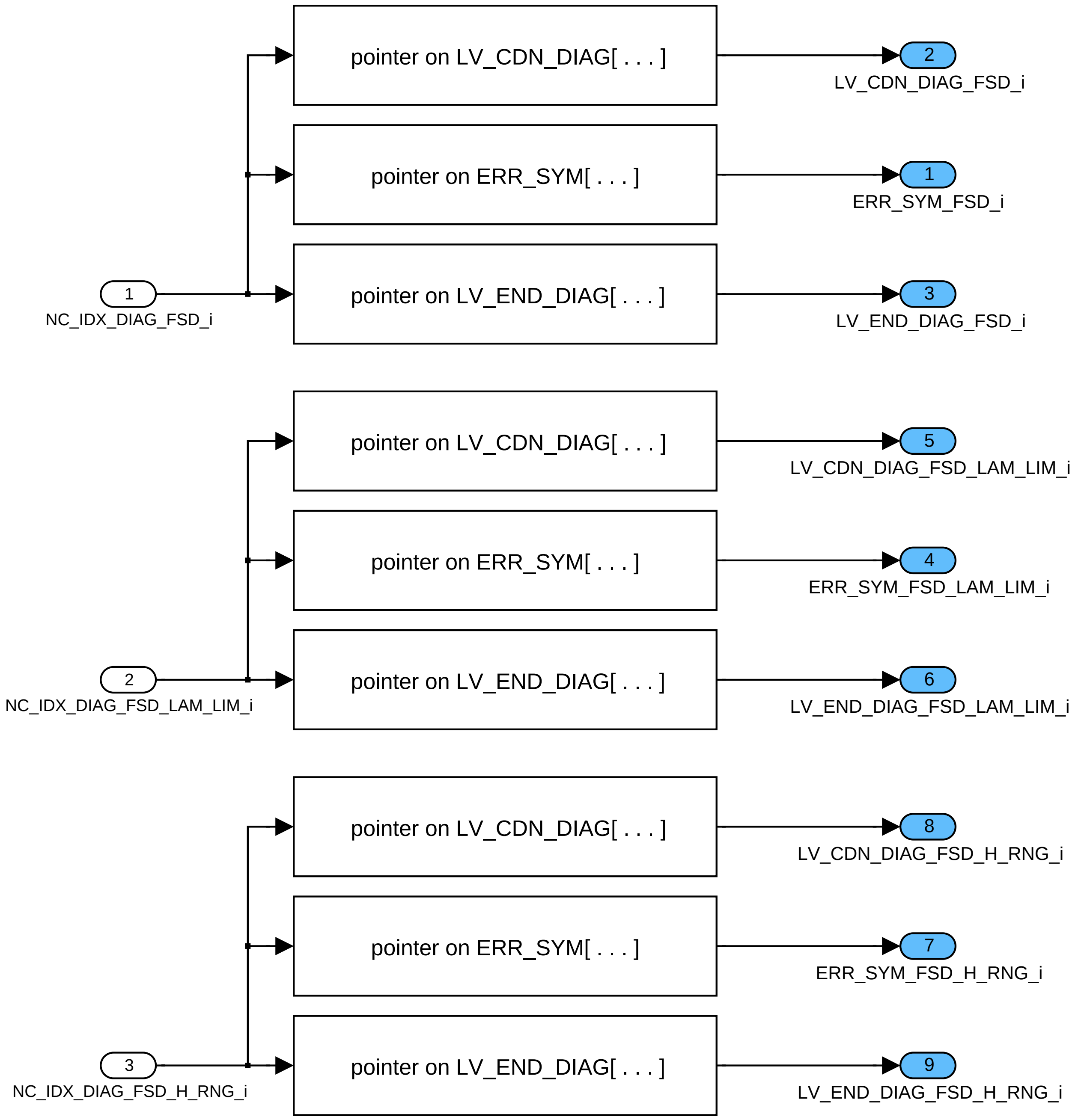


Figure 266 LACO_FCTDGFSD0/ visualizationERRMdata/ CBK_EX_SPC

46.18 Application incidences for fuel system diagnosis

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_FAC_L_RNG_LIM_MAX_FSD [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag indicating maximum limitation of lambda adaptation used by fuel system diagnosis					
LV_FAC_L_RNG_LIM_MIN_FSD [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag indicating minimum limitation of lambda adaptation used by fuel system diagnosis					
LV_INH_DIAG_FSD [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Inhibition flag for fuel system diagnosis					
LV_INH_DIAG_FSD_CP [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Inhibition flag for fuel system diagnosis due to internal states of canister purge function					
LV_INH_DIAG_FSD_IVVT_TMP [NC_CBK_EX_NR]	-	0... 1H	0... 1	1	[-]
Temporary flag that can inhibit the fuel system diagnosis due to IVVT errrors					
LV_INH_DIAG_FSD_NOT_CBK_SPC	-	0... 1H	0... 1	1	[-]
Temporary flag that can inhibit the fuel system diagnosis due not exhaust bank specific errrors					
LV_MFF_ADD_LIM_MAX_FSD [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
flag indicating top limitation of lambda adaptation for FSD module					
LV_MFF_ADD_LIM_MIN_FSD [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
flag indicating bottom limitation of lambda adaptation for FSD module					
T_FSD_SET_END_DIAG [NC_CBK_EX_NR]	V	0... FFFFH	0... 1310.7	0.02	[s]
Timer to force end of diagnostic for FSD					

Input Data:

LV_FAC_LAM_LIM_MIN [NC_CBK_EX_NR]	LV_FAC_LAM_LIM_MAX [NC_CBK_EX_NR]	LV_ERR_EL_CPS	LV_ERR_MAF
LV_ERR_MAP	LV_ERR_TCO	LV_ERR_TPS	LV_ERR_VCV
LV_MIS_STATE_A	LV_MIS_STATE_B1	LV_MIS_STATE_B4	CL_MMV
STATE_CP	LV_CP_CLOSE_ACT	LV_ERR_MEC_IVVT_EX [NC_NR_CBK_IVVT]	LV_ERR_MEC_IVVT_IN [NC_NR_CBK_IVVT]
LV_ERR_SLV_IVVT_EX	LV_ERR_SLV_IVVT_IN	NC_NR_CBK_IVVT	NLC_IVVT_EX
NLC_IVVT_IN	LV_ACT_SA_EOL	LV_LIH_ERR_CRK	NC_CBK_EX_NR
LV_ERR_MAP_PLAUS	LV_ERR_DIAGCPS	LV_ERR_MAP_DIP_SENS	LV_ERR_MAP_DIP_PLAUS
LV_ERR_MAP_DIP_SHIFT	LV_ERR_LS_UP [NC_CBK_EX_NR]	LV_ERR_DYN_VLD_LS_UP [NC_CBK_EX_NR]	LV_ERR_AMP
LV_ERR_AMP_PLAUS	LV_ERR_DELTA_I_LAM [NC_CBK_EX_NR]	LV_ERR_VLS_DOWN_DIF [NC_CBK_EX_NR]	LV_ERR_FUP
LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_AMP_PLAUS_CUS	LV_ERR_FUP_ORNG
LV_ERR_FUP_ST	LV_ERR_TTIP_MES_LSH_UP [NC_CBK_EX_NR]	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_CAM_CUS
LV_FAC_L_RNG_LIM_MIN_EXT_LAM_AD [NC_CBK_EX_NR]	LV_FAC_L_RNG_LIM_MAX_EXT_LAM_AD [NC_CBK_EX_NR]	LV_FAC_L_RNG_LIM_MIN_LAM_AD [NC_CBK_EX_NR]	LV_FAC_L_RNG_LIM_MAX_LAM_AD [NC_CBK_EX_NR]
LV_LAM_AD_EXT	LV_ERR_MAP_TPS_PLAUS	LV_FSD_ACT [NC_CBK_EX_NR]	STATE_LAM_AD [NC_CBK_EX_NR]
NC_IDX_DIAG_FSD [NC_CBK_EX_NR]	LV_MFF_ADD_LIM_MIN_LAM_AD [NC_CBK_EX_NR]	LV_MFF_ADD_LIM_MAX_LAM_AD [NC_CBK_EX_NR]	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CL_MMV_MAX_FSD	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Maximum value for CL_MMV to permit FSD					
C_T_MAX_FSD_SET_END_DIAG	1	0... FFFFH	0... 1310.7	0.02	[s]
Timer threshold to force end of diagnostic for FSD					
LC_FSD_SET_END_DIAG_ENA	1	0... 1H	0... 1	1	[-]
Calibration flag to enable forced setting of end of diagnostic for FSD					
LC_INH_FSD_MAN_DEAC	1	0... 1H	0... 1	1	[-]
Manual deactivation of fuel system diagnosis inhibition					
LC_MIL_ERR_SLV_IVVT_IN	1	0... 1H	0... 1	1	[-]
LV_ERR_SLV_IVVT_IN switches the malfunction indication light on					

Import Actions:

ACTION_ERRM_GetLvEndDiag(**IN** <IDX_DIAG>, **OUT** <LV_END_DIAG>)
ACTION_ERRM_NoFilterSymptom(**IN** <IDX_DIAG>, **IN** <LV_CDN_DIAG>, **IN** <ERR_SYM>, **IN** <LV_ERR_SET>, **IN** <LV_ERR_RST>, **IN** <LV_END_DIAG>, **OUT** <LV_ERR>)

General Information

This function manages the inhibition of the fuel system diagnosis (FSD) due to system errors and canister purge states.

NC_CBK_EX_NR defines the number of exhaust banks.
For vector elements the variable extension _i is used in the model instead of [i] as found in the textual description.
Configuration for diagnostic symptoms:

Diagnostic FSD[i]	Symptom description	Symptom	Filter type
Fuel System Diagnosis for lambda adaptation monitoring – relevant to malfunction indication light	*maximum limit of additive adaptation value reached*	*SYM_0*	*NO*
	minimum limit of additive adaptation value reached	*SYM_1*	
	maximum limit of multiplicative adaptation value (lower area) reached	*SYM_2*	
	minimum limit of multiplicative adaptation value (lower area) reached	*SYM_3*	

Diagnostic FSD_LAM_LIM[i]	Symptom description	Symptom	Filter type
Fuel System Diagnosis in area where lambda controller output is concerned - relevant to malfunction indication light	*lambda control in dead stop (upper limit)*	*SYM_0*	*NO*
	lambda control in dead stop (lower limit)	*SYM_1*	

Diagnostic FSD_H_RNG[i]	**Symptom description**	**Symptom**	**Filter type**
Fuel System Diagnosis in upper multiplicative adaptation learning area	*maximum limit of multiplicative adaptation value (upper area) reached*	*SYM_0*	*NO*
	minimum limit of multiplicative adaptation value (upper area) reached	*SYM_1*	

Application Conditions

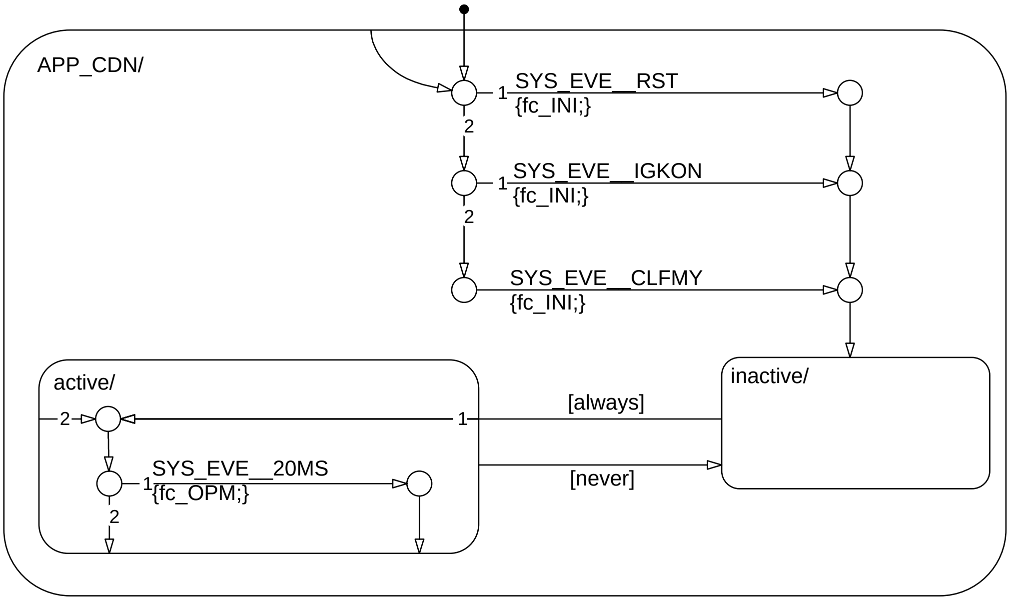
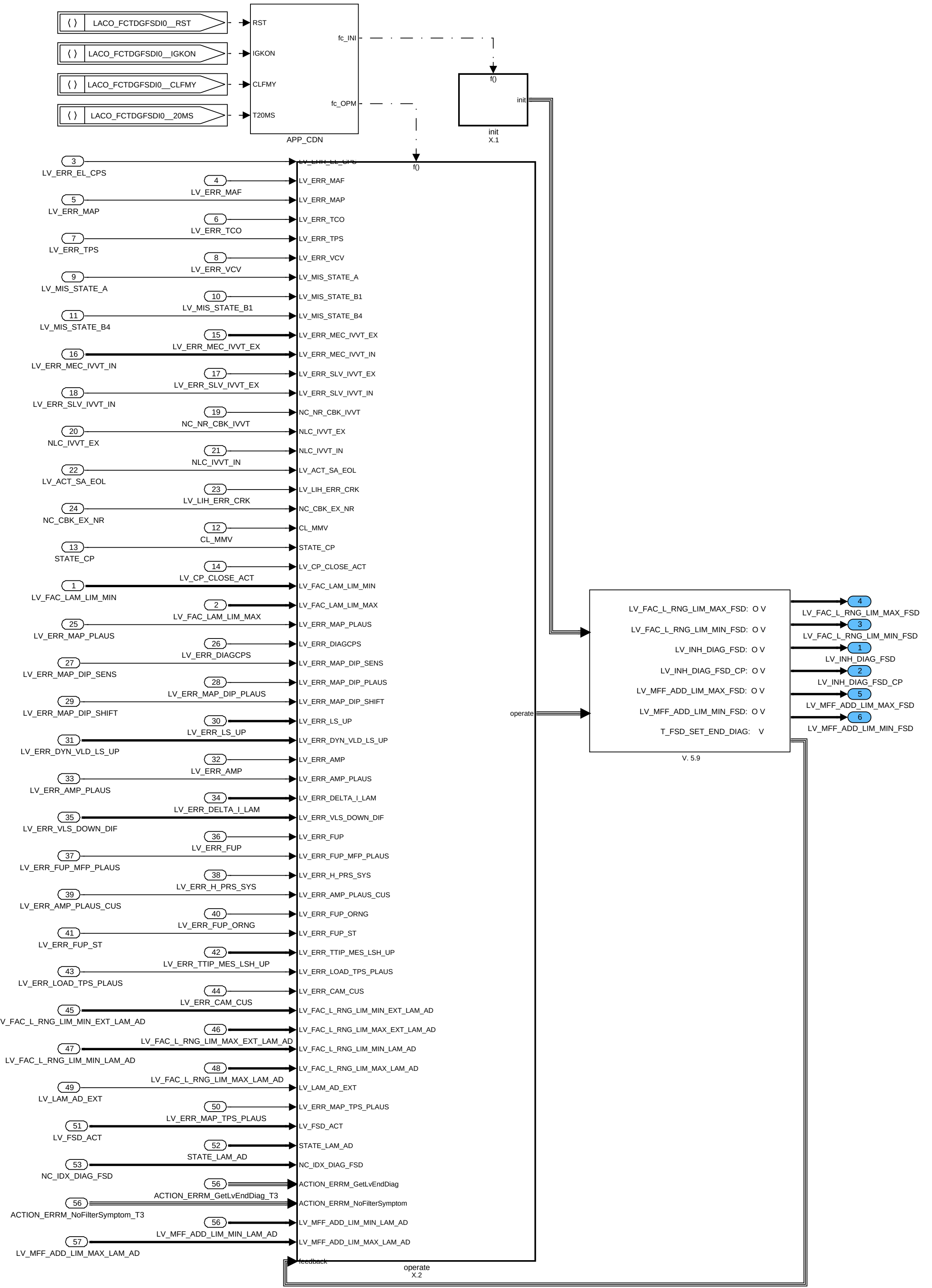


Figure 267:



SDA_SRS / SDA V 5.2.5 18−May−2007

Figure 268:

46.18.1 Initialization

The inhibition flags LV_INH_DIAG_FSD[i] and LV_INH_DIAG_FSD_CP[i] are both set to 1. All other variables are initialzed with 0.

46.18.1.1 Calculation of initialization

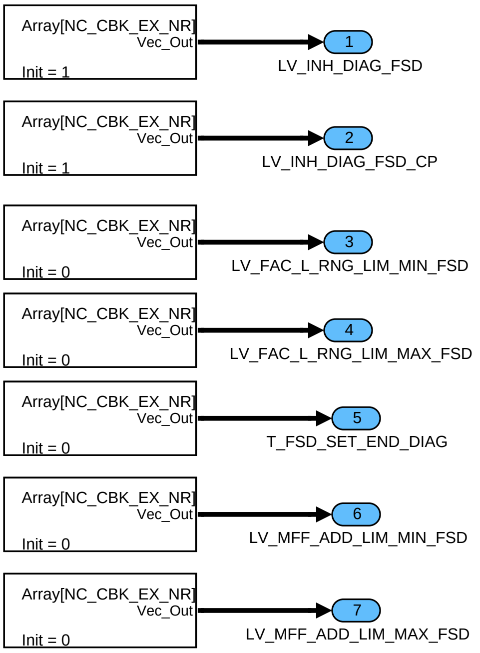


Figure 269:

46.18.2 Formula section

46.18.2.1 Inhibition of fuel system diagnosis

LV_INH_DIAG_FSD[i] represents the influence of errors deactivating the fuel system diagnosis. All exhaust bank independent errors are summarized in the temporary and not visible variable LV_INH_DIAG_FSD_NOT_CBK_SPC.
The temporary and not visible inhibition flag LV_INH_DIAG_FSD_IVVT_TMP is set if an IVVT error (intake or exhaust) affects the relevant exhaust bank for FSD.
If LC_INH_FSD_MAN_DEAC is set to 1 fuel system diagnosis is not inhibited by a present error.

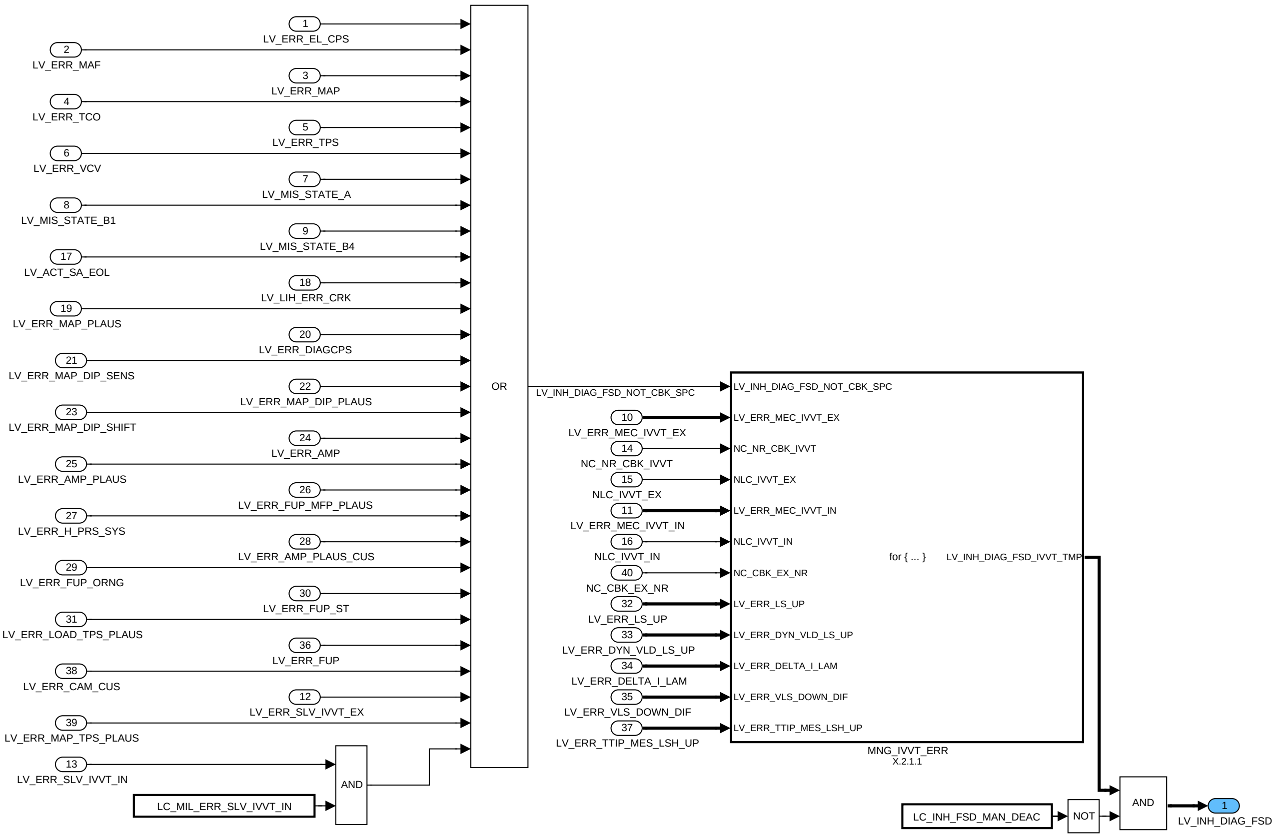


Figure 270:

46.18.2.1.1 Exhaust bank specific inhibition due to IVVT errors

The temporary and not visible inhibition flags LV_INH_DIAG_FSD_IVVT_IN/EX_TMP are set bank specifically depending on whether IVVT for exhaust and/or intake bank is present or not. If not present these flags are 0. If the IVVT banks and the FSD exhaust banks are identical then bank specific inhibition is possible.

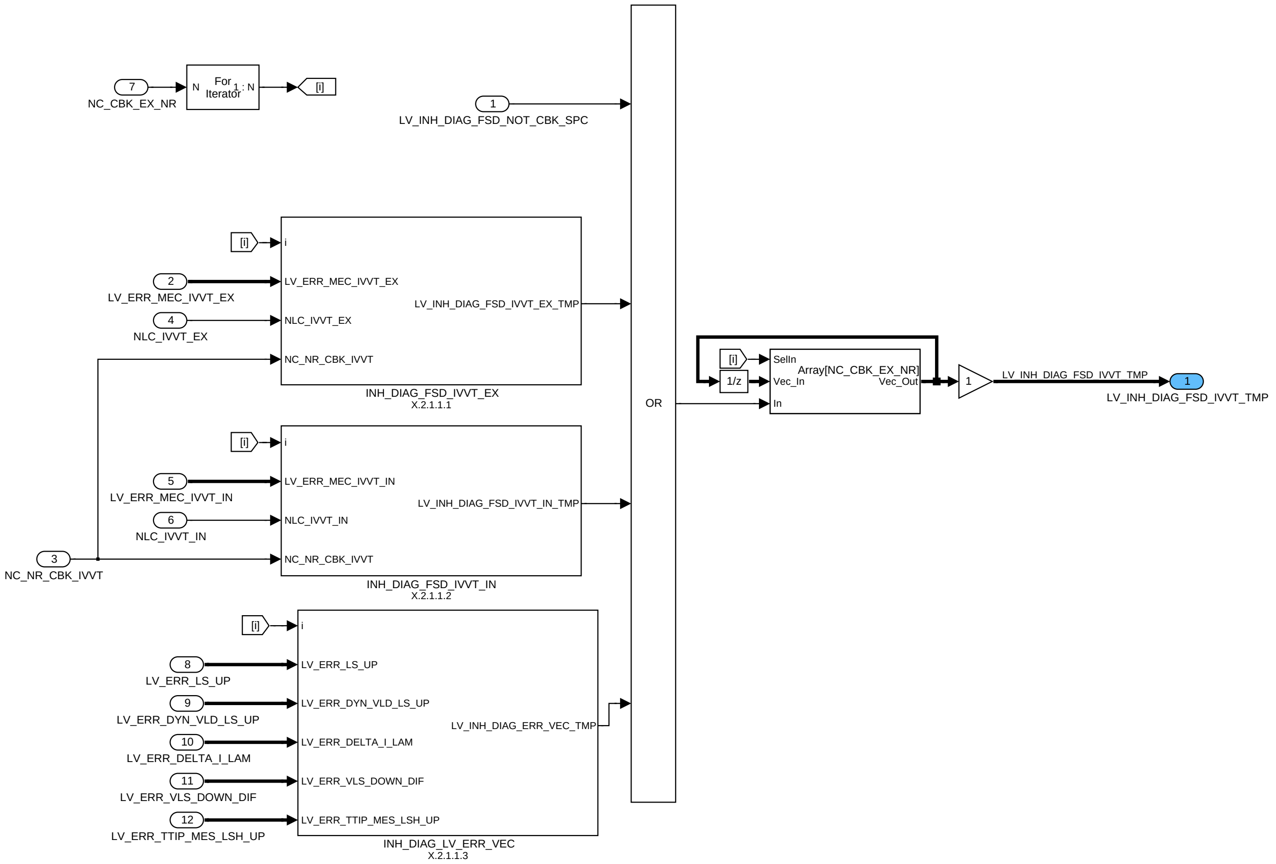


Figure 271:

46.18.2.1.1.1 Manage IVVT errors for exhaust bank

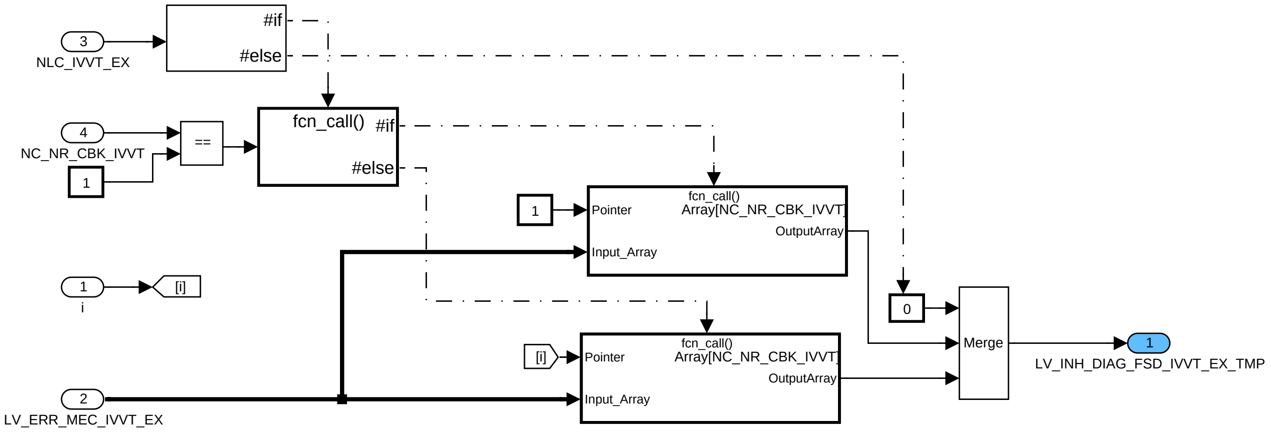


Figure 272:

46.18.2.1.1.2 Manage IVVT errors for intake bank

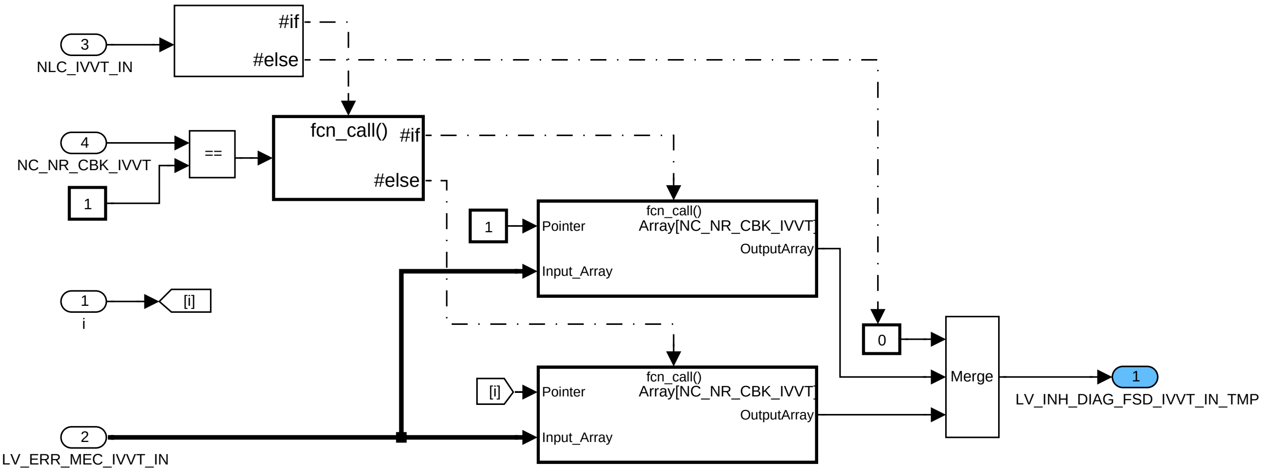


Figure 273:

46.18.2.1.1.3 Exhaust bank specific inhibition due to error vectors

LV_INH_DIAG_FSD_IVVT_TMP is set bank specific according to the error vectors.

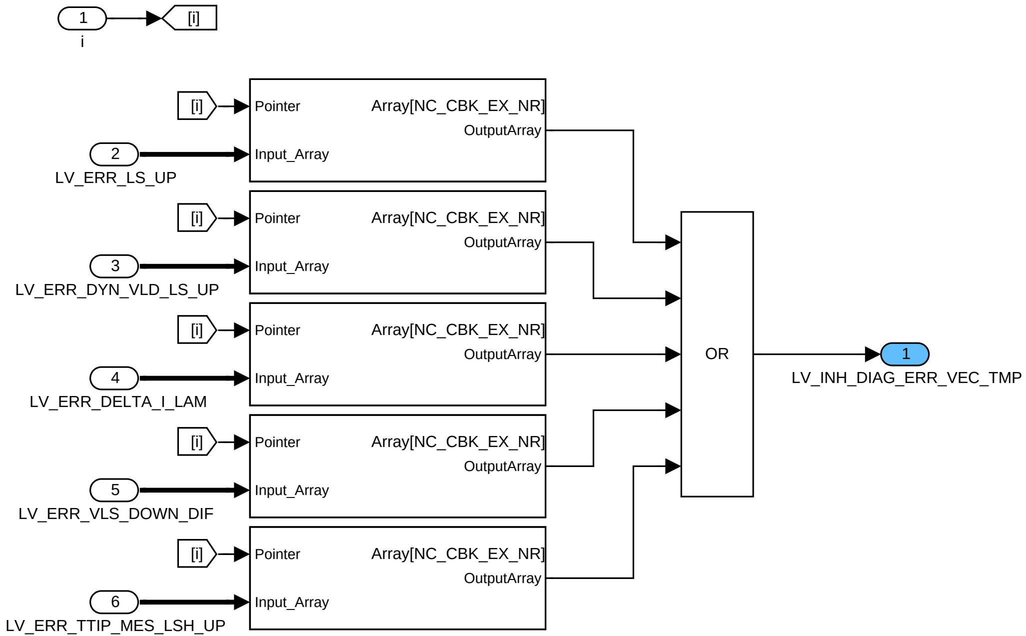


Figure 274:

46.18.2.2 Inhibition of fuel system diagnosis due to canister purge

46.18.2.2.1 Exhaust bank specific functionality

LV_INH_DIAG_FSD_CP[i] respects the influence of canister purge on FSD. The inhibition flag is not set when the canister purge valve is closed, in MAX PURGE at low charcoal canister load and in MIN_PURGE when the lambda controller is at one of its limits.

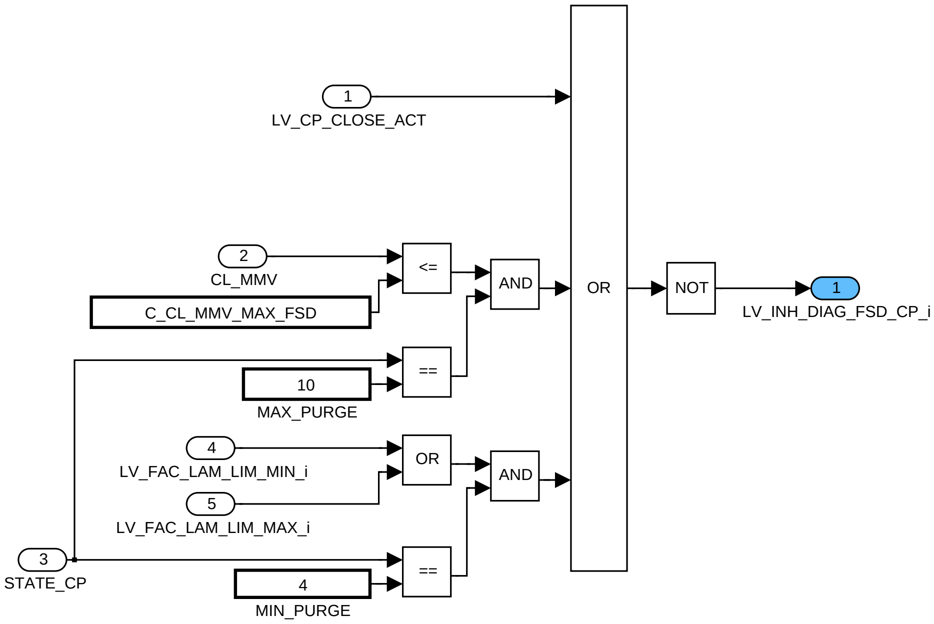


Figure 275:

46.18.2.3 Determination of limitation flags for lambda adaption (in- / outside LACO)

Determination if indicator flag for minimum and maximum limitation of lambda adaptation used by fuel system diagnosis is calculated inside of outside of LACO, according to LV_LAM_AD_EXT.

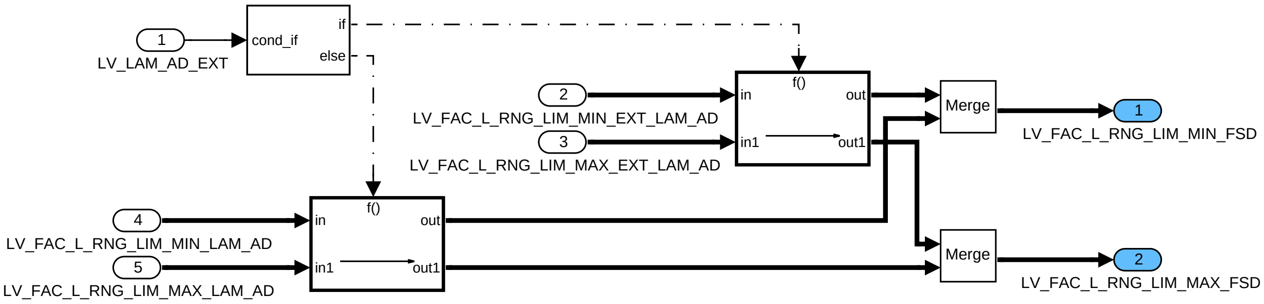


Figure 276:

46.18.2.4 Force FSD end diag flag to 1

46.18.2.4.1 Exhaust bank specific functionality

46.18.2.4.1.1 Increment timer

The timer T_SET_END_DIAG_FSD[i] that forces FSD end of diagnostic flag to 1 is incremented if general FSD activation conditions are fulfilled and engine is running in one of the lambda adaptation fields independent whether adaptation is active or not.

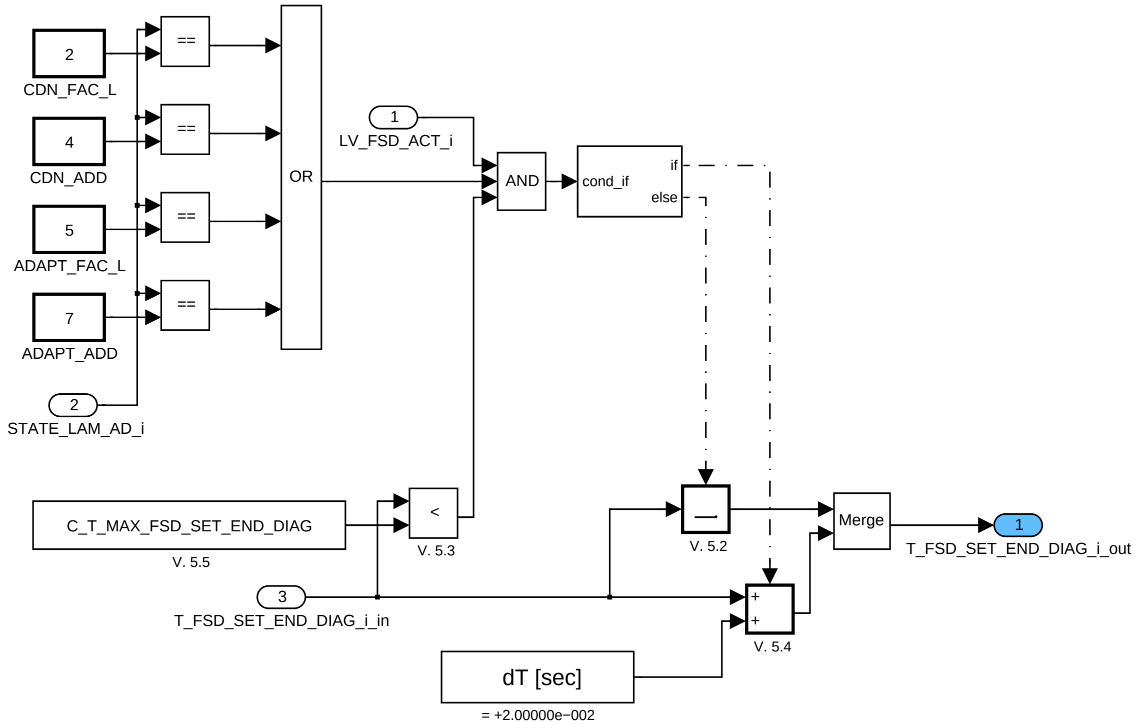


Figure 277:

46.18.2.4.1.2 Timer elapsed

As soon as the timer T_SET_END_DIAG_FSD[i] reaches a calibration threshold the error management is called to set the end of diagnosis information (if not yet set).
This feature can be disabled by the calibration flag LC_FSD_SET_END_DIAG_ENA.

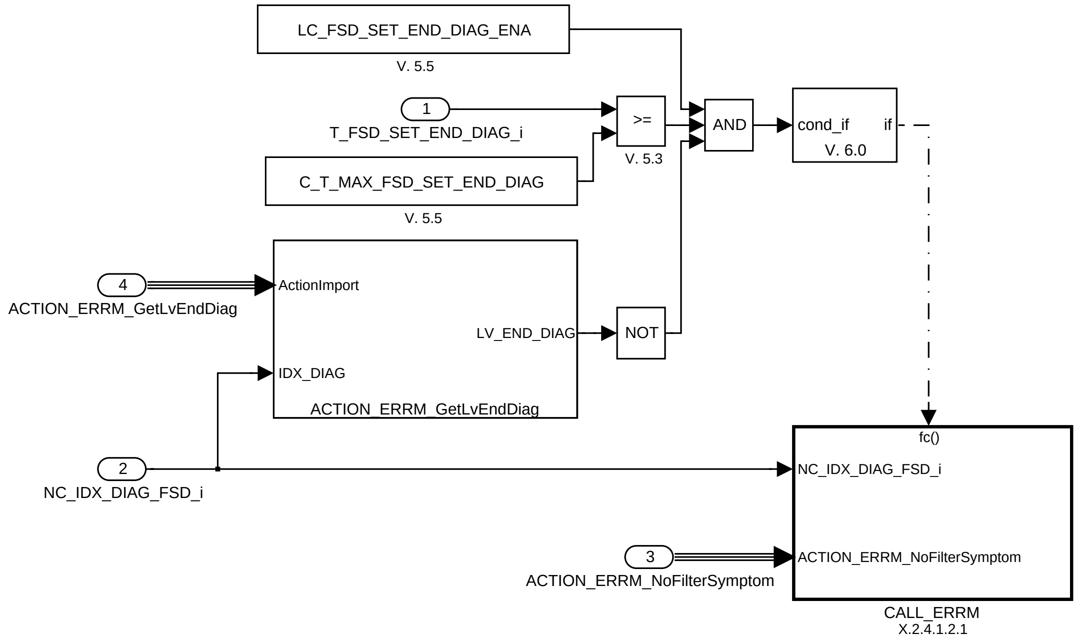


Figure 278:

46.18.2.4.1.2.1 Call Error Management

Error management is called with action NoFilterSymptom in order to set end of FSD diagnostic information.

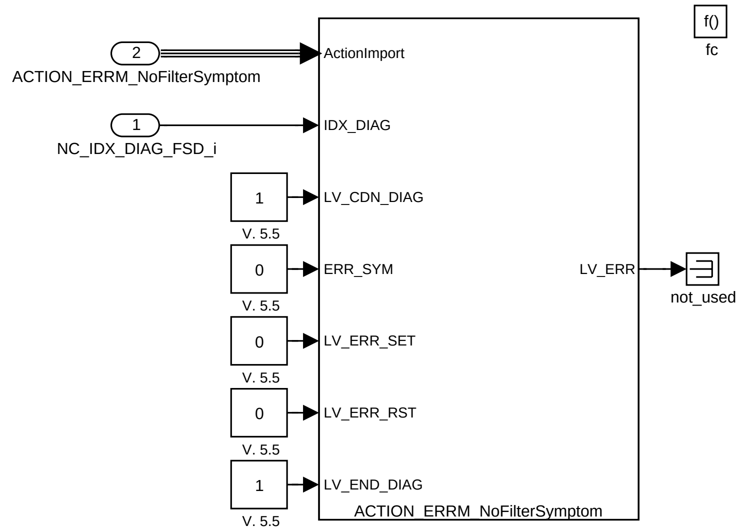


Figure 279:

46.18.2.5 Calculation of LV_MFF_ADD_LIM_MIN_FSD and LV_MFF_ADD_LIM_MAX_FSD

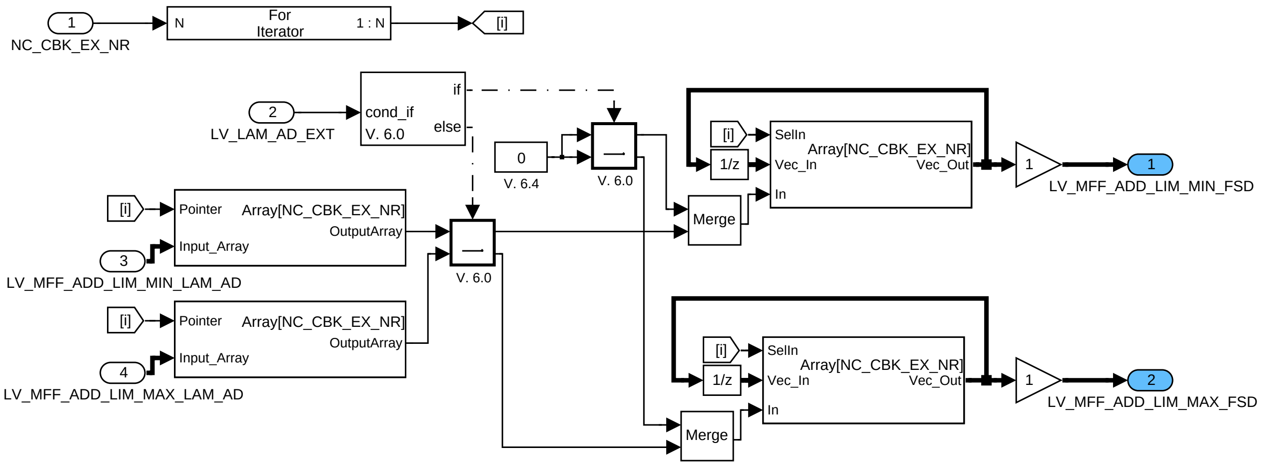


Figure 280:

46.19 Dynamic Fuel Trim

The trim controller causes a shift of the WRAF sensor signal (upstream of the catalyst) or/and a shift of the lambda set point (in opposite direction). The output components (P share, I share, D-share, the sum of all or the adaptation value) of the trim controller can be linked in the application incidence for the lambda controller to corresponding inputs of the lambda controller. A compensation of ageing effects of the WRAF sensor as well as a better observance of the catalyst window is reached because of these shifts.

The control value is composed of P, I and D shares. All are determined depending on the trim control sensor signal (binary signal). The difference of the sensor voltage from set point value is the basic characteristic. The set point value can be applied depending on the operating point, so that the dynamic lambda can be adjusted according to the operating range.

46.19.1 Calibration interfering functions

- Fuel path (injection realisation) and air path (intake manifold model)
- Heater functions of upstream and downstream sensors
- Linear Lambda Control
- Forced Lambda stimulation
- Catalyst Diagnosis

46.19.2 Calibration flowchart

The sequence is also described by the order in Calibration method.

46.19.3 Calibration method

The binary sensor signal (VLS_DOWN[i]) should be measured during emission tests. If the cycle show emission problems with NOx, and a correlation with VLS_DOWN[i] can be observed (VLS_DOWN[i] < 450 mV when NOx ↑) , then the P share intervention for VLS_DIF_LAM_ADJ[i] > 0 can be increased. When HC ↑ and/or CO ↑ during emission test, then the P share intervention for VLS_DIF_LAM_ADJ[i] > 0 can be decreased.

The set point for the sensor signal voltage should be evaluated by measuring the catalyst efficiency in all operational areas. The downstream sensor signal voltage showing the best catalyst efficiency should be used as set point in that operational point.

46.19.3.1 Common Activation Conditions

C_TEMP_CAT_MIN_LAM_ADJ_ACT	The constant represents the minimum threshold for of the Catalyst temperature for activation of the trim controller. **(e.g. 300 °C)**
C_EFF_CAT_DIAG_MAX_LAM_ADJ_ACT	The constant is the threshold for the EFF_CAT_DIAG above which the trim controller will be deactivated. **(e.g. 1)**
C_MAF_INT_LAM_P_ADJ_ACT_FAST[NC_CBK_EX_NR]	The constant is the integral threshold for MAF for reactivation of the trim controller (P-share) from inactive state. **(e.g. 3 g)**

C_MAF_INT_LAM_P_ADJ_ACT_SLOW	The constant is the integral threshold for MAF for reactivation of the trim controller (P-share) after Lambda set point violation, Lambda controller at hard limit or open-loop operation of Lambda controller. **(e.g. 5 g)**
C_MAF_INT_LAM_ADJ_P_ACT_PURGE	The constant is the integral threshold for MAF for reactivation of the trim controller (P-share) after PUC. **(e.g. 10 g)**
C_MAF_INT_LAM_I_ADJ_ACT_FAST[NC_CBK_EX_NR]	The constant is the integral threshold for MAF for reactivation of the trim controller (I-share) from inactive state. **(e.g. 15 g)**
C_MAF_INT_LAM_I_ADJ_ACT_SLOW	The constant is the integral threshold for MAF for reactivation of the trim controller (I-share) after Lambda set point violation, Lambda controller at hard limit or open-loop operation of Lambda controller. **(e.g. 35 g)**
C_MAF_INT_LAM_ADJ_I_ACT_PURGE	The constant is the integral threshold for MAF for reactivation of the trim controller (I-share) after PUC. **(e.g. 50 g)**
C_MAF_INT_LAM_D_ADJ_ACT_FAST[NC_CBK_EX_NR]	The constant is the integral threshold for MAF for reactivation of the trim controller (D-share) from inactive state. **(e.g. 10 g)**
C_MAF_INT_LAM_D_ADJ_ACT_SLOW	The constant is the integral threshold for MAF for reactivation of the trim controller (D-share) after Lambda set point violation, Lambda controller at hard limit or open-loop operation of Lambda controller. **(e.g. 15 g)**
C_MAF_INT_LAM_ADJ_D_ACT_PURGE	The constant is the integral threshold for MAF for reactivation of the trim controller (D-share) after PUC. **(e.g. 25 g)**
C_TCO_LAM_ADJ_P_ACT_HST	Fast activation of trim controller P-share in case of hot start. If TCO is above this threshold and T_AST below limit (and all other conditions are fulfilled) the P-share is activated immediately. (**e.g. 75 °C**)
C_T_AST_LAM_ADJ_P_ACT_HST	Fast activation of trim controller P-share in case of hot start. If TCO is above threshold and T_AST below this limit (and all other conditions are fulfilled) the P-share is activated immediately. (**e.g. 20 s**)
C_LAMB_SP_MIN_LAM_ADJ_ACT	Minimum Lambda setpoint for overall trim controller activation. Defines setpoint range valid for Lambda eq. 1 operation. (**min. value of IP_LAMB_BAS**) Full load enrichment setpoints in IP_LAMB_BAS may no be used as minimal thresholds for trim controller activation.
C_LAMB_SP_MAX_LAM_ADJ_ACT	Maximum Lambda setpoint for overall trim controller activation. Defines setpoint range valid for Lambda eq. 1 operation. (**max. value of IP_LAMB_BAS**)

46.19.3.2 *Lambda Voltage Setpoint*

IP_VLS_SP_LAM_ADJ[NC_CBK_EX_NR]	The setpoint of the sensor signal voltage should be evaluated measuring the Catalyst Efficiency for the whole operating range. At each operating point the sensor signal voltage with the best catalyst efficiency should be chosen as the setpoint. IP_VLS_SP_LAM_ADJ[i] is an N-MAF-dependent map. **(e.g. 650 mV for all points)**

IP_VLS_SP_DELTA_LAM_ADJ	Shift of the downstream signal setpoint based on EFF_CAT_DIAG to take into account stringent NOx increase with catalyst ageing. **(e.g. 0 for all points)**
IP_VLS_DELTA_LAM_ADJ_CAT_DIAG	Map is defined in Appl. Inc. Offset of VLS_DOWN_MV because of non-linearity of the binary sensor in case of break through **(e.g. 0 for all points)**
IP_CRLC_VLS_LAM_ADJ_CAT_DIAG	Map is defined in Appl. Inc. Aim of calculation is, to filter the oscillations of the VLS_DOWN signal to remain in the plateau of P-share resp. I-share during catalyst diagnosis (filtered amplitude is 10% of VLS_DOWN) (see Figure 1) **(IP_CRLC_VLS_LAM_ADJ_CAT_DIAG = 2*π*T_SAMPLE/(10*T_SUM_AFL_AFR_LAM+2*π*T_SAMPLE))**

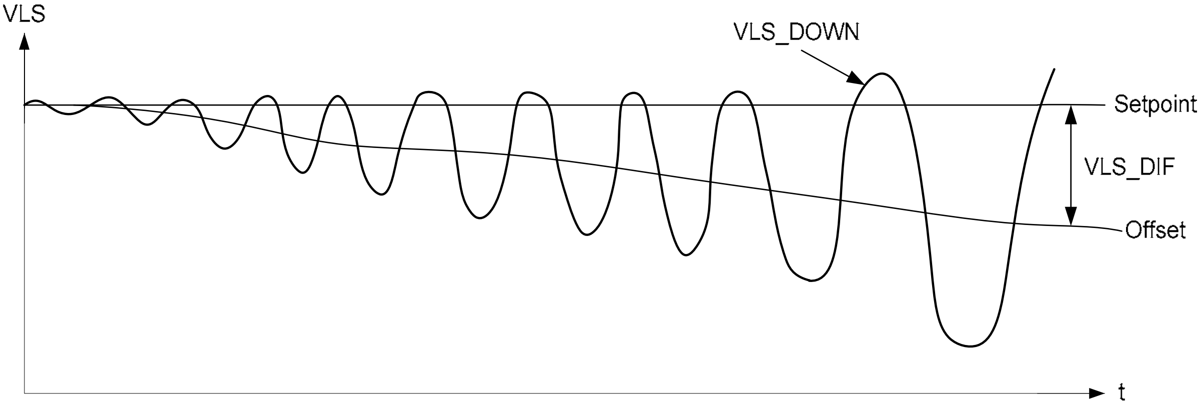


Figure 1: mean value shift

46.19.3.3 *P-share Specific Activation and Calculation*

C_MAF_MIN_LAM_ADJ_P_ACT	This constant forms the minimum MAF threshold for the activation of the Proportional component of the trim controller. **(e.g. 65 mg/stk)**
C_MAF_MAX_LAM_ADJ_P_ACT	This constant forms the maximum MAF threshold for the activation of the Proportional component of the trim controller. **(e.g. 501 mg/stk)**
C_N_MIN_LAM_ADJ_P_ACT	This constant forms the minimum Engine Speed threshold for the activation of the Proportional component of the trim controller. **(e.g. 544 rpm)**
C_N_MAX_LAM_ADJ_P_ACT	This constant forms the maximum Engine Speed threshold for the activation of the Proportional component of the trim controller. **(e.g. 4224 rpm)**
IP_LAMB_DELTA_P_LAM_ADJ	This forms the Trim Proportional Correction based on the voltage error of the Downstream sensor from the Setpoint. The benefit of the P component is rejection of transient disturbance to reduce NOx emission. The dependency is only on VLS_DIF_LAM_ADJ. **(e.g. see table below)**
IP_FAC_LAMB_DELTA_P_LAM_ADJ [NC_CBK_EX_NR]	This is a weighting factor for the P component based on MAF. **(e.g. see table below)**

IP_FAC_ADD_LAMB_DELTA_P	This is a weighting factor for the P component based on EFF_CAT_DIAG. So the normal P operation can be fine tuned with the aging of the Catalyst. **(e.g. see table below)**
IP_LAMB_DELTA_P_LAM_ADJ_DIAG	This forms the Trim Proportional Correction based on the voltage error of the Downstream sensor from the Setpoint in the Cat Diagnosis phase. **(e.g. same as for IP_LAMB_DELTA_P_LAM_ADJ)**

LDPM_VLS_DIF_LAM_ADJ_1_LACO				*-0.21973*	*-0.18066*	*-0.13184*	*-0.07324*	*-0.04395*	*-0.02930*
IP_LAMB_DELTA_P_LAM_ADJ				-0.02197	-0.01501	-0.00891	-0.00488	-0.00043	0.00000
0.000	*0.029297*	*0.048828*	*0.068359*	*0.097656*	*0.170898*	*0.229492*	*0.288086*	*0.332031*	*0.371094*
0.00000	0.00000	0.00037	0.00409	0.00812	0.01300	0.01703	0.03003	0.04999	0.07501

LDPM_MAF_CYL_1_LACO	*20.000*	*35.000*	*60.000*	*90.000*	*150.000*	*230.000*
IP_FAC_LAMB_DELTA_P_LAM_ADJ [NC_CBK_EX_NR]	0.980	0.870	0.800	0.740	0.600	0.500

LDPM_EFF_CAT_DIAG_1_LACO	*0.000*	*0.055*	*0.180*	*0.297*	*0.484*	*0.578*	*0.766*	*1.000*
IP_FAC_ADD_LAMB_DELTA_P	0.996	0.914	0.750	0.648	0.508	0.375	0.285	0.230

46.19.3.4 I-share Specific Activation and Calculation

IP_MAF_MIN_LAM_ADJ_I_ACT	This constant forms the minimum MAF threshold for the activation of the Integral component of the trim controller. **(e.g. see table below)**
IP_MAF_MAX_LAM_ADJ_I_ACT	This constant forms the maximum MAF threshold for the activation of the Integral component of the trim controller. **(e.g. see table below)**
LC_LAMB_PLS_NOT_CHK_LAM_ADJ	Disable active forced Lambda stimulation as condition for trim controller I-share activation. (**e.g. 0**)
LC_LAM_ADJ_LDC_NOT_CHK	Disable limited dynamics as condition for activation of trim controller I-share activation. (**e.g. 0**)

Calibration of IP_MAF_MIN_LAM_ADJ_I_ACT and IP_MAF_MAX_LAM_ADJ_I_ACT:

The calculation of I component should be carried out in part load. Therefore, the minimum engine speed should be chosen over the nominal idle speed. A low load leads to a long response time of lambda sensor; therefore in the calibration low values of mass air flow should be avoided. The maximum value of allowed engine speed for calculation of I component should be less than the maximum occurred engine speed during the emission test cycles.

In order to define the range of MAF_CYL with constant adaptation value, drive several operating points. The calibration should be performed using a system with a 100K aged catalyst, which has a low oxygen storage capacity, and a slowed-down sensor (high lambda amplitude). For the whole range of allowed MAF_CYL the downstream lambda sensor signal should be stable.

LDPM_N_32_2_LACO	*544*	*608*	*1792*	*2400*	*3488*	*4000*

IP_MAF_MIN_LAM_ADJ_I_ACT	152.52	87.15	81.71	81.71	119.84	201.54

LDPM_N_32_2_LACO	*544*	*608*	*1792*	*2400*	*3488*	*4000*
IP_MAF_MAX_LAM_ADJ_I_ACT	98.05	190.65	408.53	408.53	392.19	201.54

IP_LAMB_DELTA_I_LAM_ADJ	This forms the Trim Integral Correction based on the voltage error of the Downstream sensor from the setpoint. **(e.g. see table below)**
IP_CRLC_LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]	This is a weighting factor for the I component based on MAF. **(e.g. see table below)**
IP_LAMB_DELTA_I_LAM_ADJ_DIAG	This forms the Trim Integral Correction based on the voltage error of the Downstream sensor from the Setpoint in the Cat Diagnosis phase. There is also a dependency on the EFF_CAT_DIAG. **(e.g. see table below)**

LDPM_VLS_DIF_LAM_ADJ_1_LACO	*-0.21973*	*-0.18066*	*-0.13184*	*-0.07324*	*-0.04395*	*-0.02930*
IP_LAMB_DELTA_I_LAM_ADJ	-0.000851	-0.000450	-0.000221	-0.000008	-0.000004	0.000

0,000	*0.029297*	*0.048828*	*0.068359*	*0.097656*	*0.170898*	*0.229492*	*0.288086*	*0.332031*	*0.371094*
0.000	0.000	0.000004	0.000008	0.000130	0.000225	0.000301	0.000401	0.000542	0.000790

LDPM_MAF_CYL_1_LACO	*20.000*	*35.000*	*60.000*	*90.000*	*150.000*	*230.000*
IP_CRLC_LAMB_DELTA_I_LAM_ADJ [NC_CBK_EX_NR]	0.002	0.003	0.004	0.005	0.007	0.008

LDPM_VLS_DIF_LAM_ADJ_2_LACO			*-0.220*	*-0.181*	*-0.132*	*-0.073*	*-0.049*	*-0.029*
IP_LAMB_DELTA_I_LAM_ADJ_DIAG	LDPM_EFF_CAT_DIAG_1_LACO	*0.000*	-0.005	-0.002	-0.001	0.000	0.000	0.000
		0.055	-0.008	-0.003	-0.001	-0.001	0.000	0.000
		0.180	-0.014	-0.005	-0.002	-0.001	0.000	0.000
		0.297	-0.019	-0.007	-0.003	-0.001	0.000	0.000
		0.484	-0.023	-0.008	-0.003	-0.002	0.000	0.000
		0.578	-0.033	-0.011	-0.005	-0.002	0.000	0.000
		0.766	-0.035	-0.012	-0.005	-0.002	0.000	0.000
		1.000	-0.039	-0.014	-0.005	-0.003	-0.001	0.000

0.000	*0.029*	*0.049*	*0.068*	*0.088*	*0.151*	*0.220*	*0.288*	*0.342*	*0.400*
0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.002	0.002	0.002
0.000	0.000	0.000	0.001	0.001	0.001	0.002	0.003	0.003	0.004
0.000	0.000	0.000	0.001	0.001	0.002	0.003	0.005	0.006	0.007
0.000	0.000	0.001	0.001	0.002	0.002	0.004	0.006	0.008	0.009
0.000	0.000	0.001	0.002	0.002	0.003	0.005	0.008	0.010	0.011
0.000	0.000	0.001	0.003	0.003	0.004	0.007	0.011	0.014	0.016
0.000	0.000	0.001	0.003	0.003	0.004	0.007	0.011	0.014	0.017
0.000	0.000	0.001	0.003	0.004	0.005	0.008	0.013	0.016	0.019

Calibration of I-share calculation parameters:

The integral term of dynamic fuel trim shall be stable after the three first pre-conditioning cycles. For the chosen operating point the controller output shall decrease if the deviation is negative (rich side) and increase if the deviation is positive (lean side).

Overshoots shall be avoided. For the chosen operating point set first IP_CRLC_LAMB_DELTA_I_LAM_ADJ[i] = 1 and then adjust the value of IP_LAMB_DELTA_I_LAM_ADJ.

IP_CRLC_LAMB_DELTA_I_LAM_ADJ[i] has influence on the adaptation speed.

C_LAMB_DELTA_I_MIN_LAM_ADJ	This forms the minimum value possible for the Integral component of the trim controller. (**e.g. -0.035…-0.025**)
C_LAMB_DELTA_I_MAX_LAM_ADJ	This forms the maximum value possible for the Integral component of the trim controller. (**e.g. +0.025…+0.035**)
LC_LAM_ADJ_I_SAVE_CAT_DIAG	With this parameter set to 1 the I-share calculated during the catalyst diagnosis is discarded afterwards. (**e.g. 1**)

46.19.3.5 Integral Component Mean Value

C_CRLC_LAMB_DELTA_I_MMV_LAM_ADJ	This forms the correlation constant for the Integral Share mean value calculation during the normal trim control phase. **(e.g. 0.001)**

46.19.3.6 D-share Specific Activation and Calculation

C_MAF_MIN_LAM_ADJ_D_ACT	This constant forms the minimum MAF threshold for the activation of the D-share of the trim controller. **(e.g. same as for P-share)**
C_MAF_MAX_LAM_ADJ_D_ACT	This constant forms the maximum MAF threshold for the activation of the D-share of the trim controller. **(e.g. same as for P-share)**
C_N_MIN_LAM_ADJ_D_ACT	This constant forms the minimum Engine Speed threshold for the activation of the D-share of the trim controller. **(e.g. same as for P-share)**
C_N_MAX_LAM_ADJ_D_ACT	This constant forms the maximum Engine Speed threshold for the activation of the D-share of the trim controller. **(e.g. same as for P-share)**

IP_LAMB_DELTA_D_LAM_ADJ	This forms the Trim D-share Correction based on the voltage error of the Downstream sensor from the Setpoint. The benefit of the D-share is rejection of transient disturbance to reduce NOx emission. The dependency is only on VLS_DIF_LAM_ADJ. **(e.g. see table below)**
IP_FAC_LAMB_DELTA_D_LAM_ADJ[NC_CBK_EX_NR]	This is a weighting factor for the D-share based on MAF. **(e.g. see table below)**
IP_FAC_ADD_LAMB_DELTA_D	This is a weighting factor for the D-share based on EFF_CAT_DIAG. So the normal D operation can be fine tuned with the aging of the Catalyst. **(e.g. see table below)**

IP_CRLC_VLS_DIF_MMV_LAM_ADJ	Filter parameter for calculation of the moving mean value of the voltage difference. The parameter is MAF_CYL and EFF_CAT_DIAG dependent in order to consider longer durations of the D-share at low air mass flows and shorter due to OSC deterioration with an aged catalyst. **(e.g. see table below)**

LDPM_VLS_DIF_DELTA_IP_LAMB_D				*-0.63965*	*-0.40039*	*-0.28809*	*-0.11719*	*-0.05859*	*-0.02930*
IP_LAMB_DELTA_D_LAM_ADJ				-0.02698	-0.01898	-0.01477	-0.00769	-0.00397	-0.00128
-0.00977	*0,000*	*0.00977*	*0.02930*	*0.05859*	*0.09766*	*0.16602*	*0.30273*	*0.40039*	*0.64453*
0.00000	0.00000	0.00000	0.00134	0.00409	0.00726	0.01202	0.01898	0.02301	0.02972

LDPM_MAF_CYL_1_LACO	*20.000*	*35.000*	*60.000*	*90.000*	*150.000*	*230.000*
IP_FAC_LAMB_DELTA_D_LAM_ADJ [NC_CBK_EX_NR]	0.950	0.870	0.800	0.740	0.600	0.500

LDPM_EFF_CAT_DIAG_1_LACO	*0.000*	*0.055*	*0.180*	*0.297*	*0.484*	*0.578*	*0.766*	*1.000*
IP_FAC_ADD_LAMB_DELTA_D	0.988	0.898	0.750	0.648	0.520	0.469	0.371	0.301

LDPM_MAF_CYL_1_LACO			*33*	*66*	*100*	*150*	*200*	*400*
IP_CRLC_VLS_DIF_MMV_LAM_ADJ	LDPM_EFF_CAT_DIAG_1_LACO	*0.04688*	0.00003	0.00005	0.00011	0.00011	0.00011	0.00011
		0.07813	0.00003	0.00005	0.00011	0.00011	0.00011	0.00011
		0.14844	0.00005	0.00006	0.00012	0.00012	0.00012	0.00012
		0.39844	0.00011	0.00012	0.00018	0.00018	0.00018	0.00018
		0.60156	0.00012	0.00014	0.0002	0.0002	0.0002	0.0002
		0.79688	0.00015	0.00017	0.00023	0.00023	0.00023	0.00023
		1	0.00523	0.00525	0.00531	0.00531	0.00531	0.00531
		1.25	0.0103	0.01031	0.01038	0.01038	0.01038	0.01038

46.20 Lambda adaptation (LACO_ISPCLADA0)

The Lambda Adaptation is compensating air/fuel ratio variations based on vehicle to vehicle differences caused by air path and/or fuel path errors via an additive and a multiplicative correction in the injection set point calculation (component variations for injector pressure regulator / injection valves / small air leakage's / bypass air for throttle plate / MAF or Map sensor / intake system, ...). Furthermore long term variations of air/fuel metering are covered (component ageing for e.g. injector pressure regulator / injection valves / throttle plate and throttle body deposit. / MAF or Map sensor, ...).

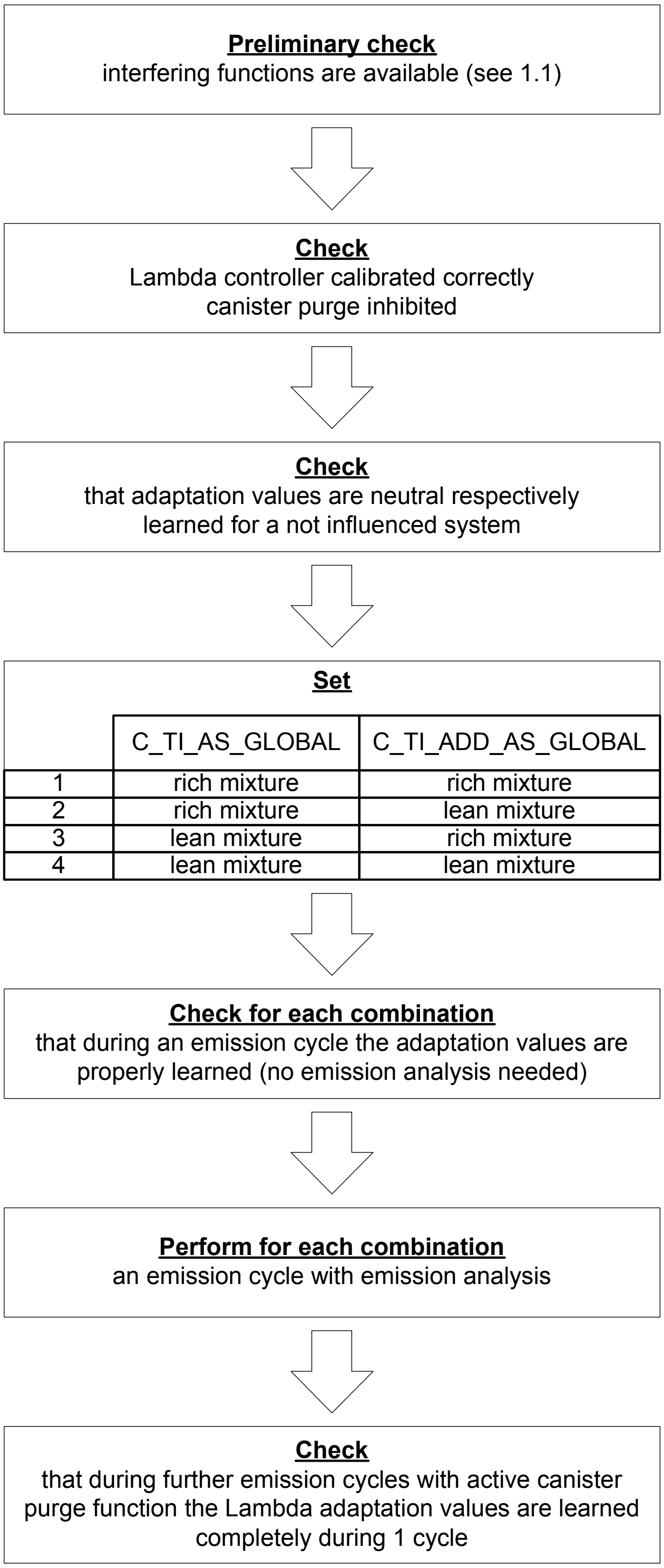
A typical example for an additive correction is the variation of air leakage's as they are constant in idle speed and negligible for higher charges. A typical example for a multiplicative correction is the variation of pressure regulator, which has to be applied for all charges.

In an early development phase for a new engine/car large variations of the lambda adaptive values may give you a hint (like the lambda control itself) for a principle problem / error in the field mentioned above (wrong component used) but also on the intake system general (e.g. mismatch between CAM-shaft and CRK-shaft) and also on the outlet system (change of exhaust line which provides a different exhaust back pressure).

46.20.1 Calibration interfering functions

- Lambda control (LACO_REQGNLACO0)
- Fuel System Diagnosis (LACO_FCTDGFSD0)
- Evaporative system control (EVAC_REQCOTIME0)
- Fuel mass setpoint calculation (FMSP_REGGNMFFSP0)

46.20.2 Calibration flowchart



Preliminary check
interfering functions are available (see 1.1)

Check
Lambda controller calibrated correctly
canister purge inhibited

Check
that adaptation values are neutral respectively
learned for a not influenced system

Set

	C_TI_AS_GLOBAL	C_TI_ADD_AS_GLOBAL
1	rich mixture	rich mixture
2	rich mixture	lean mixture
3	lean mixture	rich mixture
4	lean mixture	lean mixture

Check for each combination
that during an emission cycle the adaptation values are
properly learned (no emission analysis needed)

Perform for each combination
an emission cycle with emission analysis

Check
that during further emission cycles with active canister
purge function the Lambda adaptation values are learned
completely during 1 cycle

The following values are to be recorded with application system during emission cycle every (sample rate of at least 20 ms is recommended):

LACO_REQGNLACO0 – "Lambda Control (fuel mass)":

FAC_LAM_MV_MMV[i]
FAC_LAM_LIM[i]
LV_LAM_LSCL[i]
LV_LAM_STOP[i]

EVAC_MDLAD0 – "Evaporative emission control":

STATE_CP
LV_CP_CLOSE_ACT

EVAC_REQCOTIME0 – "Time control between EVAP and Lambda adaptation":

STATE_T_SDL_CP

LACO_ISPCLADA0 – "Lambda adaptation":

STATE_LAM_AD[i]
MFF_ADD_LAM_AD_OUT[i]
MFF_ADD_LAM_AD[i]
MFF_DELTA_ADD_LAM_AD[i]
FAC_LAM_AD_OUT[i]
FAC_LAM_AD[i]
FAC_L_RNG_LAM_AD[i]
FAC_DELTA_L_RNG_LAM_AD[i]
FAC_H_RNG_LAM_AD[i]
FAC_DELTA_H_RNG_LAM_AD[i]
LV_LAM_AD_END
T_PRI_TOT_LAM_AD

LACO_ISPCLADAPI0 – " Application Incidences for Lambda adaptation":

LV_LAM_AD_ACT[i]

46.20.3 Calibration method

The normal range for the lambda controller may have ±5% in steady state which would represent a good calibration. This range may increase up to ±15% during transient conditions. As the lambda adaptation has to cover long term drifts and/ or vehicle to vehicle deviations one has to face to a value resulting out of this for example (rough estimation):

Tolerances of Injectors + MAF/MAP + Fuel Pressure + ... = ±5% ± 5% ± 5% = ±15%

Taking both into account the working range for the lambda sensor has to be at least ±30%. On the other hand a wider range should not be chosen to guarantee a driveability at all in case of a completely wrong lambda sensor signal; eg. due to exhaust line leakage. The range for the lambda adaptation has to be smaller than the one for the lambda control to avoid a “blocked” system which cannot correct wrongly learned values (for example during low fuel tank level).

Weighting factors

IP_FAC_N_MFF_ADD_LAM_AD	Engine speed weighting of additive adaptation factor (offset). (**e.g. 1 for all points**)
IP_FAC_N_FAC_LAM_AD	Engine speed weighting of multiplicative adaptation factor. (**e.g. 1 for all points**)

IP_FAC_WUP_MFF_ADD_LAM_AD	Warm-up weighting of additive adaptation factor (offset). (**e.g. 1 for all points**) *For the warm-up phase (in case TCO <= C_TCO_MIN_LAM_AD) additional weighting factors can be applied to lower the adaptive influence if needed.*
IP_FAC_WUP_FAC_LAM_AD	Warm-up weighting of multiplicative adaptation factor. (**e.g. 1 for all points**) *see remark at IP_FAC_WUP_MFF_ADD_LAM_AD*

For each of the outputs MFF_ADD_LAM_AD[i] and FAC_LAM_AD[i] the respective weighting factor depending on engine speed can be applied. They should be normally calibrated to 1 to have both outputs applied over the whole engine speed range. Another calibration strategy would be a weighting factor for the additive adaptive value decreasing from 1 to 0 for 1300 to 1500 rpm while the factor for the multiplicative adaptive value is working in the opposite direction to avoid any influence on each other.

Priority management between Lambda adaptation and canister purge

ID_T_PRI_MFF_ADD_RNG_LAM_AD	Priority management between lambda adaptation and canister purge. Priority time according to offset difference. (**e.g. see Table 1**) *The right calibration depends on engine displacement and engine auxiliaries and on the overall strategy of "Time control between EVAP and Lambda adaptation".*
ID_T_PRI_FAC_L_RNG_LAM_AD	Priority management between lambda adaptation and canister purge. Priority time according to factor difference in lower learning area. (**e.g. see Table 1**)
ID_T_PRI_FAC_H_RNG_LAM_AD	Priority management between lambda adaptation and canister purge. Priority time according to factor difference in upper learning area. (**e.g. see Table 1**)
C_T_TRI_MAX_LAM_AD	Maximum priority time to set temporary end of adaptation flag. (**e.g. 88 s**) Must below maximum of map values (ID_T_PRI…).

ID_T_PRI_MFF_ADD_RNG_LAM_AD	30	40	60	90	50	30	20	15	s
LDP_MFF_DELTA_ADD_LAM_AD_T_PRI	-0,47	-0,03	-0,17	-0,03	0,02	0,017	0,33	0,46	mg/stk

ID_T_PRI_FAC_L_RNG_LAM_AD	30	40	60	90	50	35	30	20	s
LDP_FAC_DELTA_L_RNG_LAM_AD_PRI	-1,199	-0,900	-0,500	-0,101	0,079	0,500	0,700	1,199	%

ID_T_PRI_FAC_H_RNG_LAM_AD	30	40	60	90	50	35	30	20	s
LDP_FAC_DELTA_H_RNG_LAM_AD_PRI	-1,199	-0,900	-0,500	-0,101	0,079	0,500	0,700	1,199	%

Table 1: Example for priority management between lambda adaptation and canister purge

Definition of learning areas

C_N_TOL_ADD_RNG_LAM_AD	Maximum engine speed for offset Lambda adaptation field definition. (**e.g. 1000 rpm**)
C_MFF_SP_TOL_ADD_RNG_LAM_AD	Maximum fuel mass setpoint for offset Lambda adaptation field definition. (**e.g. 14 mg/stk**)
C_N_BOL_FAC_L_RNG_LAM_AD	Minimum speed for lower multiplicative Lambda adaptation field definition. (**e.g. 1400 rpm**)
C_N_TOL_FAC_L_RNG_LAM_AD	Maximum speed for lower multiplicative Lambda adaptation field definition. (**e.g. 3000 rpm**)

ID_MFF_SP_BOL_FAC_L_RNG_LAM_AD	Minimum fuel mass setpoint for lower multiplicative Lambda adaptation field definition. (**e.g. see Table 2**)
ID_MFF_SP_TOL_FAC_L_RNG_LAM_AD	Maximum fuel mass setpoint for lower multiplicative Lambda adaptation field definition. (**e.g. see Table 2**)
C_N_BOL_FAC_H_RNG_LAM_AD	Minimum speed for upper multiplicative Lambda adaptation field definition. (**e.g. 2500 rpm**)
C_N_TOL_FAC_H_RNG_LAM_AD	Maximum speed for upper multiplicative Lambda adaptation field definition. (**e.g. 4200 rpm**)
ID_MFF_SP_BOL_FAC_H_RNG_LAM_AD	Minimum fuel mass setpoint for upper multiplicative Lambda adaptation field definition. (**e.g. see Table 2**)
ID_MFF_SP_TOL_FAC_H_RNG_LAM_AD	Maximum fuel mass setpoint for upper multiplicative Lambda adaptation field definition. (**e.g. see Table 2**)

Three learning field have to be specified. In one field a fuel mass flow offset for MFF_SP[i] is learned and in two fields a factor that is multiplied with MFF_SP[i] is learned. The 3 learned values are stored in the non-volatile memory. The following points must be considered when calibrating the field borders:

- the three defined learning areas must not overlap each other.
- the additive value should cover the idle speed area.
- operating points in part load covered in the emission cycle should lie within the learning area of the lower multiplicative factor.
- operating points in higher part load covered in the emission cycle may lie within the learning area of the lower multiplicative factor.

ID_MFF_SP_BOL_FAC_L_RNG_LAM_AD	7	7	7	7	7	7	7	7	mg/stk
LDP_N_32_ID_MFF_SP_BOL_L_RNG	0	1504	1536	3360	4512	5632	6752	8000	rpm

ID_MFF_SP_TOL_FAC_L_RNG_LAM_AD	23	23	23	23	23	23	23	23	mg/stk
LDP_N_32_ID_MFF_SP_TOL_L_RNG	0	1504	1536	3360	4512	5632	6752	8000	rpm

ID_MFF_SP_BOL_FAC_H_RNG_LAM_AD	30	30	30	30	30	30	30	30	mg/stk
LDP_N_32_ID_MFF_SP_BOL_H_RNG	0	1120	3008	3360	4512	5632	6752	8000	rpm

ID_MFF_SP_TOL_FAC_H_RNG_LAM_AD	45	45	45	45	45	45	45	45	mg/stk
LDP_N_32_ID_MFF_SP_TOL_H_RNG	0	1120	3008	3360	4512	5632	6752	8000	rpm

Table 2: Example for definition of adaptation learning area

Interpolation between lower and upper adaptation factor

IP_MFF_SP_MAX_FAC_L_RNG_LAM_AD	Maximum fuel mass setpoint to consider the lower multiplicative adaptation factor only. Above this threshold adaptation value is found by interpolation between lower and upper adaptation factor. (**e.g. see Table 3**) *Values should lie within the specified lower learning area.*
IP_MFF_SP_MIN_FAC_H_RNG_LAM_AD	Minimum fuel mass setpoint to consider the lower multiplicative adaptation factor only. Below this threshold adaptation value is found by interpolation between lower and upper adaptation factor. (**e.g. see Table 3**) *Values should lie within the specified upper learning area.*

IP_MFF_SP_MAX_FAC_L_RNG_LAM_AD	23	23	23	23	23	23	23	23	mg/stk
LDPM_N_32_5_LACO	0	1120	3008	3360	4512	5632	6752	8000	rpm

IP_MFF_SP_MIN_FAC_H_RNG_LAM_AD	30	30	30	30	30	30	30	30	mg/stk
LDPM_N_32_5_LACO	0	1120	3008	3360	4512	5632	6752	8000	rpm

Table 3: Example for interpolation range of mulitplicative adaptation factor

Limitation of adaptation values

C_MFF_MIN_ADD_RNG_LAM_AD	Minimum limitation of offset adaptation. (**e.g. -2.5 mg/stk**) *The range for the lambda adaptation has to be smaller than the one for the lambda control to avoid a “blocked” system which cannot correct wrongly learned values (for example during low fuel tank level).*
C_MFF_MAX_ADD_RNG_LAM_AD	Maximum limitation of offset adaptation. (**e.g. +2.5 mg/stk**) *see remark at C_MFF_MIN_ADD_RNG_LAM_AD*
C_FAC_MIN_L_RNG_LAM_AD	Minimum limitation of lower multiplicative adaptation factor (**e.g. -20%**) *see remark at C_MFF_MIN_ADD_RNG_LAM_AD*
C_FAC_MAX_L_RNG_LAM_AD	Maximum limitation of lower multiplicative adaptation factor (**e.g. +20%**) *see remark at C_MFF_MIN_ADD_RNG_LAM_AD*
C_FAC_MIN_H_RNG_LAM_AD	Minimum limitation of upper multiplicative adaptation factor (**e.g. -20%**) *see remark at C_MFF_MIN_ADD_RNG_LAM_AD*
C_FAC_MAX_H_RNG_LAM_AD	Maximum limitation of upper multiplicative adaptation factor (**e.g. +20%**) *see remark at C_MFF_MIN_ADD_RNG_LAM_AD*

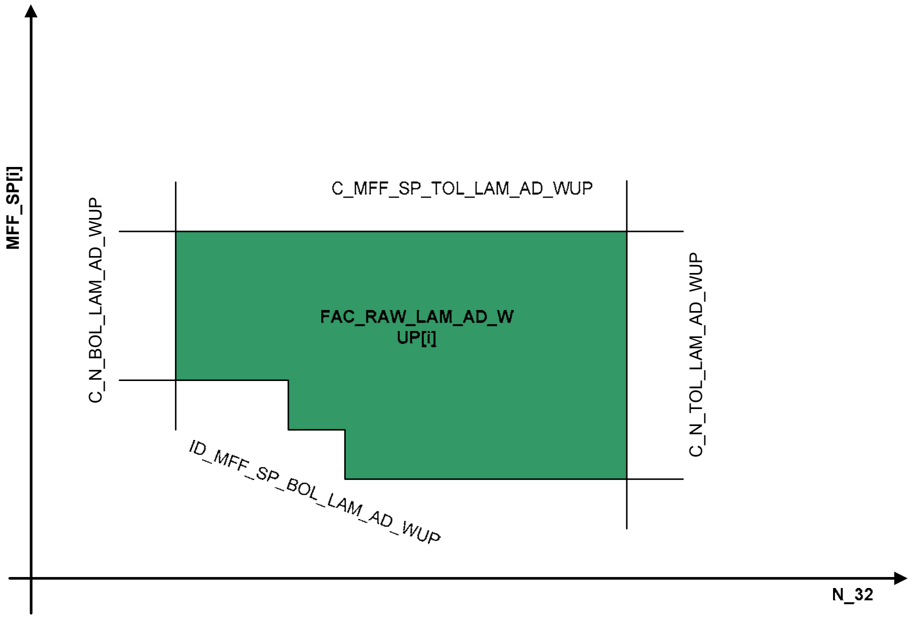
Force Lambda eq. 1 conditions

LC_MFF_ADD_RNG_AFS_REQ_LAM_AD	Enable request flag to for combustion manager to force Lambda eq. 1 conditions for Lambda adaptation in additive learning area (**0** for MPI, 1 for HPDI depending on project philosophy)
LC_FAC_L_RNG_AFS_REQ_LAM_AD	Enable request flag to for combustion manager to force Lambda eq. 1 conditions for Lambda adaptation in lower multiplicative learning area (**0** for MPI, 1 for HPDI depending on project philosophy)
LC_FAC_H_RNG_AFS_REQ_LAM_AD	Enable request flag to for combustion manager to force Lambda eq. 1 conditions for Lambda adaptation in upper multiplicative learning area (**0** for MPI, 1 for HPDI depending on project philosophy)

46.21 Lambda adaptation in warm-up phase (LACO_ISPCLADW0)

An adaptive corrections applied to compensate serial production tolerances of components at low temperatures and reduce thereby the Lambda controller excursions. The correction factors (%) are learned in five temperature ranges. They are derived from the Lambda controller output and are learned in a calibratable speed/load range. They are, however, used for calculating the injection time for all engine states. Learning of these factors requires that the Lambda controller is in the state 'ON'.

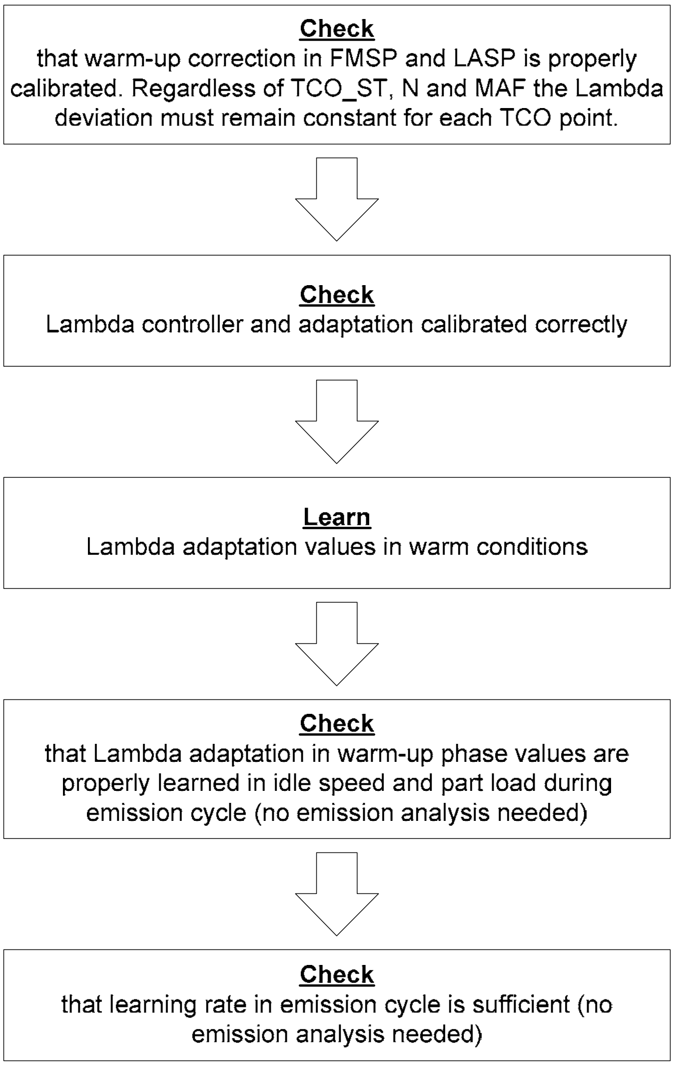
Depending on MFF_SP[i] and N_32 the adaptation fields is defined as follows:.



46.21.1 Calibration interfering functions

- Lambda control (LACO_REQGNLACO0)
- Lambda adaptation (LACO_ISPCLADA0)
- Fuel System Diagnosis (LACO_FCTDGFSD0)
- Fuel mass setpoint calculation (FMSP_REGGNMFFSP0)
- Warm-up correction (FMSP – 700A)
- Basic Lambda Setpoint including Homogeneous Lean Mode (LASP – 702P)
-

46.21.2 Calibration flowchart



Check
that warm-up correction in FMSP and LASP is properly calibrated. Regardless of TCO_ST, N and MAF the Lambda deviation must remain constant for each TCO point.

Check
Lambda controller and adaptation calibrated correctly

Learn
Lambda adaptation values in warm conditions

Check
that Lambda adaptation in warm-up phase values are properly learned in idle speed and part load during emission cycle (no emission analysis needed)

Check
that learning rate in emission cycle is sufficient (no emission analysis needed)

46.21.3 Calibration method

Learning of adaptation factors

C_CRLC_FAC_DIF_LAM_AD_WUP_MV	Correlation constant for Lambda controller output without Lambda adaptation in warm-up phase correction (**e.g. 0.05**) *Chose value in such way, that no major noise influence is seen, but still the shape should be similar to that of the controller output, especially with regard to the bumps and dents. The location of the largest deviation from the 0%-line should be similar for the filtered and not filtered Lambda controller out. The difference should be less than 10s and not more than one resolution step of TCO.*
C_CTR_TCO_MIN_INTER	Threshold above which counter has to rise to detect the lowest temperature valid for learning. (**e.g. 10**) *The threshold should not be exceeded by a dithering counter prior to the largest deviation of the controller output. However, the detection should not be delayed unnecessary once the largest deviation is passed.*
C_T_MAX_LAM_AD_WUP_LDC	Minimum time for which the conditions required for learning have to be fulfilled. (**e.g. 5 s**)

C_T_MAX_LAM_AD_WUP_TRA_PHA	Minimum time after first activation of Lambda controller to enable learning of adaptation values. (**e.g. 10 s**) *The time criteria has to be satisfied in parallel to the counter criteria to enable learning.*
C_FAC_LAM_AD_WUP_MIN_GAP	Minimum (absolute) difference to previously learned adaptation factors to update adaptation factor. (**e.g. 0%**)
C_CRLC_FAC_LAM_TCO_X	Correlation constant for updating adaptation factor. (**e.g. 0.7**)
C_CRLC_FAC_LAM_TCO_FIRST_DC	Correlation constant for updating adaptation factor in first driving cycle after initialisation of ECU or clearing of adaptation values. This factor enables a faster learning in case no previous adaptation values are available. (**e.g. 1**)

Limitation of adaptation value

IP_FAC_LAM_AD_WUP_MIN	Minimum limitation of adaptation factor in warm-up phase. (**e.g. -15% for all points**) *The range for the lambda adaptation has to be smaller than the one for the lambda control.*
IP_FAC_LAM_AD_WUP_MAX	Maximum limitation of adaptation factor in warm-up phase (**e.g. +20% for all points**) *see remark at IP_FAC_LAM_AD_WUP_MIN*
C_FAC_LAM_AD_WUP_MIN_LSCL_OFF	Minimum limitation of adaptation factor in warm-up phase while Lambda controller is off. (**e.g. -15%**) *see remark at IP_FAC_LAM_AD_WUP_MIN*
C_FAC_LAM_AD_WUP_MAX_LSCL_OFF	Maxmum limitation of adaptation factor in warm-up phase while Lambda controller is off. (**e.g. +20%**) *see remark at IP_FAC_LAM_AD_WUP_MIN*
C_FAC_LAM_AD_WUP_MIN_INTER	Minimum limitation of adaptation factor in warm-up phase after activation of Lambda controller but before detection of minimum temperature valid for learning. (**e.g. -15%**) *see remark at IP_FAC_LAM_AD_WUP_MIN*
C_FAC_LAM_AD_WUP_MAX_INTER	Maximum limitation of adaptation factor in warm-up phase after activation of Lambda controller but before detection of minimum temperature valid for learning. (**e.g. +20%**) *see remark at IP_FAC_LAM_AD_WUP_MIN*
C_CRLC_GLOBAL_FAC_LAM_AD_WUP	Factor to limit the adaptive influence, if needed. (**e.g. 1**)

Definition of learning areas

C_N_BOL_LAM_AD_WUP	Minimum speed for adaptation field definition. (**e.g. 750 rpm**)
C_N_TOL_LAM_AD_WUP	Maximum speed for adaptation field definition. (**e.g. 4000 rpm**)
ID_MFF_SP_BOL_LAM_AD_WUP	Minimum fuel mass setpoint for adaptation field definition. (**e.g. 7 mg/stk for all points**)
C_MFF_SP_TOL_LAM_AD_WUP	Maximum fuel mass setpoint for adaptation field definition. (**e.g. 23 mg/stk**)

Temperature sampling point definition

C_TCO_A_LAM_AD_WUP	Coolant temperature of sampling point A to learn adaptation value. (**e.g. -10 °C**)
C_TCO_B_LAM_AD_WUP	Coolant temperature of sampling point B to learn adaptation value. (**e.g. 0 °C**)
C_TCO_C_LAM_AD_WUP	Coolant temperature of sampling point C to learn adaptation value. (**e.g. +10 °C**)

C_TCO_D_LAM_AD_WUP	Coolant temperature of sampling point D to learn adaptation value. (**e.g. +20 °C**)
C_TCO_E_LAM_AD_WUP	Coolant temperature of sampling point E to learn adaptation value. (**e.g. +30 °C**)
C_TCO_F_LAM_AD_WUP	Coolant temperature at which adaptation in warm-up phase value is faded out. (**e.g. 40 °C**) *This must be lower than C_TCO_MIN_LAM_AD to differentiate between Lambda adaptation in warm conditions and Lambda adaptation in cold conditions (warm-up phase).*
C_TCO_LAM_AD_WUP_RNG	Coolant temperature range above and below TCO sampling point A to E in which the respective controller ouput can be learned. (**e.g. 2 °C**) *Parameter has diametrical effect on learning rate and on accuracy. On the one hand, larger values assist the learning as the temperature window in which learning is possible is increased. On the other hand, the accuracy decreases since the distance between two interpolation points may increase. Therefore, the less monoton the curve is the larger the possible error will be.*

Application of adaptation factor

C_FAC_LAM_AD_WUP_DELTA	Maximum step size to ramp adaptation factor to new value. (**e.g. 1 %**) *Ramp should not be steeper than system reaction.*
LC_FAC_RAW_LAM_AD_WUP_LIM_ACT	Logical variable to force application of adaptation value learnt at lowest temperature to all interpolation points below this sampling point. (**e.g. 1**)
LC_LAM_AD_WUP_CLR_INH	Logical variable to exclude all data of this fuction which is stored in the non-volatile memory from initialization. This variable offers the opportunity to prevent the loss of adaptattion data in case of failure memory clearing (**e.g. 0**) *For application purposes only!*

Detection of change in fuel quality

ID_FAC_THD_CHG_FQ_DET	Threshold of difference in adaptation value compared to previously learnt values to trigger change in fuel quality detection and execute shift of all adaptation values. (**e.g. 5% for all values**)
IP_FAC_CHG_FQ_DET_SCA	Scaling factor by which all adaptation values are shifted in case of change in fuel quality detection. (**e.g. 1 for all values**)

Auto shift functionality

C_THD_DIF_L_RNG_LAM_AD	Threshold of change in adaptation value compared to previous shift that must be exceeded to execute the auto shift of adaptation values in warm-up phase. (**e.g. 0**) *Set to maximum to deactivate functionality.*
C_THD_FAC_DELTA_L_RNG	Threshold below which the change in adaptation values in low range FAC_DELTA_L_RNG_LAM_AD[i] must fall to all shift of adaptation values in warm-up phase. (**e.g. 0.05**)
IP_FAC_WUP_FAC_LAM_AD_WUP	Copy of map defined in Lambda adaptation. (**values must be identical to IP_FAC_WUP_FAC_LAM_AD!**)

LC_LAM_AD_WUP_SHIFT_ACT	Logical variable to allow auto shift in all driving cycle (and not only in first driving cycle after clearing of adaptation values. (**e.g. 1**)

Write down of adaption value to next lower value

C_TCO_RNG_WR_DOWN_LAM_AD_WUP	Maximum temperature difference between two adaptation values to enable (if all other conditions are fulfilled) write down of adaptation value to next lower interpolation point. (**e.g. 0 °C**) *Functionality is not recommended and shall therefore be deactivated.*
LC_FAC_WR_DOWN_LAM_AD_WUP	Logical variable to enable write down to next lower adaptation value. (**e.g. 0**) *Functionality is not recommended and shall therefore be deactivated.*

46.22 Lambda control for WRAF sensor

46.22.1 Calibration interfering function:

- Forced lambda stimulation (linear control) (EGTR_DEFSPFSTI0): Lambda stimulation is used for system identification as "lambda set point step generator".
- Lambda Adaptation (LACO_ISPCLADA0): Can be used to adjust lambda = 1 condition before start system identification; must be inactive during system identification.
- Dynamic fuel trim (LACO_REQGNTRIM0): Trim control should be deactivated to not generate a set point shift with the P share output (eg. set C_TEMP_CAT_MIN_LAM_ADJ_ACT to maximum value).
- Evaporative emission control (EVAC_MDLAD0): Canister purge must be switched off during system identification (eg. set LC_CP_ACT to 0 or C_TCO_MIN_CP to maximum; LC_CP_ACT is only available in AGGR version of EVAC).
- Activation conditions and in cylinders EGR ratio setpoints calculation (EGRC_ISPCL0): EGR must not be active during system identification (eg. set C_N_32_MIN_ACT_EGR to maximum value).
- Catalyst efficiency diagnosis (EGTR_FCTDGCEF0): Catalyst efficiency diagnosis must be deactivated during system identification by setting C_N_32_CAT_DIAG_DEAC_THD to 0.
- Offset adjustment of oxygen sensor (linear control) (EGCP_ADCPRLSL1): The offset can be learned once immediately after engine start. Then the function must be deactivated to not disturb the system identification (offset adjustment would lead to interruption of forced lambda stimulation). The function can be deactivated by setting C_LAMB_LS_UP_MIN_OFS_ADJ and C_LAMB_LS_UP_MAX_OFS_ADJ both to the same value eg. to 0.
- Linear Oxygen sensor diagnosis (EGCP_SIGDGLSL2): The WRAF Sensor Dynamic Diagnosis is part of this diagnosis function and needs the calibration of the time for the WRAF sensor model for the aged limit sensor.

46.22.2 Calibration Flowchart:

46.22.3 Calibration method:

	First order time lag of the WRAF sensor model **(result of the system identification; calculated with the PT1-Tool out of measurements)**
IP_T1_LSL_UP_AGI_LIM	First order time lag of the WRAF sensor model of the aged sensor **(result of the system identification; calculated with the PT1-Tool out of measurements)**
IP_FAC_EG_DLY	Weighting factor for the calculation of the exhaust gas delay **(result of the system identification; calculated with the PT1-Tool out of measurements)**
IP_LAMB_SP_THE_PUC	Theoretical lambda-setpoint during PUC **(e.g. see table below)**

LDP_MAF_INT_PUC_IP_LAMB_SP_THE	0	607	1213	1820	g
IP_LAMB_SP_THE_PUC	2.5	2.5	2.5	2.5	-

Calibration data for controller parameter:

The IP-maps for determination of the controller parameter can be calculated with the **LAM_CTL_Tool** (SPT).

IP_FAC_GAIN_P_LAM	Gain of lambda controller P share **(calculated with the LAM_CTL_Tool based on the result of the identification of the plant)**
IP_FAC_GAIN_I_LAM	Gain of lambda controller I share **(calculated with the LAM_CTL_Tool based on the result of the identification of the plant)**
IP_FAC_GAIN_I2_LAM	Gain of lambda controller I^2 share **(calculated with the LAM_CTL_Tool based on the result of the identification of the plant)**
IP_FAC_GAIN_D_LAM	Gain of lambda controller D share **(calculated with the LAM_CTL_Tool based on the result of the identification of the plant)**

Calibration data for controller parameter in case of LV_LAM_GAIN_SWI = 1:

Calibration Data for extended O_2 balancing based on down stream signal evaluation

IP_FAC_GAIN_P_LAM_GAIN_SWI	Gain of lambda controller P share in case of LV_LAM_GAIN_SWI is set **(calculated with the LAM_CTL_Tool based on the results of the identification of the plant)**
IP_FAC_GAIN_I_LAM_GAIN_SWI	Gain of lambda controller I share in case of LV_LAM_GAIN_SWI is set **(calculated with the LAM_CTL_Tool based on the results of the identification of the plant)**
IP_FAC_GAIN_I2_LAM_GAIN_SWI	Gain of lambda controller I^2 share in case of LV_LAM_GAIN_SWI is set **(calculated with the LAM_CTL_Tool based on the results of the identification of the plant)**
IP_FAC_GAIN_D_LAM_GAIN_SWI	Gain of lambda controller D share in case of LV_LAM_GAIN_SWI is set **(calculated with the LAM_CTL_Tool based on the results of the identification of the plant)**

Calibration Data for controller activation:

ID_TCO_MIN_LAM_LSCL_IS	Minimum TCO for lambda controller activation at idle speed **(e.g. see table below)**
ID_TCO_MIN_LAM_LSCL	Minimum TCO for lambda controller activation **(e.g. see table below)**
ID_T_MAX_LSCL_ACT_TCO	Maximum time value up to that the controller can be activated by TCO condition **(e.g. see table below)** *Depends on legal requirements: time after that lambda controller must get active.*
C_N_MIN_LAM_LSCL	Minimum engine speed for lambda controller activation **(e.g. 416 rpm)**
ID_T_AST_MIN_LSCL_ACT	Minimum time after start that must be elapsed before activating the lambda controller **(e.g. see table below)** *Depends on wether it's necessary to have a reproduceable time to switch on the lamba controller after start. Serial dispersion can be taken into account.*

LDPM_TCO_ST_1_LACO	-30.0	-9.0	0.0	20.0	50.0	90.0	°C
ID_TCO_MIN_LAM_LSCL	0.0	9.8	15	20.0	30.0	30.0	°C

LDPM_TCO_ST_1_LACO	-30.0	-9.0	0.0	20.0	50.0	90.0	°C
ID_TCO_MIN_LAM_LSCL_IS	0.0	9.8	15	20.0	30.0	30.0	°C

LDPM_TCO_ST_1_LACO	-30.0	-9.0	0.0	20.3	50.3	90.0	°C
ID_T_MAX_LSCL_ACT_TCO	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.	s

LDPM_TCO_ST_1_LACO	-30.0	-9.0	0.0	20.3	50.3	90.0	°C
ID_T_AST_MIN_LSCL_ACT	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.	s

Calibration data for correlation constants:

Phys. limits for correlation constants are 0...1

High values mean a weak filtering of the signal (1 : no filtering), low values mean a strong filtering of the signal.

C_CRLC_FAC_LAM_MV	Correlation constant for calculation of the controller output mean value FAC_LAM_MV **(e.g. 0.1)**
C_CRLC_FAC_LAM_MV_MMV	Correlation constant for the calculation of the filtered controller output mean value FAC_LAM_MV **(e.g. 0.05)** *Depending on lambda adaption strategy speed.*
C_CRLC_FAC_LAM_MV_MMV_LDC	Correlation constant for the calculation of moving mean value for limited dynamics detection **(e.g. 0.1)**
C_CRLC_FAC_DIF_LAM_IN_MMV	Correlation constant for the calculation of FAC_DIF_LAM_IN_MMV **(e.g. 0.2)**
C_CRLC_VLS_DOWN_MMV_LAM	Correlation constant for the calculation of VLS_DOWN_MMV_LAM **(e.g. 0.082)**
C_CRLC_FAC_LAM_MV_MMV_LDC_DIAG	Correlation constant for the calculation of the moving mean value of lambda controller output, used by limited dynamics for cat efficiency diagnosis **(e.g. 0.08)**

Calibration data for controller limits:

C_FAC_LAM_OUT_MIN	Bottom limitation of the lambda controller output **(e.g. -25.0 %)**
C_FAC_LAM_OUT_MAX	Upper limitation of the lambda controller output **(e.g. +25.0 %)**
C_FAC_LAM_OUT_MIN_CP	Bottom limitation of the lambda controller output during canister purge ramp open phase **(e.g. -35.0 %)**
C_FAC_LAM_OUT_MAX_CP	Upper limitation of the lambda controller output during canister purge ramp open phase **(e.g. +35.0 %)**
C_FAC_LAM_P_MIN_LAM	Bottom limitation of lambda controller P share **(e.g. -25 %)**
C_FAC_LAM_P_MAX_LAM	Upper limitation of lambda controller P share **(e.g. +25 %)**
C_FAC_LAM_P_MIN_LAM_CP	Bottom limitation of lambda controller P share during active canister purge **(e.g. -35 %)**
C_FAC_LAM_P_MAX_LAM_CP	Upper limitation of lambda controller P share during active canister purge **(e.g. +35 %)**
C_FAC_LAM_I_I2_SUM_MIN_LAM	Bottom limitation of lambda controller I share + I^2 share **(e.g. -25 %)**
C_FAC_LAM_I_I2_SUM_MAX_LAM	Upper limitation of lambda controller I share + I^2 share **(e.g. +25 %)**
C_FAC_LAM_I_I2_SUM_MIN_LAM_CP	Bottom limitation of lambda controller I share + I^2 share during active canister purge **(e.g. -35 %)**
C_FAC_LAM_I_I2_SUM_MAX_LAM_CP	Upper limitation of lambda controller I share + I^2 share during active canister purge **(e.g. +35 %)**
C_FAC_LAM_D_MIN_LAM	Bottom limitation of the lambda controller D share **(e.g. -10 %)**
C_FAC_LAM_D_MAX_LAM	Upper limitation of the lambda controller D share **(e.g. +10 %)**
IP_FAC_LAM_MIN_NOT_STAT_CDN	Bottom limit of lambda controller output under non stationary operating conditions **(e.g. see table below)**
IP_FAC_LAM_MAX_NOT_STAT_CDN	Upper limit of lambda controller output under non stationary operating conditions **(e.g. see table below)**
C_FAC_LAM_DELTA_LSCL_OFF	Output signal delta to ramp the output signal to zero in case of deactivated lambda controller **(e.g. 0.999 %)**
LC_LAM_I2_ACT	Calibration flag to switch between PII^2D and PID lambda controller **(e.g. 1 for PII^2D)**

LDPM_FAC_LAM_MV_1_LACO	-50.0	-50.0	-50.0	-50.0	-50.0	-40.0	40.0	50.0	%
IP_FAC_LAM_MIN_NOT_STAT_CDN	-50.0	-50.0	-50.0	-50.0	-50.0	-50.0	30.0	40.0	%

LDPM_FAC_LAM_MV_1_LACO	-50.0	-50.0	-50.0	-50.0	-50.0	-40.0	40.0	50.0	%
IP_FAC_LAM_MAX_NOT_STAT_CDN	-50.0	-50.0	-50.0	-50.0	-40.0	-30.0	50.0	50.0	%

Calibration data for extended O^2 balancing based on down stream signal evaluation:

C_VLS_DOWN_THD_AFL_LAM	Lean treshold of downstream sensor signal for I and I^2 share calculation **(e.g. 0.230 V)**
C_VLS_DOWN_THD_AFR_LAM	Rich treshold of downstream sensor signal for I and I^2 share correction **(e.g 0.850 V)**
C_TOUT_I_I2_SHIFT_VLD	Time out for I and I^2 share shift based on downstream signal evaluation **(e.g. 3-5 s (depending on control parameters))**

46.23 Fuel system diagnosis (30B006/30B075)

46.23.1 Calibration interfering functions

- Fuel system monitoring – Application Incidences (30B00J)
- Lambda control (30701E)
- Lambda adaptation (30701F)
- Evaporative emission control (309008)
- Time control between EVAP and Lambda adaptation (30903P)

46.23.2 Calibration flowchart

46.23.2.1 Functional check

Preliminary check

Interfering functions are available (see 1.1)

⇩

Initialization check

LV_FSD_ACT[i] = 0
LV_INH_FSD_STOP_OIL = 1
CTR_STOP_FSD *restored from NVMY*
DLY_DEC_FSD_STOP_OIL = 255
T_DEC_FSD_STOP_OIL = 0

T_LAM_AD_ACT_FSD[i] = 0
CTR_LAM_LIM_LAM_AD_REQ[i] = 0
LV_LAM_LIM_LAM_AD[i] = 0
LV_LAM_ORNG_LAM_AD_REQ[i] = 0

LV_ERR_FSD[i] = 0
LV_CDN_DIAG_FSD[i] = 0
ERR_SYM_FSD[i] = NO_SYM
LV_END_DIAG_FSD[i] = 0

T_SUM_MIN_FSD[i] = 0
T_SUM_MAX_FSD[i] = 0
T_SUM_RST_FSD[i] = 0

T_SUM_END_DIAG_WIN_FSD[i] = 0
T_SUM_END_DIAG_WIN_FSD_ADD[i] = 0
T_SUM_END_DIAG_WIN_FSD_FAC_L[i] = 0
LV_END_DIAG_WIN_FSD[i] = 0

LV_ERR_FSD_LAM_LIM[i] = 0
LV_CDN_DIAG_FSD_LAM_LIM[i] = 0
ERR_SYM_FSD_LAM_LIM[i] = NO_SYM
LV_END_DIAG_FSD_LAM_LIM[i] = 0

T_SUM_MIN_FSD_LAM_LIM[i] = 0
T_SUM_MAX_FSD_LAM_LIM[i] = 0
T_SUM_RST_FSD_LAM_LIM[i] = 0

T_SUM_END_DIAG_WIN_FSD_LAM[i] = 0
LV_END_DIAG_WIN_FSD_LAM_LIM[i] = 0

LV_ERR_FSD_H_RNG[i] = 0
LV_CDN_DIAG_FSD_H_RNG[i] = 0
ERR_SYM_FSD_H_RNG[i] = NO_SYM
LV_END_DIAG_FSD_H_RNG[i] = 0

T_SUM_MIN_FSD_H_RNG[i] = 0
T_SUM_MAX_FSD_H_RNG[i] = 0
T_SUM_RST_FSD_H_RNG[i] = 0

T_SUM_END_DIAG_WIN_FSD_H_RNG[i] = 0

⇩

Set

C_TOIL_MIN_DEC_FSD_STOP_OIL
and
ID_CTR_INC_FSD_STOP_OIL
and
ID_DLY_DEC_FSD_STOP_OIL
to values given in table 1.

⇩

Check

Calculation of CTR_STOP_FSD
At transition from engine stop to engine run
CTR_STOP_FSD is incremented by
ID_CTR_INC_FSD_STOP_OIL

⇩

Check

LV_INH_FSD_STOP_OIL

is 1 as long as CTR_STOP_FSD above
C_CTR_MAX_FSD_STOP_OIL.

⇩

Check

Update of CTR_STOP_FSD
Calculation of T_DEC_FSD_STOP_OIL

Depending on ID_DLY_DEC_FSD_STOP_OIL each time the delay elapses (indicated by T_DEC_FSD_STOP_OIL) CTR_STOP_FSD is decremented by 1.

⇩

Set

C_AMP_MIN_FSD = 0
C_MAF_MIN_FSD = 0
C_N_MIN_FSD = 0
C_TAM_MIN_FSD = 0
C_TCO_MIN_FSD = 0
C_TIA_MIN_FSD = 0

C_T_LAM_AD_ACT_MIN_FSD = 15
C_CTR_MAX_LAM_LIM_LAM_AD_REQ = 150
C_CTR_INC_LAM_LIM_LAM_AD_REQ = 1
C_CTR_INC_LAM_LIM_LAM_AD_REQ = 1

IP_FAC_LAM_OUT_MIN_FSD = -2…0%
IP_FAC_LAM_OUT_MAX_FSD = 0…2%

C_T_SUM_MIN_THD_FSD = 1310.7s
C_T_SUM_MAX_THD_FSD = 1310.7s
C_T_SUM_MIN_THD_FSD_LAM_LIM = 1310.7s
C_T_SUM_MAX_THD_FSD_LAM_LIM = 1310.7s
C_T_SUM_RST_FSD = 20s

C_CRLC_T_SUM_WIN_FSD_RST = 0.2

⇩

Check

LV_FSD_ACT[i] = 1 and
LV_CDN_DIAG_FSD_LAM_LIM[i] = 1

For LV_LAM_LSCL[i] = 1,
LV_INH_DIAG_FSD[i] = 0 and
LV_INH_DIAG_FSD_CP = 0

⇩

Check

LV_FSD_ACT[i] = 1 and
LV_CDN_DIAG_FSD[i] = 1

For LV_MFF_ADD_RNG_LAM_AD[i] = 1 or
LV_FAC_L_RNG_LAM_AD[i] = 1

⇩

Check

LV_FSD_ACT[i] = 1 and
LV_CDN_DIAG_FSD_H_RNG[i] = 1

For LV_FAC_H_RNG_LAM_AD[i] = 1

⇩

Check

Incrementation of CTR_LAM_LIM_LAM_AD_REQ[i]

For FAC_LAM_LIM[i] outside window defined by
IP_FAC_LAM_OUT_MIN_FSD = -2…0%
IP_FAC_LAM_OUT_MAX_FSD = 0…2%

⇩

Check

LV_LAM_LIM_LAM_AD[i] = 1

For CTR_LAM_LIM_LAM_AD_REQ[i] >=
C_CTR_MAX_LAM_LIM_LAM_AD_REQ

⇩

Check

Incrementation of T_LAM_AD_ACT_FSD[i]

For LV_LAM_LIM_LAM_AD[i] = 1 and
(LV_MFF_ADD_RNG_LAM_AD[i] = 1 or
LV_FAC_L_RNG_LAM_AD[i] = 1 or
LV_FAC_H_RNG_LAM_AD[i] = 1)

⇩

Check

Reset of T_LAM_AD_ACT_FSD[i] = 0
CTR_LAM_LIM_LAM_AD_REQ[i] = 0
LV_LAM_LIM_LAM_AD[i] = 0
LV_LAM_ORNG_LAM_AD_REQ[i] = 0

For T_LAM_AD_ACT_FSD >=
C_T_LAM_AD_ACT_MIN_FSD

⇩

Check error detection lambda controller at limit (FSD_LAM_LIM)

Set C_TI_AS_GLOBAL from 1 to 1.4.

Incrementation of T_SUM_MIN_FSD_LAM_LIM[i]
incrementation of T_SUM_RST_FSD_LAM_LIM[i]
ERR_SYM_FSD_LAM_LIM[i] = SYM_1

as soon as LV_FAC_LAM_LIM_MIN[i] = 1

Set C_TI_AS_GLOBAL from 1 to 0.7.

Incrementation of T_SUM_MAX_FSD_LAM_LIM[i]
incrementation of T_SUM_RST_FSD_LAM_LIM[i]
ERR_SYM_FSD_LAM_LIM[i] = SYM_0

as soon as LV_FAC_LAM_LIM_MAX[i] = 1

LV_ERR_FSD_LAM_LIM[i] = 1
LV_END_DIAG_FSD_LAM_LIM[i] = 1

For T_SUM_MIN_FSD_LAM_LIM[i] >
C_T_SUM_MIN_THD_FSD_LAM_LIM
or
T_SUM_MAX_FSD_LAM_LIM[i] >
C_T_SUM_MAX_THD_FSD_LAM_LIM

⇩

Check

Reset of T_SUM_RST_FSD_LAM_LIM[i] = 0
T_SUM_MIN_FSD_LAM_LIM[i] = 0
T_SUM_MAX_FSD_LAM_LIM[i] = 0

For _SUM_RST_FSD_LAM_LIM[i] >=
C_T_SUM_RST_FSD

⇩

Check SCDN 1

Incrementation of T_SUM_END_DIAG_WIN_FSD_LAM_LIM[i]

As soon as
LV_CDN_DIAG_FSD_LAM_LIM[i] = 1

⇩

Check SCDN 2

LV_END_DIAG_WIN_FSD_LAM_LIM[i] = 1 (for 20ms)
LV_END_DIAG_FSD_LAM_LIM[i] = 1 (if not yet set)
Reset of T_SUM_END_DIAG_WIN_FSD_LAM_LIM[i] = 0.02s

For T_SUM_END_DIAG_WIN_FSD_LAM_LIM[i] >=
C_T_SUM_RST_FSD or
reset of T_SUM_RST_FSD_LAM_LIM[i] = 0.02

⇩

Check

STATE_FSD_LAM_LIM[i] is set to the corresponding state (NO_ERR, LAM_MIN or LAM_MAX) depending on the error symptom

As soon as LV_END_DIAG_FSD_LAM_LIM[i] = 1

⇩

Check error detection lambda adaptation at limit (FSD)

Set C_TI_AS_GLOBAL from 1 to 1.4.

Incrementation of T_SUM_MIN_FSD[i]
incrementation of T_SUM_RST_FSD[i]
ERR_SYM_FSD[i] = SYM_1

as soon as
(LV_MFF_ADD_RNG_LAM_AD[i] = 1 and
LV_MFF_ADD_LIM_MIN_LAM_AD[i] = 1)
or
(LV_FAC_L_RNG_LAM_AD[i] = 1 and
LV_FAC_L_RNG_LIM_MIN_LAM_AD[i] = 1)

Set C_TI_AS_GLOBAL from 1 to 0.7.

Incrementation of T_SUM_MAX_FSD[i]
incrementation of T_SUM_RST_FSD[i]
ERR_SYM_FSD[i] = SYM_0

as soon as
(LV_MFF_ADD_RNG_LAM_AD[i] = 1 and
LV_MFF_ADD_LIM_MAX_LAM_AD[i] = 1)
or
(LV_FAC_L_RNG_LAM_AD[i] = 1 and
LV_FAC_L_RNG_LIM_MAX_LAM_AD[i] = 1)

LV_ERR_FSD[i] = 1
LV_END_DIAG_FSD[i] = 1

For T_SUM_MIN_FSD[i] >
C_T_SUM_MIN_THD_FSD
or
T_SUM_MAX_FSD[i] >
C_T_SUM_MAX_THD_FSD

⇩

Check

Reset of T_SUM_RST_FSD[i] = 0
T_SUM_MIN_FSD[i] = 0
T_SUM_MAX_FSD[i] = 0

For T_SUM_RST_FSD[i] >= C_T_SUM_RST_FSD

⇩

Check SCDN 1

Incrementation of T_SUM_END_DIAG_WIN_FSD[i]

As soon as
LV_CDN_DIAG_FSD[i] = 1

Incrementation of T_SUM_END_DIAG_WIN_FSD_ADD[i]
Incrementation of T_SUM_END_DIAG_WIN_FSD_FAC_L[i]

As soon as
LV_CDN_DIAG_FSD[i] = 1 and depending on the lambda adaptation state

⇩

Check SCDN 2

LV_END_DIAG_WIN_FSD[i] = 1 (for 20ms)
LV_END_DIAG_FSD[i] = 1 (if not yet set)
Reset of T_SUM_END_DIAG_WIN_FSD[i] = 0.02s
T_SUM_END_DIAG_WIN_FSD_ADD[i] = 0.02s
T_SUM_END_DIAG_WIN_FSD_FAC_L[i] = 0.02s

For
(T_SUM_END_DIAG_WIN_FSD[i] >= C_T_SUM_RST_FSD and
T_SUM_END_DIAG_WIN_FSD_ADD[i] >=
C_T_SUM_RST_FSD * C_CRLC_T_SUM_MIN_WIN_RST and
T_SUM_END_DIAG_WIN_FSD_FAC_L[i] >=
C_T_SUM_RST_FSD * C_CRLC_T_SUM_MIN_WIN_RST
or
reset of T_SUM_RST_FSD[i] = 0.02

⇩

Check error detection lambda adaptation at limit (FSD_H_RNG)

Set C_TI_AS_GLOBAL from 1 to 1.4.

Incrementation of T_SUM_MIN_FSD_H_RNG[i]
incrementation of T_SUM_RST_FSD_H_RNG [i]
ERR_SYM_FSD_H_RNG [i] = SYM_1

as soon as
(LV_FAC_H_RNG_LAM_AD[i] = 1 and
LV_FAC_H_RNG_LIM_MIN_LAM_AD[i] = 1)

Set C_TI_AS_GLOBAL from 1 to 0.7.

Incrementation of T_SUM_MAX_FSD_H_RNG [i]
incrementation of T_SUM_RST_FSD_H_RNG [i]
ERR_SYM_FSD_H_RNG [i] = SYM_0

as soon as
(LV_FAC_H_RNG_LAM_AD[i] = 1 and
LV_FAC_H_RNG_LIM_MAX_LAM_AD[i] = 1)

LV_ERR_FSD_H_RNG [i] = 1
LV_END_DIAG_FSD_H_RNG [i] = 1

For T_SUM_MIN_FSD_H_RNG [i] >
C_T_SUM_MIN_THD_FSD
or
T_SUM_MAX_FSD_H_RNG [i] >
C_T_SUM_MAX_THD_FSD

⇩

Check

Reset of T_SUM_RST_FSD_H_RNG [i] = 0
T_SUM_MIN_FSD_H_RNG [i] = 0
T_SUM_MAX_FSD_H_RNG [i] = 0

For T_SUM_RST_FSD_H_RNG [i] >=
C_T_SUM_RST_FSD

46.23.3 Calibration

Define and verify

Lambda adaptation thresholds (see 30701F)

C_MFF_MIN_ADD_RNG_LAM_AD
C_MFF_MAX_ADD_RNG_LAM_AD
C_FAC_MIN_L_RNG_LAM_AD
C_FAC_MAX_L_RNG_LAM_AD

⇩

Define and verify

Lambda controller thresholds (see 30701E)

C_FAC_LAM_OUT_MAX
C_FAC_LAM_OUT_MAX

⇩

Define and verify

Lambda adaptation thresholds; to be checked whether the FAC_H range and therefore these thresholds are used in the project (see 30701F)

C_FAC_MIN_H_RNG_LAM_AD
C_FAC_MAX_H_RNG_LAM_AD

If FAC_H adaptation range is not used C_T_SUM_MIN_WIN_FSD_H_RNG must be set to 0.

⇩

Verify

Time control between CP and Lamdba Adaptation (see 30903P)

⇩

Apply

Upper and lower limit of lambda controller ouput to request forced lambda adaptation

IP_FAC_LAM_OUT_MIN/MAX_FSD =
approx. 70…80% of lambda controller limitation

⇩

Apply

Counter threshold and increment / decrement for forced activation of lambda adaptation

C_CTR_MAX_LAM_LIM_LAM_AD_REQ
C_CTR_INC_LAM_LIM_LAM_AD_REQ
C_CTR_DEC_LAM_LIM_LAM_AD_REQ

⇩

Apply

Timer threshold to reset forced activation of lambda adaptation

C_T_LAM_AD_ACT_MIN_FSD

⇩

_AMP_MIN_FSD
_MAF_MIN_FSD
_N_MIN_FSD
_TAM_MIN_FSD
_TCO_MIN_FSD
_TIA_MIN_FSD

_TOIL_MIN_DEC_FSD_STOP_OIL
_CTR_MAX_FSD_STOP_OIL
)_CTR_INC_FSD_STOP_OIL
)_DLY_DEC_FSD_STOP_OIL

as soon as the general activation conditions are fulfilled and
ed. That means the function monitors the lambda controller
the lambda adaptation is activated. As soon as the lambda
after an applicable time) the request for lambda adaptation will
[i] = 1). Figure 1 shows the monitoring area of fuel system
area of lambda adaptation is subset of lambda controller

30B075:

C_N_MIN_FSD	Engine speed threshold for fuel system diagnosis (**e.g. 608rpm**)
C_MAF_MIN_FSD	Mass air flow threshold for fuel system diagnosis (**e.g. 71mg/stk**)
C_TCO_MIN_FSD	Minimum threshold of TCO for fuel system diagnosis (**e.g. 65°C**)
C_AMP_MIN_FSD	Minimum ambient pressure threshold for fuel system diagnosis (**e.g. 699.99hPa**)
C_TIA_MIN_FSD	Minimum intake air temperature threshold for fuel system diagnosis (**e.g. -10.5 °C**)
C_TAM_MIN_FSD	Minimum ambient temperature threshold for fuel system diagnosis (**e.g. -10.50°C**)
C_CTR_MAX_FSD_STOP_OIL	Maximum value for the debounce counter CTR_STOP_FSD for oil dilution detection in fuel system diagnosis (**e.g. 30**)
C_TOIL_MIN_DEC_FSD_STOP_OIL	Minimum oil temperature threshold to activate timer to debounce oil dilution detection in fuel system diagnosis (**e.g. 0°C**).
ID_CTR_INC_FSD_STOP_OIL	Increasing value for the debounce counter CTR_STOP_FSD for oil dilution detection in fuel system diagnosis (**e.g. see Table 1**)
ID_DLY_DEC_FSD_STOP_OIL	Time delay for decreasing the debounce counter CTR_STOP_FSD for oil dilution detection in fuel system diagnosis (**e.g. see Table 1**)

30B006:

C_CTR_MAX_LAM_LIM_LAM_AD_REQ	Counter threshold for forced activation of lambda adaptation (**e.g. 150**)
C_CTR_INC_LAM_LIM_LAM_AD_REQ	Counter decrement for activation of forced lambda adaptation (**e.g. 1**)

C_CTR_DEC_LAM_LIM_LAM_AD_REQ	Counter increment for activation of forced lambda adaptation (**e.g. 1**)
C_T_LAM_AD_ACT_MIN_FSD	Minimum time during which forced Lambda adaptation shall be active. This value depends on the adaptation rate. (**e.g. 30s**)
IP_FAC_LAM_OUT_MIN_FSD	The lower limit of lambda controller output to request forced lambda adaptation (**e.g. 70…80% of lambda controller minimum limitation; see 30701E; e.g. see Table 2)**
IP_FAC_LAM_OUT_MAX_FSD	The upper limit of lambda controller output to request forced lambda adaptation (**e.g. 70…80% of lambda controller maximum limitation; see 30701E; e.g. see Table 2)**
C_T_SUM_MIN_THD_FSD	Fuel system diagnosis timer threshold for lambda adaptation monitoring - minimum limit (**e.g. 35s**)
C_T_SUM_MAX_THD_FSD	Fuel system diagnosis timer threshold for lambda adaptation monitoring - maximum limit (**e.g. 35s**).
C_T_SUM_MIN_THD_FSD_LAM_LIM	Fuel system diagnosis timer threshold for lambda controller monitoring - minimum limit (**e.g. 60 s**)
C_T_SUM_MAX_THD_FSD_LAM_LIM	Fuel system diagnosis timer threshold for lambda controller monitoring - maximum limit (**e.g. 60 s**)
C_T_SUM_RST_FSD	Fuel system diagnosis reset threshold (**e.g. 120 s**)
C_CRLC_T_SUM_WIN_FSD_RST	Minimal fraction of counter threshold that must be spent in ADD as well as L_RNG for error healing (**e.g. 0.2**)

ID_CTR_INC_FSD_STOP_OIL [-]	**15**	**10**	**5**	**0**
LDP_TCO_ST_ID_CTR_INC_FSD_STOP [°C]	-30	0	18	69.75
ID_DLY_DEC_FSD_STOP_OIL [-]	**18**	**12**	**6**	**3**
LDP_TOIL_ID_DLY_DEC_FSD_OIL [°C]	20	50	85	98

Table 2: Example for values of fuel system diagnosis inhibition due to fuel dilution in engine oil

IP_FAC_LAM_OUT_MAX_FSD [%]	**40**	**35**	**31**	**29**	**27**	**25**
LDPM_TCO_4_LACO [°C]	-30	-20	-10	0	5	10
IP_FAC_LAM_OUT_MIN_FSD [%]	**-40**	**-35**	**-31**	**-29**	**-27**	**-25**
LDPM_TCO_4_LACO [°C]	-30	-20	-10	0	5	10

Table 2: Example for values of lambda controller limitation of fuel system diagnosis

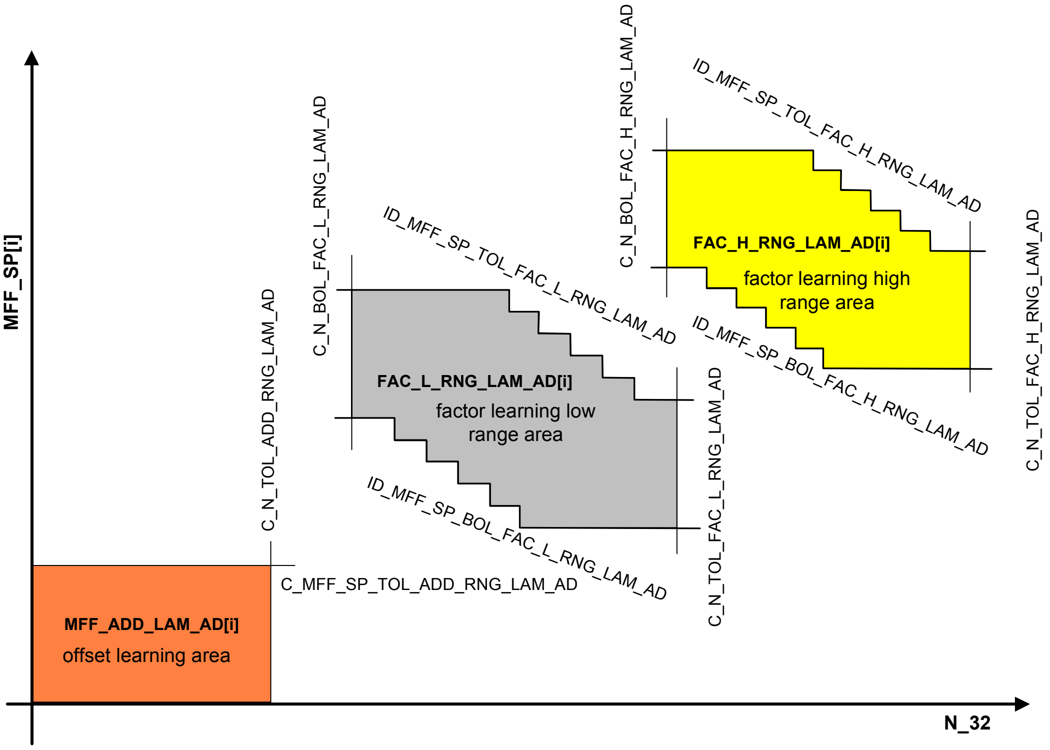


Figure 3: Lambda adaptation areas for fuel system diagnosis.

46.23.4 Apply ERRM - Diagnostic integration plan

Refer to:
https://intranet.siemensvdo.com/NR/rdonlyres/954BD3F3-96F5-4808-82AC-B72675E7AB78/428048/DIAGintegrationPlan3.doc

46.24 Customer adaptation module: AGGR LACO

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_lrad_akt1	O/V	0...1H	0...1	1	[-]
Bit indicating KLANN Bk1 shall be activated (STATE_LAM_AD[1] == "FAC_L")					
B_lrad_akt2	O/V	0...1H	0...1	1	[-]
Bit indicating KLANN Bk2 shall be activated (STATE_LAM_AD[2] == "FAC_L")					
B_lrad_deak1	O/V	0...1H	0...1	1	[-]
Bit indicating KLANN Bk1 shall be deactivated due to an error					
B_lrad_deak2	O/V	0...1H	0...1	1	[-]
Bit indicating KLANN Bk2 shall be deactivated due to an error					
B_zwgemad_1	O/V	0...1H	0...1	1	[-]
Zwangsgemischadaption Bank 1					
B_zwgemad_2	O/V	0...1H	0...1	1	[-]
Zwangsgemischadaption Bank 2					
CTR_KM_CAN_OLD	V	0...FFFFH	0...655350	10	[km]
Last vehicle kilometer update					
CTR_KM_DELTA_LAM_ADJ_LAM_AD_CUS	V/S	0...FFFFH	0...655350	10	[km]
Vehicle kilometer diff. between current value and the one for the last long term adaptation					
CTR_KM_LAM_ADJ_LAM_AD_CUS	V/S	0...FFFFH	0...655350	10	[km]
Vehicle kilometer at the last long term customer lambda adaptation stored in NVMY					
CTR_LAM_ADJ_LAM_AD_CUS	V/S	0...FFH	0...255	1	[-]
Current number of long term adaptations carried out					
EGY_LAM_ADJ_COR_LAM_AD_CUS[NC_CYL_NR]	O/V	8000...7FFFH	-127.50194...127.49805	3.8911e-3	[mJ]
Energy value of the Long term customer Lambda Adaption					
ERR_SYM_LAM_AD_CUS[NC_CBK_EX_NR]	O/V	0...8H	0...8	1	[-]
Detected symptom of the diagnosis for the long term adaptation					
F_lakor[NC_CBK_EX_NR]	O/V	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Factor for the injection time correction					
F_lrad_in1	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Faktor Lambdaadaption Bank 1					
F_lrad_in2	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Faktor Lambdaadaption Bank 2					
FAC_LAM_AD_CUS[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lambda Adaption carried out by customer (multiplicative share)					
FAC_LAM_AD_CUS_SHIFT[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Output value for shift of Lambda adaptation due to customer function					
FAC_LAM_ADJ_COR_LAM_AD_CUS[NC_CBK_EX_NR]	O/V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Long term customer Lambda Adaption (multiplicative share) via fuel mass setpoint					
FAC_LAM_ADJ_LAM_AD_CUS[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Long term customer Lambda Adaption (multiplicative share)					
FAC_LAM_ADJ_ST_LAM_AD_CUS[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Long term customer Lambda Adaption during engine start (multiplicative share)					
FAC_LAM_ADJ_TMP_LAM_AD_CUS[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Temporary value of the Long term customer Lambda Adaption (multiplicative share)					
FAC_LAM_CYL_SEL_LAM_AD_CUS[NC_CYL_NR]	V/S	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Cylinder individual long term customer Lambda Adaption (multiplicative share)					
FAC_LAM_TMP_ST_LAM_AD_CUS[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Temporary value of the Long term customer Lambda Adaption for the engine start(multiplicative share)					

FAC_LAM_UPD_LAM_AD_CUS[NC_CBK_EX_NR]	V	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
New update value to the Long term customer Lambda Adaption					
LV_CDN_DIAG_LAM_AD_CUS[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Condition of the diagnosis for the long term adaptation fullfiled					
LV_END_DIAG_LAM_AD_CUS[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
End of the diagnosis for the long term adaptation					
LV_ERR_LAM_AD_CUS[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag that indicates that the long term adaptation exceeds its limits					
LV_FAC_LAM_AD_CUS_SHIFT[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag to request shift of Lambda adaptation due to customer function					
LV_FAC_LIM_MAX_LAM_AD_CUS[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag indicating that the customer lambda adaptation is over the max threshold					
LV_FAC_LIM_MIN_LAM_AD_CUS[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flga indicating that the customer lambda adaptation is under the min threshold					
LV_INH_LSCL_CUS[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
-					
LV_LAM_ADJ_LAM_AD_CUS_ENA	V	0...1H	0...1	1	[-]
Conditions to carriy out the long term adaptations are fullfiled					

Input data:

B_la_stopp1	B_la_stopp2	B_lradap1_mn	B_lradap1_mx
B_lradap2_mn	B_lradap2_mx	CTR_KM_CAN	F_lradap1
F_lradap2	FAC_L_RNG_LAM_AD[NC_CBK_EX_NR]	FAC_LAM_LIM_FIL[NC_CBK_EX_NR]	FAC_LAM_MV_MMV[NC_CBK_EX_NR]
FAC_LAM_PCTL_CUS[NC_CBK_EX_NR]	Klann_mw1	Klann_mw2	Klann_test1
Klann_test2	LC_AD_CLR_LONG_LAM_1	LC_AD_CLR_LONG_LAM_2	LV_AD_CLR_LONG_LAM_EXT_ADJ[NC_CBK_EX_NR]
LV_FAC_LAM_AD_SHIFT_END[NC_CBK_EX_NR]	LV_LAM_AD_DEAC_ERR[NC_CBK_EX_NR]	LV_LAM_AD_EXT	LV_LAM_LIM_LAM_AD[NC_CBK_EX_NR]
NC_CBK_EX_NR	NC_CYL_NR	NC_LAMB_REF	STATE_IV_CHG
STATE_LAM_AD[NC_CBK_EX_NR]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_LAM_AD_CUS	1	0...FFH	0...255	1	[-]
Anti bounce counter increment of the diagnosis for the long term adaptation					
C_ABC_INC_LAM_AD_CUS	1	0...FFH	0...255	1	[-]
Anti bounce counter increment of the diagnosis for the long term adaptation					
C_ABC_MAX_LAM_AD_CUS	1	0...FFH	0...255	1	[-]
Anti bounce counter increment of the diagnosis for the long term adaptation					
C_CRLC_LAM_ADJ_LAM_AD_CUS	1	0...FFH	0...1	3.9216e-3	[-]
Correlation constant for the long term adaptation					
C_CRLC_LAM_ADJ_MAX_LAM_AD_CUS	1	0...FFH	0...1	3.9216e-3	[-]
Correlation constant to calculate the maximun input to the long term adaptation					
C_CTR_KM_THD_LAM_AD_CUS	1	0...FFFFH	0...655350	10	[km]
Distance to carry out next long term customer lambda adaptation					
C_CTR_MAX_LAM_ADJ_LAM_AD_CUS	1	0...FFH	0...255	1	[-]
Number of maximum allowed long term adaptations (if LC_CTR_MAX_LAM_ADJ_LAM_AD_CUS = 1)					
C_EGY_LAM_ADJ_LAM_AD_CUS	1	0...FFFFH	0...255	3.8911e-3	[mJ]
Conversion factor for the long term adaptation from % MFF to mJ					
C_FAC_LAM_ADJ_MAX_LAM_AD_CUS	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Maximun limit of the long term customer lambda adaptation					
C_FAC_LAM_ADJ_MIN_LAM_AD_CUS	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Minimun limit of the long term customer lambda adaptation					
C_FAC_LAM_ADJ_ST_MAX_LAM_AD_CUS	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Maximun limit of the long term customer lambda adaptation during engine start					
C_FAC_LAM_ADJ_ST_MIN_LAM_AD_CUS	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Minimun limit of the long term customer lambda adaptation during engine start					
C_FAC_LAM_ERR_MAX_LAM_AD_CUS	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Upper threshold value of the long term customer lambda adaptation for the diagnosis					
C_FAC_LAM_ERR_MIN_LAM_AD_CUS	1	8000...7FFFH	-50...49.99847	1.5259e-3	[%]
Lower threshold value of the long term customer lambda adaptation for the diagnosis					
IP_FAC_COR_ST_LAM_AD_CUS	8	0...FFH	0...1	3.9216e-3	[-]
LDP_TCO_ST_LAM_AD_CUS	8	0...FEH	-48...142.5	0.75	[°C]
Weighting factor for the long term adaptation during engine start					
LC_CTR_MAX_LAM_ADJ_LAM_AD_CUS	1	0...1H	0...1	1	[-]
Logical switch to enable a limitation in the number of long term adaptations					
LC_EGY_LAM_AD_CUS_ENA	1	0...1H	0...1	1	[-]
Logical switch to enable long term adaptation via injector needle lift correction					
LC_EGY_LAM_AD_CUS_ST_ENA	1	0...1H	0...1	1	[-]
Logical switch to enable long term adaptation via injector needle lift correction during engine start					
LC_LAM_AD_CUS_CLR	1	0...1H	0...1	1	[-]
Logical switch to reset the anti bounce counter for the long term adaptation diagnosis					
LC_LAM_ADJ_LAM_AD_CUS_ENA	1	0...1H	0...1	1	[-]
Logical switch to enable the long term lambda adaptation					

46.24.1 Outputs for BMW functions which are defined as LACO exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions (for BMW variables):

Initialisation at reset or at exit power latch phase: 0

Recurrence :	10ms:	B_zwgemad_1, B_zwgemad_2
	20ms:	F_lakor[NC_CBK_EX_NR]
	100ms:	F_lrad_in1/2, St_lam_ad_1/2, B_lrad_akt1/2, B_lrad_deak1/2

Activation: every engine state, except power latch phase

Decativation: at power latch phase

Values at deactivation: 0

Formula section:

Remark: all formulas except the two 1st are valid in a **physical** meaning

```
B_zwgemad_1 =LV_LAM_LIM_LAM_AD[1]
B_zwgemad_2 = LV_LAM_LIM_LAM_AD[2]
F_lrad_in1 = FAC_LAM_MV_MMV[1]
F_lrad_in2 = FAC_LAM_MV_MMV[2]
B_lrad_deak1 = LV_LAM_AD_DEAC_ERR[0]
B_lrad_deak2 = LV_LAM_AD_DEAC_ERR[1]

If STATE_LAM_AD[1] == "ADAPT_FAC_L"
or STATE_LAM_AD[1] == " ADAPT_ADD"
Then B_lrad_akt1 = 1
Else   B_lrad_akt1 = 0
Endif

If STATE_LAM_AD[2] == " ADAPT_FAC_L"
or STATE_LAM_AD[2] == " ADAPT_ADD"
Then B_lrad_akt2 = 1
Else   B_lrad_akt2 = 0
Endif

F_lakor[NC_CBK_EX_NR] =    FAC_LAM_LIM_FIL[NC_CBK_EX_NR]
                          +  FAC_LAM_PCTL_CUS[NC_CBK_EX_NR]
```

46.24.2 Outputs of customer to SV functions

Input data:

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset:

FAC_LAM_CYL_SEL_LAM_AD_CUS[x], CTR_KM_LAM_ADJ_LAM_AD_CUS CTR_KM_DELTA_LAM_ADJ_LAM_AD_CUS and CTR_LAM_ADJ_LAM_AD_CUS shall be initialised with their stored values in NVMY.

```
// The sum below shall be done for each exhaust gas bank over the cylinders belonging
//   to it.
FAC_LAM_ADJ_LAM_AD_CUS[i] =
  {ΣFAC_LAM_CYL_SEL_LAM_AD_CUS[x]} / (NC_CYL_NR / NC_CBK_EX_NR)
//
```

All others variables shall be initialised with 0.

Recurrence : 10 ms,

except the Long Term Adaptation which some parts shall have a 1s recurrence

Activation: at every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
LV_INH_LSCL_CUS[1]   =    B_la_stopp1
LV_INH_LSCL_CUS[2]   =    B_la_stopp2
```

```
% Contiuously update of current customer Lambda Adaptation value
FAC_LAM_AD_CUS[1]   =    F_lradap1
FAC_LAM_AD_CUS[2]   =    F_lradap2
```

```
LV_FAC_LIM_MAX_LAM_AD_CUS[1] = B_lradap1_mx
LV_FAC_LIM_MAX_LAM_AD_CUS[2] = B_lradap2_mx
LV_FAC_LIM_MIN_LAM_AD_CUS[1] = B_lradap1_mn
LV_FAC_LIM_MIN_LAM_AD_CUS[2] = B_lradap2_mn
```

% Long term Lambda Adaptation

// case: no update beeing carry out
// calculation below shall be done once

```
IF LV_ST = 0 -> 1  and  LV_ST_END = 0
    FAC_LAM_TMP_ST_LAM_AD_CUS[i] =
            FAC_LAM_ADJ_LAM_AD_CUS[i]*IP_FAC_COR_ST_LAM_AD_CUS
```

```
        FAC_LAM_ADJ_ST_LAM_AD_CUS[i] =
             MIN(MAX(FAC_LAM_TMP_ST_LAM_AD_CUS[i]
             ,C_FAC_LAM_ADJ_ST_MIN_LAM_AD_CUS),
             C_FAC_LAM_ADJ_ST_MAX_LAM_AD_CUS)

        IF(a) LC_EGY_LAM_AD_CUS_ST_ENA = 1
        THEN(a)  %long term adaptation via injector needle lift correction
                For x= 0 to NC_CYL_NR
                  EGY_LAM_ADJ_COR_LAM_AD_CUS[x] =
                          FAC_LAM_ADJ_ST_LAM_AD_CUS[NC_LAMB_REF[x]]*
                          C_EGY_LAM_ADJ_LAM_AD_CUS
                Endfor

                FAC_LAM_ADJ_COR_LAM_AD_CUS[i] = 0     [%]

        ELSE(a) %long term adaptation via fuel mass setpoint correction
           EGY_LAM_ADJ_COR_LAM_AD_CUS[x] = 0   [mJ]
           FAC_LAM_ADJ_COR_LAM_AD_CUS[i]= FAC_LAM_ADJ_ST_LAM_AD_CUS[i]
        ENDIF(a)
    END




    // case:  no update beeing carry out
    // calculation of long term adaptation after start
    IF  LV_ST_END = 0 -> 1
        IF(b) LC_EGY_LAM_AD_CUS_ENA = 1
        THEN(b)  %long term adaptation via injector needle lift correction
                For x= 0 to NC_CYL_NR
                  EGY_LAM_ADJ_COR_LAM_AD_CUS[x] =
                          FAC_LAM_ADJ_LAM_AD_CUS[NC_LAMB_REF[x]]*
                          C_EGY_LAM_ADJ_LAM_AD_CUS
                Endfor

                FAC_LAM_ADJ_COR_LAM_AD_CUS[i] = 0     [%]

        ELSE(b) %long term adaptation via fuel mass setpoint correction
                EGY_LAM_ADJ_COR_LAM_AD_CUS[x] = 0   [mJ]
                FAC_LAM_ADJ_COR_LAM_AD_CUS[i] =FAC_LAM_ADJ_LAM_AD_CUS[i]
        ENDIF(b)
    ENDIF
```

Recurrence = 1s

// case: clear adaptions value for bank 1 by external request:

IF LV_AD_CLR_LONG_LAM_EXT_ADJ[1] = 1 **or** LC_AD_CLR_LONG_LAM_1 =1

```
THEN
            CTR_KM_DELTA_LAM_ADJ_LAM_AD_CUS        = 0
            CTR_KM_LAM_ADJ_LAM_AD_CUS              = 0
            CTR_LAM_ADJ_LAM_AD_CUS                 = 0
            FAC_LAM_ADJ_COR_LAM_AD_CUS[1]          = 0
            FAC_LAM_ADJ_LAM_AD_CUS[1]              = 0
            FAC_LAM_CYL_SEL_LAM_AD_CUS[x] = 0, x for all cyl. in bank 1
            EGY_LAM_ADJ_COR_LAM_AD_CUS[x] = 0, x for all cyl. in bank 1
ENDIF

// clear adaptions value for bank 2 by external request:

IF  LV_AD_CLR_LONG_LAM_EXT_ADJ[2] = 1 or  LC_AD_CLR_LONG_LAM_2  = 1
THEN
            CTR_KM_DELTA_LAM_ADJ_LAM_AD_CUS        = 0
            CTR_KM_LAM_ADJ_LAM_AD_CUS              = 0
            CTR_LAM_ADJ_LAM_AD_CUS                 = 0
            FAC_LAM_ADJ_COR_LAM_AD_CUS[2]          = 0
            FAC_LAM_ADJ_LAM_AD_CUS[2] = 0
            FAC_LAM_CYL_SEL_LAM_AD_CUS[x] = 0, x for all cyl. in bank 2
            EGY_LAM_ADJ_COR_LAM_AD_CUS[x] = 0, x for all cyl. in bank 2
ENDIF

// clear cylinder individual adaptions value when an injector change is detected

For x= 0 to NC_CYL_NR
  IF STATE_IV_CHG[x] = 1
  Then
     FAC_LAM_CYL_SEL_LAM_AD_CUS[x] = 0
   Endif
Endfor

IF any bit of STATE_IV_CHG[x] is set to 1
THEN // do the calculation below

     // The sum below shall be done for each exhaust gas bank over the cylinders belonging
     //   to it.
     FAC_LAM_ADJ_LAM_AD_CUS[i] =
       {ΣFAC_LAM_CYL_SEL_LAM_AD_CUS[x]} / (NC_CYL_NR / NC_CBK_EX_NR)
     //

ENDIF
```

Recurrence = 1s

// case: update to be carried out

```
IF(1) CTR_KM_CAN > CTR_KM_CAN_OLD & LC_LAM_ADJ_LAM_AD_CUS_ENA =1
THEN(1)
    CTR_KM_CAN_OLD = CTR_KM_CAN
    IF(1b) CTR_LAM_ADJ_LAM_AD_CUS  ≤ C_CTR_MAX_LAM_ADJ_LAM_AD_CUS
    OR(1b)  LC_CTR_MAX_LAM_ADJ_LAM_AD_CUS = 0
    THEN(1b)
        LV_LAM_ADJ_LAM_AD_CUS_ENA = 1
    ELSE(1b)
        LV_LAM_ADJ_LAM_AD_CUS_ENA = 0
    ENDIF(1b)
ELSE(1)
    LV_LAM_ADJ_LAM_AD_CUS_ENA = 0
    LV_CDN_DIAG_LAM_AD_CUS[i] = 0
ENDIF(1)
```

```
IF(2) LV_LAM_ADJ_LAM_AD_CUS_ENA = 1
THEN(2)  %update long term adaptation temporary counter
    CTR_KM_DELTA_LAM_ADJ_LAM_AD_CUS =
                                    CTR_KM_CAN - CTR_KM_LAM_ADJ_LAM_AD_CUS

    IF(3) CTR_KM_DELTA_LAM_ADJ_LAM_AD_CUS> C_CTR_KM_THD_LAM_AD_CUS
    THEN(3) % Long term adaptation enabled

       CTR_KM_DELTA_LAM_ADJ_LAM_AD_CUS= 0
       CTR_KM_LAM_ADJ_LAM_AD_CUS  = CTR_KM_CAN
       CTR_LAM_ADJ_LAM_AD_CUS  = CTR_LAM_ADJ_LAM_AD_CUS + 1

       IF(4) LV_LAM_AD_EXT = 1
           % long term adaptation based on the BMW KLANN adaptation values
           FAC_LAM_ADJ_TMP_LAM_AD_CUS[1] = Klann_mw1*100%
           FAC_LAM_ADJ_TMP_LAM_AD_CUS[2] = Klann_ mw2*100%
           LV_FAC_LAM_AD_CUS_SHIFT[i] = 0
       Else(4)
           % long term adaptation based on the SiemensVDO adaptation values
           FAC_LAM_ADJ_TMP_LAM_AD_CUS[1] = FAC_L_RNG_LAM_AD[1]
           FAC_LAM_ADJ_TMP_LAM_AD_CUS[2] = FAC_L_RNG_LAM_AD[2]
           LV_FAC_LAM_AD_CUS_SHIFT[i] = 1
       Endif(4)

       FAC_LAM_ADJ_MAX_LAM_AD_CUS[i] =
         FAC_LAM_ADJ_TMP_LAM_AD_CUS[i]* C_CRLC_LAM_ADJ_MAX_LAM_AD_CUS

       FAC_LAM_UPD_LAM_AD_CUS[i]_old = FAC_LAM_ADJ_LAM_AD_CUS[i]
```

```
  FAC_LAM_UPD_LAM_AD_CUS[i]_$_{new}$ =
  FAC_LAM_ADJ_MAX_LAM_AD_CUS[i]*C_CRLC_LAM_ADJ_LAM_AD_CUS  +
  FAC_LAM_UPD_LAM_AD_CUS[i]_$_{old}$*(1- C_CRLC_LAM_ADJ_LAM_AD_CUS )

  FAC_LAM_ADJ_LIM_LAM_AD_CUS[i]=
              MIN(MAX(FAC_LAM_UPD_LAM_AD_CUS[i],
              C_FAC_LAM_ADJ_MIN_LAM_AD_CUS),
              C_FAC_LAM_ADJ_MAX_LAM_AD_CUS)

  For x= 0 to NC_CYL_NR
      FAC_LAM_CYL_SEL_LAM_AD_CUS[x]_$_{old}$ =
              FAC_LAM_CYL_SEL_LAM_AD_CUS[x]

      FAC_LAM_CYL_SEL_LAM_AD_CUS[x]_$_{new}$ =
              FAC_LAM_CYL_SEL_LAM_AD_CUS[x]_$_{old}$ +
              (FAC_LAM_ADJ_LIM_LAM_AD_CUS[NC_LAMB_REF[x]]  -
              FAC_LAM_ADJ_LAM_AD_CUS[NC_LAMB_REF[x]])
  Endfor

  IF LV_FAC_LAM_AD_CUS_SHIFT[i] = 1
  THEN
      FAC_LAM_AD_CUS_SHIFT[i] =
              (FAC_LAM_ADJ_LIM_LAM_AD_CUS[i]  -
              FAC_LAM_ADJ_LAM_AD_CUS[i])
  ELSE
      FAC_LAM_AD_CUS_SHIFT[i] = 0
  ENDIF

// The sum below shall be done for each exhaust gas bank over the cylinders belonging
//   to it.


FAC_LAM_ADJ_LAM_AD_CUS[i] =
  {ΣFAC_LAM_CYL_SEL_LAM_AD_CUS[x]} / (NC_CYL_NR / NC_CBK_EX_NR)
//

  IF LC_EGY_LAM_AD_CUS_ENA = 1
  THEN  %long term adaptation via injector needle lift correction
      For x= 0 to NC_CYL_NR
         EGY_LAM_ADJ_COR_LAM_AD_CUS[x] =
              FAC_LAM_ADJ_LAM_AD_CUS[NC_LAMB_REF[x]]*
              C_EGY_LAM_ADJ_LAM_AD_CUS
      Endfor

      FAC_LAM_ADJ_COR_LAM_AD_CUS[i] = 0     [%]

  ELSE %long term adaptation via fuel mass setpoint correction
      EGY_LAM_ADJ_COR_LAM_AD_CUS[x] = 0   [mJ]
      FAC_LAM_ADJ_COR_LAM_AD_CUS[i] =FAC_LAM_ADJ_LAM_AD_CUS[i]
  ENDIF
```

```
        % Diagnosis of long term adaptation

        LV_CDN_DIAG_LAM_AD_CUS[i] = 1

        IF FAC_LAM_ADJ_LAM_AD_CUS[i] > C_FAC_LAM_ERR_MAX_LAM_AD_CUS
        OR FAC_LAM_ADJ_LAM_AD_CUS[i] < C_FAC_LAM_ERR_MIN_LAM_AD_CUS
        THEN
            ERR_SYM_LAM_AD_CUS[i] = "SYM_1"
            % anti-bounce counter is incremented automatically by the error management
        ELSE
            ERR_SYM_LAM_AD_CUS[i] = "NO_SYM"
            % anti-bounce counter is decremented automatically by the error management
        ENDIF

            % In case the anti-bounce counter achieves the calibrated max threshold the
        error bit LV_ERR_LAM_AD_CUS[i] will be set automatically by the error management


    ELSE(3)
        LV_CDN_DIAG_LAM_AD_CUS[i] = 0
    ENDIF(3)
ENDIF(2)

IF(4) LV_FAC_LAM_AD SHIFT_END[i] = 1
THEN(4)
    LV_FAC_LAM_AD_CUS_SHIFT[i] = 0
    FAC_LAM_AD_CUS_SHIFT[i] = 0
ENDIF(4)


IF LC_LAM_AD_CUS_CLR = 1
THEN clear anti-bounce counter
ENDIF
```

Configuration for diagnostic symptoms:

Diagnostic GS	Symptom description	Symptom	Filter type
Diagnosis for the Long Term Adaptation	-	*SYM_0*	STD
	Long term adaptation too high	*SYM_1*	
	-	*SYM_2*	
	-	*SYM_3*	

47 Lambda setpoint

Table of Contents

Index

47.1 Customer adaptation module: AGGR LASP

47.1.1 Outputs for BMW for SV aggregates

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_BAS[NC_CBK_EX_NR]	O/V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Bank selective basic lambda setpoint					
LAMB_BAS_H_RES[NC_CBK_EX_NR]	O/V	0...7FFFFH	0...31.99993	0.061e-3	[-]
Bank selective basic lambda setpoint with high resolution					
LAMB_HOM_AFL	O/V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda setpoint for homogeneous lean combustion					
LAMB_SP	O/V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda setpoint					
LAMB_SP[NC_CBK_EX_NR]	O/V	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda setpoint					
LAMB_SP_HOM[NC_CBK_EX_NR]	O/V	0...7FFFH	0...1.99993	0.061e-3	[-]
Lambda setpoint with high resolution					

Input data:

Dla_soll_puls[NC_CBK_EX_NR]	La_bas1	La_bas2	La_ref_homm
La_sollreg[NC_CBK_EX_NR]	NC_CBK_EX_NR		

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes.

Application conditions:

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Initialisation : 1

Recurrence: 10 ms

Except LAMB_HOM_AFL: 20 ms

Except LAMB_SP_HOM[1] and LAMB_SP_HOM[2], which shall be segment synchronous; LAMB_SP_HOM[1] shall be synchronised with the update rate of La_sollreg1 and similarly LAMB_SP_HOM[2] with La_sollreg2.

Except LAMB_SP[1], LAMB_SP[2], LAMB_SP: segment synchronous

Formula section:

LAMB_SP_HOM[1] = La_sollreg[1] – Dla_soll_puls[1]
LAMB_SP_HOM[2] = La_sollreg[2] – Dla_soll_puls[2]
LAMB_SP[1] = La_sollreg[1] – Dla_soll_puls[1]
LAMB_SP[2] = La_sollreg[2] – Dla_soll_puls[2]
LAMB_SP = (LAMB_SP[1]+ LAMB_SP[2]) / 2

LAMB_HOM_AFL = La_ref_homm
LAMB_BAS_H_RES[NC_CBK_EX_NR] = La_ref_homm
LAMB_BAS[1] = La_bas1
LAMB_BAS[2] = La_bas2

48 Messurement and Calibration Protocol Communication

Table of Contents

Index

48.1 Multiplexed segment-synchronous output variables

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_IGC_0_5_H_RNG	V/O	FA60h..5A0h	-90…90	0.0625	[°CRK]
Ignition angle applied on the respective cylinder (0 to 5)					
IGA_KNK_0_5	V	0...80H	-48...0	0.375	[°CRK]
Spark retard due to knocking combustion of the respective cylinder (0 to 5)					
KNK_THD_0_5	V	0...FFH	0...4.998	1.95E-02	[V]
Knock threshold of the respective cylinder (0 to 5)					
KNKS_0_5	V	0...FFFFH	0...4.99992	7.63E-05	[V]
Knock control signal of the respective cylinder (0 to 5)					
NL_0_5	V	0...FFFFH	0...4.99992	7.63E-05	[V]
Noise level of the respective cylinder (0 to 5)					
TI_1_HOM_0_5	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, homogeneous mode, first pulse (0 to 5)					
TI_2_HOM_0_5	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, homogeneous mode, second pulse (0 to 5)					
TI_1_S_0_5	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, stratified mode, fist pulse (0 to 5)					
TI_2_S_0_5	V	0...FFFFH	0...65.535	0.001	[ms]
Cylinder individual injection time, stratified mode, second pulse (0 to 5)					
KNKS_REL_NL_0_5	V/O	0...FFFFH	0...0.99998	1.53E-05	[-]
Relative knock value					

Input data:

IGA_IGC_H_RNG[NC_CYL_NR]	IGA_KNK[NC_CYL_NR]	NL[NC_CYL_NR]	KNK_THD[NC_CYL_NR]
KNKS[NC_CYL_NR]	C_IGA_INI_H_RNG	C_KNK_THD_MAX	KNKS_REL_NL[NC_CYL_NR]
TI_1_HOM[NC_CYL_NR]	TI_2_HOM[NC_CYL_NR]	TI_1_S[NC_CYL_NR]	TI_2_S[NC_CYL_NR]

FUNCTION DESCRIPTION:

General information:

The purpose of introducing multiplexed segment-synchronous variables is to be able to “watch” several online-values (segment-synchronous) via only one variable during calibration. This way the number of used online-variables can be reduced during measurements while still having the possibility to measure the data of all cylinders.

Description:

E.g.: The variable IGA_0_5 shows the ignition angle of the currently firing cylinder as online-output-data by writing the different succeeding IGA-values (firing order) into the respective segments of the variable.

Application conditions:

Initialisation: IGA_IGC_0_5_H_RNG = C_IGA_INI _H_RNG

KNK_THD_0_5 = 5V at reset and engine running to engine stop
IGA_KNK_0_5 initialised as IGA_KNK_x
KNKS_0_5 initialised as KNKS_x
NL_0_5 = C_KNK_THD_MAX at reset or engine running to engine stop
TI_1/2_HOM/S_0_5 = 0 ms
KNKS_REL_NL_0_5 = initialised as KNKS_REL_NL_x

Recurrence: (update rate) every segment

Activation: every engine operating state

Deactivation: ------

Formula section:

IGA_IGC_0_5_H_RNG:	IGA_IGCH_RNG of cylinder x = value segment x
IGA_KNK_0_5:	IGA_KNK of cylinder x = value segment x
KNK_THD_0_5:	KNK_THD of cylinder x = value segment x
KNKS_0_5:	KNKS of cylinder x = value segment x
NL_0_5:	NL of cylinder x = segment x
TI_1_HOM_0_5:	calculated out of TI_1_HOM[NC_CLY_NR]
TI_2_HOM_0_5:	calculated out of TI_2_HOM[NC_CLY_NR]
TI_1_S_0_5:	calculated out of TI_1_S[NC_CLY_NR]
TI_2_S_0_5:	calculated out of TI_2_S[NC_CLY_NR]
KNKS_REL_NL_0_5:	KNKS_REL_NL x = segment x

49 Minimal fuel mass adaptation

Table of Contents

Index

49.1 MFMA LAM_AD_INJ Configuration Data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_PHA_LAM_AD_INJ	O	0...FFH	0...255	1	[-]
Number of phases for lambda adaptation via injection mode					
NC_NR_MIS_REP_LAM_AD_INJ	O	0...FFH	0...255	1	[-]
Maximum number of interruptions of lambda adapation via injection mode due to misfire per cylinder					
NC_NR_MPL_INJ_LAM_AD_INJ	O	0...FFH	0...255	1	[-]
Number of multiple injections for lambda adaptation via injection mode					

General information:

The following describes the general rules for determination of the configuration data

49.1.1 Global Configuration Data

Here are listed the configuration data, which can be used in other aggregates:

Data	Value

49.1.2 Local Configuration Data

Here are listed the configuration data, which are used:

Data	Value
NC_NR_PHA_LAM_AD_INJ	7
NC_NR_MIS_REP_LAM_AD_INJ	10
NC_NR_MPL_INJ_LAM_AD_INJ	3

49.2 Activation of Lambda adaptation via injection mode

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_ABC_DET_MIS_LAM_AD_INJ [NC_CBK_EX_NR]	V	0... FFH	0... 255	1	[-]
Counter for misfire events during lambda adaptation for injection mode					
CTR_AD_COLD_LAM_AD_INJ	O/V/S	0... FFH	0... 255	1	[-]
Counter for finished adaptation cycles for cold adaptation for lambda adaptation via injection mode					
CTR_AD_HOT_LAM_AD_INJ	O/V/S	0... FFH	0... 255	1	[-]
Counter for finished adaptation cycles for hot adaptation for lambda adaptation via injection mode					
CTR_MIS_AD_MISS_COLD_LAM_AD_INJ [NC_CYL_NR]	V	0... FFH	0... 255	1	[-]
Counter for missed adaptation cycles by misfire in a driving cycle at cold temperature for lambda adaptation via injection mode					
CTR_MIS_AD_MISS_HOT_LAM_AD_INJ [NC_CYL_NR]	V	0... FFH	0... 255	1	[-]
Counter for missed adaptation cycles by misfire in a driving cycle at hot temperature for lambda adaptation via injection mode					
CTR_MIS_DC_COLD_LAM_AD_INJ [NC_CYL_NR]	V/S	0... FFH	0... 255	1	[-]
Number of driving cycles with not finished adaptations due to misfire at cold temperature for lambda adaptation via injection mode					
CTR_MIS_DC_HOT_LAM_AD_INJ [NC_CYL_NR]	V/S	0... FFH	0... 255	1	[-]
Number of driving cycles with not finished adaptations due to misfire at hot temperature for lambda adaptation via injection mode					
CTR_MIS_LAM_AD_INJ [NC_CBK_EX_NR]	O/V	0... FFH	0... 255	1	[-]
Number of following adaptaions with misfire detection at lambda adaptation via injection mode					
DIST_LAM_AD_INJ_COLD	V/S	0... FFFFH	0... 524280	8	[km]
Distance threshold for next start of lambda adaptation via injection mode at cold condition					
DIST_LAM_AD_INJ_HOT	V/S	0... FFFFH	0... 524280	8	[km]
Distance threshold for next start of lambda adaptation via injection mode at hot condition					
FAC_LAM_MV_OFS_LDC_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Filtered lambda controller output offset for limited dynamics calculation of lambda adaptation via injection mode					
FAC_TQ_REQ_OFS_LDC_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
Torque scaling factor offset for limited dynamics calculation of lambda adaptation via injection mode					
LV_ERR_MIS_LAM_AD_INJ [NC_CYL_NR]	O/V/S	0... 1H	0... 1	1	[-]
Flag indicating a repeating misfire error during lambda adaptation via injection mode					
LV_LAM_AD_INJ_ACT	O/V	0... 1H	0... 1	1	[-]
Lambda adaptation via injection mode active					
LV_LAM_AD_INJ_ACT_COLD [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Activation of lambda adaptation via injection mode for lower temperature range					
LV_LAM_AD_INJ_ACT_HOT [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Activation of lambda adaptation via injection mode for upper temperature range					
LV_LAM_AD_INJ_EXT_ENA	O/V	0... 1H	0... 1	1	[-]
Set the external enable the lambda adaption via injection mode with a test device					
LV_LAM_AD_INJ_INTR	O/V	0... 1H	0... 1	1	[-]
Interuption of adaptation for lambda adaptation via injection mode					
LV_LAM_AD_INJ_INTR_2 [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Maximum number of adaptations with misfire reached in adaptation cycle for lambda adaptation via injection mode					
LV_LAM_AD_INJ_INTR_LAM	O/V	0... 1H	0... 1	1	[-]
Interuption of adaptation for lambda adaptation via injection mode due to lambda controller threshold					
LV_LAM_AD_INJ_REP	O/V	0... 1H	0... 1	1	[-]
Repeat of adaptation for lambda adaptation via injection mode					

LV_LDC_LAM_AD_INJ [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Limited dynamic conditions for lambda adaptation via injection mode					
LV_MIS_LAM_AD_INJ [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Misfire detected during adaptation for lambda adaptation via injection mode					
MAF_INT_LDC_LAM_AD_INJ [NC_CBK_EX_NR]	V	0... FFFFH	0... 1820.41666667	0.0277778	[g]
Air mass flow integral during limited dynamics for lambda adaptation via injection mode					
MAF_OFS_LDC_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Air mass flow offset for limited dynamics calculation for lambda adaptation via injection mode					
N_OFS_LDC_LAM_AD_INJ [NC_CBK_EX_NR]	V	E020... 1FE0H	-8160 ...8160	1	[rpm]
Engine speed offset for limited dynamics calculation of lambda adaptation via injection mode					
TEMP_LAM_AD_INJ	O/V	0... FEH	-48... 142.5	0.75	[°C]
Temperature for lambda adaptation via injection mode					

Input Data:

LV_INH_LAM_AD_INJ [NC_CBK_EX_NR]	LV_MPL_INJ_ACT [NC_CBK_EX_NR]	STATE_LAM_AD_INJ_ACT [NC_CBK_EX_NR]	LV_LAM_AD_INJ_COLD_END [NC_CBK_EX_NR]
LV_LAM_AD_INJ_HOT_END [NC_CBK_EX_NR]	NR_CYL_LAM_AD_INJ [NC_CBK_EX_NR]	TQ_IS_SNG_INJ_LAM_AD_INJ	LV_LAM_AD_INJ_MV_CLC_END [NC_CBK_EX_NR]
LV_SNG_INJ_ACT [NC_CBK_EX_NR]	C_CRLC_TQ_DIF_SLOW_LAM_AD_INJ	N_32	MFF_SP_S_SWI_HOM
MFF_SP_MV	TCO	TFU_IV	FAC_LAM_MV [NC_CBK_EX_NR]
NC_CBK_EX_NR	T_AST	TFU_INJ	N_DELTA_LDC
MAF_DELTA_LDC	FAC_TQ_REQ_DELTA_LDC	FAC_LAM_MV_DELTA_LDC [NC_CBK_EX_NR]	MAF_CYL
LV_HOM_ACT	LV_CYL_BAL_LAM_SEL_AD_HOT_DC	LV_CYL_BAL_LAM_SEL_AD_COLD_DC	DIST_KWP
LV_DET_CFM_MIS	SEG_NR_ER	TQ_DIF_I_IS	TQ_DIF_P_D_SLOW_IS
NC_CYL_NR	TEMP_CAT_DYN_MDL [NC_CBK_EX_NR]	STATE_IV_CHG	TEMP_CAPA_IV_MV
LV_LAM_AD_INJ_CLR_AD_EXT	LV_CAT_PURGE_ACT [NC_CBK_EX_NR]	TCO_ST	LC_MFF_ADD_LAM_AD_INJ_CLR

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CTR_ABC_MAX_MIS_LAM_AD_INJ	1	0... FFH	0... 255	1	[-]
Threshold of misfire events to interrupt an adaptation for lambda adaptation via injection mode					
C_CTR_MAX_DC_LAM_AD_INJ	1	0... FFH	0... 255	1	[-]
Number of driving cycles with missed adaptation by misfire to set error flag for lambda adaptation via injection mode					
C_CTR_MIS_AD_MAX_LAM_AD_INJ	1	0... FFH	0... 255	1	[-]
Maximum numbr of missed adaptation in a driving cycle to start lambda adaptation via injection mode					
C_CTR_MIS_LAM_DC_MIN_LAM_AD_INJ	1	0... FFH	0... 255	1	[-]
Number of adaptation cycles with misfire to increment misfire driving cycle counter for lambda adaptation via injection mode					
C_CTR_MIS_MAX_LAM_AD_INJ	1	0... FFH	0... 255	1	[-]
Maximum number of adaptions with misfire for one cylinder in an adaptation cycle for lmbda adaptation via injection mode					
C_FAC_LAM_LDC_MPL_LAM_AD_INJ	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Lambda controller window for limited dynamic conditions at multiple injection for lambda adaptation via injection mode					
C_FAC_LAM_LDC_SNG_LAM_AD_INJ	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Lambda controller window for limited dynamic conditions at single injection for lambda adaptation via injection mode					

C_FAC_LAM_MV_MAX_AD_LAM_AD_INJ	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Maximum lambda controller threshold to interrupt the lambda adaptation via injection mode					
C_FAC_LAM_MV_MAX_LAM_AD_INJ	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Maximum limit on mean value of lambda controller output for lambda adaptation via injection mode					
C_FAC_LAM_MV_MIN_AD_LAM_AD_INJ	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Minimum lambda controller threshold to interrupt the lambda adaptation via injection mode					
C_FAC_LAM_MV_MIN_LAM_AD_INJ	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Minimum limit on mean value of lambda controller output for lambda adaptation via injection mode					
C_FAC_TQ_REQ_DYW_LDC_LAM_AD_INJ	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Torque scaling factor window for limited dynamic conditions Lambda adaptation via injection mode					
C_MAF_DYW_LDC_LAM_AD_INJ	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Air mass flow window for limited dynamic conditions of lambda adaptation via injection mode					
C_MAF_INT_LDC_LAM_AD_INJ	1	0... FFFFH	0... 1820.41666667	0.0277778	[g]
Air mass flow integral for duration of violation of limited dynamic conditions for lambda adaptation via injection mode					
C_N_BOL_LAM_AD_INJ	1	0... FFH	0... 8160	32	[rpm]
Lower limit on engine speed for lambda adaptation via injection mode					
C_N_DYW_LDC_LAM_AD_INJ	1	0... 1FE0H	0... 8160	1	[rpm]
Engine speed window for limited dynamic conditions trim control					
C_N_TOL_LAM_AD_INJ	1	0... FFH	0... 8160	32	[rpm]
Upper limit on engine speed for lambda adaptation via injection mode					
C_NR_MIN_AD_LAM_AD_INJ_COLD_ADD	1	0... FFH	0... 255	1	[-]
Minimum number of hot adaptations before the cold adaptation starts					
C_TCO_MIN_LAM_AD_INJ	1	0... FEH	-48... 142.5	0.75	[°C]
Minimum TCO to start the lambda adaptation via injection mode					
C_TEMP_CAT_MAX_LAM_AD_INJ	1	0... 7FFFH	-273.15... 1774.7875	0.0625	[°C]
Maximum catalyst temperature to start the lambda adaptation via injection mode					
C_TEMP_LAM_AD_INJ_ENA	1	0 1 2 3	TCO TFU_IV TFU_INJ TEMP_CAPA_IV_MV	-	[-]
Temperature for calculation in lambda adaptation via injection mode					
C_TEMP_MAX_LAM_AD_INJ_COLD	1	0... FEH	-48... 142.5	0.75	[°C]
Maximum temperature for cold adaptation range for lambda adaptation via injection mode					
C_TEMP_MAX_LAM_AD_INJ_HOT	1	0... FEH	-48... 142.5	0.75	[°C]
Maximum temperature for hot adaptation range for lambda adaptation via injection mode					
C_TEMP_MIN_LAM_AD_INJ_COLD	1	0... FEH	-48... 142.5	0.75	[°C]
Minimum temperature for cold adaptation range for lambda adaptation via injection mode					
C_TEMP_MIN_LAM_AD_INJ_HOT	1	0... FEH	-48... 142.5	0.75	[°C]
Minimum temperature for hot adaptation range for lambda adaptation via injection mode					
C_TQ_DIF_IS_AD_LAM_AD_INJ	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Idle speed controller threshold to repeat a adaptation of lambda adaptation via injection mode					
ID_MFF_SP_BOL_LAM_AD_INJ	6	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDPM_MFF_SP_1_MFMA	6	0... FFH	0... 8160	32	[rpm]
Lower limit on MFF_SP for lambda adaptation via injection mode					
ID_MFF_SP_TOL_LAM_AD_INJ	6	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDPM_MFF_SP_1_MFMA	6	0... FFH	0... 8160	32	[rpm]
Upper limit on MFF_SP for lambda adaptation via injection mode					
IP_DIST_LAM_AD_INJ_COLD	2*4	0... FFFFH	0... 524280	8	[km]
LDP_CTR_COLD_IP_DIST_LAM_AD_INJ	4	0... FFH	0... 255	1	[-]
LDPM_DIST_1_LAM_AD_INJ	2	0... FFFFH	0... 524280	8	[km]

Update of cold distance threshold for lambda adaptaiton via injection mode					
IP_DIST_LAM_AD_INJ_HOT	2*4	0... FFFFH	0... 524280	8	[km]
LDP_CTR_HOT_IP_DIST_LAM_AD_INJ	4	0... FFH	0... 255	1	[-]
LDPM_DIST_1_LAM_AD_INJ	2	0... FFFFH	0... 524280	8	[km]
Update of hot distance threshold for lambda adaptaiton via injection mode					
IP_T_AST_ENA_LAM_AD_INJ	8	0... FFFFH	0... 6553.5	0.1	[s]
LDP_TCO_ST_IP_T_AST_LAM_AD_INJ	8	0... FEH	-48... 142.5	0.75	[°C]
Minimum time for the activation of lambda adaptation via injection mode depending on TCO_ST					
LC_CTR_MIS_DC_LAM_AD_INJ_CLR	1	0... 1H	0... 1	1	[-]
Enable to clear the CTR_MIS_DC_COLD/HOT_LAM_AD_INJ at finished learning of all cylinders					
LC_CYL_BAL_COLD_DEAC_LAM_AD_INJ	1	0... 1H	0... 1	1	[-]
Deactivation of CILC condition for cold lambda activation via injection mode					
LC_CYL_BAL_HOT_DEAC_LAM_AD_INJ	1	0... 1H	0... 1	1	[-]
Deactivation of CILC condition for hot lambda activation via injection mode					
LC_DIST_CLR_LAM_AD_INJ	1	0... 1H	0... 1	1	[-]
Clear the distance threshold counter for lambda adaptation via injection mode					
LC_ERR_CLR_LAM_AD_INJ	1	0... 1H	0... 1	1	[-]
Clear the error in lambda adaptation via injection mode					
LC_LAM_AD_INJ_ACT_COLD_MAN	1	0... 1H	0... 1	1	[-]
Manual activation of cold conditons for lambda adaptation via injection mode					
LC_LAM_AD_INJ_ACT_HOT_MAN	1	0... 1H	0... 1	1	[-]
Manual activation of hot conditons for lambda adaptation via injection mode					

Export Actions:

ACTION_MFMA_GetEnableCondition(**OUT** <PRM_LV_LAM_AD_INJ_EXT_ENA>)
Enable of the lambda adaptation via injection mode
ACTION_MFMA_SetEnableCondition(**IN** <PRM_FLAG_PAR_IN>)
Activates lambda adaption via tester

Description for Actions

ACTION_MFMA_GetEnableCondition(**OUT** <PRM_LV_LAM_AD_INJ_EXT_ENA>)					
Enable of the lambda adaptation via injection mode					
Parameter	**Type**	**Hex.Limits**	**Phys.Limits**	**Resol.**	**Unit**
PRM_LV_LAM_AD_INJ_EXT_ENA	OUT	0... 1H	0... 1	1	[-]
Set the external enable the lambda adaption via injection mode with a test device					

ACTION_MFMA_SetEnableCondition(**IN** <PRM_FLAG_PAR_IN>)					
Activates lambda adaption via tester					
Parameter	**Type**	**Hex.Limits**	**Phys.Limits**	**Resol.**	**Unit**
PRM_FLAG_PAR_IN	IN	0... 1H	0... 1	1	[-]
Set the flag FLAG_PAR_IN to 1 if a tester device activate thbe lambda adaptation via injection mode					

General Information

This module handles the activation for the upper and lower temperature range for the lambda adaptation via injection mode and the engine speed controller, the lambda controller and the misfire detection are monitored during the adaptation.

Application Conditions

Initialization: RST, IGKON, NVMINI, NVMRES, NVMSTO
Activation: 20MS: always
SEG: always
ES2ERU: always
Deactivation: never

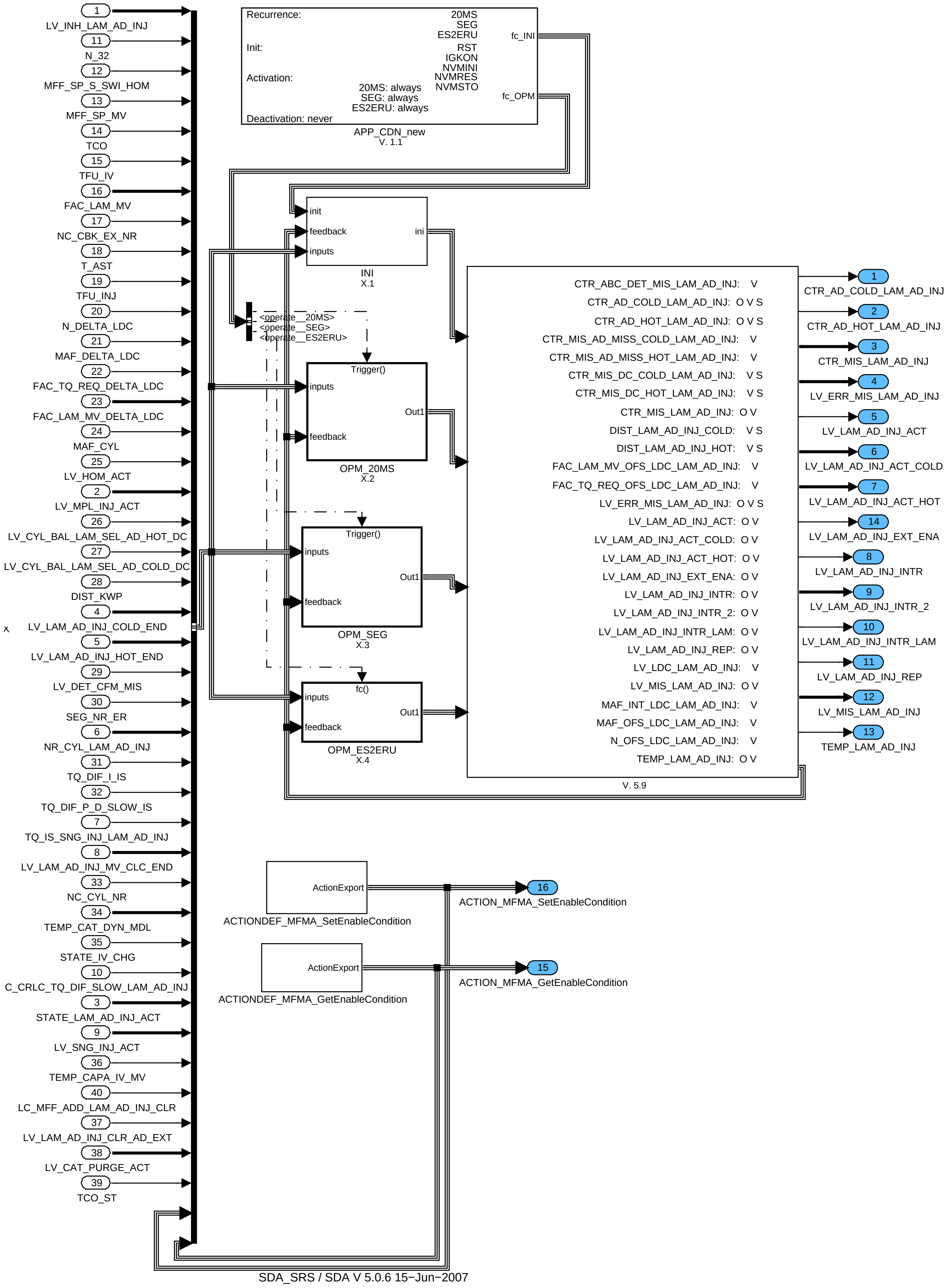


Figure 1:
Path: MFMA_ISPCLAIMA0

49.2.1 Initialization

Initialization of the variables at ignition on and reading and writing of the variables stored in the non volatile memory.

49.2.1.1 Initialization of the online variables at reset

All variables are initialized with 0 at reset.

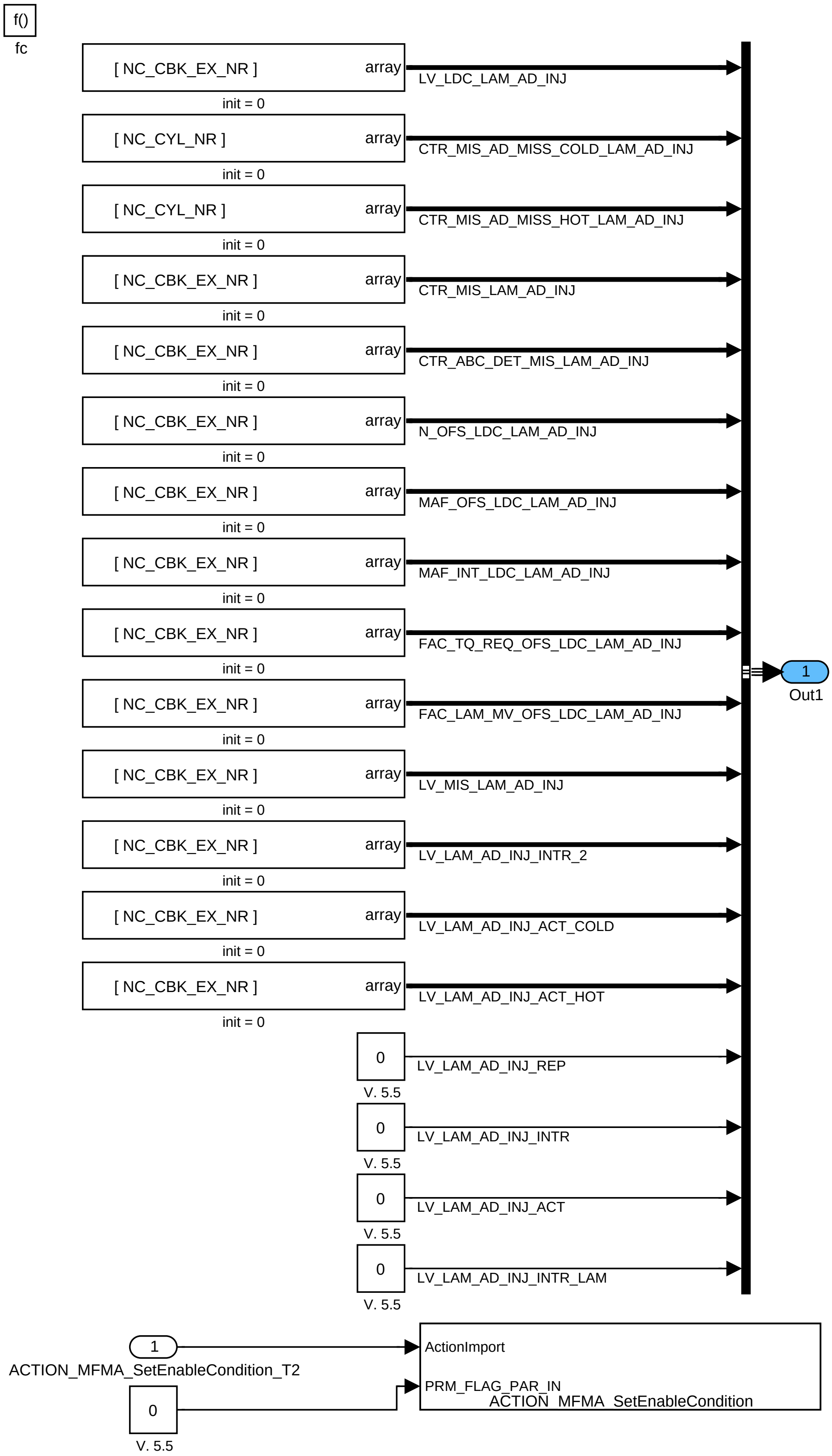


Figure 2:
Path: MFMA_ISPCLAIMA0/INI/RST

49.2.1.2 Initialization of the online variables at IGK on

All values, except CTR_MIS_AD_MISS_COLD/HOT_LAM_AD_INJ, are initialized with 0 at IGKON.

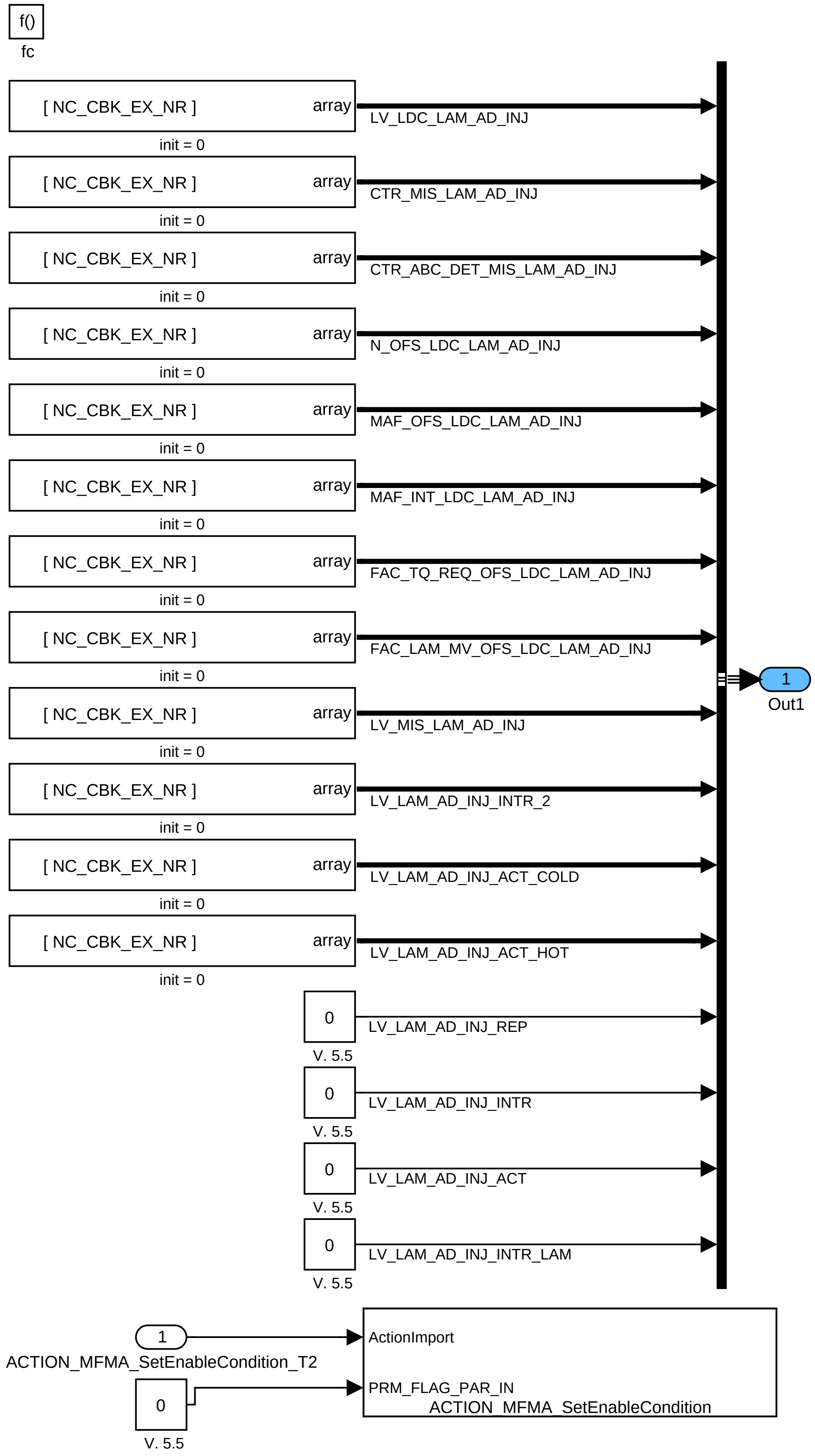


Figure 3:
Path: MFMA_ISPCLAIMA0/INI/IGKON

49.2.1.3 Initialization and reading of non volatile memory variables

The values are read from the NVMY memory. The first initialization of the values is done in an other subsystem, DIST_LAM_AD_INJ_COLD/HOT are not initialized with zero.

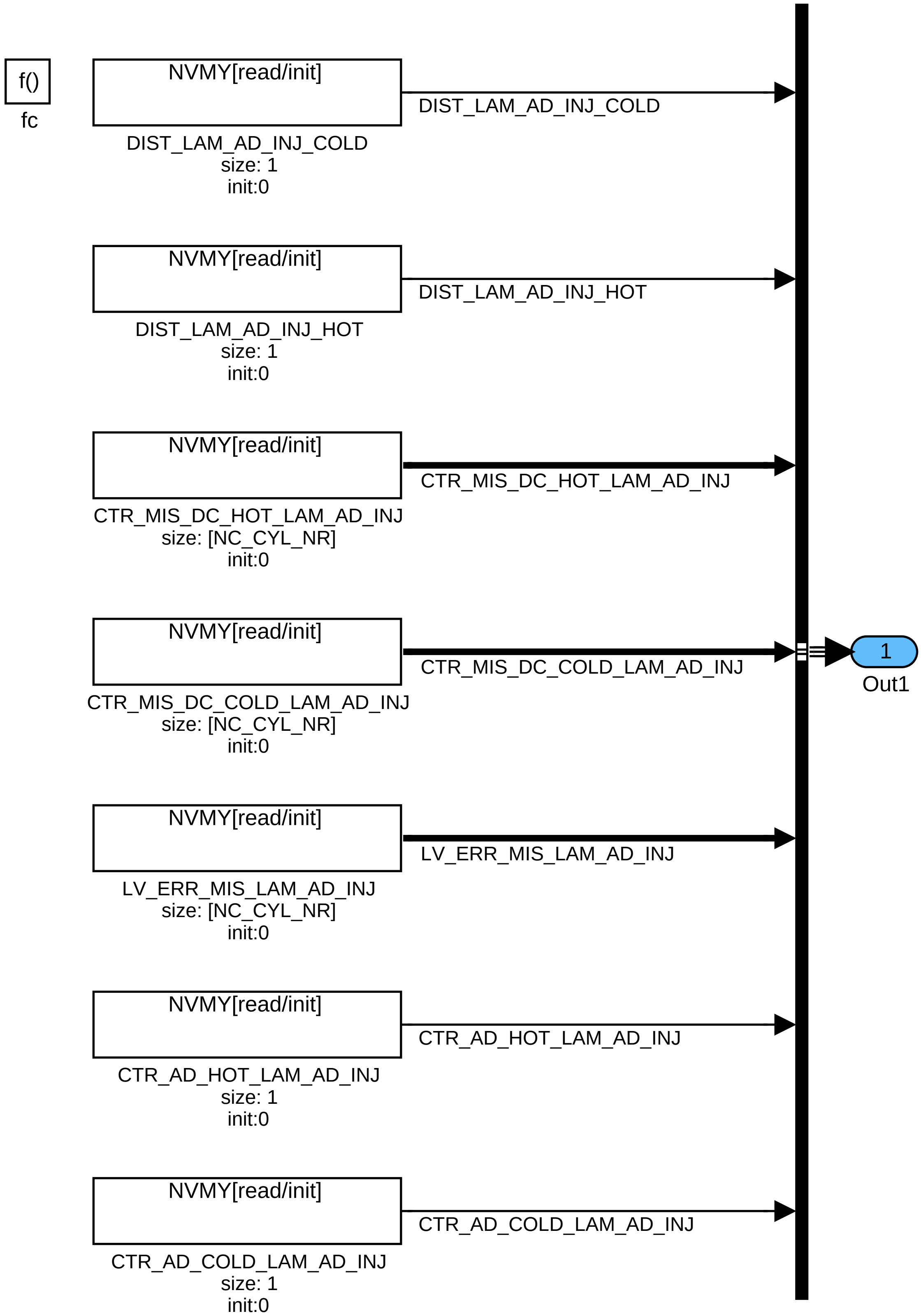


Figure 4:
Path: MFMA_ISPCLAIMA0/INI/NVMINI_NVMRES

49.2.1.4 Writing of non volatile memory variables

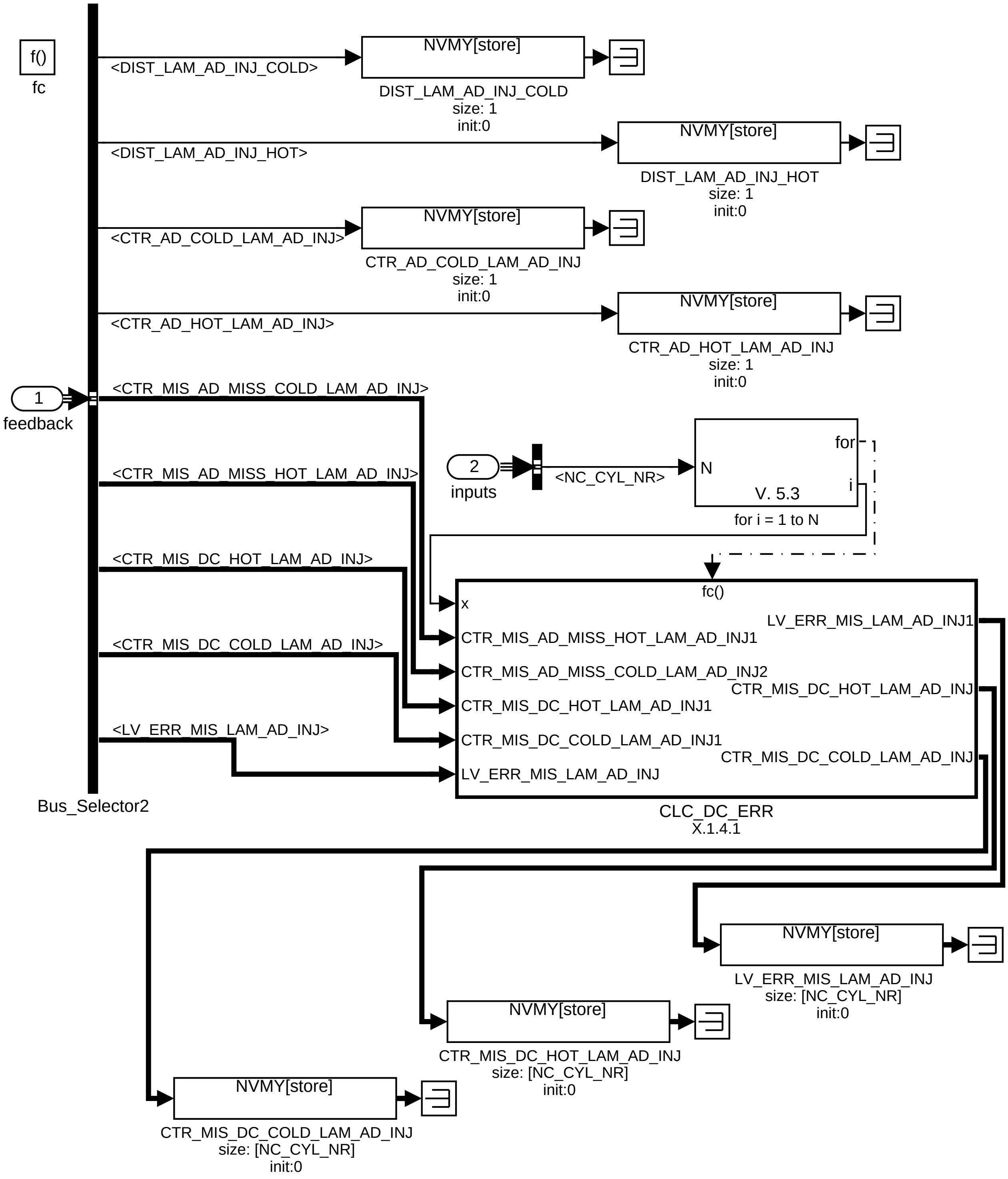


Figure 5:
Path: MFMA_ISPCLAIMA0/INI/NVMSTO

49.2.1.4.1 Calculation of driving cycle counter and creation of error flags

For each cylinder the number of driving cycles with missed adaptation due to misfire are counted. If the counter exceeds a threshold, the error flag will be set at the end of a driving cycle.

49.2.1.4.1.1 Calculation of driving cycle counter and error flags

If the number of missed adaptation cycles due to misfire in a driving cycle exceeds the threshold C_CTR_MIS_LAM_DC_MIN_LAM_AD_INJ, the counter for driving cycles with missed adaptation is increased. If the number of driving cycles with failed adaptations reaches the threshold C_CTR_MAX_DC_LAM_AD_INJ, the error flag for the depending cylinder is set.

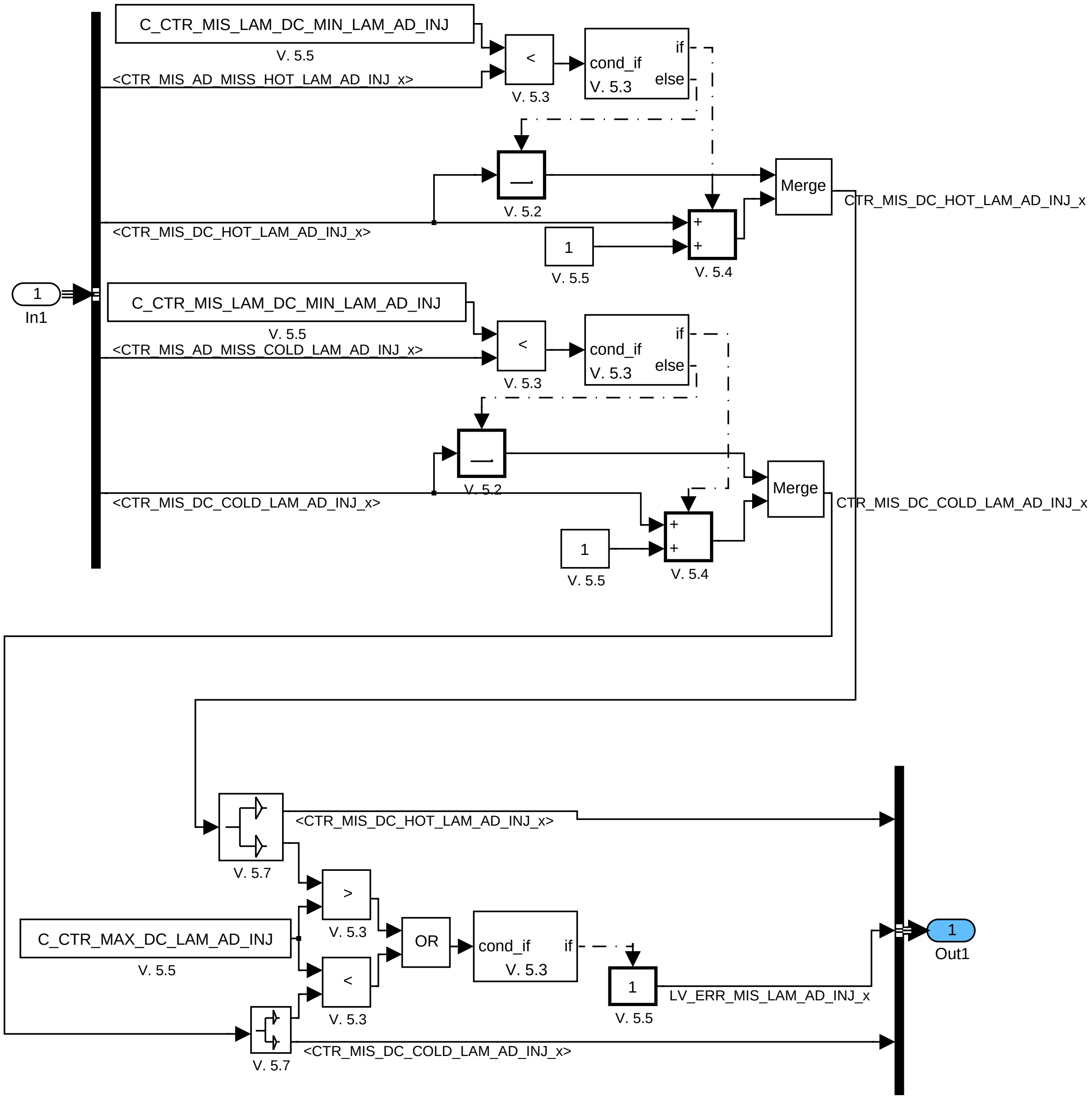


Figure 6:
Path: MFMA_ISPCLAIMA0/INI/NVMSTO/CLC_DC_ERR/CLC_CTR_MIS_DC

49.2.1.5 NVMY initialization

The distance counter is initialized at non volatile memory initialization with the sum of the mileage counter and the output of the map IP_DIST_LAM_AD_INJ_HOT/COLD. With this calculation the first adaptation after NVMY initialization will start after a mileage distance defined in the map. All other values are initialized with zero.

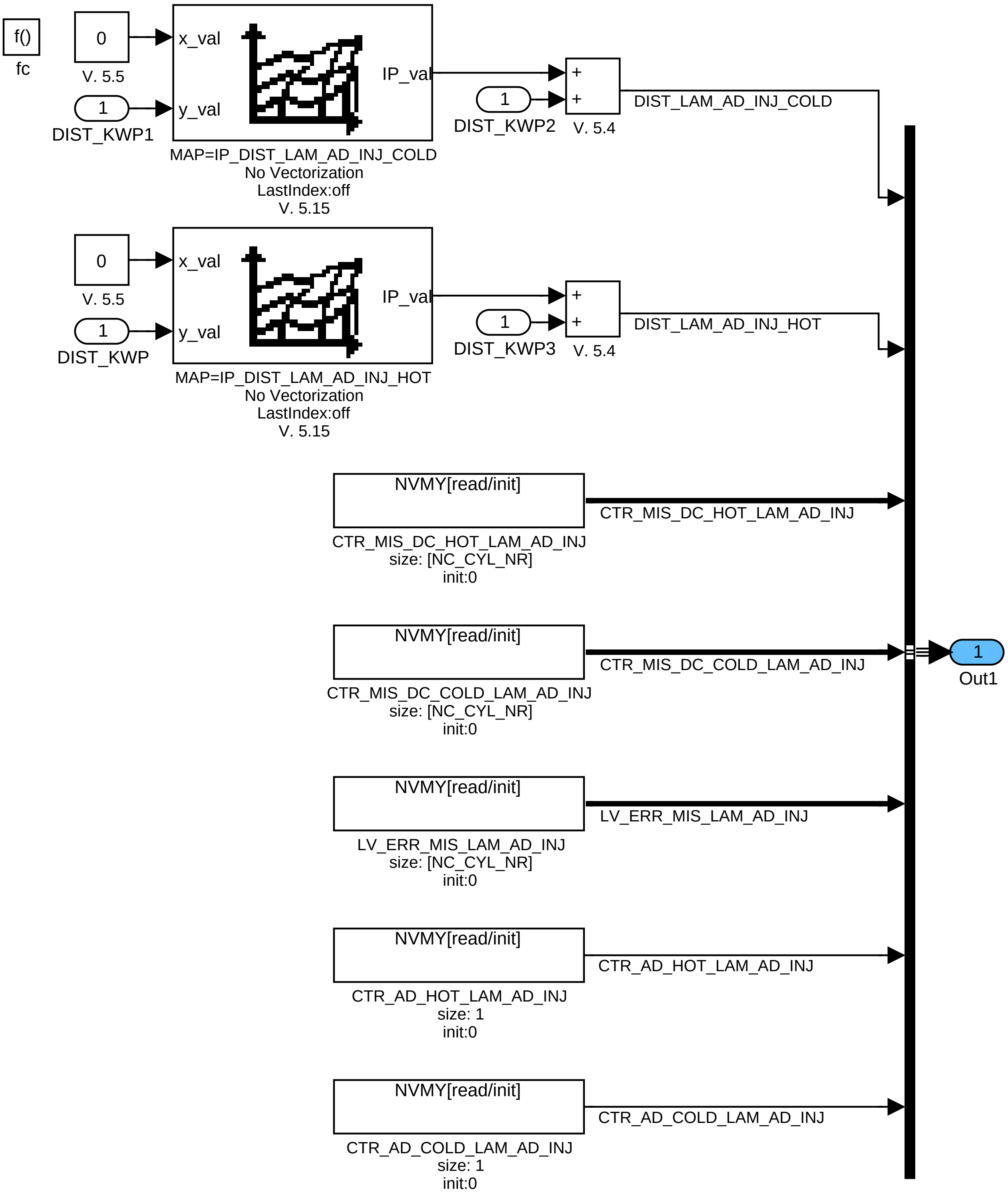


Figure 7:
Path: MFMA_ISPCLAIMA0/INI/NVMINI_NVMINI

49.2.2 Determination of the adaptation conditions

The determination of adaptation conditions is divided into bank specific and bank independent calculations.

49.2.2.1 Bank independent calculations

The bank independent calculations are divided in general calculations, temperature calculations and the clearance of interruption and repeat flags.

49.2.2.1.1 General activation conditions

For activation of the adaptation the fuel mass flow and the engine speed has to be inside limitations. Also a certain time after engine start has to be passed.

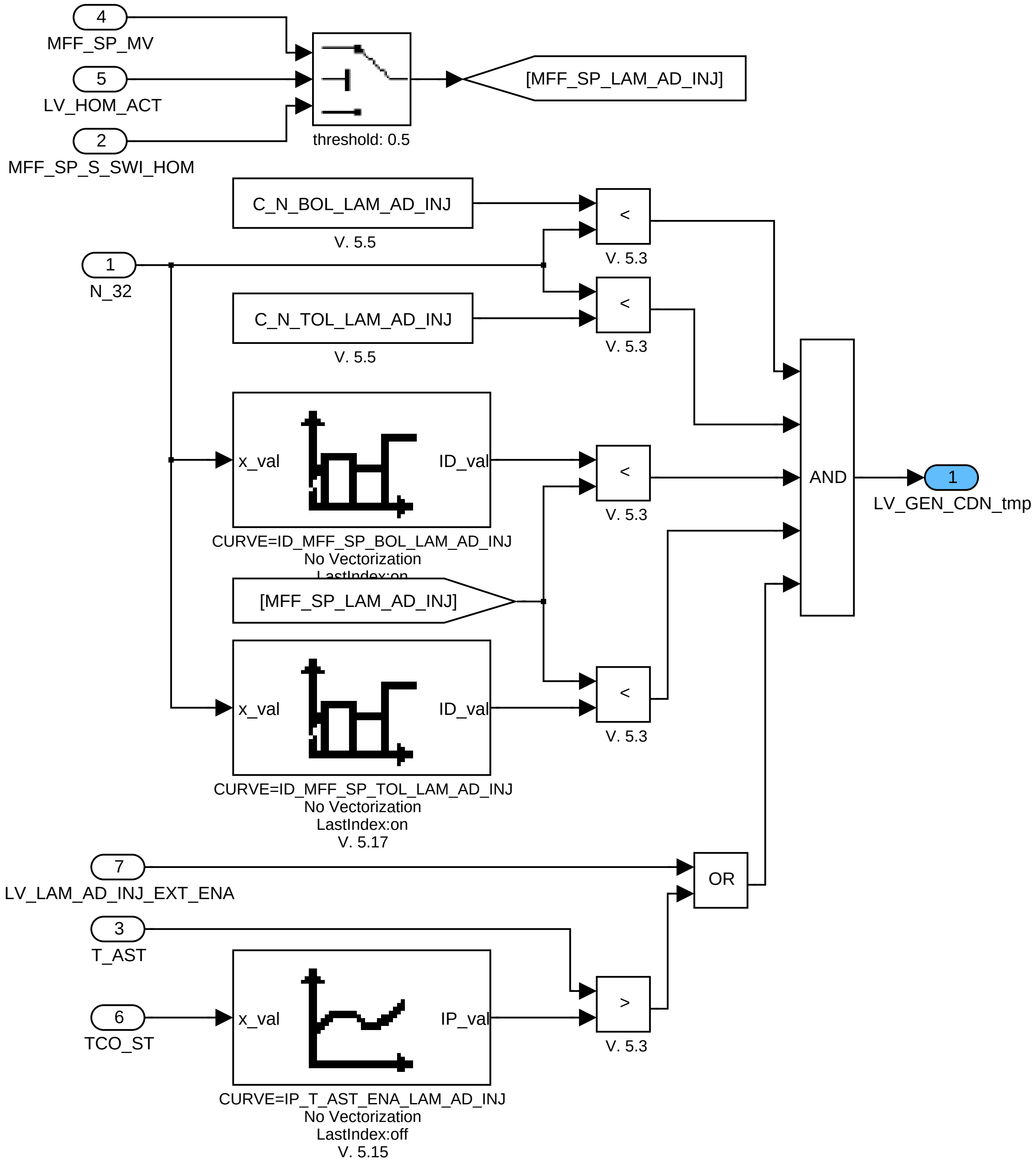


Figure 8:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_INDEP_CLC/GEN_CDN

49.2.2.1.2 Temperature range activation conditions

To activate the adaptation for cold and hot adaptation range the depending temperature can bee chosen between coolant temperature and fuel temperature. For activation the temperature has to be inside limitations, the general conditions has to be fulfilled, the cylinder individual lambda controller has to be ready and the condition for the mileage counter has to be fulfilled. Also the maximum number of adaptation cycles in the driving cycle has not to be reached.

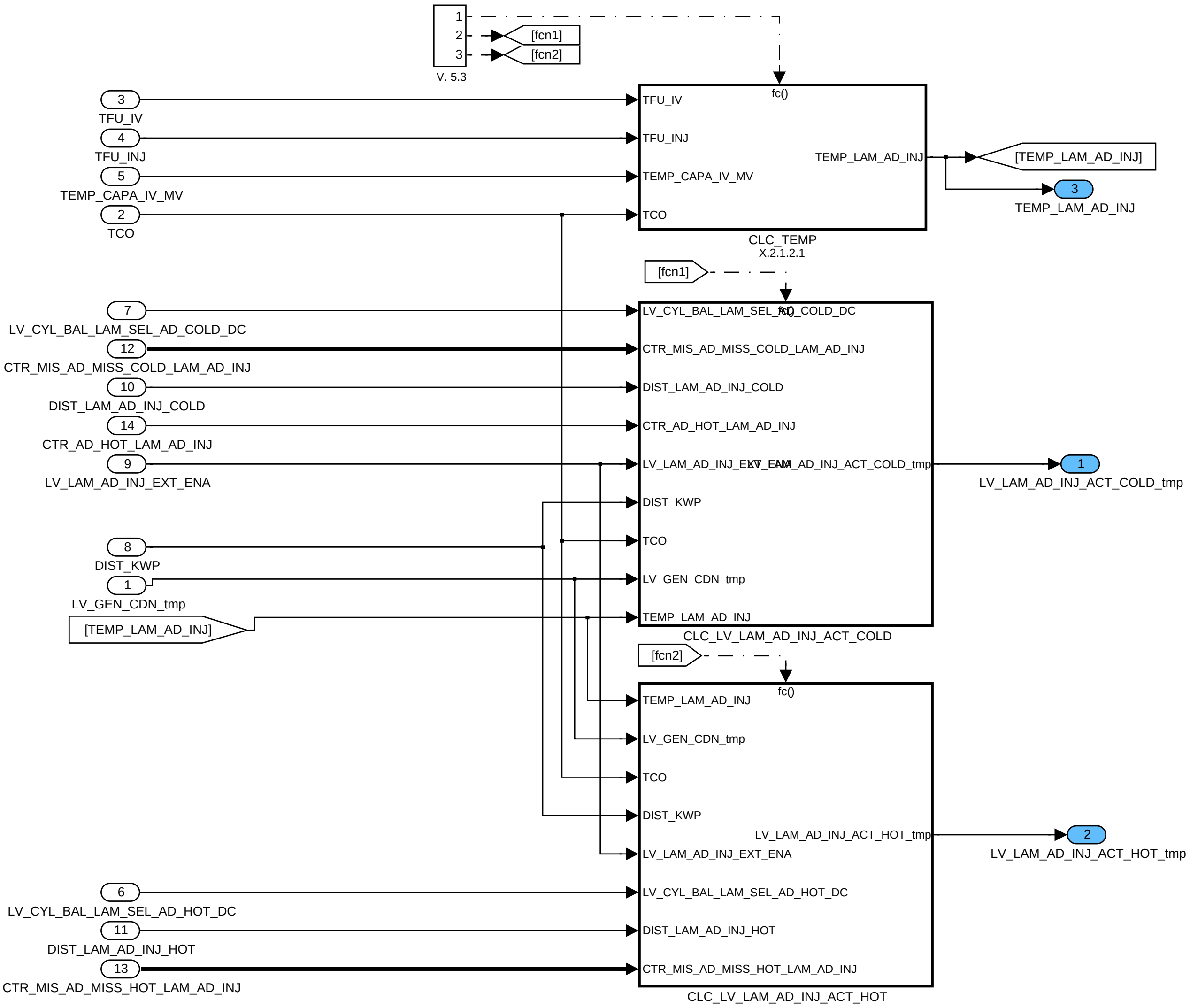


Figure 9:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_INDEP_CLC/TEMP_CDN

49.2.2.1.2.1 Selection of temperature value

Using the calibration C_TEMP_LAM_AD_INJ_ENA the temperature for use within the lambda adaptation via injection mode can be chosen.

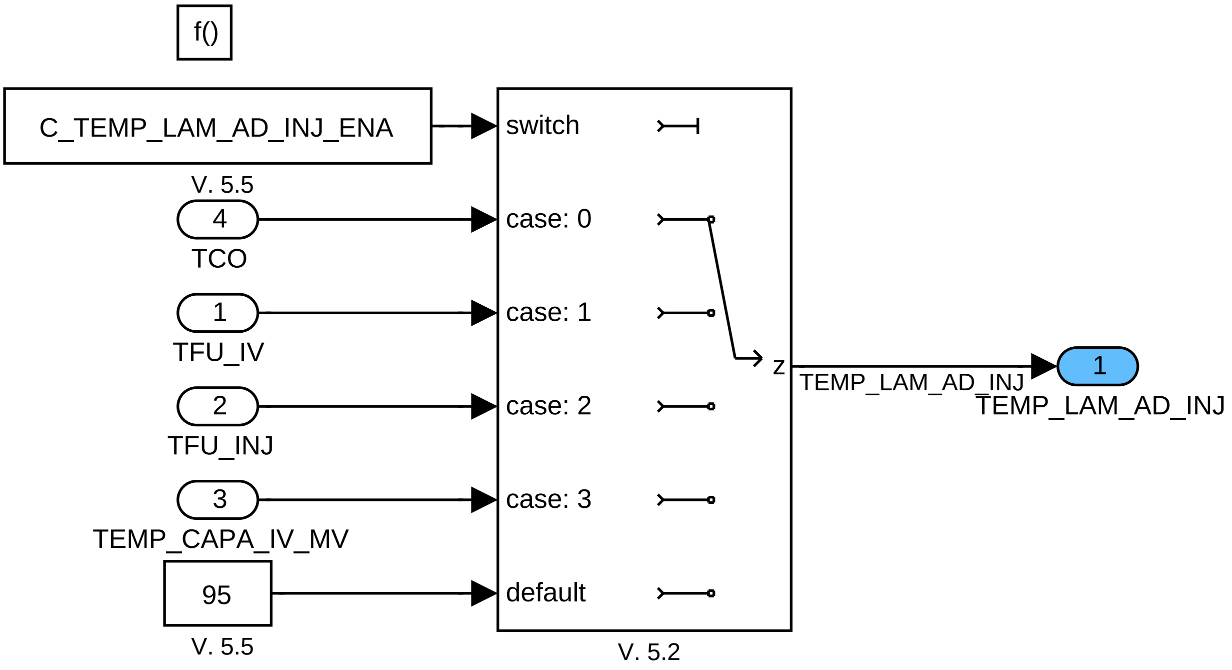


Figure 10:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_INDEP_CLC/TEMP_CDN/CLC_TEMP

49.2.2.1.2.2 Condition for the lower adaptation range

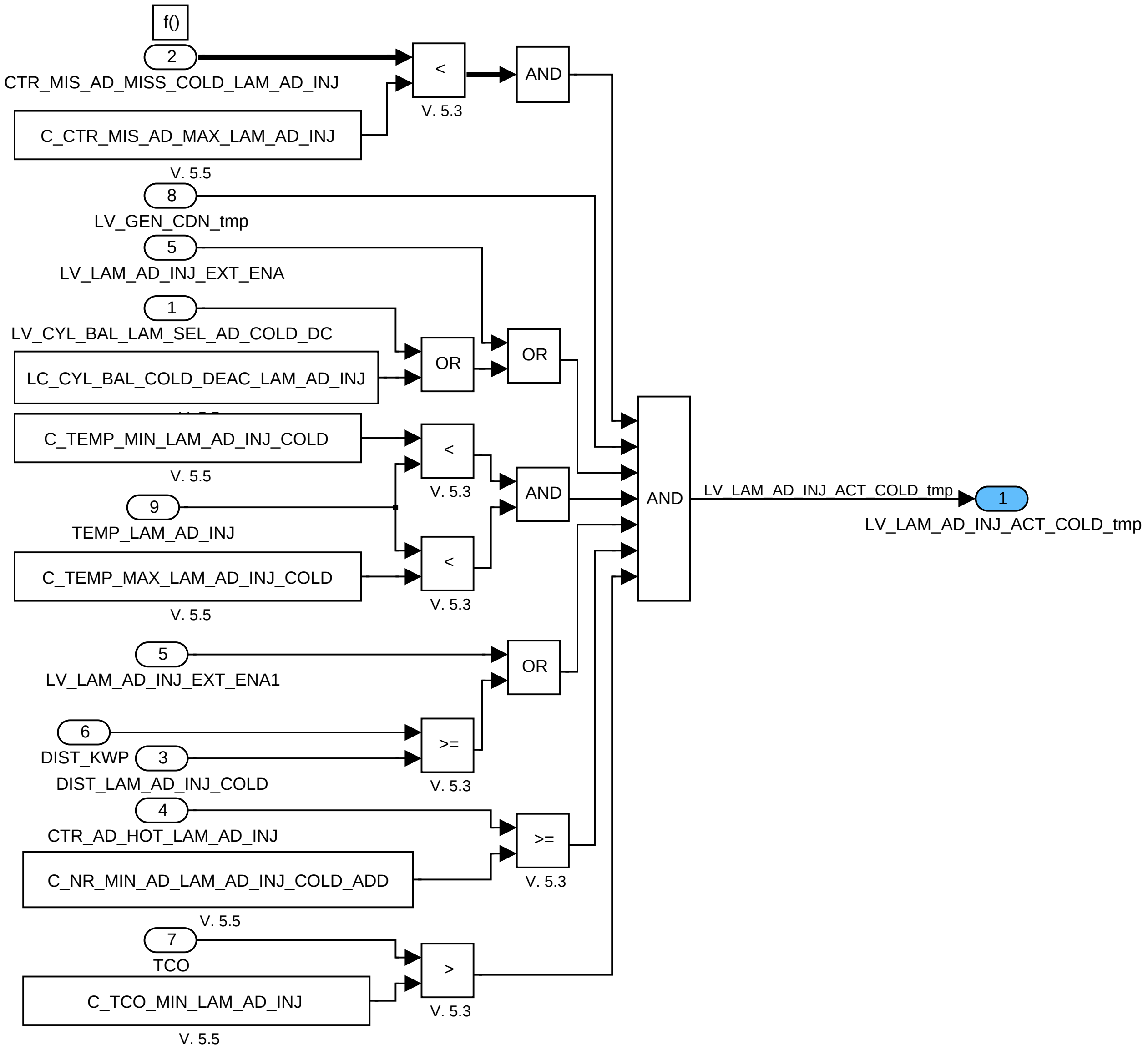


Figure 11:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_INDEP_CLC/TEMP_CDN/CLC_LV_LAM_AD_INJ_ACT_COLD

49.2.2.1.2.3 Condition for the upper adaptation range

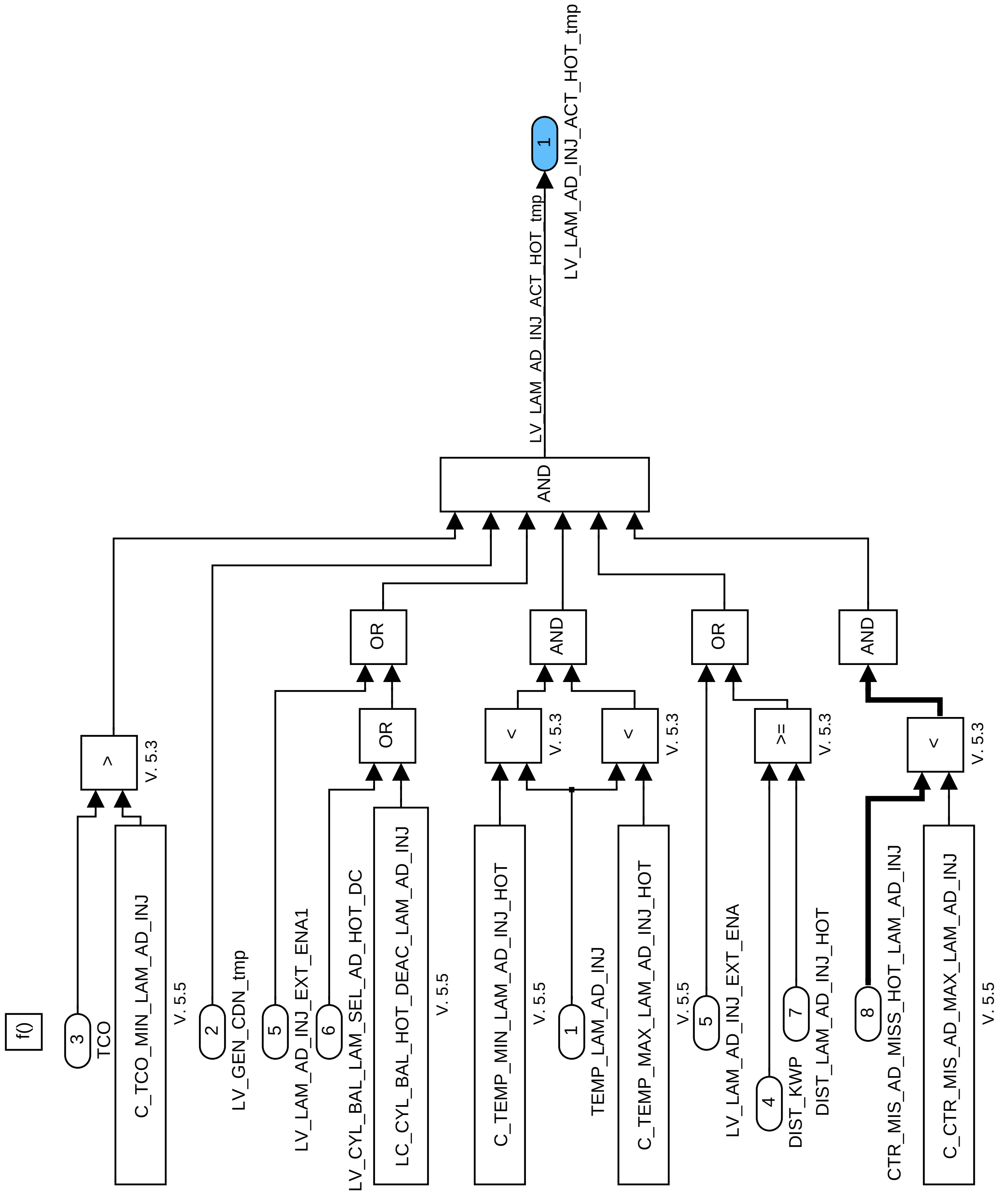


Figure 12:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_INDEP_CLC/TEMP_CDN/CLC_LV_LAM_AD_INJ_ACT_HOT

49.2.2.1.3 Clear the values for interruption and repeat

If all banks are not in multiple and not in single injections, the flags to interrupt and repeat the adaptation are reset. In single injection for all banks the flags and counter for misfire are reset.

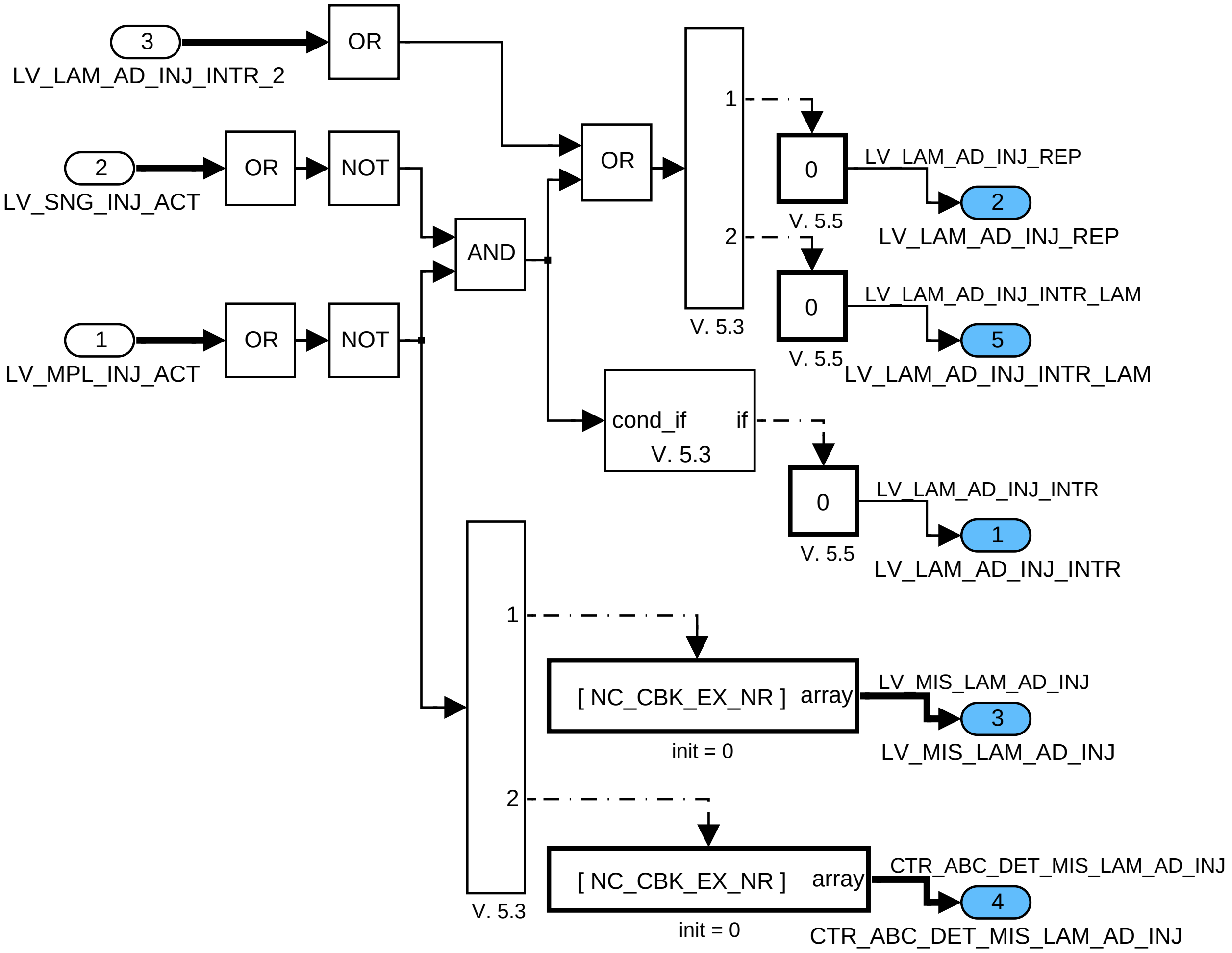


Figure 13:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_INDEP_CLC/CLR_LV_INTR_REP

49.2.2.1.4 Reset of LV_LAM_AD_INJ_EXT_ENA

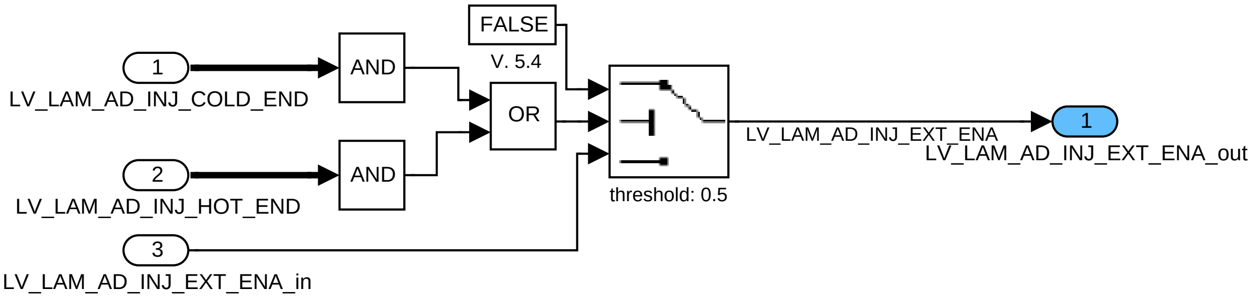


Figure 14:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_INDEP_CLC/CLR_LV_EXT_ENA

49.2.2.2 Bank specific calculations

The following calculations are done for all cylinder banks. Inside the model the bank specific values are denoted with the suffix _i whereas in the textual description they are referenced via [i].

49.2.2.2.1 Limited dynamics determination

The flag LV_LDC_LAM_AD_INJ[i] is calculated using N, MAF; FAQ_TQ_REQ and FAC_LAM_MV conditions. During multiple injections the monitoring of FAC_LAM_MV can be changed by using different thresholds.

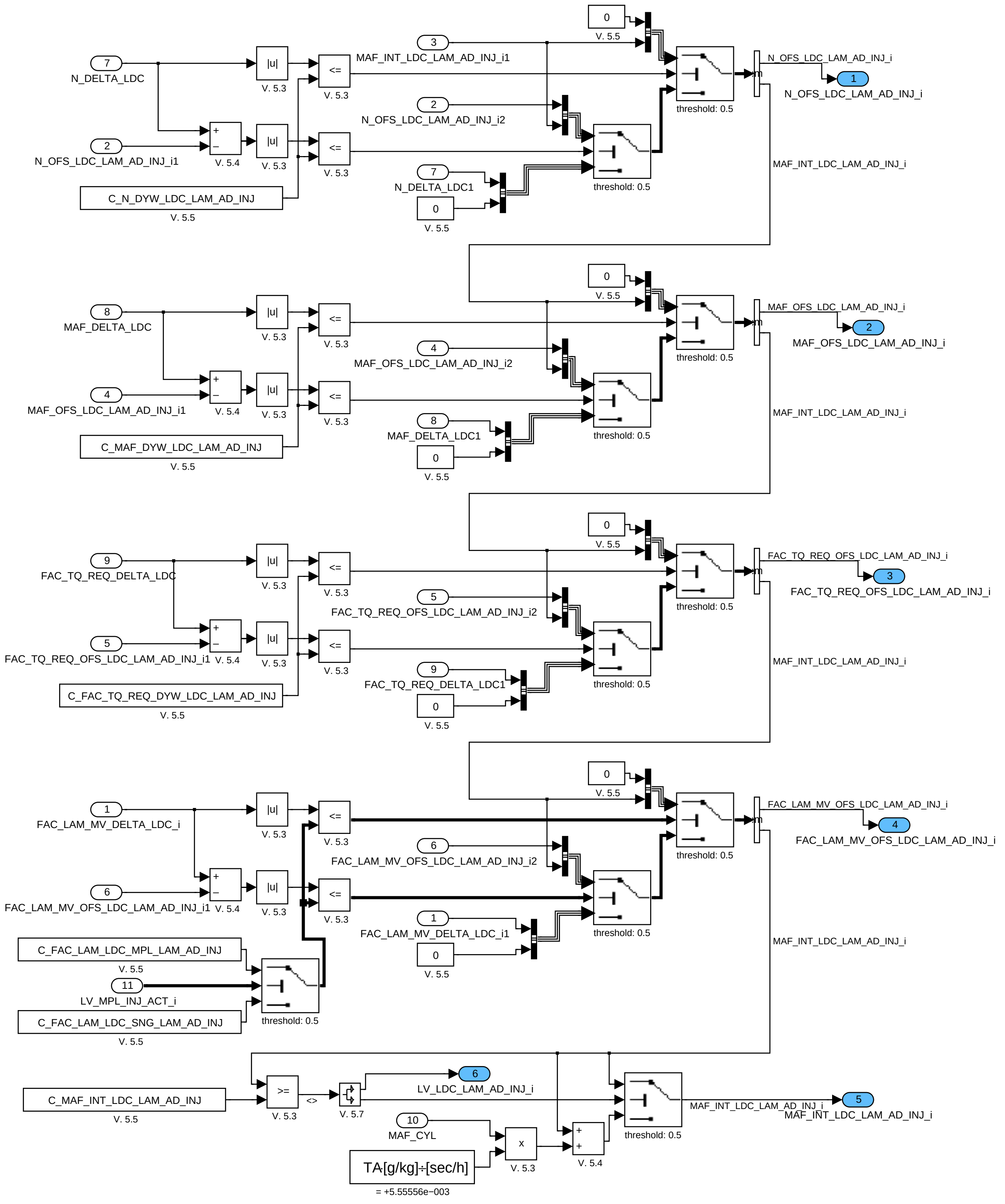


Figure 15:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/LDC_CLC

49.2.2.2.2 Determination of adaptation conditions for upper or lower temperature range

If the limited dynamics are given and the lambda controller operates within defined limits, depending on the temperature conditions the adaptation for the upper or lower temperature range is enabled, if all banks fulfil these conditions and the catalyst temperature has to be below a threshold. Or the adaptation can be activated manually via calibration value.

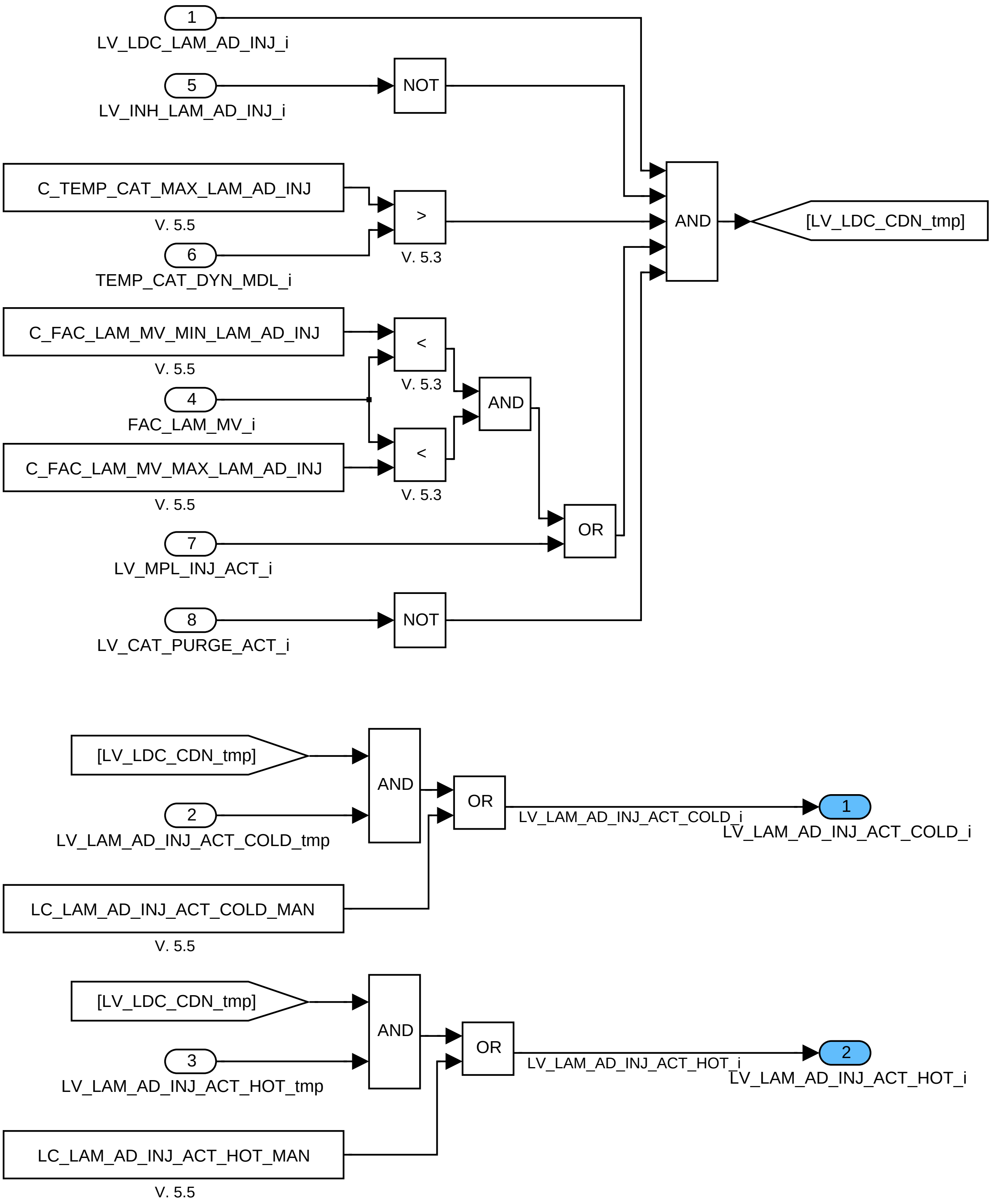


Figure 16:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_HOT_COLD_CDN

49.2.2.2.3 Monitoring of ENSC, LACO and MISF

During the multiple injection phase the output of the engine speed controller, the lambda controller and the misfire detection are monitored.At a certain number of misfire events or at a certain deviation in the lambda controller output the adaptation is interrupted immediately and the adaptation is repeated.For a deviation in the engine speed controller, the adaptation is at first finished and then repeated.

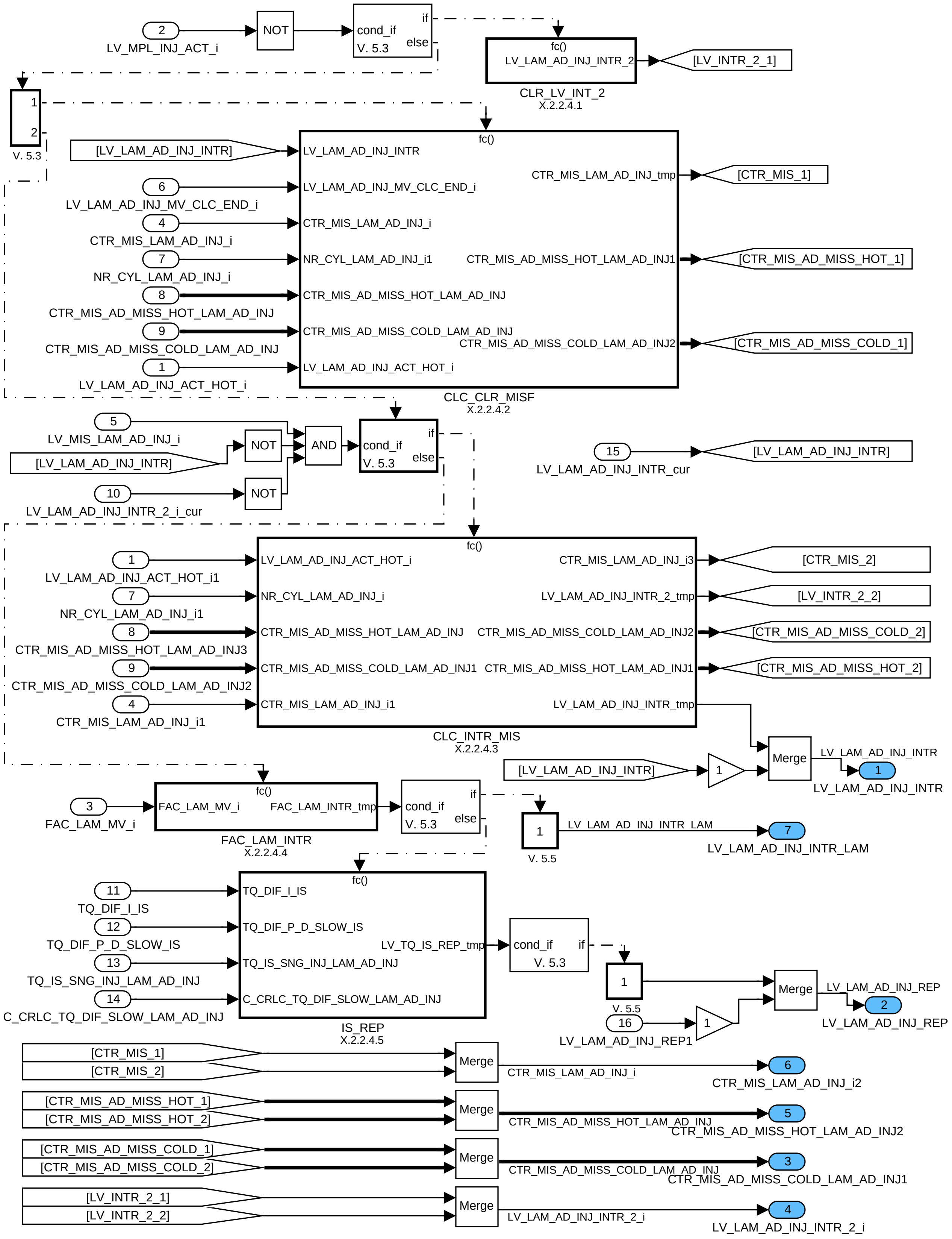


Figure 17:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP

49.2.2.2.3.1 Clear interruption flag

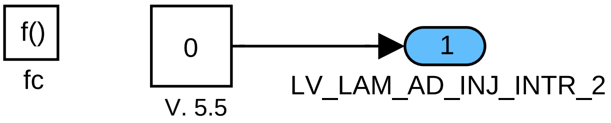


Figure 18:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP/CLR_LV_INT_2

49.2.2.2.3.2 Clear the counter for misfire handling

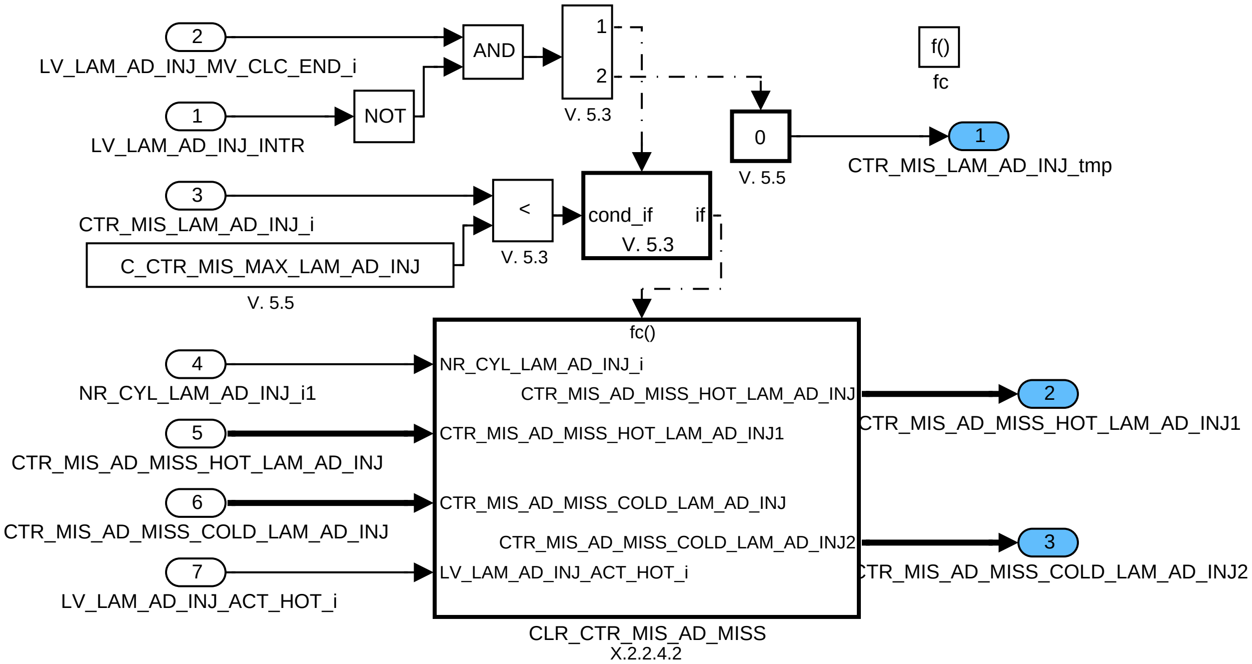


Figure 19:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP/CLC_CLR_MISF

49.2.2.2.3.2.1 Clear temperature depending counters for misfire handling

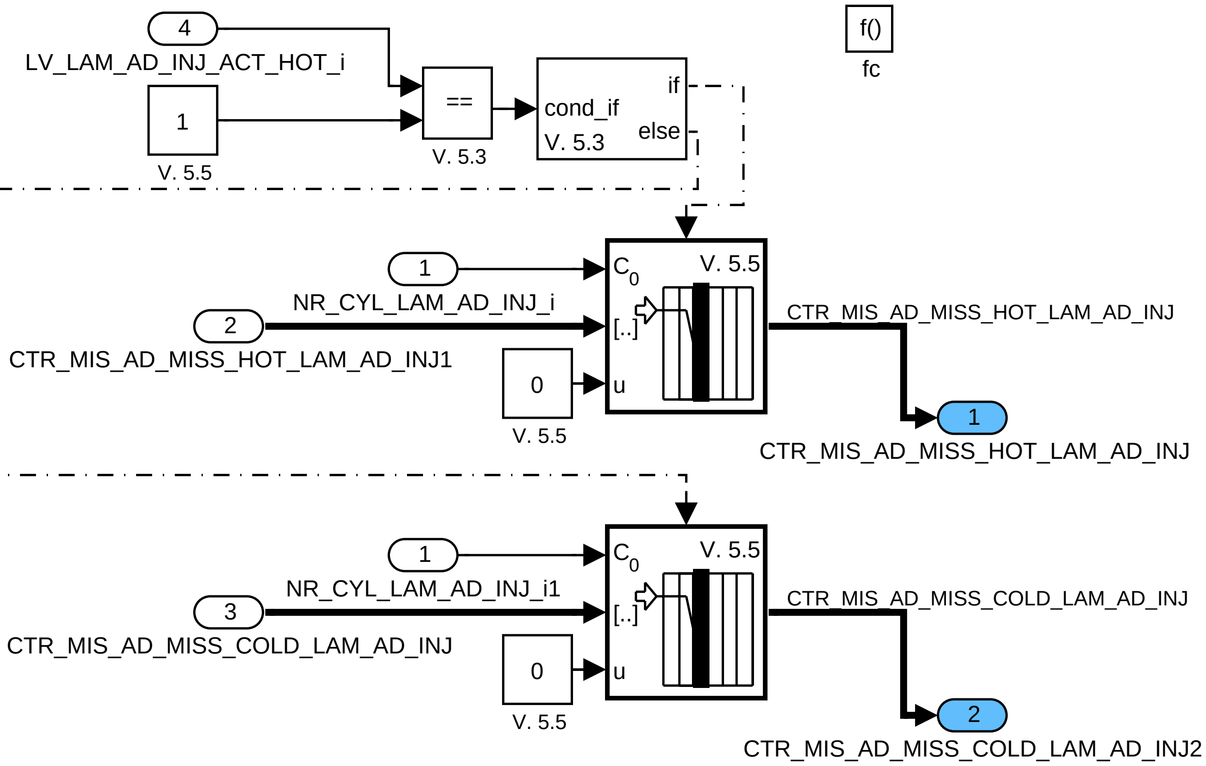


Figure 20:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP/CLC_CLR_MISF/CLR_CTR_MIS_AD_MISS

49.2.2.2.3.3 Calculate counter for misfire handling

This calculation are done, if a misfire during multiple injection is detected. At misfire the misfire counter is incremented an the interruption flag is set, which leads to a repeat of the adaptation. If the maximum number of repeated adaptation due to misfire are reached, a second interruption flag is set, a counter is incremented and the adaptation is continued with all left cylinders.

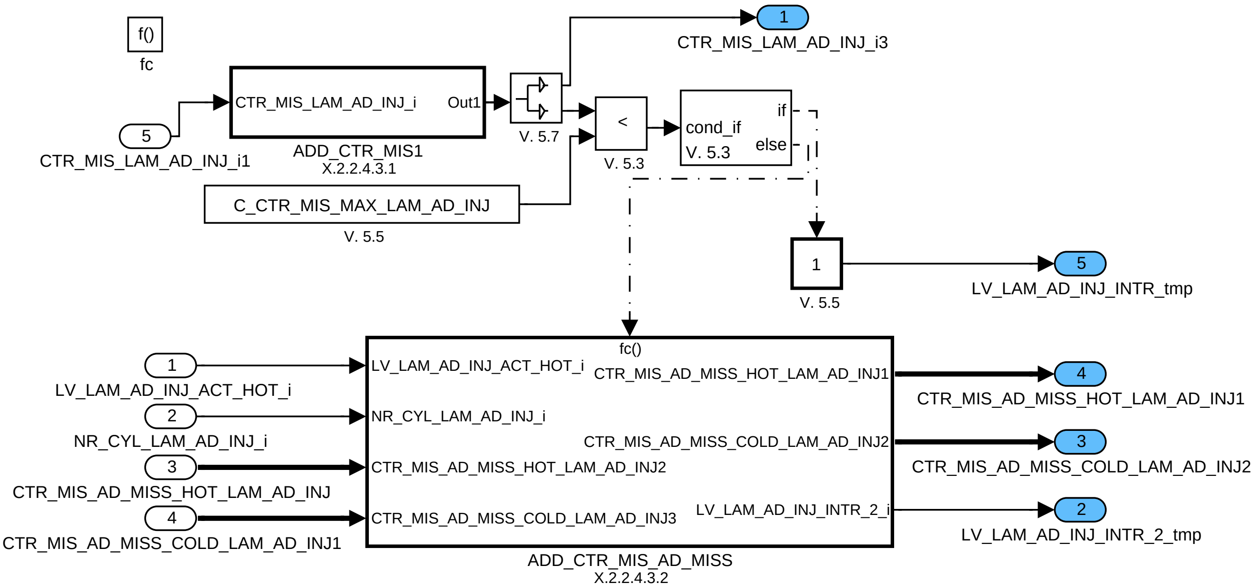


Figure 21:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP/CLC_INTR_MIS

49.2.2.2.3.3.1 Incrementation of CTR_MIS_LAM_AD_INJ

The counter is incremented and limited.

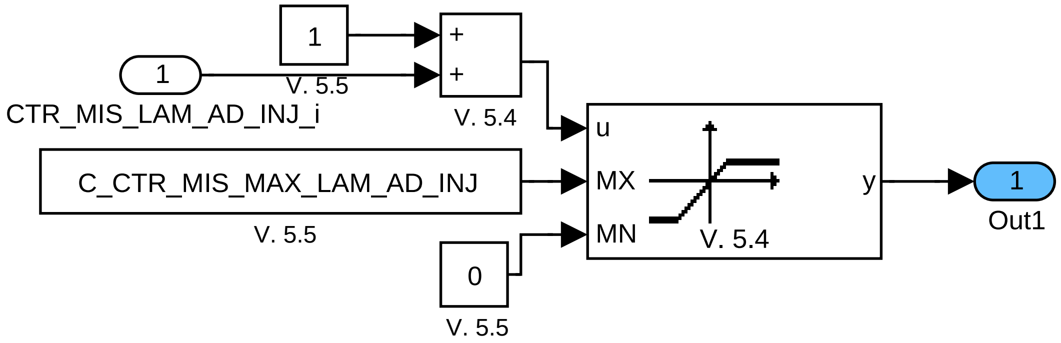


Figure 22:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP/CLC_INTR_MIS/ADD_CTR_MIS1

49.2.2.2.3.3.2 Incrementation of of CTR_MIS_AD_MISS_COLD/HOT_LAM_AD_INJ

Depending on the temperature range the counters are incremented.

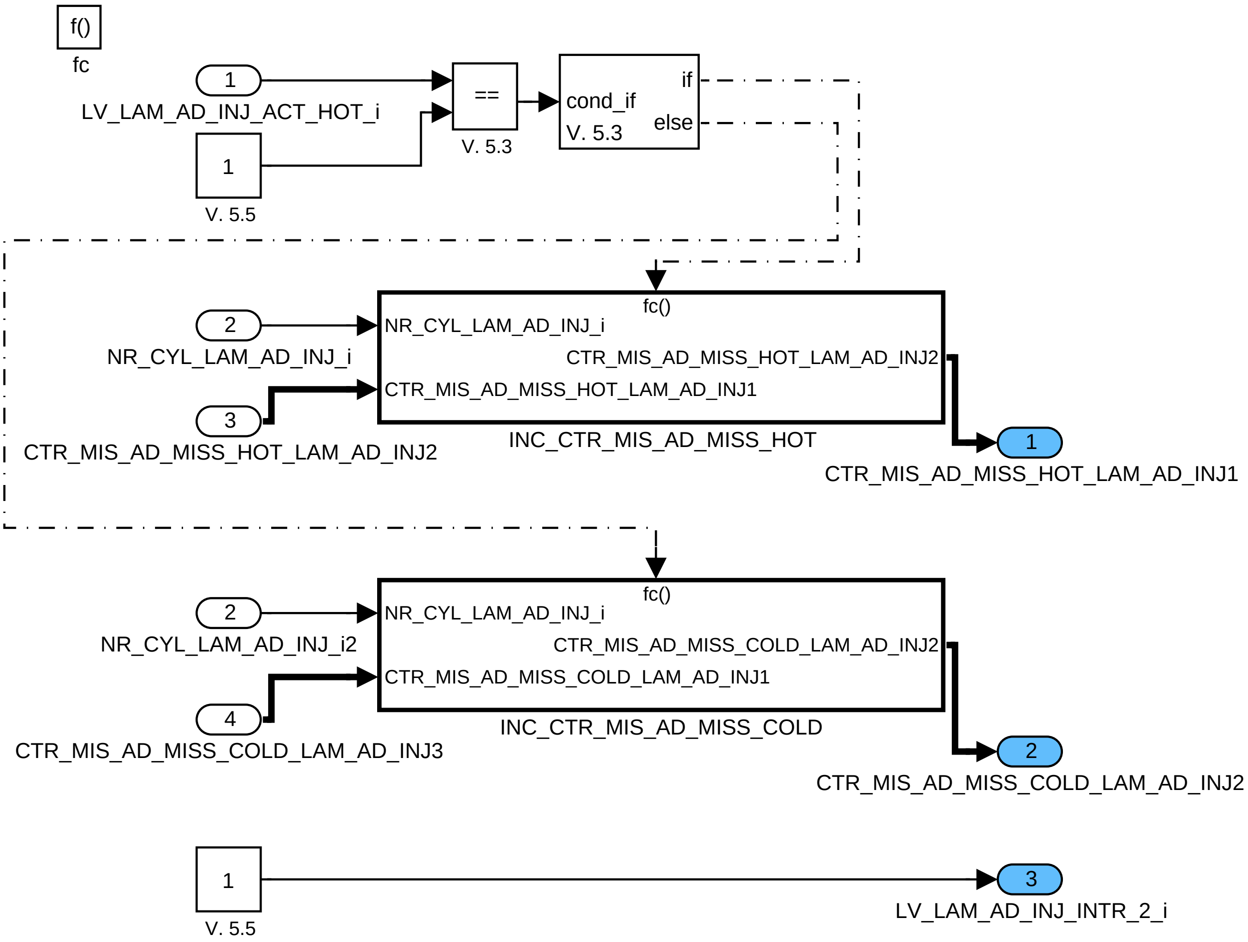


Figure 23:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP/CLC_INTR_MIS/ADD_CTR_MIS_AD_MISS

49.2.2.2.3.4 Conditions for interruption due to lambda conditions

If the lambda controller output exceeds the limitations, the adaptation is interrupted.

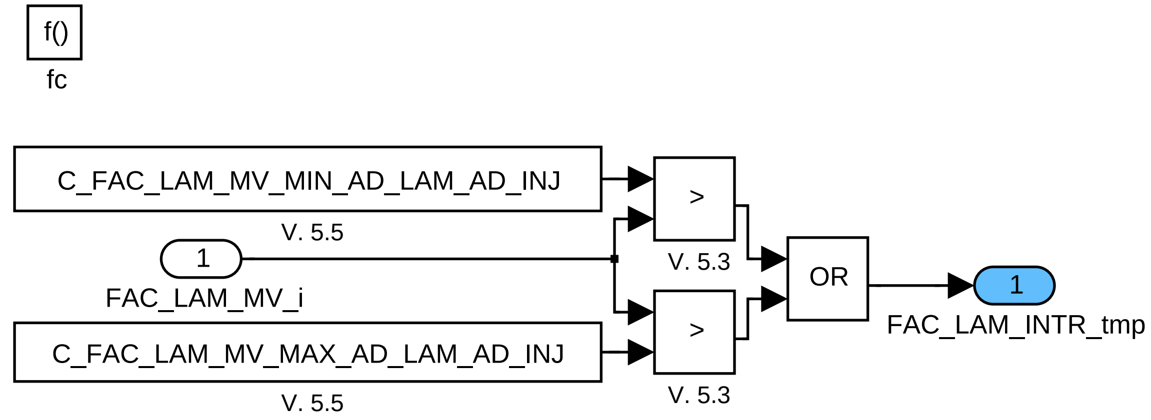


Figure 24:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP/FAC_LAM_INTR

49.2.2.2.3.5 Conditions for interruptions due to ENSC controller output deviation

For a big difference in the idle speed controller in single and multiple injection, the adaptation will be finished and repeted.

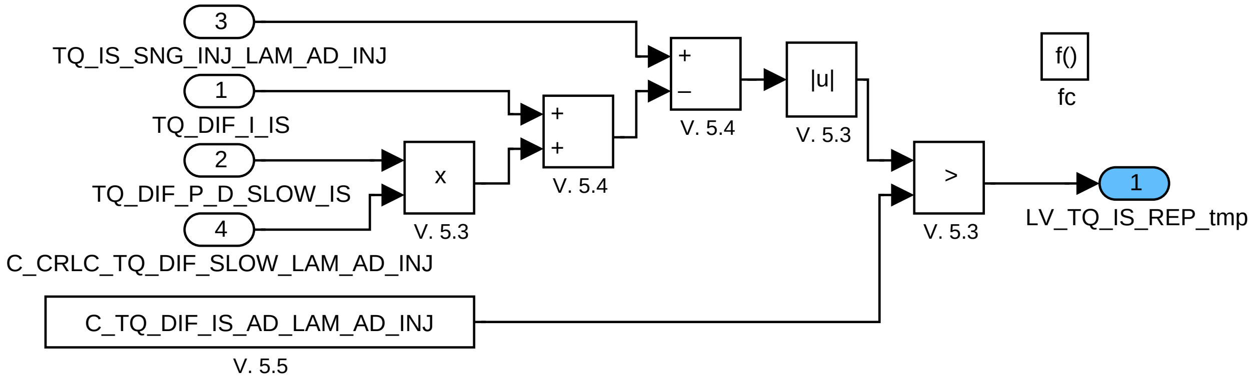


Figure 25:
Path: MFMA_ISPCLAIMA0/OPM_20MS/BANK_SPECIFIC_CLC/CLC_INTR_REP/IS_REP

49.2.2.3 Calculation of general activation flag and update of the distance counter

The general activation flag is set if the bank specific condition for hot or cold conditions are given. Also the update of the mileage threshold is updated.

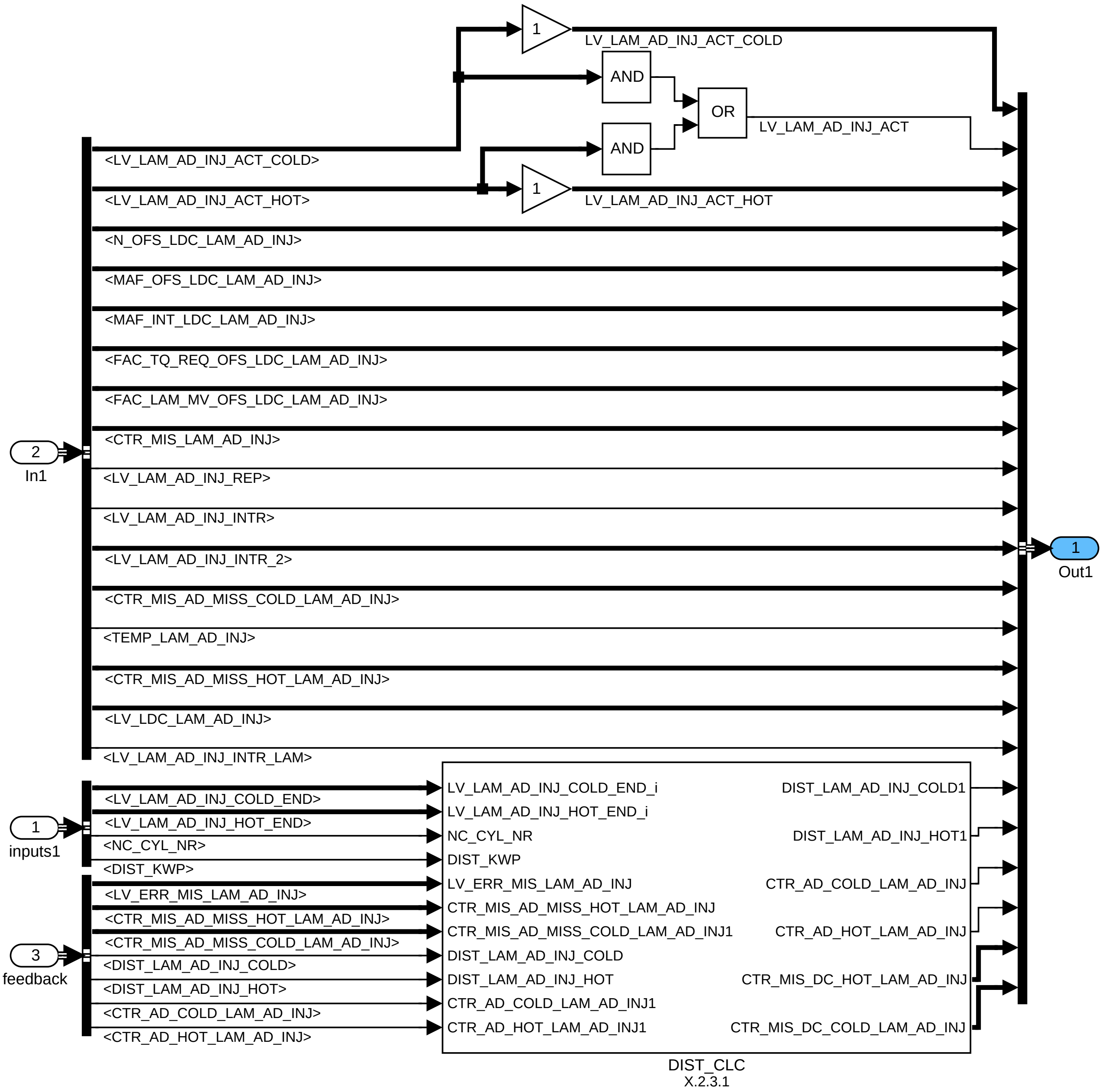


Figure 26:
Path: MFMA_ISPCLAIMA0/OPM_20MS/CLC_ACT

49.2.2.3.1 Update of the mileage threshold

The mileage threshold is separately updated for hot and cold adaptation if the adaptation is finished successfully for all cylinders and no repeat of adaptation due to misfire is necessary.

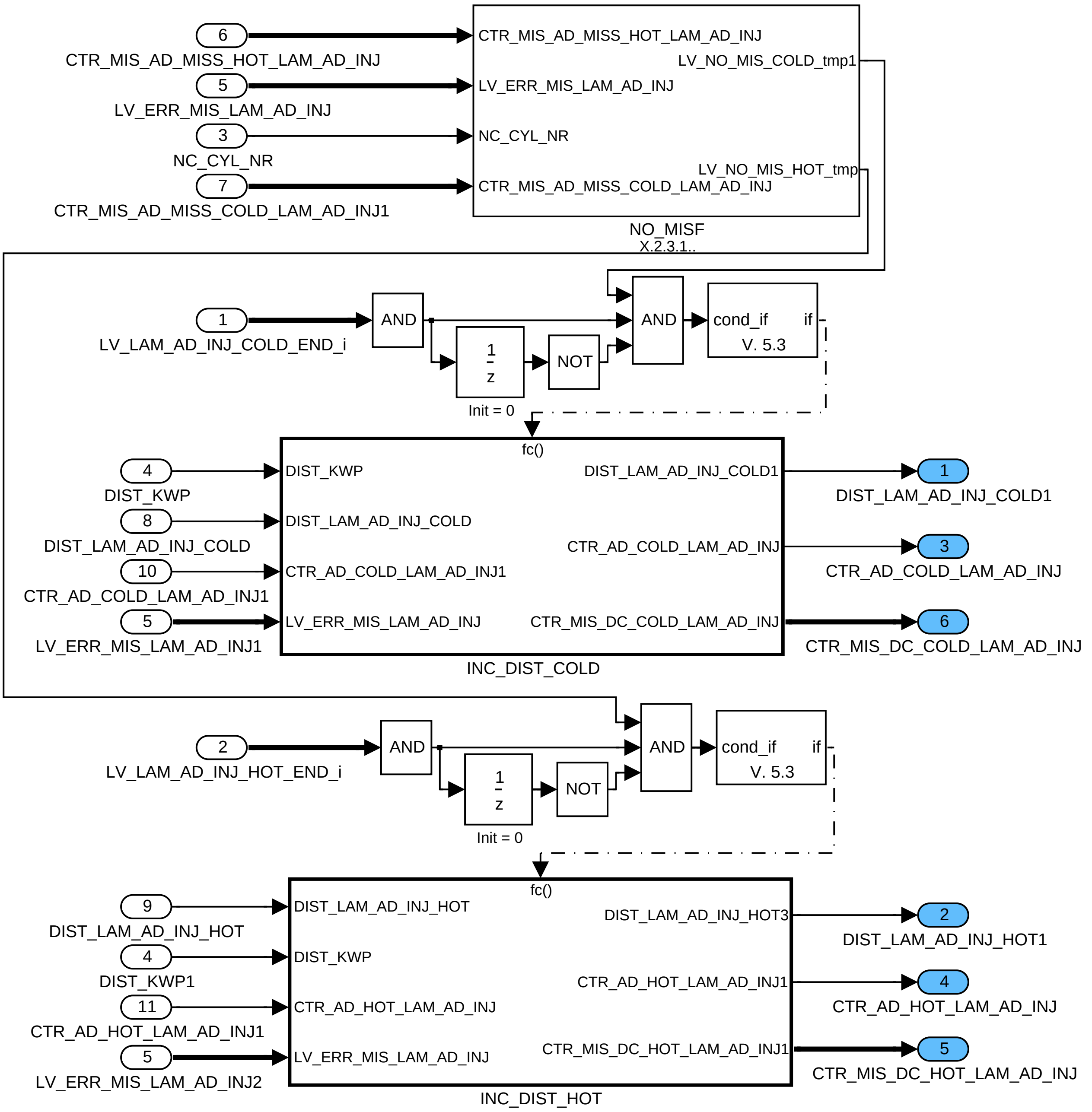


Figure 27:
Path: MFMA_ISPCLAIMA0/OPM_20MS/CLC_ACT/DIST_CLC

49.2.2.3.1.1 Distance threshold incrementation for cold adaptation

At successful adaptation the distance counter will be incremented; the CTR_AD_COLD_LAM_AD_INJ will be incremented and if no error is present and the LC_CTR_MIS_DC_LAM_AD_INJ_CLR is set the CTR_MIS_DC_COLD_LAM_AD_INJ is cleared.

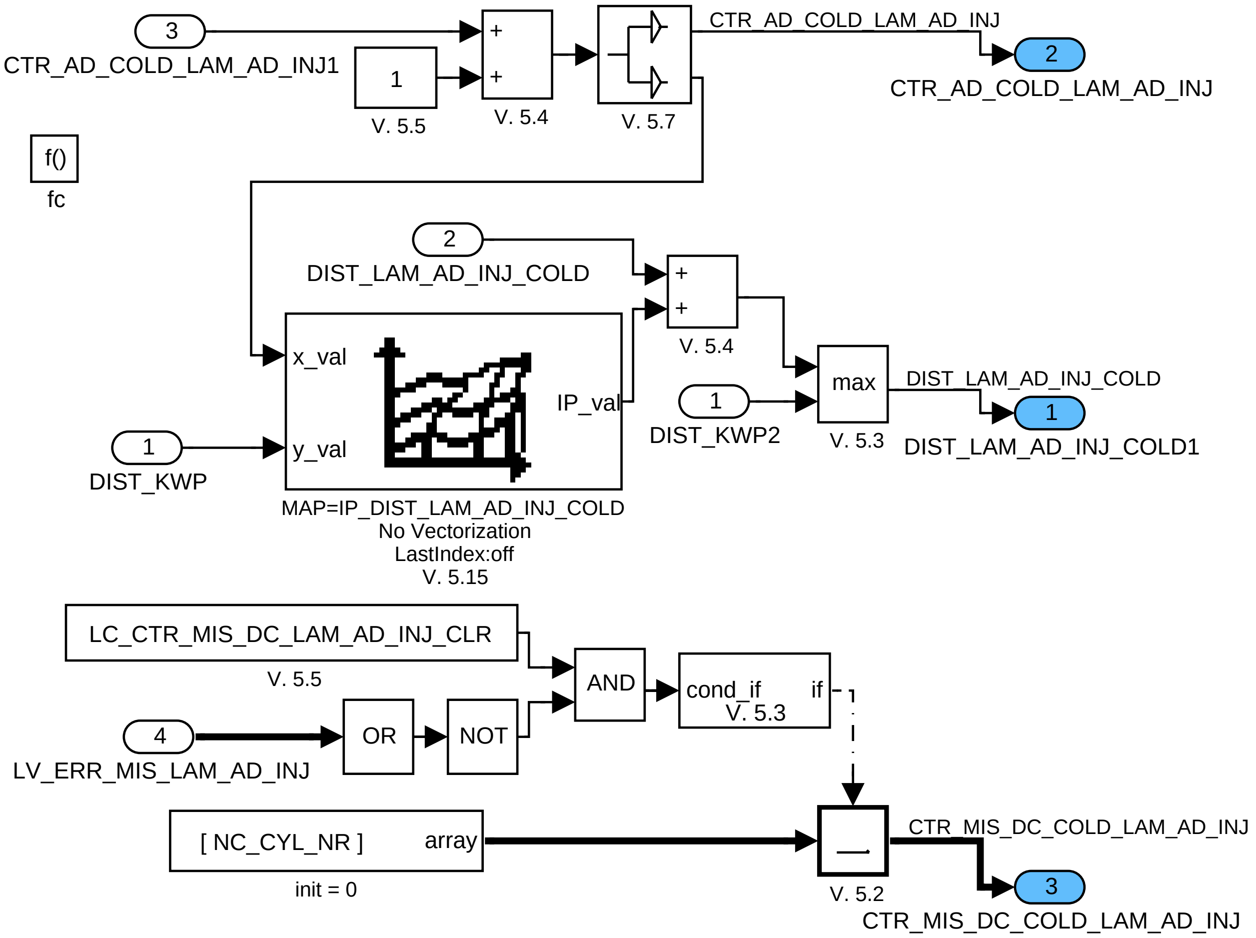


Figure 28:
Path: MFMA_ISPCLAIMA0/OPM_20MS/CLC_ACT/DIST_CLC/INC_DIST_COLD

49.2.2.3.1.2 Distance threshold incrementation for hot adaptation

At successful adaptation the distance counter will be incremented; the CTR_AD_HOT_LAM_AD_INJ will be incremented and if no error is present and the LC_CTR_MIS_DC_LAM_AD_INJ_CLR is set the CTR_MIS_DC_HOT_LAM_AD_INJ is cleared.

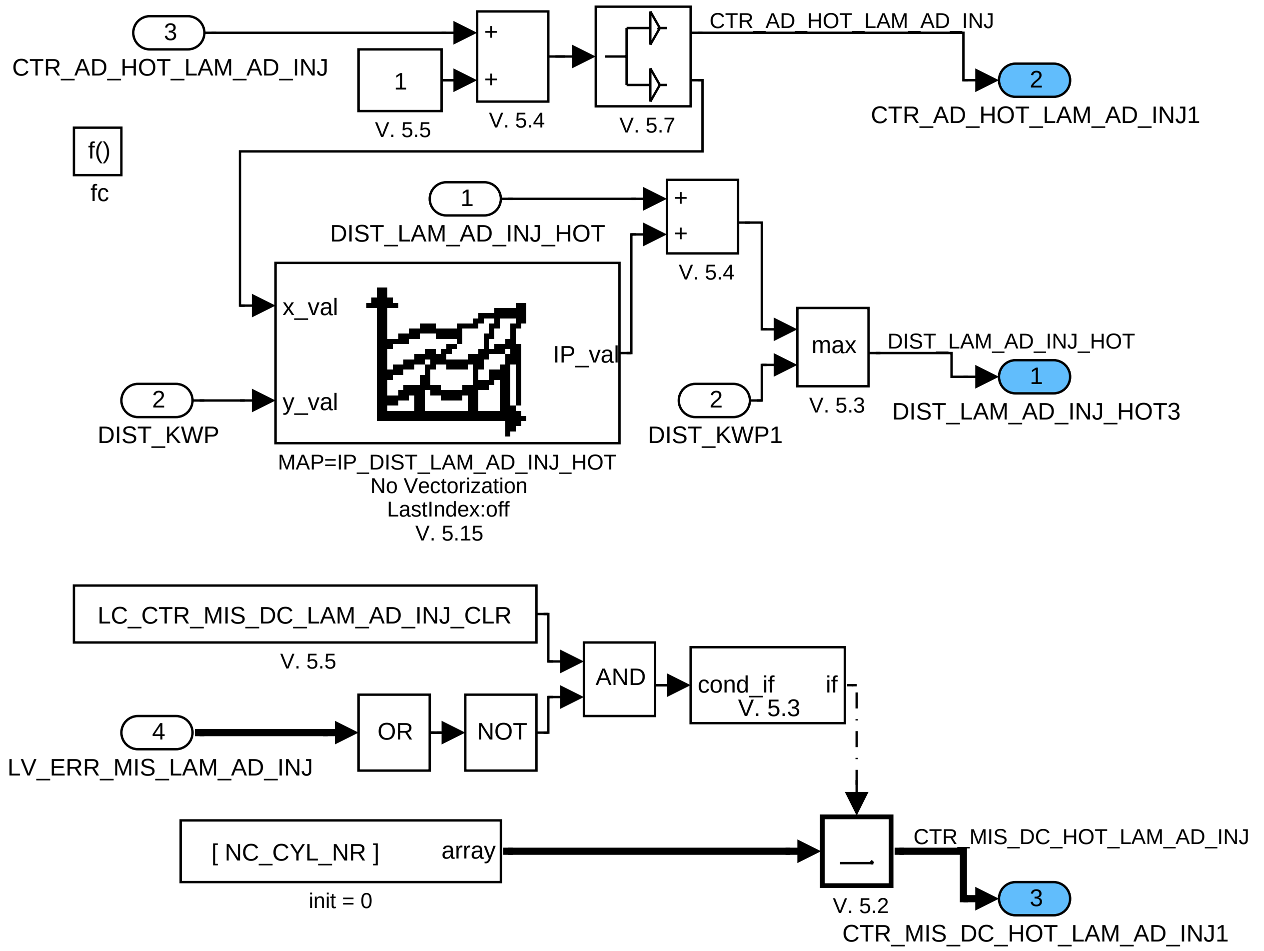


Figure 29:
Path: MFMA_ISPCLAIMA0/OPM_20MS/CLC_ACT/DIST_CLC/INC_DIST_HOT

49.2.2.3.1.3 No misfire condition for mileage threshold update

The mileage threshold can be updated, if the condition "no misfire" is present for all cylinders.

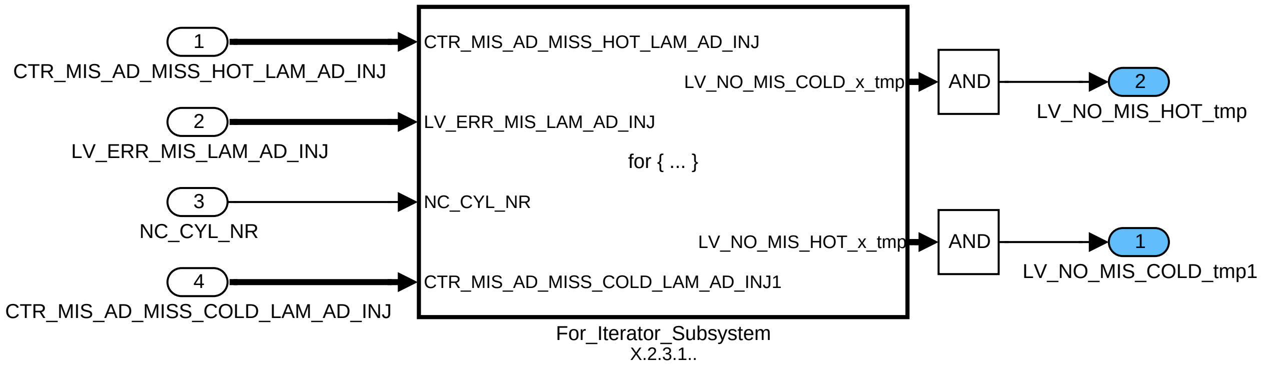


Figure 30:
Path: MFMA_ISPCLAIMA0/OPM_20MS/CLC_ACT/DIST_CLC/NO_MISF

49.2.2.3.1.3.1 No misfire condition

For each cylinder it tested, if no misfire was present or the error for misfire is already present.

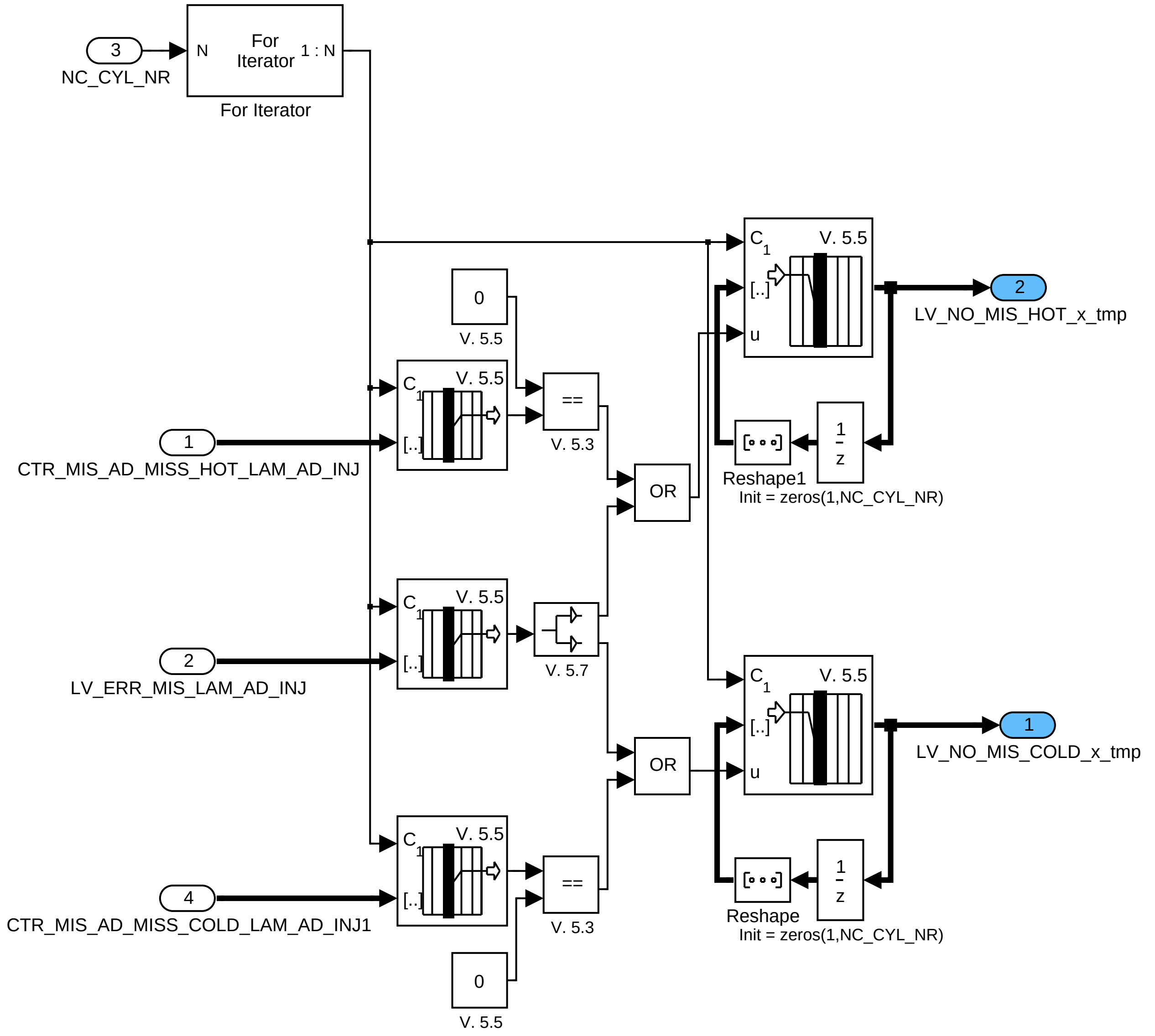


Figure 31:
Path: MFMA_ISPCLAIMA0/OPM_20MS/CLC_ACT/DIST_CLC/NO_MISF/For_Iterator_Subsystem

49.2.3 Segment synchronous calculations to detect misfire

Calculations to detect misfire with the same recurrence as the misfire calculations. Misfire will be detected only at multiple injection phase, represented by STATE_LAM_AD_INJ_ACT > 2 (single injection)

49.2.3.1 Detection of misfire

If misfire is detected for the cylinder with multiple injections, a misfire counter is incremented. The flag indicating misfire is set, if the counter reach a threshold.

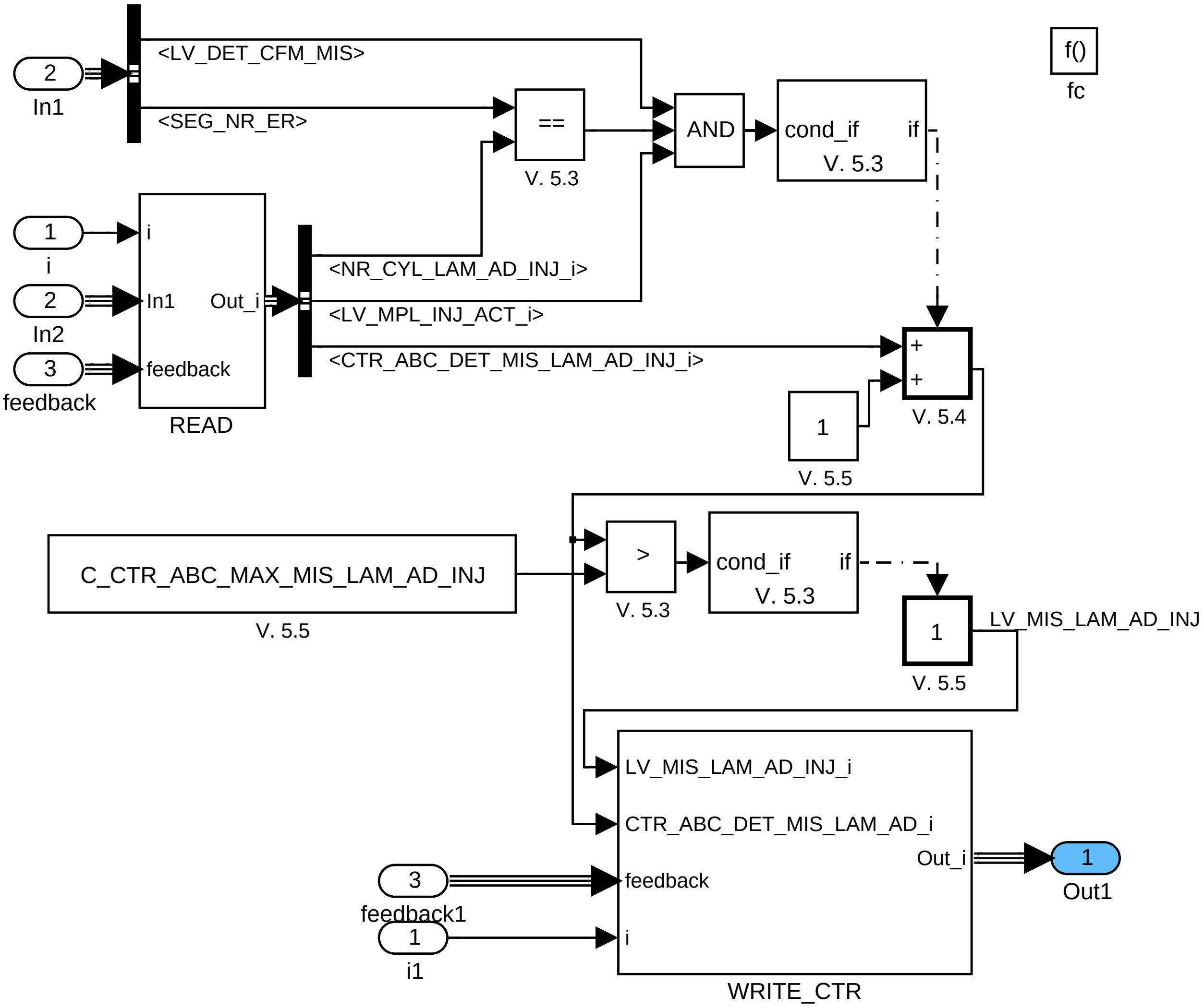


Figure 32:
Path: MFMA_ISPCLAIMA0/OPM_SEG/CLC_MIS

49.2.4 Calculations at event engine stop to engine run

Cylinder individual calculations to clear the error flags.

49.2.4.1 Clear error flags, distance threshold and counters

At a change of an injector, the CTR_AD_HOT/COLD_LAM_AD_INJ and the error flags and counter are cleared. The distance threshold and the error can also be cleared by calibration data. If a resetting of the NVMY data via service tester is requested with the flag LV_LAM_AD_INJ_CLR_AD_EXT, the counters and the distance thresholds are cleared.

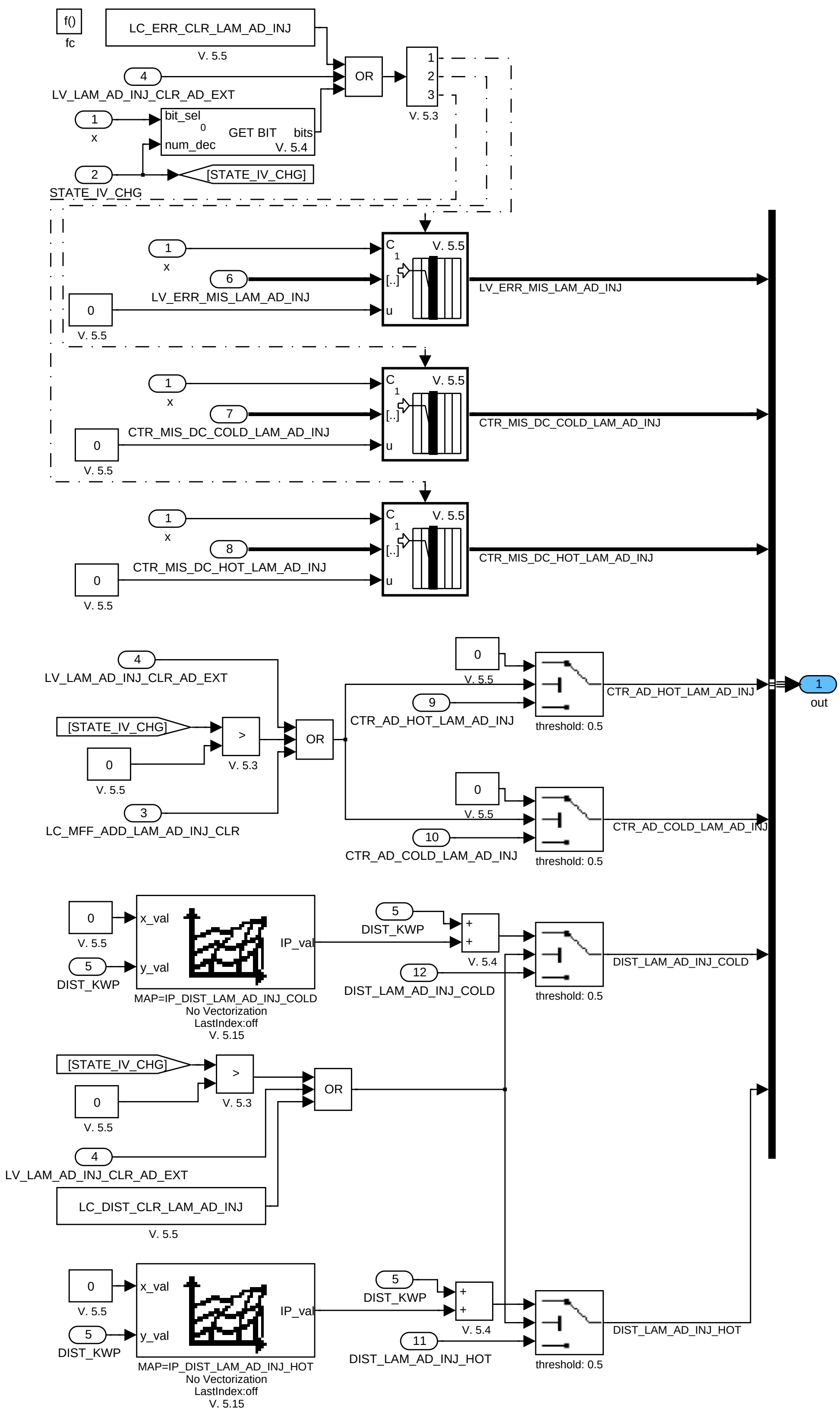


Figure 33:
Path: MFMA_ISPCLAIMA0/OPM_ES2ERU/CLR_CTR_AND_ERR

49.2.5 Detailed description for Action: ACTION_MFMA_GetEnableCondition

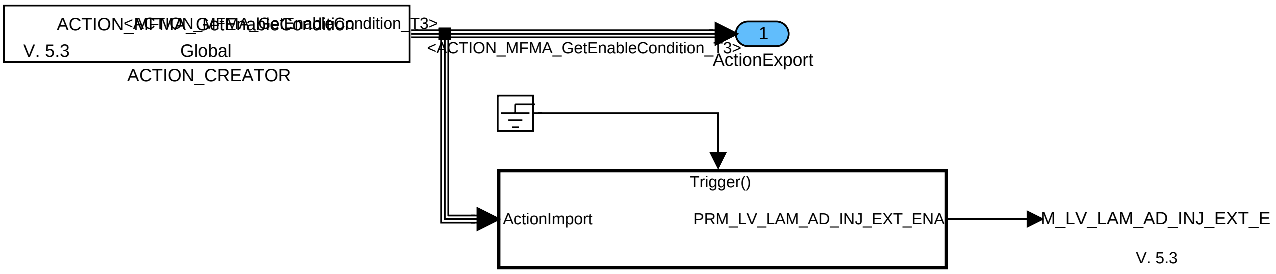


Figure 34:
Path: MFMA_ISPCLAIMA0/ACTIONDEF_MFMA_GetEnableCondition

49.2.5.1 No title given

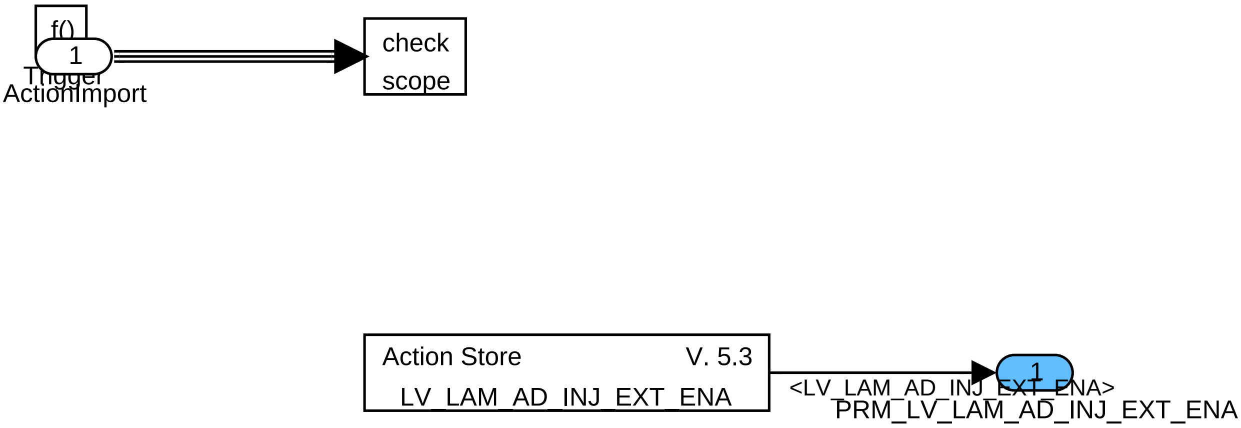


Figure 35:
Path: MFMA_ISPCLAIMA0/ACTIONDEF_MFMA_GetEnableCondition/ACTION_MFMA_GetEnableCondition

49.2.6 Detailed description for Action: ACTION_MFMA_SetEnableCondition

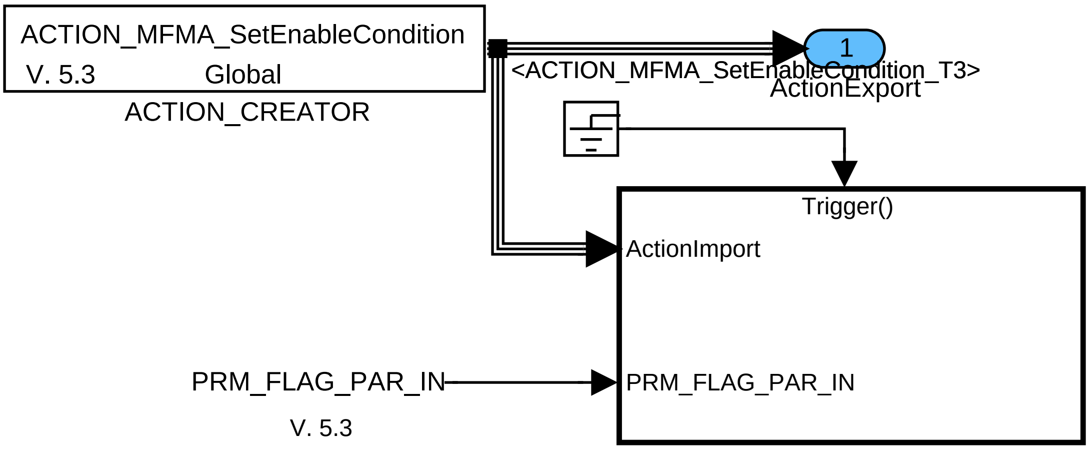


Figure 36:
Path: MFMA_ISPCLAIMA0/ACTIONDEF_MFMA_SetEnableCondition

49.2.6.1 No title given

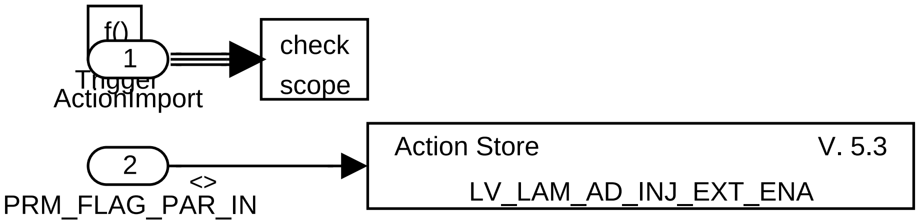


Figure 37:
Path: MFMA_ISPCLAIMA0/ACTIONDEF_MFMA_SetEnableCondition/ACTION_MFMA_SetEnableCondition

49.3 Coordination of Lambda adaptation via injection mode

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_LAM_AD_INJ_COLD_END [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag indicating that the lambda adaptation via injection mode for the lower temp range is finished					
LV_LAM_AD_INJ_HOT_END [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag indicating that the lambda adaptation via injection mode for the higher temp range is finished					
LV_LAM_AD_INJ_PHA_0 [NC_CBK_EX_NR]	V	0... 1H	0... 1	1	[-]
Flag to memorize that first adaptation phase is active					
LV_MPL_INJ_ACT [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag indicating that multiple injection for lambda adaptation via injection mode is active					
LV_SNG_INJ_ACT [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Flag indicating that single injection for lambda adaptation via injection mode is active					
NR_CYL_LAM_AD_INJ [NC_CBK_EX_NR]	O/V	80... 7FH	-128 ...127	1	[-]
Defines the cylinder that is currently using multiple injections for lambda adaptation					
NR_PHA_LAM_AD_INJ_COLD [NC_CBK_EX_NR]	V/S	0... FH	0... 15	1	[-]
Currently desired lambda adaptation via injection mode phase for the lower temp range					
NR_PHA_LAM_AD_INJ_HOT [NC_CBK_EX_NR]	V/S	0... FH	0... 15	1	[-]
Currently desired lambda adaptation via injection mode phase for the upper temp range					
STATE_LAM_AD_INJ_ACT [NC_CBK_EX_NR]	O/V	0	NOT_ACT	-	[-]
		1	SNG_ALL_L		
		2	SNG_ALL_H		
		3	MPL_CYL0_L		
		4	MPL_CYL0_H		
		5	MPL_CYL1_L		
		6	MPL_CYL1_H		
		7	MPL_CYL2_L		
		8	MPL_CYL2_H		
		9	MPL_CYL3_L		
		10	MPL_CYL3_H		
		11	MPL_CYL4_L		
		12	MPL_CYL4_H		
		13	MPL_CYL5_L		
		14	MPL_CYL5_H		
		15	MPL_CYL6_L		
		16	MPL_CYL6_H		
		17	MPL_CYL7_L		
		18	MPL_CYL7_H		
Current injeciton state for each cylinder requested by lambda adaptation via injection mode					
STATE_LAM_AD_INJ_MNG [NC_CBK_EX_NR]	V	0	NOT_ACT	-	[-]
		1	CLC_CYL_STATE		
		2	WAIT_LAM_AD_INJ		
State of the lamdba adaptation via injection mode manager					

Input Data:

LV_LAM_AD_INJ_ACT_COLD [NC_CBK_EX_NR]	LV_LAM_AD_INJ_ACT_HOT [NC_CBK_EX_NR]	LV_LAM_AD_INJ_MV_CLC_END [NC_CBK_EX_NR]	LV_LAM_AD_INJ_INTR
LV_LAM_AD_INJ_INTR_2 [NC_CBK_EX_NR]	LV_LAM_AD_INJ_REP	LV_ERR_MIS_LAM_AD_INJ [NC_CYL_NR]	LV_LAM_AD_INJ_INTR_LAM
LV_LAM_AD_INJ_CUS_ACK [NC_CBK_EX_NR]	NC_CBK_EX_NR	STATE_LS [NC_CBK_EX_NR]	LV_LAM_AD_INJ_CLR_AD_EXT
NC_NR_PHA_LAM_AD_INJ			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_NR_PHA_LAM_AD_INJ	1	0... FH	0... 15	1	[-]
Number of adaptation cycles for lambda adaptation via injection mode					
ID_NR_CYL_LAM_AD_INJ [NC_CBK_EX_NR]	NC_NR_PHA_LAM_AD_INJ	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	NOT_ACT SNG_ALL_L SNG_ALL_H MPL_CYL0_L MPL_CYL0_H MPL_CYL1_L MPL_CYL1_H MPL_CYL2_L MPL_CYL2_H MPL_CYL3_L MPL_CYL3_H MPL_CYL4_L MPL_CYL4_H MPL_CYL5_L MPL_CYL5_H MPL_CYL6_L MPL_CYL6_H MPL_CYL7_L MPL_CYL7_H	-	[-]
LDP_NR_ID_NR_CYL_LAM_AD_INJ	NC_NR_PHA_LAM_AD_INJ	0... FH	0... 15	1	[-]
Desired injection order for all banks					

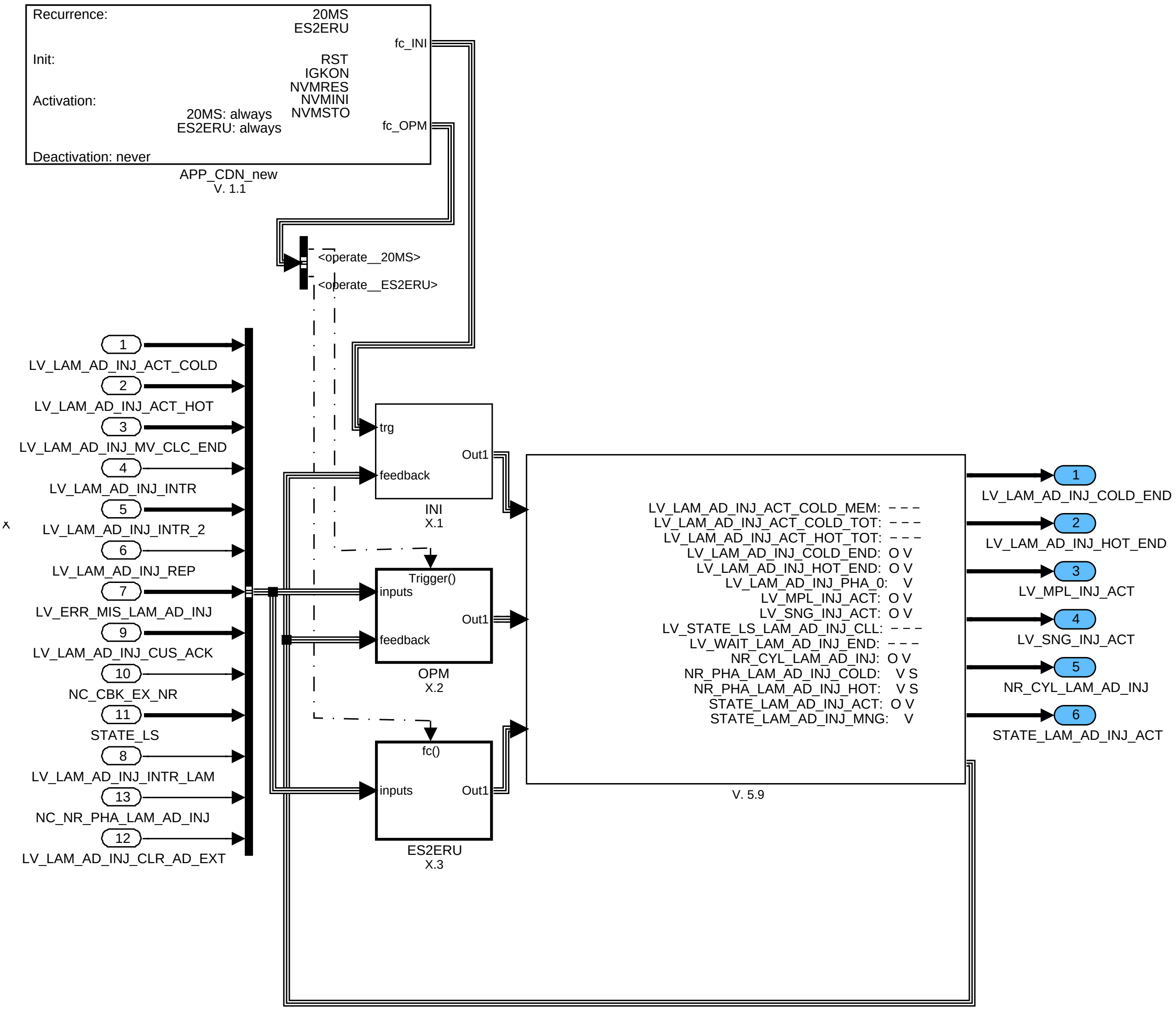
General Information

This module coordinates the usage of single and multiple injections for lambda adaptation via injection mode.

Application Conditions

Initialization: RST, IGKON, NVMRES, NVMINI, NVMSTO
Activation: 20MS: always
ES2ERU: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.0.4 29−Aug−2006

Figure 38:
Path: MFMA_REQCOAIMM0

49.3.1 Initialization and NVMY management

49.3.1.1 Reading from non-volatile memory

The phase number NR_PHA_LAM_AD_INJ_COLD/HOT are initialized from the non-volatile memory.

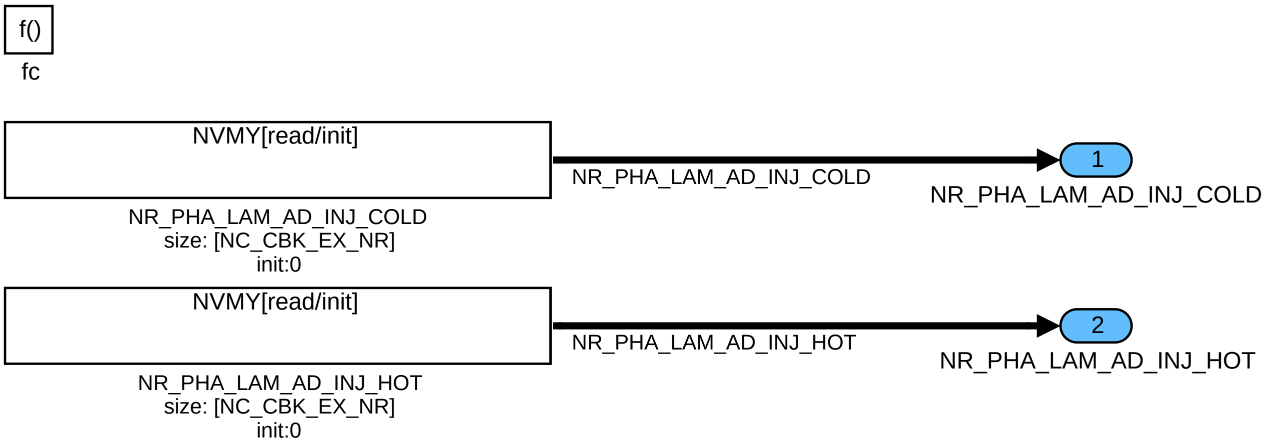


Figure 39:
Path: MFMA_REQCOAIMM0/INI/NVMY_RD

49.3.1.2 Writing to the non-volatile memory

After powerlatch the values of NR_PHA_LAM_AD_INJ_COLD/HOT are written to the non-volatile memory.

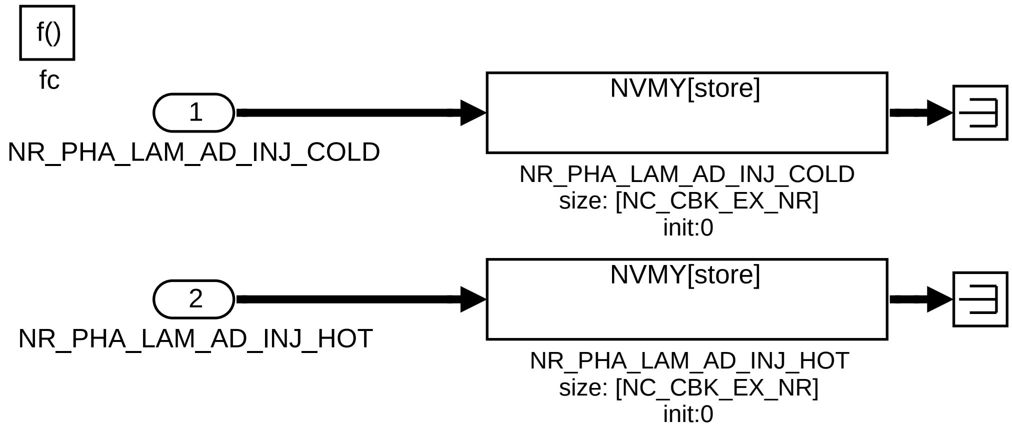


Figure 40:
Path: MFMA_REQCOAIMM0/INI/NVMY_WR

49.3.1.3 Initialization

All variables are initialized with 0 at reset or IGKON.

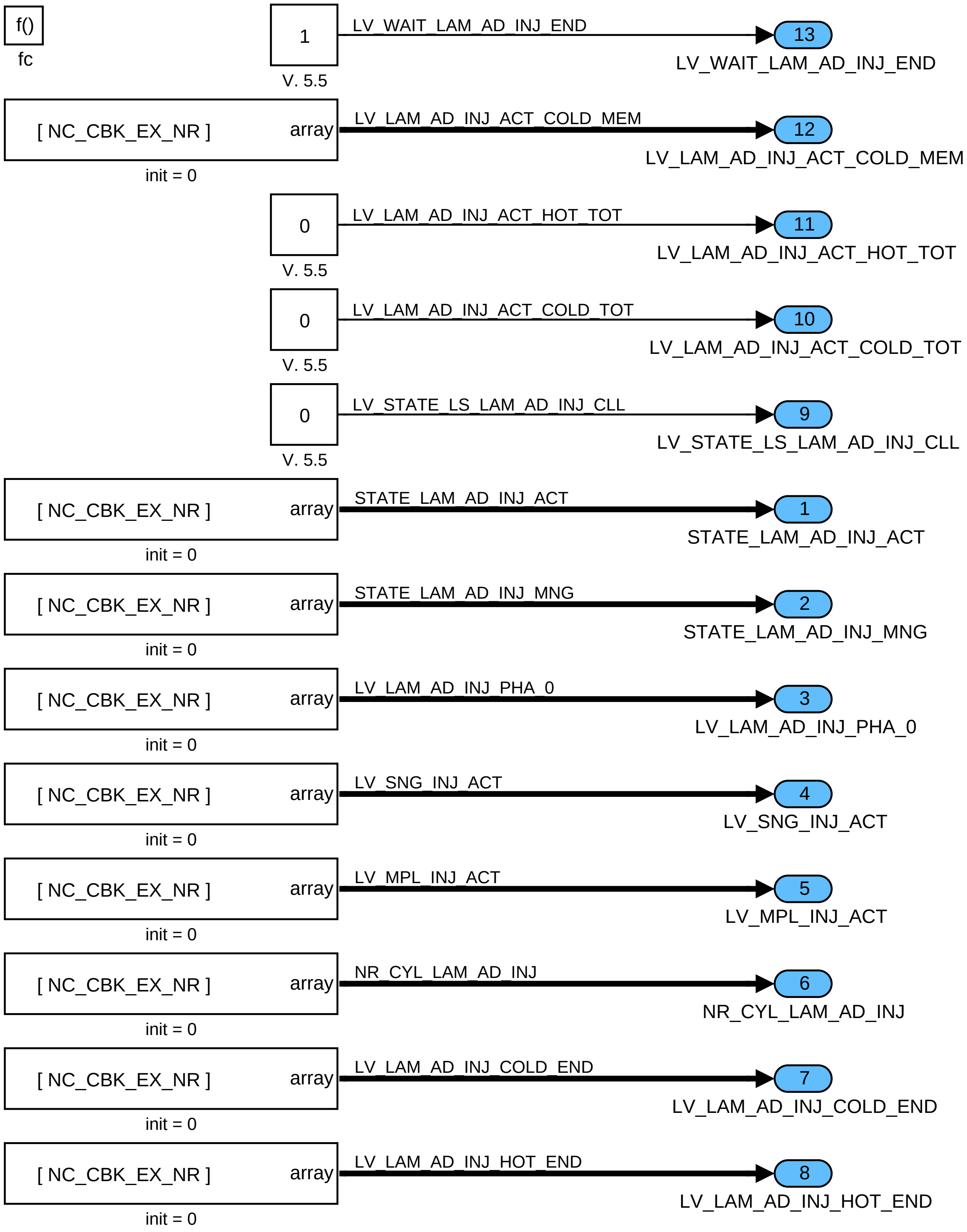


Figure 41:
Path: MFMA_REQCOAIMM0/INI/INI

49.3.2 Bank specific calculations

All calculations are done for every bank. The value for the specific bank is denoted in the graphical model with the suffix _i whereas in the textual description the bank specific values are referred to with [i].

49.3.2.1 Preliminary calculations

49.3.2.1.1 Calculation of LV_LAM_AD_INJ_ACT_COLD_MEM, LV_WAIT_LAM_AD_INJ_END and LV_STATE_LS_LAM_AD_INJ_CLL

The flag LV_WAIT_LAM_AD_INJ_END is set if the adaptation signals the end of the mean value calculation for both banks. The values of LV_LAM_AD_INJ_ACT_COLD are memorized in LV_LAM_AD_INJ_ACT_COLD_MEM as this is used for determination of the phase numbers NR_PHA_LAM_AD_INJ_COLD/HOT. To get a bank synchronous switching of the state machine, the activation conditions for the lambda controller are checked for all banks and stored in LV_STATE_LS_LAM_AD_INJ_CLL.

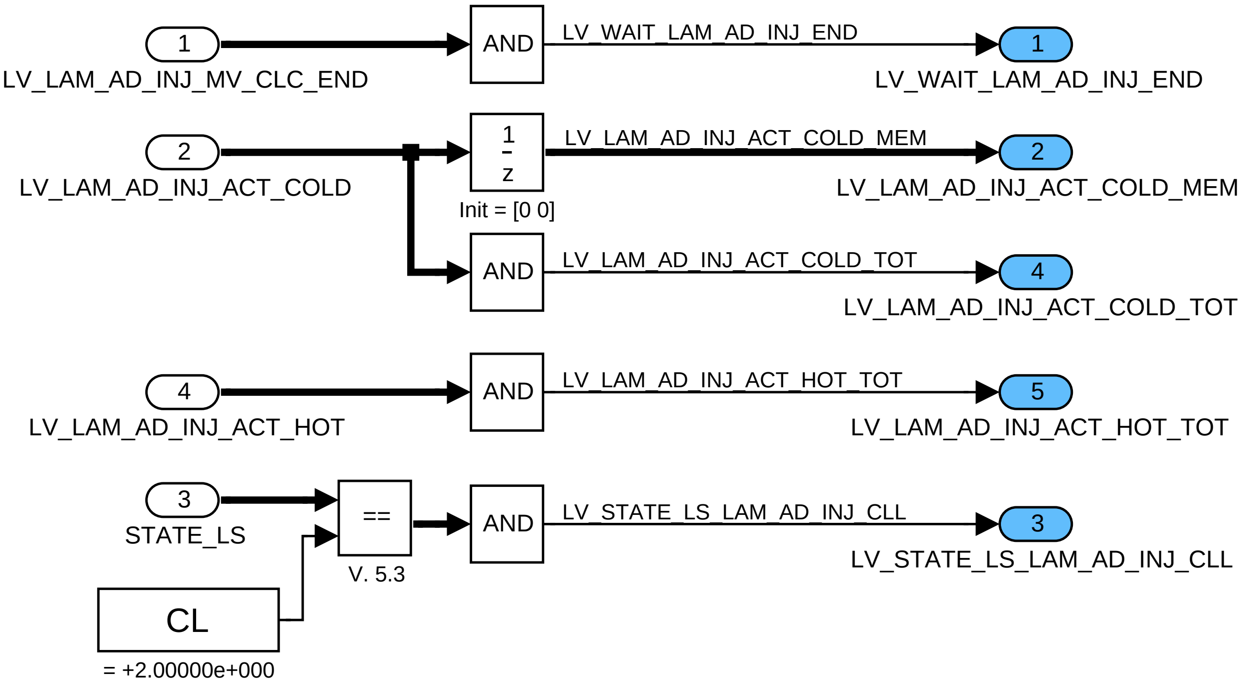


Figure 42:
Path: MFMA_REQCOAIMM0/OPM/CLC_MEM/CLC

49.3.2.2 Adaptation manager

The adaptation manager consists of three states STATE_LAM_AD_INJ[i]:
NOT_ACT: inactive state
CLC_CYL_STATE: intermediate state to determine single/multiple injection-mode for the next adaptation phase
WAIT_LAM_AD_INJ: the adaptation enabled for the injection-mode defined in the previous state

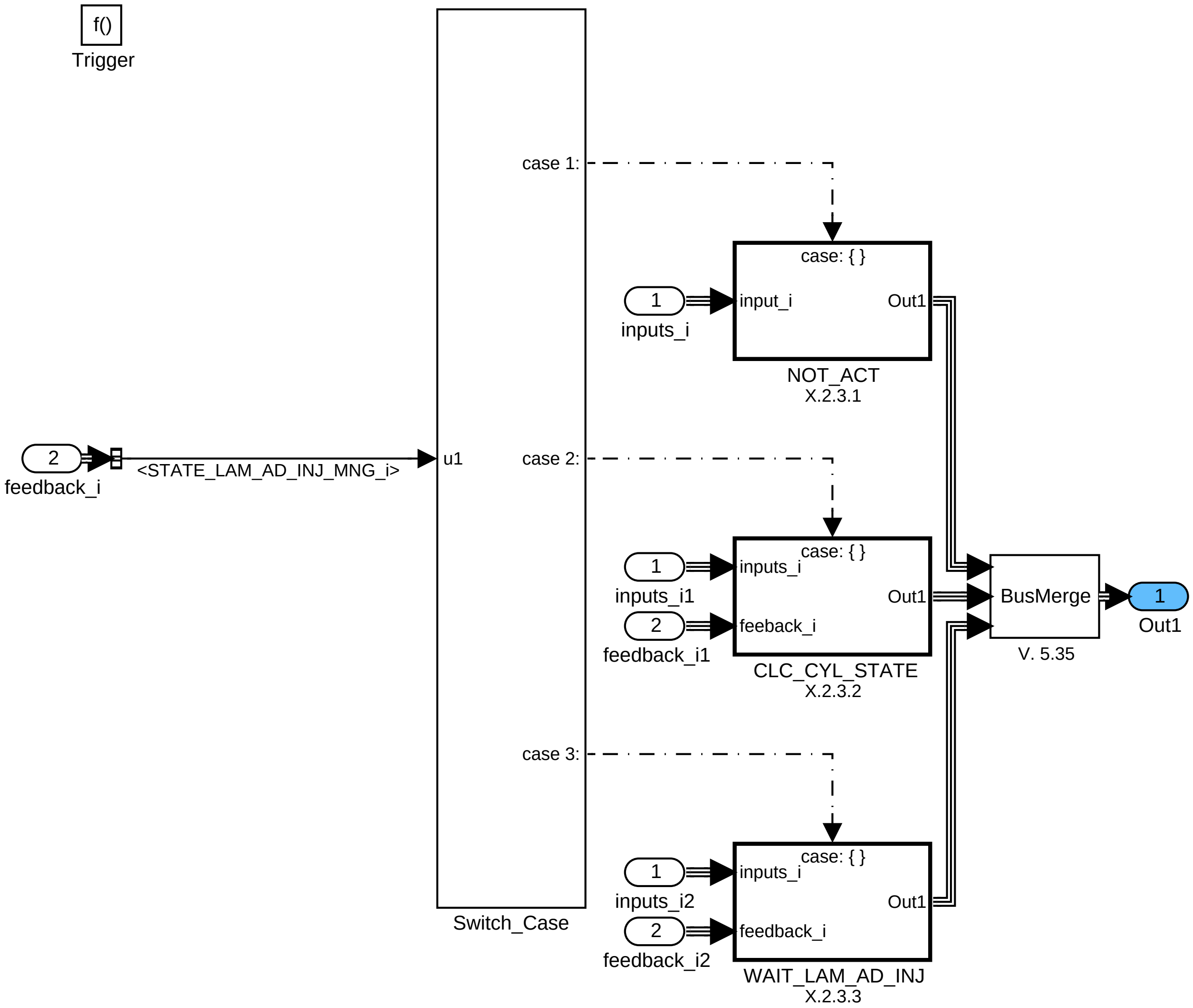


Figure 43:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG

49.3.2.2.1 NOT_ACT

In this state the adaptation is inactive. If the conditions for the adaptation in the upper or lower temperature range apply, the injection mode control is requested by setting the state STATE_LAM_AD_INJ_ACT[i] different from zero.

49.3.2.2.1.1 Transition conditions

If the adaptation conditions apply the injection mode control is requested and the state is left if it is granted (LV_LAM_AD_INJ_CUS_ACK = 1).

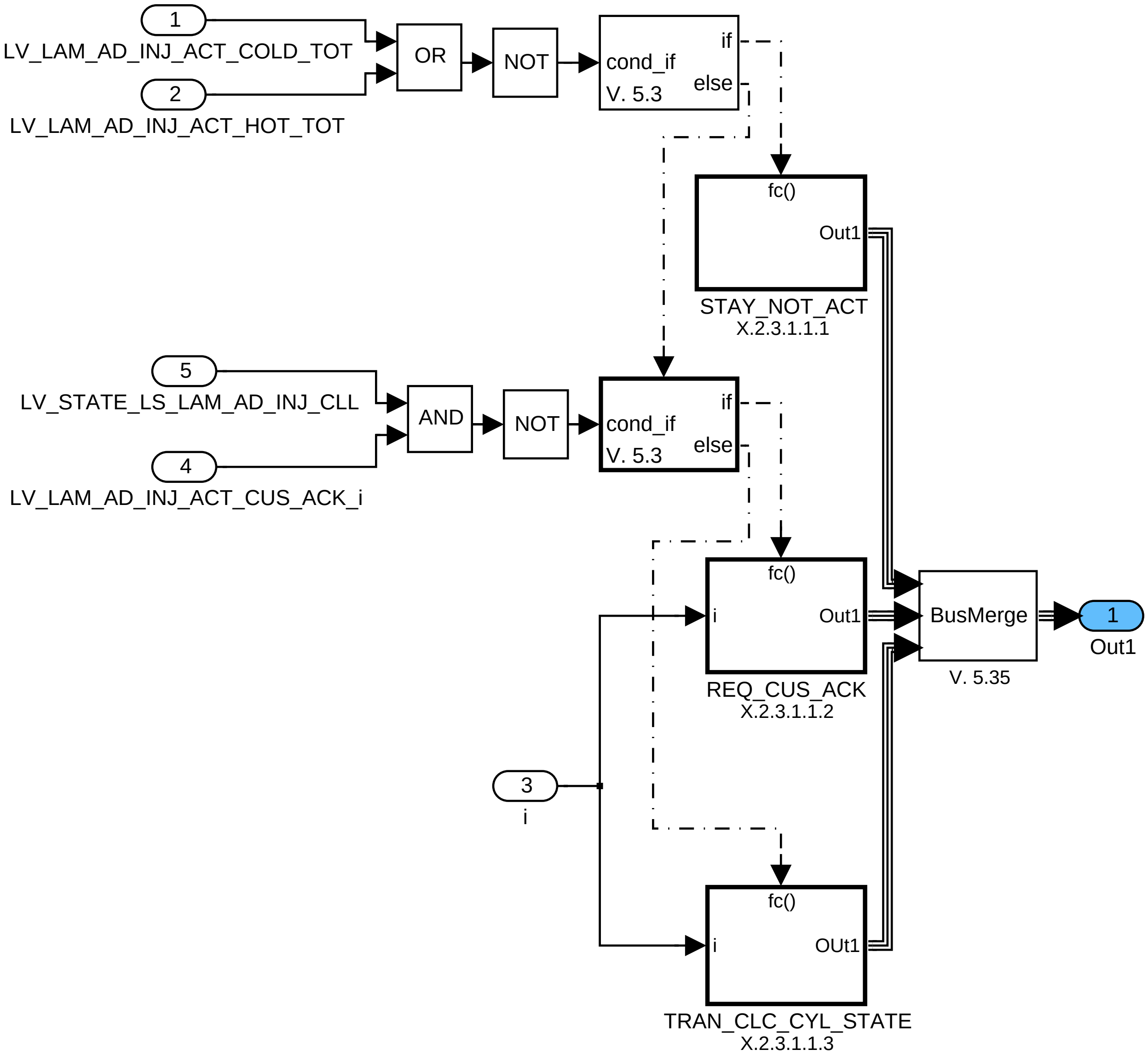


Figure 44:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/NOT_ACT/CLC

49.3.2.2.1.1.1 Remaining in NOT_ACT

49.3.2.2.1.1.1.1 Reset of the request for the injection mode control

The request is reset if the conditions are no more fulfilled.

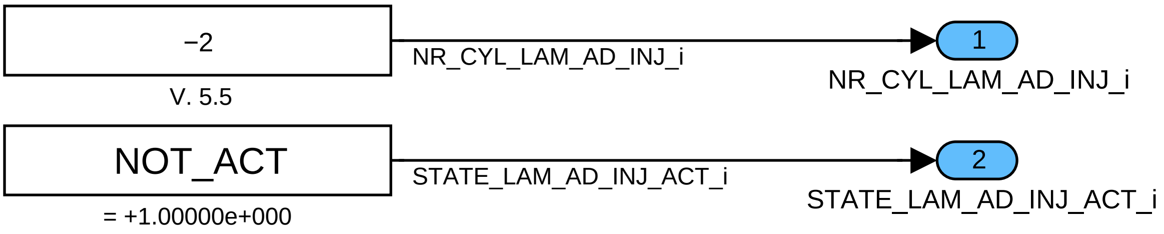


Figure 45:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/NOT_ACT/CLC/STAY_NOT_ACT/CLC

49.3.2.2.1.1.2 Start of injection mode control handshake

The injection mode for the bank is requested.

49.3.2.2.1.1.2.1 Request for injection mode control

The control of the injection mode is requested by setting the STATE_LAM_AD_INJ_ACT[i] to the first element of ID_NR_CYL_LAM_AD_INJ[i], which should be set to one or two to request single injections.

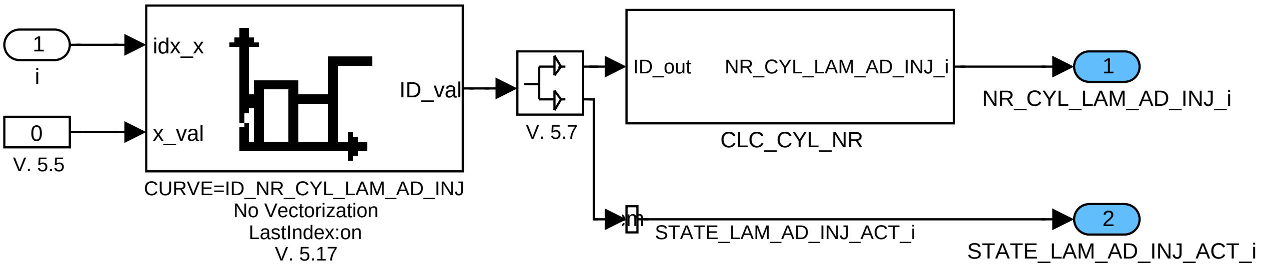


Figure 46:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/NOT_ACT/CLC/REQ_CUS_ACK/CLC

49.3.2.2.1.1.3 Injection mode control granted

49.3.2.2.1.1.3.1 Transition to CLC_CYL_STATE

If the acknowledge bit for injection mode control LV _LAM_AD_INJ_CUS_ACK is set, the state is left towards CLC_CYL_STATE.

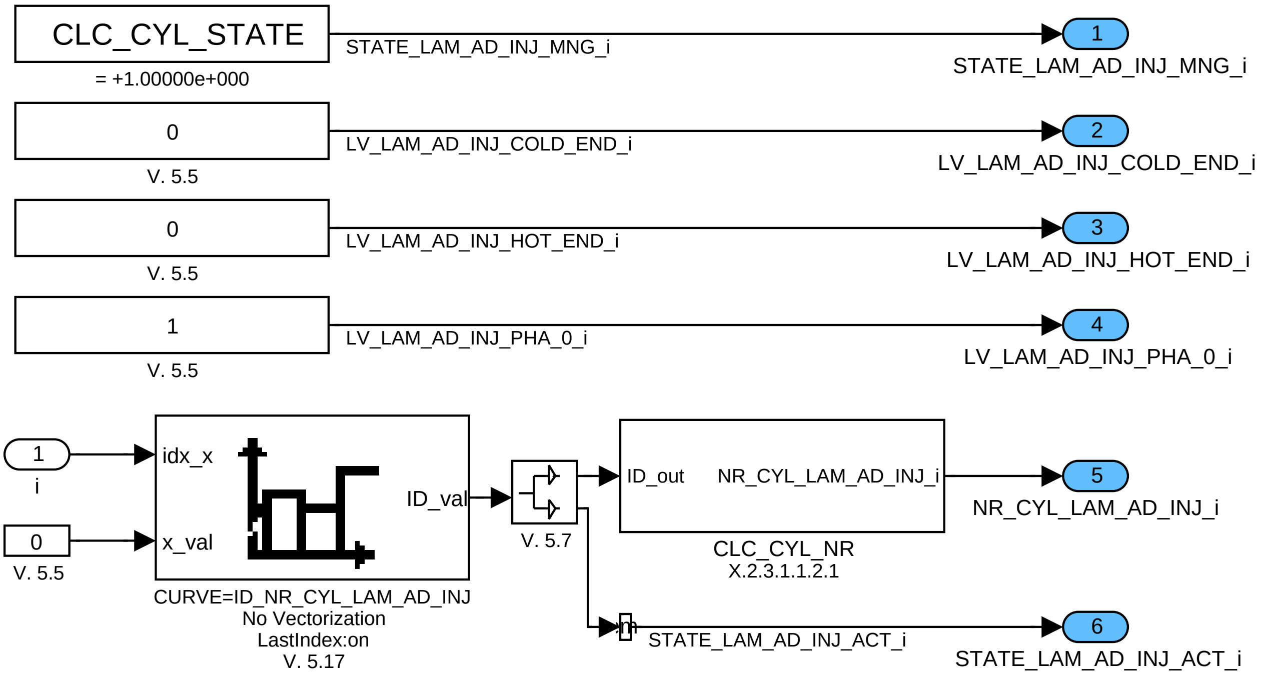


Figure 47:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/NOT_ACT/CLC/TRAN_CLC_CYL_STATE/CLC

49.3.2.2.1.1.3.1.1 Calculation of the cylinder number from STATE_LAM_AD_INJ_ACR

The conversion from ID_out[i] to NR_CYL_LAM_AD_INJ[i] can be done in the code as follows:
NR_CYL_LAM_AD_INJ[i] = (signed)((ID_out+1) >> 1) - 2;
Therewith the conversion should be done as follows:

ID_OUT	NR_CYL_LAM_AD_INJ
0	-2
1	-1
2	-1
3	0
4	0
...	...
18	7

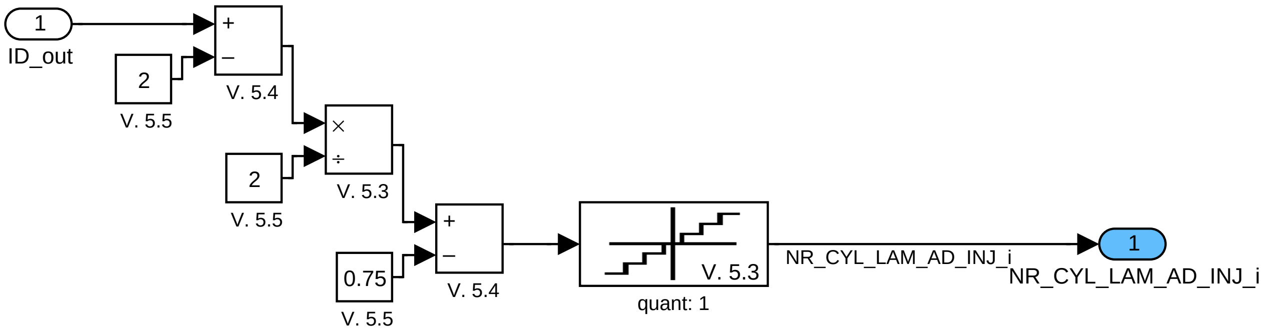


Figure 48:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/NOT_ACT/CLC/TRAN_CLC_CYL_STATE/CLC/CLC_CYL_NR

49.3.2.2.2 CLC_CYL_STATE

This state is used for determination of the injection mode for the next adaptation phase.

49.3.2.2.2.1 Transition conditions

If the cylinder number of the current bank NR_CYL_LAM_AD_INJ[i] is equal to -2, the current adaptation is assumed to be finished and the state is left to NOT_ACT. If the adaptation is interrupted due to the lack of the necessary conditions, a transition to NOT_ACT is performed, too. Otherwise the state changes to WAIT_LAM_AD_INJ.

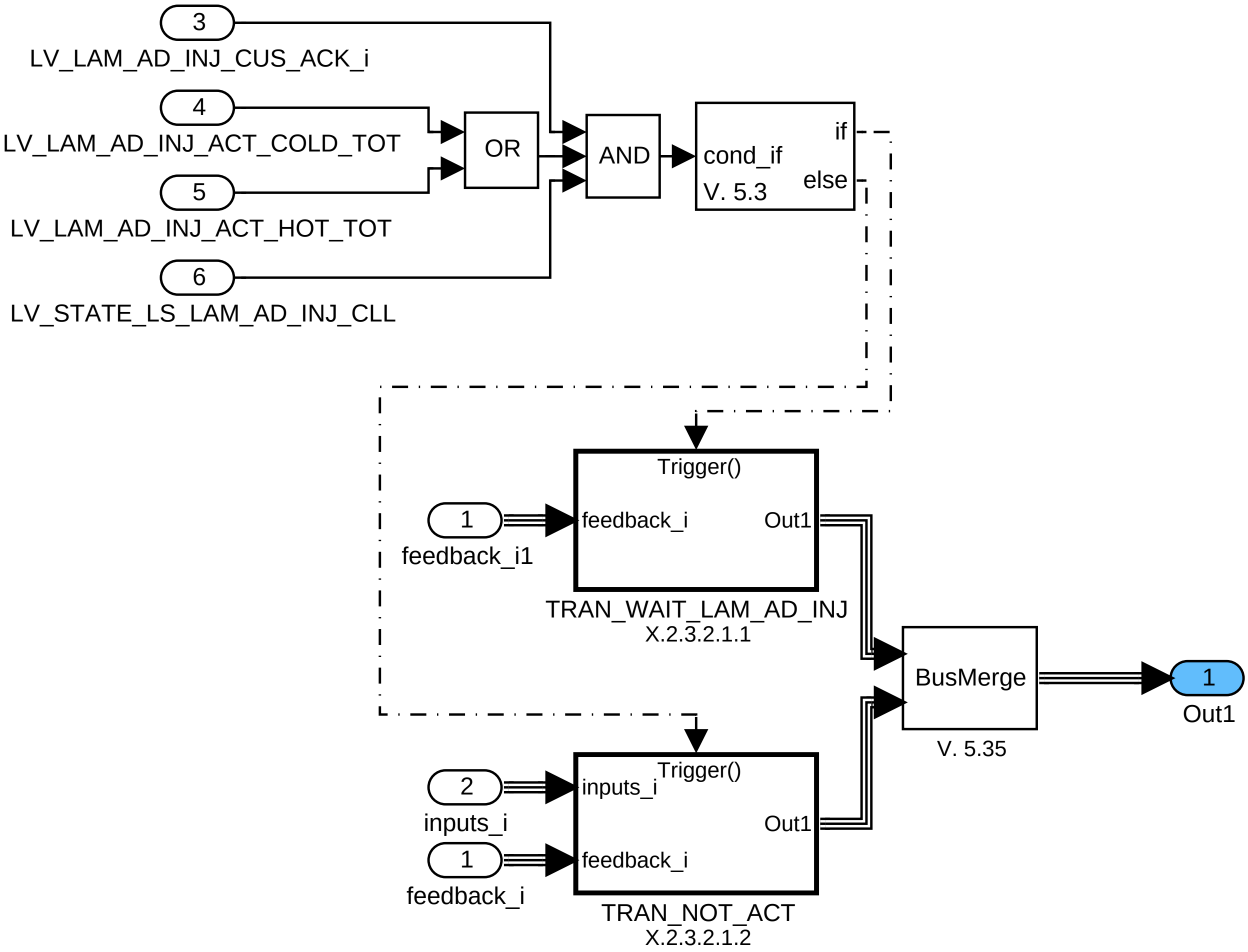


Figure 49:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/CLC_CYL_STATE/CLC

49.3.2.2.2.1.1 Transition to WAIT_LAM_AD_INJ

49.3.2.2.2.1.1.1 Setting of LV_SNG/MPL_ACT

If the STATE_LAM_AD_INJ_ACT[i] is bigger than 2, multiple injections for this cylinder number are requested and therefore LV_MPL_INJ_ACT[i] is set. Otherwise LV_SNG_INJ_ACT[i] is set to enable the corresponding adaptation.

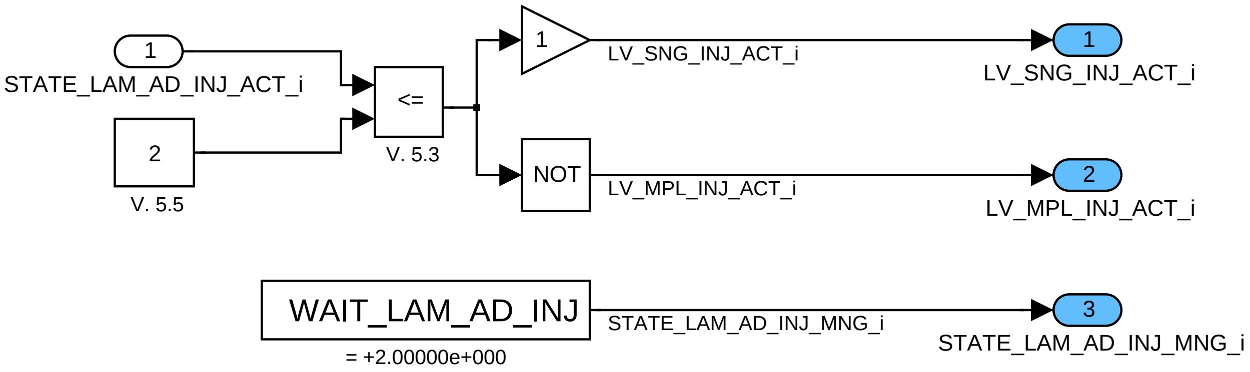


Figure 50:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/CLC_CYL_STATE/CLC/TRAN_WAIT_LAM_AD_INJ/CLC

49.3.2.2.2.1.2 Transition to NOT_ACT due to interruption

49.3.2.2.2.1.2.1 Saving of the last completed phase number

If the adaptation is interrupted, the last completed adaptation phase number is stored in NR_PHA_LAM_AD_INJ_COLD/HOT[i], to make sure the adaptation is continued with the correct phase number if the conditions apply again and therefore all cylinders are adapted.

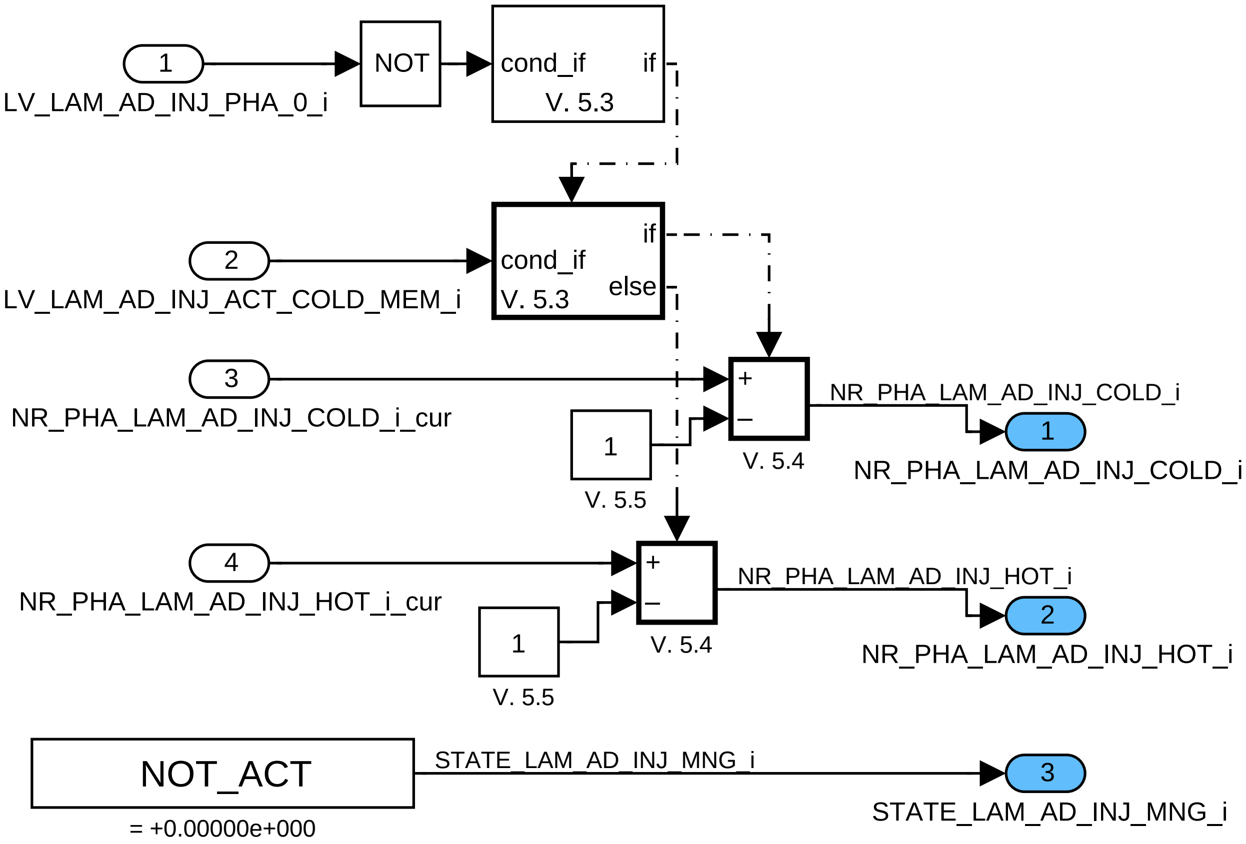


Figure 51:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/CLC_CYL_STATE/CLC/TRAN_NOT_ACT/CLC

49.3.2.2.3 WAIT_LAM_AD_INJ

This state waits for the current adaptation phase to be completed or the adaptation conditions to be stopped.

49.3.2.2.3.1 Transitions conditions

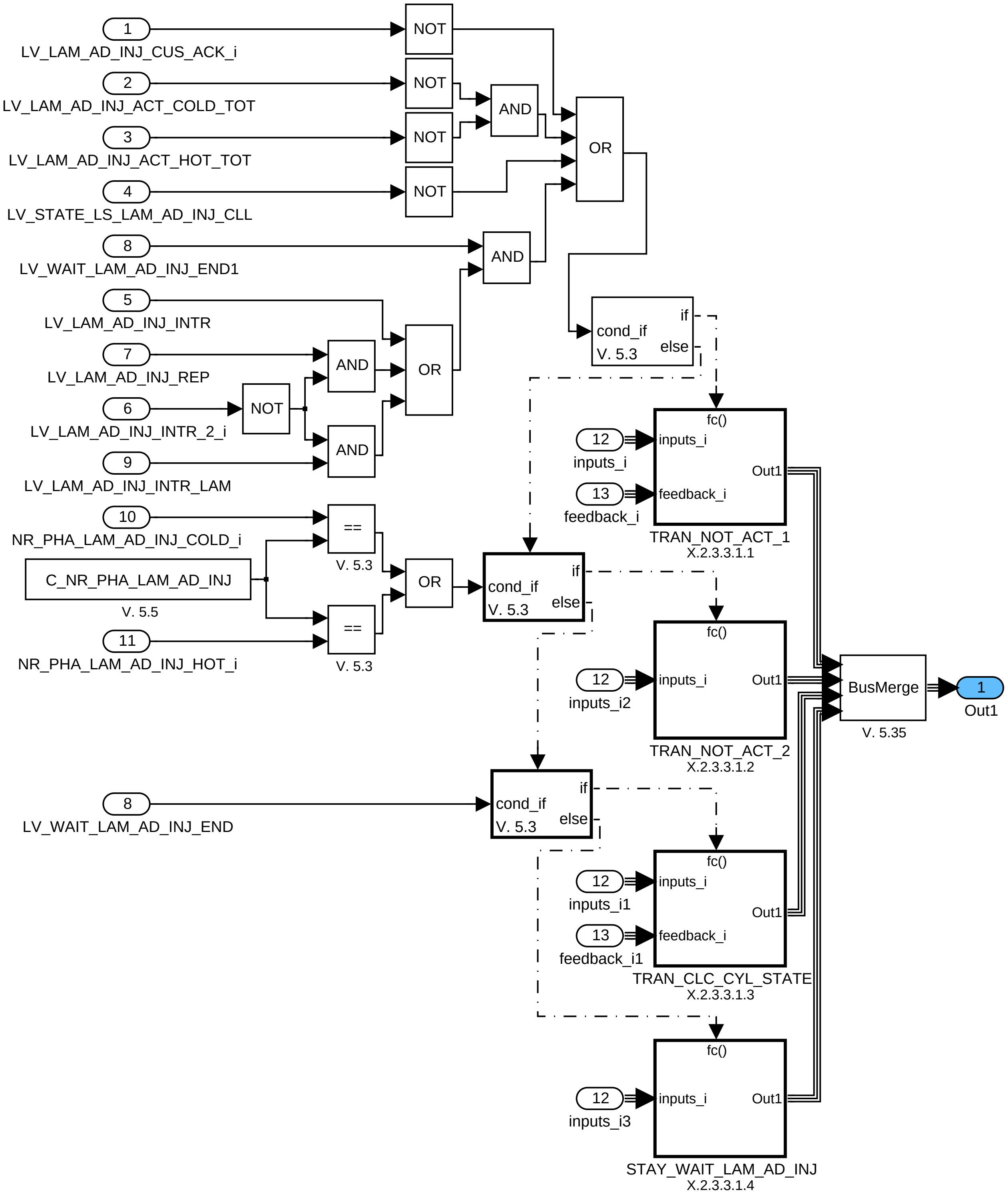


Figure 52:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/WAIT_LAM_AD_INJ/CLC

49.3.2.2.3.1.1 Transition to NOT_ACT due to interruption

49.3.2.2.3.1.1.1 Saving of the last completed phase number

If the adaptation is interrupted, the last completed adaptation phase number is stored in NR_PHA_LAM_AD_INJ_COLD/HOT[i], to make sure the adaptation is continued with the correct phase number if the conditions apply again and therefore all cylinders are adapted. Furthermore the activation flags LV_SNG/MPL_ACT[i] are reset, to ensure that the current adaptation phase is stopped.

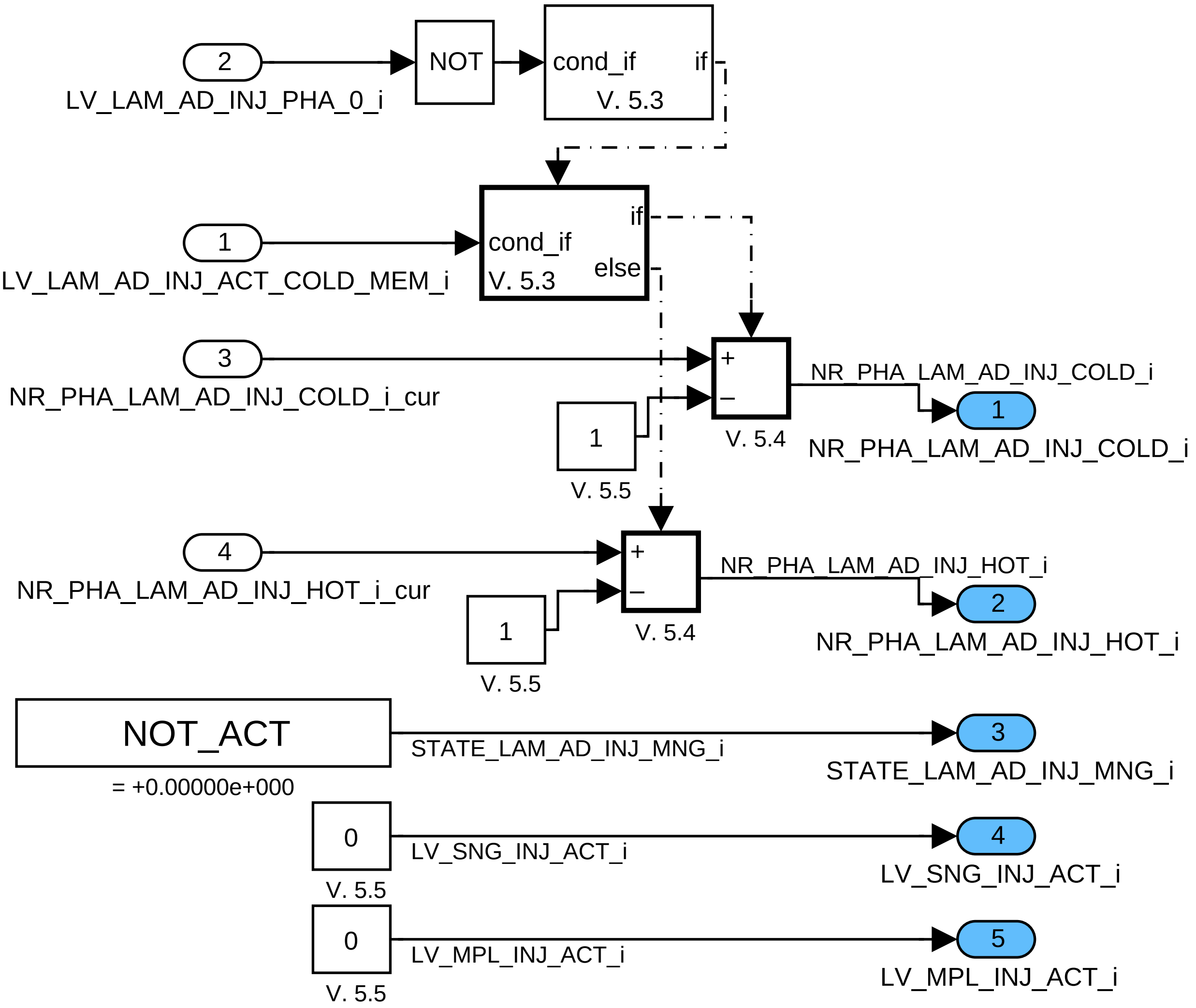


Figure 53:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/WAIT_LAM_AD_INJ/CLC/TRAN_NOT_ACT_1/CLC

49.3.2.2.3.1.2 Transition to NOT_ACT due to finished adaptation

49.3.2.2.3.1.2.1 Setting of LV_LAM_AD_INJ_COLD/HOT_END

As the adaptation for the current temperature range is finished, the corresponding end-flag is set and the corresponding phase number NR_PHA_LAM_AD_INJ_COLD/HOT[i] is reset to 0. Therefore the adaptation starts from the beginning if the conditions apply for the next time.

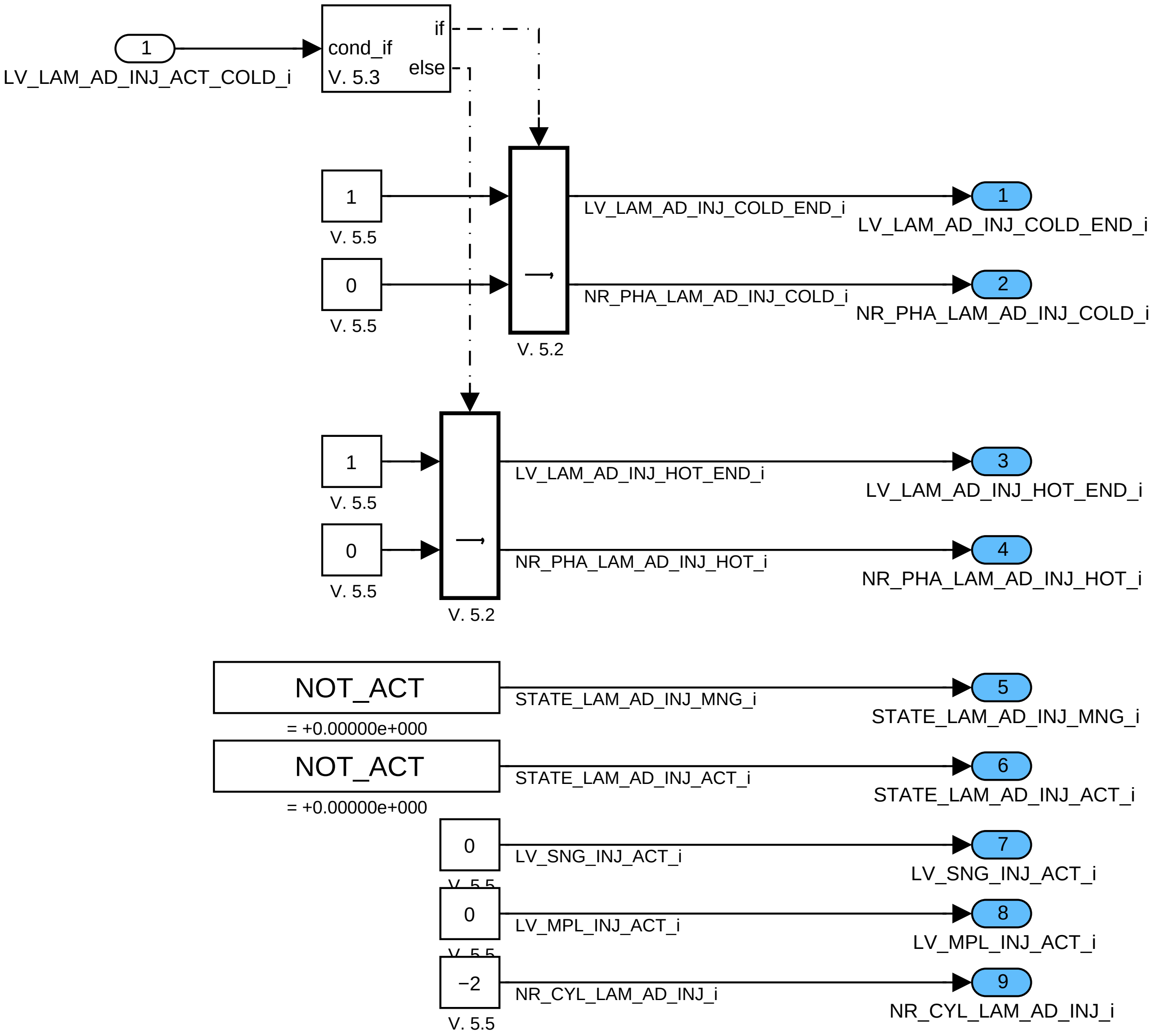


Figure 54:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/WAIT_LAM_AD_INJ/CLC/TRAN_NOT_ACT_2/CLC

49.3.2.2.3.1.3 Transition to CLC_CYL_STATE

49.3.2.2.3.1.3.1 Determination of the injection mode for the next adaptation phase

The injection mode for the next adaptation phase is determined from the map ID_NR_CYL_LAM_AD_INJ[i] and requested via STATE_LAM_AD_INJ_ACT[i]. If multiple injections for one cylinder of the current bank is requested, the corresponding cylinder number is stored in NR_CYL_LAM_AD_INJ[i]. Otherwise -1 (single injection) or -2(end of adaptation) is used.

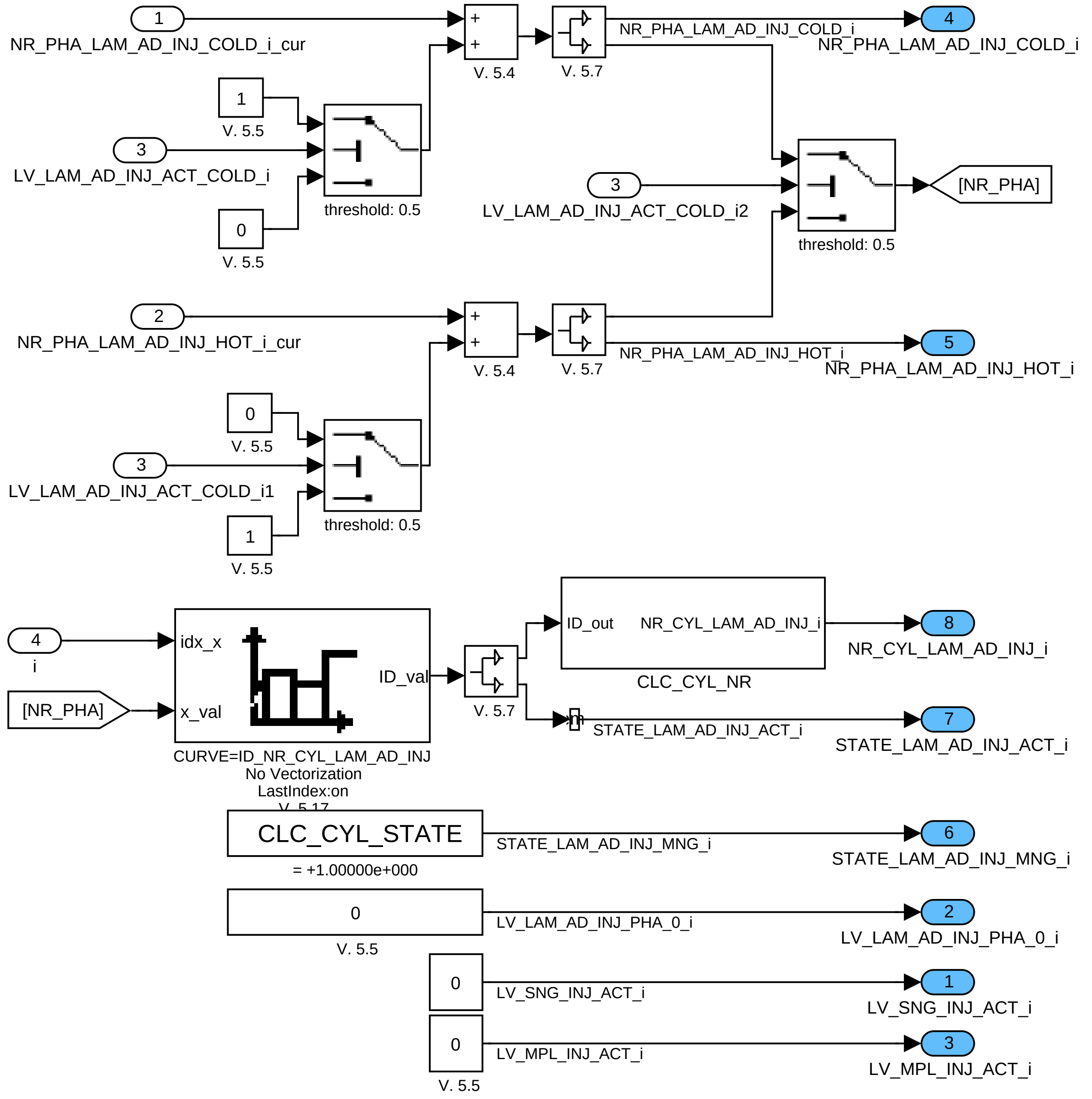


Figure 55:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/WAIT_LAM_AD_INJ/CLC/TRAN_CLC_CYL_STATE/CLC_1

49.3.2.2.3.1.3.2 Cancelling of multiple injection due to misfire error

If the flag LV_ERR_MIS_LAM_AD_INJ[x] (where x denotes the current cylinder number) is set, i.e. a repeated misfire error is detected, the request for multiple injections is cancelled and another single injection phase will be started to continue with the adaptation of the other cylinders.

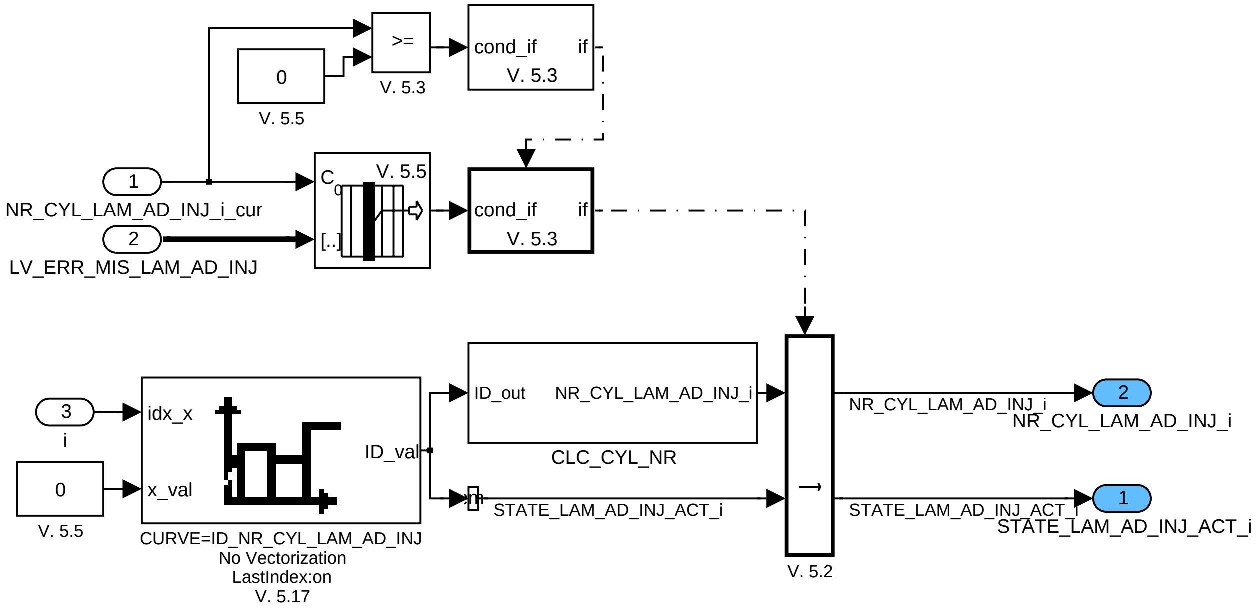


Figure 56:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/WAIT_LAM_AD_INJ/CLC/TRAN_CLC_CYL_STATE/CLC_2

49.3.2.2.3.1.4 Waiting for the end of the current adaptation cycle

49.3.2.2.3.1.4.1 Resetting of LV_SNG/MPL_INJ_ACT

If one bank signals the end of its adaptation phase (LV_LAM_AD_INJ_MV_CLC_END[i]), the corresponding activation flags LV_SNG/MPL_INJ_ACT[i] are reset. If the flag LV_LAM_AD_INJ_INTR_2[i] is set, repeated misfire is detected on the current cylinder and therefore single injection is requested.

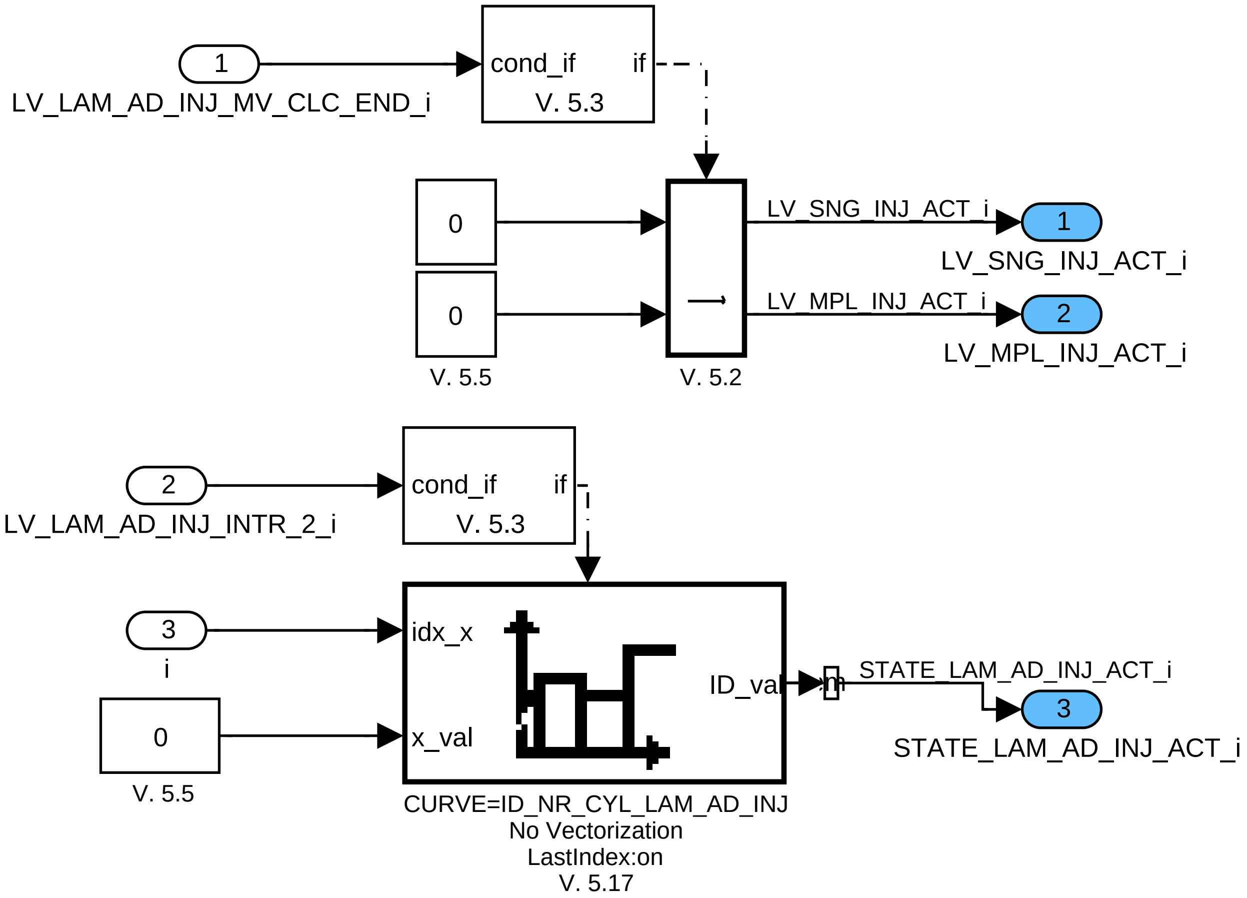


Figure 57:
Path: MFMA_REQCOAIMM0/OPM/STATE_MNG/WAIT_LAM_AD_INJ/CLC/STAY_WAIT_LAM_AD_INJ/CLC

49.3.3 Operations at transition engine stop to engine run

49.3.3.1 Resetting of the phase numbers

If a resetting of the NVMY data via service tester is requested with the flag LV_LAM_AD_INJ_CLR_AD_EXT, the phase numbers of both banks for the upper and lower temperature range are reset to 0.

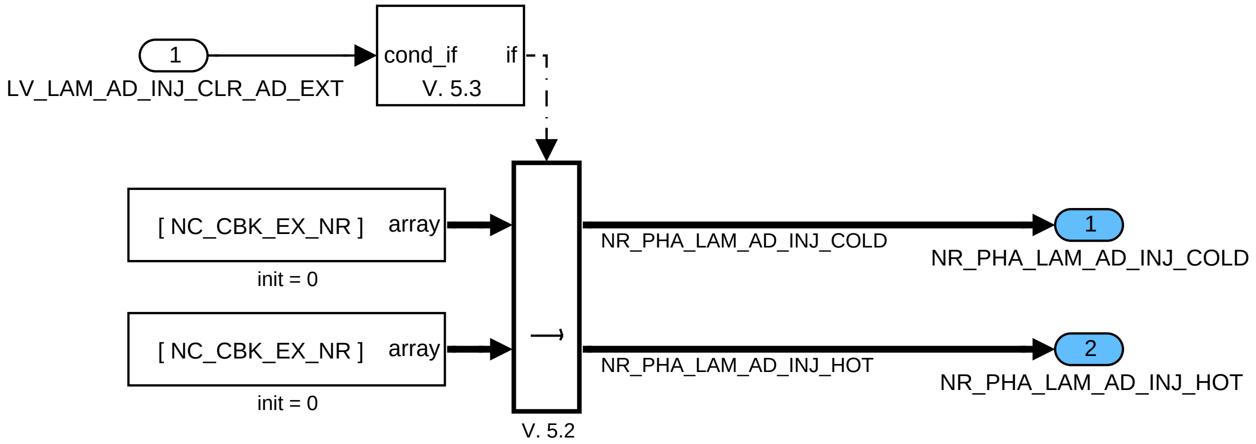


Figure 58:
Path: MFMA_REQCOAIMM0/ES2ERU/CLC

49.4 Lambda adaptation via injection mode (Appl. Inc.)

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_INH_LAM_AD_INJ [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Lambda adapation via injection mode inhibition flag					

Input Data:

NC_CBK_EX_NR	LV_ERR_CPS	LV_ERR_MEC_OPEN_CPS	LV_ERR_DIAGCPS
LV_ERR_TCO	LV_ERR_MAF	LV_ERR_TPS	LV_ERR_TPS_1
LV_ERR_TPS_2	LV_ERR_TPS_AD	LV_ERR_TPS_MAF_1	LV_ERR_TPS_MAF_2
LV_ERR_TPS_ST_CHK_1	LV_ERR_TPS_ST_CHK_2	LV_ERR_TPS_RATIO	LV_ERR_RATIO_CHK
LV_ERR_LOAD_TPS_PLAUS	LV_ERR_MAP	LV_ERR_MAP_PLAUS	LV_ERR_MAP_DIP_SHIFT
LV_ERR_MAP_TPS_PLAUS	LV_ERR_IVVT	LV_ERR_AMP	LV_ERR_AMP_PLAUS
LV_ERR_VCV	LV_ERR_VCV_PLAUS	LV_ERR_FUP	LV_ERR_FUP_EFP
LV_ERR_FUP_MFP_PLAUS	LV_ERR_H_PRS_SYS	LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3
LV_ERR_MTC_DR	LV_ERR_CAM	LV_LIH_ERR_CRK	LV_ERR_CRK
LV_ERR_FTL_MIN	LV_ERR_TIA_IM	LV_ERR_TIA_THR	LV_MIS_STATE_B1
LV_MIS_STATE_B4	LV_ACT_DIAGCPS	STATE_CAT_DIAG [NC_CBK_EX_NR]	LV_ERR_CAM_DE_IVVT_IN [NC_NR_CBK_IVVT]
LV_ERR_CAM_DE_IVVT_EX [NC_NR_CBK_IVVT]	LV_ERR_REF_CRK_CAM_IN [NC_NR_CAM_CBK]	LV_ERR_REF_CRK_CAM_EX [NC_NR_CAM_CBK]	LV_ERR_TOOTH_OFF_IN [NC_NR_CAM_CBK]
LV_ERR_TOOTH_OFF_EX [NC_NR_CAM_CBK]	LV_ERR_IV [NC_CYL_NR]	LV_ERR_CYL_BAL_LAM [NC_CYL_NR]	LV_ERR_DELTA_I_LAM [NC_CBK_EX_NR]
LV_ERR_VLS_DOWN_DIF [NC_CBK_EX_NR]	LV_ERR_TTIP_MES_LSH_UP [NC_CBK_EX_NR]	STATE_LS [NC_CBK_EX_NR]	LV_ERR_LSH_DOWN [NC_CBK_EX_NR]
LV_ERR_CAT_DIAG [NC_CBK_EX_NR]	LV_ERR_FSD [NC_CBK_EX_NR]	LV_ERR_FSD_LAM_LIM [NC_CBK_EX_NR]	LV_ERR_WG_DR [NC_CBK_EX_NR]

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
LC_LAM_AD_INJ_INH	1	0... 1H	0... 1	1	[-]
Manual inhibition of lambda adaptation via injection mode					

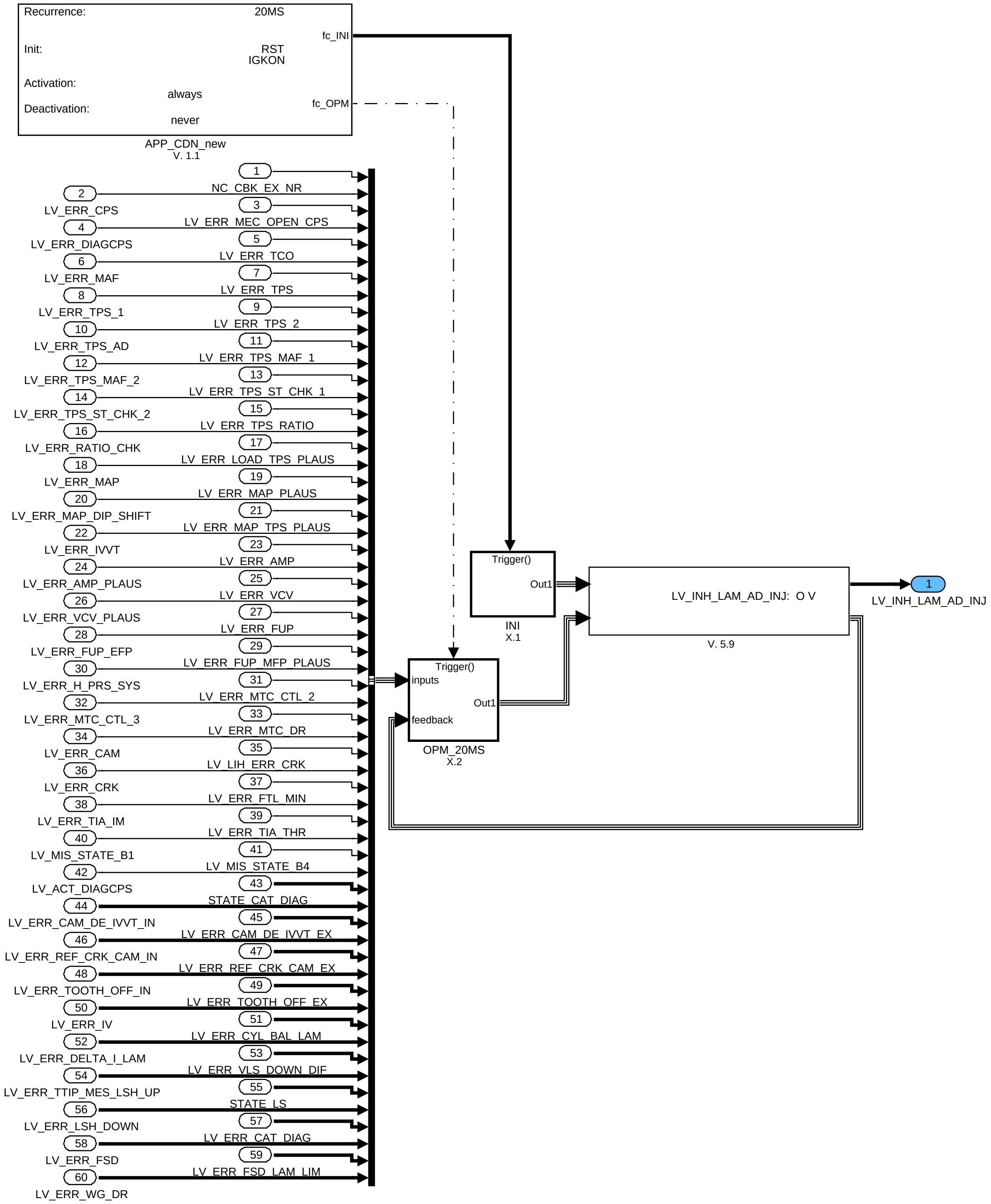
General Information

In the application incidence module the inhibition flag LV_INH_LAM_AD_INJ[i] for every exhaust cylinder bank is calculated.

Application Conditions

Initialization: RST, IGKON
Recurrence: 20MS
Activation: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.0.4 17−May−2006

Figure 59:
Path: MFMA_ISPCLAIMI0

49.4.1 Initialization

The inhibition flag is initialized at reset and at ignition key on.

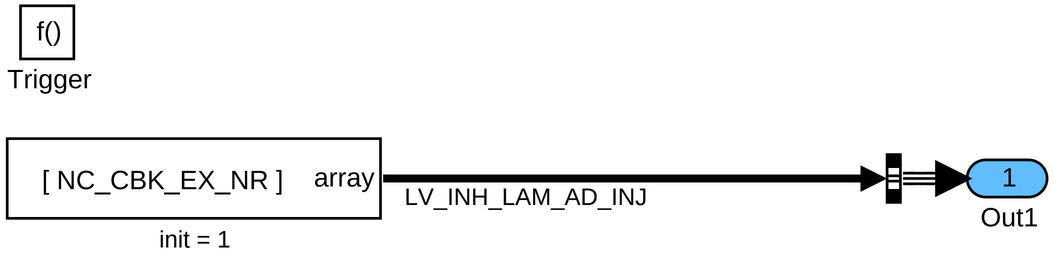


Figure 60:
Path: MFMA_ISPCLAIMI0/INI

49.4.2 Inhibition conditions calculation

The inhibition conditions calculation is divided into two parts. The first part consists of conditions, which do not depend on a specific exhaust cylinder bank, and therefore any of the conditions will inhibit the adaptation of all banks. The second part consists only of conditions, which depend on a specific bank and will inhibit only the adaptation on the corresponding bank.

49.4.2.1 Bank independent conditions

The bank independent conditions are also split in two parts, where the first part uses all scalar conditions/error flags and the second part uses the non-scalar (array) conditions.

49.4.2.1.1 Scalar conditions

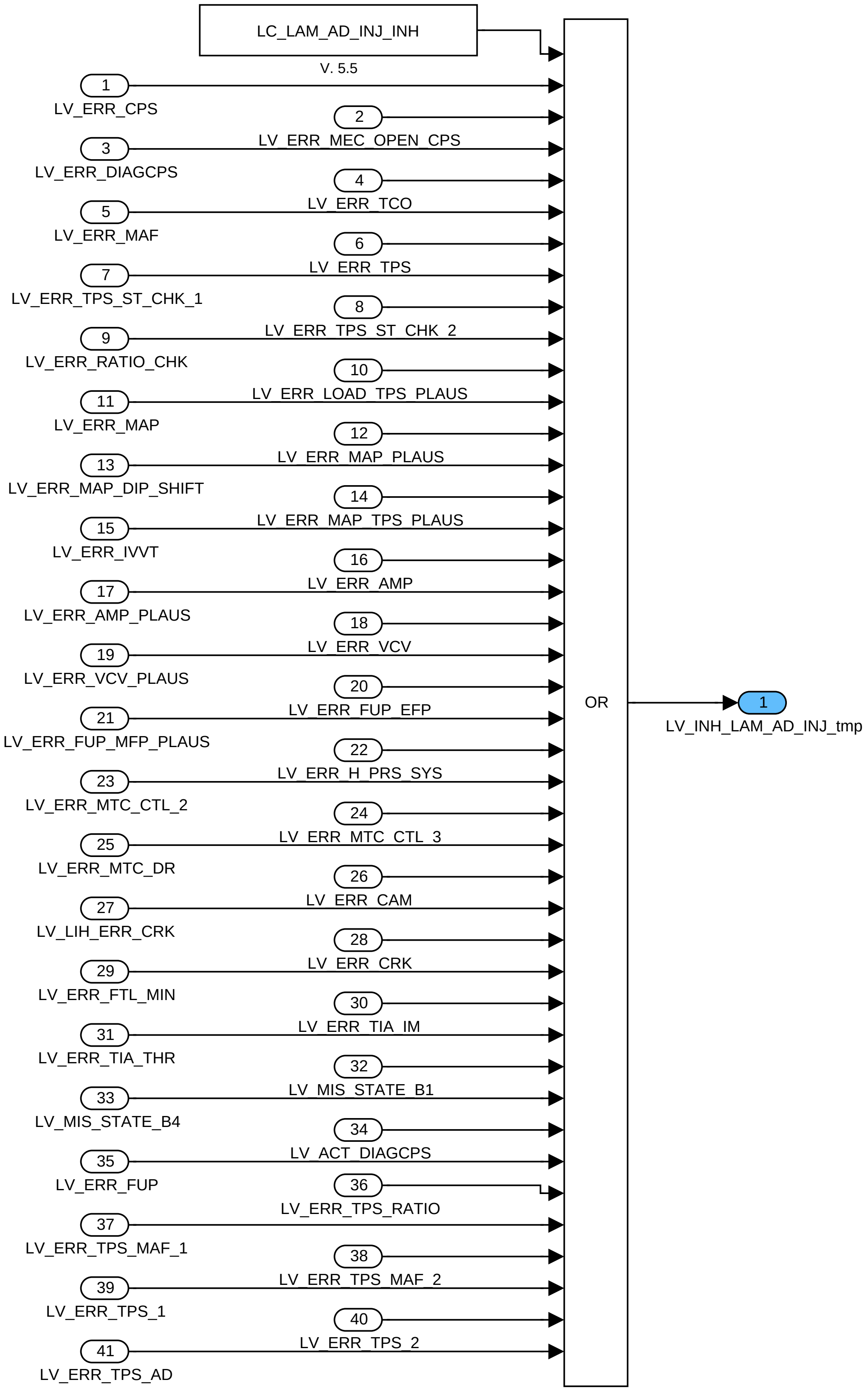


Figure 61:
Path: MFMA_ISPCLAIMI0/OPM_20MS/BANK_INDEP_CLC/CLC_1

49.4.2.1.2 Non-scalar conditions

In this block the all conditions whose dimension can be bigger than 1 are used for the calculation of the inhibition flag. The OR-blocks with only one input return true of at least one index of the array is set to 1.

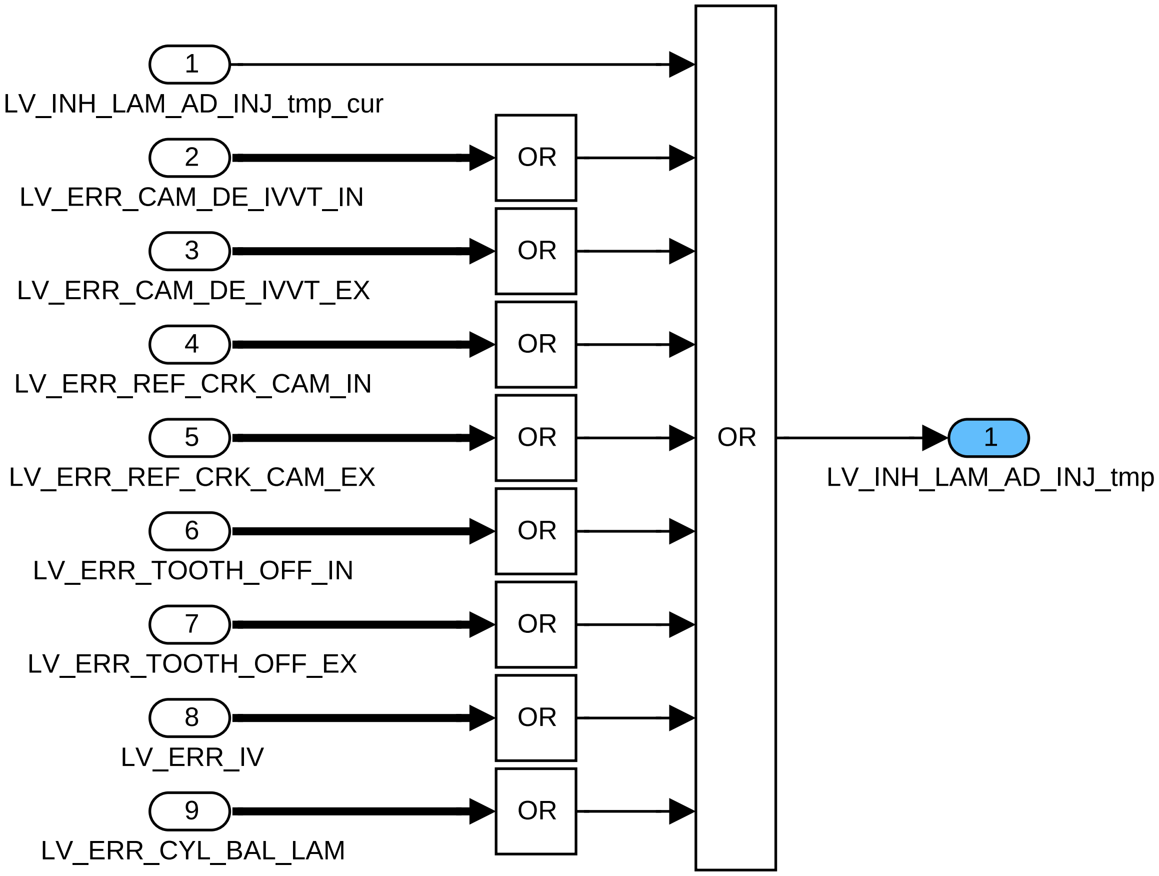


Figure 62:
Path: MFMA_ISPCLAIMI0/OPM_20MS/BANK_INDEP_CLC/CLC_2

49.4.2.2 Bank specific calculations

The bank specific calculations are done for every exhaust cylinder bank (1...NC_CBK_EX_NR). The index of the current cylinder bank is denoted with the suffix _i in the model whereas in the text square brackets are used (e.g. LV_INH_LAM_AD_INJ[i]).

49.4.2.2.1 Bank selective inhibition conditions

If one of the bank selective conditions applies, only the adaptation of the corresponding bank is inhibited. If LV_INH_LAM_AD_INJ_tmp (bank independent condition) is set, both banks will be inhibited.

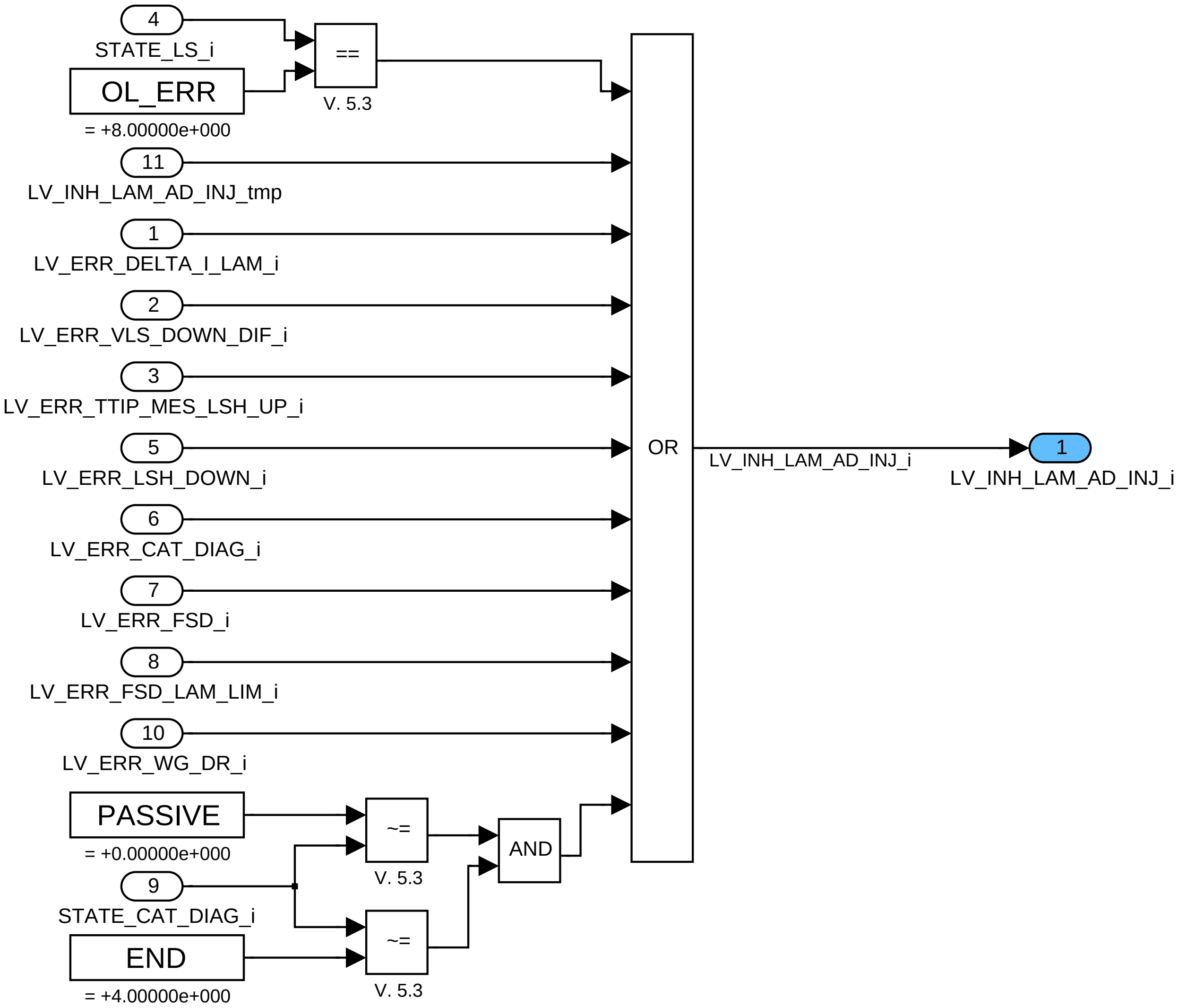


Figure 63:
Path: MFMA_ISPCLAIMI0/OPM_20MS/BANK_SPECIFIC_CLC/CLC

49.5 Lambda adaptation via injection mode

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
FAC_LAM_ADJ_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Lambda controller difference between single injection and multiple injection					
FAC_LAM_MV_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Lamda controller mean value for lambda adaptation via injection mode					
FAC_LAM_MV_MMV_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Lamda controller moving mean value for lambda adaptation via injection mode					
FAC_LAM_MV_MPL_INJ_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Lambda controller mean value at multiple injection					
FAC_LAM_MV_SNG_INJ_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Lambda controller mean value at single injection					
LV_LAM_AD_INJ_MV_CLC_END [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
Lambda mean value calculation for lambda adaptation via injection mode finished					
MAF_INT_LAM_AD_INJ [NC_CBK_EX_NR]	V	0... FFFFH	0... 1023.984375	0.015625	[g]
MAF integral for lambda adaptation via injection					
MFF_ADD_COLD_LAM_AD_INJ [NC_CYL_NR]	V/S	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Additional fuel flow at cold condition for lambda adaptation via injection					
MFF_ADD_HOT_LAM_AD_INJ [NC_CYL_NR]	V/S	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Additional fuel flow at hot condition for lambda adaptation via injection					
MFF_ADD_LAM_AD_INJ [NC_CYL_NR]	O/V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Additional fuel mass correction for minimum injection					
MFF_ADD_MIS_COLD_LAM_AD_INJ [NC_CYL_NR]	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Additional fuel mass in case of misfire for lower temperature adapation range					
MFF_ADD_MIS_HOT_LAM_AD_INJ [NC_CYL_NR]	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Additional fuel mass in case of misfire for upper temperature adapation range					
MFF_DELTA_ADD_LAM_AD_INJ [NC_CBK_EX_NR]	V	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Fuel mass difference for one injection pulse for multiple injection					
STATE_LAM_AD_INJ [NC_CBK_EX_NR]	O/V	0 1 2 3	NOT_ACT TRA_PHA MV_CLC LAM_AD	-	[-]
Lambda adaptation via injection mode state					
T_INT_LAM_AD_INJ [NC_CBK_EX_NR]	V	0... FFFFH	0... 1310.7	0.02	[s]
Time integral for lambda adaptation via injection					
TQ_IS_SNG_INJ_LAM_AD_INJ	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Idle speed controller output for monitoring in lambda adaptation via injection mode					

Input Data:

LV_SNG_INJ_ACT [NC_CBK_EX_NR]	LV_MPL_INJ_ACT [NC_CBK_EX_NR]	NR_CYL_LAM_AD_INJ [NC_CBK_EX_NR]	LV_LAM_AD_INJ_ACT_COLD [NC_CBK_EX_NR]

LV_LAM_AD_INJ_INTR	LV_LAM_AD_INJ_INTR_2 [NC_CBK_EX_NR]	CTR_MIS_LAM_AD_INJ [NC_CBK_EX_NR]	TEMP_LAM_AD_INJ
LV_MIS_LAM_AD_INJ [NC_CBK_EX_NR]	LV_LAM_AD_INJ_INTR_LAM	CTR_AD_COLD_LAM_AD_INJ	CTR_AD_HOT_LAM_AD_INJ
C_TEMP_MIN_LAM_AD_INJ_HOT	C_TEMP_MAX_LAM_AD_INJ_COLD	FAC_LAM_LIM [NC_CBK_EX_NR]	MAF_CYL
MFF_SP [NC_CBK_EX_NR]	NC_CBK_EX_NR	NC_CYL_NR	MFF_SP_1_HOM [NC_CYL_NR]
TQ_DIF_I_IS	TQ_DIF_P_D_SLOW_IS	STATE_IV_CHG	LV_LAM_AD_INJ_CLR_AD_EXT
NC_NR_MPL_INJ_LAM_AD_INJ	NC_NR_MIS_REP_LAM_AD_INJ		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CRLC_FAC_LAM_MMV_LAM_AD_INJ	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlaton constant for lambda controller output moving mean value					
C_CRLC_FAC_LAM_MV_LAM_AD_INJ	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlaton constant for lambda controller output mean value					
C_CRLC_MFF_ADD_COLD_LAM_AD_INJ	1	0... 80H	0... 1	7.8125e-3	[-]
Correlation constant for filterd cold adaptation value					
C_CRLC_MFF_ADD_HOT_LAM_AD_INJ	1	0... 80H	0... 1	7.8125e-3	[-]
Correlation constant for filterd hot adaptation value					
C_CRLC_TQ_DIF_SLOW_LAM_AD_INJ	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlation constant for use of TQ_DIF_P_D_SLOW_IS in TQ_IS_SNG_INJ_LAM_AD_INJ calculation					
C_FAC_LAM_ADJ_MIN_LAM_AD_INJ	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Min. difference between lambda mean value at SNG and MPL inj. to update the adaptation value					
C_FAC_LAM_FIL_DIF_LAM_AD_INJ	1	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Threshold for difference of first and second filterd lambda value to detect lambda stabilization					
C_FAC_MFF_COR_CYL_LAM_AD_INJ [NC_CYL_NR]	1	0... FFH	0... 3.984375	0.015625	[-]
Weighting factor for each cylinder to map adjust cylinder selective influence on lambda signal					
C_MAF_INT_MV_CLC_LAM_AD_INJ	1	0... FFFFH	0... 1023.984375	0.015625	[g]
Time integral threshold for mean value phase end					
C_MAF_INT_MV_CLC_LAM_AD_INJ_2	1	0... FFFFH	0... 1023.984375	0.015625	[g]
Time integral threshold for mean value phase end if LV_LAM_AD_INJ_EXT_ENA = 1					
C_MAF_INT_TRA_PHA_LAM_AD_INJ	1	0... FFFFH	0... 1023.984375	0.015625	[g]
MAF integral threshold for transient phase end					
C_MAF_INT_TRA_PHA_LAM_AD_INJ_2	1	0... FFFFH	0... 1023.984375	0.015625	[g]
MAF integral threshold for transient phase end if LV_LAM_AD_INJ_EXT_ENA = 1					
C_MFF_ADD_LAM_AD_MAX	1	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Maximum limit for adaptation values for lambda adaptation via injection mode					
C_MFF_ADD_LAM_AD_MIN	1	8000... 7FFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
Minimum limit for adaptation values for lambda adaptation via injection mode					
C_T_INT_MV_CLC_LAM_AD_INJ	1	0... FFFFH	0... 1310.7	0.02	[s]
Time integral threshold for mean value phase end					
C_T_INT_MV_CLC_LAM_AD_INJ_2	1	0... FFFFH	0... 1310.7	0.02	[s]
Time integral threshold for mean value phase end if LV_LAM_AD_INJ_EXT_ENA = 1					
C_T_INT_TRA_PHA_LAM_AD_INJ	1	0... FFFFH	0... 1310.7	0.02	[s]
Time integral threshold for transient phase end					
C_T_INT_TRA_PHA_LAM_AD_INJ_2	1	0... FFFFH	0... 1310.7	0.02	[s]
Time integral threshold for transient phase end if LV_LAM_AD_INJ_EXT_ENA = 1					
ID_CRLC_MFF_ADD_COLD_LAM_AD_INJ	4	0... 80H	0... 1	7.8125e-3	[-]

LDP_CTR_NR_COLD_ID_CTR	4	0... FFH	0... 255	1	[-]
Correlation constant for filtered cold adaptation value					
ID_CRLC_MFF_ADD_HOT_LAM_AD_INJ	4	0... 80H	0... 1	7.8125e-3	[-]
LDP_CTR_NR_HOT_ID_CTR	4	0... FFH	0... 255	1	[-]
Correlation constant for filtered hot adaptation value					
ID_MFF_DELTA_MIS_LAM_AD_INJ	NC_NR_MIS_REP_LAM_AD_INJ	0... FFFFH	-694.510597391 ...694.489402609	0.0211948	[mg/stk]
LDP_CTR_MIS_ID_MFF_LAM_AD_INJ	NC_NR_MIS_REP_LAM_AD_INJ	0... FFH	0... 255	1	[-]
Additional fuel mass in case of misfire for lambda adapation via injection mode					
IP_MFF_ADD_LAM_AD_INJ	8	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDP_MFF_IP_MFF_ADD_LAM_AD_INJ	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Normalization of the adaptation value to the reference fuel mass					
LC_MFF_ADD_LAM_AD_INJ_CLR	1	0... 1H	0... 1	1	[-]
Switch to clear adaptation values of lambda adaptation via injection mode					
LC_MFF_CLC_ADD_LAM_COLD_ADD	1	0... 1H	0... 1	1	[-]
Activation of MFF correction calculation by using the hot adapt. value for the whole range and the cold adapt. value as offset for the cold area					

Import Actions:

ACTION_MFMA_GetEnableCondition(**OUT** <PRM_LV_LAM_AD_INJ_EXT_ENA>)

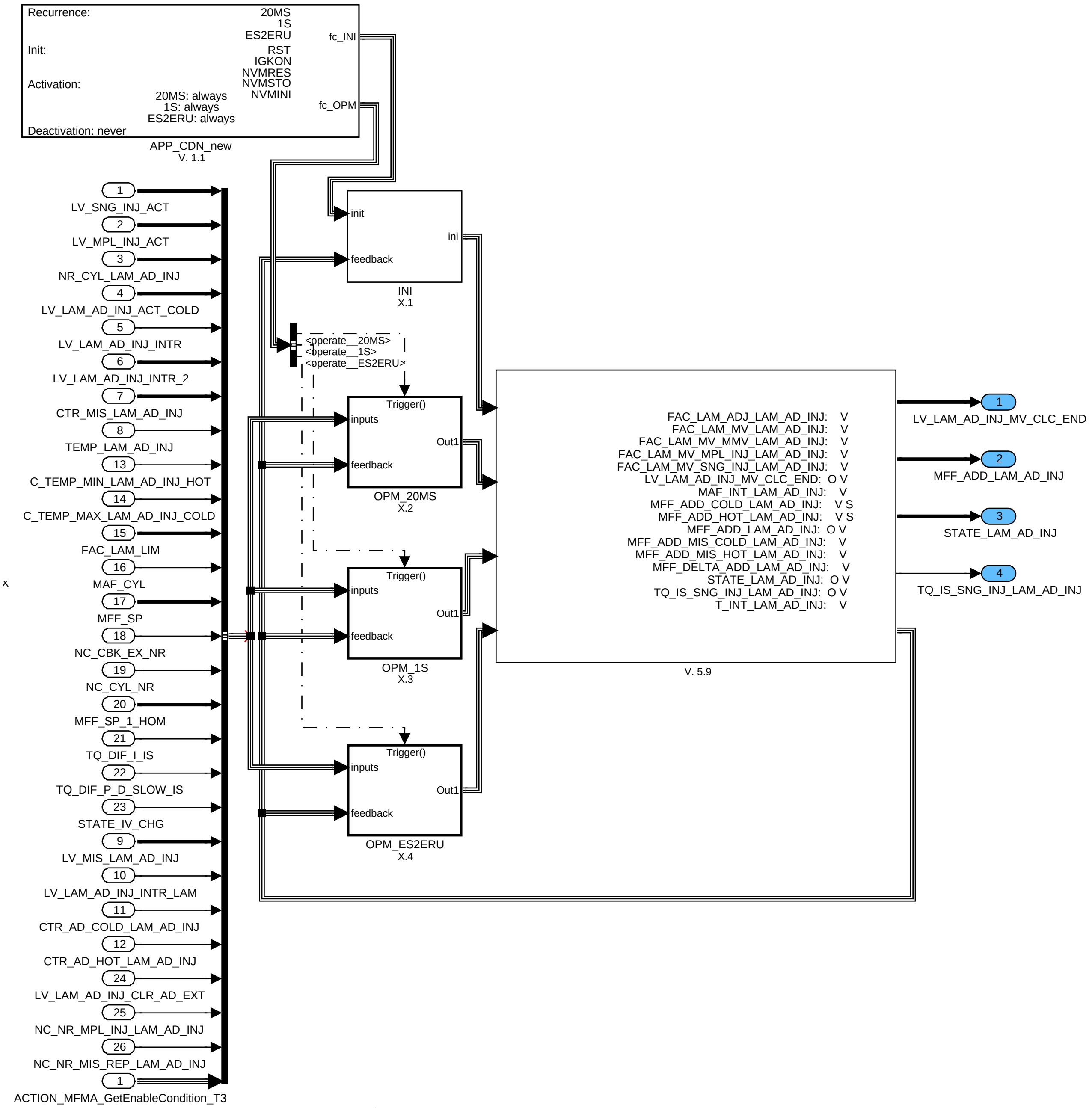
General Information

The adaptation of the additional fuel mass at minimum fuel mass injection is divided in a time slice with single injection and with multiple injections. At the beginning of each phase a certain time is waited in a transient phase for system stabilization. In a second phase the lambda mean value is calculated. At the end of the phase with multiple injections, the additional fuel at minimum injection is calculated from the lambda controller output difference of the single and multiple injection phase.

Application Conditions

Initialization: RST, IGKON, NVMRES, NVMSTO, NVMINI
Activation: 20MS: always
1S: always
ES2ERU: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.0.4 30−Oct−2006

Figure 64:
Path: MFMA_ISPCLAIM0

49.5.1 Initialization and NVMY management

49.5.1.1 Initialization at RST and IGKON

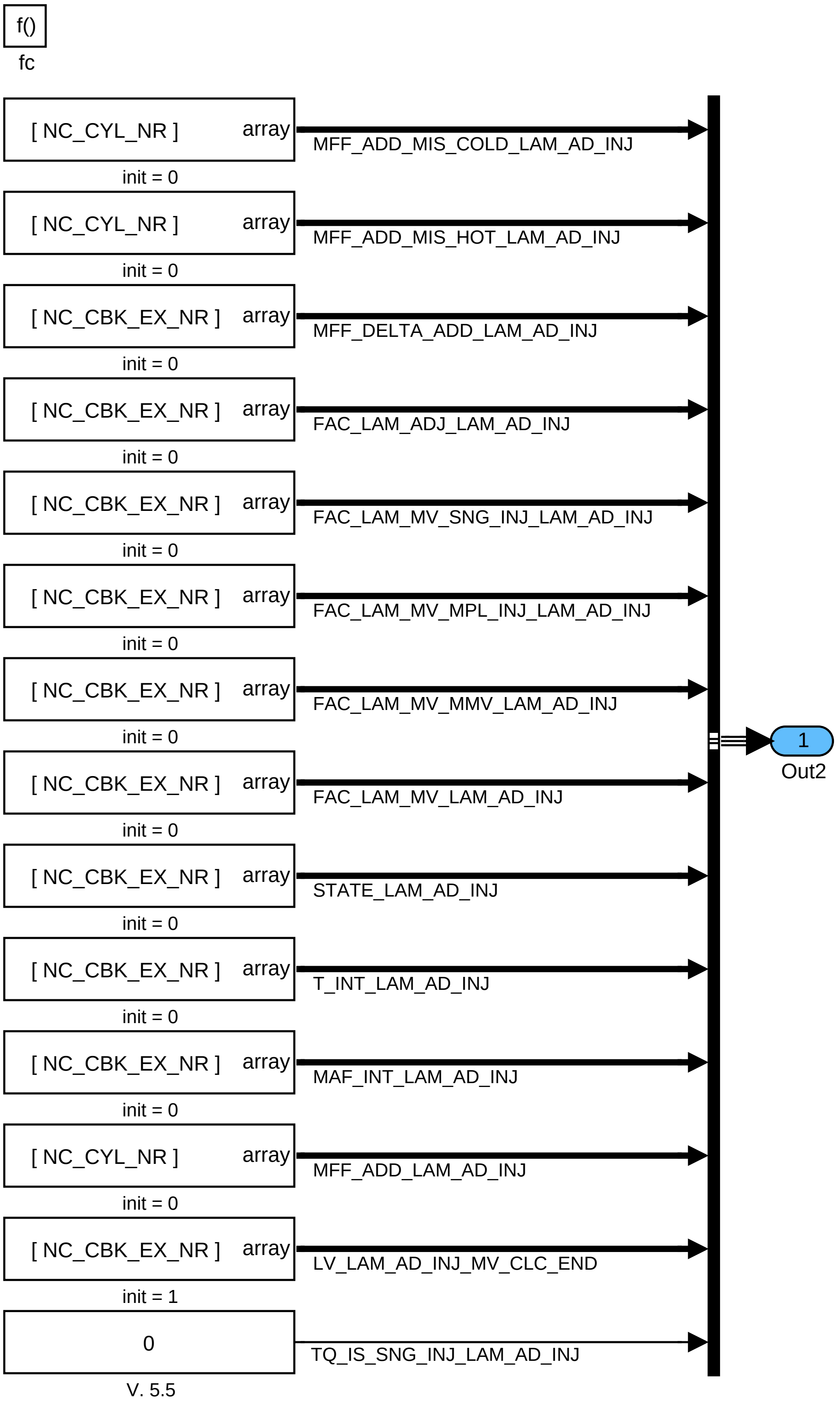


Figure 65:
Path: MFMA_ISPCLAIM0/INI/RST_IGKON

49.5.1.2 Initialization and reading of NVMY values

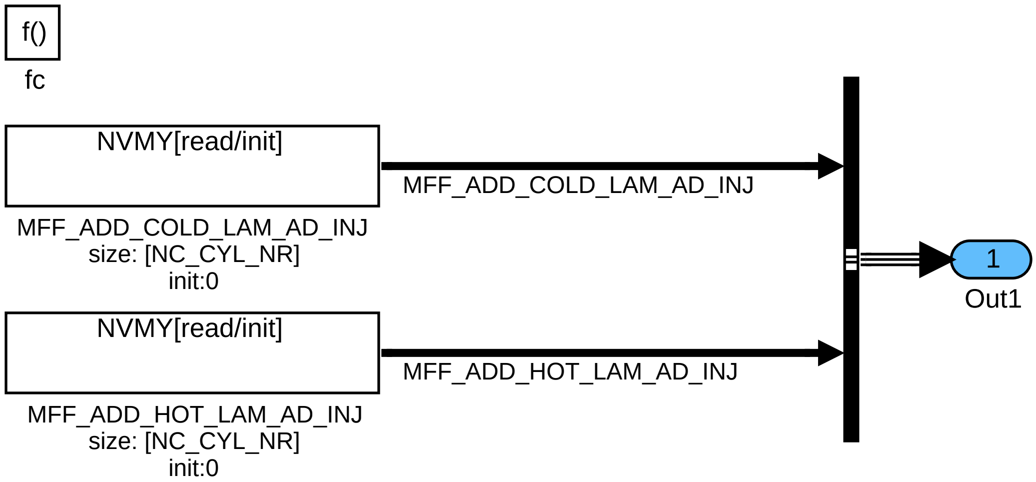


Figure 66:
Path: MFMA_ISPCLAIM0/INI/NVMINI_NVMRES

49.5.1.3 Writing of the NVMY values

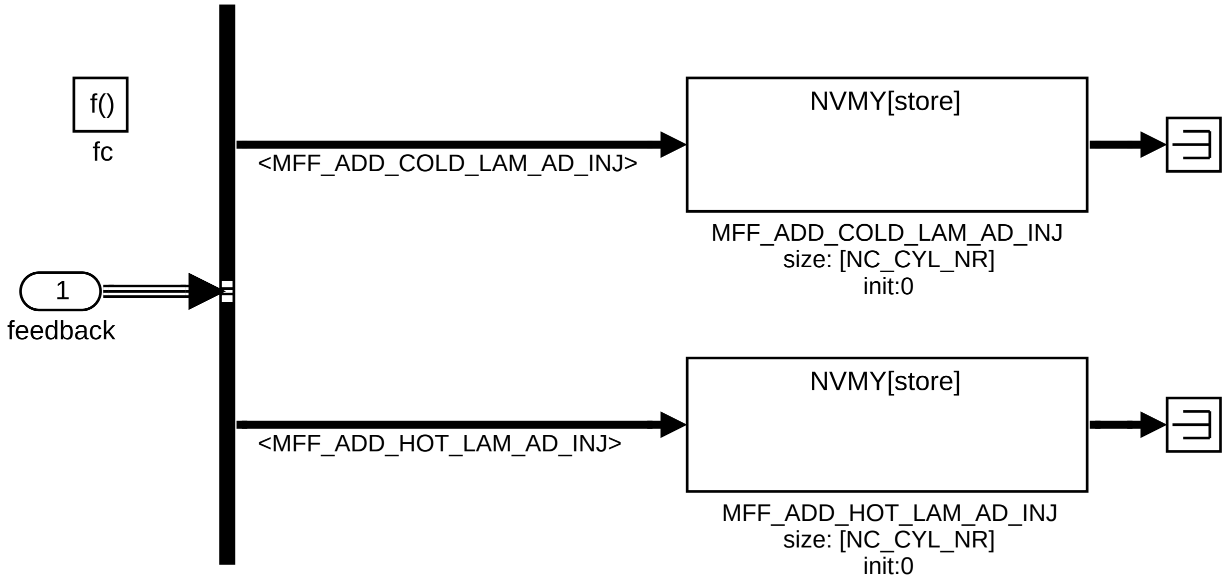


Figure 67:
Path: MFMA_ISPCLAIM0/INI/NVMSTO

49.5.2 20ms operations

All calculations are done for each exhaust cylinder bank, where the bank specific values are denoted within the graphical model with the suffix _i, in the textual description they are referenced via [i].

49.5.2.1 State machine and general calculations

Calculation of state transitions, state actions and general calculations.

49.5.2.1.1 State transitions

49.5.2.1.1.1 State selection

49.5.2.1.1.1.1 NOT_ACT

If the next adaptation cycle starts (either with single or multiple injections), the state is left towards TRA_PHA.

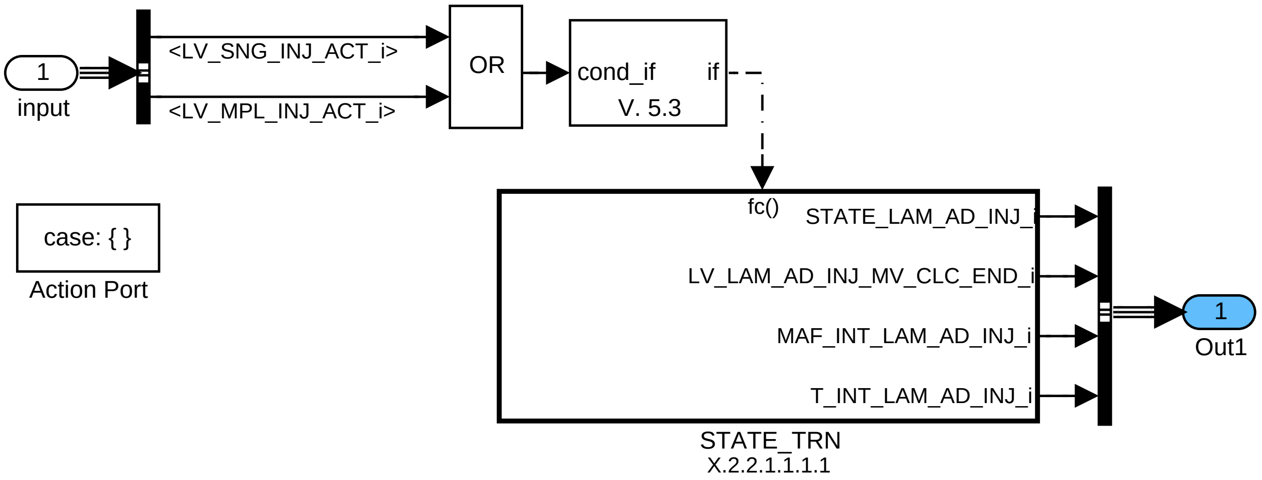


Figure 68:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/NOT_ACT

49.5.2.1.1.1.1.1 Transition to TRA_PHA

On the transition to TRA_PHA the MAF and time integrals are set to 0 and furthermore the end flag used in the manager module is reset.

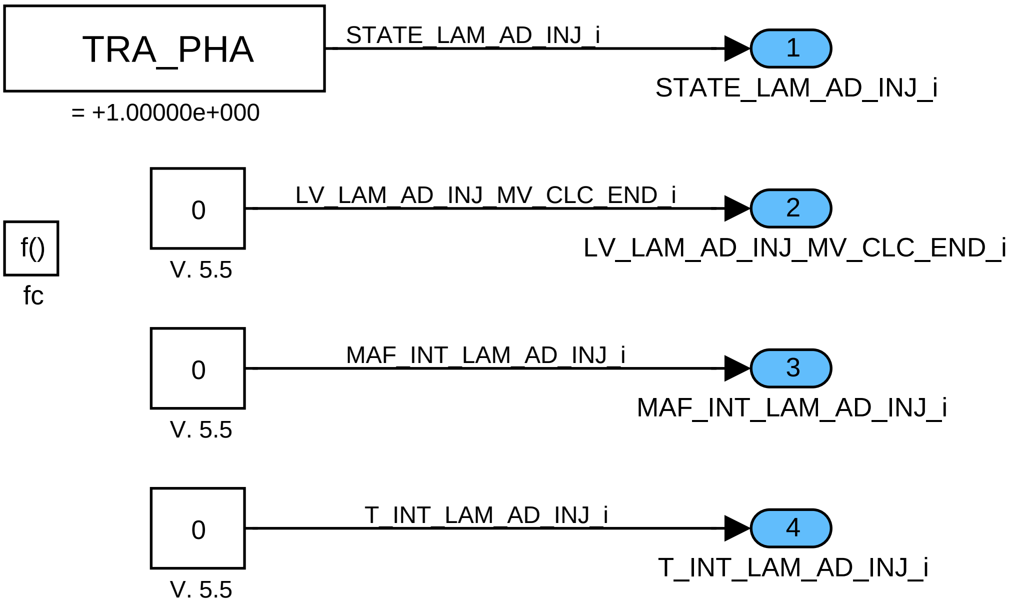


Figure 69:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/NOT_ACT/STATE_TRN

49.5.2.1.1.1.2 TRA_PHA

The state TRA_PHA is usually left towards MV_CLC, but if the adaptation is interrupted, the state will change back to NOT_ACT.

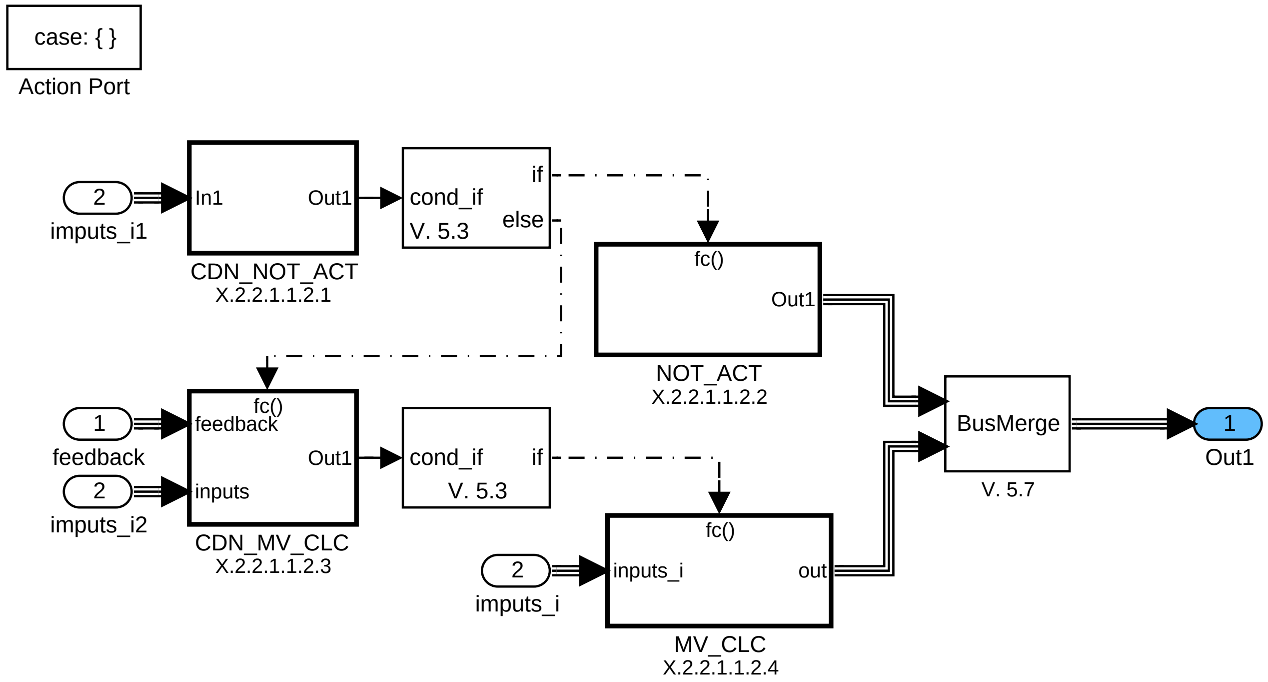


Figure 70:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/TRA_PHA

49.5.2.1.1.1.2.1 Conditions for NOT_ACT

If none of the flags LV_SNG/MPL_INJ_ACT is set, the state is left towards NOT_ACT.

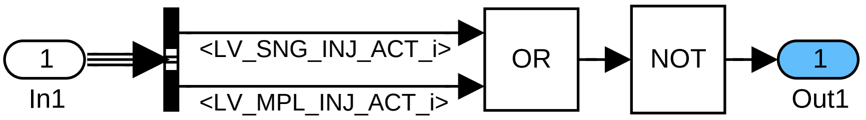


Figure 71:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/TRA_PHA/CDN_NOT_ACT

49.5.2.1.1.1.2.2 Transition to NOT_ACT

Figure 72:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/TRA_PHA/NOT_ACT

49.5.2.1.1.1.2.3 Conditions for MV_CLC

If the MAF and time integrals have reached a threshold and if the lambda controller output has stabilized, the state is left towards MV_CLC. Furthermore this transition is done if misfire is detected or the lambda controller had reached its limitations.

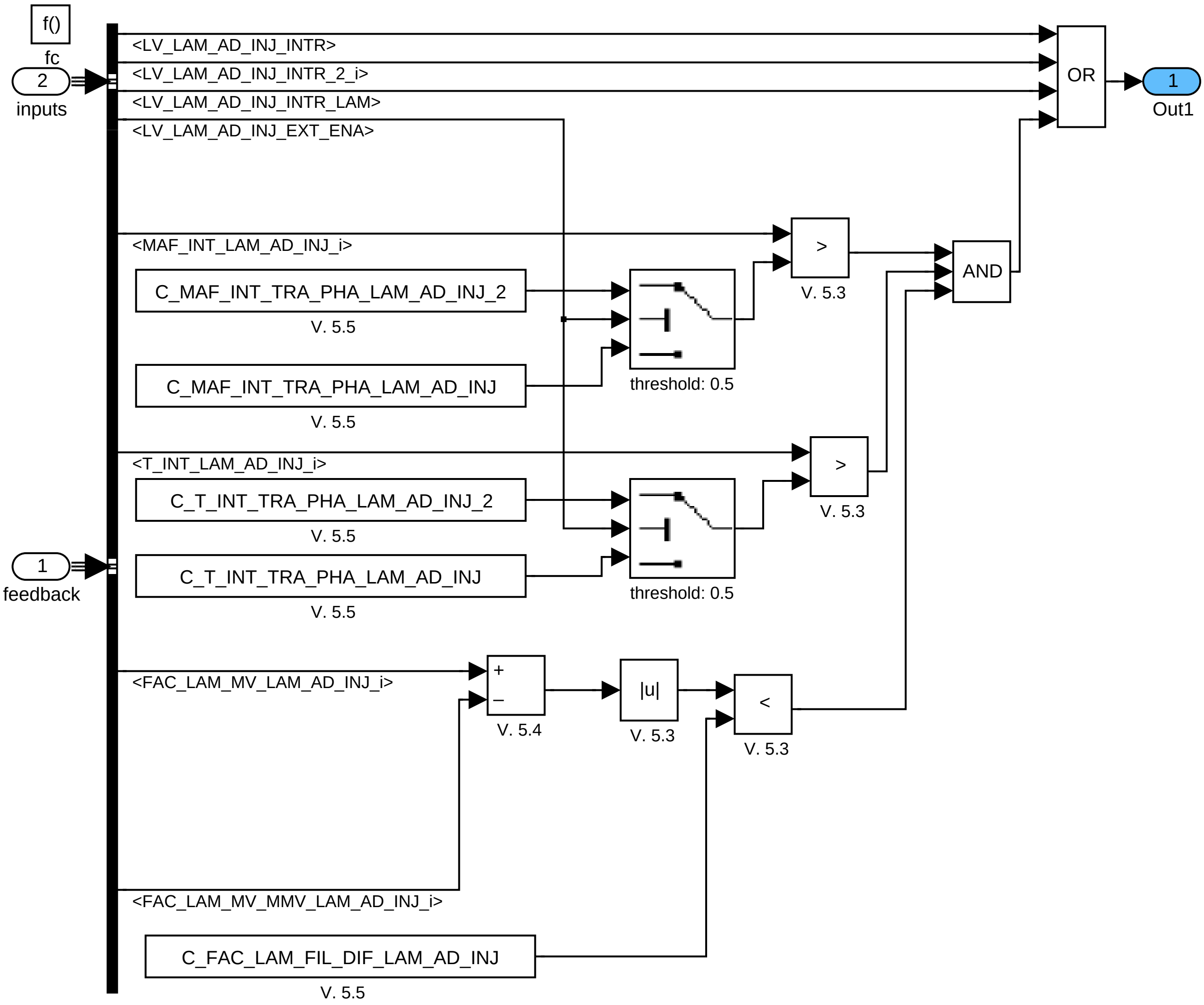


Figure 73:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/TRA_PHA/CDN_MV_CLC

49.5.2.1.1.1.2.4 Transition to MV_CLC

On transition to MV_CLC the MAF and time integrals are reset.

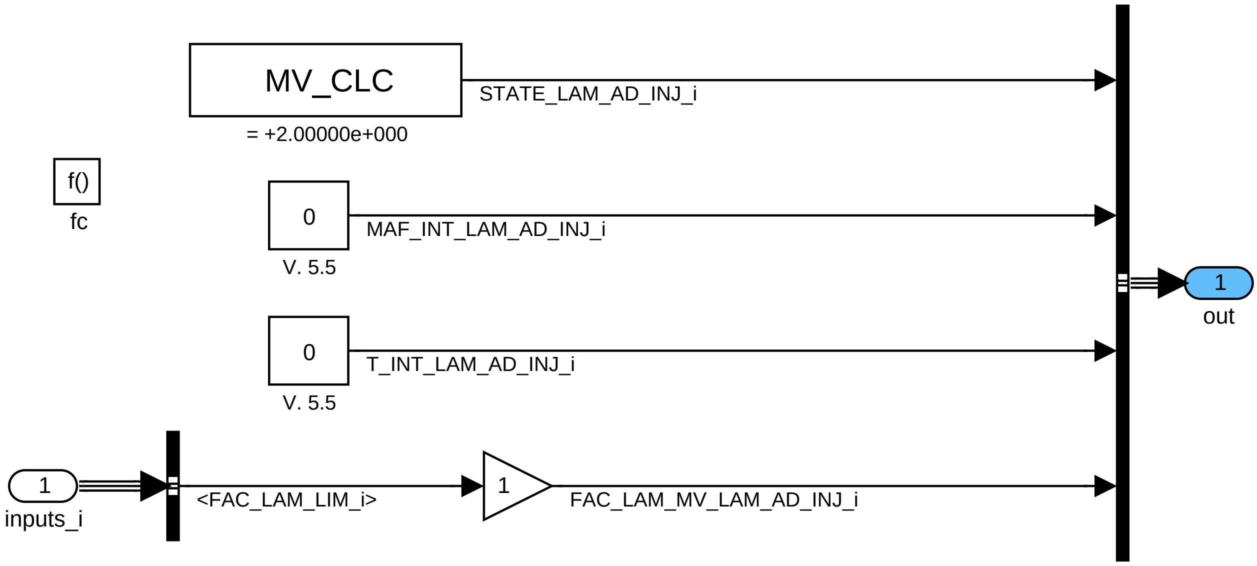


Figure 74:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/TRA_PHA/MV_CLC

49.5.2.1.1.1.3 MV_CLC

The state MV_CLC is usually left towards NOT_ACT after a single injection adaptation phase or due to interruption, and towards LAM_AD after a multiple injection adaptation phase.

49.5.2.1.1.1.3.1 Interruption of the mean-value calculation

If neither single nor multiple injection adaptation is allowed anymore, the state is left towards NOT_ACT.

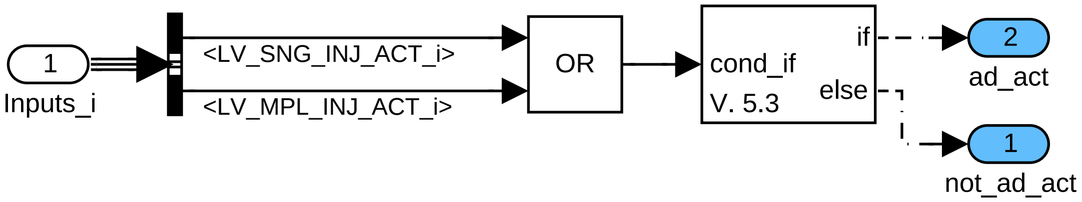


Figure 75:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/MV_CLC/CDN_AD_ACT

49.5.2.1.1.1.3.2 Conditions for normal end of MV_CLC

If the MAF and time integrals reach a threshold, the state is left to NOT_ACT after single injections and to LAM_AD after multiple injections. If misfire is detected or the lambda controller had reached its limitations, the state is also left immediately.

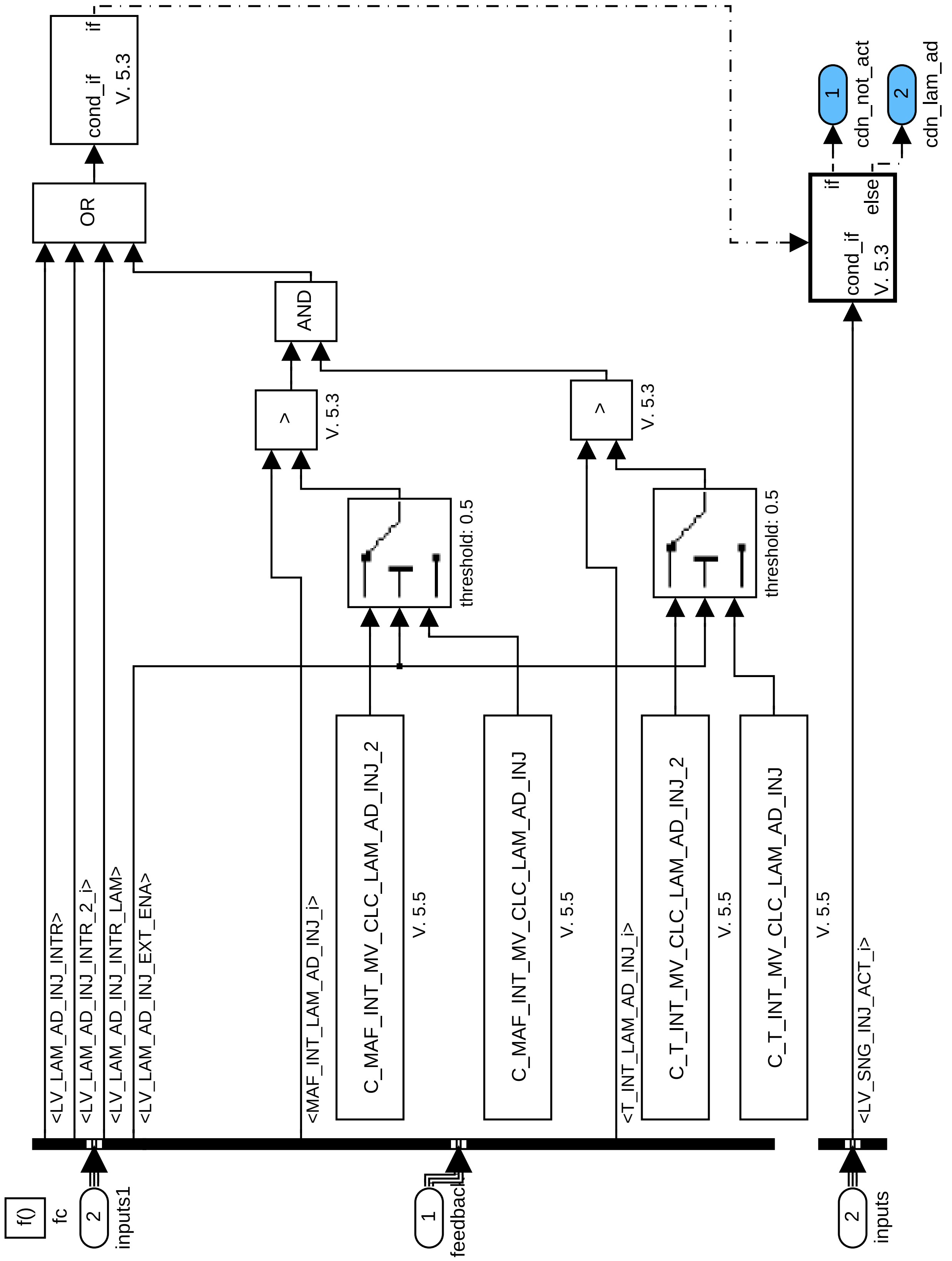


Figure 76:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/MV_CLC/CDN_INT_MAX

49.5.2.1.1.1.3.3 Transition to NOT_ACT due to interruption

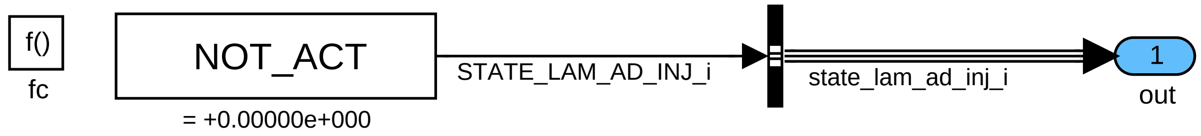


Figure 77:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/MV_CLC/NOT_ACT_1

49.5.2.1.1.1.3.4 Transition to NOT_ACT due to normal ending

49.5.2.1.1.1.3.4.1 Memorization of lambda and engine speed controller outputs

The lambda and engine speed controller outputs are memorized for comparison after/in the multiple injection adaptation phases.

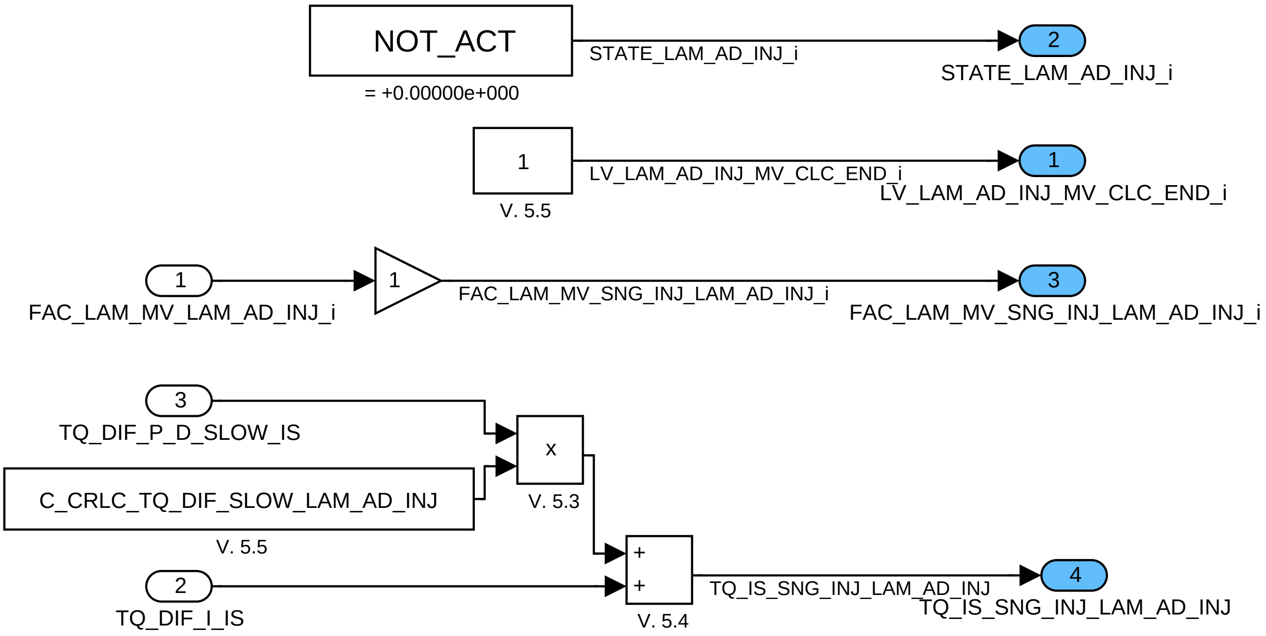


Figure 78:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/MV_CLC/NOT_ACT_2/CLC

49.5.2.1.1.1.3.5 Transition to LAM_AD

49.5.2.1.1.1.3.5.1 Calculation of FAC_LAM_MV_MPL_INJ_LAM_AD_INJ[i]

The lambda controller output (mean value) of the multiple injection phase is stored.

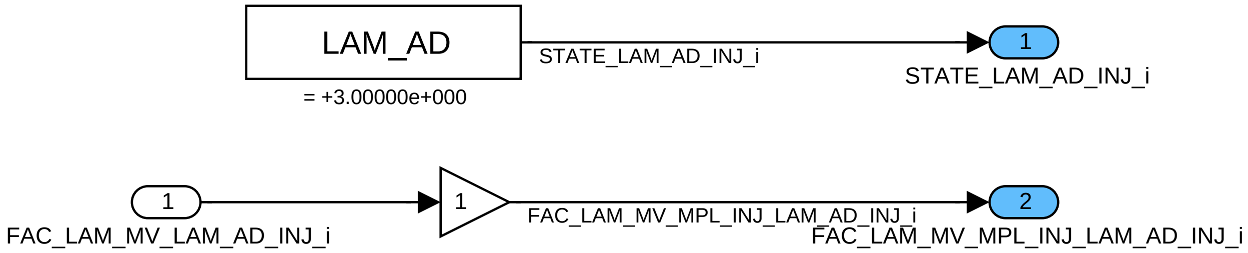


Figure 79:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/MV_CLC/LAM_AD/CLC

49.5.2.1.1.4 LAM_AD

The state LAM_AD is left immediately towards NO_ACT after the adaptation value has be calculated.

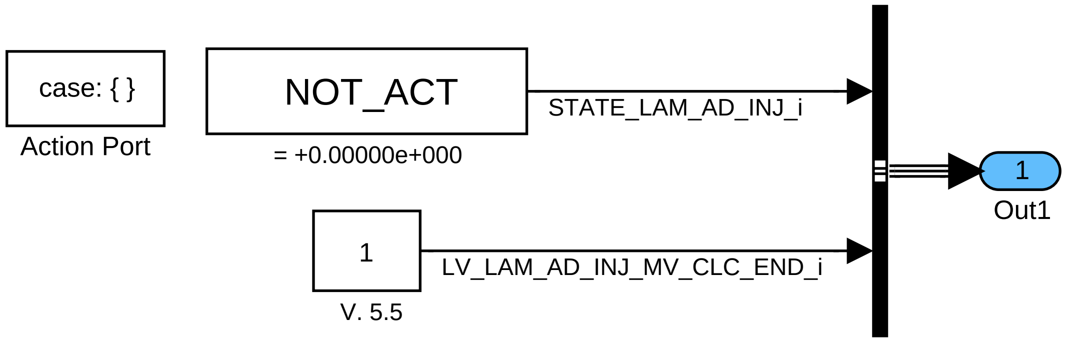


Figure 80:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_TRANSITION/CLC/LAM_AD

49.5.2.1.2 State actions

49.5.2.1.2.1 State selection

49.5.2.1.2.1.1 TRA_PHA

49.5.2.1.2.1.1.1 MAF and time integral calculation

During the state TRA_PHA the MAF and time integrals are calculated.

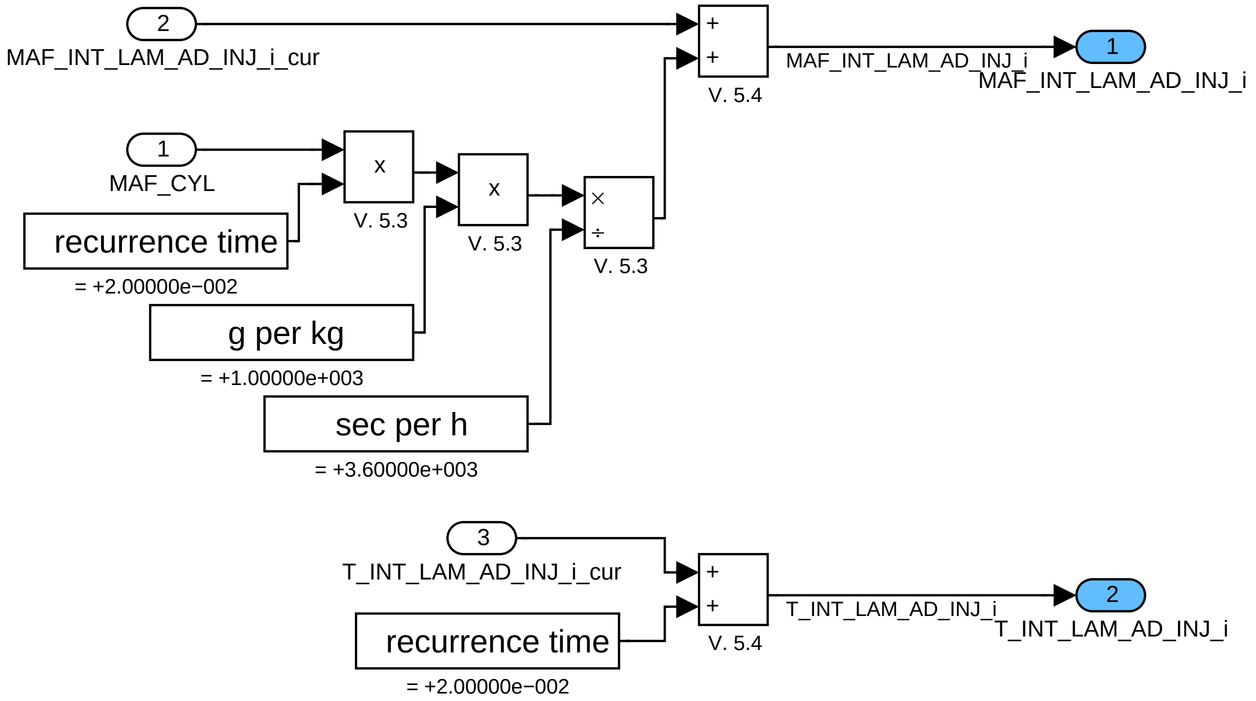


Figure 81:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/TRA_PHA/CLC

49.5.2.1.2.1.2 MV_CLC

49.5.2.1.2.1.2.1 MAF and time integral calculation

During the state MV_CLC the MAF and time integrals are calculated.

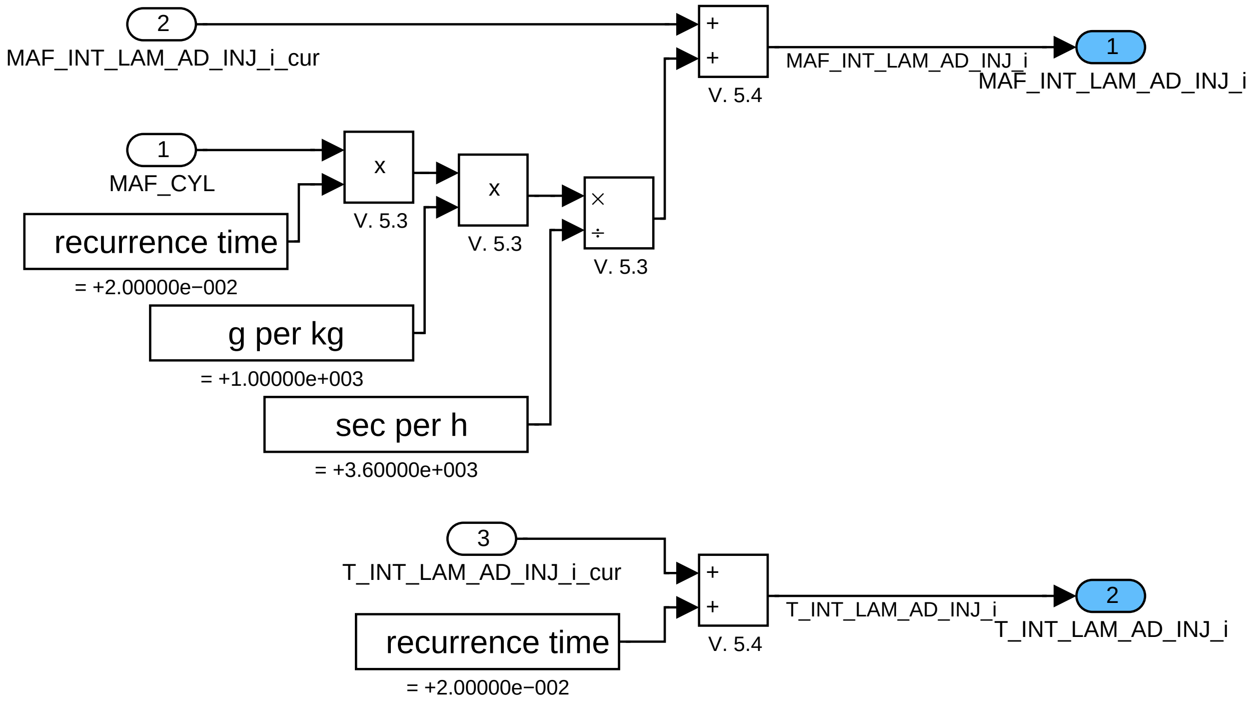


Figure 82:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/MV_CLC/CLC

49.5.2.1.2.1.3 LAM_AD

During the state LAM_AD the lambda controller output difference and the corresponding fuel mass caused by multiple injections are calculated and then stored in the adaptation values of the active temperature range.

49.5.2.1.2.1.3.1 Lambda controller difference and fuel mass calculation

49.5.2.1.2.1.3.1.1 Calculation of FAC_LAM and MFF_DELTA

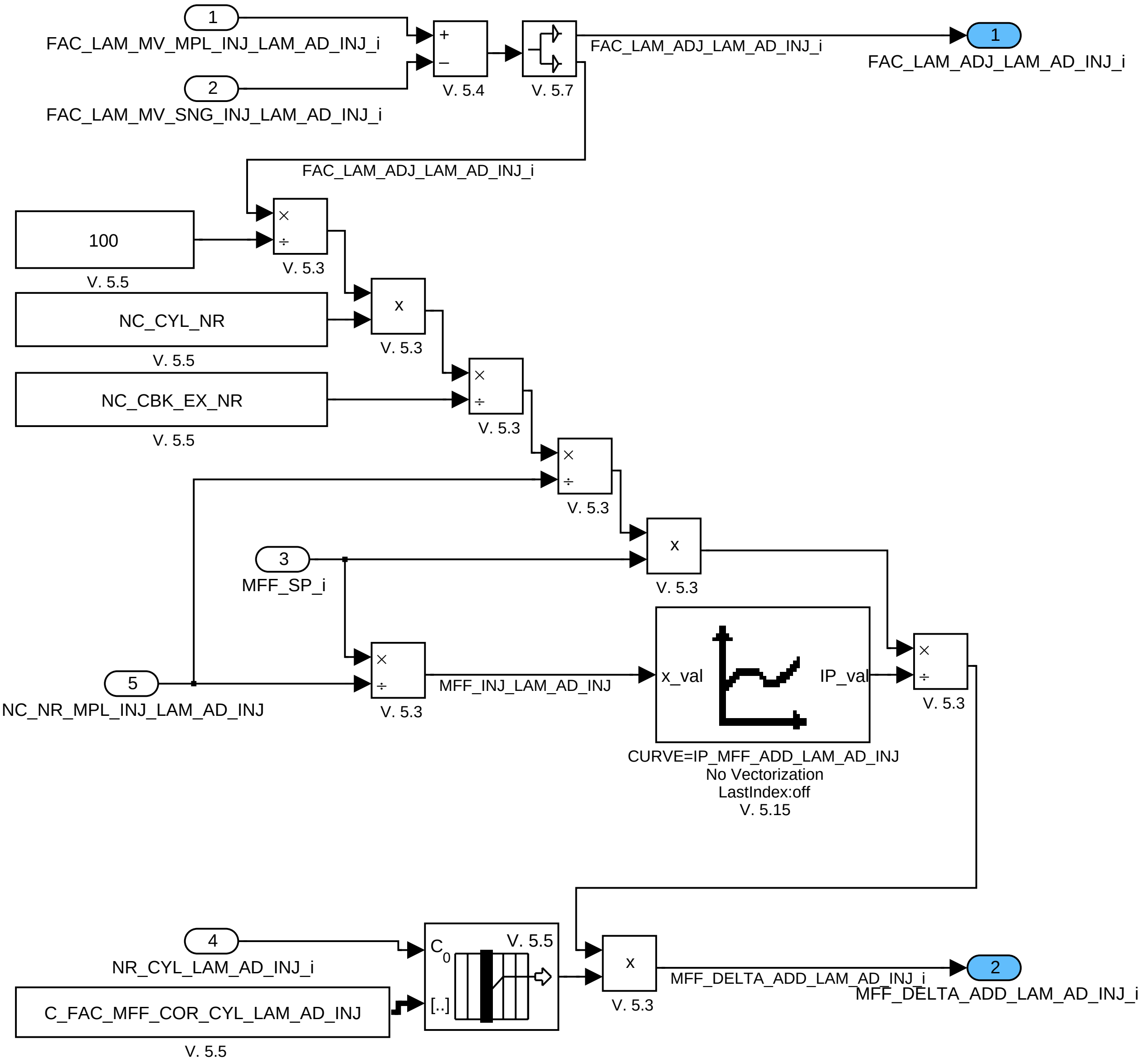


Figure 83:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/LAM_AD/CLC_MFF_DELTA/CLC

49.5.2.1.2.1.3.2 Lower temperature range

If the lower temperature range is active, the adaptation values are written to the corresponding variables.

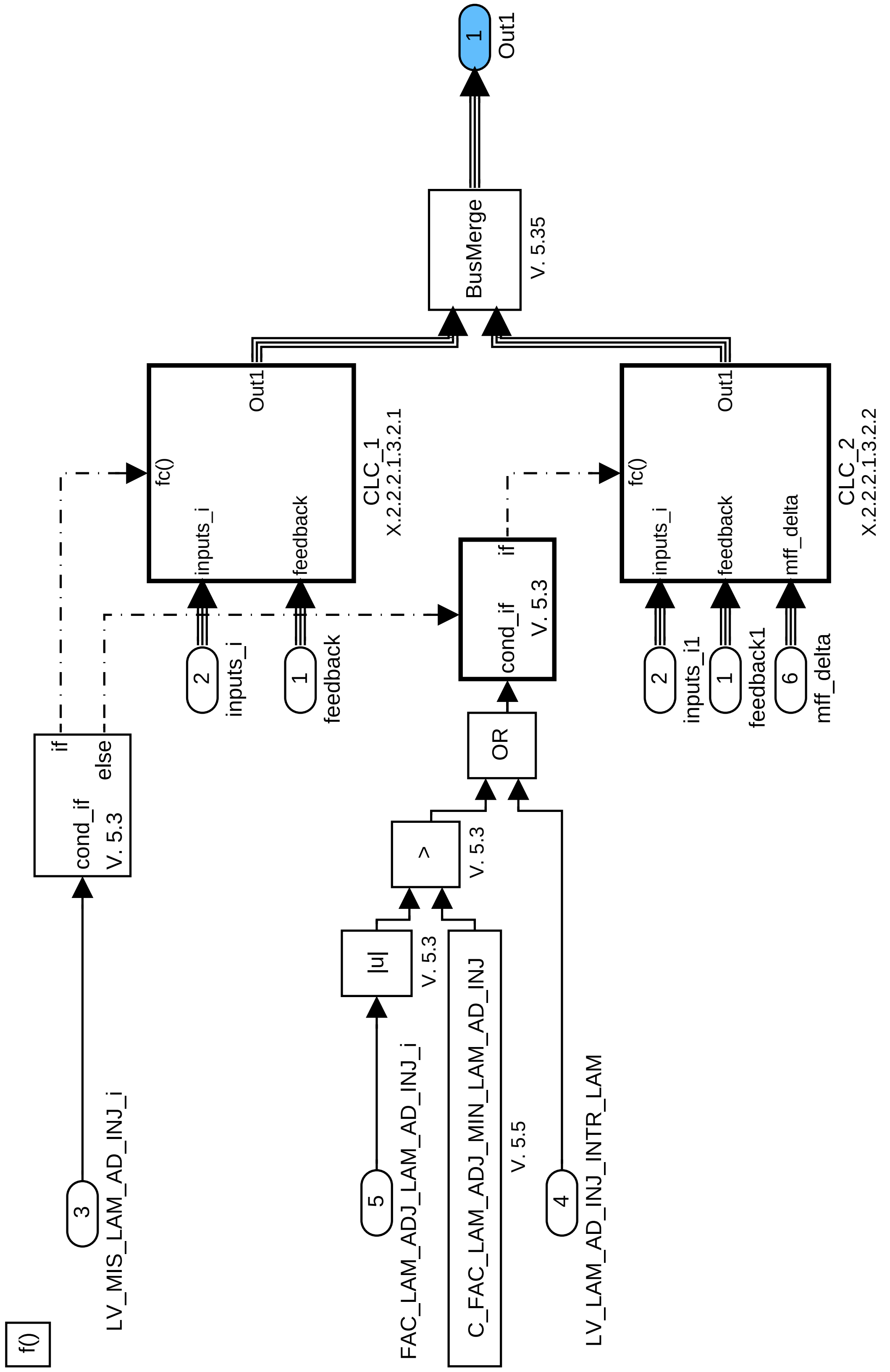


Figure 84:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/LAM_AD/CLC_COLD

49.5.2.1.2.1.3.2.1 Correction of adaptation values in case of misfire

49.5.2.1.2.1.3.2.1.1 Determination of corrective fuel mass for misfire

In case misfire was detected on the active cylinder during multiple injections, an additional fuel mass is determined from the map ID_MFF_DELTA_MIS_LAM_AD_INJ.

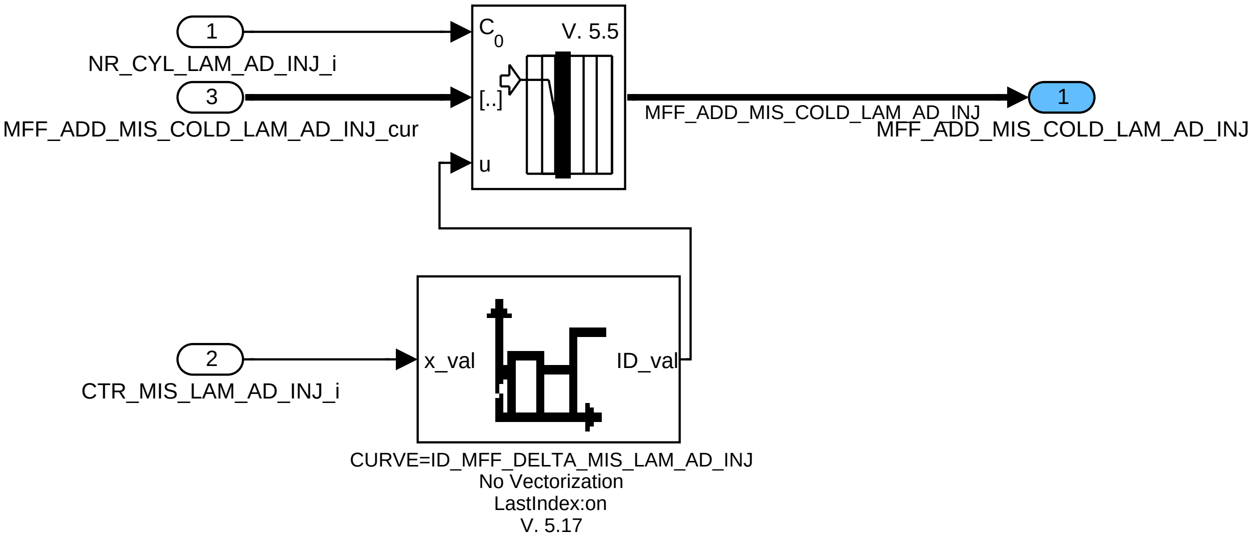


Figure 85:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/LAM_AD/CLC_COLD/CLC_1/CLC

49.5.2.1.2.1.3.2.2 Calculation of the adaptation value

49.5.2.1.2.1.3.2.2.1 Calculation of MFF_ADD_COLD_LAM_AD_INJ

The adaptation value is determined from its old value and the actual calculated difference using the correlation factor C_CRLC_MFF_ADD_COLD_LAM_AD_INJ. Afterwards the additional misfire fuel mass is reset to 0.

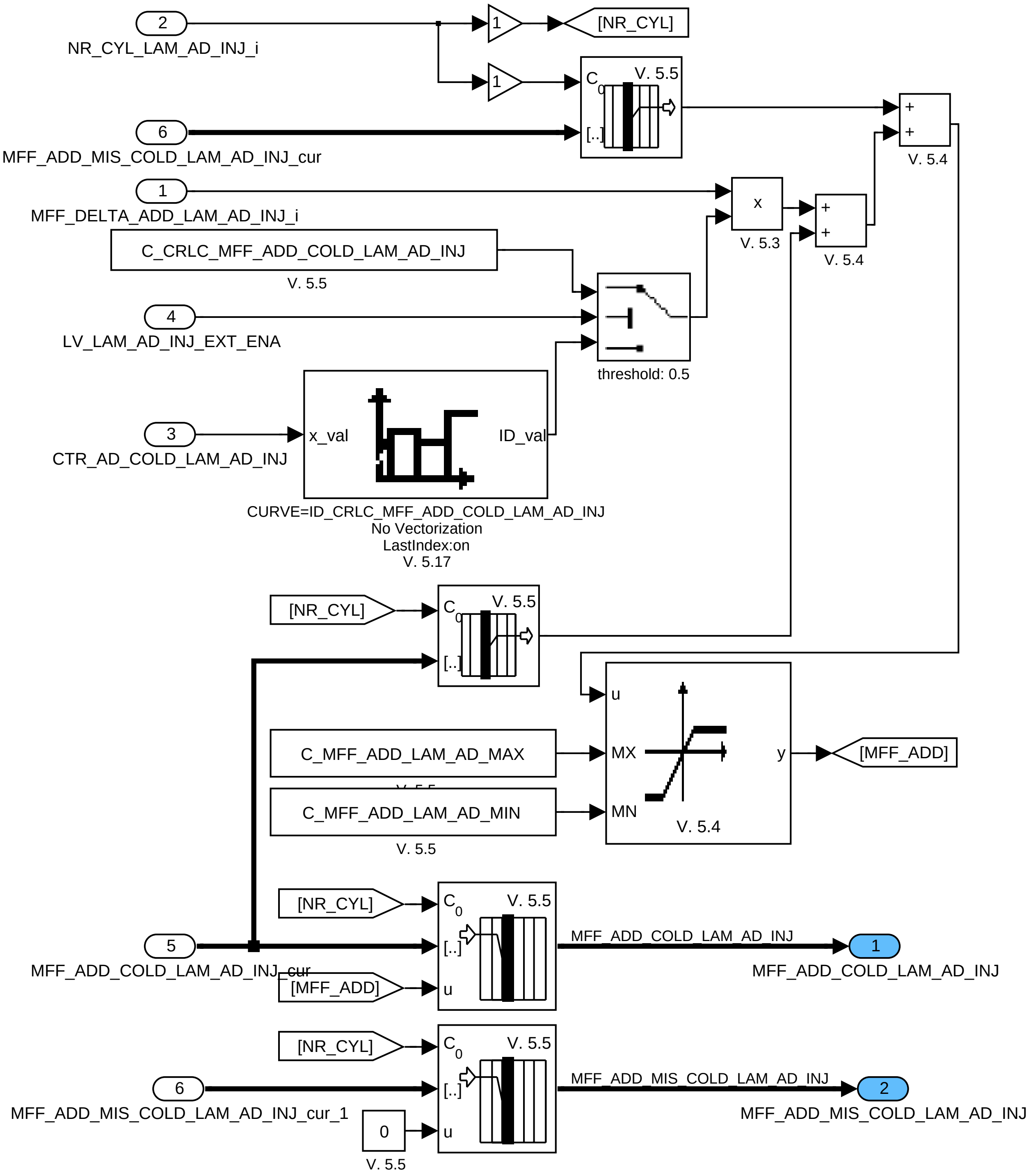


Figure 86:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/LAM_AD/CLC_COLD/CLC_2/CLC

49.5.2.1.2.1.3.3 Upper temperature range

If the upper temperature range is active, the adaptation values are written to the corresponding variables.

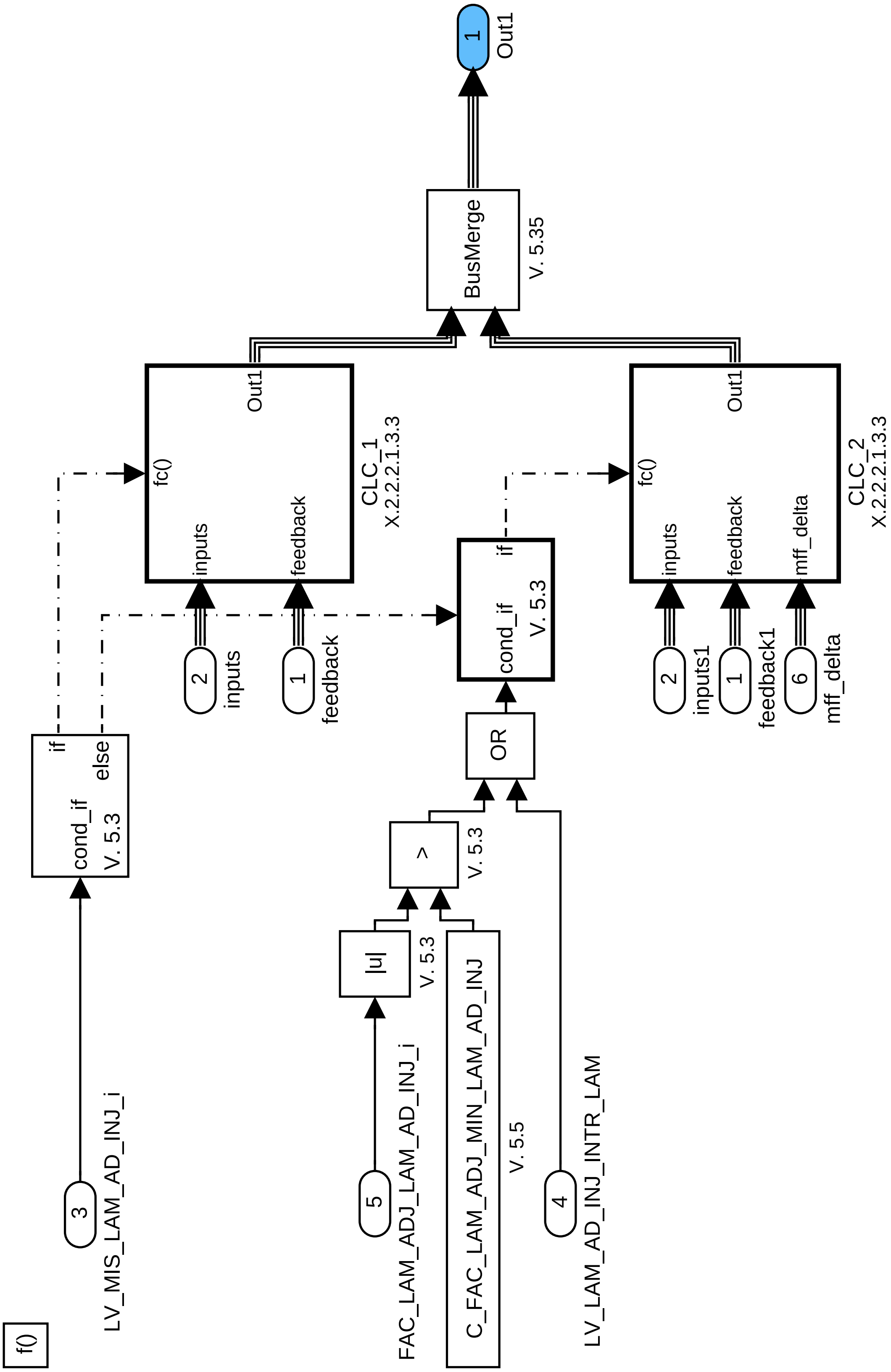


Figure 87:

Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/LAM_AD/CLC_HOT

49.5.2.1.2.1.3.3.1 Correction of adaptation values in case of misfire

49.5.2.1.2.1.3.3.1.1 Determination of corrective fuel mass for misfire

In case misfire was detected on the active cylinder during multiple injections, an additional fuel mass is determined from the map ID_MFF_DELTA_MIS_LAM_AD_INJ.

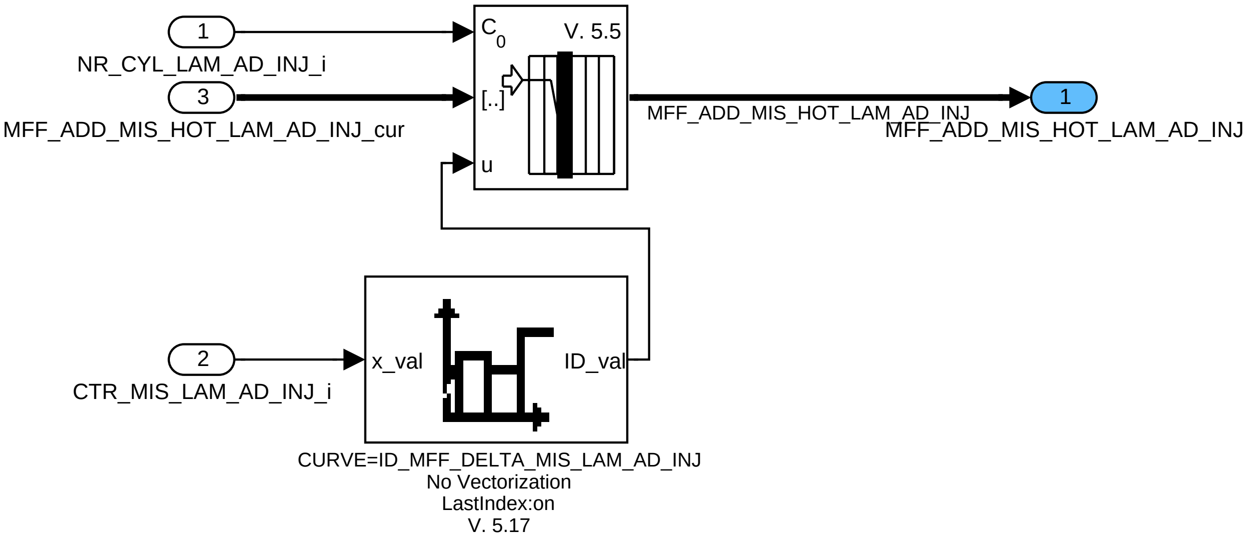


Figure 88:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/LAM_AD/CLC_HOT/CLC_1/CLC

49.5.2.1.2.1.3.3.2 Calculation of the adaptation value

49.5.2.1.2.1.3.3.2.1 Calculation of MFF_ADD_HOT_LAM_AD_INJ

The adaptation value is determined from its old value and the actual calculated difference using the correlation factor C_CRLC_MFF_ADD_HOT_LAM_AD_INJ. Afterwards the additional misfire fuel mass is reset to 0.

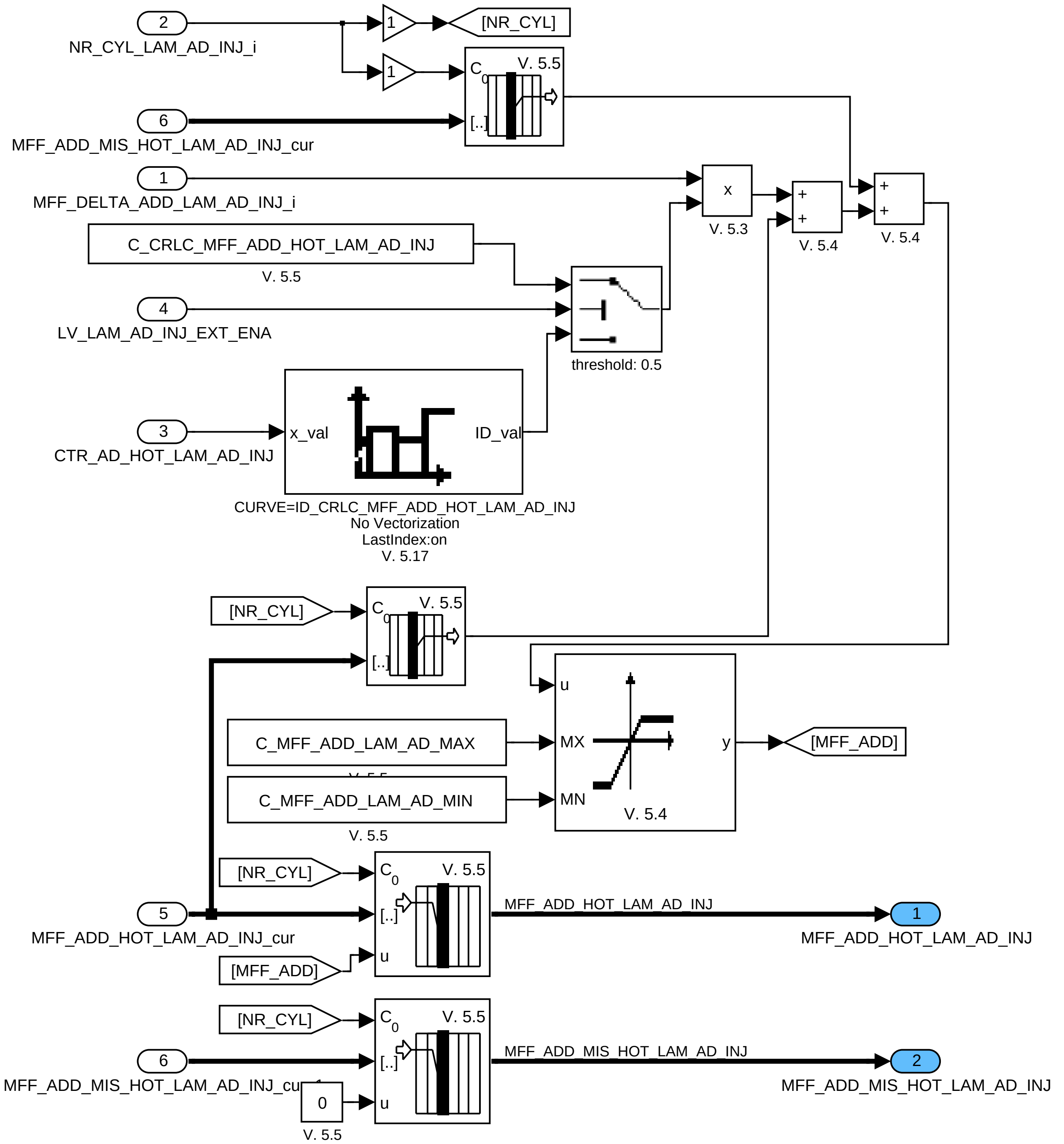


Figure 89:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/STATE_ACTION/CLC/LAM_AD/CLC_HOT/CLC_2/CLC

49.5.2.1.3 General calculations

The mean value calculations are done for the current active bank(s).

49.5.2.1.3.1 Mean value calculation

The mean value calculations are done for the current active bank(s).

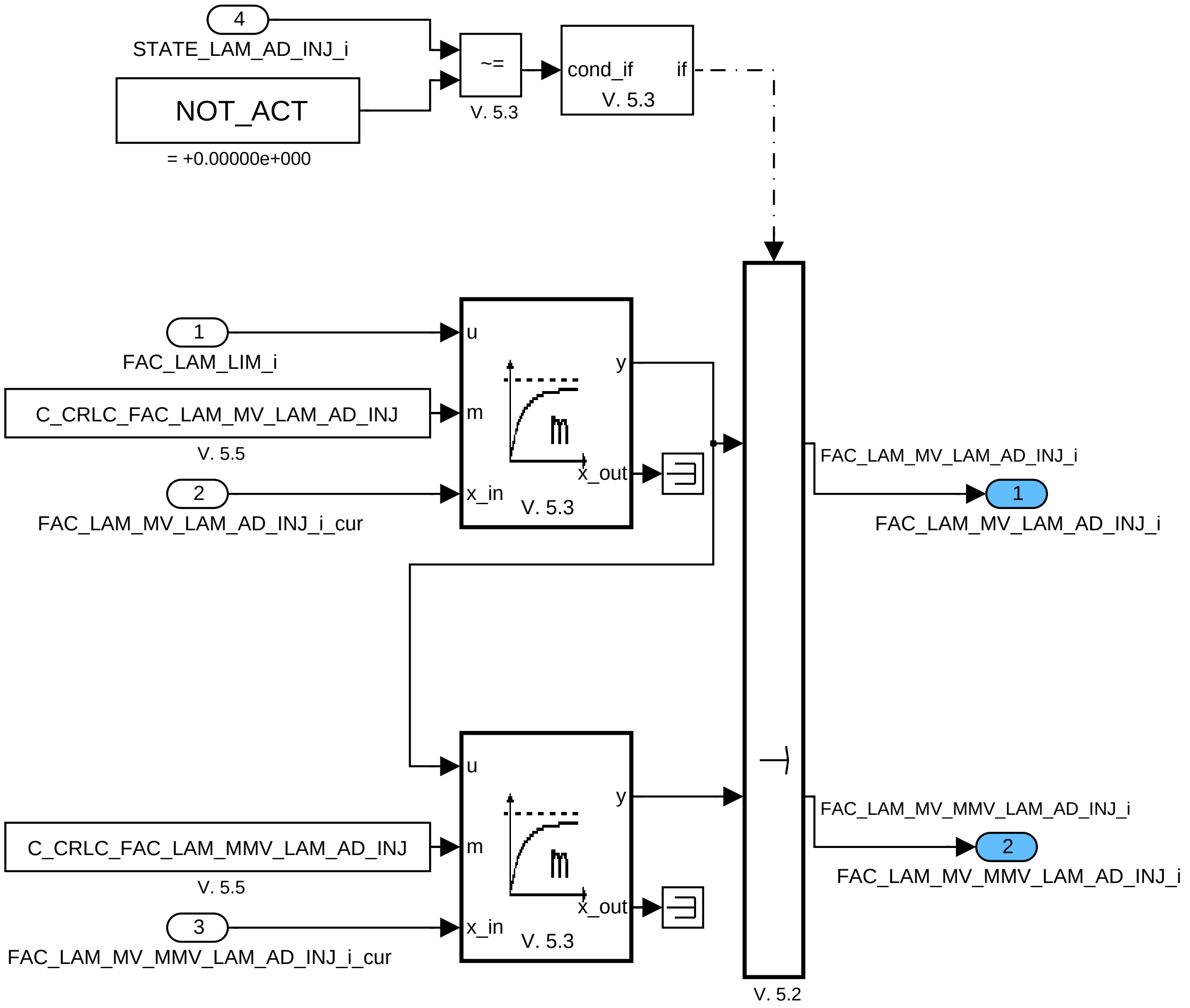


Figure 90:
Path: MFMA_ISPCLAIM0/OPM_20MS/CLC/GEN_CALC/CLC

49.5.3 1s operations

All calculations are done for each cylinder, where the cylinder specific values are denoted within the graphical model with the suffix _x, in the textual description they are referenced via [x].

49.5.3.1 Calculation of the adaptation value

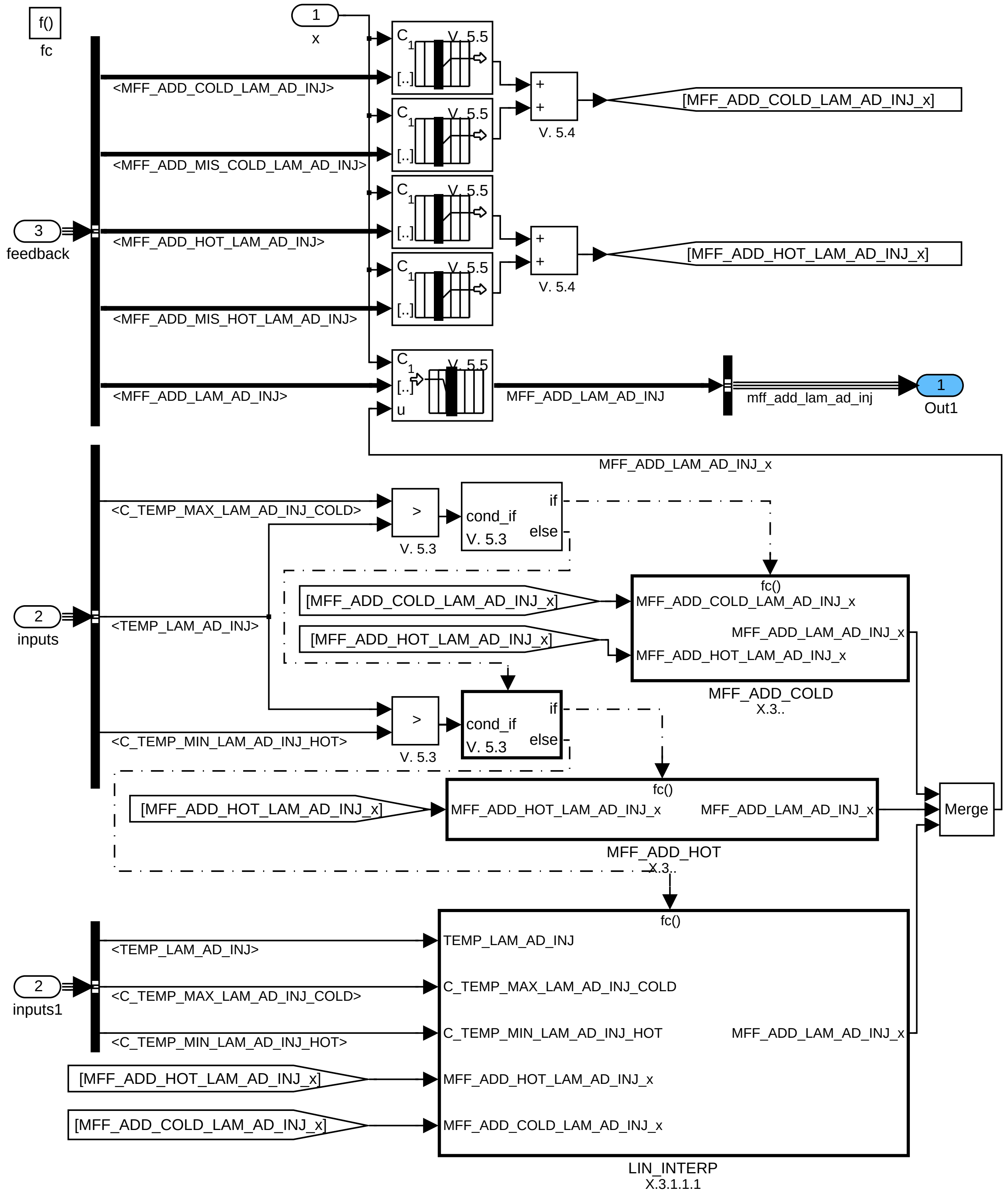


Figure 91:
Path: MFMA_ISPCLAIM0/OPM_1S/CLC_AD_VALUE

49.5.3.1.1 Linear interpolation

Depending on the calibration data LC_MFF_CLC_ADD_LAM_COLD_ADD two different ways of calculation for the adaptation value can be chosen. For setting the calibration to zero here is a linear interpolation between the upper and lower temperature range. For one the cold adaptation value is added as an offset to the hot adaptation value. The offset is faded out between the upper temperature threshold for the cold area and the lower threshold for the hot area.

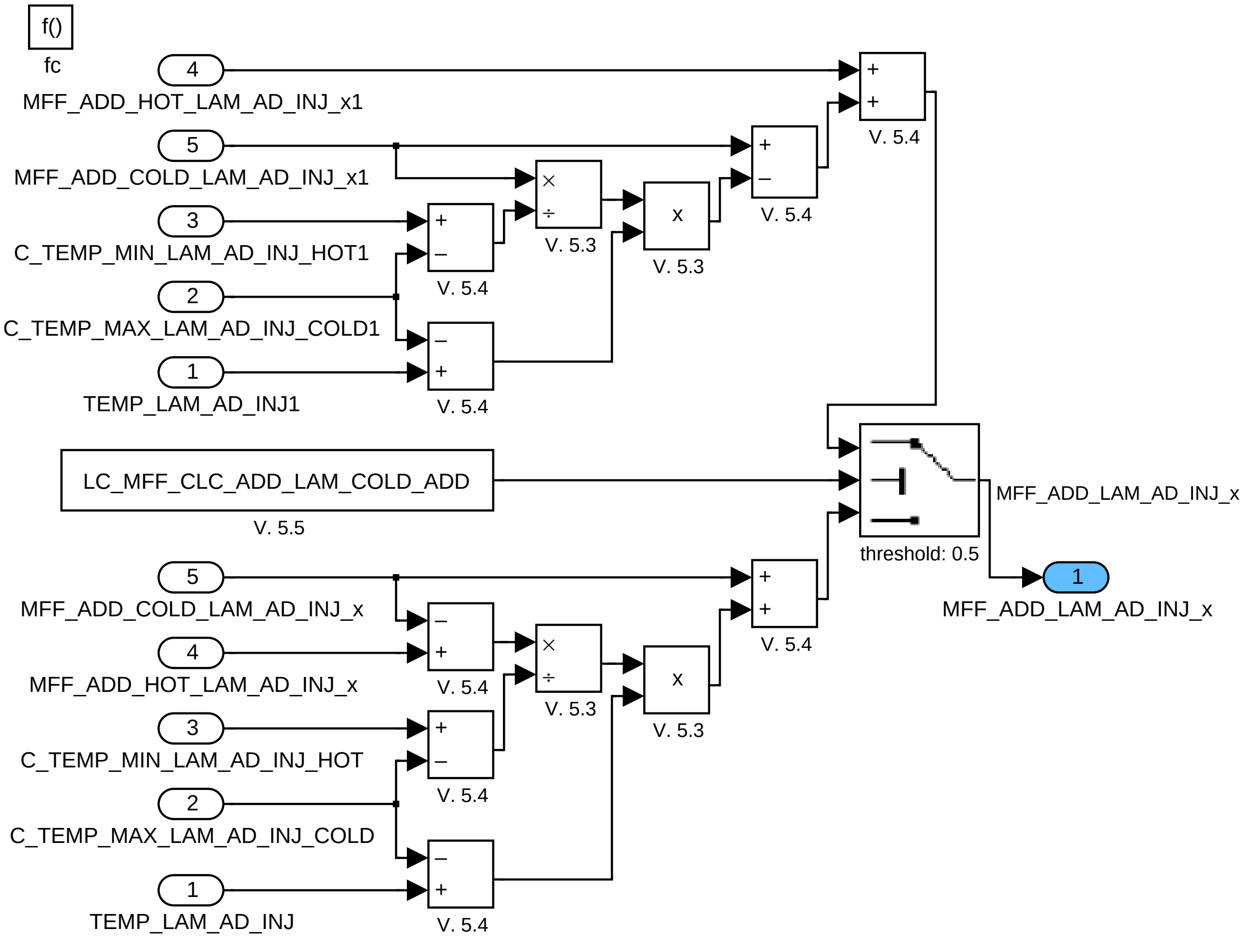


Figure 92:
Path: MFMA_ISPCLAIM0/OPM_1S/CLC_AD_VALUE/LIN_INTERP

49.5.3.1.2 Cold adaptation value

Depending on the calibration value LC_MFF_CLC_ADD_LAM_COLD_ADD the adaptation value for the cold range can be used alone or the adaptation of the cold range can be used as an offset for the hot range adaptation value.

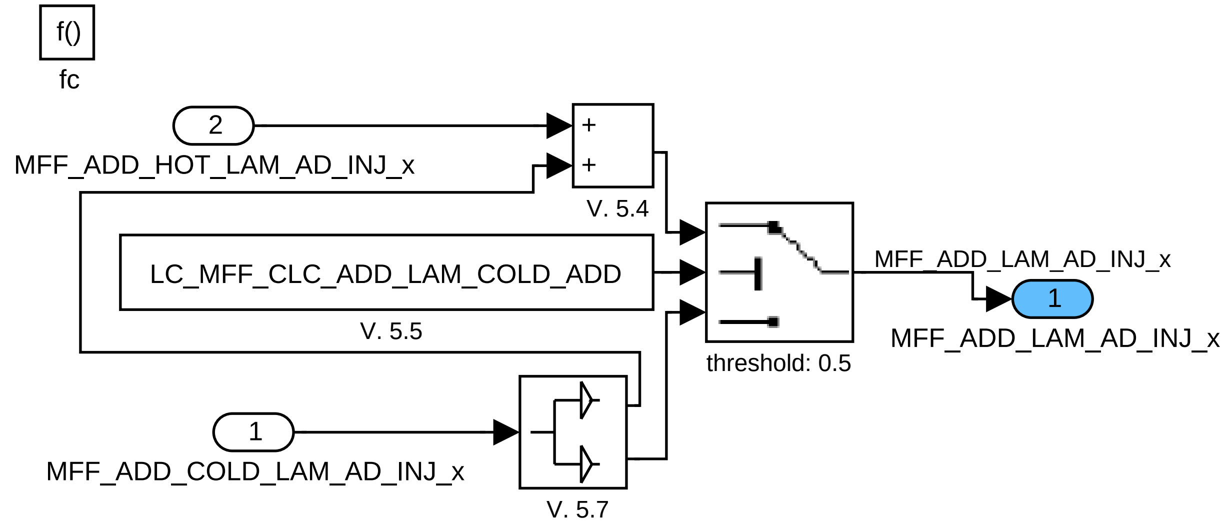


Figure 93:
Path: MFMA_ISPCLAIM0/OPM_1S/CLC_AD_VALUE/MFF_ADD_COLD

49.5.3.1.3 Hot adaptation value

At hot temperature range, the hot adaptation value is valid.

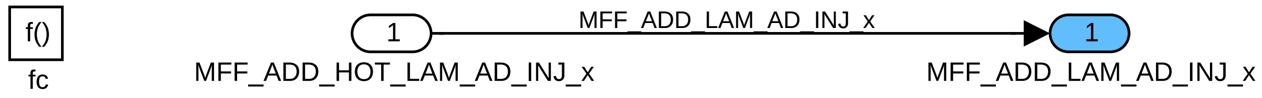


Figure 94:
Path: MFMA_ISPCLAIM0/OPM_1S/CLC_AD_VALUE/MFF_ADD_HOT

49.5.4 Resetting of adaptation values

The need to reset the adaptation values is checked every time at the transition engine off to engine on. The condition is checked for each cylinder, where the cylinder specific values are denoted within the graphical model with the suffix _x, in the textual description they are referenced via [x].

49.5.4.1 Reset due to manual request or injector change

If an injector was changed (i.e. the corresponding STATE_IV_CHG is set to one), the corresponding adaptation value is reset. If a manual reset is requested via LC_MFF_ADD_LAM_AD_INJ_CLR, all adaptation values are set to 0.
If a resetting of the NVMY data via service tester is requested with the flag LV_LAM_AD_INJ_CLR_AD_EXT, the adaptation values are also reset to 0.

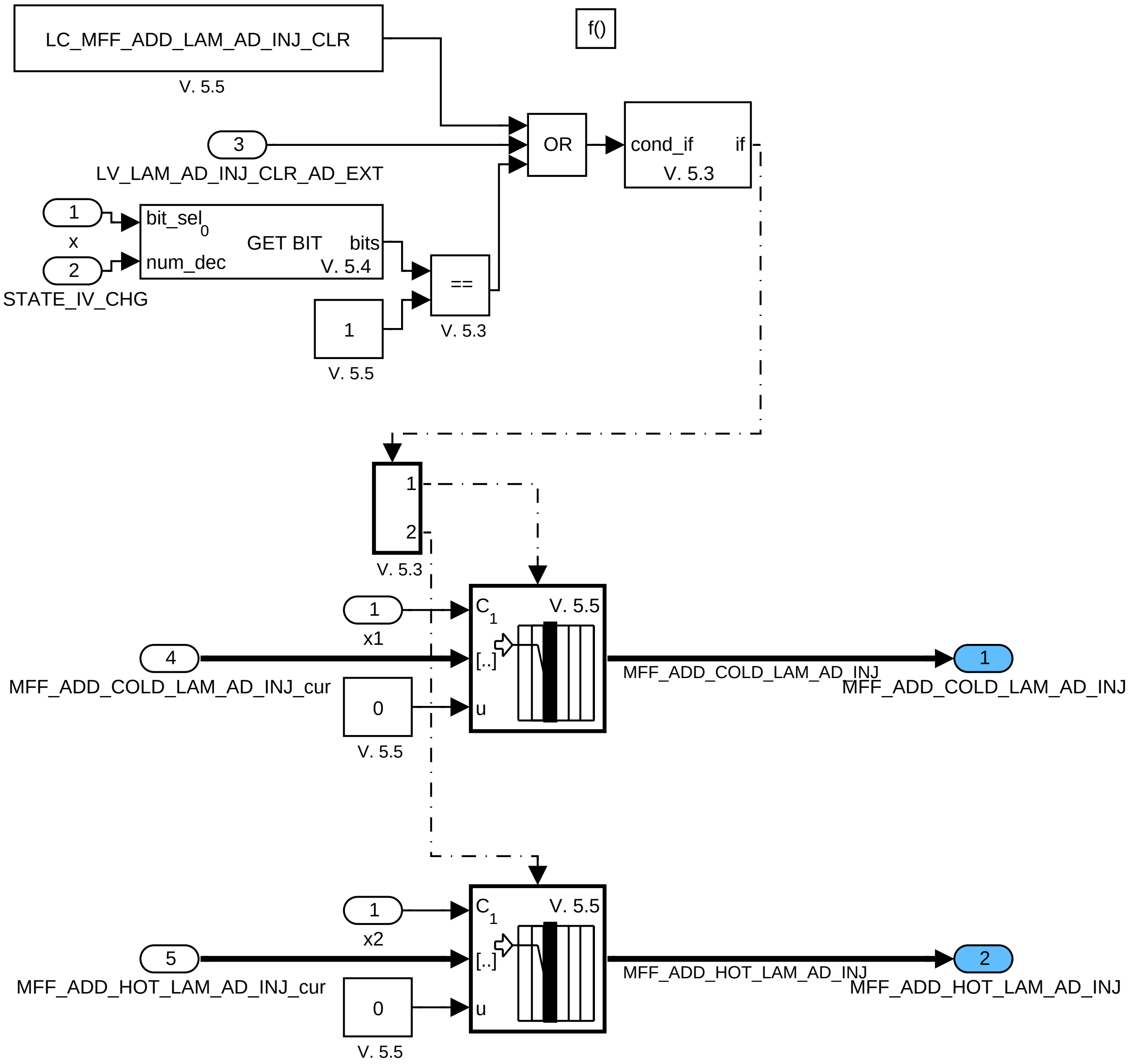


Figure 95:
Path: MFMA_ISPCLAIM0/OPM_ES2ERU/CLC

49.6 Calibration hints for function MFMA- LAM_AD_INJ Minimum fuel mass adaptation- Lambda adaptation via injection mode

Calibration interfering function:

- calibrate lambda controller before
- calibrate engine speed controller before

Calibration flowchart:

The sequence of calibration is described by the order in Calibration process

Calibration process:

49.6.1 Non calibrated data

NC_NR_PHA_LAM_AD_INJ	For maximum flexibility use "number cylinder per bank" * 2 + 1; After final calibration the number can be reduced for RAM optimization.
NC_NR_MIS_REP_LAM_AD_INJ	Number of tries to adapt a single cylinder that shows continuous misfire.
NC_NR_MPL_INJ_LAM_AD_INJ	Number of injection pulses for multiple injection mode. The value is only used for calculation of the adaptation value and has no influence on the injection itself.

49.6.2 Lambda adaptation via injection mode (Appl. Inc.)

LC_LAM_AD_INJ_INH	Manual inhibition of the adaptation. Only the manual activation conditions have a higher priority.

49.6.3 Activation of Lambda adaptation via injection mode

C_CTR_ABC_MAX_MIS_LAM_AD_INJ	Maximum number of misfire events for one cylinder at multiple injections to detect misfire.
C_CTR_MAX_DC_LAM_AD_INJ	Maximum number of driving cycles with a non successful adaptation for at least one cylinder. If the threshold is reached, an error flag is set and the cylinder will not be adapted anymore.
C_CTR_MIS_AD_MAX_LAM_AD_INJ	Maximum number of repeated adaptations for all cylinders due to misfire for at least one cylinder in a driving cycle. The value has to be equal or bigger then C_CTR_MIS_LAM_DC_MIN_LAM_AD_INJ
C_CTR_MIS_LAM_DC_MIN_LAM_AD_INJ	Threshold for number of repeated adaptations for all cylinders due to misfire for at least one cylinder in a driving cycle. If the threshold is at least reached, the counter for driving cycles with missed adaptations due to misfire is incremented.
C_CTR_MIS_MAX_LAM_AD_INJ	Maximum number of following tries for one cylinder due to misfire. The value has to be smaller or equal to NC_NR_MIS_REP_LAM_AD_INJ.
C_FAC_LAM_LDC_MPL_LAM_AD_INJ	Limited dynamic threshold for lambda controller during multiple injection. The threshold has to be set as high as the influence of multiple injections has no impact on the limited dynamic condition.
C_FAC_LAM_LDC_SNG_LAM_AD_INJ	Limited dynamic threshold for lambda controller during single injection.
C_FAC_LAM_MV_MAX_AD_LAM_AD_INJ	Maximum threshold during multiple injections to stop and repeat the adaptation. The activation condition C_FAC_LAM_MV_MAX_LAM_AD_INJ has to be taken in account.
C_FAC_LAM_MV_MIN_AD_LAM_AD_INJ	Minimum threshold during multiple injections to stop and repeat the adaptation. The activation condition C_FAC_LAM_MV_MAX_LAM_AD_INJ has to be taken in account.
C_FAC_LAM_MV_MAX_LAM_AD_INJ	Lambda controller output threshold for activation. Be sure that the distance to C_FAC_LAM_MV_MAX_AD_LAM_AD_INJ is as high as an adaptation can be calculated.
C_FAC_LAM_MV_MIN_LAM_AD_INJ	Lambda controller output threshold for activation. Be sure that the distance to C_FAC_LAM_MV_MIN_AD_LAM_AD_INJ is as high as an adaptation can be calculated.
C_FAC_TQ_REQ_DYW_LDC_LAM_AD_INJ	Limited dynamic threshold for torque.
C_MAF_DYW_LDC_LAM_AD_INJ	Limited dynamic threshold for mass air flow.
C_MAF_INT_LDC_LAM_AD_INJ	Max threshold for mass air flow integration for limited dynamics calculation.
C_N_BOL_LAM_AD_INJ	Engine speed threshold for activation.
C_N_TOL_LAM_AD_INJ	Engine speed threshold for activation.
C_N_DYW_LDC_LAM_AD_INJ	Limited dynamic threshold for engine speed.
C_NR_MIN_AD_LAM_AD_INJ_COLD_ADD	Number of hot adaptations to be performed before a cold adaptation can start.
C_T_AST_MIN_LAM_AD_INJ_MOD	Minimum time after start to be waited before the

	adaptation start.
C_TCO_MIN_LAM_AD_INJ	Minimum coolant temperature to be waited before the adaptation start.
C_TEMP_CAT_MAX_LAM_AD_INJ	Catalyst threshold for catalyst protection.
C_TEMP_LAM_AD_INJ_ENA	Selection for temperature input. The cold and hot areas are calculated with these input values.
C_TEMP_MIN_LAM_AD_INJ_COLD	Minimum threshold for cold area.
C_TEMP_MAX_LAM_AD_INJ_COLD	Maximum threshold for cold area.
C_TEMP_MIN_LAM_AD_INJ_HOT	Minimum threshold for hot area.
C_TEMP_MAX_LAM_AD_INJ_HOT	Maximum threshold for hot area.
ID_MFF_SP_BOL_LAM_AD_INJ	Minimum fuel mass to activate the adaptation
ID_MFF_SP_TOL_LAM_AD_INJ	Maximum fuel mass to activate the adaptation
IP_DIST_LAM_AD_INJ_COLD	Additive vale for distance threshold at successful adaptation. A calibration with zero at the fist element for both axes can be used to define a threshold for first initialization and an initialization after injector change.
IP_DIST_LAM_AD_INJ_HOT	Additive vale for distance threshold at successful adaptation. A calibration with zero at the fist element for both axes can be used to define a threshold for first initialization and an initialization after injector change.
LC_LAM_AD_INJ_ACT_COLD_MAN	Manual activation for cold adaptation range. Be sure that the lambda controller is able to operate in closed loop.
LC_LAM_AD_INJ_ACT_HOT_MAN	Manual activation for hot adaptation range. Be sure that the lambda controller is able to operate in closed loop.
LC_CTR_MIS_DC_LAM_AD_INJ_CLR	The driving cycle counter for adaptations with misfire is cleared after a successful adaptation if the flag is set.

49.6.4 Coordination of Lambda adaptation via injection mode

C_NR_PHA_LAM_AD_INJ	Maximum number of filled elements (single or multiple injections) in the ID_NR_CYL_LAM_AD_INJ +1.
ID_NR_CYL_LAM_AD_INJ	The sequence of multiple and single injection for each cylinder is defined. The first element has to be filled with SINGLE_ALL. Fill all unused elements (> C_NR_PHA_LAM_AD_INJ) with NOT_ACT.

49.6.5 Lambda adaptation via injection mode

C_CRLC_FAC_LAM_MMV_LAM_AD_INJ	Filter constant for lambda controller moving mean value.
C_CRLC_FAC_LAM_MV_LAM_AD_INJ	Filter constant for lambda controller mean value.
C_CRLC_TQ_DIF_SLOW_LAM_AD_INJ	Filter constant to weight the influence of the P- and D-Path of the idle speed controller.
C_FAC_LAM_ADJ_MIN_LAM_AD_INJ	Minimum lambda controller deviation of single and multiple injections to update the adaptation values.
C_FAC_LAM_FIL_DIF_LAM_AD_INJ	Threshold for difference of lambda controller mean value and moving mean value to allow transition to MV_CLC. With this value a stable lambda controller can be detected.
C_FAC_MFF_COR_CYL_LAM_AD_INJ	Cylinder individual factor to weight the adaptation result.
C_MAF_INT_MV_CLC_LAM_AD_INJ	Minimum mass air flow integration to reach a stable lambda controller at mean value calculation phase.
C_MAF_INT_TRA_PHA_LAM_AD_INJ	Minimum mass air flow integration to reach a stable lambda controller at transient phase.
C_MFF_ADD_LAM_AD_MAX	Maximum limitation of adaptation value.
C_MFF_ADD_LAM_AD_MIN	Minimum limitation of adaptation value.
C_T_INT_MV_CLC_LAM_AD_INJ	Minimum time integration to reach a stable lambda controller at mean value calculation phase.
C_T_INT_TRA_PHA_LAM_AD_INJ	Minimum time integration to reach a stable lambda controller at transient phase.
ID_CRLC_MFF_ADD_COLD_LAM_AD_INJ	Correlation constant for wighting the adaptaiton result.
ID_CRLC_MFF_ADD_HOT_LAM_AD_INJ	Correlation constant for wighting the adaptaiton result.
ID_MFF_DELTA_MIS_LAM_AD_INJ	Corrective fuel mass at detected misfire. The last element can be calculated to zero, so that after a failed adaptation due to misfire no more correction are done.
IP_MFF_ADD_LAM_AD_INJ	Normalization to reference fuell mall. This has to be calculated together with the incection module.
LC_MFF_ADD_LAM_AD_INJ_CLR	The adaptaiton values will be deleted at transition engine off to on.
LC_MFF_CLC_ADD_LAM_COLD_ADD	Two strategies can be choosen: - the hot and cold range are calculated seperately - the hot adaptation is valid for the whole temperatue range. The cold adaptation is valid for the cold range as an offset.

50 Miscellaneous

Table of Contents

Index

50.1 Bios H-Bridge Diagnosis Function for DISA

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
BIOS_HBR_FAULT_STATUS_VIM	O	0H 1H	NO_ERROR ETC_HBR_ERROR	0	[-]
STATUS OF H-BRIDGE DISA					

FUNCTION DESCRIPTION:

General information:

The function Bios H-Bridge Diagnosis manages detection of H-Bridge faults witch deactivate the DISA H-Bridge.

Description:

The following error conditions can be distinguished:

- Undervoltage
- Overcurrent
- Overtemperature

Application conditions:

Initialisation: 0
Recurrence: 10 ms

Formula section:

```
if      one error condition is fulfilled
then    BIOS_HBR_FAULT_STATUS_VIM = 1
else    BIOS_HBR_FAULT_STATUS_VIM = 0
endif
```

50.2 Exhaust flap actuation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_EF	V/O	0...1H	0...1	1	[-]
Logical value exhaust flap, 1 = closed flap					
FAC_TQ_EF_THD	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Switching threshold for exhaust flap					
FAC_TQ_EF_THD_CTOP	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Switching threshold for exhaust flap without hysteresis					
N_EF	V	0...FFH	0...8160	32	[rpm]
N for exhaust flap					
T_DLY_EF	V	0...FFH	0...25.5	0.1	[s]
Time delay for closing exhaust flap					
LV_CDN_AC_VS	-	0...1H	0...1	1	[-]
Logical value minimum speed AC exceeded					

Input data:

GEAR_EF	N_32	VS	LV_CH_N_SP_IS
FAC_TQ_REQ	T_AST	LV_AT	LV_VAR_EF
LV_ST	LV_ES	LV_CTOP	LV_EF_EXT_ADJ
LV_ACT_EF_EXT_ADJ			

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_EF_NR	1	1...4H	1...4	1	[-]
Number of the exhaust flaps in system					

Import actions:

ACTION_INFR_SetEf(IN<ef_nr>, IN <Lv_ef>)

General information:

To optimize the vehicle noise behaviour, an exhaust flap can be activated depending on the engine speed (N_EF) and the load (FAC_TQ_REQ). The exhaust flap is closed below a FAC_TQ_REQ value threshold, it is opened above the threshold.
The exhaust flap activation is only activated after the time C_T_EF_AST after the transition from Start.

The switching threshold (FAC_TQ_EF_THD) is depending on enginespeed (N_EF), the selected gear (GEAR_EF), the transmission (manuell or automatic) and the convertible top (LV_CTOP = 0 closed, or LV_CTOP = 1 open).

To prevent the exhaust flap from switching continuously during constant driving at the switching threshold, two hysteresises (C_N_EF_HYS and C_FAC_TQ_EF_THD_HYS) are introduced for the switching process.
The closing of the exhaust flap is delayed depending on the enginespeed (N_32).

During catalyst heating with increased idle speed setpoint, the state of the exhaust flap is adjustable by C_STATE_CH_EF.

At low vehicle speeds the state of the exhaust flap is adjustable by C_STATE_VS_MIN_EF_xx, but C_STATE_CH_EF has higher priority as C_STATE_VS_MIN_EF_xx (xx = MT/AT).
For MSD85 is NC_EF_NR=2 and for all other projects is NC_EF_NR=1.

Application conditions:

Initialisation: all variables are initialised with 0 at deactivation

Recurrence: 100 ms

Activation: LV_IGK = 1 and (LV_VAR_EF = 1 **or** LV_EF_EXT_ADJ = 1)

Deactivation: LV_IGK = 0;

at Deactivation LV_EF = LC_EF_CONF // is still calculated during PWL;

Formula section:

- Control of sound exhaust flap by external adjustment

```
If      LV_EF_EXT_ADJ = 1
Then    LV_EF = LV_ACT_EF_EXT_ADJ
Else    external adjustment passive/standard function active
```

- Flap during start and engine stopped

```
    If LV_ST = 1 or LV_ES = 1
    Then LV_EF = LC_EF_CONF      \\ flap opened; in case of LC_EF_CONF = 1 flap closed
    Endif
```

- Highest priority is ‘time after start’, P1

```
If      VS > C_VS_MIN_EF_AC_2
Then    LV_CDN_AC_VS = 1        *** initialised at deactivation***
Endif;
```

```
If   T_AST ≤ C_T_EF_AST    AND   LV_CDN_AC_VS = 0
  then    LV_EF = LC_STATE_EF_AST                    see next priority P2
endif
```

- Next priority is ‘catalyst heating with increased idle speed setpoint’, P2

```
If   LV_CH_N_SP_IS = 1
  then    LV_EF = LC_STATE_CH_EF                     see next priority P3
```

```
endif
```

- Next priority is ‘Accelerated passing by’, P3

```
If    LV_EF_AC_ACT = 1
LV_EF = LV_EF_AC_ACT                                    see next priority P4
endif
```

- Next priority is ‘vehicle speed dependency’, P4

Automatic transmission vehicle (LV_AT = 1):

```
If    VS  <  C_VS_MIN_EF_AT
  then      LV_EF  =  LC_STATE_VS_MIN_EF_AT              see next priority P5
endif
```

Manual transmission vehicle (LV_AT = 0):

```
If    VS  <  C_VS_MIN_EF_MT
  then      LV_EF  =  LC_STATE_VS_MIN_EF_MT              see next priority P5
endif
```

- Next priority is 'convertible top' , P5 *\\ calculation of thresholdvalues*

- enginespeed hysteresis

```
    If LV_EF = 0
    Then  N_EF = N_32 + C_N_EF_HYS
    Else  N_EF = N_32
```

- Calculation switching threshold (FAC_TQ_EF_THD_CTOP)

```
IF (1) LV_AT = 0                                 \\ manuel transmission
    IF LV_CTOP = 0                               \\ convertibel top closed
    Then  FAC_TQ_EF_THD_CTOP =
          ID_FAC_TQ_EF_THD_CTOP_CLOSE_MT (N_EF; GEAR_EF)
    Else  FAC_TQ_EF_THD_CTOP =
          ID_FAC_TQ_EF_THD_CTOP_OPEN_MT (N_EF; GEAR_EF)
ELSE (1)                                         \\ automatic transmission
    IF LV_CTOP = 0                               \\ convertibel top closed
    Then  FAC_TQ_EF_THD_CTOP =
          ID_FAC_TQ_EF_THD_CTOP_CLOSE_AT (N_EF; GEAR_EF)
    Else  FAC_TQ_EF_THD_CTOP =
          ID_FAC_TQ_EF_THD_CTOP_OPEN_AT (N_EF; GEAR_EF)
```

- Calculation switching threshold hyteresis

```
If LV_EF = 0                        \\ flap opened; in case of LC_EF_CONF = 1 flap closed
    Then  FAC_TQ_EF_THD = FAC_TQ_EF_THD_CTOP - C_FAC_TQ_EF_THD_HYS
    Else  FAC_TQ_EF_THD = FAC_TQ_EF_THD_CTOP
```

- Compare threshold and actual value with close delay

```
IF $FAC\_TQ\_REQ_n < FAC\_TQ\_EF\_THD_n$ and $FAC\_TQ\_REQ_{n-1} \geq FAC\_TQ\_EF\_THD_{n-1}$
    Then  T_DLY_EF =  IP_T_DLY_EF
    Else  T_DLY_EF – --                \\ (decremenet T_DLY_EF)

Endif

If FAC_TQ_REQ ≥ FAC_TQ_EF_THD
    Then LV_EF = 0
    Else  IF T_DLY_EF = 0
              Then  LV_EF = 1
              Else  LV_EF = 0
          Endif
Endif
```

The output stage is energized to close the normal NC_EF_NR exhaust flaps and to open the inverted NC_EF_NR exhaust flaps (depending on LC_EF_CONF).

Switch Hardware PIN
For i=0...NC_EF_NR-1

```
IF LC_EF_CONF = 0
    THEN Switch Pin (I/O Software) with LV_EF
ACTION_INFR_SetEf(IN<ef_nr>, IN <Lv_ef>)
ELSE Switch Basic (I/O Software) with not LV_EF
ACTION_INFR_SetEf(IN<ef_nr>, IN <Lv_ef>)
```

Calibration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
ID_FAC_TQ_EF_THD_CTOP_OPEN_MT	6*7	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_EF	6	0...FFH	0...8160	32	[rpm]
LDPM_GEAR_EF_1	7	0...6H	0...6	1	[-]
Switching threshold FAC_TQ_EF for convertible top open MT					
ID_FAC_TQ_EF_THD_CTOP_CLOSE_MT	6*7	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_EF	6	0...FFH	0...8160	32	[rpm]
LDPM_GEAR_EF_1	7	0...6H	0...6	1	[-]
Switching threshold FAC_TQ_EF for convertible top closed MT					
ID_FAC_TQ_EF_THD_CTOP_OPEN_AT	6*7	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_EF	6	0...FFH	0...8160	32	[rpm]
LDPM_GEAR_EF_1	7	0...6H	0...6	1	[-]
Switching threshold FAC_TQ_EF for convertible top open AT					
ID_FAC_TQ_EF_THD_CTOP_CLOSE_AT	6*7	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_EF	6	0...FFH	0...8160	32	[rpm]
LDPM_GEAR_EF_1	7	0...6H	0...6	1	[-]
Switching threshold FAC_TQ_EF for convertible top closed AT					
IP_T_DLY_EF	8	0...FFH	0...25.5	0.1	[s]
LDP_N_32_IP_T_DLY_EF	8	0...FFH	0...8160	32	[rpm]
Time delay for closing exhaust flap					
C_N_EF_HYS	1	0...FFH	0...8160	32	[rpm]
Hysteresis for switching threshold					
C_FAC_TQ_EF_THD_HYS	1	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Hysteresis for switching threshold					
LC_STATE_CH_EF	1	0...1H	0...1	1	[-]
Exhaust flap actuation in case catalyst heating with increased idle speed					
C_VS_MIN_EF_AT	1	0...FFH	0...255	1	[km/h]
Speed threshold for continuously opened exhaust flap, automatically shifted transmission					
C_VS_MIN_EF_MT	1	0...FFH	0...255	1	[km/h]
Speed threshold for continuously opened exhaust flap, manually shifted transmission					
C_T_EF_AST	1	0...F8H	0...62	0.25	[s]
Cranking time for EF actuation active					
LC_STATE_VS_MIN_EF_AT	1	0...1H	0...1	1	[-]
Exhaust flap actuation in case of vs < C_VS_MIN_EF_AT for AT version					
LC_STATE_VS_MIN_EF_MT	1	0...1H	0...1	1	[-]
Exhaust flap actuation in case of vs < C_VS_MIN_EF_MT for MT version					
LC_STATE_EF_AST	1	0...1H	0...1	1	[-]
State exhaust flap after start					
LC_EF_CONF	1	0...1H	0...1	1	[-]
Switch for inverted exhaust flap; = 0 normal flap; = 1 inverted flap					

50.2.1 Torque limitation for accelerated passing-by

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_EF_AC_ACT	O/V	0...1H	0...1	1	[-]
Boolean for accelerated passing by active					
LV_EF_AC	V	0...1H	0...1	1	[-]
Exhaust flap for accelerated passing by active via pedal value condition					
LV_EF_AC_CDN	V	0...1H	0...1	1	[-]
Boolean condition for detection of accelerated passing-by met					
T_DLY_EF_AC	V	0...FFFFH	0...655.35	0.01	[s]
delay time to start TQ change calculation					
T_VS_CDN_EF_AC	V	0...FFFFH	0...655.35	0.01	[s]
Minimum time condition for detection of accelerated passing by					
TQI_REQ_SLOW_EF_AC	V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Torque request during accelerated passing by for deactivation					
TQI_REQ_EF_AC_THD	V	0...7FFFH	0...1023.96875	0.03125	[Nm]
Torque start value accelerated passing by					
PV_AV_THD_EF_AC	V	0...FFH	0...99.60937	0.390625	[%]
Pedal value threshold for activation of accelerated passing by					
VAR_GB_EF_AC	V	0...1H	0...1	1	[-]
Indication for gearbox variant: 0 = manual, 1 = automatic					

Input data:

TQI_REQ_SLOW	VS	TCO	PV_AV
LV_AT	GR_MT	GR_AT	LV_FL
	TQ_REQ_CLU	TQ_LOSS	

FUNCTION DESCRIPTION:

Application conditions:

Recurrence: 10 ms

Initialisation at reset: all = 0

50.2.1.1 Detection of accelerated passing by

Application conditions:

Activation/Deactivation:

```
if      {(LV_AT = 0   and (C_GR_MT_EF_1 <= GR_MT <= C_GR_MT_EF_2))
        or
        (LV_AT = 1    and (C_GR_AT_EF_1 <= GR_AT <= C_GR_AT_EF_2))}
        and
        TCO > C_TCO_EF_AC
then    Formula section below
else    LV_EF_AC_CDN           = 0
        LV_EF_AC               = 0
        LV_EF_AC_ACT           = 0
        T_VS_CDN_EF_AC         is reset
        T_DLY_EF_AC            = 0
endif
```

Formula section:

- calculation of TQI_REQ_EF_AC_THD:

```
if      LV_AT = 0
then    VAR_GB_EF_AC = 0
    if      GR_MT = C_GR_MT_EF_1
    then        TQI_REQ_EF_AC_THD = C_TQI_REQ_EF_AC_THD_GR_1
    else        TQI_REQ_EF_AC_THD = C_TQI_REQ_EF_AC_THD_GR_2
    endif
else    VAR_GB_EF_AC = 1
        TQI_REQ_EF_AC_THD = C_TQI_REQ_EF_AC_THD_GR_AT
endif
```

- Vehicle speed condition LV_EF_AC_CDN:

```
if          TQI_REQ_SLOW  <  TQI_REQ_EF_AC_THD
            and
            (LV_AT = 0 or (GR_AT > C_GR_AT_EF_1))
            and
            LV_EF_AC_ACT = 0
then   if   VS > C_VS_MIN_EF_AC_1  and
            VS < C_VS_MAX_EF_AC_1
            % => tight range for VS-condition fulfilled  %
       then    if   T_VS_CDN_EF_AC  <  C_T_VS_CDN_EF_AC
               then        T_VS_CDN_EF_AC is incremented
                           LV_EF_AC_CDN = 0
               else        LV_EF_AC_CDN = 1
               endif
       else         LV_EF_AC_CDN = 0
                    if     VS > C_VS_MIN_EF_AC_2  and
                           VS < C_VS_MAX_EF_AC_2
                     % => wide range for VS-condition fulfilled  %
                    then   T_VS_CDN_EF_AC(n) = T_VS_CDN_EF_AC(n-1)
                    else   T_VS_CDN_EF_AC is reset
                    endif
       endif
else        if      TQI_REQ_SLOW >= TQI_REQ_EF_AC_THD       and
            (LV_AT = 0 or (GR_AT > C_GR_AT_EF_1))           and
            LV_EF_AC_CDN = 0
            then           T_VS_CDN_EF_AC is reset
            endif
endif
```

- Activation via pedal value LV_EF_AC:

```
if        LV_AT = 1
then      PV_AV_THD_EF_AC = C_PV_AV_THD_EF_AC_AT
else      PV_AV_THD_EF_AC = C_PV_AV_THD_EF_AC
endif

if        LV_EF_AC_CDN = 1                          and
          {LV_FL = 1   or    PV_AV > PV_AV_THD_EF_AC}
then      LV_EF_AC = 1
else      LV_EF_AC = 0
endif
```

50.2.1.2 Torque limitation for accelerated passing by

Application conditions:

Activation: LV_EF_AC = 1

Deactivation: LV_EF_AC = 0
at deactivation:
TQI_REQ_SLOW_EF_AC = 7FFF hex / 1024Nm
LV_EF_AC_ACT = 0
T_DLY_EF_AC = 0
LV_EF_AC_CDN = 0

Formula section:

- Torque limitation is active LV_EF_AC_ACT:

```
If        LV_EF_AC_ACT = 0
then
  if  TQI_REQ_SLOW > TQI_REQ_EF_AC_THD
  then    TQI_REQ_SLOW_EF_AC   = TQI_REQ_EF_AC_THD
          T_DLY_EF_AC      = 0
          LV_EF_AC_ACT     = 1
  endif
else      T_DLY_EF_AC is incremented
  if  T_DLY_EF_AC  >=  C_T_DLY_EF_AC
  then    TQI_REQ_SLOW_EF_AC =          MIN(TQ_REQ_CLU – TQ_LOSS;
                                        IP_TQI_REQ_SLOW_EF_AC)
  endif

  if  TQI_REQ_SLOW_EF_AC >= C_FAC_END_EF_AC * (TQ_REQ_CLU – TQ_LOSS)
  then    LV_EF_AC_ACT              = 0
          LV_EF_AC                  = 0
          LV_EF_AC_CDN              = 0
          T_VS_CDN_EF_AC            is reset
          T_DLY_EF_AC      = 0
          TQI_REQ_SLOW_EF_AC   = 7FFF hex / 1024Nm
  endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_MIN_EF_AC_1	1	0...FFH	0...255	1	[km/h]
Lower vehicle speed tight threshold for detection of accelerated passing by					
C_VS_MAX_EF_AC_1	1	0...FFH	0...255	1	[km/h]
Upper vehicle speed tight threshold for detection of accelerated passing by					
C_T_VS_CDN_EF_AC	1	0...FFFFH	0...655.35	0.01	[s]
Minimum time condition for detection of accelerated passing by					
C_T_DLY_EF_AC	1	0...FFFFH	0...655.35	0.01	[s]
delay time to start TQ change calculation					
C_PV_AV_THD_EF_AC	1	0...FFH	0...99.60937	0.390625	[%]
Pedal value threshold for activation of accelerated passing by					
C_PV_AV_THD_EF_AC_AT	1	0...FFH	0...99.60937	0.390625	[%]
Pedal value threshold for activation of accelerated passing by for AT					
C_TQI_REQ_EF_AC_THD_GR_1	1	0...7FFFH	0...1023.96875	0.03125	[Nm]
Torque request limit during accelerated passing by in selected gear 1					
C_TQI_REQ_EF_AC_THD_GR_2	1	0...7FFFH	0...1023.96875	0.03125	[Nm]
Torque request limit during accelerated passing by in selected gear 2					
C_TQI_REQ_EF_AC_THD_GR_AT	1	0...7FFFH	0...1023.96875	0.03125	[Nm]
Torque request limit during accelerated passing by for AT					
C_TCO_EF_AC	1	0...FEH	-48...142.5	0.75	[°C]
TCO threshold for detection of accelerated passing by					
C_VS_MIN_EF_AC_2	1	0...FFH	0...255	1	[km/h]
Lower vehicle speed wide threshold for detection of accelerated passing by					
C_VS_MAX_EF_AC_2	1	0...FFH	0...255	1	[km/h]
Upper vehicle speed wide threshold for detection of accelerated passing by					
IP_TQI_REQ_SLOW_EF_AC	2*6	0...7FFFH	0...1023.96875	0.03125	[Nm]
LDP_VAR_GB_EF_AC	2	0...1H	0...1	1	[-]
LDP_VS_EF_AC	6	0...FFH	0...255	1	[km/h]
Torque request during accelerated passing by for deactivation					
C_FAC_END_EF_AC	1	0...FFH	0...1	3.9216e-3	[-]
Calibration factor to end detection for accelerated passing by					
C_GR_MT_EF_1	1	0...FFH	0...255	1	[-]
Manual gearbox: accelerated pass gear 1					
C_GR_MT_EF_2	1	0...FFH	0...255	1	[-]
Manual gearbox: accelerated pass gear 2					
C_GR_AT_EF_1	1	0...FFH	0...255	1	[-]
Automatic gearbox: accelerated pass gear 1					
C_GR_AT_EF_2	1	0...FFH	0...255	1	[-]
Automatic gearbox: accelerated pass gear 2					

50.3 Sound Flap Control / Sport switch

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TQ_SOF_THD	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Switching threshold for sound flap					
FAC_TQ_SOF_THD_CTOP	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Switching threshold for sound flap without hysteresis					
LV_SOF	O/V	0...1H	0...1	1	[-]
logical value sound flap (1 = activated)					
LV_SOF_SWI	O/V	0...1H	0...1	1	[-]
Sport-mode switch acting					
LV_SOF_SWI_AMT	O/V	0...1H	0...1	1	[-]
Sport-mode switch AMT acting					
LV_SOF_SWI_AMT_REQ	O/V	0...1H	0...1	1	[-]
Sport-mode AMT required by switch					
LV_SOF_SWI_REQ	O/V	0...1H	0...1	1	[-]
Sport-mode required by switch					
N_SOF	V	0...FFH	0...8160	32	[rpm]
N for sound flap					
T_DLY_SOF	V	0...FFH	0...25.5	0.1	[s]
Time delay for closing sound flap					

Input data:

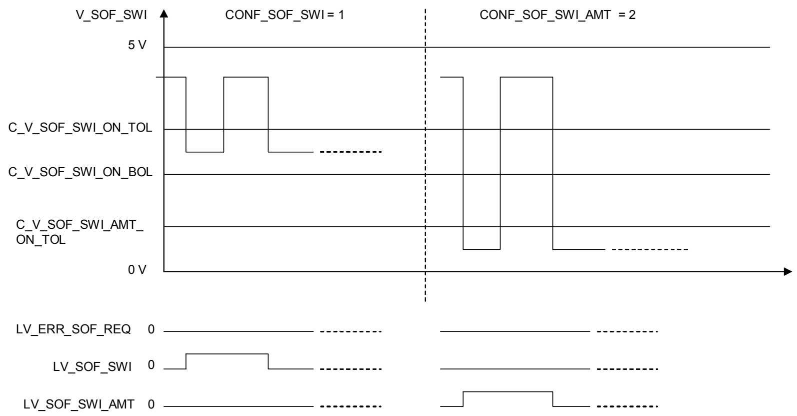
CONF_SOF_SWI	FAC_TQ_REQ	GEAR_EF	LV_ACT_SOF_EXT_ADJ
LV_AT	LV_CITY	LV_CTOP	LV_ERR_BN_GEAR_REV
LV_ERR_BN_VEH_MOD	LV_ERR_CAN_BOFF	LV_ERR_SOF_INH_MON	LV_ERR_SOF_REQ
LV_ES	LV_ES	LV_IGK	LV_SOF_EXT_ADJ
LV_ST	LV_VAR_BN	LV_VAR_SOF	N_32
STATE_VEH_MOD	T_AST	V_SOF_SWI	VS

50.3.1 Acquisition of sport-mode-switch

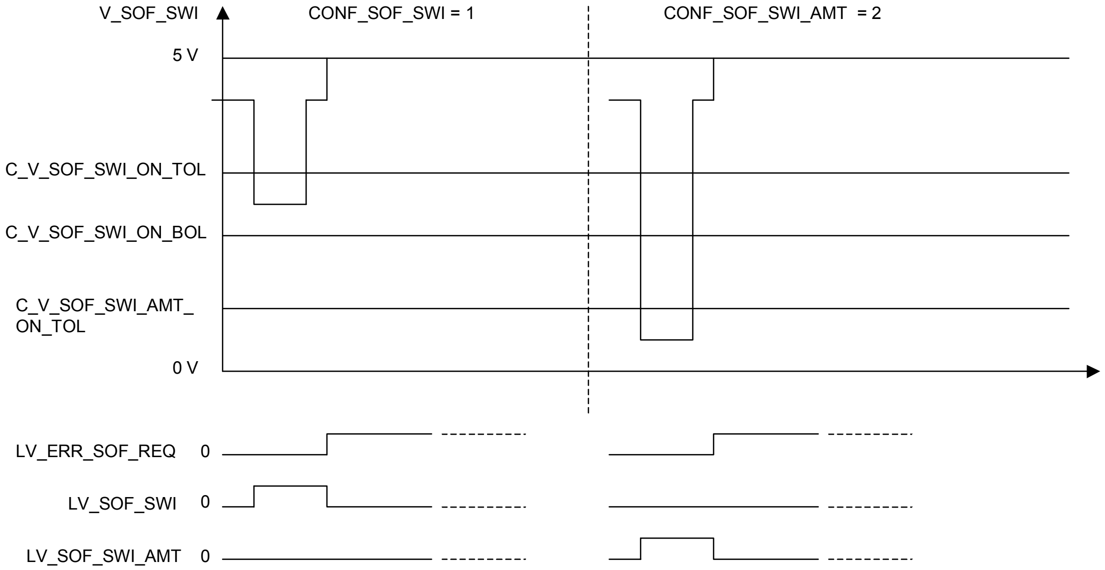
General information:

Calculation of the sport-mode request by taking the voltage of the sport-mode-switch.
The request acts according to the rising egde of the voltage signal.
There must be for 3 cyles the same result of LV_SOF_SWI(_AMT) to accept the level.

Signal flow diagram:



In case of failure LV_ERR_SOF_REQ =1 inhibitation of LV_SOF_SWI(_AMT):



Application conditions:

Initialisation:	all variables are initialised with 0 at:
	reset or LV_ES = 0-->1
Recurrence:	10ms
Activation:	every engine state

Formula section:

If (1) LV_VAR_BN = 0
Then (1)

If(2) LV_ERR_SOF_REQ = 0
Then(2)

If(3) (V_SOF_SWI > C_V_SOF_SWI_ON_TOL) changes to
(C_V_SOF_SWI_ON_BOL ≤ V_SOF_SWI ≤ C_V_SOF_SWI_ON_TOL)
(equal to 0 ->1)
***And** this result is valid for 3 evaluations*

```
            Then(3) LV_SOF_SWI = ! LV_SOF_SWI
            Endif(3)

            If(4)   (V_SOF_SWI > C_V_SOF_SWI_AMT_ON_TOL) changes to
                    (V_SOF_SWI ≤ C_V_SOF_SWI_AMT_ON_TOL)
                    (equal to 0 ->1)
                    And this result is valid for 3 evaluations
            Then(4) LV_SOF_SWI_AMT = ! LV_SOF_SWI_AMT
            Endif(4)

        Else(2)     LV_SOF_SWI = 0
                    LV_SOF_SWI_AMT = 0
        Endif(2)

    Else (1)

        If (5)      LV_VAR_BN = 1                      And
                    LV_ERR_BN_VEH_MOD = 0              And
                    LV_ERR_CAN_BOFF = 0                And
                    LV_ERR_BN_GEAR_REV = 0             And
                    STATE_VEH_MOD = 2H
        Then (5)    LV_SOF_SWI = 1
        Else (5)    LV_SOF_SWI = 0
        Endif(5)

    Endif (1)
```

<u>Calculation of LV_SOF_SWI_REQ and LV_SOF_SWI_AMT_REQ:</u>

```
If      (LV_CITY = 0 Or
        LC_CONF_CITY_ENA = 1)          And
        CONF_SOF_SWI     ≠ 2           And
        LC_VAR_SOF_SWI   = 1           And
        LV_SOF_SWI       = 1           And
        LV_ERR_SOF_INH_MON = 0
Then    LV_SOF_SWI_REQ  = 1
Else    LV_SOF_SWI_REQ  = 0
Endif

If      LV_CITY = 0                    And
        CONF_SOF_SWI            = 2    And
```

```
            LC_VAR_SOF_SWI_AMT   = 1     And
            LV_SOF_SWI_AMT       = 1
Then        LV_SOF_SWI_AMT_REQ   = 1
Else        LV_SOF_SWI_AMT_REQ   = 0
Endif
```

50.3.2 Acquisition of sound-flap

General information:

To optimize the vehicle noise behaviour, an sound flap can be activated depending on the engine speed (N_SOF) and the load (FAC_TQ_REQ). The sound flap is closed below a FAC_TQ_REQ threshold, it is opened above the threshold.
The sound flap is only activ after the time C_T_SOF_AST after the transition from Start.

The switching threshold (FAC_TQ_SOF_THD) is depending on enginespeed (N_SOF), the selected gear (GEAR_EF), the transmission (manuell or automatic) and the convertible top (LV_CTOP = 0 closed, or LV_CTOP = 1 open).

To prevent the sound flap from switching continuously during constant driving at the switching threshold, two hysteresises (C_N_SOF_HYS and C_FAC_TQ_SOF_THD_HYS) are introduced for the switching process.
The closing of the sound flap is delayed depending on the enginespeed (N_32).

At low vehicle speeds the state of the sound flap is adjustable by C_STATE_VS_MIN_SOF_xx (xx = MT/AT).

The sound flap can be tested via KWP protocol at EOL or at service station by setting LV_SOF_EXT_ADJ = 1.

Application conditions:

Initialisation:	all variables are initialised with 0 at deactivation
Recurrence:	100ms
Activation:	LV_VAR_SOF = 1 **or** LV_SOF_EXT_ADJ = 1
Deactivation:	LV_IGK = 0

Formula section:

Control of sound-flap by external adjustment:

```
IF          LV_SOF_EXT_ADJ = 1
THEN        LV_SOF = LV_ACT_SOF_EXT_ADJ
Endif
```

Control of sound-flap

- **SOF during start and engine stopped**

```
    If LV_ST = 1 or LV_ES = 1
    Then LV_SOF = 0                                  \\ flap closed
```

```
    Endif
```

- Highest priority is ‘time after start’,   P1

```
If    T_AST  ≤  C_T_SOF_AST
  then      LV_SOF  =  LC_STATE_SOF_AST                    see next priority  P2
endif
```

- Next priority is ‘vehicle speed dependency’,   P2

Automatic transmission vehicle (LV_AT = 1):

```
If    VS  <  C_VS_MIN_SOF_AT
  then      LV_SOF  =  LC_STATE_VS_MIN_SOF_AT              see next priority P3
endif
```

Manual transmission vehicle (LV_AT = 0):

```
If    VS  <  C_VS_MIN_SOF_MT
  then      LV_SOF  =  LC_STATE_VS_MIN_SOF_MT              see next priority  P3
endif
```

- Next priority is 'convertible top' , P3   *\\ calculation of thresholdvalues*

- enginespeed hysteresis

```
    If LV_SOF = 0
    Then   N_SOF = N_32 + C_N_SOF_HYS
    Else   N_SOF = N_32
    Endif
```

- Calculation switching threshold (FAC_TQ_SOF_THD_CTOP)

```
IF (1) LV_AT = 0                                        \\ manuel transmission
    IF LV_CTOP = 0                                      \\ convertibel top closed
    Then   FAC_TQ_SOF_THD_CTOP =
             ID_FAC_TQ_SOF_THD_CTOP_CLOSE_MT (N_SOF; GEAR_EF)
    Else   FAC_TQ_SOF_THD_CTOP =                        \\ convertibel top open
             ID_FAC_TQ_SOF_THD_CTOP_OPEN_MT (N_SOF; GEAR_EF)
ELSE (1)                                                \\ automatic transmission
    IF LV_CTOP = 0                                      \\ convertibel top closed
    Then   FAC_TQ_SOF_THD_CTOP =
             ID_FAC_TQ_SOF_THD_CTOP_CLOSE_AT (N_SOF; GEAR_EF)
    Else   FAC_TQ_SOF_THD_CTOP =                        \\ convertibel top open
             ID_FAC_TQ_SOF_THD_CTOP_OPEN_AT (N_SOF; GEAR_EF)
```

- Calculation switching threshold hyteresis

```
If LV_SOF = 0                                              \\ flap closed
Then   FAC_TQ_SOF_THD = FAC_TQ_SOF_THD_CTOP -
       C_FAC_TQ_SOF_THD_HYS
Else   FAC_TQ_SOF_THD = FAC_TQ_SOF_THD_CTOP                \\ flap open
```

- Compare threshold and actual value with close delay

```
IF FAC_TQ_REQ $_{n}$ < FAC_TQ_SOF_THD $_{n}$ and FAC_TQ_REQ $_{n-1}$ $\geq$ FAC_TQ_SOF_THD $_{n-1}$
    Then   T_DLY_SOF =  IP_T_DLY_SOF
    Else   T_DLY_SOF – --                        \\ (decremenet T_DLY_SOF)

Endif

IF    FAC_TQ_REQ $\geq$ FAC_TQ_SOF_THD
    Then LV_SOF = 1                              \\ flap open
           Else   IF T_DLY_SOF = 0
                  Then         LV_SOF = 0        \\ flap closed
                  Else         LV_SOF = 1        \\ flap open
                  Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_TQ_SOF_THD_HYS	1	8000...7FFFH	-2...1.99993	0.061e-3	[-]
Hysteresis for switching threshold sound flap					
C_N_SOF_HYS	1	0...FFH	0...8160	32	[rpm]
N hysteresis SOF					
C_T_SOF_AST	1	0...FFH	0...25.5	0.1	[s]
cranking time for SOF actuation active					
C_V_SOF_SWI_AMT_ON_TOL	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage-threshold top limit for sport switch mode ON (AMT)					
C_V_SOF_SWI_ON_BOL	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage-threshold bottom limit for sport switch mode ON					
C_V_SOF_SWI_ON_TOL	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage-threshold top limit for sport switch mode ON					
C_VS_MIN_SOF_AT	1	0...FFH	0...255	1	[km/h]
Speed threshold for continously opened sound flap (automatic transmission)					
C_VS_MIN_SOF_MT	1	0...FFH	0...255	1	[km/h]
Speed threshold for continously opened sound flap (manuel transmission)					
ID_FAC_TQ_SOF_THD_CTOP_CLOSE_AT	4*7	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_SOF	4	0...FFH	0...8160	32	[rpm]
LDPM_GEAR_EF_1	7	0...7H	0...7	1	[-]
Switching threshold FAC_TQ_SOF for convertible top closed (automatic transmission)					
ID_FAC_TQ_SOF_THD_CTOP_CLOSE_MT	4*7	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_SOF	4	0...FFH	0...8160	32	[rpm]
LDPM_GEAR_EF_1	7	0...7H	0...7	1	[-]
Switching threshold FAC_TQ_SOF for convertible top closed (manuel transmission)					
ID_FAC_TQ_SOF_THD_CTOP_OPEN_AT	4*7	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_SOF	4	0...FFH	0...8160	32	[rpm]
LDPM_GEAR_EF_1	7	0...7H	0...7	1	[-]
Switching threshold FAC_TQ_SOF for convertible top open (automatic transmission)					
ID_FAC_TQ_SOF_THD_CTOP_OPEN_MT	4*7	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDPM_N_SOF	4	0...FFH	0...8160	32	[rpm]
LDPM_GEAR_EF_1	7	0...7H	0...7	1	[-]
Switching threshold FAC_TQ_SOF for convertible top open (manuel transmission)					
IP_T_DLY_SOF	8	0...FFH	0...25.5	0.1	[s]
LDP_N_32	8	0...FFH	0...8160	32	[rpm]
Time delay for closing sound flap					
LC_CONF_CITY_ENA	1	0...1H	0...1	1	[-]
Enable condition regarding LV_CITY (1=on / 0=off)					
LC_STATE_SOF_AST	1	0...1H	0...1	1	[-]
State sound flap during C_T_SOF_AST					
LC_STATE_VS_MIN_SOF_AT	1	0...1H	0...1	1	[-]
State sound flap in case of VS < C_VS_MIN_SOF_AT (automatic transmission)					
LC_STATE_VS_MIN_SOF_MT	1	0...1H	0...1	1	[-]
State sound flap in case of VS < C_VS_MIN_SOF_MT (manuel transmission)					
LC_VAR_SOF_SWI	1	0...1H	0...1	1	[-]
Calibration switch for sport switch (1=on / 0=off)					
LC_VAR_SOF_SWI_AMT	1	0...1H	0...1	1	[-]
Calibration switch for sport switch in AMT vehicles (1=on / 0=off)					

50.4 Sport-switch diagnosis

50.4.1 Input signal (sport switch) range check

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SOF_REQ	V/O	0...1H	0...1	1	[-]
Error flag sport-mode switch request					
LV_CDN_DIAG_SOF_REQ	V/O	0...1H	0...1	1	[-]
Diagnosis condition					
ERR_SYM_SOF_REQ	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error					
LV_END_DIAG_SOF_REQ	V/O	0...1H	0...1	1	[-]
End of Diagnosis					

Input data:

V_SOF_SWI	LV_IGK	CONF_SOF_SWI	LV_VAR_BN

FUNCTION DESCRIPTION:

General information:

This diagnosis is for checking the range of input signal voltage from the sport-mode-switch.

Description:

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Ubatt or line break   (= SYM_0)
 │ │ └───> Short circuit to ground                (= SYM_1)
 │ └─────> -                                      (= SYM_2)
 └───────> -                                      (= SYM_3)
```

Application conditions:

Initialisation: at reset or LV_IGK = 0 → 1 all 0

Recurrence: 10 ms

Activation:

If LV_VAR_BN = 0 **and**

LV_IGK = 1 **and**

CONF_SOF_SWI > 0 **and**

LV_ERR_SOF_REQ = 0

Then LV_CDN_DIAG_SOF_REQ = 1

Else LV_CDN_DIAG_SOF_REQ = 0

```
        Endif
```

Formula section:

Short circuit to ground:

```
If      V_SOF_SWI < C_V_SOF_SWI_MIN    AND
        CONF_SOF_SWI = 1
Then    ERR_SYM_SOF_REQ = SYM_1
        LV_ERR_SOF_REQ = 1     (after debounce)
Endif
```

Short circuit to Ubatt or open line:

```
If      V_SOF_SWI > C_V_SOF_SWI_MAX
Then    ERR_SYM_SOF_REQ = SYM_0
        LV_ERR_SOF_REQ = 1     (after debounce)
Endif
```

Calculation of end of diagnosis:

For calculation of LV_END_DIAG_SOF_REQ see “Anti-bounce – algorithmus, calculation of the end of diagnosis”.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_SOF_SWI_MIN	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Input voltage treshold min					
C_V_SOF_SWI_MAX	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Input voltage treshold max					
C_ABC_INC_SOF_REQ	1	0...FFH	0...255	1	[-]
-					
C_ABC_MAX_SOF_REQ	1	1...FFH	1...255	1	[-]
-					

50.5 Sound flap diagnosis (LV_ERR_SOF)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SOF	V/O	0...1H	0...1	1	[-]
Error flag sound-flap					
LV_CDN_DIAG_SOF	V/O	0...1H	0...1	1	[-]
Diagnosis condition					
ERR_SYM_SOF	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Detected error					
LV_END_DIAG_SOF	V/O	0...1H	0...1	1	[-]
End of Diagnosis					
LV_INH_DIAG_SOF	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition SOF diagnosis					
CDN_DIAG_SOF	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of SOF bit 0 : diagnosis condition for symptom SCP (SYM_0) bit 1 : diagnosis condition for symptom SCG (SYM_1) bit 1 : diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_SOF	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for SOF (only parameter)					

Input data:

LV_CDN_VB_MIN_DIAG	LV_VAR_SOF	LV_IGK	LV_ERR_SPI_MPS

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the sound flap signal from the driver which controls the sound flap. Diagnosis is only done if variant ‘sound flap’ is choosen.

Description:

For error detection algorithm see "Electrical diagnosis of powerstage outputs ATIC39".

(Static control of PIN)

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Vbatt        (= SYM_0)
 │ │ └───> Short circuit to ground       (= SYM_1)
 │ └─────> Open load                     (= SYM_2)
 └───────> -                             (= SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK = 0 → 1 or ECU reset)

Recurrence: 100ms

Activation: LV_VAR_SOF = 1

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_SOF = 1) the diagnosis conditions are set according infrastructure information: CDN_DIAG_SOF.

Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_SOF.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_SOF and ERR_DIAG_SOF

This algorithm determines:

ERR_SYM_SOF	(Detected error symptom)
LV_ERR_SOF	(Error flag for debounced error)
LV_CDN_DIAG_SOF	(Diagnosis condition information)
LV_END_DIAG_SOF	(End of diagnosis information)

Diagnosis inhibition:

```
IF      LV_IGK = 1                      AND
        LV_CDN_VB_MIN_DIAG = 1                AND
        LV_ERR_SPI_MPS = 0
THEN    LV_INH_DIAG_SOF = 0
ELSE    LV_INH_DIAG_SOF = 1
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SOF	1	0...FFH	0...255	1	[-]
Debounce counter increment – SOF diagnosis					
C_ABC_MAX_SOF	1	1...FFH	1...255	1	[-]
Debounce counter maximum value – SOF diagnosis					

50.6 SPI-Bus diagnosis

50.6.1 SPI-Bus diagnosis for Multiple powerstage

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SPI_MPS	V/O	0...1H	0...1	1	[-]
Present failure SPI-bus diagnosis for Multi-powerstage					
LV_CDN_DIAG_SPI_MPS	V/O	0...1H	0...1	1	[-]
Diagnosis condition SPI-bus for Multi-powerstage					
ERR_SYM_SPI_MPS	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH BH CH DH EH FH	NO_SYM SYM_0 SYM_1 SYM_0, SYM_1 SYM_2 SYM_0, SYM_2 SYM_1, SYM_2 SYM_0, SYM_1, SYM_2 SYM_3 SYM_0, SYM_3 SYM_1, SYM_3 SYM_0, SYM_1, SYM_3 SYM_2, SYM_3 SYM_0, SYM_2, SYM_3 SYM_1, SYM_2, SYM_3 SYM_0, SYM_1, SYM_2, SYM_3	0	[-]
Detected symptom SPI-busdiagnosis for Multi-powerstage					
LV_END_DIAG_SPI_MPS	V/O	0...1H	0...1	1	[-]
End of diagnosis SPI-bus for Multi-powerstage					

Input data:

LV_IGK			

Description:

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
│ │ │ └──> -              (= SYM_0)
│ │ └────> -              (= SYM_1)
│ └──────> SPI bus off    (=SYM_2)
└────────> -              (=SYM_3)
```

Application conditions:

Initialization :	all 0 at LV_IGK 0->1 or reset
Activation:	*at every engine states*
Recurrency :	10 ms

Formula section:

Diagnosis conditon:

LV_CDN_DIAG_SPI_MPS = 1

Error detection

```
IF      BSW detects SPI bus error
THEN    ERR_SYM_SPI_MPS = SYM_2      //LV_ERR_SPI_MPS is set after debounce
ELSE    ERR_SYM_SPI_MPS = NO_SYM     //LV_ERR_SPI_MPS is reset after rebounce
ENDIF
```

End of diagnosis:

LV_END_DIAG_SPI_MPS is calculated by error management if diagnosis condition is active.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SPI_MPS	1	0...FFH	0...255	1	[-]
Anti bounce increment SPI_MPS diagnosis					
C_ABC_MAX_SPI_MPS	1	1...FFH	1...255	1	[-]
Anti bounce maximum SPI_MPS diagnosis					

50.6.2 SPI-Bus diagnosis for knock sensor

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_SPI_KNK	V/O	0...1H	0...1	1	[-]
Present failure SPI-bus diagnosis for Knock sensor					
LV_CDN_DIAG_SPI_KNK	V/O	0...1H	0...1	1	[-]
Diagnosis condition SPI-bus for Knock sensor					
ERR_SYM_SPI_KNK	V/O	0H	NO_SYM	0	[-]
		1H	SYM_0		
		2H	SYM_1		
		3H	SYM_0, SYM_1		
		4H	SYM_2		
		5H	SYM_0, SYM_2		
		6H	SYM_1, SYM_2		
		7H	SYM_0, SYM_1, SYM_2		
		8H	SYM_3		
		9H	SYM_0, SYM_3		
		AH	SYM_1, SYM_3		
		BH	SYM_0, SYM_1, SYM_3		
		CH	SYM_2, SYM_3		
		DH	SYM_0, SYM_2, SYM_3		
		EH	SYM_1, SYM_2, SYM_3		
		FH	SYM_0, SYM_1, SYM_2, SYM_3		
Detected symptom SPI-busdiagnosis for Knock sensor					
LV_END_DIAG_SPI_KNK	V/O	0...1H	0...1	1	[-]
End of diagnosis SPI-bus for Knock sensor					

Input data:

LV_IGK	N_32		

Description:

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -              (= SYM_0)
 │ │ └───> -              (= SYM_1)
 │ └─────> SPI bus off    (=SYM_2)
 └───────> -              (=SYM_3)
```

Application conditions:

Initialization : all 0 at LV_IGK 0->1 or reset

Activation: N_32 > 320 (1/min)

Recurrency : 100 ms

Formula section:

Diagnosis conditon:

LV_CDN_DIAG_SPI_KNK = 1

Error detection

```
IF      BSW detects SPI bus error
THEN    ERR_SYM_SPI_KNK = SYM_2       //LV_ERR_SPI_KNK is set after debounce
ELSE    ERR_SYM_SPI_KNK = NO_SYM      //LV_ERR_SPI_KNK is reset after rebounce
ENDIF
```

End of diagnosis:

LV_END_DIAG_SPI_KNK is calculated by error management if diagnosis condition is active.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_SPI_KNK	1	0...FFH	0...255	1	[-]
Anti bounce increment SPI_KNK diagnosis					
C_ABC_MAX_SPI_KNK	1	1...FFH	1...255	1	[-]
Anti bounce maximum SPI_KNK diagnosis					

50.7 Cus adap module: COMR

50.7.1 VVTI interface adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Vse_btsoll	V/O	0...FFFFH	-128...52300	0.8	[°CRK]
Cam setpoint for inlet cam via external service device (Tester)					
B_btvse	V/O	0...1H	0...1	1	[-]
condition for cam setpoint for inlet cam via external service device (Tester)					
Vsa_btsoll	V/O	0...FFFFH	-128...52300	0.8	[°CRK]
Cam setpoint for exhaust cam via external service device (Tester)					
B_btvsa	V/O	0...1H	0...1	1	[-]
condition for cam setpoint for exhaust cam via external service device (Tester)					
Pbrems	V/O	0...FFH	0...99.60937	0.390625	[bar]
brake pressure					

Input data:

CAM_SP_IN_EXT_ADJ	CAM_SP_EX_EXT_ADJ	LV_CAM_SP_IN_EXT_ADJ	LV_CAM_SP_EX_EXT_ADJ
ECU_STATE	BRAKE_PRS		

FUNCTION DESCRIPTION:

Adaption to BMW environment.

Remark: The really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset: 0, except :
Vse_btsoll = 0 °crk
Vsa_btsoll = 0 °crk

Recurrence: 100ms: Pbrems

10 ms: all other variables

Activation: at every engine operating state

Deactivation: ---

Formula section:

```
If ECU_STATE = "PWL" then

    Vse_btsoll   = 0 °crk
    Vsa_btsoll   = 0 °crk
    B_btvse      = 0
    B_btvsa      = 0

elseif
   Inlet CAM:          Vse_btsoll = CAM_SP_IN_EXT_ADJ
                       B_btvse = LV_CAM_SP_IN_EXT_ADJ
   Exhaust CAM:        Vsa_btsoll = |CAM_SP_EX_EXT_ADJ|
                       B_btvsa = LV_CAM_SP_EX_EXT_ADJ

    brake pressure:      Pbrems = BRAKE_PRS
endif
```

50.7.2 CAN interface adaptation - Outputs to BMW- environment

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
A_quer	O/V	8000...7FFFH	-64...63.99804	1.9531e-3	[m/s²]
Vehicle acceleration transversal					
Acfzgl	O/V	80...7FH	-32...31.75	0.25	[m/s²]
vehicle acceleration longitudinal					
Acfzgq	O/V	80...7FH	-32...31.75	0.25	[m/s²]
Fahrzeugquerbeschleunigung					
B_abliactiv	O/V	0...1H	0...1	1	[-]
state of low beam: 0 = off, 1 = on					
B_anhang	O/V	0...1H	0...1	1	[-]
information of trailer					
B_ckl30	O/V	0...1H	0...1	1	[-]
condition kl30					
B_ckl50	O/V	0...1H	0...1	1	[-]
condition terminal ¿50 ¿(via CAN) active					
B_dscakt	O/V	0...1H	0...1	1	[-]
DSC Aktivität					
B_dzm_dyn	O/V	0...1H	0...1	1	[-]
Anforderung Drehzahlmesserdynamisierung					
B_egs_khl1	O/V	0...1H	0...1	1	[-]
condition for request of cooling stage 1 frorm gear box					
B_egs_khl2	O/V	0...1H	0...1	1	[-]
condition for request of cooling stage 2 frorm gear box					
B_fagurt	O/V	0...1H	0...1	1	[-]
Fahrergurtschloß geschlossen					
B_gangwechsel_gs	O/V	0...1H	0...1	1	[-]
Gangwechsel aktiv					
B_gen_ext	O	0...1H	0...1	1	[-]
status for external adjustment generator					
B_hz	O/V	0...1H	0...1	1	[-]
Increased engine heating power request by AC / INSTR3					
B_iliactiv	O/V	0...1H	0...1	1	[-]
state of interior lighting: 0 = off, 1 = on					
B_keinhs_gs	O/V	0...1H	0...1	1	[-]
Wenn=0 wird die Maxdrehzahl mit Prädiktion abgeregelt					
B_ldm_ena	O	0...1H	0...1	1	[-]
status for required torque via LDM enabled					
B_ldm_nofil	O/V	0...1H	0...1	1	[-]
Fast torque adjustment by LDM					
B_lklps_kl1	O/V	0...1H	0...1	1	[-]
Anforderung LKLPS "auf" von IHKA					
B_lklps_kl2	O/V	0...1H	0...1	1	[-]
Anforderung LKLPS "ganz auf" von IHKA					
B_msadltgpd	O/V	0...1H	0...1	1	[-]
Diagnose MSA-Leitung: Bit für die peramanente Deaktivierung MSA					
B_msadltgtd	O/V	0...1H	0...1	1	[-]
Diagnose MSA-Leitung: Bit für die temporäre Deaktivierung MSA					
B_nosa	O/V	0...1H	0...1	1	[-]
Fuel cutoff prohibition by LDM					
B_schalt_ldm	O/V	0...1H	0...1	1	[-]
Anforderung Schalthinweis über CC-Meldung vom LDM					
B_taleer	O/V	0...1H	0...1	1	[-]
Bedingung Tank leer					
B_tstkvs	O/V	0...1H	0...1	1	[-]
Bedingung Funktionsanforderung Kraftstoffsystemdiagnose					
Batt_class	O/V/S	0...FFH	0...255	1	[-]

Classification of the vehicle battery					
Baw_tester	O/V	0...FFFFH	0...65535	1	[-]
Testeranforderung von Zulieferer					
Bordnetz	O/V	1...2H	1...2	1	[-]
=1: CAN 11Hex =2: CAN 12Hex; bordnetz2000 is present					
Bosfxid2r	O/V/S	0...FFFFH	0...65535	1	[-]
ID_Funktion_BOS_Rückstellung_2					
Bosid2r	O/V/S	0...FFH	0...255	1	[-]
ID_2_BOS_Rückstellung_2					
Bosjzielr	O/V/S	0...FFH	0...255	1	[-]
Zieltermin_Monat_BOS_Rückstellung_2					
Bosmzielr	O/V/S	0...FFH	0...255	1	[-]
Zieltermin_Jahr_BOS_Rückstellung_2					
Bosprog2r	O/V/S	0...FFH	0...255	1	[-]
Prognose_Intervall_Zeit_BOS_Rückstellung_2					
Bosrlsmr	O/V	8000...7FFFH	-327680... 327670	10	[km]
Remaining milage for oil (from combi)					
Bosrw2r	O/V/S	0...FFH	0...255000	1000	[km]
Prognose_Intervall_Weg_BOS_Rückstellung_2					
Bosun	O/V	0...FFH	0...255	1	[-]
Einheit für Bosrlsmr					
Bosunt	O/V	0...FFH	0...255	1	[-]
Einheit für Bosrlsmt					
Bosv	O/V	0...FFH	0...255	1	[-]
availability of kombi					
Dm_ab_fws	O/V	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Fahrwiderstand					
I_eff_gs	O/V	0...FFFFH	0...511.992	0.0078125	[1/m]
Strangverstärkung					
Id_bosrst	O/V	0...FFH	0...255	1	[-]
ID2_BOS_resetting					
Id_bosrtw	O/V	0...FFFFH	0...65535	1	[-]
ID_function_BOS_resetting					
Ikurz_ogr	O/V/S	0...FFH	0...2550	10	[A]
maximum short-current from CAS (vehicle-specific)					
Ikurz_ugr	O/V/S	0...FFH	0...2550	10	[A]
minimum short-current from CAS (vehicle-specific)					
Ipm_typ_fp	O/V	0...FFH	0...255	1	[-]
iPM-Fahrertyp zur Beeinflussung der Pedalauswertung					
Ipm_typ_mdkw	O/V	0...FFH	0...255	1	[-]
iPM-Fahrertyp zur Beeinflussung der Soll-Momentenbestimmung					
Iwakeupgr	O/V/S	0...FFH	0...0.255	0.001	[A]
No-load-current (vehicle-specific)					
Km_st	O/V	0...FFFFH	0...655350	10	[km]
Kilometer reading at engine start based on last DC					
Km_st_1	O/V/S	0...FFFFFFFFH	0...4294967295	1	[-]
Wegstrecke_km auf 1 km genau					
Kvaverbr	O/V	0...FFFFH	0...65535	1	[µl]
fuel consumption [micro liter]					
La_soll_tester[NC_CBK_EX_NR]	O/V	0...FFFFH	0...15.99975	0.2441e-3	[-]
Lambdaanforderung Tester					
Md_kupp_schalt	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Kupplungsaufnahmemoment					
Md_rad_asrl	O/V	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Momentenbegrenzung ASR langsamer Pfad					
Md_rad_asrs	O/V	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Momentenbegrenzung ASR schneller Pfad					

Md_rad_msrl	O/V	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Begrenzungsmoment aus Motorschleppmomentenregelung langsamer Pfad					
Md_rad_msrs	O/V	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Begrenzungsmoment aus Motorschleppmomentenregelung schneller Pfad					
Md_rad_soll	O/V	80000000... 7FFFFFFFH	-214748364.8... 214748364.7	0.1	[Nm]
Soll-Radmoment					
N_ab_m	O/V	8000...7FFFH	-32768...32767	1	[rpm]
gemessene Abtriebsdrehzahl					
N_idlms	O/V	0...FFH	0...255	1	[-]
request idlespeed-increase from powermanagement					
N_rad_hl	O/V	8000...7FFFH	-512...511.98437	0.015625	[rad/s]
Raddrehzahl hnten links					
N_rad_hr	O/V	8000...7FFFH	-512...511.98437	0.015625	[rad/s]
Raddrehzahl hnten rechts					
N_rad_vl	O/V	8000...7FFFH	-512...511.98437	0.015625	[rad/s]
Raddrehzahl vorn links					
N_rad_vr	O/V	8000...7FFFH	-512...511.98437	0.015625	[rad/s]
Raddrehzahl vorn rechts					
Nelueft_klima	O/V	0...FFH	0...99.60937	0.390625	[%]
Anforderung E-Lüfter Drehzahl von IHKA als PWM-Signal					
Nelueft_lenk	O/V	0...FFH	0...99.60937	0.390625	[%]
E-Lüfteranforderung von derLenkhilfe					
Nturb	O/V	8000...7FFFH	-32768...32767	1	[rpm]
primary gear box revolution					
Pmsv[15]	O/V/S	0...FFH	0...255	1	[-]
powermanagement ¿Standverbraucher¿					
Prg_getr	O/V	0...FH	0...15	1	[-]
State mode ETCU					
Qs_anf_klima	O/V	0...FFH	0...99.60937	0.390625	[%]
heat flow request					
St_dsc_can	O/V	0...FFH	0...255	1	[-]
Status DSC					
St_dsc_mradsoll	O/V	0...FFH	0...255	1	[-]
Sollwert Qualifier zum Signal Soll_Radmoment_Antriebsstrang_Summe_Stabilisierung					
St_fas_mradsoll	O/V	0...FFFFH	0...65535	1	[-]
Sollwert Qualifier zum Signal Soll_Radmoment_Antriebsstrang_Summe_FAS					
St_fzg_ass	O/V	0...FH	0...15	1	[-]
Status ldm via CAN					
St_kupp_smg	O/V	0...FFH	0...255	1	[-]
Status Kupplung SMG-Getriebe					
St_wk	O/V	0...FFH	0...255	1	[-]
Status converter clutch					
T_jahr	O/V	0...FFFFH	0...65535	1	[-]
Borddatum, Jahr					
T_monate	O/V	0...FFFFH	0...65535	1	[-]
Borddatum, Monat					
T_tag	O/V	0...FFFFH	0...65535	1	[-]
Borddatum, Tag					
U_gen_ext	O/V	0...FFFFH	0...6553.5	0.1	[V]
adjustment value for alternator voltage setpoint					
V_can	O/V	0...FFFFH	0...6553.5	0.1	[km/h]
Vehicle speed from CAN					
V_krapu	O/V	0...FFH	0...255	1	[l/h]
Required amount of fuel					
V_rad	O/V	0...FFFFH	0...6553.5	0.1	[km/h]
Fahrzeuggeschwindigkeit vom DSC-Raddrehzahl					
Zrbosr	O/V	0...FFH	0...255	1	[-]
counter_resetting_BOS_resetting					

Ztageabs	O/V	0...FFFFH	0...65535	1	[-]
day counter absolute (counts the days at sufficient mains voltage)					

Input data:

AC_VEH_LGT_TCS	AC_VEH_TRV_TCS	Bosconf	CONF_SWI_EFP_OUT
CTR_KM_BN	CTR_KM_CAN	CUR_SC_MAX_CAN	CUR_SC_MIN_CAN
CUR_WKU_CAN	DIST_RESI_OIL_KM	dm_ab_fws	ECU_STATE
EFPPWM_CAN	FAC_GB_GAIN	FAC_IS_INC_REQ	FAC_POW_MNG_VST_CNS[15]
FCO	HEAT_REQ_PERC	idbosrst	idfbosrt_w
IDX_BAT_CAN	LAMB_SP_EXT_ADJ[NC_CBK_EX_NR]	LC_EFPPWM_CTL	lpm_typ_fp
lpm_typ_mdkw	LV_ACT_ECRAS_EXT_ADJ	LV_DRIV_BELT_CLOSE	LV_GS
LV_GS_IDC_LDM	LV_KEY_AUX	LV_KEY_AUX	LV_LDM_ENA
LV_LTG_HDLP_L_ON	LV_LTG_INL_ON	LV_N_DISP_DYN	LV_N_REL_CWP_SP_EXT_ADJ
LV_REQ_HEAT	LV_RLY_ST_CAN	LV_STATE_TRL	LV_TCS_CTL_ACT
LV_V_ALTER_SP_EXT_ADJ	LV_VAR_BN	LV_VAR_BN	N_ECF
N_ECF_CAN	N_GB	N_GB_OUT	N_WHEEL_FN_LE
N_WHEEL_FN_RI	N_WHEEL_RE_LE	N_WHEEL_RE_RI	OPM_EXT_REQ
QOIL_DS_RST_CAN_1_5	QOIL_DS_RST_CAN_2_3	QOIL_DS_RST_CAN_2_5	QOIL_DS_RST_CAN_2_6
QOIL_DS_RST_KWP_9	STATE_CC	STATE_CLU_AMT	STATE_DI_PUC
STATE_LDM	STATE_SP_DYN_WHEEL	STATE_TCS_CAN	STATE_TEMP_GB
STATE_TQ_WHEEL_DRIVASI	STATE_TQ_WHEEL_TCS_SLOW	T_CAN	T_CLK_ICL_DISP_1
T_CLK_ICL_DISP_2	T_CLK_ICL_DISP_3	TQ_SP_WHEEL	TQ_WHEEL_TCS_FAST
TQ_WHEEL_TCS_SLOW	V_ALTER_SP_EXT_ADJ	vbost	VFF_EFP
VS_H	zrbosr	LV_IGK_OFF_ACK_PERM_DEAC	LV_IGK_OFF_ACK_TMP_DEAC
STATE_ETCU_PROG_INFO	WHEEL		

FUNCTION DESCRIPTION:

Adaption to BMW environment.

Application conditions:

Initialisation at reset or at exit power latch phase:

0: B_dscakt, B_dzm_dyn, B_gangwechsel_gs, B_keinhs_gs, B_hz, B_iliactiv, B_abliactiv,Qs_anf_klima, Ztageabs, B_ckl50, B_ckl30, B_ldm_nofil, B_nosa, Acfzgq, Acfzgl, l_eff_gs, T_jahr, T_monate, T_tag, Md_rad_soll, Nelueft_klima, Nelueft_lenk, St_wk, N_idlms, Bosrlsmr B_ldm_ena, B_gen_ext, V_can, A_quer, B_tstkvs, St_fzg_ass, Kvaverbr, B_egs_khl1, B_egs_khl2, Bosv, Nturb, B_anhang, St_kupp_smg, V_krapu,Bosanfsgbd, Bosun, Bosunt , B_taleer, Dm_ab_fws, lpm_typ_fp, lpm_typ_mdkw, Baw_tester, Md_kupp_schalt, N_rad_hl, N_rad_hr, N_rad_vl, N_rad_vr, St_dsc_mradsoll, St_fas_mradsoll, N_ab_m, B_schalt_ldm, B_fagurt, St_dsc_can, B_msadltgpd, B_msadltgtd, Prg_getr, V_rad

1: B_lklps_kl1, B_lklps_kl2,

1.0: La_soll_tester[NC_CBK_EX_NR]

32000 Nm: Md_rad_asrl, Md_rad_asrs

-32000 Nm: Md_rad_msrl, Md_rad_msrs

Ikurz_ogr = value out of non-volatile memory after reset
Ikurz_ugr = value out of non-volatile memory after reset
Iwakeupgr = value out of non-volatile memory after reset
Bosfxid2r = value out of non-volatile memory after reset
Bosid2r = value out of non-volatile memory after reset
Bosprog2r = value out of non-volatile memory after reset
Bosrw2r = value out of non-volatile memory after reset
Km_st_1 = value out of non-volatile memory after reset
Bosjzielr = value out of non-volatile memory after reset
Bosmzielr = value out of non-volatile memory after reset
Batt_class = value out of non-volatile memory after reset
U_gen_ext = first calculated value

```
Pmsv[15] = FAC_POW_MNG_VST_CNS[15]      //from non volatile memory

if    LV_VAR_BN = 1
then  Bordnetz = 2
else  Bordnetz = 1
endif
```

special values at power latch phase:

0: B_dscakt, B_dzm_dyn, _gangwechsel_gs, B_keinhs_gs, B_hz, I_eff_gs, Md_rad_soll, Nelueft_klima, St_wk, A_quer, B_taleer, B_tstkvs, B_ldm_nofil, B_nosa, Bosrlsmr, V_krapu, Nturb, Baw_tester, Md_kupp_schalt, N_rad_hl, N_rad_hr, N_rad_vl, N_rad_vr, St_dsc_mradsoll, St_fas_mradsoll, N_ab_m, B_msadltgpd, B_msadltgtd, Prg_getr
1: B_lklps_kl1, B_lklps_kl2, B_fagurt, St_dsc_can
last valid value: B_gen_ext, U_gen_ext
last calculated value: Acfzgl, Acfzgq

1.0: La_soll_tester[NC_CBK_EX_NR]
32000 Nm: Md_rad_asrl, Md_rad_asrs

-32000 Nm: Md_rad_msrl, Md_rad_msrs

Recurrence: **1000ms**: Bosv, Qs_anf_klima, Ztageabs, Acfzgq, Acfzgl, Bosfxid2r, Bosid2r, Bosprog2r, Km_st_1, Bosjzielr, Bosmzielr, Bosrw2r, T_jahr, T_monate, T_tag, B_anhang, Bosrlsmr, Zrbosr, Bosun
Bosunt

200ms: B_hz , Km_st, Kvaverbr,B_lklps_kl1, B_lklps_kl2
100ms: B_dscakt, Ikurz_ogr, Ikurz_ugr, Iwakeupgr, Batt_class, B_iliactiv,B_abliactiv, B_ckl50, Nelueft_klima, Nelueft_lenk, B_gen_ext, U_gen_ext, B_taleer, B_tstkvs, Ikurz_ogr, Ikurz_ugr, Iwakeupgr, N_idlms, Prg_getr

10ms: B_dzm_dyn, B_gangwechsel_gs, B_keinhs_gs, B_egs_khl1, B_egs_khl2, Pmsv[15], B_ldm_nofil, B_nosa, I_eff_gs, Md_rad_soll, V_krapu, Nturb, St_wk, B_ldm_ena, V_can, A_quer, St_fzg_ass, St_kupp_smg, La_soll_tester[NC_CBK_EX_NR], Baw_tester, Dm_ab_fws, Ipm_typ_fp, Ipm_typ_mdkw, Md_rad_asrl, Md_rad_asrs, Md_rad_msrl, Md_rad_msrs, N_ab_m, N_rad_hl, N_rad_hr, N_rad_vl, N_rad_vr, St_dsc_mradsoll, St_fas_mradsoll, B_schalt_ldm, B_fagurt, St_dsc_can, B_msadltgpd, B_msadltgtd, V_rad

Note: B_hz has to be updated directly after LV_REQ_HEAT has been calculated.

Activation: After reset

Deactivation: After power latch phase

Formula section:

Prg_getr = STATE_ETCU_PROG_INFO
B_msadltgpd = LV_IGK_OFF_ACK_PERM_DEAC
B_msadltgtd = LV_IGK_OFF_ACK_TMP_DEAC
B_dscakt = LV_TCS_CTL_ACT
B_dzm_dyn = LV_N_DISP_DYN
B_fagurt = LV_DRIV_BELT_CLOSE
B_gangwechsel_gs = LV_GS
B_hz = LV_REQ_HEAT
B_keinhs_gs = 0
Km_st = CTR_KM_CAN (update every 200ms)
Kvaverbr = FCO (low word)
Md_rad_asrl = TQ_WHEEL_TCS_SLOW
Md_rad_asrs = TQ_WHEEL_TCS_FAST
Md_rad_msrl = TQ_WHEEL_TCS_SLOW
Md_rad_msrs = TQ_WHEEL_TCS_FAST
N_ab_m = N_GB_OUT
N_rad_hl = N_WHEEL_RE_LE
N_rad_hr = N_WHEEL_RE_RI
N_rad_vl = N_WHEEL_FN_LE
N_rad_vr = N_WHEEL_FN_RI
St_dsc_mradsoll = STATE_TQ_WHEEL_TCS_SLOW
St_fas_mradsoll = STATE_TQ_WHEEL_DRIV_ASI

B_iliactiv = LV_LTG_INL_ON

```
B_abliactiv =        LV_LTG_HDLP_L_ON

Zrbosr        = zrbosr

if  STATE_TEMP_GB = 1H
then      B_egs_khl1 = 1
else      B_egs_khl1 = 0
endif

if  STATE_TEMP_GB = 2H
then      B_egs_khl2 = 1
else      B_egs_khl2 = 0
endif

if        LV_VAR_BN = 1
then      Bosv = vbost
else      Bosv = 0
endif

Qs_anf_klima         = HEAT_REQ_PERC

Nturb                = N_GB

B_anhang             = LV_STATE_TRL

Pmsv[15]             = FAC_POW_MNG_VST_CNS[15]

Ztageabs             = T_CAN

B_ckl50              = LV_RLY_ST_CAN

B_ckl30              = LV_KEY_AUX

If           Bit 1 or Bit 2 of  STATE_SP_DYN_WHEEL is set to 1
then         B_ldm_nofil = 1
else         B_ldm_nofil = 0
endif

If           STATE_DI_PUC = 1
then         B_nosa = 1
else         B_nosa = 0

endif

Acfzgl               =        AC_VEH_LGT_TCS

Acfzgq               =        AC_VEH_TRV_TCS

Bosfxid2r            =        QOIL_DS_RST_CAN_2_2

Bosid2r              =        QOIL_DS_RST_CAN_2_1

Bosrw2r              =        QOIL_DS_RST_CAN_2_3

I_eff_gs             =        FAC_GB_GAIN

Km_st_1              =        CTR_KM_BN

Bosjzielr            =        QOIL_DS_RST_CAN_2_5

Bosmzielr            =        QOIL_DS_RST_CAN_2_4

Bosprog2r            =        QOIL_DS_RST_CAN_2_6

T_jahr                        =        T_CLK_ICL_DISP_3
```

T_monate = T_CLK_ICL_DISP_2
T_tag = T_CLK_ICL_DISP_1
Md_rad_soll = TQ_SP_WHEEL

Nelueft_klima = (N_ECF/14) * 100%
Nelueft_lenk = N_ECF_CAN
St_wk = STATE_CC
Bosrlsmr = QOIL_DS_RST_CAN_1_5
B_ldm_ena = LV_LDM_ENA
B_gen_ext = LV_V_ALTER_SP_EXT_ADJ
V_can = VS_H
A_quer = AC_VEH_TRV_TCS
U_gen_ext = V_ALTER_SP_EXT_ADJ
B_tstkvs = LC_REQ_FSD_EOL
B_lklps_kl1 = 1
B_lklps_kl2 = 1
Batt_class = IDX_BAT_CAN
Ikurz_ogr = CUR_SC_MAX_CAN
Ikurz_ugr = CUR_SC_MIN_CAN
Iwakeupgr = CUR_WKU_CAN
Id_bosrtw = idfbosrt_w
Id_bosrst = idbosrst
N_idlms = FAC_IS_INC_REQ
St_fzg_ass = STATE_LDM
St_kupp_smg = STATE_CLU_AMT

Bosun = DIST_RESI_OIL_KM
Bosunt = QOIL_DS_RST_KWP_9

```
//    V_krapu        =  RQAM_FU,  because RQAM_FU is not exported from CAN-spec. use:
If(1)         LV_VAR_BN = 1
Then(1)
              // for  BN200=, taken from 17400J03.00x:
     If            CONF_SWI_EFP_OUT       =       !1
     Then
              If            LC_EFPPWM_CTL = 0
              Then          V_krapu       =       VFF_EFP
              Else    V_krapu       =       EFPPWM_CAN
     Endif
Else(1)
              // for CAN11, taken from 17400J02.00x:
              V_krapu                 = VFF_EFP
```

```
Endif(1)

Dm_ab_fws                  = dm_ab_fws
Ipm_typ_fp                 = ipm_typ_fp
Ipm_typ_mdkw               = ipm_typ_mdkw
La_soll_tester[NC_CBK_EX_NR] = LAMB_SP_EXT_ADJ[NC_CBK_EX_NR]
Baw_tester                 = OPM_EXT_REQ

if        (LV_FTL_CAN_ERR==0 and LV_FTL_OBD_INH_L==1)
then      B_taleer = 1
else      B_taleer = 0
endif


B_schalt_ldm       = LV_GS_IDC_LDM
St_dsc_can         = STATE_TCS_CAN
V_rad              = WHEEL
```

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_REQ_FSD_EOL	1	0...1H	0...1	1	[-]
Switch for external adjustment of fuel-system diagnosis					

50.7.3 CAN interface adaptation - Inputs from BMW- environment

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
absch_korr	O/V/S	8000...7FFFH	-50...49.99847	1.5259e-3	[-]
Korrekturwert Abschaltung					
CUR_RNG_CTL	O/V	0...FFH	0...255	1	[-]
Control current path					
GR_DT	O/V	0...FFH	0...31.875	0.125	[-]
Gear ratio factor for monitoring					
GS_IDC_DISP_RAW	O/V	0...FFH	0...255	1	[-]
Gearshift signal - display type					
GS_IDC_GEAR_RAW	O/V	0...FFH	0...255	1	[-]
Gearshift signal - setpoint for gear					
idfbosmg_w	O/V	0...FFFFH	0...65535	1	[-]
ID for engine oil (BOS Kombi)					
LV_CAN_SND_MSG_PWR_MNG_0	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung POWERMANAGMENT VERBRAUCHERSTEUERUNG Sendeanforderung"					
LV_CAN_SND_MSG_PWR_MNG_1	O/V	0...1H	0...1	1	[-]
Flag indicating "Bedingung POWERMANAGMENT BATTERIESPANNUNG Sendeanforderung"					
LV_GS_DOWN	O/V	0...1H	0...1	1	[-]
Gear-shift down					
LV_GS_UP	O/V	0...1H	0...1	1	[-]
Gear-shift up					
LV_INH_STST_CDN	O/V	0...1H	0...1	1	[-]
Condition for Start-Stop					
LV_LDM_DRIV_ACT	O/V	0...1H	0...1	1	[-]
Condition LDM driver active					
LV_LDM_OFF	O/V	0...1H	0...1	1	[-]
LDM off requested by DME					
LV_OIL_CNS_WARN_1	O/V	0...1H	0...1	1	[-]
LV warning oil consumption					
LV_OIL_CNS_WARN_2	O/V	0...1H	0...1	1	[-]
LV warning oil consumption					
LV_STST_STOP	O/V	0...1H	0...1	1	[-]
flag indicating engine is in STST(MSA)-Stop					
N_DISP_DYN	O/V	0...1FE0H	0...8160	1	[rpm]
dynamisated N for display in ICL (Kombi)					
POW_CTL_PARK_CNS	O/V	0...FFH	0...255	1	[-]
Control park consumer					
POW_CTL_PRI_PEAK_RED	O/V	0...FFH	0...255	1	[-]
Control priority peak reduction					
POW_CTL_PRI_PEAK_RED_CFT	O/V	0...FFH	0...255	1	[-]
Control priority peak reduction comfort					
POW_CTL_PWR_CNS_1	O/V	0...FFFFH	0...65535	1	[-]
Control power consumer					
POW_CTL_PWR_CNS_2	O/V	0...FFH	0...127.5	0.5	[%]
Control power special consumer					
rqpcos	O/V	0...FFH	0...255	1	[-]
Request park consumer					
selspcos	O/V	0...FFH	0...255	1	[-]
Selection special consumers					
st_ldstgen	O/V	0...FFH	0...255	1	[-]
generator load					
STATE_EGY_CNS_OFF	O/V	0...FFH	0...255	1	[-]
state all electrical consumers off					
STATE_ENGG_POS	O/V	0...FFH	0...255	1	[-]
Status Kraftschluss					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_TQ_WHEEL	O/V	0...FFH	0...255	1	[-]
state torque wheel					
STATE_TQ_WHEEL_DRIV_ASI_AV	O/V	0...FFH	0...255	1	[-]
actual state of TQ_WHEEL_DRIV_ASI functionality (get from layer)					
STATE_TQ_WHEEL_TCS_AV	O/V	0...FFH	0...255	1	[-]
Actual state of TQ_Wheel_TCS functionality (get from Layer)					
stpcos	O/V	0...FFH	0...255	1	[-]
Status park consumers					
TQ_ECU_ETCU	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Actual torque ECU/ETCU					
TQ_MAX_ACC	O/V	0...FFH	0...127.5	0.5	[Nm]
air-conditioning-compressor limit torque					
TQ_MAX_WHEEL	O/V	0...FFFFH	-32000...33535	1	[Nm]
Max. Radmoment					
TQ_MIN_WHEEL_L	O/V	0...FFFFH	-32000...33535	1	[Nm]
Schleppmoment unbefeuert					
TQ_WHEEL	O/V	0...FFFFH	-32000...33535	1	[Nm]
Ist-Moment an den Antriebsrädern					

Input data:

Absch_korr	B_hschalt_komb	B_ldm_akt04	B_ldm_off
B_lklps	B_mol_vb	B_mol_vl	B_msaccid2
B_msastops	B_pmbsdanf	B_pmvsdanf	B_rschalt_komb
B_rschalt_komb	B_verb_off	Ctrcbr	Ctrpcos
Ctrprio	Ctrpriocf	Ctrpwrcos	Ctrpwspcos
I_ges	Id_bosmg	LV_VAR_DCC	Md_can_dmee
Md_max_klima	Md_rad_ist	Md_rad_max	Md_rad_min
N_dzm	Rqpcos	Selspcos	Spa_art
Spa_gang	St_dsc_mradist	St_fas_mradist	St_ldm_kupp
St_ldstgen	St_mdinfo_ges	STATE_ACK_IGK_OFF	Stpcos
STATE_IGK_HW			

FUNCTION DESCRIPTION:

Adaption to BMW- environment

Application conditions:

Initialisation: 0

Except:

st_ldstgen = 2

absch_korr: from nonvolatile memory

Recurrence: 100 ms except:

1000 ms: idfbosmg_w, absch_korr, LV_OIL_CNS_WARN_1, LV_OIL_CNS_WARN_2

10 ms: St_mdinfo_ges, TQ_WHEEL, TQ_MAX_WHEEL, TQ_MIN_WHEEL_L, STATE_ENGG_POS, LV_LDM_OFF, LV_GS_UP, LV_GS_DOWN, LV_LDM_DRIV_ACT, GR_DT, STATE_TQ_WHEEL, TQ_MAX_ACC, STATE_TQ_WHEEL_DRIV_ASI_AV, STATE_TQ_WHEEL_TCS_AV, LV_INH_STST_CDN GS_IDC_GEAR_RAW, GS_IDC_DISP_RAW, TQ_ECU_ETCU, N_DISP_DYN, LV_STST_STOP

Activation: After reset

***Except**:*

0 : St_mdinfo_ges, TQ_WHEEL, TQ_MAX_WHEEL, TQ_MIN_WHEEL_L, STATE_ENGG_POS

Value at PWL

0: LV_LDM_OFF, LV_GS_UP, LV_GS_DOWN, LV_LDM_DRIV_ACT, GR_DT, STATE_TQ_WHEEL GS_IDC_GEAR_RAW, GS_IDC_DISP_RAW, TQ_ECU_ETCU

Last calculated value: LV_OIL_CNS_WARN_1, LV_OIL_CNS_WARN_2

ECU_STATE = WAKE_UP: absch_korr

Deactivation: After power latch phase

Formula section:

All formulas are valid in a physical meaning!

TQ_MAX_ACC = Md_max_klima
LV_OIL_CNS_WARN_1 = B_mol_vb
LV_OIL_CNS_WARN_2 = B_mol_vl
CUR_RNG_CTL = Ctrcbr
POW_CTL_PARK_CNS = Ctrpcos
POW_CTL_PRI_PEAK_RED = Ctrprio
POW_CTL_PRI_PEAK_RED_CFT = Ctrpriocf
POW_CTL_PWR_CNS_1 = Ctrpwrcos
POW_CTL_PWR_CNS_2 = Ctrpwspcos
idfbosmg_w = Id_bosmg
rqpcos = Rqpcos
selspcos = Selspcos
stpcos = Stpcos

```
st_ldstgen                      =       St_ldstgen
absch_korr                      =       Absch_korr

IF          B_verb_off = 1
THEN        STATE_EGY_CNS_OFF = 1
ELSE        STATE_EGY_CNS_OFF = 0
ENDIF

TQ_WHEEL                        =       Md_rad_ist
STATE_TQ_WHEEL                  =       St_mdinfo_ges
TQ_MAX_WHEEL                    =       Md_rad_max
TQ_MIN_WHEEL_L                  =       Md_rad_min
STATE_ENGG_POS                  =       St_ldm_kupp


If          B_pmvsdanf = 1
Then        LV_CAN_SND_MSG_PWR_MNG_0        =       B_pmvsdanf
endif


//check if transition in STATE_ACK_IGK_OFF, if yes set LV_CAN_SND_.._1 = 1 for 3
recurrencies (300ms):
    If            STATE_ACK_IGK_OFF 0->1
               or STATE_IGK_HW 2->0
               or STATE_IGK_HW 1->0
               or STATE_IGK_HW 1->2
    Then        counter_ack = 3
    Else        counter_ack = counter_ack -1
    Endif

    If          B_pmbsdanf = 1       or      counter_ack > 0
    Then        LV_CAN_SND_MSG_PWR_MNG_1        = 1
    ~~Else        LV_CAN_SND_MSG_PWR_MNG_1        = 0~~
    Endif

    LV_LDM_OFF                      = B_ldm_off

    If          LV_VAR_DCC = 1 or  (B_hschalt_komb = 1 and  B_rschalt_komb = 1)
    Then        LV_GS_UP    =       0
    Else        LV_GS_UP    =       B_hschalt_komb
    Endif

    If          LV_VAR_DCC          =       1       or
                (B_hschalt_komb     =       1       and
                 B_rschalt_komb     =       1)
    Then        LV_GS_DOWN          =       0
    Else        LV_GS_DOWN          =       B_rschalt_komb
    Endif

    LV_LDM_DRIV_ACT                 =       B_ldm_akt04
    GR_DT                           =       I_ges
```

STATE_TQ_WHEEL_TCS_AV = St_dsc_mradist

STATE_TQ_WHEEL_DRIV_ASI_AV = St_fas_mradist

LV_INH_STST_CDN = B_msaccid2

GS_IDC_GEAR_RAW = Spa_gang

GS_IDC_DISP_RAW = Spa_art

TQ_ECU_ETCU = Md_can_dmee (conversion!)

N_DISP_DYN = N_dzm
LV_STST_STOP = B_msastops

50.7.4 Common variables adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Fahrzeug	O/V	0...FFH	0...255	1	[-]
Vehicle variant					
pminfo1	O/V/S	0...FFFFH	0...65535	1	[-]
information storage 1 from power management					
pminfo2	O/V/S	0...FFH	0...255	1	[-]
information storage 2 from power management					

Input data:

VAR_VEH	Pminfo1[37]	Pminfo2[29]	

FUNCTION DESCRIPTION:

Adaption to BMW environment.

General information:

The variable Fahrzeug is updated only during ECU initialization.

Application conditions:

Initialisation at reset:

```
IF   VAR_VEH = 1 THEN   Fahrzeug = 65
IF   VAR_VEH = 2 THEN   Fahrzeug = 90
IF   VAR_VEH = 3
  THEN   Fahrzeug = 46
  ELSE   Fahrzeug = 60
ENDIF
```

Recurrence: pminfo1, pminfo2 100ms

Activation: at every engine operating state

Deactivation: -

Formula section:

pminfo1 = Pminfo1[37]
pminfo2 = Pminfo2[29]

50.8 AGGR MISC adaptation

50.8.1 Outputs for BMW functions which are defined as MISC exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_abgasklappe	V/O	0...1H	0...1	1	[-]
Bedingung Schaltung Abgasklappe (True = Klappe geöffnet)					
B_sport	V/O	0...1H	0...1	1	[-]
Sport switch activated					
St_fzg_1	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
carrier double-word 1 of all learned variants for customer environment					
St_fzg_2	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
carrier double-word 1 of all learned variants for customer environment					
Voltank	V/O	0...9F6H	0...255	0.1	[l]
Fuel tank level					
W_afs	V/O	8000H..7FFFH	-3276.8…3276.6	0.1	[°]
Steering wheel sensor angle					
W_dafs	V/O	8000H..7FFFH	-3276.8…3276.6	0.1	[°/s]
Steering wheel sensor angle velocity					
B_ftauf1	V/O	0...1H	0...1	1	[-]
Boolean for driver's door open					
B_mhauf1	V/O	0...1H	0...1	1	[-]
Boolean for engine hood open					
B_msafzg	V/O	0...1H	0...1	1	[-]
Variante MSA-Fahrzeug					
B_msataster	V/O	0...1H	0...1	1	[-]
Boolean for MSA button pressed					
Tvngang	V/O	0...FFFFH	0...655.35	0.01	[%]
PWM of neutral gear sensor					
Pbremsu	V/O	0...FFFFH	0...65535	1	[hPa]
Vacuum of brake servo unit					
PWM_NEUT_PSN_GB	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
Pulse width of the PLCD-Sensor					
B_kupp1	O/V	0...1H	0...1	1	[-]
St_bgkuppl					

Input data:

LV_RNG_L_REQ	STATE_VAR_DET_CUS_1	STATE_VAR_DET_CUS_2	PBSU
FTL	ANG_PSTE	VEL_ANG_PSTE	LV_KEY_VLD
LV_DRIV_DOOR_OPEN	NC_CYL_NR	LV_HOOD_OPEN	LV_STST_SWI_ACT
ECU_STATE	LV_VAR_STST	LV_EF	LV_CS_2

Import actions:

#if (NC_CYL_NR = 4)
ACTION_INFR_GetPwmNgs(OUT <pwm_neut_psn_gb>)
This action reads the PWM value of the PLCD-Sensor (NGS)
#endif

Application conditions:

Initialisation at reset: 0
except first calculation B_msafzg = LV_VAR_STST

Recurrence: 10ms
Activation: every engine state

Deactivation: -

Formula section:

```
B_kupp1         = LV_CS_2
B_msafzg        = LV_VAR_STST
B_sport         = LV_RNG_L_REQ
St_fzg_1        = STATE_VAR_DET_CUS_1
St_fzg_2        = STATE_VAR_DET_CUS_2
if   ECU_STATE = PWL
then
     Tvngang    = 0
Elseif
     B_abgasklappe      = not(LV_EF)
     Voltank            = FTL
     W_afs              = ANG_PSTE
     W_dafs             = VEL_ANG_PSTE
     B_ftauf1           = LV_DRIV_DOOR_OPEN
     B_mhauf1           = LV_HOOD_OPEN
     B_msataster        = LV_STST_SWI_ACT
     Pbremsu            = PBSU
     If              NC_CYL_NR  = 4
     then
             ACTION_INFR_GetPwmNgs(PWM_NEUT_PSN_GB)
             Tvngang       = PWM_NEUT_PSN_GB
     else    PWM_NEUT_PSN_GB = 0
             Tvngang       = 0
     endif

endif
```

50.9 Customer adaptation module: AGGR COMS

50.9.1 Outputs for BMW functions which are exported data from IGR (Intelligente Generatorreglung)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_cdigronw	O/V	0...1H	0...1	1	[-]
IGR-Codierdaten schreiben bzw. einlesen in die DDE/DME					
B_cdxenonw	O/V	0...1H	0...1	1	[-]
Xenonverbau-Codierdaten schreiben					
B_diagigr	O/V	0...1H	0...1	1	[-]
Flag "Diagnosejob gesetzt"					
B_fe	O/V	0...1H	0...1	1	[-]
Flag " Fertigungsmodus aktive"					
B_fetrawedeak	O/V	0...1H	0...1	1	[-]
Flag " Fertigungs-, Transport- und Werkstattmodus inaktive"					
B_schalt_dkg	O/V	0...1H	0...1	1	[-]
Drehzahlregelung aktiv					
B_tra	O/V	0...1H	0...1	1	[-]
Flag " Transportmodus aktive"					
B_we	O/V	0...1H	0...1	1	[-]
Flag " Werkstattmodus aktive"					
Fetrawe	O/V	0...FFH	0...255	1	[-]
Implementierung Fertigungs-, Transport- und Werkstattmodus mit Tester auslesen					
Mdk_kupp_verl	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Kupplungslastmoment bei aktiver Drehzahlregelung					
N_schalt_dkg	O/V	8000...7FFFH	-32768...32767	1	[rpm]
Vom DKG geforderte Motordrehzahl					
St_kupp_dkg	O/V	0...FFH	0...255	1	[-]
Zustand der Anfahrkupplung					
St_modus_dkg	O/V	0...FFH	0...255	1	[-]
Status torque request from TCT (DKG)					
St_rennstart_dkg	O/V	0...FFH	0...255	1	[-]
Statuswort Rennstart DKG					
Verbstatus[10]	O/V	0...FFH	0...255	1	[-]
Verbraucherstatus					

Input data:

LV_ALTER_CTL_ENA	LV_ALTER_CTL_EXT_ADJ	LV_LTG_GAS_ENA	LV_N_SP_TCT
N_SP_TCT	STATE_EGY_MIN_KWP	STATE_ETCU_CLU	STATE_MAX_AC_ST
STATE_TCT_INTV	STATE_VEH_CNS[10]	TQ_TCT_CAN	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisations: first calculation at reset,

Except: Verbstatus[10] : 0

0 at reset:

Mdk_kupp_verl, St_dkg_schalt, N_schalt_dkg, B_schalt_dkg, St_kupp_dkg, St_rennstart_dkg, St_modus_dkg

0 at power latch phase:

Mdk_kupp_verl, St_dkg, N_schalt_dkg, B_schalt_dkg, St_kupp_dkg, St_rennstart_dkg, St_modus_dkg

Recurrence : 100 ms: B_cdigronw, B_cdxenonw, B_diagigr, B_fe, B_fetrawedeak,
B_tra, B_we, Fetrawe, Verbstatus[10]

20ms: St_rennstart_dkg

10ms: Mdk_kupp_verl, St_dkg, N_schalt_dkg, B_schalt_dkg,
St_kupp_dkg, St_modus_dkg

Activation: *At every ECU state (including WakeUp)*

Deactivation: *After power latch phase*

Formula section:

Remark: all formulas are valid in a **physical** meaning

Verbstatus[10] = STATE_VEH_CNS[10]
B_cdigronw = LV_ALTER_CTL_ENA
B_cdxenonw = LV_LTG_GAS_ENA
B_diagigr = LV_ALTER_CTL_EXT_ADJ
Fetrawe = STATE_EGY_MIN_KWP

```
If      STATE_EGY_MIN_KWP = "PASSIV"
Then    B_fetrawedeak = 1
Else    B_fetrawedeak = 0
Endif

If      STATE_EGY_MIN_KWP = "EGY_1"
Then    B_fe = 1
Else    B_fe = 0
Endif

If      STATE_EGY_MIN_KWP = "EGY_2"
Then    B_tra = 1
Else    B_tra = 0
Endif

If      STATE_EGY_MIN_KWP = "EGY_3"
Then    B_we = 1
Else    B_we = 0
Endif
```

Mdk_kupp_verl = TQ_TCT_CAN

N_schalt_dkg = N_SP_TCT

B_schalt_dkg = LV_N_SP_TCT

St_kupp_dkg = STATE_ETCU_CLU

St_rennstart_dkg = STATE_MAX_AC_ST

St_modus_dkg = STATE_TCT_INTV

51 Misfire monitoring

Table of Contents

Index

51.1 MISF General

51.1.1 General description

The goal of the MISF is execute a misfire monitoring according legal requirements (conditions, engine area, errors & symptoms handling...) and provide to the ERRM aggregate following CARB misfire legal informations to manage the MIL and corresponding error Pcodes

- **CARB A misfire failure criterion:**

  Risk of catalyst damage, monitoring interval over 200 crankshaft during the driving cycle.

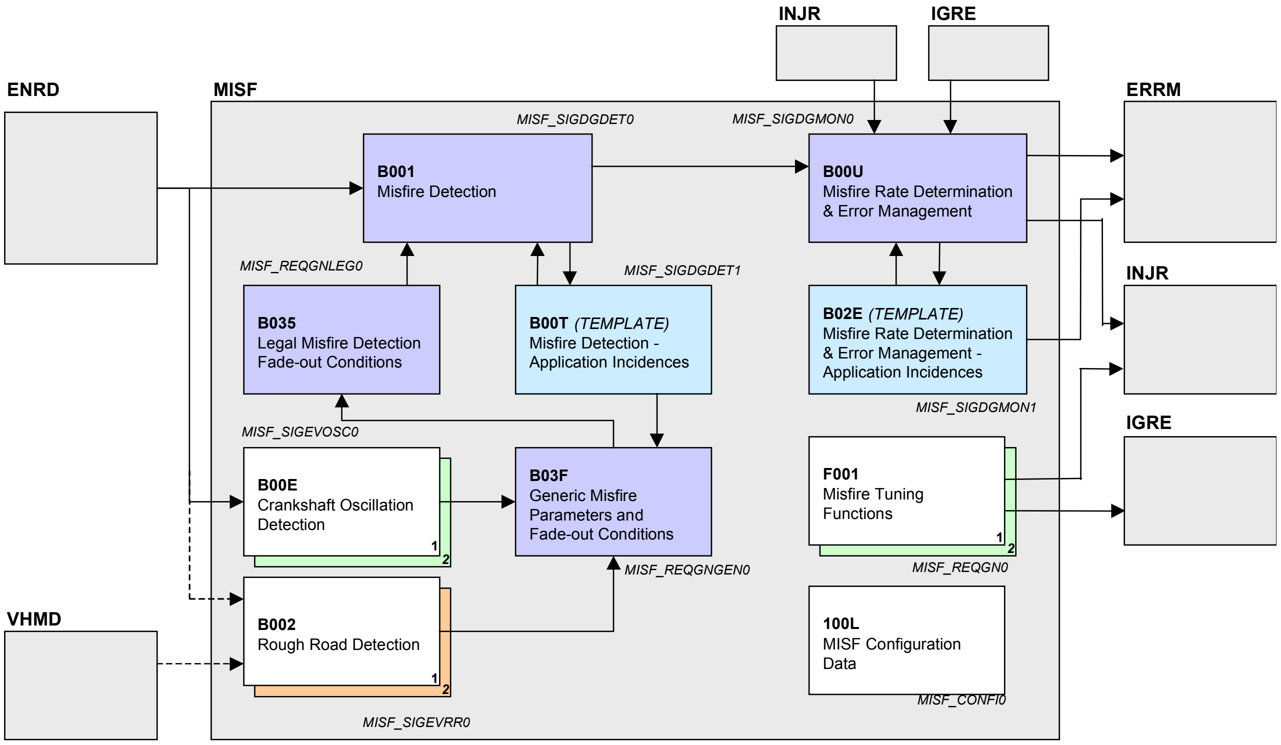
- **CARB B1 misfire failure criterion:**

  Emission increase, monitoring interval over the first 1000 crankshaft revolutions of the driving cycle.

- **CARB B4 misfire failure criterion:**

  Emission increase, monitoring interval over 1000 crankshaft revolutions. For error detection, misfire must be take place for 4 monitoring intervals (consecutive or not).

51.1.2 Architecture Overview



51.1.3 Description of the containing functions

51.1.3.1.1 B00E - Crankshaft oscillation detection

The crankshaft oscillation detection module uses engine roughness components to identify drivetrain oscillations than could cause wrong misfire detections in a defined engine operating area. If such oscillation occurs, this module triggers a fade-out.

51.1.3.1.2 B002 - Rough road detection

The rough road detection module uses informations coming from wheel speed sensor(s) or from ABS module (via harness or CAN) to identify transmission oscillations than could cause wrong misfire detections in a defined engine operating area. If such oscillation occurs, this module triggers a fade-out.

51.1.3.1.3 B03F - Generic misfire parameters & fade-out conditions

This module provides parameters needed for the misfire detection process (data delayed, data stacks, and zero load line...). It also managed all generic misfire fade-out conditions linked to the misfire detection method.

51.1.3.1.4 B035 - Legal misfire detection fade-out conditions

This module manages all legal misfire fade-out conditions defined in legal texts (US & EC).

51.1.3.1.5 B00T - Misfire detection - Application incidences

This module defines specific corrections for detection thresholds, application specific fade-out with standardised outputs to the generic modules.

51.1.3.1.6 B001 - Misfire detection

Detection core function based on engine roughness index provided by ENRD.

51.1.3.1.7 B00T - Misfire detection - Application incidences

Some functionality for calibration ease can be launched after detection.

51.1.3.1.8 B02E - Misfire rate determination & error management - Application Incidences

This module defines application specific fade-out & informations with standardised outputs to the generic misfire rate determination module. Diagnosis conditions are defined in this module.

Must be executed before B00U module.

51.1.3.1.9 B00U - Misfire rate determination & error management

This module defines the Misfire criterions according legal texts description *(MIS_A = misfire damage catalyst criterion, MIS_B1 = misfire emission criterion at engine warm-up & MIS_B4 = misfire emission criterion)* and identify the cylinders in failures

51.1.3.1.10 B02E - Misfire rate determination & error management - Application Incidences

This module defines misfire errors according the type chosen by NC_TREAT_DIAG_MIS: errors defined per misfire criterions or errors defined per cylinder.

Failures like Misfire with low fuel tank level, Multiple cylinder misfire, Random cylinder misfire are also managed in this module.

Must be executed after B00U module.

51.1.3.1.11 F001 - Misfire Tuning Functions

This module allows to generate misfire patterns (continuous or pseudo random) via injection and/or ignition shut-off interfaces.

This module is optionnal and can be integrated during validation and calibration stages. It is strongly recommended to remove this functionally on serial product software (integration choice via NC_USE_MIS_GEN compilation switch).

51.2 MISF Configuration Data

Input data:

NLC_TREAT_DIAG_MIS	NLC_USE_ER_STND_MIS	NLC_USE_MIS_GEN	NC_CONF_RR_MIS
NLC_USE_CRK_OSC_MIS	NC_SIZE_THD_ER_BUF	NC_CBK_EX_NR_MISF[NC_CYL_NR]	NC_CYL_NR
SEG_NR_ER			

General information:

The following describes the general rules for determination of the configuration data

51.2.1 Global configuration data

Here are listed the configuration data, which can be used in other aggregates :

Data	Value
NLC_TREAT_DIAG_MIS	**1**

51.2.2 Local configuration data

Here are listed the configuration data, which are used only in the MISF aggregate.

Data	Value
NC_MISF_VERS	**1**
NLC_USE_ER_STND_MIS	**0**
NC_SIZE_THD_ER_BUF	**4**
NC_CONF_RR_MIS	**OTHER**
NLC_USE_CRK_OSC_MIS	**1**
NLC_USE_MIS_GEN	**1**
NC_CBK_EX_NR_MISF[NC_CYL_NR]	*see table below*

Configuration for NC_CBK_EX_NR_MISF:

Cylinders 0, 2 & 4 located on exhaust cylinder bank 0
Cylinders 1, 3 & 5 located on exhaust cylinder bank 1

SEG_NR_ER	0	1	2	3	4	5
NC_CBK_EX_NR_MISF	0	1	0	1	0	1

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_MISF_VERS	1	0...FFH	0...255	1	[-]
MISF aggregate version					

51.3 Misfire detection

51.3.1 Global fade out switch for misfire detection and monitoring

Description:

There is one global switch which can be used for inhibiting all the MISF aggregate modules:

- Generic misfire parameters & fade-out conditions	: Aggr MISF, chapter OBDII
- Legal misfire detection fade-out conditions	: Aggr MISF, chapter OBDII
- Rough road detection	: Aggr MISF, chapter OBDII
- Crankshaft oscillation detection	: Aggr MISF, chapter OBDII
- Misfire detection	: Aggr MISF, chapter OBDII
- Appl. Inc. for misfire detection	: Aggr MISF, chapter OBDII
- Misfire rate and criterions determination	: Aggr MISF, chapter OBDII
- Appl. Inc. for Misfire rate and criterions determination	: Aggr MISF, chapter OBDII

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
LC_MIS_INH	1	0...1H	0...1	1	[-]
Global switch to disable all misfire monitoring related modules (Inhibition with = 1)					

51.3.2 Misfire detection thresholds determination for engine roughness index

Output data:

Name	Mode	Hex. Limit	Phys. Limit	Resol.	Unit
FAC_THD_ER	V	0...400H	0...4	3.91E-03	[–]
Global gain multiplicative correction applied to basic misfire detection thresholds					
THD_ER	V/O	8000...7FFFH	-32768...32767	1	[-]
Misfire detection threshold for current ER value					
THD_ER_BUF[NC_SIZE_THD_ER_BUF]	V	8000...7FFFH	-32768...32767	1	[-]
Misfire detection threshold stack for ER evaluation					
THD_ER_CYL[NC_CYL_NR]	V	8000...7FFFH	-32768...32767	1	[-]
Cylinder Misfire detection threshold for current ER value					

Input data:

N_32	TCO	LV_AT	FAC_THD_APP_ER
N	LV_IS	LOAD_MIS	INH_INJ
SEG_NR_ER	LV_ENA_SEG_T_MES	NC_CMB_CONF	LV_S_ACT
LV_HOM_AFL_ACT	LC_MIS_INH	LV_DRI	LV_CVT
LV_VAR_4WD	THD_ER_CLC		

Function description:

ER strategy misfire detection is based on a comparison between a cylinder specific ER value (nominal or normalised) and a threshold who is relative to the engine parameters when this same cylinder was in intake phase (threshold relative to combustion conditions image).

The engine roughness value ER depends strongly on the current operating point, both in its spread during undisturbed engine operation, and in the signal amplitude in case of an actual misfire.

ER misfire detection thresholds are defined according basic threshold maps and threshold corrections:

- *A basic threshold map:*

This map is function of engine load and speed which are calibrated for engine at operating temperature as well as for adapted transmitter gear. One map is assigned through the vehicle transmission type, idle speed condition & the combustion mode.

- *A threshold temperature correction:*

A temperature correction is applied to compensate larger signal amplitudes arise of ER values at lower engine temperatures. This correction is relative to TCO value.

- *A threshold correction during active injection valve cut-off:*

If one or several injection valves are cut off selectively by the engine management system (IV diagnoses ...), then the cylinder or cylinders are excluded from the misfire check. Due to the system, the signal amplitude of the ER value is reduced with a misfire at another cylinder. To improve the detection for this period, the absolute amount of the threshold can be reduced multiplicatively with the factor IP_FAC_THD_IV_ER(N_32) or IP_FAC_THD_IV_S_ER(N_32) *(in stratifed mode)*. *FAC_THD_IV_ER is a temporary data.*

- *A threshold correction defined in application incidences:*

FAC_THD_APP_ER is a threshold multiplicative correction who is defined in the misfire detection application incidences file.

51.3.2.1 Multiplicative correction for misfire thresholds

Application conditions:

Initialisation: *FAC_THD_IV_ER = 1*
Recurrence: every ENRD segment task
Activation: *LV_ENA_SEG_T_MES = 1*
And *LC_MIS_INH = 0*
Deactivation: *LV_ENA_SEG_T_MES = 0*
Or *LC_MIS_INH = 1*

Formula section:

```
#IF NC_CMB_CONF = AFS_S Or AFS_AFL_S
    If      INH_INJ = 0
    Then    FAC_THD_IV_ER = 1
    Elself  LV_S_ACT = 1
    Then    FAC_THD_IV_ER = IP_FAC_THD_IV_S_ER(N_32)
    Else    FAC_THD_IV_ER = IP_FAC_THD_IV_ER(N_32)
    EndIf
```

```
#ELSE
    If      INH_INJ = 0
    Then    FAC_THD_IV_ER = 1
    Else    FAC_THD_IV_ER = IP_FAC_THD_IV_ER(N_32)
    EndIf
#ENDIF
FAC_THD_ER = IP_FAC_THD_TCO_ER(TCO) * FAC_THD_IV_ER * FAC_THD_APP_ER
```

51.3.2.2 Misfire detection threshold buffers management

To manage the delay between cylinder intake phase and cylinder ER values, buffers are used to store threshold values before using for detection.

THD_ER_BUF[NC_SIZE_THD_ER_BUF] buffer is managed as a FIFO stack.

THD_ER_BUF[0] threshold value is the one obtained at current segment.

THD_ER is threshold value delayed and who will be used for detection with current segment ER value. This delay between ER value and cylinder corresponding threshold depends on engine cylinder number (NC_CYL_NR).

51.3.2.3 Misfire detection threshold based on engine roughness value (THD_ER)

By using the ER value for misfire detection, the engine speed has the greatest influence on basic thresholds definition. On the other hand, by using the ER_STND values for misfire detection, the engine load has the greatest influence on basic thresholds definition.

The influence of the engine load on the detection thresholds can change rapidly during instationary operation, it is urgently necessary to determine and to temporarily store the threshold values already at the time of the actual combustion before the calculatory check.

note: at each ER segment occurence, THD_ER_BUF is shifted with one memory unit (FIFO management) before determination process.

Application conditions:

Initialisation: at ECU reset **Or** engine stop

```
    THD_ER = -32768
    //  ER Threshold buffer init on ER buffer reinitialisation
    For k = 0 : NC_SIZE_THD_ER_BUF-1
            THD_ER_BUF[k] = -32768    // For all buffer cells
    EndFor
    For x = 0 : NC_CYL_NR-1
            THD_ER_CYL[x] = -32768    // For all cylinders
    EndFor
```

Recurrence: every ENRD segment task

Activation: *LV_ENA_SEG_T_MES = 1*

***And** LC_MIS_INH = 0*

```
Deactivation:  LV_ENA_SEG_T_MES = 0
               Or LC_MIS_INH = 1
Actions on Deactivation event:
    THD_ER = -32768
    // ER Threshold buffer init on ER buffer reinitialisation
    For k = 0 : NC_SIZE_THD_ER_BUF-1
        THD_ER_BUF[k] = -32768    // For all buffer cells
    EndFor
    For x = 0 : NC_CYL_NR-1
        THD_ER_CYL[x] = -32768    // For all cylinders
    EndFor
```

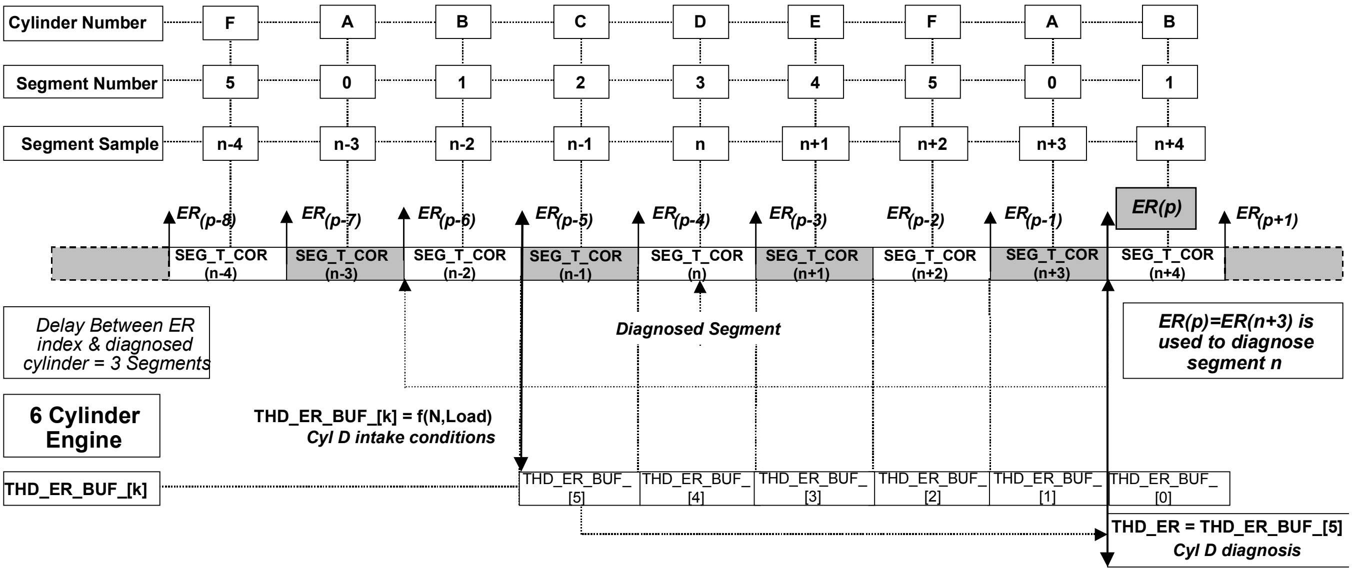
Formula section:

THD_ER_BUF[0] = THD_ER_CLC * FAC_THD_ER

THD_ER = THD_ER_BUF[NC_SIZE_THD_ER_BUF-1]   // Last cell of THD_ER buffer

THD_ER_CYL[SEG_NR_ER] = THD_ER

6 cylinder engine example



Calibration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
IP_FAC_THD_TCO_ER	8	0...1FFH	0...1.99609	3.9063e-3	[-]
LDP_TCO_IP_FAC_THD_TCO_ER	8	0...FEH	-48...142.5	0.75	[°C]
Temperature-dependent threshold reduction					
IP_FAC_THD_IV_ER	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_N_32_IP_FAC_THD_IV_ER	6	0...FFH	0...8160	32	[rpm]
Factor for adaptation of the threshold in case of active cylinder cut-off					
#IF NC_CMB_CONF = AFS_S Or AFS_AFL_S					
IP_FAC_THD_IV_S_ER	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_N_32_IP_FAC_THD_IV_S_ER	6	0...FFH	0...8160	32	[rpm]
Factor for adaptation of the threshold in case of active cylinder cut-off in stratified combustion mode					
#ENDIF					

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_SIZE_THD_ER_BUF	1	0...8H	0...8	1	[-]
Misfire detection threshold buffer size					

51.3.3 Misfire detection based on engine roughness evaluation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DET_MIS	O/V	0...1H	0...1	1	[-]
Single misfire detected and confirmed (after fade out test)					
LV_INH_DET_MIS	O/V	0...1H	0...1	1	[-]
General misfire detection fade out					
LV_DET_THD_MIS	V	0...1H	0...1	1	[-]
Single misfire detected based only on threshold criterion (before fade out test)					
MIS_STATE_ER	V	0...FFH	0...255	1	[-]
Status carrier byte of actual detected misfire through engine roughness index - 1 bit / cylinder					
MIS_DET_CYL_INH	V	0...FFH	0...255	1	[-]
Cylinder misfiring detection inhibition carrier byte when a fade out condition occurred (after masking & delay)					

Input data:

LV_REQ_INH_MIS	LV_ENA_ER	ER	ER_STND
THD_ER	LC_MIS_INH	LV_ZDLY_DIAG_MIS	SEG_NR_ER
LV_DIAG_MIS	NC_CYL_NR	LV_ER_CLC_MIS_DET_CUS_SPC	MIS_STATE_CUS_SPC

General information:

The system identifies misfiring by monitoring engine roughness index. Misfiring causes the angular velocity of the crankshaft to drop in an angular range specific to the cylinder in question.

Misfire detection is based on nominal engine roughness (ER) by using THD_ER detection threshold.

The identification of the cylinder(s) detected in misfire is realised according SEG_NR_ER segment reference (see definition and scheme in ENRD aggregate) *(Chapter system variables)*.

THD_ER is already phased according ER delay *(see detection thresholds management)*.

Application conditions:

*Initialisation: at ECU reset, at Engine Stop **Or** at Deactivation Event*

LV_DET_MIS = 0

LV_DET_THD_MIS = 0

LV_INH_DET_MIS = 1

MIS_STATE_ER = 0

MIS_DET_CYL_INH = $2^{NC\_CYL\_NR} - 1$

Recurrence: every segment task

Activation: *LV_ENA_ER = 1*

***And** LC_MIS_INH = 0*

Deactivation: *LV_ENA_ER = 0*

***Or** LC_MIS_INH = 1*

Update rate:

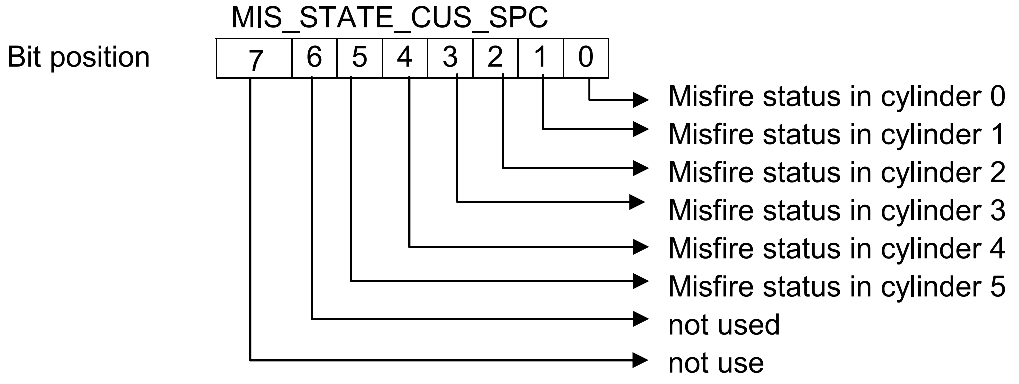
ENRD segment task

Forumla section:

Misfire detection criterion

```
#IF NLC_USE_ER_STND_MIS = 0         // ER based detection
    If      ( ER < THD_ER )
            OR
            {
             LC_MIS_INH_CUS_SPC = 0                 AND
             LV_ER_CLC_MIS_DET_CUS_SPC =1    AND
            (Bit (SEG_NR_ER)  of  MIS_STATE_CUS_SPC = 1 )
            }
                                         // Read bit position of MIS_STATE_CUS_SPC
    Then    LV_DET_THD_MIS = 1          // basic detection before fade-out
    Else    LV_DET_THD_MIS = 0
    EndIf
#ELSE                                   // ER_STND based detection
    If      ( ER_STND < THD_ER )
            OR
            {
             LC_MIS_INH_CUS_SPC = 0                 AND
             LV_ER_CLC_MIS_DET_CUS_SPC  =1  AND
            (Bit (SEG_NR_ER)  of  MIS_STATE_CUS_SPC = 1 )
            }
                                        // Read bit position of MIS_STATE_CUS_SPC
    Then    LV_DET_THD_MIS = 1          // basic detection before fade-out
    Else    LV_DET_THD_MIS = 0
    EndIf
#ENDIF
```

// Customer specific misfire detection algorithm, has been introduced in this release, If there is catalyst heating and all related conditions are met, then LV_ER_CLC_MIS_DET_CUS_SPC is set to 1. This enables customer specific algorithm. MIS_STATE_CUS_SPC is a status byte, and the information from the misfiring cylinder are stored in it.



Value in Bit position = 0, No misfire detected.
Value in Bit position = 1, Misfire detected on respective cylinder.

General fade-out management

```
If    LV_REQ_INH_MIS = 0         // no fade-out request on going (legal and generic)
      And   ( LV_ZDLY_DIAG_MIS = 1 Or LV_DIAG_MIS = 1 )  // diagnosis phase started
Then        LV_INH_DET_MIS = 0
            MIS_DET_CYL_INH[SEG_NR_ER] = 0
            (corresponding cylinder bit in MIS_DET_CYL_INH structure is set to 0,
            misfire detection will be enabled on this cylinder
Else        LV_INH_DET_MIS = 1
            MIS_DET_CYL_INH[SEG_NR_ER] = 1
            (corresponding cylinder bit in MIS_DET_CYL_INH structure is set to 1,
            misfire detection will be disabled on this cylinder
EndIf
```

Misfire detection & cylinder identification

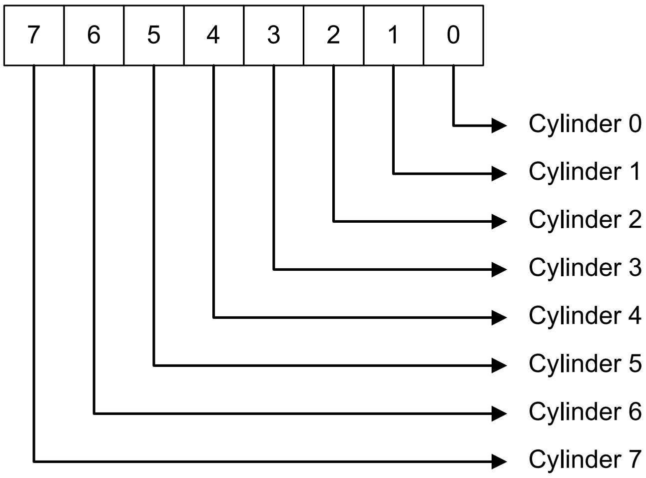
```
If      LV_DET_THD_MIS = 1
        And     LV_INH_DET_MIS = 0          // no fade out on monitored cylinder
Then            LV_DET_MIS = 1              // an instantaneous misfiring is detected
                MIS_STATE_ER[SEG_NR_ER] = 1
                Cylinder specific bit in MIS_STATE_ER carrier structure is set to 1 according SEG_NR_ER
                segment reference.
Else            LV_DET_MIS = 0
                MIS_STATE_ER[SEG_NR_ER] = 0
                Cylinder specific bit in MIS_STATE_ER carrier structure is set to 0 according SEG_NR_ER
                segment reference.
EndIf
```

Misfire cylinder fade out carrier definition:

MIS_DET_CYL_INH : Carrier used for cylinder fade-out reference
Shows the cylinders that will be in fade-out during the detection phase

MIS_STATE_ER *:* Carrier used for identification of the cylinder detected in misfire
Shows the cylinders that will be detected in misfire

Carrier stucture valid for
MIS_STATE_ER
and
MIS_DET_CYL_INH



Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
LC_MIS_INH_CUS_SPC	1	0...1H	0...1	1	[-]
Global switch to disable customer specific misfire monitoring (Inhibition with = 1)					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_USE_ER_STND_MIS	1	0...1H	0...1	1	[-]
Misfire detection integration mode (based on ER index = 0, based on ER_STND index = 1)					

51.4 Misfire detection - Application incidences

51.4.1 Global fade out condition (N_32 threshold)

Input data:

N_32			

FUNCTION DESCRIPTION:

General information:

There is one global condition N_32 > C_N_THD_MIS_INH which can be used for inhibition (deactivation) of the following modules:

- Appl. Inc. for engine roughness calc.	: chapter System variables
- Engine roughness calculation	: chapter System variables
- Appl. Inc. for misfire detection	: chapter OBDII
- Misfire detection	: chapter OBDII
- Misfire rate detection and error handling	: chapter OBDII

Formula section:

```
If      N_32 > C_N_THD_MIS_INH
Then    Engine roughness/Misfire detection is inhibited
Else    no inhibition due to this condition
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_THD_MIS_INH	1	0...FFH	0...8160	32	[rpm]
Global condition for inhibiting (deactivation) of all misfire modules					

51.4.2 Misfire detection threshold correction gain specific to application

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_THD_APP_ER	V/O	0...1FFH	0...1.99609	3.9063e-3	[-]
Misfire detection threshold correction gain specific to application					

Input data:

TQ_ADD_CH	LV_IS	LV_S_ACT	LC_MIS_INH
LV_ENA_SEG_T_MES			

Description:

To ensure misfire detection during active catalyst heating in IS (in homogenous mode only), the threshold is multiplied with a factor depending on the torque reserve at catalyst heating TQ_ADD_CH.

Application conditions:

Initialization: all 1 at reset and at each activation and deactivation

Recurrency: segment

Activation: LV_ENA_SEG_T_MES = 1 **and** LC_MIS_INH = 0

Deactivation: LV_ENA_SEG_T_MES = 0 **or** LC_MIS_INH = 1

Formula section:

```
If      LV_IS = 1           and
        LV_S_ACT = 0
Then    FAC_THD_APP_ER = IP_FAC_THD_ER_CH(TQ_ADD_CH)
Else    FAC_THD_APP_ER = 1
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_THD_ER_CH	4	0...1FFH	0...1.99609	3.9063e-3	[-]
LDP_TQ_ADD_CH_IP_FAC_THD_ER_CH	4	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Threshold correction depending on TQ_ADD_CH for catalyst heating					

51.4.3 Misfire detection inhibition related to OBDI diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_OBD_DET_MIS	V/O	0...1H	0...1	1	[-]
OBD I misfire detection process inhibition flag					

Input data:

LV_ERR_SEG_AD_ER	LV_ERR_MAF	LC_MIS_INH	LV_ERR_IVVT
LV_ENA_ER	LV_ERR_TPS		

Description:

Misfire detection process is inhibited when one of the following OBD I errors occurs.

Depending on calibration, IVVT diagnosis is considered for inhibition or not.

Application conditions:

Initialization: all 0 at reset and at each activation and deactivation

Recurrency: updated every segment before misfire detection function

Activation: LV_ENA_ER = 1 **and** LC_MIS_INH = 0

Deactivation: LV_ENA_ER = 0 **or** LC_MIS_ INH = 1

Formula section:

```
If      LV_ERR_MAF = 1                 // mass air flow signal error detected
   or   LV_ERR_TPS = 1                 // throttle error
   or   [ LV_ERR_IVVT = 1 and LC_USE_IVVT_INH_MIS = 1]   // VANOS error
   or   LV_ERR_SEG_AD_ER = 1   // Segment adaptation error

Then    LV_INH_OBD_DET_MIS  = 1
Else    LV_INH_OBD_DET_MIS  = 0
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_USE_IVVT_INH_MIS	1	0...1H	0...1	1	[-]
Enable IVVT diagnosis for Inhibition of misfire detection					

51.4.4 Misfire detection inhibition related to AMT gearshift intervention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MIS_INH_IV_KNK	V	0...1H	0...1	1	[-]
Misfire detection process inhibition due to injection shut of from knock control					
LV_MIS_INH_CS	V	0...1H	0...1	1	[-]
Misfire detection process inhibition during clutch transition.					
LV_INH_APP_DET_MIS	O/V	0...1H	0...1	1	[-]
Misfire detection process inhibition flag (application specific).					

Input data:

LC_MIS_INH	LV_ENA_ER	LV_GS	INH_IV_KNK
LV_CS_CUS			

FUNCTION DESCRIPTION:

General information:

LV_INH_APP_DET_MIS is used for project-specific inhibition of Misfire detection, e.g.:

- AMT gearshift intervention

Description:

When AMT gearshift intervention is detected (LV_GS = 1) misfire detection can be suppresed because this load jump can cause a segment period jump and crankshaft vibration, depending on the engine operating state and load request.

If AMT gearshift intervention has been detected (LV_GS changes from 1 to 0 or 0 to 1), Misfire detection can be suppressed for the applicable constant period C_T_AMT_GS_DLY_MIS starting at transition LV_GS 1->0.

Application conditions:

Initialization:	all 0 at reset and at each activation and deactivation		
Recurrence:	updated every segment before misfire detection function		
Activation:	LV_ENA_ER = 1	**and**	LC_MIS_INH = 0
Deactivation:	LV_ENA_ER = 0	**or**	LC_MIS_ INH = 1

Formula section:

Cylinder fuel shut-off:

// if any one cylinder is switched off from Knock control fuinction, then misfire is deactivated. This can be activated using the switch LC_MIS_INH_IV_KNK.

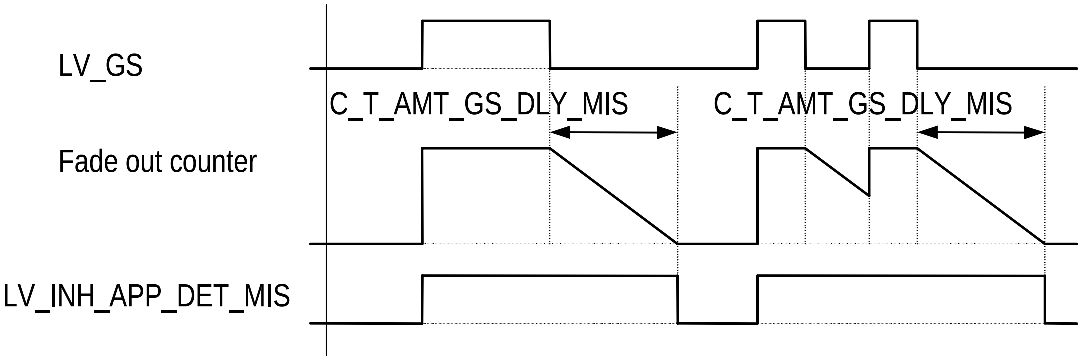
NOTE: The Activation by LC_MIS_INH_IV_KNK has to be checked with CARB

```
If(1)  INH_IV_KNK > 0       // indexed by cylinder bit position
Then(1)     LV_MIS_INH_IV_KNK = 1 after a  delay of C_NR_TDC_IV_OFF_DET_MIS_KNK

            TDC,s
Else(1)     LV_ MIS_INH_IV_KNK = 0 after a delay of C_NR_TDC_IV_OFF_DET_MIS_KNK
                                                                        TDC,s
EndIf(1)


If(2) (  ( LV_CS_CUS 1->0  or  LV_CS_CUS 0->1)  AND  LC_MIS_INH_CS=1  )
Then(2)  LV_MIS_INH_CS = 1 for C_T_CS_DLY_MIS    TDC,s
Else(2)  LV_MIS_INH_CS = 0
Endif(2)
```

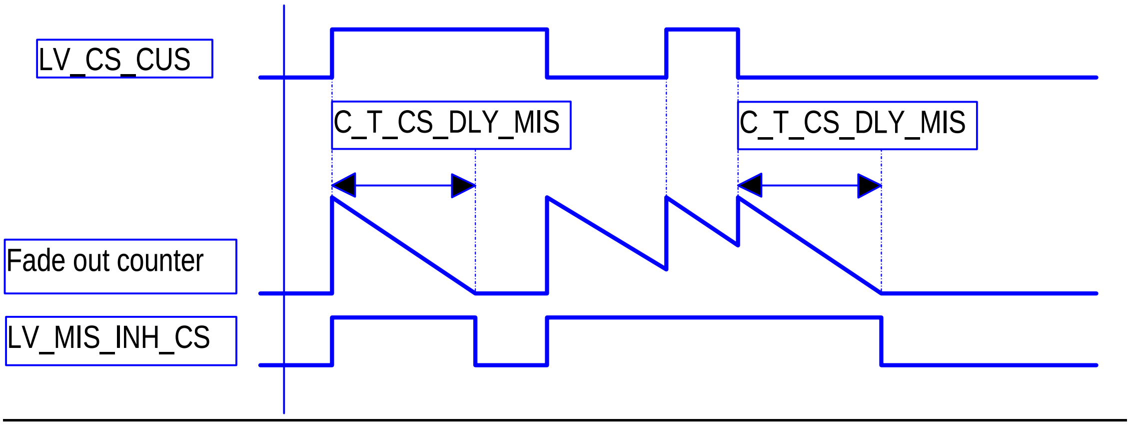
// inhibition of misfire function during gear change, is not allowed for CARB applications.

```
If(3)       LV_GS = 1
     Or     LV_GS 1->0 and Time C_T_AMT_GS_DLY_MIS is active
     Or     LV_MIS_INH_IV_KNK =1     AND    LC_MIS_INH_IV_KNK = 1
     Or     LV_MIS_INH_CS = 1
Then(3)     LV_INH_APP_DET_MIS = 1
Else(3)     LV_INH_APP_DET_MIS = 0
EndIf(3)
```

Fade out behaviour summary for AMT gear shift :



Fade out behaviour summary for MT – Change in clutch signal



Calibration data:

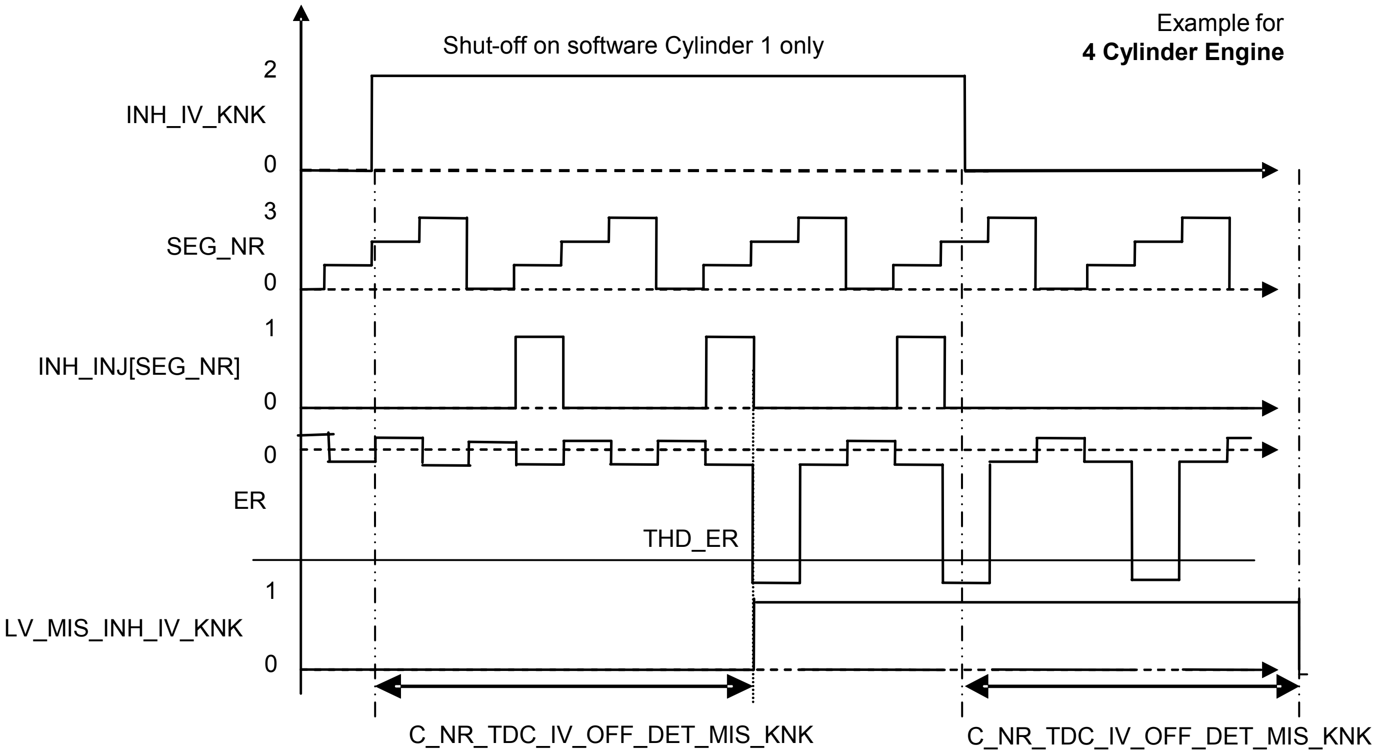
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NR_TDC_IV_OFF_DET_MIS_KNK	1	0...FFH	0...255	1	[-]
TDC delay to synchronise cylinder shut-off (from Knock control) information with the misfire detection index					
LC_MIS_INH_IV_KNK	1	0...1H	0...1	1	[-]
Switch to enable misfire deactivation due to injection switch off from knock control					
LC_MIS_INH_CS	1	0...1H	0...1	1	[-]
Switch to enable misfire deactivation during Clutch transition					
C_T_CS_DLY_MIS	1	0...FFH	0...255	1	[-]
No. of TDC,s for which misfire is deactivated during clutch transition.					
C_T_AMT_GS_DLY_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when gearshift of AMT has been detected					

// Initial appliation value for C_NR_TDC_IV_OFF_DET_MIS_KNK, should be taken from C_NR_TDC_IV_OFF_DET_MIS + *NC_CYL_NR-2*

There is a delay of 2 segments for calculation of INH_IV_KNK, hence delay of

NC_CYL_NR-2 segments to shut off the injector. Injection switch off only in the next segment calculation.

Further there is a delay of C_NR_TDC_IV_OFF_DET_MIS TDC,s for ER calculation. Hence total delay is C_NR_TDC_IV_OFF_DET_MIS + *NC_CYL_NR-2*



51.4.5 Rough road detection fade out condition

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
LV_INH_APP_RR	V/O	0...1H	0...1	1	[-]
Rough road detection appl. Inc. fade out					

Input data:

LC_MIS_INH	LV_ERR_VS		

Description:

Rough road detection is inhibited if no VS signal is available (LV_ERR_VS = 1).

Application conditions:

Initialisation: all 0 at reset and at each activation and deactivation

Recurrence: same as rough road detection update rate

Activation: LC_MIS_INH = 0

Deactivation: LC_MIS_ INH = 1

Formula section:

```
If      LV_ERR_VS = 1
Then    LV_INH_APP_RR = 1
Else    LV_INH_APP_RR = 0
Endif
```

51.4.6 Misfire low fuel tank level informations

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_FTL_L_DET_MIS	V/O	0...1H	0...1	1	[-]
Misfire detection inhibition flag in case of low fuel tank level					
LV_FTL_L_DIAG_MIS	V/O	0...1H	0...1	1	[-]
Request to allow error management of misfire occurence with low fuel tank level					

Input data:

LV_FTL_OBD_INH_L	LC_MIS_INH		

Description:

- Misfire detection can be supressed if low fuel is detected (LV_INH_FTL_L_DET_MIS).
- Low fuel signal can be used to calculate P-Code “Misfire with low fuel” P313.

Application conditions:

Initialisation: all 0 at reset and at each activation and deactivation

Recurrence: 200ms *//same as fuel tank level information*

Activation: LC_MIS_INH = 0

Deactivation: LC_MIS_INH = 1

Formula section:

Low fuel tank level information for misfire detection inhibition

```
If    LV_FTL_OBD_INH_L = 1
      And   LC_FTL_L_DET_MIS = 1
Then        LV_INH_FTL_L_DET_MIS = 1
Else        LV_INH_FTL_L_DET_MIS = 0
EndIf
```

Low fuel tank level information for error management of symptom "misfire with low fuel tank level".

```
If    LV_FTL_OBD_INH_L = 1
      And   LC_FTL_L_DIAG_MIS = 1
Then        LV_FTL_L_DIAG_MIS = 1
Else        LV_FTL_L_DIAG_MIS = 0
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_FTL_L_DET_MIS	1	0...1H	0...1	1	[-]
Inhibition of misfire detection in case of low fuel tank level					
LC_FTL_L_DIAG_MIS	1	0...1H	0...1	1	[-]
Enable error management of misfire with low fuel tank level					

51.4.7 Total counter misfire detection

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
CTR_TOT_MIS	V/S	0...FFFFFFFFH	0...4294967295	1	[-]
Overall misfire detection counter					

Input data:

LV_DET_MIS	LC_MIS_INH		

FUNCTION DESCRIPTION:

General information:

This counter is incremented at each misfire detection based on misfire index evaluation *(before crossed diagnosis with IGC & IV OBDI diags evaluation)*. Thus, it's possible to determine the total number of misfire detected during engine lifetime.

Application conditions:

Initialisation: restored from NVMY

Recurrence: updated every segment after LV_DET_MIS calculation

Activation: LC_MIS_INH = 0

Deactivation: LC_MIS_INH = 1

Formula section:

On NV memory formatting or on NV memory corruption detection :

CTR_TOT_MIS = 0

Processing :

at each transition LV_DET_MIS 0 -> 1

CTR_TOT_MIS = CTR_TOT_MIS + 1

51.4.8 Crankshaft oscillation detection fade-out

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
LV_INH_CRK_OSC_DET	V/O	0...1H	0...1	1	[-]
Crankshaft oscillation detection fade out					

Input data:

LV_ERR_VS	LC_MIS_INH		

FUNCTION DESCRIPTION:

Crankshaft oscillation detection process can be inhibited when one function and/or error requires an inhibition.

Note : This bit is necessary only if the crankshaft oscillation detection module is used in the project (NLC_USE_CRK_OSC_MIS = 1)

Application conditions:

Initialisation:	all 0 at reset and at each activation and deactivation
Recurrence:	Segment, before Crankshaft oscillation detection
Activation:	LC_MIS_INH = 0
Deactivation:	LC_MIS_ INH = 1

Formula section:

```
If      LV_ERR_VS = 1
Then    LV_INH_CRK_OSC_DET = 1
Else    LV_INH_CRK_OSC_DET = 0
EndIf
```

51.4.9 Mean value of the corrected basic ignition angles

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
IGA_BAS_COR_MV	V/O	0...FFH	-35.625...60	0.375	[°CRK]
Mean value of the corrected basic ignition angles					

Input data:

IGA_BAS_COR			

Application conditions:

Initialisation: IGA_BAS_COR_MV = 0 at reset

Recurrence: 10 ms

Activation: LV_IGK = 1

Formula section:

IGA_BAS_COR_MV = IGA_BAS_COR

51.4.10 Correction gain for engine load specific to application

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_GAIN_LOAD_MIS	V/O	0...1FFH	0...1.99609	3.9063e-3	[-]
Correction gain of the normalised engine load for misfire, specific to application					

Input data:

LV_ENA_SEG_T_MES	LC_MIS_INH		

Description:

Here can be defined a gain who can change the normalised engine load dedicated to misfire monitoring according specific load corrections.

This correction can be applied during torque reduction sequences specific to project.

Application conditions:

Activation: *LV_ENA_SEG_T_MES = 1*

And *LC_MIS_INH = 0*

Deactivation: *LV_ENA_SEG_T_MES= 0*

Or *LC_MIS_INH = 1*

Initialisation: FAC_GAIN_LOAD_MIS = 1

Recurrency: -

51.4.11 Current Misfire Detection Threshold

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
THD_ER_CLC	V/O	8000...7FFFH	-32768...32767	1	[-]
Value of the table used for Misfire detection threshold at current segment					
THD_ER_CLC_PREV	V	8000...7FFFH	-32768...32767	1	[-]
Previous value of the table used for misfire detection threshold at current segment					
LV_ER_CLC_PREV_INI	-	0...1H	0...1	1	[-]
Boolean for initialisation of THD_ER_CLC_PREV at start of gradient limitation					

Input data:

N	LV_AT	OPM_AV	LV_IS
LOAD_MIS	LC_MIS_INH	LV_ENA_SEG_T_MES	NC_CMB_CONF
LV_VAR_TCT			

General information:

THD_ER_CLC represents the current value used for determination of the misfire detection threshold. This data must be calculated before THD_ER calculation and after LOAD_MIS calculation.

Description:

Keyword configuration table:

NC_CMB_CONF	OPM_AV	xx
AFS	2	AFS
AFS_AFL	2	AFS
AFS_AFL	3	HOM_S
AFS_S	1	S
AFS_S	2	AFS
AFS_AFL_S	1	S
AFS_AFL_S	2	AFS
AFS_AFL_S	3	HOM_S

Application conditions:

Initialisation: at ECU reset **Or** engine stop
THD_ER_CLC = -32768
THD_ER_CLC_PREV = -32768
LV_ER_CLC_PREV_INI = 1

Recurrence: every ENRD segment task

Activation: *LV_ENA_SEG_T_MES = 1*
***And** LC_MIS_INH = 0*

Deactivation: *LV_ENA_SEG_T_MES = 0*
***Or** LC_MIS_INH = 1*

Actions on Deactivation event:

```
        THD_ER_CLC = -32768
        LV_ER_CLC_PREV_INI = 1
```

Formula section:

Convention for combustion mode determination:

```
#If  NC_CMB_CONF = AFS_AFL Or AFS_AFL_S
    If      OPM_AV = 3          (homogeneous-stratified)
    Then    xx = "HOM_S"
    EndIf
#EndIf
#If  NC_CMB_CONF = AFS_S  Or AFS_AFL_S
    If      OPM_AV = 1          (stratified)
    Then    xx = "S"
    EndIf
#EndIf
If          OPM_AV <> 1  and  OPM_AV <> 3          (homogeneous)
Then        xx = "AFS"
EndIf
```

Convention for transmission type determination:

```
If(1)  LV_AT = 1
Then(1)
            yy= AT
ElseIf(2)   LV_VAR_TCT = 1
     Then(2)
            yy= TCT
     Else(2)
            yy= MT
     Endif(2)
Endif(1)
```

Calculation of THD_ER_CLC:

```
If          LV_IS = 1
Then        THD_ER_CLC = IP_THD_xx_ER_IS_yy(N,LOAD_MIS)
Else        THD_ER_CLC = IP_THD_xx_ER_yy(N,LOAD_MIS)
End
```

Initialisation of THD_ER_CLC_PREV:

```
If          LV_ER_CLC_PREV_INI = 1
then        THD_ER_CLC_PREV = THD_ER_CLC
            LV_ER_CLC_PREV_INI = 0
endif
```

Positive gradient limitation of THD_ER_CLC:

```
If          THD_ER_CLC - THD_ER_CLC_PREV > C_THD_ER_CLC_GRD
then        THD_ER_CLC = THD_ER_CLC_PREV + C_THD_ER_CLC_GRD
else        if      THD_ER_CLC > THD_ER_CLC_PREV +
                                        C_THD_ER_CLC_GRD
            then    THD_ER_CLC = THD_ER_CLC_PREV +
                                                C_THD_ER_CLC_GRD
            else    THD_ER_CLC = THD_ER_CLC  // no change!
            endif
endif
```

Store previous value of THD_ER_CLC:

```
THD_ER_CLC_PREV = THD_ER_CLC
```

Calibration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
C_THD_ER_CLC_GRD	1	0...7FFFH	0...32767	1	[µs]
Gradient to limit the misfire detection threshold THD_ER_CLC in positive direction					
C_N_32_MAX_IS_MIS	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed to apply specific ER detection threshold in idle speed					
IP_THD_AFS_ER_AT	12*8	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_AFS_ER_AT	12	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_AFS_ER_AT	8	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in AFS mode, AT vehicle transmission type					
IP_THD_AFS_ER_MT	12*8	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_AFS_ER_MT	12	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_AFS_ER_MT	8	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in AFS mode, MT vehicle transmission type					
IP_THD_AFS_ER_TCT	12*8	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_AFS_ER_TCT	12	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_AFS_ER_TCT	8	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in AFS mode, TCT vehicle transmission type					
IP_THD_AFS_ER_IS_MT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_AFS_ER_IS_MT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_THD_AFS_ER_IS_MT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in AFS combustion mode & in idle speed, MT vehicle					
IP_THD_AFS_ER_IS_AT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_AFS_ER_IS_AT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_THD_AFS_ER_IS_AT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in AFS combustion mode & in idle speed, AT vehicle					
IP_THD_AFS_ER_IS_TCT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_AFS_ER_IS_TCT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_THD_AFS_ER_IS_TCT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in AFS combustion mode & in idle speed, TCT vehicle					
#IF NC_CMB_CONF = AFS_AFL Or AFS_AFL_S					
IP_THD_HOM_S_ER_AT	9*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_HOM_S_ER_AT	9	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_HOM_S_ER_AT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in HOM_S mode, AT vehicle transmission type					
IP_THD_HOM_S_ER_MT	9*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_HOM_S_ER_MT	9	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_HOM_S_ER_MT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in HOM_S mode, MT vehicle transmission type					
IP_THD_HOM_S_ER_TCT	9*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_HOM_S_ER_TCT	9	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_THD_HOM_S_ER_TCT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in HOM_S mode, TCT vehicle transmission type					
IP_THD_HOM_S_ER_IS_AT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_HOM_S_ER_IS_AT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_THD_HOM_S_ER_IS_AT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in HOM_S combustion mode & in idle speed, AT vehicle transmission type					
IP_THD_HOM_S_ER_IS_MT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_HOM_S_ER_IS_MT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_THD_HOM_S_ER_IS_MT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in HOM_S combustion mode & in idle speed, MT vehicle transmission type					
IP_THD_HOM_S_ER_IS_TCT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_HOM_S_ER_IS_TCT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_IP_THD_HOM_S_ER_IS_TCT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in HOM_S combustion mode & in idle speed, TCT vehicle transmission type					
#ENDIF					
#IF NC_CMB_CONF = AFS_S Or AFS_AFL_S					
IP_THD_S_ER_AT	9*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_S_ER_AT	9	0...1FE0H	0...8160	1	[rpm]

LDP_LOAD_MIS_IP_THD_S_ER_AT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in S mode, AT vehicle transmission type					
IP_THD_S_ER_MT	9*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_S_ER_MT	9	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_S_ER_MT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in S mode, MT vehicle transmission type					
IP_THD_S_ER_TCT	9*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_S_ER_TCT	9	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_S_ER_TCT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in S mode, TCT vehicle transmission type					
IP_THD_S_ER_IS_AT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_S_ER_IS_AT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_S_ER_IS_AT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in S combustion mode & in idle speed, AT vehicle transmission type					
IP_THD_S_ER_IS_MT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_S_ER_IS_MT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_S_ER_IS_MT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in S combustion mode & in idle speed, MT vehicle transmission type					
IP_THD_S_ER_IS_TCT	4*6	8000...7FFFH	-32768...32767	1	[-]
LDP_N_IP_THD_S_ER_IS_TCT	4	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS_IP_THD_S_ER_IS_TCT	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
ER threshold for misfire detection in S combustion mode & in idle speed, TCT vehicle transmission type					
#ENDIF					

51.4.12 Zero Load Line for Misfire detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LOAD_MIN_MIS	V/O	0...7FFFH	0...99.99694	3.0518e-3	[%]
Normalised engine "zero" load for misfire detection					

Input data:

N	TCO	LV_AT	NC_CMB_CONF
LC_MIS_INH	AMP	OPM_AV	LV_VAR_TCT

FUNCTION DESCRIPTION:

The zero load line is a function of the engine speed and depends on the engine combustion mode (for GDI application), coolant temperature and atmospheric pressure.

Application conditions:

Initialisation: Variable is initialised to 0 at reset, at engine stop **Or** at LC_MIS_INH 0 to 1 transition

Recurrence: every segment task

Activation: LV_ENA_SEG_T_MES = 1

And LC_MIS_INH = 0

Deactivation: LV_ENA_SEG_T_MES = 0

Or LC_MIS_INH = 1

Formula section:

Minimum engine load and zero load lines

Convention for combustion mode determination:

```
#If NC_CMB_CONF = AFS_AFL Or AFS_AFL_S
    If      OPM_AV = 3          (homogeneous-stratified)
    Then    xx = "HOM_S"
    EndIf
#EndIf
#If NC_CMB_CONF = AFS_S Or AFS_AFL_S
    If      OPM_AV = 1          (stratified)
    Then    xx = "S"
    EndIf
#EndIf

If          OPM_AV <> 1  and  OPM_AV <> 3 (homogeneous)
```

```
Then        xx = "AFS"
EndIf
```

Convention for transmission type determination:

```
If(1) LV_AT = 1
Then(1)
            yy= AT
ElseIf(2)   LV_VAR_TCT = 1
    Then(2)
            yy= TCT
    Else(2)
            yy= MT
    Endif(2)
Endif(1)
```

```
LOAD_MIN_MIS = IP_LOAD_MIN_MIS_xx_yy(N, TCO)
                    + IP_LOAD_MIN_OFS_AMP_MIS_xx_yy(N, AMP)
```

Calibration data:

Name	Dim	Hex. Limits	Phys. Limits	Resol.	Unit
IP_LOAD_MIN_MIS_AFS_MT	8*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_IP_LOAD_MIN_MIS_AFS_MT	8	0...1FE0H	0...8160	1	[rpm]
LDP_TCO_IP_LOAD_MIN_MIS_AFS_MT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed and coolant temperature in AFS combustion mode, with MT vehicle					
IP_LOAD_MIN_MIS_AFS_AT	8*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_IP_LOAD_MIN_MIS_AFS_AT	8	0...1FE0H	0...8160	1	[rpm]
LDP_TCO_IP_LOAD_MIN_MIS_AFS_AT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed and coolant temperature in AFS combustion mode, with AT vehicle					
IP_LOAD_MIN_MIS_AFS_TCT	8*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_IP_LOAD_MIN_MIS_AFS_TCT	8	0...1FE0H	0...8160	1	[rpm]
LDP_TCO_IP_LOAD_MIN_MIS_AFS_TCT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed and coolant temperature in AFS combustion mode, with TCT vehicle					
IP_LOAD_MIN_OFS_AMP_MIS_AFS_MT	8*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_IP_LOAD_AMP_MIS_AFS_MT	8	0...1FE0H	0...8160	1	[rpm]
LDP_AMP_IP_LOAD_AMP_MIS_AFS_MT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in AFS with MT vehicle					
IP_LOAD_MIN_OFS_AMP_MIS_AFS_AT	8*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_IP_LOAD_AMP_MIS_AFS_AT	8	0...1FE0H	0...8160	1	[rpm]
LDP_AMP_IP_LOAD_AMP_MIS_AFS_AT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in AFS with AT vehicle					
IP_LOAD_MIN_OFS_AMP_MIS_AFS_TCT	8*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_IP_LOAD_AMP_MIS_AFS_TCT	8	0...1FE0H	0...8160	1	[rpm]
LDP_AMP_IP_LOAD_AMP_MIS_AFS_TCT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in AFS with TCT vehicle					
#IF NC_CMB_CONF = AFS_AFL Or AFS_AFL_S					

Name	Dim	Hex. Limits	Phys. Limits	Resol.	Unit
IP_LOAD_MIN_MIS_HOM_S_MT	6*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_32_IP_LOAD_MIN_HOM_S_MT	6	0...FFH	0...8160	32	[rpm]
LDP_TCO_IP_LOAD_MIN_HOM_S_MT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed and coolant temperature in HOM_S combustion mode, with MT vehicle					
IP_LOAD_MIN_MIS_HOM_S_AT	6*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_32_IP_LOAD_MIN_HOM_S_AT	6	0...FFH	0...8160	32	[rpm]
LDP_TCO_IP_LOAD_MIN_HOM_S_AT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed and coolant temperature in HOM_S combustion mode, with AT vehicle					
IP_LOAD_MIN_MIS_HOM_S_TCT	6*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_32_IP_LOAD_MIN_HOM_S_TCT	6	0...FFH	0...8160	32	[rpm]
LDP_TCO_IP_LOAD_MIN_HOM_S_TCT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed and coolant temperature in HOM_S combustion mode, with TCT vehicle					
IP_LOAD_MIN_OFS_AMP_HOM_S_MT	6*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_32_IP_LOAD_AMP_HOM_S_MT	6	0...FFH	0...8160	32	[rpm]
LDP_AMP_IP_LOAD_AMP_HOM_S_MT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in HOM_S with MT vehicle					
IP_LOAD_MIN_OFS_AMP_HOM_S_AT	6*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_32_IP_LOAD_AMP_HOM_S_AT	6	0...FFH	0...8160	32	[rpm]
LDP_AMP_IP_LOAD_AMP_HOM_S_AT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in HOM_S with AT vehicle					
IP_LOAD_MIN_OFS_AMP_HOM_S_TCT	6*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_32_IP_LOAD_AMP_HOM_S_TCT	6	0...FFH	0...8160	32	[rpm]
LDP_AMP_IP_LOAD_AMP_HOM_S_TCT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in HOM_S with TCT vehicle					
#ENDIF					
#IF NC_CMB_CONF = AFS_S Or AFS_AFL_S					
IP_LOAD_MIN_MIS_S_MT	6*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_32_IP_LOAD_MIN_MIS_S_MT	6	0...FFH	0...8160	32	[rpm]
LDP_TCO_IP_LOAD_MIN_MIS_S_MT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed anc coolant temperature in S combustion mode, with MT vehicle					
IP_LOAD_MIN_MIS_S_AT	6*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_32_IP_LOAD_MIN_MIS_S_AT	6	0...FFH	0...8160	32	[rpm]
LDP_TCO_IP_LOAD_MIN_MIS_S_AT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed anc coolant temperature in S combustion mode, with AT vehicle					
IP_LOAD_MIN_MIS_S_TCT	6*6	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_N_32_IP_LOAD_MIN_MIS_S_TCT	6	0...FFH	0...8160	32	[rpm]
LDP_TCO_IP_LOAD_MIN_MIS_S_TCT	6	0...FEH	-48...142.5	0.75	[°C]
Zero load curve vs engine speed anc coolant temperature in S combustion mode, with TCT vehicle					
IP_LOAD_MIN_OFS_AMP_MIS_S_MT	6*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_32_IP_LOAD_AMP_MIS_S_MT	6	0...FFH	0...8160	32	[rpm]
LDP_AMP_IP_LOAD_AMP_MIS_S_MT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in S with MT vehicle					
IP_LOAD_MIN_OFS_AMP_MIS_S_AT	6*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_32_IP_LOAD_AMP_MIS_S_AT	6	0...FFH	0...8160	32	[rpm]
LDP_AMP_IP_LOAD_AMP_MIS_S_AT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in S with AT vehicle					
IP_LOAD_MIN_OFS_AMP_MIS_S_TCT	6*7	0...7FFFH	-50...49.99389	3.0518e-3	[%]
LDP_N_32_IP_LOAD_AMP_MIS_S_TCT	6	0...FFH	0...8160	32	[rpm]
LDP_AMP_IP_LOAD_AMP_MIS_S_TCT	7	0...FFFFH	0...5434	0.0829175	[hPa]
Extra load from zero load curve vs. engine speed & atmospheric pressure in S with AT vehicle					
#ENDIF					

51.4.13 Misfire Combustion mode transients informations

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_CMB_TRA_MIS	V/O	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to combustion mode transients					

Input data:

OPM_AV	NC_CMB_CONF		

Application conditions:

While the combustion manager is in transient state, the misfire detection can be disabled to have a proper base for evaluation of the ER-misfire detection thresholds, which are defined for the stationary modes (AFS, AFL & S).

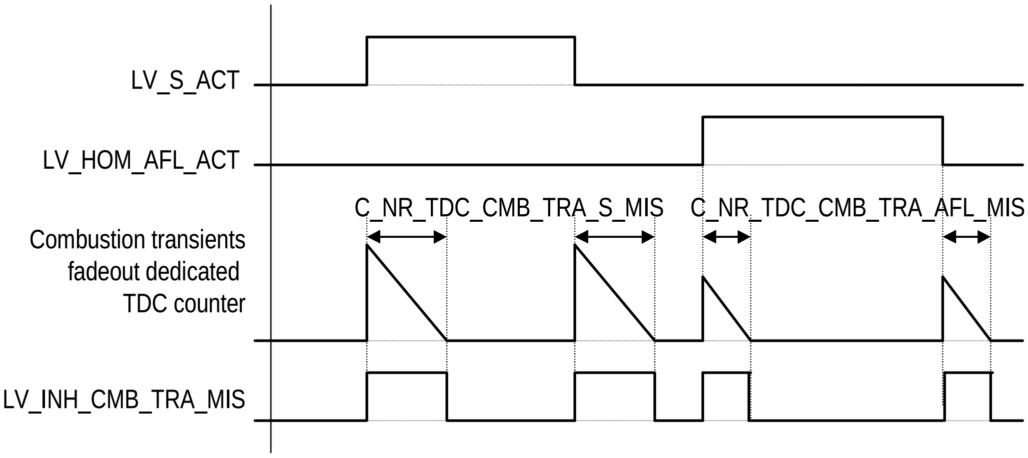
This function must be called before the generic specification named "Generic fade out conditions for misfire detection"

Initialisation: *at ECU reset, at Engine Stop* ***Or*** *at LC_MIS_INH 0 to 1 transition*

LV_INH_CMB_TRA_MIS = 1

Formula section:

```
#If  NC_CMB_CONF = AFS_S Or AFS_AFL_S
     If     OPM_AV = x -> 1      (x is different of 1)
     Then   LV_INH_CMB_TRA_MIS is set to 1 during C_NR_TDC_CMB_TRA_S_MIS tdc's
     EndIf
#EndIf

#If  NC_CMB_CONF = AFS_AFL Or AFS_AFL_S
     If     OPM_AV = x -> 3      (x is different of 3)
     Then   LV_INH_CMB_TRA_MIS is set to 1 during C_NR_TDC_CMB_TRA_AFL_MIS tdc's
     EndIf
#EndIf
     If     OPM_AV = x -> 2      (x is different of 2)
     Then   LV_INH_CMB_TRA_MIS is set to 1 during C_NR_TDC_CMB_TRA_HOM_MIS tdc's
     EndIf
```



Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
#IF NC_CMB_CONF = AFS_S Or AFS_AFL_S					
C_NR_TDC_CMB_TRA_S_MIS	1	0...FFH	0...255	1	[-]
Fade out TDC's duration when engine combustion manager enter or leave stratified combustion mode					
#ENDIF					
#IF NC_CMB_CONF = AFS_AFL Or AFS_AFL_S					
C_NR_TDC_CMB_TRA_AFL_MIS	1	0...FFH	0...255	1	[-]
Fade out TDC's duration when engine combustion manager enter or leave air/fuel lean combustion mode					
#ENDIF					
C_NR_TDC_CMB_TRA_HOM_MIS	1	0...FFH	0...255	1	[-]
Fade out TDC s duration when engine combustion manager enter in homogen combustion mode					

51.5 Generic misfire parameters and fade-out conditions

51.5.1 Misfire detection engine parameters

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LOAD_MIS	V/O	0...7FFFH	0...99.99694	3.0518e-3	[%]
Normalised engine load for misfire detection - sampled at current misfire segment					
LOAD_MIS_1	-	0...7FFFH	0...99.99694	3.0518e-3	[%]
Normalised engine load for misfire detection - 1 misfire segment before					
LOAD_MIS_2	-	0...7FFFH	0...99.99694	3.0518e-3	[%]
Normalised engine load for misfire detection - 2 misfire segment before					
LOAD_MIS_3	V	0...7FFFH	0...99.99694	3.0518e-3	[%]
Normalised engine load for misfire detection - 3 misfire segment before					
LOAD_GRD_MIS	V/O	0...7FFFH	0...99.99694	3.0518e-3	[%]
Normalised engine load gradient for misfire detection					

Input data:

LC_MIS_INH	TCO	MAP	LV_ENA_SEG_T_MES
TQI_AV	AMP	MAF	FAC_GAIN_LOAD_MIS

FUNCTION DESCRIPTION:

In order to cover all the combustion states of an engine (Hom. AFS, AFL & Strat.) in torque based engine control, a general variable LOAD_MIS that represents the ratio of engine load – whatever should be the load definition depending on engine combustion state (*eg. MAF in Hom. and TQI_AV in Strat.*) – is defined for the purpose of the Misfire function and the calculation of the ER-misfire detection thresholds.

Depending on the project choice configuration (*with C_CONF_LOAD_MIS*), in Homogenous combustion mode this Misfire load variable should be either MAF, MAP or TQI_AV. In the other hand this Misfire load variable is always TQI_AV in Stratified combustion mode.

Legally, the presence of engine misfire in the engine operating region is bounded by the positive torque line (i.e. engine load with the transmission in neutral) and an engine speed value.

The zero load line is a function of the engine speed and depends on the engine combustion mode (for GDI application), coolant temperature and atmospheric pressure.

Application conditions:

Initialisation: All variables are initialised to 0 at reset, at engine stop **Or** at LC_MIS_INH 0 to 1 transition

LOAD_GRD_MIS is first time calculated when LOAD_MIS stack have been fill in (4 segments).

Recurrence: every segment task

Activation: LV_ENA_SEG_T_MES = 1
And LC_MIS_INH = 0

Deactivation: LV_ENA_SEG_T_MES = 0
Or LC_MIS_INH = 1

Formula section:

Normalised engine load dedicated to misfire detection

For each misfire segment, the following stack is serviced:

```
LOAD_MIS_3 =LOAD_MIS_2
LOAD_MIS_2 =LOAD_MIS_1
LOAD_MIS_1 = LOAD_MIS
If        C_CONF_LOAD_MIS = 0
    Or    LV_S_ACT = 1
Then      LOAD_MIS_Temp = TQI_AV / IP_TQI_AV_MAX_MIS(N_32)      // TQI_AV based
ElseIf    C_CONF_LOAD_MIS = 1
Then      LOAD_MIS_Temp = MAF / IP_MAF_MAX_MIS(N32)             // MAF based
Else      LOAD_MIS_Temp = MAP / C_MAP_MAX_MIS                   // MAP based
EndIf
LOAD_MIS = FAC_GAIN_LOAD_MIS * LOAD_Temp        // Application incidence correction
LOAD_GRD_MIS = | LOAD_MIS - LOAD_MIS_3 |
```

Note : LOAD_MIS calculation steps are checked against division by zero
LOAD_MIS_Temp is a non visible, temporary value

Calibration data:

Name	Dim	Hex. Limits	Phys. Limits	Resol.	Unit
C_MAP_MAX_MIS	1	0...FFFFH	0...5434	0.0829175	[hPa]
Maximum manifold air pressure value to determine engine load ratio scale for misfire detection					
C_CONF_LOAD_MIS	1	0...2H	0...2	1	[-]
LOAD_MIS switch configuration versus engine combustion mode: (=0) LOAD_MIS always relative to TQI_AV (=1) LOAD_MIS relative to MAF in homog. & rel. to TQI_AV in stratif. mode, (=2) LOAD_MIS relative to MAP in homog. & rel. to TQI_AV in stratif. mode					
IP_TQI_AV_MAX_MIS	8	0...7FFFH	0...1023.97	0.03125	[Nm]
LDP_N_32_IP_TQI_AV_MAX_MIS	8	0...FFH	0...8160	32	[rpm]
Maximum torque value to determine engine load ratio scale for misfire detection					
IP_MAF_MAX_MIS	6	0...FFFFH	0...1389	0.0211948	[mg/stk]
LDP_N_32_IP_MAF_MAX_MIS	6	0...FFH	0...8160	32	[rpm]
Maximum air mass value to determine engine load ratio scale for misfire detection					

51.5.2 Generic fade out conditions for misfire detection

Output data:

Name	Mode	Hex. Limit	Phys. Limit	Resol.	Unit
MIS_DET_CDN_APP_INH	V	0...FFFFH	0...65535	1	[-]
Misfiring APP INC detection fade out requests carrier word (before masking)					
MIS_DET_CDN_APP_INH_NR	V	0...FFH	0...255	1	[-]
Misfiring APP INC detection fade out requests number (after masking)					
MAP_DIF_MIS	V	0...FFFFH	0...5434	0.0829175	[hPa]
Manifold air pressure difference between 2 misfire segment tasks					
IGA_DIF_MIS	V	0...FFH	0...95.625	0.375	[°CRK]
Absolute Ignition angle difference between 2 misfire segment tasks					
IGA_BAS_DIF_MIS	V	0...FFH	-35.625...60	0.375	[°CRK]
Ignition angle difference between IGA_BAS_COR_MV and IGA_AV, segment sample					
IGA_MIS_2	-	0...FFH	-35.625...60	0.375	[°CRK]
Ignition angle 2 misfire segment before					
IGA_MIS_1	-	0...FFH	-35.625...60	0.375	[°CRK]
Ignition angle 1 misfire segment before					
IGA_MIS	V	0...FFH	-35.625...60	0.375	[°CRK]
Ignition angle at current misfire segment					
LV_REQ_APP_INH_MIS	V/O	0...1H	0...1	1	[-]
Misfire detection APP INC fade out request flag					
LV_INH_MAP_DIF_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to MAP_DIF condition					
LV_INH_LOAD_GRD_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to LOAD_GRD_MIS condition					
LV_INH_TPS_GRD_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to TPS gradient condition					
LV_INH_IGA_DIF_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to IGA_DIF condition					
LV_INH_ACC_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to Air Conditionner transient condition					

Input data:

IGA_BAS_COR_MV	MAF	TPS_AV	LV_INH_APP_DET_MIS
LV_IGA_GRD_ACT	N_32	LV_S_ACT	LV_ACCOUT_RLY
TCO	LV_STATE_RR	LV_INH_FTL_L_DET_MIS	IGA_AV[NC_CYL_NR]
CTR_T_ZDLY_MIS	TPS_GRD	LV_INH_CMB_TRA_MIS	NC_CMB_CONF
LV_STATE_CRK_OSC	LV_INH_OBD_DET_MIS	LV_ENA_ER	SEG_NR
LC_MIS_INH	LV_IS	N	MAP

General information:

Generic fade out (LV_REQ_APP_INH_MIS) is based on all conditions who could create unreliable conditions for misfire detection :

- Engine gradients (engine load, throttle, combustion mode & ignition angle transients).
- Crankshaft jolt effects who can distrub detection process (rough road, drivetrain oscillations, air conditionner & accessoiries activation/deactivation).
- OBDI errors on one of the sensors used by the misfire monitoring process.

All these different conditions are merge into a carrier (MIS_DET_CDN_APP_INH).

Application conditions:

Initialisation: at ECU reset, at Engine Stop **Or** at LC_MIS_INH 0 to 1 transition

LV_REQ_APP_INH_MIS = 1
MIS_DET_CDN_APP_INH = 2047
MIS_DET_CDN_APP_INH_NR = 11
MAP_DIF_MIS = 0
IGA_DIF_MIS = 0
IGA_BAS_DIF_MIS = 0
IGA_MIS_2 = 0
IGA_MIS_1 = 0
IGA_MIS = 0

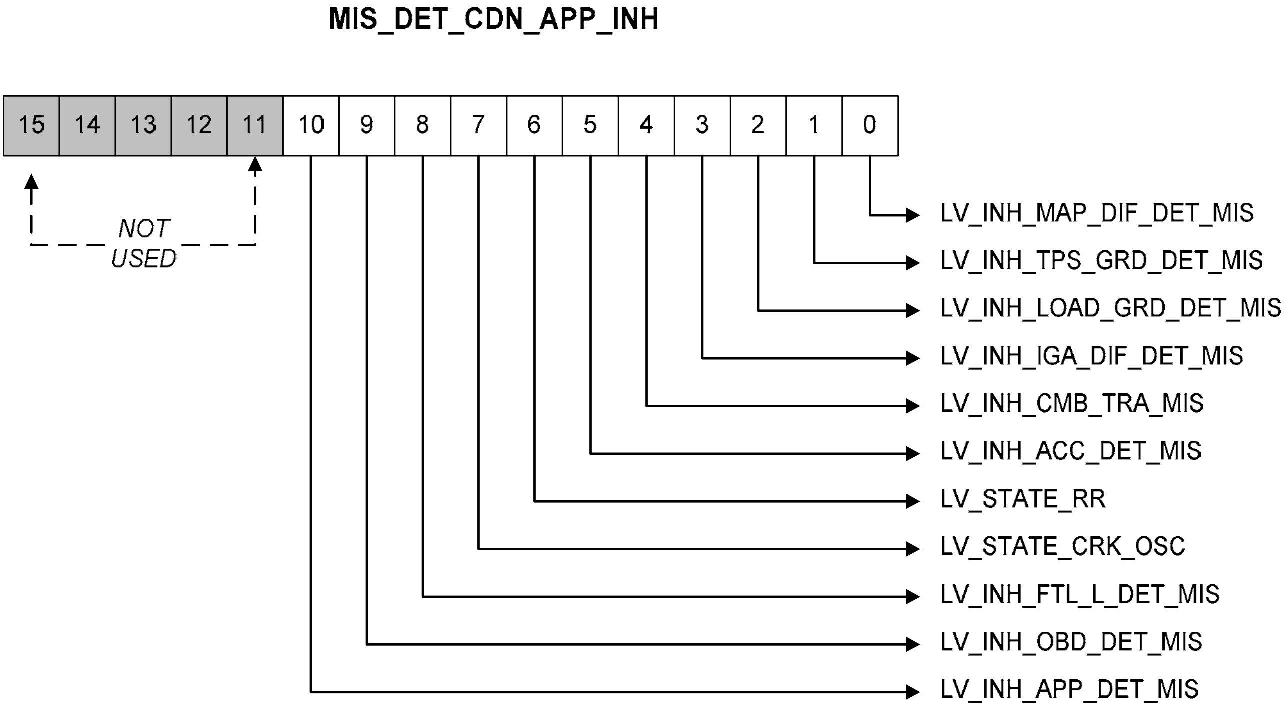
Recurrence: segment task for data process
10 ms for free running timers

Activation: LV_ENA_ER = 1
And LC_MIS_INH = 0

Deactivation: LV_ENA_ER = 0
Or LC_MIS_INH = 1

Remark : Timers used for delay are 10 ms free running decounters, they are not linked to activation/deactivation conditions described above.

51.5.2.1 Generic fade out condition carrier definition



<u>remark</u> : MIS_DET_CDN_APP_INH is updated at current misfire task

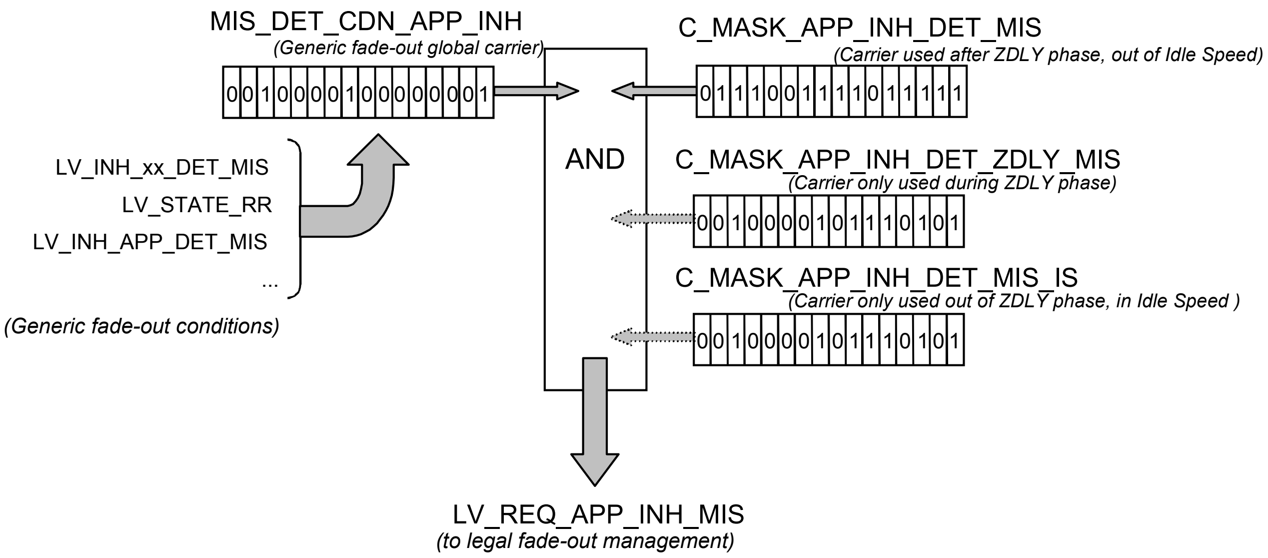
51.5.2.2 Configurable generic fade-out management

Generic fade out management during zero delay phase, nominal phase & idle speed engine state can be configurated via 3 bitfield masks that allow to take in account or not some conditions in a phase and not in the other.

If the corresponding bit in the MIS_DET_CDN_APP_INH carrier structure is set to 0 in C_MASK_APP_INH_DET_ZDLY_MIS, C_MASK_APP_INH_DET_MIS or C_MASK_APP_INH_DET_MIS_IS calibration, then the corresponding condition will not fade-out misfire detection.

For the definition of C_MASK_APP_INH_DET_ZDLY_MIS, C_MASK_APP_INH_DET_MIS & C_MASK_APP_INH_DET_MIS_IS masks please refer to the legal requirements & customer recommendations.

<u>Overview:</u>



<u>Formula section:</u>

```
If(1) CTR_T_ZDLY_MIS != 0
Then(1)    // ZDLY phase
    MIS_DET_CDN_APP_INH_NR =
            sum(MIS_DET_CDN_APP_INH & C_MASK_APP_INH_DET_ZDLY_MIS)
    (bitfield operations)
Else(1)
    If(2)    LV_IS = 1
    Then(2)        // Idle Speed engine state
            MIS_DET_CDN_APP_INH_NR =
                    sum(MIS_DET_CDN_APP_INH & C_MASK_APP_INH_DET_MIS_IS)
            (bitfield operations)
    Else(2)        // Out of idle speed engine state
            MIS_DET_CDN_APP_INH_NR =
```

```
            sum(MIS_DET_CDN_APP_INH & C_MASK_APP_INH_DET_MIS)
        (bitfield operations)
    EndIf(2)
EndIf(1)

If    MIS_DET_CDN_APP_INH_NR != 0
Then        LV_REQ_APP_INH_MIS = 1
Else        LV_REQ_APP_INH_MIS = 0
EndIf
```

51.5.2.3 Maximum manifold air-pressure gradient

Due to trailing throttle / acceleration transition problems, it could be necessary to disable misfire detection for a short period when the manifold air-pressure gradient exceeds an applicable value.

Application conditions:

In all homogeneous combustion mode, Misfire detection is disabled when the amount of the MAP gradient exceeds IP_MAP_DIF_MAX_MIS(MAP)

Initialisation: *at ECU reset, at Engine Stop **Or** at LC_MIS_INH 0 to 1 transition*

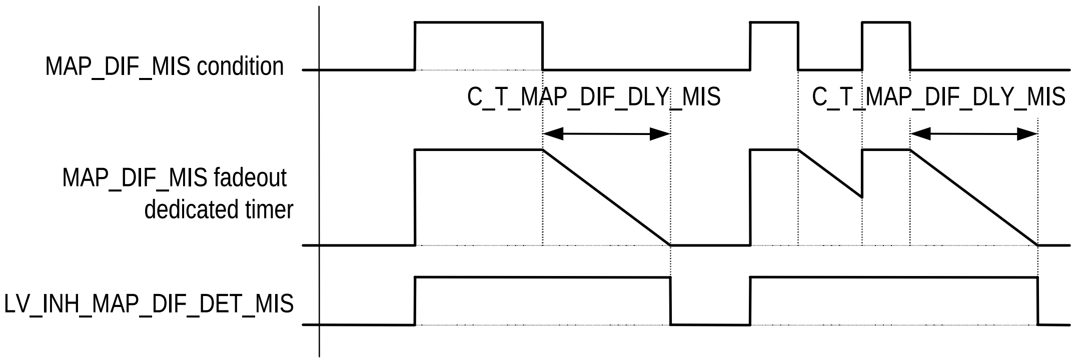
LV_INH_MAP_DIF_DET_MIS = 1

Formula section:

$MAP\_DIF\_MIS = | MAP_{(n)} - MAP_{(n-2)} |$

```
If    LV_S_ACT = 0
      And MAP_DIF_MIS > IP_MAP_DIF_MAX_MIS(MAP)
Then
      After condition rising edge triggering, LV_INH_MAP_DIF_DET_MIS flag is set to 1 as long as condition is true.
      After condition falling edge triggering, LV_INH_MAP_DIF_DET_MIS flag is hold to 1 for a period of C_T_MAP_DIF_DLY_MIS.
EndIf
```

Fade out behaviour summary:



51.5.2.4 Maximum throttle gradient

Due to trailing throttle / acceleration transient problems at low load, it is necessary to disable misfire detection for a short period when the throttle gradient exceeds an applicable value.

Application conditions:

Misfire detection can be disabled when the amount of the throttle gradient exceeds the applicable value hereunder mentioned. But during the 0-delay activation of the misfire monitoring, driver-induced fade-out is allowed:

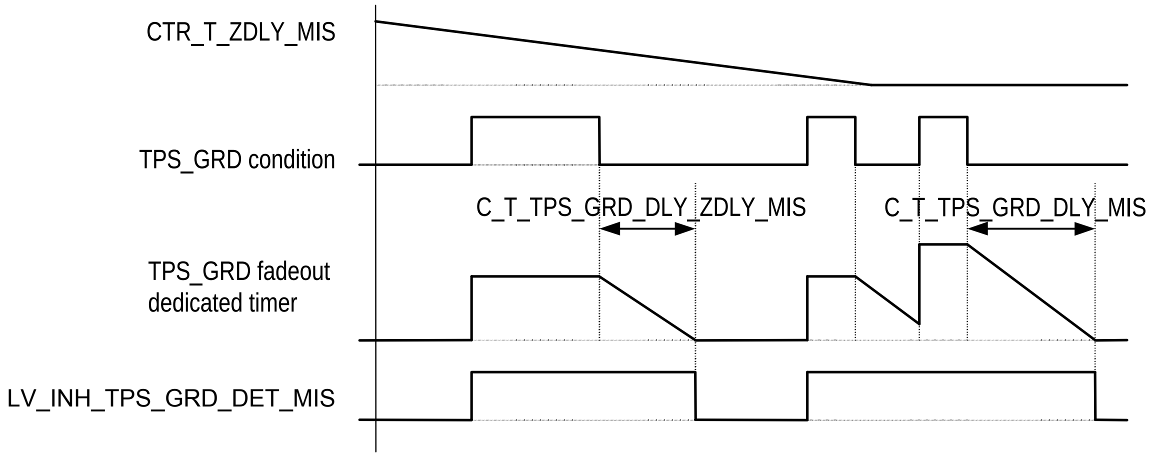
Initialisation: *at ECU reset, at Engine Stop **Or** at LC_MIS_INH 0 to 1 transition*

LV_INH_TPS_GRD_DET_MIS = 1

Formula section:

```
If(1) CTR_T_ZDLY_MIS = 0
Then(1)
    If(2)   TPS_GRD > IP_TPS_GRD_MAX_MIS(TPS_AV)
    Then(2)
            After condition rising edge triggering, LV_INH_TPS_GRD_DET_MIS flag is set to 1 as long as condition is true.
            After condition falling edge triggering, LV_INH_TPS_GRD_DET_MIS flag is hold to 1 for a period of C_T_TPS_GRD_DLY_MIS.
    EndIf(2)
Else(1)
    If(2)   TPS_GRD > IP_TPS_GRD_ZDLY_MIS(TPS_AV)
    Then(2)
            After condition rising edge triggering, LV_INH_TPS_GRD_DET_MIS flag is set to 1 as long as condition is true.
            After condition falling edge triggering, LV_INH_TPS_GRD_DET_MIS flag is hold to 1 for a period of C_T_TPS_GRD_DLY_ZDLY_MIS.
    EndIf(2)
EndIf(1)
```

Fade out behaviour summary:



51.5.2.5 Maximum engine load gradient

Due to trailing throttle / acceleration transient problems at low load, it could be necessary to disable misfire detection for a short period when the engine load exceeds an applicable value.

Application conditions:

Misfire detection can be disabled when the amount of engine load gradient exceeds a calibration value, this value is switched according combustion mode. Fade out delay is also specific to combustion mode.

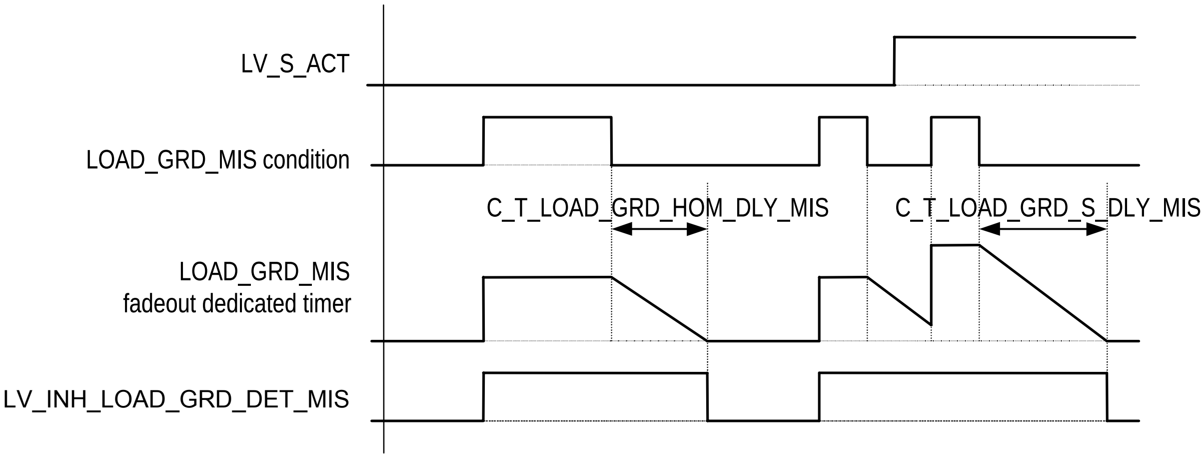
Initialisation: *at ECU reset, at Engine Stop **Or** at LC_MIS_INH 0 to 1 transition*

LV_INH_LOAD_GRD_DET_MIS = 1

Formula section:

```
If(1) LV_S_ACT = 0
Then(1)
    If(2)   LOAD_GRD_MIS > IP_LOAD_GRD_MIS(LOAD_MIS)
    Then(2)
            After condition rising edge triggering, LV_INH_LOAD_GRD_DET_MIS flag is set to 1 as long as condition is true.
            After condition falling edge triggering, LV_INH_LOAD_GRD_DET_MIS flag is hold to 1 for a period of C_T_LOAD_GRD_HOM_DLY_MIS.
    EndIf(2)
EndIf(1)
#If  NC_CMB_CONF = AFS_S Or AFS_AFL_S
    If(1)   LV_S_ACT = 1
    Then(1)
            If(2)   LOAD_GRD_MIS > IP_LOAD_GRD_S_MIS(LOAD_MIS)
            Then(2)
                    After condition rising edge triggering, LV_INH_LOAD_GRD_DET_MIS flag is set to 1 as long as condition is true.
                    After condition falling edge triggering, LV_INH_LOAD_GRD_DET_MIS flag is hold to 1 for a period of C_T_LOAD_GRD_S_DLY_MIS.
            EndIf(2)
    EndIf(1)
#EndIf
```

Fade out behaviour summary:



51.5.2.6 Ignition retardation without change limitation

Various functions or CAN requests (e.g.. gear-shift signal. torque reduction. etc...) cause an ignition - timing retardation, without change limitation. Misfire detection can be disabled when the amount of the ignition timing retardation is too important.

Several criterion are available to disable the misfire monitoring according the ignition angle behaviour. Two relative criterions dedicated to focus on IGA transients, one in idle speed and one in part load, plus one absolute criterion dedicated to focus on too important IGA values compared to the

Initialisation: *at ECU reset, at Engine Stop* ***Or*** *at LC_MIS_INH 0 to 1 transition*

LV_INH_IGA_DIF_DET_MIS = 1

Formula section:

For each TDC, the following stacks are serviced after having calculated IGA_AV_x.

IGA_DIF_MIS = | IGA_MIS_2 - IGA_MIS |

IGA_BAS_DIF_MIS = IGA_BAS_COR_MV - IGA_MIS

IGA_MIS_2 = IGA_MIS_1

IGA_MIS_1 = IGA_MIS

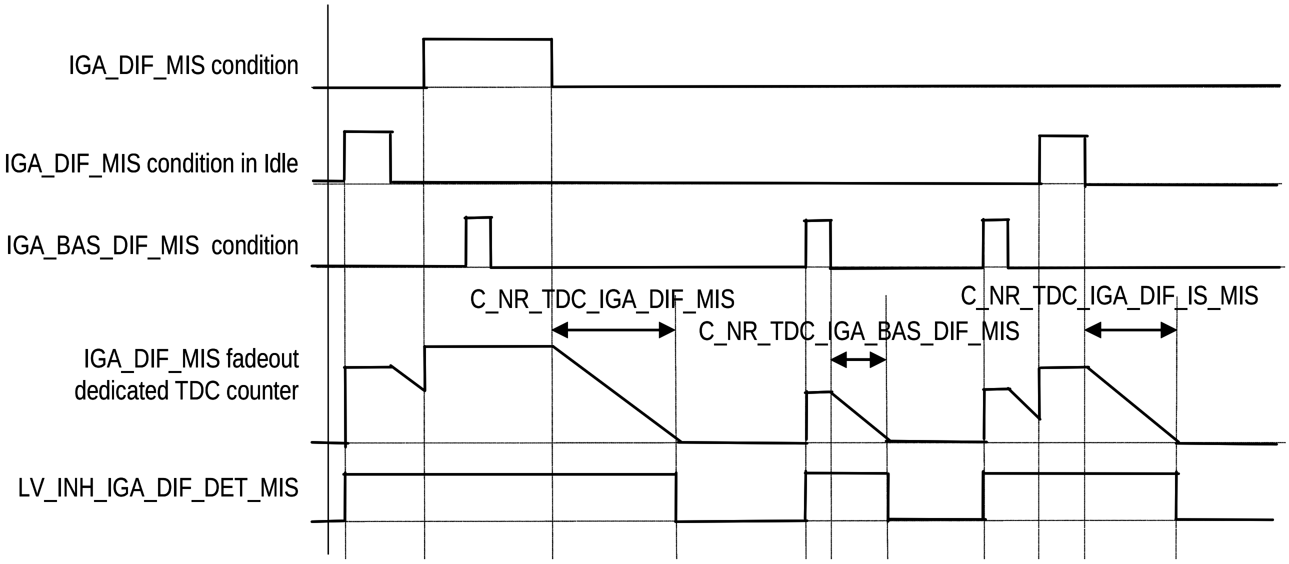
IGA_MIS = IGA_AV[SEG_NR]

```
If(1) LV_S_ACT = 0
Then(1)
    If(2)   LV_IS = 1
            And N_32 < C_N_32_MAX_IGA_IS_MIS
    Then(2)
            If(3)    IGA_DIF_MIS > IP_DELTA_IGA_IS_MIS(TCO, LOAD_MIS_2)
            Then(3)      After condition rising edge triggering, LV_INH_IGA_DIF_DET_MIS
                         flag is set to 1 as long as condition is true.
            Else(3)      After condition falling edge triggering,
                         LV_INH_IGA_DIF_DET_MIS flag is hold to 1 for a period of
                         C_NR_TDC_IGA_DIF_IS_MIS.
```

```
                        When this period is exceeded without retriggering,
                        LV_INH_IGA_DIF_DET_MIS releases to 0.
            EndIf(3)
        Else(2) // out of idle speed conditions
            If(3)       IGA_DIF_MIS > IP_DELTA_IGA_MIS(N_32, LOAD_MIS)
            Then(3)     After condition rising edge triggering, LV_INH_IGA_DIF_DET_MIS
                        flag is set to 1 as long as condition is true.
            Else(3)     After condition falling edge triggering,
                        LV_INH_IGA_DIF_DET_MIS flag is hold to 1 for a period of
                        C_NR_TDC_IGA_DIF_MIS.
                        When this period is exceeded without retriggering,
                        LV_INH_IGA_DIF_DET_MIS releases to 0.
            EndIf(3)
        EndIf(2)
        If(2)   LV_IGA_GRD_ACT = 0      // out of IGA engine start calculation phase
            And IGA_BAS_DIF_MIS > IP_DELTA_IGA_BAS_MIS(N_MIS,LOAD_MIS)
        Then(2)     After condition rising edge triggering, LV_INH_IGA_DIF_DET_MIS flag is
                    set to 1 as long as condition is true.
        Else(2)     After condition falling edge triggering, LV_INH_IGA_DIF_DET_MIS flag is
                    hold to 1 for a period of C_NR_TDC_IGA_BAS_DIF_MIS.
                    When this period is exceeded without retriggering,
                    LV_INH_IGA_DIF_DET_MIS releases to 0.
        EndIf (2)
    EndIf(1)
```

// note: the TDC decounter is retriggered only if the considered preload calibration is greater than the decounter actual value.

Fade out behaviour summary:



51.5.2.7 Combustion mode transients

Application conditions:

While the combustion manager is in transient state, the misfire detection can be disabled to have a proper base for evaluation of the ER-misfire detection thresholds, which are defined for the stationary modes (AFS, AFL & S). See definiton in Application Incidences file. (LV_INH_CMB_TRA_MIS = 1).

51.5.2.8 Air - conditioning compressor activation

When the air - conditioning compressor is switched on, an additional load is briefly applied to the engine.
This load jump can cause a segment period jump and crankshaft vibration, depending on the engine operating state and load request.

Application conditions:

Misfire detection can be suppressed for the applicable constant period C_T_ACCOUT_DLY_MIS, starting if the air - conditioning compressor is switched on or off (LV_ACCOUT_RLY changes from 0 to 1 or from 1 to 0), during this period LV_INH_ACC_DET_MIS is set to 1.

Initialisation: *at ECU reset, at Engine Stop* ***Or*** *at LC_MIS_INH 0 to 1 transition*

LV_INH_ACC_DET_MIS = 1

Formula section:

```
If    LV_ACCOUT_RLY = 0 -> 1
      Or     LV_ACCOUT_RLY = 1 -> 0
Then         LV_INH_ACC_DET_MIS is set to 1 during C_T_ACCOUT_DLY_MIS
EndIf
```

51.5.2.9 Rough road detection

Application conditions:

Misfire detection can be suppressed when the status of the *rough road detection* is active (LV_STATE_RR = 1).

51.5.2.10 Crankshaft oscillation detection

Application conditions:

Misfire detection can be suppressed when the status of the *crankshaft oscillation detection* is active (LV_STATE_CRK_OSC = 1).

51.5.2.11 Low fuel level

Application conditions:

Misfire detection can be suppressed when the low fuel level is detected (LV_INH_FTL_L_DET_MIS=1).

51.5.2.12 OBDI diagnosis fade-out

Application conditions:

LV_INH_OBD_DET_MIS combines all OBD project specific error bits to generate an input for the generic misfire detection fade-out conditions. *See definition in Misfire detection - Application Incidences file.* Misfiring detection can be suppressed if LV_INH_OBD_DET_MIS = 1

51.5.2.13 Application incidences fade-out

Application conditions:

LV_INH_APP_DET_MIS combines all project specific special conditions to generate an input for the generic misfire detection fade-out conditions. *See definition in Misfire detection - Application Incidences file*. Misfiring detection can be suppressed if LV_INH_APP_DET_MIS = 1

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MASK_APP_INH_DET_ZDLY_MIS	1	0...FFFFH	0...65535	1	[-]
MIS_DET_CDN_APP_INH carrier structure fade out configuration mask (misfire detection in zero delay phase)					
C_MASK_APP_INH_DET_MIS	1	0...FFFFH	0...65535	1	[-]
MIS_DET_CDN_APP_INH carrier structure fade out configuration mask (detection in nominal phase out of IS)					
C_MASK_APP_INH_DET_MIS_IS	1	0...FFFFH	0...65535	1	[-]
MIS_DET_CDN_APP_INH carrier structure fade out configuration mask (detection in nominal phase & in IS)					
C_T_MAP_DIF_DLY_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when manifold air pressure gradient has been detected.					
C_T_TPS_GRD_DLY_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when maximum throttle gradient has been detected.					
C_T_TPS_GRD_DLY_ZDLY_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when maximum throttle gradient has been detected during zero delay starting phase					
C_NR_TDC_IGA_DIF_MIS	1	0...FFH	0...255	1	[-]
Fade out duration when IGA difference per TDC has been detected.					
C_NR_TDC_IGA_DIF_IS_MIS	1	0...FFH	0...255	1	[-]
Fade out duration when relative IGA criterion in idle speed has been detected					
C_NR_TDC_IGA_BAS_DIF_MIS	1	0...FFH	0...255	1	[-]
Fade out duration when absolute IGA criterion has been detected					
C_T_LOAD_GRD_HOM_DLY_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when maximum LOAD_GRD_MIS has been detected in homogeneous combustion modes					
C_T_ACCOUT_DLY_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when air - conditioning compressor has been switched on.					
C_N_32_MAX_IGA_IS_MIS	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed to apply iga_dif_mis in idle speed					
IP_TPS_GRD_MAX_MIS	10	0...FFH	0...2987.5	11.715686	[°TPS/s]
LDP_TPS_AV_IP_TPS_GRD_MAX_MIS	10	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Maximum throttle gradient threshold for fade out condition.					
IP_TPS_GRD_ZDLY_MIS	6	0...FFH	0...2987.5	11.715686	[°TPS/s]
LDP_TPS_AV_IP_TPS_GRD_ZDLY_MIS	6	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Maximum throttle gradient threshold for fade out condition during zero delay misfire monitoring activation					
IP_LOAD_GRD_MIS	12	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_LOAD_MIS_IP_LOAD_GRD_MIS	12	0...7FFFH	0...99.99694	3.0518e-3	[%]
Maximum LOAD_MIS gradient for fade out condition in homogeneous combustion modes					
IP_MAP_DIF_MAX_MIS	12	0...FFFFH	0...5434	0.0829175	[hPa]
LDP_MAP_IP_MAP_DIF_MAX_MIS	12	0...FFFFH	0...5434	0.0829175	[hPa]
Maximum MAP gradient for fade out condition.					
IP_DELTA_IGA_MIS	8*8	0...FFH	0...95.625	0.375	[°CRK]
LDP_N_32_IP_DELTA_IGA_MIS	8	0...FFH	0...8160	32	[rpm]
LDP_LOAD_MIS_IP_DELTA_IGA_MIS	8	0...7FFFH	0...99.99694	3.0518e-3	[%]
Spark advance angle difference at part load for misfire inhibition					
IP_DELTA_IGA_IS_MIS	4*4	0...FFH	0...95.625	0.375	[°CRK]
LDP_TCO_IP_DELTA_IGA_IS_MIS	4	0...FEH	-48...142.5	0.75	[°C]
LDP_LOAD_IP_DELTA_IGA_IS_MIS	4	0...7FFFH	0...99.99694	3.0518e-3	[%]
Spark advance angle difference in idle speed for misfire inhibition					
IP_DELTA_IGA_BAS_MIS	8*8	0...FFH	-35.625...60	0.375	[°CRK]
LDP_N_32_IP_DELTA_IGA_BAS_MIS	8	0...FFH	0...8160	32	[rpm]
LDP_LOAD_IP_DELTA_IGA_BAS_MIS	8	0...7FFFH	0...99.99694	3.0518e-3	[%]
Absolute spark advance angle difference for misfire inhibition					
#IF NC_CMB_CONF = AFS_S Or AFS_AFL_S					
IP_LOAD_GRD_S_MIS	12	0...7FFFH	0...99.99694	3.0518e-3	[%]
LDP_LOAD_MIS_IP_LOAD_GRD_S_MIS	12	0...7FFFH	0...99.99694	3.0518e-3	[%]
Maximum LOAD_MIS gradient for fade out condition in stratified combustion mode					
C_T_LOAD_GRD_S_DLY_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when maximum LOAD_GRD_MIS has been detected in stratified combustion mode					
#ENDIF					

51.6 Legal Misfire Detection Fade-Out Conditions

51.6.1 General information

This chapter is relative to misfire detection fade out management as defined by legal texts linked to functionnal OBD diagnosis (OBDII, EOBD...).

Continuous misfire monitoring from engine starting instant:

- ***<u>Engine start instant</u>*** *: the point when the engine reaches a speed 150 rpm below the normal, warmed-up idle speed (as determined in the drive position for vehicles equipped with an automatic transmission) (CARB definition).*
  - *<u>One engine cycle (2 crankshaft revolutions) is allowed to initialise misfire detection process</u> (see CTR_TDC_ZDLY_MIS using).*

Fade out conditions may disable misfire monitoring with the following conditions:

- **Minimum engine speed** : N_IS_SP - 150 rpm *(US)*
- **Maximum engine speed** : Redline engine speed *(US)* or 4500 rpm *(EC)*
- **Minimum engine load** :

  Zero load line (engine positive torque line)

  A line joining the point (3000 rpm, zero load) and a point (6000 rpm, 135 hPa *(or 4 inches Hg)* above the zero load line)

- **Minimum coolant or ambiant temperature**

  Below 20°F (US) or -7°C (EC) *(equivalent),*

  Or If Engine start coolant temperature is below 20°F, until current temperature reaches warm up engine temperature 70°F.

- **Minimum atmospheric pressure**

  Elevations above 8000 feet above sea level (US) or 2 500 meters (EC)

- **Time after start**

  A 5s time after start fade-out can be allowed for EC market

  Such kind of fade out is prohibited for US market (see CARB requirement for zero delay misfire monitoring activation)

- **Cylinder shut-off**

  During fuel cut-off phase, on a specific cylinder shut-off or the number of cylinder in shut-off is too important, misfire diagnosis may be fade out *(no injection/combustion occurs)*

51.6.2 Misfire detection fade out management

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_TDC_ZDLY_MIS	V	0...FFFFH	0...65535	1	[-]
TDC decounter for disabled diagnostic					
CTR_T_ZDLY_MIS	V/O	0...FFFFH	0...655.35	0.01	[s]
10 ms decounter for diagnostic fade out (general purpose)					
LV_DIAG_MIS	V/O	0...1H	0...1	1	[-]
Nominal Misfire Monitoring Phase					
LV_ZDLY_DIAG_MIS	V/O	0...1H	0...1	1	[-]
Zero Delay Misfire Monitoring Phase					
STATE_DIAG_MIS	V	0H 1H 2H 3H 4H	INI_PHA PREP_0_DLY 0_DLY_MON PREP_MON NOM_MON	0	[-]
Misfire Detection Phase State Machine					

Input data:

N	LV_RUN_ENG	LV_ES	

FUNCTION DESCRIPTION:

CARB requests an activation of the misfire monitoring and diagnosis as off engine start. Particular fade-out conditions are required during the first 5 seconds since engine running. In this section the flag representing the request is defined, and is used in several other modules that are concerned with this requirement *(LC_REQ_ZDLY_MIS = 1).*

For non-CARB applications, this special monitoring is not required and has to be de-activated *(LC_REQ_ZDLY_MIS = 0).*

A state-based approach is taken for the activation of the misfire detection algorithms, distinguishing the separate monitoring during the first 5 seconds and the 'nominal' monitoring afterwards, and preparatory phases. During the monitoring phases, the normal fade-out mechanisms are used, corrected for some fade-outs that are not allowed during the first 5 seconds.

Application conditions:

Activation:

Engine Start LV_RUN_ENG = 1

Deactivation:

Engine Stop LV_RUN_ENG = 0 **Or** ECU reset

Remark: At deactivation, ALL internal and external variables related to the legal fade out misfire detection conditions must be reset to 0.
In nominal mode (after engine start and zero delay period), the state machine executes no operations.

Formula section:

At engine start-up (LV_RUN_ENG switches from 0 to 1) the system starts in STATE_DIAG_MIS = 0 (initialisation phase).

In the case of a CARB-application, special monitoring takes placed during C_T_ZDLY_MIS seconds. The timer is started at detection of engine running:

```
If        LC_REQ_ZDLY_MIS = 1
    And   STATE_DIAG_MIS = 0
Then      CTR_T_ZDLY_MIS = C_T_ZDLY_MIS
EndIf
```

Transition initialisation state to preparation states for 0-delay OR to nominal monitoring

```
If        LC_REQ_ZDLY_MIS = 1
    And   STATE_DIAG_MIS = 0
    And   N > C_N_MIN_ZDLY_MIS
Then      Zero-Delay Activation of Misfire Detection
          STATE_DIAG_MIS = 1
          CTR_TDC_ZDLY_MIS = C_NR_TDC_ZDLY_MIS
ElseIf    LC_REQ_ZDLY_MIS = 0
    And   STATE_DIAG_MIS = 0
    And   N > C_N_MIN_ZDLY_MIS
Then      Standard Activation of Misfire Detection
          STATE_DIAG_MIS = 3
          CTR_TDC_ZDLY_MIS = C_NR_TDC_ZDLY_MIS
Else      Initialisation Phase
          LV_DIAG_MIS = 0
          LV_ZDLY_DIAG_MIS = 0
EndIf
```

Transition back to the initialisation phase

```
If        LV_ES = 1
    And   STATE_DIAG_MIS <> 0
Then      In case of engine stop, return to initialisation phase
          STATE_DIAG_MIS = 0
EndIf
```

Transition preparation state 0-delay monitoring to 0-delay monitoring state

```
If          LC_REQ_ZDLY_MIS = 1           CARB application
    And     STATE_DIAG_MIS = 1            0-delay preparation state
    And     N > C_N_MIN_ZDLY_MIS          engine speed over threshold
    And     CTR_TDC_ZDLY_MIS = 0          1 engine cycle passed
Then        STATE_DIAG_MIS = 2
            LV_ZDLY_DIAG_MIS = 1
EndIf
```

Transition preparation nominal state to nominal monitoring state

```
If          LC_REQ_ZDLY_MIS = 0           standard application
    And     STATE_DIAG_MIS = 3            preparation monitoring state
    And     N > C_N_MIN_ZDLY_MIS          engine speed over threshold
    And     CTR_TDC_ZDLY_MIS = 0          1 engine cycle passed
Then        STATE_DIAG_MIS = 4
            LV_DIAG_MIS = 1
EndIf
```

Transition 0-delay monitoring state to nominal monitoring state

```
If  LC_REQ_ZDLY_MIS = 1                   CARB application
    And     STATE_DIAG_MIS = 2            0-delay monitoring state
    And     N > C_N_MIN_ZDLY_MIS          engine speed over threshold
    And     CTR_TDC_ZDLY_MIS = 0          1 engine cycle passed
    And     CTR_T_ZDLY_MIS = 0            timer finished (typically 5 seconds)
Then        STATE_DIAG_MIS = 4
            LV_DIAG_MIS = 1
            LV_ZDLY_DIAG_MIS = 0
EndIf
```

The counter CTR_T_ZDLY_MIS is decremented at every time instance of 10 ms when STATE_DIAG_MIS != 0.

The counter CTR_TDC_ZDLY_MIS is decremented with 1 at every TDC.

(both counters are saturated to 0)

Rermark: During the 0-delay misfire detection, no fade-outs are allowed for crankshaft oscillation, rough road detection and zero-load line conditions (see appropriate sections). However, it is allowed to have so-called driver-induced fade-outs (e.g. gradient on pedal value). This latter section has been added.

At hot resets, the system starts again in the initialisation phase.

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
LC_REQ_ZDLY_MIS	1	0...1H	0...1	1	[-]
Activation of CARB-required 0-delay monitoring					
C_N_MIN_ZDLY_MIS	1	0...1FE0H	0...8160	1	[rpm]
Minimum monitoring engine speed for engine start monitoring phase					
C_T_ZDLY_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Duration for engine start specific misfire monitoring (5s according CARB)					
C_NR_TDC_ZDLY_MIS	1	0...FFH	0...255	1	[-]
Standard TDC count disablement tdc number before monitoring (engine start phase)					

51.6.3 Misfire detection legal fade out conditions

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
MIS_DET_CDN_INH	V	0...3FH	0...63	1	[-]
Misfiring detection fade out requests carrier byte (before masking)					
MIS_DET_CDN_INH_NR	V	0...FFH	0...255	1	[-]
Misfiring detection fade out requests number (after masking)					
LV_REQ_INH_MIS	V/O	0...1H	0...1	1	[-]
Misfire detection fade out request flag					
LV_PUC_DET_MIS	V	0...1H	0...1	1	[-]
Fuel Cut-Off dedicated to misfire detection, including a delay time for fuel reactivation					
LV_INH_N_MAX_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to maximum engine speed condition					
LV_INH_N_MIN_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to minimum engine speed condition					
LV_INH_LOAD_MIN_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to zero load line condition					
LV_INH_TCO_MIN_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flagdue to minimum temperature condition					
LV_INH_AMP_MIN_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flagdue to minimum atmospheric pressure condition					
LV_INH_ST_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to engine start condition					
LV_INH_IV_OFF_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out request flag due to cylinder shut-off condition					
LV_IV_OFF_DET_MIS	V	0...1H	0...1	1	[-]
Misfire detection fade out flag due to cylinder shut-off, phased with misfire detection index					

Input data:

N	LOAD_MIS	LOAD_MIN_MIS	TCO
AMP	LV_ST	SEG_NR	CTR_T_ZDLY_MIS
LV_AT	LV_DC	INH_INJ	TCO_ST
LV_REQ_APP_INH_MIS	LV_INJ_CUT	SUM_INH_INJ	C_N_MIN_ZDLY_MIS
LC_MIS_INH			

FUNCTION DESCRIPTION:

General information:

Legal fade out (LV_REQ_INH_MIS) is based on all fade out conditions allowed by the legal texts for continuous misfire monitoring.

All these different fade-out conditions are merge into a global carrier (MIS_DET_CDN_INH).

Timers used for fade-out conditions delays are free running timers who are not deactivated according following conditions.

Application conditions: *(global to Misfire detection legal fade out conditions chapter)*

Initialisation: at Reset, Engine Stop, **Or** on Deactivation event

LV_PUC_DET_MIS = 0

LV_IV_OFF_DET_MIS = 1

LV_REQ_INH_MIS = 1

MIS_DET_CDN_INH = 127

MIS_DET_CDN_INH_NR = 7

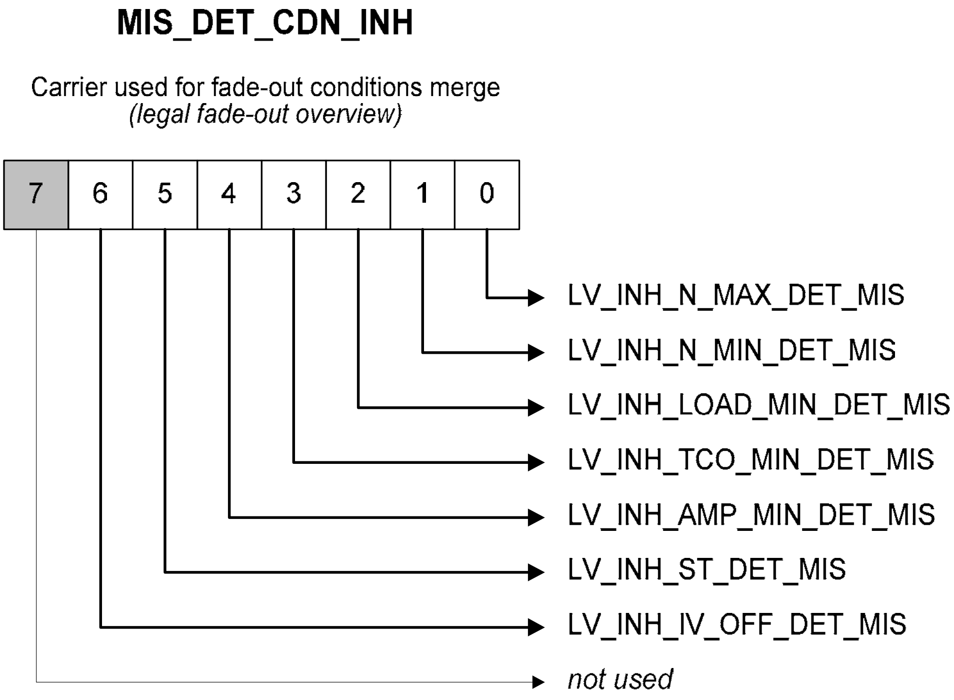
Recurrence: every segment task

Activation: *LC_MIS_INH = 0*

Deactivation: *LC_MIS_INH = 1*

51.6.3.1 Fade out carrier definition

Misfire fade out conditions carrier:



51.6.3.2 Configurable Fade-out Management

Application conditions:

Fade out management during zero delay phase & nominal phase can be configurated via 2 bitfield masks that allow to take in account or not some conditions in a phase and not in the other.

If the corresponding bit in the MIS_DET_CDN_INH carrier structure is set to 0 in C_MASK_INH_DET_ZDLY_MIS or C_MASK_INH_DET_MIS calibration, then the corresponding condition will not fade-out misfire detection.

For the definition of C_MASK_INH_DET_ZDLY_MIS & C_MASK_INH_DET_MIS masks please refer to the legal requirements & customer recommendations.

Formula section:

```
If    CTR_T_ZDLY_MIS ≠ 0
Then
      MIS_DET_CDN_INH_NR = sum(MIS_DET_CDN_INH & C_MASK_INH_DET_ZDLY_MIS)
      (bitfield operations)
Else
      MIS_DET_CDN_INH_NR = sum(MIS_DET_CDN_INH & C_MASK_INH_DET_MIS)
      (bitfield operations)
EndIf
If    MIS_DET_CDN_INH_NR ≠ 0
      Or      LV_REQ_APP_INH_MIS = 1      // Fade out request linked to misfire detection method
Then          LV_REQ_INH_MIS = 1          // General fade out request
Else          LV_REQ_INH_MIS = 0
EndIf
```

51.6.3.3 Maximum engine speed fade-out

Application conditions:

The misfire monitoring can be inhibated when the engine speed reachs the maximum engine speed imposed by the market target : Redline engine speed for US applications or 4500 rpm for EC applications.

Moreover this information is use to stop misfire monitoring process for high speed if not required (limitation of cpu load especially for european application).

Formula section:

```
If    ( LV_AT = 1   And   N > C_N_MAX_MIS_AT )
      Or      ( LV_AT = 0   And   N > C_N_MAX_MIS_MT )
Then          LV_INH_N_MAX_DET_MIS = 1
Else          LV_INH_N_MAX_DET_MIS = 0
EndIf
```

51.6.3.4 Minimum engine speed fade-out

The irregular engine operation test is performed as soon as the above-mentioned condition is disabled.

Important:

- C_N_MIN_MIS is defined by the current configuration of the index used for misfire detection in the current application. C_N_MIN_MIS should be lower than C_N_MIN_ZDLY_MIS.
- C_N_MIN_ZDLY_MIS is defined as the engine speed set-point for a warm engine minus 150 rpm. After passing below the C_N_MIN_ZDLY_MIS threshold, one is allowed one engine cycle (meaning for example for a 4-cylinder engine 4 tdc's) before reactivating the output of the misfire detection algorithm. Misfire detection is suppressed for a period of C_NR_TDC_INH_MIS tdc's.
- It should be checked in each application that within the allowed number of TDC's proper misfire detection index values have been re-obtained. Typically, it should be possible to restart misfire detection earlier.

Application conditions:

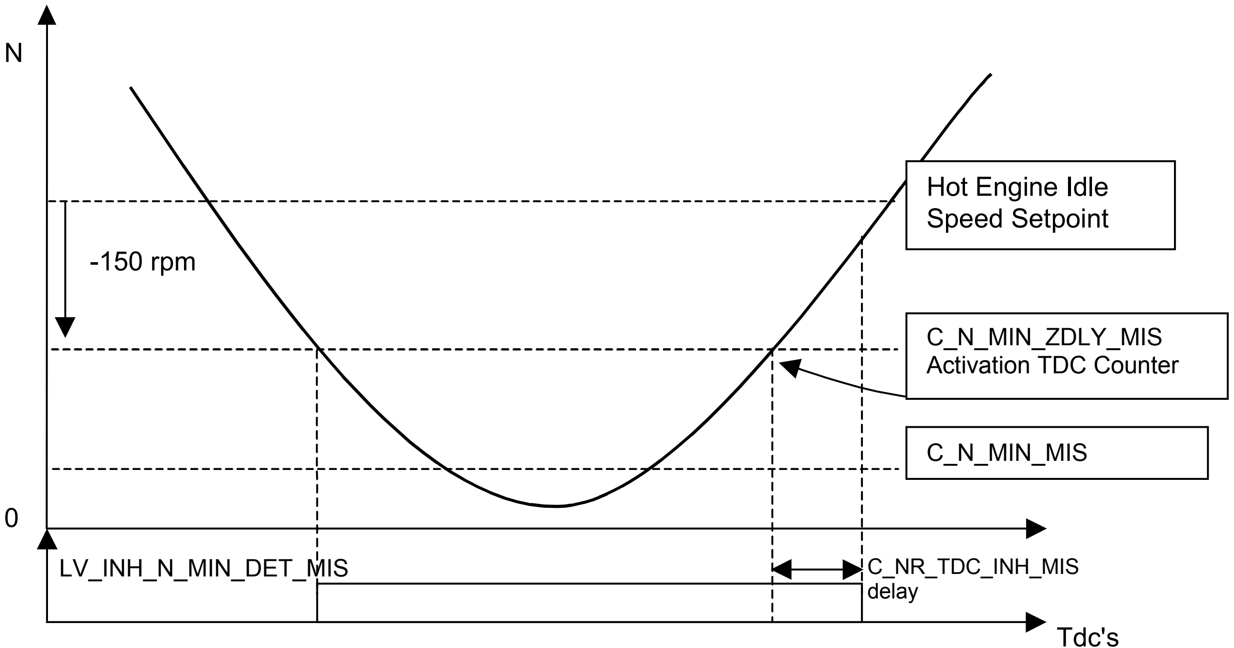
N < C_N_MIN_ZDLY_MIS

Engine speed below CARB requirements *(used for US applications)*.

Typically, once N < C_N_MIN_ZDLY_MIS, the output of the misfire detection is inhibited, allthough the calculation is still valid. C_NR_TDC_ZDLY_MIS tdc's decounter is started at passing engine speed C_N_MIN_ZDLY_MIS again. Once the decounter is at 0, misfire detection index output is allowed.

N < C_N_MIN_MIS

Engine speed below misfire detection index resolution *(used for CE applications)*. If the engine speed falls below C_N_MIN_MIS.



Explanation on the mechanism to re-activate the misfire detection after an engine speed drop

As the misfiring test is not mandatory and risky over a certain engine speed threshold, a fade out condition is tested on engine rpm (not allowed for US market).

Activation: *LV_INH_N_MAX_DET_MIS = 0*

Deactivation: *LV_INH_N_MAX_DET_MIS = 1*

Action on deactivation event: *LV_INH_N_MIN_DET_MIS = 1*

Formula section:

```
If    N < C_N_MIN_ZDLY_MIS
      Or    N < C_N_MIN_MIS
Then        LV_INH_N_MIN_DET_MIS = 1
Else        LV_INH_N_MIN_DET_MIS = 0
                caution : see specific reactivation conditions above  when
                N > C_N_MIN_ZDLY_MIS, C_NR_TDC_INH_MIS delay used
EndIf
```

51.6.3.5 Minimum engine load

Application conditions:

Below a specific load (zero load), fluctuations in engine speed cannot be detected, even in the event of misfire. In this case, the misfire detection is disabled.

Activation: *LV_INH_N_MAX_DET_MIS = 0*

Deactivation: *LV_INH_N_MAX_DET_MIS = 1*

Action on deactivation event: *LV_INH_LOAD_MIN_DET_MIS = 1*

Formula section:

```
    If      LOAD_MIS < LOAD_MIN_MIS
    Then    LV_INH_LOAD_MIN_DET_MIS = 1
    Else    LV_INH_LOAD_MIN_DET_MIS = 0
    EndIf
```

51.6.3.6 Minimum coolant temperature

Application conditions:

Misfire detection can be disabled when the coolant temperature is below a threshold (20°F / -7°C). Additionally when the coolant temperature at engine start is below this threshold, the misfire monitoring can be disable on that driving cycle until the current engine temperature reaches a warm up temperature threshold (70°F / 21°C):

Activation: *LV_INH_N_MAX_DET_MIS = 0*

Deactivation: *LV_INH_N_MAX_DET_MIS = 1*

Action on deactivation event: *LV_INH_TCO_MIN_DET_MIS = 1*

Formula section:

LV_TEMP : temporary bit used to check only one occurence per engine start

Initialisation : *LV_TEMP = 0 at ECU reset* ***Or*** *LV_DC 0 -> 1*

```
    If      TCO < C_TCO_MIN_MIS
```

```
            Or     ( TCO_ST < C_TCO_MIN_MIS
                   And    TCO < C_TCO_WUP_MIN_MIS
                   And    LV_TEMP = 0 )
      Then  LV_INH_TCO_MIN_DET_MIS = 1
      Else  LV_INH_TCO_MIN_DET_MIS = 0
            LV_TEMP = 1
      EndIf
```

51.6.3.7 Minimum atmospheric pressure

Application conditions:

Misfire detection can be disabled when the atmospheric pressure is below a threshold:

Activation: LV_INH_N_MAX_DET_MIS = 0

Deactivation: LV_INH_N_MAX_DET_MIS = 1

Action on deactivation event: LV_INH_AMP_MIN_DET_MIS = 1

Formula section:

```
If     AMP < C_AMP_MIN_MIS
Then   LV_INH_AMP_MIN_DET_MIS = 1
Else   LV_INH_AMP_MIN_DET_MIS = 0
EndIf
```

51.6.3.8 After start

Application conditions:

Activation: LV_INH_N_MAX_DET_MIS = 0

Deactivation: LV_INH_N_MAX_DET_MIS = 1

Action on deactivation event: LV_INH_ST_DET_MIS = 1

Misfire detection can be suppressed (see legal requirements - forbidden for CARB zero delay requirements) for a C_T_DLY_ST_MIS duration after the engine exits the operating state start (LV_ST), during this period LV_INH_ST_DET_MIS is set to 1, even if C_T_DLY_ST_MIS = 0.

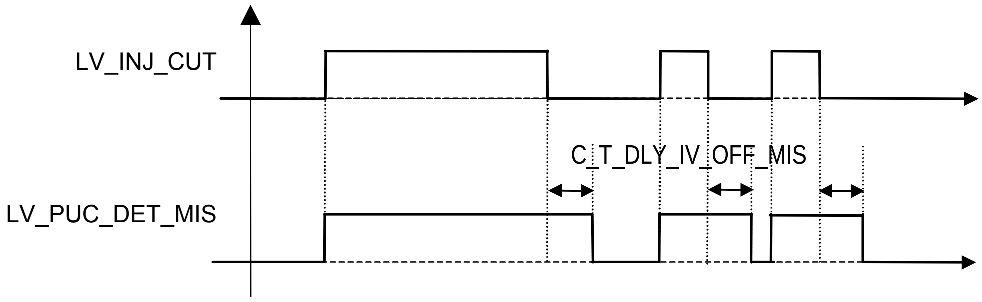
51.6.3.9 Injection shut-off

Application conditions:

Misfire detection can be suppressed for the cylinder(s), for which injection has been shut-off.

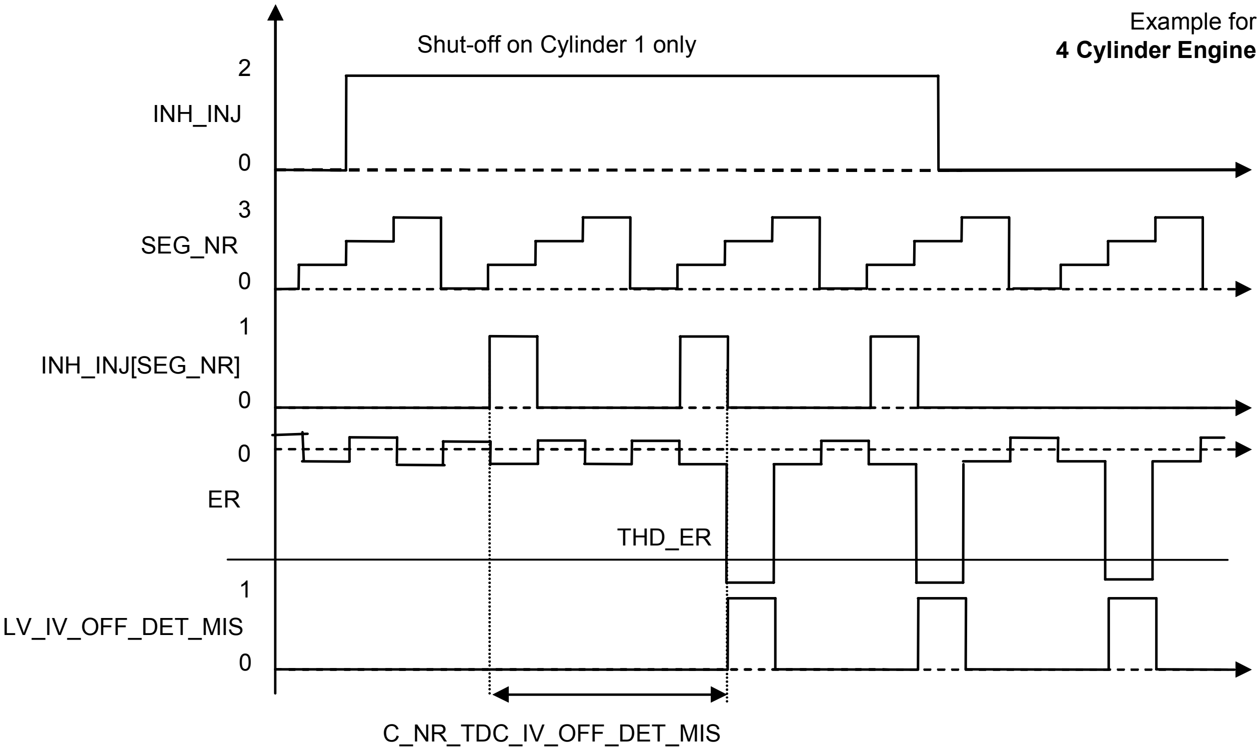
General fuel shut-off:

LV_INJ_CUT indicates a general fuel shut-off. LV_PUC_DET_MIS bit includes a calibration delay time (C_T_DLY_IV_OFF_MIS) triggered at fuel cut-off condition falling edge, in a way to start the misfire detection only after crankshaft oscillations due to fuel reactivation.



Cylinder specific shut-off:

If the current cylinder checked by misfire detection is shut-off by any EMS function even MISF limp home functionnality), a fade-out can be apply for the misfire detection of this cylinder. Due to the delay between the misfire index and the injection current informations, a calibration delay (C_NR_TDC_IV_OFF_DET_MIS) has been introduced on LV_IV_OFF_DET_MIS. This delay synchronises MISF & INJR informations in a way to apply correctly the fade-out on the proper cylinder during individual cylinder shut-off operations, and to be able the perform the misfire monitoring on the other cylinders who are not in shut-off mode.



Limit number of cylinders shut-off:

If the number of shut-off cylinders reaches a calibration data (C_SUM_INH_IV_MAX_DET_MIS), reliable misfire detection is impossible. In such case is preferable to inhibate misfire detection.

Formula section:

General fuel shut-off:

If LV_INJ_CUT = 1 *// All cylinders are shut off*

Then LV_PUC_DET_MIS = 1

Else LV_PUC_DET_MIS = 0 after C_T_DLY_IV_OFF_MIS delay time

```
EndIf
```

Cylinder fuel shut-off:

```
If    INH_INJ[SEG_NR] = 1          // indexed by cylinder bit position
Then      LV_IV_OFF_DET_MIS = 1 after a C_NR_TDC_IV_OFF_DET_MIS delay
Else      LV_IV_OFF_DET_MIS = 0 after a C_NR_TDC_IV_OFF_DET_MIS delay
EndIf
```

Activation: LV_INH_N_MAX_DET_MIS = 0

Deactivation: LV_INH_N_MAX_DET_MIS = 1

Action on deactivation event: LV_INH_IV_OFF_DET_MIS = 1

Formula section:

General definition:

```
If    LV_PUC_DET_MIS = 1
      Or  LV_IV_OFF_DET_MIS = 1
      Or  SUM_INH_INJ >= C_SUM_INH_IV_MAX_MIS
Then      LV_INH_IV_OFF_DET_MIS = 1
Else      LV_INH_IV_OFF_DET_MIS = 0
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MASK_INH_DET_ZDLY_MIS	1	0...FFH	0...255	1	[-]
MIS_DET_CDN_INH carrier structure fade out configuration mask (misfire detection zero delay phase)					
C_MASK_INH_DET_MIS	1	0...FFH	0...255	1	[-]
MIS_DET_CDN_INH carrier structure fade out configuration mask (misfire detection nominal phase)					
C_AMP_MIN_MIS	1	0...FFFFH	0...5434	8.29E-02	[hPa]
Minimum atmospheric pressure for misfire detection					
C_N_MIN_MIS	1	0...1FE0H	0...8160	1	[rpm]
Minimum engine speed for misfire detection.					
C_N_MAX_MIS_AT	1	0...1FE0H	0...8160	1	[rpm]
Maximum engine speed for misfire detection / automatic transmission.					
C_N_MAX_MIS_MT	1	0...1FE0H	0...8160	1	[rpm]
Maximum engine speed for misfire detection / manual transmission.					
C_TCO_MIN_MIS	1	0...FEH	-48...142.5	0.75	[°C]
Minimum coolant temperature for misfire detection					
C_TCO_WUP_MIN_MIS	1	0...FEH	-48...142.5	0.75	[°C]
Minimum warm up coolant temperature for misfire detection					
C_T_DLY_ST_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration when engine operating state start LV_ST has been exited.					
C_T_DLY_IV_OFF_MIS	1	0...FFFFH	0...655.35	0.01	[s]
Fade out duration after injection is back on.					
C_NR_TDC_INH_MIS	1	0...FFH	0...255	1	[-]
Fade out duration in TDC when engine speed rise up to C_N_MIN_ZDLY_MIS					
C_SUM_INH_IV_MAX_MIS	1	0...8H	0...8	1	[-]
Maximum number of cylinders in shut-off to fade-out					
C_NR_TDC_IV_OFF_DET_MIS	1	0...7H	0...7	1	[-]
TDC delay to synchronise cylinder shut-off information with the misfire detection index					

51.7 Misfire rate determination and error management

General information:

The misfire rate determination and misfire specific error management are applied to the following CARB misfire legal definitions:

- **CARB A misfire failure criterion** *(LV_MIS_STATE_A)*:

Risk of catalyst damage, monitoring interval over 200 crankshaft revolutions (= 200 * NC_CYL_NR / 2 TDCs) during the driving cycle.

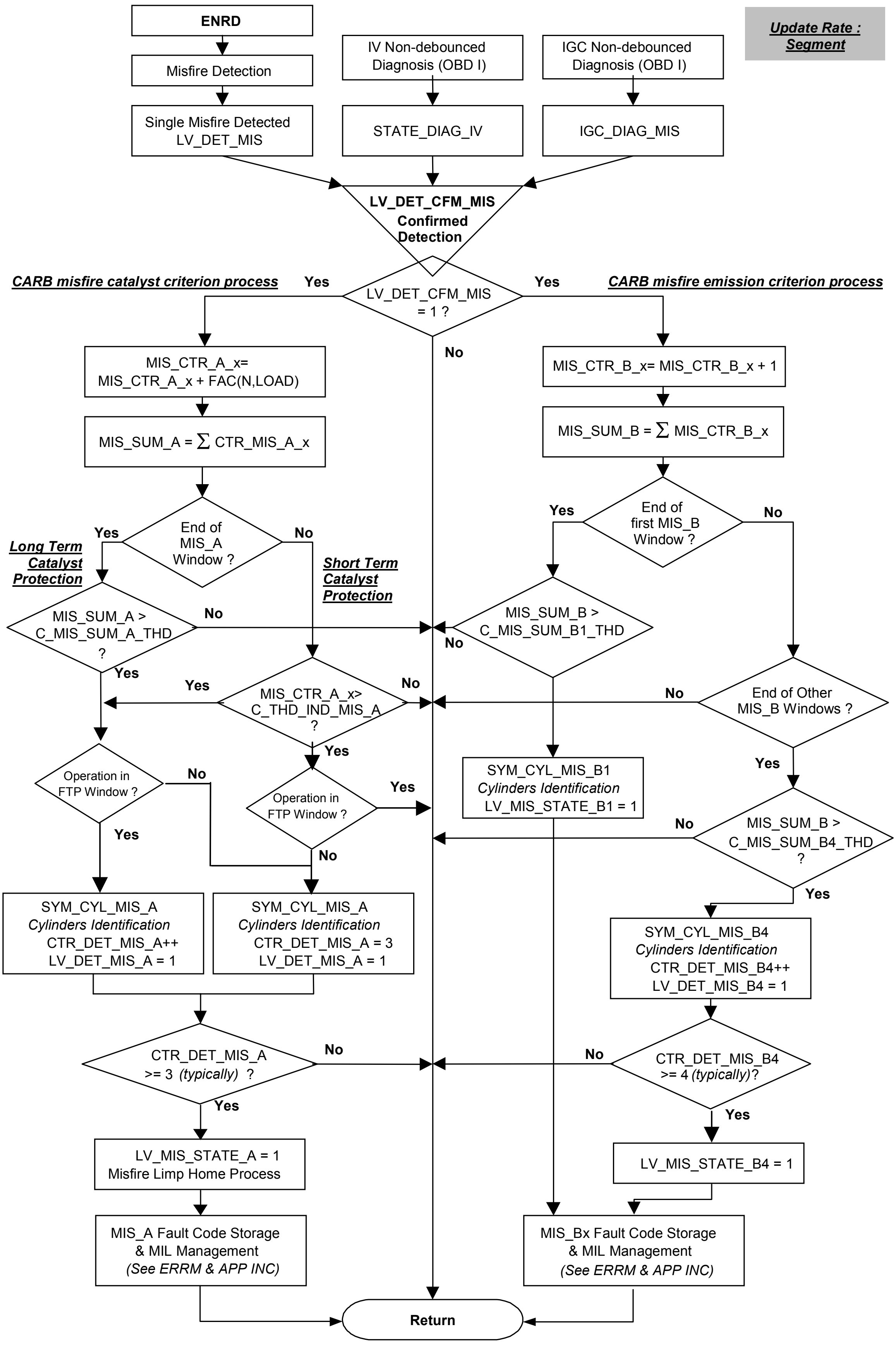
- **CARB B1 misfire failure criterion** *(LV_MIS_STATE_B1):*

Emission increase, monitoring interval over the first 1000 crankshaft revolutions (= 1000 * NC_CYL_NR / 2 TDCs) of the driving cycle.

- **CARB B4 misfire failure criterion** *(LV_MIS_STATE_B4):*

Emission increase, monitoring interval over 1000 crankshaft revolutions (= 1000 * NC_CYL_NR / 2 TDCs). For criterion confirmation, misfire must be take place for 4 monitoring intervals (consecutive or not).

MIS_A, MIS_B1 & MIS_B4 Handling Flowchart diagram:



51.7.1 Combination of different diagnosis reliability levels (misfire detection, ignition, injection)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SUM_FAC_DIAG_MIS	V	0...2FDH	0...5.97656	0.0078125	[-]
Sum of the weighting factor from diagnosis combinations for final misfire decision.					
FAC_IGC_DIAG_MIS	V	0...FFH	0...1.99218	0.0078125	[-]
Reliability weighting factor for ignition diagnosis (ignition failure detected)					
FAC_IV_DIAG_MIS	V	0...FFH	0...1.99218	0.0078125	[-]
Reliability weighting factor for injection valve (injection valve failure detected)					
LV_DET_CFM_MIS	V/O	0...1H	0...1	1	[-]
Flag for individual confirmed misfire detection after crossed diagnosis					

Input data:

LC_MIS_INH	STATE_DIAG_IV	IGC_DIAG_MIS	LOAD_MIS
N_32	LV_INH_IGC_DIAG_MIS	SEG_NR_ER	FAC_ER_DIAG_MIS
LV_INH_IV_DIAG_MIS	LV_DET_MIS		

FUNCTION DESCRIPTION:

General information:

Due to their physical principle, the individual diagnostic functions have a different diagnostic relevance depending on the operating point.

To achieve the best possible detection of misfires, the individual diagnostic methods are combined. Here with an assumed misfire of a part function, an operating point dependent weighting factor, which reflects the reliability of this part function in the current operating point, is taken from a map. If the sum of these three values now exceeds a threshold for one combustion, then misfire is detected.

The value which is not debounced is used as input value of the IV and IGCFB diagnoses. The injection for a cylinder is cut off exclusively by the CARB error mechanism and not by the IV and IGCFB diagnostic functions.

Application conditions:

Misfire segment after detection, before CARB diagnosis

Application conditions:

Update rate: every segment

Activation:

LC_MIS_INH = 0

Deactivation:

LC_MIS_INH = 1

Initialisation on ECU reset, on LV_DC 0 to 1 transition and on LC_MIS_INH 0 to 1 transition:

SUM_FAC_DIAG_MIS = 0

FAC_IGC_DIAG_MIS = 0

FAC_IV_DIAG_MIS = 0

LV_DET_CFM_MIS = 0

Formula section:

The output from the misfire detection, injection and ignition diagnosis functions is influenced by weighting interpolated table, giving the reliability of the individual functions.

- A weighting value of **0** means that detection is not possible for this engine operating point.

- A value equal to or greater than **1** indicates reliable detection.

- The values in between represent the corresponding levels of reliability.

- Irregular engine operation weighting (LV_DET_MIS = 1):

If misfiring is suspected by the irregular engine operation test, a weighting factor for the corresponding engine operating point is applied. This factor is derived from tables according combustion mode and determine in application incidence. The corresponding result is in FAC_ER_DIAG_MIS data.

- Ignition diagnosis weighting (IGC_DIAG_MIS):

IGC_DIAG_MIS is the not debounced carrier from Ignition diagnosis, which has to be delayed due to misfire detection *(according segment reference SEG_NR_ER-1, cylinder x compression phase)*. If misfiring is suspected by the ignition diagnosis function, a weighting factor for the corresponding engine operating point is applied.

This factor is derived from IP_FAC_IGC_DIAG_MIS table.

For the time the number of cylinders shut off due to misfire detection limp home reaches or exceeds the applicable number of C_MIS_NR_OFF_IV_IGCFB, FAC_IGC_DIAG factor is forced to ''1'' *(test based on INH_IV_MIS carrier)*.

```
If(1) IGC_DIAG_MIS[SEG_NR_ER] = 1
    And LV_INH_IGC_DIAG_MIS = 0
Then(1)
  If(2)     number of bits set to "1" in INH_IV_MIS carrier structure >=
            C_MIS_NR_OFF_IV_IGCFB
  Then(2)   FAC_IGC_DIAG_MIS = 1
  Else(2)   FAC_IGC_DIAG_MIS = IP_FAC_IGC_DIAG_MIS(N_32, LOAD_MIS)
  EndIf(2)
Else(1)     FAC_IGC_DIAG_MIS = 0
EndIf(1)
```

- Injection diagnosis weighting (STATE_DIAG_IV) :

STATE_DIAG_IV is a not debounced carrier from Injection valves diagnosis.

Injection valve in default is identifed according segment reference SEG_NR_ER within the bitfield structure STATE_DIAG_IV.

If misfiring is suspected by the injection valve diagnosis function, a weighting factor for the corresponding engine combustion mode is applied.

```
If    STATE_DIAG_IV[SEG_NR_ER] = 1
```

```
    And   LV_INH_IV_DIAG_MIS = 0
Then      FAC_IV_DIAG_MIS = IP_FAC_IV_DIAG_MIS(N_32, LOAD_MIS)
Else      FAC_IV_DIAG_MIS = 0
EndIf
```

- Formation of a cylinder - specific diagnosis weighting sum :

The sum of the three weighting values SUM_FAC_DIAG_MIS is performed for the individual cylinder. as soon as one of the three diagnosis function suspects misfiring.

```
SUM_FAC_DIAG_MIS = FAC_ER_DIAG_MIS
                   + FAC_IGC_DIAG_MIS
                   + FAC_IV_DIAG_FAC
If    SUM_FAC_DIAG_MIS >= 1
Then      LV_DET_CFM_MIS = 1      misfire status is confirmed
Else      LV_DET_CFM_MIS = 0      no misfire
EndIf
```

Calibration data:

Name	Dim	Hex. Limits	Phys. Limits	Resol.	Unit
C_MIS_NR_OFF_IV_IGCFB	1	0...8H	0...8	1	[-]
Threshold for number of cylinders shut off due to misfire					
IP_FAC_IGC_DIAG_MIS	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_N_32_IP_FAC_IGC_DIAG_MIS	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_IP_FAC_IGC_DIAG_MIS	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for ignition diagnosis (ignition failure detected).					
IP_FAC_IV_DIAG_MIS	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_N_32_IP_FAC_IGC_DIAG_MIS	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_MIS_IP_FAC_IV_DIAG_MIS	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for injection valve diagnosis (injection valve failure detected).					

51.7.2 Misfire detection counters (ISO 15031 Data)

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
CTR_MIS_TOT_NVMY	V/S	0...FFFFFFFFH	0...4294967295	1	[-]
Overall misfire detection counter					
CTR_MIS_TOT_DC	V/S	0...FFFFH	0...65535	1	[-]
Former / current driving cycle overall misfire detection counter					
CTR_MIS_DC_CYL[NC_CYL_NR]	V/S	0...FFFFH	0...65535	1	[-]
Former / current driving cycle misfire detection cylinder counters (ISO15031 Data $0C)					
CTR_MIS_DC_MMV_CYL[NC_CYL_NR]	V/S	0...FFFFH	0...65535	1	[-]
Exponential weighted moving average (EWMA) misfire counters (ISO15031 Data $0B)					

Input data:

LV_DC	LV_ENA_ER	LC_MIS_INH	SEG_NR_ER

FUNCTION DESCRIPTION:

General information:

These counters are incremented at each misfire detection based on misfire index evaluation *(after crossed diagnosis with IGC & IV OBDI diags evaluation)*. Thus, it's possible to determine the total number of misfire detected during engine lifetime and during current driving cycle.

Moreover, cylinder misfire counters have been introduced for Mode 6 communication service, as described in last ISO specification available (ISO/DIS 15031-5.8).

Note: CTR_MIS_TOT_DC & CTR_MIS_DC_CYL[NC_CYL_NR] are initialised only at engine running to keep the information after engine stop.

51.7.2.1 NVMY Data Formating

On non volatile memory formatting or on NV memory corruption detection:

```
CTR_MIS_TOT_NVMY = 0
CTR_MIS_TOT_DC = 0
For(1) x = 0 : NC_CYL_NR-1
    CTR_MIS_DC_CYL[x] = 0
    CTR_MIS_DC_MMV_CYL[x] = 0
EndFor(1)
```

51.7.2.2 Initialisation of former / current driving cycle counters

Initialisations:

- at ECU reset

  NVMY data loading
- at LV_DC 0 -> 1 transition event

  CTR_MIS_TOT_DC = 0

```
For(1) x = 0 : NC_CYL_NR-1
    CTR_MIS_DC_CYL[x] = 0
EndFor(1)
```

51.7.2.3 Increment of misfire counters during current driving cycle

Application conditions:

Activation: *LV_ENA_ER = 1*

And *LC_MIS_INH = 0*

Deactivation: *LV_ENA_ER = 0*

Or *LC_MIS_INH = 1*

Update rate: Segment task

Formula section:

```
If    LV_DET_CFM_MIS = 1
Then      CTR_MIS_TOT_DC = CTR_MIS_TOT_DC + 1                  // with saturation
          CTR_MIS_TOT_NVMY = CTR_MIS_TOT_NVMY + 1              // with saturation
          CTR_MIS_DC_CYL[SEG_NR_ER] = CTR_MIS_DC_CYL[SEG_NR_ER] + 1
                                                                // with saturation
EndIf
```

51.7.2.4 Update of the EWMA misfire counters

Exponential weighted moving average (EWMA) misfire counters are updated at the beginning of the driving cycle based on former driving cycle values.

Update rate: at LV_DC 0 -> 1 transition event *before CTR_MIS_DC_CYL[x] reset*

```
For x = 0 : NC_CYL_NR-1
    CTR_MIS_DC_MMV_CYL[x] =
        0.9 * CTR_MIS_DC_MMV_CYL[x] + 0.1 * CTR_MIS_DC_CYL[x]
EndFor
```

51.7.3 Exhaust cylinder bank reference for misfire detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SEG_NR_CBK_ER	V	0...FFH	0...255	1	[-]
Exhaust cylinder bank reference for misfire monitoring					

Input data:

SEG_NR_ER	LV_SYN_ENG	NC_CBK_EX_NR	NC_CYL_NR

FUNCTION DESCRIPTION:

For catalyst damage and/or emission criterion, misfire diagnosis functions can be set via calibration data in a way to be able to detect criterions according exhaust cylinder bank phasing instead of a global criterion detection.

Exhaust cylinder bank reference for misfire detection allows to increase these criterions on the dedicated cylinder bank weighting factor if the misfire detected has effect on the impacted cylinder bank and not on the other (if multiple exhaust cylinder bank design).

Description:

The exhaust cylinder bank reference SEG_NR_CBK_ER is determinate according the segment reference SEG_NR_ER. It already includes the delay introduced by the misfire index chosen.

Application conditions:

Update rate: Segment task

Activation: *LV_SYN_ENG = 1*

Deactivation: *LV_SYN_ENG = 0*

Initialisation on Deactivation condition:

SEG_NR_CBK_ER = 0

Initialisation: at ECU reset

SEG_NR_CBK_ER = 0

Formula section:

SEG_NR_CBK_ER = NC_CBK_EX_NR_MISF[SEG_NR_ER]

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CBK_EX_NR_MISF[NC_CYL_NR]	-	0...FFH	0...255	1	[-]
Exhaust cylinder bank allocation according misfire segment phasing					

51.7.4 Determination of CARB A misfire criterion, causing catalyst damage

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DET_MIS_A	V	0...1H	0...1	1	[-]
CARB A misfire criterion detected in actual 200 rev. window (not debounced)					
LV_MIS_STATE_A	V/O	0...1H	0...1	1	[-]
CARB A misfire criterion confirmed (debounced)					
LV_INH_IV_MIS_A	V/O	0...1H	0...1	1	[-]
At least one cylinder is shut-off by a CARB A misfire criterion confirmed					
CTR_MIS_A_LIH_CYL[NC_CYL_NR]	V	0...FFH	0...255	1	[-]
Cylinder dedicated counters for CARB A Limp-home phases on current DC					
MIS_A_FAC	V	0...64H	0...100	1	[-]
Catalyst damage weighting factor to increment cylinder specific counters when misfire is detected					
MIS_A_FAC_BUF[NC_SIZE_THD_ER_BUF]	V	0...64H	0...100	1	[-]
Catalyst damage weighting factor buffer					
INH_IV_MIS	V/O	0...FFH	0...255	1	[-]
Identification of cylinders shut off by misfire CARB A					
MIS_CTR_A[NC_CYL_NR]	V/O	0...FFFFH	0...65535	1	[-]
Misfire sums (cylinder individual) after combination and weighting in CARB A window.					
MIS_NR_TDC_A	V	0...FFFFH	0...65535	1	[-]
TDC counter CARB A window (200 crankshaft revolutions).					
MIS_SUM_A	V/O	0...FFFFH	0...65535	1	[-]
CARB A misfire weighted sum (global)					
MIS_SUM_A_CBK[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
CARB A misfire weighted sum (dedicated to Exhaut cylinder bank)					
CTR_CHK_MIS_A	V	0...FFH	0...255	1	[-]
CARB A windows occurrence counter					
CTR_FTP_CDN_MIS_A	V	0...FFFFH	0...65535	1	[-]
TDC counter for FTP condition detection					
CTR_DET_MIS_A	V	0...FFH	0...255	1	[-]
CARB A misfire criterion detected counter					
LV_FTP_MIS_A	V	0...1H	0...1	1	[-]
Boolean for FTP emission cycle condition status at the end of CARB A window					
LV_END_MIS_A	V/O	0...1H	0...1	1	[-]
CARB A misfire criterion determination end					
LV_END_WIN_MIS_A	V/O	0...1H	0...1	1	[-]
End of CARB A window for similar condition					
LV_ERR_IN_WIN_MIS_A	V/O	0...1H	0...1	1	[-]
End during CARB A window for similar condition					
SYM_CYL_MIS_A	V/O	0...FFFFH	0...65535	1	[-]
CARB A misfire criterion symptoms					

Input data:

NC_CBK_EX_NR	LOAD_MIS	FAC_MIS_A_THD_IND_APP	LV_CDN_MIS_A
LV_DC	NC_CYL_NR	LV_REQ_APP_CTR_MIS_A	INH_INJ
SEG_NR_ER	LV_AT	LV_ES	LV_PUC
LC_MIS_INH	FAC_MIS_A_APP	N	FAC_MIS_SUM_A_THD_APP
FAC_MIS_A_THD_IND_APP			

Description:

With a CARB A misfire rate criterion, the emissions are increased and there is a risk of catalyst damage if the cylinder causing the failure criterion can not be shut off.

Detection:

The check for misfire rate violation **CARB A** comprises 200 crankshaft revolutions *(100 * NC_CYL_NR tdc's)* for statistic misfiring and cyclewise monitoring for best possible catalyst protection.

If a confirmed misfiring is detected on ER current TDC (LV_DET_CFM_MIS=1), a weighting factor (MIS_A_FAC) is added to the cylinder-specific counter MIS_CTR_A[SEG_NR_ER], extracted from an interpolated table *(see APP. INC.)* who depends on engine combustion mode and current operating point. The increment represents the severity of catalyst damage.

Statistic:

The monitoring for Misfire **CARB A** is performed continuously in intervals of 200 crankshaft revolutions.

Monitoring during CARB A window:

To guarantee best possible catalyst protection, the cylinder-specific counters are monitored each cycle. For that purpose, MIS_CTR_A[SEG_NR_ER] is compared to the applicable value C_MIS_A_IND_THD representing the critical threshold for single cylinder misfiring.

If the threshold is reached outside of the FTP cycle, the concerning cylinder is shut off immediately. CARB A misfire criterion is confirmed, else the criterion will be evaluated at the end of the 200 crankshaft revolution window.

Monitoring at the end of 200 crankshaft revolutions:

At the end of each monitoring interval the cylinder-specific sums MIS_CTR_A[NC_CYL_NR] and the global sum (MIS_SUM_A) are available.

If the sum MIS_SUM_A is greater than the threshold C_MIS_SUM_A_THD, the cylinder with the highest misfire rate is determined (by comparing the different MIS_CTR_A[NC_CYL_NR] values).

Then operation inside/outside of the FTP-area is checked by comparison of the counter CTR_FTP_CDN_MIS_A / MIS_NR_TDC_A with the threshold C_RATIO_FTP_CDN_MIS_A. If the FTP tdc's ratio is lower than the threshold, not enough operating points have been inside the FTP-window and therefore operation outside FTP-area is detected (LV_FTP_MIS_A=0). This is done in order to fulfil the CARB-regulations, which require MIL-illumination at first threshold exceeding outside the FTP-area and at third exceeding inside the FTP-area.

Therefore the counter CTR_DET_MIS_A is incremented inside the FTP-area at each threshold exceeding. That means it reaches the maximum of C_NR_DET_MAX_MIS_A = 3 only if the threshold is exceeded in 3 succeeding monitoring intervals.

Outside the FTP-area the failure CARB A is validated immediately. The counter is reset only after an ECU-reset.

Even if we are inside the FTP-area, a specific application request (according catalyst temperature or else) LV_REQ_APP_CTR_MIS_A can force to validate immediately the failure CARB A *(see legal requirements & customer wishes).* This request is defined in associated Application Incidences file.

Note: cylinder and Exhaut cylinder bank specific counters are respectively indexed with SEG_NR_ER and SEG_NR_CBK_ER phase reference counters to take in account ERND delay in misfire monitoring process

Application conditions:

Initialisation: *at ECU reset, at LV_DC 0 - > 1 transition, at LC_MIS_INH 0 to 1 transition & at MIS_A or MIS_[NC_CYL_NR] errors clearing*

```
SYM_CYL_MIS_A          = 0
INH_IV_MIS             = 0
MIS_NR_TDC_A           = 0
MIS_SUM_A              = 0
CTR_FTP_CDN_MIS_A      = 0
CTR_DET_MIS_A          = 0
CTR_CHK_MIS_A          = 0
LV_END_WIN_MIS_A       = 0
LV_END_MIS_A           = 0
LV_FTP_MIS_A           = 0
LV_MIS_STATE_A         = 0
LV_INH_IV_MIS_A        = 0
LV_DET_MIS_A           = 0
LV_ERR_IN_WIN_MIS_A    = 0
For x = 0 : NC_CYL_NR-1
    MIS_CTR_A[x] = 0                // All cylinder specific counters
    CTR_MIS_A_LIH_CYL[x] = 0        // All cylinder specific counters
EndFor
For i = 0 : NC_CBK_EX_NR-1
    MIS_SUM_A_CBK[i] = 0       // All Exhaut cylinder bank specific counters
EndFor
```

Update rate: segment task

Activation:

```
LV_DC = 1
And LC_MIS_INH = 0
```

Deactivation:

```
LV_DC = 0
Or LC_MIS_INH = 1
```

51.7.4.1 Determination of catalyst damage weighting factor

Catalyst damage weighting factor is determinated according engine combustion mode & buffered to take in account admission phase (SEG_NR_ER - 2).

MIS_A_FAC_BUF is managed as a rotary buffer, MIS_A_FAC_BUF[0] is the weighting factor for the actual segment value.

At each ER segment recurrence:

FAC_MIS_A_APP depends on the combustion mode and determine in application incidence (must be calculated first)

MIS_A_FAC_BUF is managed as a rotary buffer

```
MIS_A_FAC_BUF[current_position] = FAC_MIS_A_APP
MIS_A_FAC = MIS_A_FAC_BUF          [oldest_position]       // points on the oldest value of the buffer
```

51.7.4.2 CARB A window management (main process)

```
If(1)   LV_CDN_MIS_A = 1                  // MIS_A monitoring active (see App Inc file)
Then(1)
    If(2)   LV_DET_CFM_MIS = 1            // Individual misfire detected
    Then(2)     Call 'CARB A Misfire counters increment process' (see description below)
    Else(2)     MIS_A Misfire counters unchanged
    EndIf(2)
    If(2)       MIS_NR_TDC_A < 100 * NC_CYL_NR
    Then(2)     MIS_NR_TDC_A = MIS_NR_TDC_A + 1
                LV_END_WIN_MIS_A = 0              // End of window not reached
                If(3)       LOAD_MIS < C_LOAD_MIS_FTP
                And         N < C_N_MIS_FTP       // FTP window conditions
                Then(3)     CTR_FTP_CDN_MIS_A = CTR_FTP_CDN_MIS_A + 1
                EndIf(3)
                call 'Monitoring process during CARB A window' (see description below)
    Else(2)     call 'Monitoring process at CARB A window end' (see description below)
                call 'CARB A criterion end detection process' (see description below)
                call 'CARB A window reset process' (see description below)
```

```
        EndIf(2)
    Else(1)   no process              // Counters & flags unchanged
    EndIf(1)
```

51.7.4.3 CARB A Misfire counters increment process

```
    MIS_SUM_A = MIS_SUM_A + MIS_A_FAC
    MIS_SUM_A_CBK[SEG_NR_CBK_ER] =
                        MIS_SUM_A_CBK[SEG_NR_CBK_ER] + MIS_A_FAC
                        // Assign only ER current specific Exhaut cylinder bank counter
    MIS_CTR_A[SEG_NR_ER] = MIS_CTR_A[SEG_NR_ER] + MIS_A_FAC
                        // Assign only ER current specific cylinder counter
```

51.7.4.4 Monitoring process during CARB A window *(short term catalyst protection & limp home)*

```
    If(1) { [ LV_MIS_A_DIAG_REQ_APP = 0
        And ( ( LV_AT = 0 And MIS_CTR_A[SEG_NR_ER] ≥ C_MIS_A_IND_THD_MT )
                  Or
                ( LV_AT = 1 And MIS_CTR_A[SEG_NR_ER] ≥ C_MIS_A_IND_THD_AT ) )
              ]
                        // Test based only on ER current specific cylinder counter
        Or [ LV_MIS_A_DIAG_REQ_APP = 1
              And ( ( LV_AT = 0 And MIS_CTR_A[SEG_NR_ER] ≥
                          FAC_MIS_A_THD_IND_APP * C_MIS_A_IND_THD_MT )
                  Or ( LV_AT = 1 And MIS_CTR_A[SEG_NR_ER] ≥
                          FAC_MIS_A_THD_IND_APP * C_MIS_A_IND_THD_AT ) ) ] }
    Then(1)
      Short term FTP conditions check
          If(2) [ LV_AT = 1 And
            CTR_FTP_CDN_MIS_A/MIS_NR_TDC_A < C_RATIO_FTP_CDN_IND_MIS_AT ]
          Or [ LV_AT = 0 And
            CTR_FTP_CDN_MIS_A/MIS_NR_TDC_A < C_RATIO_FTP_CDN_IND_MIS_MT ]
              // Out of FTP cycle inside CARB A window
          Then(2)  Set concerned cylinder bit in structure SYM_CYL_MIS_A to 1
              LV_DET_MIS_A = 1          // Misfire CARB A criterion detected
              LV_MIS_STATE_A = 1        // Misfire CARB A criterion confirmed
              LV_ERR_IN_WIN_MIS_A = 1
              CTR_DET_MIS_A = max(CTR_DET_MIS_A+1, C_NR_DET_MAX_MIS_A)
```

```
                LV_END_MIS_A = 1
                // Substract Current cylinder MIS_A & MIS_SUM_A_CBK counters on global & cylinder bank
                MIS_A sums to avoid wrong global MIS_A detection at end of current Long term diagnosis
                MIS_SUM_A = MIS_SUM_A - MIS_CTR_A[SEG_NR_ER]
                MIS_SUM_A_CBK[SEG_NR_CBK_ER] =
                    MIS_SUM_A_CBK[SEG_NR_CBK_ER] - MIS_CTR_A[SEG_NR_ER]
                MIS_CTR_A[SEG_NR_ER] = 0
                                    // Avoid to retrigger criterion until end of the current window
                If(3)    LC_ENA_MIS_A_IND_LIH = 1
                Then(3)
                    Shut off cylinder indexed by SEG_NR_ER via INH_IV_MIS carrier
                    Evaluate 'Misfire Cylinder Limp Home process'
                                                            (see description afterwards)
                EndIf(3)
            Else(2)        no operation   // MIS_A criterion evaluated at the end of MIS_A window or
                                          // at the next cylinder evaluation
            EndIf(2)
Else(1)         flags & data unchanged
EndIf(1)
```

```
#If(1)      NLC_ENA_SCDN_NEW = 1
```

In this case, we do exactly the same operations that those done in "**'Monitoring process at CARB A window end'.**

```
#EndIf(1)
```

51.7.4.5 Monitoring process at CARB A window end *(long term catalyst protection & limp home)*

<u>*Global CARB A FTP conditions check*</u>

```
If      [ LV_AT = 1   And
            CTR_FTP_CDN_MIS_A / MIS_NR_TDC_A ≥ C_RATIO_FTP_CDN_MIS_AT ]
        Or    [ LV_AT = 0   And
            CTR_FTP_CDN_MIS_A / MIS_NR_TDC_A ≥ C_RATIO_FTP_CDN_MIS_MT ]
Then        LV_FTP_MIS_A =  1
```

```
Else        LV_FTP_MIS_A =  0
EndIf
```

Criterion detection process

```
// Be carefull, the counter of window A (CTR_CHK_MIS_A ) is incremented only at the end of the window and not during.
CTR_CHK_MIS_A = CTR_CHK_MIS_A + 1
If(1)  {  LV_MIS_A_DIAG_REQ_APP = 0
          And  [   ( LV_AT =  0 And
                       ( MIS_SUM_A  ≥  C_MIS_SUM_A_THD_MT
                           Or     MIS_SUM_A_CBK[0...NC_CBK_EX_NR-1]
                                  ≥ C_MIS_SUM_A_CBK_THD_MT )  // Check for each exhaust CBK )
                   Or ( LV_AT =  1 And
                       ( MIS_SUM_A  ≥  C_MIS_SUM_A_THD_AT
                           Or     MIS_SUM_A_CBK[0...NC_CBK_EX_NR-1]
                                  ≥ C_MIS_SUM_A_CBK_THD_AT )  // Check for each exhaust CBK ) ] }
       Or {  LV_MIS_A_DIAG_REQ_APP = 1
               And  [  ( LV_AT = 0 And
                   ( MIS_SUM_A  ≥ FAC_MIS_SUM_A_THD_APP * C_MIS_SUM_A_THD_MT
                           Or     MIS_SUM_A_CBK[0...NC_CBK_EX_NR-1]
                                  ≥ FAC_MIS_SUM_A_THD_APP *
C_MIS_SUM_A_CBK_THD_MT )  // Check for each exhaust CBK )
                   Or ( LV_AT = 1 And
                   ( MIS_SUM_A  ≥ FAC_MIS_SUM_A_THD_APP * C_MIS_SUM_A_THD_AT)
                           Or     MIS_SUM_A_CBK[0...NC_CBK_EX_NR-1]
                                      ≥ FAC_MIS_SUM_A_THD_APP *
       C_MIS_SUM_A_CBK_THD_AT )  // Check for each exhaust CBK ) ] }
Then(1)
       If(2) LV_ERR_IN_WIN_MIS_A = 0
       Then(2)
          LV_DET_MIS_A = 1
          LV_ERR_IN_WIN_MIS_A = 1
```

As soon as the misfire criterion detected counter reached his threshold, we directly set the error. To avoid to stored it in the same window a second time, we used *lv_err_in_win_mis_a flag.*

```
          If(3)   LV_FTP_MIS_A = 0                            // Outside of FTP cycle
                  Or     LV_REQ_APP_CTR_MIS_A = 1   // Application specific request
          Then(3)  CTR_DET_MIS_A=max(CTR_DET_MIS_A+1, C_NR_DET_MAX_MIS_A)
```

```
            Else(3)  CTR_DET_MIS_A = CTR_DET_MIS_A + 1
            EndIf(3)
            If(3)  CTR_DET_MIS_A  ≥ C_NR_DET_MAX_MIS_A
            Then(3)      // Global MIS_A default
              LV_MIS_STATE_A = 1
              // MIS_A cylinder identification
                For(4) x = 0 to NC_CYL_NR - 1 // Check for each cylinder x
                  If(5) MIS_CTR_A[x] > C_MIS_A_MIN_NR * MIS_SUM_A
                  Then(5)  Set in SYM_CYL_MIS_A the cylinder x concerned bit to 1
                    If(6) LC_ENA_MIS_A_LIH = 1 And
                          [ LV_AT = 0 And
                          CTR_DET_MIS_A  ≥ ID_NR_DET_ENA_MIS_A_LIH_MT(N_32) ]
                          Or  [ LV_AT = 1 And
                          CTR_DET_MIS_A  ≥ ID_NR_DET_ENA_MIS_A_LIH_AT(N_32) ]
                    Then(6)     Shut off cylinder x via INH_IV_MIS carrier
                                Evaluate 'Misfire Cylinder Limp Home process'
                                                   (see description afterwards)
                    Else(6)     No operation
                    EndIf(6)
                  EndIf(5)
                EndFor(4)
            // MIS_A random cylinder pattern
                If(4)    No specific cylinder counters comply with C_MIS_A_MIN_NR *
                           MIS_SUM_A criterion
                Then(4)
                  If(5)  LC_CONF_RND_DET_MIS_A = 0
                  Then(5)  Evaluate Cylinder with the highest  MIS_CTR_A[x]  value
                             Set concerned cylinder bit to 1 in SYM_CYL_MIS_A carrier
                             Set RDN bit (random) to 0 in SYM_CYL_MIS_A carrier
                  Else(5)    Set RDN bit (random) to 1 in SYM_CYL_MIS_A carrier
                  Endif(5)
                Else(4)  Set RDN bit (random) to 0 in SYM_CYL_MIS_A carrier
                EndIf(4)
            EndIf(3)
          EndIf(2)
      Else(1)
```

```
    // Be carefull, the 3 following datas are reseted only at the end of the window and not
    during.
              SYM_CYL_MIS_A = 0
              LV_DET_MIS_A = 0
              LV_MIS_STATE_A = 0
EndIf(1)
// MIS_A default management in case of cylinder shut-off
If      INH_IV_MIS ≠ 0  And LC_CONF_MIS_A_LIH = 1
Then      LV_DET_MIS_A = 1
          LV_MIS_STATE_A = 1
          Set in SYM_CYL_MIS_A the cylinder bit(s) who are set in INH_IV_MIS (OR mask)
Else      no action
EndIf
// MIS_A multiple cylinder identification
If      LC_CONF_MPL_DET_MIS_A = 1
        And  Number of cylinder(s) set to 1 in SYM_CYL_MIS_A is equal or over
               C_NR_CYL_MPL_MIS_A
Then      Set MPL bit (multiple) to 1 in SYM_CYL_MIS_A carrier
Else      Set MPL bit (multiple) to 0 in SYM_CYL_MIS_A carrier
End
```

51.7.4.6 CARB A end detection process

```
If   LV_FTP_MIS_A = 0
     Or      CTR_DET_MIS_A  ≥ C_NR_DET_MAX_MIS_A
     Or      CTR_CHK_MIS_A  ≥ C_NR_DET_MAX_MIS_A
Then         LV_END_MIS_A = 1
EndIf
LV_END_WIN_MIS_A = 1                  // Set for one segment
```

51.7.4.7 CARB A window reset process

```
MIS_NR_TDC_A = 0
MIS_SUM_A = 0
CTR_FTP_CDN_MIS_A = 0
For x = 0 : NC_CYL_NR-1
     MIS_CTR_A[x] = 0              // All cylinder specific counters
EndFor
For i = 0 : NC_CBK_EX_NR-1
```

MIS_SUM_A_CBK[i] = 0 *// All exhaust cylinder bank specific counters*

EndFor

<u>Reset of Misfire Status</u>:

The Misfire Status CARB A is resetted at the end of the driving cycle → see formula section

51.7.4.8 Misfire Cylinder Identification SYM_CYL_MIS_A

The location of detected cylinders with misfire status CARB A is coded in a carrier word:

```
           SYM_CYL_MIS_A
┌─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ └──────> Cylinder 0
 │ │ │ │ │ │ │ │ │ │ │ │ │ │ └────────> Cylinder 1
 │ │ │ │ │ │ │ │ │ │ │ │ │ └──────────> Cylinder 2
 │ │ │ │ │ │ │ │ │ │ │ │ └────────────> Cylinder 3
 │ │ │ │ │ │ │ │ │ │ │ └──────────────> Cylinder 4
 │ │ │ │ │ │ │ │ │ │ └────────────────> Cylinder 5
 │ │ │ │ │ │ │ │ │ └──────────────────> Cylinder 6
 │ │ │ │ │ │ │ │ └────────────────────> Cylinder 7
 │ │ │ │ │ │ │ └──────────────────────> Cylinder 8
 │ │ │ │ │ │ └────────────────────────> Cylinder 9
 │ │ │ │ │ └──────────────────────────> Cylinder 10
 │ │ │ │ └────────────────────────────> Cylinder 11
 │ │ │ └──────────────────────────────> RDN
 │ │ └────────────────────────────────> MPL
 │ └──────────────────────────────────> -
 └────────────────────────────────────> -
```

51.7.4.9 Misfire Cylinder Limp Home process:

Cylinder shut off is limited by a C_MIS_MAX_NR_OFF_IV maximum number of cylinders that can be shut-off simultaneously

Each time, the misfire cylinder limp home process on one (or more) cylinder(s) is triggered, the corresponding cylinder counter(s) CTR_MIS_A_LIH_CYL[NC_CYL_NR] is(are) unitary incremented.

The location of switched off Cylinder(s) x is set to 1 in INH_IV_MIS, other cylinder bits are set to 0.

<u>On the same driving cycle</u>, if after a cylinder shut-off cancellation, a new cylinder MIS_A criterion occurs during CARB A window, the sytem must shut-off this cylinder once more.

<u>Carrier structure :</u>

```
   INH_IV_MIS
┌─┬─┬─┬─┬─┬─┬─┬─┐
└─┴─┴─┴─┴─┴─┴─┴─┘
 │ │ │ │ │ │ │ └──────> Cylinder 0
 │ │ │ │ │ │ └────────> Cylinder 1
 │ │ │ │ │ └──────────> Cylinder 2
 │ │ │ │ └────────────> Cylinder 3
 │ │ │ └──────────────> Cylinder 4
 │ │ └────────────────> Cylinder 5
 │ │
```

```
 │  └──────────────────> Cylinder 6
 └─────────────────────> Cylinder 7
```

51.7.4.10 Conditions for Cylinder switch back on

Update rate: *every 100ms, independently of MIS_A window process & conditions*

Cylinder(s) x can be switched on according two different methods:

1. Cylinder(s) x can be switched back on at the end of the driving cycle

   LC_CONF_INH_IV_MIS = 0

2. Cylinder(s) x can be switched back on after a defined number of fuel cut-off phases **and** at the end of the driving cycle

   LC_CONF_INH_IV_MIS = 1

   For this purpose, the system performs a test for the applicable number C_MIS_MIN_NR_PUC of trailing throttle fuel cut-off LV_PUC cycles, since the last cylinder(s) shutoff.

   Therefore, the trailing throttle fuel cut-off LV_PUC flag must have been active for the applicable minimum period C_T_MIS_MIN_PUC.

   If the condition concerning the minimum number of trailing throttle fuel cut-off events is met. Cylinder shut-off is cancelled at the end of the last trailing throttle fuel cut-off phase only for cylinders who answer to the conditions CTR_MIS_A_LIH_CYL[x] < C_NR_MAX_MIS_A_LIH_CYL *(check for all cylinders),* all other cylinders stay in fuel cut-off up to end of the driving cycle.

Cylinder shut-off is also cancelled after engine stopped LV_ES.

51.7.4.11 Process on cylinder shut-off cancellation

Update rate: *every TDC, independently of MIS_A window process & conditions*

```
If    Any 1 -> 0 cylinder bit(s) transition within INH_IV_MIS
      And LC_CONF_MIS_A_LIH = 1
Then  Corresponding bit(s) within SYM_CYL_MIS_A is set to 0
      If    LC_CONF_MPL_DET_MIS_A = 1
      And   Number of remaining cylinder bit(s) set to 1 in SYM_CYL_MIS_A is less than
            C_NR_CYL_MPL_MIS_A cylinders
      Then  Set MPL bit (multiple) to 0 in SYM_CYL_MIS_A carrier
      EndIf
      If    No more cylinder bit(s) or RND bit are set to 1 whitin SYM_CYL_MIS_A structure
      Then     LV_DET_MIS_A = 0
               LV_MIS_STATE_A = 0
      EndIf
Else  No action
EndIf
```

```
If    Any cylinder bit(s) within INH_IV_MIS is set to 1
      And     Corresponding bit(s) within INH_INJ is (are) also set to 1
Then          LV_INH_IV_MIS_A = 1
Else          LV_INH_IV_MIS_A = 0
EndIf
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LOAD_MIS_FTP	1	0...7FFFH	0...99.99694	3.0518e-3	[%]
FTP cycle engine load threshold					
C_N_MIS_FTP	1	0...1FE0H	0...8160	1	[rpm]
FTP cycle engine speed threshold					
C_MIS_A_IND_THD_MT	1	0...FFFFH	0...65535	1	[-]
Cylinder individual threshold used for cylinder shut-off during CARB A criterion window with MT					
C_MIS_A_IND_THD_AT	1	0...FFFFH	0...65535	1	[-]
Cylinder individual threshold used for cylinder shut-off during CARB A criterion window with AT					
C_RATIO_FTP_CDN_MIS_AT	1	0...FFH	0...99.60937	0.390625	[%]
Covering ratio to detect FTP conditions for long term protection with AT vehicle					
C_RATIO_FTP_CDN_MIS_MT	1	0...FFH	0...99.60937	0.390625	[%]
Covering ratio to detect FTP conditions for long term protection with MT vehicle					
C_RATIO_FTP_CDN_IND_MIS_AT	1	0...FFH	0...99.60937	0.390625	[%]
Covering ratio to detect FTP conditions for short term protection with AT vehicle					
C_RATIO_FTP_CDN_IND_MIS_MT	1	0...FFH	0...99.60937	0.390625	[%]
Covering ratio to detect FTP conditions for short term protection with MT vehicle					
C_MIS_A_MIN_NR	1	0...FFH	0...99.60937	0.390625	[%]
Cylinder identification ratio for CARB A criterion detection					
C_MIS_SUM_A_THD_MT	1	0...FFFFH	0...65535	1	[-]
Cylinder global threshold for misfire status CARB A detection with MT					
C_MIS_SUM_A_CBK_THD_MT	1	0...FFFFH	0...65535	1	[-]
Exhaust cylinder bank threshold for misfire status CARB A detection with MT					
C_MIS_SUM_A_THD_AT	1	0...FFFFH	0...65535	1	[-]
Cylinder global threshold for misfire status CARB A detection with AT					
C_MIS_SUM_A_CBK_THD_AT	1	0...FFFFH	0...65535	1	[-]
Exhaust cylinder bank threshold for misfire status CARB A detection with AT					
C_MIS_MAX_NR_OFF_IV	1	0...4H	0...4	1	[-]
Maximum number of allowed cylinder to be shut-off after detection of the misfire status CARB A.					
C_NR_DET_MAX_MIS_A	1	0...FFH	0...255	1	[-]
Maximum value to confirm misfire CARB A status detection (typical: 3) for long term protection					
C_MIS_MIN_NR_PUC	1	1...FFH	1...255	1	[-]
Minimum number of trailing throttle fuel cut-off phases LV_PUC before cancelling cylinder(s) shut-off.					
C_T_MIS_MIN_PUC	1	1...3E8H	0.1...100	0.1	[s]
Minimum duration of trailing throttle fuel cut-off phase to be validated like this.					
C_NR_CYL_MPL_MIS_A	1	0...FFH	0...255	1	[-]
Number of cylinder in MIS_A to set the multiple misfire CARB A status (typical: 2)					
C_NR_MAX_MIS_A_LIH_CYL	1	0...FFH	0...255	1	[-]
Maximum number of MIS_A cylinder limp-home cycle to disable fuel reactivation during current DC					
LC_CONF_INH_IV_MIS	1	0...1H	0...1	1	[-]
Cylinder switch back on configuration mode : switch back on at the end of driving cycle (=0) or after a defined number of fuel cut-off phases (=1)					
LC_CONF_RND_DET_MIS_A	1	0...1H	0...1	1	[-]
Configuration for random cylinder detection when no specific cylinder identified, (=0) Cyl. with the highest value is identified in SYM_CYL_MIS_A or (=1) No identification is realised and RDN bit is set to 1 in SYM_CYL_MIS_A					
LC_CONF_MPL_DET_MIS_A	1	0...1H	0...1	1	[-]
Configuration for multiple cylinder status set in SYM_CYL_MIS_A (multiple misfire pattern Pcode saved) (=1)					
LC_CONF_MIS_A_LIH	1	0...1H	0...1	1	[-]
Configuration to select MIL mode in case of cylinder shut-off (=0, LV_MIS_STATE_A=0, MIL is not set by cylinder shut-off function), (=1, LV_MIS_STATE_A=1, MIL is ON until end of cylinder shut-off, see mode chosen with LC_CONF_INH_IV_MIS)					
LC_ENA_MIS_A_IND_LIH	1	0...1H	0...1	1	[-]
Enable the misfire cylinder limp home process during the CARB A window (short term catalyst protection) (=1, active)					
LC_ENA_MIS_A_LIH	1	0...1H	0...1	1	[-]
Enable the misfire cylinder limp home process at the end of the CARB A window (long term catalyst protection) (=1, active)					
ID_NR_DET_ENA_MIS_A_LIH_MT	8	0...FFH	0...255	1	[-]

LDPM_N_32_1_MISF	8	0...FFH	0...8160	32	[rpm]
Number of CARB-A detections to enable MIS_A cylinder switch-off limp-home for MT					
ID_NR_DET_ENA_MIS_A_LIH_AT	8	0...FFH	0...255	1	[-]
LDPM_N_32_1_MISF	8	0...FFH	0...8160	32	[rpm]
Number of CARB-A detections to enable MIS_A cylinder switch-off limp-home for AT					

51.7.5 Determination of CARB B1 & B4 misfire criterions, causing increased emissions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MIS_STATE_B1	V/O	0...1H	0...1	1	[-]
CARB B1 misfire criterion confirmed					
LV_DET_MIS_B4	V	0...1H	0...1	1	[-]
CARB B4 misfire criterion detected in actual 1000 rev. window, (not debounced)					
LV_MIS_STATE_B4	V/O	0...1H	0...1	1	[-]
CARB B4 misfire criterion confirmed (debounced)					
LV_MIS_STATE_B	V/O	0...1H	0...1	1	[-]
CARB B (B1 or B4) misfire criterion confirmed (debounced)					
MIS_CTR_B[NC_CYL_NR]	V/O	0...FFFFH	0...65535	1	[-]
Misfire sums (cylinder individual) after combination in CARB B window					
MIS_NR_TDC_B	V	0...FFFFH	0...65535	1	[-]
TDC counter CARB B window (1000 crankshaft revolutions)					
MIS_SUM_B	V/O	0...FFFFH	0...65535	1	[-]
CARB B misfire sum (global)					
MIS_SUM_B_CBK[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
CARB B misfire sum (exhaut cylinder bank dedicated)					
CTR_CHK_MIS_B4	V	0...FFH	0...255	1	[-]
CARB B4 window occurrence counter					
CTR_DET_MIS_B4	V	0...FFH	0...255	1	[-]
CARB B4 misfire criterion detected counter					
LV_END_MIS_B1	V/O	0...1H	0...1	1	[-]
CARB B1misfire criterion determination end					
LV_END_MIS_B4	V/O	0...1H	0...1	1	[-]
CARB B4 misfire criterion determination end					
LV_END_WIN_MIS_B1	V/O	0...1H	0...1	1	[-]
End of CARB B1 window for similar condition					
LV_ERR_IN_WIN_MIS_B1	V	0...1H	0...1	1	[-]
Error during CARB B1 window for similar condition					
LV_ERR_IN_WIN_MIS_B4	V	0...1H	0...1	1	[-]
Error during CARB B4 window for similar condition					
LV_ERR_MIS_A_IN_WIN_B	V/O	0...1H	0...1	1	[-]
Error MIS_A during CARB B1/B4 window for similar condition					
LV_END_WIN_MIS_B4	V/O	0...1H	0...1	1	[-]
End of CARB B4 window for similar condition					
SYM_CYL_MIS_B1	V/O	0...FFFFH	0...65535	1	[-]
CARB B1 misfire criterion symptoms					
SYM_CYL_MIS_B4	V/O	0...FFFFH	0...65535	1	[-]
CARB B4 misfire criterion symptoms					

Input data:

NLC_ENA_SCDN_NEW	FAC_MIS_SUM_B4_THD_APP	LV_AT	LV_CDN_MIS_B1
LV_CDN_MIS_B4	SEG_NR_ER	NC_CBK_EX_NR	LC_MIS_INH
FAC_MIS_SUM_B1_THD_APP	NC_CYL_NR	LV_MIS_B_DIAG_REQ_APP	LV_DC

Description:

The purpose is to detect a misfire rate causing an emission increase. The failure criterion entry is performed by direct statistical evaluation.

Detection:

The check for misfire rate violation CARB comprises 1.000 crankshaft revolutions (500*NC_CYL_NR tdc's).

Each combustion during these 1.000 crankshaft revolutions is monitored for misfiring.

If a confirmed misfiring is detected on ER current TDC (LV_DET_CFM_MIS=1), the cylinder-specific misfire counter MIS_CTR_B[SEG_NR_ER] is incremented by one.

In every driving cycle a distinction is made between misfires detected during the first 1000 crankshaft revs after engine start (**CARB B1**) and misfires detected in later observing intervals. In the second case, a misfire rate violation must take place during four monitoring intervals to recognise misfire (**CARB B4**).

Statistic:

At the end of the first 1.000 crankshaft revolutions, the added cylinder-specific sums are evaluated. If the sum exceeds the applicable value C_THD_MIS_B1_AT/MT, **CARB B1** misfire criterion is detected.

At the end of each following 1.000 crankshaft revolutions interval, the added cylinder-specific sums are evaluated. If the sum exceeds the applicable value C_THD_MIS_B1_AT/MT the anti-bounce counter CTR_DET_MIS_B4 will be incremented till its maximum C_NR_DET_MAX_MIS_B4 (typical = 3) is reached. If the sum does not exceed C_THD_MIS_B1_AT/MT, CTR_DET_MIS_B4 remains unchanged.

To filter out those cylinders with low misfire rates, the cylinder-specific counters MIS_CTR_B[NC_CYL_NR] shall be compared to the applicable value C_THD_MIS_B_MIN before storage takes place.

If none of the cylinder-specific counters is higher than C_THD_MIS_B_MIN, a maximum choice is done and the cylinder with the highest misfire rate is entered into failure memory.

Application conditions:

Initialisation: *at ECU reset, at LV_DC 0 -> 1 transition, at LC_MIS_INH 0 to 1 transition & at MIS_B1/B4 or MIS_[NC_CYL_NR] errors clearing*

LV_MIS_STATE_B1 = 0
LV_DET_MIS_B4 = 0
LV_MIS_STATE_B4 = 0
LV_MIS_STATE_B = 0
SYM_CYL_MIS_B1 = 0
SYM_CYL_MIS_B4 = 0
MIS_NR_TDC_B = 0
MIS_SUM_B = 0
CTR_DET_MIS_B4 = 0
CTR_CHK_MIS_B4 = 0
LV_END_MIS_B1 = 0
LV_END_MIS_B4 = 0
LV_END_WIN_MIS_B1 = 0
LV_END_WIN_MIS_B4 = 0

```
LV_ERR_IN_WIN_MIS_B1 = 0
LV_ERR_IN_WIN_MIS_B4 = 0
LV_ERR_MIS_A_IN_WIN_B = 0
For x = 0 : NC_CYL_NR-1
        MIS_CTR_B[x] = 0            // All cylinder specific counters
EndFor
For i = 0 : NC_CBK_EX_NR-1
        MIS_SUM_B_CBK[i] = 0        // All Exhaut cylinder bank specific counters
EndFor
```

Update rate: every segment

Activation:

```
LV_DC = 1
And LC_MIS_INH = 0
```

Deactivation:

```
LV_DC = 0
Or LC_MIS_INH = 1
```

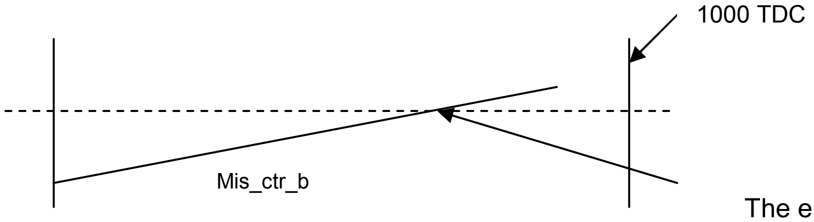
Formula section:

51.7.5.1 CARB B1/B4 windows management *(main process)*

```
If(1) LV_CDN_MIS_B1 = 1                // CARB B1 monitoring active (see App Inc file)
      Or LV_CDN_MIS_B4 = 1             // CARB B4 monitoring active (see App Inc file)
Then(1)
      If(2)   LV_DET_CFM_MIS = 1       // Misfire detected
      Then(2)    call 'CARB B1/B4 Misfire counters increment process'
                 (see description below)
      Else(2)    MIS_B1/B4 counters unchanged
      EndIf(2)
      If(2)   LV_ERR_IN_WIN_MIS_A = 1
      Then(2)    LV_ERR_MIS_A_IN_WIN_B = 1
      EndIf(2)
      If(2)      MIS_NR_TDC_B < 500 * NC_CYL_NR
      Then(2)    MIS_NR_TDC_B= MIS_NR_TDC_B  + 1
                 LV_END_WIN_MIS_B1 = 0
                 LV_END_WIN_MIS_B4 = 0
                 If(3) NLC_ENA_SCDN_NEW = 1
```

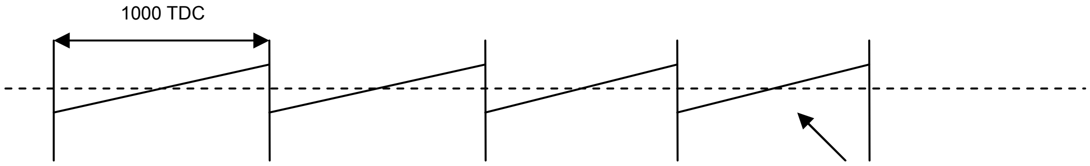
```
                    'Monitoring process during CARB B1/B4 window'
                  EndIf(3)

                  (End of windows not reached)
        Else(2)   call 'Monitoring process at CARB B1/B4 window end'
                   (see description below)
                  call 'CARB B1/B4 ends detection process'
                   (see description below)
                  call 'CARB B1/B4 window reset process' (see description below)
        EndIf(2)
        If(1)  [ LV_MIS_STATE_B1 = 1
                    And ( LV_END_MIS_B4 = 0 Or LC_CONF_MIS_STATE_B = 1 ) ]
             Or     LV_MIS_STATE_B4 = 1
        Then(1)     LV_MIS_STATE_B = 1
        Else(1)     LV_MIS_STATE_B = 0
        EndIf(1)
    Else(1)   no process (Counters &  flags unchanged)
    EndIf(1)
```

MIS_B1 Management



The error is direcly stored without waiting the end of the B1 window

MIS_B4 Management



The error is stored without waiting the end of B4 window, but during the fourth window with misfire detected.

51.7.5.2 CARB B1/B4 Misfire counters increment process

```
MIS_SUM_B = MIS_SUM_B + 1
MIS_SUM_B_CBK[SEG_NR_CBK_ER] = MIS_SUM_B_CBK[SEG_NR_CBK_ER] + 1
                    // Assign only ER current specific exhaut cylinder bank counter
MIS_CTR_B[SEG_NR_ER] = MIS_CTR_B[SEG_NR_ER] + 1
                    // Assign only ER current specific cylinder counter
```

51.7.5.3 Monitoring process at CARB B1/B4 window end

```
If(1)  LV_END_MIS_B1 = 0  // End of CARB B1 window
Then(1)
    If(2)  { LV_MIS_B_DIAG_REQ_APP = 0 And
               [ ( MIS_SUM_B ≥ C_MIS_SUM_B1_THD_MT And LV_AT = 0 )
               Or   ( MIS_SUM_B ≥ C_MIS_SUM_B1_THD_AT And LV_AT = 1 )
               Or   MIS_SUM_B_CBK[0...NC_CBK_EX_NR-1]
                        ≥ C_MIS_SUM_B1_CBK_THD ] }
               // Check for each exhaust cylinder banks
           Or
           { LV_MIS_B_DIAG_REQ_APP = 1 And
               [ ( MIS_SUM_B ≥
```

```
                        ( FAC_MIS_SUM_B1_THD_APP * C_MIS_SUM_B1_THD_MT )
                        And LV_AT = 0 )
                Or      ( MIS_SUM_B ≥
                        ( FAC_MIS_SUM_B1_THD_APP * C_MIS_SUM_B1_THD_AT )
                        And LV_AT = 1 )
                Or      MIS_SUM_B_CBK[0...NC_CBK_EX_NR-1]
                            ≥ FAC_MIS_SUM_B1_THD_APP * C_MIS_SUM_B1_CBK_THD]}
                // Check for each exhaust cylinder banks

        Then(2)     call 'CARB B1 criterion determination process' (see description below)
        Else(2)     no criterion detection
                    LV_MIS_STATE_B1 = 0
        EndIf(2)
    Else(1)
    // Be carefull, the counter of window B4 (CTR_CHK_MIS_B4 ) is incremented only at the end
    of the window and not during.
        CTR_CHK_MIS_B4 = CTR_CHK_MIS_B4 + 1   // at the End of CARB B4 window
        If(2)   { LV_MIS_B_DIAG_REQ_APP = 0 And
                [ ( MIS_SUM_B ≥ C_MIS_SUM_B4_THD_MT And LV_AT = 0 )
                Or      ( MIS_SUM_B ≥ C_MIS_SUM_B4_THD_AT And LV_AT = 1 )
                Or      MIS_SUM_B_CBK[0...NC_CBK_EX_NR-1]
                            ≥ C_MIS_SUM_B4_CBK_THD ] }
                // Check for each exhaust cylinder banks
            Or
            { LV_MIS_B_DIAG_REQ_APP = 1 And
                [ ( MIS_SUM_B ≥
                    ( FAC_MIS_SUM_B4_THD_APP * C_MIS_SUM_B4_THD_MT )
                        And LV_AT = 0 )
                  Or ( MIS_SUM_B ≥
                    ( FAC_MIS_SUM_B4_THD_APP * C_MIS_SUM_B4_THD_AT )
                        And LV_AT = 1 )
                  Or    MIS_SUM_B_CBK[0...NC_CBK_EX_NR-1]
                            ≥ FAC_MIS_SUM_B4_THD_APP * C_MIS_SUM_B4_CBK_THD]}
                // Check for each exhaust cylinder banks

        Then(2)     call 'CARB B4 criterion determination process' (see description below)
```

```
        Else(2)     no criterion detection
                // Be carefull, the 3 following datas are reseted only at the end of the window and not during.
                    LV_DET_MIS_B4 = 0
                    LV_MIS_STATE_B4 = 0
                    SYM_CYL_MIS_B4 = 0
        EndIf(2)
    EndIf(1)
```

51.7.5.4 Monitoring process during CARB B1/B4 window

This part is realized during CARB B1/B4 window when NLC_ENA_SCDN_NEW=1.

In this case, we do exactly the same operations that those done in "**'Monitoring process at CARB B1/B4 window end'.**"

51.7.5.5 CARB B1 criterion determination process:

// Global CARB B1 default

As soon as an error occurs during B1 window, we set the error.

But if the error enters again, we do not want to store it again. This is realized by lv_err_in_win_mis_b1 flag.

```
 if(1) (LV_ERR_IN_WIN_MIS_B1 = 0)
Then(1)
    LV_MIS_STATE_B1 = 1
    LV_ERR_IN_WIN_MIS_B1 = 1
    // CARB B1 cylinder identification
    For   x = 0 to NC_CYL_NR - 1                  // Check for each cylinder x
        If(2)       MIS_CTR_B[x] > C_MIS_B1_MIN_NR * MIS_SUM_B
        Then(2)     Set in SYM_CYL_MIS_B1 the cylinder x concerned bit to 1
        EndIf(2)
    EndFor
    // CARB B1 random cylinder pattern
    If  No specific cylinder counters comply with C_MIS_B1_MIN_NR * MIS_SUM_B
        criterion
    Then
        If(3)      LC_CONF_RND_DET_MIS_B = 0
        Then(3)  Evaluate Cylinder with the highest  MIS_CTR_B[x]  value
```

```
                    Set concerned cylinder bit to 1 in SYM_CYL_MIS_B1 carrier
                    Set RDN bit (random) to 0 in SYM_CYL_MIS_B1 carrier
                Else(3)   Set RDN bit (random) to 1 in SYM_CYL_MIS_B1 carrier
                Endif(3)
            Else  Set RDN bit (random) to 0 in SYM_CYL_MIS_B1 carrier
            EndIf
            // CARB B1 multiple cylinder identification
            If  LC_CONF_MPL_DET_MIS_B = 1
            And  number of cylinder(s) set to 1 in SYM_CYL_MIS_B1 is equal or over
                C_NR_CYL_MPL_MIS_B
            Then      Set MPL bit (multiple) to 1 in SYM_CYL_MIS_B1 carrier
            Else      Set MPL bit (multiple) to 0 in SYM_CYL_MIS_B1 carrier
            End
        Endif(1)
```

51.7.5.6 CARB B4 criterion determination process

```
If(1)   (LV_ERR_IN_WIN_MIS_B4 = 0)
Then(1)
    CTR_DET_MIS_B4 = CTR_DET_MIS_B4 + 1
    LV_DET_MIS_B4 = 1
    LV_ERR_IN_WIN_MIS_B4 = 1
```

As soon as the misfire criterion detected counter reached his threshold, we directly set the error. To avoid to stored it in the same window a second time, we used *lv_err_in_win_mis_b4 flag.*

```
    If(2)   (CTR_DET_MIS_B4 ≥ C_NR_DET_MAX_MIS_B4)
    Then(2)
        // Global CARB B4 default
        LV_MIS_STATE_B4 = 1
        // CARB B4 cylinder identification
        For(3) x = 0 to NC_CYL_NR-1          // Check for each cylinder x
            If(4)   MIS_CTR_B[x] > C_MIS_B4_MIN_NR * MIS_SUM_B
            Then(4)      Set in SYM_CYL_MIS_B4 the cylinder x concerned bit to 1
             EndIf(4)
        EndFor(3)
        // CARB B4 random cylinder pattern
```

```
        If(3)   No specific cylinder counters comply with C_MIS_B4_MIN_NR *
                MIS_SUM_B criterion
        Then (3)
            If(4)     LC_CONF_RND_DET_MIS_B = 0
            Then(4)   Evaluate Cylinder with the highest  MIS_CTR_B[x]  value
                      Set concerned cylinder bit to 1 in SYM_CYL_MIS_B4 carrier
                      Set RDN bit (random) to 0 in SYM_CYL_MIS_B4 carrier
            Else(4)   Set RDN bit (random) to 1 in SYM_CYL_MIS_B4 carrier
            Endif(4)
        Else(3)     Set RDN bit (random) to 0 in SYM_CYL_MIS_B4 carrier
        EndIf(3)
        // CARB B4 multiple cylinder identification
        If(3)   LC_CONF_MPL_DET_MIS_B = 1
            And    number of cylinder(s) set to 1 in SYM_CYL_MIS_B4 is equal or over
            C_NR_CYL_MPL_MIS_B
        Then(3)     Set MPL bit (multiple) to 1 in SYM_CYL_MIS_B4 carrier
        Else(3)     Set MPL bit (multiple) to 0 in SYM_CYL_MIS_B4 carrier
        End(3)
    Else(2)
        SYM_CYL_MIS_B4 = 0
        LV_MIS_STATE_B4 = 0
    EndIf(2)
EndIf(1)
```

51.7.5.7 CARB B1/B4 ends detection process

```
If(1) LV_END_MIS_B1 = 0
Then(1)     LV_END_MIS_B1 = 1
            LV_END_WIN_MIS_B1 = 1 (set for one segment)
Else(1)
    LV_END_WIN_MIS_B4 = 1 (set for one segment)
    If(2)   CTR_DET_MIS_B4 ≥ C_NR_DET_MAX_MIS_B4
            Or CTR_CHK_MIS_B4 ≥ C_NR_DET_MAX_MIS_B4
    Then(2)         LV_END_MIS_B4 = 1
    EndIf(2)
EndIf(1)
```

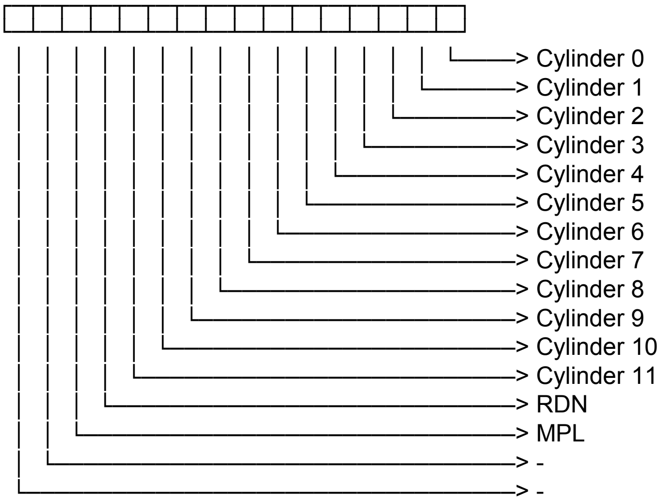
51.7.5.8 CARB B1/B4 window reset process

```
MIS_NR_TDC_B = 0
MIS_SUM_B = 0
LV_ERR_IN_WIN_MIS_B1 = 0
LV_ERR_IN_WIN_MIS_B4 = 0
LV_ERR_MIS_A_IN_WIN_B = 0
For x = 0 : NC_CYL_NR-1
    MIS_CTR_B[x] = 0            // All cylinder specific counters
EndFor
For i = 0 : NC_CBK_EX_NR-1
    MIS_SUM_B_CBK[i] = 0        // All exhaust cylinder bank specific counters
EndFor
```

51.7.5.9 SYM_CYL_MIS_B1 and SYM_CYL_MIS_B4 carriers definition

The location of detected cylinders with misfire status CARB B is coded in two carrier bytes:

SYM_CYL_MIS_B1, SYM_CYL_MIS_B4



Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_MIS_SUM_B1_THD_MT	1	0...FFFFH	0...65535	1	[-]
Global threshold for CARB B1 misfire criterion detection, LV_AT = 0					
C_MIS_SUM_B1_THD_AT	1	0...FFFFH	0...65535	1	[-]
Global threshold for CARB B1 misfire criterion detection, LV_AT = 1					
C_MIS_SUM_B4_THD_MT	1	0...FFFFH	0...65535	1	[-]
Global threshold for CARB B4 misfire criterion detection, LV_AT = 0					
C_MIS_SUM_B4_THD_AT	1	0...FFFFH	0...65535	1	[-]
Global threshold for CARB B4 misfire criterion detection, LV_AT = 1					
C_MIS_B1_MIN_NR	1	0...FFH	0...99.60937	0.390625	[%]
Cylinder identification ratio for CARB B1 criterion detection					
C_MIS_B4_MIN_NR	1	0...FFH	0...99.60937	0.390625	[%]
Cylinder identification ratio for CARB B4 criterion detection					
C_MIS_SUM_B1_CBK_THD	1	0...FFFFH	0...65535	1	[-]
Exhaut cylinder bank threshold for CARB B1 misfire criterion detection					
C_MIS_SUM_B4_CBK_THD	1	0...FFFFH	0...65535	1	[-]
Exhaut cylinder bank threshold for CARB B4 misfire criterion detection					
C_NR_CYL_MPL_MIS_B	1	0...FFH	0...255	1	[-]
Number of cylinder in MIS_Bx to set the multiple misfire CARB Bx status (typical: 2)					
C_NR_DET_MAX_MIS_B4	1	0...FFH	0...255	1	[-]
Maximum value of CTR_DET_MIS_B4 for misfire CARB B4 status detection (typical: 4)					
LC_CONF_MIS_STATE_B	1	0...1H	0...1	1	[-]
Configuration for LV_MIS_STATE_B setting (=0, setting by MIS_B1 only when end of MIS_B4 isn't reached, then only set by MIS_B4) (=1, setting by MIS_B1 or MIS_B4 in any case)					
LC_CONF_MPL_DET_MIS_B	1	0...1H	0...1	1	[-]
Configuration for multiple cylinder status set in SYM_CYL_MIS_Bx (multiple misfire pattern Pcode saved) (=1)					
LC_CONF_RND_DET_MIS_B	1	0...1H	0...1	1	[-]
Configuration for random cylinder detection when no specific cylinder identified,: (=0) Cyl. with the highest value is identified in SYM_CYL_MIS_B1 or SYM_CYL_MIS_B4 or (=1) No identification is realised and RDN bit is set to 1 in SYM_CYL_MIS_B1/B4					

51.8 Misfire Rate Determination And Error Management (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
LV_REQ_APP_CTR_MIS_A	V/O	0...1H	0...1	1	[-]
Specific request to allow out of FTP counting mode even if we are within FTP area					
LV_INH_IGC_DIAG_MIS	V/O	0...1H	0...1	1	[-]
Inhibition of misfire crossed diagnosis with IGC OBD I errors					
LV_INH_IV_DIAG_MIS	V/O	0...1H	0...1	1	[-]
Inhibition of misfire crossed diagnosis with IV OBD I errors					
LV_MIS_A_DIAG_REQ_APP	V/O	0...1H	0...1	1	[-]
Specific request to evaluate CARB A criterion with project specific calibration set according external conditions (bad fuel quality...)					
LV_MIS_B_DIAG_REQ_APP	V/O	0...1H	0...1	1	[-]
Specific request to evaluate CARB B1 & B4 criterions with project specific calibration set according external conditions (bad fuel quality...)					
FAC_MIS_A_APP	V/O	0...64H	0...100	1	[-]
Catalyst damage weighting factor used to increment counters for MIS_A criterion pending application specific conditions					
FAC_MIS_A_THD_IND_APP	V/O	0...1FFH	0...1.99609	3.9063e-3	[-]
Cylinder individual threshold used for cylinder shut-off during CARB A criterion window pending application specific conditions					
FAC_MIS_SUM_A_THD_APP	V/O	0...1FFH	0...1.99609	3.9063e-3	[-]
Global threshold used for CARB A criterion pending application specific conditions					
FAC_MIS_SUM_B1_THD_APP	V/O	0...1FFH	0...1.99609	3.9063e-3	[-]
Global threshold used for CARB B1 criterion pending application specific conditions					
FAC_MIS_SUM_B4_THD_APP	V/O	0...1FFH	0...1.99609	3.9063e-3	[-]
Global threshold used for CARB B4 criterion pending application specific conditions					
LV_CDN_MIS_A	V/O	0...1H	0...1	1	[-]
CARB A misfire criterion monitoring condition					
LV_CDN_MIS_B1	V/O	0...1H	0...1	1	[-]
CARB B1 misfire criterion monitoring condition					
LV_CDN_MIS_B4	V/O	0...1H	0...1	1	[-]
CARB B4 misfire criterion monitoring condition					
LV_CDN_DIAG_MIS[NC_CYL_NR]	V/O	0...1H	0...1	1	[-]
Diagnosis condition specific cylinder misfire					
LV_ERR_MIS[NC_CYL_NR]	V/O	0...1H	0...1	1	[-]
Present error flag specific cylinder misfire					
ERR_SYM_MIS[NC_CYL_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom specific cylinder misfire					
LV_END_DIAG_MIS[NC_CYL_NR]	V/O	0...1H	0...1	1	[-]
End of diagnosis flag					
LV_CDN_DIAG_MIS_FTL_L	V/O	0...1H	0...1	1	[-]
Diagnosis condition misfire with low fuel					
ERR_SYM_MIS_FTL_L	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom misfire with low fuel					
LV_ERR_MIS_FTL_L	V/O	0...1H	0...1	1	[-]
Present error flag misfire with low fuel					
LV_END_DIAG_MIS_FTL_L	V/O	0...1H	0...1	1	[-]
End of diagnosis flag					
LV_CDN_DIAG_MIS_MPL	V/O	0...1H	0...1	1	[-]
Diagnosis condition multiple cylinder misfire					

ERR_SYM_MIS_MPL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom multiple cylinder misfire					
LV_ERR_MIS_MPL	V/O	0...1H	0...1	1	[-]
Present error flag multiple cylinder misfire					
LV_END_DIAG_MIS_MPL	V/O	0...1H	0...1	1	[-]
End of diagnosis flag					
CTR_MIS_DET	V/O/S	0...FFFFH	0...65535	1	[-]
Number of misfire events of all cylinders in one DC					
CTR_SEG_MIS_ACT	V/O/S	0...FFFFFFFFH	0...4294967295	1	[-]
Number of segments with active misfire detection					
FAC_ER_DIAG_MIS	V/O	0...FFH	0...1.99218	0.0078125	[-]
Reliability weighting factor for detection based on misfire detection index (ER)					
CTR_MIS_DET_CYL[NC_CYL_NR]	V/O/S	0...FFFFH	0...65535	1	[-]
Number of lifetime misfire events per cylinder					

Input data:

LV_END_MIS_B1	LV_INH_DET_MIS	LV_DC	TEG_CAT_UP_MDL_MAX
SYM_CYL_MIS_A	SYM_CYL_MIS_B1	SYM_CYL_MIS_B4	LV_MIS_STATE_A
LV_MIS_STATE_B1	LV_MIS_STATE_B4	LV_DET_MIS	LV_END_WIN_MIS_B1
LV_END_WIN_MIS_B4	LV_ERR_IN_WIN_MIS_A	LV_ERR_IN_WIN_MIS_B4	LV_ERR_MIS_A_IN_WIN_B
NLC_ENA_SCDN_NEW	LV_REQ_INH_MIS	SEG_NR_ER	LV_RUN_ENG
LV_FTL_L_DIAG_MIS	LV_END_MIS_B4	LV_END_MIS_B1	LC_CONF_MPL_DET_MIS_B
LC_CONF_MPL_DET_MIS_A	OPM_AV	NC_CMB_CONF	N
LOAD_MIS	N_32	LC_MIS_INH	

Import actions:

ACTION_ERRM_CdnDiagScdn (IN <XX>)
This action is used to calculate a ratio to recognise the similar condition with or without failure
ACTION_ERRM_EndWinScdn (IN <XX>, IN <XX>)
This is used for recognition of similar condition with or without failure

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TEG_CAT_UP_MIS_A_MAX	1	0...7FF0H	0...2047	0.0625	[°C]
TEG Threshold for Misfire A counter management					
LC_CTR_CONF_DIAG_MIS	1	0...1H	0...1	1	[-]
Diagnosis TDC counting mode : counting all TDCs (=0) or only diagnosed TDC's (=1)					
LC_CTR_CONF_OSC_MIS	1	0...1H	0...1	1	[-]
Diagnosis TDC counting mode when crankshaft oscillations occurs: stop counting TDC's (=0) or counting allowed TDC's (=1)					
IP_MIS_A_FAC_AFS	6*6	0...64H	0...100	1	[-]
LDP_N__MIS_A_FAC_AFS	6	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS__MIS_A_FAC_AFS	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Catalyst damage weighting factor used for MIS_A counters increment (AFS combustion mode)					
#IF NC_CMB_CONF = AFS_AFL Or AFS_AFL_S					
IP_MIS_A_FAC_HOM_S	6*6	0...64H	0...100	1	[-]
LDP_N__MIS_A_FAC_HOM_S	6	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS__MIS_A_FAC_HOM_S	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Catalyst damage weighting factor used for MIS_A counters increment (HOM_S combustion mode)					

#ENDIF					
#IF NC_CMB_CONF = AFS_S Or AFS_AFL_S					
IP_MIS_A_FAC_S	6*6	0...64H	0...100	1	[-]
LDP_N__MIS_A_FAC_S	6	0...1FE0H	0...8160	1	[rpm]
LDP_LOAD_MIS__MIS_A_FAC_S	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Catalyst damage weighting factor used for MIS_A counters increment (Stratified combustion mode)					
#ENDIF					
IP_FAC_ER_AFS_DIAG_MIS	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_N_32_IP_FAC_ER_AFS_DIAG_MIS	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_IP_FAC_ER_AFS_DIAG_MIS	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for irregular engine operation (engine speed analysis) in AFS combustion mode					
#IF NC_CMB_CONF = AFS_AFL Or AFS_AFL_S					
IP_FAC_ER_HOM_S_DIAG_MIS	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_N_32_IP_FAC_ER_HOM_S_MIS	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_IP_FAC_ER_HOM_S_MIS	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for irregular engine operation (engine speed analysis) in HOM_S combustion mode					
#ENDIF					
#IF NC_CMB_CONF = AFS_S Or AFS_AFL_S					
IP_FAC_ER_S_DIAG_MIS	6*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_N_32_IP_FAC_ER_S_DIAG_MIS	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_IP_FAC_ER_S_DIAG_MIS	6	0...7FFFH	0...99.99694	3.0518e-3	[%]
Weighting factor for irregular engine operation (engine speed analysis) in S combustion mode					
#ENDIF					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_TREAT_DIAG_MIS	-	0...1H	0...1	1	[-]
Misfire default failures generated per symptoms CARB_A, CARB_B1 & CARB_B4 (NLC_TREAT_DIAG_MIS = 0) or per specific cylinder misfire errors (NLC_TREAT_DIAG_MIS = 1)					

51.8.1 Application Incidences data for diagnosis

Description:

LV_REQ_APP_CTR_MIS_A is used for Misfire A calculation considering the load for the catalyst due to exhaust temperature.

Application conditions:

Initialisation: 0 at reset

Recurrence: ENRD segment synchronous,
After misfire detection, Before misfire diagnosis

Activation: LC_MIS_INH = 0 **and** LV_DC = 1

Deactivation: LC_MIS_INH = 1 **or** LV_DC = 0

Formula section:

LV_INH_IV_DIAG_MIS = LV_INH_DET_MIS

LV_INH_IGC_DIAG_MIS = LV_INH_DET_MIS

Calculation of LV_REQ_APP_CTR_MIS_A

If TEG_CAT_UP_MDL_MAX > C_TEG_CAT_UP_MIS_A_MAX

Then LV_REQ_APP_CTR_MIS_A = 1

```
Else      LV_REQ_APP_CTR_MIS_A = 0
EndIf
```

Request to evaluate CARB A criterion with project specific calibration set

```
LV_MIS_A_DIAG_REQ_APP  = 0
FAC_MIS_A_THD_IND_APP = 0
FAC_MIS_SUM_A_THD_APP = 0
```

Request to evaluate CARB B1 & B4 criterions with project specific calibration set

```
LV_MIS_B_DIAG_REQ_APP  = 0
FAC_MIS_SUM_B1_THD_APP = 0
FAC_MIS_SUM_B4_THD_APP = 0
```

51.8.2 Combustion mode influence on the catalyst damage weigthing factor

At each ER segment recurrence:

FAC_MIS_A_APP must be calculated before "Determination of catalyst damage weighting factor" in the generic specification named **Misfire rate and error management**

Convention for combustion mode determination:

```
#If  NC_CMB_CONF = AFS_AFL Or AFS_AFL_S
    If     OPM_AV = 3          (homogeneous-stratified)
    Then   xx = "HOM_S"
    EndIf
#EndIf
#If  NC_CMB_CONF = AFS_S Or AFS_AFL_S
    If     OPM_AV = 1          (stratified)
    Then   xx = "S"
    EndIf
#EndIf
If         OPM_AV <> 1 and OPM_AV <> 3   (homogeneous)
Then       xx = "AFS"
EndIf

    FAC_MIS_A_APP= IP_MIS_A_FAC_xx(N, LOAD_MIS)
```

51.8.3 Combustion mode gestion for combination of different diagnosis relability level

FUNCTION DESCRIPTION:

General information:

FAC_ER_DIAG_MIS must be calculated before the determination of SUM_FAC_DIAG_MIS (in the generic specification named Misfire rate determination and error mamagement)

Application conditions:

Misfire segment after detection, before CARB diagnosis

Update rate: every segment

Activation:

LC_MIS_INH = 0

Deactivation:

LC_MIS_INH = 1

Formula section:

Initialisation on ECU reset, on LV_DC 0 to 1 transition and on LC_MIS_INH 0 to 1 transition:

FAC_ER_DIAG_MIS = 0

Irregular engine operation weighting (LV_DET_MIS = 1):

If misfiring is suspected by the irregular engine operation test, a weighting factor for the corresponding engine operating point is applied. This factor is derived from tables according combustion mode.

Convention for combustion mode determination:

```
#If  NC_CMB_CONF = AFS_AFL Or AFS_AFL_S
     If     OPM_AV = 3          (homogeneous-stratified)
     Then   xx = "HOM_S"
     EndIf
#EndIf
#If  NC_CMB_CONF = AFS_S Or AFS_AFL_S
     If     OPM_AV = 1          (stratified)
     Then   xx = "S"
     EndIf
#EndIf
If          OPM_AV <> 1 and OPM_AV <> 3   (homogeneous)
Then        xx = "AFS"
```

```
EndIf

If LV_DET_MIS = 1
Then     FAC_ER_DIAG_MIS = IP_FAC_ER_xx_DIAG_MIS(N_32, LOAD_MIS)
Else     FAC_ER_DIAG_MIS = 0
EndIf
```

51.8.4 Determination of CARB misfire criterion monitoring conditions *(ER algorithm integration, NC_MISF_VERS = 1)*

Description:

Depending on customers & legal requirements reached, two Tdc's counting modes are available :

- counting each Tdc's even if some aren't diagnosed *(LC_CTR_CONF_DIAG_MIS = 0)*
- counting only diagnosed Tdc's *(LC_CTR_CONF_DIAG_MIS = 1)*

LV_DC	LC_CTR_CONF_DIAG_MIS	LV_INH_DET_MIS	LC_CTR_CONF_OSC_MIS	LV_STATE_CRK_OSC	LV_CDN_MIS_A LV_CDN_MIS_B1 LV_CDN_MIS_B4	MIS_A, MIS_B1 & MIS_B4 Monitoring Modes
0	x	x	x	x	0	Out of Driving Cycle
1	0	x	x	x	1	All TDC's Counting Mode
1	1	1	x	x	0	Diagnosed TDC's Counting Mode / Classical Fade Out
1	1	0	0	0	1	Diagnosed TDC's Counting Mode / No Fade Out
1	1	0	0	1	0	Classical Fade Out **with** CRK_OSC / Diagnosed TDC's Counting Mode
1	1	0	1	0	1	Diagnosed TDC's Counting Mode / No Fade Out
1	1	0	1	1	1	Classical Fade Out **without** CRK_OSC / Diagnosed TDC's Counting Mode

Application conditions:

Initialization: 0 at LV_DC 0->1 **or** reset **or** LC_MIS_INH 0 to 1 transition

Recurrence: ENRD segment synchronous,

After misfire detection, Before misfire diagnosis

Activation: LC_MIS_INH = 0 **and** LV_DC = 1

Deactivation: LC_MIS_INH = 1 **or** LV_DC = 0

Formula section:

```
If    NC_MISF_VERS = 1
      Module Integrated
Else
      Module Not Integrated
EndIf


If(1)   LC_CTR_CONF_DIAG_MIS = 0                     // Counting each TDC's
```

```
          Or
          { ( LC_CTR_CONF_DIAG_MIS = 1 And LV_INH_DET_MIS = 0 )
          And
          [ LC_CTR_CONF_OSC_MIS = 1
                Or ( LC_CTR_CONF_OSC_MIS = 0 And LV_STATE_CRK_OSC = 0) ] }
                // if Crankshaft oscillation detection module integrated, to be deleted if not
    Then(1)
          LV_CDN_MIS_A = 1                                // Carb A diagnosis active
          If(2)  LV_END_MIS_B1 = 0
          Then(2)
                      LV_CDN_MIS_B1 = 1             // Carb B1 diagnosis active
                      LV_CDN_MIS_B4 = 0             // Carb B4 diagnosis inactive
          Else(2)
                      LV_CDN_MIS_B1 = 0             // Carb B1 diagnosis inactive
                      LV_CDN_MIS_B4 = 1             // Carb B4 diagnosis active
          EndIf(2)
    Else(1)
          LV_CDN_MIS_A = 0            // Carb A diagnosis inactive
          LV_CDN_MIS_B1 = 0           // Carb B1 diagnosis inactive
          LV_CDN_MIS_B4 = 0           // Carb B4 diagnosis inactive
    EndIf(1)
```

51.8.5 Specific cylinder misfire errors

FUNCTION DESCRIPTION:

General information:

There is one LV_ERR_MIS[x] for each cyllinder (x = 0, 1, 2, 3, 4, 5), thus a seperate calibration of P-codes for each symptom is possible.

Description:

If Misfire criterion is detected on one cylinder with at least one of the CARB_A, CARB_B1 or CARB4 default criterions, the corresponding cylinder error is set (error is set directly without debounce). The symptom is always corresponding to the last error occurence, and the error is reseted only when all misfire criterions on this cylinder have disappeared.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> - Cylinder in misfire, Criterion default : MIS_A        (= SYM_0)
 │ │ └───> - Cylinder in misfire, Criterion default : MIS_B1       (= SYM_1)
 │ └─────> - Cylinder in misfire, Criterion default : MIS_B4       (= SYM_2)
 └───────> -                                                       (= SYM_3)
```

Formula section:

Initialisation: at ECU reset **or** LV_DC 0 -> 1 **or** LC_MIS_INH 0 -> 1 for all cylinders

LV_CDN_DIAG_MIS[x] = 0

LV_END_DIAG_MIS[x] = 0

LV_ERR_MIS[x] = 0

ERR_SYM_MIS[x] = 0

Recurrence: The application recurrence is **ENRD segment**

After misfire detection, after misfire diagnosis !

Activation:

```
If(1)       LV_DC = 1
      And   LC_MIS_INH = 0
      And   ( LV_CDN_MIS_A = 1
            Or    LV_CDN_MIS_B4 = 1
            Or    LV_CDN_MIS_B1 = 1 )
Then(1)     For(2) x = 0 ... NC_CYL_NR-1
            LV_CDN_DIAG_MIS[x] = 1          // Diagnosis is active for all cylinders
                  If(3) NLC_ENA_SCDN_NEW = 1
                  Then(3) ACTION_ERRM_CdnDiagScdn (MIS[x])
```

```
                        EndIf(3)
                    EndFor(2)
    Else(1)         For(2)  x = 0 ... NC_CYL_NR-1
                    LV_CDN_DIAG_MIS[x] = 0          // Diagnosis is passive for all cylinders
                    EndFor(2)
    EndIf(1)
```

Formula section:

Remark:

Note : CARRIER*[x]* = 1, stands for bit dedicated to cylinder x inside CARRIER structure is set to 1

ERR_SYM[x]*[0]* = 1, stands for symptom 0 inside ERR_SYM[x] structure is set to 1

[x] (italic) stands for bit assignement within a data, [x] stands for data assignement within an array

Individual cylinder errors set in case of MIS_A failure

```
If(1)   LV_MIS_STATE_A = 1                          // MIS_A failure criterion
    For(2)  x = 0 to NC_CYL_NR - 1                  // Check for each cylinder x
        If(3)       SYM_CYL_MIS_A[x] = 1
        Then(3)     LV_ERR_MIS[x] = 1               // MIS failure on Cyl x
                    ERR_SYM_MIS[x][0] = 1           // MIS_A symptom on Cyl x
                    Delivery the result to Error Management
                    If(4) (NLC_ENA_SCDN_NEW=1 and LV_ERR_MIS[x]=0 ->1)
                    Then(4)     ACTION_ERRM_EndWinScdn(MIS[x],1)
                    End If(4)
        Else(3)     If(4)       SYM_CYL_MIS_B1[x] = 0
                          And   SYM_CYL_MIS_B4[x] = 0
                    Then(4)     LV_ERR_MIS[x] = 0
                                ERR_SYM_MIS[x] = 0H
                                Delivery the result to Error Management
                    Else(4)     ERR_SYM_MIS[x][0] = 0       // MIS_A symptom on Cyl x
                                Delivery the result to Error Management
                    EndIf(4)

        EndIf(3)
    EndFor(2)
 Else(1)    // No MIS_A failure criterion
```

```
            For(2)  x = 0 to NC_CYL_NR - 1                  // Check for each cylinder x
                If(3)       SYM_CYL_MIS_B1[x] = 0
                       And  SYM_CYL_MIS_B4[x] = 0
                Then(3)     LV_ERR_MIS[x] = 0
                            ERR_SYM_MIS[x] = 0H
                            Delivery the result to Error Management
                Else(3)     ERR_SYM_MIS[x][0] = 0           // Symptom MIS_A erased
                            Delivery the result to Error Management
                EndIf(3)
            EndFor(2)
        EndIf(1)
```

Individual cylinder errors set in case of MIS_B1 failure

```
    If(1) (LV_END_MIS_B1 = 0 or LV_END_WIN_MIS_B1 = 1)   // Diagnosis during window MIS_B1
    Then(1)
        If(2)   LV_MIS_STATE_B1 = 1                         // MIS_B1 failure criterion
            For(3)  x = 0 to NC_CYL_NR - 1                  // Check for each cylinder x
                If(4)       SYM_CYL_MIS_B1[x] = 1
                Then(4)     LV_ERR_MIS[x] = 1               // MIS failure on Cyl x
                            ERR_SYM_MIS[x][1] = 1           // MIS_B1 symptom on Cyl x
                            Delivery the result to Error Management
                EndIf(4)
            EndFor(3)
        Else(2)
            // No MIS_B1 failure criterion, MIS_B1 symptom on Cyl x evaluated once per driving cycle
            No operation
        EndIf(2)
        If(5) NLC_ENA_SCDN_NEW = 1
        Then(5)
            For(6)  x = 0 to NC_CYL_NR – 1
                If(7) (LV_ERR_MIS[x]=0→1 or LV_END_WIN_MIS_B1 = 1)
               ACTION_ERRM_EndWinScdn(MIS[x],LV_ERR_MIS[x]+
                LV_ERR_MIS_A_IN_WIN_Bold) // Check for each cylinder
                EndIf(7)
            EndFor(6)
        EndIf(5)
    EndIf(1)
```

Individual cylinder errors set in case of MIS_B4 failure

```
If(1) LV_MIS_STATE_B4 = 1                          // MIS_B4 failure criterion
    For(2) x = 0 to NC_CYL_NR - 1                   // Check for each cylinder x
        If(3)       SYM_CYL_MIS_B4[x] = 1
        Then(3)     LV_ERR_MIS[x] = 1               // MIS failure on Cyl x
                    ERR_SYM_MIS[x][2] = 1                 // MIS_B4 symptom on Cyl x
                     Delivery the result to Error Management

        Else(3)     If(4)       SYM_CYL_MIS_A[x] = 0
                           And  SYM_CYL_MIS_B1[x] = 0
                    Then(4)     LV_ERR_MIS[x] = 0
                                ERR_SYM_MIS[x] = 0H
                               Delivery the result to Error Management
                     Else(4)   ERR_SYM_MIS[x][2] = 0   // MIS_B4 symptom on Cyl x
                              Delivery the result to Error Management
                    EndIf(4)
        EndIf(3)
      EndFor(2)
Else(1)         // No MIS_B4 failure criterion
    For(2) x = 0 to NC_CYL_NR - 1                   // Check for each cylinder x
        If(3)       SYM_CYL_MIS_A[x] = 0
              And   SYM_CYL_MIS_B1[x] = 0
        Then(3)     LV_ERR_MIS[x] = 0
                    ERR_SYM_MIS[x] = 0H
                     Delivery the result to Error Management
        Else(3)     ERR_SYM_MIS[x][2] = 0                 // Symptom MIS_B4 erased
                     Delivery the result to Error Management
        EndIf(3)
      EndFor(2)
EndIf(1)
If(1) (NLC_ENA_SCDN_NEW = 1)
Then(1)
   For(2) x = 0 to NC_CYL_NR - 1
      If(3) (LV_ERR_MIS[x]=0 & LV_END_WIN_MIS_B4=1 )
      Then(3) ACTION_ERRM_EndWinScdn (MIS[x], LV_ERR_IN_WIN_MIS_B4old +
              LV_ERR_MIS_A_IN_WIN_Bold )
```

```
                    // previous values of lv_err_mis_a_in_win_b and lv_err_in_win_mis_b4.
            Else If(4) (LV_ERR_MIS[x]=1 & LV_ERR_IN_WIN_MIS_B4=0->1 )
                    ACTION_ERRM_EndWinScdn (MIS[x], LV_ERR_IN_WIN_MIS_B4)
                End If(4)
            End If(3)
        EndFor(2)
End If(1)
```

End of Diagnosis:

```
If      LV_ERR_MIS[x] = 1
Then    LV_END_DIAG_MIS[x] = 1                      // for all cylinders
Else    LV_END_DIAG_MIS[x] = LV_END_MIS_B4          // for all cylinders
EndIf
```

51.8.6 Multiple cylinder misfire

Description:

If at the end of the concerned checking range (LV_END_WIN_MIS_A), misfire is detected on two or more cylinders, then the “Multiple cylinder misfire” is detected.
Depending if only MIS_A, MIS_B1, MIS_B4 or a combination is active, the concerned bits are set. The error is set directly without debounce.

Error-symptoms are defined to this diagnosis function as following :

```
☐☐☐☐
 │ │ │ └─> Multiple cylinder MIS_A detected      (= SYM_0)
 │ │ └───> Multiple cylinder MIS_B1 detected     (= SYM_1)
 │ └─────> Multiple cylinder MIS_B4 detected     (= SYM_2)
 └───────> Multiple cylinder misfire detected    (= SYM_3)
```

Application conditions:

Initialisation: all outputs with 0 at LV_DC 0->1 **or** reset

Recurrence: The application recurrence is **ENRD segment,**
After misfire detection, After misfire diagnosis

Activation:

```
If      LV_DC = 1
        And     LV_CDN_MIS_A = 1
        And     LC_MIS_INH = 0
        And     ( LC_CONF_MPL_DET_MIS_A = 1          or
                LC_CONF_MPL_DET_MIS_B = 1 )
        And     LV_END_WIN_MIS_A = 1
```

```
    Then    LV_CDN_DIAG_MIS_MPL = 1     // diagnosis is active
    Else    LV_CDN_DIAG_MIS_MPL = 0     // diagnosis is passive
    Endif
```

Formula section:

Error detection Muliple misfire without symptom detection

```
If      LV_CDN_DIAG_MIS_MPL = 1
Then    If      at least two Multiple Misfire Bits are active (Bits MPL are set in
                SYM_CYL_MIS_A, SYM_CYL_MIS_B1, SYM_CYL_MIS_B4)
        Then            ERR_SYM_MIS_MPL = SYM_3
                        LV_ERR_MIS_MPL = 1
        Error detection Muliple misfire MIS_A
        Else If  only MPL Misfire MIS_A is active (Bit MPL is set in SYM_CYL_MIS_A)
        Then            ERR_SYM_MIS_MPL = SYM_0
                        LV_ERR_MIS_MPL = 1
        Error detection Muliple misfire MIS_B1
        Else If only MPL Misfire MIS_B1 is active (Bit MPL is set in SYM_CYL_MIS_B1)
        Then            ERR_SYM_MIS_MPL = SYM_1
                        LV_ERR_MIS_MPL = 1
        Error detection Muliple misfire MIS_B4
        Else If only MPL Misfire MIS_B4 is active (Bit MPL is set in SYM_CYL_MIS_B4)
        Then            ERR_SYM_MIS_MPL = SYM_2
                        LV_ERR_MIS_MPL = 1
        No Muliple misfire
        Else If         no BIT Multiple Misfire is active (all Bits MPL are 0)
        Then            ERR_SYM_MIS_MPL = NO_SYM
                        LV_ERR_MIS_MPL = 0
        Endif
Else    waiting for diagnosis condition ( end of diagnosis window MIS_A)
Endif
```

End of Diagnosis

```
If              LV_ERR_MIS_MPL   = 1    or
                LV_END_MIS_B4    = 1
Then            LV_END_DIAG_MIS_MPL = 1
Endif
```

51.8.7 Misfire with low fuel signal active

Description:

If Misfire is detected (CARB_A / CARB_B1/4) with low fuel signal active then a error bit is set in order to supply a own P-code requested by CARB (error is set directly without debounce)

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                                   (= SYM_0)
 │ │ └───> Misfire with active fuel-tank reserve (= SYM_1)
 │ └─────> -                                   (= SYM_2)
 └───────> -                                   (= SYM_3)
```

Application conditions:

Initialisation: all outputs with 0 at LV_DC 0->1 **or** reset

Recurrence: The application recurrence is ***ENRD segment***,

After misfire detection, After misfire diagnosis

Activation:

```
If          LV_DC = 1
      And   LV_CDN_MIS_A = 1
      And   LC_MIS_INH = 0
Then        LV_CDN_DIAG_MIS_FTL_L = 1    (diagnosis is active)
Else        LV_CDN_DIAG_MIS_FTL_L = 0    (diagnosis is passive)
Endif
```

Formula section:

Error detection

```
If          ( LV_MIS_STATE_A  = 1          or
              LV_MIS_STATE_B1  = 1         or
              LV_MIS_STATE_B4  = 1 )
      and
              LV_FTL_L_DIAG_MIS = 1        fuel reseve active and calibrated

Then          ERR_SYM_MIS_FTL_L = SYM_1
              LV_ERR_MIS_FTL_L = 1               set directly
Else          ERR_SYM_MIS_FTL_L = NO_SYM
              LV_ERR_MIS_FTL_L = 0               reset
Endif
```

End of Diagnosis

```
If      LV_ERR_MIS_FTL_L = 1      or
        LV_END_MIS_B4      = 1
Then    LV_END_DIAG_MIS_FTL_L = 1
Endif
```

51.8.8 Misfire event counters

Description:

The counters should be used for reading out diagnosis data in the workshop. All Data are stored in NVMY. The DC – related counters (CTR_SEG_MIS_ACT and CTR_MIS_DET) are initialized if engine is started or if NVMY is cleared.

Application conditions:

Initialisation: CTR_SEG_MIS_ACT and CTR_MIS_DET are initialized from MVMY and set to 0 at LV_RUN_ENG 0->1 **or** at clearing misfire NVMY variables.

CTR_MIS_DET_CYL[i] are initialized from MVMY and only set to 0 at clearing misfire NVMY variables

Recurrence: ER segment task, after ER determination process & Misfire detection

Activation:	LV_REQ_INH_MIS = 0	increment counters
Deactivation:	LV_REQ_INH_MIS = 1	counters are unchanged

Formula section:

Calculation of CTR_SEG_MIS_ACT

$CTR\_SEG\_MIS\_ACT_{n} = CTR\_SEG\_MIS\_ACT_{n-1} + 1$

//counting every segment if MSF detetion is active

Calculation of CTR_MIS_DET

```
If      LV_DET_MIS = 1
Then    $CTR\_MIS\_DET_{n} = CTR\_MIS\_DET_{n-1} + 1$
        $CTR\_MIS\_DET\_CYL[SEG\_NR\_ER]_{n} =$
        $CTR\_MIS\_DET\_CYL[SEG\_NR\_ER]_{n-1} + 1$

Endif
```

Remark for all counters:

If the counters reaches the end of Hex-limit, then the inrementing process is stopped and value remain on max value.

51.9 Crankshaft oscillation detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_STATE_CRK_OSC	V/O	0...1H	0...1	1	[-]
Status of crankshaft oscillation condition					
LV_CRK_OSC_DET_ACT	V	0...1H	0...1	1	[-]
Crankshaft oscillation detection active					
DRV0_ER_SUM_OSC	V	0...FFFFH	0...65535	1	[µs]
Integrated DRV0_ER value for amplitude criterion reseted every sign change or DRV0					
DRV0_ER_SUM_THD_OSC	V	0...FFFFH	0...65535	1	[µs]
Threshold for DRV0_ER_SUM_OSC to fulfill amplitude criteria					
SEG_SUM_FRQ_CRK_OSC	V	0...FFH	0...255	1	[-]
Segment counter for frequency criterion reseted every sign change or DRV0					
RATIO_VS_N_CRK_OSC	V	0...FFH	0...0.05	1.96E-04	[(km/h)/rpm]
Ratio VS / N for gear area determination					
CTR_TDC_CRK_OSC_ACT	V	0...FFH	0...255	1	[-]
TDC counter to enable crankshaft oscillation detection after misfiring detection					
LV_FRQ_CRIT_CRK_OSC	V	0...FFH	0...255	1	[-]
Critical crankshaft oscillaction frequency detected					
LV_AMPL_CRIT_CRK_OSC	V	0...FFH	0...255	1	[-]
Critical crankshaft oscillaction amplitude detected					

Input data:

N_32	LV_DET_CFM_MIS	VS	DRV0_ER
N	LOAD_MIS	LV_ENA_ER	LC_MIS_INH
LV_INH_CRK_OSC_DET			

51.9.1 General information

The misfire detection based on engine roughness index (ER) may be disturbed by crankshaft oscillations when single misfire (random) occurs, especially for front drive vehicles crankshaft in the low engine speed / high engine load area (in combination with a 3rd, 4th or 5th gear ratio). A single misfire acts as a drivetrain / crankshaft oscillations trigger, in this case the ER index is distrubed by these oscillations, in some conditions it is practically not possible to distinguish on ER index, real misfire towards speed drop caused through crankshaft speed oscillations.

The trigger for such oscillations can be a single misfire, an obstacle, a big torque change or others instantaneous conditions.

The principle of this crankshaft / drivetrain oscillation detection is to detect such an oscillation and then to fade-out the misfire detection for a short period to avoid over-detection.

Initialisation: *on ECU reset, on LC_MIS_INH 0 to 1 transition*

SEG_SUM_FRQ_CRK_OSC = 0

DRV0_ER_SUM_OSC = 0

LV_STATE_CRK_OSC = 0

Application conditions:

Activation/Deactivation:

```
RATIO_VS_N_CRK_OSC  = VS / N
If    LV_DET_CFM_MIS = 1
Then        CTR_TDC_CRK_OSC_ACT = C_NR_TDC_MAX_CRK_OSC
ElseIf      CTR_TDC_CRK_OSC_ACT ≠ 0
Then        CTR_TDC_CRK_OSC_ACT = CTR_TDC_CRK_OSC_ACT - 1
EndIf
If    LV_ENA_ER = 1
      And   LC_MIS_INH = 0
      And   LV_INH_CRK_OSC_DET = 0
      And   N_32 < C_N_MAX_CRK_OSC
      And   RATIO_VS_N_CRK_OSC > C_RATIO_VS_N_MIN_CRK_OSC
      And   CTR_TDC_CRK_OSC_ACT ≠ 0
Then        LV_CRK_OSC_DET_ACT = 1        // Crankshaft oscillation detection active
Else        LV_CRK_OSC_DET_ACT = 0        // Crankshaft oscillation detection inactive
            SEG_SUM_FRQ_CRK_OSC = 0
            DRV0_ER_SUM_OSC = 0
EndIf
```

Update rate: Segment

51.9.2 Detection description

The final decision about crankshaft oscillation consists of two parts:

- The detection of a critical engine roughness oscillation frequency.
- The detection of a critical engine roughness oscillation amplitude.

If both parts fulfilled at the same time their own detection criterion, a crankshaft oscillation is detected. The reason therefore is, that even at constant speed due to normal little speed variations a high frequency or at a normal acceleration phase a big amplitude is possible.

51.9.2.1 Determination of a critical engine roughness frequency

Application conditions:

Activation: LV_CRK_OSC_DET_ACT = 1

Deactivation: LV_CRK_OSC_DET_ACT = 0

Formula section:

```
If    Bit sign between $DRV0\_ER_{(n-1)}$ and $DRV0\_ER_{(n)}$ changes
Then        SEG_SUM_FRQ_CRK_OSC = IP_SEG_MIN_FRQ_CRK_OSC(N_32)
ElseIf      SEG_SUM_FRQ_CRK_OSC ≠ 0
Then        SEG_SUM_FRQ_CRK_OSC = SEG_SUM_FRQ_CRK_OSC - 1
```

```
EndIf
If    SEG_SUM_FRQ_CRK_OSC ≠ 0
Then        LV_FRQ_CRIT_CRK_OSC = 1        // Critical crankshaft frequency status is active
Else        LV_FRQ_CRIT_CRK_OSC = 0
EndIf
```

51.9.2.2 Determination of a critical engine roughness oscillation amplitude

Application conditions:

Activation: LV_CRK_OSC_DET_ACT = 1

Deactivation: LV_CRK_OSC_DET_ACT = 0

Formula section:

```
If    Bit sign between DRV0_ER$_{(n-1)}$ and DRV0_ER$_{(n)}$ changes
Then        DRV0_ER_SUM_OSC$_{(n)}$ = |DRV0_ER$_{(n)}$|
Else        DRV0_ER_SUM_OSC$_{(n)}$ = DRV0_ER_SUM_OSC$_{(n-1)}$ + |DRV0_ER$_{(n)}$|
            // with saturation
EndIf
DRV0_ER_SUM_THD_OSC = IP_DRV0_ER_THD_OSC(N_32, LOAD_MIS)
If    DRV0_ER_SUM_OSC > DRV0_ER_SUM_THD_OSC
Then        LV_AMPL_CRIT_CRK_OSC = 1        // Critical crankshaft amplitude status is active
Else        LV_AMPL_CRIT_CRK_OSC = 0
EndIf
```

51.9.2.3 Determination of crankshaft oscillation final status

Application conditions:

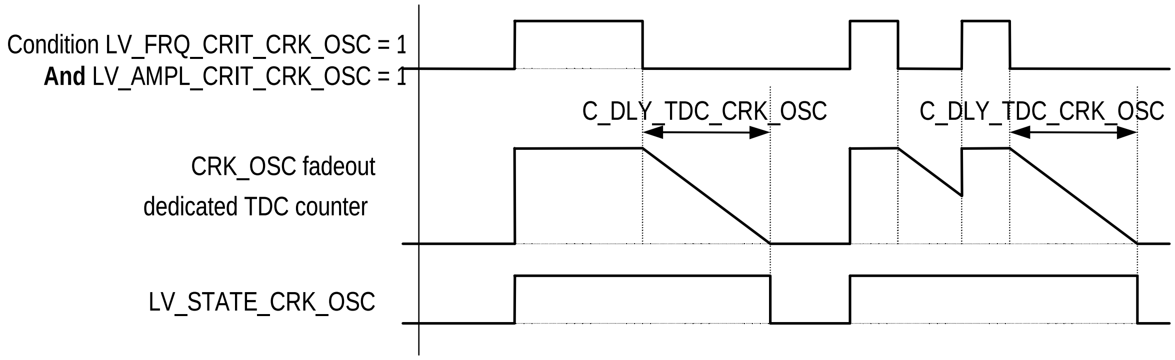
Activation: LV_CRK_OSC_DET_ACT = 1

Deactivation: LV_CRK_OSC_DET_ACT = 0

Formula section:

If in the same time, both conditions are active, a crankshaft oscillation status is triggered :

```
If    LV_FRQ_CRIT_CRK_OSC = 1
      And   LV_AMPL_CRIT_CRK_OSC = 1
Then  After condition rising edge triggering, LV_STATE_CRK_OSC flag is set to 1 as long
      as condition is true.
      After condition falling edge triggering, LV_STATE_CRK_OSC flag is hold to 1 for a
      period of C_DLY_TDC_CRK_OSC tdc's, even if LV_CRK_OSC_DET_ACT is then
      set to 0.
EndIf
```



Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_CRK_OSC	1	0...FFH	0...8160	32	[rpm]
Maximum speed for crankshaft oscillation calculation					
C_DLY_TDC_CRK_OSC	1	0...FFH	0...255	1	[-]
Active TDC’s of crankshaft oscillation status after detection triggering					
C_NR_TDC_MAX_CRK_OSC	1	0...FFH	0...255	1	[-]
Maximum TDC after detected misfire to activate crankshaft oscillation detection process					
C_RATIO_VS_N_MIN_CRK_OSC	1	0...FFH	0...0.05	1.96E-04	[(km/h)/rpm]
Minimum VS / N ratio to activate crankshaft oscillation detection					
IP_SEG_MIN_FRQ_CRK_OSC	4	0...FFH	0...255	1	[-]
LDP_N_32_IP_SEG_MIN_FRQ_CRK_OSC	4	0...FFH	0...8160	32	[rpm]
Minimum number of segment after turn of speed direction for detecting critical high frequency					
IP_DRV0_ER_THD_OSC	6*6	0...FFFFH	0...65535	1	[µs]
LDP_N_32_IP_DRV0_ER_THD_OSC	6	0...FFH	0...8160	32	[rpm]
LDP_LOAD_MIS_IP_DRV0_ER_THD_OSC	6	0...7FFFH	0...99.99694	3.05E-03	[%]
DRV0 amplitude threshold to detect critical crankshaft oscillation amplitude					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_USE_CRK_OSC_MIS	1	0...1H	0...1	1	[-]
Crankshaft oscillation detection module used (=1) or stub version (=0)					

51.10 Rough road detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_RR_PREV_INI	V	0...1H	0...1	1	[-]
Boolean for initialisation of VS_RR_EDGE_T/CTR_AV(n-1) at activation of function					
LV_STATE_RR	V/O	0...1H	0...1	1	[-]
Boolean for state of rough road detection (No : Yes).					
SEG_T_MES_0_RR	V	0...1FFFFFFFH	0...4294967288	8	[µs]
Actual segment time measurement from the wheel speed signal.					
SEG_T_MES_1_RR	V	0...1FFFFFFFH	0...4294967288	8	[µs]
Previous segment time measurement from the wheel speed signal.					
SUM_RR	V	0...FFH	0...255	1	[-]
Counter of rough road detection events.					
VS_RR_EDGE_CTR_AV	V	0...FFFFH	0...65535	1	[-]
actual value of vehicle speed tooth counter on rough road (free running counter)					
VS_RR_EDGE_T_AV	V	0...FFFFFFFFH	0...34359738.36	0.008	[ms]
actual timestamp of last tooth on rough road(free running timer)					
VS_RR_EDGE_T_AV_MAX	V	0...FFFFFFFFH	0...34359738.36	0.008	[ms]
actual maximum timestamp of last tooth on rough road(free running timer)					
WHEEL_GRD_MMV	V	0...3FFFH	0...999.93896	0.0610352	[°/oo]
Rough road value					
WHEEL_GRD_MMV_THD	V	0...3FFFH	0...999.93896	0.0610352	[°/oo]
Rough road value threshold					

Input data:

VS	LV_ERR_VS	LV_AT	

Import Actions:

ACTION_INFR_GetVsRrPulsStamp(**OUT**< vs_rr_edge_ctr_av >, **OUT**<vs_rr_edge_t_av>, **OUT**<vs_rr_edge_t_av_max)
Reading of information about vehicle speed on rough road

The above imported action is defined in the VHMD – IRS (Infrastructure Requirement Specification)

FUNCTION DESCRIPTION:

General information:

Rough road conditions must be detected to prevent erroneous misfire detection because of the influence jolty tracks have on the crankshaft as a result of transmission via the drive train.
The speed disturbance of one wheel can be used to calculate the road conditions. Therefore, the control unit uses the signal from the driven wheel on the right side (ABS sensor or an additive sensor for the version without ABS) because the probability of the presence of a jolty track is greater than left.

However, transmission from the left to the right side has also been observed depending on the chassis and other circumstances, so that a limited detection of unilateral rough-road conditions on the left side is also possible.
The minimum number of integrated teeth for one segment depends on the number of teeth per revolution of the wheel speed. An angle of about 30 ° is preferable (e.g. an integration of 4 teeth is used for a wheel with 48 teeth). This number is determined by the non calibrateable constant NC_SEG_TOOTH_RR.
At higher speed ranges the number of integrated teeth for one segment is determined by the recurrence time of 10 ms.

Application conditions:

Initialisation:	SUM_RR = 0 at ECU-reset	
	LV_RR_PREV_INI = 1 at ECU-reset and at function deactivation	
	all other outputs = 0 at ECU-reset and at function deactivation	
Recurrence:	10 ms	
Activation:	LV_ERR_VS = 0	**and**
	VS $\geq$ C_VS_MIN_RR	**and**
	VS $\leq$ C_VS_MAX_RR	
Deactivation:	LV_ERR_VS = 1	**or**
	VS $<$ C_VS_MIN_RR	**or**
	VS $>$ C_VS_MAX_RR	

Remark: to avoid unnecessary burden of the micro controler it is recommended to calibrate C_VS_MAX_RR according the needs. At vehicle speeds above approximately 100 km/h there is no more feed back from wheels to crankshaft.

Formula section:

ACTION_INFR_GetVsRrPulsStamp(VS_RR_EDGE_CTR_AV, VS_RR_EDGE_T_AV, VS_RR_EDGE_T_AV_MAX)

Initialisation of VS_RR_EDGE_T_AV/CTR(n-1) and new_rr_clc:

```
If      LV_RR_PREV_INI = 1
then    VS_RR_EDGE_T_AV(n-1) = VS_RR_EDGE_T_AV(n)
        VS_RR_EDGE_CTR_AV(n-1) = VS_RR_EDGE_CTR_AV(n)
        new_rr_clc = 2
        LV_RR_PREV_INI = 0
endif
```

a) Segment period acquisition

The function uses consecutive segment times for calculation of a wheel speed gradient. The segments are built with a minimum number of NC_SEG_TOOTH_RR edges and afterwards normalized to a length of 8 teeth.

If VS_RR_EDGE_CTR_AV $_n$ – VS_RR_EDGE_CTR_AV $_{n-1}$ >= NC_SEG_TOOTH_RR
then

$$\mathrm{SEG\_T\_MES\_0\_RR} = \left| \frac{(\mathrm{VS\_RR\_EDGE\_T\_AV}_n - \mathrm{VS\_RR\_EDGE\_T\_AV}_{n-1}) * 8}{\mathrm{VS\_RR\_EDGE\_CTR\_AV}_n - \mathrm{VS\_RR\_EDGE\_CTR\_AV}_{n-1}} \right|$$

SEG_T_MES_1_RR is the segment period preceding SEG_T_MES_0_RR.

endif

b) Calculation of rough road value

$$WHEEL\_GRD = \left| \frac{SEG\_T\_MES\_1\_RR - SEG\_T\_MES\_0\_RR}{SEG\_T\_MES\_1\_RR} \right| * 1000[‰]$$

The raw values WHEEL_GRD are used in a floating averaging, to restrict suppression to actual rough road conditions.

$$\mathrm{WHEEL\_GRD\_MMV}_n = \mathrm{WHEEL\_GRD\_MMV}_{n-1} + \mathrm{C\_WHEEL\_GRD\_CRLC\_RR} * (\mathrm{WHEEL\_GRD}_n - \mathrm{WHEEL\_GRD\_MMV}_{n-1})$$

When the calculation of the wheel speed segment duration is stopped due to the vehicle speed condition (see above), the calculation of WHEEL_GRD and WHEEL_GRD_MMV is stopped as well until again 2 consecutive segment durations could be determined (realised by the internal SW counter new_rr_clc).

Set of VS_RR_EDGE_T_AV/CTR(n-1) after each calculation of WHEEL_GRD(_MMV):
VS_RR_EDGE_T_AV(n-1) = VS_RR_EDGE_T_AV(n)
VS_RR_EDGE_CTR_AV(n-1) = VS_RR_EDGE_CTR_AV(n)

c) Detection of rough road status

WHEEL_GRD_MMV_THD = IP_WHEEL_GRD_MMV_THD_XX__VS

(XX = AT for automatic transmission, LV_AT = 1 ; XX = MT for manual transmission, LV_AT = 0)

```
If   WHEEL_GRD_MMV ≥ WHEEL_GRD_MMV_THD
     then   LV_STATE_RR = 1    and  timer DLY_ER_RR becomes started    and
            increment SUM_RR by one
     elseif timer DLY_ER_RR > 0
            then LV_STATE_RR = 1   and  decrement timer DLY_ER_RR
            else LV_STATE_RR = 0
     endif
endif
```

The start value of the timer DLY_ER_RR is C_DLY_ER_RR.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_MIN_RR	1	0...FFH	0...255	1	[km/h]
Minimum vehicle speed for rough road detection.					
C_VS_MAX_RR	1	0...FFH	0...255	1	[km/h]
Maximum vehicle speed for rough road detection.					
C_WHEEL_GRD_CRLC_RR	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation factor for floating averaging of the rough road raw values					
C_DLY_ER_RR	1	0...FFFFH	0...655350	10	[ms]
Fade out time for engine roughness after rough road detection					
IP_WHEEL_GRD_MMV_THD_AT__VS	9	0...1052H	0...255	0.0610352	[°/oo]
Threshold versus vehicle speed for rough road detection, automatic transm					
IP_WHEEL_GRD_MMV_THD_MT__VS	9	0...1052H	0...255	0.0610352	[°/oo]
Threshold versus vehicle speed for rough road detection, manual transm.					
LDPM_VS_WHEEL_GRD_MMV_THD	9	0...FFH	0...255	1	[km/h]
-					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_SEG_TOOTH_RR	1	1...FFH	1...255	1	[-]
Number of wheel speed signal teeth to build one segment, typical value = 4					

51.11 Misfire tuning functions

51.11.1 Misfire software generator

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
INH_IV_MIS_GEN	V/O	0...FFH	0...255	1	[-]
Identification of cylinders shut off by injection misfire generator					
INH_IGC_MIS_GEN	V/O	0...FFH	0...255	1	[-]
Identification of cylinders shut off by ignition misfire generator					
LV_MIS_GEN	V	0...1H	0...1	1	[-]
Flag to indicate the missing ignition and/or injection (current segment)					
LV_MIS_GEN_DET	V	0...1H	0...1	1	[-]
Flag to indicate the missing ignition and/or injection, delayed to be synchronised with misfire detection flags					

Input data:

SEG_NR	N	TD_IGC[NC_CYL_NR]	IGA_AV[NC_CYL_NR]
EOI_LIM[NC_CYL_NR]	SOI_MAX		

FUNCTION DESCRIPTION:

According the project SI integration plan and/or customer requirements/wishes, a software misfire generator can be included on EMS softwares and used during integration validation and calibration stages. It is strongly recommended to remove this functionally on serial product software: integration choice via NLC_USE_MIS_GEN compilation switch.

Application conditions:

Initialisation: at reset & engine stalling

LV_MIS_GEN = 0

LV_MIS_GEN_DET = 0

INH_IV_MIS_GEN = 0

INH_IGC_MIS_GEN = 0

No function calls to INJR & IGRE

Recurrence: Segment

Activation:

The regular or random misfiring generation is applied on ignition and/or injection sequences.

```
If      LC_IGN_OFF_MIS_GEN = 1
Then    Ignition cut-off mode
        (using INH_IGC_MIS_GEN carrier, see § Ignition misfire realisation)
        LV_MIS_GEN is set to 1 according misfiring generation mode selected
        (pseudo random or regular mode, see dedicated chapters below)
EndIf
If      LC_INJ_OFF_MIS_GEN = 1
```

```
Then    Injection cut-off mode
        (using INH_IV_MIS_GEN carrier, see § Injection misfire realisation)
        LV_MIS_GEN is set to 1 according misfiring generation mode selected
        (pseudo random or regular mode, see dedicated chapters below)
EndIf
If      LC_IGN_OFF_MIS_GEN = 1
        And LC_INJ_OFF_MIS_GEN = 1
Then    Ignition and injection off modes (for the same TDC)
        (using INH_IGC_MIS_GEN & INH_IV_MIS_GEN carrier)
        LV_MIS_GEN is set to 1 according misfiring generation mode selected
        (pseudo random or regular mode, see dedicated chapters below)
EndIf
If      LC_IGN_OFF_MIS_GEN = 0
        And LC_INJ_OFF_MIS_GEN = 0
Then    No misfire generation allowed
        LV_MIS_GEN = 0
        INH_IV_MIS_GEN = 0
        INH_IGC_MIS_GEN = 0
        No function calls to INJR & IGRE
EndIf
LV_MIS_GEN_DET = LV_MIS_GEN delayed with C_NR_TDC_DLY_MIS_GEN tdc's
```

51.11.1.1 Pseudo random misfiring generation mode

Application conditions:

The misfiring generation mode is authorised when the misfiring tuning function and the pseudo random misfiring tuning function are active.

This misfiring generation mode is global to all cylinders.

Recurrence: Segment

Activation:

```
If      LC_REQ_MIS_GEN = 1
    And C_MOD_MIS_GEN = 0
    And SEG_NR = C_SEG_ST_MIS_GEN
Then    Pseudo random misfiring generation is triggered
EndIf
```

Deactivation:

```
If      LC_REQ_MIS_GEN = 0
    Or  C_MOD_MIS_GEN ≠ 0
Then    Pseudo random misfiring generation mode is disabled
EndIf
```

Formula section:

A random ignition and/or injection is missing following the bitfield sequence ID_RND_PAT_MIS_GEN, where each bit corresponds to a TDC.

If the current bit value is 0 then the ignition and/or injection are missing; else ignition and injection are realised.

During the TDC where the ignition and/or injection are missing **And** if at least one of the cut-off mode is allowed (LC_IGN_OFF_MIS_GEN = 1 **Or** LC_INJ_OFF_MIS_GEN = 1), the flag LV_MIS_GEN is set to 1 else the flag LV_MIS_GEN is set to 0.

51.11.1.2 Regular misfiring generation mode

Application conditions:

This generation mode is authorised when the misfiring tuning function and the regular misfiring tuning function are active.

This misfiring generation mode is global to all cylinders.

Recurrence: Segment

Activation:

```
If      LC_REQ_MIS_GEN = 1
    And C_MOD_MIS_GEN = 1
    And SEG_NR = C_SEG_ST_MIS_GEN
Then    Regular misfiring generation mode is triggered
EndIf
```

Deactivation:

```
If      LC_REQ_MIS_GEN = 0
    Or  C_MOD_MIS_GEN ≠ 1
Then    Regular misfiring generation mode is disabled
EndIf
```

Formula section:

Regular ignition and/or injection are missing every C_MIS_RATE tdc.

During the TDC where the ignition and/or injection are missing **And** if at least one of the cut-off mode is allowed (LC_IGN_OFF_MIS_GEN = 1 **Or** LC_INJ_OFF_MIS_GEN = 1), the flag LV_MIS_GEN is set to 1 else the flag LV_MIS_GEN is set to 0.

51.11.1.3 Cylinder regular misfiring generation mode

Application conditions:

This generation mode is authorised when the misfiring tuning function and the regular misfiring tuning function are active.

This misfiring mode is dedicated to specific cylinder generation.

Recurrence: Segment

Activation:

```
If    LC_REQ_MIS_GEN = 1
      And    C_MOD_MIS_GEN = 2
      And    SEG_NR = C_SEG_ST_MIS_GEN
Then         Cylilnder regular misfiring generation mode is triggered
EndIf
```

Deactivation:

```
If    LC_REQ_MIS_GEN = 0
      Or     C_MOD_MIS_GEN ≠ 2
Then         Cylinder regular misfiring generation mode is disabled
EndIf
```

Formula section:

```
If(1)    LC_MIS_RATE_ENA_CYL[SEG_NR] = 1
Then(1)  Ignition and/or injection are missing on cylinder SEG_NR every
         C_RATE_MIS_CYL[SEG_NR] tdc's.
      If(2)    During the TDC where the ignition And/Or injection are missing
               And at least one of the cut-off mode is allowed (LC_IGN_OFF_MIS_GEN = 1 Or
               LC_INJ_OFF_MIS_GEN = 1),
      Then(2)        LV_MIS_GEN = 1
      Else(2)        LV_MIS_GEN = 0
      EndIf(2)
Else(1)  No Ignition and/or injection are missing on cylinder SEG_NR
         LV_MIS_GEN is set to 0
EndIf(1)
```

51.11.1.4 Injection misfire realisation

Recurrence: Every engine cycle, phased with required cylinder injection

Formula section:

The injection misfire realisation is realised via the INH_IV_MIS_GEN carrier. This is an input for the Injection Cylinder Shut-Off Application Incidences file.

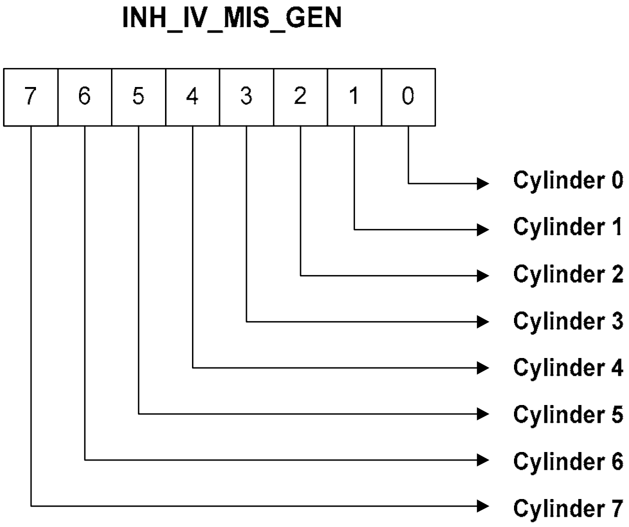
Cylinder switched off by injection software misfire generator is set to 1 within INH_IV_MIS_GEN carrier.

All others cylinder bits are reset to 0 only if not set by the misfire pattern defined for the current engine cycle

Important note on injection shut-off triggering:

To be effective, the injection shut-off on Cylinder x must be triggered before the injection time starts on this cylinder. To evaluate the correct phasing for the injection shut-off, the maximum SOI *(start of injection angle)* is evaluated.

INH_IV_MIS_GEN carrier must be updated at the segment just before the maximum start of injection (SOI_MAX + EOI_LIM[0]) to be strictly taken in account on this injection phase.



51.11.1.5 Ignition misfire realisation

Recurrence: Every engine cycle, phased with required cylinder /coil ignition

Formula section:

The ignition misfire realisation is realised via the INH_IGC_MIS_GEN carrier. This is an input for the Ignition Cylinder Shut-Off Application Incidences file.

Cylinder switched off by ignition software misfire generator is set to 1 within INH_IGC_MIS_GEN carrier.

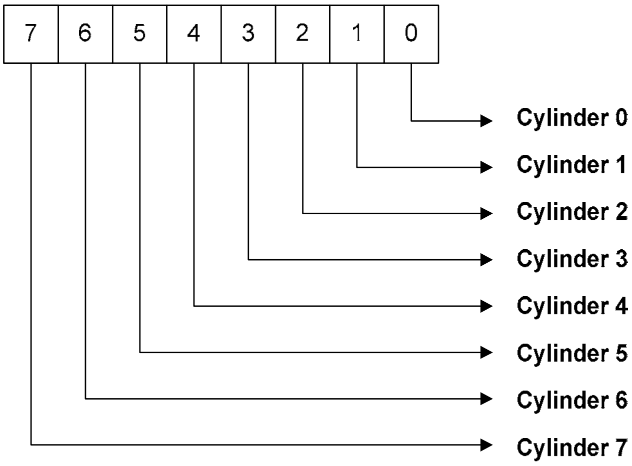
All others cylinder bits are reset to 0 only if not set by the misfire pattern defined for the current engine cycle

Important note on ignition shut-off triggering:

To be effective, the ignition shut-off on cylinder x must be triggered on the segment before the dwell time on the coil y starts for this cylinder x.

To evaluate the correct phasing for the ignition shut-off, the period TD_IGC[x] must be converted in angle and added to the IGA_AV[x] angle to defined the correct segment for triggering.

INH_IGC_MIS_GEN



Calibration data:

Name	Dim.	Hex. limits	Phys. limits	Resol.	Unit
LC_REQ_MIS_GEN	1	0...1H	0...1	1	[-]
Misfire software generator request					
C_MOD_MIS_GEN	1	0...2H	0...2	1	[-]
Misfire software generator function mode					
C_MIS_RATE	1	0...FFH	0...255	1	[-]
Misfire rate for regular misfire generator mode					
C_MIS_RATE_CYL[NC_CYL_NR]	1	0...FFH	0...255	1	[-]
Misfire rate for cylinder regular misfire generator mode					
LC_IGN_OFF_MIS_GEN	1	0...1H	0...1	1	[-]
Boolean for misfire generation with ignition cut-off (=1)					
LC_INJ_OFF_MIS_GEN	1	0...1H	0...1	1	[-]
Boolean for misfire generation with injection cut-off (=1)					
C_SEG_ST_MIS_GEN	1	0...7H	0...7	1	[-]
Start segment for misfire software generator triggering					
C_NR_TDC_DLY_MIS_GEN	1	0...FH	0...15	1	[-]
Delay to synchronise LV_MIS_GEN_DET with misfire detection flags					
LC_MIS_RATE_ENA_CYL[NC_CYL_NR]	1	0...1H	0...1	1	[-]
Booleans to activate cylinder regular misfire generator mode per cylinder					
ID_RND_PAT_MIS_GEN	16*16	0...1H	0...1	1	[-]
LDP_X_ID_RND_PAT_MIS_GEN	16	0...FFH	0...255	1	[-]
LDP_Y_ID_RND_PAT_MIS_GEN	16	0...FH	0...15	1	[-]
Table of Boolean to define the pseudo random misfire sequence					

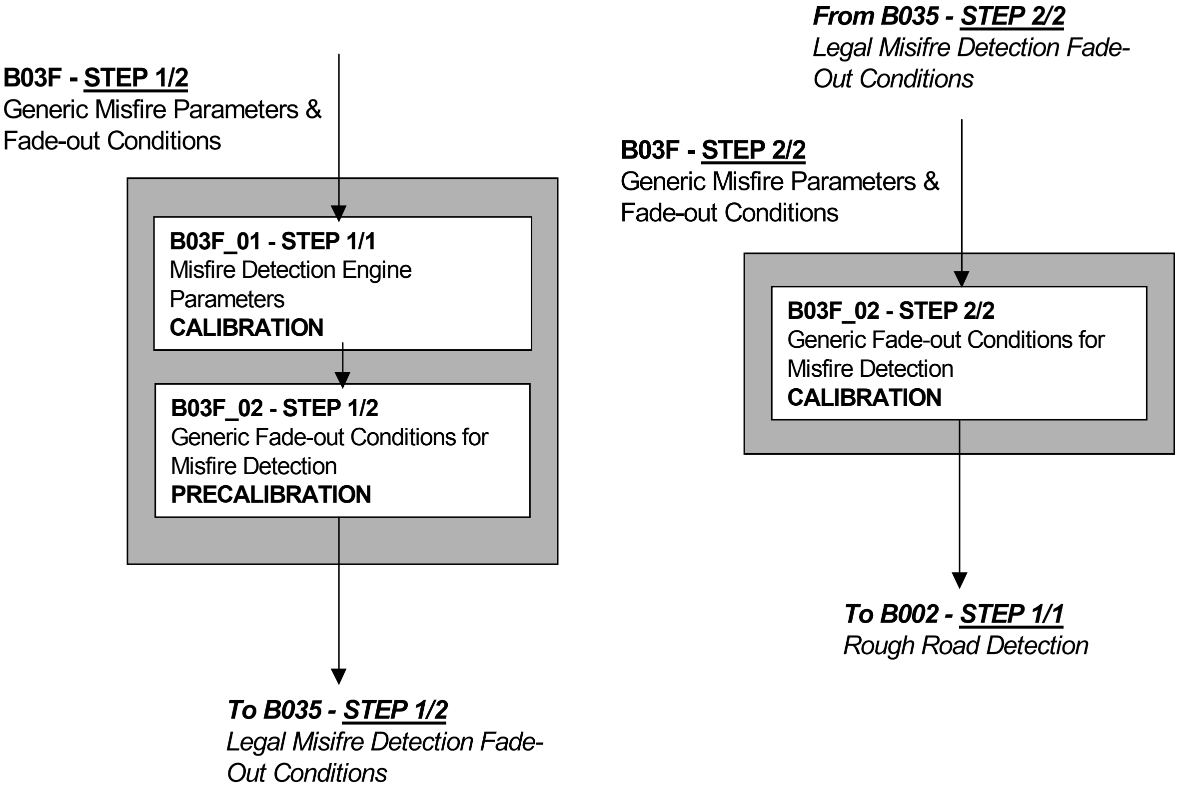
Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_USE_MIS_GEN	1	0...1H	0...1	1	[-]
Misfire software generator integration					

51.12 Misfire fade out (gen)

51.12.1 Tuning hints for Generic Misfire Parameters & Fade-out Conditions (Module B03F)

51.12.1.1 Module calibration flowchart



.

51.12.1.2 B03F_01 : Misfire detection engine parameters

51.12.1.2.1 General Information

In this chapter, LOAD_MIS is defined as a "transparent" engine load independently of the EMS structure type (based on MAP, MAF, TQI or else...). This variable is strictly dedicated to the misfire monitoring process. It allows commuting from one reference to another depending on different conditions (engine combustion mode, engine states or others).

This engine load is essentially used for:

- Zero load line definition
- Engine load breakpoint definition for misfire detection thresholds, gradient detections, catalyst destruction threshold ...

Today, engine load LOAD_MIS is defined as a ratio between the current system variable used for system control (see application system definition) and its maximum value.

MPI in MAP structure: LOAD_MIS = MAP / C_MAP_MAX_MIS

MPI in MAF structure: LOAD_MIS = MAF / IP_MAF_MAX_MIS(N_32)

MPI in Torque structure: LOAD_MIS = TQI_AV / IP_TQI_AV_MAX_MIS(N_32)

For MAF and TQI_AV, the maximum reached values partly depend on engine speed. The normalisations according IP_MAF_MAX_MIS & IP_TQI_AV_MAX_MIS allow reaching 100% of the engine load independently of the engine speed.

Hint: LOAD_MIS can be based on TQI_AV (actual indicated torque value). But with torque structure, TQI_AV quick torque requests can induce noise on detection threshold determination; so on some applications it's better to use MAF or MAP and multiply LOAD_MIS by ignition efficiency to correct torque reductions *(SI choice, introduction of an application multiplicative correction planned from MISF_2.1.x version)*.

GDI structure: one or two engine load references can be used MAP/MAF & TQI_AV.

MAP or MAF system variables can be used for homogeneous combustion modes (AFS and AFL) and TQI_AV or MFF for stratified combustion (engine torque mainly proportional to fuel mass).

TQI_AV can also be used for homogenous and stratified combustion modes (but take care on noise ratio).

Remark: MAF information has a slower response regarding to MAP for dynamic behaviour of the strategy *(could depend on applications)*.

51.12.1.2.2 Module calibration prerequisites

Airpath calibrated

Legal target known (for zero load line definition)

Engine speed redline known, engine capable to reach this engine speed (for US applications)

51.12.1.2.3 Calibration Step B03F_01 – Step 1 / 1

51.12.1.2.3.1 Calibration of engine load reference LOAD_MIS

C_MAP_MAX_MIS is often set to MAP maximum value obtain on your application (LOAD_MIS = 100%), but can be minimised next to have the best full-scale ratio for mean weather/altitude conditions (~1014hPa). No correction for atmospheric pressure is included due to the large ER signal noise ratio at high engine load.

IP_MAF_MAX_MIS is often set to MAF maximum value observed on the application *(see air path calibration step and intake architecture)* for each engine speed, but can also be minimised next to have the best full scale ratio for mean intake conditions.

IP_TQI_AV_MAX_MIS, is an image of the maximum torque setpoint reached in the engine torque reference map for each engine speed breakpoint.

According maximum MAF and/or TQI_AV characterization vs. engine speed that can be reached by our engine, define these calibrations in a way to reach LOAD_MIS = 100% for each engine speed when in wide open throttle.

51.12.1.2.3.2 Calibration of the engine zero load line

LOAD_MIN_MIS, the engine zero load line is based on a basic shape (zero torque line plus legal inhibition triangle defined according legal target), a coolant temperature correction and an atmospheric pressure correction.

51.12.1.2.3.2.1 Basic zero load line definition

Basic zero load line are determinated according, the transmission type (MT, AT) & the combustion mode (AFS, AFL or S)

Conditions: hot engines, low friction, no consumers, low altitude

Determination:

First with hot engine, drive the engine forced in desired combustion mode; with no gear engaged for manual transmission or in neutral for automatic transmission in all speed points (idle speed, 1000 rpm to max. engine speed for OBDII applications by step of 500 rpm) and note the load (LOAD_MIS[%], and/or MAP and TQI_AV engine parameter).

Test has to be evaluated on several engines/transmissions to take in account engine/car fluctuations.

Calibrate the values of IP_LOAD_MIN_MIS_xx_yy 5% below these values in the speed area up to 3000 rpm in a way to be able to detect misfire in idle speed & close to the zero load line.

This basic calibration will be firstly evaluated with constant TCO value, as a consequence IP_LOAD_MIN_MIS_xx_yy will be defined as an unique row.

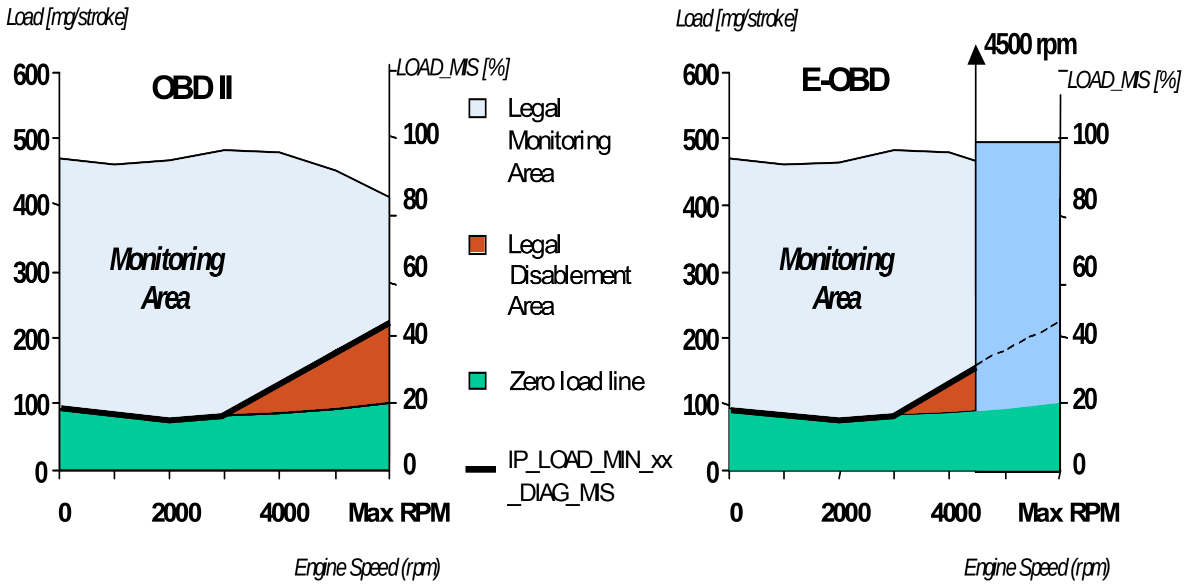
51.12.1.2.3.2.2 OBD II triangular offset

In the CARB mail-out #96-34, the disablement region from 3000 rpm to 6000 rpm is described with a line from 3000 rpm to 6000 rpm and the load increased with 4 inch Hg at 6000 rpm (i.e. ≈ 135 hPa). In this region, an offset has to be added. For that, a correspondence between 135 hPa offset and a LOAD_MIS offset has to be evaluated.

This offset can be applied by the calibrations IP_LOAD_MIN_OFS_AMP_MIS_xx_yy, evaluated at sea level.

51.12.1.2.3.2.3 E-OBD triangular offset

TThe disablement region from 3000 rpm to the max. speed (i.e.: minimum between 4500 or 1000 greater the highest speed occurring in ECE + EUDC cycle) is described with a line from 3000 rpm to the max. speed and the load increased with 133 hPa at this maximum speed. For that, a correspondence between 135 hPa offset and a LOAD_MIS offset has to be evaluated (if LOAD_MIS not based on MAP).



As for US application, the EC offset can be applied by the calibrations IP_LOAD_MIN_OFS_AMP_MIS_xx_yy, evaluated at sea level.

Remark: Today for GDI engines who are able to be in stratified combustion mode above 3000 rpm, no specific E-OBD law articles are related to GDI for equivalencies between 135hPa offset and a value in torque or mass fuel. A triangular offset between 3000 and 4500-rpm can be applied with a final value at 4500 rpm included between 15 and 20% of the maximum torque that this engine can provide in stratified combustion mode.

51.12.1.2.3.2.4 Engine coolant temperature correction

TCO correction for cold engine start.

Evaluate zero load line influence with cold starts and engine warm-up (global offset versus TCO) for each combustion mode used, this offset correction will be relative to engine friction effects versus engine coolant temperature.

Note: very often AFL and S combustion modes are not used under cold engine conditions. Moreover is really hard to evaluate the zero load line high engine speed and TCO low.

This TCO variations allow to complete the determination of the IP_LOAD_MIN_MIS_xx_yy calibrations.

51.12.1.2.3.2.5 Atmospheric pressure correction

AMP correction for zero load line altitude compensation (to be executed in altitude mission).

Evaluate zero load line variation versus different altitudes and different combustion mode. This correction can be applied in a relative way to the basic triangular offset formely defined, it completes the definition of IP_LOAD_MIN_OFS_AMP_MIS_xx_yy calibration .

51.12.2 Tuning hints for generic fade out conditions for misfire detection (B03F_02)

51.12.2.1 General Information

Generic fade out flag (LV_REQ_APP_INH_MIS) is based on all conditions that could create unreliable conditions for misfire detection :

- Engine gradients (engine load, engine speed, throttle, combustion mode & ignition angle transients).
- Crankshaft jolt effects that can disturb detection process (rough road, drivetrain oscillations, and air conditioner & accessories activation/deactivation).
- OBDI errors on one of the sensors used by the misfire monitoring process.

All these different conditions are merge into a carrier (MIS_DET_CDN_APP_INH).

51.12.2.2 Calibration advice for US applications

For OBD-II, take care that on a complete reference cycle this fade-out occurs to often, in the course of time the CARB reduce more and more the possibilities to use fade out based on IGA, TPS, MAF transients. In any case see with the executive officer the possibility to keep this functionality if you can provide an engineering evaluation to prove that you aren't able to detect misfire in such case in reliable conditions.

51.12.2.3 Module calibration prerequisites

Good vehicle driveability (at least 80% of the driveability calibration reached)

51.12.2.4 Calibration Step B03F_02 - Step 1 / 2

<u>Module precalibration</u>

A short way to disable all the generic fade-out described in this chapter is to set carrier masks to zero (all fade-out disabled).

C_MASK_APP_INH_DET_ZDLY_MIS = 0

C_MASK_APP_INH_DET_MIS = 0

C_MASK_APP_INH_DET_MIS_IS = 0

Thus MIS_DET_CDN_APP_INH_NR & LV_REQ_APP_INH_MIS will be always set to 0.

51.12.2.5 Calibration Step B03F_02 - Step 2 / 2

51.12.2.5.1 Configurable generic fade-outs management

A maskable fade-out management has been introduced to answer to the incertitude of the final 2002 CARB mail-out contents (always a draft up to now), especially for the fade-out management during and out of the "zero delay" phase. Moreover, as quite all fade-out must be discussed and justified to the executive officer, depending on results some fade-out could be cancelled in last minutes.

This maskable fade-out management can be very useful during calibration steps to remove non-tuneable fade-outs like OBDI errors and/or external function requests.

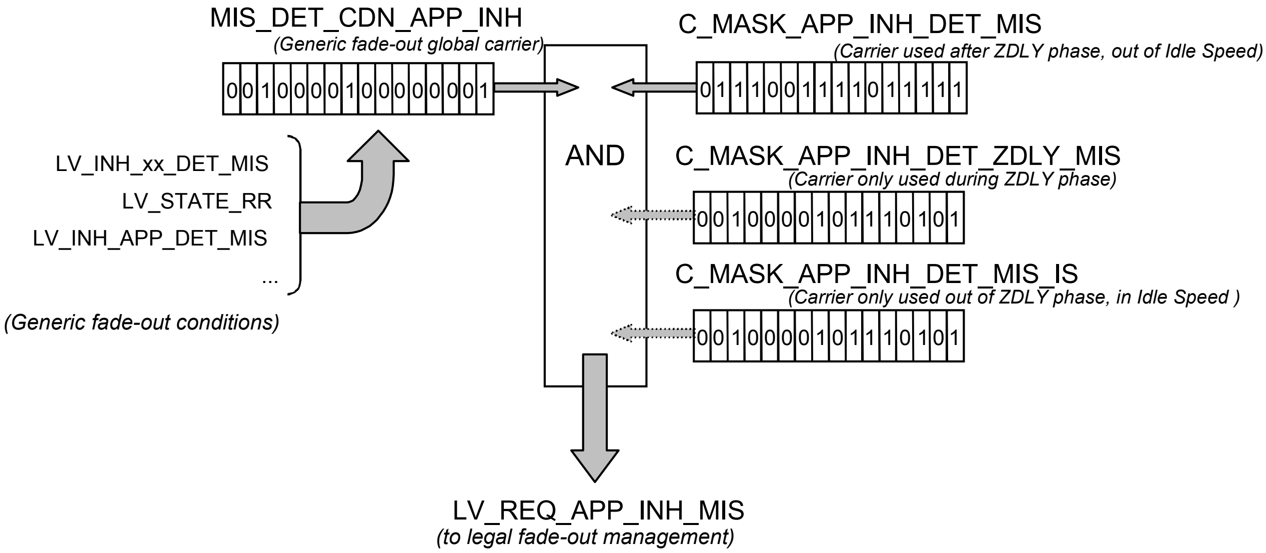
The maskable fade-out management can totally configure the fade-outs required or not for the legal target you require.

All generic fade-out conditions (LV_INH_xxx flags) are merged within a carrier structure MIS_DET_CDN_APP_INH (see definition below).

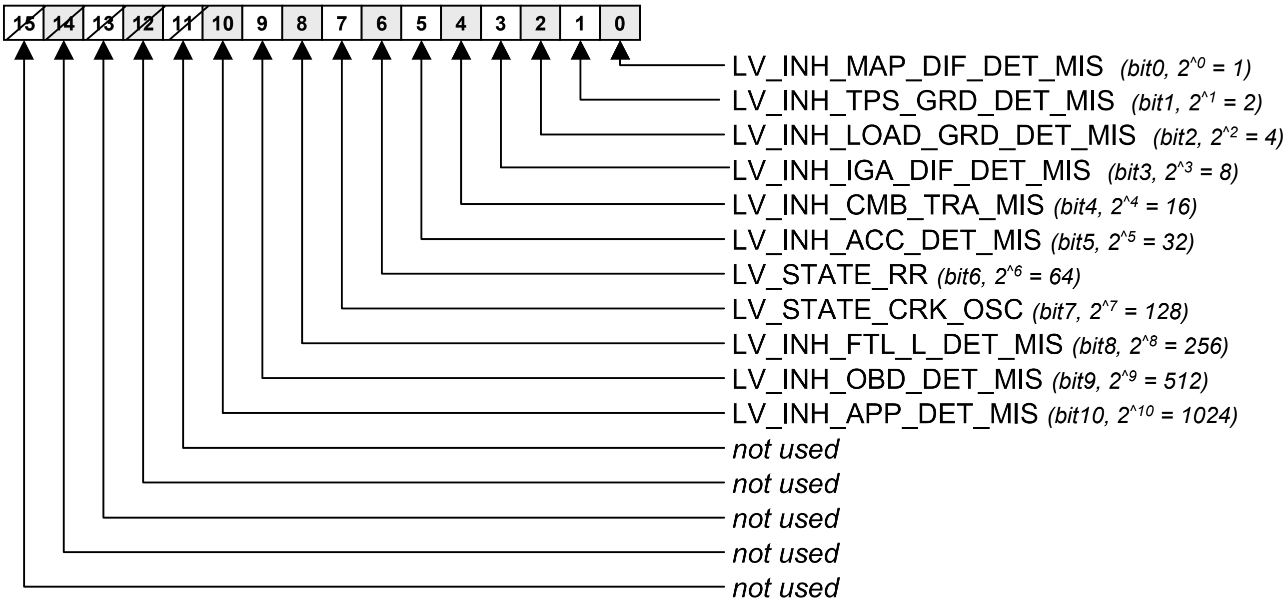
Three different mask calibrations (C_MASK_APP_INH_DET_ZDLY_MIS during & C_MASK_APP_INH_DET_MIS / C_MASK_APP_INH_DET_MIS_IS out of ZDLY period) are applied on this carrier to evaluate the number of fade-out flags set to 1 and validated by the masking.

Example: C_MASK_APP_INH_DET_ZDLY_MIS = 0x00 will disable all generic fade-outs during the ZDLY & C_MASK_APP_INH_DET_MIS = 0x7FF will enable all generic fade-outs out of the ZDLY period in part load & C_MASK_APP_INH_DET_MIS_IS = 0x7FF will enable all generic fade-outs out of the ZDLY period in idle speed.

Note: Bits 11 to 15 of calibrations C_MASK_APP_INH_DET_ZDLY_MIS, C_MASK_APP_INH_DET_MIS & C_MASK_APP_INH_DET_MIS_IS have no effects.



MIS_DET_CDN_APP_INH Carrier Structure



51.12.2.5.2 Maximum manifold air-pressure gradient

Due to trailing throttle / acceleration transition problems (especially for engines with dual mass flywheels), it could be necessary to disable misfire detection for a short period when the manifold air-pressure gradient exceeds an applicable value.

To evaluate the fade-out necessary in such conditions, please check at the transitions from fuel cut off to part load and vice versa as well as all transitions from low load to high load, if some speed fluctuations are critical for misdetection of misfire.

Measure in parallel the different engine load like informations available (TPS, MAP, MAF, TQI_AV & LOAD_MIS) who will be necessary for others fade-out calibrations *(LV_INH_LOAD_GRD_DET_MIS, LV_INH_TPS_GRD_DET_MIS)*.

For LV_INH_MAP_DIF_DET_MIS determination, look at which load difference the engine speed fluctuation is sufficiently forced to increase the risk of wrong misfire detection *(to be evaluated regarding ER, THD_ER & N_GRD values, use of ER_THD_RATIO would be appreciated if integrated, see misfire detection App. Inc. file)*.

Test Profiles:

- At low engine load (close to the zero load line LOAD_MIN_MIS): small values for the MAP difference could be necessary to fade out the detection. The risk of wrong detection is high in such conditions due to low Signal Noise Ratio on ER index.
- At high load, very big values can be calibrated, as there is less danger for such crankshaft fluctuations (larger ER signal noise ratio).

Evaluate in such conditions MAP_DIF_MIS versus MAP to calibrate the IP_MAP_DIF_MAX_MIS calibration map.

C_T_MAP_DIF_DLY_MIS must be set from 1.5 to 2 times the longest disturbance on ER index due to MAP_DIF_MIS you've observed during these tests.

Note for US applications

Not necessary and/or unjustified fade out at normal load conditions should be avoided according CARB requirements for US applications *(no fade-out based on TPS, Loads, MAP gradients should occur on FTP cycles).*

Fade out allowed for US: *" ... engine speed, load, or torque transients due to throttle movements more rapid than occurs over the US06 cycle for the worst case vehicle within each test group...".*

Note for stratified combustion: as in stratified combustion, the MAP information isn't relative to the engine load, LV_INH_MAP_DIF_MIS is always set to 0 in such mode. The engine load gradient in stratified is managed by LOAD_MIS gradient that is relative to torque information is this case.

51.12.2.6 Maximum throttle gradient

Due to trailing throttle / acceleration transient problems at low load, it is necessary to disable misfire detection for a short period when the throttle gradient exceeds an applicable value

Re-use the files obtained for LV_INH_MAP_DIF_DET_MIS calibration and look at which throttle gradient the engine speed fluctuation is sufficiently forced to increase the risk of wrong misfire detection *(to be evaluated regarding ER, THD_ER & N_GRD values)*.

The fade out through throttle gradient normally works at the transition from part load to fuel-cut-off. The air mass gradient is in this case too slow (LOAD_MIS based on MAF) and TPS_GRD is more accurate than MAP in such conditions.

Remark: The most critical point is a short time before undershoot the zero load line.

Evaluate in such conditions TPS_GRD versus TPS_AV to calibrate the IP_TPS_GRD_MAX_MIS calibration map.

C_T_TPS_GRD_DLY_MIS must be set from 1.5 to 2 times the longest disturbance on ER index due to TPS_GRD you've observed during these same tests.

Note for US: after discussion with our internal OBD Program Coordinator, it has been decided to introduce a specific TPS_GRD based fade-out during Zero Delay phase. This fade-out is dedicated to engine cranking misfire monitoring. IP_TPS_GRD_ZDLY_MIS must be greater than IP_TPS_GRD_MAX_MIS in a way to not trigger fade-out during nominal engine cranking, except when TPS is highly driver induced.

C_T_TPS_GRD_DLY_ZDLY_MIS must be set from 1.5 to 2 times the longest disturbance on ER index due to TPS_GRD you've observed during the 'Zero Delay Period' *(5s after engine start)*.

51.12.2.7 Maximum engine load gradient

Misfire detection can be disabled when the amount of engine load gradient exceeds a calibration value, this value is switched according the engine combustion mode. Fade out delay is also specific to combustion mode.

Re-use the former files obtained for LV_INH_MAP_DIF_DET_MIS calibration and look at which LOAD_GRD_MIS the engine speed fluctuation is sufficiently forced to increase the risk of wrong misfire detection *(to be evaluated regarding ER, THD_ER & N_GRD values, use of ER_THD_RATIO would be appreciated if integrated)*.

Note: If LOAD_MIS is based on MAP value, the fade-out flags LV_INH_LOAD_GRD_DET_MIS & LV_INH_MAP_DIF_DET_MIS will be equivalent if calibrated in the same way.

LV_INH_LOAD_GRD_DET_MIS is better used when LOAD_MIS defined according TQI_AV and MAF system variables *(C_CONF_LOAD_MIS = 0 or 1)*.

Test Profiles are equivalent to those used for LV_INH_MAP_DIF_DET_MIS

Evaluate in such conditions LOAD_GRD_MIS versus LOAD_MIS to calibrate the IP_LOAD_GRD_MIS calibration map.

C_T_LOAD_GRD_HOM_DLY_MIS must be set from 1.5 to 2 times the longest disturbance on ER index due to LOAD_GRD_MIS you've observed during these tests *(also AFL if necessary).*

Note for GDI Applications:

In stratified combustion mode, LV_INH_LOAD_GRD_DET_MIS is the main engine load gradient fade-out. An accurate calibration is required in such mode especially in idle speed and at low engine load (close to stratified LOAD_MIN_MIS values).

Test profiles required for calibration in stratified are mainly low engine and part-load up to the highest engine speed where stratified combustion is active.

In the same way C_T_LOAD_GRD_S_DLY_MIS must be set from 1.5 to 2 times the longest disturbance on ER index due to LOAD_GRD_MIS you've observed during stratified combustion tests.

51.12.2.8 Ignition retardation without change limitation

Various functions and/or CAN requests (e.g.. gearshift signal, traction control, torque structure reduction. etc...) could cause an ignition angle retardation.

Misfire detection can be disabled when the amount of the ignition timing retardation changes is too important OR/AND when the global ignition timing retardation is too important.

The fade-out conditions are split in 2 modes:

- 2 calibrations maps focus on the IGA transients (cylinder to cylinder)

  IP_DELTA_IGA_IS_MIS is used to detect unreliable conditions for detection according a certain amount of ignition angle transients. only used in idle speed.

  IP_DELTA_IGA_MIS is used to detect unreliable conditions for detection according a certain amount of ignition angle transients. only used in part load.

- 1 calibration map focus on the difference between the basic ignition angle and the current one.

  IP_DELTA_IGA_BAS_MIS is used to detect unreliable conditions for detection due to an important ignition angle retardation (rough combustions).

The experience has often shown that an advance reduction of 10°Crk could cause a torque fluctuation that can be critical for the misfire detection in lowest engine load conditions (lowest signal noise ratio on ER index).

Critical points : focus on low engine load with strong IGA corrections, idle speed & any torque reduction request case

Therefore, check at low and medium load and speed conditions at such advance (torque) reductions, if any problem occurs with misfire detection (ER decreases) and how long the disturbance last.

If there is any problem, calibrate the table to fade out for the duration of the disturbance; otherwise calibrate 20°Crk and a minimum duration of 10 tdc's.

Note for stratified combustion: as in stratified combustion, IGA corrections (if they exist) have minimum influence on engine load, LV_INH_IGA_DIF_DET_MIS is always set to 0 in such mode. The torque reduction in stratified is managed by LOAD_MIS gradient that is directly relative to torque information is such case.

51.12.2.9 Combustion mode transients

While the combustion manager is in transient state (change from a torque reference to another), the misfire detection must be disabled to have a proper base for the evaluation of the ER-misfire detection thresholds, which are defined for the stationary modes (AFS, AFL & S).

When combustion mode transients occur, during a certain number of Tdc's due to ER delay, LOAD_MIS and threshold maps switch, the misfire detection could be disturbed.

The experience has shown that a fade out of 8 tdc's for C_NR_TDC_CMB_TRA_S_MIS and C_NR_TDC_CMB_TRA_AFL_MIS is a minimum number. To be evaluated on each application according the driveability calibration target applied for the whole combustion structure.

51.12.2.10 Air - conditioning compressor activation

When the air - conditioning compressor is switched on, an additional load is briefly applied to the engine. This load jump can cause a segment period jump and crankshaft vibration, depending on the engine operating state and load request.

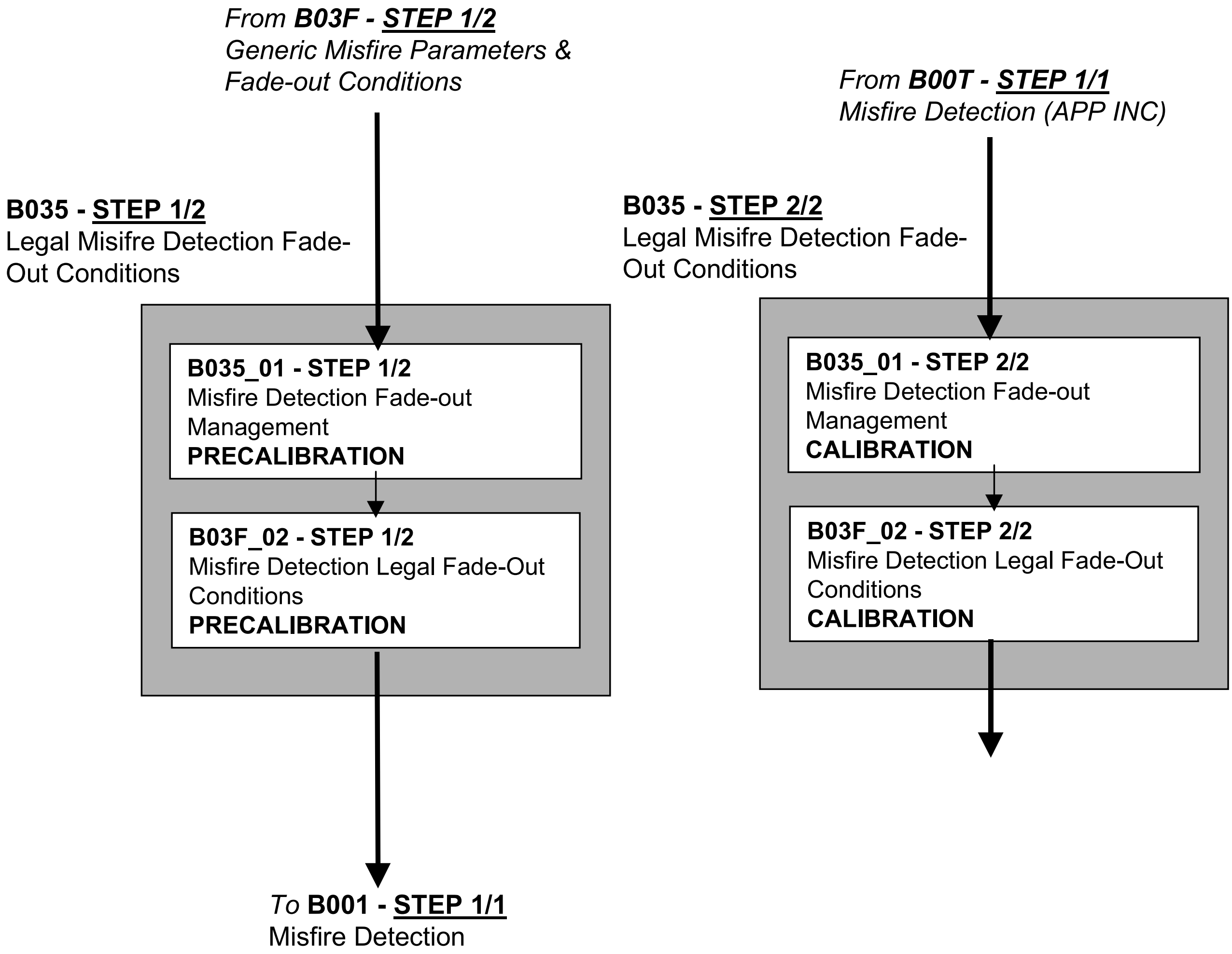
For C_T_ACCOUT_DLY_MIS calibration, check at different engine speed and load points, if the engine roughness ER could produce wrong misfire detection at the turn-on and off of the air conditioning compressor and the duration of the disturbance that could occur.

Calibrate this delay 2 times the longest disturbance you've observed.

Note for US application: This fade-out is no longer allowed for US

51.12.3 Tuning hints for Legal Misfire Detection Fade-out Conditions (Module B035)

51.12.3.1 Module calibration flowchart



51.12.3.2 B035_1 : Misfire detection fade out management

51.12.3.2.1 General information

For US applications, CARB requests an activation of the misfire monitoring and diagnosis as off engine start. Particular fade-out conditions are required during the first 5 seconds since engine running.

For non-CARB applications, this special monitoring is not required and has to be de-activated *(LC_REQ_ZDLY_MIS = 0).*

A state machine has been introduced for the activation of the misfire detection algorithms, distinguishing the separate monitoring during the first 5 seconds, called 'Zero Delay Phase' and the 'Nominal' monitoring afterwards, and preparatory phases.

During the monitoring phases, the nominal fade-out mechanisms are used, corrected for some fade-outs that could be not allowed during the first 5 seconds *(to be confirmed with official mail-out)*.

"... (3.3) Monitoring Conditions:
(3.3.1) Manufacturers shall continuously monitor for misfire under the following conditions:
(A) From no later <u>than the end of the second crankshaft revolution</u> after engine start,
(B) During the rise time and settling time for engine speed to reach the desired idle engine speed at engine start-up (i.e., "flare-up" and "flare-down"), ... "

51.12.3.2.2 B035_1, Step 1/2, Example of calibration

LC_REQ_ZDLY_MIS = 0, *// no specific Zero Delay Phase management*

C_N_MIN_ZDLY_MIS = 0

C_T_ZDLY_MIS = 0

C_NR_TDC_ZDLY_MIS = 0

51.12.3.2.3 B035_1, Step 2/2, Calibration

For Non US applications, no specific Zero Delay Phase management is required up to now. Please contact your E-OBD coordinator to confirm this information.

In such case, LV_ZDLY_DIAG_MIS, CTR_TDC_ZDLY_MIS & CTR_T_ZDLY_MIS will be set always set to 0.

LC_REQ_ZDLY_MIS = 0

C_N_MIN_ZDLY_MIS = 0, minimum engine speed will be managed in module B035_2.

C_T_ZDLY_MIS = 0, fade-out during engine start period will be managed in module B035_2.

C_NR_TDC_ZDLY_MIS = 0

For US applications,

CTR_T_ZDLY_MIS, the zero delay period is fixed by law to 5s (to be confirmed by OBD coordinator) after legal definition of engine start

C_NR_TDC_ZDLY_MIS = NC_CYL_NR tdc's, it's maximum value corresponding to "...*no later than the end of the second crankshaft revolution after engine start...*"

C_N_MIN_ZDLY_MIS corresponds to the engine speed that is the legal definition of engine start : "... “Engine start” is defined as the point when the engine reaches a speed 150 rpm below the normal, warmed-up idle speed (as determined in the drive position for vehicles equipped with an automatic transmission)..."

51.12.3.3 B035_2 : Misfire detection legal fade out conditions

51.12.3.3.1 General information:

Legal fade out (LV_REQ_INH_MIS) is based on all fade out conditions allowed by the legal texts on going and known for continuous misfire monitoring.

All these different fade-out conditions are merge into a global carrier (MIS_DET_CDN_INH).

51.12.3.3.2 B035_2, Step 1/2, example of calibration

Module precalibration

A short way to disable all the legal fade-out described in this chapter is to set carrier masks to zero (all fade-out disabled).

C_MASK_INH_DET_ZDLY_MIS = 0

C_MASK_INH_DET_MIS = 0

Thus MIS_DET_CDN_INH_NR & LV_REQ_INH_MIS will be always set to 0.

51.12.3.3.3 B035_2, Step 2/2, Calibration

51.12.3.3.3.1 Configurable legal fade-outs management

In the same way that for generic fade-outs, a maskable fade-out management has been introduced to answer to the incertitude of the final 2002 CARB mail-out contents (always a draft up to now), especially for the fade-out management during and out of the "zero delay" phase.

Moreover this maskable fade-out management can be very useful during calibration steps to quickly enable the misfire detection without detune other legal fade-outs.

The maskable fade-out management can totally configure the fade-outs required or not for the legal target you require.

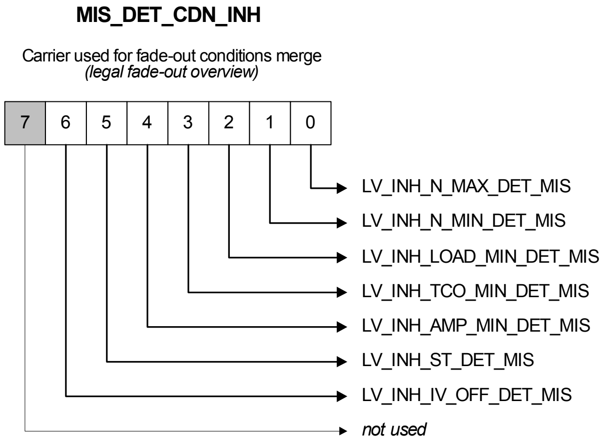
All generic fade-out conditions (LV_INH_xxx flags) are merged within a carrier structure MIS_DET_CDN_INH (see definition below).

Two different mask calibrations (C_MASK_INH_DET_ZDLY_MIS during & C_MASK_INH_DET_MIS out of ZDLY period) are applied on this carrier to evaluate the number of fade-out flags set to 1 <u>and</u> validated by the masking.

<u>Example:</u> C_MASK_INH_DET_ZDLY_MIS = 0x00 will disable all legal fade-outs during the ZDLY & C_MASK_INH_DET_MIS = 0x7F will enable all legal fade-outs out of the ZDLY period.

<u>Note:</u> Bit 7 of calibrations C_MASK_INH_DET_ZDLY_MIS & C_MASK_INH_DET_MIS has no effects on carrier.

MIS_DET_CDN_INH : Carrier used for fade-out conditions merge *(legal fade-out overview)*



For US applications, its recommended to set to 0 the LV_INH_ST_DET_MIS corresponding bit within C_MASK_INH_DET_MIS and C_MASK_INH_DET_ZDLY_MIS mask calibration to avoid any fade-out during engine start *(who is forbidden for US applications)*.

51.12.3.3.3.2 Engine speed range fade-out

The misfire monitoring can be inhibited when the engine speed reaches the maximum engine speed imposed by the market target : Redline engine speed for US applications or 4500 rpm for EC applications.

Moreover this information is used to stop the misfire monitoring process for high speed if not required (limitation of CPU load especially for European application).

The misfire monitoring can be inhibited when the engine speed is below the engine speed corresponding to engine start speed.

<u>for US applications:</u>

C_N_MIN_MIS must below C_N_MIN_ZDLY_MIS, set to the lowest engine speed where the engine roughness index is still declared valid (LV_ENA_ER = 1)

C_N_MAX_MIS_xT = Engine speed red line, fuel cut-off limiter

"... “Redline engine speed” shall be defined by the manufacturer as either the recommended maximum engine speed as normally displayed on instrument panel tachometers or the engine speed at which fuel shutoff occurs... "

for non US applications:

C_N_MIN_MIS, same calibration than C_N_MIN_ZDLY_MIS for US is recommended *(hot engine idle speed - 150rpm)*

C_N_MAX_MIS is 4500 rpm, can be higher if customer still want to diagnosis catalyst overheating criterion. In such case be sure that your input hardware stage is conform to execute such functionality (crankshaft input signal sensitivity & filtering).

51.12.3.3.3.3 Minimum engine load (zero load line)

Below a specific load (zero torque line), fluctuations in engine speed cannot be detected, even in the event of misfire. In this case, the misfire detection is disabled.

Difference between US and Non-US applications stands in the LOAD_MIN_MIS zero load line definition. See the dedicated LOAD_MIN_MIS calibration chapter.

LOAD_MIN_MIS is compared to actual engine load LOAD_MIS, both are determinated in module 'Misfire detection engine parameters'.

A fade-out is applied as soon as actual engine load is below the zero load line (out of legal monitoring area).

51.12.3.3.3.4 Minimum coolant temperature

Misfire detection can be disabled when the coolant temperature is below a threshold (20°F / -7°C). Additionally when the coolant temperature at engine start is below this threshold, the misfire monitoring can be disabled on that driving cycle until the current engine temperature reaches a warm up temperature threshold (70°F / 21°C):

C_TCO_MIN_MIS : 20°F (-6.75°C) under executive officer approval for OBD-II

C_TCO_WUP_MIN_MIS : 70°F (21°C)

Equivalent calibrations can be applied to non US calibrations.

51.12.3.3.3.5 Minimum atmospheric pressure

Misfire detection can be disabled when the atmospheric pressure is below a threshold:

C_AMP_MIN_MIS : 8000 feet above sea level under executive officer approval for OBD-II (see average correspondence to atmospheric pressure), 2500 m for E-OBD.

Correspondence in atmospehric pressure : 750 hPa frequently used.

51.12.3.3.3.6 Inhibition delay after start

Non US applications

Check at cold start conditions, how long the engine needs to get a stabilised condition after leaving start. The expected value must be less than 5s. This value is the maximum value allowed by the E-OBD.

US applications

Today for US applications, such fade out are now **forbidden**. Misfire monitoring has to be active at least at legal engine start definition + one engine cycle.

To disable LV_INH_ST_DET_MIS at engine start, set corresponding bit to 0 in C_MASK_INH_DET_ZDLY_MIS carrier calibration.

51.12.3.3.3.7 Injection shut-off

Misfire detection can be suppressed for the cylinder(s), for which injection has been shut-off.

General cylinder shut-off (all cylinders off)

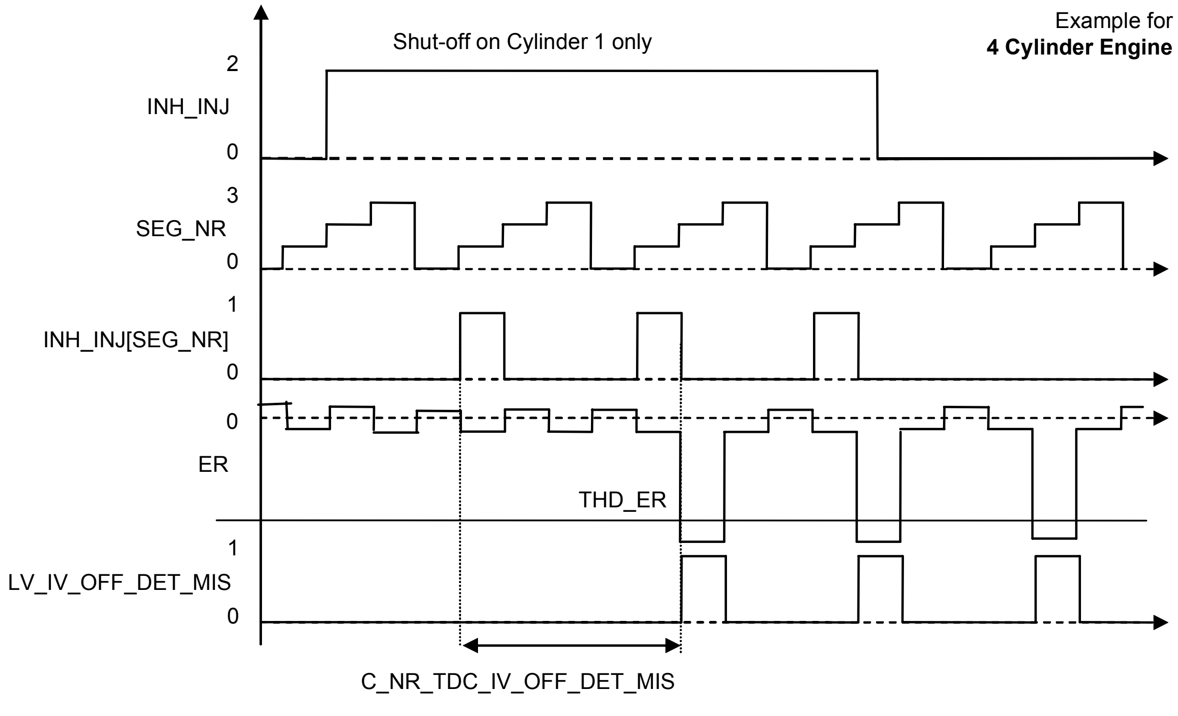
The flag LV_PUC_DET_MIS includes a calibration delay C_T_DLY_IV_OFF_MIS retriggered at each fuel cut-off condition falling edge, in a way to start the misfire detection only after crankshaft oscillations due to fuel reactivation.

To calibrate C_T_DLY_IV_OFF_MIS, execute several road tests with gear ratios allowing general fuel cut-off sequences. Observe at fuel reactivation instant, the disturbances on ENRD index that could generate misfire wrong detection (in positive torque). Set C_T_DLY_IV_OFF_MIS from 1.5 to 2 times the longest disturbances you've observed during these tests.

Cylinder specific shut-off:

If the current cylinder checked by misfire detection is shut-off by any EMS function (even MISF limp home), a fade-out can be apply for the misfire detection of this cylinder.

As it exists a delay between the instant where the cylinder is in shut-off and the backwash on ER index to detect the misfire, a tdc delay (C_NR_TDC_IV_OFF_DET_MIS) has been introduced on LV_IV_OFF_DET_MS flag to properly phase the fade-out information LV_IV_OFF_DET_MIS and the ER signal.



As example for a 4 cylinder engine, C_NR_TDC_IV_OFF_DET_MIS = 5 tdc's.

Check process:

To check quickly and simply this synchronisation, we will use the individual cylinder shut-off fade-out, check its action on ER, THD_ER, LV_INH_DET_MIS, LV_DET_THD_MIS & LV_DET_MIS.

Prerequisite: LV_INH_IV_OFF_DET_MIS must be taken in account in legal fade-out structure to be able to set the fade-out merge bit LV_REQ_INH_MIS. For that, please set the bit 6 to 1 (see MIS_DET_CDN_INH structure definition) in calibrations C_MASK_INH_DET_MIS & C_MASK_INH_DET_ZDLY_MIS (ex: 64).

The goal is to:

- Apply a single fuel shut-off on a cylinder (here via the misfire generator with injection), LV_MIS_GEN = 1.
- Observe LV_INH_IV_OFF_DET_MIS = 1, when this cylinder shut-off is effective.
- Observe next the ER value in misfire (no fuel), at this instant LV_INH_DET_MIS must be set to 1, LV_DET_THD_MIS must be set to 1 (if detection calibrated on this engine operating point) but LV_DET_MIS must stay to 0 due to this fade-out.

Limit number of cylinders shut-off:

If the number of shut-off cylinders reaches a calibration data (C_SUM_INH_IV_MAX_DET_MIS), reliable misfire detection could be impossible. In such case is preferable to inhibit misfire detection to avoid wrong misfire detections.

This number of cylinders in shut-off to disable the misfire detection depends on NC_CYL_NR, the number of engine cylinders.

The experience has shown that

C_SUM_INH_IV_MAX_MIS = 1 for NC_CYL_NR = 3

C_SUM_INH_IV_MAX_MIS = 2 for NC_CYL_NR = 4 and 5

C_SUM_INH_IV_MAX_MIS = 3 for NC_CYL_NR = 6 and 8

Limitation of this functionality: misfire detection could be unreliable also depending on the position of the cylinders in shut-off regarding to the crankshaft (mechanical balance). To evaluate that effect, during misfire diagnosis create MIS_A with different cylinder positions and numbers with C_SUM_INH_IV_MAX_MIS = NC_CYL_NR to see the detection efficiency & the number of wrong detections in such conditions. Apply finally the minimum number of cylinder in shut-off, to be sure of a reliable misfire detection in any case.

51.12.4 Tuning hints for Rough road detection (Module B002)

VHMD version (NC_CONF_RR_MIS = VHMD)

51.12.4.1 General information

Rough road conditions must be detected to prevent erroneous misfire detection that could be caused by jollity tracks influence on crankshaft via the drive train.

The speed disturbance of vehicle wheels is used to evaluate the rough road conditions.

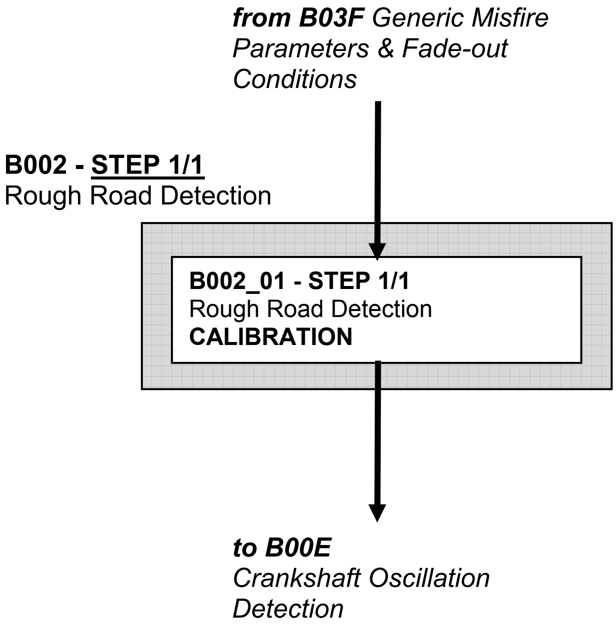
The EMS control unit can use two kinds of signal coming from wheel speed sensors:

WSS : EMS uses directly a wheel speed sensor as an input and acquires wheel segment times by its own timer.

CAN : EMS uses indirectly a wheel speed information coming from a CAN chassis module like ABS or TCS.

The VHMD (vehicle motion determination) aggregate manage in a transparent manner the configuration (one or several wheel speed sensors and TCS/ABS units communication through CAN...) and dataflow to provide a unique input to the rough road detection module (VS_DRV0_STD) plus a control flag (LV_VS_DRV0_STD_NOT_ENA) who inform that this information is valid or not.

51.12.4.2 Module calibration flowchart



51.12.4.3 B002_2: Rough road indexes determination

Filter coefficients for rough road index:

C_CRLC_VS_DRV0_STD_INC & C_CRLC_VS_DRV0_STD_DEC

Precals: C_CRLC_VS_DRV0_STD_INC = 0.3 & C_CRLC_VS_DRV0_STD_DEC = 0.1

Wheel gradient filtering constant

The filtering constant is responsible to avoid a rough road triggering at short disturbances on the one hand and on the other to ensure a detection fast enough this means before an oscillation is induced on the crankshaft.

The coefficient C_CRLC_VS_DRV0_STD_INC is used for rising rough road values, and C_CRLC_VS_DRV0_STD_DEC is for the decay of these rough road values. This differentiation allows to have a fast rise of the rough road values and a smooth decay.

The proposed constant C_CRLC_VS_DRV0_STD_DEC = 0.1 should be a suitable compromise between both demands.

If at the validation phase many erroneous rough road detection due to short disturbances are observed, a smaller filtering constant e.g. 0.05 should be tested *(stronger filtering to avoid high frequency noises).*

On the other hand, If rough road shocks and disturbances are not detected due to a too important filtering, this means a too slow reaction of the detection is observed, a bigger value e.g. 0.3 should be tested *(lighter filtering to take in account medium frequencies)*.

51.12.4.4 B002_3: Detection of rough road status

Precalibrations:

Following values have to be calibrated at the begin to enable the rough-road detection process.

- C_VS_MAX_RR: 120 km/h
- C_VS_MIN_RR: 4 km/h

Following measurements have to be made:

Sampling rate = 10ms

Basic data, for rough road calibration: VS, SUM_RR, VS_DRV0_STD and VS_DRV0_STD_MMV.

Data to evaluate rough road effects on misfire detection: ER *(or ER_STND)* & THD_ER.

Data to evaluate rough road fade-out sequencing toward misfire detection: LV_INH_DET_MIS, MIS_DET_CDN_APP_INH & MIS_DET_CDN_INH

Data to evaluate misfire wrong detection: LV_DET_MIS & CTR_TOT_DC_MIS

Process:

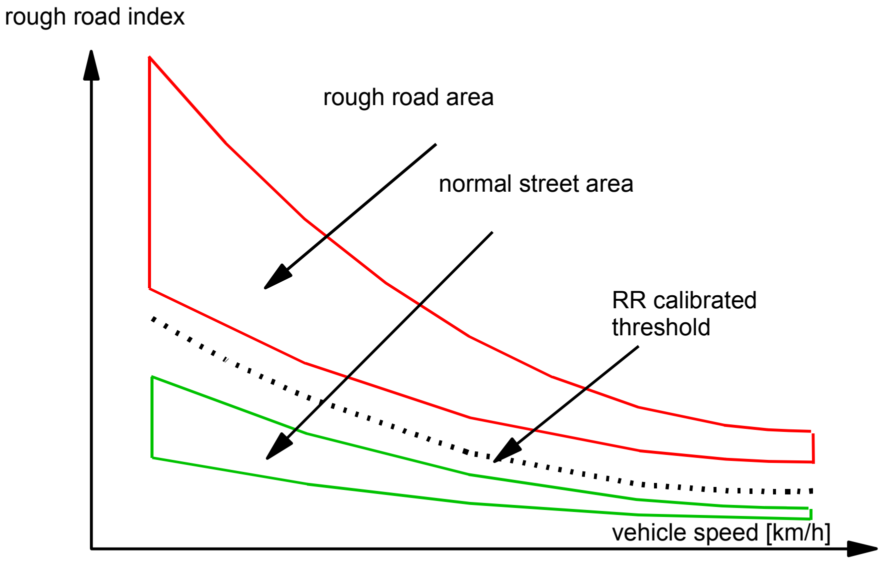
The vehicle has to be driven on normal roads (asphalt road, concrete street) to get the nominal noise (upper limit for threshold) at different vehicle speeds.

Therefore drive the vehicle with different gears and small acceleration from low to high engine rpm. A maximum velocity of 120 km/h should be sufficient.

Second step, same measurements have to be made on rough road (as described above) to get the amplitude under such conditions. If the rough road distance is not long enough, more measurements with constant speed are also possible. The thresholds IP_VS_DRV0_STD_MMV_THD_AT or IP_VS_DRV0_STD_MMV_THD_MT have to be calibrated in the middle between normal street and rough road (see following graphic).

(Remark: some restrictions for the vehicle speed due to driveability of the vehicle must be accepted)

Due to the high dynamic, the breakpoints should be calibrated nearer at low vehicle velocity in comparison to higher velocity.



Rough road fade out delay: C_T_DLY_RR

The fade out time for misfire detection **:** C_T_DLY_RR should be about two times longer than the worst case time between disappearance of rough road influence and the complete disappearance of the crankshaft oscillation on engine roughness index.

For the disappearance of the rough road influence, the non-filtered wheel speed gradient and the engine roughness index should be observed in parallel to the rough road value *(fastest signals)*.

51.12.4.5 B002_4: Vehicle speed range for rough road calculation & detection:

Minimum vehicle speed: C_VS_MIN_RR

The absolute minimum is the specified minimum vehicle speed from the wheel speed sensor. As at normal driving condition a speed below 5 km/h with gear engaged is impossible, this value is suggested.

Moreover, according new regulations for US legal target, it is recommended to adapt this value to avoid wrong rough road detections at vehicle start (high wheel speed gradient in such case). Take care that no rough road detections occur during US emission cycles.

Maximum vehicle speed: C_VS_MAX_RR

Depending on the drivetrain and car body design, crankshaft speed disturbances are only possible up to a certain vehicle speed. The upper critical speed should be known from the former tests for the misfire detection.

A maximum vehicle speed higher than 120 km/h should be avoided because of ECU runtime and high accuracy decreasing on wheel speed gradient.

51.12.4.6 B002_5: Limp-home state

According legal requirements / customers wishes, a limp-home calibration is available.

Recommended for US: LC_STATE_RR_LIH = 0.

In case of WSS error and or ABS/TPS module error and/or CAN channel error, the misfire monitoring must not be faded-out. LV_STATE_RR = LC_STATE_RR_LIH = 0.

For such case, you can apply a specific error who inform that one of the misfire criterions has been reached but with the rough road module in limp-home (LV_ERR_MIS_RR). Please consult for this point, the dedicated chapter in the tuning guide for misfire diagnosis part.

Up to now for CE, a fade-out of the misfire monitoring can be applied in case of rough road module error. Please contact our OBD expert to be informed of any change on this status.

51.12.4.7 Vehicle Validation

A separated vehicle validation plan has to be fixed based on the experience of the experts, (different wheel pressure, driving with trailer, ABS influence, traction control,...) and generally on customer "rough road" track profiles & requirements.

Focus on worst cases of rough road influence for the misfire monitoring:

- lowest signal/noise ratio on engine roughness index (close to zero load line & medium/high engine speed)
- washboard rough roads (medium frequencies) where there's a clear overlap between road oscillations and engine firing events.

Check misfire detection counter CTR_TOT_DC_MIS during rough road tests to evaluate wrong misfire detections and rough road fade-out efficiency.

51.12.4.8 Fleet Validation

A special driving program has to be performed to get a wide range of different environmental conditions for the fleet, especially that bad road conditions are included. Nevertheless it is useful to drive some cars without any program.

The counter SUM_RR can be used for determine non-typical behaviour of a vehicle

51.12.5 Tuning hints for Crankshaft oscillation detection (Module B00E)

Used only if NLC_USE_CRK_OSC_MIS = 1

51.12.5.1 General information

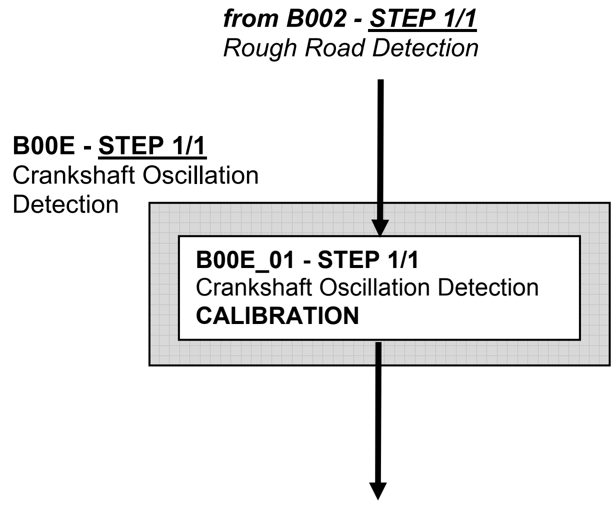
In case of a single misfire detected at low engine speed / high engine load, in combination with a high gear (e.g. 3rd, 4th, 5th gear), drivetrain & crankshaft oscillations can occurs, it would not be possible to ensure a proper misfire calculation due to oscillations amplitude. The trigger of such an oscillation may be a single misfire, an obstacle, a big torque change or other instantaneous conditions.

A fade-out can be applied according your legal target to disable misfire detection during these oscillations *(please consult your OBD specialist / customer to ensure the possibility to use such a functionality, if no please set NLC_USE_CRK_OSC_MIS = 0 on your application)*.

The crankshaft oscillation detection is based on two criterions during a fixed tdc's window after a detected misfire: detection of a critical frequency & detection of a critical amplitude.

If both parts fulfilled at the same time their own detection criterion, a crankshaft oscillation is detected. The reason therefore is that even at constant speed due to normal little speed variations a ER high frequency, or during a normal acceleration phase, a big ER amplitude are possible.

51.12.5.2 Module calibration flowchart



51.12.5.3 Determination of critical conditions for misfire detection

Basis for the calibration of the crankshaft oscillation is knowledge of the critical areas (speed, load, and gear) for the misfire detections.

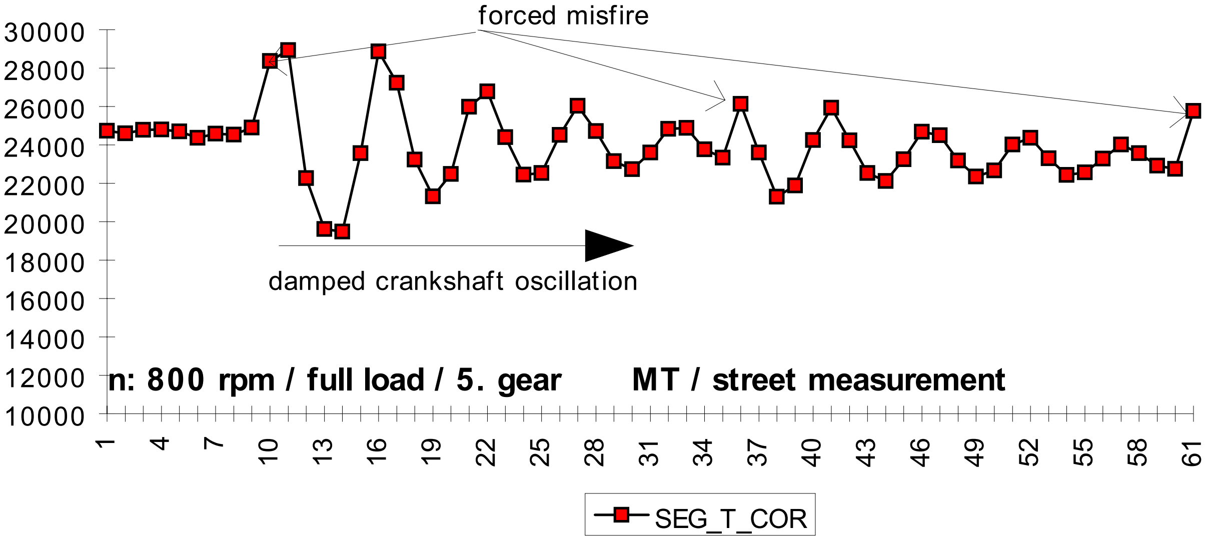
Therefore, tests on the street have to be carried out for the misfire detection in order to determine such conditions.

Engine Speed:	800 / 1000 / 1200 / 1400 / 1600 / 1800
Engine Load:	idle / 3 steps / full load
Gear:	1 / 2 / 3 / 4 / 5
Misfire Pattern:	without misfire / 1cyl. In misfiring / Misfire rate 1/33 (pseudo-random)

An analysis of the data must be made for the check, under which conditions false misfire is detected.

Observe ER indexes toward detection thresholds to evaluate the risk of wrong detection.

Example for critical oscillation after 1/33 misfire.



51.12.5.4 B00E_1: Maximum engine speed for Crankshaft oscillation detection

Set C_N_MAX_CRK_OSC to the maximum engine speed that you've observed oscillations phenomenon that could produce wrong misfire detections. *Typical: 2000 rpm*

51.12.5.5 B00E_2: Minimum VS / N ratio to activate crankshaft oscillation detection

Set C_RATIO_VS_N_MIN_CRK_OSC to the minimum engine speed / vehicle speed ratio where you've observed oscillations phenomenon that could produce wrong misfire detections. *Typical: corresponding to 3rd gear. The value C_RATIO_VS_N_MIN_CRK_OSC should be calibrated in the middle between the critical gear (e.g. 4th gear) and the gear below (3rd gear) to ensure the detection (e.g. 0.030).*

Maximum TDC after detected misfire to activate crankshaft oscillation detection process

The monitoring process to analyse crankshaft detection will be limited to a window of C_NR_TDC_MAX_CRK_OSC tdc's after LV_DET_CFM_MIS is set.

Set C_NR_TDC_MAX_CRK_OSC to the high number tdc's corresponding to the longer oscillations that you've observed induced by a generated misfire. *Depends mainly on drivetrain design.*

51.12.5.6 B00E_3: Determination of a critical engine roughness oscillation frequency

It allows retriggering the detection criterion for a defined number of tdc's after each DRV0_ER sign change. Equivalent to a restart of oscillation detection. In case of crankshaft oscillation, it must be checked how many subsequent DRV0_ER values are needed to distinguish clear from DRV0_ER single value at misfire and crankshaft oscillation.

Hint: IP_SEG_MIN_FRQ_CRK_OSC values in the complete engine speed range are = 2, this seems to be the best compromise (No critical frequency will be detected at 1cyl. misfiring). *To be checked on your application.*

51.12.5.7 B00E_4: Determination of a critical engine roughness oscillation amplitude

Each time DRV0_ER changes its sign, the process to evaluate the amplitude of the oscillation is retriggered. If this oscillation exceeds the amplitude DRV0_ER_SUM_THD_OSC defined versus N_32 and LOAD_MIS, thus a critical engine roughness oscillation amplitude is declared (*LV_AMPL_CRIT_CRK_OSC = 1).*

Remark: The threshold is linked to the value SEG_SUM_FRQ_CRK_OSC. In case of a 2 for calibration, the threshold must be calibrated for the sum of two subsequent DRV0_ER values after sign change of DRV0_ER. There must be a clear difference between one DRV0_ER value at misfire and two values at crankshaft oscillation.

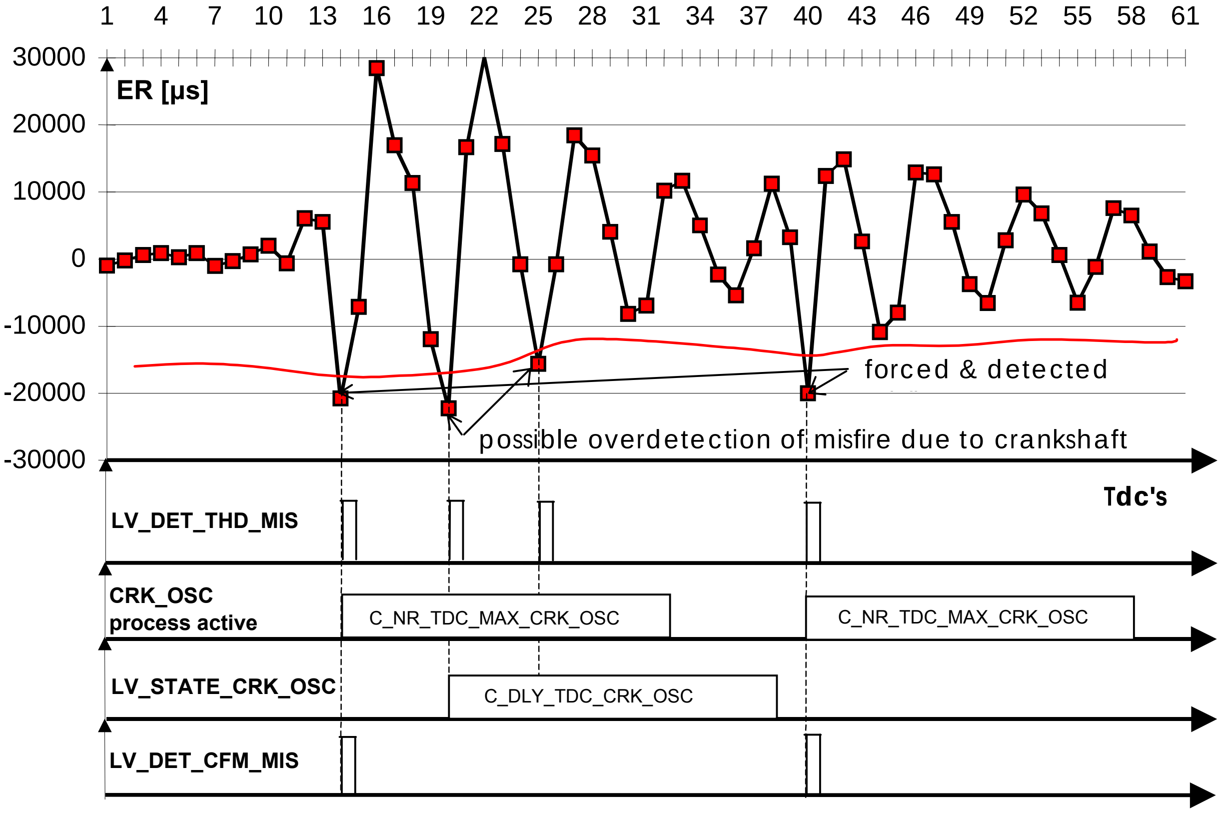
IP_DRV0_ER_THD_OSC(N_32, LOAD_MIS) must be lower than basic misfire detection thresholds in a way to avoid any wrong misfire detection.

51.12.5.8 B00E_5: Determination of crankshaft oscillation final status

A crankshaft oscillation is detected *(LV_STATE_CRK_OSC=1)* when in the same time, application conditions are active *(LV_CRK_OSC_DET_ACT=1),* a crankshaft oscillation has reach critical frequency *(LV_FRQ_CRIT_CRK_OSC=1)* and amplitude criterions (*LV_AMPL_CRIT_CRK_OSC=1).*

LV_STATE_CRK_OSC is hold to 1 for C_DLY_TDC_CRK_OSC tdc's. This delay must be set according the maximum impact time of the longer oscillation that could induce wrong misfire detection. *Practically C_DLY_TDC_CRK_OSC is very close to C_NR_TDC_MAX_CRK_OSC.*

Example:



1
4
7
10
13
16
19
22
25
28
31
34
37
40
43
46
49
52
55
58
61
30000
20000
10000
0
-10000
-20000
-30000
ER [µs]
forced & detected
possible overdetection of misfire due to crankshaft
Tdc's
LV_DET_THD_MIS
CRK_OSC
process active
C_NR_TDC_MAX_CRK_OSC
C_NR_TDC_MAX_CRK_OSC
LV_STATE_CRK_OSC
C_DLY_TDC_CRK_OSC
LV_DET_CFM_MIS

51.13 Misfire OBDII (gen)

51.13.1 Calibration hints for function Misfire Detection (module B001)

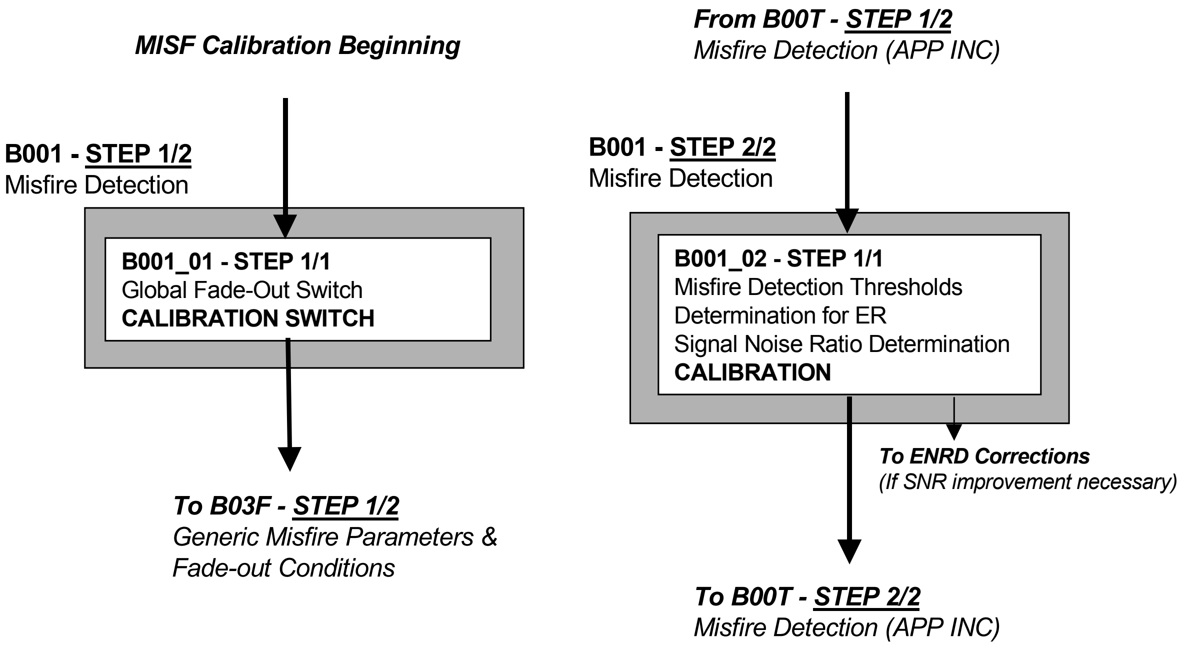
For NC_MISF_VERS = 1, the detection can be evaluated according two different indexes:

NLC_USE_ER_STND = 0 use ER nominal index

NLC_USE_ER_STND = 1 use ER normalized index (ER_STND)

Calibration methods for both configurations are strictly identical, only the data behavior & range are different.

51.13.1.1 Module calibration flowchart



51.13.1.2 B001_01: Misfire Monitoring Global Fade-Out Switch

Calibration data:

LC_MIS_INH	1	0 ... 1H	0 ... 1	1	-
Global switch to disable all misfire monitoring related modules (Inhibition with = 1)					

51.13.1.2.1 General information

There is one global switch that is used to inhibit quickly all the MISF aggregate modules

It's the highest priority fade-out for MISF aggregate.

51.13.1.2.2 Calibration Step B001_01 – Step 1 / 1

Before to start any MISF aggregate calibration, please check that LC_MIS_INH is correctly set to 0, to enable each aggregate modules call and activate each nominal process.

Otherwise, processes are totally frozen and data initialized.

51.13.1.3 B001_02: Misfire detection thresholds determination for engine roughness index

51.13.1.3.1 General information

ER misfire detection method is based on a comparison between an engine roughness index (ER or ER_STND) value and a threshold that is relative to the engine parameters when the monitored cylinder was in intake phase *(threshold relative to combustion conditions image, memory effect managed by a buffer)*.

The ER engine roughness value depends strongly on the current operating point, both in its spread during undisturbed engine operation, and in the signal amplitude in case of an actual misfire.

ER misfire detection thresholds are defined according basic threshold maps and threshold corrections:

- *A basic threshold map:* IP_THD_y_ER_x

Basic threshold maps are singular for Transmission Type, ER index type and combustion mode.

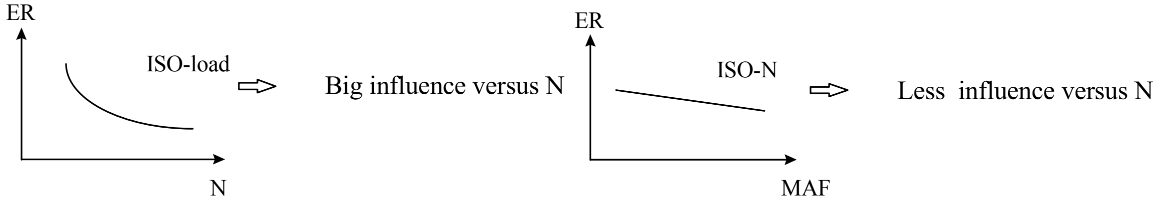
_x: stands for _MT or _AT

_y: stands for _AFS, _AFL or _S

Theses maps are function of engine load and speed, which are calibrated for engine at operating temperature as well as for, adapted transmitter gear.

IP_THD_y_ER_x maps are switched according the vehicle transmission type, idle speed condition & the combustion mode.

The nominal engine roughness threshold THD_ER is influenced mainly by the engine speed. Nevertheless, the air mass has to be taken into account (if NLC_USE_ER_STND = 0)

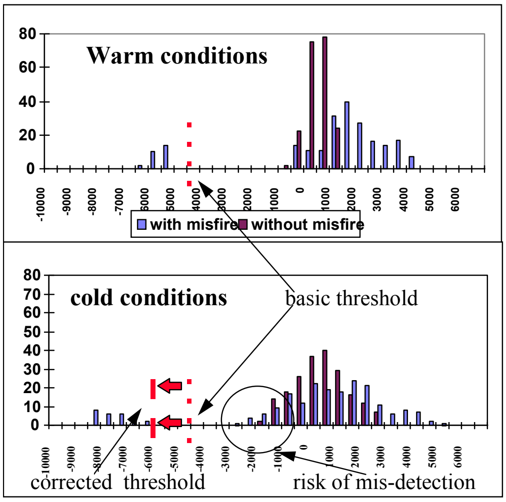


Otherwise, the normalized engine roughness threshold using ER_STND index is influenced essentially by the engine load (if NLC_USE_ER_STND = 1)

- *A threshold temperature correction:*

At cold conditions, the engine friction resistance is more important as well as the combustion is irregular, therefore it exists a possibility of misfire misdetection if the same threshold as warm conditions is kept.

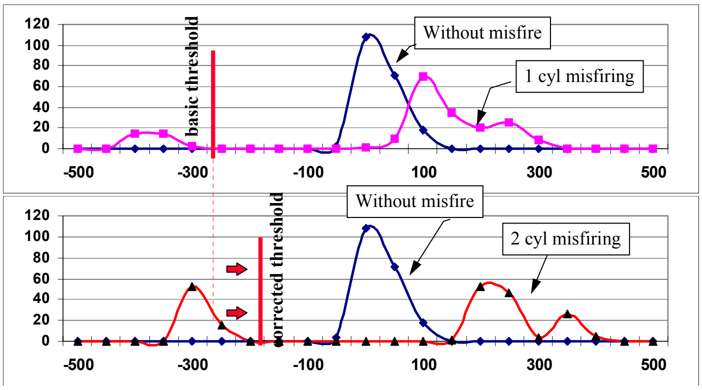
A temperature correction is applied to compensate larger signal amplitudes arise of ER values at lower engine temperatures. This correction is relative to TCO value.



- *A threshold correction during active injection valve cut-off:*

If one or several injection valves are cut off selectively by the engine management system (IV diagnosis...), then the cylinder or cylinders are excluded from the misfire check.

Due to the system, the signal amplitude of the ER value is reduced with a misfire at another cylinder. To improve the detection for this period, the absolute amount of the threshold can be reduced with the factor IP_FAC_THD_IV_ER(N_32) or IP_FAC_THD_IV_S_ER(N_32) *(in stratified mode)*.



- *A threshold correction defined in application incidences:*

FAC_THD_APP_ER is a threshold multiplicative correction who is defined in the misfire detection application incidences file.

51.13.1.3.2 Equipment

Chassis dyno in engine speed control

Remark:

It is better to perform the calibration of the basic threshold on a single roll chassis dyno than on a twin-rolls chassis-dyno. The experience has shown that a single roll chassis-dyno is more representative of the street than the twin-rolls chassis-dyno.

Car with dummy catalyst

One car per transmission type (AT, MT, else)

Engine warm

One misfire generator (software or external).

Remark:

If external misfire generator used please inhibited the IV and/or IGC diagnosis for no limp-home feedback.

51.13.1.3.3 Calibration Prerequisites

ENRD calibration frozen,

ENRD adaptive process executed and locked

ENRD calculation running & LC_MIS_INH = 0

LOAD_MIN_MIS (legal zero load line, *inhibition triangle included*) already calibrated

51.13.1.3.4 Calibration Step B001_02 – Step 1 / 1

51.13.1.3.4.1 SNR evaluation of ER index and basic detection threshold determination

Note: This calibration step is fundamental to evaluate if the ER index can answer to legal requirements, in the same time the acquisition files obtained during this analysis step will be used to determinate the basic detection threshold.

For SNR criterion definition see definition chapter §3.3.6.

The signal / noise ratio is observed statistically with histogram figures as presented. In case of misfire, ER indexes will drop of a certain gap. This gap is proportional to engine load & speed; as a consequence the SNR is also proportional to engine load & speed.

Under certain conditions (correct adaptive process achieved, no important cylinder unbalancing, no quick dyno rolls control...) ER values that are image of engine "good" combustions & "misfire" can be considered as two normal distributions.

Each distribution has its own average value and standard deviation

Population 1 *(misfire population)*

ER_M1 = mean(ER samples in misfire)

ER_STD1 = std(ER samples in misfire)

Population 2 *(combustion population)*:

ER_M2 = mean(ER samples in combustion)

ER_STD2 = std(ER samples in combustion)

ER SNR criterion: 2 * | ER_M2 - ER_M1 | / [ER_STD1 + ER_STD2]

For each ER histogram a basic misfire detection threshold will be determinate as:

ER Basic Threshold: | ER_M2 - ER_M1 | - 2 * (ER_STD1 + ER_STD2)

51.13.1.3.4.2 Measurement files

Measurement type:

Crankshaft synchronous *(for example SAM2000 flight recorder with Tdc's update rate, oscilloscope tool can't perform TDC acquisition files),* *Chassis dyno in engine speed control mode,*

Engine stabilized.

Channels:

SEG_NR_ER, ER or ER_STND

Basic software inputs *(T_SEG_ER & SEG_T_COR if post computation needed due to new ENRD calibration)*

Engine speed, Engine load *(LOAD_MIS, MAP, MAF, TQI_AV...)*

LV_MIS_GEN_DET if software misfire generator integrated (NLC_USE_MIS_GEN = 1)

Sample numbers: 3000 tdc's recommended for a representative statistical analysis.

One file is acquired per typical misfire generator ratios *(1/2, 1/4 and 1/33 misfire patterns, example for a 4cyl engine)*, per engine operating points & per combustion modes available.

1/2 means one misfire for 2 tdc's (50%): simulation of coil driver trouble for half-static ignition systems on 4cyl engine. *1/3 for 6cyl engine.*

1/4 means one misfire for 4 tdc's (25%): simulation of coil driver trouble for full static ignition systems or injector trouble on 4cyl engine. *1/6 for 6cyl engine.*

1/33, means one misfire for 33 tdc's (3%): 'random pattern', simulation of MIS_Bx emission criterion, increase close to 1.5 times EURO IV target for a 4cyl engine (to be confirmed on your application, less on 5,6 & 8cyl engines). Important note: the 'random' pattern must be an odd ratio in a way to generate a misfire on all cylinders (worst case).

Engine operating points considered:

Proposal *(see also according your legal requirements, calibration timing and engine load references and combustion mode area)*

N [rpm]	N_IS_SP	800	1250	1500	1750	2000	2500	3000	3500	4000	4500	5000*	5500*	Redline*

US applications

LOAD_MIS [%]	LOAD_MIN_MIS (ZLL)	30	35	40	50	60	70	80	90

Measurement recommendations:

Operate one acquisition per misfire pattern.

If you try to link 1/2, 1/4 and 1/33 misfire patterns on only one acquisition file, engine system variables like engine load will fluctuate. As a consequence ER indexes will change also, so statistical analysis will be wrong due to histograms highly disturbed.

Recommended operating mode: In order to avoid ER indexes gradients, switch on the misfire generator with the required ratio, checked engine speed, reach engine load setpoint and finally trigger the acquisition file when operating point is stabilized.

Input files *(theoretical):*

MPI, CE applications MT: 3 misfire patterns * 11 * 9 ~= 300 acquisition files

MPI, CE applications AT/MT: 2 * 3 misfire patterns * 11 * 9 ~= 600 acquisition files

MPI, US applications AT: 3 misfire patterns * 13 * 9 ~= 350 acquisition files

GDI, CE applications MT: 3 misfire patterns * 11 * 9 ~= 300 acquisition files (AFS)

3 misfire patterns * 9 * 6 ~= 160 acquisition files, (S mode)

Total (460 files)

Note: for some combinations between engine operating points and misfire pattern, acquisitions are impossible (engine stalling, engine behavior, stabilization impossible...) especially for low engine speed/load. But in any case, please control that you've executed an acquisition with as least the misfire "random" pattern (1/33 for 4cyl engine).

51.13.1.3.4.3 Acquisition file analysis & calibration

Manual analysis with MS Excel (or else)

For each acquisition files, calculate mean and standard deviation values of ER indexes (combustions and misfires population).

To separate combustions and misfire populations, you can apply a threshold/criterion at 3/4 of the difference between max and min ER values *(criterion to be checked with low engine load files)*

Calculate for each acquisition file the SNR & corresponding basic detection threshold

For each engine operating point, you have to compare the Signal Noise Ratio of the three different misfire patterns and keep the worst one and its associated basic detection threshold, in order to be sure to have at least this SNR ratio for any kind of misfire pattern observed at this engine operating point.

This method is direct but laborious and very long.

51.13.1.3.5 Misfire signal noise target

To be compatible with legal requirements, you have to check that for each monitored operating point, the corresponding SNR has to be at least > 5.

For the misfire monitoring strategy, if for one operating point ER index has a signal noise ratio below 5, it means that an overlap between misfire and combustion populations is possible: risk of wrong detection

If during this calibration step, you remark such a trouble on any acquisition file, please inform SI team and/or Domain 7 to evaluate crankshaft segment drifts, crankshaft torsion or else who could be the trouble source (see dedicated chapter in ENRD tuning guide).

In such case, Torsion and/or adaptive value corrections could be applied to ENRD to increase its signal/noise ratio.

For applications who have large crankshaft oscillations at low engine speed who could strongly decrease the signal noise ratio in this area, a local fade-out is recommended by using the module B00E 'Crankshaft Oscillation Detection' with MISF configuration NLC_USE_CRK_OSC_MIS = 1

51.13.1.3.6 Coolant temperature correction for misfire detection.

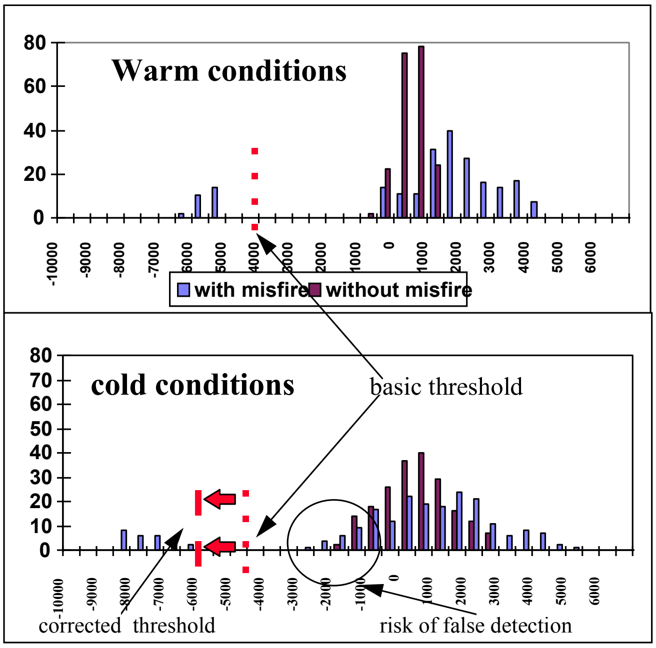
When the engine is in cold conditions, it is more resistant to rotation and the combustion process is more irregular. Therefore it can be necessary to increase the detection threshold in order to avoid misdetections.

The noise of the engine roughness values has to be checked at low temperature conditions and compared to the values with a warm engine.

The calibration value of IP_FAC_THD_TCO_ER at warm condition should be 1, and with an increased value at lower temperature. E.g.: the noise amount on ER index is about 1,2 times bigger at 0°C than at 90°C.

Perform several engine crankings with different coolant temperatures (-20°C ... 20°C...) and apply different misfire ratio. Correct IP_FAC_THD_TCO_ER map in order to be able to detect each misfire pattern on different coolant temperature.

Classic Target: more than 75% of misfire detection efficiency at minimum monitoring coolant temperature (TCO > -7.5°C)



51.13.1.3.7 Cylinder shut-off correction for misfire detection.

The ER value is bigger for one cylinder misfiring than in the case of two cylinders misfiring:

If cylinders in misfire are non-successive, it's very often due to DRV2_MMV_ER correction that increases global ER values.

If cylinders in misfire are successive, the DRV2_MMV_ER effect is present but moreover it's also due to lowest deceleration on second cylinder.

Therefore, when the injector is shut-off on one cylinder, the threshold can be modified for the remaining cylinders.

Make at several engine speeds and load points, measurements with one cylinder misfiring and two cylinders misfiring and calculate at every operating point the ratio of ER between both cases.

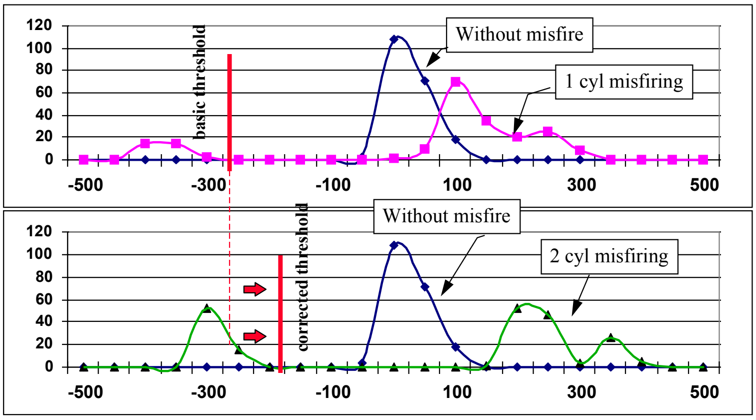
The biggest ER ratio hence measured can than be used for the calibration of the IV correction.

E.g. one cylinder misfiring ER = -400 µs

two cylinders misfiring ER = -300 µs

-300 / -400 = 0.75; if this is the biggest value then calibrate 0.75

If there is at other operating points a ratio of 0.8, then calibrate 0.8.



51.13.1.4 B001_03: Management of general misfire detection fade out according engine roughness index phasing

51.13.1.4.1 NC_SIZE_THD_ER_BUF Integration Check

NC_SIZE_THD_ER_BUF defines the misfire detection threshold buffer size.

We have a correct value of NC_SIZE_THD_ER_BUF when the detection threshold THD_ER is delayed in a way to be synchronized by the correct ER value during accelerations and decelerations.

51.13.1.5 Misfire detection based on engine roughness evaluation B001_04

51.13.1.5.1 Calibration Step

No Calibration Step specific to this section.

Remark on LV_DET_THD_MIS & LV_INH_DET_MIS for calibration validation:

LV_DET_THD_MIS has been introduced to observe the validity of the detection threshold even if the misfire detection is in fade-out due to any reason.

It can be helpful to limit for instance the fade-out occurrence during FTP cycle and to see exactly in which conditions fade-out of the detection is really necessary.

51.13.2 Calibration hints for misfire software generator (Module F001)

51.13.2.1 General Information

According the project SI integration plan and/or customer requirements/wishes, a software misfire generator can be included on EMS software and used during integration validation and calibration stages.

It is strongly recommended to remove this functionally on serial product software: NLC_USE_MIS_GEN = 0

The software misfire generator is integrated if NLC_USE_MIS_GEN = 1

Moreover, for a correct integration of the generator and a full functionality, the inhibit carriers (INH_IV_MIS_GEN & INH_IGC_MIS_GEN) must be integrated:

- Combine INH_IGC_MIS_GEN (bitwise or logical operation with other carriers) with INH_IGC in Ignition Activation Module (6024 - Hook Module) see IGRE definition.
- No specific integration work is necessary for INH_IV_MIS_GEN, as there is a specific link between this carrier and the INJR aggregate

The misfire generator is useful for the following calibration items:

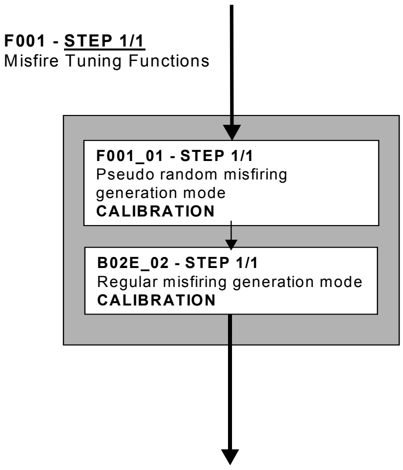
- Misfire Detection: Detection threshold determination, Detection efficiency validation ...
- Misfire Diagnosis: Determination of the MIS_A catalyst destruction criterion & MIS_B emission criterion

Two different generation modes are available:

- Pseudo-random misfiring generation mode mainly used for robustness test of the misfire detection efficiency.
- Regular misfiring generation mode, used to simulate the main misfire patterns (one/two coils off, one cylinder off, emission pattern)

LC_REQ_MIS_GEN is the main calibration switch to enable (=1) or disable (=0) the misfire generator.

51.13.2.2 Module calibration flowchart



51.13.2.3 F001_01: Pseudo random misfiring generation mode

Pseudo random pattern:

The generation mode "Pseudo random misfiring" is based on LC_RND_PAT_MIS_GEN, an index table of 256 cells; each cell corresponds to a TDC scanned by the generator, providing a 256 tdc's sequence.

If a cell is set to "0", it means that this TDC will be set in misfire by ignition (if LC_IGN_OFF_MIS_GEN = 1) and/or by injection (LC_INJ_OFF_MIS_GEN = 1). Otherwise, this TDC will be in firing.

Fill the table LC_RND_PAT_MIS_GEN according the misfire patterns you want to apply.

For a correct scanning of the table, please set the index breakpoints as following:

ldp_x_id_rnd_pat_mis_gen																
-	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240

ldp_y_id_rnd_pat_mis_gen																
-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Generator Activation:

To activate this misfire generation mode, it is recommended to activate the generator by the following sequence:

1. Check that the table LC_RND_PAT_MIS_GEN is mainly filled with "1" data, to avoid immediate stalling.
2. Select the reference segment cylinder C_SEG_ST_MIS_GEN *(first cylinder affected by the first misfire generator cell)*
3. Chose the way to generate the misfires, LC_IGN_OFF_MIS_GEN = 1 and/or LC_INJ_OFF_MIS_GEN = 1
4. Select the misfire generation mode C_MOD_MIS_GEN = 0
5. Switch on the generator LC_REQ_MIS_GEN = 1

If no misfire occurs, please check that at least one of LC_IGN_OFF_MIS_GEN or LC_INJ_OFF_MIS_GEN calibration is set to 1.

To stop the misfire generator, set LC_REQ_MIS_GEN = 0.

51.13.2.4 F001_02: Regular misfiring generation mode

Regular pattern:

The generation mode "Regular misfiring" is based on C_MIS_RATE calibration. The generator will produce a misfiring every C_MIS_RATE tdc's.

Examples:

To perform a 1/2 misfire pattern on a 4 cylinder engine, set C_MIS_RATE = 2

To perform a cylinder off misfire pattern on a 6 cylinder engine, set C_MIS_RATE = 6

To apply a cylinder "roll-over" misfire pattern (misfire generated each time on a different cylinder), set C_MIS_RATE to an odd value *(example: C_MIS_RATE = 33 is often used for misfire detection efficiency tests)*

Generator Activation:

To activate this misfire generation mode, it is recommended to activate the generator by the following sequence:

1. Check that the calibration C_MIS_RATE is not 0 or 1, to avoid immediate stalling.
2. Select the reference segment cylinder C_SEG_ST_MIS_GEN *(first cylinder affected by the regular ratio)*
3. Select the misfire generation mode C_MOD_MIS_GEN = 1
4. Chose the way to generate the misfires, LC_IGN_OFF_MIS_GEN = 1 and/or LC_INJ_OFF_MIS_GEN = 1
5. Switch on the generator LC_REQ_MIS_GEN = 1

If no misfire occurs, please check that at least one of LC_IGN_OFF_MIS_GEN or LC_INJ_OFF_MIS_GEN calibration is set to 1.

To stop the misfire generator, set LC_REQ_MIS_GEN = 0.

51.13.2.5 F001_03: Cylinder regular misfiring generation mode

Cylinder regular pattern:

The generation mode "cylinder regular misfiring" is similar to the 'regular misfiring' mode, but act like a unique misfire generator per cylinder. In this way different misfire ratios can be applied to different cylinders.

The generator will produce a misfiring on cylinder x every NC_CYL_NR * C_MIS_RATE_CYL_x tdc's.

Example:

With a 4 cylinder engine, to perform a 1/2 misfire pattern on a cylinder 3, set C_MIS_RATE_CYL_3 = 2 and LC_MIS_RATE_ENA_CYL_2 = 1.

As a result, we will have a misfire rate of 50% on cylinder 3, but will be a 12.5% ratio as global ratio for the engine if only this cylinder is set in misfire.

Generator Activation:

To activate this misfire generation mode, it is recommended to activate the generator by the following sequence:

1. Select the cylinder(s) x who will be impacted by the misfire generator with LC_MIS_RATE_ENA_CYL_x = 1
2. Check that the calibration C_MIS_RATE_CYL[NC_CYL_NR] is not 0 or 1, to avoid immediate stalling.
3. Select the ratios you want to apply to the different cylinders C_MIS_RATE_CYL[NC_CYL_NR]
4. Select the misfire generation mode C_MOD_MIS_GEN = 2
5. Chose the way to generate the misfires, LC_IGN_OFF_MIS_GEN = 1 and/or LC_INJ_OFF_MIS_GEN = 1
6. Switch on the generator LC_REQ_MIS_GEN = 1

If no misfire occurs, please check that at least one of LC_IGN_OFF_MIS_GEN or LC_INJ_OFF_MIS_GEN calibration is set to 1.

To stop the misfire generator, set LC_REQ_MIS_GEN = 0.

51.13.2.6 F001_04: Misfire generator informations

Two informations are available from the misfire generator.

LV_MIS_GEN is an internal data used for synchronization between request and exchange with INJR (injection realization) and IGRE (ignition realization) aggregates. Mainly useful during validation & integration process. This flag is set to 1 as soon as the misfire generator has requested a cylinder shut-off (segment synchronous).

LV_MIS_GEN_DET is identical to LV_MIS_GEN, but delayed with C_NR_TDC_DLY_MIS_GEN tdc's. This flag is useful to synchronize misfire generator request and result on the ER signal for the misfire detection, and evaluate if the misfire generated has been properly detected. Afterwards, by using this information and LV_DET_MIS, we can after evaluate the wrong detection ratio and/or the non detection ratio with post processing tools or/and by visualization.

Calibration evaluation:

To find the proper phasing of LV_MIS_GEN_DET, please increment C_NR_TDC_DLY_MIS_GEN up to have the proper phasing between this bit and the ER value in misfire.

For this, apply a low misfire ratio, like C_MIS_RATE = 33 and observe on a segment synchronous measurement when LV_MIS_GEN_DET = 1 and the misfire on ER will be present.

As example: for a 4 cylinder engine C_NR_TDC_DLY_MIS_GEN = 9.

51.14 Misfire OBDII (cus)

51.14.1 Calibration hints for Misfire Detection (APP INC) (Module B00T)

For the module B00T who is application specific, no specific calibration method is recommended in this tuning guide.

Please contact your project SI Team and Domain for calibration recommendations if specific algorithm has been integrated

51.14.1.1 B00T_01: Detection threshold correction – Application specific gain

FAC_THD_APP_ER is a gain specified between 0 and 2 that correct the basic detection threshold defined in the generic core function.

This gain can be modified between 0.5 and 1 in case of specific conditions when the detection is difficult (catalyst heating period...).

Otherwise, when the risk of wrong detection becomes too important, this gain can be set between 1 and 2, in a way to minimize the detection sensitivity in very noisy conditions on the ER index (case of important knock events, limp-home...).

Can be also used in case of specific end of line process.

FAC_THD_APP_ER = 1 has no effect.

51.14.1.2 B00T_02: Misfire engine load – Application specific gain

The gain FAC_GAIN_LOAD_MIS allows to modify the engine load information used in ENRD and MISF in case of torque reduction according the system variable chosen as reference by calibration.

FAC_GAIN_LOAD_MIS = 1 has no effect.

51.14.1.3 B00T_03: Inhibition based on OBI diagnosis

LV_INH_OBD_DET_MIS has to be set to 1 when at least one of the most important sensors providing input system variables for MISF is declared in failure.

Typical: CRK, CAM, TPS, MAF, MAP ...

LV_INH_OBD_DET_MIS = 0 has no effect.

51.14.1.4 B00T_04: Inhibition based on application functions

LV_INH_APP_DET_MIS has to be set to 1 when at least one of the functions that could corrupt the misfire detection is declared active.

Typical: ESP, canister purge on going...

LV_INH_APP_DET_MIS = 0 has no effect.

51.14.1.5 B00T_05: Inhibition of the crankshaft oscillation detection function

LV_INH_CRK_OSC_DET has to be set to 1 when conditions to detect crankshaft oscillations are unreliable, or not necessary.

LV_INH_CRK_OSC_DET = 0 has no effect.

51.14.1.6 B00T_06: Misfire low fuel tank level informations

Two different low fuel tank level informations have been introduced in MISF answering to two different methods generally used by customers.

LV_INH_FTL_L_DET_MIS, who is a low fuel inhibition bit dedicated to misfire detection. This bit is used and located on bit 8 of MIS_DET_CDN_APP_INH the generic fade out condition carrier.

If LC_FTL_L_DET_MIS is set to 1, a misfire detection fade-out coming from low fuel tank level information is possible.

Take care of your legal "Low Fuel Tank Level" definition:

- for US you can declare a low fuel level when level is **15 percent** or less of the nominal capacity of the fuel tank.
- for Europe you can declare a low fuel level when level is **20 percent** or less of the nominal capacity of the fuel tank.

Note: with the maskable fade-out management, even if LC_FTL_L_DET_MIS = 1, if the bit 8 of MIS_DET_CDN_APP_INH is masked to 0, by C_MASK_APP_INH_DET_MIS and/or C_MASK_APP_INH_DET_ZDLY_MIS, the low fuel tank level fade-out will be cancelled. Check that this fade-out is allowed on both side (LC_FTL_L_DET_MIS and by C_MASK_APP_INH_DET_MIS and/or C_MASK_APP_INH_DET_ZDLY_MIS).

LV_FTL_L_DIAG_MIS is a low fuel information, which will allow the error management to save a specific misfire error called 'MIS_FTL' and a specific Pcode 'P313'.

Note: If LC_FTL_DET_MIS = 1 and corresponding bits 6 inside calibrations C_MASK_APP_INH_DET_MIS & C_MASK_APP_INH_DET_ZDLY_MIS are set to 1, in such case LC_FTL_L_DIAG_MIS = 1 has not effect (misfire errors not possible in low fuel, because the detection is fade-out).

Combination LC_FTL_DET_MIS = 0 & LC_FTL_L_DIAG_MIS = 0, will process low fuel tank level as a nominal condition for misfire monitoring.

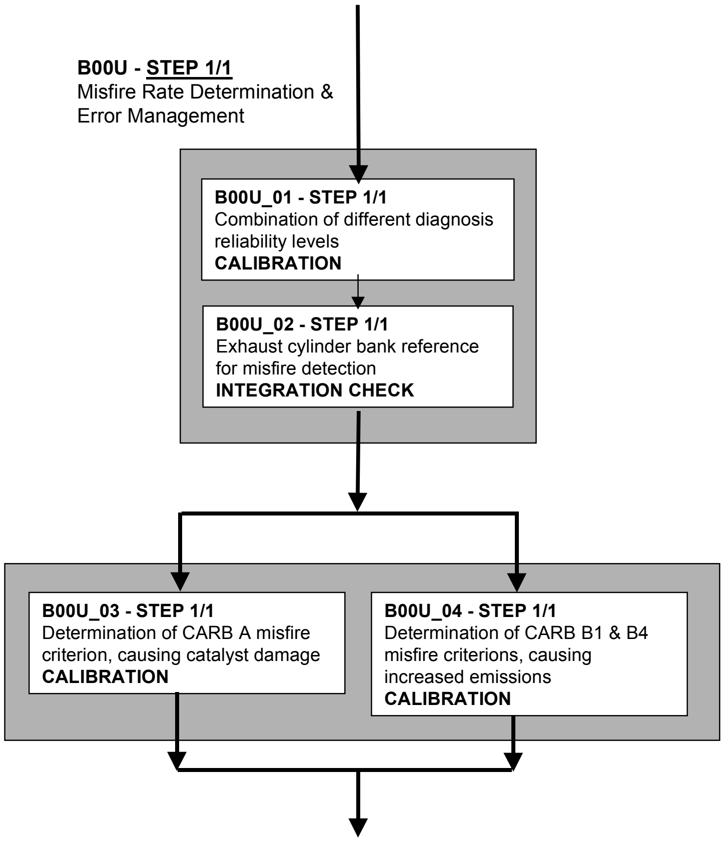
51.14.1.7 B00T_07: Misfire detection calibration ease

This ratio between ER index and detection threshold is clipped by the C_ER_THD_RATIO_MAX calibration.

To be suitable, ER_THD_RATIO must be clipped at least at 100% (misfire occurence).

51.15 Misfire rate determ. (gen)

51.15.1 Module calibration flowchart



51.15.2 B00U_01: Combination of different diagnosis reliability levels

51.15.2.1 General information

Due to their physical principle & design, the individual diagnostic functions have a different reliability depending on the current operating point.

To perform best possible confirmation of misfire detection, the function combines the ENRD+MISF method results with non debounced informations of ignition and injector diagnosis.

This functionality can be useful on some specific operating points where the misfire detection efficiency could be affected by: low load, engine combustion mode, number of cylinders in fuel shut-off...

Moreover, to use of ignition and injection diagnosis informations for misfire detection imply that correlation between non-debounced electric diagnosis and misfire are always true, if not the combination with IV or IGC diagnosis must be avoid in concerned area.

LV_INH_IGC_DIAG_MIS & LV_INH_IV_DIAG_MIS must be set in module B02E according OBDI IGC & IV diagnosis conditions to avoid wrong detections & misfire monitoring conditions *(crossed diagnosis is not necessary when out of monitoring area)*.

51.15.2.2 Calibration Prerequisites & Precalibrations

IGC & IV diagnosis calibrated.

IGC & IV diagnosis efficiencies are known on every engine operating points *(please contact your HW pilot and diagnosis tuning team to have these informations).*

Misfire detection efficiency results according engine operating points, combustion modes and misfire patterns.

Note: If the misfire detection efficiency is 100% on every operating points every combustion modes and with every misfire patterns; the non-debounced diagnosis information from Ignition coils & Injection valves are not necessary.

In such case, please set IP_FAC_IV_DIAG_MIS = 0, IP_FAC_IGC_DIAG_MIS = 0 and IP_FAC_ER_x_DIAG_MIS = 1.

These calibrations can be used as precalibration as long as IGC & IV diagnosis efficiencies are unknown.

51.15.2.3 Calibration Step B00U_01 – Step 1 / 1

The principle of the combination is that in every point optimised reliability for the misfire detection is reached even if the single diagnostic is not so strong in some points.

Calibration of IP_FAC_ER_y_DIAG_MIS

If misfiring is suspected by ENRD & MISF result (LV_DET_MIS=1), a weighting factor IP_FAC_ER_y_DIAG_MIS is applied to FAC_ER_DIAG_MIS for the corresponding operating point according the current engine combustion mode.

- A weighting value of **0** means that the detection is not possible for this engine operating point & combustion mode.
- A value equal to or greater than **1** indicates that the detection is reliable for this engine operating point & combustion mode.

Nominal case: IP_FAC_ER_y_DIAG_MIS = 1

Calibration of IP_FAC_IGC_DIAG_MIS

If misfire is suspected by the ignition diagnosis (no ignition spark, firing period too short) a weighting factor FAC_IGC_DIAG_MIS is applied for the corresponding operating point.

Moreover, this weighting factor can be forced to 1, when the number of cylinders in shut-off due to MIS_A criterion reaches C_MIS_NR_OFF_IV_IGCFB cylinders.

Recommended value: C_MIS_NR_OFF_IV_IGCFB = 1

Remark: if C_MIS_NR_OFF_IV_IGCFB >= C_SUM_INH_IV_MAX_MIS (see MISF tuning guide part 1), FAC_IGC_DIAG_MIS will have no effect as the misfire diagnosis will be disable by the flag LV_INH_IV_OFF_DET_MIS.

Nominal case: IP_FAC_IGC_DIAG_MIS = 0

Calibration of IP_FAC_IV_DIAG_MIS

If misfire is suspected by the injection valve diagnosis (wide-open injector, injection period too short) a weighting factor FAC_IV_DIAG_MIS is applied for the corresponding operating point.

Nominal case: IP_FAC_IV_DIAG_MIS = 0

51.15.3 B00U_02: Exhaust cylinder bank reference for misfire detection

51.15.3.1 General information

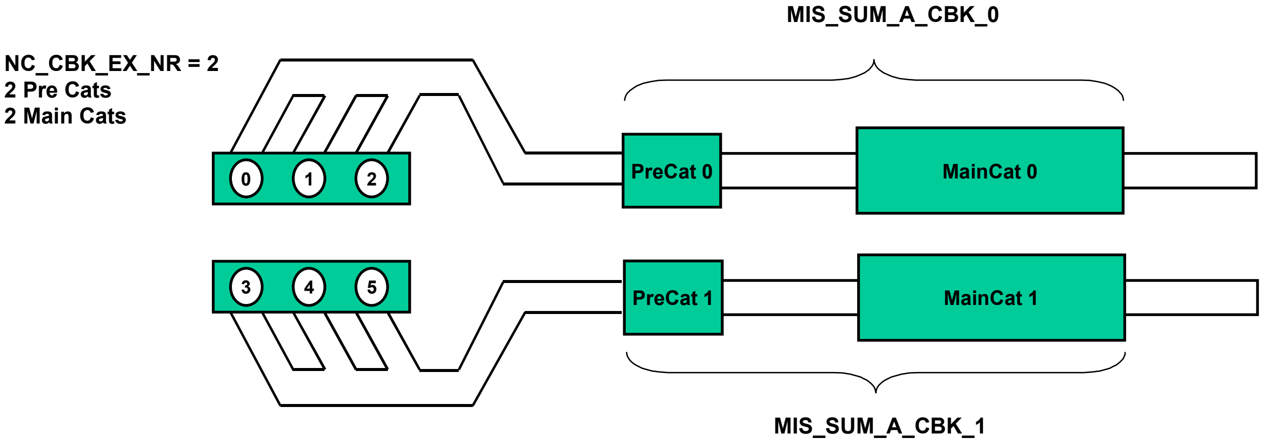
Exhaust cylinder bank reference for misfire detection allows to increase MIS_A & MIS_Bx criterions on the dedicated cylinder bank weighting factor if the misfire detected has effect on the impacted cylinder bank and not on the other (if multiple exhaust cylinder bank design).

general specification

NC_CBK_EX_NR_MISF is a configuration table that takes in account the number of cylinder banks & the number of cylinders.

In case of a project with multiple exhaust cylinder banks, it allows to switch on the correct cylinder bank according the cylinder detected in misfire with the correct ER phasing.

Example:



SEG_NR	0	1	2	3	4	5
NC_CBK_EX_NR_MISF	0	0	0	1	1	1

51.15.3.2 Integration Check

Please check that SEG_NR_CBK_ER is correctly set according SEG_NR_ER values and your configuration data set in module 100L (MISF Configuration Data).

51.15.4 B00U_03: Determination of CARB A misfire criterion, causing catalyst damage

51.15.4.1 General information

LV_STATE_MIS_A is one of two malfunction criterions of CARB OBDII misfire monitoring requirements.

It's the malfunction criterion related to misfire ratio causing catalyst damage. CARB A criterion is evaluated each 200-crankshaft revolutions (long term diagnosis); moreover a short-term diagnosis is available in case of very strong misfire rate detected (short-term diagnosis).

51.15.4.2 Calibration Prerequisites

INJR, IGRE, torque & enrichment calibration frozen

Exhaust line design frozen

No specific misfire detection and diagnosis prerequisite (can be perform with no ENRD & MISF calibration set)

51.15.4.3 Equipment

Exhaust line with temperature sensor on each Pre and Main catalyst.

Misfire generator (software or hardware) in ignition shut-off & regular mode. *(See software misfire generator user manual in following chapters)*

51.15.4.4 Calibration Advises

Do not perform this calibration step on chassis dyno; the risk of fire is important on vehicle.

Better to execute it on engine bench, but please note that the cooling of the exhaust line(s) on engine bench is not representative of the nominal cooling process on vehicle.

Check several time during this calibration step, that each catalyst is still active *(no melting):* use catalyst diagnosis if calibrated.

51.15.4.5 Calibration Step B00U_03 – Step 1 / 1

51.15.4.5.1 Catalyst critical overheating tests

This calibration step concerns IP_MIS_A_FAC_AFS, IP_MIS_A_FAC_AFL & IP_MIS_A_FAC_S maps.

IP_MIS_A_FAC_AFS is the map of a weighting factor depending on engine load and engine speed proportional to the misfire destructive effect on the catalyst(s) due to a misfire for AFS combustion mode.

Calibration Process:

In AFS mode, for each operating point (N, LOAD_MIS) that you need to monitor *(check if US or Europe legal target)*, the minimum misfiring rate that enables to reach the catalyst(s) temperature limit(s) is searched.
Start from 0% misfire rate then increase by 2% steps for each engine operating points and notice the misfire rate obtained as soon as you've reached the critical temperature on one of the different sensors of the different catalysts used on the engine.
Note for each point, C_MIS_RATE & C_SEG_ST_MIS_GEN *(useful if even misfire rate is applied to know which cylinder bank is impacted)* used and which catalyst reaches the first the critical temperature.
Between two different operating points, cut the engine misfire generator to cool down the catalyst.
For example:
N: 1000 to Nmax [rpm] (4500 rpm for Europe, Redline for US) with a N step of 1000 rpm
LOAD_MIS: Zero load line to maximum LOAD_MIS with a LOAD_MIS step of 10%

51.15.4.5.1.1 Calibration of MIS_A long term catalyst diagnosis

Calibration step concerning C_MIS_SUM_A_THD, C_MIS_SUM_A_CBK_THD, C_MIS_A_MIN_ER, IP_MIS_A_FAC_y

As it is obvious that the 5% misfiring at (6000 rpm, 600 mg/stk) is more dangerous for catalyst damage than the 20% misfire at (1000 rpm, 300 mg/stk). That's why in order to give a better representation to the weighting factor; the value 100 is attributed to the minimum misfiring rate over all operating points *(MIS_RATE_MIN, not a system data)*. That enables to have a factor MIS_A_FAC between 0...100.

C_MIS_SUM_A_THD = 100 * NC_CYL_NR * *MIS_RATE_MIN*

MIS_A_FAC = 100 will correspond to these 5% *(as an example)* and therefore C_MIS_A_THD = 3000 in case of a 6 cylinder engine and C_MIS_A_THD = 2000 in case of a 4 cylinder engine.

The factor IP_MIS_A_FAC_y has to be calculated for each operating point as follow:

MIS_A_FAC = C_MIS_SUM_A_THD / (*Misfiring Rate obtained on current operating point* * NC_CYL_NR)

Example: 4 cylinder engine, AFS combustion mode

*(MIS_RATE_MIN = 4% @ LOAD_MIS = 100% and N = 4500 rpm), ratio are used in formula as entire numbers; MIS_RATE_MIN = 4 => **C_MIS_SUM_A_THD = 1600***

AFS_MODE LOAD_MIS

	31.8	39.4	49.3	74.0	88.8	98.6
MAP						
N	322	400	500	750	900	1000
1000	40	37	33	30	27	25
1500	37	35	32	28	26	22
2000	34	33	31	27	25	20
2500	30	28	25	25	20	18
3000	27	26	24	23	16.7	13
3500	26	24	22	18	12	8
4000	25	22	18	14	5	4
4500	25	20	16	11	4	4

IP_MIS_A_FAC

	LOAD_MIS					
	31.8	39.4	49.3	74.0	88.8	98.6
MAP						
N	322	400	500	750	900	1000
1000	10	11	12	13	15	16
1500	11	11	13	14	15	18
2000	12	12	13	15	16	20
2500	13	14	16	16	20	22
3000	15	15	17	17	24	31
3500	15	17	18	22	33	50
4000	16	18	22	29	80	100
4500	16	20	25	36	100	100

Proceed in the same way for all other combustion modes available on your application.

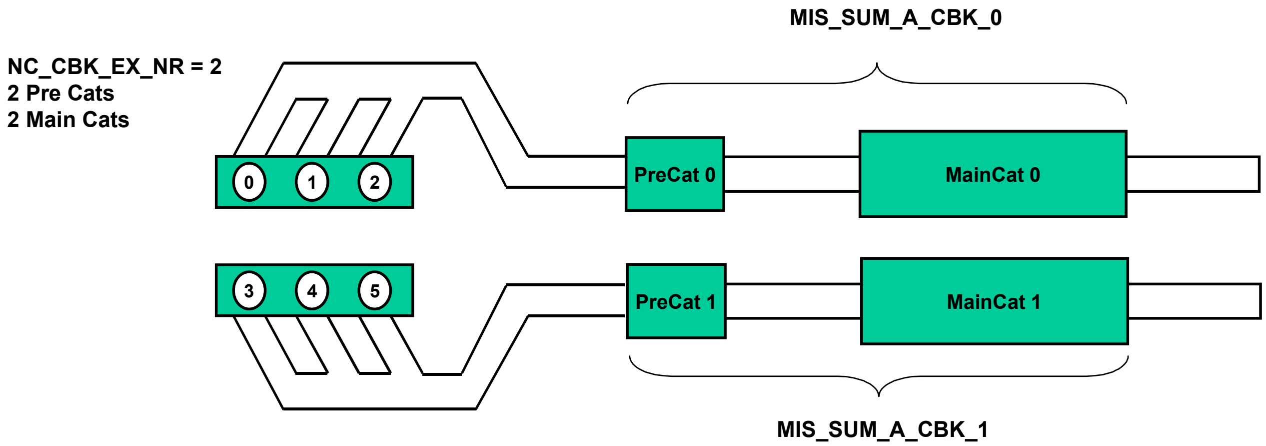
<u>Note for engines with multiple exhaust branches:</u>

For engine with multiple exhaust branches, specifics weighting factor sums MIS_SUM_A_CBK[NC_CBK_EX_NR] have been introduced for dedicated catalyst monitoring, in a way to only diagnose the catalyst impacted by the misfire detected. MIS_A is detected as soon as a MIS_SUM_A_CBK[i] counter exceeds the threshold C_MIS_SUM_A_CBK_THD.

If global catalyst detection preferred, set C_MIS_SUM_A_CBK_THD *= 65535*

If cylinder bank detection preferred, set C_MIS_SUM_A_THD *= 65535*

<u>Example 1:</u>

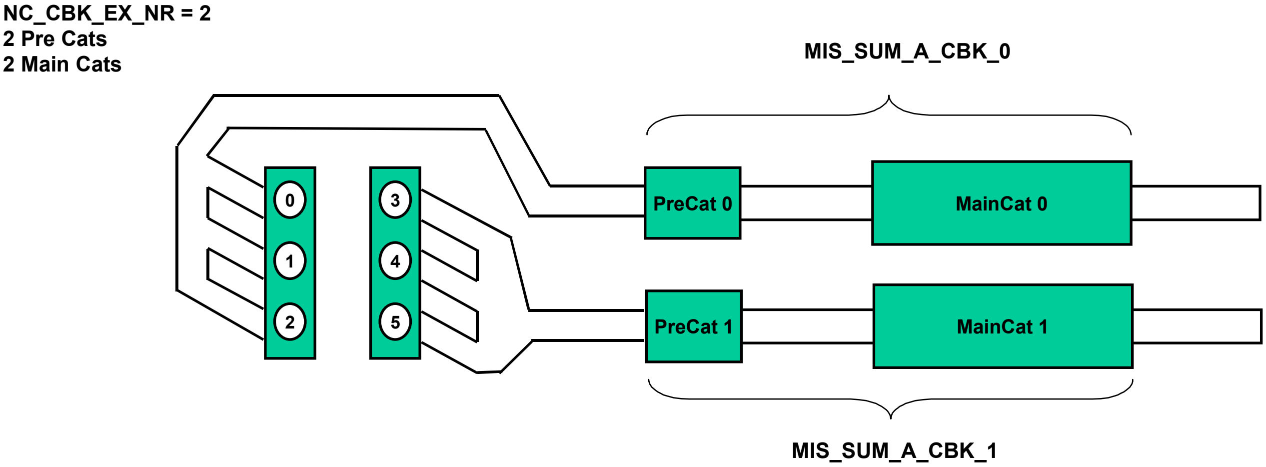


With this exhaust design; **C_MIS_SUM_A_THD** (global threshold) can't be applied, as there are no common catalyst parts between cylinder banks, so the use of MIS_SUM_A_CBK[0] & MIS_SUM_A_CBK[1] is recommended for MIS_A detection with C_MIS_A_CBK_THD threshold.

To know the misfire MIS_A criterion on only one exhaust line during MIS_A criterion calibration, apply each time a C_MIS_RATE (misfire ratio) that could be divided by NC_CYL_NR = 6 (to generate misfire always on the same exhaust), and chose C_SEG_ST_MIS_GEN to select CBK_0 (segments 0, 1 or 2) or CBK_1 (segments 3, 4 or 5).

Example 2:

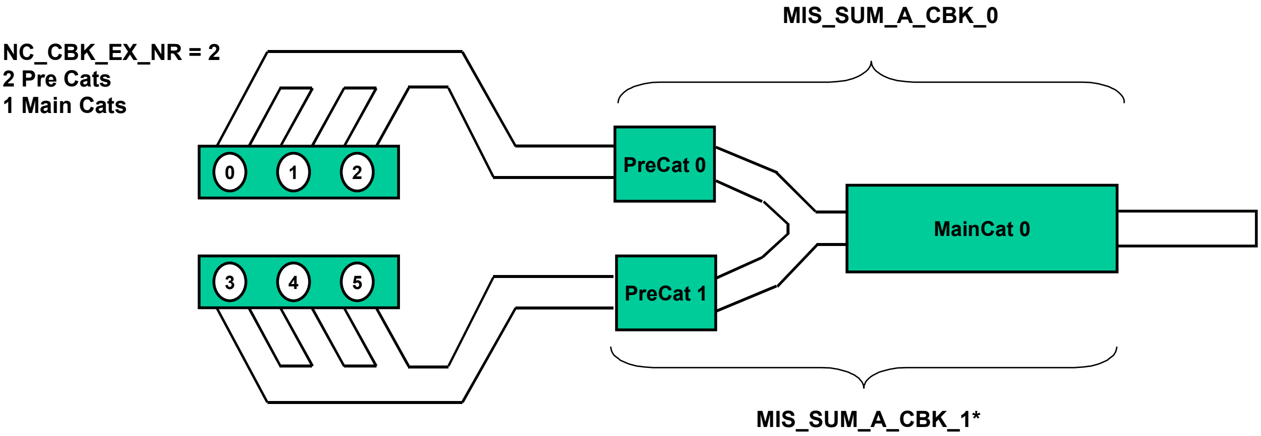
Same configuration than example 1 but with asymmetric exhaust lines *(can occurs on transversal mounted 6 cylinder engines)*



With asymmetric exhaust lines, the misfire MIS_A criterion has to be evaluated separately on the two exhaust lines during MIS_A criterion calibration, apply each time a C_MIS_RATE (misfire ratio) that could be divided by NC_CYL_NR = 6 (to generate misfire always on the same exhaust), and chose C_SEG_ST_MIS_GEN to select CBK_0 (segments 0, 1 or 2) or CBK_1 (segments 3, 4 or 5).

Next apply to the system the calibration set, corresponding to the worst case: exhaust line who will reach the critical temperature on its precatalyst or main catalyst with the lowest misfire ratio.

Example 3:



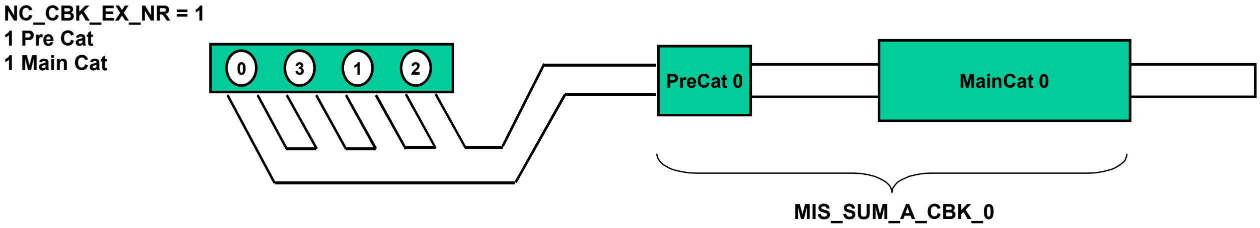
(used only if Pre CATs reach always critical temperature before Main Cat)

With this exhaust design, the choice between C_MIS_SUM_A_THD (global threshold) & C_MIS_SUM_A_CBK_THD (cylinder bank dedicated threshold), depends on the results obtained.

If the precatalysts (it's very often the case) reaches every time the critical temperature with least misfire rate than for main catalyst, the method using C_MIS_SUM_A_CBK_THD (cylinder bank) is recommended.

Otherwise, if the main catalyst reaches every time the critical temperature with least misfire rate than for precatalysts, the method using C_MIS_SUM_A_THD (global) is recommended.

Example 4:



With this exhaust design, C_MIS_SUM_A_THD & C_MIS_SUM_A_CBK_THD act exactly in the same way on the criterion detection *(MIS_SUM_A = MIS_SUM_A_CBK[0]).*

51.15.4.5.1.2 Calibration of MIS_A short term catalyst diagnosis

Nominal legal diagnosis requires a 200 crankshaft revolutions to evaluate MIS_A criterion (long term MIS_A diagnosis) with 3 occurrences if within FTP cycle, or with only 1 occurrence if out of FTP cycle.

In case of every strong misfire rate, this wait of 3 occurrences could be enough to damage one of the catalysts before to apply the cylinder shut-off limp-home.

A MIS_A misfire criterion is set within the 200 crankshaft revolutions window as soon as one of the MIS_CTR_A[NC_CYL_NR] cylinder dedicated counters exceeds the cylinder individual C_MIS_A_IND_THD threshold.

Recommendation for C_MIS_A_IND_THD calibration.

C_MIS_A_IND_THD = [1.5 * C_MIS_SUM_A_THD] / NC_CYL_NR

The 1.5 coefficient can be modified according your safety target for risk of overdetection.

51.15.4.5.1.3 MIS_A Auxiliary calibrations

FTP conditions for MIS_A default criterion.

C_LOAD_MIS_FTP correspond to the maximum load reached on an FTP cycle.

C_N_MIS_FTP corresponds to the maximum engine speed reached on an FTP cycle.

To identify if the current MIS_A window was executed in FTP cycle conditions, at each tdc CTR_FTP_CDN_MIS_A is unitary incremented.

Process during the MIS_A window when a short term is possible due to MIS_CTR_A > C_MIS_A_IND_THD_xT:

In such case, if CTR_FTP_CDN_MIS_A / MIS_NR_TDC_A < C_RATIO_FTP_CDN_IND_MIS_xT, the short term part of the MIS_A window will be declared as 'out of the FTP window', and the LV_MIS_STATE_A default will be set and the limp-home can be triggered.

(_xT stands for _AT if LV_AT=1, stands for _MT otherwise)

Process at the end of the MIS_A window:

If CTR_FTP_CDN_MIS_A / MIS_NR_TDC_A > C_RATIO_FTP_CDN_MIS_x tdc's, then LV_FTP_MIS_A is set to 1.

To declare that a window is within a FTP cycle, you must set 80% < C_CTR_FTP_CDN_MIS_A < 100% of MIS_A window tdc's number *(100 * NC_CYL_NR)*.

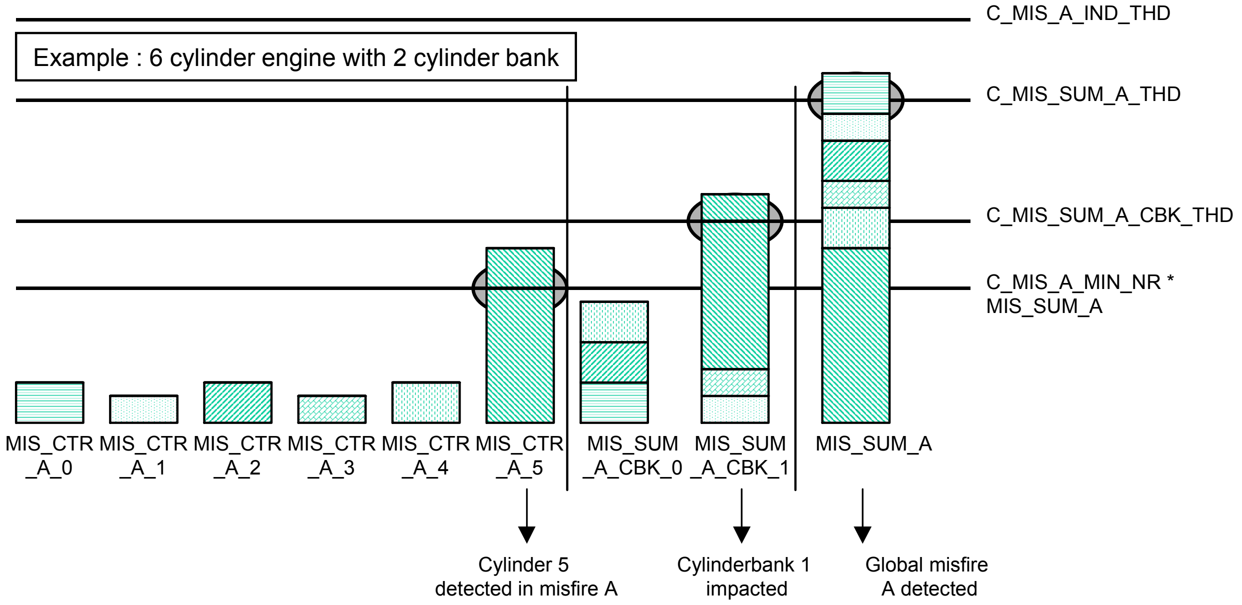
C_RATIO_FTP_CDN_IND_MIS_xT is recommended identical as C_RATIO_FTP_CDN_MIS_xT.

MIS_A default present.

If no MIS_A, MIS_B1 or MIS_B4 error have been already detected, if we are outside the FTP area or if there's no specific request (LV_REQ_APP_CTR_MIS_A=1), LV_STATE_MIS_A needs C_NR_DET_MAX_MIS_A occurrences *(consecutive or not)* within a driving cycle to be set *(at most* C_NR_DET_MAX_MIS_A = *3 according regulations).*

Cylinder identification

The cylinder identification is based on an analysis of each misfire cylinder specific counters; the one(s) identified with a number of misfire at least greater than global misfire sum MIS_SUM_A multiplied with a gain C_MIS_A_MIN_NR (defined between 0 and 100%). This ratio depends of maximum number of Pcode you wish to provide on one MIS_A failure and / or to detect half-static ignition modules default.



If no specific cylinder is identified with the C_MIS_A_MIN_NR * MIS_SUM_A criterion, LC_CONF_RND_DET_MIS_A will select the way to manage such situation.

If LC_CONF_RND_DET_MIS_A = 0, the diagnosis evaluate the cylinder with the highest weighting factor. No random misfire is stored

Random misfire identification

If LC_CONF_RND_DET_MIS_A = 1, the diagnosis will store in the MIS_A symptom a dedicated RDN information. This information will produce next in the error management, the Misfire random information (Pcode 0x300).

Multiple misfire identification

If LC_CONF_MPL_DET_MIS_A = 1 and the number of cylinders set to 1 within SYM_CYL_MIS_A is equal or over C_NR_CYL_MPL_MIS_A, an additional information in MIS_A symptom (MPL). This information will produce next in the error management, the Misfire multiple information (Pcode 0x300, equals to RND).

MIL management in case of MIS_A limp-home

If LC_CONF_MIS_A_LIH = 1 and at least a cylinder is in shut-off due to a misfire MIS_A limp home, even if no misfire is currently detected, LV_MIS_STATE_A will be forced to 1 to allow a MIL blinking during MIS_A limp-home.

Cylinder shut-off on MIS_A default criterion.

In case of short term catalyst protection (monitoring process during CARB A window) or if in long term catalyst protection (monitoring process at the end of CARB A window) and if at least a cylinder in misfire has been identified, a cylinder shut-off limp-home can be triggered to stop fuel injection in the exhaust line.

Moreover, this cylinder shut-off limp-home is allowed for short-term diagnosis only if LC_ENA_MIS_A_LIH = 1 and CTR_DET_MIS_A > ID_NR_ENA_MIS_A_LIH_xT. The table ID_NR_ENA_MIS_A_LIH_xT allows to delay the cylinder shut-off even if the LV_MIS_STATE_A is present. The limp-home will be applied only if the default is still present after ID_NR_ENA_MIS_A_LIH_xT MIS_A occurrences.

For no specific effect with the tables ID_NR_ENA_MIS_A_LIH_xT, set all cells equals to 3 and/or like C_NR_DET_MAX_MIS_A.

The cylinder identified in misfire for such cases can be shut-off by using the carrier INH_IV_MIS to inform the fuel system which cylinder is impacted by this limp-home.

C_MIS_MAX_NR_OFF_IV indicates how many cylinders can be shut-off by this limp home maximally.

Preferred: 1 for a 4-cyl engine, 2 for a 6 cylinder engine *(see customer requirements).*

If C_MIS_MAX_NR_OFF_IV = 0, this limp-home mode will be disable.

Conditions for Cylinder switch back on:

Cylinder(s) x can be switch on according two different methods:

3. Cylinder(s) x can be switched back on at the end of the driving cycle

   LC_CONF_INH_IV_MIS = 0

4. Cylinder(s) x can be switched back on after a defined number of fuel cut-off phases

   LC_CONF_INH_IV_MIS = 1

With C_MIS_MIN_NR_PUC is the minimum number fuel cut-off phases *(LV_PUC)* before cancelling cylinder(s) shut-off, each one with a minimum duration of C_T_MIS_MIN_PUC *[s]*.

Moreover, to avoid infinite cylinder switch back on when there's a really important trouble on one cylinder, when LC_CONF_INH_IV_MIS = 1, the reactivation on one cylinder is limited to C_NR_MAX_MIS_A_LIH_CYL on the current driving cycle.

This limitation is per cylinder and not global.

By using C_NR_MAX_MIS_A_LIH_CYL = 3, each time you will have a cylinder shut-off due to MIS_A diagnosis on cylinder x, the counter CTR_MIS_A_LIH_CYL_x will be limited incremented by one. After 2 occurrences of limp-home > fuel reactivation after PUC phase, at the next fuel shut-off request on this cylinder x, the fuel reactivation will be impossible on this cylinder for the current driving cycle. Reactivation is possible on the others cylinders up to reach C_NR_MAX_MIS_A_LIH_CYL.

At the next driving cycle, this control process restarts from 0.

Project specific request to modulate MIS_A default criterions.

The project can define in their application incidences file, different parameters that can bypass or modulate the generic thresholds for MIS_A detection criterion.

- LV_REQ_APP_CTR_MIS_A, if set to 1 during diagnosis, acts as an accelerator of the diagnosis (equivalent to LV_FTP = 0).
- LV_MIS_A_DIAG_REQ_APP, if set to 1 during diagnosis, changes all the generic thresholds with the following gains (between 0 and 2):
    - FAC_MIS_A_APP & FAC_MIS_SUM_A_THD_APP for the MIS_A criterions at the end of the window
    - FAC_MIS_A_THD_IND_APP for the MIS_A criterion during the window.

51.15.5 B00U_04: Determination of CARB B1 & B4 misfire criterions, causing increased emissions

51.15.5.1 General information

LV_STATE_MIS_B1 & LV_STATE_MIS_B4 are the emission malfunction criterions of CARB OBDII misfire monitoring requirements.

CARB B1 criterion is evaluated at the end of the first 1000 crankshaft revolutions of the driving cycle. CARB B4 is based on all other 1000 crankshaft revolutions windows of the driving cycle.

51.15.5.2 Calibration Prerequisites

Emission calibration frozen *(for each transmission type)*

No specific misfire detection and diagnosis prerequisite (can be perform with no ENRD & MISF calibration set)

51.15.5.3 Calibration Step B00U_04 – Step 1 / 1

CARB B1 and B4 are based on the same conception than MIS_A, but with a unitary increment step (independent to engine operating point).

51.15.5.4 Calibration of C_MIS_SUM_B1/4_THD_xT (detection thresholds)

Calibration should be done once all emissions calibrations enable to reach at least 90 % of the emission target.

Use always the same calibration data set for each cycle performed during this calibration step.

Perform emission cycles with an 80K catalyst while generating misfiring

Goal: Execute for each transmission type several FTP (or MVEG) test cycles with an increasing number of misfires to show the correlation between misfire rate and emission increase. (See graphic below)

Adjust the value of the misfiring rate in order to obtain 1,5 times the emission standard level of one pollutant (ex: often HC reaches its limit in first).

C_MIS_B_THD stands for C_MIS_SUM_B1_THD_AT, C_MIS_SUM_B4_THD_AT, C_MIS_SUM_B1_THD_MT, C_MIS_SUM_B4_THD_MT

Start with a reference cycle (no misfire generation), and then increase the misfire rate by 1% step. As soon as you reach 1,5 times one of emission criterions of your vehicle standard, note the corresponding misfire rate *(MIS_RATE_Emissions)*.

To calculate C_MIS_B_THD = (*MIS_RATE_Emissions* / 100) * NC_CYL_NR * 500

Example: *MIS_RATE_Emissions* is 2,5% for a 4-cyl engine, then C_MIS_B_THD = (2.5/100)* 2000 = 50

The effective threshold C_MIS_B_THD should be close to this value, but nevertheless modulated by your misfire detection efficiency on cycle.

Finally, an emission test has to be run afterwards with the same misfire rate successfully. Perform this calibration step for each transmission type vehicles.

Note for Europe application: C_MIS_SUM_B1_THD_xT can be set differently of C_MIS_SUM_B4_THD_xT, to avoid overdetection at engine start. Contact your OBD co-ordinator for legals update on this topic.

Note for US application: today it's no more allowed to apply a specific emission threshold for MIS_B1 regarding to MIS_B4 nominal threshold, C_MIS_B_THD = C_MIS_SUM_B1_THD_AT/MT = C_MIS_SUM_B4_THD_AT/MT.

Example:

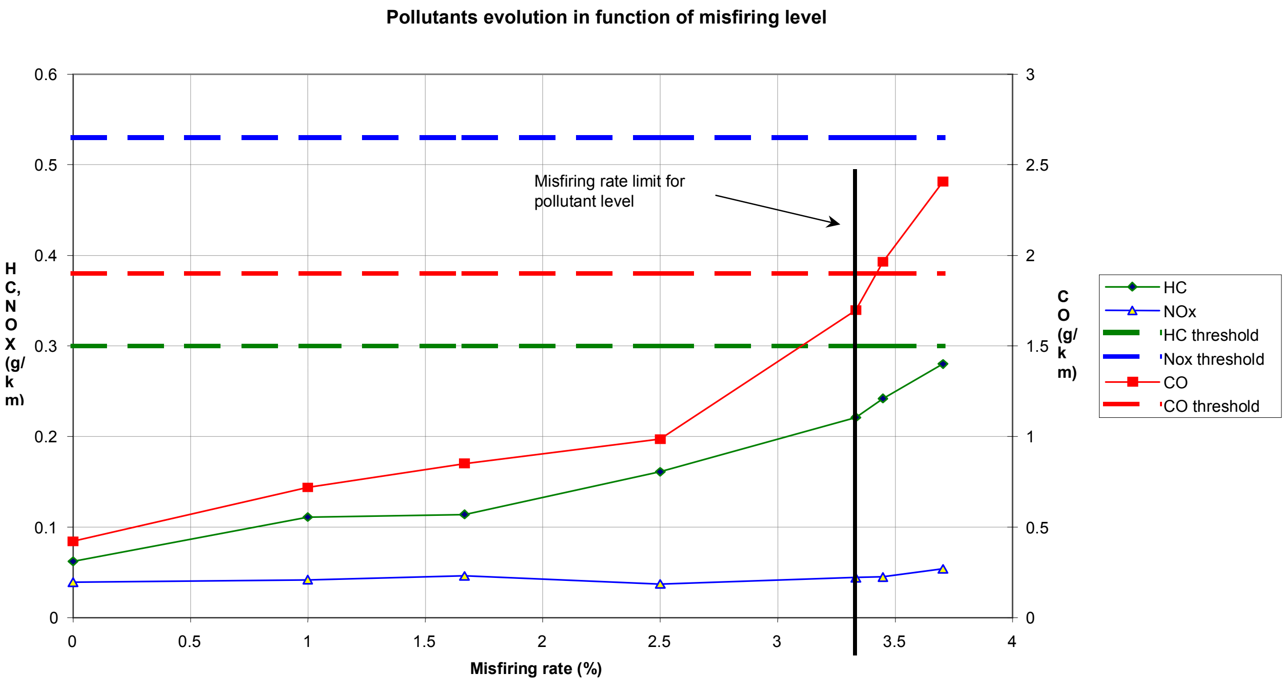
The 1.5 times the level of emission given by the customer (HC : 0.3 g/km ; CO : 1.9 g/km ; Nox : 0.53 g/km on an ECE+EUDC cycle, Europe application, *customer specific emission level, don't use it as a reference, use your application emission standards*).

To calibrate the threshold, several pollutant cycles (7) were made with increasing misfire rate.

On this application the CO target is reached first

CO > 1.9 g/km, C_MIS_RATE = 29, Misfire rate = 3.44%, NC_CYL_NR = 4

C_MIS_SUM_Bx_THD_MT = 68



<u>Note for engines with multiple exhaust branches:</u>

For engine with multiple exhaust branches, specifics sums MIS_SUM_B_CBK[NC_CBK_EX_NR] have been introduced for dedicated emission monitoring per catalyst branch, in a way to only diagnose the catalyst emission impacted by the misfire detected. MIS_B1 / MIS_B4 are detected as soon as a MIS_SUM_B_CBK_i counter exceeds the thresholds C_MIS_SUM_B1/4_CBK_THD*.*

If global catalyst detection preferred, set C_MIS_SUM_B1/B4_CBK_THD = *65535*

If cylinder bank detection preferred, set C_MIS_SUM_B1/B4_THD = *65535*

The MIS_B1/B4 detection per cylinder bank is mainly dedicated for projects that use asymmetric exhaust design (see example 2, in MIS_A chapter)

With asymmetric exhaust lines, the misfire MIS_Bx criterions have to be evaluated <u>separately on the two exhausts lines</u> during MIS_B criterion calibration.

Apply each time a C_MIS_RATE (misfire ratio) that could be divided by NC_CYL_NR = 6 (to generate misfire always on the exhaust), and chose C_SEG_ST_MIS_GEN to select CBK_0 (segments 0, 1 or 2) or CBK_1 (segments 3, 4 or 5).

51.15.5.5 Auxiliary calibrations

<u>MIS_B1 default present.</u>

As there's only one MIS_B1 window per driving cycle, if it occurs, the MIS_B1 default is immediately defined as present at the end this window.

<u>MIS_B4 default present.</u>

To be declared present, LV_STATE_MIS_B4 needs C_NR_DET_MAX_MIS_B4 occurrences *(consecutive or not)* within a driving cycle to be set *(at most C_NR_DET_MAX_MIS_A = 4).*

<u>Cylinder identification</u>

The cylinder identification is identical to principle applied on MIS_A default criterion.

There're specific gains for MIS_B1 and MIS_B4, respectively C_MIS_B1_MIN_NR & C_MIS_B4_MIN_NR.

If no specific cylinder is identified with the C_MIS_B1_MIN_NR (or C_MIS_B4_MIN_NR) * MIS_SUM_B criterion, LC_CONF_RND_DET_MIS_B will select the way to manage such situation.

<u>Random misfire identification</u>

If LC_CONF_RND_DET_MIS_B = 1, the diagnosis will store in the MIS_B1 and/or MIS_B4 symptoms a dedicated RDN information. These informations will produce next in the error management, the Misfire random information (Pcode 0x300).

<u>Multiple misfire identification</u>

If LC_CONF_MPL_DET_MIS_B = 1 and the number of cylinders set to 1 within SYM_CYL_MIS_B1 and/or SYM_CYL_MIS_B4 is equal or over C_NR_CYL_MPL_MIS_B, an additional information in MIS_B1 and/or MIS_B4 symptom (MPL). These informations will produce next in the error management, the Misfire multiple information (Pcode 0x300, equals to RND).

<u>Project specific request to modulate MIS_Bx default criterions.</u>

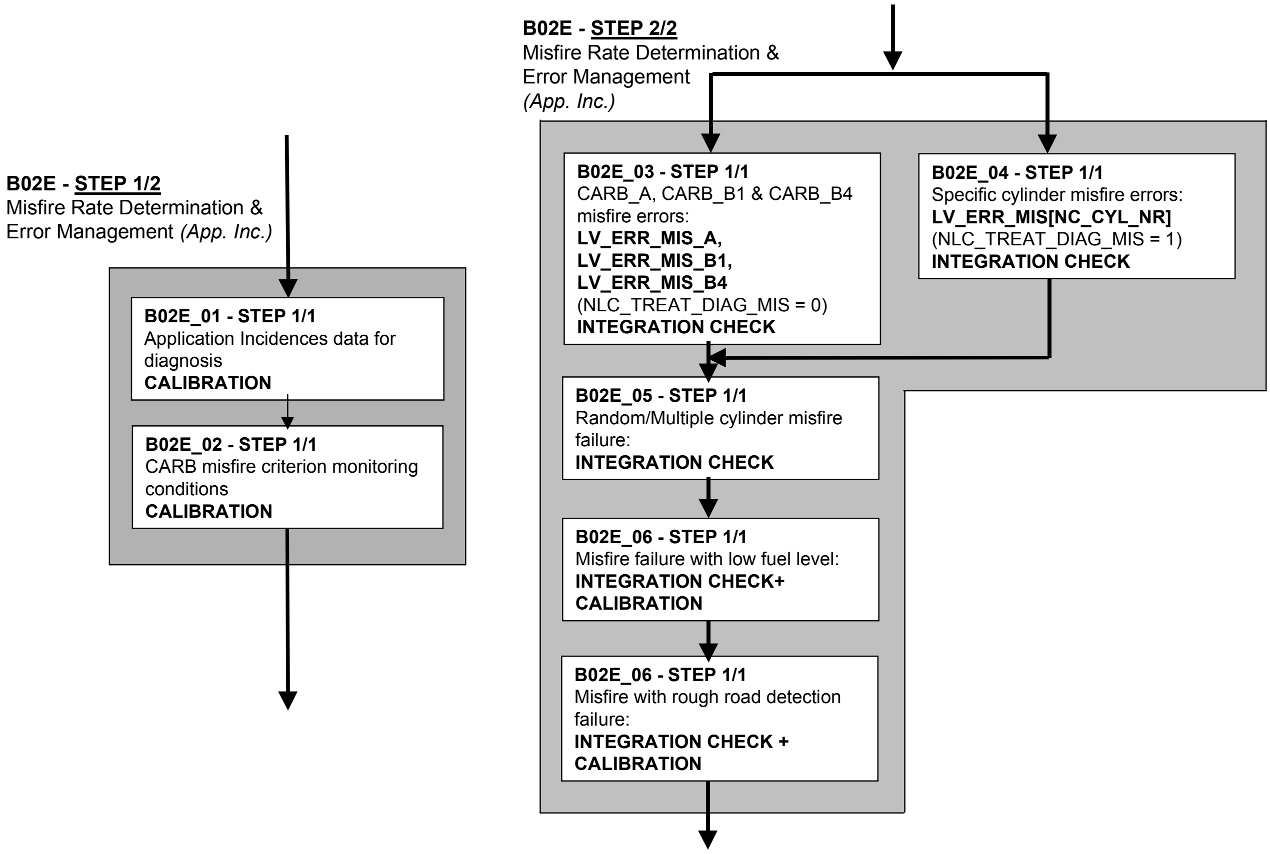
The project can define in their application incidences file, different parameters that can bypass or modulate the generic thresholds for MIS_B1 and/or MIS_B4 detection criterions.

LV_MIS_B_DIAG_REQ_APP, if set to 1 during diagnosis, changes all the generic thresholds with the following gains (between 0 and 2):

- FAC_MIS_SUM_B1_THD_APP at the end of the MIS_B1 window
- FAC_MIS_SUM_B4_THD_APP at the end of the MIS_B4 windows

51.16 Misfire rate determ. (cus)

51.16.1 Module calibration flowchart



51.16.2 B02E_01 : Application Incidences data for diagnosis

51.16.2.1 General information

This module will specify the definition of LV_REQ_APP_CTR_MIS_A : an application specific request to force the out of FTP counting mode (MIS_A long term diagnosis confirmed with one occurrence) even if we are within FTP area.

51.16.3 B02E_02 : Determination of CARB misfire criterion monitoring conditions

LV_DC	LC_CTR_CONF_DIAG_MIS	LV_INH_DET_MIS	LC_CTR_CONF_OSC_MIS	LV_STATE_CRK_OSC	LV_CDN_MIS_A LV_CDN_MIS_B1 LV_CDN_MIS_B4	MIS_A, MIS_B1 & MIS_B4 Monitoring Modes
0	x	x	x	x	0	Out of Driving Cycle
1	0	x	x	x	1	All TDC's Counting Mode
1	1	1	x	x	0	Diagnosed TDC's Counting Mode / Classical Fade Out
1	1	0	0	0	1	Diagnosed TDC's Counting Mode / No Fade Out
1	1	0	0	1	0	Classical Fade Out **with** CRK_OSC / Diagnosed TDC's Counting Mode
1	1	0	1	0	1	Diagnosed TDC's Counting Mode / No Fade Out
1	1	0	1	1	1	Classical Fade Out **without** CRK_OSC / Diagnosed TDC's Counting Mode

51.16.3.1 General information

Depending on customers & legal requirements reached, two Tdc's counting modes are available:

- Counting each Tdc's even if some aren't diagnosed *(LC_CTR_CONF_DIAG_MIS = 0)*
- Counting only diagnosed Tdc's *(LC_CTR_CONF_DIAG_MIS = 1)*

51.16.3.2 Calibration Step B02E_01 – Step 1 / 1

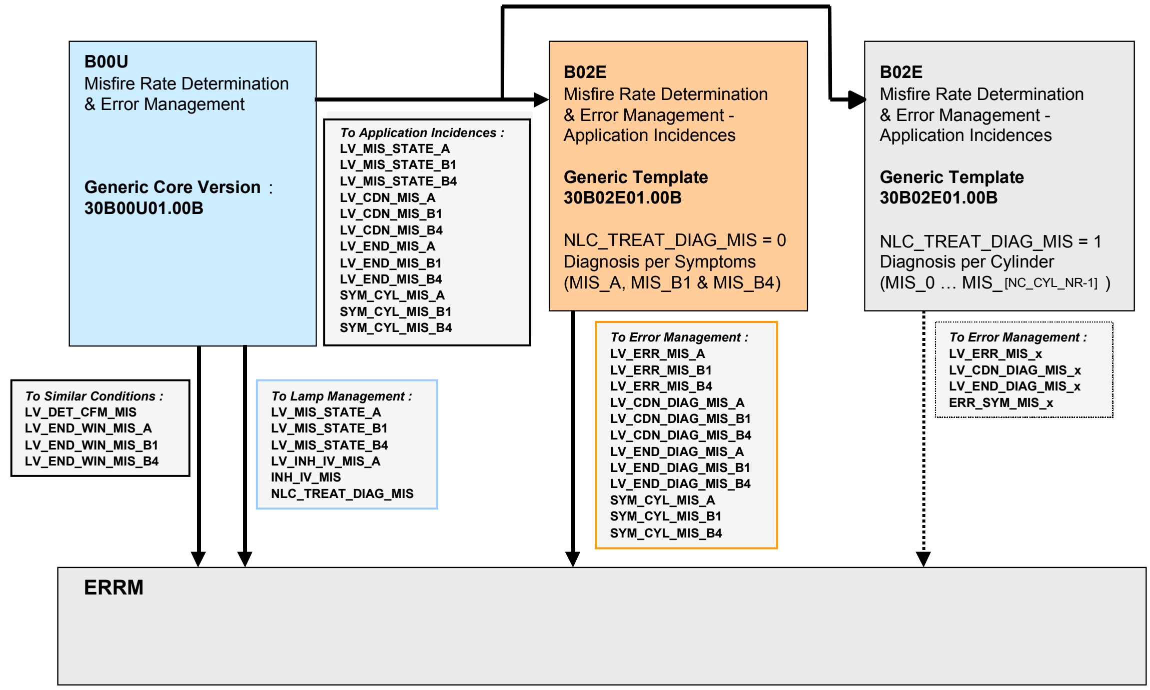
LC_CTR_CONF_DIAG_MIS = 0, is not recommended as the results in term of misfire ratio are always reduced in such case: number of misfire detection / number of TDC diagnosed (LV_CDN_MIS_x = 1). As a result it could produce MIS_A and/or MIS_Bx non-detections according the MISF detection fade-out calibration.

Please note that LC_CTR_CONF_OSC_MIS has not effect on diagnosis conditions, when NLC_USE_CRK_OSC_MIS = 0 (crankshaft oscillation detection not integrated).

51.16.4 B02E_03 : CARB_A, CARB_B1 & CARB_B4 misfire errors (NLC_TREAT_DIAG_MIS = 0)

NLC_TREAT_DIAG_MIS = 0,
Errors *= misfire criterions (MIS_A, MIS_B1 & MIS_B4),*
Symptoms *= Cylinders identified in misfire*

NLC_TREAT_DIAG_MIS = 1,
Errors *= cylinders (MIS_x wit x = 0 … NC_CYL_NR-1),*
Symptoms *= misfire criterions (MIS_A, MIS_B1 & MIS_B4)*



Diagnosis failure class

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
#IF NLC_TREAT_DIAG_MIS = 0					
Misfire CARB A	CYL_0	1	P0300	P0301	MIL_ON_FLL / CARB_H
	CYL_1	2		P0302	
	CYL_2	3		P0303	
	CYL_3	4		P0304	
	CYL_4	5		P0305	
	CYL_5	6		P0306	
	CYL_6	7		P0307	
	CYL_7	8		P0308	
	CYL_8	9		P0309	
	CYL_9	10		P0310	
	CYL_10	11		P0311	
	CYL_11	12		P0312	
	RDN	13		P0300	
MIS_A	MPL	14		P0300	

Misfire CARB B1	CYL_0	1	P0300	P0301	MIL_ON / CARB_H
	CYL_1	2		P0302	
	CYL_2	3		P0303	
	CYL_3	4		P0304	
	CYL_4	5		P0305	
	CYL_5	6		P0306	
	CYL_6	7		P0307	
	CYL_7	8		P0308	
	CYL_8	9		P0309	
	CYL_9	10		P0310	
	CYL_10	11		P0311	
	CYL_11	12		P0312	
	RDN	13		P0300	
MIS_B1	MPL	14		P0300	
Misfire CARB B4	CYL_0	1	P0300	P0301	MIL_ON / CARB_H
	CYL_1	2		P0302	
	CYL_2	3		P0303	
	CYL_3	4		P0304	
	CYL_4	5		P0305	
	CYL_5	6		P0306	
	CYL_6	7		P0307	
	CYL_7	8		P0308	
	CYL_8	9		P0309	
	CYL_9	10		P0310	
	CYL_10	11		P0311	
	CYL_11	12		P0312	
	RDN	13		P0300	
MIS_B4	MPL	14		P0300	

There's two different ways to transmit to the Error Management (ERRM) the misfire monitoring diagnosis informations.

NLC_TREAT_DIAG_MIS = 0, Three misfire errors are defined (LV_ERR_MIS_A, LV_ERR_MIS_B1 & LV_ERR_MIS_B4) per misfire legal criterion; attached symptoms are defined per cylinders.

This case is the nominal one, described by US & Europe legal texts.

The DTC's generated provide only Cylinder identification informations (P300...P313), no informations allowing to know the criterion of the errors stored in FMY (MIS_A, MIS_B1 & MIS_B4).

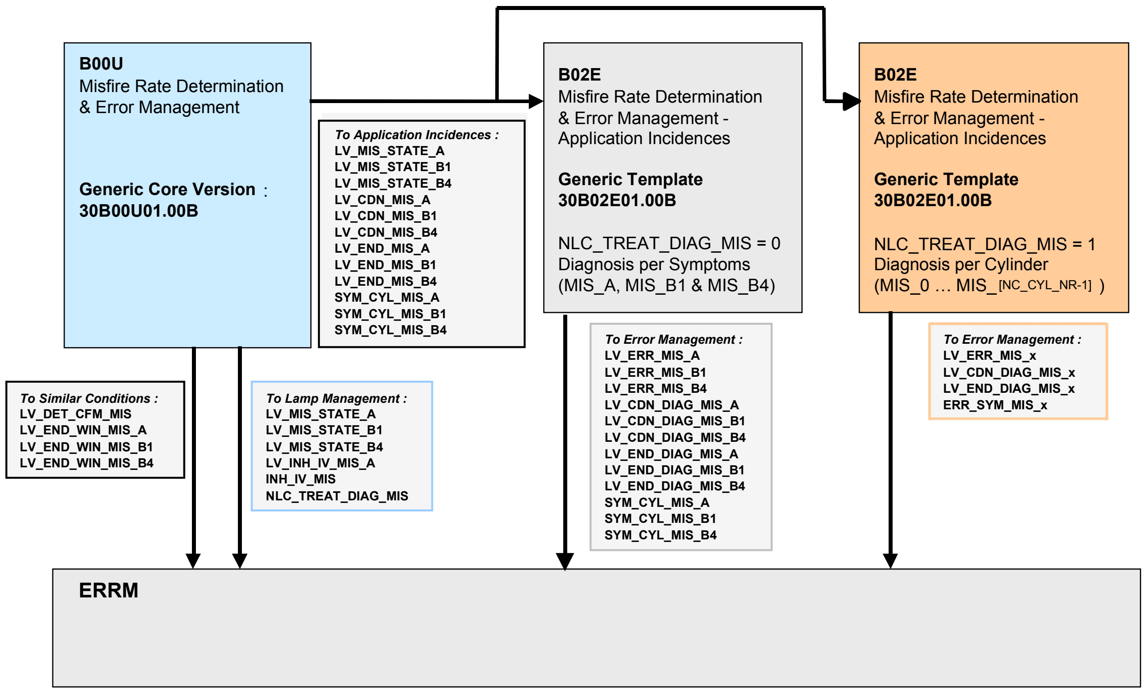
Default calibrations for DTC's of this solution *(see diagnosis failure class table above)* are exactly the codes coming from document SAE J2012 Revised March 1999 in appendix C, called Powertrain System Diagnostic Trouble Codes.

Contact our OBD co-ordinator/Customer to know if modifications occur on these Codes since 1999.

51.16.5 B02E_03 : Specific cylinder misfire errors (NLC_TREAT_DIAG_MIS = 1)

NLC_TREAT_DIAG_MIS = 0,
Errors *= misfire criterions (MIS_A, MIS_B1 & MIS_B4),*
Symptoms *= Cylinders identified in misfire*

NLC_TREAT_DIAG_MIS = 1,
Errors *= cylinders (MIS_x wit x = 0 … NC_CYL_NR-1),*
Symptoms *= misfire criterions (MIS_A, MIS_B1 & MIS_B4)*



Diagnosis failure class *(example for a 6-cylinder engine)*

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
#IF NLC_TREAT_DIAG_MIS = 1					
Misfire Cylinder 1	MIS_A	1	-	P301	MIL_ON_FLL / CARB_H
	MIS_B1	2		P321	MIL_ON / CARB_H
MIS_1	MIS_B4	3		P331	MIL_ON / CARB_H
Misfire Cylinder 2	MIS_A	1	-	P302	MIL_ON_FLL / CARB_H
	MIS_B1	2		P322	MIL_ON / CARB_H
MIS_2	MIS_B4	3		P332	MIL_ON / CARB_H
Misfire Cylinder 3	MIS_A	1	-	P303	MIL_ON_FLL / CARB_H
	MIS_B1	2		P323	MIL_ON / CARB_H
MIS_3	MIS_B4	3		P333	MIL_ON / CARB_H
Misfire Cylinder 4	MIS_A	1	-	P304	MIL_ON_FLL / CARB_H
	MIS_B1	2		P324	MIL_ON / CARB_H
MIS_4	MIS_B4	3		P334	MIL_ON / CARB_H
Misfire Cylinder 5	MIS_A	1	-	P305	MIL_ON_FLL / CARB_H
	MIS_B1	2		P325	MIL_ON / CARB_H

MIS_5	MIS_B4	3		P335	MIL_ON / CARB_H
Misfire Cylinder 6	MIS_A	1	-	P306	MIL_ON_FLL / CARB_H
	MIS_B1	2		P326	MIL_ON / CARB_H
MIS_6	MIS_B4	3		P336	MIL_ON / CARB_H

The second way to transmit the misfire monitoring diagnosis informations to the Error Management (ERRM) is NLC_TREAT_DIAG_MIS = 1,

One misfire errors is produced per cylinder (LV_ERR_MIS_0 ... LV_ERR_MIS_[NC_CYL_NR-1]); attached symptoms are defined per misfire legal criterions.

This case is a customers group solution.

The DTC's generated provide cylinder identification informations & the misfire criterion of the errors stored in FMY.

DTC's codes set in the diagnosis failure class table above, are only example; please see with your customer the extended codes that could be used of additional symptoms.

51.16.6 B02E_04 : Random/Multiple cylinder misfire failure

Diagnosis failure class

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
MIS_MPL					
Random/Multiple Misfire	MIS_MPL	1	P300	P300	NO_LAMP / NO_CARB_H
	MIS_RND	2		P300	
		3			
MIS_MPL					

For the solution using NLC_TREAT_DIAG_MIS = 1, the definition of a specific Random/Multiple cylinder misfire error is required, to be able to generate the DTC P300, when no cylinder could be identified.

51.16.7 B02E_05 : Misfire with low fuel signal active

Diagnosis failure class

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
MIS_FTL_L					
lisfire Detected with low fuel	FTL_L	1	P313	P313	NO_LAMP / NO_CARB_H
		2			
		3			
MIS_FTL_L					

A specific error can be set when a Misfire Criterion is detected (LV_MIS_STATE_x = 1) and a low fuel tank level is detected at the same time.

To allow this error, please set LC_CONF_ERR_MIS_FTL_L = 1.

Note: if LC_FTL_L_DET_MIS = 1 and a low fuel status occurs, the misfire detection is out of diagnosis (LV_CDN_DIAG_MIS_x = 0), so it will be impossible to detect misfire errors while low fuel tank level. In such case, the integration of LV_ERR_MIS_FTL_L is not necessary.

51.16.8 B02E_06 : Misfire with rough road detection failure

Diagnosis failure class

Diagnosis	Symptom	Nr	P-Code/ Failure	P-Code/ Symptom	Failure class A/B
MIS_RR					
Misfire with RR failure	MIS_RR	1	P314	P314	NO_LAMP / NO_CARB_H
		2			
		3			
MIS_RR					

A specific error can be set when a Misfire Criterion is detected (LV_MIS_STATE_x = 1) and when the Rough Road detection module is in limp-home.

This error will be set only for US applications who don't allow a misfire detection fade-out when the rough road detection function is in limp-home. So, in such case, the misfire errors will be stored but with an additional error LV_ERR_MIS_RR who informs that the misfire error could be due to a rough road module limp-home.

This error could occurs only is LC_STATE_RR_LIH = 0 in the rough road detection module (B002) *(LV_STATE_RR = 0 in limp-home)*.

51.17 Customer adaptation module: MISF (Misfire detection)

51.17.1 Outputs for BMW functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_auss_b1	V/O	0...1H	0...1	1	[-]
Bedingung 1. 1000er Intervall					
Zr_auss_a	V/O	0...FFFFH	0...65535	1	[-]
Aussetzer CARB A					
Zr_auss_b[NC_CYL_NR]	V/O	0...FFFFH	0...65535	1	[-]
Aussetzer CARB, zylinderindividuell					
Zr_auss_suma	V/O	0...FFFFH	0...65535	1	[-]
Aussetzer Summe CARB A					
Zr_auss_sumb	V/O	0...FFFFH	0...65535	1	[-]
Aussetzer Summe CARB B					
Zr_ausszyk_b4	V/O	0...FFH	0...255	1	[-]
CARB B4 Zyklen mit Aussetzern					
Zrbk_auss_a[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Aussetzer CARB A, bankweise					
Zrbkmx_auss_a[NC_CBK_EX_NR]	V/O	0...FFFFH	0...65535	1	[-]
Schwelle Aussetzer CARB A, bankweise					
Zrmx_auss_a	V/O	0...FFFFH	0...65535	1	[-]
Schwelle Aussetzer CARB A					
Zrmx_auss_b1	V/O	0...FFFFH	0...65535	1	[-]
Schwelle Aussetzer CARB B1					
Zrmx_auss_b4	V/O	0...FFFFH	0...65535	1	[-]
Schwelle Aussetzer CARB B4					

Input data:

C_MIS_SUM_B4_THD_MT	ECU_STATE	MIS_SUM_A	MIS_CTR_B[NC_CYL_NR]
MIS_SUM_A_CBK[NC_CBK_EX_NR]	C_MIS_SUM_A_CBK_THD_MT	C_MIS_SUM_A_CBK_THD_AT	C_MIS_SUM_A_THD_MT
C_MIS_SUM_A_THD_AT	C_MIS_SUM_B4_THD_MT	C_MIS_SUM_B4_THD_AT	
MIS_SUM_A	MIS_SUM_B		

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit powerlatch: 0

Recurrence : 10ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
If(1)   ECU_STATE = "PWL"
  Then(1)
        B_auss_b1 = 0
  Elseif(1)
        B_auss_b1                        = not(LV_END_MIS_B1)
        Zr_auss_a                        = MIS_SUM_A
        Zr_ausszyk_b4                    = CTR_DET_MIS_B4
        Zrbk_auss_a[NC_CBK_EX_NR]        = MIS_SUM_A_CBK[NC_CBK_EX_NR]
        Zr_auss_b[NC_CYL_NR]             = MIS_CTR_B[NC_CYL_NR]
        Zr_auss_suma                     = MIS_SUM_A
        Zr_auss_sumb                     = MIS_SUM_B
        If(2) LV_AT = 1
        Then(2)
          Zrbkmx_auss_a[NC_CBK_EX_NR] = C_MIS_SUM_A_CBK_THD_AT
          Zrmx_auss_a                 =  C_MIS_SUM_A_THD_AT
          Zrmx_auss_b1                =  C_MIS_SUM_B1_THD_AT
          Zrmx_auss_b4                =  C_MIS_SUM_B4_THD_AT
        Elseif(2)
          Zrbkmx_auss_a[NC_CBK_EX_NR] = C_MIS_SUM_A_CBK_THD_MT
          Zrmx_auss_a                 =  C_MIS_SUM_A_THD_MT
          Zrmx_auss_b1                =  C_MIS_SUM_B1_THD_MT
          Zrmx_auss_b4                =  C_MIS_SUM_B4_THD_MT
        Endif(2)
Endif(1)
```

51.17.2 Outputs for SV MISF Aggregate

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ER_CLC_MIS_DET_CUS_SPC	O/V	0...1H	0...1	1	[-]
Enable engine roughness and misfire detection based on customer specific algorithm					
MIS_STATE_CUS_SPC	O/V	0...FFH	0...255	1	[-]
Status carrier byte of actual detected misfire through customer specific algorithm					
MIS_DET_ENA_CUS_SPC	O/V	0...FFH	0...255	1	[-]
Misfire detection enabling condition based on customer specific algorithm, Carrier structure					

Input data:

St_aekh	St_aekh_ae	B_aekh_akt	

FUNCTION DESCRIPTION:

Adaptation to BMW environment

Application conditions:

Initialisation: at reset or at exit powerlatch: *0*

Recurrence: Segment

Activation: every engine state

Formula section:

LV_ER_CLC_MIS_DET_CUS_SPC = B_aekh_akt

MIS_STATE_CUS_SPC = St_aekh_ae

MIS_DET_ENA_CUS_SPC = St_aekh

Important Note for the calling sequence :

51.17.3 The following calling sequence should be maintained only for the segment task which

51.17.4 calculates outputs for SV MISF Aggregate. That is the segment task for this function

51.17.5 should be called only after the following calculation

1) Misfire calculation functions from SV.
2) VRAMO – Misfire calcuation from BMW

   After VRAMO, copying BMW variables to SV variables is done, and hence used in next segment raster of SV misfire functions.

52 NOx determination

Table of Contents

Index

52.1 NOXD Configuration Data

FUNCTION DESCRIPTION:

General information:

The following describes the general rules for determination of the configuration data:

- All configuration data shall be defined by project with information from customer.

Formula section:

Global configuration data

Here are listed the configuration data, which can be used in other aggregates:

Data	Value
NC_NOX_SENS_CONF	1

Local configuration data

Here are listed the configuration data, which are used only in the NOXD aggregate:

Data	Value
-	

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NOX_SENS_CONF	-	0...2H	0...2	1	[-]
Number of NOx sensors within the exhaust system					

52.2 AGGR adaptation: NOXD

FUNCTION DESCRIPTION:

General information:

This module handles all variable interfaces of aggregate NOXD. It adapts inputs to aggregate NOXD and outputs from aggregate NOXD on project specific requirements or aggregate structure requirements.

52.2.1 Inputs to aggregate NOXD – 10ms (fast variable assignments)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_NOX_SENS_CAN_BOFF	V/O	0...1H	0...1	1	[-]
CAN bus, at which the NOx sensor is connected, has the state "busoff"					
LV_ERR_NS_CAN_BOFF	V/O	0...1H	0...1	1	[-]
CAN bus, at which the NOx sensor is connected, has the state "busoff"					
LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor CAN message lost failure is present					
~~T_NOX_NS_MDL[NC_NOX_SENS_CONF]~~	~~–~~	~~0...FFFFH~~	~~0...655.35~~	~~0.01~~	~~[s]~~
~~Timer for debounce of signal state "valid" after re-entry into state "valid"~~					

Input data:

NC_NOX_SENS_CONF	LV_ERR_LOCAN_BOFF	LV_ERR_TOUT_NOX_SENS[NC_NOX_SENS_CONF]	~~T_NOX_SENS_NOX_MDL[NC_NOX_SENS_CONF]~~

Application conditions:

Initialisation: at reset:

LV_ERR_NOX_SENS_CAN_BOFF = 0
LV_ERR_NS_CAN_BOFF = 0
LV_ERR_NS_CAN_MSG_LOST[i] = 0
~~T_NOX_NS_MDL[i] = 0~~

Recurrence: 10ms

Activation: after reset

Deactivation: -

Formula section:

LV_ERR_NS_CAN_BOFF = LV_ERR_LOCAN_BOFF

LV_ERR_NS_CAN_MSG_LOST[i] = LV_ERR_TOUT_NOX_SENS[i]

LV_ERR_NOX_SENS_CAN_BOFF = LV_ERR_LOCAN_BOFF

~~T_NOX_NS_MDL[i] = T_NOX_SENS_NOX_MDL[i]~~

52.2.2 Outputs of aggregate NOXD – 10ms

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TEMP_NS_OK[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor CAN message: temperature is OK					
LV_VB_NS_OK[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor CAN message: power supply voltage is OK					
LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the NOx signal has reached the valid state after the start					
LV_TEMP_MIN_THD_CAN[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Dew point recognition for NOx-Sensor					
LV_VLS_NS_DIAG_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the binary O2 signal is valid for use into the diagnosis functions					
LV_LAMB_NS_DIAG_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the Lambda signal is valid for use into the diagnosis functions					
LV_LAMB_NS_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the Lambda signal is valid					
LV_VLS_NS_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the binary O2 signal is valid					
NOX_NS[NC_NOX_SENS_CONF]	V/O	FF9C...05DCH	-100...1500	1	[ppm]
NOx concentration value, measured by NOx-Sensor					
NOX_NS_DIAG[NC_NOX_SENS_CONF]	V/O	FF9C...05DCH	-100...1500	1	[ppm]
NOx concentration value for diagnosis functions, measured by NOx sensor					
LAMB_NS[NC_NOX_SENS_CONF]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda value, measured by NOx-Sensor					
VLS_NS[NC_NOX_SENS_CONF]	V/O	0...578H	-200...1200	1	[mV]
Binary O2 signal voltage for diagnosis functions, raw value, measured by NOx-Sensor					
LV_NOX_NS_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the NOx signal is valid					
LAMB_NS_DIAG[NC_NOX_SENS_CONF]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda value for diagnosis functions, measured by NOx-Sensor					
T_LAMB_NS_HLD[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...655.35	0.01	[s]
Timer for hold of last valid Lambda signal value					
T_NOX_NS_HLD[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...655.35	0.01	[s]
Timer for hold of last valid NOx signal value					
VLS_NS_DIAG[NC_NOX_SENS_CONF]	V/O	0...578H	-200...1200	1	[mV]
Binary O2 signal voltage for diagnosis functions, raw value, measured by NOx-Sensor					
T_VLS_NS_HLD[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...655.35	0.01	[s]
Timer for hold of last valid binary O2 signal value					
LV_NOX_NS_DIAG_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the NOx signal is valid for use into the diagnosis functions					
LV_NOX_NS_MDL_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the NOx signal is valid for use. The state "valid" is debounced present.					
T_TEMP_NS[NC_NOX_SENS_CONF]	V	0...FFFFH	0...655.35	0.01	[s]
Timer for debounce of NOx sensor temperature readiness					
LV_LAMB_NS_AUTH[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the Lambda signal has be reached the valid state after start					
LV_VLS_NS_AUTH[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the binary O2 signal has be reached the valid state after start					
T_NOX_NS_MDL[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...655.35	0.01	[s]

Timer for debounce of signal state "valid" after re-entry into state "valid"

Input data:

NOX_SENS_TEMP[NC_NOX_SENS_CONF]	LV_NOX_SENS_NOX_AUTH[NC_NOX_SENS_CONF]	LV_CAN_TEMP_MIN_THD[NC_NOX_SENS_CONF]	LV_NOX_SENS_VLS_DIAG_VLD[NC_NOX_SENS_CONF]
LV_NOX_SENS_LAMB_DIAG_VLD[NC_NOX_SENS_CONF]	LV_NOX_SENS_LAMB_VLD[NC_NOX_SENS_CONF]	LV_NOX_SENS_VLS_VLD[NC_NOX_SENS_CONF]	NOX_NOX_SENS[NC_NOX_SENS_CONF]
NOX_NOX_SENS_DIAG[NC_NOX_SENS_CONF]	LAMB_NOX_SENS[NC_NOX_SENS_CONF]	VLS_NOX_SENS[NC_NOX_SENS_CONF]	LV_NOX_SENS_NOX_VLD[NC_NOX_SENS_CONF]
LAMB_NOX_SENS_DIAG[NC_NOX_SENS_CONF]	T_NOX_SENS_LAMB_HLD[NC_NOX_SENS_CONF]	T_NOX_SENS_NOX_HLD[NC_NOX_SENS_CONF]	VLS_NOX_SENS_DIAG[NC_NOX_SENS_CONF]
T_NOX_SENS_VLS_HLD[NC_NOX_SENS_CONF]	LV_NOX_SENS_NOX_DIAG_VLD[NC_NOX_SENS_CONF]	LV_NOX_SENS_NOX_MDL_VLD[NC_NOX_SENS_CONF]	LV_NOX_SENS_LAMB_AUTH[NC_NOX_SENS_CONF]
LV_NOX_SENS_VLS_AUTH[NC_NOX_SENS_CONF]	T_NOX_SENS_NOX_MDL[NC_NOX_SENS_CONF]		

Application conditions:

Initialisation: at reset:

LV_TEMP_NS_OK[i] = 0
LV_VB_NS_OK[i] = 0
LV_NOX_NS_AUTH[i] = 0
LV_TEMP_MIN_THD_CAN[i] = 0
LV_VLS_NS_DIAG_VLD[i] = 0
LV_LAMB_NS_DIAG_VLD[i] = 0
LV_LAMB_NS_VLD[i] = 0
LV_VLS_NS_VLD[i] = 0
NOX_NS[i] = 0
NOX_NS_DIAG[i] = 0
LAMB_NS[i] = 0
VLS_NS[i] = 0
LV_NOX_NS_VLD[i] = 0
LAMB_NS_DIAG[i] = 0
T_LAMB_NS_HLD[i] = 0
T_NOX_NS_HLD[i] = 0
VLS_NS_DIAG[i] = 0
T_VLS_NS_HLD[i] = 0
LV_NOX_NS_DIAG_VLD[i] = 0
LV_NOX_NS_MDL_VLD[i] = 0
T_TEMP_NS[i] = 0
LV_LAMB_NS_AUTH[i] = 0

LV_VLS_NS_AUTH[i] = 0

T_NOX_NS_MDL[i] = 0

Recurrence: 10ms

Activation: after reset

Deactivation: -

Formula section:

```
LV_TEMP_MIN_THD_CAN[i] = LV_CAN_TEMP_MIN_THD[i]

if      bit0( NOX_SENS_TEMP[i] ) = 1
then    if      T_TEMP_NS[i] >= C_T_TEMP_NS
        then    LV_TEMP_NS_OK[i] = 1
        else    LV_TEMP_NS_OK[i] = 0
                increment T_TEMP_NS[i]
        endif
else    LV_TEMP_NS_OK[i] = 0
        T_TEMP_NS[i] = 0
endif

LV_VB_NS_OK[i] = bit1( NOX_SENS_TEMP[i] )

NOX_NS[i] = NOX_NOX_SENS[i]
NOX_NS_DIAG[i] = NOX_NOX_SENS_DIAG[i]
T_NOX_NS_HLD[i] = T_NOX_SENS_NOX_HLD[i]
LV_NOX_NS_VLD[i] = LV_NOX_SENS_NOX_VLD[i]
LV_NOX_NS_DIAG_VLD[i] = LV_NOX_SENS_NOX_DIAG_VLD[i]
LV_NOX_NS_AUTH[i] = LV_NOX_SENS_NOX_AUTH[i]
T_NOX_NS_MDL[i] = T_NOX_SENS_NOX_MDL[i]

LAMB_NS[i] = LAMB_NOX_SENS[i]
LAMB_NS_DIAG[i] = LAMB_NOX_SENS_DIAG[i]
T_LAMB_NS_HLD[i] = T_NOX_SENS_LAMB_HLD[i]
```

LV_LAMB_NS_VLD[i] = LV_NOX_SENS_LAMB_VLD[i]
LV_LAMB_NS_DIAG_VLD[i] = LV_NOX_SENS_LAMB_DIAG_VLD[i]
LV_LAMB_NS_AUTH[i] = LV_NOX_SENS_LAMB_AUTH[i]

VLS_NS[i] = VLS_NOX_SENS[i]
VLS_NS_DIAG[i] = VLS_NOX_SENS_DIAG[i]
T_VLS_NS_HLD[i] = T_NOX_SENS_VLS_HLD[i]
LV_VLS_NS_VLD[i] = LV_NOX_SENS_VLS_VLD[i]
LV_VLS_NS_DIAG_VLD[i] = LV_NOX_SENS_VLS_DIAG_VLD[i]
LV_VLS_NS_AUTH[i] = LV_NOX_SENS_VLS_AUTH[i]

LV_NOX_NS_MDL_VLD[i] = LV_NOX_SENS_NOX_MDL_VLD[i]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_TEMP_NS	1	0...FFFFH	0...655.35	0.01	[s]
Minimum time of sensor temperature readiness to set LV_TEMP_NS_OK[i]					

52.3 NOx sensor Interface

Input data:

LV_IGK			

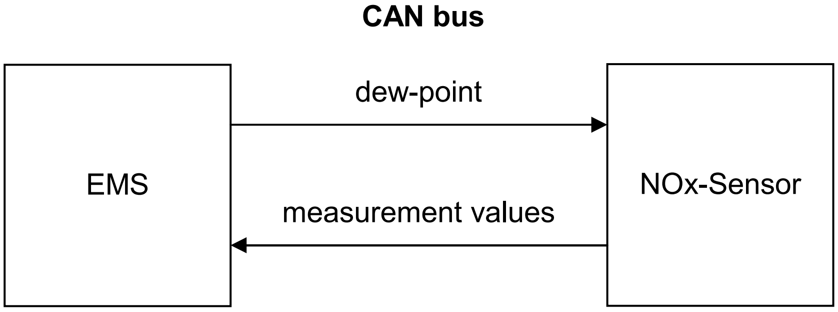
FUNCTION DESCRIPTION:

General information:

The number of NOx sensors in the system is defined by NC_NOX_SENS_CONF. The value of NC_NOX_SENS_CONF is to set within the project specific module ‘List of Configuration Data’.

```
if    NC_NOX_SENS_CONF = 0
then  the system contains no NOx sensor
      nothing is to calculate, no NOx sensor variables are declared
else  if    NC_NOX_SENS_CONF = 1
      then  the system contains 1 NOx sensor
            all NOx sensor variables are xxx_1 (i = 1)
      else  if    NC_NOX_SENS_CONF = 2
            then  the system contains 2 NOx sensors
                  all NOx sensor variables for exhaust line 1 are xxx_1 (i = 1)
                  all NOx sensor variables for exhaust line 2 are xxx_2 (i = 2)
            endif
      endif
endif
```

The NOx sensor Interface handles the variables and CAN messages for the use of NOx sensor into the EMS.



The EMS sends the dew point to the NOx sensor and receives measurement values and sensor status information from the NOx sensor via CAN bus. The CAN message structure and the CAN message ID’s are defined project specific, e.g. at module ‘NOx sensor Interface (Application Incidences)’.
The dew point handling is also performed in a project specific module, e.g. at module ‘NOx sensor Interface (Application Incidences)’.

Description:

The NOx sensor sends messages with actual measurement values, sensor status and a failure byte to the EMS.

The information from the NOx sensor status byte are combined with the CAN bus failure information's to the signal valid indicators. These CAN bus failures are lost message from the NOx sensor and the state CAN busoff.

Application conditions:

Recurrence: 10 ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

52.3.1 NOx concentration signal

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
NOX_NOX_SENS_DIAG[NC_NOX_SENS_CONF]	V/O	FF9C...05DCH	-100...1500	1	[ppm]
NOx concentration value for diagnosis functions, measured by NOx-Sensor					
NOX_NOX_SENS_DIAG_H_RES[NC_NOX_SENS_CONF]	-	FF9C0000H...05DC0000H	-100...1500	1/0x10000	ppm
32bit variable of NOX_NOX_SENS_DIAG_i for internal use only					
NOX_NOX_SENS_H_RES[NC_NOX_SENS_CONF]	-	FF9C0000H...05DC0000H	-100...1500	1/0x10000	ppm
32bit variable of NOX_NOX_SENS_i for internal use only					
LV_NOX_SENS_NOX_DIAG_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the NOx signal is valid for use into the diagnosis functions					
NOX_NOX_SENS[NC_NOX_SENS_CONF]	V/O	FF9C...05DCH	-100...1500	1	[ppm]
NOx concentration value, measured by NOx-Sensor					
LV_NOX_SENS_NOX_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the NOx signal is valid					
LV_NOX_SENS_NOX_AUTH[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the NOx signal has be reached the valid state after the start					
T_NOX_SENS_NOX_HLD[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...655.35	0.01	[s]
Timer for hold of last valid NOx signal value					
LV_NOX_SENS_NOX_MDL_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the NOx signal is valid for use. The state "valid" is debounced present.					
T_NOX_SENS_NOX_MDL[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...655.35	0.01	[s]
Timer for debounce of signal state "valid" after re-entry into state "valid"					
T_NOX_NS_OSC[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...655.35	0.01	[s]
Time since last reason for an oscillation of NOx signal was detected					
LAMB_NS_DIAG_GRD_OSC[NC_NOX_SENS_CONF]	V	8000...7FFFH	-32...31.99902	0.9766e-3	[-]
Gradient of NOx sensor lambda signal for detection of NOx signal oscillations					
LAMB_NS_DIAG_OLD_OSC[NC_NOX_SENS_CONF]	-	0...7FFFH	0...31.99902	0.9766e-3	[-]
Old value of NOx sensor lambda signal for detection of NOx signal oscillations					
STATE_NOX_OLD_OSC	-	0H 1H 2H 3H 4H 5H	PASSIV LOAD REGENERATION WAIT STOP WARMUP	1	[-]
Old value of NOx catalyst state for detection of NOx signal oscillations					
STATE_ENG_OLD_OSC	-	0H 1H 2H 3H 4H 5H	ES ST IS PL PU PUC	1	[-]
Old value of engine operating state for detection of NOx signal oscillations					

Input data:

CAN_NOX_NOX_SENS[NC_NOX_SENS_CONF]	CAN_STATE_NOX_SENS[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_BOFF
LV_IGK	NOX_SENS_TEMP[NC_NOX_SENS_CONF]	LV_CAN_TEMP_MIN_THD[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
STATE_NOX	STATE_ENG	LAMB_NS_DIAG[NC_NOX_SENS_CONF]	CAN_STATE_NS_TMP[NC_NOX_SENS_CONF]
C_STATE_NS_MASK_MODE_2			

FUNCTION DESCRIPTION:

General information:

The EMS receives the NOx concentration signal from the NOx sensor by CAN. After that the NOx concentration signal is evaluated by NOx sensor status information and filtered within the EMS.

Description:

The NOx output signal NOX_NOX_SENS_DIAG_i is used for the NOx sensor diagnosis functions. It is generated by filter of CAN_NOX_NOX_SENS_i, which is received from the NOx sensor.

The NOx output signal NOX_NOX_SENS_i is used for the NOx catalyst control functions. It is generated by filter of CAN_NOX_NOX_SENS_i after an analysis of sensor status information CAN_STATE_NOX_SENS_i. A new NOX_NOX_SENS_i value is calculated if no CAN bus error is occurred and CAN_STATE_NOX_SENS_i indicates full sensor readiness by CAN_STATE_NOX_SENS_i = 1EH. All other bits of CAN_STATE_NOX_SENS_i have no relevance for full readiness. So the system is authorised to use the NOx signal.

The bit LV_NOX_SENS_NOX_AUTH_i is set to 1 after the NOx sensor was uninterrupted full ready for at least C_T_NOX_SENS_NOX_AUTH. LV_NOX_SENS_NOX_AUTH_i is reset to 0 when the temperature of NOx sensor element is out of control range and the power supply of NOx sensor is out of specified range (bit PWR_OK of NOX_SENS_TEMP_i = 0) or when the engine is stopped or when the NOx sensor dew point is reset. This is necessary in order to deactivate the NOx sensor OBD II diagnosis functions.

The variable LV_NOX_SENS_NOX_DIAG_VLD_i is reset to 0 every time when the sensor loses the full readiness. So the diagnosis functions are informed in order to exclude a wrong diagnosis. At the same time starts the timer T_NOX_SENS_NOX_HLD_i. It runs up to the maximum value. An overflow is not allowed. The timer is stopped and reset, when the sensor reaches its full readiness.

The variable LV_NOX_SENS_NOX_MDL_VLD_i is used in the module ‘NOx catalyst model controller’. The variable LV_NOX_SENS_NOX_MDL_VLD_i informs this module about the NOx signal state. It is reset to 0 every time when the sensor loses the full readiness. It is set to 1 after the NOx signal was valid for the time C_T_NOX_SENS_NOX_MDL.

Note! The timer T_NOX_SENS_NOX_MDL_i runs up to the maximum value. An overflow is not allowed.

If the permanently authorisation of NOx signal is activated by LC_NOX_SENS_NOX_VLD_i = 1 and LV_NOX_SENS_NOX_AUTH_i = 1 then it is allowed to calculate a new NOX_NOX_SENS_i value without full sensor readiness. The variable LV_NOX_SENS_NOX_VLD_i is set to 1 so that the system is authorised to use the NOx signal. After the engine start the permanently authorisation of NOx signal is blocked up to the use of sensor signals is allowed by LV_NOX_SENS_NOX_AUTH_i = 1. Then the calculation of NOX_NOX_SENS_i values is started.

When the sensor loses its full readiness the last valid NOx signal value NOX_NOX_SENS_i and LV_NOX_SENS_NOX_VLD_i are held up to the time C_T_NOX_SENS_NOX_HLD is passed. After that the NOx signal NOX_NOX_SENS_i is set to C_NOX_NOX_SENS_i and LV_NOX_SENS_NOX_VLD_i is reset.

If the manual setting of NOx values is active LC_NOX_SENS_NOX_MAN_i = 1 then NOX_NOX_SENS_DIAG_i and NOX_NOX_SENS_i are set to C_NOX_NOX_SENS_i, LV_NOX_SENS_NOX_VLD_i is set to 1 and LV_NOX_SENS_NOX_DIAG_VLD_i is reset.

The calculation of NOX_NOX_SENS_i and NOX_NOX_SENS_DIAG_i shall be done with an internal resolution of 32 bit.

Application conditions:

Initialisation: at reset, at LV_IGK = 1 -> 0:

NOX_NOX_SENS_i = C_NOX_NOX_SENS_i
NOX_NOX_SENS_DIAG_i = C_NOX_NOX_SENS_i
LV_NOX_SENS_NOX_VLD_i = 0
LV_NOX_SENS_NOX_DIAG_VLD_i = 0
LV_NOX_SENS_NOX_AUTH_i = 0
LV_NOX_SENS_NOX_MDL_VLD_i = 0
T_NOX_SENS_NOX_HLD_i = 0
T_NOX_SENS_NOX_MDL_i = 0

Formula section:

```
If(1)  LC_NOX_SENS_NOX_MAN_i = 0                              AND
       LV_CAN_TEMP_MIN_THD_i = 1                              AND
       CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_2
then(1)
       if(2)  LV_ERR_NS_CAN_MSG_LOST[i] = 0                    AND
              LV_ERR_NS_CAN_BOFF = 0                           AND
              ( CAN_STATE_NOX_SENS[i] bitwise AND
                    C_STATE_NOX_NS_MASK ) = C_STATE_NOX_NS_MASK
              ~~( Bit B1 of CAN_STATE_NOX_SENS_i = 1          AND~~
              ~~  Bit B2 of CAN_STATE_NOX_SENS_i = 1          AND~~
              ~~  Bit B3 of CAN_STATE_NOX_SENS_i = 1          AND~~
              ~~  Bit B4 of CAN_STATE_NOX_SENS_i = 1          AND~~
              ~~  Bit B6 of CAN_STATE_NOX_SENS_i = 0          AND~~
              ~~  Bit B7 of CAN_STATE_NOX_SENS_i = 0 )~~
       then(2)
              if(3)  LV_NOX_SENS_NOX_AUTH_i = 1
              then(3) T_NOX_SENS_NOX_HLD_i = 0
                      LV_NOX_SENS_NOX_DIAG_VLD_i = 1
                      LV_NOX_SENS_NOX_VLD_i = 1

                      increment T_NOX_SENS_NOX_MDL_i

                      if(4)   T_NOX_SENS_NOX_MDL_i >= C_T_NOX_SENS_NOX_MDL
                      then(4) LV_NOX_SENS_NOX_MDL_VLD_i = 1
                      endif(4)
```

```
            else(3)  increment T_NOX_SENS_NOX_HLD_i

                     if(5)  T_NOX_SENS_NOX_HLD_i >= C_T_NOX_SENS_NOX_AUTH

                     then(5)  LV_NOX_SENS_NOX_AUTH_i = 1
                              LV_NOX_SENS_NOX_VLD_i = 1
                              LV_NOX_SENS_NOX_DIAG_VLD_i = 1

                     endif(5)

            endif(3)

            if(6)  LV_NOX_SENS_NOX_DIAG_VLD_i(n-1) = 0  AND
                   LV_NOX_SENS_NOX_DIAG_VLD_i(n) = 1

            then(6)  NOX_NOX_SENS_i(n) =
                       CAN_NOX_NOX_SENS_i * C_CRLC_NOX_NOX_SENS +
                       NOX_NOX_SENS_DIAG_i(n-1) * (1 - C_CRLC_NOX_NOX_SENS)

            else(6)  NOX_NOX_SENS_i(n) =
                       CAN_NOX_NOX_SENS_i * C_CRLC_NOX_NOX_SENS +
                       NOX_NOX_SENS_i(n-1) * (1 - C_CRLC_NOX_NOX_SENS)

            endif(6)

   else(2)  LV_NOX_SENS_NOX_DIAG_VLD_i = 0
            T_NOX_SENS_NOX_MDL_i = 0
            LV_NOX_SENS_NOX_MDL_VLD_i = 0

            if(7)  LV_NOX_SENS_NOX_AUTH_i = 1

            then(7)  increment T_NOX_SENS_NOX_HLD_i

            else(7)  T_NOX_SENS_NOX_HLD_i = 0
                     NOX_NOX_SENS_i = C_NOX_NOX_SENS_i
                     LV_NOX_SENS_NOX_VLD_i = 0

            endif(7)

            if(8)  LC_NOX_SENS_NOX_VLD_i = 1              AND
                   LV_NOX_SENS_NOX_AUTH_i = 1

            then(8)  NOX_NOX_SENS_i(n) =
                       CAN_NOX_NOX_SENS_i * C_CRLC_NOX_NOX_SENS +
                       NOX_NOX_SENS_i(n-1) * (1 - C_CRLC_NOX_NOX_SENS)
                     LV_NOX_SENS_NOX_VLD_i = 1

            else(8)

                     if(9)  T_NOX_SENS_NOX_HLD_i >= C_T_NOX_SENS_NOX_HLD

                     then(9)  NOX_NOX_SENS_i = C_NOX_NOX_SENS_i
                              LV_NOX_SENS_NOX_VLD_i = 0

                     endif(9)

            endif(8)

   endif(2)

   if(10) ( LV_NOX_SENS_NOX_AUTH_i = 1                  AND
            LV_NOX_SENS_NOX_DIAG_VLD_i = 0              AND
            Bit PWR_OK of NOX_SENS_TEMP_i = 0 )
```

```
        then(10)  NOX_NOX_SENS_i = C_NOX_NOX_SENS_i
                  NOX_NOX_SENS_DIAG_i = C_NOX_NOX_SENS_i
                  LV_NOX_SENS_NOX_VLD_i = LC_NOX_SENS_NOX_MAN_i
                  LV_NOX_SENS_NOX_DIAG_VLD_i = 0
                  LV_NOX_SENS_NOX_AUTH_i = LC_NOX_SENS_NOX_MAN_i
                  LV_NOX_SENS_NOX_MDL_VLD_i = 0
                  T_NOX_SENS_NOX_HLD_i = 0
                  T_NOX_SENS_NOX_MDL_i = 0

        else(10)  NOX_NOX_SENS_DIAG_i(n) =
                    CAN_NOX_NOX_SENS_i * C_CRLC_NOX_NOX_SENS_DIAG +
                    NOX_NOX_SENS_DIAG_i(n-1) * (1 - C_CRLC_NOX_NOX_SENS_DIAG)

        endif(10)

else(1) NOX_NOX_SENS_i = C_NOX_NOX_SENS_i
        NOX_NOX_SENS_DIAG_i = C_NOX_NOX_SENS_i
        LV_NOX_SENS_NOX_VLD_i = LC_NOX_SENS_NOX_MAN_i
        LV_NOX_SENS_NOX_DIAG_VLD_i = 0
        LV_NOX_SENS_NOX_AUTH_i = LC_NOX_SENS_NOX_MAN_i
        LV_NOX_SENS_NOX_MDL_VLD_i = 0
        T_NOX_SENS_NOX_HLD_i = 0
        T_NOX_SENS_NOX_MDL_i = 0

endif(1)
```

Prediction of NOx signal oscillations

```
LAMB_NS_DIAG_GRD_OSC_i = LAMB_NS_DIAG_i – LAMB_NS_DIAG_OLD_OSC_i


If(11)      ( LC_STATE_NOX_CHG_OSC                          AND
              STATE_NOX != STATE_NOX_OLD_OSC )                        OR

            ( LC_STATE_ENG_CHG_OSC                          AND
            STATE_ENG != STATE_ENG_OLD_OSC )                          OR

            LAMB_NS_DIAG_GRD_OSC_i > C_LAMB_NS_GRD_MAX_OSC            OR

            LAMB_NS_DIAG_GRD_OSC_i < C_LAMB_NS_GRD_MIN_OSC

Then(11)    T_NOX_NS_OSC_i = 0

Else(11)    increment T_NOX_NS_OSC_i                        (10ms)

Endif(11)


LAMB_NS_DIAG_OLD_OSC_i = LAMB_NS_DIAG_i

STATE_NOX_OLD_OSC = STATE_NOX

STATE_ENG_OLD_OSC = STATE_ENG
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NOX_SENS_NOX_MAN[NC_NOX_SENS_CONF]	1	0...1H	0...1	1	[-]
Manual setting of NOx concentration value on/ off					
C_NOX_NOX_SENS[NC_NOX_SENS_CONF]	1	FF9C...05DCH	-100...1500	1	[ppm]
Manual NOx concentration value					
LC_NOX_SENS_NOX_VLD[NC_NOX_SENS_CONF]	1	0...1H	0...1	1	[-]
Permanently authorisation NOx signal on/ off					
C_T_NOX_SENS_NOX_HLD	1	0...FFFFH	0...655.35	0.01	[s]
Maximum time of hold the last valid NOx signal value					
C_CRLC_NOX_NOX_SENS	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for 1st NOx signal filter					
C_CRLC_NOX_NOX_SENS_DIAG	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for 2nd NOx signal filter					
C_T_NOX_SENS_NOX_AUTH	1	0...FFFFH	0...655.35	0.01	[s]
Minimum time of full NOx sensor readiness to set the NOx signal authorisation bit					
C_T_NOX_SENS_NOX_MDL	1	0...FFFFH	0...655.35	0.01	[s]
Minimum NOx signal "valid" time for set of model "valid" bit					
LC_STATE_NOX_CHG_OSC	1	0...1H	0...1	1	[-]
Enable observation of changes of STATE_NOX for detection of NOx signal oscillations					
LC_STATE_ENG_CHG_OSC	1	0...1H	0...1	1	[-]
Enable observation of changes of STATE_ENG for detection of NOx signal oscillations					
C_LAMB_NS_GRD_MAX_OSC	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
NOx signal oscillations will be detected if gradient of NOx sensor lambda signal is higher than this value					
C_LAMB_NS_GRD_MIN_OSC	1	8000...0H	-32...0	0.9766e-3	[-]
NOx signal oscillations will be detected if gradient of NOx sensor lambda signal is lower than this negative value					
C_STATE_NOX_NS_MASK	1	0...FFH	0...255	1	[-]
Bit-mask for detection of valid NOx signal from NOx sensor					

52.3.2 Lambda signal

Output data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LAMB_NOX_SENS_DIAG[NC_NOX_SENS_CONF]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda value for diagnosis functions, measured by NOx sensor					
LV_NOX_SENS_LAMB_DIAG_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the Lambda signal is valid for use into the diagnosis functions					
LAMB_NOX_SENS[NC_NOX_SENS_CONF]	V/O	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda value, measured by NOx sensor					
LV_NOX_SENS_LAMB_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the Lambda signal is valid					
LV_NOX_SENS_LAMB_AUTH[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the Lambda signal has be reached the valid state after start					
T_NOX_SENS_LAMB_HLD[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...655.35	0.01	[s]
Timer for hold of last valid Lambda signal value					

Input data:

CAN_LAMB_NOX_SENS[NC_NOX_SENS_CONF]	CAN_STATE_NOX_SENS[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_BOFF
LV_IGK	LV_CAN_TEMP_MIN_THD[NC_NOX_SENS_CONF]		NOX_SENS_TEMP[NC_NOX_SENS_CONF]
NC_NOX_SENS_CONF	CAN_STATE_NS_TMP[NC_NOX_SENS_CONF]	C_STATE_NS_MASK_MODE_1	C_STATE_NS_MASK_MODE_2
C_STATE_NS_MASK_MODE_5			

FUNCTION DESCRIPTION:

General information:

The EMS receives the Lambda signal from the NOx sensor by CAN. The Lambda signal is issued in the kind of 1000/ λ. The 1000/ λ value was chosen for an easy data transmission by CAN.

Description:

The Lambda signal CAN_LAMB_NOX_SENS_i, which is received from the NOx sensor, is converted from 1000/ λ value to λ and then it is stored at LAMB_NOX_SENS_DIAG_i. After that the Lambda signal is evaluated by the sensor status CAN_STATE_NOX_SENS_i.

The value of LAMB_NOX_SENS_DIAG_i is copied to LAMB_NOX_SENS_i if no CAN bus error is occurred and CAN_STATE_NOX_SENS_i indicates at least level 2 sensor readiness by CAN_STATE_NOX_SENS_i = 0EH. For level 2 sensor readiness must be checked the bits B1, B2, B3 and B6 of CAN_STATE_NOX_SENS_i. All other bits of CAN_STATE_NOX_SENS_i have no relevance for level 2 readiness. At the same time the variables LV_NOX_SENS_LAMB_VLD_i and LV_NOX_SENS_LAMB_DIAG_VLD_i are set to 1. So the system is authorised to use the Lambda signal.

The variable LV_NOX_SENS_LAMB_DIAG_VLD_i is reset to 0 every time when the sensor loses the level 2 readiness. So the diagnosis functions are informed in order to exclude a wrong diagnosis. At the same time starts the timer T_NOX_SENS_LAMB_HLD_i. It runs up to the maximum value. An overflow is not allowed. The timer is stopped and reset, when the sensor reaches the level 2 readiness.

If the permanently authorisation of Lambda signal is activated by LC_NOX_SENS_LAMB_VLD_i = 1 and LV_NOX_SENS_NOX_AUTH_i = 1 then it is allowed to pass the Lambda signal to LAMB_NOX_SENS_i without level 2 sensor readiness. The variable LV_NOX_SENS_LAMB_VLD_i is set to 1 so that the system is authorised to use the Lambda signal. After the engine start the permanently authorisation of Lambda signal is blocked up to LV_NOX_SENS_LAMB_AUTH_i = 1. Then the Lambda signal values are passed to LAMB_NOX_SENS_i.

When the sensor loses the level 2 readiness the last valid Lambda signal value LAMB_NOX_SENS_i and LV_NOX_SENS_LAMB_VLD_i are held up to the time C_T_NOX_SENS_LAMB_HLD is passed. After that the Lambda signal LAMB_NOX_SENS_i is set to C_LAMB_NOX_SENS_i and LV_NOX_SENS_LAMB_VLD_i is reset.

If the manual setting of Lambda values is active (LC_NOX_SENS_LAMB_MAN_i = 1) then LAMB_NOX_SENS_DIAG_i and LAMB_NOX_SENS_i are set to C_LAMB_NOX_SENS_i, LV_NOX_SENS_LAMB_VLD_i is set to 1 and LV_NOX_SENS_LAMB_DIAG_VLD_i is reset.

<u>Application conditions:</u>

Initialisation: at reset, at LV_IGK = 1 -> 0:

LAMB_NOX_SENS_DIAG_i = C_LAMB_NOX_SENS_i
LAMB_NOX_SENS_i = C_LAMB_NOX_SENS_i
LV_NOX_SENS_LAMB_VLD_i = 0
LV_NOX_SENS_LAMB_DIAG_VLD_i = 0
LV_NOX_SENS_LAMB_AUTH_i = 0
T_NOX_SENS_LAMB_HLD_i = 0

<u>Formula section:</u>

```
if(1)  LC_NOX_SENS_LAMB_MAN_i = 0                                  AND
       LV_CAN_TEMP_MIN_THD_i = 1                                   AND
       ( CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_1        OR
         CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_2        OR
         CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_5 )
then(1)  LAMB_NOX_SENS_DIAG_i =  1000 / CAN_LAMB_NOX_SENS_i

      if(2)  LV_ERR_NS_CAN_MSG_LOST[i] = 0                         AND
             LV_ERR_NS_CAN_BOFF = 0                                AND
             ( CAN_STATE_NOX_SENS[i] bitwise AND
                   C_STATE_LAMB_NS_MASK ) = C_STATE_LAMB_NS_MASK
             ~~( Bit B1 of CAN_STATE_NOX_SENS_i = 1            AND~~
             ~~  Bit B2 of CAN_STATE_NOX_SENS_i = 1            AND~~
             ~~  Bit B3 of CAN_STATE_NOX_SENS_i = 1            AND~~
             ~~  Bit B6 of CAN_STATE_NOX_SENS_i = 0 )~~

      then(2)  LAMB_NOX_SENS_i =  LAMB_NOX_SENS_DIAG_i

            if(3)  LV_NOX_SENS_LAMB_AUTH_i = 1
```

```
    then(3) LV_NOX_SENS_LAMB_VLD_i = 1
            LV_NOX_SENS_LAMB_DIAG_VLD_i = 1
            T_NOX_SENS_LAMB_HLD_i = 0
    else(3) increment T_NOX_SENS_LAMB_HLD_i
        if(4)   T_NOX_SENS_LAMB_HLD_i >= C_T_NOX_SENS_LAMB_AUTH
        then(4) LV_NOX_SENS_LAMB_AUTH_i = 1
                LV_NOX_SENS_LAMB_VLD_i = 1
                LV_NOX_SENS_LAMB_DIAG_VLD_i = 1
        endif(4)
    endif(3)
else(2)
    if(5)   LV_NOX_SENS_LAMB_AUTH_i = 1
    then(5)
        if(6)   Bit PWR_OK of NOX_SENS_TEMP_i = 1
        then(6) LV_NOX_SENS_LAMB_DIAG_VLD_i = 0
                increment T_NOX_SENS_LAMB_HLD_i
            if(7)   LC_NOX_SENS_LAMB_VLD_i = 1
            then(7) LAMB_NOX_SENS_i =  LAMB_NOX_SENS_DIAG_i
                    LV_NOX_SENS_LAMB_VLD_i = 1
            else(7)
                if(8)   T_NOX_SENS_LAMB_HLD_i >=
                                   C_T_NOX_SENS_LAMB_HLD
                then(8) LAMB_NOX_SENS_i = C_LAMB_NOX_SENS_i
                        LV_NOX_SENS_LAMB_VLD_i = 0
                endif(8)
            endif(7)
        else(6)
            LAMB_NOX_SENS_i = C_LAMB_NOX_SENS_i
            LAMB_NOX_SENS_DIAG_i = C_LAMB_NOX_SENS_i
            LV_NOX_SENS_LAMB_VLD_i = LC_NOX_SENS_LAMB_MAN_i
            LV_NOX_SENS_LAMB_DIAG_VLD_i = 0
            LV_NOX_SENS_LAMB_AUTH_i = LC_NOX_SENS_LAMB_MAN_i
            T_NOX_SENS_LAMB_HLD_i = 0
        endif(6)
    else(5) LAMB_NOX_SENS_i = C_LAMB_NOX_SENS_i
            LAMB_NOX_SENS_DIAG_i = C_LAMB_NOX_SENS_i
            LV_NOX_SENS_LAMB_VLD_i = LC_NOX_SENS_LAMB_MAN_i
            LV_NOX_SENS_LAMB_DIAG_VLD_i = 0
            LV_NOX_SENS_LAMB_AUTH_i = LC_NOX_SENS_LAMB_MAN_i
            T_NOX_SENS_LAMB_HLD_i = 0
    endif(5)
endif(2)
```

```
else(1) LAMB_NOX_SENS_i = C_LAMB_NOX_SENS_i
        LAMB_NOX_SENS_DIAG_i = C_LAMB_NOX_SENS_i
        LV_NOX_SENS_LAMB_VLD_i = LC_NOX_SENS_LAMB_MAN_i
        LV_NOX_SENS_LAMB_DIAG_VLD_i = 0
        LV_NOX_SENS_LAMB_AUTH_i = LC_NOX_SENS_LAMB_MAN_i
        T_NOX_SENS_LAMB_HLD_i = 0
endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NOX_SENS_LAMB_MAN[NC_NOX_SENS_CONF]	1	0...1H	0...1	1	[-]
Manual setting of Lambda value on/ off					
C_LAMB_NOX_SENS[NC_NOX_SENS_CONF]	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Manual Lambda value					
LC_NOX_SENS_LAMB_VLD[NC_NOX_SENS_CONF]	1	0...1H	0...1	1	[-]
Permanently authorisation Lambda signal on/ off					
C_T_NOX_SENS_LAMB_HLD	1	0...FFFFH	0...655.35	0.01	[s]
Maximum time of hold the last valid Lambda signal value					
C_T_NOX_SENS_LAMB_AUTH	1	0...FFFFH	0...655.35	0.01	[s]
Minimum time of Lambda signal readiness to set the authorisation bit					
C_STATE_LAMB_NS_MASK	1	0...FFH	0...255	1	[-]
Bit-mask for detection of valid Lambda signal from NOx sensor					

52.3.3 Binary O2 signal

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
VLS_NOX_SENS_DIAG[NC_NOX_SENS_CONF]	V/O	0...578H	-200...1200	1	[mV]
Binary O2 signal voltage for diagnosis functions, raw value, measured by NOx sensor					
LV_NOX_SENS_VLS_DIAG_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the binary O2 signal is valid for use into the diagnosis functions					
VLS_NOX_SENS[NC_NOX_SENS_CONF]	V/O	0...578H	-200...1200	1	[mV]
Binary O2 signal voltage, raw value, measured by NOx sensor					
LV_NOX_SENS_VLS_VLD[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the binary O2 signal is valid					
LV_NOX_SENS_VLS_AUTH[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Indicator that the binary O2 signal has be reached the state "valid" after start					
T_NOX_SENS_VLS_HLD[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...655.35	0.01	[s]
Timer for hold of last valid binary O2 signal value					

Input data:

CAN_VLS_NOX_SENS[NC_NOX_SENS_CONF]	CAN_STATE_NOX_SENS[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_BOFF
LV_IGK	LV_CAN_TEMP_MIN_THD[NC_NOX_SENS_CONF]		NOX_SENS_TEMP[NC_NOX_SENS_CONF]
NC_NOX_SENS_CONF	CAN_STATE_NS_TMP[NC_NOX_SENS_CONF]	C_STATE_NS_MASK_MODE_1	C_STATE_NS_MASK_MODE_2
C_STATE_NS_MASK_MODE_5			

FUNCTION DESCRIPTION:

General information:

The EMS receives the binary O2 signal from the NOx sensor by CAN. The principle characteristic of the NOx sensor’s binary signal is the same as of a binary O2 sensor.

Description:

The binary O2 signal CAN_VLS_NOX_SENS_i, which is received from the NOx sensor, is copied to VLS_NOX_SENS_DIAG_i. After that the binary O2 signal is evaluated by the sensor status CAN_STATE_NOX_SENS_i.

The value of CAN_VLS_NOX_SENS_i is copied to VLS_NOX_SENS_i if no CAN bus error is occurred and CAN_STATE_NOX_SENS_i indicates at least level 2 sensor readiness by CAN_STATE_NOX_SENS_i = 0EH. For level 2 sensor readiness must be checked the bits B1, B2, B3 and B6 of CAN_STATE_NOX_SENS_i. All other bits of CAN_STATE_NOX_SENS_i have no relevance for level 2 readiness. At the same time the variables LV_NOX_SENS_VLS_VLD_i and LV_NOX_SENS_VLS_DIAG_VLD_i are set to 1. So the system is authorised to use the binary O2 signal.

The variable LV_NOX_SENS_VLS_DIAG_VLD_i is reset to 0 every time when the sensor loses the level 2 readiness. So the diagnosis functions are informed in order to exclude a wrong diagnosis. At the same time starts the timer T_NOX_SENS_VLS_HLD_i. It runs up to the maximum value. An overflow is not allowed. The timer is stopped and reset, when the sensor reaches the level 2 readiness.

If the permanently authorisation of binary O2 signal is activated by LC_NOX_SENS_VLS_VLD_i = 1 and LV_NOX_SENS_NOX_AUTH_i = 1 then it is allowed to pass the binary O2 signal to VLS_NOX_SENS_i without level 2 sensor readiness. The variable LV_NOX_SENS_VLS_VLD_i is set to 1 so that the system is authorised to use the binary O2 signal. After the engine start the permanently authorisation of binary O2 signal is blocked up to LV_NOX_SENS_VLS_AUTH_i = 1. Then the binary O2 signal values are passed to VLS_NOX_SENS_i.

When the sensor loses the level 2 readiness the last valid binary O2 signal value VLS_NOX_SENS_i and LV_NOX_SENS_VLS_VLD_i are held up to the time C_T_NOX_SENS_VLS_HLD is passed. After that the binary O2 signal VLS_NOX_SENS_i is set to C_VLS_NOX_SENS_i and LV_NOX_SENS_VLS_VLD_i is reset.

If the manual setting of binary O2 values is active LC_NOX_SENS_VLS_MAN_i = 1 then VLS_NOX_SENS_DIAG_i and VLS_NOX_SENS_i are set to C_VLS_NOX_SENS_i, LV_NOX_SENS_VLS_VLD_i is set to 1 and LV_NOX_SENS_VLS_DIAG_VLD_i is reset.

Application conditions:

Initialisation: at reset, at LV_IGK = 1 -> 0:

VLS_NOX_SENS_DIAG_i = C_VLS_NOX_SENS_i
VLS_NOX_SENS_i = C_VLS_NOX_SENS_i
LV_NOX_SENS_VLS_VLD_i = 0
LV_NOX_SENS_VLS_DIAG_VLD_i = 0
LV_NOX_SENS_VLS_AUTH_i = 0
T_NOX_SENS_VLS_HLD_i = 0

Formula section:

```
if(1)  LC_NOX_SENS_VLS_MAN_i = 0                                      AND
       LV_CAN_TEMP_MIN_THD_i = 1                                      AND
       ( CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_1          OR
         CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_2          OR
         CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_5 )
then(1)  VLS_NOX_SENS_DIAG_i = CAN_VLS_NOX_SENS_i
         if(2)  LV_ERR_NS_CAN_MSG_LOST[i] = 0                         AND
                LV_ERR_NS_CAN_BOFF = 0                                AND
                ( CAN_STATE_NOX_SENS[i] bitwise AND
                      C_STATE_VLS_NS_MASK ) = C_STATE_VLS_NS_MASK
                ~~( Bit B1 of CAN_STATE_NOX_SENS_i = 1             AND~~
                ~~  Bit B2 of CAN_STATE_NOX_SENS_i = 1             AND~~
                ~~  Bit B3 of CAN_STATE_NOX_SENS_i = 1             AND~~
                ~~  Bit B6 of CAN_STATE_NOX_SENS_i = 0 )~~
         then(2)  VLS_NOX_SENS_i = VLS_NOX_SENS_DIAG_i
                  if(3)  LV_NOX_SENS_VLS_AUTH_i = 1
                  then(3)  LV_NOX_SENS_VLS_VLD_i = 1
                           LV_NOX_SENS_VLS_DIAG_VLD_i = 1
                           T_NOX_SENS_VLS_HLD_i = 0
                  else(3)  increment T_NOX_SENS_VLS_HLD_i
```

```
                    if(4)  T_NOX_SENS_VLS_HLD_i >= C_T_NOX_SENS_VLS_AUTH
                    then(4)  LV_NOX_SENS_VLS_AUTH_i = 1
                             LV_NOX_SENS_VLS_VLD_i = 1
                             LV_NOX_SENS_VLS_DIAG_VLD_i = 1
                    endif(4)
                endif(3)
            else(2)
                if(5)  LV_NOX_SENS_VLS_AUTH_i = 1
                then(5)
                    if(6)  Bit PWR_OK of NOX_SENS_TEMP_i = 1
                    then(6)  LV_NOX_SENS_VLS_DIAG_VLD_i = 0
                             increment T_NOX_SENS_VLS_HLD_i
                        if(7)  LC_NOX_SENS_VLS_VLD_i = 1
                        then(7)  VLS_NOX_SENS_i = VLS_NOX_SENS_DIAG_i
                                 LV_NOX_SENS_VLS_VLD_i = 1
                        else(7)
                            if(8)  T_NOX_SENS_VLS_HLD_i >=
                                             C_T_NOX_SENS_VLS_HLD
                            then(8)  VLS_NOX_SENS_i = C_VLS_NOX_SENS_i
                                     LV_NOX_SENS_VLS_VLD_i = 0
                            endif(8)
                        endif(7)
                    else(6)  VLS_NOX_SENS_i = C_VLS_NOX_SENS_i
                             VLS_NOX_SENS_DIAG_i = C_VLS_NOX_SENS_i
                             LV_NOX_SENS_VLS_VLD_i = LC_NOX_SENS_VLS_MAN_i
                             LV_NOX_SENS_VLS_DIAG_VLD_i = 0
                             LV_NOX_SENS_VLS_AUTH_i = LC_NOX_SENS_VLS_MAN_i
                             T_NOX_SENS_VLS_HLD_i = 0
                    endif(6)
                else(5)  VLS_NOX_SENS_i = C_VLS_NOX_SENS_i
                         VLS_NOX_SENS_DIAG_i = C_VLS_NOX_SENS_i
                         LV_NOX_SENS_VLS_VLD_i = LC_NOX_SENS_VLS_MAN_i
                         LV_NOX_SENS_VLS_DIAG_VLD_i = 0
                         LV_NOX_SENS_VLS_AUTH_i = LC_NOX_SENS_VLS_MAN_i
                         T_NOX_SENS_VLS_HLD_i = 0
                endif(5)
            endif(2)
        else(1)  VLS_NOX_SENS_i = C_VLS_NOX_SENS_i
                 VLS_NOX_SENS_DIAG_i = C_VLS_NOX_SENS_i
                 LV_NOX_SENS_VLS_VLD_i = LC_NOX_SENS_VLS_MAN_i
                 LV_NOX_SENS_VLS_DIAG_VLD_i = 0
                 LV_NOX_SENS_VLS_AUTH_i = LC_NOX_SENS_VLS_MAN_i
                 T_NOX_SENS_VLS_HLD_i = 0
```

endif(1)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NOX_SENS_VLS_MAN[NC_NOX_SENS_CONF]	1	0...1H	0...1	1	[-]
Manual setting of binary O2 value on/ off					
C_VLS_NOX_SENS[NC_NOX_SENS_CONF]	1	0...578H	-200...1200	1	[mV]
Manual binary O2 value					
LC_NOX_SENS_VLS_VLD[NC_NOX_SENS_CONF]	1	0...1H	0...1	1	[-]
Permanently authorisation binary O2 signal on/ off					
C_T_NOX_SENS_VLS_HLD	1	0...FFFFH	0...655.35	0.01	[s]
Maximum time of hold the last valid binary O2 signal value					
C_T_NOX_SENS_VLS_AUTH	1	0...FFFFH	0...655.35	0.01	[s]
Minimum time of binary O2 signal readiness to set the authorisation bit					
C_STATE_VLS_NS_MASK	1	0...FFH	0...255	1	[-]
Bit-mask for detection of valid Binary O2 signal from NOx sensor					

52.3.4 NOx sensor errors

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
ERR_NS[NC_NOX_SENS_CONF]	V/O	0...FFH	0...255	1	[-]
NOx sensor failure byte					

Input data:

CAN_ERR_NOX_SENS[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_BOFF	LV_IGK
NC_NOX_SENS_CONF			

FUNCTION DESCRIPTION:

General information:

The NOx sensor monitors the connection between the NOx sensor interface unit and the sensor element. The sensor interface is able to detect open wires and short circuits between the sensor connection wires.

The NOx sensor sends the monitoring result together with the measurement data as failure byte to the EMS. The EMS uses this failure byte for the NOx sensor diagnosis.

Description:

The NOx sensor's failure byte is valid, if no CAN bus error is occurred. In this case the failure byte CAN_ERR_NOX_SENS_i is copied to the variable ERR_NS[i], so that it can be evaluated by the NOx sensor diagnosis function. Otherwise the variable ERR_NS[i] is reset.

If the manual setting of error byte is active LC_NOX_SENS_ERR_MAN_i = 1 then ERR_NS[i] is set to C_ERR_NOX_SENS_i.

Application conditions:

Initialisation: at reset, at LV_IGK = 1 -> 0: ERR_NS[i] = 0

Formula section:

```
if    LC_NOX_SENS_ERR_MAN_i = 0
then
      if    LV_ERR_NS_CAN_MSG_LOST[i] = 0        AND
            LV_ERR_NS_CAN_BOFF = 0
      then  ERR_NS[i] = CAN_ERR_NOX_SENS_i
      else  ERR_NS[i] = 0
      endif
else  ERR_NS[i] = C_ERR_NOX_SENS_i
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NOX_SENS_ERR_MAN[NC_NOX_SENS_CONF]	1	0...1H	0...1	1	[-]
Manual setting of error byte on/ off					
C_ERR_NOX_SENS[NC_NOX_SENS_CONF]	1	0...FFH	0...255	1	[-]
Manual value for error byte					

52.3.5 Sensor temperature

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
NOX_SENS_TEMP[NC_NOX_SENS_CONF]	V/O	0...3H	0...3	1	[-]
Indicator for state of sensor heater management					
T_NOX_SENS_PWR[NC_NOX_SENS_CONF]	V	0...FFFFH	0...655.35	0.01	[s]
Timer for debounce of NOx sensor power supply failures					

Input data:

CAN_STATE_NOX_SENS[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_BOFF	NC_NOX_SENS_CONF

FUNCTION DESCRIPTION:

General information:

The NOx sensor interface unit controls the temperature of the sensor element. The state of the temperature control is transmitted to the EMS by the status byte.

The NOx sensor observes its power supply voltage. If the power supply voltage is in the range MIN < voltage < MAX then the bit B0 of CAN_STATE_NOX_SENS_i is 0.

In the case of voltage < MIN (specification value of sensor) it is possible that the sensor can not control the correct sensor element temperature. Because of this the sensor loses the readiness. The possible reasons of undervoltage are a to low battery voltage or a high voltage drop over the power supply wire (e.g. connector problem).

In the case of voltage > MAX (specification value of sensor) the sensor is shut down up to the power supply voltage is again in the allowed range (self-protection mode).

At both cases the NOx sensor diagnosis functions must be deactivated in order to prevent wrong diagnoses.

Description:

The state of sensor temperature control and the state of sensor power supply voltage are indicated by bits of NOX_SENS_TEMP_i. The meaning of each bit of NOX_SENS_TEMP_i is described at the table below.

NOX_SENS_TEMP_i	Description
Bit B0 ... TEMP_OK	Sensor temperature is ok.
Bit B1 ... PWR_OK	Sensor power supply is ok.
Bit B2	not used
Bit B3	not used
Bit B4	not used
Bit B5	not used
Bit B6	not used
Bit B7	not used

The sensor temperature is valid if no CAN bus error is occurred and the status byte CAN_STATE_NOX_SENS_i indicates level 1 readiness.

The level 1 readiness is indicated by CAN_STATE_NOX_SENS_i = 02H. For the level 1 readiness must be checked the bits B1 and B6 of CAN_STATE_NOX_SENS_i. All other bits of CAN_STATE_NOX_SENS_i have no relevance for level 1 readiness.

If the level 1 readiness is ok then bit TEMP_OK of NOX_SENS_TEMP_i is set. Otherwise bit TEMP_OK of NOX_SENS_TEMP_i is reset.

With bit B0 of CAN_STATE_NOX_SENS_i = 0 indicates the sensor that the power supply is in the correct range. Then bit PWR_OK of NOX_SENS_TEMP_i is set to 1.

At undervoltage or at overvoltage the sensor sets bit B0 of CAN_STATE_NOX_SENS_i = 1. This information is combined with the temperature information from the sensor and the timer T_NOX_SENS_PWR_i in order to debounce short power supply peaks, which are reported by sensor.

Application conditions:

Initialisation: at reset: NOX_SENS_TEMP_i = 0
T_NOX_SENS_PWR_i = 0

Formula section:

```
if    LV_ERR_NS_CAN_MSG_LOST[i] = 0                                AND
      LV_ERR_NS_CAN_BOFF = 0
then
        if    Bit B1 of CAN_STATE_NOX_SENS_i = 1                   AND
              Bit B6 of CAN_STATE_NOX_SENS_i = 0
        then  Bit TEMP_OK of NOX_SENS_TEMP_i = 1
        else  Bit TEMP_OK of NOX_SENS_TEMP_i = 0
        endif
        if    CAN_STATE_NOX_SENS_i = 255                           OR
              Bit B0 of CAN_STATE_NOX_SENS_i = 0
        then  Bit PWR_OK of NOX_SENS_TEMP_i = 1
              T_NOX_SENS_PWR_i = 0
        else
                if    Bit B0 of CAN_STATE_NOX_SENS_i = 1           AND
                      Bit B1 of CAN_STATE_NOX_SENS_i = 0           AND
                      T_NOX_SENS_PWR_i >= C_T_NOX_SENS_PWR_HLD
                then  Bit PWR_OK of NOX_SENS_TEMP_i = 0
                else  increment T_NOX_SENS_PWR_i
                endif
        endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_NOX_SENS_PWR_HLD	1	0...FFFFH	0...655.35	0.01	[s]
Debounce time for NOx sensor power supply failures					

52.3.6 NOx sensor self diagnosis states

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
STATE_NS[NC_NOX_SENS_CONF]	O/V	0...FFH	0...255	1	[-]
NOx sensor state					
LV_NS_SHIFT_DIAG_ERR[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Sensor self diagnosis result from NOx sensor					

Input data:

CAN_STATE_NS_TMP[NC_NOX_SENS_CONF]	CAN_STATE_DIAG_NS[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_BOFF
NC_NOX_SENS_CONF	C_STATE_NS_MASK_MODE_5		

FUNCTION DESCRIPTION:

General information:

The NOx sensor indicates its self diagnosis state by CAN_STATE_DIAG_NS[i]. Into this state information is the diagnosis process state and the diagnosis result coded. These information are decoded here.

Description:

Application conditions:

Initialisation: at reset: STATE_NS[i] = 0
LV_NS_SHIFT_DIAG_ERR[i] = 0

Formula section:

```
if    C_STATE_DIAG_NS_MAN[i] = 0
then
        if   LV_ERR_NS_CAN_MSG_LOST[i] = 0                    AND
             LV_ERR_NS_CAN_BOFF = 0                           AND
             CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_5
        then
             case selection on CAN_STATE_DIAG_NS[i]:
```

```
            case:   CAN_STATE_DIAG_NS[i] = 0
                    Sensor self diagnosis is running.
                    STATE_NS[i] = 1
                    LV_NS_SHIFT_DIAG_ERR[i](n) = LV_NS_SHIFT_DIAG_ERR[i](n-1)

            case:   CAN_STATE_DIAG_NS[i] = 1
                    Sensor self diagnosis was aborted.
                    Sensor is returning to normal operation mode.
                    STATE_NS[i] = 2
                    LV_NS_SHIFT_DIAG_ERR[i](n) = LV_NS_SHIFT_DIAG_ERR[i](n-1)

            case:   CAN_STATE_DIAG_NS[i] = 2
                    Sensor self diagnosis was completed. Result: "Sensor is ok".
                    Sensor is returning to normal operation mode.
                    STATE_NS[i] = 3
                    LV_NS_SHIFT_DIAG_ERR[i] = 0

            case:   CAN_STATE_DIAG_NS[i] = 3
                    Sensor self diagnosis was completed. Result: "Sensor is faulty".
                    Sensor is returning to normal operation mode.
                    STATE_NS[i] = 3
                    LV_NS_SHIFT_DIAG_ERR[i] = 1

            default:
                    Sensor self diagnosis is "off".
                    STATE_NS[i] = 0
                    LV_NS_SHIFT_DIAG_ERR[i](n) = LV_NS_SHIFT_DIAG_ERR[i](n-1)
        end case selection

    else    Sensor self diagnosis is "off".
            STATE_NS[i] = 0
            LV_NS_SHIFT_DIAG_ERR[i](n) = LV_NS_SHIFT_DIAG_ERR[i](n-1)
    endif
else
    case selection on C_STATE_DIAG_NS_MAN[i]:
        case:   C_STATE_DIAG_NS_MAN[i] = 1
                    Sensor self diagnosis is running.
                    STATE_NS[i] = 1
                    LV_NS_SHIFT_DIAG_ERR[i](n) = LV_NS_SHIFT_DIAG_ERR[i](n-1)

        case:   C_STATE_DIAG_NS_MAN[i] = 2
```

```
                Sensor self diagnosis was aborted.
                Sensor is returning to normal operation mode.
                STATE_NS[i] = 2
                LV_NS_SHIFT_DIAG_ERR[i](n) = LV_NS_SHIFT_DIAG_ERR[i](n-1)

        case:   C_STATE_DIAG_NS_MAN[i] = 3
                Sensor self diagnosis was completed. Result: "Sensor is ok".
                Sensor is returning to normal operation mode.
                STATE_NS[i] = 3
                LV_NS_SHIFT_DIAG_ERR[i] = 0

        case:   C_STATE_DIAG_NS_MAN[i] = 4
                Sensor self diagnosis was completed. Result: "Sensor is faulty".
                Sensor is returning to normal operation mode.
                STATE_NS[i] = 3
                LV_NS_SHIFT_DIAG_ERR[i] = 1

        default:
                Sensor self diagnosis is "off".
                STATE_NS[i] = 0
                LV_NS_SHIFT_DIAG_ERR[i](n) = LV_NS_SHIFT_DIAG_ERR[i](n-1)
    end case selection
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_DIAG_NS_MAN[NC_NOX_SENS_CONF]	1	0...FFH	0...255	1	[-]
Switch for manual set of NOx sensor self diagnosis state					

52.4 NOx sensor interface (Application Incidences)

Input data:

NC_NOX_SENS_CONF			

General information:

The number of NOx sensors in the system is defined by NC_NOX_SENS_CONF.

```
if    NC_NOX_SENS_CONF = 0
then  The system contains no NOx sensor
      no output variables shall be generated
else
      if    NC_NOX_SENS_CONF = 1
      then  The system contains one NOx sensor
            i = 1 ... all NOx sensor variables are xxx[i = 1]
      else
            if    NC_NOX_SENS_CONF = 2
            then  The system contains two NOx sensors
                  i = 1 ... the NOx sensor variables for exhaust line 1 are xxx[i = 1]
                  i = 2 ... the NOx sensor variables for exhaust line 2 are xxx[i = 2]
            endif
      endif
endif
```

52.4.1 NOx sensor activation data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CAN_TEMP_MIN_THD[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
NOx sensor dew point					
LV_MDL_TEMP_MIN_THD[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	[-]
Temperature model has reached the dew point for NOx sensor					
CAN_STATE_NS_DIAG_REQ[NC_NOX_SENS_CONF]	O/V	0...FH	0...15	1	[-]
Request to NOx sensor for doing a sensor self diagnosis					
LV_NS_VERS_REQ_1[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Request to NOx sensor for transmission of sensor identification data					
LV_NS_VERS_REQ_2[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Selection of NOx sensor identification data part 1 or 2					

Input data:

LV_TNT_MIN_THD_2	CAN_R_RATIO_NOX_SENS[NC_NOX_SENS_CONF]	LV_IGK	LV_DIAG_REQ_NS_SHIFT[NC_NOX_SENS_CONF]
STATE_NS_VERS[NC_NOX_SENS_CONF]	T_AST_SAE	CAN_HW_NS[NC_NOX_SENS_CONF]	

FUNCTION DESCRIPTION:

General information:

This chapter is used as interface between the variables on the CAN bus and the system variables.

The NOx sensor needs from the EMS the information, that the dew point is passed. This information is necessary for the heater management of the NOx sensor. It protects the NOx sensor in order to avoid damages by water splash of condense water in the exhaust system. The dew point recognition is performed in the ‘NOx catalyst monolith temperature model’.

If the NOx sensor has been received the dew point from the EMS, then the heater management warms up the sensor on the normal operating temperature.

Description:

The dew point flag for the NOx sensor LV_CAN_TEMP_MIN_THD_i is set, if the dew point, which is calculated by the NOx catalyst monolith temperature model, is reached or if the NOx sensor's ceramic (sensor element) was sufficiently heated up by exhaust gas. The detection of sufficient sensor element heating uses a temperature dependent resistance ratio signal, which is sent by the NOx sensor while the NOx sensor is waiting for the dew point.

The resistance ratio signal is sent instead of the NOx signal. This resistance ratio signal represents the momentary temperature of sensor element. It is defined as:

$$\mathrm{CAN\_R\_RATIO\_NOX\_SENS\_i} = \frac{R}{R_{25}} = 911 \cdot \left(1 + \alpha \cdot T + \beta \cdot T^2\right)$$

α ... material constant, $\alpha = 3.92 \cdot 10^{-3}\ K^{-1}$

β ... material constant, $\beta = -0.6 \cdot 10^{-6}\ K^{-2}$

T ... temperature of sensor element [°C]

The scope of dew point detection by sensor element temperature is an earlier set of dew point in comparison with the dew point set by NOx catalyst monolith temperature model.

The dew point is detected by sensor element temperature when the resistance ratio signal from the NOx sensor exceeds the threshold C_R_RATIO_NOX_SENS_MIN. The set of dew point by sensor element temperature is allowed only 1x after engine start. A reset of dew point is possible by the NOx catalyst temperature model within a driving cycle (e.g. very long pull fuel cutoff phase). In this case the sensor temperature condition has not the right to set the dew point again. Hence the resistance ratio signal is set to maximum value after the dew point was reached the 1st time within a driving cycle and is reset to "0" at every engine start.

Calibration hints:

A typical sensor element temperature threshold for set of dew point is:

T ≈ 300°C => C_R_RATIO_NOX_SENS_MIN = 1936

Other values for C_R_RATIO_NOX_SENS_MIN can be calculated by use of definition equation (see above).

This function can be switched off by calibration

C_R_RATIO_NOX_SENS_MIN = maximum value (7FFFH or 32767)

Application conditions:

Initialisation: at reset:

LV_CAN_TEMP_MIN_THD_i = 0
LV_MDL_TEMP_MIN_THD_i = 0
CAN_STATE_NS_DIAG_REQ[i] = 0
LV_NS_VERS_REQ_1[i] = 0
LV_NS_VERS_REQ_2[i] = 0

Recurrence: corresponding to EMS send message with CAN-ID = 140H

Activation: after reset, at every engine state

Deactivation: -

Formula section:

```
if    LV_IGK = 1
then
        if    LV_TNT_MIN_THD_2 = 1
        then  LV_CAN_TEMP_MIN_THD_i = 1      (valid for i = 1 and i = 2)
              LV_MDL_TEMP_MIN_THD_i = 1      (valid for i = 1 and i = 2)
        else
              if   LV_MDL_TEMP_MIN_THD_i = 0                            AND
                   CAN_R_RATIO_NOX_SENS_i > C_R_RATIO_NOX_SENS_MIN
```

```
                then    LV_CAN_TEMP_MIN_THD_i = 1
                else    LV_CAN_TEMP_MIN_THD_i = 0
                endif
            endif
            if  LV_DIAG_REQ_NS_SHIFT[i] = 1                              AND
                LV_CAN_TEMP_MIN_THD[i] = 1
            then    CAN_STATE_NS_DIAG_REQ[i] = C_STATE_NS_DIAG_REQ_MASK_A
            else    CAN_STATE_NS_DIAG_REQ[i] = C_STATE_NS_DIAG_REQ_MASK_OFF
            endif

    else    LV_CAN_TEMP_MIN_THD_i = 0        (valid for i = 1 and i = 2)
            LV_MDL_TEMP_MIN_THD_i = 0        (valid for i = 1 and i = 2)
            CAN_STATE_NS_DIAG_REQ[i] = C_STATE_NS_DIAG_REQ_MASK_OFF
    endif

    if  C_STATE_NS_VERS_REQ_CAN_MAN > 0
    then    Manual controlled set of request for NOx sensor identification data
            case selection on C_STATE_NS_VERS_REQ_CAN_MAN:
                case:   C_STATE_NS_VERS_REQ_CAN_MAN = 1
                        Sensor identification data, part 1 are requested
                        LV_NS_VERS_REQ_1[i] = 1
                        LV_NS_VERS_REQ_2[i] = 0

                case:   C_STATE_NS_VERS_REQ_CAN_MAN = 2
                        Sensor identification data, part 2 are requested
                        LV_NS_VERS_REQ_1[i] = 1
                        LV_NS_VERS_REQ_2[i] = 1

                default:
                        No sensor identification data are requested
                        LV_NS_VERS_REQ_1[i] = 0
                        LV_NS_VERS_REQ_2[i] = 0

            end case selection

    else    Regular controlled set of request for NOx sensor identification data
            if      CAN_HW_NS[i] > 0 OR T_AST_SAE >= C_T_AST_CAN_HW_NS_MAX
            then
                    case selection on STATE_NS_VERS[i]:
```

```
                case: STATE_NS_VERS[i] = C_STATE_NS_VERS_REQ_MASK_1

                          Sensor identification data, part 1 are requested
                          LV_NS_VERS_REQ_1[i] = 1
                          LV_NS_VERS_REQ_2[i] = 0

                case: STATE_NS_VERS[i] = C_STATE_NS_VERS_REQ_MASK_2

                          Sensor identification data, part 2 are requested
                          LV_NS_VERS_REQ_1[i] = 1
                          LV_NS_VERS_REQ_2[i] = 1

                default:

                          No sensor identification data are requested
                          LV_NS_VERS_REQ_1[i] = 0
                          LV_NS_VERS_REQ_2[i] = 0

                end case selection

        else    Sensor identification data, part 2 are requested until received or time out
                LV_NS_VERS_REQ_1[i] = 1
                LV_NS_VERS_REQ_2[i] = 1

        endif

endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_R_RATIO_NOX_SENS_MIN	1	8000...7FFFH	-32768...32767	1	[-]
Minimum sensor element temperature for set of dew point ahead of temperature model					
C_STATE_NS_DIAG_REQ_MASK_OFF	1	0...FH	0...15	1	[-]
Bit-Mask for NOx sensor self diagnosis request "OFF"					
C_STATE_NS_DIAG_REQ_MASK_A	1	0...FH	0...15	1	[-]
Bit-Mask "A" for request of NOx sensor self diagnosis ("A" ... at Lambda=1.0 conditions)					
C_STATE_NS_VERS_REQ_CAN_MAN	1	0...2H	0...2	1	[-]
Manual set of request for read out of NOx sensor identification data					
C_STATE_NS_VERS_REQ_MASK_1	1	0...FFH	0...255	1	[-]
STATE_NS_VERS bit-mask for request of NOx sensor identification data part 1					
C_STATE_NS_VERS_REQ_MASK_2	1	0...FFH	0...255	1	[-]
STATE_NS_VERS bit-mask for request of NOx sensor identification data part 2					
C_T_AST_CAN_HW_NS_MAX	1	0...FFFFH	0...65535	1	[s]
Maximum engine running time for receiving of NOx sensor hardware information					

52.4.2 CAN Protocol

General information:

The following control units participate to CAN-communication:

EMS:	Engine management system
NOX1:	NOx sensor on exhaust line 1
NOX2:	NOx sensor on exhaust line 2

CAN communication is based on messages. A message is defined at the identifier number. Valid identifier numbers are numbers from 1 to 1023 (short identifier). Messages with lower identifier number have higher priority. The length of a message can vary from 1 to 8 bytes.

Baudrate: 500 kBaud
Data format: Motorola

The NOx sensor in exhaust line 1 sends its measurement values on CAN Messages with the ID = 130H. The NOx sensor in exhaust line 2 sends its measurement values on CAN Messages with the ID = 135H.

The NOx sensors in all exhaust lines receive the dew point on the CAN Message with the ID = 140H.

52.4.2.1 Send messages by EMS

52.4.2.1.1 ID = 140H - NOx sensor dew point

Input data:

LV_CAN_TEMP_MIN_THD[NC_NOX_SENS_CONF]	LV_NS_VERS_REQ_1[NC_NOX_SENS_CONF]	LV_NS_VERS_REQ_2[NC_NOX_SENS_CONF]	CAN_STATE_NS_DIAG_REQ[NC_NOX_SENS_CONF]

General information:

ATTENTION!!!

The CAN message description in this function specification is **for information only!** It is not included within NOx sensor component change management!

This message transfers the dew point and special operation mode requests from the EMS to all NOx sensors within the system. The sending of this message is permanently active.

Description:

The calibration value NOx sensor self diagnosis run-in time C_T_RUN_DIAG_NS_SHIFT allows to overwrite the NOx sensor internal default value of this parameter. For a explanation of NOx sensor self diagnosis, its parameters and effects contact the NOx sensor supplier or see NOx sensor component specification.

Calibration hint:

It is strongly recommended to use the NOx sensor internal default value of this parameter. Hence use the According to the definitions into the NOx sensor component specification for project MSD80, the following calibration is recommended:

1. activation of functionality:

   C_T_RUN_DIAG_NS_SHIFT > 0.0 s

2. deactivation of functionality (use of NOx sensor internal default value):

   C_T_RUN_DIAG_NS_SHIFT = 0.0 s

Application conditions:

Initialisation: -

Recurrence: 100ms

Activation: *after reset*

Deactivation: -

Formula section:

CAN Frame and Signal Definitions:

ID = 140H, base time = 100ms - NOx sensor dew point

Bit:	7 MSB	6	5	4	3	2	1	0 LSB
Byte: 0	Dew point for sensor ID=130H	0	0	0	Sensor identification request 1 for sensor ID=130H	0	Sensor identification request 2 for sensor ID=130H	0
1	Dew point for sensor ID=135H	0	0	0	Sensor identification request 1 for sensor ID=135H	0	Sensor identification request 2 for sensor ID=135H	0
2	0	0	0	0	Diagnosis request for sensor ID=130H			
3	0	0	0	0	Diagnosis request for sensor ID=135H			
4	Diagnosis run-in time							
5	00H							
6	00H							
7	00H							

ID	Signal	Hex Range	Phys. Range	Resolution	Unit	Init Value	Default Value
140H	Dew point for sensor ID=130H & 135H	0...1H	0...1	1	[-]	0	0
140H	Sensor identification request 1 for sensor ID=130H & 135H	0...1H	0...1	1	[-]	0	0
140H	Sensor identification request 2 for sensor ID=130H & 135H	0...1H	0...1	1	[-]	0	0
140H	Diagnosis request for sensor ID=130H & 135H	0...FH	0...15	1	[-]	0	0
140H	Diagnosis run-in time	0...FFH	0.0...25.5	0.1	[s]	0	0

ID	Signal	Signal value	Description
140H	Dew point for sensor ID=130H & 135H	0	Dew point is not reached
		1	Dew point is reached
140H	Sensor identification request 1 for sensor ID=130H & 135H	0	No request
		1	Request for sensor identification data
140H	Sensor identification request 2 for sensor ID=130H & 135H	0	Request for part 1 of sensor identification data
		1	Request for part 2 of sensor identification data
140H	Diagnosis request for sensor ID=130H & 135H	0H (0000b)	No request, sensor self diagnosis = "off"
		1H (0001b)	Conditions for sensor self diagnosis at λ=1.0 present
		2H (0010b)	Conditions for sensor self diagnosis at pull fuel cutoff (Schub) present
		4H (0100b)	Conditions for sensor self diagnosis at λ=xxx present
		8H (1000b)	Conditions for sensor self diagnosis at λ=xxx present
		other values	Other values are not defined. => Hence they are not allowed.
140H	Diagnosis run-in time	0.0 s	No diagnosis run-in time change request from EMS. Use sensor internal value (default value).
		0.1...25.5 s	Diagnosis run-in time from EMS

ID	Signal	ECU signal variable name
140H	Dew point for sensor ID=130H	LV_CAN_TEMP_MIN_THD[1]
140H	Dew point for sensor ID=135H	LV_CAN_TEMP_MIN_THD[2] 5)
140H	Sensor identification request 1 for sensor ID=130H	LV_NS_VERS_REQ_1[1]
140H	Sensor identification request 2 for sensor ID=130H	LV_NS_VERS_REQ_2[1]
140H	Sensor identification request 1 for sensor ID=135H	LV_NS_VERS_REQ_1[2] 5)
140H	Sensor identification request 2 for sensor ID=135H	LV_NS_VERS_REQ_2[2] 5)
140H	Diagnosis request for sensor ID=130H	CAN_STATE_NS_DIAG_REQ[1]
140H	Diagnosis request for sensor ID=135H	CAN_STATE_NS_DIAG_REQ[2] 5)
140H	Diagnosis run-in time	C_T_RUN_DIAG_NS_SHIFT

5) If NC_NOX_SENS_CONF = 1 then use the i = 1 variables instead of i = 2 variables.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_RUN_DIAG_NS_SHIFT	1	0...FFH	0...25.5	0.1	[s]
NOx sensor self diagnosis run-in time					

52.4.2.2 Received messages by EMS

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CAN_NOX_NOX_SENS[NC_NOX_SENS_CONF]	O/V	FF9C...05DCH	-100...1500	1	[ppm]
NOx concentration value, measured by NOx sensor					
CAN_LAMB_NOX_SENS[NC_NOX_SENS_CONF]	O/V	0...0604H	0...1540	1	[-]
Lambda value as [1000/ Lambda], measured by NOx sensor					
CAN_VLS_NOX_SENS[NC_NOX_SENS_CONF]	O/V	0...578H	-200...1200	1	[mV]
Binary O2 signal voltage, raw value, measured by NOx sensor					
CAN_STATE_NOX_SENS[NC_NOX_SENS_CONF]	O/V	0...FFH	0...255	1	[-]
NOx sensor status byte					
CAN_ERR_NOX_SENS[NC_NOX_SENS_CONF]	O/V	0...FFH	0...255	1	[-]
NOx sensor error byte					
CAN_R_RATIO_NOX_SENS[NC_NOX_SENS_CONF]	O/V	8000...7FFFH	-32768...32767	1	[-]
Sensor element temperature depended resistance ratio as 1000 * R/ R25					
CAN_STATE_NS_TMP[NC_NOX_SENS_CONF]	O/V	0...FFH	0...255	1	[-]
NOx sensor state - Temporary signal for sensor mode switching					
CAN_SW_NS[NC_NOX_SENS_CONF]	O/V	0...FFH	0...255	1	[-]
NOx sensor software version number					
CAN_HW_NS[NC_NOX_SENS_CONF]	O/V	0...FFH	0...255	1	[-]
NOx sensor hardware version number					
CAN_STATE_DIAG_NS[NC_NOX_SENS_CONF]	O/V	0...FFH	0...255	1	[-]
NOx sensor self diagnosis state					
CAN_NOX_REF_NS[NC_NOX_SENS_CONF]	O/V	8000...7FFFH	-32768...32767	1	[-]
NOx signal - Diagnosis reference value					
CAN_NOX_DIAG_NS[NC_NOX_SENS_CONF]	O/V	8000...7FFFH	-32768...32767	1	[-]
NOx signal - Diagnosis value					
CAN_TMP_WORD_1[NC_NOX_SENS_CONF]	V	8000...7FFFH	-32768...32767	1	[-]
Temporary signal from CAN message buffer - word 1					
CAN_TMP_BYTE_1[NC_NOX_SENS_CONF]	-	0...FFH	0...255	1	[-]
Temporary signal from CAN message buffer - byte 1					
CAN_TMP_BYTE_2[NC_NOX_SENS_CONF]	-	0...FFH	0...255	1	[-]
Temporary signal from CAN message buffer - byte 2					
CAN_TMP_BYTE_3[NC_NOX_SENS_CONF]	-	0...FFH	0...255	1	[-]
Temporary signal from CAN message buffer - byte 3					
CAN_TMP_BYTE_4[NC_NOX_SENS_CONF]	-	0...FFH	0...255	1	[-]
Temporary signal from CAN message buffer - byte 4					
CAN_TMP_BYTE_5[NC_NOX_SENS_CONF]	-	0...FFH	0...255	1	[-]
Temporary signal from CAN message buffer - byte 5					
CAN_TMP_BYTE_6[NC_NOX_SENS_CONF]	-	0...FFH	0...255	1	[-]
Temporary signal from CAN message buffer - byte 6					
CAN_TMP_BYTE_8[NC_NOX_SENS_CONF]	-	0...FFH	0...255	1	[-]

Temporary signal from CAN message buffer - byte 8

Input data:

LV_IGK			

General information:

For all received messages are applicable:

```
if    LV_IGK = 0                                          OR
      lost message error of a particular message is detected
then
          if    default values for this message are defined
          then  the default values are used
          else  the previous values of this message are used
          endif
else      the received message values are used
endif
```

Description:

The NOx sensor transmits different signals, which can identified by the "Sensor status" signal. Detailed information are included into the next section with the CAN message description.

ATTENTION!!!

The CAN message description in this function specification is **for information only!** It is not included within NOx sensor component change management!

Calibration hint:

According to the definitions into the NOx sensor component specification for project MSD80, the following calibration is recommended:

C_CAN_STATE_NS_TMP_MASK = 1110 0000b = E0H = 224 (decimal)
C_STATE_NS_MASK_MODE_1 = 0100 0000b = 40H = 64 (decimal)
C_STATE_NS_MASK_MODE_2 = 0000 0000b = 00H = 0 (decimal)
C_STATE_NS_MASK_MODE_3_A = 1000 0000b = 80H = 128 (decimal)
C_STATE_NS_MASK_MODE_3_B = 1100 0000b = C0H = 192 (decimal)
C_STATE_NS_MASK_MODE_4_A = 1010 0000b = A0H = 160 (decimal)
C_STATE_NS_MASK_MODE_4_B = 1110 0000b = E0H = 224 (decimal)
C_STATE_NS_MASK_MODE_5 = 0010 0000b = 20H = 32 (decimal)

A functionality can be deactivated by mode mask calibration of 0FH.

Application conditions:

Initialisation: at reset:

CAN_R_RATIO_NOX_SENS[i] = 0
CAN_STATE_NS_TMP[i] = 0

CAN_SW_NS[i] = 0
CAN_HW_NS[i] = 0
CAN_STATE_DIAG_NS[i] = 0
CAN_NOX_REF_NS[i] = 0
CAN_NOX_DIAG_NS[i] = 0

Recurrence: 10ms

Activation: after reset, at every engine state

Deactivation: -

Formula section:

Signals, which are to copy from CAN message buffer without any condition
CAN_STATE_NOX_SENS[i] = *CAN Message buffer "Sensor status"*

CAN_STATE_NS_TMP[i] = CAN_STATE_NOX_SENS[i] **bit-wise AND** C_CAN_STATE_NS_TMP_MASK

```
case selection on CAN_STATE_NS_TMP[i]:
  case 1: CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_1
          Mode 1 – Wait on dew point  (Mask: 010x xxxx)
          Resistance ratio was received
          CAN_NOX_NOX_SENS[i] = 0
          if      CAN_R_RATIO_NOX_SENS[i](n-1) < CAN_TMP_WORD_1[i]          OR
                  LC_CAN_R_RATIO_NOX_SENS_CONT = 1
              then      CAN_R_RATIO_NOX_SENS[i] = CAN_TMP_WORD_1[i]
              else      Resistance ratio remains unchanged
                        CAN_R_RATIO_NOX_SENS[i](n) = CAN_R_RATIO_NOX_SENS[i](n-1)
          endif
          CAN_ERR_NOX_SENS[i] = CAN_TMP_BYTE_8[i]

          Signals, which are to copy from CAN message buffer
          CAN_LAMB_NOX_SENS[i] = CAN Message buffer "Lambda signal" *
          CAN_VLS_NOX_SENS[i] = CAN Message buffer "Binary O2 signal" *
          *...The CAN Message buffer value has to read out as signed integer value!

          Frozen signals (nothing is to do)
          CAN_SW_NS[i](n) = CAN_SW_NS[i](n-1)
          CAN_HW_NS[i](n) = CAN_HW_NS[i](n-1)
          CAN_STATE_DIAG_NS[i](n) = CAN_STATE_DIAG_NS[i](n-1)
          CAN_NOX_REF_NS[i](n) = CAN_NOX_REF_NS[i](n-1)
          CAN_NOX_DIAG_NS[i](n) = CAN_NOX_DIAG_NS[i](n-1)
```

case 2: CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_2

Mode 2 – Measuring (Mask: 000x xxxx)

NOx signal and error signal were received
CAN_NOX_NOX_SENS[i] = CAN_TMP_WORD_1[i]
CAN_ERR_NOX_SENS[i] = CAN_TMP_BYTE_8[i]

Signals, which are to copy from CAN message buffer
CAN_LAMB_NOX_SENS[i] = *CAN Message buffer "Lambda signal" **
CAN_VLS_NOX_SENS[i] = *CAN Message buffer "Binary O2 signal" **

*...The CAN Message buffer value has to read out as **signed integer** value!

Frozen signals (nothing is to do)
CAN_R_RATIO_NOX_SENS[i](n) = CAN_R_RATIO_NOX_SENS[i](n-1)
CAN_SW_NS[i](n) = CAN_SW_NS[i](n-1)
CAN_HW_NS[i](n) = CAN_HW_NS[i](n-1)
CAN_STATE_DIAG_NS[i](n) = CAN_STATE_DIAG_NS[i](n-1)
CAN_NOX_REF_NS[i](n) = CAN_NOX_REF_NS[i](n-1)
CAN_NOX_DIAG_NS[i](n) = CAN_NOX_DIAG_NS[i](n-1)

case 3: CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_3_A **OR**
CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_3_B

Mode 3 – Sensor identification data, part 1 (Mask: 1x0x xxxx -> 100... or 110...)

Sensor identification data, part 1 was received
- part number data are ignored
- software version is recorded
CAN_SW_NS[i] = CAN_TMP_BYTE_8[i]

Signals, which are to copy from CAN message buffer

Frozen signals (nothing is to do)
CAN_NOX_NOX_SENS[i](n) = CAN_NOX_NOX_SENS[i](n-1)
CAN_LAMB_NOX_SENS[i](n) = CAN_LAMB_NOX_SENS[i](n-1)
CAN_VLS_NOX_SENS[i](n) = CAN_VLS_NOX_SENS[i](n-1)
CAN_ERR_NOX_SENS[i](n) = CAN_ERR_NOX_SENS[i](n-1)
CAN_R_RATIO_NOX_SENS[i](n) = CAN_R_RATIO_NOX_SENS[i](n-1)
CAN_HW_NS[i](n) = CAN_HW_NS[i](n-1)
CAN_STATE_DIAG_NS[i](n) = CAN_STATE_DIAG_NS[i](n-1)
CAN_NOX_REF_NS[i](n) = CAN_NOX_REF_NS[i](n-1)
CAN_NOX_DIAG_NS[i](n) = CAN_NOX_DIAG_NS[i](n-1)

case 4: CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_4_A **OR**
CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_4_B

Mode 4 – Sensor identification data, part 2 (Mask: 1x1x xxxx -> 101... or 111...)

Sensor identification data, part 2 was received
- parameter list data is ingnored
- individual part number data are ignored
- quality status is recorded

CAN_HW_NS[i] = CAN_TMP_BYTE_8[i]

Signals, which are to copy from CAN message buffer

Frozen signals (nothing is to do)
CAN_NOX_NOX_SENS[i](n) = CAN_NOX_NOX_SENS[i](n-1)
CAN_LAMB_NOX_SENS[i](n) = CAN_LAMB_NOX_SENS[i](n-1)
CAN_VLS_NOX_SENS[i](n) = CAN_VLS_NOX_SENS[i](n-1)
CAN_ERR_NOX_SENS[i](n) = CAN_ERR_NOX_SENS[i](n-1)
CAN_R_RATIO_NOX_SENS[i](n) = CAN_R_RATIO_NOX_SENS[i](n-1)
CAN_SW_NS[i](n) = CAN_SW_NS[i](n-1)
CAN_STATE_DIAG_NS[i](n) = CAN_STATE_DIAG_NS[i](n-1)
CAN_NOX_REF_NS[i](n) = CAN_NOX_REF_NS[i](n-1)
CAN_NOX_DIAG_NS[i](n) = CAN_NOX_DIAG_NS[i](n-1)

case 5: CAN_STATE_NS_TMP[i] = C_STATE_NS_MASK_MODE_5

Mode 5 – Sensor self diagnosis (Mask: 001x xxxx)

Diagnosis signals were received
CAN_STATE_DIAG_NS[i] = CAN_TMP_BYTE_8[i]

```
if      CAN_STATE_DIAG_NS[i] <= C_STATE_DIAG_NS_REF
    then      Diagnosis reference value was received
              CAN_NOX_REF_NS[i] = CAN_TMP_WORD_1[i]
    else      Diagnosis value was received
              CAN_NOX_DIAG_NS[i] = CAN_TMP_WORD_1[i]
endif
```

Signals, which are to copy from CAN message buffer
CAN_LAMB_NOX_SENS[i] = *CAN Message buffer "Lambda signal"* *
CAN_VLS_NOX_SENS[i] = *CAN Message buffer "Binary O2 signal"* *

*...The CAN Message buffer value has to read out as **signed integer** value!

Frozen signals (nothing is to do)
CAN_NOX_NOX_SENS[i](n) = CAN_NOX_NOX_SENS[i](n-1)
CAN_ERR_NOX_SENS[i](n) = CAN_ERR_NOX_SENS[i](n-1)
CAN_R_RATIO_NOX_SENS[i](n) = CAN_R_RATIO_NOX_SENS[i](n-1)
CAN_SW_NS[i](n) = CAN_SW_NS[i](n-1)

CAN_HW_NS[i](n) = CAN_HW_NS[i](n-1)

default case:

This part is reached in case of wrong system calibration or at receive of not allowed "Sensor status" signal combination - Mask 011x xxxx. Nothing is done in this case. All signals are frozen.

Frozen signals (nothing is to do)
CAN_NOX_NOX_SENS[i](n) = CAN_NOX_NOX_SENS[i](n-1)
CAN_LAMB_NOX_SENS[i](n) = CAN_LAMB_NOX_SENS[i](n-1)
CAN_VLS_NOX_SENS[i](n) = CAN_VLS_NOX_SENS[i](n-1)
CAN_ERR_NOX_SENS[i](n) = CAN_ERR_NOX_SENS[i](n-1)
CAN_R_RATIO_NOX_SENS[i](n) = CAN_R_RATIO_NOX_SENS[i](n-1)
CAN_SW_NS[i](n) = CAN_SW_NS[i](n-1)
CAN_HW_NS[i](n) = CAN_HW_NS[i](n-1)
CAN_STATE_DIAG_NS[i](n) = CAN_STATE_DIAG_NS[i](n-1)
CAN_NOX_REF_NS[i](n) = CAN_NOX_REF_NS[i](n-1)
CAN_NOX_DIAG_NS[i](n) = CAN_NOX_DIAG_NS[i](n-1)

end case selection

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CAN_STATE_NS_TMP_MASK	1	0...FFH	0...255	1	[-]
Base bit-mask for selection of NOx sensor mode					
C_STATE_NS_MASK_MODE_1	1	0...FFH	0...255	1	[-]
Bit-mask for detection of NOx sensor data transfer mode 1 - Wait on dew point					
C_STATE_NS_MASK_MODE_2	1	0...FFH	0...255	1	[-]
Bit-mask for detection of NOx sensor data transfer mode 2 - Measuring					
C_STATE_NS_MASK_MODE_3_A	1	0...FFH	0...255	1	[-]
Bit-mask "A" for detection of NOx sensor data transfer mode 3 - Sensor identification data, part 1					
C_STATE_NS_MASK_MODE_3_B	1	0...FFH	0...255	1	[-]
Bit-mask "B" for detection of NOx sensor data transfer mode 3 - Sensor identification data, part 1					
C_STATE_NS_MASK_MODE_4_A	1	0...FFH	0...255	1	[-]
Bit-mask "A" for detection of NOx sensor data transfer mode 4 - Sensor identification data, part 2					
C_STATE_NS_MASK_MODE_4_B	1	0...FFH	0...255	1	[-]
Bit-mask "B" for detection of NOx sensor data transfer mode 4 - Sensor identification data, part 2					
C_STATE_NS_MASK_MODE_5	1	0...FFH	0...255	1	[-]
Bit-mask for detection of NOx sensor data transfer mode 5 - Sensor self diagnosis					
C_STATE_DIAG_NS_REF	1	0...FFH	0...255	1	[-]
Configuration label for distinction between "Diagnosis reference value" and "Diagnosis value"					
LC_CAN_R_RATIO_NOX_SENS_CONT	1	0...1H	0...1	1	[-]
CAN_R_RATIO_NOX_SENS calculation mode selection; 0..."follow-up" pointer; 1...continuous					

52.4.2.3 ID = 130H – NOx sensor 1 data and ID = 135H – NOx sensor 2 data

General information:

ATTENTION!!!

The CAN message description in this function specification is **for information only!** It is not included within NOx sensor component change management!

This messages transfers the measurement data, diagnosis data and sensor identification data from the NOx sensor 1 and NOx sensor 2 to the EMS.

The message content depends on sensor operation mode. It exists following sensor operation modes:

- Mode 1 – Wait on dew point
- Mode 2 – Measuring
- Mode 3 – Sensor identification data, part 1
- Mode 4 – Sensor identification data, part 2
- Mode 5 – Sensor self diagnosis

The sensor indicates its operation mode by the signal "Sensor status" which is included within each transmitted message.

Formula section:

CAN Frame and Signal Definitions:

Mode 1 – Wait on dew point: **Sensor status: 0 1 0 x x x x x (b)**

ID = 130H, base time = 10ms - NOx sensor 1 data
ID = 135H, base time = 10ms - NOx sensor 2 data

Bit:	7 MSB	6	5	4	3	2	1	0 LSB
Byte: 0	Resistance ratio signal, High byte							
1	Resistance ratio signal, Low byte							
2	Lambda signal, High byte							
3	Lambda signal, Low byte							
4	Binary O2 signal voltage, High byte							
5	Binary O2 signal voltage, Low byte							
6	Sensor status							
7	Error status							

ID	Signal	Hex Range	Phys. Range	Resolution	Unit	Init Value	Default Value
130H, 135H	Resistance ratio	8000H... 7FFFH	-32768... 32767	1	[-]	00H	
130H, 135H	Lambda signal	8000H... 7FFFH	-32768... 32767	1	[-]	00H	
130H, 135H	Binary O2 signal	8000H... 7FFFH	-32968... 32567	1	[mV]	00H	
130H, 135H	Sensor status[1)]	0...FFH	0...255	1	[-]	01H	01H
130H, 135H	Error status[2)]	0...FFH	0...255	1	[-]	0	0

NOx Sensor 1 - NC_NOX_SENS_CONF >= 1

ID	Signal	ECU signal variable name
130H	Resistance ratio	CAN_TMP_WORD_1[1]
130H	Lambda signal	CAN_LAMB_NOX_SENS[1]
130H	Binary O2 signal	CAN_VLS_NOX_SENS[1]
130H	Sensor status[1)]	CAN_STATE_NOX_SENS[1]
130H	Error status[2)]	CAN_TMP_BYTE_8[1]

NOx Sensor 2 - NC_NOX_SENS_CONF = 2

ID	Signal	ECU signal variable name
135H	Resistance ratio	CAN_TMP_WORD_1[2]
135H	Lambda signal	CAN_LAMB_NOX_SENS[2]
135H	Binary O2 signal	CAN_VLS_NOX_SENS[2]
135H	Sensor status[1)]	CAN_STATE_NOX_SENS[2]
135H	Error status[2)]	CAN_TMP_BYTE_8[2]

Mode 2 – Measuring: **Sensor status: 0 0 0 x x x x x (b)**

ID = 130H, base time = 10ms - NOx sensor 1 data
ID = 135H, base time = 10ms - NOx sensor 2 data

Bit:	7 MSB	6	5	4	3	2	1	0 LSB
Byte: 0	NOx signal, High byte							
1	NOx signal, Low byte							
2	Lambda signal, High byte							
3	Lambda signal, Low byte							
4	Binary O2 signal voltage, High byte							
5	Binary O2 signal voltage, Low byte							
6	Sensor status							
7	Error status							

ID	Signal	Hex Range	Phys. Range	Resolution	Unit	Init Value	Default Value
130H, 135H	NOx signal	8000H... 7FFFH	-32768... 32767	1	[ppm]	00H	
130H, 135H	Lambda signal	8000H... 7FFFH	-32768... 32767	1	[-]	00H	
130H, 135H	Binary O2 signal	8000H... 7FFFH	-32968... 32567	1	[mV]	00H	
130H, 135H	Sensor status[1)]	0...FFH	0...255	1	[-]	01H	01H
130H, 135H	Error status[2)]	0...FFH	0...255	1	[-]	0	0

NOx Sensor 1 - NC_NOX_SENS_CONF >= 1

ID	Signal	ECU signal variable name
130H	NOx signal	CAN_TMP_WORD_1[1]
130H	Lambda signal	CAN_LAMB_NOX_SENS[1]
130H	Binary O2 signal	CAN_VLS_NOX_SENS[1]
130H	Sensor status[1)]	CAN_STATE_NOX_SENS[1]
130H	Error status[2)]	CAN_TMP_BYTE_8[1]

NOx Sensor 2 - NC_NOX_SENS_CONF = 2

ID	Signal	ECU signal variable name
135H	NOx signal	CAN_TMP_WORD_1[2]
135H	Lambda signal	CAN_LAMB_NOX_SENS[2]
135H	Binary O2 signal	CAN_VLS_NOX_SENS[2]
135H	Sensor status[1)]	CAN_STATE_NOX_SENS[2]
135H	Error status[2)]	CAN_TMP_BYTE_8[2]

Mode 3 – Sensor identification data, part 1: Sensor status: 1 x 0 x x x x x (b)

ID = 130H, base time = 10ms - NOx sensor 1 data
ID = 135H, base time = 10ms - NOx sensor 2 data

Bit:	7 MSB	6	5	4	3	2	1	0 LSB
Byte: 0	Part number, byte 1 (MSB), (BCD coded)							
1	Part number, byte 2, (BCD coded)							
2	Part number, byte 3, (BCD coded)							
3	Part number, byte 4, (BCD coded)							
4	Part number, byte 5, (BCD coded)							
5	Part number, byte 6 (LSB), (BCD coded)							
6	Sensor status							
7	Software version							

ID	Signal	Hex Range	Phys. Range	Resolution	Unit	Init Value	Default Value
130H, 135H	Part number, byte 1 (MSB)	0... FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Part number, byte 2	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Part number, byte 3	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Part number, byte 4	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Part number, byte 5	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Part number, byte 6 (LSB)	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Sensor status1)	0...FFH	0...255	1	[-]	01H	01H
130H, 135H	Software version	0...FFH	0...255	1	[-]	0	0

NOx Sensor 1 - NC_NOX_SENS_CONF >= 1

ID	Signal	ECU signal variable name
130H	Part number, byte 1 (MSB)	CAN_TMP_BYTE_1[1]
130H	Part number, byte 2	CAN_TMP_BYTE_2[1]
130H	Part number, byte 3	CAN_TMP_BYTE_3[1]
130H	Part number, byte 4	CAN_TMP_BYTE_4[1]
130H	Part number, byte 5	CAN_TMP_BYTE_5[1]
130H	Part number, byte 6 (LSB)	CAN_TMP_BYTE_6[1]
130H	Sensor status1)	CAN_STATE_NOX_SENS[1]
130H	Software version	CAN_TMP_BYTE_8[1]

NOx Sensor 2 - NC_NOX_SENS_CONF = 2

ID	Signal	ECU signal variable name
135H	Part number, byte 1 (MSB)	CAN_TMP_BYTE_1[2]
135H	Part number, byte 2	CAN_TMP_BYTE_2[2]
135H	Part number, byte 3	CAN_TMP_BYTE_3[2]
135H	Part number, byte 4	CAN_TMP_BYTE_4[2]
135H	Part number, byte 5	CAN_TMP_BYTE_5[2]
135H	Part number, byte 6 (LSB)	CAN_TMP_BYTE_6[2]
135H	Sensor status1)	CAN_STATE_NOX_SENS[2]
135H	Software version	CAN_TMP_BYTE_8[2]

Mode 4 – Sensor identification data, part 2: Sensor status: 1 x 1 x x x x x (b)

ID = 130H, base time = 10ms - NOx sensor 1 data
ID = 135H, base time = 10ms - NOx sensor 2 data

Bit:	7 MSB	6	5	4	3	2	1	0 LSB
Byte: 0	Parameter list (BCD coded)							
1	Individual part number, byte 1 - year (BCD coded)							
2	Individual part number, byte 2 - month (BCD coded)							
3	Individual part number, byte 3 - day (BCD coded)							
4	Individual part number, byte 4 - number 1 (MSB), (BCD coded)							
5	Individual part number, byte 5 - number 2 (LSB), (BCD coded)							
6	Sensor status							
7	Quality status, (BCD coded)							

ID	Signal	Hex Range	Phys. Range	Resolution	Unit	Init Value	Default Value
130H, 135H	Parameter list	0... FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Individual part number, byte 1 - year	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Individual part number, byte 2 - month	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Individual part number, byte 3 - day	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Individual part number, byte 4 - number 1 (MSB)	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Individual part number, byte 5 - number 2 (LSB)	0...FFH	0...99 (BCD coded)	1	[-]	00H	00H
130H, 135H	Sensor status1)	0...FFH	0...255	1	[-]	01H	01H
130H, 135H	Quality status	0...FFH	0...99 (BCD coded)	1	[-]	0	0

NOx Sensor 1 - NC_NOX_SENS_CONF >= 1

ID	Signal	ECU signal variable name
130H	Parameter list	CAN_TMP_BYTE_1[1]
130H	Individual part number, byte 1 - year	CAN_TMP_BYTE_2[1]
130H	Individual part number, byte 2 - month	CAN_TMP_BYTE_3[1]
130H	Individual part number, byte 3 - day	CAN_TMP_BYTE_4[1]
130H	Individual part number, byte 4 - number 1 (MSB)	CAN_TMP_BYTE_5[1]
130H	Individual part number, byte 5 - number 2 (LSB)	CAN_TMP_BYTE_6[1]
130H	Sensor status1)	CAN_STATE_NOX_SENS[1]
130H	Quality status	CAN_TMP_BYTE_8[1]

NOx Sensor 2 - NC_NOX_SENS_CONF = 2

ID	Signal	ECU signal variable name
135H	Parameter list	CAN_TMP_BYTE_1[2]
135H	Individual part number, byte 1 - year	CAN_TMP_BYTE_2[2]
135H	Individual part number, byte 2 - month	CAN_TMP_BYTE_3[2]
135H	Individual part number, byte 3 - day	CAN_TMP_BYTE_4[2]
135H	Individual part number, byte 4 - number 1 (MSB)	CAN_TMP_BYTE_5[2]
135H	Individual part number, byte 5 - number 2 (LSB)	CAN_TMP_BYTE_6[2]
135H	Sensor status1)	CAN_STATE_NOX_SENS[2]
135H	Quality status	CAN_TMP_BYTE_8[2]

Mode 5 – Sensor self diagnosis: **Sensor status: 0 0 1 x x x x x (b)**

ID = 130H, base time = 10ms - NOx sensor 1 data
ID = 135H, base time = 10ms - NOx sensor 2 data

Bit:	7 MSB	6	5	4	3	2	1	0 LSB
Byte: 0	Diagnosis signal, High byte							
1	Diagnosis signal, Low byte							
2	Lambda signal, High byte							
3	Lambda signal, Low byte							
4	Binary O2 signal voltage, High byte							
5	Binary O2 signal voltage, Low byte							
6	Sensor status							
7	Diagnosis status							

ID	Signal	Hex Range	Phys. Range	Resolution	Unit	Init Value	Default Value
130H, 135H	Diagnosis signal4) - Diagnosis reference value - Diagnosis value	8000H... 7FFFH	-32768... 32767	1	[-]	00H	
130H, 135H	Lambda signal	8000H... 7FFFH	-32768... 32767	1	[-]	00H	
130H, 135H	Binary O2 signal	8000H... 7FFFH	-32968... 32567	1	[mV]	00H	
130H, 135H	Sensor status1)	0...FFH	0...255	1	[-]	01H	01H
130H, 135H	Diagnosis status3)	0...FFH	0...255	1	[-]	0	0

NOx Sensor 1 - NC_NOX_SENS_CONF >= 1

ID	Signal	ECU signal variable name
130H	Diagnosis signal4)	CAN_TMP_WORD_1[1]
130H	Lambda signal	CAN_LAMB_NOX_SENS[1]
130H	Binary O2 signal	CAN_VLS_NOX_SENS[1]
130H	Sensor status1)	CAN_STATE_NOX_SENS[1]
130H	Diagnosis status3)	CAN_TMP_BYTE_8[1]

NOx Sensor 2 - NC_NOX_SENS_CONF = 2

ID	Signal	ECU signal variable name
135H	Diagnosis signal[4)]	CAN_TMP_WORD_1[2]
135H	Lambda signal	CAN_LAMB_NOX_SENS[2]
135H	Binary O2 signal	CAN_VLS_NOX_SENS[2]
135H	Sensor status[1)]	CAN_STATE_NOX_SENS[2]
135H	Diagnosis status[3)]	CAN_TMP_BYTE_8[2]

[1] The following table describes the meaning of byte "Sensor status":

Byte: "Sensor status"								
B7	B6	B5	B4	B3	B2	B1	B0	Function
							X	Power supply bit. 0 ... 11.5V <= VB <= 15.5V 1 ... VB < 11.5V or VB > 15.5V) VB...battery voltage measured by NOx sensor ECU inside of NOx sensor ECU
						X		Sensor temperature indication bit. 0 ... Sensor temperature is not ok (out of operating range) 1 ... Sensor temperature is ok (within operating range)
					X			Ip1 ready indication bit. 0 ... not ok (out of operating range) 1 ... ok (within operating range)
				X				Ip0 ready indication bit. 0 ... not ok (out of operating range) 1 ... ok (within operating range)
			X					Ip2 ready indication bit. 0 ... not ok (out of operating range) 1 ... ok (within operating range)
		X						Mode switch for read out of sensor identification data and sensor self diagnosis mode For details see NOx sensor specification. Bit B7 = 0: 0 ... Normal sensor operation mode 1 ... Sensor self diagnosis mode Bit B7 = 1 (sensor identification data read out mode): 0 ... Sensor transmits sensor identification data part 1 1 ... Sensor transmits sensor identification data part 2
	X							Dew point bit. 0 ... Sensor has received dew point Byte 1 and 2 contains NOx signal. 1 ... Sensor is waiting for dew point

								Byte 1 and 2 contains resistance ratio.
X								Mode switch for read out of sensor identification data. For details see NOx sensor specification. 0 = normal sensor operation or sensor self diagnosis mode 1 = sensor identification data read out mode
								Important combinations:
0	0	0	0	0	0	0	0	0 (00H): NOx sensor is not yet active.
0	1	0	0	0	0	0	0	64 (40H): NOx sensor is waiting for dew point.
0	0	0	0	0	0	0	1	1 (01H): NOx sensor can not reach any readiness level because of power supply problem
0	0	0	0	0	0	1	0	2 (02H): NOx sensor readiness is level 1. Temperature is ok.
0	0	0	0	1	1	1	0	14 (0EH): NOx sensor readiness is level 2. Lambda and binary O2 signal are valid.
0	0	0	1	1	1	1	0	30 (1EH): NOx sensor has full readiness. NOx, Lambda and binary O2 signals are valid.
0	0	0	1	1	1	1	1	31 (1FH): NOx sensor has full readiness. All signals are valid. But a power supply problem is present.

[2] The following table describes the meaning of byte "Error status":

Byte: "Error status"								
B7	B6	B5	B4	B3	B2	B1	B0	Function
0	0	0	0	0	0	0	0	NOx sensor no failure
							X	NOx sensor error 1, Open wire at heater
						X		NOx sensor error 2, Open wire at NOx function
					X			NOx sensor error 3, Open wire at Lambda function
				X				NOx sensor error 4, Open wire at binary O2 function
			X					NOx sensor error 5, Short circuit at heater
		X						NOx sensor error 6, Short circuit at NOx function
	X							NOx sensor error 7, Short circuit at Lambda function
X								NOx sensor error 8, Short circuit at binary O2 function

[3] The following table describes the meaning of byte "Diagnosis status":

Byte: "Diagnosis status"	
Value	Function
0 (00H)	Sensor self diagnosis is running
1 (01H)	Sensor self diagnosis was aborted. Sensor is returning to normal operation mode.
2 (02H)	Sensor self diagnosis was completed. Result: "Sensor is ok". Sensor is returning to normal operation mode.
3 (03H)	Sensor self diagnosis was completed. Result: "Sensor is faulty". Sensor is returning to normal operation mode.
other values	Other values are not defined yet.

[4] The following table describes the meaning of byte "Diagnosis signal":

Byte: "Diagnosis status"		Byte: "Diagnosis signal"	
Value	Function	Signal	ECU signal variable name
0 (00H)	Diagnosis runs	Diagnosis reference value	CAN_NOX_REF_NS[i]
1 (01H)	Diagnosis aborted	Diagnosis reference value	CAN_NOX_REF_NS[i]
2 (02H)	Diagnosis completed	Diagnosis value	CAN_NOX_DIAG_NS[i]
3 (03H)	Diagnosis completed	Diagnosis value	CAN_NOX_DIAG_NS[i]
other values	not defined	not defined	not defined

52.4.2.3.1 Selection of NOx sensor configuration

General information:

The EMS system MSD80 is designed for various exhaust gas system configurations (e.g. N43UL – 1 branch system, N43OL – Y system with 1 NOx catalyst, N53 – Y system with 2 NOx catalysts, N53 development samples – full 2 branch system). Hence the adjusted value of NC_NOX_SENS_CONF (number of NOx sensors within the system) does not correspond always correctly with the number of really necessary or really used NOx sensors. Therefore NC_NOX_SENS_CONF is adjusted mostly on 2 NOx sensors even if only 1 NOx sensor is really necessary or really used.

example: MSD80 – N53

1. full 2 branch system with 2 NOx sensors:
   NC_NOX_SENS_CONF has to be adjusted to 2.
2. Y system with 2 NOx catalysts and 1 NOx sensor:
   NC_NOX_SENS_CONF should be adjusted to 1 but is adjusted to 2 because it is the same software like as for full 2 branch system.

So that these different exhaust system configurations can be handled with 1 ECU software it was implemented a functionality for copying of NOx sensor signals from line i = 1 to line i = 2.

Application conditions:

Initialisation: -

Recurrence: 10ms

Activation: after reset

Deactivation: -

Formula section:

```
if    NC_NOX_SENS_CONF = 2
then
        if    LC_NOX_SENS_2 = 0
        then
              CAN_NOX_NOX_SENS_2 = CAN_NOX_NOX_SENS_1
              CAN_LAMB_NOX_SENS_2 = CAN_LAMB_NOX_SENS_1
              CAN_VLS_NOX_SENS_2 = CAN_VLS_NOX_SENS_1
              CAN_STATE_NOX_SENS_2 = CAN_STATE_NOX_SENS_1
              CAN_ERR_NOX_SENS_2 = CAN_ERR_NOX_SENS_1
        else
              CAN_NOX_NOX_SENS_2 = CAN message data from ID=135H
              CAN_LAMB_NOX_SENS_2 = CAN message data from ID=135H
              CAN_VLS_NOX_SENS_2 = CAN message data from ID=135H
              CAN_STATE_NOX_SENS_2 = CAN message data from ID=135H
              CAN_ERR_NOX_SENS_2 = CAN message data from ID=135H
        endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NOX_SENS_2	1	0...1H	0...1	1	[-]
NOx sensor of exhaust line 2 exists (yes/ no)					

52.5 NOX signal gain adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_NS_AD_CYC[NC_NOX_SENS_CONF]	O/V/S	0...FFFFH	0...65535	1	[-]
Counter of NOx signal gain adaptations					
DIST_NT_NS_AD[NC_NOX_SENS_CONF]	O/V/S	0...FFFFH	0...524280	8	[km]
Current distance from last NOx signal gain adaptation					
FAC_NOX_NS_AD[NC_NOX_SENS_CONF]	O/V/S	0...FFFFH	0...127.99804	1.9531e-3	[-]
Adaptation of the NOx sensor characteristic shift					
LV_NS_AD_REQ	O/V	0...1H	0...1	1	[-]
Request for NOx signal gain adaptation					
STATE_OPM_REQ_NS_AD	O/V	0...FFFFH	0...65535	1	[-]
Request for operation mode for executing of NOx sensor adaptation					

Input Data:

NC_NOX_SENS_CONF			

Export Actions:

ACTION_NOXD_CleanNSAdapt()
Initialization of NOx signal shift diagnosis variables
ACTION_NOXD_WriteNSGainDiagExtAdj()
Write external adjust values for NOx signal shift diagnosis

Description for Actions

ACTION_NOXD_CleanNSAdapt()
Initialization of NOx signal shift diagnosis variables

ACTION_NOXD_WriteNSGainDiagExtAdj()
Write external adjust values for NOx signal shift diagnosis

FUNCTION DESCRIPTION:

General Information

Stub module for recource optimization

Application Conditions

Initialization: at reset
CTR_NS_AD_CYC[i] = 0
DIST_NT_NS_AD[i] = 0
FAC_NOX_NS_AD[i] = 1
LV_NS_AD_REQ[i] = 0
STATE_OPM_REQ_NS_AD[i] = 0

Recurrence:

Activation:

Deactivation:

52.5.1 Detailed description for Action: ACTION_NOXD_CleanNSAdapt

FUNCTION DESCRIPTION:

52.5.2 Detailed description for Action: ACTION_NOXD_WriteNSGainDiagExtAdj

FUNCTION DESCRIPTION:

52.6 NOx sensor setpoint shift diag - lambda intervention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_LAM_ADJ_NS_SHIFT_DIAG[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	[-]
Flag to switch trim control parameters for NS shift diagnosis					
VLS_DELTA_LAM_ADJ_NS_SHIFT[NC_CBK_EX_NR]	O/V	FC00...3FFH	-5...4.99511	4.8828e-3	[V]
Sensor voltage set point shift in case of NOx sensor shift diagnosis					
VLS_DELTA_TMP_NS_SHIFT[NC_CBK_EX_NR]	V	FC00...3FFH	-5...4.99511	4.8828e-3	[V]
Temporary sensor voltage set point shift for NOx sensor shift diagnosis					

Input data:

VLS_SP_LAM_ADJ[NC_CBK_EX_NR]	LV_NS_SHIFT_CMB_INT_REQ	NC_CBK_EX_NR	OPM_REQ_CUS

FUNCTION DESCRIPTION:

General information:

This module influences the trim control set-point during NOx sensor shift diagnosis by setting an offset to the actual value for the trim controller.

Calculation has to be done for each trim controller: i = 1 ... NC_CBK_EX_NR

Application conditions:

Recurrence: 100 ms

Activation: --

Deactivation: --

Initialisation at reset:

LV_LAM_ADJ_NS_SHIFT_DIAG[i] = 0
VLS_DELTA_LAM_ADJ_NS_SHIFT[i] = 0 V
VLS_DELTA_TMP_NS_SHIFT[i] = 0 V

Formula section:

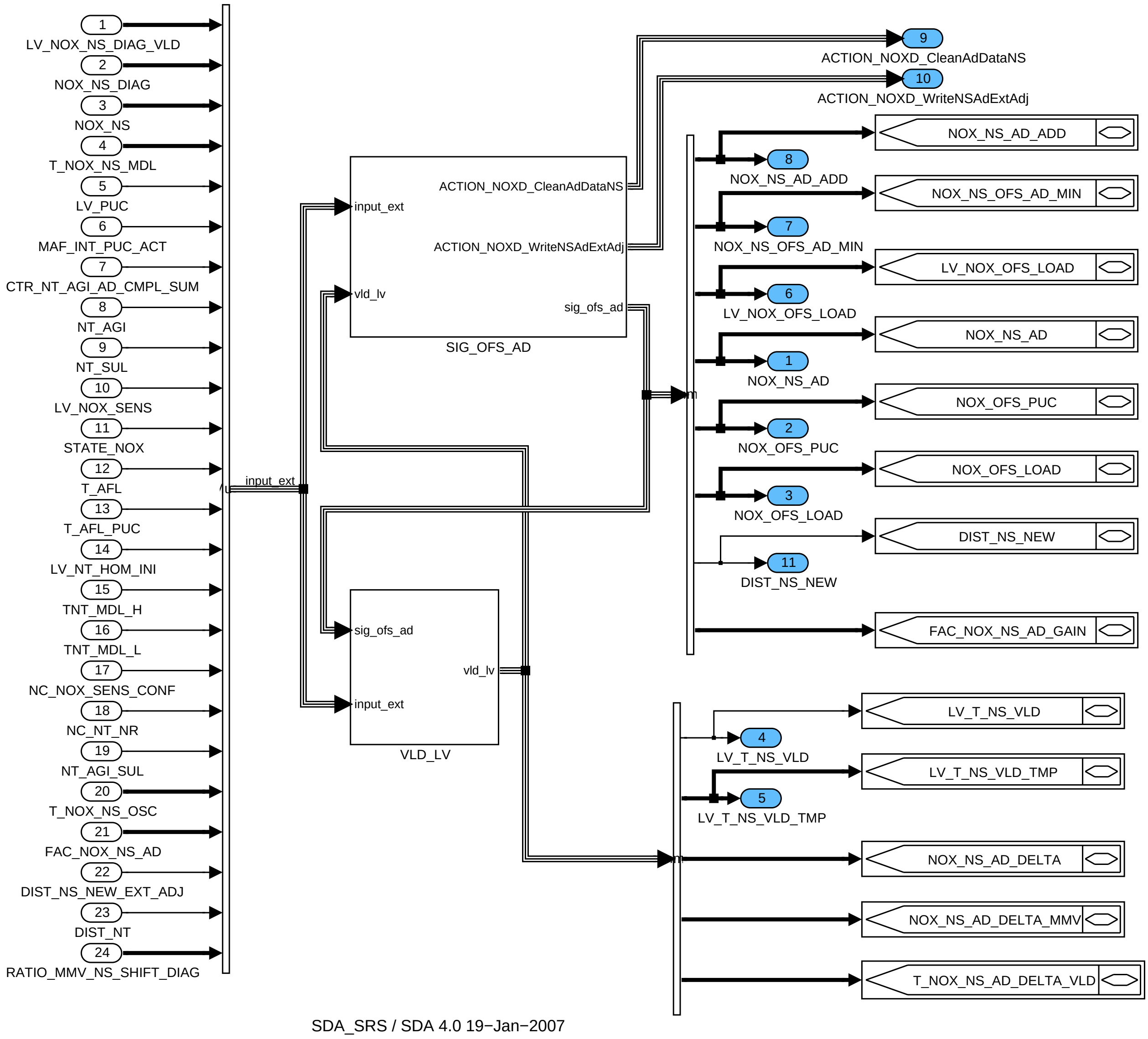
```
VLS_DELTA_TMP_NS_SHIFT[i] = VLS_SP_LAM_ADJ[i] - C_VLS_SP_NS_SHIFT[i]

If      VLS_DELTA_TMP_NS_SHIFT[i] > C_VLS_DELTA_LAM_NS_MAX[i]
Then    VLS_DELTA_TMP_NS_SHIFT[i] = C_VLS_DELTA_LAM_NS_MAX[i]
ElseIf  VLS_DELTA_TMP_NS_SHIFT[i] < C_VLS_DELTA_LAM_NS_MIN[i]
Then    VLS_DELTA_TMP_NS_SHIFT[i] = C_VLS_DELTA_LAM_NS_MIN[i]
Endif

If      LC_VLS_LAM_ADJ_NS_SHIFT[i]                              AND
        LV_NS_SHIFT_CMB_INT_REQ = 1                              AND
        (OPM_REQ_CUS = C_STATE_NS_SHIFT_CMB_ACT   OR
        C_STATE_NS_SHIFT_CMB_ACT = 0FFFFh)
Then    LV_LAM_ADJ_NS_SHIFT_DIAG[i] = 1
        VLS_DELTA_LAM_ADJ_NS_SHIFT[i] = VLS_DELTA_TMP_NS_SHIFT[i]
Else    LV_LAM_ADJ_NS_SHIFT_DIAG[i] = 0
        VLS_DELTA_LAM_ADJ_NS_SHIFT[i] = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VLS_SP_NS_SHIFT[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Trim controller set-point for NOx sensor shift diagnosis					
C_VLS_DELTA_LAM_NS_MAX[NC_CBK_EX_NR]	1	FC00...3FFH	-5...4.99511	4.8828e-3	[V]
Maximum offset for trim controller set-point for NOx sensor shift diagnosis					
C_VLS_DELTA_LAM_NS_MIN[NC_CBK_EX_NR]	1	FC00...3FFH	-5...4.99511	4.8828e-3	[V]
Minimum offset for trim controller set-point for NOx sensor shift diagnosis					
C_STATE_NS_SHIFT_CMB_ACT	1	0...FFFFH	0...65535	1	[-]
Check if combustion mode for NOx signal internal shift diagnosis is active (FFFFh means check is disabled)					
LC_VLS_LAM_ADJ_NS_SHIFT[NC_CBK_EX_NR]	1	0...1H	0...1	1	[-]
Enable offset for trim control at NOx sensor shift diagnosis					

52.7 NOx – Sensor function

Overview



SDA_SRS / SDA 4.0 19−Jan−2007

Figure 1 NOXD_MDLADNSFCN0

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NOX_NS_AD[NC_NOX_SENS_CONF]	O/V	0...5DCH	0...1.5E+3	1	ppm
NOx concentration in the exhaust gas downstream of NOx catalyst					
NOX_OFS_PUC[NC_NOX_SENS_CONF]	O/V/S	FF9C...5DCH	-100...1.5E+3	1	ppm
NOx signal offset value, measured at PUC					
NOX_OFS_LOAD[NC_NOX_SENS_CONF]	O/V/S	FF9C...5DCH	-100...1.5E+3	1	ppm
NOx signal offset value, measured after start of a LOAD phase					
LV_T_NS_VLD	O/V	0...1H	0...1	1	-
NOx sensor signal valid after regeneration					
LV_T_NS_VLD_TMP[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
NOx sensor signal valid after regeneration (bank selective)					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_NOX_OFS_LOAD[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
NOx minimum observation activation					
NOX_NS_OFS_AD_MIN[NC_NOX_SENS_CONF]	O/V	FF9C...5DCH	-100...1.5E+3	1	ppm
Minumum NOx signal after LOAD phase					
NOX_NS_AD_ADD[NC_NOX_SENS_CONF]	O/V	FF9C...5DCH	-100...1.5E+3	1	ppm
Offset adaptation value for the NOx signal					
DIST_NS_NEW	O/V/S	0...FFFFH	0...5.2428E+5	8	km
mileage counter value of last exchange of NOx sensor					
T_NOX_NS_AD_DELTA_VLD[NC_NOX_SENS_CONF]	V	0...FFH	0...25.5	0.1	s
Timer for NOx sensor signal valid regarding gradient observation					
FAC_NOX_NS_AD_GAIN[NC_NOX_SENS_CONF]	V	0...FFFFH	0...127.998047	0.00195313	-
Correction of the NOx sensor signal due to signal shift					
NOX_NS_AD_DELTA_MMV[NC_NOX_SENS_CONF]	V	FA24...5DCH	-1.5E+3...1.5E+3	1	ppm
NOx sensor signal gradient moving mean value (bank selective)					
NOX_NS_AD_DELTA[NC_NOX_SENS_CONF]	V	FA24...5DCH	-1.5E+3...1.5E+3	1	ppm
NOx sensor signal gradient (bank selective)					

Input data:

LV_NOX_NS_DIAG_VLD[NC_NOX_SENS_CONF]	NOX_NS_DIAG[NC_NOX_SENS_CONF]	NOX_NS[NC_NOX_SENS_CONF]	T_NOX_NS_MDL[NC_NOX_SENS_CONF]
LV_PUC	MAF_INT_PUC_ACT	CTR_NT_AGI_AD_CMPL_SUM	NT_AGI
NT_SUL	LV_NOX_SENS	STATE_NOX	T_AFL
T_AFL_PUC	LV_NT_HOM_INI	TNT_MDL_H	TNT_MDL_L
NC_NOX_SENS_CONF	NC_NT_NR	NT_AGI_SUL	T_NOX_NS_OSC[NC_NOX_SENS_CONF]
FAC_NOX_NS_AD[NC_NOX_SENS_CONF]	DIST_NS_NEW_EXT_ADJ	DIST_NT	RATIO_MMV_NS_SHIFT_DIAG[NC_NOX_SENS_CONF]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_NOX_NS_AD_DELTA	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Correlation factor to calculate moving mean value of the NOx sensor signal gradient					
C_CRLC_NOX_OFS_PUC	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Correlation constant for calculation of PUC offset value					
C_CTR_NT_AGI_AD_CMPL_SUM_AD_MIN	1	0...FFFFH	0...6.5535E+4	1	-
Minimum completed sulphur adaptations for set free of NOx signal offset adaptation					
C_FAC_NOX_NS_AD_MAX	1	0...FFFFH	0...127.998047	0.00195313	-
Maximal allowed correction of characteristic shift down of NOx signal					
C_FAC_NOX_NS_AD_MIN	1	0...FFFFH	0...127.998047	0.00195313	-
Minimal allowed correction of characteristic shift down of NOx signal					
C_MAF_INT_PUC_MIN_NOX	1	0...FFFFH	0...2.91267E+3	0.04444444	g
Minimum air mass flow threshold for starting the NOx signal offset adaptation at PUC					
C_NOX_ADD_ZERO_MAX	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Upper NOx signal offset adaptation value threshold to set the adaptation value to 0					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NOX_ADD_ZERO_MIN	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Lower NOx signal offset adaptation value threshold to set the adaptation value to 0					
C_NOX_AD_ADD_MAX	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Maximum limit of NOx signal offset adaptation value					
C_NOX_AD_ADD_MIN	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Minimum limit of NOx signal offset adaptation value					
C_NOX_NS_AD_ADD[NC_NOX_SENS_CONF]	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Manual NOx signal offset					
C_NOX_NS_AD_DELTA_MMV_MAX	1	FA24...5DCH	-1.5E+3...1.5E+3	1	ppm
Maximum allowed NOx sensor signal gradient					
C_NOX_NS_AD_DELTA_MMV_MIN	1	FA24...5DCH	-1.5E+3...1.5E+3	1	ppm
Minimum allowed NOx sensor signal gradient					
C_NOX_NS_AD_MAX_SENS_VLD	1	0...5DCH	0...1.5E+3	1	ppm
Maximum NOx sensor signal to enable gradient observation					
C_NOX_OFS_PUC_MAX	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Maximum NOx signal offset value at PUC phase					
C_NOX_OFS_SEL	1	0H 1H 2H 3H	PUC_OFFSET LOAD_OFFSET MIN_OFFSET MAN_OFFSET	1	-
Selection of a NOx signal adaptation method					
C_NT_AGI_NOX_OFS_MIN	1	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx catalyst aging threshold for calculation of a new NOx signal offset value					
C_NT_AGI_SUL_NOX_OFS_AD_MIN	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Minimum NOx catalyst aging factor due to sulphur for set free of NOx signal offset adaptation					
C_NT_HOM_INI_USE	1	0...3H	0...3	1	-
Configutation of adaptation function					
C_NT_SUL_NOX_OFS_AD_MAX	1	0...FFFFH	0...1.04856E+4	0.16	mg
Maximum NOx catalyst sulphur load for set free of NOx signal offset adaptation					
C_TNT_MDL_NOX_OFS_MAX	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Maximum temperature of NOx catalyst, at which the NOx signal offset adaptation at LOAD is active					
C_TNT_MDL_NOX_OFS_MIN	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Minimum temperature of NOx catalyst, at which the NOx signal offset adaptation at LOAD is active					
C_T_AFL_NOX_OFS_MIN	1	0...FFFFH	0...1.3107E+3	0.02	s
Min. lean time for calculation of NOx signal offset adaptation value					
C_T_NOX_NS_AD_DELTA_MIN	1	0...FFH	0...25.5	0.1	s
Minimum time for set NOx-Sensor signal to valid regarding NOx-Signal gradient observation					
C_T_NOX_NS_OFS	1	0...FFFFH	0...655.35	0.01	s
Min. NOx signal time for enable of NOx signal offset adaptation					
C_T_NOX_NS_OFS_PUC	1	0...FFFFH	0...655.35	0.01	s
Minimum time NOx-Signal is valid for enable of NOx signal offset adaptation in PUC phase					
C_T_NOX_NS_OSC_OFS	1	0...FFFFH	0...655.35	0.01	s
Waiting time after last detection of an oscillation of NOx signal for offset adaptation					
C_T_NS_VLD	1	0...FFFFH	0...1.3107E+3	0.02	s
Minimum lean time to set NOx sensor signal to valid					
C_T_NS_VLD_PUC	1	0...FFFFH	0...1.3107E+3	0.02	s
Minimum PUC time to set NOx sensor signal to valid					
LC_NOX_NS_OFS_VLD_ON	1	0...1H	0...1	1	-
Valid NOx-Sensor signal is necessary for this offset adaptation (on/ off)					
IP_CRLC_NOX_OFS_LOAD	4	0...FFFFH	0...0.99998474	1.52588E-5	-
LDP_NOX_NS_OFS_AD_MIN_IP_LOAD	4	0...640H	-100...1.5E+3	1	ppm
Correlation constant for calculation of LOAD offset value					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_RATIO_NS_SHIFT_COR	6	0...FFFFH	0...127.998047	0.00195313	-
LDP_RATIO_NS_SHIFT_IP_FAC_COR	6	0...FFFFH	-1...0.99996948	3.05176E-5	-
NOx signal correction factor depending on RATIO					
IP_FAC_NOX_NS_AD_COR	6x6	0...FFFFH	0...127.998047	0.00195313	-
LDP_NOX_NS_AD_IP_FAC_COR	6	0...5DCH	0...1.5E+3	1	ppm
LDP_FAC_NOX_NS_AD_IP_FAC_NOX	6	0...FFFFH	0...127.998047	0.00195313	-
Correction of gain adaptation factor due to non-linearity of the characteristic shift down					

Export actions:

ACTION_NOXD_CleanAdDataNS(OUT <trig_CLR_AD>)
Clean adaptation data of NOx Sensor
ACTION_NOXD_WriteNSAdExtAdj(OUT <trig_EXT_ADJ>)
Write external adjust values for NOx Sensor

52.7.1 NOx signal offset adaptation

NOx signal offset adaptation can be done at PUC phase. At PUC phase NOx signal is influenced by the humidity cross-sensitivity of NOx-Sensor. The NOx signal is a combination out of signal offset and humidity cross-sensitivity.

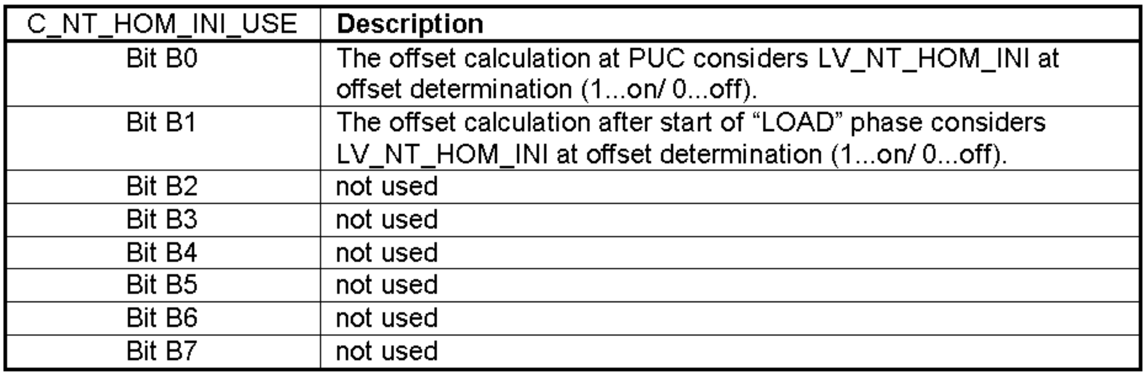
Another possibility for NOx signal offset adaptation is after the start of a new NOx catalyst “LOAD” phase. At this time the NOx catalyst stores the NOx so that the NOx emissions are nearly to 0 [ppm]. Hence the NOx signal is dominated by signal offset. Attention! After an incompletely NOx regeneration or when the NOx catalyst has got a high sulphur load this offset adaptation must be inhibited.

This NOx signal offset adaptation calculates all offset values. Which offset value is used for the NOx signal offset adaptation, is possible to calibrate.

The NOx signal NOX_NS_i is compared with C_NOX_OFS_PUC_MAX before a new adaptation value (NOX_OFS_PUC_i) is calculated. This check is for prevention of adaptation failures.

C_NOX_OFS_PUC_MAX is the sum of the maximum humidity cross-sensitivity and the maximum positive NOx signal offset according to the sensor specification.

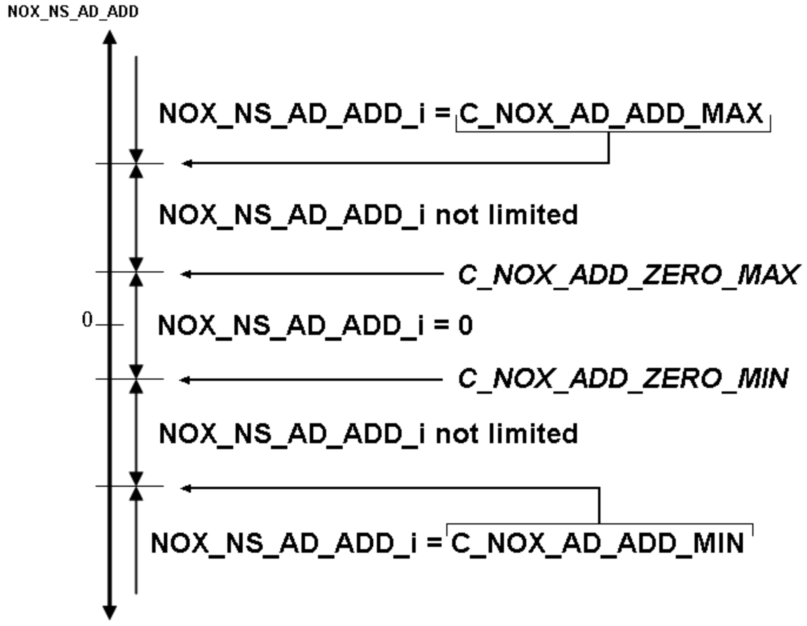
The calibration constant C_NT_HOM_INI_USE is used for independent configuration of “offset calculation at PUC” and of “offset calculation after start of LOAD phase”.



C_NT_HOM_INI_USE	**Description**
Bit B0	The offset calculation at PUC considers LV_NT_HOM_INI at offset determination (1...on/ 0...off).
Bit B1	The offset calculation after start of “LOAD” phase considers LV_NT_HOM_INI at offset determination (1...on/ 0...off).
Bit B2	not used
Bit B3	not used
Bit B4	not used
Bit B5	not used
Bit B6	not used
Bit B7	not used

The final NOx signal offset adaptation value NOX_NS_AD_ADD _i arises out of PUC offset value (NOX_OFS_PUC_i), LOAD offset value (NOX_OFS_LOAD_i) or manual offset value (C_ NOX_NS_AD_ADD_i). The used value is selected by C_NOX_OFS_SEL.

After the calculation of NOX_NS_AD_ADD_i it is limited (see below).



Setting NOX_NS_AD_ADD_i to 0 within the gap C_NOX_ADD_ZERO_MIN < NOX_NS_AD_ADD_i < C_NOX_ADD_ZERO _MAX was implemented in order to switch off the NOx signal offset adaptation if the calculated offset value is within the specified NOx signal tolerance (see NOx-Sensor specification).

NOX_NS_AD_i is the measured signal which is corrected by the signal offset NOX_NS_AD_ADD_i.

The calculation of NOX_NS_AD_i is done with a recurrence, which is in accordance with the NOx catalyst regeneration request calculation.

Additional information:

Suffix "_cur" is an abbreviation of current and means that the value is calculated in the same recurrence not first time.

Suffix "_old" means that this value is calculated last time in previous recurrence.

Application Condition

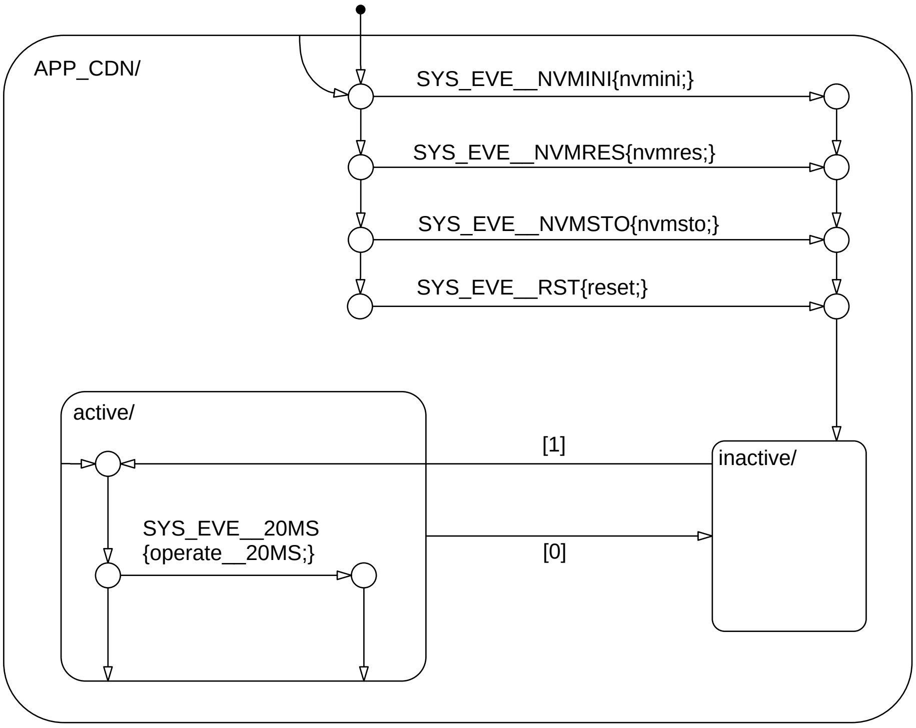


Figure 2 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ APP_CDN/ Chart

Function Description

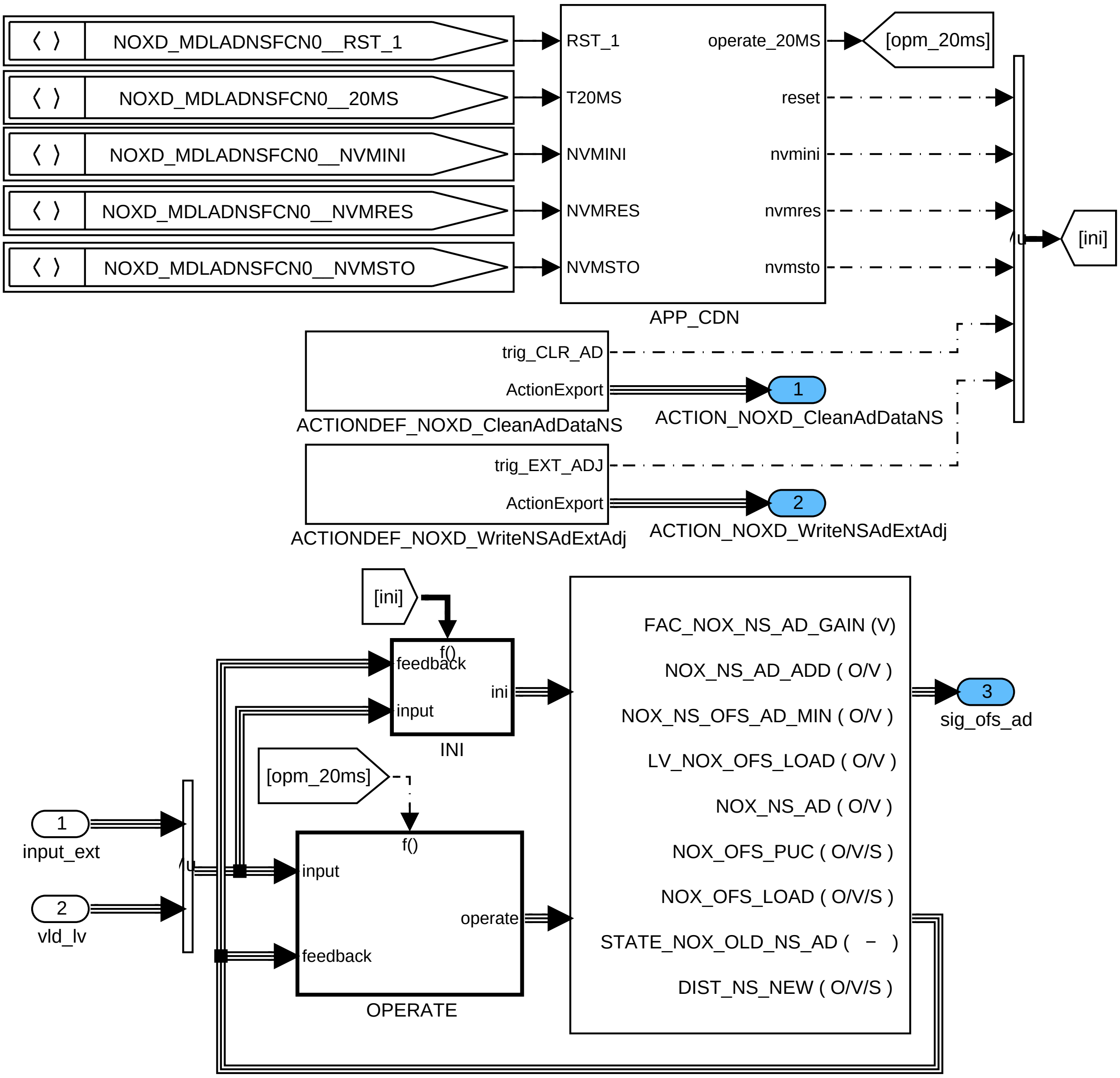


Figure 3 NOXD_MDLADNSFCN0/ SIG_OFS_AD

52.7.1.1 SUBFUNCTION: ACTIONDEF_NOXD_CleanAdDataNS

Description for ACTION_NOXD_CleanAdDataNS

ACTION_NOXD_CleanAdDataNS(OUT <trig_CLR_AD>)

Clean adaptation data of NOx Sensor

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit
trig_CLR_AD					

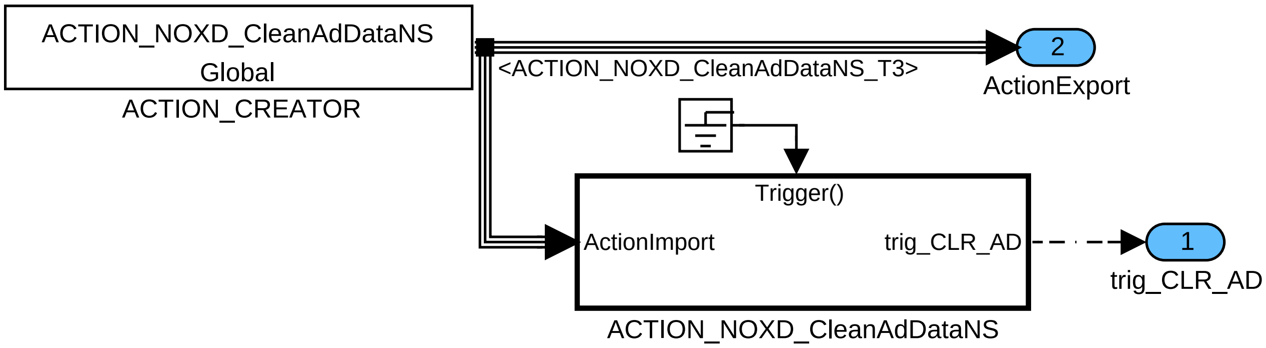


Figure 4 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ ACTIONDEF_NOXD_CleanAdDataNS

ACTION_NOXD_CLEANADDATANS

This action generates a function call to clear adaptation values.

Figure 5 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ ACTIONDEF_NOXD_CleanAdDataNS/ ACTION_NOXD_CleanAdDataNS

52.7.1.2 SUBFUNCTION: ACTIONDEF_NOXD_WriteNSAdExtAdj

Description for ACTION_NOXD_WriteNSAdExtAdj

ACTION_NOXD_WriteNSAdExtAdj(OUT <trig_EXT_ADJ>)

Write external adjust values for NOx Sensor

Parameter	Type	Hex. limits	Phys. limits	Resol.	Unit
trig_EXT_ADJ					

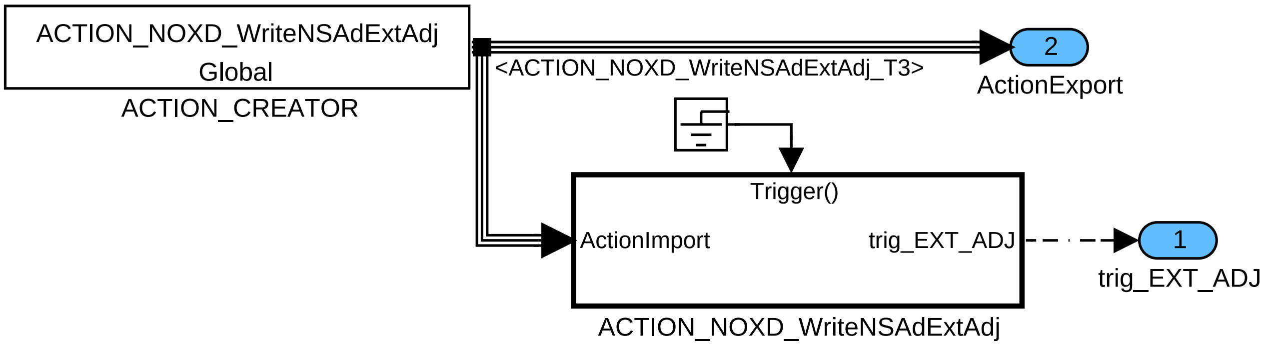


Figure 6 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ ACTIONDEF_NOXD_WriteNSAdExtAdj

ACTION_NOXD_WRITENSADEXTADJ

This action generates a function call to set adaptation values by external values.

Figure 7 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ ACTIONDEF_NOXD_WriteNSAdExtAdj/ ACTION_NOXD_WriteNSAdExtAdj

52.7.1.3 INITIALIZATION

At reset all variables except NVMY data are initialized with "0"

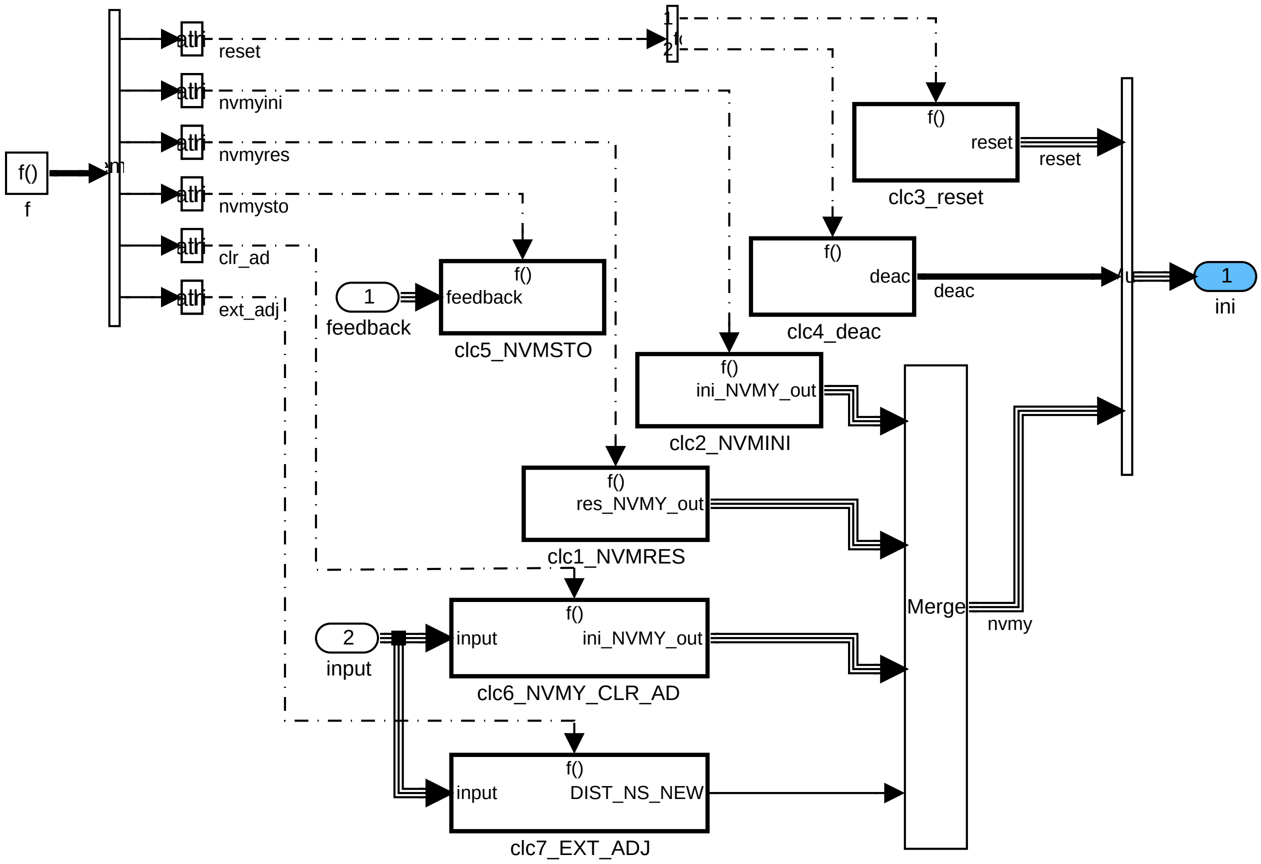


Figure 8 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ INI

Initialization of NVMY data

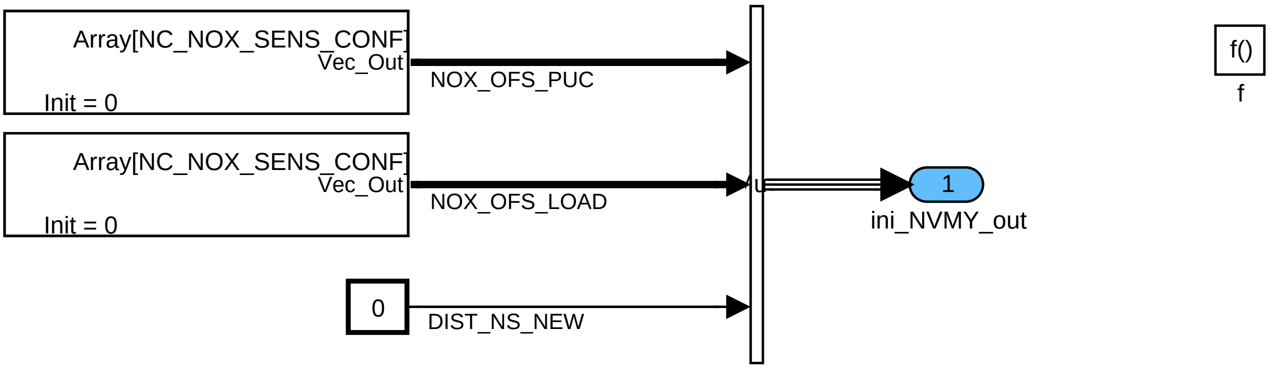


Figure 9 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ INI/ clc2_NVMINI

Initialization at deactivation

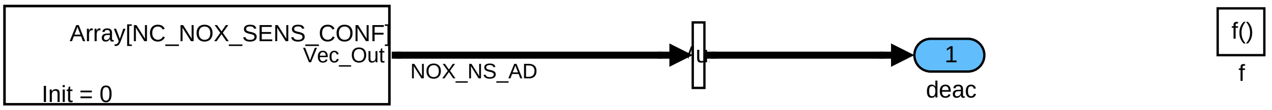


Figure 10 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ INI/ clc4_deac

Initialization if new sensor was built-in

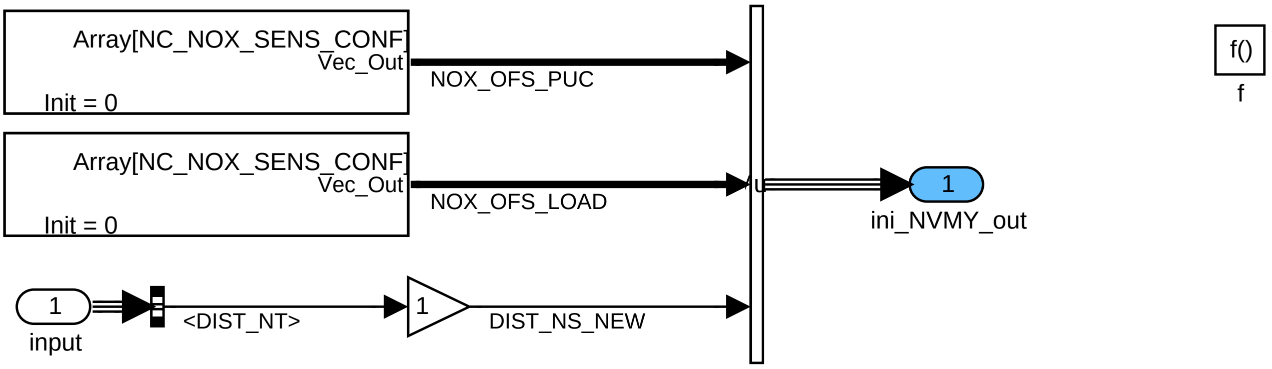


Figure 11 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ INI/ clc6_NVMY_CLR_AD

Initialization at external adjust

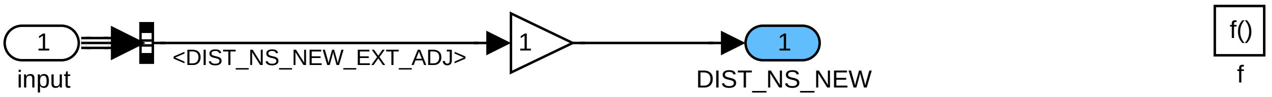


Figure 12 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ INI/ clc7_EXT_ADJ

52.7.1.4 FORMULA SECTION

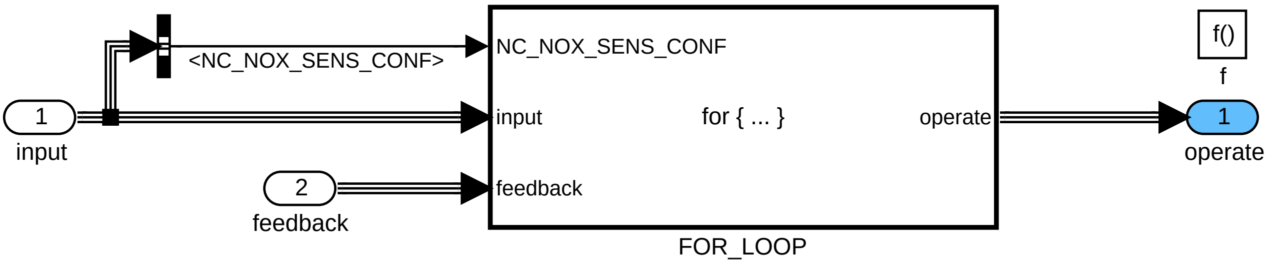


Figure 13 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE

For loop realization

For loop is realized using number of NOx Sensors (NC_NOX_SENS_CONF). The index for NOx Specific variables is "i".

Formula section for NOx Sensor specific calculations

Formula section contains:

SECTION 1: Offset value calculation

SECTION 2: Case selection for finally adaptation value calculation

SECTION 3: Limitation of chose adaptation value

SECTION 4: Realization of adaptation on measured signal

SECTION 1: Offset value calculation

This section contains:

SECTION 1.1: Offset calculation at PUC phase

SECTION 1.2: Offset calculation at LOAD phase

SECTION 1.1: Offset calculation at PUC phase

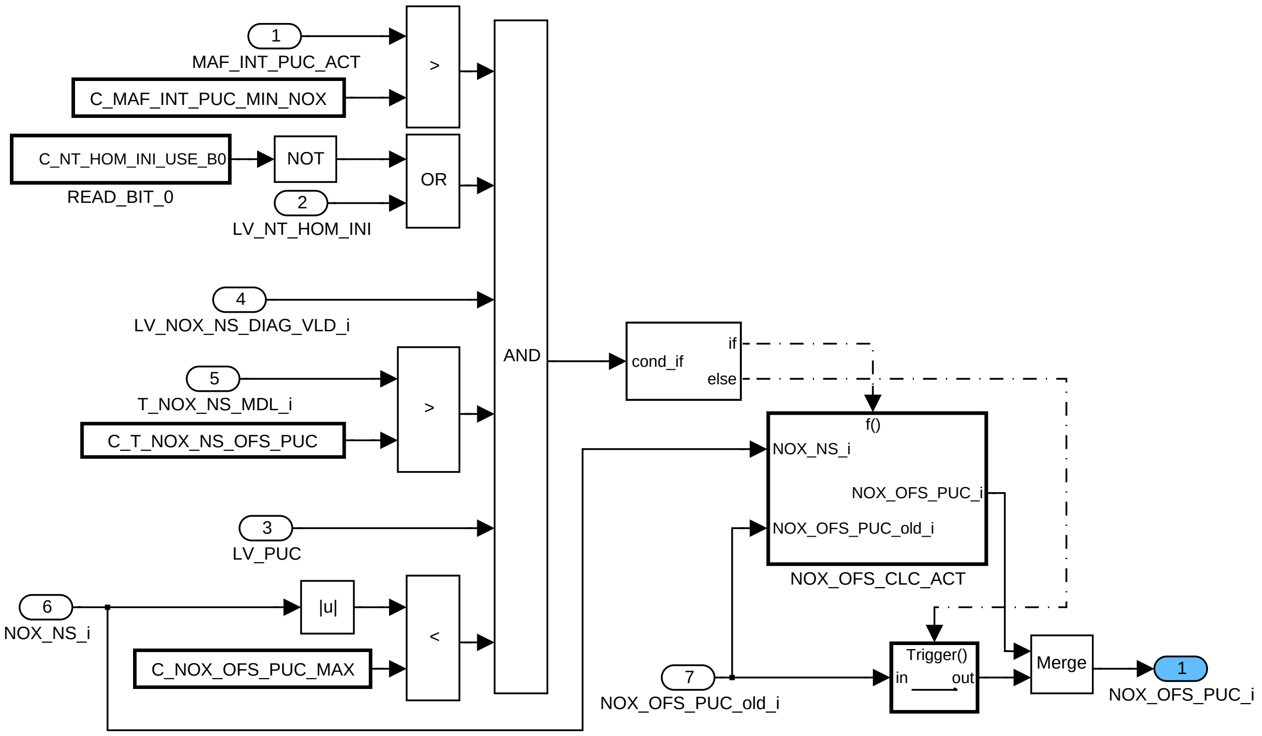


Figure 14 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC1_NOX_OFS_PUC

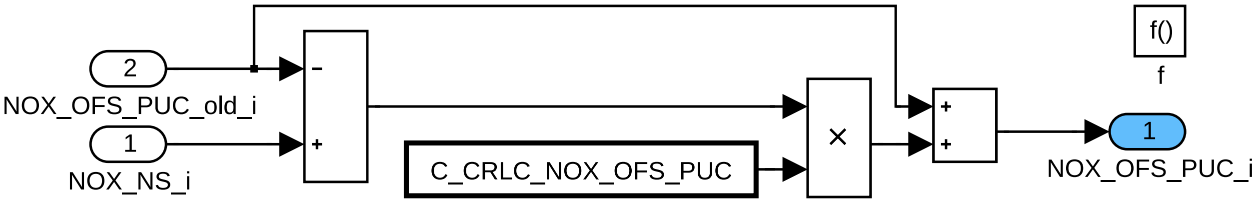


Figure 15 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC1_NOX_OFS_PUC/ NOX_OFS_CLC_ACT

SECTION 1.2: Offset calculation at LOAD phase

This section contains:

SECTION 1.2.1: Activation of NOx minimum observation

SECTION 1.2.2: Calculation of NOx sensor signal minimum value

SECTION 1.2.3: Calculation of offset value after left of LOAD phase

SECTION 1.2.4: Reset of activation flag for NOx signal minimum observation

SECTION 1.2.5: Calculation of old values

SECTION 1.2.1: Activation of NOx minimum observation

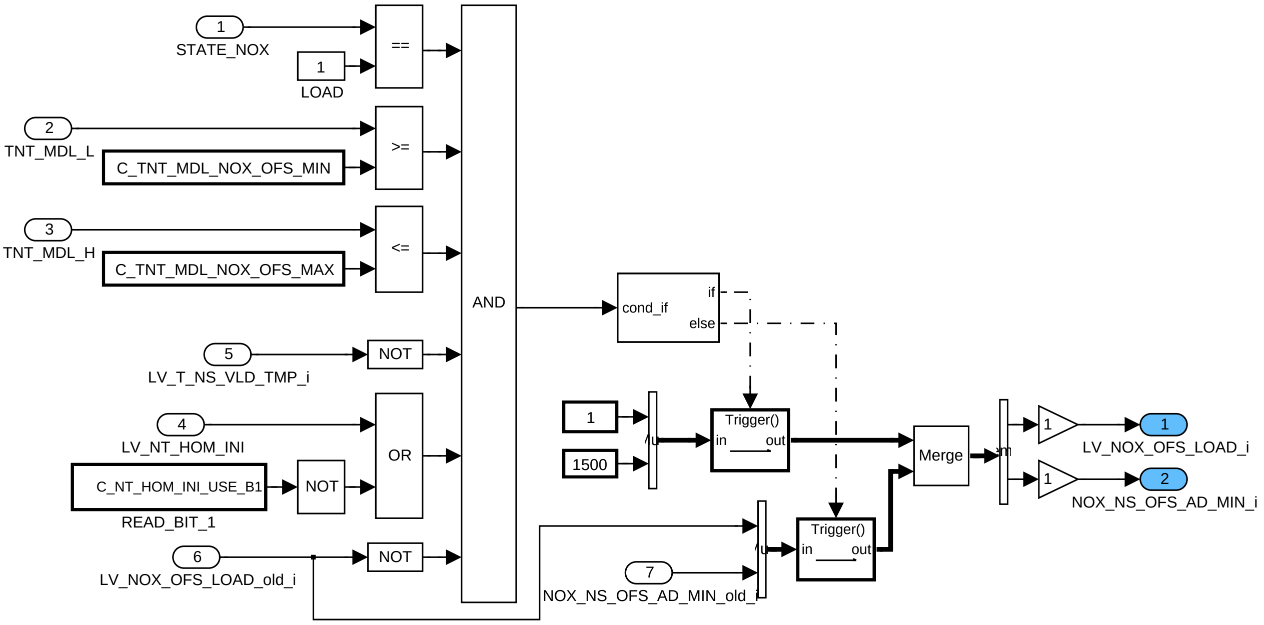


Figure 16 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC2_NOX_OFS_LOAD/ CLC1_DIAG_MIN_OBS_ACT

SECTION 1.2.2: Calculation of NOx sensor signal minimum value

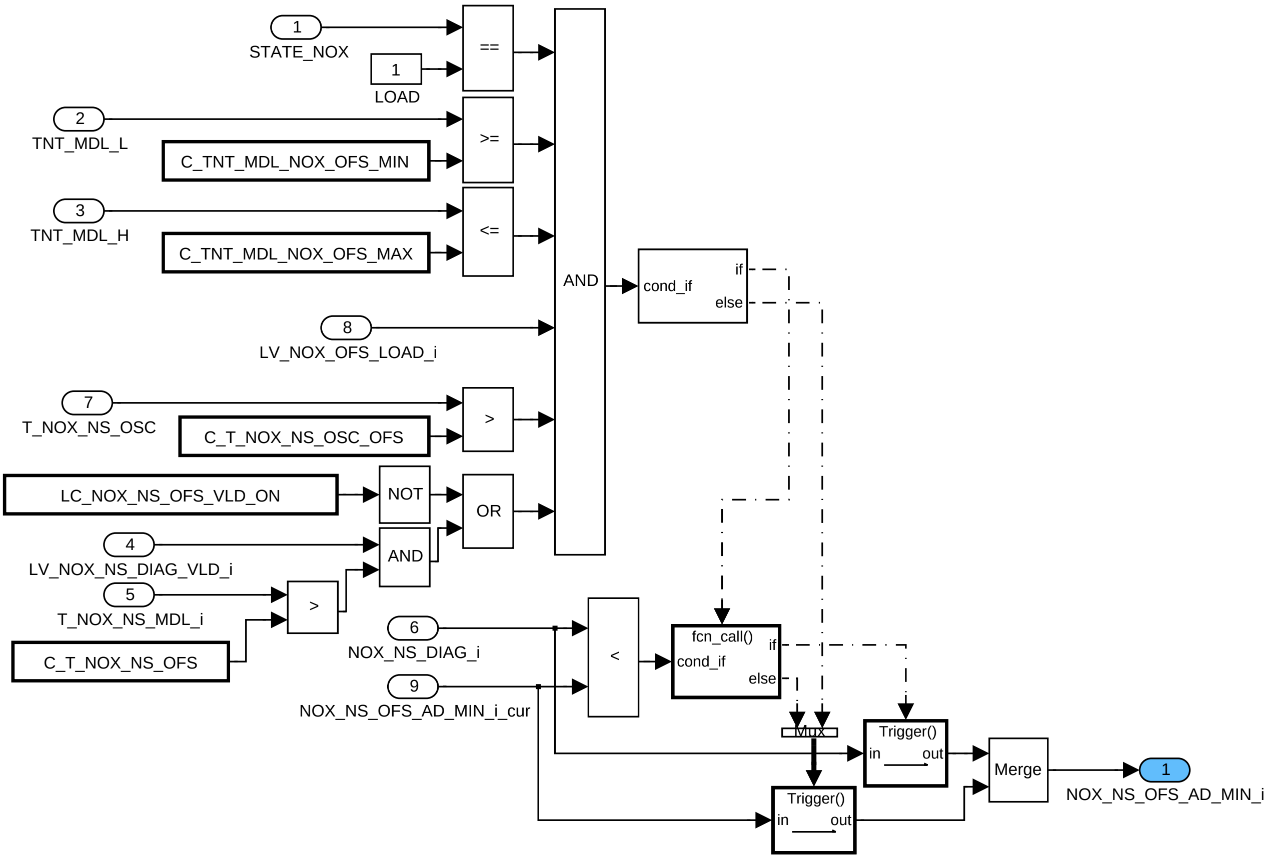


Figure 17 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC2_NOX_OFS_LOAD/ CLC2_NS_DIAG_MIN

SECTION 1.2.3: Calculation of offset value after left of LOAD phase

Calculation of activation condition for calculation

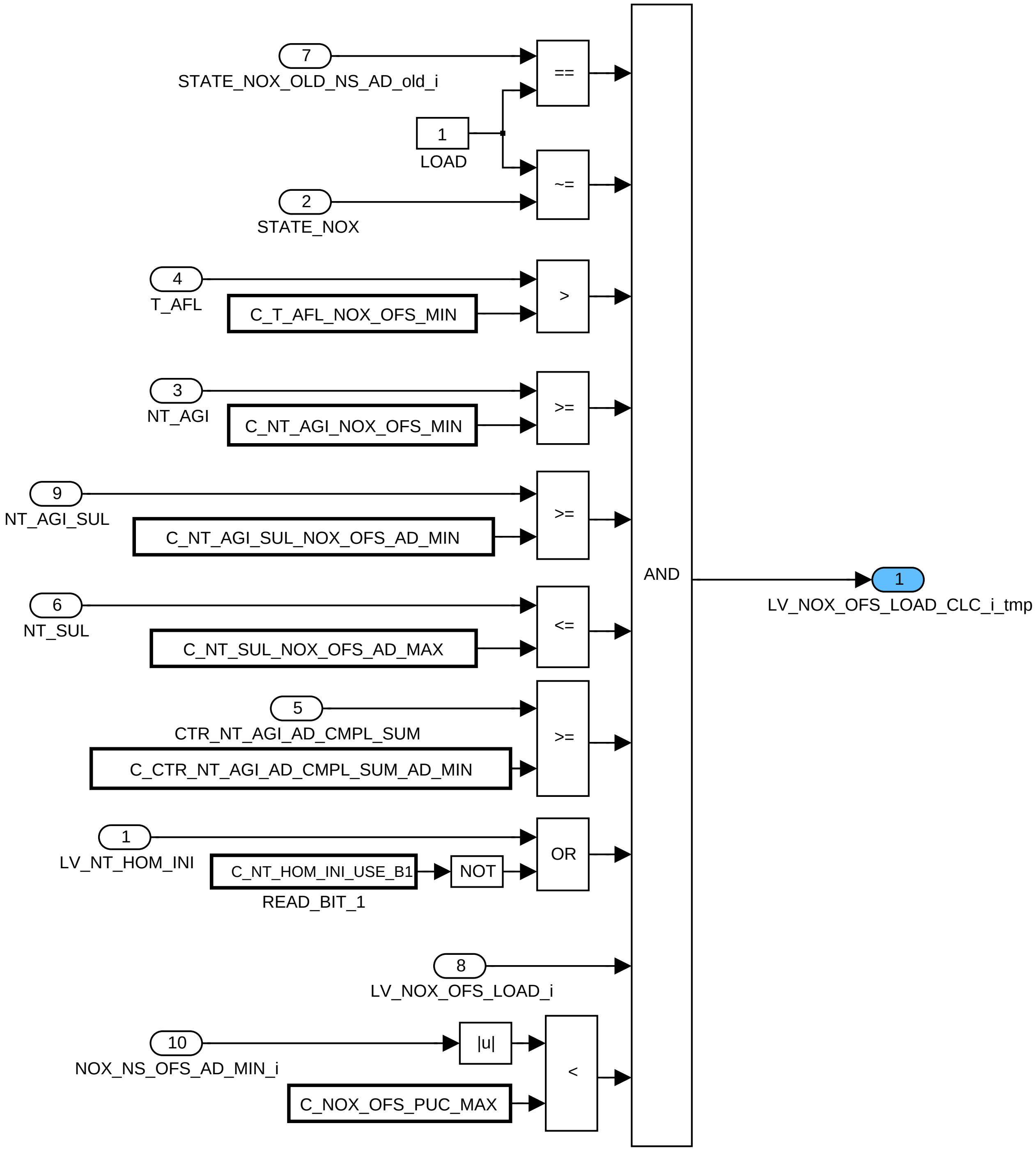


Figure 18 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC2_NOX_OFS_LOAD/ CLC3_NOX_OFS_LOAD/ CLC1_LV_ACT

Check of activation condition

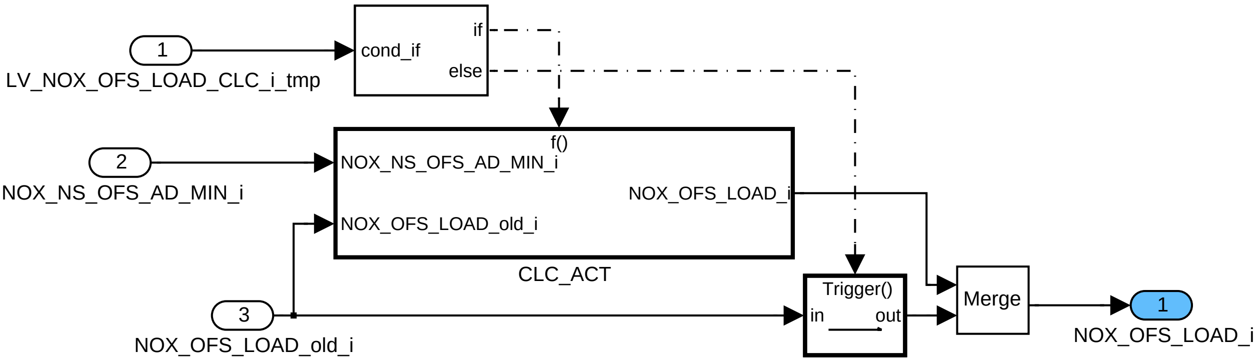


Figure 19 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC2_NOX_OFS_LOAD/ CLC3_NOX_OFS_LOAD/ CLC2_NOX_OFS_LOAD

Calculation if the activation condition is fulfilled

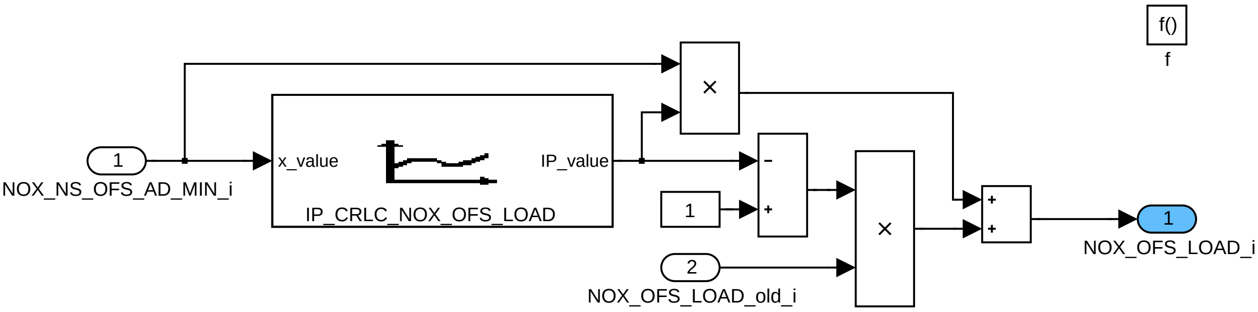


Figure 20 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC2_NOX_OFS_LOAD/ CLC3_NOX_OFS_LOAD/ CLC2_NOX_OFS_LOAD/ CLC_ACT

SECTION 1.2.4: Reset of activation flag for NOx signal minimum observation

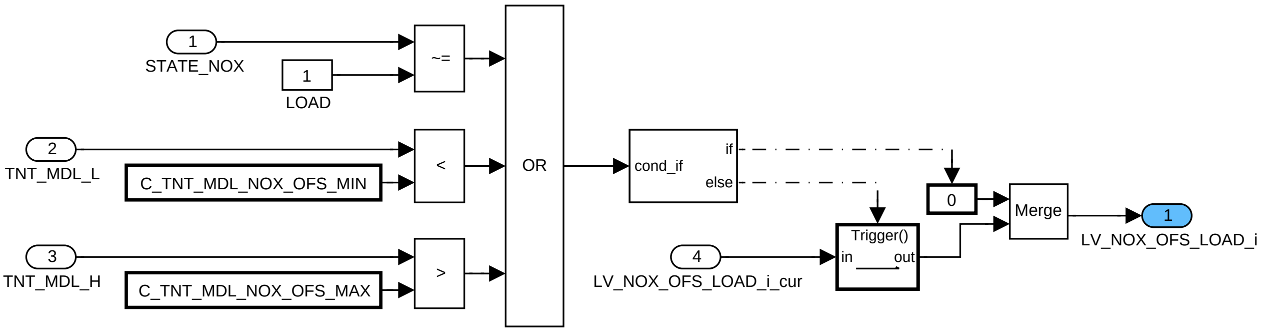


Figure 21 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC2_NOX_OFS_LOAD/ CLC4_LV_NOX_OFS_LOAD

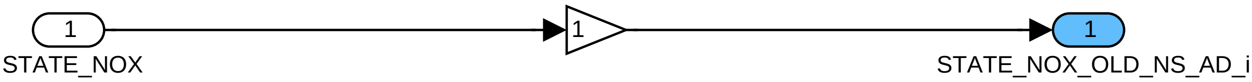


Figure 22 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_OFS/ CLC2_NOX_OFS_LOAD/ CLC5_STATE_NOX_OLD_NS_AD

SECTION 2: Case selection for finally adaptation value calculation

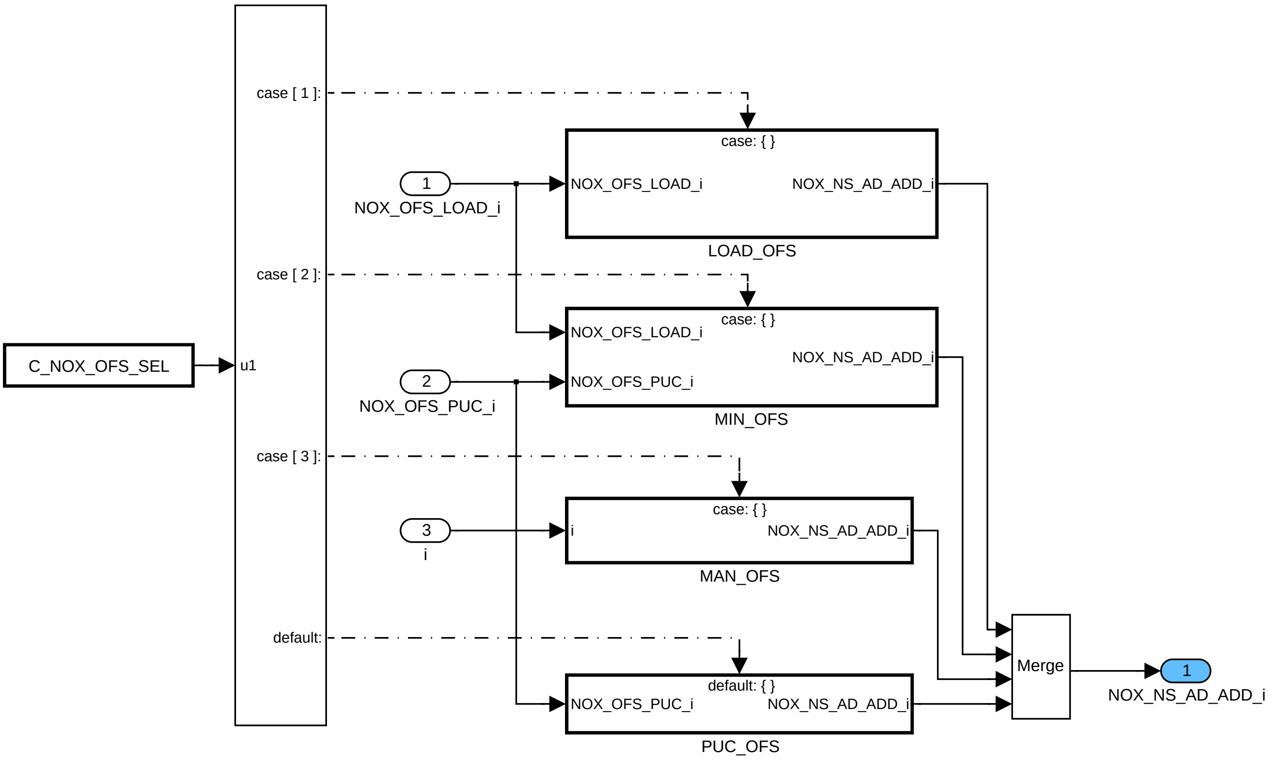


Figure 23 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC2_ADD_VALUE/ CLC1_NOX_AD_ADD

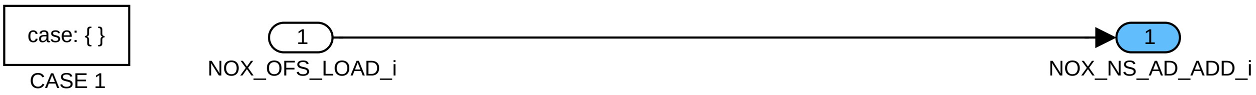


Figure 24 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC2_ADD_VALUE/ CLC1_NOX_AD_ADD/ LOAD_OFS

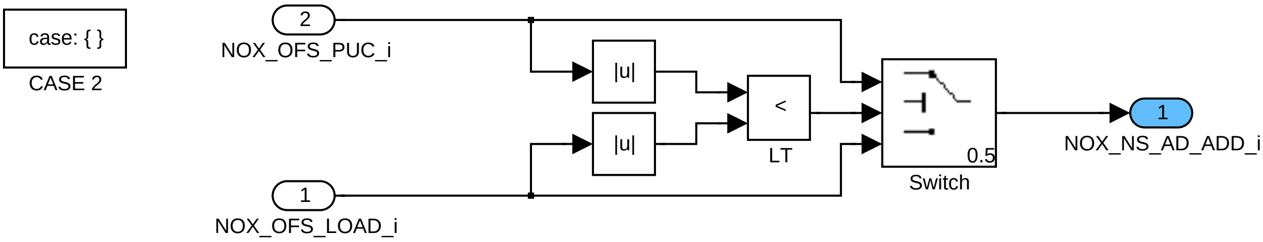


Figure 25 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC2_ADD_VALUE/ CLC1_NOX_AD_ADD/ MIN_OFS

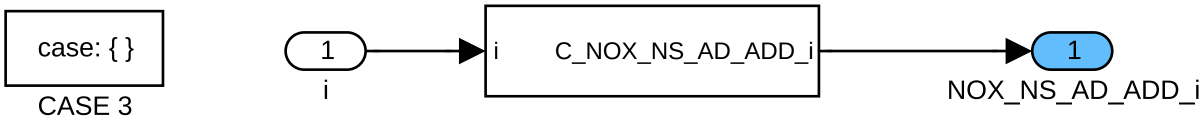


Figure 26 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC2_ADD_VALUE/ CLC1_NOX_AD_ADD/ MAN_OFS

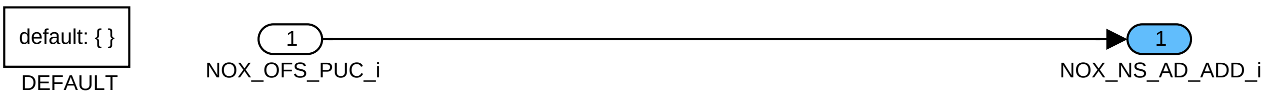


Figure 27 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC2_ADD_VALUE/ CLC1_NOX_AD_ADD/ PUC_OFS

SECTION 3: Limitation of chosen adaptation value

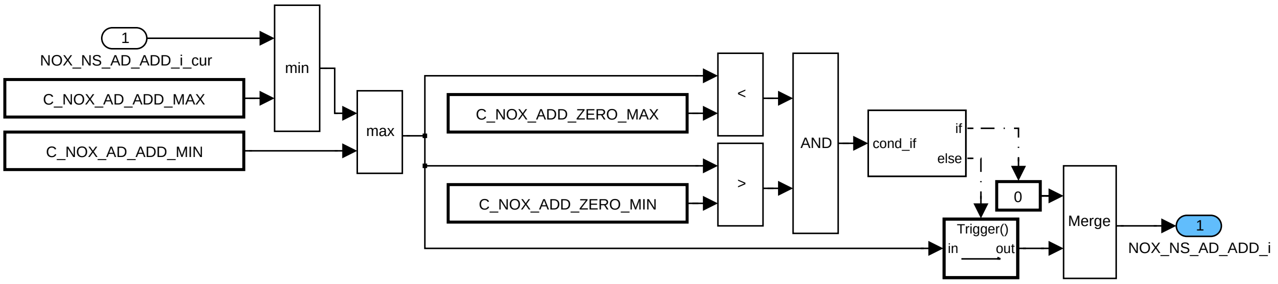


Figure 28 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC3_ADD_VALUE_LIM/ NOX_ADAPT_CLC

SECTION 4: Realization of adaptation on measured signal

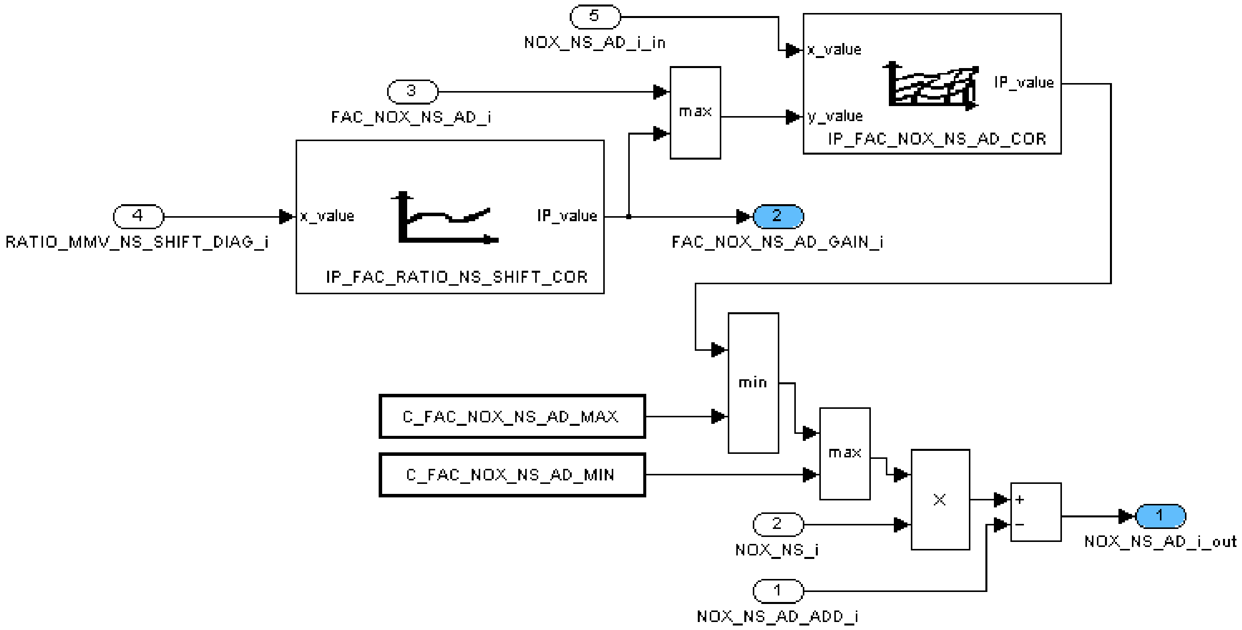


Figure 29 NOXD_MDLADNSFCN0/ SIG_OFS_AD/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC4_FINAL_ADD/ CLC1_NOX_NS_AD

52.7.2 NOx sensor valid flag after regeneration

After regeneration it is prohibited to generate a new regeneration request by the NOx sensor (LV_RGN_REQ_NOX_SENS_i = 1) if the signal is not valid. This can be triggered by observation of the signal gradient or the time C_T_NOX_SENS_VLD. The gradient has to be permanently within a certain range for a continuous time. The time C_T_NOX_SENS_VLD is controlled by the timer T_AFL. T_AFL has to exceed C_T_NOX_SENS_VLD before regeneration can be requested by the NOx sensor.

Application Condition

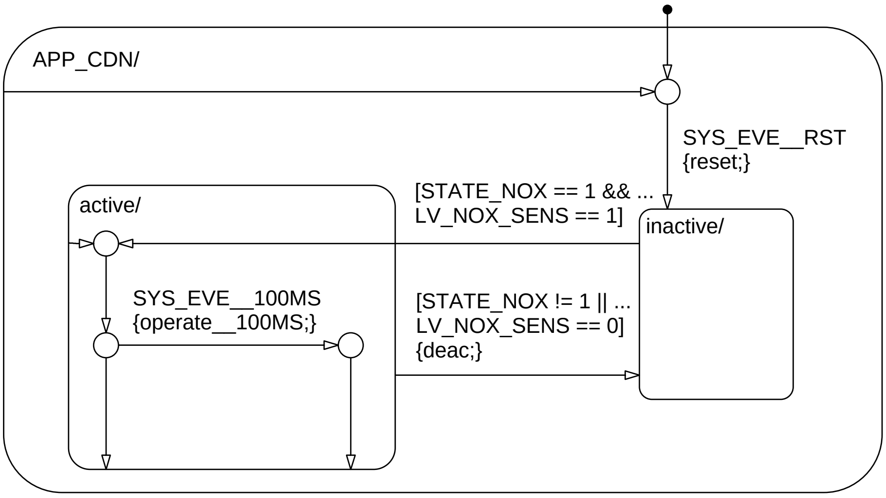


Figure 30 NOXD_MDLADNSFCN0/ VLD_LV/ APP_CDN/ Chart

Function Description

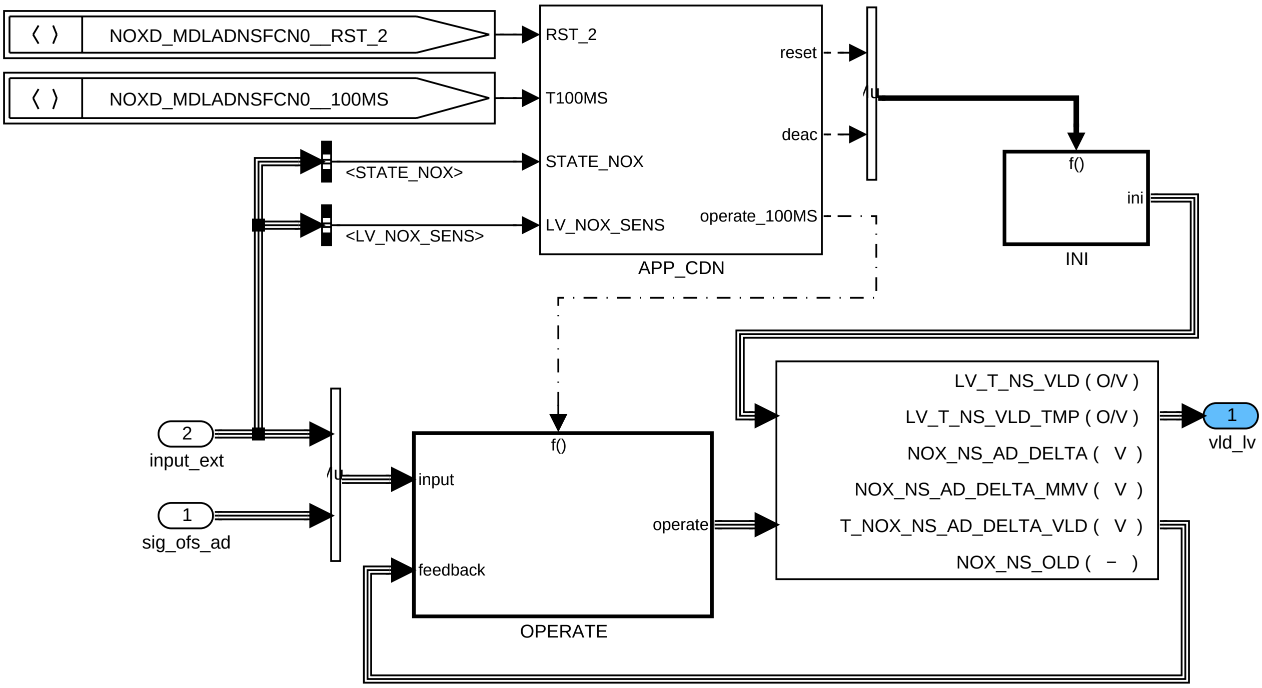


Figure 31 NOXD_MDLADNSFCN0/ VLD_LV

52.7.2.1 INITIALIZATION

At reset all variables are initialized with "0"

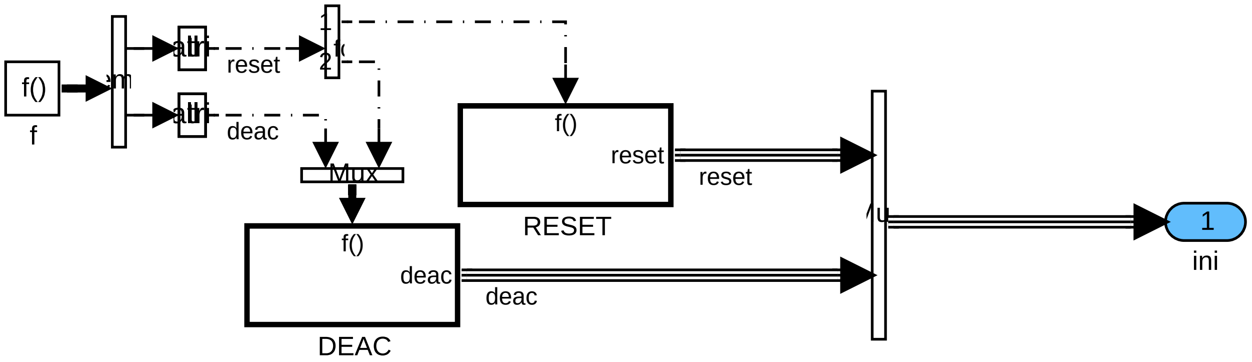


Figure 32 NOXD_MDLADNSFCN0/ VLD_LV/ INI

Initialization at deactivation

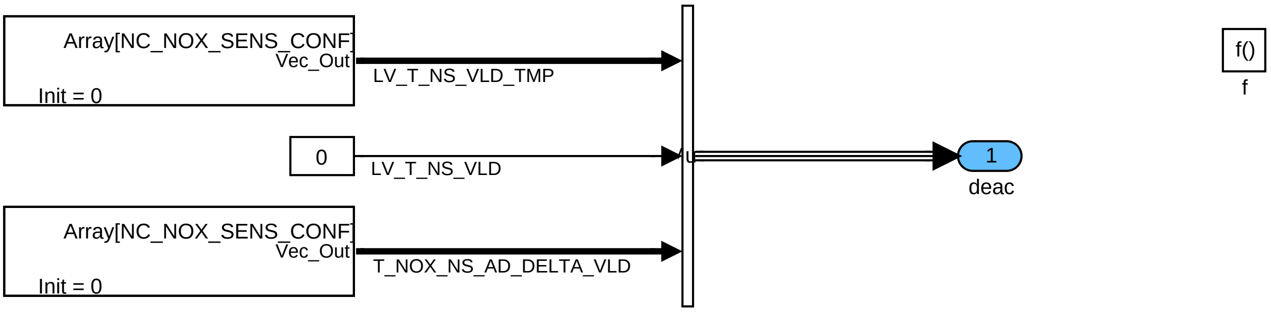


Figure 33 NOXD_MDLADNSFCN0/ VLD_LV/ INI/ DEAC

52.7.2.2 FORMULA SECTION

Formula section contains two main sections:

SECTION 1: NOx sensor specific functionality

SECTION 2: NOx sensor not specific functionality

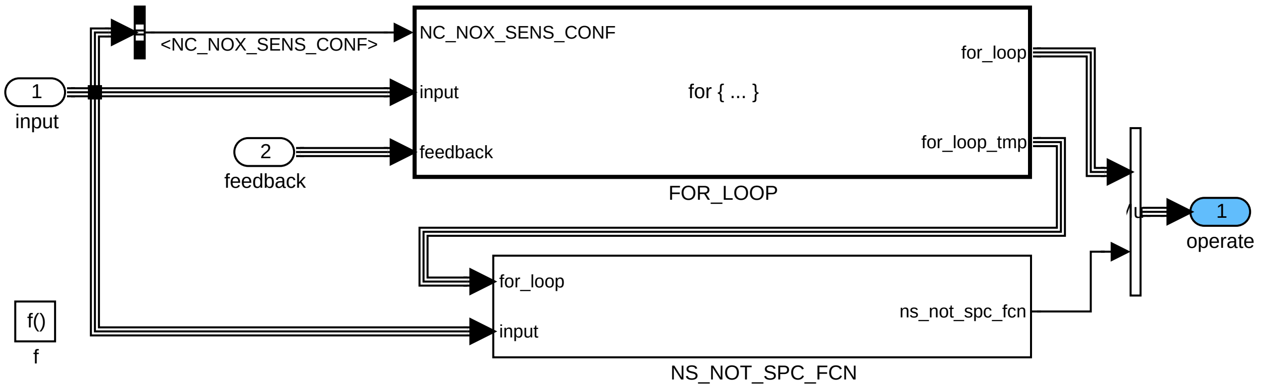


Figure 34 NOXD_MDLADNSFCN0/ VLD_LV/ OPERATE

SECTION 1: NOx sensor specific functionality

Calculation of NOx sensor signal gradient

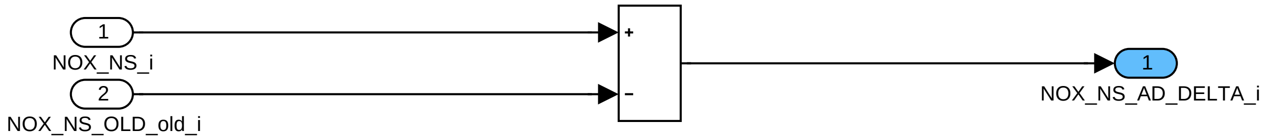


Figure 35 NOXD_MDLADNSFCN0/ VLD_LV/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC1_DELTA

Filtering of NOx sensor signal gradient

The NOx signal is going down after regeneration in normal case. The gradient observation process can set the NOx Sensor signal validation flag earlier.

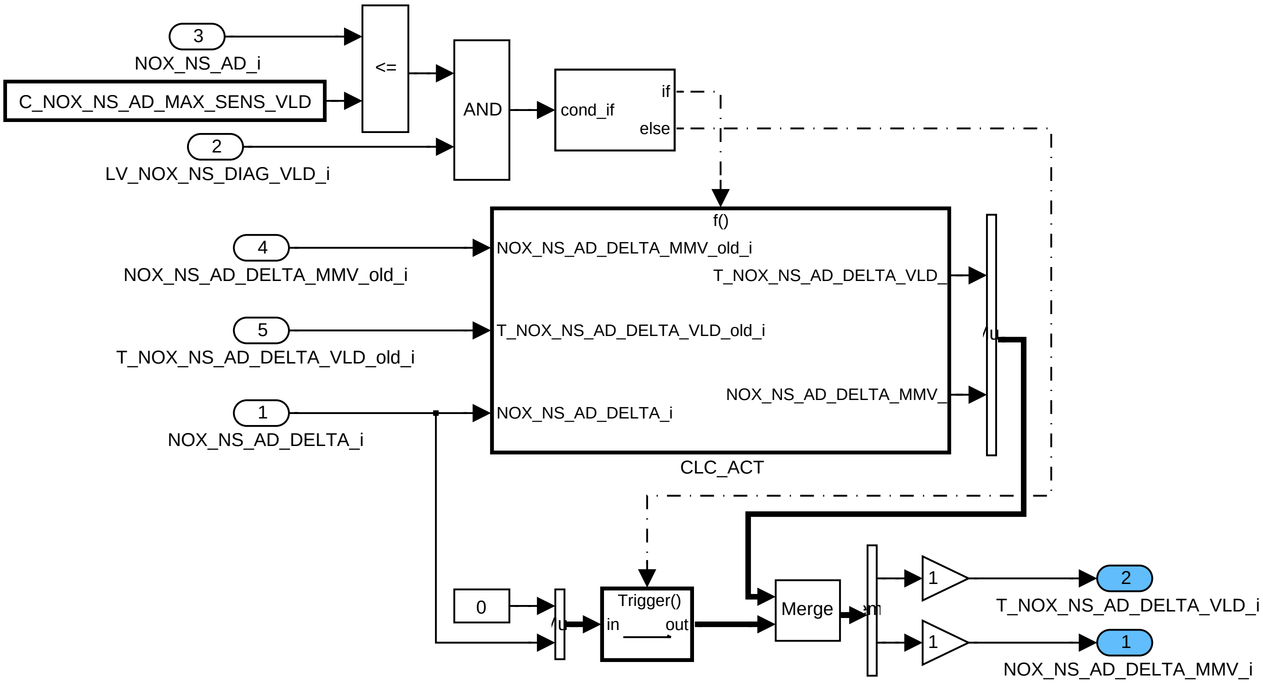


Figure 36 NOXD_MDLADNSFCN0/ VLD_LV/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC2_DELTA_MMV

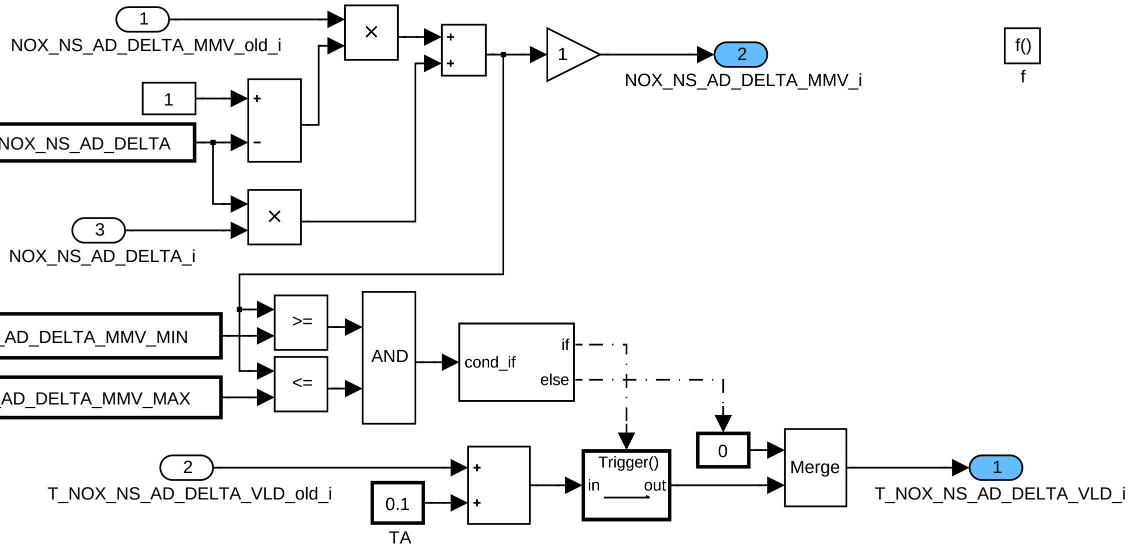


Figure 37 NOXD_MDLADNSFCN0/ VLD_LV/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC2_DELTA_MMV/ CLC_ACT

Calculation of NOx sensor specific validation bit of NOx sensor signal

Set of NOx Sensor validation flag is done using gradient observation, check of minimal time after PUC phase or minimal LOAD time.

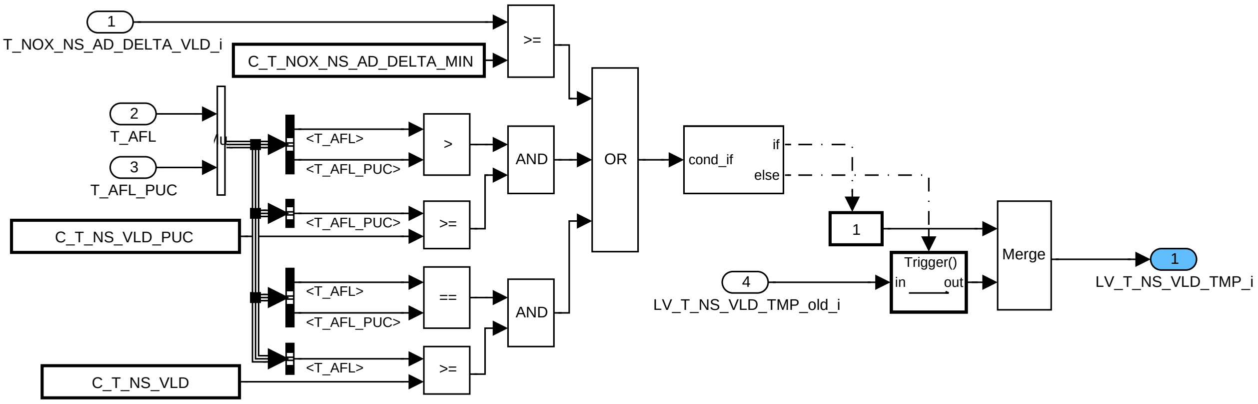


Figure 38 NOXD_MDLADNSFCN0/ VLD_LV/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC3_NS_VLD_TMP

Calculation of old values

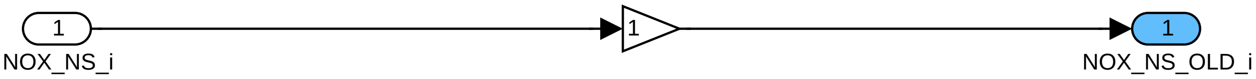


Figure 39 NOXD_MDLADNSFCN0/ VLD_LV/ OPERATE/ FOR_LOOP/ NS_SPC_FCN/ CLC4_NOX_NS_AD_OLD

SECTION 2: NOx sensor not specific functionality

Calculation of validation bit for all NOx sensors

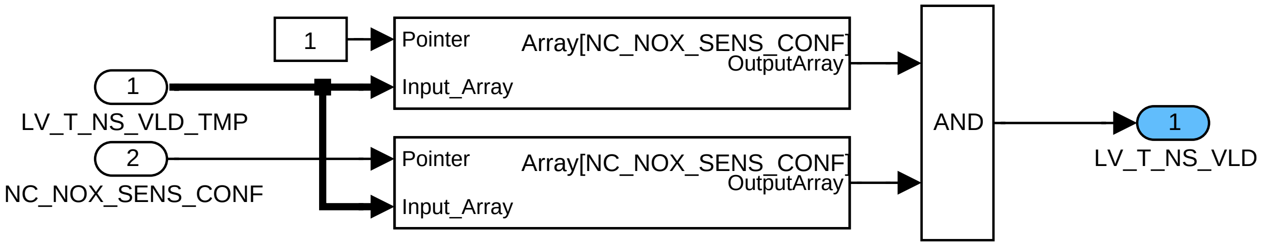


Figure 40 NOXD_MDLADNSFCN0/ VLD_LV/ OPERATE/ NS_NOT_SPC_FCN/ CLC1_SENS_VLD

52.8 NOx sensor OBDI diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CDN_DIAG_NS_OBD_1[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	[-]
Diagnostic condition to start symptom detection at NOx sensor OBD I diagnosis					
LV_ERR_NS_OBD_1[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I failure is present after filtering					
ERR_SYM_NS_OBD_1_HTP[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of NOx sensor OBD I - heater diagnosis					
LV_ERR_NS_OBD_1_HTP[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I - heater failure is present after filtering					
LV_END_DIAG_NS_OBD_1_HTP[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I - heater diagnosis is completed at least one time					
ERR_SYM_NS_OBD_1_NOX[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of NOx sensor OBD I - NOx circuit diagnosis					
LV_ERR_NS_OBD_1_NOX[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I - NOx circuit failure is present after filtering					
LV_END_DIAG_NS_OBD_1_NOX[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I – NOx circuit diagnosis is completed at least one time					
ERR_SYM_NS_OBD_1_LAMB[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of NOx sensor OBD I - linear Lambda circuit diagnosis					
LV_ERR_NS_OBD_1_LAMB[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I - linear Lambda circuit failure is present after filtering					
LV_END_DIAG_NS_OBD_1_LAMB[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I - linear Lambda circuit diagnosis is completed at least one time					
ERR_SYM_NS_OBD_1_VLS[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Symptoms of NOx sensor OBD I - binary Lambda circuit diagnosis					
LV_ERR_NS_OBD_1_VLS[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I - binary Lambda circuit failure is present after filtering					
LV_END_DIAG_NS_OBD_1_VLS[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
NOx sensor OBD I - binary Lambda circuit diagnosis is completed at least one time					

Input data:

NC_NOX_SENS_CONF	LV_INH_DIAG_NS_OBD_1[NC_NOX_SENS_CONF]	LV_ERR_NS_CAN_BOFF	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]
LV_ST_END	ERR_NS[NC_NOX_SENS_CONF]		

Import actions:

ACTION_ERRM_FilterSymptom(IN<IDX_DIAG>, IN< LV_CDN_DIAG_XX >, IN< ERR_SYM_XX >, IN< C_ABC_INC_XX >, IN< C_ABC_DEC_XX >, IN< C_ABC_MAX_XX >, OUT< LV_ERR_XX >)

FUNCTION DESCRIPTION:

General information:

The NOx sensor has an own Electronic Control Unit (ECU), which monitors the cable connection between sensor element and sensor ECU on electrical faults like open circuits or short circuits. The actual monitoring result/ the actual diagnosis result is cyclic transmitted to the EMS by the CAN messages, which include the sensor measurement signals.

The purpose of this diagnosis is to evaluate the cyclic received sensor ECU diagnosis result. If the sensor ECU reports an electrical fault (OBDI error between sensor element and sensor ECU) then this error is debounced and stored within error memory by use of standard error memory algorithm.

At ERR_SYM_NS_OBD_1_HTP, ERR_SYM_NS_OBD_1_NOX, ERR_SYM_NS_OBD_1_LAMB and ERR_SYM_NS_OBD_1_VLS combinations of error symptoms are possible (0...0x0F).

The following table describes the meaning of NOx sensor status byte:

Remark: The meaning of NOx sensor error byte was copied from NOx sensor specification of sensor manufacturer. If the sensor manufacturer makes a change at sensor error byte meaning, then this specification will not be updated!

NOx sensor error byte: ERR_NS[i]								
B7	B6	B5	B4	B3	B2	B1	B0	Function
0	0	0	0	0	0	0	0	no failure detected
							1	open wire at heater circuit
						1		open wire at NOx circuit
					1			open wire at linear Lambda circuit
				1				open wire at binary Lambda circuit
			1					short circuit at heater circuit
		1						short circuit at NOx circuit
	1							short circuit at linear Lambda circuit
1								short circuit at binary Lambda circuit

Specific diagnostic information to the project are defined in chapters ‘Diagnostic general information‘ or ‘application incidence‘ (EOL, DTC, error code number, symptom number...).

Description:

Error-symptoms of heater diagnosis are defined as following:

```
└─┴─┴─┴─┘
 │  │  │  └─> (not used)                                  (= SYM_0)
 │  │  └────> (not used)                                  (= SYM_1)
 │  └───────> open circuit at heater                      (= SYM_2)
 └──────────> short circuit at heater                     (= SYM_3)
```

Error-symptoms of NOx circuit diagnosis are defined as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> (not used)                                  (= SYM_0)
 │  │  └────> (not used)                                  (= SYM_1)
 │  └───────> open circuit at NOx circuit                 (= SYM_2)
 └──────────> short circuit at NOx circuit                (= SYM_3)
```

Error-symptoms of linear Lambda circuit diagnosis are defined as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> (not used)                                  (= SYM_0)
 │  │  └────> (not used)                                  (= SYM_1)
 │  └───────> open circuit at linear Lambda circuit       (= SYM_2)
 └──────────> short circuit at linear Lambda circuit      (= SYM_3)
```

Error-symptoms of binary Lambda circuit diagnosis are defined as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │  │  │  └─> (not used)                                  (= SYM_0)
 │  │  └────> (not used)                                  (= SYM_1)
 │  └───────> open circuit at binary Lambda circuit       (= SYM_2)
 └──────────> short circuit at binary Lambda circuit      (= SYM_3)
```

Application conditions:

Initialisation:

at reset:

LV_CDN_DIAG_NS_OBD_1[i] = 0
LV_ERR_NS_OBD_1[i] = 0

ERR_SYM_NS_OBD_1_HTP[i] = *Refer to filtering configuration for the initialisation value*
LV_ERR_NS_OBD_1_HTP[i] = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_NS_OBD_1_HTP[i] = *Refer to filtering configuration for the initialisation value*

ERR_SYM_NS_OBD_1_NOX[i] = *Refer to filtering configuration for the initialisation value*
LV_ERR_NS_OBD_1_NOX[i] = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_NS_OBD_1_NOX[i] = *Refer to filtering configuration for the initialisation value*

ERR_SYM_NS_OBD_1_LAMB[i] = *Refer to filtering configuration for the initialisation value*
LV_ERR_NS_OBD_1_LAMB[i] = *Refer to filtering configuration for the initialisation value*
LV_END_DIAG_NS_OBD_1_LAMB[i] = *Refer to filtering configuration for the initialisation value*

ERR_SYM_NS_OBD_1_VLS[i] = *Refer to filtering configuration for the initialisation value*
LV_ERR_NS_OBD_1_VLS[i] = *Refer to filtering configuration for the initialisation value*

LV_END_DIAG_NS_OBD_1_VLS[i] = *Refer to filtering configuration for the initialisation value*

at activation:

LV_CDN_DIAG_NS_OBD_1[i] = 1

at deactivation:

LV_CDN_DIAG_NS_OBD_1[i] = 0

Recurrence: 10ms

Activation: LV_ST_END = 1 **AND**
LV_ERR_NS_CAN_BOFF = 0 **AND**
LV_ERR_NS_CAN_MSG_LOST[i] = 0 **AND**
LV_INH_DIAG_NS_OBD_1[i] = 0

Deactivation: if any activation condition is not fulfilled

Formula section:

Heater circuit failures:

```
if    Bit B0 of ERR_NS[i] = 1
then  ERR_SYM_NS_OBD_1_HTP[i] = set bit SYM_2 of ERR_SYM_NS_OBD_1_HTP[i]
else  ERR_SYM_NS_OBD_1_HTP[i] = reset bit SYM_2 of ERR_SYM_NS_OBD_1_HTP[i]
endif
if    Bit B4 of ERR_NS[i] = 1
then  ERR_SYM_NS_OBD_1_HTP[i] = set bit SYM_3 of ERR_SYM_NS_OBD_1_HTP[i]
else  ERR_SYM_NS_OBD_1_HTP[i] = reset bit SYM_3 of ERR_SYM_NS_OBD_1_HTP[i]
endif
```

For failure and error management treatment the anti-bounce mechanism is called:

```
ACTION_ERRM_FilterSymptom(                              IN<NS_OBD_1_HTP[i]>,
        IN<LV_CDN_DIAG_NS_OBD_1[i]>,           IN<ERR_SYM_NS_OBD_1_HTP[i]>,
        IN<C_ABC_INC_NS_OBD_1_HTP>,                                   IN<1>,
        IN<C_ABC_MAX_NS_OBD_1_HTP>, OUT<LV_ERR_NS_OBD_1_HTP[i]>)
```

NOx circuit failures:

```
if    Bit B1 of ERR_NS[i] = 1
then  ERR_SYM_NS_OBD_1_NOX[i] = set bit SYM_2 of ERR_SYM_NS_OBD_1_NOX[i]
else  ERR_SYM_NS_OBD_1_NOX[i] = reset bit SYM_2 of ERR_SYM_NS_OBD_1_NOX[i]
endif
if    Bit B5 of ERR_NS[i] = 1
then  ERR_SYM_NS_OBD_1_NOX[i] = set bit SYM_3 of ERR_SYM_NS_OBD_1_NOX[i]
```

```
else    ERR_SYM_NS_OBD_1_NOX[i] = reset bit SYM_3 of ERR_SYM_NS_OBD_1_NOX[i]
endif
```

For failure and error management treatment the anti-bounce mechanism is called:

```
ACTION_ERRM_FilterSymptom(                              IN<NS_OBD_1_NOX[i]>,
        IN<LV_CDN_DIAG_NS_OBD_1[i]>,         IN<ERR_SYM_NS_OBD_1_NOX[i]>,
        IN<C_ABC_INC_NS_OBD_1_NOX>,                                   IN<1>,
        IN<C_ABC_MAX_NS_OBD_1_NOX>, OUT<LV_ERR_NS_OBD_1_NOX[i]>)
```

Linear Lambda circuit failures:

```
if      Bit B2 of ERR_NS[i] = 1
then    ERR_SYM_NS_OBD_1_LAMB[i] = set bit SYM_2 of ERR_SYM_NS_OBD_1_LAMB[i]
else    ERR_SYM_NS_OBD_1_LAMB[i] = reset bit SYM_2 of ERR_SYM_NS_OBD_1_LAMB[i]
endif
if      Bit B6 of ERR_NS[i] = 1
then    ERR_SYM_NS_OBD_1_LAMB[i] = set bit SYM_3 of ERR_SYM_NS_OBD_1_LAMB[i]
else    ERR_SYM_NS_OBD_1_LAMB[i] = reset bit SYM_3 of ERR_SYM_NS_OBD_1_LAMB[i]
endif
```

For failure and error management treatment the anti-bounce mechanism is called:

```
ACTION_ERRM_FilterSymptom(                              IN<NS_OBD_1_LAMB[i]>,
        IN<LV_CDN_DIAG_NS_OBD_1[i]>,         IN<ERR_SYM_NS_OBD_1_LAMB[i]>,
        IN<C_ABC_INC_NS_OBD_1_LAMB>,                                  IN<1>,
        IN<C_ABC_MAX_NS_OBD_1_LAMB>, OUT<LV_ERR_NS_OBD_1_LAMB[i]>)
```

Binary Lambda circuit failures:

```
if      Bit B3 of ERR_NS[i] = 1
then    ERR_SYM_NS_OBD_1_VLS[i] = set bit SYM_2 of ERR_SYM_NS_OBD_1_VLS[i]
else    ERR_SYM_NS_OBD_1_VLS[i] = reset bit SYM_2 of ERR_SYM_NS_OBD_1_VLS[i]
endif
if      Bit B7 of ERR_NS[i] = 1
then    ERR_SYM_NS_OBD_1_VLS[i] = set bit SYM_3 of ERR_SYM_NS_OBD_1_VLS[i]
else    ERR_SYM_NS_OBD_1_VLS[i] = reset bit SYM_3 of ERR_SYM_NS_OBD_1_VLS[i]
endif
```

For failure and error management treatment the anti-bounce mechanism is called:

```
ACTION_ERRM_FilterSymptom(                              IN<NS_OBD_1_VLS[i]>,
        IN<LV_CDN_DIAG_NS_OBD_1[i]>,         IN<ERR_SYM_NS_OBD_1_VLS[i]>,
        IN<C_ABC_INC_NS_OBD_1_VLS>,                                   IN<1>,
        IN<C_ABC_MAX_NS_OBD_1_VLS>, OUT<LV_ERR_NS_OBD_1_VLS[i]>)
```

Summarise of failures:

```
if    LV_ERR_NS_OBD_1_HTP[i] = 1             OR
      LV_ERR_NS_OBD_1_NOX[i] = 1             OR
      LV_ERR_NS_OBD_1_LAMB[i] = 1            OR
      LV_ERR_NS_OBD_1_VLS[i] = 1
then  LV_ERR_NS_OBD_1[i] = 1
else  LV_ERR_NS_OBD_1[i] = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_NS_OBD_1_HTP	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment of NOx sensor OBD I - heater diagnosis					
C_ABC_MAX_NS_OBD_1_HTP	1	0...FFH	0...255	1	[-]
Maximum value of anti-bounce counter for NOx sensor OBD I - heater diagnosis					
C_ABC_INC_NS_OBD_1_NOX	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment of NOx sensor OBD I - NOx circuit diagnosis					
C_ABC_MAX_NS_OBD_1_NOX	1	0...FFH	0...255	1	[-]
Maximum value of anti-bounce counter for NOx sensor OBD I - NOx circuit diagnosis					
C_ABC_INC_NS_OBD_1_LAMB	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment of NOx sensor OBD I - linear Lambda circuit diagnosis					
C_ABC_MAX_NS_OBD_1_LAMB	1	0...FFH	0...255	1	[-]
Maximum value of anti-bounce counter for NOx sensor OBD I - linear Lambda circuit diagnosis					
C_ABC_INC_NS_OBD_1_VLS	1	0...FFH	0...255	1	[-]
Anti-bounce counter increment of NOx sensor OBD I - binary Lambda circuit diagnosis					
C_ABC_MAX_NS_OBD_1_VLS	1	0...FFH	0...255	1	[-]
Maximum value of anti-bounce counter for NOx sensor OBD I - binary Lambda circuit diagnosis					

52.9 NOx sensor OBDI diagnosis (Application Incidences)

52.9.1 Inhibition of NOx sensor error byte evaluation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_NS_OBD_1[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Inhibition of NOx sensor OBD I diagnosis					

Input data:

NC_NOX_SENS_CONF	LV_ST_END	LV_CDN_VB_CAN_DIAG	LV_VAR_NOX

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_DIAG_NS_OBD_1[i] allows to inhibit the NOx sensor OBDI diagnosis, which evaluates the NOx sensor error byte.

Application conditions:

Initialisation: at reset, at deactivation:

LV_INH_DIAG_NS_OBD_1[i] = 1

Recurrence: 100ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

```
if    (NOx message has not been received via LOCAN at least for once - LV_VAR_NOX)
      LV_VAR_NOX = 0                                    OR
      ~~LV_ERR_NS_CAN_BOFF = 1                            OR~~
      LV_CDN_VB_CAN_DIAG = 0
then  LV_INH_DIAG_NS_OBD_1[i] = 1
else  LV_INH_DIAG_NS_OBD_1[i] = 0
endif
```

Configuration for diagnostic symptoms:

Diagnostic **NS_OBD_1_ HTP[i]**	**Symptom description**	**Symptom**	**Filter type**
NOx sensor OBDI - heater diagnosis	(not used)	SYM_0	STD_INI
	(not used)	SYM_1	
	open circuit at heater	SYM_2	
	short circuit at heater	SYM_3	

Diagnostic **NS_OBD_1_ HTP[i]**	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
NOx sensor OBDI - heater diagnosis	(not used)	*SYM_0*						
	(not used)	*SYM_1*						
	open circuit at heater	*SYM_2*						
	short circuit at heater	*SYM_3*						

Diagnostic **NS_OBD_1_ NOX[i]**	**Symptom description**	**Symptom**	**Filter type**
NOx sensor OBDI - NOx circuit diagnosis	(not used)	SYM_0	STD_INI
	(not used)	SYM_1	
	open circuit at NOx circuit	SYM_2	
	short circuit at NOx circuit	SYM_3	

Diagnostic **NS_OBD_1_ NOX[i]**	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
NOx sensor OBDI - NOx circuit diagnosis	(not used)	*SYM_0*						
	(not used)	*SYM_1*						
	open circuit at NOx circuit	*SYM_2*						
	short circuit at NOx circuit	*SYM_3*						

Diagnostic **NS_OBD_1_ LAMB[i]**	**Symptom description**	**Symptom**	**Filter type**
NOx sensor OBDI - linear Lambda circuit diagnosis	(not used)	SYM_0	STD_INI
	(not used)	SYM_1	
	open circuit at linear Lambda circuit	SYM_2	
	short circuit at linear Lambda circuit	SYM_3	

Diagnostic **NS_OBD_1_ LAMB[i]**	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
NOx sensor OBDI - linear Lambda circuit diagnosis	(not used)	*SYM_0*						
	(not used)	*SYM_1*						
	open circuit at linear Lambda circuit	*SYM_2*						
	short circuit at linear Lambda circuit	*SYM_3*						

Diagnostic NS_OBD_1_VLS[i]	Symptom description	Symptom	Filter type
NOx sensor OBDI - binary Lambda circuit diagnosis	(not used)	SYM_0	STD_INI
	(not used)	SYM_1	
	open circuit at binary Lambda circuit	SYM_2	
	short circuit at binary Lambda circuit	SYM_3	

Diagnostic NS_OBD_1_VLS[i]	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx sensor OBDI - binary Lambda circuit diagnosis	(not used)	*SYM_0*						
	(not used)	*SYM_1*						
	open circuit at binary Lambda circuit	*SYM_2*						
	short circuit at binary Lambda circuit	*SYM_3*						

52.10 NOx sensor OBD II diagnosis (Application Incidences)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_NS_OBD_2_HTP[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Inhibition of NOx sensor heater performance diagnosis					
LV_INH_DIAG_NS_OBD_2_AVL[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Inhibition of NOx sensor availability diagnosis					
LV_INH_DIAG_NS_OBD_2_OFS[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Inhibition of NOx sensor offset diagnosis					
LV_INH_DIAG_NS_OBD_2_DYN[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Inhibition of NOx sensor dynamic diagnosis					
LV_INH_DIAG_NS_OBD_2_ACT[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Inhibition of NOx sensor activity diagnosis					
LV_INH_DIAG_NS_OBD_2_PLAUS[NC_NOX_SENS_CONF]	V/O	0...1H	0...1	1	[-]
Inhibition of NOx-Sensor plausibility diagnosis					
LV_INH_DIAG_NS_OBD_2_GAIN[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Inhibition of NOx sensor - NOx signal gain diagnosis					
LV_INH_DIAG_NS_OBD_2_RAW[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Inhibition of NOx sensor - NOx raw emission diagnosis					
LV_INH_DIAG_NS_OBD_2_SHIFT[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Inhibition of NOx sensor internal setpoint shift diagnosis					
LV_INH_DIAG_NS_OBD_2_VERS[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Inhibition of NOx sensor version diagnosis					
LV_ERR_LAMB_TMP	V	0...1H	0...1	1	[-]
Error on any lambda sensor detected					
LV_ERR_FUEL_TMP	V	0...1H	0...1	1	[-]
Error on any fuel component detected					
FAC_PV_GRD_ABSV_INT	V	0...FFFFH	0...65535	1	[-]
Integral of absolute value of pedal gradient for detection of vandalism					
PV_AV_OLD_AVL	-	0...FFH	0...99.60937	0.390625	[%]
Old value of global degree of activation of the accelerator pedal (low resolution)					
LV_PV_GRD_NS_DIAG	V	0...1H	0...1	1	[-]
Pedal gradient disables diagnosis					
T_LAMB_LS_UP_NOT_VLD[NC_CBK_EX_NR]	V	0...FFFFH	0...655.35	0.01	[s]
Time since LAMB_VLS_UP is not valid					
T_WOUT_NS_AD	V	0...FFFFH	0...65535	1	[s]
Time after last NOx sensor gain adaptation					

Input data:

LV_ERR_CTL_LSL_UP[NC_CBK_EX_NR]	LV_ERR_DIAGCPS	LV_ERR_EL_CPS	LV_ERR_FUP
LV_ERR_FUP_ORNG	LV_ERR_FUP_MFP_PLAUS	LV_ERR_FUP_ST	LV_ERR_H_PRS_SYS
LV_ERR_LS_DOWN[NC_CBK_EX_NR]	LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_ERR_MAF	LV_ERR_MAP

LV_ERR_MAP_TPS_PLAUS	LV_ERR_NS_CAN_BOFF	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_1[NC_NOX_SENS_CONF]
LV_ERR_OFS_LSL_UP[NC_CBK_EX_NR]	LV_ERR_TTIP_MES_LSH_UP[NC_CBK_EX_NR]	LV_ERR_VCV	LV_ST_END
NC_CBK_EX_NR	NC_NOX_SENS_CONF	PV_AV	STATE_ERR_NS_OBD_2[NC_NOX_SENS_CONF]
STATE_INH_NS_OBD_2_EXT_ADJ[NC_NOX_SENS_CONF]	LV_ERR_NT_AGI	LV_ERR_CAT_DIAG_AFL[NC_CBK_EX_NR]	LV_ERR_TEG_PCAT_DOWN
LV_ERR_NS_OBD_2[NC_NOX_SENS_CONF]	LV_NS_AD_REQ	LV_SO2P_REQ	LV_LAM_ADJ_CAT_DIAG[NC_CBK_EX_NR]
NTLD	STATE_NS[NC_NOX_SENS_CONF]	LV_LAMB_LS_UP_VLD[NC_CBK_EX_NR]	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_INH_NS_HTP[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor heater diagnosis					
C_STATE_INH_NS_AVL[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor availability diagnosis					
C_STATE_INH_NS_OFS[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor offset diagnosis					
C_STATE_INH_NS_DYN[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor dynamic diagnosis					
C_STATE_INH_NS_ACT[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor activity diagnosis					
C_STATE_INH_NS_PLAUS[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor plausibility diagnosis					
C_STATE_INH_NS_GAIN[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor - NOx signal gain diagnosis					
C_STATE_INH_NS_RAW[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor - NOx raw emission diagnosis					
C_STATE_INH_NS_SHIFT[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor internal setpoint shift diagnosis					
C_STATE_INH_NS_VERS[NC_NOX_SENS_CONF]	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx sensor version diagnosis					
C_ERR_NS_HTP_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit heater diagnosis					
C_ERR_NS_AVL_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit availability diagnosis					
C_ERR_NS_OFS_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit offset diagnosis					
C_ERR_NS_DYN_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit dynamic diagnosis					
C_ERR_NS_ACT_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit activity diagnosis					
C_ERR_NS_PLAUS_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit plausibility diagnosis					

C_ERR_NS_GAIN_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit NOx signal gain diagnosis					
C_ERR_NS_RAW_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit NOx raw emission diagnosis					
C_ERR_NS_SHIFT_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit NOx sensor internal setpoint shift diagnosis					
C_ERR_NS_VERS_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit NOx sensor version diagnosis					
C_FAC_PV_GRD_ABSV_INT_MIN	1	0...FFFFH	0...65535	1	[-]
Minimum threshold for integral of absolute value of pedal gradient for detection of vandalism					
C_FAC_PV_GRD_ABSV_INT_MAX	1	0...FFFFH	0...65535	1	[-]
Maximum threshold for integral of absolute value of pedal gradient for detection of vandalism					
C_FAC_PV_GRD_ABSV_INT_DEC	1	0...FFFFH	0...65535	1	[-]
Decrement value for integral of absolute value of pedal gradient for detection of vandalism					
C_T_LAMB_LS_UP_NOT_VLD_MAX	1	0...FFFFH	0...655.35	0.01	[s]
Time since LAMB_VLS_UP is not valid					
C_T_WOUT_NS_AD_INH_NS_DIAG	1	0...FFFFH	0...65535	1	[s]
Time after last NOx sensor gain adaptation					
LC_NS_DIAG_EXT_ADJ_ENA	1	0...1H	0...1	1	[-]
Enable inhibition of NOx sensor diags by external adjustment					
LC_T_WOUT_NS_AD	1	0...1H	0...1	1	[-]
Enable signal LV_NS_AD_REQ for detection of active gain adaptation for calculation of time after gain calculation					
C_NTLD_T_WOUT_NS_AD	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx trap loading degree for detection of active gain adaptation for calculation of time after gain adaptation					
LC_NS_SHIFT_INH_CAT_DIAG	1	0...1H	0...1	1	[-]
Shift diagnosis will be inhibited if cat diagnosis is active					

General information:

Each type of OBDII diagnosis will be inhibited, if any used input signal is not valid. Additionally, each diagnosis may be enabled or disabled without any cross dependencies.

The calculation have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

Application conditions:

Initialisation: at reset:

FAC_PV_GRD_ABSV_INT = 0
PV_AV_OLD_AVL = 0
T_LAMB_LS_UP_NOT_VLD[k] = 0 *k = 1...NC_CBK_EX_NR*

Recurrence: 10 ms

Activation:

Deactivation:

Formula section:

FAC_PV_GRD_ABSV_INT = FAC_PV_GRD_ABSV_INT
+ 16 * ABS(PV_AV - PV_AV_OLD_AVL)
- C_FAC_PV_GRD_ABSV_INT_DEC

PV_AV_OLD_AVL = PV_AV

The values PV_AV and PV_AV_OLD_AVL should be normalized to the resolution, which results in an easy addition of the hex values.

For each lambda sensor: k **= 1...NC_CBK_EX_NR**

```
If      LV_LAMB_LS_UP_VLD[k]
Then    T_LAMB_LS_UP_NOT_VLD[k] = 0
Else    increment T_LAMB_LS_UP_NOT_VLD[k]          10ms
Endif
```

Application conditions:

Initialisation: at reset and at deactivation:

LV_INH_DIAG_NS_OBD_2_HTP[i] = 1
LV_INH_DIAG_NS_OBD_2_AVL[i] = 1
LV_INH_DIAG_NS_OBD_2_OFS[i] = 1
LV_INH_DIAG_NS_OBD_2_DYN[i] = 1
LV_INH_DIAG_NS_OBD_2_ACT[i] = 1
LV_INH_DIAG_NS_OBD_2_PLAUS[i] = 1
LV_INH_DIAG_NS_OBD_2_GAIN[i] = 1
LV_INH_DIAG_NS_OBD_2_RAW[i] = 1
LV_INH_DIAG_NS_OBD_2_SHIFT[i] = 1
LV_PV_GRD_NS_DIAG = 0
T_WOUT_NS_AD = 65535s

Recurrence: 1 s

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

Temporary flag (shows an error on any lambda sensor)

```
#if NC_CBK_EX_NR = 1
        LV_ERR_LAMB_TMP =   LV_ERR_LS_UP[1]
        OR
                            LV_ERR_LS_DOWN[1]
        OR
                            LV_ERR_TTIP_MES_LSH_UP[1]
        OR
                            LV_ERR_CTL_LSL_UP[1]                      OR
                            LV_ERR_OFS_LSL_UP[1]                      OR
                    T_LAMB_LS_UP_NOT_VLD[1] > C_T_LAMB_LS_UP_NOT_VLD_MAX

#endif
#if NC_CBK_EX_NR = 2
        LV_ERR_LAMB_TMP =   (LV_ERR_LS_UP[1] OR LV_ERR_LS_UP[2])      OR
                            (LV_ERR_LS_DOWN[1] OR LV_ERR_LS_DOWN[2])  OR
                            (LV_ERR_TTIP_MES_LSH_UP[1]                OR
                            LV_ERR_TTIP_MES_LSH_UP[2])                OR
                            LV_ERR_CTL_LSL_UP[1]                      OR
                            LV_ERR_CTL_LSL_UP[2]                      OR
                            LV_ERR_OFS_LSL_UP[1]                      OR
                            LV_ERR_OFS_LSL_UP[2]                      OR
                  T_LAMB_LS_UP_NOT_VLD[1] > C_T_LAMB_LS_UP_NOT_VLD_MAX OR
                  T_LAMB_LS_UP_NOT_VLD[2] > C_T_LAMB_LS_UP_NOT_VLD_MAX
#endif
```

Temporary flag for fuel depending errors

```
LV_ERR_FUEL_TMP =   LV_ERR_EL_CPS                             OR
                    LV_ERR_DIAGCPS                            OR
                    LV_ERR_FUP                                OR
                    LV_ERR_FUP_MFP_PLAUS                      OR
                    LV_ERR_H_PRS_SYS                          OR
                    LV_ERR_FUP_ORNG                           OR
                    LV_ERR_FUP_ST                             OR
                    LV_ERR_VCV
```

Time after last NOx sensor gain adaptation

```
If          (LV_NS_AD_REQ = 1 AND LC_T_WOUT_NS_AD = 1)            OR
            NTLD > C_NTLD_T_WOUT_NS_AD
```

```
Then        T_WOUT_NS_AD = 0
Else        T_WOUT_NS_AD = T_WOUT_NS_AD + 1s
Endif
```

Inhibition mode of Heater diagnosis

```
case selection on C_STATE_INH_NS_HTP[i]
   C_STATE_INH_NS_HTP[i] = 0 ('AUTO'):
      if   LV_ERR_NS_OBD_1[i] = 1                                        OR
           LV_ERR_NS_CAN_MSG_LOST[i] = 1                                 OR
           LV_ERR_NS_CAN_BOFF = 1                                        OR
           LV_ERR_MAF = 1                                                OR
           LV_ERR_MAP = 1                                                OR
           LV_ERR_MAP_TPS_PLAUS = 1                                      OR
           (Bit 0 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
           LC_NS_DIAG_EXT_ADJ_ENA = 1)                                   OR
            (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_HTP_BIT_SEL[i]) != 0
      then   LV_INH_DIAG_NS_OBD_2_HTP[i] = 1
      else   LV_INH_DIAG_NS_OBD_2_HTP[i] = 0
      endif

   C_STATE_INH_NS_HTP[i] = 1 ('DISABLE'):
      LV_INH_DIAG_NS_OBD_2_HTP[i] = 1

   C_STATE_INH_NS_HTP[i] = 2 ('ENABLE'):
      LV_INH_DIAG_NS_OBD_2_HTP[i] = 0
end case selection
```

Inhibition mode of signal availability diagnosis

```
case selection on C_STATE_INH_NS_AVL[i]
   C_STATE_INH_NS_AVL[i] = 0 ('AUTO'):
      If        FAC_PV_GRD_ABSV_INT < C_FAC_PV_GRD_ABSV_INT_MIN
      Then      LV_PV_GRD_NS_DIAG = 0
      Else      If        FAC_PV_GRD_ABSV_INT > C_FAC_PV_GRD_ABSV_INT_MAX
                Then      LV_PV_GRD_NS_DIAG = 1
                Endif
      Endif

      if   LV_ERR_NS_OBD_1[i] = 1                                        OR
           LV_ERR_NS_CAN_MSG_LOST[i] = 1                                 OR
```

```
               LV_ERR_NS_CAN_BOFF = 1
                                                                                OR
               LV_PV_GRD_NS_DIAG = 1                                            OR
               (Bit 3 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
               LC_NS_DIAG_EXT_ADJ_ENA = 1)                                      OR
                (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_AVL_BIT_SEL[i]) != 0
        then   LV_INH_DIAG_NS_OBD_2_AVL[i] = 1
        else   LV_INH_DIAG_NS_OBD_2_AVL[i] = 0
        endif

    C_STATE_INH_NS_AVL[i] = 1 ('DISABLE'):
        LV_INH_DIAG_NS_OBD_2_AVL[i] = 1

    C_STATE_INH_NS_AVL[i] = 2 ('ENABLE'):
        LV_INH_DIAG_NS_OBD_2_AVL[i] = 0
end case selection
```

Inhibition mode of signal offset diagnosis

```
case selection on C_STATE_INH_NS_OFS[i]
    C_STATE_INH_NS_OFS[i] = 0 ('AUTO'):
        if     LV_ERR_NS_OBD_1[i] = 1                                           OR
               LV_ERR_NS_CAN_MSG_LOST[i] = 1                                    OR
               LV_ERR_NS_CAN_BOFF = 1                                           OR
               (Bit 2 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
               LC_NS_DIAG_EXT_ADJ_ENA = 1)                                      OR
                (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_OFS_BIT_SEL[i]) != 0
        then   LV_INH_DIAG_NS_OBD_2_OFS[i] = 1
        else   LV_INH_DIAG_NS_OBD_2_OFS[i] = 0
        endif

    C_STATE_INH_NS_OFS[i] = 1 ('DISABLE'):
        LV_INH_DIAG_NS_OBD_2_OFS[i] = 1

    C_STATE_INH_NS_OFS[i] = 2 ('ENABLE'):
        LV_INH_DIAG_NS_OBD_2_OFS[i] = 0
end case selection
```

Inhibition mode of signal dynamic diagnosis

```
case selection on C_STATE_INH_NS_DYN[i]
```

```
    C_STATE_INH_NS_DYN[i] = 0 ('AUTO'):
        if    LV_ERR_NS_OBD_1[i] = 1                                      OR
              LV_ERR_NS_CAN_MSG_LOST[i] = 1                               OR
              LV_ERR_NS_CAN_BOFF = 1                                      OR
              LV_ERR_LAMB_TMP = 1                                         OR
              LV_ERR_FUEL_TMP = 1                                         OR
              LV_ERR_MAF = 1                                              OR
              LV_ERR_MAP = 1
                                                                          OR
              LV_ERR_MAP_TPS_PLAUS = 1                                    OR
              (Bit 7 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
              LC_NS_DIAG_EXT_ADJ_ENA = 1)                                 OR
               (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_DYN_BIT_SEL[i]) != 0
        then   LV_INH_DIAG_NS_OBD_2_DYN[i] = 1
        else   LV_INH_DIAG_NS_OBD_2_DYN[i] = 0
        endif

    C_STATE_INH_NS_DYN[i] = 1 ('DISABLE'):
        LV_INH_DIAG_NS_OBD_2_DYN[i] = 1

    C_STATE_INH_NS_DYN[i] = 2 ('ENABLE'):
        LV_INH_DIAG_NS_OBD_2_DYN[i] = 0
end case selection
```

Inhibition mode of signal activity diagnosis

```
case selection on C_STATE_INH_NS_ACT[i]
    C_STATE_INH_NS_ACT[i] = 0 ('AUTO'):
        if    LV_ERR_NS_OBD_1[i] = 1                                      OR
              LV_ERR_NS_CAN_MSG_LOST[i] = 1                               OR
              LV_ERR_NS_CAN_BOFF = 1
                                                                          OR
              (Bit 8 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
              LC_NS_DIAG_EXT_ADJ_ENA = 1)                                 OR
               (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_ACT_BIT_SEL[i]) != 0
        then   LV_INH_DIAG_NS_OBD_2_ACT[i] = 1
        else   LV_INH_DIAG_NS_OBD_2_ACT[i] = 0
        endif

    C_STATE_INH_NS_ACT[i] = 1 ('DISABLE'):
        LV_INH_DIAG_NS_OBD_2_ACT[i] = 1
```

```
    C_STATE_INH_NS_ACT[i] = 2 ('ENABLE'):
      LV_INH_DIAG_NS_OBD_2_ACT[i] = 0
  end case selection
```

Inhibition mode of signal plausibility diagnosis (LSL_UP_DOWN, PUC, AFR, STOP)

```
  case selection on C_STATE_INH_NS_PLAUS[i]
    C_STATE_INH_NS_PLAUS[i] = 0 ('AUTO'):
      if   LV_ERR_NS_OBD_1[i] = 1                                OR
           LV_ERR_NS_CAN_MSG_LOST[i] = 1                         OR
           LV_ERR_NS_CAN_BOFF = 1                                OR
           LV_ERR_LAMB_TMP = 1                                   OR
           LV_ERR_FUEL_TMP = 1                                   OR
           LV_ERR_MAF = 1                                        OR
           LV_ERR_MAP = 1                                        OR
           LV_ERR_MAP_TPS_PLAUS = 1                              OR
           (Bit 1 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
           LC_NS_DIAG_EXT_ADJ_ENA = 1)                           OR
           T_WOUT_NS_AD < C_T_WOUT_NS_AD_INH_NS_DIAG             OR
           STATE_NS[i] != 0                                      OR
            (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_PLAUS_BIT_SEL[i]) != 0
      then    LV_INH_DIAG_NS_OBD_2_PLAUS[i] = 1
      else    LV_INH_DIAG_NS_OBD_2_PLAUS[i] = 0
      endif

    C_STATE_INH_NS_PLAUS[i] = 1 ('DISABLE'):
      LV_INH_DIAG_NS_OBD_2_PLAUS[i] = 1

    C_STATE_INH_NS_PLAUS[i] = 2 ('ENABLE'):
      LV_INH_DIAG_NS_OBD_2_PLAUS[i] = 0
  end case selection
```

Inhibition mode of NOx signal gain diagnosis

```
  case selection on C_STATE_INH_NS_GAIN[i]
    C_STATE_INH_NS_GAIN[i] = 0 ('AUTO'):
      if   LV_ERR_NS_OBD_1[i] = 1                                OR
           LV_ERR_NS_CAN_MSG_LOST[i] = 1                         OR
           LV_ERR_NS_CAN_BOFF = 1                                OR
           LV_ERR_LAMB_TMP = 1                                   OR
           LV_ERR_FUEL_TMP = 1                                   OR
           LV_ERR_MAF = 1                                        OR
           LV_ERR_MAP = 1                                        OR
           LV_ERR_MAP_TPS_PLAUS = 1                              OR
```

```
          LV_ERR_NT_AGI = 1                                              OR
          LV_ERR_NS_OBD_2[i] = 1                                         OR
          LV_ERR_CAT_DIAG_AFL[1] = 1                                     OR
          (NC_CBK_EX_NR = 2 AND LV_ERR_CAT_DIAG_AFL[2] = 1)               OR
          LV_ERR_TEG_PCAT_DOWN = 1                                       OR
          LV_SO2P_REQ = 1                                                OR
          (Bit 12 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
          LC_NS_DIAG_EXT_ADJ_ENA = 1)                                     OR
           (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_GAIN_BIT_SEL[i]) != 0
    then  LV_INH_DIAG_NS_OBD_2_GAIN[i] = 1
    else  LV_INH_DIAG_NS_OBD_2_GAIN[i] = 0
    endif

  C_STATE_INH_NS_GAIN[i] = 1 ('DISABLE'):
    LV_INH_DIAG_NS_OBD_2_GAIN[i] = 1

  C_STATE_INH_NS_GAIN[i] = 2 ('ENABLE'):
    LV_INH_DIAG_NS_OBD_2_GAIN[i] = 0
end case selection
```

Inhibition mode of NOx raw emission diagnosis

```
case selection on C_STATE_INH_NS_RAW[i]
  C_STATE_INH_NS_RAW[i] = 0 ('AUTO'):
    if    LV_ERR_NS_OBD_1[i] = 1                                         OR
          LV_ERR_NS_CAN_MSG_LOST[i] = 1                                  OR
          LV_ERR_NS_CAN_BOFF = 1                                         OR
          LV_ERR_LAMB_TMP = 1                                            OR
          LV_ERR_FUEL_TMP = 1                                            OR
          LV_ERR_MAF = 1                                                 OR
          LV_ERR_MAP = 1                                                 OR
          LV_ERR_MAP_TPS_PLAUS = 1                                       OR
          (Bit 13 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
          LC_NS_DIAG_EXT_ADJ_ENA = 1)                                     OR
           (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_RAW_BIT_SEL[i]) != 0
    then  LV_INH_DIAG_NS_OBD_2_RAW[i] = 1
    else  LV_INH_DIAG_NS_OBD_2_RAW[i] = 0
    endif

  C_STATE_INH_NS_RAW[i] = 1 ('DISABLE'):
    LV_INH_DIAG_NS_OBD_2_RAW[i] = 1

  C_STATE_INH_NS_RAW[i] = 2 ('ENABLE'):
```

```
        LV_INH_DIAG_NS_OBD_2_RAW[i] = 0
end case selection
```

Inhibition mode of NOx sensor internal setpoint shift diagnosis

```
case selection on C_STATE_INH_NS_SHIFT[i]
    C_STATE_INH_NS_SHIFT[i] = 0 ('AUTO'):
        if  LV_ERR_NS_OBD_1[i] = 1                                          OR
            LV_ERR_NS_CAN_MSG_LOST[i] = 1                                   OR
            LV_ERR_NS_CAN_BOFF = 1                                          OR
            LV_ERR_LAMB_TMP = 1                                             OR
            LV_ERR_FUEL_TMP = 1                                             OR
            LV_ERR_MAF = 1                                                  OR
            LV_ERR_MAP = 1                                                  OR
            LV_ERR_MAP_TPS_PLAUS = 1                                        OR
            (LC_NS_SHIFT_INH_CAT_DIAG                        AND
            ( LV_LAM_ADJ_CAT_DIAG[1] = 1                OR
              (NC_CBK_EX_NR = 2 AND LV_LAM_ADJ_CAT_DIAG[2] = 1) ) )         OR
            (Bit 14 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
            LC_NS_DIAG_EXT_ADJ_ENA = 1)                                     OR
             (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_SHIFT_BIT_SEL[i]) != 0
        then    LV_INH_DIAG_NS_OBD_2_SHIFT[i] = 1
        else    LV_INH_DIAG_NS_OBD_2_SHIFT[i] = 0
        endif

    C_STATE_INH_NS_SHIFT[i] = 1 ('DISABLE'):
        LV_INH_DIAG_NS_OBD_2_SHIFT[i] = 1

    C_STATE_INH_NS_SHIFT[i] = 2 ('ENABLE'):
        LV_INH_DIAG_NS_OBD_2_SHIFT[i] = 0
end case selection
```

Inhibition mode of NOx sensor version diagnosis

```
case selection on C_STATE_INH_NS_VERS[i]
    C_STATE_INH_NS_VERS[i] = 0 ('AUTO'):
        if  LV_ERR_NS_CAN_MSG_LOST[i] = 1                                   OR
            LV_ERR_NS_CAN_BOFF = 1                                          OR
            (Bit 15 (STATE_INH_NS_OBD_2_EXT_ADJ[i]) = 1 AND
            LC_NS_DIAG_EXT_ADJ_ENA = 1)                                     OR
             (STATE_ERR_NS_OBD_2[i] bitwise AND C_ERR_NS_VERS_BIT_SEL[i]) != 0
        then    LV_INH_DIAG_NS_OBD_2_VERS[i] = 1
        else    LV_INH_DIAG_NS_OBD_2_VERS[i] = 0
```

```
        endif

    C_STATE_INH_NS_VERS[i] = 1 ('DISABLE'):
        LV_INH_DIAG_NS_OBD_2_VERS[i] = 1

    C_STATE_INH_NS_VERS[i] = 2 ('ENABLE'):
        LV_INH_DIAG_NS_OBD_2_VERS[i] = 0
  end case selection
```

Configuration for diagnostic symptoms:

Heater diag (B07A)

Diagnostic NS_HTP	Symptom description	Symptom	Filter type
NOx sensor heater diagnosis	Heater performance too low at start	SYM_0	STD_INI
	Heater performance too low during normal operation	*SYM_1*	
	Power supply failure	*SYM_2*	
		SYM_3	

Diagnostic NS_HTP	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx sensor heater diagnosis	Heater performance too low at start	*SYM_0*						
	Heater performance too low during normal operation	*SYM_1*						
	Power supply failure	*SYM_2*						
		SYM_3						

Signal availability (B07B)

Diagnostic NS_AVL	Symptom description	Symptom	Filter type
NOx sensor availability diagnosis	Signal not available at start	SYM_0	STD_INI
	Signal not available during normal operation	*SYM_1*	
		SYM_2	
		SYM_3	

Diagnostic NS_AVL	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx sensor availability diagnosis	Signal not available at start	*SYM_0*						
	Signal not available during normal operation	*SYM_1*						
		SYM_2						
		SYM_3						

Signal offset (B07D)

Diagnostic NS_OFS	Symptom description	Symptom	Filter type
NOx sensor offset diagnosis	NOx signal offset failure	SYM_0	STD_INI
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic NS_OFS	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx sensor offset diagnosis	NOx signal offset failure	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

Binary lambda signal dynamic (B07E)

Diagnostic NS_VLS_DYN	Symptom description	Symptom	Filter type
NOx sensor binary dynamic diagnosis	Binary dynamic too low	SYM_0	STD_INI
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic NS_VLS_DYN	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx sensor binary dynamic diagnosis	Binary dynamic too low	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

Signal activity (B07F)

Diagnostic NS_ACT	Symptom description	Symptom	Filter type
NOx sensor activity diagnosis	NOx signal activity too low	SYM_0	STD_INI
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic NS_ACT	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx sensor activity diagnosis	NOx signal activity too low	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

Signal plausibility up/down (B07X)

Diagnostic NS_LSL_UP_DOWN	Symptom description	Symptom	Filter type
NOx sensor plausibility up/down diagnosis	Lambda signal up/downstream not plausible	SYM_0	STD_INI
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic NS_LSL_UP_DOWN	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx sensor plausibility up/down diagnosis	Lambda signal up/downstream not plausible	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

Signal plausibility at PUC (B07Y)

Diagnostic **NS_PUC**	**Symptom description**	**Symptom**	**Filter type**
NOx sensor plausibility diagnosis at PUC	Binary lambda signal too rich	SYM_0	STD_INI
	Linear lambda too rich	*SYM_1*	
	NOx signal too low	*SYM_2*	
	NOx signal too high	*SYM_3*	

Diagnostic **NS_PUC**	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
NOx sensor plausibility diagnosis at PUC	Binary lambda signal too rich	*SYM_0*						
	Linear lambda too rich	*SYM_1*						
	NOx signal too low	*SYM_2*						
	NOx signal too high	*SYM_3*						

Signal plausibility at rich conditions (B07Z)

Diagnostic **NS_AFR**	**Symptom description**	**Symptom**	**Filter type**
NOx sensor plausibility diagnosis at rich conditions	Binary lambda signal too lean	SYM_0	STD_INI
	Linear lambda too lean	*SYM_1*	
	NOx signal too low	*SYM_2*	
		SYM_3	

Diagnostic **NS_AFR**	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
NOx sensor plausibility diagnosis at rich conditions	Binary lambda signal too lean	*SYM_0*						
	Linear lambda too lean	*SYM_1*						
	NOx signal too low	*SYM_2*						
		SYM_3						

Regeneration stop observation (B080)

Diagnostic **NS_STOP**	**Symptom description**	**Symptom**	**Filter type**
NOx sensor diagnosis at regeneration stop	Regeneration agent failure	SYM_0	STD_INI
	Time out	*SYM_1*	
		SYM_2	
		SYM_3	

Diagnostic **NS_STOP**	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
NOx sensor diagnosis at regeneration stop	Regeneration agent failure	*SYM_0*						
	Time out	*SYM_1*						
		SYM_2						
		SYM_3						

NOx signal gain (B0A0)

Diagnostic NS_GAIN	Symptom description	Symptom	Filter type
NOx signal gain diagnosis	NOx signal gain failure	SYM_0	STD_INI
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic NS_GAIN	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx signal gain diagnosis	NOx signal gain failure	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

NOx raw emission (B0A1)

Diagnostic NS_RAW	Symptom description	Symptom	Filter type
NOx raw emission	NOx raw emission plausibility failure	SYM_0	STD_INI
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic NS_GAIN	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx raw emission	NOx raw emission plausibility failure	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

NOx sensor setpoint shift (B0A3)

Diagnostic NS_SHIFT	Symptom description	Symptom	Filter type
NOx sensor setpoint shift	Setpoint shift failure	SYM_0	STD_INI
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic NS_SHIFT	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx sensor setpoint shift	Setpoint shift failure	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

NOx sensor version (B0A4)

Diagnostic **NS_VERS**	**Symptom description**	**Symptom**	**Filter type**
NOx sensor version	Wrong version	SYM_0	STD_INI
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic **NS_VERS**	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
NOx sensor version	Wrong version	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

52.11 NOx sensor OBDII diag General

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_NS_OBD_2_ERR[NC_NOX_SENS_CONF]	O/V/S	0...FFH	0...255	1	-
Number of consecutive driving cycles at which a NOx sensor failure was detected					
LV_ERR_NS_OBD_2[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Result of NOx sensor OBD II diagnosis					
STATE_ERR_NS_OBD_2[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...6.5535E+4	1	-
Collection of all error bits of OBDII diagnosis					
STATE_END_NS_OBD_2[NC_NOX_SENS_CONF]	O/V	0...FFFFH	0...6.5535E+4	1	-
Collection of all diagnosis end bits of OBDII diagnosis					
LV_NS_LDC[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Limited dynamics for NOx sensor state evaluation					
MAF_INT_LDC_DLY_NS[NC_NOX_SENS_CONF]	V	0...FFFFH	0...1.82042E+3	0.02777778	g
Integral of air mass flow since limited dynamic conditions are fulfilled					
LAMB_LS_UP_MMV_NS[NC_CBK_EX_NR]	V	0...7FFFH	0...31.9990234	9.76563E-4	-
Moving mean value of upstream lambda					
LV_LAMB_LS_UP_LDC_SNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Limited dynamic of signal LAMB_LS_UP (one condition for each exhaust bank)					
LV_LAMB_LS_UP_LDC	V	0...1H	0...1	1	-
Limited dynamic of signal LAMB_LS_UP (one condition for all exhaust banks)					
MAF_MMV_NS	V	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Moving mean value of mass air flow					
N_MMV_NS	V	0...1FE0H	0...8.16E+3	1	rpm
Moving mean value of engine speed					

Input data:

LAMB_LS_UP[NC_CBK_EX_NR]	LV_INH_DIAG_NS_OBD_2_PLAUS[NC_NOX_SENS_CONF]	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_ST_END
MAF	MAF_KGH	N	NC_CBK_EX_NR
NC_NOX_SENS_CONF	LV_ERR_NS_HTP[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_HTP[NC_NOX_SENS_CONF]	LV_ERR_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]
LV_END_DIAG_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]	LV_ERR_NS_OFS[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_OFS[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_AVL[NC_NOX_SENS_CONF]
LV_ERR_NS_AVL[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_PUC[NC_NOX_SENS_CONF]	LV_ERR_NS_PUC[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_AFR[NC_NOX_SENS_CONF]
LV_ERR_NS_AFR[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_STOP[NC_NOX_SENS_CONF]	LV_ERR_NS_STOP[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_VLS_DYN[NC_NOX_SENS_CONF]
LV_ERR_NS_VLS_DYN[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_ACT[NC_NOX_SENS_CONF]	LV_ERR_NS_ACT[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_GAIN[NC_NOX_SENS_CONF]
LV_ERR_NS_GAIN[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_SHIFT[NC_NOX_SENS_CONF]	LV_ERR_NS_SHIFT[NC_NOX_SENS_CONF]	LV_END_DIAG_NS_VERS[NC_NOX_SENS_CONF]
LV_ERR_NS_VERS[NC_NOX_SENS_CONF]	FAC_NOX_NS_AD[NC_NOX_SENS_CONF]	C_FAC_NOX_NS_AD_AFS_MAX	C_FAC_NOX_NS_AD_AFS_MIN
RATIO_MMV_NS_SHIFT_DIAG[NC_NOX_SENS_CO	C_RATIO_NS_SHIFT_DIAG_AFS_MAX	C_RATIO_NS_SHIFT_DIAG_AFS_MIN	

NF]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_LAMB_LS_UP_MMV_NS	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Filter constant for calculation of moving mean value of upstream lambda					
C_CRLC_MAF_MMV_NS	1	0...FFFFH	0...0.99998474	1.52588E-5	-
filter constant for calculation of moving mean value of mass air flow					
C_CRLC_N_MMV_NS	1	0...FFFFH	0...0.99998474	1.52588E-5	-
filter constant for calculation of moving mean value of engine speed					
C_CTR_NS_OBD_2_ERR_MAX	1	0...FFH	0...255	1	-
Maximum number of consecutive driving cycles at which a NOx sensor failure was detected					
C_ERR_END_NS_OBD_2_BIT_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...6.5535E+4	1	-
Selection of end of diagnosis signal that will be evaluated before decrement counter					
C_ERR_RST_NS_OBD_2	1	0...2H	0...2	1	-
Selection of reset strategy of global NOx sensor OBD2 error					
C_MAF_DYW_NS	1	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Limited dynamic threshold of MAF					
C_MAF_INT_LDC_DLY_NS	1	0...FFFFH	0...1.82042E+3	0.02777778	g
MAF integral threshold after reaching limited dynamic conditions, before LV_NS_LDC is set					
C_N_DYW_NS	1	0...1FE0H	0...8.16E+3	1	rpm
Limited dynamic threshold of engine speed					
C_STATE_ERR_NS_OBD_2_SEL[NC_NOX_SENS_CONF]	1	0...FFFFH	0...6.5535E+4	1	-
Selection of STATE_ERR_NS_OBD_2 error bits					
IP_LAMB_LS_UP_DYW_NS	4	0...7FFFH	0...31.9990234	9.76563E-4	-
LDP_LAMB_LS_UP_IP_LAMB_DYW_NS	4	0...7FFFH	0...31.9990234	9.76563E-4	-
Lambda signal value of the WRAF sensor					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_FAC_MAF_INT_LDC	1	0...FFFFH	0...1.02398437	1.5625E-5	s
normalization factor for calculation of integral of MAF_KGH					

52.11.1 General information

This module consists a limited dynamics evaluation for the NOx sensor and the global NOx sensor OBD II error flag.

The reset strategy of this error bit LV_ERR_NS_OBD_2_i can be selected by C_ERR_RST_NS_OBD_2_i. It can be chosen between following possibilities for error bit reset:

- reset only at clear of error memory
- reset at every start of new driving cycle
- manual reset, permanent reset.

Application Condition

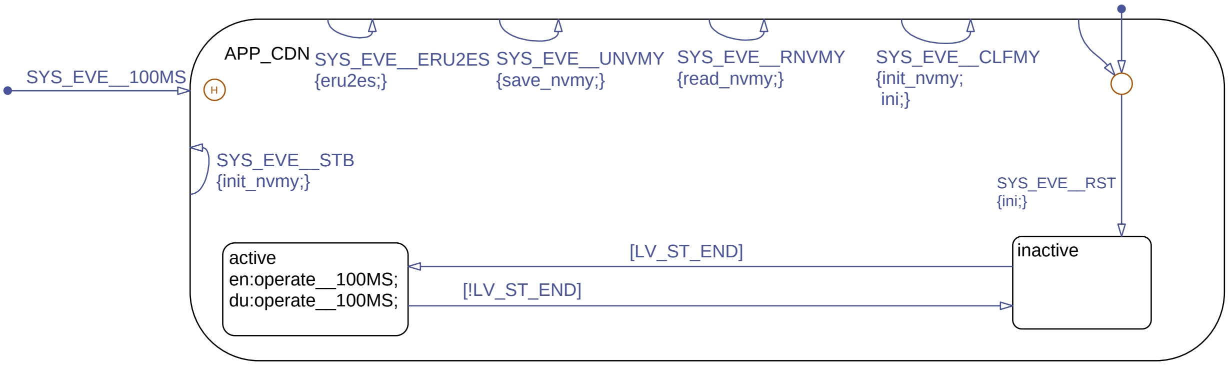
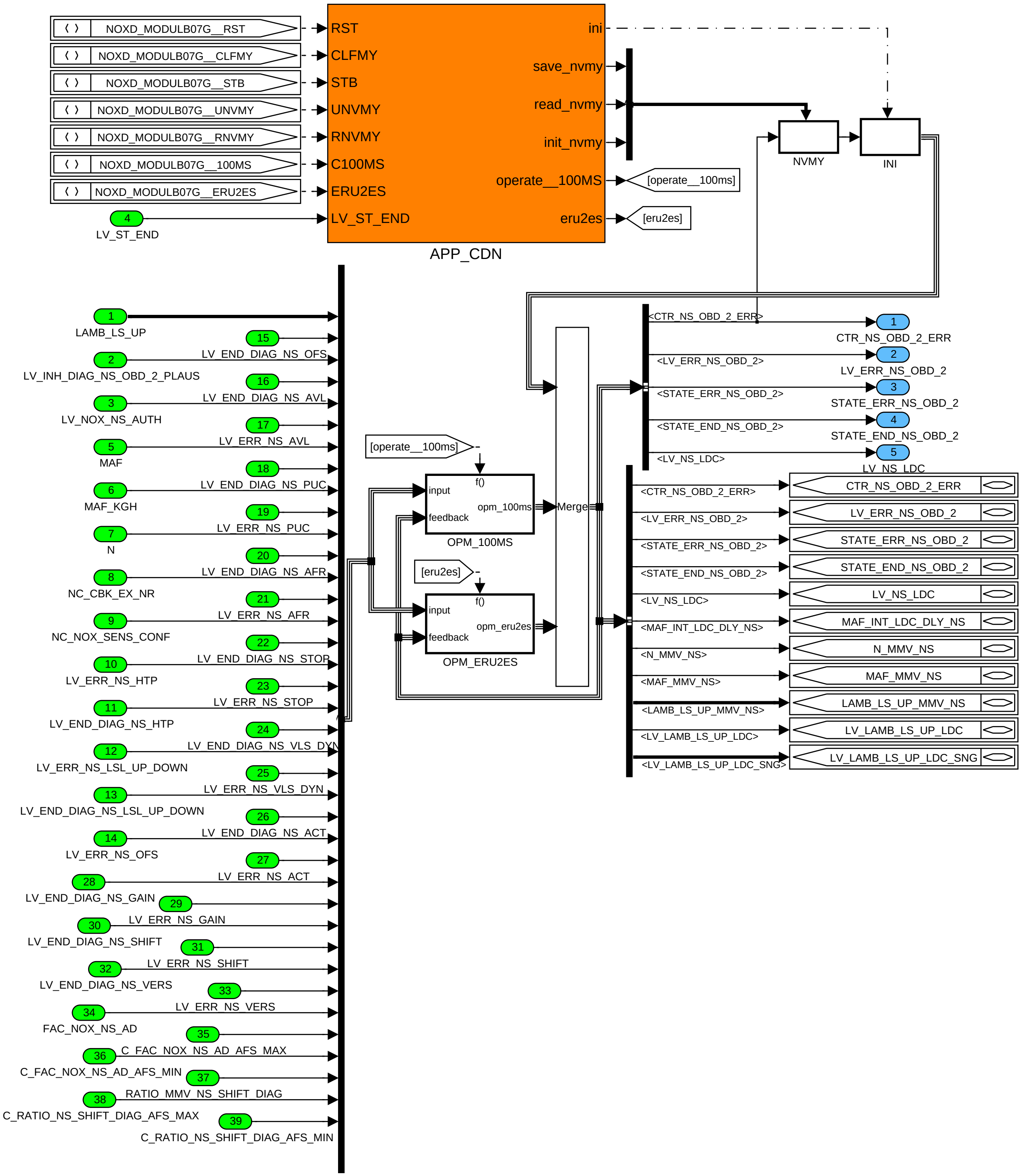


Figure 41 NOXD_MODULB07G/ APP_CDN/ Chart



SDA_SRS / SDA 4.0 08−Feb−2007

Figure 42 NOXD_MODULB07G

52.11.1.1 Initialization at reset or at clear failure memory

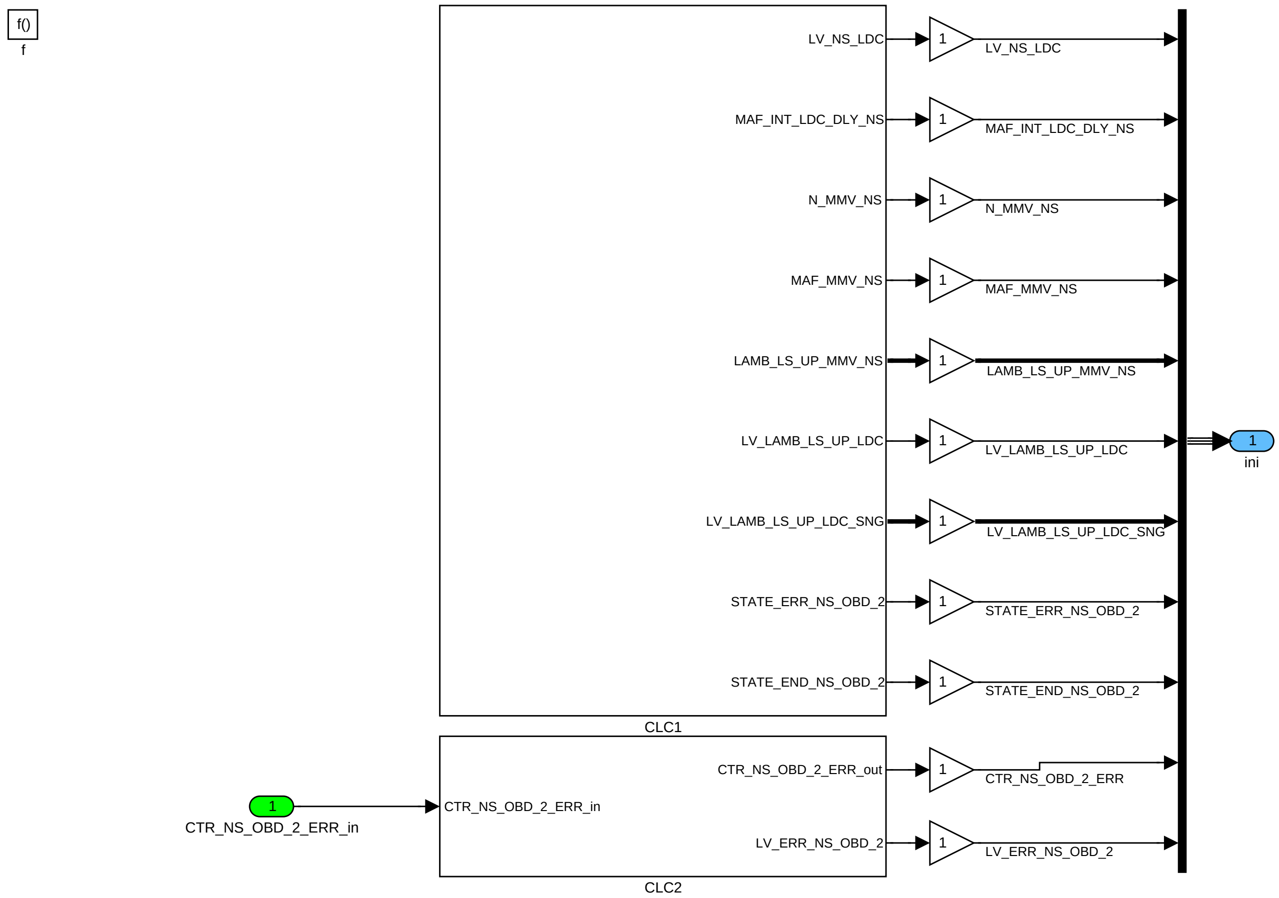


Figure 43 NOXD_MODULB07G/ INI

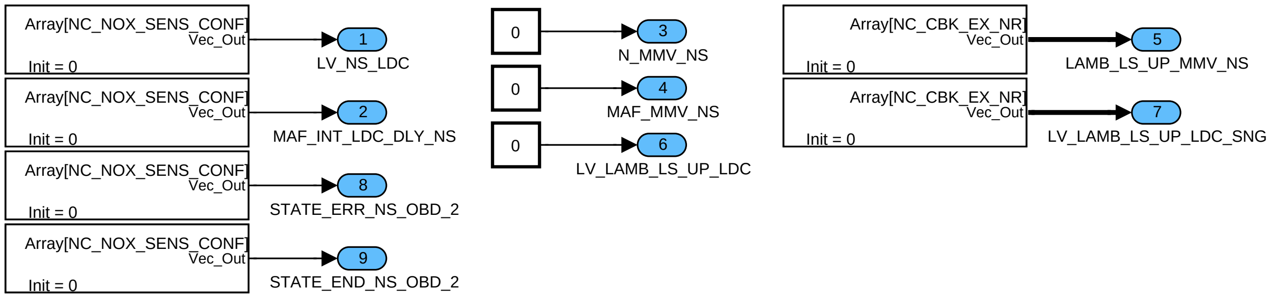


Figure 44 NOXD_MODULB07G/ INI/ CLC1

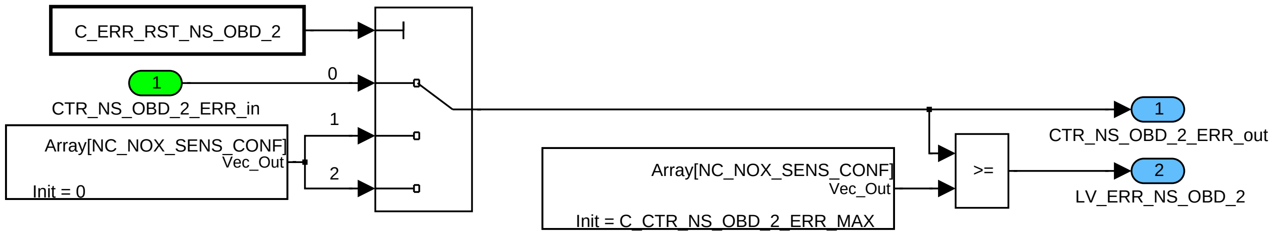


Figure 45 NOXD_MODULB07G/ INI/ CLC2

52.11.1.2 Management of non volatile memory

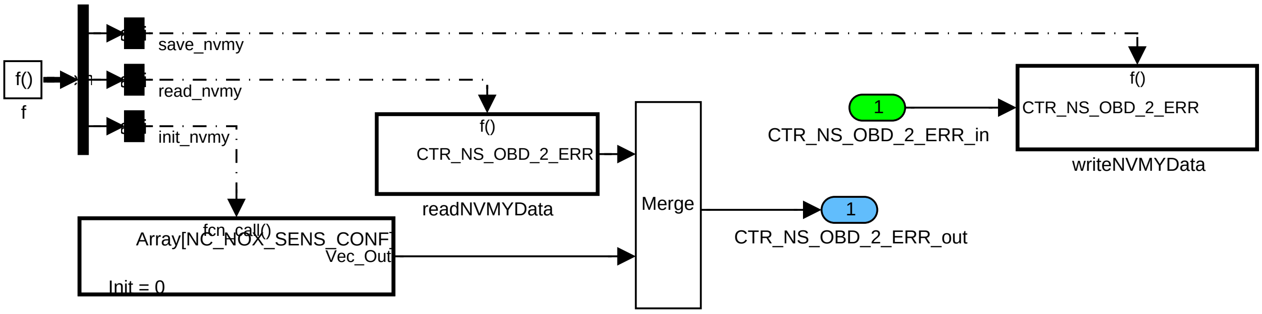


Figure 46 NOXD_MODULB07G/ NVMY

52.11.1.3 Loop control 100 ms

The calculation inside FOR_LOOP_LS have to be done for each upstream lambda sensor, controlled by configuration data NC_CBK_EX_NR.

If NC_CBK_EX_NR = 1, then calculation for i = 1.

If NC_CBK_EX_NR = 2, then calculation for i = 1 and 2.

The calculation inside NO_LOOP have to be done only once.

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

NC_NOX_SENS_CONF cannot be greater than NC_CBK_EX_NR.

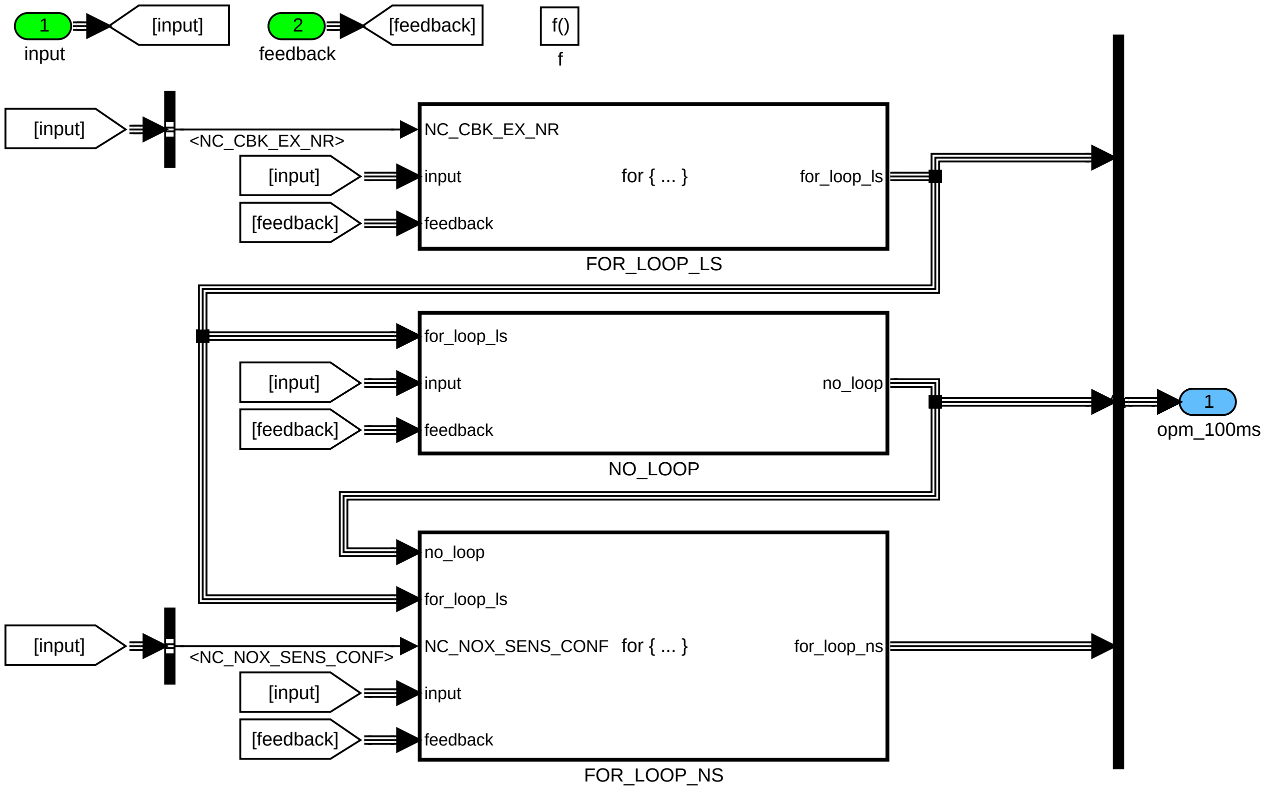


Figure 47 NOXD_MODULB07G/ OPM_100MS

Calculation for each upstream lambda sensor

Filter for upstream lambda

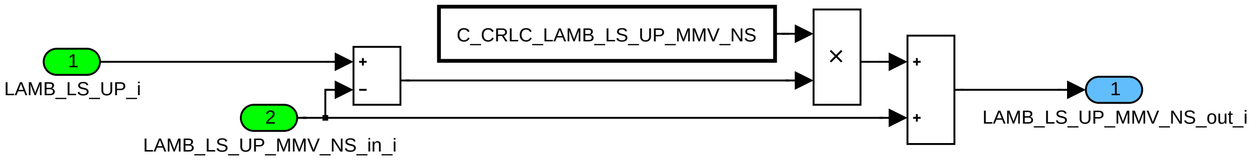


Figure 48 NOXD_MODULB07G/ OPM_100MS/ FOR_LOOP_LS/ CLC1

Check of deviation of upstream lambda from its mean value

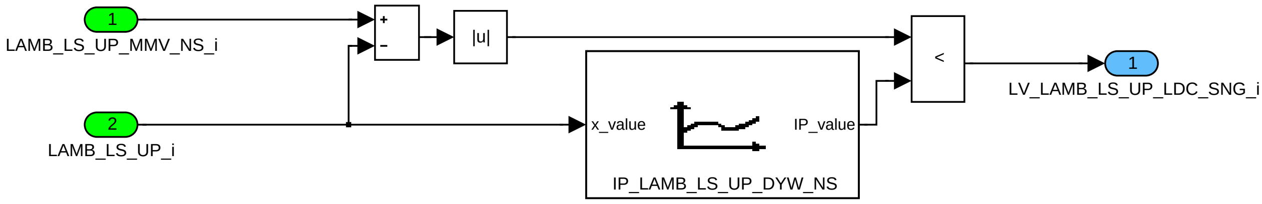


Figure 49 NOXD_MODULB07G/ OPM_100MS/ FOR_LOOP_LS/ CLC2

Bank independent calculation

Filter of motor speed

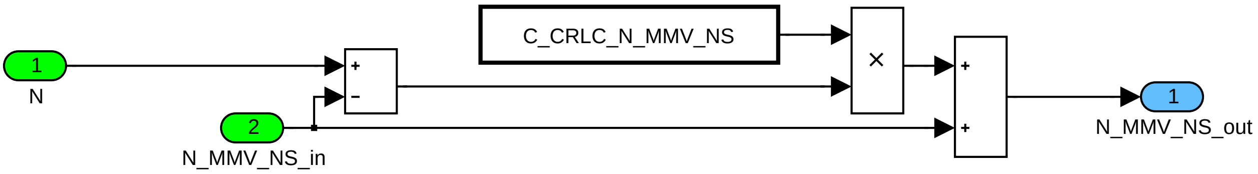


Figure 50 NOXD_MODULB07G/ OPM_100MS/ NO_LOOP/ CLC1

Filter of mass air flow

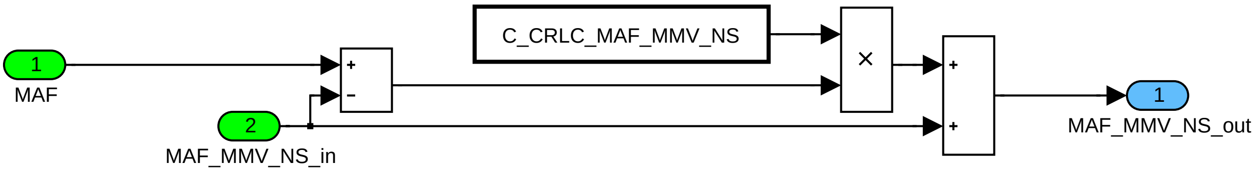


Figure 51 NOXD_MODULB07G/ OPM_100MS/ NO_LOOP/ CLC2

Lambda deviation signal for all banks

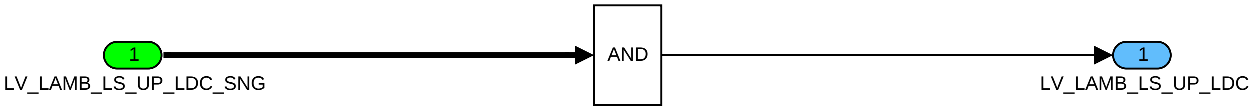


Figure 52 NOXD_MODULB07G/ OPM_100MS/ NO_LOOP/ CLC3

Calculation for each NOx sensor

Evaluation of limited dynamics

The dynamic is limited if several conditions are valid for a longer time, controlled by the MAF integral.

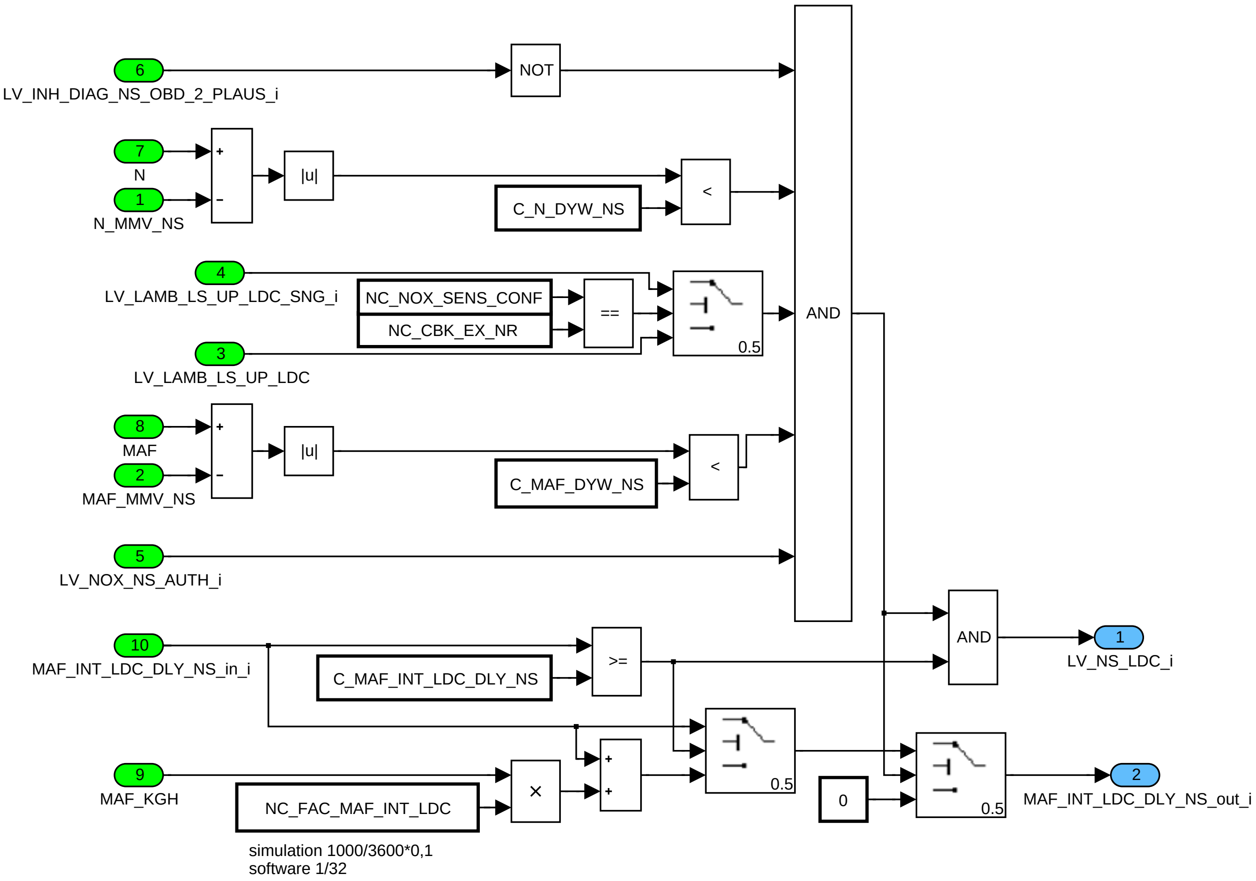


Figure 53 NOXD_MODULB07G/ OPM_100MS/ FOR_LOOP_NS/ CLC1

Collection of error information of all OBD II diagnosis types

The results of all diagnosis types will be collected in two status variables.

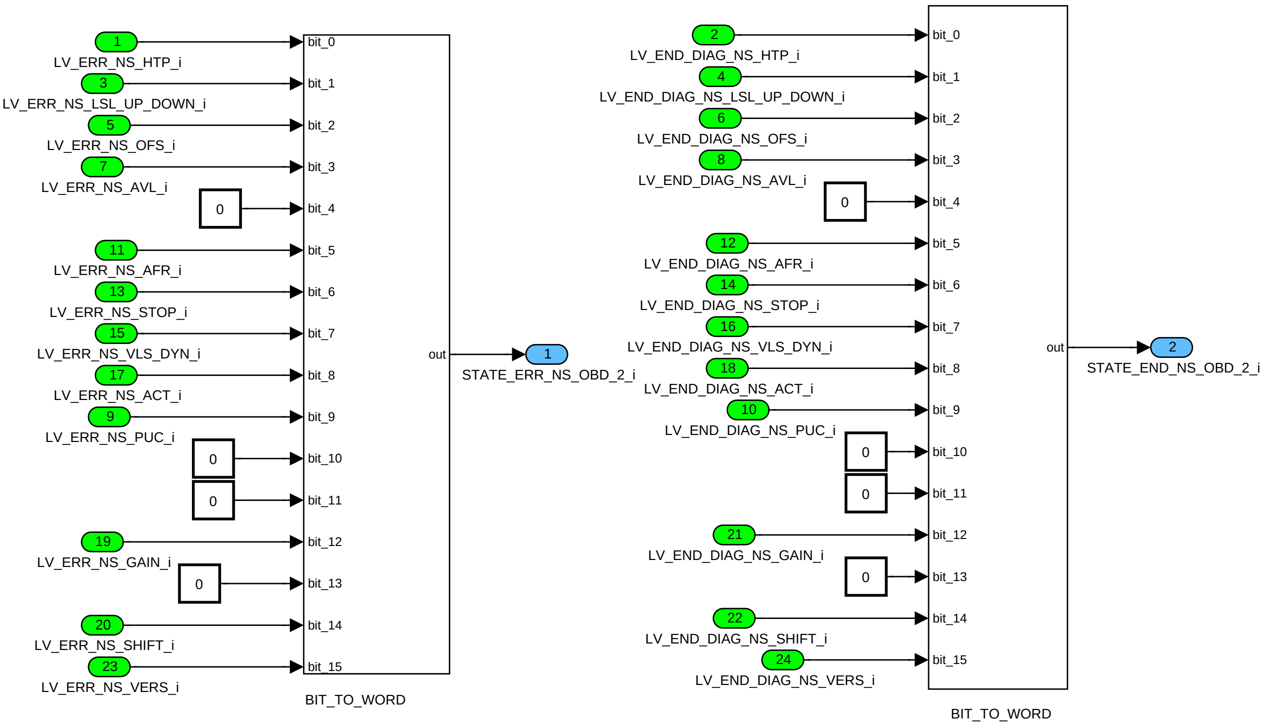


Figure 54 NOXD_MODULB07G/ OPM_100MS/ FOR_LOOP_NS/ CLC2

Global OBD II error bit

If any diagnosis detects an error, then the global error bit will be set.

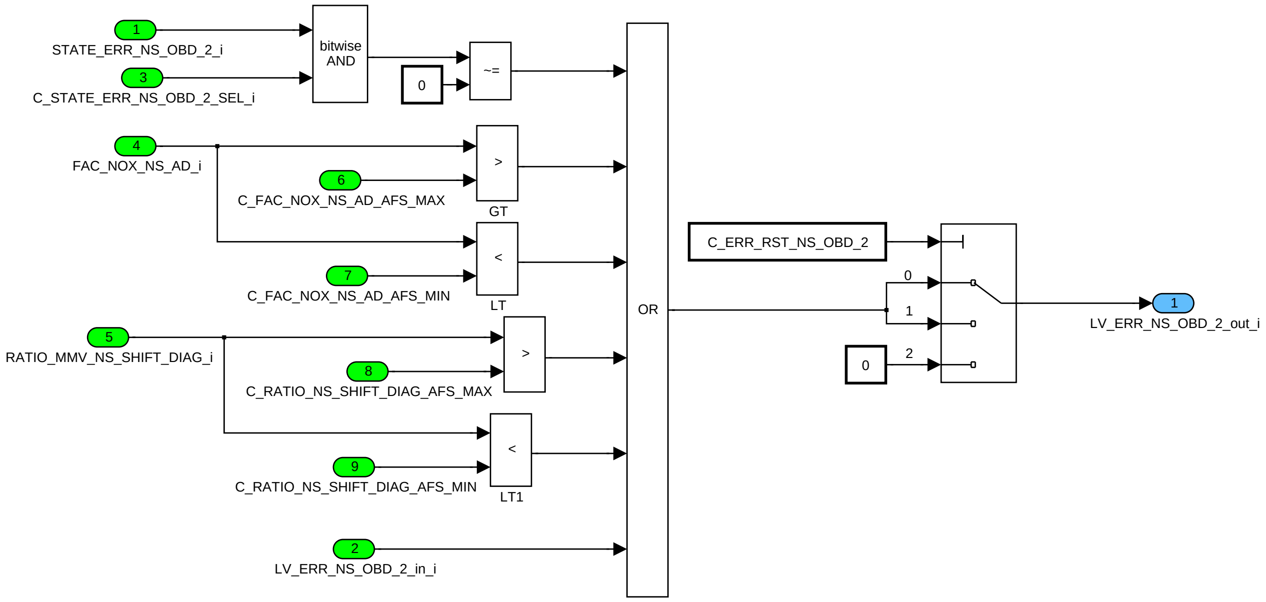


Figure 55 NOXD_MODULB07G/ OPM_100MS/ FOR_LOOP_NS/ CLC3

52.11.1.4 Loop control at engine run to engine stop

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

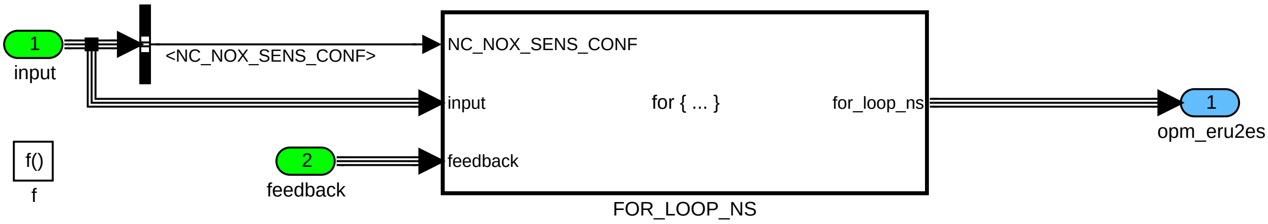


Figure 56 NOXD_MODULB07G/ OPM_ERU2ES

Error counter

For each NOx sensor a counter will be incremented if any OBD II error was detected or decremented if all selected end of OBD II diagnosis signals are valid. This is only done once at the end of the driving cycle. This selection will be done with the bit mask C_ERR_NS_OBD_2_BIT_SEL_i.

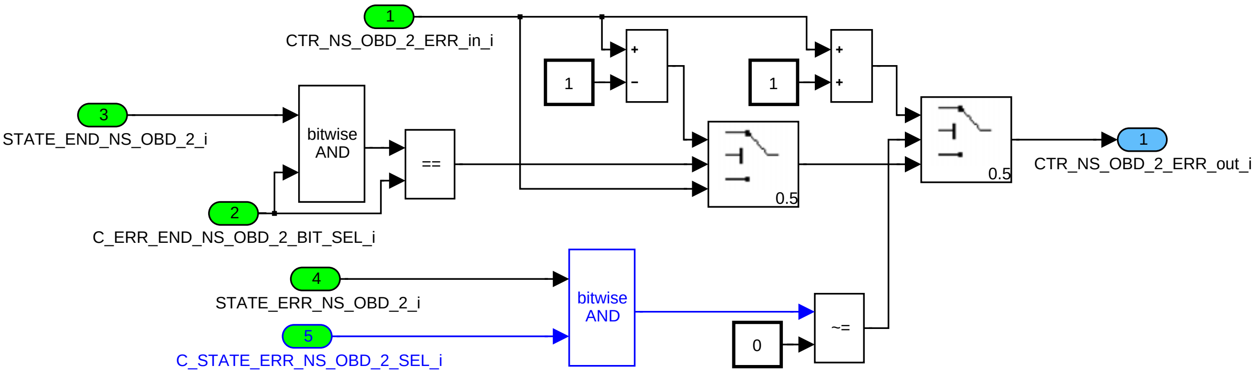


Figure 57 NOXD_MODULB07G/ OPM_ERU2ES/ FOR_LOOP_NS/ CLC1

Error flag for next driving cycle

Figure 58 NOXD_MODULB07G/ OPM_ERU2ES/ FOR_LOOP_NS/ CLC2

52.12 Heater performance check

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_HTP[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor heater diagnosis					
LV_ERR_NS_HTP[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of NOx sensor heater diagnosis					
ERR_SYM_NS_HTP[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor heater diagnosis					
STATE_NS_HTP_DIAG[NC_NOX_SENS_CONF]	V	0H 1H 2H 3H 4H	OFF START PASSIVE ACTIVE POWER	1	-
State of heater performance diagnosis at normal operation					
T_NS_HTP_DIAG[NC_NOX_SENS_CONF]	V	0...FFH	0...255	1	s
Timer for NOx sensor heater diagnosis					
LV_CDN_DIAG_NS_HTP[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of NOx sensor heater diagnosis					

Input data:

LV_ST_END	LV_INH_DIAG_NS_OBD_2_HTP[NC_NOX_SENS_CONF]	LV_PUC	LV_TEMP_NS_OK[NC_NOX_SENS_CONF]
LV_TNT_MIN_THD_2	LV_VB_NS_OK[NC_NOX_SENS_CONF]	MAF_INT_PUC_ACT	NC_IDX_DIAG_NS_HTP[NC_NOX_SENS_CONF]
NC_NOX_SENS_CONF	TNT_MDL_L	VB	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_HTP	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor heater diagnosis					
C_ABC_INC_NS_HTP	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor heater diagnosis					
C_ABC_MAX_NS_HTP	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx sensor heater diagnosis					
C_MAF_INT_PUC_HTP_MAX	1	0...FFFFH	0...2.91267E+3	0.04444444	g
MAF threshold for break off a heater performance diagnosis					
C_TNT_MDL_HTP_MIN	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Minimum temperature of NOx catalyst, at which a NOx sensor heater test can be started					
C_T_NS_HTP_MAX	1	0...FFH	0...255	1	s
Maximum time of invalid sensor temperature at normal operation					
C_T_NS_HTP_ST_MAX	1	0...FFH	0...255	1	s
Maximum heating up time of NOx sensor					
C_VB_MAX_NS	1	0...FFH	0...25.8984375	0.1015625	V
Maximum battery voltage for correct NOx sensor operation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VB_MIN_NS	1	0...FFH	0...25.8984375	0.1015625	V
Minimum battery voltage for correct NOx sensor operation					

Import actions:

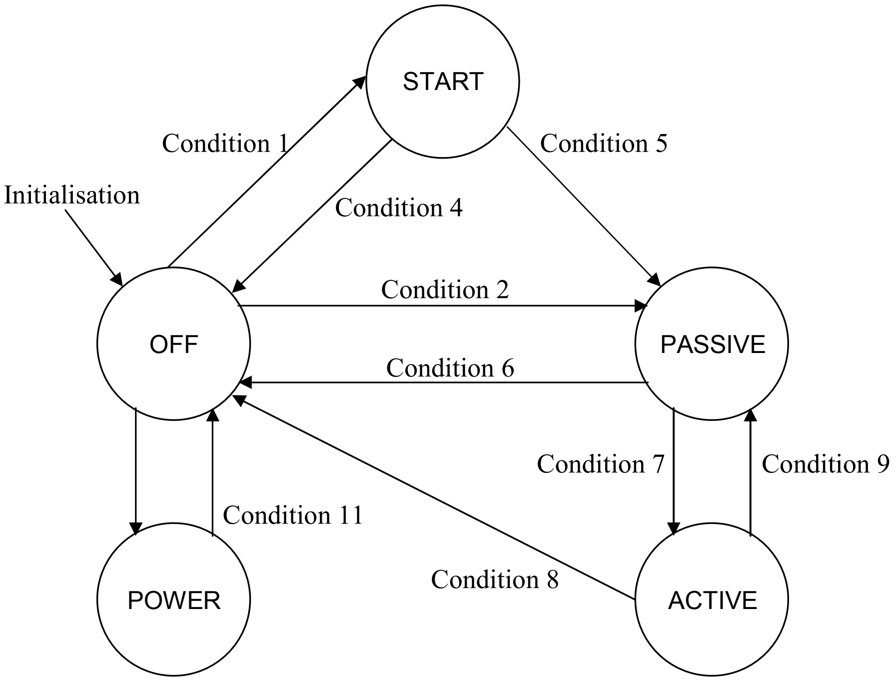
ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

52.12.1 General information

If the sensor is OK, then the diagnosis state machine is in the state PASSIVE, see next picture. If the temperature of the sensor is not OK, then a diagnosis will be made inside state ACTIVE. If the power supply is not OK, then a diagnosis will be made inside state POWER.

Additionally, at start-up a diagnosis of the sensor temperature behaviour will be made inside state START.

Signal flow diagram:



Application Condition

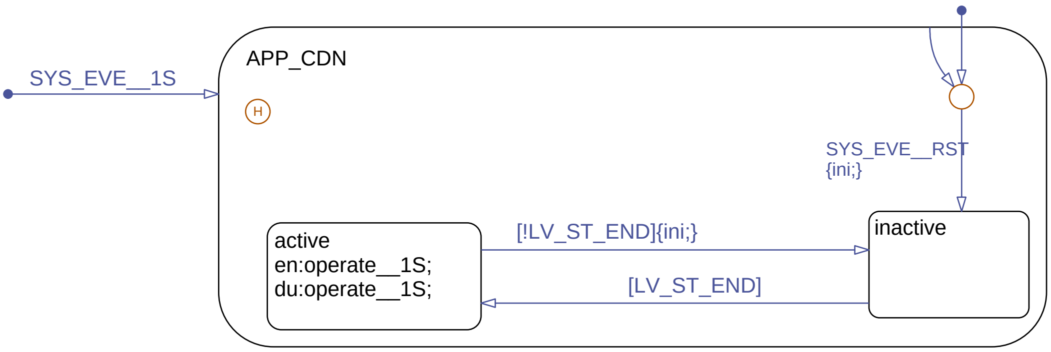
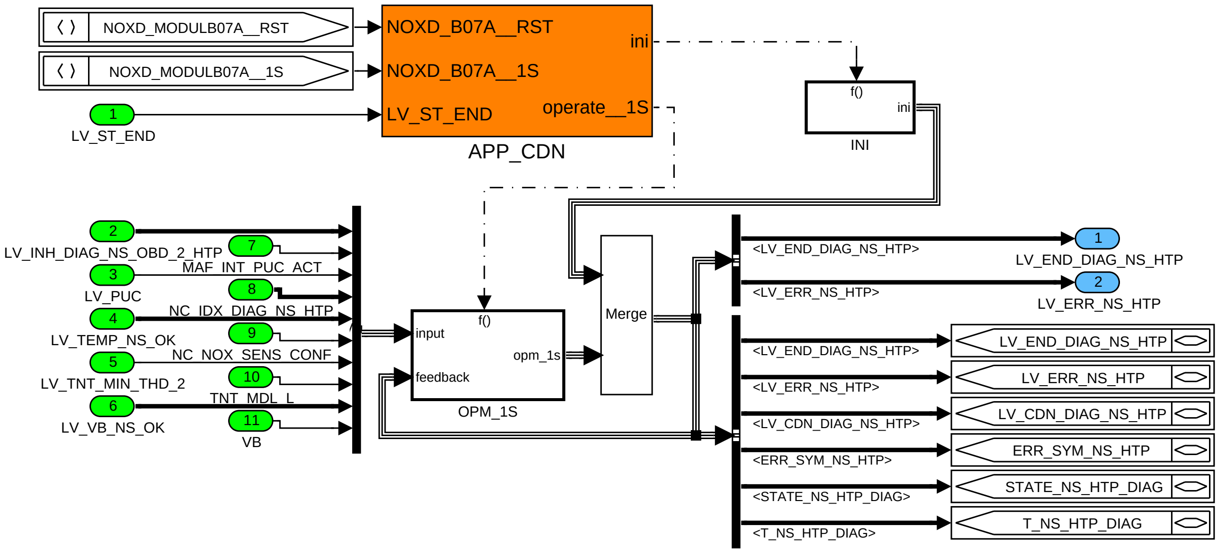


Figure 59 NOXD_MODULB07A/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 15−Dec−2004

Figure 60 NOXD_MODULB07A

52.12.1.1 Initialization at reset and deactivation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >)

Figure 61 NOXD_MODULB07A/ INI/ CLC

52.12.1.2 Calculation for each NOx sensor

The calculation have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

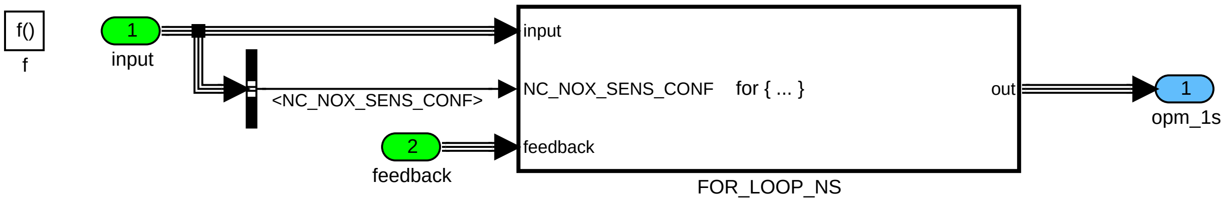


Figure 62 NOXD_MODULB07A/ OPM_1S

Main calculation

If the diagnosis is inhibited, then all values will be set to zero, else the diagnosis will be calculated.

LV_ERR_NS_HTP will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_HTP will be get by ACTION_ERRM_GetLvEndDiag.

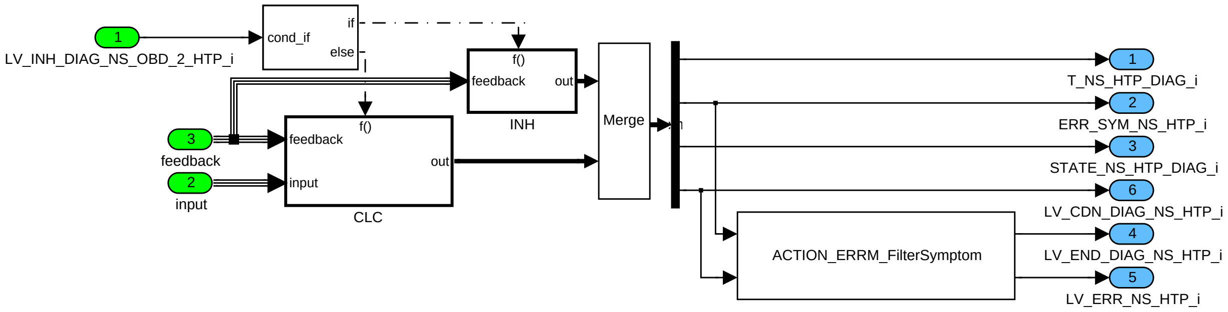


Figure 63 NOXD_MODULB07A/ OPM_1S/ FOR_LOOP_NS/ MAIN

Diagnosis inhibited

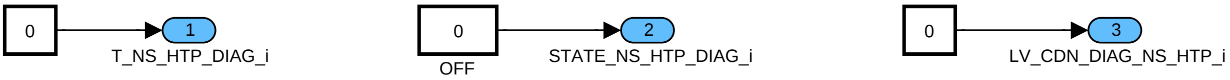


Figure 64 NOXD_MODULB07A/ OPM_1S/ FOR_LOOP_NS/ MAIN/ INH/ CLC

State machine

The default case is case 0 (OFF).

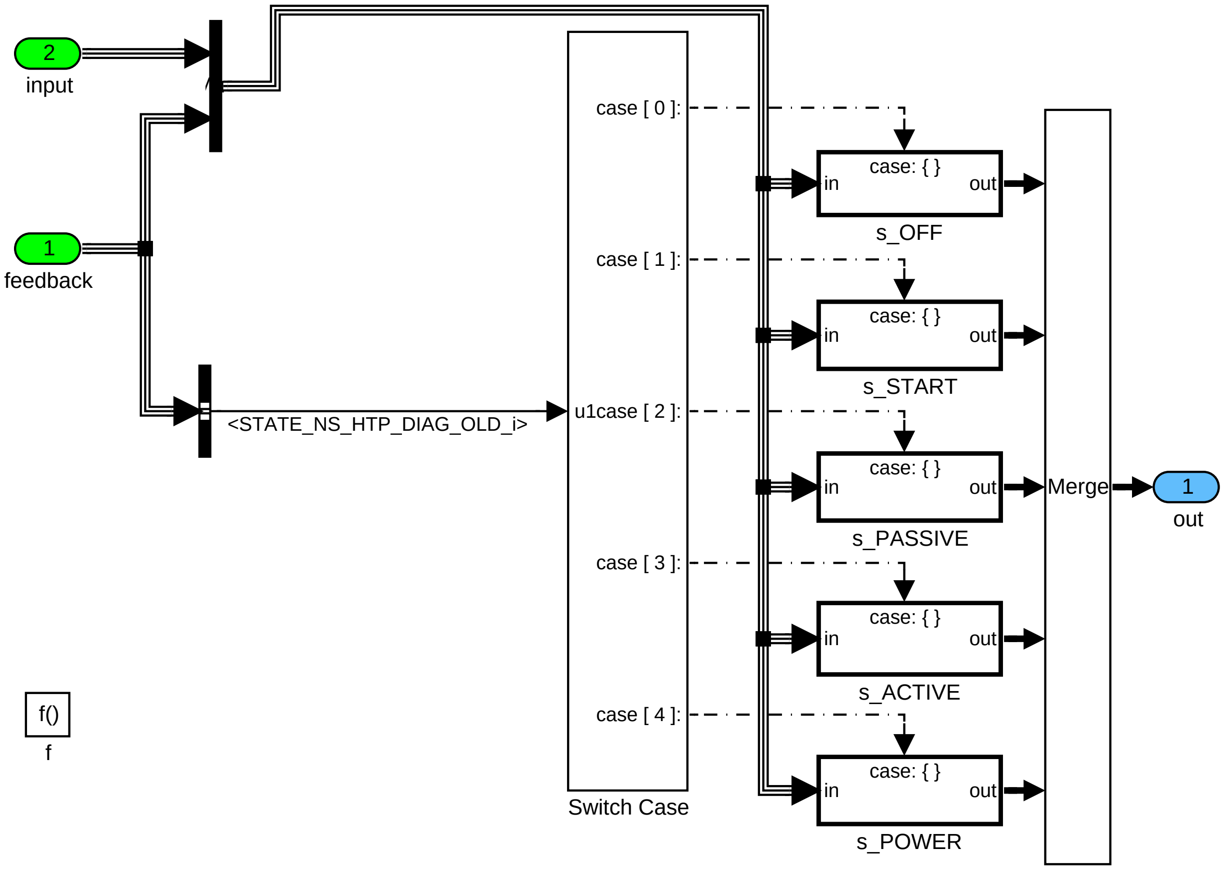


Figure 65 NOXD_MODULB07A/ OPM_1S/ FOR_LOOP_NS/ MAIN/ CLC

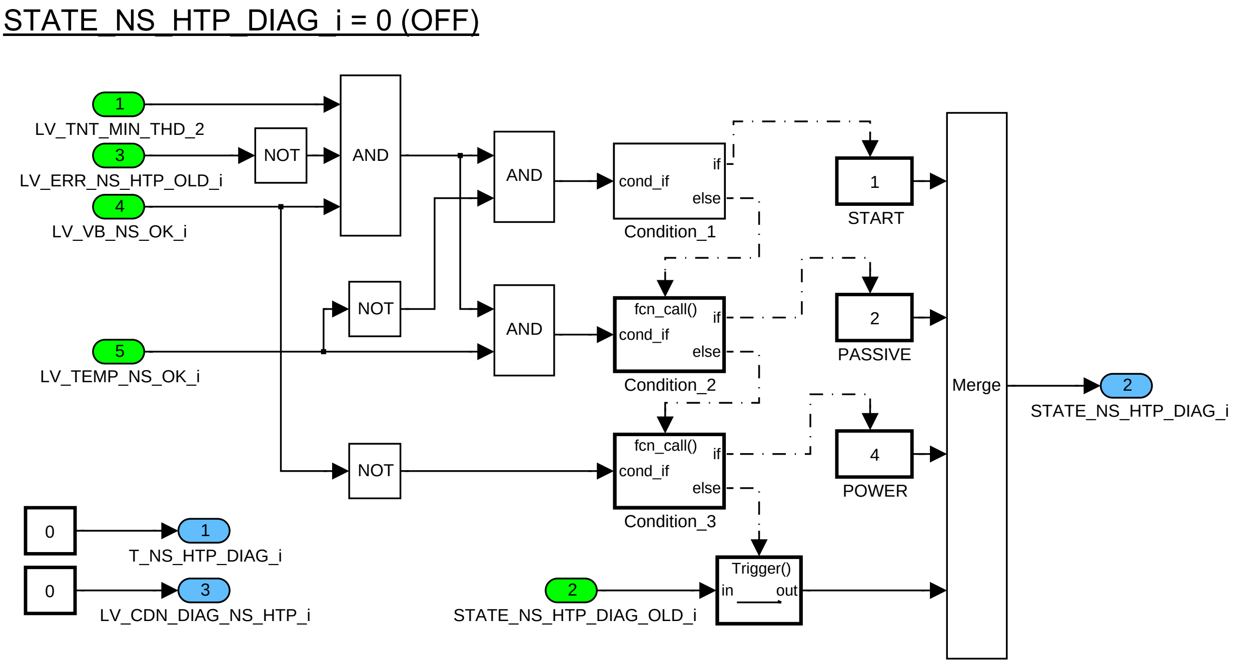


Figure 66 NOXD_MODULB07A/ OPM_1S/ FOR_LOOP_NS/ MAIN/ CLC/ s_OFF/ CLC

STATE_NS_HTP_DIAG_i = 1 (START)

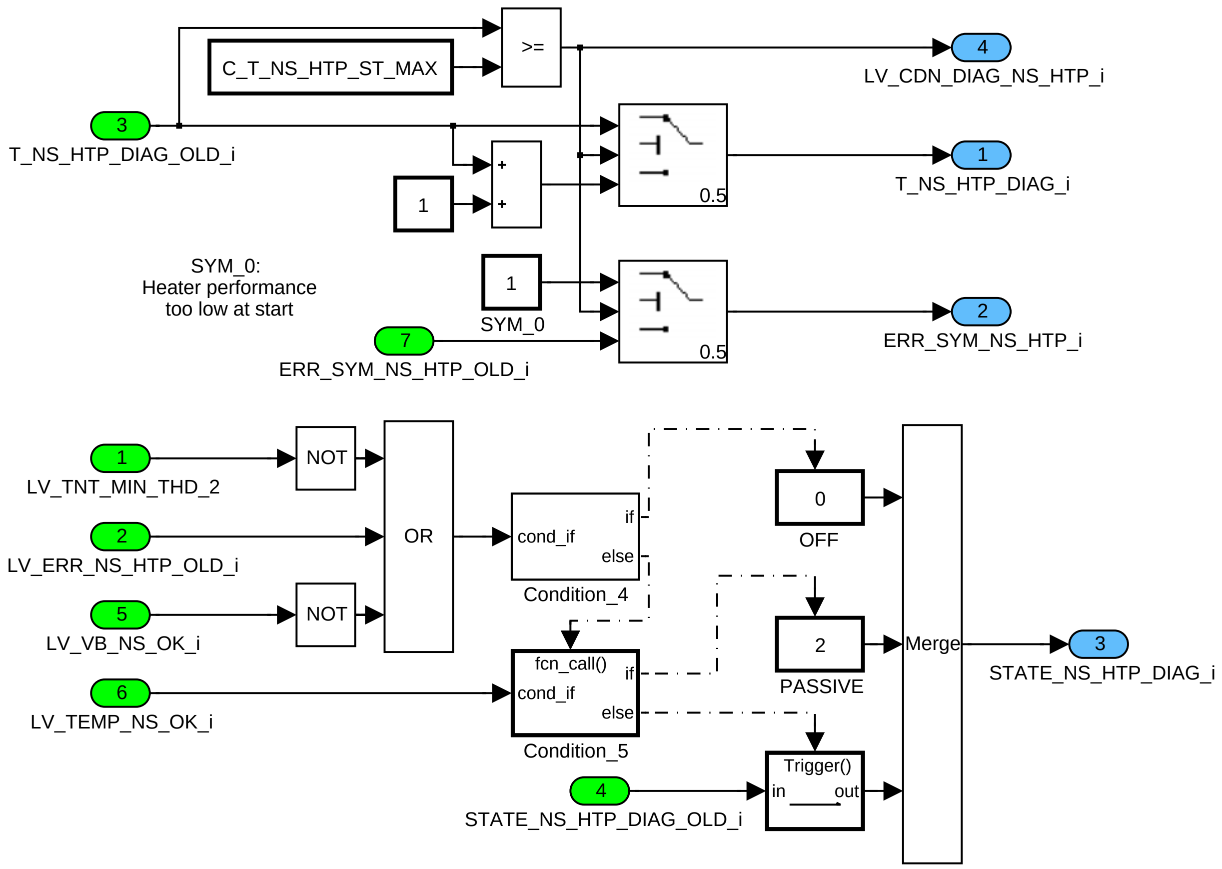


Figure 67 NOXD_MODULB07A/ OPM_1S/ FOR_LOOP_NS/ MAIN/ CLC/ s_START/ CLC

STATE_NS_HTP_DIAG_i = 2 (PASSIVE)

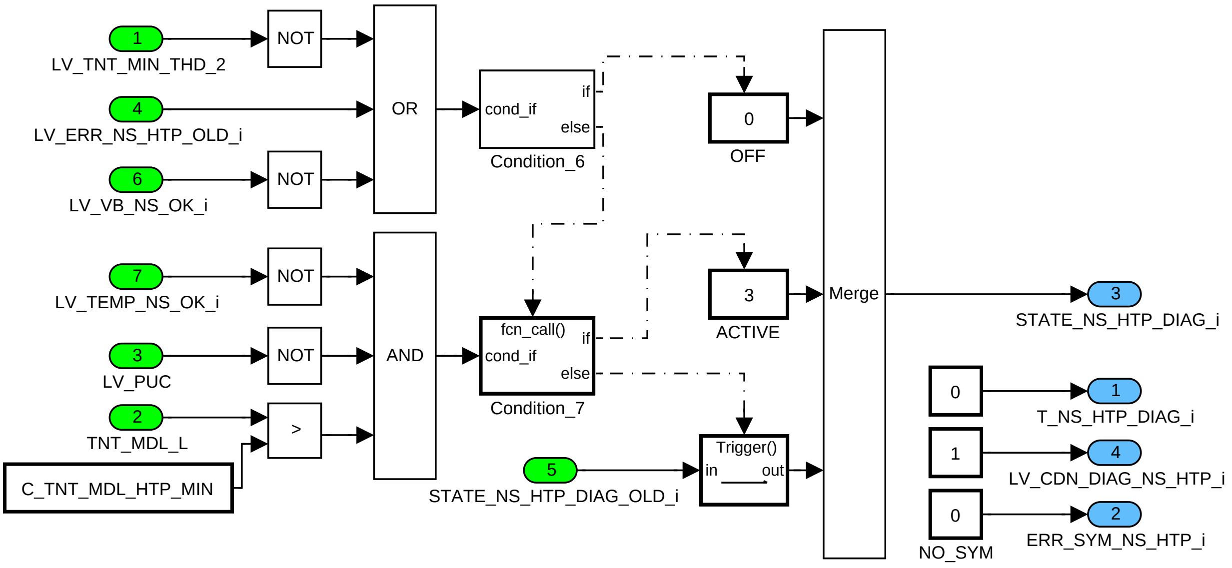


Figure 68 NOXD_MODULB07A/ OPM_1S/ FOR_LOOP_NS/ MAIN/ CLC/ s_PASSIVE/ CLC

STATE_NS_HTP_DIAG_i = 3 (ACTIVE)

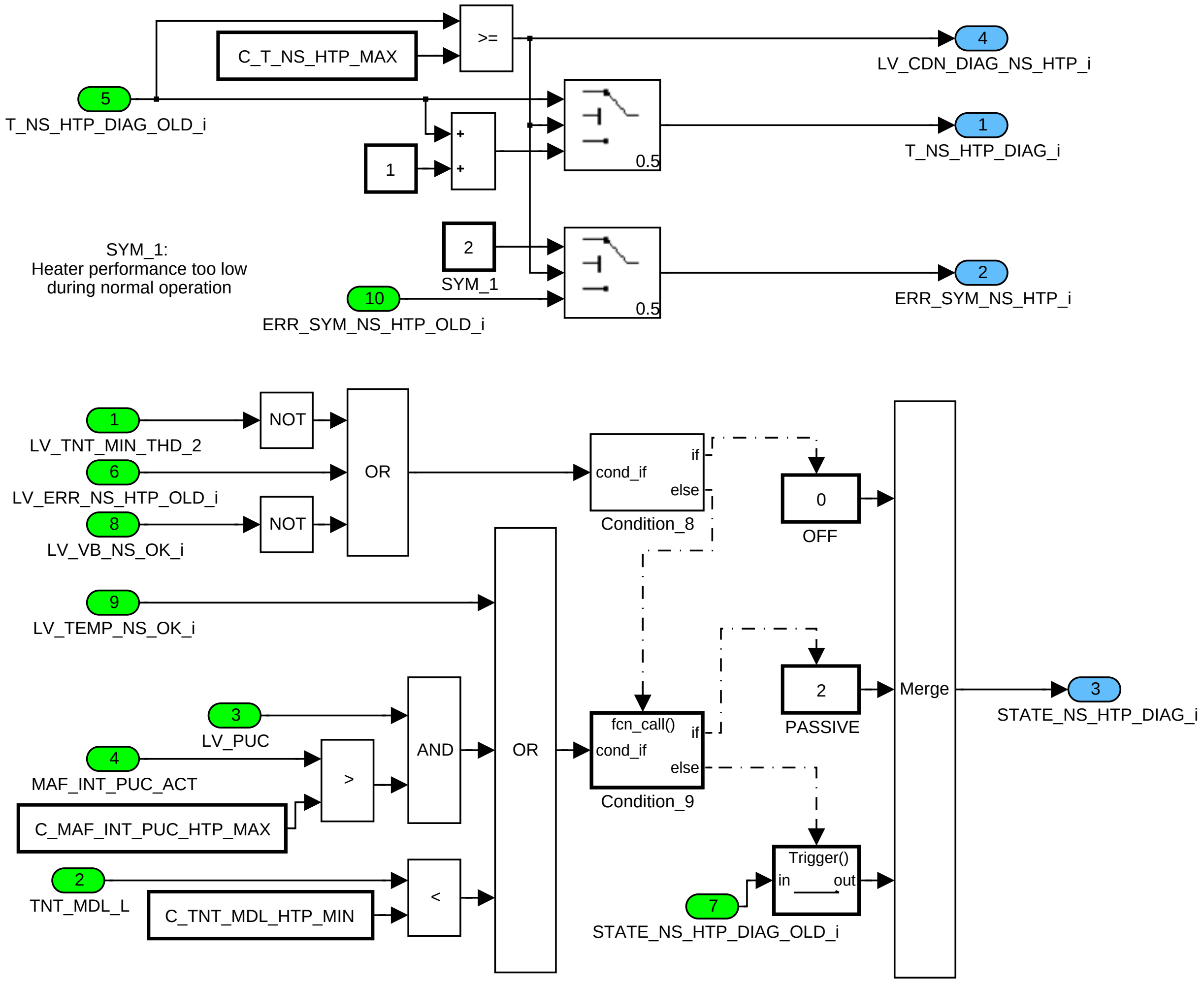


Figure 69 NOXD_MODULB07A/ OPM_1S/ FOR_LOOP_NS/ MAIN/ CLC/ s_ACTIVE/ CLC

STATE_NS_HTP_DIAG_i = 4 (POWER)

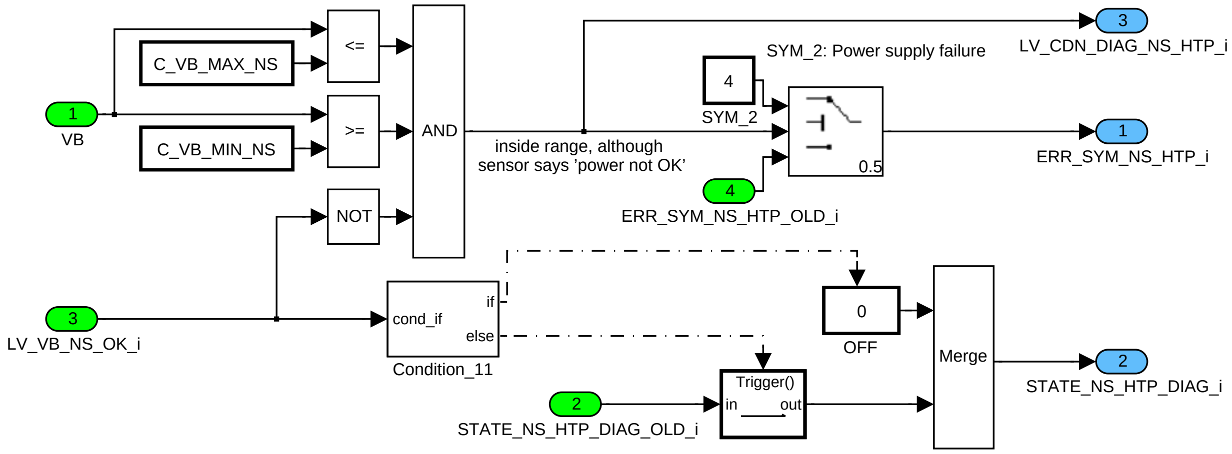


Figure 70 NOXD_MODULB07A/ OPM_1S/ FOR_LOOP_NS/ MAIN/ CLC/ s_POWER/ CLC

52.13 NOx sensor OBDII diag Signal availability

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_AVL[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor availability diagnosis					
LV_ERR_NS_AVL[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of NOx sensor availability diagnosis					
ERR_SYM_NS_AVL[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor availability diagnosis					
FRC_NS_AVL[NC_NOX_SENS_CONF]	V	0...FFFFH	0...6.5535E+4	1	-
Frequency counter for NOx signal available					
LV_CDN_DIAG_NS_AVL[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of NOx sensor availability diagnosis					
STATE_NS_DIAG_AVL[NC_NOX_SENS_CONF]	V	0H 1H 2H 3H	INIT PASSIVE START ACTIVE	1	-
State of NOx sensor availability diagnosis					
TCC_NS_AVL[NC_NOX_SENS_CONF]	V	0...FFFFH	0...6.5535E+4	1	-
Maximum test cycle counter value; used for detection of end of diagnosis					

Input data:

LV_ST_END	LV_HOM_AFS_ACT	LV_HOM_AFS_REQ	LV_INH_DIAG_NS_OBD_2_AVL[NC_NOX_SENS_CONF]
LV_INH_NT_RGN_REQ	LV_MIS_STATE_A	LV_MIS_STATE_B1	LV_MIS_STATE_B4
LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NOX_NS_DIAG_VLD[NC_NOX_SENS_CONF]	LV_PU	LV_SCC[NC_CBK_EX_NR]
LV_S_REQ_EGR	LV_TEMP_MIN_THD_CAN[NC_NOX_SENS_CONF]	LV_TEMP_NS_OK[NC_NOX_SENS_CONF]	LV_T_NS_VLD_TMP[NC_NOX_SENS_CONF]
LV_VB_NS_OK[NC_NOX_SENS_CONF]	NC_CBK_EX_NR	NC_IDX_DIAG_NS_AVL[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
STATE_NOX	T_NOX_NS_HLD[NC_NOX_SENS_CONF]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_AVL	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor availability diagnosis					
C_ABC_INC_NS_AVL	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor availability diagnosis					
C_ABC_MAX_NS_AVL	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx sensor availability diagnosis					
C_FRC_NS_AVL_MAX	1	0...FFFFH	0...6.5535E+4	1	-
Maximum number of cycles with an invalid NOx signal during a NOx signal availability evaluation					
C_TCC_NS_AVL_MAX	1	0...FFFFH	0...6.5535E+4	1	-
Number of test cycles for a NOx signal availability evaluation					
C_TCC_NS_AVL_ST_MAX	1	0...FFFFH	0...6.5535E+4	1	-
Maximum number of test cycles after start after which the NOx signal must be available					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_NOX_NS_ST_DIAG_AVL	1	0...FFFFH	0...655.35	0.01	s
Minimum time of full NOx-Sensor readiness to stop the NOx-Sensor start up diagnosis					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

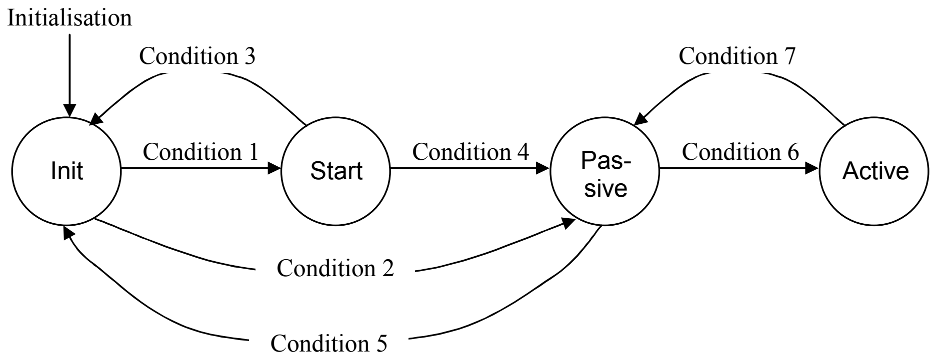
52.13.1 General information

This diagnosis observes the availability of NOx signal after the start and during the “LOAD” phases of NOx catalyst. This function shall detect when the NOx sensor is not able to transmit valid signals because of controller problems such as oscillations.

In normal operation mode the diagnosis is in state ACTIVE if suitable lean conditions are reached, otherwise the state PASSIVE is set. Inside ACTIVE the signal LV_NOX_NS_DIAG_VLD_i will be observed by a statistical method.

At the start-up the diagnosis observes the time which the sensor needs in order to reach the full readiness. This is necessary in order to detect NOx sensors with instable control loops. It will be done inside the state START.

Signal flow diagram:



Application Condition

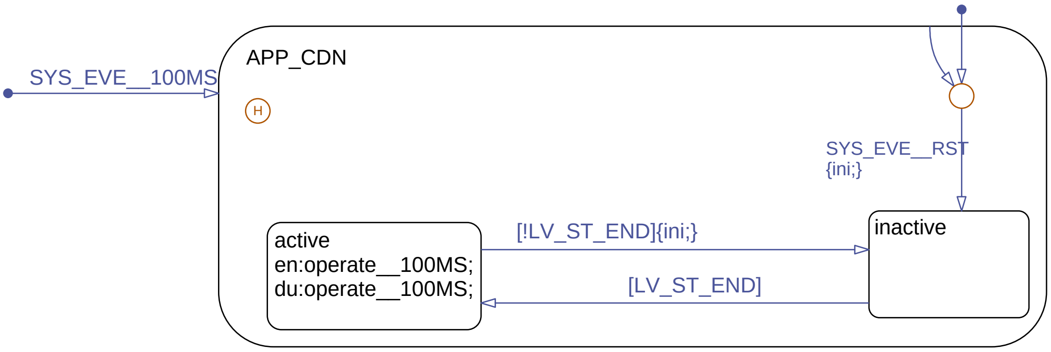
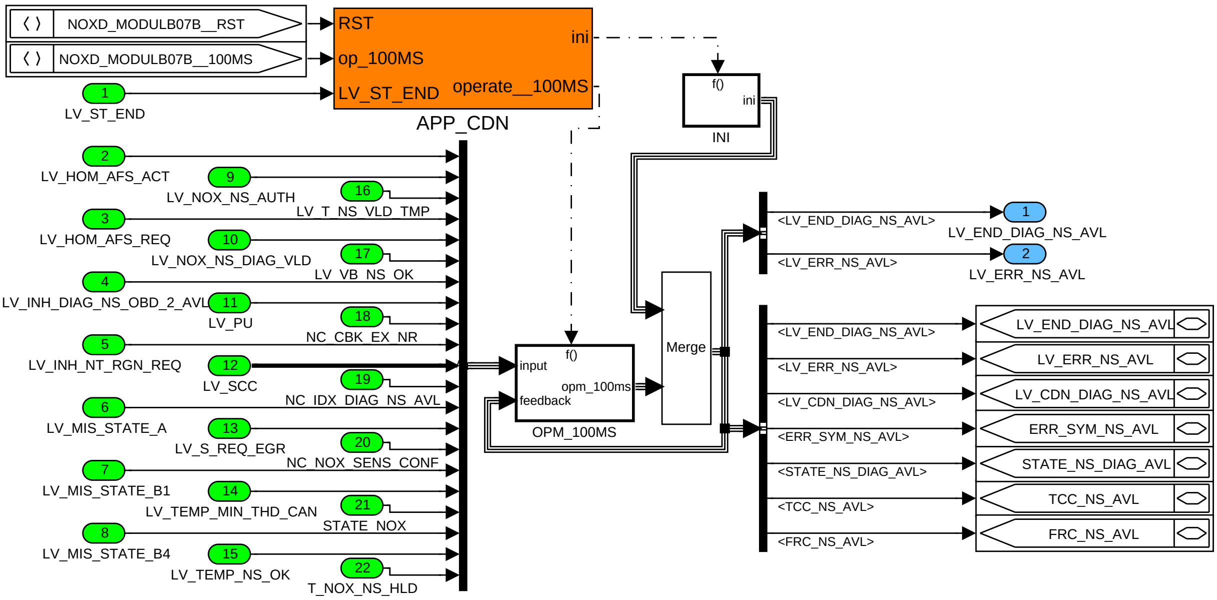


Figure 71 NOXD_MODULB07B/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 10−Mar−2005

Figure 72 NOXD_MODULB07B

52.13.1.1 Initialization at reset and deactivation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >)

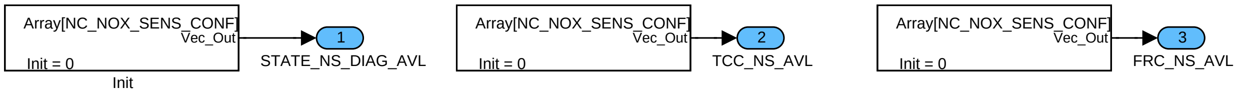


Figure 73 NOXD_MODULB07B/ INI/ CLC

52.13.1.2 Loop control

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

The calculation inside NO_LOOP have to be done only once.

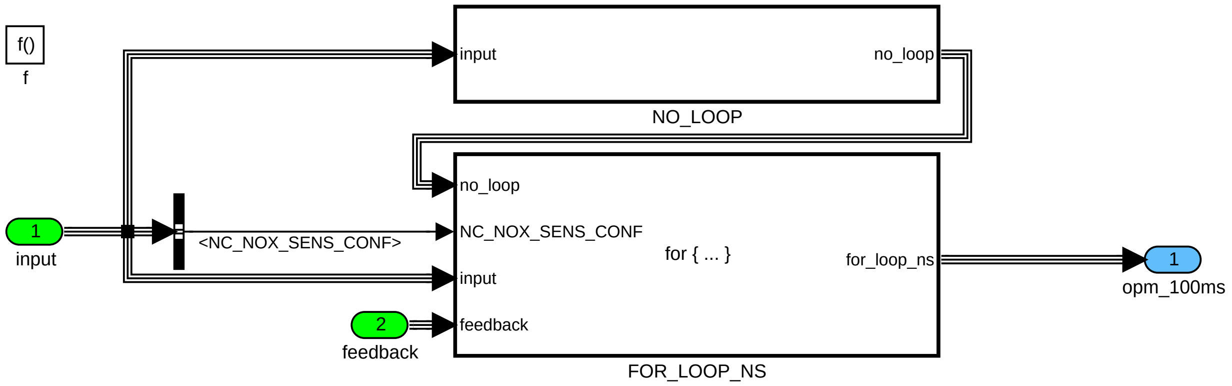


Figure 74 NOXD_MODULB07B/ OPM_100MS

Bank independent calculation

This condition forces the state machine for each NOx sensor to INIT or PASSIVE and prevent the transition to state START or ACTIVE.

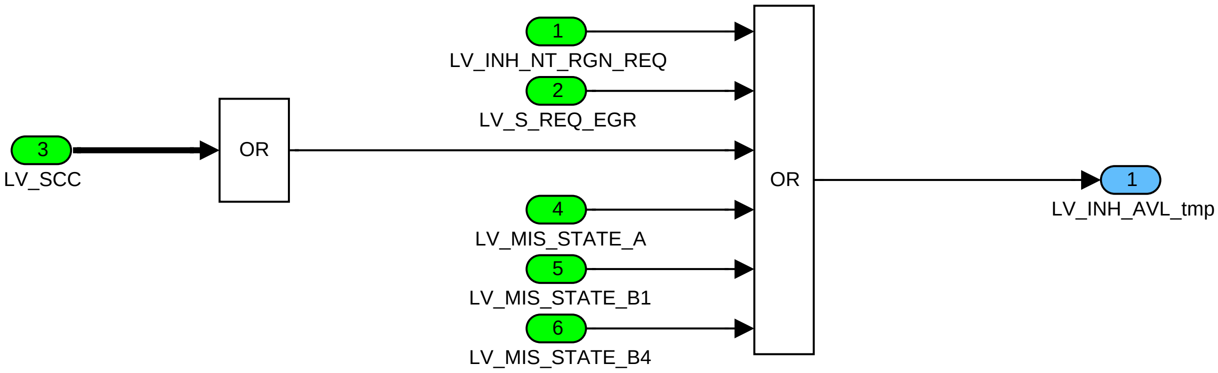


Figure 75 NOXD_MODULB07B/ OPM_100MS/ NO_LOOP/ CLC

Calculation for each NOx sensor

If the diagnosis for a bank is inhibited, then all values will be set to zero, else the diagnosis will be calculated.

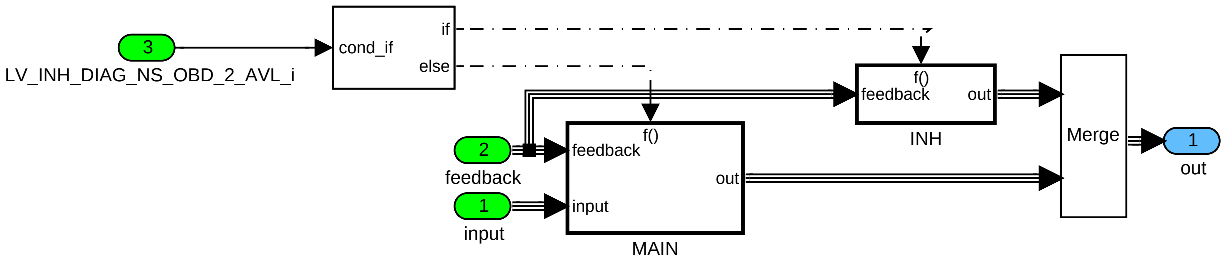


Figure 76 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC

Diagnosis inhibited

Figure 77 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ INH/ CLC

State machine

The default case is is case 0 (INIT).

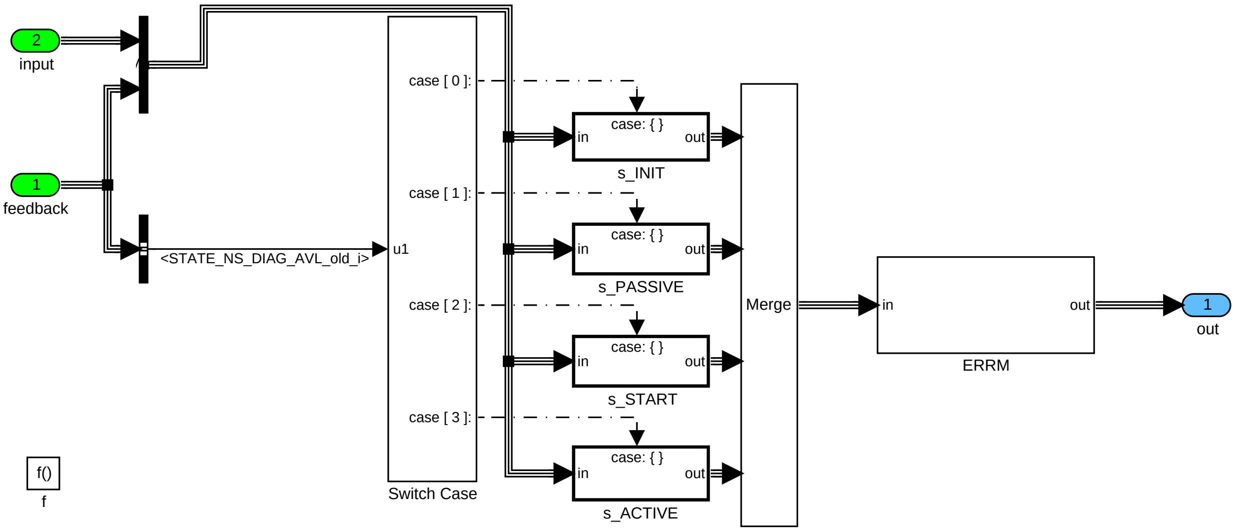


Figure 78 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN

STATE_NS_DIAG_AVL_i = 0 (INIT)

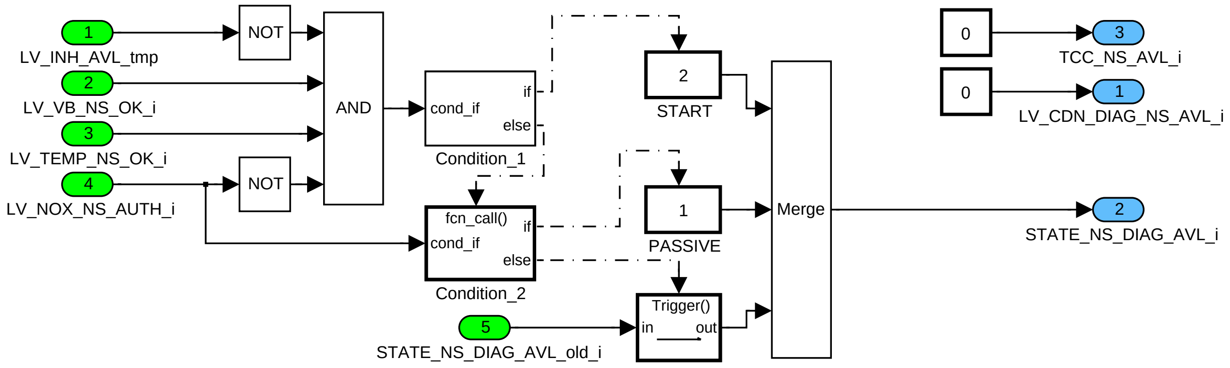


Figure 79 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_INIT/ CLC

STATE_NS_DIAG_AVL_i = 1 (PASSIVE)

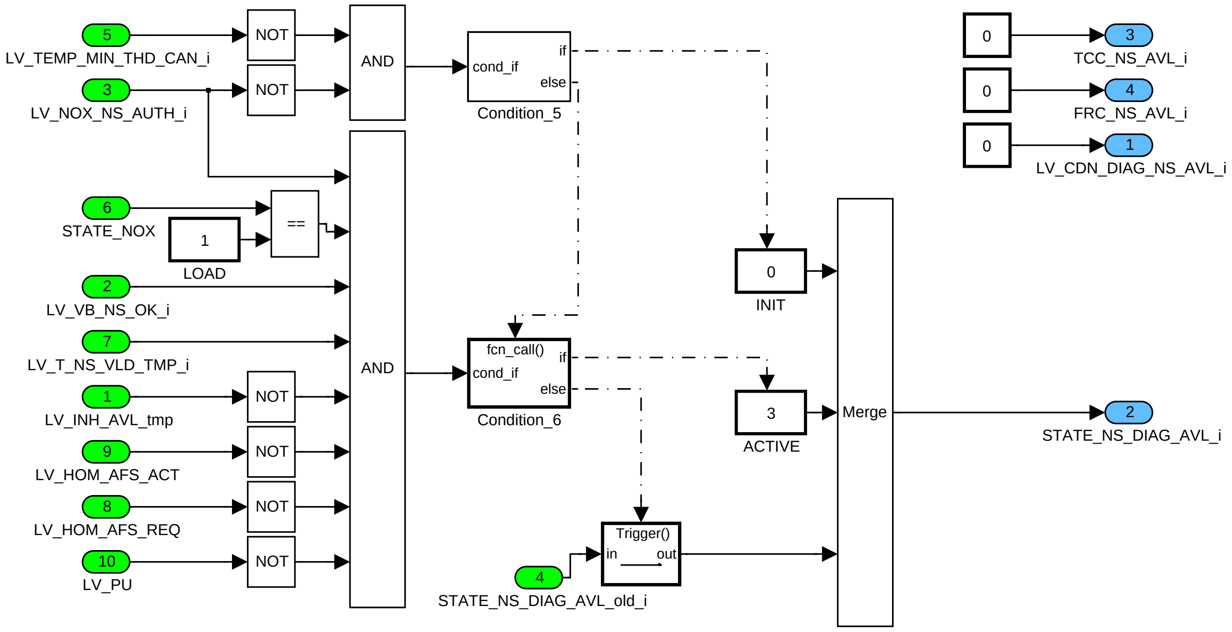


Figure 80 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_PASSIVE/ CLC

STATE_NS_DIAG_AVL_i = 2 (START)

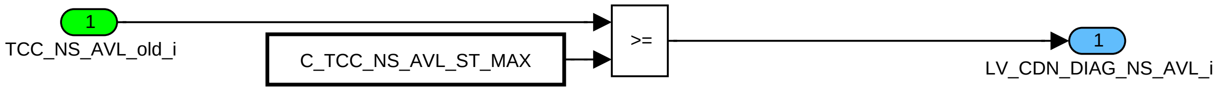


Figure 81 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_START/ CLC1

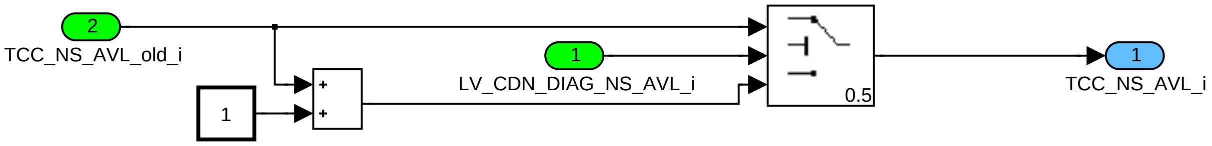


Figure 82 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_START/ CLC2

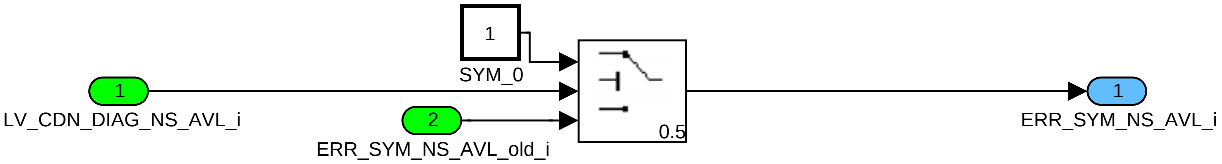


Figure 83 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_START/ CLC3

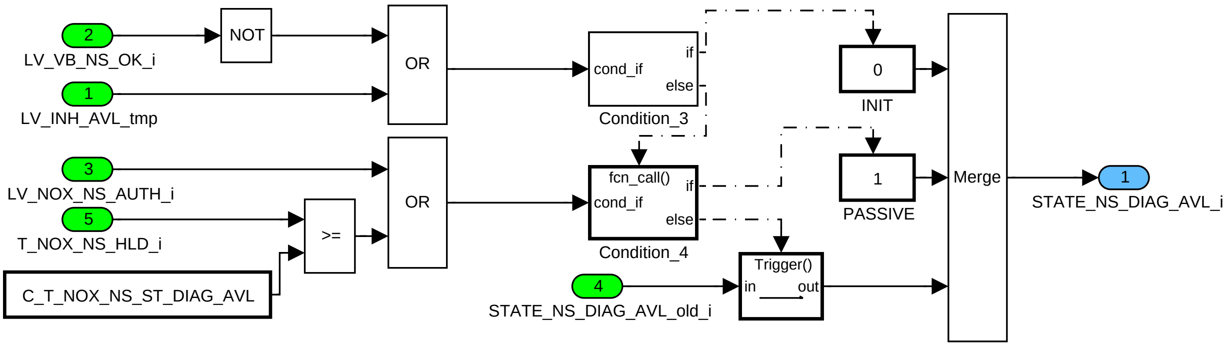


Figure 84 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_START/ CLC4

STATE_NS_DIAG_AVL_i = 3 (ACTIVE)

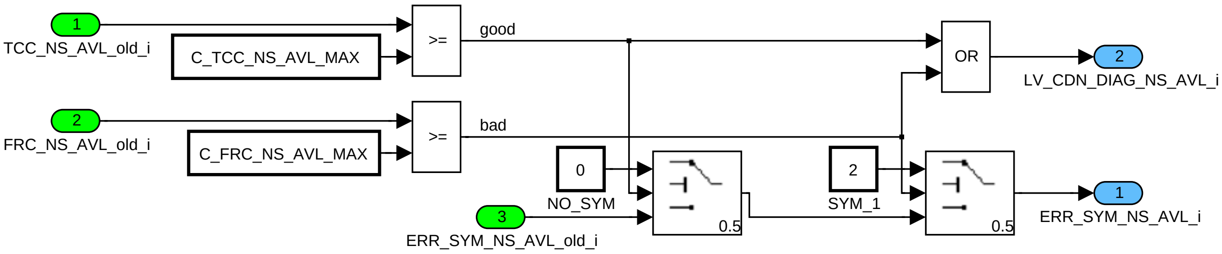


Figure 85 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_ACTIVE/ CLC1

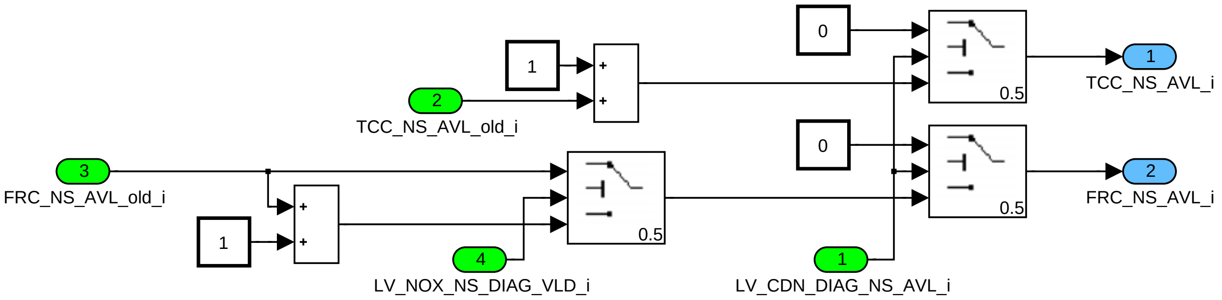


Figure 86 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_ACTIVE/ CLC2

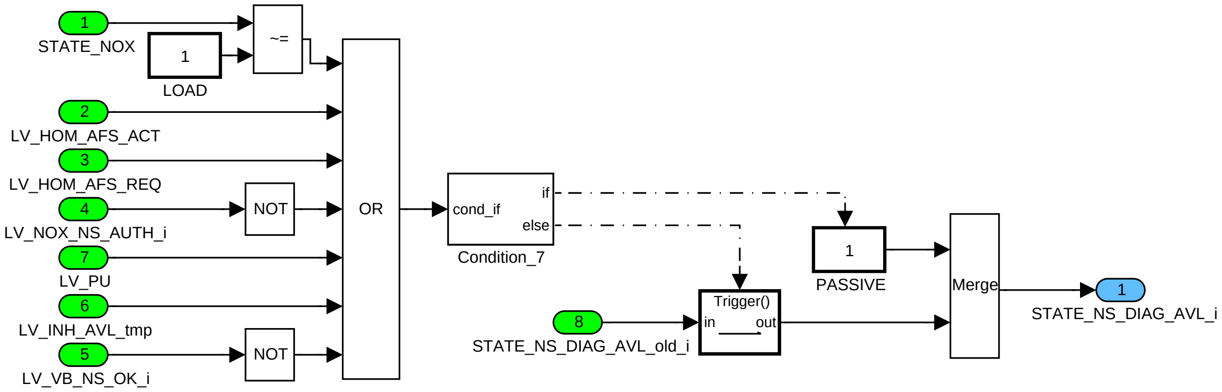


Figure 87 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_ACTIVE/ CLC3

Error management

LV_ERR_NS_AVL will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_AVL will be get by ACTION_ERRM_GetLvEndDiag.

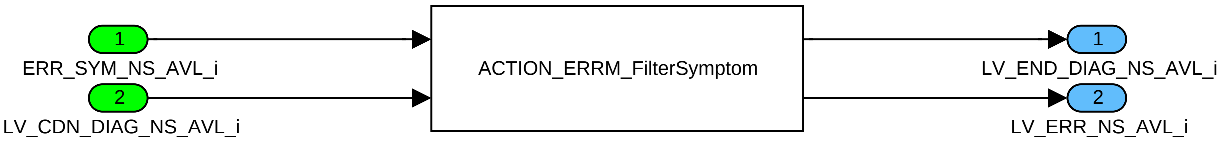


Figure 88 NOXD_MODULB07B/ OPM_100MS/ FOR_LOOP_NS/ CLC/ MAIN/ ERRM/ CLC

52.14 NOx sensor OBDII diag Signal plausibility up/down

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor plausibility diagnosis between upstream and downstream WRAF signal					
LV_ERR_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of plausibility diagnosis between upstream and downstream WRAF signal					
ERR_SYM_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the plausibility diagnosis between upstream and downstream WRAF signal					
MAF_INT_LAMB_LS_UP[NC_CBK_EX_NR]	V	0...FFFFH	0...1.82042E+3	0.02777778	g
Integral of air mass flow for plausibility diagnosis between upstream and downstream WRAF signal					
LV_CDN_DIAG_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of plausibility diagnosis between upstream and downstream WRAF signal					
LV_LAMB_NOT_PLAUS_SNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Lambda signals are not plausible					
LV_FAC_RANGE_SNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Signal FAC_LSL_GAIN_AD is inside the desired range for diagnosis					
LV_LAMB_VLD_TMP	V	0...1H	0...1	1	-
Bank independent values are valid for diagnosis					
LV_LAMB_RANGE_SNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Lambda signals are inside the desired range for diagnosis					
TCC_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]	V	0...FFH	0...255	1	-
Test cycle counter for plausibility check of WRAF signals					

Input data:

FAC_LSL_GAIN_AD[NC_CBK_EX_NR]	LAMB_DELTA_I_LAM_ADJ[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]	LAMB_NS_DIAG[NC_NOX_SENS_CONF]
LV_INH_DIAG_NS_OBD_2_PLAUS[NC_NOX_SENS_CONF]	LV_LAMB_NS_DIAG_VLD[NC_NOX_SENS_CONF]	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NS_LDC[NC_NOX_SENS_CONF]
LV_PU	LV_ST_END	MAF	MAF_KGH
N_32	NC_CBK_EX_NR	NC_IDX_DIAG_NS_LSL_UP_DOWN[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
STATE_CMB_CTL	T_LAMB_NS_HLD[NC_NOX_SENS_CONF]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_LSL_UP_DOWN	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor plausibility diagnosis between WRAF signals					
C_ABC_INC_NS_LSL_UP_DOWN	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor plausibility diagnosis between WRAF signals					
C_ABC_MAX_NS_LSL_UP_DOWN	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx sensor plausibility diagnosis between WRAF signals					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_LSL_GAIN_AD_MAX	1	0...FFFFH	0...1.99996948	3.05176E-5	-
Maximum WRAF sensor gain adaptation factor for activation of plausibility check					
C_FAC_LSL_GAIN_AD_MIN	1	0...FFFFH	0...1.99996948	3.05176E-5	-
Minimum WRAF sensor gain adaptation factor for activation of plausibility check					
C_LAMB_DIF_MAX_LSL_UP_DOWN	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Activation threshould for lambda difference between both banks					
C_LAMB_PLAUS_LSL_UP_DOWN_MAX	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Maximum lambda threshold for deactivation of plausibility check					
C_LAMB_PLAUS_LSL_UP_DOWN_MIN	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Minimum lambda threshold for deactivation of plausibility check					
C_MAF_INT_LAMB_PLAUS	1	0...FFFFH	0...1.82042E+3	0.02777778	g
MAF threshold for starting the Lambda plausibility diagnosis of NOx-Sensor					
C_TCC_NS_LSL_UP_DOWN_MAX	1	0...FFH	0...255	1	-
Maximum number of test cycles per steady state phase					
C_T_LAMB_NS_HLD_DIAG_PLAUS	1	0...7FFFH	0...327.67	0.01	s
Time with lambda signal state = invalid, after which a plausability diagnosis is started					
IP_LAMB_LS_UP_PLAUS_GRD_MAX	4	0...7FFFH	0...31.9990234	9.76563E-4	-
LDP_LAMB_LS_UP_IP_LAMB_GRD	4	0...7FFFH	0...31.9990234	9.76563E-4	-
Maximum allowed gradient of LAMB_LS_UP to calculate the MAF integral					
IP_LAMB_PLAUS_LSL_UP_DOWN	4	0...7FFFH	0...31.9990234	9.76563E-4	-
LDP_LAMB_LS_UP_IP_LAMB_UP_DOWN	4	0...7FFFH	0...31.9990234	9.76563E-4	-
Signal tolerance between upstream and downstream WRAF signal					
IP_MAF_MIN_LAMB_LS_UP_PLAUS	6	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
LDP_N_32_IP_MAF_MIN_LAMB_PLAUS	6	0...FFH	0...8.16E+3	32	rpm
Minimum MAF threshold for activation of upstream and downstream WRAF signal plausibility check					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_FAC_MAF_INT_PLAUS	1	0...FFFFH	0...1.02398437	1.5625E-5	s
normalization factor for calculation of integral of MAF_KGH					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

52.14.1 General information

This signal plausibility check compares the upstream oxygen signal (WRAF sensor) with the WRAF signal of NOx-Sensor, which is downstream of NOx catalyst. The result of this check can indicate a strong discrepancy between the two sensors, which can be caused by reference air contamination.

Application Condition

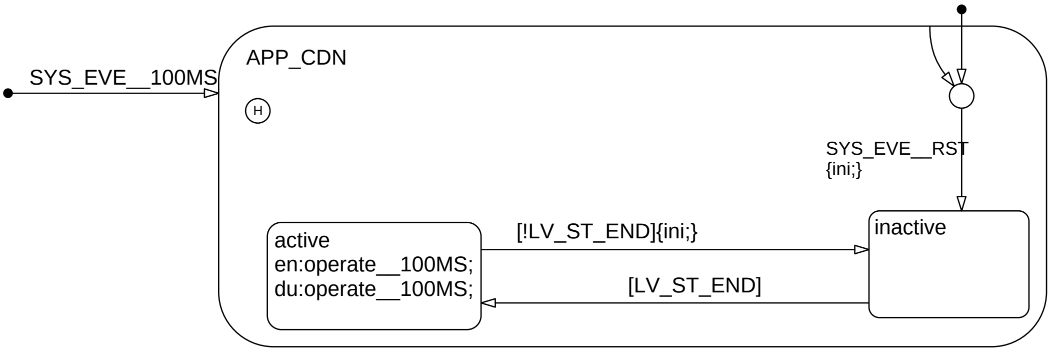
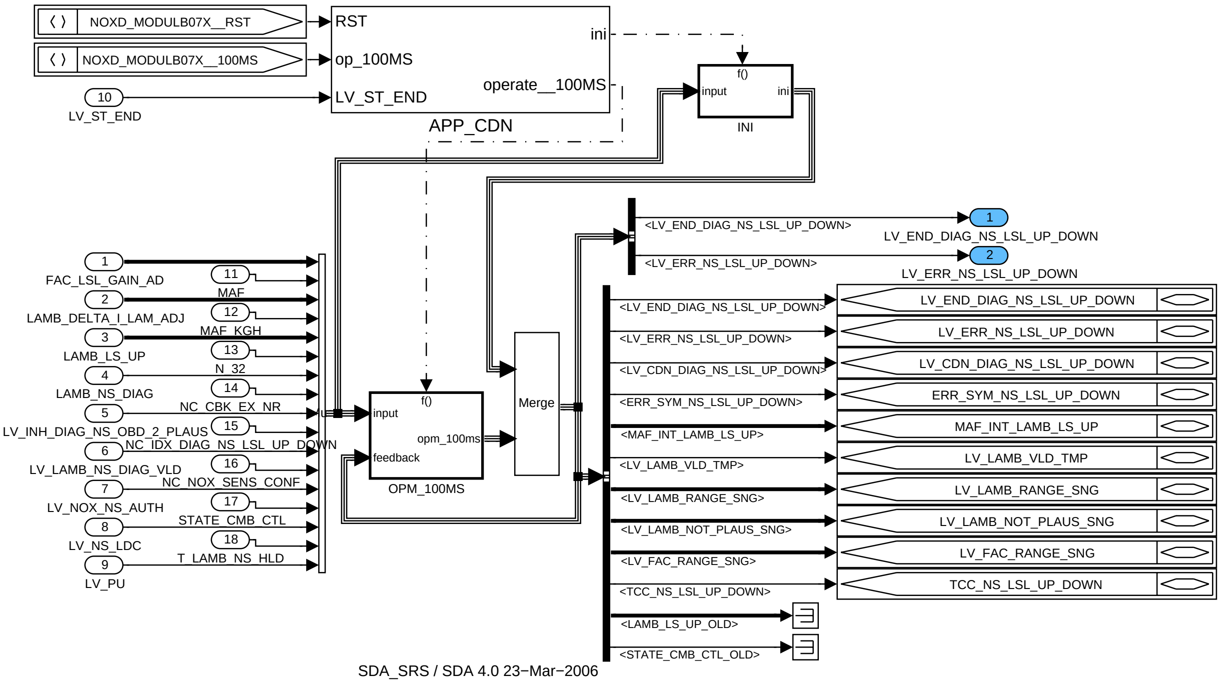


Figure 89 NOXD_MODULB07X/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 23−Mar−2006

Figure 90 NOXD_MODULB07X

52.14.1.1 Initialization at reset and deactivation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >)

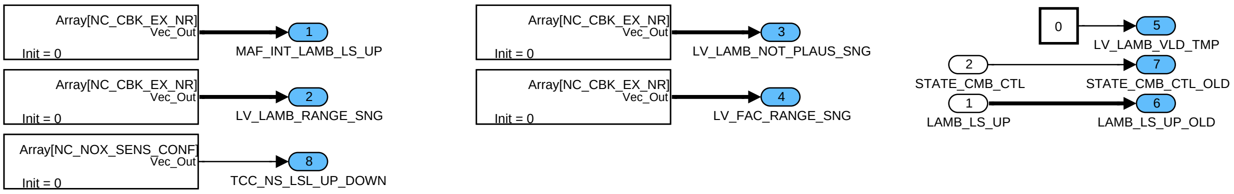


Figure 91 NOXD_MODULB07X/ INI/ CLC

52.14.1.2 Loop control

The calculation inside FOR_LOOP_LS have to be done for each upstream lambda sensor, controlled by configuration data NC_CBK_EX_NR.

If NC_CBK_EX_NR = 1, then calculation for i = 1.

If NC_CBK_EX_NR = 2, then calculation for i = 1 and 2.

The calculation inside both NO_LOOP subsystems have to be done only once.

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

NC_NOX_SENS_CONF cannot be greater than NC_CBK_EX_NR.

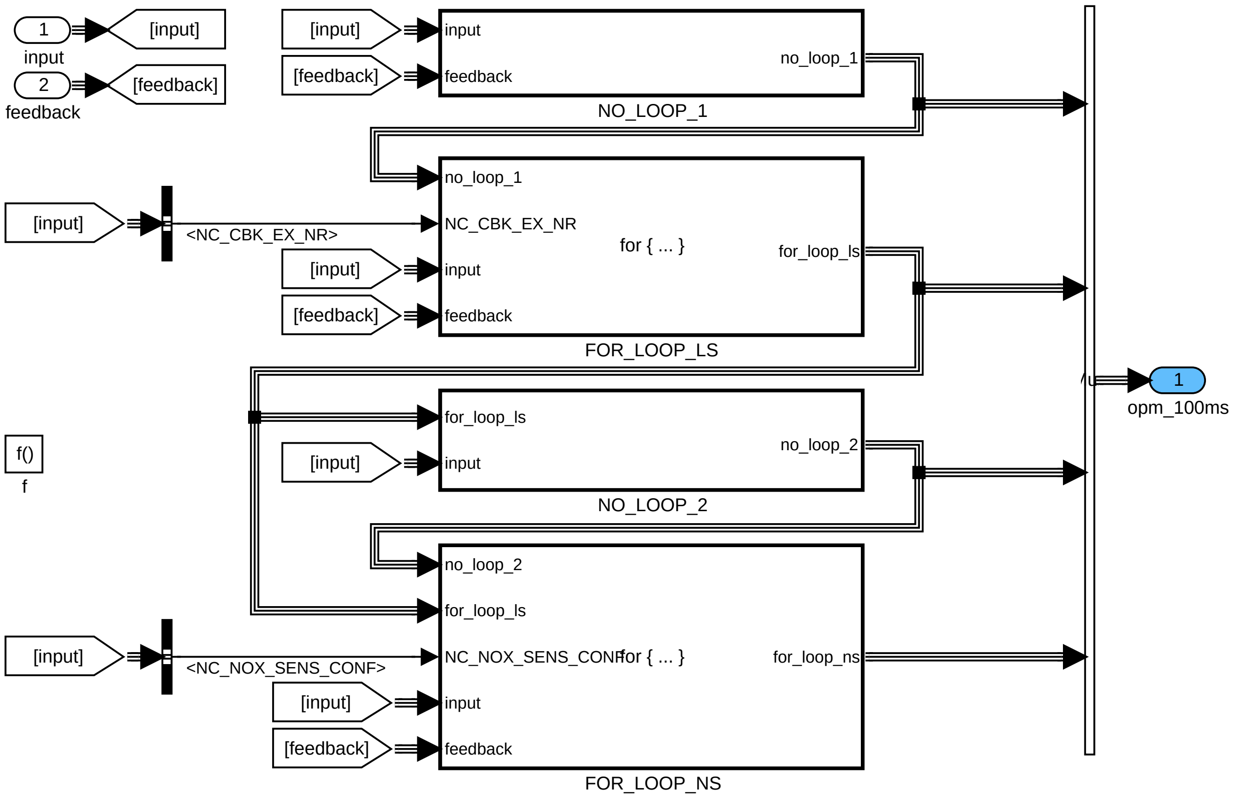


Figure 92 NOXD_MODULB07X/ OPM_100MS

Bank independent calculation (part 1)

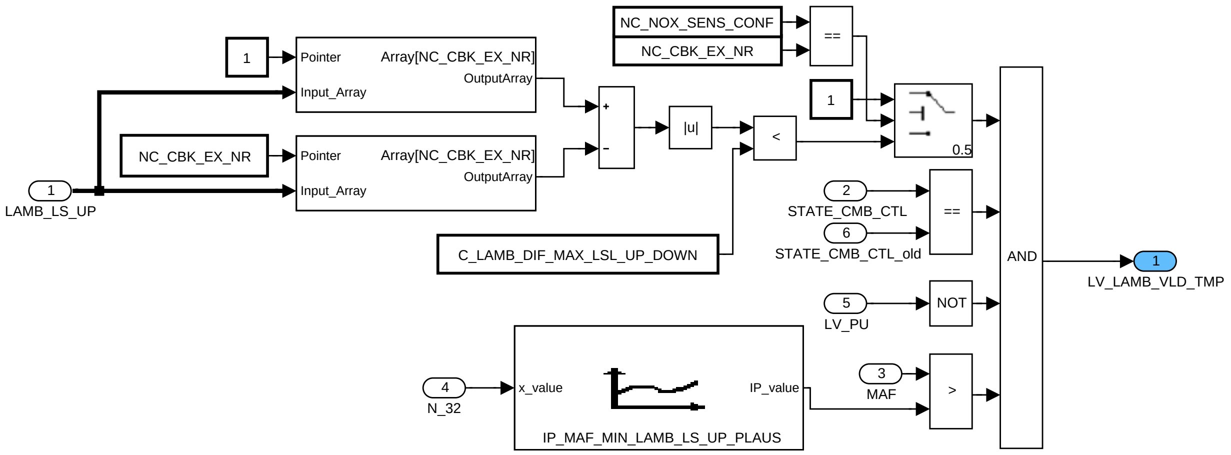


Figure 93 NOXD_MODULB07X/ OPM_100MS/ NO_LOOP_1/ CLC

Calculation for each upstream lambda sensor

Integral of air mass flow

The MAF integral will be calculated if the operating condition are nearly stationary, otherwise the integral will be reset.

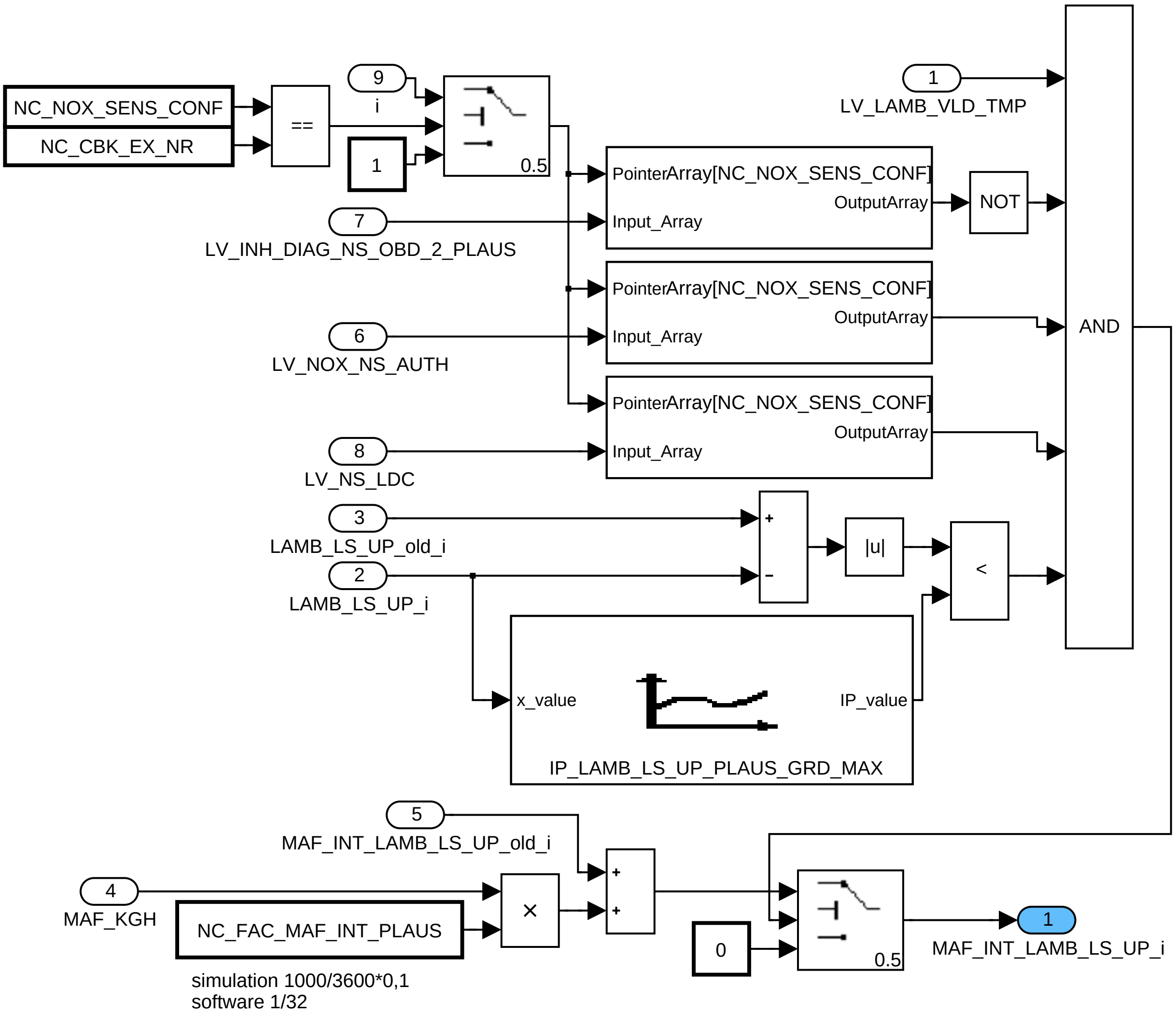


Figure 94 NOXD_MODULB07X/ OPM_100MS/ FOR_LOOP_LS/ CLC1

Lambda range

The diagnosis will be activated if the upstream lambda is in a desired range and the MAF integral shows that also behind the NOx trap should be the same lambda.

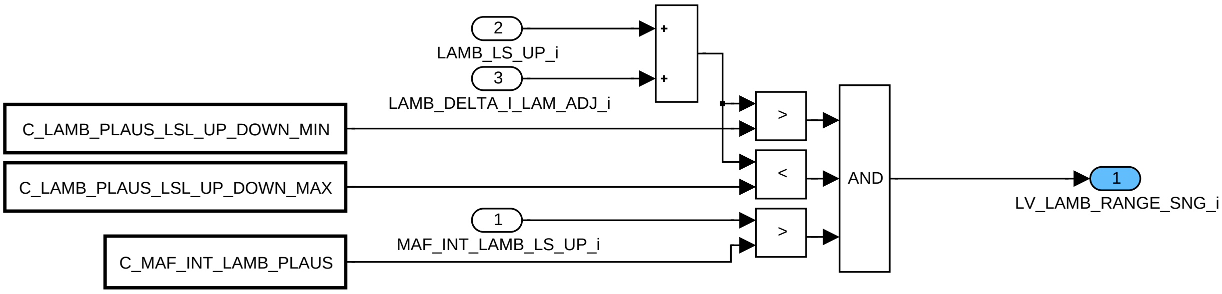


Figure 95 NOXD_MODULB07X/ OPM_100MS/ FOR_LOOP_LS/ CLC2

Lambda comparison

The linear lambda of the NOx sensor will be compared with the upstream lambda to get the diagnosis result.

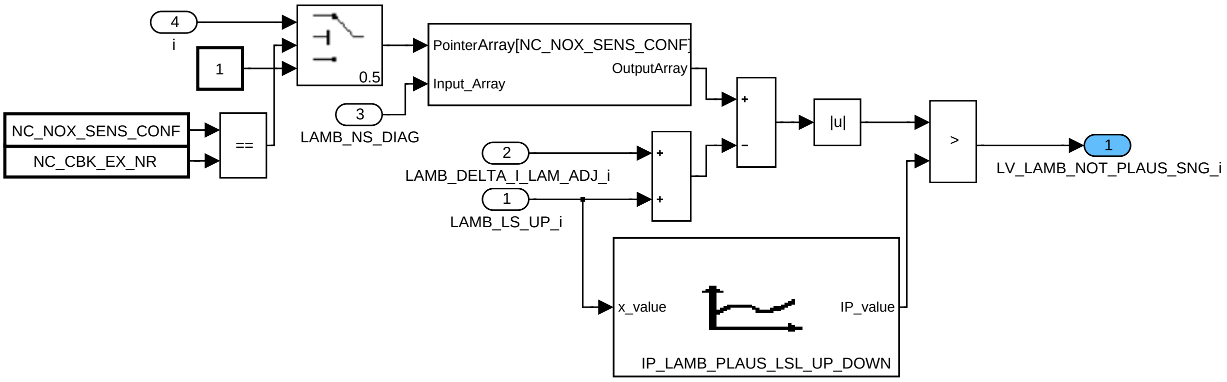


Figure 96 NOXD_MODULB07X/ OPM_100MS/ FOR_LOOP_LS/ CLC3

Lambda controller gain adaptation

The lambda controller gain adaptation have to be in a desired range to avoid a wrong diagnosis result for the NOx sensor during the comparison with the upstream lambda sensor.

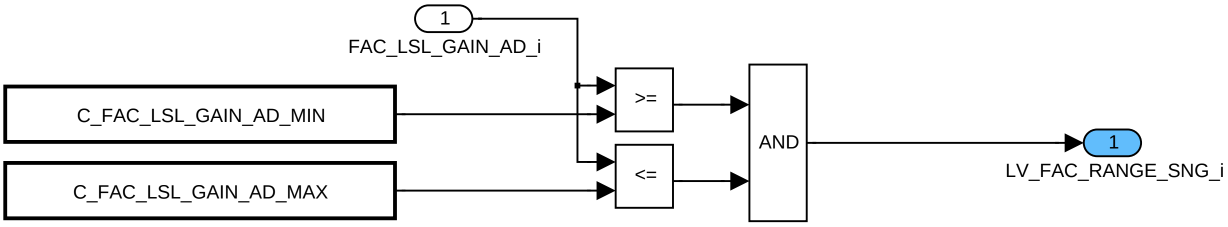


Figure 97 NOXD_MODULB07X/ OPM_100MS/ FOR_LOOP_LS/ CLC4

Bank independent calculation (part 2)

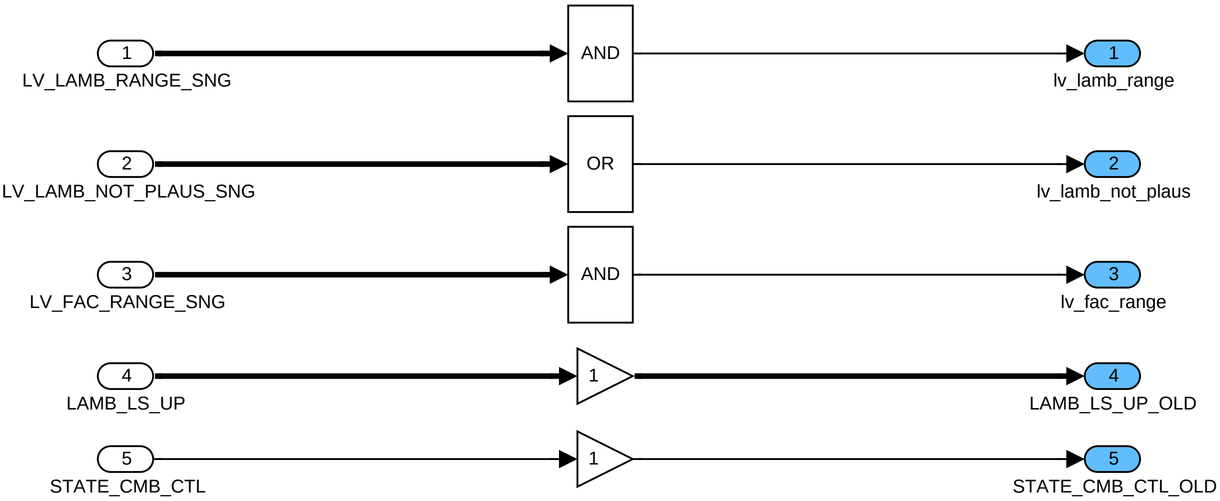


Figure 98 NOXD_MODULB07X/ OPM_100MS/ NO_LOOP_2/ CLC

Calculation for each NOx sensor

Diagnosis evaluation

For each NOx sensor diagnosis result have to be used either the signals of the same bank or resulting signals if a NOX sensor sees the exhaust gas from more than one bank.

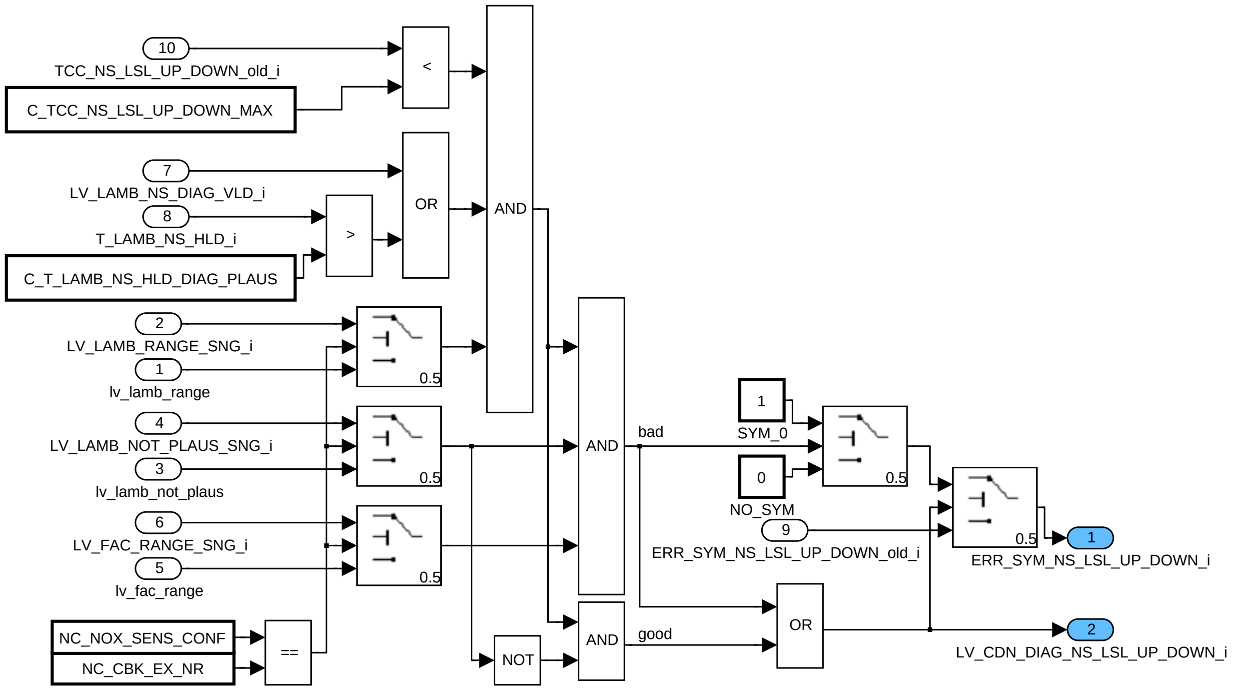


Figure 99 NOXD_MODULB07X/ OPM_100MS/ FOR_LOOP_NS/ CLC1

Test cycle counter

The number of test cycles per steady state phase may be calibrated.

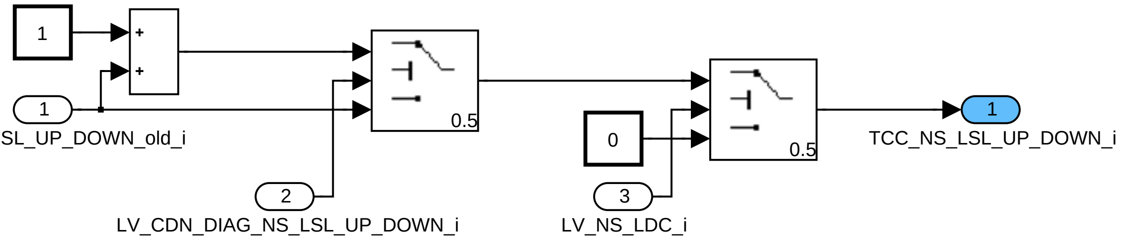


Figure 100 NOXD_MODULB07X/ OPM_100MS/ FOR_LOOP_NS/ CLC2

Error management

LV_ERR_NS_LSL_UP_DOWN will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_LSL_UP_DOWN will be get by ACTION_ERRM_GetLvEndDiag.

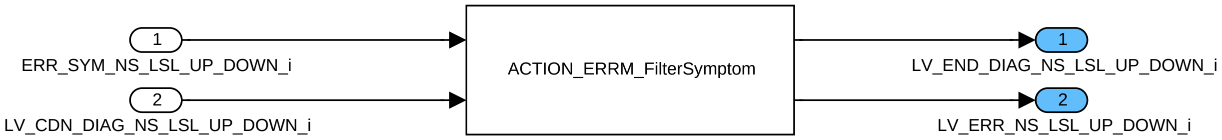


Figure 101 NOXD_MODULB07X/ OPM_100MS/ FOR_LOOP_NS/ CLC3

52.15 NOx sensor OBDII diag Signal plausibility at PUC

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_PUC[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor plausibility diagnosis at PUC					
LV_ERR_NS_PUC[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of NOx sensor plausibility diagnosis at PUC					
ERR_SYM_NS_PUC[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor plausibility diagnosis at PUC					
LV_CDN_DIAG_NS_PUC[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of NOx sensor plausibility diagnosis at PUC					
LV_LAMB_VLD_PUC	V	0...1H	0...1	1	-
All lambda signals are valid for diagnosis at PUC					
LV_LAMB_VLD_PUC_SNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Lambda signal is valid for diagnosis at PUC (bench selective)					
TCC_NS_PUC_DIAG[NC_NOX_SENS_CONF]	V	0...FFH	0...255	1	-
Test cycle counter for plausibility checks at actual PUC phase					

Input data:

LAMB_LS_UP[NC_CBK_EX_NR]	LAMB_NS_DIAG[NC_NOX_SENS_CONF]	LV_LAMB_NS_DIAG_VLD[NC_NOX_SENS_CONF]	LV_INH_DIAG_NS_OBD_2_PLAUS[NC_NOX_SENS_CONF]
LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NOX_NS_DIAG_VLD[NC_NOX_SENS_CONF]	LV_NT_HOM_INI	LV_PUC
LV_VLS_NS_DIAG_VLD[NC_NOX_SENS_CONF]	MAF_INT_PUC_ACT	NC_CBK_EX_NR	NC_IDX_DIAG_NS_PUC[NC_NOX_SENS_CONF]
NC_NOX_SENS_CONF	NOX_NS_DIAG[NC_NOX_SENS_CONF]	NTLD	T_VLS_NS_HLD[NC_NOX_SENS_CONF]
T_LAMB_NS_HLD[NC_NOX_SENS_CONF]	T_NOX_NS_HLD[NC_NOX_SENS_CONF]	VLS_DOWN[NC_CBK_EX_NR]	VLS_NS_DIAG[NC_NOX_SENS_CONF]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_PUC	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor plausibility diagnosis at PUC					
C_ABC_INC_NS_PUC	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor plausibility diagnosis at PUC					
C_ABC_MAX_NS_PUC	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx sensor plausibility diagnosis at PUC					
C_LAMB_NS_DIAG_PUC_MIN	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Minimum Lambda value of NOx-Sensor at PUC					
C_LAMB_PUC_PLAUS_MIN	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Minimum upstream Lambda value to start the NOx-Sensor PUC check					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MAF_INT_PUC_PLAUS_MIN	1	0...FFFFH	0...1.82042E+3	0.02777778	g
MAF threshold for starting the PUC diagnosis					
C_NOX_NS_DIAG_PUC_MAX	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Maximum NOx concentration signal value of NOx-Sensor at PUC					
C_NOX_NS_DIAG_PUC_MIN	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Minimum NOx concentration signal value of NOx-Sensor at PUC					
C_NTLD_NS_PLAUS_PUC	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Maximum NOx loading of NOx catalyst for detecting a NOx signal failure					
C_TCC_NS_PUC_DIAG	1	0...FFH	0...255	1	-
Number of test cycles at a PUC phase					
C_T_LAMB_NS_HLD_DIAG_PUC	1	0...7FFFH	0...327.67	0.01	s
Time with lambda signal state = invalid, after which a PUC diagnosis is started					
C_T_NOX_NS_HLD_DIAG_PUC	1	0...7FFFH	0...327.67	0.01	s
Time with NOx signal state = invalid, after which a PUC diagnosis is started					
C_T_VLS_NS_HLD_DIAG_PUC	1	0...7FFFH	0...327.67	0.01	s
Time with binary O2 signal state = invalid, after which a PUC diagnosis is started					
C_VLS_DOWN_PUC_PLAUS_MAX[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511719	0.00488281	V
Maximum VLS_DOWN signal value to start the NOx-Sensor PUC check					
C_VLS_NS_DIAG_PUC_MAX	1	0...578H	-200...1.2E+3	1	mV
Maximum binary O2 signal voltage of NOx-Sensor at PUC					
LC_NT_HOM_INI_PUC_PLAUS	1	0...1H	0...1	1	-
NOx signal plausibility diagnosis at PUC runs only after a complete NOx catalyst regeneration					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

52.15.1 General information

This signal check tests all NOx sensor measurement signals on theirs plausibility at the special engine operating state ‘pull fuel cut-off’ (PUC). With this check shall be detected a sensor poisoning or signal offset problems.

Application Condition

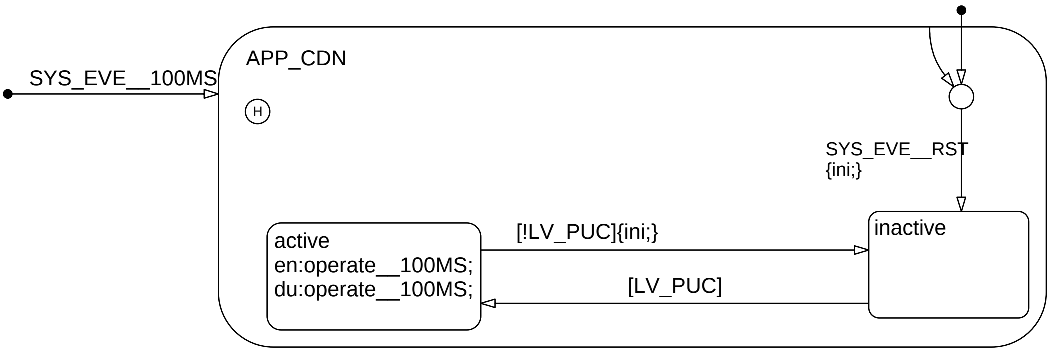
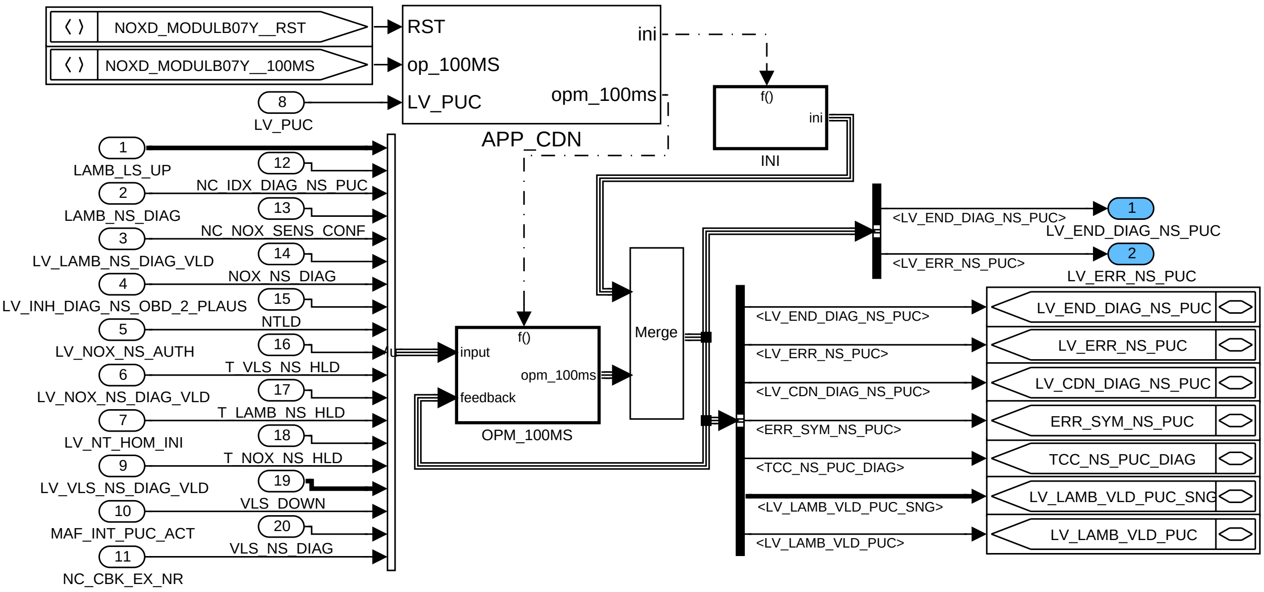


Figure 102 NOXD_MODULB07Y/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 28−Sep−2006

Figure 103 NOXD_MODULB07Y

52.15.1.1 Initialization at reset and deactivation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >)

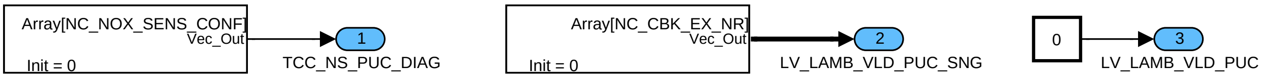


Figure 104 NOXD_MODULB07Y/ INI/ CLC

52.15.1.2 Loop control

The calculation inside FOR_LOOP_LS have to be done for each upstream lambda sensor, controlled by configuration data NC_CBK_EX_NR.

If NC_CBK_EX_NR = 1, then calculation for i = 1.

If NC_CBK_EX_NR = 2, then calculation for i = 1 and 2.

The calculation inside both NO_LOOP subsystems have to be done only once.

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

NC_NOX_SENS_CONF cannot be greater than NC_CBK_EX_NR.

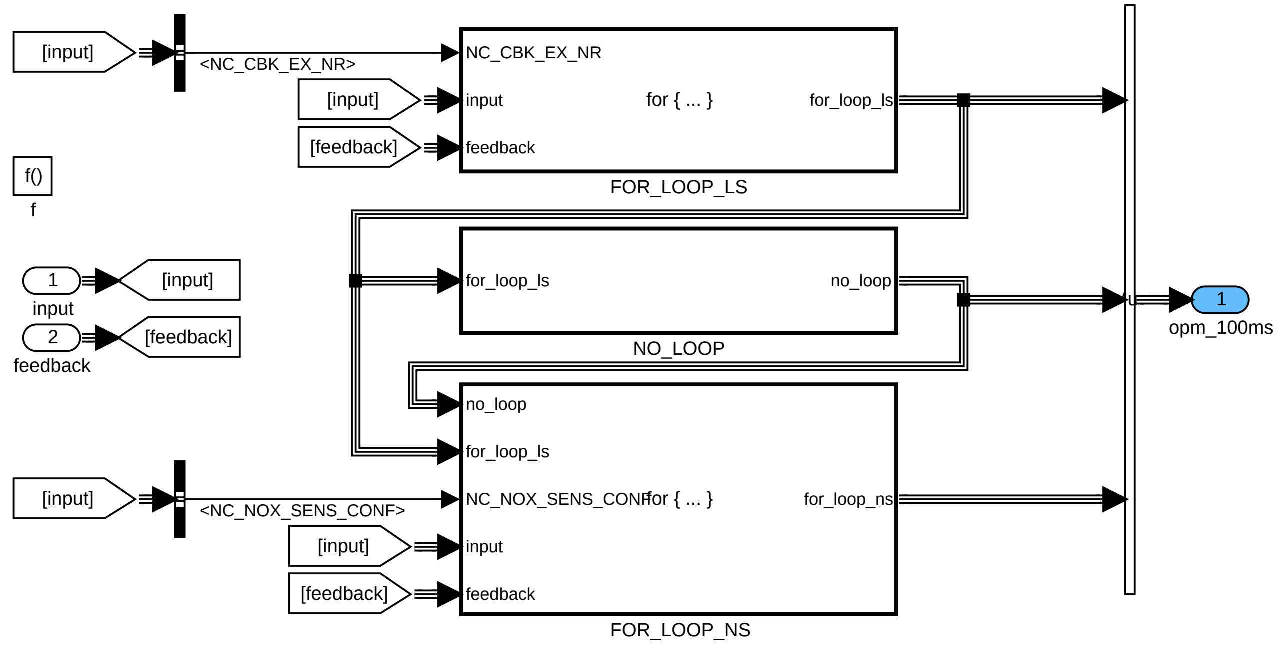


Figure 105 NOXD_MODULB07Y/ OPM_100MS

Calculation for each upstream lambda sensor

Lambda sensor lean check

Linear as well as binary lambda sensors of a bank will be checked if PUC is active.

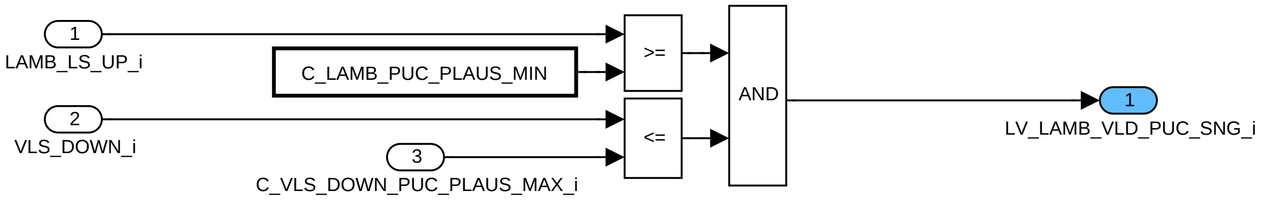


Figure 106 NOXD_MODULB07Y/ OPM_100MS/ FOR_LOOP_LS/ CLC

Bank independent calculation

The signal shows that all lambda sensors of all banks measures lean exhaust gas due to PUC.

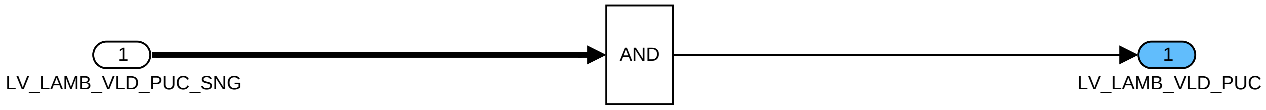


Figure 107 NOXD_MODULB07Y/ OPM_100MS/ NO_LOOP/ CLC

Calculation for each NOx sensor

Activation of diagnosis

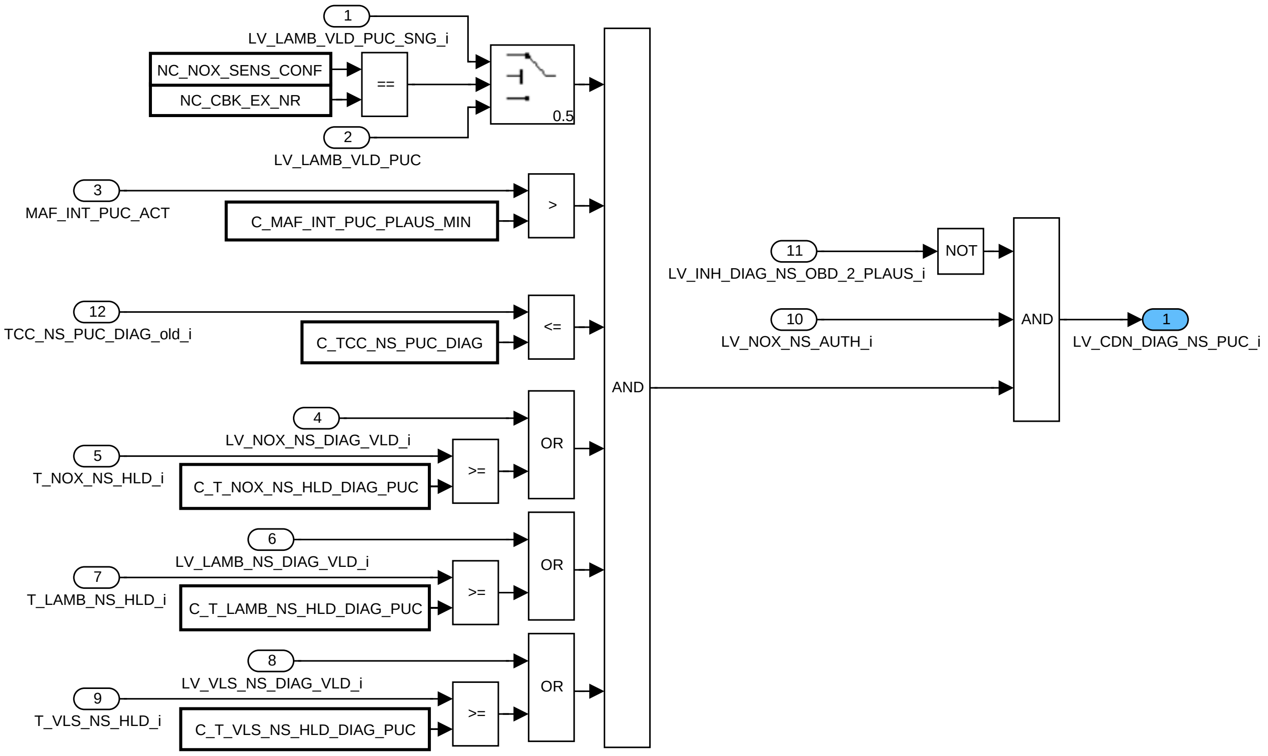


Figure 108 NOXD_MODULB07Y/ OPM_100MS/ FOR_LOOP_NS/ CLC1

Plausibility check counter

At every PUC phase the number of plausibility checks is counted with this counter and limited with C_TCC_NS_PUC_DIAG.

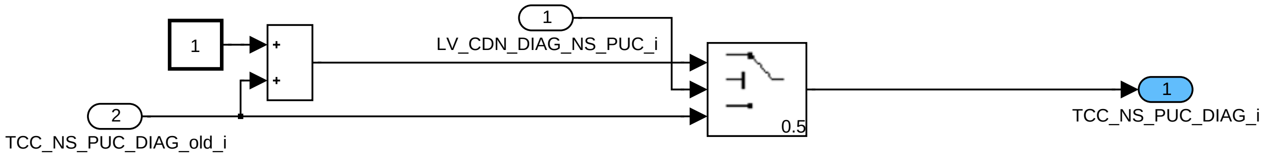


Figure 109 NOXD_MODULB07Y/ OPM_100MS/ FOR_LOOP_NS/ CLC2

Signal diagnosis

Both lambda signals of the NOx sensor will be compared with a threshould.

The NOx signal will be checked with a range.

If the signal is higher than the top limit, then a special check have to be done. The reason is a NOx loading of NOx storage catalyst at start of 'pull fuel cut-off' phase and the fact that the NOx catalyst loses a part of its stored NOx at 'pull fuel cut-off'. Hence a NOx signal, which is higher than the diagnosis threshold at 'pull fuel cut-off' diagnosis time, shall be interpreted as sensor failure only if the NOx loading of NOx catalyst is low at start of 'pull fuel cut-off' diagnosis time.

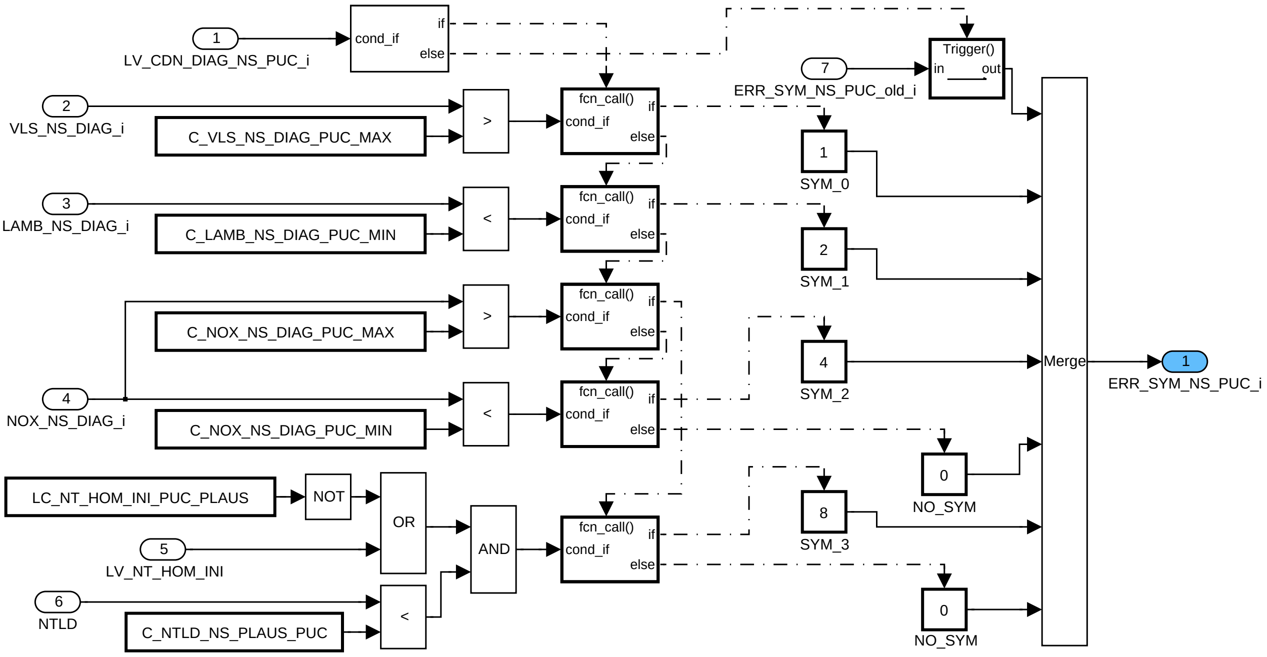


Figure 110 NOXD_MODULB07Y/ OPM_100MS/ FOR_LOOP_NS/ CLC3

Error management

LV_ERR_NS_PUC will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_PUC will be get by ACTION_ERRM_GetLvEndDiag.

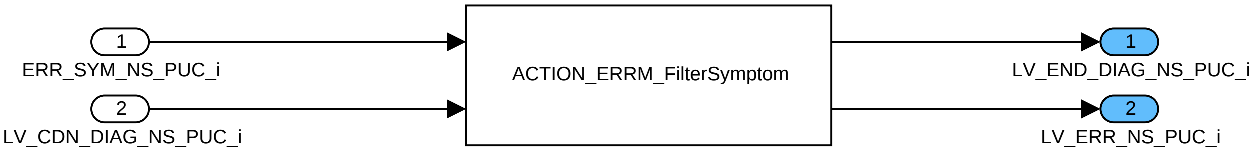


Figure 111 NOXD_MODULB07Y/ OPM_100MS/ FOR_LOOP_NS/ CLC4

52.16 NOx sensor OBDII diag Signal plausibility at rich conditions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_AFR[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor plausibility diagnosis at rich condition					
LV_ERR_NS_AFR[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of NOx sensor rich diagnosis					
ERR_SYM_NS_AFR[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor plausibility diagnosis at rich condition					
LV_CDN_DIAG_NS_AFR[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of NOx sensor plausibility diagnosis at rich condition					
LV_FAC_RANGE_AFR_SNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Signal FAC_LSL_GAIN_AD is inside the desired range for diagnosis at rich condtion					
LV_LAMB_VLD_AFR_SNG[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Lambda signal is valid for diagnosis at rich condition (bench selective)					
LV_MAF_AFR_DIAG_VLD	V	0...1H	0...1	1	-
Mass air flow higher than threshold to allow diagnosis					
MAF_INT_AFR_DIAG[NC_NOX_SENS_CONF]	V	0...FFFFH	0...7.28167E+3	0.11111111	g
Sucked air mass after leaving rich condition					
TCC_NS_AFR_DIAG[NC_NOX_SENS_CONF]	V	0...FFH	0...255	1	-
Test cycle counter for plausibility checks at actual rich phase					

Input data:

FAC_LSL_GAIN_AD[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]	LAMB_NS_DIAG[NC_NOX_SENS_CONF]	LV_INH_DIAG_NS_OBD_2_PLAUS[NC_NOX_SENS_CONF]
LV_LAMB_NS_DIAG_VLD[NC_NOX_SENS_CONF]	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NOX_NS_DIAG_VLD[NC_NOX_SENS_CONF]	LV_VLS_NS_DIAG_VLD[NC_NOX_SENS_CONF]
MAF_KGH	NC_CBK_EX_NR	NC_IDX_DIAG_NS_AFR[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
NOX_NS_DIAG[NC_NOX_SENS_CONF]	T_LAMB_NS_HLD[NC_NOX_SENS_CONF]	T_NOX_NS_HLD[NC_NOX_SENS_CONF]	T_VLS_NS_HLD[NC_NOX_SENS_CONF]
VLS_NS_DIAG[NC_NOX_SENS_CONF]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_AFR	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor plausibility diagnosis at rich condition					
C_ABC_INC_NS_AFR	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor plausibility diagnosis at rich condition					
C_ABC_MAX_NS_AFR	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx sensor plausibility diagnosis at rich condition					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_LSL_GAIN_AD_AFR_MAX	1	0...FFFFH	0...1.99996948	3.05176E-5	-
Maximum WRAF sensor gain adaptation factor for activation of plausibility check at rich condition					
C_FAC_LSL_GAIN_AD_AFR_MIN	1	0...FFFFH	0...1.99996948	3.05176E-5	-
Minimum WRAF sensor gain adaptation factor for activation of plausibility check at rich condition					
C_LAMB_AFR_PLAUS_MAX	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Maximum upstream Lambda value to start the NOx sensor rich check					
C_LAMB_NS_DIAG_AFR_MAX	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Maximum lambda value of NOx sensor at rich condition					
C_MAF_INT_AFR_DIAG_MIN	1	0...FFFFH	0...7.28167E+3	0.11111111	g
MAF threshold for starting the rich diagnosis					
C_MAF_KGH_AFR_DIAG_MIN	1	0...FFFFH	0...2.04797E+3	0.03125	kg/h
Threshold of mass air flow to allow NOx sensor diagnosis at rich conditions					
C_NOX_NS_DIAG_AFR_MIN	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Minimum NOx concentration signal value of NOx sensor at rich condition					
C_TCC_NS_AFR_DIAG_MAX	1	0...FFH	0...255	1	-
Maximum number of test cycles per rich phase					
C_T_LAMB_NS_HLD_DIAG_AFR	1	0...7FFFH	0...327.67	0.01	s
Time with lambda signal state = invalid, after which a plausibility diagnosis at rich condition is started					
C_T_NOX_NS_HLD_DIAG_AFR	1	0...7FFFH	0...327.67	0.01	s
Time with NOx signal state = invalid, after which a plausibility diagnosis at rich condition is started					
C_T_VLS_NS_HLD_DIAG_AFR	1	0...7FFFH	0...327.67	0.01	s
Time with binary O2 signal state = invalid, after which a plausibility diagnosis at rich condition is started					
C_VLS_NS_DIAG_AFR_MIN	1	0...578H	-200...1.2E+3	1	mV
Minimum binary O2 signal voltage of NOx sensor at rich condition					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_FAC_MAF_INT_AFR	1	0...FFFFH	0...1.02398437	1.5625E-5	s
normalization factor for calculation of integral of MAF_KGH					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

52.16.1 General information

This plausibility test checks the sensor signals at rich exhaust gas. With this check shall be detected a sensor poisoning.

Application Condition

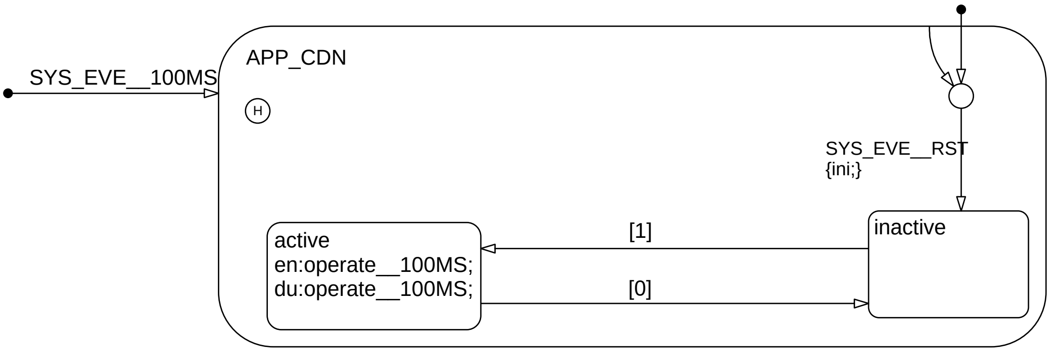
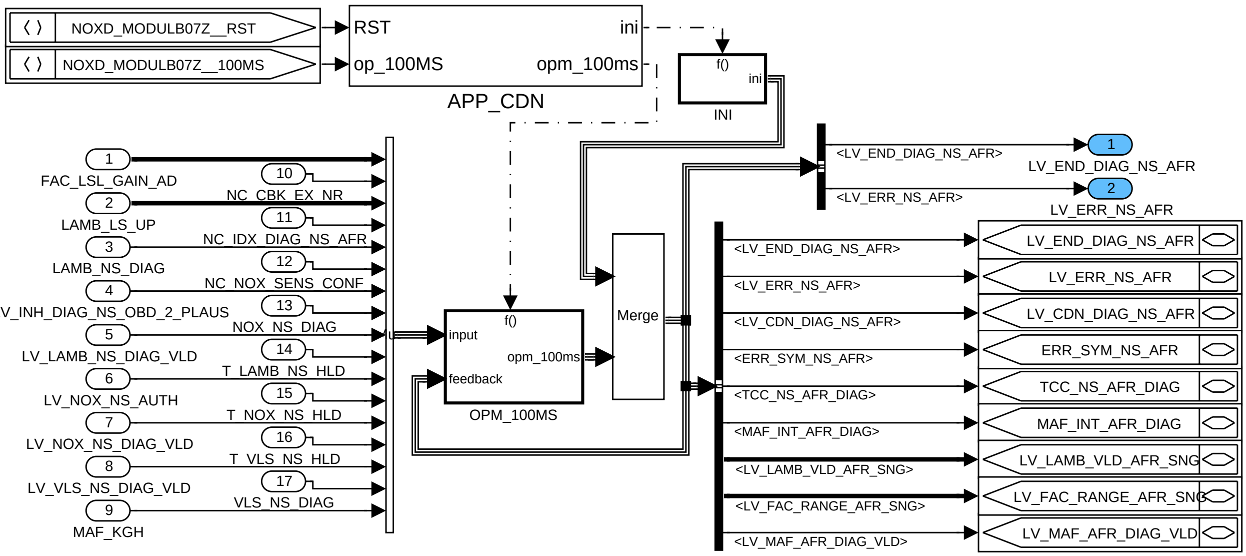


Figure 112 NOXD_MODULB07Z/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 08−Sep−2006

Figure 113 NOXD_MODULB07Z

52.16.1.1 Initialization at reset and deactivation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >)

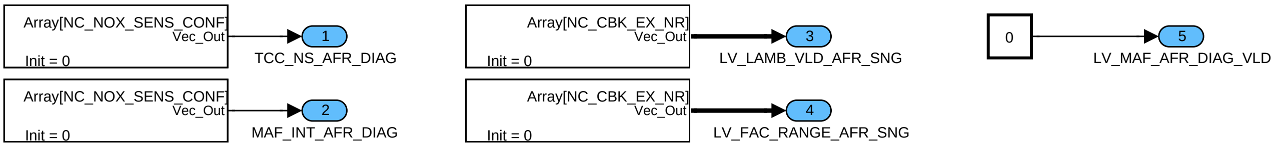


Figure 114 NOXD_MODULB07Z/ INI/ CLC

52.16.1.2 Loop control

The calculation inside FOR_LOOP_LS have to be done for each upstream lambda sensor, controlled by configuration data NC_CBK_EX_NR.

If NC_CBK_EX_NR = 1, then calculation for i = 1.

If NC_CBK_EX_NR = 2, then calculation for i = 1 and 2.

The calculation inside both NO_LOOP subsystems have to be done only once.

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

NC_NOX_SENS_CONF cannot be greater than NC_CBK_EX_NR.

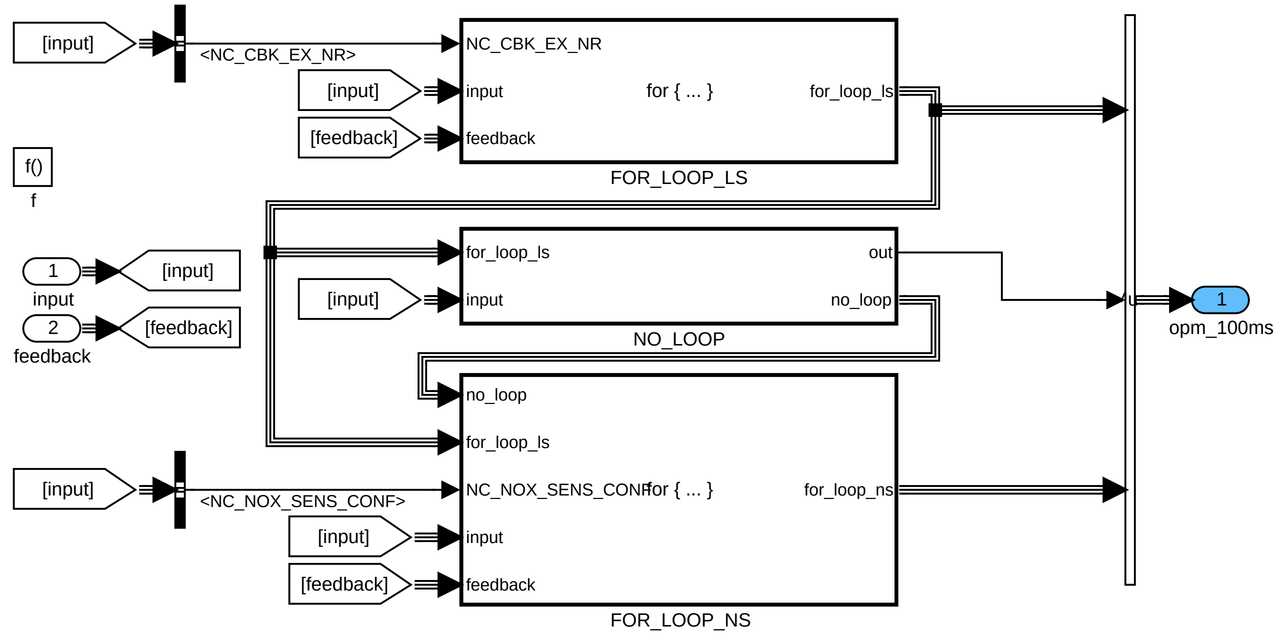


Figure 115 NOXD_MODULB07Z/ OPM_100MS

Calculation for each upstream lambda sensor

Lambda sensor rich check

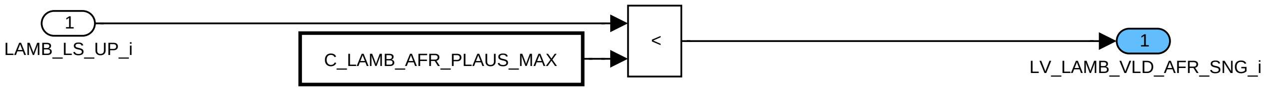


Figure 116 NOXD_MODULB07Z/ OPM_100MS/ FOR_LOOP_LS/ CLC1

Lambda controller gain adaptation

The lambda controller gain adaptation have to be in a desired range to avoid a wrong diagnosis result for the NOx sensor.

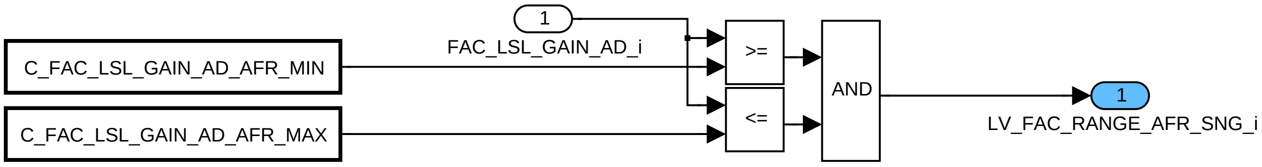


Figure 117 NOXD_MODULB07Z/ OPM_100MS/ FOR_LOOP_LS/ CLC2

Bank independent calculation

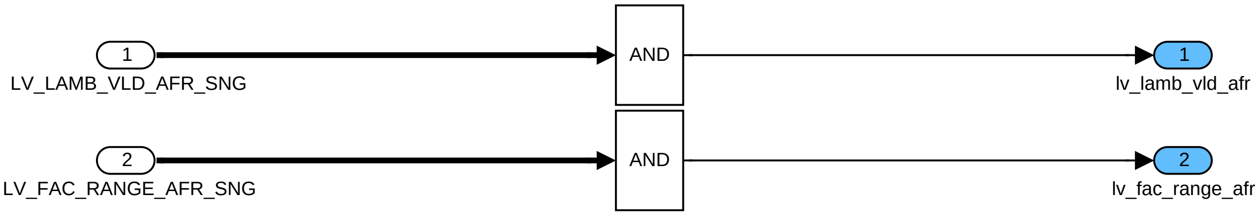


Figure 118 NOXD_MODULB07Z/ OPM_100MS/ NO_LOOP/ CLC1

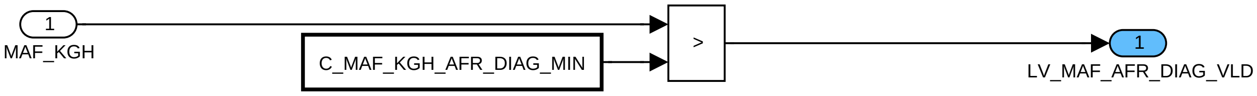


Figure 119 NOXD_MODULB07Z/ OPM_100MS/ NO_LOOP/ CLC2

Calculation for each NOx sensor

MAF integral and plausibility check counter

At every rich phase the number of plausibility checks will be counted with this counter and limited with C_MAF_INT_AFR_DIAG_MIN.

The MAF integral will start if the upstream lambda becomes higher than the rich threshould due to identify, when the rich exhaust gas should be arrived at the NOx sensor.

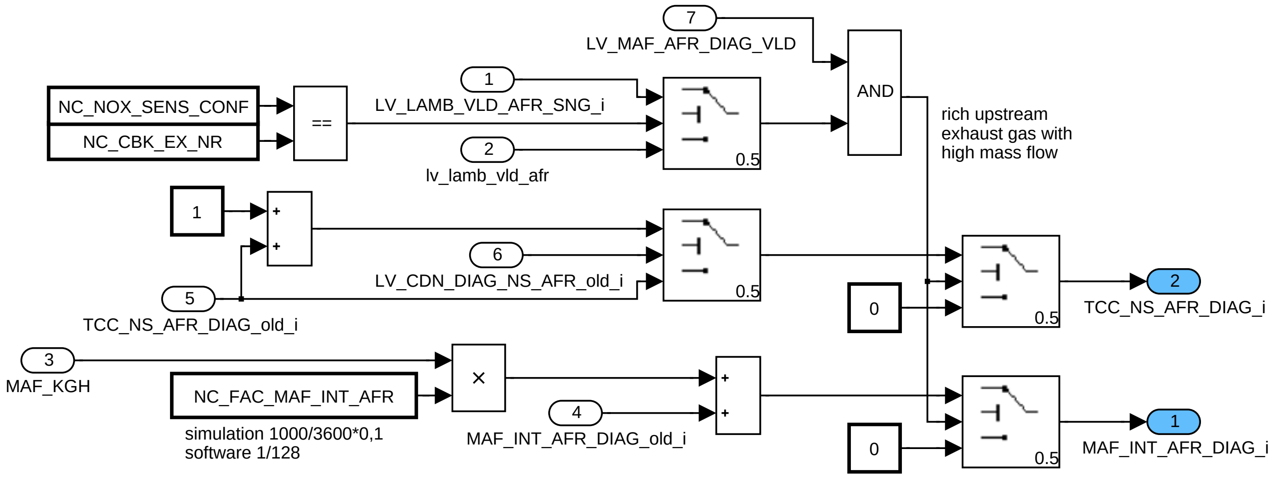


Figure 120 NOXD_MODULB07Z/ OPM_100MS/ FOR_LOOP_NS/ CLC1

Diagnosis evaluation

The NOx sensor may only be identified as defect if the lambda control adaptation is inside the desired range. Outside this range the sensor may only be identified as OK.

The check of the NOx threshould C_NOX_NS_DIAG_AFR_MIN uses the NOx sensor cross sensitivity on NH3, which is originate in the catalyst at rich exhaust gas conditions.

Application hint: The constant C_MAF_INT_AFR_DIAG_MIN shall be calibrated so that this diagnosis is not activated during a NOx catalyst regeneration.

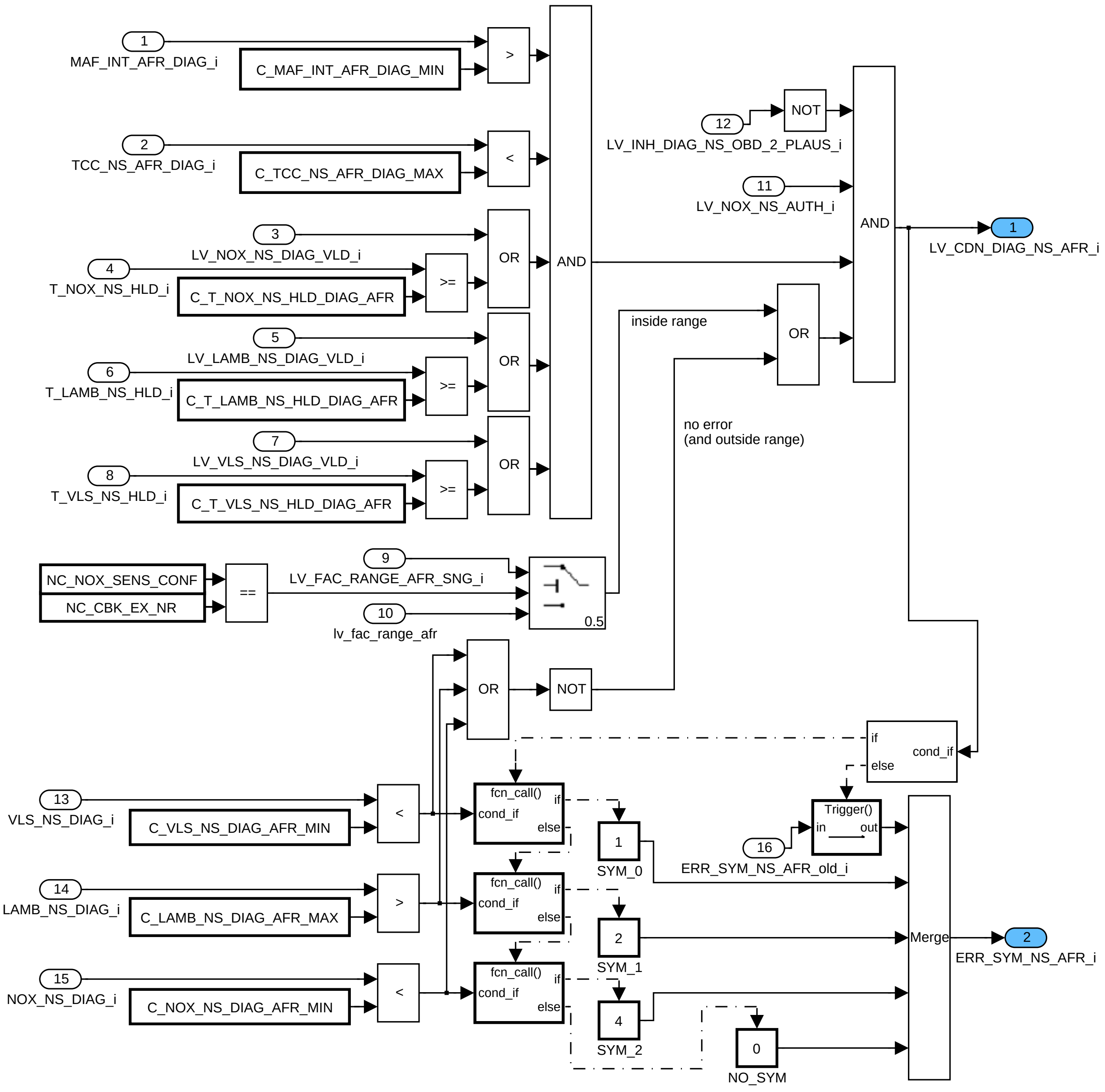


Figure 121 NOXD_MODULB07Z/ OPM_100MS/ FOR_LOOP_NS/ CLC2

Error management

LV_ERR_NS_AFR will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_AFR will be get by ACTION_ERRM_GetLvEndDiag.

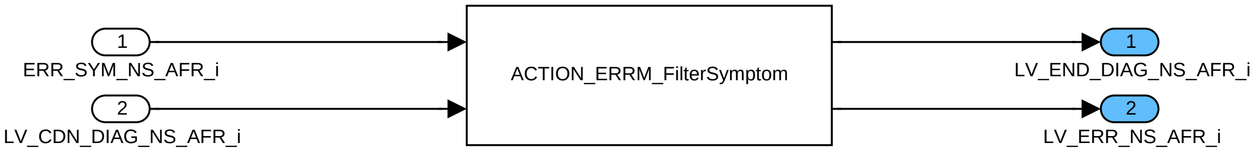


Figure 122 NOXD_MODULB07Z/ OPM_100MS/ FOR_LOOP_NS/ CLC3

52.17 NOx sensor OBDII diag Regeneration stop observation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_STOP[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor plausibility diagnosis at regeneration stop					
LV_ERR_NS_STOP[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of NOx sensor diagnosis at regeneration stop					
LV_INH_NT_RGN_STOP_MDL_DIAG[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Request to NOx catalyst management to inhibit the model based NOx regeneration stop					
CTR_RGN_STOP_MDL[NC_NOX_SENS_CONF]	V	0...FFH	0...255	1	-
Number of successive regenerations, which are stopped by model					
ERR_SYM_NS_STOP[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor plausibility diagnosis at regeneration stop					
LV_CDN_DIAG_NS_STOP[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of NOx sensor plausibility diagnosis at regeneration stop					
LV_INH_NTL_DEC_INT_DIAG[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Inhibition of signal diagnosis by regeneration agent integral because od high NOx catalyst temperature difference					
LV_NS_STOP_DIAG_RUN[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Indicator that regeneration stop observation is running					
NTL_DEC_INT_THD[NC_NOX_SENS_CONF]	V	0...FFFFH	0...1.04856E+4	0.16	mg
Integral of the lack of oxygen during regeneration phase					
T_MAF_KGH_RGN_MIN[NC_NOX_SENS_CONF]	V	0...FFFFH	0...655.35	0.01	s
Timer which measures the regeneration duration at which th mass air flow is higher than a threshold					

Input data:

LAMB_LS_UP[NC_CBK_EX_NR]	LAMB_NS[NC_NOX_SENS_CONF]	LV_HOM_AFS_ACT	LV_INH_DIAG_NS_OBD_2_PLAUS[NC_NOX_SENS_CONF]
LV_LAMB_NS_VLD[NC_NOX_SENS_CONF]	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NT_MDL_AFR	LV_NT_RGN_REQ
LV_NT_RGN_STOP_SENS	LV_NT_SENS_AFR[NC_NT_NR]	LV_ST_END	LV_NT_TOUT_AFR
LV_VLS_NS_VLD[NC_NOX_SENS_CONF]	MAF_KGH	NC_IDX_DIAG_NS_STOP[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
NC_NT_NR	NTL_DEC_INT[NC_NT_NR]	NTL_RGN_ST	TNT_MDL_1[NC_NT_NR]
TNT_MDL_2[NC_NT_NR]	TNT_MDL_H	TNT_MDL_L	TNT_MDL_MV
TNT_MDL_MV_SNG[NC_NT_NR]	T_RGN	VLS_NS[NC_NOX_SENS_CONF]	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_STOP	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor diagnosis at regeneration stop					
C_ABC_INC_NS_STOP	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor diagnosis at regeneration stop					
C_ABC_MAX_NS_STOP	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx sensor diagnosis at regeneration stop					
C_CTR_RGN_STOP_MDL_MAX	1	0...FFH	0...255	1	-
maximum number of successive regenerations, which are stopped by model					
C_LAMB_LS_UP_NS_STOP_DIAG	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Lambda signal threshold for counting time of rich exhaust gas with not too low flow					
C_LAMB_NS_RGN_STOP_DIAG	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Diagnosis threshold for detection of WRAF signal capability to stop a regeneration					
C_MAF_KGH_RGN_MIN	1	0...FFFFH	0...2.04797E+3	0.03125	kg/h
Minimum mass air flow for set free of failure count					
C_NTL_DEC_INT_DFT_THD	1	0...FFFFH	0...1.04856E+4	0.16	mg
Default diagnosis threshold					
C_NTL_RGN_ST_MIN	1	0...FFFFH	0...1.04856E+4	0.16	mg
Threshold for switch between default and calculated diagnosis threshold					
C_STATE_SENS_AFR_NS_DIAG[NC_NOX_SENS_CONF]	1	0H 1H 2H 3H 4H	VLS_DOWN VLS_NOX_SENS LAMB_NOX_SENS LS_NT_DOWN NONE	1	-
bench-selective mode to determine rich exhaust gas downstream NOx trap					
C_TNT_MDL_DIF_MAX	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Temperature threshold for set free of signal diagnosis by regeneration agent integral					
C_TNT_NS_DIAG_STOP_MAX	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Threshold for disabling NOx sensor regeneration stop diagnosis at high NOx trap temperature					
C_T_MAF_KGH_RGN_MIN	1	0...FFFFH	0...655.35	0.01	s
Time threshold for set free of failure count					
C_T_RGN_NS_RGN_STOP_DIAG	1	0...FFFFH	0...655.35	0.01	s
Minimum regeneration duration for activation of regeneration stop observation					
C_VLS_NS_RGN_STOP_DIAG	1	0...578H	-200...1.2E+3	1	mV
Diagnosis threshold for detection of binary Lambda signal capability to stop a regeneration					
IP_FAC_NTL_RGN_ST	6	0...FFFFH	0...255	0.00389105	-
LDP_TNT_MDL_MV_IP_FAC_NTL_RGN	6	0...FFFFH	0...1.02398E+3	0.015625	°C
Temperature dependent factor for calculation of diagnosis threshold					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

52.17.1 General description

This diagnosis observes the Lambda signal of NOx sensor, which was selected to stop a NOx catalyst regeneration. The signal is checked on its capability to stop a NOx catalyst regeneration. It is possible that the signal can not exceed the NOx catalyst regeneration stop threshold because of sensor aging or a defect. In this case the regeneration is stopped by a time limit. If this problem is present then it is not possible to fulfil the emission regulations. Then it is necessary to inhibit the engine lean burn operation mode and to mark the NOx sensor as “defect”.

Application Condition

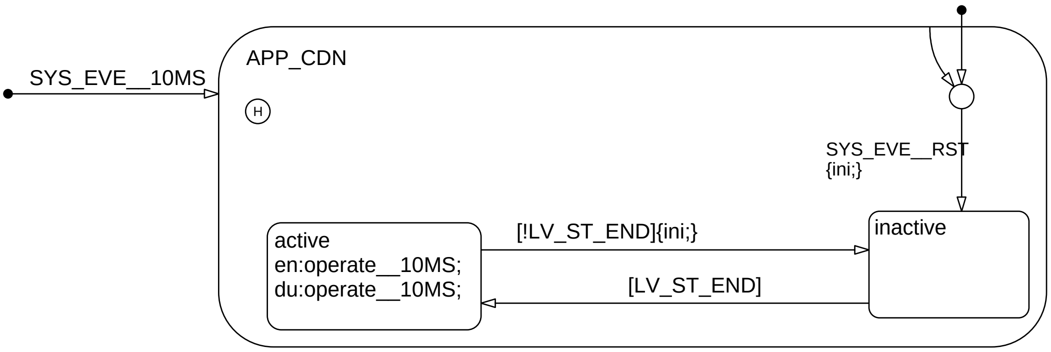
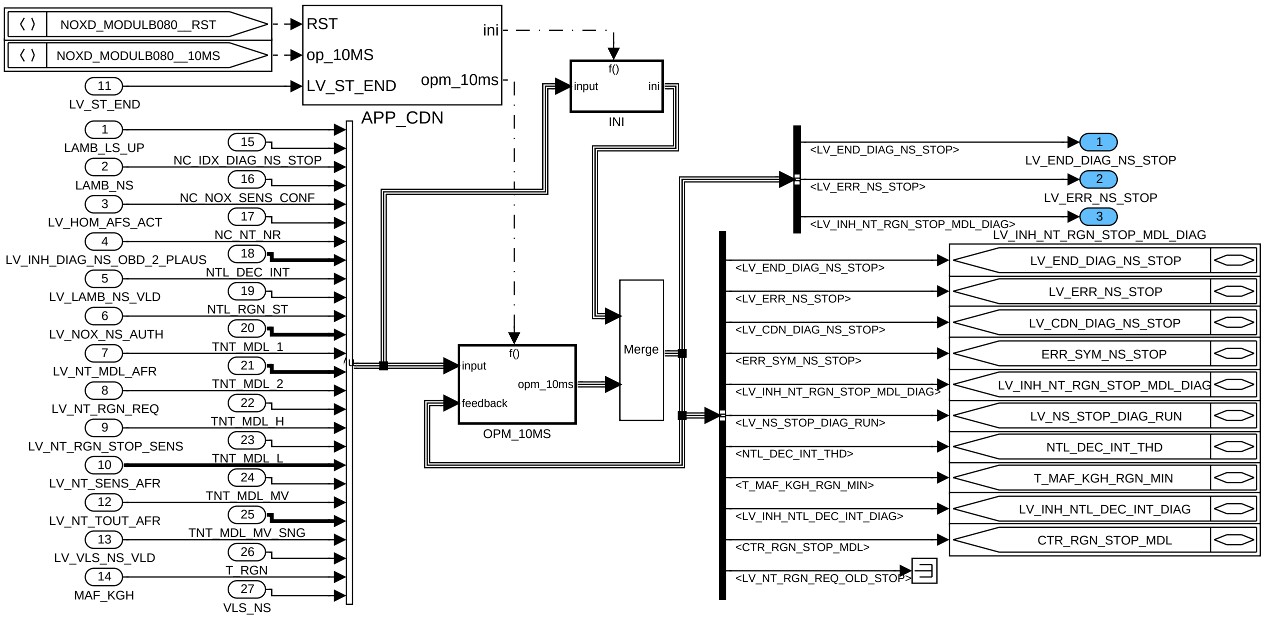


Figure 123 NOXD_MODULB080/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 13−Sep−2006

Figure 124 NOXD_MODULB080

52.17.1.1 Initialization at reset and deactivation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >)

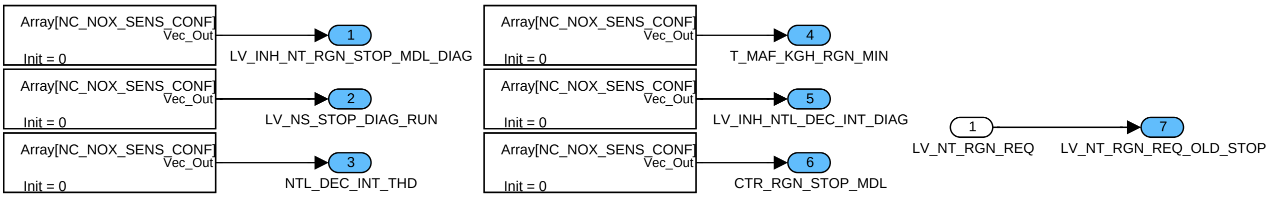


Figure 125 NOXD_MODULB080/ INI/ CLC

52.17.1.2 Loop control

The calculation inside both NO_LOOP subsystems have to be done only once.

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

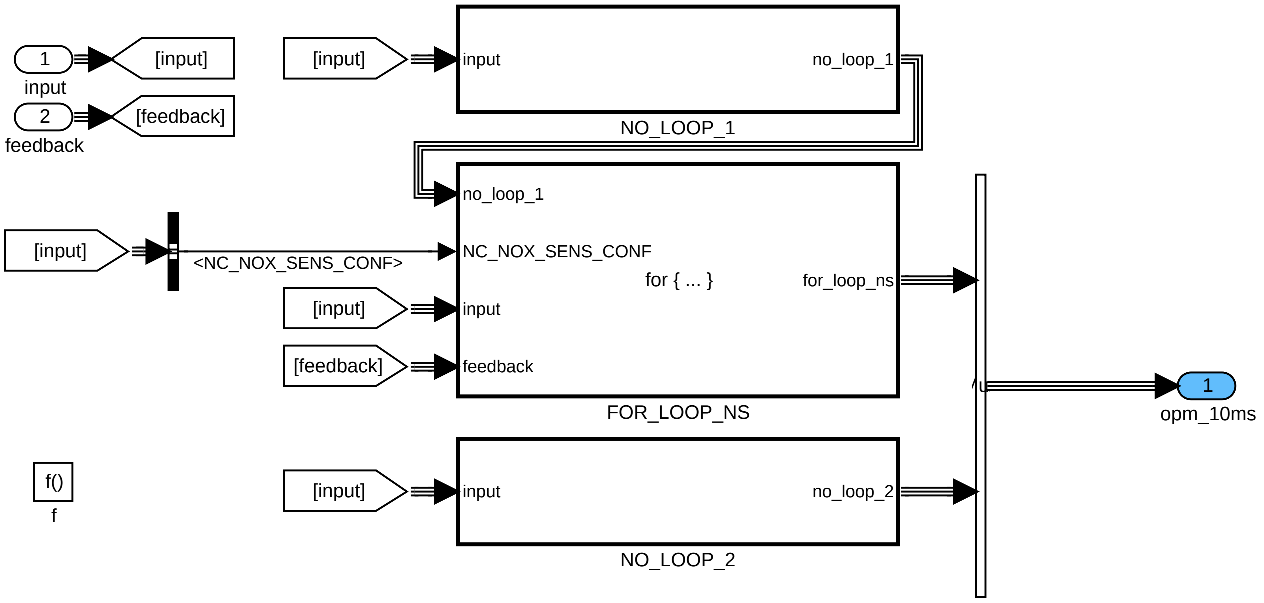


Figure 126 NOXD_MODULB080/ OPM_10MS

Bank independent calculation (part 1)

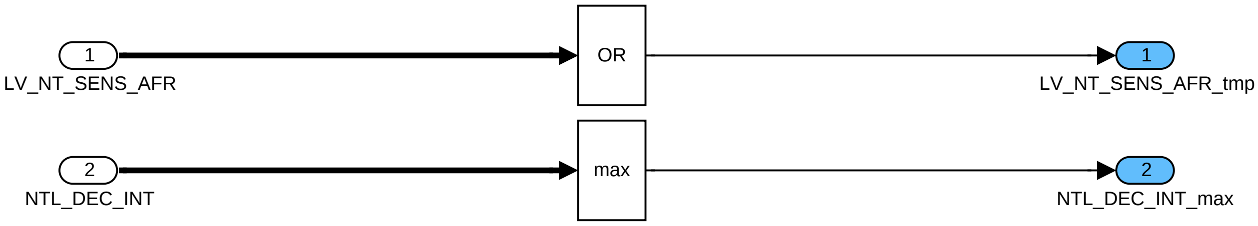


Figure 127 NOXD_MODULB080/ OPM_10MS/ NO_LOOP_1/ CLC

Calculation for each NOx sensor

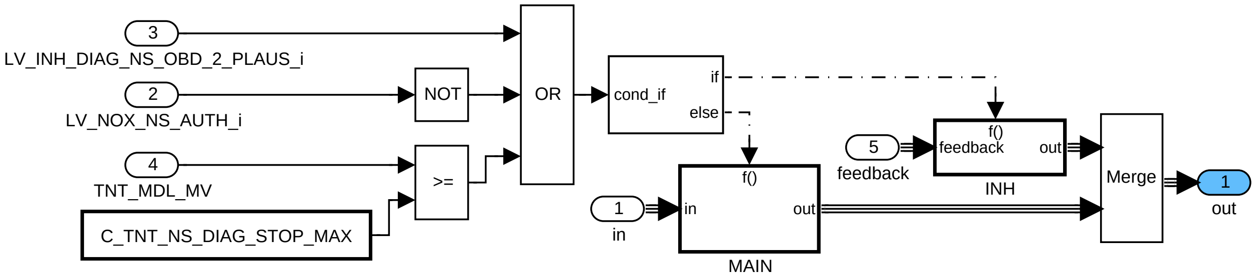


Figure 128 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC

Inhibition of diagnosis for one sensor

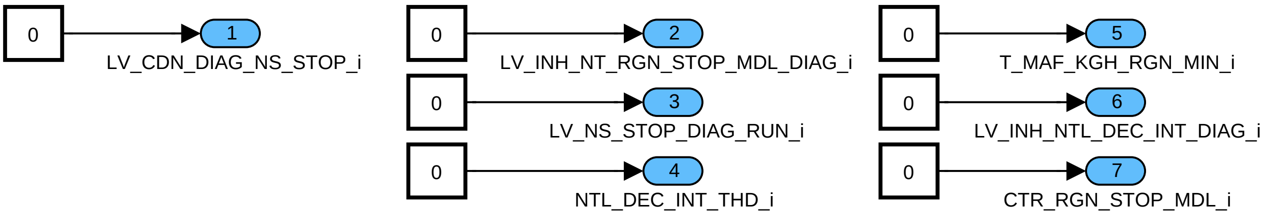


Figure 129 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ INH/ CLC

Main part of diagnosis

Start of regeneration stop observation

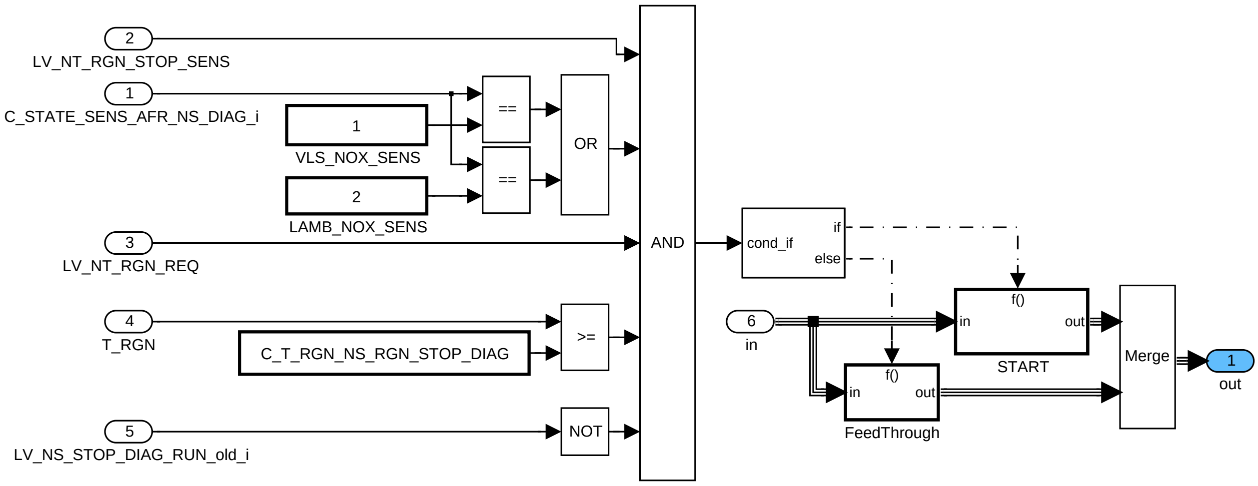


Figure 130 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC1

The diagnosis threshold (maximum necessary amount of regeneration agent) NTL_DEC_INT_THD_i is calculated by the absolute NOx and O2 loading of NOx catalyst at start of regeneration NTL_RGN_ST and a safety factor. The safety factor depends on the mean NOx catalyst monolith temperature in order to consider the NOx storage capability of catalyst.

Additionally the temperature difference between first and second NOx catalyst monolith is determined at the start of regeneration. If this temperature threshold is higher than the threshold C_TNT_MDL_DIF_MAX then regeneration agent diagnosis is inhibited because the NOx catalyst regeneration behaviour is not exactly predictable.

At the first regeneration (after warm up of catalyst) NTL_RGN_ST is about 0. Then the default diagnosis threshold C_NTL_DEC_INT_DFT_THD is used.

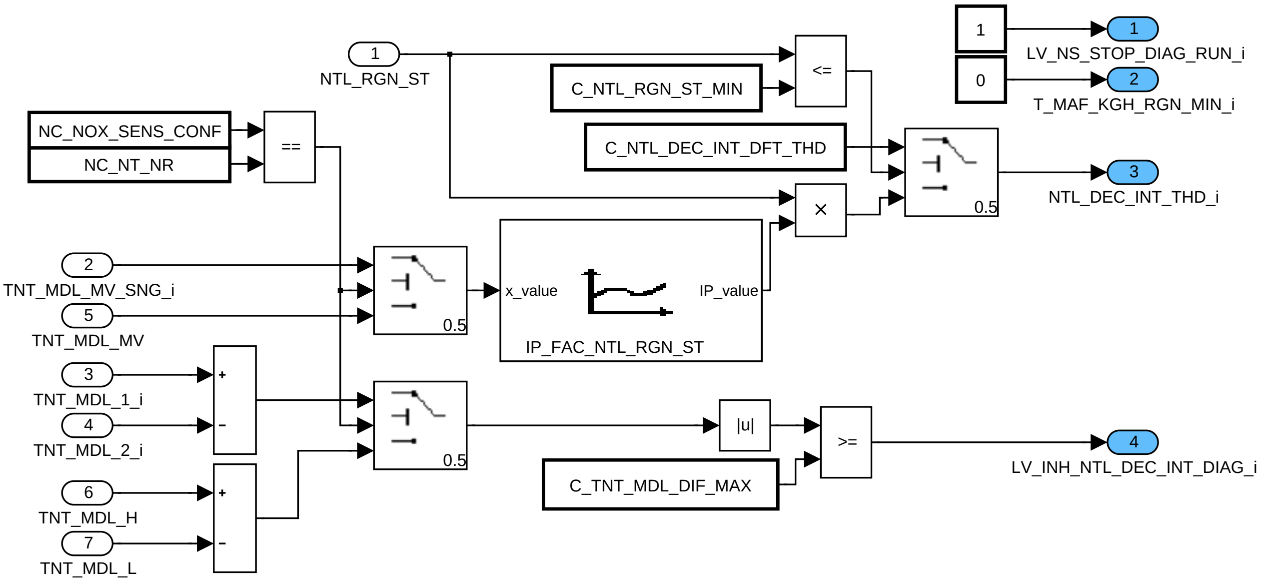


Figure 131 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC1/ START/ CLC

Observation of regeneration agent integral during regeneration

The system knows the absolute loading of NOx catalyst with NOx and O2. On this base it is possible to determine the necessary amount of regeneration agent. When this amount of regeneration agent was transferred to the NOx catalyst then the NOx catalyst shall be cleared from NOx and O2. Now the sensor signal must stop the NOx regeneration by exceeding of regeneration stop threshold. Otherwise the sensor signal is wrong.

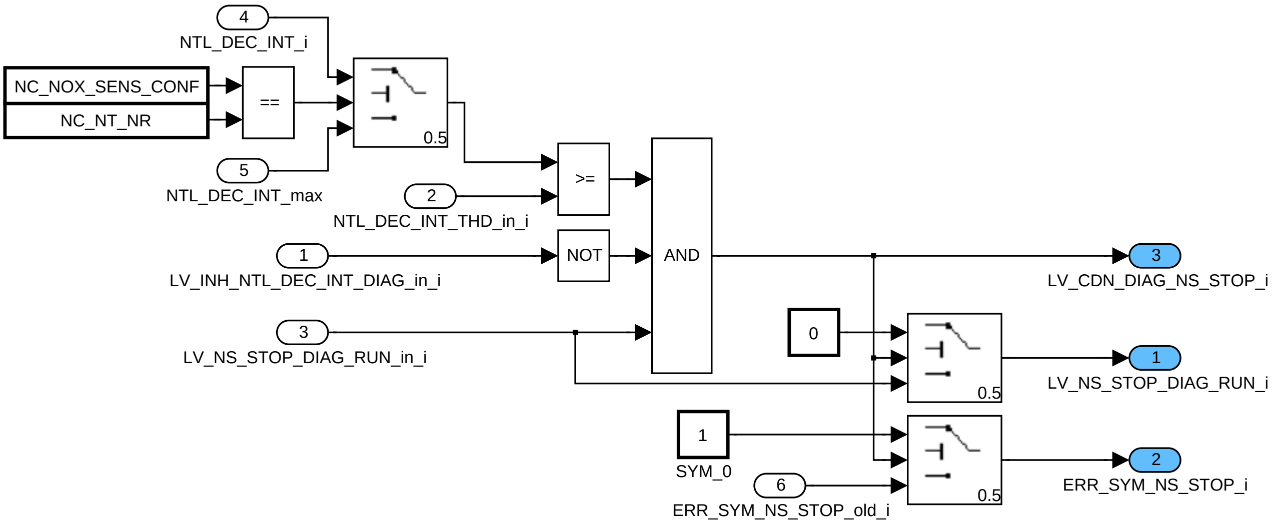


Figure 132 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC2

Observation of air flow and lambda during regeneration

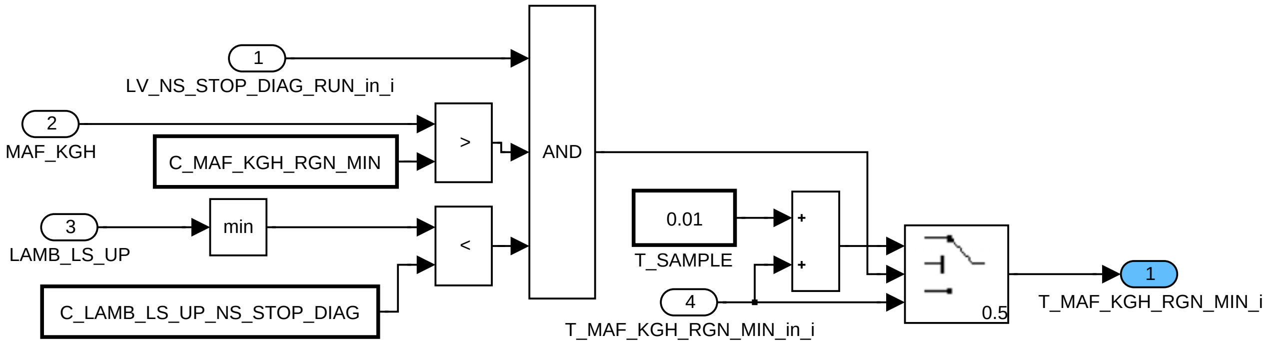


Figure 133 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC3

Detection of regeneration stop

The NOx catalyst regeneration stop can be caused by

- rich exhaust gas, which is detected by a sensor based lambda signal
- NOx catalyst model
- maximum regeneration duration (time out).

The NOx catalyst management indicates the regeneration stop method by the bits LV_SENS_AFR_i (sensor stop), LV_NT_MDL_AFR (catalyst model) and LV_T_OUT_AFR (time out). This bits are used to observe the lambda signals of NOx sensor on its capability to stop a NOx catalyst regeneration.

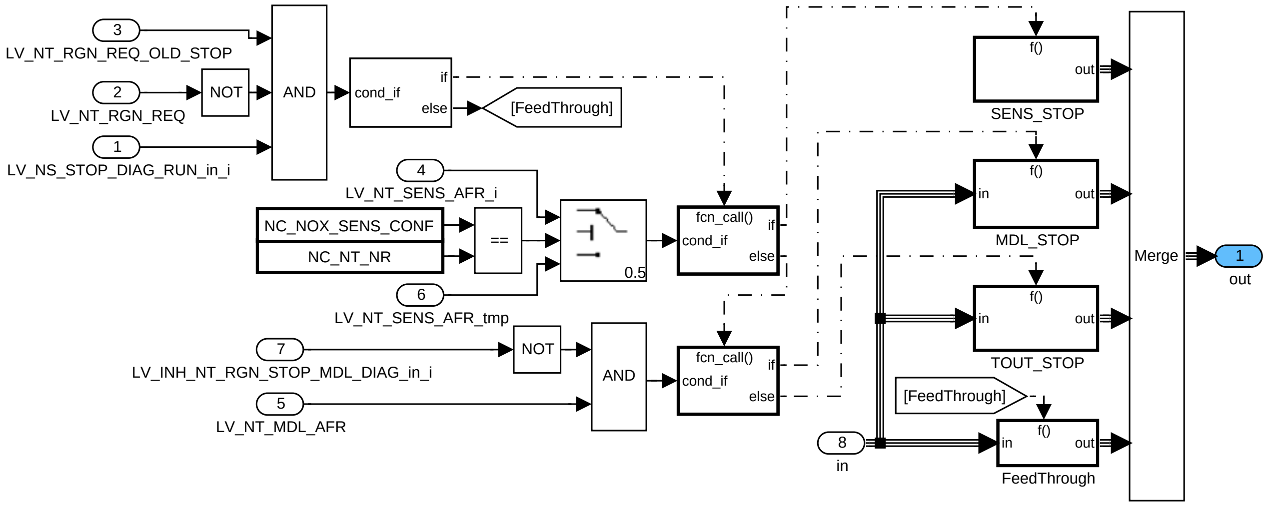


Figure 134 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC4

Regeneration stop by sensor

Normally the NOx catalyst regeneration is stopped by a sensor signal. Then the sensor is classified as OK.

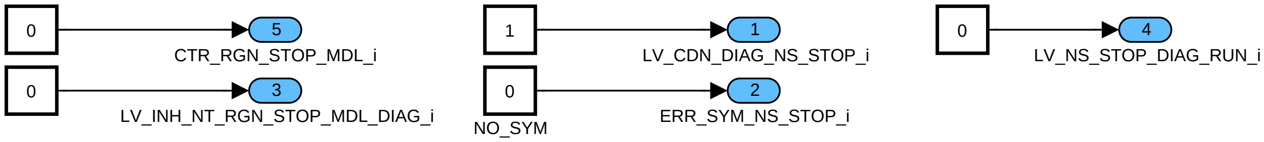


Figure 135 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC4/ SENS_STOP/ CLC

Regeneration stop by model

If the regeneration was stopped by the NOx catalyst model, then it is not possible to classify the sensor as OK or defect. Hence the number of successive regenerations, which are stopped by model, is determined by CTR_RGN_STOP_MDL_i. If CTR_RGN_STOP_MDL_i exceeds a threshold then the regeneration stop by model is inhibited by set of LV_INH_NT_RGN_STOP_MDL_DIAG_i. Now the next regeneration can be stopped only by sensor (LV_SENS_AFR_i) or by time out (LV_T_OUT_AFR). If the next regeneration is stopped by sensor then the sensor is OK. Remark: In this case the calibration values and/ or adaptation values of model should be checked.

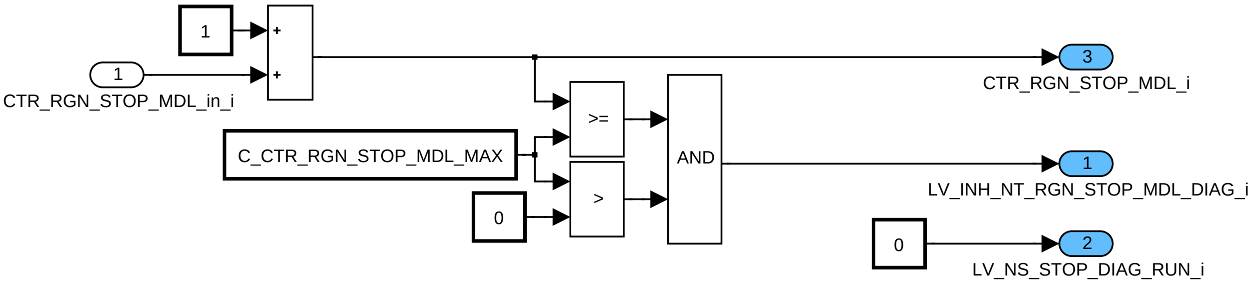


Figure 136 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC4/ MDL_STOP/ CLC

Regeneration stop by time out

If the regeneration is stopped by time out then the sensor is not able to stop the regeneration (sensor is defect) or the calibration of maximum regeneration duration is to short. Especially regenerations with a low air flow (idle speed) are critical. Hence the mass air flow is observed during the regenerations. The observation measures the time at which the mass air flow is higher than the threshold C_MAF_KGH_RGN_MIN. If a regeneration time out is reached now then it is checked the duration at which the mass air flow was higher than the threshold C_MAF_KGH_RGN_MIN. When the regeneration duration with a sufficient mass air flow is lower than C_T_MAF_KGH_MIN then this signal diagnosis is inhibited, the failure counter is not incremented. When the regeneration duration with a sufficient mass air flow is higher than C_T_MAF_KGH_MIN then the sensor is classified as defect and the failure counter is incremented.

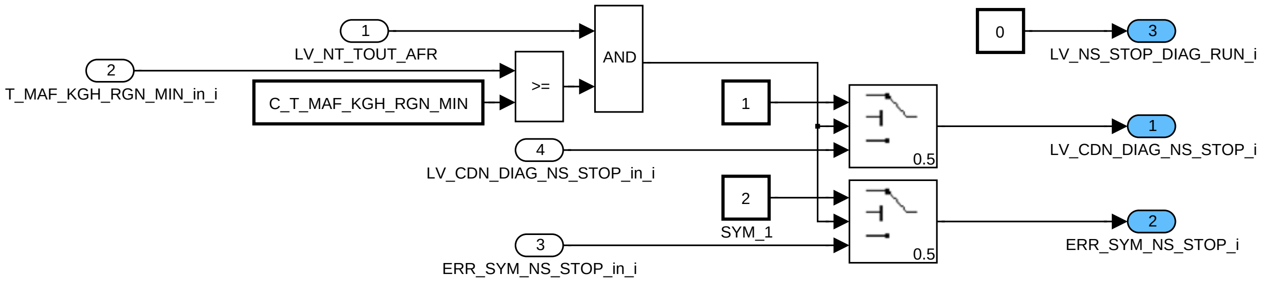


Figure 137 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC4/ TOUT_STOP/ CLC

Lambda 1 operation

It is possible that the Lambda signals of NOx-Sensor exceeds the regeneration stop threshold at homogeneous Lambda = 1.0 engine operation. This fact is used to classify the sensor as OK.

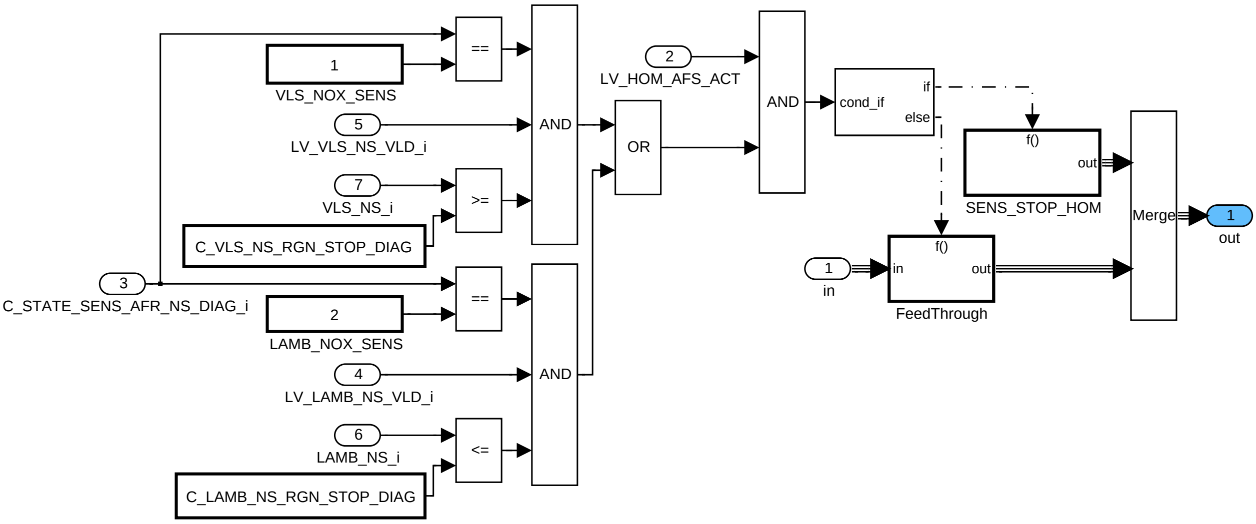


Figure 138 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC5

Figure 139 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC5/ SENS_STOP_HOM/ CLC

Error management

LV_ERR_NS_STOP will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_STOP will be get by ACTION_ERRM_GetLvEndDiag.

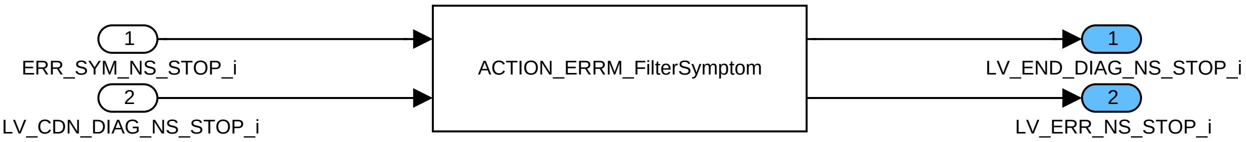


Figure 140 NOXD_MODULB080/ OPM_10MS/ FOR_LOOP_NS/ CLC/ MAIN/ CLC6

Bank independent calculation (part 2)

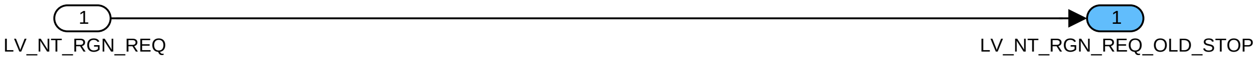


Figure 141 NOXD_MODULB080/ OPM_10MS/ NO_LOOP_2/ CLC

52.18 NOx signal gain adaptation diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_GAIN[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Diagnostic performed at last one time for NOx signal gain adaptation					
LV_ERR_NS_GAIN[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	[-]
Present failure after filtering of the NOx signal gain adaptation diagnosis					

Input data:

		NC_NOX_SENS_CONF	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_NOX_NS_AD_AFS_MAX	1	0...FFFFH	0...127.99804	1.9531e-3	[-]
NOx signal adaptation factor - maximum value - for requesting Lambda=1.0 operation					
C_FAC_NOX_NS_AD_AFS_MIN	1	0...FFFFH	0...127.99804	1.9531e-3	[-]
NOx signal adaptation factor - minimum value - for requesting Lambda=1.0 operation					

FUNCTION DESCRIPTION:

General information:

Stub module for recource optimization

Application conditions:

Initialisation: at reset:

LV_END_DIAG_NS_GAIN[i] = 0

LV_ERR_NS_GAIN[i] = 0

Recurrence:

Activation:

Deactivation:

Formula section:

52.19 NOx sensor setpoint shift diag manager

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_NS_SHIFT_CYC [NC_NOX_SENS_CONF]	O/V/S	0... FFFFH	0... 65535	1	[-]
Counter of finished NOx signal shift diagnosis					
CTR_NS_SHIFT_CYC_NOT_VLD [NC_NOX_SENS_CONF]	V	0... FFH	0... 255	1	[-]
Counter for not valid diagnosis cycles					
CTR_NS_SHIFT_CYC_VLD [NC_NOX_SENS_CONF]	V	0... FFH	0... 255	1	[-]
Counter of valid diagnosis cycles					
CTR_NS_SHIFT_DIAG_INH	V	0... FFH	0... 255	1	[-]
Counter of successful regenerations					
DIST_NT_NS_SHIFT [NC_NOX_SENS_CONF]	O/V/S	0... FFFFH	0... 524280	8	[km]
Distance between two active NOx signal diagnosis					
FCO_DELTA_NS_SHIFT [NC_NOX_SENS_CONF]	O/V	0... FFFFFFFFH	0... 4.29497e+9	1	[µl]
Additional fuel consumption for NOx signal shift diagnosis					
LAMB_NS_MMV_NS_SHIFT [NC_NOX_SENS_CONF]	V	0... 7FFFFFFFH	0... 31.9999999851	14.9012e-9	[-]
Filtered LAMB_NS signal					
LAMB_NS_MMV_NS_SHIFT_DIF_ABSV [NC_NOX_SENS_CONF]	V	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Absolut value of difference between filtered and origin LAMB_NS signals					
LV_DIAG_ACT_CDN_DYN_NS_TMP [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Dynamic activation conditions for the shift diagnosis due to NOx Sensor check					
LV_DIAG_ACT_CDN_DYN_TMP	V	0... 1H	0... 1	1	[-]
Dynamic activation conditions for the shift diagnosis					
LV_DIAG_ACT_CDN_STAT_NS_TMP [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Static activation conditions for the shift diagnosis due to NOx Sensor check					
LV_DIAG_ACT_CDN_STAT_TMP	V	0... 1H	0... 1	1	[-]
Static activation conditions for the shift diagnosis					
LV_DIAG_ACT_CDN_SUM [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Summary activation conditions for the shift diagnosis					
LV_DIAG_ACT_CDN_VLS_DOWN_TMP	V	0... 1H	0... 1	1	[-]
Dynamic activation conditions for the shift diagnosis					
LV_DIAG_NS_SHIFT_END [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
The diagnosis was executed last time					
LV_DIAG_REQ_NS_SHIFT [NC_NOX_SENS_CONF]	O/V	0... 1H	0... 1	1	[-]
Request for NOx sensor for active signal shift diagnosis					
LV_END_DIAG_NS_SHIFT_MNG_OLD [NC_NOX_SENS_CONF]	-	0... 1H	0... 1	1	[-]
Old value of LV_END_DIAG_NS_SHIFT					
LV_ERR_NS_SHIFT_MNG_OLD [NC_NOX_SENS_CONF]	-	0... 1H	0... 1	1	[-]
Old value of LV_ERR_NS_SHIFT					
LV_LAMB_REQ_DEAC_NS [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Deactivation combustion request due to NOx sensor binary signal					
LV_NS_SHIFT_CMB_INT_REQ	O/V	0... 1H	0... 1	1	[-]
Request for combustion managment					

LV_NS_SHIFT_CMB_INT_REQ_SNG [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Request for combustion managment, NOx sensor specific					
LV_NS_SHIFT_DIAG_ACT_EXT_ADJ [NC_NOX_SENS_CONF]	O/V	0... 1H	0... 1	1	[-]
Activation of the diagnosis by service tool					
LV_NS_SHIFT_DIAG_DEAC_EXT_ADJ [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Manual deactivation of the diagnosis by service tool					
LV_NS_SHIFT_DIAG_EXT_ADJ [NC_NOX_SENS_CONF]	O/V	0... 1H	0... 1	1	[-]
External adjustment values for NOx signal shift diagnosis available					
LV_NS_SHIFT_DIAG_INI [NC_NOX_SENS_CONF]	O/V	0... 1H	0... 1	1	[-]
Initialization of NOx signal shift diagnosis variables					
MAF_CYL_INT_NS_SHIFT	V	0... FFFFH	0... 1820.41666667	0.0277778	[g]
MAF integral for inhibition of the diagnosis after PUC phase					
MAF_CYL_INT_NS_SHIFT_DYN [NC_NOX_SENS_CONF]	V	0... FFFFH	0... 1820.41666667	0.0277778	[g]
MAF integal during limited dynamics are fulfilled					
MAF_MMV_NS_SHIFT	V	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Filtered MAF signal					
MAF_MMV_NS_SHIFT_DIF_ABSV	V	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Absoult value of limited dynamic value of MAF					
STATE_EOL_KWP_NS_SHIFT_DIAG [NC_NOX_SENS_CONF]	O/V	0 1 2 3 4 5 6 7 8 9	NOT_START ST_INH PAR_NOT_PLAUS WAIT_REL UNDEF ACT END_WOUT_RESULT ABORTED END_WOUT_ERR END_WITH_ERR	-	[-]
State of NOx-SensorShiftDiagnosis					
STATE_NS_SHIFT_CMB_REQ	O/V	0... FFFFH	0... 65535	1	[-]
Requested mode for combustion manager					
STATE_NS_SHIFT_DIAG [NC_NOX_SENS_CONF]	O/V	0... FFFFH	0... 65535	1	[-]
NOx signal shift diagnosis state					
T_AFL_NS_SHIFT_DIAG_INH	V	0... FFFFH	0... 6553.5	0.1	[s]
Time since last successful regeneration					
T_NS_SHIFT_CMB_INT_REQ	V	0... FFFFH	0... 6553.5	0.1	[s]
Time for calculation of combustion request					
T_NS_SHIFT_CMB_INT_REQ_SUM	V	0... FFFFH	0... 6553.5	0.1	[s]
Timer of active request in one DC					
T_NS_SHIFT_DEAC_SO2P	V	0... FFFFH	0... 6553.5	0.1	[s]
Timer after desulphurization					
T_NS_SHIFT_DEAC_TEMP [NC_NOX_SENS_CONF]	V	0... 9F6H	0... 255	0.1	[s]
Timer for activation of the NOx signal shift diagnosis after set of the dew point					
T_NS_SHIFT_READY_WAIT [NC_NOX_SENS_CONF]	V	0... FFH	0... 25.5	0.1	[s]
Timer for waiting for sensor readiness					
T_NS_SHIFT_WAIT_REP [NC_NOX_SENS_CONF]	V	0... FFFFH	0... 6553.5	0.1	[s]

Timer for repetition of NOx signal shift diagnosis					
T_NS_SHIFT_WAIT_REP_REQ [NC_NOX_SENS_CONF]	V	0... FFFFH	0... 6553.5	0.1	[s]
Timer for repetition of NOx signal shift diagnosis combustion request					
TQI_MMV_NS_SHIFT	V	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Filtered TQI_AV_signal					
TQI_MMV_NS_SHIFT_DIF_ABSV	V	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Absoult value of limited dynamic value of TQI					
VLS_DOWN_MMV_NS_SHIFT [NC_CBK_EX_NR]	V	0... FFFFH	0... 4.99992371	76.2939e-6	[V]
Filtered VLS_DOWN signal					
VLS_NS_MMV_NS_SHIFT [NC_NOX_SENS_CONF]	V	80000000... 7FFFFFFFH	-1200... 1199.99999944	558.794e-9	[mV]
Filtered VLS_NS_REF_MMV_NS_SHIFT signal					
VLS_NS_MMV_NS_SHIFT_DIF_ABSV [NC_NOX_SENS_CONF]	V	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Absoult value of difference between first and second filtered VLS_NS signal					
VLS_NS_REF_CHK_DIAG_VLD [NC_NOX_SENS_CONF]	V	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Reference VLS_NS signal for check of the diagnosis validity					
VLS_NS_REF_MMV_NS_SHIFT [NC_NOX_SENS_CONF]	V	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Filtered VLS_NS signal					
VLS_NS_REF_MMV_NS_SHIFT_MAX [NC_NOX_SENS_CONF]	V	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Maximum reference value for check validation of the NOx sensor binary signal					
VLS_NS_REF_MMV_NS_SHIFT_MIN [NC_NOX_SENS_CONF]	V	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Minimum reference value for check validation of the NOx sensor binary signal					

Input Data:

NC_NOX_SENS_CONF	T_AST_SAE	MAF	OPM_REQ_CUS
OPM_AV	VLS_DOWN [NC_CBK_EX_NR]	NC_CBK_EX_NR	LV_LAM_ADJ_ACT [NC_CBK_EX_NR]
LV_END_DIAG_NS_SHIFT [NC_NOX_SENS_CONF]	LV_RGN_NT_REQ	TQI_AV	STATE_NS [NC_NOX_SENS_CONF]
NTLD	LV_INH_DIAG_NS_OBD_2_SHIFT [NC_NOX_SENS_CONF]	DIST_NT	DIST_NT_NS_SHIFT_EXT_ADJ [NC_NOX_SENS_CONF]
LV_SENS_AFR [NC_NT_NR]	STATE_NOX	LV_SO2P_REQ	LAMB_LS_UP [NC_CBK_EX_NR]
LV_ERR_NS_SHIFT [NC_NOX_SENS_CONF]	LV_PUC	MAF_CYL	VLS_NS [NC_NOX_SENS_CONF]
TNT_MDL_MV	VS	N_32	TIA
AMP	NOX_NS_DIAG [NC_NOX_SENS_CONF]	NOX_NS_AD [NC_NOX_SENS_CONF]	LV_NS_AD_REQ
CTR_NS_SHIFT_CYC_EXT_ADJ [NC_NOX_SENS_CONF]	LAMB_NS [NC_NOX_SENS_CONF]	LV_TEMP_NS_OK [NC_NOX_SENS_CONF]	TCO_ST
TCO_ST_DC	O2L_CAT_DIAG_AFL [NC_CBK_EX_NR]	EFF_CAT_DIAG_HOM [NC_CBK_EX_NR]	NT_AGI_THERMO
DIST_NS_NEW	FCO_AV_1	FCO_AV_2	NC_CYL_NR
FCO			

Calibration Data:

Name	**Dim**	**Hex.Limits**	**Phys.Limits**	**Resol.**	**Unit**

C_AMP_MIN_NS_SHIFT_DIAG_ACT	1	0... FFFFH	0... 5434	0.0829175	[hPa]
Minimum ambient presure for activation of the diagnosis					
C_CRLC_LAMB_NS_MMV_NS_SHIFT	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation factor for LAMB_NS filtering					
C_CRLC_MAF_MMV_NS_SHIFT	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation factor for MAF filtering					
C_CRLC_TQI_MMV_NS_SHIFT	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation factor for TQI filtering					
C_CRLC_VLS_DOWN_MMV_NS_SHIFT	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation factor for VLS_DOWN filtering					
C_CRLC_VLS_NS_MMV_NS_SHIFT	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation factor for VLS_NS second filtering					
C_CRLC_VLS_NS_REF_MMV_NS_SHIFT	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation factor for VLS_NS first filtering					
C_CTR_NS_SHIFT_CYC_NOT_VLD_EXT	1	0... FFH	0... 255	1	[-]
Maximal number of additional not valid diagnosis cycles - valid at activation by service tool					
C_CTR_NS_SHIFT_CYC_NOT_VLD_MAX	1	0... FFH	0... 255	1	[-]
Maximal number of additional not valid diagnosis cycles					
C_CTR_NS_SHIFT_CYC_VLD_EXT	1	0... FFH	0... 255	1	[-]
Maximal number of additional valid diagnosis cycles - valid at activation by service tool					
C_CTR_NS_SHIFT_CYC_VLD_MAX	1	0... FFH	0... 255	1	[-]
Maximal number of additional valid diagnosis cycles					
C_CTR_THD_NS_SHIFT_DIAG_INH	1	0... FFH	0... 255	1	[-]
Minimum number of successful regenerations for activation of the NOx signal shift diagnosis					
C_DIST_NT_NS_SHIFT_MIN	1	0... FFFFH	0... 524280	8	[km]
Minimum distance between two active NOx signal diagnosis					
C_DIST_NT_NS_SHIFT_MIN_NEW	1	0... FFFFH	0... 524280	8	[km]
Minimum distance for activation first active NOx signal diagnosis					
C_EFF_CAT_MAX_DIAG_HOM_NS_SHIFT [NC_CBK_EX_NR]	1	0... FFH	0... 1.9921875	7.8125e-3	[-]
Maximum homogeneous catalyst diagnosis value for activation of the diagnosis					
C_FAC_AGI_MIN_THERMO_NS_SHIFT	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Minimum thermal aging factor of NT for activation of the diagnosis					
C_LAMB_MAX_NS_SHIFT_DIAG_ACT	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Maximum lambda threshold for activation of the NOx signal shift diagnosis					
C_LAMB_MIN_NS_SHIFT_DIAG_ACT	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Minimum lambda threshold for activation of the NOx signal shift diagnosis					
C_LAMB_NS_MMV_THD_LDC_ABSV	1	0... FFFFH	0... 31.9995117188	488.281e-6	[-]
Limited dynamic threshold for LAMB_NS signal					
C_LAMB_NS_THD_NS_SHIFT_MAX	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Maximum NOx sensor lambda signal for activation of the diagnosis					
C_LAMB_NS_THD_NS_SHIFT_MIN	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Minimum NOx sensor lambda signal for activation of the diagnosis					
C_MAF_CYL_INT_NS_SHIFT_INI	1	0... FFFFH	0... 1820.41666667	0.0277778	[g]
Initialization value of MAF integral after PUC phase					
C_MAF_MMV_THD_NS_SHIFT	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]

Threshold for limited dynamic of MAF					
C_MAF_THD_MAX_NS_SHIFT	1	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Upper MAF threshold for activation of the diagnosis					
C_MAF_THD_MIN_NS_SHIFT	1	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Lower MAF threshold for activation of the diagnosis					
C_N_32_MAX_NS_SHIFT_DIAG_ACT	1	0... FFH	0... 8160	32	[rpm]
Maximum N_32 value for activation of the diagnosis					
C_N_32_MIN_NS_SHIFT_DIAG_ACT	1	0... FFH	0... 8160	32	[rpm]
Minimum N_32 value for activation of the diagnosis					
C_NOX_MAX_NS_SHIFT_DIAG_ACT	1	FF9C... 5DCH	-100... 1500	1	[ppm]
Maximum value of NOX signal for activation of the diagnosis					
C_NTLD_THD_MAX_NS_SHIFT	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Allowed NOx trap loading grad for activation of the NOx signal shift diagnosis					
C_NTLD_THD_MAX_NS_SHIFT_ACT_REQ	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Allowed NOx trap loading degree for active NOx signal shift diagnosis					
C_NTLD_THD_MIN_NS_SHIFT_ACT_REQ	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Minimum allowed NOx trap loading degree for active NOx signal shift diagnosis					
C_NTLD_THD_NS_SHIFT_DIAG_INH	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Minimum NTLD value for recognition of long lean phase					
C_O2L_CAT_MIN_DIAG_AFL_NS_SHIFT [NC_CBK_EX_NR]	1	0... FFFFH	0... 2.61684895833	39.9306e-6	[g]
Minimum lean catalyst diagnosis value for activation of the diagnosis					
C_STATE_NS_REP_INI_NS_SHIFT	1	0... FFH	0... 255	1	[-]
Initialization of repetition timer in state DIAG					
C_STATE_NS_SHIFT_CMB_REQ	1	0... FFFFH	0... 65535	1	[-]
Combustion mode request for NOx signal internal shift diagnosis					
C_T_AFL_THD_NS_SHIFT_DIAG_INH	1	0... FFFFH	0... 6553.5	0.1	[s]
Minimum lean time since last successful regeneration for inhibition of the NOx signal shift diagnosis					
C_T_MIN_NS_SHIFT_DEAC_SO2P	1	0... FFFFH	0... 6553.5	0.1	[s]
Minimum time for inhibition of the diagnosis after desulphurization					
C_T_NS_SHIFT_CMB_INT_REQ_MAX	1	0... FFFFH	0... 6553.5	0.1	[s]
Maximum time for set of combustion request for NOx signal shift diagnosis					
C_T_NS_SHIFT_CMB_MAX_NS_CHK	1	0... FFFFH	0... 6553.5	0.1	[s]
Waiting time for valid binary NOx sensor signal					
C_T_NS_SHIFT_DEAC_TEMP_MIN	1	0... 9F6H	0... 255	0.1	[s]
Minimum time for activation of the NOx signal shift diagnosis after set of dew point					
C_T_NS_SHIFT_READY_WAIT_MAX	1	0... FFH	0... 25.5	0.1	[s]
Maximum time for waiting for NOx sensor readiness for diagnosis					
C_T_NS_SHIFT_WAIT_REP_MIN	1	0... FFFFH	0... 6553.5	0.1	[s]
Minimum time for repetition of the NOx sensor shift diagnosis					
C_T_NS_SHIFT_WAIT_REP_MIN_REQ	1	0... FFFFH	0... 6553.5	0.1	[s]
Minimum time for repetition of the HOM request for NOx sensor shift diagnosis					
C_TIA_MAX_NS_SHIFT_DIAG_ACT	1	0... FEH	-48... 142.5	0.75	[°C]
Maximum TIA value for activation of the diagnosis					
C_TIA_MIN_NS_SHIFT_DIAG_ACT	1	0... FEH	-48... 142.5	0.75	[°C]
MinimumTIA value for activation of the diagnosis					
C_TNT_THD_MAX_NS_SHIFT	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Upper TNT threshold for activation of the diagnosis					
C_TNT_THD_MIN_NS_SHIFT	1	0... FFFFH	0... 1023.984375	0.015625	[°C]
Lower TNT threshold for activation of the diagnosis					
C_TQI_MMV_THD_HOM_REQ_NS_SHIFT	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limited dynamic threshold of TQI for set of the HOM request					
C_TQI_MMV_THD_NS_SHIFT	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Threshold for limited dynamic of TQI_AV					

C_VLS_DOWN_MAX_NS_SHIFT [NC_CBK_EX_NR]	1	0... FFFFH	0... 4.99992371	76.2939e-6	[V]
Maximum threshold for VLS_DOWN for activation of the function					
C_VLS_DOWN_MIN_NS_SHIFT [NC_CBK_EX_NR]	1	0... FFFFH	0... 4.99992371	76.2939e-6	[V]
Maximum threshold for VLS_DOWN for activation of the function					
C_VLS_NS_MMV_THD_LDC_ABSV	1	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Limited dynamic threshold for LAMB_NS signal					
C_VLS_NS_THD_CMB_REQ_NS_SHIFT	1	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Threshold of binary NOx sensor signal for re-activation of the combustion request					
C_VLS_NS_THD_NS_SHIFT_EXT	1	0... 578H	-200... 1200	1	[mV]
Maximum threshold for VLS_NS for activation of the diagnosis, for external manual activation					
C_VLS_NS_THD_NS_SHIFT_MAX	1	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Maximum threshold for VLS_NS for activation of the diagnosis					
C_VLS_NS_THD_NS_SHIFT_MIN	1	8000... 7FFFH	-1200... 1199.96337891	0.0366211	[mV]
Minimum threshold for VLS_NS for activation of the diagnosis					
C_VS_MAX_NS_SHIFT_DIAG_ACT	1	0... FFH	0... 255	1	[km/h]
Maximum VS value for activation of the diagnosis					
C_VS_MIN_NS_SHIFT_DIAG_ACT	1	0... FFH	0... 255	1	[km/h]
Minimum VS value for activation of the diagnosis					
IP_FCO_DELTA_NS_SHIFT_MAX	6	0... FFFFFFFFH	0... 4.29497e+9	1	[µl]
LDP_FCO_IP_FCO_DELTA_NS_SHIFT	6	0... FFFFFFFFH	0... 4.29497e+9	1	[µl]
Maximal allowed additional fuel consumption for NOx signal shift diagnosis					
IP_MAF_CYL_INT_NS_SHIFT_INI_DYN	4	0... FFFFH	0... 1820.42	0.0277778	[g]
LDP_N_32_IP_MAF_CYL_INT_SHIFT	4	0... FFH	0... 8160	32	[rpm]
Initalization value of MAF integral for diagnosis activation due to limited dynamics					
IP_T_AST_MIN_NS_SHIFT	6	0... FFFFH	0... 65535	1	[s]
LDP_TCO_ST_IP_T_AST_MIN_SHIFT	6	0... FEH	-48... 142.5	0.75	[°C]
Minimum time after engine start for function activation					
IP_T_NS_SHIFT_CMB_INT_REQ_SUM	6	0... FFFFH	0... 6553.5	0.1	[s]
LDP_T_AST_SAE_IP_T_NS_SHIFT	6	0... FFFFH	0... 65535	1	[s]
Inhibition of combustion request due to total request time					
IP_VLS_NS_THD_DELTA_MAX_MMV	4	0... FFFFH	-1200... 1199.96337891	0.0366211	[mV]
LDPM_VLS_NS_IP_VLS_NS_VLD	4	0... FFFFH	-1200... 1199.96337891	0.0366211	[mV]
Threshold to check difference between maximum and actual value of binary NOx sensor signal					
IP_VLS_NS_THD_DELTA_MAX_REF	4	0... FFFFH	-1200... 1199.96337891	0.0366211	[mV]
LDPM_VLS_NS_IP_VLS_NS_VLD	4	0... FFFFH	-1200... 1199.96337891	0.0366211	[mV]
Threshold to check difference between maximum and reference of binary NOx sensor signal					
IP_VLS_NS_THD_DELTA_MIN_REF	4	0... FFFFH	-1200... 1199.96337891	0.0366211	[mV]
LDPM_VLS_NS_IP_VLS_NS_VLD	4	0... FFFFH	-1200... 1199.96337891	0.0366211	[mV]
Threshold to check difference between reference and minimum of binary NOx sensor signal					
LC_DIST_NT_SHIFT_DIAG_ACT	1	0... 1H	0... 1	1	[-]
Activation of saving of distance value between diagnosis at writing of SAE values					
LC_LAM_ADJ_ACT_NS_SHIFT [NC_CBK_EX_NR]	1	0... 1H	0... 1	1	[-]
Manual deactivation of check of trim controller					
LC_NOX_SIG_SWI	1	0... 1H	0... 1	1	[-]
Manual choice of NOX signal for activation					
LC_NS_AD_REQ_MAN_DEAC	1	0... 1H	0... 1	1	[-]

Manual deactivation of inhibition during NOx signal gain adaptation					
LC_NS_SHIFT_SET_HOM_REQ	1	0... 1H	0... 1	1	[-]
Activation of keeping of the HOM request independently on it set conditions					
LC_SWI_REP_ENA_DIAG_END	1	0... 1H	0... 1	1	[-]
Manual deactivation of check of readiness flag of the NOx signal shift OBDII diagnosis					
LC_SWI_REP_ENA_NS_SHIFT	1	0... 1H	0... 1	1	[-]
Manual reinitialization of the finished diagnosis					
LC_TCO_SWI_NS_SHIFT	1	0... 1H	0... 1	1	[-]
Switch between TCO_ST and TCO_ST_DC					

Export Actions:

ACTION_NOXD_CleanNSShiftAdapt()
Initialization of NOx signal shift diagnosis values
ACTION_NOXD_EndNNShiftDiag()
Manual end of the NOx signal Shift Diagnosis
ACTION_NOXD_StartNNShiftDiag()
Manual initialization of the NOx signal Shift Diagnosis
ACTION_NOXD_WriteNSShiftDiagExtAdj()
Write external adjust values for NOx Sensor shift diagnosis

Description for Actions

ACTION_NOXD_CleanNSShiftAdapt()
Initialization of NOx signal shift diagnosis values

ACTION_NOXD_EndNNShiftDiag()
Manual end of the NOx signal Shift Diagnosis

ACTION_NOXD_StartNNShiftDiag()
Manual initialization of the NOx signal Shift Diagnosis

ACTION_NOXD_WriteNSShiftDiagExtAdj()
Write external adjust values for NOx Sensor shift diagnosis

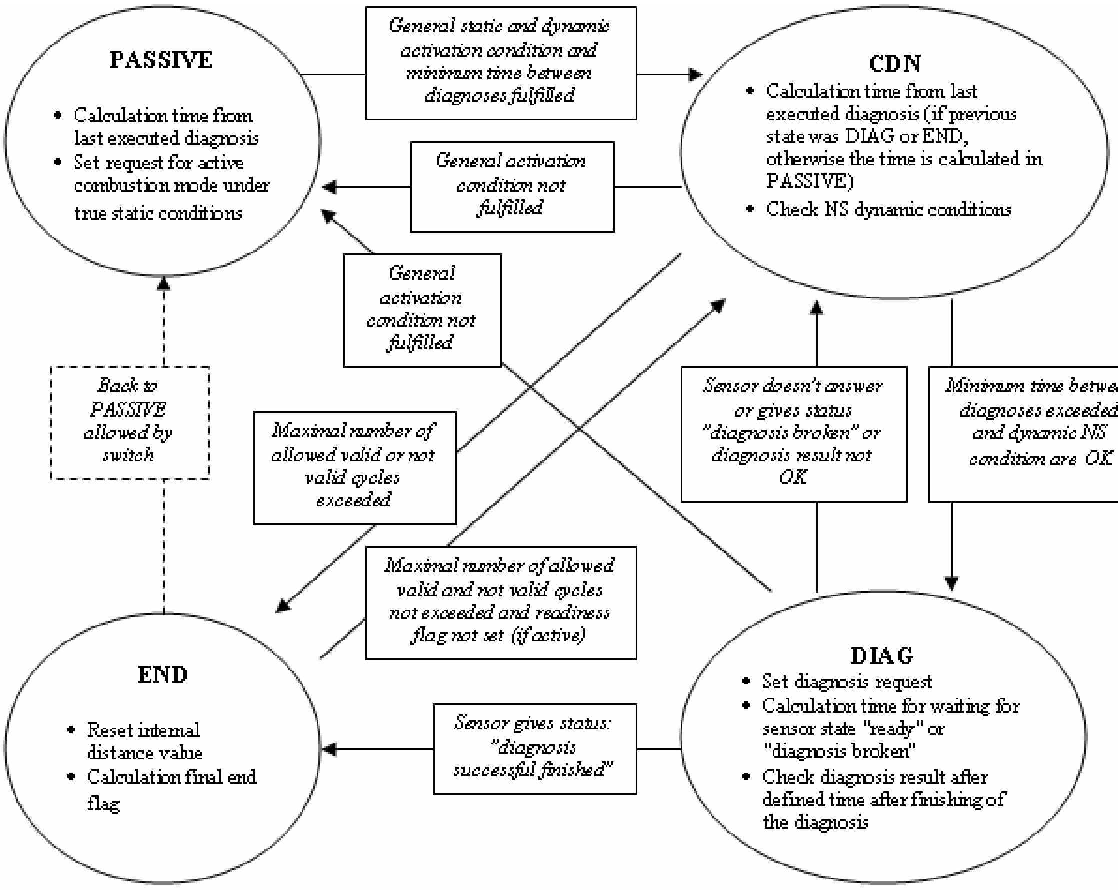
General Information

Introduction of NOx signal internal setpoint shift diagnosis.
When activation conditions are fulfilled, then a request for the NOx signal shift is generated. In this case, via modification of NOx sensor internal signals the diagnosis is executed. After the NOx sensor delivered it status, the diagnosis is either finished or paused, depending on allowed length.

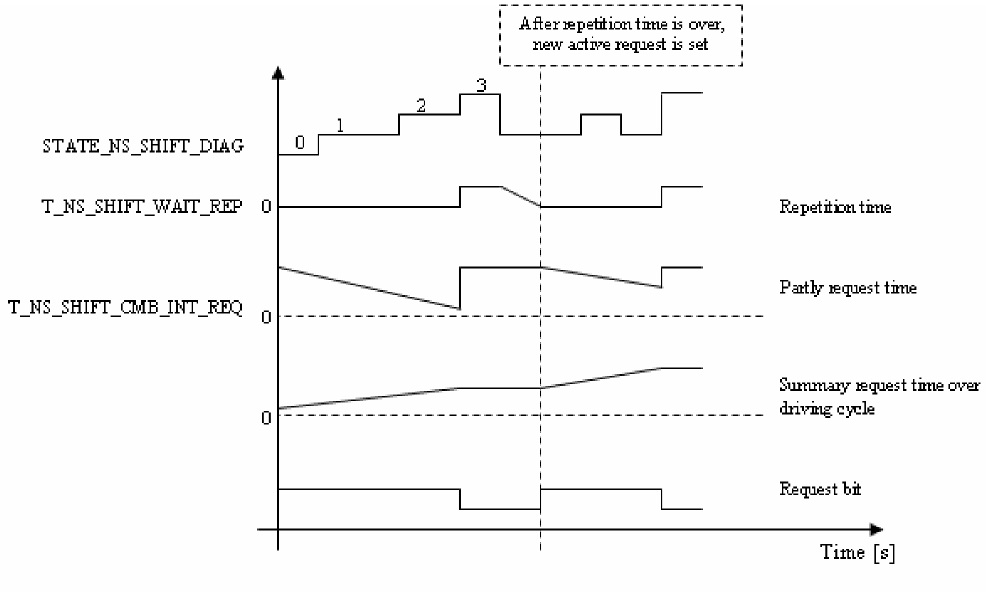
Description of the STATE_NS_SHIFT_DIAG actions:
PASSIVE: check of activation condition, especially combustion state
CDN: if the combustion conditions are fulfilled, switch to DIAG state is done after minimum time between two diagnoses. In case of executed maximum of valid or not valid diagnoses, switch to END state is done.
DIAG: if the combustion conditions are fulfilled, request for the diagnosis is sent to the NOx sensor. If the sensor doesn't confirm it readiness, transition to the CDN is performed and counter for not valid cycle calculated. In case of valid diagnosis cycle, transition to the END state is done

END: if maximum number of not valid or valid diagnoses is not exceeded, transition to the CDN state is done. In all other cases (except manual switch), no any transition is done and the diagnosis is finally finished

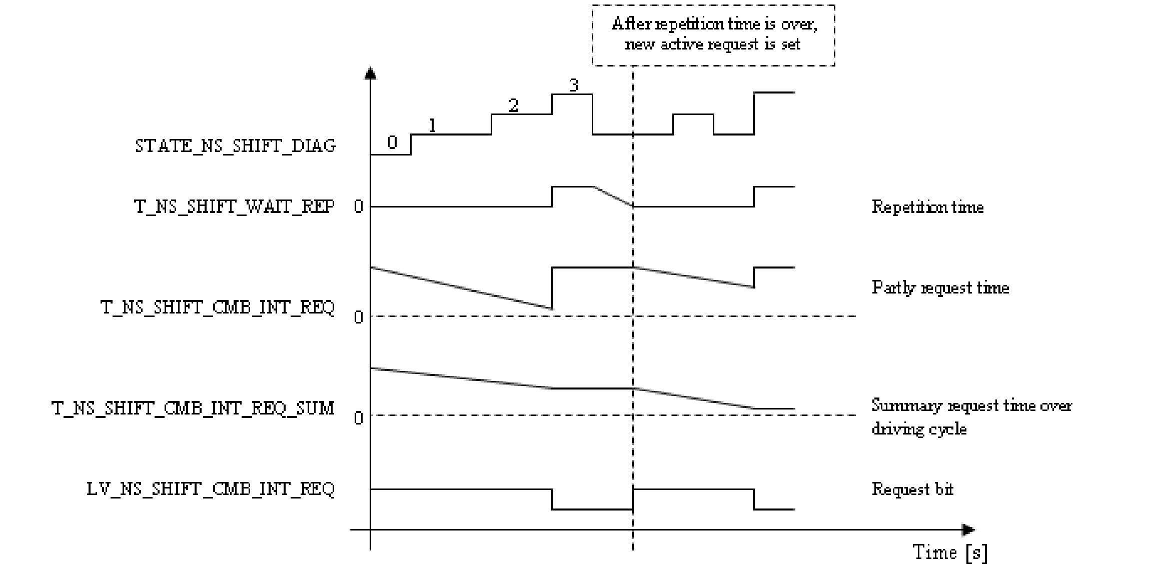
Short overview of states:



Short overview of valid and not valid counters:



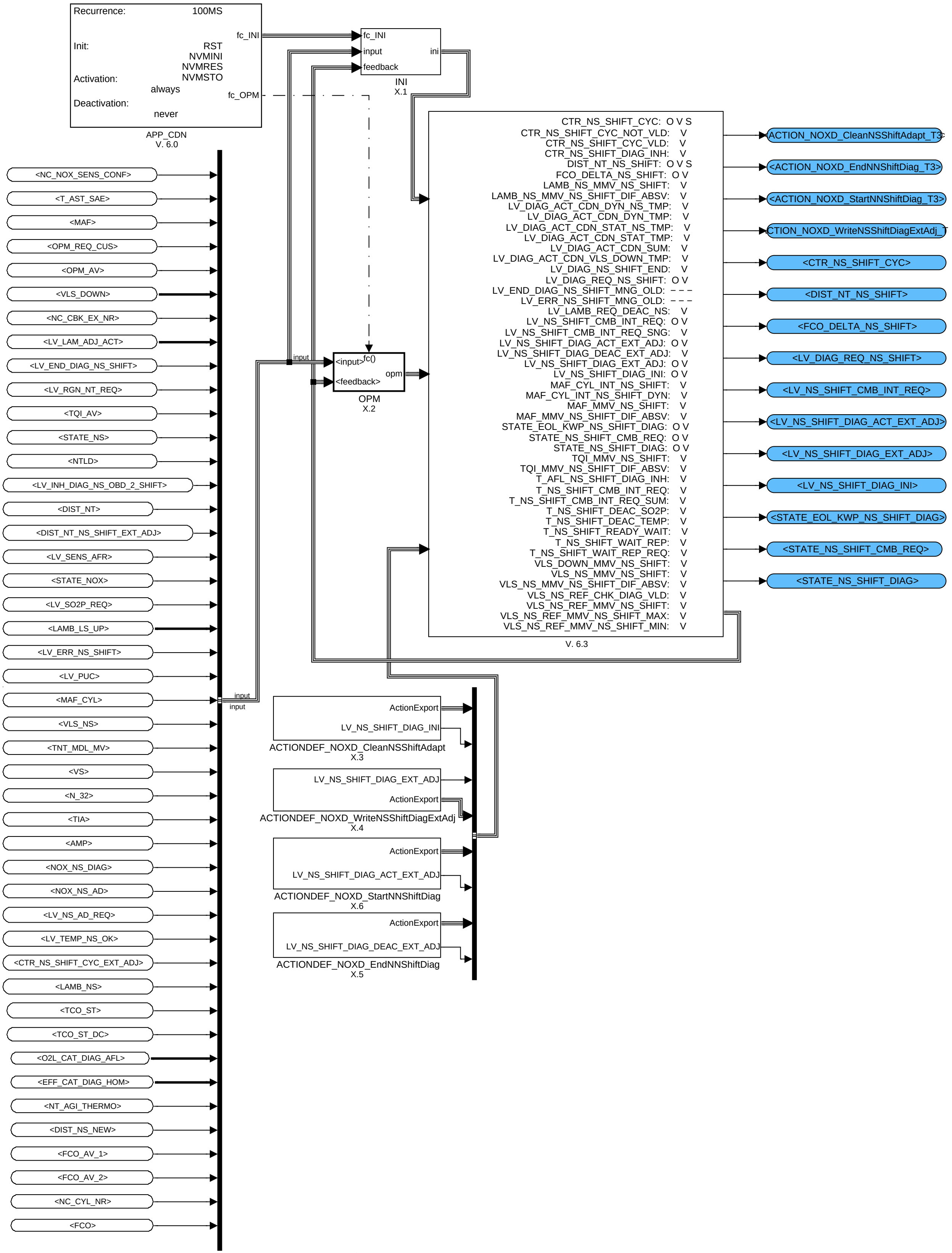
Short overview of timers:



Application Conditions

Initialization: RST, NVMINI, NVMRES, NVMSTO
Recurrence: 100MS
Activation: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.2.7 23−Apr−2007

Figure 142:

52.19.1 INITIALIZATION

52.19.1.1 Initialization at RESET

All variables except listed below are initialized by "0"

52.19.1.1.1 List of variables which are not initialized by "0"

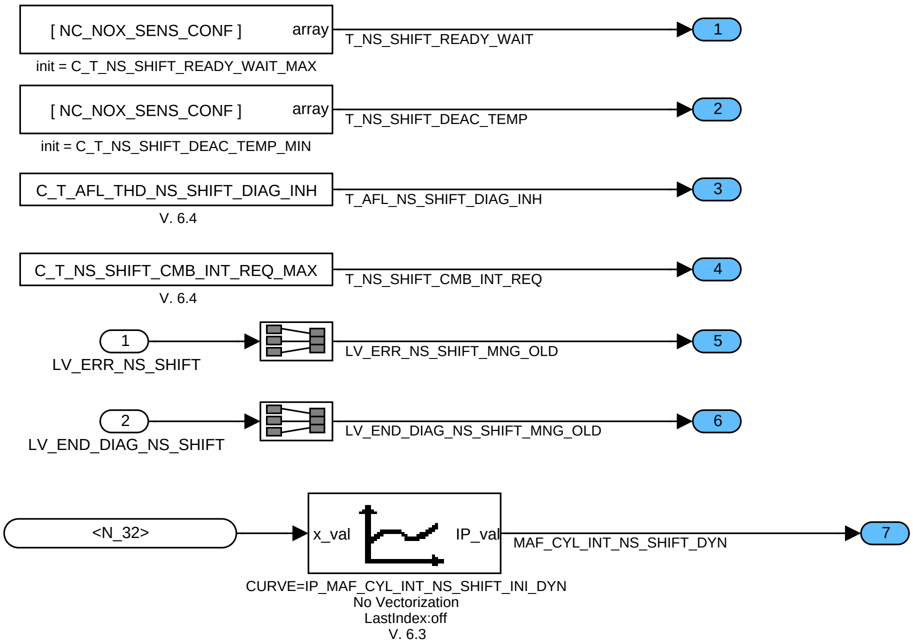


Figure 143:

52.19.1.2 Initialization of NVMY data at memory fault

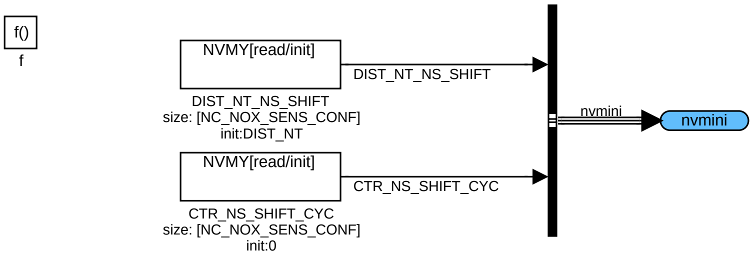


Figure 144:

52.19.2 Formula section

52.19.2.1 General activation condition

52.19.2.1.1 Check of loading degree of the NOx storage catalyst after long lean phase

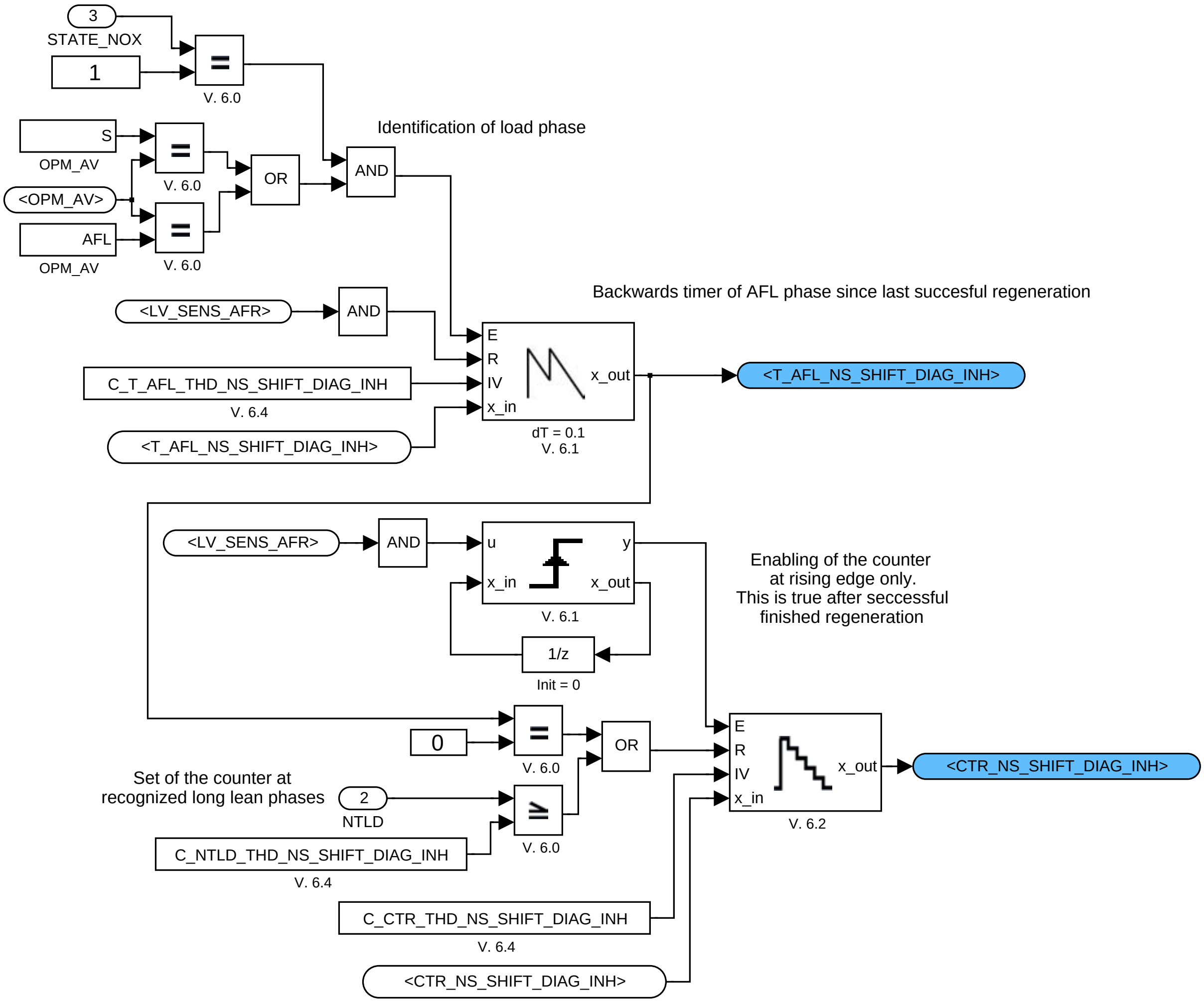
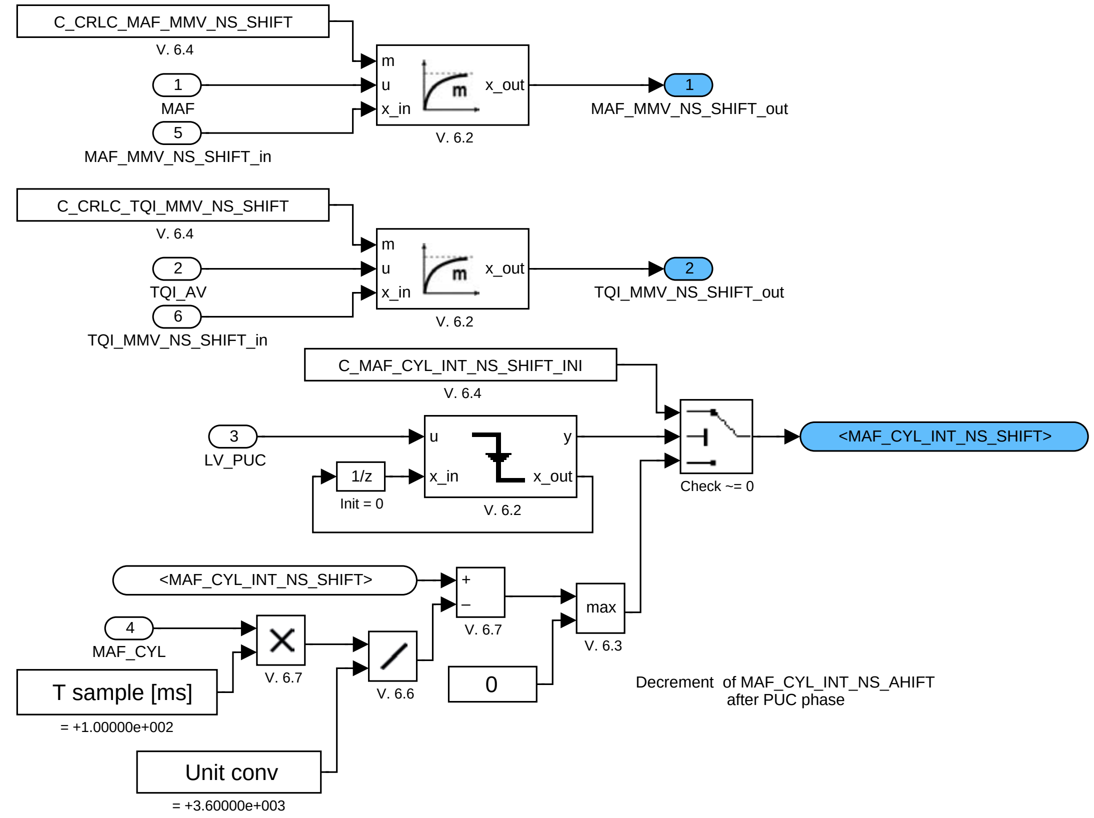


Figure 145:

52.19.2.1.2 Calculation of limited dynamics



Decrement of MAF_CYL_INT_NS_AHIFT after PUC phase

Sample time is represented in [ms]. Unit conversion from [kg_h] to [g_ms]:
1 [kg_h] = 1000_3600000 [g_ms] = 1_3600 [g_ms]

Figure 146:

52.19.2.1.3 Calculation of timer after desulphurization

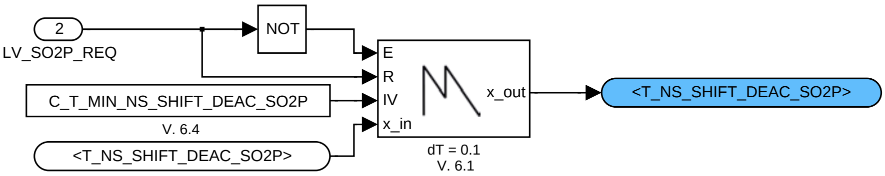


Figure 147:

52.19.2.1.4 Check static conditions

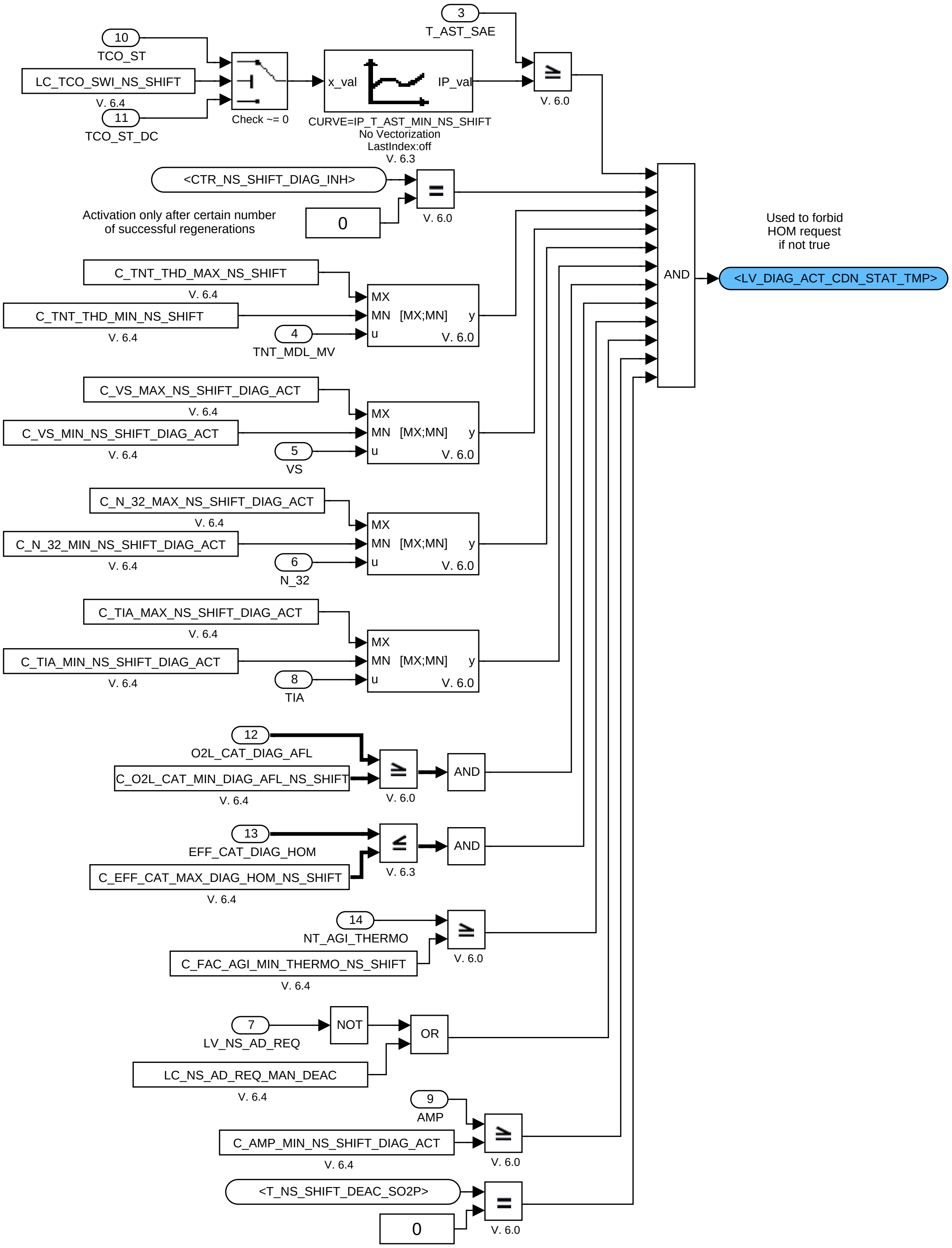


Figure 148:

52.19.2.1.5 Dynamic conditions

52.19.2.1.5.1 Calculation dynamic conditions

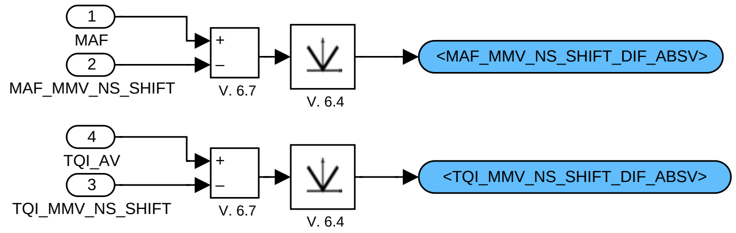


Figure 149:

52.19.2.1.5.2 Check dynamic conditions

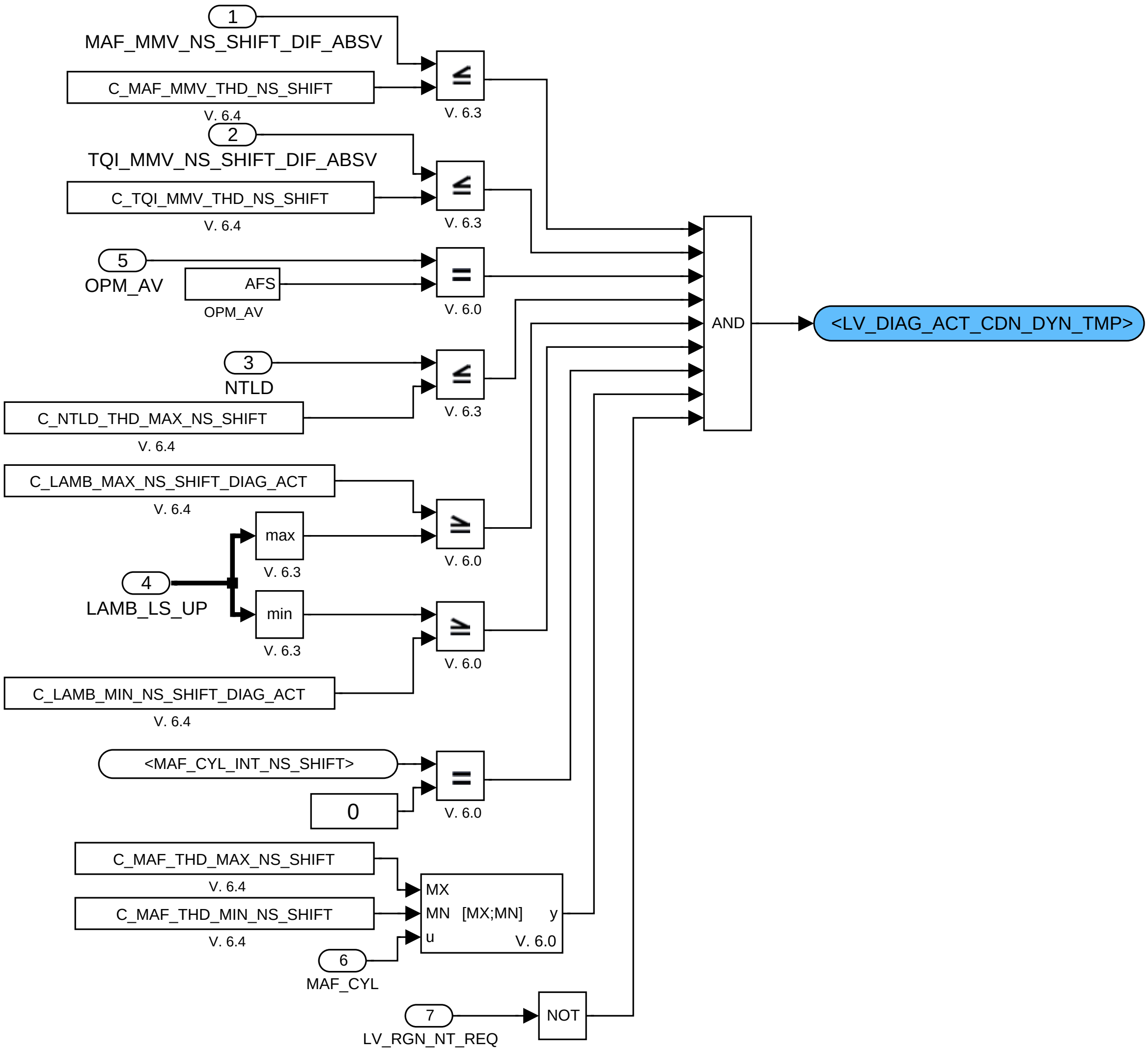


Figure 150:

52.19.2.1.6 Introduction multiple bank system for check binary lambda signal

52.19.2.1.6.1 Check binary lambda signal after pre-catalyst

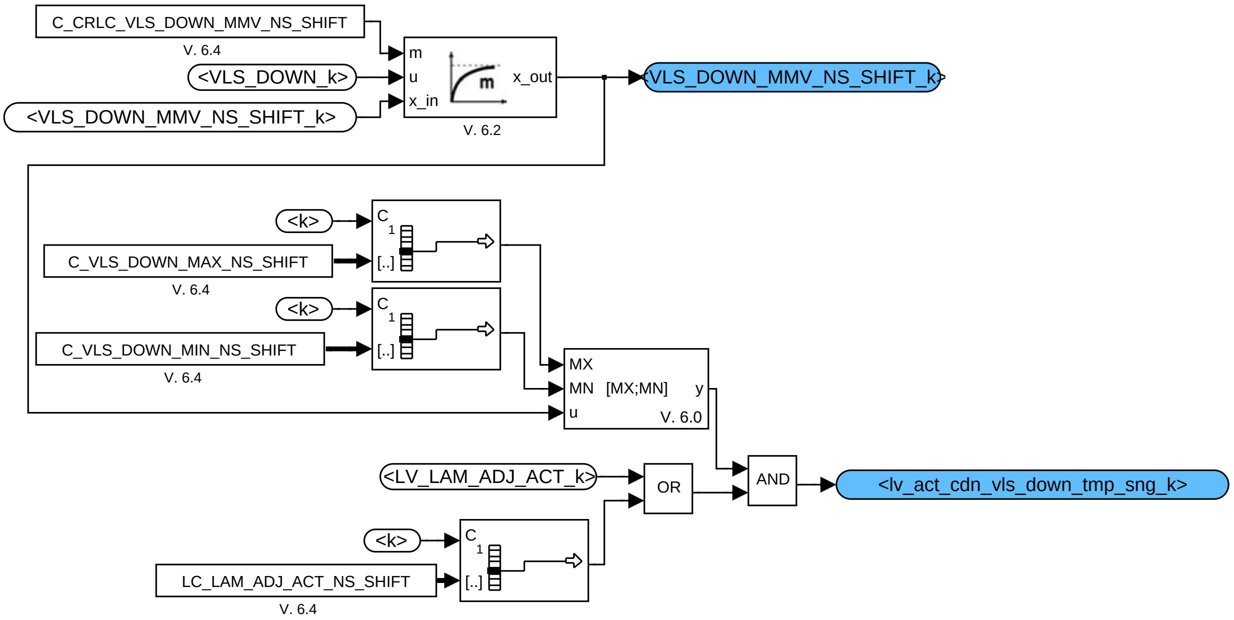


Figure 151:

52.19.2.1.7 Check activation due VLS_DOWN signal over all exhaust banks

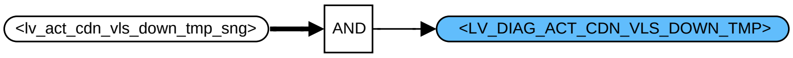


Figure 152:

52.19.2.2 Introduction multiple sensor system

52.19.2.2.1 Summary activation flag

52.19.2.2.1.1 Check static and dynamic conditions of NOx sensor signal

52.19.2.2.1.1.1 Static conditions

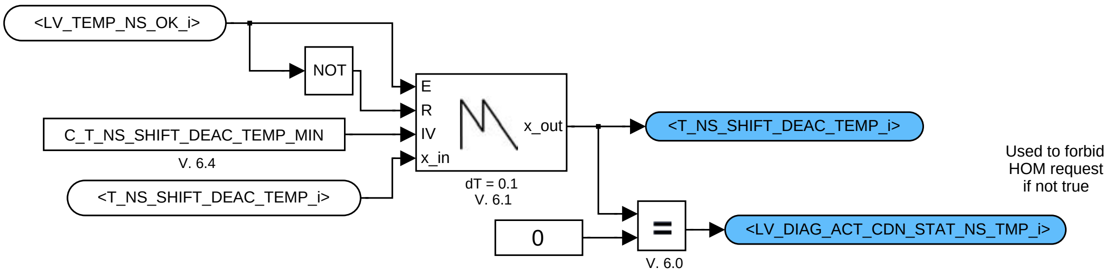


Figure 153:

52.19.2.2.1.1.2 Filtering of the VLS_NS signal

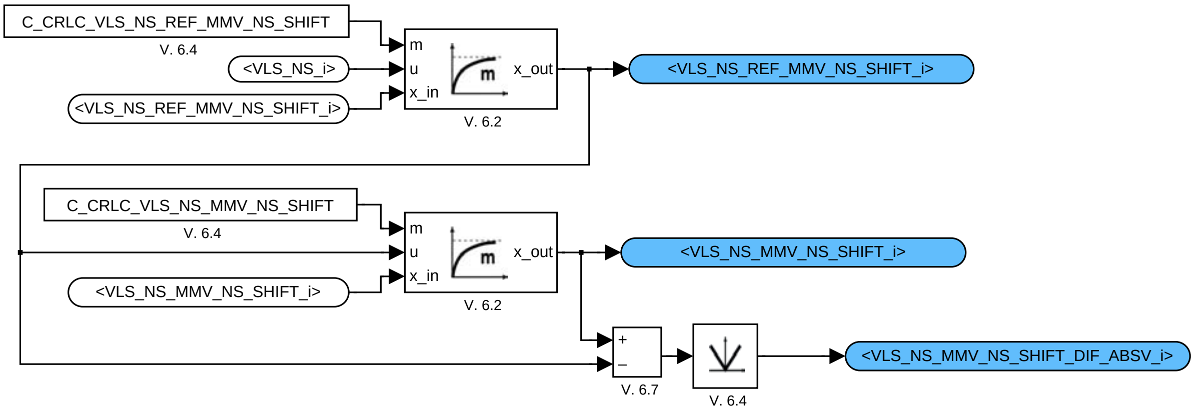


Figure 154:

52.19.2.2.1.1.3 Dynamic of NOx sensor lambda signal

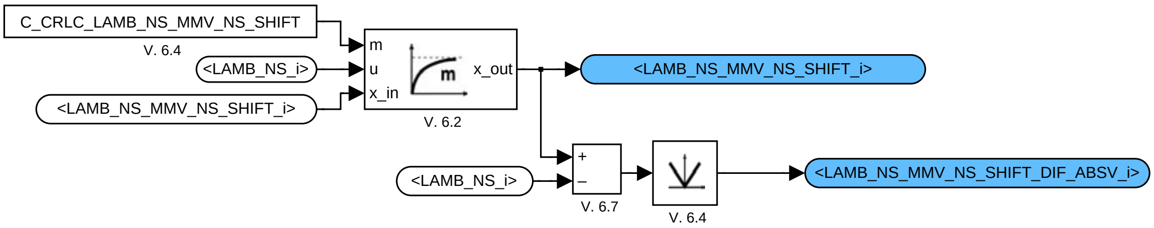


Figure 155:

52.19.2.2.1.1.4 Calculation MAF integral at fulfilled dynamic conditions

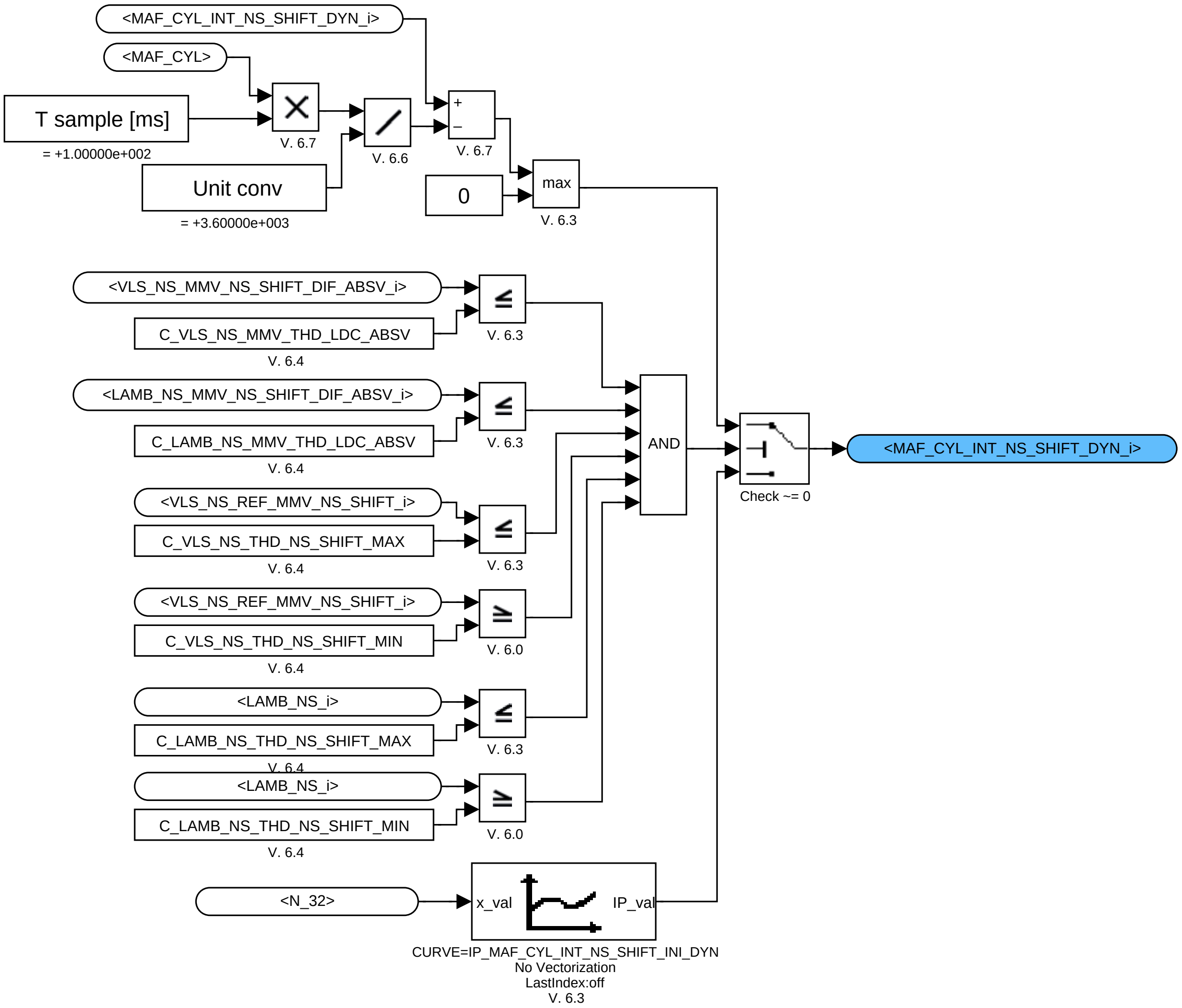


Figure 156:

52.19.2.2.1.1.5 Calculation of the repetition timer

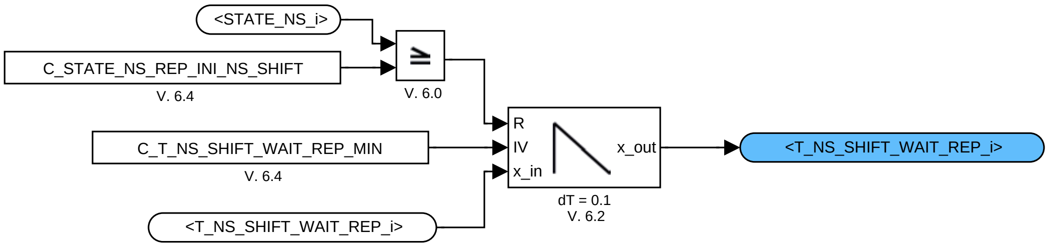


Figure 157:

52.19.2.2.1.1.6 Final dynamic conditions

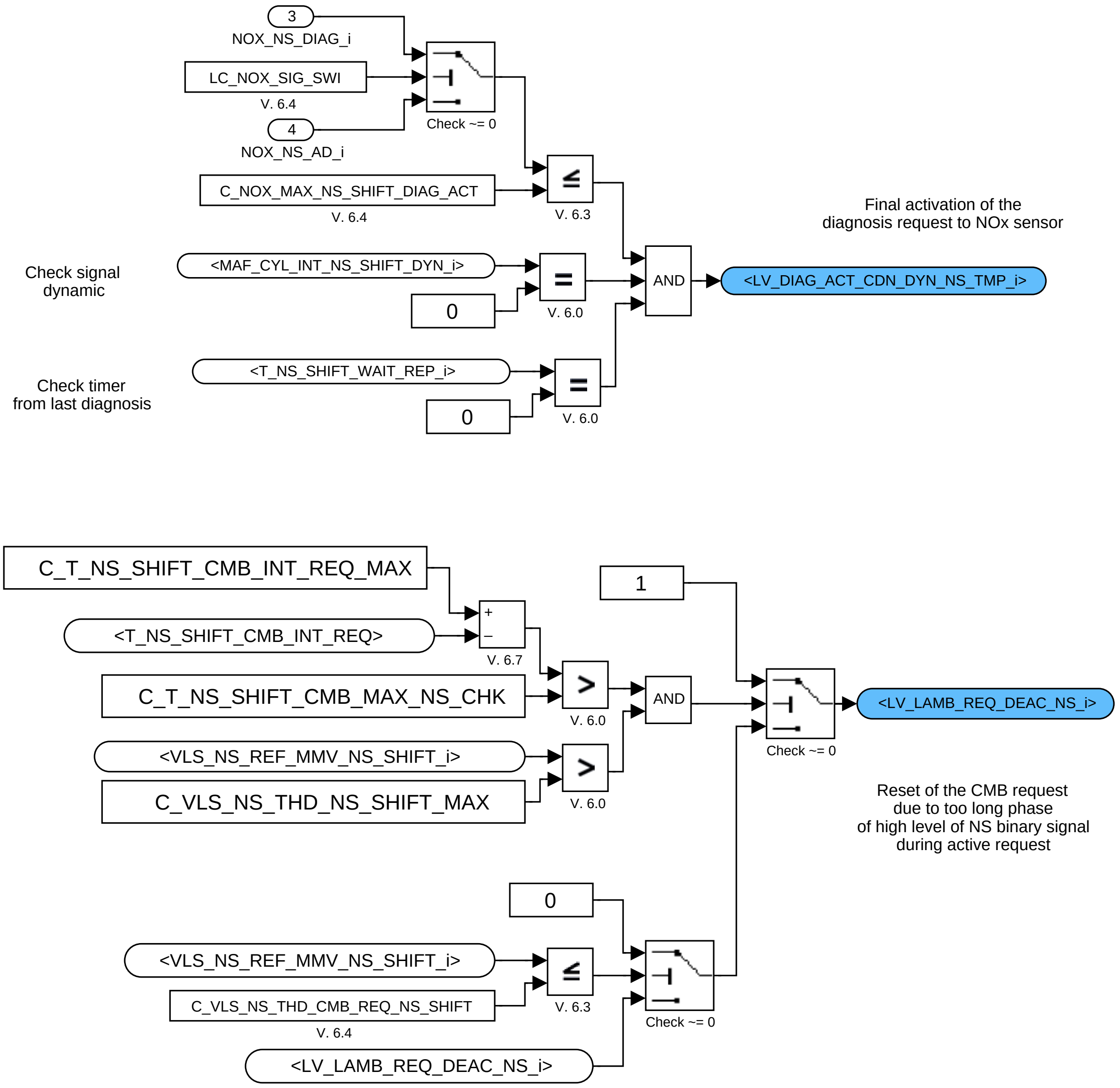


Figure 158:

52.19.2.2.1.1.7 Calculation of additional fuel consumption for activation of combustion request

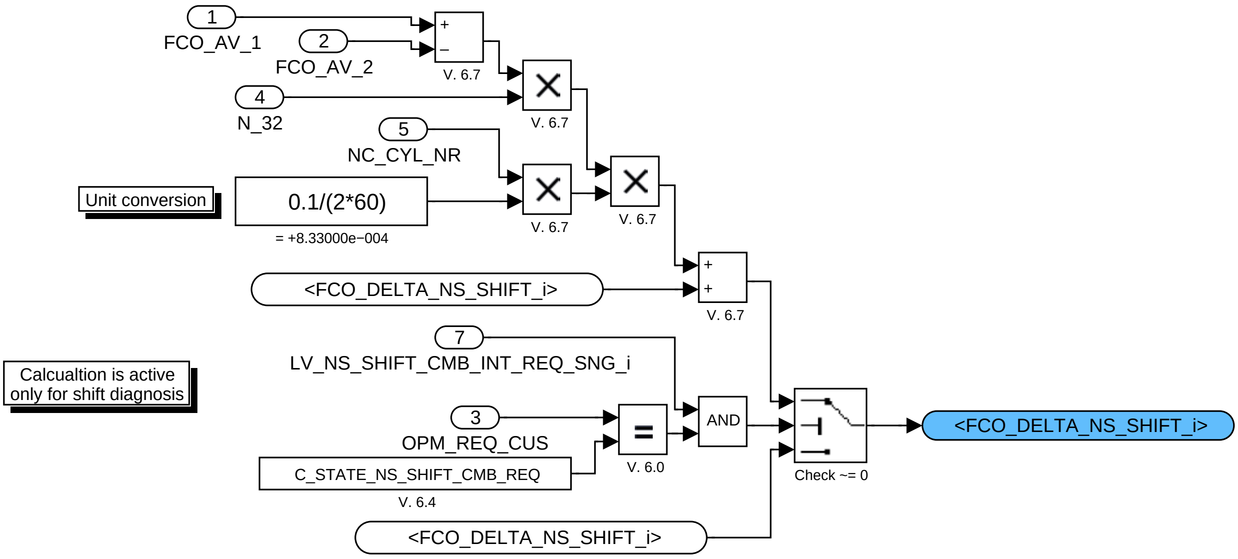


Figure 159:

52.19.2.2.1.2 Check summary activation conditions

Parallel to the common activation condition, the diagnosis can be activated by service tool

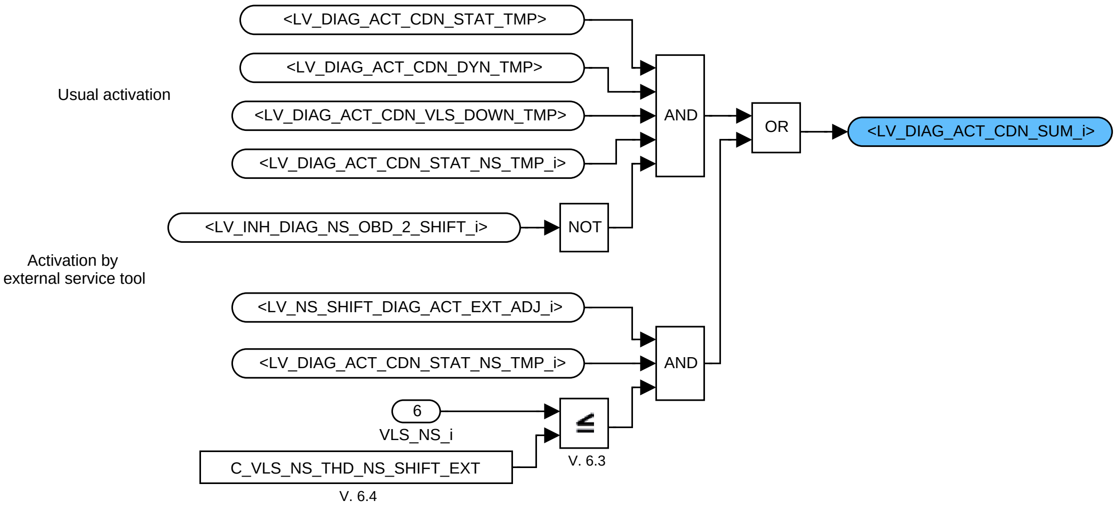


Figure 160:

52.19.2.2.2 Check transition conditions of STATE_NS_SHIFT_DIAG

52.19.2.2.2.1 From PASSIVE

52.19.2.2.2.1.1 Check transition conditions from PASSIVE

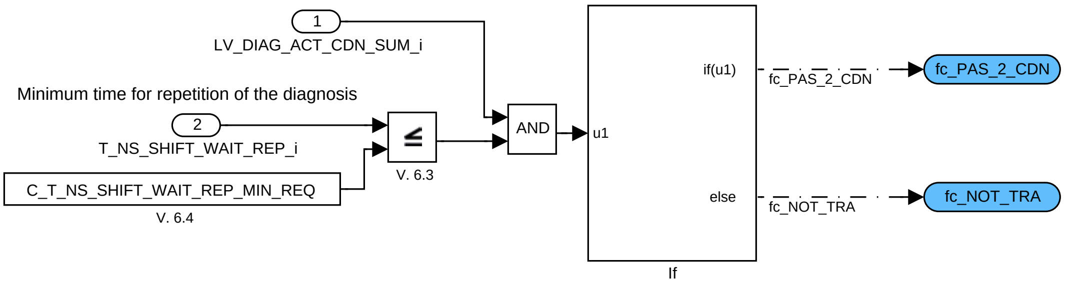


Figure 161:

52.19.2.2.2.1.2 Transition PASSIVE to CDN - actions

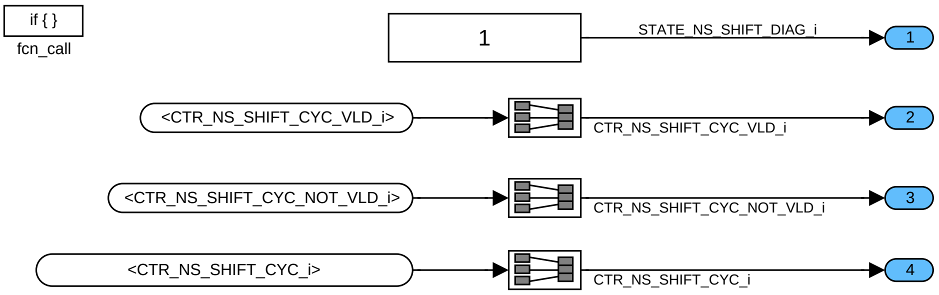


Figure 162:

52.19.2.2.2.1.3 Actions at no transition from PASSIVE

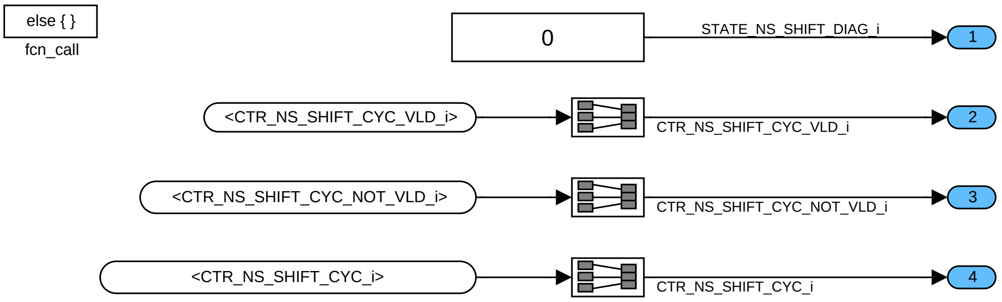


Figure 163:

52.19.2.2.2.2 From CDN

52.19.2.2.2.2.1 Check transition conditions from CDN

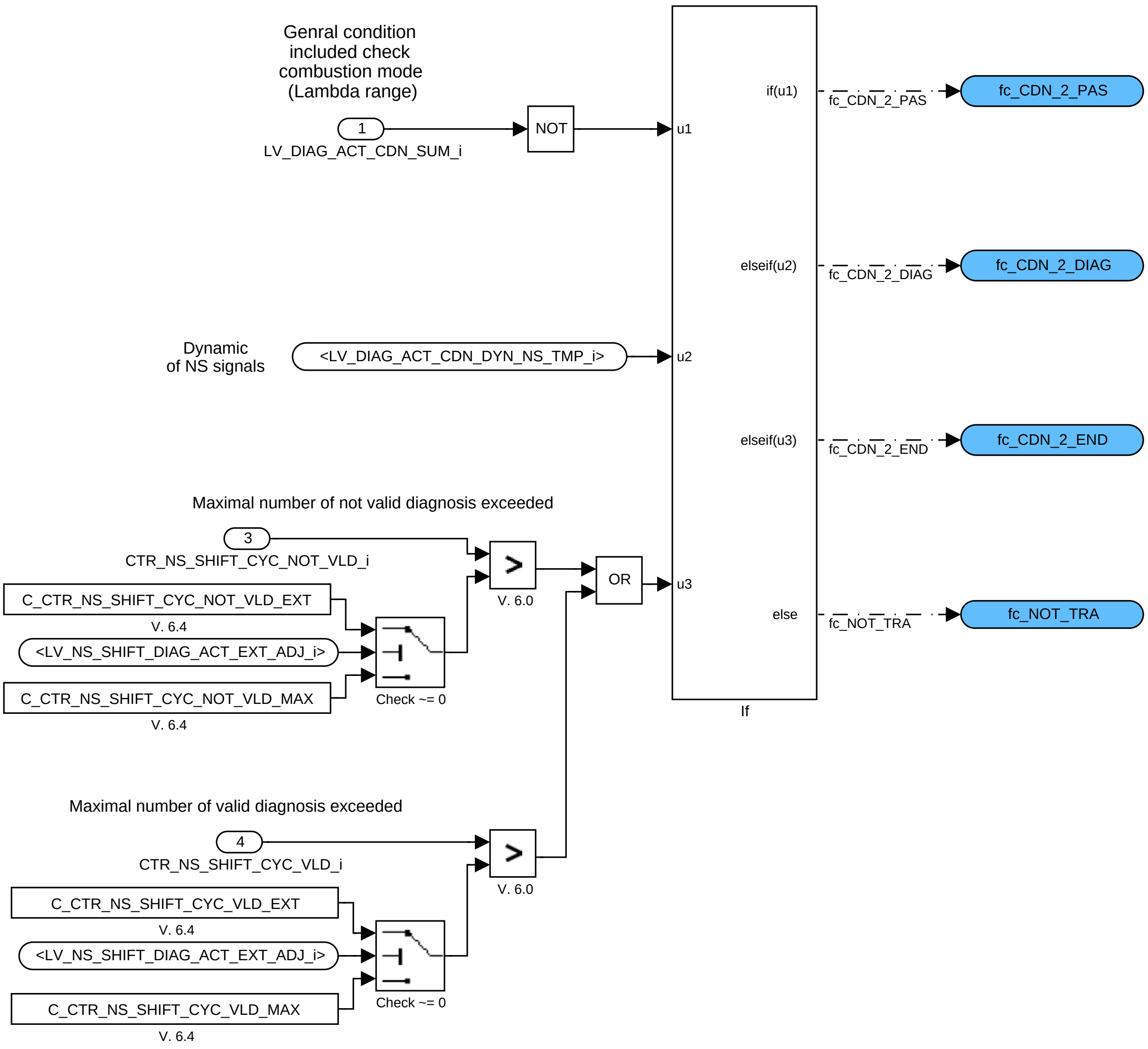


Figure 164:

52.19.2.2.2.2.2 Transition CDN to PASSIVE - actions

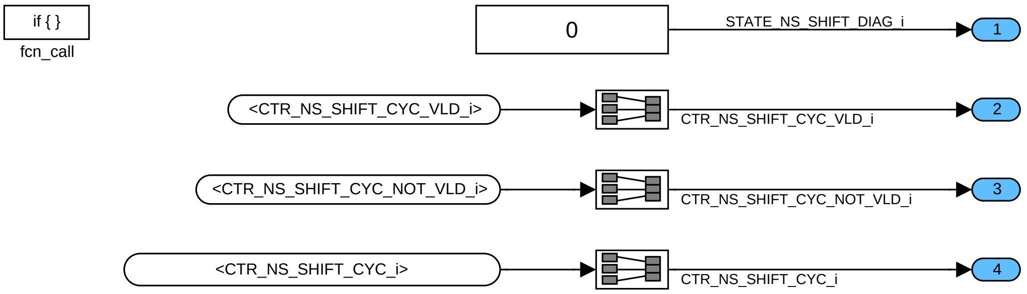


Figure 165:

52.19.2.2.2.2.3 Transition CDN to DIAG - actions

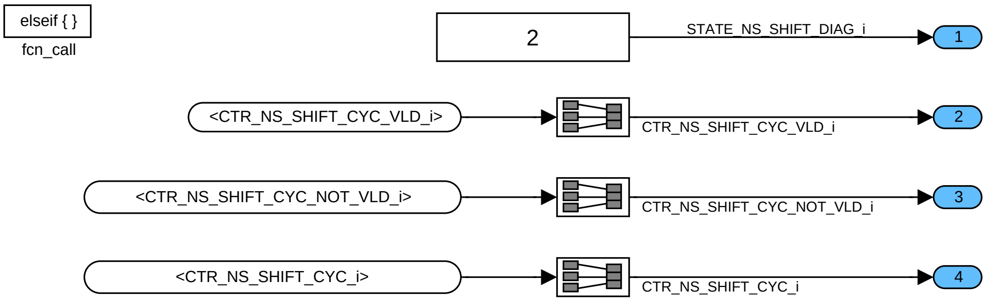


Figure 166:

52.19.2.2.2.2.4 Transition CDN to END - actions

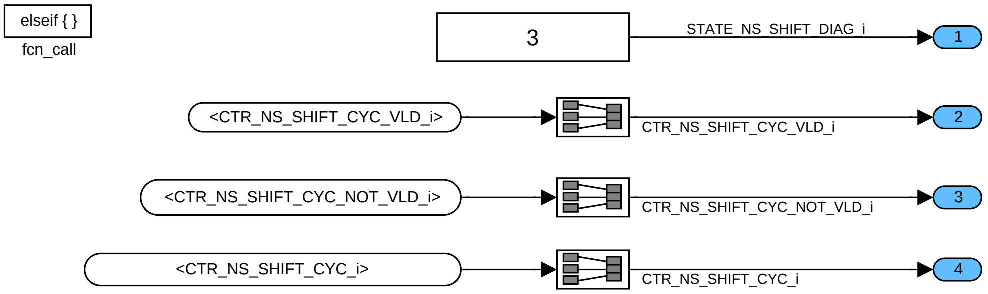


Figure 167:

52.19.2.2.2.2.5 Actions at no transition from CDN

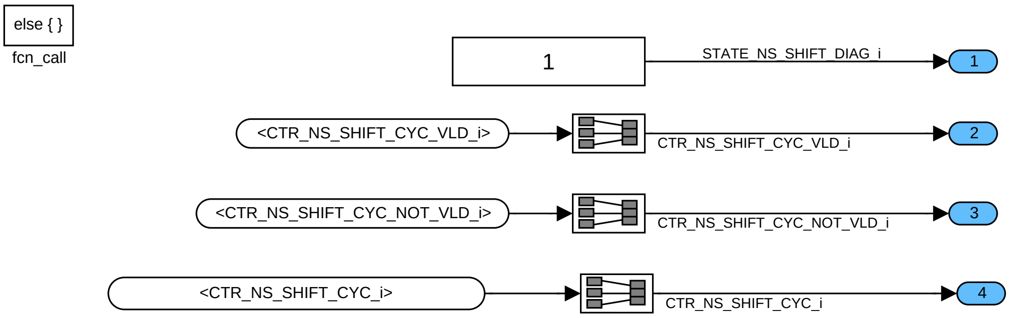


Figure 168:

52.19.2.2.2.3 From DIAG

52.19.2.2.2.3.1 Check transition conditions from DIAG

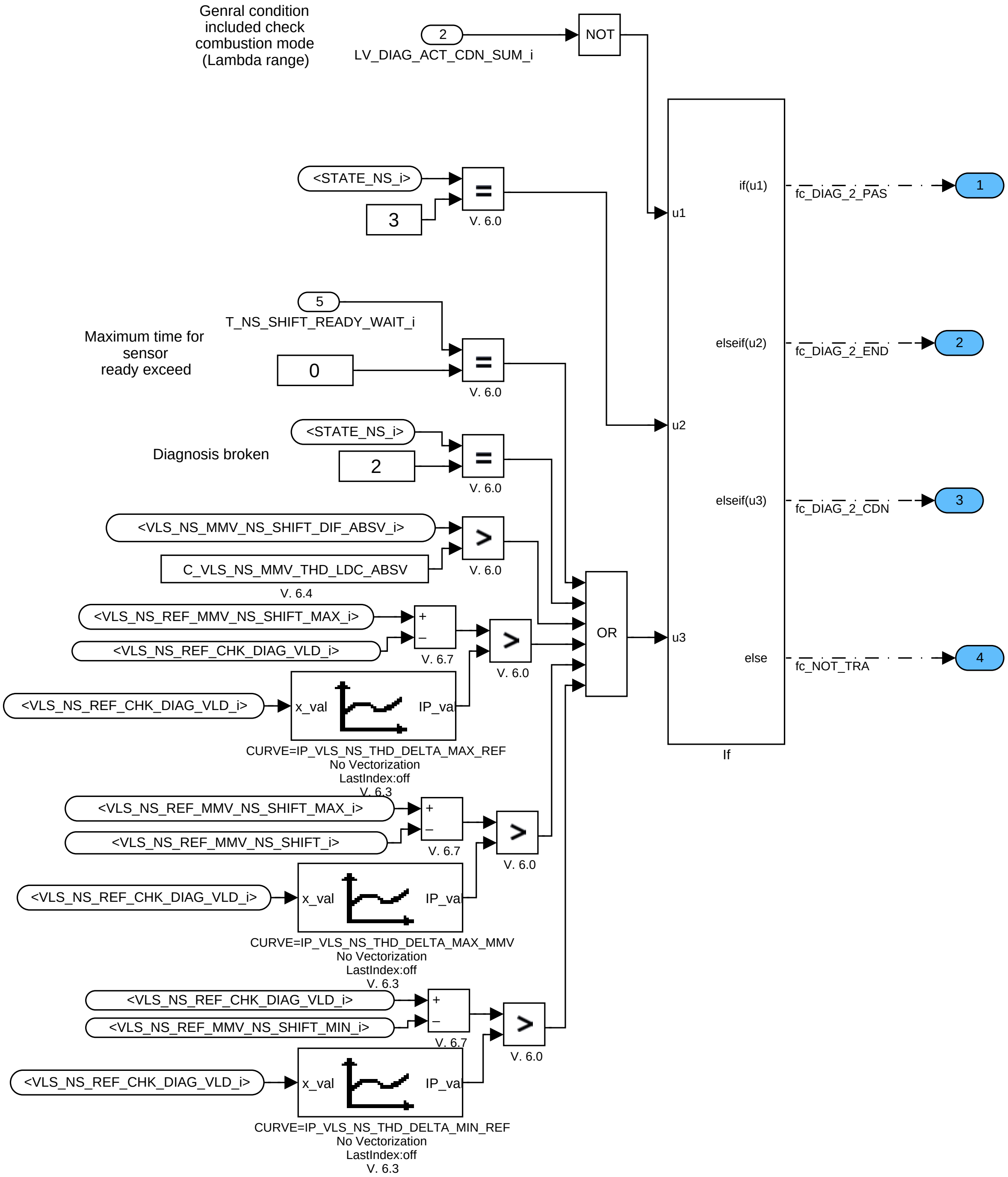


Figure 169:

52.19.2.2.2.3.2 Transition DIAG to PASSIVE - actions

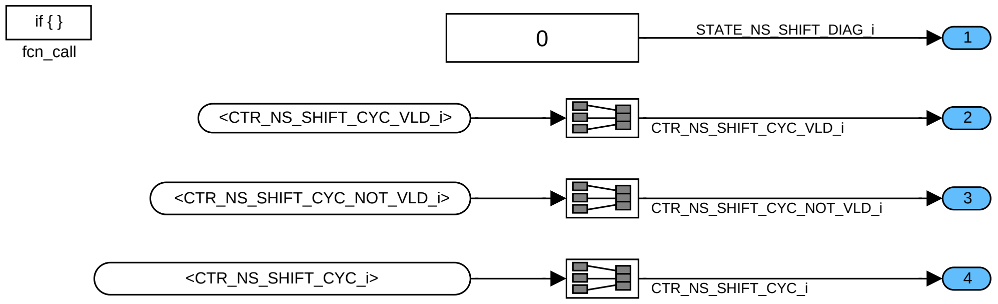


Figure 170:

52.19.2.2.2.3.3 Transition DIAG to CDN - actions

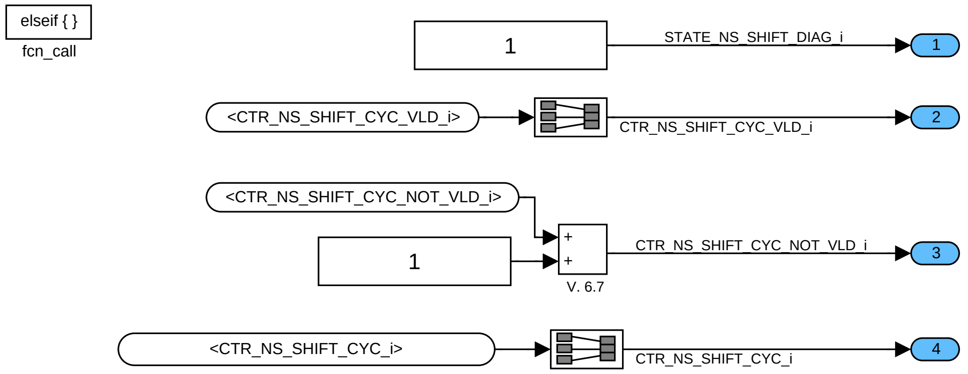


Figure 171:

52.19.2.2.2.3.4 Transition DIAG to END - actions

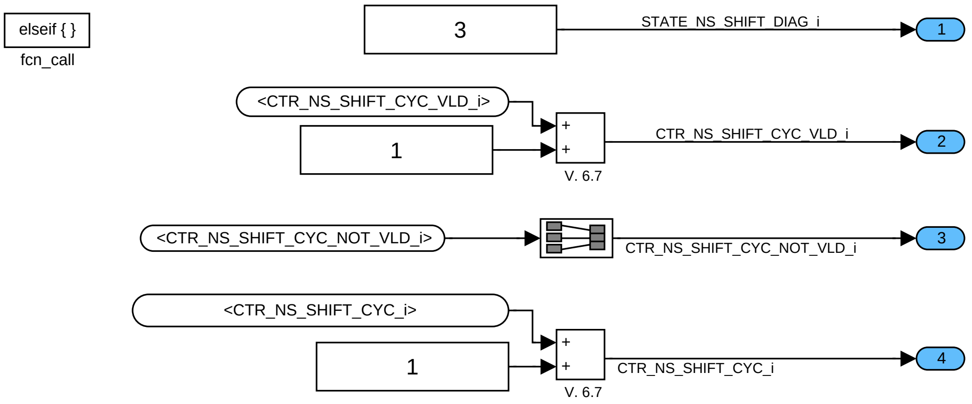


Figure 172:

52.19.2.2.2.3.5 Actions at no transition from DIAG

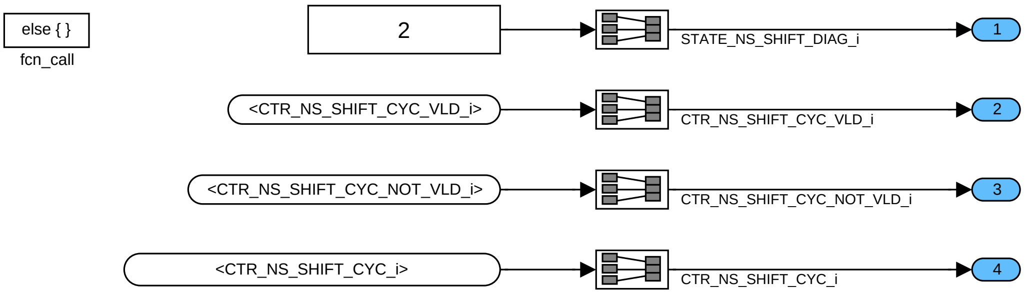


Figure 173:

52.19.2.2.2.4 From END

52.19.2.2.2.4.1 Check transition conditions from END

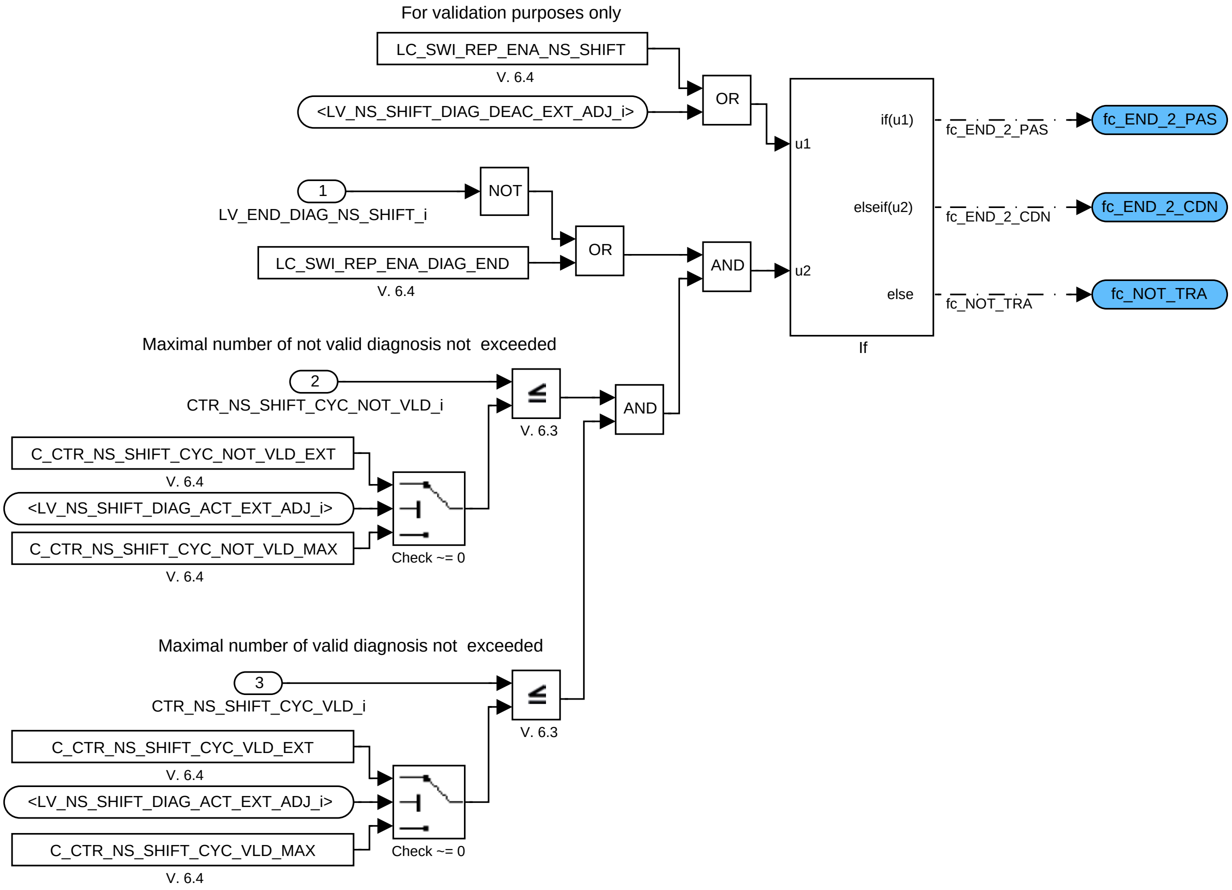


Figure 174:

52.19.2.2.2.4.2 Transition END to PASSIVE - actions

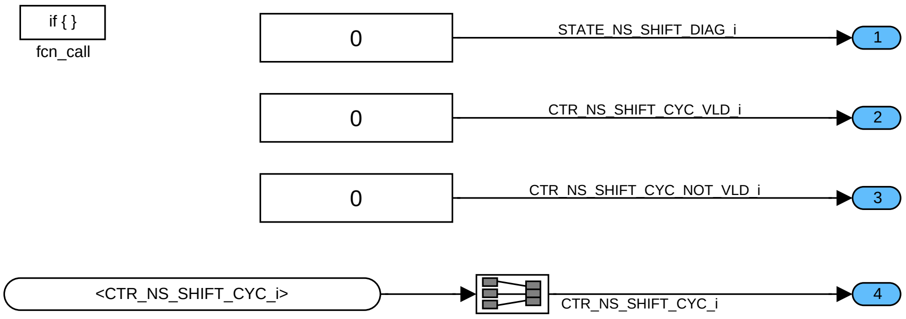


Figure 175:

52.19.2.2.2.4.3 Transition END to CDN - actions

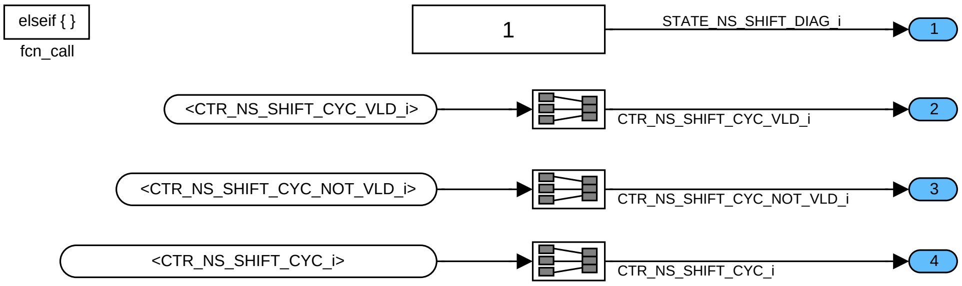


Figure 176:

52.19.2.2.2.4.4 Actions at no transition from END

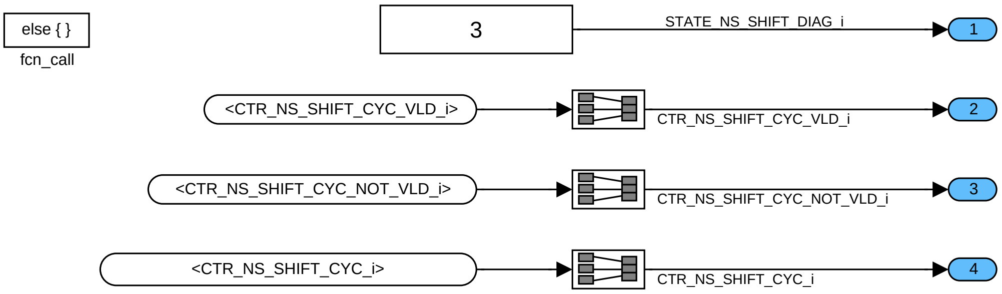


Figure 177:

52.19.2.2.3 Actions calculation of STATE_NS_SHIFT_DIAG

52.19.2.2.3.1 State PASSIVE

52.19.2.2.3.1.1 Calculation request of homogeneous mode

52.19.2.2.3.1.1.1 Combustion request repetition timer

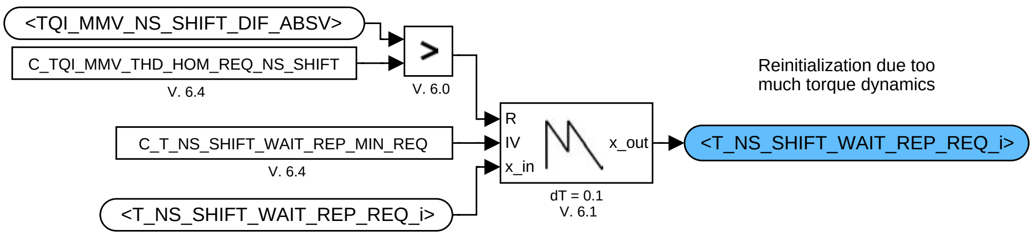


Figure 178:

52.19.2.2.3.1.1.2 Check condition for combustion request

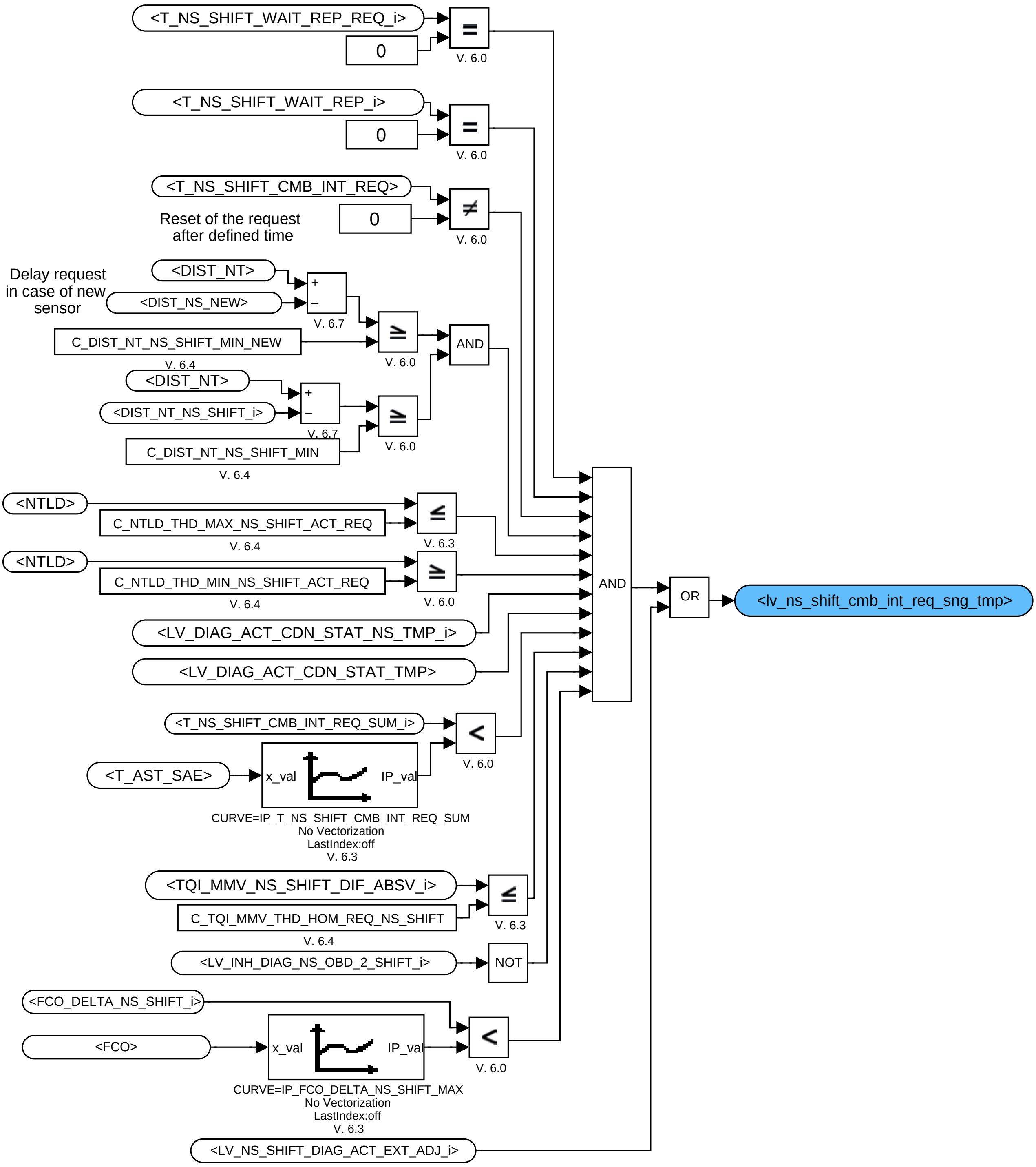


Figure 179:

52.19.2.2.3.1.1.3 Set/Reset of the combustion request

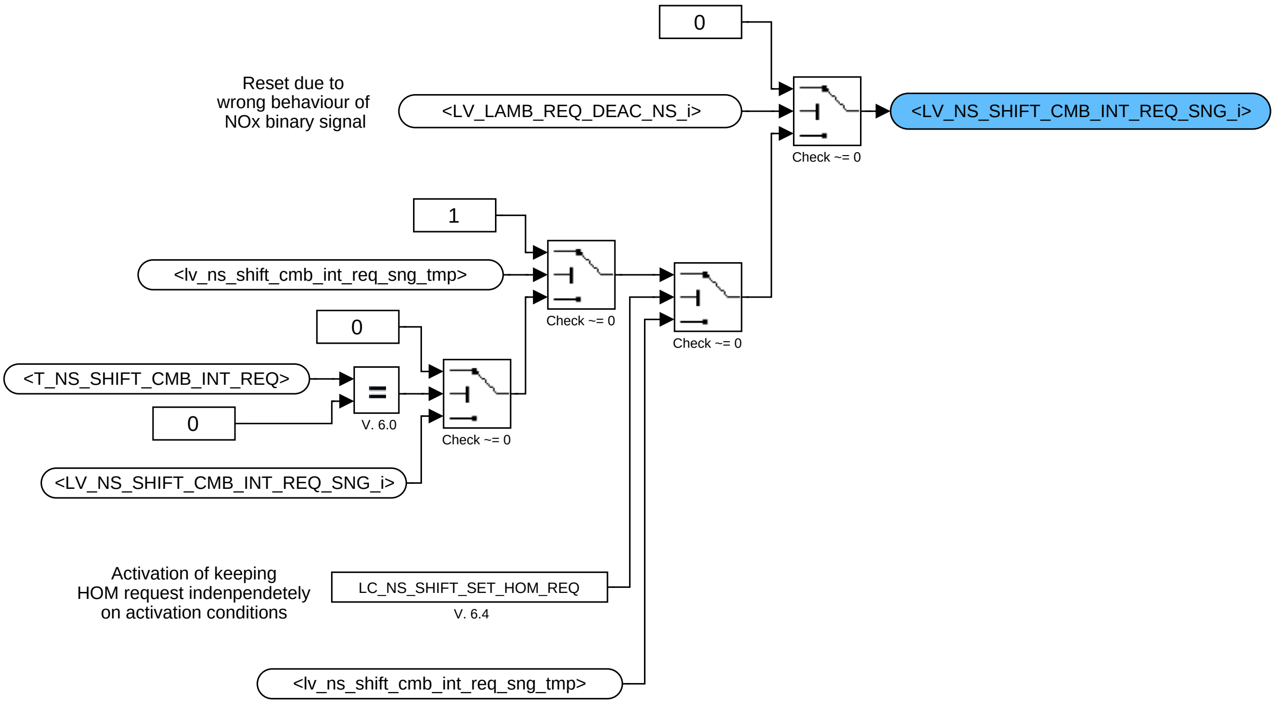


Figure 180:

52.19.2.2.3.1.2 Calculation of timers and initialization other variables

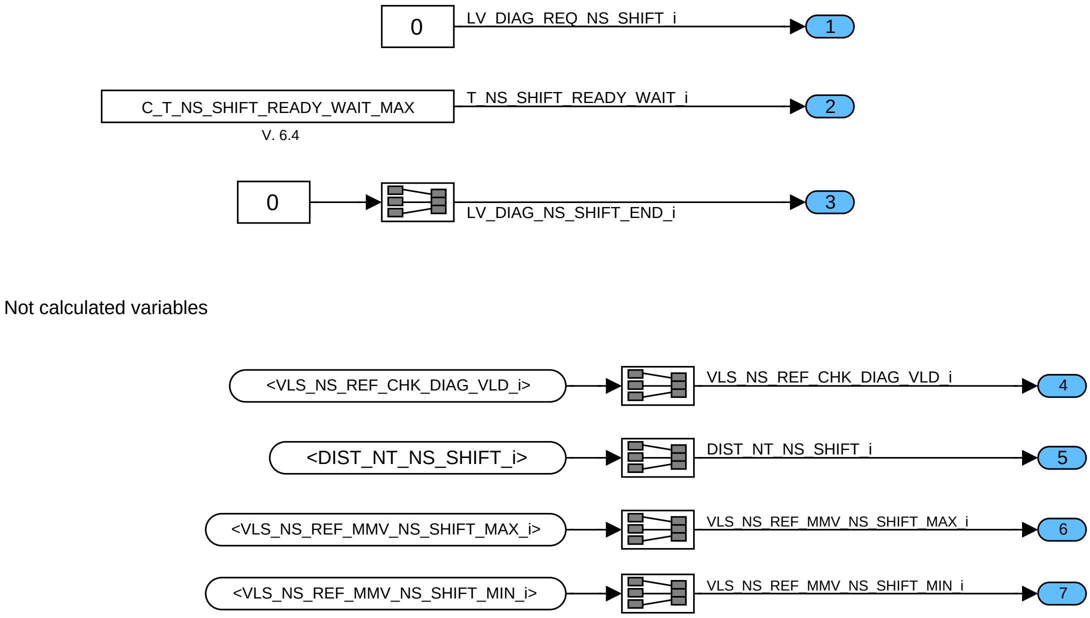


Figure 181:

52.19.2.2.3.2 State CDN

52.19.2.2.3.2.1 Calculations

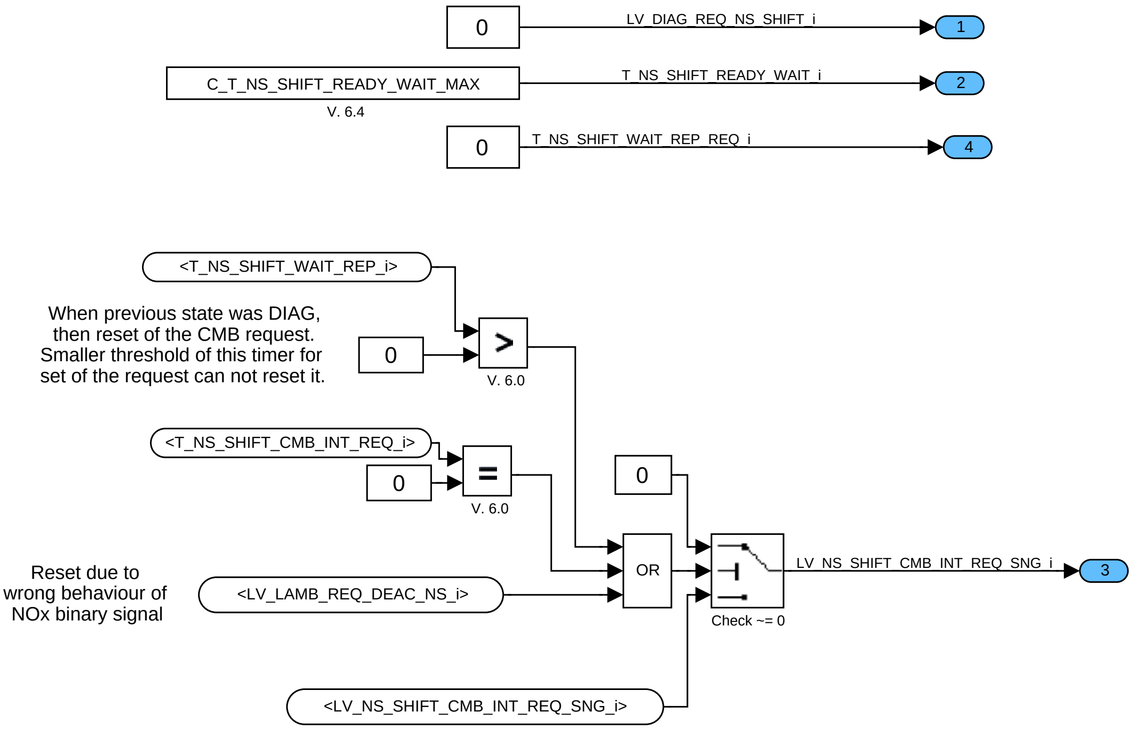


Figure 182:

52.19.2.2.3.2.2 Not calculated variables

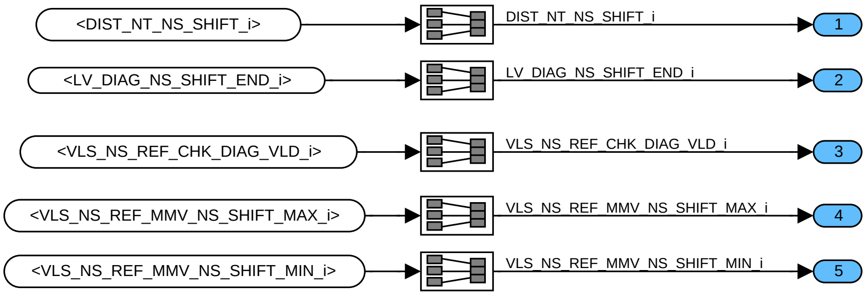


Figure 183:

52.19.2.2.3.3 State DIAG

52.19.2.2.3.3.1 Calculation timers for waiting and repetition

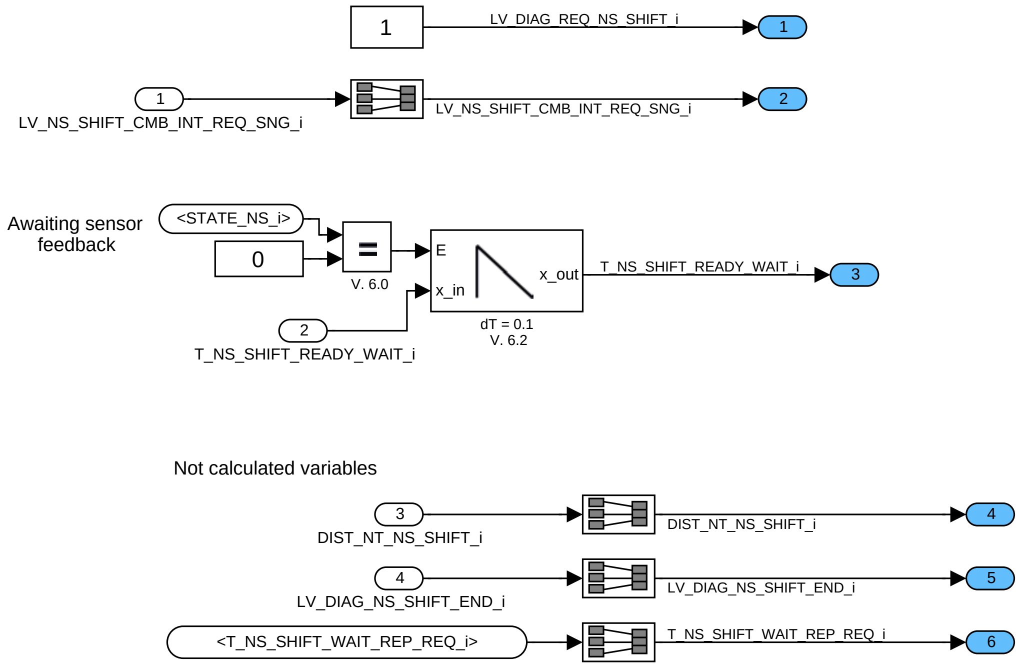


Figure 184:

52.19.2.2.3.3.2 Check validation of diagnosis result

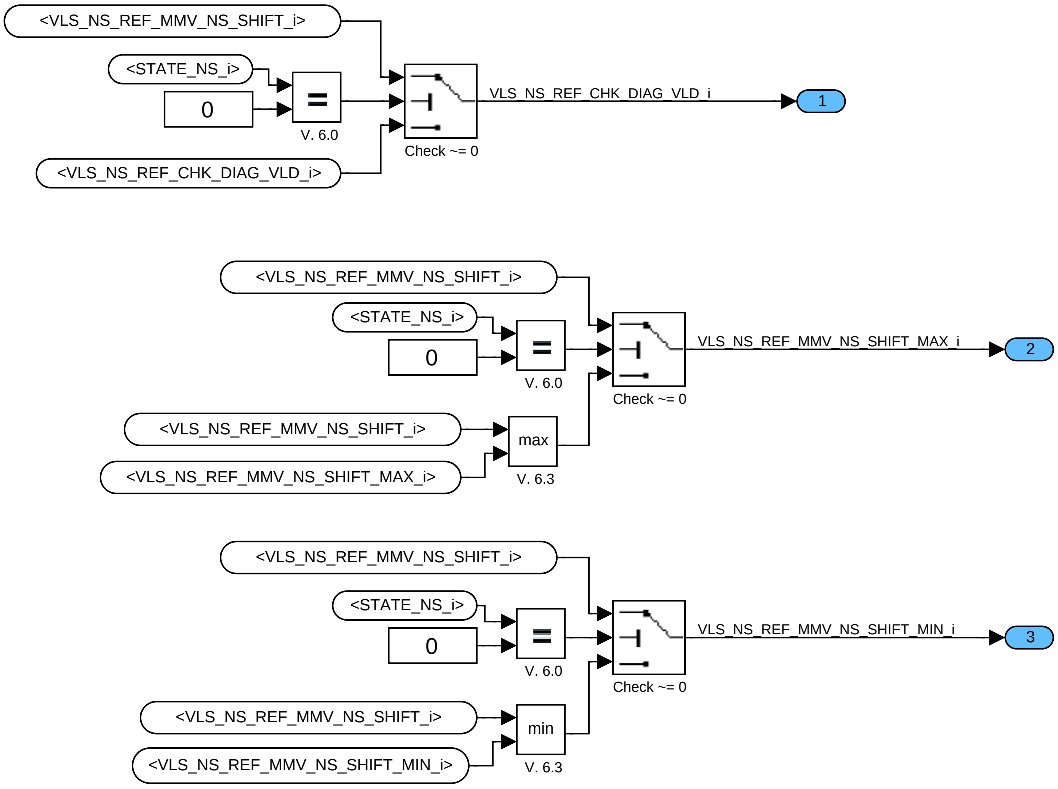


Figure 185:

52.19.2.2.3.4 State END

52.19.2.2.3.4.1 Calculations

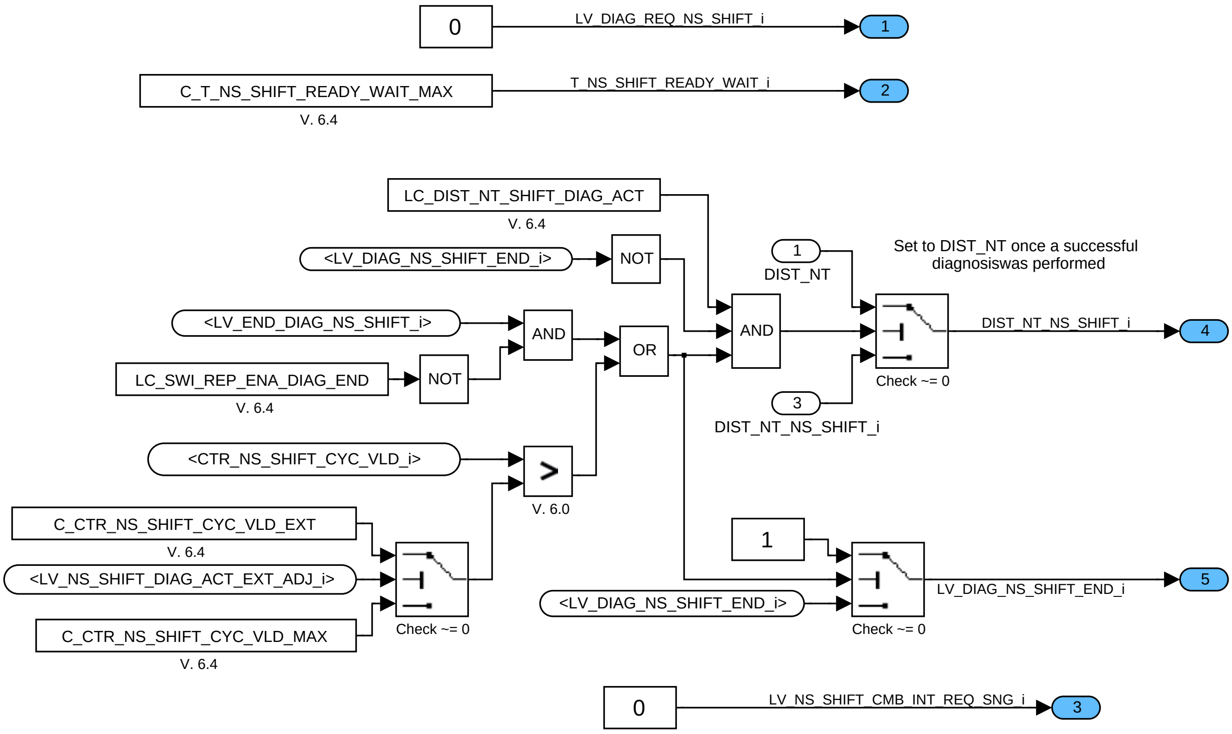


Figure 186:

52.19.2.2.3.4.2 Not calculated variables

52.19.2.2.4 Calculations at availability of external values

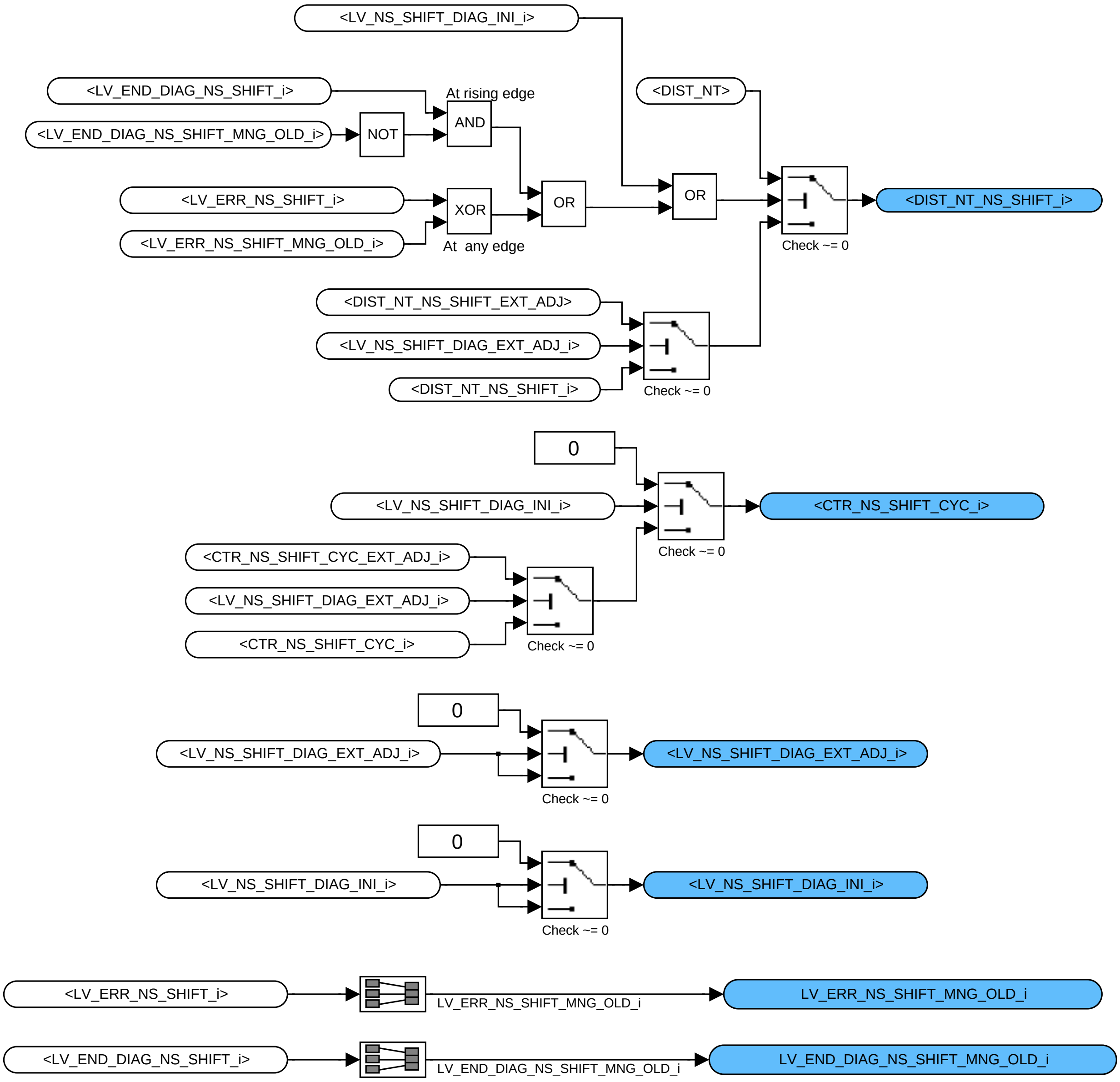


Figure 187:

52.19.2.2.5 Calculation of diagnosis status for external service tool

52.19.2.2.5.1 Managing of the START and STOP switches

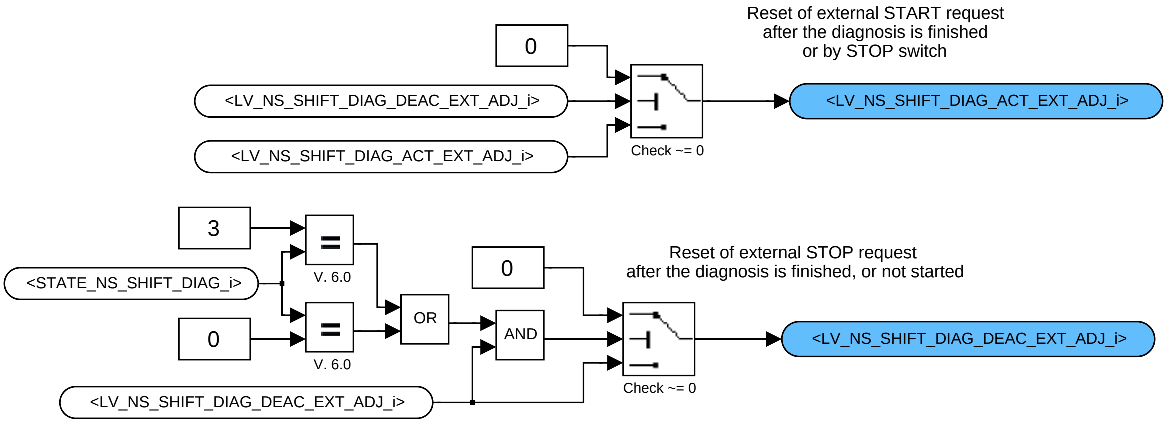


Figure 188:

52.19.2.2.5.2 KWP value calculation

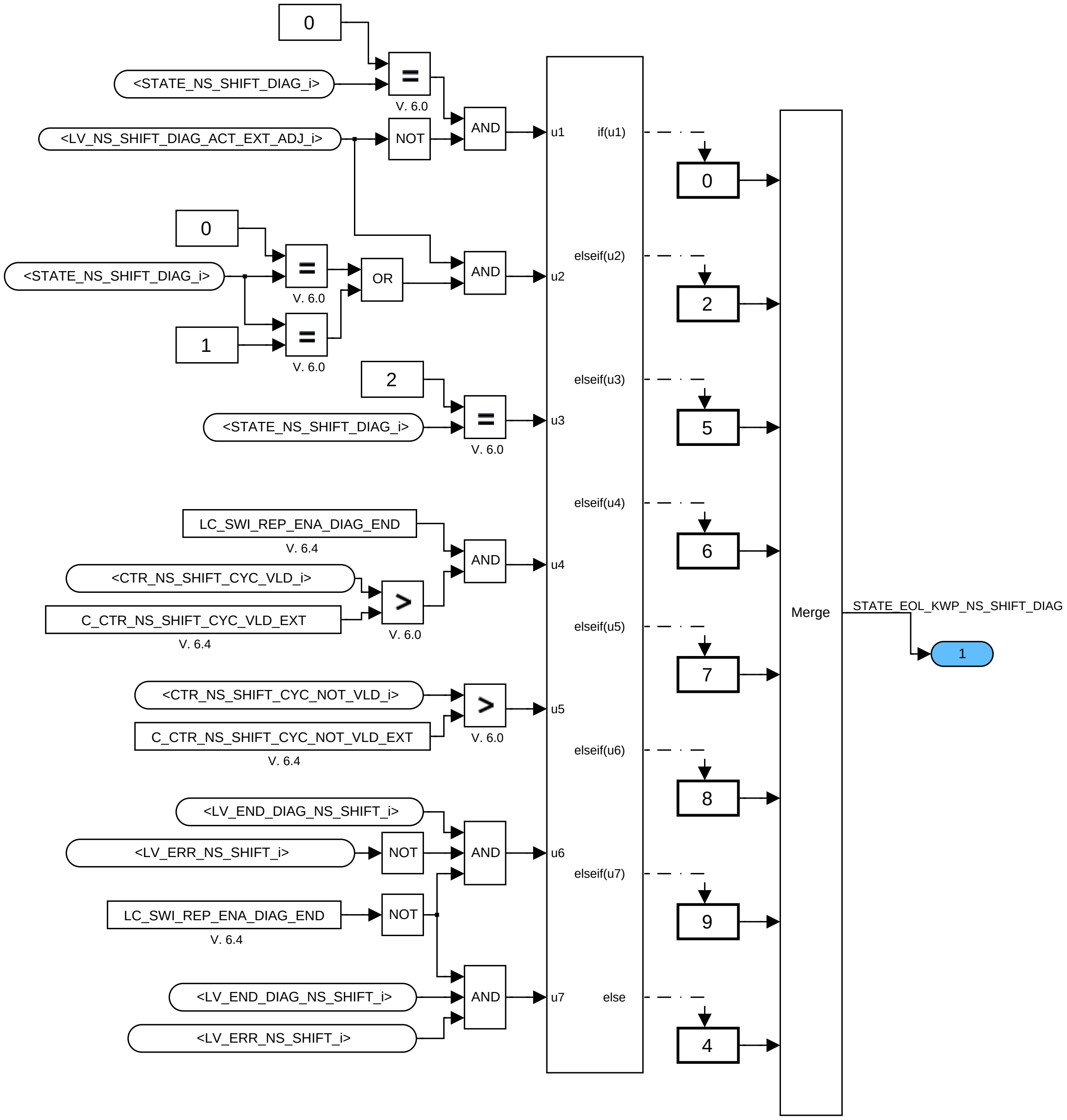


Figure 189:

52.19.2.3 Synchronization of all sensors and generation summary request for combustion manager

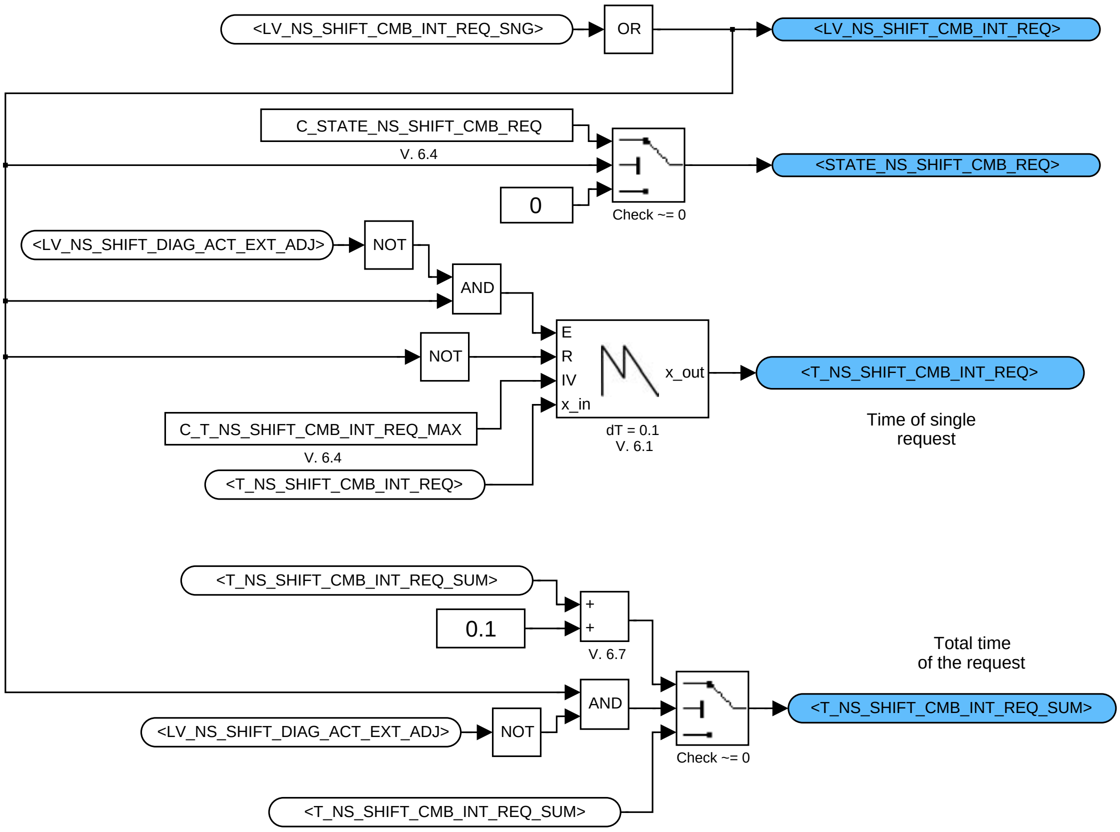


Figure 190:

52.19.3 Detailed description for Action: ACTION_NOXD_CleanNSShiftAdapt

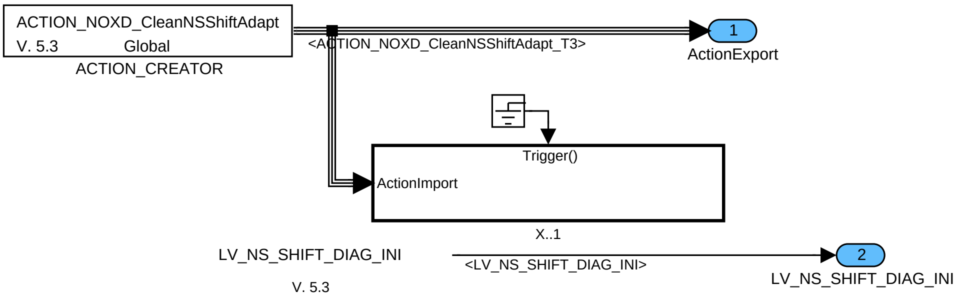


Figure 191:

52.19.3.1 Calculations in ACTION_NOXD_CLEANNSSHIFTADAPT

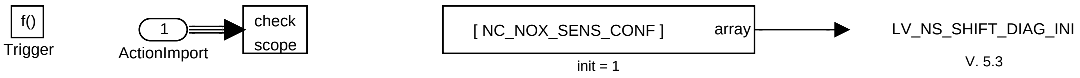


Figure 192:

52.19.4 Detailed description for Action: ACTION_NOXD_EndNNShiftDiag

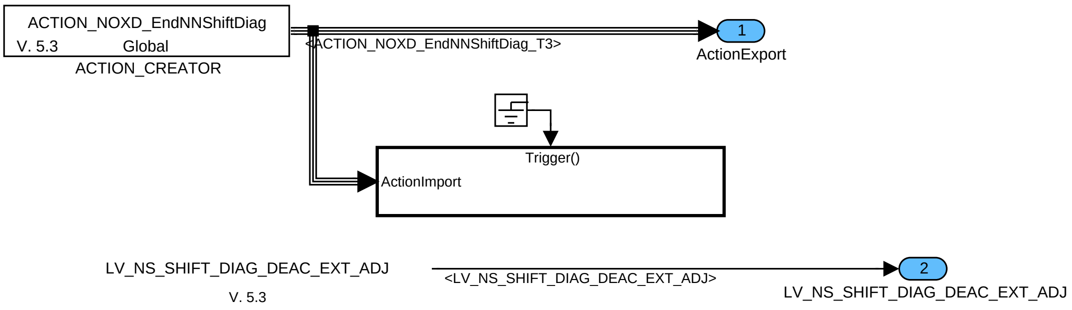


Figure 193:

52.19.4.1 Calculation in ACTION_NOXD_ENDNNSHIFTDIAG

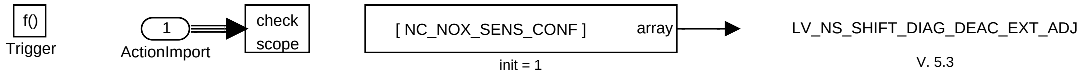


Figure 194:

52.19.5 Detailed description for Action: ACTION_NOXD_StartNNShiftDiag

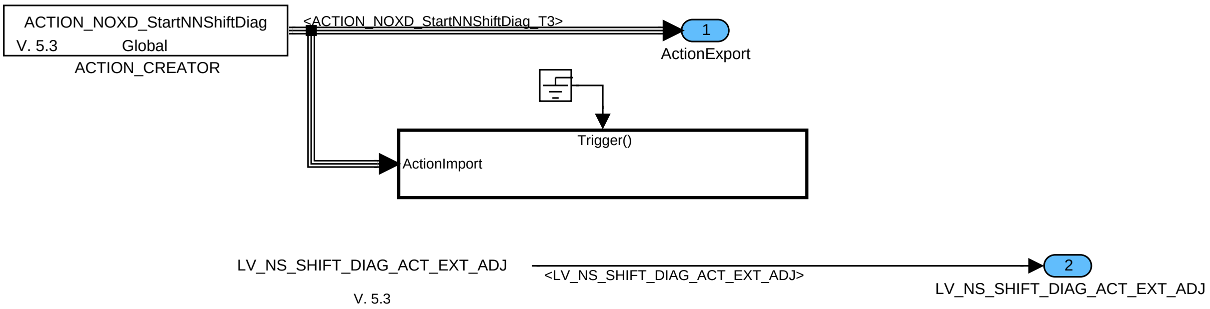


Figure 195:

52.19.5.1 Calculation in ACTION_NOXD_STARTNNSHIFTDIAG

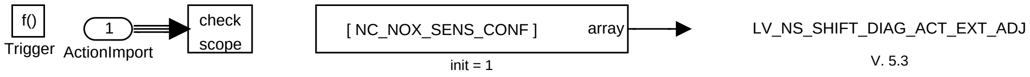


Figure 196:

52.19.6 Detailed description for Action: ACTION_NOXD_WriteNSShiftDiagExtAdj

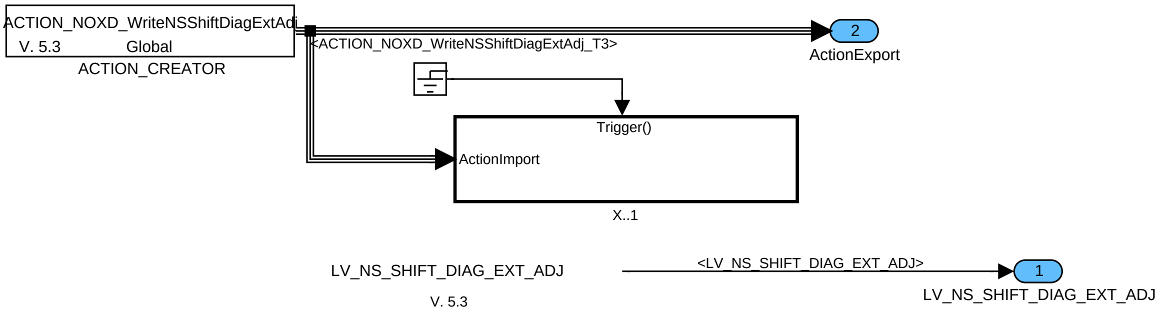


Figure 197:

52.19.6.1 Calculation in ACTION_NOXD_WRITENSSHIFTDIAGEXTADJ

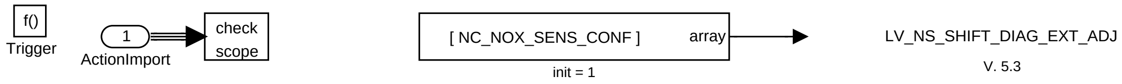


Figure 198:

52.20 NOx signal internal setpoint shift diagnosis

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CRLC_RATIO_MMV_NS_SHIFT_DIAG [NC_NOX_SENS_CONF]	O/V	0... FFH	0... 1	3.92157e-3	[-]
Correlation constant of NOx signal shift diagnosis mean value					
CTR_RATIO_MMV_DIAG_NS_SHIFT [NC_NOX_SENS_CONF]	O/V/S	0... FFH	0... 255	1	[-]
Counter of single diagnosis cycles for calculation of mean value					
ERR_SYM_NS_SHIFT [NC_NOX_SENS_CONF]	V	0 1 2 4 8	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	-	[-]
Detected error symptom for NOx signal internal setpoint shift diagnosis					
LV_CDN_DIAG_NS_SHIFT [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Diagnostic condition to start symptom detection (seto to one when condition is fulfilled)					
LV_END_DIAG_NS_SHIFT [NC_NOX_SENS_CONF]	O/V	0... 1H	0... 1	1	[-]
NOx sensor internal setpoint shift diagnosis was performed at last one time					
LV_ERR_NS_SHIFT [NC_NOX_SENS_CONF]	O/V	0... 1H	0... 1	1	[-]
Present failure after filtering of the NOx sensor internal setpoint shift diagnosis					
LV_NS_SHIFT_DIAG_MMV_EXT_ADJ [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
External values of MMV values for the NOx signal shift diagnosis available					
LV_NS_SHIFT_DIAG_MMV_INI [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Initialization of MMV values for the NOx signal shift diagnosis					
LV_RATIO_SUM_CLC_ACT [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Calculation of RATIO active					
RATIO_MMV_NS_SHIFT_DIAG [NC_NOX_SENS_CONF]	O/V/S	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
NOx signal shift diagnosis mean value					
RATIO_NS_SHIFT_DIAG [NC_NOX_SENS_CONF]	O/V	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
NOx signal shift diagnosis value					
RATIO_NS_SHIFT_DIAG_BOL_SAE [NC_NOX_SENS_CONF]	O/V/S	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
Mode 06 - minimum value of test limit					
RATIO_NS_SHIFT_DIAG_SAE [NC_NOX_SENS_CONF]	O/V/S	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
Mode 06 - test value					
RATIO_NS_SHIFT_DIAG_TOL_SAE [NC_NOX_SENS_CONF]	O/V/S	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
Mode 06 - maximum value of test limit					
RATIO_SUM_NS_SHIFT_DIAG [NC_NOX_SENS_CONF]	O/V/S	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[-]
Numerator for calculation of diagnosis mean value					
STATE_NS_SHIFT_DIAG_OLD [NC_NOX_SENS_CONF]	-	0... FFFFH	0... 65535	1	[-]
Old value of STATE_NS_SHIFT_DIAG					

Input Data:

NC_NOX_SENS_CONF	LV_INH_DIAG_NS_OBD_2_SHIFT [NC_NOX_SENS_CONF]	STATE_NS_SHIFT_DIAG [NC_NOX_SENS_CONF]	CAN_NOX_DIAG_NS [NC_NOX_SENS_CONF]
TNT_MDL_MV	LV_NS_SHIFT_DIAG_ERR [NC_NOX_SENS_CONF]	LV_NS_SHIFT_DIAG_ACT_EXT_ADJ [NC_NOX_SENS_CONF]	CAN_NOX_REF_NS [NC_NOX_SENS_CONF]
MAF_CYL	RATIO_MMV_NS_SHIFT_DIAG_EXT_ADJ [NC_NOX_SENS_CONF]		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ABC_INC_NS_SHIFT	1	0... FFH	0... 255	1	[-]
Antibounce counter increment					
C_ABC_MAX_NS_SHIFT	1	1... FFH	1... 255	1	[-]
Maximum value for antibounce counter					
C_ABC_MAX_NS_SHIFT_EXT_ACT	1	1... FFH	1... 255	1	[-]
Maximum value for antibounce counter - for external manual activation					
C_CRLC_RATIO_NS_SHIFT_DIAG_MIN	1	0... FFH	0... 1	3.92157e-3	[-]
Minimum Correlation constant for NOx signal shift diagnosis mean value					
C_CTR_RATIO_MMV_DIAG_NS_SHIFT	1	1... FFH	1... 255	1	[-]
Maximum of single diagnosis cycles for calculation of mean value					
C_CTR_RATIO_MMV_SHIFT_EXT_ACT	1	1... FFH	1... 255	1	[-]
Maximum of single diagnosis cycles for calculation of mean value - external activation					
C_RATIO_NS_SHIFT_DIAG_AFS_MAX	1	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
Maximal threshold for NOx signal shift diagnosis for requesting Lambda=1.0 operation					
C_RATIO_NS_SHIFT_DIAG_AFS_MIN	1	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
Minimal threshold for NOx signal shift diagnosis for requesting Lambda=1.0 operation					
C_RATIO_NS_SHIFT_DIAG_THD_MAX	1	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
Maximal threshold for NOx signal shift diagnosis					
C_RATIO_NS_SHIFT_DIAG_THD_MIN	1	8000... 7FFFH	-1... 0.99996948242	30.5176e-6	[-]
Minimal threshold for NOx signal shift diagnosis					
IP_CRLC_RATIO_MMV_NS_SHIFT_DIAG	6*4	0... FFH	0... 1	3.92157e-3	[-]
LDP_RATIO_DIF_IP_CRLC_RATIO_MMV	6	0... FFFFH	-1... 0.99996948	30.5176e-6	[-]
LDP_RATIO_IP_CRLC_RATIO_MMV	4	0... FFFFH	-1... 0.99996948	30.5176e-6	[-]
Filter constant for NOx signal shift diagnosis mean value					
IP_FAC_MAF_NS_SHIFT	8	0... FFH	0... 3.984375	0.015625	[-]
LDP_MAF_IP_FAC_NS_SHIFT	8	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Correction factor of NOx signal depending on MAF_CYL					
IP_FAC_TNT_MDL_MV_NS_SHIFT	8	0... FFH	0... 3.984375	0.015625	[-]
LDP_TNT_MDL_MV_IP_NS_SHIFT	8	0... FFFFH	0... 1023.984375	0.015625	[°C]
Correction factor of NOx signal depending on TNT_MDL_MV					
LC_NS_SHIFT_DIAG_ERR_ACT	1	0... 1H	0... 1	1	[-]
Switch to result of diagnosis comming from NOx sensor software					

Export Actions:

ACTION_NOXD_CleanMMVNSAdapt()	
Initialization of NOx signal shift MMV diagnosis values	
ACTION_NOXD_WriteMMVNSExtAdj()	
Write external adjustment values to NOx signal shift MMV diagnosis values	

Description for Actions

ACTION_NOXD_CleanMMVNSAdapt()
Initialization of NOx signal shift MMV diagnosis values

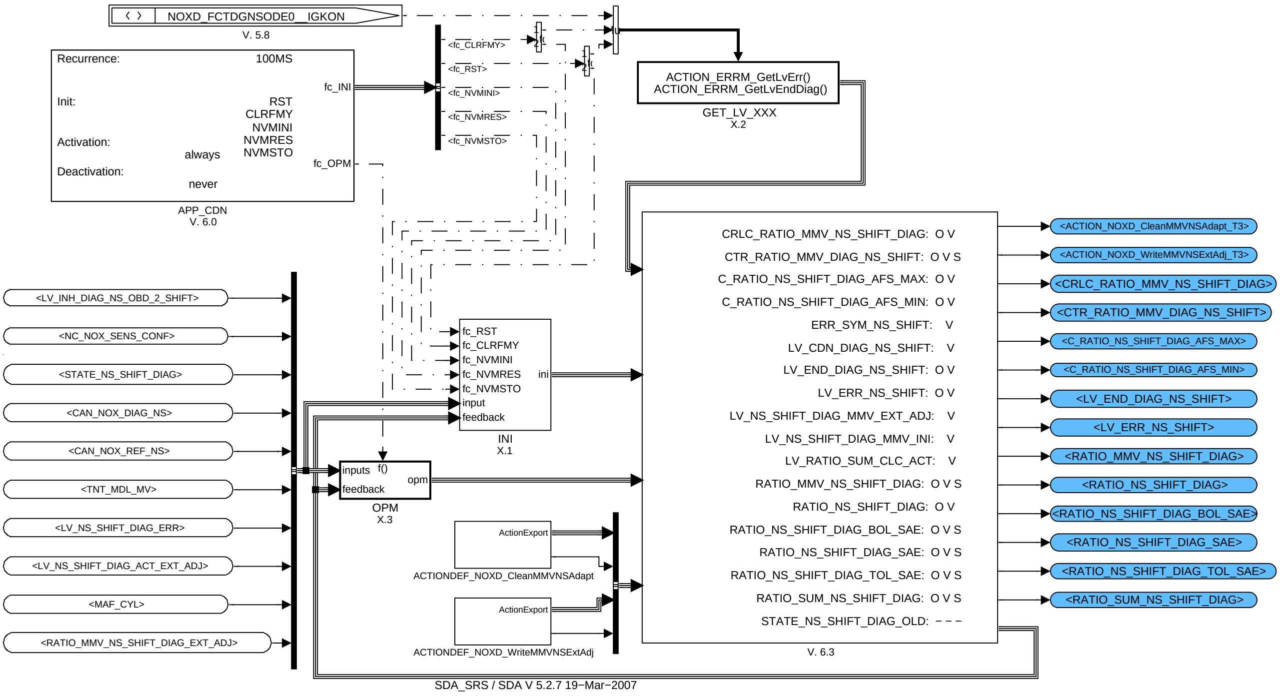
ACTION_NOXD_WriteMMVNSExtAdj()
Write external adjustment values to NOx signal shift MMV diagnosis values

General Information

Application Conditions

Initialization: RST, CLRFMY, NVMINI, NVMRES, NVMSTO
Recurrence: 100MS
Activation: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.2.7 19−Mar−2007

Figure 199:

52.20.1 Initialization and Management of nonvolatile memory

52.20.1.1 Initialization at Reset

The old values will be initialized with the regarding inputs, due to avoid an edge detection after reset.

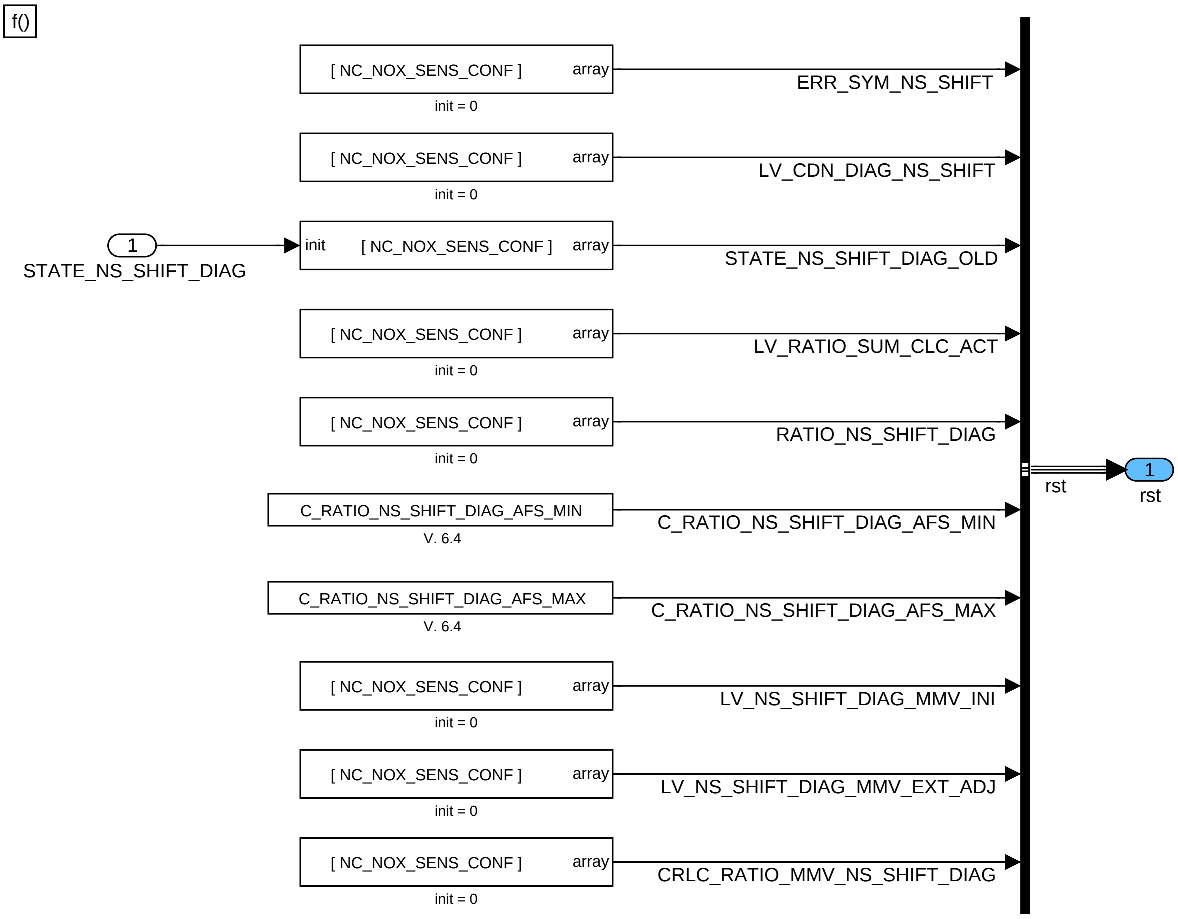


Figure 200:

52.20.1.2 Initialization of NVMY variables

All Mode 6 values will be stored in the nonvolatile memory. The values will be reset to zero, if failure memory will be cleared.

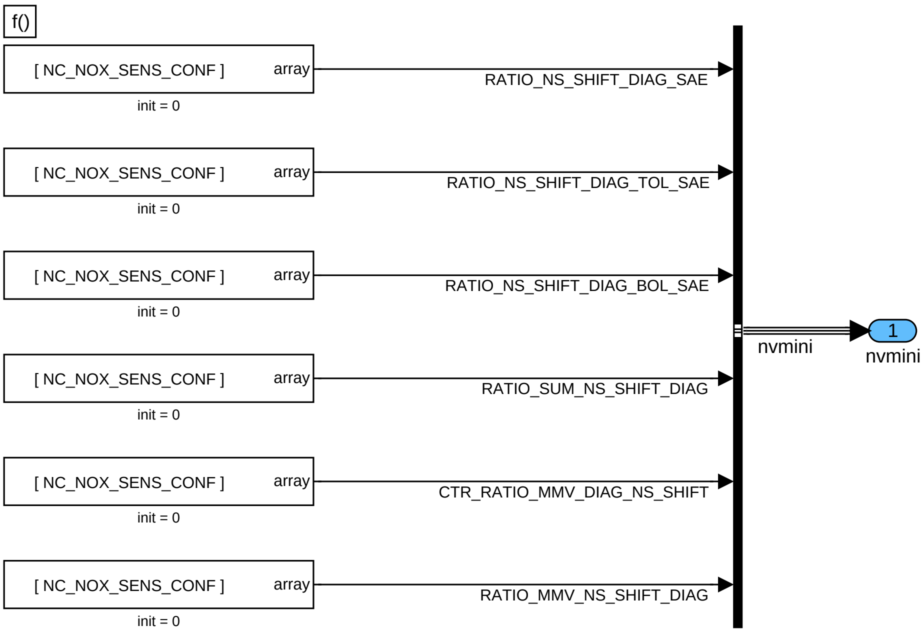


Figure 201:

52.20.1.3 Actions at clear fault memory

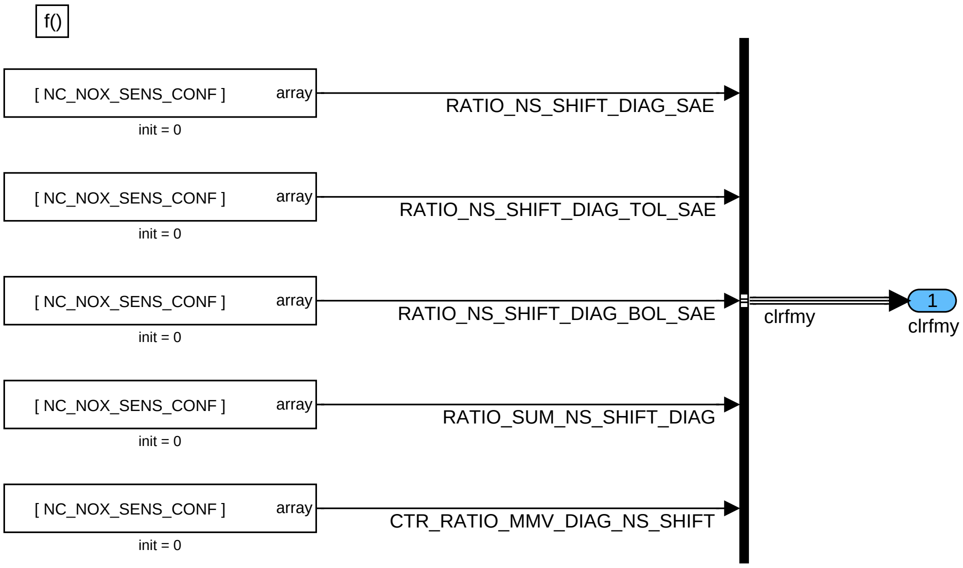


Figure 202:

52.20.2 Synchronization of the LV_ERR_NS_SHIFT and LV_END_DIAG_NS_SHIFT

Synchronisation between LV_ERR_NS_SHIFT and its result from ERRM is done using action ACTION_ERRM_GetLvErr() similarly synchronisation between LV_END_DIAG_NS_SHIFT and its result from ERRM is done using action ACTION_ERRM_GetLvEndDiag()at RESET ,IGKON and CLRFMY events.

52.20.3 Operation at 100MS

52.20.3.1 Introduction of multiple sensor system

52.20.3.1.1 Calculation of the diagnosis results

52.20.3.1.1.1 Determination of diagnostic condition and calculation of counter

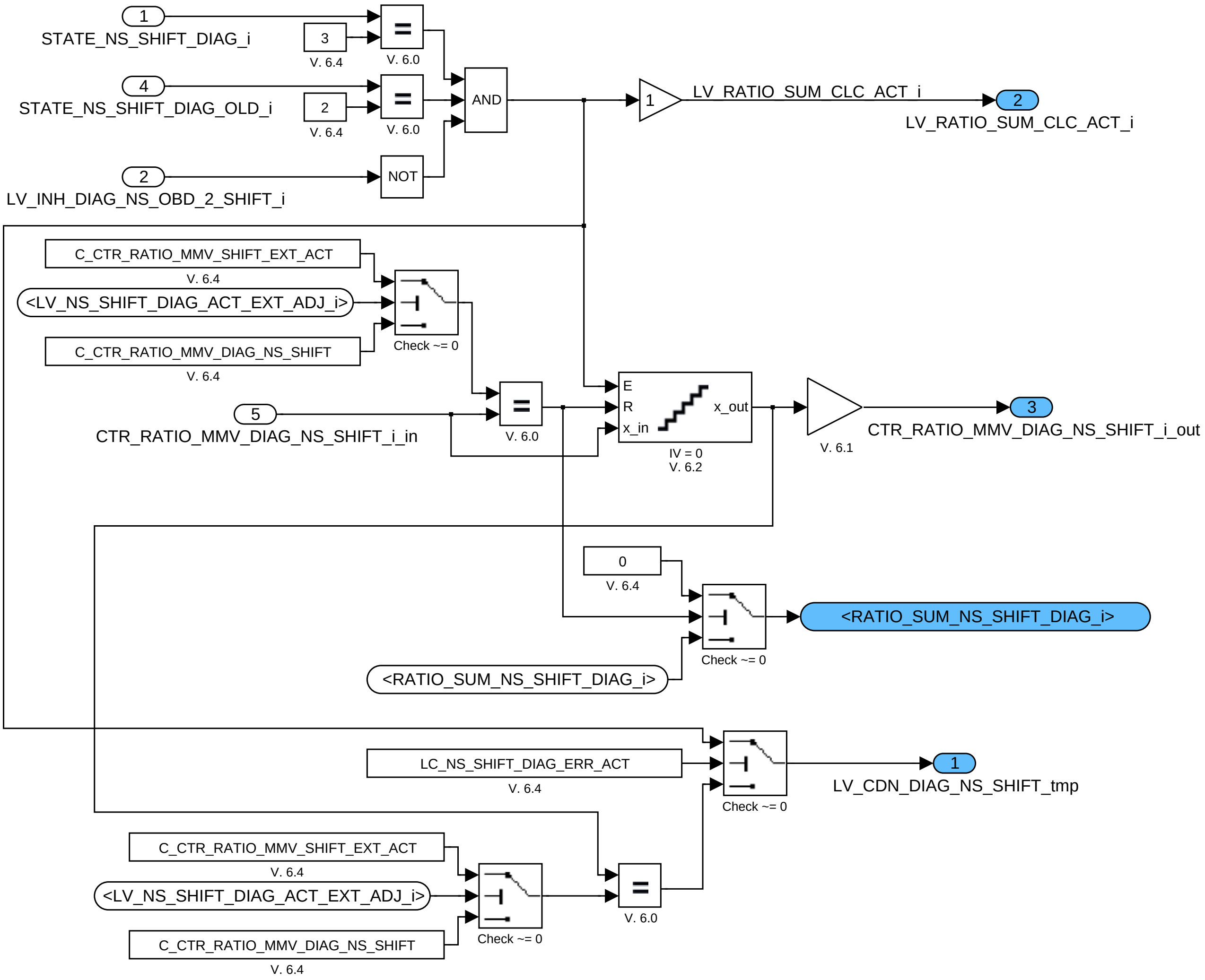


Figure 203:

52.20.3.1.1.2 Calculation of single diagnosis value

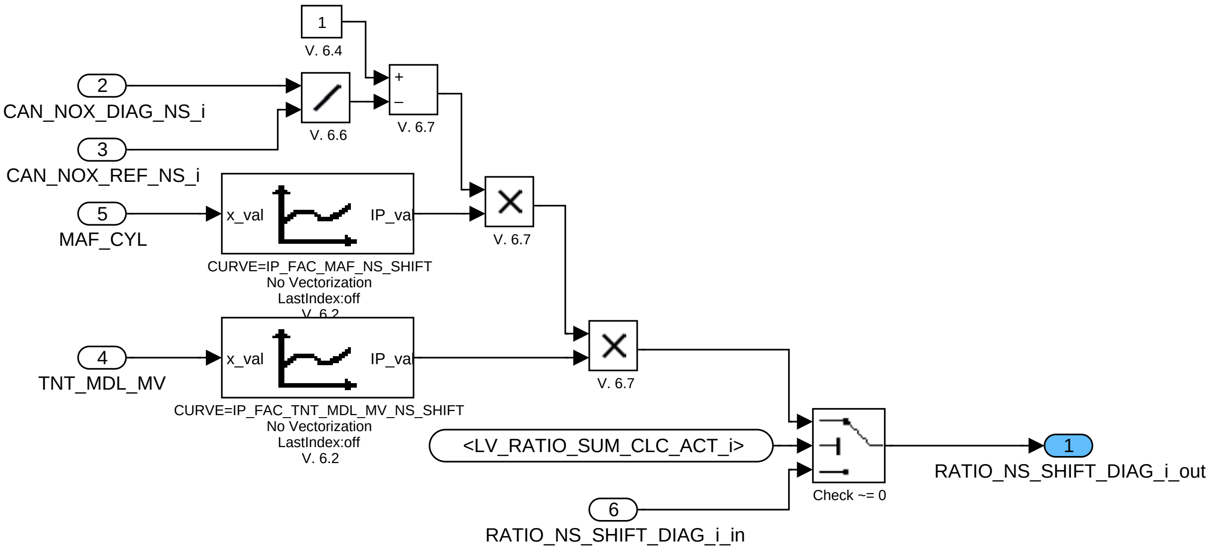


Figure 204:

52.20.3.1.1.3 Calculation of mean value over defined number of single diagnosis values

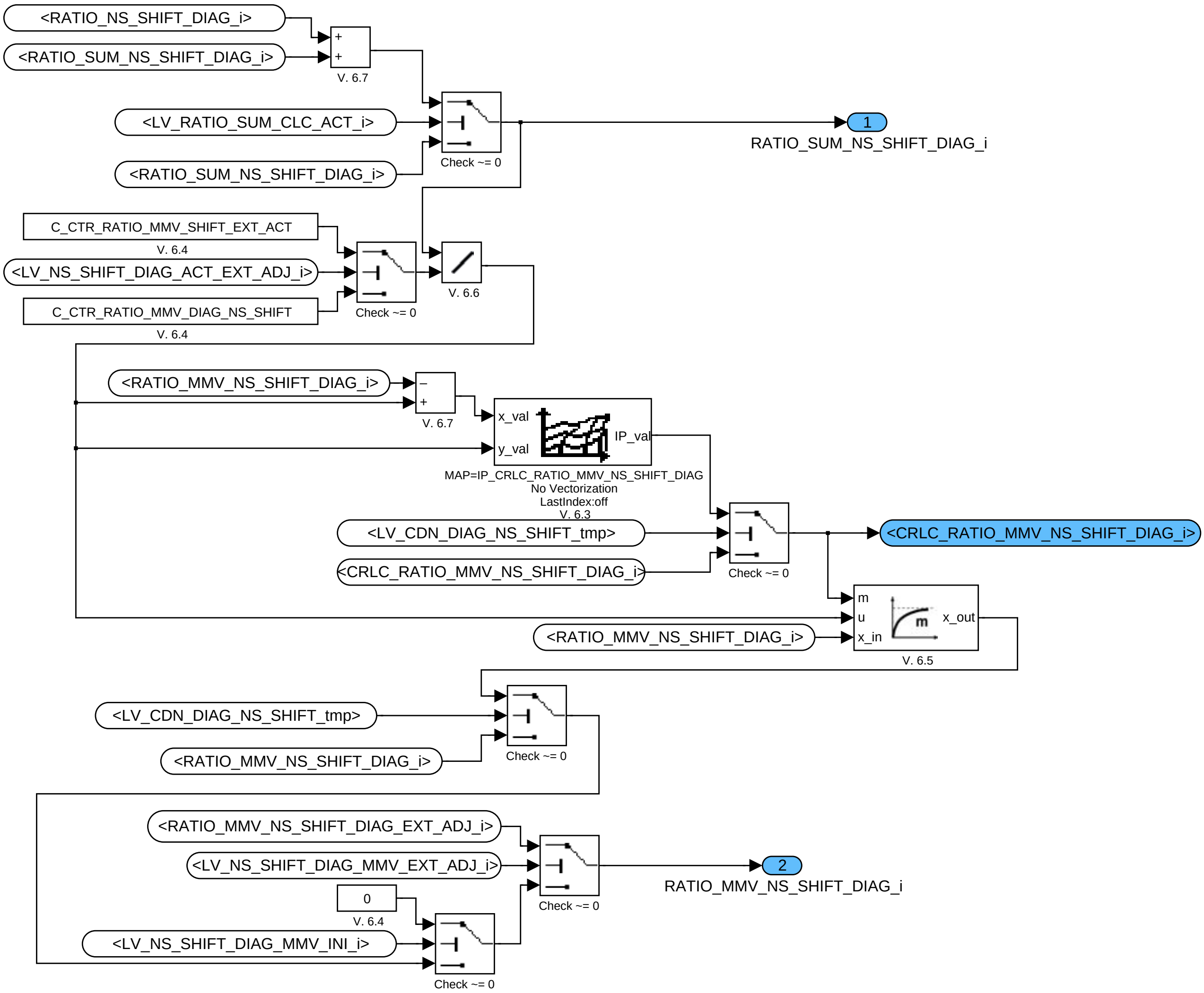


Figure 205:

52.20.3.1.1.4 Determination of error symptom

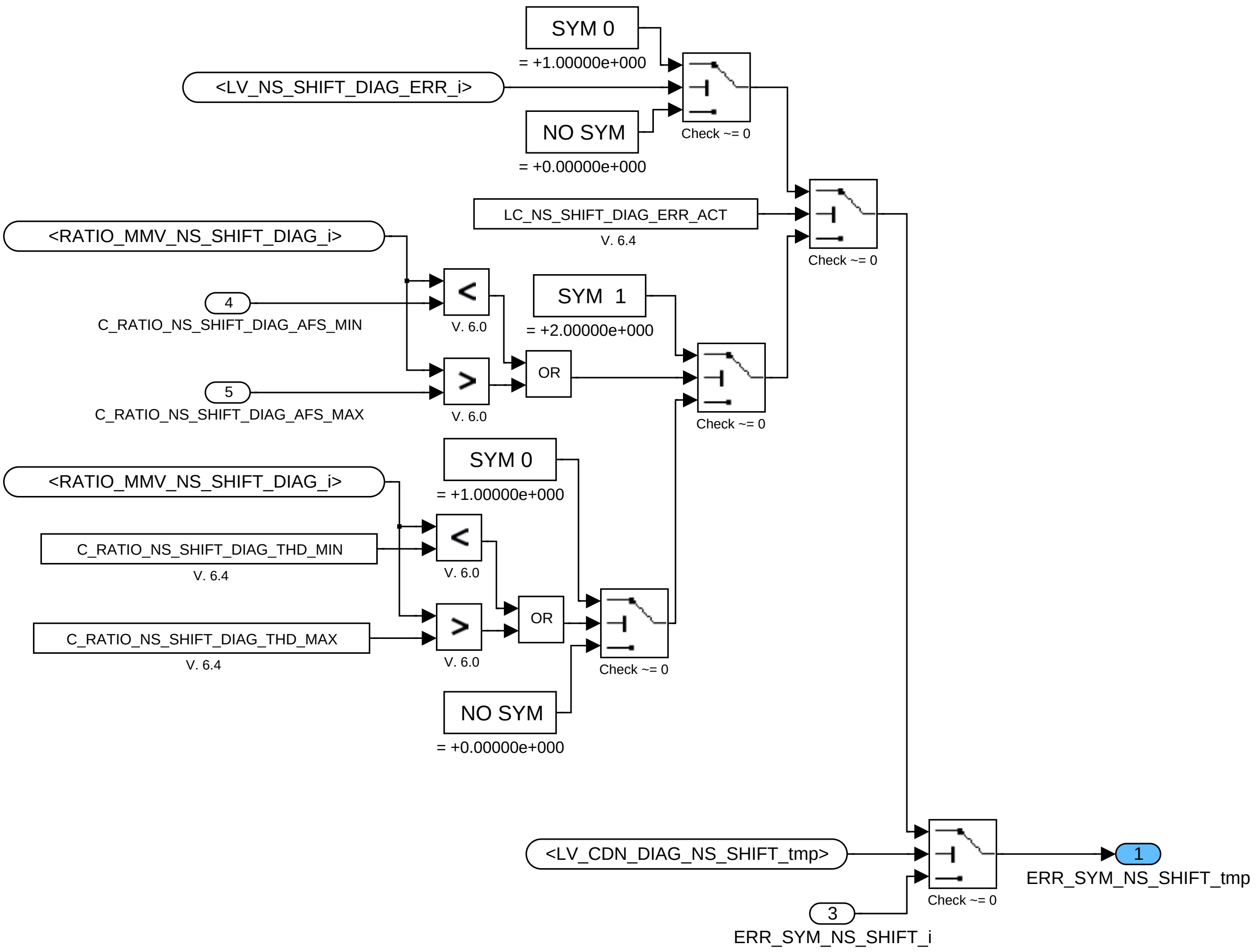


Figure 206:

52.20.3.1.1.5 Call for ERRM

Import actions:

ACTION_ERRM_FilterSymptom(IN<XX>, IN<lv_cdn_diag_XX>, IN<err_sym_XX >, IN<C_ABC_INC_XX >, IN<C_ABC_DEC_XX >, IN<C_ABC_MAX_XX >, OUT<LV_ERR_XX>)
The action computes the elementary anti-bounce filter for one failure treatment and returns filter results

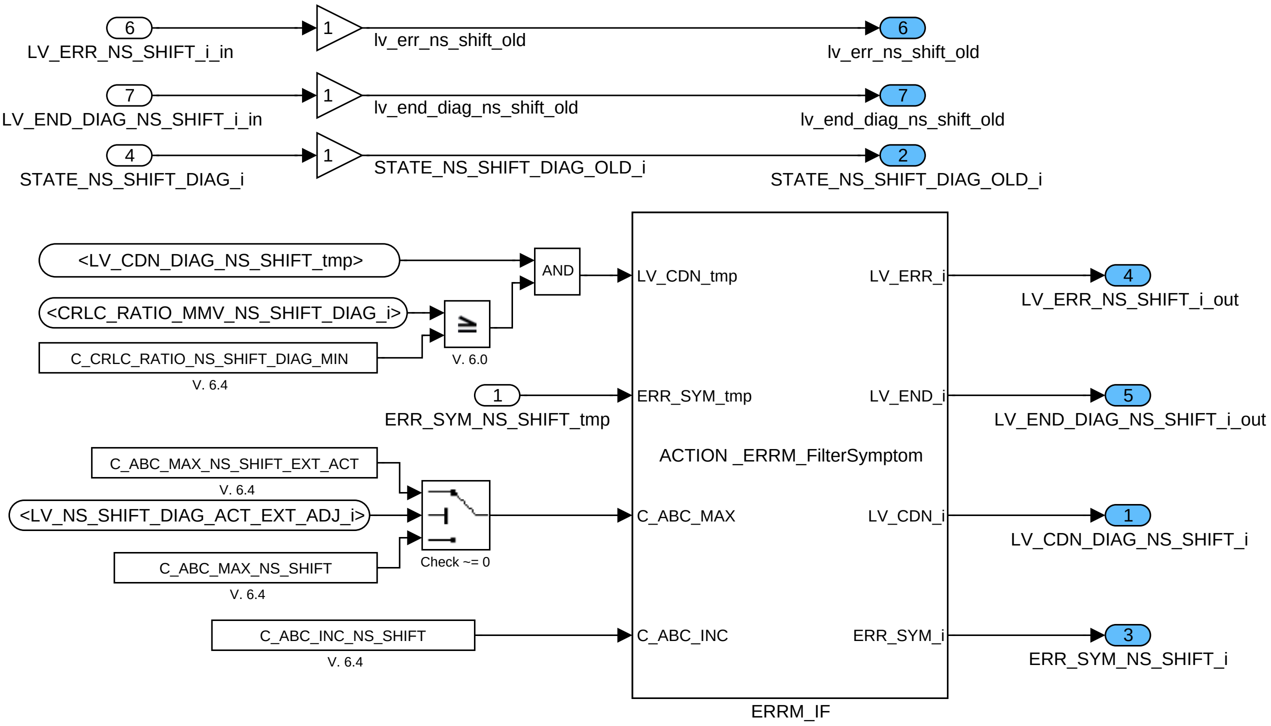


Figure 207:

52.20.3.1.1.6 Diagnosis Mode 6 handling

Mode 6 values will be stored, if error management is finishing the diagnosis or is setting an error.

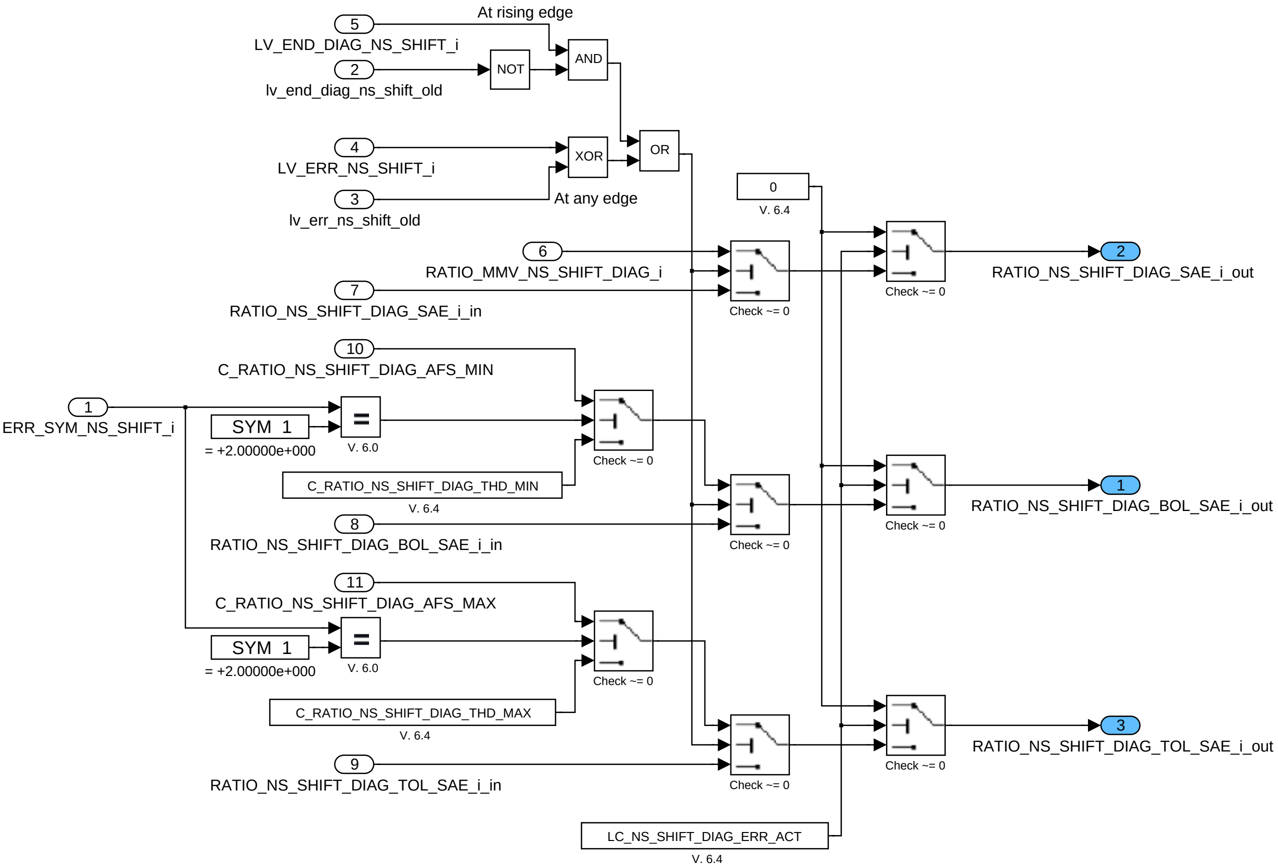


Figure 208:

52.20.3.1.1.7 Reset bits for writing external values or initialization of MMV values

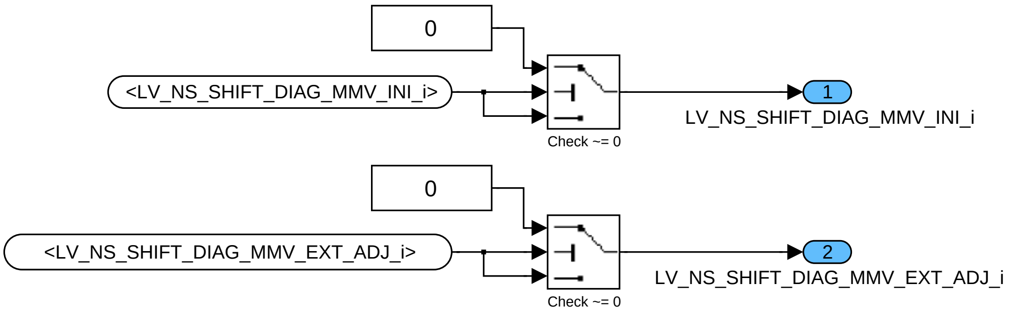


Figure 209:

52.20.4 Detailed description for Action: ACTION_NOXD_CleanMMVNSAdapt

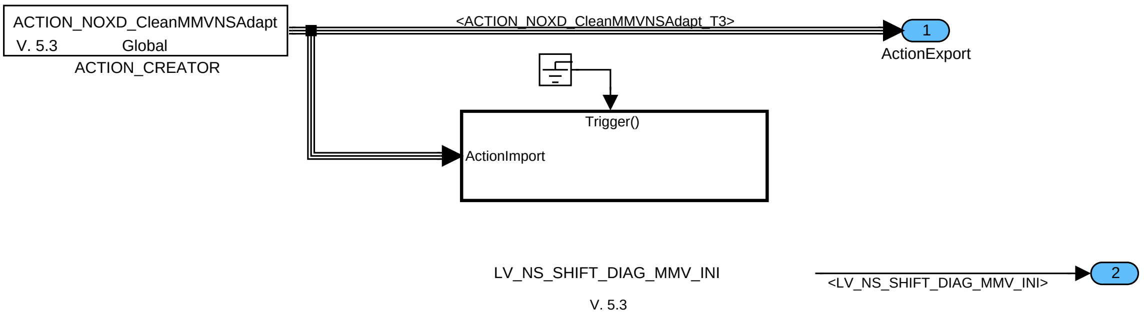


Figure 210:

52.20.4.1 No title given

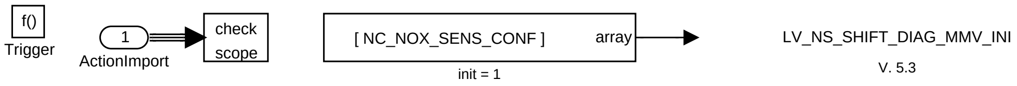


Figure 211:

52.20.5 Detailed description for Action: ACTION_NOXD_WriteMMVNSExtAdj

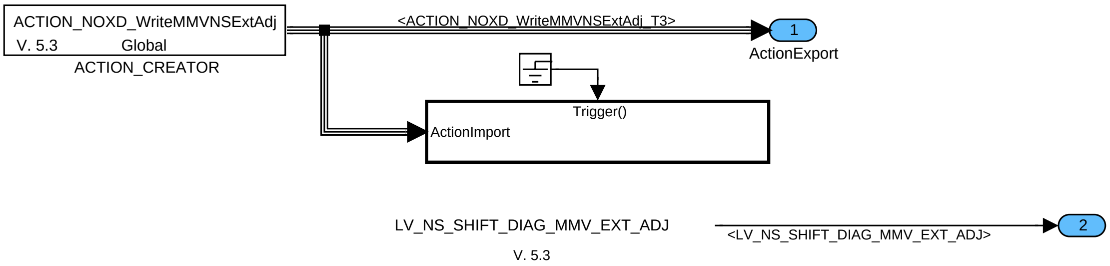


Figure 212:

52.20.5.1 No title given

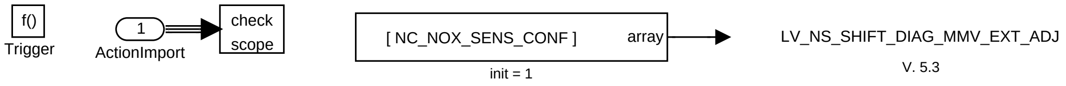


Figure 213:

52.21 NOx Sensor version check diagnosis

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
ERR_SYM_NS_VERS [NC_NOX_SENS_CONF]	V	0 1 2 4 8	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	-	[-]
Detected error symptom for NOx sensor version check diagnosis					
LV_CDN_DIAG_NS_VERS [NC_NOX_SENS_CONF]	V	0... 1H	0... 1	1	[-]
Diagnostic condition to start symptom detection (seto to one when condition is fulfilled)					
LV_END_DIAG_NS_VERS [NC_NOX_SENS_CONF]	O/V	0... 1H	0... 1	1	[-]
NOx sensor version check diagnosis was performed at last one time					
LV_ERR_NS_VERS [NC_NOX_SENS_CONF]	O/V	0... 1H	0... 1	1	[-]
Present failure after filtering of NOx sensor. NOx-Sensor version check					
NOX_CAN_SW_VERS_DIAG_BOL_SAE [NC_NOX_SENS_CONF]	O/V/S	0... FFH	0... 255	1	[-]
Mode 06 - minimum value of test limit for NOx sensor version check diagnosis					
NOX_CAN_SW_VERS_DIAG_SAE [NC_NOX_SENS_CONF]	O/V/S	0... FFH	0... 255	1	[-]
Mode 06 - test value of NOx sensor version check diagnosis					
STATE_NS_VERS [NC_NOX_SENS_CONF]	O/V	0... FFH	0... 255	1	[-]
State for NOx Sensor version check diagnosis					

Input Data:

NC_NOX_SENS_CONF	CAN_STATE_NS_TMP [NC_NOX_SENS_CONF]	T_AST_SAE	CAN_SW_NS [NC_NOX_SENS_CONF]
LV_DC	C_STATE_NS_MASK_MODE_3_A	C_STATE_NS_MASK_MODE_3_B	LV_INH_DIAG_NS_OBD_2_VERS [NC_NOX_SENS_CONF]

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ABC_INC_NS_VERS	1	0... FFH	0... 255	1	[-]
Antibounce counter increment					
C_ABC_MAX_NS_VERS	1	1... FFH	1... 255	1	[-]
Maximum value for antibounce counter					
C_NOX_NS_DIAG_VERS_SW_THD	1	0... FFH	0... 255	1	[-]
Theshold for NOx sensor version check diagnosis value					
C_T_AST_MIN_NS_VERS_DIAG	1	0... FFFFH	0... 65535	1	[s]
Minimum value of PID1F Cumulated time					
LC_NS_VERS_CHK_REP	1	0... 1H	0... 1	1	[-]
switch for request of NOx Sensor version check diagnosis value					

General Information

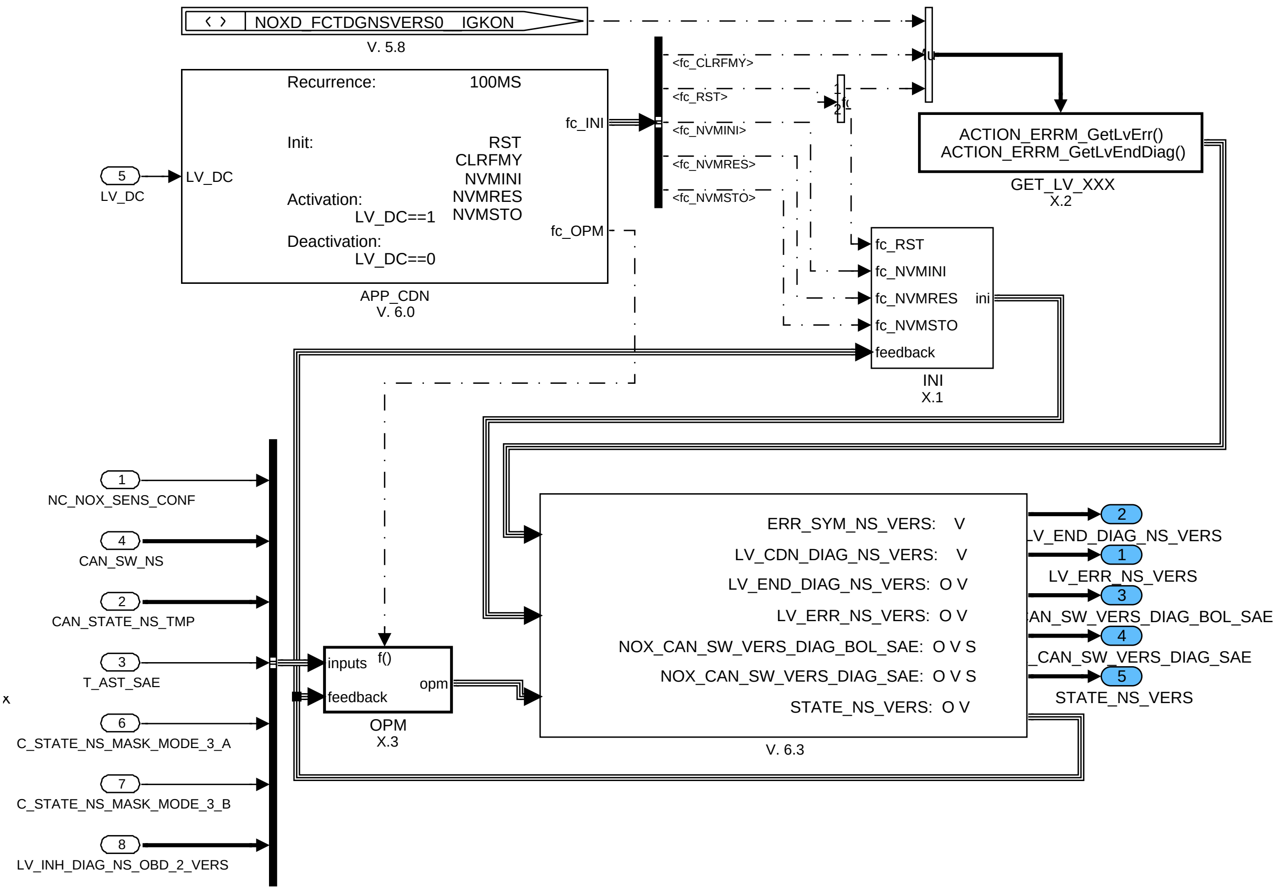
This is evaluation of NOx sensor software version. In case of too less version, an error shall be generated.
Meaning of states:

PAS passive state, no actions are executed
REQ request for NOx sensor software version
CHK check of the NOx sensor software version and calculation of diagnosis values
END diagnosis finished

Application Conditions

Initialization: RST, CLRFMY, NVMINI, NVMRES, NVMSTO
Recurrence: 100MS
Activation: LV_DC==1
Deactivation: LV_DC==0

Function description



SDA_SRS / SDA V 5.2 06−Dec−2006

Figure 214:

52.21.1 Initialization and Management of nonvolatile memory

52.21.1.1 Initialization at Reset

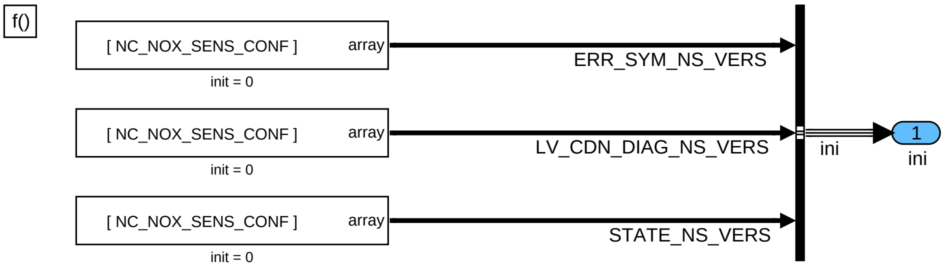


Figure 215:

52.21.1.2 Initialization of NVMY data

All Mode 6 values will be stored in the nonvolatile memory. The values will be reset to zero, if failure memory will be cleared.

52.21.2 Synchronization of the LV_ERR_NS_VERS and LV_END_DIAG_NS_VERS

Synchronisation between LV_ERR_NS_VERS and its result from ERRM is done using action ACTION_ERRM_GetLvErr() similarly synchronisation between LV_END_DIAG_NS_ VERS and its result from ERRM is done using action ACTION_ERRM_GetLvEndDiag()at RESET ,IGKON and CLRFMY events.

52.21.3 Operation at 100ms

52.21.3.1 Introduction of multiple sensor system

52.21.3.1.1 Calculation of the diagnosis results

52.21.3.1.1.1 Determination of diagnostic condition

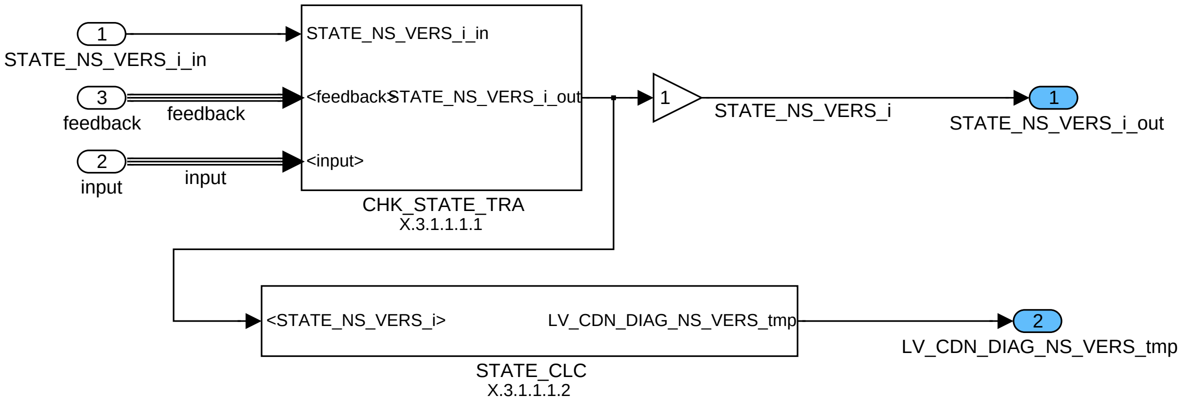


Figure 216:

52.21.3.1.1.1.1 Check transition conditions of STATE_NS_VERS

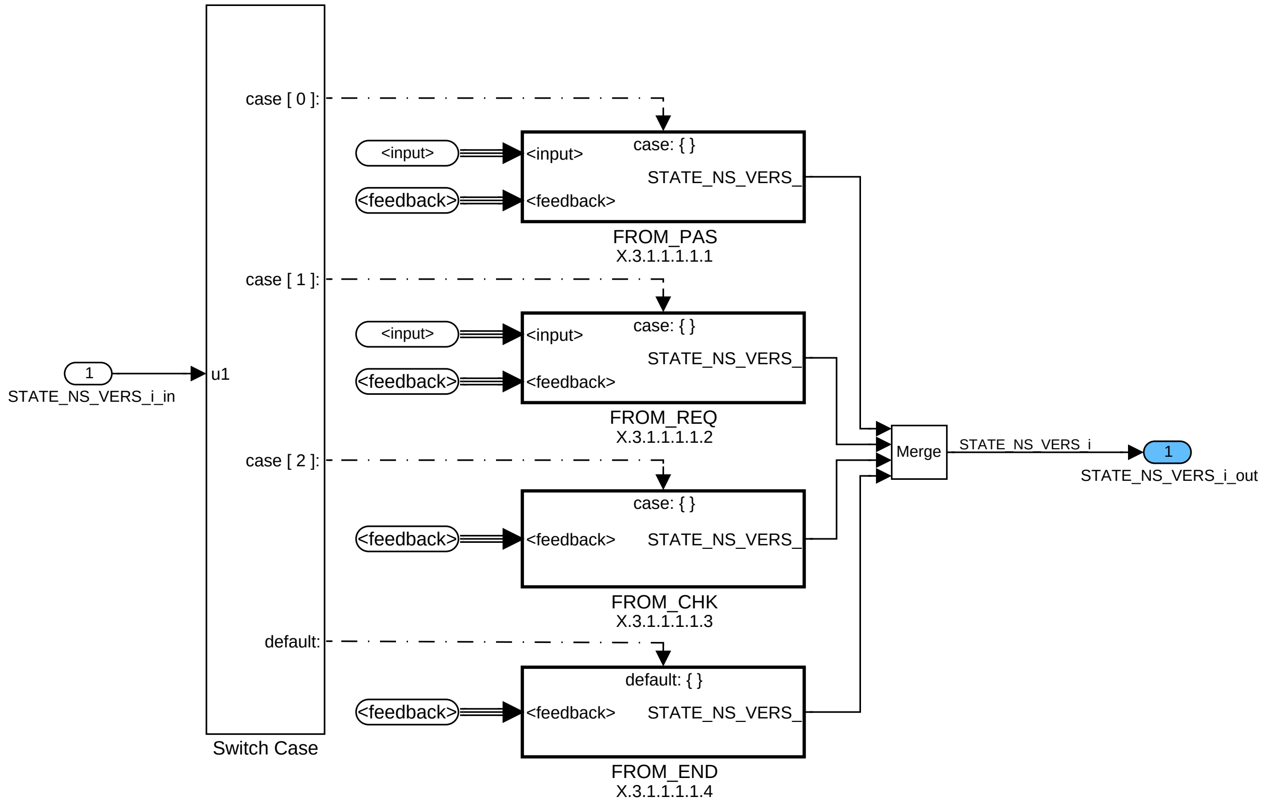


Figure 217:

52.21.3.1.1.1.1.1 From PASSIVE

52.21.3.1.1.1.1.1.1 Check transition conditions from PASSIVE and Transition PASSIVE to REQUEST actions

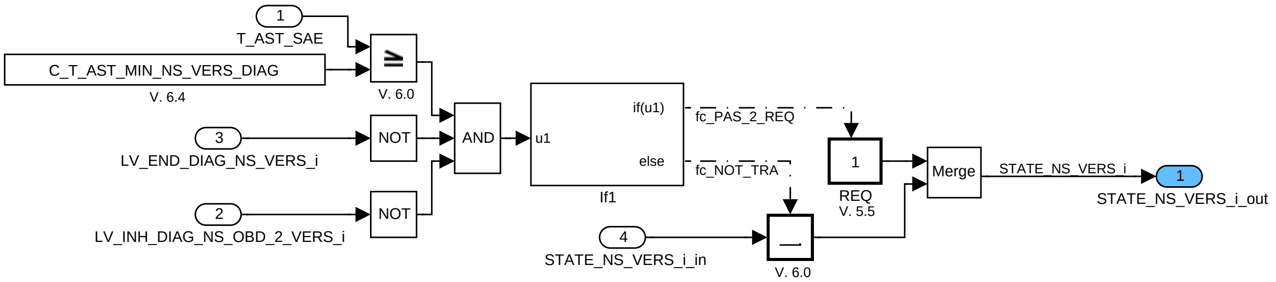


Figure 218:

52.21.3.1.1.1.1.2 From REQUEST

52.21.3.1.1.1.1.2.1 Check transition conditions from REQUEST and Transition REQUEST to CHECK actions

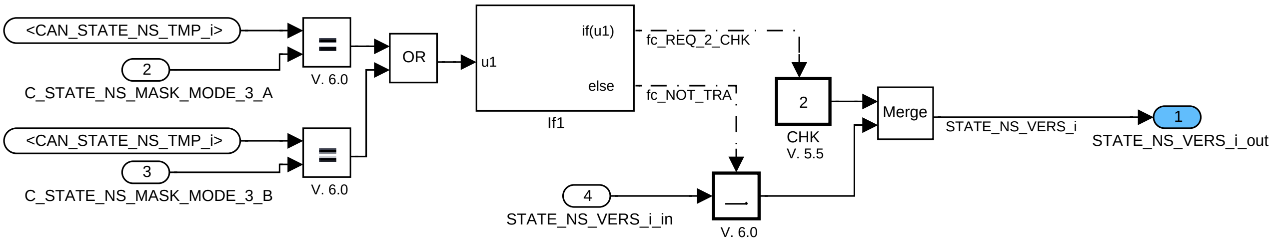


Figure 219:

52.21.3.1.1.1.1.3 From CHECK

52.21.3.1.1.1.1.3.1 Check transition conditions from CHECK and Transition CHECK to END actions

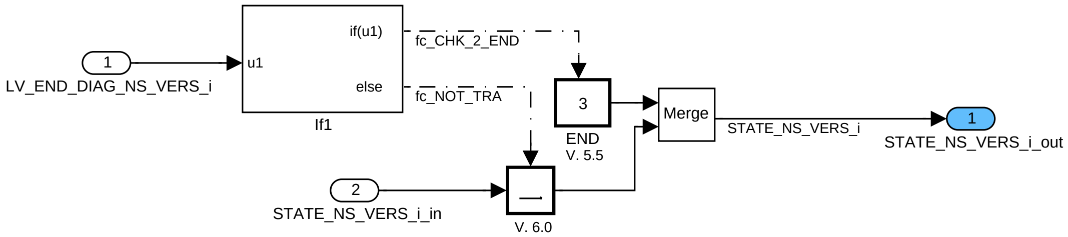


Figure 220:

52.21.3.1.1.1.1.4 From END

52.21.3.1.1.1.1.4.1 Check transition conditions from END and Transition END to PASSIVE actions

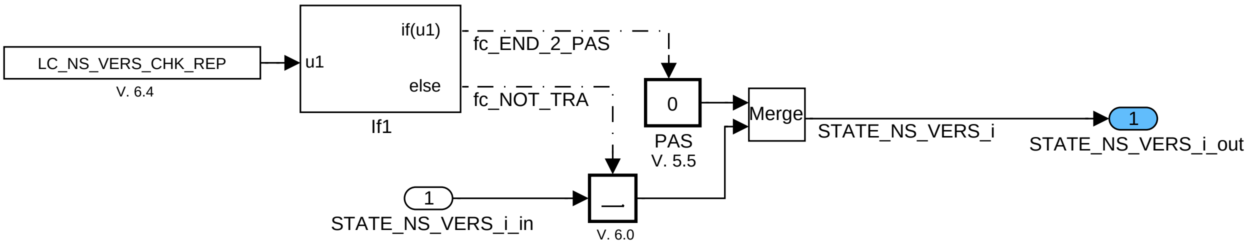


Figure 221:

52.21.3.1.1.1.2 Actions calculation of STATE_NS_VERS

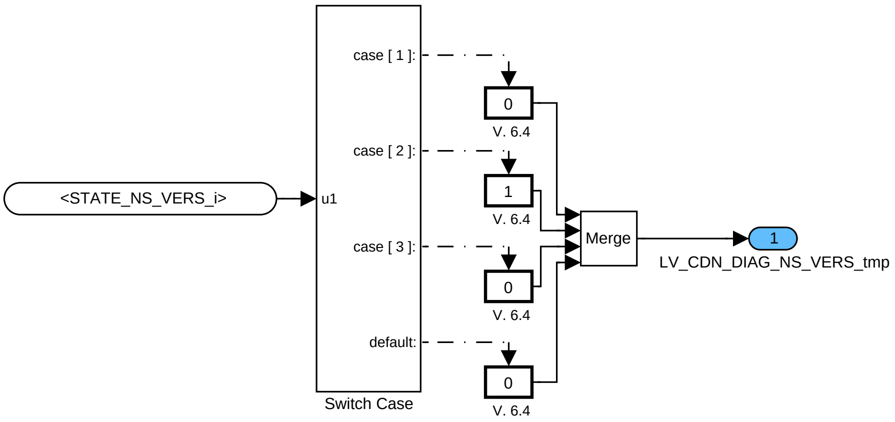


Figure 222:

52.21.3.1.1.2 Determination of error symptom

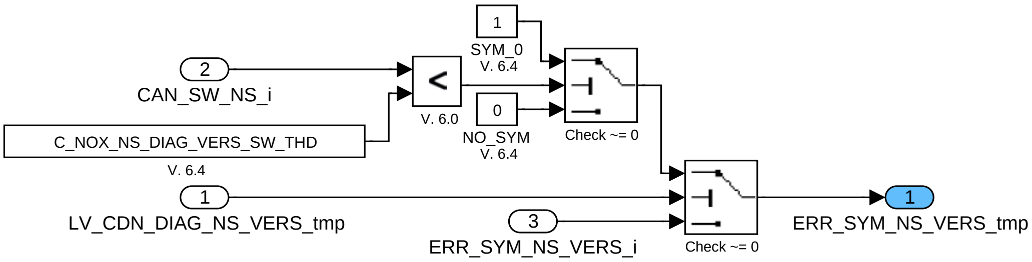


Figure 223:

52.21.3.1.1.3 Call for ERRM

Import actions:

ACTION_ERRM_FilterSymptom(IN<XX>, IN<lv_cdn_diag_XX>, IN<err_sym_XX >, IN<C_ABC_INC_XX >, IN<C_ABC_DEC_XX >, IN<C_ABC_MAX_XX >, OUT<LV_ERR_XX>)
The action computes the elementary anti-bounce filter for one failure treatment and returns filter results

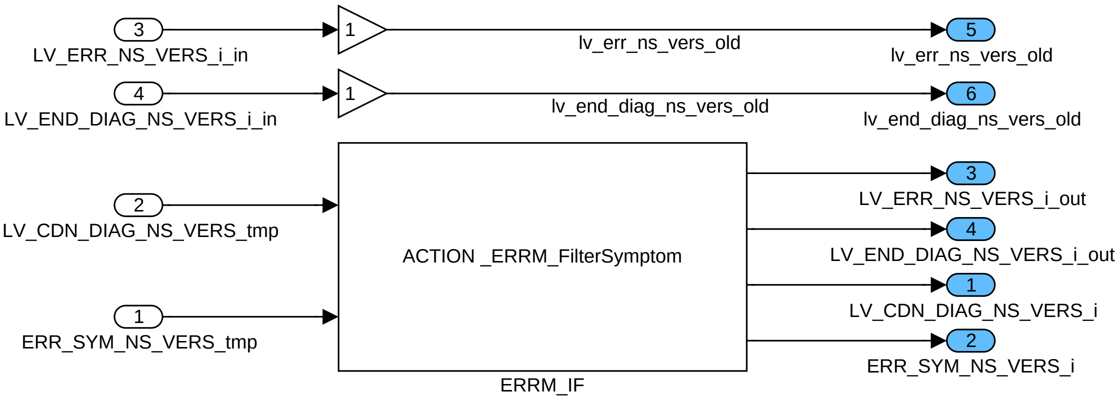


Figure 224:

52.21.3.1.1.4 Diagnosis Mode 6 handling

Mode 6 values will be stored, if error management is finishing the diagnosis or is setting an error.

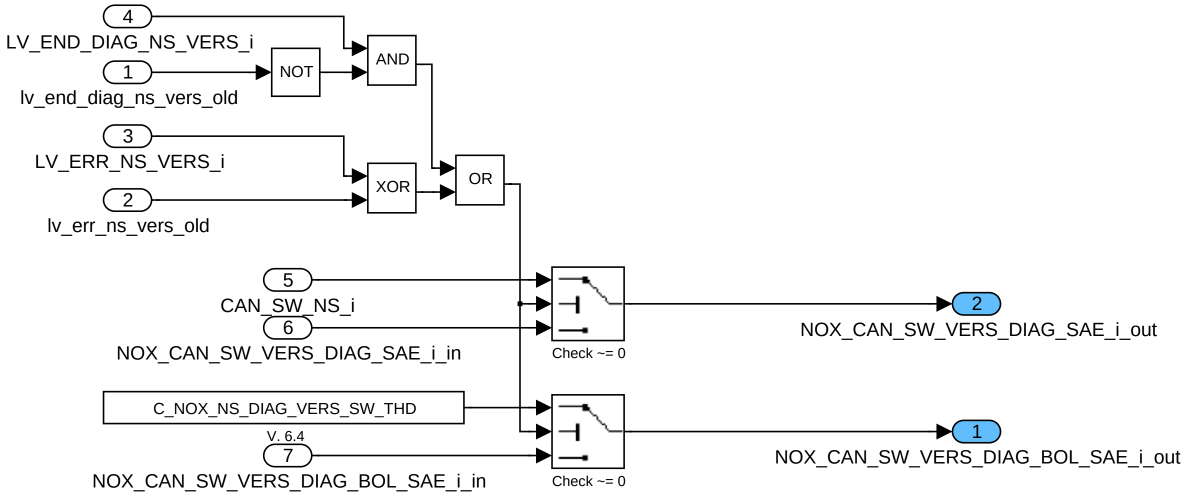


Figure 225:

52.22 NOx sensor OBDII diag Signal offset

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_OFS[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor offset diagnosis					
LV_ERR_NS_OFS[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of NOx sensor offset diagnosis					
TCC_NS_OFS[NC_NOX_SENS_CONF]	O/V	0...FFH	0...255	1	-
Maximum test cycle counter value; used for detection of end of diagnosis					
CTR_RGN_INH_OFS_DIAG	O/V/S	0...FFH	0...255	1	-
Counter for regenerations after a too long lean phase before NOx sensor offset diagnosis will be enabled					
LV_DIAG_NS_OFS_VLD	V	0...1H	0...1	1	-
Offset diagnosis result is valid if the NOx raw emissions are low enough to store them in the NOx catalyst					
LV_SENS_AFR_INH_OFS_DIAG	V	0...1H	0...1	1	-
downstream sensor measures rich exhaust gas on all branches to detect successful regeneration					
T_AFL_INH_OFS_DIAG	V	0...FFFFH	0...655.35	0.01	s
elapsed lean time since last successful regeneration					
T_NOX_OFS_DIAG_MIN[NC_NOX_SENS_CONF]	V	0...FFFFH	0...655.35	0.01	s
Time for active minimum observation for NOx signal offset diagnosis					
ERR_SYM_NS_OFS[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor offset diagnosis					
LV_CDN_DIAG_NS_OFS[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of NOx sensor offset diagnosis					
NOX_OFS_LOAD_DIAG[NC_NOX_SENS_CONF]	V	FF9C...5DCH	-100...1.5E+3	1	ppm
NOx signal minimum for offset diagnosis					

Input data:

CTR_NT_AGI_AD_CMPL_SUM	LV_INH_DIAG_NS_OBD_2_OFS[NC_NOX_SENS_CONF]	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NOX_NS_DIAG_VLD[NC_NOX_SENS_CONF]
LV_NT_HOM_INI	LV_SENS_AFR[NC_NT_NR]	NC_IDX_DIAG_NS_OFS[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
NC_NT_NR	NOX_COR_RED	NOX_NS_AD[NC_NOX_SENS_CONF]	NOX_NS[NC_NOX_SENS_CONF]
NOX_NS_DIAG[NC_NOX_SENS_CONF]	NT_AGI	NT_AGI_SUL	NT_AGI_SUL_SNG[NC_NT_NR]
NT_SUL	NT_SUL_32[NC_NT_NR]	NTLD	STATE_NOX
TNT_MDL_H	TNT_MDL_L	T_AFL	T_NOX_NS_MDL[NC_NOX_SENS_CONF]
T_NOX_NS_OSC[NC_NOX_SENS_CONF]			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_OFS	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor offset diagnosis					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_NS_OFS	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor offset diagnosis					
C_ABC_MAX_NS_OFS	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx sensor offset diagnosis					
C_CTR_NT_AGI_AD_CMPL_SUM_MIN	1	0...FFFFH	0...6.5535E+4	1	-
Minimum completed sulphur adaptations for set free of NOx signal offset diagnosis					
C_CTR_RGN_INH_OFS_DIAG	1	0...FFH	0...255	1	-
Number of regenerations after a too long lean phase before NOx sensor offset diagnosis will be enabled					
C_NOX_MAX_NS_DIAG_OFS	1	0...FFFFH	0...1.02398E+3	0.015625	mg/s
NOx raw emission limit to allow NOx sensor offset diagnosis					
C_NOX_NS_OFS_MAX_THD	1	FF9C...5DCH	-100...1.5E+3	1	ppm
NOx signal offset threshold for positive NOx signal values					
C_NOX_NS_OFS_MIN_THD	1	FF9C...5DCH	-100...1.5E+3	1	ppm
NOx signal offset threshold for negative NOx signal values					
C_NOX_OFS_DIAG_SEL	1	0H 1H 2H	NOX_NS_AD NOX_NS NOX_NS_DIAG	1	-
Selection of diagnosis signal					
C_NOX_OFS_LOAD_OK_MAX	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Positive NOx signal offset threshold for counting the diagnosis result as "ok"					
C_NOX_OFS_LOAD_OK_MIN	1	FF9C...5DCH	-100...1.5E+3	1	ppm
Negative NOx signal offset threshold for counting the diagnosis result as "ok"					
C_NTLD_INH_OFS_DIAG	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Maximum NOx trap loading degree to detect too long lean phase					
C_NT_AGI_NOX_OFS_DIAG_MIN	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Threshold for NOx trap aging factor					
C_NT_AGI_SUL_NOX_OFS_MIN	1	0...FFFFFFFFH	0...1	2.3283E-10	-
Minimum NOx catalyst aging factor due to sulphur for set free of NOx signal offset diagnosis					
C_NT_SUL_NOX_OFS_MAX	1	0...FFFFH	0...1.04856E+4	0.16	mg
Maximum NOx catalyst sulphur load for set free of NOx signal offset diagnosis					
C_SYM_NS_OFS	1	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor offset diagnosis (see NOx sensor OBDII diag - Appl. Inc.)					
C_TCC_NS_OFS_MAX	1	0...FFH	0...255	1	-
Maximum test cycle counter value; used for detection of end of diagnosis					
C_TNT_MDL_NOX_OFS_DIAG_MAX	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Maximum threshold for modelled NOx catalyst monolith temperatures					
C_TNT_MDL_NOX_OFS_DIAG_MIN	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Minimum threshold for modelled NOx catalyst monolith temperatures					
C_T_AFL_INH_OFS_DIAG	1	0...FFFFH	0...655.35	0.01	s
Maximum elapsed lean time since last successful regeneration to detect too long lean phase					
C_T_AFL_NOX_OFS_DIAG_MIN	1	0...FFFFH	0...1.3107E+3	0.02	s
Threshold for lean mixture cycle time					
C_T_AFL_NOX_OFS_NEG_MIN	1	0...FFFFH	0...1.3107E+3	0.02	s
Minimum lean engine operation time for evaluation of negative NOx signal offset threshold					
C_T_AFL_NOX_OFS_OK_MIN	1	0...FFFFH	0...1.3107E+3	0.02	s
Minimum lean engine operation time for counting the diagnosis result as "ok"					
C_T_NOX_NS_DIAG_OFS	1	0...FFFFH	0...655.35	0.01	s
Threshold for timer T_NOX_NS_MDL to activate minimum aquisition					
C_T_NOX_NS_OSC_OFS_DIAG	1	0...FFFFH	0...655.35	0.01	s
Waiting time after last detection of an oscillation of NOx signal for offset diagnosis					
C_T_NOX_OFS_DIAG_MIN	1	0...FFFFH	0...655.35	0.01	s
Minimum time for active minimum observation for NOx signal offset diagnosis					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_CTR_RGN_INH_OFS_DIAG_RST	1	0...1H	0...1	1	-
counter of valid regenerations will be reset at initialization					
LC_NT_HOM_INI_DIAG_SEL	1	0...1H	0...1	1	-
manual flag: NOx trap initialised due to sufficient hom. stoic. operation (accumulation effects eliminated)					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

52.22.1 General information

This diagnosis checks the NOx signal behaviour after the start of a NOx catalyst “LOAD” phase. The use of NOx signal for generation of a NOx catalyst regeneration request is blocked for a limited time after the start of a NOx catalyst “LOAD” phase. This is necessary because of NH3 cross sensitivity of sensor. This cross sensitivity pushes up the NOx signal up to the upper limit at the end of a NOx catalyst regeneration. Hence the sensor needs a recovery time after the start of next NOx catalyst “LOAD” phase. During this recovery time the NOx signal must go to a level which is lower than the threshold for start of a new regeneration. Otherwise it is a sensor failure present.

Application Condition

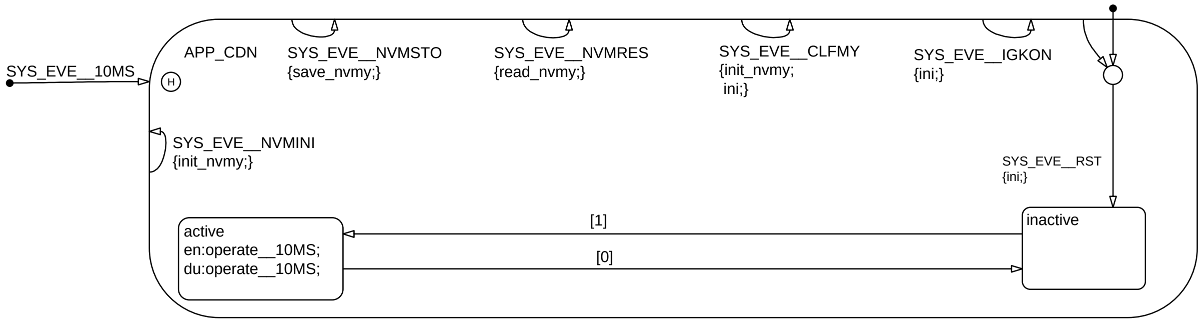
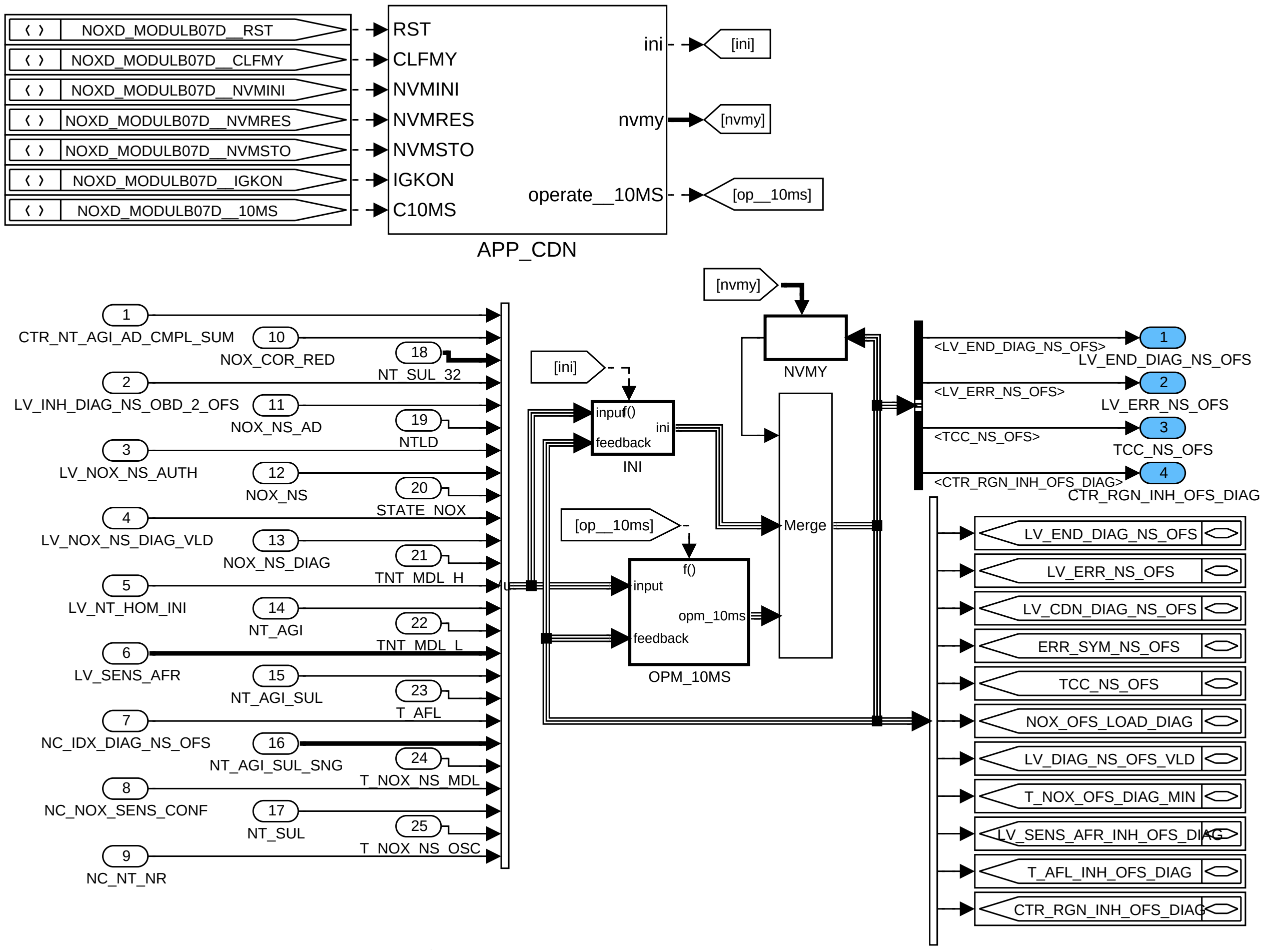


Figure 226 NOXD_MODULB07D/ APP_CDN/ Chart



SDA_SRS / SDA 4.0 22−Nov−2006

Figure 227 NOXD_MODULB07D

52.22.1.1 Management of non volatile memory

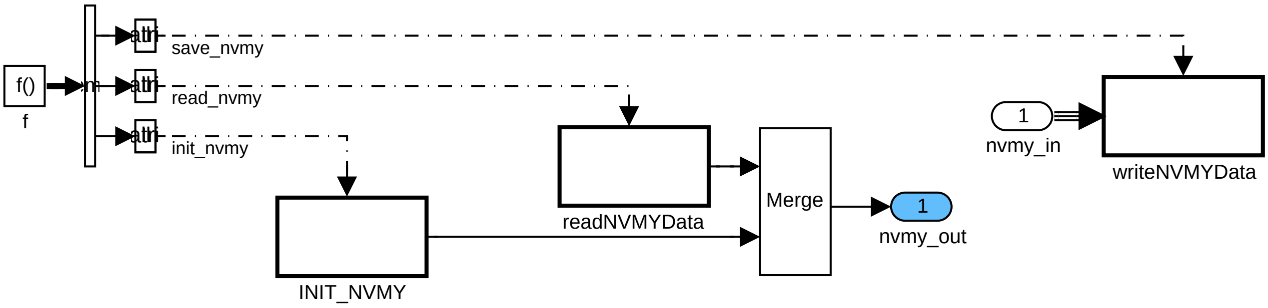


Figure 228 NOXD_MODULB07D/ NVMY

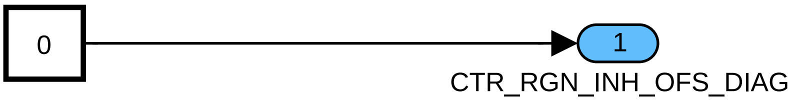


Figure 229 NOXD_MODULB07D/ NVMY/ INIT_NVMY/ CLC

52.22.1.2 Initialization

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >)

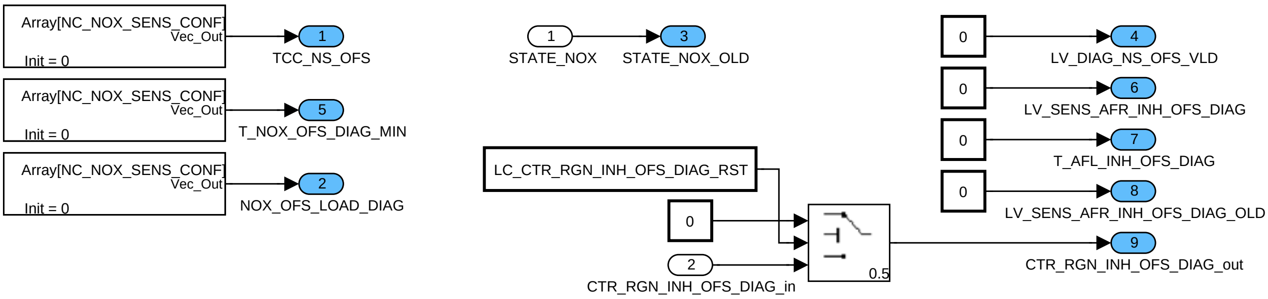


Figure 230 NOXD_MODULB07D/ INI/ CLC

52.22.1.3 Loop control

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

The calculation inside NO_LOOP have to be done only once.

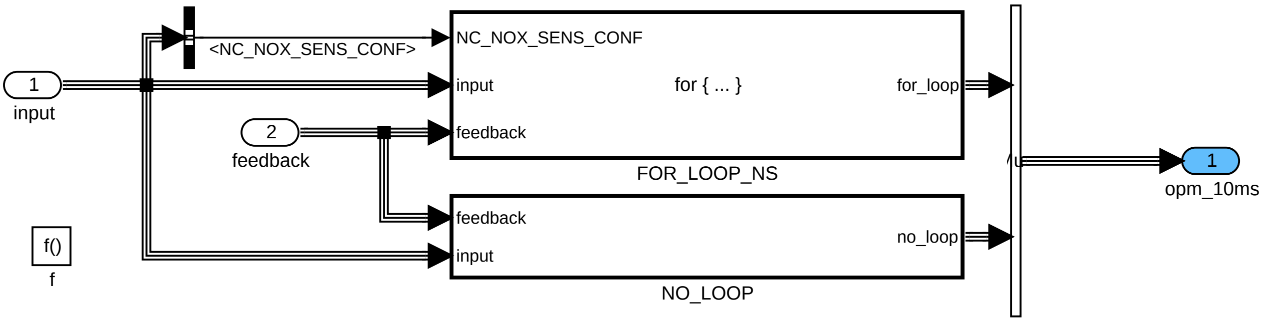


Figure 231 NOXD_MODULB07D/ OPM_10MS

Calculation for each NOx sensor

Minimum search

The minimum of the selected NOx signal during the load phase will be calculated.

NOX_NS_AD:

The signal NOX_NS_AD_i is a 'strong' filtered and offset adapted NOx signal. The lower limit of NOX_NS_AD_i value range is 0 ppm. Hence this selection does not allow to diagnose a negative NOx signal offset.

NOX_NS:

The signal NOX_NS_i is a 'strong' filtered NOx signal. This selection allows to diagnose a negative NOx signal offset.

NOX_NS_DIAG:

The signal NOX_NS_DIAG_i is a 'light' filtered NOx signal. This selection allows to diagnose a negative NOx signal offset.

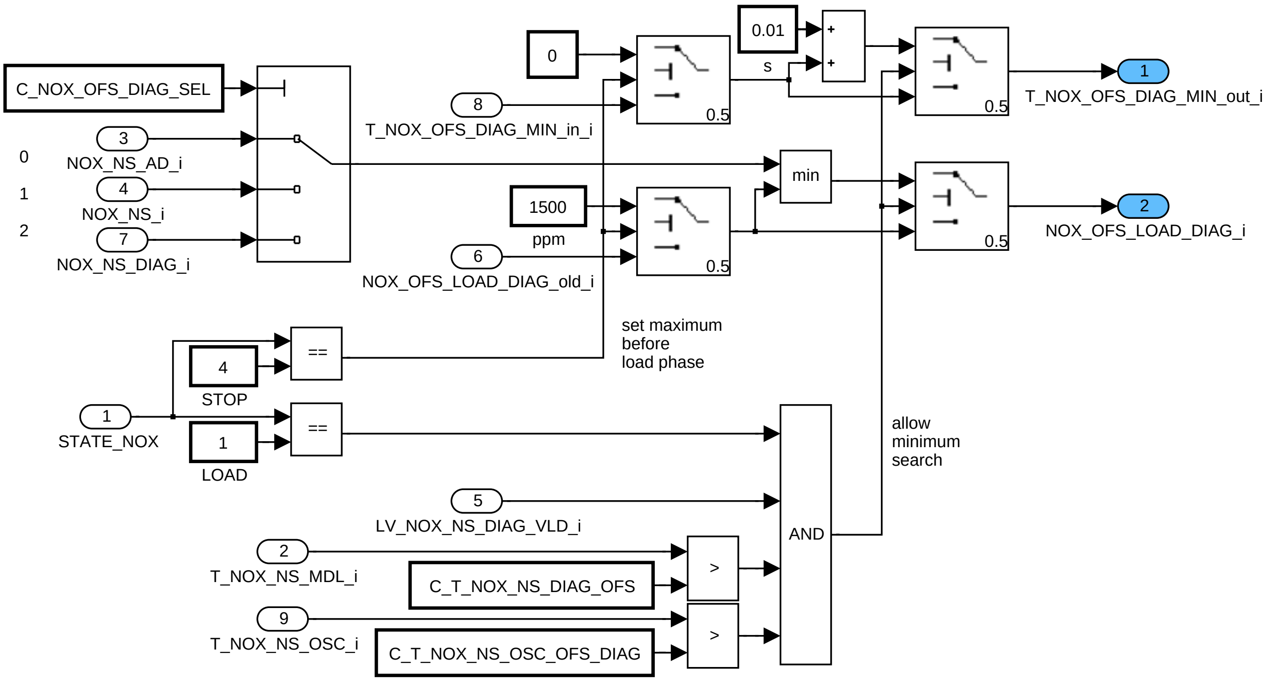


Figure 232 NOXD_MODULB07D/ OPM_10MS/ FOR_LOOP_NS/ CLC1

Condition for evaluation of the diagnosis result

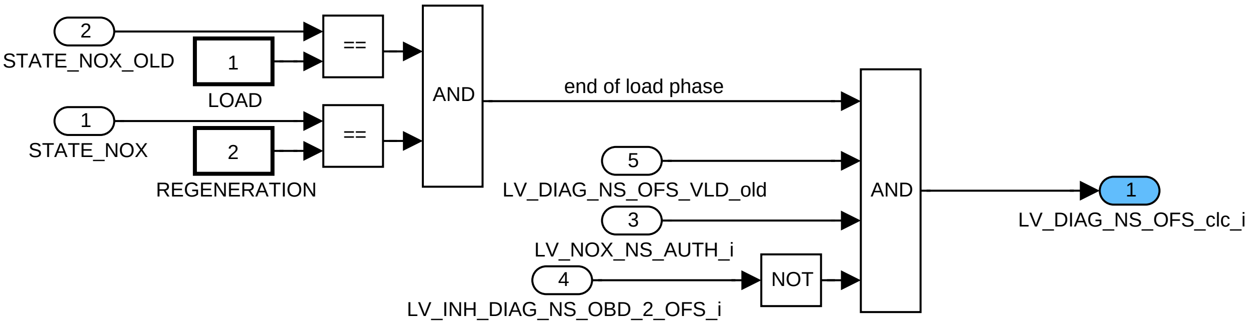
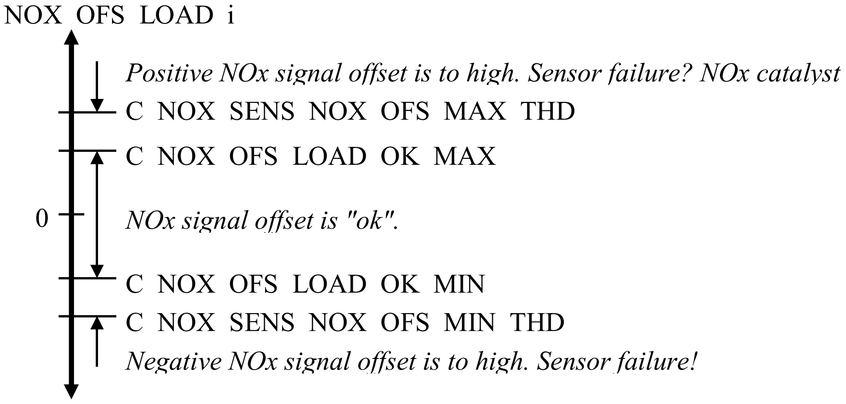


Figure 233 NOXD_MODULB07D/ OPM_10MS/ FOR_LOOP_NS/ CLC2

Minimum validation

At the end of the load phase the minimum of the NOx signal will be evaluated to get the result if the sensor is OK.



Application hint: The constant C_NOX_NS_OFS_MIN_THD shall be set to a negative value. So it is possible to detect a high negative NOx signal offset.

Application hint: The constant C_SYM_NS_OFS has to be set to the symptom as described in NOx sensor OBDII (Appl. Inc.).

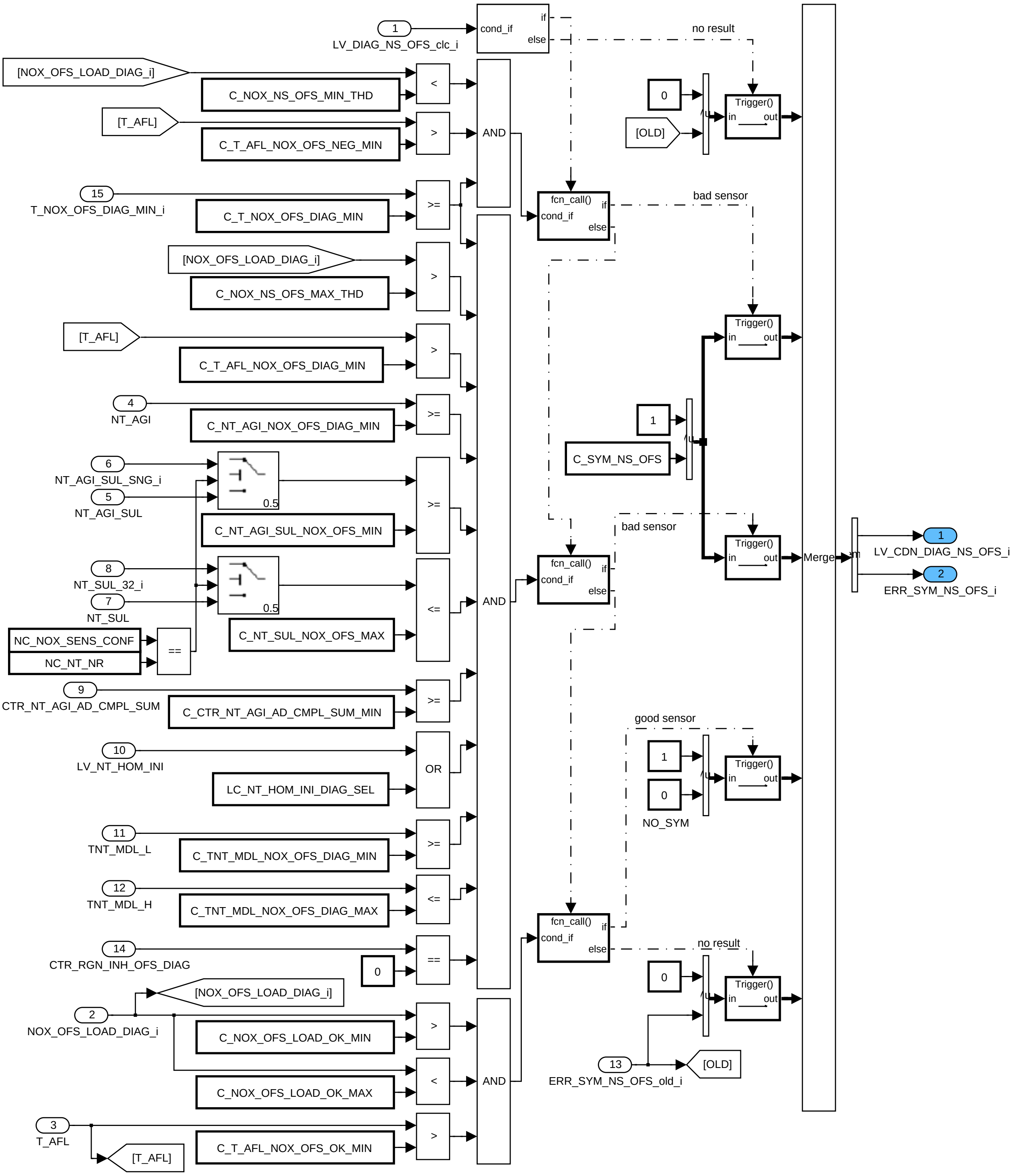


Figure 234 NOXD_MODULB07D/ OPM_10MS/ FOR_LOOP_NS/ CLC3

Test cycle counter

The test cycle counter TCC_NS_OFS_i counts the number of completed offset diagnoses. This information is needed for detection of diagnosis readiness and decrementation of counter for consecutive driving cycles with or without a NOx sensor failure.

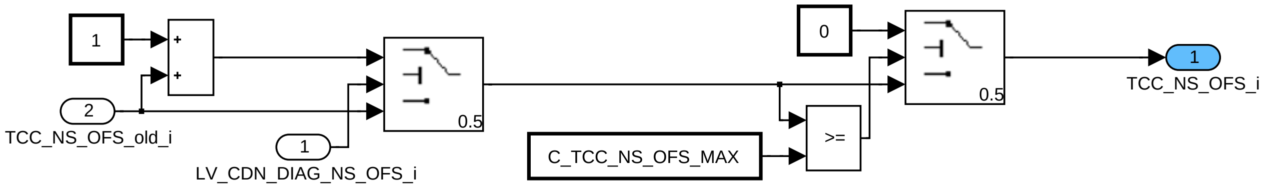


Figure 235 NOXD_MODULB07D/ OPM_10MS/ FOR_LOOP_NS/ CLC4

Error management

LV_ERR_NS_OFS will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_OFS will be get by ACTION_ERRM_GetLvEndDiag.

Figure 236 NOXD_MODULB07D/ OPM_10MS/ FOR_LOOP_NS/ CLC5

Bank independent calculation

Observation of valid LOAD-REGERNERATION cycle

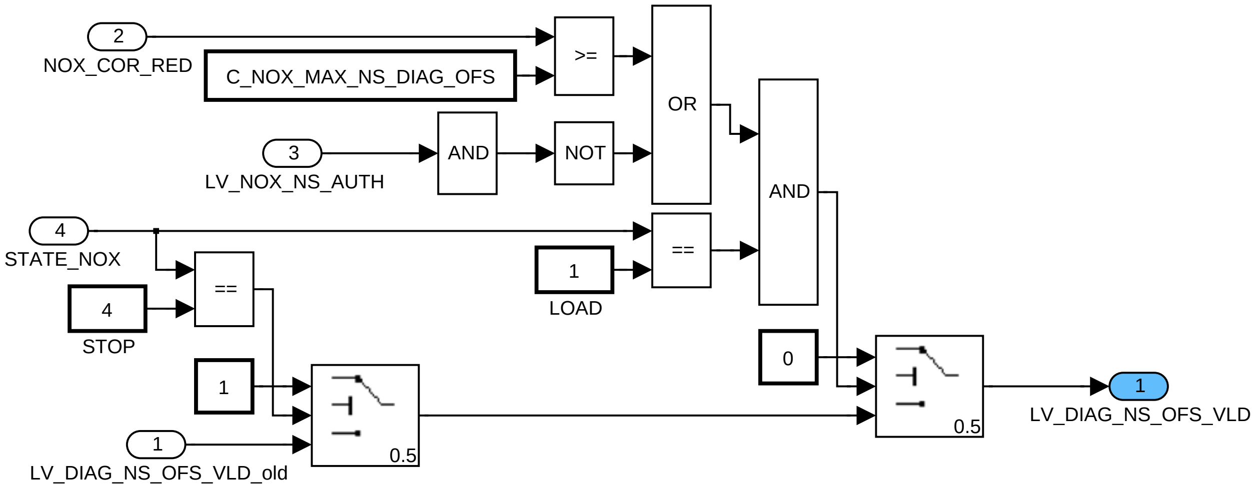


Figure 237 NOXD_MODULB07D/ OPM_10MS/ NO_LOOP/ CLC1

Detection of valid regeneration of all NOx traps

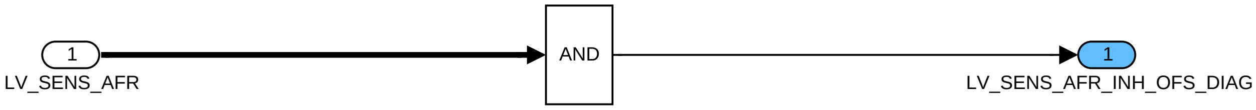


Figure 238 NOXD_MODULB07D/ OPM_10MS/ NO_LOOP/ CLC2

Lean time between two valid regenerations

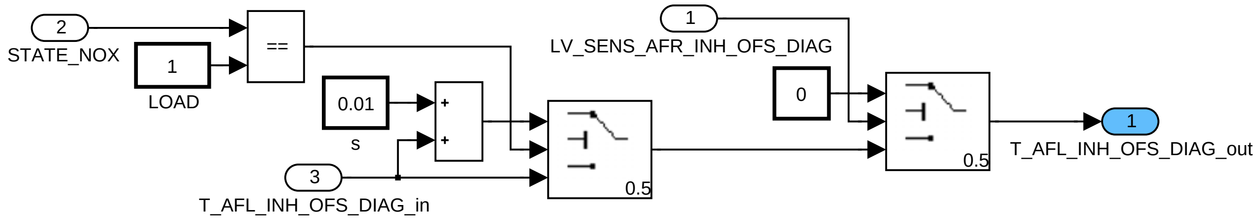


Figure 239 NOXD_MODULB07D/ OPM_10MS/ NO_LOOP/ CLC3

Counter of valid regenerations after a too long lean phase

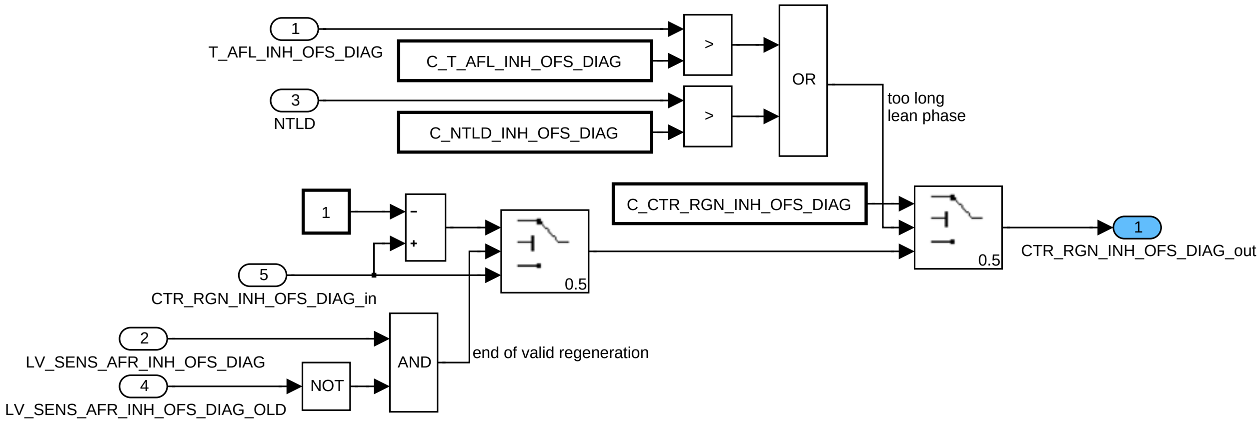


Figure 240 NOXD_MODULB07D/ OPM_10MS/ NO_LOOP/ CLC4

Old values for edge detection

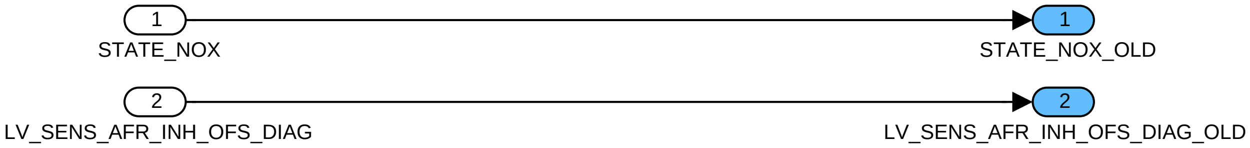


Figure 241 NOXD_MODULB07D/ OPM_10MS/ NO_LOOP/ CLC5

52.23 NOx sensor OBDII diag Binary lambda signal dynamic

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NS_VLS_DYN[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor dynamic diagnosis of binary signal					
LV_ERR_NS_VLS_DYN[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of NOx sensor dynamic diagnosis of binary signal					
ERR_SYM_NS_VLS_DYN[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor dynamic diagnosis of binary signal					
LV_CDN_DIAG_NS_VLS_DYN[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of NOx sensor dynamic diagnosis of binary signal					
STATE_VLS_RGN_NT_DIAG[NC_NOX_SENS_CONF]	V	0H 1H 2H	WAIT INIT DIAG	1	-
State of dynamic diagnosis for binary signal					
T_VLS_NS_SWI[NC_NOX_SENS_CONF]	V	0...FFH	0...5.1	0.02	s
Timer for signal switch time measurement					
T_WAIT_VLS_DYN[NC_NOX_SENS_CONF]	V	0...FFH	0...5.1	0.02	s
Timer for waiting of stratified mode activation					
VLS_NS_RGN_NT_END[NC_NOX_SENS_CONF]	V	0...578H	-200...1.2E+3	1	mV
NOx sensor binary O2 signal voltage at the time of leaving NOx catalyst regeneration phase					
LV_NT_RGN_REQ_OLD	-	0...1H	0...1	1	-
Old value of LV_NT_RGN_REQ for edge detection					

Input data:

LAMB_LS_UP[NC_CBK_EX_NR]	LV_INH_DIAG_NS_OBD_2_DYN[NC_NOX_SENS_CONF]	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NT_RGN_REQ
LV_NT_RGN_REQ_AD	LV_ST_END	LV_S_ACT	LV_VLS_NS_DIAG_VLD[NC_NOX_SENS_CONF]
MAF_KGH	NC_CBK_EX_NR	NC_IDX_DIAG_NS_VLS_DYN[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
STATE_NOX	T_VLS_NS_HLD[NC_NOX_SENS_CONF]	VLS_NS_DIAG[NC_NOX_SENS_CONF]	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_VLS_DYN	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor dynamic diagnosis of binary signal					
C_ABC_INC_NS_VLS_DYN	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor dynamic diagnosis of binary signal					
C_ABC_MAX_NS_VLS_DYN	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx sensor dynamic diagnosis of binary signal					
C_LAMB_VLS_NS_DYN_MIN	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Minimum upstream WRAF signal value during the dynamic diagnosis					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_VLS_NS_HLD_DYN_DIAG	1	0...7FFFH	0...327.67	0.01	s
Time with binary O2 signal state = "invalid", after which a diagnosis is started					
C_T_WAIT_VLS_DYN_MAX	1	0...FFH	0...5.1	0.02	s
Maximum time for waiting of stratified mode activation					
C_VLS_NS_DIAG_SWI_END	1	0...578H	-200...1.2E+3	1	mV
Binary O2 signal threshold for stop of switch time diagnosis					
C_VLS_NS_DIAG_SWI_ST	1	0...578H	-200...1.2E+3	1	mV
Binary O2 signal threshold for start of switch time diagnosis					
C_VLS_NS_RGN_NT_END_MIN	1	0...578H	-200...1.2E+3	1	mV
Minimum binary O2 signal voltage after a NOx catalyst regeneration for start a switch time diagnosis					
LC_VLS_NS_DYN_VLD_ON	1	0...1H	0...1	1	-
Valid binary O2 signal of NOx-Sensor necessary for this diagnosis					
IP_T_VLS_NS_SWI_MAX	6	0...FFH	0...5.1	0.02	s
LDP_MAF_KGH_IP_T_VLS_NS_SWI_MAX	6	0...FFFFH	0...2.04797E+3	0.03125	kg/h
Diagnosis threshold for binary O2 signal switch time diagnosis					

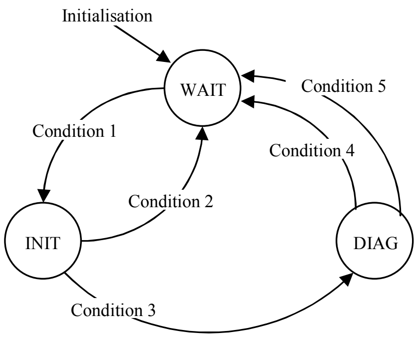
Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

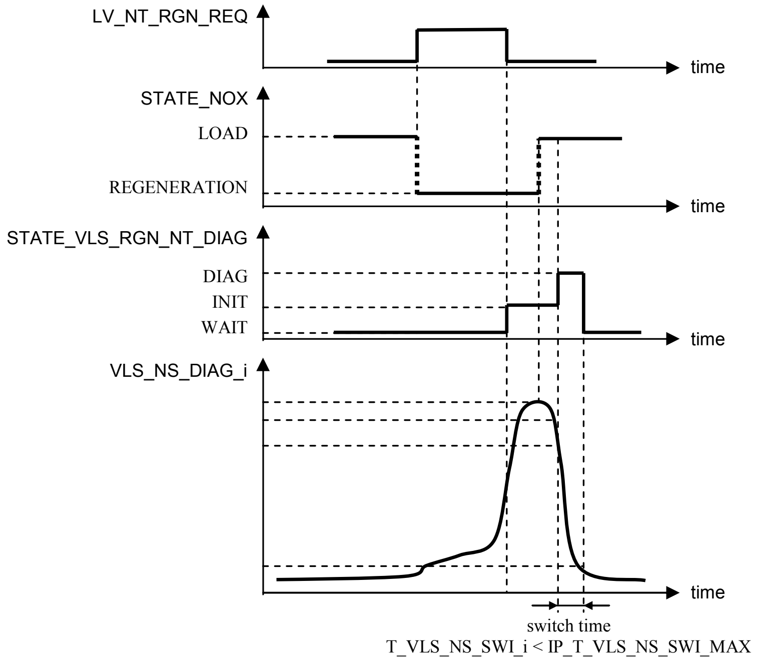
52.23.1 General description

This diagnosis observes the dynamic of the binary O2 signal of NOx sensor after a NOx catalyst regeneration.

Signal flow:



Characteristic signals:



Application Condition

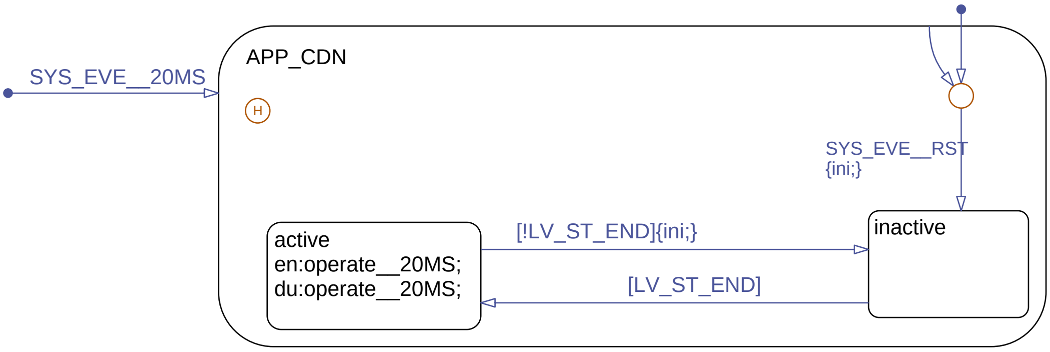
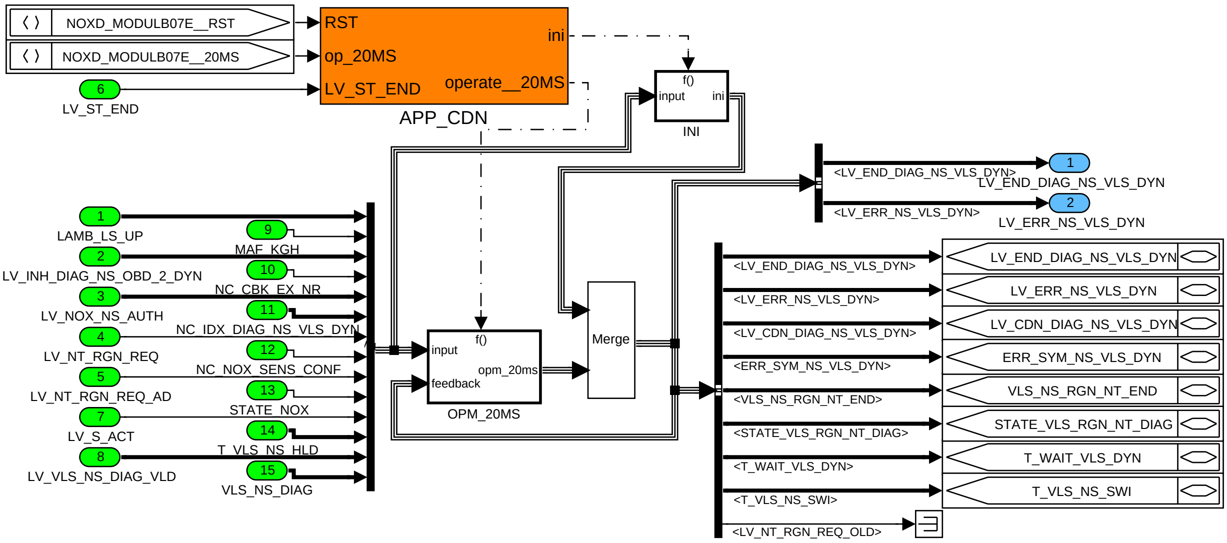


Figure 242 NOXD_MODULB07E/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 15−Dec−2004

Figure 243 NOXD_MODULB07E

52.23.1.1 Initialization at reset and deactivation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >)

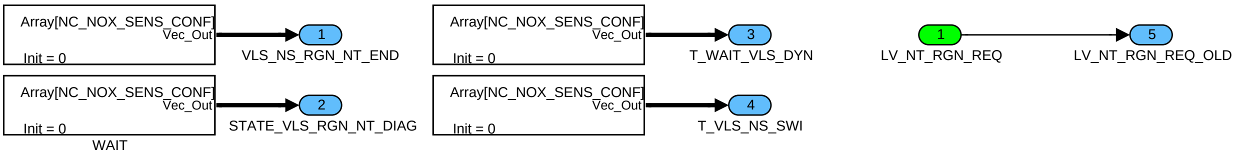


Figure 244 NOXD_MODULB07E/ INI/ CLC

52.23.1.2 Loop control

The calculation inside both NO_LOOP subsystems have to be done only once.

The calculation inside FOR_LOOP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

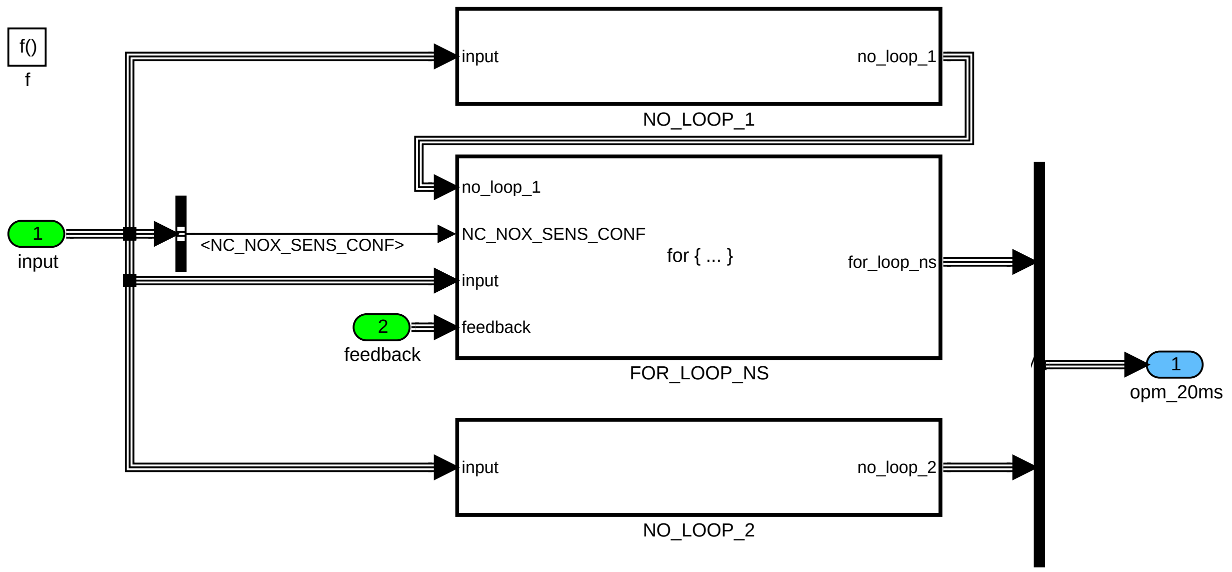


Figure 245 NOXD_MODULB07E/ OPM_20MS

Bank independent calculation (part 1)

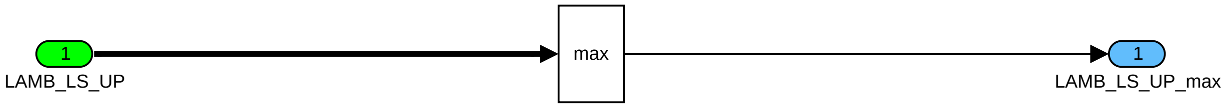


Figure 246 NOXD_MODULB07E/ OPM_20MS/ NO_LOOP_1/ CLC

Calculation for each NOx sensor

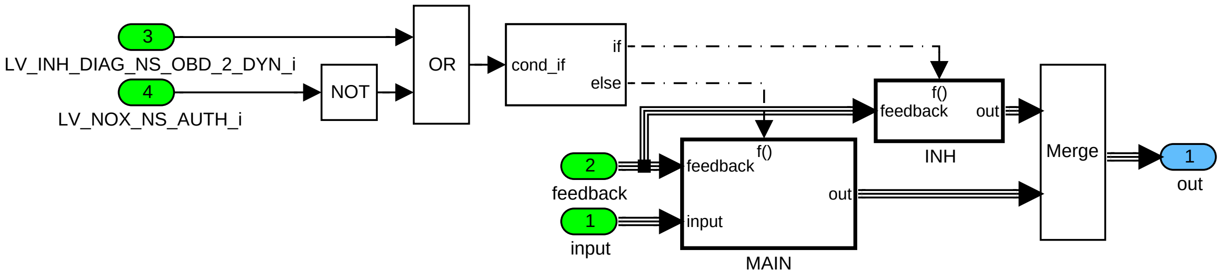


Figure 247 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC

Inhibition of diagnosis for one sensor

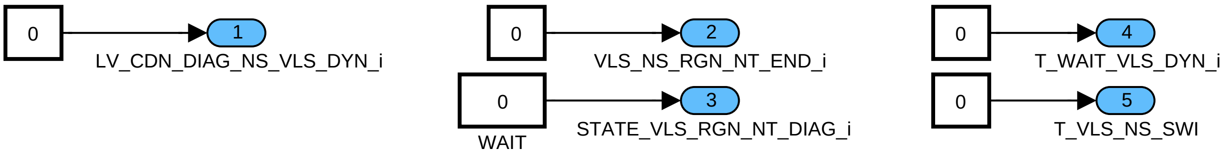


Figure 248 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ INH/ CLC

State machine

This diagnosis is divided into 3 states:

- WAIT: waiting for the end of a regeneration phase of NOx catalyst

- INIT: regeneration phase left, initial conditions for diagnosis present
- DIAG: Diagnosis runs

The default case is is case 0 (WAIT).

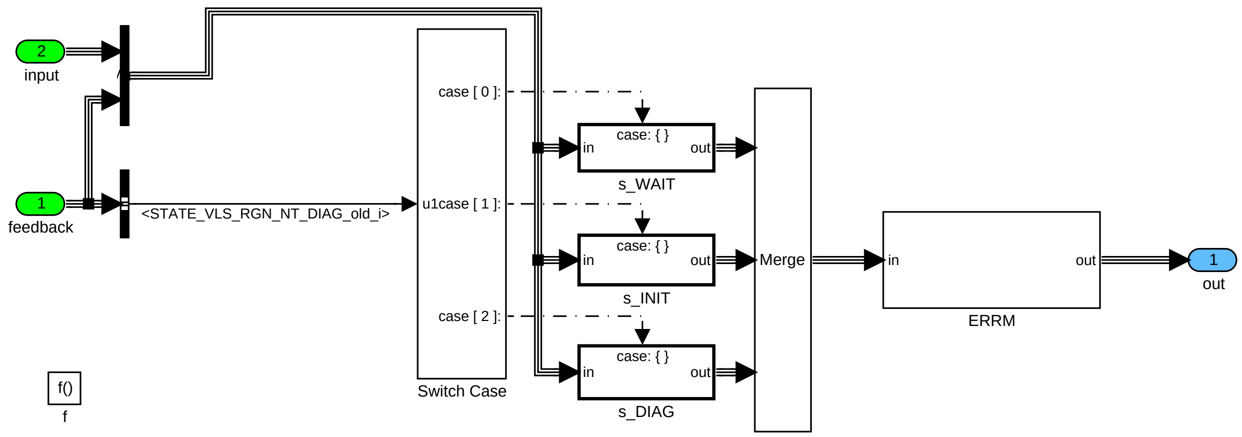


Figure 249 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ MAIN

STATE_VLS_RGN_NT_DIAG_i = 0 (WAIT)

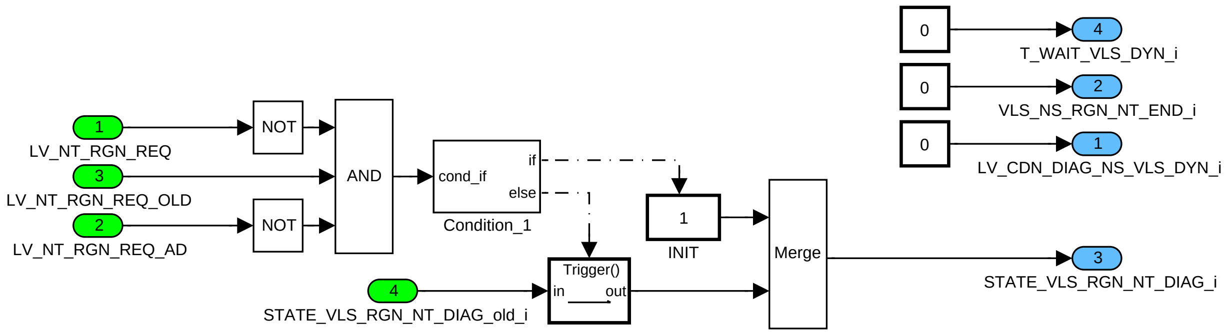


Figure 250 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_WAIT/ CLC

STATE_VLS_RGN_NT_DIAG_i = 1 (INIT)

This timer limits the phase INIT. If no diagnosis will be started in this time, then the state machine switches back to WAIT.

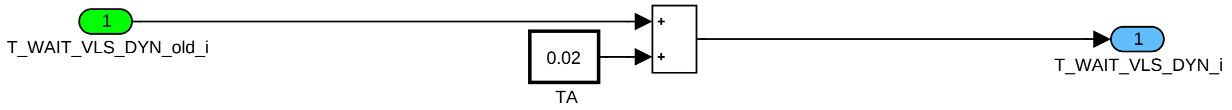


Figure 251 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_INIT/ CLC1

During the waiting the binary O2 signal voltage is checked on their validity. If the signal is valid or if a special condition is fulfilled then the last binary O2 signal value is stored.

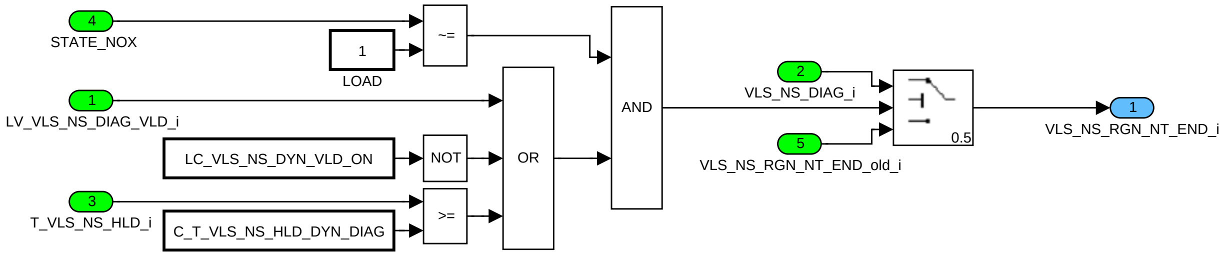


Figure 252 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_INIT/ CLC2

The diagnosis will be started if the binary signal has a suitable behaviour. The threshould C_VLS_NS_DIAG_SWI_ST defines the start of the gradient check.

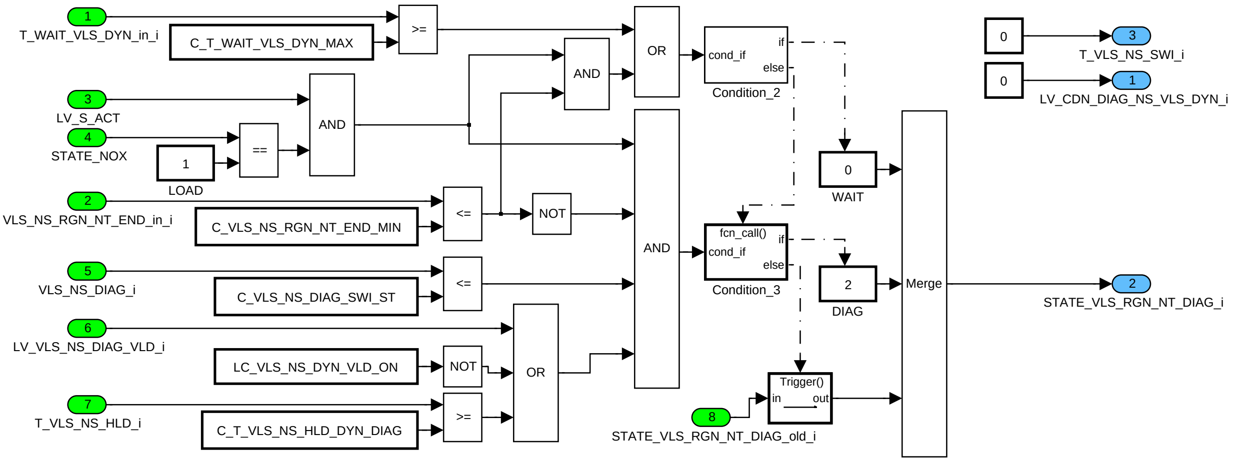


Figure 253 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_INIT/ CLC3

STATE_VLS_RGN_NT_DIAG_i = 2 (DIAG)

The time of the gradient of the binary signal will be measured.

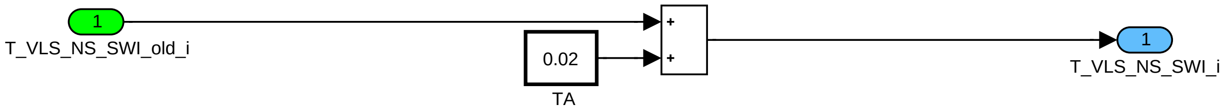


Figure 254 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_DIAG/ CLC1

The threshould C_VLS_NS_DIAG_SWI_END defines the end of the gradient check. The elapsed time for the gradient will be evaluated to decide whether the sensor dynamic is OK or not.

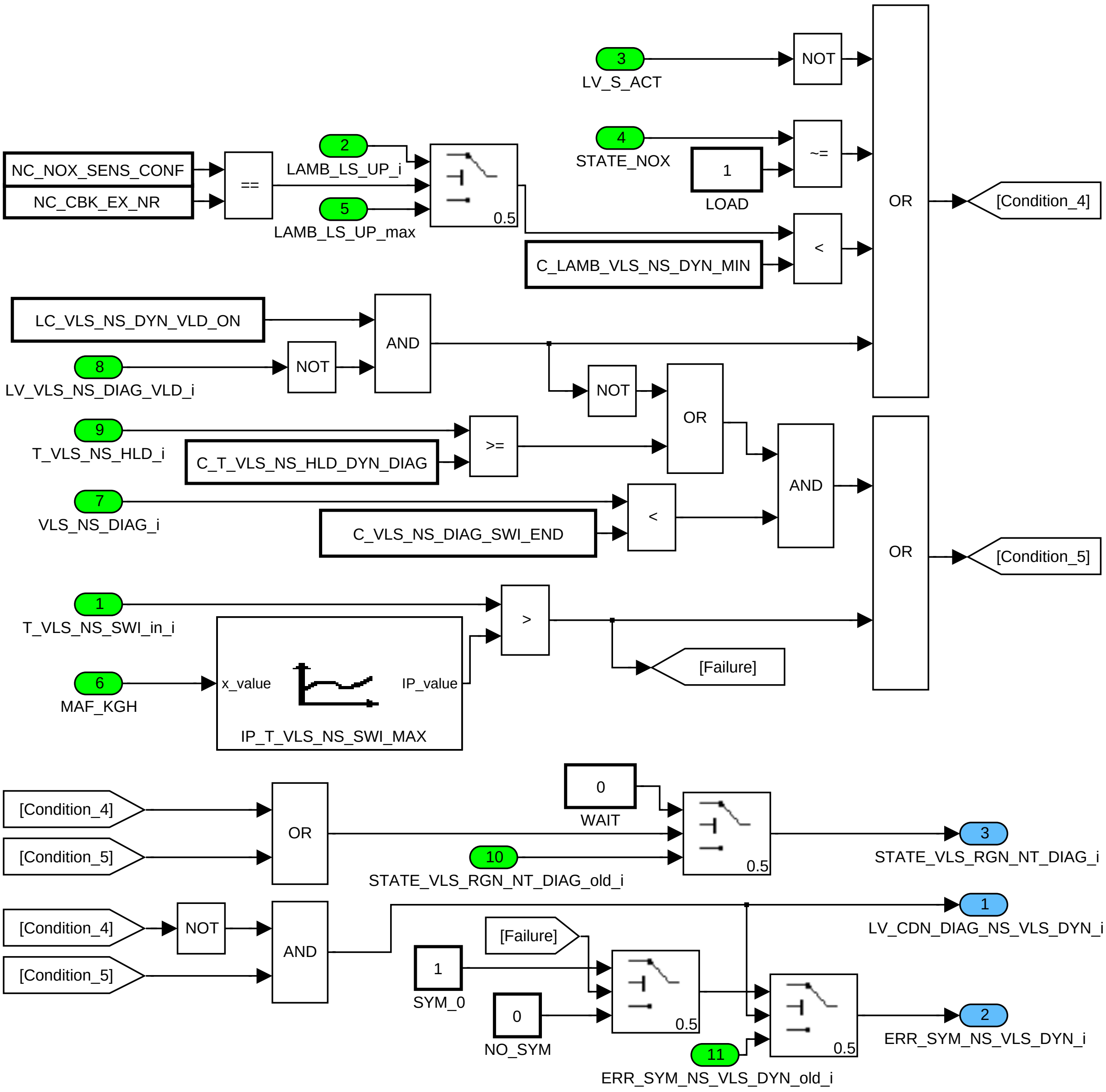


Figure 255 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ MAIN/ s_DIAG/ CLC2

Error management

LV_ERR_NS_VLS_DYN will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_VLS_DYN will be get by ACTION_ERRM_GetLvEndDiag.

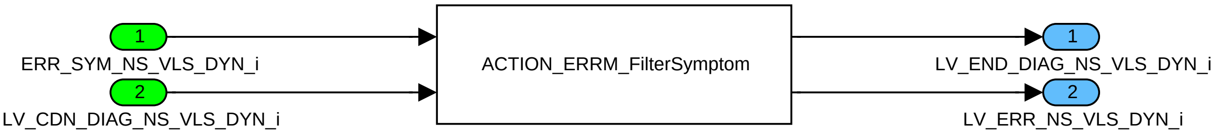


Figure 256 NOXD_MODULB07E/ OPM_20MS/ FOR_LOOP_NS/ CLC/ MAIN/ ERRM/ CLC

Bank independent calculation (part 2)

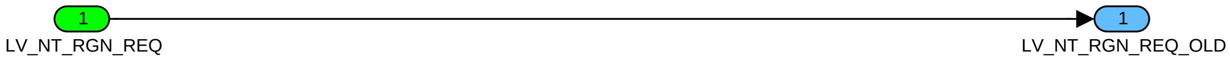


Figure 257 NOXD_MODULB07E/ OPM_20MS/ NO_LOOP_2/ CLC

52.24 NOx sensor OBDII diag Signal activity

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_NOX_NS_ACT[NC_NOX_SENS_CONF]	O/V/S	0...1H	0...1	1	-
NOx signal activity detected					
LV_END_DIAG_NS_ACT[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
End of NOx sensor activity diagnosis					
LV_ERR_NS_ACT[NC_NOX_SENS_CONF]	O/V	0...1H	0...1	1	-
Error flag of NOx sensor activity diagnosis					
CTR_NOX_RGN_DIAG_ACT[NC_NOX_SENS_CONF]	V	0...FFH	0...255	1	-
Number of suitable regenerations during activity diagnosis					
ERR_SYM_NS_ACT[NC_NOX_SENS_CONF]	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx sensor activity diagnosis					
LV_CDN_DIAG_NS_ACT[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Diagnosis condition of NOx sensor activity diagnosis					
LV_NOX_NS_ACT_NVMY[NC_NOX_SENS_CONF]	V	0...1H	0...1	1	-
Value LV_NOX_NS_ACT from last driving cycle stored in non-volatile memory					
LV_NT_RGN_REQ_ACT_OLD	V	0...1H	0...1	1	-
Old value of LV_NT_RGN_REQ for edge detection, used for activity diagnosis					
NOX_NS_DIAG_ACT_GRD_MAX[NC_NOX_SENS_CONF]	V	0...5DCH	0...1.5E+3	1	ppm
Maximum NOx signal gradient during activity diagnosis					
NOX_NS_DIAG_ACT_MAX[NC_NOX_SENS_CONF]	V	0...5DCH	0...1.5E+3	1	ppm
Maximum NOx signal value during activity diagnosis					
NOX_NS_DIAG_ACT_MMV[NC_NOX_SENS_CONF]	V	0...5DCH	0...1.5E+3	1	ppm
Filtered value of NOx concentration measured by NOx sensor for diagnosis functions					
NOX_NS_DIAG_ACT_OLD[NC_NOX_SENS_CONF]	V	FF9C...5DCH	-100...1.5E+3	1	ppm
Old value of NOx concentration value for diagnosis, used for activity diagnosis					
T_NS_DIAG_ACT[NC_NOX_SENS_CONF]	V	0...FFFFH	0...6.5535E+3	0.1	s
Timer for NOx sensor activity diagnosis					
T_RGN_ACT_OLD	V	0...FFFFH	0...655.35	0.01	s
Old value of time counter for regeneration active					

Input data:

LV_INH_DIAG_NS_OBD_2_ACT[NC_NOX_SENS_CONF]	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NT_RGN_REQ	LV_PL
LV_SENS_AFR[NC_NT_NR]	LV_ST_END	NC_IDX_DIAG_NS_ACT[NC_NOX_SENS_CONF]	NC_NOX_SENS_CONF
NC_NT_NR	NOX_NS_DIAG[NC_NOX_SENS_CONF]	T_RGN	TNT_MDL_H
TNT_MDL_L			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NS_ACT	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx sensor activity diagnosis					
C_ABC_INC_NS_ACT	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx sensor activity diagnosis					
C_ABC_MAX_NS_ACT	1	1...FFH	1...255	1	s
Anti bounce counter maximum for the NOx sensor activity diagnosis					
C_CRLC_NOX_NS_DIAG_ACT	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Filter constant for calculation of moving mean value of NOx value for diagnosis					
C_CTR_NOX_RGN_DIAG_ACT_MAX	1	0...FFH	0...255	1	-
Maximum number of regenerations for NOx signal activity diagnosis					
C_NOX_NS_DIAG_ACT_GRD_MIN	1	0...5DCH	0...1.5E+3	1	ppm
Minimum NOx signal gradient for registration of maximum NOx signal value					
C_NOX_NS_DIAG_ACT_MIN	1	0...5DCH	0...1.5E+3	1	ppm
Minimum NOx signal value as diagnosis threshould					
C_NOX_NS_DIAG_GRD_MIN	1	0...5DCH	0...1.5E+3	1	ppm
Minimum NOx signal gradient as diagnosis threshould					
C_TNT_DIAG_ACT_MAX	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Maximum NOx trap temperature for NOx sensor activity diagnosis					
C_TNT_DIAG_ACT_MIN	1	0...FFFFH	0...1.02398E+3	0.015625	°C
Minimum NOx trap temperature for NOx sensor activity diagnosis					
C_T_NS_DIAG_ACT_MAX	1	0...FFFFH	0...6.5535E+3	0.1	s
Time threshold to enable the lean operation possibility without successful NOx signal activity diagnosis					
C_T_RGN_DIAG_ACT_MIN	1	0...FFFFH	0...655.35	0.01	s
Minimum regeneration duration for counting the regeneration within the NOx signal activity diagnosis					
LC_NOX_NS_ACT_INI	1	0...1H	0...1	1	-
Init value for LV_NOX_NS_ACT at initialization of non-volatile memory					
LC_NOX_NS_ACT_RST	1	0...1H	0...1	1	-
Selection of reset strategy for LV_NOX_NS_ACT					
LC_SENS_AFR_DIAG_ACT_OFF	1	0...1H	0...1	1	-
Switch if LV_SENS_AFR will be used for counting of regenerations					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

52.24.1 General information

This diagnosis observes the NOx signal and the NOx signal gradient in order to check that the NOx signal is able to generate a NOx catalyst regeneration request.

Application Condition

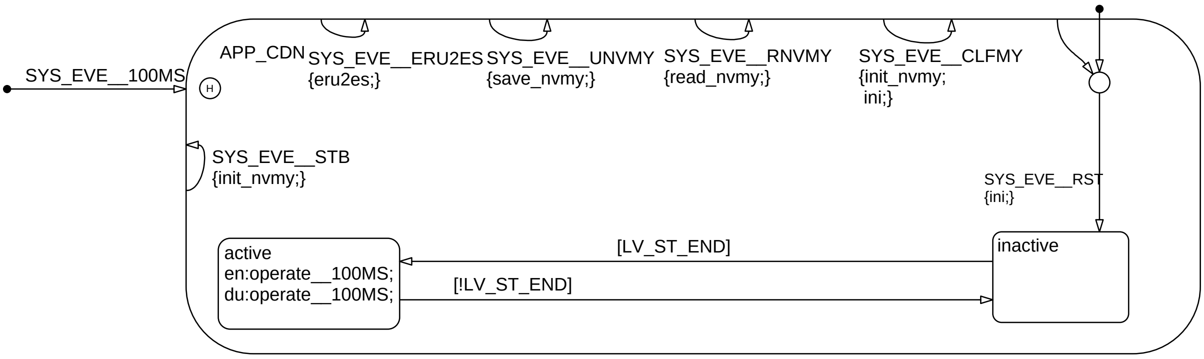
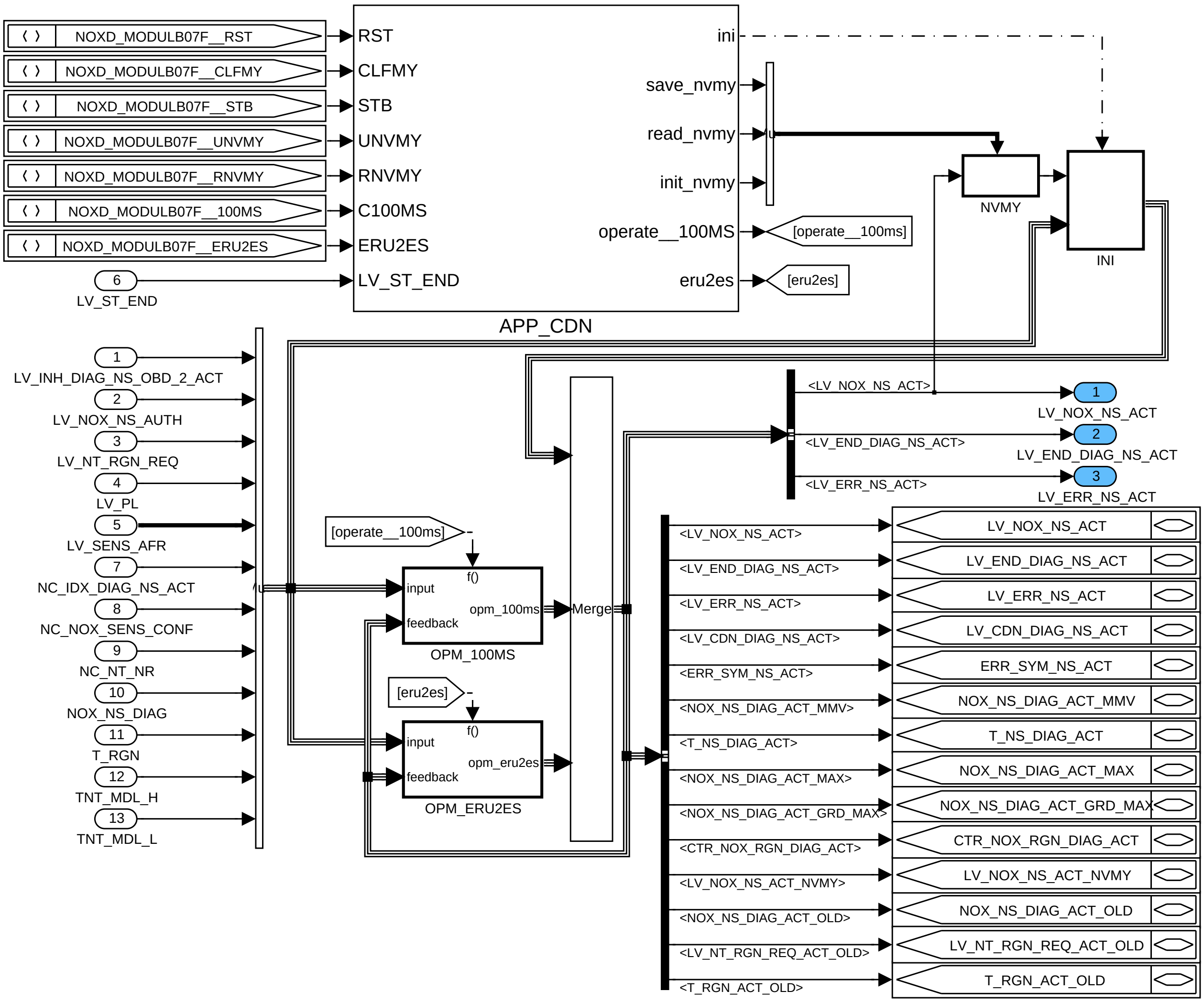


Figure 258 NOXD_MODULB07F/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 26−Sep−2006

Figure 259 NOXD_MODULB07F

52.24.1.1 Management of non volatile memory

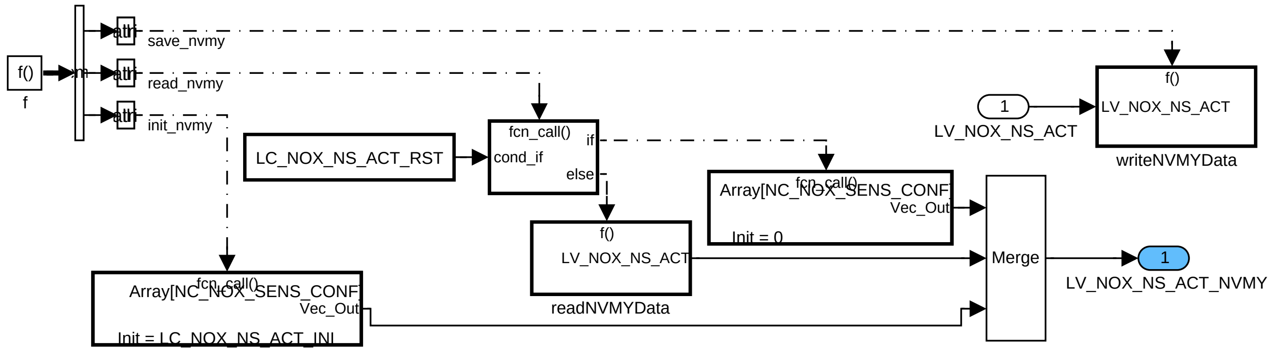


Figure 260 NOXD_MODULB07F/ NVMY

52.24.1.2 Initialization at activation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT< LV_ERR_XX >)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT< LV_END_DIAG_XX >) with XX=NS_ACT

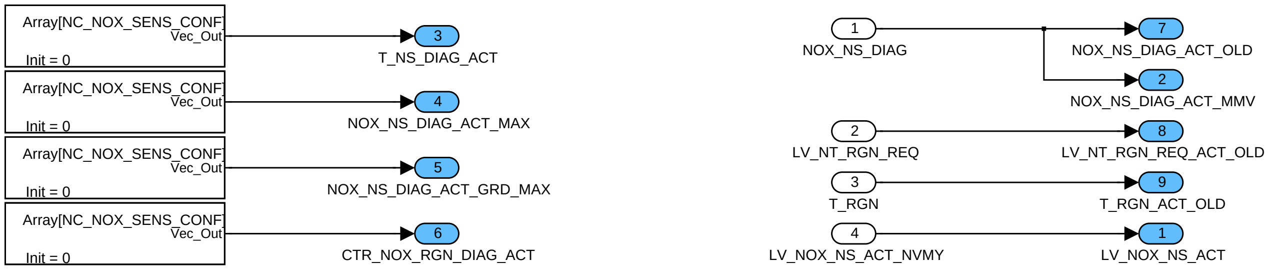


Figure 261 NOXD_MODULB07F/ INI/ CLC

52.24.1.3 Loop control 100 ms

The calculation inside FLP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

The calculation inside NOT_FLP have to be done only once.

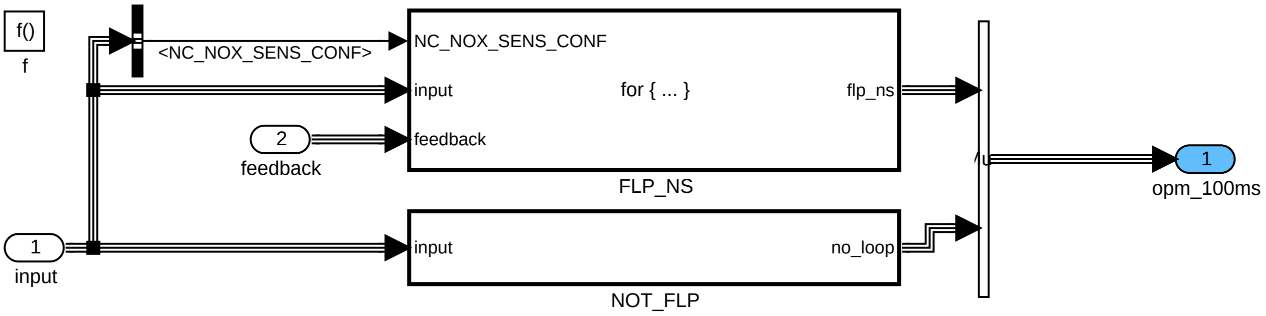


Figure 262 NOXD_MODULB07F/ OPM_100MS

Calculation for each NOx sensor

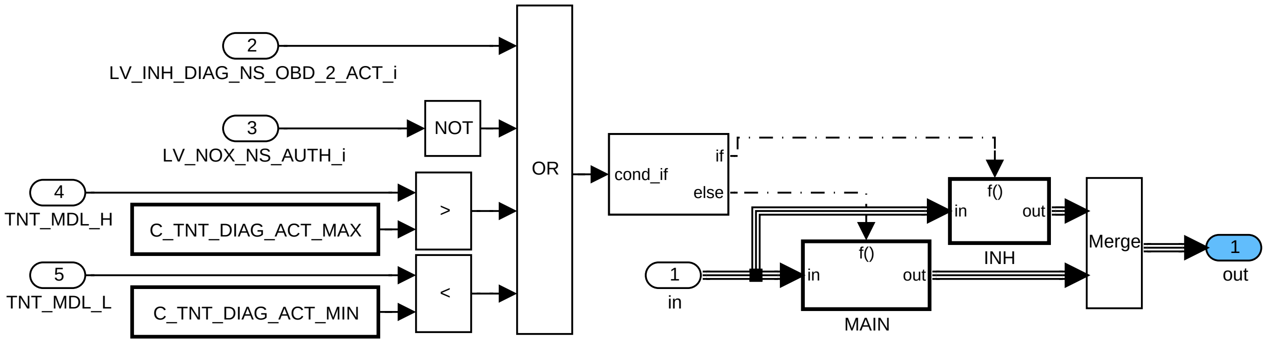


Figure 263 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC

Diagnosis inhibited for one sensor

Figure 264 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ INH/ CLC

Main calculation

Filter of NOx signal for diagnosis

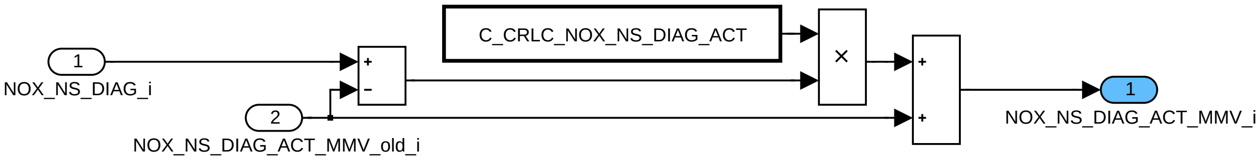


Figure 265 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC1

Maximum of NOx signal gradient

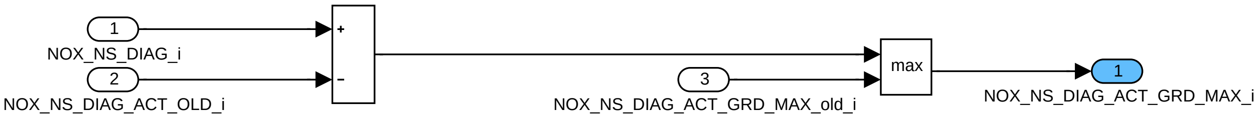


Figure 266 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC2

Maximum of NOx signal

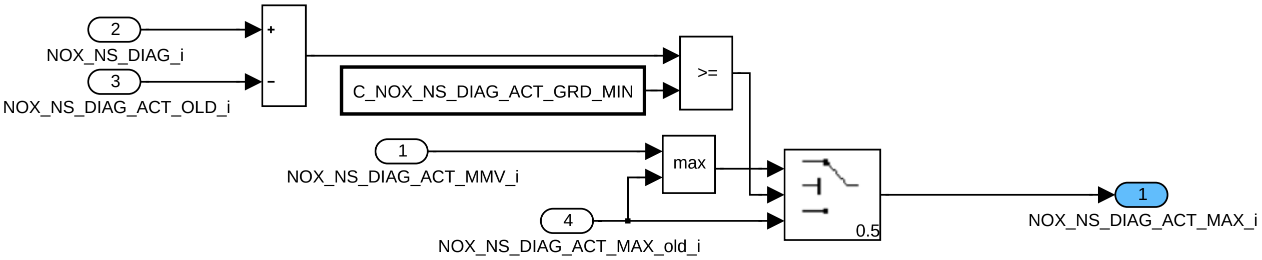


Figure 267 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC3

Diagnosis thresholds

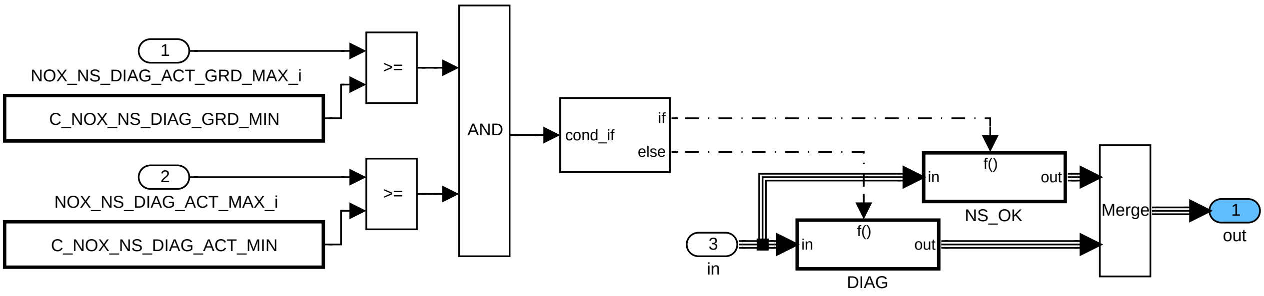


Figure 268 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC4

NOx sensor is OK

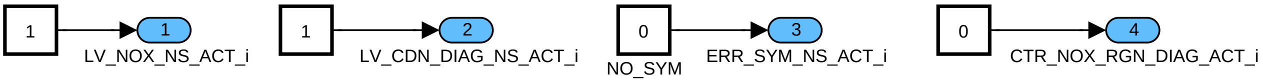


Figure 269 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC4/ NS_OK/ CLC

NOx sensor diagnosis is running

The maximum detection will be divided into two periods. The first period is a calibrated time of part load. The second period is a calibrated number of suitable regenerations. A NOx sensor activity error will be set if during both periods the maxima have not reached the diagnosis threshoulds.

Time of part load is measured

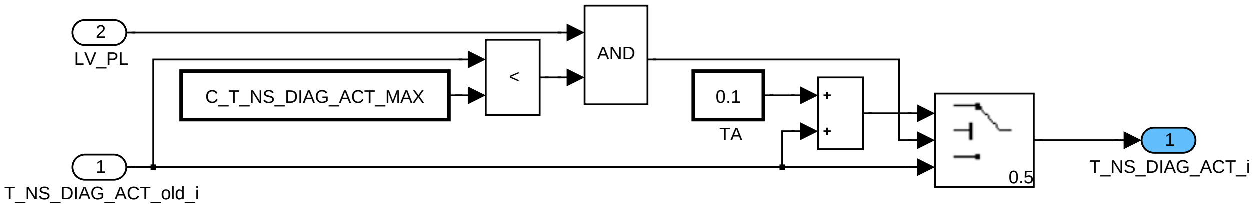


Figure 270 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC4/ DIAG/ CLC1

NOx sensor error decision

After the calibrated number of regenerations the error will be set.

Figure 271 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC4/ DIAG/ CLC2

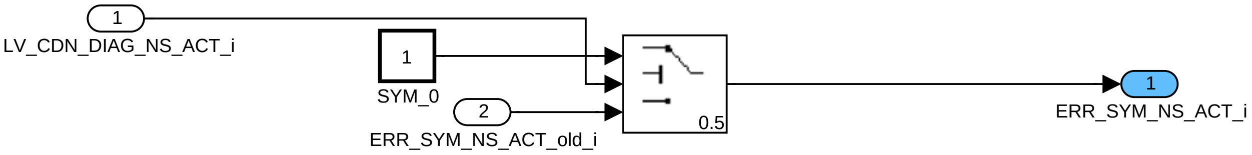


Figure 272 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC4/ DIAG/ CLC3

Control of stratified mode

If an error after the second period will be detected, then the stratified mode will be inhibited.

Otherwise, if the first period is over, then the stratified mode will be allowed, although no activity was detected during the first period. In this period activity of the NOx signal should be monitored due to the stimulation by regeneration.

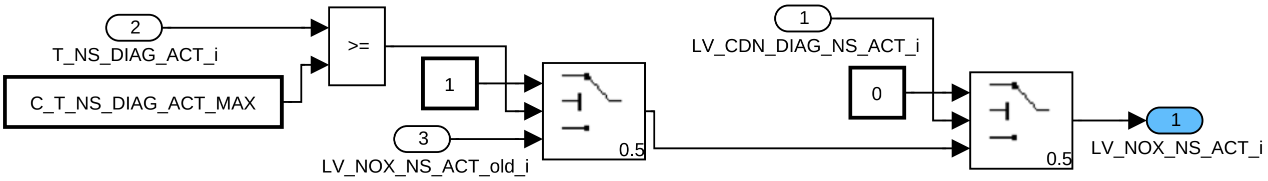


Figure 273 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC4/ DIAG/ CLC4

Regeneration counter

During the second period the number of suitable regenerations will be counted.

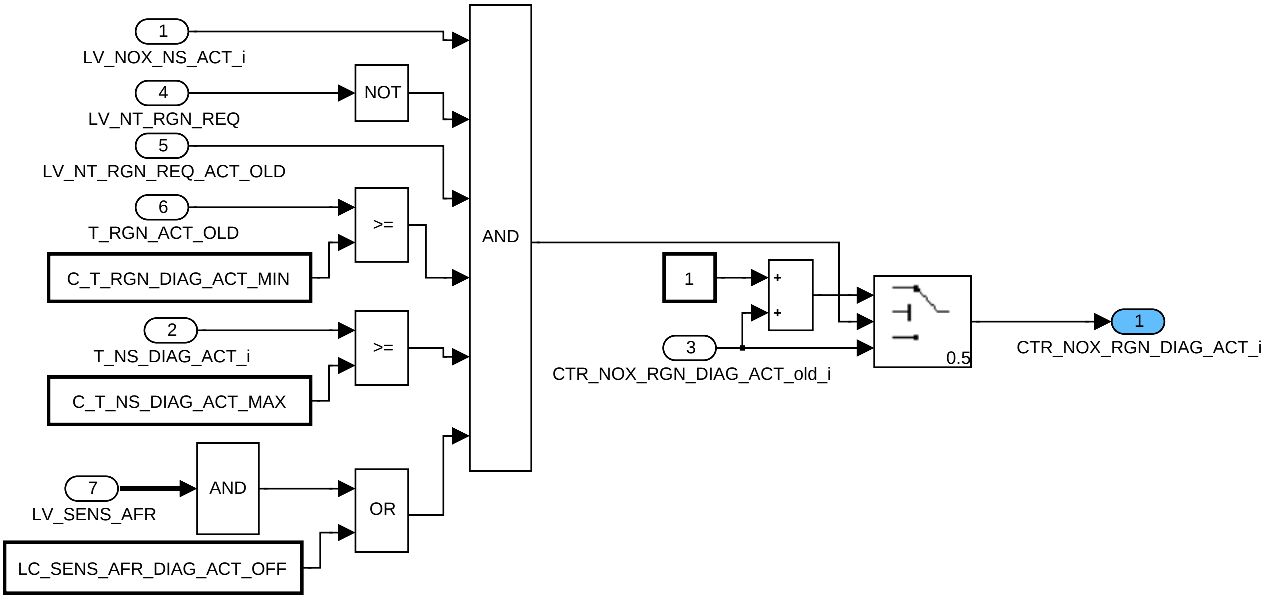


Figure 274 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC4/ DIAG/ CLC5

Error management

LV_ERR_NS_ACT will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NS_ACT will be get by ACTION_ERRM_GetLvEndDiag.

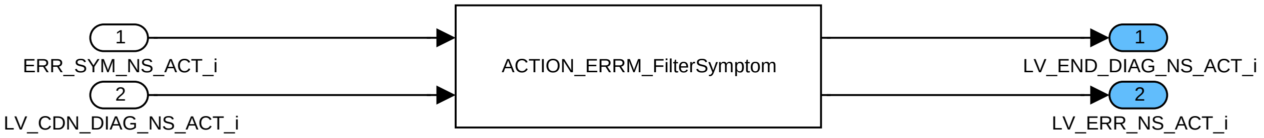


Figure 275 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC5

Old value for gradient calculation

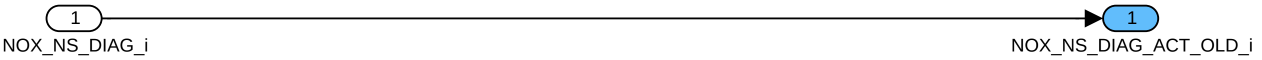


Figure 276 NOXD_MODULB07F/ OPM_100MS/ FLP_NS/ CLC/ MAIN/ CLC6

Bank independent calculation

The old values of the inputs will be stored for edge detection and because the regeneration time before the edge is needed.

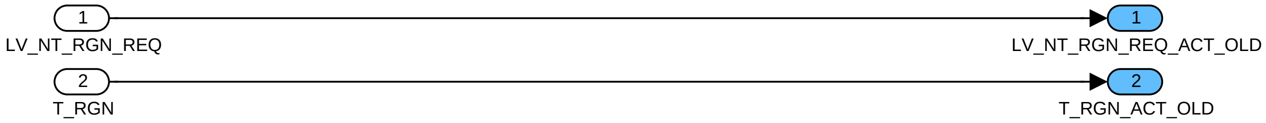


Figure 277 NOXD_MODULB07F/ OPM_100MS/ NOT_FLP/ CLC

52.24.1.4 Loop control at engine run to engine stop

The calculation inside FLP_NS have to be done for each NOx sensor, controlled by configuration data NC_NOX_SENS_CONF.

If NC_NOX_SENS_CONF = 1, then calculation for i = 1.

If NC_NOX_SENS_CONF = 2, then calculation for i = 1 and 2.

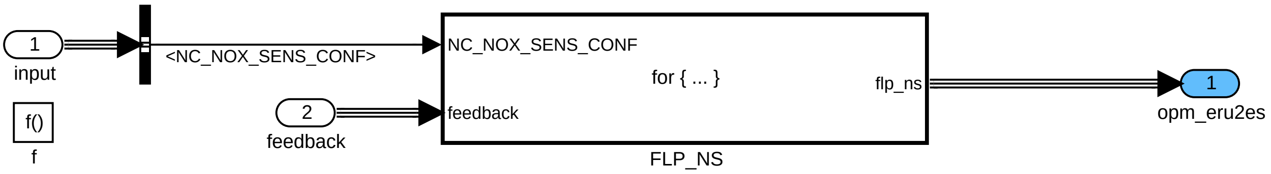


Figure 278 NOXD_MODULB07F/ OPM_ERU2ES

Decision about storage of diagnosis result

The new activity result will be stored in the non-volatile memory if the diagnosis was finished. Otherwise the stored value from the last driving cycle will be kept.

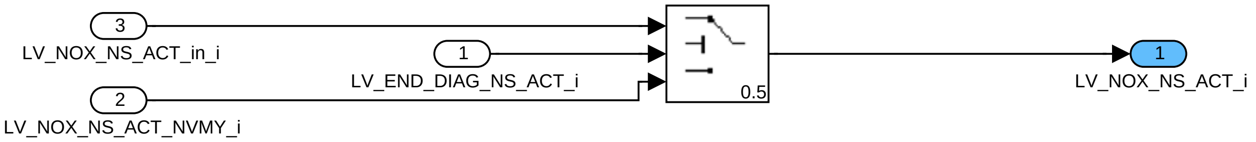


Figure 279 NOXD_MODULB07F/ OPM_ERU2ES/ FLP_NS/ CLC

52.25 Customer adaptation module: AGGR NOXD

52.25.1 Outputs for BMW functions which are not defined as NOXD exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_fi_nsobd1	O/V	0...1H	0...1	1	[-]
Summenfehler Nox Sensor OBD1					
B_fi_nsobd2	O/V	0...1H	0...1	1	[-]
Summenfehler Nox Sensor OBD2					
B_nsadap_anf	O/V	0...1H	0...1	1	[-]
Status Anforderung Noxsensoradaption					
B_nsdiag_anf	O/V	0...1H	0...1	1	[-]
Status Anforderung Noxsensordiagnose					
Baw_nsadap	O/V	0...FFFFH	0...65535	1	[-]
Betriebsarten-Anforderung Noxsensoradaption					
Baw_nsdiag	O/V	0...FFFFH	0...65535	1	[-]
Betriebsarten-Anforderung Noxsensordiagnose					

Input data:

ECU_STATE	LV_ERR_NS_OBD_1[NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_2[NC_NOX_SENS_CONF]	LV_NS_AD_REQ
LV_NS_SHIFT_CMB_INT_REQ	NC_NOX_SENS_CONF	STATE_NS_SHIFT_CMB_REQ	STATE_OPM_REQ_NS_AD

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation at reset and at exit "PWL": 0

Recurrence : 10 ms

Activation: every engine state

Formula section:

```
B_fi_nsobd1       =  LV_ERR_NS_OBD_1[1]
B_fi_nsobd2       =  LV_ERR_NS_OBD_2[1]
If    ECU_STAT = "PWL"
Then         B_nsadap_anf  = 0
             B_nsdiag_anf = 0
             Baw_nsadap = 0
             Baw_nsdiag = 0
Else         B_nsadap_anf = LV_NS_AD_REQ
             B_nsdiag_anf = LV_NS_SHIFT_CMB_INT_REQ
             Baw_nsadap = STATE_OPM_REQ_NS_AD
             Baw_nsdiag = STATE_NS_SHIFT_CMB_REQ
Endif
```

53 NOx management

Chapter NOx management	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 8230 of 9643

Table of Contents

Index

53.1 NOXM Configuration Data

FUNCTION DESCRIPTION:

General information:

The following describes the general rules for determination of the configuration data:

- All configuration data shall be defined by project with information from customer.

Formula section:

Global configuration data

Here are listed the configuration data, which can be used in other aggregates:

Data	Value
NC_NT_NR	2

Local configuration data

Here are listed the configuration data, which are used only in the NOXM aggregate:

Data	Value
NC_NT_CONF	1

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NT_NR	-	1...4H	1...4	1	[-]
Number of NOx catalysts					
NC_NT_CONF	-	0...1H	0...1	1	[-]
configuration data for single/twin branched exhaust line (0: one NOx trap, 1: two NOx traps)					

53.2 AGGR adaptation: NOXM

FUNCTION DESCRIPTION:

General information:

This module handles all variable interfaces of aggregate NOXM. It adapts inputs to aggregate NOXM and outputs from aggregate NOXM on project specific requirements or aggregate structure requirements.

53.2.1 Inputs to aggregate NOXM – 20ms

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CLU_SWI	O/V	0...1H	0...1	1	[-]
Boolean for clutch switch detection					

Input data:

LV_AT	LV_IM_CS_PN	LV_VAR_TCT	

Application conditions:

Initialisation: at reset:

LV_CLU_SWI = 0

Recurrence: 20ms

Activation: after reset

Deactivation: -

Formula section:

LV_CLU_SWI = (LV_IM_CS_PN = 1) **AND** (LV_AT = 0) **AND** (LV_VAR_TCT = 0)

53.2.2 Outputs of aggregate NOXM – 20ms

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_NT_RGN_REQ	V/O	0...1H	0...1	1	[-]
flag inhibiting the request of a NOx catalyst regeneration					
LV_NT_MDL_AFR	V/O	0...1H	0...1	1	[-]
Model rich exhaust gas (stop of regeneration)					
LV_NT_RGN_STOP_SENS	V/O	0...1H	0...1	1	[-]
Flag indicating: Stop of Rgn by sensor enabled					
LV_NT_SENS_AFR[NC_NT_NR]	V/O	0...1H	0...1	1	[-]
downstream sensor measures rich exhaust gas (stop of regeneration)					
LV_NT_TOUT_AFR	V/O	0...1H	0...1	1	[-]
rich exhaust gas after timeout (stop of regeneration)					
LV_NT_RGN_REQ	V/O	0...1H	0...1	1	[-]
Logical value for regeneration phase request					
LV_RGN_REQ_AD	V/O	0...1H	0...1	1	[-]
Request for a full regeneration for adaptation or elongated stratified mode					

Input data:

NC_NT_NR	LV_INH_RGN_REQ	LV_NT_RGN_REQ_AD	LV_MDL_AFR
LV_RGN_STOP_SENS	LV_SENS_AFR[NC_NT_NR]	LV_TOUT_AFR	LV_RGN_NT_REQ

Application conditions:

Initialisation: at reset:

LV_INH_NT_RGN_REQ = 0

LV_RGN_REQ_AD = 0

LV_NT_MDL_AFR = 0

LV_NT_RGN_STOP_SENS = 0

LV_NT_SENS_AFR[i] = 0

LV_NT_TOUT_AFR = 0

LV_NT_RGN_REQ = 0

~~NOX_PPM_RGN_ST[i] = 0~~

Recurrence: 20ms

Activation: after reset

Deactivation: -

Formula section:

LV_INH_NT_RGN_REQ = LV_INH_RGN_REQ

LV_RGN_REQ_AD = LV_NT_RGN_REQ_AD

LV_NT_MDL_AFR = LV_MDL_AFR

LV_NT_RGN_STOP_SENS = LV_RGN_STOP_SENS

LV_NT_SENS_AFR[i] = LV_SENS_AFR[i]

LV_NT_TOUT_AFR = LV_TOUT_AFR

LV_NT_RGN_REQ = LV_RGN_NT_REQ

~~NOX_PPM_RGN_ST[i] = NOX_NS_AD_RGN_ST[i]~~

53.3 Lambda setpoint for catalyst regeneration

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_RGN_CYC_LAMB_RGN	O/V	0... FFH	0... 255	1	[-]
Regenerations counter for calculating of alternate flag					
FAC_NT_AGI_RGN	V	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
NOx catalyst aging factor for Lambda setpoint correction factor determination					
LAMB_RGN [NC_CBK_EX_NR]	O/V	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda setpoint for catalyst regeneration					
LAMB_RGN_FAC [NC_CBK_EX_NR]	V	0... FFH	0... 1.9921875	7.8125e-3	[-]
factor for the correction of the regeneration lambda					
LV_LAMB_PARK_AFL_DEAC	V	0... 1H	0... 1	1	[-]
Deactivation flag for regeneration depending on T_RGN_2					
LV_LAMB_SP_SWI	O/V	0... 1H	0... 1	1	[-]
Flag of alternate					
LV_RGN_NT_REQ_OLD	-	0... 1H	0... 1	1	[-]
Old logical value for regeneration phase request					
LV_SENS_RGN_1_READY_OLD	-	0... 1H	0... 1	1	[-]
Old value for end of first regeneration flag					
MAF_KGH_HOM	O/V	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Calculated MAF_KGH for homogeneous operation					
STATE_RGN [NC_CBK_EX_NR]	O/V	0 1 2 3 4	H_AMPL PARK L_AMPL PASSIVE EQUAL	-	[-]
NOx regeneration state					
T_RGN	O/V	0... FFFFH	0... 655.35	0.01	[s]
Time counter for regeneration active					
T_RGN_2	O/V	0... FFFFH	0... 655.35	0.01	[s]
Time counter for second regeneration					
VLS_NS_OLD [NC_NOX_SENS_CONF]	-	0... 578H	-200... 1200	1	[mV]
Old binary O2 signal voltage, raw value, measured by NOx-Sensor					

Input Data:

NC_CBK_EX_NR	LV_ST	LV_ES	MAF_SP_TQI
N_32	MAF	LV_NT_ACT	LV_RGN_NT_REQ
STATE_NOX	LV_SENS_RGN_1_READY	VLS_NS [NC_NOX_SENS_CONF]	LV_SENS_AFR [NC_NT_NR]
LC_SENS_AFR_MOD	NC_NOX_SENS_CONF	NC_NT_NR	LV_VLS_GRD_NEG
TNT_MDL_MV_SNG [NC_NT_NR]	LV_VLS_NS_VLD [NC_NOX_SENS_CONF]	NT_AGI	NT_AGI_THERMO
NT_AGI_SUL	FAC_NT_AGI_LIM		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CTR_RGN_SWI_ACT	1	1... FFH	1... 255	1	[-]
Number of regenerations with initialization of STATE_RGN when the alternate flag is 1					
C_CTR_RGN_SWI_DEAC	1	1... FFH	1... 255	1	[-]
Number of regenerations with initialization of STATE_RGN when the alternate flag is 0					

C_FAC_NT_AGI_RGN_MAN	1	0... FFH	0... 1.9921875	7.8125e-3	[-]
Manual set value for FAC_NT_AGI_RGN					
C_LAMB_FAC_L_AMPL_NOT_GRD	1	0... FFH	0... 1.9921875	7.8125e-3	[-]
factor for the correction of the regeneration after waiting time without negative gradient of VLS_NS					
C_LAMB_PARK_BAS	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Basis lambda setpoint for park					
C_LAMB_PAS_BAS	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Basis lambda setpoint for passive state					
C_LAMB_RGN_MAX_EQU	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Maximum Lambda value for equal regeneration mode					
C_LAMB_RGN_MAX_H_AMPL	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Maximum value for high amplitude regeneration lambda					
C_LAMB_RGN_MAX_L_AMPL	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Maximum value for low amplitude regeneration lambda					
C_STATE_FAC_NT_AGI_RGN	1	0 1 2 3 4	ALL THERMO SUL MAN LIM	-	[-]
Mode switch for FAC_NT_AGI_RGN					
C_STATE_LAMB_RGN_1_H_AMPL	1	0 1 2 3	NONE VLS T_RGN T_RGN_2	-	[-]
Lambda setpoint correction strategy for first high amplitude regeneration					
C_STATE_LAMB_RGN_2_H_AMPL	1	0 1 2 3	NONE VLS T_RGN T_RGN_2	-	[-]
Lambda setpoint correction strategy for second high amplitude regeneration					
C_STATE_LAMB_RGN_EQU	1	0 1 2 3	NONE VLS T_RGN T_RGN_2	-	[-]
Lambda setpoint correction strategy for equal mode regeneration					
C_STATE_LAMB_RGN_L_AMPL	1	0 1 2 3	NONE VLS_MAF T_RGN T_RGN_2	-	[-]
Strategy for low amplitude regeneration					
C_STATE_RGN_1_PARK	1	0 1 2 3	NONE VLS T_RGN T_RGN_2	-	[-]
Park strategy for first regeneration					
C_STATE_RGN_2_PARK	1	0 1 2 3	NONE VLS T_RGN T_RGN_2	-	[-]
Park strategy for second regeneration					
C_STATE_RGN_INI_1	1	0 1 2 3 4	H_AMPL PARK L_AMPL PASSIVE EQUAL	-	[-]
Initialisation state for first bank					

C_STATE_RGN_INI_2	1	0 1 2 3 4	H_AMPL PARK L_AMPL PASSIVE EQUAL	-	[-]
Initialisation state for second bank					
C_STATE_RGN_INI_VLS_NOT_VLD_1	1	0 1 2 3 4	H_AMPL PARK L_AMPL PASSIVE EQUAL	-	[-]
Initialisation state for first bank as long as NOx sensor VLS signal is not valid					
C_STATE_RGN_INI_VLS_NOT_VLD_2	1	0 1 2 3 4	H_AMPL PARK L_AMPL PASSIVE EQUAL	-	[-]
Initialisation state for second bank as long as NOx sensor VLS signal is not valid					
C_T_PURGE_H_AMPL	1	0... FFFFH	0... 655.35	0.01	[s]
Purge time for high amplitude regeneration					
C_T_PURGE_L_AMPL	1	0... FFFFH	0... 655.35	0.01	[s]
Purge time for low amplitude regeneration					
C_T_RGN_2_NOT_GRD	1	0... FFFFH	0... 655.35	0.01	[s]
After this time the second bank will be regenerated without waiting for negative gradient of VLS_NS					
IP_LAMB_FAC_EQU_NT_AGI	6	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_FAC_NT_AGI_RGN_IP_EQU	6	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Lambda setpoint correction factor for equal regeneration mode depending on NOx catalyst aging factor					
IP_LAMB_FAC_EQU_T_RGN	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_T_RGN_EQU_IP_LAMB_FAC	8	0... FFFFH	0... 655.35	0.01	[s]
Lambda setpoint correction factor for equal mode regeneration depending on T_RGN					
IP_LAMB_FAC_H_AMPL_1_VLS	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_VLS_H_AMPL_1_IP_LAMB_FAC	8	0... 578H	-200... 1200	1	[mV]
Lambda setpoint correction factor for first high amplitude regeneration depending on binary NS lambda signal					
IP_LAMB_FAC_H_AMPL_2_VLS	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_VLS_H_AMPL_2_IP_LAMB_FAC	8	0... 578H	-200... 1200	1	[mV]
Lambda setpoint correction factor for second high amplitude regeneration depending on binary NS lambda signal					
IP_LAMB_FAC_H_AMPL_NT_AGI	6	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_FAC_NT_AGI_RGN_IP_H_AMPL	6	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Lambda setpoint correction factor for high amplitude regeneration mode depending on NOx catalyst aging factor					
IP_LAMB_FAC_L_AMPL_2_VLS	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_VLS_L_AMPL_2_IP_LAMB_FAC	8	0... 578H	-200... 1200	1	[mV]
Lambda setpoint correction factor for second low amplitude regeneration depending on binary NS lambda signal					
IP_LAMB_FAC_L_AMPL_NT_AGI	6	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_FAC_NT_AGI_RGN_IP_L_AMPL	6	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Lambda setpoint correction factor for low amplitude regeneration mode depending on NOx catalyst aging factor					
IP_LAMB_FAC_L_AMPL_T_RGN_1	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_T_RGN_1_L_AMPL_IP_LAMB_FAC	8	0... FFFFH	0... 655.35	0.01	[s]
Lambda setpoint correction factor for low amplitude regeneration depending on T_RGN					
IP_LAMB_FAC_L_AMPL_T_RGN_2	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_T_RGN_2_L_AMPL_IP_LAMB_FAC	8	0... FFFFH	0... 655.35	0.01	[s]
Lambda setpoint correction factor for low amplitude regeneration depending on T_RGN_2					
IP_LAMB_FAC_L_AMPL_VLS_MAF_KGH	4*8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_MAF_KGH_L_AMPL_IP_LAMB_FAC	4	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
LDP_VLS_L_AMPL_1_IP_LAMB_FAC	8	0... 578H	-200... 1200	1	[mV]
Lambda setpoint correction factor for first low amplitude regeneration depending on MAF and VLS_NS					
IP_LAMB_FAC_PARK_1_T_RGN	8	0... FFH	0... 1.9921875	7.8125e-3	[-]

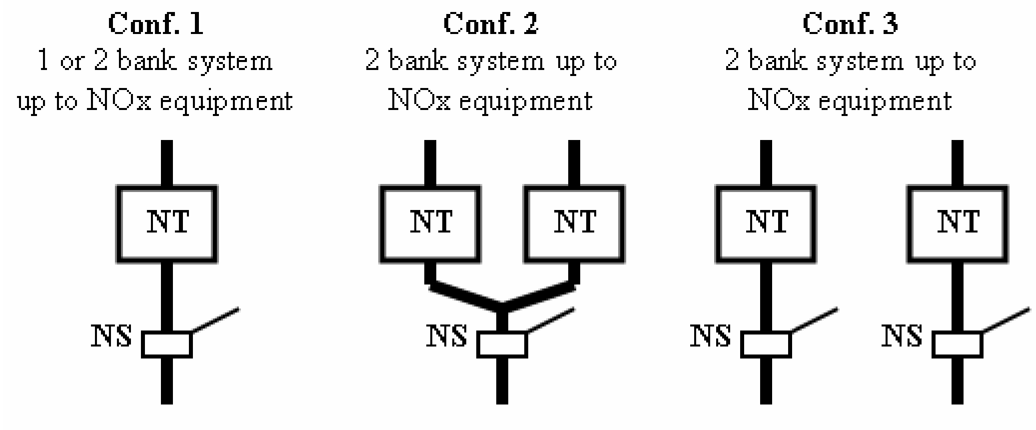
LDP_T_RGN_1_PARK_IP_LAMB_FAC	8	0... FFFFH	0... 655.35	0.01	[s]
Lambda setpoint correction factor for park for first regeneration depending on T_RGN					
IP_LAMB_FAC_PARK_1_VLS	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_VLS_PARK_1_IP_LAMB_FAC	8	0... 578H	-200... 1200	1	[mV]
Lambda setpoint correction factor for park depending on binary NS lambda signal					
IP_LAMB_FAC_PARK_2_T_RGN_2	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
LDP_T_RGN_2_PARK_IP_LAMB_FAC	8	0... FFFFH	0... 655.35	0.01	[s]
Lambda setpoint correction factor for park depending on T_RGN_2					
IP_LAMB_PURGE_H_AMPL	8*6	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_TNT_MDL_MV_SNG_1_NOXM	6	0... FFFFH	0... 1023.984375	0.015625	[°C]
LDPM_MAF_KGH_HOM_1_NOXM	8	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Basis lambda purge for high amplitude regeneration					
IP_LAMB_PURGE_L_AMPL	8*6	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_TNT_MDL_MV_SNG_1_NOXM	6	0... FFFFH	0... 1023.984375	0.015625	[°C]
LDPM_MAF_KGH_HOM_1_NOXM	8	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Basis lambda purge for low amplitude regeneration					
IP_LAMB_RGN_EQU_BAS	8*6	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_TNT_MDL_MV_SNG_1_NOXM	6	0... FFFFH	0... 1023.984375	0.015625	[°C]
LDPM_MAF_KGH_HOM_1_NOXM	8	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Basis lambda for equal mode regeneration					
IP_LAMB_RGN_H_AMPL_BAS	8*6	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_TNT_MDL_MV_SNG_1_NOXM	6	0... FFFFH	0... 1023.984375	0.015625	[°C]
LDPM_MAF_KGH_HOM_1_NOXM	8	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Basis lambda for high amplitude regeneration					
IP_LAMB_RGN_L_AMPL_BAS	8*6	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDPM_TNT_MDL_MV_SNG_1_NOXM	6	0... FFFFH	0... 1023.984375	0.015625	[°C]
LDPM_MAF_KGH_HOM_1_NOXM	8	0... FFFFH	0... 2047.96875	0.03125	[kg/h]
Basis lambda for low amplitude regeneration					
IP_LAMB_RGN_MAX_PARK	8*8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDP_N_32_IP_LAMB_RGN_MAX_PARK	8	0... FFH	0... 8160	32	[rpm]
LDP_MAF_IP_LAMB_RGN_MAX_PARK	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
Maximum lambda during park phase of regeneration					

Configuration Data:

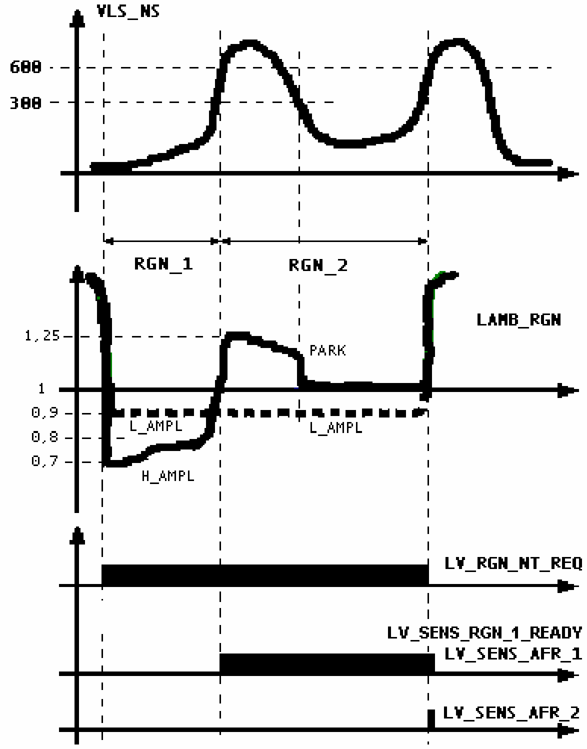
Name	Mode	Hex.Limits	Phys.Limits	Resol.	Unit
NC_MAF_FAC_CYL	1	0... FFFFH	0... 11111.1111111	0.1695447	[-]
conversion factor between MAF					

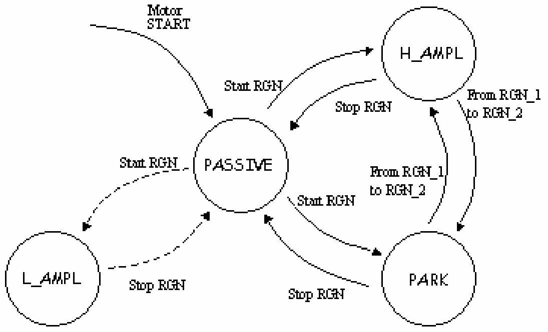
General Information

The requirement of NOxstorage catalyst regeneration is the engine is operated with a rich mixture. This is the lambda setpoint for catalyst regeneration edition for single- and twin-branched exhaust lines, also for the Y configuration with two NOx-Traps (NT) and one NOx-Sensor (Conf 2). In this case the lambda setpoint of the already regenerated NT is set to the park value C_LAMB_PARK_BAS. The covered exhaust gas configurations are shown in the picture below:



The regeneration curves, state flow and correction options for example configuration shows the picture:





Options of correction factor calculation for RGN_1:

	PURGE	VLS	T_RGN	VLS/MAF	NONE
L_AMPL	✓	–	✓	✓	✓
H_AMPL	✓	✓	–	–	✓
PARK	–	✓	✓	–	✓

Options of correction factor calculation for RGN_2:

	PURGE	VLS	T_RGN_2	VLS/MAF	NONE
PARK	–	–	✓	–	✓
L_AMPL	–	✓	✓	–	✓
H_AMPL	–	✓	–	–	✓

NC_CBK_EX_NR is the number of exhaust gas lines. Therefore
i = 1, for cylinder bank 1
i = 2, for cylinder bank 2.
NC_NOX_SENS_CONF is the number of NOx-Sensors. Therefore
NC_NOX_SENS_CONF = 0, for no NOx-Sensor
NC_NOX_SENS_CONF = 1 (n = 1), for one NOx-Sensor
NC_NOX_SENS_CONF = 2 (n = i), for two NOx-Sensors
NC_NT_NR is the number of NOx Catalysts. Therefore
NC_NT_NR = 1 (k = 1), for one NOx Catalyst
NC_NT_NR = 2 (k = i), for two NOx Catalysts

Application Conditions

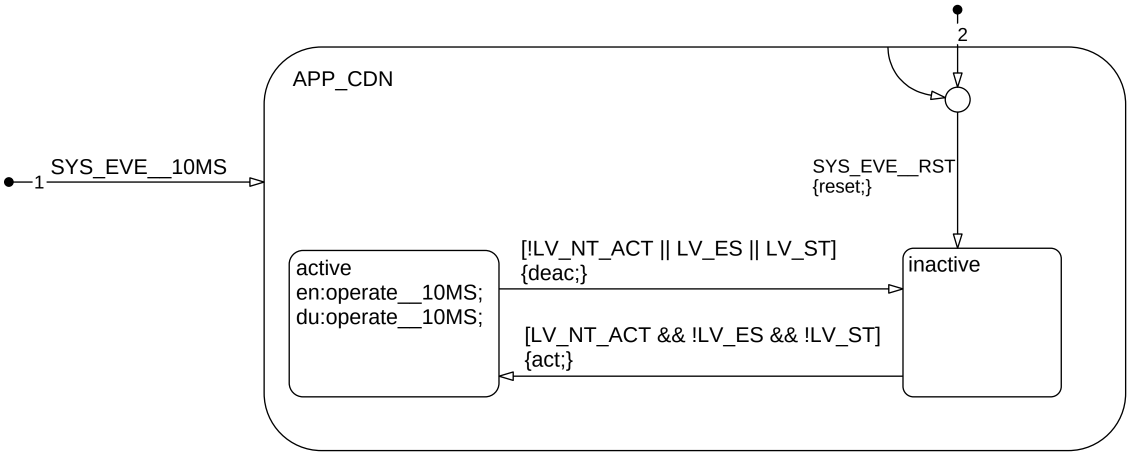


Figure 1:

Function description

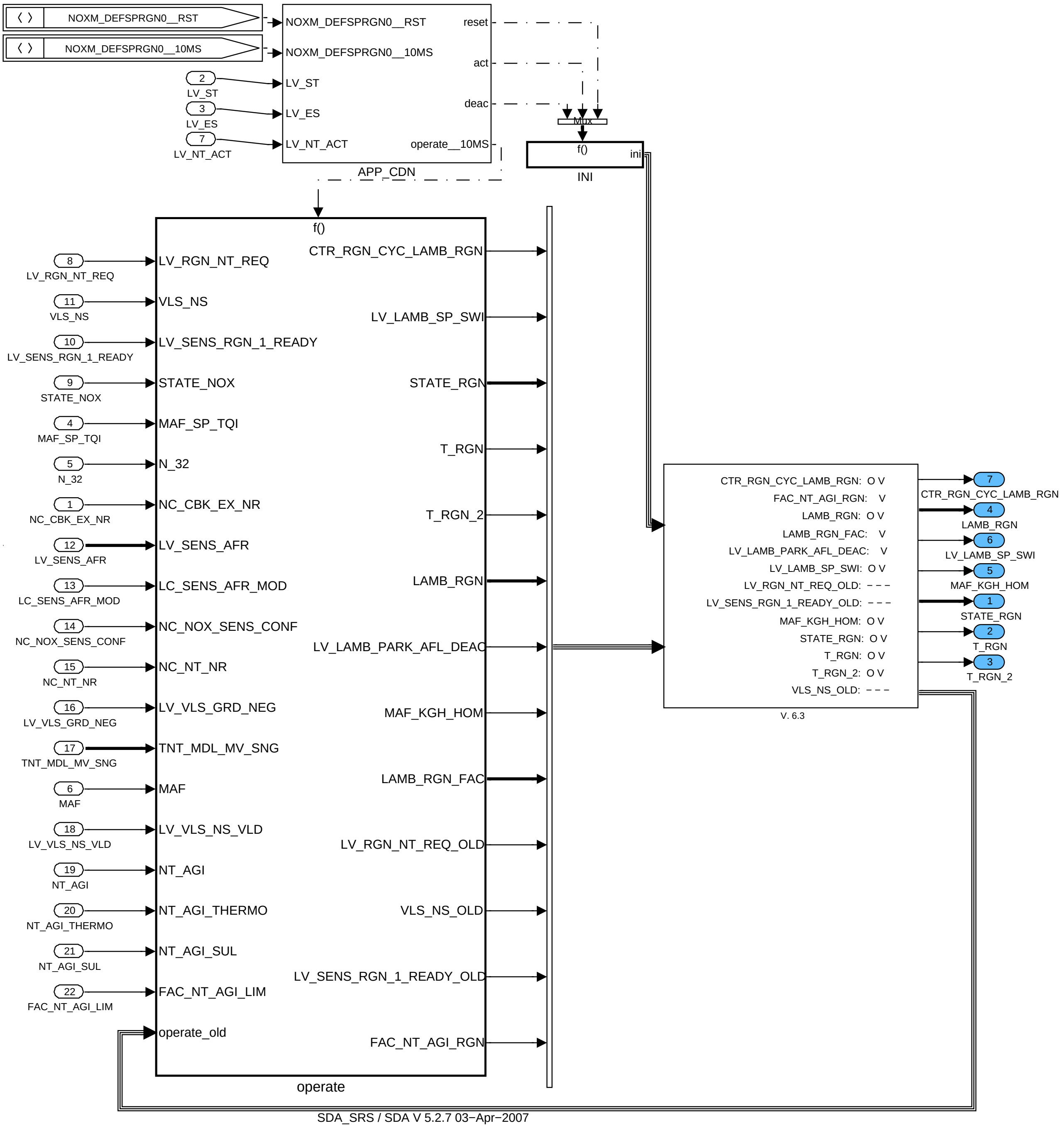


Figure 2:

53.3.1 INITIALIZATION AT RESET, ACTIVATION AND DEACTIVATION

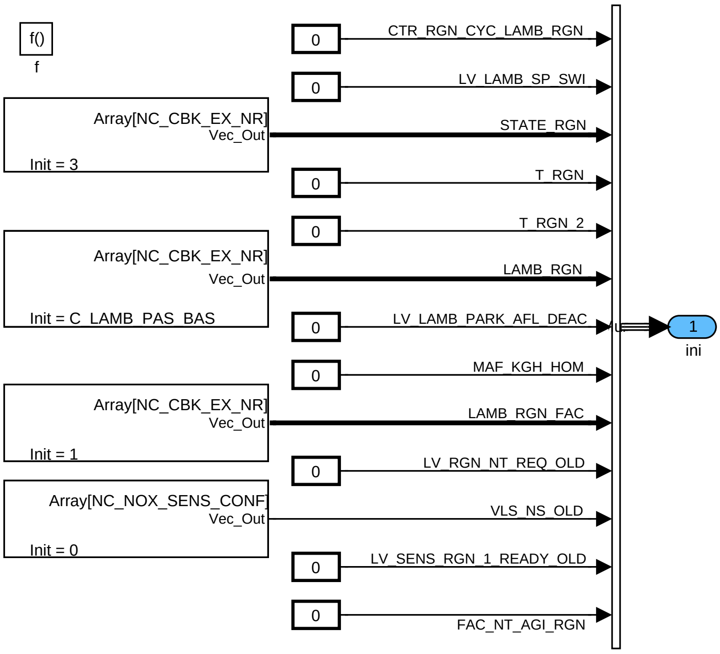


Figure 3:

53.3.2 FORMULA SECTION

The function consists of two main sections:
SECTION 1: Initialization of the Regeneration State at start of the regeneration, calculation of the regeneration time and of the lambda basis for the low amplitude regeneration
SECTION 2: Calculation of the Regeneration State and the Regeneration Lambda Setpoint

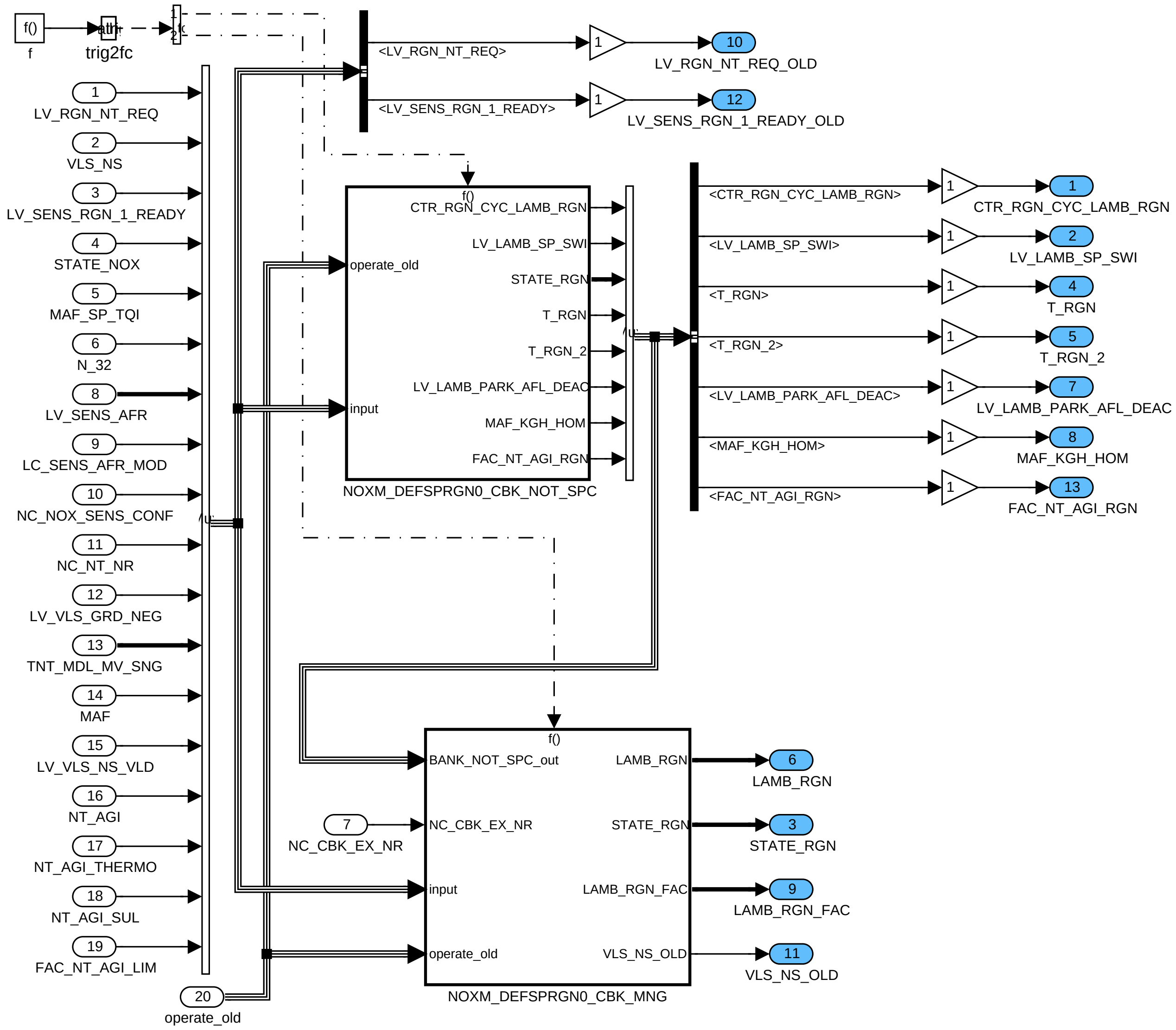

f()
f
trig2fc
<LV_RGN_NT_REQ>
<LV_SENS_RGN_1_READY>
1
10
LV_RGN_NT_REQ_OLD
12
LV_SENS_RGN_1_READY_OLD
1
LV_RGN_NT_REQ
2
VLS_NS
3
LV_SENS_RGN_1_READY
4
STATE_NOX
5
MAF_SP_TQI
6
N_32
8
LV_SENS_AFR
9
LC_SENS_AFR_MOD
10
NC_NOX_SENS_CONF
11
NC_NT_NR
12
LV_VLS_GRD_NEG
13
TNT_MDL_MV_SNG
14
MAF
15
LV_VLS_NS_VLD
16
NT_AGI
17
NT_AGI_THERMO
18
NT_AGI_SUL
19
FAC_NT_AGI_LIM
20
operate_old
f()
operate_old
input
CTR_RGN_CYC_LAMB_RGN
LV_LAMB_SP_SWI
STATE_RGN
T_RGN
T_RGN_2
LV_LAMB_PARK_AFL_DEAC
MAF_KGH_HOM
FAC_NT_AGI_RGN
NOXM_DEFSPRGN0_CBK_NOT_SPC
<CTR_RGN_CYC_LAMB_RGN>
<LV_LAMB_SP_SWI>
<T_RGN>
<T_RGN_2>
<LV_LAMB_PARK_AFL_DEAC>
<MAF_KGH_HOM>
<FAC_NT_AGI_RGN>
1
CTR_RGN_CYC_LAMB_RGN
2
LV_LAMB_SP_SWI
4
T_RGN
5
T_RGN_2
7
LV_LAMB_PARK_AFL_DEAC
8
MAF_KGH_HOM
13
FAC_NT_AGI_RGN
f()
BANK_NOT_SPC_out
NC_CBK_EX_NR
input
operate_old
LAMB_RGN
STATE_RGN
LAMB_RGN_FAC
VLS_NS_OLD
NOXM_DEFSPRGN0_CBK_MNG
7
NC_CBK_EX_NR
6
LAMB_RGN
3
STATE_RGN
9
LAMB_RGN_FAC
11
VLS_NS_OLD

Figure 4:

53.3.2.1 SECTION 1: Initialization of Regeneration State at start of the regeneration, calculation of regeneration time and lambda basis for the low amplitude regeneration

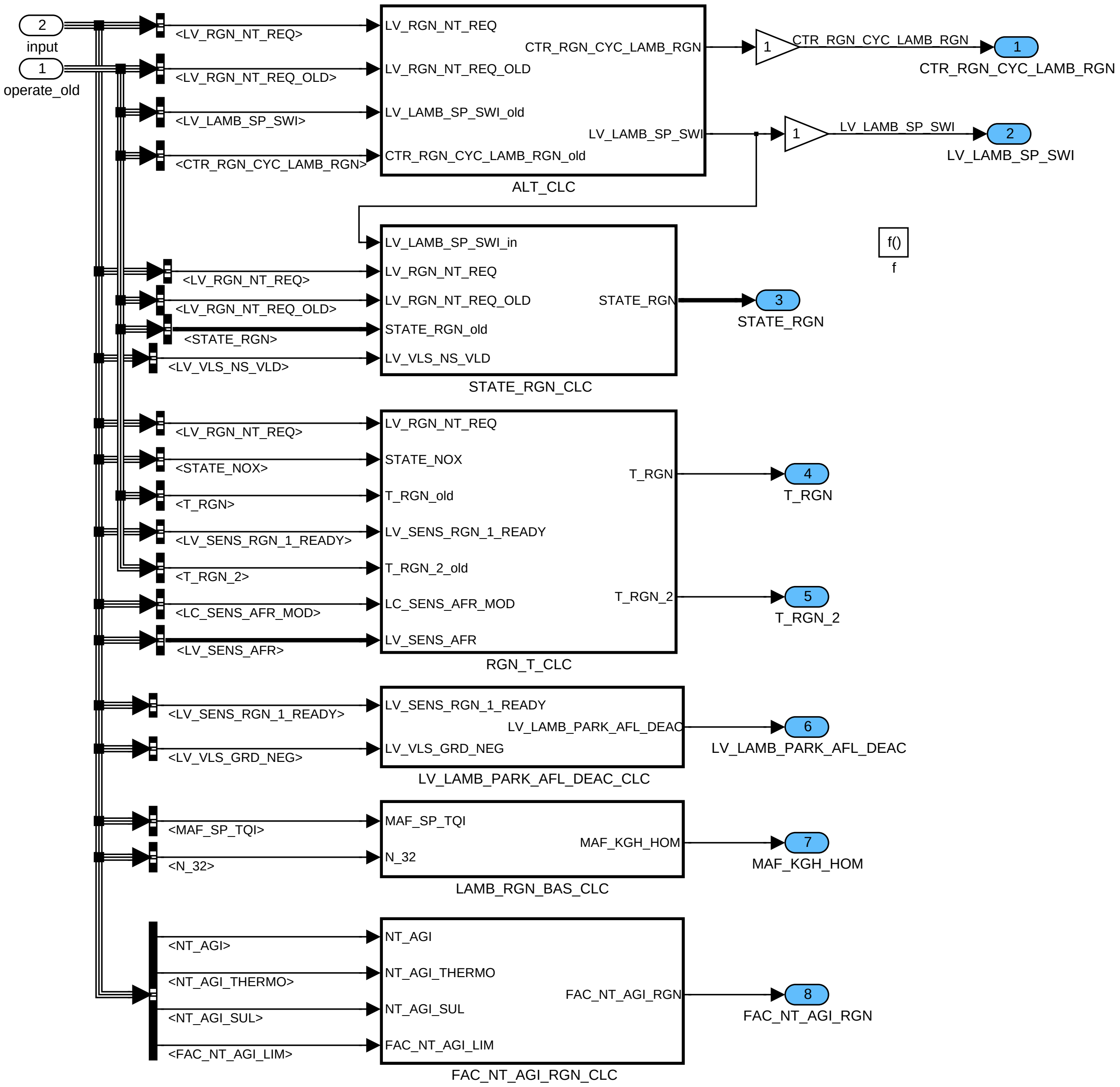


Figure 5:

53.3.2.1.1 Calculation of regeneration counter and alternate flag

The regeneration counter is incremented at entering in regeneration. The alternate flag is used for switching of initialization values for the STATE_RGN between both banks. The interval between switching is set by the C_CTR_RGN_SWI_DEAC (alternate flag is not active) and by the C_CTR_RGN_SWI_ACT (alternate flag is active).

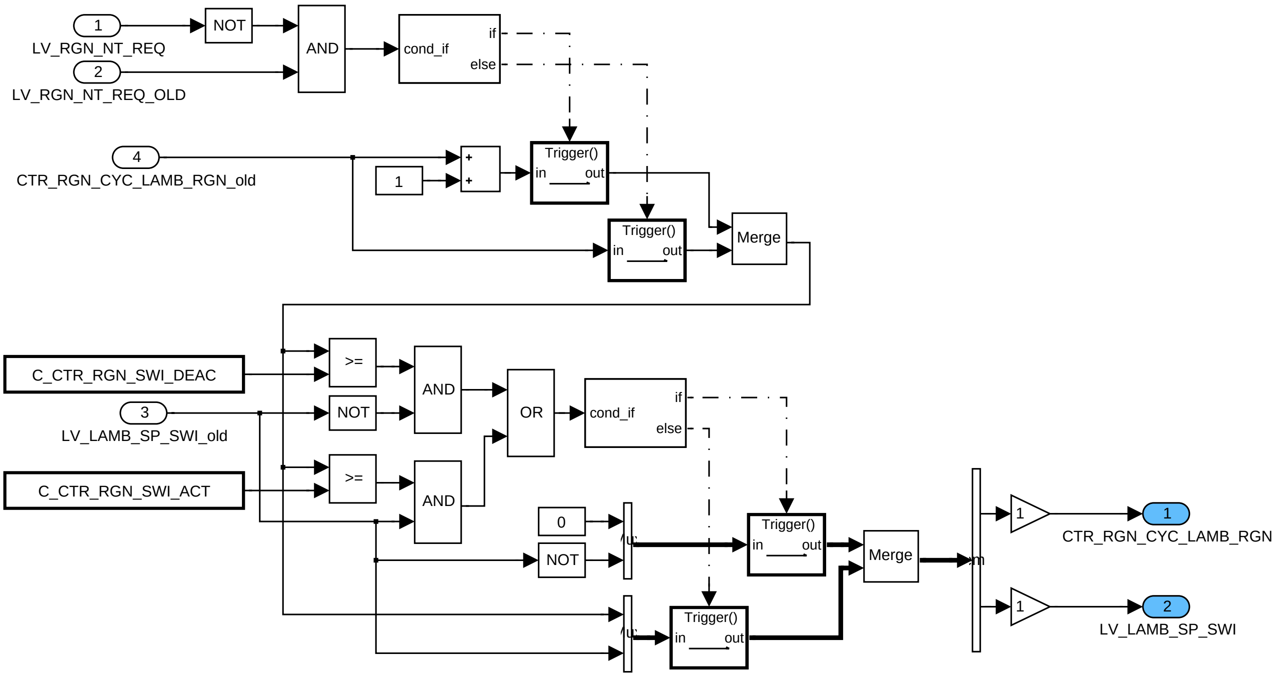


Figure 6:

53.3.2.1.2 Initialization of regeneration state

The regeneration strategy for all banks is set at start of regeneration. The regeneration state is initialized depending on alternate flag.

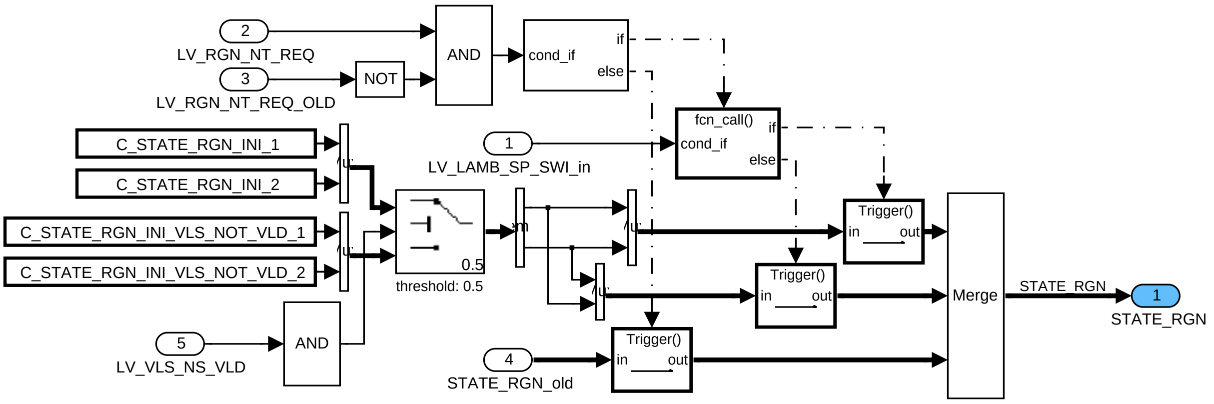


Figure 7:

53.3.2.1.3 Calculation of time for the first and the second regeneration

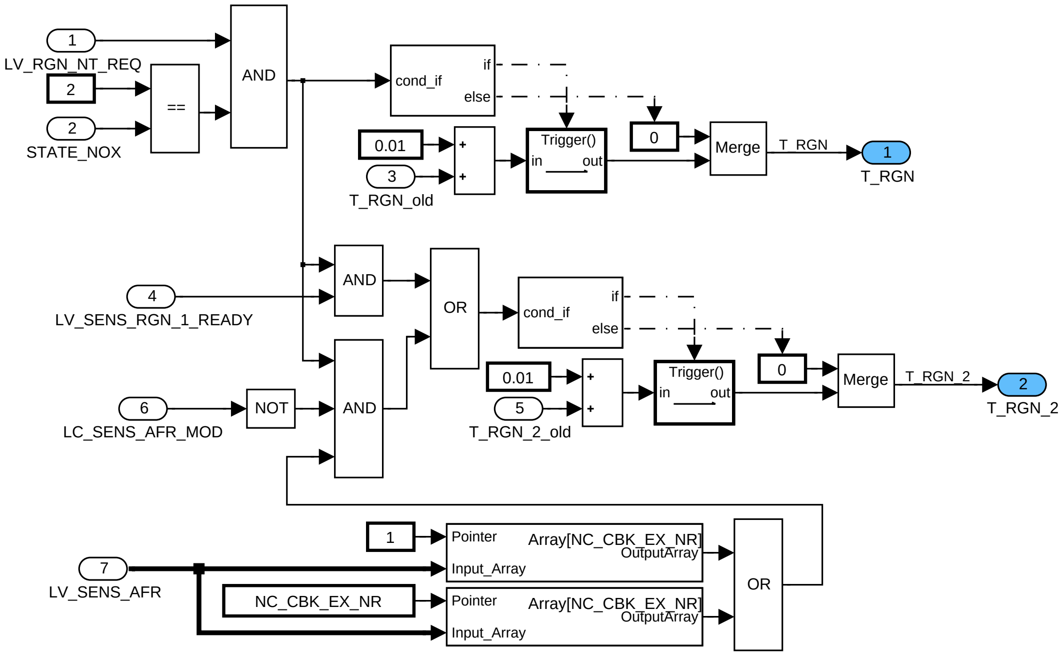


Figure 8:

53.3.2.1.4 Deactivation of lean lambda setpoint after switch to the state "PARK".

To ensure the binary signal of the NOx Sensor can fall down after first regeneration, the deactivation of the lean lambda setpoint take place only when defined negative gradient has been reached.

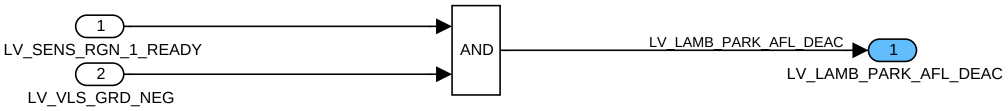


Figure 9:

53.3.2.1.5 Calculation of MAF_KGH for homogeneous operation

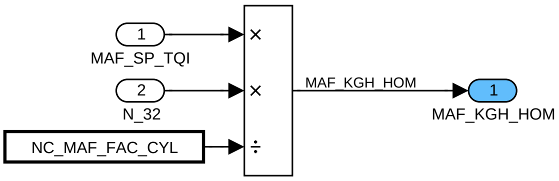


Figure 10:

53.3.2.1.6 Calculation of NOx catalyst aging factor for correction of basic Lambda setpoint for NOx regeneration

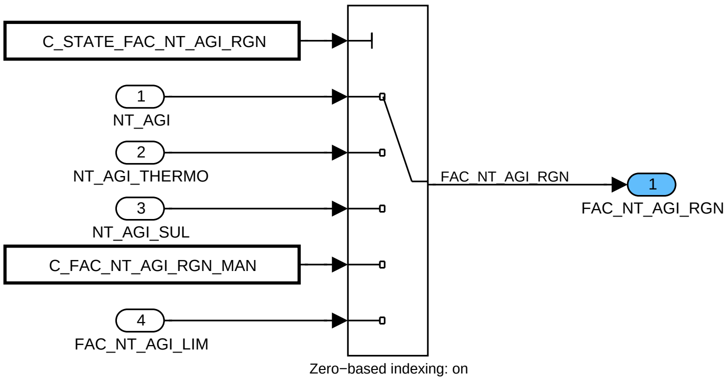


Figure 11:

53.3.2.2 FOR Loop

53.3.2.2.1 FOR Loop structure

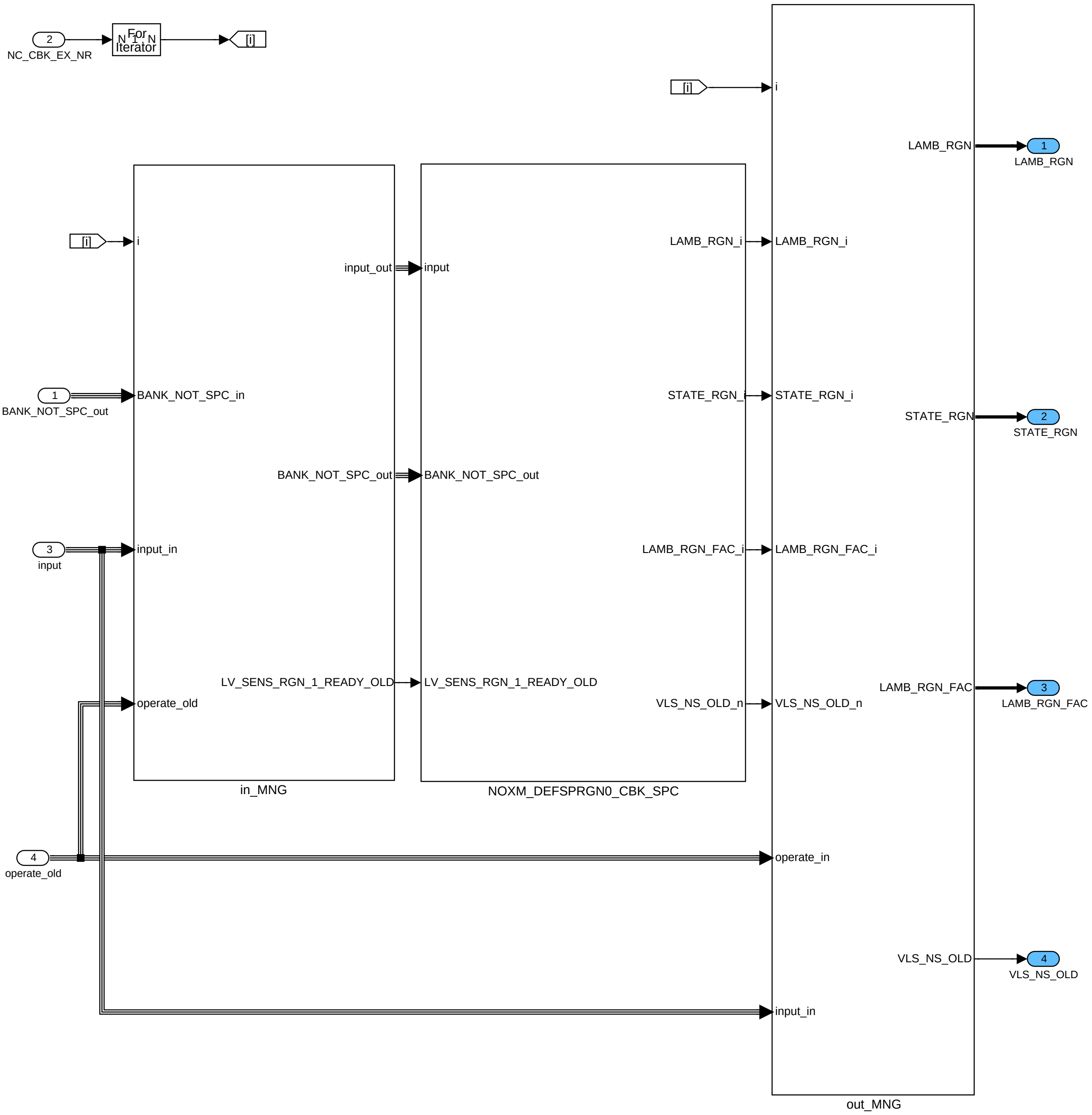


Figure 12:

53.3.2.2.1.1 Signal selection out of signal vectors

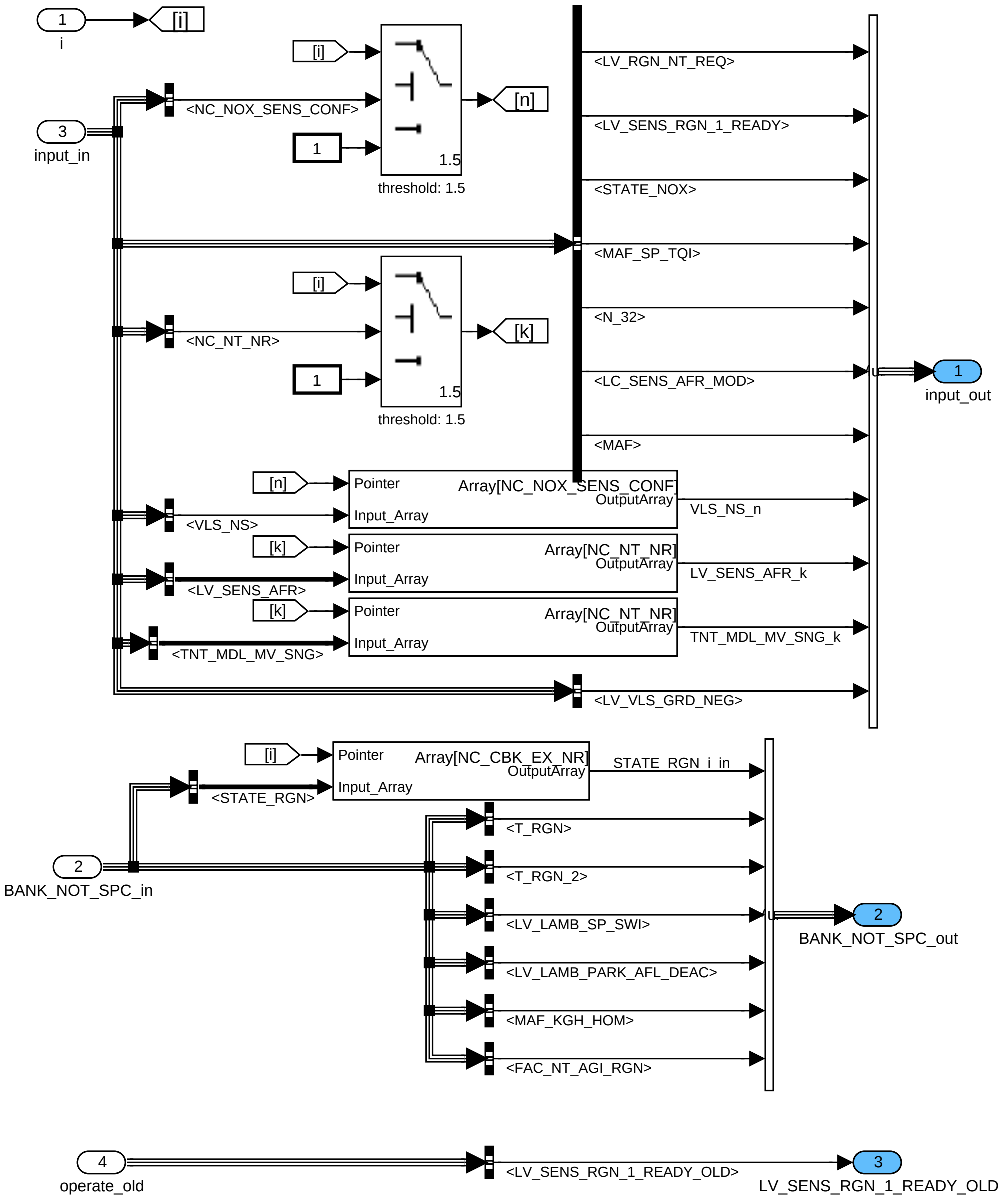


Figure 13:

53.3.2.2.1.2 SECTION 2: Calculation of the Regeneration State and the Regeneration Lambda Setpoint

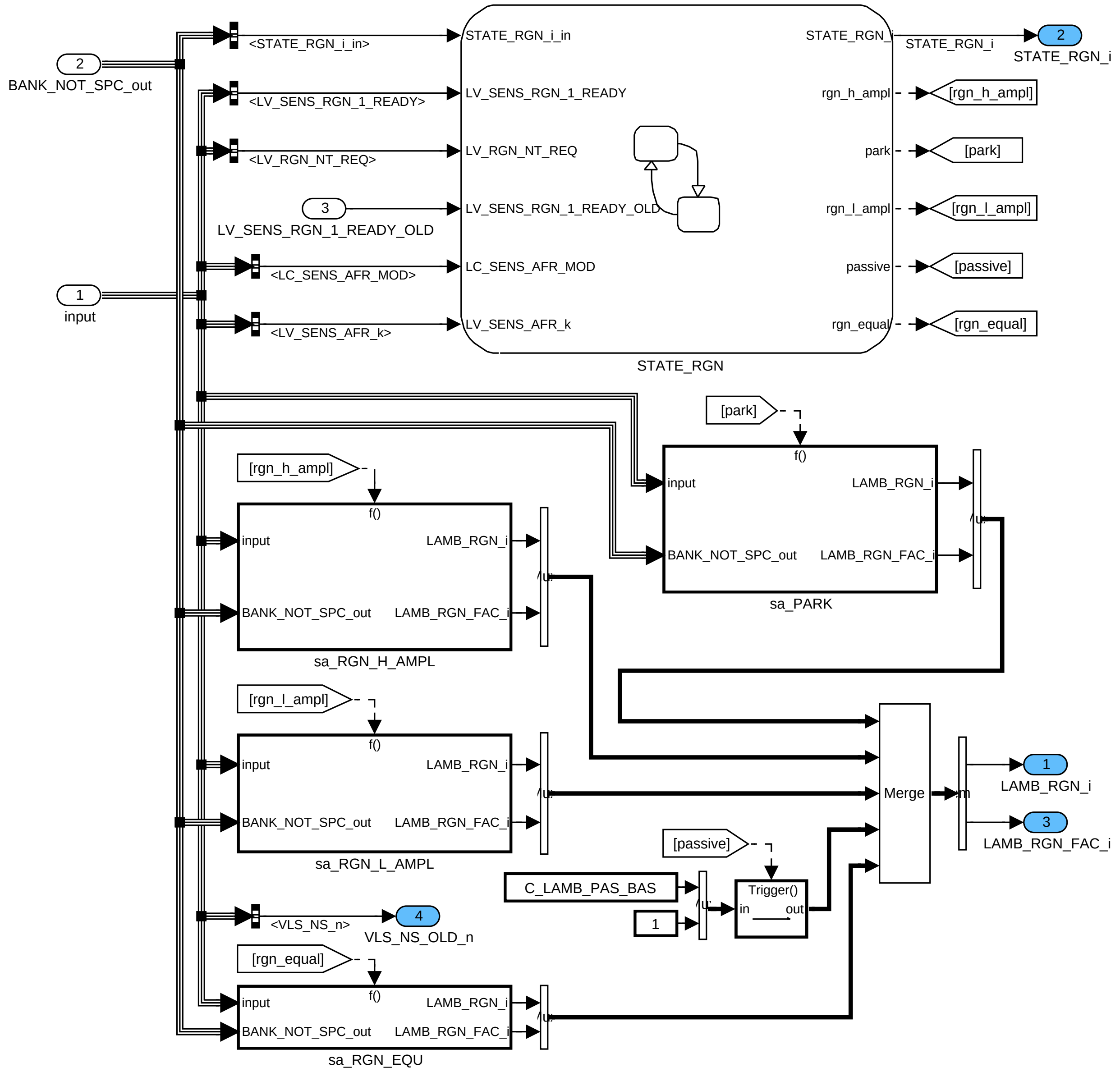


Figure 14:

53.3.2.2.1.2.1 STATE_RGN

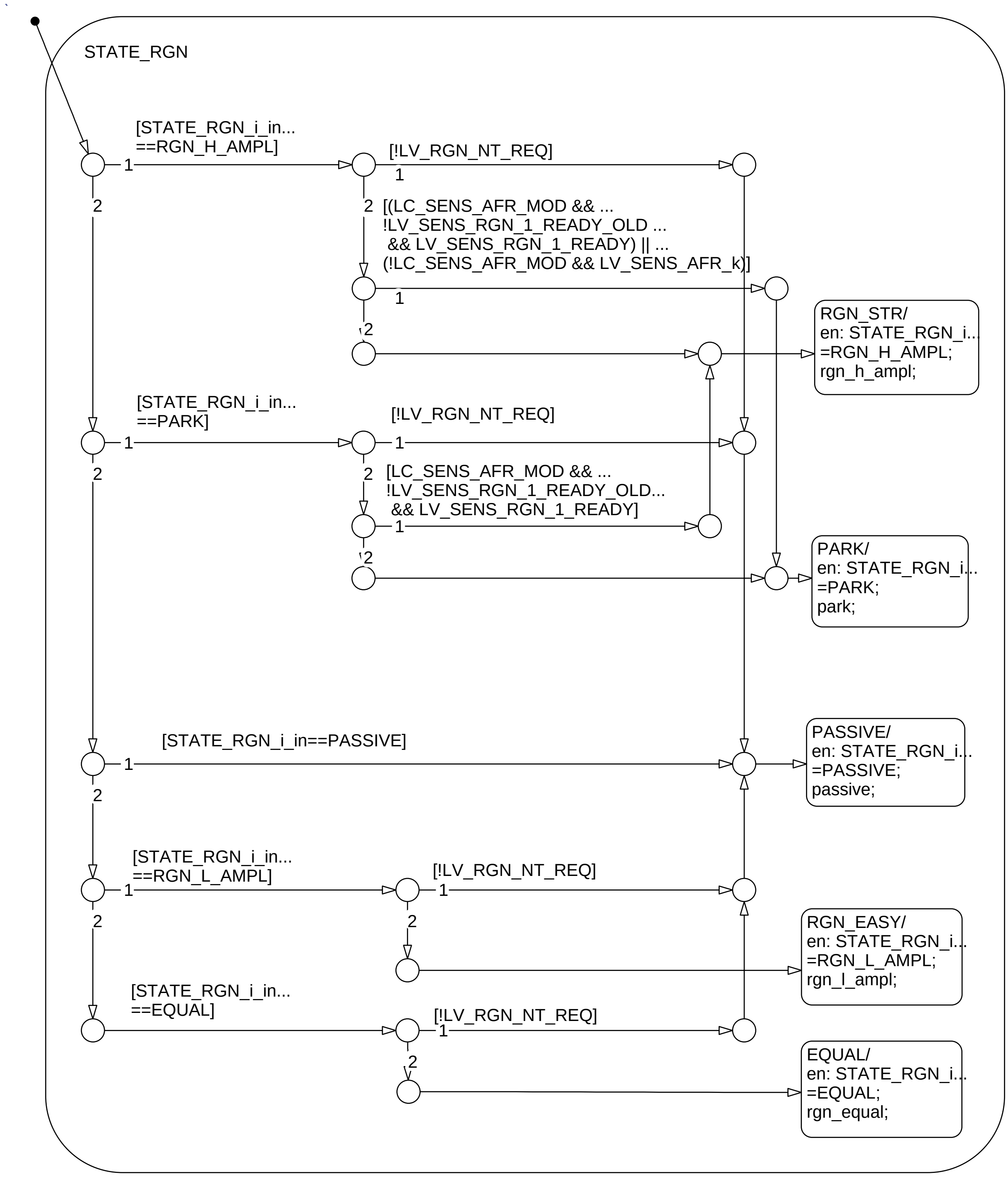


Figure 15:

53.3.2.2.1.2.2 Regeneration with High Amplitude

The high amplitude is set for bank which should be regenerated faster. This state is also entered when the bank was initialized with the state "PARK" and the first regeneration is finished. The lambda value is limited to the maximal value to avoid a jump to lean condition.

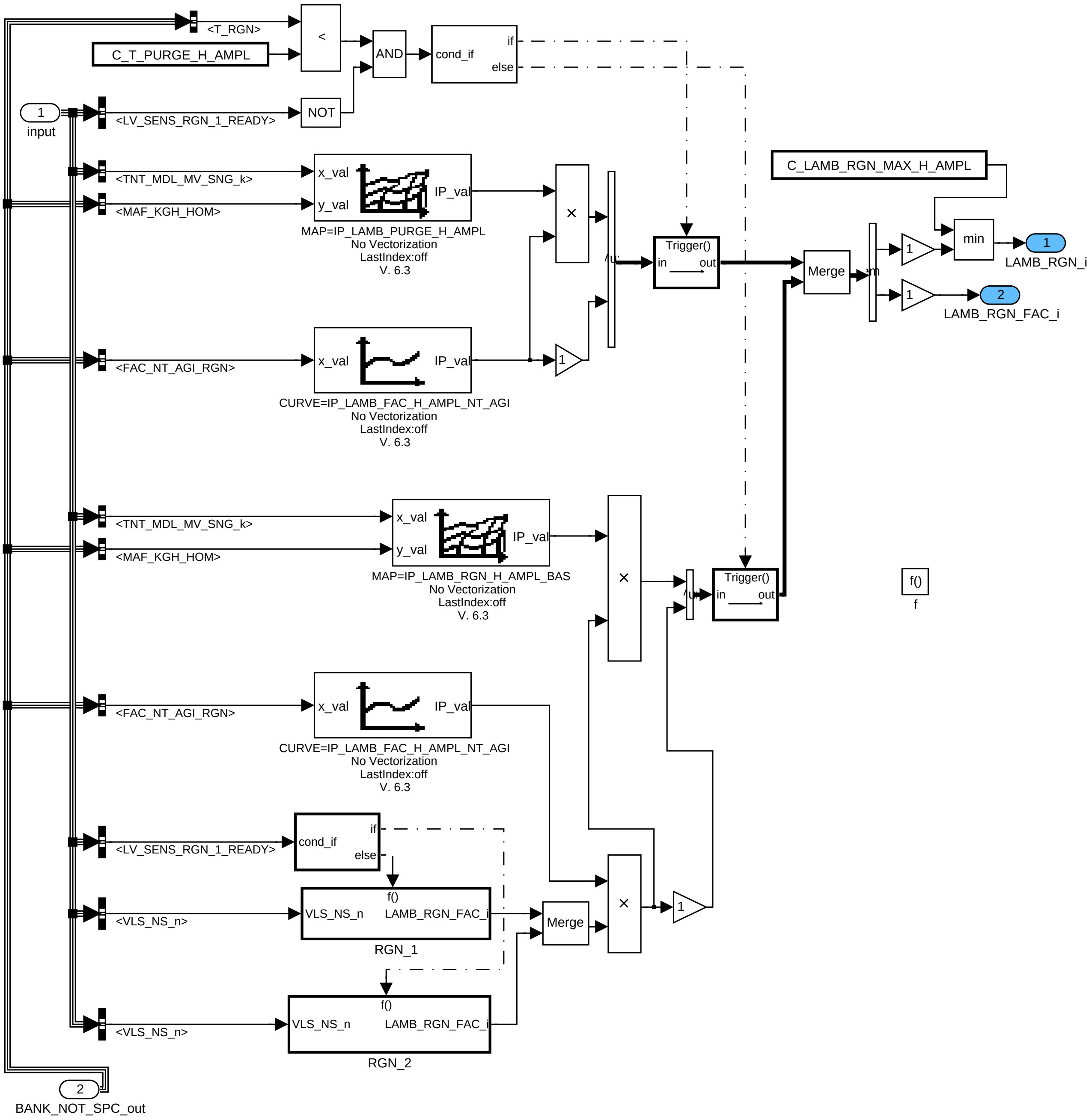


Figure 16:

53.3.2.2.1.2.2.1 Regeneration with High Amplitude - calculation of the Correction Factor at the first regeneration

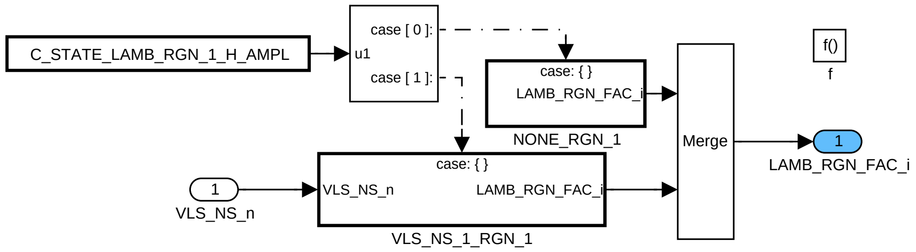


Figure 17:

53.3.2.2.1.2.2.1.1 Regeneration with High Amplitude - 1st regeneration - no Lambda correction

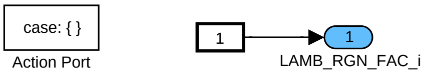


Figure 18:

53.3.2.2.1.2.2.1.2 Regeneration with High Amplitude - 1st regeneration - Lambda correction by binary O2 signal of NOx sensor

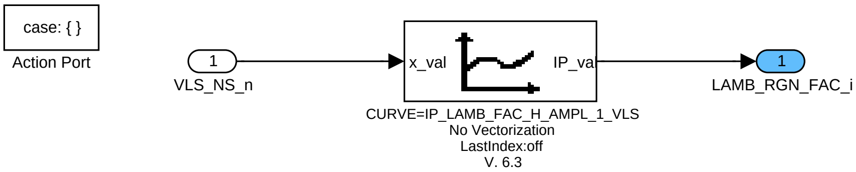


Figure 19:

53.3.2.2.1.2.2.2 Regeneration with High Amplitude - calculation of the Correction Factor at the second regeneration

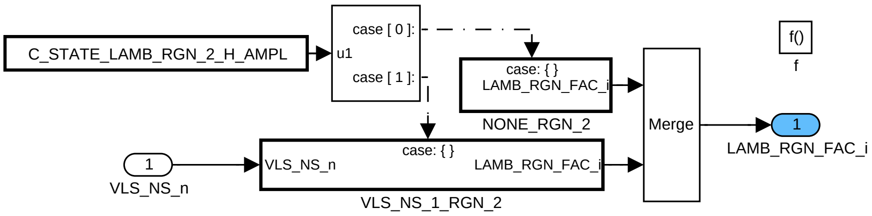


Figure 20:

53.3.2.2.1.2.2.2.1 Regeneration with High Amplitude - 2nd regeneration - no Lambda correction

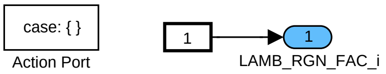


Figure 21:

53.3.2.2.1.2.2.2.2 Regeneration with High Amplitude - 2nd regeneration - Lambda correction by binary O2 signal of NOx sensor

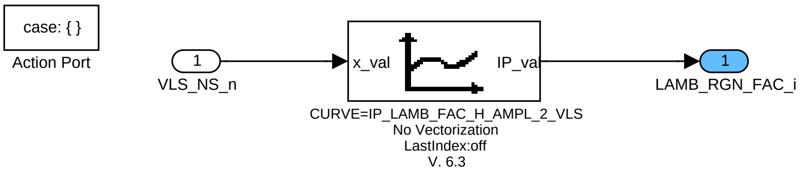


Figure 22:

53.3.2.2.1.2.3 Park phase for lambda setpoint

When the bank regenerated with high amplitude is finished, the lambda value is set to neutral or to lean combustion condition. The bank can be also at entering in regeneration initialized with the state "PARK".

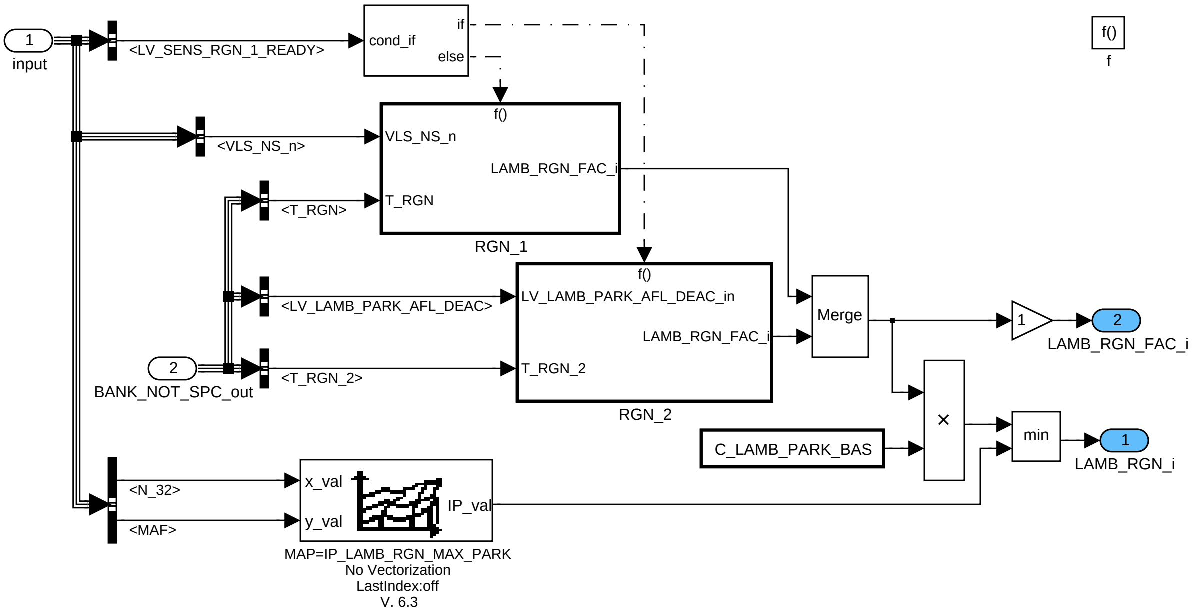


Figure 23:

53.3.2.2.1.2.3.1 Park phase for lambda setpoint - calculation of the Correction Factor at the first part of regeneration

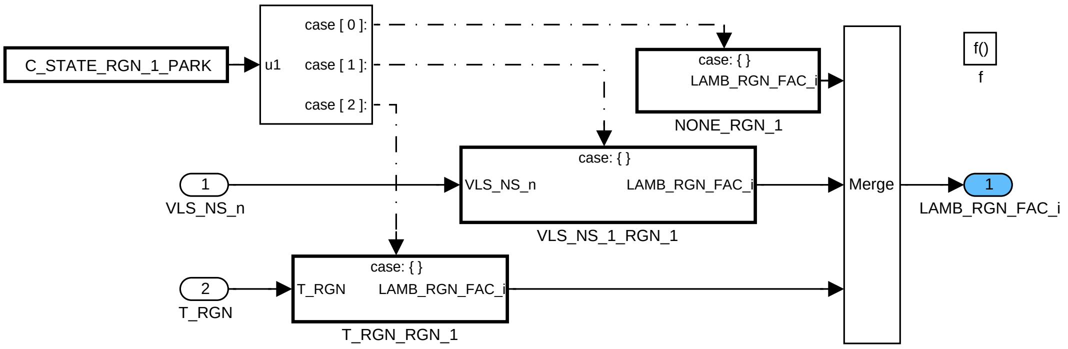


Figure 24:

53.3.2.2.1.2.3.1.1 Park phase - 1st regeneration - no Lambda correction

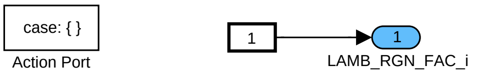


Figure 25:

53.3.2.2.1.2.3.1.2 Park phase - 1st regeneration - Lambda correction by binary O2 signal of NOx sensor

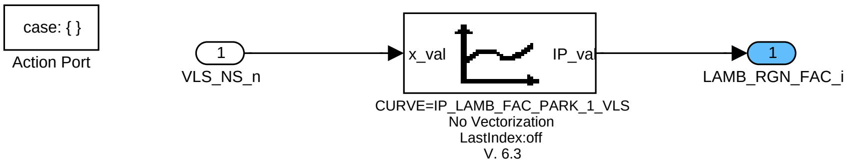


Figure 26:

53.3.2.2.1.2.3.1.3 Park phase - 1st regeneration - Lambda correction by regeneration time

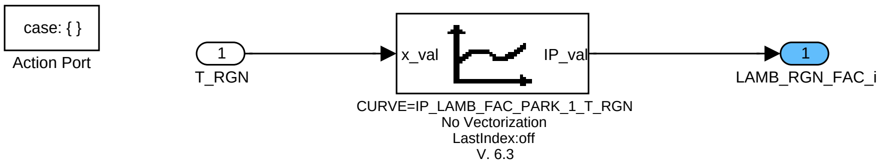


Figure 27:

53.3.2.2.1.2.3.2 Park phase for lambda setpoint - calculation of the Correction Factor at the part of second regeneration

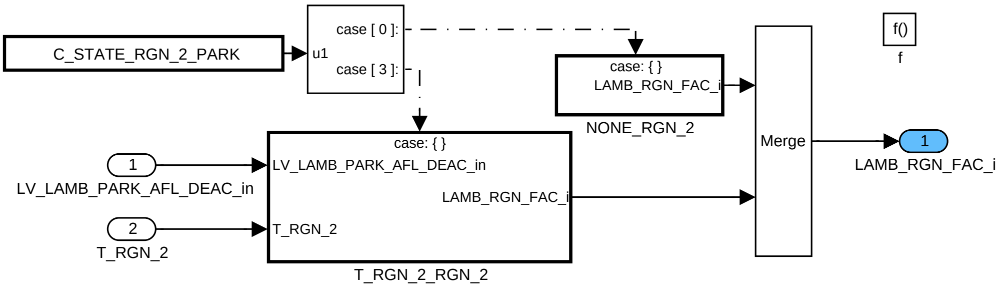


Figure 28:

53.3.2.2.1.2.3.2.1 Park phase - 2nd regeneration - no Lambda correction

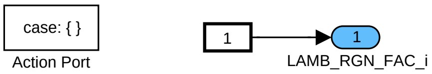


Figure 29:

53.3.2.2.1.2.3.2.2 Park phase - 2nd regeneration - Lambda correction by regeneration time of 2nd regeneration

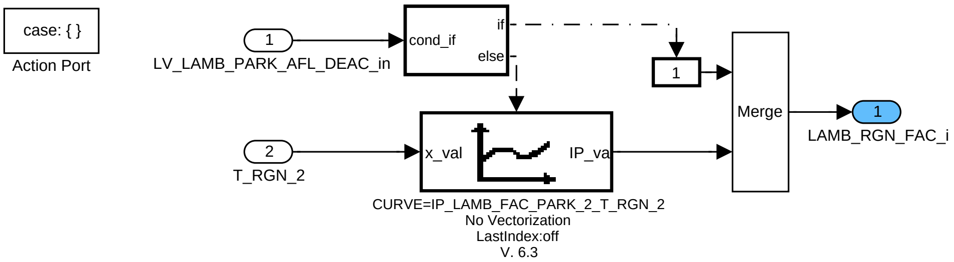


Figure 30:

53.3.2.2.1.2.4 Regeneration with Low Amplitude

The low amplitude is set for bank which should be regenerated longer. The lambda value is limited to the maximal value to avoid a jump to lean condition.

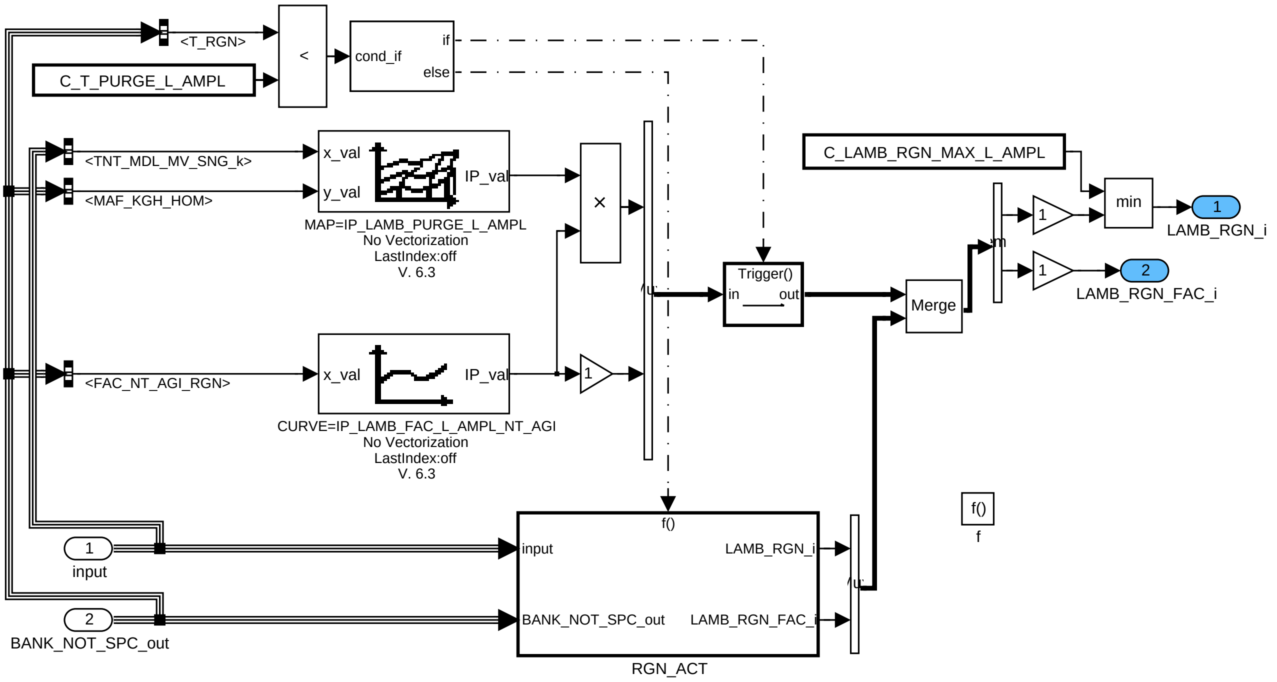

<T_RGN>
C_T_PURGE_L_AMPL
<
cond_if
if
else
<TNT_MDL_MV_SNG_k>
<MAF_KGH_HOM>
x_val
y_val
IP_val
MAP=IP_LAMB_PURGE_L_AMPL
No Vectorization
LastIndex:off
V. 6.3
×
<FAC_NT_AGI_RGN>
x_val
IP_val
CURVE=IP_LAMB_FAC_L_AMPL_NT_AGI
No Vectorization
LastIndex:off
V. 6.3
1
Trigger()
in
out
Merge
C_LAMB_RGN_MAX_L_AMPL
min
1
LAMB_RGN_i
1
2
LAMB_RGN_FAC_i
f()
f
1
input
2
BANK_NOT_SPC_out
f()
input
LAMB_RGN_i
BANK_NOT_SPC_out
LAMB_RGN_FAC_i
RGN_ACT

Figure 31:

53.3.2.2.1.2.4.1 Regeneration with Low Amplitude - calculation of the Correction Factor

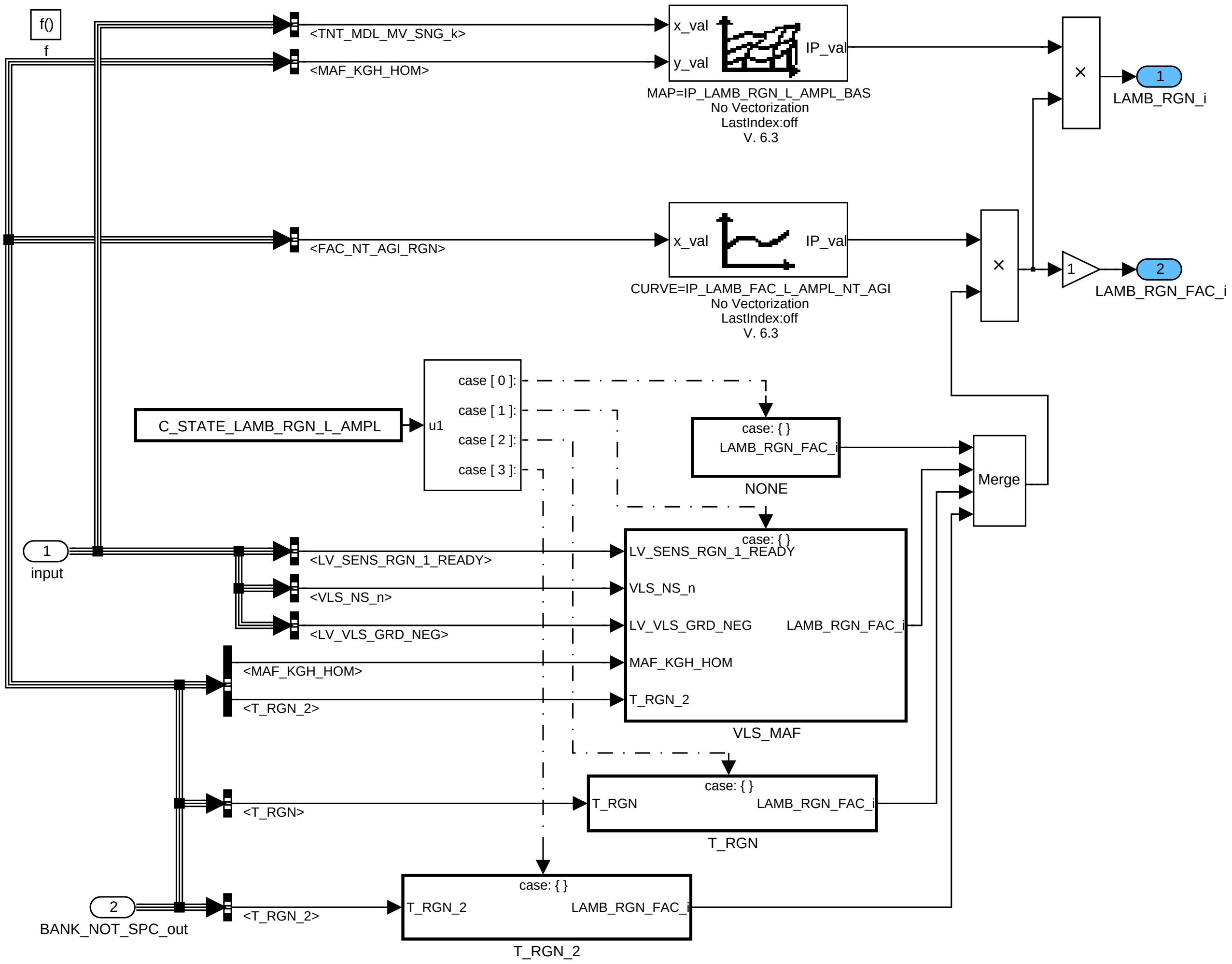


Figure 32:

53.3.2.2.1.2.4.1.1 Regeneration with Low Amplitude - no Lambda correction

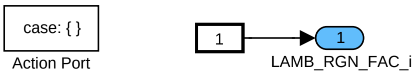


Figure 33:

53.3.2.2.1.2.4.1.2 Regeneration with Low Amplitude - Lambda correction by binary O2 signal of NOx sensor and air mass flow

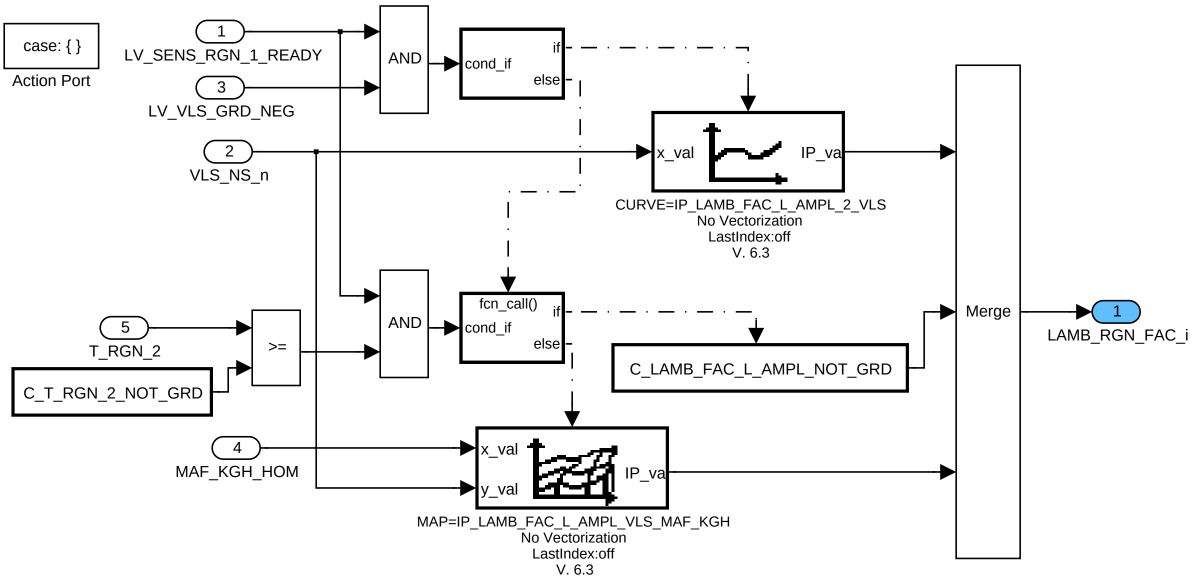


Figure 34:

53.3.2.2.1.2.4.1.3 Regeneration with Low Amplitude - Lambda correction by regeneration time

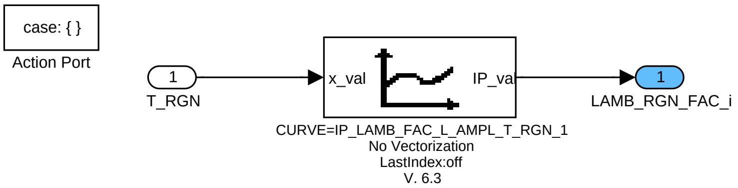


Figure 35:

53.3.2.2.1.2.4.1.4 Regeneration with Low Amplitude - Lambda correction by regeneration time of 2nd regeneration

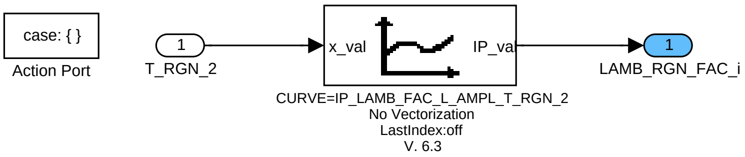


Figure 36:

53.3.2.2.1.2.5 Regeneration with equal amplitude

The equal amplitude is set for both banks as long as the NOx sensor VLS signal is not valid.

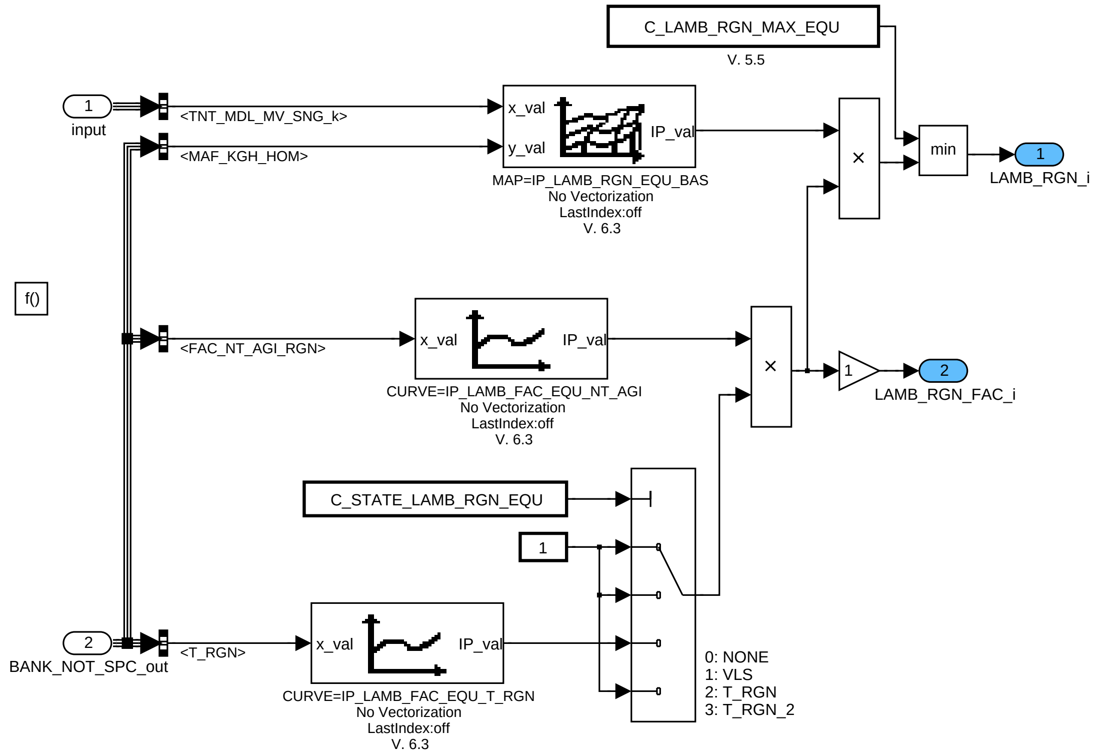


Figure 37:

53.3.2.2.1.3 Signal combination to signal vectors

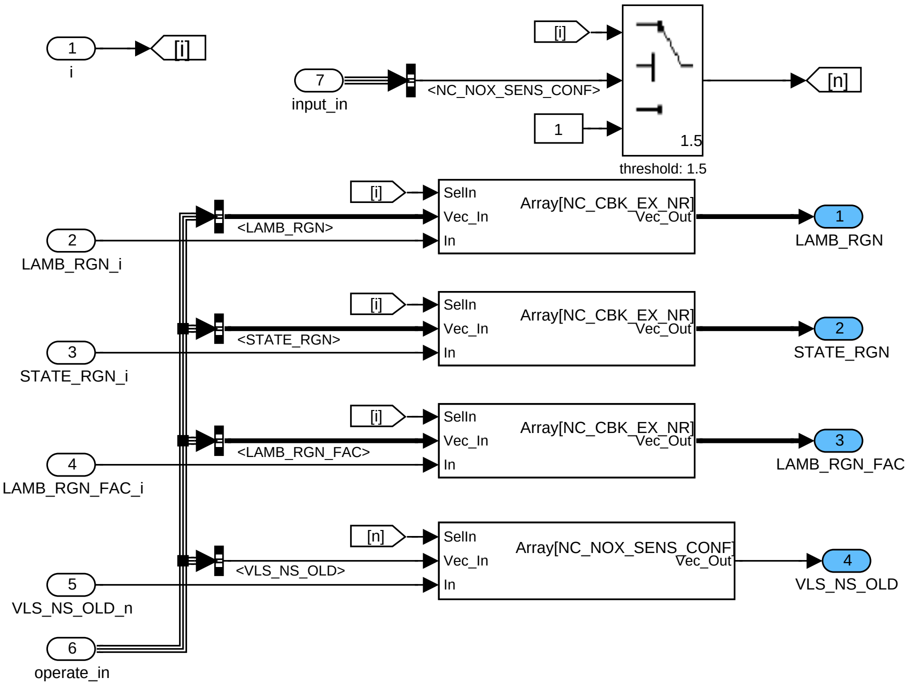


Figure 38:

53.4 NOx engine out emission homogeneous-stratified mode

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_NOX_PRED_CLC_INH	O/V	0... 1H	0... 1	1	[-]
Inhibition of calculation of predicted NOx emissions					
MAP_DIF_NOX_HOMS	V	8000... 7FFFH	-2717.04145876 ...2716.95854124	0.0829175	[hPa]
Pressure difference to basis ambient pressure of the model (homogeneous-stratified combustion mode)					
NOX_COR_HOMS	O/V	0... FFFFH	0... 1023.984375	0.015625	[mg/s]
Corrected engine out NOx emission in homogeneous-stratified combustion mode					
NOX_COR_HOMS_PRED	O/V	0... FFFFH	0... 1023.984375	0.015625	[mg/s]
Predicted engine out NOx emission for homogeneous-stratified combustion mode					
NOX_COR_HOMS_PRED_TMP	V	0... FFFFH	0... 2047.96875	0.03125	[mg/s]
Predicted engine out NOx emission if homogeneous-stratified combustion mode would be active (intermediate)					
NOX_COR_HOMS_TMP	V	0... FFFFH	0... 2047.96875	0.03125	[mg/s]
Corrected engine out NOx emission in homogeneous-stratified combustion mode (intermediate)					
RATIO_EGR_DIF_NOX_HOMS	V	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Exhaust gas recirculation ratio difference to basis calibration (homogeneous-stratified combustion mode)					
TCO_DIF_NOX_HOMS	V	80... 7FH	-96... 95.25	0.75	[°C]
Cooling water temperature difference to basis cooling temperature of the model (homogeneous-stratified combustion mode)					

Input Data:

EGR_RATIO	FAC_MAF_MAX	GEAR	LV_AT
LV_CLU_SWI	LV_GS	LV_NOX_COR_NS_AD	LV_ST_END
MAP_MES	MFF_SP_HOMS	MFF_SP_HOMS_PRED	N_32
PQ_EGR	PQ_EGR_SP	TCO	TIA_IM
LV_VAR_TCT			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_AMP_BAS_NOX_HOMS	1	0... FFFFH	0... 5434	0.0829175	[hPa]
Basis ambient pressure for NOx engine out emission model (homogeneous-stratified combustion mode)					
C_CRLC_HOMS_PRED_FALL	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation constant for predicted value of falling NOx emissions at HOMS					
C_CRLC_HOMS_PRED_RISE	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation constant for predicted value of rising NOx emissions at HOMS					
C_TCO_BAS_NOX_HOMS	1	0... FEH	-48... 142.5	0.75	[°C]
Basis cooling water temperature for NOx engine out emission model (homogeneous-stratified combustion mode)					
IP_FAC_MAF_MAX_NOX_HOMS	4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDP_FAC_MAF_MAX_IP_FAC_HOMS	4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Engine out NOx emission correction due to the throttle factor (homogeneous-stratified combustion mode)					
IP_NOX_EGR_RATIO_BAS_HOMS	8*12	0... FFFFH	-50... 49.9984741211	1.52588e-3	[%]
LDPM_MFF_SP_1_NOXM	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDPM_N_32_1_NOXM	12	0... FFH	0... 8160	32	[rpm]
EGR reference value at NOx emission basis calibration (homogeneous-stratified combustion mode)					

IP_NOX_EGR_RATIO_DIF_HOMS	8*8	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDPM_PQ_EGR_1_NOXM	8	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
LDP_RATIO_EGR_DIF_IP_NOX_HOMS	8	0... FFFFH	-50... 49.9984741211	1.52588e-3	[%]
Engine out NOx emission correction due to EGR deviation (homogeneous-stratified combustion mode)					
IP_NOX_HOMS	8*12	0... FFFFH	0... 1023.984375	0.015625	[mg/s]
LDPM_MFF_SP_1_NOXM	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDPM_N_32_1_NOXM	12	0... FFH	0... 8160	32	[rpm]
NOx engine out base emission in homogeneous-stratified combustion mode					
IP_NOX_HOMS_NS_AD	8*8	0... FFFFH	0... 1023.984375	0.015625	[mg/s]
LDP_MFF_SP_IP_NOX_HOMS_NS_AD	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDP_N_32_IP_NOX_HOMS_NS_AD	8	0... FFH	0... 8160	32	[rpm]
NOx engine out base emission in homogeneous operation for NOx signal gain adaptation					
IP_NOX_MAP_DIF_HOMS	8*8	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDPM_PQ_EGR_1_NOXM	8	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
LDPM_MAP_DIF_NOX_1_NOXM	8	0... FFFFH	-2717.04145876 ...2716.95854124	0.0829175	[hPa]
Engine out NOx emission correction due to ambient pressure deviation (homogeneous-stratified combustion mode)					
IP_NOX_TCO_HOMS	8*4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDPM_MAP_DIF_NOX_1_NOXM	8	0... FFFFH	-2717.04145876 ...2716.95854124	0.0829175	[hPa]
LDPM_TCO_DIF_NOX_1_NOXM	4	0... FFH	-96... 95.25	0.75	[°C]
Engine out NOx emission correction due to cooling water temperature (homogeneous-stratified combustion mode)					
IP_NOX_TCO_HOMS_PRED	4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDPM_TCO_DIF_NOX_1_NOXM	4	0... FFH	-96... 95.25	0.75	[°C]
Prediction of engine out NOx emission correction due to cooling water temperature (homogeneous-stratified combustion mode)					
IP_NOX_TIA_HOMS	8*4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDP_N_32_IP_NOX_TIA_HOMS	8	0... FFH	0... 8160	32	[rpm]
LDP_TIA_IM_IP_NOX_TIA_HOMS	4	0... FEH	-48... 142.5	0.75	[°C]
Engine out NOx emission correction due to intake air temperature (homogeneous-stratified combustion mode)					
LC_GS_PRED_ENA	1	0... 1H	0... 1	1	[-]
Selection if NOx emission prediction calculation will be disabled during LV_GS=1					
LC_MFF_HOMS_PRED_ENA	1	0... 1H	0... 1	1	[-]
Selection for predicted emissions (0: MFF_SP_HOMS, 1: MFF_SP_HOMS_PRED)					
LC_MT_PRED_ENA	1	0... 1H	0... 1	1	[-]
Selection if NOx emission prediction calculation will be disabled during manual gear shift					

General Information

This module calculates the NOx engine out emission mass flow at homogeneous-stratified mode. This module guesses also a predicted value, if the homogeneous-stratified mode would become active at the actual conditions.
The calculations consist of an basis calibration, depending on fuel mass, motor speed and ambient temperature. This value will be corrected by several factors, which depend from the deviation of state variables compared to the basis calibration.

Application Conditions

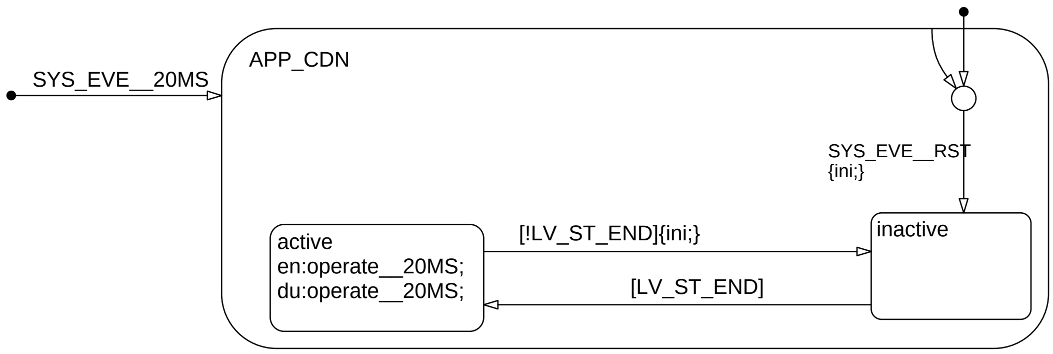
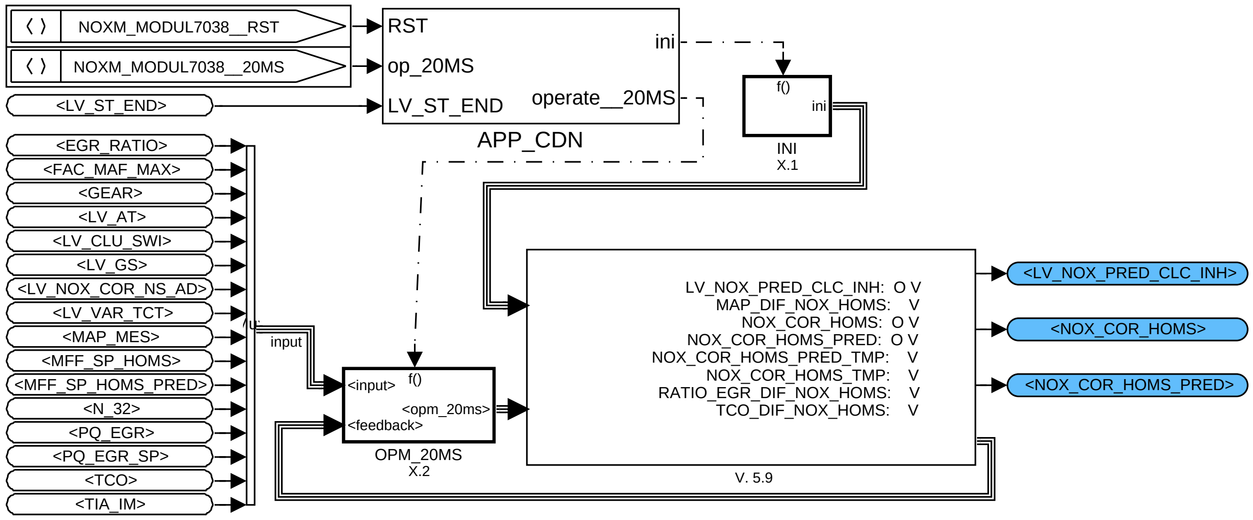


Figure 39:
Path: NOXM_MODUL7038/APP_CDN/Chart

Function description



SDA_SRS / SDA V 5.0.4 21−Nov−2006

Figure 40:
Path: NOXM_MODUL7038

53.4.1 Initialization at reset and deactivation

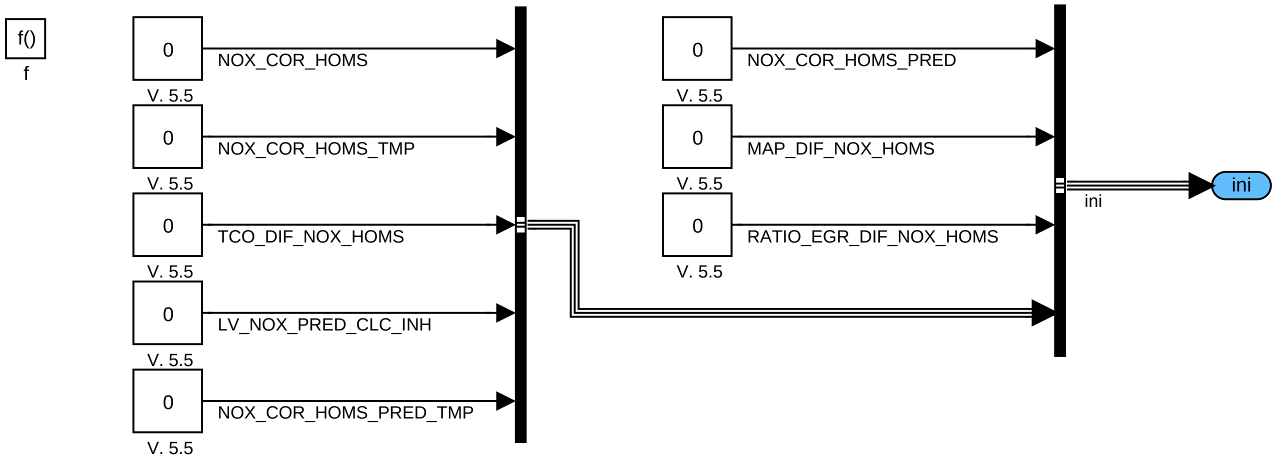


Figure 41:
Path: NOXM_MODUL7038/INI

53.4.2 Calculation at 20 ms

53.4.2.1 Deviation of air pressure

Figure 42:
Path: NOXM_MODUL7038/OPM_20MS/CLC1

53.4.2.2 Deviation of cooling water temperature

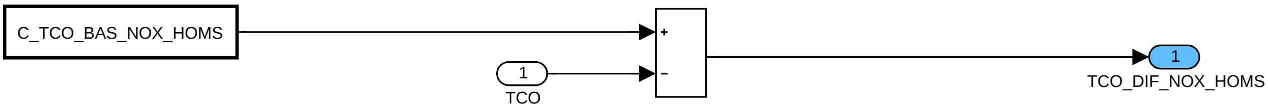


Figure 43:
Path: NOXM_MODUL7038/OPM_20MS/CLC2

53.4.2.3 Deviation of exhaust gas recirculation

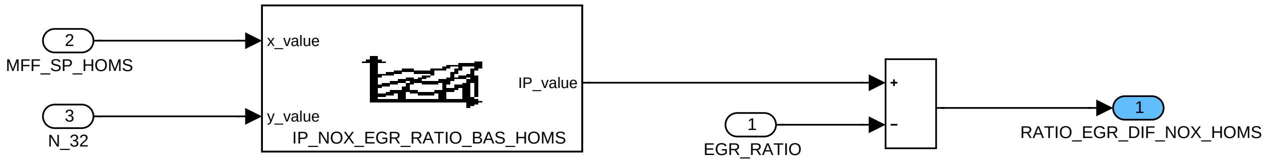


Figure 44:
Path: NOXM_MODUL7038/OPM_20MS/CLC3

53.4.2.4 Basis calibration of NOx engine out emissions

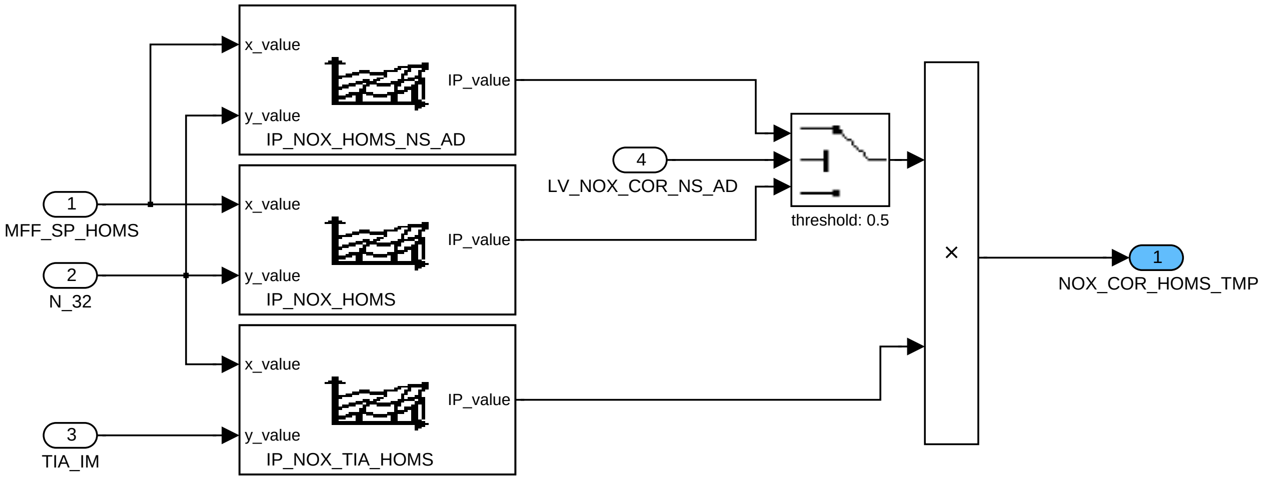


Figure 45:
Path: NOXM_MODUL7038/OPM_20MS/CLC4

53.4.2.5 Corrected engine out NOx emissions

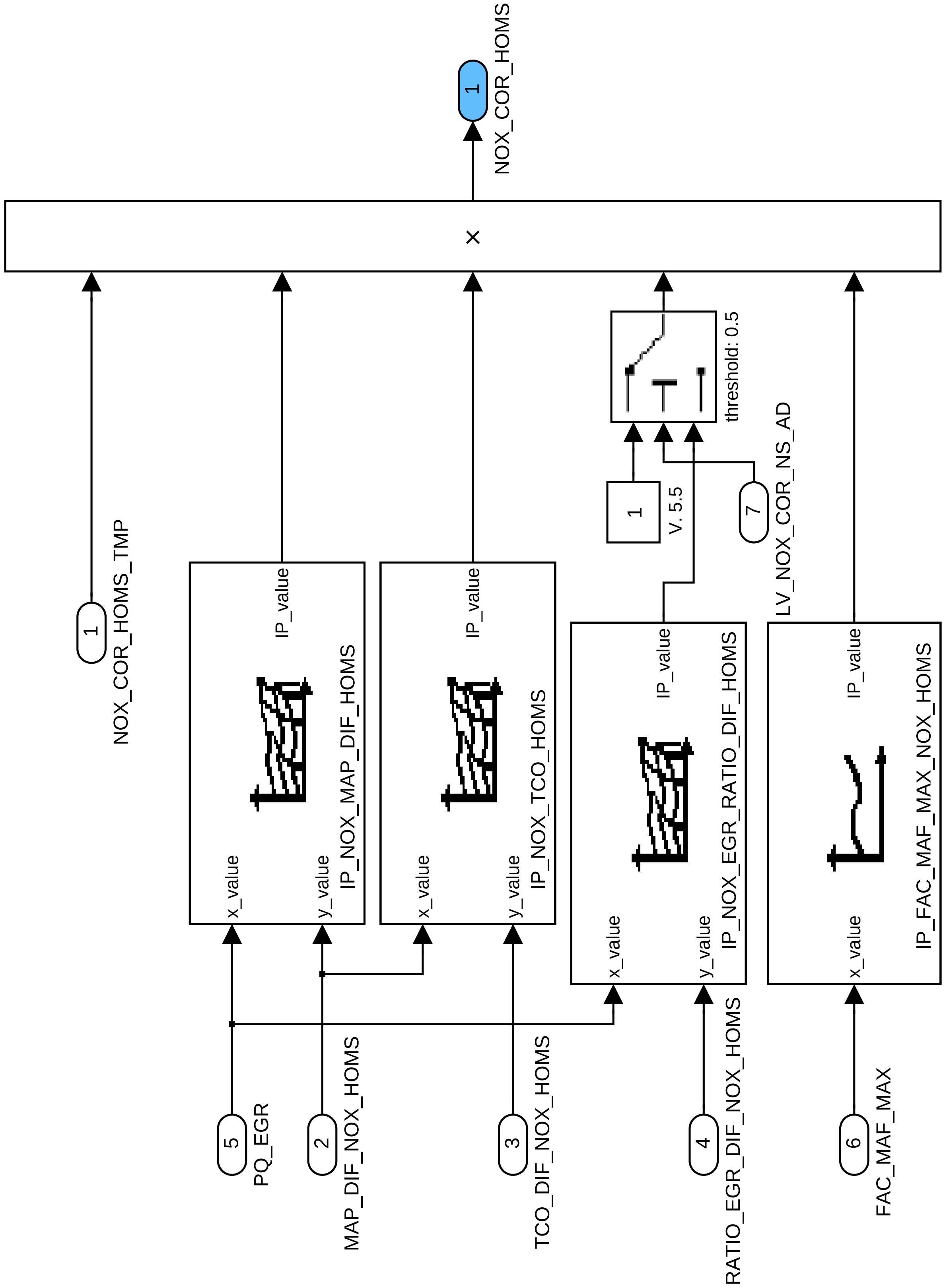


Figure 46:
Path: NOXM_MODUL7038/OPM_20MS/CLC5

53.4.2.6 Condition to stop calculation of predicted NOx emissions

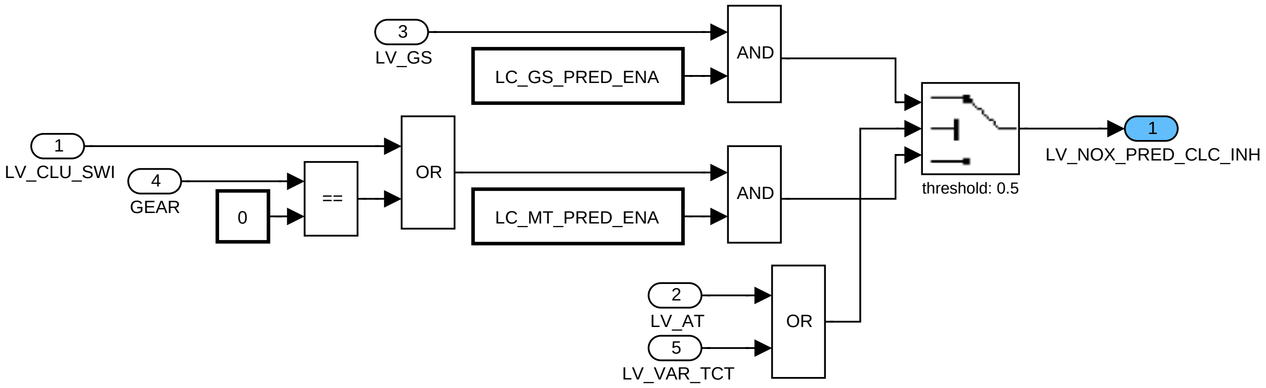


Figure 47:
Path: NOXM_MODUL7038/OPM_20MS/CLC7

53.4.2.7 Basis for predicted NOx emissions

The calibration maps are the same as for actual emissions. They deliver the predicted emissions, because the predicted fuel mass MFF_SP_HOMS_PRED may be used here.

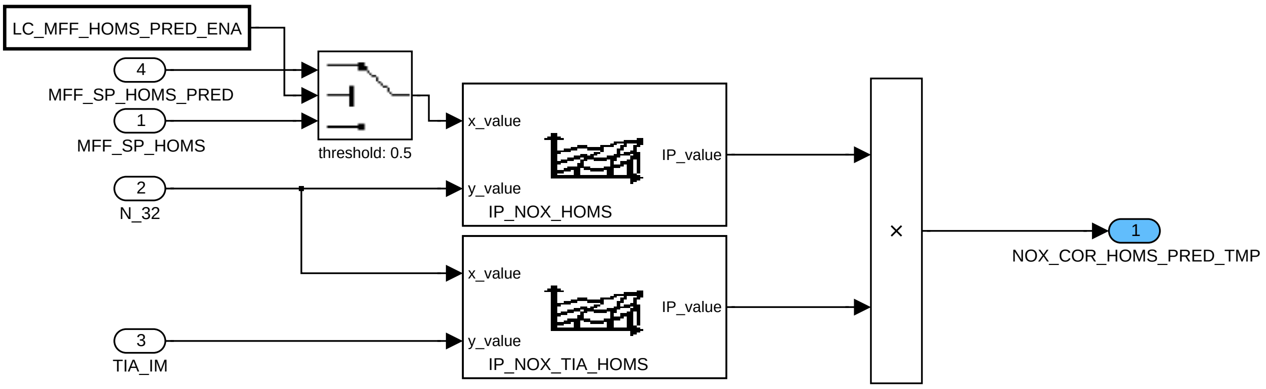


Figure 48:
Path: NOXM_MODUL7038/OPM_20MS/CLC8

53.4.2.8 Predicted engine out NOx emissions

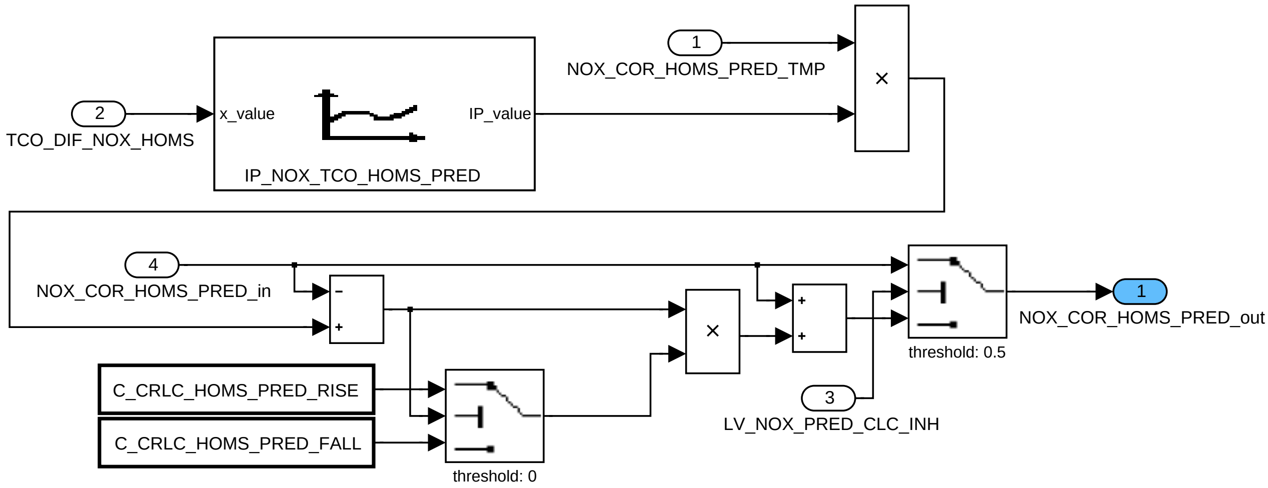


Figure 49:
Path: NOXM_MODUL7038/OPM_20MS/CLC9

53.5 NOx engine out emission stratified mode

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_NOX_COR_NS_AD	O/V	0... 1H	0... 1	1	[-]
NOx emission calculation for NOx signal gain adaptation					
MAP_DIF_NOX_S	V	8000... 7FFFH	-2717.04145876 ...2716.95854124	0.0829175	[hPa]
Pressure difference to basis ambient pressure of the model					
NOX_COR_S	O/V	0... FFFFH	0... 1023.984375	0.015625	[mg/s]
Corrected engine out NOx emission in stratified operation mode					
NOX_COR_S_PRED	O/V	0... FFFFH	0... 1023.984375	0.015625	[mg/s]
Predicted engine out NOx emission for stratified operation mode					
NOX_COR_S_PRED_TMP	V	0... FFFFH	0... 2047.96875	0.03125	[mg/s]
Predicted engine out NOx emission for stratified operation mode (intermediate)					
NOX_COR_S_TMP	V	0... FFFFH	0... 2047.96875	0.03125	[mg/s]
Corrected engine out NOx emission in stratified operation mode (intermediate)					
RATIO_EGR_DIF_NOX_S	V	8000... 7FFFH	-50... 49.9984741211	1.52588e-3	[%]
Exhaust gas recirculation ratio difference to basis calibration					
TCO_DIF_NOX_S	V	80... 7FH	-96... 95.25	0.75	[°C]
Cooling water temperature difference to basis cooling temperature of the model					

Input Data:

EGR_RATIO	FAC_MAF_MAX	LV_NOX_PRED_CLC_INH	LV_ST_END
MAP_MES	MFF_SP_S	MFF_SP_S_PRED	N_32
NC_NOX_SENS_CONF	PQ_EGR	PQ_EGR_SP	OPM_REQ_CUS
TCO	TIA_IM		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_AMP_BAS_NOX_S	1	0... FFFFH	0... 5434	0.0829175	[hPa]
Basis ambient pressure for NOx engine out emission model					
C_CRLC_S_PRED_FALL	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation constant for predicted value of falling NOx emissions at S					
C_CRLC_S_PRED_RISE	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Correlation constant for predicted value of rising NOx emissions at S					
C_STATE_OPM_NOX_S_NS_AD	1	0... FFFFH	0... 65535	1	[-]
State of combustion mode for switch to correction factor valid at NOx signal gain adaptation					
C_TCO_BAS_NOX_S	1	0... FEH	-48... 142.5	0.75	[°C]
Basis cooling water temperature for NOx engine out emission model					
IP_FAC_MAF_MAX_NOX_S	4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDP_FAC_MAF_MAX_IP_FAC_S	4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Engine out NOx emission correction due to the throttle factor					
IP_NOX_EGR_RATIO_BAS_S	12*12	0... FFFFH	-50... 49.9984741211	1.52588e-3	[%]
LDPM_MFF_SP_2_NOXM	12	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDPM_N_32_3_NOXM	12	0... FFH	0... 8160	32	[rpm]

EGR reference value at NOx emission basis calibration					
IP_NOX_EGR_RATIO_DIF_S	8*8	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDPM_PQ_EGR_2_NOXM	8	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
LDP_RATIO_EGR_DIF_IP_NOX_S	8	0... FFFFH	-50... 49.9984741211	1.52588e-3	[%]
Engine out NOx emission correction due to EGR deviation					
IP_NOX_MAP_DIF_S	8*8	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDPM_PQ_EGR_2_NOXM	8	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
LDPM_MAP_DIF_NOX_2_NOXM	8	0... FFFFH	-2717.04145876 ...2716.95854124	0.0829175	[hPa]
Engine out NOx emission correction due to ambient pressure deviation					
IP_NOX_S	12*12	0... FFFFH	0... 1023.984375	0.015625	[mg/s]
LDPM_MFF_SP_2_NOXM	12	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDPM_N_32_3_NOXM	12	0... FFH	0... 8160	32	[rpm]
NOx engine out base emission in stratified operation					
IP_NOX_S_NS_AD	8*8	0... FFFFH	0... 1023.984375	0.015625	[mg/s]
LDP_MFF_SP_IP_NOX_S_NS_AD	8	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDP_N_32_IP_NOX_S_NS_AD	8	0... FFH	0... 8160	32	[rpm]
NOx engine out base emission in stratified operation for NOx signal gain adaptation					
IP_NOX_TCO_S	8*4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDPM_MAP_DIF_NOX_2_NOXM	8	0... FFFFH	-2717.04145876 ...2716.95854124	0.0829175	[hPa]
LDPM_TCO_DIF_NOX_2_NOXM	4	0... FFH	-96... 95.25	0.75	[°C]
Engine out NOx emission correction due to cooling water temperature					
IP_NOX_TCO_S_PRED	4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDPM_TCO_DIF_NOX_2_NOXM	4	0... FFH	-96... 95.25	0.75	[°C]
Prediction of engine out NOx emission correction due to cooling water temperature					
IP_NOX_TIA_S	8*4	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
LDP_N_32_IP_NOX_TIA_S	8	0... FFH	0... 8160	32	[rpm]
LDP_TIA_IM_IP_NOX_TIA_S	4	0... FEH	-48... 142.5	0.75	[°C]
Engine out NOx emission correction due to intake air temperature					
LC_MFF_S_PRED_ENA	1	0... 1H	0... 1	1	[-]
Selection for predicted emissions (0: MFF_SP_S, 1: MFF_SP_S_PRED)					

General Information

This module calculates the NOx engine out emission mass flow at stratified mode. This module guesses also a predicted value, if the stratified mode would become active at the actual conditions.
The calculations consist of an basis calibration, depending on fuel mass, motor speed and ambient temperature. This value will be corrected by several factors, which depend from the deviation of state variables compared to the basis calibration.

Application Conditions

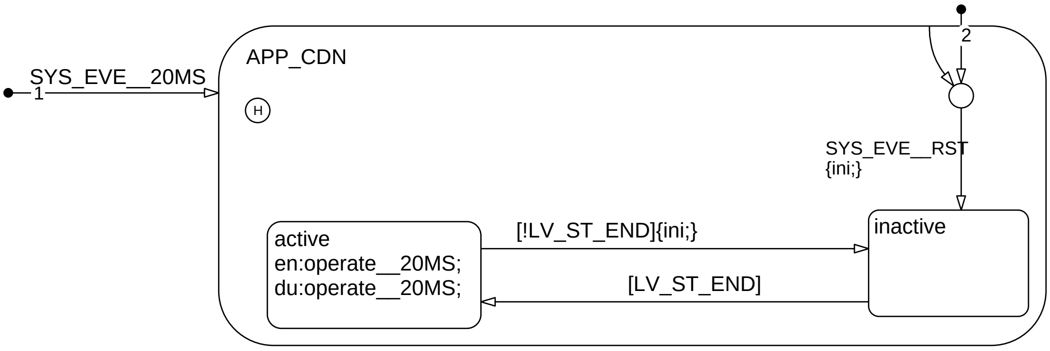


Figure 50:

Function description

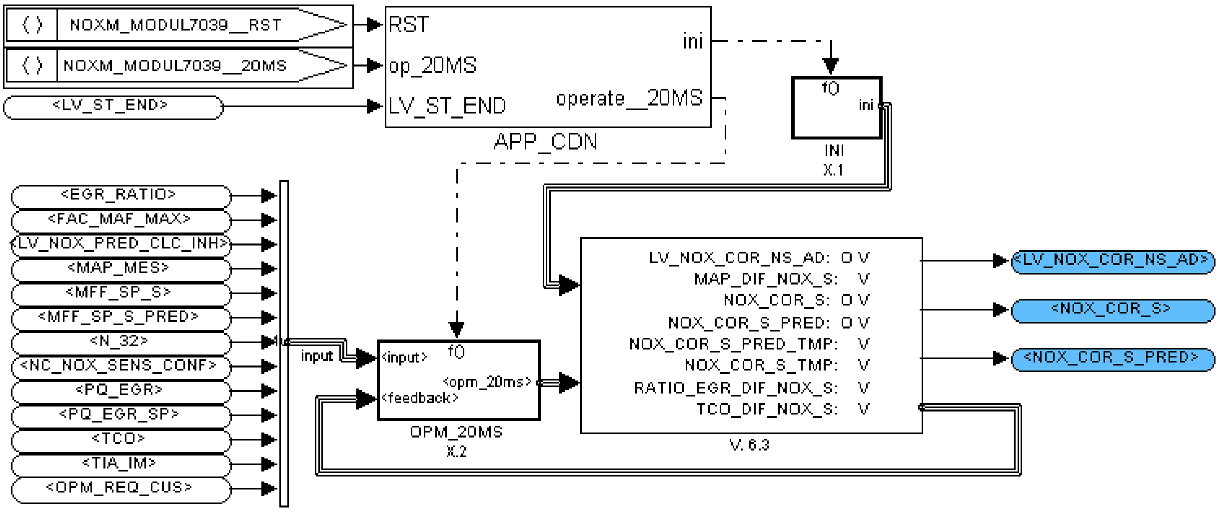


Figure 51:

53.5.1 Initialization at reset and deactivation

53.5.2 Calculation at 20 ms

53.5.2.1 Deviation of air pressure

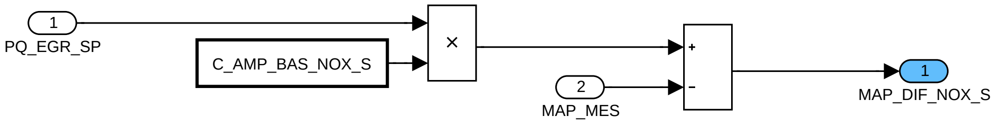


Figure 52:

53.5.2.2 Deviation of cooling water temperature

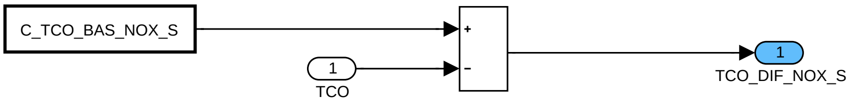


Figure 53:

53.5.2.3 Deviation of exhaust gas recirculation

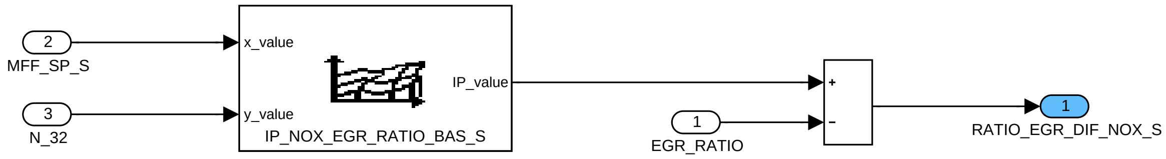


Figure 54:

53.5.2.4 Detection of NOx signal gain adaptation

If the NOx signal gain adaptation for any NOx sensor is in required combustion mode, then a special basis calibration matrix is used.

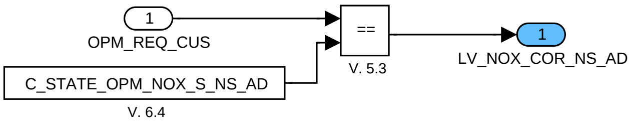


Figure 55:

53.5.2.5 Basis calibration of NOx engine out emissions

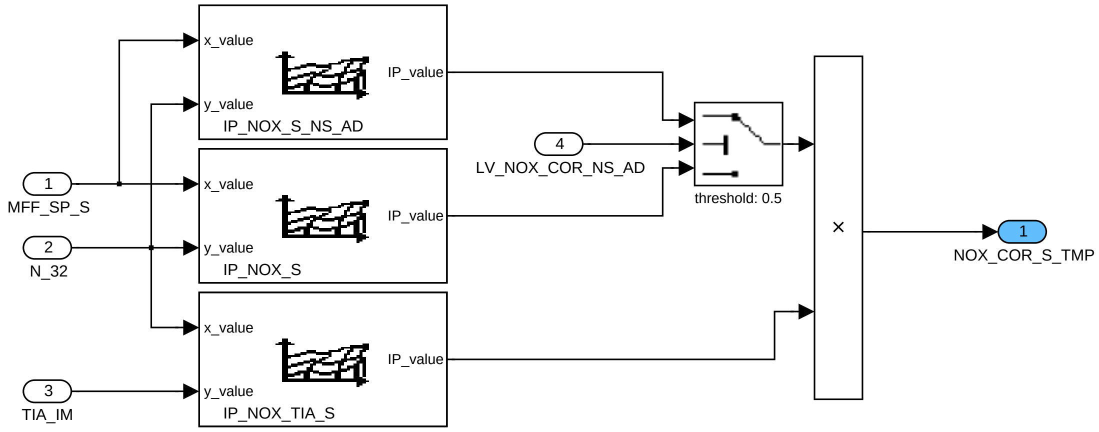


Figure 56:

53.5.2.6 Corrected engine out NOx emissions

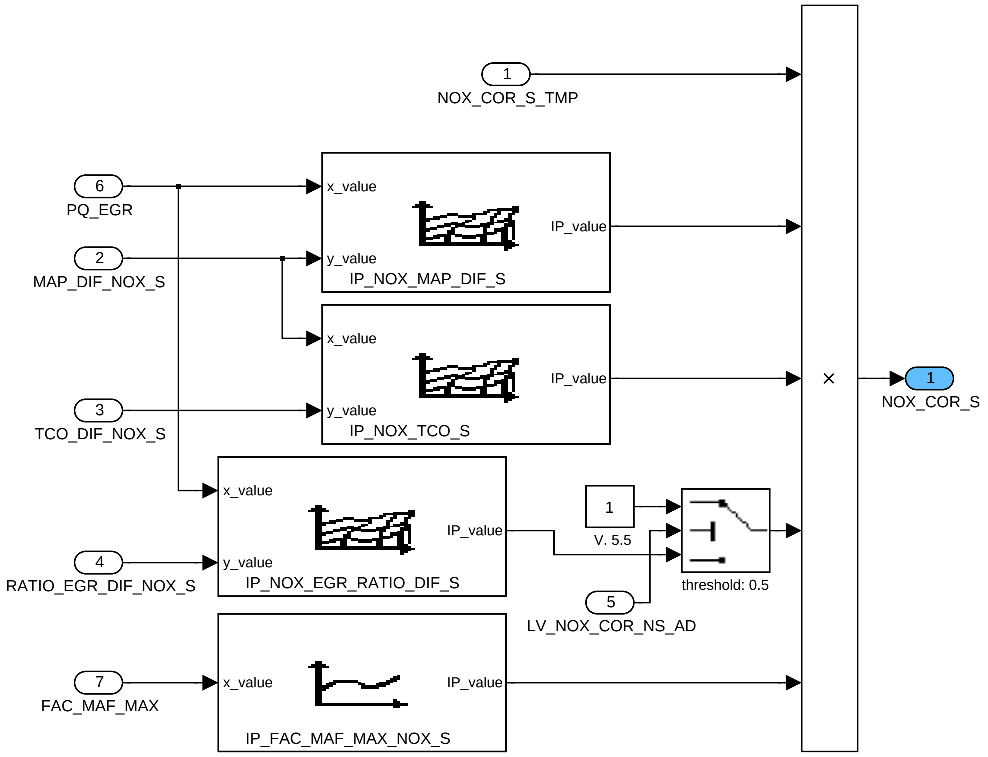


Figure 57:

53.5.2.7 Basis for predicted NOx emissions

The calibration maps are the same as for actual emissions. They deliver the predicted emissions, because the predicted fuel mass MFF_SP_S_PRED may be used here.

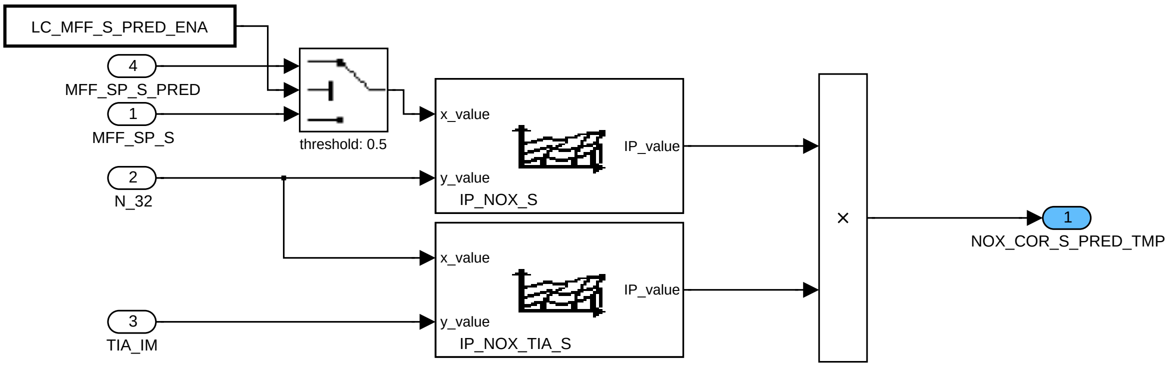


Figure 58:

53.5.2.8 Predicted engine out NOx emissions

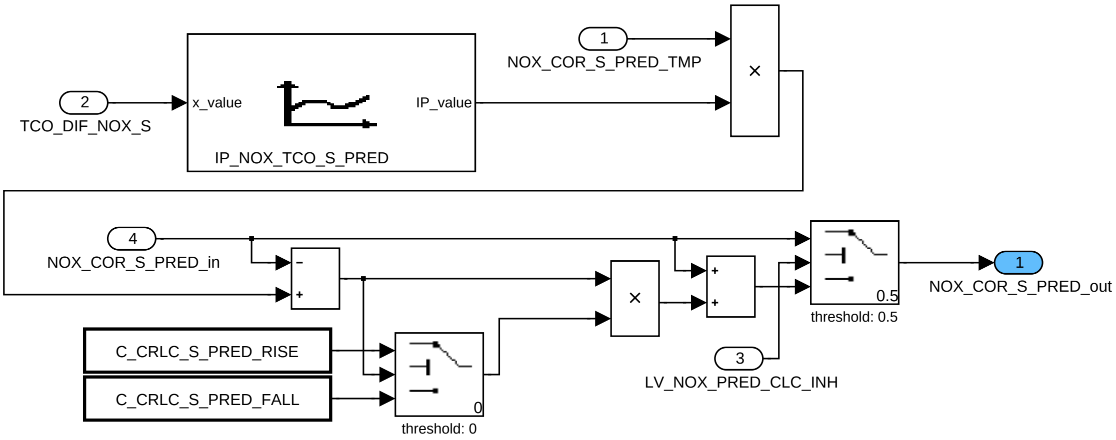


Figure 59:

53.6 NOx catalyst management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_NOX	O/V	0H 1H 2H 3H 4H 5H	PASSIV LOAD REGENERATION WAIT STOP WARMUP	1	[-]
NOx catalyst state					
LV_NT_ACT	O/V	0...1H	0...1	1	[-]
Flag for NOx catalyst function activation					
LV_RGN_NT_REQ	O/V	0...1H	0...1	1	[-]
Logical Variable for NOx trap regeneration phase request					
LV_NT_AFS_REQ	O/V	0...1H	0...1	1	[-]
Request of lambda = 1 operation mode					
LV_NOX_SENS	O/V	0...1H	0...1	1	[-]
Logical Variable for the existence of a NOx-Sensor					
LV_NOX_SENS_RGN	O/V	0...1H	0...1	1	[-]
Regeneration request only due to NOx sensor (loading model blocked)					
LV_RGN_AGI[NC_NT_NR]	O/V	0...1H	0...1	1	[-]
Flag used for communication with aging function					
LV_RGN_AGI_VLD[NC_NT_NR]	O/V	0...1H	0...1	1	[-]
Flag used for communication with aging function					
LV_TRIG_AFR_AFL	O/V	0...1H	0...1	1	[-]
Logical value for triggering a transition of rich to lean operation					
LV_AFL_NOT	O/V	0...1H	0...1	1	[-]
Logical value: lambda1 or richer operation is requested or active					
LV_RGN_STOP_SENS	O/V	0...1H	0...1	1	[-]
Flag indicating: Stop of Rgn by sensor enabled					
LV_RGN_STOP_MDL	O/V	0...1H	0...1	1	[-]
Flag indicating: Stop of Rgn out of model enabled					
LV_RGN_STOP_TOUT	O/V	0...1H	0...1	1	[-]
Flag indicating: Stop of Rgn by time counter enabled					
LV_RGN_REQ_NTLD	O/V	0...1H	0...1	1	[-]
Logical value for regeneration phase request because of the NOx trap loading degree					
LV_RGN_REQ_NTLD_AFS	O/V	0...1H	0...1	1	[-]
Logical value for regeneration phase request before switching to stoichiometric operation or rich operation					
LV_RGN_REQ_NOX_SENS	O/V	0...1H	0...1	1	[-]
Regeneration request caused by the NOx-Sensor signal					
LV_RGN_REQ_NOX_SENS_TMP[NC_NT_NR]	O/V	0...1H	0...1	1	[-]
Bench-selective regeneration request caused by the NOx-Sensor signal					
LV_RGN_CDN	O/V	0...1H	0...1	1	[-]
Flag for testing whether or not regeneration conditions are given					
LV_ACT_INT_PUC_AFL[NC_NT_NR]	O/V	0...1H	0...1	1	[-]
Request for catalyst purge activation when leaving lean burn condition					
LV_SENS_AFR[NC_NT_NR]	O/V	0...1H	0...1	1	[-]
downstream sensor measures rich exhaust gas (stop of regeneration)					
LV_SENS_AFR_TRIM_TMP[NC_NT_NR]	O/V	0...1H	0...1	1	[-]
downstream sensor measures rich exhaust gas (homogeneous detection threshold) (bench-selective value)					
LV_SENS_AFR_TRIM	O/V	0...1H	0...1	1	[-]
downstream sensor sees rich exhaust gas (homogeneous detection threshold)					
LV_NT_HOM_INI	O/V	0...1H	0...1	1	[-]
flag: NOx trap initialised due to sufficient hom. stoic. operation (accumulation effects eliminated)					
LV_NOX_SENS_NOX_VLD_RGN_ST[NC_NT_NR]	O/V	0...1H	0...1	1	[-]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
flag: NOx sensor NOx signal validity at the moment the NOx sensor rgn request is set					
LV_T_AFL_MIN	O/V	0...1H	0...1	1	[-]
flag: minimum lean time for setting a request elapsed					
NOX_CONC_RAW	O/V	0...FFFFH	0...65535	1	[ppm]
NOx concentration engine out					
NOX_CONC	O/V	0...FFFFH	0...65535	1	[ppm]
NOx concentration engine out (after raw emission adaptation correction)					
NOX_CONC_MMV	V	0...FFFFH	0...65535	1	[ppm]
filtered NOx engine out concentration					
NOX_CONC_RED	O/V	0...FFFFH	0...65535	1	[ppm]
corrected NOx concentration in catalyst (incl. SSR)					
NOX_CONC_RED_MMV	O/V	0...FFFFH	0...65535	1	[ppm]
filtered corrected NOx concentration in catalyst (incl. SSR)					
NOX_CONC_RED_MMV_RGN_ST	O/V	0...FFFFH	0...65535	1	[ppm]
filtered corrected NOx concentration in catalyst (incl. SSR) at NOx regeneration start					
NOX_NS_AD_TMP[NC_NOX_SENS_CONF]	V	0...05DCH	0...1500	1	[ppm]
temporary internal NOx sensor NOx value for regeneration request calculation					
NOX_COR	V	0...FFFFH	0...1023.98437	0.015625	[mg/s]
raw engine out NOx emission					
NOX_COR_RED	O/V	0...FFFFH	0...1023.98437	0.015625	[mg/s]
corrected engine out NOx emission (incl. SSR)					
NOX_FLOW	V	0...FFFFH	0...1023.98437	0.015625	[mg/s]
NOx emission flow which is stored in the catalyst					
NOX_OUT_MDL	O/V	0...FFFFH	0...1023.98437	0.015625	[mg/s]
modelled downstream NOx emission					
NOX_OUT_MDL_INT	O/V	0...FFFFH	0...10485.6	0.16	[mg]
Integral of modelled downstream NOx emission for one lean / rich cycle					
NOX_OUT_INT_TMP[NC_NOX_SENS_CONF]	V	0...FFFFH	0...10485.6	0.16	[mg]
temporary internal NOx out integral value for NOx sensor regeneration request calculation					
NOX_COR_INT	V	0...FFFFH	0...10485.6	0.16	[mg]
Integral of modelled NOx raw emissions (pre NOx catalyst) at current lean operation phase					
NT_EFF	O/V	0...8000H	0...1	0.0305e-3	[-]
NOx trap efficiency					
NT_EFF_COR	V	0...FFH	0...3.98437	0.015625	[-]
NOx trap efficiency correction factor					
NT_EFF_COR_RGN_ST	O/V	0...FFH	0...3.98437	0.015625	[-]
NOx trap efficiency correction factor at regeneration start					
NT_STC_RNG_H	V	0...FFFFH	0...20971.2	0.32	[mg]
NOx trap storage capacity					
NT_STC_BAS_RNG_H	O/V	0...FFFFH	0...20971.2	0.32	[mg]
NOx trap base storage capacity					
NTL	O/V/S	0...FFFFH	0...10485.6	0.16	[mg]
NOx catalyst absolute loading					
NTL_H	O/V	0...FFFFH	0...20971.2	0.32	[mg]
NOx catalyst absolute loading, expanded/ higher value range					
NTL_RGN_ST	O/V	0...FFFFH	0...10485.6	0.16	[mg]
NOx trap absolute loading (O2+NOx) when regeneration starts					
NTL_RGN_ST_NOX	O/V	0...FFFFH	0...10485.6	0.16	[mg]
NOx trap absolute loading (only NOx) when regeneration starts					
NTL_MIN	O/V	0...FFFFH	0...10485.6	0.16	[mg]
Minimum NOx trap loading					
NT_O2_STC	O/V	0...FFFFH	0...10485.6	0.16	[mg]
NOx trap oxygen storage capacity					
NT_O2_STC_BAS	O/V	0...FFFFH	0...10485.6	0.16	[mg]
NOx trap base oxygen storage capacity					
O2L	O/V	0...FFFFH	0...10485.6	0.16	[mg]
O2 absolute loading of the NOx trap					
NTL_DEC[NC_NT_NR]	V	0...FFFFFFFFH	0...10485.6	2.4414e-6	[mg]
Decrement of NOx trap absolute NOx loading during regeneration (lack of oxygen)					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NTL_DEC_INT[NC_NT_NR]	O/V	0...FFFFH	0...10485.6	0.16	[mg]
Integral of the lack of oxygen during regeneration phase					
NTL_DEC_INT_SWI[NC_NT_NR]	O/V	0...FFFFH	0...10485.6	0.16	[mg]
NTL_DEC_INT until the downstream sensor switches to reach exhaust gas					
FAC_RGN_DEC_SHAR	V	0...80H	0...1	0.0078125	[-]
Share factor for the reducing agent for the decrementation of NTL and O2L					
NTLD	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx trap loading degree					
NTLD_RGN_ST	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx trap loading degree at starting regeneration					
NTLD_MAX	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
loading degree threshold for starting a regeneration					
NTLD_MAX_VS	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
loading degree threshold for starting a regeneration depending on vehicle speed					
O2_FLOW	V	0...FFFFH	0...16383.75	0.25	[mg/s]
Corrected engine out O2 flow					
T_AFL	O/V	0...FFFFH	0...1310.7	0.02	[s]
elapsed time since lean phase was started					
T_AFL_PUC	O/V	0...FFFFH	0...1310.7	0.02	[s]
elapsed time since last PUC phase					
T_NT_HOM	V	0...FFFFH	0...655.35	0.01	[s]
counting back the minimum homogeneous time for detecting initialised catalyst state					
T_NT_HOM_DELTA	V	80...7FH	-1.28...1.27	0.01	[s]
decrement for T_NT_HOM counter					
SUM_NTL_CYC	O/V/S	0...FFFFH	0...65535	1	[-]
Amount of regeneration cycles carried out					
NT_AGI_AV	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
actual aging factor (ratio of actual storage capacity vs. base storage capacity)					
NOX_NS_AD_RGN_ST[NC_NT_NR]	O/V	0...05DCH	0...1500	1	[ppm]
downstream NOx concentration, measured by NOx sensor at start of regeneration					
FLOW_NT_MMV	O/V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
filtered exhaust gas flow through NOx trap					
NOX_CONC_OUT_MDL	O/V	0...FFFFH	0...65535	1	[ppm]
corrected NOx concentration after catalyst					
NOX_SENS_MAX	O/V	0...05DCH	0...1500	1	[ppm]
NOx sensor signal threshold for requesting a regeneration					
NOX_COR_RED_MMV	O/V	0...FFFFH	0...1023.98437	0.015625	[mg/s]
Moving mean value of corrected engine out NOx emission (incl. SSR)					
NOX_CONC_OUT_MDL_MMV	V	0...FFFFH	0...65535	1	[ppm]
Filtered corrected NOx concentration after catalyst					
NTLD_MDL	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx trap loading degree out of model					
LV_MDL_AFR	O/V	0...1H	0...1	1	[-]
Model rich exhaust gas (stop of regeneration)					
LV_TOUT_AFR	O/V	0...1H	0...1	1	[-]
rich exhaust gas after timeout (stop of regeneration)					
NTLD_MAX_AFS	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
maximum NOx trap loading degree for requesting a regeneration before lambda1 operation					
NT_EFF_BAS	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Basis NOx trap efficiency					
NTLD_MAX_AFS_BAS	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Basis maximum NOx trap loading degree for requesting a regeneration before lambda=1.0 operation					
T_AFL_SUM	O/V/S	0...FFFFFFFFH	0...85899345.9	0.02	[s]
sum of lean time					
NOX_SENS_MAX_BAS	V	0...05DCH	0...1500	1	[ppm]
Not corrected NOx sensor signal threshold for requesting a regeneration					
LV_SENS_RGN_1_READY	O/V	0...1H	0...1	1	[-]
End of first regeneration					
LV_VLS_GRD_NEG	O/V	0...1H	0...1	1	[-]
Negative gradient of VLS_NOX_SENS_1 between first and second regeneration					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_MIN_TMP	V	0...578H	-200...1200	1	[mV]
Temporary local minimum of VLS_NOX_SENS_1					
VLS_MAX_TMP	V	0...578H	-200...1200	1	[mV]
Temporary local maximum of VLS_NOX_SENS_1					
VLS_NOX_SENS_1_OLD	-	0...578H	-200...1200	1	[mV]
Old binary O2 signal voltage, raw value, measured by NOx-Sensor					
LV_VLS_GRD_NEG_SET_VLS	-	0...1H	0...1	1	[-]
Flag for indicate of reaching of negative gradient for set of LV_SENS_AFR_2 with VLS signal					
DIST_AFL_ST	V	0...FFFFFFFFH	0... 429496729500	100	[m]
Mileage counter value at start of lean operation phase (STATE_NOX = LOAD)					
DIST_AFL	V	0...FFFFH	0...6553.5	0.1	[km]
Mileage during current lean operation phase (STATE_NOX = LOAD)					
RATIO_NOX_OUT_INT_DIST[NC_NOX_SENS_CONF]	V	0...FFFFH	0...6.5535	0.0001	[g/km]
Ratio of NOx out emissions related to mileage of current lean operation phase					
RATIO_NOX[NC_NOX_SENS_CONF]	V	0...FFFFH	0...99.99847	1.5259e-3	[%]
Ratio of NOx out integral to NOx raw emission integral					
LV_CAT_PURGE_REQ_POST_AFL[NC_NT_NR]	O/V	0...1H	0...1	1	[-]
request for cat purge from NOx management					
T_STOP	O/V	0...FFFFH	0...1310.7	0.02	[s]
Time counter since start of STATE_NOX = STOP is active					
MAF_INT_STOP	O/V	0...FFFFH	0...1456.33333	0.0222222	[g]
Air mass integral since start of STATE_NOX = STOP is active					
MAF_INT_STOP_NTL	O/V	0...FFFFH	0...5825.33333	0.0888889	[g]
Air mass integral since start of actual STATE_NOX = STOP phase					
STATE_AFL_PURGE[NC_NT_NR]	V	0H 1H 2H	DISABLE OFF ON	1	[-]
State of catalyst purge after lean operation					
CTR_AFL_PURGE[NC_NT_NR]	V	0...FFH	0...255	1	[-]
Activation counter for catalyst purge after lean operation					
LV_NT_RGN_STOP_PUC	O/V	0...1H	0...1	1	[-]
Indicator, that a NOx regeneration was broken by engine state PUC					
T_NT_STST_ACT	V	0...FFFFH	0...655.35	0.01	[s]
Activation time of NOx catalyst state management since end of last STST cycle					
LV_NTL_DEC_INT_AFR	V	0...1H	0...1	1	[-]
Rich detection by NTL_DEC_INT ratio at the end of the regeneration of the second bank					
VLS_NOX_SENS_MMV	V	0...578H	-200...1200	1	[mV]
filtered VLS_NOX_SENS for detection of regeneration state					
LV_NTL_DEC_INT_STOP_RGN	V	0...1H	0...1	1	[-]
Ratio of NTL_DEC_INT reached to cancel the regeneration of the second bank					
LV_NT_RGN_2_NOT_VLD	V	0...1H	0...1	1	[-]
Regeneration of second bank not successful					
LV_NT_RGN_2_NOT_VLD_SET	V	0...1H	0...1	1	[-]
Set condition for Regeneration of second bank not successful					

Input data:

LV_INH_NT_ACT	LV_ST	LV_AST	LV_S_ACT
LV_PUC	LAMB_LS_UP[NC_CBK_EX_NR]	MAF_FG_CYL	LV_HOM_AFS_REQ
LV_INH_AFL	LV_INH_S	LV_HOM_ACT	LV_HOM_AFL_ACT
NOX_COR_HOMS	NOX_COR_S	TNT_MDL_MV	TNT_MDL_L
NT_AGI	NT_O2_STC_AD	LAMB_LS_UP_MIN	LAMB_LS_UP_MV
LV_RGN_REQ_AD	LV_INH_RGN_AD	LV_INH_RGN_REQ	LV_INH_RGN_REQ_NTLD

LV_SCC[NC_CBK_EX_NR]	LV_PUC_REQ	LAMB_AV_COR	
VLS_DOWN[NC_CBK_EX_NR]	VLS_NT_DOWN[NC_NOX_SENS_CONF]	VLS_NOX_SENS[NC_NOX_SENS_CONF]	LAMB_NOX_SENS[NC_NOX_SENS_CONF]
LV_NOX_SENS_NOX_VLD[NC_NOX_SENS_CONF]	NOX_NS_AD[NC_NOX_SENS_CONF]	MAF_KGH	LV_T_NS_VLD
VS	T_RGN	LV_NOX_SENS_VLS_VLD[NC_NOX_SENS_CONF]	TQI_REQ_FAST
LV_NOX_SENS_LAMB_VLD[NC_NOX_SENS_CONF]	NTLD_MDL_DIF	LV_INH_RGN_REQ_NOX_SENS	LV_NTLD_ADJ
NT_AGI_NTLD	LV_PU	LV_INH_NOX_SENS_RGN	NOX_AD_FAC
NOX_OUT_MES_INT[NC_NOX_SENS_CONF]	N_32	LV_NT_STC_MAX_AFL_ACT	LC_NOX_MDL_CTRL_ACT
LV_AT	LV_EF	LV_INH_NOX_MDL_CTRL_RGN	LV_INH_NT_RGN_STOP_MDL_DIAG[NC_NOX_SENS_CONF]
LV_ERR_EGR_2	LV_NOX_SENS_MAX_ADJ	LV_ADJ_NOX_SENS_RGN	
LV_HOM_AFS_ACT			
		NC_CBK_EX_NR	
NC_NOX_SENS_CONF	NC_NT_NR	LV_ES	T_RGN_2
LV_NOX_NS_DIAG_VLD[NC_NOX_SENS_CONF]	DIST	OPM_AV	MAF_INT_PUC_ACT
LV_CLU_SWI	LV_IS	T_IS	MAF_INT_FL_ACT
LV_FL	LV_LOAD_H	LV_IGK	LV_NT_STST
NT_AGI_S_RED	STATE_RGN[NC_CBK_EX_NR]	LV_NS_AD_REQ	LV_NS_SHIFT_CMB_INT_REQ
LV_VLS_NS_VLD[NC_NOX_SENS_CONF]			

FUNCTION DESCRIPTION:

General information:

The NOx catalyst management can handle single branch and twin branch exhaust gas lines with the configuration data NC_NT_NR.

If NC_NT_NR = 1, just one NOx catalyst is taken into account, if NC_NT_NR = 2, the software calculates the twin branch version concerning NOx catalysts.

with i = 1 ... NC_NT_NR

typical values of NC_NT_NR = 1; 2

If two or more separate cylinder banks are concerned, then

m = 1, for cylinder bank 1

m = 2, for cylinder bank 2

with m = 1 ... NC_CBK_EX_NR

typical values of NC_CBK_EX_NR = 1;2

NC_NOX_SENS_CONF considers the number of NOx sensors in the system

with k = 1 ... NC_NOX_SENS_CONF

typical values of NC_NOX_SENS_CONF = 1;2

In some passages logical expressions are calculated using information of different branches (example: A_1 and A_2). If this module is compiled for single branched exhaust lines, only the first expression (here: A_1) should be taken into consideration.

In case of LV_NT_STST = 1 all calculations of this module shall be inhibited. That's valid for all used recurrences (10ms, 20ms, 100ms, 1000ms).

Application conditions:

Initialisation: --

Recurrence: defined at each subchapter of this module,
basic recurrence of this module is 10 ms

Activation: LV_NT_STST = 0

Deactivation: LV_NT_STST = 1

53.6.1 NOx catalyst management - catalyst states

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: --

Recurrence: 10 ms

Activation: --

Deactivation: --

53.6.1.1 State transition manager

FUNCTION DESCRIPTION:

General information:

The NOx catalyst state manager should be initialized with the STATE_NOX = PASSIVE.

In order to keep the state manager as simple as possible without loss of functionality, the state STOP contains all engine states based on stoichiometric operation except the regeneration. Such possible states are besides the „ordinary“ lambda=1 operation the desulfation, the catalyst heating (see lambda=1-request chapter). Due to lambda oscillations in lambda=1 operation, the consideration of LV_NT_AFS_REQ is necessary in order to prevent an undesired leave of STOP.

The flag LV_RGN_AGI_i is set to 1 when the regeneration just starts (STATE_NOX switches from LOAD to REGENERATION) and the starting regeneration was requested by the modelled loading degree (LV_RGN_REQ_NTLD=1). It is reset, when the regeneration is stopped by a downstream sensor signal (see Sub-chapter: STATE_NOX = STOP).

STATE_NOX has to be initialised with PASSIVE

The states of the NOx Trap function are:

- ***PASSIVE*** : Initialisation of NOx catalyst management
- ***WARMUP***: Sensors ready, waiting for suitable catalyst temperature
- ***LOAD*** : storage of NOx (and oxygen) during lean burn phases
- ***REGENERATION*** : NOx-regeneration of the catalyst during a rich burn phase
- ***STOP*** : Lambda=1 operation
- ***WAIT***: During Fuel Cut Off, only oxygen crosses the catalyst

Signal flow diagram:

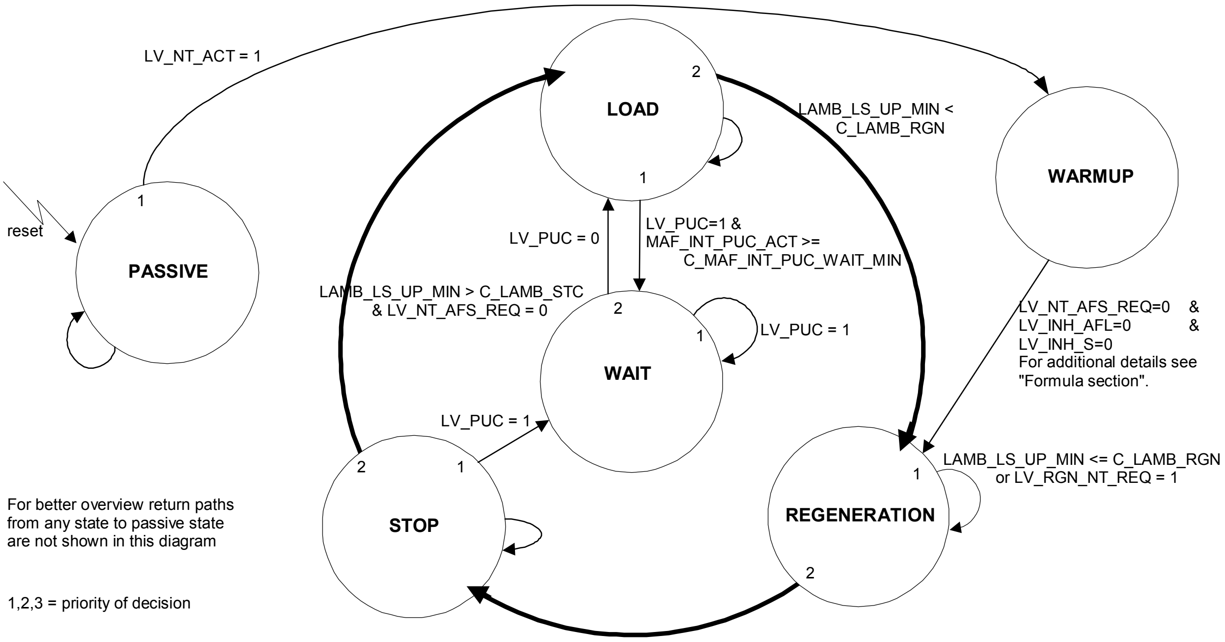


Figure 60: State transition diagram

Application conditions:

Initialisation: *LV_NT_ACT= 0 at reset*

at deactivation:

T_NT_STST_ACT = 0

Recurrence: 10ms

Activation: --

Deactivation: --

Formula section

if(135) LC_NT_STST_ACT = 0

then(135)

```
        LV_NT_ACT =
            LC_NT_ACT = 1           and
            LV_INH_NT_ACT = 0       and             % sensor readiness
            LV_ES = 0               and
            LV_AST = 0              and
            LV_ST = 0
else(135)
        LV_NT_ACT =
            LC_NT_ACT = 1           and
            LV_INH_NT_ACT = 0       and             % sensor readiness
            LV_IGK = 1
endif(135)

if (145)  T_NT_STST_ACT < maximum value (FFFFH)
then (145) increment T_NT_STST_ACT
else (145) % timer remains unchanged on its maximum value
           T_NT_STST_ACT(n) = T_NT_STST_ACT(n-1)
endif (145)
```

53.6.1.1.1 State management

```
% calculation of NTL_MIN was moved to here in order to insure the right calculation order
NTL_MIN = IP_NTL_MIN (TNT_MDL_MV)

Case STATE_NOX = PASSIVE
if (2)      LV_NT_ACT = 1
then (2)    STATE_NOX = WARMUP                      % Exit to WARMUP
else (2)    stay in PASSIVE
endif (2)

Case STATE_NOX = WARMUP
if (3)    LV_NT_ACT = 0
then (3)    STATE_NOX = PASSIVE                     % Exit to PASSIVE
else (3)
    if (3+)   LV_NT_AFS_REQ = 0                                         AND
              ( (OPM_AV <> "AFS" AND LC_STATE_NOX_OPM_AV_ACT = 1) OR
                ( (LV_INH_AFL = 0 OR LC_STATE_NOX_AFL_ACT = 1)    AND
                  (LV_INH_S = 0 OR LC_STATE_NOX_AFL_ACT = 1)
                )
              )
    then (3+)
        if (136)  LV_INH_RGN_REQ = 0  OR  LC_INH_RGN_REQ_ST = 0
        then (136)  STATE_NOX = REGENERATION   % Exit to REGENERATION
```

```
                    if (129)   LC_INH_NT_RGN_ST = 0
                    then (129) LV_RGN_NT_REQ = 1
                    else (129) LV_RGN_NT_REQ = 0
                    endif (129)
          else (136)
                    if (137)   LC_INH_NT_RGN_ST = 0
                    then (137) stay in WARMUP
                    else (137) STATE_NOX = REGENERATION        % Exit
                               LV_RGN_NT_REQ = 0
                    endif (137)
          endif (136)
     else (3+)  stay in WARMUP
     endif (3+)
endif (3)

Case STATE_NOX = LOAD
if (5)     LV_NT_ACT = 0
then (5)   STATE_NOX = PASSIVE                        % Exit to PASSIVE
else (5)
     if (5+)    LV_PUC = 1                                        AND
                MAF_INT_PUC_ACT(n-1) < MAF_INT_PUC_ACT(n)         AND
                MAF_INT_PUC_ACT(n) >= C_MAF_INT_PUC_WAIT_MIN
     then (5+)  STATE_NOX = WAIT                      % Exit to WAIT
     else (5+)  if (5++)     LAMB_LS_UP_MIN < C_LAMB_RGN
                then (5++) call "transition actions load to regeneration"
                           STATE_NOX = REGENERATION  %Exit to REGENERATION
                else (5++)  stay in LOAD
                endif (5++)
     endif (5+)
endif (5)

case STATE_NOX = REGENERATION
if (6)     LV_NT_ACT = 0
then (6)   STATE_NOX = PASSIVE                        % Exit to PASSIVE
else (6)
     if (6+)    (LV_RGN_NT_REQ = 1 or LAMB_LS_UP_MIN <= C_LAMB_RGN)
     then (6+)  STATE_NOX = REGENERATION              %Stay in REGENERATION
     else (6+)  STATE_NOX = STOP                      % Exit to STOP
                SUM_NTL_CYC = SUM_NTL_CYC + 1
     endif (6+)
endif (6)
```

```
case STATE_NOX = WAIT
if (7)      LV_NT_ACT = 0
then (7)    STATE_NOX = PASSIVE                        % Exit to PASSIVE
else (7)
        if (7+)     LV_PUC = 1
        then (7+)   STATE_NOX = WAIT                   % Stay in WAIT
        else (7+)   STATE_NOX = LOAD                   % LAMB_LS_UP > C_LAMB_STC
        endif (7+)
endif (7)

case STATE_NOX = STOP
if (8)      LV_NT_ACT = 0
then (8)    STATE_NOX = PASSIVE                        % Exit to PASSIVE
else (8)
        if (8+)     LV_PUC = 1
        then (8+)   call "transition actions on exiting STOP"
                    STATE_NOX = WAIT                   % Exit to WAIT
        else (8+)
                if (8++)    LAMB_LS_UP_MIN > C_LAMB_STC and
                            ( LV_NT_AFS_REQ = 0 or LC_STATE_NOX_INH_NT_AFS_REQ = 1)
                then (8++)  call "transition actions on exiting STOP"
                            STATE_NOX = LOAD           % Exit to LOAD
                else (8++)  stay in STOP
                endif (8++)
        endif (8+)
endif (8)
```

53.6.1.1.2 Transition actions load to regeneration

```
NTLD_RGN_ST            = NTLD_MDL
NTL_RGN_ST             = NTL / C_FAC_O2_EQU + O2L / C_FAC_COR_O2L
NTL_RGN_ST_NOX         = NTL
NOX_CONC_RED_MMV_RGN_ST = NOX_CONC_RED_MMV
NT_EFF_COR_RGN_ST      = NT_EFF_COR
FLOW_NT_MMV            = MAF_KGH
if (14)   LV_RGN_NT_REQ = 1                    % handling of LV_RGN_AGI
then (14)   LV_RGN_AGI_i = 1
endif (14)
LV_RGN_AGI_VLD_i       = 0                     % final reset
LV_TRIG_AFR_AFL        = 0                     % reset triggerbit
STATE_AFL_PURGE[i]     = DISABLED
MAF_INT_STOP_NTL       = 0
if(141)  LC_AFL_PURGE_RST = 0
then(141)   T_STOP = 0
            MAF_INT_STOP = 0
            CTR_AFL_PURGE[i] = 0
else(141)
            if(142)   NTL > IP_NTL_MIN_AFL_PURGE(VS)              AND
                      NTLD > IP_NTLD_MIN_AFL_PURGE(VS)
            then(142)   T_STOP = 0
                        MAF_INT_STOP = 0
                        CTR_AFL_PURGE[i] = 0
            endif(142)
endif(141)
```

53.6.1.1.3 Transition actions on exiting STOP

```
NOX_OUT_MDL_INT = 0
LV_NOX_SENS_NOX_VLD_RGN_ST_i = 0
LV_T_AFL_MIN = 0
LV_TOUT_AFR = 0
LV_MDL_AFR = 0
FLOW_NT_MMV = MAF_KGH
if (16)   T_NT_HOM > C_T_NT_HOM_MAX_VLD
then (16)   LV_NT_HOM_INI = 0
endif (16)
DIST_AFL_ST = DIST
NOX_COR_INT = 0
STATE_AFL_PURGE[i] = DISABLED
LV_CAT_PURGE_REQ_POST_AFL[i] = 0
```

```
MAF_INT_STOP_NTL = 0
if(143)  LC_AFL_PURGE_RST = 0
then(143)   T_STOP = 0
            MAF_INT_STOP = 0
            CTR_AFL_PURGE[i] = 0
endif(143)
```

Notes:

- For the detection of a regeneration in catalyst purge the calibration data C_LAMB_RGN has to be calibrated always more lean than the calibrated lambda for catalyst purge.

  The following table gives an overview, how variables are calculated depending on the NOx catalyst management state.

state-depending actions

	Passive/ Warmup	RGN	Load	Wait	Stop
NOX_COR_xx	0	0	C	0	0
NOX_CONC_xx	0	0	C	-	-
NT_STC_RNG_H	I	-	C	-	-
NTL	I	C	C	-	C
NTLD	0	C	C	-	C
NTLD_MDL	0	C	C	-	C
NT_EFF	0	0	C	0	0
NT_O2_STC	0	-	C	C	-
O2_FLOW	0	0	C	C	0
O2L	0	C	C	C	-
T_AFL	0	-	C	-	0
T_AFL_PUC	0	0	C	0	0
FLOW_NT_MMV	0	C	C	-	-
NTL_DEC_INT	0	C	0	0	-
Rgn request set	I	/	C	0	0
Purge request set / reset	0	C	C	0	0
Rgn request reset	/	C	/	0	0
AFS request block	0/C	C	C	C	C

I	initialise
0	set to zero
C	calculate
-	freeze value (no changes)
/	action forbidden

53.6.1.2 STATE_NOX = PASSIVE

FUNCTION DESCRIPTION:

General information:

The state PASSIVE is active during engine stop, engine start, after start, warming up and inactivated NOx catalyst function.

The values of the variables listed below are their initialisation values.

If regeneration stop mode is “model”, NTL has to be initialised with a suitable high value in order to prevent the state manager to leave first regeneration (after function start) due to low model load.

Application conditions:

Initialisation: at reset:

```
if (139)    LC_NTL_INI_ST = 1
then (139)    NTL = C_NTL_INI_ST
else (139)    NTL = max(NTL(from NVMY) * C_FAC_NTL_INI_ST,
                            C_NTL_INI_ST)
endif (139)
NTL_H = NTL
```

Recurrence: 20ms

Activation: --

Deactivation: --

Formula section:

LV_TOUT_AFR = 0

LV_MDL_AFR = 0

LV_RGN_AGI_VLD_i = 0

LV_NT_HOM_INI = 0

LV_NOX_SENS_NOX_VLD_RGN_ST_i = 0

LV_T_AFL_MIN = 0

NOX_COR = 0

NOX_COR_RED = 0

NOX_COR_RED_MMV = 0

NOX_CONC_RAW = 0

NOX_CONC = 0

NOX_CONC_MMV = 0

NOX_CONC_RED = 0

NOX_CONC_RED_MMV = 0

NOX_CONC_OUT_MDL_MMV = 0

NOX_CONC_RED_MMV_RGN_ST = 0

NTLD = 0

NTLD_MDL = 0

NT_EFF = 0

NT_EFF_COR_RGN_ST = 1

NT_O2_STC = 0

O2_FLOW = 0

O2L = 0

T_AFL = 0

T_AFL_PUC = 0

NTL_DEC_INT_i = 0

NTL_DEC_INT_SWI_i = 0

NTL_RGN_ST = 0

NTL_RGN_ST_NOX = 0

NOX_OUT_MDL = 0

NOX_OUT_MDL_INT = 0

NOX_NS_AD_RGN_ST[i] = 0

FLOW_NT_MMV = 0

NT_AGI_AV = 1

NTL_DEC_INT_i = 0

LV_RGN_AGI_i = 0

NOX_COR_INT = 0

DIST_AFL_ST = 0

DIST_AFL = 0

RATIO_NOX_OUT_INT_DIST_k = 0

RATIO_NOX_k = 0

LV_RGN_NT_REQ = 0

LV_ACT_INT_PUC_AFL_i = 0

NT_EFF_COR = 1

NT_STC_BAS_RNG_H = C_NT_STC_INI_RNG_H

LV_NT_AFS_REQ = LC_NT_AFS_REQ

NT_STC_RNG_H = C_NT_STC_INI_RNG_H

NTLD_RGN_ST = C_NTLD_RGN_ST_INI

T_NT_HOM = C_T_NT_HOM

LV_CAT_PURGE_REQ_POST_AFL[i] = 0

T_STOP = 0

```
MAF_INT_STOP                                        = 0
MAF_INT_STOP_NTL                                    = 0
CTR_AFL_PURGE[i]                                    = 0
STATE_AFL_PURGE[i]                                  = DISABLED
LV_NT_RGN_STOP_PUC                                  = 0
MAF_INT_PUC_ACT(n-1)                                = MAF_INT_PUC_ACT(n)
if (140)   LC_NTL_INI_ST = 1
then (140)   NTL = C_NTL_INI_ST
             NTL_H = NTL
endif (140)
LV_NTL_DEC_INT_AFR                                  = 0
LV_NTL_DEC_INT_STOP_RGN                             = 0
LV_NT_RGN_2_NOT_VLD                                 = 0
LV_NT_RGN_2_NOT_VLD_SET                             = 0
VLS_NOX_SENS_MMV                                    = -200
```

53.6.1.3 STATE_NOX = WARMUP

Initialisation: --

Recurrence: 100ms

Activation: --

Deactivation: --

Formula section:

```
% see also chapter 1.2.8
call "Homogeneous time counter handling"
T_NT_HOM = min (C_T_NT_HOM, T_NT_HOM(n-1) + 10 * T_NT_HOM_DELTA)
```

53.6.1.4 STATE_NOX = LOAD

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: --

Recurrence: 20ms *(including all subchapters)*

Activation: --

Deactivation: --

General information:

The time counter for lean operation is incremented in STATE_NOX = LOAD

The integral of reducing agent is reset to zero, it is only valid in REGENERATION.

Formula section:

```
if (24)   LV_HOM_AFS_REQ = 0 and (LV_S_ACT = 1 or LV_HOM_AFL_ACT = 1)
then (24)   call "NOx emissions handling"
            call "NOx trap storage capacity and loading degree"
            call "NOx trap efficiency"
            call "Oxygen storage"
    if (25)   LV_NOX_SENS = 1
    then (25)
            if (123)   NC_NT_NR = NC_NOX_SENS_CONF
            then (123)   NOX_NS_AD_RGN_ST[i] = NOX_NS_AD[k]
            else (123)   NOX_NS_AD_RGN_ST[i] = NOX_NS_AD[1]
            endif (123)
    endif (25)
endif (24)

call "Filtered exhaust gas flow"
T_AFL (n)           = T_AFL (n-1) + TA              %TA = recurrence time
T_AFL_PUC (n)       = T_AFL_PUC (n-1) + TA
NTL_DEC_INT_i       = 0
if (26)    T_AFL >= C_T_AFL_NT_HOM_RST
    and    NTLD_MDL >= C_NTLD_NT_HOM_RST
then (26) T_NT_HOM = C_T_NT_HOM
endif (26)
```

53.6.1.4.1 NOx emissions handling

General information:

NOX_RED_FAC takes into account the amount of NOx emission, chemically reduced by the catalyst.

Signal flow diagram:

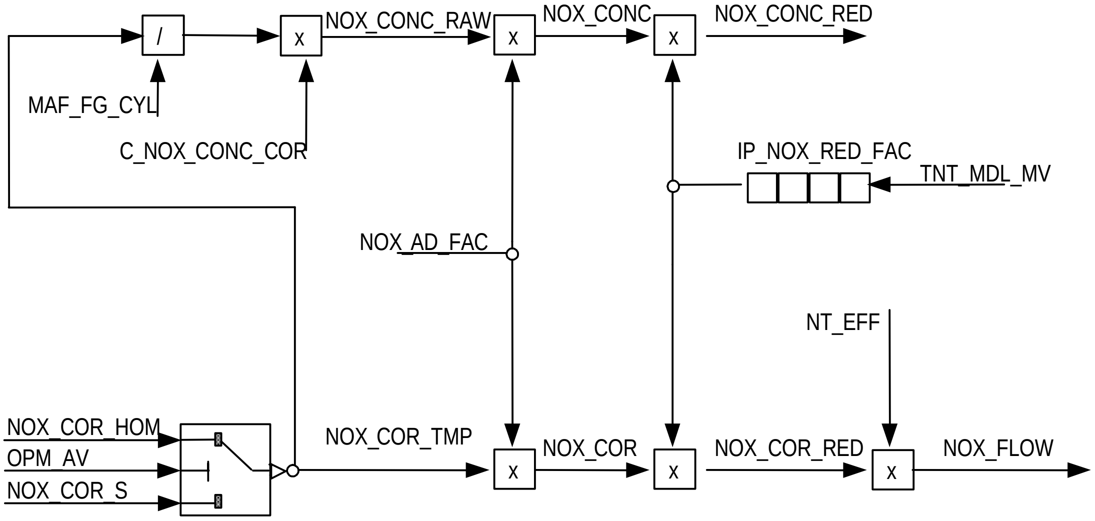


Figure 61: NOx emissions handling

Formula section:

% Functions to calculate NOX_COR_S and NOX_COR_HOMS are calculated independently of this manager. All functions using NOX_COR_S or NOX_COR_HOMS have to be calculated after that.

```
case selection (125) on OPM_AV:
    case OPM_AV = "S":
        NOX_COR_TMP = NOX_COR_S

    case OPM_AV = "AFL" or
    case OPM_AV = "AFS":
        NOX_COR_TMP = NOX_COR_HOMS

    otherwise (default):
        NOX_COR_TMP = NOX_COR_HOMS
end case selection (125)
```

NOX_COR = NOX_COR_TMP * NOX_AD_FAC

NOX_COR_INT(n) = NOX_COR_INT(n-1) + NOX_COR(n) * recurrence

NOX_COR_RED = NOX_COR * IP_NOX_RED_FAC (TNT_MDL_MV)

NOX_COR_RED_MMV$_n$ = NOX_COR_RED * C_CRLC_NOX_COR_RED

+ NOX_COR_RED_MMV$_{n-1}$ * (1 - C_CRLC_NOX_COR_RED)
NOX_FLOW = NOX_COR_RED * NT_EFF

NOX_CONC_RAW = (NOX_COR_TMP / MAF_FG_CYL) * C_NOX_CONC_COR
NOX_CONC = NOX_CONC_RAW * NOX_AD_FAC
NOX_CONC_RED = NOX_CONC * IP_NOX_RED_FAC (TNT_MDL_MV)
NOX_CONC_MMV$_n$ = NOX_CONC * C_CRLC_NOX_CONC_RED
+ NOX_CONC_MMV$_{n-1}$ * (1 - C_CRLC_NOX_CONC_RED)
NOX_CONC_RED_MMV$_n$ = NOX_CONC_RED * C_CRLC_NOX_CONC_RED
+ NOX_CONC_RED_MMV$_{n-1}$ * (1 - C_CRLC_NOX_CONC_RED)

NOX_OUT_MDL = NOX_COR_RED * (1- NT_EFF)
NOX_CONC_OUT_MDL = NOX_CONC_RED * (1- NT_EFF)
NOX_CONC_OUT_MDL_MMV$_n$ = NOX_CONC_OUT_MDL *
C_CRLC_NOX_CONC_OUT_MDL + NOX_CONC_OUT_MDL_MMV$_{n-1}$ *
(1 - C_CRLC_NOX_CONC_OUT_MDL)
NOX_OUT_MDL_INT= NOX_OUT_MDL_INT$_{n-1}$ + NOX_OUT_MDL
% NOX_OUT_MDL_INT has to be internally treated as a longint.

The factor C_NOX_CONC_COR consists of the following chemical constants:

C_NOX_CONC_COR = molmass air / molmass NO2 * (1/factor for transformation “kg/h--> kg/s”) = 29 /46 * 3600 = 2269.565

This value should be the initialization value of the calibration data.

Due to possible small values of NOX_OUT_MDL, NOX_OUT_MDL_INT has to be internally treated as a longint.

53.6.1.4.2 NOx trap storage capacity and loading degree

General information:

The basic catalyst storage capacity is worked out by a map depending on the raw NOx concentration NOX_CONC and on the modelled NOx-trap temperature mean value. The aging factor NT_AGI_NTLD is delivered from the module NOx catalyst aging and reduces the basic storage capacity with the present aging state depending on catalyst temperature.

The current load of the NOx storage catalyst is worked out by the NOx mass flow actually available to be stored in the catalyst NOX_FLOW and the NOx trap efficiency NT_EFF.

The NOx trap loading degree is the ratio of the NOx trap absolute loading and the NOx storage capacity. The increase of NTLD is limited to C_NTLD_LGRD per time step.

Signal flow diagram:

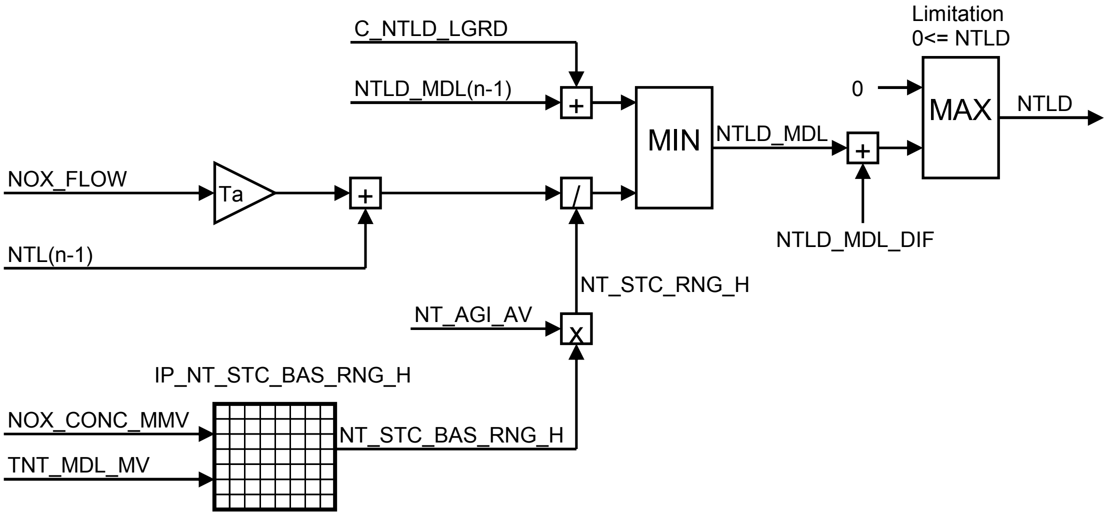


Figure 62: NOx catalyst storage capacity, absolute loading and loading degree

Formula section:

NT_STC_BAS_RNG_H = IP_NT_STC_BAS_RNG_H (NOX_CONC_MMV, TNT_MDL_MV))

NT_AGI_AV = IP_NT_AGI_RGN_FRQ (NT_AGI_NTLD, TNT_MDL_MV)

NT_STC_RNG_H = NT_STC_BAS_RNG_H * NT_AGI_AV

$NTL_n = NTL_{n-1} + NOX\_FLOW * Ta$

```
if (164)    TNT_MDL_MV >= (C_TNT_MDL_MV_NTL_H_RST_MIN +
                               C_TNT_MDL_MV_NTL_H_RST_HYS)        AND
            TNT_MDL_MV <= (C_TNT_MDL_MV_NTL_H_RST_MAX -
                               C_TNT_MDL_MV_NTL_H_RST_HYS)
then (164)    $NTL\_H_n = NTL\_H_{n-1} + NOX\_FLOW * Ta$
else (164)
        if (165)   LC_NTL_H_RST_LOAD_TNT = 1
        then (165)   NTL_H is reset
                     $NTL\_H_n = NTL\_MIN$
        else (165)   NTL_H is frozen
                     $NTL\_H_n = NTL\_H_{n-1}$
        endif (165)
endif (164)
```

Attention!
- Observe different resolutions and value ranges of NTL, NTL_H and NTL_MIN.
- NTL and NTL_H should be internally calculated as longint.

$NTLD\_MDL = \min(NTL / NT\_STC\_RNG\_H, NTLD\_MDL_{(n-1)} + C\_NTLD\_LGRD)$

NTLD = **max** (0, NTLD_MDL + NTLD_MDL_DIF)

53.6.1.4.3 NOx trap efficiency

General information:

The base NOx trap efficiency is given by a map depending on NOx trap loading degree and the modelled NOx trap temperature. The base value is corrected depending on the raw NOx mass flow NOX_COR.

Note: The value NT_EFF is limited to 1 in the program.

Signal flow diagram:

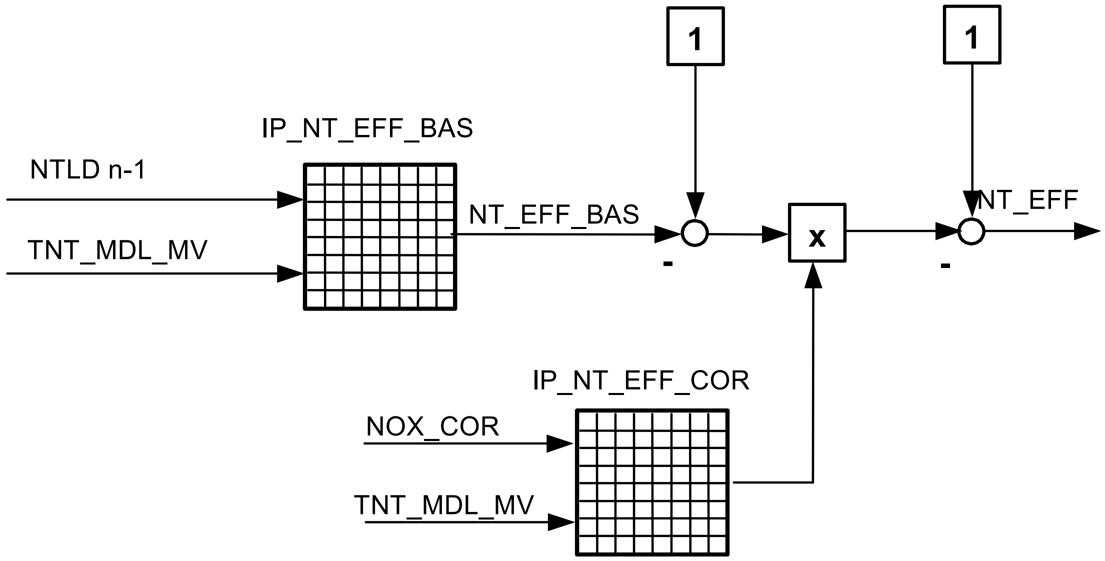


Figure 63: NOx trap efficiency (simplified)

Formula section:

if (30) LV_EF = 0

then (30) NT_EFF_BAS = IP_NT_EFF_BAS (NTLD, TNT_MDL_MV)

else (30) NT_EFF_BAS = IP_NT_EFF_BAS_EF_CLOSE (NTLD, TNT_MDL_MV)

% not shown in the figure

NT_EFF_COR = IP_NT_EFF_COR (NOX_COR, TNT_MDL_MV)

NT_EFF = 1 - ((1 - NT_EFF_BAS) * NT_EFF_COR)

53.6.1.4.4 Oxygen storage

General information:

During lean operation the NOx trap stores Oxygen. The current load of stored Oxygen in the NOx trap, O2L is worked out by a formula.

Signal flow diagram:

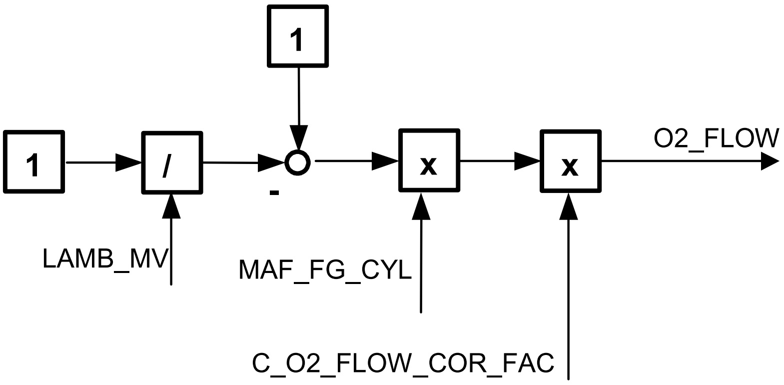


Figure 64: Oxygen mass flow

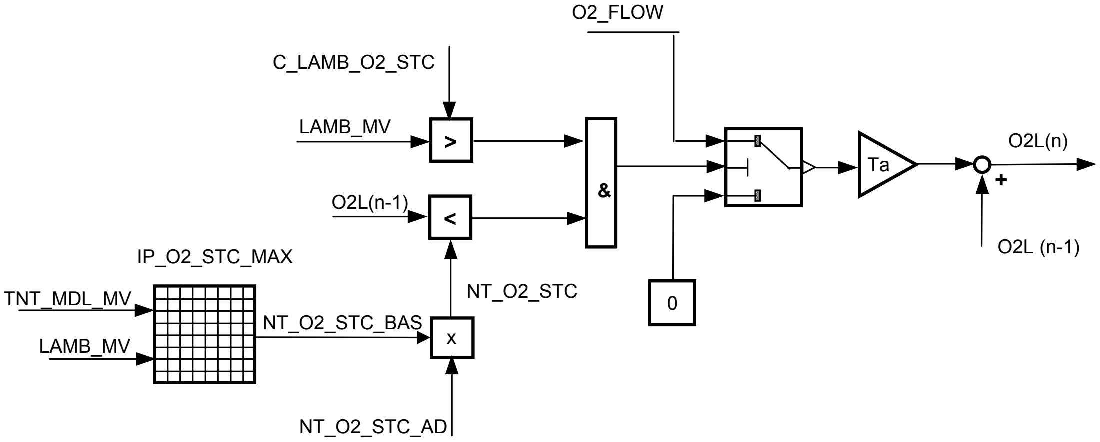


Figure 65: Oxygen storage

Formula section:

NT_O2_STC_BAS= IP_O2_STC_MAX (LAMB_LS_UP_MV, TNT_MDL_MV)

NT_O2_STC = NT_O2_STC_BAS * NT_O2_STC_AD

if (32) (LAMB_LS_UP_MV > C_LAMB_O2_STC)

and LAMB_LS_UP_MV > 1

and (O2L < NT_O2_STC)

then (32) O2_FLOW = (1 - (1/ LAMB_LS_UP_MV)) * MAF_FG_CYL * C_O2_FLOW_COR_FAC

O2L $_{(n)}$ = O2L (n-1) + O2_FLOW * Ta

else (32) O2L $_{(n)}$ = O2L (n-1)

endif (32)

The factor C_O2_FLOW_COR_FAC consists of the following constants:

C_O2_FLOW_COR_FAC = (0.21* molmass(O2)) / molmass(air) * factor for transformation of MAF_FG_CYL (kg/h) to (mg/s)) = 0.21 * 32 / 29 * 10^6 / 3600 = 64.37

This value should be the initialization value of the calibration data.

53.6.1.5 STATE_NOX = WAIT

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: --

Recurrence: 10ms

Activation: --

Deactivation: --

Formula section:

```
if (144)   MAF_INT_PUC_ACT(n-1) < MAF_INT_PUC_ACT(n)              AND
           MAF_INT_PUC_ACT(n) >= C_MAF_INT_PUC_RST_RGN_REQ
then (144) LV_RGN_NT_REQ = 0
endif (144)
```

Application conditions:

Initialisation: --

Recurrence: 20ms *(including all subchapters)*

Activation: --

Deactivation: --

53.6.1.5.1 General, NOx storage

General information:

When the engine state LV_PUC is active (-> STATE_NOX = WAIT), the NOx trap loading calculation is frozen.

Formula section:

NOX_COR = 0
NOX_COR_RED = 0
NOX_COR_RED_MMV = 0
NOX_CONC_RAW = 0
NOX_CONC = 0
NOX_CONC_RED = 0
NOX_CONC_MMV = NOX_CONC_MMV $_{n-1}$
NOX_CONC_RED_MMV = NOX_CONC_RED_MMV $_{n-1}$

NOX_CONC_OUT_MDL_MMV = NOX_CONC_OUT_MDL_MMV $_{n-1}$

NOX_OUT_MDL = 0

NT_STC_RNG_H = NT_STC_RNG_H(n-1)

NTLD = NTLD (n-1)

NTLD_MDL = NTLD_MDL (n-1)

NT_EFF = 0

T_AFL = T_AFL (n-1)

T_AFL_PUC = 0

NTL_DEC_INT_i = 0

T_NT_HOM = T_NT_HOM (n-1)

LV_CAT_PURGE_REQ_POST_AFL[i] = 0

NTL = NTL$_{n-1}$

```
if (161)    ( MAF_INT_PUC_ACT(n-1) < MAF_INT_PUC_ACT(n)     AND
              MAF_INT_PUC_ACT(n) > C_MAF_INT_PUC_NTL_H_RST )          OR
            TNT_MDL_MV < C_TNT_MDL_MV_NTL_H_RST_MIN                   OR
            TNT_MDL_MV > C_TNT_MDL_MV_NTL_H_RST_MAX
then (161)  NTL_H is reset
            NTL_H = NTL_MIN
else (161)  NTL_H is frozen
            NTL_H = NTL_H$_{n-1}$
endif (161)
```

LV_ACT_INT_PUC_AFL_i = 0

Evaluation of slope of MAF_INT_PUC_ACT moved from 20ms to 10ms.

53.6.1.5.2 Oxygen mass flow and oxygen storage

General information:

During the engine state LV_PUC the NOx catalyst stores oxygen. The current load of stored oxygen in the NOx trap is worked out by a formula.

The calculation of the following quantities is done in the same way as in STATE_NOX = LOAD.

(see subchapter “STATE_NOX = LOAD”, passage “oxygen storage”) (3 pages above)

Formula section:

call "Oxygen storage"

53.6.1.6 STATE_NOX = REGENERATION

FUNCTION DESCRIPTION:

General information:

For the regeneration of the NOx trap a rich mixture is required.

Application conditions:

Initialisation: --

Recurrence: 10ms *(including all subchapters)*

Activation: --

Deactivation: --

53.6.1.6.1 Miscellaneous

Formula section:

NOX_COR = 0
NOX_COR_RED = 0
NOX_COR_RED_MMV = 0
NOX_CONC_RAW = 0
NOX_CONC = 0
NOX_CONC_RED = 0
NOX_CONC_MMV = 0
NOX_CONC_RED_MMV = 0
NOX_CONC_OUT_MDL_MMV = 0
NOX_OUT_MDL = 0
NT_STC_RNG_H = $\text{NT\_STC\_RNG\_H}_{(n-1)}$
NT_EFF = 0
NT_O2_STC = $\text{NT\_O2\_STC}_{(n-1)}$
O2_FLOW = 0
T_AFL = $\text{T\_AFL}_{(n-1)}$
T_AFL_PUC = 0
call " Filtered exhaust gas flow "
call "Homogeneous time counter handling"
T_NT_HOM = **min** (C_T_NT_HOM, $\text{T\_NT\_HOM}_{(n-1)}$ + T_NT_HOM_DELTA)

53.6.1.6.2 Handling of trigger bits for functions using RAI signal

Formula section:

% undefined exhaust gas during regeneration

if (38) LV_SCC[m] = 1

```
        or      LV_PUC = 1
        or      (C_STATE_SENS_AFR_i = "VLS_NOX_SENS" and LV_NOX_SENS_VLS_VLD_k = 0)
        or      (C_STATE_SENS_AFR_i = "LAMB_NOX_SENS" and
                                                        LV_NOX_SENS_LAMB_VLD_k = 0)
then (38) LV_RGN_AGI_i = 0
endif (38)


% sensor switches during valid regeneration
if (39)     LV_SENS_AFR_i = 1 and LV_RGN_AGI_i = 1
then (39)   LV_RGN_AGI_i = 0
            LV_RGN_AGI_VLD_i = 1
endif (39)
```

53.6.1.6.3 Calculation of NTL_DEC and NTL_DEC_INT

General information:

For the regeneration of the NOx trap a rich mixture is required.

NTL_DEC_i represents the „lack of oxygen“ during regeneration. This lack of oxygen corresponds to the reducing agent mass. The unit of NTL_DEC therefore is [mg O2 / time step].

NTL_DEC_INT_i represents the integral of the NTL_DEC_i during regeneration. It is calculated during the regeneration phase.

NTL_DEC_INT_SWI represents the NTL_DEC_INT up to the moment the downstream sensor switches from lean to rich. This value may be used for several adaptation purposes.

Due to possible small values of NTL_DEC (defined as a longint), NTL_DEC_INT has to be internally treated as a longint.

Formula section:

```
if (155)   NC_NT_NR = 2
then (155) Twin branch system with 2 TWC and 2 NOx catalysts NC_CBK_EX_NR =
           NC_NT_NR = 2  =>  i = m = 2
    if (147)   LAMB_LS_UP[m]  < 1.0                                AND
               VLS_DOWN[m] >= C_VLS_DOWN_MIN_NTL_DEC[m]
    then (147) NTL_DEC[i] = ((1 / LAMB_LS_UP[m]) - 1) * (MAF_FG_CYL /
               NC_CBK_EX_NR)  * C_O2_FLOW_COR_FAC * Recurrence
               NTL_DEC_INT_i = NTL_DEC_INT_i (n-1) + NTL_DEC[i]
    else (147) NTL_DEC[i] = 0
               NTL_DEC_INT_i = NTL_DEC_INT_i (n-1)
    endif (147)
```

```
if (148)    LC_SENS_AFR_MOD = 0
then (148)
        if (40)      LV_SENS_AFR[i] = 0
        then (40)    NTL_DEC_INT_SWI[i] = NTL_DEC_INT_i
        endif (40)
else (148)
    case selection on STATE_RGN[1]:
        "H_AMPL":
            if (149)     LV_SENS_AFR[1] = 0
            then (149)   NTL_DEC_INT_SWI[1] = NTL_DEC_INT_1
            endif (149)

        "L_AMPL":
            if (150)     LV_SENS_AFR[2] = 0
            then (150)   NTL_DEC_INT_SWI[1] = NTL_DEC_INT_1
            endif (150)

        "PARK":                 ---
        "PASSIVE" (default):    ---
    end case selection

    case selection on STATE_RGN[2]:
        "H_AMPL":
            if (151)     LV_SENS_AFR[1] = 0
            then (151)   NTL_DEC_INT_SWI[2] = NTL_DEC_INT_2
            endif (151)

        "L_AMPL":
            if (152)     LV_SENS_AFR[2] = 0
            then (152)   NTL_DEC_INT_SWI[2] = NTL_DEC_INT_2
            endif (152)

        "PARK":                 ---
        "PASSIVE" (default):    ---
    end case selection
endif (148)
```

```
        NTL_DEC_SUM = NTL_DEC[1] + NTL_DEC[2]


    else (155)  Single branch or Y system with 1 or 2 TWC and 1 NOx catalyst
                NC_CBK_EX_NR = m = 1,2; NC_NT_NR = i = 1

        if (156)   NC_CBK_EX_NR = 2

        then (156)   Y system with 2 TWC and 1 NOx catalyst
                     NC_CBK_EX_NR = m = 2; NC_NT_NR = i = 1

            if (157)   LAMB_LS_UP_MV  < 1.0                          AND
                       VLS_DOWN[1] >= C_VLS_DOWN_MIN_NTL_DEC[1]  AND
                       VLS_DOWN[2] >= C_VLS_DOWN_MIN_NTL_DEC[2]

            then (157)   NTL_DEC[1] = ((1 / LAMB_LS_UP_MV) - 1) * (MAF_FG_CYL /
                         NC_CBK_EX_NR)  * C_O2_FLOW_COR_FAC * Recurrence

                         NTL_DEC_INT_1 = NTL_DEC_INT_1 (n-1) + NTL_DEC[1]

            else (157)   NTL_DEC[1] = 0

                         NTL_DEC_INT_1 = NTL_DEC_INT_1 (n-1)

            endif (157)


        else (156)   Single branch system with 1 TWC and 1 NOx catalyst
                     NC_CBK_EX_NR = 1; NC_NT_NR = 1  =>  m = i = 1

            if (158)   LAMB_LS_UP[m]  < 1.0                          AND
                       VLS_DOWN[m] >= C_VLS_DOWN_MIN_NTL_DEC[m]

            then (158)   NTL_DEC[1] = ((1 / LAMB_LS_UP[1]) - 1) * (MAF_FG_CYL /
                         NC_CBK_EX_NR)  * C_O2_FLOW_COR_FAC * Recurrence

                         NTL_DEC_INT_1 = NTL_DEC_INT_1 (n-1) + NTL_DEC[1]

            else (158)   NTL_DEC[1] = 0

                         NTL_DEC_INT_1 = NTL_DEC_INT_1 (n-1)

            endif (158)


        endif (156)


        if (154)   LV_SENS_AFR[1] = 0

        then (154)   NTL_DEC_INT_SWI[1] = NTL_DEC_INT_1

        endif (154)


        NTL_DEC_SUM = NTL_DEC[1]


    endif (155)
```

The factor C_O2_FLOW_COR_FAC consists of the following constants:

C_O2_FLOW_COR_FAC = (0.21* molmass(O2)) / molmass(air) * factor for transformation of MAF_FG_CYL (kg/h) to (mg/s)) = 0.21 * 32 / 29 * 10^6 / 3600 = 64.37

This value should be the initialization value of the calibration data.

NTL_DEC_INT has to be internally treated as a longint.

53.6.1.6.4 Decrementation of the NOx trap absolute loading of NOx and O2

General information:

During the regeneration phase the reducing agent is used for both, the reduction of the stored NOx and the stored O2. This is taken into account by using a share factor FAC_RGN_DEC_SHAR for the decrement of the NTL and the O2L. This share factor is calculated using the ratio of stored O2 to the sum of stored O2 and NOx.

C_FAC_O2_EQU and C_FAC_COR_O2L are based on the chemical equation of the regeneration phase.

- C_FAC_O2_EQU should be calibrated with 1.25 to be neutral.
- C_FAC_COR_O2L should be calibrated with 1 to be neutral.

The constant C_O2_FLOW_COR_FAC and is also defined in „oxygen storage“ subchapter of STATE_NOX=LOAD chapter.

Signal flow diagram:

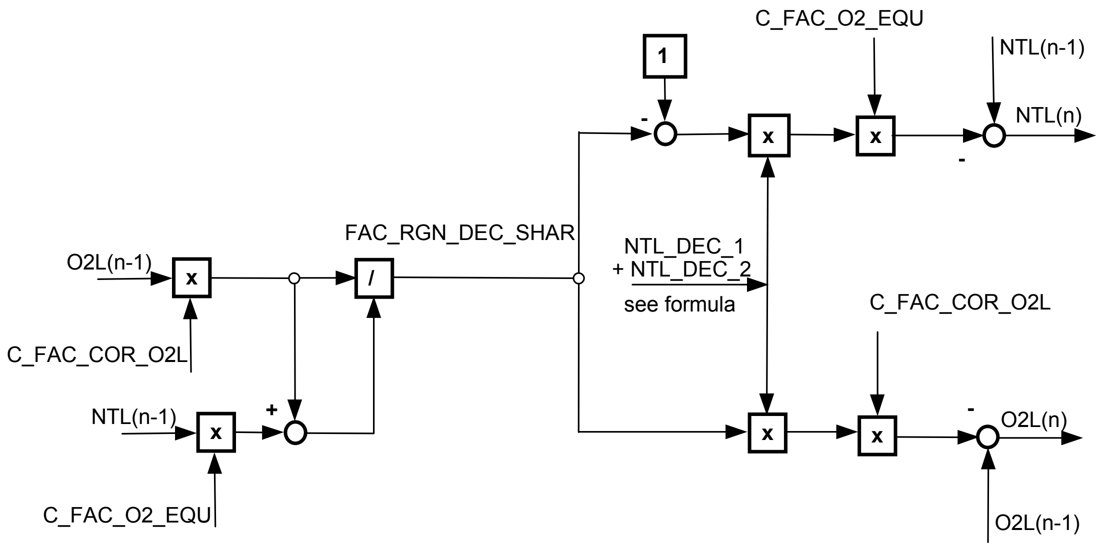


Figure 66: NOx trap loading decrement during regeneration phase

Formula section:

$$FAC\_RGN\_DEC\_SHAR = O2L_{(n-1)} * C\_FAC\_COR\_O2L / (O2L_{(n-1)} * C\_FAC\_COR\_O2L + NTL_{(n-1)} * C\_FAC\_O2\_EQU)$$

$$NTL_n = \mathbf{max}\ [NTL\_MIN, NTL_{n-1} - (1\text{-}FAC\_RGN\_DEC\_SHAR) * NTL\_DEC\_SUM * C\_FAC\_O2\_EQU]$$

$$NTL\_H_n = \mathbf{max}\ [NTL\_MIN, NTL\_H_{n-1} - (1\text{-}FAC\_RGN\_DEC\_SHAR) * NTL\_DEC\_SUM * C\_FAC\_O2\_EQU]$$

Attention!
- Observe different resolutions and value ranges of NTL, NTL_H and NTL_MIN.
- NTL and NTL_H should be internally calculated as longint.

$$O2L_{(n)} = \mathbf{max}\ (0, O2L_{(n-1)} - FAC\_RGN\_DEC\_SHAR * NTL\_DEC\_SUM * C\_FAC\_COR\_O2L)$$

$$NTLD\_MDL = NTL / NT\_STC\_RNG\_H$$

$$NTLD = max (0, NTLD\_MDL + NTLD\_MDL\_DIF)$$

53.6.1.7 STATE_NOX = STOP

FUNCTION DESCRIPTION:

General information:

When STATE_NOX remains in STOP (e.g. due to Lambda=1-engine operation), NTL_MIN can change due to trap temperature changes.

When the system remains in STATE_NOX = STOP for a longer time after a lean operation phase then the catalyst purge functionality is activated. The catalyst purge functionality is controlled by an own state machine.

Signal flow diagram:

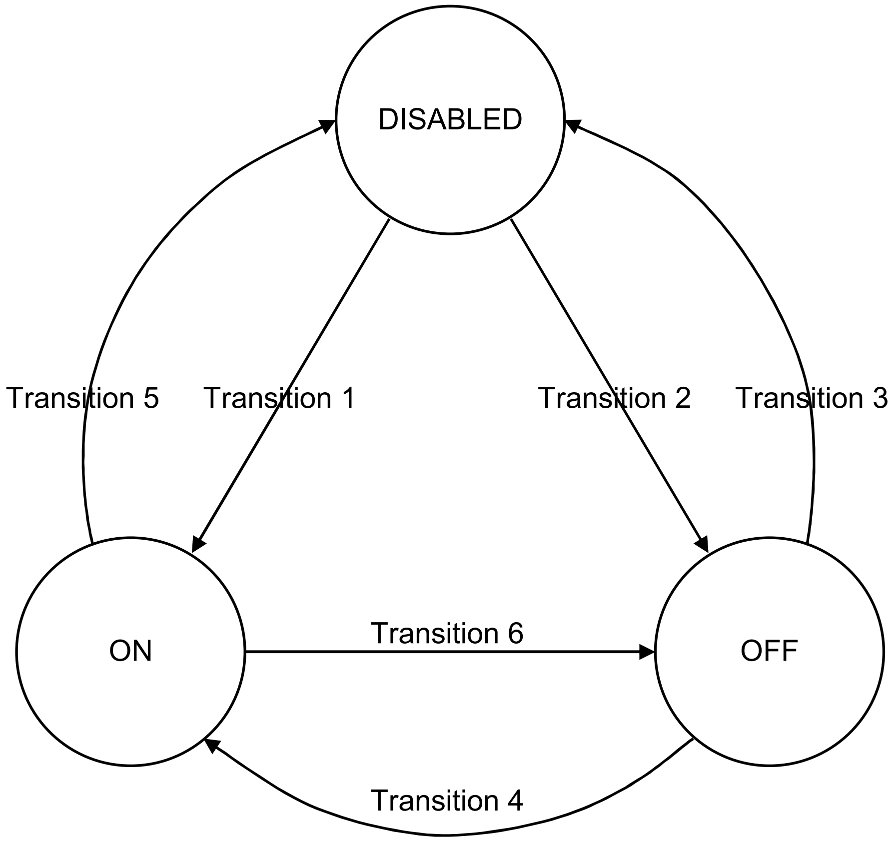


Figure 67: State diagram for activation and deactivation of catalyst purge after lean operation

Application conditions:

Initialisation: --

Recurrence: 20ms

Activation: --

Deactivation: --

Formula section:

NOX_COR = 0
NOX_COR_RED = 0
NOX_COR_RED_MMV = 0
NOX_CONC_RAW = 0
NOX_CONC = 0
NOX_CONC_RED = 0
NOX_OUT_MDL = 0
NT_STC_RNG_H = $NT\_STC\_RNG\_H_{(n-1)}$
NT_EFF = 0
NT_O2_STC = $NT\_O2\_STC_{(n-1)}$

```
O2_FLOW                                  = 0
NTL_DEC_INT_i                            = NTL_DEC_INT_i $_{(n-1)}$
NTLD_MDL                                 = NTL / NT_STC_RNG_H
NTLD                                     = max (0, NTLD_MDL + NTLD_MDL_DIF)
call "Homogeneous time counter handling"
T_NT_HOM = min (C_T_NT_HOM, T_NT_HOM$_{(n-1)}$ + 2 * T_NT_HOM_DELTA)
LV_ACT_INT_PUC_AFL_i                     = 0     % purge with low priority, final reset
LV_RGN_AGI_i                             = 0     % final reset

if (146)   T_NT_STST_ACT >= C_T_AFL_RST
then (146) T_AFL = 0
else (146) % T_AFL remains unchanged
           T_AFL(n) = T_AFL(n-1)
endif (146)

if (162)   MAF_INT_STOP_NTL < maximum value (FFFFH)
then (162) MAF_INT_STOP_NTL(n) = MAF_INT_STOP_NTL(n-1) + MAF_KGH * Recurrence
else (162) % integral remains unchanged on its maximum value
           MAF_INT_STOP_NTL(n) = MAF_INT_STOP_NTL(n-1)
endif (162)

if (45)    LV_SENS_AFR[i] = 1                % valid for all values of i
then (45)  NTL = MIN (NTL $_{(n-1)}$, NTL_MIN)
           NTL_H = MIN (NTL_H $_{(n-1)}$, NTL_MIN)
           O2L =  0
else (45)  NTL =  NTL $_{(n-1)}$
           O2L =  O2L$_{(n-1)}$
           if (163)   MAF_INT_STOP_NTL > C_MAF_INT_STOP_NTL_H_RST      OR
                      TNT_MDL_MV < C_TNT_MDL_MV_NTL_H_RST_MIN          OR
                      TNT_MDL_MV > C_TNT_MDL_MV_NTL_H_RST_MAX
           then (163) NTL_H is reset
                      NTL_H = NTL_MIN
           else (163) NTL_H is frozen
                      NTL_H = NTL_H$_{n-1}$
           endif (163)
endif (45)

if (91)    T_STOP < maximum value (FFFFH)
then (91)  increment T_STOP
```

```
else (91)    % timer remains unchanged on its maximum value
             T_STOP(n) = T_STOP(n-1)
endif (91)


if (92)      MAF_INT_STOP < maximum value (FFFFH)
then (92)    MAF_INT_STOP(n) = MAF_INT_STOP(n-1) + MAF_KGH * Recurrence
else (92)    % integral remains unchanged on its maximum value
             MAF_INT_STOP(n) = MAF_INT_STOP(n-1)
endif (92)


% State machine for set and reset of LV_CAT_PURGE_REQ_POST_AFL[i]
% The priorities of the conditions to change between states shall be defined by the order in
  which these conditions are listed within the appropriate state as described below.
% If a transition condition is fulfilled, then the following transition conditions are not
  calculated.

STATE_AFL_PURGE[i] = "DISABLED" (default state):

Transition 1:        "DISABLED" to "ON"

if (93)     LV_CAT_PURGE_REQ_POST_AFL[i] = 1                          AND
            TNT_MDL_MV > C_TNT_MDL_PURGE_MIN                          AND
            TNT_MDL_MV < C_TNT_MDL_PURGE_MAX                          AND
            VS < C_VS_PURGE_MAX                                       AND
            LV_NS_AD_REQ = 0                                          AND
            (LV_NS_SHIFT_CMB_INT_REQ = 0 OR
                     LC_NS_SHIFT_INH_AFL_PURGE = 0)                   AND
            (LV_VLS_NS_VLD[k] = 1 OR LC_VLS_NS_VLD_AFL_PURGE = 0)
then (93)   STATE_AFL_PURGE[i] = "ON"
            CTR_AFL_PURGE[i](n) = CTR_AFL_PURGE[i](n-1) + 1
else (93)


Transition 2:        "DISABLED" to "OFF"

if (94)     T_STOP <= C_T_STOP_AFL_PURGE_MAX                          AND
            MAF_INT_STOP <= C_MAF_INT_STOP_AFL_PURGE_MAX              AND
            CTR_AFL_PURGE[i] < C_CTR_AFL_PURGE_MAX                    AND
            TNT_MDL_MV > C_TNT_MDL_PURGE_MIN                          AND
            TNT_MDL_MV < C_TNT_MDL_PURGE_MAX                          AND
            VS < C_VS_PURGE_MAX                                       AND
            LV_NS_AD_REQ = 0                                          AND
            (LV_NS_SHIFT_CMB_INT_REQ = 0 OR
                     LC_NS_SHIFT_INH_AFL_PURGE = 0)                   AND
            (LV_VLS_NS_VLD[k] = 1 OR LC_VLS_NS_VLD_AFL_PURGE = 0)
then (94)   STATE_AFL_PURGE[i] = "OFF"
else (94)
```

```
Steady state actions:
LV_CAT_PURGE_REQ_POST_AFL[i] = 0

endif (94)
endif (93)


STATE_AFL_PURGE[i] = "OFF":
Transition 3:       "OFF" to "DISABLED"
if (95)   T_STOP > C_T_STOP_AFL_PURGE_MAX                             OR
          MAF_INT_STOP > C_MAF_INT_STOP_AFL_PURGE_MAX                 OR
          CTR_AFL_PURGE[i] >= C_CTR_AFL_PURGE_MAX                     OR
          TNT_MDL_MV < C_TNT_MDL_PURGE_DEAC                           OR
          TNT_MDL_MV > C_TNT_MDL_PURGE_MAX                            OR
          VS > C_VS_PURGE_MAX                                         OR
          LV_NS_AD_REQ = 1                                            OR
          (LV_NS_SHIFT_CMB_INT_REQ = 1 AND
                   LC_NS_SHIFT_INH_AFL_PURGE = 1)                     OR
          (LV_VLS_NS_VLD[k] = 0 AND LC_VLS_NS_VLD_AFL_PURGE = 1)
then (95)   STATE_AFL_PURGE[i] = "DISABLED"
            LV_CAT_PURGE_REQ_POST_AFL[i] = 0
else (95)


Transition 4:       "OFF" to "ON"
if (96)   NC_NT_NR = 2  AND  LC_SENS_AFR_MOD = 1
          % NC_NT_NR = 2 means 2 NOx catalysts are within the exhaust system
          % If NC_NT_NR = 1 or the case above then the calculation of branch 1 is sufficient.
            After that the branch 1 results are copied to branch 2.
then (96)
      case selection on C_STATE_SENS_AFL_PURGE[1]:
          case "VLS_DOWN":
                if (97)   VLS_DOWN[1] < C_VLS_DOWN_AFL_PURGE_ON
                then (97)   STATE_AFL_PURGE[1] = "ON"
                            LV_CAT_PURGE_REQ_POST_AFL[1] = 1
                            CTR_AFL_PURGE[1](n) = CTR_AFL_PURGE[1](n-1) + 1
                endif (97)


          case "VLS_NOX_SENS":
                if (98)   VLS_NOX_SENS[1] < C_VLS_NOX_SENS_AFL_PURGE_ON
```

```
            then (98)   STATE_AFL_PURGE[1] = "ON"
                        LV_CAT_PURGE_REQ_POST_AFL[1] = 1
                        CTR_AFL_PURGE[1](n) = CTR_AFL_PURGE[1](n-1) + 1
            endif (98)


        case "LAMB_NOX_SENS":
            if (99)     LAMB_NOX_SENS[1] > C_LAMB_NOX_SENS_AFL_PURGE_ON
            then (99)   STATE_AFL_PURGE[1] = "ON"
                        LV_CAT_PURGE_REQ_POST_AFL[1] = 1
                        CTR_AFL_PURGE[1](n) = CTR_AFL_PURGE[1](n-1) + 1
            endif (99)


        case "VLS_NT_DOWN":
            if (100)    VLS_NT_DOWN[1] < C_VLS_DOWN_AFL_PURGE_ON
            then (100)  STATE_AFL_PURGE[1] = "ON"
                        LV_CAT_PURGE_REQ_POST_AFL[1] = 1
                        CTR_AFL_PURGE[1](n) = CTR_AFL_PURGE[1](n-1) + 1
            endif (100)


        case "NONE":
            ---
    end case selection


    STATE_AFL_PURGE[2] = STATE_AFL_PURGE[1]
    LV_CAT_PURGE_REQ_POST_AFL[2] = LV_CAT_PURGE_REQ_POST_AFL[1]
    CTR_AFL_PURGE[2] = CTR_AFL_PURGE[1]


else (96)
    % Full 2 branch calculation.
    case selection on C_STATE_SENS_AFL_PURGE[i]:
        case "VLS_DOWN":
            if (101)    VLS_DOWN[m] < C_VLS_DOWN_AFL_PURGE_ON
            then (101)  STATE_AFL_PURGE[i] = "ON"
                        LV_CAT_PURGE_REQ_POST_AFL[i] = 1
                        CTR_AFL_PURGE[i](n) = CTR_AFL_PURGE[i](n-1) + 1
            endif (101)


        case "VLS_NOX_SENS":
```

```
                    if (102)  VLS_NOX_SENS[k] < C_VLS_NOX_SENS_AFL_PURGE_ON
                    then (102) STATE_AFL_PURGE[i] = "ON"
                               LV_CAT_PURGE_REQ_POST_AFL[i] = 1
                               CTR_AFL_PURGE[i](n) = CTR_AFL_PURGE[i](n-1) + 1
                    endif (102)


                case "LAMB_NOX_SENS":
                    if (103)  LAMB_NOX_SENS[k] > C_LAMB_NOX_SENS_AFL_PURGE_ON
                    then (103) STATE_AFL_PURGE[i] = "ON"
                               LV_CAT_PURGE_REQ_POST_AFL[i] = 1
                               CTR_AFL_PURGE[i](n) = CTR_AFL_PURGE[i](n-1) + 1
                    endif (103)


                case "VLS_NT_DOWN":
                    if (104)  VLS_NT_DOWN[k] < C_VLS_DOWN_AFL_PURGE_ON
                    then (104) STATE_AFL_PURGE[i] = "ON"
                               LV_CAT_PURGE_REQ_POST_AFL[i] = 1
                               CTR_AFL_PURGE[i](n) = CTR_AFL_PURGE[i](n-1) + 1
                    endif (104)


                case "NONE":
                    ---
            end case selection
        endif (96)


        Steady state actions:
        ---


        endif (95)



        STATE_AFL_PURGE[i] = "ON":
        Transition 5:          "ON" to "DISABLED"
        if (105) TNT_MDL_MV < C_TNT_MDL_PURGE_DEAC                                OR
                 TNT_MDL_MV > C_TNT_MDL_PURGE_MAX                                 OR
                 VS > C_VS_PURGE_MAX                                              OR
                 LV_NS_AD_REQ = 1                                                 OR
                 ( LV_NS_SHIFT_CMB_INT_REQ = 1                              AND
                     LC_NS_SHIFT_INH_AFL_PURGE = 1 )                              OR
                 ( LV_VLS_NS_VLD[k] = 0 AND LC_VLS_NS_VLD_AFL_PURGE = 1 )         OR
```

```
            ( LC_INH_AFL_PURGE_TOUT = 0                              AND
              ( T_STOP > C_T_STOP_AFL_PURGE_MAX                     OR
                MAF_INT_STOP > C_MAF_INT_STOP_AFL_PURGE_MAX         OR
                CTR_AFL_PURGE[i] > C_CTR_AFL_PURGE_MAX ) )
then (105)  STATE_AFL_PURGE[i] = "DISABLED"
            LV_CAT_PURGE_REQ_POST_AFL[i] = 0
else (105)
```

Transition 6: *"ON" to "OFF"*

```
if (106)  NC_NT_NR = 2  AND  LC_SENS_AFR_MOD = 1
          % NC_NT_NR = 2 means 2 NOx catalysts are within the exhaust system
          % If NC_NT_NR = 1 or the case above then the calculation of branch 1 is sufficient.
            After that the branch 1 results are copied to branch 2.
then (106)
    case selection on C_STATE_SENS_AFL_PURGE[1]:
        case "VLS_DOWN":
            if (107)  VLS_DOWN[1] >= C_VLS_DOWN_AFL_PURGE_OFF
            then (107)  STATE_AFL_PURGE[1] = "OFF"
                        LV_CAT_PURGE_REQ_POST_AFL[1] = 0
            endif (107)

        case "VLS_NOX_SENS":
            if (108)  VLS_NOX_SENS[1] >= C_VLS_NOX_SENS_AFL_PURGE_OFF
            then (108)  STATE_AFL_PURGE[1] = "OFF"
                        LV_CAT_PURGE_REQ_POST_AFL[1] = 0
            endif (108)

        case "LAMB_NOX_SENS":
            if (109)  LAMB_NOX_SENS[1] <= C_LAMB_NOX_SENS_AFL_PURGE_OFF
            then (109)  STATE_AFL_PURGE[1] = "OFF"
                        LV_CAT_PURGE_REQ_POST_AFL[1] = 0
            endif (109)

        case "VLS_NT_DOWN":
            if (110)  VLS_NT_DOWN[1] >= C_VLS_DOWN_AFL_PURGE_OFF
            then (110)  STATE_AFL_PURGE[1] = "OFF"
                        LV_CAT_PURGE_REQ_POST_AFL[1] = 0
            endif (110)
```

```
        case "NONE":
            ---
    end case selection

    STATE_AFL_PURGE[2] = STATE_AFL_PURGE[1]
    LV_CAT_PURGE_REQ_POST_AFL[2] = LV_CAT_PURGE_REQ_POST_AFL[1]

else (106)
    % Full 2 branch calculation.
    case selection on C_STATE_SENS_AFL_PURGE[i]:
        case "VLS_DOWN":
            if (111)  VLS_DOWN[m] >= C_VLS_DOWN_AFL_PURGE_OFF
            then (111) STATE_AFL_PURGE[i] = "OFF"
                       LV_CAT_PURGE_REQ_POST_AFL[i] = 0
            endif (111)

        case "VLS_NOX_SENS":
            if (112)  VLS_NOX_SENS[k] >= C_VLS_NOX_SENS_AFL_PURGE_OFF
            then (112) STATE_AFL_PURGE[i] = "OFF"
                       LV_CAT_PURGE_REQ_POST_AFL[i] = 0
            endif (112)

        case "LAMB_NOX_SENS":
            if (113)  LAMB_NOX_SENS[k] <= C_LAMB_NOX_SENS_AFL_PURGE_OFF
            then (113) STATE_AFL_PURGE[i] = "OFF"
                       LV_CAT_PURGE_REQ_POST_AFL[i] = 0
            endif (113)

        case "VLS_NT_DOWN":
            if (114)  VLS_NT_DOWN[k] >= C_VLS_DOWN_AFL_PURGE_OFF
            then (114) STATE_AFL_PURGE[i] = "OFF"
                       LV_CAT_PURGE_REQ_POST_AFL[i] = 0
            endif (114)

        case "NONE":
            ---
    end case selection
```

endif (106)

Steady state actions:

\---

endif (105)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_NOX_RED_FAC	4	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_TNT_MDL_MV_IP_NOX_RED_FAC	4	0...FFH	0...1020	4	[°C]
Reduction factor taking into account the chemical NOx conversion					
IP_NT_STC_BAS_RNG_H	8*8	0...FFFFH	0...20971.2	0.32	[mg]
LDP_NOX_CONC_MMV_IP_NT_STC_BAS	8	0...FFFFH	0...65535	1	[ppm]
LDPM_TNT_MDL_NOX	8	0...FFH	0...1020	4	[°C]
NOx trap storage capacity					
IP_NT_EFF_BAS	8*8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_NTLD	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_TNT_MDL_NOX	8	0...FFH	0...1020	4	[°C]
Basic NOx trap efficiency for open exhaust flap					
IP_NT_EFF_COR	8*8	0...FFH	0...3.98437	0.015625	[-]
LDP_NOX_COR_IP_NT_EFF_COR	8	0...FFFFH	0...1023.98437	0.015625	[mg/s]
LDPM_TNT_MDL_NOX	8	0...FFH	0...1020	4	[°C]
NOx trap efficiency correction					
IP_O2_STC_MAX	8*8	0...FFFFH	0...10485.6	0.16	[mg]
LDP_LAMB_LS_UP_MV_IP_O2_STC_MAX	8	0...7FFFH	0...31.99902	0.9766e-3	[-]
LDPM_TNT_MDL_NOX	8	0...FFH	0...1020	4	[°C]
max. storage capacity of O2 in the NOx-trap					
IP_NT_AGI_RGN_FRQ	8*8	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_NT_AGI_NTLD_IP_NT_AGI_RGN	8	0...FFH	0...0.99609	3.9063e-3	[-]
LDP_TNT_MDL_MV_IP_NT_AGI_RGN	8	0...FFH	0...1020	4	[°C]
modification of the actual storage capacity due to aging factor NT_AGI_NTLD					
C_LAMB_O2_STC	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda threshold for O2 storage in lean operation					
C_O2_FLOW_COR_FAC	1	0...FFFFH	0...127.99804	1.9531e-3	[-]
Correction factor for the O2 engine out flow					
C_FAC_O2_EQU	1	0...FFH	0...1.99218	0.0078125	[-]
Factor for the balance of stored NO O2					
C_FAC_COR_O2L	1	0...FFH	0...1.99218	0.0078125	[-]
Correction factor for the O2L decrementation					
C_LAMB_STC	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda threshold for lean operation					
C_LAMB_RGN	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda threshold for rich operation					
C_NTL_INI_ST	1	0...FFFFH	0...10485.6	0.16	[mg]
Initialisation value of the NOx trap absolute loading after engine start					
C_NTLD_RGN_ST_INI	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Initialisation value for NOx trap loading degree at starting regeneration					
C_NT_STC_BAS_MAX_RNG_H	1	0...FFFFH	0...20971.2	0.32	[mg]
maximum value of base NOx trap storage capacity map					
C_NT_STC_INI_RNG_H	1	0...FFFFH	0...20971.2	0.32	[mg]
initialisation value of actual and base NOx trap storagecapacity					
C_NTLD_LGRD	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
limitating gradient of loading degree					
C_NOX_CONC_COR	1	0...FFFFH	0...8192	0.1250019	[-]
Correction factor of the NOx concentration regarding the mol mass					
C_CRLC_NOX_CONC_RED	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correction factor for filtering of NOX_CONC_RED					
C_NTLD_NT_HOM_RST	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
loading degree threshold for resetting the homogeneous time counter T_NT_HOM					
C_T_AFL_NT_HOM_RST	1	0...FFFFH	0...1310.7	0.02	[s]
lean time after which the homogeneous time counter is re-initialised					
C_T_NT_HOM	1	0...FFFFH	0...655.35	0.01	[s]
Lean time threshold for resetting the homogeneous time counter T_NT_HOM					
C_T_NT_HOM_MAX_VLD	1	0...FFFFH	0...655.35	0.01	[s]
Lean time threshold for resetting the homogeneous time counter T_NT_HOM					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NT_ACT	1	0...1H	0...1	1	[-]
Logical constant for NOx catalyst function activation					
LC_NT_AFS_REQ	1	0...1H	0...1	1	[-]
switch to force hom.mode in Passive or enable lean operation					
C_CRLC_NOX_COR_RED	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correction factor for filtering of NOX_COR_RED					
C_CRLC_NOX_CONC_OUT_MDL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correction factor for filtering of NOX_CONC_OUT_MDL					
IP_NT_EFF_BAS_EF_CLOSE	8*8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_NTLD	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_TNT_MDL_NOX	8	0...FFH	0...1020	4	[°C]
Basic NOx trap efficiency for closed exhaust flap					
C_T_STOP_AFL_PURGE_MAX	1	0...FFFFH	0...1310.7	0.02	[s]
Time limit for activation of catalyst purge after lean operation at STATE_NOX = STOP					
C_MAF_INT_STOP_AFL_PURGE_MAX	1	0...FFFFH	0...1456.33333	0.0222222	[g]
Air mass integral limit for activation of catalyst purge after lean operation at STATE_NOX = STOP					
C_CTR_AFL_PURGE_MAX	1	0...FFH	0...255	1	[-]
Maximum number of activation cycles for catalyst purge after lean operation at STATE_NOX = STOP					
C_VLS_DOWN_AFL_PURGE_ON	1	0...3FFH	0...4.99511	4.8828e-3	[V]
VLS_DOWN activation threshold for catalyst purge after lean operation at STATE_NOX = STOP					
C_VLS_DOWN_AFL_PURGE_OFF	1	0...3FFH	0...4.99511	4.8828e-3	[V]
VLS_DOWN deactivation threshold for catalyst purge after lean operation at STATE_NOX = STOP					
C_VLS_NOX_SENS_AFL_PURGE_ON	1	0...578H	-200...1200	1	[mV]
VLS_NOX_SENS activation threshold for catalyst purge after lean operation at STATE_NOX = STOP					
C_VLS_NOX_SENS_AFL_PURGE_OFF	1	0...578H	-200...1200	1	[mV]
VLS_NOX_SENS deactivation threshold for catalyst purge after lean operation at STATE_NOX = STOP					
C_LAMB_NOX_SENS_AFL_PURGE_ON	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
LAMB_NOX_SENS activation threshold for catalyst purge after lean operation at STATE_NOX = STOP					
C_LAMB_NOX_SENS_AFL_PURGE_OFF	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
LAMB_NOX_SENS deactivation threshold for catalyst purge after lean operation at STATE_NOX = STOP					
C_STATE_SENS_AFL_PURGE[NC_NT_NR]	1	0H 1H 2H 3H 4H	VLS_DOWN VLS_NOX_SENS LAMB_NOX_SENS VLS_NT_DOWN NONE	1	[-]
Determination of signal for catalyst purge function after lean operation					
LC_INH_AFL_PURGE_TOUT	1	0...1H	0...1	1	[-]
Inhibition of catalyst purge break if a time out condition fulfilled					
LC_INH_NT_RGN_ST	1	0...1H	0...1	1	[-]
Inhibition of NOx regeneration request at transition STATE_NOX = 'WARMUP' -> 'REGENERATION'					
C_MAF_INT_PUC_WAIT_MIN	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Minimum air mass flow at PUC after which the switch of STATE_NOX = LOAD -> WAIT is allowed					
LC_NT_STST_ACT	1	0...1H	0...1	1	[-]
Activation of special engine start-stop functionality into NOx catalyst functionalities					
LC_INH_RGN_REQ_ST	1	0...1H	0...1	1	[-]
Check of LV_INH_RGN_REQ at transition STATE_NOX = 'WARMUP' -> 'REGENERATION'					
C_TNT_MDL_PURGE_MIN	1	0...FFH	0...1020	4	[°C]
Minimum NOx catalyst temperature for enabling of NOx catalyst purge at STATE_NOX = STOP					
C_TNT_MDL_PURGE_DEAC	1	0...FFH	0...1020	4	[°C]
NOx catalyst temperature for deactivation of NOx catalyst purge at STATE_NOX = STOP					
C_MAF_INT_PUC_RST_RGN_REQ	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Minimum air mass flow at STATE_NOX = WAIT after that the reset of LV_RGN_NT_REQ is allowed					
LC_NTL_INI_ST	1	0...1H	0...1	1	[-]
Selection of NTL initialisation method (0...by NVMY value, 1...by C_NTL_INI_ST)					
C_FAC_NTL_INI_ST	1	0...FFFFH	0...65.535	0.001	[-]
Weighting factor for initialisation of NTL after reset					
C_TNT_MDL_PURGE_MAX	1	0...FFH	0...1020	4	[°C]
Maximum NOx catalyst temperature for enabling of NOx catalyst purge at STATE_NOX = STOP					
LC_STATE_NOX_AFL_ACT	1	0...1H	0...1	1	[-]

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
Activation of NOx state management (STATE_NOX) at LV_INH_AFL=1 or LV_INH_S=1					
LC_STATE_NOX_OPM_AV_ACT	1	0...1H	0...1	1	[-]
Activation of NOx state management (STATE_NOX) at OPM_AV <> "AFS"					
IP_NTL_MIN_AFL_PURGE	6	0...FFFFH	0...10485.6	0.16	[mg]
LDP_VS_IP_NTL_MIN_AFL_PURGE	6	0...FFH	0...255	1	[km/h]
NTL reset threshold for AFL_PURGE control variables					
IP_NTLD_MIN_AFL_PURGE	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_VS_IP_NTLD_MIN_AFL_PURGE	6	0...FFH	0...255	1	[km/h]
NTLD reset threshold for AFL_PURGE control variables					
LC_AFL_PURGE_RST	1	0...1H	0...1	1	[-]
Activation of NTL and NTLD threshold for reset of NT purge control variables					
C_VS_PURGE_MAX	1	0...FFH	0...255	1	[km/h]
Maximum vehicle speed for enabling of NOx catalyst purge at STATE_NOX = STOP					
C_T_AFL_RST	1	0...FFFFH	0...655.35	0.01	[s]
Time limit for reset of T_AFL after an engine STST phase					
C_CRLC_VLS_NOX_SENS	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
filter constant for VLS_NOX_SENS					
C_FAC_NTL_DEC_INT_RGN_NOT_VLD[NC_NT_NR]	1	0...FFH	0...1.99218	0.0078125	[-]
Factor for forcing a regeneration of second bank in relation to first bank					
C_FAC_NTL_DEC_INT_STOP_RGN[NC_NT_NR]	1	0...FFH	0...1.99218	0.0078125	[-]
Factor for end of regeneration of second bank in relation to first bank					
C_VLS_DOWN_RGN_2_MIN	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Limit for detection of a not successful regeneration of second bank					
C_VLS_NS_RGN_2_VLD	1	0...578H	-200...1200	1	[mV]
Limit to enable again detection of end of regeneration of second bank by sensor					
LC_NTL_DEC_INT_AFR_ENA	1	0...1H	0...1	1	[-]
Enabling of rich detection by NTL_DEC_INT ratio at the end of the regeneration of the second bank					
C_VLS_DOWN_MIN_NTL_DEC[NC_CBK_EX_NR]	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Minimum VLS_DOWN voltage for calculation of NTL_DEC_INT					
LC_NS_SHIFT_INH_AFL_PURGE	1	0...1H	0...1	1	[-]
Deactivation of catalyst purge after lean operation at NOx sensor shift diagnosis					
LC_VLS_NS_VLD_AFL_PURGE	1	0...1H	0...1	1	[-]
Deactivation of catalyst purge after lean operation at not valid binary Lambda signal of NOx sensor					
LC_STATE_NOX_INH_NT_AFS_REQ	1	0...1H	0...1	1	[-]
Enabling of STATE_NOX = "STOP" to "LOAD" transition at LV_NT_AFS_REQ = 1					
C_TNT_MDL_MV_NTL_H_RST_MIN	1	0...FFH	0...1020	4	[°C]
Lower NOx catalyst temperature threshold for reset of NTL_H calculation					
C_TNT_MDL_MV_NTL_H_RST_MAX	1	0...FFH	0...1020	4	[°C]
Upper NOx catalyst temperature threshold for reset of NTL_H calculation					
C_TNT_MDL_MV_NTL_H_RST_HYS	1	0...FFH	0...1020	4	[°C]
NOx catalyst temperature hysteresis for NTL_H calculation					
LC_NTL_H_RST_LOAD_TNT	1	0...1H	0...1	1	[-]
Activation of NTL_H reset at out of bounds NOx catalyst temperatures at STATE_NOX = "LOAD"					
C_MAF_INT_PUC_NTL_H_RST	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Minimum mass air flow integral for reset of NTL_H at STATE_NOX = "WAIT"					
C_MAF_INT_STOP_NTL_H_RST	1	0...FFFFH	0...5825.33333	0.0888889	[g]
Minimum mass air flow integral for reset of NTL_H at STATE_NOX = "STOP"					

Notes:

- For detection of a regeneration in catalyst purge the calibration data C_LAMB_RGN has to be calibrated always more lean than the calibrated lambda for catalyst purge.

- C_NT_STC_BAS_MAX_RNG_H should represent the maximum value of the base NOx trap storage capacity map IP_NT_STC_BAS_RNG_H. If any changes occur to IP_NT_STC_BAS_RNG_H, this constant must be updated manually.

53.6.2 NOx catalyst management - request coordination

53.6.2.1 Called blocks

FUNCTION DESCRIPTION:

These blocks are executed when called

53.6.2.1.1 Filtered exhaust gas flow

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: FLOW_NT_MMV = 0

Recurrence: 20 ms

Activation: --

Deactivation: --

Formula section:

$$FLOW\_NT\_MMV = FLOW\_NT\_MMV_{(n-1)} * (1-C\_CRLC\_FLOW\_NT) + MAF\_KGH * C\_CRLC\_FLOW\_NT$$

53.6.2.1.2 Homogeneous time counter handling

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: T_NT_HOM_DELTA = 0

Recurrence: --

Activation: --

Deactivation: --

Formula section:

% decrement per 10ms

if (46) LV_HOM_AFS_ACT = 1

then (46)

```
    if (47)       LV_SENS_AFR_TRIM = 1
    then (47)     T_NT_HOM_DELTA = - IP_T_NT_HOM_DEC (MAF_KGH, LAMB_AV_COR)
    else (47)     T_NT_HOM_DELTA = IP_T_NT_HOM_INC (MAF_KGH, LAMB_AV_COR)
    endif (47)
else (46)   T_NT_HOM_DELTA = 0
endif (46)
```

53.6.2.2 General, definition of slow flags

FUNCTION DESCRIPTION:

General information:

This block is executed in any STATE_NOX

The switch LC_NOX_SENS and the logical variable LV_NOX_SENS indicate the existence of a NOx-Sensor in the exhaust system.

The regeneration of the NOx catalyst can be started by the NOx-Sensor signal or by the NOx trap loading degree model. With LV_NOX_SENS_RGN, generated by the code word LC_NOX_SENS_RGN, it is possible to decide whether only the NOx sensor should be able to request a regeneration or both, model and sensor are able to request:

- LV_NOX_SENS_RGN = 1: The regeneration is started exclusively by the NOx sensor signal. The loading degree model is blocked (the regeneration request before switching to lambda 1 is not affected).
- LV_NOX_SENS_RGN = 0: The regeneration starts when the NOx catalyst loading degree exceeds a threshold or by the signal of the NOx-Sensor (which comes first)

If the system is not equipped with a NOx-Sensor, LV_RGN_REQ_NOX_SENS is permanently set to zero.

Application conditions:

Initialisation: --

Recurrence: 1000 ms

Activation: --

Deactivation: --

Formula section:

```
LV_NOX_SENS = LC_NOX_SENS
LV_NOX_SENS_RGN =
    (LV_NOX_SENS= 1               and
    LC_NOX_SENS_RGN = 1           and
    LV_INH_NOX_SENS_RGN = 0       and
    LV_ADJ_NOX_SENS_RGN = 0)
```

53.6.2.3 General, definition of medium flags

FUNCTION DESCRIPTION:

General information:

This block is executed in any STATE_NOX except PASSIVE.

Application conditions:

Initialisation: --

Recurrence: 100ms

Activation: --

Deactivation: --

Formula section:

```
% definition of flag indicating initialised catalyst state
if (48)      T_NT_HOM = 0
then (48)    LV_NT_HOM_INI = 1
endif (48)
```

53.6.2.4 General, definition of fast flags

FUNCTION DESCRIPTION:

General information:

This block is executed in any STATE_NOX except PASSIVE.

At the beginning of a regeneration phase LV_TRIG_AFR_AFL must be 0. In order to trigger the transition from rich to lean operation the logical value LV_TRIG_AFR_AFL is set to 1 if the NTL becomes lower than C_NTL_TRIG_AFR_AFL.

Every time when a regeneration phase is finished the value SUM_NTL_CYC is incremented by 1. SUM_NTL_CYC has to be stored in the nonvolatile memory. SUM_NTL_CYC can be set to the specific value C_SUM_NTL_CYC_SP using the logical constant LC_SUM_NTL_CYC_NEW.

When either the binary lambda sensor or the NOx sensor indicates a rich exhaust gas downstream NOx trap, the logical variable LV_SENS_AFR_i is set to 1.

The mode of stopping a regeneration can be controlled by C_STATE_RGN_CTL. Thereby, it is possible to stop the regeneration only by the model, only by the sensor signal (LV_SENS_AFR) or by either model or sensor. In addition, a stop of regeneration after a constant regeneration time is possible. If a special regeneration (e.g. for adaptation purposes) was requested by LV_RGN_REQ_AD, the stop of the regeneration is forbidden until the adapting module re-enables it.

Application conditions:

Initialisation: --

Recurrence: 10ms

Activation: --

Deactivation: --

Formula section:

```
% definition of regeneration stop mode
LV_RGN_STOP_SENS =  ( LV_RGN_REQ_AD = 0 )                              and
                    ( C_STATE_RGN_CTL = SENSOR                  or
                      C_STATE_RGN_CTL = SENSOR_MODEL )

LV_RGN_STOP_MDL =   ( LV_RGN_REQ_AD = 0 )                              and
                    ( LV_INH_NT_RGN_STOP_MDL_DIAG[k] = 0
                        % valid if all [k] = 0
                        % condition should be removed at special
                        % case NC_NOX_SENS_CONF = 0
                    )                                                  and
                    ( C_STATE_RGN_CTL = MODEL                   or
                      C_STATE_RGN_CTL = SENSOR_MODEL            or
                      ( LV_VLS_NS_VLD[k] = 0 and
                        LC_VLS_NS_VLD_RGN_STOP_MDL = 1 )
                    )

LV_RGN_STOP_TOUT =  ( LV_RGN_REQ_AD = 0 )
```

% time counter for regeneration is always active; by selecting C_STATE_RGN_CTL=TIME, all other possibilities for stopping the regeneration are switched off

```
% handling of the cycle counter and over all lean-time-counter
if (49)       LC_SUM_NTL_CYC_NEW = 1
then (49)     SUM_NTL_CYC = C_SUM_NTL_CYC_SP
else (49)
          if (49+)    LV_RGN_NT_REQ = 1                            AND
                      STATE_NOX = LOAD -> REGENERATION
          then (49+)      $T\_AFL\_SUM_n = T\_AFL\_SUM_{n-1} + T\_AFL$
          endif (49+)
endif (49)
```

```
% calculation of trigger bit for end of regeneration
if (51)       NTL < C_NTL_TRIG_AFR_AFL
then (51)     LV_TRIG_AFR_AFL=1
endif (51)
```

```
if (52a)     LC_SENS_AFR_MOD = 1
then (52a) call "Rich mode calculation of binary NOx-Senor signal"
else (52a)
      if (53) LV_SENS_AFR_i = 0 or STATE_NOX <> REGENERATION
      then (53)
            case selection on C_STATE_SENS_AFR_i:
                % detection of a rich exhaust gas downstream the NOx trap
                     VLS_DOWN:
                     LV_SENS_AFR_i = (VLS_DOWN[m] >=
                                          ID_VLS_DOWN_RGN_STOP(NTLD_RGN_ST))
                     VLS_NOX_SENS:
                     if (54)          LV_NOX_SENS_VLS_VLD_k = 0
                                                   % if at least one of all _i flags is 0
                     then (54) LV_RGN_STOP_MDL=1
                     else (54)
                              LV_SENS_AFR_i =
                              (VLS_NOX_SENS_k >= C_VLS_NOX_SENS_RGN_STOP)
                              and   (NOX_NS_AD[k] >= C_NOX_NS_AD_RGN_STOP
                                     or      LV_NOX_NS_DIAG_VLD_k = 0)
                     endif (54)
                     LAMB_NOX_SENS:
                     if (55)          LV_NOX_SENS_LAMB_VLD_k = 0
                                              % if at least one of all _i flags is 0
                     then (55)        LV_RGN_STOP_MDL=1
                     else (55)        LV_SENS_AFR_i = (LAMB_NOX_SENS_k <=
                                              C_LAMB_NOX_SENS_RGN_STOP)
                     endif (55)
                     VLS_NT_DOWN:
                     LV_SENS_AFR_i = (VLS_NT_DOWN_k >=
                                          ID_VLS_DOWN_RGN_STOP(NTLD_RGN_ST))
                     NONE:
                     LV_SENS_AFR_i = 0
            end case selection
      endif (53)
endif (52a)
```

% The homogeneous time counter T_NT_HOM is decremented, when the catalyst is operated slightly rich, i.e. the upstream lambda sensor indicates homogeneous mixture and the downstream sensor indicates slightly rich mixture; As the threshold for detecting a slightly rich mixture may be different from the threshold for stopping a regeneration, a separate flag is provided.

```
case selection on C_STATE_SENS_AFR_TRIM_i:
VLS_DOWN:
  LV_SENS_AFR_TRIM_TMP_i = (VLS_DOWN[m] >= C_VLS_DOWN_AFR_TRIM)
VLS_NOX_SENS:
if (56)     LV_NOX_SENS_VLS_VLD_k = 0   % if at least one of all _k flags is 0
then (56)   LV_SENS_AFR_TRIM_TMP_i =0
else (56)   LV_SENS_AFR_TRIM_TMP_i = (VLS_NOX_SENS_k >=
                                        C_VLS_NOX_SENS_AFR_TRIM)
endif (56)
LAMB_NOX_SENS:
if (57)     LV_NOX_SENS_LAMB_VLD_k = 0 % if at least one of all _k flags is 0
then (57)   LV_SENS_AFR_TRIM_TMP_i =0
else (57)   LV_SENS_AFR_TRIM_TMP_i = (LAMB_NOX_SENS_k <= C_LAMB_NOX_SENS_AFR_TRIM)
endif (57)
VLS_NT_DOWN:
  LV_SENS_AFR_TRIM_TMP_i = (VLS_NT_DOWN_k >= C_VLS_DOWN_AFR_TRIM)
NONE:
  LV_SENS_AFR_TRIM_TMP_i = 0
end case selection
LV_SENS_AFR_TRIM = LV_SENS_AFR_TRIM_TMP_1 and LV_SENS_AFR_TRIM_TMP_2


% detection of homogeneous lambda=1 or richer operation
if (58)     {(LV_HOM_AFS_REQ = 1)
     or     [(LV_HOM_ACT = 1) and (LV_HOM_AFL_ACT = 0)]}
     and    (LV_PUC_REQ = 0)
then (58) LV_AFL_NOT = 1
else (58)   LV_AFL_NOT = 0
endif (58)
```

53.6.2.4.1 Rich mode calculation of binary NOx-Senor signal

FUNCTION DESCRIPTION:

General information:

53.6.3 Rich mode calculation of binary NOx Sensor signal (named NOXY_VLSAFCLC110 in Figure 69)

The function sets the status flag LV_SENS_AFR_1 to rich mode and parallel sets the end of first regeneration flag LV_SENS_RGN_1_READY when the binary NOx-Sensor signal goes up for the first time over a threshold and sets the status flag LV_SENS_AFR_2 to rich mode in three ways (see Figure 68 below):

a) when the same binary signal goes up for the second time over a threshold (see Figure 68a).

b) when the binary NOx-Sensor signal goes for the first time over the threshold, then it goes not down enough and finally goes up (see Figure 68b).

c) when the binary NOx-Sensor signal goes for the first time over the threshold and then goes not down and the maximal time is over (see Figure 68c).

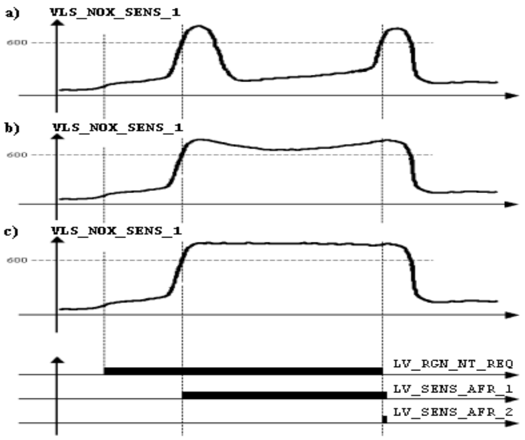


Figure 68: Signals during a NOx regeneration

This specification is special designed for systems with two banks (including two NOx-Traps) and Y-configuration of the exhaust gas system.

Description:

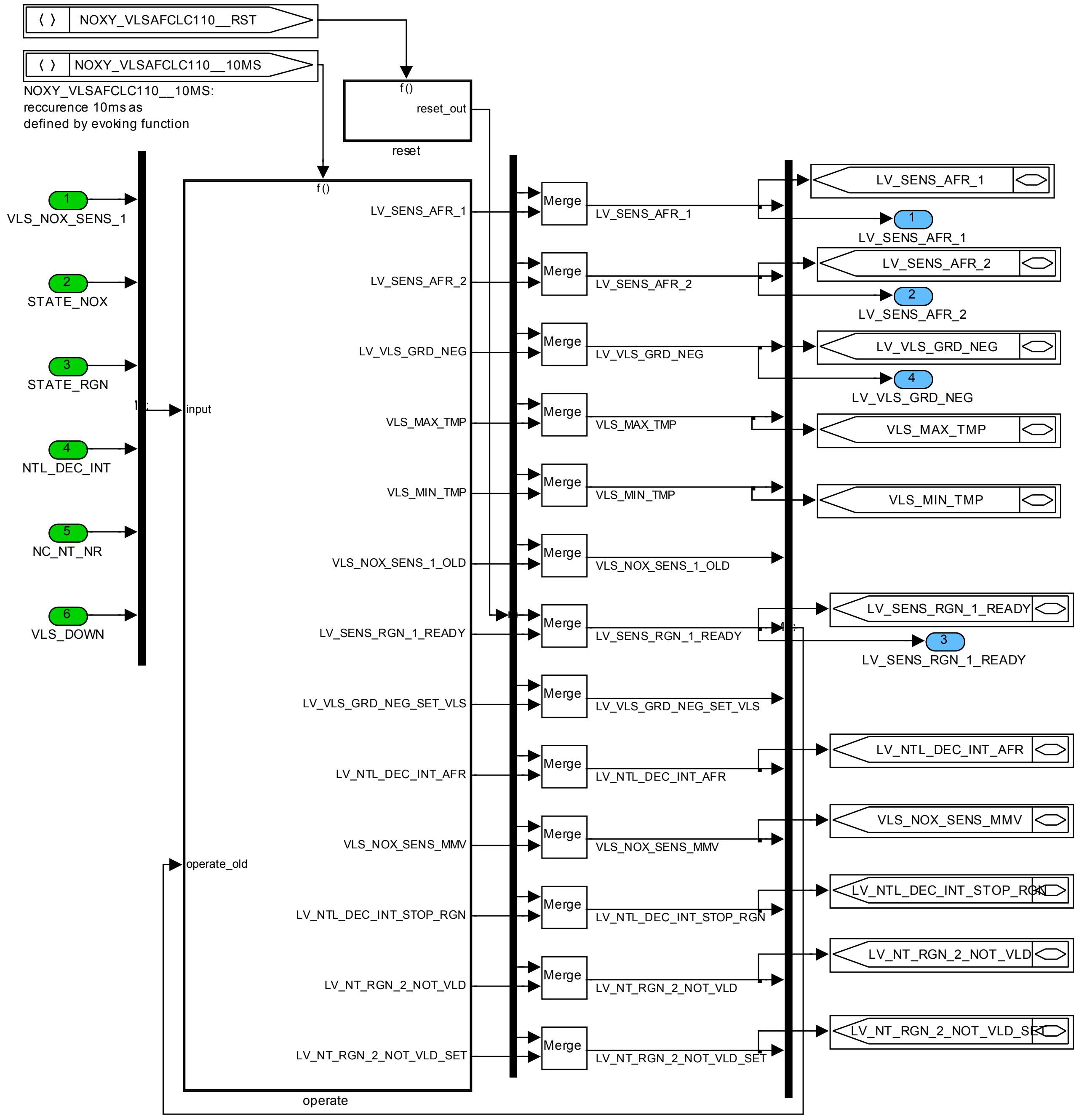


Figure 69: NOXM_VLSAFCLC

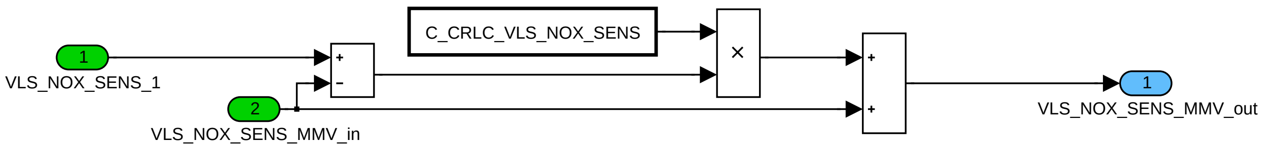


Figure 70 NOXM_VLSAFCLC/ OPERATE/ CRLC

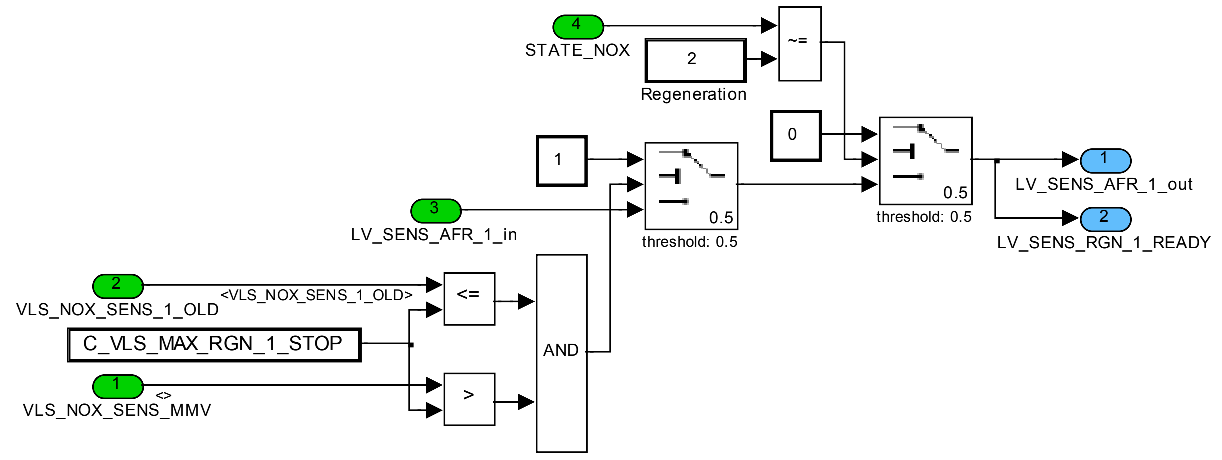


Figure 71 NOXM_VLSAFCLC/ operate/ SENS_AFR_1

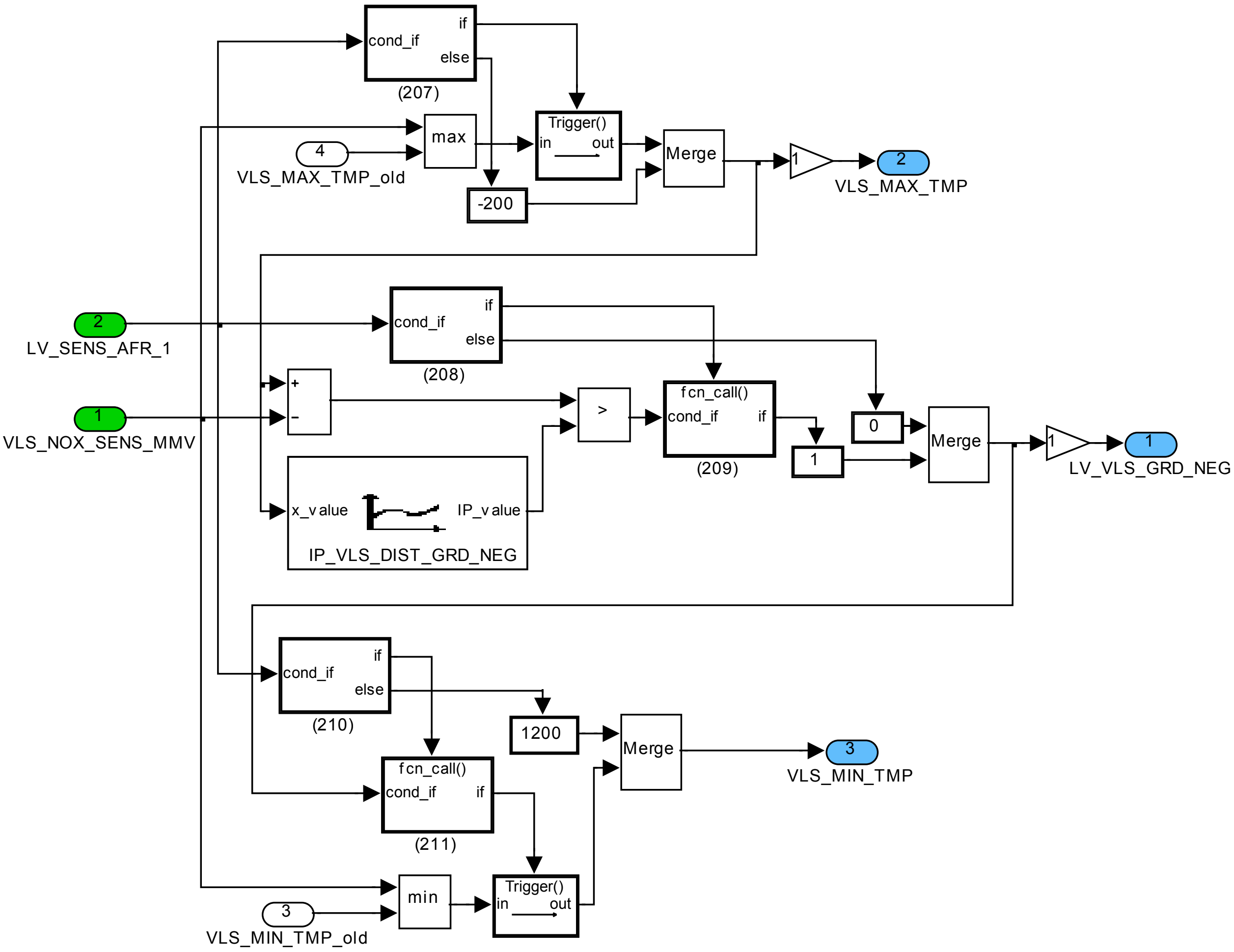


Figure 72 NOXM_VLSAFCLC/ operate/ GRD_CLC

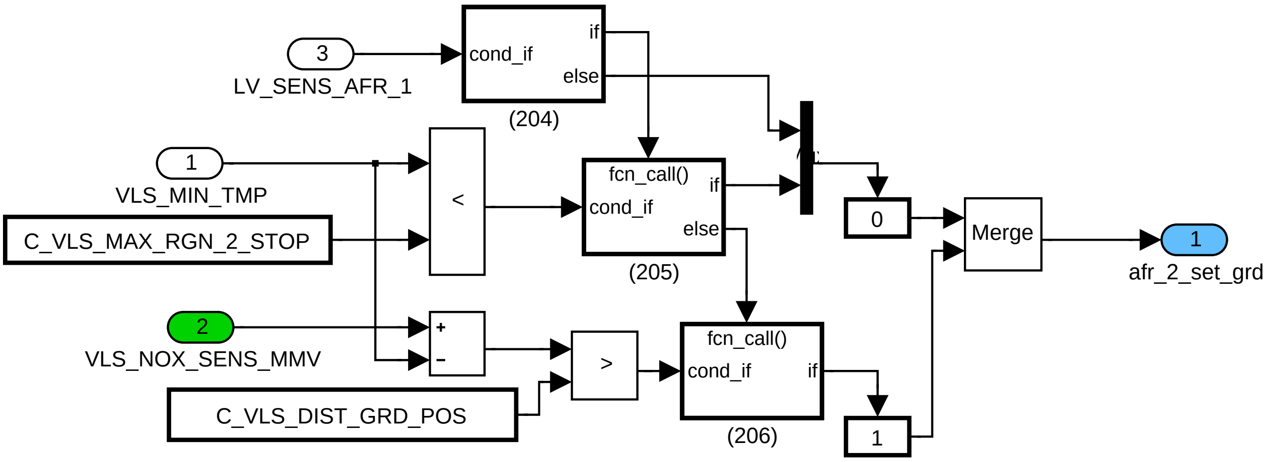


Figure 73 NOXM_VLSAFCLC/ operate/ AFR_2_SET_GRD

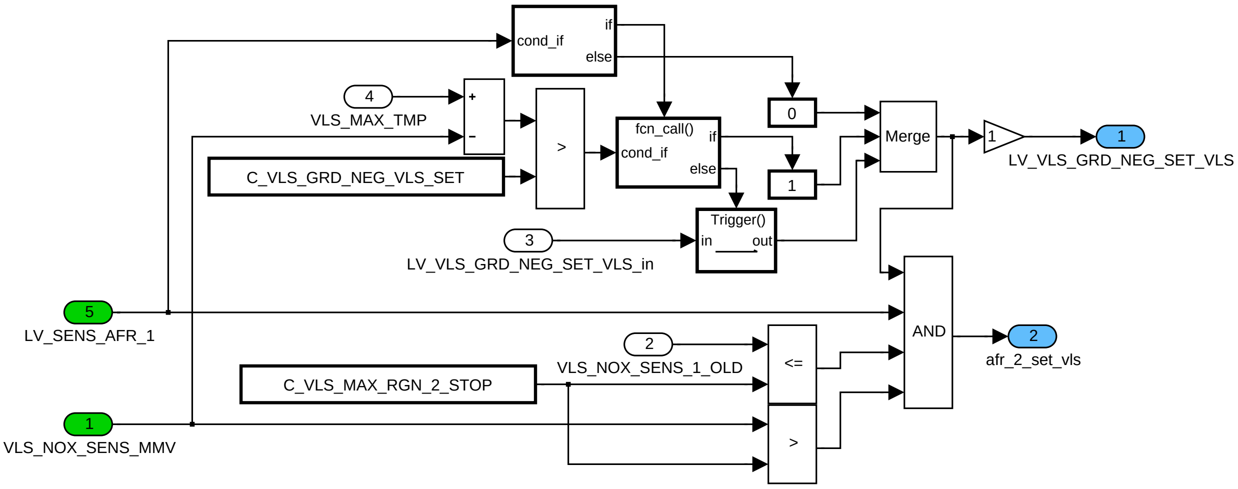
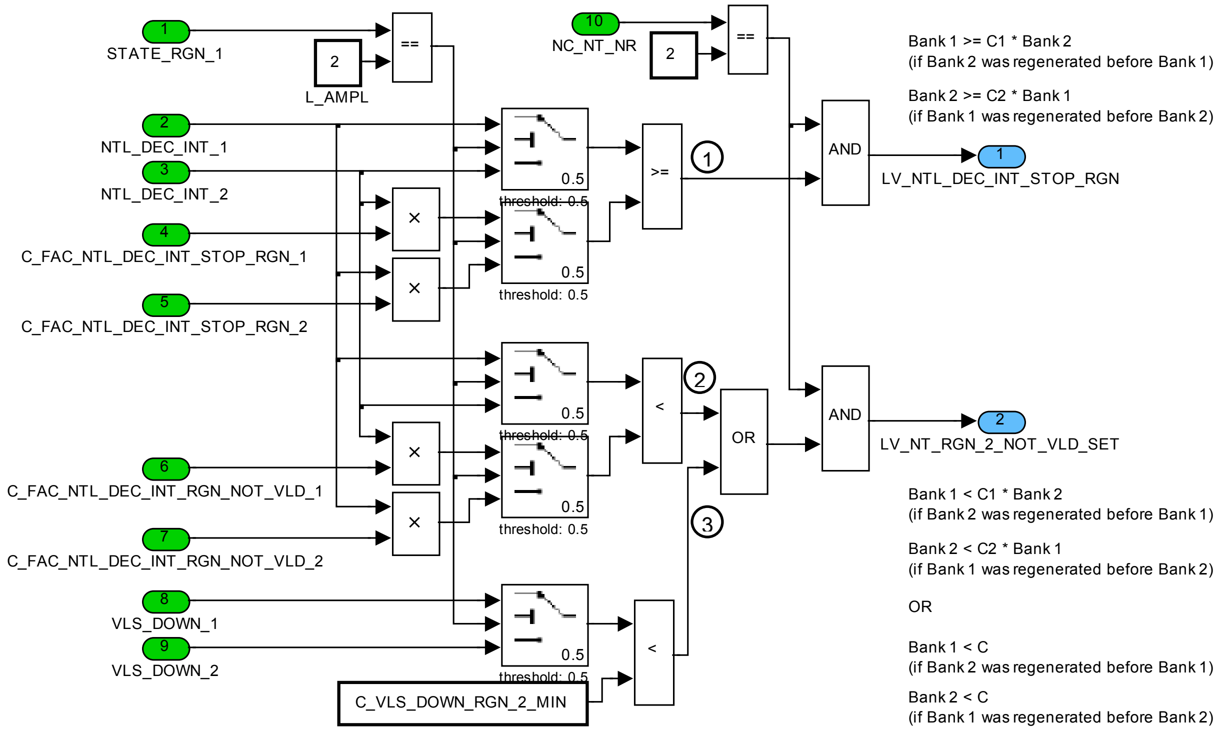


Figure 74 NOXM_VLSAFCLC/ operate/ AFR_2_SET_VLS



Bank 1 >= C1 * Bank 2
(if Bank 2 was regenerated before Bank 1)

Bank 2 >= C2 * Bank 1
(if Bank 1 was regenerated before Bank 2)

Bank 1 < C1 * Bank 2
(if Bank 2 was regenerated before Bank 1)

Bank 2 < C2 * Bank 1
(if Bank 1 was regenerated before Bank 2)

OR

Bank 1 < C
(if Bank 2 was regenerated before Bank 1)

Bank 2 < C
(if Bank 1 was regenerated before Bank 2)

For NC_NT_NR = 1 the variables NTL_DEC_INT_2, C_FAC_NTL_DEC_INT_STOP_RGN_2 and C_FAC_NTL_DEC_INT_RGN_NOT_VLD_2 are not defined. In this case use as substitute the variables NTL_DEC_INT_1, C_FAC_NTL_DEC_INT_STOP_RGN_1 and C_FAC_NTL_DEC_INT_RGN_NOT_VLD_1.

Figure 75 NOXM_VLSAFCLC/ operate/ NTL_DEC_INT_THD

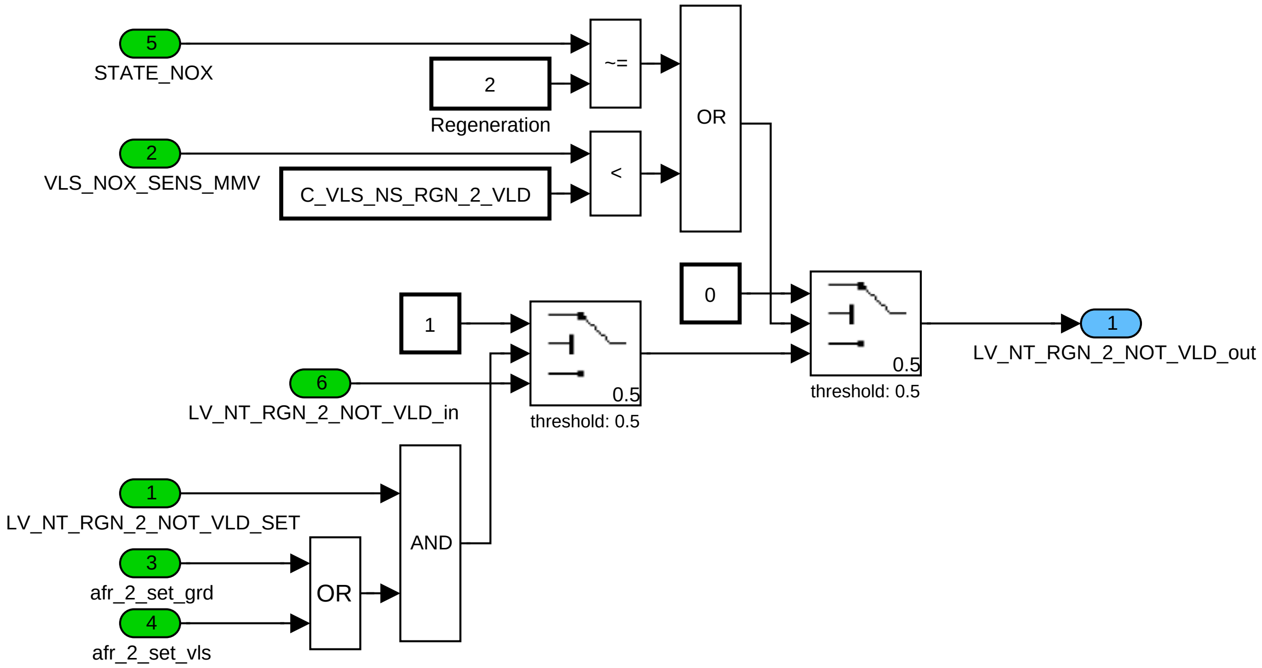


Figure 76 NOXM_VLSAFCLC/ operate/ RGN_NOT_VLD

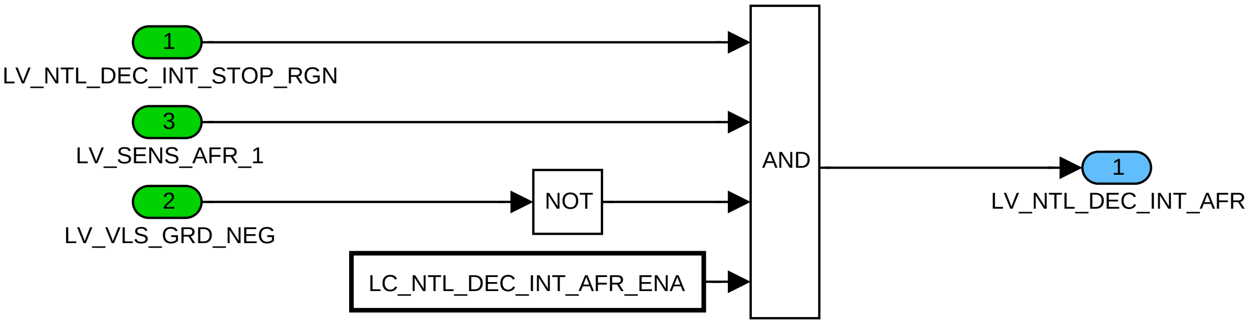
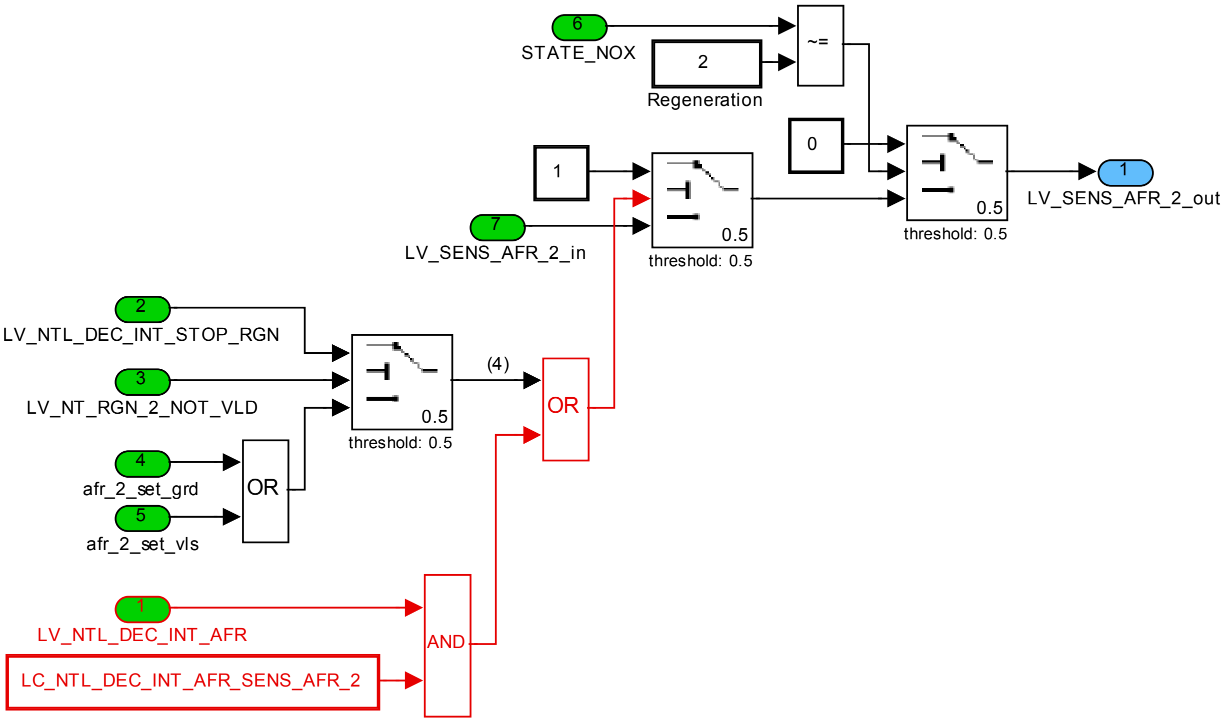


Figure 77 NOXM_VLSAFCLC/ operate/ NTL_DEC_INT_AFR



LV_SENS_AFR_2 will only be calculated if NC_NT_NR = 2

Figure 78 NOXM_VLSAFCLC/ operate/ SENS_AFR_2

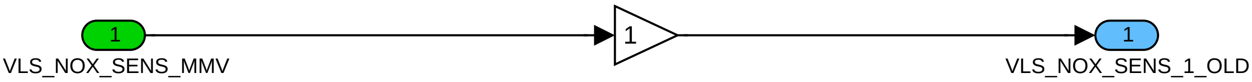


Figure 79 NOXM_VLSAFCLC/ operate/ OLD

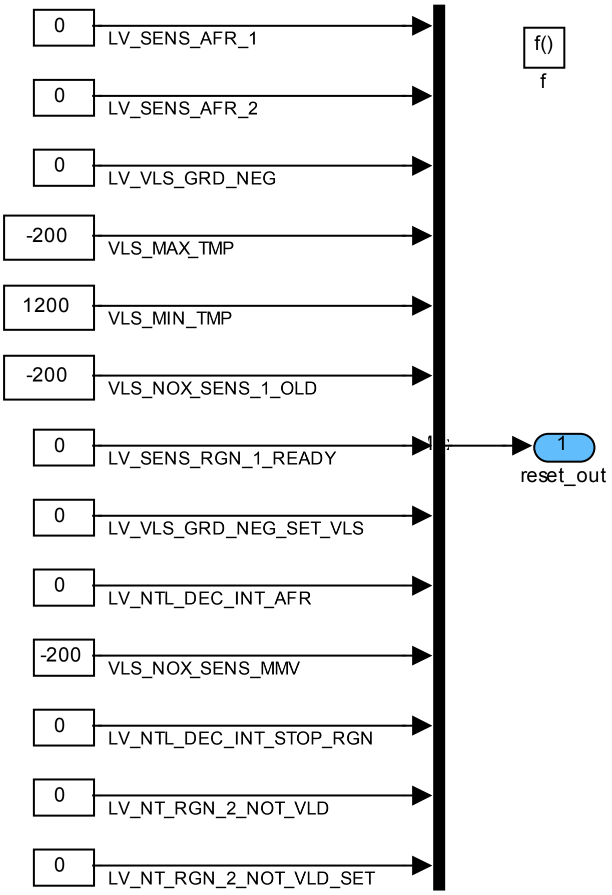


Figure 80 NOXM_VLSAFCLC/ RESET

53.6.3.1 Request of a regeneration phase

FUNCTION DESCRIPTION:

General information:

This block is executed in STATE_NOX = LOAD

A request of a regeneration of the NOx catalyst can be made by the following conditions:

• from the model by a high loading degree	LV_RGN_REQ_NTLD
• by the signal of the post-trap NOx-Sensor	LV_RGN_REQ_NOX_SENS
• before switching to lambda=1 or richer operation	LV_RGN_REQ_AFS
• by the diagnosis / adaptation module	LV_RGN_REQ_AD

In order to guarantee a minimum duration of the lean phase the value IP_T_AFL_MIN has to be exceeded by the counter T_AFL before any regeneration request can be set.

In order to limit the regeneration frequency, T_AFL has to exceed C_T_AFL_RGN_REQ_NTLD before a regeneration request can be made by the loading degree model.

After a regeneration it is prohibited to generate a new regeneration request by the NOx sensor (LV_RGN_REQ_NOX_SENS_TMP_i = 1) for the time C_T_NOX_SENS_VLD. The time C_T_NOX_SENS_VLD is controlled by the timer T_AFL. T_AFL has to exceed C_T_NOX_SENS_VLD before a regeneration can be requested by the NOx sensor.

In case of active auxiliary functionalities module, a regeneration request by the loading degree model or by the NOx sensor may be inhibited by the flag LV_INH_RGN_AD.

When switching from lean conditions to lambda=1 or richer operation a regeneration can be required. The request of this regeneration phase depends on the maximum NOx-trap loading degree for stoichiometric operation IP_NTLD_MAX_AFS_MT / IP_NTLD_MAX_AFS_AT. The check for this maximum loading degree for stoichiometric operation is carried out ever when switching to homogeneous Lambda=1 or richer engine operation which should indicated by LV_AFL_NOT=1.

A regeneration can only be requested if various conditions for allowing a regeneration are fulfilled. In this case, the logical variable LV_RGN_CDN is set to 1.

Signal flow diagram:

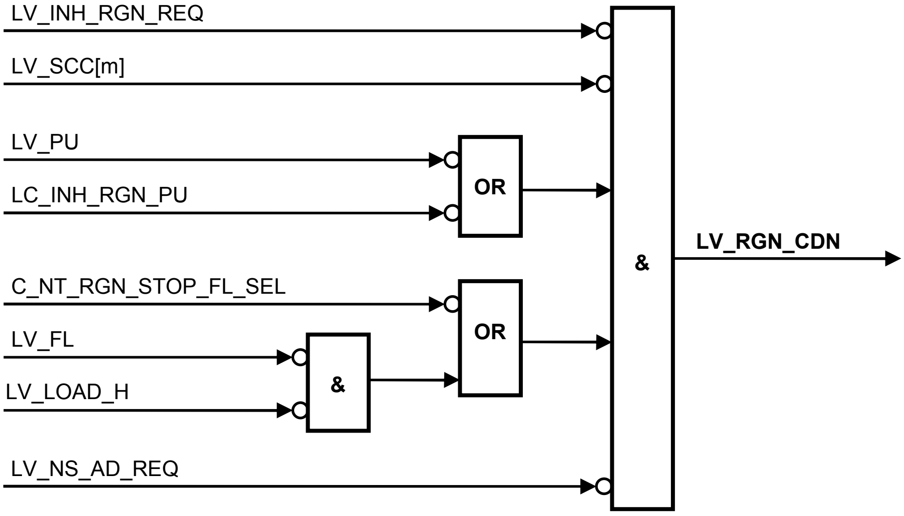


Figure 81: determination of regeneration conditions

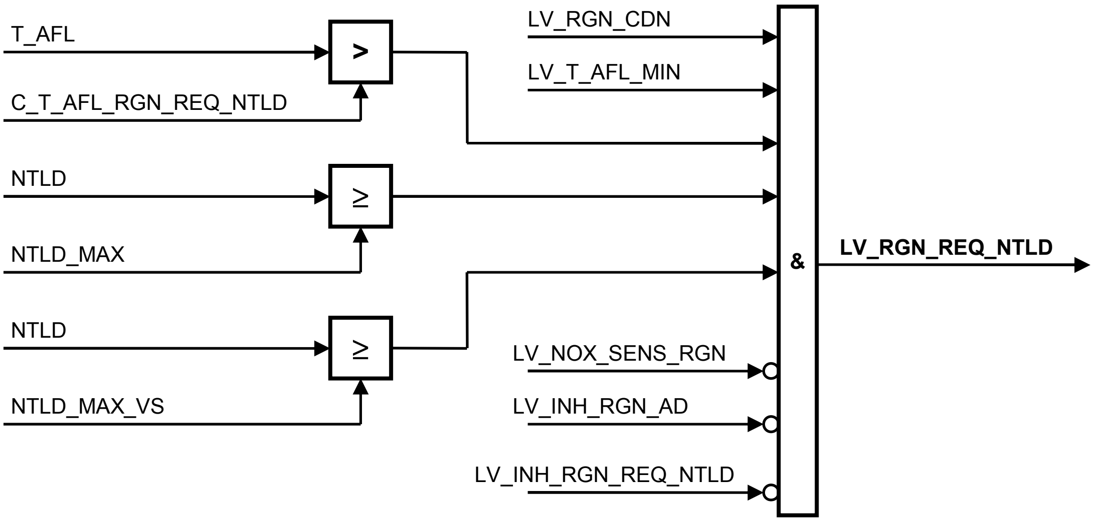


Figure 82: regeneration request due to modelled loading degree

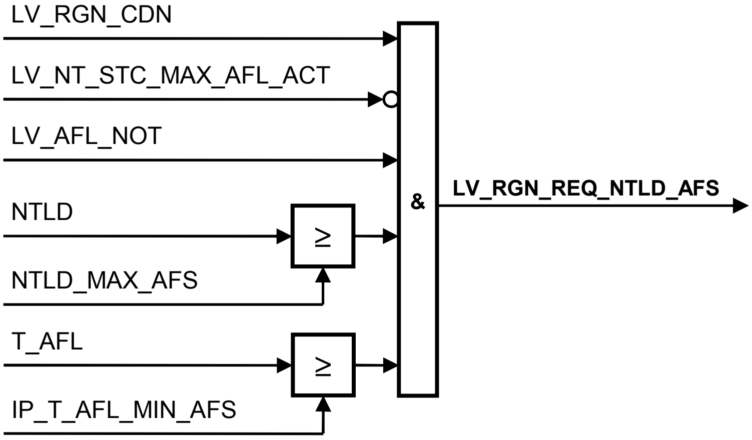


Figure 83: regeneration request due to a transition to non-lean operation

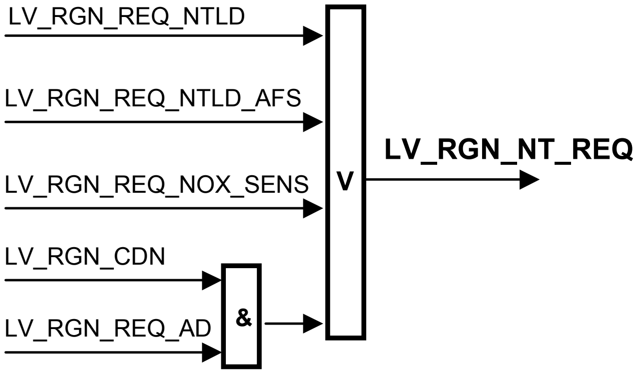


Figure 84: central regeneration request

Application conditions:

Initialisation: --

Recurrence: 20ms

Activation: --

Deactivation: --

Formula section:

```
if (59)     LV_RGN_NT_REQ = 0
then (59)   call “1.2.5.1 calculation of the sub-requests”
endif (59)

// final regeneration request
LV_RGN_NT_REQ =
            [LV_RGN_REQ_NTLD = 1
        or  (LV_RGN_REQ_NOX_SENS = 1)
        or  (LV_RGN_REQ_AD = 1 and LV_RGN_CDN = 1)
        or  LV_RGN_REQ_NTLD_AFS = 1]
```

53.6.3.1.1 Calculation of the sub-requests

```
% calculation of maximum loading degree for a model based regeneration
% (position changed)
if (61)   LV_NTLD_ADJ = 1
then (61)
          if (131)    LV_AT = 1
          then (131)  NTLD_MAX = IP_NTLD_MAX_ADJ_AT (TQI_REQ_FAST, N_32)
          else (131)  NTLD_MAX = IP_NTLD_MAX_ADJ_MT (TQI_REQ_FAST, N_32)
          endif (131)
else (61)
          if (132)    LV_AT = 1
          then (132)  NTLD_MAX = IP_NTLD_MAX_AT (TQI_REQ_FAST, N_32)
          else (132)  NTLD_MAX = IP_NTLD_MAX_MT (TQI_REQ_FAST, N_32)
          endif (132)
endif (61)

if (133)    LV_AT = 1
then (133)  NTLD_MAX_VS = IP_NTLD_MAX_VS_AT (VS, NT_AGI_S_RED)
else (133)  NTLD_MAX_VS = IP_NTLD_MAX_VS_MT (VS, NT_AGI_S_RED)
```

```
endif (133)

% general conditions for a regeneration
if (62)       (LV_INH_RGN_REQ =  0)
        and   (LV_SCC[m] = 0)
        and   (LV_PU = 0 or LC_INH_RGN_PU = 0)
        and   ( (LV_FL = 0 and LV_LOAD_H = 0) or C_NT_RGN_STOP_FL_SEL = 0)
        and   (LV_NS_AD_REQ = 0)
then (62)  LV_RGN_CDN = 1
else (62)  LV_RGN_CDN = 0
endif (62)

% minimum lean time condition
if (127)     LV_NT_RGN_STOP_PUC = 0
then (127) LV_T_AFL_MIN = (T_AFL >= IP_T_AFL_MIN(VS))
else (127)  LV_T_AFL_MIN =
                  (T_AFL >= IP_T_AFL_MIN_NT_RGN_STOP(VS))
endif (127)

NOx sensor threshold calculation:
if (63)      LV_AT = 1
then (63)   NOX_SENS_MAX_BAS = IP_NOX_SENS_MAX_AT (FLOW_NT_MMV, N_32)
else (63)   NOX_SENS_MAX_BAS = IP_NOX_SENS_MAX_MT (FLOW_NT_MMV, N_32)
endif (63)

if (64)      LV_NOX_SENS_MAX_ADJ = 0
then (64)   NOX_SENS_MAX = NOX_SENS_MAX_BAS*
                              IP_FAC_NOX_SENS_MAX (NT_AGI_S_RED,TNT_MDL_MV)
else (64)   NOX_SENS_MAX = NOX_SENS_MAX_BAS + C_NOX_SENS_MAX_OFS
endif (64)

% regeneration request by NOx sensor signal
if (65)  LV_RGN_CDN = 1
        and  LV_T_AFL_MIN = 1
        and  LV_NOX_SENS = 1
        and  LV_T_NS_VLD = 1                 % NOx sensor readiness
        and  LV_INH_RGN_REQ_NOX_SENS = 0
```

```
        and  LV_INH_RGN_AD = 0
        and  (LV_HOM_AFS_REQ = 0 and (LV_S_ACT = 1 or LV_HOM_AFL_ACT = 1))
then (65)    call "NOx sensor based regeneration request"        (see next page)
else (65)    LV_RGN_REQ_NOX_SENS = 0
endif (65)

% regeneration request by modelled loading degree
LV_RGN_REQ_NTLD =
        [LV_RGN_CDN = 1
        and  LV_T_AFL_MIN = 1
        and  T_AFL > C_T_AFL_RGN_REQ_NTLD
        and  NTLD >= NTLD_MAX
        and  NTLD >= NTLD_MAX_VS
        and  LV_NOX_SENS_RGN = 0          % rgn req only by NOx sensor
        and  LV_INH_RGN_AD = 0
        and  LV_INH_RGN_REQ_NTLD = 0]

% regeneration request before switching to lambda=1.0 operation
if (66)      LV_AT = 1
then (66)    NTLD_MAX_AFS_BAS = IP_NTLD_MAX_AFS_AT (TQI_REQ_FAST, N_32)
else (66)    NTLD_MAX_AFS_BAS = IP_NTLD_MAX_AFS_MT (TQI_REQ_FAST, N_32)
endif (66)

if (67)  LV_ERR_EGR_2 = 1
then (67)    NTLD_MAX_AFS = NTLD_MAX_AFS_BAS*C_FAC_NTLD_MAX_AFS_ERR_EGR
else (67)    NTLD_MAX_AFS = NTLD_MAX_AFS_BAS
endif (67)

if (68)      LC_NTLD_MAX_AFS = 1
then (68)    LV_RGN_REQ_NTLD_AFS =
                    [LV_RGN_CDN = 1
        and         T_AFL >= IP_T_AFL_MIN_AFS(VS)
        and         LV_AFL_NOT = 1
        and         NTLD >= NTLD_MAX_AFS
        and         LV_NT_STC_MAX_AFL_ACT = 0]
else (68)    LV_RGN_REQ_NTLD_AFS =
```

```
            [LV_RGN_CDN = 1
    and     T_AFL >= IP_T_AFL_MIN_AFS(VS)
    and     LV_AFL_NOT = 1
    and     NTLD_MDL >= NTLD_MAX_AFS
    and     LV_NT_STC_MAX_AFL_ACT = 0]
endif (68)
end
```

53.6.3.1.2 NOx sensor based regeneration request

This block is executed when called

General information:

If the system has only 1 cylinder bank then the variable LV_RGN_REQ_NOX_SENS_TMP_1 generates directly the value of the logical variable LV_RGN_REQ_NOX_SENS. If the system is equipped with 2 cylinder banks, a regeneration request is set LV_RGN_REQ_NOX_SENS = 1, if at least one cylinder bank requests a regeneration LV_RGN_REQ_NOX_SENS_TMP_i = 1.

With LV_NOX_SENS_NOX_VLD_k is indicated that the signal NOX_NS_AD[k] is valid. In this case NOX_NS_AD[k] can be used for the generation of a regeneration request. Otherwise the regeneration request can not be generated by using NOX_NS_AD[k].

A NOx regeneration request can be generated by different methods. These are:

- the NOx concentration, which is measured downstream of NOx catalyst, exceeds a threshold
- the downstream measured NOx integral exceeds a threshold
- the ratio of downstream measured NOx integral to mileage exceeds a threshold
- the ratio of downstream measured NOx integral to upstream modelled NOx integral exceeds a threshold

Recurrence: called by superior chapter

Signal flow diagram:

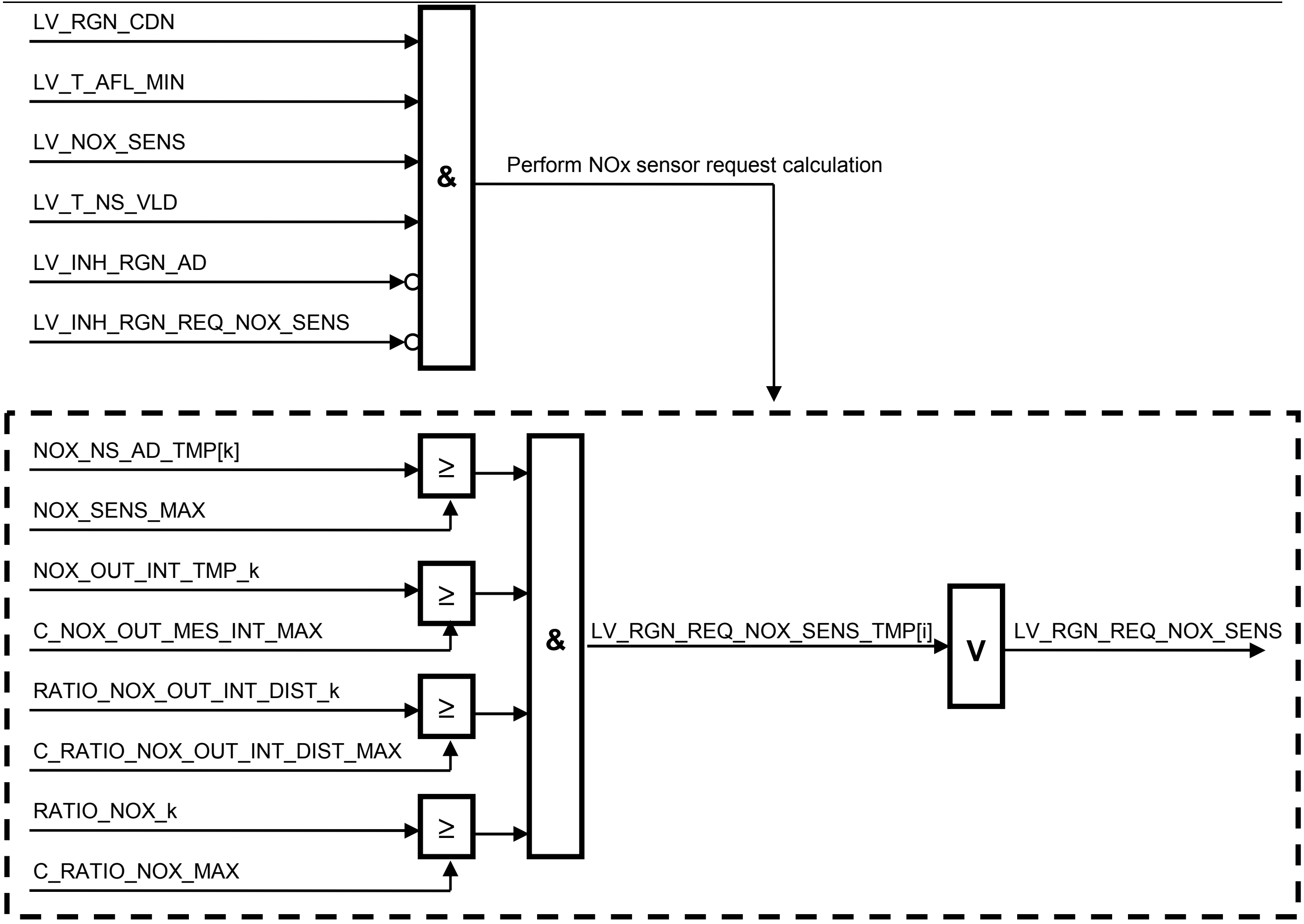


Figure 85: Request of a catalyst regeneration by NOx sensor signal

Formula section:

```
If (69)       LC_NOX_MDL_CTRL_ACT =1
    and       LV_INH_NOX_MDL_CTRL_RGN = 0
    and | NOX_CONC_OUT_MDL_MMV - NOX_NS_AD[k] | <= C_NOX_NS_AD_DIF_MON
                 % NOX_NS_AD[k] separately treated
then (69)     NOX_NS_AD_TMP[k] = NOX_CONC_OUT_MDL_MMV
              NOX_OUT_INT_TMP_k = NOX_OUT_MDL_INT
else (69) if (70)   LV_NOX_SENS_NOX_VLD_k = 1
              then (70)     NOX_NS_AD_TMP[k] = NOX_NS_AD[k]
                            NOX_OUT_INT_TMP_k = NOX_OUT_MES_INT_k
              else (70)     NOX_NS_AD_TMP[k] = 0
                            NOX_OUT_INT_TMP_k = 0
              endif (70)
endif (69)
```

DIST_AFL = (DIST – DIST_AFL_ST)

```
if (89)     DIST_AFL > C_DIST_AFL_MIN
then (89)   RATIO_NOX_OUT_INT_DIST_k = NOX_OUT_INT_TMP_k / DIST_AFL
else (89)   RATIO_NOX_OUT_INT_DIST_k = 0
endif (89)

if (90)     NOX_COR_INT > C_NOX_COR_INT_MIN
then (90)   RATIO_NOX_k = (NOX_OUT_INT_TMP_k / NOX_COR_INT) * 100%
else (90)   RATIO_NOX_k = 0
endif (90)

if (71)     NOX_NS_AD_TMP[k] >= NOX_SENS_MAX
    and     NOX_OUT_INT_TMP_k >= C_NOX_OUT_MES_INT_MAX
    and     RATIO_NOX_OUT_INT_DIST_k >= C_RATIO_NOX_OUT_INT_DIST_MAX
    and     RATIO_NOX_k >= C_RATIO_NOX_MAX
then (71)   LV_RGN_REQ_NOX_SENS_TMP_i = 1
endif (71)

if (72)     LV_RGN_REQ_NOX_SENS_TMP[1] = 1    or
            LV_RGN_REQ_NOX_SENS_TMP[2] = 1
then (72)   LV_RGN_REQ_NOX_SENS = 1
            LV_NOX_SENS_NOX_VLD_RGN_ST_i = LV_NOX_SENS_NOX_VLD_k
                    % both .._rgn_st_i flags have to be frozen if the common request is set
endif  (72)
```

53.6.3.2 Request of a NOx catalyst purge

FUNCTION DESCRIPTION:

General information:

This block is executed in STATE_NOX = LOAD

If there is no request for a regeneration before switching from lean burn condition to lambda=1 or richer mixture a NOx catalyst purge request is set dependent on the amount of stored NOx and O2 (NTL + O2L). The request is reset under the same conditions a regeneration is reset **or** when lambda exceeds C_LAMB_RGN.

The check for setting the request is only to be carried out if the request is yet unset.

LV_ACT_INT_PUC_AFL is passed to the *“catalyst enrichment function (cat-purge function for continuous lambda control)”* where a displacement of the lambda-controller setpoint will be done.

Signal flow diagram:

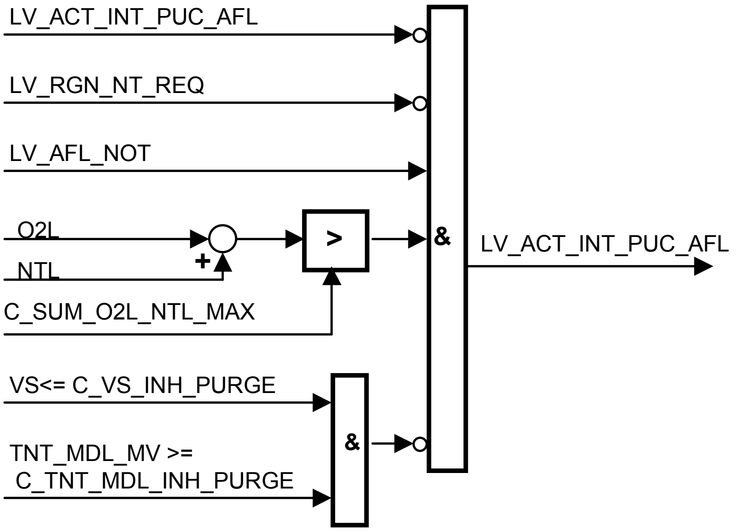


Figure 86: request of a catalyst purge

Application conditions:

Initialisation: --

Recurrence: 20ms

Activation: --

Deactivation: --

Formula section:

```
if (74)   LV_ACT_INT_PUC_AFL_i = 0                          and    % separately treated
          LV_RGN_NT_REQ = 0                                 and
          LV_AFL_NOT = 1                                    and
          (O2L + NTL) > C_SUM_O2L_NTL_MAX                   and
          LV_NT_STC_MAX_AFL_ACT = 0                         and
          not (VS <= C_VS_INH_PURGE                    and
                TNT_MDL_MV >= C_TNT_MDL_INH_PURGE)          and
          VS <= C_VS_PURGE_REQ_MAX                          and
          TNT_MDL_MV <= C_TNT_MDL_PURGE_REQ_MAX
then (74)   LV_ACT_INT_PUC_AFL_i = 1
            LV_CAT_PURGE_REQ_POST_AFL[i] = 1
else (74)
          if (130)  LV_CAT_PURGE_REQ_POST_AFL[i] = 1        and
                    LV_AFL_NOT = 0
```

```
        then (130)  LV_ACT_INT_PUC_AFL_i = 0
                    LV_CAT_PURGE_REQ_POST_AFL[i] = 0
        endif (130)
    endif (74)
```

53.6.3.3 Reset of regeneration and purge requests

FUNCTION DESCRIPTION:

General information:

This block is executed in STATE_NOX = REGENERATION

A regeneration phase is stopped under the following conditions:

- The modelled loadings of NOx and O2L both fall below a threshold
- All downstream sensors detect a rich mixture LV_SENS_AFR
- A constant time period is exceeded
- an engine state with higher priority is activated. These higher prior engine states can be fuel cut-off (PUC) or full load (FL).

The regeneration stopping conditions are valid under the following circumstances:

- A stop out of the model is possible if LV_RGN_STOP_MDL is set to 1.
- A stop by the sensor signal is possible if LV_RGN_STOP_SENS is set to 1.
- A stop by the time counter is possible if LV_RGN_STOP_TOUT is set to 1.
- Also a combination of sensor stop and model stop is possible.

The priority of the engine states fuel cut-off (PUC) and full load (FL) in relation to a running NOx regeneration are selectable by C_NT_RGN_STOP_PUC_SEL and C_NT_RGN_STOP_FL_SEL. It can be chosen:

C_NT_RGN_STOP_PUC_SEL =

0 ... the NOx catalyst control functionality does not react on the engine state fuel cut-off (PUC)

1 ... a running NOx regeneration is broken, when the air mass flow at PUC exceeds a threshold;
the minimum duration of next "LOAD" phase is defined by
IP_T_AFL_MIN_NT_RGN_STOP

2 ... a running NOx regeneration is broken, when the air mass flow at PUC exceeds a threshold and the clutch is closed;
the minimum duration of next "LOAD" phase is defined by
IP_T_AFL_MIN_NT_RGN_STOP

C_NT_RGN_STOP_FL_SEL =

0 ... the NOx catalyst control functionality does not react on the engine state full load or a high engine load near full load.

1 ... new NOx regeneration requests are inhibited up to the end of full load phase or high engine load phase;
a running NOx regeneration is broken, when the air mass flow at full load exceeds a threshold;
the minimum duration of next "LOAD" phase is defined by IP_T_AFL_MIN

2 ... new NOx regeneration requests are inhibited up to the end of full load phase or high engine load phase;
a running NOx regeneration is broken, when the air mass flow at full load exceeds a threshold;
the minimum duration of next "LOAD" phase is defined by IP_T_AFL_MIN_NT_RGN_STOP

3 ... new NOx regeneration requests are inhibited up to the end of full load phase or high engine load phase;
a running NOx regeneration is not broken

Signal flow diagram:

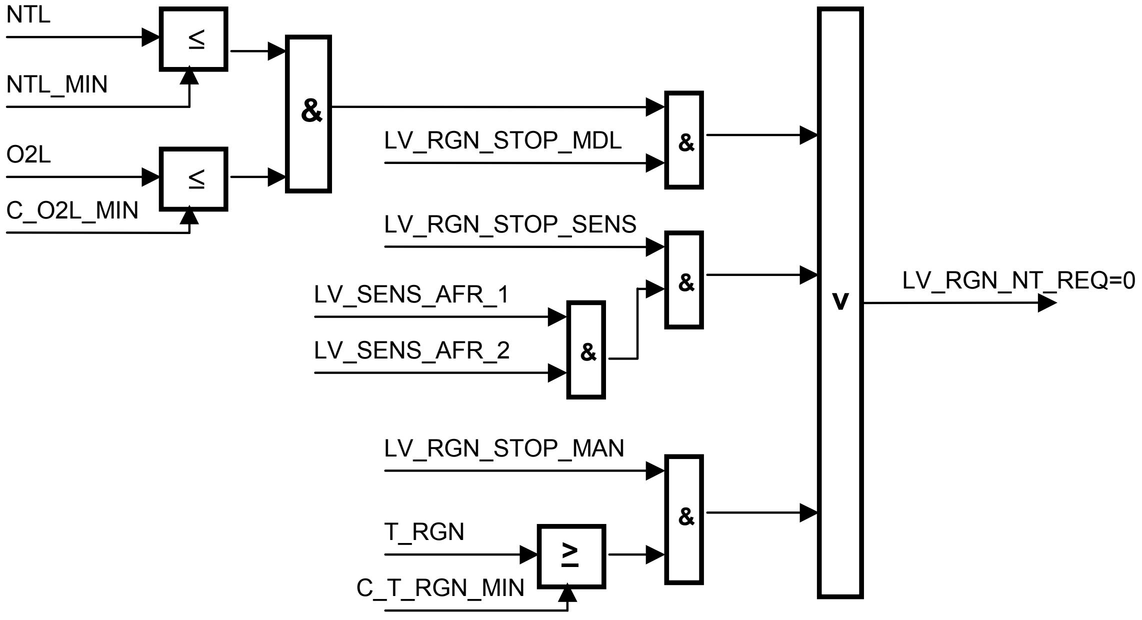


Figure 87: check for resetting the regeneration requests

Application conditions:

Initialisation: --

Recurrence: 10ms

Activation: --

Deactivation: --

Formula section:

If (75) NTL <= NTL_MIN **and** O2L <= C_O2L_MIN

and LV_TOUT_AFR = 0

```
then (75)   LV_MDL_AFR = 1
endif (75)


if (76)
    T_RGN >= IP_T_RGN_MAX (FLOW_NT_MMV, NT_AGI_S_RED)                    OR
     ( LV_SENS_AFR_1 = 1                                                AND
      ( T_RGN_2 > IP_T_MAX_RGN_2_STOP(NTL_RGN_ST, NTL_DEC_INT_1) OR
        T_RGN_2 > IP_T_MAX_RGN_2_STOP(NTL_RGN_ST, NTL_DEC_INT_2)
      )
     )                                                                   OR
    NTL_DEC_INT_1 > IP_NTL_DEC_INT_MAX(NTL_RGN_ST)                       OR
    NTL_DEC_INT_2 > IP_NTL_DEC_INT_MAX(NTL_RGN_ST)
then (76)
    if (121)   LV_INH_NT_RGN_STOP_MDL_DIAG[k] = 0        % valid if all [k] = 0
    then (121)  LV_TOUT_AFR = 1
    else (121)
        if (122)   T_RGN >= C_T_RGN_MAX_DIAG
        then (122)     LV_TOUT_AFR = 1
        endif (122)
    endif (121)
endif (76)


if (79)      ((LV_SENS_AFR_1 and LV_SENS_AFR_2)   and      % stop from sensor
             LV_RGN_STOP_SENS)                             or
             (LV_MDL_AFR = 1                      and      % stop from model
             LV_RGN_STOP_MDL)                              or
             (LV_TOUT_AFR = 1                     and      % stop from time counter
             LV_RGN_STOP_TOUT)                             or
             (C_NT_RGN_STOP_FL_SEL = 1            and      % break by full load
               (LV_FL = 1 or LV_LOAD_H = 1)       and
              MAF_INT_FL_ACT >= C_MAF_INT_FL_RGN_STOP_MIN)       or
             LV_NTL_DEC_INT_AFR                   % stop from NTL_DEC_INT ratio
then (79)    LV_RGN_CDN                   = 0
             LV_RGN_REQ_NTLD              = 0
             LV_RGN_REQ_NTLD_AFS          = 0
             LV_RGN_REQ_NOX_SENS_TMP[i]   = 0              % reset all [i] requests
             LV_RGN_REQ_NOX_SENS          = 0
             O2L                          = 0
             LV_RGN_NT_REQ                = 0
             LV_ACT_INT_PUC_AFL_i         = 0              % reset all [i] requests
             NTL                          = NTL_MIN
             NTL_H                        = NTL_MIN
             LV_NT_RGN_STOP_PUC           = 0
```

```
endif (79)

if (126)    ( C_NT_RGN_STOP_PUC_SEL = 1                         AND
              LV_PUC = 1                                        AND
              MAF_INT_PUC_ACT(n-1) < MAF_INT_PUC_ACT(n)         AND
              MAF_INT_PUC_ACT(n) >= C_MAF_INT_PUC_RGN_STOP_MIN )      OR
            ( C_NT_RGN_STOP_PUC_SEL = 2                         AND
              LV_CLU_SWI = 0                                    AND
              LV_PUC = 1                                        AND
              MAF_INT_PUC_ACT(n-1) < MAF_INT_PUC_ACT(n)         AND
              MAF_INT_PUC_ACT(n) >= C_MAF_INT_PUC_RGN_STOP_MIN )      OR
            ( LC_NT_RGN_STOP_IS = 1                             AND
              LV_IS = 1                                         AND
              T_IS >= C_T_IS_RGN_STOP_MIN )                           OR
            ( C_NT_RGN_STOP_FL_SEL = 2                          AND
              ( LV_FL = 1 or LV_LOAD_H = 1)                     AND
              MAF_INT_FL_ACT >= C_MAF_INT_FL_RGN_STOP_MIN )
            % Calibration remark: Consider failure reaction on open wire at clutch switch circuit!
then (126)  % break off NOx regeneration because of higher priority of PUC
            LV_RGN_CDN                     = 0
            LV_RGN_REQ_NTLD                = 0
            LV_RGN_REQ_NTLD_AFS            = 0
            LV_RGN_REQ_NOX_SENS_TMP[i]     = 0          % reset all [i] requests
            LV_RGN_REQ_NOX_SENS            = 0
            LV_RGN_NT_REQ                  = 0
            LV_ACT_INT_PUC_AFL_i           = 0          % reset all [i] requests
            LV_NT_RGN_STOP_PUC             = 1
            if (128)    LC_NT_RGN_STOP_INI_NTL_O2L = 1
            then (128)  NTL                = NTL_MIN
                        NTL_H              = NTL_MIN
                        O2L                = 0
            else (128)  NTL                = NTL(n-1)
                        NTL_H              = NTL_H(n-1)
                        O2L                = O2L(n-1)
            endif (128)
endif (126)

if (116)  NC_NT_NR = 2 and LC_SENS_AFR_MOD = 1
then (116)
             if (117)    LV_SENS_AFR_1 = 1
             then (117)  LV_ACT_INT_PUC_AFL_1 = 0
                         LV_ACT_INT_PUC_AFL_2 = 0
                         LV_CAT_PURGE_REQ_POST_AFL[1] = 0
                         LV_CAT_PURGE_REQ_POST_AFL[2] = 0
```

```
            endif (117)
else (116)
            if (118)  LV_SENS_AFR_i = 1
            then (118)   LV_ACT_INT_PUC_AFL_i = 0
                         LV_CAT_PURGE_REQ_POST_AFL[i] = 0
            endif (118)
endif (116)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_RGN_CTL	1	0H 1H 2H 3H	SENSOR MODEL SENSOR_MODEL TIME	1	[-]
regeneration stop mode					
C_STATE_SENS_AFR[NC_NT_NR]	1	0H 1H 2H 3H 4H	VLS_DOWN VLS_NOX_SENS LAMB_NOX_SENS VLS_NT_DOWN NONE	1	[-]
bench-selective mode to determine rich exhaust gas downstream NOx trap					
C_STATE_SENS_AFR_TRIM[NC_NT_NR]	1	0H 1H 2H 3H 4H	VLS_DOWN VLS_NOX_SENS LAMB_NOX_SENS VLS_NT_DOWN NONE	1	[-]
bench-selective mode to determine trim threshold reached by downstream sensor					
C_T_AFL_RGN_REQ_NTLD	1	0...FFFFH	0...1310.7	0.02	[s]
Minimum duration of lean operation before permitting a regeneration request from the model					
IP_T_AFL_MIN	8	0...FFFFH	0...1310.7	0.02	[s]
LDPM_VS_1_NOXM	8	0...FFH	0...255	1	[km/h]
minimum duration of lean operation before permitting a regeneration					
IP_T_AFL_MIN_NT_RGN_STOP	8	0...FFFFH	0...1310.7	0.02	[s]
LDPM_VS_1_NOXM	8	0...FFH	0...255	1	[km/h]
minimum duration of lean operation before permitting a NOx regeneration after a broken NOx regeneration					
C_VLS_NOX_SENS_RGN_STOP	1	0...578H	-200...1200	1	[mV]
binary NOx sensor signal threshold for determining rich exhaust gas					
C_VLS_NOX_SENS_AFR_TRIM	1	0...578H	-200...1200	1	[mV]
binary NOx sensor signal threshold for determining rich trim conditions					
C_LAMB_NOX_SENS_RGN_STOP	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
linear NOx sensor signal threshold for determining rich exhaust gas					
C_LAMB_NOX_SENS_AFR_TRIM	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
linear NOx sensor signal threshold for determining rich trim conditions					
C_VLS_DOWN_AFR_TRIM	1	0...3FFH	0...4.99511	4.8828e-3	[V]
binary lambda sensor signal threshold for determining rich trim conditions					
C_NOX_NS_AD_RGN_STOP	1	0...05DCH	0...1500	1	[ppm]
linear NOx sensor signal threshold for determining rich trim conditions					
C_O2L_MIN	1	0...FFFFH	0...10485.6	0.16	[mg]
minimum NOx trap oxygen loading for stopping a regeneration					
C_SUM_O2L_NTL_MAX	1	0...FFFFH	0...10485.6	0.16	[mg]
threshold of oxygen and NOx loading for requesting a NOx catalyst purge					
C_NTL_TRIG_AFR_AFL	1	0...FFFFH	0...10485.6	0.16	[mg]
minimum NOx trap NOx loading for triggering the transition from rich to lean operation					
C_SUM_NTL_CYC_SP	1	0...FFFFH	0...65535	1	[-]
NOx trap regeneration cycle counter initialisation value					
C_TNT_MDL_INH_PURGE	1	0...FFH	0...1020	4	[°C]
NOx trap temperature threshold for inhibiting a catalyst purge					
C_VS_INH_PURGE	1	0...FFH	0...255	1	[km/h]
Vehicle speed threshold for inhibiting a catalyst purge					
C_CRLC_FLOW_NT	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for calculation of filtered exhaust gas flow					
C_NOX_OUT_MES_INT_MAX	1	0...FFFFH	0...10485.6	0.16	[mg]
Threshold for integrated downstream emission (meas. by NOx sensor) for requesting a regeneration					
IP_NTL_MIN	8	0...FFFFH	0...10485.6	0.16	[mg]
LDPM_TNT_MDL_NOX	8	0...FFH	0...1020	4	[°C]
minimum NOx trap NOx loading for stopping a regeneration					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_NTLD_MAX_AT	12*8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_TQI_REQ_FAST_IP_NTLD_AT	12	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_NTLD_AT	8	0...FFH	0...8160	32	[rpm]
Maximum NOx catalyst loading for requesting a regeneration - AT vehicle					
IP_NTLD_MAX_MT	12*8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_TQI_REQ_FAST_IP_NTLD_MT	12	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_NTLD_MT	8	0...FFH	0...8160	32	[rpm]
Maximum NOx catalyst loading for requesting a regeneration - MT vehicle					
IP_NTLD_MAX_ADJ_AT	8*8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_TQI_REQ_FAST_IP_NTLD_ADJ_AT	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_NTLD_ADJ_AT	8	0...FFH	0...8160	32	[rpm]
Maximum NOx catalyst loading for requesting a regeneration (adjusted for special purposes) - AT vehicle					
IP_NTLD_MAX_ADJ_MT	8*8	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_TQI_REQ_FAST_IP_NTLD_ADJ_MT	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_NTLD_ADJ_MT	8	0...FFH	0...8160	32	[rpm]
Maximum NOx catalyst loading for requesting a regeneration (adjusted for special purposes) - MT vehicle					
ID_VLS_DOWN_RGN_STOP	4	0...3FFH	0...4.99511	4.8828e-3	[V]
LDP_NTLD_RGN_ST_ID_VLS_DOWN	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
downstream binary lambda sensor signal threshold for determining rich exhaust gas					
IP_T_NT_HOM_DEC	4*4	0...7FH	0...1.27	0.01	[s]
LDPM_MAF_KGH_IP_T_NT_HOM	4	0...FFH	0...2040	8	[kg/h]
LDPM_LAMB_AV_COR_IP_T_NT_HOM	4	0...7FFFH	0...31.99902	0.9766e-3	[-]
Weighted decrement for the homogeneous time counter					
IP_T_NT_HOM_INC	4*4	0...7FH	0...1.27	0.01	[s]
LDPM_MAF_KGH_IP_T_NT_HOM	4	0...FFH	0...2040	8	[kg/h]
LDPM_LAMB_AV_COR_IP_T_NT_HOM	4	0...7FFFH	0...31.99902	0.9766e-3	[-]
Weighted increment for the homogeneous time counter					
LC_NOX_SENS	1	0...1H	0...1	1	[-]
Logical constant for the existence of a NOx-Sensor					
LC_NOX_SENS_RGN	1	0...1H	0...1	1	[-]
switch to exclusively use the NOx sensor for requesting a regeneration					
LC_SUM_NTL_CYC_NEW	1	0...1H	0...1	1	[-]
Initialisation of regeneration cycle counter					
LC_NTLD_MAX_AFS	1	0...1H	0...1	1	[-]
Switch between NTLD and NTLD_MDL for LV_RGN_REQ_NTLD_AFS					
IP_T_RGN_MAX	4*4	0...FFFFH	0...655.35	0.01	[s]
LDP_FLOW_NT_MMV_IP_T_RGN_MAX	4	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDPM_NT_AGI_S_RED	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
regeneration time for manual regeneration duration selection					
IP_NTLD_MAX_AFS_AT	8*4	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_TQI_REQ_FAST_IP_NTLD_AFS_AT	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_NTLD_MAX_AFS_AT	4	0...FFH	0...8160	32	[rpm]
Maximum NOx trap loading degree for requesting a regeneration before lambda=1.0 operation (for automatic transmission)					
LC_INH_RGN_PU	1	0...1H	0...1	1	[-]
Switch to forbid regeneration during PU					
IP_NTLD_MAX_AFS_MT	8*4	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_TQI_REQ_FAST_IP_NTLD_AFS_MT	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_NTLD_MAX_AFS_MT	4	0...FFH	0...8160	32	[rpm]
Maximum NOx trap loading degree for requesting a regeneration before lambda=1.0 operation (for manual transmission)					
IP_NOX_SENS_MAX_MT	6*8	0...05DCH	0...1500	1	[ppm]
LDP_FLOW_NT_MMV_IP_NOX_SENS_MT	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDP_N_32_IP_NOX_SENS_MAX_MT	8	0...FFH	0...8160	32	[rpm]

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
threshold of the NOx sensor signal for requesting a regeneration for MT					
IP_FAC_NOX_SENS_MAX	4*6	0...FFH	0...1.99218	0.0078125	[-]
LDPM_NT_AGI_S_RED	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_TNT_MDL_MV_IP_FAC_NOX_SENS	6	0...FFFFH	0...1023.98437	0.015625	[°C]
Factor to correct threshold of the NOx sensor signal for requesting a regeneration					
C_FAC_NTLD_MAX_AFS_ERR_EGR	1	0...FFH	0...1.99218	0.0078125	[-]
Factor to correct homogen request in case of EGR_ERR					
IP_NOX_SENS_MAX_AT	6*8	0...05DCH	0...1500	1	[ppm]
LDP_FLOW_NT_MMV_IP_NOX_SENS_AT	6	0...FFFFH	0...2047.96875	0.03125	[kg/h]
LDP_N_32_IP_NOX_SENS_MAX_AT	8	0...FFH	0...8160	32	[rpm]
threshold of the NOx sensor signal for requesting a regeneration for AT					
C_NOX_NS_AD_DIF_MON	1	0...05DCH	0...1500	1	[ppm]
Maximum deviation between model and NOx-Sensor					
C_NOX_SENS_MAX_OFS	1	0...05DCH	0...1500	1	[ppm]
Threshold offset of the NOx sensor signal for requesting a regeneration					
LC_SENS_AFR_MOD	1	0...1H	0...1	1	[-]
Selection of calculation method for LV_SENS_AFR_i					
IP_VLS_DIST_GRD_NEG	6	0...578H	-200...1200	1	[mV]
LDP_VLS_MAX_TMP_IP_VLS_DIST	6	0...578H	-200...1200	1	[mV]
Difference between local maximum and current value for calculation of negative gradient					
C_VLS_DIST_GRD_POS	1	0...578H	-200...1200	1	[mV]
Difference between local minimum and current value for calculation of positive gradient					
C_VLS_MAX_RGN_1_STOP	1	0...578H	-200...1200	1	[mV]
Threshold value for end of first regeneration					
C_VLS_MAX_RGN_2_STOP	1	0...578H	-200...1200	1	[mV]
Threshold value for end of second regeneration					
C_VLS_GRD_NEG_VLS_SET	1	0...578H	-200...1200	1	[mV]
Negative gradient for set of LV_SENS_AFR_2 with VLS signal					
IP_T_MAX_RGN_2_STOP	6*6	0...FFFFH	0...655.35	0.01	[s]
LDP_NTL_RGN_ST_IP_T_MAX_RGN	6	0...FFFFH	0...10485.6	0.16	[mg]
LDP_NTL_DEC_INT_IP_T_MAX_RGN	6	0...FFFFH	0...10485.6	0.16	[mg]
Maximum time duration of second regeneration					
IP_NTL_DEC_INT_MAX	6	0...FFFFH	0...10485.6	0.16	[mg]
LDP_NTL_RGN_ST_IP_NTL_DEC_INT	6	0...FFFFH	0...10485.6	0.16	[mg]
Maximum amount of regeneration agent					
C_DIST_AFL_MIN	1	0...FFFFH	0...6553.5	0.1	[km]
Minimum mileage for calculation of RATIO_NOX_OUT_INT_DIST					
C_RATIO_NOX_OUT_INT_DIST_MAX	1	0...FFFFH	0...6.5535	0.0001	[g/km]
Maximum ratio of NOx out integral to mileage at current lean phase; threshold for start of a NOx regeneration					
C_NOX_COR_INT_MIN	1	0...FFFFH	0...10485.6	0.16	[mg]
Minimum NOx raw emission integral value for calculation of RATIO_NOX					
C_RATIO_NOX_MAX	1	0...FFFFH	0...99.99847	1.5259e-3	[%]
Maximum ratio of NOx out integral to NOx raw emission integral; threshold for start of a NOx regenration					
C_T_RGN_MAX_DIAG	1	0...FFFFH	0...655.35	0.01	[s]
Maximum regeneration duration if sensor test is activated					
C_NT_RGN_STOP_PUC_SEL	1	0...2H	0...2	1	[-]
Method selection for break off a running NOx regeneration if PUC has higher priority					
LC_NT_RGN_STOP_IS	1	0...1H	0...1	1	[-]
Method selection for break off a running NOx regeneration if idle speed has higher priority					
C_MAF_INT_PUC_RGN_STOP_MIN	1	0...FFFFH	0...2912.66666	0.0444444	[g]
Minimum air mass flow at PUC after which a NOx regeneration break off is allowed					
C_T_IS_RGN_STOP_MIN	1	0...FFFFH	0...655.35	0.01	[s]
Minimum idle speed duration after which a NOx regeneration break off is allowed					
LC_NT_RGN_STOP_INI_NTL_O2L	1	0...1H	0...1	1	[-]
Method selection for handling of NTL and O2L at break off of a NOx regeneration by PUC or idle speed					
C_NT_RGN_STOP_FL_SEL	1	0...3H	0...3	1	[-]
Selection of NOx regeneration handling in kind of full load					
C_MAF_INT_FL_RGN_STOP_MIN	1	0...FFFFH	0...6553.5	0.1	[g]
Minimum air mass flow at full load after which a NOx regeneration break is allowed					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_NTLD_MAX_VS_AT	6*6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_VS_IP_NTLD_MAX_VS_AT	6	0...FFH	0...255	1	[km/h]
LDP_NT_AGI_S_RED_IP_NTLD_VS_AT	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx catalyst loading for requesting a NOx regeneration depending on vehicle speed - AT vehicle					
IP_NTLD_MAX_VS_MT	6*6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_VS_IP_NTLD_MAX_VS_MT	6	0...FFH	0...255	1	[km/h]
LDP_NT_AGI_S_RED_IP_NTLD_VS_MT	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx catalyst loading for requesting a NOx regeneration depending on vehicle speed - MT vehicle					
IP_T_AFL_MIN_AFS	8	0...FFFFH	0...1310.7	0.02	[s]
LDP_VS_IP_T_AFL_MIN_AFS	8	0...FFH	0...255	1	[km/h]
Minimum duration of lean operation before permitting a regeneration at a switch to Lambda=1.0 operation					
C_TNT_MDL_PURGE_REQ_MAX	1	0...FFH	0...1020	4	[°C]
Maximum NOx catalyst temperature threshold for set of catalyst purge request					
C_VS_PURGE_REQ_MAX	1	0...FFH	0...255	1	[km/h]
Maximum vehicle speed threshold for set of catalyst purge request					
LC_VLS_NS_VLD_RGN_STOP_MDL	1	0...1H	0...1	1	[-]
NOx regeneration stop by model at invalid binary Lambda signal of NOx sensor					
LC_NTL_DEC_INT_AFR_SENS_AFR_2	1	0...1H	0...1	1	[-]
Enables the set of LV_SENS_AFR_2 by LV_NTL_DEC_INT_AFR					

53.6.3.4 Request of lambda =1 operation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TNT_MDL_MAX	V	0...FFH	0...1020	4	[°C]
Maximum NOx trap temperature for lean operation					
TNT_MDL_MIN	V	0...FFH	0...1020	4	[°C]
Minimum NOx trap temperature for lean operation					
LV_NT_AFS_REQ_PRED	V	0...1H	0...1	1	[-]
Prediction result for request of homogeneous stoichiometric operation					
T_NT_AFS_REQ_PRED	V	0...FFFFH	0...6553.5	0.1	[s]
Duration of active homogeneous stoichiometric operation request due to high predicted NOx row emissions					
TQI_MMV_NOX	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Filtered actual indicated engine torque for NOx emission aftertreatment					
NOX_S_PRED_THD	V	0...FFFFH	0...1023.98437	0.015625	[mg/s]
NOx threshold for set of Lambda = 1.0 operation request - stratified prediction					
NOX_S_PRED_HYS	V	0...FFFFH	-1024... 1023.96875	0.03125	[mg/s]
NOx hysteresis for reset of Lambda = 1.0 operation request - stratified prediction					
NOX_HOMS_PRED_THD	V	0...FFFFH	0...1023.98437	0.015625	[mg/s]
NOx threshold for set of Lambda = 1.0 operation request - homogeneous lean prediction					
NOX_HOMS_PRED_HYS	V	0...FFFFH	-1024... 1023.96875	0.03125	[mg/s]
NOx hysteresis for reset of Lambda = 1.0 operation request - homogeneous lean prediction					
VS_MMV_NOX	V	0...7FFFH	0...255.99218	0.0078125	[km/h]
Filtered vehicle speed for detection of transient driving conditions					
VS_DIF_NOX	V	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Vehicle speed difference for detection of transient driving conditions					
VS_DIF_THD	V	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Vehicle speed difference threshold for set of Lambda=1.0 operation request					
VS_DIF_HYS	V	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Vehicle speed difference hysteresis for reset of Lambda=1.0 operation request					

Input data:

LV_NT_AFS_REQ_AGI	TNT_MDL_H	TNT_MDL_L	LV_SO2P_REQ
	LV_INH_NT_AFL		
TNT_MDL_MAX_OFS		LV_TNT_MDL_MAX_OFS	LV_IS
NOX_COR_S_PRED	NOX_COR_HOMS_PRED		TQI_AV
NT_AGI_S_RED			

FUNCTION DESCRIPTION:

General information:

This block is executed in any STATE_NOX except PASSIVE.

Depending on various conditions the lean operation of the engine can be forbidden using LV_NT_AFS_REQ. This flag is an input to the combustion manager. There, it forces the combustion manager to switch to lambda=1 homogeneous mode.

All conditions leading to stoichiometric homogeneous operation of all modules of the NOx trap functions (including NOx catalyst aging and NOx catalyst desulfation) are handled by this subroutine in order to get only one flag as interface to the combustion manager. These conditions are:

- catalyst temperature too high
- catalyst temperature too low
- aging state too bad
- desulfation introduced
- other reasons considered in the corresponding application incidences module

Application conditions:

Initialisation: at reset:

TNT_MDL_MAX = 0
TNT_MDL_MIN = 0
LV_NT_AFS_REQ_PRED = 0
T_NT_AFS_REQ_PRED = 0
TQI_MMV_NOX = 0
VS_MMV_NOX = 0
VS_DIF_NOX = 0
VS_DIF_THD = 0
VS_DIF_HYS = 0

Recurrence: 100 ms

Activation: --

Deactivation: --

Formula section:

```
TQI_MMV_NOX(n) = TQI_AV * C_CRLC_TQI_MMV_NOX +
                 TQI_MMV_NOX(n-1)  * (1 - C_CRLC_TQI_MMV_NOX)
VS_MMV_NOX(n) = VS * C_CRLC_VS_MMV_NOX +
                 VS_MMV_NOX(n-1)  * (1 - C_CRLC_VS_MMV_NOX)
VS_DIF_NOX = VS – VS_MMV_NOX

if (124)     LV_NT_AFS_REQ_PRED = 1  AND  LV_RGN_NT_REQ = 0
then (124)   increment T_NT_AFS_REQ_PRED
else (124)   T_NT_AFS_REQ_PRED = 0
endif (124)

if (134)     LV_AT = 1
then (134)   TNT_MDL_MAX = IP_TNT_MDL_MAX_AT(NT_AGI_S_RED, VS) +
                           TNT_MDL_MAX_OFS
             if (88)      LV_IS = 1
             then (88)    TNT_MDL_MIN = IP_TNT_MDL_MIN_IS_AT(NT_AGI_S_RED)
             else (88)    TNT_MDL_MIN = IP_TNT_MDL_MIN_AT(NT_AGI_S_RED)
             endif (88)
             NOX_S_PRED_THD = IP_NOX_S_AFS_REQ_AT(TNT_MDL_L, NT_AGI_S_RED)
             NOX_S_PRED_HYS = IP_NOX_S_AFS_REQ_HYS_AT(N_32, TQI_MMV_NOX)
             NOX_HOMS_PRED_THD =
                           IP_NOX_HOMS_AFS_REQ_AT(TNT_MDL_L, NT_AGI_S_RED)
             NOX_HOMS_PRED_HYS =
```

```
                        IP_NOX_HOMS_AFS_REQ_HYS_AT(N_32, TQI_MMV_NOX)
            VS_DIF_THD = IP_VS_DIF_THD_BAS_AT(TNT_MDL_L, NT_AGI_S_RED) +
                        IP_VS_DIF_THD_COR_AT(VS_MMV_NOX)
            VS_DIF_HYS = IP_VS_DIF_HYS_AT(VS_MMV_NOX)

else (134)  TNT_MDL_MAX = IP_TNT_MDL_MAX_MT(NT_AGI_S_RED, VS) +
                        TNT_MDL_MAX_OFS
            if (138)    LV_IS = 1
            then (138)  TNT_MDL_MIN = IP_TNT_MDL_MIN_IS_MT(NT_AGI_S_RED)
            else (138)  TNT_MDL_MIN = IP_TNT_MDL_MIN_MT(NT_AGI_S_RED)
            endif (138)
            NOX_S_PRED_THD = IP_NOX_S_AFS_REQ_MT(TNT_MDL_L, NT_AGI_S_RED)
            NOX_S_PRED_HYS = IP_NOX_S_AFS_REQ_HYS_MT(N_32, TQI_MMV_NOX)
            NOX_HOMS_PRED_THD =
                        IP_NOX_HOMS_AFS_REQ_MT(TNT_MDL_L, NT_AGI_S_RED)
            NOX_HOMS_PRED_HYS =
                        IP_NOX_HOMS_AFS_REQ_HYS_MT(N_32, TQI_MMV_NOX)
            VS_DIF_THD = IP_VS_DIF_THD_BAS_MT(TNT_MDL_L, NT_AGI_S_RED) +
                        IP_VS_DIF_THD_COR_MT(VS_MMV_NOX)
            VS_DIF_HYS = IP_VS_DIF_HYS_MT(VS_MMV_NOX)
endif (134)


if (119)    LV_RGN_NT_REQ = 1
    and     ( NOX_COR_S_PRED > NOX_S_PRED_THD
              or    NOX_COR_HOMS_PRED > NOX_HOMS_PRED_THD )
    and     ( VS_DIF_NOX > VS_DIF_THD )
then (119)  LV_NT_AFS_REQ_PRED = 1
else (119)
        if (120)  T_NT_AFS_REQ_PRED > C_T_NT_AFS_REQ_PRED_MAX
            or ( LV_HOM_AFS_ACT = 1
                and  LV_NT_AFS_REQ_PRED = 1
                and  ( ( NOX_COR_S_PRED <=
                            ( NOX_S_PRED_THD - NOX_S_PRED_HYS )
                        and  NOX_COR_HOMS_PRED <=
                            ( NOX_HOMS_PRED_THD - NOX_HOMS_PRED_HYS )
                       )
                     or  ( VS_DIF_NOX <= ( VS_DIF_THD - VS_DIF_HYS ) )
                     )
               )
        then (120)   LV_NT_AFS_REQ_PRED = 0
        endif (120)
endif (119)


if (86)     TNT_MDL_H > TNT_MDL_MAX                      % cat. temp. too high
  or        TNT_MDL_L < TNT_MDL_MIN - C_TNT_MDL_MIN_HYS  % tnt too low
```

```
   or        LV_NT_AFS_REQ_AGI      = 1                 % aging bad
   or        LV_SO2P_REQ            = 1                 % SO2P introduced
   or        LV_NT_AFS_REQ_PRED     = 1
   or        LV_INH_NT_AFL          = 1                 % flag from appl. inc. module
   or        ( LV_NS_SHIFT_CMB_INT_REQ = 1
               and  LC_NT_AFS_REQ_NS_SHIFT = 1 )
 then (86)   LV_NT_AFS_REQ = 1                          % lambda=1 request
 endif (86)

 if (87)     ( TNT_MDL_H < TNT_MDL_MAX - C_TNT_MDL_MAX_HYS
   and       TNT_MDL_L >= TNT_MDL_MIN
   and       LV_NT_AFS_REQ_AGI       = 0
   and       LV_SO2P_REQ             = 0
   and       LV_NT_AFS_REQ_PRED      = 0
   and       LV_INH_NT_AFL           = 0
   and       ( LV_NS_SHIFT_CMB_INT_REQ = 0
               or  LC_NT_AFS_REQ_NS_SHIFT = 0 ) )
   or        ( LV_TNT_MDL_MAX_OFS = 1 and
             TNT_MDL_H < TNT_MDL_MAX - C_TNT_MDL_MAX_HYS )
 then (87)   LV_NT_AFS_REQ    = 0
 endif (87)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TNT_MDL_MAX_HYS	1	0...FFH	0...1020	4	[°C]
Temperature hysteresis for lambda=1-mode deactivation due to high temperature					
C_TNT_MDL_MIN_HYS	1	0...FFH	0...1020	4	[°C]
Temperature hysteresis for lambda=1-mode deactivation due to low temperature					
IP_TNT_MDL_MAX_AT	6*6	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_TNT_MAX_AT	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_VS_IP_TNT_MAX_AT	6	0...FFH	0...255	1	[km/h]
Maximum NOx catalyst temperature for lean operation - AT vehicle					
IP_TNT_MDL_MAX_MT	6*6	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_TNT_MAX_MT	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_VS_IP_TNT_MAX_MT	6	0...FFH	0...255	1	[km/h]
Maximum NOx catalyst temperature for lean operation - MT vehicle					
IP_TNT_MDL_MIN_AT	4	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_TNT_MIN_AT	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
Minimum NOx catalyst temperature for lean operation - AT vehicle					
IP_TNT_MDL_MIN_MT	4	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_TNT_MIN_MT	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
Minimum NOx catalyst temperature for lean operation - MT vehicle					
IP_TNT_MDL_MIN_IS_AT	4	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_TNT_IS_AT	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
Minimum NOx catalyst temperature for lean operation at idle speed - AT vehicle					
IP_TNT_MDL_MIN_IS_MT	4	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_TNT_IS_MT	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
Minimum NOx catalyst temperature for lean operation at idle speed - MT vehicle					
IP_NOX_S_AFS_REQ_AT	8*6	0...FFFFH	0...1023.98437	0.015625	[mg/s]
LDP_TNT_MDL_L_IP_NOX_S_AT	8	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_NOX_S_AT	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx threshold for set of Lambda = 1.0 operation request - stratified prediction - AT vehicle					
IP_NOX_S_AFS_REQ_MT	8*6	0...FFFFH	0...1023.98437	0.015625	[mg/s]
LDP_TNT_MDL_L_IP_NOX_S_MT	8	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_NOX_S_MT	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx threshold for set of Lambda = 1.0 operation request - stratified prediction - MT vehicle					
IP_NOX_S_AFS_REQ_HYS_AT	6*6	0...FFFFH	-1024... 1023.96875	0.03125	[mg/s]
LDP_N_32_IP_NOX_S_AT	6	0...FFH	0...8160	32	[rpm]
LDP_TQI_MMV_NOX_IP_NOX_S_AT	6	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
NOx hysteresis for reset of Lambda = 1.0 operation request - stratified prediction - AT vehicle					
IP_NOX_S_AFS_REQ_HYS_MT	6*6	0...FFFFH	-1024... 1023.96875	0.03125	[mg/s]
LDP_N_32_IP_NOX_S_MT	6	0...FFH	0...8160	32	[rpm]
LDP_TQI_MMV_NOX_IP_NOX_S_MT	6	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
NOx hysteresis for reset of Lambda = 1.0 operation request - stratified prediction - MT vehicle					
IP_NOX_HOMS_AFS_REQ_AT	8*6	0...FFFFH	0...1023.98437	0.015625	[mg/s]
LDP_TNT_MDL_L_IP_NOX_HOMS_AT	8	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_NOX_HOMS_AT	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx threshold for set of Lambda = 1.0 operation request - homogeneous lean prediction - AT vehicle					
IP_NOX_HOMS_AFS_REQ_MT	8*6	0...FFFFH	0...1023.98437	0.015625	[mg/s]
LDP_TNT_MDL_L_IP_NOX_HOMS_MT	8	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_NOX_HOMS_MT	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx threshold for set of Lambda = 1.0 operation request - homogeneous lean prediction - MT vehicle					
IP_NOX_HOMS_AFS_REQ_HYS_AT	6*6	0...FFFFH	-1024... 1023.96875	0.03125	[mg/s]
LDP_N_32_IP_NOX_HOMS_AT	6	0...FFH	0...8160	32	[rpm]
LDP_TQI_MMV_NOX_IP_NOX_HOMS_AT	6	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
NOx hysteresis for reset of Lambda = 1.0 operation request - homogeneous lean prediction - AT vehicle					

IP_NOX_HOMS_AFS_REQ_HYS_MT	6*6	0...FFFFH	-1024... 1023.96875	0.03125	[mg/s]
LDP_N_32_IP_NOX_HOMS_MT	6	0...FFH	0...8160	32	[rpm]
LDP_TQI_MMV_NOX_IP_NOX_HOMS_MT	6	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
NOx hysteresis for reset of Lambda = 1.0 operation request - homogeneous lean prediction - MT vehicle					
C_T_NT_AFS_REQ_PRED_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum duration of of active AFS request due to high predicted NOx row emissions					
C_CRLC_TQI_MMV_NOX	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for calculation of filtered actual indicated engine torque					
C_CRLC_VS_MMV_NOX	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for calculation of filtered vehicle speed					
IP_VS_DIF_THD_BAS_AT	8*8	0...FFFFH	-256...255.99218	0.0078125	[km/h]
LDP_TNT_MDL_L_IP_VS_THD_AT	8	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_VS_THD_AT	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
Vehicle speed difference threshold for set of Lambda=1.0 operation request - AT vehicle					
IP_VS_DIF_THD_COR_AT	8	0...FFFFH	-256...255.99218	0.0078125	[km/h]
LDP_VS_MMV_NOX_IP_VS_COR_AT	8	0...7FFFH	0...255.99218	0.0078125	[km/h]
Vehicle speed difference threshold correction for set of Lambda=1.0 operation request - AT vehicle					
IP_VS_DIF_HYS_AT	8	0...FFFFH	-256...255.99218	0.0078125	[km/h]
LDP_VS_MMV_NOX_IP_VS_HYS_AT	8	0...7FFFH	0...255.99218	0.0078125	[km/h]
Vehicle speed difference hysteresis for reset of Lambda=1.0 operation request - AT vehicle					
IP_VS_DIF_THD_BAS_MT	8*8	0...FFFFH	-256...255.99218	0.0078125	[km/h]
LDP_TNT_MDL_L_IP_VS_THD_MT	8	0...FFH	0...1020	4	[°C]
LDP_NT_AGI_S_RED_IP_VS_THD_MT	8	0...FFFFH	0...0.99998	0.0153e-3	[-]
Vehicle speed difference threshold for set of Lambda=1.0 operation request - MT vehicle					
IP_VS_DIF_THD_COR_MT	8	0...FFFFH	-256...255.99218	0.0078125	[km/h]
LDP_VS_MMV_NOX_IP_VS_COR_MT	8	0...7FFFH	0...255.99218	0.0078125	[km/h]
Vehicle speed difference threshold correction for set of Lambda=1.0 operation request - MT vehicle					
IP_VS_DIF_HYS_MT	8	0...FFFFH	-256...255.99218	0.0078125	[km/h]
LDP_VS_MMV_NOX_IP_VS_HYS_MT	8	0...7FFFH	0...255.99218	0.0078125	[km/h]
Vehicle speed difference hysteresis for reset of Lambda=1.0 operation request - MT vehicle					
LC_NT_AFS_REQ_NS_SHIFT	1	0...1H	0...1	1	[-]
Activation of Lambda=1.0 operation request by for NOx sensor shift diagnosis by LV_NT_AFS_REQ					

53.7 NOx catalyst management (Application Incidences)

53.7.1 Inhibiting the NOx catalyst management

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_NT_ACT	O/V	0...1H	0...1	1	[-]
flag inhibiting the NOx catalyst management					
T_IPLSL_VLD_DLY	V	0...FFH	0...25.5	0.1	[s]
Timer for hiding of short LV_IPLSL_VLD[i] = 0 phases at a MSA start					

Input data:

	LV_ERR_CAM	LV_ERR_MAF	LV_ERR_TCO
	NC_CBK_EX_NR	NC_NOX_SENS_CONF	LV_IPLSL_VLD[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

Number of NOx sensors:	NC_NOX_SENS_CONF	k = 1...NC_NOX_SENS_CONF
Number of lambda sensors:	NC_CBK_EX_NR	i = 1...NC_CBK_EX_NR

Typical configurations:
[NC_CBK_EX_NR:NC_NOX_SENS_CONF] =[1:1], [2:2], [2:1]

Description:

The timer T_IPLSL_VLD_DLY is limited to its maximum value. An overflow is not allowed.

Application conditions:

Initialisation: at reset

LV_INH_NT_ACT = 0
T_IPLSL_VLD_DLY = 0

Recurrence: 100 ms

Activation: at every engine state

Deactivation: -

Formula section:

```
if      LV_IPLSL_VLD[i] = 0                          % valid if at least one [i] = 0
then    T_IPLSL_VLD_DLY(n) = T_IPLSL_VLD_DLY(n-1) + Recurrence
else    T_IPLSL_VLD_DLY = 0
endif
```

```
if       ( ( LV_IPLSL_VLD[i] = 0                     % valid if at least one [i] = 0
             (readiness of upstream oxygen sensor)
             and T_IPLSL_VLD_DLY >= C_T_IPLSL_VLD_DLY
           )
      or     LV_ERR_TCO = 1
             (error currently present on error location “coolant temperature sensor”)
      or     LV_ERR_MAF = 1
             (error currently present on mass air flow acquisition)
      or     LV_ERR_CAM = 1
             (global error on camshaft sensor)
         )
      and LC_INH_NT_ACT_OFF = 0
then     LV_INH_NT_ACT = 1
else     LV_INH_NT_ACT = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_NT_ACT_OFF	1	0...1H	0...1	1	[-]
Codeword for the manual setting of the inhibiting bit					
C_T_IPLSL_VLD_DLY	1	0...FFH	0...25.5	0.1	[s]
Delay time, during there LV_IPLSL_VLD[i] = 0 is hided for NOx catalyst management					

53.7.2 Inhibiting the NOx catalyst regeneration request

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_RGN_REQ	O/V	0...1H	0...1	1	[-]
flag inhibiting the request of a NOx catalyst regeneration					

Input data:

TNT_MDL_H	LV_IS	VS	LV_NT_ACT
LV_VLS_NS_VLD[NC_NOX_SENS_CONF]			

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: at reset

LV_INH_RGN_REQ = 1

Recurrence: 100 ms

Activation: LV_NT_ACT = 1

Deactivation: LV_NT_ACT = 0

Formula section:

```
if     TNT_MDL_H  >  C_TNT_MDL_THD_RGN_NOT                          OR
       VS < C_VS_MIN_RGN                                             OR
       (LV_IS = 1  AND  LC_INH_RGN_IS = 1)                           OR
       (LV_VLS_NS_VLD[k] = 0           AND       % valid if at least one [k] = 0
       LC_VLS_NS_VLD_INH_RGN_ENA = 1)
         (readiness of the binary O2 signal from the NOx-Sensor)

then     LV_INH_RGN_REQ = 1

else

       if     TNT_MDL_H  <  ( C_TNT_MDL_THD_RGN_NOT -
                                  C_TNT_MDL_THD_RGN_NOT_HYS )        AND
              VS >= C_VS_MIN_RGN                                     AND
              (LV_IS = 0  OR  LC_INH_RGN_IS = 0)                     AND
              (LV_VLS_NS_VLD[k] = 1        OR          % valid for all [k] = 1
              LC_VLS_NS_VLD_INH_RGN_ENA = 0)

       then  LV_INH_RGN_REQ = 0

       endif

endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TNT_MDL_THD_RGN_NOT	1	0...FFH	0...1020	4	[°C]
NOx trap monolith temperature max. threshold for inhibit a regeneration					
C_TNT_MDL_THD_RGN_NOT_HYS	1	0...FFH	0...1020	4	[°C]
NOx trap monolith temperature hysteresis for inhibit a regeneration					
C_VS_MIN_RGN	1	0...FFH	0...255	1	[km/h]
Minimum vehicle speed for requesting a NOx regeneration					
LC_INH_RGN_IS	1	0...1H	0...1	1	[-]
Inhibition of NOx regeneration requests at idle speed					
LC_VLS_NS_VLD_INH_RGN_ENA	1	0...1H	0...1	1	[-]
Use LV_VLS_NS_VLD for calculation of LV_INH_RGN_REQ					

53.7.3 Inhibiting the regeneration request due to modeled loading degree

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_RGN_REQ_NTLD	O/V	0...1H	0...1	1	[-]
flag inhibiting the request of a NOx catalyst regeneration due to high loading degree					

Input data:

LV_NT_ACT	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_NOX_NS_VLD[NC_NOX_SENS_CONF]	
LV_AT	NT_AGI	DIST_NT	

FUNCTION DESCRIPTION:

This is a dummy file provided for project-specific reasons to disable regeneration request due to high loading degree. Contrary to the previous subchapter, this new inhibition flag does not inhibit a regeneration due to a switch to homogeneous operation due to torque request by driver.

Application conditions:

Initialisation: at reset

LV_INH_RGN_REQ_NTLD = 0

Recurrence: 100 ms

Activation: LV_NT_ACT = 1

Deactivation: LV_NT_ACT = 0

Formula section:

```
if    LC_INH_RGN_REQ_NTLD = 1                                          OR
      ( LV_NOX_NS_AUTH[k] = 1          AND      % valid if all [k] = 1
        LC_INH_NTLD_REQ_NOX_SENS_AUTH = 1 )
                                       OR
      ( LV_NOX_NS_VLD[k] = 1           AND      valid if all [k] = 1
        LC_INH_NTLD_REQ_NOX_SENS_VLD = 1 )                             OR
      ( LV_AT = 0  AND  NT_AGI > C_NT_AGI_INH_RGN_REQ_NTLD_MT )
                                       OR
      ( LV_AT = 1  AND  NT_AGI > C_NT_AGI_INH_RGN_REQ_NTLD_AT )
                                       OR
      ( LV_AT = 0  AND  DIST_NT < C_DIST_NT_INH_RGN_REQ_NTLD_MT )      OR
      ( LV_AT = 1  AND  DIST_NT < C_DIST_NT_INH_RGN_REQ_NTLD_AT )
then    LV_INH_RGN_REQ_NTLD = 1
else    LV_INH_RGN_REQ_NTLD = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_RGN_REQ_NTLD	1	0...1H	0...1	1	[-]
Codeword for general inhibition of regeneration request due to loading degree					
LC_INH_NTLD_REQ_NOX_SENS_AUTH	1	0...1H	0...1	1	[-]
Inhibition of NOx regeneration requests due to NTLD after the NOx sensor has reached its readiness					
LC_INH_NTLD_REQ_NOX_SENS_VLD	1	0...1H	0...1	1	[-]
Inhibition of NOx regeneration requests due to NTLD, if the NOx signal is valid					
C_NT_AGI_INH_RGN_REQ_NTLD_AT	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Inhibition of NOx regeneration requests due to NTLD at new NOx catalyst - AT vehicle					
C_NT_AGI_INH_RGN_REQ_NTLD_MT	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Inhibition of NOx regeneration requests due to NTLD at new NOx catalyst - MT vehicle					
C_DIST_NT_INH_RGN_REQ_NTLD_AT	1	0...FFFFH	0...524280	8	[km]
Inhibition of NOx regeneration requests due to NTLD at new vehicle - AT vehicle					
C_DIST_NT_INH_RGN_REQ_NTLD_MT	1	0...FFFFH	0...524280	8	[km]
Inhibition of NOx regeneration requests due to NTLD at new vehicle - MT vehicle					

53.7.4 Inhibiting the lean operation (manual request of air/fuel stoichiometric)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_NT_AFL	O/V	0...1H	0...1	1	[-]
flag inhibiting lean operation (requests air/fuel stoichiometric)					

Input data:

LV_ERR_NS_CAN_BOFF	LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_1[NC_NOX_SENS_CONF]	
LV_ERR_LS_UP[NC_CBK_EX_NR]	STATE_ERR_NS_OBD_2[NC_NOX_SENS_CONF]		

FUNCTION DESCRIPTION:

The logical value LV_INH_NT_AFL allows to force a Lambda = 1.0 operation due to project specific requirements. Lambda = 1.0 operation is forced by an error of the NOx-Sensor or manual setting.

Application conditions:

Initialisation: at reset

LV_INH_NT_AFL = 0

Recurrence: 100 ms

Activation: every time

Deactivation: -

Formula section:

```
if  (    LV_ERR_NS_CAN_BOFF = 1
    or   LV_ERR_NS_CAN_MSG_LOST[k] = 1                  % valid if at least one [k] = 1
```

```
        or   LV_ERR_NS_OBD_1[k] = 1                        % valid if at least one [k] = 1
        or   ( STATE_ERR_NS_OBD_2[k] bitwise AND
               C_ERR_NS_NT_AFL_BIT_SEL ) != 0              % valid if at least one [k] = 1
        or   LV_ERR_LS_UP[i] = 1                           % valid if at least one [i] = 1
        or   LV_INH_NT_AFL_NS_READY = 1
        or   LC_INH_NT_AFL = 1
      )
      and  LC_INH_NT_AFL_OFF = 0
then       LV_INH_NT_AFL = 1
else       LV_INH_NT_AFL = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_NT_AFL	1	0...1H	0...1	1	[-]
Codeword for forcing to Lambda=1 operation					
LC_INH_NT_AFL_OFF	1	0...1H	0...1	1	[-]
Manual setting of inhibiting Lambda = 1 request					
C_ERR_NS_NT_AFL_BIT_SEL	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to set lean engine operation request					

53.7.5 Change to model when invalid NOx-Sensor Signal flow diagram

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_RGN_REQ_NOX_SENS	O/V	0...1H	0...1	1	[-]
Inhibition flag for NOx catalyst regeneration request by NOx sensor readiness					
LV_INH_NOX_SENS_RGN	O/V	0...1H	0...1	1	[-]
Inhibition flag for NOx catalyst regeneration-stop request by NOx sensor readiness					
LV_NTLD_ADJ	O/V	0...1H	0...1	1	[-]
Flag to activate NOx catalyst loading degree threshold for regeneration without NOx sensor readiness					
STATE_RGN_REQ	V	0H 1H	SENSOR MODEL	1	[-]
State of regeneration request					

Input data:

LV_NOX_NS_VLD[NC_NOX_SENS_CONF]	LV_T_NS_VLD	STATE_NOX	NTLD
T_NOX_NS_HLD[NC_NOX_SENS_CONF]	LV_NOX_NS_ACT[NC_NOX_SENS_CONF]	STATE_ERR_NS_OBD_2[NC_NOX_SENS_CONF]	STATE_END_NS_OBD_2[NC_NOX_SENS_CONF]
LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_1[NC_NOX_SENS_CONF]		

FUNCTION DESCRIPTION:

If the function is active (LC_INH_REQ_NOX_SENS_OFF = 0) and the NOx signal is not valid during the NOx catalyst loading phase the mode of the NOx regeneration request is switched from "SENSOR" to "MODEL". The NOx regeneration request by NOx sensor is as long as forbidden until the NOx sensor OBDII diagnosis is carried out and no error is appeared.

Application conditions:

Initialisation: at reset, at deactivation:

LV_INH_RGN_REQ_NOX_SENS = 0
LV_INH_NOX_SENS_RGN = 0
LV_NTLD_ADJ = 0
STATE_RGN_REQ = SENSOR

Recurrence: 100ms.

Activation: LC_INH_REQ_NOX_SENS_OFF = 0

Deactivation: LC_INH_REQ_NOX_SENS_OFF = 1

Formula section:

Default State:

STATE_RGN_REQ = SENSOR

Transitions:

STATE_RGN_REQ: **SENSOR to MODEL**

```
if    STATE_NOX = LOAD
      and    (LV_NOX_NS_VLD[k] = 0
             and T_NOX_NS_HLD[k] >= C_T_NOX_SENS_VLD_STATE_RGN_REQ)
                                        % valid if at least one condition for [k] = true
      and    LV_T_NS_VLD = 1
      and    NTLD >= C_NTLD_STATE_RGN_REQ
then         STATE_RGN_REQ = MODEL
endif
```

STATE_RGN_REQ: **MODEL to SENSOR**

```
if    % valid if all conditions for [k] = true
      ( (bit-wise not(STATE_END_NS_OBD_2[k]) bit-wise or STATE_ERR_NS_OBD_2[k] )
      bit-wise and C_STATE_NS_OBD_BIT_SEL_MDL ) = 0
      and    LV_ERR_NS_OBD_1[k]= 0
      and    LV_NOX_NS_AUTH[k] = 1
      and    LV_NOX_NS_ACT[k] = 1
```

```
then     STATE_RGN_REQ = SENSOR
else     STATE_RGN_REQ  unchanged
endif
```

Static state actions:

STATE_RGN_REQ = **SENSOR**

LV_INH_RGN_REQ_NOX_SENS = 0

LV_INH_NOX_SENS_RGN = 0

LV_NTLD_ADJ = 0

STATE_RGN_REQ = **MODEL**

LV_INH_RGN_REQ_NOX_SENS = 1

LV_INH_NOX_SENS_RGN = 1

LV_NTLD_ADJ = 1

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_REQ_NOX_SENS_OFF	1	0...1H	0...1	1	[-]
Manual setting of inhibiting exclusively regeneration out of NOx sensor					
C_NTLD_STATE_RGN_REQ	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Minimum NOx catalyst loadig degree to switch to model (< NTLD_MAX)					
C_T_NOX_SENS_VLD_STATE_RGN_REQ	1	0...FFFFH	0...655.35	0.01	[s]
Minimum time with NOx sensor not valid to switch to model					
C_STATE_NS_OBD_BIT_SEL_MDL	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor OBDII diagnosis types for changing from model to sensor mode					

53.7.6 Catalyst cooling function

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TNT_MDL_MAX_OFS	O/V	0...1H	0...1	1	[-]
Catalyst cooling function activation flag					
TNT_MDL_MAX_OFS	O/V	80...7FH	-512...508	4	[°C]
NOx catalyst temperature offset to enable catalyst cooling at standing car					
T_VS_IS_NT	V	0...FFH	0...25.5	0.1	[s]
Catalyst cooling function activation timer					

Input data:

VS	TNT_MDL_H	NT_SUL	

Application conditions:

Initialisation: at reset, at deactivation

LV_TNT_MDL_MAX_OFS = 0
TNT_MDL_MAX_OFS = 0
T_VS_IS_NT = 0

Recurrence: 100ms

Activation: LC_TNT_MDL_MAX_OFS = 1

Deactivation: LC_TNT_MDL_MAX_OFS = 0

Formula section:

```
if      VS <= C_VS_TNT_MDL_MIN
    and     TNT_MDL_H > C_TNT_MDL_H_MAX_OFS
    and     NT_SUL >= C_NT_SUL_MIN_OFS
then        if      T_VS_IS_NT <= C_T_VS_IS_NT
            then    if      T_VS_IS_NT < maximum value (FFH)
                    then    T_VS_IS_NT = T_VS_IS_NT + Recurrence
                    else    T_VS_IS_NT(n) = T_VS_IS_NT(n-1)
                    endif
                    TNT_MDL_MAX_OFS = C_TNT_MDL_MAX_OFS
                    LV_TNT_MDL_MAX_OFS = 1
            else    TNT_MDL_MAX_OFS = 0
                     LV_TNT_MDL_MAX_OFS = 0
            endif
else        if      VS <= C_VS_TNT_MDL_MIN_OFF
                and TNT_MDL_H <= C_TNT_MDL_H_MAX_OFS -
                                         C_TNT_MDL_H_MAX_OFS_HYS
```

```
        then  T_VS_IS_NT = 0
              TNT_MDL_MAX_OFS = 0
              LV_TNT_MDL_MAX_OFS = 0
        endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TNT_MDL_MAX_OFS	1	0...1H	0...1	1	[-]
Switch to enable cat cooling at standing car					
C_T_VS_IS_NT	1	0...FFH	0...25.5	0.1	[s]
Duration cat cooling at standing car is allowed					
C_VS_TNT_MDL_MIN	1	0...FFH	0...255	1	[km/h]
Vehicle speed threshold to enable cat cooling at standing car					
C_VS_TNT_MDL_MIN_OFF	1	0...FFH	0...255	1	[km/h]
Vehicle speed threshold to disable catalyst cooling					
C_TNT_MDL_MAX_OFS	1	80...7FH	-512...508	4	[°C]
NOx catalyst temperature offset to enable cat cooling at standing car					
C_TNT_MDL_H_MAX_OFS	1	0...FFH	0...1020	4	[°C]
Min NOx catalyst temperature to enable cat cooling at standing car					
C_TNT_MDL_H_MAX_OFS_HYS	1	0...FFH	0...1020	4	[°C]
NOx catalyst temperature hysteresis to disable catalyst cooling					
C_NT_SUL_MIN_OFS	1	0...FFFFH	0...10485.6	0.16	[mg]
Min NOx catalyst sulphur load to enable cat cooling at standing car					

53.7.7 Enabling NOx Catalyst regeneration request by model

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ADJ_NOX_SENS_RGN	O/V	0...1H	0...1	1	[-]
Exclusively regeneration by NOx-Sensor ist allowed (=0)					

Input data:

LV_ERR_EGR_2			

Application conditions:

Initialisation: at reset

LV_ADJ_NOX_SENS_RGN = 0

Recurrence: 10 ms

Activation: at every engine state

Deactivation: -

Formula section:

```
if    LV_ERR_EGR_2 = 1
then  LV_ADJ_NOX_SENS_RGN = 1
```

```
else        LV_ADJ_NOX_SENS_RGN = 0
endif
```

53.7.8 Calculation of full load air mass flow integral

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MAF_INT_FL_ACT	O/V	0...FFFFH	0...6553.5	0.1	[g]
Air mass flow since activation of full load (LV_FL = 0->1)					
LV_LOAD_H	O/V	0...1H	0...1	1	[-]
High engine load detected					

Input data:

LV_NT_ACT	LV_FL	MAF_KGH	TQI_SP_MAF
N_32			

General information:

The air mass flow integral MAF_INT_FL_ACT is used for detection of short and long full load phases or high engine load phases inside of NOx catalyst management functionality. This classification of full load phases or high engine load phases is required for the decision whether a running NOx regeneration is terminated because of higher full load priority or not.

Description:

The constant NC_FAC_MAF_INT_FL includes the recurrence and converts the physical unit of MAF_KGH from [kg/h] to [g/ms].

NC_FAC_MAF_INT_FL = Recurrence/ 3.6

Note! The calculation of MAF_INT_FL_ACT shall be stopped if the maximum value of value range is reached. An overflow is not allowed.

Application conditions:

Initialisation: at reset, at deactivation:

MAF_INT_FL_ACT = 0
LV_LOAD_H = 0

Recurrence: 100ms

Activation: LV_NT_ACT = 1

Deactivation: LV_NT_ACT = 0

Formula section:

```
if      TQI_SP_MAF > IP_TQI_SP_MAF_LOAD_H(N_32)
then        LV_LOAD_H = 1
else
        if      TQI_SP_MAF < IP_TQI_SP_MAF_LOAD_H(N_32) - C_TQI_SP_MAF_LOAD_H_HYS
```

```
        then    LV_LOAD_H = 0
        else    LV_LOAD_H remains unchanged
        endif
endif

if      LV_FL = 1 OR LV_LOAD_H = 1
then    MAF_INT_FL_ACT(n) = MAF_INT_FL_ACT(n-1) + MAF_KGH * NC_FAC_MAF_INT_FL
else    MAF_INT_FL_ACT = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQI_SP_MAF_LOAD_H	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_LOAD_H	8	0...FFH	0...8160	32	[rpm]
Torque threshold for detecting high engine load					
C_TQI_SP_MAF_LOAD_H_HYS	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque threshold hysteresis for reset of high engine load flag					

53.7.9 Stop/ start functionality activation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_NT_STST	O/V	0...1H	0...1	1	[-]
Activation/ deactivation of stop/ start functionalities into NOx catalyst functions					

Input data:

LC_NT_STST_ACT	LV_STST_STOP_CYC	N_32	

General information:

LV_NT_STST = 1 activates into the NOx catalyst control functions special variable calculations in order to handle the stop/ start phases.

Description:

The special stop/ start functionalities into the NOx catalyst control functions are enabled or disabled by LC_NT_STST_ACT.

Application conditions:

Initialisation: at reset, at deactivation:

LV_NT_STST = 0

Recurrence: 10ms

Activation: LC_NT_STST_ACT = 1

Deactivation: LC_NT_STST_ACT = 0

Formula section:

```
if    LV_STST_STOP_CYC = 1  AND  N_32 < C_N_32_NT_STST_ACT
then      LV_NT_STST = 1
else
          if    LV_STST_STOP_CYC = 0                              OR
                ( LV_STST_STOP_CYC = 1  AND  N_32 > C_N_32_NT_STST_DEAC )
          then      LV_NT_STST = 0
          endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_32_NT_STST_ACT	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for activation of stop/ start functionalities into NOx catalyst functions					
C_N_32_NT_STST_DEAC	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for deactivation of stop/ start functionalities into NOx catalyst functions					

53.7.10 Request of Lambda=1.0 operation due to missing NOx sensor readiness

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_NT_AFL_NS_READY	O/V	0...1H	0...1	1	[-]
Request of Lambda=1.0 engine operation due to missing NOx sensor readiness					
CTR_RGN_MDL_CTL	O/V	0...FFH	0...255	1	[-]
Counter of model controlled NOx regenerations					
T_AFL_MDL_CTL	O/V	0...FFFFH	0...6553.5	0.1	[s]
Model controlled lean engine operation time after x model controlled NOx regenerations					
T_AFL_MDL_CTL_SUM	O/V	0...FFFFH	0...6553.5	0.1	[s]
Summary model controlled lean engine operation time at actual driving cycle					
TCO_ST_NS	O/V	0...FEH	-48...142.5	0.75	[°C]
Coolant temperature at start					

Input data:

TCO_ST	TCO_ST_DC	LV_RGN_NT_REQ	LV_NOX_NS_AUTH[NC_NOX_SENS_CONF]
LV_VLS_NS_AUTH[NC_NOX_SENS_CONF]	T_RGN	STATE_NOX	OPM_AV
T_AST_SAE			

General information:

The engine system is able to start with lean engine operation without a ready NOx sensor for NOx emission control. In this case the NOx emission control runs completely model controlled. The aim of this section is the limitation of this system state by requesting Lambda=1.0 engine operation up to the NOx sensor reaches its operation readiness.

Description:

Application conditions:

Initialisation: at reset, at deactivation:

LV_INH_NT_AFL_NS_READY = 0
CTR_RGN_MDL_CTL = 0
T_AFL_MDL_CTL = 0
T_AFL_MDL_CTL_SUM = 0
TCO_ST_NS = 0

Recurrence: 100ms

Activation: LV_IGK = 1

Deactivation: -

Formula section:

Selection of TCO_ST signal

```
if    LC_TCO_ST_SEL = 1
then      TCO_ST_NS = TCO_ST_DC
else      TCO_ST_NS = TCO_ST
endif
```

Counting of model controlled NOx regenerations

```
if      (LV_NOX_NS_AUTH[k] = 0 OR LC_NOX_NS_AUTH_CHK = 0)          AND
        (LV_VLS_NS_AUTH[k] = 0 OR LC_VLS_NS_AUTH_CHK = 0)          AND
        LV_RGN_NT_REQ(n-1) = 1                                     AND
        LV_RGN_NT_REQ(n) = 0                                       AND
        T_RGN(n-1) >= C_T_RGN_MIN_NS_READY
then    Model controlled NOx regeneration detected
        increment CTR_RGN_MDL_CTL
endif
```

Measurement of time without NOx sensor readiness

```
if      (LV_NOX_NS_AUTH[k] = 0 OR LC_NOX_NS_AUTH_CHK = 0)          AND
        (LV_VLS_NS_AUTH[k] = 0 OR LC_VLS_NS_AUTH_CHK = 0)          AND
        (STATE_NOX = "LOAD" OR LC_STATE_NOX_CHK = 0)               AND
        (OPM_AV = "S" OR OPM_AV = "AFL" OR LC_OPM_AV_CHK = 0)
then    Model controlled lean engine operation detected
        increment T_AFL_MDL_CTL_SUM

        if      CTR_RGN_MDL_CTL >= ID_CTR_RGN_MDL_CTL_MAX(TCO_ST_NS)
        then    Model controlled lean engine operation time after x model controlled NOx
                regenerations
                increment T_AFL_MDL_CTL
        endif
endif
```

Request of Lambda=1.0 engine operation

```
if      (LV_NOX_NS_AUTH[k] = 0 OR LC_NOX_NS_AUTH_CHK = 0)          AND
        (LV_VLS_NS_AUTH[k] = 0 OR LC_VLS_NS_AUTH_CHK = 0)
then    Waiting on NOx sensor readiness
        if      CTR_RGN_MDL_CTL >= ID_CTR_RGN_MDL_CTL_MAX(TCO_ST_NS)        AND
                T_AFL_MDL_CTL >= IP_T_AFL_MDL_CTL_MAX(TCO_ST_NS)            AND
                T_AFL_MDL_CTL_SUM >= IP_T_AFL_MDL_CTL_SUM_MAX(TCO_ST_NS)    AND
                T_AST_SAE >= IP_T_AST_SAE_MDL_CTL_MAX(TCO_ST_NS)
        then    Maximum model controlled lean engine operation duration is reached
                Set of Lambda=1.0 operation request
                LV_INH_NT_AFL_NS_READY = 1
        else    Model controlled lean engine operation is running
                LV_INH_NT_AFL_NS_READY = 0
        endif
else    NOx sensor has reached its readiness
```

```
        if  T_AST_SAE >= C_T_AST_MIN_NS_READY
                Reset of Lambda=1.0 operation request
                LV_INH_NT_AFL_NS_READY = 0
                Reset of counter & timers after NOx sensor readiness
                CTR_RGN_MDL_CTL = 0
                T_AFL_MDL_CTL = 0
                T_AFL_MDL_CTL_SUM = 0
        endif
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TCO_ST_SEL	1	0...1H	0...1	1	[-]
Selection of TCO_ST (=0) or TCO_ST_DC (=1) for use at this functionality					
LC_NOX_NS_AUTH_CHK	1	0...1H	0...1	1	[-]
Activation of LV_NOX_NS_AUTH[...] for NOx sensor readiness check					
LC_VLS_NS_AUTH_CHK	1	0...1H	0...1	1	[-]
Activation of LV_VLS_NS_AUTH[...] for NOx sensor readiness check					
LC_STATE_NOX_CHK	1	0...1H	0...1	1	[-]
Activation of STATE_NOX for determination of model controlled lean engine operation time					
LC_OPM_AV_CHK	1	0...1H	0...1	1	[-]
Activation of OPM_AV for determination of model controlled lean engine operation time					
C_T_RGN_MIN_NS_READY	1	0...FFFFH	0...655.35	0.01	[s]
Minimum NOx regeneration duration for counting as model controlled NOx regeneration					
ID_CTR_RGN_MDL_CTL_MAX	6	0...FFH	0...255	1	[-]
LDP_TCO_ST_NS_ID_CTR_RGN_MDL	6	0...FEH	-48...142.5	0.75	[°C]
Maximum number of model controlled NOx regenerations					
IP_T_AFL_MDL_CTL_MAX	6	0...FFFFH	0...6553.5	0.1	[s]
LDP_TCO_ST_NS_IP_T_AFL_MDL	6	0...FEH	-48...142.5	0.75	[°C]
Maximum model controlled lean engine operation time after x model controlled NOx regenerations					
IP_T_AFL_MDL_CTL_SUM_MAX	6	0...FFFFH	0...6553.5	0.1	[s]
LDP_TCO_ST_NS_IP_T_AFL_MDL_SUM	6	0...FEH	-48...142.5	0.75	[°C]
Maximum model controlled lean engine operation time at actual driving cycle					
IP_T_AST_SAE_MDL_CTL_MAX	6	0...FFFFH	0...65535	1	[s]
LDP_TCO_ST_NS_IP_T_AST_SAE	6	0...FEH	-48...142.5	0.75	[°C]
Maximum model controlled lean engine operation duration after start					
C_T_AST_MIN_NS_READY	1	0...FFFFH	0...65535	1	[s]
Minimum time after start to allow lean operation in case of late readiness of NOx sensor					

53.8 NOx tailpipe emission

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NOX_OUT_MES_INT[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...10485.6	0.16	[mg]
Absolute tailpipe NOx emission during stratified operation mode					
NOX_OUT_MES[NC_NOX_SENS_CONF]	V/O	0...FFFFH	0...1023.98437	0.015625	[mg/s]
Tailpipe NOx emission					

Input data:

NOX_NS_AD[NC_NOX_SENS_CONF]	STATE_NOX	MAF_FG_CYL	C_NOX_CONC_COR
LV_NOX_SENS	LV_NOX_NS_VLD[NC_NOX_SENS_CONF]	LV_ERR_NS_OBD_1[NC_NOX_SENS_CONF]	
LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_T_NS_VLD	NC_NOX_SENS_CONF	STATE_ERR_NS_OBD_2[NC_NOX_SENS_CONF]

FUNCTION DESCRIPTION:

LV_ERR_NS_OBD_2[NC_NOX_SENS_CONF]

General information

The NOx signal of the NOx sensor is used to calculate the NOx tailpipe emission during lean operation mode.

The NOx concentration measured behind the NOx catalyst is converted into a mass flow. This NOx mass flow is integrated up to the absolute NOx tailpipe emission during a lean phase.

NC_NOX_SENS_CONF is the number of NOx sensors within the system. Therefore the index i represents:

i = 1 ... signals based on NOx sensor 1, which is mounted at cylinder bank 1 or at a summary tailpipe section of a Y exhaust line configuration

i = 2 ... signals based on NOx sensor 2, which is mounted at cylinder bank 2

Application conditions:

Initialisation: at reset

NOX_OUT_MES_INT[i] = 0
NOX_OUT_MES[i] = 0

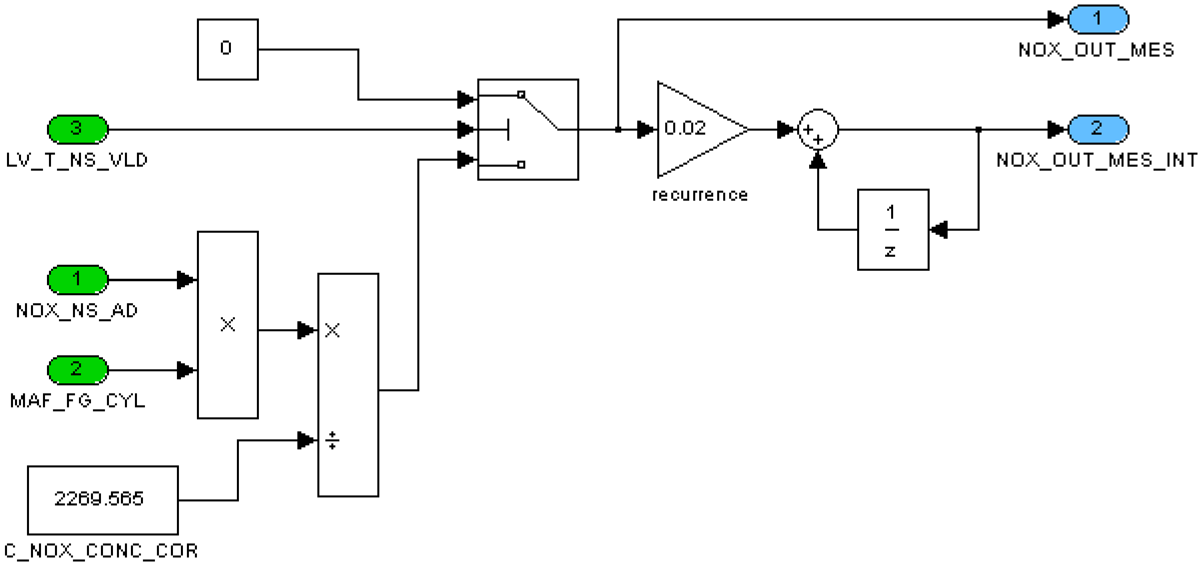
Recurrence: *20 ms*

Activation: **if** NC_NOX_SENS_CONF > 0

then every time

```
        endif

Deactivation:  NC_NOX_SENS_CONF = 0
```

Signal flow diagram:



This signal flow diagram shows the calculation principle. For details see the "Formula section" below.

Formula section:

```
if (1)   STATE_NOX(n-1) = STOP                                AND
         STATE_NOX(n) = NOT (STOP)
then (1)   NOX_OUT_MES_INT[i] = 0
endif (1)

if (2)   LC_NOX_OUT_MES_INT_ACT = 1
then (2)
         if (3)   STATE_NOX(n) = LOAD                         AND
                  LV_NOX_SENS = 1                             AND
                  LV_NOX_NS_VLD[i] = 1                        AND
                  LV_ERR_NS_CAN_MSG_LOST_[i] = 0              AND
                  LV_ERR_NS_OBD_1[i]i = 0                     AND
                  ~~LV_ERR_NOX_SENSS_OBD_2[i]i = 0            AND~~
                  ( STATE_ERR_NS_OBD_2[i] bitwise AND
                            C_ERR_NS_NOX_OUT_BIT_SEL ) != 0
         then (3)
                  if (4)   LV_T_NS_VLD = 0
```

```
                then (4)   NOX_OUT_MES[i] = 0
                else (4)   NOX_OUT_MES[i] = NOX_NS_AD[i] *
                                            (MAF_FG_CYL / NC_NOX_SENS_CONF) /
                                            C_NOX_CONC_COR
                endif (4)
                NOX_OUT_MES_INT[i] (n) = NOX_OUT_MES_INT[i] (n-1) +
                                            NOX_OUT_MES[i] * recurrence
                (Internally calculation with higher resolution)
        else (3)   NOX_OUT_MES_INT[i] (n) = NOX_OUT_MES_INT[i] (n-1)
        endif (3)
else (2)   NOX_OUT_MES_INT[i] = 0
endif (2)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NOX_OUT_MES_INT_ACT	1	0...1H	0...1	1	[-]
Switch to enable calculation of NOx tailpipe emission					
C_ERR_NS_NOX_OUT_BIT_SEL	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit NOX_OUT_MES... calculations					

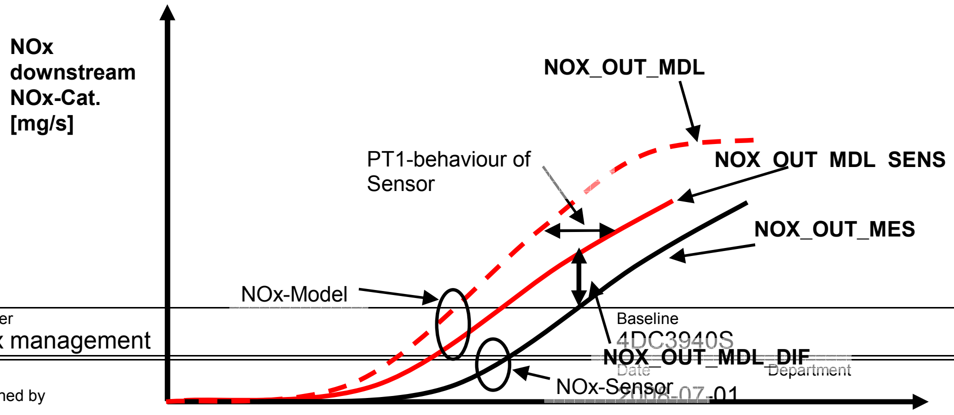
53.9 NOx Catalyst model controller

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>NOX_OUT_MDL_SENS</td><td>V</td><td>0...FFFFH</td><td>0...1023.98437</td><td>0.015625</td><td>[mg/s]</td></tr>
<tr><td colspan="6">modelled NOx emission flow downstream the NOx Catalyst</td></tr>
<tr><td>NOX_OUT_MDL_DIF[NC_NOX_SENS_CONF]</td><td>V</td><td>8000...7FFFH</td><td>-512...511.98437</td><td>0.015625</td><td>[mg/s]</td></tr>
<tr><td colspan="6">Deviation between measured and calculated NOx-Flow after NOx-Catalyst (bench selective)</td></tr>
<tr><td>NTLD_MDL_DIF_I</td><td>V</td><td>8000...7FFFH</td><td>-0.5...0.49998</td><td>0.0153e-3</td><td>[-]</td></tr>
<tr><td colspan="6">I-value of the closed loop control NOx catalyst model</td></tr>
<tr><td>NTLD_MDL_DIF_P</td><td>V</td><td>8000...7FFFH</td><td>-0.5...0.49998</td><td>0.0153e-3</td><td>[-]</td></tr>
<tr><td colspan="6">P-value of the closed loop control NOx catalyst model</td></tr>
<tr><td>NTLD_MDL_DIF</td><td>V/O</td><td>8000...7FFFH</td><td>-0.5...0.49998</td><td>0.0153e-3</td><td>[-]</td></tr>
<tr><td colspan="6">Correction of the closed loop control NOx catalyst model</td></tr>
<tr><td>NOX_OUT_MES_MMV[NC_NOX_SENS_CONF]</td><td>V</td><td>0...FFFFH</td><td>0...1023.98437</td><td>0.015625</td><td>[mg/s]</td></tr>
<tr><td colspan="6">Moving mean value of measured NOx-Flow after NOx-Catalyst (bench selective)</td></tr>
<tr><td>LV_NOX_RGN_CMPL</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Boolean whether regenerations are "completely" or not</td></tr>
<tr><td>T_NOX_MDL_CTRL_ACT[NC_NOX_SENS_CONF]</td><td>V</td><td>0...FFFFH</td><td>0...1310.7</td><td>0.02</td><td>[s]</td></tr>
<tr><td colspan="6">Time controller is active (bench selective)</td></tr>
<tr><td>LV_INH_NOX_MDL_CTRL_RGN</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Prohibition of regeneration by controller controlled NOx-Signal</td></tr>
<tr><td>T_NOX_MDL_CTRL_ENA[NC_NOX_SENS_CONF]</td><td>V</td><td>0...FFFFH</td><td>0...1310.7</td><td>0.02</td><td>[s]</td></tr>
<tr><td colspan="6">Time controller is enable (bench selective)</td></tr>
<tr><td>NTLD_MDL_DIF_I_SNG[NC_NOX_SENS_CONF]</td><td>V</td><td>8000...7FFFH</td><td>-0.5...0.49998</td><td>0.0153e-3</td><td>[-]</td></tr>
<tr><td colspan="6">I-value of the closed loop control NOx catalyst model - bank selective value</td></tr>
<tr><td>NTLD_MDL_DIF_P_SNG[NC_NOX_SENS_CONF]</td><td>V</td><td>8000...7FFFH</td><td>-0.5...0.49998</td><td>0.0153e-3</td><td>[-]</td></tr>
<tr><td colspan="6">P-value of the closed loop control NOx catalyst model - bank selective value</td></tr>
<tr><td>LV_INH_NOX_MDL_CTRL_RGN_SNG[NC_NOX_SENS_CONF]</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Prohibition of regeneration by controller controlled NOx-Signal</td></tr>
</table>

Input data:

MAF_KGH	NOX_OUT_MES[NC_NOX_SENS_CONF]	NOX_COR_RED_MMV	STATE_NOX
NTLD	NOX_OUT_MDL	LV_NOX_NS_MDL_VLD[NC_NOX_SENS_CONF]	LV_T_NS_VLD_TMP[NC_NOX_SENS_CONF]
T_AFL	NOX_NS_AD[NC_NOX_SENS_CONF]	LV_HOM_AFL_ACT	LV_HOM_AFS_REQ
TNT_MDL_MV	LV_S_ACT	NOX_CONC_OUT_MDL	NC_NOX_SENS_CONF



FUNCTION DESCRIPTION:

Number of NOx sensors NC_NOX_SENS_CONF i = 1...NC_NOX_SENS_CONF

General information:

The NOx catalyst model (NOx catalyst management) provides the modelled NOx emission flow downstream the NOx Catalyst, NOX_OUT_MDL, which corresponds to NOX_OUT_MES, the measured NOx emission flow downstream the NOx Catalyst. The deviations between modelled and measured values are minimised with a PI-controller influencing NTLD used in the NOx catalyst model.

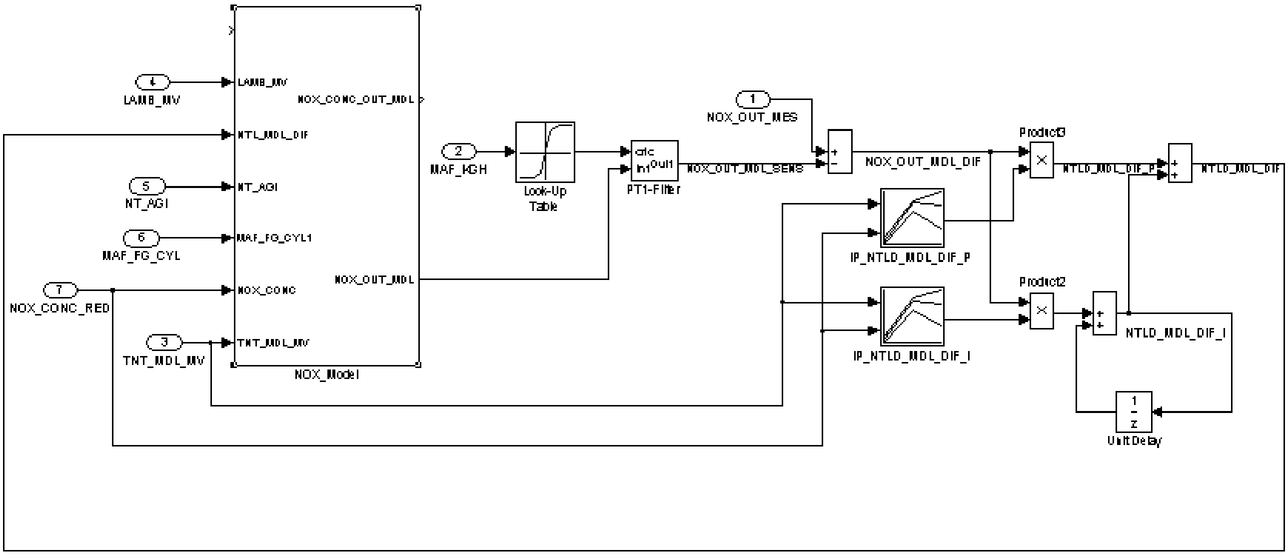
If the system is equipped with 2 cylinder banks, a regeneration request is set if at least one cylinder bank requests a regeneration. Due to the NOx catalyst model ist calibrated with the NOx-Sensor measuring the higher value.

Application conditions:

Initialisation:	at engine stop -> start: *LV_INH_NOX_MDL_CTRL_RGN_SNG[i] = LC_INH_NOX_MDL_CTRL_INI*
Recurrence:	20 ms
Activation:	LC_NOX_MDL_CTRL_ACT = 1
Deactivation:	**if** LC_NOX_MDL_CTRL_ACT = 0

```
if    LC_NOX_MDL_CTRL_ACT = 0
then  NTLD_MDL_DIF = 0,
      T_NOX_MDL_CTRL_ENA[i] = 0
      T_NOX_MDL_CTRL_ACT[i] = 0
      NTLD_MDL_DIF_I = 0
      NTLD_MDL_DIF_I_SNG[i] = 0
      NOX_OUT_MES_MMV[i] = NOX_OUT_MES[i]
      NOX_OUT_MDL_SENS =  NOX_OUT_MDL
      LV_INH_NOX_MDL_CTRL_RGN_SNG[i] = 0
      LV_INH_NOX_MDL_CTRL_RGN = 0
```

Signal flow diagrams:



Figur 1: controller during NOX_STATE = LOAD (simplified)

Formula section:

53.9.1 Computation of model deviation

LV_NOX_RGN_CMPL = LC_NOX_RGN_CMPL

when case STATE_NOX = LOAD

$$NOX\_OUT\_MDL\_SENS_{N} = IP\_CRLC\_NOX\_SENS * NOX\_OUT\_MDL + (1 - IP\_CRLC\_NOX\_SENS) * NOX\_OUT\_MDL\_SENS_{N-1}$$

```
          % reset of time counter
if        T_AFL =< T_NOX_MDL_CTRL_ENA[i]
then      T_NOX_MDL_CTRL_ENA[i] = 0
          T_NOX_MDL_CTRL_ACT[i] = 0
          LV_INH_NOX_MDL_CTRL_RGN_SNG[i] = LC_INH_NOX_MDL_CTRL_INI
endif

          % Initialisation of moving meanvalue of NOx flow downstream NOx Cat
if        LV_NOX_NS_MDL_VLD[i] = 0 --> 1
```

```
then        NOX_OUT_MES_MMV[i] = NOX_OUT_MES[i]
endif


            % Time counter of possible controller activation
if          LV_T_NS_VLD_TMP[i] = 1 and
            ((NTLD >= C_NTLD_CTRL_MIN
            or      NOX_NS_AD[i]  >= C_NOX_PPM_CTRL_MIN))
then        T_NOX_MDL_CTRL_ENA[i]$_N$ = T_NOX_MDL_CTRL_ENA[i]$_{N-1}$ + 20ms
else        T_NOX_MDL_CTRL_ENA[i]$_N$ = T_NOX_MDL_CTRL_ENA[i]$_{N-1}$
endif


            % Observation of active Time of the controller
if          T_NOX_MDL_CTRL_ACT[i]  / T_NOX_MDL_CTRL_ENA[i]  <
                                            C_T_NOX_MDL_INH_CTRL_MIN
            and     T_NOX_MDL_CTRL_ENA[i]$_N$ >= C_T_NOX_MDL_CTRL_ENA_MIN
then        LV_INH_NOX_MDL_CTRL_RGN_SNG[i] = 1
endif
            % Calculation of I- and P-part of the correction
if(1)       LV_T_NS_VLD_TMP[i] = 1
and         LV_NOX_NS_MDL_VLD[i] = 1
and         LV_HOM_AFS_REQ = 0
and         (LV_S_ACT = 1or LV_HOM_AFL_ACT = 1)
and         |NOX_OUT_MES[i]$_N$ - NOX_OUT_MES[i]$_{N-1}$| < C_NOX_OUT_MES_GRD_MAX
then(1)     NOX_OUT_MES_MMV[i]$_N$ = NOX_OUT_MES_MMV[i]$_{N-1}$ *
                    (1 - C_CRLC_NOX_OUT_MES_MMV) + NOX_OUT_MES[i] *
                                            C_CRLC_NOX_OUT_MES_MMV
    If (2)  max (NOX_CONC_OUT_MDL, NOX_NS_AD[i]) >= C_NOX_DIF_MON_MIN
    then(2) if(3)    | NOX_OUT_MDL - NOX_OUT_MES_MMV[i]| >= C_NOX_DIF_MON
            then(3)         LV_INH_NOX_MDL_CTRL_RGN_SNG[i] = 1
            else(3)         LV_INH_NOX_MDL_CTRL_RGN_SNG[i] = 0
            endif(3)
    endif(2)
    if (2)  |NOX_OUT_MES[i] - NOX_OUT_MES_MMV[i]|< C_NOX_OUT_MES_MMV_MAX
            and     (NTLD >= C_NTLD_CTRL_MIN
                    or      NOX_NS_AD[i] >= C_NOX_PPM_CTRL_MIN)
            and     NTLD + NTLD_MDL_DIF > 0
```

```
        then(2)
                NOX_OUT_MDL_DIF[i] = NOX_OUT_MES[i] - NOX_OUT_MDL_SENS
                $NTLD\_MDL\_DIF\_I\_SNG[i]_N = NTLD\_MDL\_DIF\_I\_SNG[i]_{N-1}$ +
                                IP_NTLD_MDL_DIF_I * NOX_OUT_MDL_DIF[i]
                $NTLD\_MDL\_DIF\_P\_SNG[i]_N$ = IP_NTLD_MDL_DIF_P * NOX_OUT_MDL_DIF[i]
                $T\_NOX\_MDL\_CTRL\_ACT[i]_N = T\_NOX\_MDL\_CTRL\_ACT[i]_{N-1}$ + recurrence
        else(2) $NTLD\_MDL\_DIF\_I\_SNG[i]_N = NTLD\_MDL\_DIF\_I\_SNG[i]_{N-1}$
                if(3)       LC_INH_P_HLD = 0
                then(3)     $NTLD\_MDL\_DIF\_P\_SNG[i]_N = NTLD\_MDL\_DIF\_P\_SNG[i]_{N-1}$
                else(3)     $NTLD\_MDL\_DIF\_P\_SNG[i]_N = 0$
                endif(3)
        endif(2)
    else(1)     $NTLD\_MDL\_DIF\_I\_SNG[i]_N = NTLD\_MDL\_DIF\_I\_SNG[i]_{N-1}$
                if(3)       LC_INH_P_HLD = 0
                then(3)     $NTLD\_MDL\_DIF\_P\_SNG[i]_N = NTLD\_MDL\_DIF\_P\_SNG[i]_{N-1}$
                else(3)     $NTLD\_MDL\_DIF\_P\_SNG[i]_N = 0$
                endif(3)
                $NOX\_OUT\_MES\_MMV[i]_N = NOX\_OUT\_MES\_MMV[i]_{N-1}$
        endif(2)
    endif(1)
```

Limitation of the integral part

```
if(1)       NTLD_MDL_DIF_I_SNG[i] < C_NTLD_MDL_DIF_I_MIN
then(1)     NTLD_MDL_DIF_I_SNG[i] = C_NTLD_MDL_DIF_I_MIN
endif(1)
if(1)       NTLD_MDL_DIF_I_SNG[i] > C_NTLD_MDL_DIF_I_MAX
then(1)     NTLD_MDL_DIF_I_SNG[i] = C_NTLD_MDL_DIF_I_MAX
endif(1)
```

Selection of bank for the NOx modell controller

```
if    NOX_OUT_MES_MMV[1] > NOX_OUT_MES_MMV[2]
then        NTLD_MDL_DIF_I = NTLD_MDL_DIF_I_SNG[1]
            NTLD_MDL_DIF_P = NTLD_MDL_DIF_P_SNG[1]
            LV_INH_NOX_MDL_CTRL_RGN = LV_INH_NOX_MDL_CTRL_RGN_SNG[1]
else        NTLD_MDL_DIF_I = NTLD_MDL_DIF_I_SNG[2]
            NTLD_MDL_DIF_P = NTLD_MDL_DIF_P_SNG[2]
```

```
        LV_INH_NOX_MDL_CTRL_RGN = LV_INH_NOX_MDL_CTRL_RGN_SNG[2]


when case STATE_NOX = REGENERATION
NOX_OUT_MES_MMV[i] = NOX_OUT_MES[i]
NOX_OUT_MDL_SENS =  NOX_OUT_MDL
NTLD_MDL_DIF_P = 0
NTLD_MDL_DIF_P_SNG[i] = 0
if(1) LV_NOX_RGN_CMPL = 1
then(1)     NTLD_MDL_DIF_I = 0
else(1)     if(2)     NTLD_MDL_DIF_I > 0
            then(2) $NTLD\_MDL\_DIF\_I_N = max(0,NTLD\_MDL\_DIF\_I_{N-1} - C\_NTLD\_DEC\_RGN)$
            else(2) $NTLD\_MDL\_DIF\_I_N = min(0, NTLD\_MDL\_DIF\_I_{N-1} + C\_NTLD\_DEC\_RGN)$
            endif(2)
endif (1)
NTLD_MDL_DIF_I_SNG[i] = NTLD_MDL_DIF_I

when case STATE_NOX = STOP
NOX_OUT_MES_MMV[i] = NOX_OUT_MES[i]
NOX_OUT_MDL_SENS =  NOX_OUT_MDL
NTLD_MDL_DIF_P = 0
NTLD_MDL_DIF_P_SNG[i] = 0
if(1) LV_NOX_RGN_CMPL = 1
then(1)     NTLD_MDL_DIF_I = 0
else(1)     if(2)     NTLD_MDL_DIF_I > 0
            then(2) $NTLD\_MDL\_DIF\_I_N = max(0,NTLD\_MDL\_DIF\_I_{N-1} - C\_NTLD\_DEC\_STOP)$
            else(2) $NTLD\_MDL\_DIF\_I_N = min(0,NTLD\_MDL\_DIF\_I_{N-1} + C\_NTLD\_DEC\_STOP)$
            endif(2)
endif(1)
NTLD_MDL_DIF_I_SNG[i] = NTLD_MDL_DIF_I

when case STATE_NOX = WAIT
NOX_OUT_MES_MMV[i] = NOX_OUT_MES[i]
NOX_OUT_MDL_SENS =  NOX_OUT_MDL
NTLD_MDL_DIF_I_SNG[i] and NTLD_MDL_DIF_I unchanged
NTLD_MDL_DIF_P_SNG[i] and NTLD_MDL_DIF_P unchanged
```

when state STATE_NOX = PASSIVE or WARMUP

NOX_OUT_MES_MMV[i] = NOX_OUT_MES[i]

NOX_OUT_MDL_SENS = NOX_OUT_MDL

NTLD_MDL_DIF_I = 0

NTLD_MDL_DIF_P = 0

NTLD_MDL_DIF_P_SNG[i] = 0

NTLD_MDL_DIF_I_SNG[i] = 0

Calculation of the total correction **(in all NOx_States)**

NTLD_MDL_DIF = NTLD_MDL_DIF_I + NTLD_MDL_DIF_P

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NTLD_MDL_DIF_I_MIN	1	8000...7FFFH	-0.5...0.49998	0.0153e-3	[-]
Minimum value of the I-part of the closed loop control NOx catalyst model					
C_NTLD_MDL_DIF_I_MAX	1	8000...7FFFH	-0.5...0.49998	0.0153e-3	[-]
Maximum value of the I-part of the closed loop control NOx catalyst model					
LC_NOX_RGN_CMPL	1	0...1H	0...1	1	[-]
Switch whether regenerations are "completely" or not					
C_NTLD_DEC_STOP	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Ramp gradient for a gradual deactivation of I-share during STATE_NOX = STOP					
C_NTLD_DEC_RGN	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Ramp gradient for a gradual deactivation of I-share during STATE_NOX = REGENERATION					
C_NTLD_CTRL_MIN	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Min NOx trap loading degree to start NOx catalyst model controller					
C_NOX_OUT_MES_GRD_MAX	1	0...FFFFH	0...1023.98437	0.015625	[mg/s]
Maximum gradient of measured NOx-flow downstream NOx-Cat.					
C_NOX_OUT_MES_MMV_MAX	1	0...FFFFH	0...1023.98437	0.015625	[mg/s]
Maximum value for "limited dynamic" for measured NOx-flow downstream NOx-Cat.					
LC_NOX_MDL_CTRL_ACT	1	0...1H	0...1	1	[-]
Switch to activate closed loop control fo the NOx-Catalyst model					
C_NOX_PPM_CTRL_MIN	1	0...05DCH	0...1500	1	[ppm]
Minimum threshold for measured NOx-flow downstream NOx-Cat					
IP_NTLD_MDL_DIF_I	6*6	0...FFFFH	0...0.49999	7.6294e-6	[s/mg]
LDPM_NOX_COR_RED_MMV	6	0...FFFFH	0...1023.98437	0.015625	[mg/s]
LDPM_TNT_MDL_MMV	6	0...FFFFH	0...1023.98437	0.015625	[°C]
Correlation constant for the I-share of the NTLD-controller					
IP_NTLD_MDL_DIF_P	6*6	0...FFFFH	0...0.49999	7.6294e-6	[s/mg]
LDPM_NOX_COR_RED_MMV	6	0...FFFFH	0...1023.98437	0.015625	[mg/s]
LDPM_TNT_MDL_MMV	6	0...FFFFH	0...1023.98437	0.015625	[°C]
Correlation constant for the P-share of the NTLD-controller					
C_CRLC_NOX_OUT_MES_MMV	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant for Moving mean value of measured NOx-Flow after NOx-Catalyst					
IP_CRLC_NOX_SENS	4	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDP_MAF_KGH_IP_CRLC_NOX_SENS	4	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Correlation constant for the first order time delay of the calculated NOx-Flow after NOx-Catalyst					
LC_INH_P_HLD	1	0...1H	0...1	1	[-]
Switch to inhibt holding of p-share					
C_NOX_DIF_MON	1	0...FFFFH	0...1023.98437	0.015625	[mg/s]
Max. deviation between model and NOx-Sensor					
C_NOX_DIF_MON_MIN	1	0...05DCH	0...1500	1	[ppm]
Threshold for monitoring deviation between model and NOx-Sensor					
C_T_NOX_MDL_INH_CTRL_MIN	1	0...FFH	0...0.99609	3.9063e-3	[-]
Threshold for Time controller must be active to allow regenerationsrequest					
C_T_NOX_MDL_CTRL_ENA_MIN	1	0...FFFFH	0...1310.7	0.02	[s]
Time controller is enabled					
LC_INH_NOX_MDL_CTRL_INI	1	0...1H	0...1	1	[-]
Initialisation value for LV_INH_NOX_MDL_CTRL_RGN_SNG[i]					

53.10 NOX Engine out Emission Adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NOX_AD_FAC	V/O	0...FFH	0...1.99218	0.0078125	[-]
NOx emission adaptation factor					
LV_NOX_AD_CMPL	V/O	0...1H	0...1	1	[-]
Logical value for active NOx emission adaptation					
NTL_RGN_ST_NOX_MES[NC_NT_NR]	V	0...FFFFH	0...10485.6	0.16	[mg]
Measured NOx reducing agent mass					
NOX_AD_FAC_SNG[NC_NT_NR]	V/O/S	0...FFH	0...1.99218	0.0078125	[-]
NOx emission adaptation factor					
LV_NOX_AD_DYW	V/O	0...1H	0...1	1	[-]
Logical value for active limited dynamics					
NOX_COR_S_MMV	V	0...FFFFH	0...1023.98437	0.015625	[mg/s]
Moving mean value of NOx emission					
MAF_FG_CYL_MMV	V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Moving mean value of mass air flow					
NOX_AD_FAC_RAW[NC_NT_NR]	V	0...FFH	0...1.99218	0.0078125	[-]
Raw value of NOx emission adaptation factor					
CTR_NOX_AD_FAC	V	0...FFH	0...255	1	[-]
adaptation counter for engine out emissions					

Input data:

NTL_DEC_INT_SWI[NC_NT_NR]	NT_O2_STC	C_FAC_O2_EQU	NTL_RGN_ST_NOX
NOX_OUT_MDL_INT	NOX_OUT_MES_INT[NC_NOX_SENS_CONF]	STATE_NOX	NOX_COR_S
MAF_FG_CYL	LV_AGI_VLD	NTLD_RGN_ST	LV_RGN_AGI_VLD[NC_NT_NR]
LV_NOX_NS_VLD[NC_NOX_SENS_CONF]	TNT_MDL_H	TNT_MDL_L	LV_NOX_AD_INH
LV_T_NS_VLD	LAMB_LS_UP[NC_CBK_EX_NR]	C_N_32_NT_AGI_MIN	C_N_32_NT_AGI_MAX
C_MFF_SP_S_NT_AGI_MIN	LV_HOM_AFS_REQ	N_32	C_MFF_SP_S_NT_AGI_MAX
VS	MFF_SP_S	NC_NT_NR	NC_NOX_SENS_CONF
NC_CBK_EX_NR			

FUNCTION DESCRIPTION:

General information:

Number of NOx catalyst:	NC_NT_NR	i = 1...NC_NT_NR
Number of NOx sensors:	NC_NOX_SENS_CONF	k = 1...NC_NOX_SENS_CONF
Number of lambda sensors:	NC_CBK_EX_NR	m = 1...NC_CBK_EX_NR

Typical configurations:
[NC_CBK_EX_NR:NC_NT_NR:NC_NOX_SENS_CONF] =[1:1:1], [2:2:2], [2:2:1], [2:1:1]

53.10.1 Limited Dynamics

Initialisation: -

Recurrence: 20 ms

Activation: at any state

Formula section:

```
if (1)      STATE_NOX = STOP --> !STOP          % initialisation of LV_NOX_AD_DYW
then(1)     LV_NOX_AD_DYW = 1
            NOX_COR_S_MMV (n) = NOX_COR_S
endif(1)

MAF_FG_CYL_MMV (n) = MAF_FG_CYL_MMV (n-1) * (1 - C_CRLC_MAF_FG_CYL) +
 C_CRLC_MAF_FG_CYL * MAF_FG_CYL

Limited dynamics for stratified operation:

if(1) STATE_NOX = LOAD and LV_HOM_AFS_REQ = 0
                        %monitoring limited dynamics during stratified
                mode
then(1)
NOX_COR_S_MMV (n) = NOX_COR_S_MMV (n-1) * (1 - C_CRLC_NOX_AD) +
                    C_CRLC_NOX_AD * NOX_COR_S
    if(2) |NOX_COR_S - NOX_COR_S_MMV| > C_NOX_AD_DYW or
            (LV_NOX_NS_VLD[k] = 0 and LV_T_NS_VLD = 1)
                                % valid if any condition for [k] is true
            or     LAMB_LS_UP[m] < C_LAMB_NOX_AD_DYW
                                % valid if any condition for [m] is true
    or      N_32 <= C_N_32_NT_AGI_MIN
            or     N_32 >= C_N_32_NT_AGI_MAX
            or     MFF_SP_S <= C_MFF_SP_S_NT_AGI_MIN
            or     MFF_SP_S >= C_MFF_SP_S_NT_AGI_MAX
    then(2)        LV_NOX_AD_DYW = 0
    else(2)        LV_NOX_AD_DYW remains unchanged
    endif(2)
else(1) LV_NOX_AD_DYW remains unchanged
```

```
endif(1)
```

Limited dynamics for regeneration:

```
if(1) STATE_NOX = REGENERATION %monitoring limited dynamics during stratified mode
then(1)
    if(2) | MAF_FG_CYL - MAF_FG_CYL_MMV | > C_MAF_FG_CYL_DYW
    or      VS ≤ C_VS_MIN_RGN_NOX_AD
    or      N_32 < C_N_32_MIN_RGN_NOX_AD
    or      MAF_FG_CYL < C_MAF_MIN_RGN_NOX_AD
    or      LV_PUC = 1
     then(2)        LV_NOX_AD_DYW = 0
    else(2)         LV_NOX_AD_DYW remains unchanged
    endif(2)
else(1) LV_NOX_AD_DYW remains unchanged
endif(1)
```

53.10.2 Emission Adaptation

Application conditions:

Initialisation: *At flashing: NOX_AD_FAC_SNG[i] = 1*

At reset: LV_NOX_AD_CMPL = 0
At reset: NOX_AD_FAC = (NOX_AD_FAC_SNG[1] + NOX_AD_FAC_SNG[2]) /NC_NT_NR

Recurrence: 100 ms

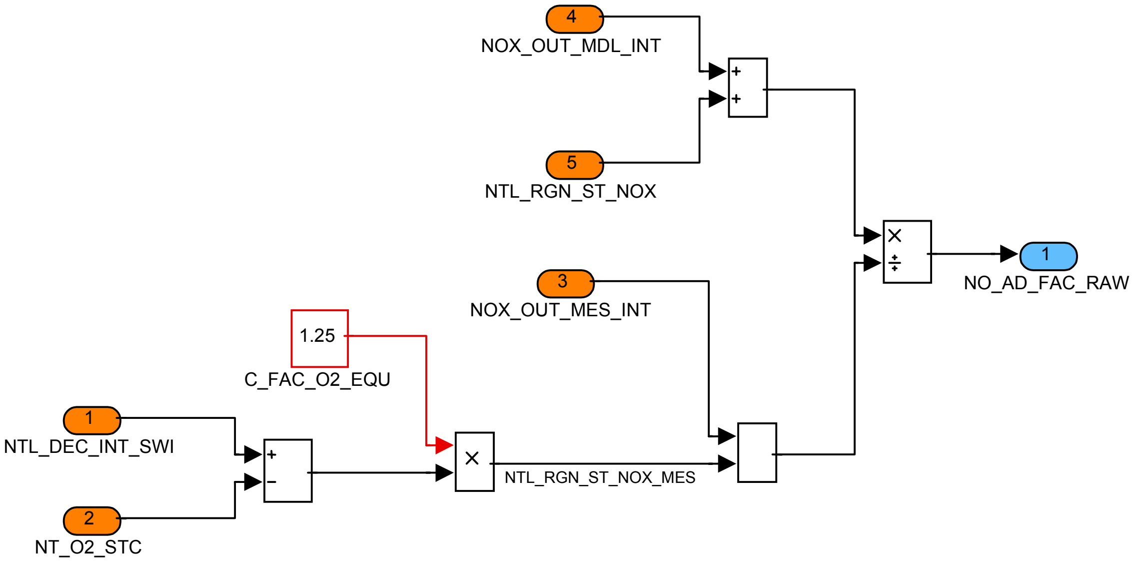
Activation: LC_NOX_AD_ACT = 1

Deactivation: **if** LC_NOX_AD_ACT = 0

then Initialisation*:* LV_NOX_AD_CMPL = 1

NOX_AD_FAC = (NOX_AD_FAC_SNG[1] +
NOX_AD_FAC_SNG[2])/ NC_NT_NR

Signal flow diagram:



Formula section:

```
if(1) LV_AGI_VLD = 0 -->1                %determination of aging factor is completed
then(1) LV_NOX_AD_CMPL = 0               % release of emission adaptation
endif(1)

LV_RGN_AGI_VLD_TMP = LV_RGN_AGI_VLD[1] and LV_RGN_AGI_VLD[2]
if(1) LV_RGN_AGI_VLD_TMP = 0 --> 1       % regeneration finished
    and LV_NOX_AD_DYW = 1                % limited dynamic conditions fulfilled
    and (LV_NOX_AD_CMPL = 0 or LC_NOX_AD_ON = 1)
                              % adaptation released or permanent adaptation choosen
    and NTLD_RGN_ST >= C_NTLD_MIN_NOX_AD
    and NTLD_RGN_ST < C_NTLD_MAX_NOX_AD
    and TNT_MDL_L >= C_TNT_MDL_NOX_AD_MIN
```

```
        and TNT_MDL_H <= C_TNT_MDL_NOX_AD_MAX
        and LV_NOX_AD_INH = 0
    then(1)
        NTL_RGN_ST_NOX_MES[i]=(NTL_DEC_INT_SWI[i] - NT_O2_STC/NC_NT_NR)
                                                          *C_FAC_02_EQU

        NOX_AD_FAC_RAW[i] = (NC_NT_NR *
                        (NTL_RGN_ST_NOX_MES[i] +
                        NC_NOX_SENS_CONF/NC_NT_NR * NOX_OUT_MES_INT[k])) /
                        (NTL_RGN_ST_NOX + NOX_OUT_MDL_INT)

        NOX_AD_FAC_SNG[i](n)=NOX_AD_FAC_SNG[i](n-1) + C_CRLC_NOX_AD_FAC *
                                    (NOX_AD_FAC_RAW[i] - 1)

        Increment CTR_NOX_AD_FAC

        if(2) |NOX_AD_FAC_RAW[i] - NOX_AD_FAC_SNG[i]| < C_NOX_AD_FAC_DIF
                                        (for each bank)
            and CTR_NOX_AD_FAC > C_CTR_NOX_AD_FAC_MIN
        then(2) LV_NOX_AD_CMPL = 1
            CTR_NOX_AD_FAC = 0
            NOX_AD_FAC = (NOX_AD_FAC_SNG[1] + NOX_AD_FAC_SNG[2])/
                                                          NC_NT_NR
        else(2) NOX_AD_FAC remains unchanged
        endif(2)
    else(1)     NOX_AD_FAC remains unchanged
    endif(1)
```

53.10.3 Manual initialisation of emission adaptation factor

Recurrence: *100 ms*

Activation: at any state (also if *LC_NOX_AD_ACT = 0)*

```
if LC_NOX_AD_FAC_INI = 1
```

```
then NOX_AD_FAC_SNG[i] = C_NOX_AD_FAC_INI
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NOX_AD_ACT	1	0...1H	0...1	1	[-]
Switch to activate NOx emission adaptation					
C_CRLC_NOX_AD	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant of NOx emission					
C_NOX_AD_DYW	1	0...FFFFH	0...1023.98437	0.015625	[mg/s]
NOx emission limit for limited dynamics					
C_CRLC_MAF_FG_CYL	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation constant of air mass flow					
C_MAF_FG_CYL_DYW	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
air mass flow limit for limited dynamics					
LC_NOX_AD_ON	1	0...1H	0...1	1	[-]
Switch to enable permanent NOx emission adaptation					
C_NTLD_MIN_NOX_AD	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
min. NOx trap loading degree for starting adaptation					
C_CRLC_NOX_AD_FAC	1	0...FFH	0...1	3.9216e-3	[-]
Correlation constant of NOX emission adaptation factor					
C_NOX_AD_FAC_DIF	1	0...FFH	0...1.99218	0.0078125	[-]
Max. difference of NOX emission adaptation factor for finishing adaptation					
C_CTR_NOX_AD_FAC_MIN	1	0...FFH	0...255	1	[-]
min value of adaptations counter to finish adaptation					
C_TNT_MDL_NOX_AD_MIN	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabing/disabling emission adaptation					
C_TNT_MDL_NOX_AD_MAX	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabing/disabling emission adaptation					
C_NTLD_MAX_NOX_AD	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
max. NOx trap loading degree for starting adaptation					
LC_NOX_AD_FAC_INI	1	0...1H	0...1	1	[-]
Switch for manual initialization of Emission dadaptation factor					
C_NOX_AD_FAC_INI	1	0...FFH	0...1.99218	0.0078125	[-]
Emission dadaptation factor initialization value					
C_LAMB_NOX_AD_DYW	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda lthreshold for limited dynamics					
C_VS_MIN_RGN_NOX_AD	1	0...FFH	0...255	1	[km/h]
Minimum vehicle speed for use adaptation value					
C_N_32_MIN_RGN_NOX_AD	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed for use adaptation value					
C_MAF_MIN_RGN_NOX_AD	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Minimum mass air flow for use adaptation value					

53.11 NOx engine out emission adaptation (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_NOX_AD_INH	V/O	0...1H	0...1	1	[-]
Inhibiting of NOx engine out emission adaptation					

Input data:

LV_ST_END			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_INH_NOX_AD	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx emission adaptation					

General information:

The NOx engine out emission adaptation will be inhibited, if any used input signal is not valid. Additionally, the adaptation may be enabled or disabled without any cross dependencies.

Application conditions:

Initialisation: at reset and at deactivation: LV_NOX_AD_INH = 1

Recurrence: 100 ms

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

```
case selection on C_STATE_INH_NOX_AD
  C_STATE_INH_NOX_AD = 0 ('AUTO'):
    LV_NOX_AD_INH = 0                    (no cross dependencies)

  C_STATE_INH_NOX_AD = 1 ('DISABLE'):
    LV_NOX_AD_INH = 1

  C_STATE_INH_NOX_AD = 2 ('ENABLE'):
    LV_NOX_AD_INH = 0
end case selection
```

53.12 NOx Catalyst Aging

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NT_AGI	O/V/S	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor					
NT_AGI_SUL	O/V/S	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor due to sulphur load					
NT_AGI_SUL_SNG[NC_NT_NR]	O/V/S	0...FFFFFFFFH	0...1	2.3283E-10	-
NOx trap aging factor due to sulphur load (bench selective)					
NT_AGI_THERMO	O/V/S	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor due to thermal aging					
NT_AGI_THERMO_SNG[NC_NT_NR]	O/V/S	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor due to thermal aging (bench selective)					
NT_AGI_OBS_SNG[NC_NT_NR]	O/V	0...FFFFH	0...0.99998474	1.52588E-5	-
quick aging monitoring value (bench selective)					
NT_AGI_OBS	O/V	0...FFFFH	0...0.99998474	1.52588E-5	-
quick aging monitoring value					
NT_AGI_S_RED	O/V	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor					
NT_AGI_NTLD	O/V	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor					
LV_NT_AFS_REQ_AGI	O/V/S	0...1H	0...1	1	-
Logical value for the request of lambda=1 operation					
LV_SO2P_REQ_1	O/V/S	0...1H	0...1	1	-
Request of a desulfation					
LV_SO2P_REQ_2	O/V/S	0...1H	0...1	1	-
Request of a desulfation (forces catalyst heating)					
LV_AGI_VLD	O/V	0...1H	0...1	1	-
Logical value for completed aging adaption					
LV_CDN_NT_AGI_THERMO_AD	O/V	0...1H	0...1	1	-
flag: thermal aging adaptation enabled (due to low sulphur load)					
LV_NT_AGI_ENA	O/V	0...1H	0...1	1	-
aging factor (sulphur or thermal) calculation enabled					
LV_EGR_RATIO_SP_AGI_ACT	O/V	0...1H	0...1	1	-
request to close EGR valve for better NT aging adaptation performance					
NT_SUL_UPD	O/V	0...FFFFH	0...1.04856E+4	0.16	mg
interface variable for update of sulphur model by aging function					
CTR_NT_AGI_AD_CMPL	O/V	0...FEH	0...254	1	-
counter of completed aging adaptation cycles during driving cycle					
NT_SUL_AGI	O/V	0...FFFFH	0...1.04856E+4	0.16	mg
Interface variable for update of sulphur release model by aging function					
CTR_NT_AGI_AD_CMPL_SUM	O/V/S	0...FFFFH	0...6.5535E+4	1	-
counter of completed aging adaptation					
LV_NT_AGI_ENA_AFS	O/V	0...1H	0...1	1	-
value for HOM_REQ befor aging adaptation					
LV_NT_AGI_ENA_AFS_TMP	O/V	0...1H	0...1	1	-
Temorary value for HOM_REQ befor aging adaptation					
NT_SUL_AGI_SNG[NC_NT_NR]	O/V	0...FFFFH	0...1.04856E+4	0.16	mg
Interface variable for update of sulphur release model by aging function (bench selective)					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_NT_AFS_REQ_AGI_TMP_3	O/V/S	0...1H	0...1	1	-
Logical value for the request of lambda =1 operation					
LV_SO2P_REQ_FQ	O/V/S	0...1H	0...1	1	-
Logical value for active FQ adaptation					
FAC_SU_CONC_FUEL	O/V	0...FFH	0...255	1	-
Interface factor to sulphur model for calibrating fuel sulphur concentration					
CTR_NT_AGI_SO2P_FQ	O/V/S	0...FFFFH	0...6.5535E+4	1	-
counter of completed aging adaptation during FQ adaptation					
NT_AGI_SO2P_FQ_SUM	O/V/S	0...FFFFFFH	0...255.999985	1.52588E-5	-
Sum of NOx trap aging factor during FQ adaptation					
LV_NT_AGI_EXT_ADJ	O/V	0...1H	0...1	1	-
External data for NOx trap aging values available					
FAC_NT_AGI_LIM	O/V/S	0...FFFFH	0...0.99998474	1.52588E-5	-
Limited NOx trap aging factor					
FAC_NT_AGI_THERMO_SNG[NC_NT_NR]	O/V	0...FFFFH	0...0.99998474	1.52588E-5	-
Temporary aging factor					
LV_NT_AGI_EXT_RES	O/V	0...1H	0...1	1	-
External reset of data for NOx trap aging values					
NT_AGI_BAS[NC_NT_NR]	O/V	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor base value (not defined in binary Lambda-Sensor mode)					
LV_SWI_MDL_THERMO_AGI_ACT	V	0...1H	0...1	1	-
Switch to modeled aging factor before sulphur aging model caluclation active					
LV_NOX_NS_SIG_CHK_NT_AGI	V	0...1H	0...1	1	-
Check of the NOx signal shift and gain adaptation					
NOX_EFF[NC_NT_NR]	V	0...FFFFH	0...0.99998474	1.52588E-5	-
actual NOx trap efficiency (not defined in binary Lambda-Sensor mode) (bench selective)					
LV_AGI_AD_TRIG	V	0...1H	0...1	1	-
Logical value for the request of adaptaion of aging factor					
LV_NT_AGI_BAS_ENA	V	0...1H	0...1	1	-
basic aging factor calculation enabled					
LV_NT_AGI_ENA_TMP	V	0...1H	0...1	1	-
temporary value for LV_NT_AGI_ENA calculation					
LV_NT_AGI_ENA_TNT	V	0...1H	0...1	1	-
temperature initialisation condition for aging adaptation given					
CTR_NT_AGI_AD	V	0...FEH	0...254	1	-
counter of aging adaptation attempts (with closed EGR valve)					
NTLD_AGI[NC_NT_NR]	V	0...FFFFH	0...0.99998474	1.52588E-5	-
"reverse" NOx trap loading degree					
LV_NT_AFS_REQ_AGI_TMP_2	V	0...1H	0...1	1	-
Logical value for the request of lambda=1 operation					
LV_NT_AFS_REQ_AGI_TMP_1	V	0...1H	0...1	1	-
Logical value for the request of lambda=1 operation					
DIF_CTR_NT_AGI_CMPL_SUM	V	0...FFFFH	0...6.5535E+4	1	-
Completed aging adaptation due to FQ adaptation					
NT_AGI_SO2P_FQ	V	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor					
LV_NT_AGI_TRIG_ENA	V	0...1H	0...1	1	-
Enabling of trigger for aging calculation					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_NOX_OLD_AGI	-	0H 1H 2H 3H 4H 5H	PASSIV LOAD REGENERATION WAIT STOP WARMUP	1	-
Old value of STATE_NOX for edge detection					

Input data:

LV_SO2P_ACT	LV_NT_AGI_OBS_ENA	C_NT_STC_BAS_MAX_RNG_H	LV_NOX_SENS
NT_AGI_AV	LV_RGN_REQ_NOX_SENS	LV_INH_NT_AGI	NOX_CONC_RED_MMV_RGN_ST
NOX_NS_AD_RGN_ST[NC_NT_NR]	NT_AGI_INC_SO2P[NC_NT_NR]	NTLD_MDL	NTL_RGN_ST_NOX
NT_EFF_COR_RGN_ST	LV_NT_HOM_INI	LV_NOX_AD_CMPL	NT_SUL
VS	TNT_MDL_MV	NT_STC_BAS_RNG_H	LV_NT_AGI_INI_EXT_ADJ_NEW_CAT
STATE_NOX	LV_EF	LC_NOX_RGN_CMPL	LV_S_ACT
LV_NOX_SENS_NOX_VLD_RGN_ST[NC_NT_NR]	LV_INH_NT_AGI_BAS	T_AFL	NT_SUL_H
LV_NT_AGI_OBS_EGR_ENA	NC_NT_NR	LV_INH_NT_AGI_TNT_BAS	LV_NT_SO2P_EXT_ADJ_ACT
LV_RGN_REQ_NTLD	NTLD	LV_SO2P_REQ	LV_SUL_EXT_ADJ
NT_AGI_SO2P_FQ_SUM_EXT_ADJ	NT_AGI_SUL_SNG_EXT_ADJ[NC_NT_NR]	NT_AGI_THERMO_SNG_EXT_ADJ[NC_NT_NR]	CTR_NT_AGI_AD_CMPL_SUM_EXT_ADJ
CTR_NT_AGI_SO2P_FQ_EXT_ADJ	LV_NT_AFS_REQ_AGI_TMP_3_EXT_ADJ	LV_SO2P_REQ_2_EXT_ADJ	LV_SO2P_REQ_FQ_EXT_ADJ
DIST_NT	FAC_NT_AGI_LIM_EXT_ADJ	FAC_NT_AGI_MDL_THERMO	STATE_ERR_NS_OBD_2[NC_NOX_SENS_CONF]
LV_ERR_NS_OBD_1[NC_NOX_SENS_CONF]	LV_SUL_EXT_RES	NC_NOX_SENS_CONF	DIST_NS_NEW
RATIO_MMV_NS_SHIFT_DIAG[NC_NOX_SENS_CONF]	FAC_NOX_NS_AD[NC_NOX_SENS_CONF]		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_NT_AGI_OBS	1	0...FFFFH	0...0.99998474	1.52588E-5	-
filter constant for NT_AGI_OBS calculation					
C_CRLC_NT_AGI_SUL	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Correlation constant for the calculation of the aging factor due sulphur load					
C_CRLC_NT_AGI_THERMO_DEC	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Correlation constant for the calculation of the thermal aging factor (decreasing factor)					
C_CRLC_NT_AGI_THERMO_INC	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Correlation constant for the calculation of the thermal aging factor (increasing factor)					
C_CTR_NT_AGI_AD_CMPL_MAX	1	0...FFH	0...255	1	-
max. number of successfull adapt. per cycle					
C_CTR_NT_AGI_AD_MAX	1	0...FFH	0...255	1	-
max. number of attemps for adaptation (incl. successful attempts)					
C_CTR_NT_AGI_SO2P_FQ_MIN	1	0...FFFFH	0...6.5535E+4	1	-
Counter thresholdf or aging adaptation after FQ-desulfation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DIST_NT_AGI_LIM	1	0...FFFFH	0...5.2428E+5	8	km
mileage threshold for limited NOx trap aging factor					
C_DIST_NT_AGI_MDL_1	1	0...FFFFH	0...5.2428E+5	8	km
Threshold of distance for switching to modeled factor					
C_DIST_NT_AGI_MDL_2	1	0...FFFFH	0...5.2428E+5	8	km
Threshold of distance for switching to modeled factor, included sulfur model					
C_FAC_AGI_THD_SENS_DIF_MDL_1	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Threshold for difference between aging factors					
C_FAC_AGI_THD_SENS_DIF_MDL_2	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Threshold for difference between aging factors, including sulfur model					
C_FAC_NOX_NS_AD_AGI_MDL_MAX	1	0...FFFFH	0...127.998047	0.00195313	-
Maximal threshold of NOx signal gain adaptation to switch to modeled aging factor					
C_FAC_NOX_NS_AD_AGI_MDL_MIN	1	0...FFFFH	0...127.998047	0.00195313	-
Minimal threshold of NOx signal gain adaptation to switch to modeled aging factor					
C_FAC_NT_AGI_MDL_ACT_1	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Threshold of NT aging for switching to modeled factor					
C_FAC_NT_AGI_MDL_ACT_2	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Threshold of NT aging for switching to modeled factor, included sulfur model					
C_FAC_SU_CONC_FUEL	1	0...FFH	0...255	1	-
Interface factor to sulphur model for calibrating fuel sulphur concentration					
C_NOX_RGN_ST_MAX_AGI	1	0...5DCH	0...1.5E+3	1	ppm
max. NOx sensor signal threshold for enabling an aging adaptation					
C_NOX_RGN_ST_MIN_AGI	1	0...5DCH	0...1.5E+3	1	ppm
min. NOx sensor signal threshold for enabling an aging adaptation					
C_NTLD_SO2P_FQ	1	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx loading degree to stop aging adaptation after FQ-desulfation					
C_NTLD_TRIG_AGI_MAX_RGN_NS	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Maximum NOx trap loading degree for aging calculation at regeneration request by NOx sensor					
C_NTLD_TRIG_AGI_MAX_RGN_NTLD	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Maximum NOx trap loading degree for aging calculation at regeneration request by NTLD threshold					
C_NTLD_TRIG_AGI_MIN_RGN_NS	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Minimum NOx trap loading degree for aging calculation at regeneration request by NOx sensor					
C_NTLD_TRIG_AGI_MIN_RGN_NTLD	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Minimum NOx trap loading degree for aging calculation at regeneration request by NTLD threshold					
C_NT_AGI_1	1	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor threshold for the desulfation request 1					
C_NT_AGI_2	1	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor threshold for the desulfation request 2					
C_NT_AGI_3	1	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor threshold for the request of lambda=1 operation					
C_NT_AGI_INI[NC_NT_NR]	1	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor initializing value (branch selective)					
C_NT_AGI_NTLD_MAN	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Manual set of NT_AGI_NTLD					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NT_AGI_OBS_MIN	1	0...FFFFH	0...0.99998474	1.52588E-5	-
aging monitoring minimum threshold to request lambda=1 operation					
C_NT_AGI_OBS_MIN_2	1	0...FFFFH	0...0.99998474	1.52588E-5	-
aging monitoring minimum threshold 2 to request l 1 operation					
C_NT_AGI_OBS_MIN_EGR_ERR	1	0...FFFFH	0...0.99998474	1.52588E-5	-
aging monitoring minimum threshold to request l 1 operation if EGR valve stucked close					
C_NT_AGI_OBS_MIN_IS	1	0...FFFFH	0...0.99998474	1.52588E-5	-
aging monitoring minimum threshold to request lambda=1 operation during idle speed					
C_NT_AGI_S_RED_MAN	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Manual set of NT_AGI_S_RED					
C_NT_AGI_THERMO_SO2P_FQ_INI	1	0...FFFFH	0...0.99998474	1.52588E-5	-
Initial aging value after FQ-desulfation					
C_NT_SUL_MAX_1	1	0...FFFFH	0...1.04856E+4	0.16	mg
sulphur model threshold to request set desulfation request 1					
C_NT_SUL_MAX_DEAC	1	0...FFFFH	0...1.04856E+4	0.16	mg
sulphur model threshold to request reset desulfation request 2 during active desulfation					
C_NT_SUL_SO2P_FQ	1	0...FFFFH	0...1.04856E+4	0.16	mg
sulphur model value to start FQ-desulfation					
C_RATIO_NS_SHIFT_AGI_MDL_MAX	1	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Maximal threshold of NOx signal shift to switch to modeled aging factor					
C_RATIO_NS_SHIFT_AGI_MDL_MIN	1	8000...7FFFH	-1...0.99996948	3.05176E-5	-
Minimal threshold of NOx signal shift to switch to modeled aging factor					
C_STATE_ERR_NS_MDL_SWI	1	0...FFFFH	0...6.5535E+4	1	-
Choice of OBD 2 diagnosis for switching from sensor to model					
C_STATE_NT_AGI_NTLD	1	0H 1H 2H 3H 4H	ALL THERMO SUL MAN LIM	1	-
Mode to switch NT_AGI_NTLD					
C_STATE_NT_AGI_S_RED	1	0H 1H 2H 3H 4H	ALL THERMO SUL MAN LIM	1	-
Mode to switch NT_AGI_S_RED					
C_SUL_THD_AGI_THERMO	1	0...FFFFH	0...1.04856E+4	0.16	mg
threshold for reset of thermal aging adaptation / SUL no more neglectable					
C_TNT_MDL_AGI_ENA_TNT	1	0...FFH	0...1.02E+3	4	°C
threshold for temperature initialisation condition					
C_TNT_MDL_NT_AGI_OBS_INI	1	0...FFH	0...1.02E+3	4	°C
threshold for resetting the quick aging monitoring values					
C_T_AFL_MAX_AGI_ENA	1	0...FFFFH	0...1.3107E+3	0.02	s
maximum lean phase time without aging enabling bit yet set					
C_VS_OBS_MIN	1	0...FFH	0...255	1	km/h
Vehicle speed threshold to switch aging monitoring					
LC_AGI_AD_TRIG_MAN	1	0...1H	0...1	1	-
Manual trigger for starting aging adaption					
LC_ENA_SUL_UPD	1	0...1H	0...1	1	-
Logical constant to enable adaptation between sulfur modell and aging factor detemination					
LC_NOX_AD_CMPL_OFF	1	0...1H	0...1	1	-
switch to ignore NOx raw emission adaptation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_NT_AGI_AD_CTR_RST_TNT	1	0...1H	0...1	1	-
switch for resetting the adaptation cycle counters depending on NT temperature					
LC_NT_AGI_EGR_CLOSE	1	0...1H	0...1	1	-
switch: close EGR during aging adaptation					
LC_NT_AGI_INI_MAN	1	0...1H	0...1	1	-
Manual reset of the logical variables for desulfation					
LC_NT_AGI_MAN	1	0...1H	0...1	1	-
Logical constant for manual switch of aging factor determination					
LC_NT_AGI_MIN	1	0...1H	0...1	1	-
Logical constant for selecting NT_AGI					
LC_NT_AGI_NT_SUL_MPG	1	0...1H	0...1	1	-
Logical constant to enable calibration of NT_SUL with aging determination					
LC_NT_AGI_OBS_INI	1	0...1H	0...1	1	-
switch for manual initialisation of quick aging monitoring values					
LC_NT_AGI_SUL_INI	1	0...1H	0...1	1	-
NOx trap aging factor due sulphur load initializing value (branch selective)					
LC_NT_AGI_THERMO_INI	1	0...1H	0...1	1	-
NOx trap aging factor due thermal aging initializing value (branch selective)					
LC_SUL_SWI_1_THERMO_ACT	1	0...1H	0...1	1	-
Manual activation of switching to modeled aging factor at calculation of sulphur aging					
LC_SUL_THERMO_COR	1	0...1H	0...1	1	-
Logical constant for switching betwenn absolut and relativ NT_SUL					
LC_SUL_UPD	1	0...1H	0...1	1	-
Logical constant to enable calibration of NT_AGI_SUL with sulfur modell					
ID_FAC_NT_AGI_DIF_NS	6	0...FFFFH	0...0.99998474	1.52588E-5	-
LDP_DIST_NS_DIF_ID_FAC_NT_AGI	6	0...FFFFH	0...5.2428E+5	8	km
Safety margin for aged NOx sensor for thermal NT aging factor derived from NOx sensor					
IP_FAC_NT_AGI	6	0...FFFFH	0...1.99996948	3.05176E-5	-
LDP_TNT_MDL_MV_IP_FAC_NT_AGI	6	0...FFFFH	0...1.02398E+3	0.015625	°C
Correction factor for NT_AGI depending on NOx catalyst temperature					
IP_NT_AGI_NT_SUL	6	0...FFFFH	0...0.99998474	1.52588E-5	-
LDP_NT_SUL_IP_NT_AGI_NT_SUL	6	0...FFFFH	0...1.04856E+4	0.16	mg
Correlation sulphur loading - aging factor due sulphur load					
IP_NT_SUL_H_MIN	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDPM_NT_AGI_THERMO	6	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx Trap sulphur loading threshold (for high sulphured fuel) for desulfation request 2					
IP_NT_SUL_MAX_2	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDPM_NT_AGI_THERMO	6	0...FFFFH	0...0.99998474	1.52588E-5	-
sulphur model threshold to request set desulfation request 2					
IP_NT_SUL_MAX_2_DEAC	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDPM_NT_AGI_THERMO	6	0...FFFFH	0...0.99998474	1.52588E-5	-
sulphur model threshold to request reset desulfation request 2					
IP_NT_SUL_MAX_3	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDPM_NT_AGI_THERMO	6	0...FFFFH	0...0.99998474	1.52588E-5	-
sulphur model threshold to request l 1 operation					
IP_NT_SUL_NT_AGI	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDP_NT_AGI_SUL_IP_NT_SUL_NT_AGI	6	0...FFFFH	0...0.99998474	1.52588E-5	-
Correlation aging factor due sulphur load - sulphur loading					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_NTLD_AGI	8x8	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_NOX_EFF	8	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_TNT_MDL_NOX_NT	8	0...FFFFH	0...1.02398E+3	0.015625	°C
NOx trap loading degree , has to be inverse to IP_NT_EFF_BAS					
IP_NTLD_AGI_EF_CLOSE	8x8	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_NOX_EFF	8	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_TNT_MDL_NOX_NT	8	0...FFFFH	0...1.02398E+3	0.015625	°C
NOx trap loading degree , has to be inverse to IP_NT_EFF_BAS_EF_CLOSE					

Export actions:

ACTION_NOXM_CleanNTAdaptAgi()
Initialization of NOx trap adaptation values of aging module
ACTION_NOXM_WriteAgingExtAdj()
New external values for aging calculation available

53.12.1 General information

The NOx catalyst aging function can handle single branched and twin branched exhaust gas lines with the configuration data NC_NT_NR defined in the NOx catalyst management module.

If NC_NT_NR=1, just one exhaust line is taken into account, if NC_NT_NR=2, the software calculates the twin branch version.

i characterises the variable of branches which can be counted up

The NOx storage catalyst looses its storage capacity due to thermal aging and sulfur poisoning.

The aging is determined by the NOx-signal of a NOx-sensor, mounted downstream the NOx catalyst. The information, which kind of sensor is installed is included in LV_NOX_SENS.

The aging state of the catalyst is evaluated by an aging factor NT_AGI_SNG_i. The value of this factor is 1 in the case of a fresh catalyst, a value near zero indicates a destroyed catalyst. The resulting aging factor NT_AGI is the minimum of all existing NT_AGI_SNG_i.

The aging factors NT_AGI_SNG_i mainly will be decreased by the aging determination in this routine. Due to desulfation, the aging state of the catalyst can be improved (storage capacity increases). This increase is calculated by the desulfation function and passed to the aging function by NT_AGI_INC_SO2P. This value is set to 0 if no desulfation is active, otherwise it includes the incremetation of the aging factor per time step. This increment is added to the aging factor at every call of the aging function.

Handling of the aging states

When the resulting aging factor NT_AGI falls below the threshold C_NT_AGI_1 the desulfation request bit LV_SO2P_REQ_1 is set.

When the aging factor falls below the threshold C_NT_AGI_2 the desulfation request bit LV_SO2P_REQ_2 is set, this means an active catalyst heating is required.

When the aging factor falls below the threshold C_NT_AGI_3 a desulfation is requested. If this desulfation is not successful lean operation has to be forbidden, the request bit LV_NT_AFS_REQ_AGI for lambda=1 operation due to aging state is set.

The request for a desulfation or for operation in lambda=1 mode can be manually reseted by LC_NT_AGI_INI_MAN, in this case NT_AGI_SNG_i is initialized with C_NT_AGI_INI_i.

The values NT_AGI, NT_AGI_i have to be stored in the non-volatile memory.

Calculation of the base aging factor

The basis aging factor is determined by the NOx-signal of the NOx-Sensor NOX_PPM_i and the calculated reduced NOx raw emissions NOX_CONC_RED. In the moment the first NOx-Sensor detects an inadmissible high NOx concentration downstrem the NOx catalyst and as a consequence LV_RGN_REQ_NOX_SENS is switched from 0 to 1, the actual storage capacity is calculated and compared with the base storage capacity of a fresh catalyst. The aging factor is calculated using the ratio of the actual storage capacity and the base storage capacity NT_STC. The calculation is carried out once when the regeneration starts.

Signal flow diagram

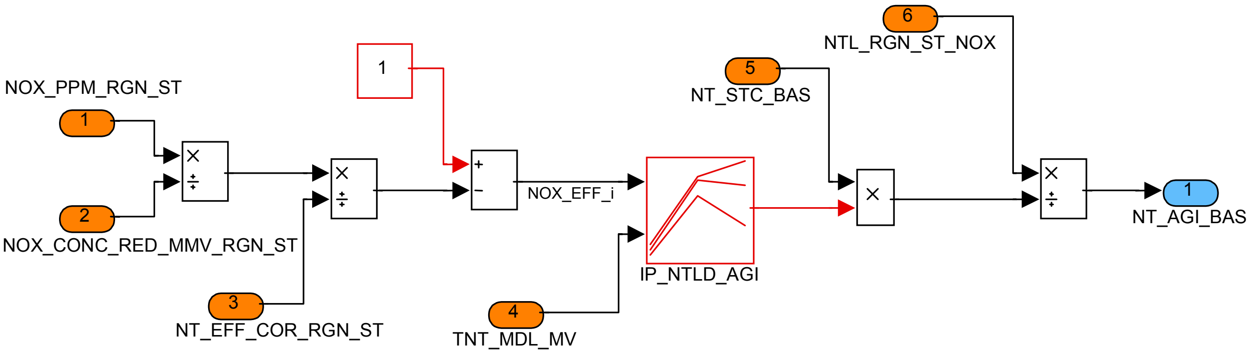


figure: Aging factor calculation // NOx- Sensor installed

Application Condition

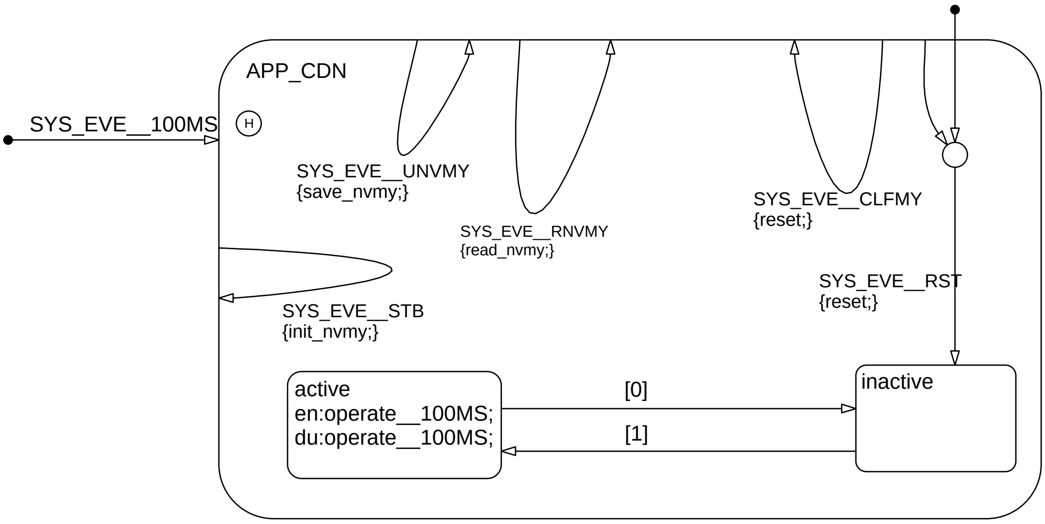
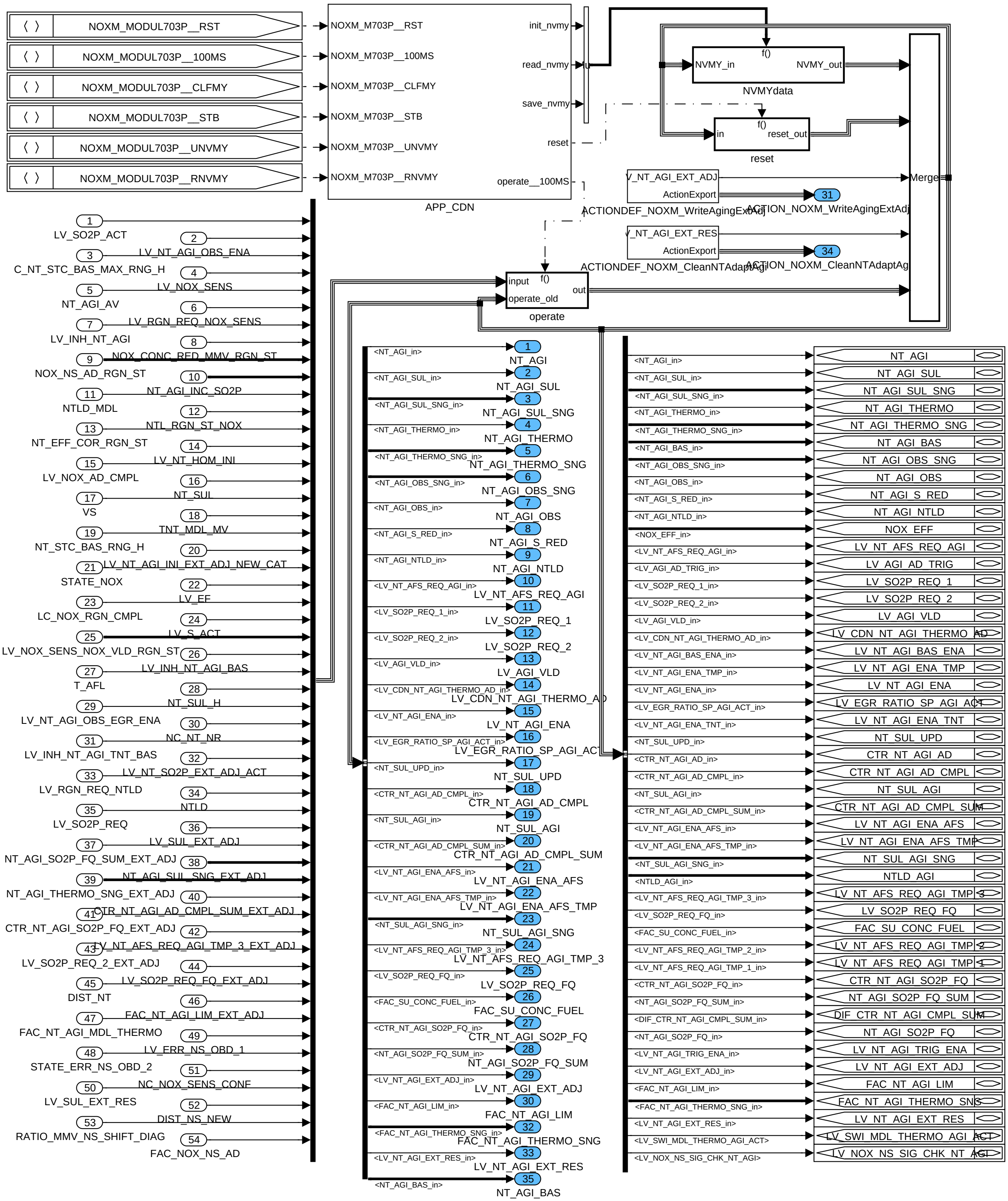


Figure 88 NOXM_MODUL703P/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 16−Feb−2007

Figure 89 NOXM_MODUL703P

53.12.1.1 SUBFUNCTION: ACTIONDEF_NOXM_CleanNTAdaptAgi

Description for ACTION_NOXM_CleanNTAdaptAgi

ACTION_NOXM_CleanNTAdaptAgi()					
Initialization of NOx trap adaptation values of aging module					
Parameter	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**

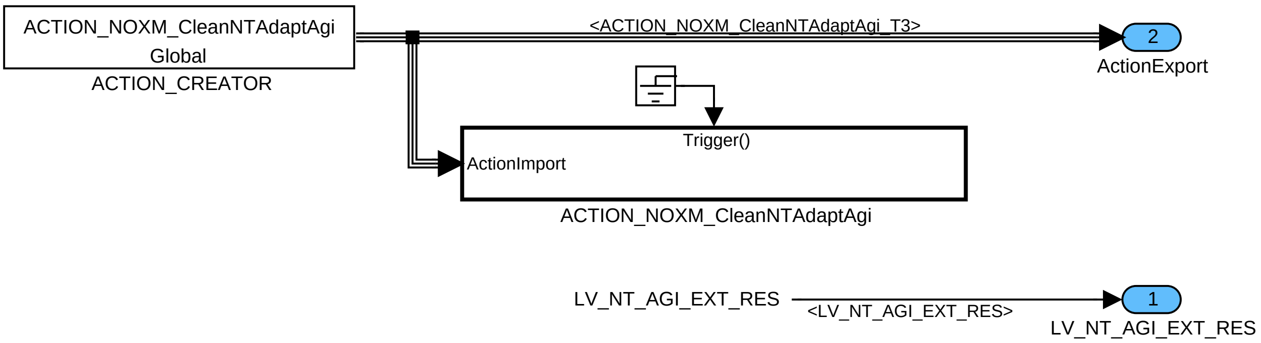


Figure 90 NOXM_MODUL703P/ ACTIONDEF_NOXM_CleanNTAdaptAgi

The flag LV_NT_AGI_EXT_RES is set by this action. It shows that the non-volatile values have to be reset inside next 100 ms recurrence (if sulfur model values are already set to zero).

Figure 91 NOXM_MODUL703P/ ACTIONDEF_NOXM_CleanNTAdaptAgi/ ACTION_NOXM_CleanNTAdaptAgi

53.12.1.2 SUBFUNCTION: ACTIONDEF_NOXM_WriteAgingExtAdj

Description for ACTION_NOXM_WriteAgingExtAdj

ACTION_NOXM_WriteAgingExtAdj()					
New external values for aging calculation available					
Parameter	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**

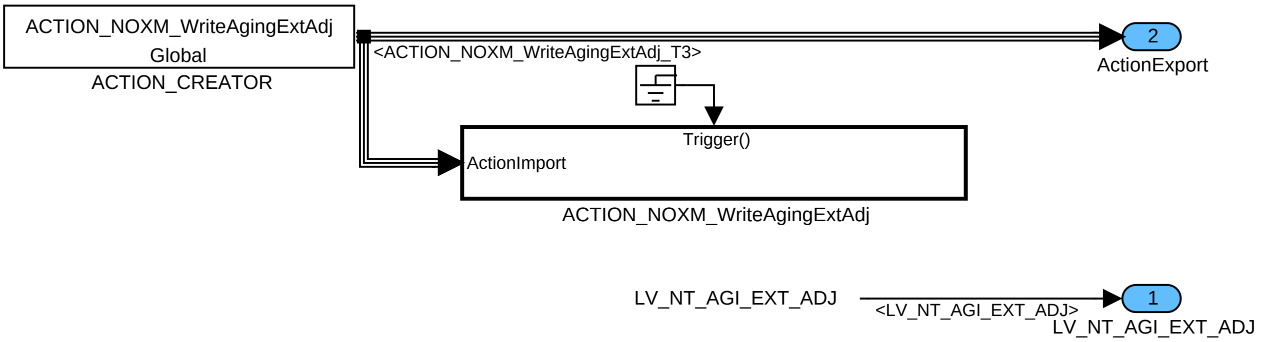


Figure 92 NOXM_MODUL703P/ ACTIONDEF_NOXM_WriteAgingExtAdj

The flag LV_NT_AGI_EXT_ADJ is set by this action. It shows that the external values have to be used inside next 100 ms recurrence (if sulfur model values are already updated).

Figure 93 NOXM_MODUL703P/ ACTIONDEF_NOXM_WriteAgingExtAdj/ ACTION_NOXM_WriteAgingExtAdj

53.12.1.3 OPERATE

This picture shows the main sequence of the aging loop. Both calibration values allow to switch between manual initializing and normal aging calculation.

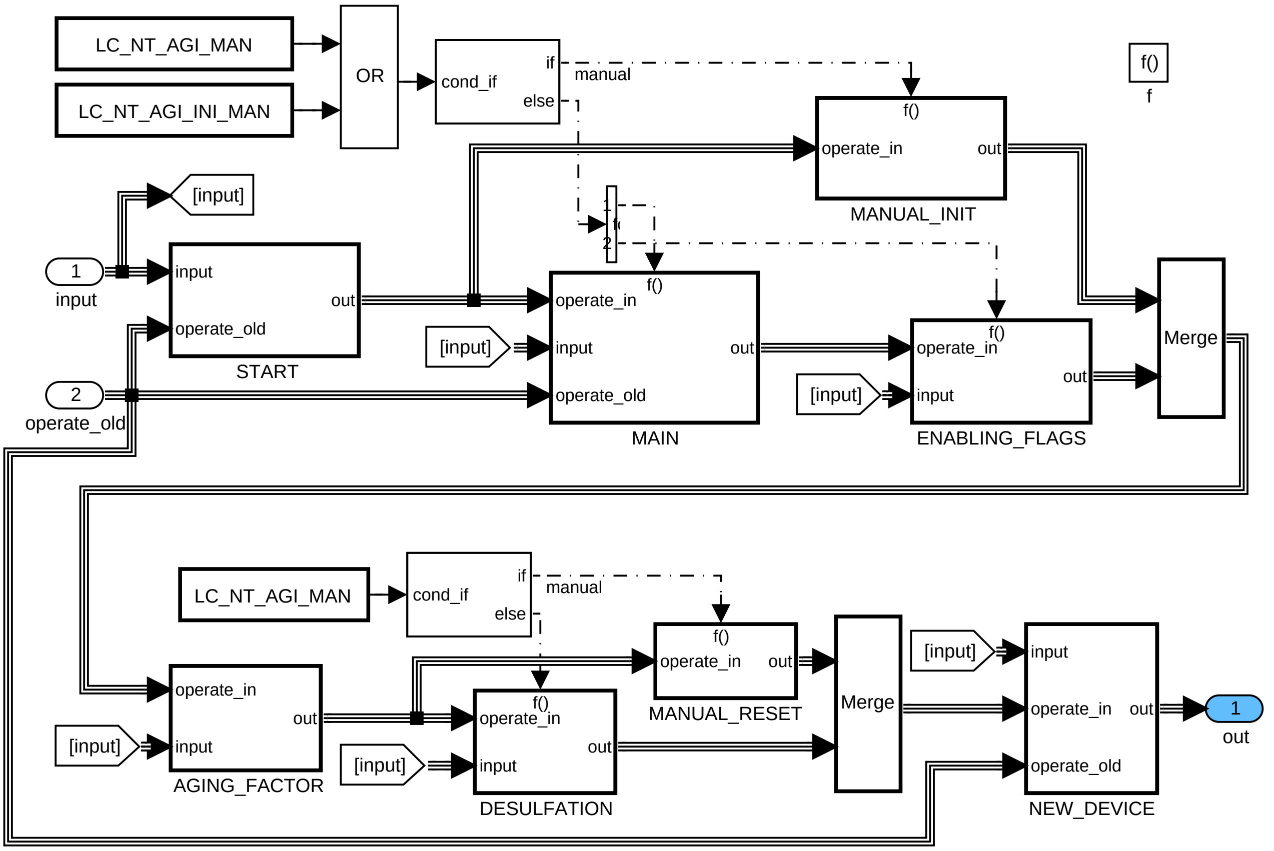


Figure 94 NOXM_MODUL703P/ operate

START: calculation of start conditions

- definition of the area for thermal aging adaptation
- temperature initalisiation condition for enabling aging adaptation
- conditions and trigger for base aging factor calculation

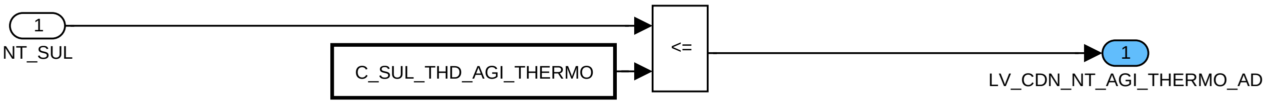


Figure 95 NOXM_MODUL703P/ operate/ START/ CLC1

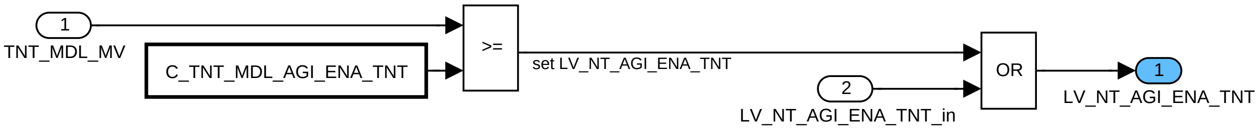


Figure 96 NOXM_MODUL703P/ operate/ START/ CLC2

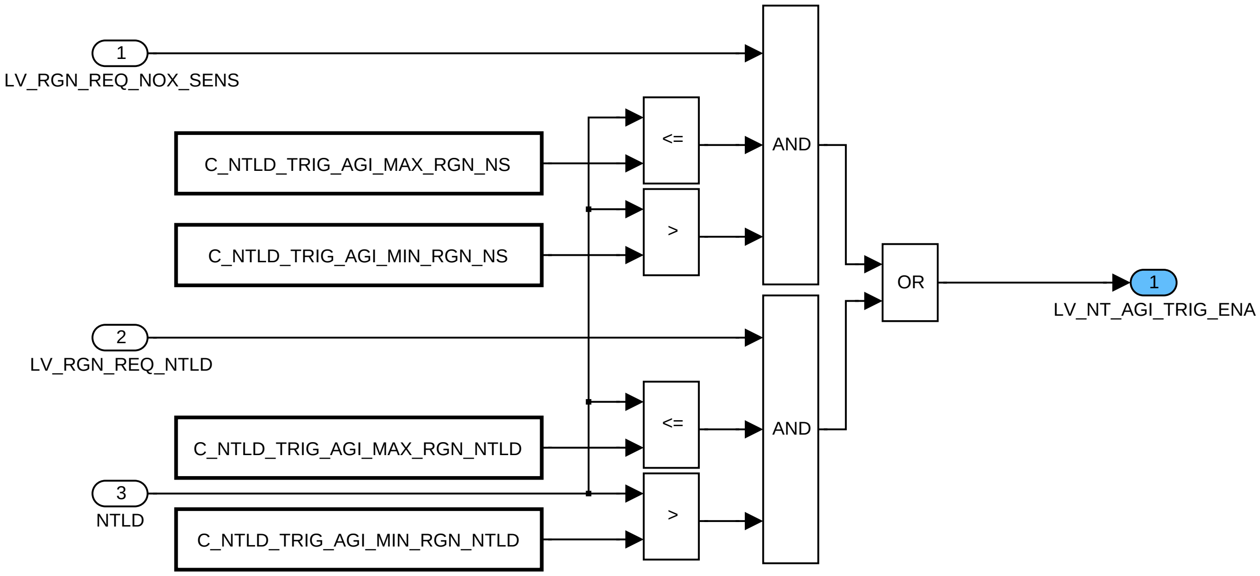


Figure 97 NOXM_MODUL703P/ operate/ START/ CLC3

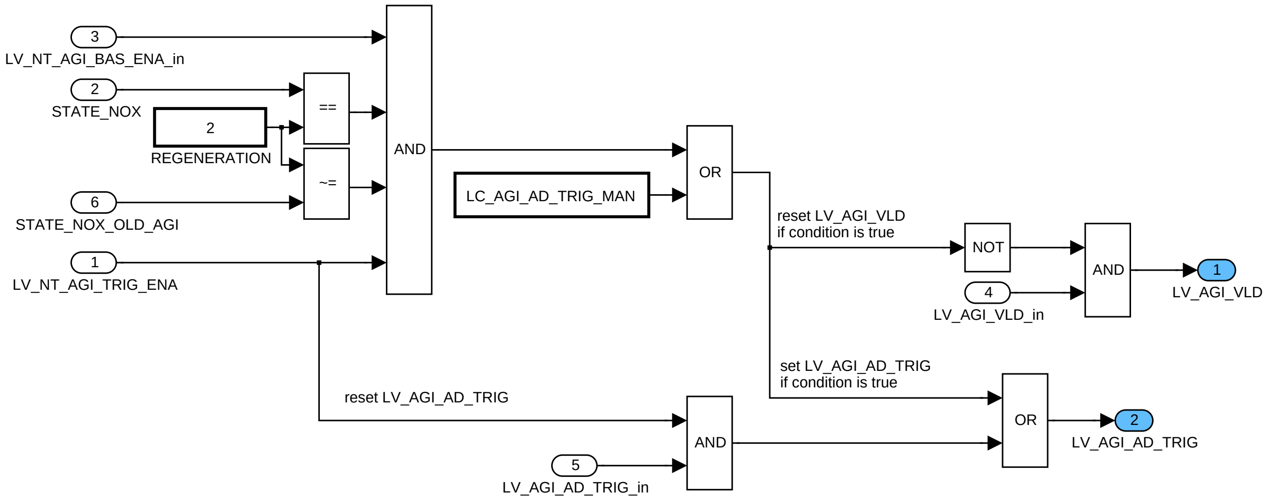


Figure 98 NOXM_MODUL703P/ operate/ START/ CLC4

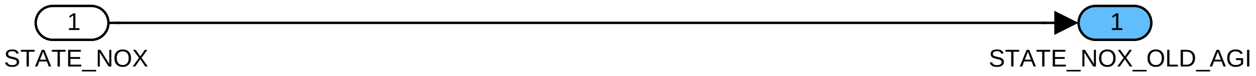


Figure 99 NOXM_MODUL703P/ operate/ START/ CLC5

MANUAL_INIT: Manual initialisation of sulphur and thermal aging factors

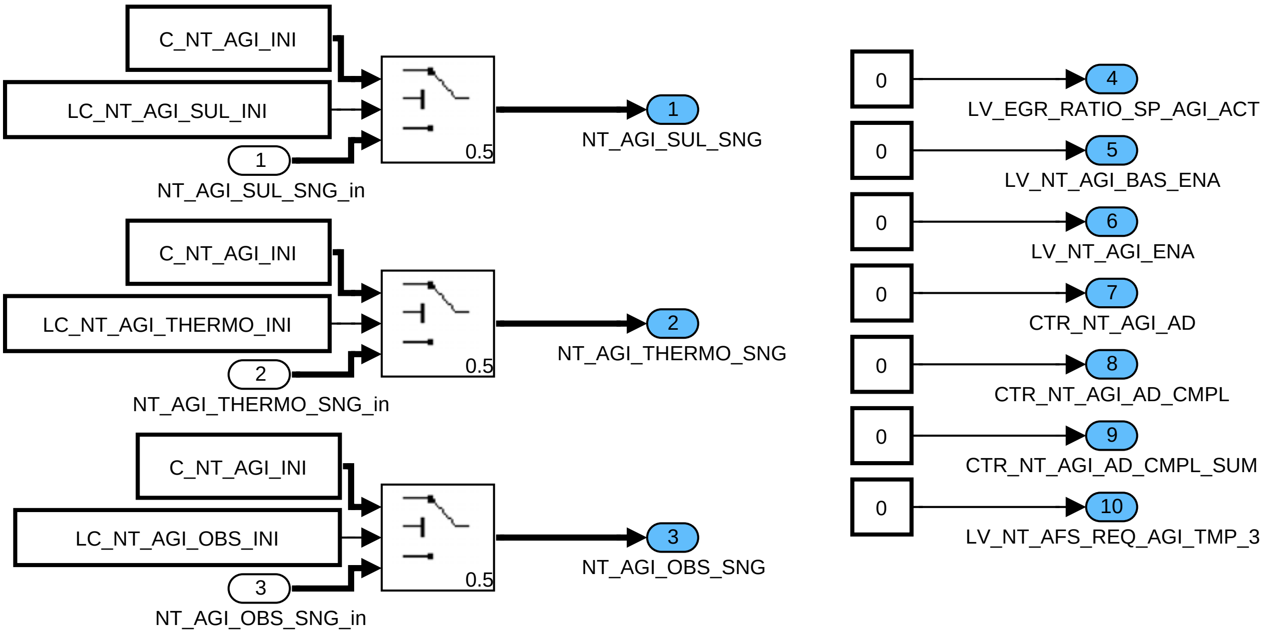


Figure 100 NOXM_MODUL703P/ operate/ MANUAL_INIT/ CLC

Check result of the NOx signal shift diagnosis an gain adaptation

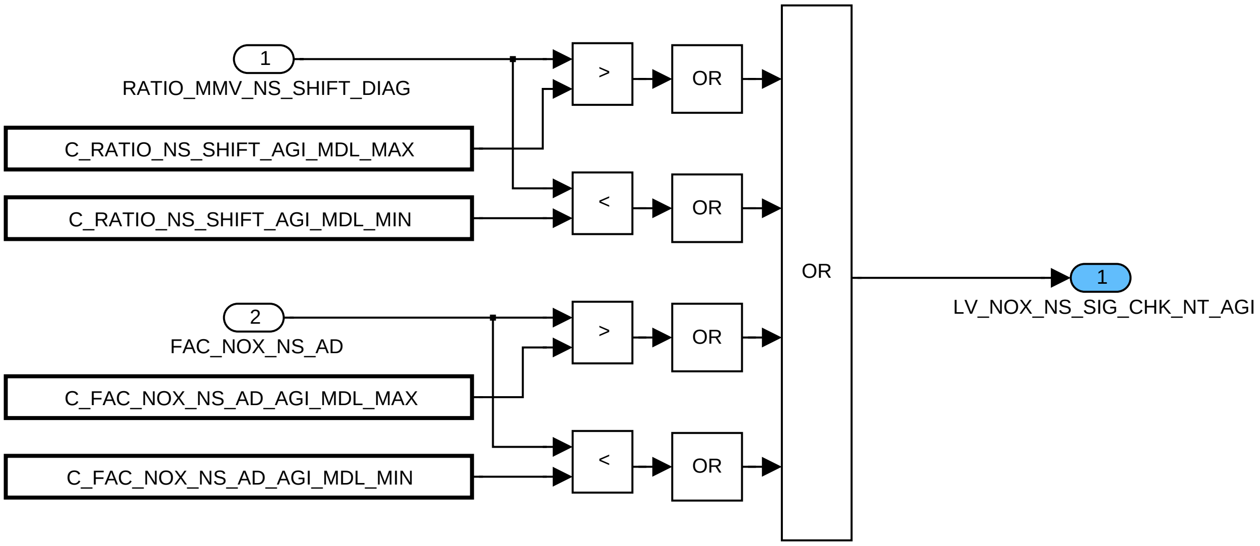


Figure 101 NOXM_MODUL703P/ operate/ MAIN/ CBK/ CHK_RATIO

Calculation of LV_SWI_MDL_THERMO_AGI_ACT

Calculation of Switch to modeled aging factor before sulphur aging model caluclation is active

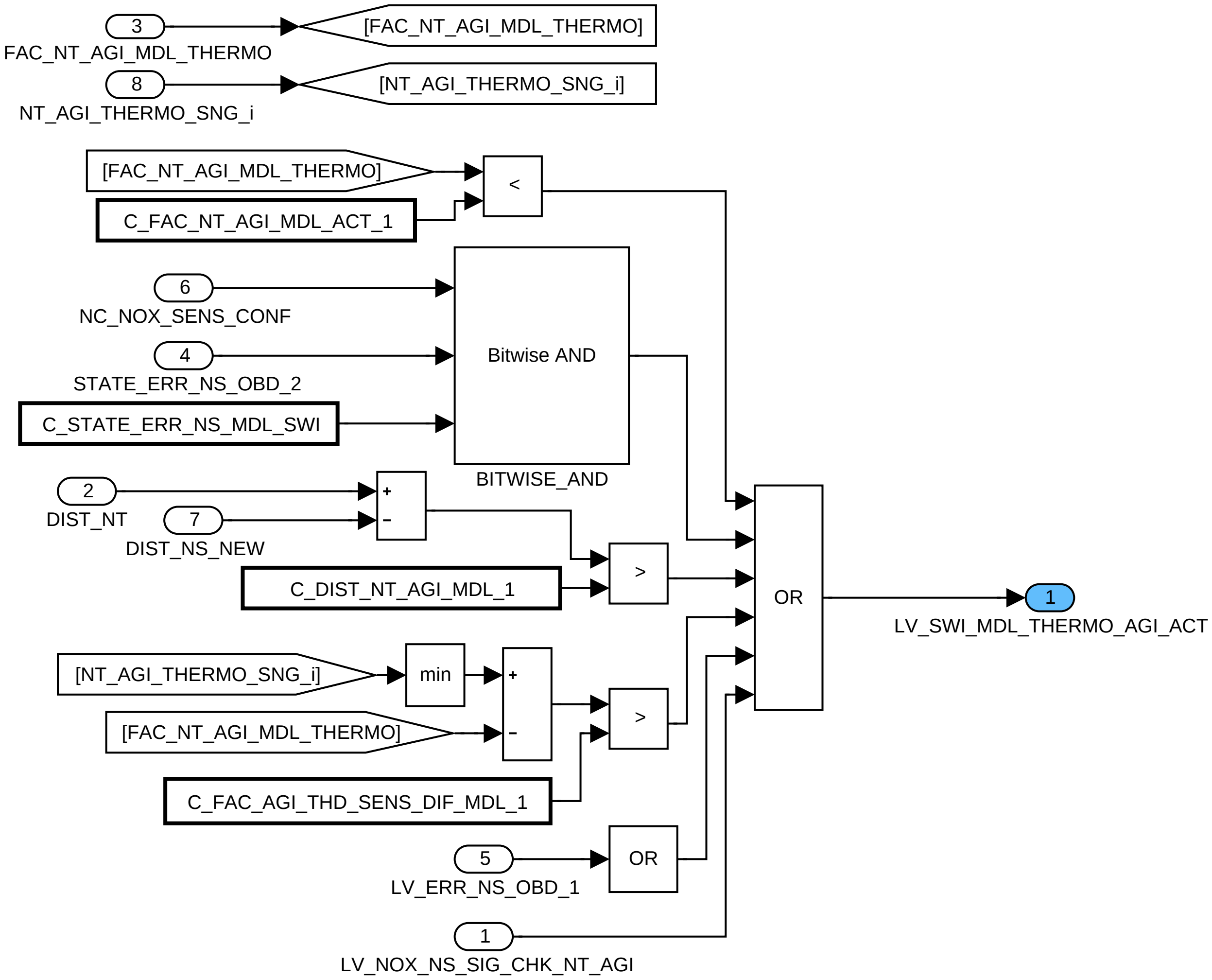


Figure 102 NOXM_MODUL703P/ operate/ MAIN/ CBK/ CLC_LV_SWI

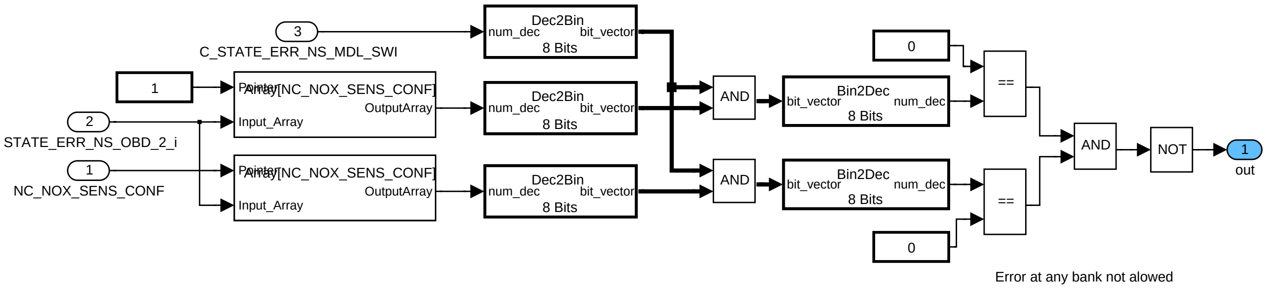


Figure 103 NOXM_MODUL703P/ operate/ MAIN/ CBK/ CLC_LV_SWI/ BITWISE_AND

MAIN (for each NOx trap)

The following calculation will be done for each NOx trap, controlled by NC_NT_NR.

Aging calculation will only be done if a rising edge of the trigger signal LV_AGI_AD_TRIG occurs and the NOx sensor reaches a threshold at start of regeneration.

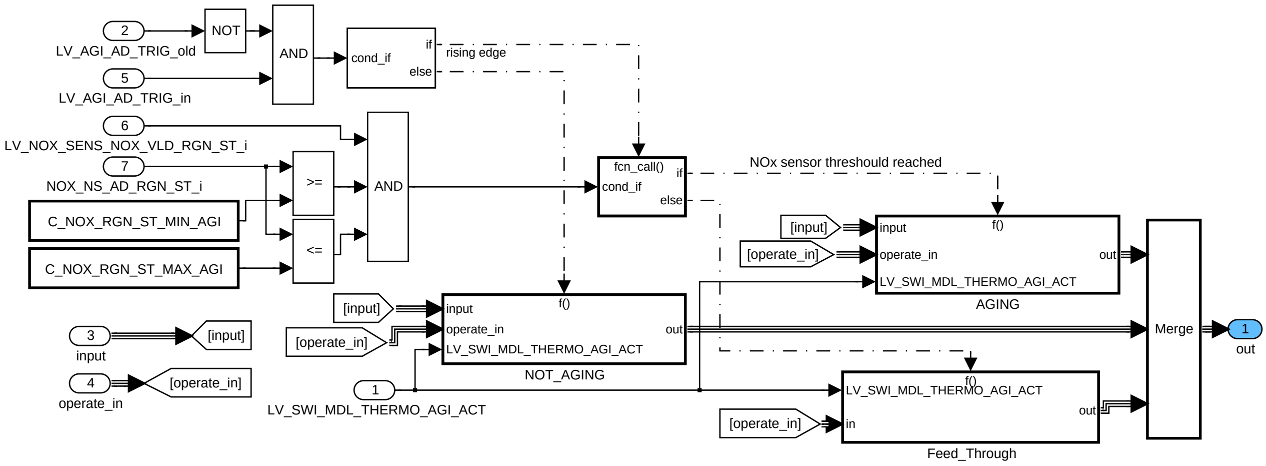


Figure 104 NOXM_MODUL703P/ operate/ MAIN/ CBK/ SELECT

Calculation of the new base aging factor

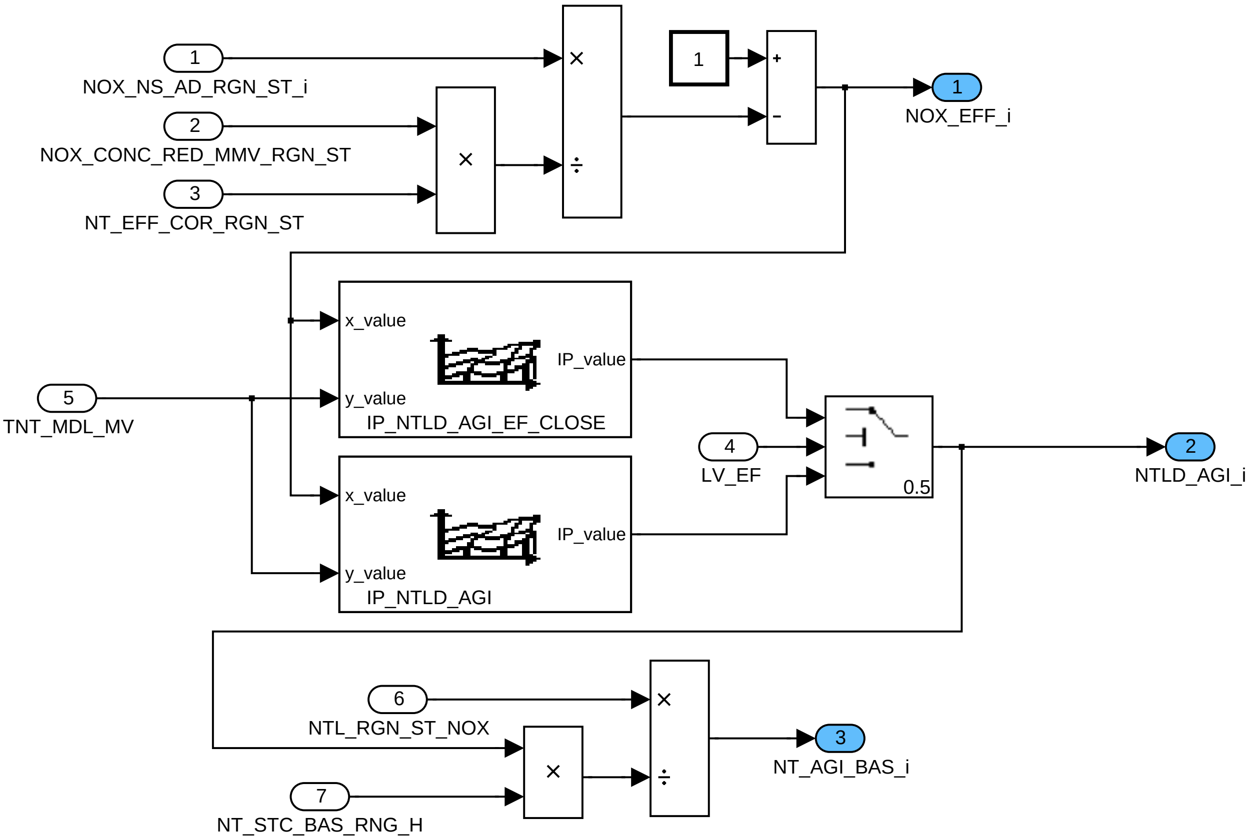


Figure 105 NOXM_MODUL703P/ operate/ MAIN/ CBK/ SELECT/ AGING/ CLC1

Calculation of quick aging monitoring factors by filtering of the base aging factor

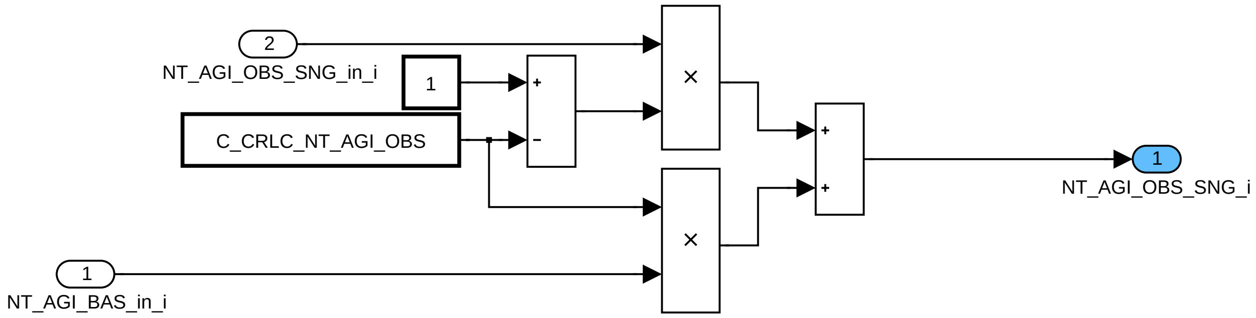


Figure 106 NOXM_MODUL703P/ operate/ MAIN/ CBK/ SELECT/ AGING/ CLC2

Depending on sulfur load, either the aging factor due to thermal aging (this picture) or the aging factor due to sulphur load will be calculated (next picture). Both results will be get by filtering of the base aging factor.

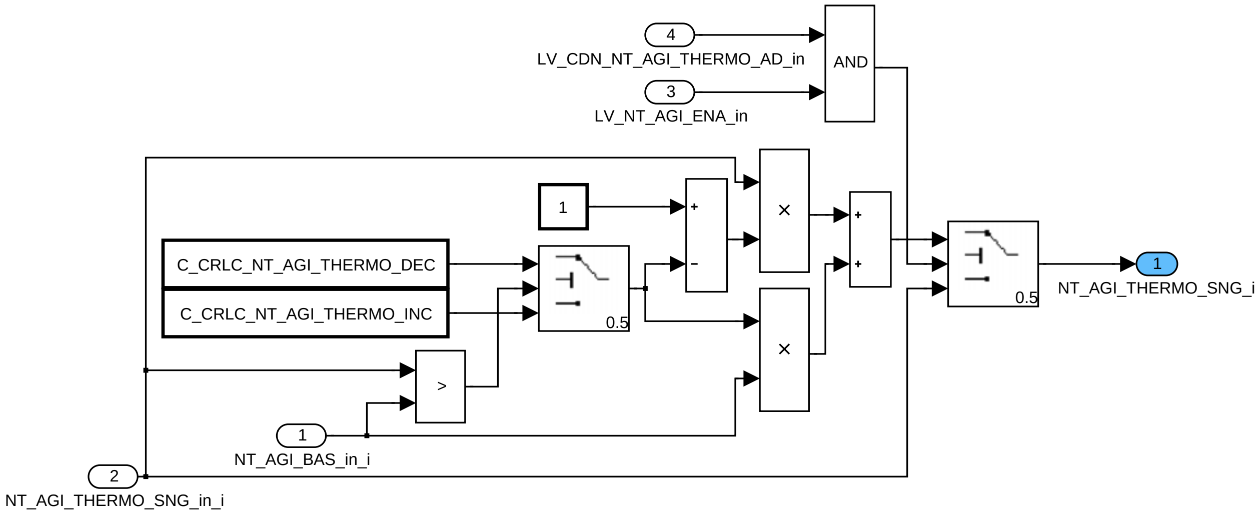


Figure 107 NOXM_MODUL703P/ operate/ MAIN/ CBK/ SELECT/ AGING/ CLC3

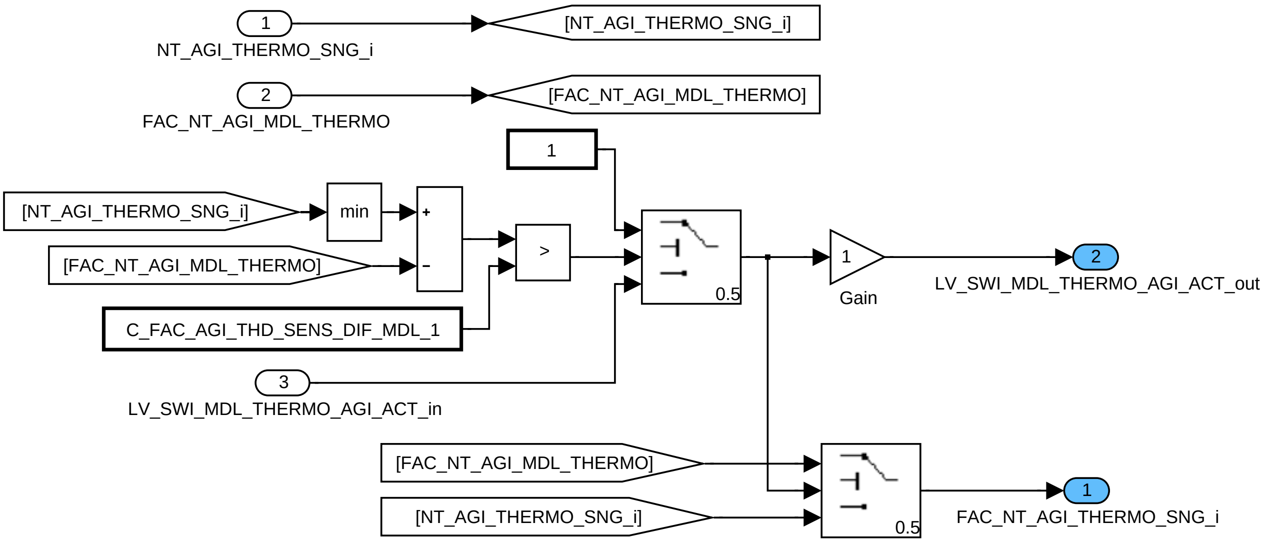


Figure 108 NOXM_MODUL703P/ operate/ MAIN/ CBK/ SELECT/ AGING/ CLC4

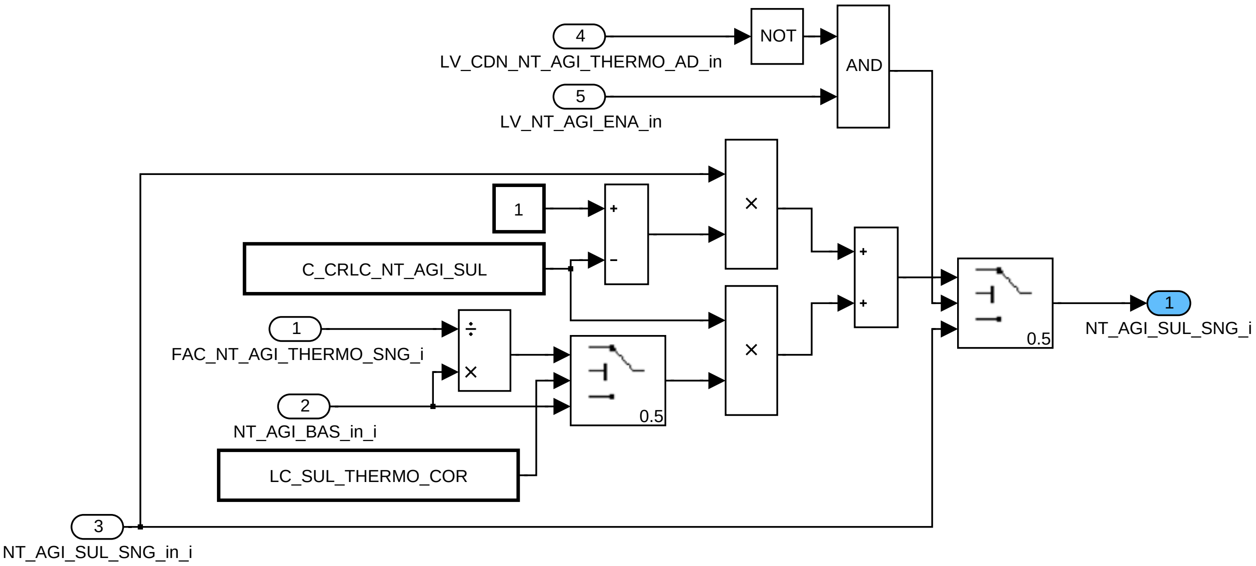


Figure 109 NOXM_MODUL703P/ operate/ MAIN/ CBK/ SELECT/ AGING/ CLC5

Update of sulphur aging factor during desulfation (during desulfation, the sulphur load is decreased and the sulphur load aging factor is incremented).

Initialisation of quick aging monitoring factors

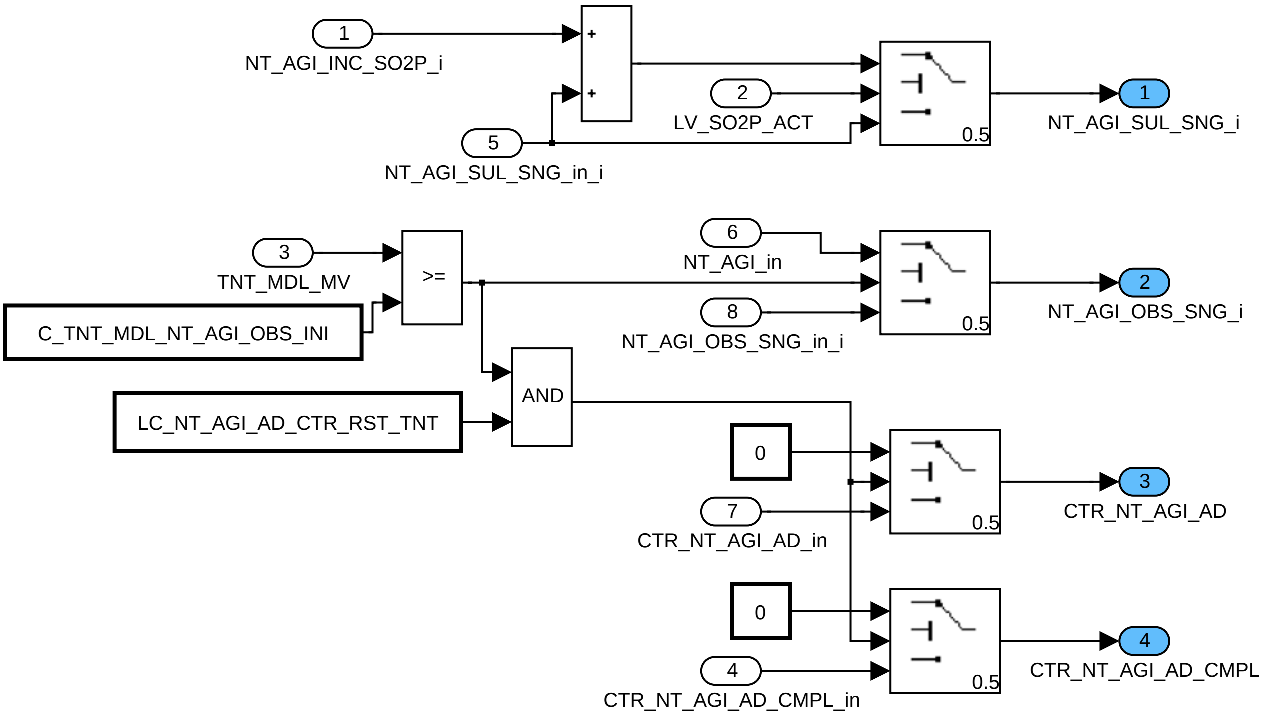


Figure 110 NOXM_MODUL703P/ operate/ MAIN/ CBK/ SELECT/ NOT_AGING/ CLC

MAIN (after calculations for each NOx trap)

The incrementation of the counters and the integral for the fuel quality calculation have to be calculated once if the condition is true for one bank or both banks.

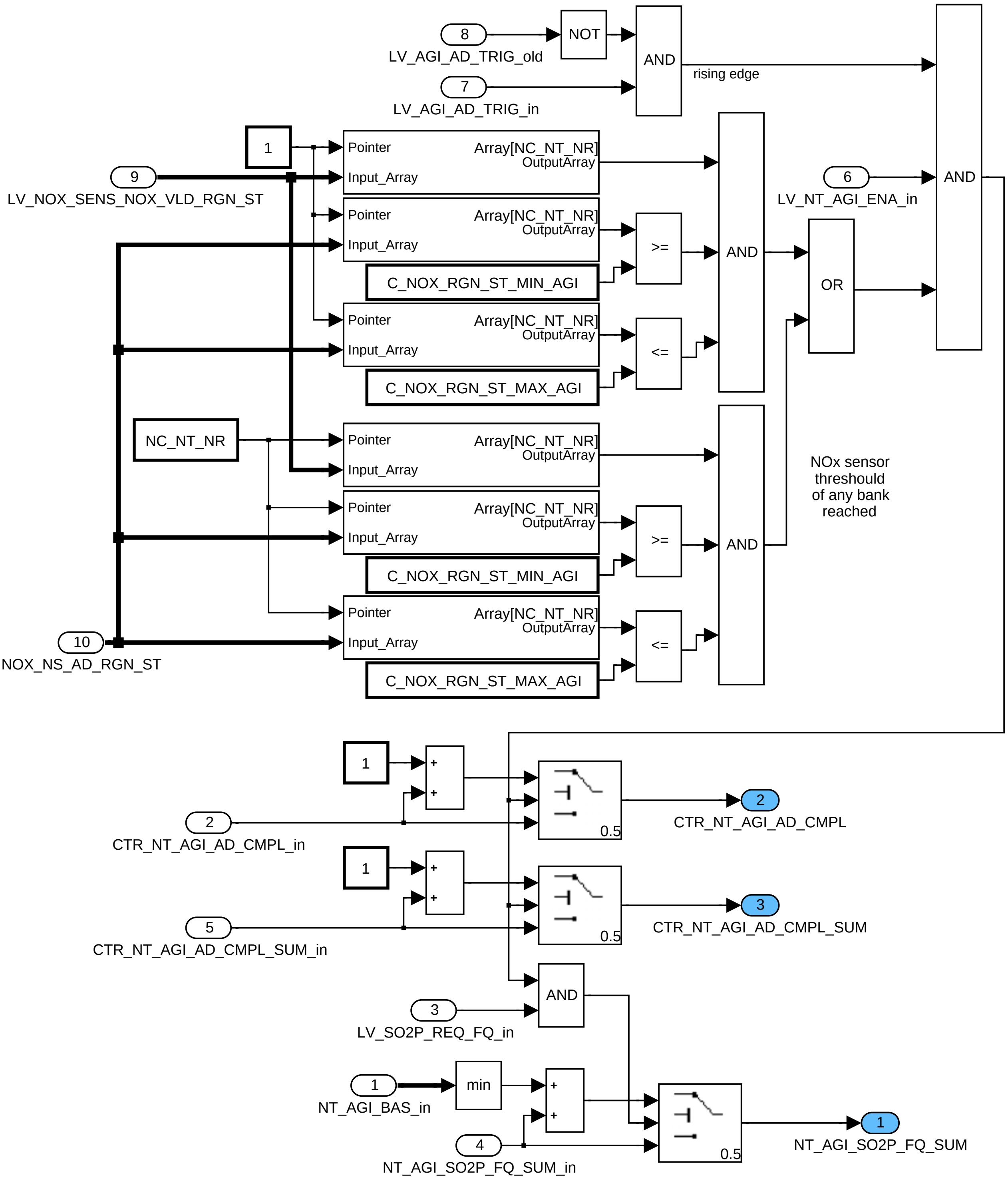


Figure 111 NOXM_MODUL703P/ operate/ MAIN/ CTR

ENABLING_FLAGS

The calculation of the aging enabling flags is done after the aging calculation, thereby it's possible to reset LV_NT_AGI_ENA in regeneration, the aging calculation is done at the first calculation of the main loop in regeneration.

Conditions to enable base aging fctor calculation/quick monitoring

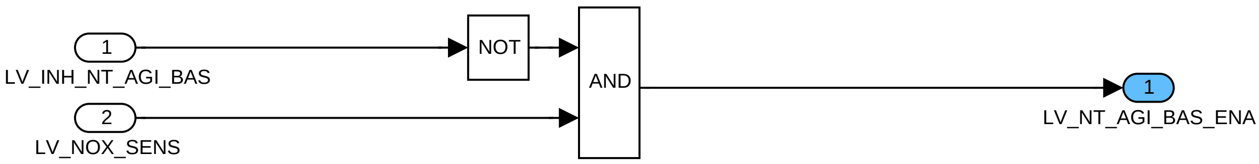


Figure 112 NOXM_MODUL703P/ operate/ ENABLING_FLAGS/ CLC1

Condition to enable aging factor calculation

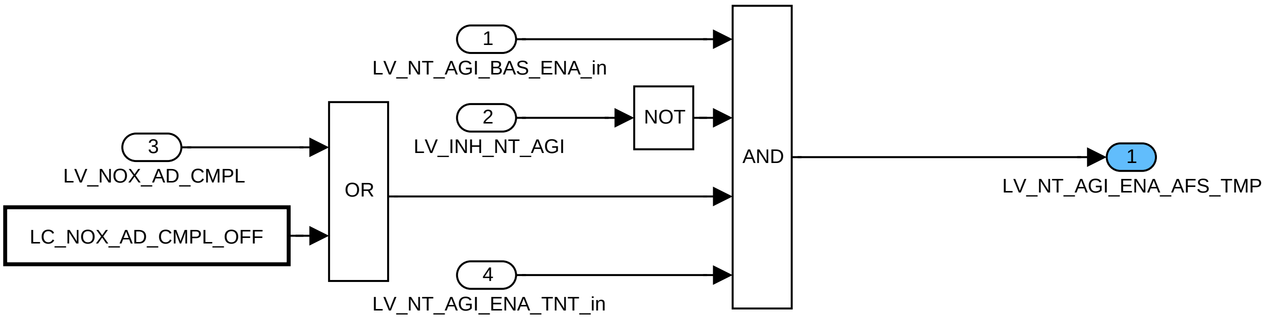


Figure 113 NOXM_MODUL703P/ operate/ ENABLING_FLAGS/ CLC2

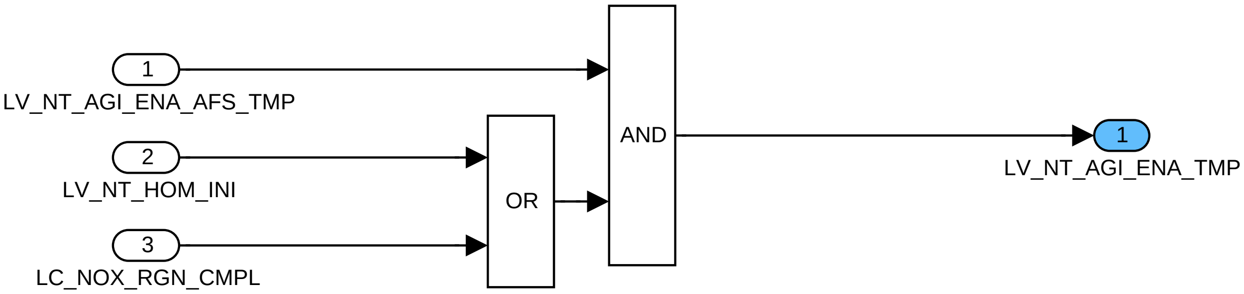


Figure 114 NOXM_MODUL703P/ operate/ ENABLING_FLAGS/ CLC3

The T_AFL condition uses only lean phases whith LV_NT_AGI_ENA given at the beginning of the lean phase; aim is to prevent a closing of the EGR valve in the “middle” of a lean phase due to aging adaptation conditions given starting there as well as performing an adaptation after a lean phase with adaptation condition given only temporary.

CTR_NT_AGI_AD and CTR_NT_AGI_AD_CMPL have physical limits from 0 to 254 whereas the corresponing threshold constants C_CTR_NT_AGI_AD_MAX & C_CTR_NT_AGI_AD_CMPL_MAX have physical limits from 0 to 255. Thus by setting the constants to their maximum value the condition CTR < thd is always true and no limitation of the adaptation attemps / adaptation cycles will occur

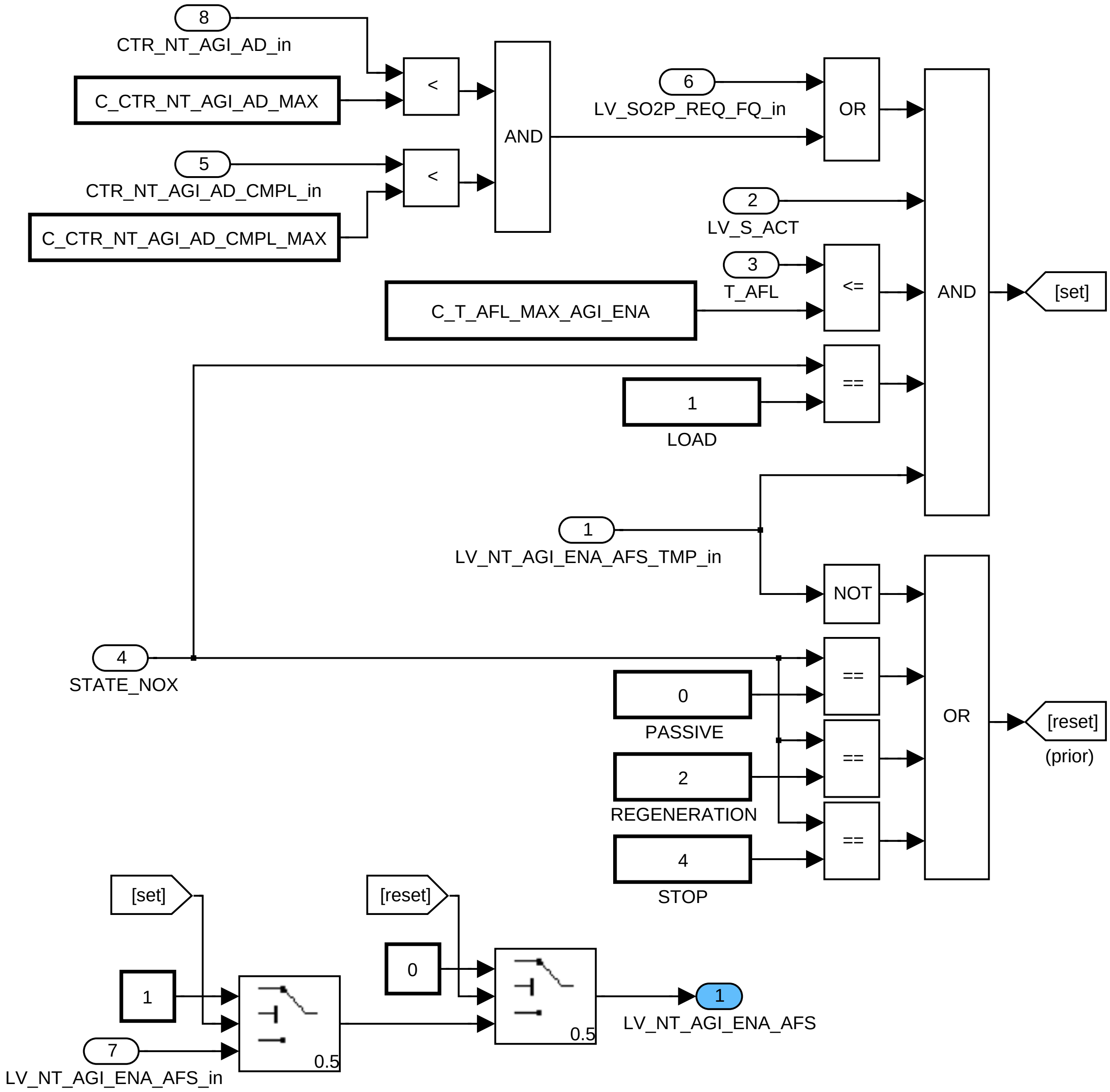


Figure 115 NOXM_MODUL703P/ operate/ ENABLING_FLAGS/ CLC4

Conditions to close EGR valve during aging adaptation

The attemps counter is only calculated if EGR was closed; if the function is calibrated “passive” (i.e. without closing EGR) the number of attempts is not limited.

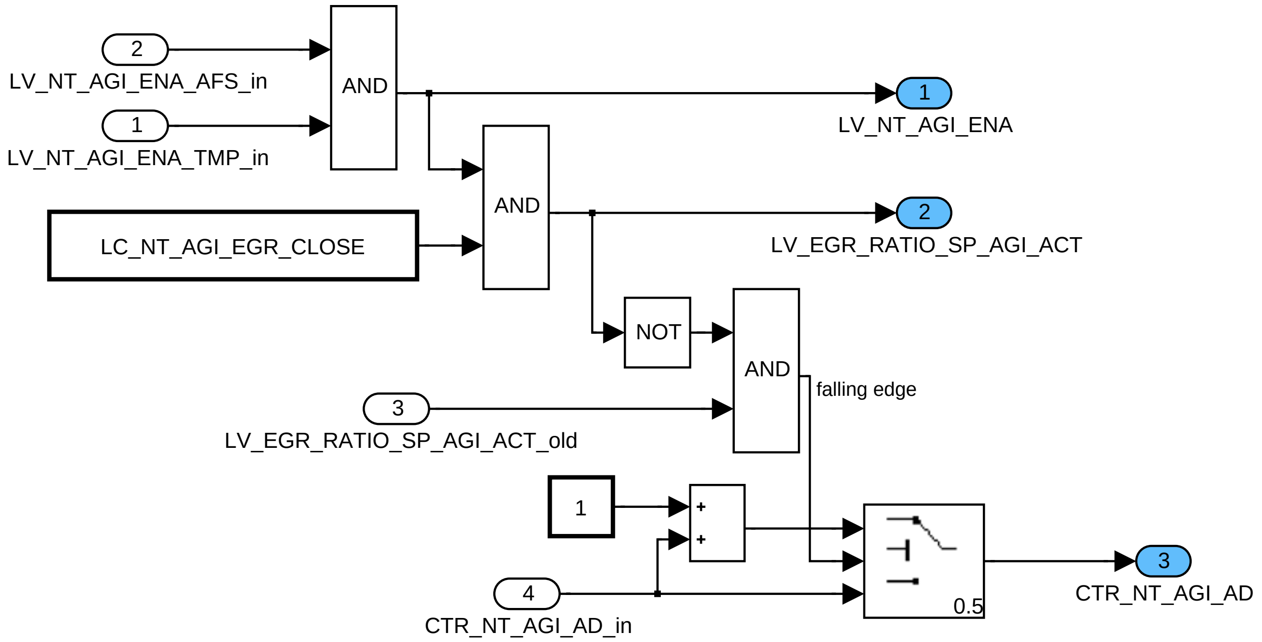


Figure 116 NOXM_MODUL703P/ operate/ ENABLING_FLAGS/ CLC5

AGING_FACTOR: Aging factor handling

NT_SUL_AGI_SNG will be calculated from the aging factor due to sulphur load for each bank, defined by NC_NT_NR.

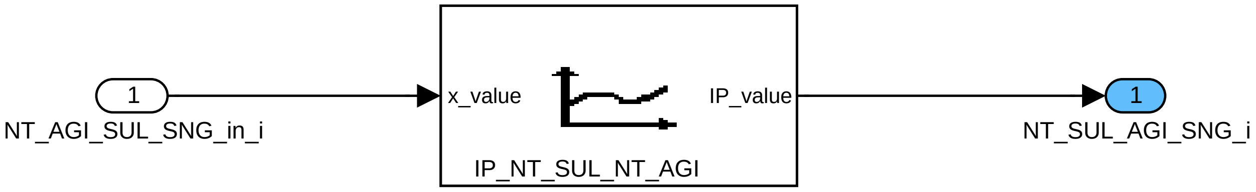


Figure 117 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CBK/ CLC1

Bank selective values will be combined: minimum of aging factors and maximum of sulphur load.

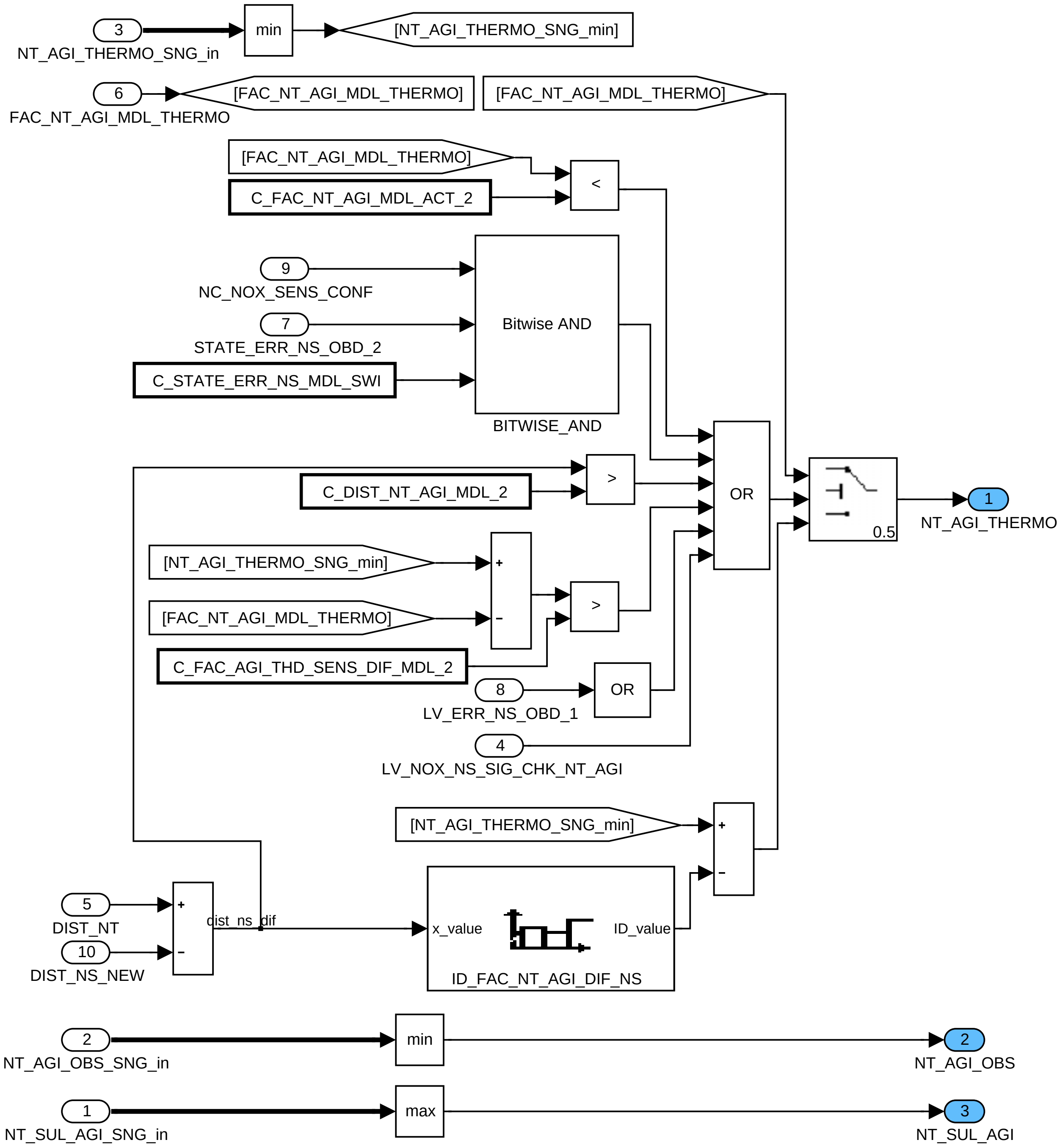


Figure 118 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CLC2

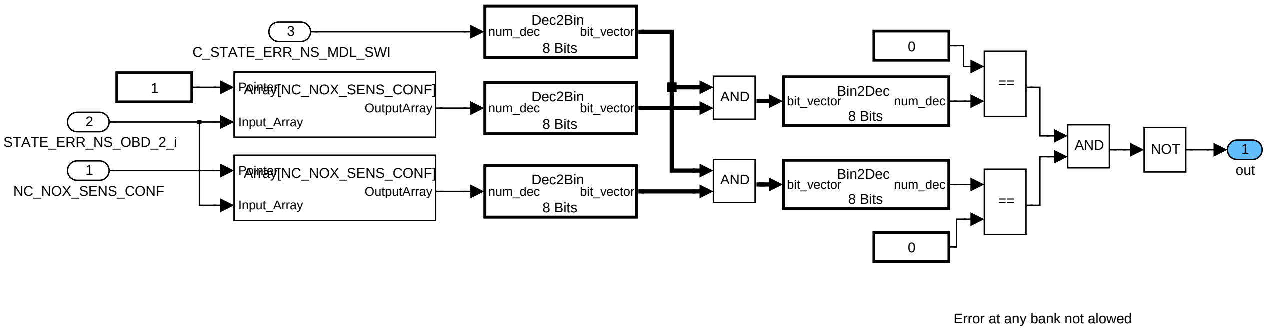


Figure 119 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CLC2/ BITWISE_AND

The NOx trap aging factor may be updated from updated from sulphur model (this picture) or vice versa (next picture)

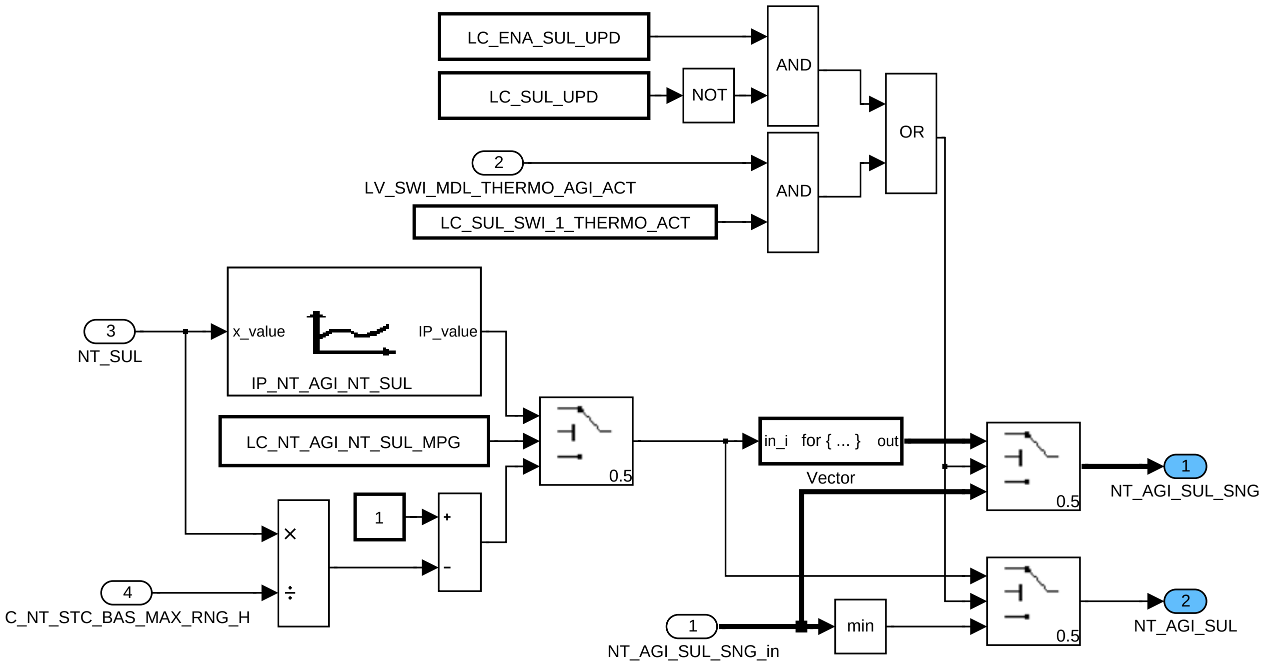


Figure 120 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CLC3

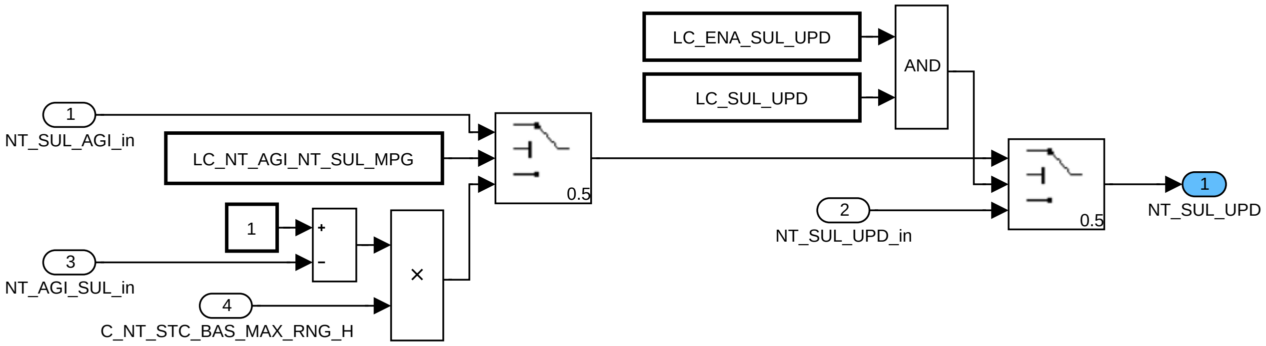


Figure 121 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CLC4

The following pictures show that NOx trap aging factors may be selected from different sources.

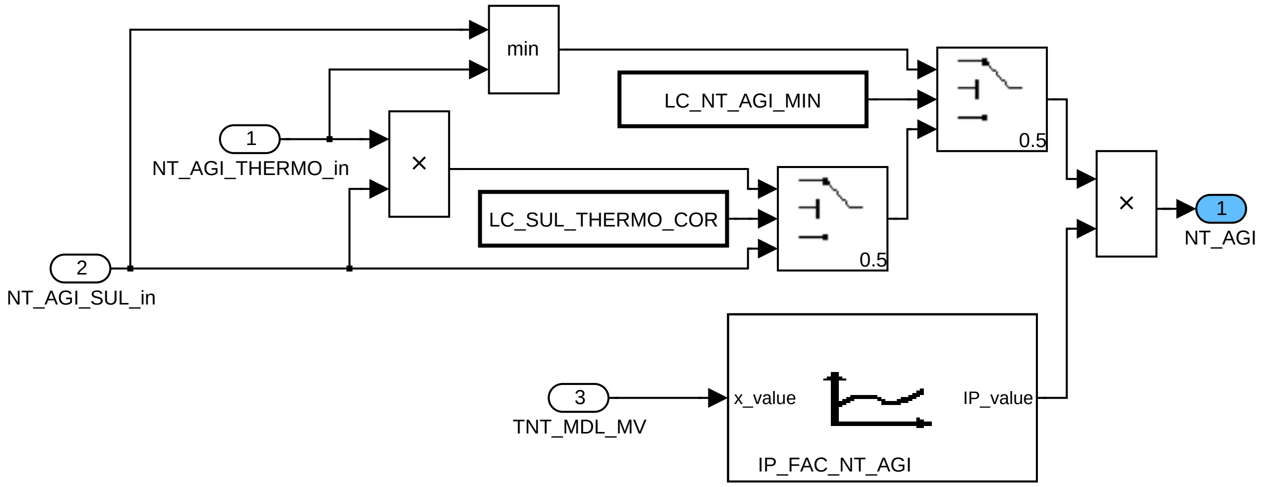


Figure 122 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CLC5

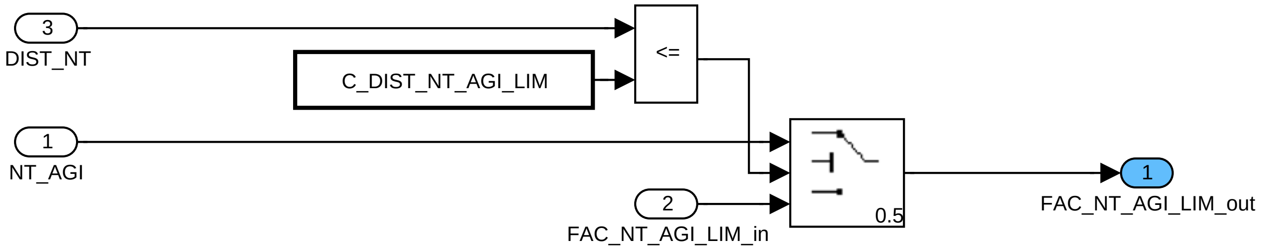


Figure 123 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CLC6

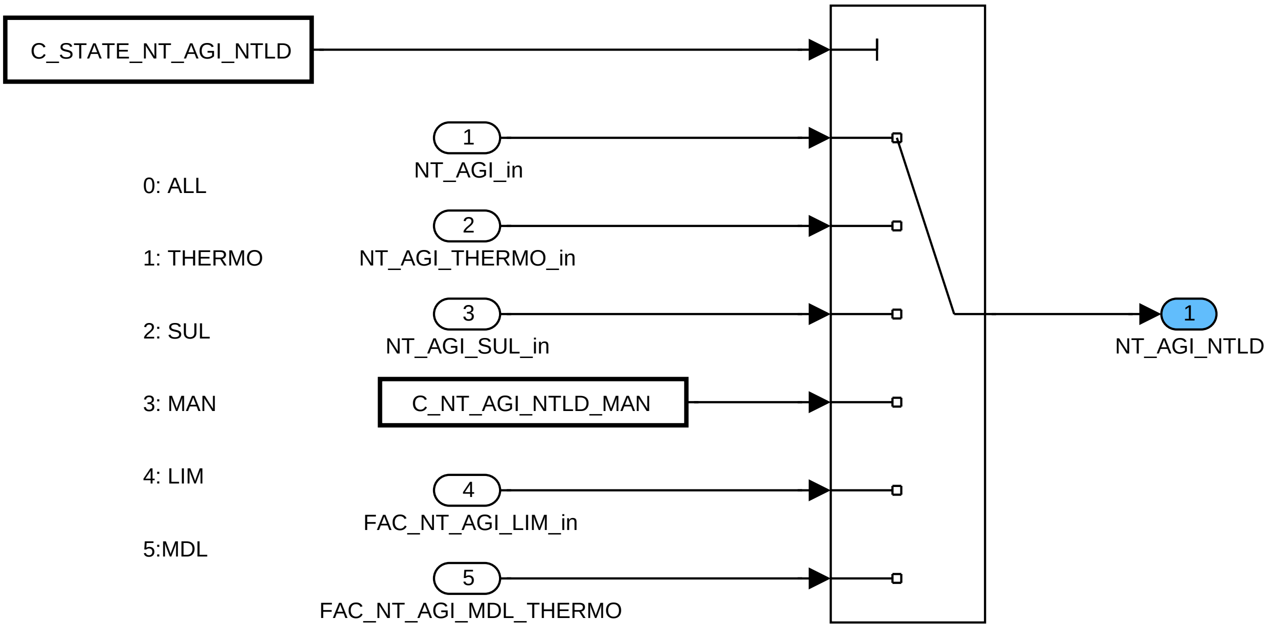


Figure 124 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CLC7

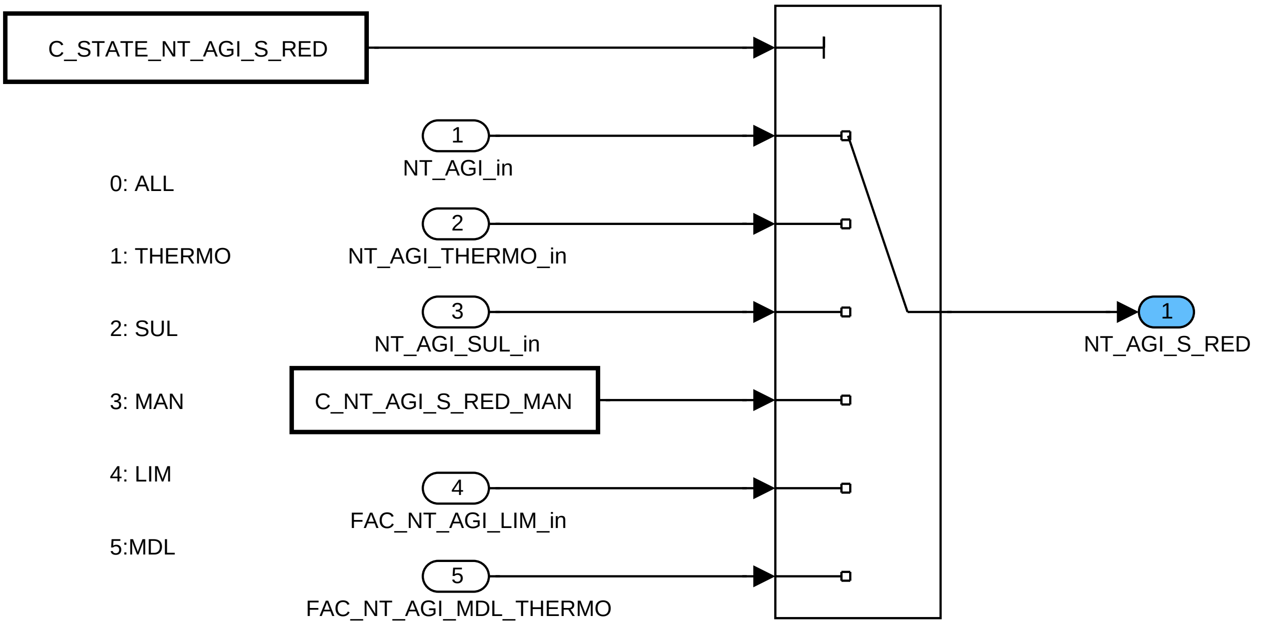


Figure 125 NOXM_MODUL703P/ operate/ AGING_FACTOR/ CLC8

MANUAL_RESET: Manual reset of desulfation requests

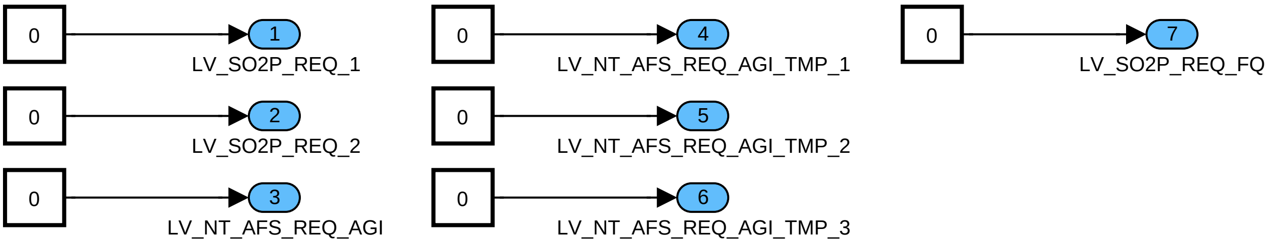


Figure 126 NOXM_MODUL703P/ operate/ MANUAL_RESET/ CLC

DESULFATION

If aging factor or sulphur load reaches the first threshold, then the first desulfation request will be set.

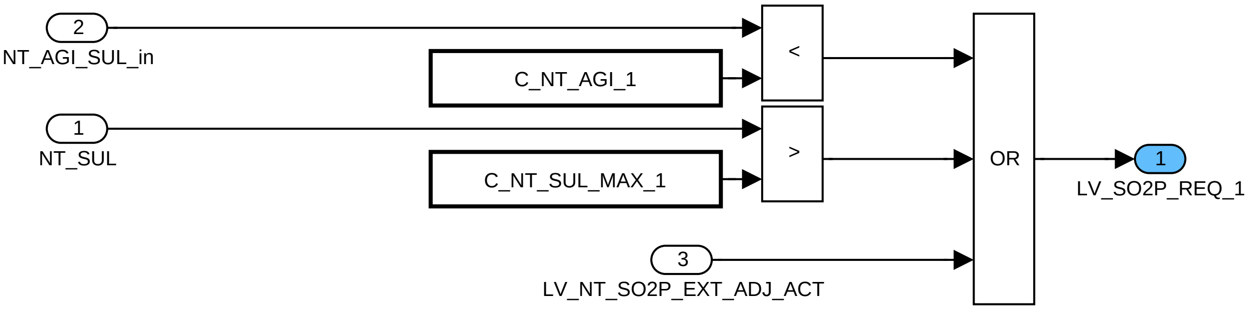


Figure 127 NOXM_MODUL703P/ operate/ DESULFATION/ CLC1

If the values become worse and reach the second threshold, then the second desulfation request will be set. In standard cases with no need of separate reset conditions for

LV_SO2P_REQ_2, IP_NT_SUL_MAX_2_DEAC and IP_NT_SUL_MAX_2 have to be calibrated identically

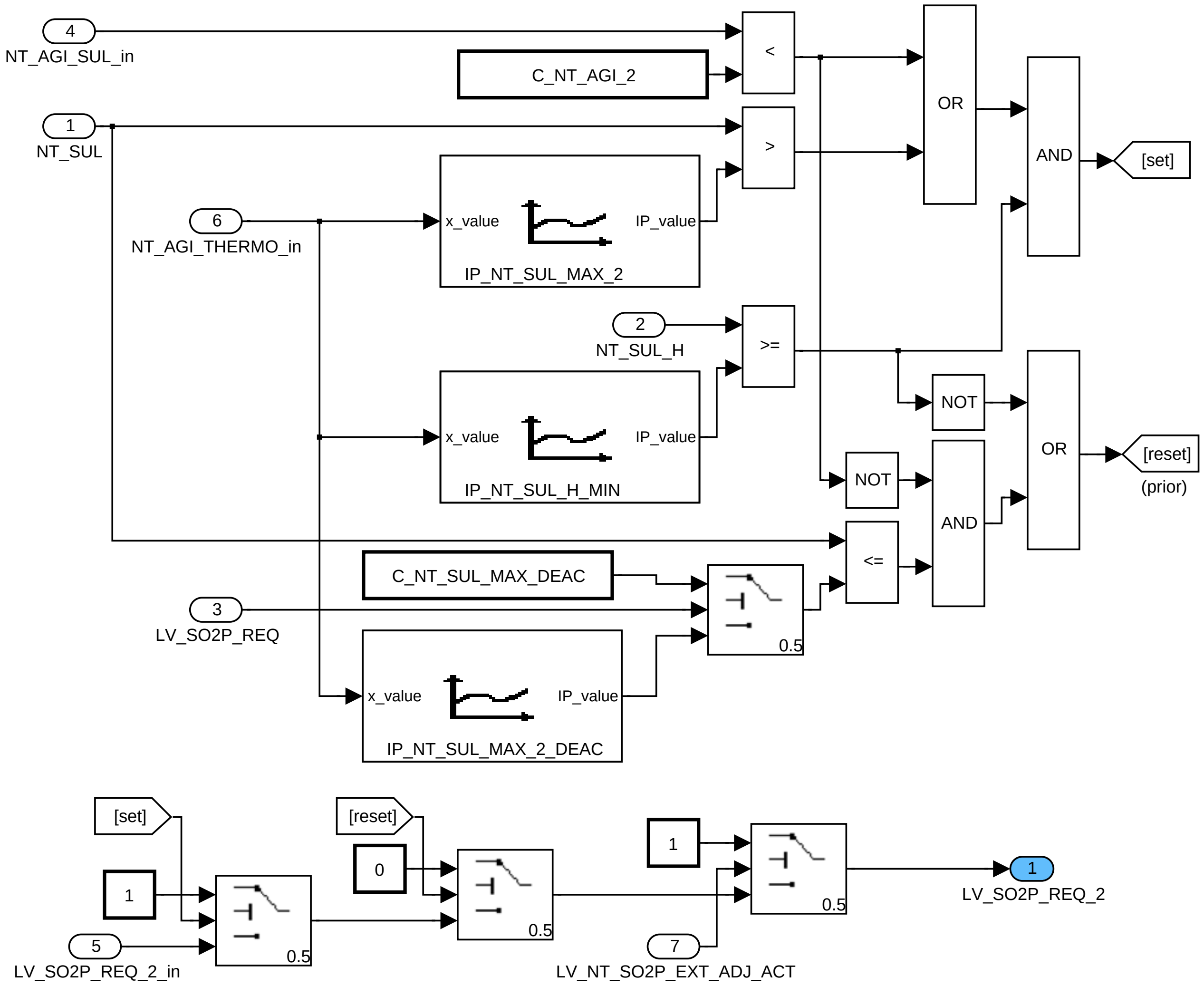


Figure 128 NOXM_MODUL703P/ operate/ DESULFATION/ CLC2

Condition for lambda=1 request

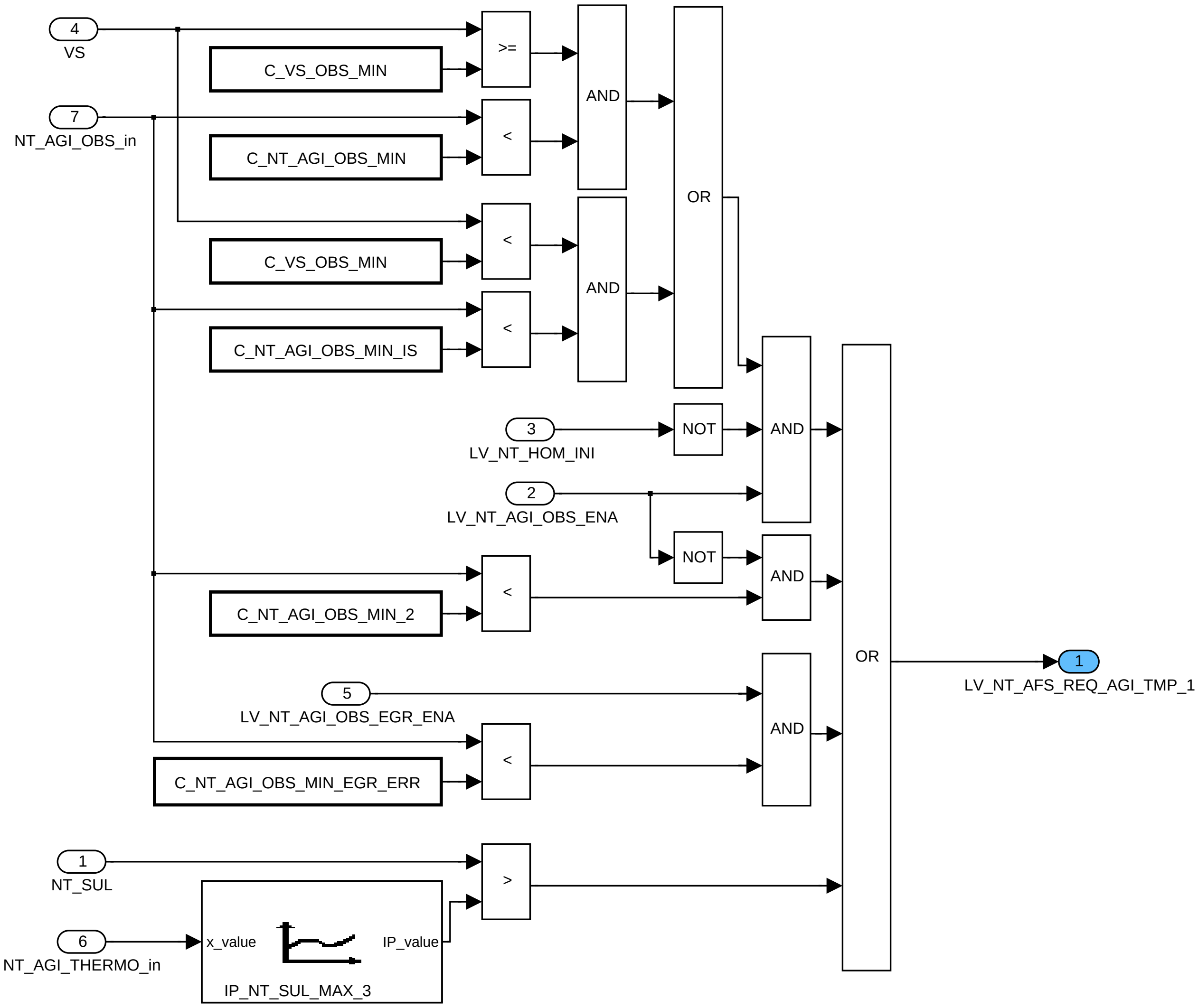


Figure 129 NOXM_MODUL703P/ operate/ DESULFATION/ CLC3

Structure of fuel quality check

If the thermal aging factor reaches the third threshold, then the fuel quality check will be started. In that case a desulfation will be requested and the sulfur model will be calculated with higher sulfur ratio, controlled by FAC_SU_CONC_FUEL.

After desulfation (LV_SO2P_REQ_1=0) the prohibition of stratified mode (LV_NT_AFS_REQ_AGI_TMP_2) will be canceled.

Either the aging factor is OK and the fuel check will be finished immediately (sub-system OK) or after a calibratable number of cycles a decicion will be made (sub-system END) whether the NOx trap is OK or not. If OK the fuel quality check will be finished, if not the lambda=1 mode will be forced (LV_NT_AFS_REQ_AGI_TMP_3=1) because the trap is defect.

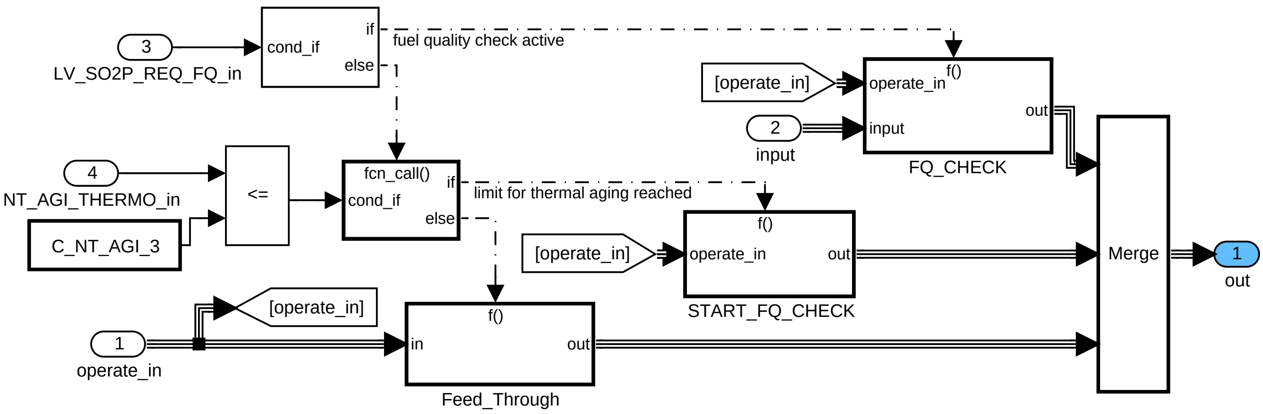


Figure 130 NOXM_MODUL703P/ operate/ DESULFATION/ CLC4

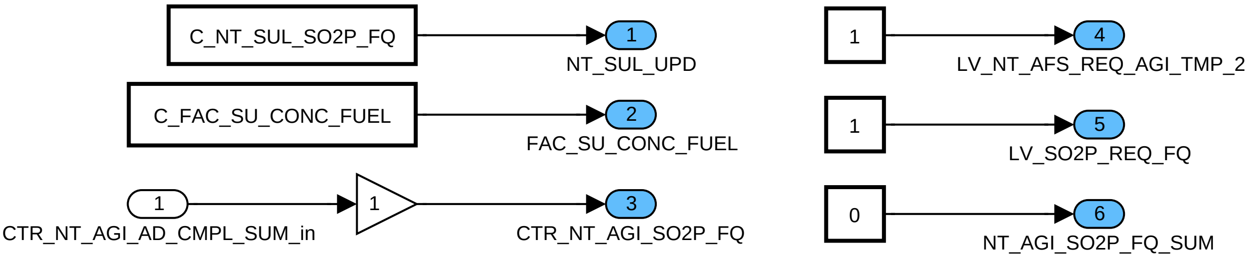


Figure 131 NOXM_MODUL703P/ operate/ DESULFATION/ CLC4/ START_FQ_CHECK/ CLC

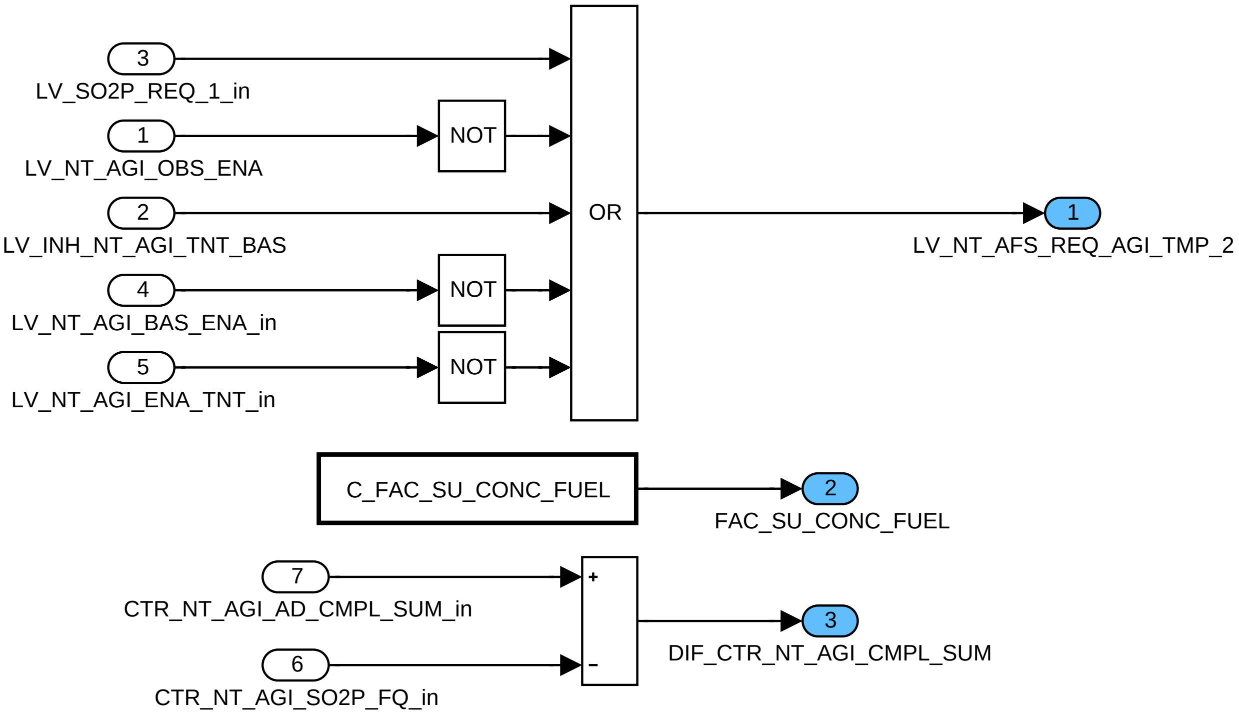


Figure 132 NOXM_MODUL703P/ operate/ DESULFATION/ CLC4/ FQ_CHECK/ CLC1

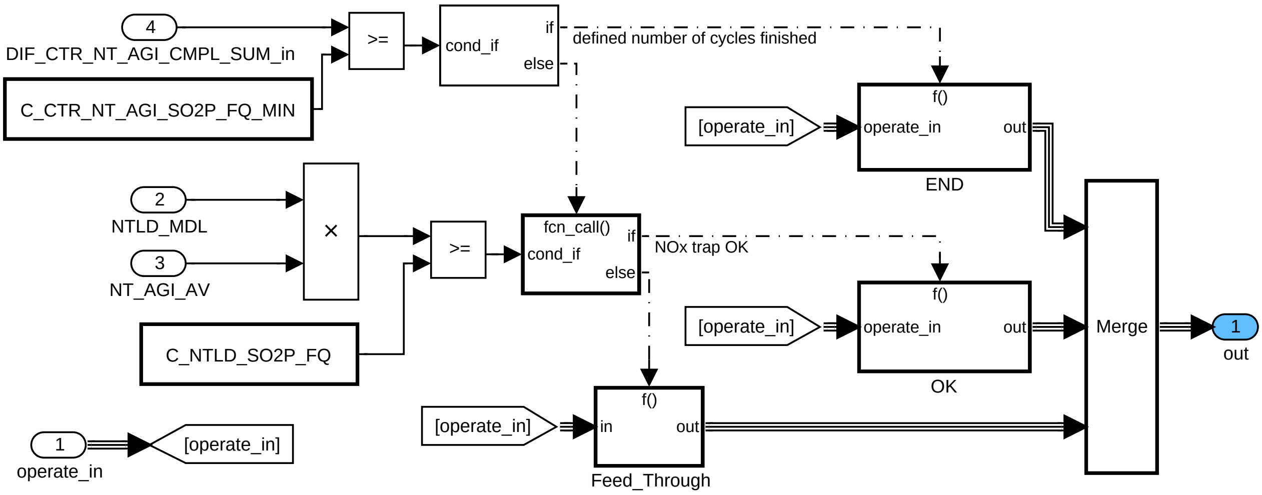


Figure 133 NOXM_MODUL703P/ operate/ DESULFATION/ CLC4/ FQ_CHECK/ CLC2

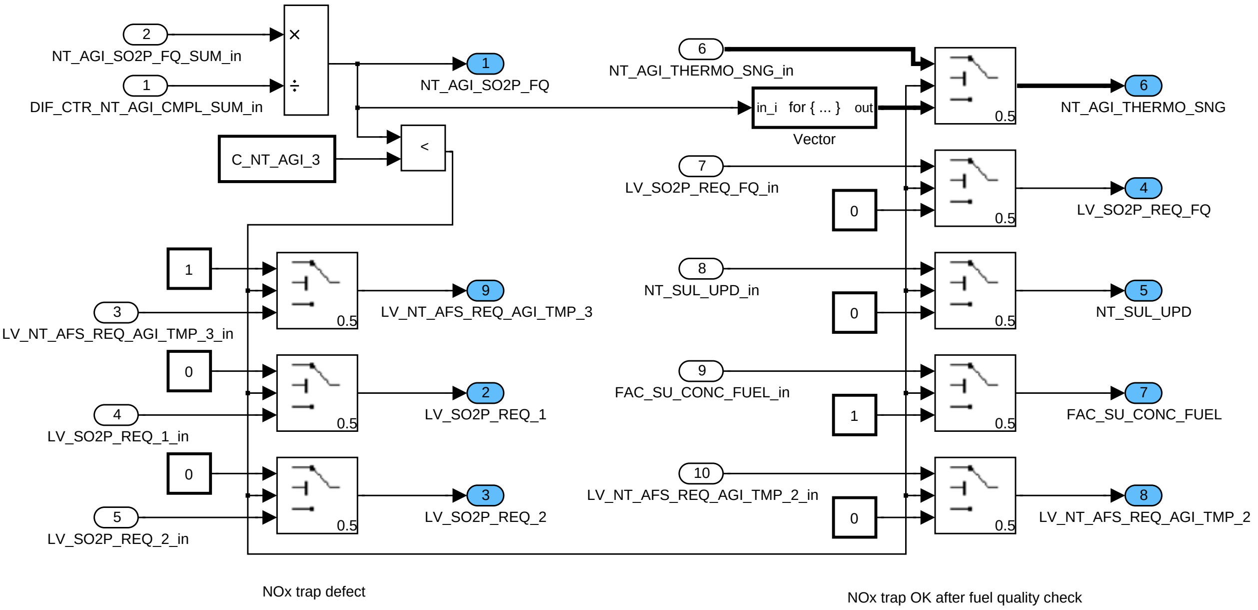


Figure 134 NOXM_MODUL703P/ operate/ DESULFATION/ CLC4/ FQ_CHECK/ CLC2/ END/ CLC

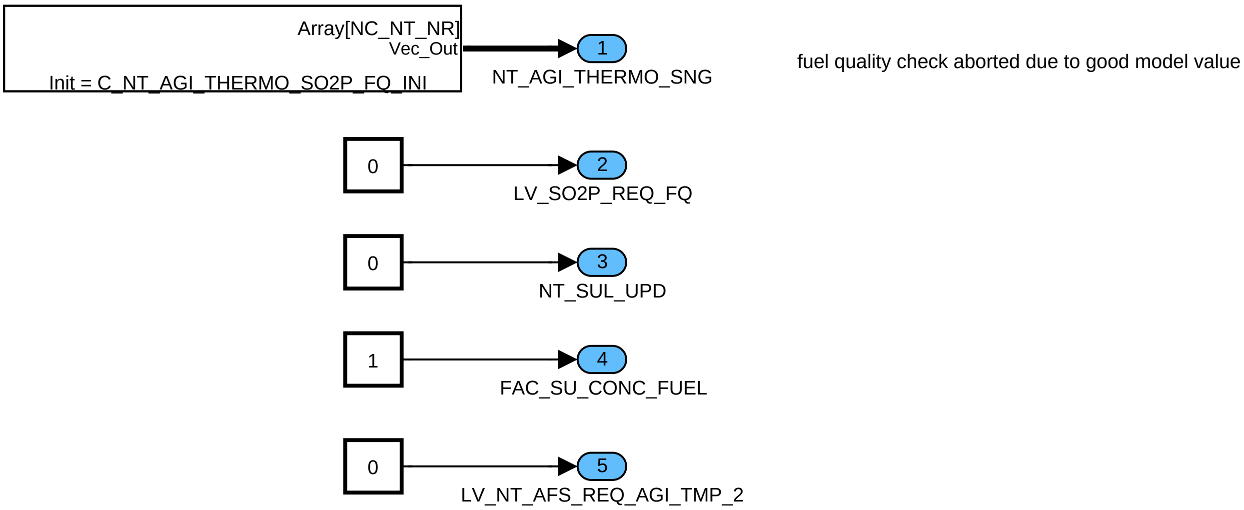


Figure 135 NOXM_MODUL703P/ operate/ DESULFATION/ CLC4/ FQ_CHECK/ CLC2/ OK/ CLC

Collection of all lambda=1 requests.

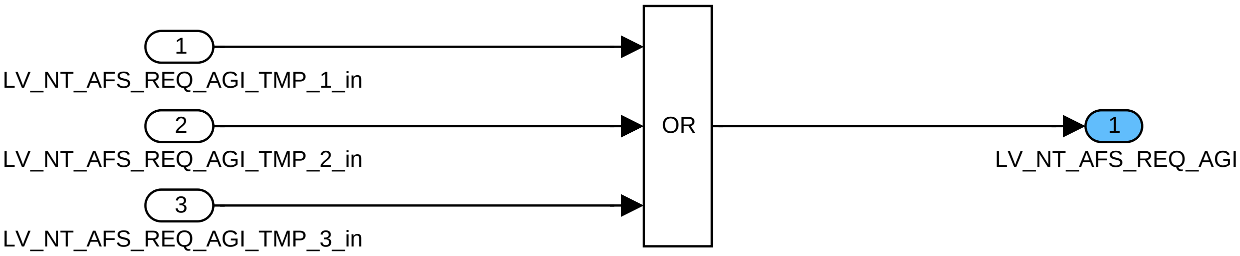


Figure 136 NOXM_MODUL703P/ operate/ DESULFATION/ CLC5

NEW_DEVICE: Initialisation by scan tool

If a new NOx trap is built in, then a initialization via a scan tool have to be made.

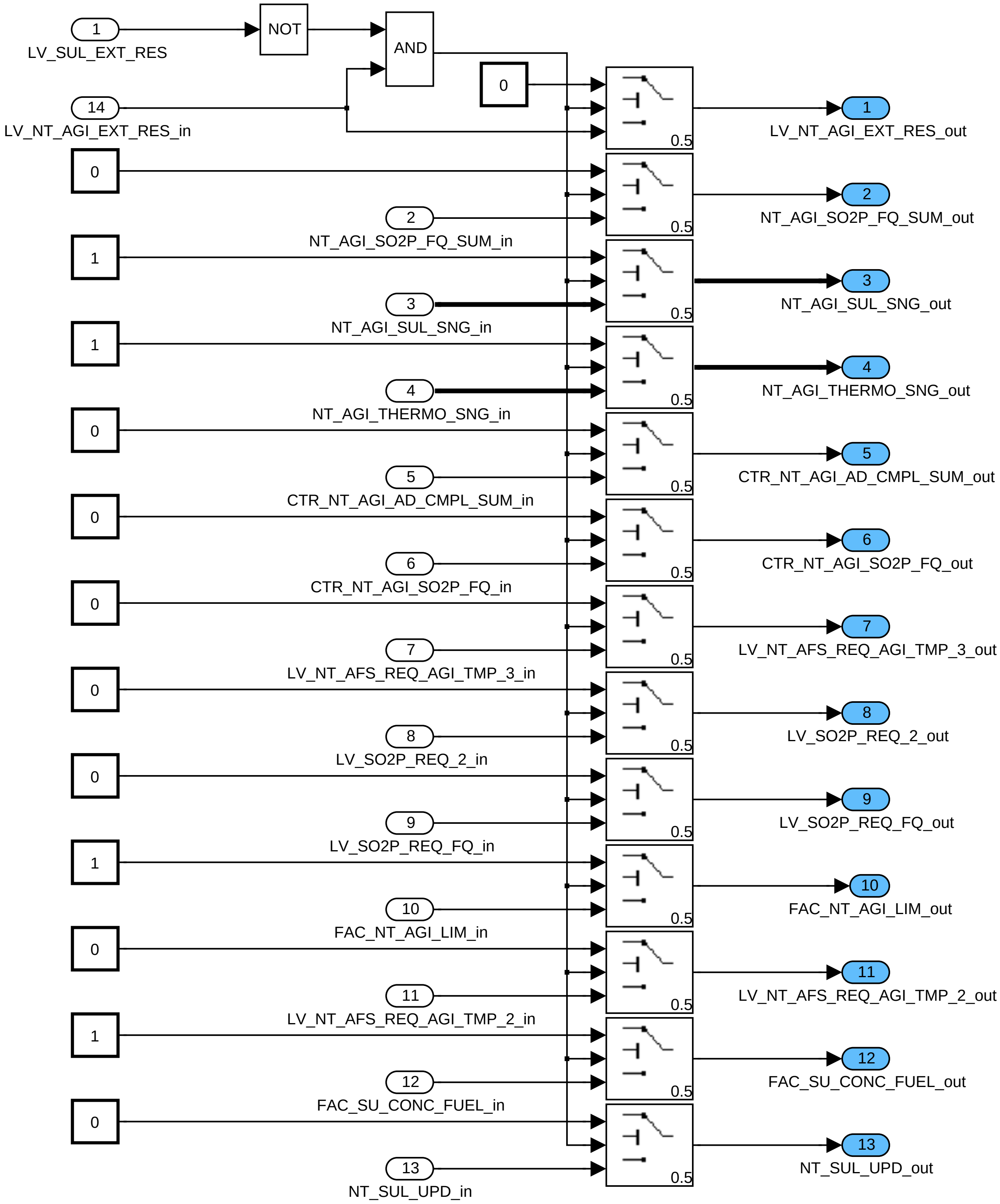


Figure 137 NOXM_MODUL703P/ operate/ NEW_DEVICE/ CLC1

Update of external adjustment values

If the flag LV_NT_AGI_EXT_ADJ was set by action ACTION_NOXM_WriteAgingExtAdj and the external sulfur model values are already written, then the external values will be used and the flag will be reset.

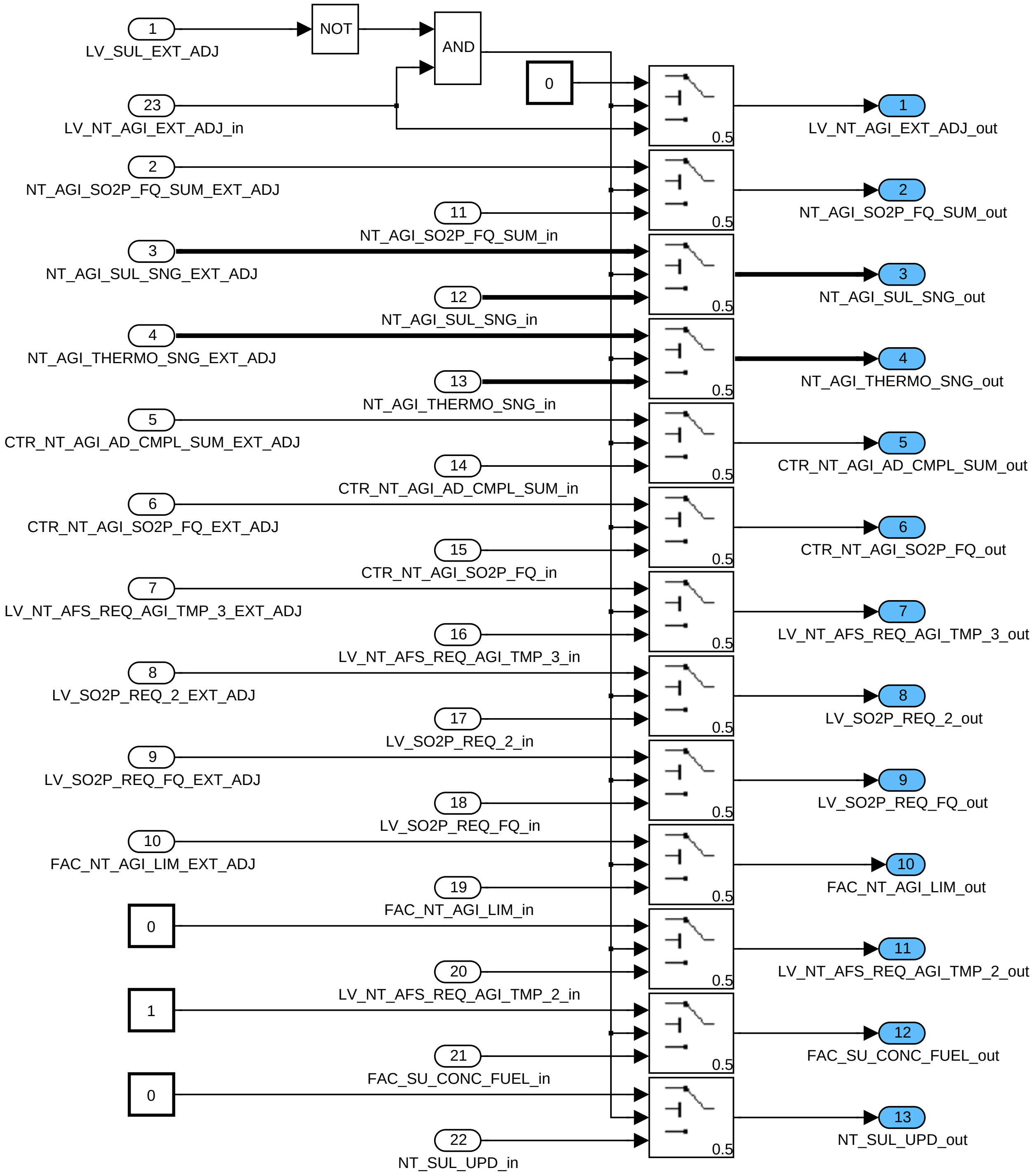


Figure 138 NOXM_MODUL703P/ operate/ NEW_DEVICE/ CLC2

53.12.1.4 RESET

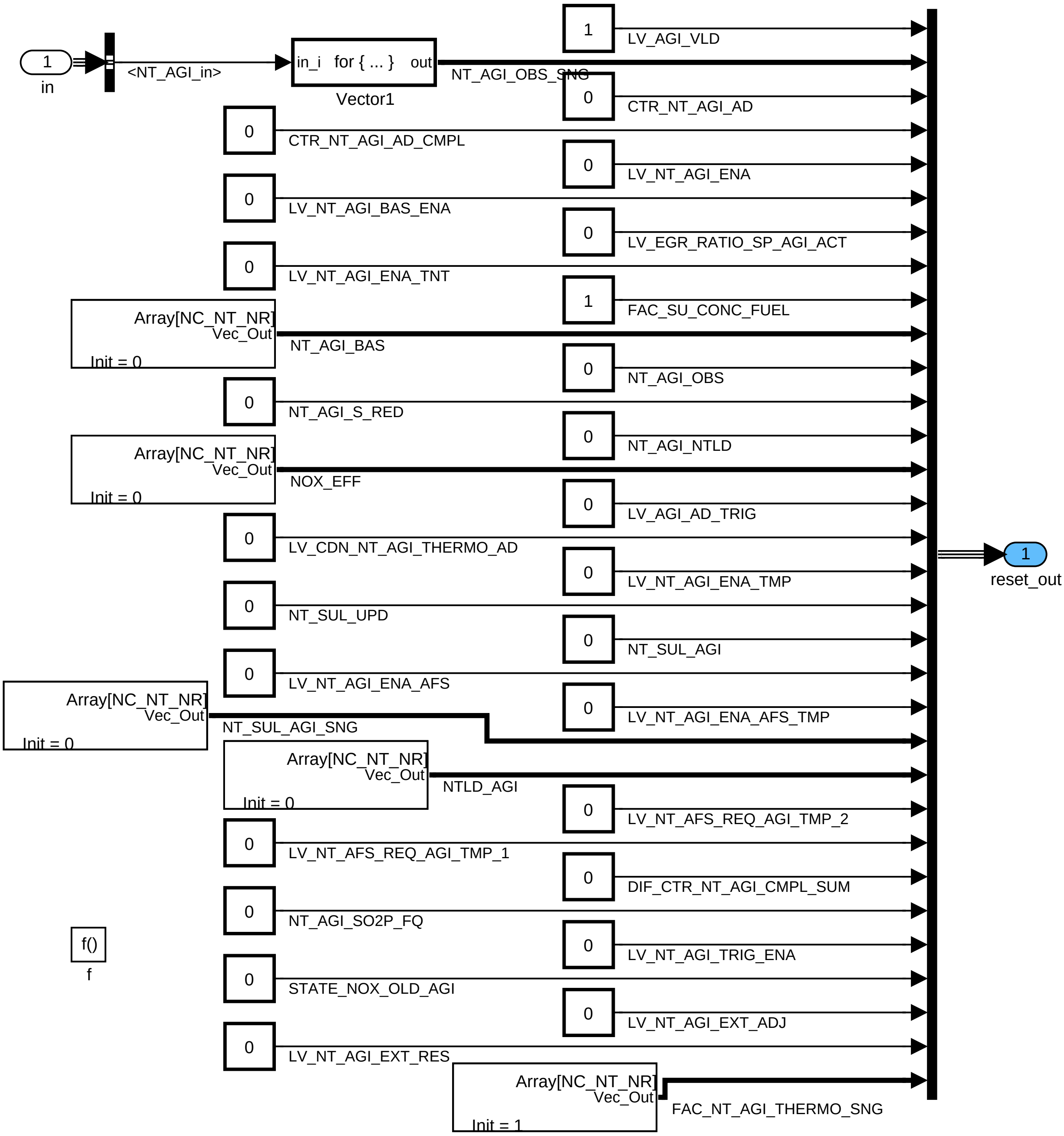


Figure 139 NOXM_MODUL703P/ reset

53.12.1.5 Handling of non volatile memory

The non-volatile data will be read out of the memory at the after power-on and stored in the memory before power-off. The initialization of the non-volatile memory is shown in the next picture.

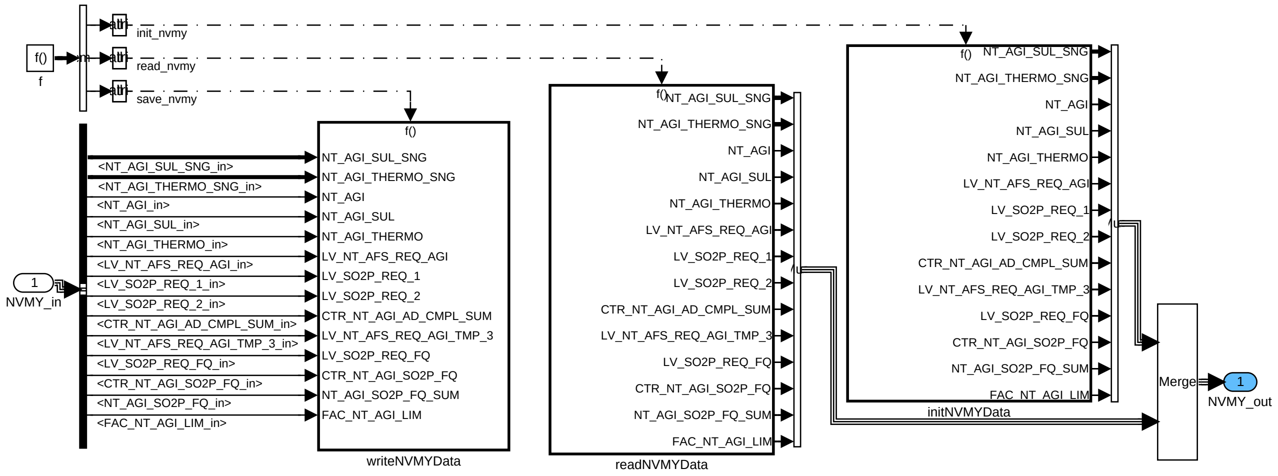


Figure 140 NOXM_MODUL703P/ NVMYdata

Initialization of non-volatile memory

The aging factors will be set to one, that means fresh catalyst. All other values will be initialized to its inactive value.

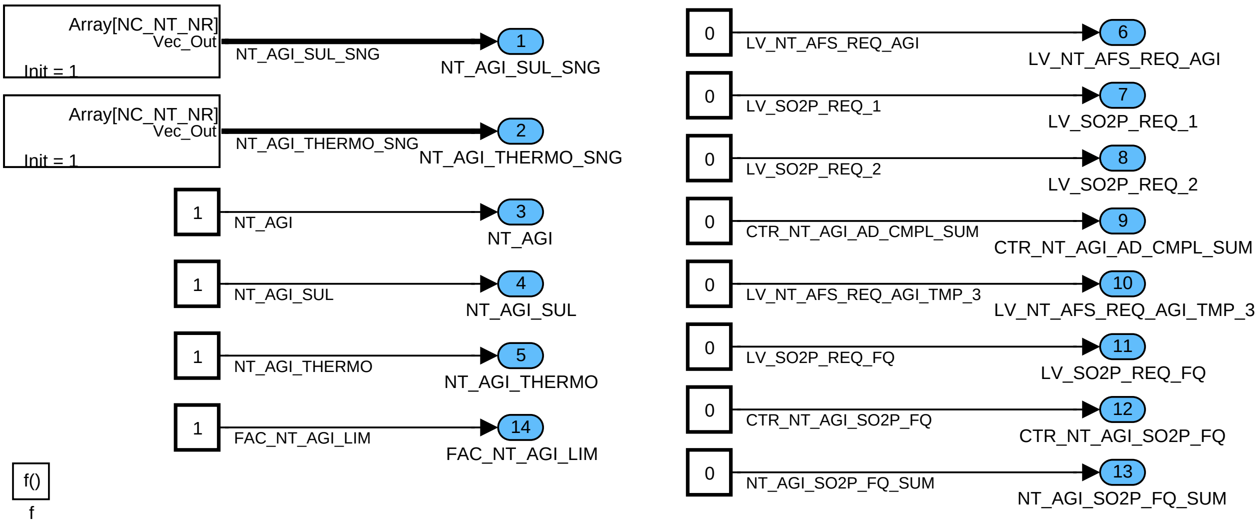


Figure 141 NOXM_MODUL703P/ NVMYdata/ initNVMYData

53.13 Application Incidence for the NOx catalyst aging

53.13.1 Activation Conditions for NOx catalyst aging

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_NT_AGI	V/O	0...1H	0...1	1	[-]
flag inhibiting the NOx catalyst aging adaptation					
LV_INH_NT_AGI_TNT	V	0...1H	0...1	1	[-]
inhibition of the NOx catalyst aging adaptation due to temperature window control					
LV_INH_NT_AGI_ERR	V/O	0...1H	0...1	1	[-]
inhibition of the NOx catalyst aging adaptation due error on sensor or subsystem					
TNT_MDL_NT_AGI_DYW[NC_NT_NR]	V	0...FFFFH	0...1023.98437	0.015625	[°C]
NOx trap temperature of limited dynamic window out of TNT_MDL_MV					
LV_INH_NT_AGI_THD	V	0...1H	0...1	1	[-]
inhibition of the NOx catalyst aging adaptation due to MFF or N					
LV_INH_NT_AGI_BAS	V/O	0...1H	0...1	1	[-]
flag inhibiting the NOx catalyst aging adaptation					
LV_NOX_SENS_MAX_ADJ	V/O	0...1H	0...1	1	[-]
Flag to enable adjustment of NOx Sensor threshold for regeneration by aging adaptation					
LV_NT_AGI_OBS_ENA	V/O	0...1H	0...1	1	[-]
Activation Conditions for continuous Aging Observation					
LV_NT_AGI_OBS_EGR_ENA	V/O	0...1H	0...1	1	[-]
Activation Conditions for continuous Aging Observation					
LV_INH_NT_AGI_TNT_BAS	V/O	0...1H	0...1	1	[-]
inhibition of the NOx catalyst aging adaptation due to temperature window control					
NT_AGI_THERMO_ST	V	0...FFFFH	0...0.99998	0.0153e-3	[-]
NOx trap aging factor due to thermal aging at motor start					
CTR_KM_CAN_ST	V	0...FFFFH	0...655350	10	[km]
vehicle kilometer reading from Kombi at motor start					
NT_AGI_THERMO_GRD	V/O/S	8000...0H	-0.1...0	3.0518e-6	[1/km]
Gradient of NT_AGI_THERMO from motor start to stop of last driving cycle normalized to distance					

Input data:

NC_NT_NR	NC_CBK_EX_NR	NC_NOX_SENS_CONF	TOIL
TNT_MDL_L	TNT_MDL_H	TNT_MDL_MV	N_32
LV_NOX_SENS	STATE_NOX	TIA_IM	AMP
MFF_SP_S	TNT_MDL_1[NC_NT_NR]	TNT_MDL_2[NC_NT_NR]	LV_NT_AGI_ENA
LV_HOM_AFS_REQ	STATE_CAT_DIAG[NC_CBK_EX_NR]	LV_NOX_AD_DYW	LV_ERR_CPS
LV_ERR_EGR	LV_ERR_EGR_2	LV_ERR_TIA_IM	LV_ERR_CAT_DIAG[NC_CBK_EX_NR]
LV_ERR_NS_OBD_1[NC_NOX_SENS_CONF]		LV_ERR_NS_CAN_MSG_LOST[NC_NOX_SENS_CONF]	LV_ERR_EL_LS_DOWN[NC_CBK_EX_NR]
LV_ERR_OBD_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_LSH_DOWN[NC_CBK_EX_NR]	LV_ERR_TEG_PCAT_DOWN[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]
NT_AGI_THERMO	CTR_KM_CAN	STATE_ERR_NS_OBD_2[NC_NOX_SENS_CONF]	

FUNCTION DESCRIPTION:

General information:

This flag deactivates the NOx catalyst aging adaptation when errors occure which must not stop the lean operation and the NOx catalyst management, but can cause an inaccuracy in the model, which must not be adapted.

With the calibration data LC_INH_NT_AGI_MAN it is possible to deactivate the NOx trap aging adaptation manually.
With the calibration data LC_INH_NT_AGI_OFF it is possible to make the adaptation allways working. These bits are useful for the calibration phase.

Number of NOx catalyst:	NC_NT_NR	i = 1...NC_NT_NR
Number of NOx sensors:	NC_NOX_SENS_CONF	k = 1...NC_NOX_SENS_CONF
Number of lambda sensors:	NC_CBK_EX_NR	m = 1...NC_CBK_EX_NR

Typical configurations:
[NC_CBK_EX_NR:NC_NT_NR:NC_NOX_SENS_CONF] =[1:1:1], [2:2:2], [2:2:1], [2:1:1]

Application conditions:

Recurrence: 10 ms
Activation: --
Deactivation: --
Initialisation: --

Formula section:

```
if(0)      STATE_NOX = STOP --> !STOP
then(0)    LV_INH_NT_AGI_THD = 0
           LV_INH_NT_AGI_TNT = 0
           TNT_MDL_NT_AGI_DYW[i] = TNT_MDL_1[i]
endif(0)
```

Application conditions:

Recurrence: 100 ms has to be calculated before the modul "NOX Catalyst aging" xx703P03.00x

Activation: --

Deactivation: --

Initialisation: LV_INH_NT_AGI = 0
TNT_MDL_NT_AGI_DYW[i] = 0

```
% NOx catalyst temperature dynamic window
$\mathrm{TNT\_MDL\_NT\_AGI\_DYW[i]}_{n}$ =
        ($\mathrm{TNT\_MDL\_NT\_AGI\_DYW[i]}_{n-1}$ * (1- C_CRLC_TNT_MDL_NT_AGI_DYW))
      + (TNT_MDL_1[i] * C_CRLC_TNT_MDL_NT_AGI_DYW)

% NOx catalyst temperature conditions for aging
if (1)   | TNT_MDL_NT_AGI_DYW[i] - TNT_MDL_1[i] | > C_TNT_MDL_NT_AGI_DYW
    % at least one of the two benches has to fulfill the conditions
    or          TNT_MDL_L <= C_TNT_MDL_NT_AGI_MIN
    or          TNT_MDL_H >= C_TNT_MDL_NT_AGI_MAX
    or          TNT_MDL_MV <= C_TNT_MDL_MV_AGI_MIN
    or          TNT_MDL_MV >= C_TNT_MDL_MV_AGI_MAX
    or          TNT_MDL_H - TNT_MDL_L >= C_TNT_MDL_NT_AGI_DELTA
    or          TIA_IM < C_TIA_NT_AGI_MIN
    or          TOIL <= C_TOIL_NT_AGI_MIN
    or          TOIL > C_TOIL_NT_AGI_MAX
    or          AMP < C_AMP_NT_AGI_MIN
then (1)        LV_INH_NT_AGI_TNT_BAS = 1
else (1)        LV_INH_NT_AGI_TNT_BAS  = 0
endif (1)

if(2)       STATE_NOX = LOAD and LV_HOM_AFS_REQ = 0
then(2)     % NOx catalyst temperature dynamic window
    % NOx catalyst temperature conditions for aging
    if(3)       LV_INH_NT_AGI_TNT_BAS = 1
    then(3)     LV_INH_NT_AGI_TNT = 1
    else(3)     LV_INH_NT_AGI_TNT remains unchanged
```

```
        endif(3)


        % monitoring of engine speed and Fuel mass flow
        if(4)       N_32 <= C_N_32_NT_AGI_MIN
        or          N_32 >= C_N_32_NT_AGI_MAX
        or          MFF_SP_S <= C_MFF_SP_S_NT_AGI_MIN
        or          MFF_SP_S >= C_MFF_SP_S_NT_AGI_MAX
        or          LAMB_LS_UP[m] < C_LAMB_NT_AGI_MIN
                    % valid if at least one [m] < C_LAMB_NT_AGI_MIN
        then(4)     LV_INH_NT_AGI_THD = 1
        else(4)     LV_INH_NT_AGI_THD remains unchanged
        endif(4)
    else(2)     LV_INH_NT_AGI_THD remains unchanged
                LV_INH_NT_AGI_TNT remains unchanged
    endif(2)

    if (5)      LC_INH_NT_AGI_OFF = 1
    then (5)    LV_INH_NT_AGI_ERR = 0
                LV_INH_NT_AGI = 0

    else (5)
        % component error check for aging
        LV_INH_NT_AGI_ERR =
        (       LV_ERR_TEG_PCAT_DOWN[m] = 1             % valid if at least one [m] = 1
                (error currently present on location “Exh.gas temp.sens.downstr. precat”)
        or      LV_ERR_TIA_IM = 1
                (error currently present on location “temperature sensor intake air”)
        or      LV_ERR_EGR = 1
                (error currently present on exhaust gas recirculation system)
        or      LV_ERR_EGR_2  = 1
                (EGR valve stuck)
        or      LV_ERR_CPS = 1
                (electr.error currently present on error location canister purge solenoid)
        or      LV_ERR_CAT_DIAG[m] = 1                  % valid if at least one [m] = 1
                (pre catalyst diagnosis value reached the threshold)

        (Downstream lambda sensor specific errors)
        or      [LV_NOX_SENS = 0
                and (           (failure bit for downstream oxygen sensor)
                        or      LV_ERR_EL_LS_DOWN[m] = 1    % valid if at least one [m] = 1
```

```
                 or      LV_ERR_LSH_DOWN[m] =1          % valid if at least one [m] = 1
                         (downstr. oxygen sensor heater power stage)
                 or      LV_ERR_OBD_LSH_DOWN[m] = 1)] % valid if at least one [m] = 1

        (NOx sensor specific errors)
        or   [LV_NOX_SENS = 1
             and (       LV_ERR_NS_OBD_1[k] = 1         % valid if at least one [k] = 1
                         (OBD I error from smart NOx sensor)
                 ~~or      LV_ERR_NS_OBD_2[k] = 1         % valid if at least one [k] = 1~~
                 or      ( STATE_ERR_NS_OBD_2[k] bitwise AND
                           C_ERR_NS_NT_AGI_BIT_SEL ) != 0  % valid if at least one [k] = 1
                         (OBD II error on NOx sensor)
                 or      LV_ERR_NS_CAN_MSG_LOST[k] =1
                         % valid if at least one [k] = 1
                 )
             ]
        )

        % building the central inhibition flag for aging function
        LV_INH_NT_AGI =
        (    LC_INH_NT_AGI_MAN = 1
             (codeword for manual setting of inhibiting the aging adaptation)
        or   LV_INH_NT_AGI_TNT = 1
             (NOx trap temperature window control)
        or   LV_INH_NT_AGI_ERR = 1
             (error on adaptation relevant sensors / subsystems present)
        or   LV_NOX_AD_DYW = 0
             (limited dynamics concerning NOx concentration and MAF)
        or   LV_INH_NT_AGI_THD = 1)
    endif (5)

    LV_INH_NT_AGI_BAS =
        (    LC_INH_NT_AGI_MAN = 1
             (codeword for manual setting of inhibiting the aging adaptation)
        or   LV_INH_NT_AGI_ERR = 1)
```

53.13.2 Adjustment of NOx Sensor threshold for regeneration

```
    If (6)   LC_NOX_SENS_MAX_AGI_ADJ_ENA =1
      and    STATE_CAT_DIAG[m] = "END"                % valid if all [m] = "END"
      and    LV_NT_AGI_ENA = 1
```

```
then (6)    LV_NOX_SENS_MAX_ADJ = 1
else  (6)   LV_NOX_SENS_MAX_ADJ = 0
endif (6)
```

53.13.3 Activation Conditions for continuous Aging Observation

```
if (7)      STATE_CAT_DIAG[m] = “END”                % valid if all [m] = “END”
then (7)    LV_NT_AGI_OBS_ENA = 1
            LV_NT_AGI_OBS_EGR_ENA = 0
else (7)    LV_NT_AGI_OBS_ENA = 0
            if (8)          LV_ERR_EGR
            then (8)        LV_NT_AGI_OBS_EGR_ENA = 1
            endif (8)
endif (7)
```

53.13.4 Gradient of NT_AGI_THERMO

<u>Application conditions:</u>

Recurrence: --

Activation: --

Deactivation: --

Initialisation:

```
        at transition 'engine stop to engine run':
                NT_AGI_THERMO_ST = NT_AGI_THERMO
                CTR_KM_CAN_ST = CTR_KM_CAN
        at transition 'engine run to engine stop':
                if (CTR_KM_CAN-CTR_KM_CAN_ST) > C_CTR_KM_CAN_DIF_MIN
                        AND NT_AGI_THERMO < NT_AGI_THERMO_ST
                then
                    NT_AGI_THERMO_GRD =
                        (NT_AGI_THERMO – NT_AGI_THERMO_ST) /
                        (CTR_KM_CAN – CTR_KM_CAN_ST)
                endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_NT_AGI_MAN	1	0...1H	0...1	1	[-]
Deativation flag (for the manual setting of the inhibiting bit NOx trap aging adaptation)					
LC_INH_NT_AGI_OFF	1	0...1H	0...1	1	[-]
Deativation flag for the deactivation of inhibiting the NOx trap aging adaptation (1 = disenable deactivation)					
C_TNT_MDL_NT_AGI_MIN	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabling / disabling an aging adaptation					
C_TNT_MDL_NT_AGI_MAX	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabling / disabling an aging adaptation					
C_TNT_MDL_NT_AGI_DELTA	1	0...FFH	0...1020	4	[°C]
max. deviation between higher and lower NT temp. for enabling an aging adaptation					
C_TNT_MDL_NT_AGI_DYW	1	0...FFH	0...1020	4	[°C]
Temperature threshold for dynamic window limitation					
C_CRLC_TNT_MDL_NT_AGI_DYW	1	0...FFFFH	0...0.99998	0.0153e-3	[-]
Correlation factor for calculation of TNT of dynamic window limitation					
C_N_32_NT_AGI_MIN	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for enabling / disabling an aging adaptation					
C_N_32_NT_AGI_MAX	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for enabling / disabling an aging adaptation					
C_MFF_SP_S_NT_AGI_MIN	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass flow threshold for enabling / disabling an aging adaptation					
C_MFF_SP_S_NT_AGI_MAX	1	0...FFFFH	0...1389	0.0211948	[mg/stk]
Fuel mass flow threshold for enabling / disabling an aging adaptation					
C_LAMB_NT_AGI_MIN	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
Lambda threshold for enabling / disabling an aging adaptation					
C_TIA_NT_AGI_MIN	1	0...FEH	-48...142.5	0.75	[°C]
Min. Intake Air Temperature threshold for enabling / disabling an aging adaptation					
C_TOIL_NT_AGI_MIN	1	0...C8H	-40...160	1	[°C]
Min. Oil Temperature threshold for enabling / disabling an aging adaptation					
C_TOIL_NT_AGI_MAX	1	0...C8H	-40...160	1	[°C]
Max. Oil Temperature threshold for enabling / disabling an aging adaptation					
C_AMP_NT_AGI_MIN	1	0...FFFFH	0...5434	0.0829175	[hPa]
Min. Ambient Pressure threshold for enabling / disabling an aging adaptation					
C_TNT_MDL_MV_AGI_MIN	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabling / disabling an aging adaptation					
C_TNT_MDL_MV_AGI_MAX	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabling / disabling an aging adaptation					
LC_NOX_SENS_MAX_AGI_ADJ_ENA	1	0...1H	0...1	1	[-]
Switch to enable adjustment of NOx Sensor threshold for regeneration by aging adaptation					
C_ERR_EGR_3_REQ_CTR_MIN	1	0...FFH	0...255	1	[-]
threshold for enabling aging factor observation considering EGR problems					
C_CTR_KM_CAN_DIF_MIN	1	0...FFFFH	0...655350	10	[km]
minimum vehicle kilometer for calculation of thermal aging gradient					
C_ERR_NS_NT_AGI_BIT_SEL	1	0...FFFFH	0...65535	1	[-]
Selection of NOx sensor diagnosis errors to inhibit NOx catalyst aging determination					

53.14 NOx catalyst sulphur model

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NT_SUL_32[NC_NT_NR]	O/V/S	0...FFFFFFFFH	0...1.04856E+4	2.44137E-6	mg
NOx trap sulphur loading with high resolution					
NT_SUL	O/V	0...FFFFH	0...1.04856E+4	0.16	mg
NOx trap sulphur loading					
NT_SUL_H_32[NC_NT_NR]	O/V/S	0...FFFFFFFFH	0...1.04856E+4	2.44137E-6	mg
NOx trap sulphur loading with high resolution for high sulphured fuel					
NT_SUL_H	O/V	0...FFFFH	0...1.04856E+4	0.16	mg
NOx trap sulphur loading for high sulphured fuel					
NT_SUL_MAX[NC_NT_NR]	O/V	0...FFFFH	0...1.04856E+4	0.16	mg
maximum NOx trap sulphur loading					
NT_AGI_INC_SO2P[NC_NT_NR]	O/V	0...FFFFFFFFH	0...1	2.3283E-10	-
Nox trap ageing factor increment per time step due to sulphur release					
LV_SUL_EXT_ADJ	O/V	0...1H	0...1	1	-
External data for sulfur model available					
LV_SUL_EXT_RES	O/V	0...1H	0...1	1	-
External reset of data for sulfur model					
NT_SUL_H_DELTA[NC_NT_NR]	V	80000000...7FFFFFFFH	-5.2428E+3 ... 5.2428E+3	2.44137E-6	mg
NT_SUL_H delta due to desulfation or loading					
NT_SUL_H_SO2P_BEG[NC_NT_NR]	V	0...FFFFH	0...1.04856E+4	0.16	mg
NOx trap sulphur loading for high sulphured fuel at desulfation begin					
NT_SUL_AGI_DELTA[NC_NT_NR]	V	80000000...7FFFFFFFH	-5.2428E+3 ... 5.2428E+3	2.44137E-6	mg
NT_SUL_AGI delta due to desulfation					
NT_SUL_DELTA[NC_NT_NR]	V	80000000...7FFFFFFFH	-5.2428E+3 ... 5.2428E+3	2.44137E-6	mg
NT_SUL delta due to desulfation or loading					
NT_SUL_SO2P_BEG[NC_NT_NR]	V	0...FFFFH	0...1.04856E+4	0.16	mg
NOx trap sulphur loading at desulfation begin					
MFF_OFS_EGR_NT_SUL	-	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Correction of mass fuel flow setpoint for sulphur model due to unsymmetrical EGR branch					
NT_SUL_UPD_OLD	-	0...FFFFH	0...1.04856E+4	0.16	mg
old value of interface variable for update of sulphur model by aging function					

Input data:

LV_SO2P_ACT	T_SO2P_DLY	C_NT_STC_BAS_MAX_RNG_H	FAC_SU_CONC_FUEL
N_32	STATE_NOX	LV_PUC	NT_SUL_UPD
LAMB_SP[NC_CBK_EX_NR]	LAMB_LS_UP[NC_CBK_EX_NR]	TNT_MDL_MV_SNG[NC_NT_NR]	NT_AGI_THERMO_SNG[NC_NT_NR]
MFF_SP[NC_CBK_EX_NR]	NT_SUL_AGI_SNG[NC_NT_NR]	NC_CBK_EX_NR	NC_NT_NR
EGR_RATIO	NT_SUL_32_EXT_ADJ[NC_NT_NR]	NT_SUL_H_32_EXT_ADJ[NC_NT_NR]	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_SU_NOX	1	0...FFFFH	0...2.81215E-8	4.29E-13	-
Correlation constant					
C_LAMB_SP_MIN_SUL	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Lambda threshold for sulphur poisonning in homogenous lambda 1					
C_NT_SUL_INI	1	0...FFFFH	0...1.04856E+4	0.16	mg
Manual initialisation value for NOx trap sulphur model					
C_STATE_EGR_CBK	1	0...2H	0...2	1	-
Configuration of asymmetrical EGR branch for sulphur model (0: no unsymmetry, 1 or 2: EGR branch on this bank)					
C_SU_CONC_FUEL	1	0...FFH	0...510	2	ppm
Fuel sulphur concentration					
C_SU_CONC_FUEL_H	1	0...FFH	0...510	2	ppm
Fuel sulphur concentration for high sulphured fuel					
LC_NT_SUL_INI_MAN	1	0...1H	0...1	1	-
Logical constant for manual initialisation of NOx trap sulphur model					
LC_NT_SUL_LIM_NOT	1	0...1H	0...1	1	-
Logical constant for continue incrementation of the layer 3 above NT_SUL_MAX_3					
IP_CRLC_SUL_AFS	4	0...FFFFH	0...1	1.5259E-5	-
LDP_TNT_MDL_MV_IP_CRLC_SUL_AFS	4	0...FFFFH	0...1.02398E+3	0.015625	°C
Conversion factor for sulphur poisonning in homogenous lambda 1 according to TNT_MDL_MV					
IP_NT_SUL_DEC_FAC_2	6	0...FFFFH	0...1	1.5259E-5	-
LDP_NT_AGI_THERMO_IP_NT_SUL_2	6	0...FFFFH	0...0.99998474	1.52588E-5	-
Correction factor for sulphur release on all layers on thermal aging factor					
IP_NT_SUL_MAX	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDPM_NT_AGI_THERMO	6	0...FFFFH	0...0.99998474	1.52588E-5	-
maximum sulphur loading for layer 1 respecting NT_AGI_THERMO					
IP_T_SO2P_DLY	4	0...FFFFH	0...6.5535E+3	0.1	s
LDPM_LAMB_MV_IP_T_SO2P_DLY	4	0...7FFFH	0...31.9990234	9.76563E-4	-
Delay time before sulfur release					
IP_NT_SUL_DEC_BAS	4x4	0...FFFFH	0...40.959375	6.25E-4	mg
LDPM_LAMB_MV_IP_NT_SUL_DEC_BAS	4	0...7FFFH	0...31.9990234	9.76563E-4	-
LDP_TNT_MDL_MV_IP_NT_SUL_DEC	4	0...FFFFH	0...1.02398E+3	0.015625	°C
Basic sulphur loading decrement per time step (layer 1)					
IP_NT_SUL_DEC_FAC_1	6x6	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_NT_SUL_IP_NT_SUL_DEC_FAC_1	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDP_NT_SUL_SO2P_BEG_IP_NT_SUL_1	6	0...FFFFH	0...1.04856E+4	0.16	mg
Correction factor for sulphur release					
IP_NT_SUL_H_DEC_FAC_1	6x6	0...FFFFH	0...0.99998474	1.52588E-5	-
LDPM_NT_SUL_IP_NT_SUL_DEC_FAC_1	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDP_NT_SUL_SO2P_BEG_IP_NT_SUL_1	6	0...FFFFH	0...1.04856E+4	0.16	mg
Correction factor for sulphur release for high sulphured fuel					

Export actions:

ACTION_NOXM_CleanNTAdaptSul()
Initialization of NOx trap adaptation values of sulfur module
ACTION_NOXM_WriteSulfurExtAdj()
New external values for sulfur model available

53.14.1 NOx catalyst sulphur model (NOXM modul 704G)

General information:

The NOx catalyst sulphur model can handle single branched and twin branched exhaust gas lines with the configuration data NC_NT_NR defined in the NOx catalyst management module.

If NC_NT_NR=1, just one exhaust line is taken into account, if NC_NT_NR=2, the software calculates the twin branch version.

i belongs to NC_NT_NR.

Description:

The NOx catalyst sulphur model calculates the NOx catalyst sulphur loading.

The sulphur loading is increased only in lean burn phase (-> STATE_NOX = LOAD). The sulphur loading increment is calculated out of the fuel sulphur concentration and converted to a NOx mass loading equivalent (sulphur load in NO2 mass loading equivalent is loss of NOx storage capacity).

When the desulfation is active, the sulphur which is chemically stored in the NOx-trap is released depending on the NOx-trap monolith temperature, lambda, the modeled sulphur load NT_SUL on the corresponding layer and the value of NT_AGI_THERMO.

During the desulfation the NOx-trap sulphur load is decremented until it reaches 0.

C_CRLC_SU_NOX includes

- the factor for transformation of MFF_SP * N_32 {mg/stk * 1/min} to a fuel mass flow
  {mg / TA} =0,1s / 2 / 60s = 1 / 1200)
- and the factor for number of cylinders and number of NOx traps
  = NC_CYL_NR / NC_NT_NR
- and the factor for transformation of C_SU_CONC_FUEL {ppm} to mass part sulphur
  in fuel {mg S/mg fuel} = 1E-6
- and the factor for transformation of sulphur load {mg S} to sulphur load NO2 equivalent
  {mg NO2} = 2 * 46 / 32 = 2,875.

Example values:

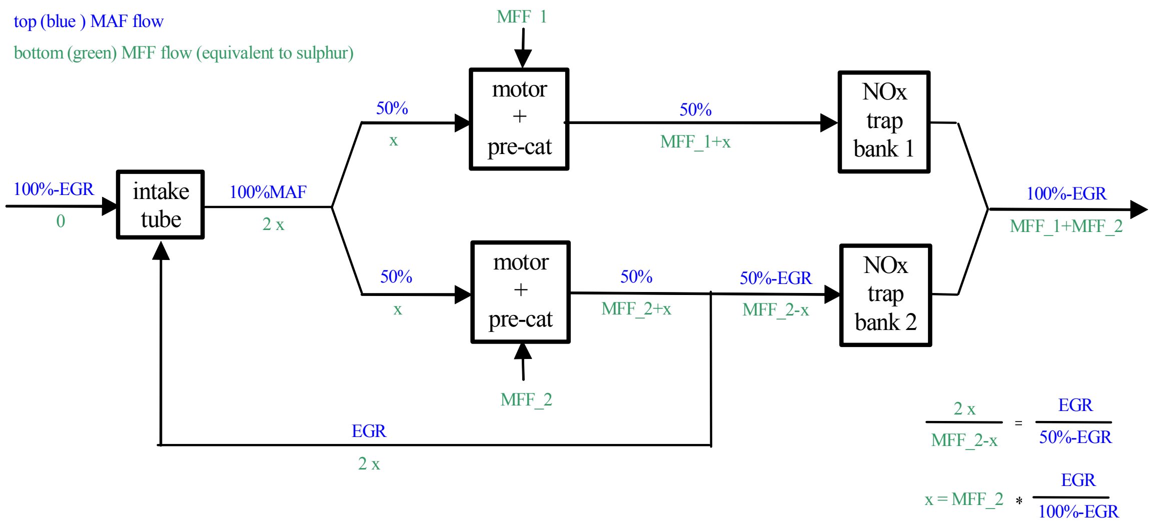
- four cylinder and one NOx trap: 9,5833 E-9
- six cylinder and two NOx traps: 7,1875 E-9

Calculation inside this function is done by high resolution variable NT_SUL_32. Valid output to other functions is low resolution variable NT_SUL.

Unsymmetrical EGR branch

If the exhaust gas recirculation is unsymmetrical (exaust gas is taken from one bank and added to the intake air for both banks) then the different sulfur loading of the NOx traps will be considered by setting C_STATE_EGR_CBK to the bank with the EGR branch. In a symmetrical or a one-bank-system C_STATE_EGR_CBK have to be set to 0.

The splitting of the sulfur to both NOx traps is shown in the following picture. The sulphur is proportional to the fuel flow and the fuel split is controlled by the air flow, which depends from the EGR ratio.



Application Condition

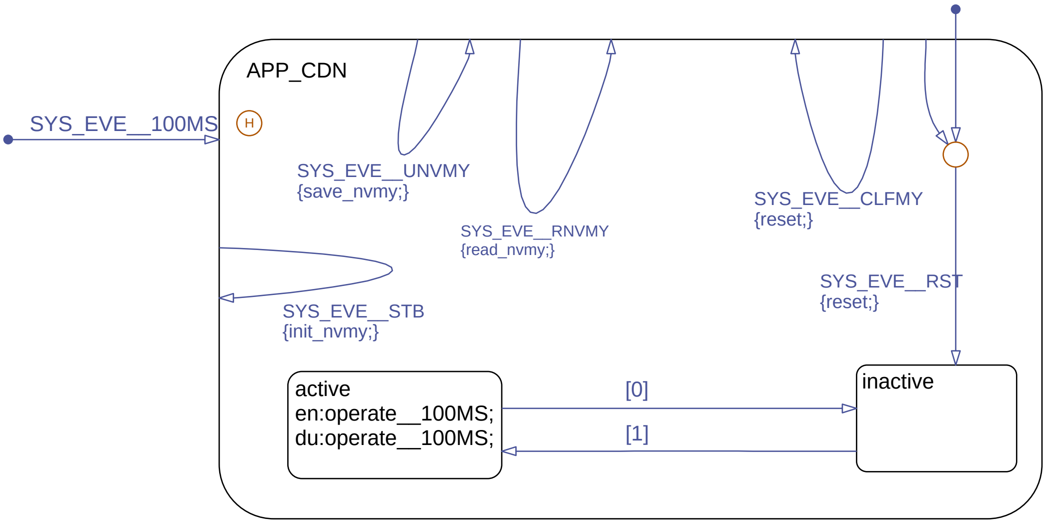
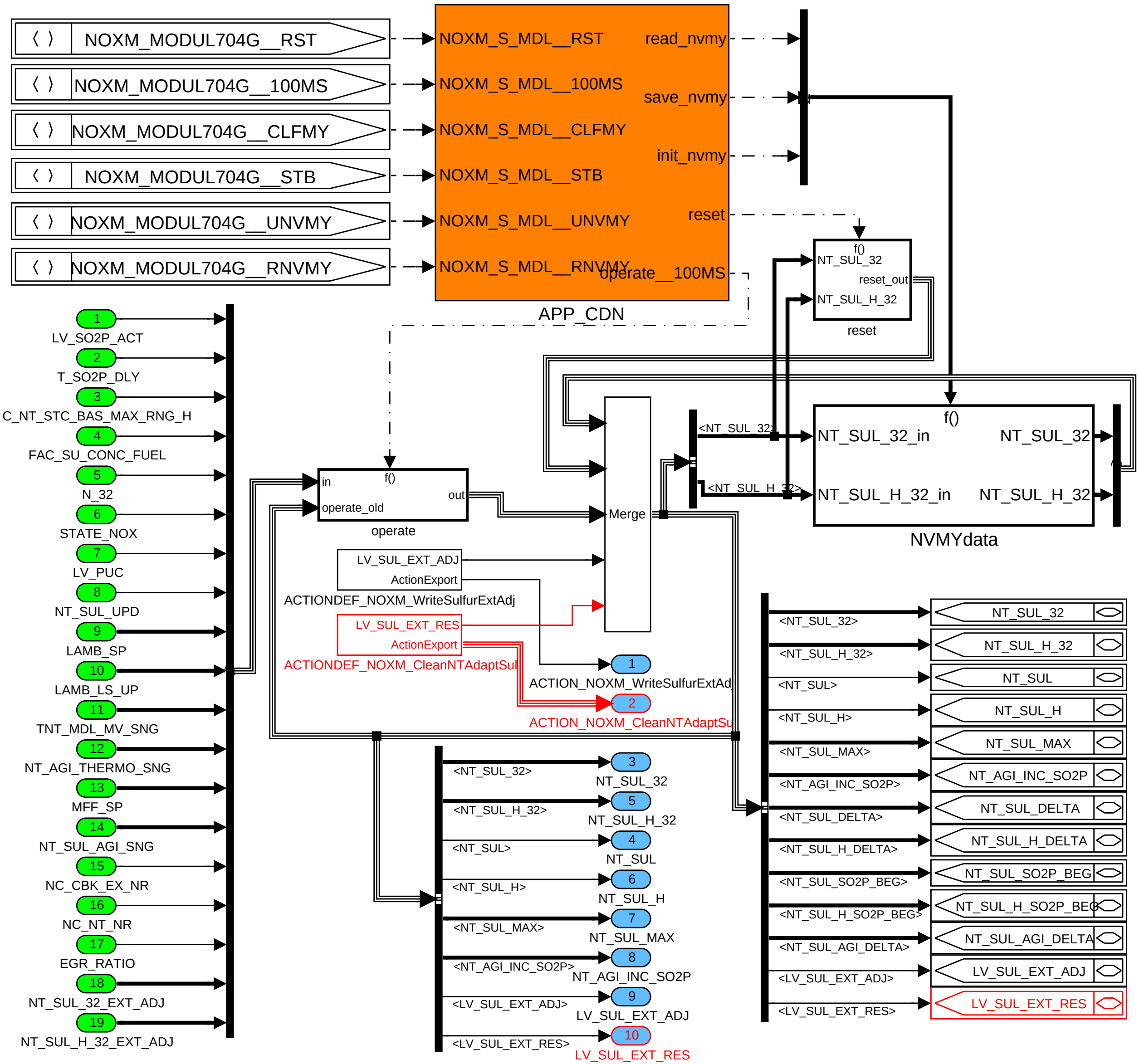


Figure 142 NOXM_MODUL704G/ APP_CDN/ Chart



SDA_SRS / SDA 4.0 01−Dec−2006

Figure 143 NOXM_MODUL704G

53.14.1.1 SUBFUNCTION: ACTIONDEF_NOXM_CleanNTAdaptSul

Description for ACTION_NOXM_CleanNTAdaptSul

ACTION_NOXM_CleanNTAdaptSul()					
Initialization of NOx trap adaptation values of sulfur module					
Parameter	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**

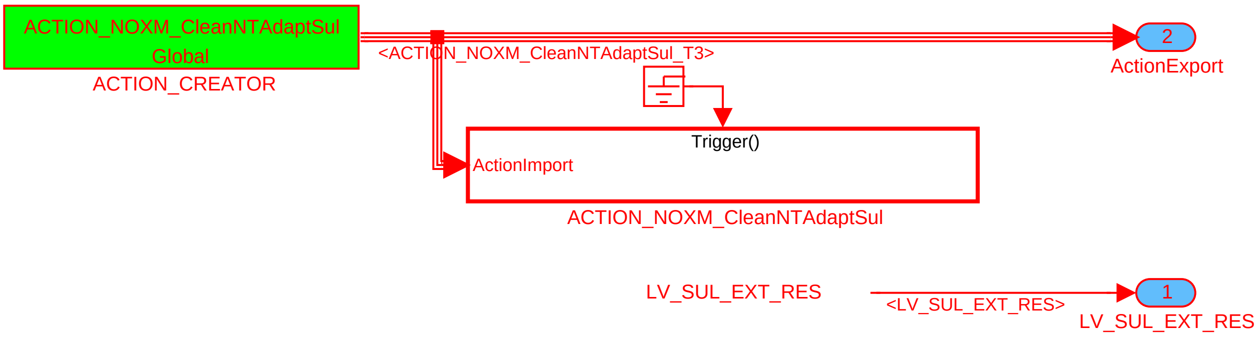


Figure 144 NOXM_MODUL704G/ ACTIONDEF_NOXM_CleanNTAdaptSul

The flag LV_SUL_EXT_RST is set by this action. It shows that the sulfur values have to be reset inside next 100 ms recurrence.

Figure 145 NOXM_MODUL704G/ ACTIONDEF_NOXM_CleanNTAdaptSul/ ACTION_NOXM_CleanNTAdaptSul

53.14.1.2 SUBFUNCTION: ACTIONDEF_NOXM_WriteSulfurExtAdj

Description for ACTION_NOXM_WriteSulfurExtAdj

ACTION_NOXM_WriteSulfurExtAdj()					
New external values for sulfur model available					
Parameter	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**

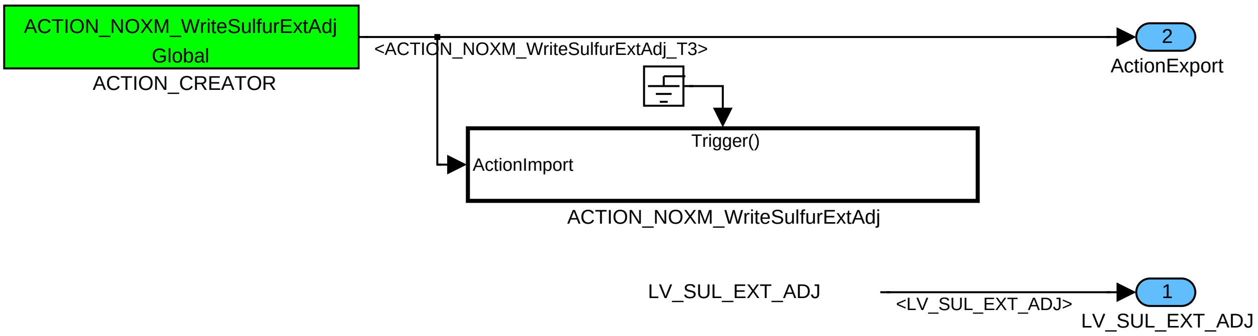


Figure 146 NOXM_MODUL704G/ ACTIONDEF_NOXM_WriteSulfurExtAdj

The flag LV_SUL_EXT_ADJ is set by this action. It shows that the external values have to be used inside next 100 ms recurrence.

Figure 147 NOXM_MODUL704G/ ACTIONDEF_NOXM_WriteSulfurExtAdj/ ACTION_NOXM_WriteSulfurExtAdj

53.14.1.3 OPERATE

Calculation before MAIN

Before the MAIN part, which is calculated for each bank, the difference of the fuel due to the unsymmetrical EGR will be calculated. It will be added or subtracted inside MAIN to both banks.

(If MFF_SP_2 is not existing caused by NC_CBK_EX_NR = 1, but access demanded by other variables depending on other indices of bigger size, then MFF_SP_1 is used.)

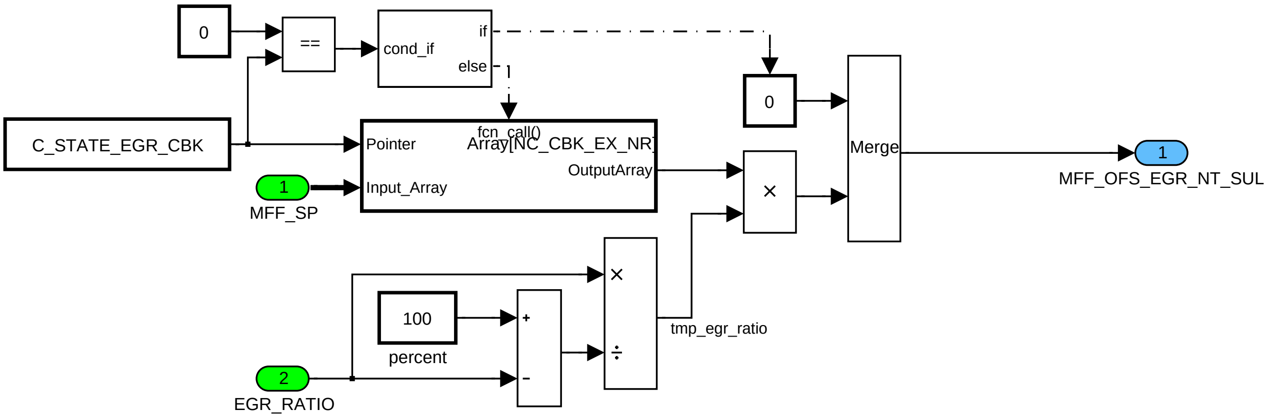


Figure 148 NOXM_MODUL704G/ operate/ BEFORE_MAIN

MAIN (for each NOx trap)

The function consists of the following parts

- manual initialization
- sulphur load update (from NOx catalyst aging)
- sulphur loading during LOAD phase
- sulphur release during desulfation

  (splitted in two parts: before and after T_SO2P_DLY)

The for loop of the MAIN part is controlled by the number of NOx traps

NC_NT_NR = 1 means one NOx trap (calculation for i = 1)

NC_NT_NR = 2 means two NOx traps (calculation for i = 1 and 2)

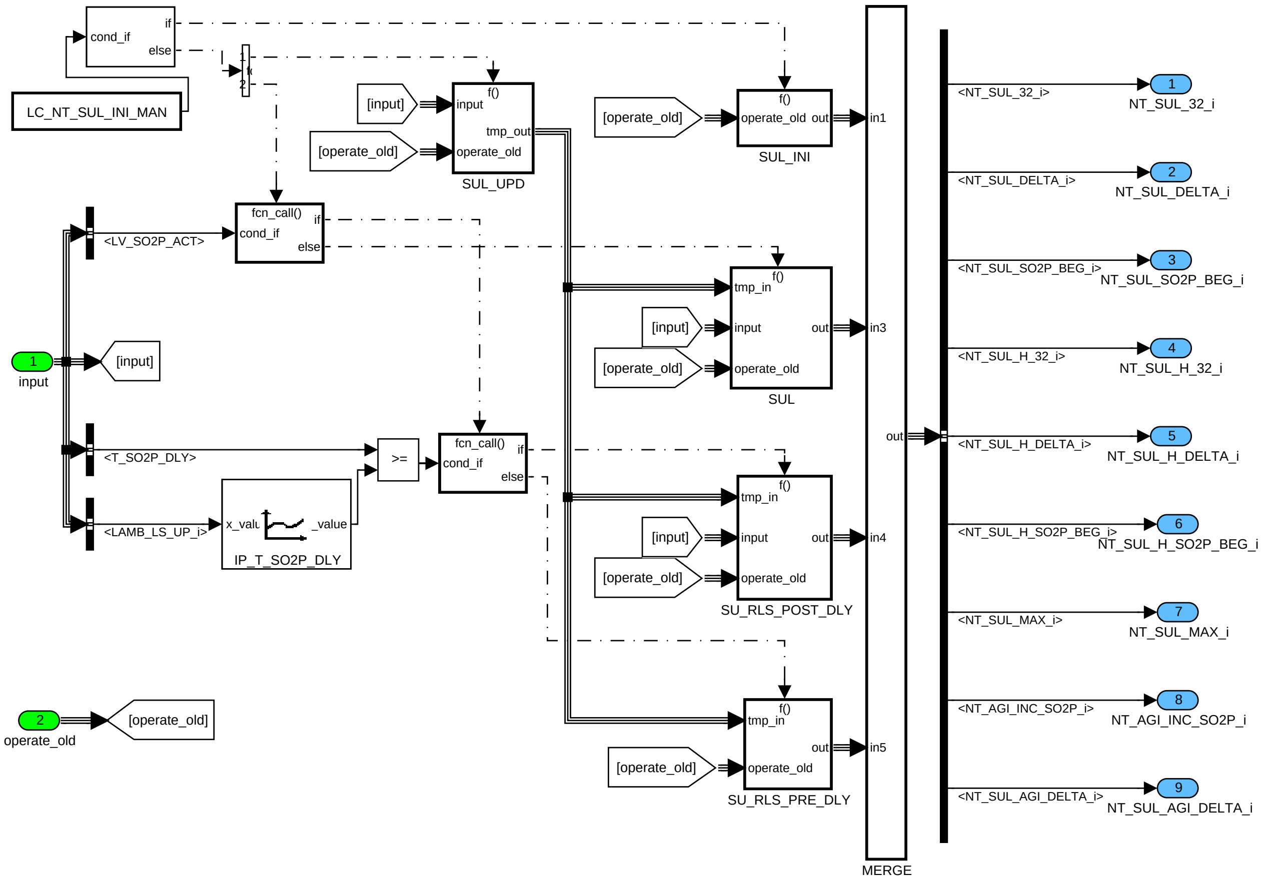


Figure 149 NOXM_MODUL704G/ operate/ CBK_MNG/ MAIN

Manual initialization

If LC_NT_SUL_INI_MAN ist set, then the sulfur load of the NOx traps will be initialized with C_NT_SUL_INI.

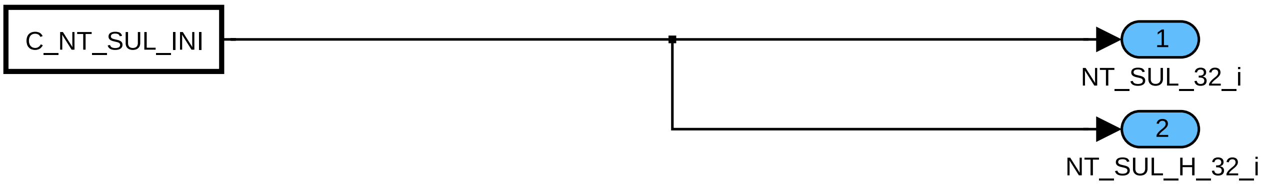


Figure 150 NOXM_MODUL704G/ operate/ CBK_MNG/ MAIN/ SUL_INI/ CLC

Sulphur load update

The update value NT_SUL_UPD comes from the aging modul. The update takes place if the value is new and not zero.

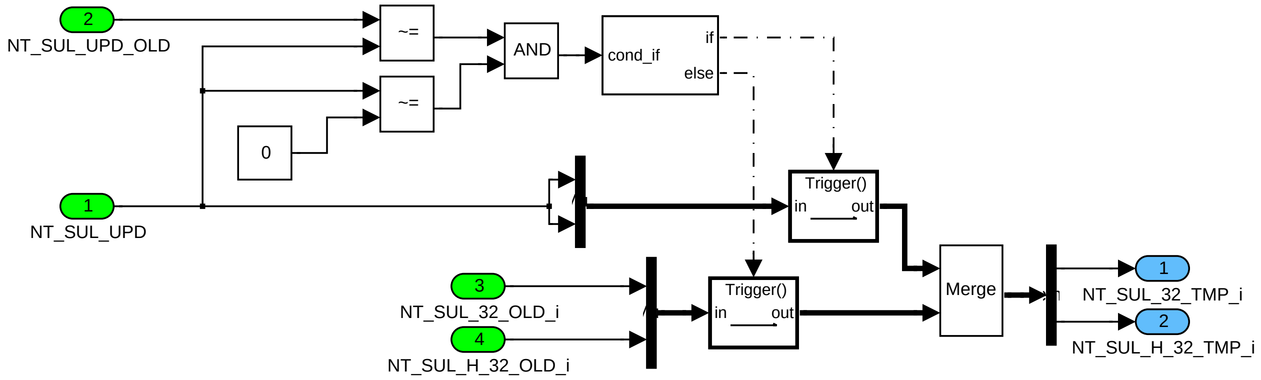


Figure 151 NOXM_MODUL704G/ operate/ CBK_MNG/ MAIN/ SUL_UPD/ CLC

Calculation of sulphur loading increment

The offset of the EGR depending fuel will be added to the bank without EGR branch and subtracted to the bank with EGR. C_STATE_EGR_CBK selects the bank with EGR branch. From these corrected fuel values the sulphur increment for the NOx traps will be calculated.

The temporary variable mff_sp_tmp_i has the same resolution and limits as MFF_SP.

(If MFF_SP_2 is not existing caused by NC_CBK_EX_NR = 1, but access demanded by other variables depending on other indices of bigger size, then MFF_SP_1 is used.)

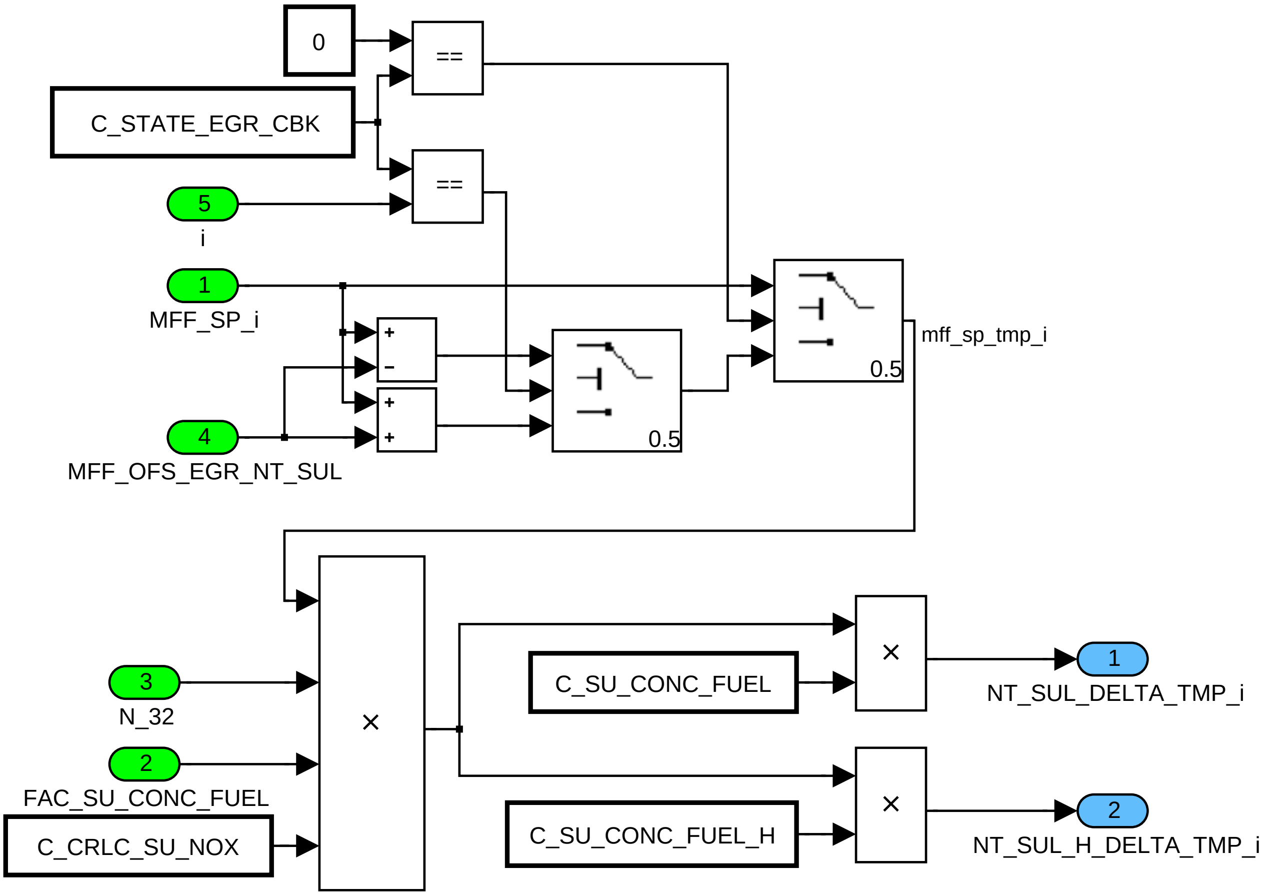


Figure 152 NOXM_MODUL704G/ operate/ CBK_MNG/ MAIN/ SUL/ CLC/ CLC1

Selection of sulphur loading increment

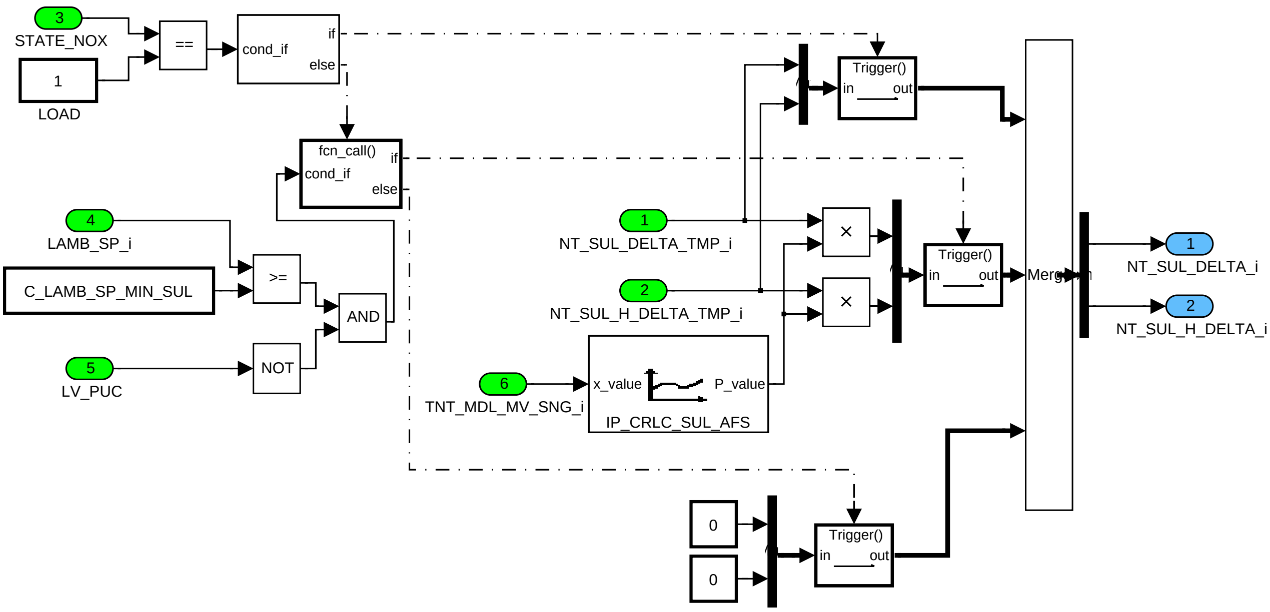


Figure 153 NOXM_MODUL704G/ operate/ CBK_MNG/ MAIN/ SUL/ CLC/ CLC2

Integral of sulphur

The sulphur increments will be added until a temperature dependent maximum. The limitation may be disabled. by LC_NT_SUL_LIM_NOT.

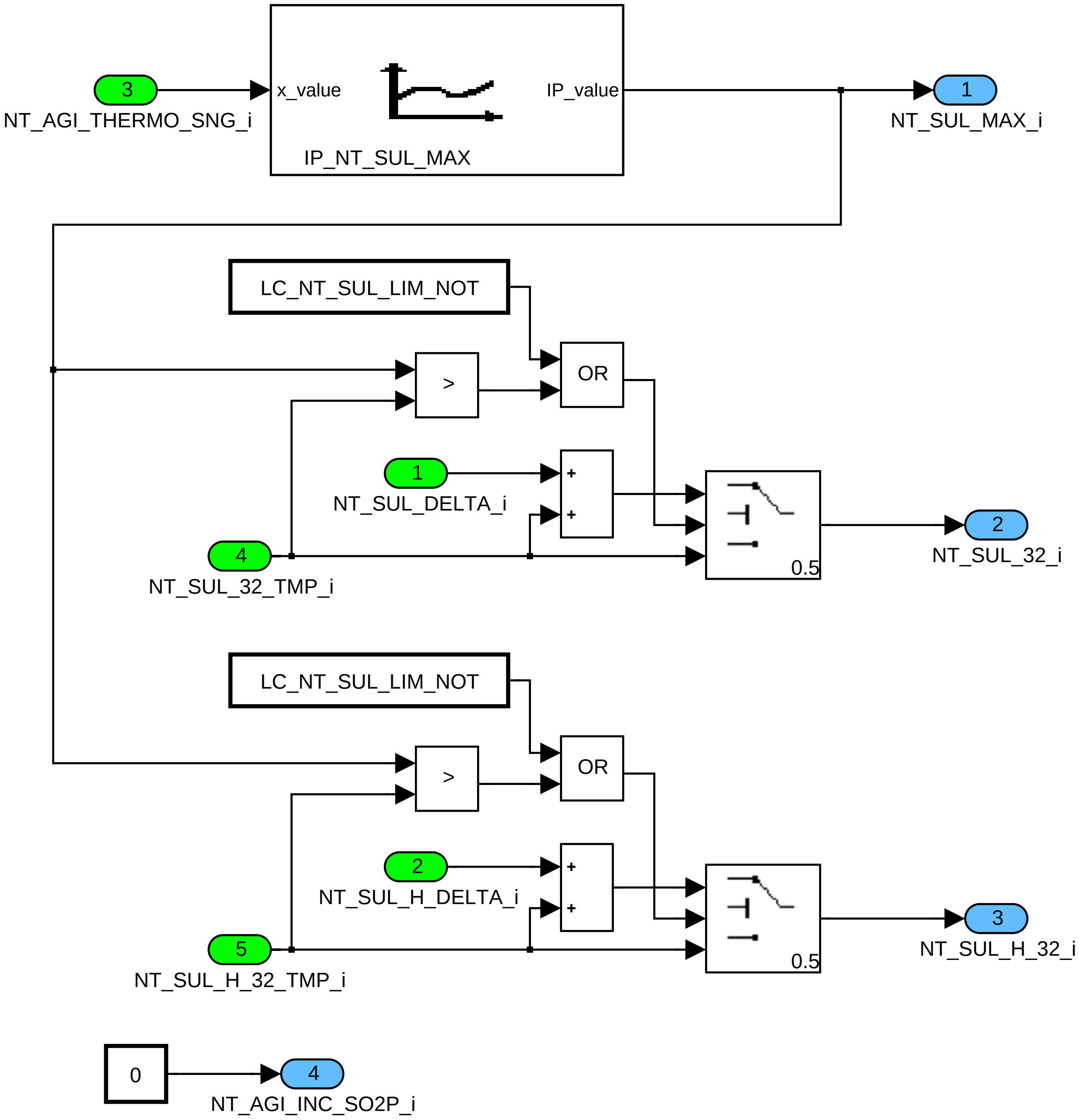


Figure 154 NOXM_MODUL704G/ operate/ CBK_MNG/ MAIN/ SUL/ CLC/ CLC3

Sulphur release phase post after delay time

The decrement of sulfur during release phase will be calculated by maps. The downward integration is limited to zero.

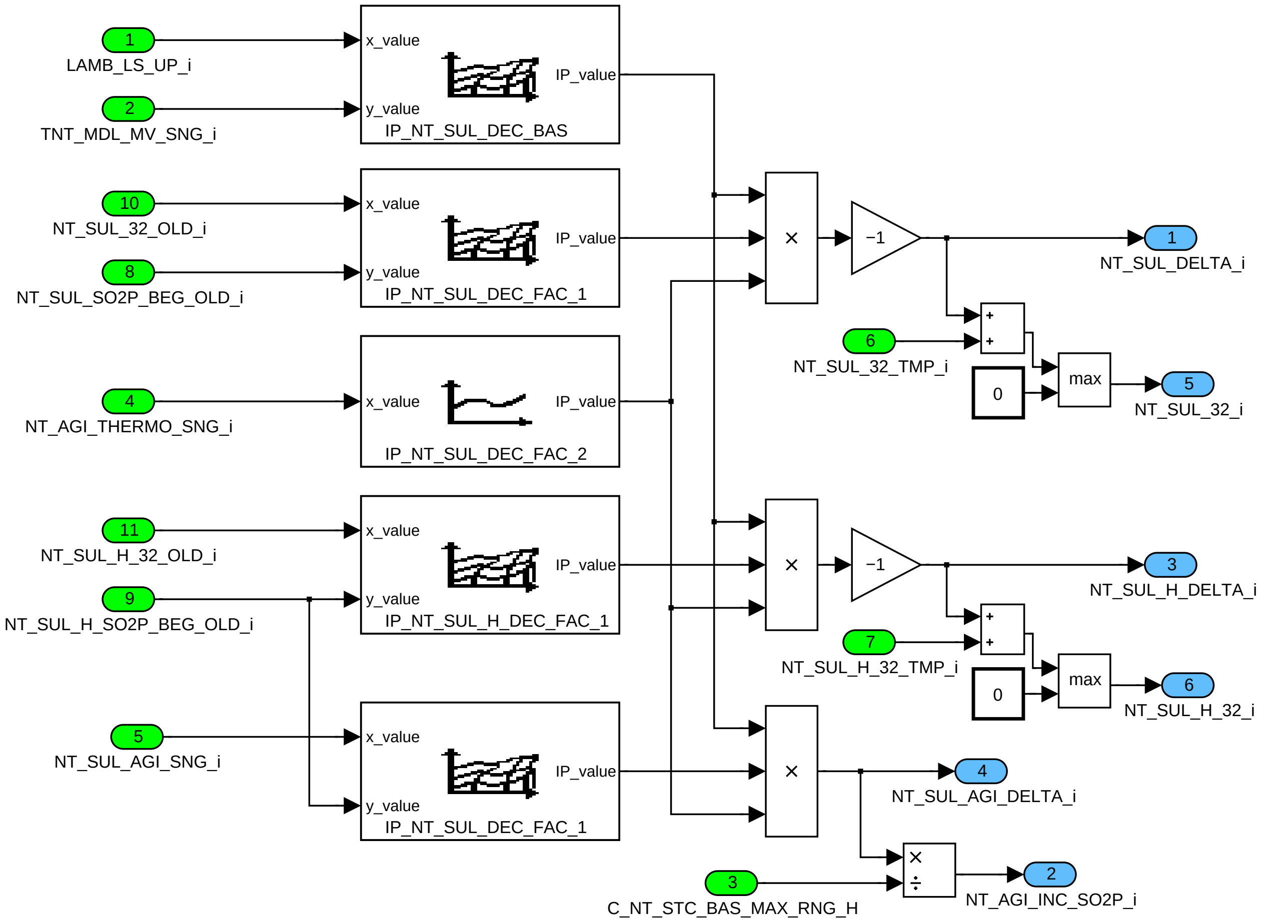


Figure 155 NOXM_MODUL704G/ operate/ CBK_MNG/ MAIN/ SU_RLS_POST_DLY/ CLC

Sulphur release phase before delay time

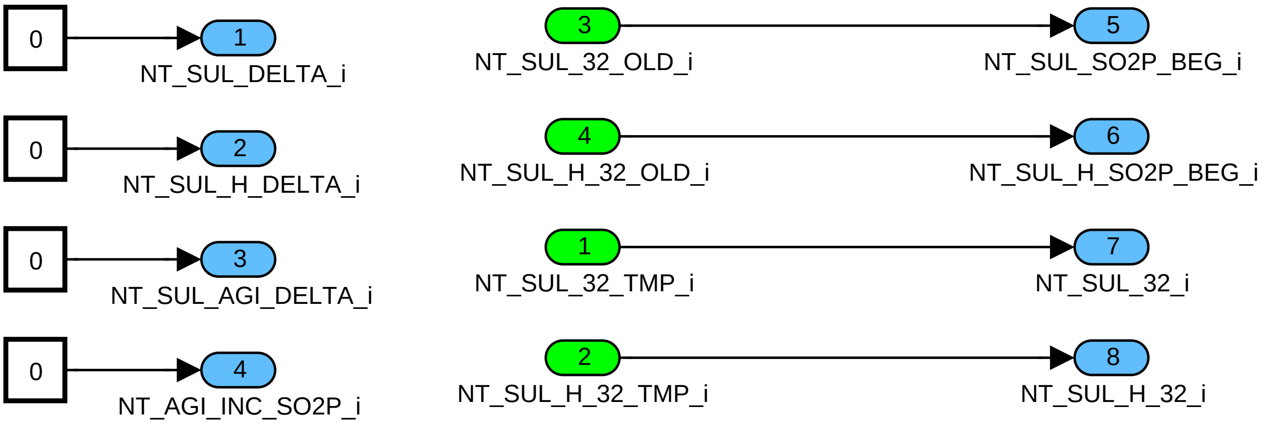


Figure 156 NOXM_MODUL704G/ operate/ CBK_MNG/ MAIN/ SU_RLS_PRE_DLY/ CLC

Update of external adjustment values

If flag LV_SUL_EXT_ADJ was set by action ACTION_NOXM_WriteSulfurExtAdj, then the external values will be used and the flag will be reset.

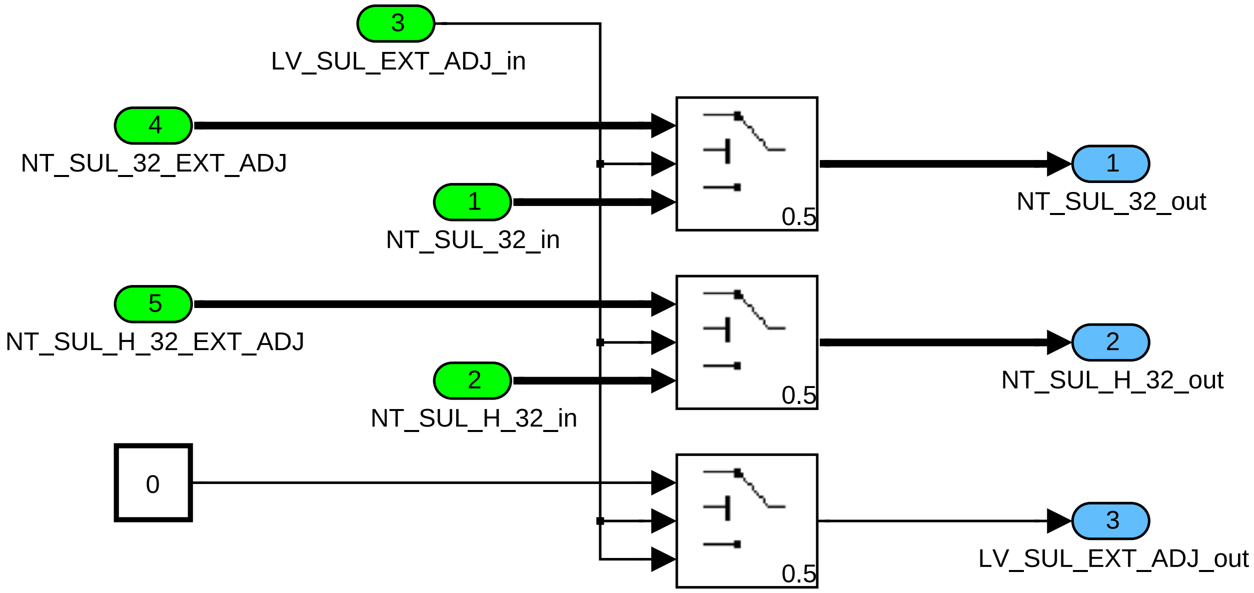


Figure 157 NOXM_MODUL704G/ operate/ AFTER_MAIN/ CLC1

Reset of sulfur values in case of new NOx traps

If flag LV_SUL_EXT_RES was set by action ACTION_NOXM_CleanNTAdaptSul, then the sulfur values will be set to zero and the flag will be reset.

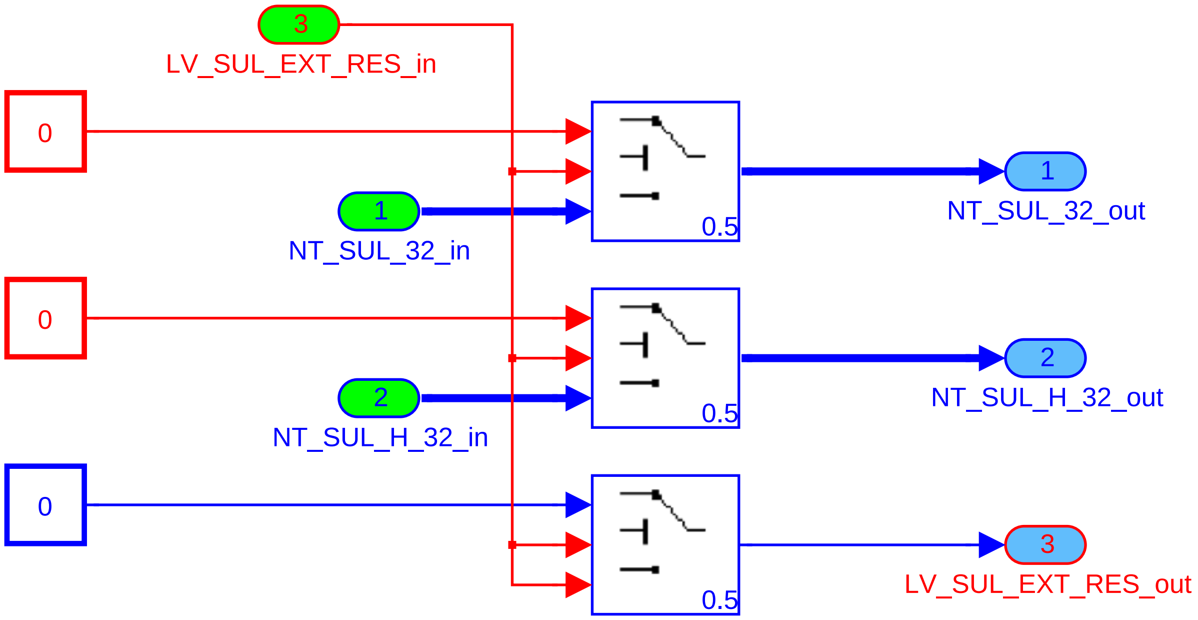


Figure 158 NOXM_MODUL704G/ operate/ AFTER_MAIN/ CLC2

Maximum calculation

After the MAIN part, which is calculated for each bank, the old value of the input NT_SUL_UPD and the maximum of the internal 32-bit values of all banks will be calculated as 16-bit value.

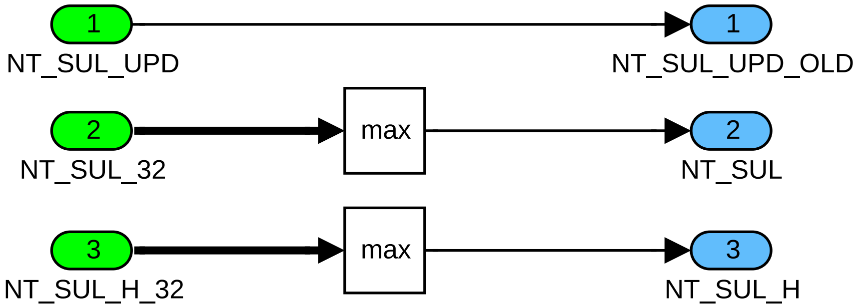


Figure 159 NOXM_MODUL704G/ operate/ AFTER_MAIN/ CLC3

53.14.1.4 Handling of non volatile memory for NT_SUL_32 and NT_SUL_H_32

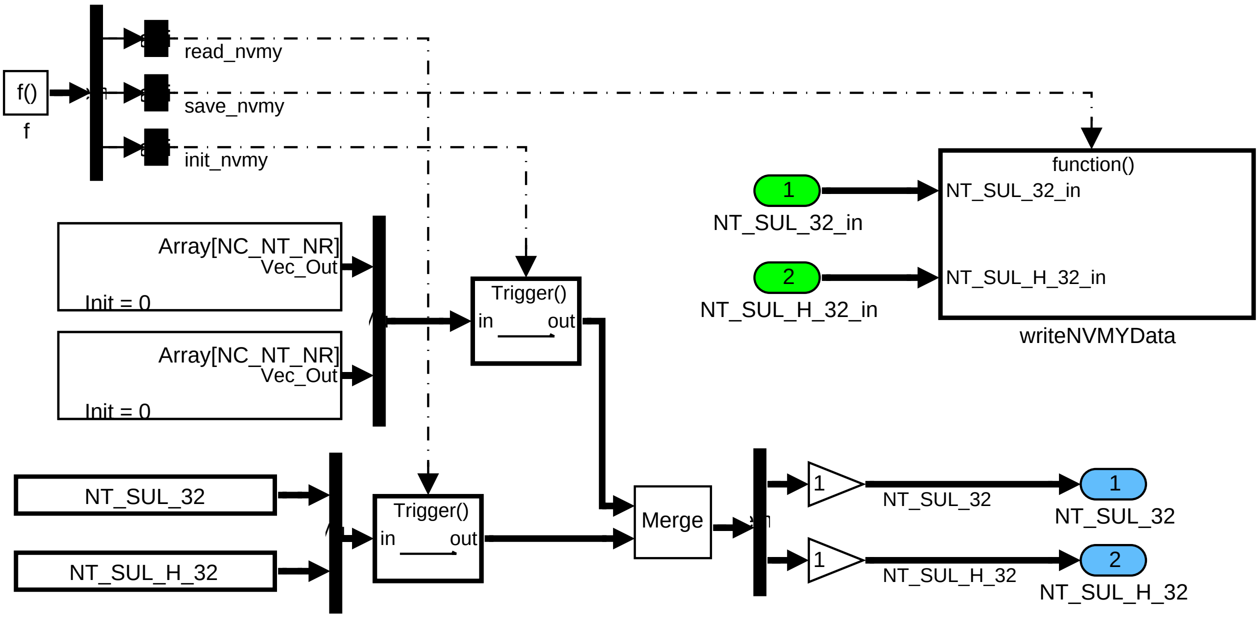


Figure 160 NOXM_MODUL704G/ NVMYdata

53.14.1.5 SUBFUNCTION: reset

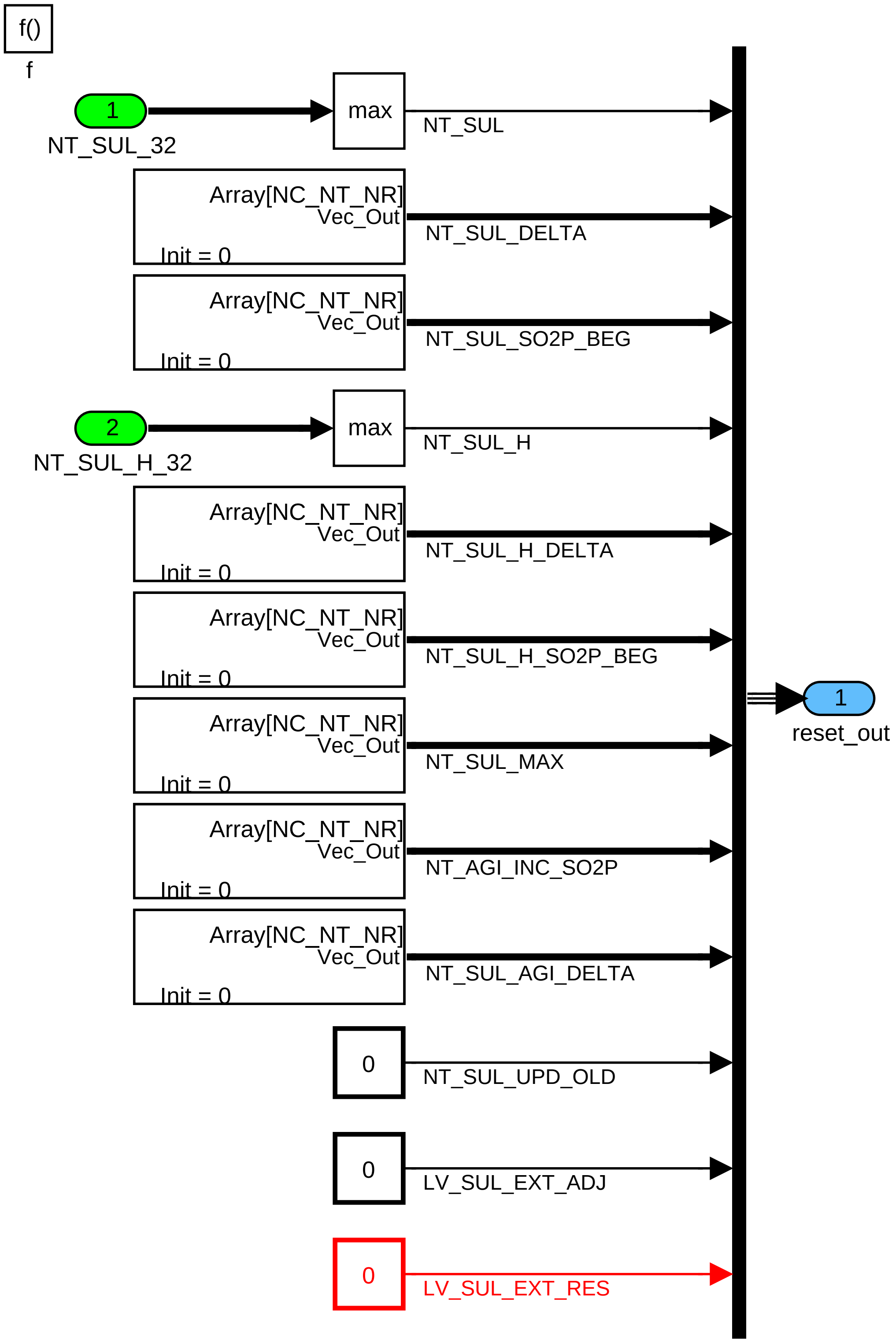


Figure 161 NOXM_MODUL704G/ reset

53.15 NOx Catalyst Desulfation

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
CTR_SO2P_PLS	V	0... FFH	0... 255	1	[-]
Counter for alternating desulfation cycles before activation of fast desulfation					
LAMB_SO2P [NC_CBK_EX_NR]	O/V	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Lambda setpoint for desulfation					
LV_NT_SO2P_INH_IS	V	0... 1H	0... 1	1	[-]
Inhibition of desulfation at idle speed					
LV_SO2P_ACT	O/V	0... 1H	0... 1	1	[-]
NOx trap desulfation active					
LV_SO2P_ACT_LAMB [NC_CBK_EX_NR]	O/V	0... 1H	0... 1	1	[-]
NOx trap desulfation lambda condition fulfilled					
LV_SO2P_ACT_TNT	O/V	0... 1H	0... 1	1	[-]
NOx trap desulfation temperature condition fulfilled					
LV_SO2P_AFR_CORD	O/V	0... 1H	0... 1	1	[-]
Rich operation for fast desulfation request					
LV_SO2P_AFR_CTR	V	0... 1H	0... 1	1	[-]
Active if any exhaust bank is rich, used for lean-rich edge detection					
LV_SO2P_FAST_REQ	O/V	0... 1H	0... 1	1	[-]
Fast desulfation request					
LV_SO2P_LAMB_PULS	O/V	0... 1H	0... 1	1	[-]
Lambda alternation request					
LV_SO2P_REQ	O/V	0... 1H	0... 1	1	[-]
General desulfation request					
T_SO2P_DLY	O/V	0... FFFFH	0... 6553.5	0.1	[s]
Delay time counter before sulfur release					
T_SO2P_NOT	O/V	0... FFFFH	0... 6553.5	0.1	[s]
Time counter for non-desulfation atmosphere					
VLS_REF_SO2P [NC_CBK_EX_NR]	V	0... 3FFH	0... 4.9951171875	4.88281e-3	[V]
Reference signal value for desulfation atmosphere detection					

Input Data:

CL_MMV	IP_VLS_REF_LAMB_LS_UP	IP_VLS_REF_LAMB_NS	IP_VLS_REF_VLS_NS
LAMB_LS_UP [NC_CBK_EX_NR]	LAMB_NS [NC_NOX_SENS_CONF]	LV_INH_NT_SO2P_REQ	LV_IS
LV_SO2P_AFR [NC_CBK_EX_NR]	LV_SO2P_REQ_1	LV_NT_SO2P_EXT_ADJ_ACT	LV_NT_SO2P_FAST_REQ_EXT
NC_CBK_EX_NR	NC_NOX_SENS_CONF	NT_AGI_SUL	NT_SUL
TNT_MDL_H	TNT_MDL_MV	VLS_DOWN [NC_CBK_EX_NR]	VLS_NS [NC_NOX_SENS_CONF]
VLS_NT_DOWN [NC_NOX_SENS_CONF]	VS		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_CL_MMV_SO2P_IS_MAX	1	0... FFFFH	0... 1.99996948242	30.5176e-6	[-]
Maximum canister load for NOx trap desulfation at idle speed					

C_LAMB_SO2P_PAS	1	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
Passive lambda value if the desulfation is not active					
C_NT_AGI_SO2P_MAX	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Maximal NOx trap aging factor					
C_NT_AGI_SUL_SO2P_FAST	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
NOx trap aging threshold for fast desulfation					
C_NT_AGI_SUL_SO2P_FAST_HYS	1	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
NOx trap aging hysteresis for fast desulfation					
C_NT_SUL_SO2P_MAX	1	0... FFFFH	0... 10485.6	0.16	[mg]
Maximal NOx trap sulphur load out of sulphur model					
C_STATE_SENS_VLS_DOWN_SO2P_ACT [NC_CBK_EX_NR]	1	0 1 2 3 4 5	LAMB_LS_UP VLS_DOWN VLS_NS LAMB_NS VLS_NT_DOWN MANUAL	-	[-]
Mode to determine the sensor used for detection of a desulfating atmosphere					
C_T_SO2P_NOT_MAX	1	0... FFFFH	0... 6553.5	0.1	[s]
Maximal time for desulfation interruption before desulfation timer reset					
C_TNT_MDL_SO2P_FAST	1	0... FFH	0... 1020	4	[°C]
NOx trap temperature threshold for switching to fast desulfation					
C_TNT_MDL_SO2P_FAST_HYS	1	0... FFH	0... 1020	4	[°C]
NOx trap temperature hysteresis for switching back to alternating desulfation					
C_TNT_MDL_SO2P_HYS	1	0... FFH	0... 1020	4	[°C]
NOx trap monolith temperature hysteresis for SO2 purge deactivation					
C_TNT_MDL_SO2P_MIN	1	0... FFH	0... 1020	4	[°C]
NOx trap monolith temperature threshold for SO2 purge possible					
C_TNT_MDL_SO2P_MIN_EXT	1	0... FFH	0... 1020	4	[°C]
NOx trap monolith temperature threshold for SO2 purge possible by external request					
C_VLS_REF_SO2P	1	0... 3FFH	0... 4.9951171875	4.88281e-3	[V]
Reference signal manual value					
C_VLS_SO2P_HYS	1	0... 3FFH	0... 4.9951171875	4.88281e-3	[V]
VLS hysteresis for SO2 purge active detection					
C_VLS_SO2P_MIN	1	0... 3FFH	0... 4.9951171875	4.88281e-3	[V]
Threshold for SO2 purge activation					
C_VS_SO2P_FAST_HYS	1	0... FFH	0... 255	1	[km/h]
Vehicle speed hysteresis for switching back to alternating desulfation					
IP_CTR_SO2P_PLS_AGI	4	0... FFH	0... 255	1	[-]
LDP_AGI_SUL_IP_CTR	4	0... FFFFH	0... 0.99998474121	15.2588e-6	[-]
Number of alternating desulfation cycles before fast desulfation (depending on sulfur aging)					
IP_CTR_SO2P_PLS_SUL	4	0... FFH	0... 255	1	[-]
LDP_SUL_IP_CTR	4	0... FFFFH	0... 10485.6	0.16	[mg]
Number of alternating desulfation cycles before fast desulfation (depending on sulfur load)					
IP_LAMB_SO2P	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDP_TNT_MDL_MV_IP_LAMB_SO2P	8	0... FFFFH	0... 1023.984375	0.015625	[°C]
Required lambda setpoint during desulfation					
IP_LAMB_SO2P_EXT	8	0... 7FFFH	0... 31.9990234375	976.563e-6	[-]
LDP_TNT_MDL_MV_IP_LAMB_SO2P_EXT	8	0... FFFFH	0... 1023.984375	0.015625	[°C]
Required lambda setpoint during desulfation by external request					

IP_VS_SO2P_FAST_THD	4	0... FFH	0... 255	1	[km/h]
LDP_TNT_MDL_MV_IP_VS_SO2P_FAST	4	0... FFH	0... 1020	4	[°C]
Vehicle speed threshold for switching to fast desulfation according to TNT_MDL_MV					
LC_SO2P_FAST_MAN	1	0... 1H	0... 1	1	[-]
Manual switch to fast desulfation mode					

General Information

Gasoline contains sulphur, which is chemically stored in the NOx catalyst. Due to this sulphur compounds, the NOx trap storage capacity and the NOx trap efficiency will decrease. The sulphates can be decomposed under high temperature with rich mixture. This catalyst state (high temperature, rich exhaust gas composition) is called NOx catalyst desulfation.
If a desulfation request is set, the function is able to request catalyst heating. When an enough high monolith-temperature is reached the desulfation is started by switching to a suitable lambda value. This desulfation lambda depends on the actual minimum monolith temperature and on the desulfation mode.
Two desulfation modes are implemented: a fast way with a single, rich lambda value or a way with a lambda alternating around lambda=1 and thereby with two main lambda values, a lower rich value and an upper lean value. This way permits low H2S emissions.
The decision, which desulfation process is chosen, is made dependent on the vehicle speed or a manual application.
It can't be guaranteed the setting of a lambda request is followed immediately by a change of lambda. For this reason, this function contents a detection of a desulfating atmosphere (that means a high temperature level and a rich exhaust gas composition.

Application Conditions

Initialization: RST
Recurrence: 100MS
Activation: always
Deactivation: never

Function description

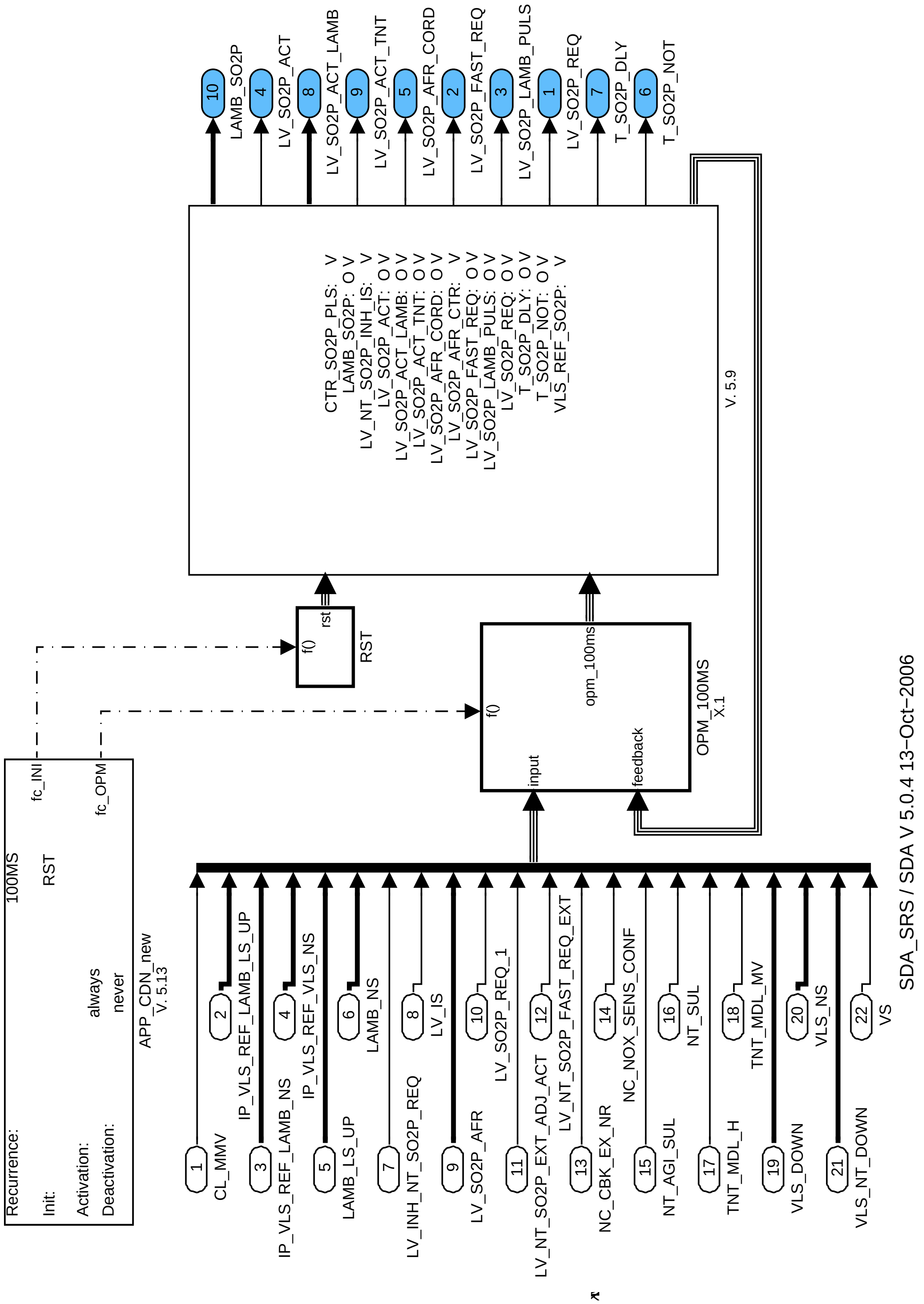


Figure 162:
Path: NOXM_REQGNNTD0

53.15.1 FORMULA SECTION

The bank specific and bank not specific calculations are done separately.
First the bank specific calculations are done (SO2P_ACT_MNG). This section contains detection of the desulfation atmosphere.
Bank not specific and bank diagonal calculations are included in subsystem REQ_CLC. In this section the request for desulfation is calculated.

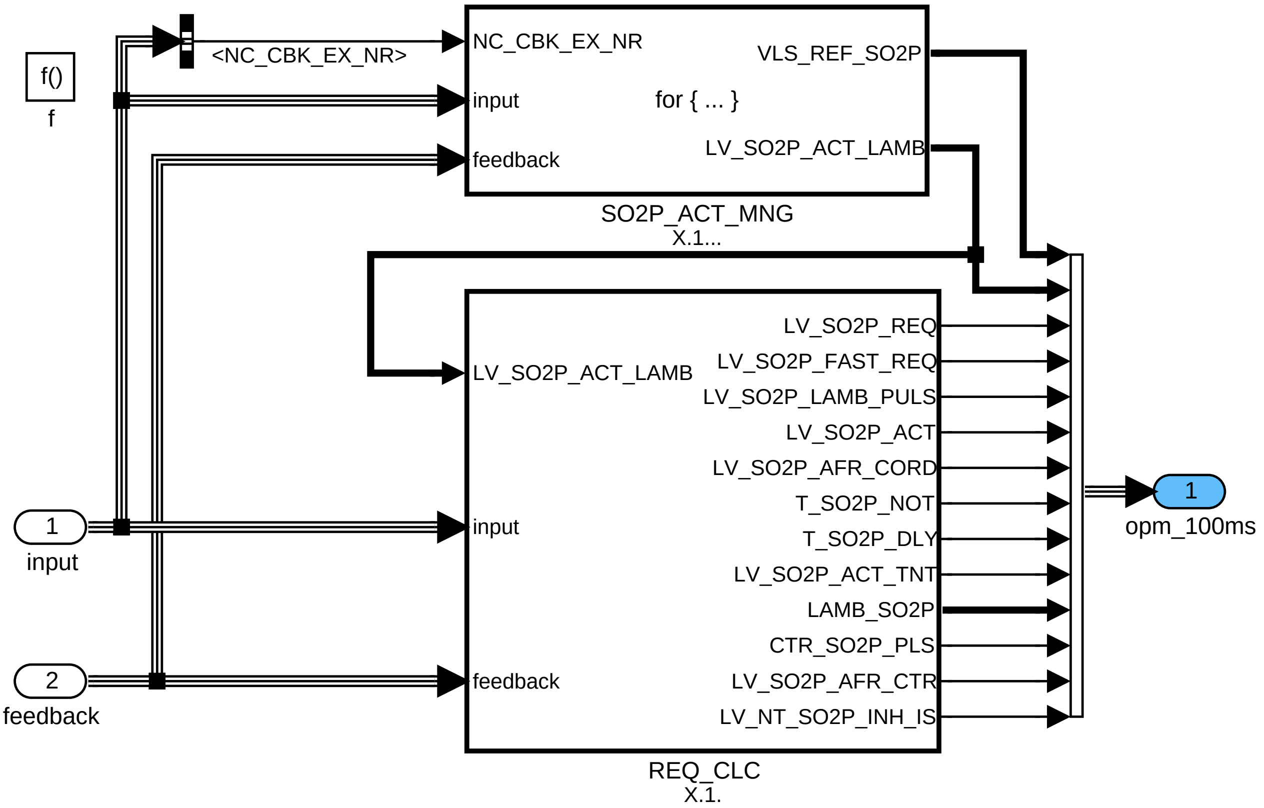


Figure 163:
Path: NOXM_REQGNNTD0/OPM_100MS

53.15.1.1 BANK SPECIFIC CALCULATIONS

53.15.1.1.1 Management of different exhaust gas configurations

Index "k" corresponds to number of NOx Sensors in the system. Supported is only configuration where number of NOx Sensors is less or equal to number of exhaust banks. If one NOx Sensor and two banks exist then for both banks shall be taken the same NOx Sensor voltage signal

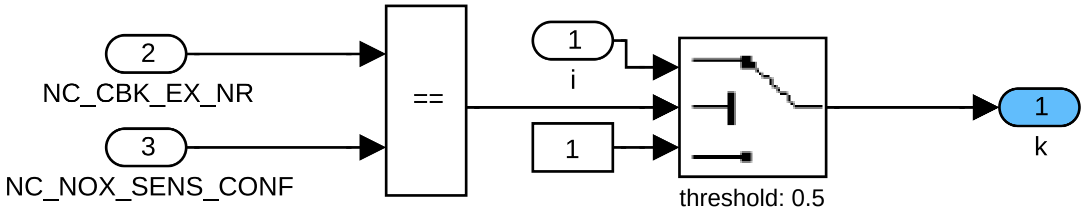


Figure 164:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/CLC_k

53.15.1.1.2 Detection of a desulfating or non-desulfating atmosphere

For the detection of the exhaust gas composition one of available signals is chosen and converted to binary voltage value of lambda if needed. After that the desulfation atmosphere can be detected using Lambda value.

53.15.1.1.2.1 Selection of sensor signal

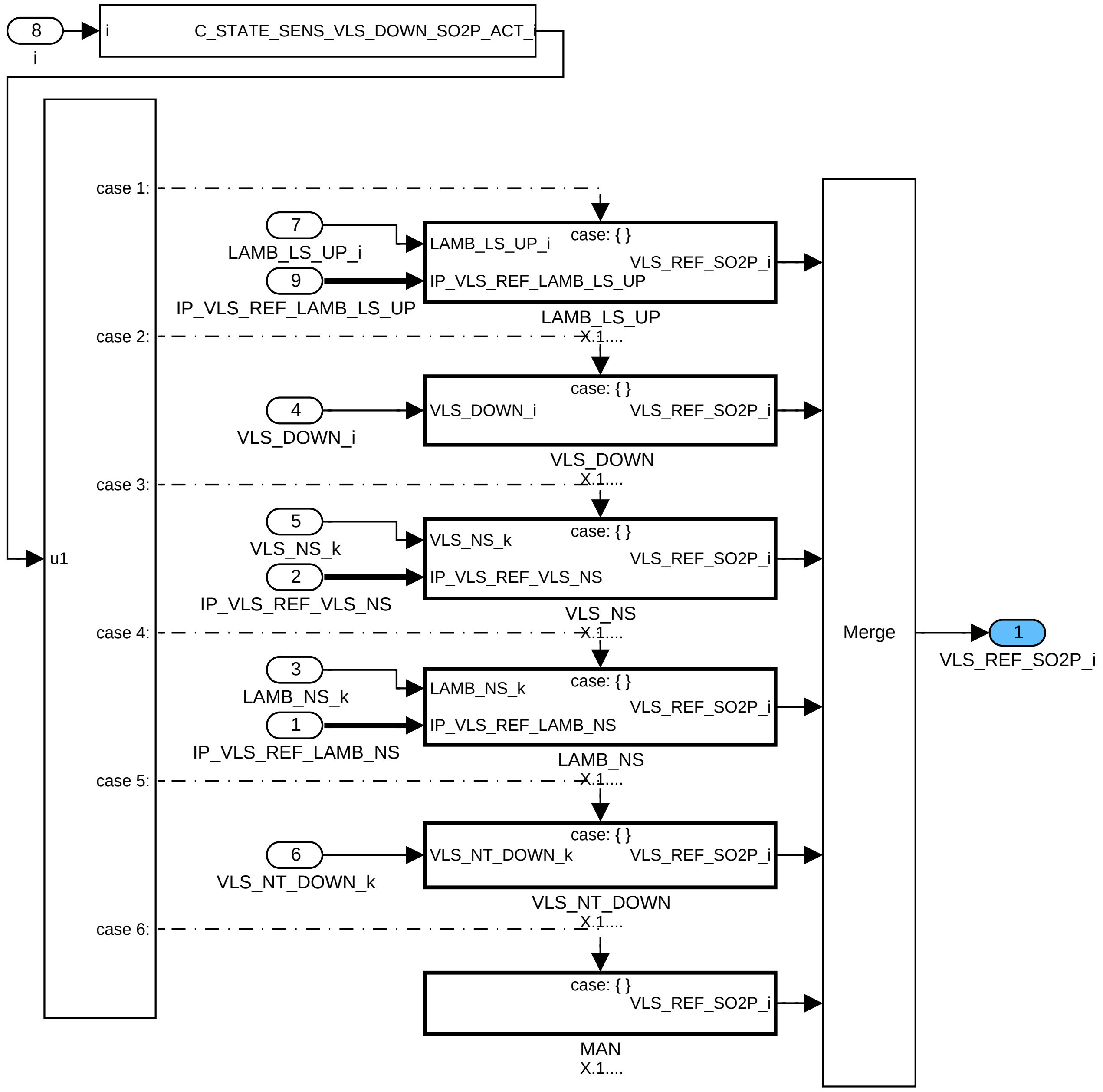


Figure 165:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/SO2P_ACT/CHK_SEL_STATE

53.15.1.1.2.1.1 Linear lambda before pre-catalyst

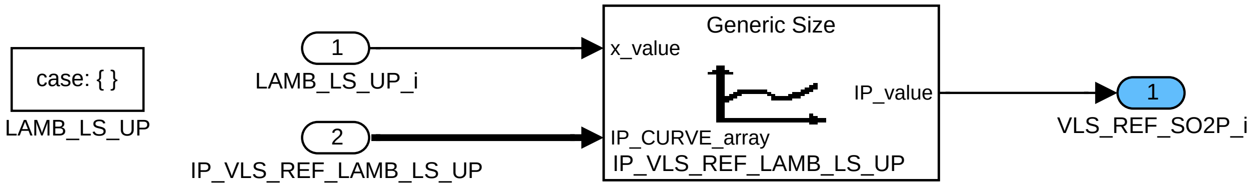


Figure 166:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/SO2P_ACT/CHK_SEL_STATE/LAMB_LS_UP

53.15.1.1.2.1.2 Binary lambda sensor after pre-catalyst

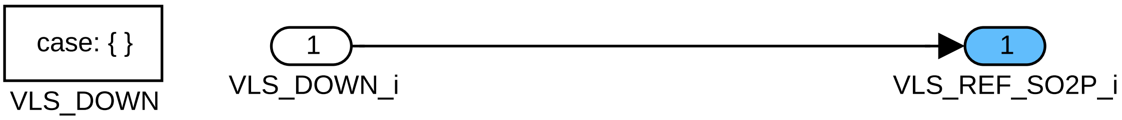


Figure 167:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/SO2P_ACT/CHK_SEL_STATE/VLS_DOWN

53.15.1.1.2.1.3 Binary lambda signal of NOx sensor

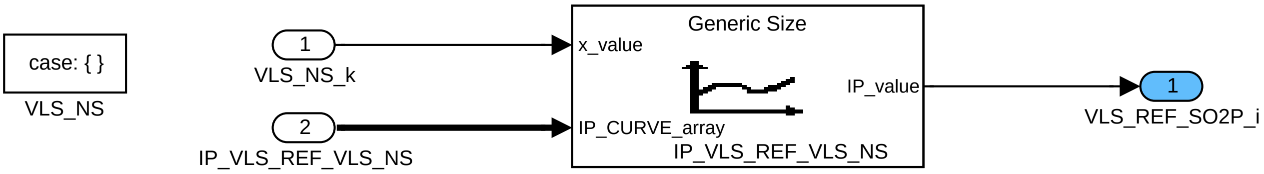


Figure 168:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/SO2P_ACT/CHK_SEL_STATE/VLS_NS

53.15.1.1.2.1.4 Linear lambda signal of NOx sensor

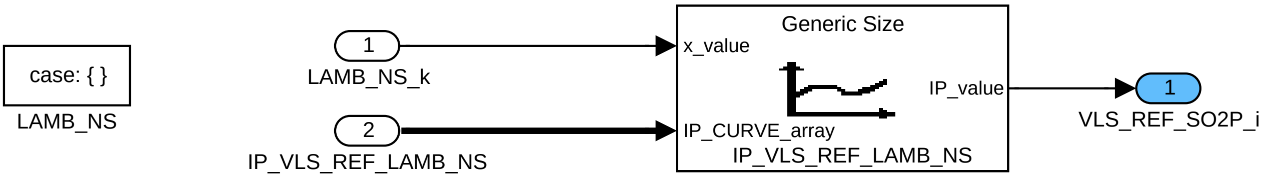


Figure 169:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/SO2P_ACT/CHK_SEL_STATE/LAMB_NS

53.15.1.1.2.1.5 Binary lambda sensor after NOx trap

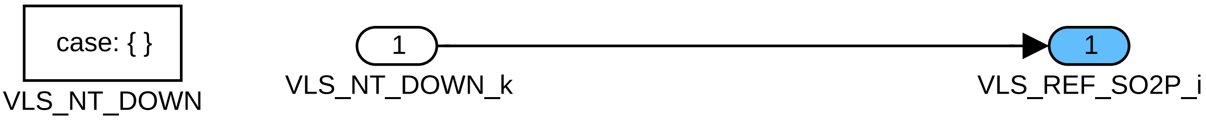


Figure 170:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/SO2P_ACT/CHK_SEL_STATE/VLS_NT_DOWN

53.15.1.1.2.1.6 Manuel value

Figure 171:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/SO2P_ACT/CHK_SEL_STATE/MAN

53.15.1.1.2.2 Detection of suitable lambda for desulfation

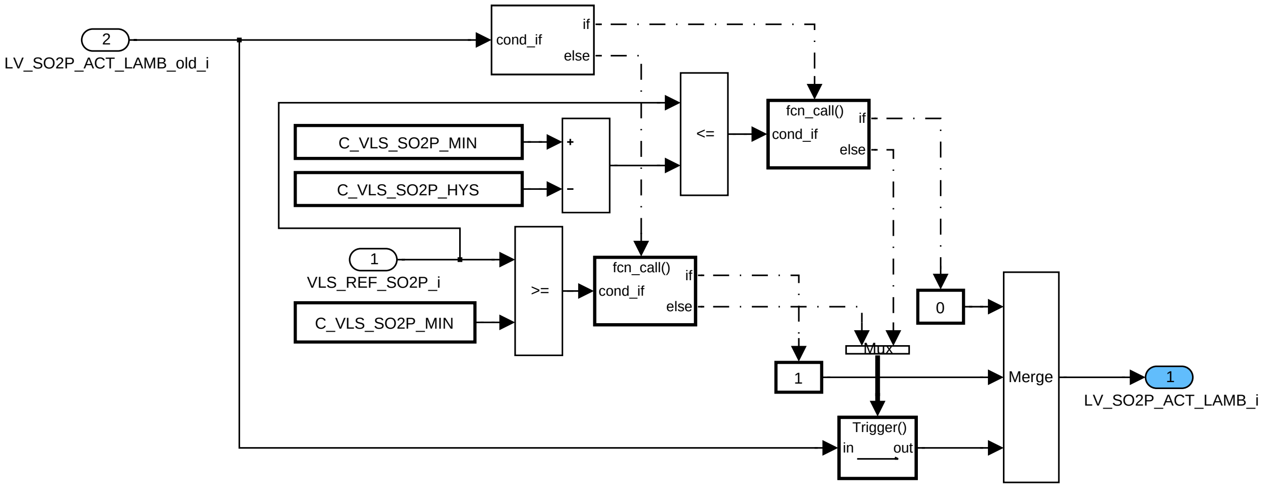


Figure 172:
Path: NOXM_REQGNNTD0/OPM_100MS/SO2P_ACT_MNG/SO2P_ACT/CHK_COMB_CDN

53.15.1.2 BANK NOT SPECIFIC AND BANK DIAGONAL CALCULATIONS

53.15.1.2.1 Selection of minimum temperature for desulfation

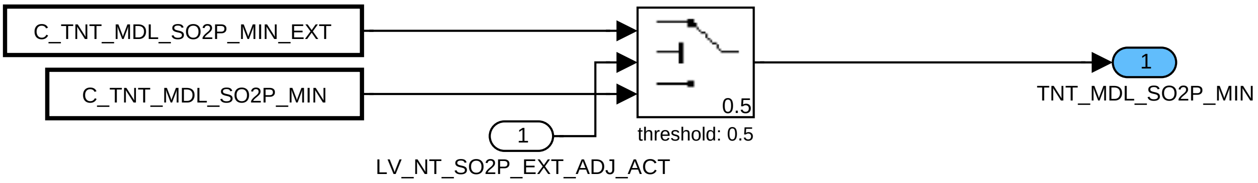


Figure 173:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/TNT_MDL_SO2P_MIN

53.15.1.2.2 Inhibition of desulfation at idle speed

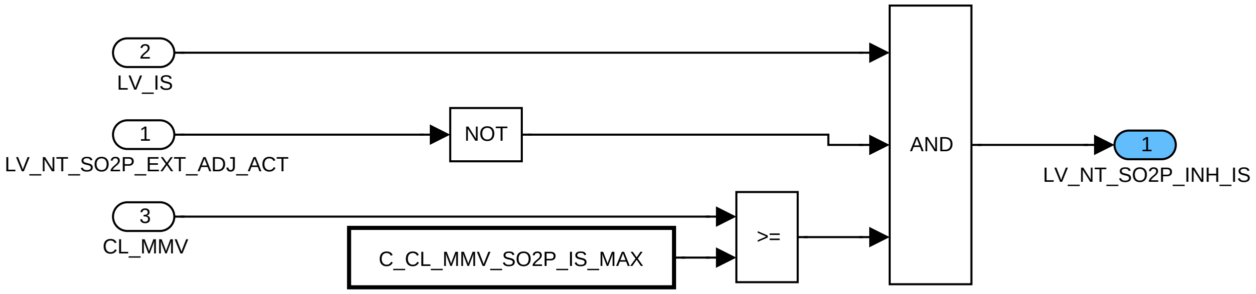


Figure 174:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_INH_IS

53.15.1.2.3 Desulfation request start condition

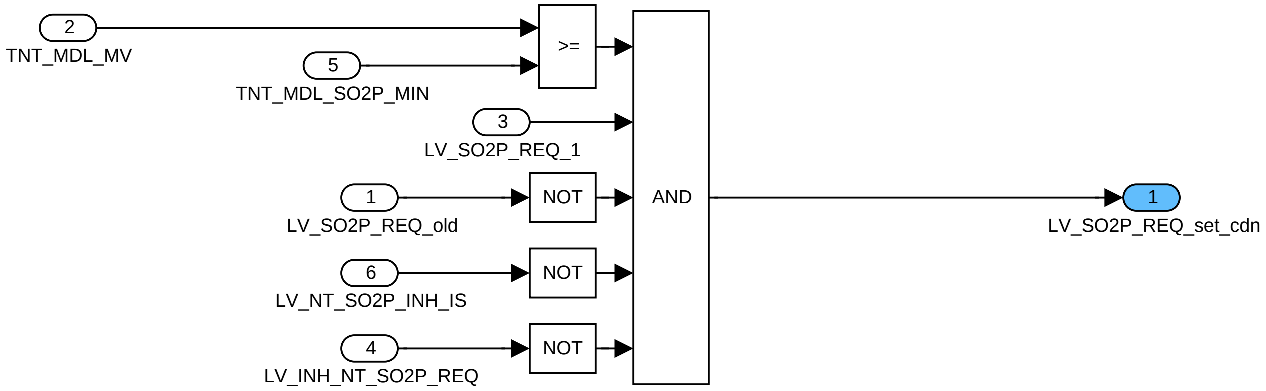


Figure 175:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_REQ_CDN

53.15.1.2.4 Calculation of the desulfation request

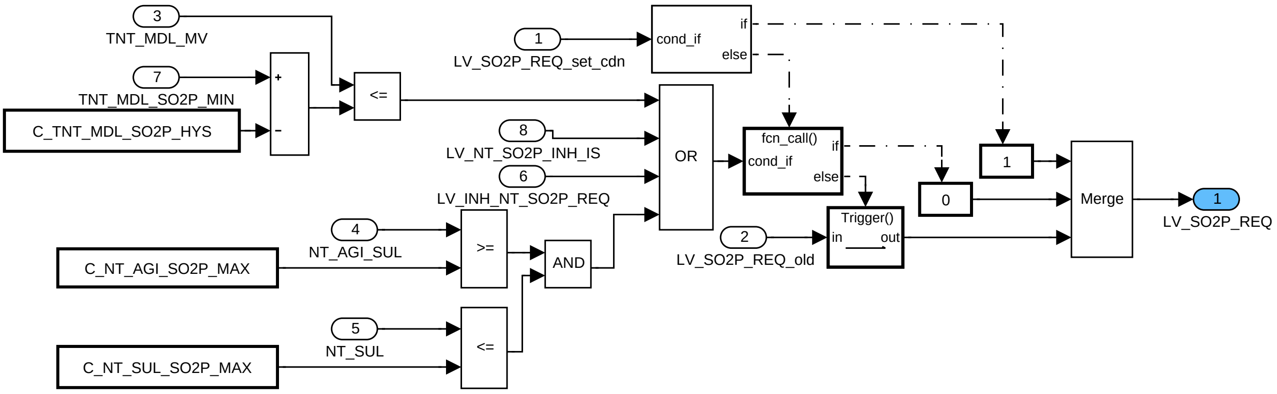


Figure 176:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_REQ

53.15.1.2.5 Determination of desulfation mode

Alternating or fast desulfation is used depending on several conditions. Additionally, at the start of a fast desulfation some alternating desulfation cycles may be calibrated.

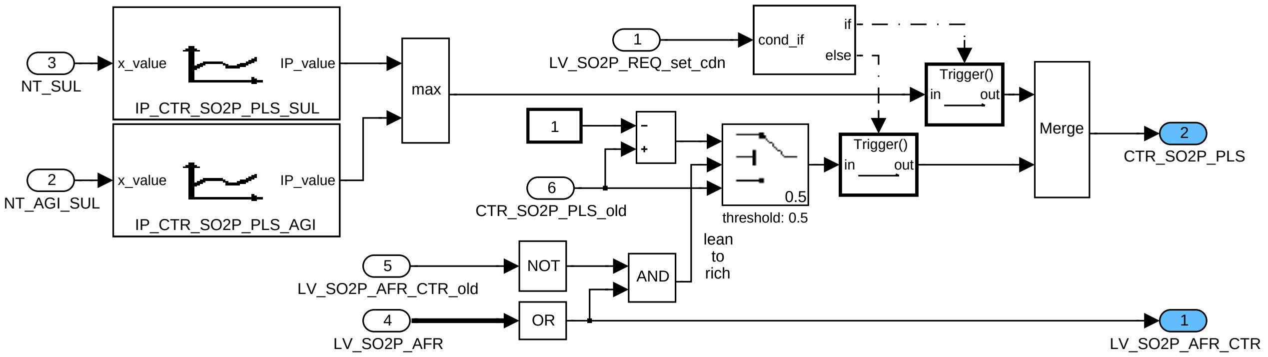


Figure 177:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/CTR_SO2P

53.15.1.2.6 Fast or alternating desulfation

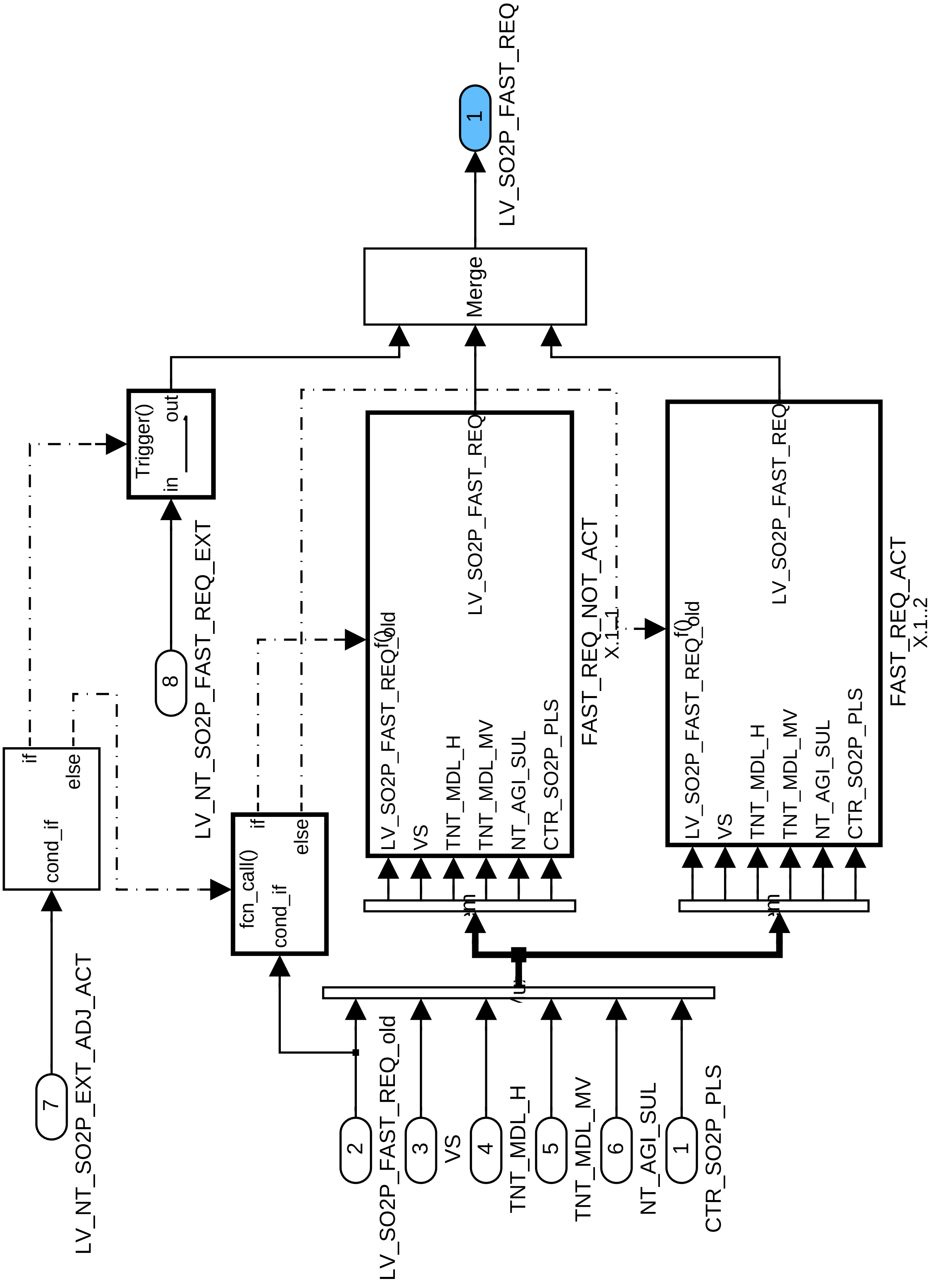


Figure 178:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_FAST_REQ

53.15.1.2.6.1 FAST_REQ_NOT_ACT

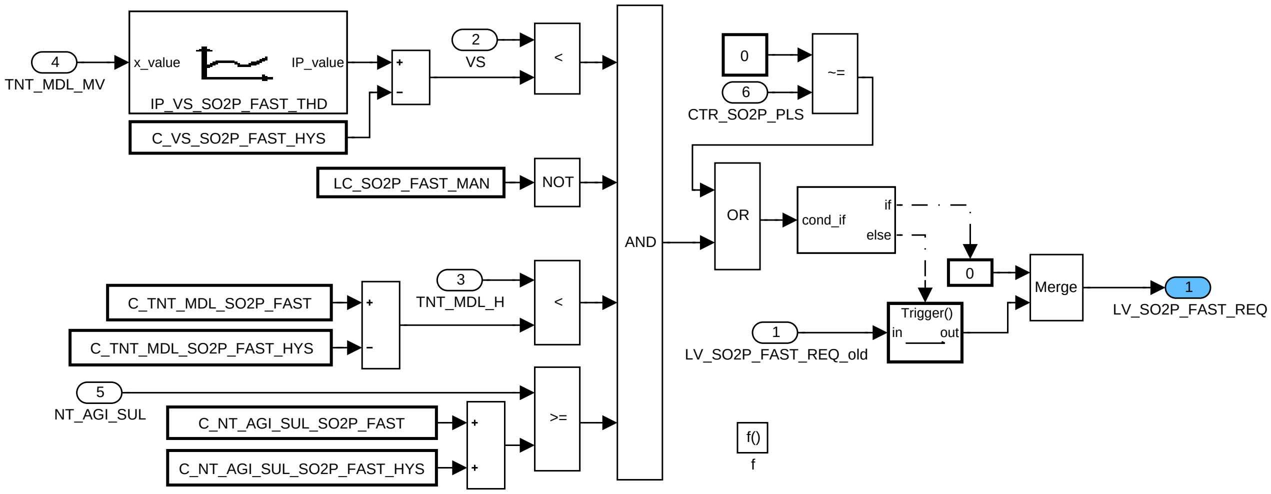


Figure 179:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_FAST_REQ/FAST_REQ_NOT_ACT

53.15.1.2.6.2 FAST_REQ_ACT

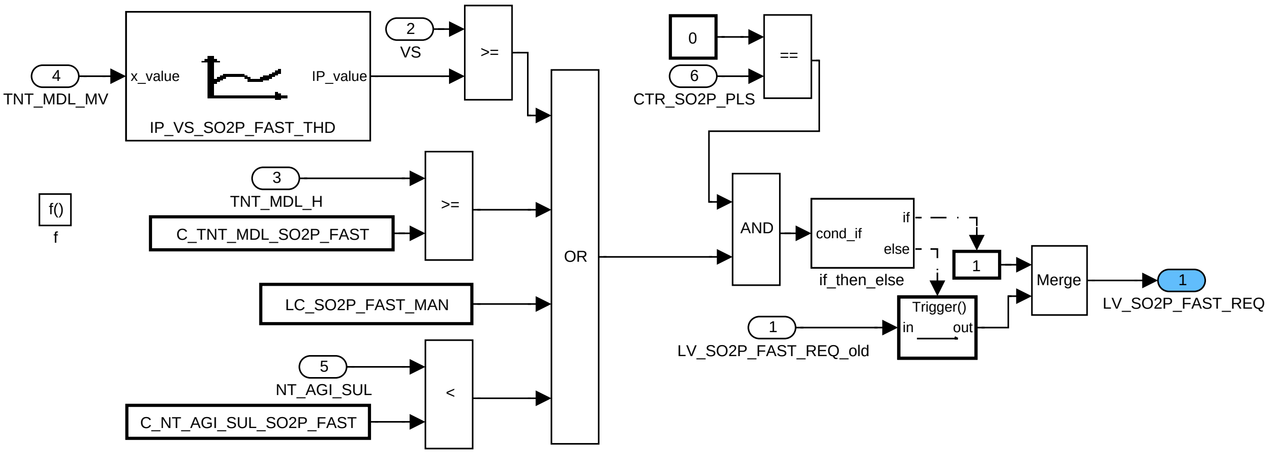


Figure 180:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_FAST_REQ/FAST_REQ_ACT

53.15.1.2.7 Determination of the flags for the desulfation lambda alternation and catalyst heating manager

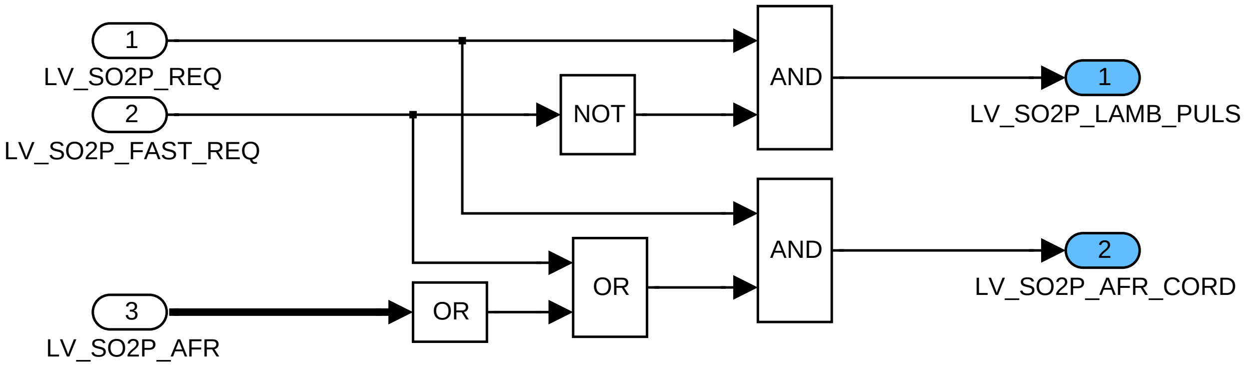


Figure 181:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_LAMB_PULS

53.15.1.2.8 Calculation of activation flag for desulfation regarding modeled NOx trap temperature

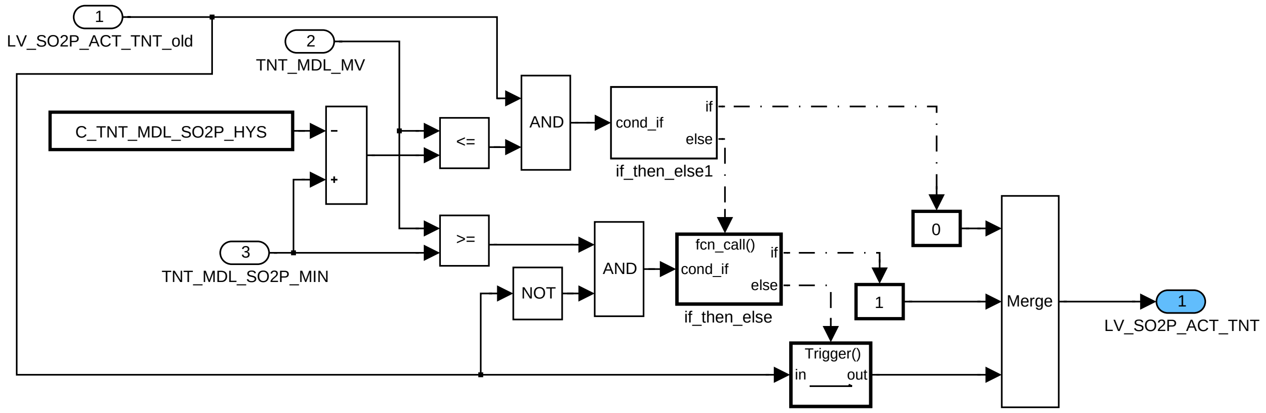


Figure 182:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_ACT_TNT

53.15.1.2.9 Set of desulfation activation indicator and calculation of non-desulfation time

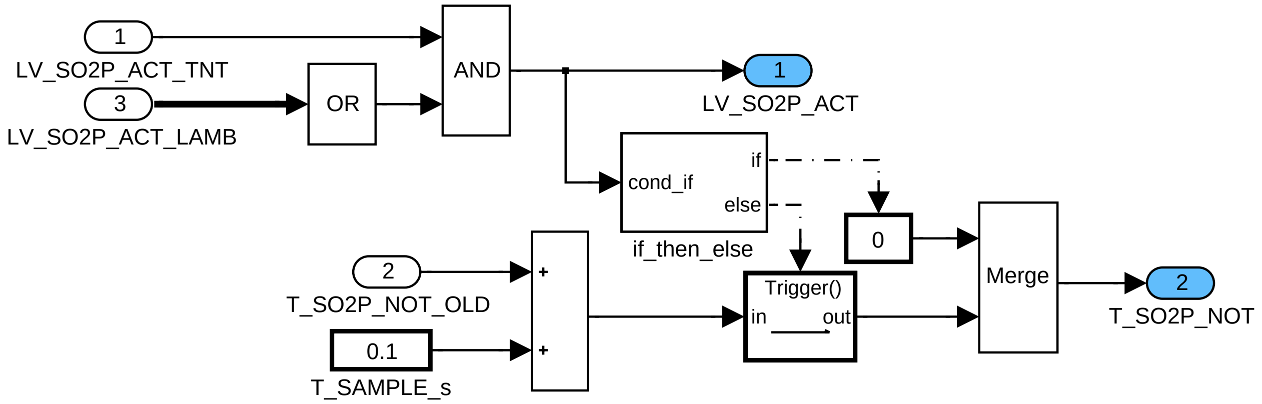


Figure 183:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_ACT

53.15.1.2.10 Calculation of delay time before the sulfur release is active

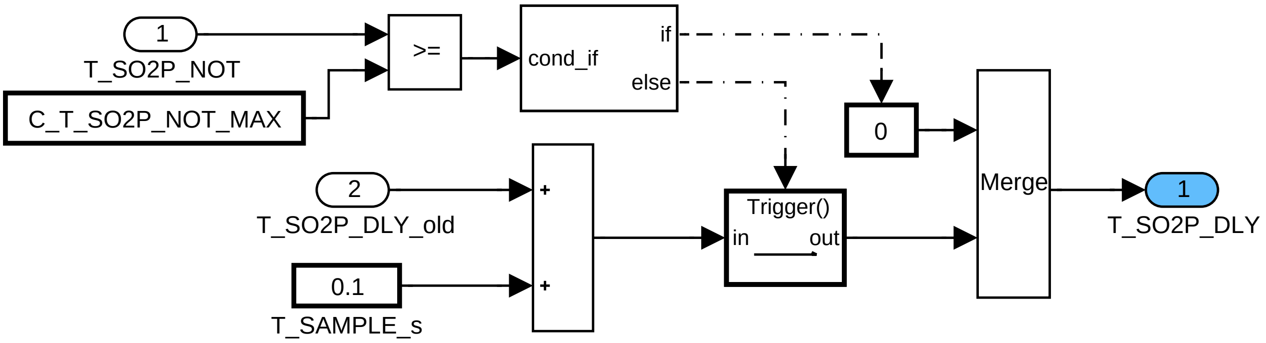


Figure 184:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/SO2P_DLY

53.15.1.2.11 Lambda setpoint for fast NOx Catalyst desulfation

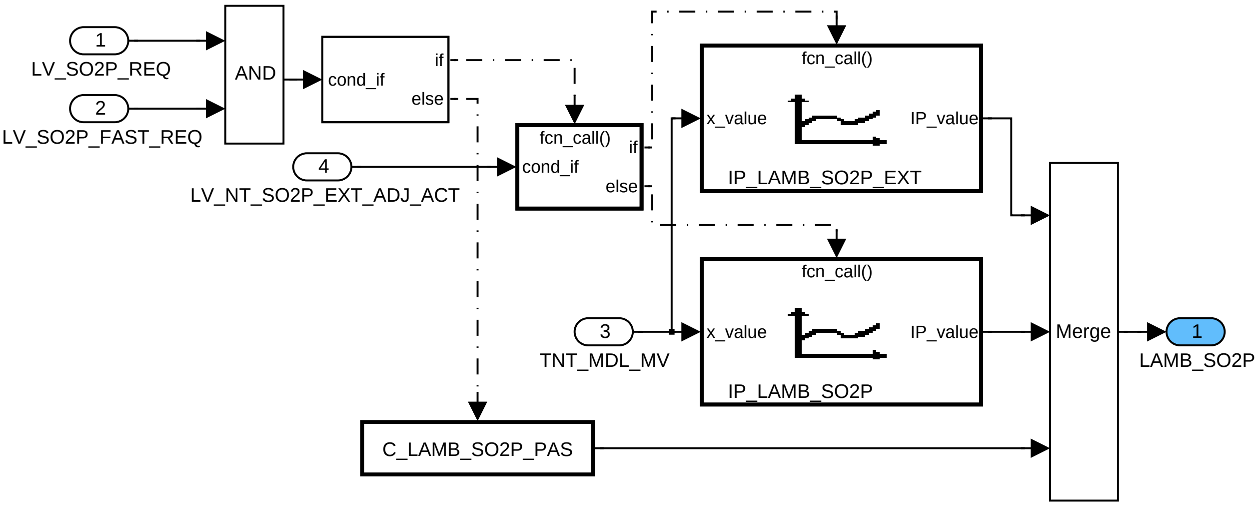


Figure 185:
Path: NOXM_REQGNNTD0/OPM_100MS/REQ_CLC/CLC_LAMB_SO2P

53.16 Application Incidences for NOx catalyst desulfation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_KWP_SO2P	O/V	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H	NOT_START ST_INH PAR_NOT_PLAUS WAIT_REL UNDEF ACT END_WOUT_RESULT ABORTED END_WOUT_ERR END_WITH_ERR	1	[-]
State variable for desulfation by external request					
LV_NT_SO2P_EXT_ADJ_ACT	O/V	0...1H	0...1	1	[-]
NOx trap desulfation started by external adjustment is active					
STATE_NT_SO2P_EXT_ADJ_ACT	O/V	0H 1H 2H 3H	PASSIVE SPEED-UP HEATING-UP DESULFATION	1	[-]
Additional status information for desulfation by external request					
LV_NT_SO2P_FAST_REQ_EXT	O/V	0...1H	0...1	1	[-]
Fast desulfation request controlled by external request					
T_NT_SO2P_EXT_ADJ_ACT	O/V	0...FFFFH	0...6553.5	0.1	[s]
Timer for desulfation by external request					
LV_NT_SO2P_EXT_ADJ_ENA	V	0...1H	0...1	1	[-]
Condition for enabling desulfation by external request					
LV_NT_SO2P_EXT_ADJ_ST	V	0...1H	0...1	1	[-]
Start condition for desulfation by external request					
LV_NT_SO2P_EXT_ADJ_STOP	V	0...1H	0...1	1	[-]
Stop condition for desulfation by external request					
LV_CH_SO2P_WOUT_LIM	O/V/S	0...1H	0...1	1	[-]
Catalyst heating for desulfation without any limitation due to fuel consumption					
LV_INH_NT_SO2P_REQ	O/V	0...1H	0...1	1	[-]
Deactivation of desulfation request					

Input data:

GEAR	LV_CLU_SWI	LV_ERR_TEG_PCAT_DOWN	LV_NT_SO2P_EXT_ADJ_REQ_ST
LV_NT_SO2P_EXT_ADJ_REQ_NOT_STOP	LV_SO2P_REQ_2	N_32	NT_SUL
PV_AV	STATE_GEAR_REV_AT_AMT	STATE_NT_SO2P_EXT_ADJ	TNT_MDL_MV
VS			

Export actions:

ACTION_NOXM_StartCatHeatDesu()
Start catalyst heating for desulfation without any limitation due to fuel consumption
ACTION_NOXM_StopCatHeatDesu()
Stop catalyst heating for desulfation without any limitation due to fuel consumption

FUNCTION DESCRIPTION:

Description for ACTION_NOXM_StartCatHeatDesu

ACTION_NOXM_StartCatHeatDesu()					
Start catalyst heating for desulfation without any limitation due to fuel consumption					
Parameter	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**

```
If      LV_SO2P_REQ_2 = 1              % catalyst heating already requested
Then    LV_CH_SO2P_WOUT_LIM = 1        % fuel consumption not limited for CH
End If
```

Description for ACTION_NOXM_StopCatHeatDesu

ACTION_NOXM_StopCatHeatDesu()					
Stop catalyst heating for desulfation without any limitation due to fuel consumption					
Parameter	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**

```
LV_CH_SO2P_WOUT_LIM = 0                % fuel consumption limited for CH
```

Application conditions:

This module controls the desulfation that is forced by an external request.

Initialization : reset all values from output table except LV_CH_SO2P_WOUT_LIM to zero

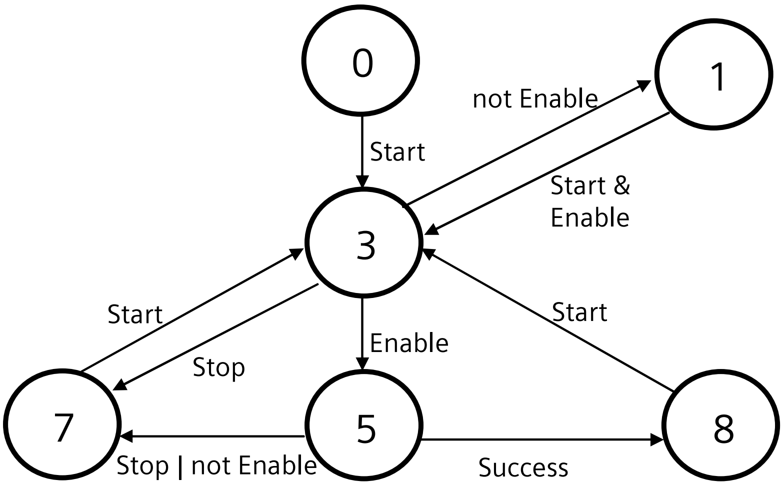
Recurrence : **100** ms

Formula section:

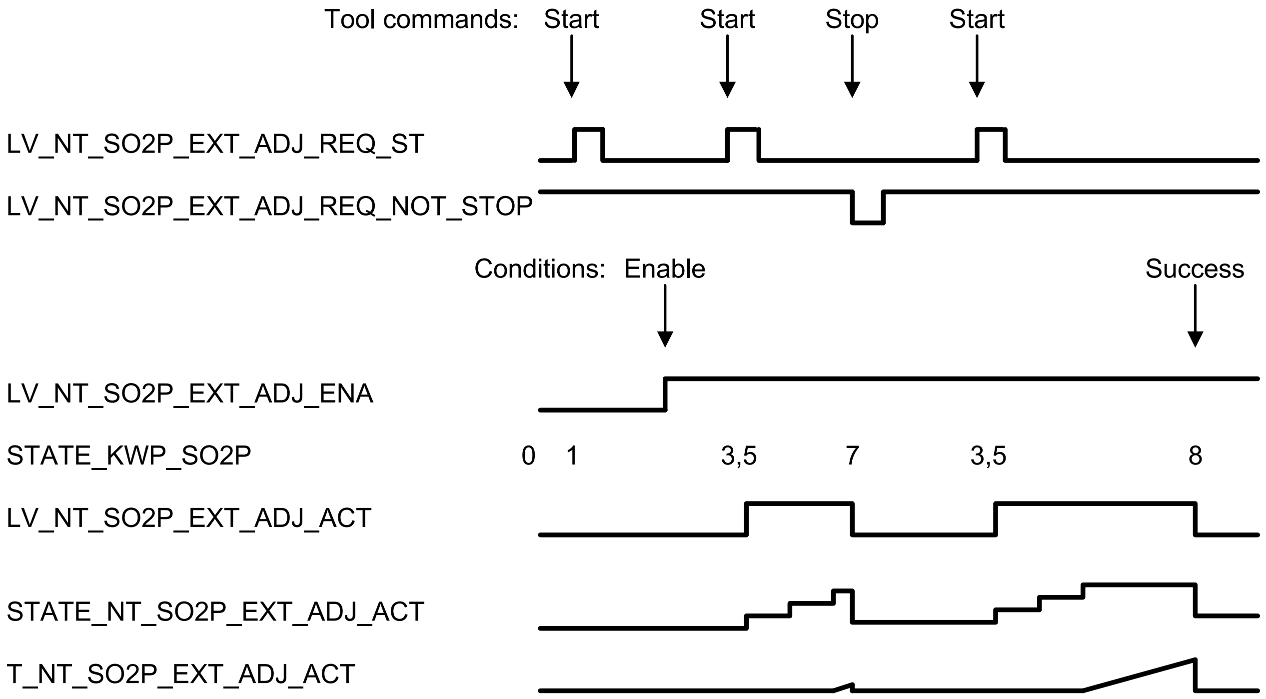
Catalyst heating without limitations due to fuel consumptions

```
If      LV_SO2P_REQ_2 = 0              % desulfation successful finished
Then    LV_CH_SO2P_WOUT_LIM = 0        % fuel consumption limited for CH
End If
```

State machine for desulfation by scan tool



Diagram



```
LV_NT_SO2P_EXT_ADJ_ENA =   N_32 >= C_N_IS_SO2P_MIN          AND        % Enable
                           GEAR = 0                         AND
                           STATE_GEAR_REV_AT_AMT = 0        AND
                           LV_CLU_SWI = 0                   AND
                           VS <= C_VS_SO2P_EXT_MAX          AND
                           PV_AV <= C_PV_SO2P_EXT_MAX


LV_NT_SO2P_EXT_ADJ_ST =
            LV_NT_SO2P_EXT_ADJ_REQ_ST = 1                   OR         % Start
            LC_NT_SO2P_EXT_MAN = 0→1

LV_NT_SO2P_EXT_ADJ_STOP =
            LV_NT_SO2P_EXT_ADJ_REQ_NOT_STOP = 0             OR         % Stop
            LC_NT_SO2P_EXT_MAN = 1→0

case selection on STATE_KWP_SO2P
 STATE_KWP_SO2P = 0:
      If        LV_NT_SO2P_EXT_ADJ_ST = 1                          % Start
      Then      STATE_KWP_SO2P = 3
      End If

 STATE_KWP_SO2P = 1:
      If        LV_NT_SO2P_EXT_ADJ_ST = 1         AND              % Start
                LV_NT_SO2P_EXT_ADJ_ENA = 1                         % Enable
      Then      STATE_KWP_SO2P = 3
      End If

 STATE_KWP_SO2P = 3:
      If        LV_NT_SO2P_EXT_ADJ_STOP = 1                        % Stop
      Then      STATE_KWP_SO2P = 7
      Else If   LV_NT_SO2P_EXT_ADJ_ENA = 1                         % Enable
      Then      STATE_KWP_SO2P = 5
      Else      STATE_KWP_SO2P = 1
      End If

 STATE_KWP_SO2P = 5:
      If        LV_NT_SO2P_EXT_ADJ_STOP = 1       OR               % Stop
                LV_NT_SO2P_EXT_ADJ_ENA = 0                         % not Enable
      Then      STATE_KWP_SO2P = 7

                                                                   % Success
      Else If   T_NT_SO2P_EXT_ADJ_ACT > C_T_SO2P_EXT_MAX           OR
                (T_NT_SO2P_EXT_ADJ_ACT > C_T_SO2P_EXT_MIN          AND
```

```
                NT_SUL <= C_NT_SUL_SO2P_EXT_MIN)
        Then    STATE_KWP_SO2P = 8
        End If

STATE_KWP_SO2P = 7:
        If      LV_NT_SO2P_EXT_ADJ_ST = 1                       % Start
        Then    STATE_KWP_SO2P = 3
        End If

STATE_KWP_SO2P = 8:
        If      LV_NT_SO2P_EXT_ADJ_ST = 1                       % Start
        Then    STATE_KWP_SO2P = 3
        End If

default:
        STATE_KWP_SO2P = 0
end case selection
```

Control of idle speed reference, catalyst heating and desulfation

```
If      STATE_KWP_SO2P = 5                                      % ACT
Then    LV_NT_SO2P_EXT_ADJ_ACT = 1
Else    LV_NT_SO2P_EXT_ADJ_ACT = 0
End If
```

Desulfation status

```
If      STATE_KWP_SO2P = 5                                      % ACT
Then    If      N_32 < C_N_SO2P_EXT_MIN_DISP
        Then    STATE_NT_SO2P_EXT_ADJ_ACT = 1                   % SPEED-UP
        Else If TNT_MDL_MV < C_TNT_SO2P_EXT_MIN_DISP
        Then    STATE_NT_SO2P_EXT_ADJ_ACT = 2                   % HEATING-UP
        Else    STATE_NT_SO2P_EXT_ADJ_ACT = 3                   % DESULFATION
        End If
Else    STATE_NT_SO2P_EXT_ADJ_ACT = 0                           % PASSIVE
End If
```

Desulfation timer

```
If      STATE_KWP_SO2P = 5                                      % ACT
Then    If      STATE_NT_SO2P_EXT_ADJ_ACT = 3                   % DESULFATION
        Then    increment T_NT_SO2P_EXT_ADJ_ACT
        End If
Else    T_NT_SO2P_EXT_ADJ_ACT = 0
End If
```

Control of fast or alternating desulfation at external request

```
case selection on C_STATE_INH_FAST_SO2P_EXT
        C_STATE_INH_FAST_SO2P_EXT = 0:                      % AUTO
                if      STATE_NT_SO2P_EXT_ADJ = 1           % ALTERNATING
                then    LV_NT_SO2P_FAST_REQ_EXT = 0         % ALTERNATING
                else    LV_NT_SO2P_FAST_REQ_EXT = 1         % FAST
                endif

        C_STATE_INH_FAST_SO2P_EXT = 1:                      % DISABLE
                LV_NT_SO2P_FAST_REQ_EXT = 0                 % ALTERNATING

        C_STATE_INH_FAST_SO2P_EXT = 2:                      % ENABLE
                LV_NT_SO2P_FAST_REQ_EXT = 1                 % FAST
end case selection
```

Inhibition mode of desulfation request

```
case selection on C_STATE_INH_NT_SO2P_REQ
        C_STATE_INH_NT_SO2P_REQ = 0 ('AUTO'):
                if      LV_ERR_TEG_PCAT_DOWN = 1
                then    LV_INH_NT_SO2P_REQ = 1
                else    LV_INH_NT_SO2P_REQ = 0
                endif

        C_STATE_INH_NT_SO2P_REQ = 1 ('DISABLE'):
                LV_INH_NT_SO2P_REQ = 1

        C_STATE_INH_NT_SO2P_REQ = 2 ('ENABLE'):
                LV_INH_NT_SO2P_REQ = 0
end case selection
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_INH_FAST_SO2P_EXT	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode for fast or alternating desulfation at external request					
LC_NT_SO2P_EXT_MAN	1	0...1H	0...1	1	[-]
Manual start and stop of external requested desulfation					
C_VS_SO2P_EXT_MAX	1	0...FFH	0...255	1	[km/h]
Vehicle speed threshold for disabling external request for desulfation					
C_PV_SO2P_EXT_MAX	1	0...FFH	0...99.60937	0.390625	[%]

Accelerator pedal threshold for disabling external request for desulfation					
C_T_SO2P_EXT_MAX	1	0...FFFFH	0...6553.5	0.1	[s]
Maximum time for desulfation by external request					
C_T_SO2P_EXT_MIN	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time for desulfation by external request					
C_NT_SUL_SO2P_EXT_MIN	1	0...FFFFH	0...10485.6	0.16	[mg]
NOx trap sulphur loading limit for end of desulfation by external request					
C_N_SO2P_EXT_MIN_DISP	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold for change display status from Speed-up to Heating-up					
C_TNT_SO2P_EXT_MIN_DISP	1	0...FFFFH	0...1023.98437	0.015625	[°C]
NOx trap temperature threshold for change display status from Speed-up to Heating-up					
C_N_IS_SO2P_MIN	1	0...FFH	0...8160	32	[rpm]
Threshold for detecting running engine					
C_STATE_INH_NT_SO2P_REQ	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of desulfation request					

53.17 NOx catalyst desulfation lambda stimulation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LAMB_PULS_SO2P[NC_CBK_EX_NR]	O/V	80...7FH	-0.125 ... 0.12402344	9.76563E-4	-
Delta lambda to pass to the lambda stimulation coordination routine					
LV_SO2P_AFR[NC_CBK_EX_NR]	O/V	0...1H	0...1	1	-
Rich phase of alternation					
CTR_VLS_DOWN_AFL_CYC[NC_CBK_EX_NR]	V	0...FFFFH	0...6.5535E+4	1	-
Denotes number of AF cycles since lambda controller last activated at LS downstream position					
T_AFL_DOWN_LOCK[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Timer indicating duration of LV_AFL lock at LS downstream position					
T_LAM_DLY_SO2P[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Total LAM -P-jump delay time for transition due to desulfation lambda alternation					
T_VLS_DOWN_CYC_AFL[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Lean mixture cycle time at LS downstream position					
T_VLS_DOWN_CYC_AFR[NC_CBK_EX_NR]	V	0...FFFFH	0...1.3107E+3	0.02	s
Rich mixture cycle time at LS downstream position					
VLS_DOWN_REF[NC_CBK_EX_NR]	V	0...3FFH	0...4.99511719	0.00488281	V
Voltage signal observed for downstream AF cycle evaluation					
LAMB_PULS_SO2P_H_RES[NC_CBK_EX_NR]	V	8000...7FFFH	-0.125 ... 0.12499619	3.8147E-6	-
Delta lambda with high resolution					
LV_VLS_DOWN_AFL[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Monitoring sensor has detected "lean"					
LV_VLS_DOWN_TRA[NC_CBK_EX_NR]	V	0...1H	0...1	1	-
Transition of Air-Fuel mixture by downstream sensor					

Input data:

LV_SO2P_LAMB_PULS	LAMB_NS[NC_NOX_SENS_CONF]	LAMB_SP[NC_CBK_EX_NR]	LV_FL
LV_INH_LAMB_PULS_SO2P	LV_IS	LV_PL	LV_PU
LV_PUC	LV_SCC[NC_CBK_EX_NR]	LV_ST_END	LV_S_ACT
MAF	NC_NOX_SENS_CONF	N_32	VLS_DOWN[NC_CBK_EX_NR]
VLS_NS[NC_NOX_SENS_CONF]	VLS_NT_DOWN[NC_NOX_SENS_CONF]	NC_CBK_EX_NR	LAMB_LS_UP[NC_CBK_EX_NR]

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DELTA_LAMB_AFL_SO2P	1	8000...7FFFH	-0.125 ... 0.12499619	3.8147E-6	-
Delta lambda value for lean phase of desulfation					
C_DELTA_LAMB_AFR_SO2P	1	8000...7FFFH	-0.125 ... 0.12499619	3.8147E-6	-
Delta lambda value for rich phase of desulfation					
C_DELTA_LAMB_SP_SO2P	1	0...800H	0...2	9.76563E-4	-
Permissible deviation of the lambda-controller setpoint for desultation with lambda alternation					

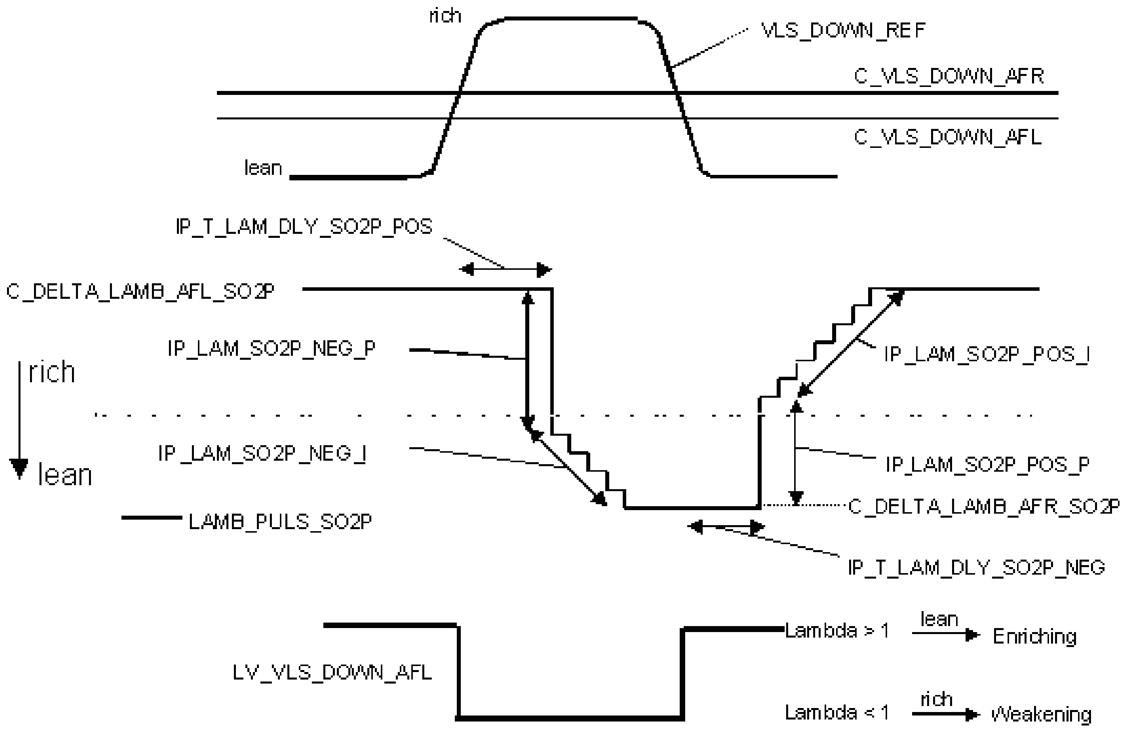
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LAMB_SP_SO2P	1	0...7FFFH	0...31.9990234	9.76563E-4	-
Basis lambda setpoint for calculation of activation condition for desulfation					
C_STATE_VLS_SENS_REF[NC_CBK_EX_NR]	1	0H 1H 2H 3H 4H 5H	LAMB_LS_UP VLS_DOWN VLS_NS LAMB_NS VLS_NT_DOWN MANUAL	1	-
Mode to determine sensor signal in the first sensor position					
C_T_VLS_DOWN_CYC_AFL_MAX	1	0...FFFFH	0...1.3107E+3	0.02	s
Maximum lean mixture cycle time at LS downstream position					
C_T_VLS_DOWN_CYC_AFR_MAX	1	0...FFFFH	0...1.3107E+3	0.02	s
Maximum rich mixture cycle time at LS downstream position					
C_VLS_DOWN_AFL	1	0...3FFH	0...4.995117	0.00488281	V
Threshold for detection of lean mixture at LS downstream position					
C_VLS_DOWN_AFR	1	0...3FFH	0...4.995117	0.00488281	V
Threshold for detection of rich mixture at LS downstream position					
C_VLS_REF_SO2P_LAMB_SP	1	0...3FFH	0...4.99511719	0.00488281	V
Reference signal manual value					
LC_T_AFL_DOWN_CTL	1	0...1H	0...1	1	-
Consider time thresholds for switching lean to rich or rich to lean					
IP_VLS_REF_LAMB_LS_UP	8	0...3FFH	0...4.99511719	0.00488281	V
LDP_LAMB_LS_UP_IP_VLS_REF	8	0...7FFFH	0...31.9990234	9.76563E-4	-
Reference map for upstream lambda to binary lambda value					
IP_VLS_REF_LAMB_NS	8	0...3FFH	0...4.99511719	0.00488281	V
LDP_LAMB_NS_IP_VLS_REF_LAMB_NS	8	0...7FFFH	0...31.9990234	9.76563E-4	-
Scaling map for sensor on the first position from NOx Sensor lambda signal to VLS signal					
IP_VLS_REF_VLS_NS	8	0...3FFH	0...4.99511719	0.00488281	V
LDP_VLS_NS_IP_VLS_REF_VLS_NS	8	0...578H	-200...1.2E+3	1	mV
Scaling map sensor from NOx Sensor binary signal to VLS					
IP_LAM_SO2P_NEG_I	8x8	0...7FH	0...0.12402344	9.76563E-4	-
LDPM_N_32_1_LAMB_PULS_SO2P	8	0...FFH	0...8.16E+3	32	rpm
LDPM_MAF_1_LAMB_PULS_SO2P	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Basic value of LAMB_PULS_SO2P integral component when lambda < 1					
IP_LAM_SO2P_NEG_P	8x8	0...7FH	0...0.12402344	9.76563E-4	-
LDPM_N_32_1_LAMB_PULS_SO2P	8	0...FFH	0...8.16E+3	32	rpm
LDPM_MAF_1_LAMB_PULS_SO2P	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Basic value of the proportional component of delta lambda during transition from lean to rich					
IP_LAM_SO2P_POS_I	8x8	0...7FH	0...0.12402344	9.76563E-4	-
LDPM_N_32_1_LAMB_PULS_SO2P	8	0...FFH	0...8.16E+3	32	rpm
LDPM_MAF_1_LAMB_PULS_SO2P	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Basic value of LAMB_PULS_SO2P integral component when lambda > 1					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_LAM_SO2P_POS_P	8x8	0...7FH	0...0.12402344	9.76563E-4	-
LDPM_N_32_1_LAMB_PULS_SO2P	8	0...FFH	0...8.16E+3	32	rpm
LDPM_MAF_1_LAMB_PULS_SO2P	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Basic value of the proportional component of delta lambda during transition from rich to lean					
IP_T_AFL_DOWN_LOCK_AFL	8x8	0...FFFFH	0...1.3107E+3	0.02	s
LDPM_N_32_1_LAMB_PULS_REF	8	0...FFH	0...8.16E+3	32	[rpm]
LDPM_MAF_2_LAMB_PULS_REF	8	0...FFFFH	0...1.389E+3	0.0211948	[mg/stk]
LV_AFL transition lock-out time after transition of AF mixture to lean at LS downstream position					
IP_T_AFL_DOWN_LOCK_AFR	8x8	0...FFFFH	0...1.3107E+3	0.02	s
LDPM_N_32_1_LAMB_PULS_REF	8	0...FFH	0...8.16E+3	32	[rpm]
LDPM_MAF_2_LAMB_PULS_REF	8	0...FFFFH	0...1.389E+3	0.0211948	[mg/stk]
LV_AFL transition lock-out time after transition of AF mixture to rich at LS downstream position					
IP_T_LAM_DLY_SO2P_NEG	8x8	0...FFFFH	0...1.3107E+3	0.02	s
LDPM_N_32_1_LAMB_PULS_SO2P	8	0...FFH	0...8.16E+3	32	rpm
LDPM_MAF_1_LAMB_PULS_SO2P	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Basic LAM -P-jump delay time from rich to lean transition					
IP_T_LAM_DLY_SO2P_POS	8x8	0...FFFFH	0...1.3107E+3	0.02	s
LDPM_N_32_1_LAMB_PULS_SO2P	8	0...FFH	0...8.16E+3	32	rpm
LDPM_MAF_1_LAMB_PULS_SO2P	8	0...FFFFH	0...1.389E+3	0.02119478	mg/stk
Basic LAM -P-jump delay time from lean to rich transition					

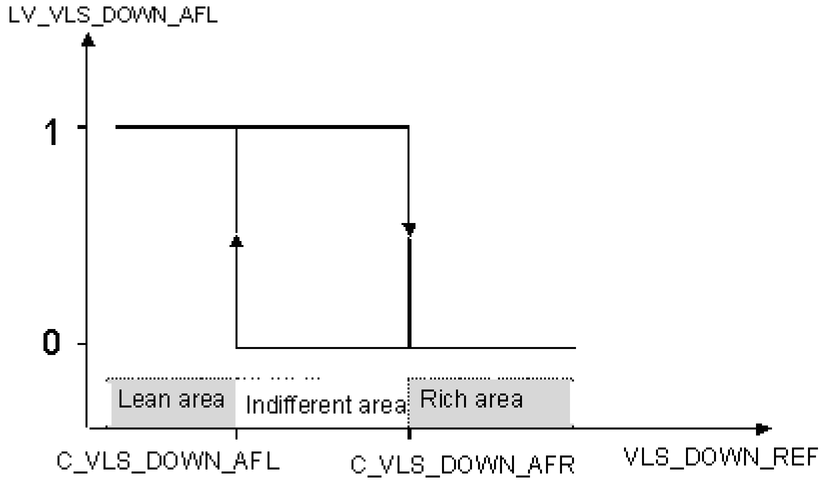
53.17.1 General information

The sub function *“Desulfation controlled lambda alternation”* allows a desulfation with an alternating lambda value controlled by the output voltage of the downstream sensor. The deviation of the basic lambda setpoint will only be carried out, if the lambda setpoint is in the range of 1. Alternation is defined by the maximum rich rsp. lean lambda shift, the p-jump and the delay time at rich or lean side befor p-jump. Gradients are defined according to engine conditions.

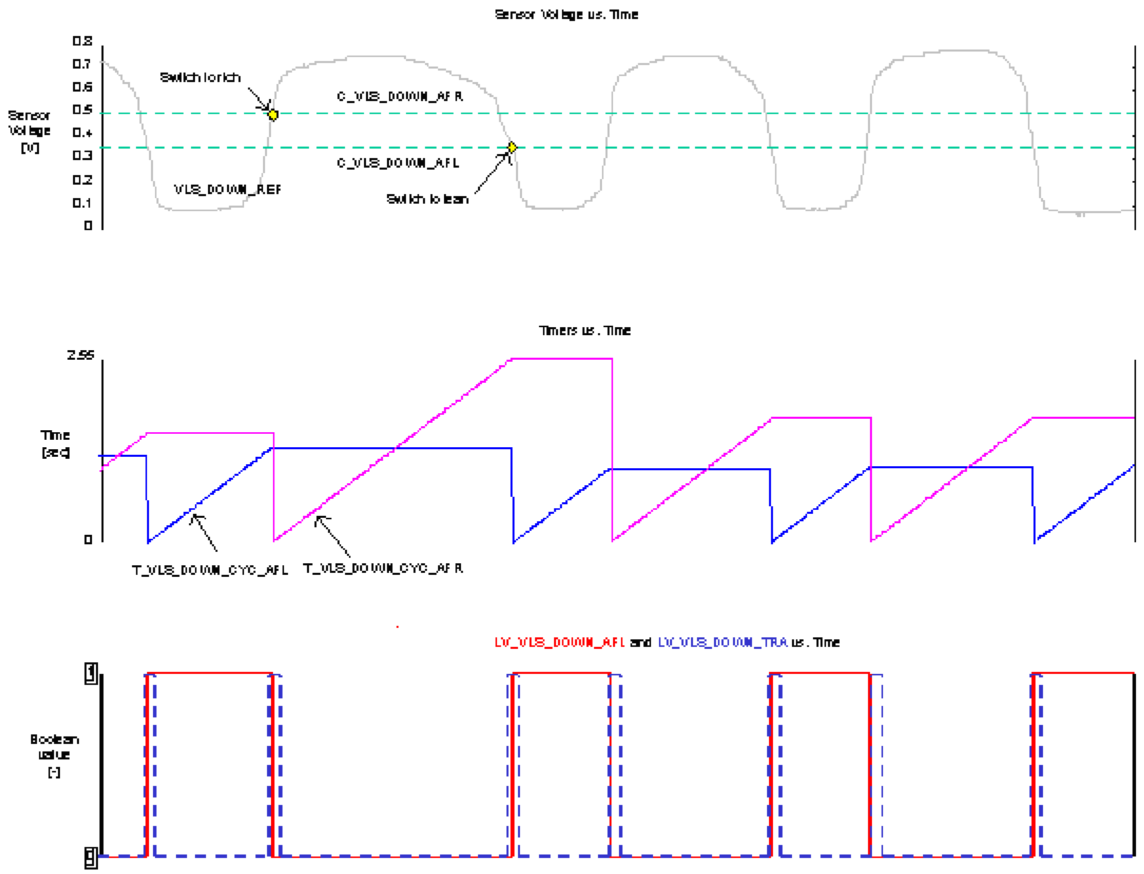
Definition of lambda deviation for lambda controller:



State and hysteresis of LV_VLS_DOWN_AFL as function of VLS_DOWN:



Physical relationship between values, Air-Fuel cycle evaluation at LS downstream:



Application Condition

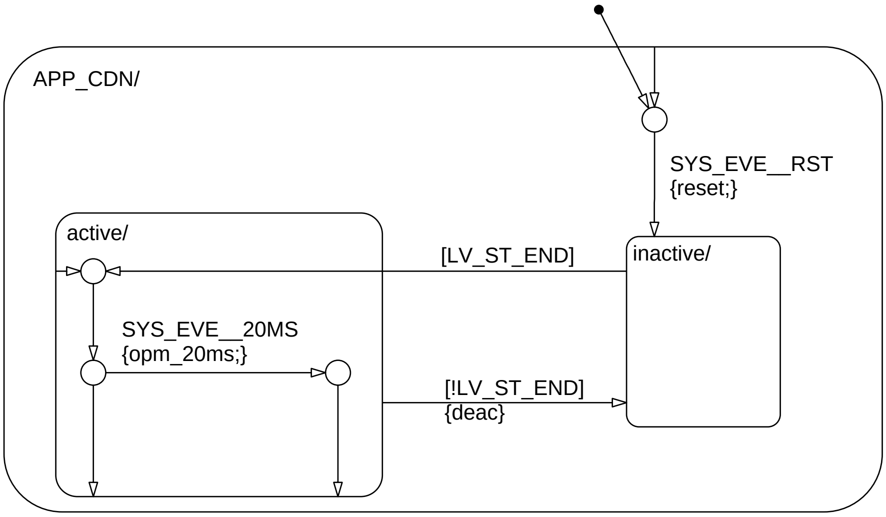
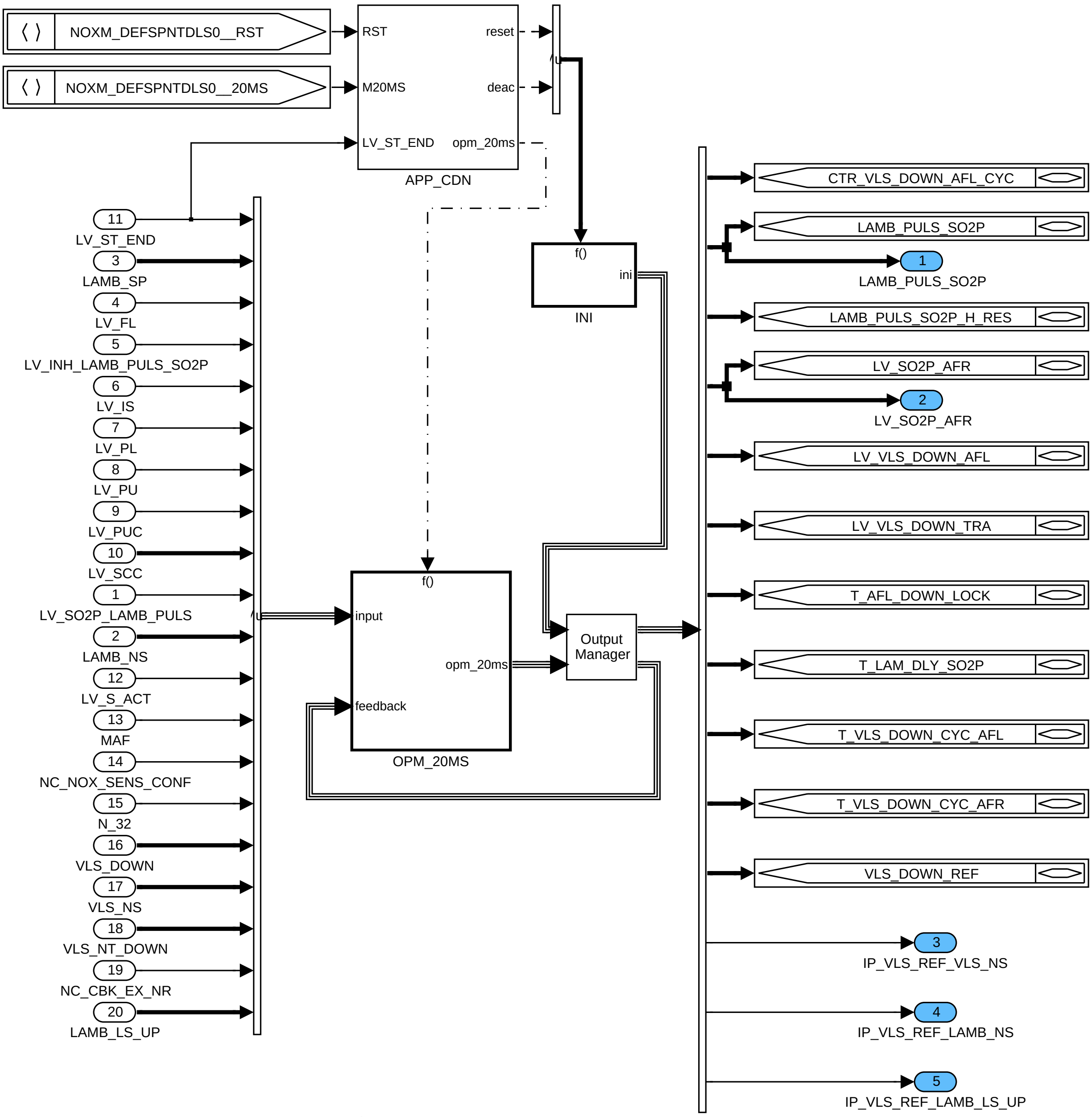


Figure 186 NOXM_DEFSPNTDLS0/ APP_CDN/ Chart1

Function Description



SDA_SRS / SDA 4.0 15−Mar−2005

Figure 187 NOXM_DEFSPNTDLS0

53.17.1.1 INITIALIZATION

Initialization at reset

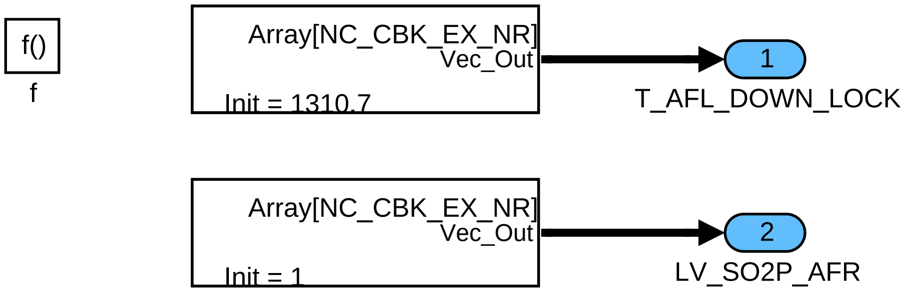


Figure 188 NOXM_DEFSPNTDLS0/ INI/ INI_reset

Initialization at deactivation

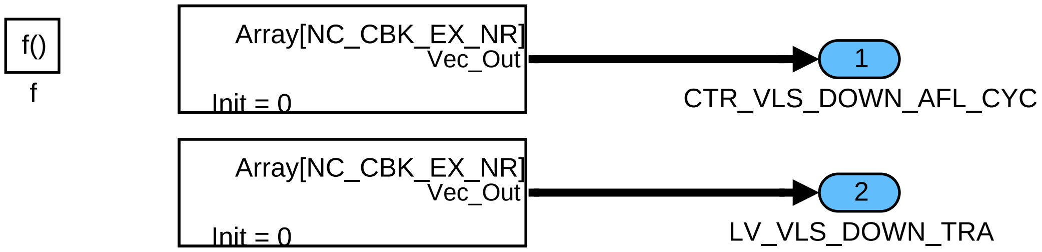


Figure 189 NOXM_DEFSPNTDLS0/ INI/ INI_deac

53.17.1.2 FORMULA SECTION

The function contains of two main Sections:

SECTION 1: Detection of lean/rich mixture

SECTION 2: Realization of lambda controller deviation

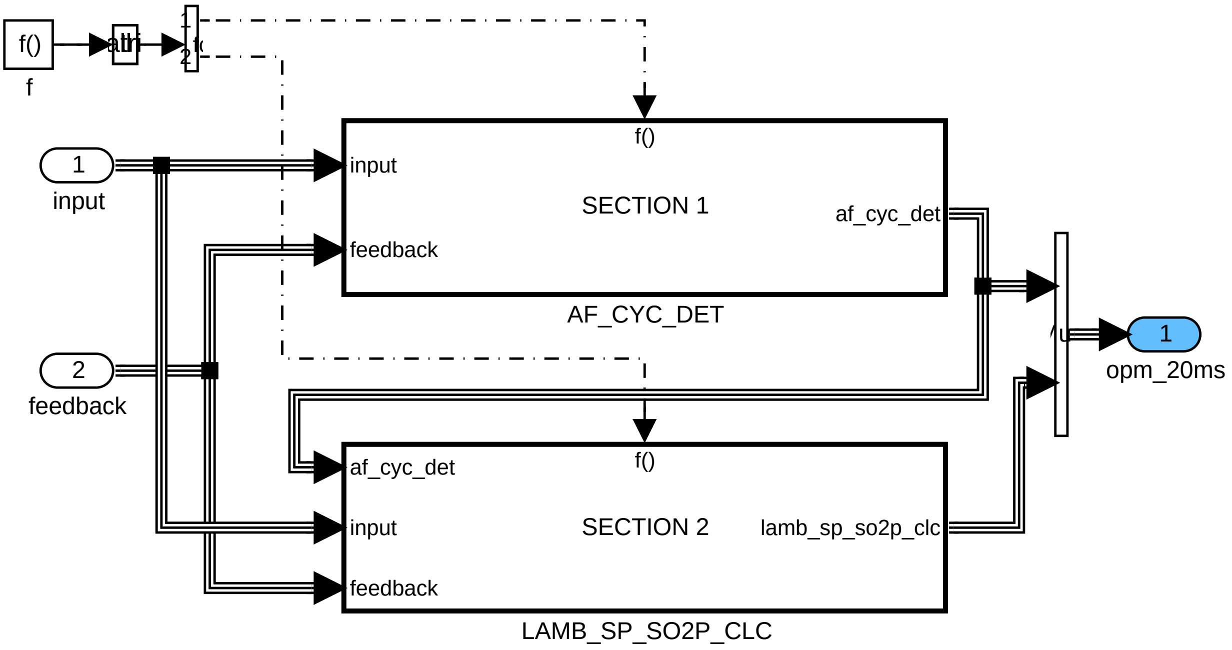


Figure 190 NOXM_DEFSPNTDLS0/ OPM_20MS

SECTION 1: Detection of lean/rich mixture

The function shall provide an indication of the current Air-Fuel mixture ratio at LS downstream position, i.e. mixture lean or rich, and determine the lean and rich cycle time separately for further evaluation by other functions. This shall be carried out as follows:

AF transition detection:

There are several possibilities as reference for oxygen signal depending on existing sensors and their provided signals: NOx-sensor or binary oxygen sensor. The choosen output signal is scaled if necessary to the reference signal VLS_DOWN_REF.

The function shall detect a change in AF mixture from lean to rich or vice versa ignoring possible jitter in the sensor signal. This shall be facilitated by the following method: If the timer T_AFL_DOWN_LOCK falls below the threshold IP_T_AFL_DOWN_LOCK_xxx, the timer shall be incremented and the transition flag LV_VLS_DOWN_TRA shall be reset. Should, during following recurrence cycles, the timer equal or exceed the threshold and if the AF mixture flag is currently indicating lean, the function shall observe the VLS_DOWN_i signal for a change to rich as indicated by VLS_DOWN_i exceeding the threshold C_VLS_DOWN_AFR. Once this has occurred, the LV_VLS_DOWN_AFL flag shall be changed to indicate a rich mixture (LV_VLS_DOWN_AFL = 0), the counter CTR_VLS_DOWN_AFL_CYC shall be incremented to indicate the number of complete AF cycles since the last activation of the lambda controller, the timer T_AFL_DOWN_LOCK shall be reset and LV_VLS_DOWN_TRA is set to indicate the AF mixture transition. The same procedure, except increment of counter CTR_VLS_DOWN_AFL_CYC, applies for a currently rich mixture. A transition to a lean mixture shall be detected by VLS_DOWN falling below threshold C_VLS_DOWN_AFL.

Remark: The timer T_AFL_DOWN_LOCK and associated engine operating point dependent threshold IP_T_AFL_DOWN_LOCK_xxx reduce the effect of rapid oxygen sensor signal changes, noise generated by cylinder to cylinder AF mixture deviations, that would cause the lambda controller to execute a P-jump. The mapped lock-out time represents the delay between injection of the AF mixture and the burnt AF mixture reaching and being detected by the downstream oxygen sensor. The lock-out time shall not exceed the exhaust gas delay for any particular engine operating point otherwise the lambda controller frequency will be affected. Downstream oxygen sensor signal changes occurring within this delay time may not be considered to be due to the change in controlled lambda but from noise and shall be ignored.

Timer / counter incrementation:

The function shall increment the respective timer dependent on the current state of the AF mixture, as determined by LV_VLS_DOWN_AFL (lean mixture; increment T_VLS_DOWN_CYC_AFL. rich mixture; increment T_VLS_DOWN_CYC_AFR). Should at any time the desulfation control be deactivated, the function shall reset the counter CTR_VLS_DOWN_AFL_CYC.

As calibrateable back up solution a switch of LV_VLS_DOWN_AFL can also be forced by exceeding the time counter threshold for the corresponding cycle time (C_T_VLS_DOWN_CYC_AFL/AFR_MAX)

Increment T_VLS_DOWN_CYC_AFL / Increment T_VLS_DOWN_CYC_AFR

The timers T_VLS_DOWN_CYC_AFL and T_VLS_DOWN_CYC_AFR shall be incremented according to their specified conditions until they reach a maximum value of FFFFH at which point they shall no longer be incremented.

Increment CTR_VLS_DOWN_AFL_CYC

The Air-Fuel cycle counter CTR_VLS_DOWN_AFL_CYC shall be incremented for every lean to rich AF mixture transition, to indicate that a complete AF cycle has passed. The counter shall be incremented in such manner until it reaches a maximum value of FFFFH at which point it shall no longer be incremented.

Increment T_AFL_DOWN_LOCK

The timer T_AFL_DOWN_LOCK shall be incremented according to the specified conditions until it reaches a maximum value of FFFFH at which point it shall no longer be incremented.

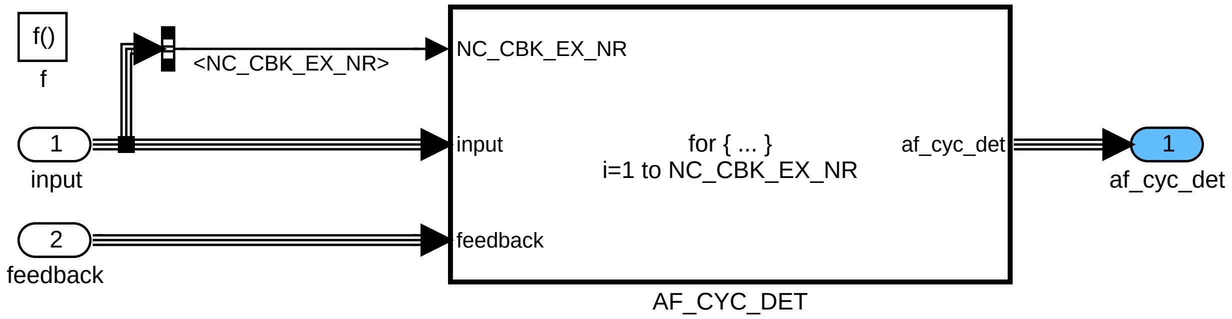


Figure 191 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET

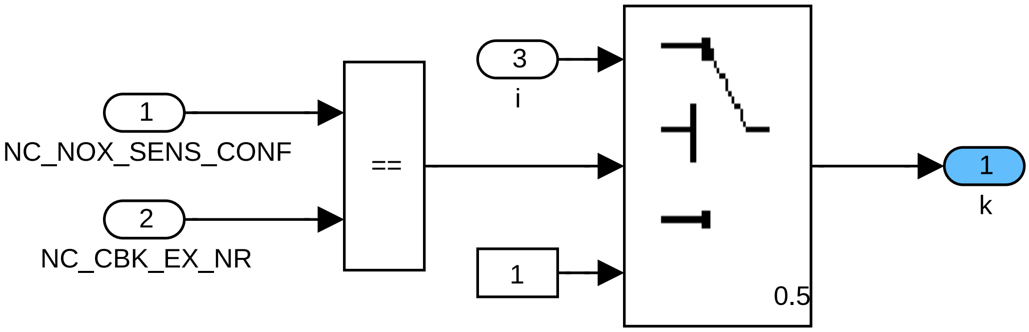


Figure 192 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ INPUT/ CLC_k

Content of SECTION 1

Section 1 contains of two other sections:

SECTION 1.1: Calculation of a voltage reference signal (CASE_SEL)

SECTION 1.2: Detection of transition between rich/lean phases and current mixture state (CHK_AF_CDN)

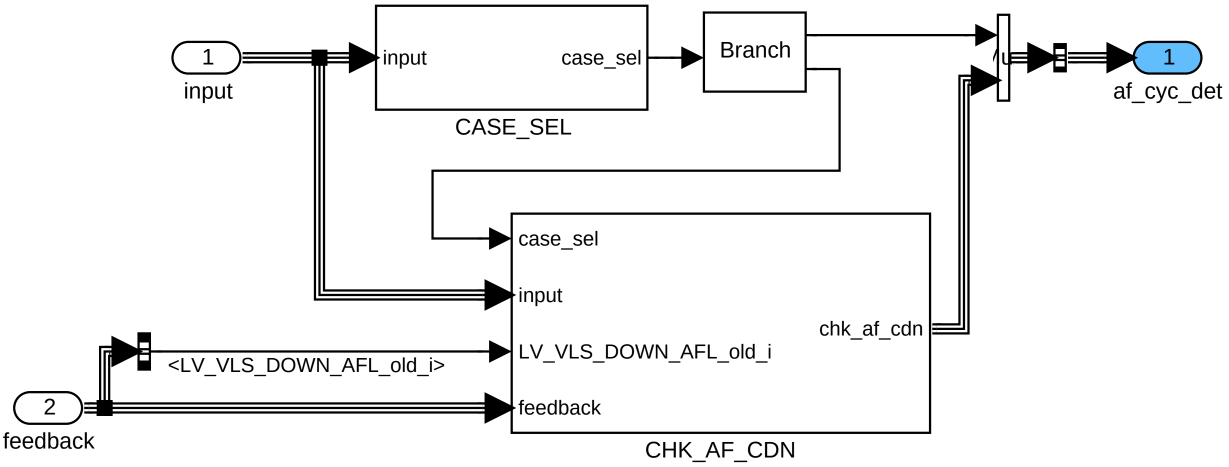


Figure 193 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET

SECTION 1.1: Calculation of a voltage reference signal

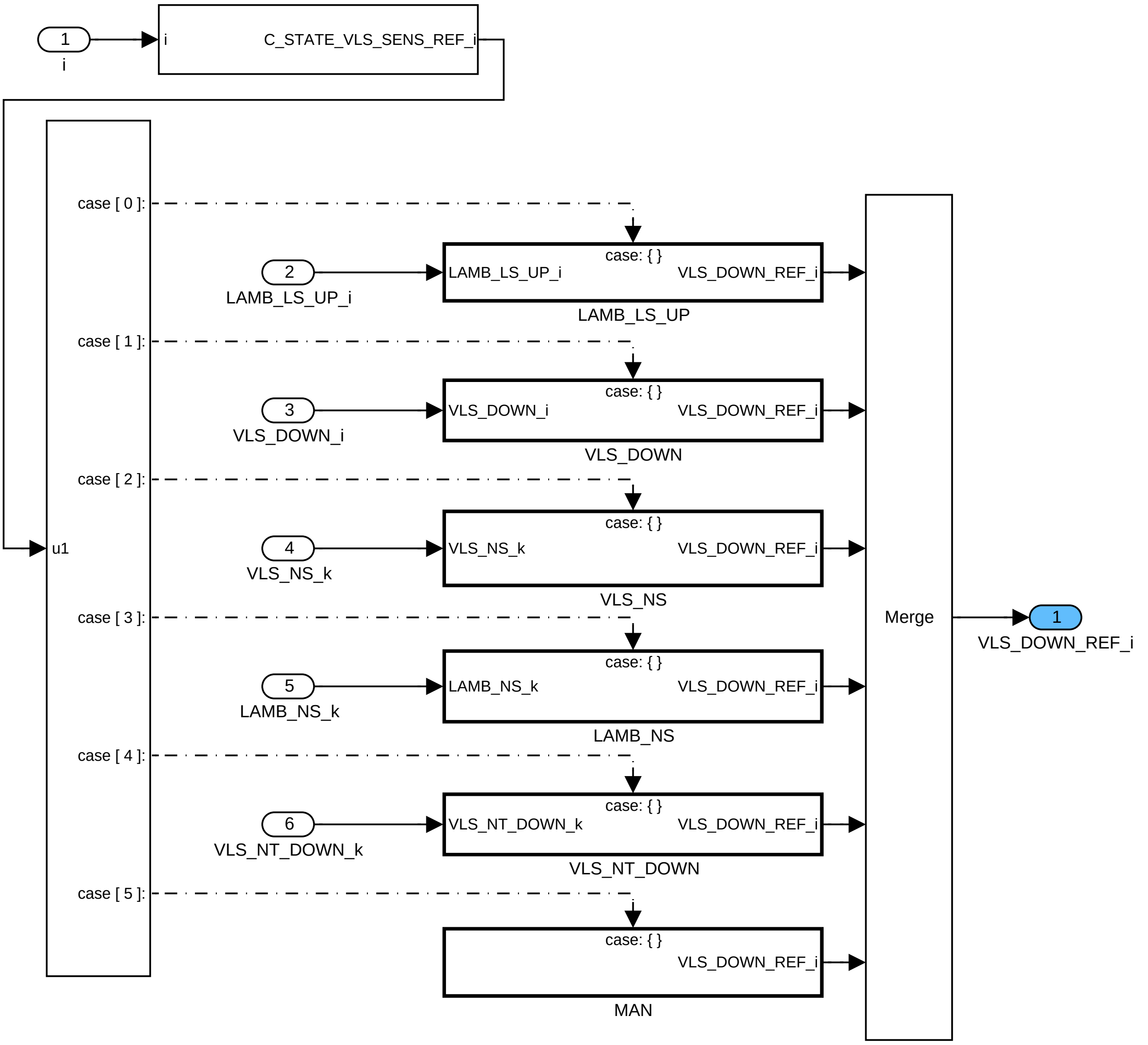


Figure 194 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CASE_SEL/ CASE_SEL

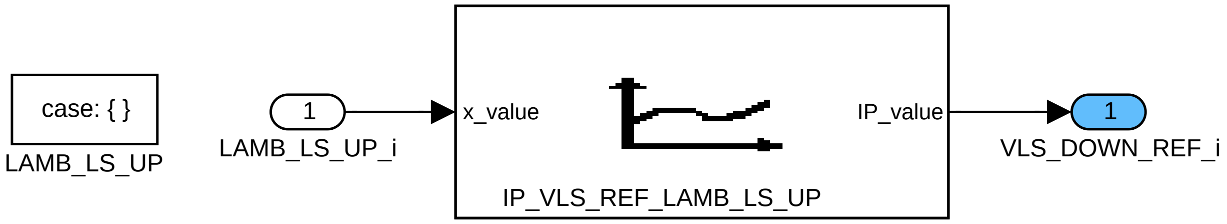


Figure 195 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CASE_SEL/ CASE_SEL/ LAMB_LS_UP

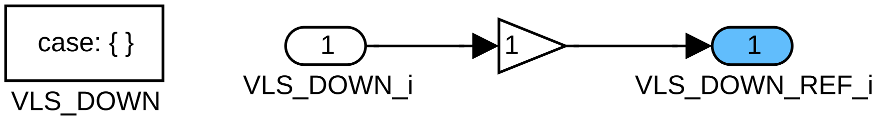


Figure 196 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CASE_SEL/ CASE_SEL/ VLS_DOWN

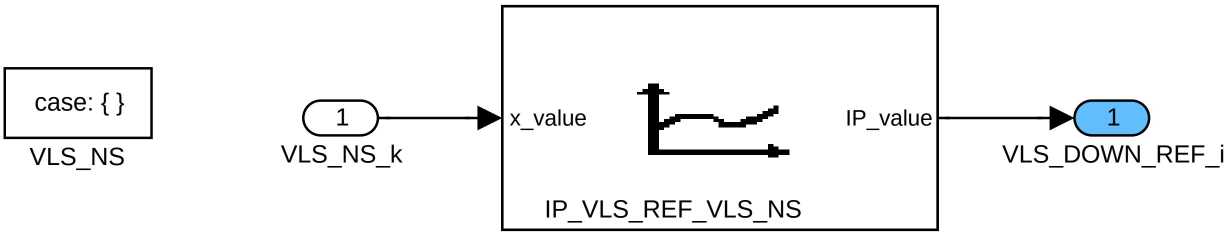


Figure 197 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CASE_SEL/ CASE_SEL/ VLS_NS

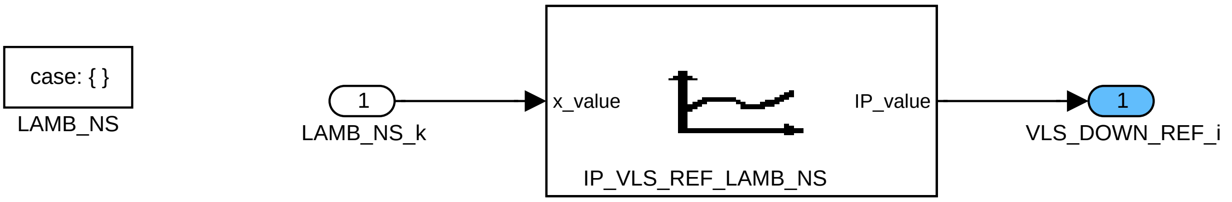


Figure 198 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CASE_SEL/ CASE_SEL/ LAMB_NS

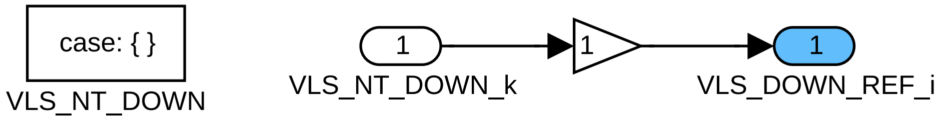


Figure 199 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CASE_SEL/ CASE_SEL/ VLS_NT_DOWN

Figure 200 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CASE_SEL/ CASE_SEL/ MAN

SECTION 1.2: Detection of transition between rich/lean phases and current mixture state

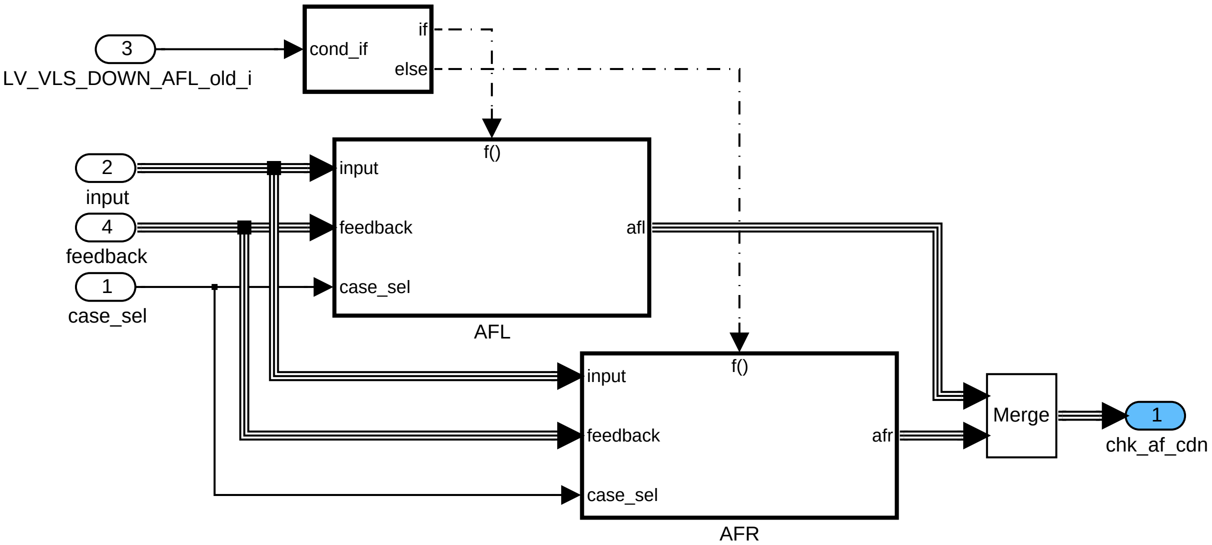


Figure 201 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CHK_AF_CDN

SECTION 1.2.1: AFL phase

Calculation of timer for lean cycle

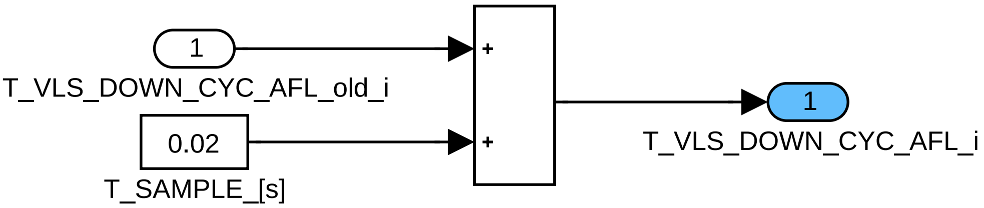


Figure 202 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CHK_AF_CDN/ AFL/ T_CYC_AFL_CLC

Calculation of transition condition

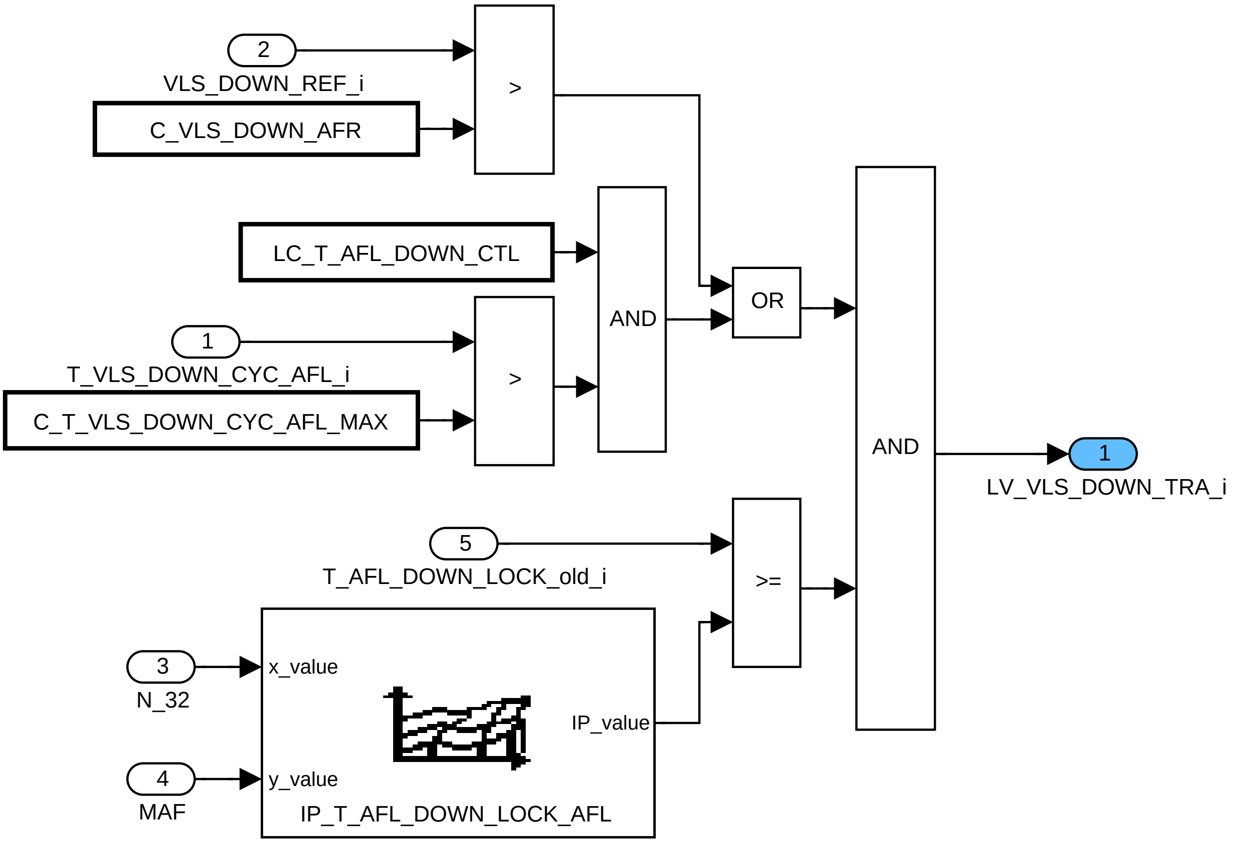


Figure 203 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CHK_AF_CDN/ AFL/ LV_TRA_CLC

Calculation of mixture state

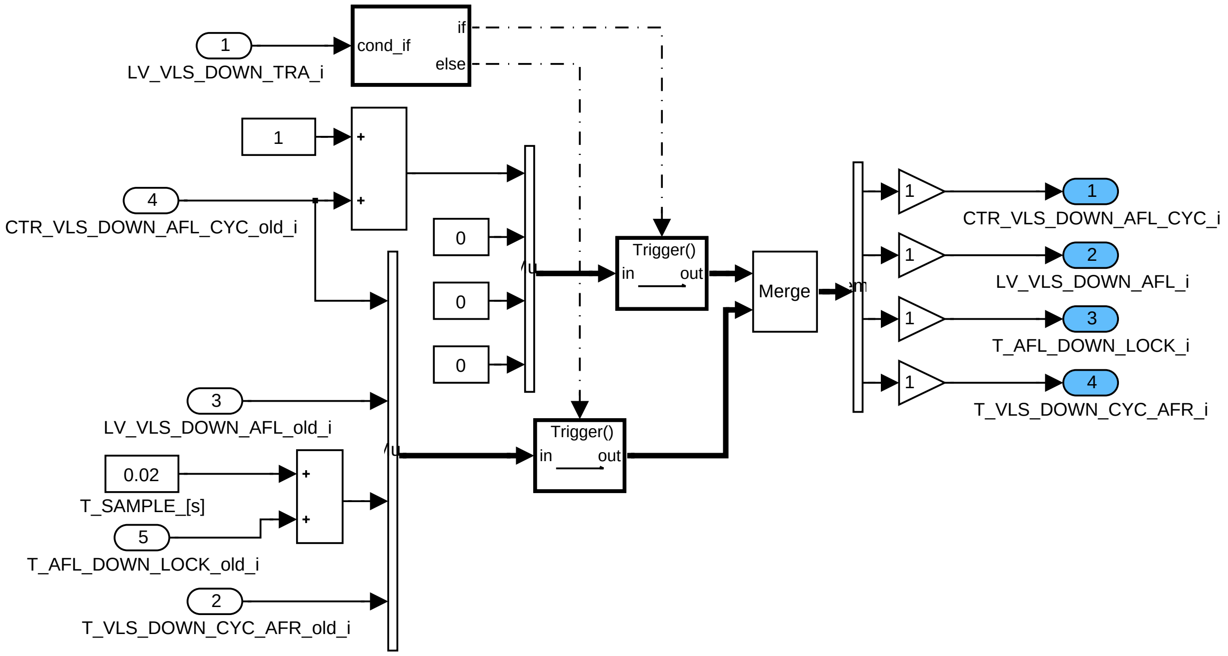


Figure 204 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CHK_AF_CDN/ AFL/ LV_AFL_CLC

SECTION 1.2.2: AFR phase

Calculation of timer for rich cycle

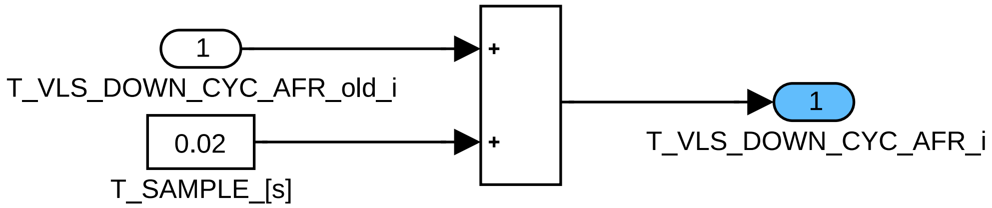


Figure 205 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CHK_AF_CDN/ AFR/ T_CYC_AFR_CLC

Calculation of transition condition

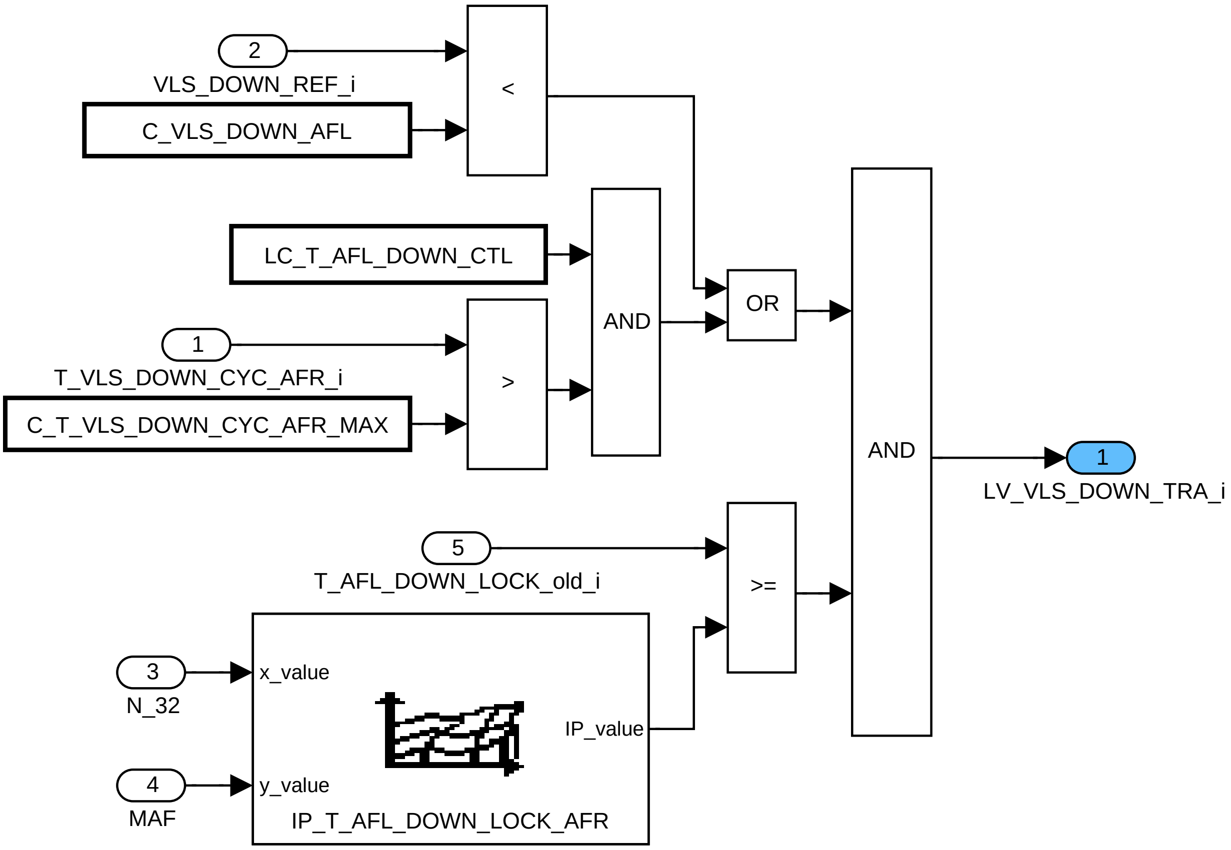


Figure 206 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CHK_AF_CDN/ AFR/ LV_TRA_CLC

Calculation of mixture state

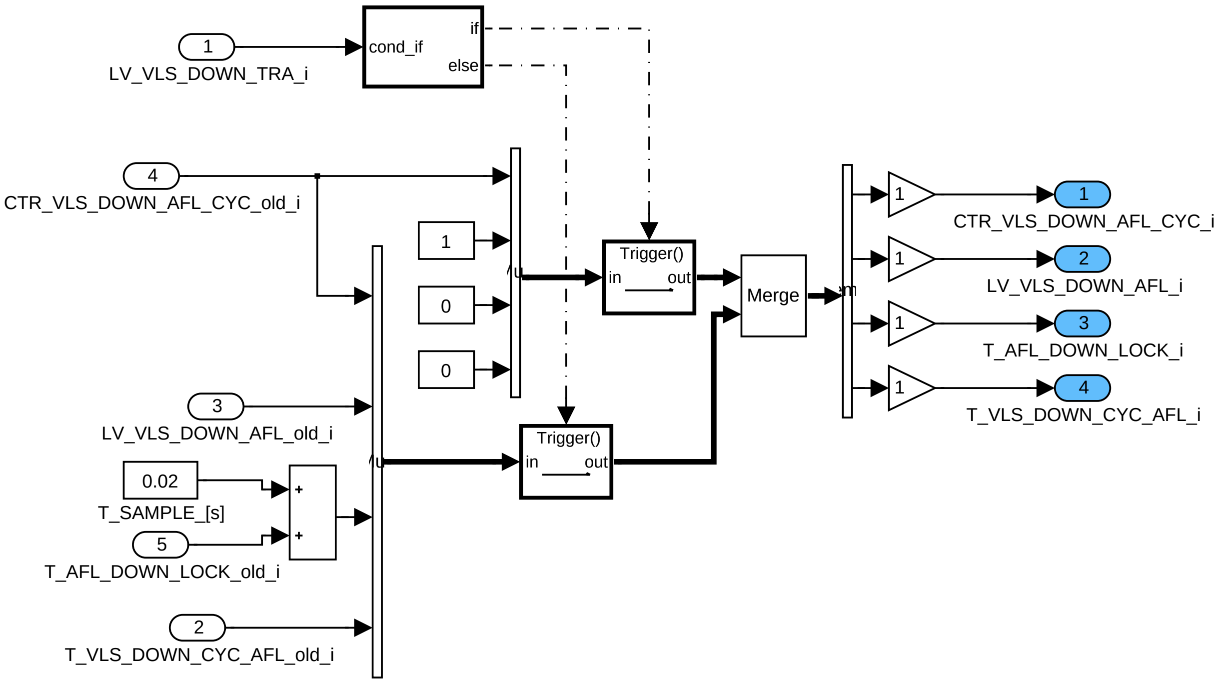


Figure 207 NOXM_DEFSPNTDLS0/ OPM_20MS/ AF_CYC_DET/ AF_CYC_DET/ AF_CYC_DET/ CHK_AF_CDN/ AFR/ LV_AFR_CLC

SECTION 2: Realization of lambda controller deviation

The lambda control correction adjusts the fuel-air mixture to richness 1 in average. The lambda controller uses a PI strategy. It is adjusted to produce a leaner or a richer mixture according to the oxygen sensor information.

If LAMB_PULS_SO2P exceeds the limitations C_DELTA_LAMB_AFL/AFR_SO2P then LAMB_PULS_SO2P is limited to those values.

The calibration constants C_T_LAM_DLY_SO2P_POS and C_T_LAM_DLY_SO2P_NEG are provided for the pre-control of the P-jump delay. The IP_LAM_XX_YY maps have common data points.

The last calculated LAMB_PULS_SO2P value before the corresponding Lambda transition from lean to rich and from rich to lean remains constant until the delay time T_LAM_DLY_SO2P has come to an end, i.e. I component is frozen. LAMB_PULS_SO2P can not be changed or reset during this time.

To provide best performance (drivability, interface to other functions as e.g. desulfation) all banks reaching lean status wait with P-jump to rich side until last bank also has reached lean status (common P-jump to rich side).

High resolution variable LAMB_PULS_SO2P_H_RES_i shall be used for internal calculation. LAMB_PULS_SO2P_i with low resolution shall be the relevant output for other functions.

Remark: The transition flag LV_VLS_DOWN_TRA is just set during one recurrence of the corresponding specification. It has to be ensured that this spike is noticed by this part.

This SECTION contains of two main chapters:

SECTION 2.1: Calculation of activation condition and check of all banks (not for loop)

SECTION 2.2: Lambda deviation calculation (for loop)

SECTION 2.1: Calculation of activation condition and check of all banks (not for loop)

Activation condition for desulfation

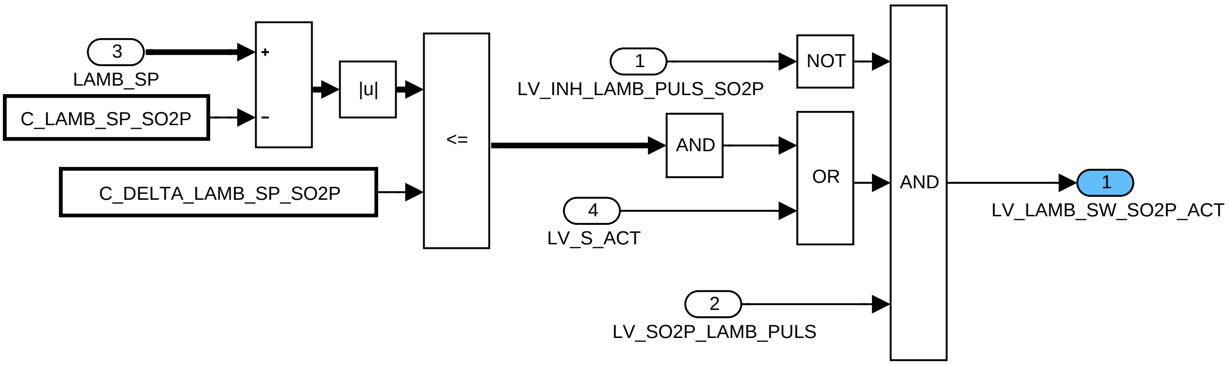


Figure 208 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ ACT__AFL_DET/ SO2P_ACT_CDN

Check all exhaust banks

Index "any" means that minimum for one bank the condition is fulfilled.

Index "all" means that for all banks the condition must be fulfilled.

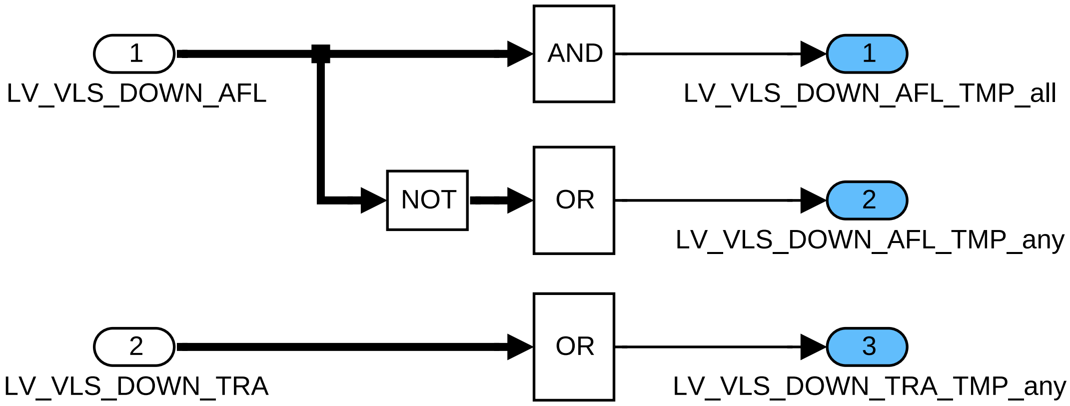


Figure 209 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ ACT__AFL_DET/ AFL_DOWN_DET

SECTION 2.2: Lambda deviation calculation (for loop)

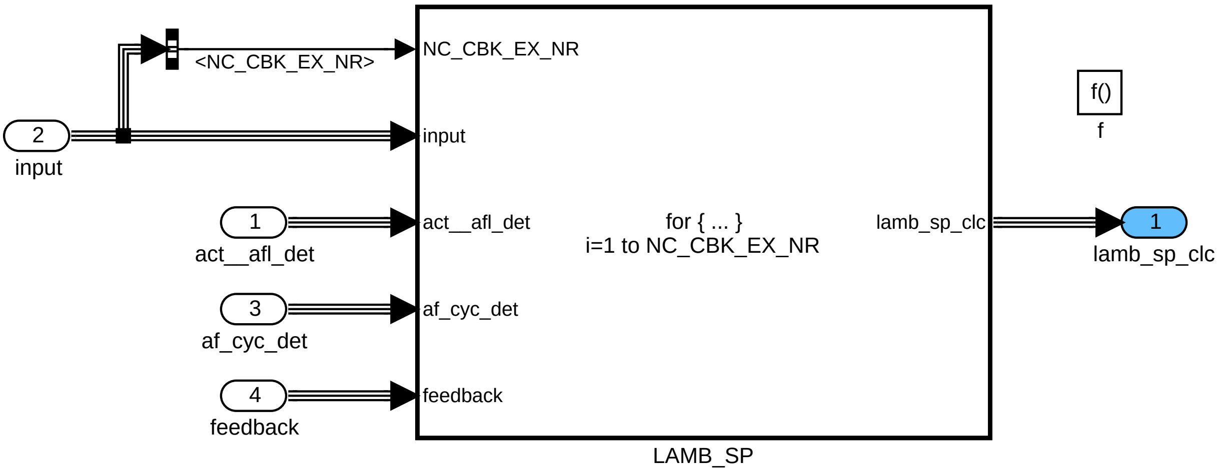


Figure 210 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP

SECTION 2.2.1: Calculation of lambda deviation for desulfation (H_RES)

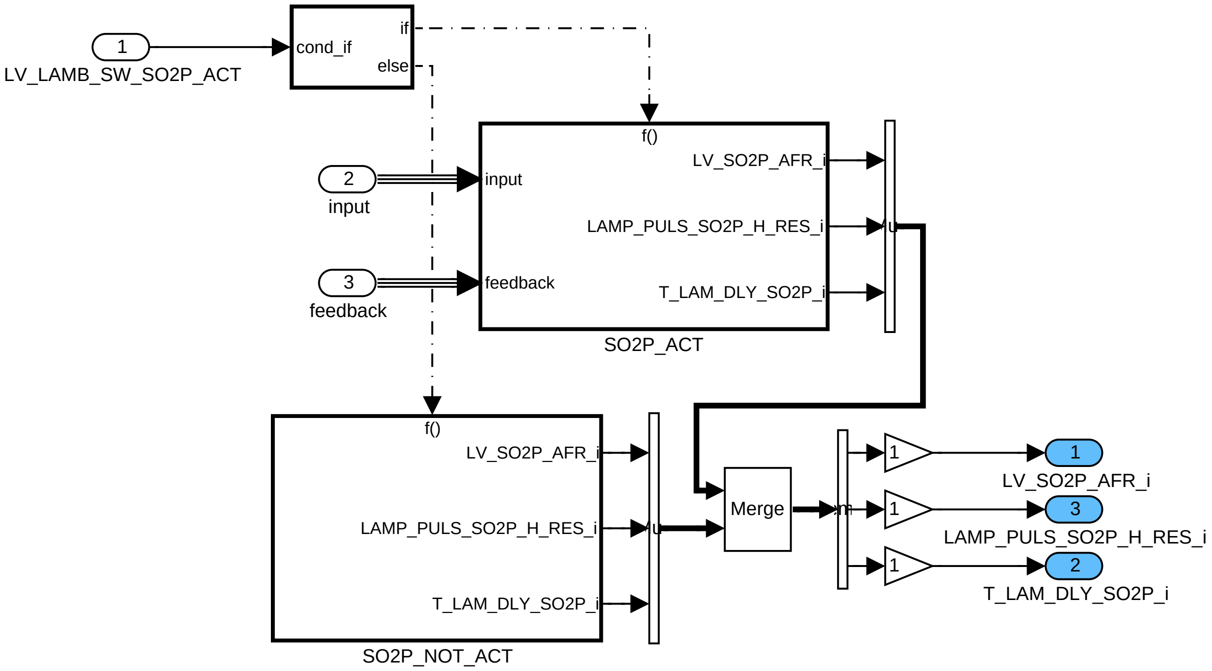


Figure 211 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC

SECTION 2.2.1.1: Desulfation active (SO2P_ACT)

SECTION 2.2.1.1.1: Check of combustion condition 1:

Calculation of lambda deviation at trailing throttle fuel cut-off or full load or single cylinder cut off

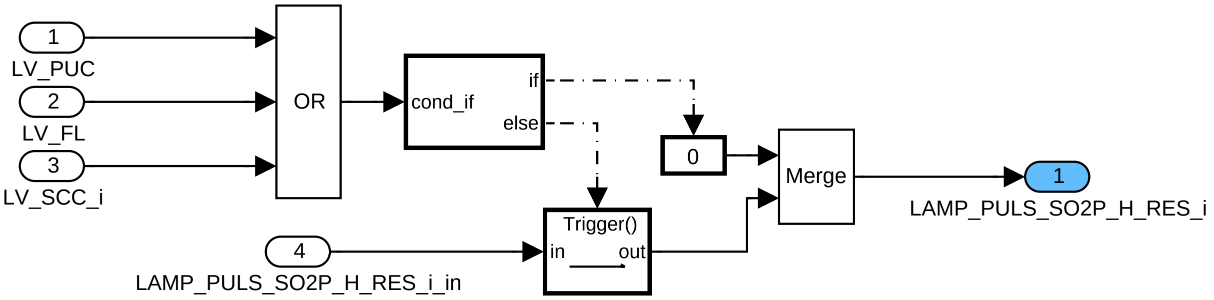


Figure 212 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_1

SECTION 2.2.1.1.2: Check of combustion condition 2:

Controller setting at part load or trailing throttle or idle

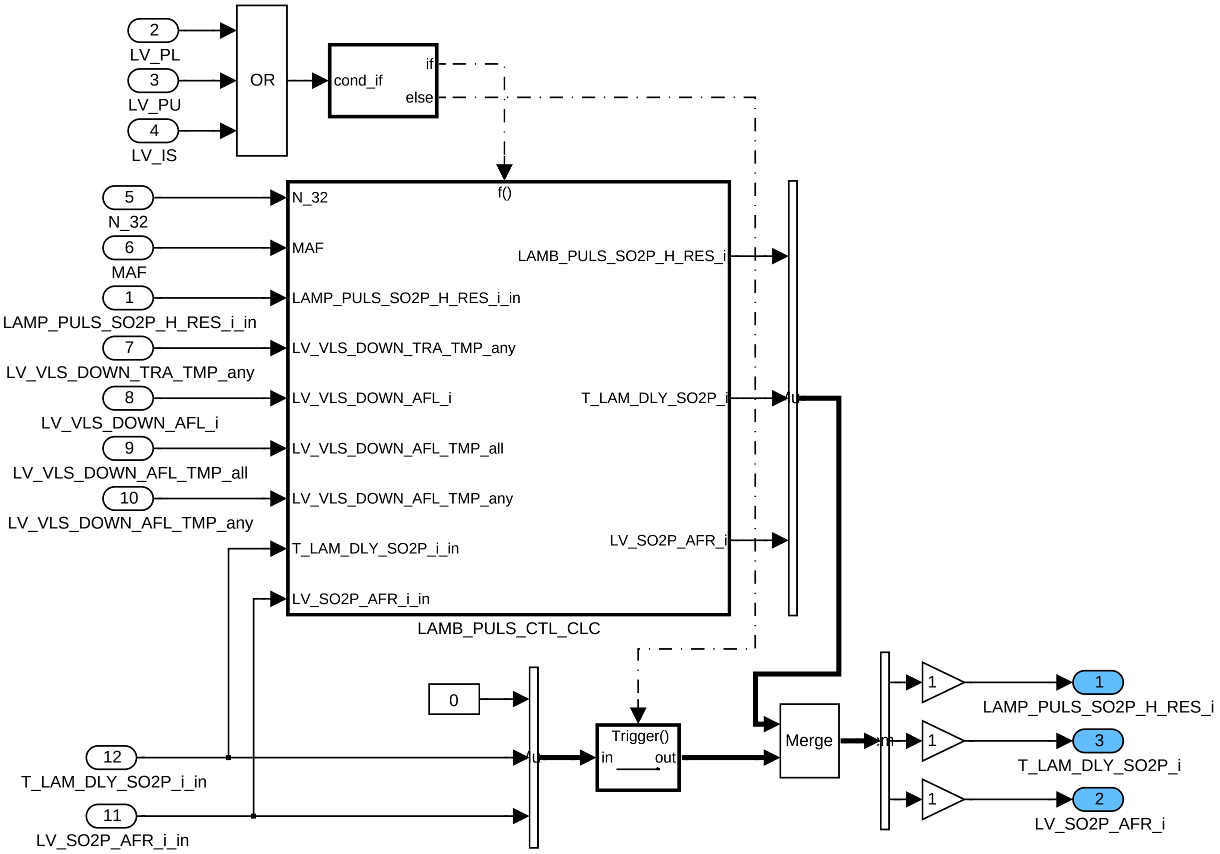


Figure 213 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2

SECTION 2.2.1.1.2.1: LAMB_PULS_CTL_CLC

In this section the controller setting is calculated for transition phase and during rich/lean phase

SECTION 2.2.1.1.2.1.1: Controller setting for transition phase

Initialization of delay time at changing of combustion mixture and calculation which phase is active

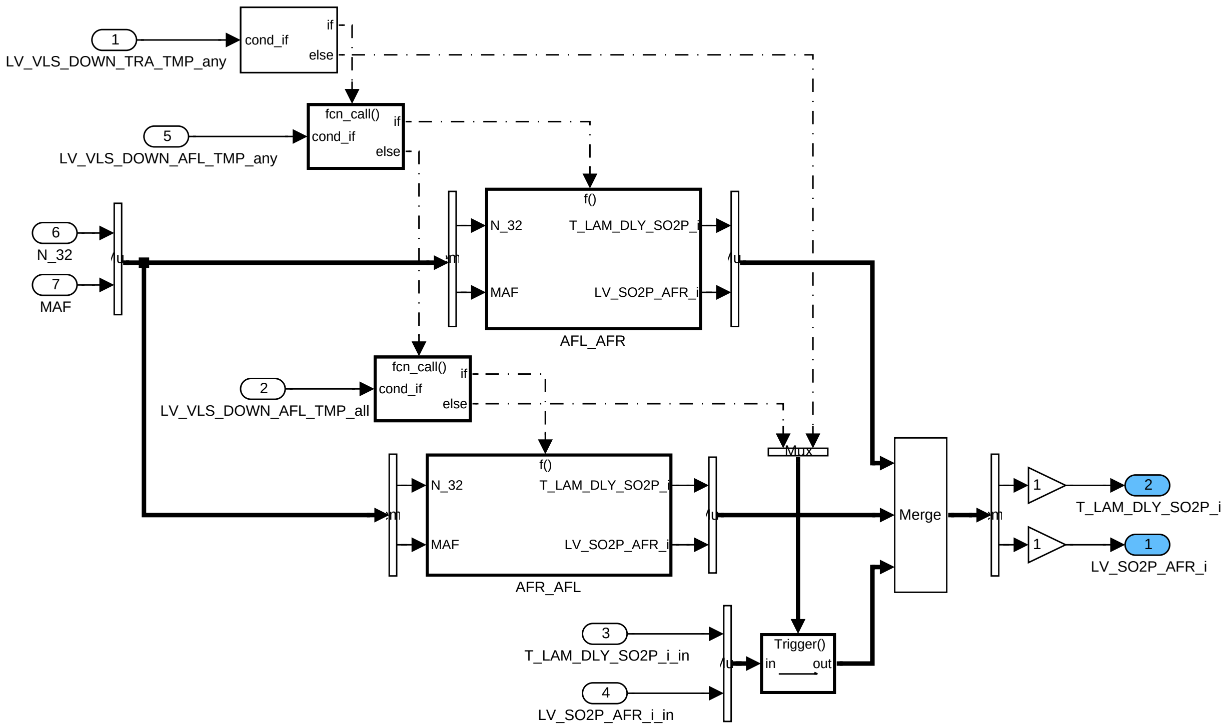


Figure 214 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ TRA

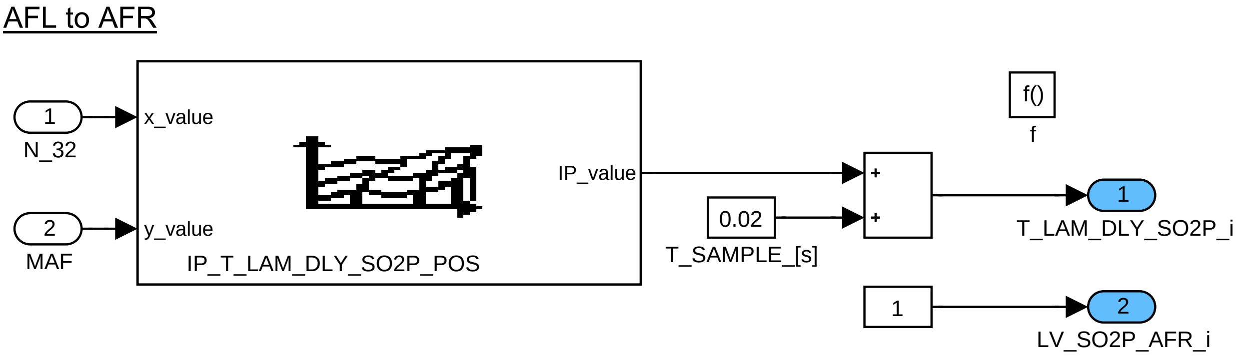


Figure 215 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ TRA/ AFL_AFR

AFR to AFL

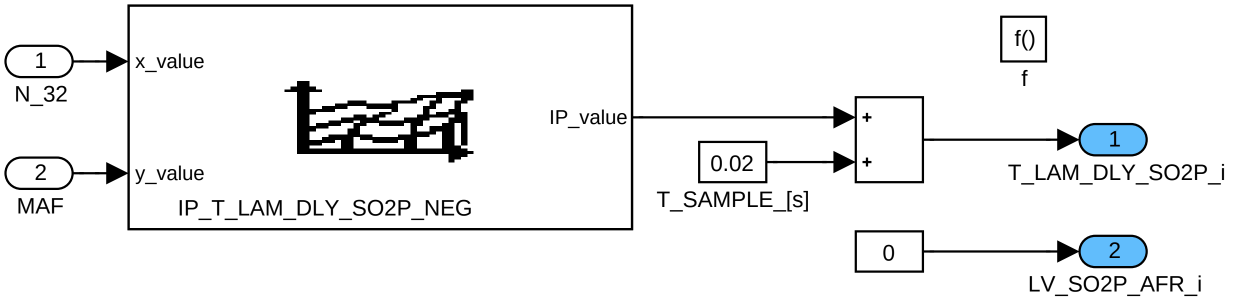


Figure 216 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ TRA/ AFR_AFL

SECTION 2.2.1.1.2.1.2: Controller setting for lean/rich phase

If the delay time has been running over the "I" component of controller is calculated

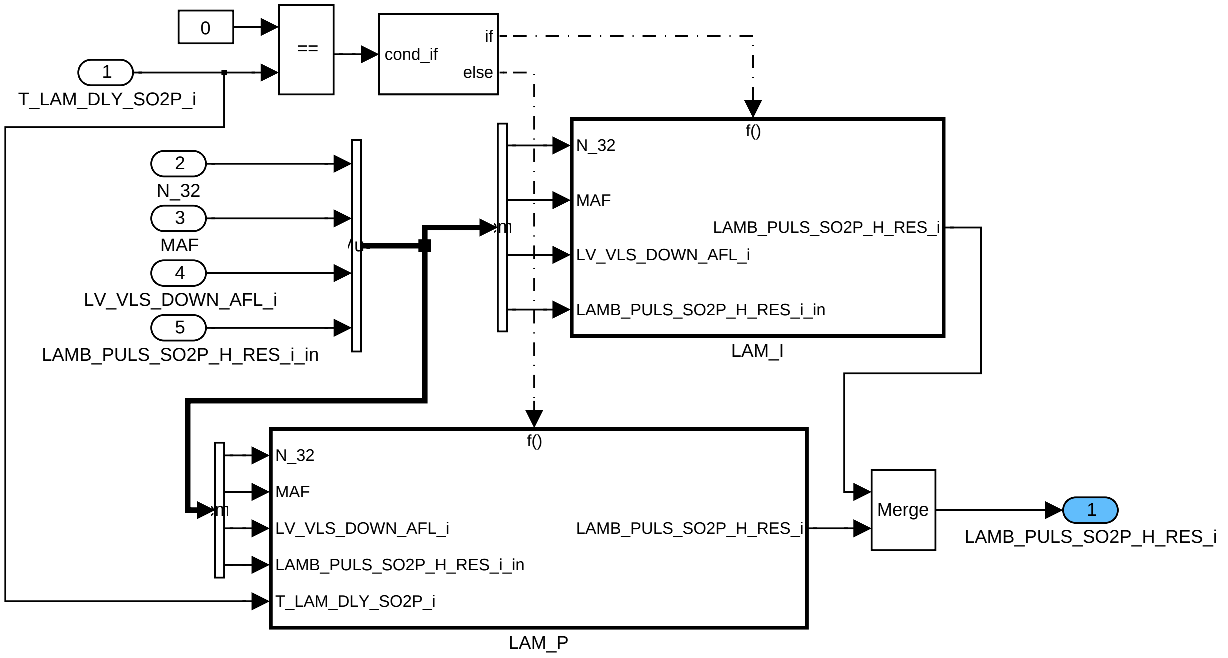


Figure 217 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ CTL

LAM_I

Calculation of controller "I" component

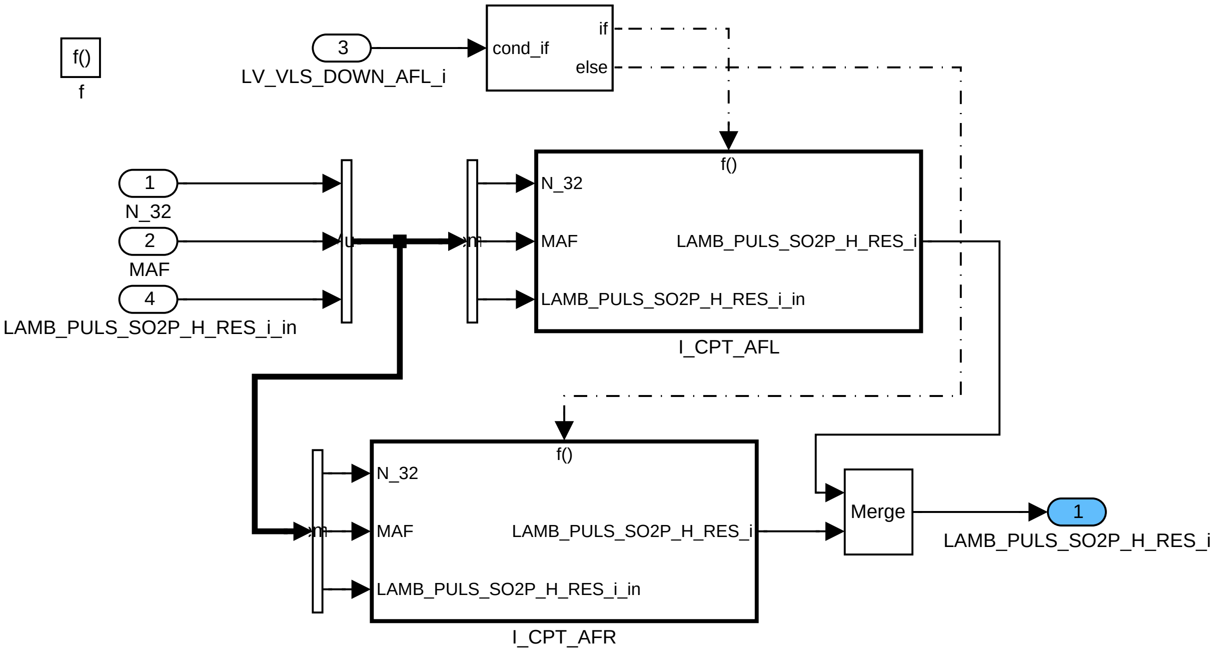


Figure 218 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ CTL/ LAM_I

LAM_I/I_CPT_AFL

Controller "I" component for lean phase

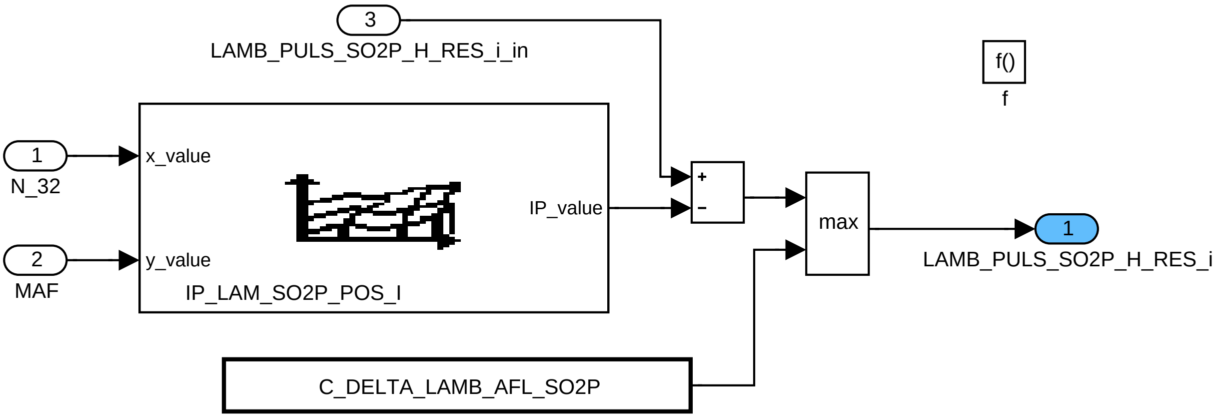


Figure 219 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ CTL/ LAM_I/ I_CPT_AFL

LAM_I/I_CPT_AFR

Controller "I" component for rich phase

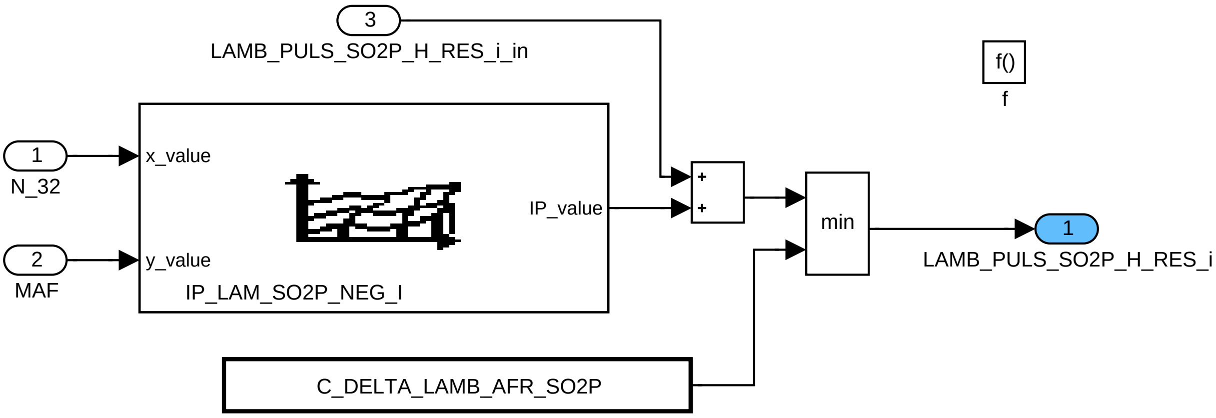


Figure 220 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ CTL/ LAM_I/ I_CPT_AFR

LAM_P

Calculation of controller "P" component. At the last cycle before delay time is over the "P" component is calculated for realization of "P" jump

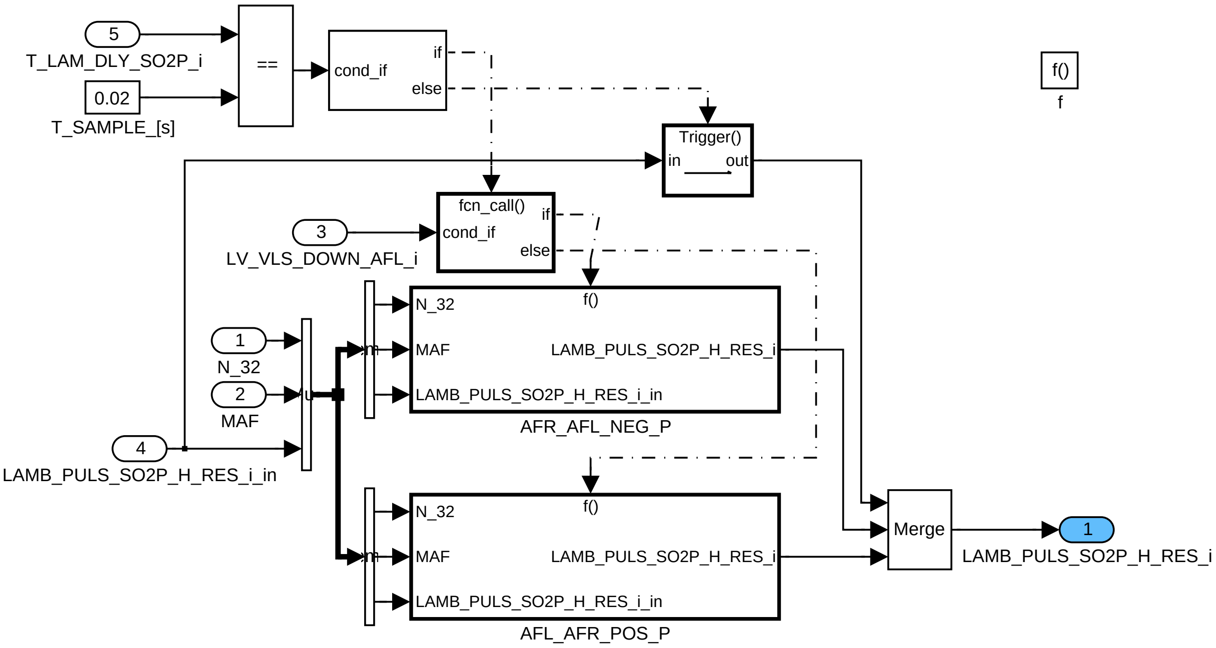


Figure 221 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ CTL/ LAM_P

LAM_P/AFR_AFL_NEG_P

Controller "P" component for lean to rich transition

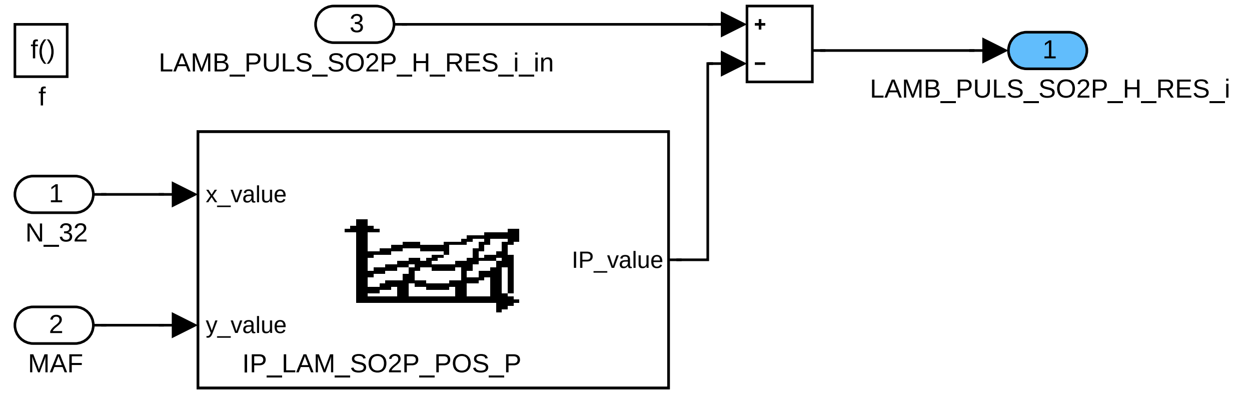


Figure 222 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ CTL/ LAM_P/ AFR_AFL_NEG_P

LAM_P/AFL_AFR_POS_P

Controller "P" component for rich to lean transition

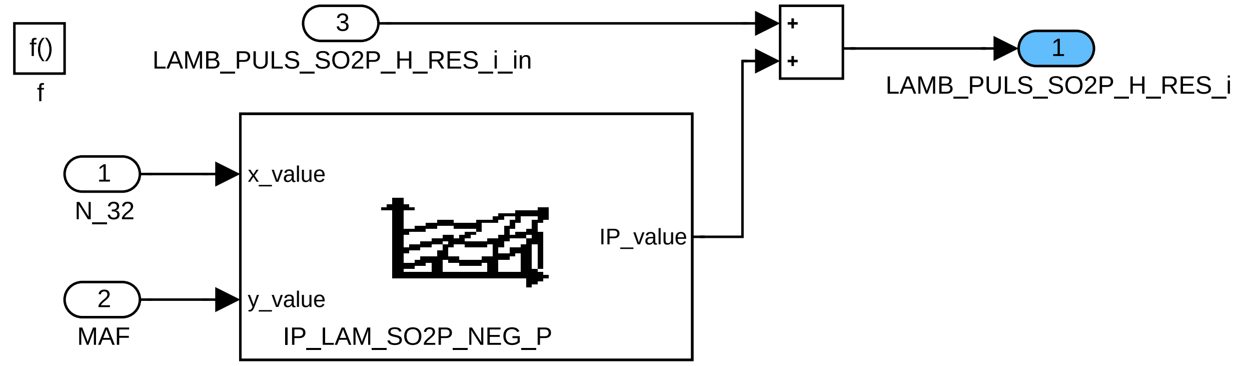


Figure 223 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ CMB_CDN_2/ LAMB_PULS_CTL_CLC/ CTL/ LAM_P/ AFL_AFR_POS_P

SECTION 2.2.1.1.3: Calculation of delay time for controller setting

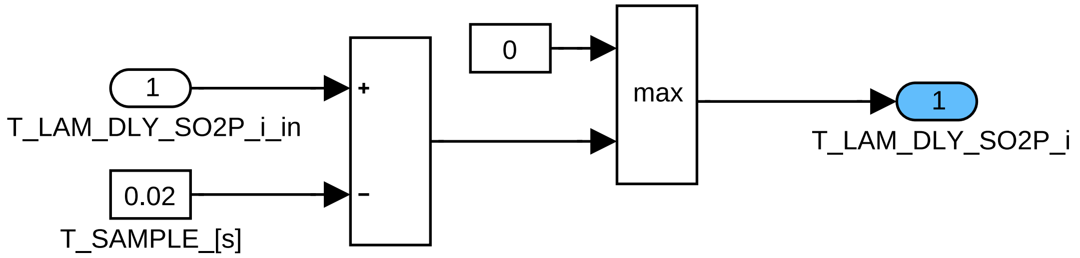


Figure 224 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_ACT/ T_LAM_DLY_CLC

SECTION 2.2.1.2: Desulfation not active (SO2P_NOT_ACT)

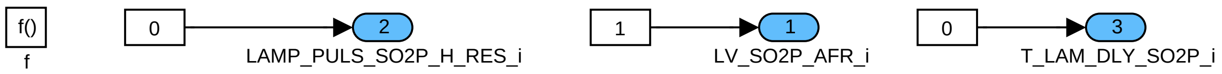


Figure 225 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_H_RES_CLC/ SO2P_NOT_ACT

SECTION 2.2.2: Calculation of lambda deviation for desulfation

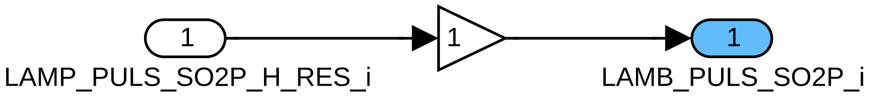


Figure 226 NOXM_DEFSPNTDLS0/ OPM_20MS/ LAMB_SP_SO2P_CLC/ LAMB_SP/ LAMB_SP/ LAMB_SP/ LAMB_SP_CLC

53.18 Application Incidence for NOx catalyst desulfation - λ stimulation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_LAMB_PULS_SO2P	V/O	0...1H	0...1	1	[-]
flag inhibiting the NOx catalyst desulfation - lambda stimulation					

Input data:

LV_ERR_LS_DOWN[NC_CBK_EX_NR]			

Description:

This flag deactivates the NOx catalyst desulfation with lambda stimulation.

With the calibration data LC_INH_LAMB_PULS_SO2P_MAN it is possible to deactivate the NOx trap desulfation with lambda stimulation.
With the calibration data LC_INH_ LAMB_PULS_SO2P_OFF it is possible to make the NOx trap desulfation with lambda stimulation allways working.

Application conditions:

Initialisation: 0 at reset

Recurrence: 100ms

Activation: at every engine state

```
If      LC_INH_LAMB_PULS_SO2P_OFF = 1
then    LV_INH_LAMB_PULS_SO2P = 0    % no further calculation necessary
endif
```

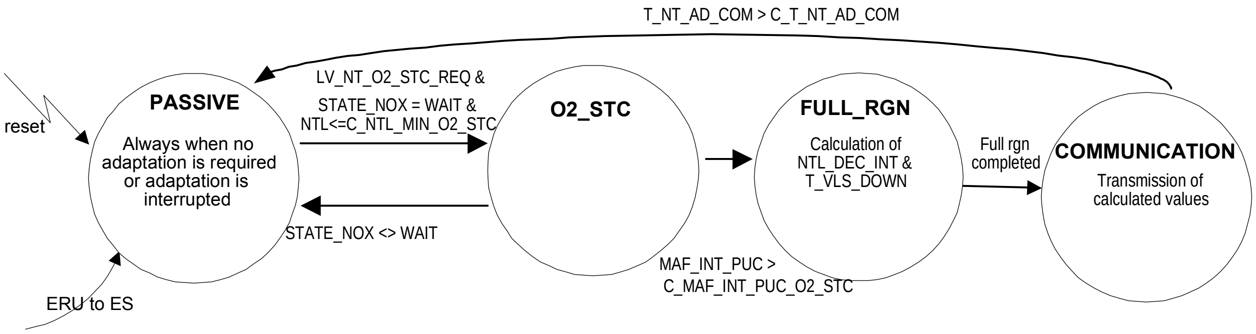
Formula section:

```
If      LV_ERR_LS_DOWN[i] = 1            // at least one
    or  LC_INH_LAMB_PULS_SO2P_MAN = 1
then    LV_INH_LAMB_PULS_SO2P = 1
else    LV_INH_LAMB_PULS_SO2P = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_LAMB_PULS_SO2P_MAN	1	0...1H	0...1	1	[-]
Codeword for the manual setting of the inhibiting bit NOx trap desulfation by lambda stimulation					
LC_INH_LAMB_PULS_SO2P_OFF	1	0...1H	0...1	1	[-]
Codeword for the deactivation of inhibiting the NOx trap desulfation by lambda stimulation					

53.19 NOx catalyst management - auxiliary functionalities



Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_NOX_PRJ	V/O	0H 2H 3H 4H	PASSIVE FULL_RGN COMMUNICATION O2_STC	1	[-]
state of auxiliary functionalities manager					
LV_NT_PRJ_ACT	V/O	0...1H	0...1	1	[-]
flag indicating an activated aux. func. module					
LV_INH_RGN_AD	V/O	0...1H	0...1	1	[-]
flag inhibiting a regeneration request by NOx-Sensor or loading degree model					
LV_NT_RGN_REQ_AD	V/O	0...1H	0...1	1	[-]
Request for a full regeneration for adaptation or elongated stratified mode					
LV_RGN_REQ_AD_VLD	V	0...1H	0...1	1	[-]
Flag to set a valid request for a full regeneration					
LV_NT_O2_STC_ACT	V/O	0...1H	0...1	1	[-]
flag for active O2_STC measurement					
LV_NT_AD_VLD	V/O	0...1H	0...1	1	[-]
flag indicating that the evaluation of the adaption values is allowed					
LV_NT_AD_CMPL	V/O	0...1H	0...1	1	[-]
flag for comminication with NOx catalyst aging module indicating a successfull adaptation cycle					
LV_RGN_CDN_AD	V	0...1H	0...1	1	[-]
flag for validity of sensor signal during regeneration					
MAF_INT_PUC_O2_STC	V	0...FFFFH	0...2912.66666	0.0444444	[g]
integral of mass air flow while PUC active, modified value for O2-STC-measurement					
T_VLS_DOWN_AFL[NC_NT_NR]	V/O	0...FFFFH	0...655.35	0.01	[s]
switch time counter					
T_VLS_DOWN_SECU[NC_NT_NR]	V	0...FFFFH	0...655.35	0.01	[s]
time during which sensor signal must pass from 1. vls-thd to 2. vls-thd (safety procedure)					
T_NT_AD_COM	V	0...FFH	0...25.5	0.1	[s]
time counter for duration of communication state					
NTL_DEC_INT_AD_SNG[NC_NT_NR]	V/O	0...FFFFH	0...10485.6	0.16	[mg]
integral of reducing agent until sensor switch					
NTL_DEC_INT_THD_TMP	V	0...FFFFH	0...10485.6	0.16	[mg]
integral of reducing agent to be achieved to stop full regeneration					
LV_NT_STC_MAX_ACT	V/O	0...1H	0...1	1	[-]
flag for active full saturation					
LV_NT_STC_MAX_AFL_ACT	V/O	0...1H	0...1	1	[-]
flag for active lean phase of full saturation					
LV_SENS_AFR_O2[NC_NT_NR]	V/O	0...1H	0...1	1	[-]
threshold for detecting rich mixture downstream exceeded					

LV_SENS_AFR_AD_SECU[NC_NT_NR]	V	0...1H	0...1	1	[-]
security threshold for sensor switch detection exceeded					
NTL_DEC_INT_AD	V/O	0...FFFFH	0...10485.6	0.16	[mg]
RAI signal during adaptation (sum of both benches)					

Input data:

LV_NT_ACT	LV_INH_NT_PRJ_ACT	NTL	NTL_DEC_INT[NC_NT_NR]
LV_NT_O2_STC_REQ	LV_INH_NT_O2_STC	LV_PUC	STATE_NOX
MAF_KGH	T_RGN	VS	N_32
VLS_DOWN[NC_CBK_EX_NR]	VLS_NT_DOWN[NC_NOX_SENS_CONF]	VLS_NS[NC_NOX_SENS_CONF]	LAMB_NS[NC_NOX_SENS_CONF]
MAF_INT_PUC_ACT	LV_INH_NT_RGN_REQ	LV_NT_SENS_AFR[NC_NT_NR]	T_AFL
LV_S_REQ_EGR	NTL_DEC_INT_SWI[NC_NT_NR]	NC_NT_NR	NC_NOX_SENS_CONF
NC_CBK_EX_NR			

Number of NOx catalyst:	NC_NT_NR	i = 1...NC_NT_NR
Number of NOx sensors:	NC_NOX_SENS_CONF	k = 1...NC_NOX_SENS_CONF
Number of lambda sensors:	NC_CBK_EX_NR	m = 1...NC_CBK_EX_NR

Typical configurations:
[NC_CBK_EX_NR:NC_NT_NR:NC_NOX_SENS_CONF] =[1:1:1], [2:2:2], [2:2:1], [2:1:1]

In some passages logical expressions are calculated using informations of different braches (example: A[1] and A[2]). If this module is compiled for single branched exhaust lines, only the first expression (here: A[1]) should be taken into consideration.

53.19.1 Auxiliary functionalities manager

General information:

This module coordinates the different submodules and thereby prevents undesired parallel activities of the different submodules.

Recurrence: 100 msec

Initialisation:

to be carried out at RESET and ERU to ES

LV_INH_RGN_AD = 0

LV_NT_RGN_REQ_AD = 0

LV_RGN_REQ_AD_VLD = 0

STATE_NOX_PRJ = PASSIVE

LV_NT_O2_STC_ACT = 0

LV_NT_AD_CMPL = 0

LV_NT_STC_MAX_ACT = 0

LV_NT_STC_MAX_AFL_ACT = 0

General

LV_NT_PRJ_ACT = LC_NT_PRJ_ACT **and** **!**LV_INH_NT_PRJ_ACT **and** LV_NT_ACT

Case selection on STATE_NOX_PRJ:

```
any STATE_NOX_PRJ
If      LV_NT_PRJ_ACT = 0
then    STATE_NOX_PRJ = PASSIVE
        LV_RGN_REQ_AD_VLD = 0
endif

STATE_NOX_PRJ PASSIVE
If      LV_NT_PRJ_ACT = 1
and     LV_NT_O2_STC_REQ = 1
and     LV_INH_NT_O2_STC = 0
and     STATE_NOX = WAIT
and     NTL <= C_NTL_MAX_O2_STC
then    STATE_NOX_PRJ = O2_STC
else    stay in PASSIVE
endif

STATE_NOX_PRJ O2_STC
If      STATE_NOX <> WAIT
then    if      MAF_INT_PUC_O2_STC >= C_MAF_INT_PUC_O2_STC
        and     LV_NT_AD_VLD = 1
        then    LV_NT_RGN_REQ_AD = 1
                STATE_NOX_PRJ = FULL_RGN
        else    STATE_NOX_PRJ = PASSIVE
        endif
else    stay in O2_STC
endif

STATE_NOX_PRJ FULL_RGN
If      LV_NT_RGN_REQ_AD = 0
then    STATE_NOX_PRJ = COMMUNICATION
        LV_INH_RGN_AD = 0
else    stay in FULL_RGN
endif

STATE_NOX_PRJ COMMUNICATION
If      LV_NT_AD_CMPL = 0
```

```
then      STATE_NOX_PRJ = PASSIVE
else      stay in COMMUNICATION
endif
```

53.19.2 STATE_NOX_PRJ = PASSIVE

General information:

The state PASSIVE is active during inactive auxiliary functionalities function.

Recurrence: 100 msec

Formula section:

```
LV_INH_RGN_AD=        0
If        LV_RGN_REQ_AD_VLD = 0
then      LV_NT_RGN_REQ_AD = 0
else      LV_NT_RGN_REQ_AD = 1
endif
LV_NT_O2_STC_ACT=   0
T_VLS_DOWN_AFL[i] =   0
T_VLS_DOWN_SECU[i] =0
T_NT_AD_COM =          0
LV_NT_AD_VLD=          1
LV_NT_AD_CMPL=        0
MAF_INT_PUC_O2_STC = 0
```

53.19.3 STATE_NOX_PRJ = O2_STC

General information:

The state O2_STC is active during PUC, if an O2_STC measurement is requested.

Recurrence: 100 ms

Formula section:

```
LV_INH_RGN_AD = 1
LV_NT_O2_STC_ACT = 1

if MAF_INT_PUC_ACT <> 0 then MAF_INT_PUC_O2_STC = MAF_INT_PUC_ACT
if LV_INH_NT_O2_STC = 1 then LV_NT_AD_VLD = 0
```

53.19.4 STATE_NOX_PRJ = FULL_RGN

General information:

The state FULL_RGN enables a full regeneration of the NOx catalyst.

After the full saturation of the NOx storage catalyst a full regeneration is requested. During the full regeneration the integral of the reducing agent (NOx catalyst management) and the Lambda-Sensor switch time is detected. Depending on the results the actual available storage capacity can be determined and the calculated amount of stored sulphur in the NOx catalyst is confirmed or has to be corrected.

The full regeneration should be carried out with a constant rich mixture, this is managed by the flag LV_NT_RGN_REQ_AD. This flag prevents that the lambda setpoint during the regeneration is pulled to a leaner value (handled in the chapter “Lambda Setpoint for Catalyst Regeneration”).

Recurrence: **10** ms

Formula section:

```
call “regeneration conditions control”
If      LV_SENS_AFR_O2[i] = 0
and     STATE_NOX = "REGENERATION"
then    call “determination of downstream sensor switch”
end
if LV_RGN_CDN_AD = 0
then LV_NT_AD_VLD = 0
endif


if STATE_NOX=REGENERATION
then
    call “security time counter”
    if LV_SENS_AFR_O2[i] = 0
    then T_VLS_DOWN_AFL[i] (n) = T_RGN
    else
        if LV_SENS_AFR_O2[i] (n-1) =0
        then NTL_DEC_INT_AD_SNG[i] = NTL_DEC_INT[i]
                NTL_DEC_INT_AD = NTL_DEC_INT_AD_SNG[1] + NTL_DEC_INT_AD_SNG[2]
        endif                                   % if lv_sens_afr_o2[i] (n-1)...
    endif                                       % if lv_sens_afr_o2[i] ...
    if (LV_SENS_AFR_O2[1] = 1) and (LV_SENS_AFR_O2[2] = 1)
    then LV_NT_RGN_REQ_AD = 0                   % reset full rgn bit
    endif
```

```
endif                                        % if state_nox = regeneration
```

53.19.4.1 regeneration conditions control

```
If      VS ≤ C_VS_MIN_RGN_AD
or      N_32 < C_N_32_MIN_RGN_AD
or      MAF_KGH < C_MAF_MIN_RGN_AD
or      LV_PUC = 1
then    LV_RGN_CDN_AD = 0
else    LV_RGN_CDN_AD = 1
endif
```

53.19.4.2 determination of downstream sensor switch

```
case selection on C_STATE_SENS_AFR_AD[i]:
VLS_DOWN:
    #if NC_NT_NR == NC_CBK_EX_NR
        LV_SENS_AFR_O2[i] =          (VLS_DOWN[m] >= C_VLS_DOWN_AFR_O2)
        LV_SENS_AFR_AD_SECU[i] =     (VLS_DOWN[m] >= C_VLS_DOWN_AFR_AD_SECU)
    #else
        LV_SENS_AFR_O2[i] =          ( (VLS_DOWN[1]) >= C_VLS_DOWN_AFR_O2) OR
                                       (VLS_DOWN[2]) >= C_VLS_DOWN_AFR_O2) )
        LV_SENS_AFR_AD_SECU[i] =     ( (VLS_DOWN[1]) >= C_VLS_DOWN_AFR_AD_SECU) OR
                                       (VLS_DOWN[2]) >= C_VLS_DOWN_AFR_AD_SECU) )
    #endif
VLS_NS:
    #if NC_NT_NR == NC_NOX_SENS_CONF
        LV_SENS_AFR_O2[i] =          (VLS_NS[k] >= C_VLS_NOX_SENS_AFR_O2)
        LV_SENS_AFR_AD_SECU[i] =     (VLS_NS[k] >= C_VLS_NOX_SENS_AFR_AD_SECU)
    #else
        LV_SENS_AFR_O2[i] =          (VLS_NS[1] >= C_VLS_NOX_SENS_AFR_O2)
        LV_SENS_AFR_AD_SECU[i] =     (VLS_NS[1] >= C_VLS_NOX_SENS_AFR_AD_SECU)
    #endif
LAMB_NS:
    #if NC_NT_NR == NC_NOX_SENS_CONF
        LV_SENS_AFR_O2[i] =          (LAMB_NS[k] <= C_LAMB_NOX_SENS_AFR_O2)
        LV_SENS_AFR_AD_SECU[i]=      (LAMB_NS[k] <= C_LAMB_NOX_SENS_AFR_AD_SECU)
    #else
        LV_SENS_AFR_O2[i] =          (LAMB_NS[1] <= C_LAMB_NOX_SENS_AFR_O2)
        LV_SENS_AFR_AD_SECU[i]=      (LAMB_NS[1] <= C_LAMB_NOX_SENS_AFR_AD_SECU)
    #endif
VLS_NT_DOWN:
```

```
        #if NC_NT_NR == NC_NOX_SENS_CONF
            LV_SENS_AFR_O2[i] =          (VLS_NT_DOWN[k] >= C_VLS_DOWN_AFR_O2)
            LV_SENS_AFR_AD_SECU[i]=      (VLS_NT_DOWN[k] >= C_VLS_DOWN_AFR_AD_SECU)
        #else
            LV_SENS_AFR_O2[i] =          (VLS_NT_DOWN[1] >= C_VLS_DOWN_AFR_O2)
            LV_SENS_AFR_AD_SECU[i]=      (VLS_NT_DOWN[1] >= C_VLS_DOWN_AFR_AD_SECU)
        #endif
  end case selection
```

53.19.4.3 security time counter

```
if    LV_SENS_AFR_AD_SECU[i] = 0           % security time counter
then  T_VLS_DOWN_SECU[i] = T_RGN
else
   if      T_RGN – T_VLS_DOWN_SECU[i] >= C_T_VLS_DOWN_SECU
   then    LV_NT_AD_VLD = 0                % adaptation cycle no more valid
           LV_NT_RGN_REQ_AD = 0            % safety exit if sensor never reaches thd2
   endif
endif
```

53.19.5 STATE_NOX_PRJ = COMMUNICATION

General information:

The state COMMUNICATION handles the communication with the requesting modules after successfull completion of adaptation cycle.

Recurrence: 100 ms

Formula section:

```
T_NT_AD_COM          = T_NT_AD_COM + TA
LV_NT_AD_CMPL         =
    (T_NT_AD_COM <= C_T_NT_AD_COM)
```

53.19.6 Additional reducing agent mass for complete regeneration after elongated stratified phase

Description:

To compensate deep storage of NOx during extended stratified mode a additional reducing agent mass for completion of regeneration is calculated. Regeneration phase is elongated accordig to duration of lean phase.

Application conditions:

Recurrence: 20ms

Activation:

```
If      LC_RGN_REQ_AD_VLD_ACT = 1
then    "function active"
        at every engine state; independend from STATE_NOX_PRJ
else    "function not active"
        LV_RGN_REQ_AD_VLD = 0
endif
```

Formula section:

```
If      (LV_INH_NT_RGN_REQ = 1
and     STATE_NOX = "LOAD")
or      LV_S_REQ_EGR = 1
then    LV_RGN_REQ_AD_VLD = 1
endif

If      LV_RGN_REQ_AD_VLD = 1
and     STATE_NOX = "REGENERATION"
then    If      LV_NT_SENS_AFR[i] = 1
                % for all i
        then    NTL_DEC_INT_THD_TMP =
                MAX(NTL_DEC_INT_SWI[i]) + IP_NTL_DEC_INT_OFS_T_AFL
                %consideration of higher NTL_DEC_INT_SWI[i]
        else    NTL_DEC_INT_THD_TMP = "FFFFH"
                %REMARK: safety threshold calculation not to reset request
        endif
        If      NTL_DEC_INT[i] ≥ NTL_DEC_INT_THD_TMP
                % for all i
        then    LV_RGN_REQ_AD_VLD = 0
        endif
endif
```

Calibration data:

Name	Dim.	Hex. Limit	Phys. Limit	Resol.	Unit.
LC_NT_PRJ_ACT	1	0...1H	0...1	1	[-]
Logical constant for auxiliary functionalities module activation					
LC_RGN_REQ_AD_VLD_ACT	1	0...1H	0...1	1	[-]
Logical constant for activation of extended regeneration phase after elongated stratified mode					
C_MAF_INT_PUC_O2_STC	1	0...FFFFH	0...2912.66666	0.0444444	[g]
required integral of air mass in PUC before starting an O2_STC measurement					
C_NTL_MAX_O2_STC	1	0...FFFFH	0...10485.6	0.16	[mg]
maximum permissible NOx load for O2_STC measurement					
C_T_VLS_DOWN_SECU	1	0...FFFFH	0...655.35	0.01	[s]
time within sensor signal must pass from 1. thd to 2. thd					
C_T_NT_AD_COM	1	0...FFH	0...25.5	0.1	[s]
time thresholc for duration of communication state					
C_VS_MIN_RGN_AD	1	0...FFH	0...255	1	[km/h]
minimum vehicle speed for use of adaptation value					
C_N_32_MIN_RGN_AD	1	0...FFH	0...8160	32	[rpm]
minimum engine speed for use of adaptation value					
C_MAF_MIN_RGN_AD	1	0...FFFFH	0...2047.96875	0.03125	[kg/h]
minimum mass air flow for use of adaptation value					
C_STATE_SENS_AFR_AD[NC_NT_NR]	1	0H 1H 2H 3H	VLS_DOWN VLS_NS LAMB_NS VLS_NT_DOWN	1	[-]
mode to determine rich exhaust gas downstream NOx trap					
C_VLS_DOWN_AFR_O2	1	0...3FFH	0...4.99511	4.8828e-3	[V]
downstream binary lambda sensor signal threshold for determining rich exhaust gas (osc adaptation)					
C_VLS_DOWN_AFR_AD_SECU	1	0...3FFH	0...4.99511	4.8828e-3	[V]
1. sensor thd. after which 2. thd. must be reached within fixed time					
C_VLS_NOX_SENS_AFR_O2	1	0...578H	-200...1200	1	[mV]
binary NOx sensor signal threshold for determining rich exhaust gas (osc adaptation)					
C_VLS_NOX_SENS_AFR_AD_SECU	1	0...578H	-200...1200	1	[mV]
1. sensor thd. after which 2. thd. must be reached within fixed time					
C_LAMB_NOX_SENS_AFR_O2	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
linear NOx sensor signal threshold for determining rich exhaust gas (osc adaptation)					
IP_NTL_DEC_INT_OFS_T_AFL	8	0...FFFFH	0...10485.6	0.16	[mg]
LDP_T_AFL_IP_NTL_DEC_INT_OFS	8	0...FFFFH	0...1310.7	0.02	[s]
Additional reducing agent mass required after elongated stratified mode					
C_LAMB_NOX_SENS_AFR_AD_SECU	1	0...7FFFH	0...31.99902	0.9766e-3	[-]
1. sensor thd. after which 2. thd. must be reached within fixed time					

53.20 NOx catalyst management – Auxiliary functionalities (Appl.Inc.)

53.20.1 Inhibition of auxiliary functionalities module

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_NT_PRJ_ACT	V/O	0...1H	0...1	1	[-]
flag inhibiting the NOx catalyst auxiliary functionalities (complete)					

FUNCTION DESCRIPTION:

General information:

Using the flag LV_INH_NT_PRJ_ACT the whole functionality of the modul NOx catalyst auxiliary functionalities can be enabled or disabled.

Recurrence: 1000 ms

Initialisation:

LV_INH_NT_PRJ_ACT = 1

Formula section:

LV_INH_NT_PRJ_ACT = LC_INH_NT_PRJ_ACT

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_NT_PRJ_ACT	1	0...1H	0...1	1	[-]
calibration constant inhibiting the NOx catalyst auxiliary functionalities (complete)					

53.20.2 Inhibition of full saturation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_NT_STC_MAX	V/O	0...1H	0...1	1	[-]
flag inhibiting the full saturation					

FUNCTION DESCRIPTION:

General information:

The flag LV_INH_NT_STC_MAX provides the limitation of the full saturation and full regeneration to certain conditions (e.g. operation point).

Recurrence: 1000 ms

Initialisation:

LV_INH_NT_STC_MAX = 1

Formula section:

```
if LC_INH_NT_STC_MAX = 0                                and
   TNT_MDL_L >= C_TNT_MDL_NT_STC_MAX_MIN               and
   TNT_MDL_H <= C_TNT_MDL_NT_STC_MAX_MAX               and
   TNT_MDL_H - TNT_MDL_L <= C_TNT_MDL_NT_STC_MAX_DELTA
  then LV_INH_NT_STC_MAX = 0
  else LV_INH_NT_STC_MAX = 1
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_NT_STC_MAX	1	0...1H	0...1	1	[-]
calibration constant inhibiting the full saturation					
C_TNT_MDL_NT_STC_MAX_MIN	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabling / disabling an full saturation					
C_TNT_MDL_NT_STC_MAX_MAX	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabling / disabling full saturation					
C_TNT_MDL_NT_STC_MAX_DELTA	1	0...FFH	0...1020	4	[°C]
max. deviation between higher and lower NT temp. for enabling a full saturation					

53.20.3 Inhibition of oxygen storage capacity adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_NT_O2_STC	V/O	0...1H	0...1	1	[-]
Flag for inhibition of oxygen storage capacity adaptation					

Input data:

VS	TNT_MDL_H	TNT_MDL_L	

General information:

This submodule provides the limitation of the oxygen storage capacity adaptation to certain conditions (temperatue range, vehicle speed)

Recurrence: 1000 ms

Initialisation:

LV_INH_NT_O2_STC = 1

Formula section:

```
if LC_INH_NT_O2_STC = 0                              and
   VS > = C_VS_O2_STC_MIN                            and
   TNT_MDL_L >= C_TNT_MDL_O2_STC_MIN                 and
   TNT_MDL_H <= C_TNT_MDL_O2_STC_MAX                 and
   TNT_MDL_H - TNT_MDL_L <= C_TNT_MDL_O2_STC_DELTA
  then LV_INH_NT_O2_STC = 0
  else LV_INH_NT_O2_STC = 1
endif
```

Calibration data:

Name	Dim.	Hex. Limit	Phys. Limit	Resol.	Unit.
C_VS_O2_STC_MIN	1	0...FFH	0...255	1	[km/h]
minimum speed for OSC-adaptation					
C_TNT_MDL_O2_STC_MIN	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabling / disabling an O2-STC adaptation					
C_TNT_MDL_O2_STC_MAX	1	0...FFH	0...1020	4	[°C]
Temperature threshold for enabling / disabling an O2-STC adaptation					
C_TNT_MDL_O2_STC_DELTA	1	0...FFH	0...1020	4	[°C]
max. deviation between higher and lower NT temp. for enabling an O2-STC adaptation					
LC_INH_NT_O2_STC	1	0...1H	0...1	1	[-]
calibration constant inhibiting oxygen storage capacity adaptation					

53.21 NOx catalyst oxygen storage capacity adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_NT_O2_STC_REQ	V/O/S	0...1H	0...1	1	[-]
bit for requesting an OSC adaptation cycle					
LV_NT_O2_STC_LIM	V/O/S	0...1H	0...1	1	[-]
bit for limitation of OSC adaptation cycle					
LV_NT_O2_STC_VLD	V/O/S	0...1H	0...1	1	[-]
Flag for valid status of OSC adaptation measurement					
NT_O2_STC_AD	V/O/S	0...FFH	0...1.99218	0.0078125	[-]
adaptation factor for NOx catalyst oxygen storage capacity					
NT_O2_STC_OFS	V/O/S	8000...7FFFH	-10485.76... 10485.44	0.32	[mg]
adaptation offset for NOx catalyst oxygen storage capacity					
DIST_O2_STC_OLD	V/O/S	0...FFFFH	0...524280	8	[km]
mileage of last O2_STC adaptation					

Input data:

DIST_NT	LV_NT_AD_CMPL	LV_NT_O2_STC_ACT	NTL_DEC_INT_AD
NT_O2_STC_BAS	LV_NT_AD_VLD		

FUNCTION DESCRIPTION:

General information:

The oxygen storage capacity adaptation allowes an adaptation of the NOx catalyst oxygen storage capacity which will decrease within time due to thermal aging effects.

The adaptation can be taken into consideration as factor or as offset value; this module offers both possibilities.

The main adaptation value is the factor, the offset is calculated using the factor.

Description:

Application conditions:

Initialisation:

```
If      there are no stored values or checksum error has occured or scantool reset
then    NT_O2_STC_AD = 1; LV_NT_O2_STC_LIM = 1, set all other default values (=0)
else    use previously stored values
endif
```

Recurrence: 1000 ms

Activation: at every engine state

Formula section:

```
If      LC_NT_O2_STC_AD_MAN=1
then    LV_NT_O2_STC_REQ = 0
        LV_NT_O2_STC_VLD = 1
        NT_O2_STC_AD = C_NT_O2_STC_AD_MAN
        NT_O2_STC_OFS = (NT_O2_STC_AD-1) * NT_O2_STC_BAS
else    % O2_STC request coordination
        If      (DIST_NT - DIST_O2_STC_OLD) >= C_DIST_O2_STC
        or      LC_NT_O2_STC_REQ = 1
        then    LV_NT_O2_STC_REQ = 1
        endif
        If      [(DIST_NT - DIST_O2_STC_OLD) < C_DIST_O2_STC_VLD
        and     LV_NT_O2_STC_LIM = 0]
        or      LC_NT_O2_STC_VLD = 1
        then    LV_NT_O2_STC_VLD = 1
        else    LV_NT_O2_STC_VLD = 0
        endif
        % O2_STC-adaptation-factor rsp. -offset calculation
        If      LV_NT_AD_CMPL = 0 -> 1
        and     LV_NT_AD_VLD = 1                 %ad.cyc. completed
        and     LV_NT_O2_STC_ACT = 1             %ad.cyc. was osc meas.
        then    DIST_O2_STC_OLD = DIST_NT
                NT_O2_STC_OFS_TMP = (NT_O2_STC_AD - 1) * NT_O2_STC_BAS
                NT_O2_STC_OFS =                  % gradient limitation
                MAX(MIN(NTL_DEC_INT_AD – NT_O2_STC_BAS,
                NT_O2_STC_OFS_TMP + C_NT_O2_STC_DELTA),
                NT_O2_STC_OFS_TMP - C_NT_O2_STC_DELTA)
                If      NT_O2_STC_OFS = NTL_DEC_INT_AD – NT_O2_STC_BAS
                        %no limitation
                or      LC_NT_O2_STC_REP_OFF = 1
                        % no repetition of ad.cyc. when limited
                then    LV_NT_O2_STC_REQ = 0         % reset request bit
                        LV_NT_O2_STC_LIM = 0
                else    LV_NT_O2_STC_LIM = 1
```

```
            endif                                       % if nt_o2_stc_ofs_...
            NT_O2_STC_AD = 1 + NT_O2_STC_OFS / NT_O2_STC_BAS
        endif                                       % if lv_nt_ad_cmpl_ctl...
    endif                                           % if lc_nt_o2_stc_ad_man...
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_NT_O2_STC_AD_MAN	1	0...FFH	0...1.99218	0.0078125	[-]
manual initialsisation value for oxygen storage capacity adaptation value					
C_DIST_O2_STC	1	0...FFFFH	0...524280	8	[km]
mileage distance to be passed before next O2_STC measurement					
C_DIST_O2_STC_VLD	1	0...FFFFH	0...524280	8	[km]
mileage distance to be passed before O2_STC measurement is not valid anymore					
C_NT_O2_STC_DELTA	1	0...7FFFH	0...10485.44	0.32	[mg]
permissible deviation between old and new osc value					
LC_NT_O2_STC_REQ	1	0...1H	0...1	1	[-]
logical constant for manually requesting an O2_STC adaptation cycle					
LC_NT_O2_STC_VLD	1	0...1H	0...1	1	[-]
logical constant for manually setting of valid bit of O2_STC value					
LC_NT_O2_STC_AD_MAN	1	0...1H	0...1	1	[-]
logical constant for manual initialisation of NT_O2_STC_AD					
LC_NT_O2_STC_REP_OFF	1	0...1H	0...1	1	[-]
logical constant for disabling a repetition of osc-measurement in case of gradient limitation					

53.22 NOx catalyst oxygen storage capacity adaptation (Appl.Inc.)

53.22.1 General

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
DIST_NT	V/O	0...FFFFH	0...524280	8	[km]
mileage counter for NOx trap purposes					

Input data:

CTR_KM_CAN			

FUNCTION DESCRIPTION:

General information:

The range and resolution of CTR_KM_CAN and DIST_NT differ: both are words, resolution of CTR_KM_CAN (BMW): 10 km, DIST_NT: 8km

Recurrence: **1000** ms

Initialisation: DIST_NT = CTR_KM_CAN *at reset*
% CTR_KM_CAN already read from NVMY

Activation: *at every engine state*

Formula section:

DIST_NT = CTR_KM_CAN % resolution transformation

53.23 NOx catalyst aging model (Placeholder without functionality)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_NT_AGI_MDL	O/V	0...FFFFFFFFH	0...0.99999	0.2328e-9	[-]
Modeled aging of NT					
FAC_NT_AGI_MDL_THERMO	O/V	0...FFFFH	0...0.99998	0.0153e-3	[-]
Converted modeled aging factor to thermo aging factor					

Export Actions:

ACTION_NOXM_CleanNTAdaptAgiMdl()
External service for Initiaöization of Modelled Aging values
ACTION_NOXM_WriteAgiMExtAdj()
Reset of NSC modelled value using external value

FUNCTION DESCRIPTION:

General information:

This modul has no functionality. It provides only all existing interfaces for other moduls.

Application conditions:

Initialisation: at reset set all output values to 1 (means new NOx catalyst)

Recurrence: no

Activation:

Action definition:

Both actions have no functionality.

53.24 NOx storage catalyst diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_END_DIAG_NT_AGI	O/V	0...1H	0...1	1	-
End of NOx catalyst aging diagnosis					
LV_ERR_NT_AGI	O/V	0...1H	0...1	1	-
Error flag of NOx catalyst aging diagnosis					
LV_END_DIAG_NT_SO2P	O/V	0...1H	0...1	1	-
End of NOx catalyst missing desulfation diagnosis					
LV_ERR_NT_SO2P	O/V	0...1H	0...1	1	-
Error flag of NOx catalyst missing desulfation diagnosis					
ERR_SYM_NT_AGI	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx catalyst aging diagnosis					
ERR_SYM_NT_SO2P	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	-
Error symptom for the NOx catalyst missing desulfation diagnosis					
LV_CDN_DIAG_NT_AGI	V	0...1H	0...1	1	-
Diagnosis condition of NOx catalyst aging diagnosis					
LV_CDN_DIAG_NT_SO2P	V	0...1H	0...1	1	-
Diagnosis condition of NOx catalyst missing desulfation diagnosis					

Input data:

LV_INH_DIAG_NT_AGI	LV_INH_DIAG_NT_SO2P	LV_NT_AFS_REQ_AGI_TMP_3	LV_ST_END
NC_IDX_DIAG_NT_AGI	NC_IDX_DIAG_NT_SO2P	NT_AGI_SUL	NT_AGI_THERMO
NT_SUL	NT_SUL_H		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_DEC_NT_AGI	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx catalyst aging diagnosis					
C_ABC_DEC_NT_SO2P	1	0...FFH	0...255	1	-
Anti bounce counter decrement for the NOx catalyst missing desulfation diagnosis					
C_ABC_INC_NT_AGI	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx catalyst aging diagnosis					
C_ABC_INC_NT_SO2P	1	0...FFH	0...255	1	-
Anti bounce counter increment for the NOx catalyst missing desulfation diagnosis					
C_ABC_MAX_NT_AGI	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx catalyst aging diagnosis					
C_ABC_MAX_NT_SO2P	1	1...FFH	1...255	1	-
Anti bounce counter maximum for the NOx catalyst missing desulfation diagnosis					
C_FAC_NT_AGI_ERR_SO2P	1	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx trap aging factor threshold for the missing desulfation error					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_SUL_H_NT_MIN_ERR_SO2P	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDPM_NT_AGI_THERMO_IP_SUL_ERR	6	0...FFFFH	0...0.99998474	1.52588E-5	-
NOx Trap sulphur loading threshold (for high sulphured fuel) for missing desulfation error					
IP_SUL_NT_MAX_ERR_SO2P	6	0...FFFFH	0...1.04856E+4	0.16	mg
LDPM_NT_AGI_THERMO_IP_SUL_ERR	6	0...FFFFH	0...0.99998474	1.52588E-5	-
Sulphur model threshold for missing desulfation error					

Import actions:

ACTION_ERRM_FilterSymptom(IN <IDX_DIAG>, IN <LV_CDN_DIAG>, IN <ERR_SYM>, IN <ABC_INC>, IN <ABC_DEC>, IN <ABC_MAX>, OUT <LV_ERR>)
This action computes the elementary anti-bounce filter
ACTION_ERRM_GetLvEndDiag(IN <IDX_DIAG>, OUT <LV_END_DIAG>)
Action that returns the diagnosis end information
ACTION_ERRM_GetLvErr(IN <IDX_DIAG>, OUT <LV_ERR>)
Action that returns the status of the debounced failure

53.24.1 General information

If the stratified combustion mode is irreversible forbidden due to NOx catalyst aging, a failure mode entry is done.

If the NOx trap is aged reversible by sulfur, a failure mode entry is done.

Application Condition

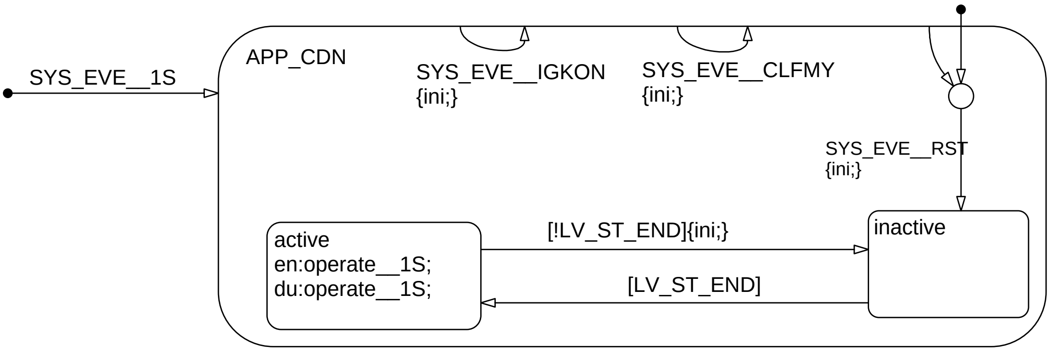
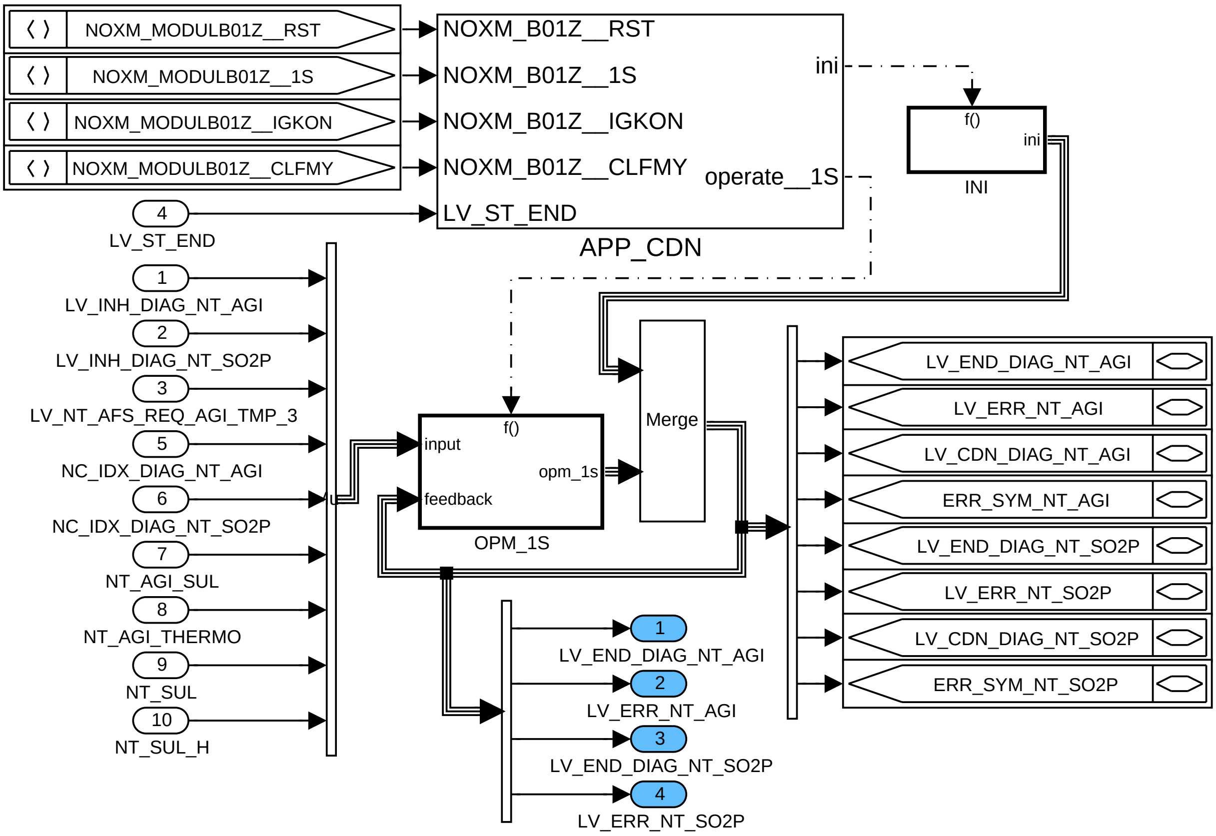


Figure 227 NOXM_MODULB01Z/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 30−Jan−2006

Figure 228 NOXM_MODULB01Z

53.24.1.1 Initialization at reset and deactivation

These actions initialized the diagnostic data according filtering configuration :

ACTION_ERRM_GetLvErr(IN<NC_IDX_DIAG_XX>, OUT<LV_ERR_XX>)

ACTION_ERRM_GetLvEndDiag(IN<NC_IDX_DIAG_XX>, OUT<LV_END_DIAG_XX>)

53.24.1.2 Calculation at 1 s

1.1.2.1 Enabling of aging diagnosis

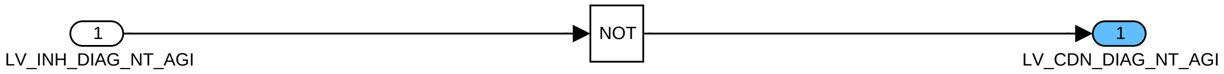


Figure 229 NOXM_MODULB01Z/ OPM_1S/ CLC1

1.1.2.2 Error symptom for aging diagnosis

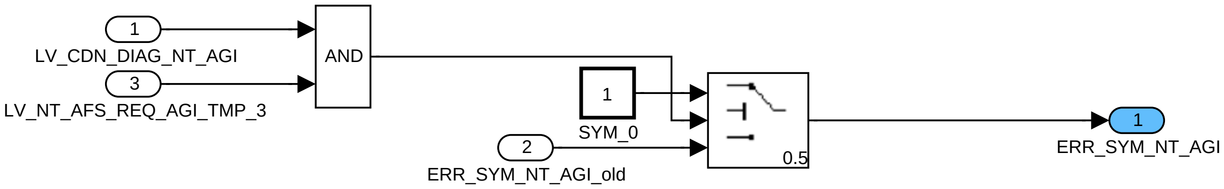


Figure 230 NOXM_MODULB01Z/ OPM_1S/ CLC2

1.1.2.3 Error management for aging diagnosis

LV_ERR_NT_AGI will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NT_AGI will be get by ACTION_ERRM_GetLvEndDiag.

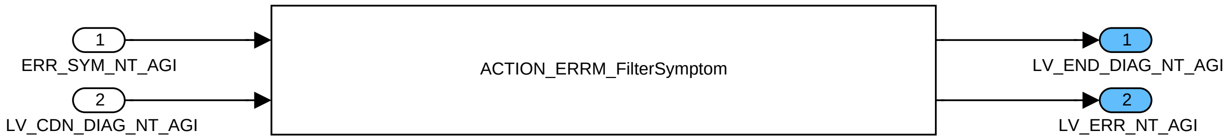


Figure 231 NOXM_MODULB01Z/ OPM_1S/ CLC3

1.1.2.4 Enabling of missing desulfation diagnosis

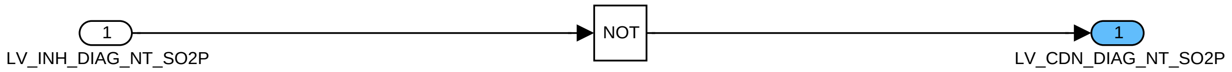


Figure 232 NOXM_MODULB01Z/ OPM_1S/ CLC4

1.1.2.5 Error symptom for missing desulfation diagnosis

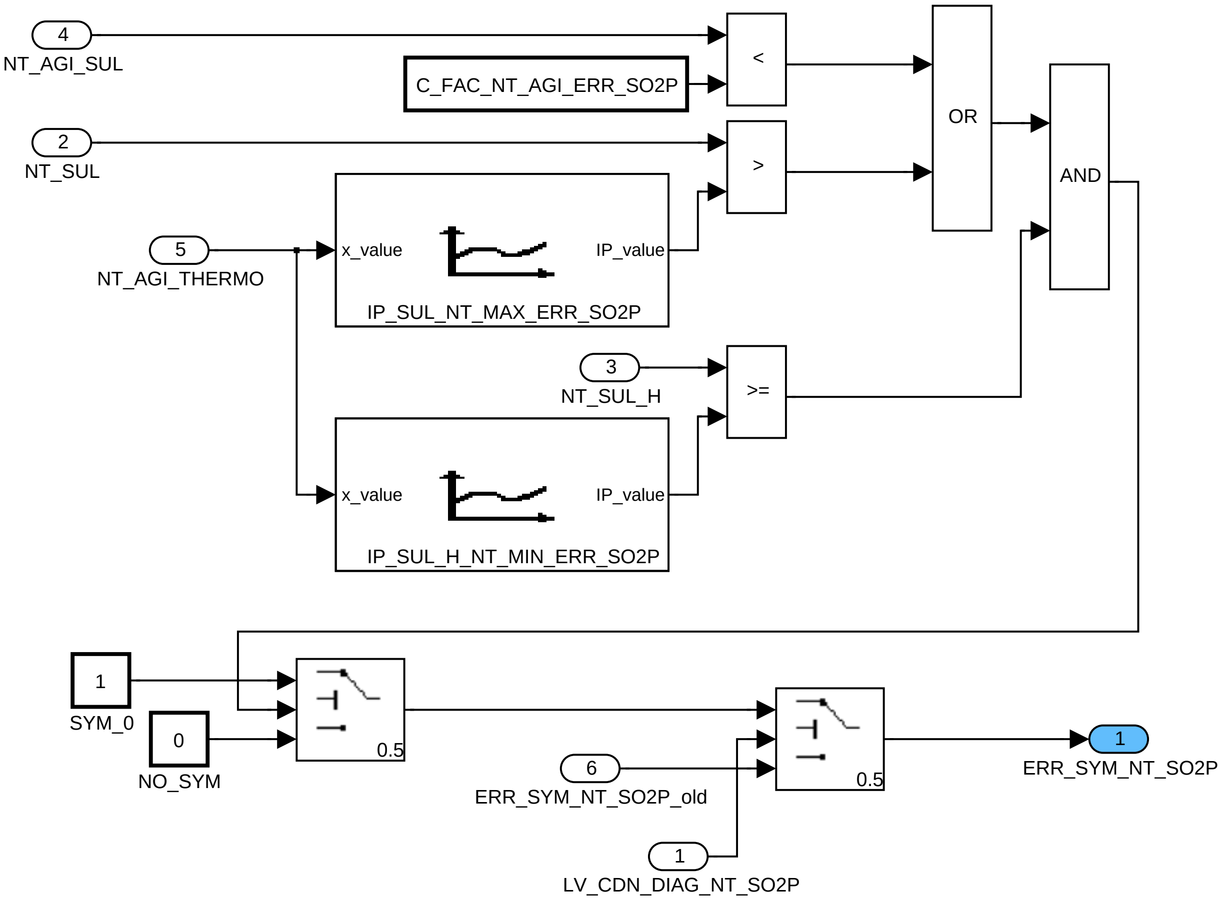


Figure 233 NOXM_MODULB01Z/ OPM_1S/ CLC5

1.1.2.6 Error management for missing desulfation diagnosis

LV_ERR_NT_AGI will be get by ACTION_ERRM_GetLvErr and LV_END_DIAG_NT_AGI will be get by ACTION_ERRM_GetLvEndDiag.

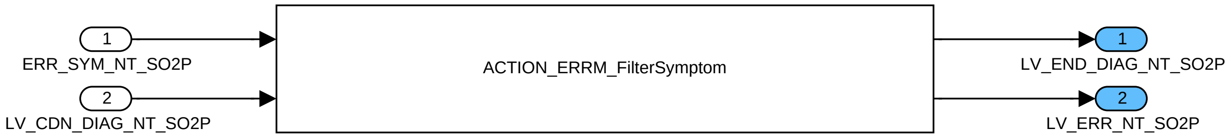


Figure 234 NOXM_MODULB01Z/ OPM_1S/ CLC6

53.25 NOx storage catalyst diagnosis (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_INH_DIAG_NT_AGI	V/O	0...1H	0...1	1	[-]
Inhibition of NOx catalyst aging diagnosis					
LV_INH_DIAG_NT_SO2P	O/V	0...1H	0...1	1	[-]
Inhibition of NOx catalyst diagnosis for missing desulfation					

Input data:

LV_ST_END			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_STATE_INH_NT_AGI	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx catalyst aging diagnosis					
C_STATE_INH_NT_SO2P	1	0H 1H 2H	AUTO DISABLE ENABLE	1	[-]
Switch for selection of inhibition mode of NOx catalyst diagnosis for missing desulfation					

General information:

Both diagnoses will be inhibited, if any used input signal is not valid. Additionally, the diagnosis may be enabled or disabled without any cross dependencies.

Application conditions:

Initialisation: at reset and at deactivation:

LV_INH_DIAG_NT_AGI = 1

LV_INH_DIAG_NT_SO2P = 1

Recurrence: 1 s

Activation: LV_ST_END = 1

Deactivation: LV_ST_END = 0

Formula section:

```
case selection on C_STATE_INH_NT_AGI
   C_STATE_INH_NT_AGI = 0 ('AUTO'):
      LV_INH_DIAG_NT_AGI = 0                    (no cross dependencies)

   C_STATE_INH_NT_AGI = 1 ('DISABLE'):
      LV_INH_DIAG_NT_AGI = 1

   C_STATE_INH_NT_AGI = 2 ('ENABLE'):
      LV_INH_DIAG_NT_AGI = 0
end case selection

case selection on C_STATE_INH_NT_SO2P
   C_STATE_INH_NT_SO2P = 0 ('AUTO'):
      LV_INH_DIAG_NT_SO2P = 0                   (no cross dependencies)

   C_STATE_INH_NT_SO2P = 1 ('DISABLE'):
      LV_INH_DIAG_NT_SO2P = 1

   C_STATE_INH_NT_SO2P = 2 ('ENABLE'):
      LV_INH_DIAG_NT_SO2P = 0
end case selection
```

Configuration for diagnostic symptoms:

Diagnostic NT_AGI	Symptom description	Symptom	Filter type
NOx storage catalyst diagnosis	NOx storage capacity too low	SYM_0	STD
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic NT_AGI	Symptoms	Ref.	DTC Symptom CARB	DTC Symptom Customer	DTC Global CARB	DTC Global Customer	Failure class A/B	Readiness code
NOx storage catalyst diagnosis	NOx storage capacity too low	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

Diagnostic **NT_SO2P**	**Symptom description**	**Symptom**	**Filter type**
NOx storage catalyst diagnosis	NOx trap sulfur load too high	SYM_0	STD
		SYM_1	
		SYM_2	
		SYM_3	

Diagnostic **NT_SO2P**	**Symptoms**	**Ref.**	**DTC Symptom CARB**	**DTC Symptom Customer**	**DTC Global CARB**	**DTC Global Customer**	**Failure class A/B**	**Readiness code**
NOx storage catalyst diagnosis	NOx trap sulfur load too high	*SYM_0*						
		SYM_1						
		SYM_2						
		SYM_3						

53.26 Cus adap module: NOXM

53.26.1 Outputs for BMW functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
F_grad_katalt	V/O	8000...0H	-0.1...0	3.0518e-6	[1/km]
Nox-Kat-Alterungsgradient					
F_katalt	V/O	0...FFFFH	0...0.99998	0.0153e-3	[-]
Nox-Kat-Alterung					
F_katalt_tabg	V/O	0...FFFFH	0...0.99998	0.0153e-3	[-]
Thermische Nox-Kat-Alterung					

Input data:

NT_AGI_THERMO_GRD	FAC_NT_AGI_LIM	NT_AGI_THERMO	ECU_STATE

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit powerlatch:

F_grad_katalt = NT_AGI_THERMO_GRD
F_katalt = FAC_NT_AGI_LIM
F_katalt_tabg = NT_AGI_ THERMO

Recurrence : 200ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
If      ECU_STATE = not("PWL")
  Then
        F_grad_katalt       = NT_AGI_THERMO_GRD
        F_katalt        = FAC_NT_AGI_LIM
        F_katalt_tabg  = NT_AGI_ THERMO
end
```

54 On-Board diagnostic communication

Table of Contents

Index

54.1 OBDC Configuration Data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_VAR_SAP_SAE	O/V	0...FFH	0...255	1	[-]
SAE status of secondary air pump (0 = not supported, >0 = supported)					
STATE_EVAP_SAE	O/V	0...FFH	0...255	1	[-]
SAE status of EVAP monitor (0 = not supported, >0 = supported)					
STATE_PSN_LS_SAE	O/V	0...FFH	0...255	1	[-]
Position of oxygen sensors for OBD (2 Banks 4 Sensors)					
STATE_PSN_LS_1_SAE	O/V	0...FFH	0...255	1	[-]
Position of oxygen sensors for OBD (4 Banks 2 Sensors)					

Input data:

PSN_LS	LV_VAR_SAP	LV_CONF_DMTL	

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_STATE_OBD_SRV	1	0...FFFFH	0...65535	1	[-]
Word which indicates the EOBD services supported by the project					
NC_STATE_PID_13_PSN_LS	1	0...FFH	0...255	1	[-]
Configuration of oxygen sensors (2 Banks 4 Sensors)					
NC_STATE_PID_1D_PSN_LS	1	0...FFH	0...255	1	[-]
Configuration of oxygen sensors (4 Banks 2 Sensors)					
NC_STATE_OBD_CONF	1	0...FFH	0...255	1	[-]
Byte which indicates the configuration of EOBD/ARB					
NLC_STATE_VAR_SAP	1	0...1H	0...1	1	[-]
Configuration of Commanded Secondary Air Status					
NLC_STATE_EVAP	1	0...1H	0...1	1	[-]
Configuration of EVAP Monitor					
NLC_STATE_PRJ_SPC_CAL	1	0...1H	0...1	1	[-]
Indicates the usage of project specific calibration data					

FUNCTION DESCRIPTION:

General information:

This module is for configuration of the On-Board-Diagnosis Communication functionality.

54.1.1 Configuration of OBD services: NC_STATE_OBD_SRV

0 : the SERVICE is not supported
1 : the SERVICE is supported

NC_STATE_OBD_SRV	Supported EOBD SERVICEs from 01h – 09h	
BIT	**Function**	**Supported**
0	SERVICE 01h : Request current powertrain diagnostic data	**1**
1	SERVICE 02h : Request powertrain freeze frame data	**1**
2	SERVICE 03h : Request emission-related diagnostic trouble codes	**1**
3	SERVICE 04h : Clear/reset emission-related diagnostic information	**1**
4	SERVICE 05h : Request oxygen sensor monitoring test results	0
5	SERVICE 06h : Request on-board monitoring test results for specific monitored systems	**1**
6	SERVICE 07h : Request emission-related diagnostic trouble codes detected during current or last completed driving cycle	**1**
7	SERVICE 08h : Request control of on-board system, test or component	0
8	SERVICE 09h : Request vehicle information	**1**
9	RESERVED	0
10	RESERVED	0
11	RESERVED	0
12	RESERVED	0
13	RESERVED	0
14	RESERVED	0
15	RESERVED	0

54.1.2 Autolearning and Microcontroller configuration: NC_STATE_OBD_CONF

NC_STATE_OBD_CONF		Supported EOBD/ARB Configuration
BIT	**Function**	**Configured**
0	Internal format of the microcontroller	0
1	Sensor configuration	**1**
2	Autolearning	**1**
3	RESERVED	0
4	RESERVED	0
5	RESERVED	0
6	RESERVED	0
7	RESERVED	0

Internal format of the microcontroller	
Value	**Description**
0	The internal format of the microcontroller is INTEL
1	The internal format of the microcontroller is MOTOROLA

Sensor configuration	
Value	**Description**
0	BINARY Sensor configuration
1	LINEAR Sensor configuration

Autolearning – for O2 sensors configuration	
Value	**Description**
0	Without autolearning
1	With autolearning

54.1.3 Project specific calibration data: NLC_STATE_PRJ_SPC_CAL

Project specific calibration data - NLC_STATE_PRJ_SPC_CAL	
Value	**Description**
0	EOBD is configured by generic configuration data
1	EOBD is configured by project specific calibration data

Formula section:

NLC_STATE_PRJ_SPC_CAL = 1

54.1.4 Configuration of oxygen sensors (2 Banks 4 Sensors): NC_STATE_PID_13_PSN_LS

NC_STATE_PID_13_PSN_LS		
BIT	**Function**	**Value**
0	Bank 1 Sensor 1 (upstream 1)	**1**
1	Bank 1 Sensor 2 (downstream 1)	**1**
2	Bank 1 Sensor 3	0
3	Bank 1 Sensor 4	0
4	Bank 2 Sensor 1 (upstream 2)	**1**
5	Bank 2 Sensor 2 (downstream 2)	**1**
6	Bank 2 Sensor 3	0
7	Bank 2 Sensor 4	0

Each bit indicates the presence or absence of the oxygen sensors for each cylinder bank, the first sensor being the nearest of the engine.

0 : sensor not present

1 : sensor present

54.1.5 Position of oxygen sensors for OBD (2 Banks 4 Sensors): STATE_PSN_LS_SAE

Formula section:

```
IF      (NC_STATE_OBD_CONF & 0x04)
THEN    With autolearning
        STATE_PSN_LS_SAE = PSN_LS          //set to the value defined by the project
ELSE    STATE_PSN_LS_SAE = NC_STATE_PID_13_PSN_LS
ENDIF
```

54.1.6 Configuration of oxygen sensors (4 Banks 2 Sensors): NC_STATE_PID_1D_PSN_LS

NC_STATE_PID_1D_PSN_LS		
BIT	**Function**	**Value**
0	Bank 1 Sensor 1 (upstream 1)	0
1	Bank 1 Sensor 2 (downstream 1)	0
2	Bank 2 Sensor 1 (upstream 2)	0
3	Bank 2 Sensor 2 (downstream 2)	0
4	Bank 3 Sensor 1	0
5	Bank 3 Sensor 2	0
6	Bank 4 Sensor 1	0
7	Bank 4 Sensor 2	0

Each bit indicates the presence or absence of the oxygen sensors for each cylinder bank, the first sensor being the nearest of the engine.

0 : sensor not present

1 : sensor present

54.1.7 Position of oxygen sensors for OBD (4 Banks 2 Sensors): STATE_PSN_LS_1_SAE

Formula section:

STATE_PSN_LS_1_SAE = NC_STATE_PID_1D_PSN_LS

54.1.8 Configuration of secondary air pump: NLC_STATE_VAR_SAP

Configuration of secondary air pump – NLC_STATE_VAR_SAP	
Value	**Description**
0	Secondary air pump **NOT** supported
1	Secondary air pump supported

Formula section:

NLC_STATE_VAR_SAP = 1

54.1.9 Position of Secondary Air Pump for OBD: STATE_VAR_SAP_SAE

Formula section:

```
IF      (NC_STATE_OBD_CONF & 04h)
THEN    With autolearning
        STATE_VAR_SAP_SAE (bit 0) = LV_VAR_SAP
ELSE    STATE_VAR_SAP_SAE is set to NLC_STATE_VAR_SAP
ENDIF
```

54.1.10 Configuration of EVAP Monitor: NLC_STATE_EVAP

Configuration of EVAP Monitor – NLC_STATE_EVAP	
Value	**Description**
0	EVAP Monitor **NOT** supported
1	EVAP Monitor supported

Formula section:

NLC_STATE_EVAP = 1

54.1.11 Value for EVAP Monitor for OBD: STATE_EVAP_SAE

Formula section:

```
IF      (NC_STATE_OBD_CONF & 04h)
THEN    With autolearning
        STATE_EVAP_SAE (bit 0) = LV_CONF_DMTL
ELSE    STATE_EVAP_SAE = NLC_STATE_EVAP
ENDIF
```

54.2 AGGR Adaptation for OBDC

54.2.1 Initialization of secondary-air variables

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
SUM_AFL_VLS_DIAG_SA_1_SAE	V/O	0...FFFFH	0...65535	1	[-]
Counter of lean events within the secondary air diagnosis					
SUM_AFL_VLS_DIAG_SA_2_SAE	V/O	0...FFFFH	0...65535	1	[-]
Counter of lean events within the secondary air diagnosis					
SAF_DIAG_MIN_SAE	O/S	0...FF00H	0...255	3.9063e-3	[kg/h]
Mean diagnosis value of minimum secondary air flow for SAE					
SAF_DIAG_MAX_SAE	O/S	0...FF00H	0...255	3.9063e-3	[kg/h]
Mean diagnosis value of maximum secondary air flow for SAE					

FUNCTION DESCRIPTION:

General information:

Secondary air system diagnosis is not supported in MSD projects, thus all needed variables must be stubbed.

Application conditions:

Initialisation: at reset all variables = 0

Recurrence: -

Activation: -

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_SA_SWI_ACQ	1	0...1H	0...1	1	[-]
Logical constant for the decision SAF_KGH through model value (=0) or SAFM is used (=1)					
C_SUM_AFL_VLS_MIN_DIAG_SA_1	1	0...FFFFH	0...65535	1	[-]
Threshold of lean events within the secondary air diagnosis					

54.2.2 Calculation of non-implemented variables

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VLS_COR_LSL[NC_CBK_EX_NR]	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Corrected output signal of lambda sensor					

Input data:

VLS_UP_COR[NC_CBK_EX_NR]			

FUNCTION DESCRIPTION:

General information:

VLS_COR_LSL[i] is not supported in MSD projects, thus all needed variables must be calculated.

Application conditions:

Initialisation: at reset = 0

Recurrence: 10ms

Activation: every engine operating state

Formula section:

VLS_COR_LSL[i] = VLS_UP_COR[i] *//calculation is done in physical way*

54.3 Mode 01h - Request current powertrain diagnostic data

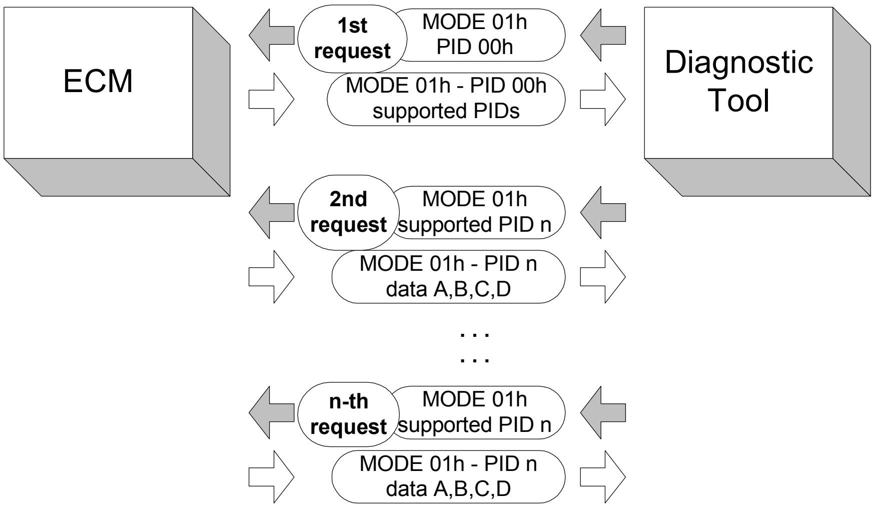
54.3.1 Functional description for ISO 14230-4

The purpose of this mode is to allow access to current emission-related data values including analog inputs and outputs, digital inputs and outputs, system status information, and calculated parameter values.

Each request contains a Parameter IDendification (PID) which indicates to the ECM the specific information to be transmitted. The information transmitted by the ECM must be the latest ones, all acquisitions shall correspond to actual measurements and not to values selected by the ECM for the limp home mode.

Not all PIDs are applicable or supported by all systems. Then, all ECUs must at least answer to PID 00h by sending back the list of the PIDs used.

You can find the exhaustive list of generic PIDs definition in chapter “PID description for mode 01h”. The specific list of supported PIDs can be found in chapter “Emission-related software and system configuration”.



Request message definition (read supported PIDs)

Bytes	Parameter name	Hex value
#01	**Request current powertrain diagnostic data**	01
#02	PID	00-20-40-60-80-A0-C0-E0

Positive reponse message definition (read supported PIDs)

Bytes	Parameter name	Cvt	Hex value
#01	**Request current powertrain diagnostic data positive response**	M	41
#02	PID	M	00-20-40-60-80-A0-C0-E0
#03	Data A	M	XX
#04	Data B	M	XX
#05	Data C	M	XX
#06	Data D	M	XX

Request message definition (read PID value)

Bytes	Parameter name	Hex value
#01	**Request current powertrain diagnostic data**	01
#02	PID	XX

Positive response message (read PID value)

Bytes	Parameter name	Cvt	Hex value
#01	**Request current powertrain diagnostic data positive response**	M	41
#02	PID	M	XX
#03	Data A	M	XX
#04	Data B (optional)	C	XX
#05	Data C (optional)	C	XX
#06	Data D (optional)	C	XX

C =Conditional - Data B -D depend on selected PID value
M = Mandatory

Negative response message

Bytes	Parameter name	Hex value
#1	**Negative response**	7F
#2	**Request current powertrain diagnostic data**	01
#3	Error code = Service not supported Sub function not supported - Invalid format	11 12

Error code

Error code	Cause
Service not supported	The requested mode is not supported (vehicle is not an EOBD or OBD2 system)
Sub function not supported - Invalid format	The requested PID is not supported or the length of the request is incorrect

54.4 Mode 02h - Request powertrain freeze frame data

Import:

ACTION_ERRM_ReadFrfByDtc	**(** **IN <TypeOfFF>** **IN <FFidentifier>** **IN <DtcIdentifier>** **IN <LevelOfDTC>** **INOUT <Frf>** **OUT <ResultFrf>** **)**
This API returns a Freeze Frame related to the DTC given in parameter.	

54.4.1 Functional description for ISO 14230-4

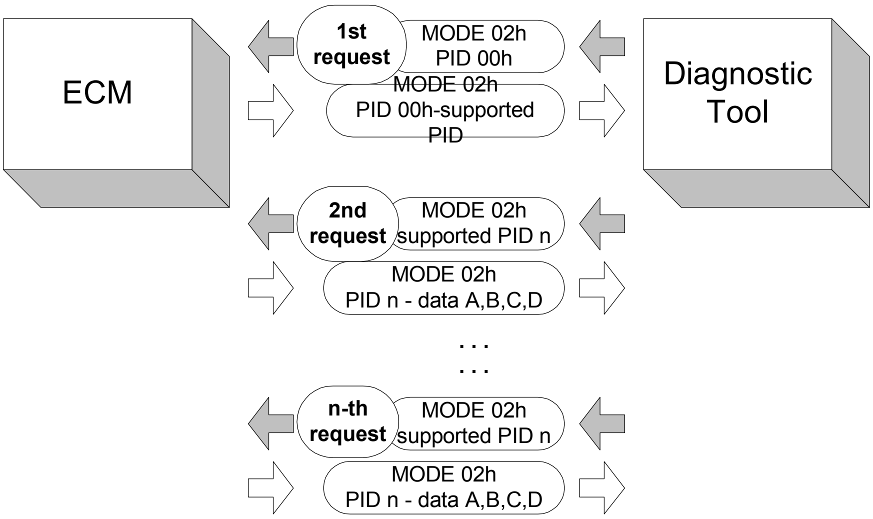
The purpose of this mode is to allow access to emission-related data values which were stored during the freeze frame required by OBD-II regulations.

Each request contains a Parameter IDendification (PID) which indicates to the ECM the specific information to be transmitted.
The information to be transmitted is the freeze frame data which has been stored by the ECM at the failure occurrence. Like mode 01h, no limp home value shall be transmitted. The definition of the failure context is given by the frame number - in this case, we only select one failure context which is defined by the frame number 00h.

Not all PIDs are applicable or supported by all systems. Then, all ECUs must at least answer to PID 00h by sending back the list of the PIDs used.

PID 02h is the DTC that caused the freeze frame data to be stored. If freeze frame data is not stored in the ECM, the system should report 00 00h as the DTC. Any data reported when the stored DTC is 00 00h may not be valid.

You can find the exhaustive list of generic PIDs definition, scaling information, and display formats for the required freeze frame in chapter “PID description for mode 02h”.
The specific list of supported PIDs for current project can be found in chapter “Emission-related software and system configuration”.



54.4.2 Message data bytes

Request message (read supported PIDs)

Bytes	Parameter name	Cvt	Hex value
#01	**Request powertrain freeze frame data**	**M**	**02**
#02	PID	M	00-20-40-60-80-A0-C0-E0
#03	Frame number	M	00

Positive response message (read supported PIDs)

Bytes	Parameter name	Cvt	Hex value
#01	**Request powertrain freeze frame data positive response**	**M**	**42**
#02	PID	M	00-20-40-60-80-A0-C0-E0
#03	Frame number	M	00
#04	Data A	M	XX
#05	Data B	M	XX
#06	Data C	M	XX
#07	Data D	M	XX

Request message (read PID value)

Bytes	Parameter name	Cvt	Hex value
#01	**Request powertrain freeze frame data**	**M**	**02**
#02	PID	M	XX
#03	Frame number	M	00

Positive response message (read PID value)

Bytes	Parameter name	Cvt	Hex value
#01	**Request powertrain freeze frame data positive response**	**M**	**42**
#02	PID	M	XX
#03	Frame number	M	00
#04	Data A	M	XX
#05	Data B	C	XX
#06	Data C	C	XX
#07	Data D	C	XX

C = Conditional Data B -D depend on selected PID value
M = Manadatory

Negative response message

Bytes	Parameter name	Hex value
#1	**Negative response**	7F
#2	**Request current powertrain diagnostic data**	02
#3	Error code = Service not supported Sub function not supported - Invalid format Condition not correct General reject	11 12 22 10

Error code

Error code	Cause
Service not supported	The requested mode is not supported (vehicle is not an EOBD or OBD2 system)
Sub function not supported - Invalid format	The requested PID is not supported The length of the request is incorrect Freeze frame number is different of 00h
Condition not correct	no freeze frame stored
General reject	Software Problem

Formula section

ACTION_ERRM_ReadFrfByDtc

```
(
IN <LAW>
IN <FIRST>
IN <NO_DTC>
IN <LAW>
INOUT <Frf>
OUT <ResultFrf>
)
```

```
IF      ResultFrf = BUFFER_FULL
        Negative response with "General reject"
ELSE
        IF      ResultFrf = FRF_PRESENT THEN
                Positive response
        ELSE
                IF      PID = 02h THEN
                        Positive response with DTC = 0000h
                ELSE
                        Negative response with "Condition not correct"
                ENDIF
        ENDIF
ENDIF
```

54.5 Service 03h - Request emission-related powertrain diagnostic trouble codes

Import action:

ACTION_ERRM_ReadDtcByTypeOfDtc	**(** **IN <TypeOfDtc>** **IN <LevelOfDtc>** **INOUT <ListOfDtc>** **OUT <ResultDtc>** **)**
This API returns a list of DTC.	

54.5.1 Functional description for ISO 14 230-4

The purpose of this mode is to enable the off-board test device to obtain stored emission-related powertrain diagnostic trouble codes (DTCs).

This operation should be divided in two phases (not necessary) :
1st phase : the diagnostic tool requests the number of emission-related troubles codes (mode = 01h, PID = 01h).
If ECM does not have stored codes, then that module shall respond with a message indicating zero codes are stored.
2nd phase : the diagnostic tool requests the diagnostic troubles codes stored (mode = 03h).

If additional trouble codes are set between the time that the number of codes are reported by an ECM, and the stored codes are reported by an ECM, then the number of codes reported could exceed the number expected by the diagnostic tool. In this case the diagnostic tool should repeat this cycle until the number of codes reported equals the number expected based on the Mode 01 response.

If there is no emission-related diagnostic trouble code to report, no response is required, but for ISO 14230-4 interfaces, the module will respond with a report containing no codes. (all Codes = 00 for the response)

Request message

Bytes	Parameter name	CVT	Hex value
#01	**Request emission-related powertrain diagnostic trouble codes**	M	03

Formula section

```
ACTION_ERRM_ReadDtcByTypeOfDtc
        (
        IN <OBD & CONFIRMED>
        IN <LAW>
        INOUT <DTC#1 to DTC#m>
        OUT <ResultDtc>
        )

IF      ResultDtc != BUFFER_FULL
        Positive response
ELSE
        Negative response general reject
ENDIF
```

First positive response message

Bytes	Parameter name	Hex value
#01	**Request emission-related powertrain diagnostic trouble codes positive response**	43
#02	Code #1 (msb)	XX or 00
#03	Code #1 (lsb)	XX or 00
#04	Code #2 (msb)	XX or 00
#05	Code #2 (lsb)	XX or 00
#06	Code #3 (msb)	XX or 00
#07	Code #3 (lsb)	XX or 00

Next positive response if more than 3 DTC

Bytes	Parameter name	Hex value
#01	**Request emission-related powertrain diagnostic trouble codes positive response**	43
#02	Code #4 (msb)	XX or 00
#03	Code #4 (lsb)	XX or 00
#04	Code #5 (msb)	XX or 00
#05	Code #5 (lsb)	XX or 00
#06	Code #6 (msb)	XX or 00
#07	Code #6 (lsb)	XX or 00

The standard does not request the order in which the diagnostic trouble codes shall be transmitted.

In the case of less than 3 codes for a response have been stored, the answer shall be completed by 0000h in order to guarantee a fixed length.

Negative response message

Bytes	Parameter name		Hex value
#1	**Negative response**		7F
#2	**Request emission-related powertrain diagnostic trouble codes**		03
#3	Error code =	Service not supported	11
		Sub function not supported - Invalid format	12
		general reject	10

Error code

Error code	Cause
Service not supported	The requested mode is not supported (vehicle is not an EOBD or OBD2 system)
Sub function not supported - Invalid format	The length of the request is incorrect

54.6 Service 04h - Clear/reset emission-related diagnostic information

Import action:

```
ACTION_ERRM_ClrInfoByTypeOfDtc
                              (
                              IN <TypeOfDTC>
                              OUT <ResultClrInfo>
                              )
```

This API clears all failure in dynamic memory

54.6.1 Functional description

The purpose of this service is to provide a means for the diagnostic tool to command on-board ECM to clear the whole failure memory (all emission-related and non emission-related diagnostic information) (See Error Management specification).

Request message

Bytes	Parameter name	Hex value
#01	**Request to clear/reset emission-related diagnostic information**	**04**

The ECM shall respond to this service request with ignition ON and the engine not running and ECM not locked by immobylizer. ECUs that cannot perform this operation under other conditions, such as with the engine running, shall respond negatively with condition not correct.

Formula section

```
ACTION_ERRM_ClrInfoByTypeOfDtc
                              (
                              IN <ALL>
                              OUT <ResultClrInfo>
                              )

IF      ResultClrInfo = INFO_CLEARED THEN
        Positive response
ELSE
        Negative response with "Condition not correct"
ENDIF
```

Refer to Error Management module

Example :
- Clear number of emission-related diagnostic trouble codes stored (Mode 01h, PID 01h)
- Clear emission-related diagnostic trouble codes (Mode 03h)
- Clear emission-related diagnostic trouble code for the freeze frame data (Mode 01h, PID 02h)
- Clear freeze frame data of the failure (Mode 02h)
- Reset status of system monitoring tests (Mode 01h, PID 01h - Readiness code)
- Clear on board monitoring test results for continuously monitored systems (Mode 07h)
- Clear distance travelled while MIL is activated (Mode 01h, PID 21h)
- Clear on-board monitoring test results for non-continuously monitored systems (Mode 06h)
  - Clear the LV_READY_XXX (inhibition of the result)

Positive response message

Bytes	Parameter name	Hex value
#01	**Request to clear/reset emission-related diagnostic information positive response**	**44**

Negative response message

Bytes	Parameter name		Hex value
#1	**Negative response**		**7F**
#2	**Request to clear/reset emission-related diagnostic information**		**04**
#3	Error code =	Service not supported	11
		Sub function not supported - Invalid format	12
		Condition not correct	22

Error code

Error code	Cause
Condition not correct	Engine running ECU locked by immobilizer
Service not supported	The requested mode is not supported (vehicle is not an EOBD or OBD2 system)
Sub function not supported - Invalid format	The length of the request is incorrect

54.7 Service 05h - Request oxygen sensor monitoring test results

Input data:

LV_READY_XXX			

54.7.1 Functional description

The purpose of this service is to allow access to the on-board oxygen sensor monitoring test results. The same information may be obtained by the use of service $06.
The request message for test results includes a Test ID value that indicates the information requested.

Many methods may be used to calculate test results for this service by different manufacturers. If data values are to be reported using these messages that are different from those specified, ranges of test values have been assigned that can be used which have standard units of measure. The external test equipment can convert these values and display them in the standard units.

The ECU will respond to this message by transmitting the requested test data last determined by the system. The latest test results are to be retained, even over multiple ignition OFF cycles, until replaced by more recent test results. Test results are requested by Test ID.

Not all test values are applicable or supported by all vehicles. An optional feature of this service is for the ECU to indicate which Test IDs are supported. Test ID $00 is a bit-encoded value that indicates support for Test IDs from $01 to $20. Test ID $20 indicates support for Test IDs $21 through $40, etc. This is the same concept as used for PID support in services $01 and $02 as specified

If Test ID $00 is not supported, then the ECU does not use this feature to indicate Test ID support.

54.7.2 Message data bytes

54.7.2.1 Request oxygen sensor monitoring test results request message definition (read supported TIDs)

— Request oxygen sensor monitoring test results request message (read supported TIDs)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request oxygen sensor monitoring test results request SID**	**M**	**05**	**SIDRQ**
#2	Test ID	M	00-20-40-60-80-A0-C0-E0	TID
#3	O2 Sensor #	M	xx	O2SNO

54.7.2.2 Request oxygen sensor monitoring test results response message definition (report supported TIDs)

— Request oxygen sensor monitoring test results response message (report supported TIDs)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request oxygen sensor monitoring test results response SID**	**M**	**45**	**SIDPR**
#2	Test ID	M	00-20-40-60-80-A0-C0-E0	TID
#3	O2 Sensor #	M	xx	O2SNO
#4	data record of supported Test IDs = [Data A: supported Test IDs,	M	xx	DATA_A
#5	Data B: supported Test IDs,	M	xx	DATA_B
#6	Data C: supported Test IDs,	M	xx	DATA_C
#7	Data D: supported Test IDs]	M	xx	DATA_D

54.7.2.3 Request oxygen sensor monitoring test results request message definition (read TID values)

— Request oxygen sensor monitoring test results request message (read TID values)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request oxygen sensor monitoring test results request SID**	**M**	**05**	**SIDRQ**
#2	Test ID	M	xx	TID
#3	O2 Sensor #	M	xx	O2SNO

54.7.2.4 Request oxygen sensor monitoring test results response message definition (report TID values)

— Request oxygen sensor monitoring test results response message (report TID values)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request oxygen sensor monitoring test results response SID**	**M**	**45**	**SIDPR**
#2	TEST ID	M	xx	TID
#3	O2 Sensor #	M	xx	O2SNO
#4	data record of Test ID = [Test Value	M	xx	TESTVAL
#5	Minimum Limit	C	xx	MINLIMIT
#6	Maximum Limit]	C	xx	MAXLIMIT
C =	Conditional — if the supported Test ID is a constant ($01 - $04) the parameters Minimum and Maximum Limit shall not be included			

54.7.3 Parameter definition

54.7.3.1 Test ID and data byte descriptions

Test ID scaling description

Test ID	Description	Min. ($00)	Max. ($FF)	Scaling/bit
$01	Rich to lean sensor threshold voltage (constant)	0 V	1.275 V	0.005 V
$02	Lean to rich sensor threshold voltage (constant)	0 V	1.275 V	0.005 V
$03	Low sensor voltage for switch time calculation (constant)	0 V	1.275 V	0.005 V
$04	High sensor voltage for switch time calculation (constant)	0 V	1.275 V	0.005 V
$05	Rich to lean sensor switch time (calculated)	0 s	1.02 s	0.004 s
$06	Lean to rich sensor switch time (calculated)	0 s	1.02 s	0.004 s
$07	Minimum sensor voltage for test cycle (calculated)	0 V	1.275 V	0.005 V
$08	Maximum sensor voltage for test cycle (calculated)	0 V	1.275 V	0.005 V
$09	Time between sensor transitions (calculated)	0 s	10.2 s	0.04 s
$0A	Sensor period (calculated)	0 s	10.2 s	0.04 s
0B	EWMA (Exponential Weighted Moving Average) misfire counts for last 10 driving cycles (calculated) Calculation: 0.1 * (current counts) + 0.9 * (previous average) Initial value for (previous average) = 0	0 count	65535 counts	1 count per bit
0C	Misfire counts for last/current driving cycles (calculated)	0 count	65535 counts	1 count per bit
$0D-$1F	reserved - to be specified by ISO			
$21-$2F	manufacturer Test ID description	0 s	1.02 s	0.004 s
$30-$3F	:	0 s	10.2 s	0.04 s
$41-$4F	:	0 V	1.275 V	0.005 V
$50-$5F	:	0 V	12.75 V	0.05 V
$61-$6F	:	0 Hz	25.5 Hz	0.1 Hz
$70-$7F	:	0 counts	255 counts	1 count
$81-$9F	manufacturer Test ID description	manufacturer specific values / units		
$A1-$BF	:	:		
$C1-$DF	:	:		
$E1-$FF	:	:		

54.7.3.2 Oxygen sensor location definition

The Oxygen sensor location value used in the request message shall indicate the Oxygen Sensor location as defined by PID $13 or $1D as specified in mode 1.

Oxygen sensor location description

Oxygen sensor location (one, and only one bit can be set to a 1)		
Bit	**Sensor location[1)]**	**Alternative sensor location[2)]**
0	Bank 1 - Sensor 1	Bank 1 - Sensor 1
1	Bank 1 - Sensor 2	Bank 1 - Sensor 2
2	Bank 1 - Sensor 3	Bank 2 - Sensor 1
3	Bank 1 - Sensor 4	Bank 2 - Sensor 2
4	Bank 2 - Sensor 1	Bank 3 - Sensor 1
5	Bank 2 - Sensor 2	Bank 3 - Sensor 2
6	Bank 2 - Sensor 3	Bank 4 - Sensor 1
7	Bank 2 - Sensor 4	Bank 4 - Sensor 2
[1)] If Service $01 PID $13 supported. [2)] If Service $01 PID $1D supported.		

54.7.3.3 Test result description

The following table defines the test result.

Test Result description

Hex	# of bytes	Description
00 - FF	1	The Test Result parameter includes either a constant or a calculated value depending on the Test ID.

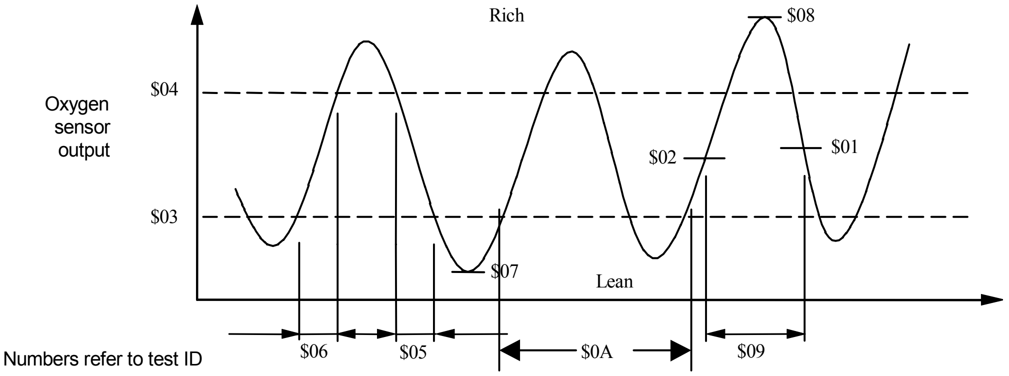
54.7.3.4 Minimum and Maximum Test Limit description

The following table defines Minimum and Maximum Test Limit.

Minimum and Maximum Test Limit description

Test Limit	# of bytes	Description
Minimum	1	The minimum test limit (only for calculated test result) is the minimum value to which the test result is compared. The Test Limit value is either a minimum or a maximum value to which the test results are compared. The Test Limit unsigned numeric value
Maximum	1	The maximum test limit (only for calculated test result) is the maximum value to which the test result is compared.

Results of latest mandated on-board oxygen sensor monitoring test, see figure below.



Test ID value example

54.8 Service 06h - Request on-board monitoring test result for non-continuously monitored system

Input data:

LV_READY_XXX			

54.8.1 Functional description

The purpose of this service is to allow access to the results (test value and test limit) for an on-board diagnostic monitoring tests of specific components/system that are not continuously monitored.
The latest test results are to be retained, even over multiple ignition 'off' cycles, until replaced by more recent test results.

Not all TIDs are applicable or supported by all systems. Then, all ECUs must at least answer to TID 00h by sending back the list of the PIDs used.

This service can be used as an alternative to service 05h to report oxygen sensor test results.

You can find the exhaustive list of generic TIDs definition in chapter “TID description for mode 06h”.
The specific list of supported TIDs for current project can be found in chapter “Emission-related software and system configuration”.

Request message (read supported PIDs)

Bytes	Parameter name	Cvt	Hex value
#01	**Request on-board monitoring test results for non-continuously monitored systems**	**M**	**06**
#02	On-board diagnostic monitor ID	M	00-20-40-60-80-A0-C0-E0

Positive response (read supported PIDs)

Bytes	Parameter name	Cvt	Hex value
#01	**Request on-board monitoring test results for non-continuously monitored systems positive response**	**M**	**46**
#02	Test ID	M	00-20-40-60-80-A0-C0-E0
#03	Test limit type and component ID	M	FF
#04	Data A	M	XX
#05	Data B	M	XX
#06	Data C	M	XX
#07	Data D	M	XX

Request message (read test result)

Bytes	Parameter name	CVT	Hex value
#01	**Request on-board monitoring test results for non-continuously monitored systems**	**M**	**06**
#02	On-board diagnostic monitor ID	M	XX

Response message (report test status)

Bytes	Parameter name		Hex value
#01	**Request on-board monitoring test results for non-continuously monitored systems positive response**	**M**	**46**
#02	Test ID	M	XX
#03	Test limit type and component ID	M	XX
#04	Test value (msb)	M	XX
#05	Test value (lsb)	M	XX
#06	Test limit (msb)	M	XX
#07	Test limit (lsb)	M	XX

Formula section:

```
IF LV_READY_XXX = ON THEN
        Negative response (No result since last reset/clear)
ELSE
        Positive response with test result
END
```

Test limit type and component ID definition

Byte #3	Value	Description
bit 0 to bit 6		Component ID - Manufacturer defined - necessary when multiple components or systems are present on the vehicle and have the same definition of test ID.
bit 7		Most significant bit indicates type of test limit, where:
	0	Test limit is maximum value - test fails if test value is greater than this value
	1	Test limit is minimum value - test fails if test value is less than this value
		If the test result should be within a range of values, two messages will be returned, one with the minimum limit value and one with the maximum limit value.

Negative response message

Bytes	Parameter name		Hex value
#1	**Negative response**		7F
#2	**Request on-board monitoring test results for non-continuously monitored systems**		06
#3	Error code =	Service not supported Sub function not supported - Invalid format Condition not correct	11 12 22

54.8.1.1 Error code

Error code	Cause
Service not supported	The requested mode is not supported (vehicle is not an EOBD or OBD2 system)
Sub function not supported - Invalid format	The requested TID is not supported or the length of the request is incorrect
Condition not correct	No result since last reset/clear or Test in progress

54.9 Service 07 - Request on_board monitoring test results for continuously monitored system

Import action:

```
ACTION_ERRM_ReadDtcByTypeOfDtc
                          (
                          IN <TypeOfDtc>
                          IN <LevelOfDtc>
                          INOUT <ListOfDtc>
                          OUT <ResultDtc>
                          )
```

This API returns a list of non confirmed DTC.

54.9.1 Functional description

The purpose of this service is to enable the off-board test device to obtain test results for emission-related powertrain components/systems that are continuously monitored during normal driving conditions.
The intented use of this data is to assist the service technician after a vehicle repair, and after clearing diagnostic information, by reporting test results after a single driving cycle. If the test failed during the driving cycle, the DTC associated with that test will be reported. Test results reported by this service do not necessarily indicate a faulty component/system. If test results indicate a failure after additional driving, then the MIL will be illuminated and a DTC will be set and reported with Mode 03, indicating a faulty component/system. Once Service $03 indicates a faulty component/system, Service $07 may or may not continue to report it as a one-trip failure also.
If an ECU does not have stored DTCs then it shall respond with a message indicating no codes are stored by setting the parameter "Number of DTC" to 00h.

Request message

Bytes	Parameter name	CVT	Hex value
#01	**Request on-board monitoring test results for continuously monitored system**	M	07

Formula section:

```
ACTION_ERRM_ReadDtcByTypeOfDtc
                          (
                          IN <OBD&TEMPORARY>
                          IN <LAW>
                          INOUT <DTC#1 to DTC#m>
                          OUT <ResultDtc>
                          )

IF      ResultDtc != NO_BUFFER_FULL
        Positive response
ELSE
        Negative response general reject
ENDIF
```

First positive response message

Bytes	Parameter name	Hex value
#01	**Request on-board monitoring test results for continuously monitored system positive response**	47
#02	Code #1 (msb)	XX or 00
#03	Code #1 (lsb)	XX or 00
#04	Code #2 (msb)	XX or 00
#05	Code #2 (lsb)	XX or 00
#06	Code #3 (msb)	XX or 00
#07	Code #3 (lsb)	XX or 00

Next positive response message

Bytes	Parameter name	Hex value
#01	**Request on-board monitoring test results for continuously monitored system positive response**	47
#02	Code #4 (msb)	XX or 00
#03	Code #4 (lsb)	XX or 00
#04	Code #5 (msb)	XX or 00
#05	Code #5 (lsb)	XX or 00
#06	Code #6 (msb)	XX or 00
#07	Code #6 (lsb)	XX or 00

If different failures have the same stored DTC number, only one DTC number shall be reported.

The standard does not request the order in which the diagnostic trouble codes shall be transmitted.

In the case of less than 3 codes have been stored, the answer shall be completed by 0000h in order to guarantee a fixed length.

Negative response message

Bytes	Parameter name		Hex value
#1	**Negative response**		7F
#2	**Request emission-related powertrain diagnostic trouble codes**		07
#3	Error code =	Service not supported Sub function not supported - Invalid format General reject	11 12 10

Error code

Error code	Cause
Service not supported	The requested mode is not supported (vehicle is not an EOBD or OBD2 system)
Sub function not supported - Invalid format	The length of the request is incorrect

54.10 PID description for mode 01h

Import action:

ACTION_ERRM_ReadReadinessCode **(** **INOUT <ReadinessCode>** **OUT <ResultReadinessCode>** **)**
This API allows to read the readiness code information
ACTION_ERRM_ReadQuantityOfDtc **(** **IN <TypeOfDtc>** **IN <LevelOfDtc>** **INOUT <Quantity>** **OUT <ResultQuantity>** **)**
This API returns the quantity of DTC with a certain type, which are stored in memory

Input data:

T_ACT_MIL_SAE	PSN_LS	OBD_TPS_REL	STATE_OBD_SA
C_CLC_READY_OBD_1	LOAD_ABSV	STATE_LS[NC_CBK_EX_NR]	LOAD_CLC
TCO_MES	FAC_LAM_LIM[NC_CBK_EX_NR]	OBD_LAM_AD[NC_CBK_EX_NR]	FUP
MAP_SAE	N	VS	IGA_IGC[NC_CYL_NR]
TIA_MES	MAF_KGH_MES	OBD_PV_2	OBD_PV_1
T_DTC_CLR	VLS_DOWN[NC_CBK_EX_NR]	C_OBD_REQ	T_AST_SAE
DIST_ACT_MIL	LAMB_LS_UP[NC_CBK_EX_NR]	VLS_COR_LSL[NC_CBK_EX_NR]	CPPWM_CPS
OBD_FTL		CTR_WUP_DTC_CLR	DIST_DTC_CLR
AMP_MES	IPLSL_COR[NC_CBK_EX_NR]	TEG_CAT_UP_MDL[NC_CBK_EX_NR]	TEG_CAT_DOWN_MDL[NC_CBK_EX_NR]
STATE_ENA_OBD	STATE_CMPL_OBD	VB	OBD_TPS_2
LAMB_SP_HOM[NC_CBK_EX_NR]	OBD_TPS_SP	OBD_TAM	OBD_TPS_1
MOD_1_PID_1_DIAG	MOD_1_PID_2_DIAG	MOD_1_PID_3_DIAG	MOD_1_PID_4_DIAG
MOD_1_PID_5_DIAG	MOD_1_PID_6_DIAG	MOD_1_PID_7_DIAG	MOD_1_PID_8_DIAG
MOD_1_PID_9_DIAG	MOD_1_PID_A_DIAG	MOD_1_PID_B_DIAG	MOD_1_PID_C_DIAG
FUP_RNG_H_MES	NC_INJ_CONF	OPG_SP_ACR	OBD_EGR_DIF

Note: This specification is according with the ISO 15031–5 issue 8.

54.10.1 Description of PID 00h - Defines Supported PIDs 01h - 20h

0 : the PID is not used
1 : the PID is used

DATA A - Supported PIDs 01h to 08h

PID 00h – DATA A - Supported PID 01h to 08h		
BIT	**Function**	**Name**
0	PID 08h : Short term fuel trim - Bank 2	MOD_1_PID_1_DIAG
1	PID 07h : Long term fuel trim - Bank 1	
2	PID 06h : Short term fuel trim - Bank 1	
3	PID 05h : Engine coolant temperature	
4	PID 04h : Calculated load value	
5	PID 03h : Fuel system status	
6	PID 02h : Trouble code that caused required freeze frame	
7	PID 01h : DTCs and supported tests	

DATA B - Supported PIDs 09h to 10h

PID 00h – DATA B - Supported PID 09h to 10h		
BIT	**Function**	**Name**
0	PID 10h : Air flow rate from MAF sensor	MOD_1_PID_2_DIAG
1	PID 0Fh : Intake air temperature	
2	PID 0Eh : Ignition time advance	
3	PID 0Dh : Vehicle speed	
4	PID 0Ch : Engine RPM	
5	PID 0Bh : Intake manifold absolute pressure	
6	PID 0Ah : Fuel pressure (gauge)	
7	PID 09h : Long term fuel trim - Bank 2	

DATA C - Supported PIDs 11h to 18h

PID 00h – DATA C - Supported PID 11h to 18h		
BIT	**Function**	**Name**
0	PID 18h : Bank 2 Sensor 1 or Bank 3 Sensor 1	MOD_1_PID_3_DIAG
1	PID 17h : Bank 1 Sensor 4 or Bank 2 Sensor 2	
2	PID 16h : Bank 1 Sensor 3 or Bank 2 Sensor 1	
3	PID 15h : Bank 1 Sensor 2	
4	PID 14h : Bank 1 Sensor 1	
5	PID 13h : Location of oxygen sensors	
6	PID 12h : Secondary air status	
7	PID 11h : Absolute throttle position sensor	

DATA D - Supported PIDs 19h to 20h

PID 00h - DATA D - Supported PID 19h to 20h		
BIT	**Function**	**Name**
0	PID 20h : PIDs supported ($21 - $40)	MOD_1_PID_4_DIAG
1	PID 1Fh : Time Since Engine Start	
2	PID 1Eh : Auxiliary Input Status	
3	PID 1Dh : Location of oxygen sensors	
4	PID 1Ch : OBD requirements	
5	PID 1Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2	
6	PID 1Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1	
7	PID 19h : Bank 2 Sensor 2 or Bank 3 Sensor 2	

54.10.2 Description of PID 01h - Monitor status since DTCs cleared

PID $01 definition

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
01	Monitor status since DTCs cleared			
	The bits in this PID shall report two pieces of information for each monitor: a) Monitor status since DTCs were last cleared, saved in NVRAM or KAM. b) Monitors supported on this vehicle.			
	Number of emission-related DTCs and MIL status	A (bit)	byte 1 of 4	DTC and MIL status:
	# of DTCs stored in this ECU	0-6	hex to decimal	DTC_CNT: xxxd
	Malfunction Indicator Light (MIL) Status	7	0 = MIL OFF; 1 = MIL ON	MIL: OFF or ON
	The MIL status shall indicate "OFF" during the key on, engine off bulb check unless the MIL has also been commanded "ON" for a detected malfunction.			
	Supported tests which are continuous	B (bit)	byte 2 of 4 (Low Nibble)	Support status of continuous monitors:
	Misfire monitoring	0	0 = monitor not supported (NO) 1 = monitor supported (YES)	MIS_SUP: NO or YES
	Misfire monitoring shall be supported on both, spark ignition and compression vehicles if the vehicle utilises a misfire monitor.			
	Fuel system monitoring	1	0 = monitor not supported (NO) 1 = monitor supported (YES)	FUEL_SUP: NO or YES
	Fuel system monitoring shall be supported on vehicles that utilise oxygen sensors for closed loop fuel feedback control, and utilise a fuel system monitor, typically spark ignition engines.			
	Comprehensive component monitoring	2	0 = monitor not supported (NO) 1 = monitor supported (YES)	CCM_SUP: NO or YES
	Comprehensive component monitoring shall be supported on spark ignition and compression ignition vehicles that utilise comprehensive component monitoring.			
	reserved (bit shall be reported as ‘0’)	3		---

PID $01 definition (continued)

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
	Status of continuous monitoring tests since DTC cleared:	B (bit)	byte 2 of 4 (High Nibble)	Completion status of continuous monitors since DTC cleared:
	Misfire monitoring	4	0 = monitor complete, or not applicable (YES) 1 = monitor not complete (NO)	MIS_RDY: YES or NO
	Misfire monitoring shall always indicate complete for spark ignition engines. Misfire monitoring shall indicate complete for compression ignition engines after the misfire evaluation is complete.			
	Fuel system monitoring	5	0 = monitor complete, or not applicable (YES) 1 = monitor not complete (NO)	FUEL_RDY: YES or NO
	Fuel system monitoring shall always indicate complete for both spark ignition and compression ignition engines.			
	Comprehensive component monitoring	6	0 = monitor complete, or not applicable (YES) 1 = monitor not complete (NO)	CCM_RDY: YES or NO
	Comprehensive component monitoring shall always indicate complete on both spark ignition and compression ignition engines. NOTE It can be assumed that by the time any non-continuous monitors are complete, continuous comprehensive component monitoring will also be complete. Bit 6 is allowed to always indicate "complete" on spark ignition vehicles that support other non-continuous monitors.			
	reserved (bit shall be reported as ‘0’)	7		---
	Supported tests run at least once per trip	C (bit)	byte 3 of 4	Support status of non-continuous monitors:
	Catalyst monitoring	0	0 = monitor not supported (NO) 1 = monitor supported (YES)	CAT_SUP: NO or YES
	Heated catalyst monitoring	1		HCAT_SUP: NO or YES
	Evaporative system monitoring	2		EVAP_SUP: NO or YES
	Secondary air system monitoring	3		AIR_SUP: NO or YES
	A/C system refrigerant monitoring	4		ACRF_SUP: NO or YES
	Oxygen sensor monitoring	5		O2S_SUP: NO or YES
	Oxygen sensor heater monitoring	6		HTR_SUP: NO or YES
	EGR system monitoring	7		EGR_SUP: NO or YES
	Status of tests run at least once per trip	D (bit)	byte 4 of 4	Completion status of non-continuous monitors since DTCs cleared:
	Catalyst monitoring	0	0 = monitor complete, or not applicable (YES) 1 = monitor not complete (NO)	CAT_RDY: YES or NO
	Heated catalyst monitoring	1		HCAT_RDY: YES or NO
	Evaporative system monitoring	2		EVAP_RDY: YES or NO
	Secondary air system monitoring	3		AIR_RDY: YES or NO
	A/C system refrigerant monitoring	4		ACRF_RDY: YES or NO
	Oxygen sensor monitoring	5		O2S_RDY: YES or NO
	Oxygen sensor heater monitoring	6		HTR_RDY: YES or NO
	EGR system monitoring	7		EGR_RDY: YES or NO

Data Byte	Application data to use		Formating conversion used
data A	(bit 0 to 6) :	NR_DTC	-
	(bit 7) :	STATE_MIL	
data B	STATE_READY_OBD_1 *		
data C	C_STATE_READY_OBD_2		
data D	STATE_READY_OBD_2		

*) The bit 0 up to 3 of this byte is calibration value C_STATE_READY_OBD_1.
The bit 4 up to 6 of this byte indicate if the tests has been performed continuously since the reinitialization of the memorized information.

Formula section:

```
ACTION_ERRM_ReadQuantityOfDtc
                                        (
                                        IN <OBD & CONFIRMED>
                                        IN <LAW>
                                        INOUT <NR_DTC>
                                        OUT <ResultQuantity>
                                        )

ACTION_ERRM_ReadReadinessCode
                                        (
                                        INOUT < C_STATE_READY_OBD_2,
                                                STATE_READY_OBD_1,
                                                STATE_READY_OBD_2>
                                        OUT <ResultReadinessCode>
                                        )

IF      (ResultQuantity = DTC_PRESENT           OR
        ResultQuantity = NO_DTC_PRESENT)        AND
        ResultReadinessCode = POSITIVE_RESPONSE
THEN
        Positive response

ELSE
        No answer
ENDIF
```

These actions should be sent at each time we use the mode 01h PID01.
Refer to “Error management” module.

54.10.3 Description of PID 03h - Status of fuel system

PID $03 definition

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
03	Fuel system 1 status:	A (bit)	byte 1 of 2	FUELSYS1:
	(unused bits shall be reported as ‘0’; no	0	1 = Open loop - has not yet satisfied conditions to go closed loop	OL
	more than one bit at a time can be set to a ‘1’	1	1 = Closed loop - using oxygen sensor(s) as feedback for fuel control	CL
	of that bank)	2	1 = Open loop due to driving conditions (e.g. power enrichment, deceleration enleanment)	OL-Drive
		3	1 = Open loop - due to detected system fault	OL-Fault
		4	1 = Closed loop, but fault with at least one oxygen sensor - may be using single oxygen sensor for fuel control	CL-Fault
		5-7	reserved (bits shall be reported as ‘0’)	---
	NOTE Fuel systems do not refer to injector banks. Fuel systems are intended to represent completely different fuel systems that can independently enter and exit closed loop fuel. Banks of injectors on a V-engine are generally not independent and share the same closed-loop enablement criteria.			
	Fuel system 2 status:	B (bit)	byte 2 of 2	FUELSYS2:
	(unused bits shall be reported as ‘0’; no	0	1 = Open loop - has not yet satisfied conditions to go closed loop	OL
	more than one bit at a time can be set to a ‘1’	1	1 = Closed loop - using oxygen sensor(s) as feedback for fuel control	CL
	of that bank)	2	1 = Open loop due to driving conditions (e.g. power enrichment, deceleration enleanment)	OL-Drive
		3	1 = Open loop - due to detected system fault	OL-Fault
		4	1 = Closed loop, but fault with at least one oxygen sensor - may be using single oxygen sensor for fuel control	CL-Fault
		5-7	reserved (bits shall be reported as ‘0’)	---

IF PSN_LS = 11H **or** 33H

THEN

Data Byte	Application data to use	Formating conversion used
data A	STATE_LS[1]	no
data B	STATE_LS[2]	no

ELSE

Data Byte	Application data to use	Formating conversion used
data A	None	no
data B	None	no

ENDIF

54.10.4 Description of PID 04h - Calculated load value

PID $04 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
04	Calculated LOAD Value!	A	0 %	100 %	100/255 %	LOAD_PCT: xxx.x %
	The OBD regulations previously defined CLV as: (current airflow / peak airflow @ sea level) * (BARO @ sea level / BARO) * 100% Various manufacturers have implemented this calculation in a variety of ways. The following definition, although a little more restrictive, will standardise and improve the accuracy the calculation. LOAD_PCT = [current airflow] / [(peak airflow at WOT@STP as a function of rpm) * (BARO/29.92) * SQRT(298/(AAT+273))] — Where: STP = Standard Temperature and Pressure = 25 ^{o}C, 29.92 in Hg BARO, SQRT = square root, — WOT = wide open throttle, AAT=Ambient Air Temperature and is in ^{o}C Characteristics of LOAD_PCT are: — Reaches 1.0 at WOT at any altitude, temperature or rpm for both naturally aspirated and boosted engines. — Indicates percent of peak available torque. — Linearly correlated with engine vacuum — Often used to schedule power enrichment. — Compression ignition engines (diesels) shall support this PID using fuel flow in place of airflow for the above calculations. NOTE Both spark ignition and compression ignition engines shall support PID $04. See PID $43 for an additional definition of engine LOAD.					

Data Byte	Application data to use	Formating conversion used
data A	LOAD_CLC	no

54.10.5 Description of PID 05h - Engine coolant temperature

PID $05 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
05	Engine Coolant Temperature	A	-40 °C	+215 °C	1 °C with -40 °C offset	ECT: xxx °C (xxx °F)
	ECT shall display engine coolant temperature derived from an engine coolant temperature sensor or a cylinder head temperature sensor. Many diesels do not use either sensor and may substitute Engine Oil Temperature instead.					

Data Byte	Application data to use	Formating conversion used
data A	TCO_MES	Shift to different range

54.10.6 Description of PID 06h - Short term fuel trim - Bank 1 and Bank 3

PID $06 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
06	Short Term Fuel Trim - Bank 1 (use if only 1 fuel trim value) Short Term Fuel Trim - Bank 3	A B	-100 % (lean)	+99.22 % (rich)	100/128 % (0 % at 128)	SHRTFT1: xxx.x % SHRTFT3: xxx.x %
	Short Term Fuel Trim Bank 1/3 shall indicate the correction being utilised by the closed loop fuel algorithm. If the fuel system is in open loop, SHRTFT1/3 shall report 0% correction. NOTE Data B shall only be included in the response message of a PID $06 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 3 supported) of PID $1D if Data B of PID $06 is supported or not.					

IF PSN_LS = 11H **or** 33H

THEN

Data Byte	Application data to use	Formating conversion used
data A	FAC_LAM_LIM[1]	Shift to different range

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	

ENDIF

54.10.7 Description of PID 07h - Long term fuel trim - Bank 1 and Bank 3

PID $07 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
07	Long Term Fuel Trim - Bank 1 (use if only 1 fuel trim value) Long Term Fuel Trim - Bank 3	A B	-100 % (lean)	+99.22 % (rich)	100/128 % (0 % at 128)	LONGFT1: xxx.x % LONGFT3: xxx.x %
	Fuel trim correction for Bank 1/3 stored in Non-volatile RAM or Keep-alive RAM. LONGFT shall indicate the correction being utilised by the fuel control algorithm at the time the data is requested, in both open loop and closed loop fuel control. If no correction is utilised in open loop fuel, LONGFT shall report 0% correction. If long-term fuel trim is not utilised at all by the fuel control algorithm, the PID shall not be supported. NOTE Data B shall only be included in the response message of a PID $07 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 3 supported) of PID $1D if Data B of PID $07 is supported or not.					

IF PSN_LS = 33H

THEN

Data Byte	Application data to use	Formating conversion used
data A	OBD_LAM_AD[1]	-

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	

ENDIF

54.10.8 Description of PID 08h - Short term fuel trim - Bank 2 and Bank 4

PID $08 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
08	Short Term Fuel Trim - Bank 2 (use if only 1 fuel trim value) Short Term Fuel Trim - Bank 4	A B	-100 % (lean)	+99.22 % (rich)	100/128 % (0 % at 128)	SHRTFT2: xxx.x % SHRTFT4: xxx.x %
	Short Term Fuel Trim Bank 2/4 shall indicate the correction being utilized by the closed loop fuel algorithm. If the fuel system is in open loop, SHRTFT24 shall report 0% correction. NOTE Data B shall only be included in the response message of a PID $08 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 4 supported) of PID $1D if Data B of PID $08 is supported or not.					

IF PSN_LS = 11H **or** 33H

THEN

Data Byte	Application data to use	Formating conversion used
data A	FAC_LAM_LIM[2]	Shift to different range

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	

ENDIF

54.10.9 Description of PID 09h - Long term fuel trim - Bank 2 and Bank 4

PID $09 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
09	Long Term Fuel Trim – Bank 2 (use if only 1 fuel trim value) Long Term Fuel Trim - Bank 4	A B	-100 % (lean)	+99.22 % (rich)	100/128 % (0 % at 128)	LONGFT2: xxx.x % LONGFT4: xxx.x %
	Fuel trim correction for Bank 2/4 stored in Non-volatile RAM or Keep-alive RAM. LONGFT shall indicate the correction being utilised by the fuel control algorithm at the time the data is requested, in both open loop and closed loop fuel control. If no correction is utilised in open loop fuel, LONGFT shall report 0% correction. If long-term fuel trim is not utilised at all by the fuel control algorithm, the PID shall not be supported. NOTE Data B shall only be included in the response message of a PID $09 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 4 supported) of PID $1D if Data B of PID $09 is supported or not.					

IF PSN_LS = 33H

THEN

Data Byte	Application data to use	Formating conversion used
data A	OBD_LAM_AD[2]	-

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	

ENDIF

54.10.10 Description of PID 0Ah - Fuel pressure (gauge)

PID $0A definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0A	Fuel Rail Pressure (gauge)	A	0 kPa (gauge)	765 kPa (gauge)	3 kPa per bit (gauge)	FRP: xxx kPa (xx.x psi)
	FRP shall display fuel rail pressure at the engine when the reading is referenced to atmosphere (gauge pressure).					

Data Byte	Application data to use	Formating conversion used
data A	FUP	Shift to different range

54.10.11 Description of PID 0Bh - Intake manifold absolute pressure

PID $0B definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0B	Intake Manifold Absolute Pressure	A	0 kPa (absolute)	255 kPa (absolute)	1 kPa per bit (absolute)	MAP: xxx kPa (xx.x inHg)
	MAP shall display manifold pressure derived from a Manifold Absolute Pressure sensor, if a sensor is utilised. If a vehicle uses both a MAP and MAF sensor, both the MAP and MAF PIDs shall be supported.					

Data Byte	Application data to use	Formating conversion used
data A	MAP_SAE	None

54.10.12 Description of PID 0Ch - Engine speed

PID $0C definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0C	Engine RPM	A , B	0 min^{-1}	16383.75 min^{-1}	¼ rpm per bit	RPM: xxxxx min^{-1}

Data Byte	Application data to use	Formating conversion used
data A	MSB (N)	Shift to different range
data B	LSB (N)	Shift to different range

54.10.13 Description of PID 0Dh - Vehicle speed

PID $0D definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0D	Vehicle Speed Sensor	A	0 km/h	255 km/h	1 km/h per bit	VSS: xxx km/h (xxx mph)
	VSS shall display vehicle road speed, if utilised by the control module strategy. Vehicle speed may be derived from a vehicle speed sensor, calculated by the PCM using other speed sensors, or obtained from the vehicle serial data communication bus.					

Data Byte	Application data to use	Formating conversion used
data A	VS	no

54.10.14 Description of PID 0Eh - Ignition timing advance

PID $0E definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0E	Ignition Timing Advance for #1 Cylinder	A	-64 °	63.5 °	½ ° with 0 ° at 128	SPARKADV: xx °
	Ignition timing spark advance for #1 cylinder (not including mechanical advance)					

Data Byte	Application data to use	Formating conversion used
data A	IGA_IGC[0]	Shift to different range

54.10.15 Description of PID 0Fh - Intake air temperature

PID $0B definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0F	Intake Air Temperature	A	-40 °C	+215 °C	1 °C with -40 °C offset	IAT: xxx °C (xxx °F)
	IAT shall display intake manifold air temperature, if utilised by the control module strategy. IAT may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	TIA_MES	Shift to different range

54.10.16 Description of PID 10h - Air flow rate from MAF

PID $0B definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
10	Air Flow Rate from Mass Air Flow Sensor	A , B	0 g/s	655.35 g/s	0.01 g/s	MAF: xxx.xx g/s (xxxx.x lb/min)
	MAF shall display the airflow rate as measured by the MAF sensor, if a sensor is utilised.					

Data Byte	Application data to use	Formating conversion used
data A	MSB (MAF_KGH_MES)	Kg/h into g/s
data B	LSB (MAF_KGH_MES)	Kg/h into g/s

54.10.17 Description of PID 11h - Absolute throttle position sensor

PID $0A - $11 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
11	Absolute Throttle Position	A	0 %	100 %	100/255 %	TP: xxx.x %
	Absolute throttle position (not “relative” or “learned” throttle position) shall be displayed as a normalised value, scaled from 0 to 100%. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed throttle position is a 1.0 volts, TP shall display (1.0 / 5.0) = 20% at closed throttle and 50% at 2.5 volts. Throttle position at idle will usually indicate greater than 0%, and throttle position at wide open throttle will usually indicate less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage. NOTE See PID $45 for a definition of Relative Throttle Position.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TPS_1	no

54.10.18 Description of PID 12h - Commanded secondary air status

With autolearning the value for the LV_VAR_SAP is defined to the value defined by the project

IF PID $12 supported and LV_VAR_SAP <> 0

THEN

PID $12 definition

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
12	Commanded Secondary Air Status	A (bit)	byte 1 of 1	AIR_STAT:
	(if supported, one, and only	0	1 = upstream of first catalytic converter	AIR_STAT: UPS
	one bit at a time can be set	1	1 = downstream of first catalytic converter inlet	AIR_STAT: DNS
	to a 1)	2	1 = atmosphere / off	AIR_STAT: OFF
		3-7	reserved (bits shall be reported as ‘0’)	---

Data Byte	Calibration data to use	Formating conversion used
data A	STATE_OBD_SA	-

ENDIF

54.10.19 Description of PID 13h - Location of oxygen sensors

PID $13 definition

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
13	Location of Oxygen Sensors	A (bit)	byte 1 of 4	O2SLOC:
	(where sensor 1 is closest to the engine. Each bit indicates the presence or absence of an oxygen sensor at the following location)	0	1 = Bank 1 - Sensor 1 present at that location	O2S11
		1	1 = Bank 1 - Sensor 2 present at that location	O2S12
		2	1 = Bank 1 - Sensor 3 present at that location	O2S13
		3	1 = Bank 1 - Sensor 4 present at that location	O2S14
		4	1 = Bank 2 - Sensor 1 present at that location	O2S21
		5	1 = Bank 2 - Sensor 2 present at that location	O2S22
		6	1 = Bank 2 - Sensor 3 present at that location	O2S23
		7	1 = Bank 2 - Sensor 4 present at that location	O2S24

With autolearning the value for the PSN_LS is defined like this:

```
IF      PID 13h is supported
THEN    PSN_LS is defined to the value defined by the project
ELSE    PSN_LS is defined to 0
ENDIF
```

Data Byte	Calibration data to use	Formating conversion used
data A	PSN_LS	no

54.10.20 Description of PID 14h - Bank 1 Sensor 1 (not supported)

54.10.21 Description of PID 15h - Bank 1 Sensor 2

PID $15 definition if PID $13 is supported!

PID (hex)	Description Use if PID $13 or PID $1D are supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
15	Oxygen Sensor Output Voltage Bank 1 – Sensor 2	A	0 V	1.275 V	0.005 V	O2Sxy: x.xxx V
	Short Term Fuel Trim (B1-S2) (associated with this sensor $FF if this sensor is not used in the calculation)	B	-100.00 % (lean)	99.22 % (rich)	100/128 % (0 % at 128)	SHRTFTxy: xxx.x %
	These PIDs shall be used for a conventional, 0 to 1 Volt oxygen sensor. Any sensor with a different full scale value shall be normalised to provide nominal full scale at $C8 (200 decimal). Wide-range/linear oxygen sensors shall use PID $25 or PID $35.					

```
IF      PSN_LS = 33H
THEN
```

Data Byte	Application data to use	Formating conversion used
data A	VLS_DOWN[1]	-
data B	0xFF	

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	-
data B	NONE	

ENDIF

54.10.22 Description of PID 16h - Bank 1 Sensor 3 or Bank 2 Sensor 1 (not supported)

54.10.23 Description of PID 17h - Bank 1 Sensor 4 or Bank 2 Sensor 2 (not supported)

54.10.24 Description of PID 18h - Bank 2 Sensor 1 or Bank 3 Sensor 1 (not supported)

54.10.25 Description of PID 19h - Bank 2 Sensor 2 or Bank 3 Sensor 2

PID $19 definition if PID $13 is supported!

PID (hex)	Description Use if PID $13 is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
19	Oxygen Sensor Output Voltage Bank 2 – Sensor 2	A	0 V	1.275 V	0.005 V	O2Sxy: x.xxx V
	Short Term Fuel Trim (B2-S2) (associated with this sensor $FF if this sensor is not used in the calculation)	B	-100.00 % (lean)	99.22 % (rich)	100/128 % (0 % at 128)	SHRTFTxy: xxx.x %
	These PIDs shall be used for a conventional, 0 to 1 Volt oxygen sensor. Any sensor with a different full scale value shall be normalised to provide nominal full scale at $C8 (200 decimal). Wide-range/linear oxygen sensors shall use PID $29 or PID $39.					

IF PSN_LS = 33H

THEN

Data Byte	Application data to use	Formating conversion used
Data A	VLS_DOWN[2]	-
Data B	0xFF	

ELSE

Data Byte	Application data to use	Formating conversion used
Data A	NONE	-
Data B	NONE	

ENDIF

54.10.26 Description of PID 1Ah - Bank 2 Sensor 3 or Bank 4 Sensor 1 (not supported)

54.10.27 Description of PID 1Bh - Bank 2 Sensor 4 or Bank 4 Sensor 2 (not supported)

54.10.28 Description of PID 1Ch - OBD requirements

PID $1C definition

PID (hex)	Description	Data byte	Scaling	External test equipment SI (Metric) / English display
1C	OBD requirements to which vehicle is designed	A (hex)	byte 1 of 1 (State Encoded Variable)	OBDSUP:
		01	OBD II (California ARB)	OBD II
		02	OBD (Federal EPA)	OBD
		03	OBD and OBD II	OBD and OBD II
		04	OBD I	OBD I
		05	Not OBD compliant	NO OBD
		06	EOBD	EOBD
		07	EOBD and OBD II	EOBD and OBD II
		08	EOBD and OBD	EOBD and OBD
		09	EOBD, OBD and OBD II	EOBD, OBD and OBD II
		0A - FF	reserved by document	---

Data Byte	Calibration data to use	Formating conversion used
data A	C_OBD_REQ	-

54.10.29 Description of PID 1Dh - Location of oxygen sensors (not supported)

54.10.30 Description of PID 1Eh - Auxiliary input status (not supported)

54.10.31 Description of PID 1Fh - Time Since Engine Start

PID $1F definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
1F	Time Since Engine Start	A , B	0 sec.	65,535 sec.	1 second per count	RUNTM: xxxxx sec.
	RUNTM shall increment while the engine is running. It shall freeze if the engine stalls. RUNTM shall be reset to zero during every control module power-up and when entering the key-on, engine off position. RUNTM is limited to 65,535 seconds and shall not wrap around to zero.					

Data Byte	Application data to use	Formating conversion used
data A	MSB (T_AST_SAE)	-
data B	LSB (T_AST_SAE)	-

54.10.32 Description of PID 20h - Supported PIDs 21h - 40h

0 : the PID is not used
1 : the PID is used

DATA A - Supported PIDs 21h - 28h

PID 20h - DATA A - Supported PID 21h to 28h		
BIT	**Function**	**Name**
0	PID 28h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	MOD_1_PID_5_DIAG
1	PID 27h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	
2	PID 26h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	
3	PID 25h : Bank 1 Sensor 2 (Wide range O2S)	
4	PID 24h : Bank 1 Sensor 1 (Wide range O2S)	
5	PID 23h : Diesel fuel pressure (gauge)	
6	PID 22h : Relative fuel pressure	
7	PID 21h : Distance travelled while MIL is activated	

DATA B - Supported PIDs 29h to 30h

PID 20h - DATA B - Supported PID 29h to 30h		
BIT	**Function**	**Name**
0	PID 30h : Number of warm-ups since diagnostic trouble codes cleared	MOD_1_PID_6_DIAG
1	PID 2Fh : Fuel Level Input	

2	PID 2Eh : Commanded Evaporative Purge	
3	PID 2Dh : EGR Error	
4	PID 2Ch : Command EGR	
5	PID 2Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	
6	PID 2Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	
7	PID 29h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	

DATA C - Supported PIDs 31h to 38h

PID 20h - DATA C - Supported PID 31h to 38h		
BIT	**Function**	**Name**
0	PID 38h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	MOD_1_PID_7_DIAG
1	PID 37h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	
2	PID 36h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	
3	PID 35h : Bank 1 Sensor 2 (Wide range O2S)	
4	PID 34h : Bank 1 Sensor 1 (Wide range O2S)	
5	PID 33h : Barometric Pressure	
6	PID 32h : Evap System Vapor Pressure	
7	PID 31h : Distance since diagnostic trouble codes cleared	

DATA D - Supported PIDs 39h to 40h

PID 20h - DATA D - Supported PID 39h to 40h		
BIT	**Function**	**Name**
0	PID 40h : PIDs supported ($41 - $60)	MOD_1_PID_8_DIAG
1	PID 3Fh : Catalyst Temperature Bank 2, Sensor 2	
2	PID 3Eh : Catalyst Temperature Bank 1, Sensor 2	
3	PID 3Dh : Catalyst Temperature Bank 2, Sensor 1	
4	PID 3Ch : Catalyst Temperature Bank 1, Sensor 1	
5	PID 3Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	
6	PID 3Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	
7	PID 39h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	

54.10.33 Description of PID 21h - Distance travelled while MIL is activated

PID $21 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/ bit	External test equipment SI (Metric) / English display
21	**Distance Travelled While MIL is Activated**	A , B	0 km	65535 km	1 km per count	MIL_DIST: xxxxx km (xxxxx miles)
	Conditions for “Distance travelled” counter: • reset to $0000 when MIL state changes from deactivated to activated by this ECU • accumulate counts in km if MIL is activated (ON) • do not change value while MIL is not activated (OFF) • reset to $0000 if diagnostic information is cleared either by service $04 or 40 warm-up cycles without MIL activated • do not wrap to $0000 if value is $FFFF					

Data Byte	Application data to use	Formating conversion used
data A	MSB (DIST_ACT_MIL)	-
data B	LSB (DIST_ACT_MIL)	

54.10.34 Description of PID 22h - Fuel Rail Pressure relative to manifold vacuum (not supported)

54.10.35 Description of PID 23h - Fuel Rail Pressure

IF NC_INF_CONF <> 0 *(only supported for MSD)*

THEN

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
23	Fuel rail pressure	A , B	0 kPa	655350 kPa	10 kPa per bit unsigned, 1 kPA = 0.1450377 PSI	FRP: xxxxxx kPa (xxxxx.x PSI
	FRP shall display fuel rail pressure at the engine when the reading is referenced to atmosphere (gauge pressure)					

Data Byte	Application data to use	Formatting conversion used
data A	MSB (FUP_RNG_H_MES)	Shift to different range
data B	LSB (FUP_RNG_H_MES)	Shift to different range

ENDIF

54.10.36 Description of PID 24h - Bank 1 Sensor 1 (Wide range)

PID $24 definition

PID (hex)	Description Use if PID $13 or PID $1D is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
24	Equivalence Ratio (lambda) (B1-S1) Bank 1 – Sensor 1 (wide range O2S)	A , B	0	1.999	0.0000305	EQ_RATxy: x.xxx
	Oxygen Sensor Voltage (B1-S1)	C , D	0 V	7.999 V	0.000122 V	O2Sxy: x.xxx V
	PID $24 shall be used for linear or wide-ratio Oxygen Sensors when equivalence ratio and voltage are displayed					

IF PSN_LS = 11H or 33H
THEN

Data Byte	Application data to use	Formating conversion used
data A	MSB (LAMB_LS_UP[1])	Shift to different range
data B	LSB (LAMB_LS_UP[1])	=
data C	MSB (VLS_COR_LSL[1])	=
data D	LSB (VLS_COR_LSL[1])	=

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	=
data C	NONE	=
data D	NONE	=

ENDIF

54.10.37 Description of PID 25h - Bank 1 Sensor 2 (Wide range) (not supported)

54.10.38 Description of PID 26h - Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range) (not supported)

54.10.39 Description of PID 27h - Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range) (not supported)

54.10.40 Description of PID 28h - Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range)

PID $28 definition if PID $13 is supported

PID (hex)	Description Use if PID $13 is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
28	Equivalence Ratio (lambda) Bank 2 – Sensor 1 (wide range O2S)	A , B	0	1.999	0.0000305	EQ_RATxy: x.xxx
	Oxygen Sensor Voltage (B2-S1)	C , D	0 V	7.999 V	0.000122 V	O2Sxy: x.xxx V
	PIDs $28 shall be used for linear or wide-ratio Oxygen Sensors when equivalence ratio and voltage are displayed					

IF PSN_LS = 11H or 33H

THEN

Data Byte	Application data to use	Formating conversion used
data A	MSB (LAMB_LS_UP[2])	Shift to different range
data B	LSB (LAMB_LS_UP[2])	=
data C	MSB (VLS_COR_LSL[2])	=
data D	LSB (VLS_COR_LSL[2])	=

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	=
data C	NONE	=
data D	NONE	=

ENDIF

54.10.41 Description of PID 29h - Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range) (not supported)

54.10.42 Description of PID 2Ah - Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range) (not supported)

54.10.43 Description of PID 2Bh - Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range) (not supported)

54.10.44 Description of PID 2C - Commanded EGR

IF NC_INF_CONF <> 0 *(only supported for MSD)*

THEN

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
2C	Commanded EGR	A	0% (no flow)	100% (max. flow)	100/255 %	EGR_PCT: xxx.x%
	Commanded EGR displayed as a percent. EGR_PCT shall be normalised to the maximum EGR commanded output control parameter. EGR systems use a variety of methods to control the amount of EGR delivered to the engine. 1) If an on/off solenoid is used – EGR_PCT shall display 0% when the EGR is commanded off, 100% when the EGR system is commanded on. 2) If a vacuum solenoid is duty cycled, the EGR duty cycle from 0 to 100% shall be displayed. 3) If a linear or stepper motor valve is used, the fully closed position shall be displayed as 0%, the fully open position shall be displayed as 100%. Intermediate positions shall be displayed as a percent of the full-open position. For example, a stepper-motor EGR valve that moves from 0 to 128 counts shall display 0% at 0 counts, 100% at 128 counts and 50% at 64 counts. 4) Any other actuation method shall be normalised to display 0% when no EGR is commanded and 100% at the maximum commanded EGR position.					

Data Byte	Application data to use	Formating conversion used
data A	OPG_SP_ACR	-

ENDIF

54.10.45 Description of PID 2D definition - EGR Error

IF NC_INF_CONF <> 0 *(only supported for MSD)*

THEN

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
2D	EGR Error = (EGR actual – EGR commanded) / EGR commanded * 100%	A	-100 % (less than commanded)	+99.22 % (more than commanded)	100/128 % (0 % at 128)	EGR_ERR: xxx.x%

EGR error, as a percent of commanded EGR. Often, EGR valve control outputs are not in the same engineering units as the EGR feedback input sensors. For example, an EGR valve can be controlled using a duty-cycled vacuum solenoid, however, the feedback input sensor is a position sensor. This makes it impossible to display "actual" versus "commanded" in the same engineering units. EGR error solved this problem by displaying a normalised (non-dimensional) EGR system feedback parameter. EGR error is defined to be:

(actual EGR – commanded EGR) / commanded EGR.

For example if 10% EGR is commanded and 5% is delivered to the engine, the EGR_ERR is (5% - 10%) / 10 = -50% error.

EGR_ERR may be computed using various control parameters such as position, steps, counts, etc. All EGR systems must react to quickly changing conditions in the engine; therefore, EGR_ERR will generally show errors during transient conditions. Under steady condition, the error will be minimized (no necessarily zero, however) if the EGR system is under control.

If the control system does not use closed loop control, EGR_ERR shall not be supported.

Data Byte	Application data to use	Formating conversion used
data A	OBD_EGR_DIF	-

ENDIF

54.10.46 Description of PID 2Eh - Commanded Evaporative Purge

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
2E	Commanded Evaporative Purge	A	0% no flow	100% max. flow	100/255 %	EVAP_PCT: xxx.x %
	Commanded evaporative purge control valve displayed as a percent. EVAP_PCT shall be normalised to the maximum EVAP purge commanded output control parameter. 1) If an on/off solenoid is used – EVAP_PCT shall display 0% when purge is commanded off, 100% when purge is commanded on. 2) If a vacuum solenoid is duty cycled, the EVAP purge valve duty cycle from 0 to 100% shall be displayed. 3) If a linear or stepper motor valve is used, the fully closed position shall be displayed as 0%, the fully open position shall be displayed as 100%. Intermediate positions shall be displayed as a percent of the full-open position. For example, a stepper-motor EVAP purge valve that moves from 0 to 128 counts shall display 0% at 0 counts, 100% at 128 counts and 50% at 64 counts. 4) Any other actuation method shall be normalised to display 0% when no purge is commanded and 100% at the maximum commanded purge position/flow.					

Data Byte	Application data to use	Formating conversion used
data A	CPPWM_CPS	-

54.10.47 Description of PID 2Fh - Fuel Level Input

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
2F	Fuel Level Input	A	0% no fuel	100% max. fuel capacity	100/255 %	FLI: xxx.x %
	FLI shall indicate nominal fuel tank liquid fill capacity as a percent of maximum, if utilised by the control module for OBD monitoring . FLI may be obtained directly from a sensor, may be obtained indirectly via the vehicle serial data communication bus, or may be inferred by the control strategy using other sensor inputs. Vehicles that use gaseous fuels shall display the percent of useable fuel capacity.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_FTL	no

54.10.48 Description of PID 30h - Number of warm-ups since diagnostic trouble codes cleared

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
30	Number of warm-ups since diagnostic trouble codes cleared	A	0	255	1 warm-up per count	WARM_UPS: xxx
	Number of OBD warm-up cycles since all DTCs were cleared (via an external test eqiupment or possibly, a battery disconnect). A warm-up is defined in the OBD regulations to be sufficient vehicle operation such that coolant temperature rises by at least 22 °C (40 °F) from engine starting and reaches a minimum temperature of 70 °C (160 deg °F) (60 °C (140 °F) for diesels). This PID is not associated with any particular DTC. It is simply an indication for I/M, of the last time an external test equipment was used to clear DTCs. If greater than 255 warm ups have occurred, WARM_UPS shall remain at 255 and not wrap to zero.					

Data Byte	Application data to use	Formating conversion used
data A	CTR_WUP_DTC_CLR	-

54.10.49 Description of PID 31h - Distance since diagnostic trouble codes cleared

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
31	Distance since diagnostic trouble codes cleared (same scaling as PID $21)	A , B	0 km	65,535 km	1 km per count	CLR_DIST: xxxxx km (xxxxx miles)
	Distance accumulated since DTCs were cleared (via an external test equipment or possibly, a battery disconnect). This PID is not associated with any particular DTC. It is simply an indication for I/M (Inspection/Maintenance), of the last time an external test equipment was used to clear DTCs. If greater than 65,535 km have occurred, CLR_DIST shall remain at 65,535 km and not wrap to zero.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(DIST_DTC_CLR)	-
data B	LSB(DIST_DTC_CLR)	

54.10.50 Description of PID 32h - Evap System Vapor Pressure (not supported)

54.10.51 Description of PID 33h - Barometric Pressure

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
33	Barometric Pressure	A	0 kPa (absolute)	255 kPa (absolute)	1 kPa per bit (absolute)	BARO: xxx kPa (xx.x inHg)
	Barometric pressure utilised by the control module. BARO is normally obtained from a dedicated BARO sensor, from a MAP sensor at key-on and during certain modes of driving, or inferred from a MAF sensor and other inputs during certain modes of driving. The control module shall report BARO from whatever source it is derived from. NOTE Some weather services report local BARO values adjusted to sea level. In these cases, the reported value may not match the displayed value on the external test equipment. NOTE If BARO is inferred while driving and stored in non-volatile RAM or Keep-alive RAM, BARO may not be accurate after a battery disconnect or total memory clear.					

Data Byte	Application data to use	Formating conversion used
data A	AMP_MES	Shift to different range

54.10.52 Description of PID 34h - Bank 1 Sensor 1 (Wide range)

PID $34 definition

PID (hex)	Description Use if PID $13 or PID $1D is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
34	Equivalence Ratio (lambda) Bank 1 - Sensor 1 (wide range O2S)	A , B	0	1.999	0.0000305	EQ_RATxy: x.xxx
	Oxygen Sensor Current (B1-S1)	C , D	-128 mA	127.996 mA	0.00390625 mA ($8000 = 0 mA)	O2Sxy: x.xxx mA
	PID $34 shall be used for linear or wide-ratio Oxygen Sensors when equivalence ratio and current are displayed					

IF PSN_LS = 11H or 33H

THEN

Data Byte	Application data to use	Formating conversion used
data A	LAMB_LS_UP[1]	-
data B	LAMB_LS_UP[1]	
data C	IPLSL_COR[1]	-
data D	IPLSL_COR[1]	

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	-
data B	NONE	
data C	NONE	-
data D	NONE	

ENDIF

54.10.53 Description of PID 35h - Bank 1 Sensor 2 (Wide range) (not supported)

54.10.54 Description of PID 36h - Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range) (not supported)

54.10.55 Description of PID 37h - Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range) (not supported)

54.10.56 Description of PID 38h - Bank 2 Sensor 1 (Wide range)

PID $38 definition if PID $13 is supported

PID (hex)	Description Use if PID $13 is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
38	Equivalence Ratio (lambda) Bank 2 - Sensor 1 (wide range O2S)	A , B	0	1.999	0.0000305	EQ_RATxy: x.xxx
	Oxygen Sensor Current (B2-S1)	C , D	-128 mA	127.996 mA	0.00390625 mA ($8000 = 0 mA)	O2Sxy: x.xxx mA
	PIDs $38 shall be used for linear or wide-ratio Oxygen Sensors when equivalence ratio and current are displayed					

IF PSN_LS = 11H or 33H

THEN

Data Byte	Application data to use	Formating conversion used
data A	LAMB_LS_UP[2]	-
data B	LAMB_LS_UP[2]	
data C	IPLSL_COR[2]	-
data D	IPLSL_COR[2]	

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	-
data B	NONE	
data C	NONE	-
data D	NONE	

ENDIF

54.10.57 Description of PID 39h - Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range) (not supported)

54.10.58 Description of PID 3Ah - Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range) (not supported)

54.10.59 Description of PID 3Bh - Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range) (not supported)

54.10.60 Description of PID 3C - Catalyst Temperature Bank 1, Sensor 1

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
3C	Catalyst Temperature Bank 1, Sensor 1	A , B	-40 °C	+6513.5 °C	0.1 °C / bit with -40 °C offset	CATEMP11: xxxx.x °C (xxxx.x °F)
	CATEMP11 shall display catalyst substrate temperature for a bank 1 catalyst, if utilised by the control module strategy for OBD monitoring, or the Bank 1, Sensor 1 catalyst temperature sensor. CATEMP11 may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(TEG_CAT_UP_MDL[1])	Shift to different range =
data B	LSB(TEG_CAT_UP_MDL[1])	

54.10.61 Description of PID 3D - Catalyst Temperature Bank 2, Sensor 1

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
3D	Catalyst Temperature Bank 2, Sensor 1	A , B	-40 °C	+6513.5 °C	0.1 °C / bit with -40 °C offset	CATEMP21: xxxx.x °C (xxxx.x °F)
	CATEMP21 shall display catalyst substrate temperature for a bank 2 catalyst, if utilised by the control module strategy for OBD monitoring, or the Bank 2, Sensor 1 catalyst temperature sensor. CATEMP21 may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(TEG_CAT_UP_MDL[2])	Shift to different range =
data B	LSB(TEG_CAT_UP_MDL[2])	

54.10.62 Description of PID 3E - Catalyst Temperature Bank 1, Sensor 2

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
3E	Catalyst Temperature Bank 1, Sensor 2	A , B	-40 °C	+6513.5 °C	0.1 °C / bit with -40 °C offset	CATEMP12: xxxx.x °C (xxxx.x °F)
	CATEMP12 shall display catalyst substrate temperature for an additional bank 1 catalyst, if utilised by the control module strategy for OBD monitoring, or the Bank 1, Sensor 2 catalyst temperature sensor. CATEMP12 may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(TEG_CAT_DOWN_MDL[1])	Shift to different range =
data B	LSB(TEG_CAT_DOWN_MDL[1])	

54.10.63 Description of PID 3F Catalyst Temperature Bank 2, Sensor 2

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
3F	Catalyst Temperature Bank 2, Sensor 2	A , B	-40 °C	+6513.5 °C	0.1 °C / bit with -40 °C offset	CATEMP22: xxxx.x °C (xxxx.x °F)
	CATEMP22 shall display catalyst substrate temperature for an additional bank 2 catalyst, if utilised by the control module strategy for OBD monitoring, or the Bank 2, Sensor 2 catalyst temperature sensor. CATEMP22 may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(TEG_CAT_DOWN_MDL[2])	Shift to different range =
data B	LSB(TEG_CAT_DOWN_MDL[2])	

54.10.64 Description of PID 40h - Supported PIDs 41h - 60h

0 : the PID is not used
1 : the PID is used

DATA A - Supported PIDs 41h - 48h

PID 40h - DATA A - Supported PID 41h to 48h

BIT	Function	Name
0	PID 48h : Absolute Throttle Position C	MOD_1_PID_9_DIAG
1	PID 47h : Absolute Throttle Position B	
2	PID 46h : Ambient air temperature	
3	PID 45h : Relative Throttle Position	
4	PID 44h : Command Equivalence Ratio	
5	PID 43h : Absolue Load Value	
6	PID 42h : Control module voltage	
7	PID 41h : Monitor status this driving cycle	

PID 40h - DATA B - Supported PID 49h to 50h

BIT	Function	Name
0	PID 50h : Reserved	MOD_1_PID_A_DIAG
1	PID 4Fh : Reserved	
2	PID 4Eh : Time since diagnostic trouble codes cleared	
3	PID 4Dh : Minutes run by the engine while MIL activated	
4	PID 4Ch : Commanded Throttle Actuator Control	
5	PID 4Bh : Accelerator Pedal Position F	
6	PID 4Ah : Accelerator Pedal Position E	
7	PID 49h : Accelerator Pedal Position D	

DATA C - Supported PIDs 51h to 58h

PID 40h - DATA C - Supported PID 51h to 58h

BIT	Function	Name
0	PID 58h : Reserved	MOD_1_PID_B_DIAG
1	PID 57h : Reserved	
2	PID 56h : Reserved	
3	PID 55h : Reserved	
4	PID 54h : Reserved	
5	PID 53h : Reserved	
6	PID 52h : Reserved	
7	PID 51h : Reserved	

DATA D - Supported PIDs 59h to 60h

PID 40h - DATA D - Supported PID 59h to 60h		
BIT	**Function**	**Name**
0	PID 60h : Reserved	MOD_1_PID_C_DIAG
1	PID 5Fh : Reserved	
2	PID 5Eh : Reserved	
3	PID 5Dh : Reserved	
4	PID 5Ch : Reserved	
5	PID 5Bh : Reserved	
6	PID 5Ah : Reserved	
7	PID 59h : Reserved	

54.10.65 Description of PID 41 - Monitor status this driving cycle

PID $41 definition

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
	The bit in this PID shall report two pieces of information for each monitor: 1) Monitor enable status for the current driving cycle. This bit shall indicate when a monitor is disabled in a manner such that there is no way for the driver to operate the vehicle for the remainder of the driving cycle and make the monitor run. Typical examples are: Engine-off soak not long enough (e.g., cold start temperature conditions not satisfied) Monitor maximum time limit or number of attempts/aborts exceeded Ambient air temperature too low or too high BARO too low (high altitude) The monitor shall not indicate "disabled" for operator-controlled conditions such as rpm, load, throttle position, minimum time limit not exceeded, ECT, TP, etc. 2) Monitor completion status for the current driving/monitoring cycle. Status shall be reset to "not complete" upon starting a new monitoring cycle. Note that some monitoring cycles can include various engine-operating conditions; other monitoring cycles begin after the ignition key is turned off. Some status bits on a given vehicle can utilise engine-running monitoring cycles while others can utilise engine-off monitoring cycles. Resetting the bits to "not complete" upon starting the engine will accommodate most engine-running and engine-off monitoring cycles, however, manufacturers are free to define their own monitoring cycles. NOTE PID $41 bits shall be utilised for all non-continuous monitors which are supported, and change completion status in PID $01. If a non-continuous monitor is not supported or always shows "complete", the corresponding PID $41 bits shall indicate disabled and complete. PID $41 bits may be utilised at the vehicle manufacturer's discretion for all continuous monitors which are supported with the exception of bit 03 which shall always showed CCM as enabled for spark ignition and compression ignition engines.			

PID $41 definition (continued)

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
		A (bit)	**byte 1 of 4**	
	Reserved - shall be reported as $00	0-7		---
	Enable status of continuous monitors this monitoring cycle:	**B (bit)**	byte 2 of 4 (Low Nibble)	Enable status of continuous monitors this monitoring cycle: NO means disabled for rest of this monitoring cycle or not supported in PID $01, YES means enabled for this monitoring cycle.
	Misfire monitoring	0	0 = monitor disabled for rest of this monitoring cycle or not supported (NO) 1 = monitor enabled for this monitoring cycle (YES)	MIS_ENA: NO or YES
	Fuel system monitoring	1		FUEL_ENA: NO or YES
	Comprehensive component monitoring	2		CCM_ENA: YES
	reserved (bit shall be reported as ‘0’)	3		

PID $41 definition (continued)

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
	Completion status of continuous monitors this monitoring cycle:	**B (bit)**	**byte 2 of 4 (High Nibble)**	**Completion status of continuous monitors this monitoring cycle:**
	Misfire monitoring C_CLC_READY_OBD_1[0]	4	See PID $01 to determine which monitors are supported 0 = monitor complete this monitoring cycle, or not supported (YES) 1 = monitor not complete this monitoring cycle (NO)	MIS_CMPL: YES or NO
	Fuel system monitoring C_CLC_READY_OBD_1[1]	5		FUELCMPL: YES or NO
	Comprehensive component monitoring C_CLC_READY_OBD_1[2]	6		CCM_CMPL: YES or NO
	reserved (bit shall be reported as ‘0’) C_CLC_READY_OBD_1[3]	7		

PID $41 definition (continued)

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
	Enable status of non-continuous monitors this monitoring cycle:	**C (bit)**	**byte 3 of 4**	**Enable status of non-continuous monitors this monitoring cycle:**
	Catalyst monitoring	0	0 = monitor disabled for rest of this monitoring cycle (NO) 1 = monitor enabled for this monitoring cycle (YES)	CAT_ENA: YES or NO
	Heated catalyst monitoring	1		HCAT_ENA: YES or NO
	Evaporative system monitoring	2		EVAP_ENA: YES or NO
	Secondary air system monitoring	3		AIR_ENA: YES or NO
	A/C system refrigerant monitoring	4		ACRF_ENA: YES or NO
	Oxygen sensor monitoring	5		O2S_ENA: YES or NO
	Oxygen sensor heater monitoring	6		HTR_ENA: YES or NO
	EGR system monitoring	7		EGR_ENA: YES or NO

PID $41 definition (continued)

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
	Completion status of non-continuous monitors this monitoring cycle:	**D (bit)**	**byte 4 of 4**	**Completion status of non-continuous monitors this monitoring cycle:**
	Catalyst monitoring	0	See PID $01 to determine which monitors are supported. 0 = monitor complete this monitoring cycle, or not supported (YES) 1 = monitor not complete this monitoring cycle (NO)	CAT_CMPL: YES or NO
	Heated catalyst monitoring	1		HCATCMPL: YES or NO
	Evaporative system monitoring	2		EVAPCMPL: YES or NO
	Secondary air system monitoring	3		AIR_CMPL: YES or NO
	A/C system refrigerant monitoring	4		ACRFCMPL: YES or NO
	Oxygen sensor monitoring	5		O2S_CMPL: YES or NO
	Oxygen sensor heater monitoring	6		HTR_CMPL: YES or NO
	EGR system monitoring	7		EGR_CMPL: YES or NO

Data Byte	Application data to use	Formating conversion used
data A	0x00	-
data B	C_CLC_READY_OBD_1[0...3] + STATE_ENA_OBD[0...3]	
data C	STATE_ENA_OBD[8..15]	
data D	STATE_CMPL_OBD[8..15]	

54.10.66 Description of PID 42 - Control module voltage

PID $42 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
42	Control module voltage	A , B	0 V	65.535 V	0.001 V per bit	VPWR: xx.xxx V
	VPWR – power input to the control module. VPWR is normally battery voltage, less any voltage drop in the circuit between the battery and the control module. NOTE 42-volt vehicles may utilise multiple voltages for different systems on the vehicle. VPWR represents the voltage at the control module; it may be significantly different than battery voltage.					

Data Byte	Application data to use	Formating conversion used
data A	MSB (VB)	Shift to different range
data B	LSB (VB)	

54.10.67 Description of PID 43 - Absolute Load Value

PID $43 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
43	Absolute Load Value	A , B	0 %	25700%	100/255 %	LOAD_ABS: xxx.x%
	The absolute load value has some different characteristics than the LOAD_PCT defined in PID 04 This definition, although restrictive, will standardise the calculation. LOAD_ABS is the normalised value of air mass per intake stroke displayed as a percent. LOAD_ABS = [air mass (g/intake stroke)] / [1.184 (g/intake stroke) * cylinder displacement in litres] Derivation: air mass (g / intake stroke) = [total engine air mass (g/sec)] / [rpm (revs/min)* (1 min / 60 sec) * (1/2 # of cylinders (strokes /rev)] LOAD_ABS = [air mass (g)/intake stroke] / [maximum air mass (g)/intake stroke at WOT@STP at 100% volumetric efficiency] * 100% Where: STP = Standard Temperature and Pressure = 25 ^{o}C, 29.92 in Hg (1013 hPa) BARO, WOT = wide open throttle The quantity (maximum air mass (g)/intake stoke at WOT@STP at 100% volumetric efficiency) is a constant for a given cylinder swept volume. The constant is 1.184 (g/litre 3) * cylinder displacement (litre 3/intake stroke) based on air density at STP Characteristics of LOAD_ABS are: Ranges from 0 to approximately 0.95 for naturally aspirated engines, 0 – 4 for boosted engines Linearly correlated with engine indicated and brake torque Often used to schedule spark and EGR rates Peak value of LOAD_ABS correlates with volumetric efficiency at WOT. Indicates the pumping efficiency of the engine for diagnostic purposes. Spark ignition engine are required to support PID $43. Compression ignition (diesel) engines are not required to support this PID. NOTE See PID $04 for an additional definition of engine LOAD..					

Data Byte	Application data to use	Formating conversion used
data A	MSB (LOAD_ABSV)	no
data B	LSB (LOAD_ABSV)	no

54.10.68 Description of PID 44 - Commanded Equivalence Ratio

Table .1 — PID $44 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
44	Commanded Equivalence Ratio	A , B	0	1.999	0.0000305	EQ_RAT: x.xxx
	Fuel systems that utilise conventional oxygen sensor shall display the commanded open loop equivalence ratio while the fuel control system is in open loop. EQ_RAT shall indicate 1.0 while in closed loop fuel. Fuel systems that utilise wide-range/linear oxygen sensors shall display the commanded equivalence ratio in both open loop and closed loop operation. To obtain the actual A/F ratio being commanded, multiply the stoichiometric A/F ratio by the equivalence ratio. For example, for gasoline, stoichiometric is 14.64:1 ratio. If the fuel control system was commanding an 0.95 EQ_RAT, the commanded A/F ratio to the engine would be 14.64 * 0.95 = 13.9 A/F					

Data Byte	Application data to use	Formating conversion used
data A	MSB(LAMB_SP_HOM[1])	Shift to different resolution =
data B	LSB(LAMB_SP_HOM[1])	

54.10.69 Description of PID 45 - Relative Throttle Position

PID $45 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
45	Relative Throttle Position	A	0 %	100 %	100/255 %	TP_R: xxx.x %
	Relative or "learned" throttle position shall be displayed as a normalised value, scaled from 0 to 100%. TP_R should display a value of 0% at the "learned closed-throttle position. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed throttle position is a 1.0 volts, TP shall display (1.0 – 1.0 / 5.0) = 0% at closed throttle and 30% at 2.5 volts. Because of the closed-throttle offset, wide open throttle will usually indicate substantially less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage. See PID $11 for a definition of Absolute Throttle Position.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TPS_REL	-

54.10.70 Description of PID 46 - Ambient air temperature

PID $46 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
46	Ambient air temperature (same scaling as IAT - $0F)	A	-40 °C	+215 °C	1 °C with -40 °C offset	AAT: xxx °C / xxx °F
	AAT shall display ambient air temperature, if utilised by the control module strategy for OBD monitoring. AAT may be obtained directly from a sensor, may be obtained indirectly via the vehicle serial data communication bus, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TAM	-

54.10.71 Description of PID 47 - Absolute Throttle Position B

PID $47 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
47	Absolute Throttle Position B	A	0 %	100 %	100/255 %	TP_B: xxx.x %
	Absolute throttle position B, if utilised by the control module, (not “relative” or “learned” throttle position) shall be displayed as a normalised value, scaled from 0 to 100%. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed throttle position is a 1.0 volts, TP_B shall display (1.0 / 5.0) = 20% at closed throttle and 50% at 2.5 volts. Throttle position at idle will usually indicate greater than 0%, and throttle position at wide open throttle will usually indicate less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TPS_2	-

54.10.72 Description of PID 48 - Absolute Throttle Position C (not supported)

54.10.73 Description of PID 49 - Accelerator Pedal Position D

PID $49 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
49	Accelerator Pedal Position D	A	0 %	100 %	100/255 %	APP_D: xxx.x %
	Accelerator Pedal Position D, if utilised by the control module, (not “relative” or “learned” pedal position) shall be displayed as a normalised value, scaled from 0 to 100%. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed pedal is 1.0 volt, APP_D shall display (1.0 / 5.0) = 20% at closed pedal and 50% at 2.5 volts. Pedal position at idle will usually indicate greater than 0%, and pedal position at wide open pedal will usually indicate less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_PV_1	-

54.10.74 Description of PID 4A - Accelerator Pedal Position E

PID $4A definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
4A	Accelerator Pedal Position E	A	0 %	100 %	100/255 %	APP_E: xxx.x %
	Accelerator Pedal Position E, if utilised by the control module, (not “relative” or “learned” pedal position) shall be displayed as a normalised value, scaled from 0 to 100%. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed pedal is 1.0 volt, APP_E shall display (1.0 / 5.0) = 20% at closed pedal and 50% at 2.5 volts. Pedal position at idle will usually indicate greater than 0%, and pedal position at wide open pedal will usually indicate less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_PV_2	-

54.10.75 Description of PID 4B - Accelerator Pedal Position F (not supported)

54.10.76 Description of PID 4C - Commanded Throttle Actuator Control

PID $4C definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
4C	Commanded Throttle Actuator Control	A	0% (closed throttle)	100% (wide open throttle)	100/255 %	TAC_PCT: xxx.x%
	Commanded TAC displayed as a percent. TAC_PCT shall be normalised to the maximum TAC commanded output control parameter. TAC systems use a variety of methods to control the amount of throttle opening. 1) If a linear or stepper motor is used, the fully closed throttle position shall be displayed as 0%, the fully open throttle position shall be displayed as 100%. Intermediate positions shall be displayed as a percent of the full-open throttle position. For example, a stepper-motor TAC that moves the throttle from 0 to 128 counts shall display 0% at 0 counts, 100% at 128 counts and 50% at 64 counts. 2) Any other actuation method shall be normalised to display 0% when the throttle is commanded closed and 100% when the throttle is commanded open.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TPS_SP	-

54.10.77 Description of PID 4D - Minutes run by the engine while MIL activated

PID $4D definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/ bit	External test equipment SI (Metric) / English display
4D	Minutes run by the engine while MIL activated	A, B	0 min	65535 min 1092.25 hours	1 min per count	MIL_TIME: xxxx hrs, xx min
	Conditions for “Minutes run by the engine while MIL activated” counter: • reset to $0000 when MIL state changes from deactivated to activated by this ECU • accumulate counts in minutes if MIL is activated (ON) • do not change value while MIL is not activated (OFF) • reset to $0000 if diagnostic information is cleared either by service $04 or 40 warm-up cycles without MIL activated • do not wrap to $0000 if value is $FFFF					

Data Byte	Application data to use	Formating conversion used
data A	MSB (T_ACT_MIL_SAE)	-
data B	LSB (T_ACT_MIL_SAE)	

54.10.78 Description of PID 4E - Time since diagnostic trouble codes cleared

PID $4E definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
4E	Time since diagnostic trouble codes cleared	A, B	0 min	65535 min 1092.25 hours	1 min per count	CLR_TIME: xxxx hrs, xx min
	Time accumulated since DTCs were cleared (via an external test equipment or possibly, a battery disconnect). This PID is not associated with any particular DTC. It is simply an indication for I/M (Inspection/Maintenance), of the last time an external test equipment was used to clear DTCs. If greater than 65,535 min have occurred, CLR_TIME shall remain at 65,535 min and not wrap to zero.					

Data Byte	Application data to use	Formating conversion used
data A	MSB (T_DTC_CLR)	-
data B	LSB (T_DTC_CLR)	

54.10.79 Description of PID 4F up to FFh - Reserved

PID $4F - $FF definition

PID (hex)	Description	Data Byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
4F - FF	Reserved by document	---	---	---	---	---

54.11 PID description for mode 02h

Input data:

MOD_2_PID_1_DIAG	MOD_2_PID_2_DIAG	MOD_2_PID_3_DIAG	MOD_2_PID_4_DIAG
MOD_2_PID_5_DIAG	MOD_2_PID_6_DIAG	MOD_2_PID_7_DIAG	MOD_2_PID_8_DIAG
MOD_2_PID_9_DIAG	MOD_2_PID_A_DIAG	MOD_2_PID_B_DIAG	MOD_2_PID_C_DIAG
PSN_LS	NC_INJ_CONF		

Import action:

```
ACTION_ERRM_ReadFrfByDtc(
                                   IN <TypeOfFF>,
                                   IN <FFIdentifier>,
                                   IN <DtcIdentifier>,
                                   IN < LevelOfDtc>,
                                   INOUT <Frf>,
                                   OUT <ResultFrf>
                                  )
```

This API returns a freeze frame or a part of freeze frame.

General information:

This specification is according with the ISO 15031–5 issue 8.

" IDX" is used instead of the array definition [NC_NR_ERR_DYN].

At PSN_LS you can see the configuration of the oxygen-sensors:

PSN_LS:

BIT	Function	Status
0	Bank 1 Sensor 1	1= available, 0 = not available
1	Bank 1 Sensor 2	1= available, 0 = not available
2	Bank 1 Sensor 3	1= available, 0 = not available
3	Bank 1 Sensor 4	1= available, 0 = not available
4	Bank 2 Sensor 1	1= available, 0 = not available
5	Bank 2 Sensor 2	1= available, 0 = not available
6	Bank 2 Sensor 3	1= available, 0 = not available
7	Bank 2 Sensor 4	1= available, 0 = not available

54.11.1 Description of PID 00h - Supported PIDs 01h - 20h

0 : the PID is not used
1 : the PID is used

DATA A - Supported PIDs 01h to 08h

PID 00h - DATA A - Supported PID 01h to 08h		
BIT	**Function**	**Name**
0	PID 08h : Short term fuel trim - Bank 2 and Bank 4	MOD_2_PID_1_DIAG
1	PID 07h : Long term fuel trim - Bank 1 and Bank 3	
2	PID 06h : Short term fuel trim - Bank 1 and Bank 3	
3	PID 05h : Engine coolant temperature	
4	PID 04h : Calculated load value	
5	PID 03h : Fuel system status	
6	PID 02h : Trouble code that caused required freeze frame	
7	PID 01h : Monitor status since DTCs cleared	

DATA B - Supported PIDs 09h to 10h

PID 00h - DATA B - Supported PID 09h to 10h		
BIT	**Function**	**Name**
0	PID 10h : Air flow rate from MAF sensor	MOD_2_PID_2_DIAG
1	PID 0Fh : Intake air temperature	
2	PID 0Eh : Ignition time advance	
3	PID 0Dh : Vehicle speed	
4	PID 0Ch : Engine RPM	
5	PID 0Bh : Intake manifold absolute pressure	
6	PID 0Ah : Fuel pressure (gauge)	
7	PID 09h : Long term fuel trim - Bank 2	

DATA C - Supported PIDs 11h to 18h

PID 00h - DATA C - Supported PID 11h to 18h		
BIT	**Function**	**Name**
0	PID 18h : Bank 2 Sensor 1 or Bank 3 Sensor 1	MOD_2_PID_3_DIAG
1	PID 17h : Bank 1 Sensor 4 or Bank 2 Sensor 2	
2	PID 16h : Bank 1 Sensor 3 or Bank 2 Sensor 1	
3	PID 15h : Bank 1 Sensor 2	
4	PID 14h : Bank 1 Sensor 1	
5	PID 13h : Location of Oxygen Sensors	
6	PID 12h : Secondary air status	
7	PID 11h : Absolute throttle position sensor	

DATA D - Supported PIDs 19h to 20h

PID 00h - DATA D - Supported PID 19h to 20h		
BIT	**Function**	**Name**
0	PID 20h : PIDs supported ($21 - $40)	MOD_2_PID_4_DIAG
1	PID 1Fh : Time Since Engine Start	
2	PID 1Eh : Auxiliary Input Status	
3	PID 1Dh : Location of oxygen sensors	
4	PID 1Ch : OBD requirements to which vehicle is designed	
5	PID 1Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2	
6	PID 1Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1	
7	PID 19h : Bank 2 Sensor 2 or Bank 3 Sensor 2	

54.11.1.1 Description of PID 02h - Diagnostic trouble codes

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling	External test equipment SI (Metric) / English display
02	DTC that caused required freeze frame data storage ($0000 indicates no freeze frame data)	A, B	00 00	FF FF	Hexadecimal e.g. P01AB (DTCs defined in ISO 15031-6)	DTCFRZF: Pxxxx, Cxxxx, Bxxxx, Uxxxx

<u>**Formula section:**</u>

CALL API :

```
ACTION_ERRM_ReadFrfByDtc        (
                                IN <FRF_LAW>,
                                IN <FIRST>,
                                IN <NO_DTC>,
                                IN <DTC_LAW>
                                INOUT <Frf>
                                OUT <ResultFrf>
                                )
```

IF ResultFrf == NO_FRF_PRESENT

THEN

Data Byte	Application data to use	Formating conversion used
data A	`0x00`	-
data B	`0x00`	

ELSE

Data Byte	Application data to use	Formating conversion used
data A	`Frf[0]`	-
data B	`Frf[1]`	

ENDIF

54.11.1.2 Description of PID 03h - Status of fuel system

```
If      PSN_LS = 11H or 33H              // upstream sensors available
Then    DATA_A = ENVD_OBD_1_IDX          // FRF of STATE_LS_1
        DATA_B = ENVD_OBD_2_IDX          //FRF of STATE_LS_2
Else    PID 03h is not supported
Endif
Else
```

54.11.1.3 Description of PID 04h - Calculated load value

```
DATA_A = ENVD_OBD_3_IDX          // FRF of LOAD_CLC
```

54.11.1.4 Description of PID 05h - Engine coolant temperature

```
DATA_A = ENVD_OBD_4_IDX          // FRF of TCO_MES
```

54.11.1.5 Description of PID 06h - Short term fuel trim - Bank 1 and Bank 3

```
If      PSN_LS = 11H or 33H      // upstream sensors available
Then    PID 06h is supported
        DATA_A = ENVD_OBD_5_IDX  // FRF of OBD_LAM_COR_1
        DATA_B = NONE            //Bank 3 not supported
Else    PID 06h is not supported
Endif
```

54.11.1.6 Description of PID 07h - Long term fuel trim - Bank 1 and Bank 3

```
If      PSN_LS = 33H             // upstream and downstream sensors available
Then    PID 07h is supported
        DATA_A = ENVD_OBD_6_IDX  // FRF of OBD_LAM_AD_1
        DATA_B = NONE            //Bank 3 not supported
Else    PID 07h is not supported
Endif
```

54.11.1.7 Description of PID 08h - Short term fuel trim - Bank 2 and Bank 4

```
If      PSN_LS = 11H or 33H      // upstream sensors available
Then    PID 08h is supported
        DATA_A = ENVD_OBD_7_IDX  // FRF of OBD_LAM_COR_2
        DATA_B = NONE            //Bank 3 not supported
Else    PID 08h is not supported
Endif
```

54.11.1.8 Description of PID 09h - Long term fuel trim - Bank 2 and Bank 4

```
If      PSN_LS = 33H             // upstream and downstream sensors available
Then    PID 09h is supported
        DATA_A = ENVD_OBD_8_IDX  // FRF of OBD_LAM_AD_2
        DATA_B = NONE            //Bank 3 not supported
Else    PID 07h is not supported
Endif
```

54.11.1.9 Description of PID 0Ah - Fuel pressure (gage)

DATA_A = ENVD_OBD_9_IDX *// FRF of OBD_FUP*

54.11.1.10 Description of PID 0Bh - Intake manifold absolute pressure

DATA_A = ENVD_OBD_10_IDX *// FRF of MAP_SAE*

54.11.1.11 Description of PID 0Ch - Engine speed

DATA_A = ENVD_OBD_11_IDX *// FRF of OBD_N (high byte)*

DATA_B = ENVD_OBD_12_IDX *// FRF of OBD_N (low byte)*

54.11.1.12 Description of PID 0Dh - Vehicle speed

DATA_A = ENVD_OBD_13_IDX *// FRF of VS*

54.11.1.13 Description of PID 0Eh - Ignition timing advance

DATA_A = ENVD_OBD_14_IDX *// FRF of OBD_IGA_IGC, cylinder 1*

54.11.1.14 Description of PID 0Fh - Intake air temperature

DATA_A = ENVD_OBD_15_IDX *// FRF of OBD_TIA*

54.11.1.15 Description of PID 10h - Air flow rate from MAF

DATA_A = ENVD_OBD_16_IDX *// FRF of OBD_MAF (high byte)*

DATA_B = ENVD_OBD_17_IDX *// FRF of OBD_MAF (low byte)*

54.11.1.16 Description of PID 11h - Absolute throttle position sensor

DATA_A = ENVD_OBD_18_IDX *// FRF of OBD_TPS_1*

54.11.1.17 Description of PID 12h - Commanded secondary air status

DATA_A = ENVD_OBD_19_IDX *// FRF of STATE_OBD_SA*

54.11.1.18 Description of PID 1Fh - Time Since Engine Start

DATA_A = ENVD_OBD_20_IDX *// FRF of T_AST_SAE (high byte)*

DATA_B = ENVD_OBD_21_IDX *// FRF of T_AST_SAE (low byte)*

54.11.2 Description of PID 20h - Supported PIDs 21h - 40h

0 : the PID is not used, 1 : the PID is used

DATA A - Supported PIDs 21h - 28h

PID 20h - DATA A - Supported PID 21h to 28h

BIT	Function	Name
0	PID 28h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	MOD_2_PID_5_DIAG
1	PID 27h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	
2	PID 26h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	
3	PID 25h : Bank 1 Sensor 2 (Wide range O2S)	
4	PID 24h : Bank 1 Sensor 1 (Wide range O2S)	
5	PID 23h : Diesel fuel pressure (gauge)	
6	PID 22h : Relative fuel pressure	
7	PID 21h : Distance travelled while MIL is activated	

DATA B - Supported PIDs 29h to 30h

PID 20h - DATA B - Supported PID 29h to 30h

BIT	Function	Name
0	PID 30h : Number of warm-ups since diagnostic trouble codes cleared	MOD_2_PID_6_DIAG
1	PID 2Fh : Fuel Level Input	
2	PID 2Eh : Commanded Evaporative Purge	
3	PID 2Dh : EGR Error	
4	PID 2Ch : Command EGR	
5	PID 2Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	
6	PID 2Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	
7	PID 29h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	

DATA C - Supported PIDs 31h to 38h

PID 20h - DATA C - Supported PID 31h to 38h

BIT	Function	Name
0	PID 38h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	MOD_2_PID_7_DIAG
1	PID 37h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	
2	PID 36h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	
3	PID 35h : Bank 1 Sensor 2 (Wide range O2S)	
4	PID 34h : Bank 1 Sensor 1 (Wide range O2S)	
5	PID 33h : Barometric Pressure	
6	PID 32h : Evap System Vapor Pressure	
7	PID 31h : Distance since diagnostic trouble codes cleared	

DATA D - Supported PIDs 39h to 40h

PID 20h - DATA D - Supported PID 39h to 40h

BIT	Function	Name
0	PID 40h : PIDs supported ($41 - $60)	MOD_2_PID_8_DIAG
1	PID 3Fh : Catalyst Temperature Bank 2, Sensor 2	
2	PID 3Eh : Catalyst Temperature Bank 1, Sensor 2	
3	PID 3Dh : Catalyst Temperature Bank 2, Sensor 1	
4	PID 3Ch : Catalyst Temperature Bank 1, Sensor 1	
5	PID 3Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	
6	PID 3Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	
7	PID 39h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	

54.11.2.1 Description of PID 23h – Fuel pressure

```
If      NC_INJ_CONF <> 0                        //only supported for MSD
Then    DATA_A = ENVD_OBD_22_IDX                // FRF of OBD_FUP_RNG_H
        DATA_B = ENVD_OBD_23_IDX                // FRF of OBD_FUP_RNG_H
Endif
```

54.11.2.2 Description of PID 2Eh – Commanded Evaporative purge

```
DATA_A = ENVD_OBD_24_IDX          // FRF of  CPPWM_CPS
```

54.11.2.3 Description of PID 2Fh - Fuel Level Input

```
DATA_A = ENVD_OBD_25_IDX          // FRF of OBD_FTL
```

54.11.2.4 Description of PID 33h - Barometric Pressure

```
DATA_A = ENVD_OBD_26_IDX          // FRF of OBD_AMP
```

54.11.3 Description of PID 40h - Supported PIDs 41h - 60h

0 : the PID is not used
1 : the PID is used

DATA A - Supported PIDs 41h - 48h

PID 40h - DATA A - Supported PID 41h to 48h		
BIT	**Function**	**Name**
0	PID 48h : Absolute Throttle Position C	MOD_2_PID_9_DIAG
1	PID 47h : Absolute Throttle Position B	
2	PID 46h : Ambient air temperature	
3	PID 45h : Relative Throttle Position	
4	PID 44h : Command Equivalence Ratio	
5	PID 43h : Absolue Load Value	
6	PID 42h : Control module voltage	
7	PID 41h : Monitor status this driving cycle	

DATA B - Supported PIDs 49h to 50h

PID 40h - DATA B - Supported PID 49h to 50h		
BIT	**Function**	**Name**
0	PID 50h : Reserved	MOD_2_PID_A_DIAG
1	PID 4Fh : Reserved	
2	PID 4Eh : Time since diagnostic trouble codes cleared	
3	PID 4Dh : Minutes run by the engine while MIL activated	
4	PID 4Ch : Commanded Throttle Actuator Control	
5	PID 4Bh : Accelerator Pedal Position F	
6	PID 4Ah : Accelerator Pedal Position E	
7	PID 49h : Accelerator Pedal Position D	

DATA C - Supported PIDs 51h to 58h

PID 40h - DATA C - Supported PID 51h to 58h		
BIT	**Function**	**Name**
0	PID 58h : Reserved	MOD_2_PID_B_DIAG
1	PID 57h : Reserved	
2	PID 56h : Reserved	
3	PID 55h : Reserved	
4	PID 54h : Reserved	
5	PID 53h : Reserved	
6	PID 52h : Reserved	
7	PID 51h : Reserved	

DATA D - Supported PIDs 59h to 60h

PID 40h - DATA D - Supported PID 59h to 60h		
BIT	**Function**	**Name**
0	PID 60h : Reserved	MOD_2_PID_C_DIAG
1	PID 5Fh : Reserved	
2	PID 5Eh : Reserved	
3	PID 5Dh : Reserved	
4	PID 5Ch : Reserved	
5	PID 5Bh : Reserved	
6	PID 5Ah : Reserved	
7	PID 59h : Reserved	

54.11.3.1 Description of PID 42 - Control module voltage

DATA_A = ENVD_OBD_27_IDX *// FRF of OBD_VB (high byte)*

DATA_B = ENVD_OBD_28_IDX *// FRF of OBD_VB (low byte)*

54.11.3.2 Description of PID 43 - Absolute Load Value

DATA_A = ENVD_OBD_29_IDX *// FRF of LOAD_ABSV (high byte)*

DATA_B = ENVD_OBD_30_IDX *// FRF of LOAD_ABSV (low byte)*

54.11.3.3 Description of PID 44 - Commanded Equivalence Ratio

DATA_A = ENVD_OBD_31_IDX *// FRF of OBD_LAMP_SP (high byte)*

DATA_B = ENVD_OBD_32_IDX *// FRF of OBD_LAMP_SP (low byte)*

54.11.3.4 Description of PID 45 - Relative Throttle Position

DATA_A = ENVD_OBD_33_IDX *// FRF of OBD_TPS_REL*

54.11.3.5 Description of PID 46 - Ambient air temperature

DATA_A = ENVD_OBD_34_IDX *// FRF of OBD_TAM*

54.11.3.6 Description of PID 47 - Absolute Throttle Position B

DATA_A = ENVD_OBD_35_IDX *// FRF of OBD_TPS_2*

54.11.3.7 Description of PID 49 - Accelerator Pedal Position D

DATA_A = ENVD_OBD_36_IDX *// FRF of OBD_PV_1*

54.11.3.8 Description of PID 4A - Accelerator Pedal Position E

DATA_A = ENVD_OBD_37_IDX *// FRF of OBD_PV_2*

54.11.3.9 Description of PID 4C – Commanded throttle actuator control

DATA_A = ENVD_OBD_38_IDX *// FRF of OBD_TPS_SP*

54.12 PID description for Mode 05

Input data:

MOD_5_TID_1_DIAG	MOD_5_TID_2_DIAG	MOD_5_TID_3_DIAG	MOD_5_TID_4_DIAG
MOD_5_TID_5_DIAG	MOD_5_TID_6_DIAG	MOD_5_TID_7_DIAG	MOD_5_TID_8_DIAG
MOD_5_TID_9_DIAG	MOD_5_TID_10_DIAG	MOD_5_TID_11_DIAG	MOD_5_TID_12_DIAG
MOD_5_TID_13_DIAG	MOD_5_TID_14_DIAG	MOD_5_TID_15_DIAG	MOD_5_TID_16_DIAG
MOD_5_TID_17_DIAG	MOD_5_TID_18_DIAG	MOD_5_TID_19_DIAG	MOD_5_TID_20_DIAG
C_VLS_DOWN_CYC_MIN_SAE	C_VLS_DOWN_CYC_MAX_SAE	PSN_LS	C_LAMB_BAS_COR_MIN
VLS_DOWN_MAX_DC[NC_CBK_EX_NR]	C_LAMB_DELTA_I_MAX_DIAG	LAMB_LS_UP[NC_CBK_EX_NR]	C_LAMB_DELTA_I_MIN_DIAG
FAC_DYN_LSL_DIAG_TOL_SAE[NC_CBK_EX_NR]	FAC_DYN_LSL_DIAG_SAE[NC_CBK_EX_NR]	LAMB_DELTA_I_SAVE_DIAG[NC_CBK_EX_NR]	VLS_DOWN_MIN_DC[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

Read on-board oxygen sensor monitoring test results by an external tester. The use of this mode is optional, depending on the method used by the vehicle manufacturer to comply with the requirement for oxygen sensor monitoring.
Mode05 is only supported if lambda sensors are mounted (PSN_LS = 11H or 33H, see below).

PSN_LS:

BIT	Function	Status
0	Bank 1 Sensor 1	1= available, 0 = not available
1	Bank 1 Sensor 2	1= available, 0 = not available
2	Bank 1 Sensor 3	1= available, 0 = not available
3	Bank 1 Sensor 4	1= available, 0 = not available
4	Bank 2 Sensor 1	1= available, 0 = not available
5	Bank 2 Sensor 2	1= available, 0 = not available
6	Bank 2 Sensor 3	1= available, 0 = not available
7	Bank 2 Sensor 4	1= available, 0 = not available

Formula section:

```
If      PSN_LS = 33h            //means, that
              Bank1 / Sensor 1 and
              Bank2 / Sensor 1 and
              Bank1 / Sensor 2 and
              Bank2 / Sensor 2            are available.
Then    all defined TID are supported
ElseIf  PSN_LS = 11h //means, that Bank1 / Sensor1 and Bank2 Sensor1 are available.
Then    only TID $81, $82, $83, $83 are supported
Else    Mode 5 is not supported
Endif
```

54.12.1 Supported Test ID of MODE $05

54.12.1.1 MODE $05, Test ID $00, DATA A — Supported TIDs $01 ... $08

MODE $05 — Request Oxygen Sensor Monitoring Test Results				
TID $00 — DATA A — Supported TIDs $01 ... $08				
BIT	Function	Supported if	output available for	
0	TID $08: Maximum sensor voltage for test cycle (calculated)	PSN_LS = 33H and MOD_5_TID_1_DIAG (bit 0)	upstream	No
			downstream	**Yes**
1	TID $07: Minimum sensor voltage for test cycle (calculated)	PSN_LS = 33H and MOD_5_TID_1_DIAG (bit 1)	upstream	No
			downstream	**Yes**
2	TID $06: Lean to rich sensor switch time (calculated)	Not supported	upstream	No
3	TID $05: Rich to lean sensor switch time (calculated)	Not supported	upstream	No
4	TID $04: High sensor voltage for switch time calculation (constant)	Not supported	upstream	No
5	TID $03: Low sensor voltage for switch time calculation (constant)	Not supported	upstream	No
6	TID $02: Lean to rich sensor threshold voltage (constant)	PSN_LS = 33H and MOD_5_TID_1_DIAG (bit 6)	upstream	No
			downstream	**Yes**
7	TID $01: Rich to lean sensor threshold voltage (constant) ,	PSN_LS = 33H and MOD_5_TID_1_DIAG (bit 7)	upstream	No
			downstream	**Yes**

54.12.1.2 MODE $05, Test ID $00, DATA B / C / D — Supported TIDs $09 ... $20 (Reserved)

54.12.1.3 MODE $05, Test ID $20, DATA A / B / C / D — Supported TIDs $21 ... $40 (Reserved)

54.12.1.4 MODE $05, Test ID $40, DATA A / B / C / D — Supported TIDs $41 ... $60 (Reserved)

54.12.1.5 MODE $05, Test ID $60, DATA A / B / C / D — Supported TIDs $61 ... $80 (Reserved)

54.12.1.6 MODE $05, Test ID $80, DATA A — Supported TIDs $81 ... $88

MODE $05 — Request Oxygen Sensor Monitoring Test Results				
TID $80 — DATA A — Supported TIDs $81 ... $88				
BIT	Function	Supported if	output available	
0	TID $88: reserved			
1	TID $87: reserved			
2	TID $86: reserved			
3	TID $85: reserved			
4	TID $84: fuel trim diagnosis	PSN_LS = 33H or 11H and MOD_5_TID_17_DIAG (bit 3)	upstream	**Yes**
5	TID $83: sensor dynamics diagnosis	PSN_LS = 33H or 11H and MOD_5_TID_17_DIAG (bit 2)	upstream	**Yes**
6	TID $82: actual a/f setpoint	PSN_LS = 33H or 11H and MOD_5_TID_17_DIAG (bit 1)	upstream	**Yes**
7	TID $81: actual a/f-ratio linear	PSN_LS = 33H or 11H and MOD_5_TID_17_DIAG (bit 0)	upstream	**Yes**

54.12.1.7 MODE $05, Test ID $80, DATA B — Supported TIDs $89 ... $A0 (Reserved)

54.12.2 PID definition for MODE 05

54.12.2.1 MODE $05, Test ID $01 Downstream (B1S2 and B2S2) , Rich to lean threshold voltage

MODE $05 — Request Oxygen Sensor Monitoring Test Results	
TID $01 — Rich to lean sensor threshold voltage downstream B1S2 B2S2 (constant)	
	Project System Variables
test value	C_VLS_DOWN_CYC_MIN_SAE
min. value	**not required**
max. value	**not required**

TID $01 — DATA A – Test value			
	SAE 1979		Project definitions
min. value	0[V]	min. value	0[V] (00H)
max. value	1.275[V]	max. value	1.275[V] (FFH)
scaling	0.005[V] (1 V @ C8h)	scaling	0.005[V]
display		test value	C_VLS_DOWN_CYC_MIN_SAE

54.12.2.2 MODE $05, Test ID $02 Downstream (B1S2 and B2S2) , Lean to rich threshold voltage

MODE $05 — Request Oxygen Sensor Monitoring Test Results	
TID $02 — Lean to rich sensor threshold voltage downstream B1S2 B2S2 (constant)	
	Project System Variables
test value	C_VLS_DOWN_CYC_MAX_SAE
min. value	**not required**
max. value	**not required**

TID $02 — DATA A – Test value			
	SAE 1979		Project definitions
min. value	0[V]	min. value	0[V] (00H)
max. value	1.275[V]	max. value	1.275[V] (FFH)
scaling	0.005[V] (1 V @ C8h)	scaling	0.005[V]
display		test value	C_VLS_DOWN_CYC_MAX_SAE

54.12.2.3 MODE $05, Test ID $07 Downstream (B1S2 and B2S2), Minimum voltage for test cycle

MODE $05 — Request Oxygen Sensor Monitoring Test Results	
TID $07 — Minimum sensor voltage for test cycle downstream B1S2 B2S2	
	Project System Variables
test value	VLS_DOWN_MIN_DC[NC_CBK_EX_NR]
min. value	0[V] (00H)
max. value	1.275[V] (FFH)

TID $07 — DATA A – Test value			
	SAE 1979		Project definitions
min. value	0[V] (00H)	min. value	0[V] (00H)
max. value	1.275[V] (FFH)	max. value	5[V] (3FFH)
scaling	0.005[V]	scaling	0.005[V]
display		test value	VLS_DOWN_MIN_DC[NC_CBK_EX_NR]
TID $07 — DATA B - Minimum test limit			
	SAE 1979		Project definitions
min. value	0[V] (00H)	*min. value*	
max. value	1.275[V] (FFH)	*max. value*	
scaling	0.005[V]	*scaling*	
display		*test value*	0[V] (0H)
TID $07 — DATA C - Maximum test limit			
	SAE 1979		Project definitions
min. value	0[V] (00H)	*min. value*	
max. value	1.275[V] (FFH)	*max. value*	
scaling	0.005[V]	*scaling*	
display		*test value*	1.275[V] (FFH)

54.12.2.4 MODE $05, Test ID $08 Downstream (B1S2 and B2S2), Maximum voltage for test cycle

MODE $05 — Request Oxygen Sensor Monitoring Test Results	
TID $08 — Maximum sensor voltage for test cycle downstream B1S2 B2S2	
	Project System Variables
test value	VLS_DOWN_MAX_DC[NC_CBK_EX_NR]
min. value	0[V] (00H)
max. value	1.275[V] (FFH)

TID $08 — DATA A – Test value			
	SAE 1979		Project definitions
min. value	0[V] (00H)	*min. value*	0[V] (00H)
max. value	1.275[V] (FFH)	*max. value*	5[V] (3FFH)
scaling	0.005[V]	*scaling*	0.005[V]
display		*test value*	VLS_DOWN_MAX_DC[NC_CBK_EX_NR]
TID $08 — DATA B - Minimum test limit			
	SAE 1979		Project definitions
min. value	0[V] (00H)	*min. value*	
max. value	1.275[V] (FFH)	*max. value*	
scaling	0.005[V]	*scaling*	
display		*test value*	0[V] (0H)
TID $08 — DATA C - Maximum test limit			
	SAE 1979		Project definitions
min. value	0[V] (00H)	*min. value*	
max. value	1.275[V] (FFH)	*max. value*	
scaling	0.005[V]	*scaling*	
display		*test value*	1.275[V] (FFH)

54.12.2.5 MODE $05, Test ID $81 Upstream (B1S1 and B2S1) , Actual A/F-ratio

MODE $05 — Request Oxygen Sensor Monitoring Test Results	
TID $81 — Actual A/F-ratio WRAF sensor upstream B1S1 B2S1 (calculated)	
	Project System Variables
test value	LAMB_LS_UP[NC_CBK_EX_NR]
min. value	C_LAMB_BAS_COR_MIN
max. value	2 (FFH)

TID $81 — DATA A – Test value			
	SAE 1979		Project definitions
min. value	0 (00H)	min. value	0s (00H)
max. value	2 (FFH)	max. value	32 (7FFFH)
scaling	0.00781	scaling	0.98e-3
display		test value	LAMB_LS_UP[NC_CBK_EX_NR]
TID $81 — DATA B - Minimum test limit			
	SAE 1979		Project definitions
min. value	0 (00H)	min. value	0s (00H)
max. value	2 (FFH)	max. value	32 (7FFFH)
scaling	0.00781	scaling	0.98e-3
display		min. limit	C_LAMB_BAS_COR_MIN
TID $81 — DATA C - Maximum test limit			
	SAE 1979		Project definitions
min. value	0 (00H)	min. value	
max. value	2 (FFH)	max. value	
scaling	0.00781	scaling	
display		max. limit	2 (FFH)

54.12.2.6 MODE $05, Test ID $82 Upstream (B1S1 and B2S1), Actual A/F-setpoint

MODE $05 — Request Oxygen Sensor Monitoring Test Results	
TID $82 — Actual A/F-setpoint WRAF sensor upstream B1S1 B2S1 (calculated)	
	Project System Variables
test value	LAMB_SP[NC_CBK_EX_NR]
min. value	C_LAMB_BAS_COR_MIN
max. value	2 (FFH)

TID $82 — DATA A – Test value			
	SAE 1979		Project definitions
min. value	0 (00H)	*min. value*	0s (00H)
max. value	2 (FFH)	*max. value*	32 (7FFFH)
scaling	0.00781	*scaling*	0.98e-3
display		*test value*	LAMB_SP[NC_CBK_EX_NR]
TID $82 — DATA B - Minimum test limit			
	SAE 1979		Project definitions
min. value	0 (00H)	*min. value*	0s (00H)
max. value	2 (FFH)	*max. value*	32 (7FFFH)
scaling	0.00781	*scaling*	0.98e-3
display		*min. limit*	C_LAMB_BAS_COR_MIN
TID $82 — DATA C - Maximum test limit			
	SAE 1979		Project definitions
min. value	0 (00H)	*min. value*	
max. value	2 (FFH)	*max. value*	
scaling	0.00781	*scaling*	
display		*max. limit*	2 (FFH)

54.12.2.7 MODE $05, Test ID $83 Upstream (B1S1 and B2S1), Dynamic diagnosis

MODE $05 — Request Oxygen Sensor Monitoring Test Results	
TID $83 — sensor dynamic upstream B1S1 B2S1 (calculated)	
	Project System Variables
test value	FAC_DYN_LSL_DIAG_SAE[NC_CBK_EX_NR]
min. value	0 (00H)
max. value	FAC_DYN_LSL_DIAG_TOL_SAE[NC_CBK_EX_NR]

TID $83 — DATA A – Test value			
	SAE 1979		Project definitions
min. value	0 (00H)	min. value	-327.68 (8000H)
max. value	3.99998 (FFH)	max. value	327.67 (7FFFH)
scaling	0.0156	scaling	0.01 [-]
display		test value	FAC_DYN_LSL_DIAG_SAE[i]
TID $83 — DATA B — Minimum test limit			
	SAE 1979		Project definitions
min. value	0 (00H)	min. value	
max. value	3.99998 (FFH)	max. value	
scaling	0.0156	scaling	
display		min. limit	0 (00H)
TID $83 — DATA C — Maximum test limit			
	SAE 1979		Project definitions
min. value	0 (00H)	min. value	
max. value	3.99998 (FFH)	max. value	327.67 (7FFFH)
scaling	0.0156	scaling	0.01 [-]
display		max. limit	FAC_DYN_LSL_DIAG_TOL_SAE[i]

54.12.2.8 MODE $05, Test ID $84 Upstream (B1S1 and B2S1), Fuel trim sensor diagnosis

MODE $05 — Request Oxygen Sensor Monitoring Test Results	
TID $84 — fuel trim sensor diagnosis upstream B1S1 B2S1 (calculated)	
	Project System Variables
test value	LAMB_DELTA_I_SAVE_DIAG[NC_CBK_EX_NR]
min. value	C_LAMB_DELTA_I_MIN_DIAG
max. value	C_LAMB_DELTA_I_MAX_DIAG

TID $84 — DATA A - Test value			
	manuf. def.		Project definitions
min. value	-0.125 (00H)	min. value	-0.125 (F800H)
max. value	+0.125 (FFH)	max. value	+0.125 (800H)
scaling	9.80392e-4 (0.25/256)	scaling	6,1035E-5 [-]
display		test value	LAMB_DELTA_I_SAVE_DIAG[i]
TID $84 — DATA B Minimum test limit			
	manuf. def.		Project definitions
min. value	-0.125 (00H)	min. value	-0.125 (F800H)
max. value	+0.125 (FFH)	max. value	+0.125 (800H)
scaling	9.80392e-4 (0.25/256)	scaling	6,1035E-5 [-]
display		min. limit	C_LAMB_DELTA_I_MIN_DIAG
TID $84 — DATA C — Maximum test limit			
	manuf. def.		Project definitions
min. value	-0.125 (00H)	min. value	-0.125 (F800H)
max. value	+0.125 (FFH)	max. value	+0.125 (800H)
scaling	9.80392e-4 (0.25/256)	scaling	6,1035E-5 [-]
display		*max. limit*	C_LAMB_DELTA_I_MAX_DIAG

54.13 TID description for mode 06h – ISO 14230

Input data:

MAF_DIAGCPS_SAE	MOD_6_TID_2_DIAG	MOD_6_TID_3_DIAG	MOD_6_TID_4_DIAG
CAT_DIAG_MOD_6[NC_CBK_EX_NR]	CAT_MAX_DIAG_MOD_6[NC_CBK_EX_NR]	LV_READY_XX	LV_ERR_MEM_XX
PSN_LS	FAC_DYN_LSL_DIAG_TOL_SAE[NC_CBK_EX_NR]	FAC_DYN_LSL_DIAG_SAE[NC_CBK_EX_NR]	CTR_SAVE_SWT_LS_DOWN[NC_CBK_EX_NR]
C_CTR_SWT_LS_DOWN	C_TTIP_MAX_OBD_LSH_UP	C_TTIP_MIN_OBD_LSH_UP	LC_SA_SWI_ACQ
C_SUM_AFL_VLS_MIN_DIAG_SA_1	SAF_DIAG_MIN_SAE	C_SAF_DIAG_MIN	SAF_DIAG_MAX_SAE
C_SAF_DIAG_MAX	SUM_AFL_VLS_DIAG_SA_1_SAE	SUM_AFL_VLS_DIAG_SA_2_SAE	LV_CONF_DMTL
MAF_DIAGCPS_THD_SAE	MAF_DIAGCPS_SAE	CL_MMV_SAE	C_CL_MMV_DIAGCPS
SUM_DIAG_DIAGCPS_SAE	C_SUM_DIAGCPS_MAX	M6_CUR_DMTL_REF_LEAK	M6_CUR_DMTL_SMALL_LEAK_END
M6_CUR_DMTL_COR_FIL_CID19	C_CUR_DMTL_REF_LEAK_MAX	M6_CUR_DMTL_DMTLS_TEST	M6_CUR_DMTL_THD_DMTLS_TEST
M6_CTR_CNL_SMALL_LEAK_MES	C_SUM_CNL_SMALL_LEAK_MES_MAX	M6_CUR_DMTL_COR_FIL_CID18	C_CUR_DMTL_REF_LEAK_MIN
M6_CUR_DMTL_ROUGH_LEAK_LEN_END	M6_CUR_DMTL_THD_ROUGH_LEAK_LEN	M6_CUR_DMTL_ROUGH_LEAK_END	M6_CUR_DMTL_THD_ROUGH_LEAK
R_IT_THD_OBD_LSH_DOWN[NC_CBK_EX_NR]	R_IT_OBD_LSH_DOWN[NC_CBK_EX_NR]	TTIP_MES_LS_UP_OBD_LSH_UP[NC_CBK_EX_NR]	

54.13.1 Overview of all implemented TID's

FUNCTION DESCRIPTION:

General information:

The CID is calculatated depending if the Threshold is working as a "Maximum" or "Minimum" threshold. The "Minimum" CID is always the "Maximum" CID + **80h**.

For example:

TID 03h / CID 05/85h : Secondary Air Monitor 1 – LinLam Min flow bank 1/2
=> CID 05h is a "Maximum" Threshold
=> CID 85h is a "Minimum" Threshold (85h = 05h + **80h**)

List of all TIDs
Catalyst monitor
TID 01h / CID 01h : Catalyst monitor Bank1
TID 01h / CID 02h : Catalyst monitor Bank2
Oxygen sensor monitor
TID 02h / CID 15h : Oxygen sensor monitor Bank1 – Sensor1 – Dynamic
TID 02h / CID 30h : Oxygen sensor monitor Bank1 – Sensor2 – Switchtime
TID 02h / CID 18 : Oxygen sensor monitor Bank2 – Sensor1 – Dynamic
TID 02h / CID 32h : Oxygen sensor monitor Bank2 – Sensor2 – Switchtime
Secondary air monitor
TID 03h / CID 20h : Secondary Air Monitor 2 – SA – meter Min flow, Step 1
TID 03h / CID 21h : Secondary Air Monitor 2 – SA – meter Min flow, Step 2
TID 03h / CID 05h(85h) : Secondary Air Monitor 1 – LinLam Min flow bank 1/2
Evaporative monitor
TID 05h / CID 07h(87h) : EVAP monitor – CPS check – Step 1
TID 05h / CID 08h : EVAP monitor – CPS check – Step 2
TID 05h / CID 09h : EVAP monitor – CPS check – Step 3
TID 05h / CID 13h : EVAP monitor – Module error – Max error
TID 05h / CID 14h : EVAP monitor – Module error – Plausibility error
TID 05h / CID 17h : EVAP monitor – Module error – Signal error
TID 05h / CID 12h(92h) : EVAP monitor – Module error – Min error
TID 05h / CID 15h(95h) : EVAP monitor – DMTL – Rough leak – Short cycle
TID 05h / CID 16h(96h) : EVAP monitor – DMTL – Small leak
TID 05h / CID 18h(98h) : EVAP monitor – DMTL – Rough leak – Long cycle
Oxygen heater monitor
TID 06h / CID 03h : Oxygen Sensor Heater Monitor Bank 1 – Sensor 2 - R
TID 06h / CID 04h : Oxygen Sensor Heater Monitor Bank 2 – Sensor 2 - R
TID 06h / CID 05h/85h:Oxygen Sensor Heater Monitor Bank 1–Sensor 1 - TTIP
TID 06h / CID 06h/86h:Oxygen Sensor Heater Monitor Bank 2–Sensor 1 - TTIP

54.13.2 Supported TIDs 01h – C0h

54.13.2.1 Description of TID 00h - Supported TIDs 01h - 20h

TID 00h - DATA A - Supported TID 01h to 08h		
BIT	**Function**	**Name**
0	TID 08h : Reserved	MOD_6_TID_1_DIAG
1	TID 07h : Reserved	
2	TID 06h : Oxygen sensor heater monitor	
3	TID 05h : Evaporative monitor	
4	TID 04h : Reserved	
5	TID 03h : Secondary air monitor	
6	TID 02h : Oxygen sensor monitor	
7	TID 01h : Catalyst monitor	
TID 00h - DATA B - Supported TID 09h to 10h		
BIT	**Function**	**Name**
0	TID 10h : reserved	MOD_6_TID_2_DIAG
1	TID 0Fh : reserved	
2	TID 0Eh : reserved	
3	TID 0Dh : reserved	
4	TID 0Ch : reserved	
5	TID 0Bh : reserved	
6	TID 0Ah : reserved	
7	TID 09h : reserved	
TID 00h - DATA C - Supported TID 11h to 18h		
BIT	**Function**	**Name**
0	TID 18h : reserved	MOD_6_TID_3_DIAG
1	TID 17h : reserved	
2	TID 16h : reserved	
3	TID 15h : reserved	
4	TID 14h : reserved	
5	TID 13h : reserved	
6	TID 12h : reserved	
7	TID 11h : reserved	
TID 00h - DATA D - Supported TID 19h to 20h		
BIT	**Function**	**Name**
0	TID 20h : TIDs supported ($21 - $40)	MOD_6_TID_4_DIAG
1	TID 1Fh : reserved	
2	TID 1Eh : reserved	
3	TID 1Dh : reserved	
4	TID 1Ch : reserved	
5	TID 1Bh : reserved	
6	TID 1Ah : reserved	
7	TID 19h : reserved	

54.13.3 Description of TID 01 - Catalyst monitoring

General information:

TID 01 is only supported if the vehicle is equipped with lambda sensors UP and DOWN

Formula section:

```
If      PSN_LS = 33H
Then    TID 01 is supported
Else    TID 01 is not supported
Endif
```

54.13.3.1 CID 01h - Bank 1

OSC method (linear system)	
Std/Manufacturer defined **CID**	01h
Unit and scaling	NONE
Test value	CAT_DIAG_MOD_6[1]
Min test limit	NONE
Max test limit	CAT_MAX_DIAG_MOD_6[1]

Formula section:

```
IF      LV_READY_CAT_DIAG[1] = 0          OR
        LV_ERR_MEM_CAT_DIAG[1] = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.3.2 CID 02h - Bank 2

OSC method (linear system)	
Std/Manufacturer defined **CID**	02h
Unit and scaling	NONE
Test value	CAT_DIAG_MOD_6[2]
Min test limit	NONE
Max test limit	CAT_MAX_DIAG_MOD_6[2]

Formula section:

```
IF      LV_READY_CAT_DIAG[2] = 0          OR
        LV_ERR_MEM_CAT_DIAG[2] = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.4 Description of TID 02h - Lambda sensor diagnosis

General information:

TID 02 is only supported if the vehicle is equipped with lambda sensors UP and DOWN or UP.

Formula section:

```
If      PSN_LS = 33H or 11H
Then    TID 02 is supported
    If      PSN_LS = 11H
    Then    CID 30h / 32h are not supported
            //no Downstream sensors
    Endif
Else    TID 02 is not supported
Endif
```

54.13.4.1 CID 15h – Dynamic diagnosis (linear), Bank 1

Sensor dynamic diagnosis	
Std/Manufacturer defined **CID**	15h (MAX)
Unit and scaling	NONE
Test value	FAC_DYN_LSL_DIAG_SAE[1]
Min test limit	0
Max test limit	FAC_DYN_LSL_DIAG_TOL_SAE[1]

Formula section:

```
IF      LV_READY_DYN_VLD_LS_UP[1] = 0      OR
        LV_ERR_MEM_DYN_VLD_LS_UP[1] = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.4.2 CID 18h - Dynamic diagnosis (linear), Bank 2

Sensor dynamic diagnosis	
Std/Manufacturer defined **CID**	18h (MAX)
Unit and scaling	NONE
Test value	FAC_DYN_LSL_DIAG_SAE[2]
Min test limit	0
Max test limit	FAC_DYN_LSL_DIAG_TOL_SAE[2]

Formula section:

```
IF      LV_READY_DYN_VLD_LS_UP[2] = 0        OR
        LV_ERR_MEM_DYN_VLD_LS_UP[2] = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.4.3 CID 30h - Switchtime diagnosis (downstream binary), Bank 1

Switchtime diagnosis	
Std/Manufacturer defined **CID**	30h
Unit and scaling	NONE
Test value	CTR_SAVE_SWT_LS_DOWN[1]
Min test limit	NONE
Max test limit	C_CTR_SWT_LS_DOWN

Formula section:

```
IF      LV_READY_SWT_LS_DOWN[1] = 0        OR
        LV_ERR_MEM_SWT_LS_DOWN[1] = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.4.4 CID 32h - Switchtime diagnosis (downstream binary), Bank 2

Switchtime diagnosis	
Std/Manufacturer defined CID	32h
Unit and scaling	NONE
Test value	CTR_SAVE_SWT_LS_DOWN[2]
Min test limit	NONE
Max test limit	C_CTR_SWT_LS_DOWN

Formula section:

```
IF      LV_READY_SWT_LS_DOWN[2] = 0        OR
        LV_ERR_MEM_SWT_LS_DOWN[2] = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.5 Description of TID 03h - Secondary air monitoring

General information:

TID 03h is only supported if SA pump is learnt. Depending of hardware configuration in the car only one type of SA monitioring is active (via lambda-sensor **or** flow meter).

Formula section:

```
If      LV_VAR_SAP = 1
Then    TID 03 is supported
    If      LC_SA_SWI_ACQ = 1
    Then    CID 05h(=85h) are not supported        //no diagnosis via lambda sensor
    Else    CID 20h and 21h are not supported      //no diagnosis via SAF meter
    Endif
Else    TID 03 is not supported
Endif
```

54.13.5.1 CID 20h/21h - Secondary air monitor with flow meter (SAFM)

Monitoring of minimum flow rate	
Std/Manufacturer defined **CID**	20h
Unit and scaling	NONE
Test value	SAF_DIAG_MIN_SAE
Min test limit	NONE
Max test limit	C_SAF_DIAG_MIN
Monitoring of minimum flow rate	
Std/Manufacturer defined **CID**	21h
Unit and scaling	NONE
Test value	SAF_DIAG_MAX_SAE
Min test limit	NONE
Max test limit	C_SAF_DIAG_MAX

Formula section:

```
IF      LV_READY_SA_SYS = 0          OR
        LV_ERR_MEM_SA_SYS = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
Endif
```

54.13.5.2 CID 85h - Secondary air monitor with lambda sensor (LISL)

Monitoring of minimum flow rate, Bank 1 / Bank 2	
Std/Manufacturer defined **CID**	85h (identifier 05h + 80h)
Unit and scaling	NONE
Test value	**MIN [** SUM_AFL_VLS_DIAG_SA_1_SAE; SUM_AFL_VLS_DIAG_SA_2_SAE **]**
Min test limit	C_SUM_AFL_VLS_MIN_DIAG_SA_1
Max test limit	NONE

Formula section:

```
IF      LV_READY_SA_SAP = 0        OR
        LV_ERR_MEM_SA_SAP = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.6 Description of TID 05h - Evaporative Monitor

General information:

All DMTL related CID's are only supported if the CAR is equipped with the DMTL module.

Formula section:

If LV_CONF_DMTL = 0

Then CID's 13h/14h/17h/ 12h(=92h)/15h(=95h)/16h(=96h)/18h(=98h)

... are not supported

Else all CID's of TID 05h are supported

Endif

54.13.6.1 CID 09h/87h/88h - Purge Flow monitor (TEV check)

Functional check TEV – Step 1	
Std/Manufacturer defined **CID**	87h (identifier 07h + 80h)
Unit and scaling	NONE
Test value	CL_MMV_SAE
Min test limit	C_CL_MMV_DIAGCPS
Max test limit	NONE
Functional check TEV – Step 2	
Std/Manufacturer defined **CID**	08h
Unit and scaling	NONE
Test value	SUM_DIAG_DIAGCPS_SAE
Min test limit	NONE
Max test limit	C_SUM_DIAGCPS_MAX
Functional check TEV – Step 3	
Std/Manufacturer defined **CID**	09h
Unit and scaling	NONE
Test value	MAF_DIAGCPS_SAE
Min test limit	NONE
Max test limit	MAF_DIAGCPS_THD_SAE

Formula section:

IF LV_READY_DIAGCPS = 0 **OR**

LV_ERR_MEM_DIAGCPS = 1

THEN Positive response with test result

ELSE Positive response with all data equal to 0000h

ENDIF

54.13.6.2 CID 13h/14h/17h/92h - DMTL module diagnosis

DMTL module – MAX error	
Std/Manufacturer defined **CID**	13h
Unit and scaling	NONE
Test value	M6_CUR_DMTL_COR_FIL_CID19
Min test limit	NONE
Max test limit	C_CUR_DMTL_REF_LEAK_MAX
DMTL module – Plausibility error	
Std/Manufacturer defined **CID**	14h
Unit and scaling	NONE
Test value	M6_CUR_DMTL_DMTLS_TEST
Min test limit	NONE
Max test limit	M6_CUR_DMTL_THD_DMTLS_TEST
DMTL module - Signal error	
Std/Manufacturer defined **CID**	17h
Unit and scaling	NONE
Test value	M6_CTR_CNL_SMALL_LEAK_MES
Min test limit	NONE
Max test limit	C_SUM_CNL_SMALL_LEAK_MES_MAX
DMTL module – MIN error	
Std/Manufacturer defined **CID**	92h (identifier 12h + 80h)
Unit and scaling	NONE
Test value	M6_CUR_DMTL_COR_FIL_CID18
Min test limit	C_CUR_DMTL_REF_LEAK_MIN
Max test limit	NONE

Formula section:

```
IF      LV_READY_DMTLM = 0        OR
        LV_ERR_MEM_DMTLM = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.6.3 CID 95h/98h - Rough-leak diagnosis

Leak detection DMTL – Long cycle	
Std/Manufacturer defined **CID**	98h (identifier 18h + 80h)
Unit and scaling	NONE
Test value	M6_CUR_DMTL_ROUGH_LEAK_LEN_END
Min test limit	M6_CUR_DMTL_THD_ROUGH_LEAK_LEN
Max test limit	NONE
Leak detection DMTL – Short cycle	
Std/Manufacturer defined **CID**	95h (identifier 15h + 80h)

Unit and scaling	NONE
Test value	M6_CUR_DMTL_ROUGH_LEAK_END
Min test limit	M6_CUR_DMTL_THD_ROUGH_LEAK
Max test limit	NONE

Formula section:

```
IF      LV_READY_ROUGH_LEAK = 0        OR
        LV_ERR_MEM_ROUGH_LEAK = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.6.4 CID 96h - Small-leak diagnosis

Leak detection DMTL – Short cycle	
Std/Manufacturer defined **CID**	96h (identifier 16h + 80h)
Unit and scaling	NONE
Test value	M6_CUR_DMTL_SMALL_LEAK_END
Min test limit	M6_CUR_DMTL_REF_LEAK
Max test limit	NONE

Formula section:

```
IF      LV_READY_SMALL_LEAK = 0        OR
        LV_ERR_MEM_SMALL_LEAK = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.7 Description of TID 06h - Oxygen sensor heater monitoring

General information:

TID 06 is only supported if the vehicle is equipped with lambda sensors UP and DOWN or UP.

Formula section:

```
If      PSN_LS = 33H or 11H
Then    TID 06 is supported
    If      PSN_LS = 11H
    Then    CID 03h / 04h are not supported    //no Downstream sensors
    Endif
Else    TID 06 is not supported
Endif
```

54.13.7.1 CID 05h / 85h - Oxygen sensor heater monitor, Bank 1 Sensor 1 (linear)

Sensor temperature	
Std/Manufacturer defined **CID**	05h (MAX) 85h (MIN)
Unit and scaling	NONE
Test value	TTIP_MES_LS_UP_OBD_LSH_UP[1]
Min test limit	C_TTIP_MIN_OBD_LSH_UP
Max test limit	C_TTIP_MAX_OBD_LSH_UP

Formula section:

```
IF      LV_READY_OBD_VLD_LSH_UP[1] = 0        OR
        LV_ERR_MEM_OBD_VLD_LSH_UP[1] = 1
THEN    Positive response with test result
ELSE    Positive response with all data equal to 0000h
ENDIF
```

54.13.7.2 CID 03h - Oxygen sensor heater monitor, Bank 1 Sensor 2 (binary)

Sensor resistance	
Std/Manufacturer defined **CID**	03h
Unit and scaling	NONE
Test value	R_IT_OBD_LSH_DOWN[1]
Min test limit	NONE
Max test limit	R_IT_THD_OBD_LSH_DOWN[1]

Formula section:

```
IF      LV_READY_OBD_LSH_DOWN[1] = 0        OR
```

```
          LV_ERR_MEM_OBD_LSH_DOWN[1] = 1
THEN      Positive response with test result
ELSE      Positive response with all data equal to 0000h
ENDIF
```

54.13.7.3 CID 06h / 86h - Oxygen sensor heater monitor, Bank 2 Sensor 1 (linear)

Sensor temperature	
Std/Manufacturer defined **CID**	06h (MAX) 86h (MIN)
Unit and scaling	NONE
Test value	TTIP_MES_LS_UP_OBD_LSH_UP[2]
Min test limit	C_TTIP_MIN_OBD_LSH_UP
Max test limit	C_TTIP_MAX_OBD_LSH_UP

Formula section:

```
IF        LV_READY_OBD_VLD_LSH_UP[2] = 0          OR
          LV_ERR_MEM_OBD_VLD_LSH_UP[2] = 1
THEN      Positive response with test result
ELSE      Positive response with all data equal to 0000h
ENDIF
```

54.13.7.4 CID 04h - Oxygen sensor heater monitor, Bank 2 Sensor 2 (binary)

Sensor resistance	
Std/Manufacturer defined **CID**	04h
Unit and scaling	NONE
Test value	R_IT_OBD_LSH_DOWN[2]
Min test limit	NONE
Max test limit	R_IT_THD_OBD_LSH_DOWN[2]

Formula section:

```
IF        LV_READY_OBD_LSH_DOWN[2] = 0            OR
          LV_ERR_MEM_OBD_LSH_DOWN[2] = 1
THEN      Positive response with test result
ELSE      Positive response with all data equal to 0000h
ENDIF
```

54.14 VIT description for mode 09h

Input data:

MOD_9_VIT_1_DIAG	MOD_9_VIT_2_DIAG	MOD_9_VIT_3_DIAG	MOD_9_VIT_4_DIAG

Import action:

ACTION_ERRM_SelectRbmData (**INOUT < ListRbmData >,** **OUT < ResultRbmData >** **)**
This API calculates and returns the Rate-Based Monitoring data to be transmitted to the Scan-Tool, when requested by the Mode 09$ (InfoType $08).

54.14.1 Description of Mode 09h, VIT 00h - All supported VITs

VIT 00h - DATA A - Supported VIT 01h to 08h		
BIT	**Function**	**name**
0	VIT 08h : In-Use Performance Tracking	MOD_9_VIT_1_DIAG
1	VIT 07h : Message Count IPT	
2	VIT 06h : Calibration Verification Numbers	
3	VIT 05h : Number of response for Calibration Verification Numbers	
4	VIT 04h : Calibration IDs	
5	VIT 03h : Number of response for Calibration IDs	
6	VIT 02h : Vehicle Identification Number	
7	VIT 01h : Number of response for Vehicle Identification Number	
VIT 00h - DATA B - Supported VIT 09h to 10h		
BIT	**Function**	**name**
0	VIT 10h : reserved	MOD_9_VIT_2_DIAG
1	VIT 0Fh : reserved	
2	VIT 0Eh : reserved	
3	VIT 0Dh : reserved	
4	VIT 0Ch : reserved	
5	VIT 0Bh : reserved	
6	VIT 0Ah : reserved	
7	VIT 09h : reserved	
VIT 00h - DATA C - Supported VIT 11h to 18h		
BIT	**Function**	**name**
0	VIT 18h : reserved	MOD_9_VIT_3_DIAG
1	VIT 17h : reserved	
2	VIT 16h : reserved	
3	VIT 15h : reserved	
4	VIT 14h : reserved	
5	VIT 13h : reserved	
6	VIT 12h : reserved	
7	VIT 11h : reserved	
VIT 00h - DATA D - Supported VIT 19h to 20h		

BIT	Function	name
0	VIT 20h : VITs supported ($21 - $40)	MOD_9_VIT_4_DIAG
1	VIT 1Fh : reserved	
2	VIT 1Eh : reserved	
3	VIT 1Dh : reserved	
4	VIT 1Ch : reserved	
5	VIT 1Bh : reserved	
6	VIT 1Ah : reserved	
7	VIT 19h : reserved	

54.14.2 Description of Mode 09h, VIT 01h – Messagecount VIN

54.14.3 Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: MessageCount VIN	01	INFTYP

Positive responses (read InfoType value)

Request vehicle information response message

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: MessageCount VIN	01	INFTYP
#3	MessageCount VIN = 5 response messages	05	MC_CALID

54.14.4 Description of Mode 09h, VIT 02h - VIN

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: VIN	02	INFTYP

54.14.5 Response positive report VIN

First positive response

— Request vehicle information response message (1)

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: VIN	02	INFTYP
#3	MessageCount VIN = 1st response message	01	MC_CALID
#4	Data A: Pad-byte	00	byte always 0
#5	Data B: Pad-byte	00	byte always 0
#6	Data C: Pad-byte	00	byte always 0
#7	Data D:	XX*	VIN char 0

* 30h, if there is no VIN.

Second positive response

Request vehicle information response message (2)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: VIN	02	INFTYP
#3	MessageCount VIN = 2nd response message	02	MC_CALID
#4	Data A:	XX*	VIN char 1
#5	Data B:	XX*	VIN char 2
#6	Data C:	XX*	VIN char 3
#7	Data D:	XX*	VIN char 4

* 30h, if there is no VIN.

thirth positive response

— Request vehicle information response message (3)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: VIN	02	INFTYP
#3	MessageCount VIN = 3rd response message	03	MC_CALID
#4	Data A:	XX*	VIN char 5
#5	Data B:	XX*	VIN char 6
#6	Data C:	XX*	VIN char 7
#7	Data D:	XX*	VIN char 8

* 30h, if there is no VIN

forth positive response

Request vehicle information response message (4)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: VIN	02	INFTYP
#3	MessageCount VIN = 4nd response message	04	MC_CALID
#4	Data A:	XX*	VIN char 9
#5	Data B:	XX*	VIN char 10
#6	Data C:	XX*	VIN char 11
#7	Data D:	XX*	VIN char 12

* 30h, if there is no VIN

fifth positive response

Request vehicle information response message (5)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: VIN	02	INFTYP
#3	MessageCount VIN = 5nd response message	05	MC_CALID
#4	Data A:	XX*	VIN char 13
#5	Data B:	XX*	VIN char 14
#6	Data C:	XX*	VIN char 15
#7	Data D:	XX*	VIN char 16

* 30h, if there is no VIN

54.14.6 Description of Mode 09h, VIT 03h – Messagecount Calibration IDs

54.14.7 Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: MessageCount Calibration Id	03	INFTYP

Positive responses (read InfoType value)

Request vehicle information response message

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: MessageCount Calibration Id	03	INFTYP
#3	MessageCount Calibration Id = 8 response messages	08	MC_CALID

54.14.8 Description of Mode 09h, VIT 04h - Calibration IDs

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: Calibration Id	04	INFTYP

54.14.9 Response positive report Calibration ID

First positive response

— Request vehicle information response message (1)

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	MessageCount Calibration Id = 1st response message	01	MC_CALID
#4	Data A: SW version number char 1	NC_BMW_HW_NR_3 *or 30	second digit in BCD
#5	Data B: SW version number char 2	NC_BMW_HW_NR_4 *or 30	first digit in BCD
#6	Data C: SW version number char 3	NC_BMW_HW_NR_4 *or 30	second digit in BCD
#7	Data D: SW version number char 4	NC_BMW_HW_NR_5 *or 30	first digit in BCD

* 30h, if there is no HW_NR.

Second positive response

Request vehicle information response message (2)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	MessageCount Calibration Id = 2nd response message	02	MC_CALID
#4	Data A: SW version number char 5	NC_BMW_HW_NR_5 *or 30	second digit in BCD
#5	Data B: SW version number char 6	NC_BMW_HW_NR_6 *or 30	first digit in BCD
#6	Data C: SW version number char 7	NC_BMW_HW_NR_6 *or 30	second digit in BCD
#7	Data D: Pad-Byte	00	byte always 0

* 30h, if there is no HW_NR.

thirth positive response

— Request vehicle information response message (3)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	MessageCount Calibration Id = 3rd response message	03	MC_CALID
#4	Data A: Pad-Byte	00	byte always 0
#5	Data B: Pad-Byte	00	byte always 0
#6	Data C: Pad-Byte	00	byte always 0
#7	Data D: Pad-Byte	00	byte always 0

forth positive response

Request vehicle information response message (4)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	MessageCount Calibration Id = 4th response message	04	MC_CALID
#4	Data A: Pad-Byte	00	byte always 0
#5	Data B: Pad-Byte	00	byte always 0
#6	Data C: Pad-Byte	00	byte always 0
#7	Data D: Pad-Byte	00	byte always 0

fifth positive response

Request vehicle information response message (5)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	MessageCount Calibration Id = 5nd response message	05	MC_CALID
#4	Data A: Calibration version number char 1	CAL_NR_3 or 30*	second digit in BCD
#5	Data B: Calibration version number char 2	CAL_NR_4 or 30 *	first digit in BCD
#6	Data C: Calibration version number char 3	CAL_NR_4 or 30 *	second digit in BCD
#7	Data D: Calibration version number char 4	CAL_NR_5 or 30 *	first digit in BCD

* if no CAL_NR in software;

sixth positive response

Request vehicle information response message (6)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	MessageCount Calibration Id = 6nd response message	06	MC_CALID
#4	Data A: Calibration version number char 5	CAL_NR_5 or 30 *	second digit in BCD
#5	Data B: Calibration version number char 6	CAL_NR_6 or 30 *	first digits in BCD
#6	Data C: Calibration version number char 7	CAL_NR_6 or 30 *	second digit in BCD
#7	Data D: Pad-Byte	00	byte always 0

* if no CAL_NR in software;

seventh positive response

Request vehicle information response message (7)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	MessageCount Calibration Id = 7nd response message	07	MC_CALID
#4	Data A: Pad-Byte	00	byte always 0
#5	Data B: Pad-Byte	00	byte always 0
#6	Data C: Pad-Byte	00	byte always 0
#7	Data D: Pad-Byte	00	byte always 0

eight positive response

Request vehicle information response message (8)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	MessageCount Calibration Id = 8nd response message	08	MC_CALID
#4	Data A: Pad-Byte	00	byte always 0
#5	Data B: Pad-Byte	00	byte always 0
#6	Data C: Pad-Byte	00	byte always 0
#7	Data D: Pad-Byte	00	byte always 0

54.14.10 Description of Mode 09h, VIT 05h – Messagecount Calibration verification numbers

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: MessageCount Calibration Verification Number	05	INFTYP

Positive responses (read InfoType value)

Request vehicle information response message

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: MessageCount Calibration Verification Number	05	INFTYP
#3	MessageCount Calibration Verification Number = 2 response messages	02	MC_CVN

54.14.11 Description of Mode 09h, VIT 06h - Calibration verification numbers

The on-board software will calculate the CVN (CRC) based on memory content in a cyclic way. The result will be stored non volatile. That means the previous value will be available till a new one is calculated.
At first initialisation of ECU no CVN is available till first one is calculated (app. 1min).

Request vehicle information request message (read InfoType values)

Data Byte	Parameter Name	Hex Value	Mnemonic
#1	**Request vehicle information request SID**	**09**	**SIDRQ**
#2	InfoType (read InfoType values)	06	INFTYP

54.14.11.1 Response positive report calibration verification number

<u>First positive response</u>

Request vehicle information response message (1)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Verification Number	06	INFTYP
#3	MessageCount Calibration Verification Number = 1st response message	01	MC_CVN
#4	Data A: MSB CRC-32 Application SW	MSB (CKS_CRC_ECU)	DATA_A
#5	Data B: …	M_H (CKS_CRC_ECU)	DATA_B
#6	Data C: …	M_L (CKS_CRC_ECU)	DATA_C
#7	Data D: LSB CRC-32 Application SW	LSB (CKS_CRC_ECU)	DATA_D

<u>Second positive response</u>

Request vehicle information response message (2)

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Verification Number	06	INFTYP
#3	MessageCount Calibration Verification Number = 2nd response message	02	MC_CVN
#4	Data A: MSB CRC-32 Calibration data	MSB (CKS_CRC_CAL)	DATA_A
#5	Data B: …	M_H (CKS_CRC_CAL)	DATA_B
#6	Data C: …	M_L (CKS_CRC_CAL)	DATA_C
#7	Data D: LSB CRC-32 Calibration data	LSB (CKS_CRC_CAL)	DATA_D

54.14.12 Description of Mode 09h, InfoType 07h - MessageCount IPT

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: MessageCount IPT	07	INFTYP

Positive responses (read InfoType value)

Request vehicle information response message

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: MessageCount IPT	07	INFTYP
#3	Number of messages to report In-use Performance Tracking = 8 response messages	08	MC_IPT

54.14.13 Description of Mode 09h, InfoType 08h

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: In-use Performance Tracking	08	INFTYP

Response positive : In-use Performance Tracking

First positive response

— Request vehicle information response message (1)

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	MessageCount Calibration Id = 1st response message	01	MC_CALID
#4	Data A: MSB (OBD Monitoring Conditions Encountered Counts – general denominator)	XX	OBDCOND_A
#5	Data B: LSB (OBD Monitoring Conditions Encountered Counts Counts – general denominator)	XX	OBDCOND_B
#6	Data C: MSB (Ignition Counter)	XX	IGNCNTR_A
#7	Data D: LSB (Ignition Counter)	XX	IGNCNTR_B

Second positive response

Request vehicle information response message (2)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	MessageCount Calibration Id = 2nd response message	02	MC_CALID
#4	Data A: MSB (Catalyst Monitor Completion Counts Bank 1 – numerator)	XX	CATCOMP1_A
#5	Data B: LSB (Catalyst Monitor Completion Counts Bank 1 – numerator)	XX	CATCOMP1_B
#6	Data C: MSB (Catalyst Monitor Conditions Encountered Counts Bank 1 – denominator)	XX	CATCOND1_A
#7	Data D: LSB (Catalyst Monitor Conditions Encountered Counts Bank 1 – denominator)	XX	CATCOND1_B

Thirth positive response

— Request vehicle information response message (3)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	MessageCount Calibration Id = 3nd response message	03	MC_CALID
#4	Data A: MSB (Catalyst Monitor Completion Counts Bank 2 – numerator)	XX	CATCOMP2_A
#5	Data B: LSB (Catalyst Monitor Completion Counts Bank 2 – numerator)	XX	CATCOMP2_B
#6	Data C: MSB (Catalyst Monitor Conditions Encountered Counts Bank 2 – denominator)	XX	CATCOND2_A
#7	Data D: LSB (Catalyst Monitor Conditions Encountered Counts Bank 2 – denominator)	XX	CATCOND2_B

Fourth positive response

— Request vehicle information response message (1)

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	MessageCount Calibration Id = 4th response message	04	MC_CALID
#4	Data A: MSB (O2 Sensor Monitor Completion Counts Bank 1 – numerator)	XX	O2SCOMP1_A
#5	Data B: LSB (O2 Sensor Monitor Completion Counts Bank 1 – numerator)	XX	O2SCOMP1_B
#6	Data C: MSB (O2 Sensor Monitor Conditions Encountered Counts Bank 1 – denominator)	XX	O2SCOND1_A
#7	Data D: LSB (O2 Sensor Monitor Conditions Encountered Counts Bank 1 – denominator)	XX	O2SCOND1_B

Fifth positive response

Request vehicle information response message (2)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	MessageCount Calibration Id = 5th response message	05	MC_CALID
#4	Data A: MSB (O2 Sensor Monitor Completion Counts Bank 2 – numerator)	XX	O2SCOMP2_A
#5	Data B: LSB (O2 Sensor Monitor Completion Counts Bank 2 – numerator)	XX	O2SCOMP2_B
#6	Data C: MSB (O2 Sensor Monitor Conditions Encountered Counts Bank 2 – denominator)	XX	O2SCOND2_A
#7	Data D: LSB (O2 Sensor Monitor Conditions Encountered Counts Bank 2 – denominator)	XX	O2SCOND2_B

Sixth positive response

— Request vehicle information response message (3)

Message direction:	ECU → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	MessageCount Calibration Id = 6th response message	06	MC_CALID
#4	Data A: MSB (EGR Monitor Completion Condition Counts – numerator)	XX	EGRCOMP_A
#5	Data B: LSB (EGR Monitor Completion Condition Counts – numerator)	XX	EGRCOMP_B
#6	Data C: MSB (EGR Monitor Conditions Encountered Counts – denominator)	XX	EGRCOND_A
#7	Data D: LSB (EGR Monitor Conditions Encountered Counts – denominator)	XX	EGRCOND_B

Seventh positive response

Request vehicle information response message (4)

Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
Message direction:	ECU → External test equipment		
Message Type:	Response		
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	MessageCount Calibration Id = 7th response message	07	MC_CALID
#4	Data A: MSB (AIR Monitor Completion Condition Counts (Secondary Air) – numerator)	XX	AIRCOMP_A
#5	Data B: LSB (AIR Monitor Completion Condition Counts (Secondary Air) – numerator)	XX	AIRCOMP_B
#6	Data C: MSB (AIR Monitor Conditions Encountered Counts (Secondary Air) – denominator)	XX	AIRCOND_A
#7	Data D: LSB (AIR Monitor Conditions Encountered Counts (Secondary Air) – denominator)	XX	AIRCOND_B

Eighth positive response

Request vehicle information response message (4)

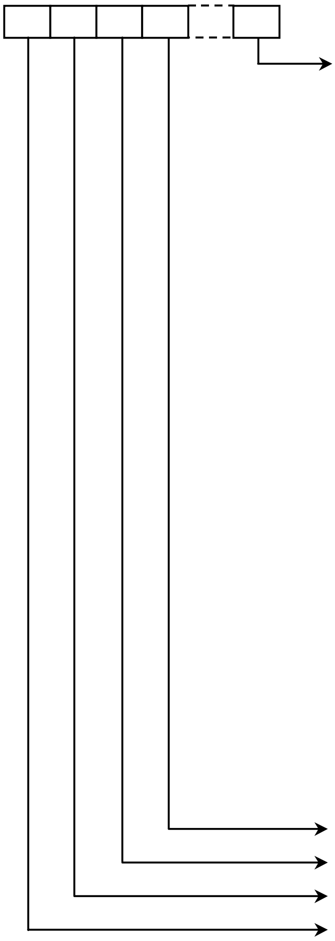
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
Message direction:	ECU → External test equipment		
Message Type:	Response		
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	MessageCount Calibration Id = 8th response message	08	MC_CALID
#4	Data A: MSB (EVAP Monitor Completion Condition Counts – numerator)	XX	EVAPCOMP_A
#5	Data B: LSB (EVAP Monitor Completion Condition Counts – numerator)	XX	EVAPCOMP_B
#6	Data C: MSB (EVAP Monitor Conditions Encountered Counts – denominator)	XX	EVAPCOND_A
#7	Data D: LSB (EVAP Monitor Conditions Encountered Counts – denominator)	XX	EVAPCOND_B

Formula section:

ACTION_ERRM_SelectRbmData (
INOUT < ListRbmData >,
OUT < ResultRbmData >
)

Call **API ACTION_ERRM_SelectRbmData**
Send positive(s) response(s)

ListRbmData : Software structure filled-up with 16 counters of 2 bytes coming from Rate-Based Monitoring



EVAP Monitor Conditions Encountered Counts (denominator - LSB byte)
EVAP Monitor Conditions Encountered Counts (denominator - MSB byte)
EVAP Monitor Completion Condition Counts (numerator - LSB byte)
EVAP Monitor Completion Condition Counts (numerator - MSB byte)
AIR Monitor Conditions Encountered Counts (denominator - LSB byte)
AIR Monitor Conditions Encountered Counts (denominator - MSB byte)
AIR Monitor Completion Condition Counts (numerator - LSB byte)
AIR Monitor Completion Condition Counts (numerator - MSB byte)
EGR Monitor Conditions Encountered Counts (denominator - LSB byte)
EGR Monitor Conditions Encountered Counts (denominator – MSB byte)
EGR Monitor Completion Condition Counts (numerator - LSB byte)
EGR Monitor Completion Condition Counts (numerator - MSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 2 (denominator – LSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 2 (denominator - MSB byte)
O2 Sensor Monitor Completion Counts Bank 2 (numerator – LSB byte)
O2 Sensor Monitor Completion Counts Bank 2 (numerator - MSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 1 (denominator - LSB byte)
O2 Sensor Monitor Conditions Encountered Counts Bank 1 (denominator - MSB byte)
O2 Sensor Monitor Completion Counts Bank 1 (numerator – LSB byte)
O2 Sensor Monitor Completion Counts Bank 1 (numerator – MSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 2 (denominator - LSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 2 (denominator - MSB byte)
Catalyst Monitor Completion Counts Bank 2 (numerator - LSB byte)
Catalyst Monitor Completion Counts Bank 2 (numerator - MSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 1 (denominator - LSB byte)
Catalyst Monitor Conditions Encountered Counts Bank 1 (denominator - MSB byte)
Catalyst Monitor Completion Counts Bank 1 (numerator - LSB byte)
Catalyst Monitor Completion Counts Bank 1 (numerator - MSB byte)
Ignition Counter (LSB byte)
Ignition Counter (MSB byte)
OBD Monitoring Conditions Encountered Counts (general denominator - LSB byte)
OBD Monitoring Conditions Encountered Counts (general denominator - MSB byte)

54.15 Software and system configuration for OBD communication

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MOD_1_PID_1_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (01h - 08h)					
MOD_1_PID_2_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (09h - 10h)					
MOD_1_PID_3_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (11h - 18h)					
MOD_1_PID_4_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (19h - 20h)					
MOD_1_PID_5_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (21h - 28h)					
MOD_1_PID_6_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (29h - 30h)					
MOD_1_PID_7_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (31h - 38h)					
MOD_1_PID_8_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (39h - 40h)					
MOD_1_PID_9_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (41h - 48h)					
MOD_1_PID_A_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (49h - 50h)					
MOD_1_PID_B_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (51h - 58h)					
MOD_1_PID_C_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 01 (59h - 60h)					
MOD_2_PID_1_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (01h - 09h)					
MOD_2_PID_2_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (09h - 10h)					
MOD_2_PID_3_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (11h - 18h)					
MOD_2_PID_4_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (19h - 20h)					
MOD_2_PID_5_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (21h - 28h)					
MOD_2_PID_6_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (29h - 30h)					
MOD_2_PID_7_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (31h - 38h)					
MOD_2_PID_8_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (39h - 40h)					
MOD_2_PID_9_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (41h - 48h)					
MOD_2_PID_A_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (49h - 50h)					
MOD_2_PID_B_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (51h - 58h)					
MOD_2_PID_C_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the PIDs supported by the mode 02 (59h - 60h)					
MOD_5_TID_1_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (01h - 08h)					
MOD_5_TID_2_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (09h - 10h)					
MOD_5_TID_3_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (11h - 18h)					
MOD_5_TID_4_DIAG	O	0...FFH	0...255	1	[-]

Byte which defines list of the TIDs supported by the mode 05 (19h - 20h)					
MOD_5_TID_5_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (21h - 28h)					
MOD_5_TID_6_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (29h - 30h)					
MOD_5_TID_7_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (31h - 38h)					
MOD_5_TID_8_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (39h - 40h)					
MOD_5_TID_9_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (41h - 48h)					
MOD_5_TID_10_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05(49h - 50h)					
MOD_5_TID_11_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (51h - 58h)					
MOD_5_TID_12_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (59h - 60h)					
MOD_5_TID_13_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (61h - 68h)					
MOD_5_TID_14_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (69h - 70h)					
MOD_5_TID_15_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (71h - 78h)					
MOD_5_TID_16_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (79h - 80h)					
MOD_5_TID_17_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (81h - 88h)					
MOD_5_TID_18_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (89h - 90h)					
MOD_5_TID_19_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05 (91h - 98h)					
MOD_5_TID_20_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 05(99h - A0h)					
MOD_6_TID_1_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 06 (01h - 08h)					
MOD_6_TID_2_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 06 (09h - 10h)					
MOD_6_TID_3_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 06 (11h - 18h)					
MOD_6_TID_4_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 06 (19h - 20h)					
MOD_9_VIT_1_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 09 (01h - 09h) for ISO 15765-4					
MOD_9_VIT_2_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 09 (09h - 10h) for ISO 15765-4					
MOD_9_VIT_3_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 09 (11h - 18h)) for ISO 15765-4					
MOD_9_VIT_4_DIAG	O	0...FFH	0...255	1	[-]
Byte which defines list of the TIDs supported by the mode 09 (19h - 20h) for ISO 15765-4					

Input data:

LV_VAR_OBDC_CAN			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MOD_1_PID_1_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (01h - 08h)					
C_MOD_1_PID_2_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (09h - 10h)					
C_MOD_1_PID_3_DIAG	1	0...FFH	0...255	1	[-]
Byte which defines the PIDs supported by the mode 01 (11h - 18h)					
C_MOD_1_PID_4_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (19h - 20h)					
C_MOD_1_PID_5_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (21h - 28h)					
C_MOD_1_PID_6_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (29h - 30h)					
C_MOD_1_PID_7_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (31h - 38h)					
C_MOD_1_PID_8_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (39h - 40h)					
C_MOD_1_PID_9_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (41h - 48h)					
C_MOD_1_PID_A_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (49h - 50h)					
C_MOD_1_PID_B_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (51h - 58h)					
C_MOD_1_PID_C_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (59h - 60h)					
C_MOD_2_PID_1_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (01h - 08h)					
C_MOD_2_PID_2_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (09h - 10h)					
C_MOD_2_PID_3_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (11h - 18h)					
C_MOD_2_PID_4_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 01 (19h - 20h)					
C_MOD_2_PID_5_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (21h - 28h)					
C_MOD_2_PID_6_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (29h - 30h)					
C_MOD_2_PID_7_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (31h - 38h)					
C_MOD_2_PID_8_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (39h - 40h)					
C_MOD_2_PID_9_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (41h - 48h)					
C_MOD_2_PID_A_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (49h - 50h)					
C_MOD_2_PID_B_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (51h - 58h)					
C_MOD_2_PID_C_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the PIDs supported by the mode 02 (59h - 60h)					
C_MOD_5_TID_1_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (01h - 08h)					
C_MOD_5_TID_2_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (09h - 10h)					

C_MOD_5_TID_3_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (11h - 18h)					
C_MOD_5_TID_4_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (19h - 20h)					
C_MOD_5_TID_5_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (21h - 28h)					
C_MOD_5_TID_6_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (29h - 30h)					
C_MOD_5_TID_7_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (31h - 38h)					
C_MOD_5_TID_8_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (39h - 40h)					
C_MOD_5_TID_9_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (41h - 48h)					
C_MOD_5_TID_10_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (49h - 50h)					
C_MOD_5_TID_11_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (51h - 58h)					
C_MOD_5_TID_12_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (59h - 60h)					
C_MOD_5_TID_13_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (61h - 68h)					
C_MOD_5_TID_14_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (69h - 70h)					
C_MOD_5_TID_15_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (71h - 78h)					
C_MOD_5_TID_16_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (79h - 80h)					
C_MOD_5_TID_17_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (81h - 88h)					
C_MOD_5_TID_18_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (89h - 90h)					
C_MOD_5_TID_19_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (91h - 98h)					
C_MOD_5_TID_20_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 05 (99h - A0h)					
C_MOD_6_TID_1_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 06 (01h - 08h)					
C_MOD_6_TID_2_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 06 (09h - 10h)					
C_MOD_6_TID_3_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 06 (11h - 18h)					
C_MOD_6_TID_4_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the TIDs supported by the mode 06 (19h - 20h)					
C_MOD_9_VIT_1_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the VITs supported by the mode 09 (01h - 08h) for ISO 15765-4					
C_MOD_9_VIT_2_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the VITs supported by the mode 09 (09h - 10h) for ISO 15765-4					
C_MOD_9_VIT_3_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the VITs supported by the mode 09 (11h - 18h) for ISO 15765-4					
C_MOD_9_VIT_4_DIAG	1	0...FFH	0...255	1	[-]
Calibration byte which defines the VITs supported by the mode 09 (19h - 20h) for ISO 15765-4					
C_OBD_REQ	1	0...FFH	0...255	1	[-]
Byte which indicates the OBD requirements					
C_CLC_READY_OBD_1	1	0...FFH	0...255	1	[-]
Calibration byte to force readiness to 0 without calculation					
C_VLS_DOWN_CYC_MIN_SAE	1	0...FFH	0...1.275	0.005	[V]
Minimum downstream sensor test cycle voltage, Mode 5					
C_VLS_DOWN_CYC_MAX_SAE	1	0...FFH	0...1.275	0.005	[V]
Maximum downstream sensor test cycle voltage, Mode 5					

LC_VAR_OBDC_CAN	1	0...1H	0...1	1	[-]
Logical switch to activate OBD communication on CAN (=1) / or K-line (=0)					
C_VLS_DOWN_AFL_AFR_SAE	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lean to rich threshold for downstream lambda sensor for MODE 06h					
C_VLS_DOWN_AFR_AFL_SAE	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Rich to lean threshold for downstream lambda sensor for MODE 06h					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_MOD_1_PID_1_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (01h - 08h)					
NC_MOD_1_PID_2_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (09h - 10h)					
NC_MOD_1_PID_3_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (11h - 18h)					
NC_MOD_1_PID_4_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (19h - 20h)					
NC_MOD_1_PID_5_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (21h - 28h)					
NC_MOD_1_PID_6_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (29h - 30h)					
NC_MOD_1_PID_7_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (31h - 38h)					
NC_MOD_1_PID_8_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (39h - 40h)					
NC_MOD_1_PID_9_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (41h - 48h)					
NC_MOD_1_PID_A_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (49h - 50h)					
NC_MOD_1_PID_B_DIAG	-	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (51h - 58h)					
NC_MOD_1_PID_C_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (59h - 60h)					
NC_MOD_2_PID_1_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (01h - 08h)					
NC_MOD_2_PID_2_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (09h - 10h)					
NC_MOD_2_PID_3_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (11h - 18h)					
NC_MOD_2_PID_4_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 01 (19h - 20h)					
NC_MOD_2_PID_5_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (21h - 28h)					
NC_MOD_2_PID_6_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (29h - 30h)					
NC_MOD_2_PID_7_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (31h - 38h)					
NC_MOD_2_PID_8_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (39h - 40h)					
NC_MOD_2_PID_9_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (41h - 48h)					
NC_MOD_2_PID_A_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (49h - 50h)					
NC_MOD_2_PID_B_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (51h - 58h)					
NC_MOD_2_PID_C_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the PIDs supported by the mode 02 (59h - 60h)					
NC_MOD_5_TID_1_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (01h - 08h)					

NC_MOD_5_TID_2_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (09h - 10h)					
NC_MOD_5_TID_3_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (11h - 18h)					
NC_MOD_5_TID_4_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (19h - 20h)					
NC_MOD_5_TID_5_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (21h - 28h)					
NC_MOD_5_TID_6_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (29h - 30h)					
NC_MOD_5_TID_7_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (31h - 38h)					
NC_MOD_5_TID_8_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines theTIDs supported by the mode 05 (39h - 40h)					
NC_MOD_5_TID_9_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (41h - 48h)					
NC_MOD_5_TID_10_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05(49h - 50h)					
NC_MOD_5_TID_11_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (51h - 58h)					
NC_MOD_5_TID_12_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (59h - 60h)					
NC_MOD_5_TID_13_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (61h - 68h)					
NC_MOD_5_TID_14_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (69h - 70h)					
NC_MOD_5_TID_15_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines theTIDs supported by the mode 05 (71h - 78h)					
NC_MOD_5_TID_15_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (71h - 78h)					
NC_MOD_5_TID_16_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (79h - 80h)					
NC_MOD_5_TID_17_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (81h - 88h)					
NC_MOD_5_TID_18_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (89h - 90h)					
NC_MOD_5_TID_19_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (91h - 98h)					
NC_MOD_5_TID_20_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 05 (99h - A0h)					
NC_MOD_6_TID_1_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 06 (01h - 08h)					
NC_MOD_6_TID_2_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 06 (09h - 10h)					
NC_MOD_6_TID_3_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 06 (11h - 18h)					
NC_MOD_6_TID_4_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the TIDs supported by the mode 06 (19h - 20h)					
NC_MOD_9_VIT_1_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the VITs supported by the mode 09 (01h - 08h) for ISO 15765-4					
NC_MOD_9_VIT_2_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the VITs supported by the mode 09 (09h - 10h)) for ISO 15765-4					
NC_MOD_9_VIT_3_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the VITs supported by the mode 09 (11h - 18h)) for ISO 15765-4					
NC_MOD_9_VIT_4_DIAG	1	0...FFH	0...255	1	[-]
Configuration byte which defines the VITs supported by the mode 09 (19h - 20h)) for ISO 15765-4					

54.15.1 Activation of OBD communication on CAN / K-line

FUNCTION DESCRIPTION:

General information:

Depending on calibration or coding, a different OBD communication can be applied.

Application conditions:

Activation: only once after ECU power-on reset

Formula section:

```
If      LC_VAR_OBDC_CAN = 1
Then    OBD communication on CAN is active (ISO 15765 – 4)
Else    If      LV_VAR_OBDC_CAN = 1
        Then    OBD communication on CAN is active (ISO 15765 – 4)
        Else    OBD communication on K-line is active (ISO 9141-2, ISO 14230-4)
        Endif
Endif
```

54.15.2 Configuration and calibration of OBD communication on K-line (ISO 9141-2, ISO 14230-4)

Description:

MOD_i_PID / TID / VIT_j_DIAG variable definition

MOD_i_PID / _TID / _VIT_j_DIAG defines supported PIDs list for mode i. This definition depends on software **configuration** (ie if it is implemented in the software) and depends on system. Indeed, for a same software, we can have different system corresponding.

To define the software configuration, we use constants (NC_MOD_i_XXX_j_DIAG)

To define system, we use functions (F_MOD_i_XXX_j_DIAG) depends on different **calibrations**.

System configuration (calibration):

The definition of F_MOD_i_XXX_j_DIAG is: C_MOD_i_XXX_j_DIAG.

Software configuration:

The definition of all NC_MOD_x_PID_y_DIAG are inside the concerning table:

e.g. in the table below NC_MOD_x_PID_y_DIAG = **11100101 = E5h**

NC_MOD_x_PID_y_DIAG		C_MOD_x_PID_y_DIAG
BIT	Function	Supported
0	PID 10h : Function H	**Yes**
1	PID 0Fh : Function G	No
2	PID 0Eh : Function F	**Yes**
3	PID 0Dh : Function E	No
4	PID 0Ch : Function D	No
5	PID 0Bh : Function C	**Yes**
6	PID 0Ah : Function B	**Yes**
7	PID 09h : Function A	**Yes**

Formula section:

MOD_i_XXX_j_DIAG = NC_MOD_i_XXX_j__DIAG && F_MOD_i_XXX_j_DIAG(calibration) with && = bit per bit and.

54.15.2.1 Supported PIDs for Mode 01h

Description:

No: the PID is not supported
Yes: the PID is supported

NC_MOD_1_PID_1_DIAG		C_MOD_1_PID_1_DIAG
BIT	Function	Supported
0	PID 08h : Short term fuel trim - Bank 2	**Yes**
1	PID 07h : Long term fuel trim - Bank 1	**Yes**
2	PID 06h : Short term fuel trim - Bank 1	**Yes**
3	PID 05h : Engine coolant temperature	**Yes**
4	PID 04h : Calculated load value	**Yes**
5	PID 03h : Fuel system status	**Yes**
6	PID 02h : Trouble code that caused required freeze frame	No
7	PID 01h : DTCs and supported tests	**Yes**

NC_MOD_1_PID_2_DIAG		C_MOD_1_PID_2_DIAG
BIT	Function	Supported
0	PID 10h : Air flow rate from MAF sensor	**Yes**
1	PID 0Fh : Intake air temperature	**Yes**
2	PID 0Eh : Ignition time advance	**Yes**
3	PID 0Dh : Vehicle speed	**Yes**
4	PID 0Ch : Engine RPM	**Yes**
5	PID 0Bh : Intake manifold absolute pressure	**Yes**
6	PID 0Ah : Fuel pressure (gauge)	**Yes**
7	PID 09h : Long term fuel trim - Bank 2	**Yes**

NC_MOD_1_PID_3_DIAG		C_MOD_1_PID_3_DIAG
BIT	Function	Supported
0	PID 18h : Bank 2 Sensor 1(binary)	No
1	PID 17h : Bank 2 Sensor 2	No
2	PID 16h : Bank 2 Sensor 1	No
3	PID 15h : Bank 1 Sensor 2	**Yes**
4	PID 14h : Bank 1 Sensor 1(binary)	No
5	PID 13h : Location of oxygen sensors	**Yes**
6	PID 12h : Secondary air status	**Yes**
7	PID 11h : Absolute throttle position sensor	**Yes**

NC_MOD_1_PID_4_DIAG		C_MOD_1_PID_4_DIAG
BIT	Function	Supported
0	PID 20h : PIDs supported ($21 - $40)	**Yes**
1	PID 1Fh : Time since engine start	**Yes**
2	PID 1Eh : Reserved	No
3	PID 1Dh : Reserved	No
4	PID 1Ch : OBD requirements	**Yes**
5	PID 1Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2	No
6	PID 1Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1	No
7	PID 19h : Bank 2 Sensor 2 or Bank 3 Sensor 2	**Yes**

NC_MOD_1_PID_5_DIAG		C_MOD_1_PID_5_DIAG
BIT	Function	Supported
0	PID 28h : Bank 2 Sensor 1 (Wide range O2S)	**Yes**
1	PID 27h : Bank 2 Sensor 2 (Wide range O2S)	No
2	PID 26h : Bank 2 Sensor 1 (Wide range O2S)	No
3	PID 25h : Bank 1 Sensor 2 (Wide range O2S)	No
4	PID 24h : Bank 1 Sensor 1 (Wide range O2S)	**Yes**
5	PID 23h : Fuel pressure high range (gauge)	**Yes**
6	PID 22h : Relative fuel pressure	No
7	PID 21h : Distance travelled while MIL is activated	**Yes**

NC_MOD_1_PID_6_DIAG		C_MOD_1_PID_6_DIAG
BIT	Function	Supported
0	PID 30h : Number of warm-ups since diagnostic trouble codes cleared	**Yes**
1	PID 2Fh : Fuel Level Input	**Yes**
2	PID 2Eh : Command Evaporative Purge	**Yes**
3	PID 2Dh : EGR Error	**Yes**
4	PID 2Ch : Command EGR	**Yes**
5	PID 2Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	No
6	PID 2Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	No
7	PID 29h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	No

NC_MOD_1_PID_7_DIAG		C_MOD_1_PID_7_DIAG
BIT	Function	Supported
0	PID 38h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	**Yes**
1	PID 37h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	No
2	PID 36h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	No
3	PID 35h : Bank 1 Sensor 2 (Wide range O2S)	No
4	PID 34h : Bank 1 Sensor 1 (Wide range O2S)	**Yes**
5	PID 33h : Barometric Pressure	**Yes**
6	PID 32h : Evap System Vapor Pressure	No
7	PID 31h : Distance since diagnostic trouble codes cleared	**Yes**

NC_MOD_1_PID_8_DIAG		C_MOD_1_PID_8_DIAG
BIT	Function	Supported
0	PID 40h : PIDs supported ($41 - $60)	**Yes**
1	PID 3Fh : Catalyst Temperature Bank 2, Sensor 2	**Yes**
2	PID 3Eh : Catalyst Temperature Bank 1, Sensor 2	**Yes**
3	PID 3Dh : Catalyst Temperature Bank 2, Sensor 1	**Yes**
4	PID 3Ch : Catalyst Temperature Bank 1, Sensor 1	**Yes**
5	PID 3Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	No
6	PID 3Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	No
7	PID 39h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	No

NC_MOD_1_PID_9_DIAG		C_MOD_1_PID_9_DIAG
BIT	Function	Supported
0	PID 48h : Absolute Throttle Position C	No
1	PID 47h : Absolute Throttle Position B	**Yes**
2	PID 46h : Ambient air temperature	**Yes**
3	PID 45h : Relative Throttle Position	**Yes**
4	PID 44h : Commanded Equivalence Ratio	**Yes**
5	PID 43h : Absolute Load Value	**Yes**
6	PID 42h : Control module voltage	**Yes**
7	PID 41h : Monitor status this driving cycle	**Yes**

NC_MOD_1_PID_A_DIAG		C_MOD_1_PID_A_DIAG
BIT	Function	Supported
0	PID 50h : Reserved	No
1	PID 4Fh : Reserved	No
2	PID 4Eh : Time since diagnostic trouble codes cleared	**Yes**
3	PID 4Dh : Minutes run by the engine while MIL activated	**Yes**
4	PID 4Ch : Commanded Throttle Actuator Control	**Yes**
5	PID 4Bh : Accelerator Pedal Position F	No
6	PID 4Ah : Accelerator Pedal Position E	**Yes**
7	PID 49h : Accelerator Pedal Position D	**Yes**

NC_MOD_1_PID_B_DIAG		C_MOD_1_PID_B_DIAG
BIT	Function	Supported
0	PID 58h : Reserved	No
1	PID 57h : Reserved	No
2	PID 56h : Reserved	No
3	PID 55h : Reserved	No
4	PID 54h : Reserved	No
5	PID 53h : Reserved	No
6	PID 52h : Reserved	No
7	PID 51h : Reserved	No

NC_MOD_1_PID_C_DIAG		C_MOD_1_PID_C_DIAG
BIT	Function	Supported
0	PID 60h : Reserved	No
1	PID 5Fh : Reserved	No
2	PID 5Eh : Reserved	No
3	PID 5Dh : Reserved	No
4	PID 5Ch : Reserved	No
5	PID 5Bh : Reserved	No
6	PID 5Ah : Reserved	No
7	PID 59h : Reserved	No

54.15.2.2 Supported PIDs for Mode 02h

Description:

No: the PID is not supported
Yes: the PID is supported

NC_MOD_2_PID_1_DIAG	C_MOD_2_PID_1_DIAG	
BIT	Function	Supported
0	PID 08h : Short term fuel trim - Bank 2 and Bank 4	**Yes**
1	PID 07h : Long term fuel trim - Bank 1 and Bank 3	**Yes**
2	PID 06h : Short term fuel trim - Bank 1 and Bank 3	**Yes**
3	PID 05h : Engine coolant temperature	**Yes**
4	PID 04h : Calculated load value	**Yes**
5	PID 03h : Fuel system status	**Yes**
6	PID 02h : Trouble code that caused required freeze frame	**Yes**
7	PID 01h : DTCs and supported tests	No

NC_MOD_2_PID_2_DIAG	C_MOD_2_PID_2_DIAG	
BIT	Function	Supported
0	PID 10h : Air flow rate from MAF sensor	**Yes**
1	PID 0Fh : Intake air temperature	**Yes**
2	PID 0Eh : Ignition time advance	**Yes**
3	PID 0Dh : Vehicle speed	**Yes**
4	PID 0Ch : Engine RPM	**Yes**
5	PID 0Bh : Intake manifold absolute pressure	**Yes**
6	PID 0Ah : Fuel pressure (gauge)	**Yes**
7	PID 09h : Long term fuel trim - Bank 2 and Bank 4	**Yes**

NC_MOD_2_PID_3_DIAG		C_MOD_2_PID_3_DIAG
BIT	Function	Supported
0	PID 18h : Not required	No
1	PID 17h : Not required	No
2	PID 16h : Not required	No
3	PID 15h : Not required	No
4	PID 14h : Not required	No
5	PID 13h : Not required	No
6	PID 12h : Secondary air status	**Yes**
7	PID 11h : Absolute throttle position sensor	**Yes**

NC_MOD_2_PID_4_DIAG		C_MOD_2_PID_4_DIAG
BIT	Function	Supported
0	PID 20h : PIDs supported ($21 - $40)	**Yes**
1	PID 1Fh : Time since engine start	**Yes**
2	PID 1Eh : Not required	No
3	PID 1Dh : Not required	No
4	PID 1Ch : Not required	No
5	PID 1Bh : Not required	No
6	PID 1Ah : Not required	No
7	PID 19h : Not required	No

NC_MOD_2_PID_5_DIAG		C_MOD_2_PID_5_DIAG
BIT	Function	Supported
0	PID 28h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	No
1	PID 27h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	No
2	PID 26h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	No
3	PID 25h : Bank 1 Sensor 2 (Wide range O2S)	No
4	PID 24h : Bank 1 Sensor 1 (Wide range O2S)	No
5	PID 23h : Diesel fuel pressure (gauge)	**Yes**
6	PID 22h : Relative fuel pressure	No
7	PID 21h : Distance travelled while MIL is activated	No

NC_MOD_2_PID_6_DIAG		C_MOD_2_PID_6_DIAG
BIT	Function	Supported
0	PID 30h : Number of warm-ups since diagnostic trouble codes cleared	No
1	PID 2Fh : Fuel Level Input	**Yes**
2	PID 2Eh : Command Evaporative Purge	**Yes**
3	PID 2Dh : EGR Error	**Yes**
4	PID 2Ch : command EGR	**Yes**
5	PID 2Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	No
6	PID 2Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	No
7	PID 29h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	No

NC_MOD_2_PID_7_DIAG		C_MOD_2_PID_7_DIAG
BIT	Function	Supported
0	PID 38h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	No
1	PID 37h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	No

2	PID 36h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	No
3	PID 35h : Bank 1 Sensor 2 (Wide range O2S)	No
4	PID 34h : Bank 1 Sensor 1 (Wide range O2S)	No
5	PID 33h : Barometric Pressure	**Yes**
6	PID 32h : Evap System Vapor Pressure	No
7	PID 31h : Distance since diagnostic trouble codes cleared	No

NC_MOD_2_PID_8_DIAG	C_MOD_2_PID_8_DIAG	
BIT	Function	Supported
0	PID 40h : PIDs supported ($41 - $60)	**Yes**
1	PID 3Fh : Catalyst Temperature Bank 2, Sensor 2	No
2	PID 3Eh : Catalyst Temperature Bank 1, Sensor 2	No
3	PID 3Dh : Catalyst Temperature Bank 2, Sensor 1	No
4	PID 3Ch : Catalyst Temperature Bank 1, Sensor 1	No
5	PID 3Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	No
6	PID 3Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	No
7	PID 39h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	No

NC_MOD_2_PID_9_DIAG	C_MOD_2_PID_9_DIAG	
BIT	Function	Supported
0	PID 48h : Absolute Throttle Position C	No
1	PID 47h : Absolute Throttle Position B	**Yes**
2	PID 46h : Ambient air temperature	**Yes**
3	PID 45h : Relative Throttle Position	**Yes**
4	PID 44h : Commanded Equivalence Ratio	**Yes**
5	PID 43h : Absolute Load Value	**Yes**
6	PID 42h : Control module voltage	**Yes**
7	PID 41h : Monitor status this driving cycle	No

NC_MOD_2_PID_A_DIAG	C_MOD_2_PID_A_DIAG	
BIT	Function	Supported
0	PID 50h : Reserved	No
1	PID 4Fh : Reserved	No
2	PID 4Eh : Time since diagnostic trouble codes cleared	No
3	PID 4Dh : Minutes run by the engine while MIL activated	No
4	PID 4Ch : Commanded Throttle Actuator Control	**Yes**
5	PID 4Bh : Accelerator Pedal Position F	No
6	PID 4Ah : Accelerator Pedal Position E	**Yes**
7	PID 49h : Accelerator Pedal Position D	**Yes**

NC_MOD_2_PID_B_DIAG		C_MOD_2_PID_B_DIAG
BIT	Function	Supported
0	PID 58h : Reserved	No
1	PID 57h : Reserved	No
2	PID 56h : Reserved	No
3	PID 55h : Reserved	No
4	PID 54h : Reserved	No
5	PID 53h : Reserved	No
6	PID 52h : Reserved	No
7	PID 51h : Reserved	No

NC_MOD_2_PID_C_DIAG		C_MOD_2_PID_C_DIAG
BIT	Function	Supported
0	PID 60h : Reserved	No
1	PID 5Fh : Reserved	No
2	PID 5Eh : Reserved	No
3	PID 5Dh : Reserved	No
4	PID 5Ch : Reserved	No
5	PID 5Bh : Reserved	No
6	PID 5Ah : Reserved	No
7	PID 59h : Reserved	No

54.15.2.3 Supported TIDs for mode 05h

Description:

No: the TID is not supported, **Yes**: the TID is supported

NC_MOD_5_TID_1_DIAG		C_MOD_5_TID_1_DIAG
BIT	Function	Supported
0	TID 08h : Maximum sensor voltage for test cycle (binary) **B1S2 B2S2**	**Yes**
1	TID 07h : Minimum sensor voltage for test cycle (binary) **B1S2 B2S2**	**Yes**
2	TID 06h : Lean to rich senor switch time (binary) B1S1 B2S1	No
3	TID 05h : Rich to lean senor switch time (binary) B1S1 B2S1	No
4	TID 04h : High sensor voltage for switch time calculation (binary) B1S1 B2S1	No
5	TID 03h : Low sensor voltage for switch time calculation (binary) B1S1 B2S1	No
6	TID 02h : Lean to rich sensor threshold voltage (binary) **B1S2 B2S2**	**Yes**
7	TID 01h : Rich to lean sensor threshold voltage (binary) **B1S2 B2S2**	**Yes**

NC_MOD_5_TID_2_DIAG		C_MOD_5_TID_2_DIAG
BIT	Function	Supported
0	TID 10h : reserved	No
1	TID 0Fh : reserved	No
2	TID 0Eh : reserved	No
3	TID 0Dh : reserved	No
4	TID 0Ch : reserved	No
5	TID 0Bh : reserved	No
6	TID 0Ah : reserved	No
7	TID 09h : reserved	No

NC_MOD_5_TID_3_DIAG		C_MOD_5_TID_3_DIAG
BIT	Function	Supported
0	TID 18h : reserved	No
1	TID 17h : reserved	No
2	TID 16h : reserved	No
3	TID 15h : reserved	No
4	TID 14h : reserved	No
5	TID 13h : reserved	No
6	TID 12h : reserved	No
7	TID 11h : reserved	No

NC_MOD_5_TID_4_DIAG		C_MOD_5_TID_4_DIAG
BIT	Function	Supported
0	TID 20h : TIDs supported ($21 - $40)	**Yes**
1	TID 1Fh : reserved	No
2	TID 1Eh : reserved	No
3	TID 1Dh : reserved	No
4	TID 1Ch : reserved	No
5	TID 1Bh : reserved	No
6	TID 1Ah : reserved	No
7	TID 19h : reserved	No

NC_MOD_5_TID_5_DIAG		C_MOD_5_TID_5_DIAG
BIT	Function	Supported
0	TID 28h : reserved	No
1	TID 27h : reserved	No
2	TID 26h : reserved	No
3	TID 25h : reserved	No
4	TID 24h : reserved	No
5	TID 23h : reserved	No
6	TID 22h : reserved	No
7	TID 21h : reserved	No

NC_MOD_5_TID_6_DIAG	C_MOD_5_TID_6_DIAG	
BIT	Function	Supported
0	TID 30h : Reserved	No
1	TID 2Fh : Reserved	No
2	TID 2Eh : Reserved	No
3	TID 2Dh : Reserved	No
4	TID 2Ch : Reserved	No
5	TID 2Bh : Reserved	No
6	TID 2Ah : Reserved	No
7	TID 29h : Reserved	No

NC_MOD_5_TID_7_DIAG	C_MOD_5_TID_7_DIAG	
BIT	Function	Supported
0	TID 38h : Reserved	No
1	TID 37h : Reserved	No
2	TID 36h : Reserved	No
3	TID 35h : Reserved	No
4	TID 34h : Reserved	No
5	TID 33h : Reserved	No
6	TID 32h : Reserved	No
7	TID 31h : Reserved	No

NC_MOD_5_TID_8_DIAG	C_MOD_5_TID_8_DIAG	
BIT	Function	Supported
0	TID 40h : TIDs supported ($41 - $60)	**Yes**
1	TID 3Fh : Reserved	No
2	TID 3Eh : Reserved	No
3	TID 3Dh : Reserved	No
4	TID 3Ch : Reserved	No
5	TID 3Bh : Reserved	No
6	TID 3Ah : Reserved	No
7	TID 39h : Reserved	No

NC_MOD_5_TID_9_DIAG	C_MOD_5_TID_9_DIAG	
BIT	Function	Supported
0	TID 48h : Reserved	No
1	TID 47h : Reserved	No
2	TID 46h : Reserved	No
3	TID 45h : Reserved	No
4	TID 44h : Reserved	No
5	TID 43h : Reserved	No
6	TID 42h : Reserved	No
7	TID 41h : Reserved	No

NC_MOD_5_TID_10_DIAG		C_MOD_5_TID_10_DIAG
BIT	Function	Supported
0	TID 50h : Reserved	No
1	TID 4Fh : Reserved	No
2	TID 4Eh : Reserved	No
3	TID 4Dh : Reserved	No
4	TID 4Ch : Reserved	No
5	TID 4Bh : Reserved	No
6	TID 4Ah : Reserved	No
7	TID 49h : Reserved	No

NC_MOD_5_TID_11_DIAG		C_MOD_5_TID_11_DIAG
BIT	Function	Supported
0	TID 58h : Reserved	No
1	TID 57h : Reserved	No
2	TID 56h : Reserved	No
3	TID 55h : Reserved	No
4	TID 54h : Reserved	No
5	TID 53h : Reserved	No
6	TID 52h : Reserved	No
7	TID 51h : Reserved	No

NC_MOD_5_TID_12_DIAG		C_MOD_5_TID_12_DIAG
BIT	Function	Supported
0	TID 60h : TIDs supported ($61 - $80)	**Yes**
1	TID 5Fh : Reserved	No
2	TID 5Eh : Reserved	No
3	TID 5Dh : Reserved	No
4	TID 5Ch : Reserved	No
5	TID 5Bh : Reserved	No
6	TID 5Ah : Reserved	No
7	TID 59h : Reserved	No

NC_MOD_5_TID_13_DIAG		C_MOD_5_TID_13_DIAG
BIT	Function	Supported
0	TID 68h : Reserved	No
1	TID 67h : Reserved	No
2	TID 66h : Reserved	No
3	TID 65h : Reserved	No
4	TID 64h : Reserved	No
5	TID 63h : Reserved	No
6	TID 62h : Reserved	No
7	TID 61h : Reserved	No

NC_MOD_5_TID_14_DIAG		C_MOD_5_TID_14_DIAG
BIT	Function	Supported
0	TID 70h : Reserved	No
1	TID 6Fh : Reserved	No
2	TID 6Eh : Reserved	No
3	TID 6Dh : Reserved	No
4	TID 6Ch : Reserved	No
5	TID 6Bh : Reserved	No
6	TID 6Ah : Reserved	No
7	TID 69h : Reserved	No

NC_MOD_5_TID_15_DIAG		C_MOD_5_TID_15_DIAG
BIT	Function	Supported
0	TID 78h : Reserved	No
1	TID 77h : Reserved	No
2	TID 76h : Reserved	No
3	TID 75h : Reserved	No
4	TID 74h : Reserved	No
5	TID 73h : Reserved	No
6	TID 72h : Reserved	No
7	TID 71h : Reserved	No

NC_MOD_5_TID_16_DIAG		C_MOD_5_TID_16_DIAG
BIT	Function	Supported
0	TID 80h : TIDs supported ($81 - $A0)	**Yes**
1	TID 7Fh : Reserved	No
2	TID 7Eh : Reserved	No
3	TID 7Dh : Reserved	No
4	TID 7Ch : Reserved	No
5	TID 7Bh : Reserved	No
6	TID 7Ah : Reserved	No
7	TID 79h : Reserved	No

NC_MOD_5_TID_17_DIAG		C_MOD_5_TID_17_DIAG
BIT	Function	Supported
0	TID 88h : not defined	No
1	TID 87h : not defined	No
2	TID 86h : not defined	No
3	TID 85h : not defined	No
4	TID 84h : Fuel trim diagnosis	**Yes**
5	TID 83h : Dynamic diagnosis	**Yes**
6	TID 82h : A/F setpoint linear	**Yes**
7	TID 81h : Measured A/F ratio linear	**Yes**

NC_MOD_5_TID_18_DIAG		C_MOD_5_TID_18_DIAG
BIT	Function	Supported
0	TID 90h : Reserved	No
1	TID 8Fh : Reserved	No
2	TID 8Eh : Reserved	No
3	TID 8Dh : Reserved	No
4	TID 8Ch : Reserved	No
5	TID 8Bh : Reserved	No
6	TID 8Ah : Reserved	No
7	TID 89h : Reserved	No

NC_MOD_5_TID_19_DIAG		C_MOD_5_TID_19_DIAG
BIT	Function	Supported
0	TID 98h : Reserved	No
1	TID 97h : Reserved	No
2	TID 96h : Reserved	No
3	TID 95h : Reserved	No
4	TID 94h : Reserved	No
5	TID 93h : Reserved	No
6	TID 92h : Reserved	No
7	TID 91h : Reserved	No

NC_MOD_5_TID_20_DIAG		C_MOD_5_TID_20_DIAG
BIT	Function	Supported
0	TID A0h : TIDs supported ($A1 - $C0)	No
1	TID 9Fh : Reserved	No
2	TID 9Eh : Reserved	No
3	TID 9Dh : Reserved	No
4	TID 9Ch : Reserved	No
5	TID 9Bh : Reserved	No
6	TID 9Ah : Reserved	No
7	TID 99h : Reserved	No

54.15.2.4 Supported TIDs for Mode 06h

Description:

No: the TID is not supported
Yes: the TID is supported

NC_MOD_6_TID_1_DIAG		C_MOD_6_TID_1_DIAG
BIT	Function	Supported
0	TID 08h : Reserved	No
1	TID 07h : Reserved	No
2	TID 06h : Oxygen sensor heater	**Yes**
3	TID 05h : Evaproative monitor	**Yes**
4	TID 04h : Reserved	No
5	TID 03h : Secondary air monitor	**Yes**
6	TID 02h : Oxygen sensor monitor	**Yes**
7	TID 01h : Catalyst monitor	**Yes**

NC_MOD_6_TID_2_DIAG		C_MOD_6_TID_2_DIAG
BIT	Function	Supported
0	TID 10h : reserved	No
1	TID 0Fh : reserved	No
2	TID 0Eh : reserved	No
3	TID 0Dh : reserved	No
4	TID 0Ch : reserved	No
5	TID 0Bh : reserved	No
6	TID 0Ah : reserved	No
7	TID 09h : reserved	No

NC_MOD_6_TID_3_DIAG		C_MOD_6_TID_3_DIAG
BIT	Function	Supported
0	TID 18h : reserved	No
1	TID 17h : reserved	No
2	TID 16h : reserved	No
3	TID 15h : reserved	No
4	TID 14h : reserved	No
5	TID 13h : reserved	No
6	TID 12h : reserved	No
7	TID 11h : reserved	No

NC_MOD_6_TID_4_DIAG		C_MOD_6_TID_4_DIAG
BIT	Function	Supported
0	TID 20h : TIDs supported ($21 - $40)	No
1	TID 1Fh : reserved	No
2	TID 1Eh : reserved	No
3	TID 1Dh : reserved	No
4	TID 1Ch : reserved	No
5	TID 1Bh : reserved	No
6	TID 1Ah : reserved	No
7	TID 19h : reserved	No

54.15.2.5 Supported TIDs for Mode 08h (not supported yet)

54.15.2.6 Supported VITs for Mode 09h

Description:

No: the VIT is not supported, **Yes**: the VIT is supported

NC_MOD_9_VIT_1_DIAG		C_MOD_9_VIT_1_DIAG
BIT	Function	Supported
0	VIT 08h : In-Use Performance Tracking	**Yes**
1	VIT 07h : Message Count IPT	**Yes**
2	VIT 06h : Calibration Verification Numbers	**Yes**
3	VIT 05h : Number of response for Calibration Verification Numbers	**Yes**
4	VIT 04h : Calibration IDs	**Yes**
5	VIT 03h : Number of response for Calibration IDs	**Yes**
6	VIT 02h : Vehicle Identification Number	**Yes**
7	VIT 01h : Number of response for Vehicle Identification Number	**Yes**

NC_MOD_9_VIT_2_DIAG		C_MOD_9_VIT_2_DIAG
BIT	Function	Supported
0	VIT 10h : Reserved	No
1	VIT 0Fh : Reserved	No
2	VIT 0Eh : Reserved	No
3	VIT 0Dh : Reserved	No
4	VIT 0Ch : Reserved	No
5	VIT 0Bh : Reserved	No
6	VIT 0Ah : Reserved	No
7	VIT 09h : Reserved	No

NC_MOD_9_VIT_3_DIAG		C_MOD_9_VIT_3_DIAG
BIT	Function	Supported
0	VIT 18h : Reserved	No
1	VIT 17h : Reserved	No
2	VIT 16h : Reserved	No
3	VIT 15h : Reserved	No
4	VIT 14h : Reserved	No
5	VIT 13h : Reserved	No
6	VIT 12h : Reserved	No
7	VIT 11h : Reserved	No

NC_MOD_9_VIT_4_DIAG		C_MOD_9_VIT_4_DIAG
BIT	Function	Supported
0	VIT 20h : VITs supported ($21 - $40)	No
1	VIT 1Fh : Reserved	No
2	VIT 1Eh : Reserved	No
3	VIT 1Dh : Reserved	No
4	VIT 1Ch : Reserved	No
5	VIT 1Bh : Reserved	No
6	VIT 1Ah : Reserved	No
7	VIT 19h : Reserved	No

54.16 Service 01h - Request current powertrain diagnostic data

54.16.1 Functional description

The purpose of this service is to allow access to current emission-related data values, including analogue inputs and outputs, digital inputs and outputs, and system status information. The request for information includes a parameter identification (PID) value that indicates to the on-board system the specific information requested. PID specifications, scaling information, and display formats are included in Annex B of ISO 15031-5.4.

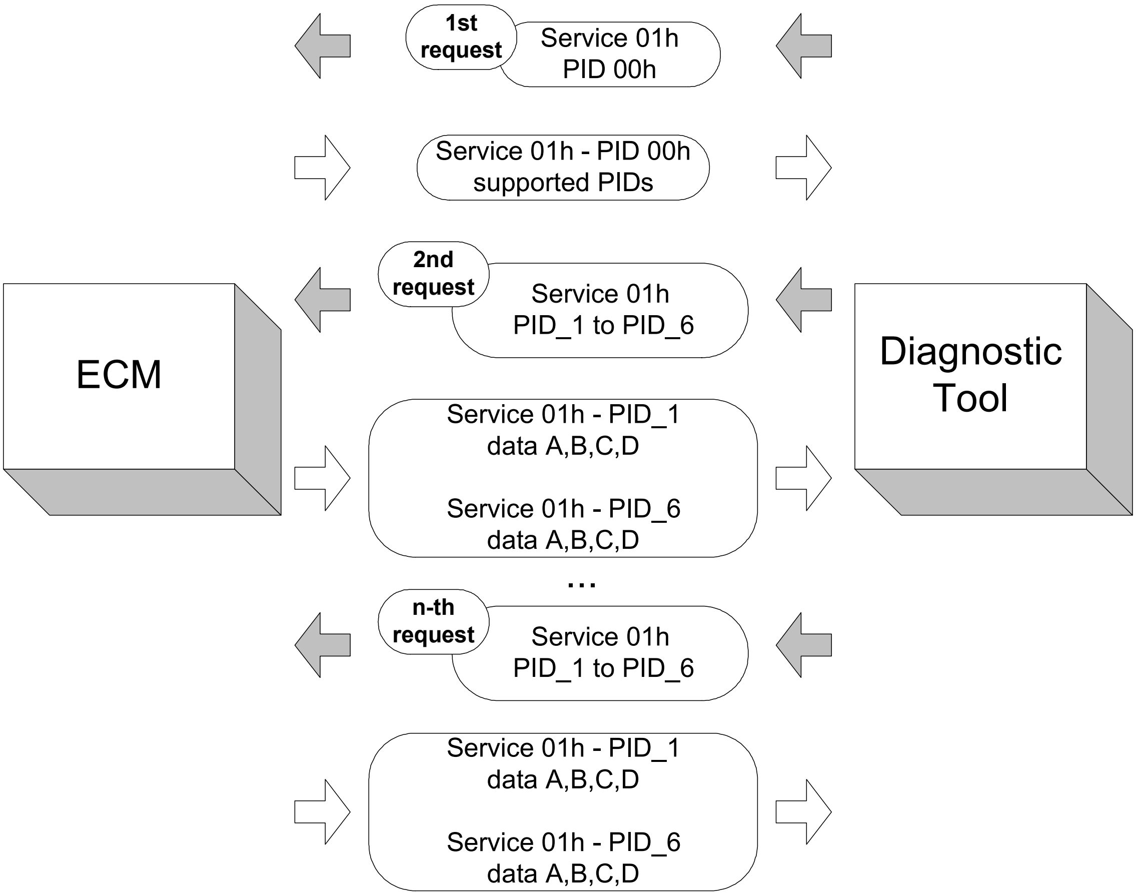
The ECU(s) will respond to this message by transmitting the requested data value last determined by the system. All data values returned for sensor readings will be actual readings, not default or substitute values used by the system because of a fault with that sensor.

Not all PIDs are applicable or supported by all systems. PID 00h is a bit-encoded value that indicates for each ECU which PIDs are supported. PID 00h indicates support for PIDs from 01h to 20h. PID 20h indicates support for PIDs 21h through 40h, etc. PID 00h is required for those ECUs that respond to a corresponding service 01h request message as specified in Annex A of ISO 15031-5.4.

IMPORTANT — All emissions-related OBD ECUs shall support service 01h and PID 00h. Service 01h with PID 00h is defined as the universal "initialization/keep alive/ping" message for all emissions-related OBD ECUs.

The request message may contain up to six (6) PIDs. An external test equipment is not allowed to request a combination of PIDs supported and PIDs, which report data values. The ECU shall support requests for up to six (6) PIDs. The request message may contain the same PID multiple times. The ECU shall treat each PID as a separate parameter and respond with data for each PID (data returned may be different for the same PID) as often as requested.

The order of the PIDs in the response message is not required to match the order in the request message.


1st
request
Service 01h
PID 00h
Service 01h - PID 00h
supported PIDs
2nd
request
Service 01h
PID_1 to PID_6
ECM
Diagnostic
Tool
Service 01h - PID_1
data A,B,C,D
Service 01h - PID_6
data A,B,C,D
...
n-th
request
Service 01h
PID_1 to PID_6
Service 01h - PID_1
data A,B,C,D
Service 01h - PID_6
data A,B,C,D

54.16.2 Message data bytes

54.16.2.1 Read supported PIDs

Request current powertrain diagnostic data request message (read supported PIDs)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request current powertrain diagnostic data request SID**	**M**	**01**	**SIDRQ**
#2	PID#1 (PIDs supported: see Annex A of ISO 15031-5.4)	M	xx	PID
#3	PID#2 (PIDs supported: see Annex A of ISO 15031-5.4)	U	xx	PID
#4	PID#3 (PIDs supported: see Annex A of ISO 15031-5.4)	U	xx	PID
#5	PID#4 (PIDs supported: see Annex A of ISO 15031-5.4)	U	xx	PID
#6	PID#5 (PIDs supported: see Annex A of ISO 15031-5.4)	U	xx	PID
#7	PID#6 (PIDs supported: see Annex A of ISO 15031-5.4)	U	xx	PID
U = User Optional — PID may be included to avoid multiple PID supported request messages				

NOTE To request PIDs supported range from C1h - FFh another request message with PID#1 = C0h and PID#2 = E0h shall be sent to the vehicle.

54.16.2.2 Report supported PIDs

ECU(s) must respond to all supported ranges if requested. A range is defined as a block of 32 PIDs (e.g. range #1: PID 01h-20h). The ECU shall not respond to unsupported PID ranges unless subsequent ranges have a supported PID(s).

Request current powertrain diagnostic data response message (report supported PIDs)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request current powertrain diagnostic data response SID**	**M**	**41**	**SIDPR**
	data record of supported PIDs = [			PIDREC_
#2	1st supported PID	M	xx	PID
#3	Data A: supported PIDs,	M	xx	DATA_A
#4	Data B: supported PIDs,	M	xx	DATA_B
#5	Data C: supported PIDs,	M	xx	DATA_C
#6	Data D: supported PIDs]	M	xx	DATA_D
:	:	:	:	:
	data record of supported PIDs = [			PIDREC_
#n-4	mth supported PID	C1	xx	PID
#n-3	Data A: supported PIDs,	C2	xx	DATA_A
#n-2	Data B: supported PIDs,	C2	xx	DATA_B
#n-1	Data C: supported PIDs,	C2	xx	DATA_C
#n	Data D: supported PIDs]	C2	xx	DATA_D
C1 = Conditional — PID value shall be the same value as included in the request message if supported by the ECU C2 = Conditional — value indicates PIDs supported; range of supported PIDs depends on selected PID value (see C1)				

NOTE The response message shall only include the PID(s) and Data A - D which are supported by the ECU. If the request message includes (a) PID value(s) which are not supported by the ECU those shall not be included in the response message.

54.16.2.3 Read PID values

Request current powertrain diagnostic data request message

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request current powertrain diagnostic data request SID**	**M**	**01**	**SIDRQ**
#2	PID#1 (see Annex B of ISO 15031-5.4)	M	xx	PID
#3	PID#2 (see Annex B of ISO 15031-5.4)	U	xx	PID
#4	PID#3 (see Annex B of ISO 15031-5.4)	U	xx	PID
#5	PID#4 (see Annex B of ISO 15031-5.4)	U	xx	PID
#6	PID#5 (see Annex B of ISO 15031-5.4)	U	xx	PID
#7	PID#6 (see Annex B of ISO 15031-5.4)	U	xx	PID
U = User Optional — the parameter may be present or not				

54.16.2.4 Report PID values

Request current powertrain diagnostic data response message

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request current powertrain diagnostic data response SID**	**M**	**41**	**SIDPR**
 #2 #3 #4 #5 #6	data record of 1st supported PID = [PID#1 data A, data B, data C, data D]	 M M C1 C1 C1	 xx xx xx xx xx	PIDREC_ PID DATA_A DATA_B DATA_C DATA_D
:	:	:	:	:
 #n-4 #n-3 #n-2 #n-1 #n	data record of mth supported PID = [PID#m data A, data B, data C, data D]	 C2 C2 C3 C3 C3	 xx xx xx xx xx	PIDREC_ PID DATA_A DATA_B DATA_C DATA_D
C1 = Conditional — "data B - D" depend on selected PID value C2 = Conditional — parameter is only present if supported by the ECU C3 = Conditional — parameters and values for "data B - D" depend on selected PID number and are only included if PID is supported by the ECU				

NOTE Not all PIDs, which are included in the request message may be supported by all emission-related ECUs, which shall comply with this specification. Therefore, each vehicle ECU, which supports at least one (1) PID, shall send a response message including the PID(s) with data.

54.16.2.5 Parameter definition

54.16.2.5.1 PIDs supported

"Annex A of ISO 15031-5.4" specifies the interpretation of the data record of supported PIDs.

54.16.2.5.2 PID and data byte descriptions

"Annex B of ISO 15031-5.4" specifies standardised emission-related parameters.

54.17 Service 01h - Request current powertrain diagnostic data (Application Incident)

Import action:

ACTION_ERRM_ReadReadinessCode **(** **INOUT <ReadinessCode>** **OUT <ResultReadinessCode>** **)**
This API allows to read the readiness code information
ACTION_ERRM_ReadQuantityOfDtc **(** **IN <TypeOfDtc>** **IN <LevelOfDtc>** **INOUT <Quantity>** **OUT <ResultQuantity>** **)**
This API returns the quantity of DTC with a certain type, which are stored in memory

Input data:

T_ACT_MIL	STATE_PSN_LS_SAE	OBD_TPS_REL	STATE_OBD_SA
C_CLC_READY_OBD_1	LOAD_ABSV	STATE_LS[NC_CBK_EX_NR]	LOAD_CLC
TCO_MES	FAC_LAM_LIM[NC_CBK_EX_NR]	OBD_LAM_AD[NC_CBK_EX_NR]	FUP
MAP_SAE	N	VS	IGA_IGC[NC_CYL_NR]
TIA_MES[NC_SENS_NR_TIA]	MAF_KGH_MES[NC_MAF_NR]	OBD_PV_2	OBD_PV_1
T_DTC_CLR	VLS_DOWN[NC_CBK_EX_NR]	C_OBD_REQ	T_AST_SAE
DIST_ACT_MIL	LAMB_LS_UP[NC_CBK_EX_NR]	VLS_COR_LSL[NC_CBK_EX_NR]	CPPWM_CPS
OBD_FTL	STATE_MIL	CTR_WUP_DTC_CLR	DIST_DTC_CLR
AMP_MES	IPLSL_COR[NC_CBK_EX_NR]	TEG_CAT_UP_MDL[NC_CBK_EX_NR]	TEG_CAT_DOWN_MDL[NC_CBK_EX_NR]
STATE_ENA_OBD	STATE_CMPL_OBD	VB	OBD_TPS_2
LAMB_SP_HOM[NC_CBK_EX_NR]	OBD_TPS_SP	OBD_TAM	OBD_TPS_1
LAMB_DELTA_AD_LAM_ADJ[NC_CBK_EX_NR]	LAMB_DELTA_LAM_ADJ[NC_CBK_EX_NR]	STATE_VAR_SAP_SAE	OPG_SP_ACR
OBD_EGR_DIF	FUP_RNG_H_MES	NC_INJ_CONF	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_STATE_SRV_1_PID	12	0...FFH	0...255	1	[-]
LDPM_STATE_PID_SRV_1_2_OBDC	12	0...BH	0...11	1	[-]
Table of bytes which defines the PID's supported by the service 01h (01h - 60h)					

General information:

Depending on calibration of ID_STATE_SRV_1_PID the PID's are supported or not.

Each LDPM[0...11] of the table represents an 8 PIDs configuration:

0 => the PID is not supported
1 => the PID is supported

54.17.1 Description of PID 00h - Defines Supported PIDs 01h - 20h

DATA A - Supported PIDs 01h to 08h

PID 00h – DATA A - Supported PID 01h to 08h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[0]
0	PID 08h : Short term fuel trim - Bank 2	1
1	PID 07h : Long term fuel trim - Bank 1	1
2	PID 06h : Short term fuel trim - Bank 1	1
3	PID 05h : Engine coolant temperature	1
4	PID 04h : Calculated load value	1
5	PID 03h : Fuel system status	1
6	PID 02h : Trouble code that caused required freeze frame	1
7	PID 01h : DTCs and supported tests	1

DATA B - Supported PIDs 09h to 10h

PID 00h – DATA B - Supported PID 09h to 10h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[1]
0	PID 10h : Air flow rate from MAF sensor	1
1	PID 0Fh : Intake air temperature	1
2	PID 0Eh : Ignition time advance	1
3	PID 0Dh : Vehicle speed	1
4	PID 0Ch : Engine RPM	1
5	PID 0Bh : Intake manifold absolute pressure	1
6	PID 0Ah : Fuel pressure (gauge)	1
7	PID 09h : Long term fuel trim - Bank 2	1

DATA C - Supported PIDs 11h to 18h

PID 00h – DATA C - Supported PID 11h to 18h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[2]
0	PID 18h : Bank 2 Sensor 1 or Bank 3 Sensor 1	0
1	PID 17h : Bank 1 Sensor 4 or Bank 2 Sensor 2	0
2	PID 16h : Bank 1 Sensor 3 or Bank 2 Sensor 1	0
3	PID 15h : Bank 1 Sensor 2	1
4	PID 14h : Bank 1 Sensor 1	0
5	PID 13h : Location of oxygen sensors	1
6	PID 12h : Secondary air status	1
7	PID 11h : Absolute throttle position sensor	1

DATA D - Supported PIDs 19h to 20h

PID 00h - DATA D - Supported PID 19h to 20h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[3]
0	PID 20h : PIDs supported ($21 - $40)	1
1	PID 1Fh : Time Since Engine Start	1
2	PID 1Eh : Auxiliary Input Status	0
3	PID 1Dh : Location of oxygen sensors	0
4	PID 1Ch : OBD requirements	1
5	PID 1Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2	0
6	PID 1Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1	0
7	PID 19h : Bank 2 Sensor 2 or Bank 3 Sensor 2	1

54.17.2 Description of PID 01h - Monitor status since DTCs cleared

PID $01 definition

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
01	Monitor status since DTCs cleared			
	The bits in this PID shall report two pieces of information for each monitor: (a) Monitor status since DTCs were last cleared, saved in NVRAM or KAM. (b) Monitors supported on this vehicle.			
	Number of emission-related DTCs and MIL status	A (bit)	byte 1 of 4	DTC and MIL status:
	# of DTCs stored in this ECU	0-6	hex to decimal	DTC_CNT: xxxd
	Malfunction Indicator Light (MIL) Status	7	0 = MIL OFF; 1 = MIL ON	MIL: OFF or ON
	The MIL status shall indicate "OFF" during the key on, engine off bulb check unless the MIL has also been commanded "ON" for a detected malfunction.			
	Supported tests which are continuous	B (bit)	byte 2 of 4 (Low Nibble)	Support status of continuous monitors:
	Misfire monitoring	0	0 = monitor not supported (NO) 1 = monitor supported (YES)	MIS_SUP: NO or YES
	Misfire monitoring shall be supported on both, spark ignition and compression vehicles if the vehicle utilises a misfire monitor.			
	Fuel system monitoring	1	0 = monitor not supported (NO) 1 = monitor supported (YES)	FUEL_SUP: NO or YES
	Fuel system monitoring shall be supported on vehicles that utilise oxygen sensors for closed loop fuel feedback control, and utilise a fuel system monitor, typically spark ignition engines.			
	Comprehensive component monitoring	2	0 = monitor not supported (NO) 1 = monitor supported (YES)	CCM_SUP: NO or YES
	Comprehensive component monitoring shall be supported on spark ignition and compression ignition vehicles that utilise comprehensive component monitoring.			
	reserved (bit shall be reported as ‘0’)	3		---

PID $01 definition (continued)

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
	Status of continuous monitoring tests since DTC cleared:	B (bit)	byte 2 of 4 (High Nibble)	Completion status of continuous monitors since DTC cleared:
	Misfire monitoring	4	0 = monitor complete, or not applicable (YES) 1 = monitor not complete (NO)	MIS_RDY: YES or NO
	Misfire monitoring shall always indicate complete for spark ignition engines. Misfire monitoring shall indicate complete for compression ignition engines after the misfire evaluation is complete.			
	Fuel system monitoring	5	0 = monitor complete, or not applicable (YES) 1 = monitor not complete (NO)	FUEL_RDY: YES or NO
	Fuel system monitoring shall always indicate complete for both spark ignition and compression ignition engines.			
	Comprehensive component monitoring	6	0 = monitor complete, or not applicable (YES) 1 = monitor not complete (NO)	CCM_RDY: YES or NO
	Comprehensive component monitoring shall always indicate complete on both spark ignition and compression ignition engines. NOTE It can be assumed that by the time any non-continuous monitors are complete, continuous comprehensive component monitoring will also be complete. Bit 6 is allowed to always indicate "complete" on spark ignition vehicles that support other non-continuous monitors.			
	reserved (bit shall be reported as ‘0’)	7		---
	Supported tests run at least once per trip	C (bit)	byte 3 of 4	Support status of non-continuous monitors:
	Catalyst monitoring	0	0 = monitor not supported (NO) 1 = monitor supported (YES)	CAT_SUP: NO or YES
	Heated catalyst monitoring	1		HCAT_SUP: NO or YES
	Evaporative system monitoring	2		EVAP_SUP: NO or YES
	Secondary air system monitoring	3		AIR_SUP: NO or YES
	A/C system refrigerant monitoring	4		ACRF_SUP: NO or YES
	Oxygen sensor monitoring	5		O2S_SUP: NO or YES
	Oxygen sensor heater monitoring	6		HTR_SUP: NO or YES
	EGR system monitoring	7		EGR_SUP: NO or YES
	Status of tests run at least once per trip	D (bit)	byte 4 of 4	Completion status of non-continuous monitors since DTCs cleared:
	Catalyst monitoring	0	0 = monitor complete, or not applicable (YES) 1 = monitor not complete (NO)	CAT_RDY: YES or NO
	Heated catalyst monitoring	1		HCAT_RDY: YES or NO
	Evaporative system monitoring	2		EVAP_RDY: YES or NO
	Secondary air system monitoring	3		AIR_RDY: YES or NO
	A/C system refrigerant monitoring	4		ACRF_RDY: YES or NO
	Oxygen sensor monitoring	5		O2S_RDY: YES or NO
	Oxygen sensor heater monitoring	6		HTR_RDY: YES or NO
	EGR system monitoring	7		EGR_RDY: YES or NO

Data Byte	Application data to use		Formating conversion used
data A	(bit 0 to 6) :	NR_DTC	-
	(bit 7) :	STATE_MIL	
data B	STATE_READY_OBD_1 *		
data C	C_STATE_READY_OBD_2		
data D	STATE_READY_OBD_2		

*) The bit 0 up to 3 of this byte is calibration value C_STATE_READY_OBD_1.
The bit 4 up to 6 of this byte indicate if the tests has been performed continuously since the reinitialization of the memorized information.

Formula section:

```
ACTION_ERRM_ReadQuantityOfDtc
                            (
                            IN <OBD & CONFIRMED>
                            IN <LAW>
                            INOUT <NR_DTC>
                            OUT <ResultQuantity>
                            )

ACTION_ERRM_ReadReadinessCode
                            (
                            INOUT < C_STATE_READY_OBD_2,
                                    STATE_READY_OBD_1,
                                     STATE_READY_OBD_2>
                            OUT <ResultReadinessCode>
                            )

IF      (ResultQuantity = DTC_PRESENT          OR
        ResultQuantity = NO_DTC_PRESENT)       AND
        ResultReadinessCode = POSITIVE_RESPONSE
THEN
        Positive response

ELSE
        No answer
ENDIF
```

These actions should be sent at each time we use the mode 01h PID01.
Refer to “Error management” module.

54.17.3 Description of PID 03h - Status of fuel system

```
IF          (ID_STATE_SRV_1_PID[2] & 20h)        // PID 13h is supported
THEN
   IF       (STATE_PSN_LS_SAE & 11h)
   THEN
```

PID $03 definition

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
03	Fuel system 1 status:	A (bit)	byte 1 of 2	FUELSYS1:
	(unused bits shall be reported as ‘0’; no	0	1 = Open loop - has not yet satisfied conditions to go closed loop	OL
	more than one bit at a time can be set to a ‘1’	1	1 = Closed loop - using oxygen sensor(s) as feedback for fuel control	CL
	of that bank)	2	1 = Open loop due to driving conditions (e.g. power enrichment, deceleration enleanment)	OL-Drive
		3	1 = Open loop - due to detected system fault	OL-Fault
		4	1 = Closed loop, but fault with at least one oxygen sensor - may be using single oxygen sensor for fuel control	CL-Fault
		5-7	reserved (bits shall be reported as ‘0’)	---
	NOTE Fuel systems do not refer to injector banks. Fuel systems are intended to represent completely different fuel systems that can independently enter and exit closed loop fuel. Banks of injectors on a V-engine are generally not independent and share the same closed-loop enablement criteria.			
	Fuel system 2 status:	B (bit)	byte 2 of 2	FUELSYS2:
	(unused bits shall be reported as ‘0’; no	0	1 = Open loop - has not yet satisfied conditions to go closed loop	OL
	more than one bit at a time can be set to a ‘1’	1	1 = Closed loop - using oxygen sensor(s) as feedback for fuel control	CL
	of that bank)	2	1 = Open loop due to driving conditions (e.g. power enrichment, deceleration enleanment)	OL-Drive
		3	1 = Open loop - due to detected system fault	OL-Fault
		4	1 = Closed loop, but fault with at least one oxygen sensor - may be using single oxygen sensor for fuel control	CL-Fault
		5-7	reserved (bits shall be reported as ‘0’)	---

Data Byte	Application data to use	Formating conversion used
data A	STATE_LS[1]	no
data B	STATE_LS[2]	no

ENDIF

54.17.4 Description of PID 04h - Calculated load value

PID $04 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
04	Calculated LOAD Value!	A	0 %	100 %	100/255 %	LOAD_PCT: xxx.x %
	The OBD regulations previously defined CLV as: (current airflow / peak airflow @ sea level) * (BARO @ sea level / BARO) * 100% Various manufacturers have implemented this calculation in a variety of ways. The following definition, although a little more restrictive, will standardise and improve the accuracy the calculation. LOAD_PCT = [current airflow] / [(peak airflow at WOT@STP as a function of rpm) * (BARO/29.92) * SQRT(298/(AAT+273))] — Where: STP = Standard Temperature and Pressure = 25 °C, 29.92 in Hg BARO, SQRT = square root, — WOT = wide open throttle, AAT=Ambient Air Temperature and is in °C Characteristics of LOAD_PCT are: — Reaches 1.0 at WOT at any altitude, temperature or rpm for both naturally aspirated and boosted engines. — Indicates percent of peak available torque. — Linearly correlated with engine vacuum — Often used to schedule power enrichment. — Compression ignition engines (diesels) shall support this PID using fuel flow in place of airflow for the above calculations. NOTE Both spark ignition and compression ignition engines shall support PID $04. See PID $43 for an additional definition of engine LOAD.					

Data Byte	Application data to use	Formating conversion used
data A	LOAD_CLC	no

54.17.5 Description of PID 05h - Engine coolant temperature

PID $05 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
05	Engine Coolant Temperature	A	-40 °C	+215 °C	1 °C with -40 °C offset	ECT: xxx °C (xxx °F)
	ECT shall display engine coolant temperature derived from an engine coolant temperature sensor or a cylinder head temperature sensor. Many diesels do not use either sensor and may substitute Engine Oil Temperature instead.					

Data Byte	Application data to use	Formating conversion used
data A	TCO_MES	Shift to different range

54.17.6 Description of PID 06h - Short term fuel trim - Bank 1 and Bank 3

PID $06 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
06	Short Term Fuel Trim - Bank 1 (use if only 1 fuel trim value)	A	-100 % (lean)	+99.22 % (rich)	100/128 % (0 % at 128)	SHRTFT1: xxx.x %
	Short Term Fuel Trim - Bank 3	B				SHRTFT3: xxx.x %
	Short Term Fuel Trim Bank 1/3 shall indicate the correction being utilised by the closed loop fuel algorithm. If the fuel system is in open loop, SHRTFT1/3 shall report 0% correction. NOTE Data B shall only be included in the response message of a PID $06 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 3 supported) of PID $1D if Data B of PID $06 is supported or not.					

```
IF   (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
THEN
   IF       (STATE_PSN_LS_SAE & 11h)
   THEN
```

Data Byte	Application data to use	Formating conversion used
data A	FAC_LAM_LIM[1]	Shift to different range

```
     ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	

```
     ENDIF
ENDIF
```

54.17.7 Description of PID 07h - Long term fuel trim - Bank 1 and Bank 3

PID $07 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
07	Long Term Fuel Trim - Bank 1 (use if only 1 fuel trim value)	A	-100 % (lean)	+99.22 % (rich)	100/128 % (0 % at 128)	LONGFT1: xxx.x %
	Long Term Fuel Trim - Bank 3	B				LONGFT3: xxx.x %
	Fuel trim correction for Bank 1/3 stored in Non-volatile RAM or Keep-alive RAM. LONGFT shall indicate the correction being utilised by the fuel control algorithm at the time the data is requested, in both open loop and closed loop fuel control. If no correction is utilised in open loop fuel, LONGFT shall report 0% correction. If long-term fuel trim is not utilised at all by the fuel control algorithm, the PID shall not be supported. NOTE Data B shall only be included in the response message of a PID $07 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 3 supported) of PID $1D if Data B of PID $07 is supported or not.					

```
IF   (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
THEN
     IF       (STATE_PSN_LS_SAE & 33h)
```

Data Byte	Application data to use	Formating conversion used
data A	OBD_LAM_AD[1]	-

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	

```
    ENDIF
ENDIF
```

54.17.8 Description of PID 08h - Short term fuel trim - Bank 2 and Bank 4

PID $08 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
08	Short Term Fuel Trim - Bank 2 (use if only 1 fuel trim value)	A	-100 % (lean)	+99.22 % (rich)	100/128 % (0 % at 128)	SHRTFT2: xxx.x %
	Short Term Fuel Trim - Bank 4	B				SHRTFT4: xxx.x %
	Short Term Fuel Trim Bank 2/4 shall indicate the correction being utilized by the closed loop fuel algorithm. If the fuel system is in open loop, SHRTFT24 shall report 0% correction. NOTE Data B shall only be included in the response message of a PID $08 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 4 supported) of PID $1D if Data B of PID $08 is supported or not.					

```
IF   (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
THEN
    IF      (STATE_PSN_LS_SAE & 11h)
    THEN
```

Data Byte	Application data to use	Formating conversion used
data A	FAC_LAM_LIM[2]	Shift to different range

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	

```
    ENDIF
ENDIF
```

54.17.9 Description of PID 09h - Long term fuel trim - Bank 2 and Bank 4

PID $09 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
09	Long Term Fuel Trim – Bank 2 (use if only 1 fuel trim value)	A	-100 % (lean)	+99.22 % (rich)	100/128 % (0 % at 128)	LONGFT2: xxx.x %
	Long Term Fuel Trim - Bank 4	B				LONGFT4: xxx.x %
	Fuel trim correction for Bank 2/4 stored in Non-volatile RAM or Keep-alive RAM. LONGFT shall indicate the correction being utilised by the fuel control algorithm at the time the data is requested, in both open loop and closed loop fuel control. If no correction is utilised in open loop fuel, LONGFT shall report 0% correction. If long-term fuel trim is not utilised at all by the fuel control algorithm, the PID shall not be supported. NOTE Data B shall only be included in the response message of a PID $09 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 4 supported) of PID $1D if Data B of PID $09 is supported or not.					

```
IF        (ID_STATE_SRV_1_PID[2] & 20h)        // PID 13h is supported
THEN
    IF    (STATE_PSN_LS_SAE & 33h)
    THEN
```

Data Byte	Application data to use	Formating conversion used
data A	OBD_LAM_AD[2]	-

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	

```
    ENDIF
ENDIF
```

54.17.10 Description of PID 0Ah - Fuel pressure (gauge)

PID $0A definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0A	Fuel Rail Pressure (gauge)	A	0 kPa (gauge)	765 kPa (gauge)	3 kPa per bit (gauge)	FRP: xxx kPa (xx.x psi)
	FRP shall display fuel rail pressure at the engine when the reading is referenced to atmosphere (gauge pressure).					

Data Byte	Application data to use	Formating conversion used
data A	FUP	Shift to different range

54.17.11 Description of PID 0Bh - Intake manifold absolute pressure

PID $0B definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0B	Intake Manifold Absolute Pressure	A	0 kPa (absolute)	255 kPa (absolute)	1 kPa per bit (absolute)	MAP: xxx kPa (xx.x inHg)
	MAP shall display manifold pressure derived from a Manifold Absolute Pressure sensor, if a sensor is utilised. If a vehicle uses both a MAP and MAF sensor, both the MAP and MAF PIDs shall be supported.					

Data Byte	Application data to use	Formating conversion used
data A	MAP_SAE	None

54.17.12 Description of PID 0Ch - Engine speed

PID $0C definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0C	Engine RPM	A , B	0 min^{-1}	16383.75 min^{-1}	¼ rpm per bit	RPM: xxxxx min^{-1}

Data Byte	Application data to use	Formating conversion used
data A	MSB (N)	Shift to different range
data B	LSB (N)	Shift to different range

54.17.13 Description of PID 0Dh - Vehicle speed

PID $0D definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0D	Vehicle Speed Sensor	A	0 km/h	255 km/h	1 km/h per bit	VSS: xxx km/h (xxx mph)
	VSS shall display vehicle road speed, if utilised by the control module strategy. Vehicle speed may be derived from a vehicle speed sensor, calculated by the PCM using other speed sensors, or obtained from the vehicle serial data communication bus.					

Data Byte	Application data to use	Formating conversion used
data A	VS	no

54.17.14 Description of PID 0Eh - Ignition timing advance

PID $0E definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0E	Ignition Timing Advance for #1 Cylinder	A	-64 °	63.5 °	½ ° with 0 ° at 128	SPARKADV: xx °
	Ignition timing spark advance for #1 cylinder (not including mechanical advance)					

Data Byte	Application data to use	Formating conversion used
data A	IGA_IGC[0]	Shift to different range

54.17.15 Description of PID 0Fh - Intake air temperature

PID $0B definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
0F	Intake Air Temperature	A	-40 °C	+215 °C	1 °C with -40 °C offset	IAT: xxx °C (xxx °F)
	IAT shall display intake manifold air temperature, if utilised by the control module strategy. IAT may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	TIA_MES	Shift to different range

54.17.16 Description of PID 10h - Air flow rate from MAF

PID $0B definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
10	Air Flow Rate from Mass Air Flow Sensor	A , B	0 g/s	655.35 g/s	0.01 g/s	MAF: xxx.xx g/s (xxxx.x lb/min)
	MAF shall display the airflow rate as measured by the MAF sensor, if a sensor is utilised.					

Data Byte	Application data to use	Formating conversion used
data A	MSB (MAF_KGH_MES)	Kg/h into g/s
data B	LSB (MAF_KGH_MES)	Kg/h into g/s

54.17.17 Description of PID 11h - Absolute throttle position sensor

PID $0A - $11 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
11	Absolute Throttle Position	A	0 %	100 %	100/255 %	TP: xxx.x %
	Absolute throttle position (not “relative” or “learned” throttle position) shall be displayed as a normalised value, scaled from 0 to 100%. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed throttle position is a 1.0 volts, TP shall display (1.0 / 5.0) = 20% at closed throttle and 50% at 2.5 volts. Throttle position at idle will usually indicate greater than 0%, and throttle position at wide open throttle will usually indicate less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage. NOTE See PID $45 for a definition of Relative Throttle Position.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TPS_1	no

54.17.18 Description of PID 12h - Commanded secondary air status

With autolearning the value for the STATE_VAR_SAP_SAE is defined to the value defined by the project.

IF (ID_STATE_SRV_1_PID[2] & 40h) **AND** (STATE_VAR_SAP_SAE <> 0)

THEN

PID $12 definition

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
12	Commanded Secondary Air Status	A (bit)	byte 1 of 1	AIR_STAT:
	(if supported, one, and only	0	1 = upstream of first catalytic converter	AIR_STAT: UPS
	one bit at a time can be set	1	1 = downstream of first catalytic converter inlet	AIR_STAT: DNS
	to a 1)	2	1 = atmosphere / off	AIR_STAT: OFF
		3-7	reserved (bits shall be reported as ‘0’)	---

Data Byte	Calibration data to use	Formating conversion used
data A	STATE_OBD_SA	-

ENDIF

A.1.1

54.17.19 Description of PID 13h - Location of oxygen sensors

PID $13 definition

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
13	Location of Oxygen Sensors	A (bit)	byte 1 of 4	O2SLOC:
	(where sensor 1 is closest to the engine. Each bit indicates the presence or absence of an oxygen sensor at the following location)	0	1 = Bank 1 - Sensor 1 present at that location	O2S11
		1	1 = Bank 1 - Sensor 2 present at that location	O2S12
		2	1 = Bank 1 - Sensor 3 present at that location	O2S13
		3	1 = Bank 1 - Sensor 4 present at that location	O2S14
		4	1 = Bank 2 - Sensor 1 present at that location	O2S21
		5	1 = Bank 2 - Sensor 2 present at that location	O2S22
		6	1 = Bank 2 - Sensor 3 present at that location	O2S23
		7	1 = Bank 2 - Sensor 4 present at that location	O2S24

```
IF      (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
THEN    STATE_PSN_LS_SAE is defined to the value defined by the project
ELSE    STATE_PSN_LS_SAE = 0
ENDIF
```

Data Byte	Calibration data to use	Formating conversion used
data A	STATE_PSN_LS_SAE	no

54.17.20 Description of PID 14h - Bank 1 Sensor 1 (not supported)

54.17.21 Description of PID 15h - Bank 1 Sensor 2

PID $15 definition if PID $13 is supported!

PID (hex)	Description Use if PID $13 or PID $1D are supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
15	Oxygen Sensor Output Voltage Bank 1 – Sensor 2	A	0 V	1.275 V	0.005 V	O2Sxy: x.xxx V
	Short Term Fuel Trim (B1-S2) (associated with this sensor $FF if this sensor is not used in the calculation)	B	-100.00 % (lean)	99.22 % (rich)	100/128 % (0 % at 128)	SHRTFTxy: xxx.x %
	These PIDs shall be used for a conventional, 0 to 1 Volt oxygen sensor. Any sensor with a different full scale value shall be normalised to provide nominal full scale at $C8 (200 decimal). Wide-range/linear oxygen sensors shall use PID $25 or PID $35.					

```
IF  (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
THEN
    IF      (STATE_PSN_LS_SAE & 33h)
```

THEN

Data Byte	Application data to use	Formating conversion used
data A	VLS_DOWN[1]	-
data B	0xFF	

ELSE

Data Byte	Application data to use	Formating conversion used
data A	NONE	-
data B	NONE	

ENDIF

ENDIF

54.17.22 Description of PID 16h - Bank 1 Sensor 3 or Bank 2 Sensor 1 (not supported)

54.17.23 Description of PID 17h - Bank 1 Sensor 4 or Bank 2 Sensor 2 (not supported)

54.17.24 Description of PID 18h - Bank 2 Sensor 1 or Bank 3 Sensor 1 (not supported)

54.17.25 Description of PID 19h - Bank 2 Sensor 2 or Bank 3 Sensor 2

PID $19 definition if PID $13 is supported!

PID (hex)	Description Use if PID $13 is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
19	Oxygen Sensor Output Voltage Bank 2 – Sensor 2	A	0 V	1.275 V	0.005 V	O2Sxy: x.xxx V
	Short Term Fuel Trim (B2-S2) (associated with this sensor $FF if this sensor is not used in the calculation)	B	-100.00 % (lean)	99.22 % (rich)	100/128 % (0 % at 128)	SHRTFTxy: xxx.x %
	These PIDs shall be used for a conventional, 0 to 1 Volt oxygen sensor. Any sensor with a different full scale value shall be normalised to provide nominal full scale at $C8 (200 decimal). Wide-range/linear oxygen sensors shall use PID $29 or PID $39.					

IF (ID_STATE_SRV_1_PID[2] & 20h) *// PID 13h is supported*

THEN

IF (STATE_PSN_LS_SAE & 33h)

THEN

Data Byte	Application data to use	Formating conversion used
Data A	VLS_DOWN[2]	-
Data B	0xFF	

ELSE

Data Byte	Application data to use	Formating conversion used
Data A	NONE	-
Data B	NONE	

ENDIF

ENDIF

54.17.26 Description of PID 1Ah - Bank 2 Sensor 3 or Bank 4 Sensor 1 (not supported)

54.17.27 Description of PID 1Bh - Bank 2 Sensor 4 or Bank 4 Sensor 2 (not supported)

54.17.28 Description of PID 1Ch - OBD requirements

PID $1C definition

PID (hex)	Description	Data byte	Scaling	External test equipment SI (Metric) / English display
1C	OBD requirements to which vehicle is designed	A (hex)	byte 1 of 1 (State Encoded Variable)	OBDSUP:
		01	OBD II (California ARB)	OBD II
		02	OBD (Federal EPA)	OBD
		03	OBD and OBD II	OBD and OBD II
		04	OBD I	OBD I
		05	Not OBD compliant	NO OBD
		06	EOBD	EOBD
		07	EOBD and OBD II	EOBD and OBD II
		08	EOBD and OBD	EOBD and OBD
		09	EOBD, OBD and OBD II	EOBD, OBD and OBD II
		0A - FF	reserved by document	---

Data Byte	Calibration data to use	Formating conversion used
data A	C_OBD_REQ	-

54.17.29 Description of PID 1Dh - Location of oxygen sensors (not supported)

54.17.30 Description of PID 1Eh - Auxiliary input status (not supported)

54.17.31 Description of PID 1Fh - Time Since Engine Start

PID $1F definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
1F	Time Since Engine Start	A , B	0 sec.	65,535 sec.	1 second per count	RUNTM: xxxxx sec.
	RUNTM shall increment while the engine is running. It shall freeze if the engine stalls. RUNTM shall be reset to zero during every control module power-up and when entering the key-on, engine off position. RUNTM is limited to 65,535 seconds and shall not wrap around to zero.					

Data Byte	Application data to use	Formating conversion used
data A	MSB (T_AST_SAE)	-
data B	LSB (T_AST_SAE)	-

54.17.32 Description of PID 20h - Supported PIDs 21h - 40h

0 : the PID is not used
1 : the PID is used

DATA A - Supported PIDs 21h - 28h

PID 20h - DATA A - Supported PID 21h to 28h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[4]
0	PID 28h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	1
1	PID 27h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	0
2	PID 26h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	0
3	PID 25h : Bank 1 Sensor 2 (Wide range O2S)	0
4	PID 24h : Bank 1 Sensor 1 (Wide range O2S)	1
5	PID 23h : Fuel pressure high-range (gauge)	1
6	PID 22h : Relative fuel pressure	0
7	PID 21h : Distance travelled while MIL is activated	1

DATA B - Supported PIDs 29h to 30h

PID 20h - DATA B - Supported PID 29h to 30h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[5]
0	PID 30h : Number of warm-ups since diagnostic trouble codes cleared	1
1	PID 2Fh : Fuel Level Input	1
2	PID 2Eh : Commanded Evaporative Purge	1
3	PID 2Dh : EGR Error	1
4	PID 2Ch : Command EGR	1
5	PID 2Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	0
6	PID 2Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	0
7	PID 29h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	0

DATA C - Supported PIDs 31h to 38h

PID 20h - DATA C - Supported PID 31h to 38h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[6]
0	PID 38h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	1
1	PID 37h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	0
2	PID 36h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	0
3	PID 35h : Bank 1 Sensor 2 (Wide range O2S)	0
4	PID 34h : Bank 1 Sensor 1 (Wide range O2S)	1
5	PID 33h : Barometric Pressure	1
6	PID 32h : Evap System Vapor Pressure	0
7	PID 31h : Distance since diagnostic trouble codes cleared	1

DATA D - Supported PIDs 39h to 40h

PID 20h - DATA D - Supported PID 39h to 40h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[7]
0	PID 40h : PIDs supported ($41 - $60)	1
1	PID 3Fh : Catalyst Temperature Bank 2, Sensor 2	1
2	PID 3Eh : Catalyst Temperature Bank 1, Sensor 2	1
3	PID 3Dh : Catalyst Temperature Bank 2, Sensor 1	1
4	PID 3Ch : Catalyst Temperature Bank 1, Sensor 1	1
5	PID 3Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	0
6	PID 3Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	0
7	PID 39h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	0

54.17.33 Description of PID 21h - Distance travelled while MIL is activated

PID $21 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/ bit	External test equipment SI (Metric) / English display
21	**Distance Travelled While MIL is Activated**	A , B	0 km	65535 km	1 km per count	MIL_DIST: xxxxx km (xxxxx miles)
	Conditions for “Distance travelled” counter: • reset to $0000 when MIL state changes from deactivated to activated by this ECU • accumulate counts in km if MIL is activated (ON) • do not change value while MIL is not activated (OFF) • reset to $0000 if diagnostic information is cleared either by service $04 or 40 warm-up cycles without MIL activated • do not wrap to $0000 if value is $FFFF					

Data Byte	Application data to use	Formating conversion used
data A	MSB (DIST_ACT_MIL)	-
data B	LSB (DIST_ACT_MIL)	

54.17.34 Description of PID 22h - Fuel Rail Pressure relative to manifold vacuum (not supported)

54.17.35 Description of PID 23h - Fuel Rail Pressure

IF NC_INF_CONF <> 0 *(only supported for MSD)*

THEN

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
23	Fuel rail pressure	A , B	0 kPa	655350 kPa	10 kPa per bit unsigned, 1 kPA = 0.1450377 PSI	FRP: xxxxxx kPa (xxxxx.x PSI
	FRP shall display fuel rail pressure at the engine when the reading is referenced to atmosphere (gauge pressure)					

Data Byte	Application data to use	Formatting conversion used
data A	MSB (FUP_RNG_H_MES)	Shift to different range
data B	LSB (FUP_RNG_H_MES)	Shift to different range

ENDIF

54.17.36 Description of PID 24h - Bank 1 Sensor 1 (Wide range)

PID $24 definition

PID (hex)	Description Use if PID $13 or PID $1D is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
24	Equivalence Ratio (lambda) (B1-S1) Bank 1 – Sensor 1 (wide range O2S)	A , B	0	1.999	0.0000305	EQ_RATxy: x.xxx
	Oxygen Sensor Voltage (B1-S1)	C , D	0 V	7.999 V	0.000122 V	O2Sxy: x.xxx V
	PID $24 shall be used for linear or wide-ratio Oxygen Sensors when equivalence ratio and voltage are displayed					

```
IF      (ID_STATE_SRV_1_PID[2] & 20h)        // PID 13h is supported
THEN
    IF  (STATE_PSN_LS_SAE & 11h)
    THEN
```

Data Byte	Application data to use	Formating conversion used
data A	MSB (LAMB_LS_UP[1])	Shift to different range
data B	LSB (LAMB_LS_UP[1])	=
data C	MSB (VLS_COR_LSL[1])	=
data D	LSB (VLS_COR_LSL[1])	=

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	=
data C	NONE	=
data D	NONE	=

```
    ENDIF
ENDIF
```

54.17.37 Description of PID 25h - Bank 1 Sensor 2 (Wide range) (not supported)

54.17.38 Description of PID 26h - Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range) (not supported)

54.17.39 Description of PID 27h - Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range) (not supported)

54.17.40 Description of PID 28h - Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range)

PID $28 definition if PID $13 is supported

PID (hex)	Description Use if PID $13 is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
28	Equivalence Ratio (lambda) Bank 2 – Sensor 1 (wide range O2S)	A , B	0	1.999	0.0000305	EQ_RATxy: x.xxx
	Oxygen Sensor Voltage (B2-S1)	C , D	0 V	7.999 V	0.000122 V	O2Sxy: x.xxx V
	PIDs $28 shall be used for linear or wide-ratio Oxygen Sensors when equivalence ratio and voltage are displayed					

```
IF        (ID_STATE_SRV_1_PID[2] & 20h)        // PID 13h is supported
THEN
   IF     (STATE_PSN_LS_SAE & 11h)
   THEN
```

Data Byte	Application data to use	Formating conversion used
data A	MSB (LAMB_LS_UP[2])	Shift to different range
data B	LSB (LAMB_LS_UP[2])	=
data C	MSB (VLS_COR_LSL[2])	=
data D	LSB (VLS_COR_LSL[2])	=

```
   ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	Shift to different range
data B	NONE	=
data C	NONE	=
data D	NONE	=

```
   ENDIF
ENDIF
```

54.17.41 Description of PID 29h - Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range) (not supported)

54.17.42 Description of PID 2Ah - Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range) (not supported)

54.17.43 Description of PID 2Bh - Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range) (not supported)

54.17.44 Description of PID 2C - Commanded EGR

IF NC_INJ_CONF <> 0 *(only supported for MSD)*

THEN

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
2C	Commanded EGR	A	0% (no flow)	100% (max. flow)	100/255 %	EGR_PCT: xxx.x%
	Commanded EGR displayed as a percent. EGR_PCT shall be normalised to the maximum EGR commanded output control parameter. EGR systems use a variety of methods to control the amount of EGR delivered to the engine. 1) If an on/off solenoid is used – EGR_PCT shall display 0% when the EGR is commanded off, 100% when the EGR system is commanded on. 2) If a vacuum solenoid is duty cycled, the EGR duty cycle from 0 to 100% shall be displayed. 3) If a linear or stepper motor valve is used, the fully closed position shall be displayed as 0%, the fully open position shall be displayed as 100%. Intermediate positions shall be displayed as a percent of the full-open position. For example, a stepper-motor EGR valve that moves from 0 to 128 counts shall display 0% at 0 counts, 100% at 128 counts and 50% at 64 counts. 4) Any other actuation method shall be normalised to display 0% when no EGR is commanded and 100% at the maximum commanded EGR position.					

Data Byte	Application data to use	Formating conversion used
data A	OPG_SP_ACR	Shift to different range

ENDIF

54.17.45 Description of PID 2D definition - EGR Error

IF NC_INJ_CONF <> 0 *(only supported for MSD)*

THEN

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
2D	EGR Error = (EGR actual – EGR commanded) / EGR commanded * 100%	A	-100 % (less than commanded)	+99.22 % (more than commanded)	100/128 % (0 % at 128)	EGR_ERR: xxx.x%
	EGR error, as a percent of commanded EGR. Often, EGR valve control outputs are not in the same engineering units as the EGR feedback input sensors. For example, an EGR valve can be controlled using a duty-cycled vacuum solenoid, however, the feedback input sensor is a position sensor. This makes it impossible to display "actual" versus "commanded" in the same engineering units. EGR error solved this problem by displaying a normalised (non-dimensional) EGR system feedback parameter. EGR error is defined to be: (actual EGR – commanded EGR) / commanded EGR. For example if 10% EGR is commanded and 5% is delivered to the engine, the EGR_ERR is (5% - 10%) / 10 = -50% error. EGR_ERR may be computed using various control parameters such as position, steps, counts, etc. All EGR systems must react to quickly changing conditions in the engine; therefore, EGR_ERR will generally show errors during transient conditions. Under steady condition, the error will be minimized (no necessarily zero, however) if the EGR system is under control. If the control system does not use closed loop control, EGR_ERR shall not be supported.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_EGR_DIF	-

ENDIF

54.17.46 Description of PID 2Eh - Commanded Evaporative Purge

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
2E	Commanded Evaporative Purge	A	0% no flow	100% max. flow	100/255 %	EVAP_PCT: xxx.x %
	Commanded evaporative purge control valve displayed as a percent. EVAP_PCT shall be normalised to the maximum EVAP purge commanded output control parameter. 1) If an on/off solenoid is used – EVAP_PCT shall display 0% when purge is commanded off, 100% when purge is commanded on. 2) If a vacuum solenoid is duty cycled, the EVAP purge valve duty cycle from 0 to 100% shall be displayed. 3) If a linear or stepper motor valve is used, the fully closed position shall be displayed as 0%, the fully open position shall be displayed as 100%. Intermediate positions shall be displayed as a percent of the full-open position. For example, a stepper-motor EVAP purge valve that moves from 0 to 128 counts shall display 0% at 0 counts, 100% at 128 counts and 50% at 64 counts. 4) Any other actuation method shall be normalised to display 0% when no purge is commanded and 100% at the maximum commanded purge position/flow.					

Data Byte	Application data to use	Formating conversion used
data A	CPPWM_CPS	-

54.17.47 Description of PID 2Fh - Fuel Level Input

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
2F	Fuel Level Input	A	0% no fuel	100% max. fuel capacity	100/255 %	FLI: xxx.x %
	FLI shall indicate nominal fuel tank liquid fill capacity as a percent of maximum, if utilised by the control module for OBD monitoring . FLI may be obtained directly from a sensor, may be obtained indirectly via the vehicle serial data communication bus, or may be inferred by the control strategy using other sensor inputs. Vehicles that use gaseous fuels shall display the percent of useable fuel capacity.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_FTL	no

54.17.48 Description of PID 30h - Number of warm-ups since diagnostic trouble codes cleared

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
30	Number of warm-ups since diagnostic trouble codes cleared	A	0	255	1 warm-up per count	WARM_UPS: xxx
	Number of OBD warm-up cycles since all DTCs were cleared (via an external test eqiupment or possibly, a battery disconnect). A warm-up is defined in the OBD regulations to be sufficient vehicle operation such that coolant temperature rises by at least 22 °C (40 °F) from engine starting and reaches a minimum temperature of 70 °C (160 deg °F) (60 °C (140 °F) for diesels). This PID is not associated with any particular DTC. It is simply an indication for I/M, of the last time an external test equipment was used to clear DTCs. If greater than 255 warm ups have occurred, WARM_UPS shall remain at 255 and not wrap to zero.					

Data Byte	Application data to use	Formating conversion used
data A	CTR_WUP_DTC_CLR	-

54.17.49 Description of PID 31h - Distance since diagnostic trouble codes cleared

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
31	Distance since diagnostic trouble codes cleared (same scaling as PID $21)	A , B	0 km	65,535 km	1 km per count	CLR_DIST: xxxxx km (xxxxx miles)
	Distance accumulated since DTCs were cleared (via an external test equipment or possibly, a battery disconnect). This PID is not associated with any particular DTC. It is simply an indication for I/M (Inspection/Maintenance), of the last time an external test equipment was used to clear DTCs. If greater than 65,535 km have occurred, CLR_DIST shall remain at 65,535 km and not wrap to zero.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(DIST_DTC_CLR)	-
data B	LSB(DIST_DTC_CLR)	

54.17.50 Description of PID 32h - Evap System Vapor Pressure (not supported)

54.17.51 Description of PID 33h - Barometric Pressure

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
33	Barometric Pressure	A	0 kPa (absolute)	255 kPa (absolute)	1 kPa per bit (absolute)	BARO: xxx kPa (xx.x inHg)
	Barometric pressure utilised by the control module. BARO is normally obtained from a dedicated BARO sensor, from a MAP sensor at key-on and during certain modes of driving, or inferred from a MAF sensor and other inputs during certain modes of driving. The control module shall report BARO from whatever source it is derived from. NOTE Some weather services report local BARO values adjusted to sea level. In these cases, the reported value may not match the displayed value on the external test equipment. NOTE If BARO is inferred while driving and stored in non-volatile RAM or Keep-alive RAM, BARO may not be accurate after a battery disconnect or total memory clear.					

Data Byte	Application data to use	Formating conversion used
data A	AMP_MES	Shift to different range

54.17.52 Description of PID 34h - Bank 1 Sensor 1 (Wide range)

PID $34 definition if PID $13 is supported

PID (hex)	Description Use if PID $13 or PID $1D is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
34	Equivalence Ratio (lambda) Bank 1 - Sensor 1 (wide range O2S)	A , B	0	1.999	0.0000305	EQ_RATxy: x.xxx
	Oxygen Sensor Current (B1-S1)	C , D	-128 mA	127.996 mA	0.00390625 mA ($8000 = 0 mA)	O2Sxy: x.xxx mA
	PID $34 shall be used for linear or wide-ratio Oxygen Sensors when equivalence ratio and current are displayed					

```
IF      (ID_STATE_SRV_1_PID[2] & 20h)      // PID 13h is supported
THEN
    IF      (STATE_PSN_LS_SAE & 11h)
    THEN
```

Data Byte	Application data to use	Formating conversion used
data A	LAMB_LS_UP[1]	-
data B	LAMB_LS_UP[1]	
data C	IPLSL_COR[1]	-
data D	IPLSL_COR[1]	

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	-
data B	NONE	
data C	NONE	-
data D	NONE	

```
    ENDIF
ENDIF
```

54.17.53 Description of PID 35h - Bank 1 Sensor 2 (Wide range) (not supported)

54.17.54 Description of PID 36h - Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range) (not supported)

54.17.55 Description of PID 37h - Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range) (not supported)

54.17.56 Description of PID 38h - Bank 2 Sensor 1 (Wide range)

PID $38 definition if PID $13 is supported

PID (hex)	Description Use if PID $13 is supported!	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
38	Equivalence Ratio (lambda) Bank 2 - Sensor 1 (wide range O2S)	A , B	0	1.999	0.0000305	EQ_RATxy: x.xxx
	Oxygen Sensor Current (B2-S1)	C , D	-128 mA	127.996 mA	0.00390625 mA ($8000 = 0 mA)	O2Sxy: x.xxx mA
	PIDs $38 shall be used for linear or wide-ratio Oxygen Sensors when equivalence ratio and current are displayed					

```
IF          (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
THEN
    IF      (STATE_PSN_LS_SAE & 11h)
    THEN
```

Data Byte	Application data to use	Formating conversion used
data A	LAMB_LS_UP[2]	-
data B	LAMB_LS_UP[2]	
data C	IPLSL_COR[2]	-
data D	IPLSL_COR[2]	

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	-
data B	NONE	
data C	NONE	-
data D	NONE	

ENDIF

ENDIF

54.17.57 Description of PID 39h - Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range) (not supported)

54.17.58 Description of PID 3Ah - Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range) (not supported)

54.17.59 Description of PID 3Bh - Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range) (not supported)

54.17.60 Description of PID 3C - Catalyst temperature Bank 1, Sensor 1

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
3C	Catalyst Temperature Bank 1, Sensor 1	A , B	-40 °C	+6513.5 °C	0.1 °C / bit with -40 °C offset	CATEMP11: xxxx.x °C (xxxx.x °F)
	CATEMP11 shall display catalyst substrate temperature for a bank 1 catalyst, if utilised by the control module strategy for OBD monitoring, or the Bank 1, Sensor 1 catalyst temperature sensor. CATEMP11 may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(TEG_CAT_UP_MDL[1])	Shift to different range
data B	LSB(TEG_CAT_UP_MDL[1])	=

54.17.61 Description of PID 3D - Catalyst temperature Bank 2, Sensor 1

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
3D	Catalyst Temperature Bank 2, Sensor 1	A , B	-40 °C	+6513.5 °C	0.1 °C / bit with -40 °C offset	CATEMP21: xxxx.x °C (xxxx.x °F)
	CATEMP21 shall display catalyst substrate temperature for a bank 2 catalyst, if utilised by the control module strategy for OBD monitoring, or the Bank 2, Sensor 1 catalyst temperature sensor. CATEMP21 may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(TEG_CAT_UP_MDL[2])	Shift to different range
data B	LSB(TEG_CAT_UP_MDL[2])	=

54.17.62 Description of PID 3E - Catalyst temperature Bank 1, Sensor 2

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
3E	Catalyst Temperature Bank 1, Sensor 2	A , B	-40 °C	+6513.5 °C	0.1 °C / bit with -40 °C offset	CATEMP12: xxxx.x °C (xxxx.x °F)
	CATEMP12 shall display catalyst substrate temperature for an additional bank 1 catalyst, if utilised by the control module strategy for OBD monitoring, or the Bank 1, Sensor 2 catalyst temperature sensor. CATEMP12 may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(TEG_CAT_DOWN_MDL[1])	Shift to different range
data B	LSB(TEG_CAT_DOWN_MDL[1])	=

54.17.63 Description of PID 3F - Catalyst temperature Bank 2, Sensor 2

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
3F	Catalyst Temperature Bank 2, Sensor 2	A , B	-40 °C	+6513.5 °C	0.1 °C / bit with -40 °C offset	CATEMP22: xxxx.x °C (xxxx.x °F)
	CATEMP22 shall display catalyst substrate temperature for an additional bank 2 catalyst, if utilised by the control module strategy for OBD monitoring, or the Bank 2, Sensor 2 catalyst temperature sensor. CATEMP22 may be obtained directly from a sensor, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	MSB(TEG_CAT_DOWN_MDL[2])	Shift to different range
data B	LSB(TEG_CAT_DOWN_MDL[2])	=

54.17.64 Description of PID 40h - Supported PIDs 41h - 60h

0 : the PID is not used
1 : the PID is used

DATA A - Supported PIDs 41h - 48h

PID 40h - DATA A - Supported PID 41h to 48h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[8]
0	PID 48h : Absolute Throttle Position C	0
1	PID 47h : Absolute Throttle Position B	1
2	PID 46h : Ambient air temperature	1
3	PID 45h : Relative Throttle Position	1
4	PID 44h : Command Equivalence Ratio	1
5	PID 43h : Absolue Load Value	1
6	PID 42h : Control module voltage	1
7	PID 41h : Monitor status this driving cycle	1

DATA A - Supported PIDs 49h - 50h

PID 40h - DATA B - Supported PID 49h to 50h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[9]
0	PID 50h : Max. value for Air Flow Rate	0
1	PID 4Fh : Max. value for Equivalence Ratio	0
2	PID 4Eh : Time since diagnostic trouble codes cleared	1
3	PID 4Dh : Minutes run by the engine while MIL activated	1
4	PID 4Ch : Commanded Throttle Actuator Control	1
5	PID 4Bh : Accelerator Pedal Position F	0
6	PID 4Ah : Accelerator Pedal Position E	1
7	PID 49h : Accelerator Pedal Position D	1

DATA C - Supported PIDs 51h to 58h

PID 40h - DATA C - Supported PID 51h to 58h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[10]
0	PID 58h : Long term fuel trim Bank 2	1
1	PID 57h : Short term fuel trim Bank 2	1
2	PID 56h : Long term fuel trim Bank 1	1
3	PID 55h : Short term fuel trim Bank 1	1
4	PID 54h : Evap System Vapor Pressure	0
5	PID 53h : Absolute Evap System Vapor Pressure	0
6	PID 52h : Alcohol Fuel Percentage	0
7	PID 51h : Type of fuel used by the vehicle	0

DATA D - Supported PIDs 59h to 60h

PID 40h - DATA D - Supported PID 59h to 60h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_1_PID[11]
0	PID 60h : Reserved	0
1	PID 5Fh : Reserved	0
2	PID 5Eh : Reserved	0
3	PID 5Dh : Reserved	0
4	PID 5Ch : Reserved	0
5	PID 5Bh : Reserved	0
6	PID 5Ah : Reserved	0
7	PID 59h : Reserved	0

54.17.65 Description of PID 41 - Monitor status this driving cycle

PID $41 definition

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
	The bit in this PID shall report two pieces of information for each monitor: 1) Monitor enable status for the current driving cycle. This bit shall indicate when a monitor is disabled in a manner such that there is no way for the driver to operate the vehicle for the remainder of the driving cycle and make the monitor run. Typical examples are: Engine-off soak not long enough (e.g., cold start temperature conditions not satisfied) Monitor maximum time limit or number of attempts/aborts exceeded Ambient air temperature too low or too high BARO too low (high altitude) The monitor shall not indicate "disabled" for operator-controlled conditions such as rpm, load, throttle position, minimum time limit not exceeded, ECT, TP, etc. 2) Monitor completion status for the current driving/monitoring cycle. Status shall be reset to "not complete" upon starting a new monitoring cycle. Note that some monitoring cycles can include various engine-operating conditions; other monitoring cycles begin after the ignition key is turned off. Some status bits on a given vehicle can utilise engine-running monitoring cycles while others can utilise engine-off monitoring cycles. Resetting the bits to "not complete" upon starting the engine will accommodate most engine-running and engine-off monitoring cycles, however, manufacturers are free to define their own monitoring cycles. NOTE PID $41 bits shall be utilised for all non-continuous monitors which are supported, and change completion status in PID $01. If a non-continuous monitor is not supported or always shows "complete", the corresponding PID $41 bits shall indicate disabled and complete. PID $41 bits may be utilised at the vehicle manufacturer's discretion for all continuous monitors which are supported with the exception of bit 03 which shall always showed CCM as enabled for spark ignition and compression ignition engines.			

PID $41 definition (continued)

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
		A (bit)	**byte 1 of 4**	
	Reserved - shall be reported as $00	0-7		---
	Enable status of continuous monitors this monitoring cycle:	**B (bit)**	**byte 2 of 4 (Low Nibble)**	**Enable status of continuous monitors this monitoring cycle: NO means disabled for rest of this monitoring cycle or not supported in PID $01, YES means enabled for this monitoring cycle.**
	Misfire monitoring	0	0 = monitor disabled for rest of this monitoring cycle or not supported (NO) 1 = monitor enabled for this monitoring cycle (YES)	MIS_ENA: NO or YES
	Fuel system monitoring	1		FUEL_ENA: NO or YES
	Comprehensive component monitoring	2		CCM_ENA: YES
	reserved (bit shall be reported as ‘0’)	3		

PID $41 definition (continued)

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
	Completion status of continuous monitors this monitoring cycle:	**B (bit)**	**byte 2 of 4 (High Nibble)**	**Completion status of continuous monitors this monitoring cycle:**
	Misfire monitoring C_CLC_READY_OBD_1[0]	4	See PID $01 to determine which monitors are supported 0 = monitor complete this monitoring cycle, or not supported (YES) 1 = monitor not complete this monitoring cycle (NO)	MIS_CMPL: YES or NO
	Fuel system monitoring C_CLC_READY_OBD_1[1]	5		FUELCMPL: YES or NO
	Comprehensive component monitoring C_CLC_READY_OBD_1[2]	6		CCM_CMPL: YES or NO
	reserved (bit shall be reported as ‘0’) C_CLC_READY_OBD_1[3]	7		

PID $41 definition (continued)

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
	Enable status of non-continuous monitors this monitoring cycle:	**C (bit)**	**byte 3 of 4**	**Enable status of non-continuous monitors this monitoring cycle:**
	Catalyst monitoring	0	0 = monitor disabled for rest of this monitoring cycle (NO) 1 = monitor enabled for this monitoring cycle (YES)	CAT_ENA: YES or NO
	Heated catalyst monitoring	1		HCAT_ENA: YES or NO
	Evaporative system monitoring	2		EVAP_ENA: YES or NO
	Secondary air system monitoring	3		AIR_ENA: YES or NO
	A/C system refrigerant monitoring	4		ACRF_ENA: YES or NO
	Oxygen sensor monitoring	5		O2S_ENA: YES or NO
	Oxygen sensor heater monitoring	6		HTR_ENA: YES or NO
	EGR system monitoring	7		EGR_ENA: YES or NO

PID $41 definition (continued)

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric) / English display
41	Monitor status this driving cycle			
	Completion status of non-continuous monitors this monitoring cycle:	**D (bit)**	**byte 4 of 4**	**Completion status of non-continuous monitors this monitoring cycle:**
	Catalyst monitoring	0	See PID $01 to determine which monitors are supported. 0 = monitor complete this monitoring cycle, or not supported (YES) 1 = monitor not complete this monitoring cycle (NO)	CAT_CMPL: YES or NO
	Heated catalyst monitoring	1		HCATCMPL: YES or NO
	Evaporative system monitoring	2		EVAPCMPL: YES or NO
	Secondary air system monitoring	3		AIR_CMPL: YES or NO
	A/C system refrigerant monitoring	4		ACRFCMPL: YES or NO
	Oxygen sensor monitoring	5		O2S_CMPL: YES or NO
	Oxygen sensor heater monitoring	6		HTR_CMPL: YES or NO
	EGR system monitoring	7		EGR_CMPL: YES or NO

Data Byte	Application data to use	Formating conversion used
data A	0x00	-
data B	C_CLC_READY_OBD_1[0...3] + STATE_ENA_OBD[0...3]	
data C	STATE_ENA_OBD[8..15]	
data D	STATE_CMPL_OBD[8..15]	

54.17.66 Description of PID 42 - Control module voltage

PID $42 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
42	Control module voltage	A , B	0 V	65.535 V	0.001 V per bit	VPWR: xx.xxx V
	VPWR – power input to the control module. VPWR is normally battery voltage, less any voltage drop in the circuit between the battery and the control module. NOTE 42-volt vehicles may utilise multiple voltages for different systems on the vehicle. VPWR represents the voltage at the control module; it may be significantly different than battery voltage.					

Data Byte	Application data to use	Formating conversion used
data A	MSB (VB)	Shift to different range
data B	LSB (VB)	

54.17.67 Description of PID 43 - Absolute Load Value

PID $43 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
43	Absolute Load Value	A , B	0 %	25700%	100/255 %	LOAD_ABS: xxx.x%
	The absolute load value has some different characteristics than the LOAD_PCT defined in PID 04 This definition, although restrictive, will standardise the calculation. LOAD_ABS is the normalised value of air mass per intake stroke displayed as a percent. LOAD_ABS = [air mass (g/intake stroke)] / [1.184 (g/intake stroke) * cylinder displacement in litres] Derivation: air mass (g / intake stroke) = [total engine air mass (g/sec)] / [rpm (revs/min)* (1 min / 60 sec) * (1/2 # of cylinders (strokes /rev)] LOAD_ABS = [air mass (g)/intake stroke] / [maximum air mass (g)/intake stroke at WOT@STP at 100% volumetric efficiency] * 100% Where: STP = Standard Temperature and Pressure = 25 °C, 29.92 in Hg (1013 hPa) BARO, WOT = wide open throttle The quantity (maximum air mass (g)/intake stoke at WOT@STP at 100% volumetric efficiency) is a constant for a given cylinder swept volume. The constant is 1.184 (g/litre 3) * cylinder displacement (litre 3/intake stroke) based on air density at STP Characteristics of LOAD_ABS are: Ranges from 0 to approximately 0.95 for naturally aspirated engines, 0 – 4 for boosted engines Linearly correlated with engine indicated and brake torque Often used to schedule spark and EGR rates Peak value of LOAD_ABS correlates with volumetric efficiency at WOT. Indicates the pumping efficiency of the engine for diagnostic purposes. Spark ignition engine are required to support PID $43. Compression ignition (diesel) engines are not required to support this PID. NOTE See PID $04 for an additional definition of engine LOAD..					

Data Byte	Application data to use	Formating conversion used
data A	MSB (LOAD_ABSV)	no
data B	LSB (LOAD_ABSV)	no

54.17.68 Description of PID 44 - Commanded Equivalence Ratio

Table .2 — PID $44 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
44	Commanded Equivalence Ratio	A , B	0	1.999	0.0000305	EQ_RAT: x.xxx
	Fuel systems that utilise conventional oxygen sensor shall display the commanded open loop equivalence ratio while the fuel control system is in open loop. EQ_RAT shall indicate 1.0 while in closed loop fuel. Fuel systems that utilise wide-range/linear oxygen sensors shall display the commanded equivalence ratio in both open loop and closed loop operation. To obtain the actual A/F ratio being commanded, multiply the stoichiometric A/F ratio by the equivalence ratio. For example, for gasoline, stoichiometric is 14.64:1 ratio. If the fuel control system was commanding an 0.95 EQ_RAT, the commanded A/F ratio to the engine would be 14.64 * 0.95 = 13.9 A/F					

Data Byte	Application data to use	Formating conversion used
data A	MSB(LAMB_SP_HOM[1])	Shift to different resolution =
data B	LSB(LAMB_SP_HOM[1])	

54.17.69 Description of PID 45 - Relative Throttle Position

PID $45 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
45	Relative Throttle Position	A	0 %	100 %	100/255 %	TP_R: xxx.x %
	Relative or "learned" throttle position shall be displayed as a normalised value, scaled from 0 to 100%. TP_R should display a value of 0% at the "learned closed-throttle position. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed throttle position is a 1.0 volts, TP shall display (1.0 – 1.0 / 5.0) = 0% at closed throttle and 30% at 2.5 volts. Because of the closed-throttle offset, wide open throttle will usually indicate substantially less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage. See PID $11 for a definition of Absolute Throttle Position.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TPS_REL	-

54.17.70 Description of PID 46 - Ambient air temperature

PID $46 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
46	Ambient air temperature (same scaling as IAT - $0F)	A	-40 °C	+215 °C	1 °C with -40 °C offset	AAT: xxx °C / xxx °F
	AAT shall display ambient air temperature, if utilised by the control module strategy for OBD monitoring. AAT may be obtained directly from a sensor, may be obtained indirectly via the vehicle serial data communication bus, or may be inferred by the control strategy using other sensor inputs.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TAM	-

54.17.71 Description of PID 47 - Absolute Throttle Position B

PID $47 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
47	Absolute Throttle Position B	A	0 %	100 %	100/255 %	TP_B: xxx.x %
	Absolute throttle position B, if utilised by the control module, (not “relative” or “learned” throttle position) shall be displayed as a normalised value, scaled from 0 to 100%. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed throttle position is a 1.0 volts, TP_B shall display (1.0 / 5.0) = 20% at closed throttle and 50% at 2.5 volts. Throttle position at idle will usually indicate greater than 0%, and throttle position at wide open throttle will usually indicate less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TPS_2	-

54.17.72 Description of PID 48 - Absolute Throttle Position C (not supported)

54.17.73 Description of PID 49 - Accelerator Pedal Position D

PID $49 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
49	Accelerator Pedal Position D	A	0 %	100 %	100/255 %	APP_D: xxx.x %
	Accelerator Pedal Position D, if utilised by the control module, (not “relative” or “learned” pedal position) shall be displayed as a normalised value, scaled from 0 to 100%. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed pedal is 1.0 volt, APP_D shall display (1.0 / 5.0) = 20% at closed pedal and 50% at 2.5 volts. Pedal position at idle will usually indicate greater than 0%, and pedal position at wide open pedal will usually indicate less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_PV_1	-

54.17.74 Description of PID 4A - Accelerator Pedal Position E

PID $4A definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
4A	Accelerator Pedal Position E	A	0 %	100 %	100/255 %	APP_E: xxx.x %
	Accelerator Pedal Position E, if utilised by the control module, (not “relative” or “learned” pedal position) shall be displayed as a normalised value, scaled from 0 to 100%. For example, if a 0 to 5.0 volt sensor is used (uses a 5.0 volt reference voltage), and the closed pedal is 1.0 volt, APP_E shall display (1.0 / 5.0) = 20% at closed pedal and 50% at 2.5 volts. Pedal position at idle will usually indicate greater than 0%, and pedal position at wide open pedal will usually indicate less than 100%. For systems where the output is proportional to the input voltage, this value is the percent of maximum input reference voltage. For systems where the output is inversely proportional to the input voltage, this value is 100% minus the percent of maximum input reference voltage.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_PV_2	-

54.17.75 Description of PID 4B - Accelerator Pedal Position F (not supported)

54.17.76 Description of PID 4C - Commanded Throttle Actuator Control

PID $4C definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
4C	Commanded Throttle Actuator Control	A	0% (closed throttle)	100% (wide open throttle)	100/255 %	TAC_PCT: xxx.x%
	Commanded TAC displayed as a percent. TAC_PCT shall be normalised to the maximum TAC commanded output control parameter. TAC systems use a variety of methods to control the amount of throttle opening. 1) If a linear or stepper motor is used, the fully closed throttle position shall be displayed as 0%, the fully open throttle position shall be displayed as 100%. Intermediate positions shall be displayed as a percent of the full-open throttle position. For example, a stepper-motor TAC that moves the throttle from 0 to 128 counts shall display 0% at 0 counts, 100% at 128 counts and 50% at 64 counts. 2) Any other actuation method shall be normalised to display 0% when the throttle is commanded closed and 100% when the throttle is commanded open.					

Data Byte	Application data to use	Formating conversion used
data A	OBD_TPS_SP	-

54.17.77 Description of PID 4D - Minutes run by the engine while MIL activated

PID $4D definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/ bit	External test equipment SI (Metric) / English display
4D	Minutes run by the engine while MIL activated	A, B	0 min	65535 min 1092.25 hours	1 min per count	MIL_TIME: xxxx hrs, xx min
	Conditions for “Minutes run by the engine while MIL activated” counter: • reset to $0000 when MIL state changes from deactivated to activated by this ECU • accumulate counts in minutes if MIL is activated (ON) • do not change value while MIL is not activated (OFF) • reset to $0000 if diagnostic information is cleared either by service $04 or 40 warm-up cycles without MIL activated • do not wrap to $0000 if value is $FFFF					

Data Byte	Application data to use	Formating conversion used
data A	MSB (T_ACT_MIL)	-
data B	LSB (T_ACT_MIL)	

54.17.78 Description of PID 4E - Time since diagnostic trouble codes cleared

PID $4E definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
4E	Time since diagnostic trouble codes cleared	A, B	0 min	65535 min 1092.25 hours	1 min per count	CLR_TIME: xxxx hrs, xx min
	Time accumulated since DTCs were cleared (via an external test equipment or possibly, a battery disconnect). This PID is not associated with any particular DTC. It is simply an indication for I/M (Inspection/Maintenance), of the last time an external test equipment was used to clear DTCs. If greater than 65,535 min have occurred, CLR_TIME shall remain at 65,535 min and not wrap to zero.					

Data Byte	Application data to use	Formating conversion used
data A	MSB (T_DTC_CLR)	-
data B	LSB (T_DTC_CLR)	

54.17.79 Description of PID 4F – Max. value for Equivalence Ratio (not supported)

54.17.80 Description of PID 50 – Max. value for Air Flow Rate (not supported)

54.17.81 Description of PID 51 – Type of fuel used by the vehicle (not supported)

54.17.82 Description of PID 52 – Alcohol Fuel Percentage (not supported)

54.17.83 Description of PID 53 – Absolute Evap System Vapor Pressure (not supported)

54.17.84 Description of PID 54 – Evap System Vapor Pressure (not supported)

54.17.85 Description of PID 55 – Short term fuel trim Bank 1

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
55	Short Term Secondary O2 Sensor Fuel Trim	A , B	-100 %	+99,22 %	100/128 %	
	Short Term Secondary O2 Sensor Fuel Trim Bank 1/3 shall indicate the correction being utilised by the closed loop fuel algorithm. If the fuel system is in open loop, STSO2FT shall report 0% correction. NOTE Data B shall only be included in the response message of a PID $55 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 3 supported) of PID $1D if Data B of PID $55 is supported or not. If PID $13 is supported, then DATA B of PID $55 shall not be supported.					

```
IF        (ID_STATE_SRV_1_PID[2] & 20h)        // PID 13h is supported
THEN
    IF        (STATE_PSN_LS_SAE & 33h)
    THEN
```

Data Byte	Application data to use	Formating conversion used
data A	LAMB_DELTA_LAM_ADJ[1]	value*100% e.g. VALUE = -0.125 => send -100%
data B	NONE	-

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	
data B	NONE	-

```
    ENDIF
ENDIF
```

54.17.86 Description of PID 56 – Long term fuel trim Bank 1

PID $56 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
56	Long Term Secondary O2 Sensor Fuel Trim	A , B	-100 %	+99,22 %	100/128 %	
	Secondary O2 Sensor Fuel trim correction for Bank 1/3 stored in Non-volatile RAM or Keep-alive RAM. LGSO2FT shall indicate the correction being utilised by the fuel control algorithm at the time the data is requested, in both open loop and closed loop fuel control. If no correction is utilised in open loop fuel, LGSO2FT shall report 0% correction. If secondary O2 sensor long-term fuel trim is not utilised at all by the fuel control algorithm, the PID shall not be supported. NOTE Data B shall only be included in the response message of a PID $56 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 3 supported) of PID $1D if Data B of PID $56 is supported or not. If PID $13 is supported, then DATA B of PID $56 shall not be supported.					

```
IF    (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
THEN
    IF          (STATE_PSN_LS_SAE & 33h)
    THEN
```

Data Byte	Application data to use	Formating conversion used
data A	LAMB_DELTA_AD_LAM_ADJ[1]	value*100% e.g. VALUE = +0.125 => send +100%
data B	NONE	-

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	-
data B	NONE	-

```
    ENDIF
ENDIF
```

54.17.87 Description of PID 57 – Short term fuel trim Bank 2

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
55	Short Term Secondary O2 Sensor Fuel Trim	A , B	-100 %	+99,22 %	100/128 %	
	Short Term Secondary O2 Sensor Fuel Trim Bank 2/4 shall indicate the correction being utilised by the closed loop fuel algorithm. If the fuel system is in open loop, STSO2FT shall report 0% correction. NOTE Data B shall only be included in the response message of a PID $57 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 4 supported) of PID $1D if Data B of PID $57 is supported or not. If PID $13 is supported, then DATA B of PID $57 shall not be supported.					

```
IF    (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
THEN
    IF          (STATE_PSN_LS_SAE & 33h)
    THEN
```

Data Byte	Application data to use	Formating conversion used
data A	LAMB_DELTA_LAM_ADJ[2]	value*100% e.g. VALUE = -0.125 => send -100%
data B	NONE	-

```
    ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	
data B	NONE	-

```
        ENDIF
    ENDIF
```

54.17.88 Description of PID 58 - Long term fuel trim Bank 2

PID $58 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
58	Long Term Secondary O2 Sensor Fuel Trim	A , B	-100 %	+99,22 %	100/128 %	
	Secondary Sensor Fuel trim correction for Bank 2/4 stored in Non-volatile RAM or Keep-alive RAM. LGSO2FT shall indicate the correction being utilised by the fuel control algorithm at the time the data is requested, in both open loop and closed loop fuel control. If no correction is utilised in open loop fuel, LGSO2FT shall report 0% correction. If post O2 sensor long-term fuel trim is not utilised at all by the fuel control algorithm, the PID shall not be supported. NOTE Data B shall only be included in the response message of a PID $58 if supported by the vehicle. If PID $1D Oxygen Sensor Location of Bank 1, 2, 3, 4 is supported then the external test equipment shall determine based on the data content (Bank 4 supported) of PID $1D if Data B of PID $58 is supported or not. If PID $13 is supported, then DATA B of PID $58 shall not be supported.					

```
    IF (ID_STATE_SRV_1_PID[2] & 20h)          // PID 13h is supported
    THEN
        IF          (STATE_PSN_LS_SAE & 33h)
        THEN
```

Data Byte	Application data to use	Formating conversion used
data A	LAMB_DELTA_AD_LAM_ADJ[2]	value*100% e.g. VALUE = +0.125 => send +100%
data B	NONE	-

```
        ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	NONE	-
data B	NONE	-

```
        ENDIF
    ENDIF
```

54.17.89 Description of PID 59 to FF - ISO/SAE reserved

54.18 Service 02h - Request powertrain freeze frame data

Import actions:

ACTION_ERRM_ReadFrfByDtc	**(** **IN <TypeOfFF>** **IN <FFidentifier>** **IN <DtcIdentifier>** **IN <LevelOfDTC>** **INOUT <Frf>** **OUT <ResultFrf>** **)**
This API returns a Freeze Frame related to the DTC given in parameter.	

54.18.1 Functional description

The purpose of this service is to allow access to emission-related data values in a freeze frame. This allows expansion to meet manufacturer specific requirements not necessarily related to the required freeze frame, and not necessarily containing the same data values as the required freeze frame. The request message includes a parameter identification (PID) value that indicates to the on-board system the specific information requested. PID specifications, scaling information, and display formats for the freeze frame are included in Annex B of ISO 15031-5.4.

The ECU(s) will respond to this message by transmitting the requested data value stored by the system. All data values returned for sensor readings will be actual stored readings, not default or substitute values used by the system because of a fault with that sensor.

Service 02h PID 02h indicates the DTC that caused the freeze frame data to be stored. If freeze frame data is not stored in the ECU, the system shall report 00 00h as the DTC.

The frame number byte will indicate 00h for the freeze frame data. Manufacturers may optionally save additional freeze frames and use this service to obtain that data by specifying the freeze frame number in the request message. If a manufacturer uses these additional freeze frames, they will be stored under conditions specified by the manufacturer, and contain data specified by the manufacturer.

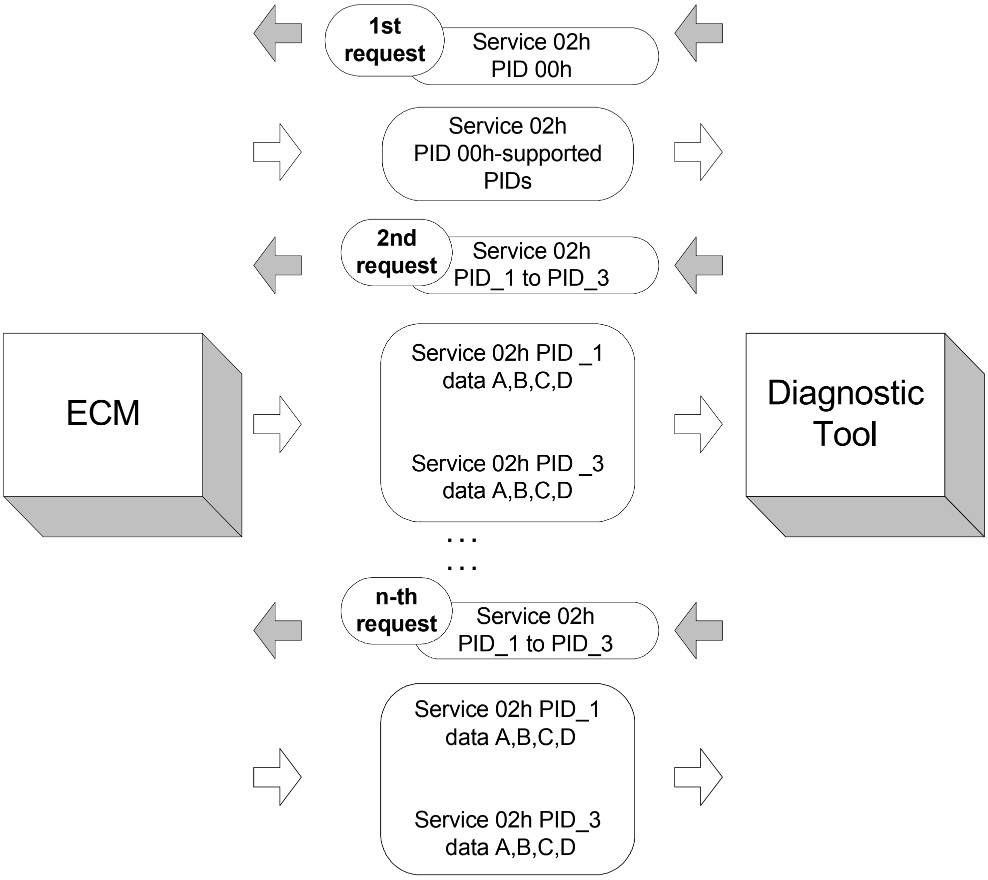
Not all PIDs are applicable or supported by all systems. PID 00h is a bit-encoded value that indicates for each ECU which PIDs are supported. PID 00h indicates support for PIDs from 01h to 20h. PID 20h indicates support for PIDs 21h through 40h, etc. PID 00h is required for those ECUs that respond to a corresponding service 02h request message as specified in Annex A of ISO 15031-5.4.

The order of the PIDs in the response message is not required to match the order in the request message.

The request message may contain up to three (3) PIDs. An external test equipment is not allowed to request a combination of PIDs supported and PIDs, which report data values. The ECU shall support requests for up to three (3) PIDs. The request message may contain the same PID multiple times. The ECU shall treat each PID as a separate parameter and respond with data for each PID as often as requested.

Proper response from server/ECU for ISO 15765-4 protocol:

Service 02h not supported	The ECU shall not respond
Service 02h supported PID requested, no Freeze Frame stored	PID 02h indicates 0000h, but if PIDs are requested, ECU must not respond except if supported PIDs (00h, 20h, …) have been requested, then the ECU shall send a response with the supported PID and data bytes
Service 02h unsupported PID requested, no Freeze Frame stored	PID 02h indicates 0000h, but if PIDs are requested, ECU must not respond except for support PIDs 00h, 20h, etc.
Service 02h supported PID requested, Freeze Frame stored	Respond within P2 timing
Service 02h unsupported PID requested, Freeze Frame stored	The ECU shall not respond



54.18.2 Message data bytes

54.18.2.1 Read supported PIDs

Request powertrain freeze frame data request message (read supported PIDs)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request powertrain freeze frame data request SID**	**M**	**02**	**SIDRQ**
#2	PID#1 (PIDs supported: Annex A of ISO 15031-5.4)	M	xx	PID
#3	frame #	M	xx	FRNO_
#4	PID#2 (PIDs supported: Annex A of ISO 15031-5.4)	U	xx	PID
#5	frame #	U/C	xx	FRNO_
#6	PID#3 (PIDs supported: Annex A of ISO 15031-5.4)	U	xx	PID
#7	frame #	U/C	xx	FRNO_

U = User Optional — PID may be included to reduce multiple PID supported request messages
C = Conditional — parameter is only included if preceding PID# is included

NOTE To request PIDs supported range from 61h - FFh, multiple request messages with PIDs = 60h, 80h, A0h, C0h and E0h shall be sent to the vehicle.

54.18.2.2 Report supported PIDs

ECU(s) must respond to all supported ranges if requested. A range is defined as a block of 32 PIDs (e.g. range #1: PID 01h-20h). The ECU shall not respond to unsupported PID ranges unless subsequent ranges have a supported PID(s).

Request powertrain freeze frame data response message (report supported PIDs)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request powertrain freeze frame data response SID**	**M**	**42**	**SIDPR**
#2	1st supported PID	M	00	PID
#3	frame #	M	xx	FRNO_
	data record of supported PIDs = [			DATAREC
#4	Data A: supported PIDs,	M	xx	DATA_A
#5	Data B: supported PIDs,	M	xx	DATA_B
#6	Data C: supported PIDs,	M	xx	DATA_C
#7	Data D: supported PIDs]	M	xx	DATA_D
:	:	:	:	:
#n-5	mth supported PID	C1	xx	PID
#n-4	frame #	C1	xx	FRNO_
	data record of supported PIDs = [			DATAREC
#n-3	Data A: supported PIDs,	C2	xx	DATA_A
#n-2	Data B: supported PIDs,	C2	xx	DATA_B
#n-1	Data C: supported PIDs,	C2	xx	DATA_C
#n	Data D: supported PIDs]	C2	xx	DATA_D

C1 = Conditional — PID value shall be the same value as included in the request message if supported by the ECU
C2 = Conditional — value indicates PIDs supported; range of supported PIDs depends on selected PID value (see C1)

NOTE The response message shall only include the PID(s) and Data A - D which are supported by the ECU. If the request message includes (a) PID value(s) which are not supported by the ECU those shall not be included in the response message.

54.18.2.3 Read freeze frame PID values

Request powertrain freeze frame data request message (read freeze frame PID values)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request powertrain freeze frame data request SID**	**M**	**02**	**SIDRQ**
#2	PID#1 (see Annex B of ISO 15031-5.4)	M	xx	PID
#3	frame #	M	xx	FRNO
#4	PID#2 (see Annex B of ISO 15031-5.4)	U	xx	PID
#5	frame #	C1	xx	FRNO
#6	PID#3 (see Annex B of ISO 15031-5.4)	U	xx	PID
#7	frame #	C1	xx	FRNO

U = User Optional — the parameter may be present or not
C1 = Conditional — parameter is only present if preceding PID# is present

54.18.2.4 Report freeze frame PID values

Request powertrain freeze frame data response message (report freeze frame PID values)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request powertrain freeze frame data response SID**	**M**	**42**	**SIDPR**
#2	1st supported PID	M	xx	PID_
#3	frame #	M	xx	FRNO_
#4 #5 #6 #7	data record of 1st supported PID = [data A, data B, data C, data D]	M C1 C1 C1	xx xx xx xx	DATA_A DATA_B DATA_C DATA_D
:	:	:	:	:
#2	mth supported PID	C2	xx	PID_
#3	frame #	C2	xx	FRNO_
#4 #5 #6 #7	data record of mth supported PID = [data A, data B, data C, data D]	C3 C4 C4 C4	xx xx xx xx	DATA_A DATA_B DATA_C DATA_D

C1 = Conditional — “data B - D” depend on selected PID
C2 = Conditional — parameter shall be the same value as included in the request message and only present if supported
C3 = Conditional — data A shall be included if preceding PID is supported
C4 = Conditional — parameters and values for “data B - D” depend on selected PID number

54.18.2.5 Parameter definition

54.18.2.5.1 PIDs supported

“Annex A of ISO 15031-5.4” specifies the interpretation of the data record of supported PIDs.

54.18.2.5.2 PID and data byte descriptions

“Annex B of ISO 15031-5.4” specifies standardised emission-related parameters.

54.18.2.5.3 Frame # description

The frame number identifies the freeze frame, which includes emission-related data values in case an emission-related DTC is detected by the ECU.

54.18.2.6 Formula section

ACTION_ERRM_ReadFrfByDtc

```
(
IN <LAW>
IN <FIRST>
IN <NO_DTC>
IN <LAW>
INOUT <Frf>
OUT <ResultFrf>
)
```

```
IF (PID = X0h)  where PID X0h means PID 00h, 20h, 40h, 60h, 80h, A0h, C0h, E0h
THEN
        Positive response
ELSE
        IF ResultFrf = FRF_PRESENT THEN
                Positive response
        ELSE
                IF PID = 02h            THEN  Positive response with DTC = 0000h
                                        ELSE  No answer
                ENDIF
        ENDIF
ENDIF
```

54.19 Service 02h - Request powertrain freeze frame data (Application Incident)

Import action:

```
ACTION_ERRM_ReadFrfByDtc(
                         IN <TypeOfFF>,
                         IN <FFIdentifier>,
                         IN <DtcIdentifier>,
                         IN < LevelOfDtc>,
                         INOUT <Frf>,
                         OUT <ResultFrf>
                        )
```

This API returns a freeze frame or a part of freeze frame.

Input data:

STATE_PSN_LS_SAE	STATE_VAR_SAP_SAE	NC_INJ_CONF	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_STATE_SRV_2_PID	12	0...FFH	0...255	1	[-]
LDPM_STATE_PID_SRV_1_2_OBDC	12	0...BH	0...11	1	[-]
Table of bytes which defines the PID's supported by the service 01h (01h - 60h)					

FUNCTION DESCRIPTION:

General information:

Depending on calibration of ID_STATE_SRV_2_PID the PID's are supported or not.

Each LDPM[0...11] of the table represents an 8 PIDs configuration:

0 => the PID is not supported
1 => the PID is supported

" IDX" is used instead of the array definition [NC_NR_ERR_DYN].

At STATE_PSN_LS_SAE you can see the configuration of the oxygen-sensors:

BIT	Function	Status
0	Bank 1 Sensor 1	1= available, 0 = not available
1	Bank 1 Sensor 2	1= available, 0 = not available
2	Bank 1 Sensor 3	1= available, 0 = not available
3	Bank 1 Sensor 4	1= available, 0 = not available
4	Bank 2 Sensor 1	1= available, 0 = not available
5	Bank 2 Sensor 2	1= available, 0 = not available
6	Bank 2 Sensor 3	1= available, 0 = not available
7	Bank 2 Sensor 4	1= available, 0 = not available

54.19.1 Description of PID 00h - Supported PIDs 01h - 20h

DATA A - Supported PIDs 01h to 08h

PID 00h - DATA A - Supported PID 01h to 08h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[0]
0	PID 08h : Short term fuel trim - Bank 2 and Bank 4	1
1	PID 07h : Long term fuel trim - Bank 1 and Bank 3	1
2	PID 06h : Short term fuel trim - Bank 1 and Bank 3	1
3	PID 05h : Engine coolant temperature	1
4	PID 04h : Calculated load value	1
5	PID 03h : Fuel system status	1
6	PID 02h : Trouble code that caused required freeze frame	1
7	PID 01h : Monitor status since DTCs cleared	0

DATA B - Supported PIDs 09h to 10h

PID 00h - DATA B - Supported PID 09h to 10h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[1]
0	PID 10h : Air flow rate from MAF sensor	1
1	PID 0Fh : Intake air temperature	1
2	PID 0Eh : Ignition time advance	1
3	PID 0Dh : Vehicle speed	1
4	PID 0Ch : Engine RPM	1
5	PID 0Bh : Intake manifold absolute pressure	1
6	PID 0Ah : Fuel pressure (gauge)	1
7	PID 09h : Long term fuel trim - Bank 2	1

DATA C - Supported PIDs 11h to 18h

PID 00h - DATA C - Supported PID 11h to 18h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[2]
0	PID 18h : Bank 2 Sensor 1 or Bank 3 Sensor 1	0
1	PID 17h : Bank 1 Sensor 4 or Bank 2 Sensor 2	0
2	PID 16h : Bank 1 Sensor 3 or Bank 2 Sensor 1	0
3	PID 15h : Bank 1 Sensor 2	0
4	PID 14h : Bank 1 Sensor 1	0
5	PID 13h : Location of Oxygen Sensors	1
6	PID 12h : Secondary air status	1
7	PID 11h : Absolute throttle position sensor	1

DATA D - Supported PIDs 19h to 20h

PID 00h - DATA D - Supported PID 19h to 20h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[3]
0	PID 20h : PIDs supported ($21 - $40)	1
1	PID 1Fh : Time Since Engine Start	1
2	PID 1Eh : Auxiliary Input Status	0
3	PID 1Dh : Location of oxygen sensors	0
4	PID 1Ch : OBD requirements to which vehicle is designed	0
5	PID 1Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2	0
6	PID 1Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1	0
7	PID 19h : Bank 2 Sensor 2 or Bank 3 Sensor 2	0

54.19.1.1 Description of PID 02h - Diagnostic trouble codes

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling	External test equipment SI (Metric) / English display
02	DTC that caused required freeze frame data storage ($0000 indicates no freeze frame data)	A, B	00 00	FF FF	Hexadecimal e.g. P01AB (DTCs defined in ISO 15031-6)	DTCFRZF: Pxxxx, Cxxxx, Bxxxx, Uxxxx

Formula section:

```
CALL API :
ACTION_ERRM_ReadFrfByDtc        (
                                IN <FRF_LAW>,
                                IN <FIRST>,
                                IN <NO_DTC>,
                                IN <DTC_LAW>
                                INOUT <Frf>
                                OUT <ResultFrf>
                                )
IF ResultFrf == NO_FRF_PRESENT
THEN
```

Data Byte	Application data to use	Formating conversion used
data A	`0x00`	-
data B	`0x00`	

```
ELSE
```

Data Byte	Application data to use	Formating conversion used
data A	`Frf[0]`	-
data B	`Frf[1]`	

```
ENDIF
```

54.19.1.2 Description of PID 03h - Status of fuel system

```
If    (ID_STATE_SRV_2_PID[2] & 20h)          // PID 13h is supported
Then
    If        (STATE_PSN_LS_SAE & 11h)           // upstream sensors available
    Then
              DATA_A = ENVD_OBD_1_IDX            // FRF of STATE_LS_1
              DATA_B = ENVD_OBD_2_IDX            //FRF of STATE_LS_2
    Endif
```

```
Endif
```

54.19.1.3 Description of PID 04h - Calculated load value

```
DATA_A = ENVD_OBD_3_IDX          // FRF of LOAD_CLC
```

54.19.1.4 Description of PID 05h - Engine coolant temperature

```
DATA_A = ENVD_OBD_4_IDX          // FRF of TCO_MES
```

54.19.1.5 Description of PID 06h - Short term fuel trim - Bank 1 and Bank 3

```
If   (ID_STATE_SRV_2_PID[2] & 20h)          // PID 13h is supported
Then
    If      (STATE_PSN_LS_SAE & 11h)        // upstream sensors available
    Then    PID 06h is supported
            DATA_A = ENVD_OBD_5_IDX         // FRF of OBD_LAM_COR_1
            DATA_B = NONE                   //Bank 3 not supported
    Endif
Endif
```

54.19.1.6 Description of PID 07h - Long term fuel trim - Bank 1 and Bank 3

```
If   (ID_STATE_SRV_2_PID[2] & 20h)          // PID 13h is supported
Then
    If      (STATE_PSN_LS_SAE & 33h)        // upstream and downstream sensors available
    Then    PID 07h is supported
            DATA_A = ENVD_OBD_6_IDX         // FRF of OBD_LAM_AD_1
            DATA_B = NONE                   //Bank 3 not supported
    Endif
Endif
```

54.19.1.7 Description of PID 08h - Short term fuel trim - Bank 2 and Bank 4

```
If   (ID_STATE_SRV_2_PID[2] & 20h)          // PID 13h is supported
Then
    If      (STATE_PSN_LS_SAE & 11h)        // upstream sensors available
    Then    PID 08h is supported
            DATA_A = ENVD_OBD_7_IDX         // FRF of OBD_LAM_COR_2
            DATA_B = NONE                   //Bank 3 not supported
    Endif
```

```
Endif
```

54.19.1.8 Description of PID 09h - Long term fuel trim - Bank 2 and Bank 4

```
If      (ID_STATE_SRV_2_PID[2] & 20h)           // PID 13h is supported
Then
        If      (STATE_PSN_LS_SAE & 33h)        // upstream  and downstream  sensors available
        Then            PID 09h is supported
                        DATA_A = ENVD_OBD_8_IDX         // FRF of OBD_LAM_AD_2
                        DATA_B = NONE                   //Bank 3 not supported
        Endif
Endif
```

54.19.1.9 Description of PID 0Ah - Fuel pressure (gauge)

```
DATA_A = ENVD_OBD_9_IDX         // FRF of OBD_FUP
```

54.19.1.10 Description of PID 0Bh - Intake manifold absolute pressure

```
DATA_A = ENVD_OBD_10_IDX        // FRF of MAP_SAE
```

54.19.1.11 Description of PID 0Ch - Engine speed

```
DATA_A = ENVD_OBD_11_IDX        // FRF of OBD_N (high byte)
DATA_B = ENVD_OBD_12_IDX        // FRF of OBD_N (low byte)
```

54.19.1.12 Description of PID 0Dh - Vehicle speed

```
DATA_A = ENVD_OBD_13_IDX        // FRF of VS
```

54.19.1.13 Description of PID 0Eh - Ignition timing advance

```
DATA_A = ENVD_OBD_14_IDX        // FRF of OBD_IGA_IGC, cylinder 1
```

54.19.1.14 Description of PID 0Fh - Intake air temperature

```
DATA_A = ENVD_OBD_15_IDX        // FRF of OBD_TIA
```

54.19.1.15 Description of PID 10h - Air flow rate from MAF

```
DATA_A = ENVD_OBD_16_IDX        // FRF of OBD_MAF (high byte)
DATA_B = ENVD_OBD_17_IDX        // FRF of OBD_MAF (low byte)
```

54.19.1.16 Description of PID 11h - Absolute throttle position sensor

```
DATA_A = ENVD_OBD_18_IDX        // FRF of OBD_TPS_1
```

54.19.1.17 Description of PID 12h - Commanded secondary air status

```
If      (ID_STATE_SRV_2_PID[2] & 40h) and (STATE_VAR_SAP_SAE <> 0)
Then    DATA_A = ENVD_OBD_19_IDX            // FRF of STATE_OBD_SA
Endif
```

54.19.1.18 Description of PID 1Fh - Time Since Engine Start

```
DATA_A = ENVD_OBD_20_IDX        // FRF of T_AST_SAE (high byte)
DATA_B = ENVD_OBD_21_IDX        // FRF of T_AST_SAE (low byte)
```

54.19.2 Description of PID 20h - Supported PIDs 21h - 40h

0 : the PID is not used, 1 : the PID is used

DATA A - Supported PIDs 21h - 28h

PID 20h - DATA A - Supported PID 21h to 28h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[4]
0	PID 28h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	0
1	PID 27h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	0
2	PID 26h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	0
3	PID 25h : Bank 1 Sensor 2 (Wide range O2S)	0
4	PID 24h : Bank 1 Sensor 1 (Wide range O2S)	0
5	PID 23h : High fuel pressure (gauge)	1
6	PID 22h : Relative fuel pressure	0
7	PID 21h : Distance travelled while MIL is activated	0

DATA B - Supported PIDs 29h to 30h

PID 20h - DATA B - Supported PID 29h to 30h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[5]
0	PID 30h : Number of warm-ups since diagnostic trouble codes cleared	0
1	PID 2Fh : Fuel Level Input	1
2	PID 2Eh : Commanded Evaporative Purge	1
3	PID 2Dh : EGR Error	0
4	PID 2Ch : Command EGR	0
5	PID 2Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	0
6	PID 2Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	0
7	PID 29h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	0

DATA C - Supported PIDs 31h to 38h

PID 20h - DATA C - Supported PID 31h to 38h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[6]
0	PID 38h : Bank 2 Sensor 1 or Bank 3 Sensor 1 (Wide range O2S)	0
1	PID 37h : Bank 1 Sensor 4 or Bank 2 Sensor 2 (Wide range O2S)	0
2	PID 36h : Bank 1 Sensor 3 or Bank 2 Sensor 1 (Wide range O2S)	0
3	PID 35h : Bank 1 Sensor 2 (Wide range O2S)	0
4	PID 34h : Bank 1 Sensor 1 (Wide range O2S)	0
5	PID 33h : Barometric Pressure	1
6	PID 32h : Evap System Vapor Pressure	0
7	PID 31h : Distance since diagnostic trouble codes cleared	0

DATA D - Supported PIDs 39h to 40h

PID 20h - DATA D - Supported PID 39h to 40h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[7]
0	PID 40h : PIDs supported ($41 - $60)	1
1	PID 3Fh : Catalyst Temperature Bank 2, Sensor 2	0
2	PID 3Eh : Catalyst Temperature Bank 1, Sensor 2	0
3	PID 3Dh : Catalyst Temperature Bank 2, Sensor 1	0
4	PID 3Ch : Catalyst Temperature Bank 1, Sensor 1	0
5	PID 3Bh : Bank 2 Sensor 4 or Bank 4 Sensor 2 (Wide range O2S)	0
6	PID 3Ah : Bank 2 Sensor 3 or Bank 4 Sensor 1 (Wide range O2S)	0
7	PID 39h : Bank 2 Sensor 2 or Bank 3 Sensor 2 (Wide range O2S)	0

54.19.2.1 Description of PID 23h – Fuel pressure

```
If      NC_INJ_CONF <> 0                //only MSD
Then    DATA_A = ENVD_OBD_22_IDX        // FRF of OBD_FUP_RNG_H
        DATA_B = ENVD_OBD_23_IDX        // FRF of OBD_FUP_RNG_H
Endif
```

54.19.2.2 Description of PID 2Eh – Commanded Evaporative purge

DATA_A = ENVD_OBD_24_IDX *// FRF of CPPWM_CPS*

54.19.2.3 Description of PID 2Fh - Fuel Level Input

DATA_A = ENVD_OBD_25_IDX *// FRF of OBD_FTL*

54.19.2.4 Description of PID 33h - Barometric Pressure

DATA_A = ENVD_OBD_26_IDX *// FRF of OBD_AMP*

54.19.3 Description of PID 40h - Supported PIDs 41h - 60h

0 : the PID is not used
1 : the PID is used

DATA A - Supported PIDs 41h - 48h

PID 40h - DATA A - Supported PID 41h to 48h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[8]
0	PID 48h : Absolute Throttle Position C	0
1	PID 47h : Absolute Throttle Position B	1
2	PID 46h : Ambient air temperature	1
3	PID 45h : Relative Throttle Position	1
4	PID 44h : Command Equivalence Ratio	1
5	PID 43h : Absolue Load Value	1
6	PID 42h : Control module voltage	1
7	PID 41h : Monitor status this driving cycle	0

DATA B - Supported PIDs 49h to 50h

PID 40h - DATA B - Supported PID 49h to 50h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[9]
0	PID 50h : Reserved	0
1	PID 4Fh : Reserved	0
2	PID 4Eh : Time since diagnostic trouble codes cleared	0
3	PID 4Dh : Minutes run by the engine while MIL activated	0
4	PID 4Ch : Commanded Throttle Actuator Control	1
5	PID 4Bh : Accelerator Pedal Position F	0
6	PID 4Ah : Accelerator Pedal Position E	1
7	PID 49h : Accelerator Pedal Position D	1

DATA C - Supported PIDs 51h to 58h

PID 40h - DATA C - Supported PID 51h to 58h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[10]
0	PID 58h : Reserved	0
1	PID 57h : Reserved	0
2	PID 56h : Reserved	0
3	PID 55h : Reserved	0
4	PID 54h : Reserved	0
5	PID 53h : Reserved	0
6	PID 52h : Reserved	0
7	PID 51h : Reserved	0

DATA D - Supported PIDs 59h to 60h

PID 40h - DATA D - Supported PID 59h to 60h		
BIT	**Function**	**Bitmask:** ID_STATE_SRV_2_PID[11]
0	PID 60h : Reserved	0
1	PID 5Fh : Reserved	0
2	PID 5Eh : Reserved	0
3	PID 5Dh : Reserved	0
4	PID 5Ch : Reserved	0
5	PID 5Bh : Reserved	0
6	PID 5Ah : Reserved	0
7	PID 59h : Reserved	0

54.19.3.1 Description of PID 42 - Control module voltage

DATA_A = ENVD_OBD_27_IDX *// FRF of OBD_VB (high byte)*

DATA_B = ENVD_OBD_28_IDX *// FRF of OBD_VB (low byte)*

54.19.3.2 Description of PID 43 - Absolute Load Value

DATA_A = ENVD_OBD_29_IDX *// FRF of LOAD_ABSV (high byte)*

DATA_B = ENVD_OBD_30_IDX *// FRF of LOAD_ABSV (low byte)*

54.19.3.3 Description of PID 44 - Commanded Equivalence Ratio

DATA_A = ENVD_OBD_31_IDX *// FRF of OBD_LAMP_SP (high byte)*

DATA_B = ENVD_OBD_32_IDX *// FRF of OBD_LAMP_SP (low byte)*

54.19.3.4 Description of PID 45 - Relative Throttle Position

DATA_A = ENVD_OBD_33_IDX *// FRF of OBD_TPS_REL*

54.19.3.5 Description of PID 46 - Ambient air temperature

DATA_A = ENVD_OBD_34_IDX *// FRF of OBD_TAM*

54.19.3.6 Description of PID 47 - Absolute Throttle Position B

DATA_A = ENVD_OBD_35_IDX *// FRF of OBD_TPS_2*

54.19.3.7 Description of PID 49 - Accelerator Pedal Position D

DATA_A = ENVD_OBD_36_IDX *// FRF of OBD_PV_1*

54.19.3.8 Description of PID 4A - Accelerator Pedal Position E

DATA_A = ENVD_OBD_37_IDX *// FRF of OBD_PV_2*

54.19.3.9 Description of PID 4C – Commanded throttle actuator control

DATA_A = ENVD_OBD_38_IDX *// FRF of OBD_TPS_SP*

54.20 Service 03h - Request emission-related powertrain diagnostic trouble codes – OBD on CAN

Import actions:

ACTION_ERRM_ReadDtcByTypeOfDtc **(** **IN <TypeOfDtc>** **IN <LevelOfDtc>** **INOUT <ListOfDtc>** **OUT <ResultDtc>** **)**
This API returns a list of DTC.
ACTION_ERRM_ReadQuantityOfDtc **(** **IN <TypeOfDtc>** **IN <LevelOfDtc>** **INOUT <Quantity>** **OUT <ResultQuantity>** **)**
This API returns the quantity of DTC with a certain type, which are stored in memory

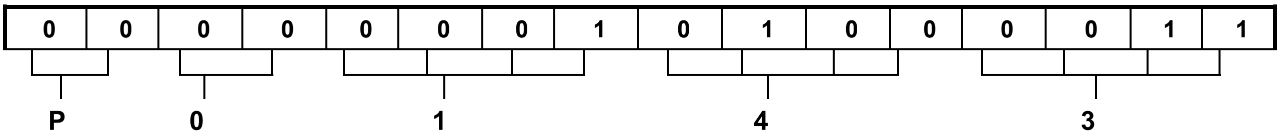
54.20.1 Functional description

The purpose of this service is to enable the external test equipment to obtain “confirmed” emission-related DTCs (diagnostic trouble codes).

Send a Service $03 request for all emission-related DTCs. Each ECU that has DTCs will respond with one (1) message containing all emission-related DTCs. If an ECU does not have emission-related DTCs then it shall respond with a message indicating no DTCs are stored by setting the parameter # of DTC to 00h

.

DTCs are transmitted in two (2) bytes of information for each DTC. The first two (2) bits (high order) of the first (1) byte for each DTC indicate whether the DTC is a Powertrain, Chassis, Body, or Network DTC (refer to ISO 15031-6 for additional interpretation of this structure). The second two (2) bits will indicate the first digit of the DTC (0 through 3). The second (2) nibble of the first (1) byte and the entire second (2) byte are the next three (3) hexadecimal characters of the actual DTC reported as hexadecimal. A Powertrain DTC transmitted as 0143h shall be displayed as P0143.



Diagnostic trouble code encoding example DTC P0143

54.20.2 Message data bytes

54.20.2.1 Request message definition

Request emission-related DTC request message

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request emission-related DTC request SID**	**M**	**03**	**SIDRQ**

54.20.2.2 Response message definition

Request emission-related DTC response message

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request emission-related DTC response SID**	**M**	**43**	**SIDPR**
#2	# of DTC = [no emission-related DTCs stored emission-related DTCs stored]	M	xx = [00, 01 - FF	#OFDTC
#3 #4	DTC#1 (High Byte) DTC#1 (Low Byte)	C C	xx xx	DTC1HI DTC1LO
:	:	:	xx	
#n-1 #n	DTC#m (High Byte) DTC#m (Low Byte)	C C	xx xx	DTCmHI DTCmLO
C = Conditional — DTC#1 - DTC#m are only included if # of DTC parameter value ≠ $00				

54.20.2.3 Parameter definition

54.20.2.3.1 # of DTC parameter description

The # of DTC parameter reports the emission-related DTC(s) currently (at the time of the request message processing) stored in the ECU(s).

54.20.2.4 Formula section

```
ACTION_ERRM_ReadQuantityOfDtc
                                        (
                                        IN <OBD&CONFIRMED>
                                        IN <LAW>
                                        INOUT <NR_DTC>
                                        OUT <ResultQuantity>
                                        )

ACTION_ERRM_ReadDtcByTypeOfDtc
                                        (
                                        IN <OBD&CONFIRMED>
                                        IN <LAW>
                                        INOUT <DTC#1 to DTC#m>
                                        OUT <ResultDtc>
                                        )

IF      ResultDtc != BUFFER_FULL THEN
        Positive response
ELSE
        No answer
ENDIF
```

54.21 Service 04h - Clear/reset emission-related diagnostic information

Export actions:

ACTION_OBDC_Srv04Cnd (OUT < PRM_STATE_SRV_4_RESP >)
This function permit to clear all failures in dynamic memory, or not

Import actions:

ACTION_ERRM_ClrInfoByTypeOfDtc(IN <TypeOfDTC> OUT <ResultClrInfo>)
This API clears all failure in dynamic memory
ACTION_OBDC_Srv04SpActions ()
This function permit to project to add their own manufacturer specific "clearing/resetting" actions
ACTION_OBDC_Srv04CndSp (OUT <>)
This action permit to clear all failures in dynamic memory, or not - contains project specific conditions This action is included only if it is defined in appl. inc. xxI06E01.xxx

Input data:

NLC_ERR_CLR_ENG_RUN_SAE	LV_IGK	LV_ES	

54.21.1 Functional description

The purpose of this service is to provide a means for the external test equipment to command ECUs to clear the whole failure memory (all emission-related and non emission-related diagnostic information) (refer to Error Management module)

This includes:

— MIL and number of diagnostic trouble codes	(can be read with Service 01h, PID 01h)
— Clear the I/M (Inspection/Maintenance) readiness bits	(Service 01h, PID 01h and 41h)
— Confirmed diagnostic trouble codes	(can be read with Service 03h)
— Pending diagnostic trouble codes	(can be read with Service 07h)
— Diagnostic trouble code for freeze frame data	(can be read with Service 02h, PID 02h)
— Freeze frame data	(can be read with Service 02h)
— Status of system monitoring tests	(can be read with Service 01h, PID 01h)
— On-board monitoring test results	(can be read with Service 06h)
— Distance travelled while MIL is activated	(can be read with Service 01h, PID 21h)
— Number of warm-ups since DTCs cleared	(can be read with Service 01h, PID 30h)
— Distance travelled since DTCs cleared	(can be read with Service 01h, PID 31h)
— Time run by the engine while MIL is activated	(can be read with Service 01h, PID 4Dh)
— Time since diagnostic trouble codes cleared	(can be read with Service 01h, PID 4Eh)
— Reset misfire counts of standardized Test ID 0Bh to zero	(can be read with Service 06h)

Description of parameter OUT used by function ACTION_OBDC_Srv04Cnd ():

PRM_STATE_SRV_4_RESP: Result to say if the function is done or not.

Authorised values for PRM_STATE_SRV_4_RESP are:
(1) The call of the function is successful
(0) The call of the function is not successful

Description for actions:

ACTION_OBDC_Srv04Cnd (OUT < PRM_STATE_SRV_4_RESP >)					
This function permit to clear all failures in dynamic memory, or not. In this function the project should define their own conditions in order to permit or not the clearing performed with Service 04h					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_SRV_4_RESP	OUT	00H..FFH	0..255	1	-
Result to say if the function is done or not					

The purpose of this function is to define the conditions in order to permit to clear all failures in dynamic memory, or not.
All ECUs shall respond to this request message with ignition ON, with the engine not running.

Formula section:

```
IF NLC_ERR_CLR_ENG_RUN_SAE THEN
        IF LV_IGK
        THEN
                PRM_STATE_SRV_4_RESP = 1
        ELSE
                PRM_STATE_SRV_4_RESP = 0
        ENDIF
ELSE
        IF LV_IGK AND LV_ES
        THEN
                PRM_STATE_SRV_4_RESP = 1
        ELSE
                PRM_STATE_SRV_4_RESP = 0
        ENDIF
ENDIF
```

54.21.2 Message data bytes

54.21.2.1 Request message

Bytes	Parameter name	Hex value
#01	**Request to clear/reset emission-related diagnostic information**	04

For safety and/or technical design reasons, ECUs that can not perform this operation under other conditions, such as with the engine running shall send a negative response message with response code 22h - conditionsNotCorrect.

54.21.2.1.1 Formula section

If there are used the generic conditions in order to check if the clearing of all failures in dynamic memory, can be performed or not, this is the sequence to be respected:

```
IF ACTION_OBDC_Srv04Cnd()
THEN
        ACTION_ERRM_ClrInfoByTypeOfDtc
                                        (
                                        IN <ALL>
                                        OUT <ResultClrInfo>
                                        )
        CALL ACTION_OBDC_Srv04SpActions()
        Positive response;
ELSE
        Negative response with "Condition not correct";
ENDIF
```

If there are used the project specific conditions in order to check if the clearing of all failures in dynamic memory, can be performed or not, this is the sequence to be respected:

```
IF ACTION_OBDC_Srv04CndSp()
THEN
        ACTION_ERRM_ClrInfoByTypeOfDtc
                                        (
                                        IN <ALL>
                                        OUT <ResultClrInfo>
                                        )
        CALL ACTION_OBDC_Srv04SpActions()
        Positive response;
ELSE
        Negative response with "Condition not correct";
ENDIF
```

54.21.2.2 Positive response message

Bytes	Parameter name	Hex value
#01	**Request to clear/reset emission-related diagnostic information positive response**	44

54.21.2.3 Negative response message

Bytes	Parameter name	Hex value
#1	**Negative response**	7F
#2	**Request to clear/reset emission-related diagnostic information**	04
#3	Error code = Condition not correct	22

54.21.2.4 Error code

Error code	Cause
Condition not correct	In case of ECU can not perform the clearing

54.22 Service 04h - Clear/reset emission-related diagnostic information (Application Incident)

Input data:

N_32	LV_LOCK_IMOB	VS	

Export actions:

ACTION_OBDC_Srv04CndSp (OUT < PRM_STATE_SRV_4_SP_RESP >)
This function permit to clear all failures in dynamic memory, or not
ACTION_OBDC_Srv04SpActions ()
This function permit to project to add their own manufacturer specific "clearing/resetting" actions

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_ERR_CLR_ENG_RUN_SAE	1	0...1H	0...1	1	[-]
Byte which indicates if the erase will be done when engine is running or not					

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_32_MAX_CLR_FMY_SRV_4	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed threshold to erease error memory in service 04					

FUNCTION DESCRIPTION:

General information:

The purpose of this function is to define the conditions in order to permit to clear all failures in dynamic memory, or not.

54.22.1 Definition of NLC_ERR_CLR_ENG_RUN_SAE

Description:

Erase with or without engine running - NLC_ERR_CLR_ENG_RUN_SAE	
Value	**Description**
0	The clearing of failures is performed when engine is NOT running
1	The clearing of failures is performed when engine is running

Formula section:

NLC_ERR_CLR_ENG_RUN_SAE = 1

54.22.2 Description of parameter OUT used by function ACTION_OBDC_Srv04CndSp():

Description:

PRM_STATE_SRV_4_SP_RESP: Result to say if the function is done or not.

Authorised values for PRM_STATE_SRV_4_SP_RESP are:
(1) The call of the function is successful
(0) The call of the function is not successful

Description for actions:

ACTION_OBDC_Srv04CndSp (OUT < PRM_STATE_SRV_4_SP_RESP >)					
This function permit to clear all failures in dynamic memory, or not. In this function the project should define their own conditions in order to permit or not the clearing performed with Service $04					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_SRV_4_SP_RESP	OUT	00H...FFH	0...255	1	[-]
Result to say if the function is done or not					

Formula section:

```
IF   LV_LOCK_IMOB = 0          AND
     VS = 0                    AND
     N_32 < C_N_32_MAX_CLR_FMY_SRV_4
THEN
     PRM_STATE_SRV_4_SP_RESP = 1
ELSE
     PRM_STATE_SRV_4_SP_RESP = 0
ENDIF
```

54.22.3 Description of parameter OUT used by function ACTION_OBDC_Srv04SpActions():

Description:

Other manufacturer specific "clearing/resetting" actions may also occur in response to the Service $04 request.

ACTION_OBDC_Srv04SpActions ( )					
This function permit to project to add their own manufacturer specific "clearing/resetting" actions performed at a Service $04 request (e.g. writing in NVMY information related to the number ok Km when the clearing was performed, the type of the tool which performed the clearing, etc)					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
see description at beginning of specification					

Formula section:

ACTION not used yet

54.23 Service 06h - Request on-board monitoring test result for non-continuously monitored system – OBD on CAN

54.23.1 Functional description

The purpose of this service is to allow access to the results for on-board diagnostic monitoring tests of specific components / systems that are continuously monitored (e.g. mis-fire monitoring) and non-continuously monitored (e.g. catalyst system).

The request message for test values includes an On-Board Diagnostic Monitor ID (see Annex D of ISO 15031-5.4) that indicates the information requested. Unit and Scaling information is included in Annex E of ISO 15031-5.4.

The vehicle manufacturer is responsible for assigning "Manufacturer Defined Test IDs" for different tests of a monitored system. The latest valid test values (results) are to be retained, even over multiple ignition OFF cycles, until replaced by more recent test values (results). Test values (results) are requested by On-Board Diagnostic Monitor ID. Test values (results) are always reported with the Minimum and Maximum Test Limits. The Unit and Scaling ID included in the response message defines the scaling and unit to be used by the external test equipment to display the test values (results), Minimum Test Limit, and Maximum Test Limit information.

If an On-Board Diagnostic Monitor has not been completed at least once since Clear/reset emission-related diagnostic information or battery disconnect, then the parameters Test Value (Results), Minimum Test Limit, and Maximum Test Limit shall be set to zero (0000h) values.

Not all On-Board Diagnostic Monitor IDs are applicable or supported by all systems. On-Board Diagnostic Monitor ID 00h is a bit-encoded value that indicates for each ECU which On-Board Diagnostic Monitor IDs are supported. On-Board Diagnostic Monitor ID 00h indicates support for On-Board Diagnostic Monitor IDs from 01h to 20h. On-Board Diagnostic Monitor ID 20h indicates support for On-Board Diagnostic Monitor IDs 21h through 40h, etc. This is the same concept for PIDs/TIDs/InfoTypes support in services 01h, 02h, 06h, 08h, and 09h. On-Board Diagnostic Monitor ID 00h is required for those ECUs that respond to a corresponding service 06h request message as specified in Annex A of ISO 15031-5.4.

The request message including supported On-Board Diagnostic Monitor IDs may contain up to six (6) OBDMIDs. A request message including an On-Board Diagnostic Monitor ID, which reports test values shall only contain one (1) OBDMID. An external test equipment is not allowed to request a combination of OBDMIDs supported and a single OBDMID, which report test values. The ECU shall support requests for up to six (6) supported OBDMIDs and only one (1) OBDMID which reports test values.

A unique method must be utilized for displaying data for monitors that have multiple tests. Many OBD monitors have multiple tests that that are done in either a serial or parallel manner. If a monitor uses multiple OBD Monitor ID/Test ID combinations that may not all complete at the same time, the following method shall be used to update the stored test results at the time of monitor completion:

After the monitor completes, update all Monitor ID/Test ID combinations (or “test results”) that were utilized by the monitor with appropriate passing or failing results. If a test result (or “Monitor ID/Test ID”) was not utilized during this monitoring event, set the Test Values and Minimum and Maximum Test Limits to their initial values (0000h, test not completed). Test results from the previously completed monitoring events shall not be mixed with test results from the current completed monitoring event.

In some cases, test results (or “Monitor ID/Test ID combinations”) will be displayed as being incomplete even though the monitor (as indicated by PID 41h) was successfully completed and either passed or failed. In other cases, some Test IDs will show passing results while others will show failing results after the monitor (as indicated by PID 41h) was successfully completed and failed. Note that OBD-II regulations prohibit a passing monitor from showing any failing test results. If an initial, serial test indicates a failure and a subsequent re-test of the system indicates a passing result, the test that was utilized to make the passing determination should be displayed, while the failing test that was utilized to make the initial determination should be reset to it's initial values (0000h, test not completed).

As an example of a serial monitor, an evaporative system monitor can fail for a large evaporative system leak and never continue to test for small leaks or very small leaks. In this case, the Test ID for the large leak would show a failing result, while the small leak test and the very small leak test would show incomplete. As an example of the parallel monitor, a purge valve flow monitor can pass by having a large rich lambda shift, a large lean lambda shift or a large engine rpm increase. If the purge valve is activated and a large rich lambda shift occurs, the Test ID for the rich lambda shift would show a passing result while the other two Test IDs would show incomplete. Since some Test IDs for a completed monitor will show incomplete, PID 41h must be used to determine monitor completion status.

54.23.2 Message data bytes

54.23.2.1 Read supported OBDMIDs

Request on-board monitoring test results for specific monitored systems request message (read supported OBDMIDs)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request on-board monitoring test results for specific monitored systems request SID**	**M**	**06**	**SIDRQ**
#2	On-Board Diagnostic Monitor ID (OBDMIDs supported: Annex A of ISO 15031-5.4)	M	xx	OBDMID
#3	On-Board Diagnostic Monitor ID (OBDMIDs supported: Annex A of ISO 15031-5.4)	U	xx	OBDMID
#4	On-Board Diagnostic Monitor ID (OBDMIDs supported: Annex A of ISO 15031-5.4)	U	xx	OBDMID
#5	On-Board Diagnostic Monitor ID (OBDMIDs supported: Annex A of ISO 15031-5.4)	U	xx	OBDMID
#6	On-Board Diagnostic Monitor ID (OBDMIDs supported: Annex A of ISO 15031-5.4)	U	xx	OBDMID
#7	On-Board Diagnostic Monitor ID (OBDMIDs supported: Annex A of ISO 15031-5.4)	U	xx	OBDMID
U = User Optional — OBDMID may be included to avoid multiple OBDMID supported request messages				

NOTE To request OBDMIDs supported range from C1h - FFh another request message with OBDMID#1 = C0h and OBDMID#2 = E0h shall be sent to the vehicle

54.23.2.2 Report supported OBDMIDs

ECU(s) must respond to all supported ranges if requested. A range is defined as a block of 32 OBDMIDs (e.g. range #1: OBDMID 01h-20h). The ECU shall not respond to unsupported OBDMID ranges unless subsequent ranges have a supported OBDMID(s).

Request on-board monitoring test results for specific monitored systems response message (report supported OBDMIDs)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request on-board monitoring test results for specific monitored systems response SID**	**M**	**46**	**SIDPR**
	data record of supported OBDMID = [			OBDMIDREC
#2	1st supported OBDMID	M	xx	OBDMID
#3	Data A: supported OBDMIDs,	M	xx	DATA_A
#4	Data B: supported OBDMIDs,	M	xx	DATA_B
#5	Data C: supported OBDMIDs,	M	xx	DATA_C
#6	Data D: supported OBDMIDs]	M	xx	DATA_D
:	:	:	:	:
	data record of supported OBDMID = [			OBDMIDREC
#n-4	m^{th} supported OBDMID	C1	xx	OBDMID
#n-3	Data A: supported OBDMIDs,	C2	xx	DATA_A
#n-2	Data B: supported OBDMIDs,	C2	xx	DATA_B
#n-1	Data C: supported OBDMIDs,	C2	xx	DATA_C
#n	Data D: supported OBDMIDs]	C2	xx	DATA_D
C1 =	Conditional — OBDMID value shall be the same value as included in the request message if supported by the ECU			
C2 =	Conditional — value indicates OBDMIDs supported; range of supported OBDMIDs depends on selected OBDMID value (see C1)			

NOTE The response message shall only include the OBDMID(s) and Data A - D which are supported by the ECU. If the request message includes (a) OBDMID value(s) which are not supported by the ECU those shall not be included in the response message.

54.23.2.3 Read OBDMID test values

Request on-board monitoring test results for specific monitored systems request message (read OBDMID test values)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request on-board monitoring test results for specific monitored systems request SID**	**M**	**06**	**SIDRQ**
#2	On-Board Diagnostic Monitor ID	M	xx	OBDMID

54.23.2.4 Report OBDMID test values

Request on-board monitoring test results for specific monitored systems response message (report OBDMID test values)

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request on-board monitoring test results for specific monitored systems response SID**	**M**	**46**	**SIDPR**
	data record of supported OBDMID = [			OBDMIDREC
#2	On-Board Diagnostic Monitor ID	M	xx	OBDMID
#3	Std./Manuf. Defined TID#1	M	xx	S/MDTID
#4	Unit And Scaling ID#1	M	xx	UASID
#5	Test Value (High Byte)#1	M	xx	TVHI
#6	Test Value (Low Byte)#1	M	xx	TVLO
#7	Min. Test Limit (High Byte)#1	M	xx	MINTLHI
#8	Min. Test Limit (Low Byte)#1	M	xx	MINTLLO
#9	Max. Test Limit (High Byte)#1	M	xx	MAXTLHI
#10	Max. Test Limit (Low Byte)#1]	M	xx	MAXTLLO
:	:	:	:	:
	data record of supported OBDMID = [			OBDMIDREC
#n-8	On-Board Diagnostic Monitor ID	C1	xx	OBDMID
#n-7	Std./Manuf. Defined TID#m	C2	xx	S/MDTID
#n-6	Unit And Scaling ID#m	C2	xx	UASID
#n-5	Test Value (High Byte)#m	C2	xx	TVHI
#n-4	Test Value (Low Byte)#m	C2	xx	TVLO
#n-3	Min. Test Limit (High Byte)#m	C2	xx	MINTLHI
#n-2	Min. Test Limit (Low Byte)#m	C2	xx	MINTLLO
#n-1	Max. Test Limit (High Byte)#m	C2	xx	MAXTLHI
#n	Max. Test Limit (Low Byte)#m]	C2	xx	MAXTLLO

C1 = Conditional — parameter is only present if more than one (1) Manufacturer Defined TID is supported by the ECU for the requested Monitor ID.

C2 = Conditional — parameter and value depends on selected Manufacturer Defined TID number and are only included if the Manufacturer Defined TID is supported by the ECU. The value shall be zero (00h) in case the On-Board Diagnostic Monitor has not been completed at least once since Clear/reset emission-related diagnostic information or battery disconnect.

54.23.2.5 Response code

Note : The response is given following the condition made by the project.
Here is an example :

Formula section:

```
IF      Conditions from the specification XXI06Q01.XXX are fullfield
        THEN
                Positive response with test result
        ELSE
                Positive response with all data equal to 0000h
ENDIF
```

54.23.3 Parameter definition

54.23.3.1 On-Board Diagnostic Monitor IDs supported

The On-Board Diagnostic Monitor IDs supported is the same concept as used for PID support in services 01h and 02h as specified in Annex A of ISO 15031-5.4.

54.23.3.2 On-Board Diagnostic Monitor ID description

The On-Board Diagnostic Monitor ID is a one (1) byte parameter and is defined in Annex D of ISO 15031-5.4. An On-Board Diagnostic Monitor may have more than one (1) monitor test (Test ID).

54.23.3.3 Standardised and Manufacturer Defined Test ID description

The Standardised and Manufacturer Defined Test ID is a one (1) byte parameter. For example, the On-Board Diagnostic Monitor " Oxygen Sensor Monitor Bank 1 - Sensor 1" may have the following Standardised Test ID:
The table below specifies the range of identifiers.

Standardised Test ID description

Range (Hex)	Description
00	ISO/SAE reserved
01	Rich to lean sensor threshold voltage (constant)
02	Lean to rich sensor threshold voltage (constant)
03	Low sensor voltage for switch time calculation (constant)
04	High sensor voltage for switch time calculation (constant)
05	Rich to lean sensor switch time (calculated)
06	Lean to rich sensor switch time (calculated)
07	Minimum sensor voltage for test cycle (calculated)
08	Maximum sensor voltage for test cycle (calculated)
09	Time between sensor transitions (calculated)
0A	Sensor period (calculated)
0B	EWMA (Exponential Weighted Moving Average) misfire counts for last ten (10) driving cycles (calculated, rounded to an integer value) General EWMA calculation: 0.1 * (current misfire counts) + 0.9 * (previous misfire counts average) Initial value for (previous misfire counts average) = 0 Note: Internal ECU calculation registers with precision higher than one count must be used and retained to calculate the contents of registers 0Bh and 0Ch to prevent rounding errors. If this is not done, these registers will never count back down to zero after misfire stops. The calculations must be done using the high precision registers, then rounded to the nearest integer value to be output as register 0Bh and 0Ch **High_Precision_EWMA_Misfire_Counts** $_{\text{current}}$ **= Rounded [(0.1) * High_Precision_Misfire_Counts** $_{\text{current}}$ **+ (0.9) * High_Precision_EWMA_Misfire_Counts** $_{\text{previous}}$**]** Where: Rounded means rounded to the nearest integer. The high precision values are never reported, they are only used for internal calculations. This TEST ID shall be reported with OBD Monitor IDs A2h – ADh and the Scaling ID 24h.
0C	Misfire counts for last/current driving cycles (calculated, rounded to an integer value)
0D - 7F	Reserved for future standardisation

Manufacturer Defined Test ID description

Range (Hex)	Description
80 - FE	Manufacturer Defined Test ID range - this parameter is an identifier for the test performed within the On-Board Diagnostic Monitor.
FF	Reserved by document

Results of latest mandated on-board oxygen sensor monitoring tests, see figure below.

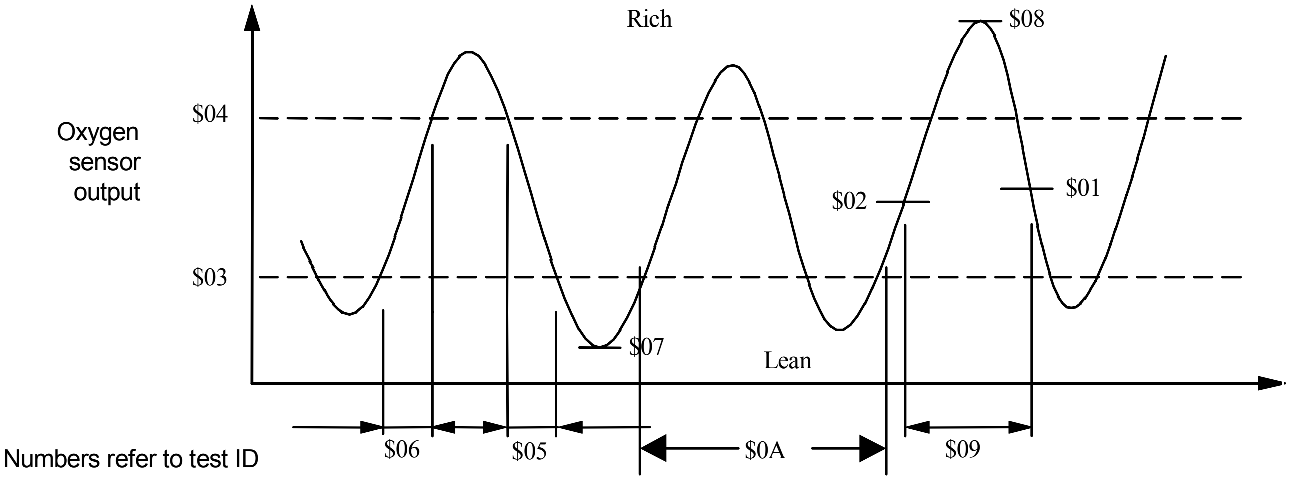


Figure 1 — Standardised Test ID value example

54.23.3.4 Unit and Scaling ID definition

The Unit and Scaling ID is a one (1) byte identifier to reference the scaling and unit to be used by the external test equipment to calculate and display the test values (results), Minimum Test Limit, and the Maximum Test Limit for the Standardised and Manufacturer Defined Test ID requested. All standardised Unit And Scaling IDs are specified in Annex E of ISO 15031-5.4.

54.23.3.5 Test Value (result) description

The Test Value represents the test result and is defined in the table below.

Test Value description

Parameter name	# of bytes	Description
Test Value	2 (High and Low Byte)	Test Value (Result) - this value shall be calculated and displayed by the external test equipment based on the Unit and Scaling ID included in the response message. The Test Value shall be within the Minimum and Maximum Test Limit to indicate a "Pass" result.

54.23.3.6 Minimum Test Limit description

The Minimum Test Limit parameter is defined in the table below.

Minimum Test Limit description

Parameter name	# of bytes	Description
Minimum Test Limit	2 (High and Low Byte)	The Minimum Test Limit shall be calculated and displayed by the external test equipment based on the Unit and Scaling ID included in the response message. The Unit and Scaling IDs are specified in Annex E of ISO 15031-5.4. The Minimum Test Limit shall be the minimum value for the monitor identified by the On-Board Diagnostic Monitor ID. For the Standardised Test IDs which are constant values the Minimum Test Limit shall be the same value as reported for the Test Value. It is the vehicle manufacturer's responsibility to provide information whether a Test Value which equals the Minimum Test Value results in a "Pass" or "Fail" condition.

54.23.3.7 Maximum Test Limit description

The Maximum Test Limit parameter is defined in the table below.

Maximum Test Limit description

Parameter name	# of bytes	Description
Maximum Test Limit	2 (High and Low Byte)	The Maximum Test Limit shall be calculated and displayed by the external test equipment based on the Unit and Scaling ID included in the response message. The Unit and Scaling IDs are specified in Annex E of ISO 15031-5.4. The Maximum Test Limit shall be the maximum value for the monitor identified by the On-Board Diagnostic Monitor ID. For the Standardised Test IDs which are constant values the Maximum Test Limit shall be the same value as reported for the Test Value. It is the vehicle manufacturer's responsibility to provide information whether a Test Value which equals the Maximum Test Value results in a "Pass" or "Fail" condition.

54.23.4 Standard definition Monitor

Standard On-Board Diagnostic Monitor ID definition

OBDMID (Hex)	On-Board Diagnostic Monitor ID name
00	OBD Monitor IDs supported (01h – 20h)
01	Oxygen Sensor Monitor Bank 1 - Sensor 1
02	Oxygen Sensor Monitor Bank 1 - Sensor 2
03	Oxygen Sensor Monitor Bank 1 - Sensor 3
04	Oxygen Sensor Monitor Bank 1 - Sensor 4
05	Oxygen Sensor Monitor Bank 2 - Sensor 1
06	Oxygen Sensor Monitor Bank 2 - Sensor 2
07	Oxygen Sensor Monitor Bank 2 - Sensor 3
08	Oxygen Sensor Monitor Bank 2 - Sensor 4
09	Oxygen Sensor Monitor Bank 3 - Sensor 1
0A	Oxygen Sensor Monitor Bank 3 - Sensor 2
0B	Oxygen Sensor Monitor Bank 3 - Sensor 3
0C	Oxygen Sensor Monitor Bank 3 - Sensor 4
0D	Oxygen Sensor Monitor Bank 4 - Sensor 1
0E	Oxygen Sensor Monitor Bank 4 - Sensor 2
0F	Oxygen Sensor Monitor Bank 4 - Sensor 3
10	Oxygen Sensor Monitor Bank 4 - Sensor 4
11 - 1F	Reserved by document for future standardisation
20	OBD Monitor IDs supported (21h – 40h)
21	Catalyst Monitor Bank 1
22	Catalyst Monitor Bank 2
23	Catalyst Monitor Bank 3
24	Catalyst Monitor Bank 4
25 – 30	Reserved by document for future standardisation
31	EGR Monitor Bank 1

OBDMID (Hex)	On-Board Diagnostic Monitor ID name
32	EGR Monitor Bank 2
33	EGR Monitor Bank 3
34	EGR Monitor Bank 4
35 - 38	Reserved by document for future standardisation
39	EVAP Monitor (Cap Off)
3A	EVAP Monitor (0.090")
3B	EVAP Monitor (0.040”)
3C	EVAP Monitor (0.020”)
3D	Purge Flow Monitor
3E - 3F	Reserved by document for future standardisation
40	OBD Monitor IDs supported (41h – 60h)
41	Oxygen Sensor Heater Monitor Bank 1 - Sensor 1
42	Oxygen Sensor Heater Monitor Bank 1 - Sensor 2
43	Oxygen Sensor Heater Monitor Bank 1 - Sensor 3
44	Oxygen Sensor Heater Monitor Bank 1 - Sensor 4
45	Oxygen Sensor Heater Monitor Bank 2 - Sensor 1
46	Oxygen Sensor Heater Monitor Bank 2 - Sensor 2
47	Oxygen Sensor Heater Monitor Bank 2 - Sensor 3
48	Oxygen Sensor Heater Monitor Bank 2 - Sensor 4
49	Oxygen Sensor Heater Monitor Bank 3 - Sensor 1
4A	Oxygen Sensor Heater Monitor Bank 3 - Sensor 2
4B	Oxygen Sensor Heater Monitor Bank 3 - Sensor 3
4C	Oxygen Sensor Heater Monitor Bank 3 - Sensor 4
4D	Oxygen Sensor Heater Monitor Bank 4 - Sensor 1
4E	Oxygen Sensor Heater Monitor Bank 4 - Sensor 2
4F	Oxygen Sensor Heater Monitor Bank 4 - Sensor 3
50	Oxygen Sensor Heater Monitor Bank 4 - Sensor 4
51 - 5F	Reserved by document for future standardisation
60	OBD Monitor IDs supported (61h – 80h)
61	Heated Catalyst Monitor Bank 1
62	Heated Catalyst Monitor Bank 2
63	Heated Catalyst Monitor Bank 3
64	Heated Catalyst Monitor Bank 4
65 - 70	Reserved by document for future standardisation
71	Secondary Air Monitor 1
72	Secondary Air Monitor 2
73	Secondary Air Monitor 3
74	Secondary Air Monitor 4

OBDMID (Hex)	On-Board Diagnostic Monitor ID name
75 - 7F	Reserved by document for future standardisation
80	OBD Monitor IDs supported (81h - A0h)
81	Fuel System Monitor Bank 1
82	Fuel System Monitor Bank 2
83	Fuel System Monitor Bank 3
84	Fuel System Monitor Bank 4
85 - 9F	Reserved by document for future standardisation
A0	OBD Monitor IDs supported (A1h - C0h)
A1	Mis-Fire Monitor General Data
A2	Mis-Fire Cylinder 1 Data
A3	Mis-Fire Cylinder 2 Data
A4	Mis-Fire Cylinder 3 Data
A5	Mis-Fire Cylinder 4 Data
A6	Mis-Fire Cylinder 5 Data
A7	Mis-Fire Cylinder 6 Data
A8	Mis-Fire Cylinder 7 Data
A9	Mis-Fire Cylinder 8 Data
AA	Mis-Fire Cylinder 9 Data
AB	Mis-Fire Cylinder 10 Data
AC	Mis-Fire Cylinder 11 Data
AD	Mis-Fire Cylinder 12 Data
AE - BF	Reserved by document for future standardisation
C0	OBD Monitor IDs supported (C1h - E0h)
C1 - DF	Reserved by document for future standardisation
E0	OBD Monitor IDs supported (E1h - FFh)
E1 - FF	Vehicle Manufacturer defined OBDMIDs

54.24 Service 06h - Request on-board monitoring test results for specific monitored systems – OBD on CAN (Application Incident)

Input data:

FAC_DYN_LSL_DIAG_SAE[NC_CBK_EX_NR]	FAC_DYN_LSL_DIAG_TOL_SAE[NC_CBK_EX_NR]	LV_READY_XX	LV_ERR_MEM_XX
VLS_DOWN_MIN_DC[NC_CBK_EX_NR]	VLS_DOWN_MAX_DC[NC_CBK_EX_NR]	CTR_MIS_DC_MMV_CYL[NC_CYL_NR]	CTR_SAVE_SWT_LS_DOWN[NC_CBK_EX_NR]
C_CTR_SWT_LS_DOWN	CTR_MIS_DC_CYL[NC_CYL_NR]	CAT_DIAG_MOD_6[NC_CBK_EX_NR]	CAT_MAX_DIAG_MOD_6[NC_CBK_EX_NR]
M6_CUR_DMTL_ROUGH_LEAK_LEN_END	M6_CUR_DMTL_THD_ROUGH_LEAK_LEN	M6_CUR_DMTL_ROUGH_LEAK_END	M6_CUR_DMTL_THD_ROUGH_LEAK
M6_CTR_CNL_SMALL_LEAK_MES	C_SUM_CNL_SMALL_LEAK_MES_MAX	M6_CUR_DMTL_DMTLS_TEST	M6_CUR_DMTL_THD_DMTLS_TEST
M6_CUR_DMTL_COR_FIL_CID18	C_CUR_DMTL_REF_LEAK_MIN	C_CUR_DMTL_REF_LEAK_MAX	M6_CUR_DMTL_SMALL_LEAK_END
M6_CUR_DMTL_REF_LEAK	CL_MMV_SAE	C_CL_MMV_DIAGCPS	SUM_DIAG_DIAGCPS_SAE
C_SUM_DIAGCPS_MAX	MAF_DIAGCPS_SAE	MAF_DIAGCPS_THD_SAE	STATE_PSN_LS_SAE
STATE_EVAP_SAE	TTIP_MES_LS_UP_OBD_LSH_UP[NC_CBK_EX_NR]	C_TTIP_MIN_OBD_LSH_UP	C_TTIP_MAX_OBD_LSH_UP
R_IT_OBD_LSH_DOWN[NC_CBK_EX_NR]	R_IT_THD_OBD_LSH_DOWN[NC_CBK_EX_NR]	STATE_VAR_SAP_SAE	LC_SA_SWI_ACQ
SUM_AFL_VLS_DIAG_SA_1_SAE	SUM_AFL_VLS_DIAG_SA_2_SAE	C_SUM_AFL_VLS_MIN_DIAG_SA_1	SAF_DIAG_MIN_SAE
C_SAF_DIAG_MIN	SAF_DIAG_MAX_SAE	C_SAF_DIAG_MAX	CTR_MIS_TOT_DC
C_VLS_DOWN_AFL_AFR_SAE	C_VLS_DOWN_AFR_AFL_SAE	C_VLS_THD_DIAG_SCP_LS_DOWN	STATE_PSN_LS_1_SAE

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_STATE_SRV_6_TID	NC_NR_TID x 1	0...FFH	0...255	1	[-]
LDP_STATE_TID_ID_STATE_SRV_6	NC_NR_TID	0...FFH	0...255	1	[-]
Table of bytes which defines the TIDs supported by the service 06h – ISO 15765-4					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_NR_TID	1	0...FFH	0...255	1	[-]
Define how many TIDs are supported by the project					

FUNCTION DESCRIPTION:

General information:

In the table **ID_STATE_SRV_6_TID** the configured OBDMID's and TID's can be calibrated.

54.24.1 Configuration for supported OBDMID's

Description:

If an OBDMID is supported, then in column "Supported" "1" is configured, if not "0" is configured.

OBDMID (HEX)	Function	Supported
01	Oxygen Sensor Monitor Bank 1 - Sensor 1	**1**
02	Oxygen Sensor Monitor Bank 1 - Sensor 2	**1**
03	Oxygen Sensor Monitor Bank 1 - Sensor 3	0
04	Oxygen Sensor Monitor Bank 1 - Sensor 4	0
05	Oxygen Sensor Monitor Bank 2 - Sensor 1	**1**
06	Oxygen Sensor Monitor Bank 2 - Sensor 2	**1**
07	Oxygen Sensor Monitor Bank 2 - Sensor 3	0
08	Oxygen Sensor Monitor Bank 2 - Sensor 4	0
09	Oxygen Sensor Monitor Bank 3 - Sensor 1	0
0A	Oxygen Sensor Monitor Bank 3 - Sensor 2	0
0B	Oxygen Sensor Monitor Bank 3 - Sensor 3	0
0C	Oxygen Sensor Monitor Bank 3 - Sensor 4	0
0D	Oxygen Sensor Monitor Bank 4 - Sensor 1	0
0E	Oxygen Sensor Monitor Bank 4 - Sensor 2	0
0F	Oxygen Sensor Monitor Bank 4 - Sensor 3	0
10	Oxygen Sensor Monitor Bank 4 - Sensor 4	0
11 - 1F	**ISO/SAE reserved**	0
20	**OBD Monitor IDs supported ($21 - $40)**	**1**
21	Catalyst Monitor Bank 1	**1**
22	Catalyst Monitor Bank 2	**1**
23	Catalyst Monitor Bank 3	0
24	Catalyst Monitor Bank 4	0
25 – 30	**ISO/SAE reserved**	0
31	EGR Monitor Bank 1	0
32	EGR Monitor Bank 2	0
33	EGR Monitor Bank 3	0
34	EGR Monitor Bank 4	0
35 - 38	**ISO/SAE reserved**	0
39	EVAP Monitor (Cap Off)	0
3A	EVAP Monitor (0.090")	0
3B	EVAP Monitor (0.040”)	**1**
3C	EVAP Monitor (0.020”)	**1**
3D	Purge Flow Monitor	**1**
3E - 3F	**ISO/SAE reserved**	0
40	**OBD Monitor IDs supported ($41 - $60)**	**1**

OBDMID (HEX)	Function	Supported
41	Oxygen Sensor Heater Monitor Bank 1 - Sensor 1	**1**
42	Oxygen Sensor Heater Monitor Bank 1 - Sensor 2	**1**
43	Oxygen Sensor Heater Monitor Bank 1 - Sensor 3	0
44	Oxygen Sensor Heater Monitor Bank 1 - Sensor 4	0
45	Oxygen Sensor Heater Monitor Bank 2 - Sensor 1	**1**
46	Oxygen Sensor Heater Monitor Bank 2 - Sensor 2	**1**
47	Oxygen Sensor Heater Monitor Bank 2 - Sensor 3	0
48	Oxygen Sensor Heater Monitor Bank 2 - Sensor 4	0
49	Oxygen Sensor Heater Monitor Bank 3 - Sensor 1	0
4A	Oxygen Sensor Heater Monitor Bank 3 - Sensor 2	0
4B	Oxygen Sensor Heater Monitor Bank 3 - Sensor 3	0
4C	Oxygen Sensor Heater Monitor Bank 3 - Sensor 4	0
4D	Oxygen Sensor Heater Monitor Bank 4 - Sensor 1	0
4E	Oxygen Sensor Heater Monitor Bank 4 - Sensor 2	0
4F	Oxygen Sensor Heater Monitor Bank 4 - Sensor 3	0
50	Oxygen Sensor Heater Monitor Bank 4 - Sensor 4	0
51 - 5F	**ISO/SAE reserved**	0
60	**OBD Monitor IDs supported ($61 - $80)**	**1**
61	Heated Catalyst Monitor Bank 1	0
62	Heated Catalyst Monitor Bank 2	0
63	Heated Catalyst Monitor Bank 3	0
64	Heated Catalyst Monitor Bank 4	0
65 - 70	**ISO/SAE reserved**	0
71	Secondary Air Monitor 1	**1**
72	Secondary Air Monitor 2	**1**
73	Secondary Air Monitor 3	0
74	Secondary Air Monitor 4	0
75 - 7F	**ISO/SAE reserved**	0
80	**OBD Monitor IDs supported ($81 - $A0)**	**1**
81	Fuel System Monitor Bank 1	0
82	Fuel System Monitor Bank 2	0
83	Fuel System Monitor Bank 3	0
84	Fuel System Monitor Bank 4	0
85 - 9F	**ISO/SAE reserved**	0
A0	**OBD Monitor IDs supported ($A1 - $C0)**	**1**
A1	Mis-Fire Monitor General Data	**1**
A2	Mis-Fire Cylinder 1 Data	**1**
A3	Mis-Fire Cylinder 2 Data	**1**
A4	Mis-Fire Cylinder 3 Data	**1**
A5	Mis-Fire Cylinder 4 Data	**1**
A6	Mis-Fire Cylinder 5 Data	**1**

OBDMID (HEX)	Function	Supported
A7	Mis-Fire Cylinder 6 Data	**1**
A8	Mis-Fire Cylinder 7 Data	0
A9	Mis-Fire Cylinder 8 Data	0
AA	Mis-Fire Cylinder 9 Data	0
AB	Mis-Fire Cylinder 10 Data	0
AC	Mis-Fire Cylinder 11 Data	0
AD	Mis-Fire Cylinder 12 Data	0
AE - BF	**ISO/SAE reserved**	0
C0	**OBD Monitor IDs supported ($C1 - $E0)**	0
C1 - DF	**ISO/SAE reserved**	0
E0	**OBD Monitor IDs supported ($E1 - $FF)**	0
E1 - FF	**Vehicle manufacturer defined OBDMIDs**	0

54.24.2 Configuration of calibratable TID's

Description:

The table ID_STATE_SRV_6_TID will be filled with calibrated TID values. Each position in the table will have an associated TID and OBDMID.

- If the calibration of the TID = xy => TID is sended with xy
- If the calibration of the TID = 0 => TID is not supported
- If the calibration of all TID's of the OBDMID = 0 => OBDMID is not supported

Example:

If for instance in a project will be supported the following TIDs:
TIDs 01h, 02h, 03h, 04h, 05h, 81h, 82h, 90h, 98h, C1h

=> configure NC_NR_TID = 10
(LDP_STATE_TID_ID_STATE_SRV_6 automatically is set to 10)
=> ID_STATE_SRV_6_TID will have the dimension 10.

Configuration of the table ID_STATE_SRV_6_TID:

pos val	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]	x[9]
	01h	**02h**	**03h**	**02h**	**05h**	**81h**	**82h**	**81h**	**98h**	**C1h**
	OBDMID yy			**OBDMID zz**				**OBDMID vv**		

Each position in table will have an associated OBDMID:
Postion 0: TID 01h=> for OBDMID yy
Postion 1: TID 02h=> for OBDMID yy

Postion 2: TID 03h=> for OBDMID yy

Postion 3: TID 02h=> for OBDMID zz
Postion 4: TID 05h=> for OBDMID zz
Postion 5: TID 81h=> for OBDMID zz
Postion 6: TID 82h=> for OBDMID zz
Postion 7: TID 81h=> for OBDMID vv
Postion 8: TID 98h => for OBDMID vv
Postion 9: TID C1h => for OBDMID vv

Each OBDMID can be disabled separately:
e.g. if at position x[0], x[1], x[2] configure value 00h => OBDMID yy is **not supported**.

Formula section:

```
IF      ALL configured TIDs for the same OBDMID have value 00h
THEN    OBDMID IS NOT SUPPORTED
ELSE    OBDMID IS SUPPORTED
ENDIF
```

54.24.3 Configuration of table-size

Description:

The table-size of ID_STATE_SRV_6_TID depends on number of defined TID's (= NC_NR_TID), see Tables 1.4 ff.

Formula section:

TID's are defined from position 0...42

=> NC_NR_TID = **43**

54.24.4 Description of OBDMID 01h - 06h (Lambda sensor diagnosis)

General information:

Additional to the calibration the support of the OBDMID's can be learnt. Depending on the learning result (see STATE_PSN_LS_SAE and STATE_PSN_LS_1_SAE) the OBDMID 01h – 06h are supported or not.

54.24.4.1 Description of OBDMID 01h - Oxygen Sensor Monitor Bank 1 - Sensor 1

Formula section:

```
IF (STATE_PSN_LS_SAE & 01h) OR (STATE_PSN_LS_1_SAE & 01h)
THEN OBDMID 01h IS SUPPORTED
ELSE OBDMID 01h IS NOT SUPPORTED
ENDIF
```

Dynamic diagnosis – TID 83	
Position in table	[0]
Unit and scaling	83h (Raw value 8000...7FFFh / -327.68...327.67)
Test value	FAC_DYN_LSL_DIAG_SAE[1]
Min test limit	0 = 0000h
Max test limit	FAC_DYN_LSL_DIAG_TOL_SAE[1] (e.g. 0.9)

Formula section:

```
IF      ID_STATE_SRV_6_TID[0] <> 0
THEN    TID is supported
    IF      LV_READY_DYN_VLD_LS_UP[1] = 0      OR
            LV_ERR_MEM_DYN_VLD_LS_UP[1] = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID is not supported
ENDIF
```

54.24.4.2 Description of OBDMID 02h - Oxygen Sensor Monitor Bank 1 - Sensor 2

Formula section:

```
IF (STATE_PSN_LS_SAE & 02h) OR (STATE_PSN_LS_1_SAE & 02h)
THEN OBDMID 02h IS SUPPORTED
ELSE OBDMID 02h IS NOT SUPPORTED
ENDIF
```

Rich to lean sensor threshold voltage – TID 01	
Position in table	[1]
Unit and scaling	0Ah (Voltage 0...FFFFH / 0...8V)
Test value	C_VLS_DOWN_AFR_AFL_SAE
Min test limit	0V = 0000h
Max test limit	C_VLS_THD_DIAG_SCP_LS_DOWN
Lean to rich sensor threshold voltage – TID 02	
Position in table	[2]
Unit and scaling	0Ah (Voltage 0...FFFFH / 0...8V)
Test value	C_VLS_DOWN_AFL_AFR_SAE
Min test limit	0V = 0000h
Max test limit	C_VLS_THD_DIAG_SCP_LS_DOWN
Low sensor voltage for switch time calculation – TID 07	
Position in table	[3]
Unit and scaling	0Ah (Voltage 0...FFFFH / 0...8V)
Test value	VLS_DOWN_MIN_DC[1]
Min test limit	0V = 0000h
Max test limit	C_VLS_THD_DIAG_SCP_LS_DOWN
High sensor voltage for switch time calculation – TID 08	
Position in table	[4]
Unit and scaling	0Ah (Voltage 0...FFFFH / 0...8V)
Test value	VLS_DOWN_MAX_DC[1]
Min test limit	0V = 0000h
Max test limit	C_VLS_THD_DIAG_SCP_LS_DOWN
Switching time diagnosis – TID 91	
Position in table	[5]
Unit and scaling	05h (Raw value 0...FFFFh / 0...2)
Test value	CTR_SAVE_SWT_LS_DOWN[1]
Min test limit	0 = 0000h
Max test limit	C_CTR_SWT_LS_DOWN (e.g. 1)

Formula section:

```
IF      ID_STATE_SRV_6_TID[1] / [2] / [3] / [4] / [5] <> 0      //each seperate
THEN    TID is supported
    IF      LV_READY_SWT_LS_DOWN[1] = 0                 OR
            LV_ERR_MEM_SWT_LS_DOWN[1] = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID is not supported
ENDIF
```

54.24.4.3 Description of OBDMID 05h - Oxygen Sensor Monitor Bank 2 - Sensor 1

Formula section:

```
IF (STATE_PSN_LS_SAE & 10h) OR (STATE_PSN_LS_1_SAE & 04h)
THEN OBDMID 05h IS SUPPORTED
ELSE OBDMID 05h IS NOT SUPPORTED
```

```
ENDIF
```

Dynamic diagnosis – TID 83	
Position in table	[6]
Unit and scaling	83h (Raw value 8000...7FFFh / -327.68...327.67)
Test value	FAC_DYN_LSL_DIAG_SAE[2]
Min test limit	0 = 0000h
Max test limit	FAC_DYN_LSL_DIAG_TOL_SAE[2] (e.g. 0.9)

Formula section:

```
IF      ID_STATE_SRV_6_TID[6] <> 0
THEN    TID is supported
    IF      LV_READY_DYN_VLD_LS_UP[2] = 0    OR
            LV_ERR_MEM_DYN_VLD_LS_UP[2] = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID is not supported
ENDIF
```

54.24.4.4 Description of OBDMID 06h - Oxygen Sensor Monitor Bank 2 - Sensor 2

Formula section:

```
IF (STATE_PSN_LS_SAE & 20h) OR (STATE_PSN_LS_1_SAE & 08h)
THEN OBDMID 06h IS SUPPORTED
ELSE OBDMID 06h IS NOT SUPPORTED
ENDIF
```

Rich to lean sensor threshold voltage – TID 01	
Position in table	[7]
Unit and scaling	0Ah (Voltage 0...FFFFH / 0...8V)
Test value	C_VLS_DOWN_AFR_AFL_SAE
Min test limit	0V = 0000h
Max test limit	C_VLS_THD_DIAG_SCP_LS_DOWN
Lean to rich sensor threshold voltage – TID 02	
Position in table	[8]
Unit and scaling	0Ah (Voltage 0...FFFFH / 0...8V)
Test value	C_VLS_DOWN_AFL_AFR_SAE
Min test limit	0V = 0000h
Max test limit	C_VLS_THD_DIAG_SCP_LS_DOWN
Low sensor voltage for switch time calculation – TID 07	
Position in table	[9]
Unit and scaling	0Ah (Voltage 0...FFFFH / 0...8V)
Test value	VLS_DOWN_MIN_DC[2]
Min test limit	0V = 0000h
Max test limit	C_VLS_THD_DIAG_SCP_LS_DOWN
High sensor voltage for switch time calculation – TID 08	

Position in table	[10]
Unit and scaling	0Ah (Voltage 0...FFFFH / 0...8V)
Test value	VLS_DOWN_MAX_DC[2]
Min test limit	0V = 0000h
Max test limit	C_VLS_THD_DIAG_SCP_LS_DOWN
Switching time diagnosis – TID 91	
Position in table	[11]
Unit and scaling	05h (Raw value 0...FFFFh / 0...2)
Test value	CTR_SAVE_SWT_LS_DOWN[2]
Min test limit	0 = 0000h
Max test limit	C_CTR_SWT_LS_DOWN (e.g. 1)

Formula section:

```
IF      ID_STATE_SRV_6_TID[7] / [8] / [9] / [10] / [11] <> 0        //each seperate
THEN    TID is supported
    IF      LV_READY_SWT_LS_DOWN[2] = 0        OR
            LV_ERR_MEM_SWT_LS_DOWN[2] = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID is not supported
ENDIF
```

54.24.5 Description of OBDMID 21 - 22h (Catalyst diagnosis)

General information:

Additional to the calibration the support of the OBDMID's can be learnt. Depending on the learning result (see STATE_PSN_LS_SAE and STATE_PSN_LS_1_SAE) the OBDMID 21h – 22h are supported or not.

54.24.5.1 Description of OBDMID 21h - Catalyst Monitor Bank 1

Formula section:

```
IF (STATE_PSN_LS_SAE & 03h) OR (STATE_PSN_LS_1_SAE & 03h)
THEN OBDMID 21h IS SUPPORTED
ELSE OBDMID 21h IS NOT SUPPORTED
ENDIF
```

OSC method (linear system) – TID 81	
Position in table	[12]
Unit and scaling	05h (Raw value 0...FFFFh / 0...2)
Test value	CAT_DIAG_MOD_6[1]
Min test limit	0 = 0000h
Max test limit	CAT_MAX_DIAG_MOD_6[1] (e.g. 0.8)

Formula section:

```
IF        ID_STATE_SRV_6_TID[12] <> 0
THEN      TID is supported
    IF        LV_READY_CAT_DIAG[1] = 0          OR
              LV_ERR_MEM_CAT_DIAG[1] = 1
    THEN      Positive response with test result
    ELSE      Positive response with all data equal to 0000h
    ENDIF
ELSE      TID is not supported
ENDIF
```

54.24.5.2 Description of OBDMID 22h - Catalyst Monitor Bank 2

Formula section:

```
IF (STATE_PSN_LS_SAE & 30h) OR (STATE_PSN_LS_1_SAE & 0Ch)
THEN OBDMID 22h IS SUPPORTED
ELSE OBDMID 22h IS NOT SUPPORTED
ENDIF
```

OSC method (linear system) – TID 81	
Position in table	[13]
Unit and scaling	05h (Raw value 0...FFFFh / 0...2)
Test value	CAT_DIAG_MOD_6[2]
Min test limit	0 = 0000h
Max test limit	CAT_MAX_DIAG_MOD_6[2] (e.g. 0.8)

Formula section:

```
IF        ID_STATE_SRV_6_TID[13] <> 0
THEN      TID is supported
    IF        LV_READY_CAT_DIAG[2] = 0          OR
              LV_ERR_MEM_CAT_DIAG[2] = 1
    THEN      Positive response with test result
    ELSE      Positive response with all data equal to 0000h
    ENDIF
ELSE      TID is not supported
ENDIF
```

54.24.6 Description of OBDMID 3Bh – 3Dh (EVAP diagnosis)

General information:

Additional to the calibration the support of the OBDMID's can be learnt. OBDMID 3Bh – 3Ch are only supported if the vehicle is equipped with DMTL module.

Formula section:

```
IF      STATE_EVAP_SAE <> 0
THEN    OBDMID 3Bh – 3Ch are supported
ELSE    OBDMID 3Bh – 3Ch are not supported
ENDIF
```

54.24.6.1 Description of OBDMID 3B - EVAP Monitor Large (0.040" / 1.0mm)

Leak detection DMTL – Long cycle – TID 81	
Position in table	[14]
Unit and scaling	0Dh (Current 0...FFFFh / 0...256 mA)
Test value	M6_CUR_DMTL_ROUGH_LEAK_LEN_END
Min test limit	M6_CUR_DMTL_THD_ROUGH_LEAK_LEN
Max test limit	FFFFh (256 mA)
Leak detection DMTL – Short cycle – TID 82	
Position in table	[15]
Unit and scaling	0Dh (Current)
Test value	M6_CUR_DMTL_ROUGH_LEAK_END
Min test limit	M6_CUR_DMTL_THD_ROUGH_LEAK
Max test limit	FFFFh (256 mA)

Formula section:

```
IF      ID_STATE_SRV_6_TID[14] / [15] <> 0          //each seperate
THEN    TID's are supported
        IF      LV_READY_ROUGH_LEAK = 0             OR
                LV_ERR_MEM_ROUGH_LEAK = 1
        THEN    Positive response with test result
        ELSE    Positive response with all data equal to 0000h
        ENDIF
ELSE    TID' s are not supported
ENDIF
```

DMTL module - Signal error – TID 8B	
Position in table	[16]
Unit and scaling	24h (Counts 0...FFFFh / 0....65535)
Test value	M6_CTR_CNL_SMALL_LEAK_MES
Min test limit	0h
Max test limit	C_SUM_CNL_SMALL_LEAK_MES_MAX
DMTL module – Plausibility error – TID 8C	
Position in table	[17]
Unit and scaling	0Dh (Current 0...FFFFh / 0...256 mA)

Test value	M6_CUR_DMTL_DMTLS_TEST
Min test limit	0h
Max test limit	M6_CUR_DMTL_THD_DMTLS_TEST
DMTL module – MIN / MAX error – TID 8D	
Position in table	[18]
Unit and scaling	0Dh (Current 0...FFFFh / 0...256 mA)
Test value	M6_CUR_DMTL_COR_FIL_CID18
Min test limit	C_CUR_DMTL_REF_LEAK_MIN
Max test limit	C_CUR_DMTL_REF_LEAK_MAX

Formula section:

```
IF          ID_STATE_SRV_6_TID[16] / [17] / [18] <> 0          //each seperate
THEN        TID's are supported
    IF          LV_READY_DMTLM = 0              OR
                LV_ERR_MEM_DMTLM = 1
    THEN        Positive response with test result
    ELSE        Positive response with all data equal to 0000h
    ENDIF
ELSE        TID's are not supported
ENDIF
```

54.24.6.2 Description of OBDMID 3C - EVAP Monitor Small (0.020" / 0.5 mm)

Leak detection DMTL – Short cycle – TID 3C	
Position in table	[19]
Unit and scaling	0Dh (Current 0...FFFFh / 0...256 mA)
Test value	M6_CUR_DMTL_SMALL_LEAK_END
Min test limit	M6_CUR_DMTL_REF_LEAK
Max test limit	FFFFh (256 mA)

Formula section:

```
IF          ID_STATE_SRV_6_TID[19] <> 0
THEN        TID is supported
    IF          LV_READY_SMALL_LEAK = 0              OR
                LV_ERR_MEM_SMALL_LEAK = 1
    THEN        Positive response with test result
    ELSE        Positive response with all data equal to 0000h
    ENDIF
ELSE        TID is not supported
ENDIF
```

54.24.6.3 Description of OBDMID 3D - EVAP Purge Flow monitor

Functional check TEV – Step 1 – TID 81

Position in table	[20]
Unit and scaling	05h (Raw value 0...FFFFh / 0...2)
Test value	CL_MMV_SAE
Min test limit	C_CL_MMV_DIAGCPS
Max test limit	FFFFh (2)
Functional check TEV – Step 2 – TID 82	
Position in table	[21]
Unit and scaling	03h (Raw value 0....FFFFh / 0...655,35)
Test value	SUM_DIAG_DIAGCPS_SAE
Min test limit	0h(0)
Max test limit	C_SUM_DIAGCPS_MAX
Functional check TEV – Step 3 – TID 83	
Position in table	[22]
Unit and scaling	2Dh (Weight per stroke 0...FFFFh / 0...655,35 mg/stk)
Test value	MAF_DIAGCPS_SAE
Min test limit	0h (0 mg/stk)
Max test limit	MAF_DIAGCPS_THD_SAE

Formula section:

```
IF          ID_STATE_SRV_6_TID[20] / [21] / [22] <> 0          //each seperate
THEN        TID's are supported
     IF             LV_READY_DIAGCPS = 0           OR
                    LV_ERR_MEM_DIAGCPS = 1
     THEN           Positive response with test result
     ELSE           Positive response with all data equal to 0000h
     ENDIF
ELSE        TID's are not supported
ENDIF
```

54.24.7 Description of OBDMID 41h – 46h (Oxygen sensor heater diagnosis)

General information:

Additional to the calibration the support of the OBDMID's can be learnt. Depending on the learning result (see STATE_PSN_LS_SAE and STATE_PSN_LS_1_SAE) the OBDMID 41h – 46h are supported or not.

54.24.7.1 Description of OBDMID 41h Oxygen Sensor Heater Monitor Bank 1 Sensor 1

Formula section:

```
IF (STATE_PSN_LS_SAE & 01h) OR (STATE_PSN_LS_1_SAE & 01h)
THEN OBDMID 41h IS SUPPORTED
ELSE OBDMID 41h IS NOT SUPPORTED
ENDIF
```

Sensor temperature – TID 85	
Position in table	[23]
Unit and scaling	96h (Temperature FE6F...4E20 / -3276,8...3276,7°C)

Test value	TTIP_MES_LS_UP_OBD_LSH_UP[1]
Min test limit	C_TTIP_MIN_OBD_LSH_UP
Max test limit	C_TTIP_MAX_OBD_LSH_UP

Formula section:

```
IF          ID_STATE_SRV_6_TID[23] <> 0
THEN        TID is supported
    IF          LV_READY_OBD_VLD_LSH_UP[1] = 0          OR
                LV_ERR_MEM_OBD_VLD_LSH_UP[1] = 1
    THEN        Positive response with test result
    ELSE        Positive response with all data equal to 0000h
    ENDIF
ELSE        TID is not supported
ENDIF
```

54.24.7.2 Description of OBDMID 42h Oxygen Sensor Heater Monitor Bank 1 Sensor 2

Formula section:

```
IF (STATE_PSN_LS_SAE & 02h) OR (STATE_PSN_LS_1_SAE & 02h)
THEN OBDMID 42h IS SUPPORTED
ELSE OBDMID 42h IS NOT SUPPORTED
ENDIF
```

Sensor resistance – TID 81	
Position in table	[24]
Unit and scaling	01h (Raw value 0...FFFFh / 0...65535)
Test value	R_IT_OBD_LSH_DOWN[1]
Min test limit	0h
Max test limit	R_IT_THD_OBD_LSH_DOWN[1]

Formula section:

```
IF          ID_STATE_SRV_6_TID[24] <> 0
THEN        TID is supported
    IF          LV_READY_OBD_LSH_DOWN[1] = 0          OR
                LV_ERR_MEM_OBD_LSH_DOWN[1] = 1
    THEN        Positive response with test result
    ELSE        Positive response with all data equal to 0000h
    ENDIF
ELSE        TID is not supported
ENDIF
```

54.24.7.3 Description of OBDMID 45h Oxygen Sensor Heater Monitor Bank 2 Sensor 1

Formula section:

```
IF (STATE_PSN_LS_SAE & 10h) OR (STATE_PSN_LS_1_SAE & 04h)
THEN OBDMID 45h IS SUPPORTED
ELSE OBDMID 45h IS NOT SUPPORTED
ENDIF
```

Sensor temperature – TID 85	
Position in table	[25]
Unit and scaling	96h (Temperature FE6F...4E20 / -3276,8...3276,7°C)
Test value	TTIP_MES_LS_UP_OBD_LSH_UP[2]
Min test limit	C_TTIP_MIN_OBD_LSH_UP
Max test limit	C_TTIP_MAX_OBD_LSH_UP

Formula section:

```
IF      ID_STATE_SRV_6_TID[25] <> 0
THEN    TID is supported
    IF      LV_READY_OBD_VLD_LSH_UP[2] = 0      OR
            LV_ERR_MEM_OBD_VLD_LSH_UP[2] = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID is not supported
ENDIF
```

54.24.7.4 Description of OBDMID 46h Oxygen Sensor Heater Monitor Bank 2 Sensor 2

Formula section:

```
IF (STATE_PSN_LS_SAE & 20h) OR (STATE_PSN_LS_1_SAE & 08h)
THEN OBDMID 46h IS SUPPORTED
ELSE OBDMID 46h IS NOT SUPPORTED
ENDIF
```

Sensor resistance – TID 81	
Position in table	[26]
Unit and scaling	01h (Raw value 0...FFFFh / 0...65535)
Test value	R_IT_OBD_LSH_DOWN[2]
Min test limit	0h
Max test limit	R_IT_THD_OBD_LSH_DOWN[2]

Formula section:

```
IF      ID_STATE_SRV_6_TID[26] <> 0
THEN    TID is supported
```

```
        IF          LV_READY_OBD_LSH_DOWN[2] = 0          OR
                    LV_ERR_MEM_OBD_LSH_DOWN[2] = 1
        THEN        Positive response with test result
        ELSE        Positive response with all data equal to 0000h
        ENDIF
    ELSE        TID is not supported
    ENDIF
```

54.24.8 Description of OBDMID 71h – 72h (Secondary air diagnosis)

General information:

Additional to the calibration the support of the OBDMID's can be learnt. OBDMID 71h – 72h are only supported if the vehicle is equipped with SA pump. Depending on Hardware configuration only one monitor is active (via lambda-sensor **or** flow meter).

Formula section:

```
IF      STATE_VAR_SAP_SAE <> 0
THEN    OBDMID 71h – 72h are supported
    IF      LC_SA_SWI_ACQ = 1
    THEN    OBDMID 71h is not supported          //SA flow meter equipped
    ELSE    OBDMID 72h is not supported          //SA flow meter not equipped
    ENDIF
ELSE    OBDMID 71h – 72h are not supported
ENDIF
```

54.24.8.1 Description of OBDMID 71h – Secondary air monitor with lambda sensor

Monitoring of minimum flow rate – TID 81	
Position in table	[27]
Unit and scaling	01h (Raw value 0...FFFFh / 0...65535)
Test value	MIN [SUM_AFL_VLS_DIAG_SA_1_SAE, SUM_AFL_VLS_DIAG_SA_2_SAE]
Min test limit	C_SUM_AFL_VLS_MIN_DIAG_SA_1
Max test limit	FFFFh (65535)

Formula section:

```
IF      ID_STATE_SRV_6_TID[27] <> 0
THEN    TID is supported
    IF      LV_READY_SA_SAP = 0          OR
            LV_ERR_MEM_SA_SAP = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID is not supported
ENDIF
```

54.24.8.2 Description of OBDMID 72h – Secondary air monitor with SA flow meter

Monitoring of minimum flow rate – TID 82	
Position in table	[28]
Unit and scaling	27h (Weight per time 0...FFFFh / 0...655,35 g/s)
Test value	SAF_DIAG_MIN_SAE
Min test limit	0h

Max test limit	C_SAF_DIAG_MIN
Monitoring of minimum flow rate – TID 83	
Position in table	[29]
Unit and scaling	27h (Weight per time 0...FFFFh / 0...655,35 g/s)
Test value	SAF_DIAG_MAX_SAE
Min test limit	0h
Max test limit	C_SAF_DIAG_MAX

Formula section:

```
IF      ID_STATE_SRV_6_TID[28] / [29] <> 0          //each seperate
THEN    TID are supported
    IF          LV_READY_SA_SYS = 0          OR
                LV_ERR_MEM_SA_SYS = 1
    THEN        Positive response with test result
    ELSE        Positive response with all data equal to 0000h
    ENDIF
ELSE    TID are not supported
ENDIF
```

54.24.9 Description of OBDMID A1h - Mis-fire Monitor General Data

Mis-Fire Monitor General Data – TID 0C	
Position in table	[30]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_TOT_DC
Min test limit	0
Max test limit	65535 = FFFFh

Formula section:

```
IF      ID_STATE_SRV_6_TID[30] <> 0           //each seperate
THEN    TID is supported
    IF      LV_READY_MIS[0] = 0          OR
            LV_ERR_MEM_MIS[0] = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID is not supported
ENDIF
```

54.24.10 Description of OBDMID A2h Mis-fire Clinder 1 Data

Mis-Fire Count for Last/Current driving cycle – TID 0C	
Position in table	[31]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_CYL[0]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh
EWMA Mis-Fire Count for Last 10 driving cycle – TID 0B	
Position in table	[32]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_MMV_CYL[0]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh

Formula section:

```
IF      ID_STATE_SRV_6_TID[31] / [32] <> 0           //each seperate
THEN    TID's are supported
    IF      LV_READY_MIS[0] = 0          OR
            LV_ERR_MEM_MIS[0] = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID's are not supported
ENDIF
```

54.24.11 Description of OBDMID A3h - Mis-fire Cylinder 2 Data

Mis-Fire Count for Last/Current driving cycle – TID 0C	
Position in table	[33]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_CYL[4]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh
EWMA Mis-Fire Count for Last 10 driving cycle – TID 0B	
Position in table	[34]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_MMV_CYL[4]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh

Formula section:

```
IF      ID_STATE_SRV_6_TID[33] / [34] <> 0        //each seperate
THEN    TID's are supported
    IF      LV_READY_MIS[4] = 0        OR
            LV_ERR_MEM_MIS[4] = 1
    THEN    Positive response with test result
    ELSE    Positive response with all data equal to 0000h
    ENDIF
ELSE    TID's are not supported
ENDIF
```

54.24.12 Description of OBDMID A4h - Mis-fire Cylinder 3 Data

Mis-Fire Count for Last/Current driving cycle – TID 0C	
Position in table	[35]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_CYL[2]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh
EWMA Mis-Fire Count for Last 10 driving cycle – TID 0B	
Position in table	[36]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_MMV_CYL[2]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh

Formula section:

```
IF      ID_STATE_SRV_6_TID[35] / [36] <> 0        //each seperate
THEN    TID's are supported
    IF      LV_READY_MIS[2] = 0        OR
            LV_ERR_MEM_MIS[2] = 1
```

```
        THEN      Positive response with test result
        ELSE      Positive response with all data equal to 0000h
        ENDIF
ELSE      TID's are not supported
ENDIF
```

54.24.13 Description of OBDMID A5h - Mis-fire Cylinder 4 Data

Mis-Fire Count for Last/Current driving cycle – TID 0C	
Position in table	[37]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_CYL[5]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh
EWMA Mis-Fire Count for Last 10 driving cycle – TID 0B	
Position in table	[38]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_MMV_CYL[5]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh

Formula section:

```
IF        ID_STATE_SRV_6_TID[37] / [38] <> 0          //each seperate
THEN      TID's are supported
        IF        LV_READY_MIS[5] = 0          OR
                  LV_ERR_MEM_MIS[5] = 1
        THEN      Positive response with test result
        ELSE      Positive response with all data equal to 0000h
        ENDIF
ELSE      TID's are not supported
ENDIF
```

54.24.14 Description of OBDMID A6h - Mis-fire Cylinder 5 Data

Mis-Fire Count for Last/Current driving cycle – TID 0C	
Position in table	[39]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_CYL[1]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh
EWMA Mis-Fire Count for Last 10 driving cycle – TID 0B	
Position in table	[40]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_MMV_CYL[1]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh

Formula section:

```
54.24.15    IF    ID_STATE_SRV_6_TID[39] / [40] <> 0   //each seperate
THEN        TID's are supported
    IF          LV_READY_MIS[1] = 0          OR
                LV_ERR_MEM_MIS[1] = 1
    THEN        Positive response with test result
    ELSE        Positive response with all data equal to 0000h
    ENDIF
ELSE        TID's are not supported
ENDIF
```

54.24.16 Description of OBDMID A7h - Mis-fire Cylinder 6 Data

Mis-Fire Count for Last/Current driving cycle – TID 0C	
Position in table	[41]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_CYL[3]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh
EWMA Mis-Fire Count for Last 10 driving cycle – TID 0B	
Position in table	[42]
Unit and scaling	24h (Counts 0...FFFFh / 0...65535)
Test value	CTR_MIS_DC_MMV_CYL[3]
Min test limit	0 =0000h
Max test limit	65535 = FFFFh

Formula section:

```
IF          ID_STATE_SRV_6_TID[41] / [42] <> 0          //each seperate
THEN        TID's are supported
    IF          LV_READY_MIS[3] = 0          OR
                LV_ERR_MEM_MIS[3] = 1
    THEN        Positive response with test result
    ELSE        Positive response with all data equal to 0000h
    ENDIF
ELSE        TID's are not supported
ENDIF
```

54.25 Service 07h - Request emission-related diagnostic trouble codes detected during current or last completed driving cycle – OBD on CAN

Import actions:

ACTION_ERRM_ReadDtcByTypeOfDtc	**(** **IN <TypeOfDtc>** **IN <LevelOfDtc>** **INOUT <ListOfDtc>** **OUT <ResultDtc>** **)**
This API returns a list of non confirmed DTC.	
ACTION_ERRM_ReadQuantityOfDtc	**(** **IN <TypeOfDtc>** **IN <LevelOfDtc>** **INOUT <Quantity>** **OUT <ResultQuantity>** **)**
This API returns the quantity of DTC with a certain type, which are stored in memory	

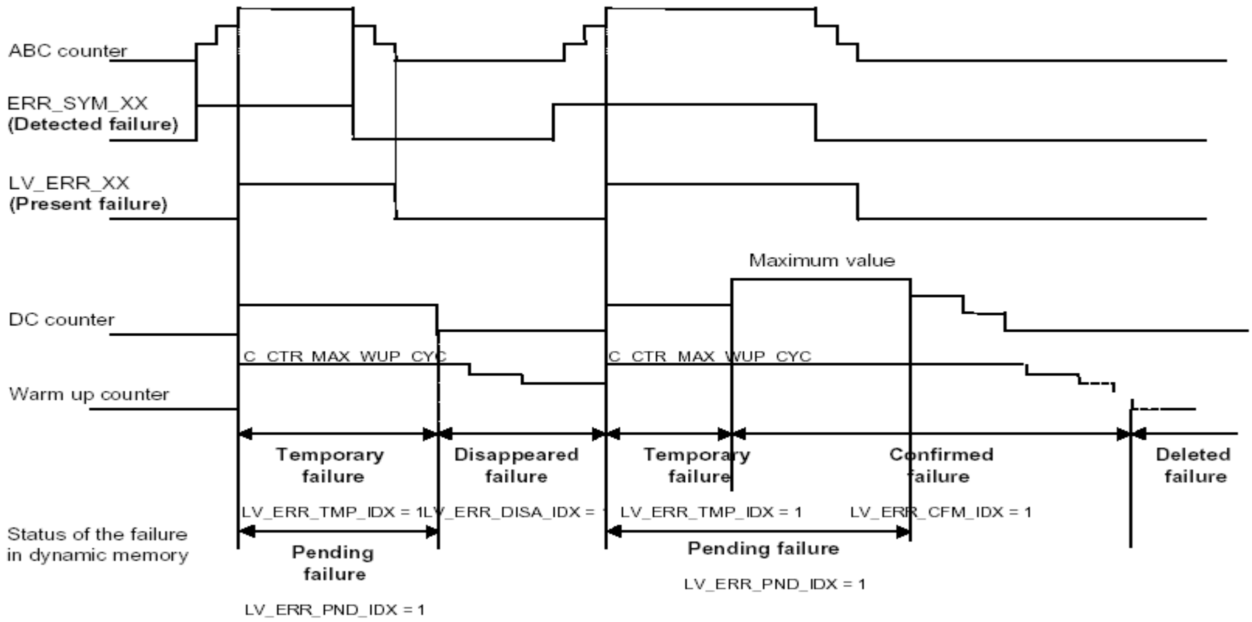
54.25.1 Functional description for ISO 15765-4

The purpose of this service is to enable the external test equipment to obtain “pending” diagnostic trouble codes detected during current or last completed driving cycle for emission-related components / systems that are tested or continuously monitored during normal driving conditions. Service 07h is required for all DTCs and is independent of Service 03h.

The intended use of this data is to assist the service technician after a vehicle repair, and after clearing diagnostic information, by reporting test results after a single driving cycle. If the test failed during the driving cycle, the DTC associated with that test will be reported. Test results reported by this service do not necessarily indicate a faulty component / system. If test results indicate a failure after additional driving, then the MIL will be illuminated and a DTC will be set and reported with service 03h, indicating a faulty component / system. This service can always be used to request the results of the latest test, independent of the setting of a DTC.

Test results for these components / systems are reported in the same format as the DTCs in Service $03 - refer to the functional description for service $03.

If an ECU does not have stored DTCs then it shall respond with a message indicating no codes are stored by setting the parameter "Number of DTC" to 00h.



54.25.2 Message data bytes

54.25.2.1 Request message definition

Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request message

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request SID**	**M**	**07**	**SIDRQ**

54.25.2.2 Response message definition

Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response message

Data Byte	Parameter Name	Cvt	Hex Value	Mnemonic
#1	**Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response SID**	**M**	**47**	**SIDPR**
#2	# of DTC = [no emission-related DTCs # of emission-related DTCs]	M	 00 01 - FF	#OFDTC
#3 #4	DTC#1 (High Byte) DTC#1 (Low Byte)	C C	xx xx	DTC1HI DTC1LO
:	:	:	xx	
#n-1 #n	DTC#m (High Byte) DTC#m (Low Byte)	C C	xx xx	DTCmHI DTCmLO
C = Conditional — DTC#1 - DTC#m are only included if # of DTC parameter value ≠ $00				

54.25.2.3 Formula section

```
ACTION_ERRM_ReadQuantityOfDtc
                                        (
                                        IN <OBD&PENDING>
                                        IN <LAW>
                                        INOUT <NR_DTC>
                                        OUT <ResultQuantity>
                                        )

ACTION_ERRM_ReadDtcByTypeOfDtc
                                        (
                                        IN <OBD&PENDING>
                                        IN <LAW>
                                        INOUT <DTC#1 to DTC#m>
                                        OUT <ResultDtc>
                                        )


IF      ResultDtc != BUFFER_FULL
        THEN
                Positive response
        ELSE
                No answer
ENDIF
```

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CKS_BOOT_1_SAE	O/S	0...FFFFFFFFH	0...4294967295	1	[-]
CRC for CBOOT					
STATE_CKS_BOOT_2_SAE	O/S	0...FFFFFFFFH	0...4294967295	1	[-]
CRC for SBOOT					
STATE_CKS_ECU_SAE	O/S	0...FFFFFFFFH	0...4294967295	1	[-]
CRC for ECU					
STATE_CKS_CAL_SAE	O/S	0...FFFFFFFFH	0...4294967295	1	[-]
CRC for CALIB					
STATE_CKS_SEL_SEG_SAE	O/S	0...FFFFFFFFH	0...4294967295	1	[-]
CRC for several or all segments of memory					

Import actions:

ACTION_OBDC_*GetVIN* (OUT <>)
This action fill the VIN with the correct values from the project
ACTION_OBDC_*GetCALIDCBoot* (OUT <>)
This action fill the CBOOT CALID with the correct values from the project
ACTION_OBDC_*GetCALIDSBoot* (OUT <>)
This action fill the SBOOT CALID with the correct values from the project
ACTION_OBDC_*GetCALIDECU* (OUT <>)
This action fill the ECU CALID with the correct values from the project
ACTION_OBDC_*GetCALIDCalib* (OUT <>)
This action fill the CALIB CALID with the correct values from the project
ACTION_OBDC_*GetCALIDSelSeg* (OUT <>)
This action fill the several or all segments of memory CALID with the correct values from the project
ACTION_OBDC_*GetECUNAME*(OUT <>)
This action fills the emission related ECU Name
ACTION_ERRM_SelectRbmData (INOUT <>, OUT <>)
This API calculates and returns the Rate-Based Monitoring data to be transmitted to the Scan-Tool, when requested by the Service 09$ (InfoType $08).
cks_service (IN <component>, IN<command>, OUT<status>)
This action performs CRC check control

Input data:

NC_STATE_CKS_CONF_CAN	NC_STATE_COMP_TYP_CAN	NC_STATE_SRV_9_CAN_VIT	

54.26.1 Functional description

The purpose of this service is to enable the off-board test device to request vehicle specific information such as Vehicle Identification Number (VIN) and calibration IDs.

54.26.1.1 Read supported Info Type

54.26.1.1.1 Request message

Bytes	Parameter name	CVT	Hex value

#1	**Request vehicle information**	M	09
#2	Info Type#1 (VITs supported : 01h - 20h)	M	00
#3	Info Type#2 (VITs supported : 21h - 40h)	U	20
#4	Info Type#3 (VITs supported : 41h - 60h)	U	40
#5	Info Type#4 (VITs supported : 61h - 80h)	U	60
#6	Info Type#5 (VITs supported : 81h - A0h)	U	80
#7	Info Type#6 (VITs supported : A1h - C0h)	U	A0

U = User optional	Info Type may be included to avoid multiple Info Type supported request messages

To request Info Types supported range from C1h to FFh another request message with Info Type#1 = C0h and Info Type#2 = E0h shall be sent to the vehicle.

54.26.1.1.2 Positive response message

Bytes	Parameter name	CVT	Hex value
#1	**Request vehicle information**	M	49
#2	Info Type = 1st supported Info Type	M	00
#4	Supported Info Type 01h - 08h	M	XX
#5	Supported Info Type 09h - 10h	M	XX
#6	Supported Info Type 11h - 18h	M	XX
#7	Supported Info Type 19h - 20h	M	XX
#n-4	Info Type = mth supported Info Type	C1	XX
#n-3	Supported Info Type m1h - m8h	C2	XX
#n-2	Supported Info Type m9h - (m+1)0h	C2	XX
#n-1	Supported Info Type (m+1)1h - (m+1)8h	C2	XX
#n	Supported Info Type (m+1)9h - (m+2)0h	C2	XX

C1 = Conditional	Info Type value shall be the same value as included in the request message if supported by the ECU
C2 = Conditional	Value indicates Info Type supported, the range of supported Info Type depends on selected Info Type value

The response message shall only include the Info Types and Data A - D which are supported by the ECU. If the request message includes (a) Info Type value(s) which are not supported by the ECU, those shall not be included in the response message.

54.26.1.2 Read Info Type values

54.26.1.2.1 Request message

Bytes	Parameter name	CVT	Hex value
#1	**Request vehicle information**	M	09
#2	Info Type (Read Info Type values)	M	02, 04, 06, 08, 0A

54.26.1.2.2 Positive response message

Bytes	Parameter name	CVT	Hex value
#1	**Request vehicle information**	M	49
#2	Info Type (Report Info Type values)	M/C1	02, 04, 06, 08, 0A
#3	Number of data items	M/C1	XX
#4	Data #1	M/C2	XX
#5	Data #2	C2	XX
#n	Data #m	C2	XX

C1 = Conditional	Info Type value shall be the same value as included in the request message
C2 = Conditional	Data #2 to Data #m depend on selected Info Type value

54.26.1.3 Negative response message

Bytes	Parameter name	CVT	Hex value
#1	**Negative response**	M	7F
#2	**Request emission-related powertrain diagnostic trouble codes**	M	09
#3	Error code = Request correctly received response pending	M	78

54.26.1.3.1 Error code

Error code	Cause
Request correctly received response pending	The on-board processor requires significant time to calculate the CRCs

54.26.2 Description of Service 09h, INFOTYPE 00h

DATA A - Supported INFOTYPEs 01h to 08h

INFOTYPE 00h - DATA A - Supported INFOTYPE 01h to 08h		
BIT	**Function**	**name**
0	INFOTYPE 08h : In-Use Performance Tracking	NC_STATE_SRV_9_CAN_VIT[0]
1	INFOTYPE 07h : Not supported	
2	INFOTYPE 06h : Calibration Verification Numbers	
3	INFOTYPE 05h : Not supported	
4	INFOTYPE 04h : Calibration IDs	
5	INFOTYPE 03h : Not supported	
6	INFOTYPE 02h : Vehicule Identification Number	
7	INFOTYPE 01h : Not supported	

DATA B - Supported INFOTYPEs 09h to 10h

INFOTYPE 00h - DATA B - Supported INFOTYPE 09h to 10h		
BIT	**Function**	**name**
0	INFOTYPE 10h : reserved	NC_STATE_SRV_9_CAN_VIT[1]
1	INFOTYPE 0Fh : reserved	
2	INFOTYPE 0Eh : reserved	
3	INFOTYPE 0Dh : reserved	
4	INFOTYPE 0Ch : reserved	
5	INFOTYPE 0Bh : reserved	
6	INFOTYPE 0Ah : ECUNAME	
7	INFOTYPE 09h : Not supported	

DATA C - Supported INFOTYPEs 11h to 18h

INFOTYPE 00h - DATA C - Supported INFOTYPE 11h to 18h		
BIT	**Function**	**name**
0	INFOTYPE 18h : reserved	NC_STATE_SRV_9_CAN_VIT[2]
1	INFOTYPE 17h : reserved	
2	INFOTYPE 16h : reserved	
3	INFOTYPE 15h : reserved	
4	INFOTYPE 14h : reserved	
5	INFOTYPE 13h : reserved	
6	INFOTYPE 12h : reserved	
7	INFOTYPE 11h : reserved	

DATA D - Supported INFOTYPEs 19h to 20h

INFOTYPE 00h - DATA D - Supported INFOTYPE 19h to 20h		
BIT	**Function**	**name**

0	INFOTYPE 20h : INFOTYPEs supported ($21 - $40)	NC_STATE_SRV_9_CAN_VIT[3]
1	INFOTYPE 1Fh : reserved	
2	INFOTYPE 1Eh : reserved	
3	INFOTYPE 1Dh : reserved	
4	INFOTYPE 1Ch : reserved	
5	INFOTYPE 1Bh : reserved	
6	INFOTYPE 1Ah : reserved	
7	INFOTYPE 19h : reserved	

54.26.3 Description of Service 09h, INFOTYPE 02h - VIN

VIN = 17 characters and should be reported as ASCII values

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: 02 - VIN (Vehicle Identification Number)	02	INFTYP

Formula section:

CALL ACTION_OBDC_*GetVIN* ()
CONSTRUCT Positive response

54.26.3.1 Response positive report Vehicle Identification Number (VIN)

Request vehicle information response message

Message direction:	ECU #1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: 02 - VIN (Vehicle Information Number)	02	INFTYP
#3	Number of data items: 01	01	NODI
#4	Data_A: VIN Character 1	XX	VIN
#5	Data_B: VIN Character 2	XX	VIN
#6	Data_C: VIN Character 3	XX	VIN
#7	Data_D: VIN Character 4	XX	VIN
#8	Data_E: VIN Character 5	XX	VIN
#9	Data_F: VIN Character 6	XX	VIN
#10	Data_G: VIN Character 7	XX	VIN
#11	Data_H: VIN Character 8	XX	VIN
#12	Data_I: VIN Character 9	XX	VIN
#13	Data_J: VIN Character 10	XX	VIN
#14	Data_K: VIN Character 11	XX	VIN
#15	Data_L: VIN Character 12	XX	VIN
#16	Data_M: VIN Character 13	XX	VIN
#17	Data_N: VIN Character 14	XX	VIN
#18	Data_O: VIN Character 15	XX	VIN
#19	Data_P: VIN Character 16	XX	VIN
#20	Data_Q: VIN Character 17	XX	VIN

54.26.4 Description of Service 09h, INFOTYPE 04h - Calibration IDs

A Calibration ID will be reported for the ECM, depending on the software architecture. Calibration ID can include a maximum of 16 ASCII characters. Calibration ID can contain only printable ASCII characters, and will be reported as ASCII values.
Calibration ID should uniquely identify the software installed in the ECM.
The amount of Calibration Id is included in the "Number of data items" parameter.
About Calibration coherence identifier definition sees the file: ECU reprogramming, coherence system (Chapter 0)

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: Calibration Id	04	INFTYP

Formula section:

```
IF (NC_STATE_COMP_TYP_CAN AND 0x01) = 0x01
THEN CALL ACTION_OBDC_GetCALIDCBoot()
ENDIF

IF (NC_STATE_COMP_TYP_CAN AND 0x02) = 0x02
THEN CALL ACTION_OBDC_GetCALIDSBoot()
ENDIF

IF (NC_STATE_COMP_TYP_CAN AND 0x04) = 0x04
THEN CALL ACTION_OBDC_GetCALIDCalib()
ENDIF

IF (NC_STATE_COMP_TYP_CAN AND 0x08) = 0x08
THEN CALL ACTION_OBDC_GetCALIDECU()
ENDIF

IF (NC_STATE_COMP_TYP_CAN AND 0x10) = 0x10
THEN CALL ACTION_OBDC_GetCALIDSelSeg()
ENDIF

CONSTRUCT Positive response
```

54.26.4.1 Response positive report Calibration ID

Request vehicle information response message

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Id	04	INFTYP
#3	Number of data items:	XX	NODI
#4	Data_A: Calibration ID for XXX: char 1	XXX	CALID
#5	Data_B: Calibration ID for XXX: char 2	XXX	CALID
#6	Data_C: Calibration ID for XXX: char 3	XXX	CALID
#7	Data_D: Calibration ID for XXX: char 4	XXX	CALID
#8	Data_E: Calibration ID for XXX: char 5	XXX	CALID
#9	Data_F: Calibration ID for XXX: char 6	XXX	CALID
#10	Data_G: Calibration ID for XXX: char 7	XXX	CALID
#11	Data_H: Calibration ID for XXX: char 8	XXX	CALID
#12	Data_I: Calibration ID for XXX: char 9	XXX	CALID
#13	Data_J: Calibration ID for XXX: char 10	XXX	CALID
#14	Data_K: Calibration ID for XXX: char 11	XXX	CALID
#15	Data_L : Calibration ID for XXX: char 12	XXX	CALID
#16	Data_M : Calibration ID for XXX: char 13	XXX	CALID
#17	Data_N : Calibration ID for XXX: char 14	XXX	CALID
#18	Data_O : Calibration ID for XXX: char 15	XXX	CALID
#19	Data_P : Calibration ID for XXX: char 16	XXX	CALID
----	------	-----	-----
#n	Data_A: Calibration ID for YYY: char 1	YYY	CALID
#n+1	Data_B: Calibration ID for YYY: char 2	YYY	CALID
# n+2	Data_C: Calibration ID for YYY: char 3	YYY	CALID
# n+3	Data_D: Calibration ID for YYY: char 4	YYY	CALID
# n+4	Data_E: Calibration ID for YYY: char 5	YYY	CALID
# n+5	Data_F: Calibration ID for YYY: char 6	YYY	CALID
# n+6	Data_G: Calibration ID for YYY: char 7	YYY	CALID
# n+7	Data_H: Calibration ID for YYY: char 8	YYY	CALID
# n+8	Data_I: Calibration ID for YYY: char 9	YYY	CALID
# n+9	Data_J: Calibration ID for YYY: char 10	YYY	CALID
# n+10	Data_K: Calibration ID for YYY: char 11	YYY	CALID
# n+11	Data_L : Calibration ID for YYY: char 12	YYY	CALID
# n+12	Data_M : Calibration ID for YYY: char 13	YYY	CALID
# n+13	Data_N : Calibration ID for YYY: char 14	YYY	CALID
# n+14	Data_O : Calibration ID for YYY: char 15	YYY	CALID
# n+15	Data_P : Calibration ID for YYY: char 16	YYY	CALID

Note:

- The sequence of the 16 bytes (Data_A – Data_P) from #n to #n+15 is depending of the components number. This is depending of how many components are configured in the configuration byte NC_STATE_COMP_TYP_CAN.
    - o e.g. (if it is configured to 1 component we have only one sequence of 16 bytes, if it is configured to 2 components we have two sequences of 16 bytes, and so on).
    - o e.g. if it is configured to 1 component than the positive response will be constructed using only one data set for CALID and the length of response is 19 bytes.

  - e.g. if it is configured to 2 components than the positive response will be constructed using two data sets for CALID XXX, and YYY and the length of response is 35 bytes.
- the order of the CALID in the response is the same like in the NC_STATE_COMP_TYP_CAN:

Positive response

SID	InfoType	1st group of 16 bytes	2nd group of 16 bytes	3rd group of 16 bytes	4th group of 16 bytes	5th group of 16 bytes
49	04	CALID for one area or a combination of several or all section areas of the ECU	ECU CALID	CALIB CALID	SBOOT CALID	CBOOT CALID

54.26.5 Description of Service 09h, INFOTYPE 06h - Calibration verification numbers

The on-board software will calculate the CVN (CRC) based on memory content
If the calculation does not use all four bytes, the CVN shall be right justified and filled with 00h.
The amount of CVNs is included in the "Number of data items" parameter.

Request vehicle information request message (read InfoType values)

Data Byte	Parameter Name	Hex Value	Mnemonic
#1	**Request vehicle information request SID**	**09**	**SIDRQ**
#2	InfoType (read InfoType values)	06	INFTYP

54.26.5.1 Response positive report calibration verification number

Request vehicle information response message

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: Calibration Verification Number	06	INFTYP
#3	Number of data items:	XX	NODI
#4	Data_A : 1st byte of CRC for XXX calculation value	XXX	DATA_A
#5	Data_B : 2nd byte of CRC for XXX calculation value	XXX	DATA_B
#6	Data_C : 3rd byte of CRC for XXX calculation value	XXX	DATA_C
#7	Data_D : 4th byte of CRC for XXX calculation value	XXX	DATA_D
----	-----	-----	-----
#n	Data_A : 1st byte of CRC for YYY calculation value	YYY	DATA_A
#n+1	Data_B : 2nd byte of CRC for YYY calculation value	YYY	DATA_B
#n+2	Data_C : 3rd byte of CRC for YYY calculation value	YYY	DATA_C
#n+3	Data_D : 4th byte of CRC for YYY calculation value	YYY	DATA_D

Note:

- The sequence of the 4 bytes (Data_A – Data_D) from #n to #n+3 is depending of the components number. This is depending of how many components are configured in the configuration byte NC_STATE_COMP_TYP_CAN.
  - e.g. (if it is configured to 1 component we have only one sequence of 4 bytes, if it is configured to 2 components we have two sequences of 4 bytes, and so on).
- the order of the CVNs in the response is the same like in the NC_STATE_COMP_TYP_CAN:

Positive response

SID	**InfoType**	**1st group of 4 bytes**	**2nd group of 4 bytes**	**3rd group of 4 bytes**	**4th group of 4 bytes**	**5th group of 4 bytes**
49	06	CRC for one area or a combination of several or all section areas of the ECU	ECU CRC	CALIB CRC	SBOOT CRC	CBOOT CRC

Application conditions:

Initialisation: At NVMINI system event:

STATE_CKS_BOOT_1_SAE = 0
STATE_CKS_BOOT_2_SAE = 0
STATE_CKS_ECU_SAE = 0
STATE_CKS_CAL_SAE = 0
STATE_CKS_SEL_SEG_SAE = 0

Recurrence: -

Activation: -

Formula section:

```
IF NC_STATE_CKS_CONF_CAN is 1
THEN
AT RESET:
          Restore values for STATE_CKS_SEL_SEG_SAE, STATE_CKS_BOOT_1_SAE,
          STATE_CKS_BOOT_2_SAE, STATE_CKS_ECU_SAE, STATE_CKS_CAL_SAE from
          NVMY
ENDIF
```

At reception of the request the correct flow is:

```
IF NC_STATE_CKS_CONF_CAN is 1
THEN (1)
      IF for the first time after a reprogramming event of the ECU(s), the
checksum computation is not ready
      THEN (2)
            A negative response message with response code 78h –
      RequestCorrectlyReceived-ResponsePending must be sent by the ECU(s) until
      the checksum computation is done and positive response message is
      available
      ELSE (2)
            Check if checksum is done
            IF checksum for all COMPONENTs, depending of the configuration of
      the project is done
            if (cks_service(component, NC_CMD_CRC_SRV_CHK_READY) == component)
            THEN (3)
                Intitialize STATE_CKS_SEL_SEG_SAE, STATE_CKS_BOOT_1_SAE,
                STATE_CKS_BOOT_2_SAE, STATE_CKS_ECU_SAE,
                STATE_CKS_CAL_SAE, with CKS values:
                IF (NC_STATE_COMP_TYP_CAN & 10h)   //Several sections
                  THEN STATE_CKS_SEL_SEG_SAE = cks_service_cift(component for
                  several sections, NC_CMD_CRC_SRV_GET_CKS)
                ENDIF
                IF (NC_STATE_COMP_TYP_CAN & 08h)   //ECU area
                  THEN STATE_CKS_ECU_SAE = cks_service_cift(component for ECU area,
                  NC_CMD_CRC_SRV_GET_CKS)
                ENDIF
                IF (NC_STATE_COMP_TYP_CAN & 04h)   //CALIB area
                  THEN STATE_CKS_CAL_SAE = cks_service_cift(component for CALIB area,
                  NC_CMD_CRC_SRV_GET_CKS)
                ENDIF
                IF (NC_STATE_COMP_TYP_CAN & 02h)   //SBOOT area
                  THEN STATE_CKS_BOOT_2_SAE = cks_service_cift(component for SBOOT
                  area, NC_CMD_CRC_SRV_GET_CKS)
                ENDIF
                IF (NC_STATE_COMP_TYP_CAN & 01h)   //CBOOT area
                  THEN STATE_CKS_BOOT_1_SAE = cks_service_cift(component for CBOOT
                  area, NC_CMD_CRC_SRV_GET_CKS)
                ENDIF
                  Send positive(s) response(s)
            ELSE (3)
                  Send positive(s) response(s) with the values for CKS restored
            at RESET
            ENDIF (3)
      ENDIF (2)
ELSE (1)
IF NC_STATE_CKS_CONF_CAN is 2
      IF crc calculation not started (the negative response « 78 pending
response » was not sent at all)
```

```
      THEN (4)
            Start crc calculation
            CALL cks_service(component, NC_CMD_CRC_SRV_START)
            Send negative response "78h pending response"
      ELSE (4)
            AT each recall before next response check if checksum is done
            IF checksum for all COMPONENTs, depending of the configuration of
the project is done
            if (cks_service(component, NC_CMD_CRC_SRV_CHK_READY) == component)
            THEN (5)
                  CALL cks_service(component, NC_CMD_CRC_SRV_GET_CKS)
                  Send positive(s) response(s)
            ELSE (5)
                  Send negative response "78h pending response"
            ENDIF (5)
      ENDIF (4)
ENDIF (1)
```

54.26.6 Description of Service 09h, InfoType 08h

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: In-use Performance Tracking	08	INFTYP

Formula section:

CALL ACTION_ERRM_SelectRbmData()
CONSTRUCT positive response

54.26.6.1 Response positive report In-Use Performance Tracking

Request vehicle information response message

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: In-use Performance Tracking	08	INFTYP
#3	Number of data items: 16	10	NODI
#4	Data A1: MSB (OBD Monitoring Conditions Encountered Counts – general denominator)	XX	OBDCOND_A
#5	Data B1: LSB (OBD Monitoring Conditions Encountered Counts – general denominator)	XX	OBDCOND_B
#6	Data C1: MSB (Ignition Counter)	XX	IGNCNTR_A
#7	Data D1: LSB (Ignition Counter)	XX	IGNCNTR_B
#8	Data A2: MSB (Catalyst Monitor Completion Counts Bank 1 – numerator)	XX	CATCOMP1_A
#9	Data B2: LSB (Catalyst Monitor Completion Counts Bank 1 – numerator)	XX	CATCOMP1_B
#10	Data C2: MSB (Catalyst Monitor Conditions Encountered Counts Bank 1 – denominator)	XX	CATCOND1_A
#11	Data D2: LSB (Catalyst Monitor Conditions Encountered Counts Bank 1 – denominator)	XX	CATCOND1_B
#12	Data A3: MSB (Catalyst Monitor Completion Counts Bank 2 – numerator)	XX	CATCOMP2_A
#13	Data B3: LSB (Catalyst Monitor Completion Counts Bank 2 – numerator)	XX	CATCOMP2_B
#14	Data C3: MSB (Catalyst Monitor Conditions Encountered Counts Bank 2 – denominator)	XX	CATCOND2_A
#15	Data D3: LSB (Catalyst Monitor Conditions Encountered Counts Bank 2 – denominator)	XX	CATCOND2_B
#16	Data A4: MSB (O2 Sensor Monitor Completion Counts Bank 1 – numerator)	XX	O2SCOMP1_A
#17	Data B4: LSB (O2 Sensor Monitor Completion Counts Bank 1 – numerator)	XX	O2SCOMP1_B
#18	Data C4: MSB (O2 Sensor Monitor Conditions Encountered Counts Bank 1 – denominator)	XX	O2SCOND1_A
#19	Data D4: LSB (O2 Sensor Monitor Conditions Encountered Counts Bank 1 – denominator)	XX	O2SCOND1_B
#20	Data A5: MSB (O2 Sensor Monitor Completion Counts Bank 2 – numerator)	XX	O2SCOMP2_A
#21	Data B5: LSB (O2 Sensor Monitor Completion Counts Bank 2 – numerator)	XX	O2SCOMP2_B
#22	Data C5: MSB (O2 Sensor Monitor Conditions Encountered Counts Bank 2 – denominator)	XX	O2SCOND2_A
#23	Data D5: LSB (O2 Sensor Monitor Conditions Encountered Counts Bank 2 – denominator)	XX	O2SCOND2_B
#24	Data A6 MSB (EGR Monitor Completion Condition Counts – numerator)	XX	EGRCOMP_A
#25	Data B6: LSB (EGR Monitor Completion Condition Counts – numerator)	XX	EGRCOMP_B
#26	Data C6: MSB (EGR Monitor Conditions Encountered Counts – denominator)	XX	EGRCOND_A
#27	Data D6: LSB (EGR Monitor Conditions Encountered Counts – denominator)	XX	EGRCOND_B
#28	Data A7: MSB (AIR Monitor Completion Condition Counts (Secondary Air) – numerator)	XX	AIRCOMP_A

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#29	Data B7: LSB (AIR Monitor Completion Condition Counts (Secondary Air) – numerator)	XX	AIRCOMP_B
#30	Data C7: MSB (AIR Monitor Conditions Encountered Counts (Secondary Air) – denominator)	XX	AIRCOND_A
#31	Data D7: LSB (AIR Monitor Conditions Encountered Counts (Secondary Air) – denominator)	XX	AIRCOND_B
#32	Data A8: MSB (EVAP Monitor Completion Condition Counts – numerator)	XX	EVAPCOMP_A
#33	Data B8: LSB (EVAP Monitor Completion Condition Counts – numerator)	XX	EVAPCOMP_B
#34	Data C8: MSB (EVAP Monitor Conditions Encountered Counts – denominator)	XX	EVAPCOND_A
#35	Data D8: LSB (EVAP Monitor Conditions Encountered Counts – denominator)	XX	EVAPCOND_B

54.26.7 Description of Service 09h, INFOTYPE 0Ah - ECUNAME

The number of the data bytes reported to the scan tool for InfoType 0Ah must be 20 ASCII characters.

Request vehicle information request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: 0A – ECUNAME	0A	INFTYP

Formula section:

CALL ACTION_OBDC_GetECUNAME()
CONSTRUCT Positive response

54.26.7.1 Response positive report ECUNAME

Request vehicle information response message

Message direction:	ECU #1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value (Hex)**	**Mnemonic**
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: 0A - ECUNAME	0A	INFTYP
#3	Number of data items: 01	01	NODI
#4	Data_A: ECU Acronym Character 1	XX	ECU
#5	Data_B: ECU Acronym Character 2	XX	ECU
#6	Data_C: ECU Acronym Character 3	XX	ECU
#7	Data_D: ECU Acronym Character 4	XX	ECU
#8	Data_E: Delimiter	"-"	-
#9	Data_F: Reported text name Character 1	YY	ECUNAME
#10	Data_G: Reported text name Character 2	YY	ECUNAME
#11	Data_H: Reported text name Character 3	YY	ECUNAME
#12	Data_I: Reported text name Character 4	YY	ECUNAME
#13	Data_J: Reported text name Character 5	YY	ECUNAME
#14	Data_K: Reported text name Character 6	YY	ECUNAME
#15	Data_L: Reported text name Character 7	YY	ECUNAME
#16	Data_M: Reported text name Character 8	YY	ECUNAME
#17	Data_N: Reported text name Character 9	YY	ECUNAME
#18	Data_O: Reported text name Character 10	YY	ECUNAME
#19	Data_P: Reported text name Character 11	YY	ECUNAME
#20	Data_Q: Reported text name Character 12	YY	ECUNAME
#21	Data_R: Reported text name Character 13	YY	ECUNAME
#22	Data_S: Reported text name Character 14	YY	ECUNAME
#23	Data_T: Reported text name Character 15	YY	ECUNAME

54.27 Service 09h - Request vehicle information – OBD on CAN (Application Incident)

Export actions:

ACTION_OBDC_*GetVIN* (OUT < PRM_STATE_VIN >)
This action fill the VIN with the correct values from the project
ACTION_OBDC_*GetCALIDCBoot* (OUT < PRM_STATE_BOOT_1 >)
This action fill the CBOOT CALID with the correct values from the project
ACTION_OBDC_*GetCALIDSBoot* (OUT < PRM_STATE_BOOT_2>)
This action fill the SBOOT CALID with the correct values from the project
ACTION_OBDC_*GetCALIDECU* (OUT < PRM_STATE_ECU>)
This action fill the ECU CALID with the correct values from the project
ACTION_OBDC_*GetCALIDCalib* (OUT < PRM_STATE_CAL>)
This action fill the CALIB CALID with the correct values from the project
ACTION_OBDC_*GetCALIDSelSeg* (OUT < PRM_STATE_SEL_SEG>)
This action fill the several or all segments of memory CALID with the correct values from the project
ACTION_OBDC_GetECUNAME(OUT <PRM_STATE_ECU_NAME>) This action fills the emission related ECU Name

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_STATE_CKS_CONF_CAN	-	0...FFH	0...255	1	[-]
byte which defines the response method for service $09 InfoType $06					
NC_STATE_COMP_TYP_CAN	-	0...FFH	0...255	1	[-]
byte which defines the components used in a ECU					
NC_STATE_SEL_SEG_CAN	-	0...FFH	0...255	1	[-]
Configuration byte which defines area of an ECU					

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_STATE_SRV_9_CAN_VIT	4	0...FFH	0...255	1	[-]
LDP_STATE_VIT_ID_STATE_SRV_9	4	0...03H	0...3	1	[-]
Table of bytes which defines the INFOTYPEs supported by the service 09h (01h – 20h) – ISO 15765-4					

54.27.1 NC_STATE_CKS_CONF_CAN

NC_STATE_CKS_CONF_CAN	
Value	**Description**
1	Method #1: The CVN(s) must not be computed on demand, but instead shall be computed at least once per trip. A trip shall be of reasonable length (e.g. 5 - 10 minutes). The computed CVN(s) shall be stored in NVM (Non Volatile Memory) for immediate access by the external test equipment. Once the computation is completed for the very first time after a reprogramming event of the ECU(s) or a battery disconnect, the results shall be made available to the external test equipment even if the engine is running. If the CVN(s) are requested before they have been computed a negative response message with response code $78 - RequestCorrectlyReceived-ResponsePending shall be sent by the ECU(s) until the positive response message is available for the ISO 14230-4 and ISO 15765-4 protocols.
2	Method #2: If method #1 does not apply the ECU(s)' on-board software shall compute the CVN(s) on an external test equipment request message. If the ECU(s) are not able to send an immediate positive response message a negative response message with response code $78 - RequestCorrectlyReceived-ResponsePending shall be sent by the ECU(s) until the positive response message is available for the ISO 14230-4 and ISO 15765-4 protocols.

Formula section:

NC_STATE_CKS_CONF_CAN = 1

54.27.2 NC_STATE_COMP_TYP_CAN

NC_STATE_COMP_TYP_CAN		
BIT	**Function**	**Supported**
0	Use CBOOT area	0
1	Use SBOOT area	0
2	Use CALIB area	**1**
3	Use ECU area	**1**
4	One area or a combination of several or all section areas of the ECU	0
5	Dummy value, always 0	0
6	Dummy value, always 0	0
7	Dummy value, always 0	0

Formula section:

see Table

54.27.3 NC_STATE_SEL_SEG_CAN

NC_STATE_SEL_SEG_CAN	
Value	**Description**
XX	The value of the define represent the area of one component or a a combination of several or all section areas of the ECU

Formula section:

NC_STATE_SEL_SEG_CAN = 0 (not used)

54.27.4 Supported INFOTYPEs for Service 09h – ISO 15765-4

Description:

In the table **ID_STATE_SRV_9_CAN_VIT** each byte represents an 8 INFOTYPEs configuration

ID_STATE_SRV_9_CAN_VIT[0]	**Supported INFOTYPEs from 01h – 08h**	
BIT	**Function**	**Supported**
0	INFOTYPE 08h : In-Use Performance Tracking	**1**
1	INFOTYPE 07h : Not used	0
2	INFOTYPE 06h : Calibration Verification Numbers	**1**
3	INFOTYPE 05h : Not used	0
4	INFOTYPE 04h : Calibration IDs	**1**
5	INFOTYPE 03h : Not used	0
6	INFOTYPE 02h : Vehicule Identification Number	**1**
7	INFOTYPE 01h : Not used	0

ID_STATE_SRV_9_CAN_VIT[1]	**Supported INFOTYPEs from 09h – 10h**	
BIT	**Function**	**Supported**
0	INFOTYPE 10h : Reserved	0
1	INFOTYPE 0Fh : Reserved	0
2	INFOTYPE 0Eh : Reserved	0
3	INFOTYPE 0Dh : Reserved	0
4	INFOTYPE 0Ch : Reserved	0
5	INFOTYPE 0Bh : Reserved	0
6	INFOTYPE 0Ah : ECUNAME	**1**
7	INFOTYPE 09h : Reserved	0

ID_STATE_SRV_9_CAN_VIT[2]	**Supported INFOTYPEs from 11h – 18h**	
BIT	**Function**	**Supported**
0	INFOTYPE 18h : Reserved	0
1	INFOTYPE 17h : Reserved	0
2	INFOTYPE 16h : Reserved	0
3	INFOTYPE 15h : Reserved	0
4	INFOTYPE 14h : Reserved	0
5	INFOTYPE 13h : Reserved	0
6	INFOTYPE 12h : Reserved	0
7	INFOTYPE 11h : Reserved	0

ID_STATE_SRV_9_CAN_VIT[3]	**Supported INFOTYPEs from 19h – 20h**	
BIT	**Function**	**Supported**
0	INFOTYPE 20h : INFOTYPEs supported ($21 - $40)	0
1	INFOTYPE 1Fh : Reserved	0
2	INFOTYPE 1Eh : Reserved	0
3	INFOTYPE 1Dh : Reserved	0
4	INFOTYPE 1Ch : Reserved	0
5	INFOTYPE 1Bh : Reserved	0
6	INFOTYPE 1Ah : Reserved	0
7	INFOTYPE 19h : Reserved	0

54.27.5 Description of Service 09h, INFOTYPE 02h – VIN

Description:

Vehicle Identification Number (VIN) = 17 characters and should be reported as ASCII values

Description for actions:

ACTION_OBDC_*GetVIN* (OUT < PRM_STATE_VIN >)					
This action fill the VIN with the correct values from the project					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_VIN[0..16]	IN	0...FFH	0...255	1	[-]
See description bellow					

FUNCTION DESCRIPTION:

Description of parameter OUT used by action **ACTION_OBDC_*GetVIN*** ():

PRM_STATE_VIN:

Software structure of 17 ASCI elements filled up with VIN. This structure has to be filled byte by byte.

PRM_STATE_VIN[0] = kwp_io_vin[0] *// VIN Character 0*
PRM_STATE_VIN[1] = kwp_io_vin[1] *// VIN Character 1*
PRM_STATE_VIN[2] = kwp_io_vin[2] *// VIN Character 2*
PRM_STATE_VIN[3] = kwp_io_vin[3] *// VIN Character 3*
PRM_STATE_VIN[4] = kwp_io_vin[4] *// VIN Character 4*
PRM_STATE_VIN[5] = kwp_io_vin[5] *// VIN Character 5*
PRM_STATE_VIN[6] = kwp_io_vin[6] *// VIN Character 6*
PRM_STATE_VIN[7] = kwp_io_vin[7] *// VIN Character 7*
PRM_STATE_VIN[8] = kwp_io_vin[8] *// VIN Character 8*
PRM_STATE_VIN[9] = kwp_io_vin[9] *// VIN Character 9*
PRM_STATE_VIN[10] = kwp_io_vin[10] *// VIN Character 10*
PRM_STATE_VIN[11] = kwp_io_vin[11] *// VIN Character 11*
PRM_STATE_VIN[12] = kwp_io_vin[12] *// VIN Character 12*
PRM_STATE_VIN[13] = kwp_io_vin[13] *// VIN Character 13*
PRM_STATE_VIN[14] = kwp_io_vin[14] *// VIN Character 14*
PRM_STATE_VIN[15] = kwp_io_vin[15] *// VIN Character 15*
PRM_STATE_VIN[16] = kwp_io_vin[16] *// VIN Character 16*

54.27.6 Description of Service 09h, INFOTYPE 04h - Calibration IDs

Description:

A Calibration ID will be reported for the ECM, depending on the software architecture. Calibration ID can include a maximum of 16 ASCII characters. Calibration ID can contain only printable ASCII characters, and will be reported as ASCII values.
Calibration ID should uniquely identify the software installed in the ECM.
The amount of Calibration Id is included in the "Number of data items" parameter.
About Calibration coherence identifier definition see the file : ECU reprogramming, coherence system (Chapter 0)

54.27.6.1 ACTION_OBDC_*GetCALIDCBoot* (OUT < PRM_STATE_BOOT_1 >)

Description for actions:

ACTION_OBDC_*GetCALIDCBoot* (OUT < PRM_STATE_BOOT_1 >)					
This action fill the CBOOT CALID with the correct values from the project					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_BOOT_1[0...15]	IN	0...FFH	0...255	1	[-]
Name of SV CBOOT-ID					

FUNCTION DESCRIPTION:

Description of parameter OUT used by action **ACTION_OBDC_*GetCALIDCBoot*** ():

PRM_STATE_BOOT_1:

Software structure of 16 ASCI elements filled up with CBOOT CALID(Calibration Identifications only for CBOOT area). This structure has to be filled byte by byte.

PRM_STATE_BOOT_1[0] = not used *//CBOOT CALID Character 1*
PRM_STATE_BOOT_1[1] = not used *//CBOOT CALID Character 2*
PRM_STATE_BOOT_1[2] = not used *//CBOOT CALID Character 3*
PRM_STATE_BOOT_1[3] = not used *//CBOOT CALID Character 4*
PRM_STATE_BOOT_1[4] = not used *//CBOOT CALID Character 5*
PRM_STATE_BOOT_1[5] = not used *//CBOOT CALID Character 6*
PRM_STATE_BOOT_1[6] = not used *//CBOOT CALID Character 7*
PRM_STATE_BOOT_1[7] = not used *//CBOOT CALID Character 8*
PRM_STATE_BOOT_1[8] = not used *//CBOOT CALID Character 9*
PRM_STATE_BOOT_1[9] = not used *//CBOOT CALID Character 10*
PRM_STATE_BOOT_1[10] = not used *//CBOOT CALID Character 11*
PRM_STATE_BOOT_1[11] = not used *//CBOOT CALID Character 12*
PRM_STATE_BOOT_1[12] = not used *//CBOOT CALID Character 13*
PRM_STATE_BOOT_1[13] = not used *//CBOOT CALID Character 14*
PRM_STATE_BOOT_1[14] = not used *//CBOOT CALID Character 15*
PRM_STATE_BOOT_1[15] = not used *//CBOOT CALID Character 16*

54.27.6.2 ACTION_OBDC_*GetCALIDSBoot* (OUT < PRM_STATE_BOOT_2 >)

Description for actions:

ACTION_OBDC_*GetCALIDSBoot* (OUT < PRM_STATE_BOOT_2 >)					
This action fill the SBOOT CALID with the correct values from the project					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_BOOT_2[0...15]	IN	0...FFH	0...255	1	[-]
Name of SV SBOOT-ID					

FUNCTION DESCRIPTION:

Description of parameter OUT used by action **ACTION_OBDC_*GetCALIDSBoot*** ():

PRM_STATE_BOOT_2:

Software structure of 16 ASCI elements filled up with SBOOT CALID(Calibration Identifications only for SBOOT area). This structure has to be filled byte by byte.

PRM_STATE_BOOT_2[0] = not used *//SBOOT CALID Character 1*

PRM_STATE_BOOT_2[1] = not used *//SBOOT CALID Character 2*

PRM_STATE_BOOT_2[2] = not used *//SBOOT CALID Character 3*

PRM_STATE_BOOT_2[3] = not used *//SBOOT CALID Character 4*

PRM_STATE_BOOT_2[4] = not used *//SBOOT CALID Character 5*

PRM_STATE_BOOT_2[5] = not used *//SBOOT CALID Character 6*

PRM_STATE_BOOT_2[6] = not used *//SBOOT CALID Character 7*

PRM_STATE_BOOT_2[7] = not used *//SBOOT CALID Character 8*

PRM_STATE_BOOT_2[8] = not used *//SBOOT CALID Character 9*

PRM_STATE_BOOT_2[9] = not used *//SBOOT CALID Character 10*

PRM_STATE_BOOT_2[10] = not used *//SBOOT CALID Character 11*

PRM_STATE_BOOT_2[11] = not used *//SBOOT CALID Character 12*

PRM_STATE_BOOT_2[12] = not used *//SBOOT CALID Character 13*

PRM_STATE_BOOT_2[13] = not used *//SBOOT CALID Character 14*

PRM_STATE_BOOT_2[14] = not used *//SBOOT CALID Character 15*

PRM_STATE_BOOT_2[15] = not used *//SBOOT CALID Character 16*

54.27.6.3 ACTION_OBDC_*GetCALIDECU* (OUT < PRM_STATE_ECU >)

Description for actions:

ACTION_OBDC_*GetCALIDECU* (OUT < PRM_STATE_ECU >)					
This action fill the ECU CALID with the correct values from the project					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_ECU[0...15]	IN	0...FFH	0...255	1	[-]
See description bellow					

FUNCTION DESCRIPTION:

Description of parameter OUT used by action **ACTION_OBDC_*GetCALIDECU*** ():

PRM_STATE_ECU:

Software structure of 16 ASCI elements filled up with ECU CALID(Calibration Identifications only for ECU area). This structure has to be filled byte by byte.

PRM_STATE_ECU[0] = low nibble from bmw_hw_nr[2] *//ECU CALID Character 1*
PRM_STATE_ECU[1] = high nibble from bmw_hw_nr[3] *//ECU CALID Character 2*
PRM_STATE_ECU[2] = low nibble from bmw_hw_nr[3] *//ECU CALID Character 3*
PRM_STATE_ECU[3] = high nibble from bmw_hw_nr[4] *//ECU CALID Character 4*
PRM_STATE_ECU[4] = low nibble from bmw_hw_nr[4] *//ECU CALID Character 5*
PRM_STATE_ECU[5] = high nibble from bmw_hw_nr[5] *//ECU CALID Character 6*
PRM_STATE_ECU[6] = low nibble from bmw_hw_nr[5] *//ECU CALID Character 7*
PRM_STATE_ECU[7] = 00h *//ECU CALID Character 8*
PRM_STATE_ECU[8] = 00h *//ECU CALID Character 9*
PRM_STATE_ECU[9] = 00h *//ECU CALID Character 10*
PRM_STATE_ECU[10] = 00h *//ECU CALID Character 11*
PRM_STATE_ECU[11] = 00h *//ECU CALID Character 12*
PRM_STATE_ECU[12] = 00h *//ECU CALID Character 13*
PRM_STATE_ECU[13] = 00h *//ECU CALID Character 14*
PRM_STATE_ECU[14] = 00h *//ECU CALID Character 15*
PRM_STATE_ECU[15] = 00h *//ECU CALID Character 16*

54.27.6.4 ACTION_OBDC_*GetCALIDCalib* (OUT < PRM_STATE_CAL >)

Description for actions:

ACTION_OBDC_*GetCALIDCalib* (OUT < PRM_STATE_CAL >)					
This action fill the CALIB CALID with the correct values from the project					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_CAL[0...15]	IN	0...FFH	0...255	1	[-]
See description bellow					

FUNCTION DESCRIPTION:

Description of parameter OUT used by action **ACTION_OBDC_*GetCALIDCalib*** ():

PRM_STATE_CAL:

Software structure of 16 ASCI elements filled up with CALIB CALID(Calibration Identifications only for Calibration area). This structure has to be filled byte by byte.

Remark:
bmw_cal_nr[0..6] has to be defined inside the project as interface, because the actual CALIB CALID must be evaluated from the BMW_UIF (reprog-history) table first.

PRM_STATE_CAL[0] = bmw_cal_nr[0] *//CALIB CALID Character 1*
PRM_STATE_CAL[1] = bmw_cal_nr[1] *//CALIB CALID Character 2*
PRM_STATE_CAL[2] = bmw_cal_nr[2] *//CALIB CALID Character 3*
PRM_STATE_CAL[3] = bmw_cal_nr[3] *//CALIB CALID Character 4*
PRM_STATE_CAL[4] = bmw_cal_nr[4] *//CALIB CALID Character 5*
PRM_STATE_CAL[5] = bmw_cal_nr[5] *//CALIB CALID Character 6*
PRM_STATE_CAL[6] = bmw_cal_nr[6] *//CALIB CALID Character 7*
PRM_STATE_CAL[7] = not used *//CALIB CALID Character 8*
PRM_STATE_CAL[8] = not used *//CALIB CALID Character 9*
PRM_STATE_CAL[9] = not used *//CALIB CALID Character 10*
PRM_STATE_CAL[10] = not used *//CALIB CALID Character 11*
PRM_STATE_CAL[11] = not used *//CALIB CALID Character 12*
PRM_STATE_CAL[12] = not used *//CALIB CALID Character 13*
PRM_STATE_CAL[13] = not used *//CALIB CALID Character 14*
PRM_STATE_CAL[14] = not used *//CALIB CALID Character 15*
PRM_STATE_CAL[15] = not used *//CALIB CALID Character 16*

54.27.6.5 ACTION_OBDC_*GetCALIDSelSeg* (OUT < PRM_STATE_SEL_SEG >)

Description for actions:

ACTION_OBDC_*GetCALIDSelSeg* (OUT < PRM_STATE_SEL_SEG >)					
This action fill the several or all segments of memory CALID with the correct values from the project					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_SEL_SEG[0...15]	IN	0...FFH	0...255	1	[-]
See description bellow					

FUNCTION DESCRIPTION:

Description of parameter OUT used by action **ACTION_OBDC_*GetCALIDSelSeg*** ():

PRM_STATE_SEL_SEG:

Software structure of 16 ASCI elements filled up with SEGMENT CALID(Calibration Identifications only for several or all segments of memory). This structure has to be filled byte by byte.

PRM_STATE_SEL_SEG[0] = not used *//SEGMENT CALID Character 1*
PRM_STATE_SEL_SEG[1] = not used *//SEGMENT CALID Character 2*
PRM_STATE_SEL_SEG[2] = not used *//SEGMENT CALID Character 3*
PRM_STATE_SEL_SEG[3] = not used *//SEGMENT CALID Character 4*
PRM_STATE_SEL_SEG[4] = not used *//SEGMENT CALID Character 5*
PRM_STATE_SEL_SEG[5] = not used *//SEGMENT CALID Character 6*
PRM_STATE_SEL_SEG[6] = not used *//SEGMENT CALID Character 7*
PRM_STATE_SEL_SEG[7] = not used *//SEGMENT CALID Character 8*
PRM_STATE_SEL_SEG[8] = not used *//SEGMENT CALID Character 9*
PRM_STATE_SEL_SEG[9] = not used *//SEGMENT CALID Character 10*
PRM_STATE_SEL_SEG[10] = not used *//SEGMENT CALID Character 11*
PRM_STATE_SEL_SEG[11] = not used *//SEGMENT CALID Character 12*
PRM_STATE_SEL_SEG[12] = not used *//SEGMENT CALID Character 13*
PRM_STATE_SEL_SEG[13] = not used *//SEGMENT CALID Character 14*
PRM_STATE_SEL_SEG[14] = not used *//SEGMENT CALID Character 15*
PRM_STATE_SEL_SEG[15] = not used *//SEGMENT CALID Character 16*

54.27.6.6 Description of Service 09h, INFOTYPE 0Ah - ECUNAME

Description for actions:

ACTION_OBDC_*GetECUNAME*(OUT <PRM_STATE_ECU_NAME>)					
This action fills the emission related ECU Name					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_STATE_ECU_NAME[0...19]	IN	0...FFH	0...255	1	[-]
See description bellow					

FUNCTION DESCRIPTION:

Description of parameter OUT used by action ACTION_OBDC_GetECUNAME():

PRM_STATE_ECU_NAME:

Software structure of 20 ASCII characters filled up with ECU Name of an emission related ECU. This structure has to be field byte by byte.

Defined field assignment:

- Data bytes 1-4, “XXXX”, contains ECU acronym, and
- Data byte 5, “-“, contains delimiter, and
- Data bytes 6-20, “YYYYYYYYYYYYYYY”, contains text name

ECU acronyms:

The first 4 bytes from the response of InfoType 0Ah have to be one of the acronyms described in the table bellow.

The emissions-related ECUs (control modules) have to report the external test equipment acronym and name as listed below. This table is not complete and emissions-related ECUs not listed in the table must be reported to ISO/SAE for definition.		
External test equipment reported acronym (max 1 – 4 chars	**Full name of Control Module/emissions-related ECU**	**External test equipment reported name (max 15 chars.)**
ABS	Anti-Lock Brake System (ABS) Control Module	**AntiLock Brake**
AFCM	Alternative Fuel Control Module	**Alt. Fuel Crtl**
AHCM	Auxiliary Heater Control Module	**Aux. Heat Crtl**
BECM	Battery Energy Control Module	**B+ Energy Crtl**
BSCM	Brake System Control Module	**Brake System**
CCM	Cruise Control Module	**Cruise Control**
CTCM	Coolant Temperature Control Module	**Cool Temp Crtl**
DMCM	Drive Motor Control Module	**Drive Mot.Crtl**
ECCI	Emissions Critical Control Information	**Emis Crit Info**
ECM	Engine Control Module	**Engine Control**
FACM	Fuel Additive Control Module	**Fuel Add. Crtl**

FICM	Fuel Injector Control Module	**Fuel Inj. Crtl**
FPCM	Fuel Pump Control Module	**Fuel Pump Crtl**
FWDC	Four-Wheel Drive Clutch Control Module	**4 Whl Dr.Cl.Crtl**
GPCM	Glow Plug Control Module	**Glow Plug Crtl**
GSM	Gear Shift Control Module	**Gear Shift Crtl**
HPCM	Hybrid Powertrain Control Module	**Hybrid Ptr Crtl**
IPC	Instrument Panel Cluster (IPC) Control Module	**Inst. Panel Cl.**
PCM	Powertrain Control Module	**Powertrain Crtl**
SGCM	Starter / Generator Control Module	**Start/Gen. Crtl**
TACM	Throttle Actuator Control Module	**Thr.Act. Crtl**
TCCM	Transfer Case Control Module	**Transf Case Crtl**
TCM	Transmission Control Module	**Transm. Crtl**

External test equipment reported name: **ECM - Engine Control**

PRM_STATE_ECU_NAME[0] = E

PRM_STATE_ECU_NAME[1] = C

PRM_STATE_ECU_NAME[2] = M

PRM_STATE_ECU_NAME[3] = $00

PRM_STATE_ECU_NAME[4] = "-" *//delimiter*

PRM_STATE_ECU_NAME[5] = E

PRM_STATE_ECU_NAME[6] = n

PRM_STATE_ECU_NAME[7] = g

PRM_STATE_ECU_NAME[8] = i

PRM_STATE_ECU_NAME[9] = n

PRM_STATE_ECU_NAME[10] = e

PRM_STATE_ECU_NAME[11] = " " *//space*

PRM_STATE_ECU_NAME[12] = C

PRM_STATE_ECU_NAME[13] = o

PRM_STATE_ECU_NAME[14] = n

PRM_STATE_ECU_NAME[15] = t

PRM_STATE_ECU_NAME[16] = r

PRM_STATE_ECU_NAME[17] = o

PRM_STATE_ECU_NAME[18] = l

PRM_STATE_ECU_NAME[19] = $00

55 Operating System Monitoring

Table of Contents

Index

55.1 CPU Load Measurement

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CPU_LOAD	O/V	0...400H	0..100	0.0977	%
Type: U16. CPU load above measurement level.					
CPU_LOAD_MAX	O/V/S	0...400H	0..100	0.0977	%
Type: U16. Maximum detected CPU load above measurement level.					
CPU_LOAD_FIL	O/V	0...400H	0..100	0.0977	%
Type: U16. Filtered CPU load above measurement level.					
CPU_LOAD_FIL_MAX	O/V/S	0...400H	0..100	0.0977	%
Type: U16. Maximum detected filtered CPU load above measurement level.					
CPU_LOAD_TSK_PER_MIN	O/V	0...FFFFFFFFH	0..4294967295	1	Ticks
Type: U32. Minimum time between 2 Instances of TASK_BG_MES.					
CPU_LOAD_TSK_PER_CTR	O/V	0...FFFFFFFFH	0..4294967295	1	-
Type: U32. Numbers of calls of TASK_BG_MES.					
CPU_LOAD_SES_CTR	O/V	8000...7FFFH	-32768..32767	1	-
Type: S16. Session counter of CPU load measurement.					

FUNCTION DESCRIPTION:

General information:

In a priority-based system, working with a background task (task configured at the lowest priority level and systematically running when no higher priority treatments are triggered), the real processor load is always 100%. Nevertheless, the CPU time spent for the background task can be considered as potential free CPU time (if a higher priority treatment is triggered, the background task will be interrupted).

As a consequence, the CPU load (CPU_LOAD) is then defined as the processor activity above the background level (priority level 0).

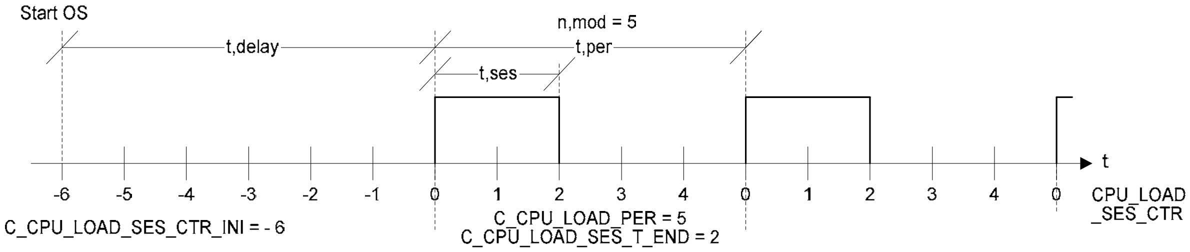
The computation of the CPU load consumed above the background level (priority 0) is based on the usage of a measurement background task (TASK_BG_MES) of fixed and known net duration $t_{\mathrm{TASK\_BG\_MES,net}}$ (automatically characterized by the minimum time between two tasks occurrences: CPU_LOAD_TSK_PER_MIN) at the priority level 1 and a time function in a task (priority > 1) with fixed recurrence (e.g. 100ms). The duration of TASK_BG_MES can be tuned with the temporization value C_CPU_LOAD_TSK_MES_DLY. The temporization is necessary to increase the measurement background task time having enough measurement resolution.

The detected minimum duration time (depending on the chosen temporization) is shown in value CPU_LOAD_TSK_PER_MIN and shall be put to C_CPU_LOAD_TSK_PER_MIN.

After having fixed C_CPU_LOAD_TSK_MES_DLY and C_CPU_LOAD_TSK_PER_MIN, these two calibrations, plus C_CPU_LOAD_PER and C_CPU_LOAD_SES_T_END (see description below) shall be set none tunable in DDS in order to make sure that no miscalibration is possible.

During a measurement session t_{ses} of period t_{per} occurring every fixed modulo counter n_{mod} (C_CPU_LOAD_PER), TASK_BG_MES is triggered once from the application and then continuously re-activates itself.

This measurement session t_{ses} starts at the beginning of the period t_{per} and can be tuned up to 100% of the period t_{per} (boundary case, where C_CPU_LOAD_SES_T_END is set equal C_CPU_LOAD_PER). The measurement session t_{ses} shall be less than or equal to 1 second. A start delay of measurement can be defined by setting the session counter CPU_LOAD_SES_CTR to a negative value at initialization (calibration C_CPU_LOAD_SES_CTR_INI).



The potential free CPU factor $\mathrm{CPU}_{\mathrm{free}}$ is obtained by multiplying the counted occurrences n (CPU_LOAD_TSK_PER_CTR) of measurement background task (TASK_BG_MES) over the measurement session with the duration of the measurement background task.

$$\mathrm{CPU}_{\mathrm{free}} = \frac{n \cdot t_{\mathrm{TASK\_BG\_MES,net}}}{t_{ses}}$$

The CPU load value (CPU_LOAD) is then deduced from the potential $\mathrm{CPU}_{\mathrm{free}}$ value.

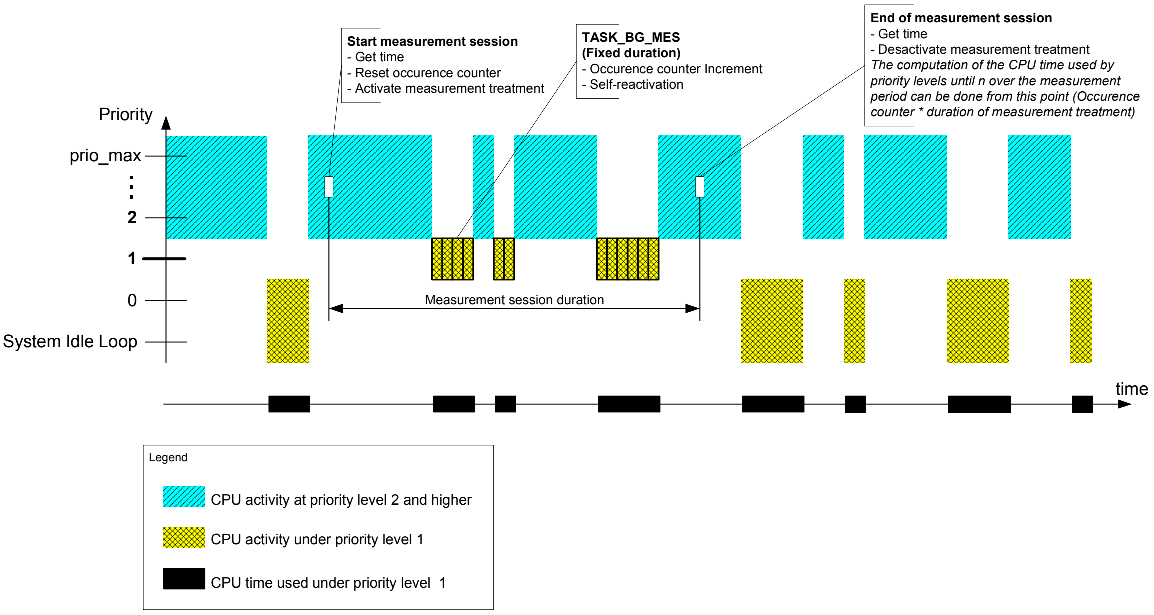
$$\mathrm{CPU}_{\mathrm{load}} = 1 - \mathrm{CPU}_{\mathrm{free}} = 1 - \frac{n \cdot t_{\mathrm{TASK\_BG\_MES,net}}}{t_{ses}}$$

Due to a continuous activity at priority level 1 during the measurement period, it must be noticed that the background treatments running at priority 0 will not occur during the measurement session.

CPU_LOAD_FIL is the PT_1 filtered value of CPU_LOAD with P=1 and T_1 set by C_CPU_LOAD_FTC (see description of calibration).

The variable CPU_LOAD_MAX represents the maximum detected CPU load value. The variable CPU_LOAD_FIL_MAX represents the maximum of the filtered CPU load value.

The following diagram summarizes the principle of the measurement method:



Start measurement session
- Get time
- Reset occurence counter
- Activate measurement treatment

TASK_BG_MES
(Fixed duration)
- Occurence counter Increment
- Self-reactivation

End of measurement session
- Get time
- Desactivate measurement treatment

*The computation of the CPU time used by priority levels until n over the measurement period can be done from this point (Occurence counter * duration of measurement treatment)*

Figure 1: The principle of the measurement method

Three APIs are available in order to have a flexible measurement method:

Measurement enable API:

This API shall be called at least at initialization if the measurement has to be done. After measurement disable API was called measurement enable API shall be used to enable measurement again.

Measurement disable API:

Shall be used to disable measurement method.

Clear Maximum API:

Shall be used to clear CPU_LOAD_MAX and CPU_LOAD_FIL_MAX.

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_CPU_LOAD_TSK_PER_MIN	1	0...FFFFH	0..65535	1	Ticks
Type: U16. Learned minimum time between 2 Instances of TASK_BG_MES.					
C_CPU_LOAD_TSK_MES_DLY	1	0...FFFFH	0..65535	1	-
Type: U16. Temporization for background task duration.					
C_CPU_LOAD_PER	1	1..7FFFH	1..32767	1	-
Type: S16. Time factor between 2 measurement sessions depending on the task recurrence.					
C_CPU_LOAD_SES_T_END	1	1..1F4H	1..500	1	-
Type: S16. End of meas. session. In a 2ms task the max. allowed session time of 1s is set by using factor 500.					
C_CPU_LOAD_SES_CTR_INI	1	8000..0H	-32768..0	1	-
Type: S16. Start delay of measurement session.					
C_CPU_LOAD_FTC	1	0..FFFFH	0..0.9999	1/65536	-
Type: U16. Filter time constant for CPU_LOAD_FIL.					

C_CPU_LOAD_TSK_PER_MIN and C_CPU_LOAD_TSK_MES_DLY:

The calibration C_CPU_LOAD_TSK_PER_MIN must be set equal to the value CPU_LOAD_TSK_PER_MIN after it is computed depending on the temporization value C_CPU_LOAD_TSK_MES_DLY.

C_CPU_LOAD_PER and C_CPU_LOAD_SES_T_END:

The measurement window is to be set with C_CPU_LOAD_PER (modulo counter of measurement session, measurement window always starts if counter is zero and if start is allowed) and C_CPU_LOAD_SES_T_END (measurement window stops at this counter value). If C_CPU_LOAD_PER is equal to C_CPU_LOAD_SES_T_END the duty cycle of the measurement is 100%.

C_CPU_LOAD_SES_CTR_INI:

With C_CPU_LOAD_SES_CTR_INI an easy start delay of measurement session can be set. The content of the value is calculated with

$$\mathrm{C\_CPU\_LOAD\_SES\_CTR\_INI} = -\frac{\text{requested start delay}}{\text{time triggered function recurrence}}$$

Example: The requested start delay after OS initialization is 500ms. The time triggered function is called every 100ms.

$$\mathrm{C\_CPU\_LOAD\_SES\_CTR\_INI} = -\frac{500\mathrm{ms}}{100\mathrm{ms}} = -5$$

C_CPU_LOAD_FTC:

Filter time constant for filtered CPU load (CPU_LOAD_FIL).

$\mathrm{C\_CPU\_LOAD\_FTC} = \frac{T \cdot n_{mod}}{T \cdot n_{mod} + T_1}$, with T = recurrence of time triggered function, n_{mod} is the modulo counter C_CPU_LOAD_PER and T1 is the needed filter time.

Example: The time triggered function is called every 100ms. n_{mod} is set to 6. The filter time shall be 2s.

$$\mathrm{C\_CPU\_LOAD\_FTC} = \frac{100\mathrm{ms} \cdot 6}{100\mathrm{ms} \cdot 6 + 2\mathrm{s}} = \frac{600\mathrm{ms}}{600\mathrm{ms} + 2000\mathrm{ms}} = 0.2308 \text{ phys.} \rightarrow 15124\text{d (3B14H)}$$

56 Power Supply

Table of Contents

Index

56.1 Acquisition of after relay battery voltage

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VB_MES	O	0...3FFH	0...28.7055	2.81E-02	[V]
Computed battery voltage					

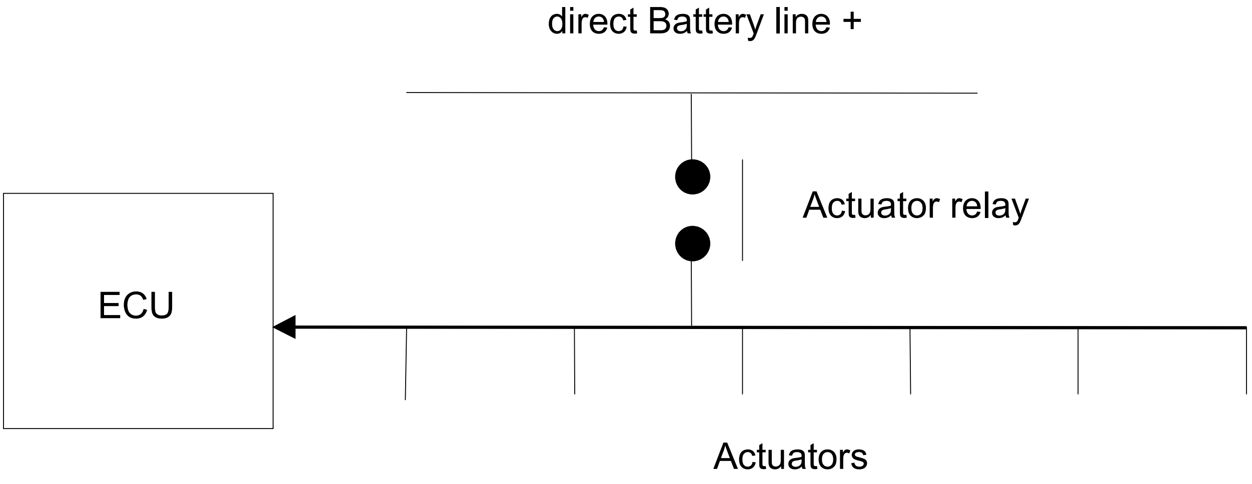
Input data:

VB_BAS			

FUNCTION DESCRIPTION:

General information:

The acquisition of the raw values is performed every 1 ms with the resolution of 10 bits.



Application conditions:

Activation: at reset

Initialization: VB_MES = 0.

Recurrence : NC_VB_SAMPLE_NR ms

System Description:

The range of VB_BAS and VB_MES depends to the voltage divider (means 0 - 5V at ADC) at the input of the µC. The input is the ADC-value of the VB-Pin.

Formula section:

$$VB\_MES = (1/NC\_VB\_SAMPLE\_NR) * \sum_{n=1}^{NC\_VB\_SAMPLE\_NR} VB\_BAS_n$$

NC_VB_SAMPLE_NR = 2^n ; 1 =< n =< 6 (NC_VB_SAMPLE_NR is initialized with 4 or 8)

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_VB_SAMPLE_NR	1	0...FFH	0...255	1	[-]
Number of acquisition for filtering					
NC_VB_MAX	1	B4...140H	18...32	0.1	[V]
Maximum battery voltage linked to the voltage dividor					

56.2 Acquisition of raw after key battery voltage

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
V_IGK_MES	V/O	0...3FFH	0...28.7055	2.81E-02	[V]
Computed ignition key voltage					

Input data:

V_IGK_BAS			

FUNCTION DESCRIPTION:

General information:

The acquisition of the raw values is performed every 1 ms with the resolution of 10 bits. V_IGK_BAS is the same ADC-value which is used in Key on/Key off detection.

Application conditions:

Activation: at reset

Initialization: V_IGK_MES = 0

Recurrence : NC_V_IGK_SAMPLE_NR ms

Formula section:

$$V\_IGK\_MES = (1/NC\_V\_IGK\_SAMPLE\_NR) * \sum_{n=1}^{NC\_V\_IGK\_SAMPLE\_NR} V\_IGK\_BAS_n$$

$NC\_V\_IGK\_SAMPLE\_NR = 2^{n}$; 1 =< n =< 6 (NC_V_IGK_SAMPLE_NR is initialized with 4 or 8)

Configuration data:

Name	DIM	Hex. limits	Phys. limits	Resol.	Unit
NC_V_IGK_MAX	1	B4...140H	18...32	0.1	[V]
Maximum battery voltage linked to the voltage divisor of the key voltage					
NC_V_IGK_SAMPLE_NR	1	0...FFH	0...255	1	[-]
Number of acquisition for filtering					

56.3 Input system event : Key off recognition

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_KEY_OFF	V	0...1H	0...1	1	[-]
Key off flag					
LV_IGK	O/V	0...1H	0...1	1	[-]
Key on flag					
LV_WAKE_UP	O/V	0...1H	0...1	1	[-]
Wake Up flag					
STATE_PLAUS_IGK_BN	O/V	0...8H	0...8	1	[-]
state plausibility IGK - signal via CAN/via HW (BN2000 only)					
LV_IGK_PREL	O/V	0...1H	0...1	1	[-]
Preliminary ignition flag for 'ignition switch off'					
T_CAN_IGK_TOUT	O/V	0...FFH	0...2550	10	[ms]
Timeout ignition key via CAN bus					
T_WAKE_UP_ON	O/V	0...FFH	0...2550	10	[ms]
Timer ECU wake up					

Input data:

V_IGK_BAS	STATE_IGK_CAN	STATE_IGK_HW	LV_INH_PWL_TRAN_ES_EL
LV_VAR_BN	PWL_LOCK_CDN	STATE_CAN_DNM_D	B_sleepwait

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_IGK_OFF_PREL	1	0...FFH	0...2.55	0.01	[s]
Delay time ignition off after driver's command					
C_T_TEMP_IGK_OFF_PREL	1	0...FFFFH	0...655.35	0.01	[s]
Temp. time prel. ignition off after CAS-HW-error during dc					
C_T_IGK_LIH	1	0...FFH	0...2.55	0.01	[s]
Timeoutzeit for LV_IGK = 1, if no Can message "Klemmenstatus" available					

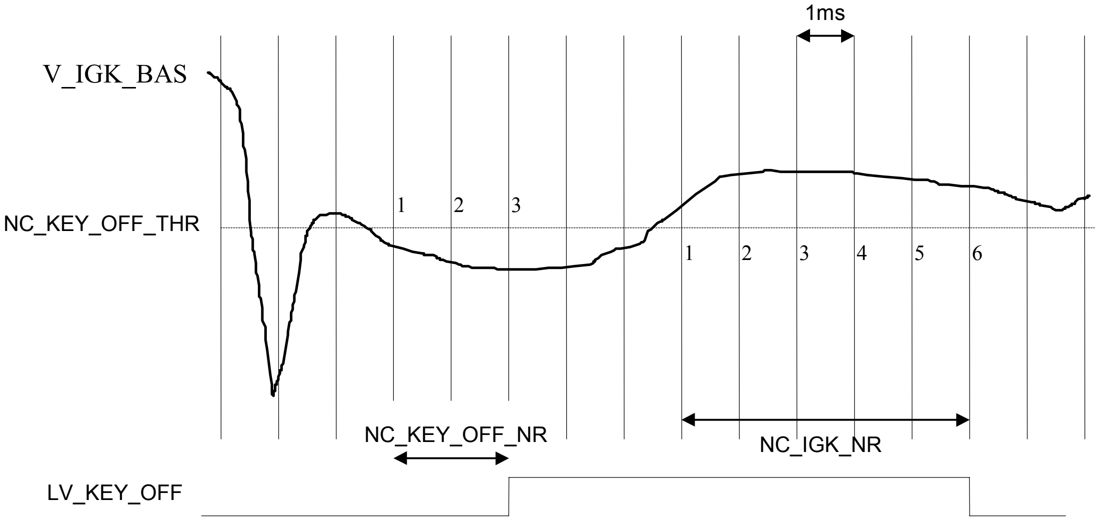
Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_KEY_OFF_THR	1	0...3FFH	0...28.7055	0.0280601	[V]
Threshold for key off detection (typical value : 142 = 4 Volts)					
NC_KEY_OFF_NR	1	0...FFH	0...255	1	[-]
Samples filter for key off detection (typical value : 3)					
NC_IGK_NR	1	0...FFH	0...255	1	[-]
Samples filter for key on detection (typical value : 6)					

FUNCTION DESCRIPTION:

General information:

According to the battery voltage raw value (V_IGK_BAS) after ignition key, the ignition key ON / OFF recognition is performed.



See on Analogic Acquisition Sampling chapter the recurrence of V_IGK_BAS acquisition.

Application conditions:

Activation: at reset

Initialization:

```
if      first value V_IGK_BAS > NC_KEY_OFF_THR
  then        LV_KEY_OFF  =  0
  else        LV_KEY_OFF  =  1
endif
```

Recurrency: 1ms

Formula section:

Ignition key ON recognised :

```
If      V_IGK_BAS > NC_KEY_OFF_THR
        for at least NC_IGK_NR number of successive samples ( * 1 msec. )
then    LV_KEY_OFF = 0 (ON)
```

Ignition key OFF recognised :

```
If      V_IGK_BAS ≤ NC_KEY_OFF_THR
        for at least NC_KEY_OFF_NR number of successive samples ( * 1 msec. )
then    LV_KEY_OFF = 1 (OFF)
```

56.4 Ignition : Key on recognition

FUNCTION DESCRIPTION:

Application conditions:

Activation: at reset

Initialization: LV_IGK = 0
LV_WAKE_UP = 0
T_IGK_LIH = C_T_IGK_LIH
LV_IGK_PREL = 1
T_DLY_IGK_OFF_PREL = C_T_DLY_IGK_OFF_PREL
T_TEMP_IGK_OFF_PREL = C_T_TEMP_IGK_OFF_PREL

Recurrency: 10ms

Formula section:

56.4.1 Vehicle without BN2000 (LV_VAR_BN = 0)

```
If    LV_KEY_OFF = 1   and   T_DLY_IGK_OFF_PREL > 0
  then      LV_IGK_PREL  =  0
            T_DLY_IGK_OFF_PREL --                    //decremented by 10ms
     if       T_DLY_IGK_OFF_PREL = 0
       then        LV_IGK  =  0
       else        LV_IGK  =  1
     endif
  else
     if       LV_INH_PWL_TRAN_ES_EL = 1   or   LV_KEY_OFF = 1
       then        LV_IGK  =  0 ,  LV_IGK_PREL  =  0
       else        LV_IGK  =  1 ,  LV_IGK_PREL  =  1 ,
                   T_DLY_IGK_OFF_PREL  =  C_T_DLY_IGK_OFF_PREL
     endif
endif
```

56.4.2 Vehicle with BN2000 (LV_VAR_BN = 1)

Determination of LV_WAKE_UP

```
If    LV_KEY_OFF = 0   and   LV_INH_PWL_TRAN_ES_EL = 0

then        LV_WAKE_UP  =  1
            T_WAKE_UP_ON++ (limit: 0xFF)

else        LV_WAKE_UP  =  0
            T_WAKE_UP_ON = 0
endif
```

NM mode depending on LV_WAKE_UP and other conditions:

```
If      LV_WAKE_UP = 1  or  B_sleepwait = 1
Then    NM mode awake
Else    NM mode bus sleep
Endif
```

Determination of STATE_PLAUS_IGK_BN and of LV_IGK=1

```
If(1)(CAN message “Klemmenstatus” available                          and
     no alive-counter error CAS   // alive counter counts regularely  and
     no check-sum error CAS       // sent checksum = calculated  checksum)
then(1)   T_IGK_LIH = C_T_IGK_LIH
          T_CAN_IGK_TOUT = 0

  if(2a)  STATE_IGK_CAN = 0h
  then(2a) LV_T_DLY_IGK_OFF_PREL = 1          // timer should be decremented

    if(3b)  LV_KEY_OFF = 0
    then(3b)       STATE_PLAUS_IGK_BN  =  1H    //SYM_0//SCB, Master CAN
    else(3b)       STATE_PLAUS_IGK_BN  =  0H    //NO_SYM
    endif(3b)

  else(2a)

    if(3c)  STATE_IGK_CAN = 1
    then(3c)       LV_IGK  =  1
                   T_DLY_IGK_OFF_PREL  =  C_T_DLY_IGK_OFF_PREL

                   LV_T_DLY_IGK_OFF_PREL = 0    //timer should be reset to max.-value

      if(4a)  LV_KEY_OFF = 0
      then(4a)     STATE_PLAUS_IGK_BN  =  0H    //NO_SYM
      else(4a)     STATE_PLAUS_IGK_BN  =  2H    //SYM_1//SCG, Master CAN
      endif(4a)

    else(3c)       STATE_PLAUS_IGK_BN  =  8H    //SYM_3//CAN-signal not plausible

      if(4b)       LV_WAKE_UP = 1
      then(4b)     LV_IGK  =  1
                   T_DLY_IGK_OFF_PREL  =  C_T_DLY_IGK_OFF_PREL
                   LV_T_DLY_IGK_OFF_PREL = 0    //timer should be reset to max.-value

      else(4b)     LV_IGK=0
      endif(4b)

    endif(3c)

  endif(2a)

else(1)

  if(2b)  STATE_CAN_DNM_D > 0 and PWL_LOCK_CDN[BN2000] = 1
  then(2b)         T_CAN_IGK_TOUT++ (limit: 0xFF)

    if(3d)  T_IGK_LIH = 0
    then(3d)       STATE_PLAUS_IGK_BN  =  4H    //SYM_2//CAS error, Master HW
```

```
            if(4c)        LV_WAKE_UP = 1
            then(4c)      LV_IGK  =  1
                          T_DLY_IGK_OFF_PREL  =  C_T_DLY_IGK_OFF_PREL
                          LV_T_DLY_IGK_OFF_PREL = 0     //timer should reset to max-value
            else(4c)      LV_IGK=0
            endif(4c)

          else(3d)        T_IGK_LIH  --
          endif(3d)

      else(2b)  T_IGK_LIH  =  C_T_IGK_LIH
                T_CAN_IGK_TOUT = 0xFF
      endif(2b)

    endif(1)
```

Calculation of Timer T_DLY_IGK_OFF_PREL

```
IF          LV_T_DLY_IGK_OFF_PREL = 1               and
            T_DLY_IGK_OFF_PREL > 0
THEN        T_DLY_IGK_OFF_PREL - -
            IF      STATE_IGK_HW = 0 or T_DLY_IGK_OFF_PREL = 0
            THEN    LV_IGK = 0
            ENDIF
ENDIF
```

Determination of LV_IGK_PREL

```
If          STATE_IGK_HW = 1
Then        if      T_TEMP_IGK_OFF_PREL  > 0
            then    T_TEMP_IGK_OFF_PREL - -
            endif
else        T_TEMP_IGK_OFF_PREL =  C_T_TEMP_IGK_OFF_PREL
Endif


If          STATE_IGK_HW = 2           or
            (T_TEMP_OFF_IGK_PREL  = 0)
then        LV_IGK_PREL  =  1
else        LV_IGK_PREL  =  0
endif
```

56.5 PWSL - Requirements to infrastructure interface

Export actions:

ACTION_INFR_GetElDiagRlyMain(OUT<cdn_diag_rly_main>, **OUT**<err_rly_main>**)**
Return of the electric diagnosis for the main relay
ACTION_INFR_GetVDcdcSopc(OUT<PRM_VDCDC>)
Return the output voltage of the DCDC converter
ACTION_INFR_SetIGK_OFF(IN <LV_IGK_OFF_ACK >)
This action sets the output pin for acknowledge IGK off / the starter relay for start stop automatic (STST)
ACTION_INFR_GetElDiagIGK_OFF(OUT<CDN_DIAG_ACK_IGK_OFF>, OUT <ERR_DIAG_ACK_IGK_OFF>)
This action returns diagnosis status of acknowledge IGK off line / start relay STST driver stage (conditions and symptoms).

Description for actions:

ACTION_INFR_GetElDiagRlyMain(OUT<cdn_diag_rly_main**>, OUT<**err_rly_main**>)**					
This action reads the value of cdn_diag_rly_main respectively err_rly_main. In case of bit 1, the diagnosis condition is fulfilled otherwise the diagnosis condition isn´t fulfilled if the bit is equal 0. In case of error err_rly_main is set to 1 or 0 if no error occurs					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
cdn_diag_rly_main	OUT	0...7H	0...7	1	-
Diagnosis condition for symptoms of lv_cdn_diag_RLY_MAIN: bit 0: Set, if diagnosis condition for symptom SCP (SYM_0) is fulfilled bit 1: Set, if diagnosis condition for symptom SCG (SYM_1) is fulfilled bit 2: Set, if diagnosis condition for symptom OC (SYM_2) is fulfilled					
err_rly_main	OUT	0...7H	0...7	1	-
Raw value of error symptom: bit 0: Set, if raw value of error symptom SCP (SYM_0) is set bit 1: Set, if raw value of error symptom SCG (SYM_1) is set bit 2: Set, if raw value of error symptom OC (SYM_2) is set					
Raw value of error symptom. The relevant bit is set, if the error has been detected.					
ACTION_INFR_SetIGK_OFF(IN < LV_IGK_OFF_ACK >)					
This action sets the output pin for acknowledge IGK off / the starter relay STST					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
LV_IGK_OFF_ACK	IN	0H 1H	output pin high output pin low	1	-
Acknowledge IGK off / Starter relay STST control bit					
ACTION_INFR_GetElDiagIGK_OFF(OUT <CDN_DIAG_ACK_IGK_OFF>, **OUT** <ERR_DIAG_ACK_IGK_OFF>**)**					
This action returns diagnosis status of acknowledge IGK off line / start relay STST driver stage (conditions and symptoms).					
Parameter	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
CDN_DIAG_ACK_IGK_OFF	OUT	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of electrical diagnosis_Bit0:diagnosis condition for symptom SYM_0 ; Bit1:diagnosis condition for symptom SYM_1; Bit2:diagnosis condition for symptom SYM_2					
ERR_DIAG_ACK_IGK_OFF	OUT	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom for acknowledge IGK off / starter relay STST control diagnosis					

FUNCTION DESCRIPTION:

General information:

Description for actions:

ACTION_INFR_GetVDcdcSopc(OUT<PRM_VDCDC>)					
Return the output voltage of the DCDC converter					
Parameter	**Type**	**Hex. Limits**	**Phys. Limits**	**Resol.**	**Unit**
PRM_VDCDC	OUT	0 ... 7FFF H	0 ... 639.98046875	640/32768	V
Returns the output voltage of the DC/DC converter					

Requirements for ACTION_INFR_GetElDiagRlyMain:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
cdn_diag_rly_main	-	-	<bit coded>	err_diag_rly_main	Diagnosis condition for each symptom **bit 0:** diagnosis condition for symptom SCP (SYM_0) **bit 1:** diagnosis condition for symptom SCG (SYM_1) **bit 2:** diagnosis condition for symptom OC (SYM_2)
err_rly_main	-	-	<bit coded>	cdn_diag_rly_main	Bitcoded result of each symptom **bit 0:** raw value of error symptom SCP (SYM_0) **bit 1:** raw value of error symptom SCG (SYM_1) . **bit 2:** raw value of error symptom OC (SYM_2)

Diagnosis: ACTION_INFR_GetElDiagRlyMain returns the electric diagnosis for the Main Relay.

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_GetVDcdcSopc:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
PRM_VDCDC	640/32768 V	-	640/32768 V		This OUT parameter returns the output voltage of the DC/DC converter

Requirements for ACTION_INFR_SetIGK_OFF

The acknowledge IGK off / starter relay STST is controlled by the ECU output pin 1-13 (N43: MSA, N53: RES_OUT_7 , N54: SAP).

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

lv_igk_off_ack	-	-	<bit coded>		0H = output pin shall be switched to high off, 1H = output pin shall be switched to low

Diagnosis: no electrical diagnosis done here

Coincidence requirements: -

Requirements for ACTION_INFR_GetElDiagIGK_OFF

This action returns the result of the electric diagnosis of acknowledge IGK off / starter relay power stage for start stop automatic (STST).

- The device readout is performed autonomous by the Infrastructure each 10 ms.
- The error informations are gathered in the Infrastructure (or-ed symptom) until the Application reads out the information by calling ACTION_INFR_GetElDiagIGK_OFF.
- After having read out the information by calling ACTION_INFR_GetElDiagIGK_OFF, the data inside the Infrastructure are reset. Resetting of cdn_diag... avoids unambigious results in case of too short calling reccurrence of ACTION_INFR_GetElDiagIGK_OFF: Reset cdn_diag... indicates, that the gathering of the information is not completely finished.

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
cdn_diag_ack_igk_off	-	-	1	err_diag_ack_igk_off	Diagnosis condition for each symptom - Bit 0 : Set, if diagnosis condition for symptom SCP (SYM_0) is fulfilled - Bit 1 : Set, if diagnosis condition for symptom SCG (SYM_1) is fulfilled - Bit 2 : Set, if diagnosis condition for symptom OC (SYM_2) is fulfilled
err_diag_ack_igk_off	-	-	1	cdn_diag_ack_igk_off	Bit encoded result of each symptom 0h : NO_SYM 1h : SYM_0 2h : SYM_1 4h : SYM_2

Diagnosis: ACTION_INFR_GetElDiagIGK_OFF returns electric diagnosis of acknowledge IGK off / start relay control STST signal

Coincidence requirements: No coincidence requirements with other events

56.6 AGGR PWSL adaptation

56.6.1 Outputs for SV aggregates, SV internally

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VB_CDN_OBD_1	V/O	0...1H	0...1	1	-
Boolean for battery voltage condition fulfilled for OBD-I diagnosis					
LV_VB_CDN_OBD_2	V/O	0...1H	0...1	1	-
Boolean for battery voltage condition fulfilled for OBD-II diagnosis					

Input data:

LV_CDN_VB_OBD1	LV_CDN_VB_OBD2	

FUNCTION DESCRIPTION:

Adaptation to aggregate environment.

Application conditions:

Initialisation: 0

Recurrence : same recurrence as corresponding input data

Activation: every engine operating state

Formula section:

LV_VB_CDN_OBD_1 = LV_CDN_VB_OBD1

LV_VB_CDN_OBD_2 = LV_CDN_VB_OBD2

56.7 Battery voltage VB

56.7.1 Computation of VB

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VB	V/O	0...FFH	0...25.89843	0.1015625	[V]
Relay - battery voltage					
V_IGK	V/O	0...FFH	0...25.89843	0.1015625	[V]
Key - battery voltage					

Input data:

VB_MES	V_IGK_MES		

FUNCTION DESCRIPTION:

General information:

The battery voltage (V_IGK_MES and VB_MES (10 bit)) is measured up to 30V, 28V or 26V (means 0-5V at ADC), depending on the project-specific solution, but the range of VB (8 bit) remains 0-26 V. For the computation of VB in hex the correction factor NC_FAC_VB_RATIO is necessary; its value depends on the project-specific battery voltage range. NC_FAC_VB_RATIO is definded in the “List of Configuration Datas”. The following table shows how NC_FAC_VB_RATIO is formed.

Measured battery voltage range (for example)	NC_FAC_VB_RATIO
0 ..... 26 V	26V / 26V
0 .. 28,8 V	28,8V / 26V
0 ..... 30 V	30V / 26V

Application conditions:

Activation: at reset

Initialisation: VB = 0

V_IGK = 0

Recurrence: 10 ms

Formula section:

HEX calculation:

VB = (VB_MES / 4) * NC_FAC_VB_RATIO

V_IGK = (V_IGK_MES / 4) * NC_FAC_VB_RATIO

VB relay:

```
If      VB_MES <= 26 Volt
Then    VB = (VB_MES / 4 ) * NC_FAC_VB_RATIO
Else    VB = 26 Volt        (FF hex)
Endif
```

VB key:

```
If      V_IGK_MES <= 26 Volt
Then    V_IGK = (V_IGK_MES / 4 ) * NC_FAC_VB_RATIO
Else    V_IGK = 26 Volt        (FF hex)
Endif
```

56.7.2 Secured battery voltage computation

Output data:

Name	V/S	Hex. limits	Phys. limits	Resol.	Unit
VB_SECU	V	0...FFH	0...25.89843	0.1015625	[V]
Battery Voltage secured for dwell computation					

Input data:

VB			

Description:

A secured battery voltage value is a mix of VB and V_IGK values which provides a safer battery voltage for functions especially in case of VB - relay wire to the ECU disconnection.

The securate battery voltage value can be used for exemple for dwell computation in order to protect ignition coil and ECU ignition output stage against too long dwell time.

VB_SECU = MAX(VB, V_IGK - NC_VB_SECU)

with VB_SECU - NC_VB_SECU satured to zero

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_VB_SECU	1	0...FFH	0...25.89843	0.1015625	[V]
Battery voltage offset (typical value = 1,02V)					
NC_FAC_VB_RATIO	1	0...FFH	0...1.99218	0.0078125	[-]
Factor to adapt the HEX-range to the range of 0 to 26 V					

56.7.3 Battery voltage conditions for diagnosis / jump start detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VB_JUMP	V/O	0...1H	0...1	1	[-]
Logical value for detected overvoltage (jump start)					
LV_CDN_VB_OBD1	V/O	0...1H	0...1	1	[-]
Boolean for battery voltage condition fulfilled for OBD diagnosis (=10V)					
LV_CDN_VB_OBD2	V/O	0...1H	0...1	1	[-]
Boolean for battery voltage condition fulfilled for OBD-II diagnosis (=11V)					
LV_CDN_VB_MIN_DIAG	V/O	0...1H	0...1	1	[-]
Boolean for battery voltage condition fulfilled for remaining diagnosis					
LV_CDN_VB_CAN_DIAG	V/O	0...1H	0...1	1	[-]
Boolean for battery voltage condition fulfilled for CAN diagnosis					
LV_CDN_VB_BN_DIAG	V/O	0...1H	0...1	1	[-]
Boolean for battery voltage condition fulfilled for BN2000 diagnosis					
LV_CDN_VB_CAN_TQ_DIAG	V/O	0...1H	0...1	1	[-]
Boolean for battery voltage condition fulfilled for TQ_CAN-interfaces					

Input data:

VB	LV_IGK		

FUNCTION DESCRIPTION:

General information:

Depending on VB there is a detection:

- Jump start
- Valid range for OBD1 diagnosis
- Valid range for OBD2 diagnosis
- Valid range for CAN diagnosis
- Valid range for BN2000 diagnosis
- Valid range for TQ-CAN interface diagnosis
- Valid range for remaining customer diagnosis

Application conditions:

Initialisation: all 0

Recurrence: 10ms

Activation: at reset

56.7.3.1 Detection of overvoltage (Jump start)

Description:

The bit LV_VB_JUMP is used by functions which must not run when overvoltage is detected for example secondary air injection to prevent damages to the components.

Formula section:

Jump start detection:

```
If      VB  > C_VB_MIN_JUMP
Then    LV_VB_JUMP = 1
Else    LV_VB_JUMP = 0
Endif
```

56.7.3.2 Battery voltage valid for OBD diagnosis

Description:

Depending of OBD1/OBD2 there is a different flag for the valid VB range

Formula section:

Valid range for OBD1 diagnosis:

```
If      VB > C_VB_MIN_OBD1 and VB < C_VB_MAX_OBD1
Then    LV_CDN_VB_OBD1 = 1
Else    LV_CDN_VB_OBD1 = 0
Endif
```

Valid range for OBD2 diagnosis:

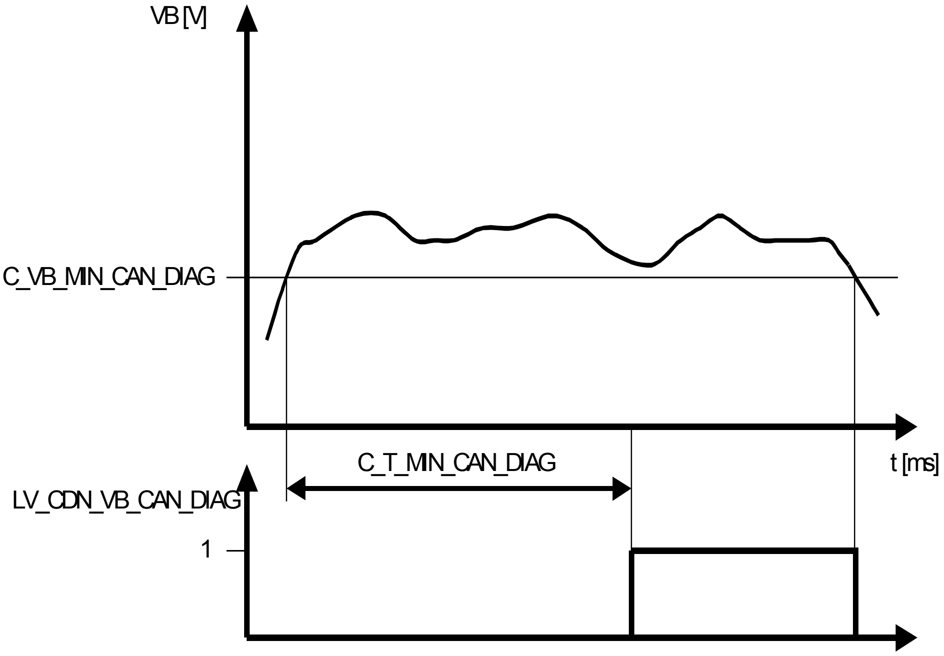
```
If      VB > C_VB_MIN_OBD2  and VB < C_VB_MAX_OBD2
Then    LV_CDN_VB_OBD2 = 1
Else    LV_CDN_VB_OBD2 = 0
Endif
```

56.7.3.3 Minimum battery voltage detection for CAN diagnosis

Description:

Due to low battery voltage CAN messages can be mutilated. So CAN timeout diagnosis is only done if battery voltage is valid for a minimum duration time.

Signal flow diagram:



Formula section:

Valid range for CAN diagnosis:

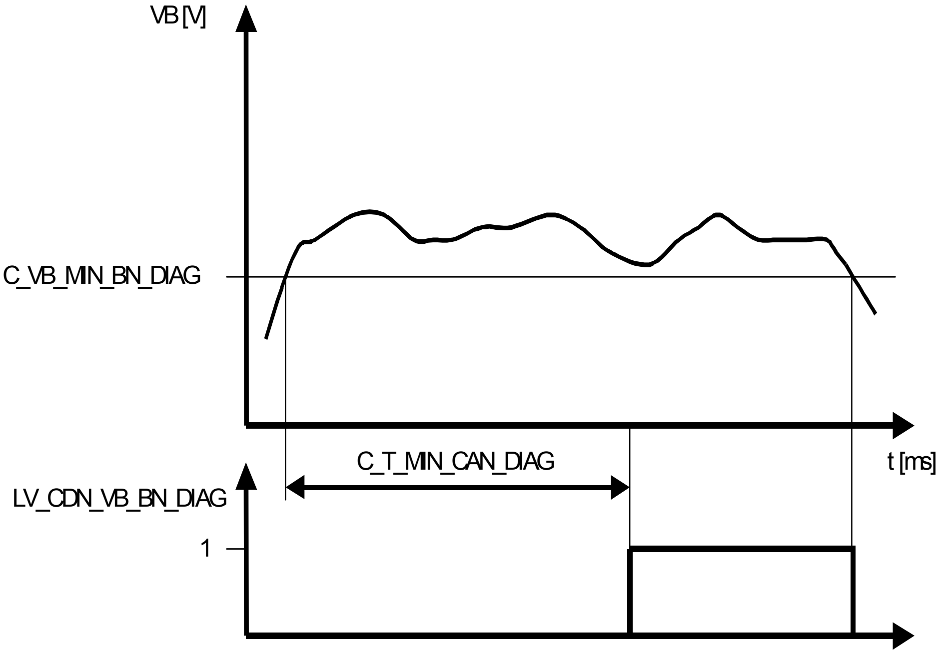
```
If      LV_IGK = 1                      and
        VB  >  C_VB_MIN_CAN_DIAG        for
        time > C_T_MIN_CAN_DIAG
         (timer always started at every transition VB  >  C_VB_CAN_DIAG
        and at transition LV_IGK 0 -> 1)
Then    LV_CDN_VB_CAN_DIAG = 1
Else    LV_CDN_VB_CAN_DIAG = 0
Endif
```

56.7.3.4 Minimum battery voltage detection for BN2000 diagnosis

Description:

Due to low battery voltage BN2000 messages can be mutilated. So BN2000 timeout diagnosis is only done if battery voltage is valid for a minimum duration time.

Signal flow diagram:



Formula section:

Valid range for BN2000 diagnosis:

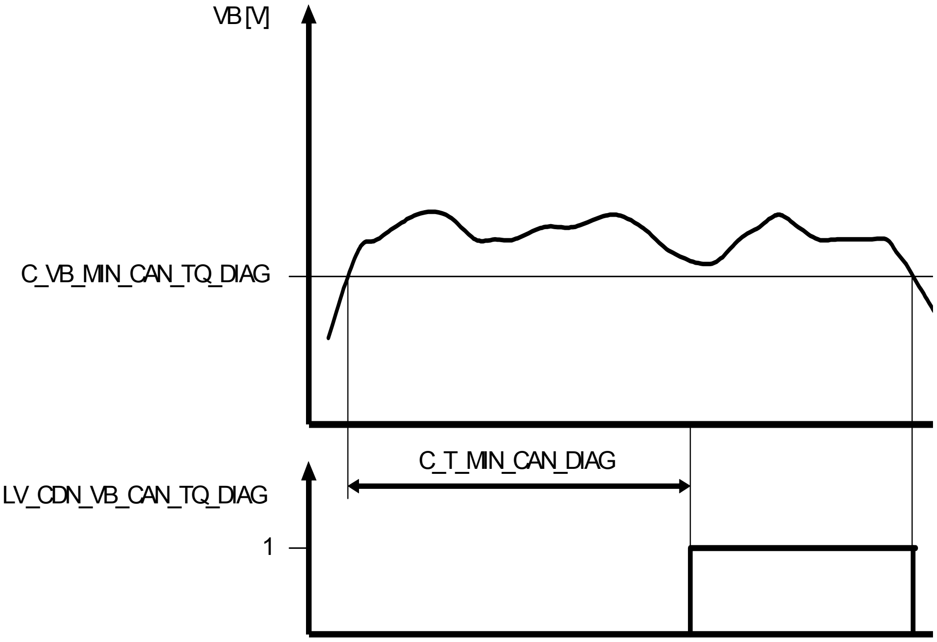
```
If      LV_IGK = 1                     and
        VB  >  C_VB_MIN_BN_DIAG        for
        time > C_T_MIN_CAN_DIAG
         (timer always started at every transition VB  >  C_VB_BN_DIAG
        and at transition LV_IGK 0 -> 1)
Then    LV_CDN_VB_BN_DIAG = 1
Else    LV_CDN_VB_BN_DIAG = 0
Endif
```

56.7.3.5 Minimum battery voltage detection for TQ_CAN-interfaces

Description:

Due to low battery voltage CAN messages can be mutilated. So the TQ_CAN-interfaces should only be enabled if battery voltage is valid for a minimum duration time.

Signal flow diagram:



Formula section:

Valid range for BN2000 diagnosis:

```
If      LV_IGK = 1                      and
        VB  >  C_VB_MIN_CAN_TQ_DIAG for
        time > C_T_MIN_CAN_DIAG
        (timer always started at every transition VB > C_VB_MIN_CAN_TQ_DIAG
        and at transition LV_IGK 0 -> 1)
Then    LV_CDN_VB_CAN_TQ_DIAG = 1
Else    LV_CDN_VB_CAN_TQ_DIAG = 0
Endif
```

56.7.3.6 Minimum Battery Voltage Detection for non OBD Diagnosis

Description:

The remaining diagnosis functions which are not covered by the above mentioned conditions are disabled below a separate threshold, which should be lower as OBD1/2 and CAN.

Formula section:

Valid range for non OBD diagnosis:

```
If      VB < C_VB_MIN_DIAG   or
        LV_IGK = 0
Then    LV_CDN_VB_MIN_DIAG = 0
Else    LV_CDN_VB_MIN_DIAG = 1
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VB_MIN_JUMP	1	0...FFH	0...25.89843	0.1015625	[V]
VB -threshold for maximum permissible battery voltage					
C_VB_MIN_OBD1	1	0...FFH	0...25.89843	0.1015625	[V]
VB –threshold for minimum permissible battery voltage for OBD diagnosis					
C_VB_MAX_OBD1	1	0...FFH	0...25.89843	0.1015625	[V]
VB -threshold for maximum permissible battery voltage for OBD diagnosis					
C_VB_MIN_OBD2	1	0...FFH	0...25.89843	0.1015625	[V]
VB -threshold for minimum permissible battery voltage for OBD2 diagnosis					
C_VB_MAX_OBD2	1	0...FFH	0...25.89843	0.1015625	[V]
VB -threshold for maximum permissible battery voltage for OBD2 diagnosis					
C_VB_MIN_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
Battery voltage threshold for low voltage detection for remaining diagnosis					
C_VB_MIN_CAN_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
Battery voltage threshold for CAN diagnosis					
C_T_MIN_CAN_DIAG	1	0...FFH	0...2550	10	[ms]
VB-Time-condition for CAN diagnosis					
C_VB_MIN_BN_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
Battery voltage threshold for BN2000 diagnosis					
C_VB_MIN_CAN_TQ_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
Battery voltage threshold for CAN-TQ interface diagnosis					

56.7.4 Moving average value for battery voltage

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VB_MMV	V/O	0...FFH	0...25.89843	0.1015625	[V]
Moving mean value of battery voltage VB					

Input data:

VB			

A moving average value VB_MMV is used in the chapter “Engine speed setpoint calculation” to define a minimum N_SP_IS in dependance of the battery voltage:

Application conditions:

Activation: *at reset*

Initialization: VB_MMV = 14 V

Recurrence: 10 ms

Formula section:

$VB\_MMV_k = VB\_MMV_{k-1} + (VB_k - VB\_MMV_{k-1}) \cdot C\_CRLC\_VB.$

Calibration data:

Name	**Dim**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
C_CRLC_VB	1	0...FFH	0...0.99609	3.91E-03	[-]
Low pass filter correlation constant for VB_MMV					

56.8 Main relay control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_RLY_MAIN	V/O	0...1H	0...1	1	[-]
Main relay control command state (0H = relay shall be switched off, 1H = relay shall be switched on)					
LV_RLY_MAIN_SYS	V	0...1H	0...1	1	[-]
System depending main relay control command state (0H = off, 1H = on)					
LV_IGK_OFF_ACK	O/V	0...1H	0...1	1	[-]
Acknowlegde IGK off state (0H = Pin level shall be switched to High, 1H = Pin level shall be switched to Low)					
T_IGK_OFF_ACK_DIAG_PLS	V	0...FFH	0...2.55	0.01	[s]
Timer for diagnosis pulse, in which acknowledge IGK off pin level is set to Low					
T_ACK_IGK_OFF	V	0...1FEH	0...5.1	0.01	[s]
Timer for setting pin level to Low in case of acknowledge IGK off					

Input data:

LV_IGK	T_PWL	C_T_MIN_PWL	PWL_LOCK_CDN
LV_ACT_RLY_MAIN_EXT_ADJ	LV_RLY_MAIN_EXT_ADJ	LV_ES	ECU_STATE
PWL_LOCK_CDN	LV_CUR_VCV_CTL_PRE_RUN	LV_RLY_ST_STST	LV_VAR_STST
LV_DC	STATE_ACK_IGK_OFF		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_RLY_MAIN_MAN_REQ	1	0...1H	0...1	1	[-]
LC for activation of manual switching of main relay (1 = active)					
LC_RLY_MAIN_MAN_ADJ	1	0...1H	0...1	1	[-]
LC for switching main relay (1 = active)					
C_T_ACK_IGK_OFF	1	0...1FEH	0...5.1	0.01	[s]
Timer calibration for control of LV_IGK_OFF_ACK-pin in case of IGK shutdown request					
LC_IGK_OFF_ACK	1	0...1H	0...1	1	[-]
LC for switching on/off functionality acknowledge IGK off (1= active)					
LC_IGK_OFF_DIAG_PLS	1	0...1H	0...1	1	[-]
LC for activate diagnosis pulse (1= activated)					
LC_IGK_OFF_DIAG_PLS_SWI	1	0...1H	0...1	1	[-]
LC for switch on diagnosis pulse in first start in DC or every start (1= diagnosis pulse activated every start)					
C_T_IGK_OFF_ACK_DIAG_PLS	1	0...FFH	0...2.55	0.01	[s]
Timer constant for diagnosis pulse, in which acknowledge IGK off pin level is set to Low					

Import actions:

ACTION_INFR_SetIGK_OFF(IN <lv_igk_off_ack>**)**
This action sets the output pin level for acknowledge IGK off / starter relay control for engine start stop automatic (STST)

56.8.1 Main relay control

FUNCTION DESCRIPTION:

General information:

Purpose of this module is to get an entry point to switch on or off the main relay and get a status of the control command through LV_RLY_MAIN.

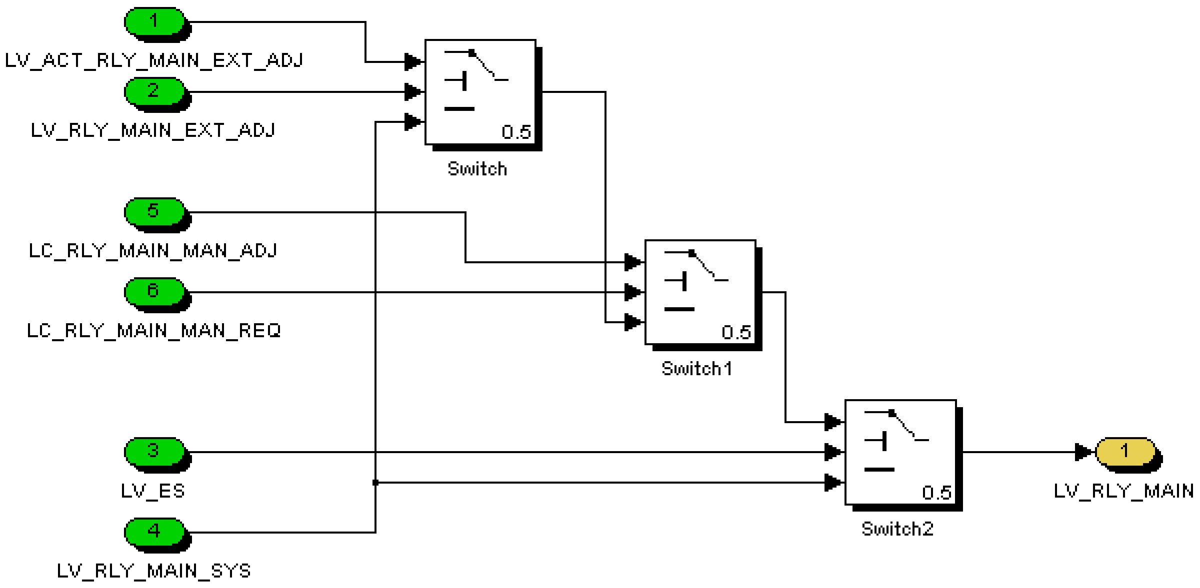
Upon ignition key transition from OFF to ON, the ECU voltage regulator is set to ON and the core microprocessor is reset. Then, so as to power up the system (system = ECU + actuators + sensors), the main relay has to be switched ON (most of actuators are supplied by the main relay).

Upon ignition key transition from ON to OFF, the ECU enters in the post-operating phase. At its end, the system shutdown is obtained by disabling the power supply on sensors and actuators. For that it is necessary to switch off the main relay. Then, the ECU can be switched off too.

To summarize, the main relay is switched on as soon as the ignition key is turned in the ON position and it remains ON until the ECU voltage regulator is switched off.

The system depending switching state is stored in the flag LV_RLY_MAIN_SYS. This request can be overruled by the request of the calibration system or tester (serial communication). To prevent destruction of components (actuators which are supplied via main relay), switching is only allowed when engine stopped (LV_ES =1). While engine is running, always the system depending switching state is used.

Signal flow diagram:



Application conditions:

Initialisation: all 0

Recurrence: 10 ms

Activation: at reset

Formula section:

```
    IF    (ECU_STATE = WAKE_UP and LV_CUR_VCV_CTL_PRE_RUN = 0)       or
          (    LV_IGK = 0                          and
               T_PWL > C_T_MIN_PWL                 and
               PWL_LOCK_CDN = 0 (mask out: NC_PWL_LOCK_CDN_NVMY,
               NC_PWL_LOCK_CDN_RLY_MAIN,
               NC_PWL_LOCK_CDN_BN2000,
               NC_PWL_LOCK_CDN_CUS)
               )

    THEN    LV_RLY_MAIN_SYS = 0
    ELSE    LV_RLY_MAIN_SYS = 1

IF(1)       LV_ES = 0
THEN(1)     LV_RLY_MAIN = LV_RLY_MAIN_SYS
ELSE(1)     IF(2)       LC_RLY_MAIN_MAN_REQ = 1
            THEN(2)     LV_RLY_MAIN = LC_RLY_MAIN_MAN_ADJ
            ELSE(2)     IF(3)       LV_RLY_MAIN_EXT_ADJ = 1
                        THEN(3)     LV_RLY_MAIN = LV_ACT_RLY_MAIN_EXT_ADJ
                        ELSE(3)     LV_RLY_MAIN = LV_RLY_MAIN_SYS
                        ENDIF(3)
            ENDIF(2)
ENDIF(1)

If    LV_RLY_MAIN = 1
Then
    Switch ON main relay        {set the status of the control command to ON }
Else
    Switch OFF main relay       {set the status of the control command to OFF}
End
```

56.8.2 Acknowledge IGK off

FUNCTION DESCRIPTION:

General information:

Purpose of this module is to send the acknowledge info via HW line for switching off KL15. The same pin is used in case of active engine start stop automatic (STST) in order to control the starter relay.

Application conditions:

Initialisation: at reset all 0
Recurrence: 10 ms
Activation: at every engine state
Deactivation: -

Formula section:

Definition of timer for pin control in case of diagnosis test pulse

```
If      LC_IGK_OFF_DIAG_PLS = 1           and    //switch for diagnosis pulse activated
        [LV_DC = 0 -> 1                   or     //diagnosis pulse only at first DC start
        (LC_IGK_OFF_DIAG_PLS_SWI = 1      and    //switch for diagnosis pulse mode
        LV_ST_END = 0 -> 1)]                     // diagnosis pulse at every start
then    T_IGK_OFF_ACK_DIAG_PLS = C_T_IGK_OFF_ACK_DIAG_PLS
endif
If      LV_ST_END = 1
then    T_IGK_OFF_ACK_DIAG_PLS = T_IGK_OFF_ACK_DIAG_PLS -10ms
else    T_IGK_OFF_ACK_DIAG_PLS = 0
endif
```

Defintion of timer for pin control in case of acknowledge IGK off

```
If      STATE_ACK_IGK_OFF(n -1) = 0       and
        STATE_ACK_IGK_OFF(n) = 1
then    T_ACK_IGK_OFF = C_T_ACK_IGK_OFF
endif

If      STATE_ACK_IGK_OFF = 1
then    T_ACK_IGK_OFF = T_ACK_IGK_OFF - 10ms
else    T_ACK_IGK_OFF = 0
endif
If      [(T_ACK_IGK_OFF > 0               or
        T_IGK_OFF_ACK_DIAG_PLS > 0)       and
        LC_IGK_OFF_ACK = 1]               or
        (LV_VAR_STST = 1                  and
        LV_RLY_ST_STST = 1)
then    LV_IGK_OFF_ACK = 1
else    LV_IGK_OFF_ACK = 0
endif
```

Control of HW-pin

ACTION_INFR_SetIGK_OFF(IN < LV_IGK_OFF_ACK >**)**

56.9 Alternator profile

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_TBL_ALTER[7][5][5]	V/O/S	0...FFH	0...99.60937	0.390625	[%]
Table with profile of alternator over alternator chip-temperature, load and speed (saved in NVMY)					
STATE_TBL_ALTER_DC[7][5][5]	V/O	0...FFFFH	0...65535	1	[s]
Table with profile of alternator over alternator chip-temperature, load and speed during driving cycle (in s)					
T_STATE_TBL_ALTER_DC	V	0...FFFFH	0...65535	1	[s]
Time since starting to write STATE_TBL_ALTER_DC					
T_STATE_TBL_ALTER	V/O/S	0...FFFFFFFFH	0...4294967295	1	[s]
Time since starting to write STATE_TBL_ALTER (saved in NVMY)					
LV_STATE_TBL_ALTER_INH	V	0...1H	0...1	1	[-]
Bit indicating that writing of STATE_TBL_ALTER_DC is stopped					

Input data:

TEMP_ALTER_MC	POW_REL_ALTER	N_ALTER	LV_ST_END
LV_ES	LC_AD_CLR	LC_AD_CLR_ALTER	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_STATE_TEMP_ALTER	7	1...7H	1...7	1	[-]
LDP_TEMP_ALTER_MC	7	0...1FH	-32...216	8	[°C]
Table with x-axis for STATE_TBL_ALTER and STATE_TBL_ALTER_DC					
ID_STATE_LOAD_ALTER	5	1...5H	1...5	1	[-]
LDP_POW_REL_ALTER	5	0...1FH	0...96.875	3.125	[%]
Table with y-axis for STATE_TBL_ALTER and STATE_TBL_ALTER_DC					
ID_STATE_N_ALTER	5	1...5H	1...5	1	[-]
LDP_N_ALTER	5	0...FFFFH	-32768...32767	1	[rpm]
Table with z-axis for STATE_TBL_ALTER and STATE_TBL_ALTER_DC					

FUNCTION DESCRIPTION:

This function delivers a alternator use profile over alternator load, alterantor chip temperatur and alterantor speed.

During a driving cycle (LV_ES = 0 and LV_ST_END = 1) the absolute temporary use profile STATE_TBL_ALTER_DC is calculated. It is a accumulated time of the related alternator operating points. Additional an timer is incremented (T_STATE_TBL_ALTER_DC) during calculation of STATE_TBL_ALTER_DC.

After a driving cycle (LV_ES 0->1) the relative use profile STATE_TBL_ALTER is calculated out of STATE_TBL_ALTER_DC, the old profile STATE_TBL_ALTER and the timer T_STATE_TBL_ALTER / _DC. This profile is saved in non volatile memory. The timer is calculated out of it's old value and the new timer T_STATE_TBL_ALTER_DC and is saved in non volatile memory. The timer T_STATE_TBL_ALTER and the profile STATE_TBL_ALTER show the behaviour of the alternator during the lifetime of the vehicle.

T_STATE_TBL_ALTER and STATE_TBL_ALTER are saved in NVMY during power latch phase.

Application conditions:

Initialisation: STATE_TBL_ALTER and T_STATE_TBL_ALTER from NVMY, all other = 0 at reset,

at LC_AD_CLR 0->1 **or** LC_AD_CLR_ALTER 0->1, T_STATE_TBL_ALTER = 0 and STATE_TBL_ALTER = 0

Recurrence: 1s

Activation: at every engine state

Formula section:

```
// Calculation of alternator profile during driving cycle:
If    LV_ES = 0  and  LV_ST_END = 1  and  LV_STATE_TBL_ALTER_INH = 0
Then        // calculation of STATE_TBL_ALTER_DC with x = ID_STATE_TEMP_ALTER,
            // y = ID_STATE_LOAD_ALTER, z = ID_STATE_N_ALTER:
            STATE_TBL_ALTER_DC[x][y][z] = STATE_TBL_ALTER_DC[x][y][z] + 1
            // increment timer:
            T_STATE_TBL_ALTER_DC ++
            // stop calculation if limit reached:
            If      T_STATE_TBL_ALTER_DC = 65335s
            Then    LV_STATE_TBL_ALTER_INH = 1
            Endif
Else        do nothing – calculation of STATE_TBL_ALTER_DC stopped
Endif
```

```
// Calculation of alterantor profile for NVMY:
If          LV_ES 0 ->1
Then        //write STATE_TBL_ALTER_DC in STATE_TBL_ALTER:
            For x=1:7
            {       For y=1:5
                    {       For z=1:5
                            {  STATE_TBL_ALTER[x][y][z] =
                               ( STATE_TBL_ALTER[x][y][z] * T_STATE_TBL_ALTER
                                 + STATE_TBL_ALTER_DC[x][y][z] )
                               / ( T_STATE_TBL_ALTER + T_STATE_TBL_ALTER_DC )
                            }
                    }
            }
```

```
            //calculate new time since starting to write STATE_TBL_ALTER:
            T_STATE_TBL_ALTER = T_STATE_TBL_ALTER + T_STATE_TBL_ALTER_DC
            //reset of temporary values:
            T_STATE_TBL_ALTER_DC = 0
            STATE_TBL_ALTER_DC[ ][ ][ ] = 0          // complete map = 0
            LV_STATE_TBL_ALTER_INH = 0
    Endif
```

56.10 BN2000: diagnosis of LV_IGK-Signal via CAN/via HW

56.10.1 Plausibility check between signal from CAn and HW-signal

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
ERR_SYM_PLAUS_IGK_BN	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
detected error symtom					
LV_ERR_PLAUS_IGK_BN	V/O	0...1H	0...1	1	[-]
plausibility error of IGK-signal from CAN/HW					
LV_CDN_DIAG_PLAUS_IGK_BN	V/O	0...1H	0...1	1	[-]
diagnosis condition					
LV_END_DIAG_PLAUS_IGK_BN	V/O	0...1H	0...1	1	[-]
end of diagnosis					

Input data:

STATE_PLAUS_IGK_BN	VB	LV_VAR_BN	

FUNCTION DESCRIPTION:

Description:

For BN2000-vehicles there are two possibilities of getting the LV_IGK-signal:

1) via CAN (from the control unit CAS)

2) via HW-signal (“Weckleitung”)

In order to detect a CAS error or a HW-error (e.g. shortcut to ground) a plausibility check has to be done between CAN-signal and HW-signal.

Error symptoms are defined to this diagnosis function as follows:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 │  │  │  └─> -   short cut to battery        (= SYM_0)
 │  │  └────> -   short cut to ground         (= SYM_1)
 │  └───────> -   CAS error                   (= SYM_2)
 └──────────> -   CAS-plausibility error      (= SYM_3)
```

Application conditions:

Initialisation: all 0 at reset

Recurrence: 10ms

Activation:

```
if      LV_VAR_BN = 1
Then    if      (VB > = C_VB_MIN_IGK_DIAG)
        then    LV_CDN_DIAG_PLAUS_IGK_BN = 1
        endif
Else    LV_CDN_DIAG_PLAUS_IGK_BN = 0
Endif
```

Deactivation:

```
if      (VB < C_VB_MIN_IGK_DIAG)        or
        at      State bus sleep
then    LV_CDN_DIAG_PLAUS_IGK_BN = 0
endif
```

Formula section:

```
IF      STATE_PLAUS_IGK_BN = 0H
THEN    ERR_SYM_PLAUS_IGK_BN = NO_SYM
        LV_ERR_PLAUS_IGK_BN = 0     after rebounce
ELSEIF  STATE_PLAUS_IGK_BN = 1H
THEN    ERR_SYM_PLAUS_IGK_BN = SYM_0            //SCB, Master CAN
        LV_ERR_PLAUS_IGK_BN = 1     after debounce
ELSEIF  STATE_PLAUS_IGK_BN = 2H
THEN    ERR_SYM_PLAUS_IGK_BN = SYM_1            //SCG, Master CAN
        LV_ERR_PLAUS_IGK_BN = 1     after debounce
ELSEIF  STATE_PLAUS_IGK_BN = 4H
THEN    ERR_SYM_PLAUS_IGK_BN = SYM_2            //CAS error, Master HW
        LV_ERR_PLAUS_IGK_BN = 1     after debounce
ELSEIF  STATE_PLAUS_IGK_BN = 8H
        LV_ERR_PLAUS_IGK_BN = 1     after debounce
THEN    ERR_SYM_PLAUS_IGK_BN = SYM_3            // CAN-signal not plausible
ENDIF
```

For calculation of condition-diagnosis-bit and end of diagnosis-bit LV_END_DIAG_PLAUS_IGK_BN see “Anti-bounce – algorithmus, calculation of the end of diagnosis”

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_PLAUS_IGK_BN	1	0...FFH	0...255	1	[-]
Debounce counter increment – igk-signal plausibility diagnosis					
C_ABC_MAX_PLAUS_IGK_BN	1	1...FFH	1...255	1	[-]
Debounce counter max. value – igk-signal plausibility diagnosis					
C_VB_MIN_IGK_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
VB threshold for de-/activation of IGK-diagnosis					

56.11 Main relay diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_RLY_MAIN	V/O	0...1H	0...1	1	[-]
Boolean for error detected at the main relay.					
T_RLY_MAIN_DIAG	V	0...FFH	0...2550	10	[ms]
Time window to perform the main relay diagnosis.					
T_DLY_RLY_MAIN_DIAG	V/O	0...FFH	0...2550	10	[ms]
delay Timer for the diagnosis to allow switching of the relay					
LV_CDN_DIAG_RLY_MAIN	V/O	0...1H	0...1	1	[-]
Diagnosis condition RLY_MAIN					
ERR_SYM_RLY_MAIN	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom RLY_MAIN					
LV_END_DIAG_RLY_MAIN	V/O	0...1H	0...1	1	[-]
End of Diagnosis RLY_MAIN					
LV_ERR_RLY_MAIN_DLY	V/O	0...1H	0...1	1	[-]
Error detected switch on relay					
LV_CDN_DIAG_RLY_MAIN_DLY	V/O	0...1H	0...1	1	[-]
Diagnosis condition RLY_MAIN_DLY					
ERR_SYM_RLY_MAIN_DLY	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom RLY_MAIN_DLY					
LV_END_DIAG_RLY_MAIN_DLY	V/O	0...1H	0...1	1	[-]
End of Diagnosis RLY_MAIN_DLY					
LV_DIAG_END_RLY_MAIN_DLY	V/O	0...1H	0...1	1	[-]
End of Diagnosis RLY_MAIN_DLY (flag not for error management)					
LV_RLY_MAIN_DLY_ERR	V/O	0...1H	0...1	1	[-]
Error detected switch on relay (flag not for error management)					
CTR_DIAG_RLY_MAIN_DLY	V	0...FFH	0...255	1	[-]
Debounce counter - main relay diagnosis for "switch on" diagnosis					
CTR_END_DIAG_RLY_MAIN_DLY	V	0...FFH	0...255	1	[-]
counter for END_DIAG - main relay diagnosis for "switch on" diagnosis					
CDN_DIAG_RLY_MAIN	-	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of RLY_MAIN bit 0: diagnosis condition for symptom SCP (SYM_0) bit1: diagnosis condition for symptom SCG(SYM_1) bit2 diagnosis condition for symptom OC (SYM_2)					
ERR_RLY_MAIN	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for RLY_MAIN (only Parameter)					
VDCDC_SOPC	V	0...7FFFH	0...639.98046	0.0195313	[V]
Output voltage of DCDC converter used for Switch-Off Path Check					
CTR_DIAG_VDCDC_SOPC	V	0...FFH	0...255	1	[-]
Debounce counter - DCDC output voltage diagnosis (error case)					
CTR_END_DIAG_VDCDC_SOPC	V	0...FFH	0...255	1	[-]
Debounce counter - DCDC output voltage diagnosis (non-error case)					

Input data:

LV_IGK	V_IGK	VB	LV_RLY_MAIN
PWL_LOCK_CDN			

Import actions:

ACTION_INFR_GetElDiagRlyMain(OUT<cdn_diag_rly_main>, **OUT**<err_rly_main>**)**
Return of the electric diagnosis for the main relay
ACTION_INFR_GetVDcdcSopc(OUT<PRM_VDCDC>)
Return the output voltage of the DCDC converter

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_RLY_MAIN_DIAG	1	0...FFH	0...2550	10	[ms]
Delay Time for the diagnosis to allow switching of the relay.					
C_V_IGK_RLY_MAIN_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
voltage threshold for detection of main relay on.					
C_VB_RLY_MAIN_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
voltage threshold for detection of battery voltage present.					
C_T_RLY_MAIN_DIAG	1	0...FFH	0...2550	10	[ms]
Time window to perform the main relay diagnosis					
C_ABC_INC_RLY_MAIN	1	0...FFH	0...255	1	[-]
Debounce counter increment - RLY_MAIN diagnosis					
C_ABC_MAX_RLY_MAIN	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - RLY_MAIN diagnosis					
C_VB_RLY_MAIN_DLY_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
Voltage threshold for main relay "too slow" diagnosis					
C_CTR_INC_DIAG_RLY_MAIN_DLY	1	0...FFH	0...255	1	[-]
Debounce counter increment - main relay "too slow" diagnosis					
C_CTR_MAX_DIAG_RLY_MAIN_DLY	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - main relay "too slow" diagnosis					
C_CTR_INC_END_DIAG_RLY_MAIN_DLY	1	0...FFH	0...255	1	[-]
counter increment for END_DIAG - main relay "too slow" diagnosis					
C_CTR_MAX_END_DIAG_RLY_MAIN_DLY	1	1...FFH	1...255	1	[-]
counter maximum value for END_DIAG - main relay "too slow" diagnosis					
C_VDCDC_SOPC_DIAG	1	0...7FFFH	0...639.98046	0.0195313	[V]
Voltage threshold for main relay "too slow" diagnosis					
C_CTR_INC_DIAG_VDCDC_SOPC	1	0...FFH	0...255	1	[-]
Debounce counter increment - main relay "too slow" diagnosis					
C_CTR_MAX_DIAG_VDCDC_SOPC	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - main relay "too slow" diagnosis					
C_CTR_INC_END_DIAG_VDCDC_SOPC	1	0...FFH	0...255	1	[-]
counter increment for END_DIAG - main relay "too slow" diagnosis					
C_CTR_MAX_END_DIAG_VDCDC_SOPC	1	1...FFH	1...255	1	[-]
counter maximum value for END_DIAG - main relay "too slow" diagnosis					

FUNCTION DESCRIPTION:

General information:

The Main Relay Diagnosis is performed to detect if the Main Relay has switched and remains on after Key-On and if it has switched off after the Power Latch Phase.

56.11.1 Main relay “switch on” / "switched off" diagnosis

Description:

The diagnosis is delayed for C_T_DLY_RLY_MAIN_DIAG to allow the relay to switch on/off. After this the actual VB is compared with a diagnostic threshold. If the error conditions are fulfilled the error is debounced.

If an error is detected after the Power Latch Phase the ECU is switched off after T_RLY_MAIN_DIAG has elapsed.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Main relays has not switched off     (= SYM_0)
 │ │ └───> Main relays has not switched on      (= SYM_1)
 │ └─────> -                                    (= SYM_2)
 └───────> -                                    (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 or reset or at clearing error memory but

T_RLY_MAIN_DIAG is calculated as described below

Recurrence:

The application recurrence for the Diagnosis is **10 ms**;
The „Main Relay Off“-Detection is performed only once after power latch.

Activation: see Formula section

Formula section:

Detection if Main Relay is on during Key-On

Call I/O Power-Stage-Diagnosis (Short to Ubatt) and ignore return value in order to restart the Power- Stage after shut off due to short circuit to battery

//this is done to try to wake up the power stage after instable battery voltage

ACTION_INFR_GetElDiagRlyMain (**OUT**<cdn_diag_rly_main>, **OUT**<err_rly_main>)

```
If(1)     LV_RLY_MAIN = 1                          and
          LV_END_DIAG_RLY_MAIN = 0                 and
          V_IGK  >  C_V_IGK_RLY_MAIN_DIAG
              //direct connection to the battery,
              to ensure a battery voltage that would be able to switch the relay
Then(1)       wait time:   C_T_DLY_RLY_MAIN_DIAG
              //time to allow the relay to switch, set only once. For continuos diagnosis set to 0
          If(2)       C_T_DLY_RLY_MAIN_DIAG        has run out  //if started, see above
          Then(2)     LV_CDN_DIAG_RLY_MAIN = 1
              If(3)       VB  <  C_VB_RLY_MAIN_DIAG
                          //connection to battery via main relay
              Then(3)     ERR_SYM_RLY_MAIN = SYM_1
                          LV_ERR_RLY_MAIN = 1      //after debounce
              Else(3)     ERR_SYM_RLY_MAIN = NO_SYM
                          LV_ERR_RLY_MAIN = 0      //after rebounce
                          //error can only be reset with LV_IGK 0->1 because diagnosis doesn't run after
                           an error has been detected !
              Endif(3)
```

```
            Else(2)     C_T_DLY_RLY_MAIN_DIAG               //is still running
            Endif(2)
    Else(1) LV_CDN_DIAG_RLY_MAIN = 0                        //no Diagnosis
    Endif(1)
    set lock condition once to extend power latch phase (PWL_LOCK_CDN)
```

Calculation end of diagnosis

LV_END_DIAG_RLY_MAIN is calculated by error management after diagnosis is active

Detection if Main Relay is off after Power Latch Phase:

```
If(1)       LV_RLY_MAIN = 0
Then(1)         LV_END_DIAG_RLY_MAIN = 0
                LV_ERR_RLY_MAIN = 0     //reset of error to be able to set it with new symptom
                start time:   C_T_DLY_RLY_MAIN_DIAG
                //time to allow the relay to switch
            If(2)       PWL_LOCK_CDN is set and
                        C_T_DLY_RLY_MAIN_DIAG has run out
            Then(2)     start  T_RLY_MAIN_DIAG  for  C_T_RLY_MAIN_DIAG
                        (start the time window during  the diagnosis is performed)
                If(3)       T_RLY_MAIN_DIAG  is still running
                Then(3)     LV_CDN_DIAG_RLY_MAIN = 1
                    If(4)       VB  >  C_VB_RLY_MAIN_DIAG
                                //connection to battery via main relay
                    Then(4)     ERR_SYM_RLY_MAIN = SYM_0
                                LV_ERR_RLY_MAIN = 1             //after debounce
                    Else(4)     ERR_SYM_RLY_MAIN = NO_SYM
                                LV_ERR_RLY_MAIN = 0             //after rebounce
                    Endif(4)
                Else(3)     LV_CDN_DIAG_RLY_MAIN = 0
                            erase lock condition for power latch phase (PWL_LOCK_CDN)
            Else(2)     T_RLY_MAIN_DIAG = 0                 //reset
                        LV_CDN_DIAG_RLY_MAIN = 0            //no switched off Diagnosis
            Endif(2)
    Else(1) LV_CDN_DIAG_RLY_MAIN = 0                        //no Diagnosis
    Endif(1)
```

56.11.2 Main relay “switched on too slow” diagnosis for pre-drive check

Description:

The main relay diagnosis is performed at reset to detect if the main relay has been switched on within the correct timing. This diagnostic is comparable to main relay switch on, but is separated for having faster reaction time.

Additionally, it is checked if the output voltage of the DCDC-converter has reached a minimum level. This minimum level is a necessary condition to perform the SOPC (switch off path check).

Once no failure has been found, diagnosis is then disabled.

The diagnosis starts after LV_IGK 0->1. After this the actual VB is compared with a diagnostic threshold. If the error conditions are fulfilled the error is debounced.

Error-symptoms are defined to this diagnosis function as following :

```
 ┌─┬─┬─┬─┐
 └─┴─┴─┴─┘
  │ │ │ └─> Main relay hasn't switched before predrive check    (= SYM_0)
  │ │ └───> -                                                   (= SYM_1)
  │ └─────> -                                                   (= SYM_2)
  └───────> DCDC converter voltage not high enough              (= SYM_3)
```

Application conditions:

Initialisation: ERRM data according filter type: **NO**

At reset : LV_CDN_DIAG_RLY_MAIN_DLY = 0
LV_ERR_RLY_MAIN_DLY = 0
ERR_SYM_RLY_MAIN_DLY = NO_SYM
CTR_END_DIAG_RLY_MAIN_DLY = 0
CTR_DIAG_RLY_MAIN_DLY = 0
LV_DIAG_END_RLY_MAIN_DLY = 0
LV_RLY_MAIN_DLY_ERR = 0
CTR_END_DIAG_VDCDC_SOPC = 0
CTR_DIAG_VDCDC_SOPC = 0

At clearing FMY: only reset of FMY variables according ERRM

Recurrence: 10 ms

Activation: LV_IGK = 1

Deactivation: LV_DIAG_END_RLY_MAIN_DLY = 1 **and** LV_RLY_MAIN_DLY_ERR = 0
set LV_CDN_DIAG_RLY_MAIN_DLY = 0

Formula section:

LV_CDN_DIAG_RLY_MAIN_DLY = 1

```
If(1)     VB  >=  C_VB_RLY_MAIN_DLY_DIAG
          (connection to battery via main relay)
Then(1)   ERR_SYM_RLY_MAIN_DLY = NO_SYM
              CTR_END_DIAG_RLY_MAIN_DLY =
              CTR_END_DIAG_RLY_MAIN_DLY  +
              C_CTR_INC_END_DIAG_RLY_MAIN_DLY (no overflow allowed)
              ACTION_INFR_GetVDcdcSopc(VDCDC_SOPC)
              If(2)   VDCDC_SOPC >= C_VDCDC_SOPC_DIAG
              Then(2)     CTR_END_DIAG_VDCDC_SOPC=
                          CTR_END_DIAG_VDCDC_SOPC +
                          C_CTR_INC_END_DIAG_VDCDC_SOPC
                          (no overflow allowed)
              Else(2)     ERR_SYM_RLY_MAIN_DLY = SYM_3 /* DCDC output voltage to low */
                      CTR_DIAG_VDCDC_SOPC =
                      CTR_ DIAG_VDCDC_SOPC +
                      C_CTR_INC_DIAG_VDCDC_SOPC (no overflow allowed)
              EndIf(2)
              If(2a)      CTR_END_DIAG_RLY_MAIN_DLY >=
                          C_CTR_MAX_END_DIAG_RLY_MAIN_DLY
                          AND
                          CTR_END_DIAG_VDCDC_SOPC >=
                          C_CTR_MAX_END_DIAG_VDCDC_SOPC
              Then(2a)    LV_END_DIAG_RLY_MAIN_DLY = 1
                          LV_DIAG_END_RLY_MAIN_DLY = 1
                          LV_ERR_RLY_MAIN_DLY = 0
                          LV_RLY_MAIN_DLY_ERR = 0
              Endif(2a)
              If(2b)      CTR_DIAG_VDCDC_SOPC >=
                          C_CTR_MAX_DIAG_VDCDC_SOPC
              Then(2b)    ERR_SYM_RLY_MAIN_DLY = SYM_3 /* DCDC output voltage to low */
                          LV_END_DIAG_RLY_MAIN_DLY = 1
                          LV_DIAG_END_RLY_MAIN_DLY = 1
                          LV_ERR_RLY_MAIN_DLY = 1
                          LV_RLY_MAIN_DLY_ERR = 1
                          CTR_END_DIAG_RLY_MAIN_DLY = 0
                          CTR_END_DIAG_VDCDC_SOPC = 0
              Endif(2b)
Else(1)   ERR_SYM_RLY_MAIN_DLY = SYM_0
```

/* not necessary to debounce VDCDC_SOPC: it is generated from VB! */

```
                CTR_DIAG_RLY_MAIN_DLY =
                CTR_ DIAG_RLY_MAIN_DLY  +
                C_CTR_INC_DIAG_RLY_MAIN_DLY (no overflow allowed)
                If(2b)      CTR_DIAG_RLY_MAIN_DLY >=
                            C_CTR_MAX_DIAG_RLY_MAIN_DLY
                Then(2b)    LV_END_DIAG_RLY_MAIN_DLY = 1
                            LV_DIAG_END_RLY_MAIN_DLY = 1
                            LV_ERR_RLY_MAIN_DLY = 1
                            LV_RLY_MAIN_DLY_ERR = 1
                            CTR_END_DIAG_RLY_MAIN_DLY = 0
                            CTR_END_DIAG_VDCDC_SOPC = 0
                Endif(2b)
Endif(1)
```

56.12 Electrical diagnosis of HW line acknowledge IGK off / starter relay for start stop automatic (STST)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_ACK_IGK_OFF	O/V	0...1H	0...1	1	[-]
Boolean for error currently present on ACK_IGK_OFF power stage					
LV_CDN_DIAG_ACK_IGK_OFF	V	0...1H	0...1	1	[-]
Diagnosis condition ACK_IGK_OFF power stage diagnosis					
ERR_SYM_ACK_IGK_OFF	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom ACK_IGK_OFF power stage diagnosis					
LV_END_DIAG_ACK_IGK_OFF	V	0...1H	0...1	1	[-]
End of diagnosis ACK_IGK_OFF power stage diagnosis					
CDN_DIAG_ACK_IGK_OFF	V	0...7H	0...7	1	[-]
Diagnosis condition ACK_IGK_OFF for each symptom					
ERR_DIAG_ACK_IGK_OFF	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for ACK_IGK_OFF (only parameter)					

Input data:

LV_IGK	LV_CDN_VB_OBD1	LV_IGK_OFF_ACK	LV_IGK_OFF_ACK_ENA
LC_IGK_OFF_ACK			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_ACK_IGK_OFF	1	0...FFH	0...255	1	[-]
Antibounce counter increment					
C_ABC_MAX_ACK_IGK_OFF	1	1...FFH	1...255	1	[-]
Maximum value for antibounce counter					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_ACK_IGK_OFF	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (= 2)					

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>, OUT<LV_ERR_XX>)
ACTION_INFR_GetElDiagIGK_OFF(OUT< Cdn_diag_ack_igk_off>, OUT< Err_diag_ack_igk_off >)

Note :

ACTION_INFR_GetElDiagIGK_OFF(), ACTION_INFR_SetElDiagIGK_OFF() are defined in the IRS (Infrastructure requirement specification)

FUNCTION DESCRIPTION:

General information:

The HW line acknowledge IGK off / starter relay for start stop automatic (STST) is driven by the ECU via an output driver. The failure detection is done by ECU Hardware.
The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements.

Description:

After activation conditions are met, the diagnosis activation is delayed for NC_PSD_DLY_ACK_IGK_OFF executions, to avoid the usage of wrong infrastructure information.

Error-symptoms and conditions: are defined to this diagnosis function as following

Diagnostic Instance		Symptoms		ABC type	CARB class
Description	**Name**	**Description**	**Nr**		
HW line acknowledge IGK_OFF / starter relay for start stop automatic (STST)	ACK_IGK_OFF	Sym 0 Short circuit to battery (SCP)	0	MPL_STD_INI	CC
		Sym 1 Short circuit to ground (SCG)	1		
		Sym 2 Open circuit (OC)	2		
		Sym 3 -	3		

Application conditions:

Initialisation: at transition LV_IGK 0 --> 1:
Set delay counter for NC_PSD_DLY_ACK_IGK_OFF

at reset:
Set delay counter for NC_PSD_DLY_ACK_IGK_OFF

Recurrence:
```
If      LV_IGK_OFF_ACK = 0
then    100 ms
else    20ms
endif
```

Activation: LC_IGK_OFF_ACK = 1 **and** LV_IGK_OFF_ACK_ENA = 1

Deactivation: LC_IGK_OFF_ACK = 0 **or** LV_IGK_OFF_ACK_ENA = 0

Formula section:

```
If      LV_IGK = 1            and
        LV_CDN_VB_OBD1 = 1
Then
```

Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_ACK_IGK_OFF and basic diagnosis conditions CDN_DIAG_ACK_IGK_OFF) received from the infrastructure:

ACTION_INFR_GetElDiagIGK_OFF (OUT<Cdn_diag_ack_igk_off>, OUT<Err_diag_ack_igk_off>)

Basic diagnosis conditions are set according infrastructure information: CDN_DIAG_ACK_IGK_OFF
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_ACK_IGK_OFF

```
        If Activation conditions are met for the NC_PSD_DLY_ACK_IGK_OFF
           recurrence of the diagnosis
        Then  [CDN_DIAG_ACK_IGK_OFF valid from Infrastructure]
        Else  CDN_DIAG_ACK_IGK_OFF = 0
        Endif

Else
        CDN_DIAG_ACK_IGK_OFF = 0
Endif
```

Failure filtering and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_ACK_IGK_OFF and ERR_DIAG_ACK_IGK_OFF.

ACTION_ERRM_FilterMulticondition (IN<ACK_IGK_OFF>, IN<CDN_DIAG_ACK_IGK_OFF>, IN<ERR_DIAG_ACK_IGK_OFF>, IN<C_ABC_INC_ACK_IGK_OFF>, IN<C_ABC_MAX_ACK_IGK_OFF>, OUT<LV_ERR_ACK_IGK_OFF>)
This algorithm determines:

ERR_DIAG_ACK_IGK_OFF	(detected error symptom for ACK_IGK_OFF diagnosis)
LV_ERR_ACK_IGK_OFF	(Error flag for debounced error of ACK_IGK_OFF)
LV_CDN_DIAG_ACK_IGK_OFF	(Diagnosis condition information)
LV_END_DIAG_ACK_IGK_OFF	(End of diagnosis information)

56.13 Functional diagnosis for generator switch off after start

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_GEN_DIAG	V/O	0...1H	0...1	1	[-]
Error generator diagnosis					
LV_CDN_DIAG_GEN_DIAG	V/O	0...1H	0...1	1	[-]
Status of diagnosis generator diagnosis					
LV_END_DIAG_GEN_DIAG	V/O	0...1H	0...1	1	[-]
End of diagnosis generator diagnosis					
ERR_SYM_GEN_DIAG	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Detected symptoms generator diagnosis					
CTR_REF_GEN_DIAG	V	0...200H	0...512	1	[-]
Reference counter for diagnosis result					
CTR_DIAG_GEN_DIAG	V	0...200H	0...512	1	[-]
Diagnosis counter for fulfilled diagnosis values					

Input data:

LV_IGK	VB_MES	LV_ALTER_SWI_OFF	POW_REL_ALTER_CLC
T_AST	LV_VB_JUMP	LV_ST_END	LV_ERR_BSD

General information:

The diagnosis is implemented to check, that the alternator has surely made an under excitation for several seconds after start end.

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 100 ms

Activation: For alternator 1: LV_IGK = 1 and LV_ST_END = 1 and
LV_ALTER_SWI_OFF = 1 and LV_ERR_BSD = 0

Deactivation: LV_END_DIAG_GEN_DIAG = 1

Formula section:

```
If  C_T_AST_GEN_DIAG_ON < T_AST ≤ C_T_AST_GEN_DIAG_OFF       AND
    LV_VB_JUMP = 0                                           AND
    ! (VB_MES > C_VB_MES_GEN_DIAG_THD_MAX          AND
    C_DFFGEN_THD_MIN < POW_REL_ALTER_CLC < C_DFFGEN_THD_MAX)
    Then  CTR_REF_GEN_DIAG ++
          If    VB_MES < C_VB_MES_GEN_DIAG_THD
                Then  CTR_DIAG_GEN_DIAG ++
          Endif
Endif
```

```
If   T_AST ≥ C_T_AST_GEN_DIAG_OFF
     Then  LV_CDN_DIAG_GEN_DIAG = 1
           If    (VB_MES > C_VB_MES_GEN_DIAG_THD_MAX  AND
                 (C_DFFGEN_THD_MAX ≤ POW_REL_ALTER_CLC       OR
                  POW_REL_ALTER_CLC ≤ C_DFFGEN_THD_MIN))
                 AND
                 CTR_REF_GEN_DIAG = !0                              AND
                 (CTR_DIAG_GEN_DIAG / CTR_REF_GEN_DIAG) * 100 % <
                 C_PERC_GEN_DIAG_VB_MAX
                 Then  ERR_SYM_GEN_DIAG = SYM_0
                       LV_ERR_GEN_DIAG = 1              (after debounce)
                 Else  ERR_SYM_GEN_DIAG = NO_SYM
                       LV_ERR_GEN_DIAG = 0              (after rebounce)
                 Endif
Endif
```

End of diagnosis calculation LV_END_DIAG_GEN_DIAG = 1:

→ *see chapter “Anti – bounce Algorithm: Calculation of the end of diagnosis”*

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_GEN_DIAG	1	0...FFH	0...255	1	[-]
Anti-bounce increment value					
C_ABC_MAX_GEN_DIAG	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value					
C_T_AST_GEN_DIAG_ON	1	0...FFFFH	0...6553.5	0.1	[s]
Begin of generator diagnosis T_AST					
C_T_AST_GEN_DIAG_OFF	1	0...FFFFH	0...6553.5	0.1	[s]
End of generator diagnosis T_AST					
C_VB_MES_GEN_DIAG_THD	1	0...3FFH	0...28.7055	2.81E-02	[V]
Battery voltage diagnosis threshold					
C_VB_MES_GEN_DIAG_THD_MAX	1	0...3FFH	0...28.7055	2.81E-02	[V]
Max. battery voltage threshold for diagnosis					
C_DFFGEN_THD_MIN	1	0...FFH	0...99.60937	0.390625	[%]
Min. DF-diagnosis threshold for generator diagnosis					
C_DFFGEN_THD_MAX	1	0...FFH	0...99.60937	0.390625	[%]
Max. DF-diagnosis threshold for generator diagnosis					
C_PERC_GEN_DIAG_VB_MAX	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Percentage of battery voltage values within diagnosis range					

56.14 Energy spare mode diag

56.14.1 LV_ERR_EGY_MIN for production and garage mode

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_EGY_MIN	V/O	0...1H	0...1	1	[-]
Energy spare mode (EGY_MIN - Energiesparmodus) error flag					
LV_CDN_DIAG_EGY_MIN	V	0...1H	0...1	1	[-]
Diagnosis condition EGY_MIN					
ERR_SYM_EGY_MIN	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom EGY_MIN					
LV_END_DIAG_EGY_MIN	V	0...1H	0...1	1	[-]
End of Diagnosis EGY_MIN					

Input data:

STATE_EGY_MIN_KWP	LV_IGK		

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis shall be to detect an activated energy spare mode. There are different energy spare modes. In this chapter the production and the garage mode are treated.

Error treatment:

Diagnostic Instance		Symptoms		ABC type	CARB class
Description	Name (XX)	Description	Nr		
Energy spare mode for production and garage mode (EGY_MIN - Energiesparmodus) error flag	EGY_MIN	Fertigungsmodus	0	NO	CC
			1		
		Werkstattmodus	2		
			3		

Possible configuration for ABC type and CARB class (see also “Table of Failure”)

ABC Type:	STD_INI, STD, MEM, MEM_INI, DEC_CAL, STC, NO, MPL_STD_INI
CARB Class:	MIS, FSD, CC, CAT, HC, EVAP, SA, AC, LS, LSH, EGR, OTHER, NO

Application conditions:

```
Initialisation:  all 0 at LV_IGK 0 -> 1 or reset
                 LV_END_DIAG_EGY_MIN is initialized by ERRM
Recurrence:      1s
Activation:      If      LV_IGK = 1 and LC_EGY_MIN_DIAG = 1
                 Then    LV_CDN_DIAG_EGY_MIN = 1
                 Endif
Deactivation:    If      LV_IGK = 0 or LC_EGY_MIN_DIAG = 0
                 Then    LV_CDN_DIAG_EGY_MIN = 0
                 Endif
```

Formula section:

```
If      STATE_EGY_MIN_KWP = 0 or STATE_EGY_MIN_KWP = 2
Then    LV_ERR_EGY_MIN = 0
        ERR_SYM_EGY_MIN = NO_SYM
Else    LV_ERR_EGY_MIN = 1
        If      STATE_EGY_MIN_KWP = 1
        Then    ERR_SYM_EGY_MIN = SYM_0
        Elseif  STATE_EGY_MIN_KWP = 3
        Then    ERR_SYM_EGY_MIN = SYM_2
        Endif
Endif
LV_END_DIAG_EGY_MIN = 1
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_EGY_MIN_DIAG	1	0...1H	0...1	1	[-]
Activation of energy spare mode diagnosis					

56.14.2 LV_ERR_EGY_MIN_2 for transportation mode

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_EGY_MIN_2	O/V	0...1H	0...1	1	[-]
Energy spare mode (EGY_MIN - Energiesparmodus) error flag for transportation mode					
LV_CDN_DIAG_EGY_MIN_2	V	0...1H	0...1	1	[-]
Diagnosis condition EGY_MIN for transportation mode					
ERR_SYM_EGY_MIN_2	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom EGY_MIN for transportation mode					
LV_END_DIAG_EGY_MIN_2	V	0...1H	0...1	1	[-]
End of Diagnosis EGY_MIN for transportation mode					

Input data:

STATE_EGY_MIN_KWP	LV_IGK		

FUNCTION DESCRIPTION:

General information:

The purpose of the diagnosis shall be to detect an activated energy spare mode. There are different energy spare modes. In this chapter the transportation mode is treated.

Error treatment:

Diagnostic Instance		Symptoms		ABC type	CARB class
Description	Name (XX)	Description	Nr		
Energy spare mode for transportation mode (EGY_MIN_2 - Energiesparmodus) error flag	EGY_MIN_2		0	NO	CC
		Transportmodus	1		
			2		
			3		

Possible configuration for ABC type and CARB class (see also “Table of Failure”)

ABC Type: STD_INI, STD, MEM, MEM_INI, DEC_CAL, STC, NO, MPL_STD_INI

CARB Class: MIS, FSD, CC, CAT, HC, EVAP, SA, AC, LS, LSH, EGR, OTHER, NO

Application conditions:

```
Initialisation:  all 0 at LV_IGK 0 -> 1 or reset
                 LV_END_DIAG_EGY_MIN_2 is initialized by ERRM
Recurrence:      1s
Activation:      If      LV_IGK = 1 and LC_EGY_MIN_DIAG_2 = 1
                 Then    LV_CDN_DIAG_EGY_MIN_2 = 1
                 Endif
Deactivation:    If      LV_IGK = 0 or LC_EGY_MIN_DIAG_2 = 0
                 Then    LV_CDN_DIAG_EGY_MIN_2 = 0
                 Endif
```

Formula section:

```
If      STATE_EGY_MIN_KWP = 2
Then    LV_ERR_EGY_MIN_2 = 1
        ERR_SYM_EGY_MIN_2 = SYM_1
Else    LV_ERR_EGY_MIN_2 = 0
        ERR_SYM_EGY_MIN_2 = NO_SYM
Endif

LV_END_DIAG_EGY_MIN_2 = 1
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_EGY_MIN_DIAG_2	1	0...1H	0...1	1	[-]
Activation of energy spare mode diagnosis for transportation mode					

56.15 Customer adaptation module: AGGR PWSL

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_bsdprot1	O/V	0...1H	0...1	1	[-]
condition for BSD protocol controller type 1					
B_bsdprot2	O/V	0...1H	0...1	1	[-]
condition for BSD protocol controller type 2					
B_crashgen	O	0...1H	0...1	1	[-]
condition alternator shut off at crash					
B_dev0detec	O/V	0...1H	0...1	1	[-]
bit 0 of learned LF_BSD_CPT_AVL					
B_dev1detec	O/V	0...1H	0...1	1	[-]
bit 1 of learned LF_BSD_CPT_AVL					
B_dev2detec	O/V	0...1H	0...1	1	[-]
bit 2 of learned LF_BSD_CPT_AVL					
B_dev3detec	O/V	0...1H	0...1	1	[-]
bit 3 of learned LF_BSD_CPT_AVL					
B_dev4detec	O/V	0...1H	0...1	1	[-]
bit 4 of learned LF_BSD_CPT_AVL					
B_dev5detec	O/V	0...1H	0...1	1	[-]
bit 5 of learned LF_BSD_CPT_AVL					
B_dev6detec	O/V	0...1H	0...1	1	[-]
bit 6 of learned LF_BSD_CPT_AVL					
B_dev7detec	O/V	0...1H	0...1	1	[-]
bit 7 of learned LF_BSD_CPT_AVL					
B_fabttreg	O/V	0...1H	0...1	1	[-]
condition note battery change					
B_fahisrst	O/V	0...1H	0...1	1	[-]
condition reset histogram					
B_fapmmess	O/V	0...1H	0...1	1	[-]
condition measure method PM					
B_genactiv	O/V	0...1H	0...1	1	[-]
condition alternator interface activ					
B_genfbsd	O/V	0...1H	0...1	1	[-]
communication fault BSD-alternator (old: LV_GEN_IF_OFF)					
B_genfel	O/V	0...1H	0...1	1	[-]
electrical fault alternator					
B_genfht	O/V	0...1H	0...1	1	[-]
(excess) temperature fault alternator					
B_genfme	O/V	0...1H	0...1	1	[-]
mechanical fault alternator					
B_ibsactiv	O/V	0...1H	0...1	1	[-]
Bedingung IBS erkannt					
B_ierr_ext	O/V	0...1H	0...1	1	[-]
Anforderung Grenzerregerstrom vom Tester					
B_jumpstart	O/V	0...1H	0...1	1	[-]
Bedingung Jumpstart					
B_kl15_ep	O/V	0...1H	0...1	1	[-]
1: ignition on					
B_kl15_vorab	O/V	0...1H	0...1	1	[-]
active during delayed ignition switch off					
B_loadre_ext	O/V	0...1H	0...1	1	[-]
Anforderung Loadresponse vom Tester					
B_lrfoff	O/V	0...1H	0...1	1	[-]
shut of the load response threshold for alternator (copy)					
B_lrreg1ak	O/V	0...1H	0...1	1	[-]
load response condition (for register 1) alternator activ					

B_rd_gentyp	O/V	0...1H	0...1	1	[-]
read alternator type and manufacturer					
B_sta	O/V	0...1H	0...1	1	[-]
flag indicating that automatic starter function is active					
B_ulcp	O/V	0...1H	0...1	1	[-]
Ultra-low-cost powermanagement enabled					
Batt_coort	O	0...FFH	0...255	1	[-]
position battery (read out of coding)					
Bsdcontrol	O	0...FFH	0...255	1	[-]
Communication coordination byte					
Bsddata	O	0...FFH	0...255	1	[-]
Content of register					
Bsdgencv	O	0...FFH	0...255	1	[-]
chip version alternator					
Bsdgenregv	O	0...FFH	0...255	1	[-]
controller version alternator					
Bsdregadr	O	0...FFH	0...255	1	[-]
Actual register number					
CTR_BSD_ERR_CHK_SUM_2	O	0...FFFFH	0...65535	1	[-]
Fehlerzähler EBSD Checksumme					
CTR_IBS_ERR_1	-	0...FFH	0...255	1	[-]
Dibs Fehlerzähler BSD					
CTR_IBS_ERR_2	-	0...FFFFH	0...65535	1	[-]
Dibs Fehlerzähler EBSD					
CTR_PWR_ERR_PREV_DIV_2	-	0...FFH	0...255	1	[-]
Vorteiler für Ibsderrs2					
CUR_ALTER_EXCT_LIM_SP	O	0...1FH	0...7.75	0.25	[A]
Setpoint alternator exciting current limitation					
CUR_ALTER_MAX	O/V	0...FFH	0...255	1	[A]
maximum current alternator					
CUR_GEN	O/V	0...FFH	0...255	1	[A]
Short term solution for satisfaction of CAN, representing the alternator current (will be replaced by CUR_GEN)					
Dfgrob	O	0...FFH	0...255	1	[-]
DF rough value (occupancy factor) from alternator					
Dfsiggen	O	0...FFH	0...99.60937	0.390625	[%]
occupancy of alternator					
Gen_manufak	O	0...FFH	0...255	1	[-]
manufacturer identifier alternator					
Gen_typkenn	O	0...FFH	0...255	1	[-]
type identifier alternator alternator					
IBS_COD	O	0...FFH	0...255	1	[-]
IBS Hardwareversion					
IBS_SW_BASE	O	0...FFH	0...255	1	[-]
IBS Softwarebaseline					
Ierr	O	0...FFH	0...31.875	0.125	[A]
exciting current alternator					
Ierr_grenz_ext	O	0...FFH	0...31.875	0.125	[A]
Grenzerregerstrom vom Tester					
Ierrfgrenz	O	0...FFH	0...31.875	0.125	[A]
actual value exciting current limitation alternator					
LOAD_BAT	O/V	0...FFH	0...99.60937	0.390625	[%]
DF-Monitor for accumulator charge in %					
LOAD_CDN_BAT	O/V/S	0...FFH	0...99.60937	0.390625	[%]
condition of battery load (for serial communication)					
LV_ALTER_SWI_OFF	O/V	0...1H	0...1	1	[-]
Logical variable indicating 'alternator switched off =1'					
LV_LOAD_RESP_ALTER_THD_ACT_SP	O/V	0...1H	0...1	1	[-]
shut of the load response threshold for alternator (setpoint)					
LV_N_SP_IS_POW_ACT	O/V	0...1H	0...1	1	[-]
1: idle speed increase due to low accumulator charge					

LV_N_SP_IS_PWR_STAB	O/V	0...1H	0...1	1	[-]
request for idle speed setpoint due to power management request for power supply stability					
N_ALTER	O/V	8000...7FFFH	-32768...32767	1	[rpm]
alternator speed					
N_ST_POW_MOD	O/V	0...FFH	0...8160	32	[rpm]
engine speed at start for power modul					
NR_ADD_IBS	O	0...FFFFFFFFH	0...4294967295	1	[-]
Zusbaunummer					
NR_IBS	O	0...FFFFFFFFH	0...4294967295	1	[-]
Seriennummer IBS					
POW_REL_ALTER_CLC	O/V	0...FFH	0...99.60937	0.390625	[%]
calculated DF-signal from alternator					
STATE_ALTER	O/V	0...FFH	0...255	1	[-]
Alternator status					
STATE_CUR_ENG_CNS	O/V	8000...7FFFH	-32768...32767	1	[A]
current consumption of engine					
STATE_IBS_SW_CHG	O	0...FFH	0...255	1	[-]
IBS SW-Änderungsstatus					
T_LOAD_RESP_ALTER_SP	O/V	0...3H	0...15	5	[s]
Setpoint for alternator 1 load response time					
Tchip	O	8000...7FFFH	-3276.8...3276.7	0.1	[°C]
chip temperature alternator					
Td_f_ierr	O	0...FFH	0...2.55	0.01	[s]
calibratible filter time for exciting current (alternator powermanagement)					
Td_rdgenreg2	O	0...FFH	0...25.5	0.1	[s]
calibratible time period for reading register 2 (alternator powermanagement)					
TEMP_ALTER	O/V	0...FEH	-48...142.5	0.75	[°C]
Temperature alternator (for environmental fault memory)					
Tlrfgen	O	0...FFH	0...25.5	0.1	[s]
actual value alternator load response time					
Tlrgen_ext	O	0...FFH	0...25.5	0.1	[s]
Loadresponse vom Tester					
TQ_LOSS_ALTER	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque consumption of alternator at crankshaft *					
U_fgen	O	0...FFFFH	0...6553.5	0.1	[V]
received voltage alternator (copy of voltage setpoint send to the bus)					
U_gensollpm	O	8000...7FFFH	-128...127.99609	3.9062e-3	[V]
Alternator voltage setpoint from power module					
Ub	O	0...FFFFH	0...6173.397	0.0942	[V]
battery voltage					
Uregnom	O	0...FFFFH	0...6553.5	0.1	[V]
nominal voltage alternator					
V_ALTER_SP	O/V	0...3FH	10.6...16.9	0.1	[V]
setpoint alternator 1 voltage					
V_LEVEL	O/V	0...FFH	0...255	1	[-]
voltage level					
VB_POW_MNG	O/V	0...FFFH	0...61.425	0.015	[V]
current battery voltage					
W_ub	O	0...FFH	0...24.021	0.0942	[V]
battery voltage raw value					

Input data:

ALTER_COD_0	ALTER_COD_1	ALTER_COD_2	ALTER_COD_3
B_bns	B_genoff	B_lroff	B_nsub
B_sleepwait	BSD_SENS_BAT_SMT_ADR	BSD_SENS_BAT_SMT_INFO	C_T_RD_PER_ALTER_TBL_2
CUR_ALTER_EXCT	CUR_ALTER_EXCT_LIM	Dffgen	Dfmonitor

ECU_STATE	I_craweng	I_gen	I_genmax
Ibhwversi	Ibsderrs1	Ibsderrs2	Ibspreco2
Ibswbase	Ibswchang	Ierrgrenz	Ktupcsctr
LF_BSD_CPT_AVL	LF_BSD_SENS_BAT_SMT_CTL	LV_ALTER_BSD_PROT_2	LV_ALTER_ERR_MEC
LV_ALTER_ERR_TEMP	LV_ALTER_ERR_IF	LV_ALTER_IF_ACT	LV_IGK
LV_IGK_PREL	LV_LOAD_RESP_ALTER_CND_1	LV_LOAD_RESP_ALTER_THD_ACT	LV_POW_MNG_BAT_CHG
LV_POW_MNG_HIS_RST	LV_POW_MNG_MES_MOD	LV_PWL_LOCK_CDN_CUS_INH	LV_SENS_BAT_SMT_ACT
LV_SENS_BAT_SMT_DET	LV_ST	LV_VAR_BN	LV_VAR_VEH
LV_VB_JUMP	Md_gennm_na	N_gen	N_nstart
NC_PWL_LOCK_CDN_CUS	POW_REL_ALTER	POW_REL_ALTER_L_RES	PWL_LOCK_CDN
Snibs	St_gen	STATE_EFP_CRASH_CAN	T_FIL_CUR_EXCT_ALTER
T_FIL_CUR_EXCT_ALTER	T_LOAD_RESP_ALTER	TEMP_ALTER_MC	Tgen
Tlrgen	U_gen	Ubt	Ulev
V_ALTER	V_ALTER_NOM	V_GEN_TAR	VB
VB_BAS	Zbibs		

56.15.1 Outputs for BMW functions which are not defined as PWSL exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the output are different from the specified values due to the input data attributes.

General information:

In case of vehicle variant E60 the DME takes over certain functionalities of the former “Powermodul”-ECU. If no IBS is implemented in the car these functionalities are calculated by the ultra-low-cost-powermanagement (an add-on to the actual powermanagement).

Application conditions:

Initialisation: all 0 after reset

except:

ALTER_COD_0 = 7 H // initalisation to max value

ALTER_COD_1 = 1F H // initalisation to max value

```
If LV_VAR_BN = 0  then                // for  CAN 11
        B_kl15_ep = 1
        B_kl15_vorab = 1
endif
```

Ub = Batt_coort = W_ub = 1st valid ADC-value

U_fgen x = 1st calculated value

Gen_manufak = Gen_typkenn = 255

B_rd_gentyp = 1

B_dev0..7detec = LF_BSD_CPT_AVL

Recurrence : all BSD driver timing

except:

Batt_coort = once at ECU init

10 ms: B_crashgen, W_ub, B_kl15_vorab, B_sta, B_jumpstart

20 ms: B_kl15_ep, Ub

50 ms: Gen_typkenn, Gen_manufak

100 ms: VB_POW_MNG, ulev, B_fabttreg, B_fahisrst, B_fapmmess,
U_gensollpm, B_ulcp , Cur_gen, B_dev0..7detec

Once only,

never changed: Ierr_grenz_ext = 31.87 A
Tlrgen_ext , B_ierr_ext, B_loadre_ext = 0

Activation: *every ECU – State (including Wakeup)*

Formula section:

Remark: all formulas are valid in a physical meaning

B_dev0..7detec = LF_BSD_CPT_AVL
B_kl15_ep = LV_IGK
B_kl15_vorab = LV_IGK_PREL
B_lrreg1ak = LV_LOAD_RESP_ALTER_CND_1
B_genactiv = LV_ALTER_IF_ACT
B_genfel = LV_ALTER_ERR_EL
B_genfme = LV_ALTER_ERR_MEC
B_genfht = LV_ALTER_ERR_TEMP
B_genfbsd = LV_ALTER_ERR_IF
B_sta = LV_ST
B_ierr_ext = 0
B_loadre_ext = 0
B_fabttreg = LV_POW_MNG_BAT_CHG
B_fahisrst = LV_POW_MNG_HIS_RST
B_fapmmess = LV_POW_MNG_MES_MOD
B_ibsactiv = LV_SENS_BAT_SMT_ACT

If LV_ALTER_BSD_PROT_2 = 1

```
  Then     B_lrfoff          = ! LV_LOAD_RESP_ALTER_THD_ACT(n-1)
  Else     B_lrfoff = ! LV_LOAD_RESP_ALTER_THD_ACT
Endif


Batt_coort        = C_BAT_PSN
Bsddata           = BSD_SENS_BAT_SMT_INFO
Bsdregadr         = BSD_SENS_BAT_SMT_ADR
Bsdcontrol        = LF_BSD_SENS_BAT_SMT_CTL
Bsdgenregv        = ALTER_COD_2
Bsdgencv          = ALTER_COD_3
Dfsiggen          = POW_REL_ALTER
Dfgrob            = POW_REL_ALTER_L_RES
Gen_manufak       = ALTER_COD_0
Gen_typkenn       = ALTER_COD_1
If ECU_STATE = "PWL"
Then  Ierr = 0
Else  Ierr = CUR_ALTER_EXCT
    B_jumpstart = LV_VB_JUMP
endif
Ierrfgrenz        = CUR_ALTER_EXCT_LIM
Tchip             = TEMP_ALTER_MC
Td_f_ierr         = T_FIL_CUR_EXCT_ALTER
Td_rdgenreg2      = C_T_RD_PER_ALTER_TBL_2
U_fgen            = V_ALTER
U_gensollpm       = V_GEN_TAR
Uregnom           = V_ALTER_NOM
VB_POW_MNG        = Ubt
Ub                = VB
ulev              = Ulev
W_ub              = VB_BAS * 5,747
```

Hint: Conversion from 0...5V (10 bit) of VB_BAS to 0...24V (8 bit) of W_ub.
(HW potential divider 28.7 V / 26 V; 5,747 = 28.7 V / 5 V)

```
If    STATE_EFP_DRASH_CAN = 2
then        B_crashgen   = 1
else        B_crashgen   = 0
endif
```

```
If    LV_VAR_BN = 1      and   LV_VAR_VEH = 0    and
      (LV_SENS_BAT_SMT_DET = 0 or LC_PWSL_ENA = 1)
  then    B_ulcp = 1
  else    B_ulcp = 0
endif

If        LV_ALTER_BSD_PROT_2  = 1
  then    B_bsdprot1           = 0     and
          B_bsdprot2           = 1
  else    B_bsdprot1           = 1     and
          B_bsdprot2           = 0
endif
```

Tlrfgen = T_LOAD_RESP_ALTER (according the table~~s~~ below)

for writing into register 1 (three bit's for information reserved)

ID_KFTLRGEN2_INV	n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	output data: T_lrfgen
tn [sec]	0	0,2	0,4	3	6	9	12	15	input data: T_LOAD_RESP_ALTER

Depending on ALTER_COD_0 and ALTER_COD_1 the load response should be read/writen out/into register 1 (see fkt-LR-write). tn [sec] can't be send by BSD directly because the number of protocoll bits are limited. This bit combination has to be translated according the tables. Translation from bit combination to tn [sec] and vice versa. If the input data is between two break points, then the minimum break point must be used.

56.15.2 Outputs for SV Aggregates, customer→SV

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: all 0 after reset

Except:

V_ALTER_SP = 1^{st} calculated value

T_LOAD_RESP_ALTER_SP = 1^{st} calculated value

Recurrence:

1000ms: LV_N_SP_IS_POW_ACT, LOAD_BAT, LOAD_CDN_BAT, ibhwversi, ibswbase, ibswchang, snibs, zbibs, B_sleepwait

100ms: V_ALTER_SP, T_LOAD_RESP_ALTER_SP, LV_ALTER_SWI_OFF, STATE_CUR_ENG_CNS, LV_LOAD_RESP_ALTER_THD_ACT_SP, st_i_gen,

STATE_ALTER, st_gen, CUR_ALTER_MAX,
TEMP_ALTER , , ibsderrs1, ibsderrs2, ibspreco2, ktupcsctr

10ms: N_ST_POW_MOD, TQ_LOSS_ALTER, POW_REL_ALTER_CLC, CUR_ALTER_EXCT_LIM_SP, T_LOAD_RESP_ALTER_SP, LV_N_SP_IS_PWR_STAB, N_ALTER

Activation: every ECU – State (including Wakeup)

Formula section:

Remark: all formulas are valid in a **physical** meaning

LV_ALTER_SWI_OFF = B_genoff

CUR_ALTER_EXCT_LIM_SP = Ierrgrenz

```
If      CUR_ALTER_EXCT_LIM_SP == 0 and LC_ALTER_EXCT_LIM_SP == 1
Then    CUR_ALTER_EXCT_LIM_SP = ID_ALTER_EXCT_LIM_SP
Else    CUR_ALTER_EXCT_LIM_SP = CUR_ALTER_EXCT_LIM_SP
Endif
```

CUR_ALTER_MAX = I_genmax

LOAD_CDN_BAT = Dfmonitor

LOAD_BAT = Dfmonitor

LV_N_SP_IS_POW_ACT = B_nsub

LV_LOAD_RESP_ALTER_THD_ACT_SP = ! B_lroff

N_ALTER = N_gen

N_ST_POW_MOD = N_nstart

POW_REL_ALTER_CLC = Dffgen

STATE_ALTER = St_gen

st_gen = STATE_ALTER (short time solution for satisfact. of CAN)

STATE_CUR_ENG_CNS = I_craweng

CUR_GEN = I_gen

TQ_LOSS_ALTER = Md_gennm_na

TEMP_ALTER = Tgen

V_ALTER_SP = U_gen

T_LOAD_RESP_ALTER_SP = Tlrgen (according the translation tables below)

for writing into register 0 (two bit's for information reserved)

ID_KFTLRGEN1	n=0	n=1	n=2	n=3	input data: T_lrgen
tn [sec]	0	3	6	9	output data: T_LOAD_RESP_ALTER_SP

for writing into register 1 (three bit's for information reserved)

ID_KFTLRGEN2	n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	input: T_lrgen
tn [sec]	0	0,2	0,4	3	6	9	12	15	output data: T_LOAD_RESP_ALTER_SP

Depending on ALTER_COD_0 and ALTER_COD_1 the load response should be read/written out/into register 0 or register 1 (see fkt-LR-write). tn [sec] can't be send by BSD directly because the number of protocoll bits are limited. This bit combination has to be translated according the tables. Translation from bit combination to tn [sec] and vice versa. If the input data is between two break points, then the minimum break point must be used.

ibhwversi = Ibhwversi

ibsderrs1 = Ibsderrs1

ibsderrs2 = Ibsderrs2

ibspreco2 = Ibspreco2

ibswbase = Ibswbase

ibswchang = Ibswchang

ktupcsctr = Ktupcsctr

snibs = Snibs

zbibs = Zbibs

LV_N_SP_IS_PWR_STAB = B_bns

```
IF      B_sleepwait = 1        and
        LV_PWL_LOCK_CDN_CUS_INH = 0
THEN    PWL_LOCK_CDN[NC_PWL_LOCK_CDN_CUS] = 1
        //"set power-latch lock condition corresponding to
        NC_PWL_LOCK_CDN_CUS in PWL_LOCK_CDN"
ELSE    PWL_LOCK_CDN[NC_PWL_LOCK_CDN_CUS] = 0
        //"reset power-latch lock condition corresponding to
        NC_PWL_LOCK_CDN_CUS in PWL_LOCK_CDN"
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_BAT_PSN	1	0...FFH	0...255	1	[-]
Position of battery in vehicle					
ID_ALTER_EXCT_LIM_SP	8	0...1FH	0...7.75	0.25	[A]
LDP_ALTER_COD_0	8	0...7H	0...7	1	[-]
Setting of CUR_ALTER_EXCT_LIM_SP in dependence of ALTER_COD_0					
LC_ALTER_EXCT_LIM_SP	1	0...1H	0...1	1	[-]
logical constant: software swich to choose BSD					
LC_PWSL_ENA	1	0...1H	0...1	1	[-]
Ultra-low-cost powermanagement enabled					

57 Secondary air

Table of Contents

Index

57.1 AGGR SAIR adaptation: Secondary Air

57.1.1 Outputs for BMW functions which are not defined as SAIR exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_fasls	O/V	0...1H	0...1	1	-
1: activation of secondary air pump is possible					
B_genslp	O/V	0...1H	0...1	1	-
1: secondary air pump is active (condition for voltage request SAP)					
B_sls	O/V	0...1H	0...1	1	-
1: secondary air pump is active					
Md_na_sls	O/V	8000...7FFFH	-3276.8 ...3276.7	0.1	Nm
torque losses secondary air pump					
Ms_sl	O/V	0...FFFFH	0...2047.96875	0.03125	kg/h
Massenstrom Sekundärluft					

Input data:

SAF_KGH			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation: all variables are initialisied with 0

Recurrence: **200 ms:** Ms_sl

Activation: every engine state

In the power latch phase:

0: Ms_sl

Formula section:

B_fasls = 0

B_genslp =0

B_sls = 0

Md_na_sls = 0 Nm

Ms_sl = SAF_KGH

58 Steering system

Chapter Steering system		Baseline 4DC3940S		
		Date	Department	Sign
Designed by		2008-07-01		
Released by		2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl			
	Document Key E002-190.49.02 SPE 000 48.0			Pages 8916 of 9643

Table of Contents

Index

58.1 AGGR STSY adaptation

58.1.1 Outputs for BMW functions which are not defined as STSY exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Md_na_sl	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Torque losses for power steering					
Mdi_res_afs	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Engine torque reserve active front steering					
Mdi_res_sl	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Engine torque reserve for power steering					

Input data:

TQ_ADD_PSTE_2	TQ_ADD_PSTE	TQ_LOSS_PSTE	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation: 0 Nm

Recurrence : 10 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

Md_na_sl = TQ_LOSS_PSTE

Mdi_res_afs = TQ_ADD_PSTE_2

Mdi_res_sl = TQ_ADD_PSTE

58.2 Power Steering acquisition variable

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
ANG_PSTE_END_LE_AD	O/V/S	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW]
relative steering wheel sensor left side adaptation					
ANG_PSTE_END_RI_AD	O/V/S	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW]
relative steering wheel sensor angle right					
ANG_PSTE_STND	O/V	8000... 7FFFH	-100... 99.9969482422	3.05176e-3	[%]
Relative steering wheel sensor angle					
LV_STG_RI_ACT	V	0... 1H	0... 1	1	[-]
LV indicating steering active (direction: 1 = right , 0 = left)					
LV_VEL_ANG_PSTE_MIN_DET	V	0... 1H	0... 1	1	[-]
LV indicating "released" steering wheel					
T_VEL_ANG_PSTE	V	0... FFH	0... 2.55	0.01	[s]
Time for detecting "released" steering wheel					
VEL_ANG_PSTE_COR	O/V	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
Filtered steering-wheel-angle velocity					
VEL_ANG_PSTE_MMV	V	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
moving mean value of steering-wheel-angle velocity					

Input Data:

LV_IGK	ANG_PSTE	VEL_ANG_PSTE	VS
LV_ERR_ANG_PSTE_CAN			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_ANG_PSTE_AD_LGRD	1	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW]
ANG_PSTE_AD incrementation					
C_ANG_PSTE_END_DIF	1	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW]
cal. difference between ANG_PSTE_END_LE/RI_AD out of NVMY and init-value (at reset)					
C_ANG_PSTE_END_LE_INI	1	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW]
initialisation value left ANG_PSTE_STND (positiv)					
C_ANG_PSTE_END_RI_INI	1	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW]
initialisation value right ANG_PSTE_STND (negativ)					
C_CRLC_VEL_ANG_PSTE	1	0... FFH	0... 0.99609375	3.90625e-3	[-]
Correlation constant for steering wheel angle velocity					
C_T_ANG_PSTE_AD	1	0... FFFFH	0... 655.35	0.01	[s]
Time during which adaptation of steering wheel angle is inhibited					
C_T_VEL_ANG_PSTE	1	0... FFH	0... 2.55	0.01	[s]
Time for detecting "released" steering wheel					
C_VEL_ANG_PSTE_MAX_AD	1	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
VEL_ANG_PSTE-threshold for adaptation of steering wheel angle					

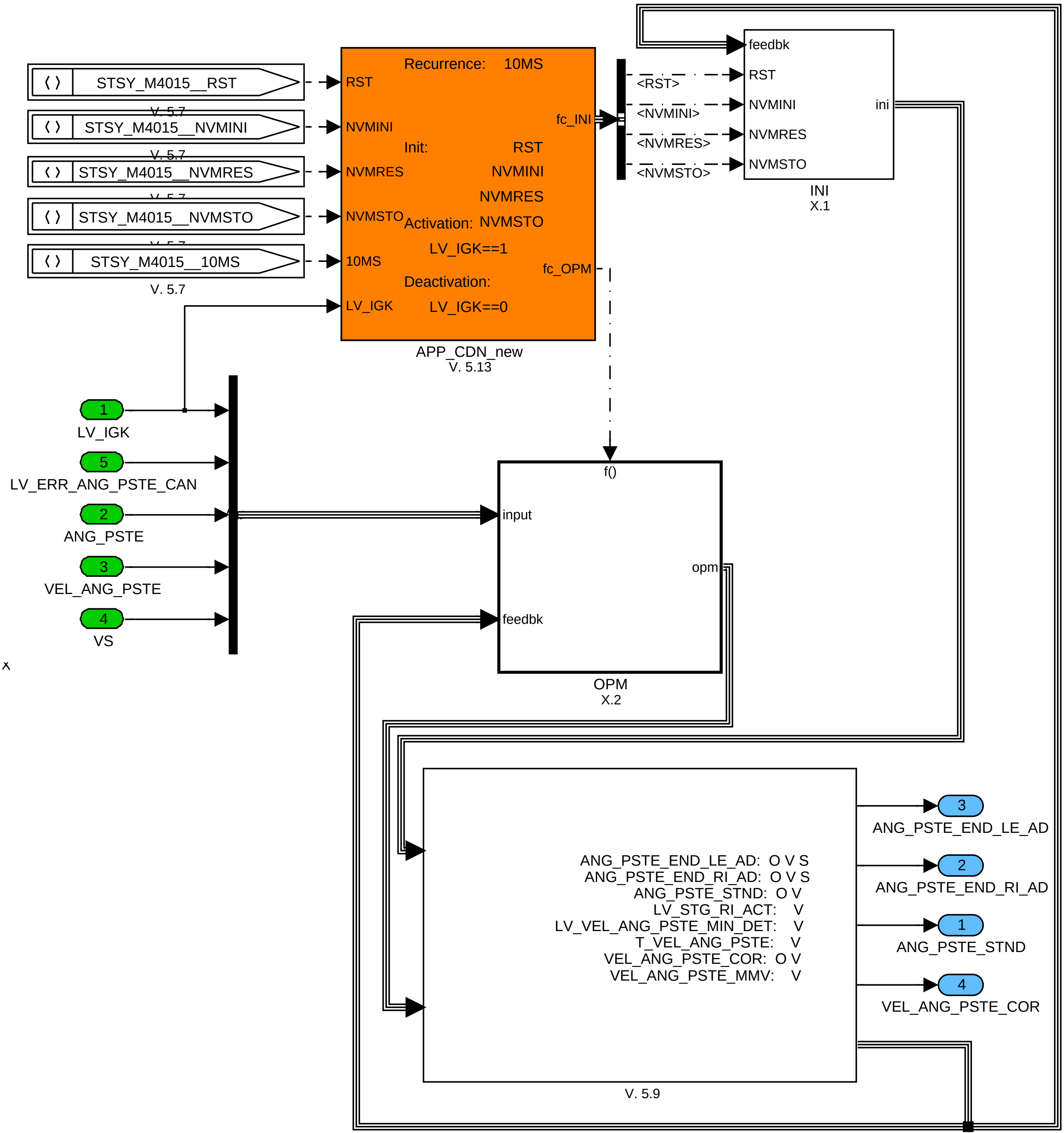
C_VEL_ANG_PSTE_MIN_1	1	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
Minimum threshold VEL_ANG_PSTE_MMV for detection of steering active					
C_VEL_ANG_PSTE_MIN_2	1	8000... 7FFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
Minimum threshold VEL_ANG_PSTE_MMV for detection of "released" steering wheel					
C_VS_PSTE	1	0... FFH	0... 255	1	[km/h]
Vehicle speed threshold for power steering					
LC_VEL_ANG_PSTE_COR	1	0... 1H	0... 1	1	[-]
Calculation of VEL_ANG_PSTE_COR depending on steering wheel state					

General Information

Application Conditions

Initialization: RST, NVMINI, NVMRES, NVMSTO
Recurrence: 10MS
Activation: LV_IGK==1
Deactivation: LV_IGK==0

Function description



SDA_SRS / SDA V 5.0.4 04−Apr−2006

Figure 1:
Path: STSY_M4015

58.2.1 Initialisation

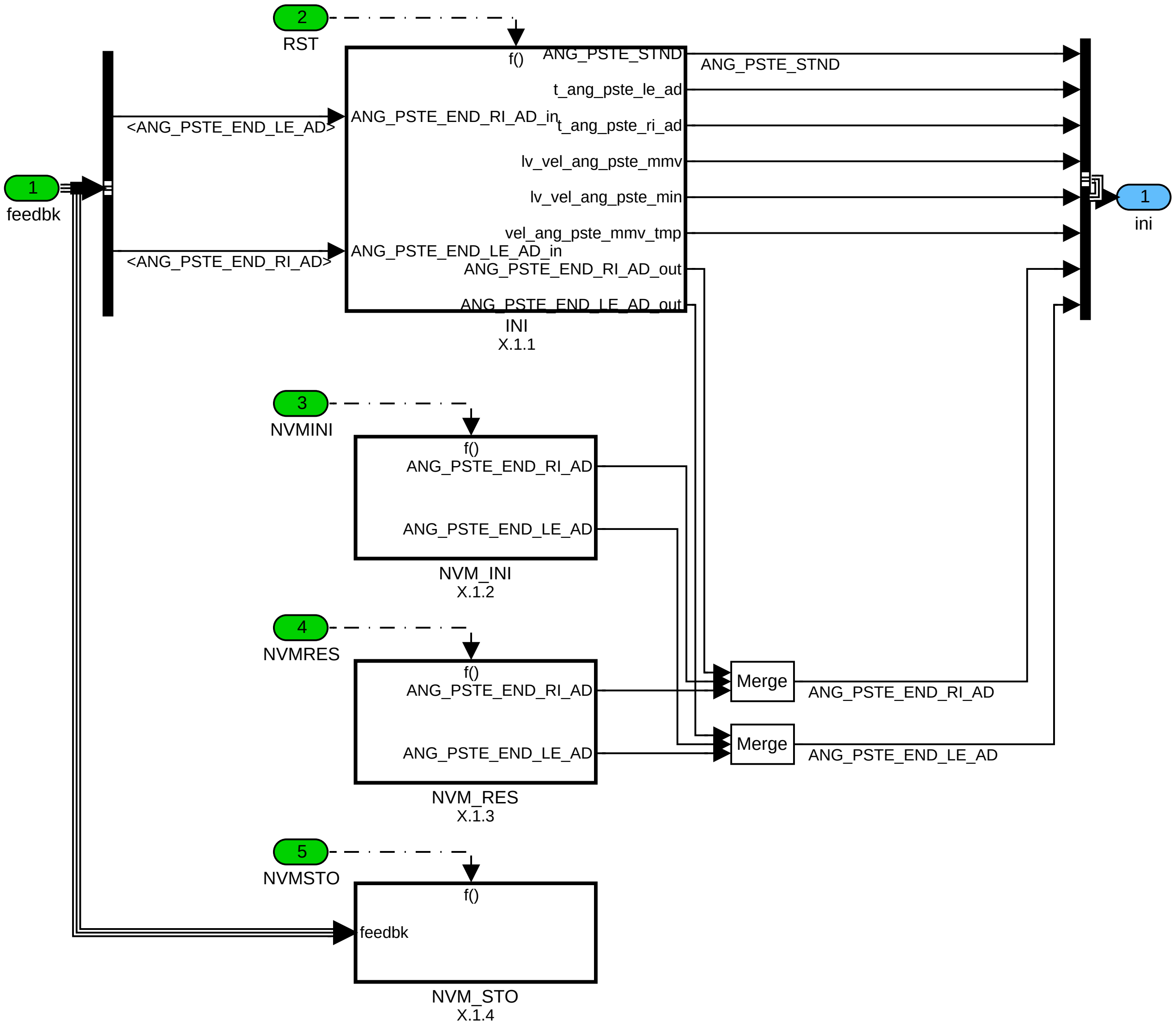


Figure 2:
Path: STSY_M4015/INI

58.2.1.1 Initialisation at reset

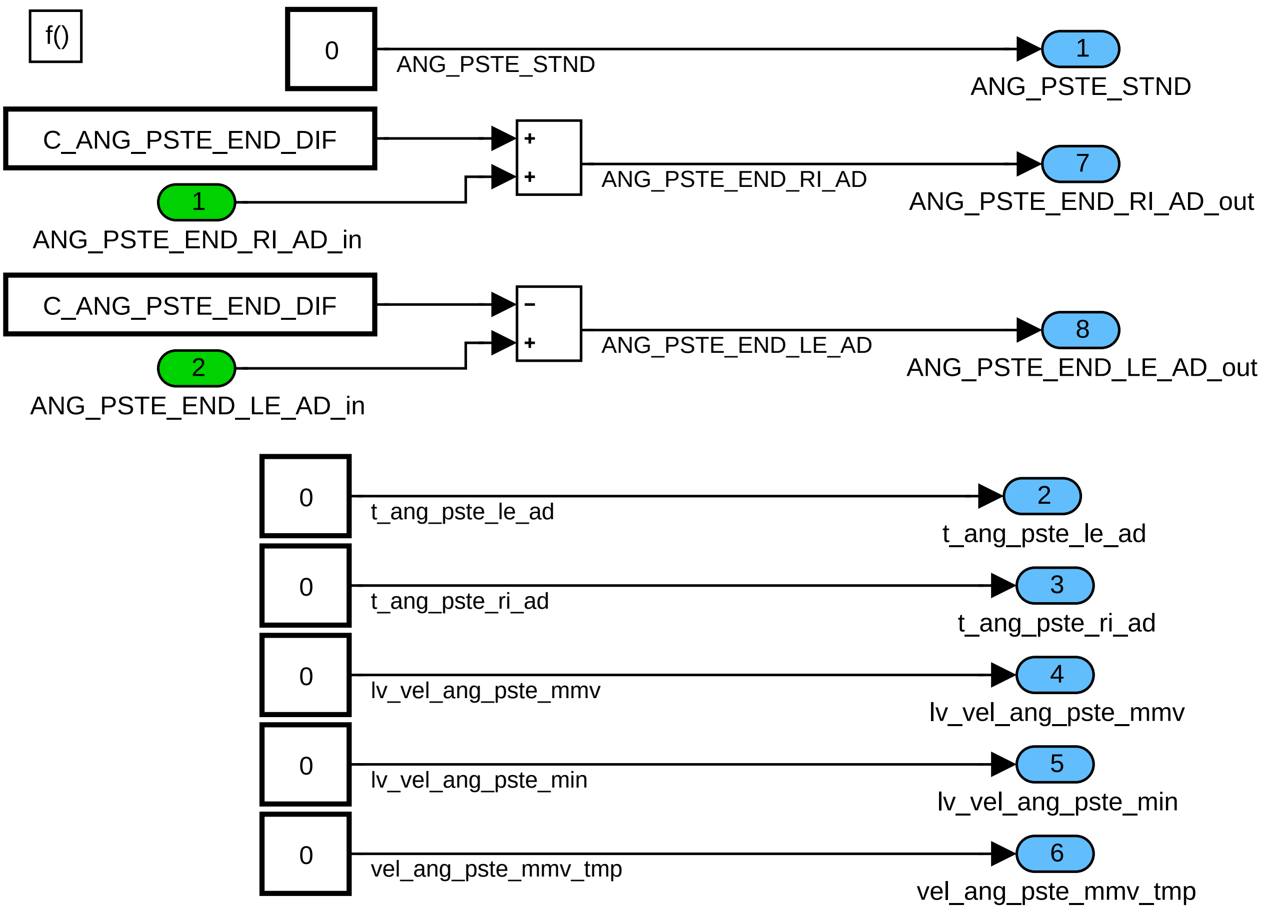


Figure 3:
Path: STSY_M4015/INI/INI

58.2.1.2 Initialisation at NVMY

In case of new ECU or if reading non-volatile memory not possible

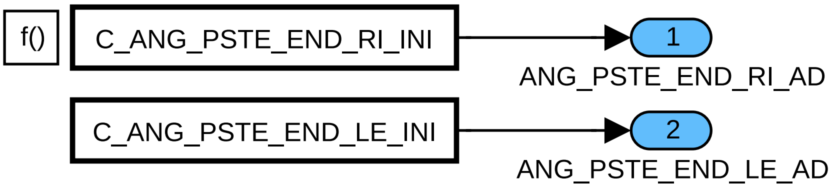


Figure 4:
Path: STSY_M4015/INI/NVM_INI

58.2.1.3 Restore of NVMY variable

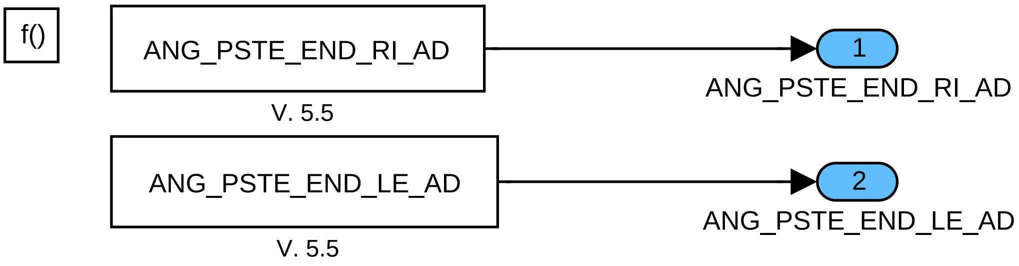


Figure 5:
Path: STSY_M4015/INI/NVM_RES

58.2.1.4 Storing of NVMY variable

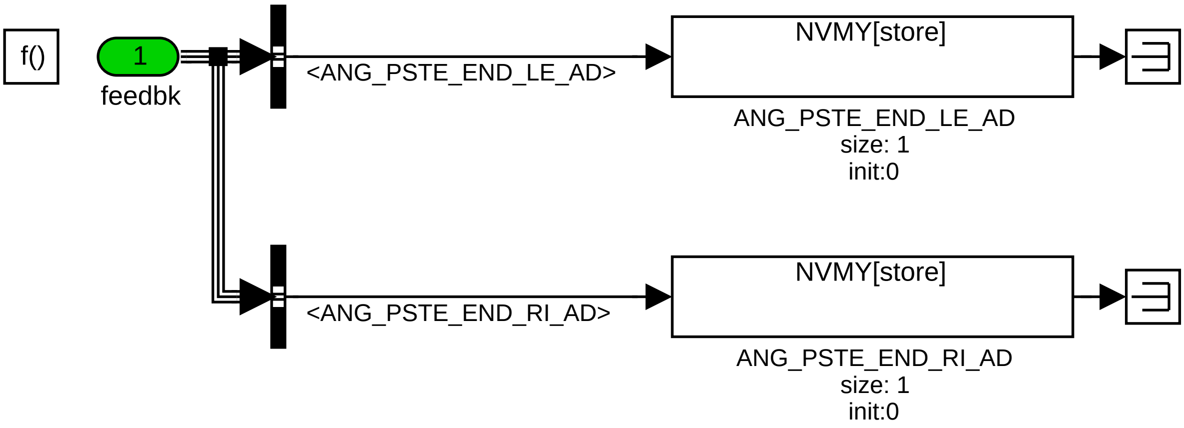


Figure 6:
Path: STSY_M4015/INI/NVM_STO

58.2.2 Function overview

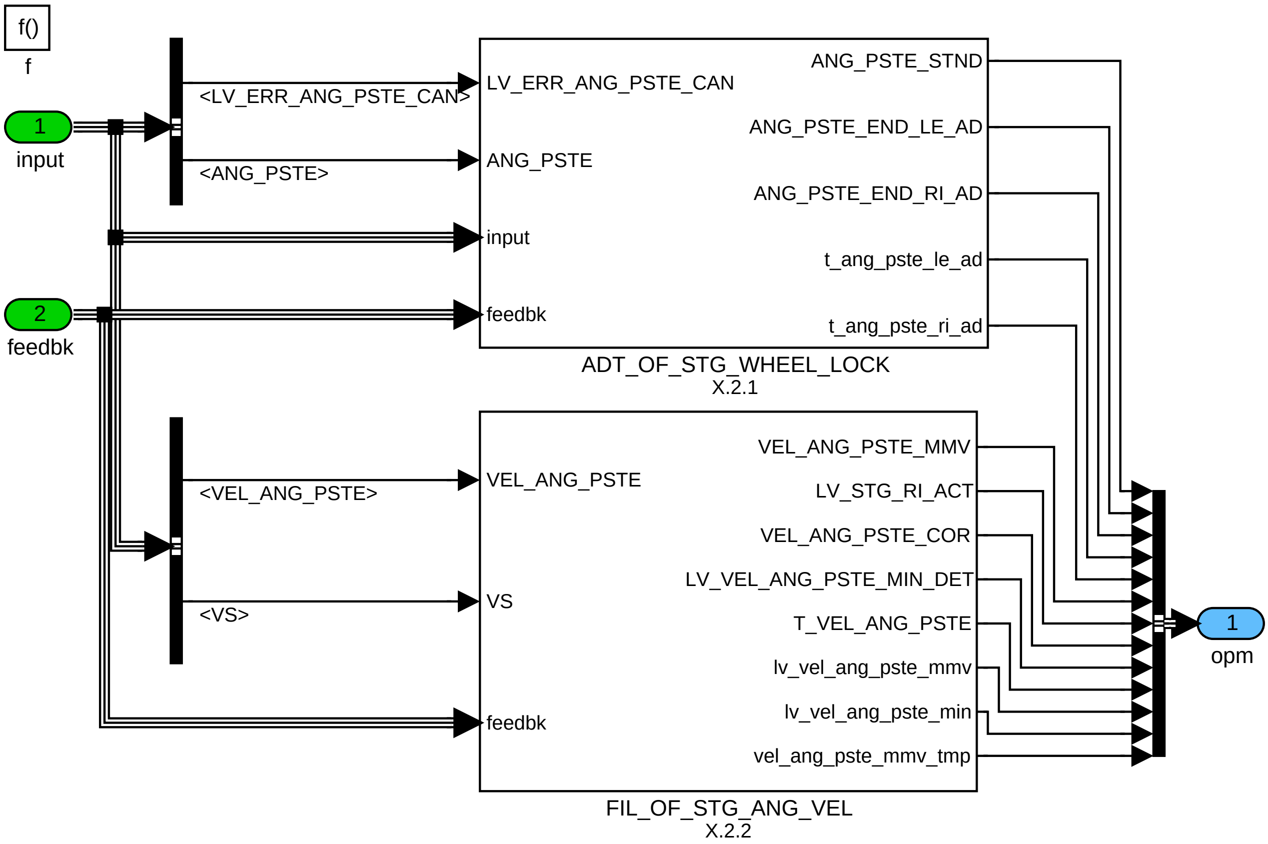


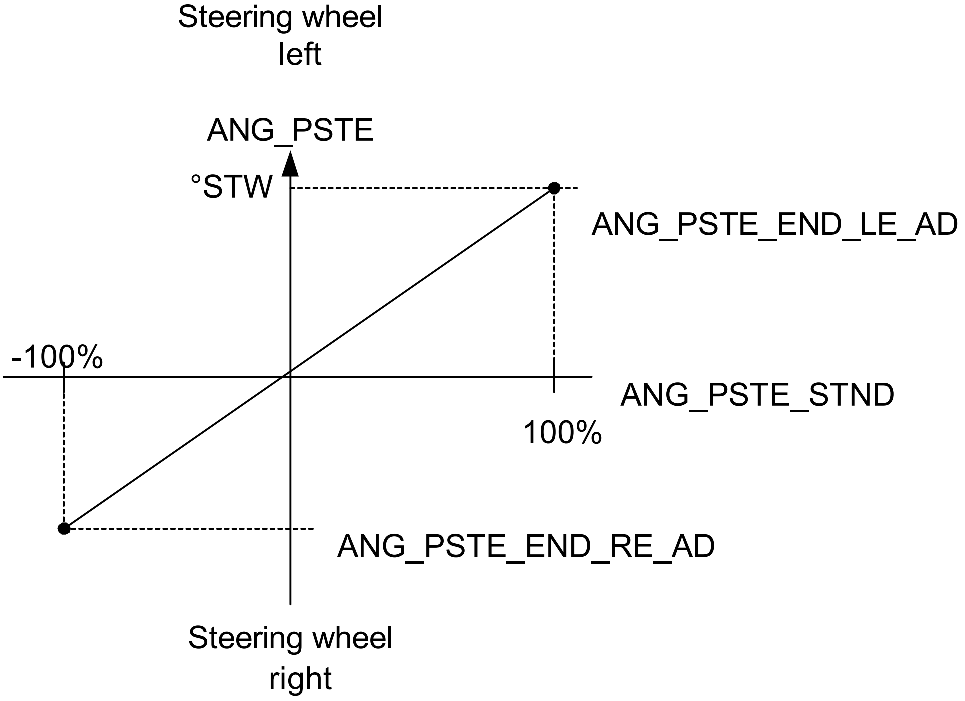
Figure 7:
Path: STSY_M4015/OPM

58.2.2.1 Adaptation of steering wheel lock

General information:
This module contains the calculation of the relative steering wheel sensor (ANG_PSTE_STND) and relating adaptations (ANG_PSTE_END_RI_AD, ANG_PSTE_END_LE_AD) depending on the steering wheel angle ANG_PSTE received via CAN.

general specification

To enable a new adaptation of the steering wheel angles after every reset, the values ANG_PSTE_END_LE/_RI_AD should be initialised as follows:
the adaptation values restored out of the non-volatile memory minus a calibratable constant.
Signal flow diagram:



Application hint:

to ensure a proper adaptation of ANG_PSTE_END_LE/RI please note that the range of the adaptation incrementation should be calibrated within the following range
C_ANG_PSTE_AD_LGRD = 0.1 °STW ... 1.5 °STW

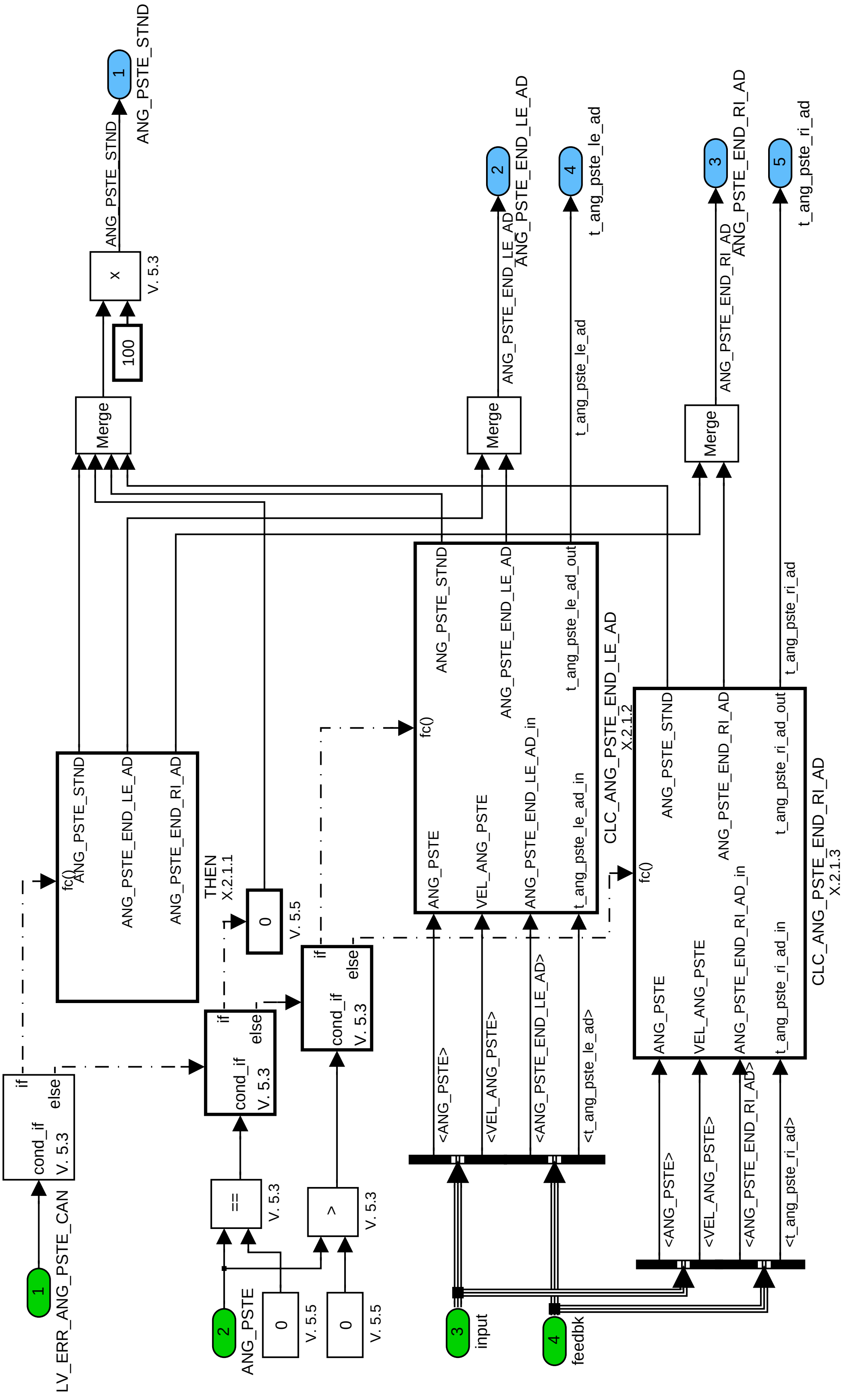


Figure 8:
Path: STSY_M4015/OPM/ADT_OF_STG_WHEEL_LOCK

58.2.2.1.1 Initialisation for variables

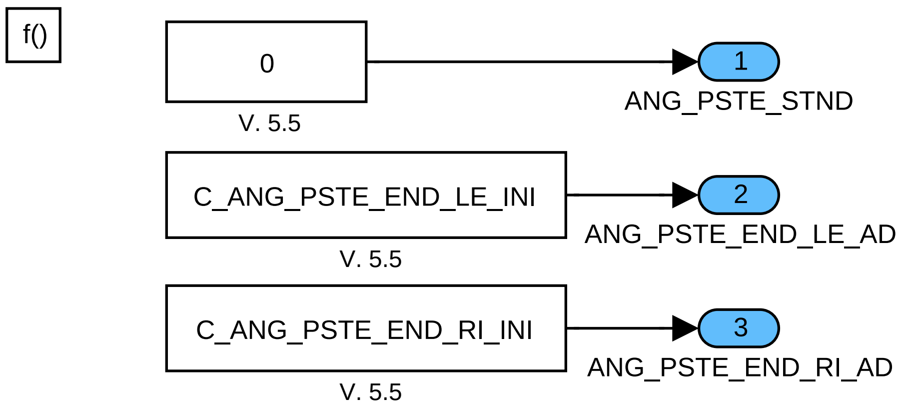


Figure 9:
Path: STSY_M4015/OPM/ADT_OF_STG_WHEEL_LOCK/THEN

58.2.2.1.2 Condition for calcualtion for the ANG_PSTE_END_LE_AD

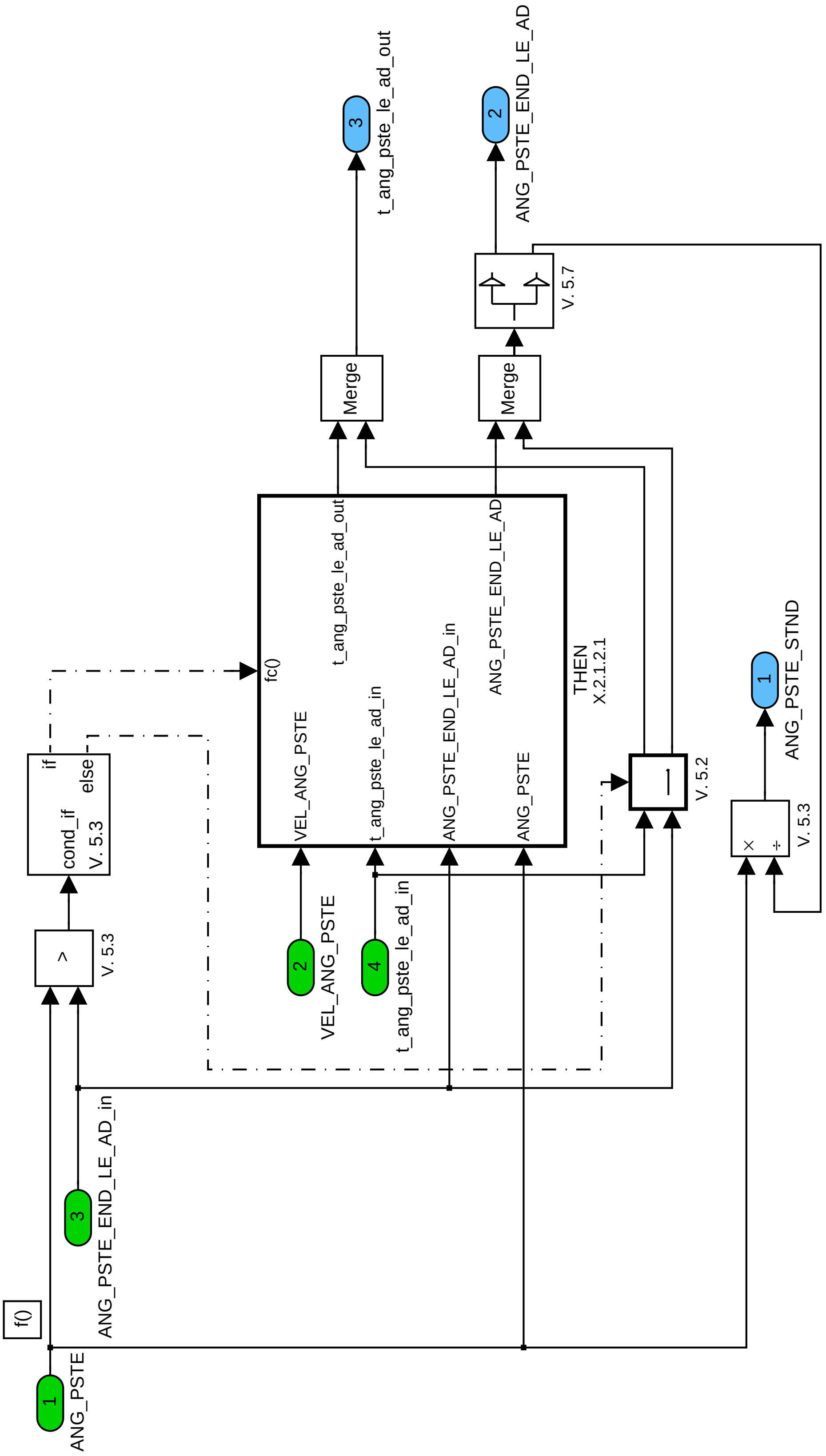


Figure 10:
Path: STSY_M4015/OPM/ADT_OF_STG_WHEEL_LOCK/CLC_ANG_PSTE_END_LE_AD

58.2.2.1.2.1 Caluculation of ANG_PSTE_END_LE_AD

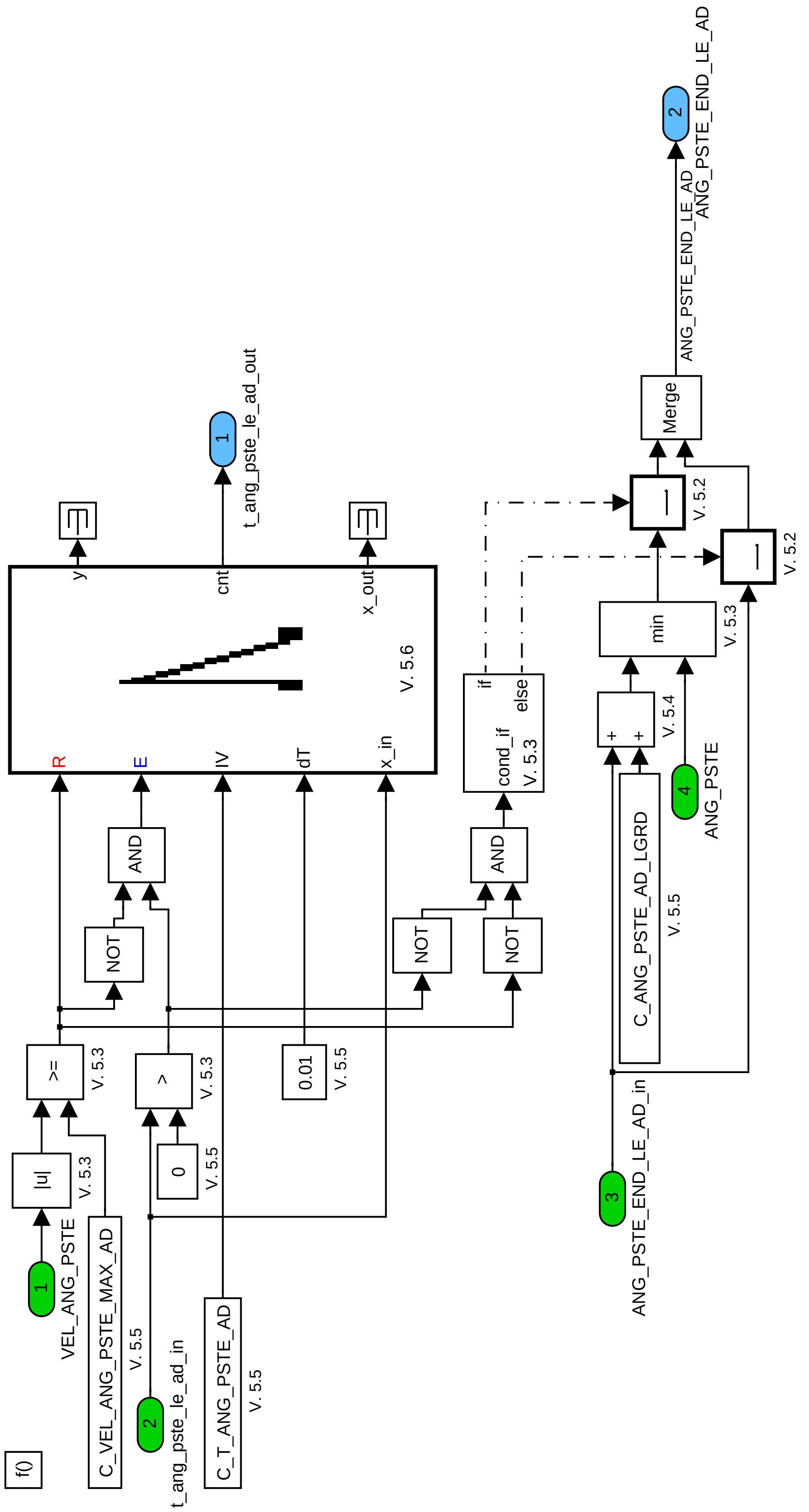


Figure 11:
Path: STSY_M4015/OPM/ADT_OF_STG_WHEEL_LOCK/CLC_ANG_PSTE_END_LE_AD/THEN

58.2.2.1.3 Condition for calcualtion for the ANG_PSTE_END_RI_AD

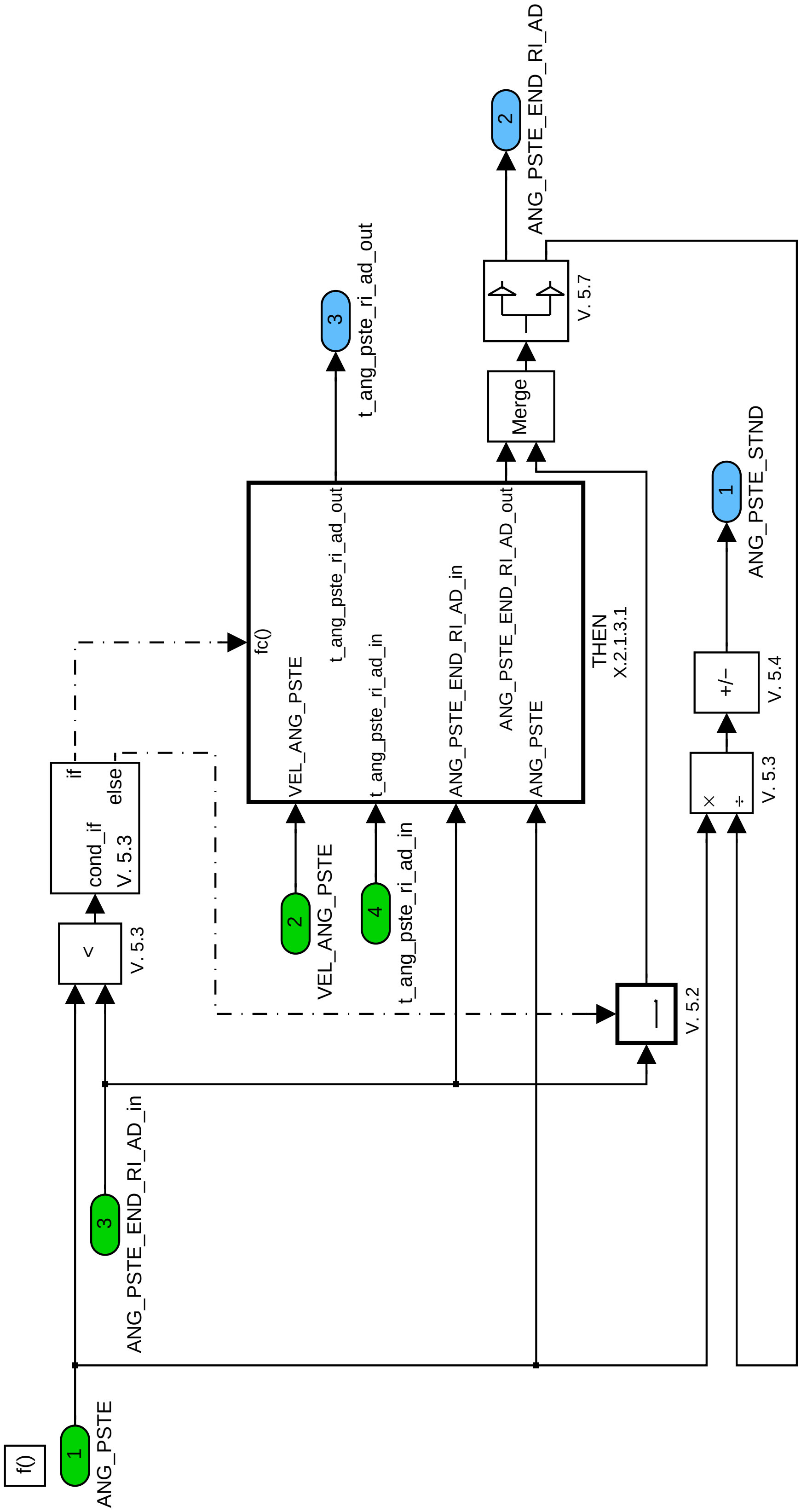


Figure 12:
Path: STSY_M4015/OPM/ADT_OF_STG_WHEEL_LOCK/CLC_ANG_PSTE_END_RI_AD

58.2.2.1.3.1 Caluculation of ANG_PSTE_END_LE_AD

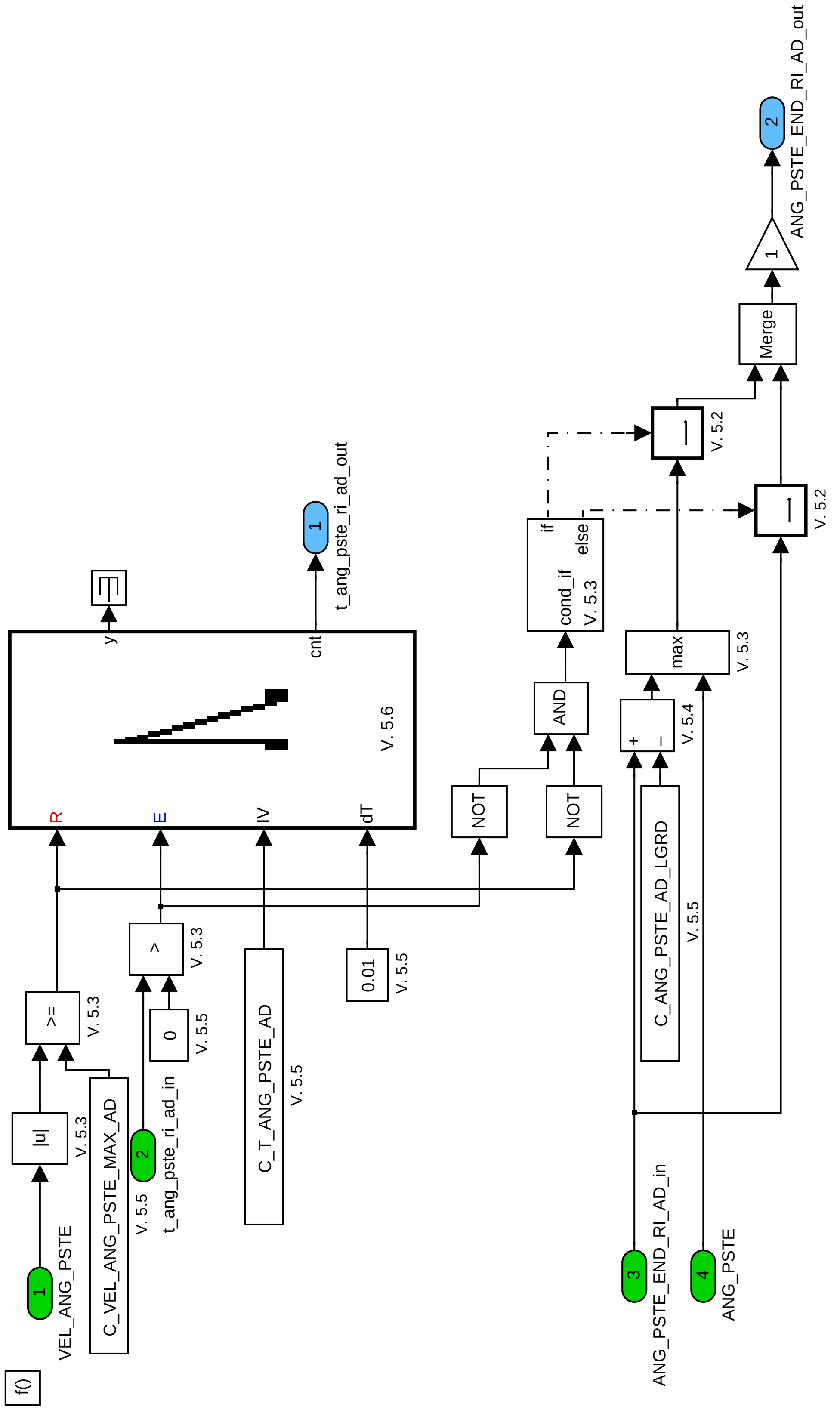


Figure 13:
Path: STSY_M4015/OPM/ADT_OF_STG_WHEEL_LOCK/CLC_ANG_PSTE_END_RI_AD/THEN

58.2.2.2 Filtering of steering wheel angle velocity

General information:
This module contains the filtering of the steering-wheel angle velocity as well as the detection of the steering direction change and the state of the steering wheel (steering wheel: released or held tight). It is used as an input for the torque loss for power-steering-calculation: if the steering-wheel is held tight the providing of a torque loss for power-steering is necessary; whereas if the steering-wheel is released only the torque (Schleppmoment) of the power-steering pump itself is sufficient.

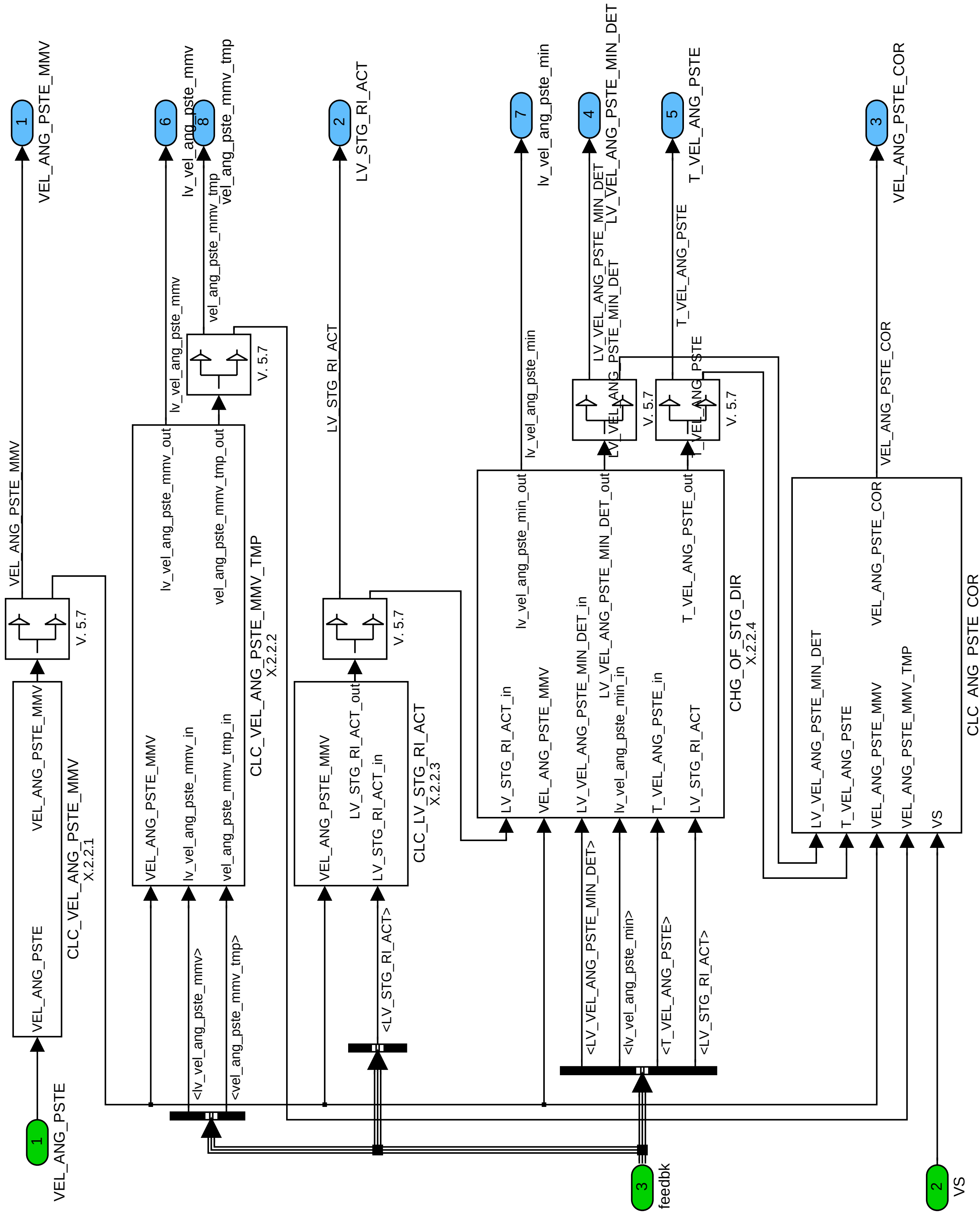


Figure 14:
Path: STSY_M4015/OPM/FIL_OF_STG_ANG_VEL

58.2.2.2.1 Moving mean value VEL_ANG_PSTE_MMV:

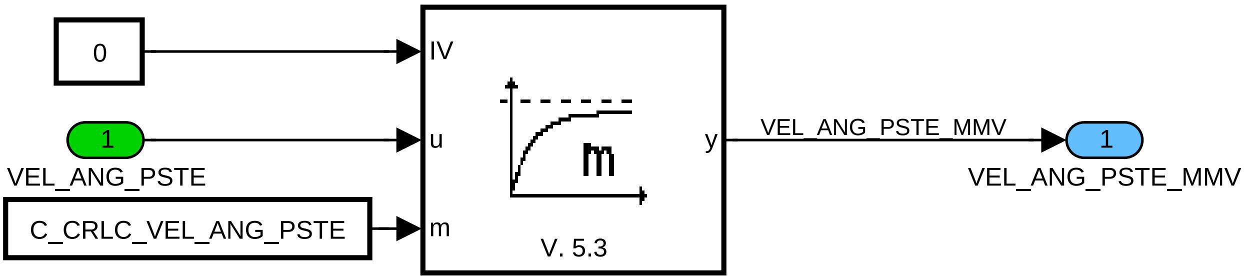


Figure 15:
Path: STSY_M4015/OPM/FIL_OF_STG_ANG_VEL/CLC_VEL_ANG_PSTE_MMV

58.2.2.2.2 Temporary moving mean value VEL_ANG_PSTE_MMV_TMP:

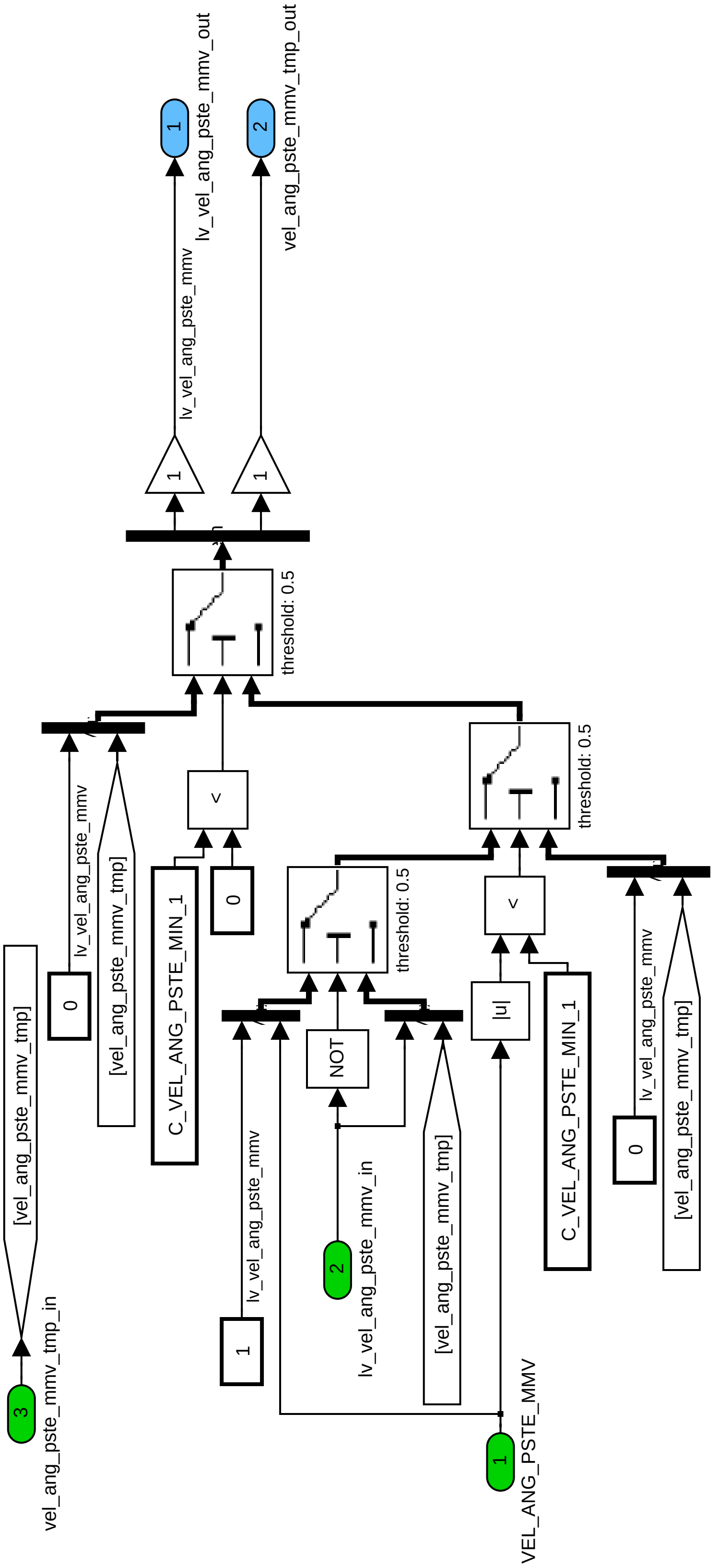


Figure 16:
Path: STSY_M4015/OPM/FIL_OF_STG_ANG_VEL/CLC_VEL_ANG_PSTE_MMV_TMP

58.2.2.2.3 Recognition of steering-direction:

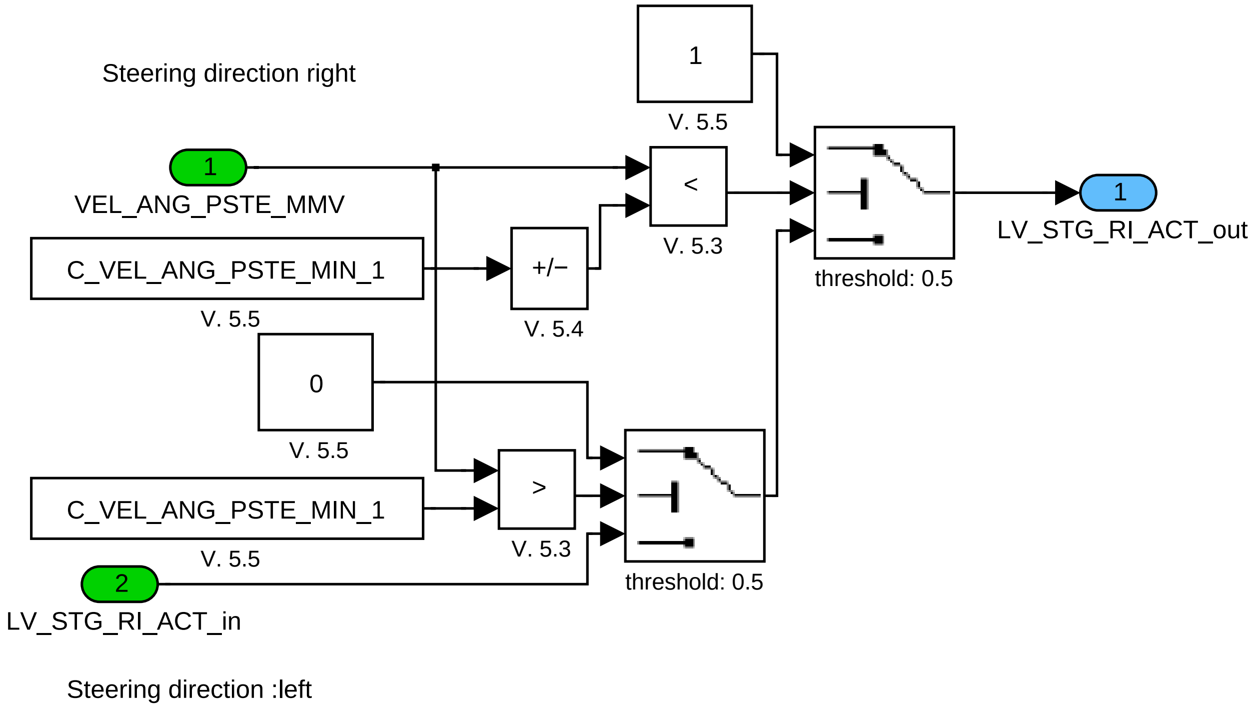


Figure 17:
Path: STSY_M4015/OPM/FIL_OF_STG_ANG_VEL/CLC_LV_STG_RI_ACT

58.2.2.2.4 Change of steering direction:

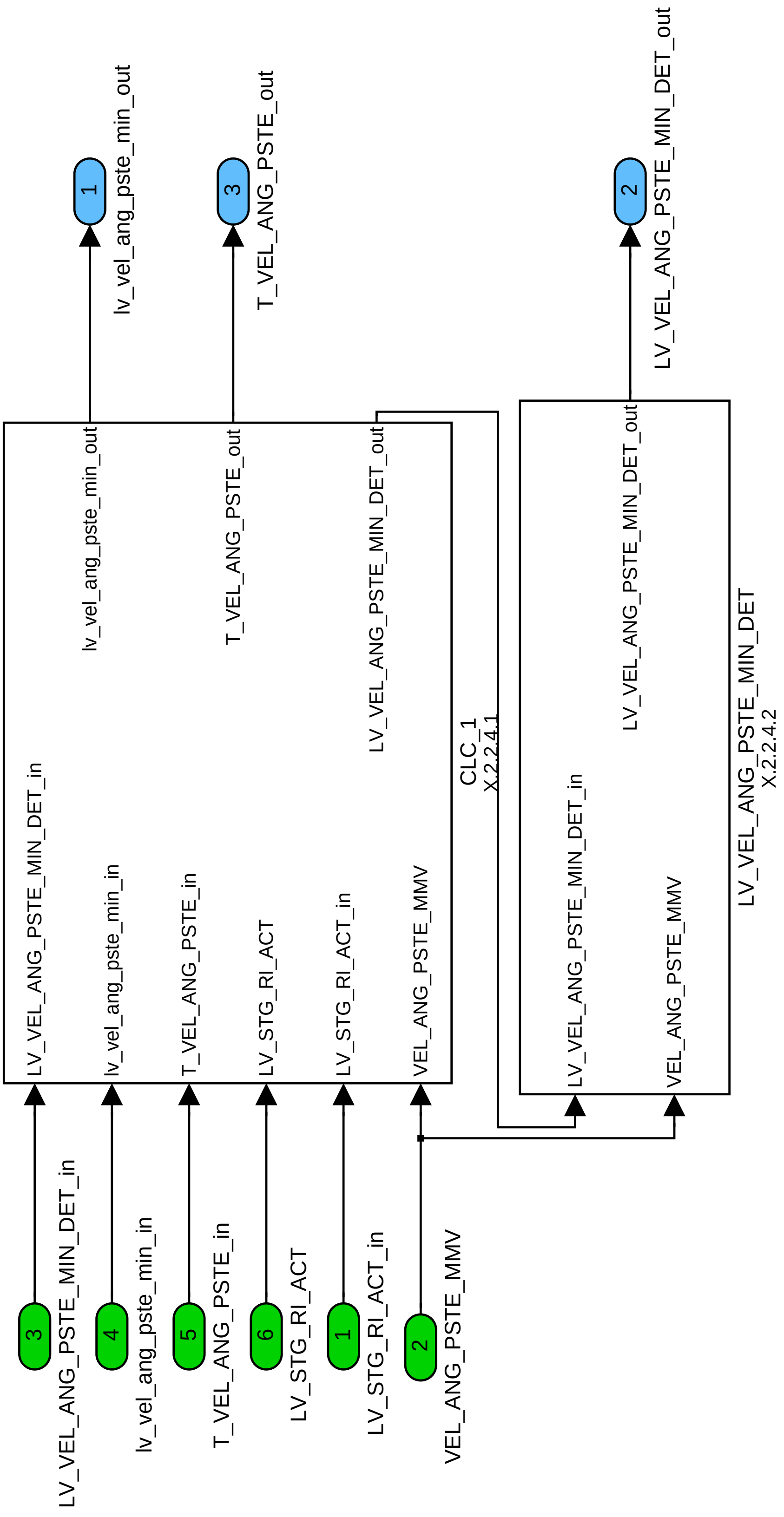


Figure 18:
Path: STSY_M4015/OPM/FIL_OF_STG_ANG_VEL/CHG_OF_STG_DIR

58.2.2.2.4.1 Calculation of T_VEL_ANG_PSTE

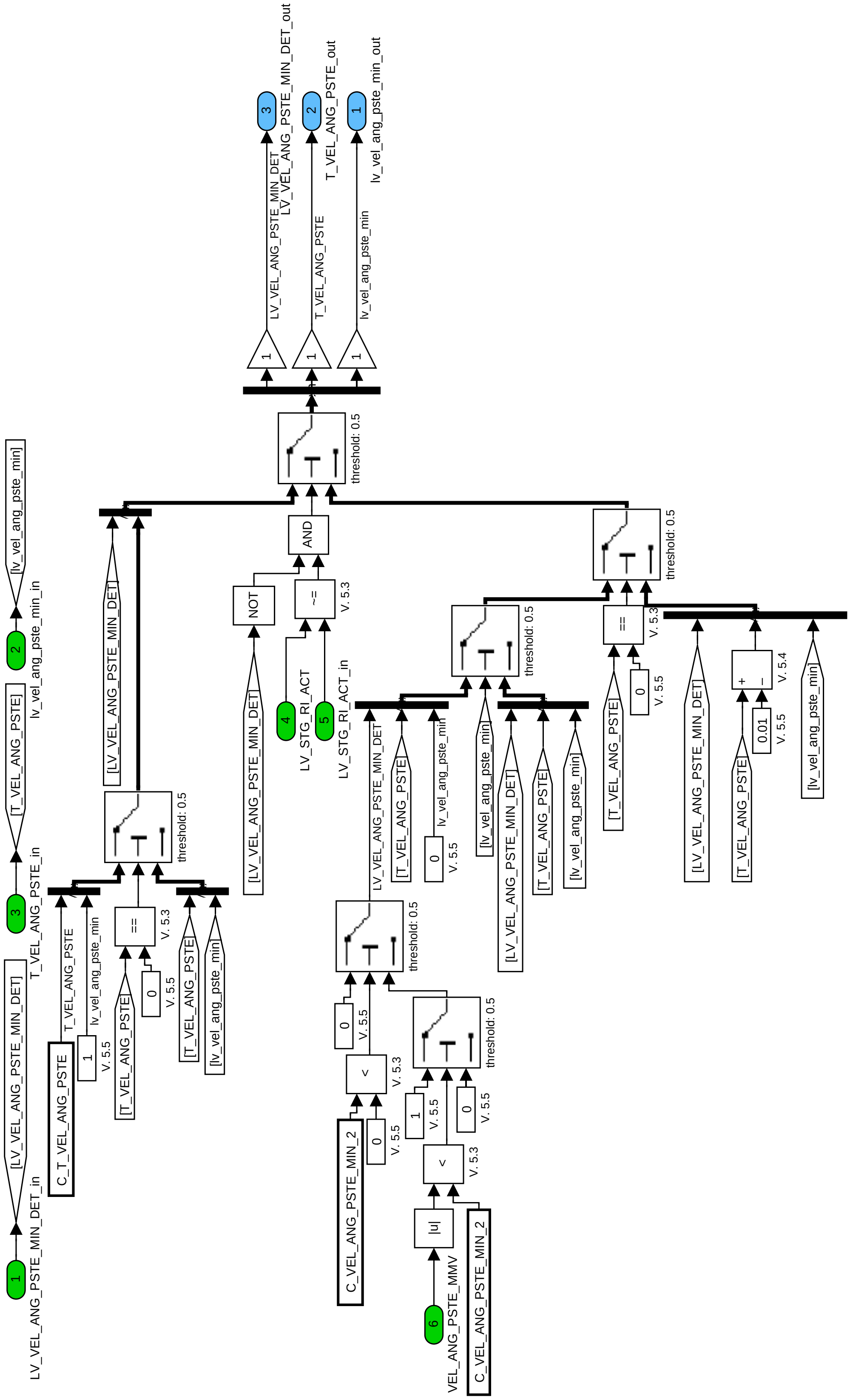


Figure 19:
Path: STSY_M4015/OPM/FIL_OF_STG_ANG_VEL/CHG_OF_STG_DIR/CLC_1

58.2.2.2.4.2 Calculation of LV_VEL_ANG_PSTE_MIN_DET

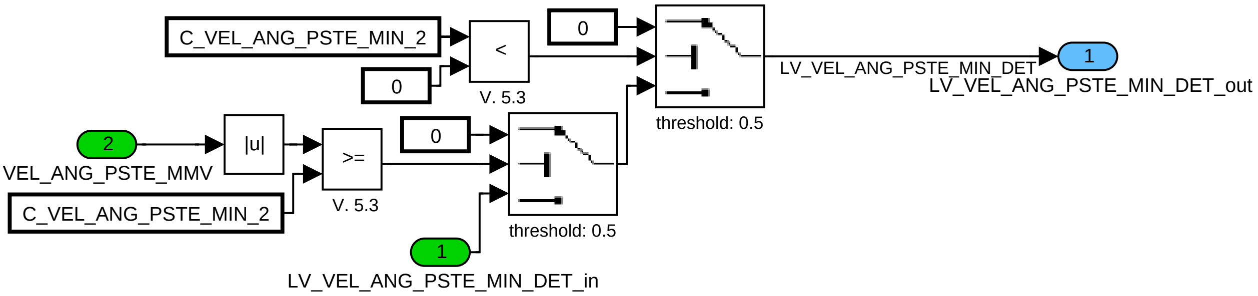


Figure 20:
Path: STSY_M4015/OPM/FIL_OF_STG_ANG_VEL/CHG_OF_STG_DIR/LV_VEL_ANG_PSTE_MIN_DET

58.2.2.2.5 Filtered steering wheel angle velocity VEL_ANG_PSTE_COR:

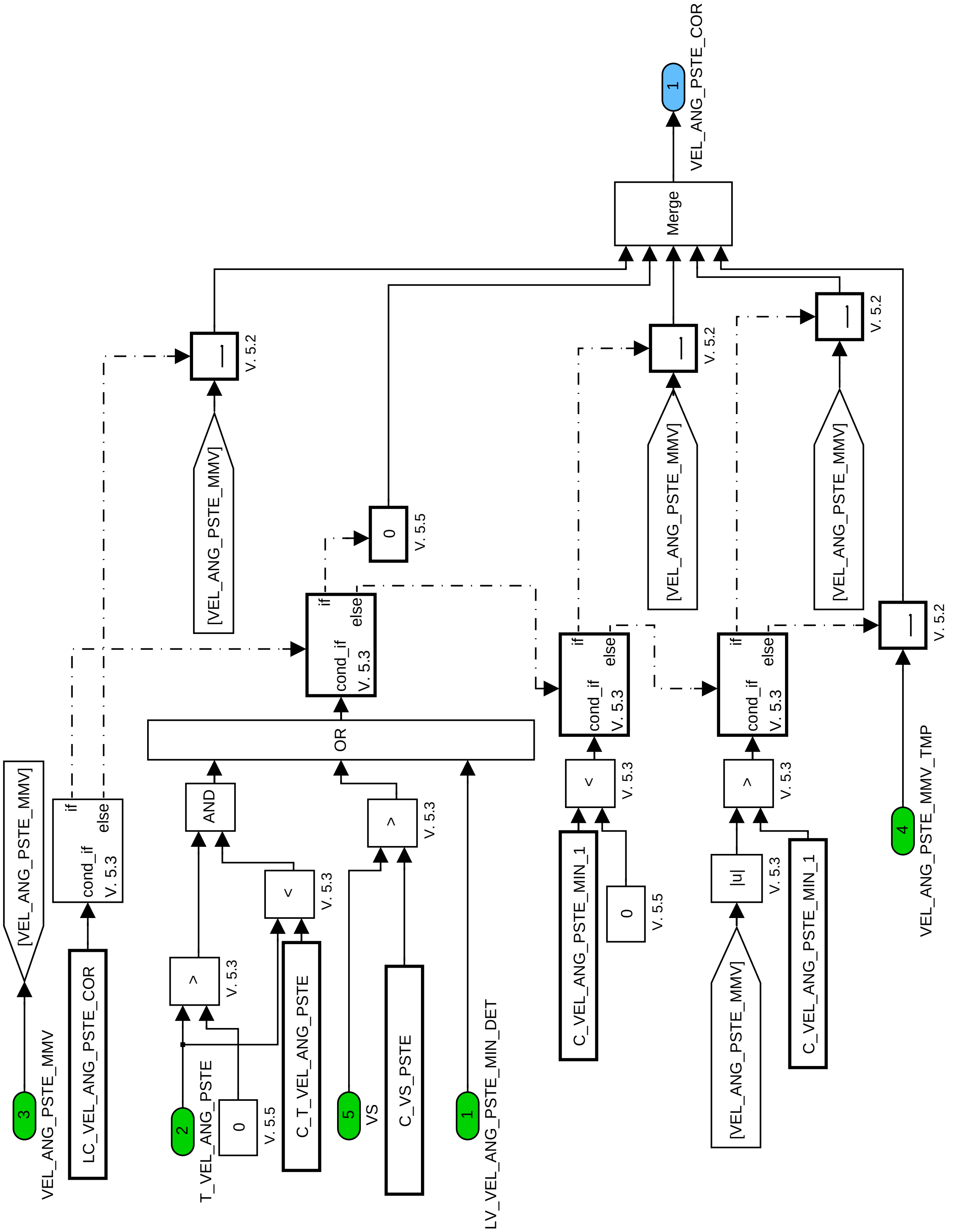


Figure 21:
Path: STSY_M4015/OPM/FIL_OF_STG_ANG_VEL/CLC_ANG_PSTE_COR

58.3 Torque Loss for power steering

Overview

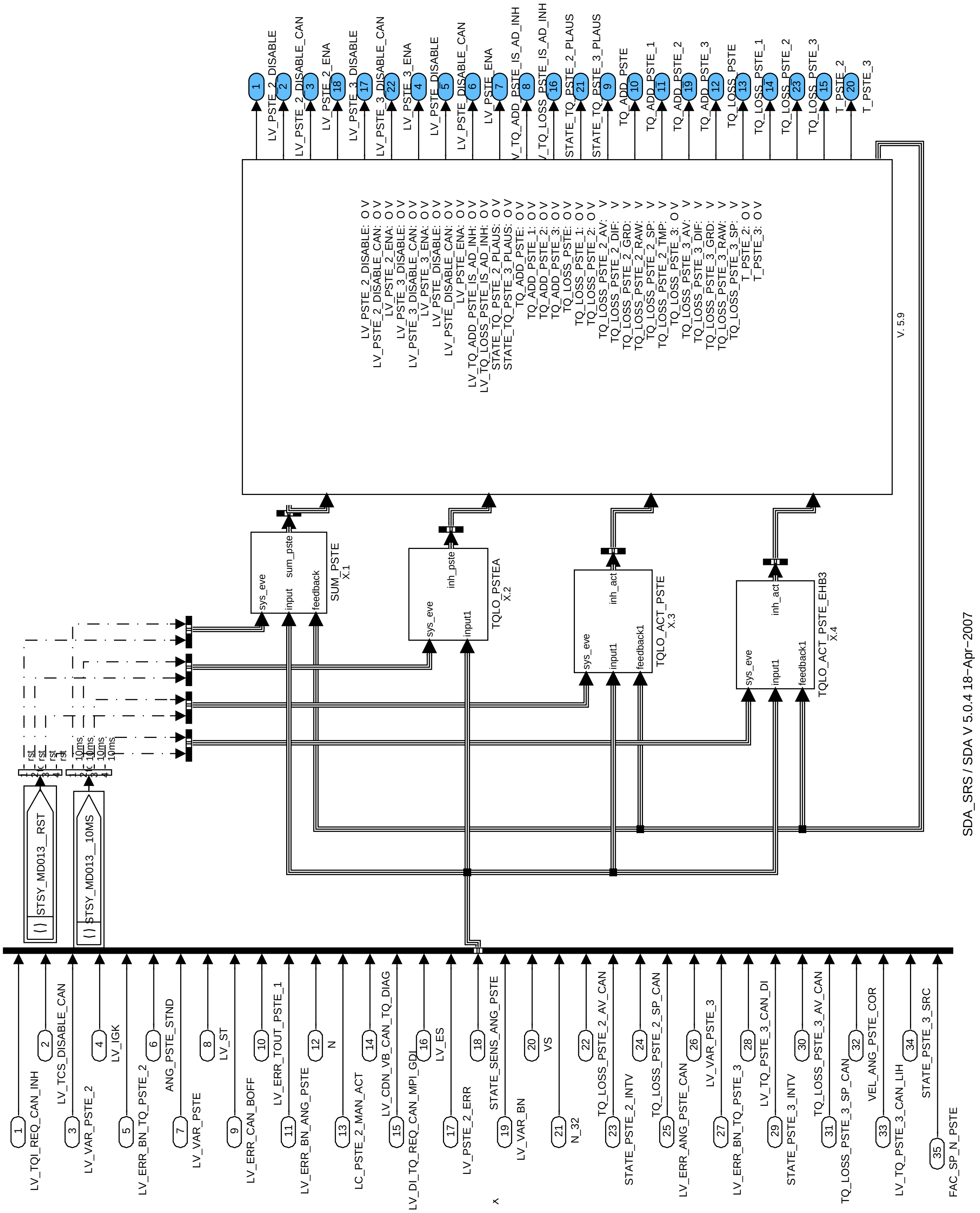


Figure 22:
Path: STSY_MD013

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_PSTE_2_DISABLE	O/V	0... 1H	0... 1	1	[-]
LV for disabling PSTE intervention due to EMS error					
LV_PSTE_2_DISABLE_CAN	O/V	0... 1H	0... 1	1	[-]
LV for disabling PSTE intervention due to CAN error					
LV_PSTE_2_ENA	O/V	0... 1H	0... 1	1	[-]
LV for enabling PSTE intervention					
LV_PSTE_3_DISABLE	O/V	0... 1H	0... 1	1	[-]
LV for disabling PSTE intervention due to EMS error					
LV_PSTE_3_DISABLE_CAN	O/V	0... 1H	0... 1	1	[-]
LV for disabling PSTE 3 intervention due to CAN error					
LV_PSTE_3_ENA	O/V	0... 1H	0... 1	1	[-]
LV for enabling PSTE intervention					
LV_PSTE_DISABLE	O/V	0... 1H	0... 1	1	[-]
LV for disabling PSTE intervention due to EMS error					
LV_PSTE_DISABLE_CAN	O/V	0... 1H	0... 1	1	[-]
LV for disabling PSTE intervention due to CAN error					
LV_PSTE_ENA	O/V	0... 1H	0... 1	1	[-]
LV for enabling PSTE intervention					
LV_TQ_ADD_PSTE_IS_AD_INH	O/V	0... 1H	0... 1	1	[-]
Flag for inhibition of IS-control adaptations due to torque reserve PSTE					
LV_TQ_LOSS_PSTE_IS_AD_INH	O/V	0... 1H	0... 1	1	[-]
Flag for inhibition of IS-control adaptations due to torque losses PSTE					
STATE_TQ_PSTE_2_PLAUS	O/V	0... FFH	0... 255	1	[-]
Bitwise coded State for PSTE_2 intervention state					
STATE_TQ_PSTE_3_PLAUS	O/V	0... FFH	0... 255	1	[-]
Bitwise coded State for PSTE_3 intervention state					
T_PSTE_2	O/V	0... FFH	0... 2.55	0.01	[s]
SP AV Timer PSTE_2					
T_PSTE_3	O/V	0... FFH	0... 2.55	0.01	[s]
SP AV Timer PSTE_3					
TQ_ADD_PSTE	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque reserve for power steering					
TQ_ADD_PSTE_1	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque reserve for power steering					
TQ_ADD_PSTE_2	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm/10 ms]
Engine torque reserve active front steering					
TQ_ADD_PSTE_3	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque reserve for eletric power steering (EHB3)					
TQ_LOSS_PSTE	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Summary Engine torque losses power steering					
TQ_LOSS_PSTE_1	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses power steering					
TQ_LOSS_PSTE_2	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses active front steering					
TQ_LOSS_PSTE_2_AV	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses active front steering active value					

TQ_LOSS_PSTE_2_DIF	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses active front steering difference					
TQ_LOSS_PSTE_2_GRD	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm/10ms]
Engine torque losses active front steering gradient					
TQ_LOSS_PSTE_2_RAW	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm/10ms]
Engine torque losses active front steering					
TQ_LOSS_PSTE_2_SP	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses active front steering set point					
TQ_LOSS_PSTE_2_TMP	V	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Intermidiate LABEL used as input 2nd axis of IP_TQ_ADD_PSTE_2_TMP					
TQ_LOSS_PSTE_3	O/V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses for electric power steering (EHB3)					
TQ_LOSS_PSTE_3_AV	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses actual value for electric power steering (EHB3)					
TQ_LOSS_PSTE_3_DIF	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses difference for electric power steering (EHB3)					
TQ_LOSS_PSTE_3_GRD	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses gradient for electric power steering (EHB3)					
TQ_LOSS_PSTE_3_RAW	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses for electric power steering (EHB3)					
TQ_LOSS_PSTE_3_SP	V	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Engine torque losses electric power steering setpoint					

Input Data:

LV_TQI_REQ_CAN_INH	LV_TCS_DISABLE_CAN	LV_VAR_PSTE_2	LV_IGK
LV_ERR_BN_TQ_PSTE_2	ANG_PSTE_STND	LV_VAR_PSTE	LV_ST
LV_ERR_CAN_BOFF	LV_ERR_TOUT_PSTE_1	LV_ERR_BN_ANG_PSTE	N
LC_PSTE_2_MAN_ACT	LV_CDN_VB_CAN_TQ_DIAG	LV_DI_TQ_REQ_CAN_MPI_GDI	LV_ES
LV_PSTE_2_ERR	STATE_SENS_ANG_PSTE	LV_VAR_BN	VS
N_32	TQ_LOSS_PSTE_2_AV_CAN	STATE_PSTE_2_INTV	TQ_LOSS_PSTE_2_SP_CAN
LV_ERR_ANG_PSTE_CAN	LV_VAR_PSTE_3	LV_ERR_BN_TQ_PSTE_3	LV_TQ_PSTE_3_CAN_DI
STATE_PSTE_3_INTV	TQ_LOSS_PSTE_3_AV_CAN	TQ_LOSS_PSTE_3_SP_CAN	VEL_ANG_PSTE_COR
LV_TQ_PSTE_3_CAN_LIH	STATE_PSTE_3_SRC	FAC_SP_N_PSTE	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_FAC_TQ_LOSS_PSTE_2_DIF	1	0... FFH	-2... 1.984375	0.015625	[-]
Correction value differences					
C_FAC_TQ_LOSS_PSTE_2_GRD	1	0... FFH	-2... 1.984375	0.015625	[-]
Correction value gradient					
C_FAC_TQ_LOSS_PSTE_3_DIF	1	0... FFH	-2... 1.984375	0.015625	[-]
Correction value differences					

C_FAC_TQ_LOSS_PSTE_3_GRD	1	0... FFH	-2... 1.984375	0.015625	[-]
Correction value gradient					
C_T_PSTE_DLY_AD	1	0... FFH	0... 2.55	0.01	[s]
Threshold torque losses PSTE for inhibition of IS-control adaptations					
C_TQ_ADD_PSTE_1_LIH	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limp home value torque reserve					
C_TQ_ADD_PSTE_AD	1	0... 7FFFH	0... 1023.97	0.03125	[Nm]
threshold torque reserve PSTE for inhibition of IS-control adaptations					
C_TQ_LOSS_PSTE_1_LIH	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Limp home value torque losses					
C_TQ_LOSS_PSTE_2_LIH	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Limp home value torque losses					
C_TQ_LOSS_PSTE_2_SP_MIN	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Min torque threshold for timer reset					
C_TQ_LOSS_PSTE_3_LIH	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Limp home value torque losses					
C_TQ_LOSS_PSTE_3_SP_MIN	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
Min torque threshold for timer reset					
C_TQ_LOSS_PSTE_AD	1	8000... 7FFFH	-1024... 1023.96875	0.03125	[Nm]
threshold torque losses PSTE for inhibition of IS-control adaptations					
IP_FAC_TQ_ADD_PSTE_2_VS	6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_2	6	0... FFH	0... 255	1	[km/h]
Active power steering torque loss correction					
IP_FAC_TQ_ADD_PSTE_3_VS_PSTE	6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
electric power steering torque loss correction					
IP_FAC_TQ_ADD_PSTE_3_VS_PSTE_2	6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
electric power steering torque loss correction					
IP_FAC_TQ_LOSS_PSTE	8	0... FFH	0... 3.984375	0.015625	[-]
LDP_N__FAC_TQ_LOSS_PSTE	8	0... 1FE0H	0... 8160	1	[rpm]
factor for weighting the torque request in dependance of speed (P=M*()					
IP_FAC_TQ_LOSS_PSTE_2	8*6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_2	6	0... FFH	0... 255	1	[km/h]
LDPM_N_32__PSTE_2	8	0... FFH	0... 8160	32	[rpm]
Factor active power steering torque loss correction vs					
IP_FAC_TQ_LOSS_PSTE_2_ANG	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDPM_VEL_ANG_PSTE_TQLO	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
factor active power steering torque loss correction °STW					
IP_FAC_TQ_LOSS_PSTE_2_AV	4*4	0... FFH	-2... 1.984375	0.015625	[-]
LDPM_N_32__PSTE_2_1	4	0... FFH	0... 8160	32	[rpm]
LDPM_T_PSTE_2__PSTE_2	4	0... FFH	0... 2.55	0.01	[s]
Active power steering torque loss correction AV					
IP_FAC_TQ_LOSS_PSTE_2_SP	4*4	0... FFH	-2... 1.984375	0.015625	[-]
LDPM_N_32__PSTE_2_1	4	0... FFH	0... 8160	32	[rpm]
LDPM_T_PSTE_2__PSTE_2	4	0... FFH	0... 2.55	0.01	[s]
Active power steering torque loss correction SP					
IP_FAC_TQ_LOSS_PSTE_2_VS	8*6	0... FFH	0... 3.984375	0.015625	[-]
LDPM_VS_TQ_LOSS_PSTE_2	6	0... FFH	0... 255	1	[km/h]

LDPM_N_32__PSTE_2	8	0... FFH	0... 8160	32	[rpm]
Factor active power steering torque loss correction vs					
IP_FAC_TQ_LOSS_PSTE_3_AV	4*4	0... FFH	-2... 1.984375	0.015625	[-]
LDPM_N_32__PSTE_3_1	4	0... FFH	0... 8160	32	[rpm]
LDPM_T_PSTE_3__PSTE_3	4	0... FFH	0... 2.55	0.01	[s]
Electric power steering torque loss correction AV					
IP_FAC_TQ_LOSS_PSTE_3_COR_PSTE	8*6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
LDPM_FAC_SP_N_PSTE_1_STSY	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
Factor electric power steering (EHB 3) torque loss correction vs					
IP_FAC_TQ_LOSS_PSTE_3_COR_PSTE2	8*6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
LDPM_FAC_SP_N_PSTE_1_STSY	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
Factor electric power steering (EHB 3) torque loss correction vs					
IP_FAC_TQ_LOSS_PSTE_3_MDL_PSTE	8*6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
LDPM_FAC_SP_N_PSTE_1_STSY	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
Factor electric power steering (EHB 3) torque loss correction vs					
IP_FAC_TQ_LOSS_PSTE_3_MDL_PSTE2	8*6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
LDPM_FAC_SP_N_PSTE_1_STSY	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
Factor electric power steering (EHB 3) torque loss correction vs					
IP_FAC_TQ_LOSS_PSTE_3_PSTE	8*6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
LDPM_FAC_SP_N_PSTE_1_STSY	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
Factor electric power steering (EHB 3) torque loss correction vs					
IP_FAC_TQ_LOSS_PSTE_3_PSTE_2	8*6	0... FFH	0... 3.984375	0.015625	[-]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
LDPM_FAC_SP_N_PSTE_1_STSY	8	0... FFH	0... 1.9921875	7.8125e-3	[-]
Factor electric power steering (EHB 3) torque loss correction vs					
IP_FAC_TQ_LOSS_PSTE_3_SP	4*4	0... FFH	-2... 1.984375	0.015625	[-]
LDPM_N_32__PSTE_3_1	4	0... FFH	0... 8160	32	[rpm]
LDPM_T_PSTE_3__PSTE_3	4	0... FFH	0... 2.55	0.01	[s]
Electric power steering torque loss correction SP					
IP_TQ_ADD_PSTE	8	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
LDPM_ANG_PSTE_STND_TQLO_1	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
Power steering torque reserve depending on engine-speed					
IP_TQ_ADD_PSTE_2	8*8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDP_TQ_LOSS_PSTE_2_SP_CAN	8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Active power steering torque reserve					
IP_TQ_ADD_PSTE_2_LIH	8*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_VS_TQ_LOSS_PSTE_2	6	0... FFH	0... 255	1	[km/h]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
Factor active power steering torque loss correction vs					
IP_TQ_ADD_PSTE_2_TMP	8*8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDP_TQ_LOSS_PSTE_2_TMP	8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
Active power steering torque reserve using different 2nd axis					

IP_TQ_ADD_PSTE_3_LIH	8*6	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_VS_TQ_LOSS_PSTE_3	6	0... FFH	0... 255	1	[km/h]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
Factor eletric power steering (EHB 3) torque loss correction vs					
IP_TQ_ADD_PSTE_3_PSTE	8*8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDP_TQ_LOSS_PSTE_3_SP_CAN	8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
electric power steering torque reserve					
IP_TQ_ADD_PSTE_3_PSTE_2	8*8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDP_TQ_LOSS_PSTE_3_SP_CAN	8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
electric power steering torque reserve					
IP_TQ_LOSS_PSTE	8*8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_ANG_PSTE_STND_TQLO_1	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDP_VEL_ANG_PSTE_TQLO_1	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
power steering torque loss versus engine-speed					
IP_TQ_LOSS_PSTE_2	8*8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDPM_VEL_ANG_PSTE_TQLO	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
Active power steering torque loss correction					
IP_TQ_LOSS_PSTE_3_ANG_PSTE	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDPM_VEL_ANG_PSTE_TQLO	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
electric power steering torque loss correction					
IP_TQ_LOSS_PSTE_3_ANG_PSTE_2	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDPM_VEL_ANG_PSTE_TQLO	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
electric power steering torque loss correction					
IP_TQ_LOSS_PSTE_3_COR_PSTE	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDPM_VEL_ANG_PSTE_TQLO	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
electric power steering torque loss correction					
IP_TQ_LOSS_PSTE_3_COR_PSTE_2	8*8	0... FFH	0... 3.984375	0.015625	[-]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDPM_VEL_ANG_PSTE_TQLO	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
electric power steering torque loss correction					
IP_TQ_LOSS_PSTE_3_MDL_PSTE	8*8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]

LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDPM_VEL_ANG_PSTE_TQLO	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
electric power steering torque loss correction					
IP_TQ_LOSS_PSTE_3_MDL_PSTE_2	8*8	0... FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_ANG_PSTE_STND_TQLO	8	0... FFFFH	-100... 99.9969482422	3.05176e-3	[%]
LDPM_VEL_ANG_PSTE_TQLO	8	0... FFFFH	-1439.98976... 1439.945815	0.043945	[°STW/s]
electric power steering torque loss correction					
LC_PSTE_2_ENA	1	0... 1H	0... 1	1	[-]
Logical variable for inhibitation of active power steering					
LC_PSTE_3_ENA	1	0... 1H	0... 1	1	[-]
Logical variable for inhibitation of eletric power steering EHB 3					
LC_PSTE_ENA	1	0... 1H	0... 1	1	[-]
Logical variable for enabling of power steering					
LC_TQ_ADD_PSTE_2_SWI	1	0... 1H	0... 1	1	[-]
Switch to change 2nd axis of IP_TQ_ADD_PSTE_2					
LC_TQ_LOSS_PSTE_2_SWI	1	0... 1H	0... 1	1	[-]
Configuration AFS switch					
LC_TQ_LOSS_PSTE_3_SWI	1	0... 1H	0... 1	1	[-]
Configuration EHB3 switch					

General Information

58.3.1 Summary of power steering losses

FUNCTION DESCRIPTION:

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: LV_IGK && (LV_VAR_PSTE_2 || LV_VAR_PSTE || LV_VAR_PSTE_3)
Deactivation: never

Function description

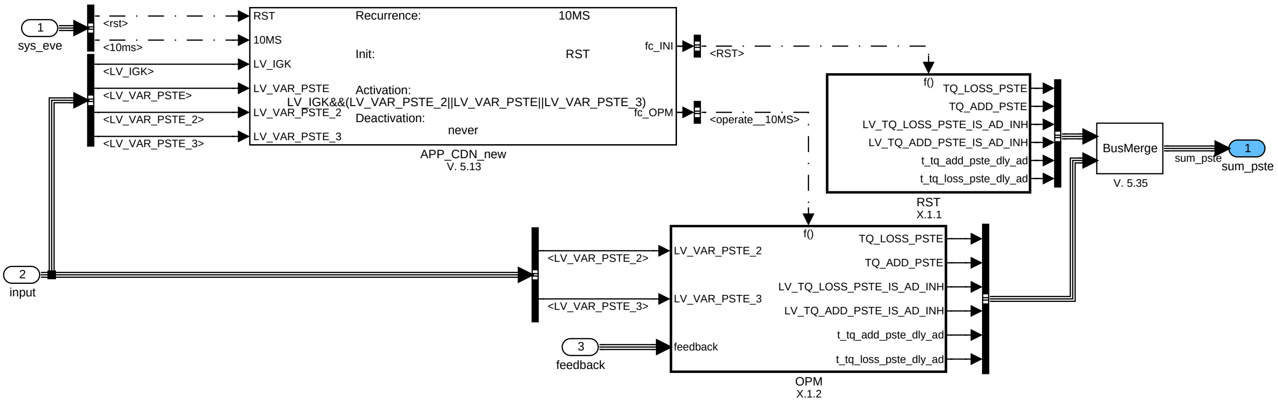


Figure 23:
Path: STSY_MD013/SUM_PSTE

58.3.1.1 Reset of variables is done here.

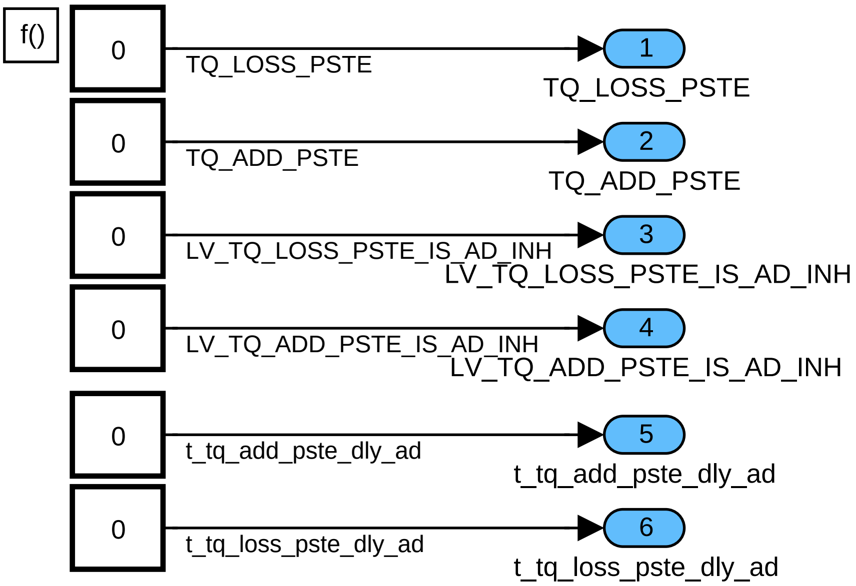


Figure 24:
Path: STSY_MD013/SUM_PSTE/RST

58.3.1.2 Formula section:

"TQ_LOSS_PSTE_1, TQ_LOSS_PSTE_2 and TQ_LOSS_PSTE_3 have to be calculate before TQ_LOSS_PSTE (using current values)."

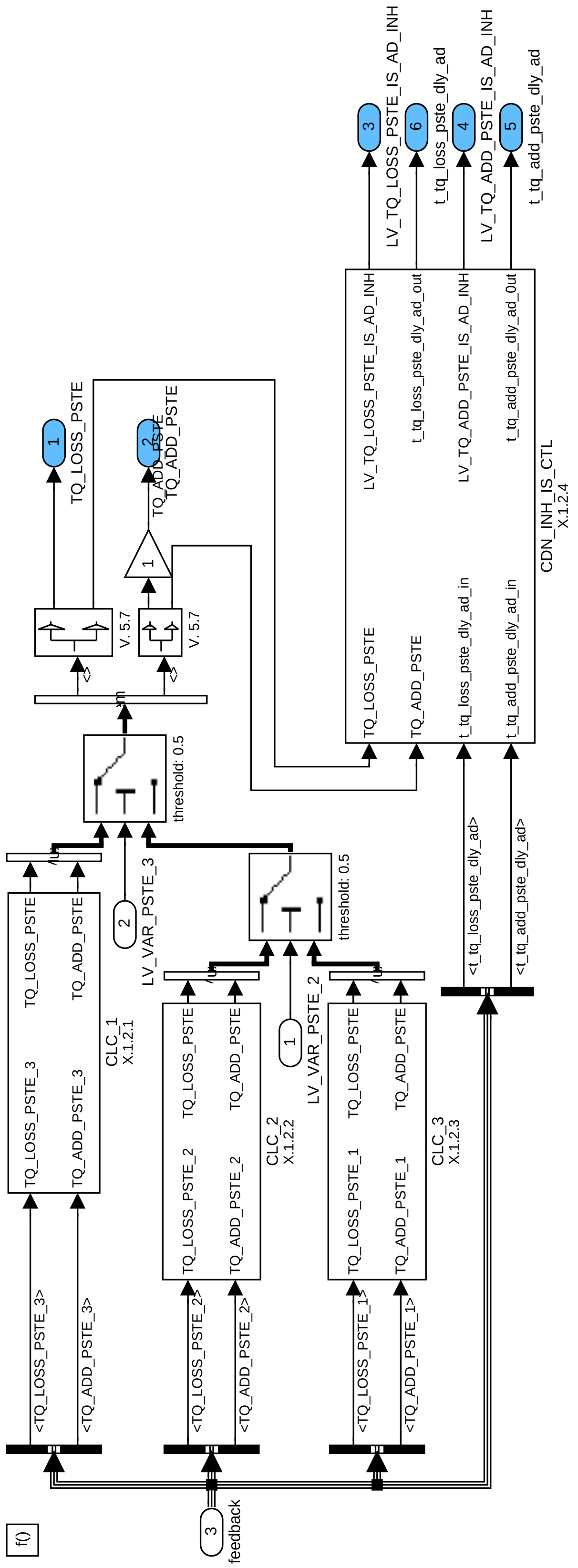


Figure 25:
Path: STSY_MD013/SUM_PSTE/OPM

58.3.1.2.1 Calculation of TQ_LOSS_PSTE and TQ_ADD_PSTE when LV_VAR_PSTE_3 is set

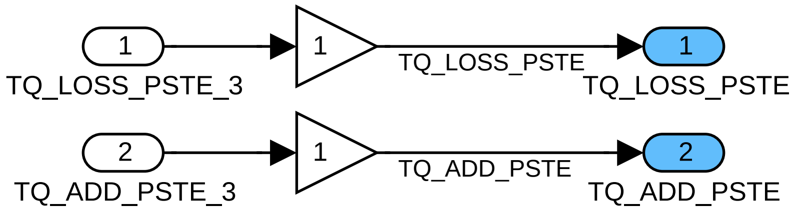


Figure 26:
Path: STSY_MD013/SUM_PSTE/OPM/CLC_1

58.3.1.2.2 Calculation of TQ_LOSS_PSTE and TQ_ADD_PSTE when LV_VAR_PSTE_2 is set

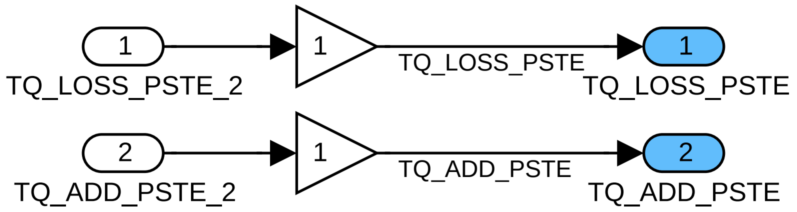


Figure 27:
Path: STSY_MD013/SUM_PSTE/OPM/CLC_2

58.3.1.2.3 Calculation of TQ_LOSS_PSTE and TQ_ADD_PSTE when LV_VAR_PSTE_2 is reset

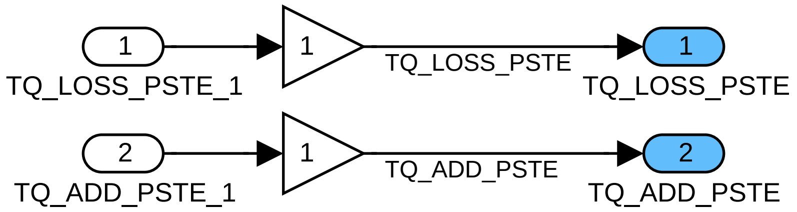


Figure 28:
Path: STSY_MD013/SUM_PSTE/OPM/CLC_3

58.3.1.2.4 Condition for inhibition of IS-control adaptations:

58.3.1.2.4.1 Calculation of LV_TQ_LOSS_PSTE_IS_AD_INH and TIMER

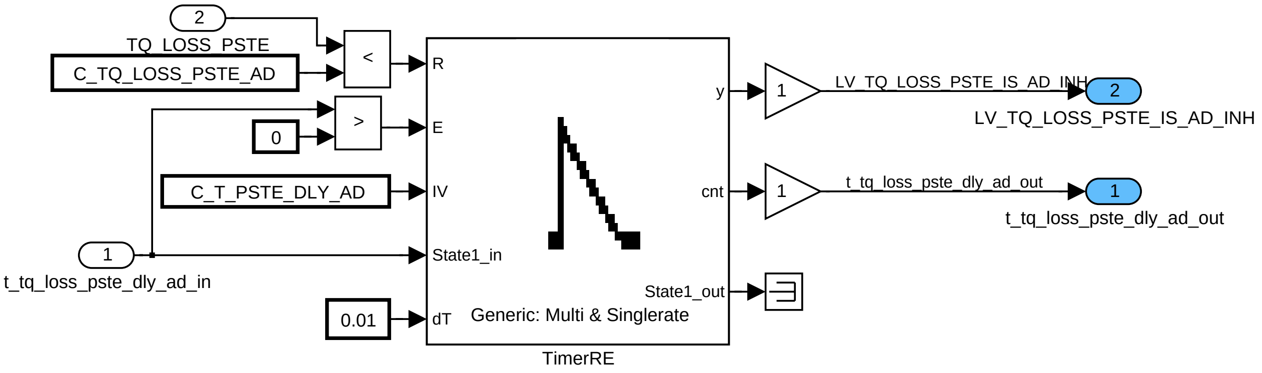


Figure 29:
Path: STSY_MD013/SUM_PSTE/OPM/CDN_INH_IS_CTL/CLC_1

58.3.1.2.4.2 Calculation of LV_TQ_ADD_PSTE_IS_AD_INH and TIMER

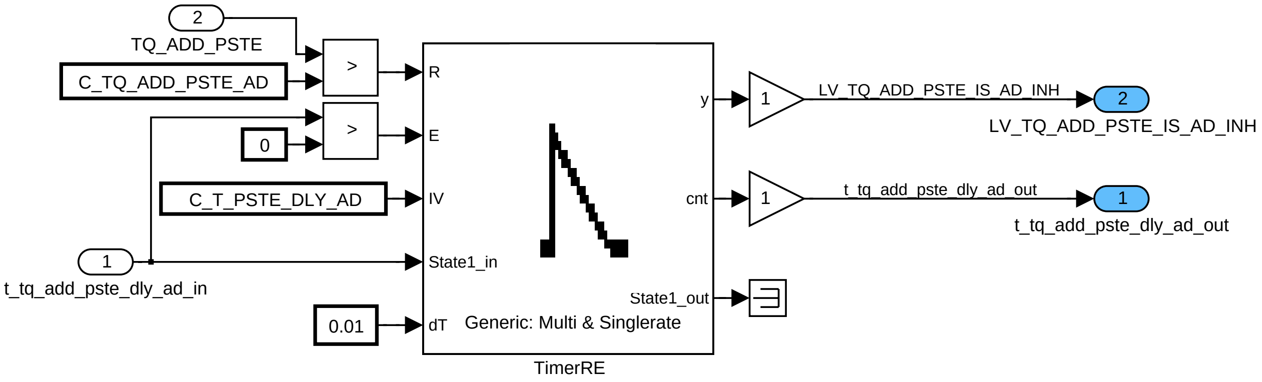


Figure 30:
Path: STSY_MD013/SUM_PSTE/OPM/CDN_INH_IS_CTL/CLC_2

58.3.2 Torque losses power steering assistance

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: LV_IGK && LV_VAR_PSTE
Deactivation: never

Function description

58.3.2.1 Reset of variables

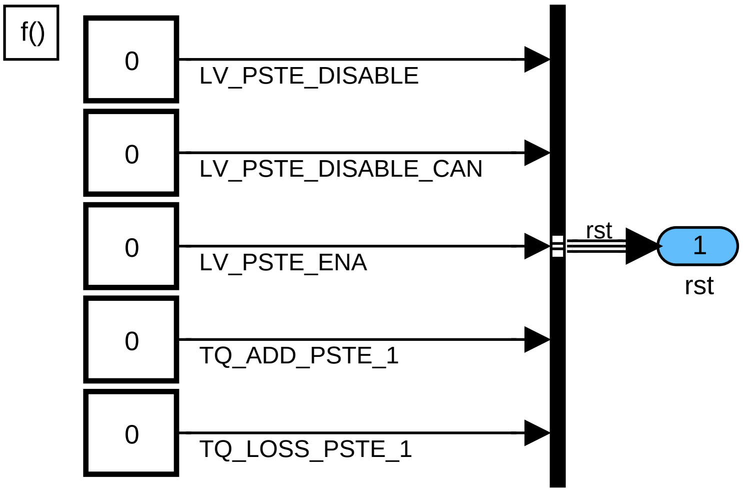


Figure 31:
Path: STSY_MD013/TQLO_PSTEA/RST

58.3.2.2 Function description.

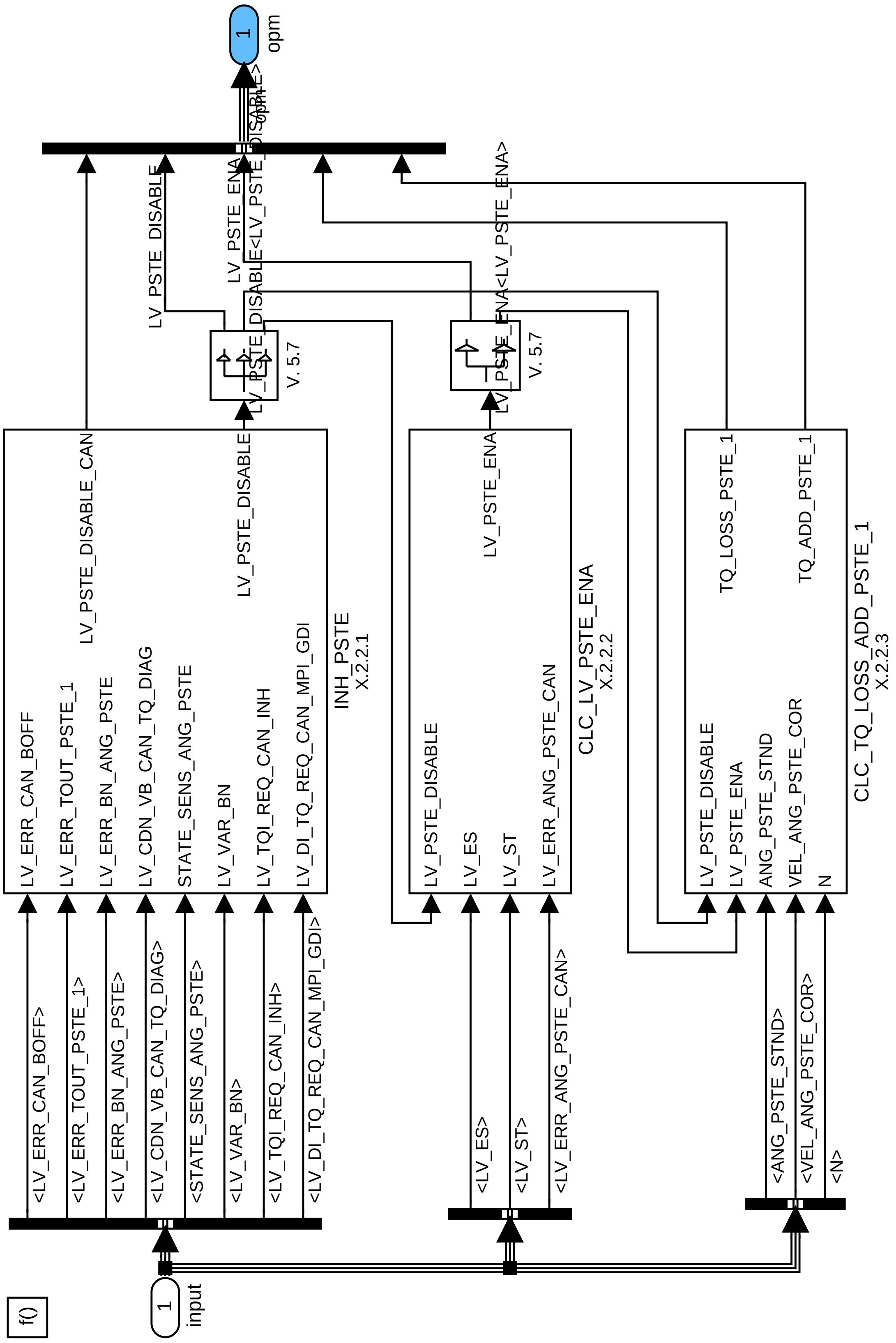


Figure 32:
Path: STSY_MD013/TQLO_PSTEA/OPM

58.3.2.2.1 Inhibition of power steering

General information:
TQ_LOSS_PSTE_1 describes the needed torque during activation of power steering. ANG_PSTE is the angle of power steering and VEL_ANG_PSTE the velocity of ANG_PSTE.

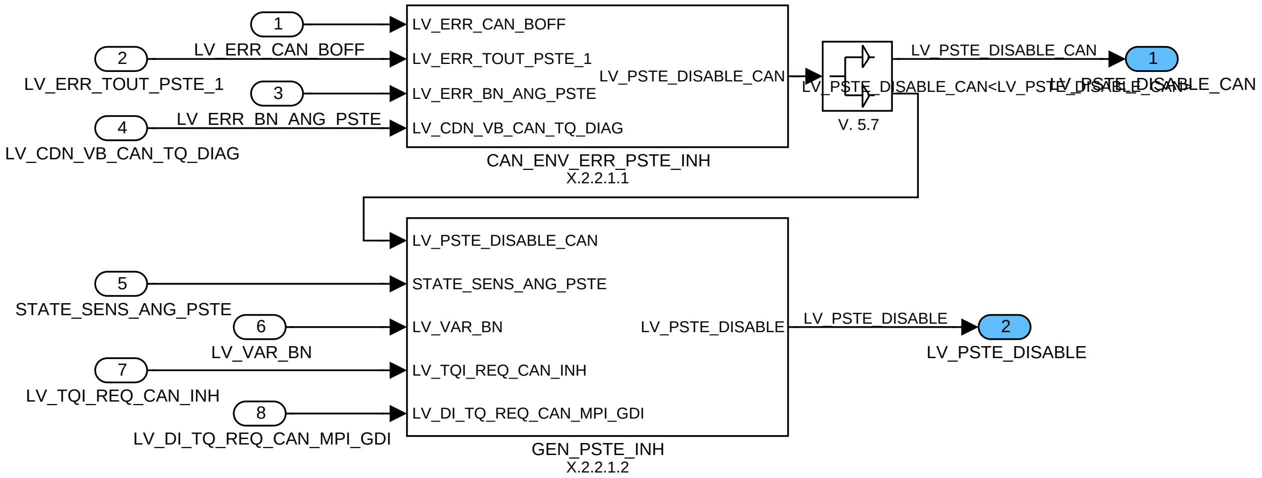


Figure 33:
Path: STSY_MD013/TQLO_PSTEA/OPM/INH_PSTE

58.3.2.2.1.1 CAN environment error PSTE inhibition:

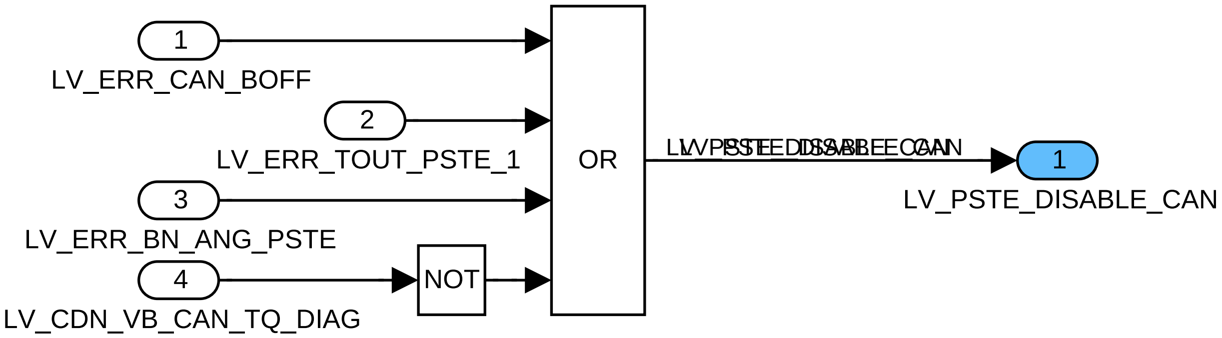


Figure 34:
Path: STSY_MD013/TQLO_PSTEA/OPM/INH_PSTE/CAN_ENV_ERR_PSTE_INH

58.3.2.2.1.2 General PSTE inhibition:

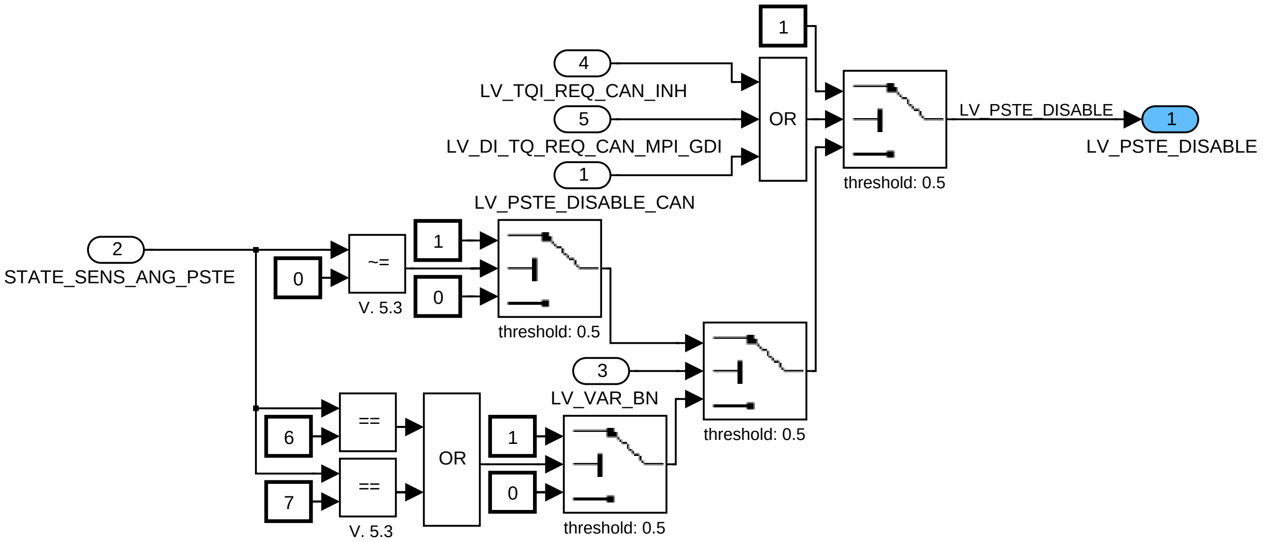


Figure 35:
Path: STSY_MD013/TQLO_PSTEA/OPM/INH_PSTE/GEN_PSTE_INH

58.3.2.2.2 Enable conditions PSTE intervention

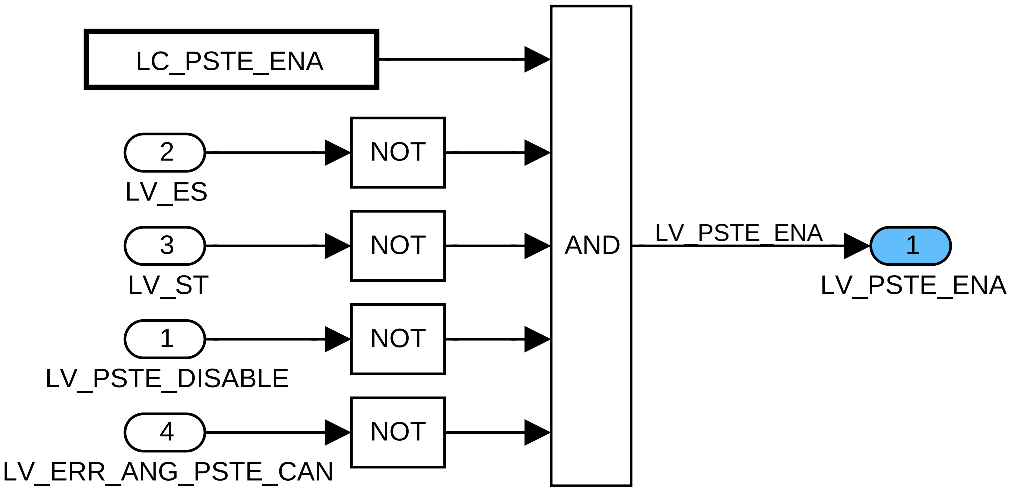


Figure 36:
Path: STSY_MD013/TQLO_PSTEA/OPM/CLC_LV_PSTE_ENA

58.3.2.2.3 Torque loss & reserve for power steering assistance

Calculation of TQ_LOSS_PSTE_1 and TQ_ADD_PSTE_1.

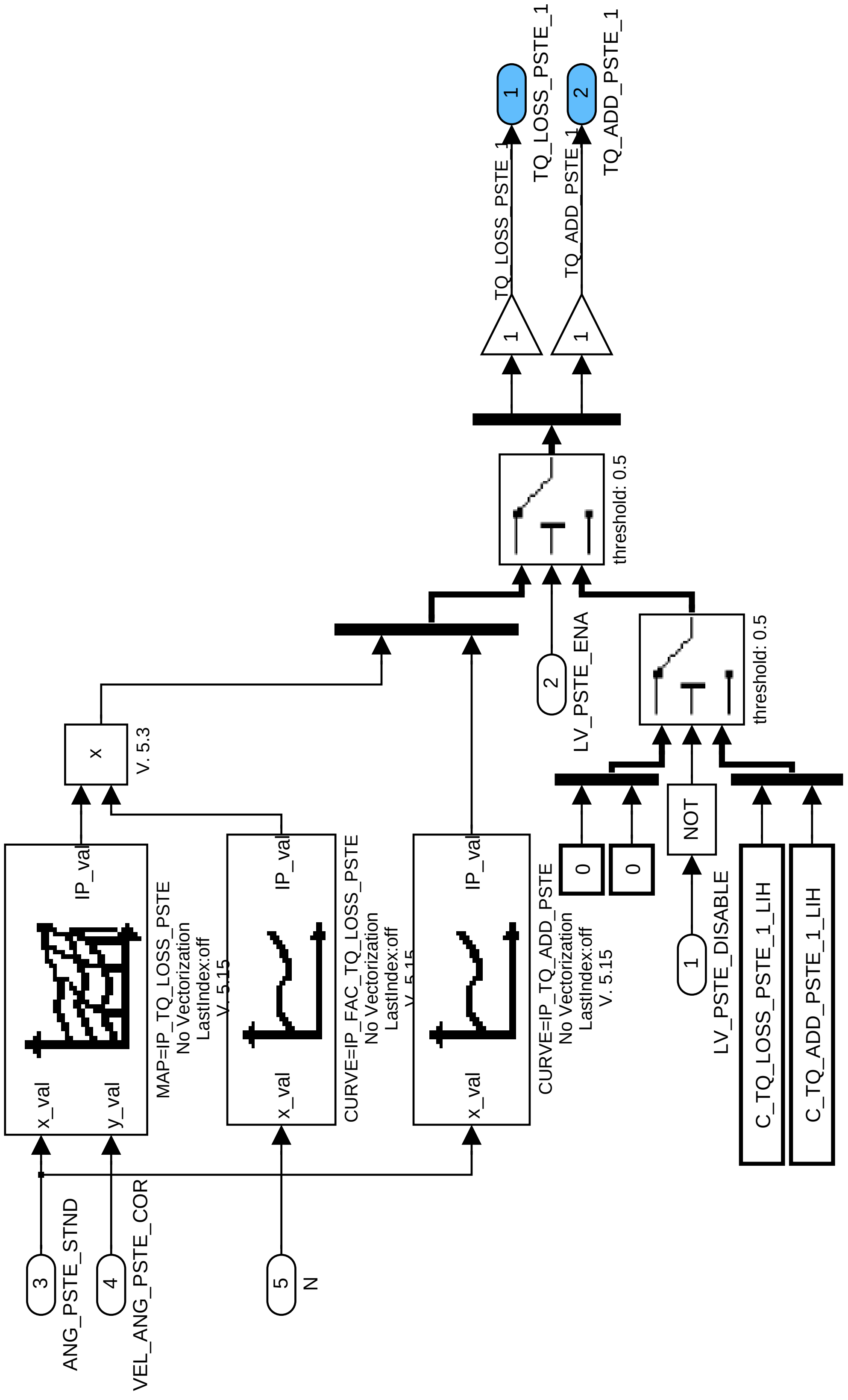


Figure 37:
Path: STSY_MD013/TQLO_PSTEA/OPM/CLC_TQ_LOSS_ADD_PSTE_1

58.3.3 Torque Loss for active power steering (BN 2000)

General information:

general specification

TQ_LOSS_PSTE_2 describes the torque demand and losses of active power steering. There are two informations the predicted and the Active Torque losses.

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: LV_IGK && LV_VAR_BN && LV_VAR_PSTE_2
Deactivation: never

Function description

58.3.3.1 Reset of variables.

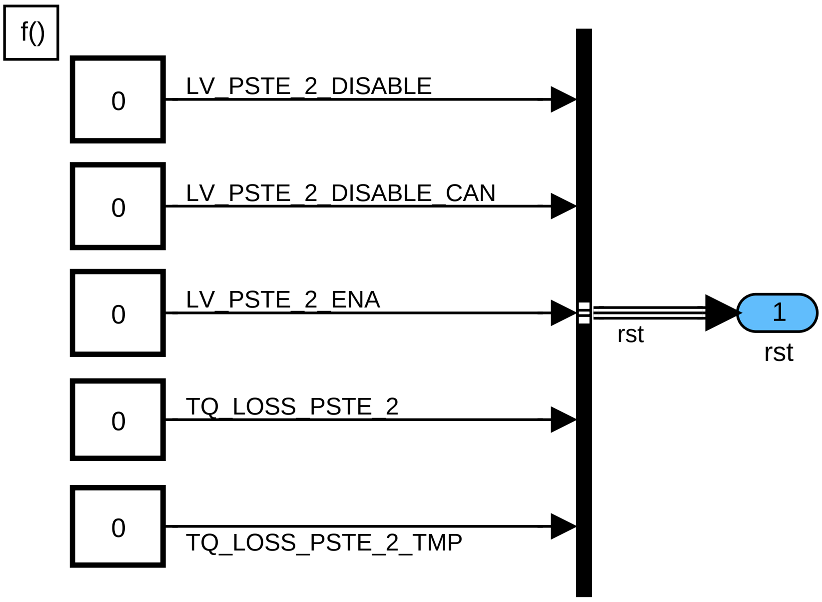


Figure 38:
Path: STSY_MD013/TQLO_ACT_PSTE/RST

58.3.3.2 Function description

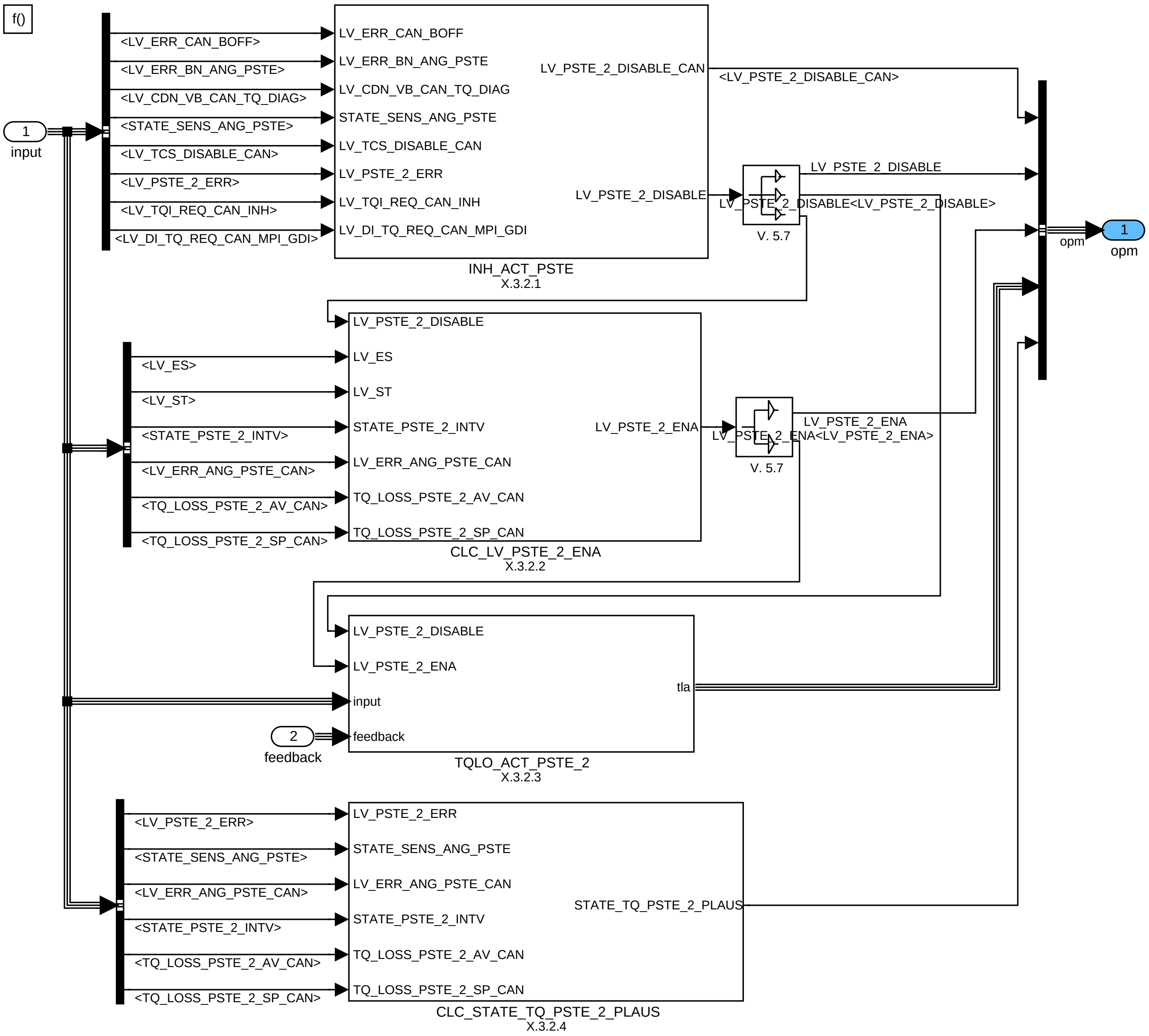


Figure 39:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM

58.3.3.2.1 Inhibition of active power steering

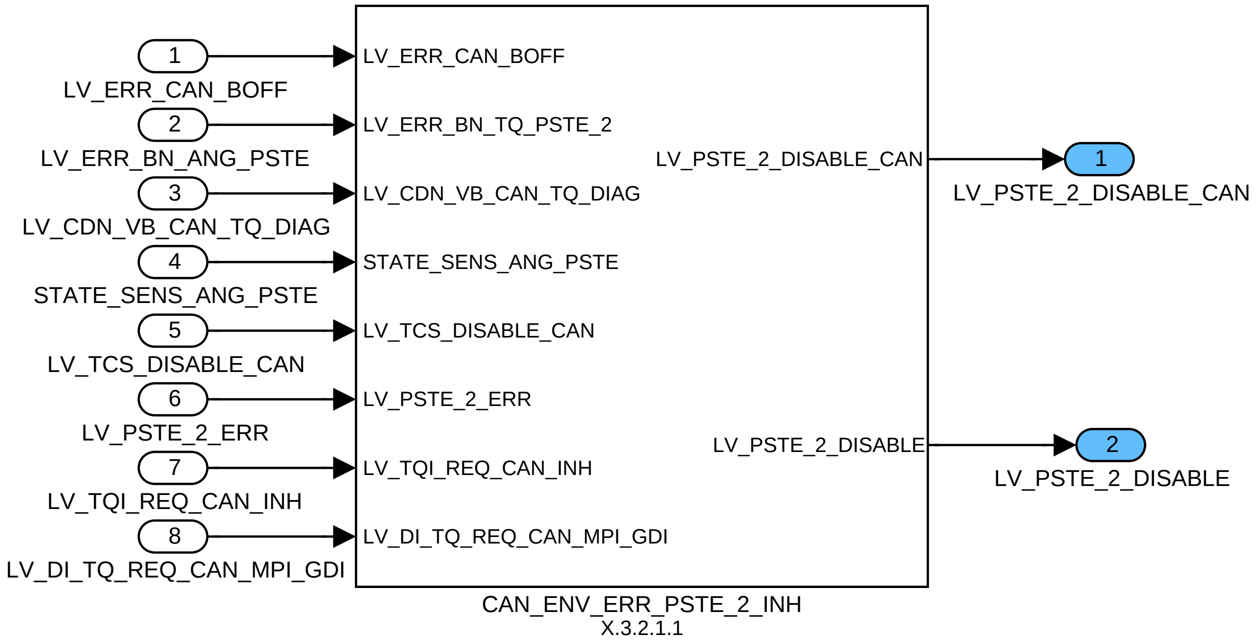


Figure 40:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/INH_ACT_PSTE

58.3.3.2.1.1 CAN environment error PSTE_2 inhibition:

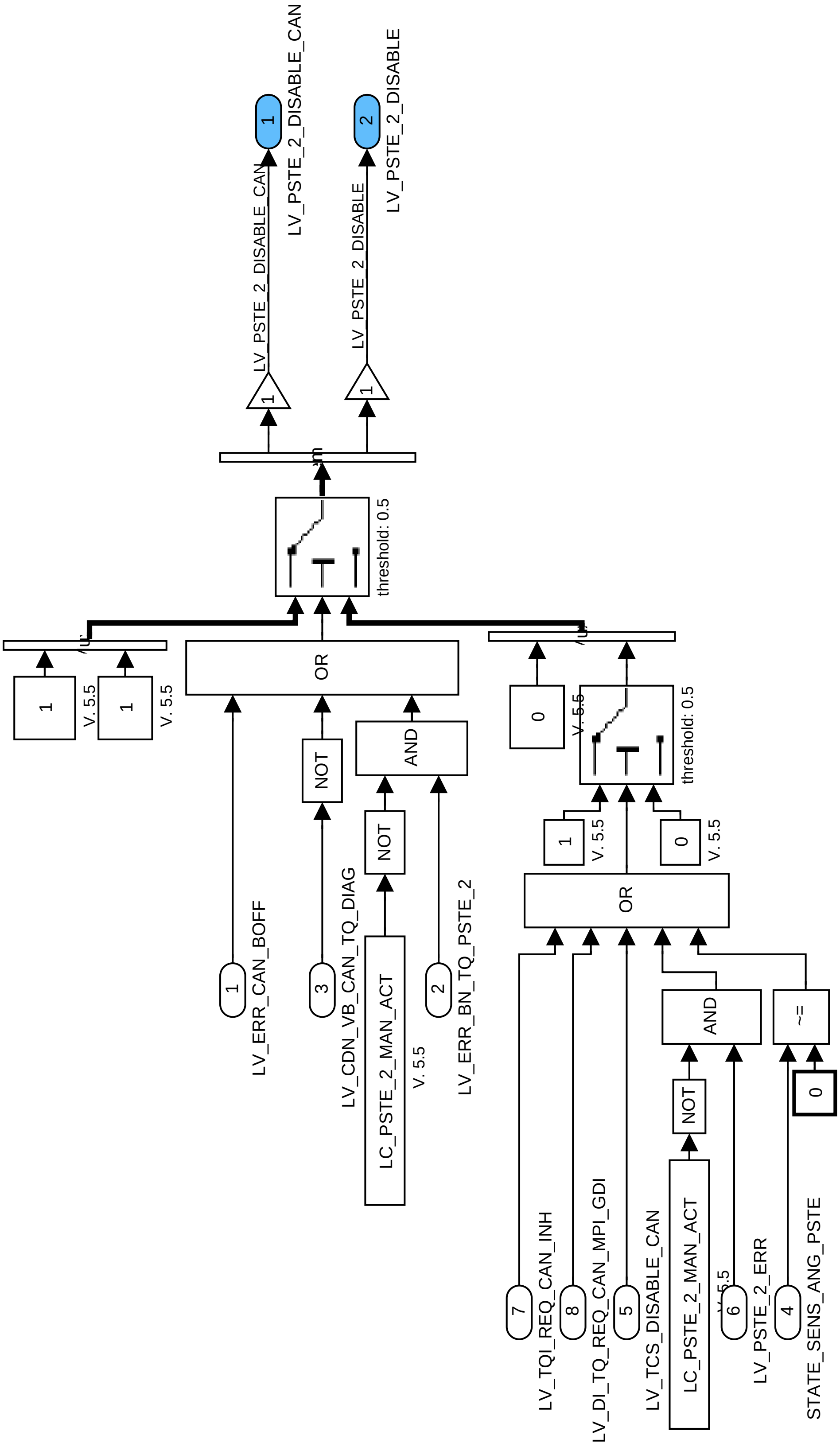


Figure 41:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/INH_ACT_PSTE/CAN_ENV_ERR_PSTE_2_INH

58.3.3.2.2 Enable conditions PSTE intervention

FUNCTION DESCRIPTION:
The State STATE_TQ_PSTE_2_PLAUS indicates the interrupted condition which is responsible for enable Bit = 0. The [x] Value shows the HEX adress inside the Bitwise coded State.

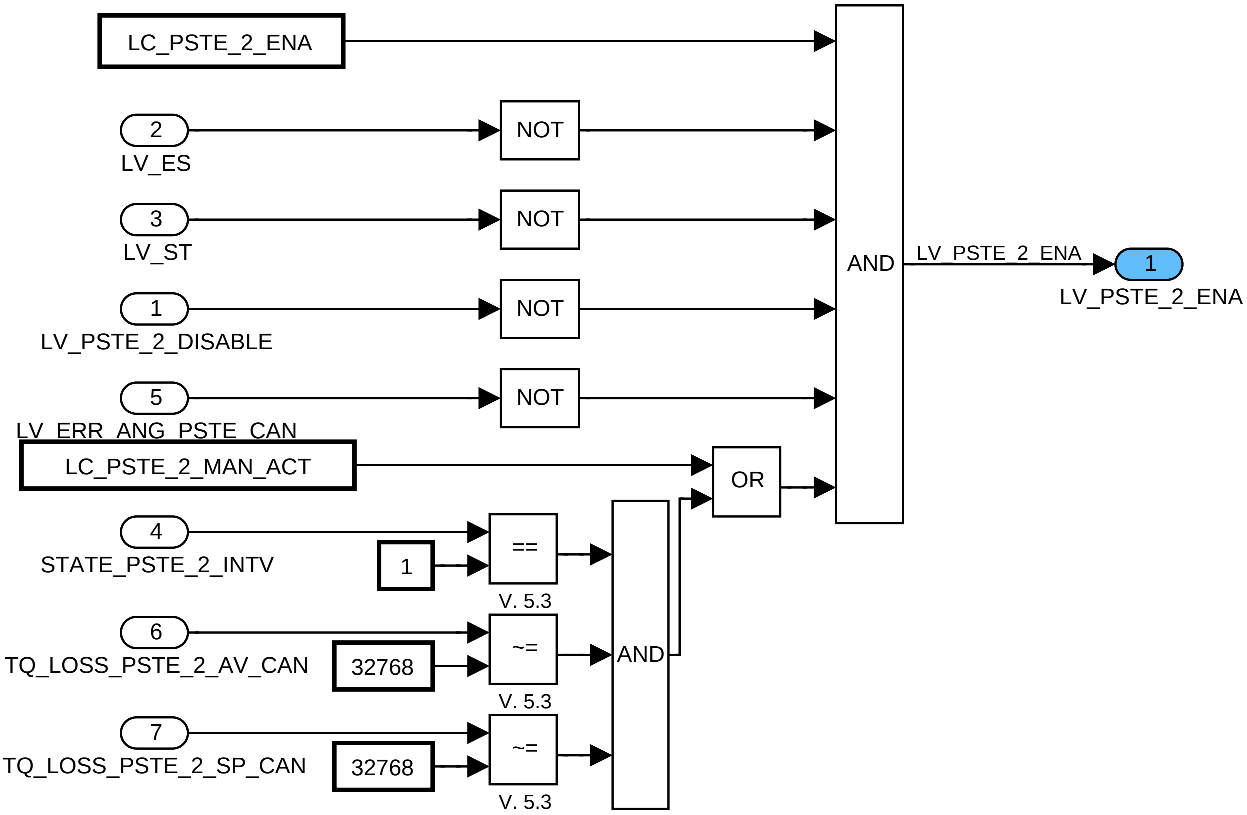


Figure 42:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/CLC_LV_PSTE_2_ENA

58.3.3.2.3 Torque loss active PSTE_2

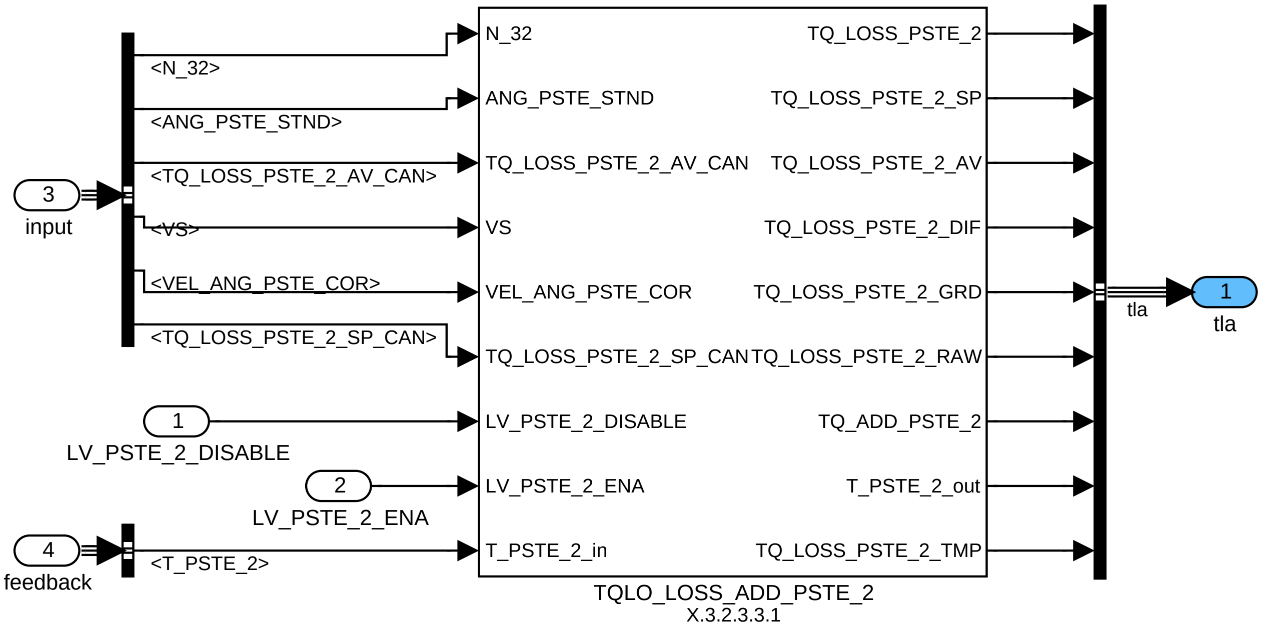


Figure 43:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2

58.3.3.2.3.1 Calculation depending on LV_PSTE_2_ENA

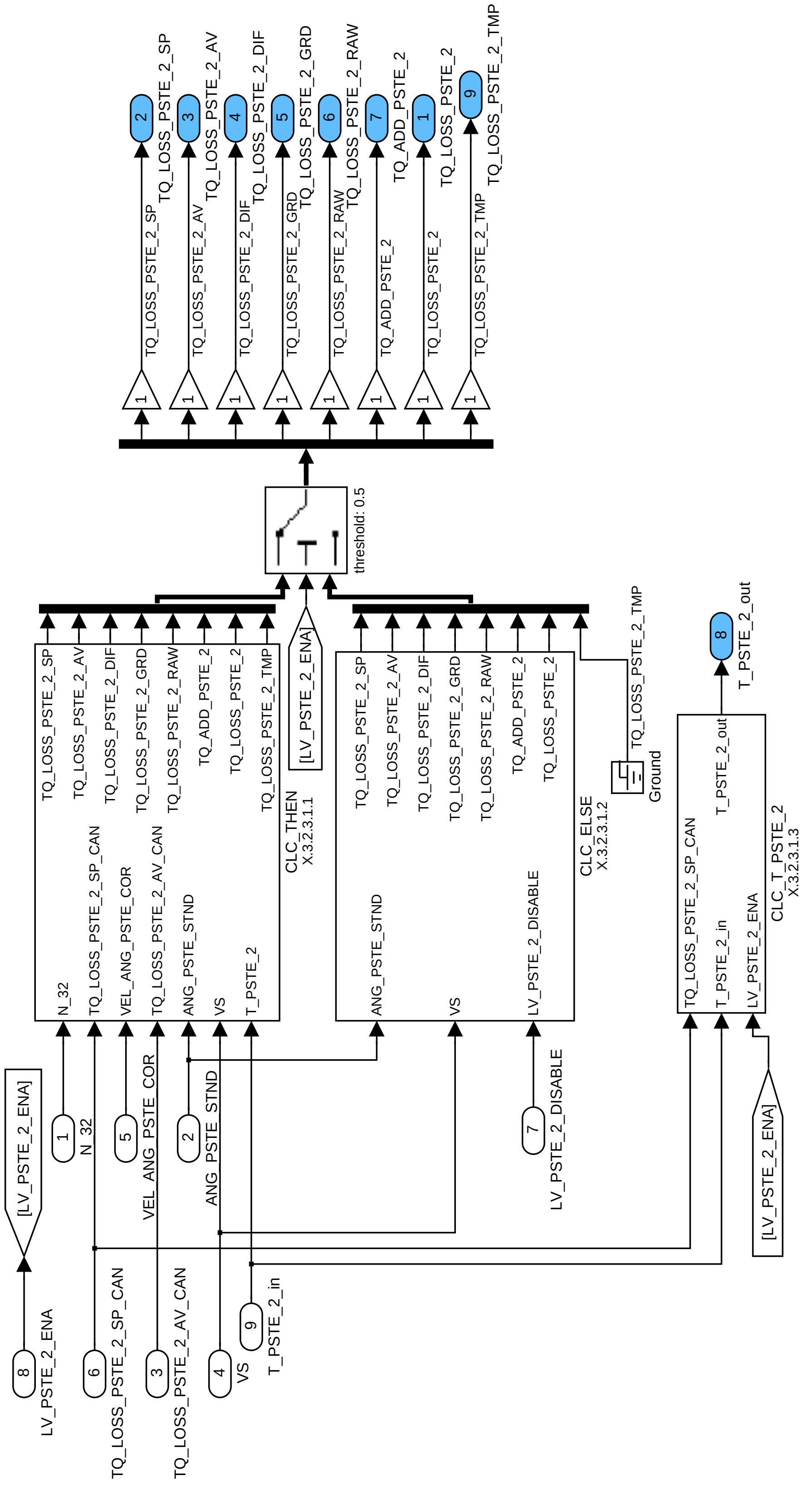


Figure 44:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2/TQLO_LOSS_ADD_PSTE_2

58.3.3.2.3.1.1 Calculation when LV_PSTE_2_ENA is true.

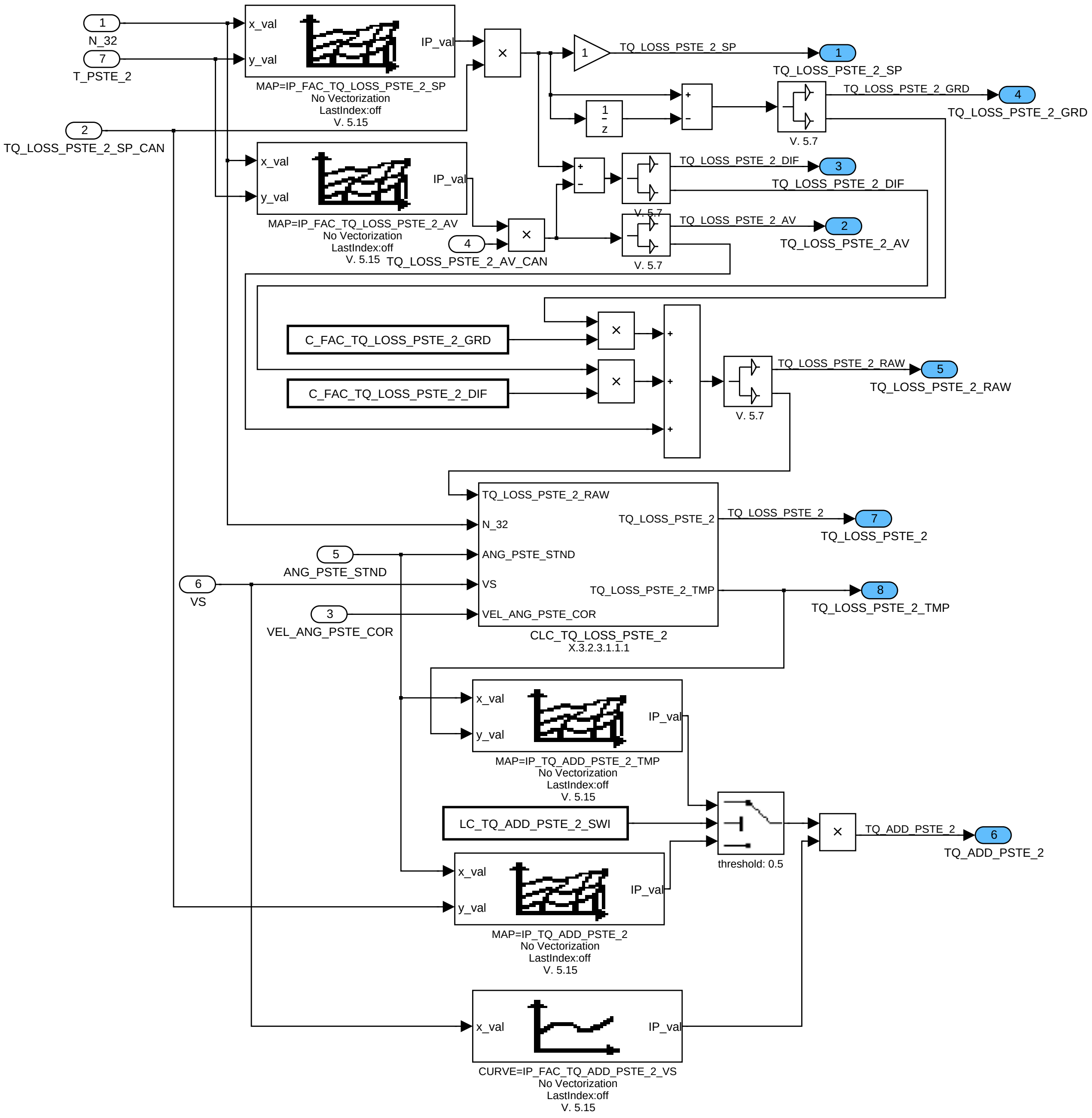


Figure 45:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2/TQLO_LOSS_ADD_PSTE_2/CLC_THEN

58.3.3.2.3.1.1.1 Calculation of TQ_LOSS_PSTE_2

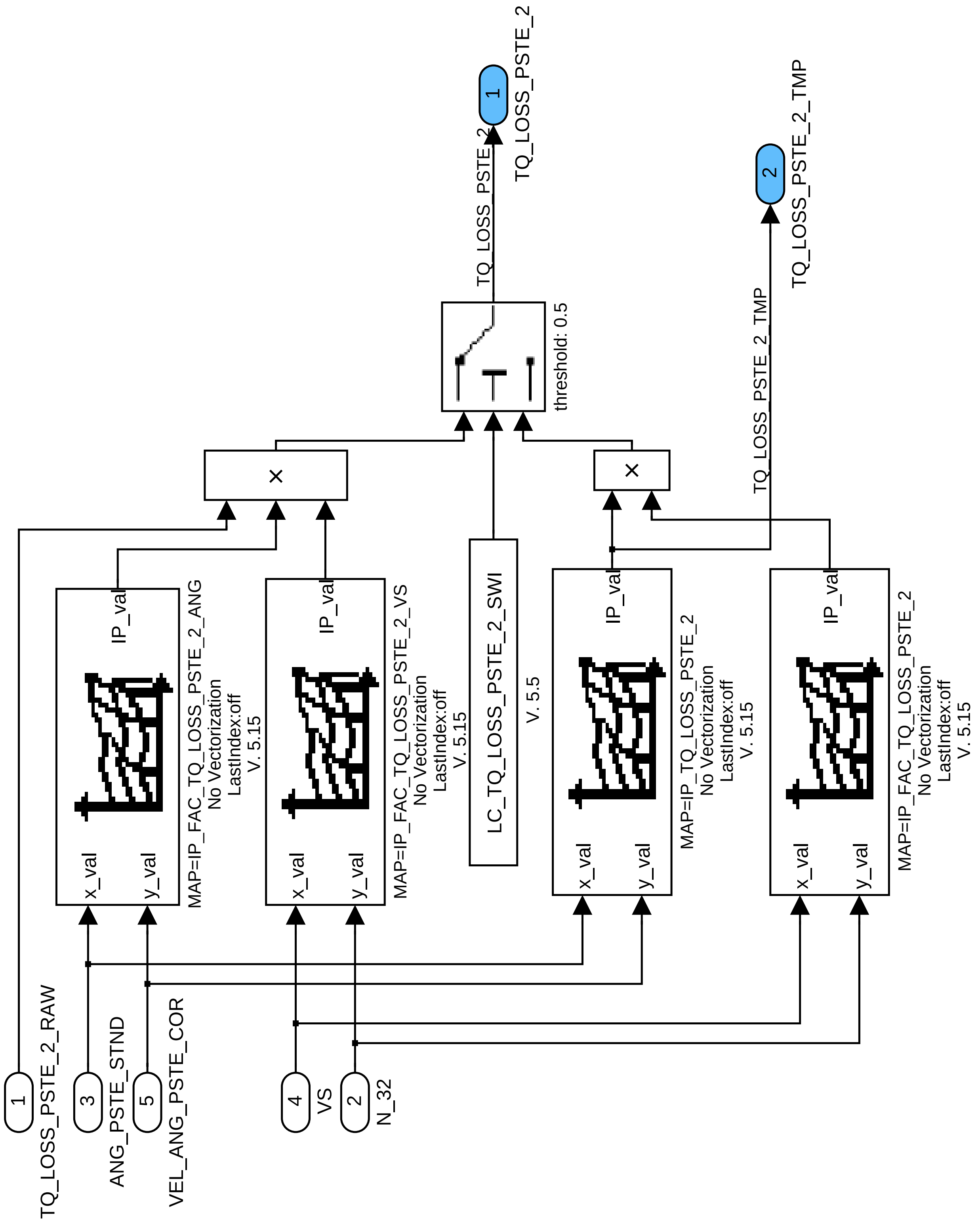


Figure 46:
Path:
STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2/TQLO_LOSS_ADD_PSTE_2/CLC_THEN/CLC_TQ_LOSS_PSTE_2

58.3.3.2.3.1.2 Calculation when LV_PSTE_2_ENA is false.

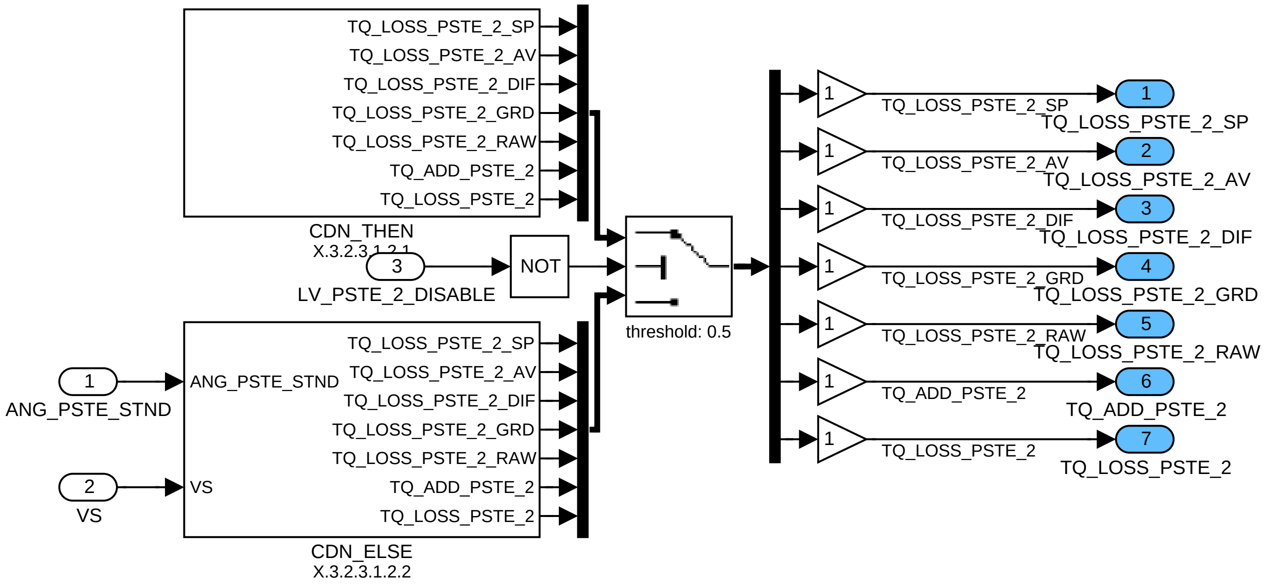


Figure 47:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2/TQLO_LOSS_ADD_PSTE_2/CLC_ELSE

58.3.3.2.3.1.2.1 Calculation when LV_PSTE_2_DISABLE is true.

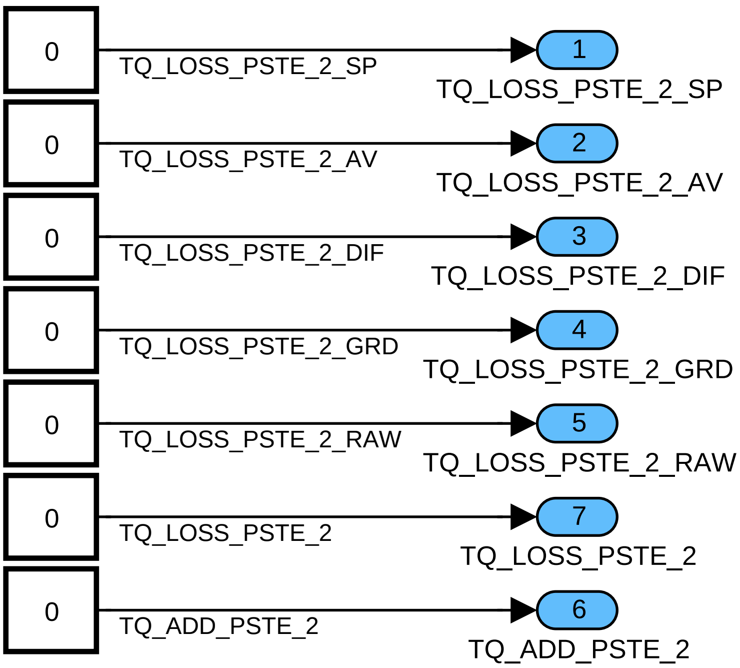


Figure 48:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2/TQLO_LOSS_ADD_PSTE_2/CLC_ELSE/CDN_THEN

58.3.3.2.3.1.2.2 Calculation depending on LV_PSTE_2_DISABLE.

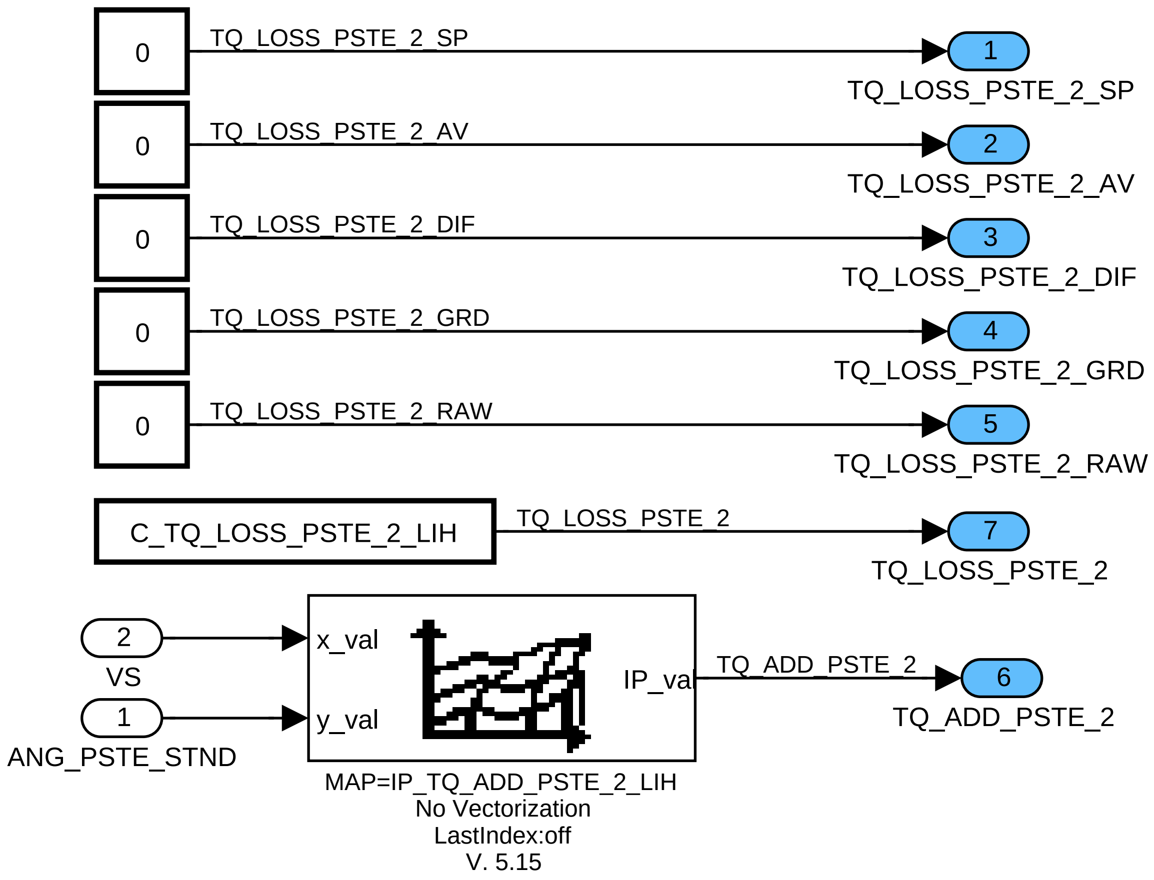


Figure 49:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2/TQLO_LOSS_ADD_PSTE_2/CLC_ELSE/CDN_ELSE

58.3.3.2.3.1.3 Timer calculation.

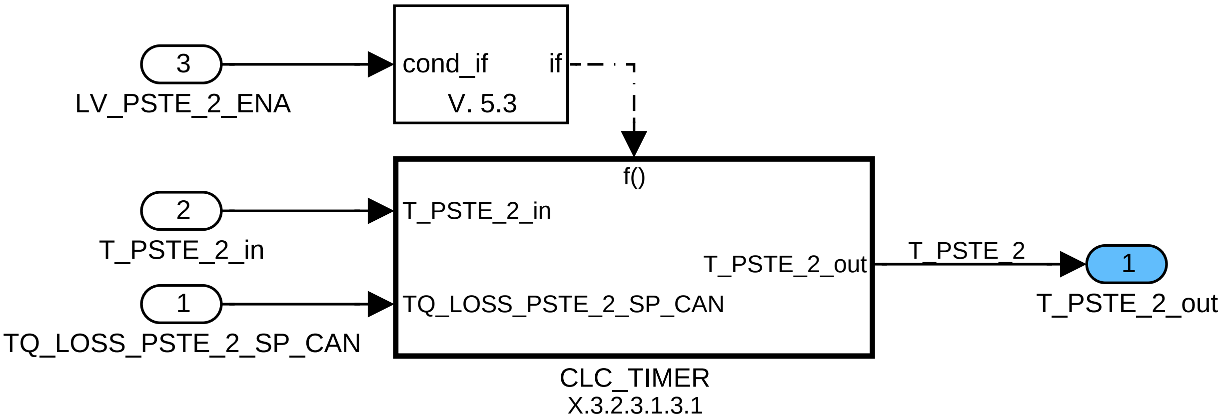


Figure 50:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2/TQLO_LOSS_ADD_PSTE_2/CLC_T_PSTE_2

58.3.3.2.3.1.3.1 Calculation of timer T_PSTE_2.

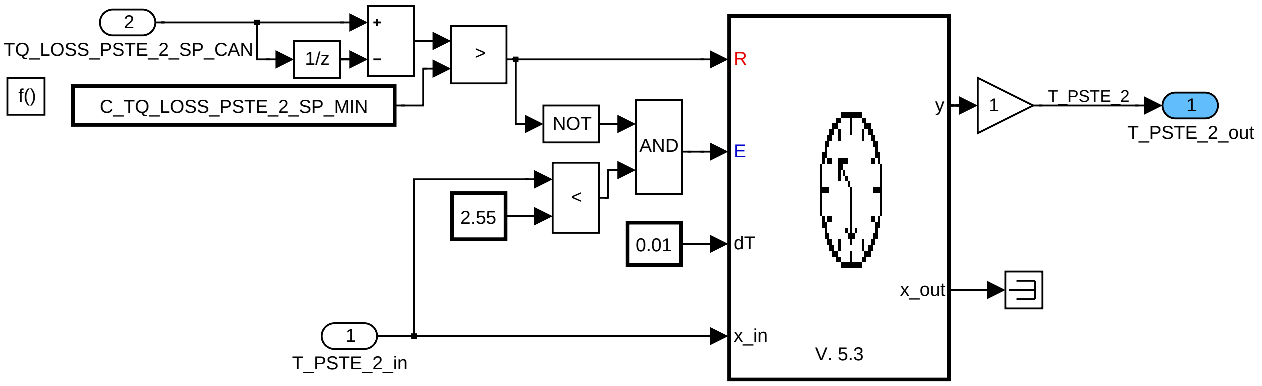


Figure 51:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/TQLO_ACT_PSTE_2/TQLO_LOSS_ADD_PSTE_2/CLC_T_PSTE_2/CLC_TIMER

58.3.3.2.4 STATE_TQ_PSTE_2_PLAUS

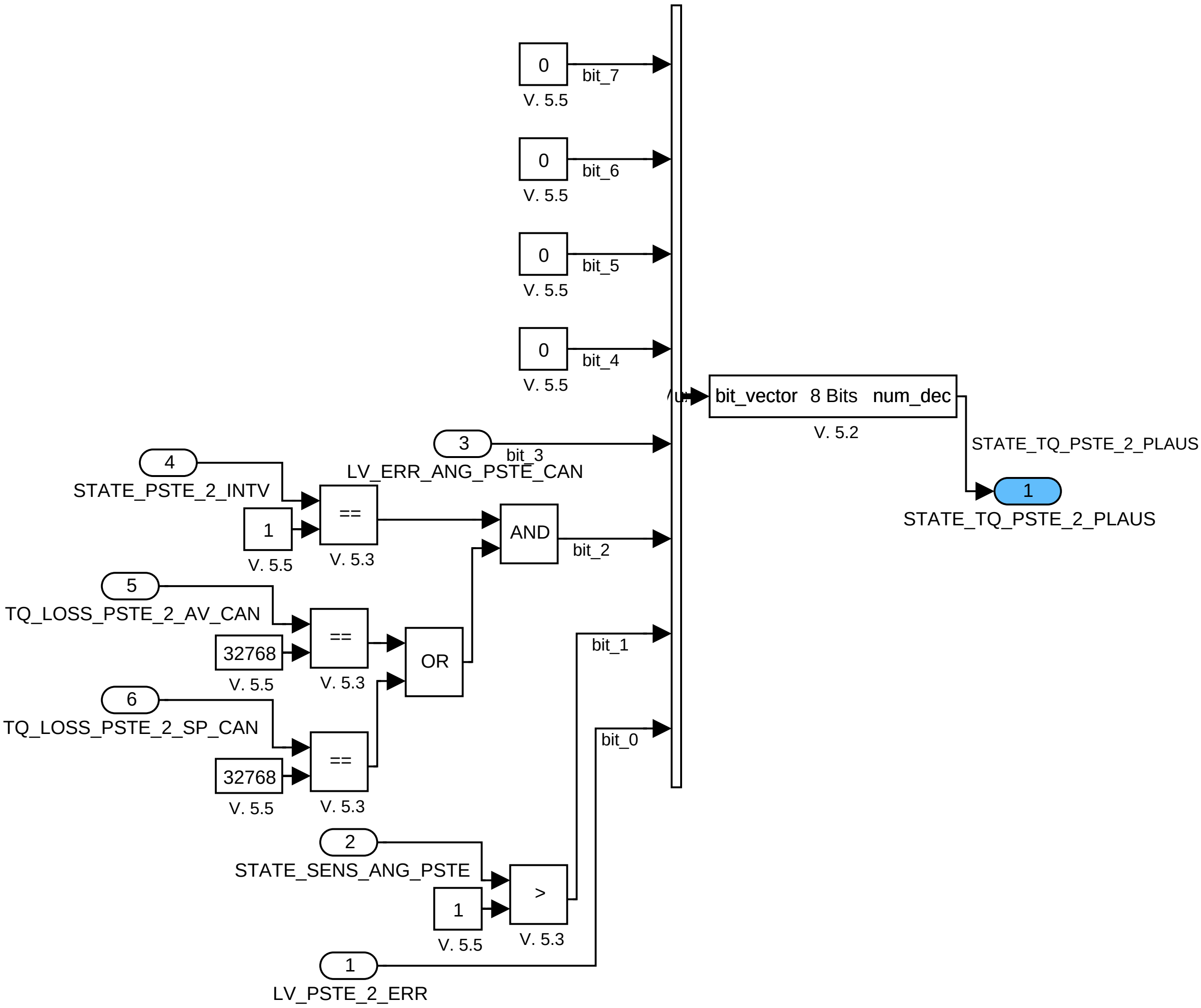


Figure 52:
Path: STSY_MD013/TQLO_ACT_PSTE/OPM/CLC_STATE_TQ_PSTE_2_PLAUS

58.3.4 Torque Loss for active power steering via EHB3 Control Unit (BN 2000)

General information:

general specification

TQ_LOSS_PSTE_3 describes the torque demand and losses of active power steering. Torque loss and reserve are calculated via torque request from EHB3-CPU.

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: LV_IGK && LV_VAR_PSTE_3
Deactivation: never

Function description

58.3.4.1 Reset of variables.

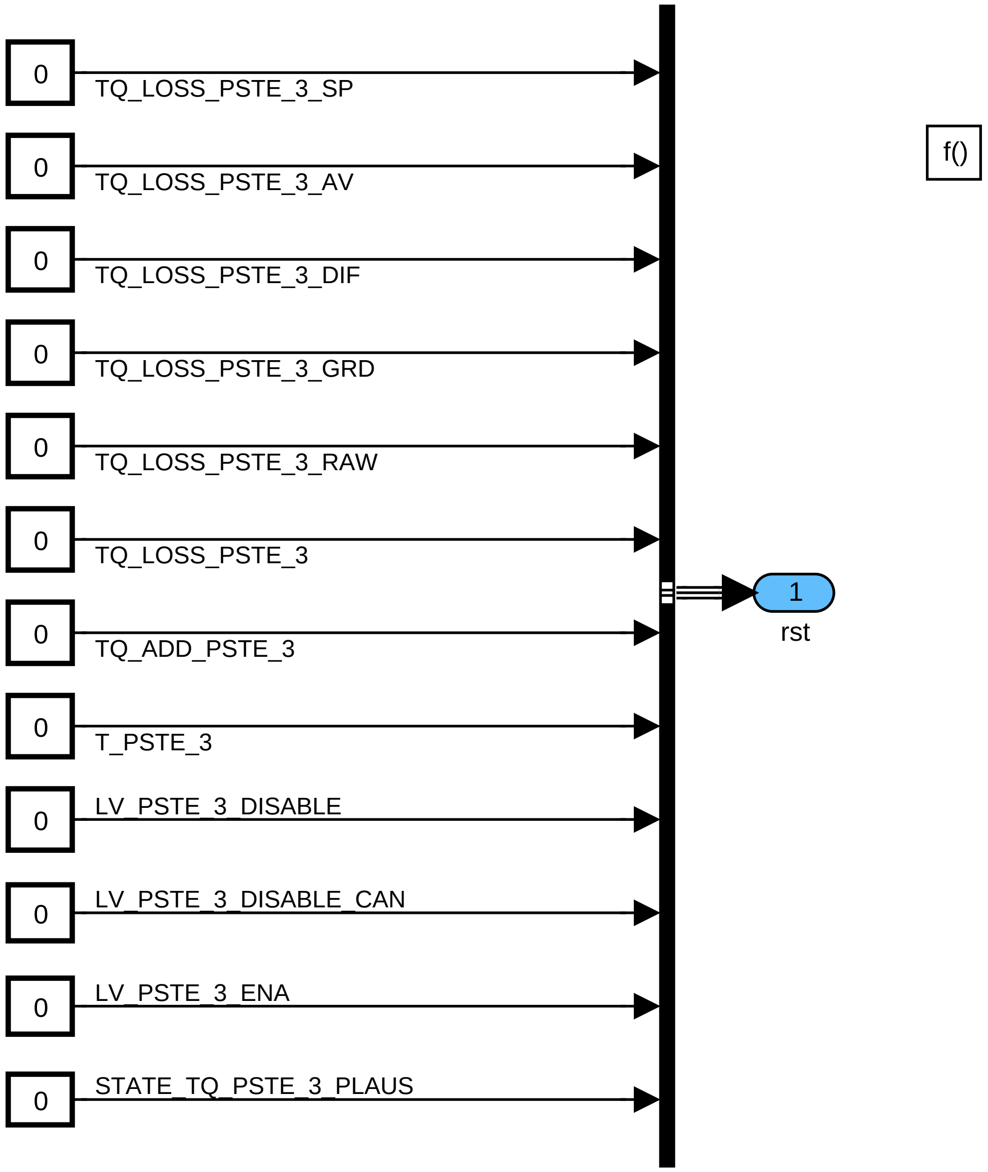


Figure 53:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/RST

58.3.4.2 Function description.

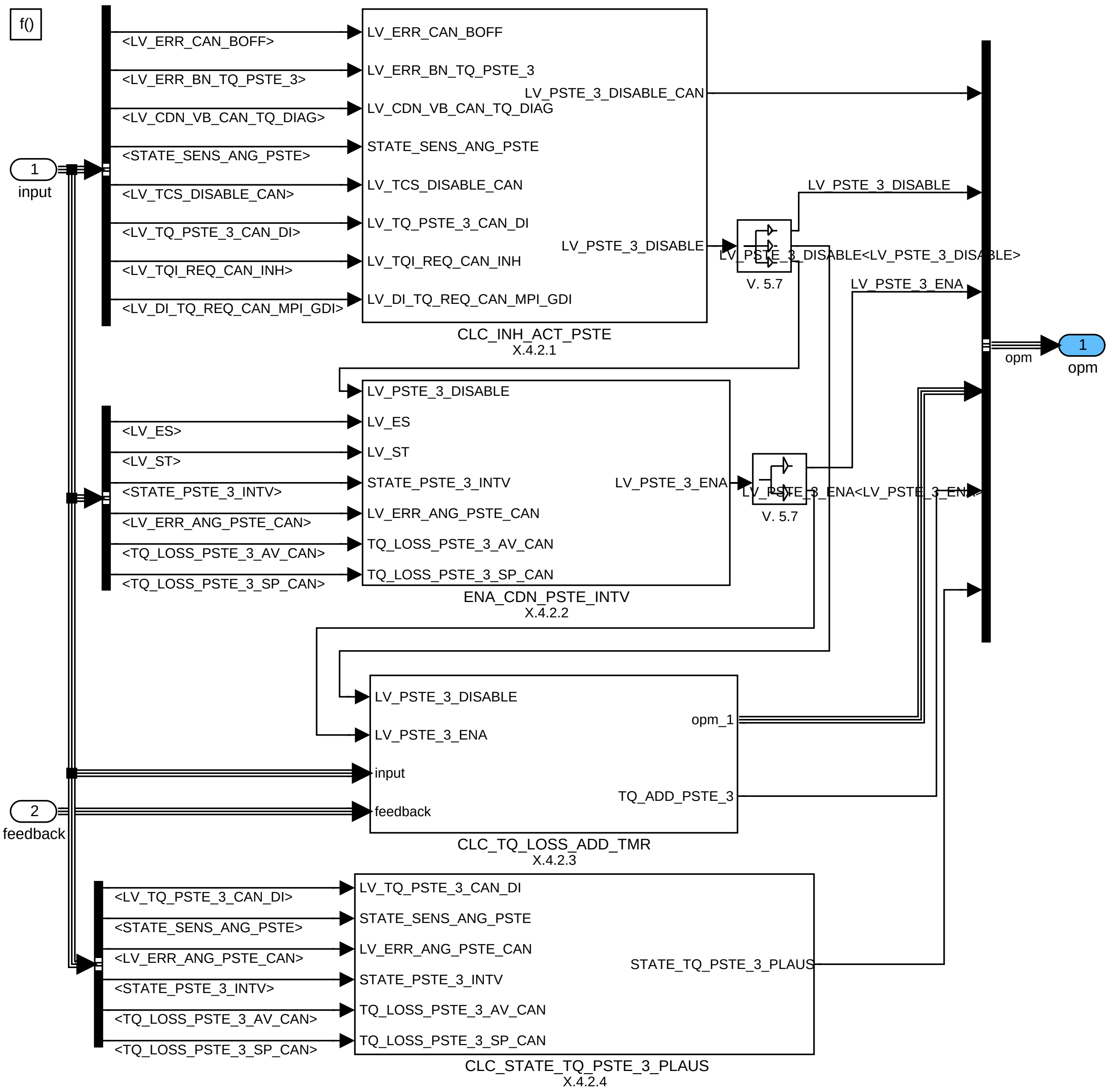


Figure 54:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM

58.3.4.2.1 Inhibition of active power steering

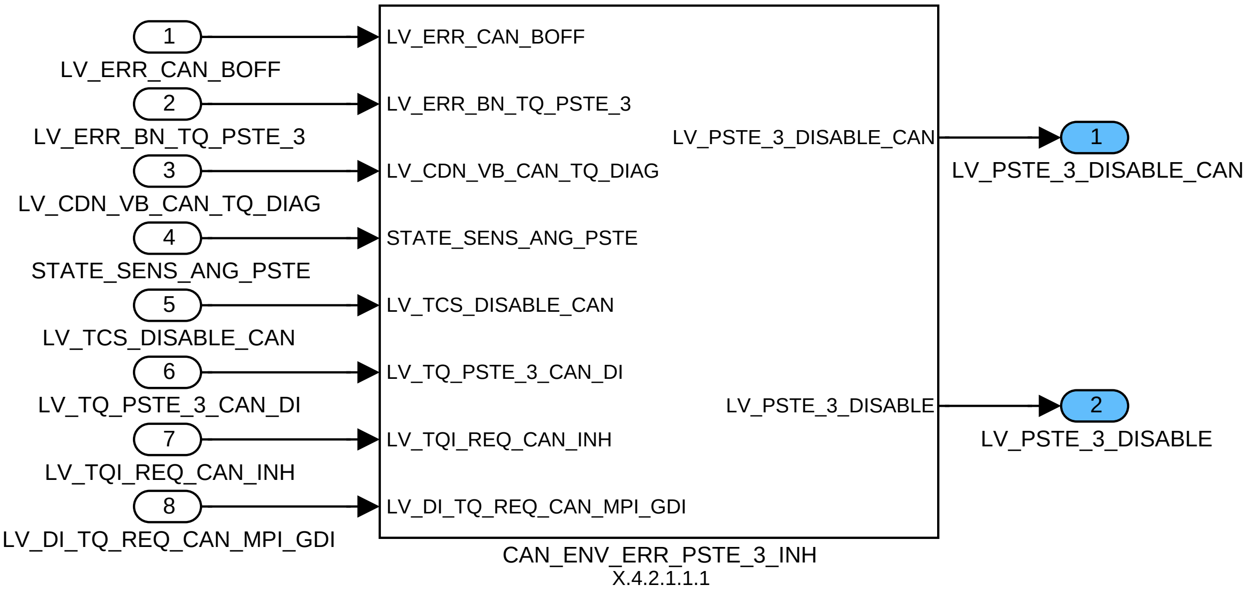


Figure 55:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_INH_ACT_PSTE

58.3.4.2.1.1 CAN environment error PSTE_3 inhibition:

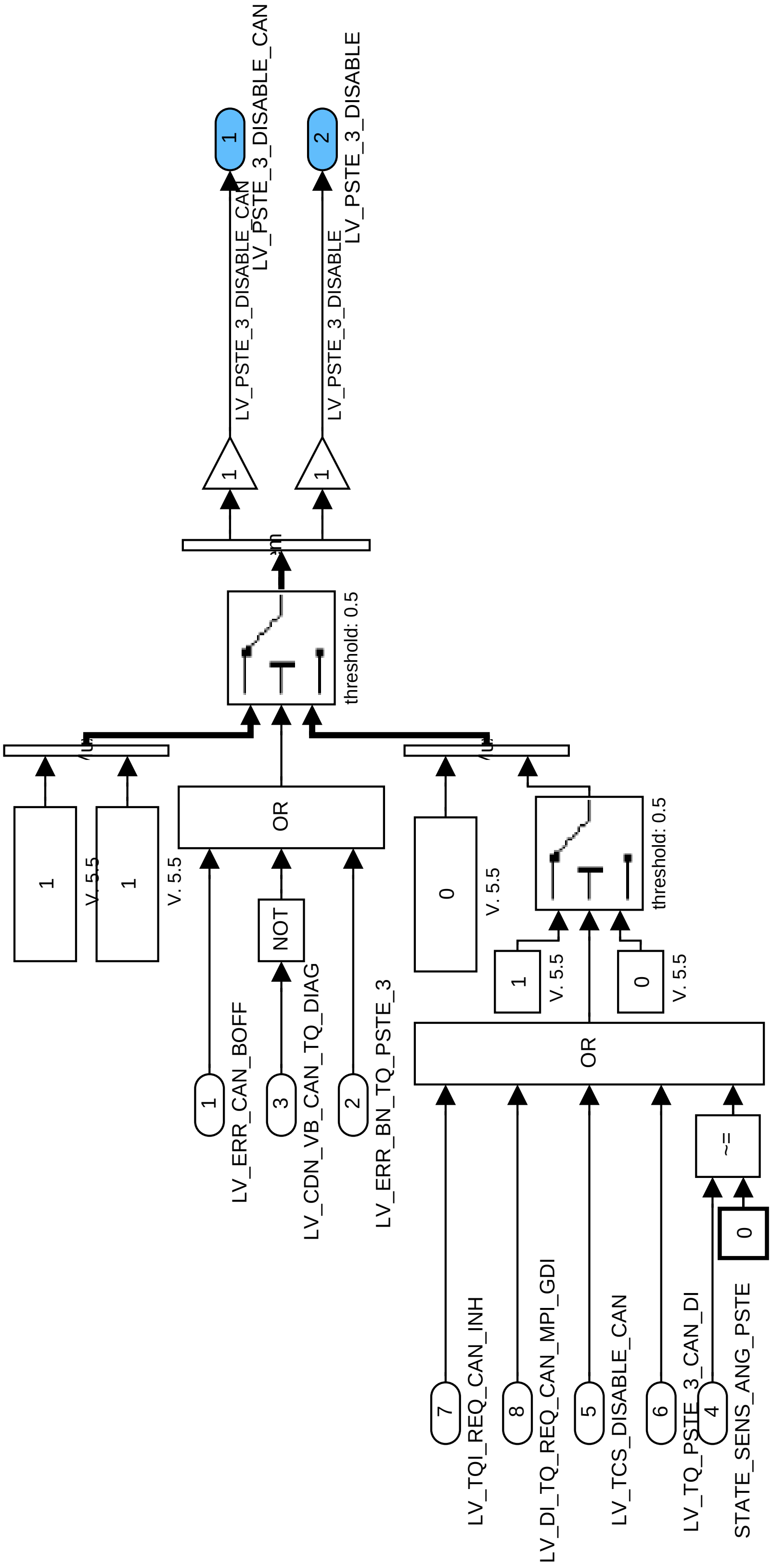


Figure 56:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_INH_ACT_PSTE/CAN_ENV_ERR_PSTE_3_INH

58.3.4.2.2 Enable conditions PSTE_3 intervention

FUNCTION DESCRIPTION:
The State STATE_TQ_PSTE_3_PLAUS indicates the interrupted condition which is responsible for enable Bit = 0. The [x] Value shows the HEX adress inside the Bitwise coded State.

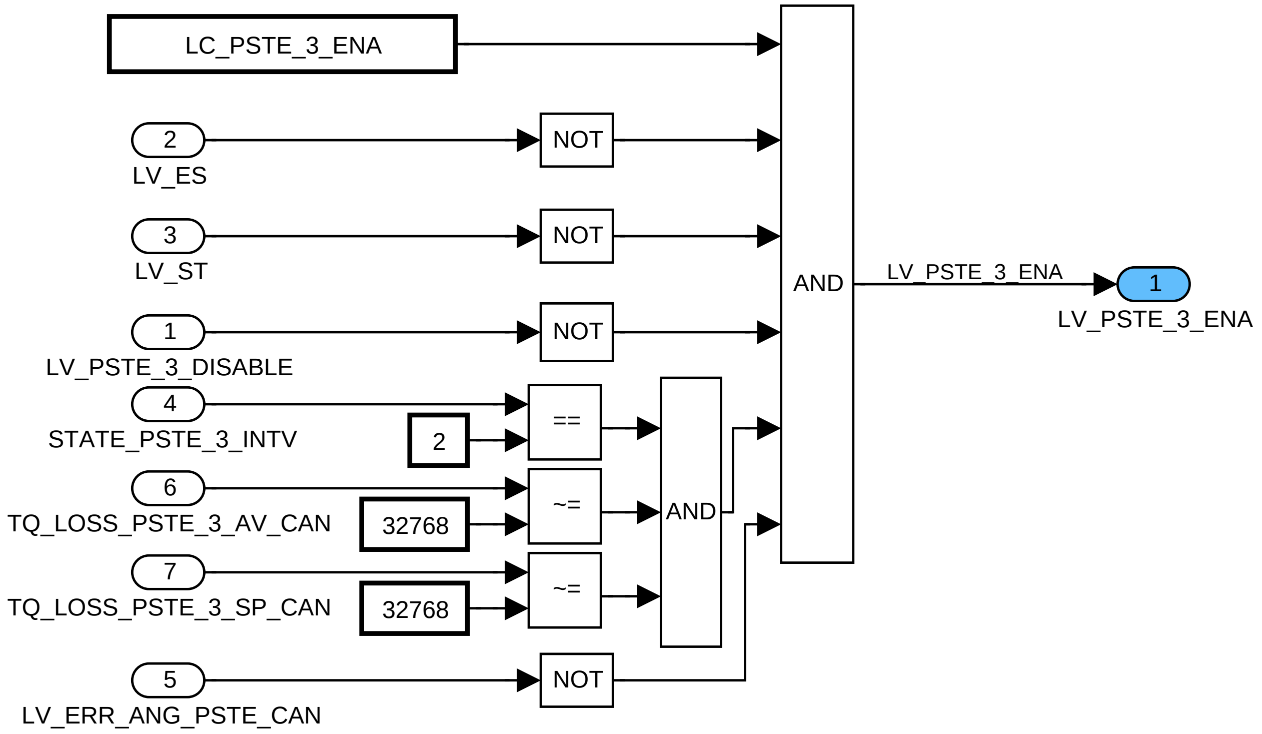


Figure 57:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/ENA_CDN_PSTE_INTV

58.3.4.2.3 Torque loss active PSTE_3 (EHB3)

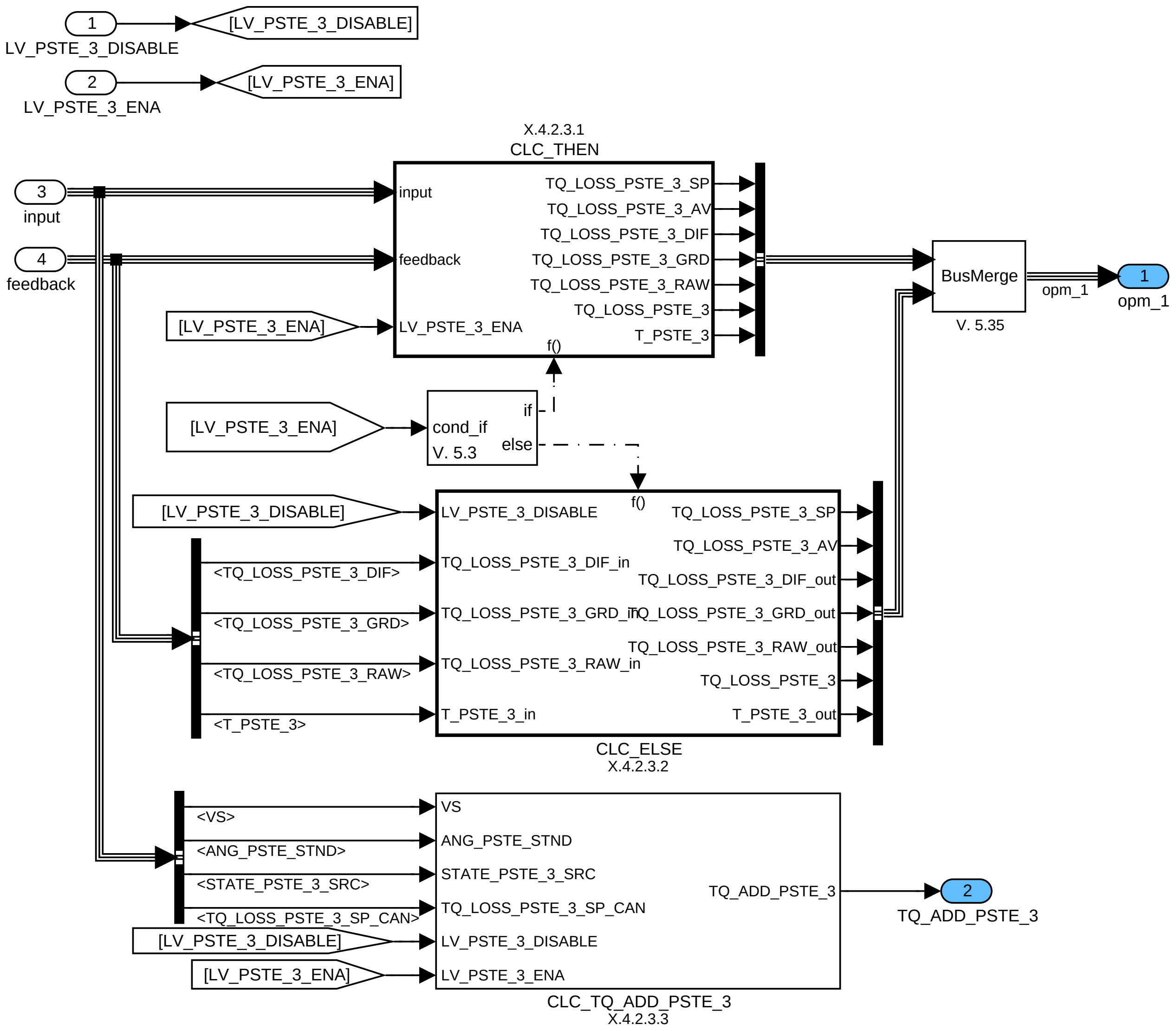


Figure 58:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR

58.3.4.2.3.1 Calculation when LV_PSTE_3_ENA is set.

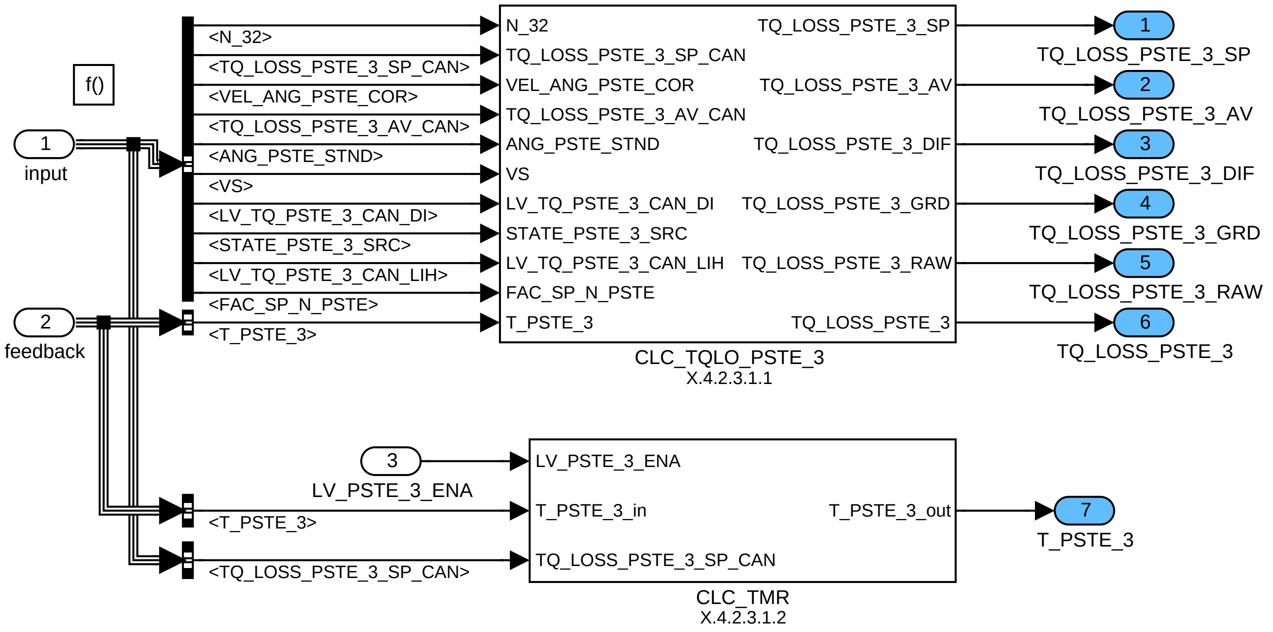


Figure 59:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN

58.3.4.2.3.1.1 Calculation part.

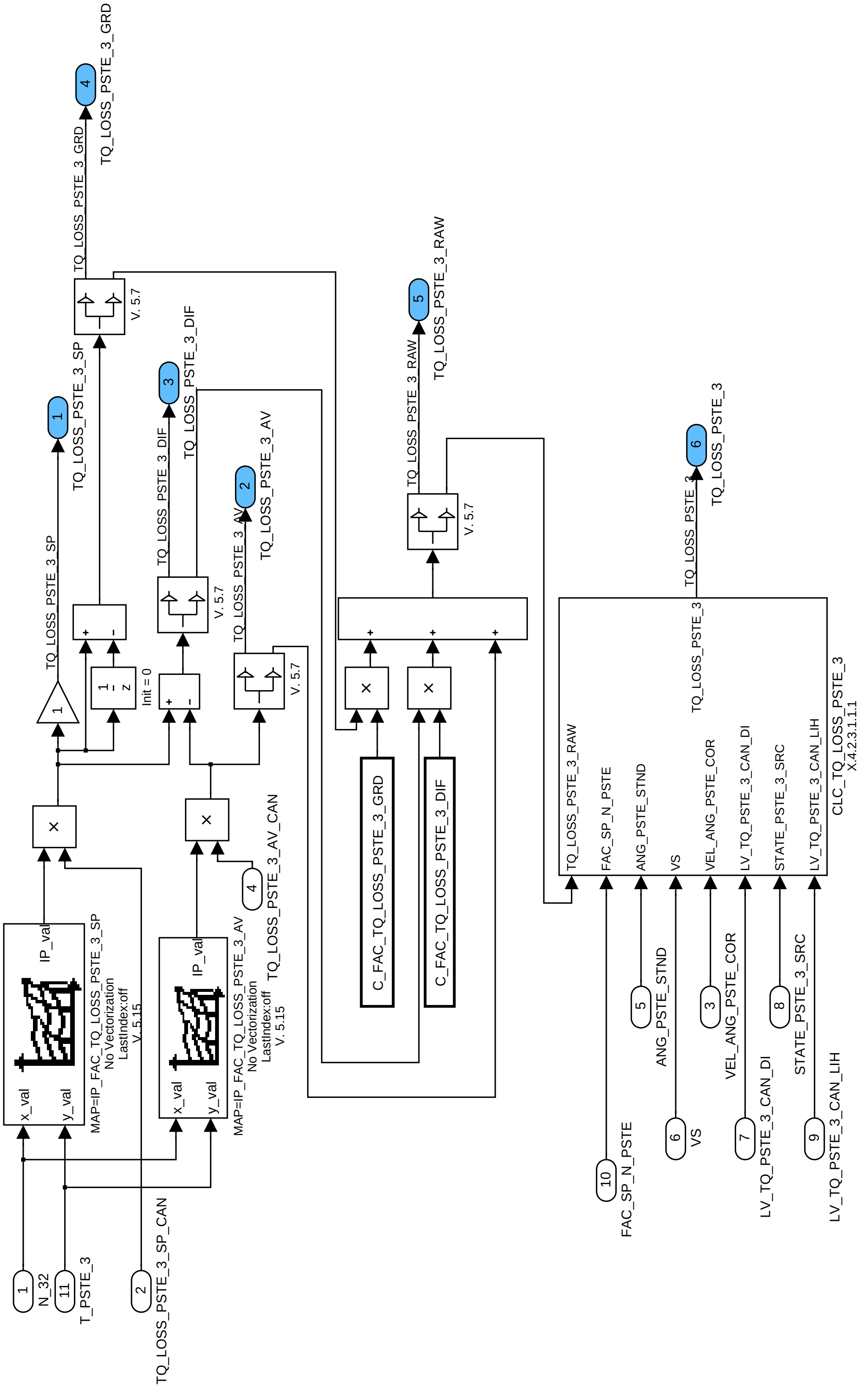


Figure 60:

Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN/CLC_TQLO_PSTE_3

58.3.4.2.3.1.1.1 Calculation of TQ_LOSS_PSTE_3

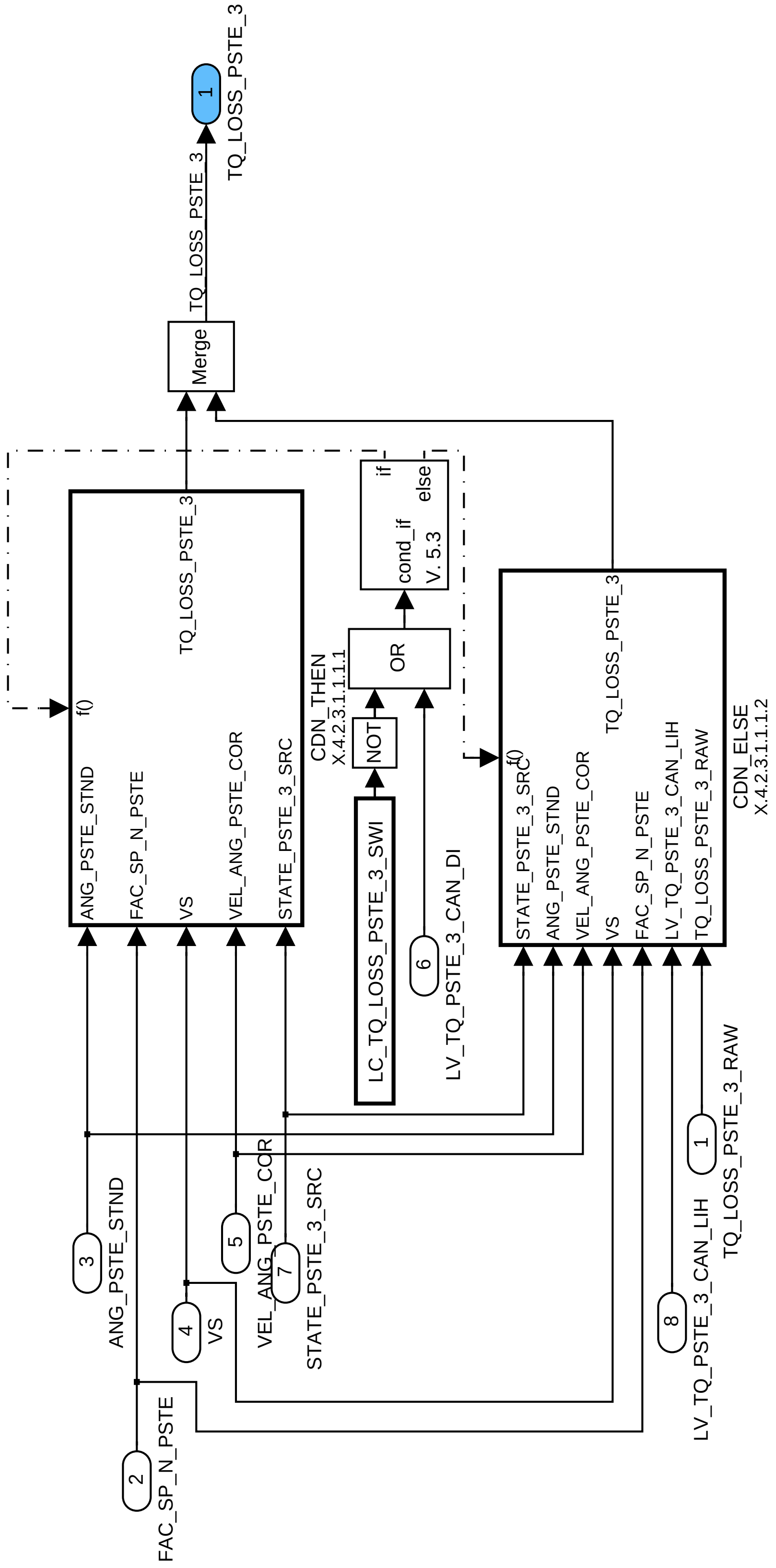


Figure 61:
Path:
STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN/CLC_TQLO_PSTE_3/CLC_TQ_LOSS_PSTE_3

58.3.4.2.3.1.1.1.1 Calculation of TQ_LOSS_PSTE_3 depending on the state variable.

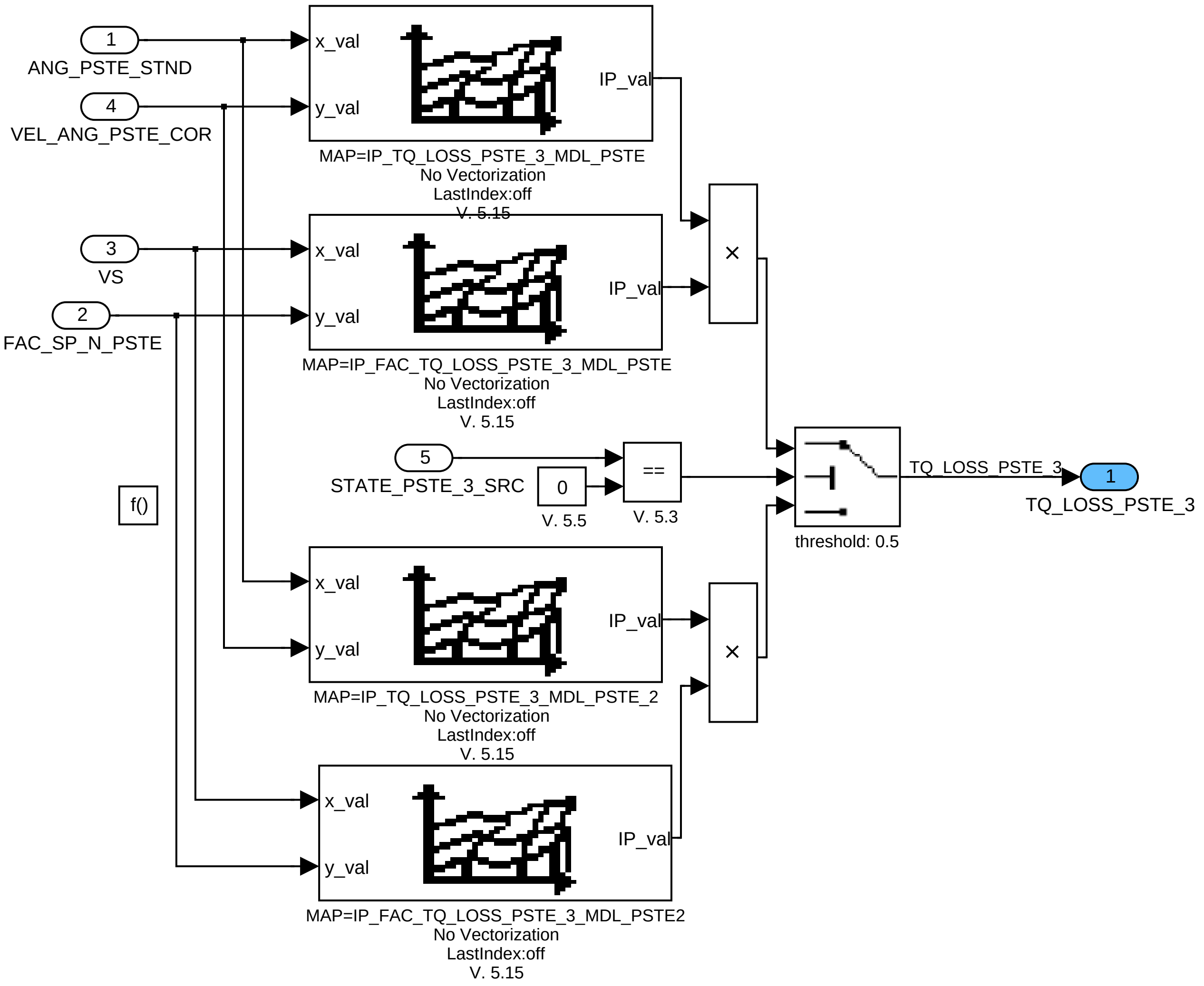


Figure 62:
Path:
STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN/CLC_TQLO_PSTE_3/CLC_TQ_LOSS_PSTE_3/CDN_THEN

58.3.4.2.3.1.1.1.2 Calculation of TQ_LOSS_PSTE_3 depending on LV_TQ_PSTE_3_CAN_LIH.

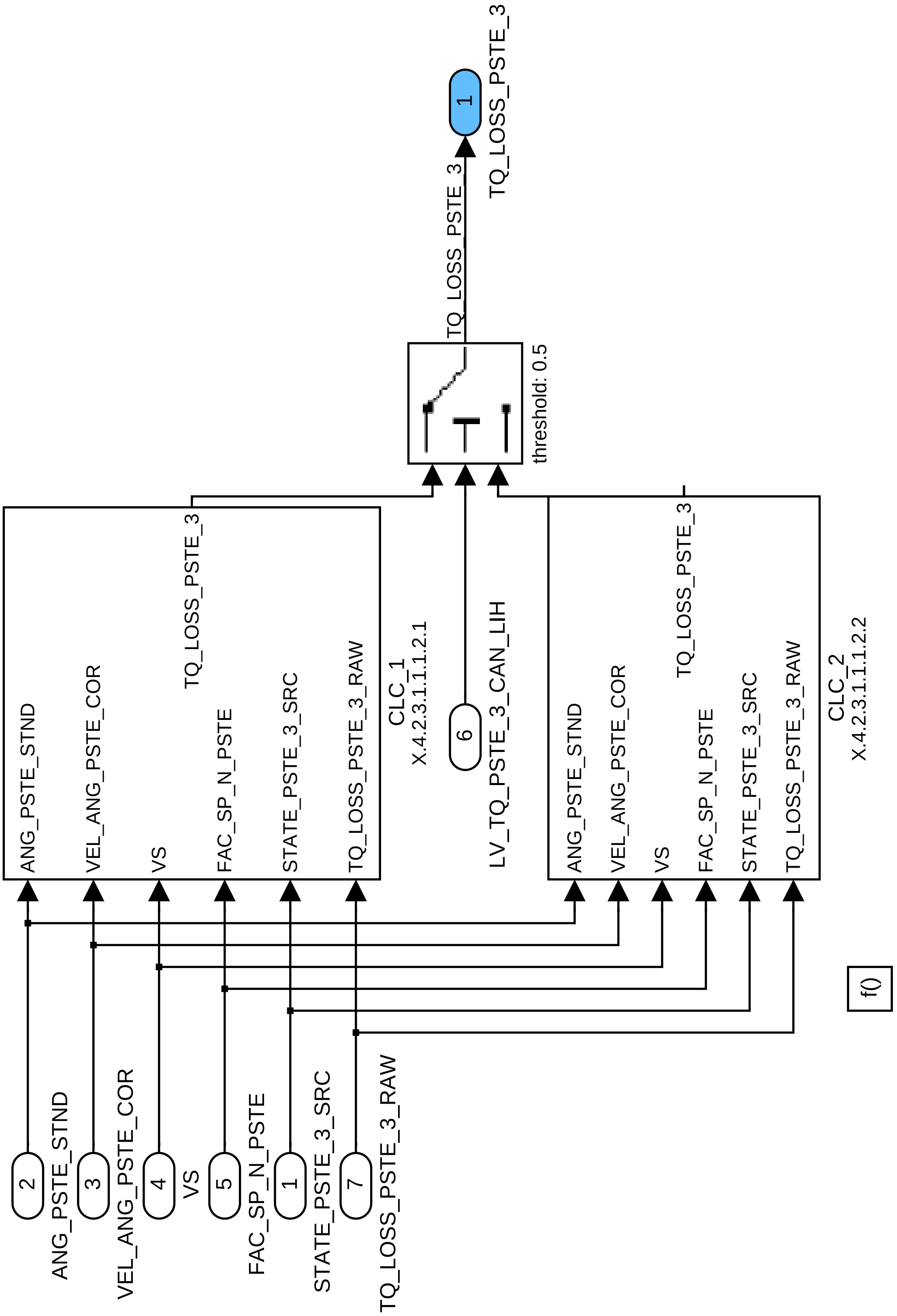


Figure 63:
Path:

STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN/CLC_TQLO_PSTE_3/CLC_TQ_LOSS_PSTE_3/CDN_ELSE

58.3.4.2.3.1.1.1.2.1 Calculation of TQ_LOSS_PSTE_3 depending on the state variable.

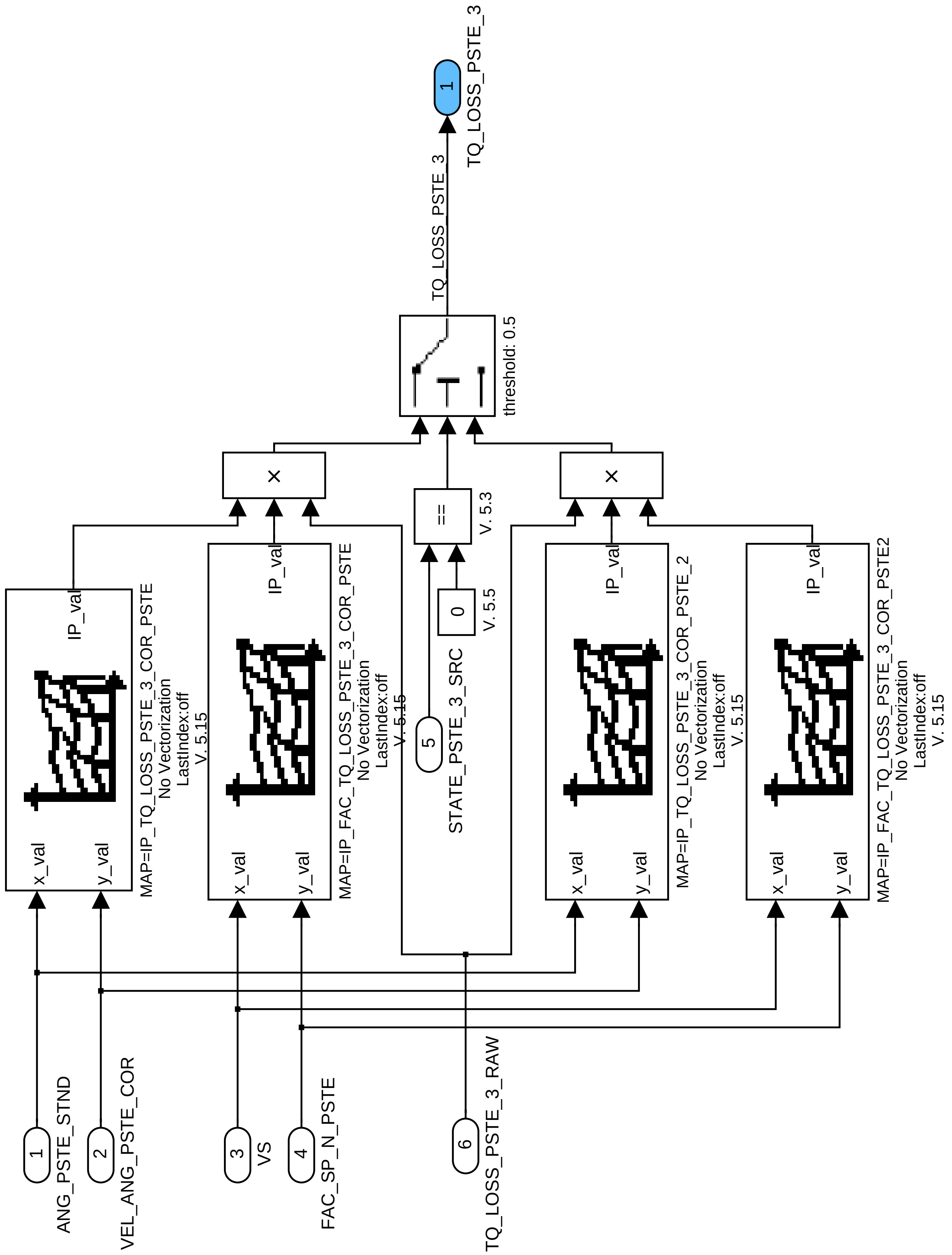


Figure 64:
Path:

STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN/CLC_TQLO_PSTE_3/CLC_TQ_LOSS_PSTE_3/CDN_ELSE/CLC_1

58.3.4.2.3.1.1.1.2.2 Calculation of TQ_LOSS_PSTE_3 depending on the state variable.

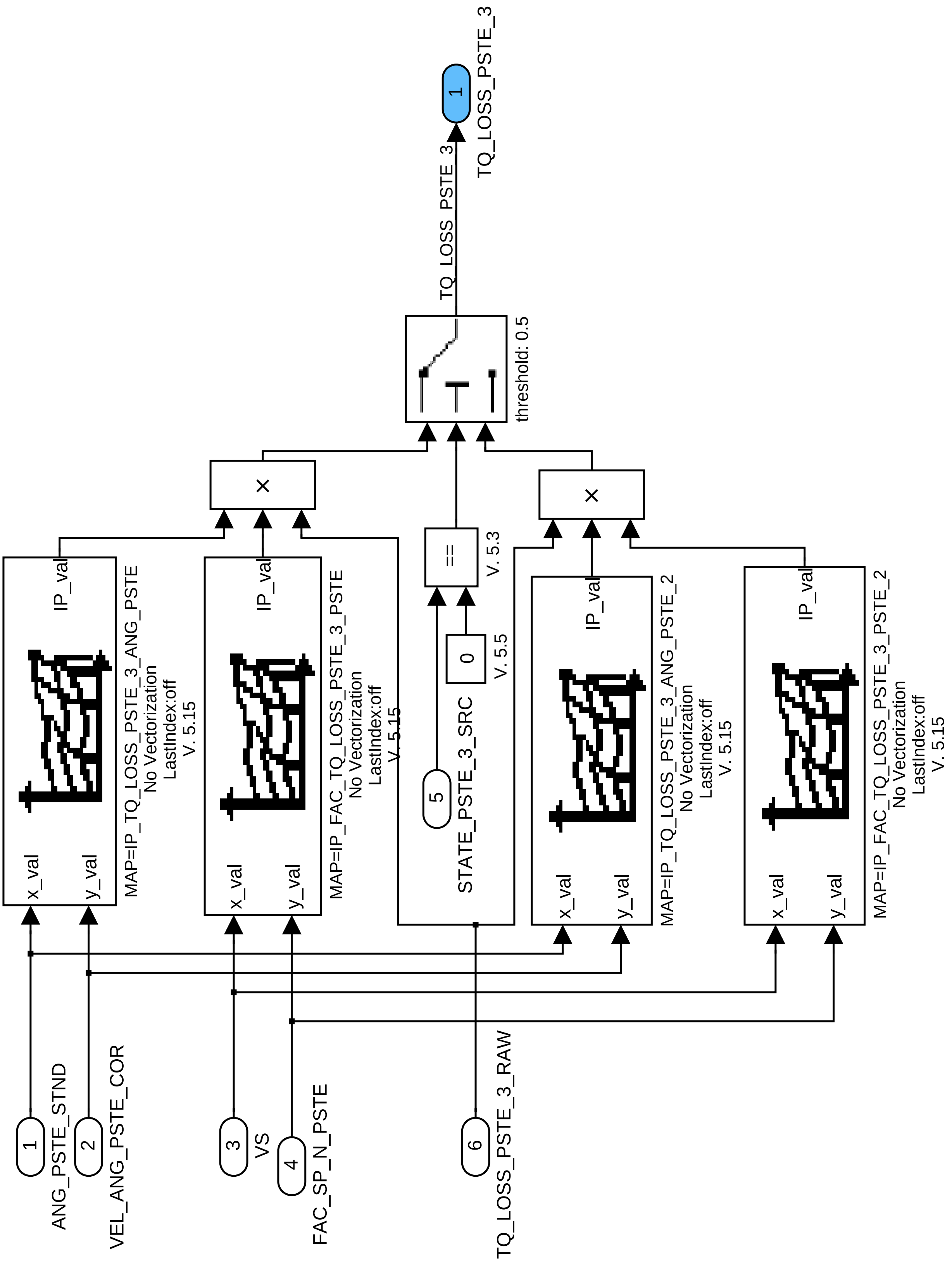


Figure 65:
Path:

STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN/CLC_TQLO_PSTE_3/CLC_TQ_LOSS_PSTE_3/CDN_ELSE/CLC_2

58.3.4.2.3.1.2 Calculation of timer.

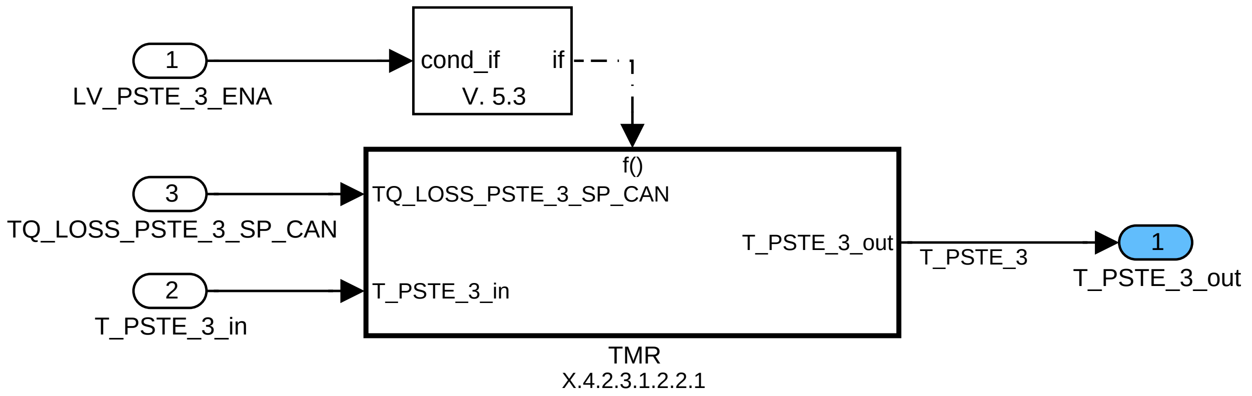


Figure 66:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN/CLC_TMR

58.3.4.2.3.1.2.1 Calculation of timer T_PSTE_3.

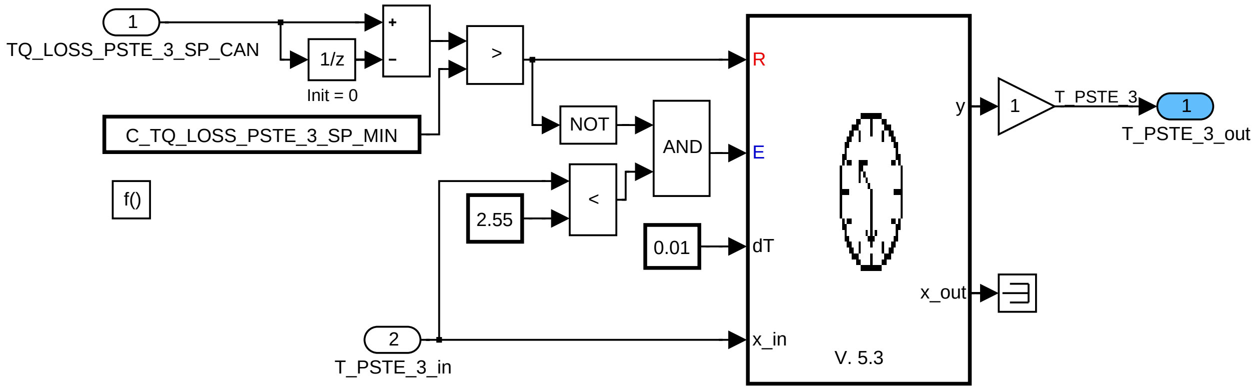
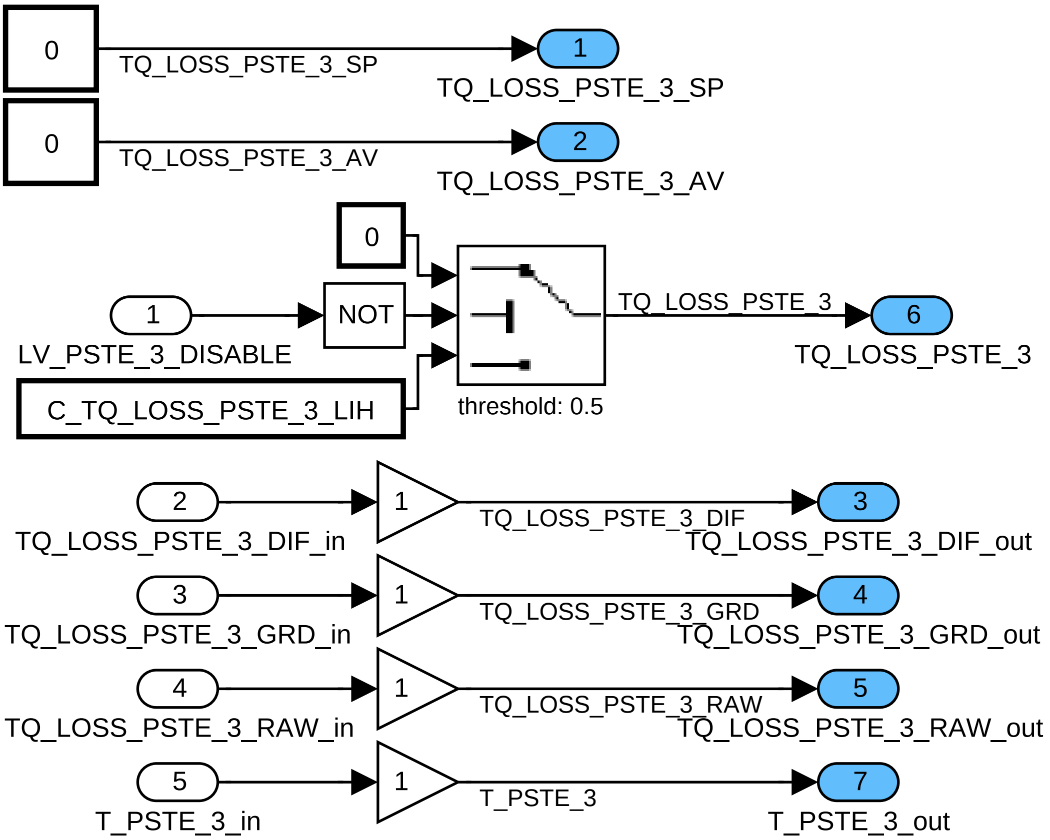


Figure 67:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_THEN/CLC_TMR/TMR

58.3.4.2.3.2 Calculation depending on LV_PSTE_3_ENA.



f()

Figure 68:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_ELSE

58.3.4.2.3.3 Calculation of TQ_ADD_PSTE_3

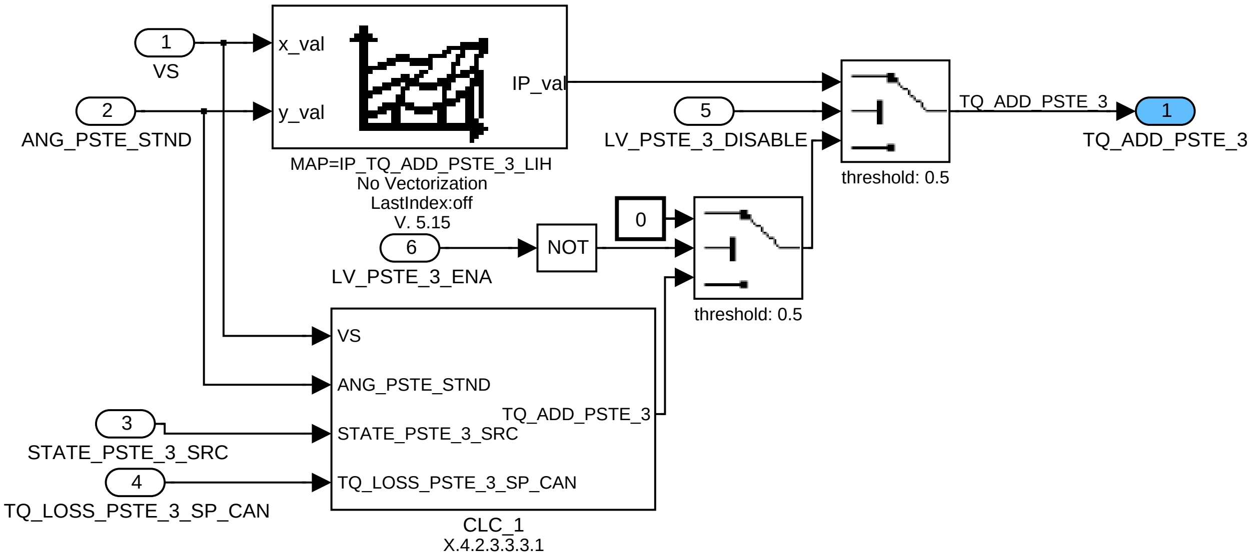


Figure 69:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_TQ_ADD_PSTE_3

58.3.4.2.3.3.1 Calculation of TQ_ADD_PSTE_3 depending on the state variable.

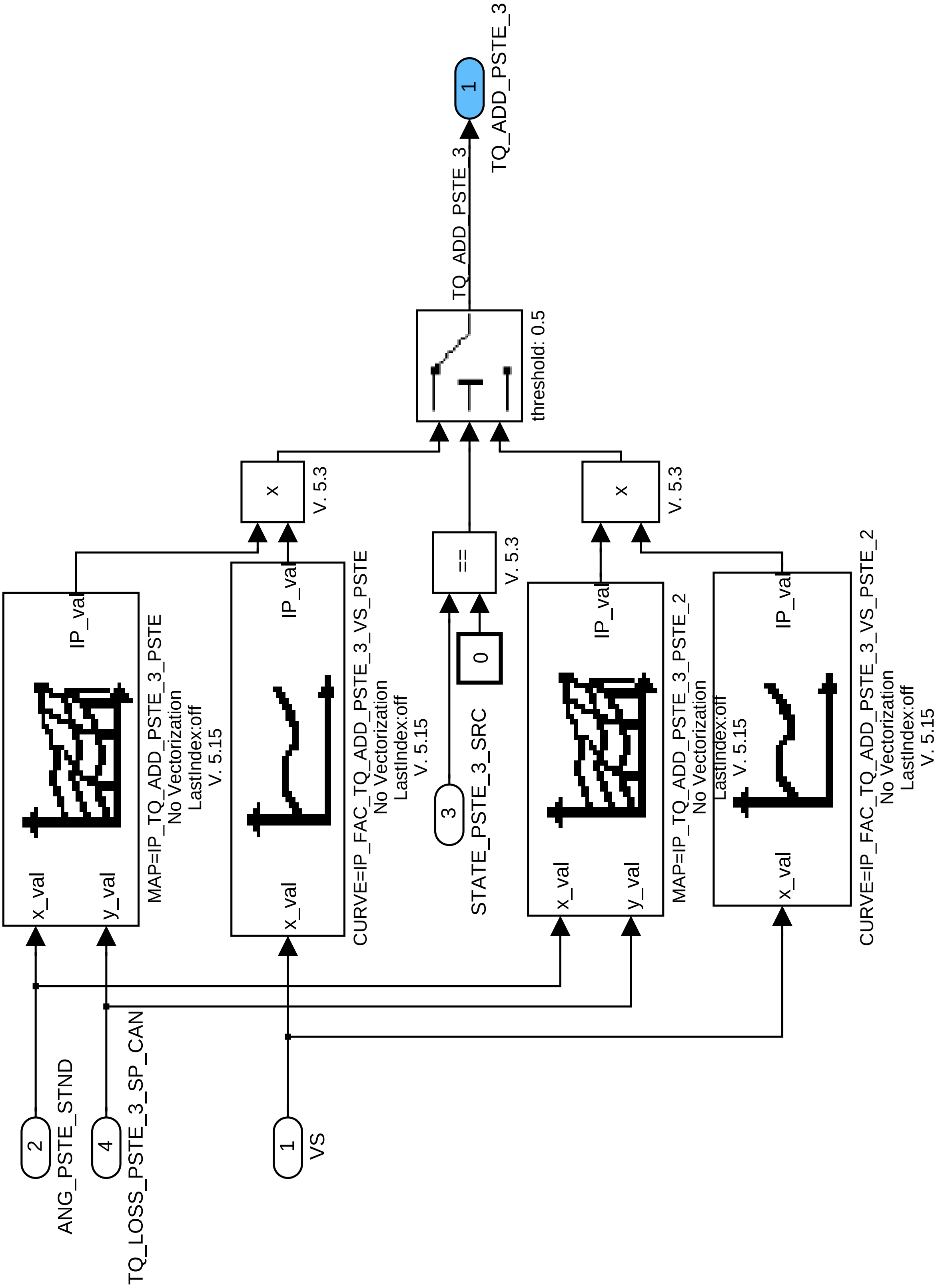


Figure 70:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_TQ_LOSS_ADD_TMR/CLC_TQ_ADD_PSTE_3/CLC_1

58.3.4.2.4 STATE_TQ_PSTE_3_PLAUS

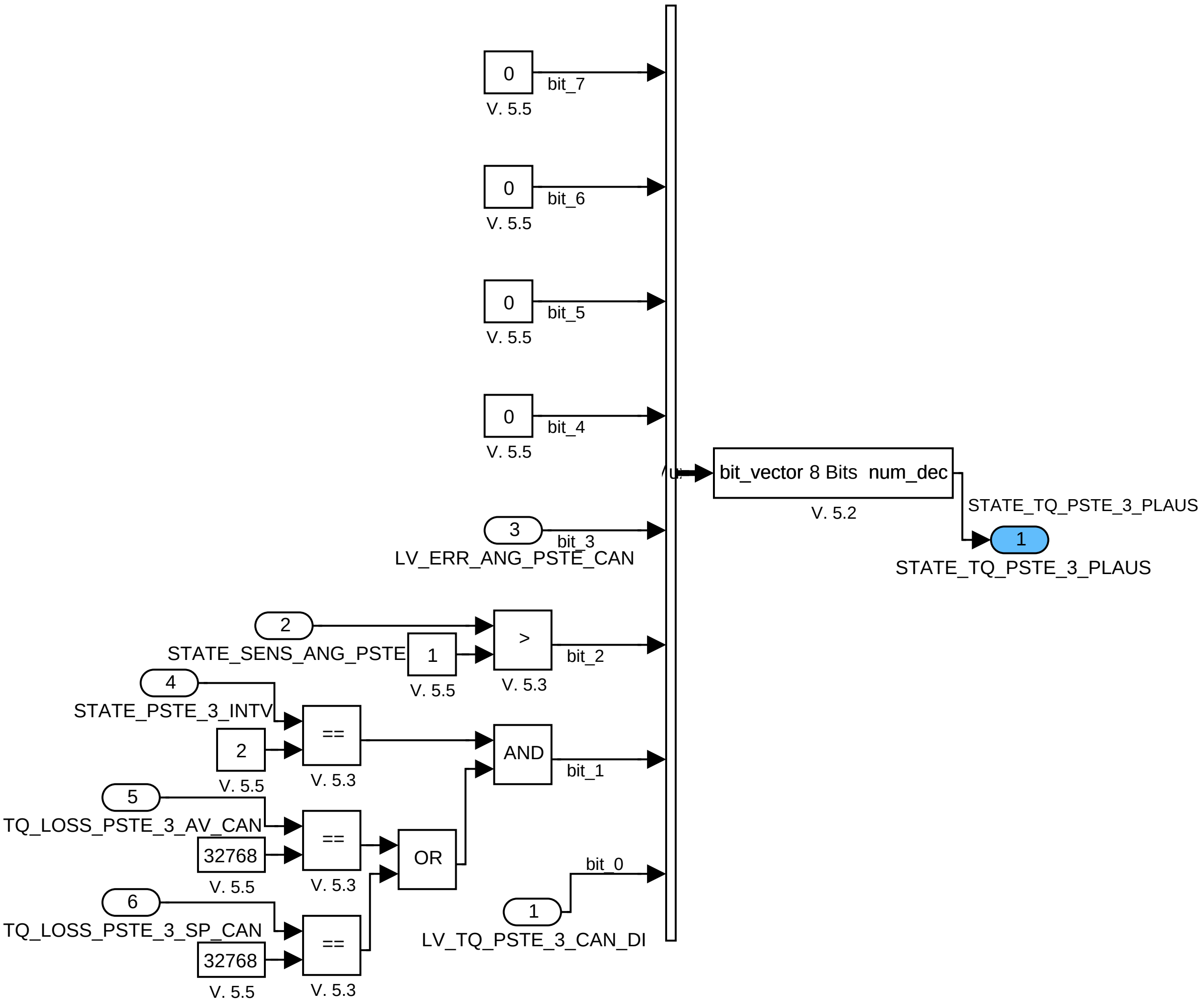


Figure 71:
Path: STSY_MD013/TQLO_ACT_PSTE_EHB3/OPM/CLC_STATE_TQ_PSTE_3_PLAUS

59 Suspension system

Table of Contents

Index

59.1 SUSY - Requirements to Infrastructure

Input data:

LV_SWI_AEB			

Export actions:

ACTION_INFR_SetAEB(IN <lv_swi_aeb>**)**
This action sets the AEB driver **ACTION_INFR_GetElDiagAEB(**OUT <Cdn_diag_aeb>, OUT <Err_diag_aeb>**)** This action reads the failure and condition information for a symptom of the AEB actuator power stage.

Description for actions:

ACTION_INFR_SetAEB (IN <lv_swi_aeb> **)**					
This action sets the AEB driver					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_swi_aeb	IN	0…1H	0…1	1	-
AEB driver pin level					

ACTION_INFR_GetElDiagAEB(OUT <Cdn_diag_aeb>, OUT <Err_diag_aeb>)					
This action reads the failure and condition information for a symptom of AEB power stage. The readout of the power stage is performed autonomous and the information is gathered. When calling this Action the information inside the Infrastructure is reset after returning the OUT parameters.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Cdn_diag_aeb	OUT	0 ... 7H	0 ... 7	1	-
Diagnosis condition for symptom: Bit 0: Set, if diagnosis condition for symptom SCP (SYM_0) is fulfilled Bit 1: Set, if diagnosis condition for symptom SCG (SYM_1) is fulfilled Bit 2: Set, if diagnosis condition for symptom OC (SYM_2) is fulfilled (all combinations are possible)					
Err_diag_aeb	OUT	0 ... 7H	0 ... 7	1	-
Raw value of error symptom: Bit 0: Set, if raw value of error symptom SCP (SYM_0) is set Bit 1: Set, if raw value of error symptom SCG (SYM_1) is set Bit 2: Set, if raw value of error symptom OC (SYM_2) is set (all combinations are possible)					
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

FUNCTION DESCRIPTION:

Requirements for ACTION_INFR_SetAEB:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

lv_swi_aeb	Not relevant	Not relevant	1	Not relevant	-

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

General information:

ACTION_INFR_GetElDiagAEB() returns the result of the electrical diagnosis of AEB actuator power stage.

_ The device readout is performed autonomous by the Infrastructure each 10ms.
_ The error informations of every symptom is gathered in the Infrastructure (or-ed symptom) until the Application reads out the information by calling ACTION_INFR_GetElDiagAEB().
_ After having read out the information by calling ACTION_INFR_GetElDiagAEB(), the data inside the Infrastructure is reset. Resetting of Cdn_diag_aeb avoids ambigious results in case of too short calling reccurrence of ACTION_INFR_GetElDiagAEB(): Reset Cdn_diag_aeb indicates, that the gathering of the information is not completely finished.

Requirements for ACTION_INFR_GetElDiagAEB:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cdn_diag_aeb	-	-	<bit coded>	Err_diag_aeb	Diagnosis condition for each symptom **bit 0:** diagnosis condition for symptom SCP (SYM_0) **bit 1:** diagnosis condition for symptom SCG (SYM_1) **bit 2:** diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
Err_diag_aeb	-	-	<bit coded>	Cdn_diag_aeb	Bitcoded result of each symptom **bit 0:** raw value of error symptom SCP (SYM_0) **bit 1:** raw value of error symptom SCG (SYM_1) **bit 2:** raw value of error symptom OC (SYM_2) The relevant bit is set, if the raw error has been detected.

Diagnosis: ACTION_INFR_GetElDiagAEB returns the electric diagnosis for AEB power stage

Coincidence requirements: no coincidence requirements to other events

59.2 Active engine bracket

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_SWI_AEB	O/V	0...1H	0...1	1	-
switch to control the active engine brackets					

Input data:

N_32	VS	OPM_AV	LV_SWI_AEB_ACT_EXT_ADJ
LV_SWI_AEB_EXT_ADJ	LV_STST_ST_REQ	LV_STST_PRE_STOP_REQ	LV_IGK
LV_IGK_PREL	LV_ST	LV_ES	LV_STST_ES
LV_VAR_AEB			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_32_SWI_AEB_HOM_AFL	1	0...FFH	0...8.16E+3	32	rpm
Engine speed threshold for switching active engine brackets in HOM_AFL mode					
C_N_32_SWI_AEB_HOM_AFS	1	0...FFH	0...8.16E+3	32	rpm
Engine speed threshold for switching active engine brackets in HOM_AFS mode					
C_N_32_SWI_AEB_HYS	1	0...FFH	0...8.16E+3	32	rpm
Engine speed hysteresis for switching active engine brackets					
C_N_32_SWI_AEB_S	1	0...FFH	0...8.16E+3	32	rpm
Engine speed threshold for switching active engine brackets in S mode					
C_VS_SWI_AEB_HOM_AFL	1	0...FFH	0...255	1	km/h
Vehicle speed threshold for switching active engine brackets in HOM_AFL mode					
C_VS_SWI_AEB_HOM_AFS	1	0...FFH	0...255	1	km/h
Vehicle speed threshold for switching active engine brackets in HOM_AFS mode					
C_VS_SWI_AEB_HYS	1	0...FFH	0...255	1	km/h
Vehicle speed hysteresis for switching active engine brackets					
C_VS_SWI_AEB_S	1	0...FFH	0...255	1	km/h
Vehicle speed threshold for switching active engine brackets in S mode					
LC_SWI_AEB_IGK_PREL_ENA	1	0...1H	0...1	1	-
Engine brackets stop position enabled to preliminary ignition shut off					
LC_SWI_AEB_ST	1	0...1H	0...1	1	-
Active engine brackets for engine start					
LC_SWI_AEB_STOP	1	0...1H	0...1	1	-
Active engine brackets for engine stop					
LC_SWI_AEB_STST_PRE_STOP_ENA	1	0...1H	0...1	1	-
Enable switch for pre-stop request					
LC_SWI_AEB_TYP	1	0...1H	0...1	1	-
Active engine brackets switch for different engine type					

59.2.1 MISC_M905G

The functionality “active engine bracket” supports the controllable engine suspension. To improve the driveability of the vehicle and to decrease the transmission of engine vibrations, it is possible to switch the engine bracket from the state “stiff” to the state “soft”.

To compensate the more rough combustion in stratified mode the engine brackets are switched into a soft characteristic.

Application Condition

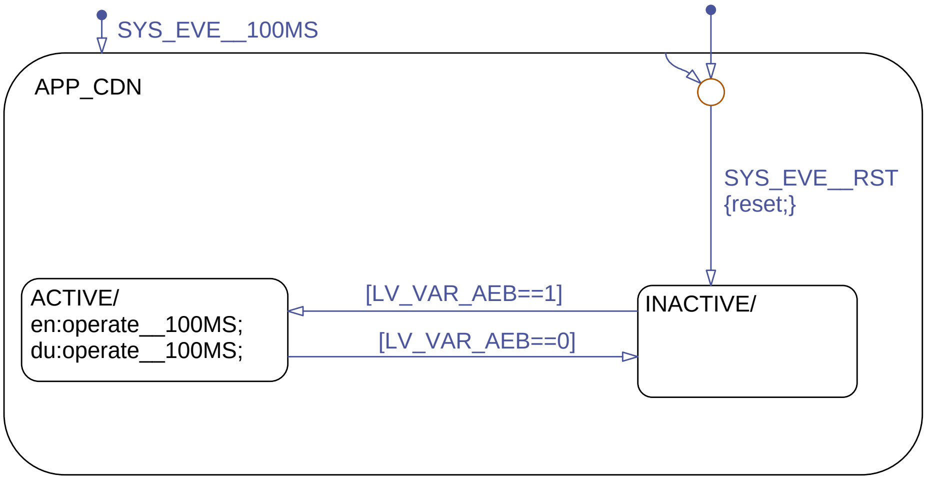
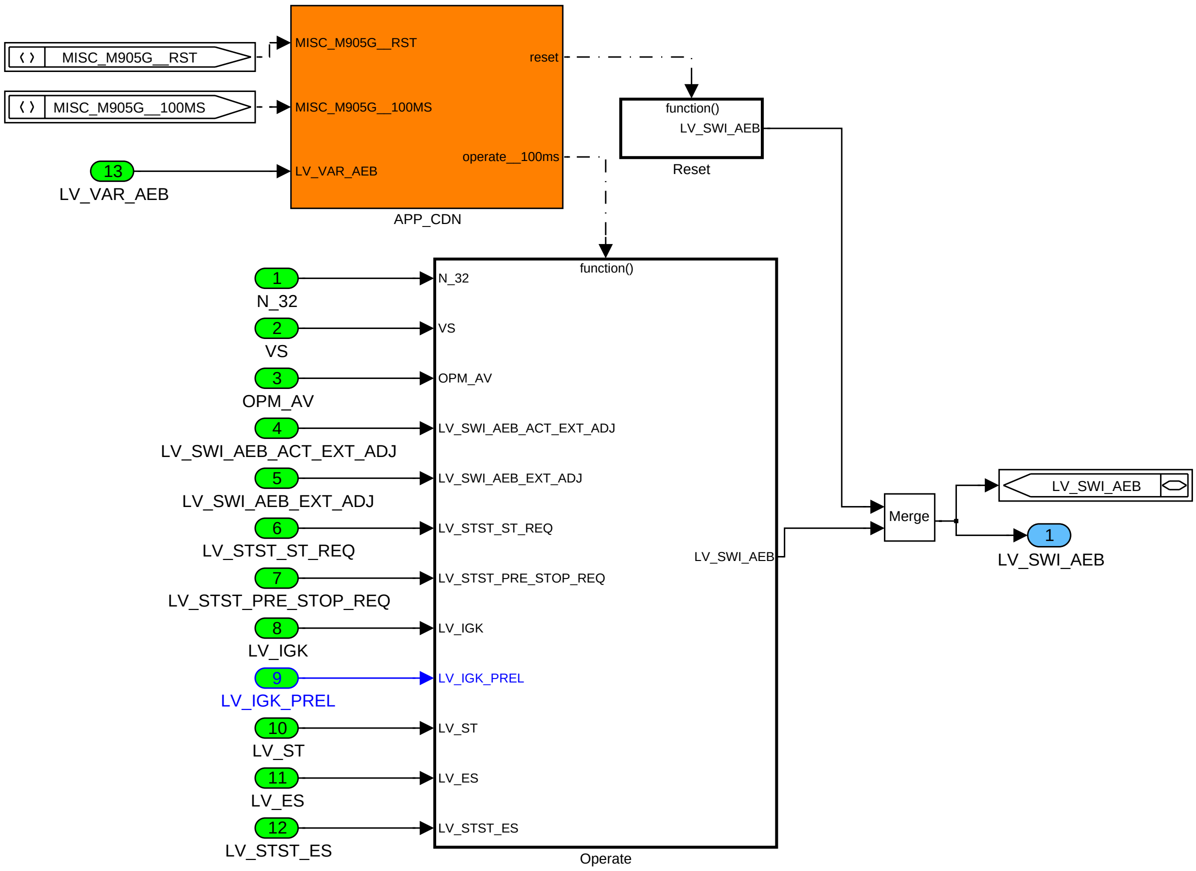


Figure 1 MISC_M905G/ APP_CDN/ Chart1

Function Description



SDA_SRS / SDA 4.0 30−May−2006

Figure 2 MISC_M905G

59.2.1.1 MISC_M905G/OPERATE

Dependent on engine speed and vehicle speed in each combustion mode, the logical variable LV_SWI_AEB is set to 1 and the corresponding ECU-Pin is set under current. In this case the engine bracket is switched to the state “soft”. Otherwise the "stiff" mode of the engine brackets is active. There is a hysteresis for engine speed and vehicle speed (common for all combustion modes) to avoid toggling of LV_SWI_AEB. The logical constant LC_SWI_AEB_TYP is introduced to take the different engine bracket types into account, which are provided from different suppliers.

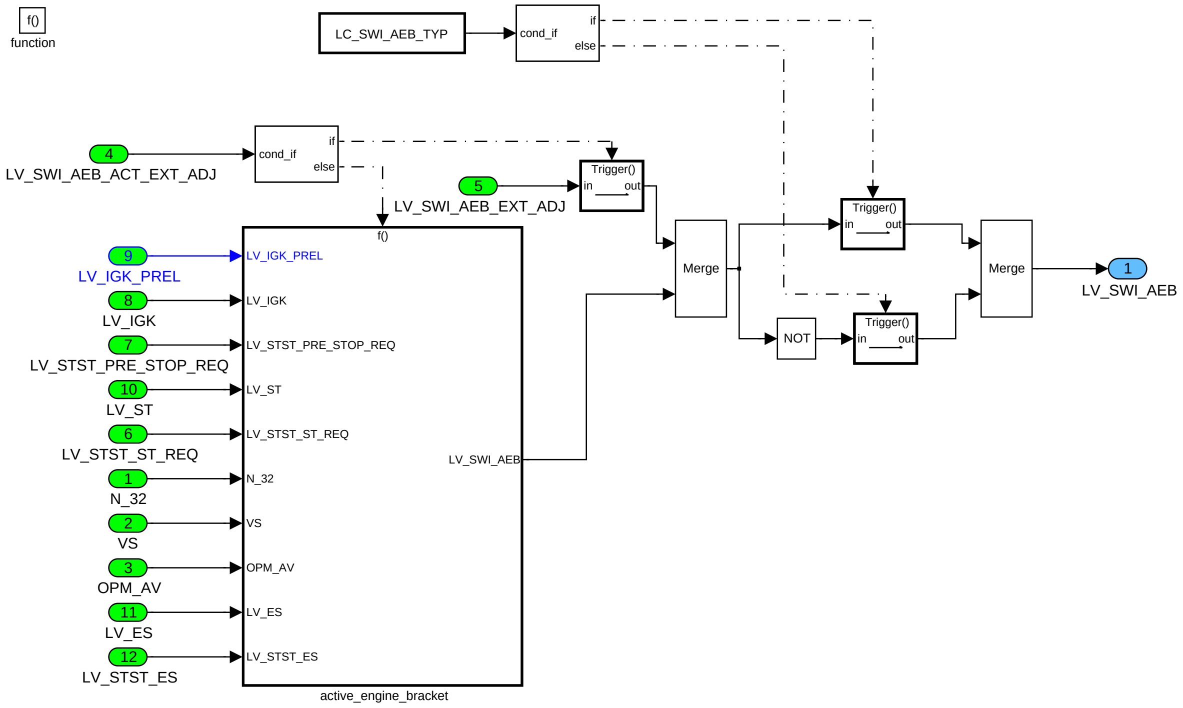


Figure 3 MISC_M905G/ Operate

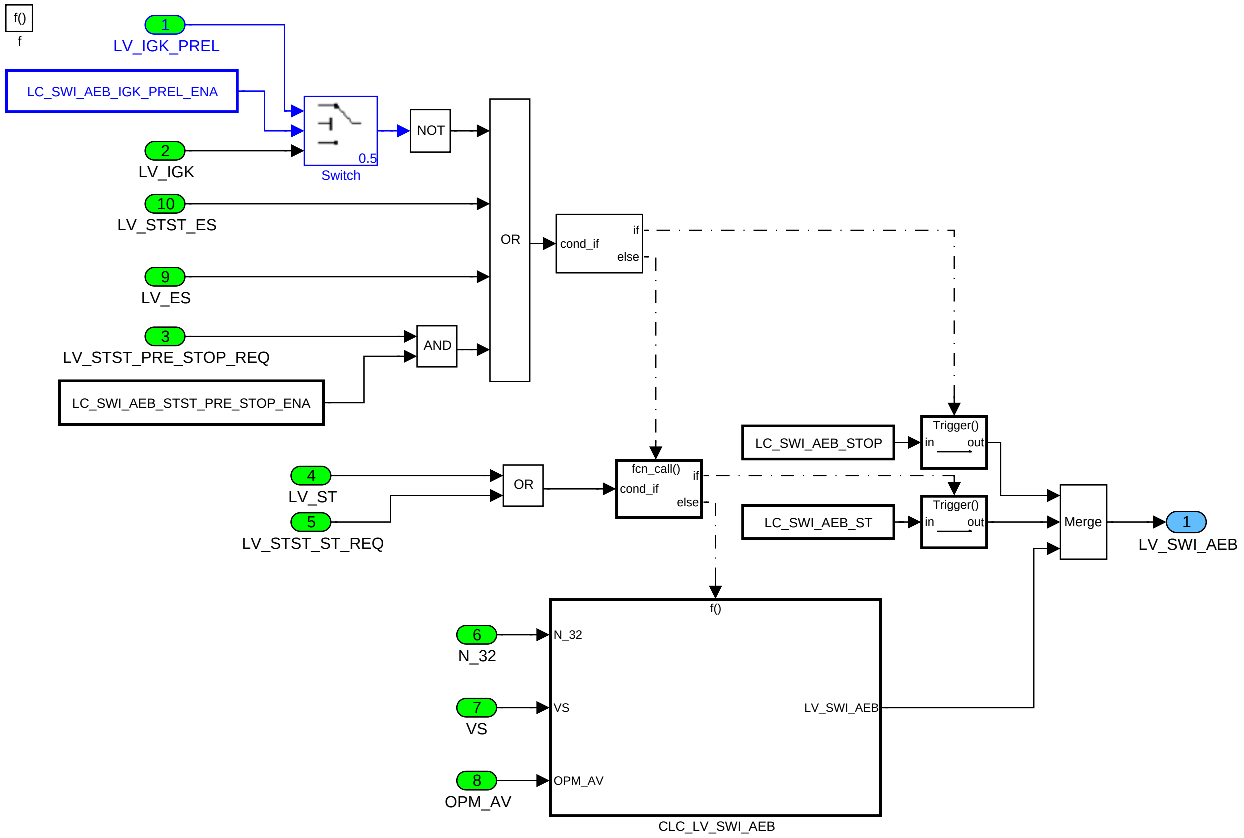


Figure 4 MISC_M905G/ Operate/ active_engine_bracket

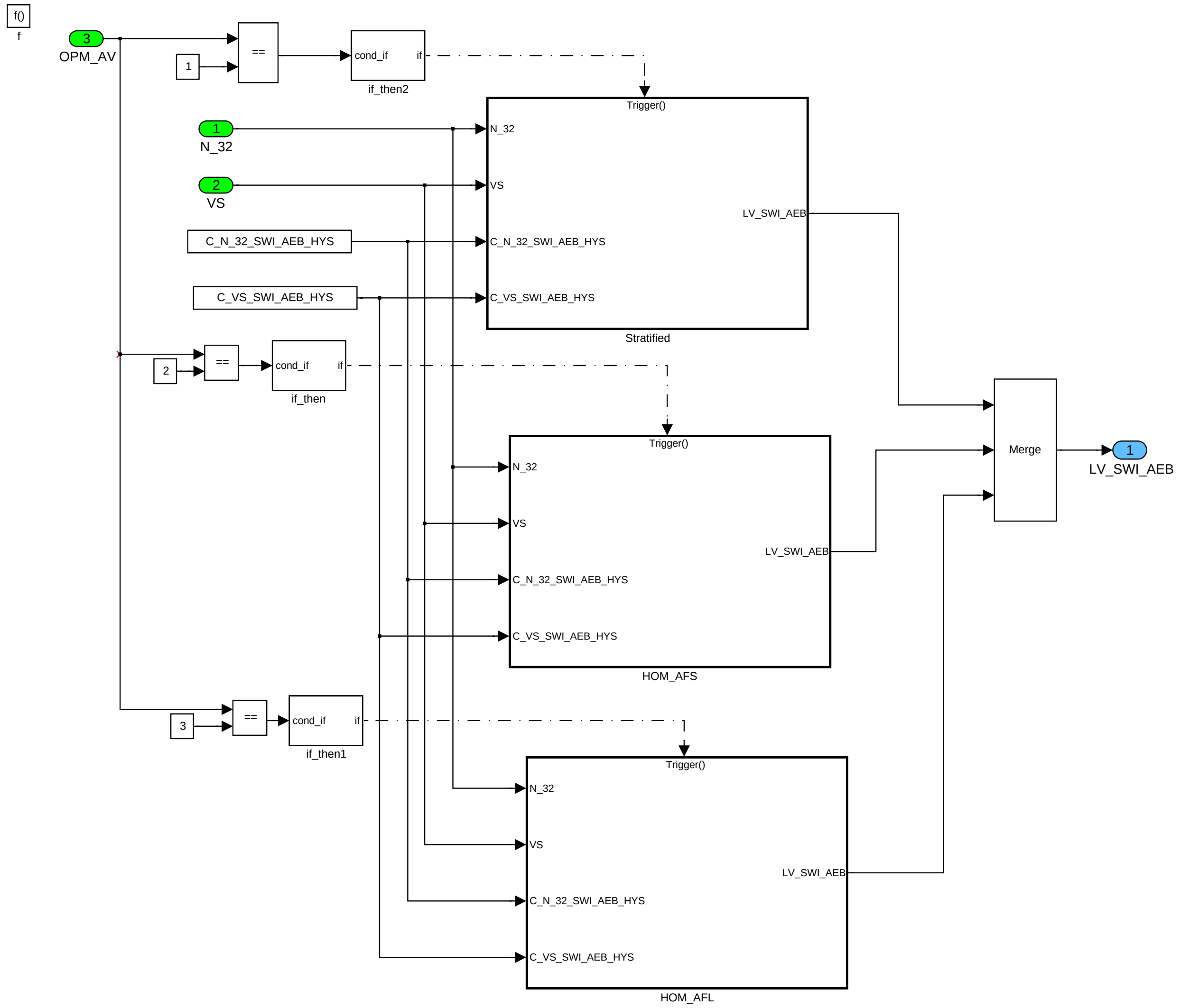


Figure 5 MISC_M905G/ Operate/ active_engine_bracket/ CLC_LV_SWI_AEB

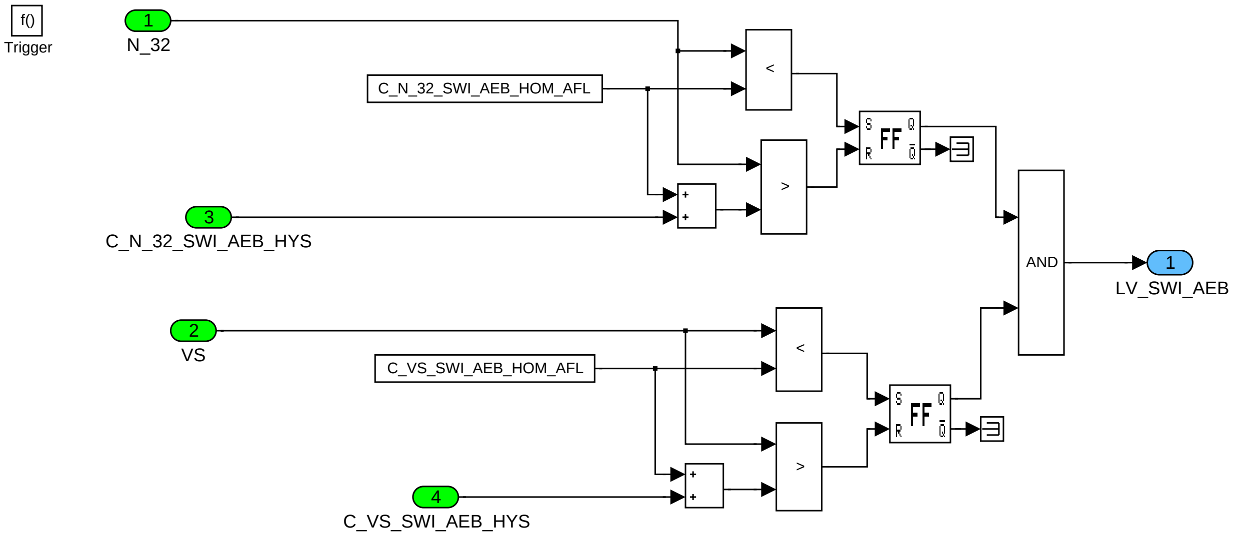


Figure 6 MISC_M905G/ Operate/ active_engine_bracket/ CLC_LV_SWI_AEB/ HOM_AFL

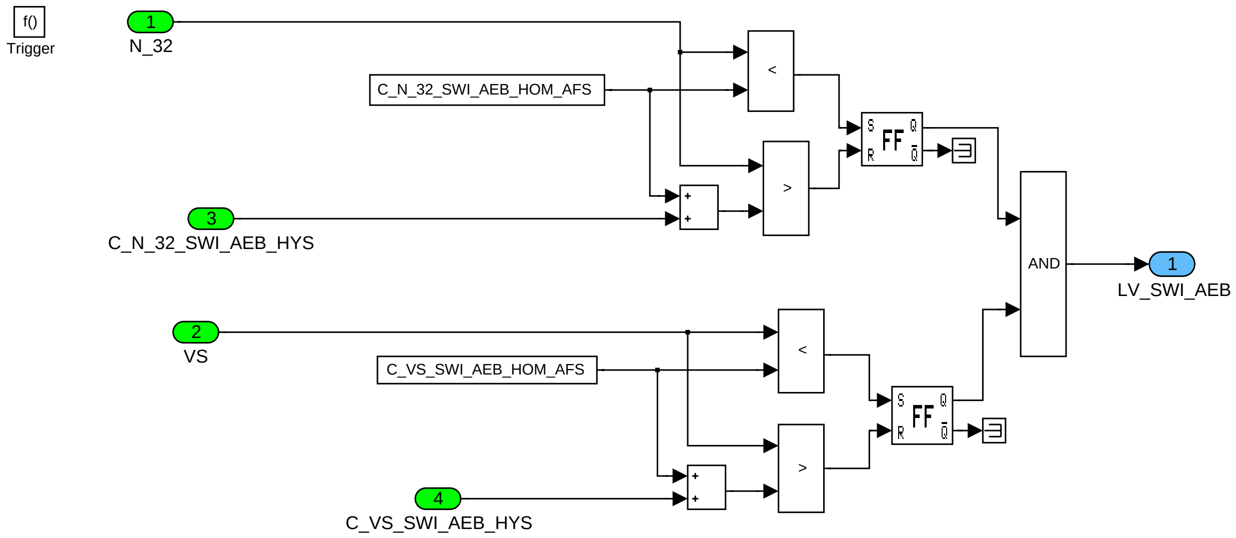


Figure 7 MISC_M905G/ Operate/ active_engine_bracket/ CLC_LV_SWI_AEB/ HOM_AFS

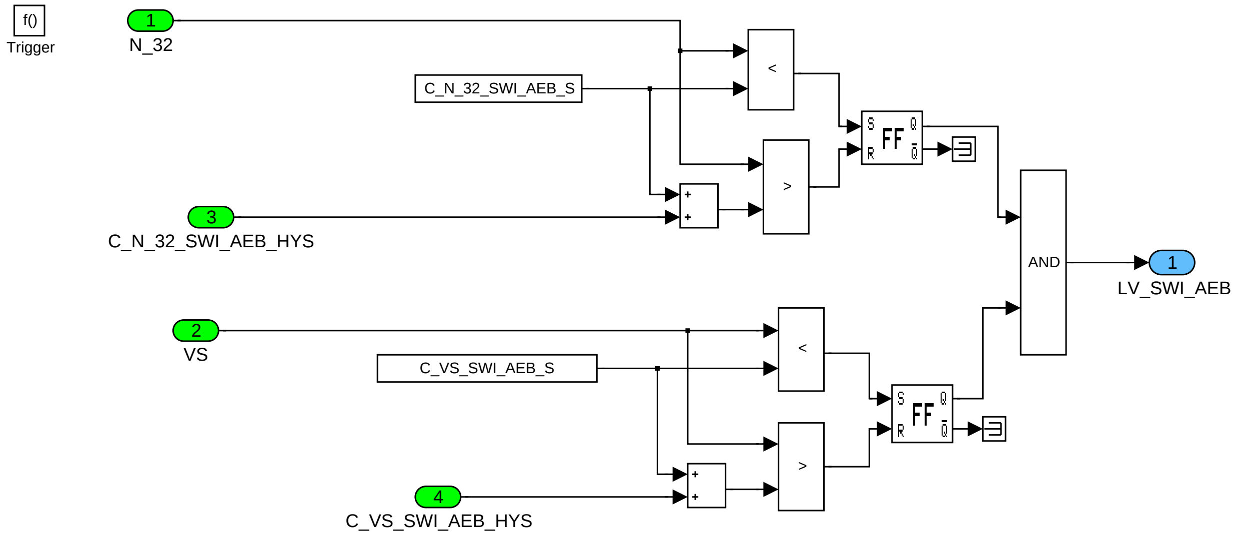


Figure 8 MISC_M905G/ Operate/ active_engine_bracket/ CLC_LV_SWI_AEB/ Stratified

59.2.1.2 SUBFUNCTION: Reset

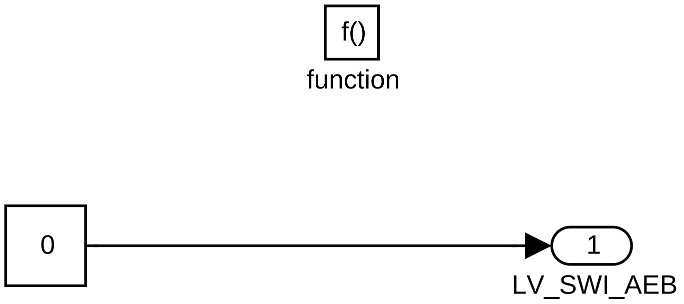


Figure 9 MISC_M905G/ Reset

59.3 Active engine brackets electrical diagnosis (AEB)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_AEB[NC_AEB_NR]	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on AEB[i] power stage					
LV_CDN_DIAG_AEB[NC_AEB_NR]	V/O	0...1H	0...1	1	[-]
Diagnosis condition AEB power stage diagnosis					
ERR_SYM_AEB[NC_AEB_NR]	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom AEB power stage diagnosis					
LV_END_DIAG_AEB[NC_AEB_NR]	V/O	0...1H	0...1	1	[-]
End of diagnosis AEB power stage diagnosis					
CDN_DIAG_AEB[NC_AEB_NR]	V	0...7H	0...7	1	[-]
Diagnosis condition AEB for each symptom					
ERR_DIAG_AEB[NC_AEB_NR]	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for AEB (only parameter)					

Input data:

LV_IGK	LV_VB_CDN_OBD_1	LV_ERR_SPI_MPS	LV_VAR_AEB

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN<CDN_DIAG_XX >, IN<ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>, OUT< LV_ERR_XX>)
ACTION_INFR_GetElDiagAEB (OUT< CDN_DIAG_XX>, OUT< ERR_DIAG_XX >)

FUNCTION DESCRIPTION:

General information:

The AEB is driven by the ECU via an output driver. The failure detection is done by ECU Hardware. The purpose of the diagnosis is to detect electrical faults as defined by OBD I requirements. The AEB can be switched on or off (no PWM-signal).

Description:

The purpose is to perform the electrical diagnosis of the AEB actuator. 3 symptoms are distinguished:

Error-symptoms and conditions: are defined to this diagnosis function as following

```
□□□□
 | | | └─> Short circuit to battery (SCP)     (= SYM_0)
 | | └───> Short circuit to ground (SCG)      (= SYM_1)
 | └─────> Open circuit (OC)                  (= SYM_2)
 └───────> Not used                           (= SYM_3)
```

Application conditions:

Initialisation: ERRM variables are initialized according Filter-type.
At Reset: CDN_DIAG_AEB = 0

Recurrence: 100ms

Activation: LV_IGK = 1 **and** LV_VAR_AEB = 1

Deactivation: LV_IGK = 0 **or** LV_VAR_AEB = 0

Formula section:

```
If    LV_ERR_SPI_MPS = 0
      and   LV_CDN_VB_OBD_1 = 1
      and   LV_IGK = 1

Then

      Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_AEB and
      diagnosis conditions CDN_DIAG_AEB) received from the infrastructure:

      ACTION_INFR_GetElDiagAEB (OUT<CDN_DIAG_AEB>, OUT<ERR_DIAG_AEB>)

      Basic diagnosis conditions are set according infrastructure information:
      CDN_DIAG_AEB
      Failure symptoms (raw value) are set according infrastructure information:
      ERR_DIAG_AEB

      If    Activation conditions are met for the NC_PSD_DLY_AEB recurrence
      Then
            { No additional diagnosis conditions are necessary }
      Else
            CDN_DIAG_AEB = 0
      Endif
Else
            CDN_DIAG_AEB = 0
Endif
```

Failure filtering and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_AEB and ERR_DIAG_AEB.

ACTION_ERRM_FilterMulticondition (IN<*AEB*>, IN<CDN_DIAG_AEB>,
IN<ERR_DIAG_AEB>, IN<C_ABC_INC_AEB>, IN<C_ABC_MAX_AEB>,
OUT< LV_ERR_AEB>)

This algorithm determines:

ERR_SYM_AEB	(detected error symptom for AEB diagnosis)
LV_ERR_AEB	(Error flag for debounced error of AEB)
LV_CDN_DIAG_AEB	(Diagnosis condition information)
LV_END_DIAG_AEB	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_AEB	1	0...FFH	0...255	1	[-]
Debounce counter increment AEB[i] power stage diagnosis					
C_ABC_MAX_AEB	1	1...FFH	1...255	1	[-]
Debounce counter maximum value AEB[i] power stage diagnosis					

Configuration for diagnostic symptoms :

Diagnostic AEB	Symptom description	Symptom	Filter type
Active engine bracket Diagnostic	*SCP*	*SYM_0*	*MPL_STD_INI*
	SCG	*SYM_1*	
	OC	*SYM_2*	

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_AEB	-	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (= 2)					
NC_AEB_NR		0...2H	0...2	1	[-]
Number of active engine brackets					

59.4 Torque Loss – Anti-Roll-Stabilisation

FUNCTION DESCRIPTION:

General information:

TQ_LOSS_ARS represents the engine torque loss due to power consumption by the ARS control unit during anti-roll-stabilisation intervention.

59.4.1 Enable conditions

59.4.1.1 Inhibitation ARS intervention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ARS_DISABLE	V/O	0...1H	0...1	1	[-]
logical variable for disabling ARS intervention due to EMS error					
LV_ARS_DISABLE_CAN	V/O	0...1H	0...1	1	[-]
logical variable for disabling ARS intervention due to CAN error					

Input data:

LV_VAR_BN	LV_IGK		LV_TCS_DISABLE_CAN
LV_ERR_TQ_LOSS_ARS_SP_CAN	LV_ERR_TQ_LOSS_ARS_AV_CAN	LV_ERR_CAN_BOFF	LV_ERR_BN_ARS
LV_TQI_REQ_CAN_INH	LV_DI_TQ_REQ_CAN_MPI_GDI	LV_ERR_BN_VS_TCS	LV_CDN_VB_CAN_TQ_DIAG

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: LV_ARS_DISABLE =0 , LV_ARS_DISABLE_CAN = 0 **at reset**

Update Rate: 10 ms

Activation: LV_VAR_BN = 1 **and** LV_IGK = 1

Deactivation: -

Formula section:

CAN environment error ARS inhibition:

If LV_ERR_CAN_BOFF = 1 **or**
LV_ERR_BN_ARS = 1 **or**
LV_CDN_VB_CAN_TQ_DIAG = 0
Then LV_ARS_DISABLE_CAN = 1
Else LV_ARS_DISABLE_CAN = 0
Endif

General ARS inhibition:

```
If      LV_TQI_REQ_CAN_INH      = 1          or
        LV_DI_TQ_REQ_CAN_MPI_GDI = 1         or
        LV_ARS_DISABLE_CAN      = 1          or
        LV_TCS_DISABLE_CAN      = 1          or
        LV_ERR_TQ_LOSS_ARS_SP_CAN = 1        or
        LV_ERR_TQ_LOSS_ARS_AV_CAN = 1
Then    LV_ARS_DISABLE = 1
Else    LV_ARS_DISABLE = 0
Endif
```

59.4.1.2 Enable conditions ARS intervention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ARS_ENA	V/O	0...1H	0...1	1	[-]
logical variable for ARS intervention enabling					
STATE_TQ_ARS_PLAUS	V/O	0...FFH	0...255	1	[-]
Bitwise coded State for ARS intervention state					

Input data:

LV_VAR_ARS	LV_ES	LV_ST	LV_IGK
LV_ERR_TQ_LOSS_ARS_AV_CAN	LV_ERR_TQ_LOSS_ARS_SP_CAN	STATE_ARS_CAN	LV_VAR_BN

FUNCTION DESCRIPTION:

The State STATE_TQ_ARS_PLAUS indicates the interrupted condition which is responsible for enable Bit = 0. The [x] Value shows the HEX adress inside the Bitwise coded State.

Application conditions:

Initialisation: LV_ARS_ENA = 0 **at reset**

Update Rate: 10 ms

Activation: LV_VAR_ARS **and** LV_IGK = 1

Deactivation: -

Formula section:

Enable conditions for ARS intervention:

If LV_ES = 0 **and**
LV_ST = 0 **and**
LV_ARS_DISABLE = 0 **and**
LV_TQ_MAX_ARS = 0 **and**
STATE_ARS_CAN = 01H
Then LV_ARS_ENA = 1
Else LV_ARS_ENA = 0

STATE_TQ_ARS_PLAUS:

If LV_ERR_TQ_LOSS_ARS_SP_CAN = 1 **or** BIT [00] set
LV_ERR_TQ_LOSS_ARS_AV_CAN = 1 BIT [01] set
If LV_TQ_MAX_ARS = 1 BIT [02] set
Then If STATE_ARS_CAN = 01H BIT [03] set

59.4.2 Maximum Torque Request

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TQ_MAX_ARS	V/O	0...1H	0...1	1	[-]
logical variable indicating maximum torque request - ARS					
TQ_LOSS_ARS_SP_MAX_1	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
ARS target torque loss - maximum					
TQ_LOSS_ARS_SP_MAX	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
ARS target torque loss - maximum (considering offset)					
TQ_LOSS_ARS_AV_MAX_1	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
ARS actual value of torque loss - maximum					
TQ_LOSS_ARS_AV_MAX	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
ARS actual value of torque loss - maximum (considering offset)					
T_DLY_TQ_MAX_ARS	V/O	0...FFFFH	0...655.35	0.01	[s]
Delay time for activation of LV_TQ_MAX_ARS					

Input data:

N_32	LV_IGK	VS_FIL	LV_VAR_BN
TQ_LOSS_ARS_AV_CAN	TQ_LOSS_ARS_SP_CAN	AC_VEH_TRV_TCS	

FUNCTION DESCRIPTION:

If either:

- the target torque loss via CAN is bigger than the minimum selection of maximum target torque loss (depending on VS_FIL, AC_VEH_TRV_TCS) and 100 Nm

or:

- the actual value of torque loss via CAN is bigger than the minimum selection of the actual value (maximum) of torque loss (depending on VS_FIL, AC_VEH_TRV_TCS) and 100 Nm

then after a time delay (C_T_DLY_TQ_MAX_ARS) LV_MAX_ARS is set to 1.

LV_MAX_ARS is indicating whether maximum torque request is present or not. The logical variable is used for validating the status of ARS.

Application conditions:

Initialisation: LV_TQ_MAX_ARS = 0
TQ_LOSS_ ARS_SP_MAX = 0 Nm
TQ_LOSS_ARS_SP_MAX_1 = 0 Nm
TQ_LOSS_ARS_AV_MAX = 0 Nm
TQ_LOSS_ARS_AV_MAX_1 = 0 Nm
T_DLY_TQ_MAX_ARS = C_T_DLY_TQ_MAX_ARS **at reset**

Update Rate: 10 ms

Activation: LV_VAR_ARS = 1 **and** LV_IGK = 1

Deactivation: -

Formula section:

```
If      VS_FIL ≤ C_V_MAX_ARS_DIAG          and
        N_32 < C_N_MAX_ARS_DIAG
Then    TQ_LOSS_ARS_SP_MAX  = TQ_LOSS_ARS_SP_MAX_1 +
                                           C_TQ_LOSS_ARS_SP_OFS
        TQ_LOSS_ARS_AV_MAX = TQ_LOSS_ARS_AV_MAX_1 +
                                           C_TQ_LOSS_ARS_AV_OFS
Else    TQ_LOSS_ARS_SP_MAX = TQ_LOSS_ARS_SP_MAX_1
        TQ_LOSS_ARS_AV_MAX = TQ_LOSS_ARS_AV_MAX_1
Endif
```

TQ_LOSS_ARS_SP_MAX_1 = IP_TQ_LOSS_ARS_SP_MAX

TQ_LOSS_ARS_AV_MAX_1 = IP_TQ_LOSS_ARS_AV_MAX

```
If      TQ_LOSS_ARS_SP_CAN > MIN (TQ_LOSS_ARS_SP_MAX, 100 Nm) or
        TQ_LOSS_ARS_AV_CAN > MIN (TQ_LOSS_ARS_AV_MAX, 100 Nm)
Then    T_DLY_TQ_MAX_ARS = T_DLY_TQ_MAX_ARS – 10ms
        If    T_DLY_TQ_MAX_ARS ≤ 0
        Then  LV_TQ_MAX_ARS = 1
        Else  LV_TQ_MAX_ARS = 0
        Endif
Else    T_DLY_TQ_MAX_ARS = C_T_DLY_TQ_MAX_ARS
        LV_TQ_MAX_ARS = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DLY_TQ_MAX_ARS	1	0...FFFFH	0...655.35	0.01	[s]
Time delay for activating LV_TQ_MAX_ARS					
C_TQ_LOSS_ARS_SP_OFS	1	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
Offset for TQ_LOSS_ARS_SP					
C_TQ_LOSS_ARS_AV_OFS	1	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
Offset for TQ_LOSS_ARS_AV					
C_V_MAX_ARS_DIAG	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
filtered vehicle speed threshold for diagnosis					
C_N_MAX_ARS_DIAG	1	0...FFH	0...8160	32	[rpm]
engine speed limit threshold for diagnosis					
IP_TQ_LOSS_ARS_SP_MAX	8*8	0...FFFFH	-1024...1023.96875	0.03125	[Nm]
LDPM_AC_VEH_TRV_TCS	8	10...FFF0H	-51.175...51.175	0.0015625	[m/s²]
LDP_VS_FIL	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
ARS target torque loss - maximum					
IP_TQ_LOSS_ARS_AV_MAX	8*8	0...FFFFH	-1024...1023.96875	0.03125	[Nm]
LDPM_AC_VEH_TRV_TCS	8	10...FFF0H	-51.175...51.175	0.0015625	[m/s²]
LDP_VS_FIL	8	0...FFFFH	0...511.99218	0.0078125	[km/h]
ARS Actual value of torque loss - maximum					

59.4.3 Torque Loss ARS

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_LOSS_ARS	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
torque loss by ARS-intervention					
TQ_LOSS_ARS_SP_CAN_MAX	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
torque loss SP via CAN by ARS-intervention					
TQ_LOSS_ARS_GRD	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque loss gradient - ARS					
TQ_LOSS_ARS_DIF	V/O	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Difference torque loss TQ_LOSS_ARS_SP_CAN - TQ_LOSS_ARS_AV_CAN					
LV_TQ_LOSS_ARS_IS_AD_INH	V/O	0...1H	0...1	1	[-]
flag for inhibition of IS-control adaptations due to torque losses ARS					

Input data:

LV_VAR_BN			

FUNCTION DESCRIPTION:

General information:

TQ_LOSS_ARS represents the engine torque loss due to power consumption by the ARS control unit during anti-roll-stabilisation intervention.

Application conditions:

Activation: LV_VAR_ARS = 1 **and** LV_IGK = 1

Deactivation: -

Initialisation: TQ_LOSS_ARS = 0 Nm **at reset**

Update Rate: 10 ms

Formula section:

Calculation of TQ_LOSS_ARS:

If LV_ARS_ENA = 1

Then TQ_LOSS_ARS_SP_CAN_MAX = IP_TQ_LOSS_ARS_SP_CAN_MAX

TQ_LOSS_ARS_GRD = TQ_LOSS_ARS_SP_CAN_MAX$_{N}$ –
TQ_LOSS_ARS_SP_CAN_MAX$_{N-1}$

TQ_LOSS_ARS_DIF = IP_TQ_LOSS_ARS_SP_CAN_MAX –
IP_TQ_LOSS_ARS_AV_CAN_MAX

TQ_LOSS_ARS = TQ_LOSS_ARS_DIF * C_FAC_TQ_LOSS_ARS_DIF +
IP_TQ_LOSS_ARS_AV_CAN_MAX + TQ_LOSS_ARS_GRD *
C_FAC_TQ_LOSS_ARS_GRD

```
Else    TQ_LOSS_ARS_GRD = 0 Nm
        TQ_LOSS_ARS_DIF = 0 Nm
        TQ_LOSS_ARS = 0 Nm
Endif
```

Condition for inhibition of IS-control adaptations:

```
If      TQ_LOSS_ARS < C_TQ_LOSS_ARS_AD
Then    T_TQ_LOSS_ARS_DLY_AD = C_T_ARS_DLY_AD
        LV_TQ_LOSS_ARS_IS_AD_INH = 1
Else    If      T_TQ_LOSS_ARS_DLY_AD > 0
        Then    decrement timer to zero
        Else    LV_TQ_LOSS_ARS_IS_AD_INH = 0
        Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQ_LOSS_ARS_SP_CAN_MAX	8*4	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_TQ_LOSS_ARS_SP_CAN	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32__TQ_LOSS_ARS	4	0...FFH	0...8160	32	[rpm]
ARS torque loss depending on torque loss (SP) via CAN and engine speed					
IP_TQ_LOSS_ARS_AV_CAN_MAX	8*4	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_TQ_LOSS_ARS_AV_CAN_MAX	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDPM_N_32__TQ_LOSS_ARS	4	0...FFH	0...8160	32	[rpm]
ARS torque loss depending on torque loss (AV) via CAN and engine speed					
C_FAC_TQ_LOSS_ARS_GRD	1	0...FFFFH	0...0.99998	1.53E-05	[-]
ARS torque loss gradient – correction factor					
C_FAC_TQ_LOSS_ARS_DIF	1	0...FFFFH	0...0.99998	1.53E-05	[-]
ARS torque loss actual value – correction factor					
C_TQ_LOSS_ARS_AD	1	8000...7FFFH	-1024...1023.96875	0.03125	[Nm]
threshold torque losses PSTE for inhibition of IS-control adaptations					
C_T_ARS_DLY_AD	1	0...FFH	0...2.55	0.01	[s]
threshold torque losses PSTE for inhibition of IS-control adaptations					

60 System Configuration

Table of Contents

Index

60.1 List of Configuration Data

General information:

This is a list of important non-calibratable data used in MSD80 6 Cylinder.

Input data:

NC_CBK_EX_NR	NC_CRK_WIN_SEG_LEN	NC_CYL_NR	NC_FAC_VB_RATIO
NC_IGBT_CUT_OFF_T	NC_IGK_ON_NR	NC_NR_TCHA	NC_KEY_OFF_NR
NC_KEY_OFF_THR	NC_KNKS_CONF	NC_KNKWB_INI	NC_KNKWE_INI
NC_MAF_FAC_CYL	NC_MISF_VERS	NC_MPL_T_MAX	NC_NR_CP_BUF
	NC_NR_EDGE_CAM_EX	NC_NR_EDGE_CAM_IN	NC_NR_ERR_DYN
NC_NR_ERR_HIS	NC_NR_GAP	NC_NR_TOOTH	NC_NR_TOOTH_GAP
NC_NR_TOOTH_STALL	NC_NR_TOOTH_TOL_ADD	NC_NR_TOOTH_TOL_MISS	NC_NR_VLD_TOOTH
NC_N_DIF_MIN_CRLC	NC_N_MAX	NC_N_MIN	NC_N_REF_MAX
NC_OFS_TDC0_REF_CRK	NC_PHA_SEG_ER_ENSD	NC_PSD_DLY_CPS	NC_PSN_SEG_TDC_REF
NC_SEG_TOOTH_RR	NC_SIZE_SEG_T_COR_BUF		NC_STATE_LSL_UP_IF
NC_STATE_VLS_UP_SIG_ACQ	NC_VB_MAX	NC_VB_SAMPLE_NR	NC_V_IGK_MAX
NLC_USE_ER_STND_MIS			

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_CBK_IN_NR	1	1...4H	1...4	1	[-]
Number of intake cylinder banks					
NC_LAM_SWI	-	0...1H	0...1	1	[-]
switch for linear lambda sensor					
NC_MAF_CONF	-	0...2H	0...2	1	[-]
System configuration flag (compiler) if an air-flow sensor is available (= 1) or not (= 0)					
NC_MAP_CONF	-	0...2H	0...2	1	[-]
System configuration flag (compiler) if a MAP sensor is available (= 1) or not (= 0)					
NC_T_SEG_FRQ	1				
Timer frequency used for T_SEG measurement					
NC_TD_LIM	-	0...30D4H	0...50	0.004	[ms]
maximum value for dwell time					
USE_SW_VER	1	1H 2H 3H 4H 5H 6H	NC_4V0 (MSV70) NC_4V4 (MSS70) NC_4V5 (MSV80) NC_4D0 (MSD80) NC_4D4 (MSD80-4Cyl) NC_4D8 (MSD80-8Cyl)	1	[-]
SW-switch indicating the current software project					
NC_NR_CBK_IVVT	-	1...2H	1...2	1	[-]
Number of camshaft cylinder banks					
NC_MAF_SENS_CONF	-	0...2H	0...2	1	[-]
Flag for MAF sensor configuration (only for turbo variant; 1 = FRQ sensor; 2 = FRQ and AN sensor)					

Description:

Name	Value
NC_CBK_IN_NR	2
NC_CBK_EX_NR	2
NC_CBK_NR	2
NC_CRK_WIN_SEG_LEN	30 °
NC_CYL_NR	6
NC_FAC_VB_RATIO	1.1076923
NC_IGBT_CUT_OFF_T	5 ms
NC_IGK_ON_NR	6
NC_KEY_OFF_NR	3
NC_KEY_OFF_THR	0.6933576 V
NC_KNKS_CONF	42
NC_KNKWB_INI	12 °crk
NC_KNKWE_INI	54 °crk
NC_LAM_SWI	1 (1 for linear Lam.)
NC_MAF_CONF	1
NC_MAF_FAC_CYL	5555.5
NC_MAP_CONF	1
NC_MISF_VERS	1
NC_MPL_T_MAX	60 ms
NC_N_DIF_MIN_CRLC	-150 1/min
NC_N_MAX	8160 1/min
NC_N_MIN	22 1/min
NC_N_REF_MAX	8160 1/min
NC_NR_CP_BUF	80
NC_NR_EDGE_CAM_EX	6
NC_NR_EDGE_CAM_IN	6
NC_NR_ERR_DYN	15
NC_NR_ERR_HIS	10
NC_NR_GAP	1
NC_NR_TOOTH	60
NC_NR_TOOTH_GAP	2
NC_NR_TOOTH_STALL	3
NC_NR_TOOTH_TOL_ADD	2
NC_NR_TOOTH_TOL_MISS	2
NC_NR_VLD_TOOTH	6
NC_OFS_TDC0_REF_CRK	60 °
NC_PHA_SEG_ER_ENSD	42 °
NC_PSD_DLY_CPS	2
NC_PSN_SEG_TDC_REF	54 °
NC_SEG_TOOTH_RR	4
NC_SIZE_SEG_T_COR_BUF	7
NC_STATE_LSL_UP_IF	1
NC_STATE_VLS_UP_SIG_ACQ	1
NC_T_SEG_FRQ	0x401CD0 (hex)
NC_TD_LIM	10 ms
NC_VB_SAMPLE_NR	8
NLC_USE_ER_STND_MIS	0
USE_SW_VER	= "NC_4D0" for MSD70
NC_NR_CBK_IVVT	1
NC_MAF_SENS_CONF	2
NC_NR_TCHA	2

60.2 Variant coding

60.2.1 General

Certain functionalities in the engine management system can be changed over or shut off via variant code or data record identifier. The variant code can be set by means of calibratable constants via application system. After the data record identifier has been changed, the control unit must be reset as the data record identification is only updated during system initialization. A further method to descern one variant from another is the automatical learning by means of diagnosis or by utilization of variant specific input signals.

60.2.2 Automatically learnt variants

<u>Output data:</u>

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_AT	V/O	0...1H	0...1	1	[-]
Automatic transmission recognized					
LV_VAR_AMT	V/O/S	0...1H	0...1	1	[-]
SSG-transmission recognized					
LV_VAR_ETCU	V/O/S	0...1H	0...1	1	[-]
BN (SSG or EGS)-transmission recognised					
LV_VAR_ACIN	V/O/S	0...1H	0...1	1	[-]
air conditioning recognized					
LV_VAR_ICL	V/O	0...1H	0...1	1	[-]
CAN instrument cluster ("Kombiinstrument") recognised					
LV_VAR_ASR	V/O/S	0...1H	0...1	1	[-]
ASR device recognised					
LV_VAR_ASR_3	V/O/S	0...1H	0...1	1	[-]
ASR 3 device recognised					
LV_VAR_ASR_4	V/O/S	0...1H	0...1	1	[-]
ASR 4 device recognised					
LV_VAR_MSW	V/O/S	0...1H	0...1	1	[-]
LV cruise control detected					
LV_VAR_BN_MSW	V/O/S	0...1H	0...1	1	[-]
LV cruise control detected - BN2000					
LV_VAR_PSTE	V/O/S	0...1H	0...1	1	[-]
LV power steering detected - BN2000					
LV_VAR_DCC	V/O/S	0...1H	0...1	1	[-]
LV distance cruise control detected - BN2000					
LV_VAR_ARS	V/O/S	0...1H	0...1	1	[-]
LV anti-roll-stabilisation detected - BN2000					
LV_VAR_PSTE_2	V/O/S	0...1H	0...1	1	[-]
PSTE_2 device recognized (BN2000 only)					
LV_VAR_BN_EFP	V/O/S	0...1H	0...1	1	[-]
EFP via CAN recognised (BN2000 only)					
LV_VAR_BN_GEAR_REV	O/V/S	0...1H	0...1	1	[-]
reverse gear via CAN recognised (BN2000/E60 - MT only)					
LV_VAR_BN_LTG_HDLP_L	O/V/S	0...1H	0...1	1	[-]
Dipped Beam via CAN recognised (BN2000 only)					
LV_VAR_BN_LDM	V/O/S	0...1H	0...1	1	[-]
LDM detected (E9x only)					
LV_VAR_BN_TRL	V/O/S	0...1H	0...1	1	[-]
TRL detected					
LV_VAR_EFP_CRASH	V/O/S	0...1H	0...1	1	[-]
EFP_CRASH-message detected					

LV_VAR_PSTE_3	V/O/S	0...1H	0...1	1	[-]
PSTE 3 (EHB3) device detected					
LV_VAR_NOX	V/O	0...1H	0...1	1	[-]
NOx sensor message detected					
LV_VAR_PBR	V/O/S	0...1H	0...1	1	[-]
Bit indicating electro-mechanical park brake variant					
LV_VAR_ETCU_3	V/O/S	0...1H	0...1	1	[-]
Transmission control unit, using 'Getriebedaten 3' message detected					
LV_VAR_TCT	O/V/S	0...1H	0...1	1	[-]
TCT-transmission recognised					
LV_VAR_ETCU_SPT	O/V/S	0...1H	0...1	1	[-]
variant sport gear box					
LV_VAR_4WD	O/V/S	0...1H	0...1	1	[-]
Logical variable indicating an 4 wheel drive variant					
LV_VAR_TQ_PBR	O/V/S	0...1H	0...1	1	[-]
Bit indicating electro-mechanical park brake torque request information variant					

Input data:

LV_ACIN	LV_ACCIN	STATE_PSTE_3_SRC	STATE_ETCU_SPT_SWI

FUNCTION DESCRIPTION:

Two states have been defined for automatic learning of the vehicle equipment:

1) Vehicle without equipment xx LV_VAR_xx = 0
2) Vehicle with equipment xx LV_VAR_xx = 1

The state which is valid in each case is saved in the non-volatile memory after the ECU self-holding phase. The state is initialised with 1) in the case of a new control unit (EEPROM has been erased) and when the variants are reset via serial protocol.
In state 1) re-learning to state 2) is possible within 3 seconds after ECU- reset and within a calibratable time after ignition key on. Learning is done by receiving the equipment related CAN- message.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_DET_VAR_CAN	1	0...FFH	0...255	1	[s]
time for learning variants via CAN after ignition key on					
LC_VAR_NOX	1	0...1H	0...1	1	[-]
Switch for learning of NOx sensor					

Transmission type

1) Variant is „Manually shifted transmission “ or transmission variant is to be learnt:

LV_AT = 0

LV_VAR_AMT = 0

LV_VAR_TCT = 0

This state is active as long as there is no non-volatile memory entry during an ECU initialisation. After this, the CAN communication will be observed to learn variants. After a timeout for the EGS message and SSG-message, LV_AT and LV_VAR_AMT will stay inactive. This is interpreted as variant „Manually shifted transmission.“

2) Variant is „Automatically shifted transmission with BN2000“:

LV_AT = 1

LV_VAR_AMT = 0

LV_VAR_TCT = 0

Re-learning conditions:

receiving CAN message “DREHMOMENT_ANF_EGS” (BN2000)

3) Variant is „Twin clutch transmission with BN2000“:

LV_VAR_TCT = 1

LV_VAR_AMT = 0

LV_AT = 0

Re-learning conditions:

receiving CAN message “DREHMOMENT_ANF_DKG” (BN2000)

4) Variant is “Automated manual transmission (SSG) with BN2000”:

LV_AT = 0

LV_VAR_AMT = 1

LV_VAR_TCT = 0

Re-learning conditions:

receiving CAN message “DREHMOMENT_ANF_SSG” (BN2000)

additional 1) Variant is “BN 2000 transmission (SSG, DKG or EGS)”:

LV_VAR_ETCU =1

Re-learning conditions:

receiving CAN message “GETRIEBEDATEN” (BN2000)

additional 2) Variant is “BN 2000 sportgear box" (DKG or EGS)”:

LV_VAR_ETCU_SPT =1

Re-learning conditions:

receiving STATE_ETCU_SPT_SWI = 1 of CAN message “GETRIEBEDATEN_2” (BN2000)

Air conditioner

1) Vehicle without air conditioner	LV_VAR_ACIN = 0
2) Vehicle with air conditioner	LV_VAR_ACIN = 1

Re-learning conditions:

receiving the LV_ACCIN bit (STATE_ACIN_CAN > 0H) of CAN message WAERMESTROM_KLIMA (IHKA, BN2000).

Dashboard with CAN interface

All vehicles are equipped with dashboard therefore

LV_VAR_ICL = 1 //default value

Vehicle with ASR device

1) Vehicle without ASR	LV_VAR_ASR = 0
2) Vehicle with ASR	LV_VAR_ASR = 1

Re-learning conditions:

receiving CAN message “DREHMOMENT_ANF_DSC” (BN2000)

LV_VAR_ASR_3 = 0

LV_VAR_ASR_4 = 0

LV_VAR_MSW = 0

BN-Vehicle with Multifunctional steering wheel (CRU interface) - (BN2000 only)

1) Vehicle without multifunctional steering wheel	LV_VAR_BN_MSW = 0
2) Vehicle with multifunctional steering wheel	LV_VAR_BN_MSW = 1

Re-learning conditions: receiving CAN-message “BEDIENUNG_TEMPOMAT” (SZ-Lenksäule)

Vehicle with PSTE (Power steering) device

1) Vehicle without PSTE	LV_VAR_PSTE = 0
2) Vehicle with PSTE	LV_VAR_PSTE = 1

Re-learning conditions:

receiving CAN message “LENKRADWINKEL” (SZ-Lenksäule, BN2000)

Distance Cruise Control (BN2000 only)

1) Vehicle without DCC	LV_VAR_DCC = 0
2) Vehicle with DCC	LV_VAR_DCC = 1

Re-learning conditions:

receiving the CAN message DREHMOMENT_ANF_ACC (ACC, BN2000)

Anti-roll-stabilisation (BN2000 only)

1) Vehicle without ARS	LV_VAR_ARS = 0
2) Vehicle with ARS	LV_VAR_ARS = 1

Re-learning conditions: receiving the CAN message STAT_ARS (ARS, BN2000)

Active Front Steering (BN2000 only)

1) Vehicle without AFS	LV_VAR_PSTE_2 = 0
2) Vehicle with AFS	LV_VAR_PSTE_2 = 1

Re-learning conditions:

receiving the CAN message DREHMOMENT_ANF_AFS (AFS, BN2000)

Electrical Fuel Pump (BN2000 only)

1) Vehicle without EFP-control-unit	LV_VAR_BN_EFP = 0
2) Vehicle with EFP-control-unit	LV_VAR_BN_EFP = 1

Re-learning conditions: receiving the CAN message STAT_EKP (EKP, BN2000)

Reverse Gear via CAN (E60 - MT only)

1) MT-Vehicle without GEAR_REV	LV_VAR_BN_GEAR_REV = 0
2) MT-Vehicle with GEAR_REV	LV_VAR_BN_GEAR_REV = 1

Re-learning conditions:

receiving the CAN message STAT_GANG_RUECKWAERTS (LM, BN2000)

Dipped Beam via CAN (BN2000 only)

1) Vehicle without LTG_HDLP_L	LV_VAR_BN_LTG_HDLP_L = 0
2) Vehicle with LTG_HDLP_L	LV_VAR_BN_LTG_HDLP_L = 1

Re-learning conditions: receiving the CAN message LAMPENZUSTAND (LM, BN2000)

Längsdynamikmodul (Longitudinal dynamics module) - LDM (E9x only)

1) Vehicle without LDM	LV_VAR_BN_LDM = 0
2) Vehicle with LDM	LV_VAR_BN_LDM = 1

Re-learning conditions:

receiving the CAN message "Anforderung Radmoment Antriebsstrang" (LDM, BN2000)

Status Anhänger (State Trailer) - TRL (BN2000 only)

1) Vehicle without TRL	LV_VAR_BN_TRL = 0
2) Vehicle with TRL	LV_VAR_BN_TRL = 1

Re-learning conditions:

receiving the CAN message STAT_ANHAENGER (K-CAN/AHM, BN2000)

Steuerung Crashabschaltung EKP (BN2000 only)

1) Vehicle without EKP_CRASH on CAN	LV_VAR_EFP_CRASH = 0
2) Vehicle with EKP_CRASH on CAN	LV_VAR_EFP_CRASH = 1

Re-learning conditions:

receiving the CAN message CTR_CRASH_SWO_EKP (MRSZ, BN2000)

EHB3 - Steering (BN2000 only)

1) Vehicle without EHB3	LV_VAR_PSTE_3 = 0
2) Vehicle with EHB3	LV_VAR_PSTE_3 = 1

Re-learning conditions:

receiving the CAN message DREHMOMENT_ANF_STE (EHB3, BN2000)

NOx – sensor

```
IF    LC_VAR_NOX = 1
THEN
      LV_VAR_NOX = 1
ELSE
      Learning of NOx - sensor
      1) Vehicle without NOx – sensor          LV_VAR_NOX = 0
      2) Vehicle with NOx – sensor             LV_VAR_NOX = 1
      Re-learning conditions:
      receiving the CAN message ID = 130h or ID = 135h on LoCAN
ENDIF
```

PBR – park brake (EMF) (BN2000 only)

1) vehicle without PBR: LV_VAR_PBR = 0
2) vehicle with PBR: LV_VAR_PBR = 1

Re-learning conditions:

Receiving the CAN message STATUS_EMF (EMF, BN2000)

TQ_PBR – park brake torque information (EMF) (BN2000 only)

1) vehicle without TQ_PBR:	LV_VAR_TQ_PBR = 0
2) vehicle with TQ_PBR:	LV_VAR_TQ_PBR = 1

Re-learning conditions:

Receiving the CAN message ST_RQ_EMF (EMF, BN2000)

Getriebedaten 3 – some transmission control units (BN2000 only)

1) Vehicle without ETCU_3 LV_VAR_ETCU_3 = 0

2) Vehicle with ETCU_3 LV_VAR_ETCU_3 = 1

Re-learning conditions:

receiving the CAN message GETRIEBEDATEN_3 (EGS_EL, BN2000)

Sollmomentanforderung – DXC message (BN2000 only)

1) Vehicle without DXC LV_VAR_4WD = 0

2) Vehicle with DXC LV_VAR_4WD = 1

Re-learning conditions:

receiving the CAN message SOLL_MOM_ANF (DXC, BN2000)

60.2.2.1 Sport-Switch / Sound-Flap

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CONF_SOF_SWI	V/O/S	0H 1H 2H	NO_SOF_SWI SOF_SWI SOF_SWI_AMT	1	[-]
Configuration of sport-switch					

Input data:

LV_SOF_SWI	LV_SOF_SWI_AMT		

Sport-Mode-Switch:

Two states have been defined for automatic learning of the Sport-Switch interface:
1) Vehicle without Sport-Switch
2) Vehicle with Sport-Switch (normal sport switch / AMT sport switch)

The state which is valid in each case is saved in the non-volatile memory after the ECU holding phase. The state is initialized with 1) in the case of a new control unit and when the variant is reseted .

Re-learning from the 1 variant to 2 variant is possible at each engine operating state. Learning is based on the evaluation of LV_SOF_SWI or LV_SOF_SWI_AMT (Switch pressed).

At each engine operating state the learning is effected.

```
If   CONF_SOF_SWI = 0
     Then If     LV_SOF_SWI = 1
            Then CONF_SOF_SWI = 1
            Else If     LV_SOF_SWI_AMT = 1
                 Then   CONF_SOF_SWI = 2
                 Else   CONF_SOF_SWI = 0
                 Endif
```

```
        Endif
    Else    CONF_SOF_SWI = CONF_SOF_SWI
Endif
```

Application conditions:

Initialisation: in case of new control unit: CONF_SOF_SWI =0

Recurrence: 100ms

Activation: at every engine operating state

Meaning of CONF_SOF_SWI:
0: no Sport-switch
1: Sport-switch
2: Sport-switch AMT^

60.2.2.2 Automatically learnt variants by power stage diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VAR_RLY_ACCOUT	V/O/S	0...1H	0...1	1	[-]
variant of ACC-relay recognised					
LV_VAR_EBOX_CFA	V/O/S	0...1H	0...1	1	[-]
variant of E-box-fan recognised					
LV_VAR_ECRAS_UP	V/O/S	0...1H	0...1	1	[-]
ECRAS system with ECRAS_UP flap detected (2 flap system)					
LV_VAR_RLY_ST	V/O/S	0...1H	0...1	1	[-]
variant of starter relay recognised					
LV_VAR_EF	V/O/S	0...1H	0...1	1	[-]
variant of exhaust flap recognised					
LV_VAR_LSH_UP	V/O/S	0...1H	0...1	1	[-]
Variant of CAT/exhaust type with upstream lambda sensors recognized					
LV_VAR_LSH_DOWN	V/O/S	0...1H	0...1	1	[-]
variant of CAT/exhaust type with upstream + downstream lambda sensors recognised					
PSN_LS	V/O	0...FFH	0...255	1	[-]
OBD-communication, OS-configuration Mode01 PID13					
PSN_LS_1	V/O	0...FFH	0...255	1	[-]
OBD-communication, OS-configuration Mode01 PID1D					
LV_VAR_SOF	V/O/S	0...1H	0...1	1	[-]
variant of sound-flap					
LV_VAR_AEB	O/V/S	0...1H	0...1	1	[-]
Active engine bracket configured [stored]					

Input data:

LV_CDN_VB_MIN_DIAG	LC_AD_CLR_VAR		

FUNCTION DESCRIPTION:

Description:

The components which are automatically learnt by powerstage diagnosis are:
--> see table in formula section.

The learning of the lambda sensors (here: via diagnosis of the lambda sensor heater) is a special case, because only specific combinations of the possibilities are valid.

Two states have been defined:
1) Vehicle without component EF, etc.
2) Vehicle with component EF, etc.

The state which is valid in each case is saved in the non-volatile memory after the ECU holding phase. The state is initialized with 1) in the case of a new control unit and when the variant is reset.

The re-learning from state 1) to state 2) is done only once after each ECU-initialisation (if the activation conditions are fulfilled). If a variant has been learnt and is afterwards reset to zero (via LC_AD_CLR_VAR), relearning is only possible after reset.

Learning of the components is based on the evaluation of the powerstage diagnosis of the related components. Therefore the components have to be activated for the learning cycle in order to get a proper power stage diagnosis. In state 1) the diagnosis is effected, however, an entry into the failure memory is suppressed.

The diagnosis result of the first 100ms-(or 200ms-)cycle must be rejected (due to initialisation). From the second cycle on the diagnosis is effected 10 times (1 check per recurrence). If at least 7 checks are error-free (symptom “open load” is zero) and valid (in case of LV_VAR_xxx = 0), the variant EF, ECRAS_UP, EBOX_CFA, AEB etc. has been learnt and LV_VAR_xxx = 1. With this state, the related powerstage diagnosis is cleared completely, i.e. the error entry is enabled.
If the variant LV_VAR_ECRAS_UP has been learnt, the detection of the ECRAS_DOWN variant (the lower ECRAS flap) is enabled (LV_VAR_ECRAS_DOWN). The algorithm and setting of LV_VAR_ECRAS_DOWN is done in ECRAS diagnosis module, not in the variant coding module.
The variant LV_VAR_EBOX_CFA is only learnt if LC_VAR_EBOX_CFA = 0. If LC_VAR_EBOX_CFA = 1, then LV_VAR_EBOX_CFA is also set 1 without learning.

Application conditions:

Initialisation: if ‘new control unit’ or
‘failure of nonvolatile memory recognized’ or
LC_AD_CLR_VAR 0→1 then LV_VAR_xxx = 0
LC_AD_CLR_SOF 0→1 then LV_VAR_SOF = 0

Recurrence:	200 ms	LSH_UP/_DOWN	(due to low PWM frequency)
	100 ms	all other components	

Activation: after every reset (once):

if LV_CDN_VB_MIN_DIAG = 1 **and**

LV_VAR_xxx = 0

Formula section:

Table for automatically learnt variants by power stage diagnosis - LV_VAR_xxx:

Component	Variant Keyword xxx	LV_VAR_xxx = 0	LV_VAR_xxx = 1
ACC-Relay	RLY_ACCOUT	no RLY_ACCOUT	RLY_ACCOUT
E-BOX-fan	EBOX_CFA	no EBOX_CFA	EBOX_CFA
Starter Relay	RLY_ST	no RLY_ST	RLY_ST
Electronically controlled radiator shutters (upper flap)	ECRAS_UP	no ECRAS_UP	ECRAS_UP
Exhaust flap	EF	no EF	EF
Sound flap	SOF	no SOF	SOF
Exhaust type	LSH_UP	no CAT, no lambda sensors	with CAT and lambda sensors
Exhaust type	LSH_DOWN	no monitor lambda sensors	with monitor lambda sensors
Active engine brackets	AEB	no AEB	AEB

Learning of LV_VAR_EBOX_CFA

```
if      LC_VAR_EBOX_CFA = 1
then    LV_VAR_EBOX_CFA = 1
else    LV_VAR_EBOX_CFA learnt by power stage diag.
endif
```

Configuration and learning of CAT/exhaust type

```
If      C_CONF_CAT = 2                 '2 banks, control + monitor sensors'   (no learning)
then    LV_VAR_LSH_UP  =  1
        LV_VAR_LSH_DOWN  =  1
endif

If      C_CONF_CAT = 1
then    learning of exhaust type:
        All 4 sensors will be checked by power stage diagnosis as mentioned above.

        If    none of the 4 sensors is recognized          'without CAT'
        then  LV_VAR_LSH_UP  =  0
              LV_VAR_LSH_DOWN  =  0
        If    none of the 2 monitor sensors (...DOWN...) is recognized but at least one
              of the 2 control sensors (...UP...) is recognized     '2 banks, control sensors'
        then  LV_VAR_LSH_UP  =  1
              LV_VAR_LSH_DOWN  =  0

        If    at least 1 of the 2 monitor sensors is recognized
                                                '2 banks, control + monitor sensors'
        then  LV_VAR_LSH_UP  =  1
              LV_VAR_LSH_DOWN  =  1
endif

If      C_CONF_CAT = 0
```

```
then      limited learning of exhaust type:
          Only the 2 control sensors will be checked.
          If     at least one sensor is recognized            '2 banks, control sensors'
          then   LV_VAR_LSH_UP  =  1
                 LV_VAR_LSH_DOWN  =  0
          Else   LV_VAR_LSH_UP  =  0                'without CAT'
                 LV_VAR_LSH_DOWN  =  0
endif

If        LV_VAR_LSH_UP = 0
then      no sensors available
          PSN_LS = 00000000 bin
          PSN_LS_1 = 00000000 bin
Endif

If        LV_VAR_LSH_UP = 1 and LV_VAR_LSH_DOWN = 0
then      PSN_LS = 00010001 bin (bank 1 sensor 1 and bank 2 sensor 1)
          PSN_LS_1 = 00000101 bin
endif

If        LV_VAR_LSH_UP = 1 and LV_VAR_LSH_DOWN = 1
then      PSN_LS = 00110011 bin (bank 1 sensor1/2 and bank 2 sensor 1/2)
          PSN_LS_1 = 00001111 bin
endif
```

Configuration and learning of AEB/active engine brakets

```
If        C_CONF_AEB = 0
then      no learning of active engine brackets
elseif    C_CONF_AEB = 1
then      learning of active engine brackets by powerstage diagnosis
elseif    C_CONF_AEB = 2
then      LV_VAR_AEB = 1
else      //same behavior like C_CONF_AEB = 1
```

learning of AEB:

Component AEB will be checked by power stage diagnosis as mentioned above.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CONF_CAT	1	0...3H	0...3	1	[-]
Configuration of exhaust type learning					
LC_VAR_EBOX_CFA	1	0...1H	0...1	1	[-]
Configuration of LV_VAR_EBOX_CFA (if 1, LV_VAR_xx =1, else LV_VAR_xx learnt by power stage diag)					
C_CONF_AEB	1	0...FFH	0...255	1	[-]
Configuration of active engine brackets learning					

60.2.3 Calibrateable or automatically learnt variants

60.2.3.1 Mass air flow meter (HFM)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VAR_MAF	V/O/S	0...1H	0...1	1	[-]
variant mass air flow meter					
CTR_AVL_MAF	V	0...FFFFFFFFH	0...4294967295	1	[-]
Counter HFM learning function for case HFM is available					
CTR_NOT_AVL_MAF	V	0...FFFFFFFFH	0...4294967295	1	[-]
Counter HFM learning function for case HFM not available					
LV_VAR_MAF_LEARNT	O/V	0...1H	0...1	1	[-]
HFM learning function has been carried out					

Input data:

LV_ES	T_PER_MAF_FRQ_BAS[NC_MAF_NR]		

General information:

The aim of these function is to check the availability of a mass air flow meter (HFM). The function is solely avtive at initial operation. The lernaed availability of a HFM can be manually overwrited by the use of C_LEARN_HFM_OVERRIDE.

Application conditions:

Initialisation: LV_VAR_MAF = 0 at first reset, if learnt -> initialised from NVMY;
LV_VAR_MAF_LEARNT = 0 at first reset, if learnt -> initialised from NVMY
CTR_AVL_MAF = 0 at reset
CTR_NOT_AVL_MAF = 0 at reset

Recurrence: *10ms*

Activation: **If** LV_ES = 1

Deactivation **If** LV_VAR_MAF = 1 **or**
LV_VAR_MAF_LEARNT = 1

Formula section:

```
If(1)       C_LEARN_HFM_OVERRIDE = 0
Then(1)     If(2)       C_T_PER_MAF_FRQ_AVL_MIN < T_PER_MAF_FRQ_BAS[i] <
                                                C_T_PER_MAF_FRQ_AVL_MAX
            Then(2)     CTR_AVL_MAF ++    and
                        $CTR\_NOT\_AVL\_MAF_n = CTR\_NOT\_AVL\_MAF_{n-1}$
                        If(3)       CTR_AVL_MAF > C_CTR_AVL_MAF
                        Then(3)     LV_VAR_MAF = 1     and
                                    LV_VAR_MAF_LEARNT = 1

            Else(2)     CTR_NOT_AVL_MAF ++      and
                        $CTR\_AVL\_MAF_n = CTR\_AVL\_MAF_{n-1}$
                        If(3)       CTR_NOT_AVL _MAF > C_CTR_ NOT_AVL_MAF
                        Then(3)     LV_VAR_MAF = 0     and
                                    LV_VAR_MAF_LEARNT = 1
                        Endif(3)
            Endif(2)

Else(1)     If(2)       C_LEARN_HFM_OVERRIDE = 1
            Then(2)     LV_VAR_MAF = 1
            Else(2)     If(3)       C_LEARN_HFM_OVERRIDE = 2
                        Then(3)     LV_VAR_MAF = 0
                        Endif(3)
            Endif(2)
Endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_PER_MAF_FRQ_AVL_MIN	1	0...FFFFH	0...65535	1	[µs]
min Threshold to learn frequency HFM component					
C_T_PER_MAF_FRQ_AVL_MAX	1	0...FFFFH	0...65535	1	[µs]
max Threshold to learn frequency HFM component					
C_CTR_AVL_MAF	1	0...FFFFFFFFH	0...4294967295	1	[-]
Minimum cycle number to learn HFM component is available					
C_LEARN_HFM_OVERRIDE	1	0...2H	0...2	1	[-]
Overwrite HFM learning: 0 = HFM learning enabled, 1 = HFM is available, 2 = HFM is not available					
C_CTR_NOT_AVL_MAF	1	0...FFFFFFFFH	0...4294967295	1	[-]
Minimum cycle number to learn HFM component is not available					

60.2.4 Engine Configuration Coding

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CBK_MPL	V/O	0...1H	0...1	1	[-]
logical variable multiple cylinder banks (0 = single bank, 1 = multiple banks, presently not available)					
LV_IVVT_IN_AND_EX	V/O	0...1H	0...1	1	[-]
logical variable IVVT (0 = inlet cam variable, 1 = inlet & exhaust cam variable)					
LV_INJ_CONF	V/O	0...1H	0...1	1	[-]
logical variable injection configuration (0 = intake manifold injection, 1 = direct injection)					
LV_ISA_CONF	V/O	0...1H	0...1	1	[-]
logical variable idle speed actuator configuration (0 = no idle speed actuator, 1 = idle speed actuator)					
VIM_TYPE	V/O	0H 1H	CLOSED_LOOP SWITCHED	1	[-]
VIM configuration					
CAM_TYPE	V/O	0H 1H 2H	3_SEG 4_SEG_SYM 4_SEG_ASYM	1	[-]
configuration of camshaft sensor disc					

General information:

The configuration is done for multiple cylinder bank, inlet and exhaust cam available, MPI or GDI, idle speed actuator (ISA), type of camshaft disc, type of Variable Intake Manifold and configuration of soundflap.
The constant NC_MAF_FAC_CYL depends on NC_CYL_NR.

Application conditions:

Initialisation: **at reset:**

LV_CBK_MPL = 0

LV_IVVT_IN_AND_EX = 1

LV_INJ_CONF = 1

LV_ISA_CONF = 0

CAM_TYPE = 0

VIM_TYPE = 0

Recurrence: -

Activation: at reset

Deactivation: -

60.2.5 Calibrateable Variants

60.2.5.1 Variant of DMTL

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CONF_DMTL	V/O	0...1H	0...1	1	[-]
configuration of DMTL					

General information:

All calibratable variants are updated only during ECU initialization.

If a component is deactivated by variant coding, the related function and diagnosis will **not** be executed.

Application conditions:

Initialisation: **at reset:** LV_CONF_DMTL = C_CONF_DMTL

Recurrence: -

Activation: at every engine operating state

Deactivation: -

Formula Section:

Meaning of LV_CONF_DMTL and C_CONF_DMTL

- 0 = no DMTL available, diagnosis of tank leakage detection is inhibited
- 1 = DMTL available, diagnosis of tank leakage detection is enabled

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CONF_DMTL	1	0...1H	0...1	1	[-]
configuration bit for tank leakage detection					

60.2.5.2 Variant of TCO_2

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VAR_TCO_2	V/O	0...1H	0...1	1	[-]
TCO_2 sensor is available (=1)					

General information:

All calibratable variants are updated only during ECU initialization.

If a component is deactivated by variant coding, the related function and diagnosis will **not** be executed.

Application conditions:

Initialisation: **at reset:** LV_VAR_TCO_2 = LC_VAR_TCO_2

Recurrence: -

Activation: at every engine operating state

Deactivation: -

Formula Section:

Meaning of LV_VAR_TCO_2 and LC_VAR_TCO_2

- 0 = no TCO_2 sensor available
- 1 = TCO_2 sensor vailable

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_VAR_TCO_2	1	0...1H	0...1	1	[-]
configuration switch for TCO_2 sensor					

60.2.5.3 Variant for EFP-control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CONF_SWI_EFP_OUT	V/O	0...2H	0...2	1	[-]
configuration of EFP-control					

General information:

All calibratable variants are updated only during ECU initialization.

If a component is deactivated by variant coding, the related function and diagnosis will **not** be executed.

Application conditions:

Initialisation: **at reset:** CONF_SWI_EFP_OUT = C_CONF_SWI_EFP_OUT

Recurrence: -

Activation: at every engine operating state

Deactivation: -

Formula Section:

Meaning of CONF_SWI_EFP_OUT and C_CONF_SWI_EFP_OUT

- 0 = no CAN-EFP available, EFP control via PWM output stage
- 1 = CAN-EFP signal via local CAN, EFP control via local CAN
- 2 = CAN-EFP signal via powertrain CAN, EFP control via powertrain CAN

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CONF_SWI_EFP_OUT	1	0...2H	0...2	1	[-]
configuration bit for EFP - control					

60.2.5.4 Variant for Oil Level or Quality sensor

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CONF_OIL_LEVEL_SENS	V/O	0...2H	0...2	1	[-]
configuration of EFP-control					

General information:

All calibratable variants are updated only during ECU initialization.

Application conditions:

Initialisation: **at reset:** *CONF_OIL_LEVEL_SENS = C_CONF_OIL_LEVEL_SENS*

Recurrence: -

Activation: at every engine operating state

Deactivation: -

Formula Section:

Meaning of CONF_OIL_LEVEL_SENS

- 0 = no Sensor
- 1 = TOENS are available
- 2 = Oil quality sensor on BSD interface are available

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CONF_OIL_LEVEL_SENS	1	0...2H	0...2	1	[-]
configuration bit for EFP - control					

60.2.5.5 -Variant for Turbo Charger or Stratified combustion

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VAR_TCHA	V/O	0...1H	0...1	1	[-]
Configuration for turbo charger					

General information:

All calibratable variants are updated only during ECU initialization.

Application conditions:

Initialisation: **at reset:** LV_VAR_TCHA = LC_TCHA_CONF

Recurrence: -

Activation: at every engine operating state

Deactivation: -

Formula Section:

Meaning of LC_TCHA_CONF and LV_VAR_TCHA

- 0 = N53 configuration
- 1 = N54 Configuration

LV_VAR_TCHA = 0 disabled Functions

DMTL		17B00401.00A
		17B00C01.00D
		17B00Y01.00C
CHRG	PUT-Diag	43A0GH01 43A0GI01
CHRG	RFP-Diag	43A08F01
CHRG	WG-Diag	43A0HF01

LV_VAR_TCHA = 1 disabled Functions

NOX:		
EGR	ACR	43301601.00A 30408501.00A 43400W01.00C 14601201.00B 30908I01.00B 30908J01.00B 30908K01.00B 30908L01.00A 43903J02.00A 30A0IK01.00B 30A0IL01.00B 30A0IM01.00A 30A0IN01.00A 30A0IO01.00A 30A0IP01.00A 30A0IQ01.00A 30A0IR01.00B 30A0IS01.00A 43R01801.00A

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TCHA_CONF	1	0...1H	0...1	1	[-]
System configuration flag for turbo charged engine (=1), for naturally aspirated engine (=0)					

60.2.5.6 Vehicle Variant

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VAR_BN	V/O	0...1H	0...1	1	[-]
PT-CAN Variant (0 = CAN11h / 1= BN2000)					
LV_VAR_VEH	V/O	0...1H	0...1	1	[-]
Vehicle Variant with power module recognised - BN2000 only					
VAR_VEH	V/O	0H 1H 2H ~~3H~~	E60 E65 PL2 ~~E83_E85~~	1	[-]
Vehicle variant					

General information:

The manual setting of the vehicle variant is possible by using the constant C_VAR_VEH. Depending on this constant the flags for "Bordnetz" and "Power module" will be set. This makes sure the compatibility to older revisions.

Depending on the variant the flags are set in the following way:

C_VAR_VEH		**LV_VAR_BN**	**LV_VAR_VEH**
Phys. value	**Hex**		
E60	00H	1	0
E65	01H	1	1
PL2	02H	1	0
~~**E83/E85**~~	~~03H~~	~~0~~	~~0~~

Application conditions:

Initialization: at ECU-reset

Recurrence: -

Activation: at reset

// Berechnung nur bei ECU-Init

// kompatible Berechung von LV_VAR_BN und LV_VAR_VEH:

```
VAR_VEH = C_VAR_VEH

LV_VAR_BN = 1              //  BN2000

    if      (VAR_VEH == E65)
    then    LV_VAR_VEH = 1     // Vehicle with Powermodule (E65)
    else    LV_VAR_VEH = 0     // Vehicle without Powermodule (E6x)
    endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VAR_VEH	1	0H 1H 2H ~~3H~~	E60 E65 PL2 ~~E83_E85~~	1	[-]
vehicle configuration code					

60.2.5.7 Variant of St_varkat

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
St_varkat	O/V	0...FFH	0...255	1	[-]
Status Variante Abgasanlage (mono/stereo)					

Input data:

NC_CBK_EX_NR		

Application conditions:

Initialisation at reset:

```
if      NC_CBK_EX_NR = 2
then    St_varkat = 1
else    St_varkat = 0
endif
```

Recurrence: -

Activation: at every engine operating state

Deactivation: -

Formula Section: -

60.2.6 Merge of learned variants for BMW- environment

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_VAR_DET_CUS_1	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
carrier double-word 1 of all learned variants for customer environment					
STATE_VAR_DET_CUS_2	V/O	0...FFFFFFFFH	0...4294967295	1	[-]
carrier double-word 2 of all learned variants for customer environment					

FUNCTION DESCRIPTION:

General information:

All learned variants LV_VAR_xx must be merged in two variables for customer use. This variables (carrier double-words) are calculated once per driving cycle after the automatically learning of variants has been finished or if LV_VAR_MAF changes from 0 -> 1. The LV_VAR- bits for each variant has to be mirrored in the bits of the carrier double-words. The places of the variants in the words are described in the table in the formula section.

Application conditions:

Initialisation: all 0

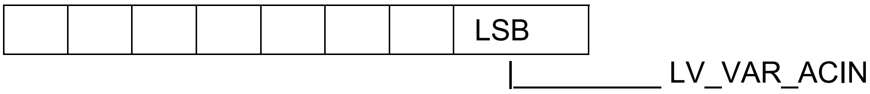
Recurrence: 1000 ms

Activation: at reset

Deactivation: -

Formula section:

The table contains the LV_VAR_xx- flags, how they have to be mirrored in the carrier double-words STATE_VAR_DET_CUS_1 and STATE_VAR_DET_CUS_2. Position 0x0 is the LSB, position 0x80000000 is the MSB of the carrier double-word (like shown in the picture below).



Content of STATE_VAR_DET_CUS_1:

Position (0 =LSB)	Position Hex	Varname
0	0x 0	LV_VAR_ACIN
1	0x 1	LV_VAR_AMT
2	0x 2	LV_VAR_ARS
3	0x 4	LV_VAR_ASR
4	0x 8	LV_VAR_ASR_3
5	0x 10	LV_VAR_ASR_4
6	0x 20	LV_VAR_BN

7	0x 40	LV_VAR_BN_EFP
8	0x 80	LV_VAR_BN_GEAR_REV
9	0x 100	LV_VAR_BN_LDM
10	0x 200	LV_VAR_BN_LTG_HDLP_L
11	0x 400	LV_VAR_BN_MSW
12	0x 800	LV_VAR_DCC
13	0x 1000	LV_VAR_EBOX_CFA
14	0x 2000	0
15	0x 4000	LV_VAR_EF
16	0x 8000	LV_VAR_ETCU
17	0x 10000	LV_VAR_ICL
18	0x 20000	LV_VAR_LSH_DOWN
19	0x 40000	LV_VAR_LSH_UP
20	0x 80000	LV_VAR_MSW
21	0x 100000	LV_VAR_PSTE
22	0x 200000	LV_VAR_PSTE_2 **OR** (LV_VAR_PSTE_3 **AND** STATE_PSTE_3_SRC=="AFS")
23	0x 400000	LV_VAR_RLY_ACCOUT
24	0x 800000	LV_VAR_RLY_ST
25	0x 1000000	LV_VAR_AEB
26	0x 2000000	*reserve*
27	0x 4000000	LV_VAR_SOF
28	0x 8000000	LV_AT
29	0x 10000000	LV_VAR_TCT
30	0x 20000000	LV_VAR_VEH
31	0x 40000000	LV_VAR_BN_TRL

Content of STATE_VAR_DET_CUS_2:

Position (0 =LSB)	**Position Hex**	**Varname**
0	0x 0	LV_VAR_MAF
1	0x 1	0
2	0x 2	LV_VAR_ETCU_SPT
3	0x 4	0
4	0x 8	0
5	0x 10	0
6	0x 20	0
7	0x 40	0
8	0x 80	0
9	0x 100	0
10	0x 200	0
11	0x 400	0
12	0x 800	0
13	0x 1000	0
14	0x 2000	0
15	0x 4000	0
16	0x 8000	0
17	0x 10000	0
18	0x 20000	0
19	0x 40000	0

20	0x 80000	0
21	0x 100000	0
22	0x 200000	0
23	0x 400000	0
24	0x 800000	0
25	0x 1000000	0
26	0x 2000000	0
27	0x 4000000	0
28	0x 8000000	0
29	0x 10000000	0
30	0x 20000000	0
31	0x 40000000	0

60.2.7 Runtime solution

Input data:

N			

Description:

The tables show the configuration switches to inhibit function blocks. If a function block is disabled (= 1) or exceeds the engine speed threshold, the outputs of these modules are then frozen.

Switch	Description	Modules
C_N_VAR_DIS_MISF	Misfire (AGGR MISF)	00S08201.00F
		17B00T02.00E
		17B02E02.00F
		17100L01.00C
		30F00102.00D
		17B00201.00A
		30B00E02.00C
C_N_VAR_DIS_ER	Engine roughness	00S08401.00H
		17100K01.00C
		17402P02.00D
		17A0EM01.00C
C_N_VAR_DIS_INJ	Cylinder balancing	30402U01.00B
LC_VAR_DIS_OBD_CAT	Cat efficiency	17B00801.00C
		01B00A02.00O
		01B02D02.00E
LC_VAR_DIS_NOX	NOx functions	02703O01.00C
		02703002.00E
		43705S01.00B
		43703N01.00B
		02706401.00A
		02706001.00B

		02704M01.00B
		02704F03.00C
		02705X01.00E
		02706301.00A
		02706202.00C
		02705101.00A
		02703Q03.00P
		02704G02.00F
		02704901.00I
		02703P03.00Q
		02705V01.00I
		02706I01.00C
		02705T01.00C
		43704A01.00B
		02702Z06.00E
		02703801.00E
		02A0B202.00B
LC_VAR_DIS_CRU	Cruise control	17G00101.00G
		17G00201.00H
		17G00301.00E
		17G00401.00C
		17G00E01.00J
LC_VAR_DIS_TQ_LOSS_ ARS	Torque loss – anti-roll-stabilization	17D02601.00E
LC_VAR_DIS_DMTL	LDP function	17B00C01.00J
LC_VAR_DIS_S_FCT	Lambda testing for stratified mode	43500J01.00A
Disabled functions	Sound flap diagnosis	43A0CJ01.00A

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_VAR_DIS_ER	1	0...FFH	0...8160	32	[rpm]
switch off threshold for engine roughness					
C_N_VAR_DIS_INJ	1	0...FFH	0...8160	32	[rpm]
switch off threshold for cylinder balancing					
LC_VAR_DIS_NOX	1	0...1H	0...1	1	[-]
switch off function for NOx functions					
LC_VAR_DIS_CRU	1	0...1H	0...1	1	[-]
switch off function cruise control					
LC_VAR_DIS_TQ_LOSS_ARS	1	0...1H	0...1	1	[-]
Switch off function torque loss ¿ anti-roll-stabilization					
LC_VAR_DIS_DMTL	1	0...1H	0...1	1	[-]
Switch off tank leakage diagnosis					
C_N_VAR_DIS_MISF	1	0...FFH	0...8160	32	[rpm]
switch off threshold for misfire					
LC_VAR_DIS_S_FCT	1	0...1H	0...1	1	[-]
Switch off stratified functions					

60.2.7.1 Learnt ECU variants

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_LRN_ECU	V/O/S	FFFFH 1111H 5A5AH A5A5H 4E4EH AEAEH BCBCH A2A2H 2A2AH E4E4H 0H	ROM_NOT_PLAUS LEARNING_FAILED C1_LOT1 C1_LOT2 C1_LOT3 C1_LOT4 SERIAL_ECU C2_LOT1 C2_LOT2 C2_LOT3 NOT_LEARNED	1	[-]
Variant ECU					
ECU_LOCK_REQ	V/O	0...1H	0...1	1	[-]
Bit indicating ECU locked					

Input data:

NC_HW_REV	NC_FLS_DFT		

Export actions:

ACTION_INFR_GetLd(OUT <V_VAR_LD>)
This action returns the LOT detection for ECU variant coding
ACTION_INFR_GetAmpDt(OUT <V_VAR_AMP**>)**
This action returns the AMP detection for ECU variant coding

Description for actions:

ACTION_INFR_GetLd(OUT <V_VAR_LD>)

This action returns the LOT detection for ECU variant coding

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
V_VAR_LD	OUT	0 ... 7FFFH	0 ... 5	1.5259e-4	V
Voltage for LOT detection					

ACTION_INFR_GetAmpDt(OUT <V_VAR_AMP**>)**

This action returns the AMP detection for ECU variant coding

Parameter	Type	Hex. Limits	Phys. Limits	Resol.	Unit
V_VAR_AMP	OUT	0 ... 7FFFH	0 ... 5	1.5259e-4	V
Voltage for AMP detection					

General information:

The detection of ECU variants is realised by a learning function. The state for a successful ECU variant detection is checked at every ECU initialisation. The according information has to be detected in a previous ECU initialisation by the learning function and has to be stored in the non-volatile memory. If the check, which ECU variant is identified, gets to a valid result, the learning function for ECU detection won't be started again. Otherwise it will start up like in the description below:

Combination table:

Muster	**Lot-Detection v_var_ld**	**AMP-Detection v_var_amp**
C1.1 Lot1	0V - 0,4V 47k Rpd	0V - 0,4V 47K Rpd
C1.1 Lot2	3,1V - 3,3V 47k Rpu	0V - 0,4V 47K Rpd
C1.2 Lot3	1,45V - 1,85V 47k / 47k - Teiler	0V - 0,4V 47K Rpd **lock engine**
C1.1 Lot4	0,83V - 1,23V 22k / 10k - Teiler	0V - 0,4V 47K Rpd
C1.1 Lot5	0,83V - 1,23V 22k / 10k - Teiler	0V - 0,4V 47K Rpd
C2 Lot1	0,93V - 1,13V 22k / 10k - Teiler	3,1V - 3,3V 47K Rpu
C2 Lot2	1,88V - 2,08V 22k / 33k - Teiler	3,1V - 3,3V 47K Rpu
C2 Lot3	2,39V - 2,59V 22k / 68k - Teiler	3,1V - 3,3V 47K Rpu
C2 serial	3,10V - 3,3V 10k Rpu	3,1V - 3,3V 47K Rpu

Comment: C1.2 Lot3 is locked for delivery due to SW not using KNK control

C1.1 Lot5 is learnt as C1.1 Lot4 because there is no difference regarding Lot-Detection

For Initialization ONLY:

```
SWITCH (NC_HW_REV)
case   STATE_LRN_ECU_C1_LOT1
               STATE_LRN_ECU = C1_LOT1  /* C1_LOT1 ecu learned */

case   STATE_LRN_ECU_C1_LOT2
               STATE_LRN_ECU = C1_LOT2 /* C1_LOT2 ecu learned */

case   STATE_LRN_ECU_C1_LOT3
               STATE_LRN_ECU = C1_LOT3 /* C1_LOT3 ecu learned */
               ECU_LOCK_REQ  = 1

case   STATE_LRN_ECU_C1_LOT4
               STATE_LRN_ECU = C1_LOT4 /* C1_LOT4 ecu learned */

case   STATE_LRN_ECU_C2_LOT1
               STATE_LRN_ECU = C2_LOT1 /* C2_LOT1 ecu learned */

case   STATE_LRN_ECU_C2_LOT2
               STATE_LRN_ECU = C2_LOT2       /* C2_LOT_2 ecu learned */

case   STATE_LRN_ECU_C2_LOT3
               STATE_LRN_ECU = C2_LOT3       /* C2_LOT_3 ecu learned */

case   STATE_LRN_ECU_C2_SERIAL:
               STATE_LRN_ECU = SERIAL_ECU            /* Serial ecu learned */

case   STATE_LRN_ECU_NOT_LEARNED:
               STATE_LRN_ECU = NOT_LEARNED                 /* No ecu learned */
break
default
       STATE_LRN_ECU = ROM_NOT_PLAUS
       /* Status cannot be stored in flash */
```

For cyclical learning:

```
ACTION_INFR_GetLd(V_VAR_LD)            /*read LOT value from the ADC */

ACTION_INFR_GetAmpDt(V_VAR_AMP)    /*read AMP value from the ADC */

If(1)     NC_HW_REV != NC_FLS_DFT
THEN(1)
    Do nothing /*nc_hw_rev already set  (plausible or not plausible)*/
ELSE(1)
    If(2) STATE_LRN_ECU == NOT_LEARNED
    THEN(2)
        If(3) CTR_LRN_ECU < 10
        THEN(3)
                CTR_LRN_ECU ++    /* time out counter*/


                If(3a)  V_VAR_AMP < 1.5
                THEN (3a)
                 /*  C1.1 Muster */
                        If(3b)  0 <= V_VAR_LD <= 0.4
                        THEN(3b)
                                STATE_LRN_ECU = C1_LOT1     /*C1.1 Lot1*/
                        ELSE IF(3b) 0.83 <= V_VAR_LD <= 1.23
                                THEN(3b)
                                        STATE_LRN_ECU = C1_LOT4          /* C1.1 Lot4 */
                        ELSE IF(3b)  1.45 <= V_VAR_LD <= 1.85
                                THEN(3b)
                                        ECU_LOCK_REQ = 1                       /*engine stop*/
                        ELSE IF(3b) 3.1 <= V_VAR_LD <= 3.3
                                THEN(3b)
                                        STATE_LRN_ECU = C1_LOT2          /*  C1.1 Lot2 */
                                ENDIF(3b)
                Else (3a)
                /* C2 Muster */
                        If(3c)  0.93 <= V_VAR_LD <= 1.13
                        THEN(3c)
                                STATE_LRN_ECU = C2_LOT1                       /* C2 Lot1*/
                        ELSE IF(3c)  1.88 <= V_VAR_LD <= 2.08
                                THEN(3c)
                                        STATE_LRN_ECU = C2_LOT2              /*  C2 Lot2 */
                        ELSE IF(3c)  2.39 <= V_VAR_LD <= 2.59
                                THEN(3c)
                                        STATE_LRN_ECU = C2_LOT3              /*   C2 Lot3 */
                        ELSE IF(3c)  3.1 <= V_VAR_LD <= 3.3
                                THEN(3c)
                                        STATE_LRN_ECU = SERIAL_ECU     /*   C2 serial */
                        ENDIF(3c)
```

```
                    ENDIF(3a)
          ELSE(3)
                        STATE_LRN_ECU = LEARNING_FAILED
          ENDIF(3)
    ELSE(2)
        If(4)  ACTION_ECM3_DisableReq()
        THEN(4)
                    DISABLE ALL INTERRUPTS
                    STORE THE DETECTED LOT INTO FLASH
                    FORCE RESET
         ELSE(4)
                    /*wait until MU shutdown*/
         ENDIF(4)

    ENDIF(2)
ENDIF(1)
```

60.3 Standard analog inputs

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
V_IGK_BAS	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Battery voltage after key raw acquisition					
VB_BAS	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Battery voltage after relay raw acquisition.					
V_LS_UP_1	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Upstream oxygen sensor Voltage raw acquisition - Bank n° 1					
V_LS_UP_2	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Upstream oxygen sensor Voltage raw acquisition - Bank n° 2					
V_AMP	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Ambient pressure sensor raw acquisition					
V_LS_DOWN_1	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Downstream oxygen sensor Voltage raw acquisition - Bank n° 1					
V_LS_DOWN_2	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Downstream oxygen sensor Voltage raw acquisition - Bank n° 2					
V_PVS_1	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage of pedal value sensor 1					
V_PVS_2	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage of pedal value sensor 2					
VCC_PVS_1	V/O	0...3FFH	0...9.99023	9.7656e-3	[V]
Voltage supply for pedal value sensor 1 (and TPS) raw acquisition					
VCC_PVS_2	V/O	0...3FFH	0...9.99023	9.7656e-3	[V]
Voltage supply for pedal value sensor 2 raw acquisition					
V_TPS_1	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Throttle position sensor 1 raw acquisition					
V_TPS_2	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Throttle position sensor 2 raw acquisition					
V_SOF_SWI	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage of sport switch					
V_SOF_SWI_MON	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage of sport flitch for monitoring					
MAP_BAS	O	0...3FFH	0...4.99511	4.8828e-3	[V]
manifold air pressure (form mux)					
V_FUP	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
LPG pressure in the fuel rail, voltage raw value (10 bits).					
V_VAR_ECU	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Voltage for ECU Lot detection					
PUT_BAS	O	0...3FFH	0...4.99511	4.8828e-3	[V]
manifold air pressure up throttle (turbo)					
V_FUP_EFP	O	8000...7FFFH	-5...4.99984	0.1526e-3	[V]
Low fuel pressure EFP sensor raw acquisition					
V_POIL	V/O	0...3FFH	0...4.99511	4.8828e-3	[V]
Oil pressure sensor raw acquisition					
V_DMTL	O	0...3FFH	0...4.99511	4.8828e-3	[V]
Diagnosis module tank leakage currency raw acquisition					

Direct analog inputs

Name	Acquisition rate (msec)	Port - Channel	Remark
V_IGK_BAS	1	2-01	
VB_BAS	1	3-01	
V_LS_UP_1	10	2-06	
V_LS_UP_2	10	2-07	
V_AMP	100	-	
V_LS_DOWN_1	10	2-20	
V_LS_DOWN_2	10	2-19	
V_PVS_1	10	1-20	
V_PVS_2	10	1-07	
VCC_PVS_1	10	1-11	only input data available
VCC_PVS_2	10	1-24	only input data available
V_TPS_1	5 and segmentsync.	5-37	
V_TPS_2	5 and segmentsync.	5-36	
V_VAR_ECU	100	-	Voltage for ECU LOT detection
V_PVS_1_MON	40	1-20	
V_PVS_2_MON	40	1-07	
V_TPS_1_MON	40	5-37	
V_TPS_2_MON	40	5-36	
V_SOF_SWI	10	2-03	Sportschalter
V_SOF_SWI_MON	10	2-03	
MAP_BAS	1	5-33	
V_FUP		7-09	
V_FUP_EFP	1	5-34	
V_POIL	10	7-03	Oil pressure sensor
V_DMTL	10	2-16	tank leakage sensor

Additional direct inputs for LV_VAR_TCHA = 1 (N54)

PUT_BAS	1	5-10	

60.4 Logic inputs

60.4.1 Standard management of logic inputs

FUNCTION DESCRIPTION:

General information:

Logic inputs are inputs whose state is acquired by a µP port read.
Logic inputs which are connected to a pull-up inside the ECU are valid only when VBR is present. Without VBR these inputs are read as a low level.

Initialization:

See general Initialization chapter.

Logic state:

The input is “ON” when the voltage level is high at the ECU connector and is “OFF” when the voltage level is low at the ECU connector.

Recurrence:

Logical inputs are read every 10 ms with a jitter of 1 ms maximum.

Filtering of the interferences:

When the state of the logical input is changed, the acquistion is checked by another acquisition after 40 µs delay. If the new acquisition is not the same the change is ignored.
This filtering is proposed as a base in order to provide a more reliable information. In case of bigger interferences or bounces, a complementary application filtering should be added.

Diagnostic:

No diagnostic is done on logic inputs into basic software.

60.5 Logic inputs

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
LV_IM_CS_PN	O/V	0...1H	0...1	1	-
clutch switch					
LV_IM_BTS	O/V	0...1H	0...1	1	-
brake test switch					
LV_IM_BLS	O/V	0...1H	0...1	1	-
brake light switch					
LV_IM_BTS_MON	O	0...1H	0...1	1	-
brake test switch for Monitoring					
LV_IM_BLS_MON	O	0...1H	0...1	1	-
brake light switch for Monitoring					

List of logic Inputs

.

Name	Port - Channel	Logic state
LV_PIN_CS_PN	E_S_KUP	Positive Logic
LV_PIN_BTS	E_S_BLTS	Positive Logic
LV_PIN_BLS	E_S_BLS	Positive Logic

60.5.1 List of logic outputs

Input data:

LV_RLY_MAIN	LV_CONF_DMTL	LV_HDMTL_ON	LV_VAR_RLY_ST
LV_EBOX_CFA	LV_ACCOUT_RLY	LV_DMTL_PUMP	LV_DMTLS
LV_RLY_ST	LV_SOF	LV_EF	LV_SWI_AEB
LV_RLY_CRCV_HEAT	LV_VAR_RLY_ACCOUT		

Description:

Name	ECU Pin Name	Definition
Switched internal	A_S_HR	main relay
Switched internal	Internal	power latch
LV_EBOX_CFA	A_S_EBOXL	E-box cooling fan
LV_ACCOUT_RLY	A_S_KOREL	air conditioning compressor relay
LV_RLY_ST	A_S_START	starter relay
LV_EF	A_S_AKL	exhaust flap, only if CONF_EF_B_KL61 = 0
LV_SWI_AEB	A_S_MLA	Automatic engine brackets
LV_SOF	A_S_ESK	sound flap
LV_RLY_CRCV_HEAT	A_T_PTC	crankcase ventilation heater relay
LV_RLY_MAIN	A_S_HR	main relay

Additional for LV_CONF_DMTL = 1

Name	ECU Pin Name	Definition
LV_DMTL_PUMP	A_S_DMTLP	tank leakage detection motor
LV_DMTLS	A_S_DMTLV	tank leakage detection vent
LV_HDMTL_ON	A_S_DMTLH	tank leakage detection heater

General information:

As there are no free logical output stages left for the activation/deactivation of the crankcase ventilation heater relay and the sound flap, PWM output stages are used for this purpose. To keep changes/efforts as lean as possible the pins which were supposed to be used for a PWM-controlled PTC-heater/sound flap are now used for the crankcase ventilation heater relay/sound flap control, thus resulting in an "incorrect" naming of the pins themself (A_T_PTC/A_T_ESK).

60.5.2 Definition of logic outputs

60.5.2.1 LV_LS_UP/DOWN_0/1

The logic outputs for the lambda sensor input resistors are set corresponding to switch SW1 described in chapter ‘System Variables, Upstream/Downstream oxygen sensor internal resistance determination’.

60.5.2.2 Logic outputs controlled by learning algorithm

ACC-Relay:

If LV_VAR_RLY_ACCOUT = 0

then	the related PIN is controlled by the learning algorithm (see chapter “General: Variant Coding/Automaticallly learnt variants via powerstage-dignosis”)
else	the PIN is controlled by LV_ACCOUT_RLY

Starter-Relay:

If	LV_VAR_RLY_ST = 0
then	the related PIN is controlled by the learning algorithm (see chapter “General: Variant Coding/Automaticallly learnt variants via powerstage-dignosis”)
else	the PIN is controlled by LV_RLY_ST

Crankcase ventilation heater relay:

As there is a 100% installation in all variants of NG6 engines the Pin is always controlled by LV_RLY_CRCV_HEAT.

60.5.3 Engine speed signal ESS

The engine speed signal ESS (ECU Pin A_F_TD) is a tooth-fixed event. The ESS signal is set at the teeth 5, 25, 45, 65, 85, 105 and reset at the teeth 15, 35, 55, 75, 95, 115. The artificially generated teeth are used in the CAM limp home. Since the cylinder group assignment for the output of the ESS signal is not decisive, the CRK limp home has no influence on the output of the ESS signal.

60.6 Variant Coding diagnosis

60.6.1 Diagnosis of successful BMW coding of variants via KWP

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
LV_ERR_VAR_COD	V/O/S	0...1H	0...1	1	[-]
Configuration variant coding error					
ERR_SYM_VAR_COD	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
symptoms of variant coding error					
LV_CDN_DIAG_VAR_COD	V/O	0...1H	0...1	1	[-]
condition for diagnosis of variant coding					
LV_END_DIAG_VAR_COD	V/O	0...1H	0...1	1	[-]
diagnosis of variant coding ended					

Input data:

LV_IGK	VIN_SHO[7]		

FUNCTION DESCRIPTION:

General information:

If not disabled by C_IDX_COD_CONV, this function checks the successful BMW coding of variants via KWP. Therefore the CAL-ID (identifier of the calibration- dataset) which is actual programmed in Flash memory is compared with the CAL-ID in the E2PROM. The CAL-ID is written to E2PROM memory only by receiving the tester command $3B_90 - Kurze Fahrgestellnummer schreiben. The receiving of this command is interpreted as successful coding.
If the two Call-IDs are not equal, a error is written into error memory. This error can only be erased by clearing the error memory.

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─→   -                                          (= SYM_0)
 │ │ └───→   -                                          (= SYM_1)
 │ └─────→   Coding data in E2PROM lost                 (= SYM_2)
 └───────→   New calibration data might not be plausible (= SYM_3)
```

Application conditions:

Initialisation: all output data are set to 0

at clearing error memory

Recurrence: 1000 ms

Activation: LV_IGK = 1 **and** LV_ERR_VAR_COD = 0

Deactivation: not activation

Formula section:

```
If(1) C_IDX_COD_CONV > 0
then(1)     LV_CDN_DIAG_VAR_COD = 1
            LV_END_DIAG_VAR_COD =  1
    if(2)   (all elements of VIN_SHO[7] (in RAM) = 0 or FFh) OR
            CRC failure E2PROM
    then(2)         LV_ERR_VAR_COD =  1
                    ERR_SYM_VAR_COD  = SYM_2
    else(2)         if(3)   CAL-ID in Flash memory ≠ CAL-ID in E2PROM
                    then(3)         LV_ERR_VAR_COD =  1
                                    ERR_SYM_VAR_COD = SYM_3
                    endif(3)
    endif(2)
endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_IDX_COD_CONV	1	0...FFH	0...255	1	[-]
BMW coding index to disable check of successful BMW coding of variants via KWP					

60.6.2 Diagnosis of incorrect engine power variant

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_VEH_POW_VAR	V/O	0...1H	0...1	1	[-]
Present error of wrong or unknown power variant					
LV_CDN_DIAG_VEH_POW_VAR	V/O	0...1H	0...1	1	[-]
Diagnosis condition wrong or unknown power variant					
ERR_SYM_VEH_POW_VAR	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected symptom wrong or unknown power variant					
LV_END_DIAG_VEH_POW_VAR	V/O	0...1H	0...1	1	[-]
End of diagnosis wrong or unknown power variant					
LV_POW_CLAS_VEH_CAN_REQ	V/O	0...1H	0...1	1	[-]
LV to request CAN signal "vehicle type"					
T_VEH_POW_VAR_ACT	V	0...FFH	0...25.5	0.1	[s]
time since function "check of power variant" activation					
T_POW_CLAS_VEH_CAN_REQ	V	0...FFH	0...25.5	0.1	[s]
time since request of CAN signal "vehicle type"					
CTR_ABC_VEH_POW_VAR	V	0...FFH	0...255	1	[-]
antibounce counter error of wrong or unknown power variant					

Input data:

LV_IGK	LV_POW_CLAS_VEH_CAN_RCV	STATE_POW_CLAS_VEH
POW_CONF_IDX_EXT_REQ	LV_ERR_CAN_BOFF	LV_VAR_BN

General information:

If not disabled by LC_VEH_POW_VAR_ERR_DIS = 1, this function checks the plausibility of the programmed engine power variant.

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └→   -                                              (= SYM_0)
 │ │ └───→   -                                              (= SYM_1)
 │ └─────→   Timeout CAN signal                             (= SYM_2)
 └───────→   Programmed engine power variants are not identical   (= SYM_3)
```

Application conditions:

Initialisation: according ABC filter Type **STD_INI** ; all other output data are set to 0 at LV_IGK 0→1 **or** reset **or** at clearing error memory

Recurrence: 100 ms

Activation: LV_IGK = 1 **and** LV_ERR_VEH_POW_VAR = 0
and LV_ERR_CAN_BOFF = 0

Deactivation: not activation

Formula section:

```
If(1)    LC_VEH_POW_VAR_ERR_DIS = 0
then(1)      T_VEH_POW_VAR_ACT  =  T_VEH_POW_VAR_ACT$_{n-1}$ + 100 ms ,
                                   limited to 25.5 s
             LV_CDN_DIAG_VEH_POW_VAR  =  1
  if(2)    T_VEH_POW_VAR_ACT  $\geq$  C_T_MIN_POW_CLAS_VEH_CAN_REQ
  then(2)
    if(2a)  LV_VAR_BN = 0
    then(2a)
      if(2b)   C_STATE_POW_CLAS_VEH $\neq$ POW_CONF_IDX_EXT_REQ
      then(2b)      ERR_SYM_VEH_POW_VAR  =  SYM_3
                    LV_ERR_VEH_POW_VAR  =  1
                    LV_END_DIAG_VEH_POW_VAR  =  1
      else(2b)
                    ERR_SYM_VEH_POW_VAR  =  NO_SYM
                    LV_END_DIAG_VEH_POW_VAR  =  1
      endif(2b)
    else(2a)
      if(3)   LV_POW_CLAS_VEH_CAN_RCV = 1
      then(3)
        if(4a)    STATE_POW_CLAS_VEH $\neq$ POW_CONF_IDX_EXT_REQ
        then(4a)    ERR_SYM_VEH_POW_VAR  =  SYM_3
                    LV_ERR_VEH_POW_VAR  =  1
                    LV_END_DIAG_VEH_POW_VAR  =  1
        else(4a)
          if(5)    STATE_POW_CLAS_VEH $\neq$ C_STATE_POW_CLAS_VEH
          then(5)   ERR_SYM_VEH_POW_VAR  =  SYM_3
                    LV_ERR_VEH_POW_VAR  =  1
                    LV_END_DIAG_VEH_POW_VAR  =  1
          else(5)
                    ERR_SYM_VEH_POW_VAR  =  NO_SYM
                    LV_END_DIAG_VEH_POW_VAR  =  1
          endif(5)
        endif(4a)
      else(3)
        if(4b)    T_POW_CLAS_VEH_CAN_REQ = C_T_MAX_POW_CLAS_VEH_CAN_RCV
        then(4b)    ERR_SYM_VEH_POW_VAR  =  SYM_2
                    CTR_ABC_VEH_POW_VAR = CTR_ABC_VEH_POW_VAR$_{n-1}$ + 1
                    T_POW_CLAS_VEH_CAN_REQ  =  0
                    LV_POW_CLAS_VEH_CAN_REQ  =  0
          if(6) CTR_ABC_VEH_POW_VAR = C_ABC_MAX_POW_CLAS_VEH_ERR
          then(6)   LV_ERR_VEH_POW_VAR  =  1
                    LV_END_DIAG_VEH_POW_VAR  =  1
          endif(6)
        else(4b)    LV_POW_CLAS_VEH_CAN_REQ  =  1
                    T_POW_CLAS_VEH_CAN_REQ  =  T_POW_CLAS_VEH_CAN_REQ$_{n-1}$
                                               + 100 ms
        endif(4b)
      endif(3)
```

```
        endif(2a)
    endif(2)
endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MIN_POW_CLAS_VEH_CAN_REQ	1	0...FFH	0...25.5	0.1	[s]
delay time before check of power variant coding					
C_T_MAX_POW_CLAS_VEH_CAN_RCV	1	0...FFH	0...25.5	0.1	[s]
timeout for receiving LV_POW_CLAS_VEH_CAN_RCV					
C_STATE_POW_CLAS_VEH	1	0...3H	0...3	1	[-]
calibrated status vehicle power class					
C_ABC_MAX_POW_CLAS_VEH_ERR	1	1...FFH	1...255	1	[-]
number of timeouts for antibounce of LV_ERR_VEH_POW_VAR					
LC_VEH_POW_VAR_ERR_DIS	1	0...1H	0...1	1	[-]
Logical constant to disable check of power variant coding					

61 Throttle

Table of Contents

Index

61.1 THRO - Requirements for infrastructure (ETC)

Input data:

LV_RLY_MTC_HEAT			

Export actions:

ACTION_INFR_SetMTCHEAT(IN <lv_rly_mtc_heat>**)**
This action sets the output pin for the “MTC_HEAT Switch” **ACTION_INFR_GetElDiagMTCHEAT(**OUT <Cdn_diag_mtcheat>, OUT <Err_diag_mtcheat>**)** This action reads the failure and condition information for a symptom of the MTCHEAT actuator power stage.

Description for actions:

ACTION_INFR_SetMTCHEAT (IN <lv_rly_mtc_heat> **)**					
This action sets the MTCHEAT driver					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
lv_rly_mtc_heat	IN	0…1H	0…1	1	-
MTCHEAT driver pin level					

ACTION_INFR_GetElDiagMTCHEAT(OUT <Cdn_diag_mtcheat>, OUT <Err_diag_mtcheat>)					
This action reads the failure and condition information for a symptom of MTCHEAT power stage. The readout of the power stage is performed autonomous and the information is gathered. When calling this Action the information inside the Infrastructure is reset after returning the OUT parameters.					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
Cdn_diag_mtcheat	OUT	0 ... 7H	0 ... 7	1	-
Diagnosis condition for symptom: Bit 0: Set, if diagnosis condition for symptom SCP (SYM_0) is fulfilled Bit 1: Set, if diagnosis condition for symptom SCG (SYM_1) is fulfilled Bit 2: Set, if diagnosis condition for symptom OC (SYM_2) is fulfilled (all combinations are possible)					
Err_diag_mtcheat	OUT	0 ... 7H	0 ... 7	1	-
Raw value of error symptom: Bit 0: Set, if raw value of error symptom SCP (SYM_0) is set Bit 1: Set, if raw value of error symptom SCG (SYM_1) is set Bit 2: Set, if raw value of error symptom OC (SYM_2) is set (all combinations are possible)					
Raw value of error symptom. The relevant bit is set, if the error has been detected.					

FUNCTION DESCRIPTION:

Requirements for ACTION_INFR_SetMTCHEAT:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**

lv_rly_mtc_heat	Not relevant	Not relevant	1	Not relevant	-

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

General information:

ACTION_INFR_GetElDiagMTCHEAT() returns the result of the electrical diagnosis of MTCHEAT actuator power stage.

_ The device readout is performed autonomous by the Infrastructure each 10ms.
_ The error informations of every symptom is gathered in the Infrastructure (or-ed symptom) until the Application reads out the information by calling ACTION_INFR_GetElDiagMTCHEAT().
_ After having read out the information by calling ACTION_INFR_GetElDiagMTCHEAT(), the data inside the Infrastructure is reset. Resetting of Cdn_diag_mtcheat avoids ambigious results in case of too short calling reccurrence of ACTION_INFR_GetElDiagMTCHEAT(): Reset Cdn_diag_mtcheat indicates, that the gathering of the information is not completely finished.

Requirements for ACTION_INFR_GetElDiagMTCHEAT:

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
Cdn_diag_mtcheat	-	-	<bit coded>	Err_diag_mtcheat	Diagnosis condition for each symptom **bit 0:** diagnosis condition for symptom SCP (SYM_0) **bit 1:** diagnosis condition for symptom SCG (SYM_1) **bit 2:** diagnosis condition for symptom OC (SYM_2) The relevant bit is set, if the condition for a valid diagnosis is fulfilled
Err_diag_mtcheat	-	-	<bit coded>	Cdn_diag_mtcheat	Bitcoded result of each symptom **bit 0:** raw value of error symptom SCP (SYM_0) **bit 1:** raw value of error symptom SCG (SYM_1) **bit 2:** raw value of error symptom OC (SYM_2) The relevant bit is set, if the raw error has been detected.

Diagnosis: ACTION_INFR_GetElDiagMTCHEAT returns the electric diagnosis for MTCHEAT power stage

Coincidence requirements: no coincidence requirements to other events

61.2 AGGR THRO adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
V_TPS_GAIN_1	-	0...3FFH	0...4.99511	4.8828e-3	[V]
Throttle position sensor 1 multiplied with 4 raw acquisition					

FUNCTION DESCRIPTION:

General information:

Adaptation module throttle variables

Application conditions:

Initialisation: all = 0 at reset

Recurrence: same as input values

Activation: every ECU operating state

Deactivation: -

61.3 Determination of the Throttle Position

Output data:

Name	V/S	Hex. limits	Phys. limits	Resol.	Unit
TPS_AV	V/O	0...3FFFH	0...119,5	0,0073	°TPS
Opening angle of the throttle valve					
TPS_AV_CTL	V/O	C000H...3FFFH	-119.5...119,5	0,0073	°TPS
Control opening angle of the throttle valve					
TPS_AV_1	V/O	0...3FFFH	0...119,5	0,0073	°TPS
Opening angle of the throttle valve of TPS-Poti 1					
TPS_AV_2	V/O	0...3FFFH	0...119,5	0,0073	°TPS
Opening angle of the throttle valve of TPS-Poti 2					
TPS_AV_CTL_1	V/O	C000H...3FFFH	-119.5...119,5	0,0073	°TPS
Control opening angle of the throttle valve of TPS-Poti 1					
TPS_AV_CTL_2	V/O	C000H...3FFFH	-119.5...119,5	0,0073	°TPS
Control opening angle of the throttle valve of TPS-Poti 2					
TPS_AV_GAIN_1	V	0...3FFFH	0...119,5	0,0073	°TPS
Opening angle of the throttle valve of amplified TPS-Poti 1					
TPS_AV_CTL_GAIN_1	V	C000H...3FFFH	-119.5...119,5	0,0073	°TPS
Control opening angle of the throttle valve of amplified TPS-Poti 1					
TPS_LIH	V/O	0...3FFFH	0...119,5	0,0073	°TPS
Limp home position of the throttle valve					
LV_TPS_GAIN_ACT_1	V/O	0...1H	0...1	1	-
Logical variable for TPS_AV based on amplified TPS 1					
V_TPS_GAIN_TOL_ENA_1	V	0...3FFH	0...5,0	0,0049	Volt
TPS1-voltage to switch from amplified to non-amplified sensor signal					
V_TPS_GAIN_BOL_ENA_1	V	0...3FFH	0...5,0	0,0049	Volt
TPS1-voltage to switch from non-amplified to amplified sensor signal					
TPS_GRD	V/O	0...FFH	0...2987.5	11.72	°TPS/s
Throttle position gradient, absolute value					
LV_TPS_GRD_UP	V/O	0...1H	0...1	1	-
Logical variable for throttle position gradient direction					
TPS_SEG_1	O/V	0...3FFFH	0...119.5	0.0073	°TPS
Segment synchronous measured throttle valve position channel 1					
TPS_SEG_GAIN_1	O/V	0...3FFFH	0...119.5	0.0073	°TPS
Segment synchronous measured throttle valve position amplified channel 1					
TPS_SEG_2	O/V	0...3FFFH	0...119.5	0.0073	°TPS
Segment synchronous measured throttle valve position channel 2					
TPS_SEG	O/V	0...3FFFH	0...119.5	0.0073	°TPS
Segment synchronous measured throttle valve position					

Input data:

V_TPS_1	V_TPS_2	V_TPS_GAIN_1	V_TPS_AD_EL_BOL_2
V_TPS_AD_LIH_1	V_TPS_AD_OFS_GAIN_1	TPS_AD_SLOP_GAIN_1	V_TPS_AD_EL_BOL_1
V_TPS_AD_LIH_2	STATE_TPS_DIAG	TPS_LIH_INI	TPS_LIH_1
TPS_LIH_2	STATE_ETC_LIH		

61.3.1 Standardization of the TPS-Voltage

FUNCTION DESCRIPTION:

General information:

With the help of the determined adaptation values, the TPS-Position is standardized using the standard slope of the throttle position sensors C_TPS_SLOP.
For a better control of smaller opening angles the amplified poti 1 is used (TPS_AV_GAIN_1).
To calculate the controller difference at the lower stop a control opening angle of the throttle value with positive/negative physical limits is needed (TPS_AV_CTL).

Application conditions:

Initialisation: TPS_LIH = TPS_LIH_INI

Recurrence: 5ms

Activation: at every operating state

Formula section:

actual value 1&2:

TPS_AV_1 = (V_TPS_1_FIL – V_TPS_AD_EL_BOL_1) * C_TPS_SLOP

TPS_AV_2 = (V_TPS_AD_EL_BOL_2 – V_TPS_2_FIL) * C_TPS_SLOP

actual contol value 1&2:

TPS_AV_CTL_1 = (V_TPS_1_FIL – V_TPS_AD_EL_BOL_1) * C_TPS_SLOP

TPS_AV_CTL_2 = (V_TPS_AD_EL_BOL_2 – V_TPS_2_FIL) * C_TPS_SLOP

actual value: amplified poti 1:

TPS_AV_GAIN_1 =

$$\left(\frac{V\_TPS\_GAIN\_1\_FIL - V\_TPS\_AD\_OFS\_GAIN\_1}{TPS\_AD\_SLOP\_GAIN\_1} - V\_TPS\_AD\_EL\_BOL\_1\right) * C\_TPS\_SLOP$$

actual control value: amplified poti 1:

TPS_AV_CTL_GAIN_1 =

$$\left(\frac{V\_TPS\_GAIN\_1\_FIL - V\_TPS\_AD\_OFS\_GAIN\_1}{TPS\_AD\_SLOP\_GAIN\_1} - V\_TPS\_AD\_EL\_BOL\_1\right) * C\_TPS\_SLOP$$

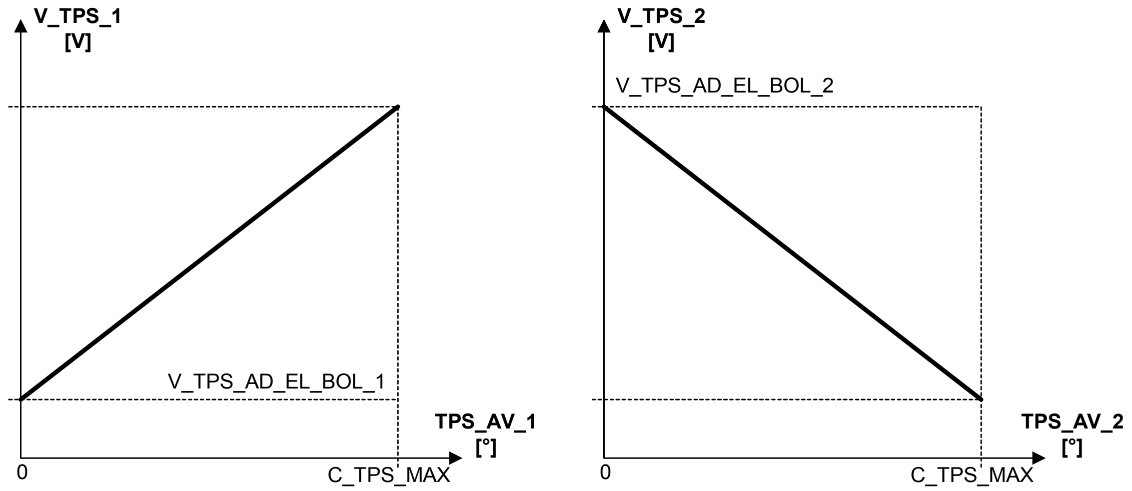


Fig. 1: **Standardization of the TPS-Voltage**

61.3.2 Selection of actual value TPS_AV / TPS_AV_CTL

Description:

The selection of actual value TPS_AV / TPS_AV_CTL depends on the hysteresis for amplifed/non-amplified poti 1 and the status of TPS/ETC diagnosis.

Formula section:

Calculation hysteresis band:

V_TPS_GAIN_TOL_ENA_1 = 5V / TPS_AD_SLOP_GAIN_1 – C_V_TPS_SWI_GAIN_1

V_TPS_GAIN_BOL_ENA_1 = V_TPS_GAIN_TOL_ENA_1 – C_V_TPS_HYS_GAIN_1

Hysteresis for amplifed/non-amplified poti 1:

Hysteresis for use of amplified / non-amplified TPS 1:

```
if      V_TPS_1 < V_TPS_GAIN_BOL_ENA_1          and
        LV_TPS_GAIN_ACT_1 = 0
then    LV_TPS_GAIN_ACT_1 = 1
    elseif    V_TPS_1 > V_TPS_GAIN_TOL_ENA_1    and
              LV_TPS_GAIN_ACT_1 = 1
    then      LV_TPS_GAIN_ACT_1 = 0
    endif
endif
```

Selection in errorfree case:

```
TPS1 &TPS2 errorfree, no TPS plausibility error:
if        STATE_TPS_DIAG = TPS_NO_ERROR
then
          if     LV_TPS_GAIN_ACT_1 = 1
          inside operation range of amplified TPS1:
          then   TPS_AV      = TPS_AV_GAIN_1
                 TPS_AV_CTL = TPS_AV_CTL_GAIN_1
                 TPS_LIH     = TPS_LIH_1
          outside operation range of amplified TPS1:
          else   TPS_AV      = TPS_AV_1
                 TPS_AV_CTL = TPS_AV_CTL_1
                 TPS_LIH     = TPS_LIH_1
          endif
endif
```

Selection in error-cases:

```
TPS1 & TPS2 erroneous or MTC without power:
if        STATE_TPS_DIAG = TPS_DBL_ERROR          or
          STATE_ETC_LIH = ETC_LIH_1               or
          STATE_ETC_LIH = ETC_LIH_2_REV           or
          STATE_ETC_LIH = ETC_LIH_2
then      TPS_AV     = C_TPS_MAX
          TPS_LIH    = TPS_LIH_INI

TPS2 erroneous:
elseif    STATE_TPS_DIAG = TPS_LIH_1
then
          if     LV_TPS_GAIN_ACT_1 = 1
          inside operation range of amplified TPS1:
          then   TPS_AV      = TPS_AV_GAIN_1
                 TPS_AV_CTL = TPS_AV_CTL_GAIN_1
                 TPS_LIH     = TPS_LIH_1
          outside operation range of amplified TPS1:
          else   TPS_AV      = TPS_AV_1
                 TPS_AV_CTL = TPS_AV_CTL_1
                 TPS_LIH     = TPS_LIH_1
          endif
```

TPS1 erroneous:

```
elseif      STATE_TPS_DIAG = TPS_LIH_2
then        TPS_AV      = TPS_AV_2
            TPS_AV_CTL  = TPS_AV_CTL_2
            TPS_LIH     = TPS_LIH_2
```

TPS plausibility error:

```
elseif      STATE_TPS_DIAG = TPS_LIH_MAX
then        TPS_AV      = MAX(TPS_AV_1; TPS_AV_2)
            TPS_AV_CTL  = MAX(TPS_AV_CTL_1; TPS_AV_CTL_2)
            TPS_LIH     = TPS_LIH_INI
Endif
```

61.3.3 TPS gradient limitation

Description:

To eliminate eventually malfunctions the potentiometer values are verified by a gradient limitation. Is the difference between the old and new potentiometer voltage value exceeding the gradient threshold then keeps the old poti value. If by the next measuring the gradient limitation exceeded again, then the new poti value will adopt.

Formula section:

```
IF          $|V\_TPS\_X_n - V\_TPS\_X_{n-1}| \le C\_V\_TPS\_GRD\_MAX$
Then                $V\_TPS\_X\_FIL = V\_TPS\_X_n$

Elseif      $|V\_TPS\_X_n - V\_TPS\_X_{n-1}| > C\_V\_TPS\_GRD\_MAX$          and
            $|V\_TPS\_X_{n-1} - V\_TPS\_X_{n-2}| > C\_V\_TPS\_GRD\_MAX$

    Then            $V\_TPS\_X\_FIL = V\_TPS\_X_n$
    Else            $V\_TPS\_X\_FIL = V\_TPS\_X\_FIL_{n-1}$
Endif
```

```
IF $|V\_TPS\_GAIN\_1_n - V\_TPS\_GAIN\_1_{n-1}| \leq C\_V\_TPS\_GAIN\_GRD\_MAX$
  Then     $V\_TPS\_GAIN\_1\_FIL_n = V\_TPS\_GAIN\_1_n$
  Elseif   $|V\_TPS\_GAIN\_1_n - V\_TPS\_GAIN\_1_{n-1}| > C\_V\_TPS\_GAIN\_GRD\_MAX$ and
           $|V\_TPS\_GAIN\_1_{n-1} - V\_TPS\_GAIN\_1_{n-2}| > C\_V\_TPS\_GAIN\_GRD\_MAX$
     Then  $V\_TPS\_GAIN\_1\_FIL_n = V\_TPS\_GAIN\_1_n$
     Else  $V\_TPS\_GAIN\_1\_FIL_n = V\_TPS\_GAIN\_1\_FIL_{n-1}$
  Endif
```

61.3.3.1 Calculation of throttle position gradient: (TPS_GRD)

Description:

The gradient is calculated as (absolute value of) difference from the actual throttle value and the throttle value i-4 (that is the throttle value which was actual 40ms before). Therefore the converted throttle values are written into a ring-buffer with 4 values.

Application conditions:

Recurrence: 10ms

Formula section:

$TPS\_GRD = | (TPS\_AV_i - TPS\_AV_{i-4}) * 25 |$

The throttle gradient direction is calculated as follows

```
If     $(TPS\_AV_i - TPS\_AV_{i-4}) < 0$
then   $LV\_TPS\_GRD\_UP = 0$     (negative throttle gradient)
else   $LV\_TPS\_GRD\_UP = 1$     (positive throttle gradient)
endif
```

61.3.4 Segment synchronous throttle valve position: (TPS_SEG)

Description:

The segment synchronous TPS is needed as input for the intake manifold model. The calculation is carried out segment synchronously.With the help of the adaptation values the TPS-position is standardized to a range between 0° and C_TPS_MAX degree.

Formula section:

$TPS\_SEG\_1 = (V\_TPS\_1 - V\_TPS\_AD\_EL\_BOL\_1) * C\_TPS\_SLOP$

$TPS\_SEG\_2 = (V\_TPS\_AD\_EL\_BOL\_2 - V\_TPS\_2) * C\_TPS\_SLOP$

$TPS\_SEG\_GAIN\_1 =$

$$\left(\frac{\mathrm{V\_TPS\_GAIN\_1 - TPS\_AD\_OFS\_GAIN\_1}}{\mathrm{TPS\_AD\_SLOP\_GAIN\_1}} - \mathrm{V\_TPS\_AD\_EL\_BOL\_1}\right) * \mathrm{C\_TPS\_SLOP}$$

0° ≤ TPS_SEG_x ≤ C_TPS_MAX.

Selection in errorfree case:

TPS1 &TPS2 errorfree, no TPS plausibility error:

```
if          STATE_TPS_DIAG = TPS_NO_ERROR
then
            if      LV_TPS_GAIN_ACT_1 = 1
            inside operation range of amplified TPS1:
            then    TPS_SEG     = TPS_SEG_GAIN_1
            outside operation range of amplified TPS1:
            else    TPS_SEG     = TPS_SEG_1
            endif
endif
```

Selection in error-cases:

TPS1 & TPS2 erroneous:

```
if          STATE_TPS_DIAG = TPS_DBL_ERROR
            STATE_ETC_LIH = ETC_LIH_1                    or
            STATE_ETC_LIH = ETC_LIH_2_REV                or
            STATE_ETC_LIH = ETC_LIH_2
then        TPS_SEG = TPS_LIH_INI
```

TPS2 erroneous:

```
elseif      STATE_TPS_DIAG = TPS_LIH_1
then
            if      LV_TPS_GAIN_ACT_1 = 1
            inside operation range of amplified TPS1:
            then    TPS_SEG     = TPS_SEG_GAIN_1
            outside operation range of amplified TPS1:
            else    TPS_SEG     = TPS_SEG_1
      endif
```

TPS1 erroneous:

```
elseif      STATE_TPS_DIAG = TPS_LIH_2
then     TPS_SEG  = TPS_SEG_2
```

TPS plausibility error:

```
else     TPS_SEG  = MAX( TPS_SEG_1; TPS_SEG_2 )
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_TPS_SWI_GAIN_1	1	0..3FFH	0...5.0	0.0049	Volt
Offset on maximum nominal voltage for switch amplified/non-amplified TPS1					
C_V_TPS_HYS_GAIN_1	1	0..3FFH	0...5.0	0.0049	Volt
Hysteresis treshold for switch amplified/non-amplified TPS1					
C_TPS_MAX	1	0...3FFFH	0...119,5	0,0073	°TPS
Maximum mechanical opening angle of the throttle valve in degree					
C_TPS_SLOP	1	0 ...FFFFH	0...48	7.3e-4	°TPS/V
Nominal value of the slope of potis 1 and 2					
C_V_TPS_GRD_ MAX	1	0..3FFH	0...5.0	0.0049	Volt
TPS voltage gradient limitation					
C_V_TPS_GAIN_GRD_ MAX	1	0..3FFH	0...5.0	0.0049	Volt
TPS voltage gradient limitation					

61.4 Throttle position sensor diagnosis

61.4.1 Diagnosis of the admissible voltage ranges V_TPS_1 / V_TPS_2

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TPS_1	O/V	0...1H	0...1	1	[-]
Electrical TPS error, channel 1					
LV_CDN_DIAG_TPS_1	V	0...1H	0...1	1	[-]
Electrical TPS 1 diagnosis condition					
ERR_SYM_TPS_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom TPS1 diagnosis					
LV_END_DIAG_TPS_1	V	0...1H	0...1	1	[-]
End of diagnosis TPS1 diagnosis					
LV_ERR_TPS_2	O/V	0...1H	0...1	1	[-]
Electrical error TPS channel 2					
LV_CDN_DIAG_TPS_2	V	0...1H	0...1	1	[-]
Electrical TPS 2 diagnosis condition					
ERR_SYM_TPS_2	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom TPS2 diagnosis					
LV_END_DIAG_TPS_2	V	0...1H	0...1	1	[-]
End of diagnosis TPS2 diagnosis					

Input data:

V_TPS_1	V_TPS_2	LV_IGK	

FUNCTION DESCRIPTION:

Description:

In the case of an error-free system, the potentiometer voltages must be within admissible limits. The following errors symptoms can be distinguished:

ERR_SYM_TPS_1:

```
|_|_|_|_|
 | | | └─> Short circuit to Vbatt or Open load     (= SYM_0)
 | | └───> Short circuit to Ground                 (= SYM_1)
 | └─────> -                                       (= SYM_2)
 └───────> -                                       (= SYM_3)
```

ERR_SYM_TPS_2:

```
|_|_|_|_|
 | | | └─> Short circuit to Vbatt                  (= SYM_0)
 | | └───> Short circuit to Ground or Open load    (= SYM_1)
```

```
 │ └────────> -                                  (= SYM_2)
 └──────────> -                                  (= SYM_3)
```

Application conditions:

Initialisation: according filtertype **STD_INI** *(all 0 at LV_IGK 0->1 or reset)*
Recurrence: 5ms
Activation:

```
If      LV_IGK = 1                     and
        LV_ERR_TPS_1/2 = 0             and
        LV_ERR_TPS_MAF_1/2 = 0         //calculation for channel 1 and 2 seperate
Then    LV_CDN_DIAG_TPS_1/2 = 1
Else    LV_CDN_DIAG_TPS_1/2 = 0
Endif
```

Formula section:

Calculation End of Diagnosis:

For calculation of LV_END_DIAG_TPS_1/2 see "Anti-bounce algorithm, calculation of the end of diagnosis".

Error detection TPS_1 short circuit to ground:

```
If      V_TPS_1  <  C_V_TPS_MIN_DIAG_1
Then    ERR_SYM_TPS_1 = SYM_1
        LV_ERR_TPS_1 = 1        //after debounce, remains active until next initialization
```

Error detection TPS_1 short circuit to Vbatt or open load:

```
Elself  V_TPS_1  >  C_V_TPS_MAX_DIAG_1
Then    ERR_SYM_TPS_1 = SYM_0
        LV_ERR_TPS_1 = 1        //after debounce, remains active until next initialization
Else    ERR_SYM_TPS_1 = NO_SYM
Endif
```

Error detection TPS_2 short circuit to ground or open load:

```
If      V_TPS_2  <  C_V_TPS_MIN_DIAG_2
Then    ERR_SYM_TPS_2 = SYM_1
        LV_ERR_TPS_2 = 1        //after debounce, remains active until next initialization
```

Error detection TPS_2 short circuit to VBatt:

```
Elself  V_TPS_2  >  C_V_TPS_MAX_DIAG_2
Then    ERR_SYM_TPS_2 = SYM_0
        LV_ERR_TPS_2 = 1        //after debounce, remains active until next initialization
Else    ERR_SYM_TPS_2 = NO_SYM
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TPS_1	1	0...FFH	0...255	1	[-]
Anti-bounce increment value TPS diagnosis					
C_ABC_MAX_TPS_1	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value TPS diagnosis					
C_ABC_INC_TPS_2	1	0...FFH	0...255	1	[-]
Anti-bounce increment value TPS diagnosis					
C_ABC_MAX_TPS_2	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value TPS diagnosis					
C_V_TPS_MIN_DIAG_1	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lower diagnostic voltage TPS 1					
C_V_TPS_MIN_DIAG_2	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Lower diagnostic voltage TPS 2					
C_V_TPS_MAX_DIAG_1	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Upper diagnostic voltage TPS 1					
C_V_TPS_MAX_DIAG_2	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Upper diagnostic voltage TPS 2					

61.4.2 Ratio – check between TPS values V_TPS_1 / V_TPS_2

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TPS_RATIO	O/V	0...1H	0...1	1	[-]
Logic variable set if TPS_RATIO error is present					
LV_CDN_DIAG_TPS_RATIO	V	0...1H	0...1	1	[-]
Diagnosis condition TPS_RATIO diagnosis					
ERR_SYM_TPS_RATIO	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom TPS_RATIO diagnosis					
LV_END_DIAG_TPS_RATIO	O/V	0...1H	0...1	1	[-]
End of diagnosis TPS_RATIO diagnosis					
LV_DET_ERR_TPS_RATIO	O/V	0...1H	0...1	1	[-]
Logic variable set if TPS_RATIO error is detected					

Input data:

LV_ST_END			

FUNCTION DESCRIPTION:

General information:

In the case of an error-free system, the Poti-voltages V_TPS_1 and V_TPS_2 must not differ by more than C_V_TPS_RATIO_HYS. The voltage V_TPS_2 is “inverted” in order to carry out the channel comparison.

Description:

If the threshold C_V_TPS_RATIO_HYS is exceeded, the TPS_RATIO error will be debounced.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
 │  │  │  └─> -                        (= SYM_0)
 │  │  └────> -                        (= SYM_1)
 │  └───────> -                        (= SYM_2)
 └──────────> TPS – Ratio exeeded      (= SYM_3)
```

Application conditions:

Initialisation: according filter type **STD_INI**

Recurrence: 5ms

Activation:

```
If          LV_IGK = 1
      and   LV_ST_END = 1
      and   LV_ERR_TPS_1/2 = 0          both channels must be errorfree
      and   LV_ERR_TPS_MAF_1/2 = 0
Then        LV_CDN_DIAG_TPS_RATIO = 1
Else        LV_CDN_DIAG_TPS_RATIO = 0
            LV_DET_ERR_TPS_RATIO = 0
Endif
```

Formula section:

```
If      | V_TPS_1 - (5[V] - V_TPS_2) |  >  C_V_TPS_RATIO_HYS
Then    ERR_SYM_TPS_RATIO = SYM_3          //anti bounce increment
        LV_DET_ERR_TPS_RATIO = 1
Else    ERR_SYM_TPS_RATIO = NO_SYM         //anti bounce decrement
        LV_DET_ERR_TPS_RATIO = 0
Endif
```

LV_ERR_TPS_RATIO , LV_END_DIAG_TPS_RATIO is calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TPS_RATIO	1	0...FFH	0...255	1	[-]
Anti-bounce increment value TPS ratio diagnosis					
C_ABC_MAX_TPS_RATIO	1	1...FFH	1...255	1	[-]
Anti-bounce maximum value TPS ratio diagnosis					
C_V_TPS_RATIO_HYS	1	0...3FFH	0...4.99511	4.8828e-3	[V]
Admissible deviation of DK angle					

61.4.3 TPS / TPS_SUB_DIAG plausibility check after RATIO-error is detected

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TPS_MAF_1	O/V	0...1H	0...1	1	[-]
TPS channel 1 is not plausible to MAF					
LV_CDN_DIAG_TPS_MAF_1	V	0...1H	0...1	1	[-]
Diagnosis condition TPS_MAF 1					
ERR_SYM_TPS_MAF_1	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom TPS_MAF 1					
LV_END_DIAG_TPS_MAF_1	V	0...1H	0...1	1	[-]
End of diagnosis TPS_MAF 1					
LV_ERR_TPS_MAF_2	O/V	0...1H	0...1	1	[-]
TPS channel 2 is not plausible to MAF					
LV_CDN_DIAG_TPS_MAF_2	V	0...1H	0...1	1	[-]
Diagnosis condition TPS_MAF 2					
ERR_SYM_TPS_MAF_2	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom TPS_MAF 1					
LV_END_DIAG_TPS_MAF_2	V	0...1H	0...1	1	[-]
End of diagnosis TPS_MAF 2					
TPS_DIF_DIAG_1	V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Difference between measured TPS_AV_1 and modelled TPS_SUB_DIAG					
TPS_DIF_DIAG_2	V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Difference between measured TPS_AV_2 and modelled TPS_SUB_DIAG					
TPS_DIF_DIAG_COR_1	O/V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Corrected value of TPS_DIF_DIAG_1					
TPS_DIF_DIAG_COR_2	O/V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Corrected value of TPS_DIF_DIAG_2					
TPS_SUM_DIF_DIAG_COR_1	O/V	0...FFFFH	0...32767.5	0.5	[°TPS]
Integrated value of TPS_DIF_DIAG_COR_1					
TPS_SUM_DIF_DIAG_COR_2	O/V	0...FFFFH	0...32767.5	0.5	[°TPS]
Integrated value of TPS_DIF_DIAG_COR_2					
TPS_SUM_DIF_DIAG_COR_1_OLD	-	0...FFFFH	0...32767.5	0.5	[°TPS]
Old value of TPS_SUM_DIF_DIAG_COR_1					
TPS_SUM_DIF_DIAG_COR_2_OLD	-	0...FFFFH	0...32767.5	0.5	[°TPS]
Old value of TPS_SUM_DIF_DIAG_COR_2					
TPS_DIF_DIAG_COR_MMV_1	V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Filtered corrected value of TPS_DIF_DIAG_COR_1					
TPS_DIF_DIAG_COR_MMV_2	V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Filtered corrected value of TPS_DIF_DIAG_COR_2					
T_TPS_DIF_INT	-	0...FFH	0...1.275	0.005	[s]
Integration timer for deviation between measured and calculated DK value					
T_DLY_TPS_DIAG_REQ_HOM_ACT	V	0...FFH	0...1.275	0.005	[s]
Delay timer to check the HOM mode after active TPS error					
T_MAX_TPS_DIF_DIAG	V	0...FFH	0...1.275	0.005	[s]
Integration timer for plausibility check in case of single channel error					

Input data:

STATE_TPS_DIAG	LV_TPS_SUB_DIAG_ERR	LV_TPS_SUB_DIAG_NOT_VLD	LV_HOM_ACT

TPS_AV_1	TPS_AV_2	TPS_SUB_DIAG	

FUNCTION DESCRIPTION:

General information:

This function is only performed if calibrated LC_USE_MAF_TPS_DIAG = 1.

If RATIO error is detected the algorithm is used to decide if channel one or two or both are not plausible. It is based on the modelled TPS value TPS_SUB_DIAG.

X = channel 1 and 2

Description:

Detection of TPS fist error:

The integration is done for C_T_TPS_DIF_INT if diagnosis conditions are active.

TPS_DIF_DIAG_X is the difference between the mesured value TPS_AV_X and the modelled TPS value TPS_SUB_DIAG.

TPS_DIF_DIAG_COR_X is the corrected value based on N, MAF_SP_TQI, PQ.

TPS_SUM_DIF_DIAG_COR_X is the integrated value of TPS_DIF_DIAG_COR_X.

Depending on TPS_SUM_DIF_DIAG_COR_X channel one or two or both are set as not plausible and the diagnosis is switched off.

The plausibility check cannot be performed if TPS_SUB_DIAG cannot be calculated in the right way because an active lambda-sensor fault or a load-sensor fault.
In this case, the two channels are declared as faulty when LV_DET_ERR_TPS_RATIO is set for C_T_TPS_DIF_INT.

Continuous TPS / TPS_SUB_DIAG monitoring for single channel

In the case of one channel is diagnosed as faulty, a plausibility check for the remaining channel to the other one is no longer possible. In this case, the substitute value TPS_DIF_DIAG_COR_X is used for the plausibility check while the engine is running. The difference between the measured TPS_AV value and the substitute value TPS is normalized and the mean value is established with a correlation constant.
The plausibility check cannot be performed when a mass air flow signal is missing; in this case, the channel is declared as faulty.
Error symptom definition see TPS-FIRST-ERROR diagnosis.

Check of the combustion mode with active TPS error

In order to perform the first / second error diagnosis the combustion-mode must be homogeneous, thus there's a delay timer which runs after a present Ratio / Single TPS error.

If the timer is elapsed without active HOM mode, a TPS double fault is set and the throttle is switched off.

Error symptoms are defined for this diagnosis function as:

ERR_SYM_TPS_MAF_1:

☐☐☐☐

```
 │ │ │ └─> -                            (= SYM_0)
 │ │ └───> -                            (= SYM_1)
 │ └─────> -                            (= SYM_2)
 └───────> TPS1 is not plausible        (= SYM_3)
```

ERR_SYM_TPS_MAF_2:

```
☐☐☐☐
 │ │ │ └─> -                            (= SYM_0)
 │ │ └───> -                            (= SYM_1)
 │ └─────> -                            (= SYM_2)
 └───────> TPS2 is not plausible        (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** clearing error memory

Recurrence: 5ms

Activation: LC_USE_MAF_TPS_DIAG = 1 **and**
STATE_TPS_DIAG = ! TPS_DBL_ERROR

Deactivation: LC_USE_MAF_TPS_DIAG = 0 **or**
STATE_TPS_DIAG = TPS_DBL_ERROR
set LV_CDN_DIAG_TPS_MAF_1/2 = 0

Formula section:

```
If(1)      LV_IGK = 1
    and    LV_ST_END = 1
    and    LV_TPS_SUB_DIAG_NOT_VLD = 0          //TPS_SUB_DIAG not valid
    and    LV_HOM_ACT = 1
Then(1)    //Main condition for Integration active
```

TPS_DIF_DIAG_X = Difference between measured and model value

TPS_DIF_DIAG_1 = ABS [TPS_AV_1 – TPS_SUB_DIAG]

TPS_DIF_DIAG_2 = ABS [TPS_AV_2 – TPS_SUB_DIAG]

TPS_DIF_DIAG_COR_X = Corrected difference value depending load / engine speed

TPS_DIF_DIAG_COR_1 = TPS_DIF_DIAG_1 *
IP_FAC_TPS_DIF_1__N__MAF_SP_TQI *
IP_FAC_TPS_DIF_2__PQ_SP

TPS_DIF_DIAG_COR_2 = TPS_DIF_DIAG_2 *
IP_FAC_TPS_DIF_1__N__MAF_SP_TQI *

IP_FAC_TPS_DIF_2__PQ_SP

```
If(2) //any TPS error active?
(LV_ERR_TPS_1 = 0 and LV_ERR_TPS_MAF_1 = 0)      or
(LV_ERR_TPS_2 = 0 and LV_ERR_TPS_MAF_2 = 0)
Then(2)
    If(3)
    (LV_ERR_TPS_1 = 1 or LV_ERR_TPS_MAF_1 = 1)    //TPS_1 faulty
    Then(3)
    LV_CDN_DIAG_TPS_MAF_2 = 1
    Call "calculation TPS_DIF_DIAG_COR_MMV_2"
        If(4)
        LV_TPS_SUB_DIAG_ERR = 1
        Then(4)
        ERR_SYM_TPS_MAF_2 = SYM_3
        LV_ERR_TPS_MAF_2 = 1
        LV_END_DIAG_TPS_MAF_2 = 1
        Elseif (4)
        (LV_TPS_SUB_DIAG_ERR = 0 and
        TPS_DIF_DIAG_COR_MMV_2 > C_TPS_DIF_DIAG_COR_MMV_MAX)
        Then(4)
            If(5)
            T_MAX_TPS_DIF_DIAG < C_T_MAX_TPS_DIF_DIAG
            Then(5)
            T_MAX_TPS_DIF_DIAG++                    //count debounce time
            Else(5)
            ERR_SYM_TPS_MAF_2 = SYM_3
            LV_ERR_TPS_MAF_2 = 1
            LV_END_DIAG_TPS_MAF_2 = 1
            Endif(5)
        Else(4)
        T_MAX_TPS_DIF_DIAG --
        Endif(4)
    Elseif(3)
    (LV_ERR_TPS_2 = 1 or LV_ERR_TPS_MAF_2 = 1)    //TPS_2 faulty
    Then(3)
    LV_CDN_DIAG_TPS_MAF_1 = 1
```

```
            Call "calculation TPS_DIF_DIAG_COR_MMV_1"
                If(4b)
                LV_TPS_SUB_DIAG_ERR = 1
                Then(4b)
                ERR_SYM_TPS_MAF_1 = SYM_3
                LV_ERR_TPS_MAF_1 = 1
                LV_END_DIAG_TPS_MAF_1 = 1
                Elseif(4b)
                (LV_TPS_SUB_DIAG_ERR = 0 and
                TPS_DIF_DIAG_COR_MMV_1 > C_TPS_DIF_DIAG_COR_MMV_MAX)
                Then(4b)
                    If(5b)
                    T_MAX_TPS_DIF_DIAG < C_T_MAX_TPS_DIF_DIAG
                    Then(5b)
                    T_MAX_TPS_DIF_DIAG ++                 //count debounce time
                    Else(5b)
                    ERR_SYM_TPS_MAF_1 = SYM_3
                    LV_ERR_TPS_MAF_1 = 1
                    LV_END_DIAG_TPS_MAF_1 = 1
                    Endif(5b)
                Else(4)
                T_MAX_TPS_DIF_DIAG --
                Endif(4b)
            Else(3)
            Call " First error detection"
            Endif(3)
        Else(2) Double TPS error, no diagnosis
        Endif(2)
    Else(1) //reset
    T_TPS_DIF_INT  = 0
    TPS_SUM_DIF_DIAG_COR_1 = 0
    TPS_SUM_DIF_DIAG_COR_2 = 0
    TPS_SUM_DIF_DIAG_COR_1_OLD = 0
    TPS_SUM_DIF_DIAG_COR_2_OLD = 0
    TPS_DIF_DIAG_COR_1 = 0
    TPS_DIF_DIAG_COR_2 = 0
    TPS_DIF_DIAG_1 = 0
    TPS_DIF_DIAG_2 = 0
    T_MAX_TPS_DIF_DIAG = 0
```

```
Endif(1)
```

61.4.3.1 Calculation of TPS_DIF_DIAG_COR_MMV_x = Filtered value

TPS_DIF_DIAG_COR_MMV_x(n) = TPS_DIF_DIAG_COR_MMV_x(n-1) +
[TPS_DIF_DIAG_COR_x - TPS_DIF_DIAG_COR_MMV_x(n-1)] *
C_TPS_SUB_CLRC_DIF

61.4.3.2 First error detection

```
If(1)
LV_DET_ERR_TPS_RATIO = 1
Then(1)
LV_CDN_DIAG_TPS_MAF_1 = 1
LV_CDN_DIAG_TPS_MAF_2 = 1
    If(2)
    T_TPS_DIF_INT < C_T_TPS_DIF_INT        //INT timer not elapsed
    Then(2)
    Integration timer  T_TPS_DIF_INT  + +        //count INT timer
    TPS_SUM_DIF_DIAG_COR_1=TPS_SUM_DIF_DIAG_COR_1_OLD + TPS_DIF_DIAG_COR_1
    TPS_SUM_DIF_DIAG_COR_2=TPS_SUM_DIF_DIAG_COR_2_OLD + TPS_DIF_DIAG_COR_2
        If(3)
        LV_TPS_SUB_DIAG_ERR = 1
        Then(3)
        ERR_SYM_TPS_MAF_1 = SYM_3
        LV_ERR_TPS_MAF_1 = 1
        LV_END_DIAG_TPS_MAF_1 = 1
        ERR_SYM_TPS_MAF_2 = SYM_3
        LV_ERR_TPS_MAF_2 = 1
        LV_END_DIAG_TPS_MAF_2 = 1
        Else(3)
            If(4) //integrated value reached thereshold
            TPS_SUM_DIF_DIAG_COR_1 > C_TPS_SUM_DIF_DIAG_COR_MAX
            Then(4)
            ERR_SYM_TPS_MAF_1 = SYM_3
            LV_ERR_TPS_MAF_1 = 1
            LV_END_DIAG_TPS_MAF_1 = 1
```

```
                If(5) // integrated value reached thereshold
                TPS_SUM_DIF_DIAG_COR_2 > C_TPS_SUM_DIF_DIAG_COR_MAX
                Then(5)
                ERR_SYM_TPS_MAF_2 = SYM_3
                LV_ERR_TPS_MAF_2 = 1
                LV_END_DIAG_TPS_MAF_2 = 1
        Else(2) //Timer ellapsed
            If(6)
            LV_TPS_SUB_DIAG_ERR = 1
            Then(6)
            ERR_SYM_TPS_MAF_1 = SYM_3
            LV_ERR_TPS_MAF_1 = 1
            LV_END_DIAG_TPS_MAF_1 = 1
            ERR_SYM_TPS_MAF_2 = SYM_3
            LV_ERR_TPS_MAF_2 = 1
            LV_END_DIAG_TPS_MAF_2 = 1
            If(7)
            TPS_SUM_DIF_DIAG_COR_1 ≥ TPS_SUM_DIF_DIAG_COR_2
            Then(7)
            ERR_SYM_TPS_MAF_1 = SYM_3
            LV_ERR_TPS_MAF_1 = 1
            LV_END_DIAG_TPS_MAF_1 = 1
            Else(7)
            ERR_SYM_TPS_MAF_2 = SYM_3
            LV_ERR_TPS_MAF_2 = 1
            LV_END_DIAG_TPS_MAF_2 = 1
            Endif
            Else(3a)
    Else(1) // set end of diagnosis
        If(8)       LV_END_DIAG_TPS_MAF_X = 0        //both
        Then(8)     LV_CDN_DIAG_TPS_MAF_X = 1        //both
                    LV_END_DIAG_TPS_MAF_X = 1        //both
        Else(8)     // Ratio symptom  no more detected, reset variables
                    LV_CDN_DIAG_TPS_MAF_X = 0        //both
                    T_TPS_DIF_INT  = 0
                    TPS_SUM_DIF_DIAG_COR_1_OLD = 0
```

```
                    TPS_SUM_DIF_DIAG_COR_2_OLD = 0
        Endif(8)
    Endif(1)
```

61.4.3.3 Check of the combustion mode with active TPS error

```
    If          LV_DET_ERR_TPS_RATIO = 1    or
                LV_ERR_TPS_1 = 1        or
                LV_ERR_TPS_2 = 1        or
                LV_ERR_TPS_MAF_1 = 1    or
                LV_ERR_TPS_MAF_2 = 1

    Then        T_DLY_TPS_DIAG_REQ_HOM_ACT ++
        If      T_DLY_TPS_DIAG_REQ_HOM_ACT>=C_T_DLY_TPS_DIAG_REQ_HOM_ACT
                and
                LV_HOM_ACT = 0
        Then    LV_CDN_DIAG_TPS_MAF_1/2 = 1         //both
                ERR_SYM_TPS_MAF_1/2 = SYM_3
                LV_ERR_TPS_MAF_1/2 = 1
                LV_END_DIAG_TPS_MAF_1/2 = 1
        Endif
    Else        T_DLY_TPS_DIAG_REQ_HOM_ACT = 0
    Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_TPS_DIF_INT	1	0...FFH	0...1.275	0.005	[s]
Integration time for deviation between measured and calculated DK value					
IP_FAC_TPS_DIF_1	8*8	0...FFH	0...1.99218	0.0078125	[-]
LDP_N_IP_FAC_TPS_DIF_1	8	0...1FE0H	0...8160	1	[rpm]
LDP_MAF_SP_TQI_IP_FAC_TPS_DIF_1	8	0...FFFFH	0...1389	0.0211948	[mg/stk]
Factor 1 for TPS_DIF_DIAG calculation					
IP_FAC_TPS_DIF_2	6	0...FFH	0...1.99218	0.0078125	[-]
LDP_PQ_SP_IP_FAC_TPS_DIF_2	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
Factor 2 for TPS_DIF_DIAG calculation					
C_TPS_SUM_DIF_DIAG_COR_MAX	1	0...FFFFH	0...32767.5	0.5	[°TPS]
Threshold of integrated corrected max. difference of measured and modelled TPS value					
C_TPS_SUB_CRLC_DIF	1	0...FFFFH	0...1	0.0153e-3	[-]
Correlation constant for filtering of error (second error)					
C_TPS_DIF_DIAG_COR_MMV_MAX	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Threshold of max. deviation of filtered TPS difference value					
LC_USE_TPS_MAF_DIAG	1	0...1H	0...1	1	[-]
Locical value to switch off TPS_MAF diagnosis (MSV70 = 0, MSD70 = 1)					
C_T_DLY_TPS_DIAG_REQ_HOM_ACT	1	0...FFH	0...1.275	0.005	[s]
Delay timer to check the HOM mode after active TPS error					
C_T_MAX_TPS_DIF_DIAG	1	0...FFH	0...1.275	0.005	[s]
Integration time for plausibility check in case of single chanel error					

61.5 Throttle position sensor diagnosis (Applic. Inc.)

61.5.1 Calculation of LV_TPS_SUB_DIAG_ERR

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TPS_SUB_DIAG_ERR	V/O	0...1H	0...1	1	[-]
TPS_SUB_DIAG is not valid due to present component error					

Input data:

LV_ERR_MAF	LV_ERR_AMP	LV_ERR_AMP_PLAUS	LV_ERR_MAP_DIP_SENS
LV_ERR_MAP_DIP_SHIFT	LV_ERR_DIAGCPS	LV_ERR_EL_CPS	LV_ERR_FSD[NC_CBK_EX_NR]
LV_VAR_MAF	LV_ERR_LS_UP[NC_CBK_EX_NR]	LV_IGK	NC_CBK_EX_NR
LV_ST_END			

Description:

TPS_SUB_DIAG is the modelled TPS value calculated out of the measured lambda and the intake manifold model. With a present component error this calculation is no more possible, thus both TPS channels are set as faulty.

Application conditions:

Initialisation: 0 at reset

Recurrence: 5ms

Activation: LV_ST_END = 1

Formula section:

```
LV_TPS_SUB_DIAG_ERR = 0
If    //OBD1 error, valid for all variants
      LV_ERR_AMP = 1                  or
      LV_ERR_AMP_PLAUS = 1            or
      LV_ERR_MAP_DIP_SENS = 1         or
      LV_ERR_MAP_DIP_SHIFT = 1
Then
      LV_TPS_SUB_DIAG_ERR = 1
ElseIf      //HFM error, valid if equiped
            LV_VAR_MAF = 1
      Then  If      LV_ERR_MAF = 1
            Then    LV_TPS_SUB_DIAG_ERR = 1
            Endif
```

```
        Else  If    LV_ERR_DIAGCPS = 1                  or
                    LV_ERR_EL_CPS = 1
              Then  LV_TPS_SUB_DIAG_ERR = 1
              Else  If    [ NC_CBK_EX_NR = 2                    and
                          ( LV_ERR_FSD[1] = 1                   and
                          LV_ERR_LS_UP[2] = 1)                  or
                          ( LV_ERR_FSD[2] = 1                   and
                          LV_ERR_LS_UP[1] = 1)                  or
                          LV_ERR_FSD[NC_CBK_EX_NR] = 1          or      //both
                          LV_ERR_LS_UP[NC_CBK_EX_NR] = 1 ]              //both
                    Then  LV_TPS_SUB_DIAG_ERR = 1
                    Else  If    NC_CBK_EX_NR = 1                        and
                                LV_ERR_FSD[1] = 1               or
                                LV_ERR_LS_UP[1] = 1
                          Then  LV_TPS_SUB_DIAG_ERR = 1
                          Endif
                    Endif
              Endif
  Endif
```

61.6 ETC driver diagnosis (LV_ERR_MTC_DR)

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>LV_ERR_MTC_DR</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">MTC_DR error present</td></tr>
<tr><td>LV_CDN_DIAG_MTC_DR</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Diagnosis condition MTC_DR</td></tr>
<tr><td>ERR_SYM_MTC_DR</td><td>V/O</td><td>0H
1H
2H
4H
8H</td><td>NO_SYM
SYM_0
SYM_1
SYM_2
SYM_3</td><td>0</td><td>[-]</td></tr>
<tr><td colspan="6">Detected symptom MTC_DR</td></tr>
<tr><td>LV_END_DIAG_MTC_DR</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">End of Diagnosis MTC_DR</td></tr>
</table>

Input data:

LV_IGK	LV_ES	LV_ST	LV_MTC_CUR_OFF
BIOS_HBR_FAULT_STATUS_MTC			

FUNCTION DESCRIPTION:

General information:

Error detection is effected via the ECU hardware (MTC H-Bridge IC).

Description:

The following errors symptoms (Error state BIOS_HBR_FAULT_STATUS_MTC is delievered from the basic software) can be distinguished:

- Undervoltage
- Overcurrent
- Overtemperature

Error-symptoms are defined to this diagnosis function as following:

```
☐☐☐☐
 │ │ │ └─> -                                    (= SYM_0)
 │ │ └───> -                                    (= SYM_1)
 │ └─────> ETC – driver is faulty               (= SYM_2)
 └───────> -                                    (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset

Recurrence: 5ms

Activation:

```
    If      LV_IGK = 1                      and
            LV_ES = 0                       and
            LV_ST = 0                       and
            LV_MTC_CUR_OFF = 0              //Position controller has power
    Then    LV_CDN_DIAG_MTC_DR = 1
    Else    LV_CDN_DIAG_MTC_DR = 0
    Endif
```

Formula section:

```
If      BIOS_HBR_FAULT_STATUS_MTC = 1
Then    ERR_SYM_MTC_DR = SYM_2
        LV_ERR_MTC_DR = 1       //after debounce
Else    ERR_SYM_MTC_DR = NO_SYM
        LV_ERR_MTC_DR = 0       //after rebounce
Endif
```

Calculation end of diagnosis:

LV_END_DIAG_MTC_DR is calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_MTC_DR	1	0...FFH	0...255	1	[-]
Debounce counter increment, MTC driver diagnosis					
C_ABC_MAX_MTC_DR	1	1...FFH	1...255	1	[-]
Debounce counter maximum value, MTC driver diagnosis					

61.7 ETC Position Controller Diagnosis (LV_ERR_MTC_CTL_n)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_MTC_CTL_1	V/O	0...1H	0...1	1	[-]
Logical variable (set. if a MTCPWM error is present)					
LV_ERR_MTC_CTL_2	V/O	0...1H	0...1	1	[-]
Logical variable (set. if a MTCPWM error is present)					
LV_ERR_MTC_CTL_3	V/O	0...1H	0...1	1	[-]
Logical variable (set. if a control difference error is present)					
LV_CDN_DIAG_MTC_CTL_1	V/O	0...1H	0...1	1	[-]
Diagnosis condition MTC_CTL_1					
LV_CDN_DIAG_MTC_CTL_2	V/O	0...1H	0...1	1	[-]
Diagnosis condition MTC_CTL_2					
LV_CDN_DIAG_MTC_CTL_3	V/O	0...1H	0...1	1	[-]
Diagnosis condition MTC_CTL_3					
ERR_SYM_MTC_CTL_1	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom MTC_CTL_1					
ERR_SYM_MTC_CTL_2	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom MTC_CTL_2					
ERR_SYM_MTC_CTL_3	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom MTC_CTL_3					
LV_END_DIAG_MTC_CTL_1	V/O	0...1H	0...1	1	[-]
End of diagnosis MTC_CTL_1					
LV_END_DIAG_MTC_CTL_2	V/O	0...1H	0...1	1	[-]
End of diagnosis MTC_CTL_2					
LV_END_DIAG_MTC_CTL_3	V/O	0...1H	0...1	1	[-]
End of diagnosis MTC_CTL_3					
T_MTCPWM_PI_DIAG	O/V	0...FFH	0...2.55	0.01	[s]
Debounce timer for MTCPWM error					

Input data:

MTCPWM	LV_TPS_AD_ACT	TPS_DIF	LV_IGK
LV_ES	LV_ST	LV_MTC_CUR_OFF	STATE_ETC_LIH
LV_TPS_SP_EXT_ADJ	LV_TPS_SP_JAM		

FUNCTION DESCRIPTION:

General information:

This diagnosis is to detect a throttle valve error or a jammed actuator.

The following diagnosis are performed:

- Pulse width modulation check
- Control difference check

Description:

Pulse width modulation check:
If the given pulse width modulation signal MTCPWM exceeds the position controller permissible maximum value C_MTCPWM_PI_MAX_DIAG for longer than C_T_MTCPWM_PI_DIAG then the error is detected and a reversible limp home is applied.

If the pulse width modulation signal lies below the threshold C_MTCPWM_PI_MAX_DIAG as long as T < C_T_MTCPWM_PI_DIAG_CUR_OFF then the jammed throttle valve could break free (LV_ERR_MTC_CTL_1 is reset to 0) and the reversible limp home is passive.

If the pulse width modulation signal lies below the threshold C_MTCPWM_PI_MAX_DIAG for longer then C_T_MTCPWM_PI_DIAG_CUR_OFF then a irreversible limp home is applied (Throttle is switched off).

Error symptoms are defined to this diagnosis as:

MTC_CTL_1

```
☐☐☐☐
 │ │ │ └─> MTC; pulse width modulation <1s exceeded    (= SYM_0)
 │ │ └───> -                                           (= SYM_1)
 │ └─────> -                                           (= SYM_2)
 └───────> -                                           (= SYM_3)
```

MTC_CTL_2

```
☐☐☐☐
 │ │ │ └─> MTC; pulse width modulation >1s exceeded    (= SYM_0)
 │ │ └───> -                                           (= SYM_1)
 │ └─────> -                                           (= SYM_2)
 └───────> -                                           (= SYM_3)
```

Checking the control difference
If the throttle valve control difference TPS_DIF exceeds the permissible maximum value C_TPS_DIF_MAX_DIAG for longer than C_T_TPS_DIF_DIAG then a error is detected and a irreversible limp home is applied (Throttle is switched off).

Error symptoms are defined to this diagnosis as:

MTC_CTL_3

```
☐☐☐☐
 │ │ │ └─> MTC; control difference check    (= SYM_0)
 │ │ └───> -                                (= SYM_1)
 │ └─────> -                                (= SYM_2)
```

```
└──────> -                                  (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset

Recurrence: 10ms

```
Activation:     If          LV_IGK = 1
                    and     LV_ST = 0
                    and     (LV_ES = 0   or  LV_TPS_SP_EXT_ADJ = 1)
                    and     LV_TPS_AD_ACT = 0
                    and     LV_MTC_CUR_OFF = 0
                    and     STATE_ETC_LIH <> 10H
                    and     LV_TPS_SP_JAM = 0
                Then        LV_CDN_DIAG_MTC_CTL_n = 1    (n = 1,2,3)
                Else        Deactivation:
                            LV_CDN_DIAG_MTC_CTL_n = 0
                Endif
```

Formula section:

Pulse width modulation check:

```
If(1)       LV_CDN_DIAG_MTC_CTL_1/2 = 1          //both
Then(1)If(2)     | MTCPWM | ≥ C_MTCPWM_PI_MAX_DIAG
    Then(2)      T_MTCPWM_PI_DIAG++
            If(3)    T_MTCPWM_PI_DIAG > C_T_MTCPWM_PI_DIAG
            Then(3)If(4)     T_MTCPWM_PI_DIAG < ...
                                     C_T_MTCPWM_PI_DIAG_CUR_OFF
                 Then(4)     ERR_SYM_MTC_CTL_1 = SYM_0
                             LV_ERR_MTC_CTL_1 = 1   //set direct
                 Else(4)     ERR_SYM_MTC_CTL_2 = SYM_0
                             LV_ERR_MTC_CTL_2 = 1   //set direct for this DC
                             ERR_SYM_MTC_CTL_1 = NO_SYM
                             LV_ERR_MTC_CTL_1 = 0   /reset
                 Endif(4)
            Else(3)     Timer is still running
            Endif(3)
    Else(2)      T_MTCPWM_PI_DIAG = 0
                 ERR_SYM_MTC_CTL_1 = NO_SYM
                 LV_ERR_MTC_CTL_1 = 0
```

```
        Endif(2)
    Else(1)     T_MTCPWM_PI_DIAG = 0
    Endif(1)
```

End of diagnosis pulse width modulation check:

```
    If          LV_CDN_DIAG_MTC_CTL_1/2 = 1          //both
    Then        Start Timer
        If          Timer  > C_T_MTCPWM_PI_DIAG
        Then        LV_END_DIAG_MTC_CTL_1 = 1
        If          Timer >= C_T_MTCPWM_PI_DIAG_CUR_OFF
        Then        LV_END_DIAG_MTC_CTL_2 = 1
        Endif
    Else        Reset timer to 0
    Endif
```

Checking the control difference

```
    If(1)       LV_CDN_DIAG_MTC_CTL_3 = 1
    Then(1)If(2)    |TPS_DIF| > C_TPS_DIF_MAX_DIAG
        Then(2)     Start Timer
            If(3)       Timer  > C_T_TPS_DIF_DIAG
            Then(3)     ERR_SYM_MTC_CTL_3 = SYM_0
                        LV_ERR_MTC_CTL_3 = 1          //set direct for this DC
            Endif(2)
        Else(2)     Reset Timer to 0
        Endif(2)
    Else(1)     Reset Timer to 0
    Endif(1)
```

End of diagnosis pulse width modulation check:

```
    If          LV_CDN_DIAG_MTC_CTL_3 = 1
    Then        Start Timer
        If          Timer  > C_T_TPS_DIF_DIAG
        Then        LV_END_DIAG_MTC_CTL_3 = 1
        Endif
    Else        Reset Timer to 0
    Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_MTCPWM_PI_MAX_DIAG	1	0...7FFFH	0...99.99694	3.05E-03	[%]
Diagnosis threshold for MTCPWM					
C_T_MTCPWM_PI_DIAG	1	0...FFH	0...2.55	0.01	[s]
Tolerance time MTCPWM-error					
C_T_MTCPWM_PI_DIAG_CUR_OFF	1	0...FFH	0...2.55	0.01	[s]
Time threshold for end “break-away”					
C_TPS_DIF_MAX_DIAG	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Maximum permissible control variance					
C_T_TPS_DIF_DIAG	1	0...FFH	0...2.55	0.01	[s]
Tolerance time control variance					

61.8 Interface for Rate – Based – Monitoring (TPS / ETC diagnosis)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_TPS	V/O	0...7H	0...7	1	[-]
Interface of TPS monitor with the Rate-Based Monitoring statistics					

Input data:

LV_END_DIAG_TPS_AD	LV_END_DIAG_TPS_AD	LV_END_DIAG_TPS_AD_BOL	LV_END_DIAG_MTC_CTL_3

FUNCTION DESCRIPTION:

General information:

With this module the interface between the TPS monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_TPS data.

Within STATE_RBM_TPS, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)
- Monitor disabled because of system malfunction (bit 1, not valid for TPS)
- Monitor indiv. RBM conditions encountered within this DC (bit 2 not valid for TPS)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_TPS = 0

on failure memory reset :

bit 1 of STATE_RBM_TPS = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

```
If    bit 0 of STATE_RBM_TPS = 0
Then        If      LV_END_DIAG_MTC_CTL_3 = 1    and
                    LV_END_DIAG_TPS_AD_BOL = 1            and
                    LV_END_DIAG_TPS_AD = 1
            Then    bit 0 of STATE_RBM_TPS = 1
            Endif
Endif
bit 2 of STATE_RBM_TPS = 1     (no individual RBM conditions)
```

61.9 Engine throttle adaptation and start-check

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TPS_AD	V/O	0...1H	0...1	1	[-]
TPS-error-bit (Adaptation)					
LV_ERR_TPS_ST_CHK_1	V/O	0...1H	0...1	1	[-]
TPS error bit (Return spring test)					
LV_ERR_TPS_ST_CHK_2	V/O	0...1H	0...1	1	[-]
TPS error bit (Return spring test)					
LV_ERR_TPS_AD_BOL	V/O	0...1H	0...1	1	[-]
Error bit (set if the continuous adaptation of MEC_BOL not passed)					
LV_CDN_DIAG_TPS_AD	V	0...1H	0...1	1	[-]
Logic variable data if diagnosis conditions TPS_AD are fulfilled					
LV_CDN_DIAG_TPS_ST_CHK_1	V	0...1H	0...1	1	[-]
Logic variable data if diagnosis conditions TPS_ST_CHK_1 are fulfilled					
LV_CDN_DIAG_TPS_ST_CHK_2	V	0...1H	0...1	1	[-]
Logic variable data if diagnosis conditions TPS_ST_CHK_2 are fulfilled					
LV_CDN_DIAG_TPS_AD_BOL	V	0...1H	0...1	1	[-]
Logic variable data if diagnosis conditions TPS_AD_BOL are fulfilled					
ERR_SYM_TPS_AD	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Detected error symptom TPS_AD					
ERR_SYM_TPS_ST_CHK_1	V	0H 1H	NO_SYM SYM_0	0	[-]
Detected error symptom TPS_ST_CHK_1					
ERR_SYM_TPS_ST_CHK_2	V	0H 1H 2H	NO_SYM SYM_0 SYM_1	0	[-]
Detected error symptom TPS_ST_CHK_2					
ERR_SYM_TPS_AD_BOL	V	0H 1H	NO_SYM SYM_0	0	[-]
Detected error symptom TPS_AD_BOL					
LV_END_DIAG_TPS_AD	V	0...1H	0...1	1	[-]
Logic variable data if end of diagnosis TPS_AD is reached					
LV_END_DIAG_TPS_ST_CHK_1	V	0...1H	0...1	1	[-]
Logic variable data if end of diagnosis TPS_ST_CHK_1 is reached					
LV_END_DIAG_TPS_ST_CHK_2	V	0...1H	0...1	1	[-]
Logic variable data if end of diagnosis TPS_ST_CHK_2 is reached					
LV_END_DIAG_TPS_AD_BOL	V	0...1H	0...1	1	[-]
Logic variable data if end of diagnosis TPS_AD_BOL is reached					
LV_N_LIM_TPS_AD	V/O	0...1H	0...1	1	[-]
Demand ETC-LIMP-HOME-2 (EMB)					
LV_TPS_AD_DIAG_CUR_OFF	V/O	0...1H	0...1	1	[-]
Demand ETC-LIMP-HOME-1					
LV_TPS_AD_REQ	V/O/S	0...1H	0...1	1	[-]
Demand TPS-adaptation					
LV_TPS_AD_ACT	V/O	0...1H	0...1	1	[-]
TPS Adaptation activ/passiv					
LV_TPS_AD_CUR_OFF	V/O	0...1H	0...1	1	[-]
Logical variable to switch of the ETC controller during TPS adaptation					
V_TPS_AD_BOL_GAIN_1	V	0...3FFH	0...4.99511	4.88E-03	[V]
Adaptation value for the lower mechanical stop of the amplification poti 1					
V_TPS_AD_EL_BOL_1	V/O/S	0...3FFH	0...4.99511	4.88E-03	[V]
Adaptation value for the lower stop, TPS-channel 1					
V_TPS_AD_EL_BOL_2	V/O/S	0...3FFH	0...4.99511	4.88E-03	[V]
Adaptation value for the lower stop, TPS-channel 2					

V_TPS_AD_LIH_1	V/O/S	0...3FFH	0...4.99511	4.88E-03	[V]
Adaptation value for the limp home position, TPS-channel 1					
V_TPS_AD_LIH_2	V/O/S	0...3FFH	0...4.99511	4.88E-03	[V]
Adaptation value for the limp home position, TPS-channel 2					
TPS_AD_SLOP_GAIN_1	V/O/S	0...FFFFH	0...7.99987	1.22E-04	[-]
Adaptation value of the amplification factor of amplified poti 1					
V_TPS_AD_OFS_GAIN_1	V/O/S	FE00...200H	-2.5...2.5	4.88E-03	[V]
Adaptation value of the amplification offset of of amplified poti 1					
V_TPS_AD_TOL_GAIN_1	V	0...3FFH	0...4.99511	4.88E-03	[V]
Amplified upper adaptation value for the adaptation of amplified of TPS-channel 1					
V_TPS_AD_TOL_1	V	0...3FFH	0...4.99511	4.88E-03	[V]
Non-amplified upper adaptation value for the adaptation of amplified of TPS-channel 1					
TPS_LIH_INI	V/O	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Limp home position of the throttle valve – initialization value					
TPS_LIH_1	V/O/S	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Limp home position of the throttle valve of TPS-Poti 1					
TPS_LIH_2	V/O/S	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Limp home position of the throttle valve of TPS-Poti 2					
TPS_SP_AD	V/O	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Throttle position setpoint during adaptation					
INH_IV_TPS_AD	O	0...FFH	0...255	1	[-]
Logical variable to inhibit injection and ignition during TPS adaptation					
TPS_AD_STEP	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H AH BH	ST_LIH_1 ST_UP_POS ST_LIH_2 ST_BOL ST_END AD_LIH_1 AD_UP_POS AD_LIH_2 AD_LOW_STOP AD_GO_LIH AD_LIH_3 AD_END	1	[-]
Steps of throttle adaptation					
CTR_ST_CHK_CHK_LIH_1_READY	V/O/S	0...FFFFH	0...65535	1	[-]
Start check check limp home 1 passed					
CTR_ST_CHK_CHK_LIH_1_ERR	V/O/S	0...FFFFH	0...65535	1	[-]
Start check check limp home 1 not passed					
LV_TPS_ST_BOL_END	V	0...1H	0...1	1	[-]
MEC_BOL check in ST_BOL routine finished					

Input data:

LV_IGK	V_TPS_1	V_TPS_2	V_TPS_GAIN_1
N	TCO	TIA	VS
VB	LV_ST	STATE_ETC_LIH	
LV_MTC_CUR_OFF	TPS_AV	TPS_LIH	
LV_ERR_TPS_1	LV_ERR_TPS_2	LV_ERR_TPS_RATIO	
LV_ERR_MTC_CTL_1	LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3	LV_ERR_MTC_DR

FUNCTION DESCRIPTION:

General information:

The throttle position is determined by a two-channeled sensor system (contact potentiometer). Both channels deliver inverse dispersing voltage signals. In order to reduce the measurement inaccuracy, the two signal voltages will be referenced to their supply voltage. After the initial engine start and component change the characteristic Potentiometer value is learnt within an adaptation routine. The value for the lower stop is stored at the end of the driving cycle in the non-volatile memory.

A plausibilization unit monitors both sensor signals, as well as the belonging supply voltage and from them establishes the system state of the THR-position acquisition (undisturbed, disturbed, THR-position not recognizable). From the voltages of both TPS-channels, the THR-position for each channel is determined, taking into consideration the adaptation values for the lower stop. Based on the determined state of the system, the resulting THR-position is determined by the individual values. In case of a disturbance of the position acquisition (LV_ERR_TPS_x ≠ 0, see chapter TPS diagnosis), the selection of the position is conducted from one of the two TPS-channels - respectively, in case of a complete collapse, the THR-control is switched off (limp home).

Application conditions:

Initialisation: all error memory outputs are initialized with 0
(LV_ERR_..., LV_CDN_DIAG_..., ERR_SYM_..., LV_END_DIAG_...)
LV_TPS_ST_BOL_END = 0 at transition LV_IGK 0 → 1
remaining outputs see chapters

Recurrence: 5ms

Activation: see Chapters

Signal flow diagram:

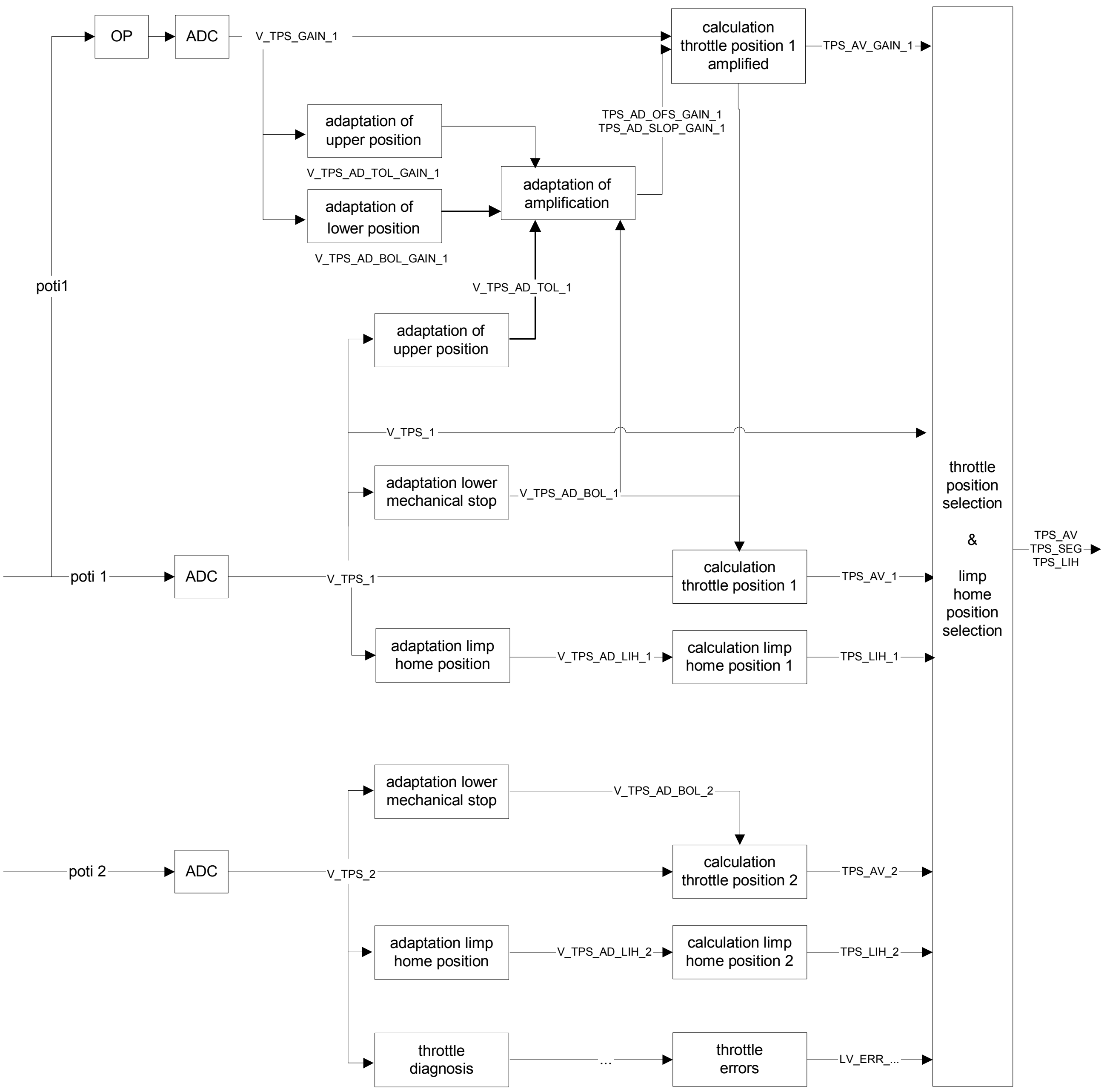
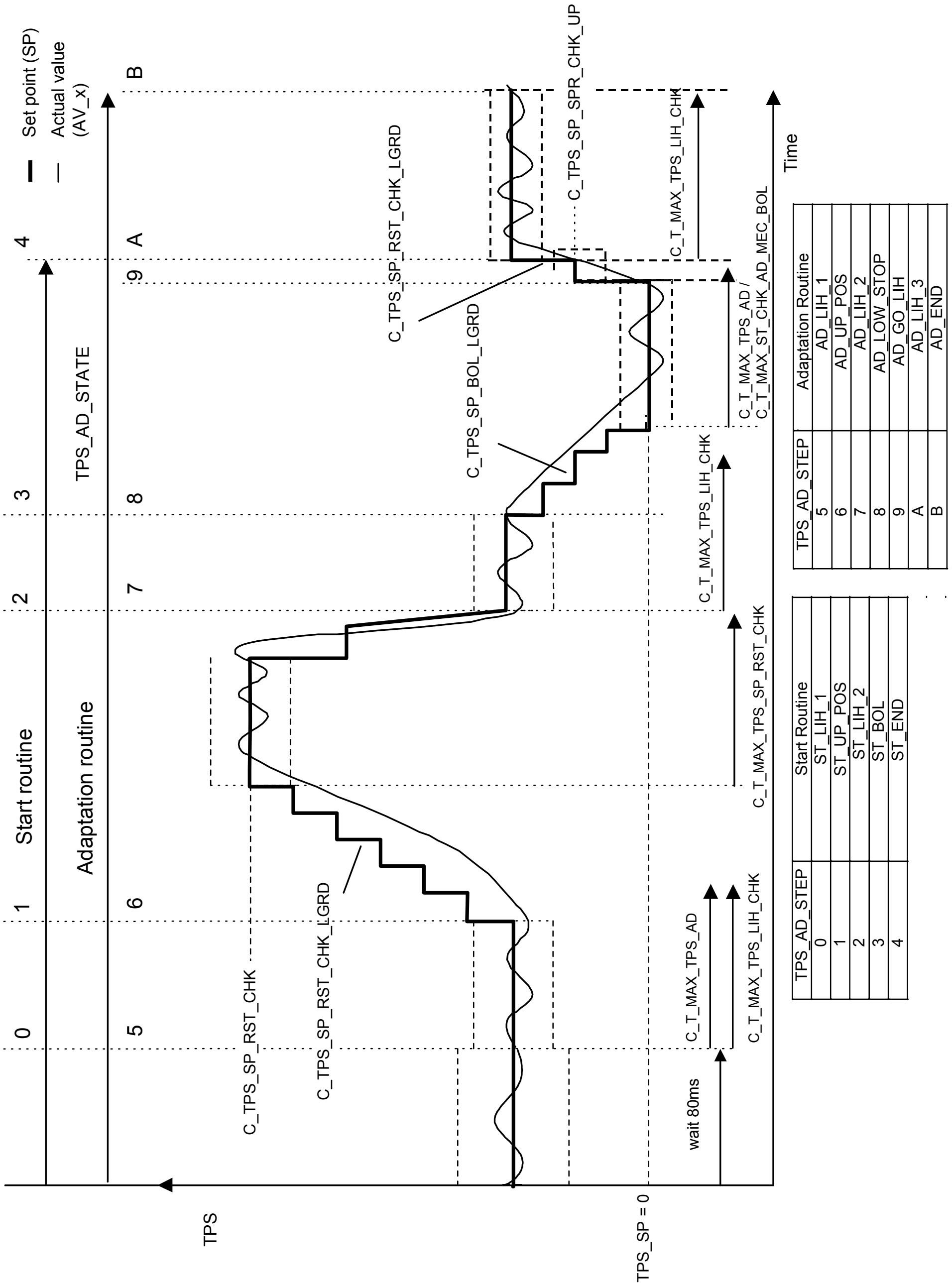


Fig. 1: Overview of the functions of the TPS-position acquisition

61.9.1 Algorithm of adaptation and start-check

Description:

The acquisition of the throttle position includes two sections: the start-check routine and adaptation routine. The start-check routine is a kind of slightly reduced adaptation routine with own thresholds and the possibility to be aborted after a first LIH-check by engine start. It is carried out at every switch on the engine control unit. In the case engine is started before start routine is finished, old adaption values for the lower mechanical stop are used. The adaptation routine starts only if the adaptation is requested.



TPS_AD_STEP	Start Routine
0	ST_LIH_1
1	ST_UP_POS
2	ST_LIH_2
3	ST_BOL
4	ST_END

TPS_AD_STEP	Adaptation Routine
5	AD_LIH_1
6	AD_UP_POS
7	AD_LIH_2
8	AD_LOW_STOP
9	AD_GO_LIH
A	AD_LIH_3
B	AD_END

Fig. 2 Time conditions of start-check / adaption routine

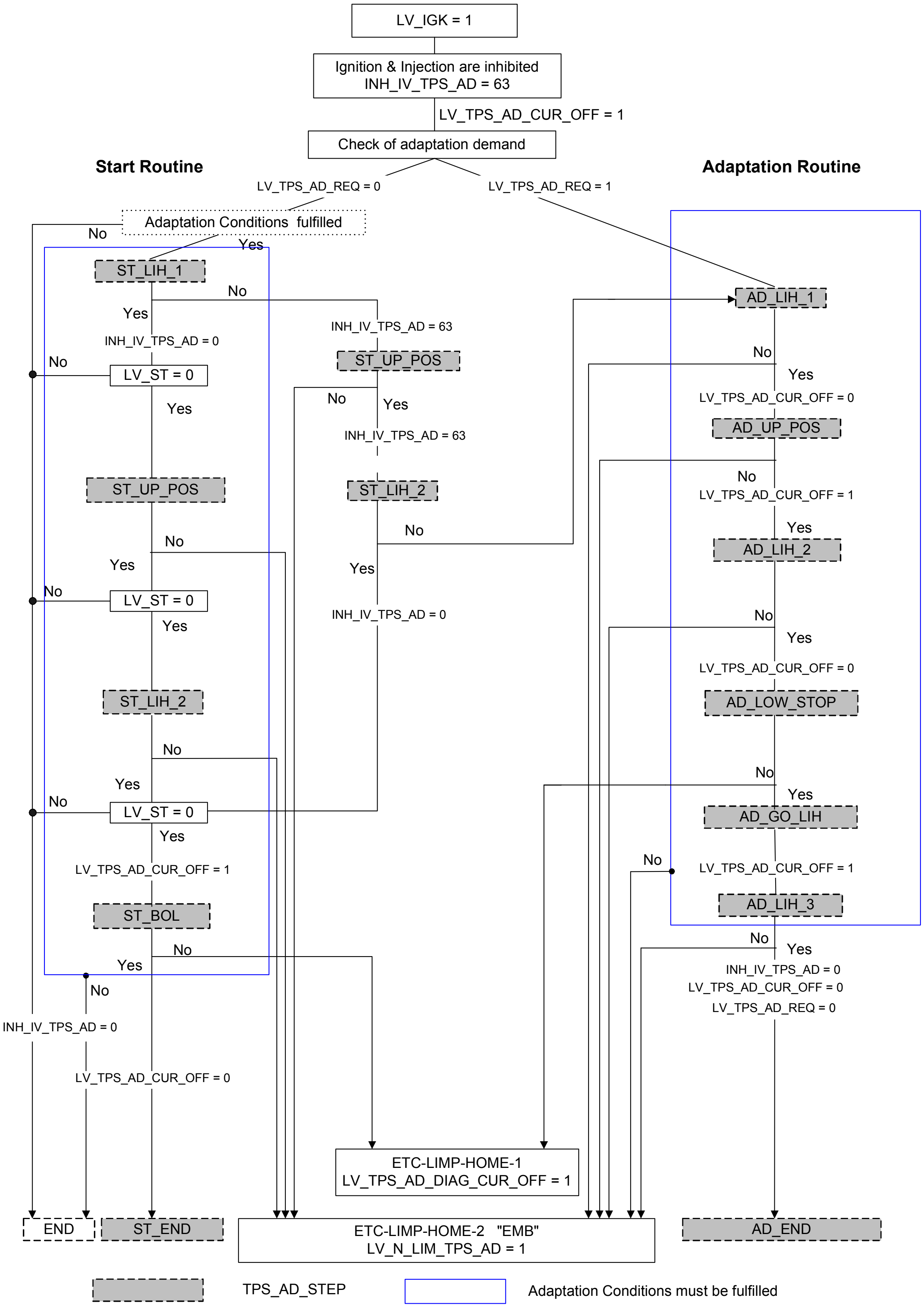


Fig. 3 Algorithm of Adaptation and Start-check

The Algorithm shows the reactions in passed and not passed conditions of both routines.

61.9.2 Initialization

General information:

The delivered voltage from the TPS-poti is a measurement for the throttle position. Because of the electrical and the mechanical tolerances, the characteristic potentiometer values must be adapted. The adaptation and diagnosis of the Poti-voltages for the lower mechanical stops and Limp Home (LIH) occurs within a set routine. By learning and checking of the Limp Home position it delivers information whether there has been an exchange in the throttle actuator. The adapted lower mechanical stop and the upper check position are used as a reference point for the calculation of the throttle valve position. The upper mechanical stop is not learned. During the adaptation phase no limitations of the THR-setpoint are active. The learned value for the lower stop and the upper position is stored at the end of the driving cycle as „non-volatile".

Formula section:

At the beginning of the adaptation (either 1st adaptation at end of line or adaptation requested by failed start check) the values are described with the following EEPROM- and Default-values:

Default (only if adaptation requested LV_TPS_AD_REQ = 1):

V_TPS_AD_LIH_1 = C_V_TPS_SP_LIH_1
V_TPS_AD_LIH_2 = C_V_TPS_SP_LIH_2
V_TPS_AD_EL_BOL_1 = C_V_TPS_AD_BOL_INI_1
V_TPS_AD_EL_BOL_2 = C_V_TPS_AD_BOL_INI_2
TPS_AD_SLOP_GAIN_1 = C_TPS_AD_SLOP_GAIN_INI_1
V_TPS_AD_OFS_GAIN_1 = C_V_TPS_AD_OFS_GAIN_INI_1
TPS_LIH_1 = TPS_LIH_2= TPS_SP_AD = TPS_LIH_INI

Always:

TPS_LIH_INI = C_TPS_SLOP / 2 *
(C_V_TPS_SP_LIH_1– C_V_TPS_AD_BOL_INI_1-
C_V_TPS_SP_LIH_2+ C_V_TPS_AD_BOL_INI_2)

TPS_SP_AD = TPS_LIH_1

61.9.3 Calculation of ERRM output variables

61.9.3.1 Diagnosis condition definition

Description:

The diagnosis condition for LV_ERR_TPS_AD is true as long as the adaptation routine is active.

The diagnosis condition for LV_ERR_TPS_ST_CHK_1 is true if TPS_AD_STEP = ST_LIH_2.

The diagnosis condition for LV_ERR_TPS_ST_CHK_2 is true if TPS_AD_STEP = ST_UP_POS **or** ST_LIH_2.

The diagnosis condition for LV_ERR_TPS_AD_BOL is true if TPS_AD_STEP = ST_BOL.

Formula section:

If TPS_AD_STEP = 5H **or** 6H **or** 7H **or** 8H **or** 9H **or** AH

Then LV_CDN_DIAG_TPS_AD = 1

```
Else    LV_CDN_DIAG_TPS_AD = 0
Endif
If      TPS_AD_STEP = 1H or 2H
Then    LV_CDN_DIAG_TPS_ST_CHK_2 = 1
Else    LV_CDN_DIAG_TPS_ST_CHK_2 = 0
Endif
If      TPS_AD_STEP = 2H
Then    LV_CDN_DIAG_TPS_ST_CHK_1 = 1
Else    LV_CDN_DIAG_TPS_ST_CHK_1 = 0
Endif
If      TPS_AD_STEP = 3H
Then    LV_CDN_DIAG_TPS_AD_BOL = 1
Else    LV_CDN_DIAG_TPS_AD_BOL = 0

Endif
```

61.9.3.2 Error symptom definition

Description:

Error symtoms are defined for this diagnosis as:

Adaptation routine

TPS_AD

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Adaptation conditions exceeded                          (= SYM_0)
 │ │ └───> Voltage value at LIH-adaptation outside range           (= SYM_1)
 │ └─────> Spring-test and CHK_LIH not passed                      (= SYM_2)
 └───────> Adaptation values – lower mechanical stop – outside range (= SYM_3)
```

Start routine

TPS_ST_CHK_1

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> New TPS adaptation necessary    (= SYM_0)
 │ │ └───> -                               (= SYM_1)
 │ └─────> -                               (= SYM_2)
 └───────> -                               (= SYM_3)
```

TPS_ST_CHK_2

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Spring-test not passed              (= SYM_0)
 │ │ └───> ST_LIH_2 not passed                 (= SYM_1)
 │ └─────> -                                   (= SYM_2)
 └───────> -                                   (= SYM_3)
```

TPS_AD_BOL

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> MEC_BOL adaptation not passed       (= SYM_0)
 │ │ └───>-                                    (= SYM_1)
 │ └─────> -                                   (= SYM_2)
 └───────> -                                   (= SYM_3)
```

61.9.3.3 Calculation end of diagnosis

Description:

The end of diagnosis in errorfree case is set on the end of Start-check/Adaptation or latest if a error is set.

Formula section:

```
Calculation end of diagnosis in error case
If      LV_ERR_TPS_XY = 1   (XY = AD or ST_CHK_1 or ST_CHK_2 or AD_BOL)
Then    LV_END_DIAG_TPS_XY = 1
Calculation end of diagnosis in errorfree case
Else    If      TPS_AD_STEP = ST_END
        Then    LV_END_DIAG_TPS_ST_CHK_1 = 1
                LV_END_DIAG_TPS_ST_CHK_2 = 1
                LV_END_DIAG_TPS_AD_BOL = 1
        Endif
        If      TPS_AD_STEP = AD_END          or
                LV_TPS_AD_REQ = 0
        Then    LV_END_DIAG_TPS_AD = 1
        Endif
Endif
```

61.9.4 Requests to activate the adaptation-routine or start-check

Description

After the control unit is switched on the activation bit for start-check or adaptation is set (LV_TPS_AD_ACT=1).

If the adaptation request LV_TPS_AD_REQ is set then the adaptation-routine will start otherwise the start-routine is activated.

Conditions for setting LV_TPS_AD_REQ:

- In initial start (first engine run or control unit exchange), that means no adaptation values are stored in the non volatile memory
- After a component exchange , if the poti values are lie outside the tolerance window
- Check sum error is current
- Initiated by the tester
- Adaptation demand requested by diagnosis e.g. if adaptation or start-check was failled

Formula section:

```
If      Adaptation request is active
Then    LV_TPS_AD_REQ = 1
Endif

If      LV_IGK 0->1
Then    LV_TPS_AD_ACT = 1           //activation bit for adaptation or start-check
        LV_TPS_AD_CUR_OFF = 1       //request to switch off ETC
        INH_IV_TPS_AD = 63          //inhibition of injection
    If      LV_TPS_AD_REQ = 0
    Then    Start routine
            TPS_AD_STEP = ST_LIH_1
    Else    Start of adaptation routine
            TPS_AD_STEP = AD_LIH_1
    Endif
Endif
```

61.9.4.1 Monitoring of adaptation or start-check conditions

Description:

To get a plausible battery voltage the check of conditions for adaptation or start-check is delayed 80 ms after control unit is switched on.

If the conditions are exceeded the routine (adaptation or start-check) is stopped, injection will be released.

If the conditions are exceeded during the adaptation routine then:

- Error “Adaptations conditions exceeded” is set
- ETC limp home 2 “EMB” is activated
- New adaptation in the next driving cycle is requested

If the conditions are exceeded during the start check routine, it will be ignored without error entry.

Formula section:

```
If(1)   LV_TPS_AD_ACT = 1        and          //Adaptation or Start-check is active
        time delay of 80ms after LV_IGK = 1 is ellapsed
Then(1)
    If(2)   LV_IGK = 0                                  or
            VB  <  C_VB_MIN_TPS_AD                      or
            N  >  C_N_MAX_TPS_AD                        or
            VS  >  2km/h (NC_VS_MAX_TPS_AD)             or
            TIA  ≤ C_TIA_MIN_TPS_AD                     or
            TCO  ≤  C_TCO_MIN_TPS_AD                    or
            TCO  ≥  C_TCO_MAX_TPS_AD                    or
            LV_ERR_TPS_1 = 1                            or
            LV_ERR_TPS_2 = 1                            or
            LV_ERR_TPS_RATIO = 1                        or
            LV_ERR_MTC_CTL_1 = 1                        or
            LV_ERR_MTC_CTL_2 = 1                        or
            LV_ERR_MTC_CTL_3 = 1                        or
            LV_ERR_MTC_DR = 1

    Then(2)     If(3)       INH_IV_TPS_AD = 63
                            and
                            TPS_AD_STEP = 1H or 2H
                Then(3)     LV_TPS_AD_ACT = 1
                            //Value remains; start routine continues; adaptation conditions for failed
                            ST_LIH_1 ignorred
                Else(3)     LV_TPS_AD_ACT = 0
                            //Stopping the TPS adaptation / start-check
                    If(4)       TPS_AD_STEP = 5H or 6H or 7H or 8H or 9H or AH
                                //Adaptation routine is active
                    Then(4)     ERR_SYM_TPS_AD = SYM_0
                                //Adaptation conditions exceeded
                                LV_ERR_TPS_AD = 1
                                LV_TPS_AD_REQ = 1
                                //New TPS adaptation necessary
                                LV_N_LIM_TPS_AD = 1
                                //ETC-LIMP-HOME-2 is requested
                    Endif(4)
                    INH_IV_TPS_AD = 0           //Activate injection
                Endif(3)
    Else(2)     continue adaptation or start-check
    Endif(2)
Endif(1)
```

61.9.5 Start check routine

General information:

The Start-check routine includes:

- First limp home check and Adaptation of the Limp Home Position
- Limp home spring test
- Second limp home check
- Adaption of lower mechanical stop

61.9.5.1 First Check of Limp Home position at Start routine (ST_LIH_1)

Description:

When checking the Limp Home position the throttle valve is staying without current hold by spring power in the Limp Home position. The actual sensor-signals are compared with their adaptations in the non-volatile memory.

If the actual voltage values during the time interval are outside the hysteresis (Fig. 4) then no agreement is recognised and a new adaptation is necessary or an engine start must be cancelled (Fig. 3).

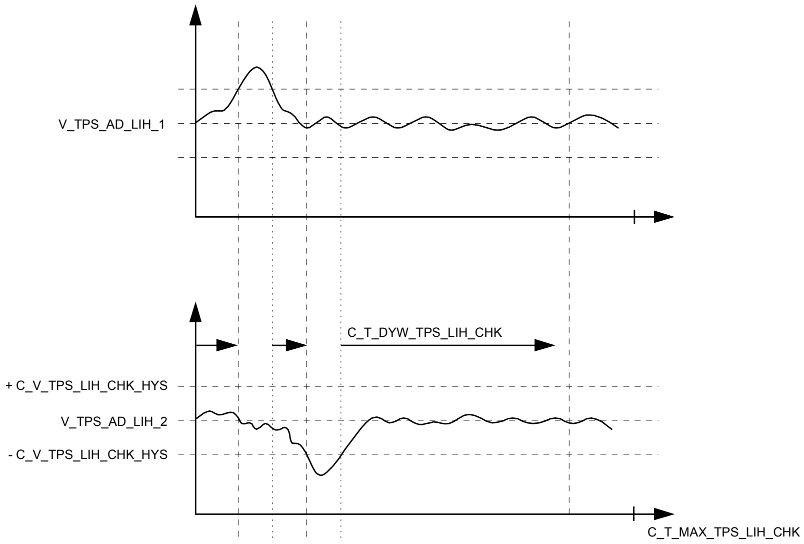


Fig. 4 Checking the Limp Home position (CHK_LIH)

Formula section:

TPS_AD_STEP = ST_LIH_1

LIH Check:

IF(1) LV_TPS_AD_ACT = 1 **and**
LV_MTC_CUR_OFF = 1 **and**
LV_TPS_AD_REQ = 0

Then(1) Start Time: C_T_MAX_TPS_LIH_CHK

```
        If(2)           T< C_T_MAX_TPS_LIH_CHK
        Then (2)        Start Time: C_T_DYW_TPS_LIH_CHK
            IF(3) $|V\_TPS\_X - V\_TPS\_AD\_LIH\_x| \le C\_V\_TPS\_LIH\_CHK\_HYS$ ( X = Poti 1, 2 )
                        and     within the interval: C_T_DYW_TPS_LIH_CHK
            Then(3)     "Check Limp home ok"
                        CTR_ST_CHK_CHK_LIH_1_READY is incremented once
                        TPS_AD_STEP = ST_UP_POS
                        LV_TPS_AD_CUR_OFF = 0
                        INH_IV_TPS_AD = 0
            Else(3)  start "Then (2)" again
        Else(2)         CTR_ST_CHK_CHK_LIH_1_ERR is incremented once
                        TPS_AD_STEP = ST_UP_POS
                        Ignition and injection remains deactivated
                        LV_TPS_AD_CUR_OFF = 0
    Else(1)     INH_IV_TPS_AD = 0
    Endif
```

Limp Home position:
Calculation of limp home positions right after CHK_LIH_1. A synchronized TPS_LIH is necessary for the Throttle catch function:

$TPS\_LIH\_1 = (V\_TPS\_AD\_LIH\_1 – V\_TPS\_AD\_EL\_BOL\_1) * C\_TPS\_SLOP$
$TPS\_LIH\_2 = (V\_TPS\_AD\_EL\_BOL\_2 – V\_TPS\_AD\_LIH\_2) * C\_TPS\_SLOP$
$0° \le TPS\_LIH\_x \le C\_TPS\_MAX$

Continuously LIH AD:

The Continuously limp home adaptation is calculated parallel to the first Limp home check. If the first Limp home Check is passed then the engine start is not locked any more and the start routine is going on with or without successfully adaptation.

```
IF(1)       LV_TPS_AD_ACT = 1                       and
            LV_MTC_CUR_OFF = 1                      and
            LV_TPS_AD_REQ = 0                       and
            "LIH CHECK 1 active":
            INH_IV_TPS_AD = 63                      and
            TPS_AD_STEP = ST_LIH_1
Then(1) Start Time: C_T_TPS_AD_HYS
    IF(2)       Within the interval:    C_T_TPS_AD_HYS              and
                $|V\_TPS\_x_{1+n} - V\_TPS\_x_1| \le C\_V\_TPS\_AD\_HYS$ (n=1,2;3;...)
    Then(2) IF(3) $\left| \frac{V\_TPS\_x_1 + V\_TPS\_x_{END}}{2} - C\_V\_TPS\_SP\_LIH\_x \right|$
```

```
                    ≤ C_V_TPS_AD_WIN_LIH
        Then(3)     $V\_TPS\_AD\_LIH\_x = \frac{V\_TPS\_x_1 + V\_TPS\_x_{END}}{2}$
        Else (3) IF(4) $\frac{V\_TPS\_x_1 + V\_TPS\_x_{END}}{2} > C\_V\_TPS\_AD\_WIN\_LIH +$
                                            $C\_V\_TPS\_SP\_LIH\_x$
            Then (4) V_TPS_AD_LIH_x = C_V_TPS_AD_WIN_LIH +
                                        C_V_TPS_SP_LIH_x
            Else (4) V_TPS_AD_LIH_x =C_V_TPS_SP_LIH_x -
                                        C_V_TPS_AD_WIN_LIH
      Else(2)  start "Then (1)" again
  Else(1)      continue with the old adaptation values
  Endif
```

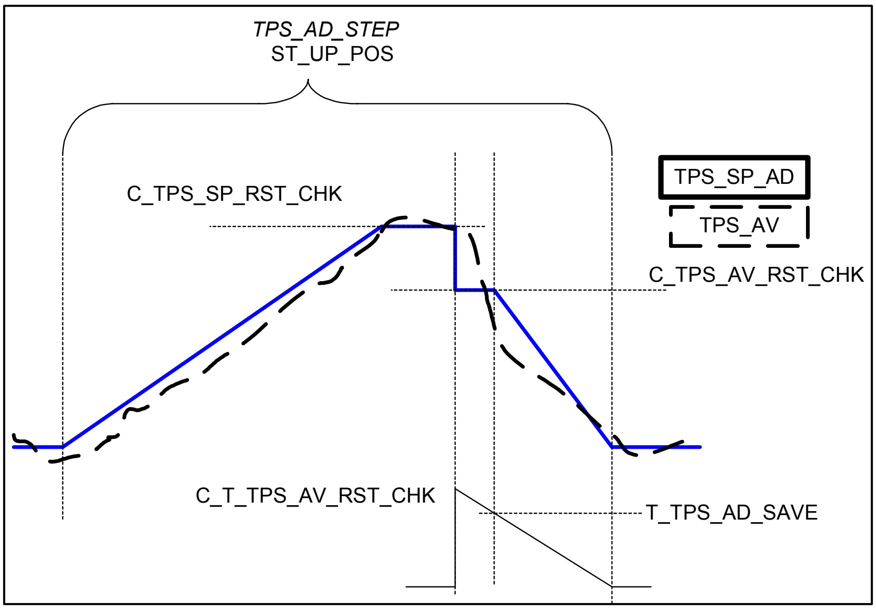
61.9.5.2 Spring Test at the Start routine (ST_UP_POS)

Description:

Out of the Limp home position the throttle position setpoint is increased using the changing limitation until the setpoint for testing the retracting spring is reached. Now a timer is started. The actual value of the throttle position has to reach the setpoint within the hysteresis within a limit maximum of time.

Regarding certain cases with closed throttle at the limp home position it is necessary to switch on the position controller near the lower mechanical stop to prevent that the throttle is going into the lower mechanical stop with high acceleration.

Signal flow diagram:



Formula section:

```
TPS_AD_STEP = ST_UP_POS
IF(1)      LV_TPS_AD_ACT = 1          and
           LV_TPS_AD_REQ = 0          and
           ( LV_ST = 0    or
           INH_IV_TPS_AD = 63 )               // ST_LIH_1 not passed
Then(1)
    If(2)  LV_TPS_AD_CUR_OFF = 0 and
           LV_MTC_CUR_OFF = 1                 // request to switch on ETC is not fulfilled
    Then(2)  // no further action
    Else(2)
           begin        TPS_SP_AD + C_TPS_SP_RST_CHK_LGRD
           until        TPS_SP_AD ≥ C_TPS_SP_RST_CHK_BAS
                  Start time: C_T_MAX_TPS_SP_RST_CHK
      If(3)       T< C_T_MAX_TPS_SP_RST_CHK                   and
                  |TPS_AV - C_TPS_SP_RST_CHK_BAS| ≤
                                              C_TPS_SP_RST_CHK_HYS
      Then(3)     LV_TPS_AD_CUR_OFF = 1
                  TPS_SP_AD = C_TPS_AV_RST_CHK
           Start Timer T_TPS_AD(dec)  =  C_T_TPS_AV_RST_CHK
           --------------------------------------
```

```
                If(4)     T_TPS_AD(dec)> 0
                Then(4)          If(5)        TPS_AV<= C_TPS_AV_RST_CHK; #1
                                 Then(5)      LV_TPS_AD_CUR_OFF = 0
                                              TPS_SP_AD = TPS_SP_AD -
                                              ((C_TPS_AV_RST_CHK - TPS_LIH) * 5ms /
                                                            T_TPS_AD_SAVE)
                                 Else(5)      go to If(4)
                                 Endif(5)
                                 #1: At transition from TPS_AV > C_TPS_AV_RST_CHK  to TPS_AV <=
                                 C_TPS_AV_RST_CHK  a snap shot is done of T_TPS_AD into
                                 T_TPS_AD_SAVE.

                Else(4)          LV_TPS_AD_CUR_OFF = 1

                Endif(4)
                ---------------------------------------
                    TPS_AD_STEP = ST_LIH_2

            Else(3)   LV_N_LIM_TPS_AD = 1                   //ETC-LIMP-HOME-2
                      ERR_SYM_TPS_ST_CHK_2 = SYM_0          // Spring-test not passed
                      LV_ERR_TPS_ST_CHK_2 = 1
        Endif(2)
    Else(1)   LV_TPS_AD_CUR_OFF = 0
    Endif(1)
```

61.9.5.3 Second Check of Limp Home position at Start routine (ST_LIH_2)

Description:

After the return spring test follows a check of limp home position again. Depending on the result of the test the start routine is finished, starts the adaptation routine or goes to security fuel cut off.

Formula section:

```
TPS_AD_STEP = ST_LIH_2
IF(1)     LV_TPS_AD_ACT = 1                            and
          LV_MTC_CUR_OFF = 1                           and
          LV_TPS_AD_REQ = 0                            and
          ( LV_ST = 0             or
          INH_IV_TPS_AD = 63 )                         // ST_LIH_1 not passed
```

```
Then(1)  Start Time: C_T_MAX_TPS_LIH_CHK
    If(2)     T< C_T_MAX_TPS_LIH_CHK
    Then (2) Start Time: C_T_DYW_TPS_LIH_CHK
        IF(3) |V_TPS_X - V_TPS_AD_LIH_x| ≤ C_V_TPS_LIH_CHK_HYS ( X = Poti 1, 2 ) and
              within the interval: C_T_DYW_TPS_LIH_CHK
        Then(3)     INH_IV_TPS_AD = 0
                    LV_TPS_AD_CUR_OFF = 0
                    TPS_AD_STEP = ST_BOL
        Else(3)     goto IF (2)
    Else (2) If(4)          INH_IV_TPS_AD = 0
            Then(4)         ERR_SYM_TPS_ST_CHK_2 = SYM_1       "CHK_LIH not passed"
                            LV_ERR_TPS_ST_CHK_2 = 1
                            LV_N_LIM_TPS_AD =1       (ETC-LIMP-HOME-2)
            Else (4)        ERR_SYM_TPS_ST_CHK_1 = SYM_0
                            "New TPS adaptation necessary"
                            LV_ERR_TPS_ST_CHK_1 = 1
                            LV_TPS_AD_REQ = 1
                            TPS_AD_STEP = AD_LIH_1
Else(1)     LV_TPS_AD_CUR_OFF = 0
Endif(1)
```

61.9.5.4 Adaption of lower mechanical stop and limp home spring test opening at Start routine (ST_BOL)

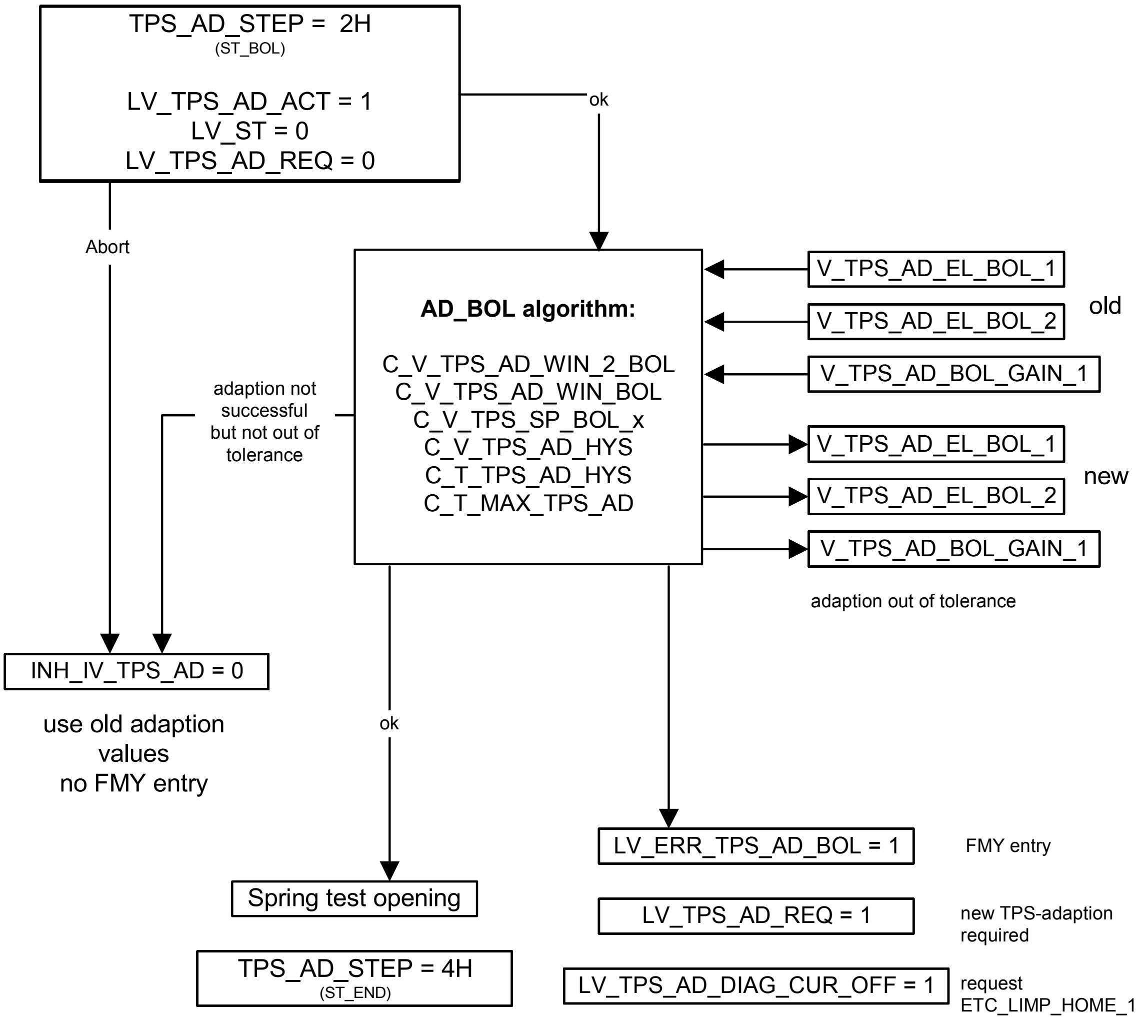
Description:

During a complete start check (not aborted by a quick start), the lower mechanical stop is learnt. After successful adaptation, the adaptation values V_TPS_AD_EL_BOL_1/2 are saved in the non-volatile memory. If the adaptation is not completed successfully (e.g. abnormal termination due to violation of the activation conditions), the adaptation value learnt last remains valid.

During adaptation, all voltage values must be within the adaptation window V_TPS_AD_EL_BOL_X ± C_V_TPS_AD_WIN_2_BOL; otherwise LV_ERR_TPS_AD_BOL is set.

A hysteresis band C_V_TPS_AD_HYS is placed around the first detected value (pots 1 and 2). If all values are within this hysteresis band during the adjustable time C_T_TPS_AD_HYS, the adaptation values are determined from the first and last value each (Fig. 6). If the value of a pot is outside of the hysteresis band, the operation is re-started for the two pots. The learning algorithm is limited by the maximum time C_T_MAX_ST_CHK_AD_MEC_BOL.

Signal flow diagram:



Formula section:

```
TPS_AD_STEP = ST_BOL
IF(1)      LV_TPS_AD_ACT = 1          and
           LV_ST = 0                  and
           LV_TPS_AD_REQ = 0
Then(1)
    If(2)      LV_TPS_ST_BOL_END = 0
    Then(2)
        If(3)      TPS_SP_AD > 0
        Then(3)    $TPS\_SP\_AD = TPS\_SP\_AD_{(n-1)} – C\_TPS\_SP\_BOL\_LGRD$
        Else(3)
```

```
Start time: C_T_MAX_ST_BOL

If(4)        |V_TPS _x - C_V_TPS_SP_BOL _x| > C_V_TPS_AD_WIN_BOL
             or
             |V_TPS _x - V_TPS_AD_EL_BOL _x| > C_V_TPS_AD_WIN_2_BOL
Then(4)
    If(5)        T ≥ C_T_MAX_ST_BOL
    Then(5)      ERR_SYM_TPS_AD_BOL = SYM_0
                 //MEC_BOL adaptation not passed
                 LV_ERR_TPS_AD_BOL = 1
                 LV_TPS_AD_REQ = 1
                 //New TPS adaptation necessary
                 LV_TPS_AD_DIAG_CUR_OFF = 1
                 //ETC-LIMP-HOME-1 demand
                 LV_TPS_AD_ACT = 0
    Else(5)      reset timer C_T_TPS_ST_BOL_HYS
    Endif(5)

Else(4)      continue  MEC BOL Adaptation
Endif(4)
    If(6)        T < C_T_MAX_ST_BOL
    Then(6)
        If(7)        Within the interval: C_T_TPS_ST_BOL_HYS
                         and
                     |V_TPS_x1+n - V_TPS_ x1 | ≤ C_V_TPS_AD_HYS        (n=1,2;3;…)
        Then(7)
                     V_TPS_AD_EL_BOL_x =
                     $\frac{V\_TPS\_x_1 + V\_TPS\_x_{END}}{2}$

                     V_TPS_AD_BOL_GAIN_1 =
                     $\frac{V\_TP1\_GAIN\_1_1 + V\_TPS\_GAIN\_1_{END}}{2}$
                     LV_TPS_ST_BOL_END = 1
```

```
                        Else(7)        go to If(6)
                    Else (6)           adaptation remains the last valid values
                                       LV_TPS_ST_BOL_END = 1
                    Endif (6)

            Endif(3)

    Else(2)
                        If8)           TPS_SP_AD < C_TPS_SP_SPR_CHK_UP

                Then(8)     TPS_SP_AD =     TPS_SP_AD(n-1) +
                                            C_TPS_SP_RST_CHK_LGRD
                Else(8)
                            LV_TPS_AD_CUR_OFF = 1
                            TPS_SP_AD = TPS_LIH_INI
                            TPS_AD_STEP = ST_END
                            // Second Limp Home Check in case of not successfull adaptation of lower stop

Else (1)            INH_IV_TPS_AD = 0
                    LV_TPS_AD_CUR_OFF = 0
Endif (1)
```

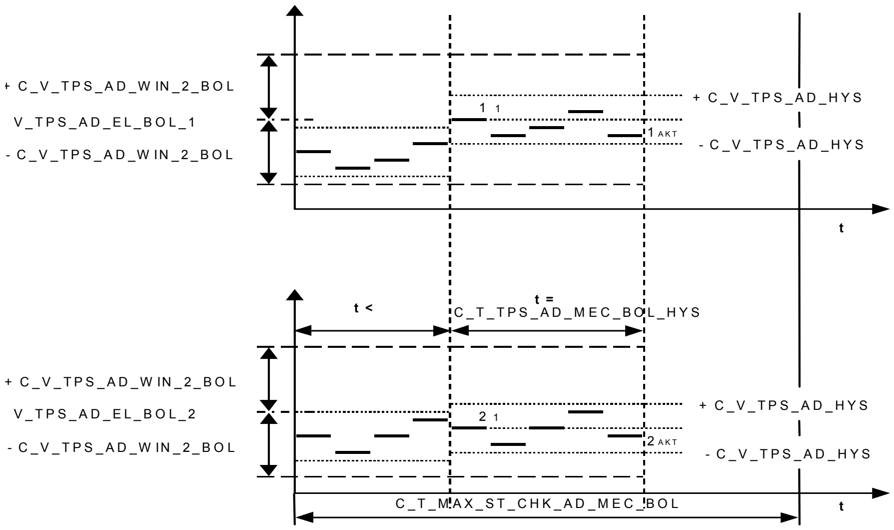


Fig. 5 Determination of the adaptation values in case of MEC_BOL adaptation during start routine

61.9.6 Adaptation routine

General information:

The Adaption routine includes:

- Limp home check and Adaptation of Limp Home Position
- Limp home spring test closing and adaptation of upper position
- Limp home check
- Limp home spring test opening and Adaptation of lower mechanical stop
- Limp home check

61.9.6.1 Adaption of Limp Home position at Adaption routine (AD_LIH_1)

Description:

The first step is the adaptation of the Limp Home position. The actuator is without current and is forced by the spring power in the limp home position.

The learning of the adaptation value depict the algorithm in Fig. 6. During the adaptation all the voltage values must be in the adaptation windows. A hysteresis band is set up around each of the first recorded values (poti 1 and 2). If all the values lie in this hysteresis band during an adjustable time then the adaptation values are determined from each first and last value. If the Poti-value lies outside the hysteresis band then the process is newly started for both Poti's. The learning algorithm is limited by the maximum time. This procedure is the same for the Limp home position and the lower mechanical stop. Just the calibration values are different.

An error memory entry occurs only at the adaptation after ECU-first initialisation not at the continuous LIH-adaptation. An error is detected if the adaptation could not be carried out during the max. time C_T_MAX_TPS_AD.

Formula section:

```
TPS_AD_STEP = AD_LIH_1
Start time: C_T_MAX_TPS_AD
    If (1)   |V_TPS _x - C_V_TPS_SP_LIH_x)| ≤ C_V_TPS_AD_WIN_LIH)
    Then(1) If(2)T < C_T_MAX_TPS_AD
            Then(2) Start Time: C_T_TPS_AD_HYS
                IF(3)  Within the interval:  C_T_TPS_AD_HYS  and
                |V_TPS _$x_{1+n}$ - V_TPS _$x_1$| ≤ C_V_TPS_AD_HYS) (n=1,2;3;…)
                Then(3)
                    $V\_TPS\_AD\_LIH\_x = \frac{V\_TPS\_x_1 + V\_TPS\_x_{END}}{2}$
                    LV_TPS_AD_CUR_OFF = 0
                    TPS_AD_STEP = AD_UP_POS
                Else(3)  goto If (2)
```

```
                Else(2)  goto Else (1)
        Else(1)     ERR_SYM_TPS_AD = SYM_1 "Voltage value at LIH-adaptation outside range"
                    LV_ERR_TPS_AD = 1
                    LV_TPS_AD_ACT = 0          (Stopping the TPS adaptation)
                    LV_N_LIM_TPS_AD = 1        (ETC-LIMP-HOME-2)
                    LV_TPS_AD_REQ = 1          (New TPS adaptation nesecarry)
                    INH_IV_TPS_AD = 0          (Activate ignition and injection)
        Endif
```

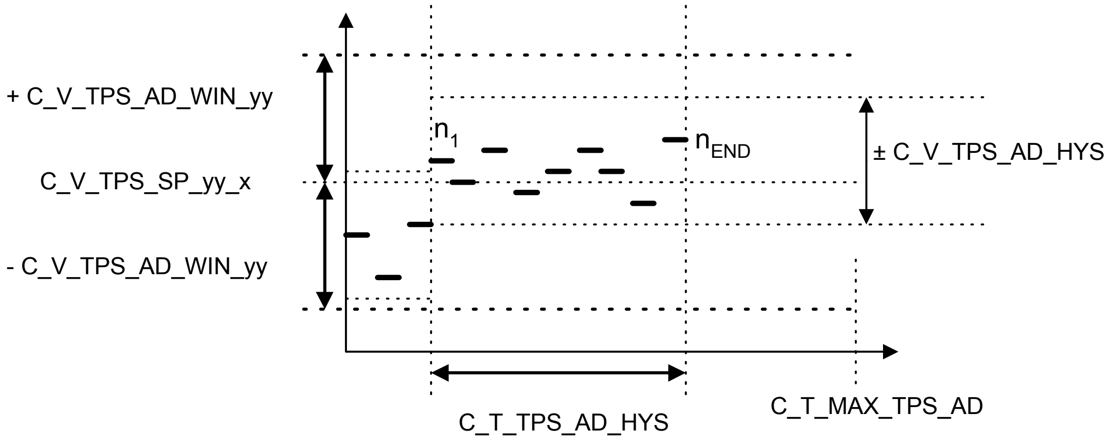


Fig. 6 Determining the adaptation value (AD_yy) yy correspond to LIH or BOL
x correspond to poti 1 or 2

61.9.6.2 Spring test at the adaptation routine (AD_UP_POS)

Description:

After passing the check of limp home position the adaptation routine goes on with the spring test. Out of the Limp home position the throttle position set point is increased using the changing limitation until the set point for testing the retracting spring is reached. Now a timer is started. The actual value of the throttle position has to reach the set point within the hysteresis and a limit of time. Then the upper position will be adapted and the throttle is returning currentless.

Formula section:

```
TPS_AD_STEP = AD_UP_POS
begin       TPS_SP_AD + C_TPS_SP_RST_CHK_LGRD
until       TPS_SP_AD ≥ C_TPS_SP_RST_CHK
Start Time : C_T_MAX_TPS_SP_RST_CHK
If(1) |TPS_AV - C_TPS_SP_RST_CHK |≤ C_TPS_SP_RST_CHK_HYS)     and
      T <  C_T_MAX_TPS_SP_RST_CHK
Then(1)     Start time C_T_MAX_TPS_AD
```

```
        IF(2)   T <  C_T_MAX_TPS_AD
      Then(2)
          If(3)  Within the interval:     C_T_TPS_AD_HYS                        and
                 $| V\_TPS\_1_{1+n} - V\_TPS\_1_{1} | \leq C\_V\_TPS\_AD\_HYS$        (n=1,2;3;…)
          Then(3)
                 New calculation of upper value of amplified poti 1
                 $V\_TPS\_AD\_TOL\_1 = \frac{V\_TPS\_1_1 + V\_TPS\_1_{END}}{2}$
                 $V\_TPS\_AD\_TOL\_GAIN\_1 = \frac{V\_TPS\_GAIN\_1_1 + V\_TPS\_GAIN\_1_{END}}{2}$
                 LV_TPS_AD_CUR_OFF = 1
                 TPS_SP_AD = C_TPS_AV_RST_CHK
                 ------------------------------------------
                 Start Timer $T\_TPS\_AD_{(dec)}$  =  C_T_TPS_AV_RST_CHK
                 If       $T\_TPS\_AD_{(dec)} > 0$
                               If       TPS_AV<= C_TPS_AV_RST_CHK; #1
                               Then  LV_TPS_AD_CUR_OFF = 0
                                        TPS_SP_AD = TPS_SP_AD -
                                        ((C_TPS_AV_RST_CHK - TPS_LIH) * 5ms /
                                                               T_TPS_AD_SAVE)
                               endif
                 Else          LV_TPS_AD_CUR_OFF = 1
                 Endif
                 #1: At transition from TPS_AV > C_TPS_AV_RST_CHK to TPS_AV <=
                 C_TPS_AV_RST_CHK  a snap shot is done of T_TPS_AD into T_TPS_AD_SAVE.

                 --------------------------------------
                 TPS_AD_STEP = AD_LIH_2
        Else(3)                goto If (2)
      Else(2)           goto Else (1)

    Else(1)     ERR_SYM_TPS_AD = SYM_2          "Spring test and CHK_LIH not passed"
                LV_ERR_TPS_AD = 1
                LV_TPS_AD_ACT = 0               (Stopping the TPS adaptation)
                LV_N_LIM_TPS_AD = 1             (ETC-LIMP-HOME-2, )
                LV_TPS_AD_REQ = 1               (New TPS adaptation nesecarry)
                INH_IV_TPS_AD = 0               (Activate ignition and injection)
    Endi
```

61.9.6.3 Second Check of Limp Home position at Adaptation routine (AD_LIH_2)

Description:

After the return spring test the adaptation routine repeats the ckeck of limp home position.

Formula section:

```
TPS_AD_STEP = AD_LIH_2
Start time: C_T_MAX_TPS_LIH_CHK
If(1)      T < C_T_MAX_TPS_LIH_CHK
Then(1)
   If(2)   within the interval: C_T_DYW_TPS_LIH_CHK                and
           |V_TPS_x - V_TPS_AD_LIH_x| ≤  C_V_TPS_LIH_CHK_HYS  ( X = Poti 1, 2 )
   Then(2) LV_TPS_AD_CUT_OFF = 0
           TPS_AD_STEP = AD_LOW_STOP
   Else(2) goto If (1)
Else(1)    ERR_SYM_TPS_AD = SYM_2       “Spring test and CHK_LIH not passed”
           LV_ERR_TPS_AD = 1
           LV_TPS_AD_ACT = 0            (Stopping the TPS adaptation)
           LV_N_LIM_TPS_AD = 1          (ETC-LIMP-HOME-2, )
           LV_TPS_AD_REQ = 1            (New TPS adaptation nesecarry)
           INH_IV_TPS_AD = 0            (Activate ignition and injection)
Endif
```

61.9.6.4 Adaptation of lower mechanical stop during adaption routine (AD_LOW_STOP)

Description:

As soon as the mechanical stop is reached, the described adaptation algorithm in Fig. 7, is triggered. The scanned data is used for both, calculation of the lower mechanical / electrical stop and for the amplification characteristic. After the adaptation of the lower stop and calculated amplification the throttle at the electrical stop opens a small angle in relation to the mechanical stop. This is necessary to prevent a chock.

Formula section:

```
TPS_AD_STEP = AD_LOW_STOP
Begin          TPS_SP_AD - C_TPS_SP_BOL_LGRD
Until          TPS_SP_AD = 0
Start Time: C_T_MAX_TPS_AD
If (1)     |V_TPS _x  - C_V_TPS_SP_BOL _x| ≤ C_V_TPS_AD_WIN_BOL)
```

```
Then(1)
  If(2)     $T < C\_T\_MAX\_TPS\_AD$
  Then(2)
    IF(3)      Within the interval: C_T_TPS_AD_HYS                         and
               $|V\_TPS\_x_{1+n} - V\_TPS\_x_1| \leq C\_V\_TPS\_AD\_HYS$
               (n=1,2;3;…)
    Then(3)
       $V\_TPS\_AD\_EL\_BOL\_x = \frac{V\_TPS\_x_1 + V\_TPS\_x_{END}}{2}$
       $V\_TPS\_AD\_BOL\_GAIN\_1 = \frac{V\_TP1\_GAIN\_1_1 + V\_TPS\_GAIN\_1_{END}}{2}$
     slope of amplifaction of amplified poti 1:
           $TPS\_AD\_SLOP\_GAIN\_1 = \frac{V\_TPS\_AD\_TOL\_GAIN\_1 - V\_TPS\_AD\_BOL\_GAIN\_1}{V\_TPS\_AD\_TOL\_1 - V\_TPS\_AD\_EL\_BOL\_1}$
     offset of amplifaction of amplified poti 1:
     $V\_TPS\_AD\_OFS\_GAIN\_1 = V\_TPS\_AD\_BOL\_GAIN\_1 - V\_TPS AD\_EL\_BOL\_1 * TPS\_AD\_SLOP\_GAIN \_1$
         TPS_AD_STEP = AD_GO_LIH
    Else(3) go to If(2)
  Else(2)  go to else(1)
Else (1)        ERR_SYM_TPS_AD = SYM_3
                "Adaptation value - lower mechanical stop – outside range"

                LV_ERR_TPS_AD = 1

                LV_TPS_AD_ACT = 0                   (Stopping the TPS adaptation)
                LV_TPS_AD_DIAG_CUR_OFF = 1          (ETC-LIMP-HOME-1 demand)
                LV_TPS_AD_REQ = 1                   (New TPS adaptation necessary)
                INH_IV_TPS_AD = 0                   (Activate ignition and injection)
Endif
```

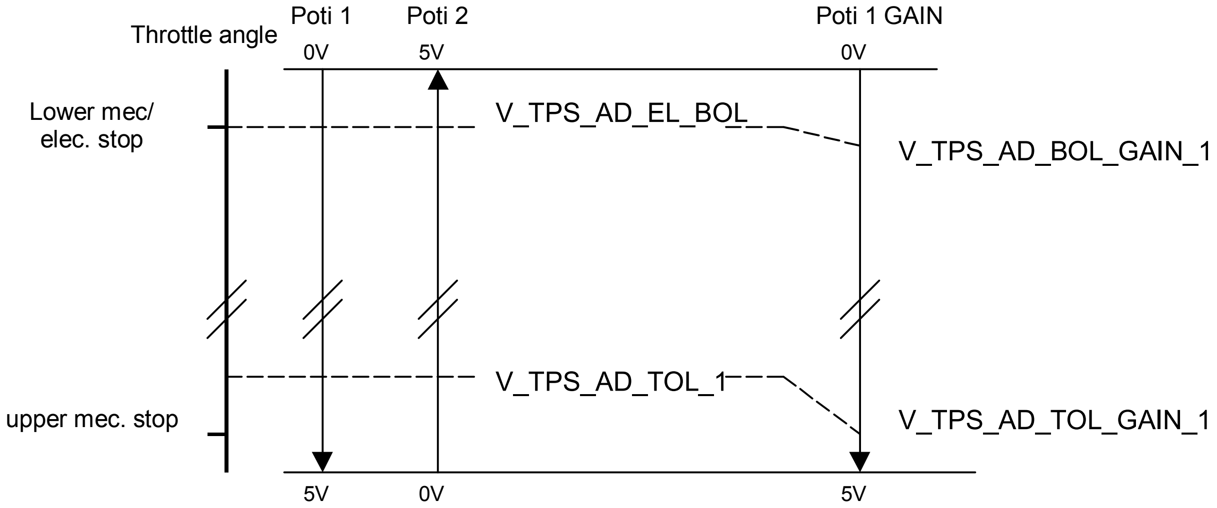


Fig 7. Definition of the adaptation positions

61.9.6.5 Limp home spring Test opening (AD_GO_LIH)

Description:

After finish the lower adaptation the MTC opens the throttle using the change limitation until the nominal values for current cut off is reached. As from that position the spring returns the throttle currentless to the Limp Home position and carries through the Limp Home Check again, because it is possible that the throttle itself clamps at the mechanical stop.

Formula section:

```
TPS_AD_STEP = AD_GO_LIH
begin
            TPS_SP_AD + C_TPS_SP_RST_CHK_LGRD
until       $TPS\_SP\_AD \geq C\_TPS\_SP\_SPR\_CHK\_UP$
LV_TPS_AD_CUR_OFF = 1
TPS_SP_AD = TPS_LIH_INI
TPS_AD_STEP = AD_LIH_3
Third Limp Home Check
```

61.9.6.6 Third Check of Limp Home position at Adaptation routine (AD_LIH_3)

Description:

After the spring test at the lower stop follows the check of limp home position again. Now the adaptation routine is finished. Depending on the passed test the adaptation request is cancelled.

Formula section:

```
TPS_AD_STEP = AD_LIH_3
Start time: C_T_MAX_TPS_LIH_CHK
If(1)         T <  C_T_MAX_TPS_LIH_CHK
Then(1)       Start Time: C_T_DYW_TPS_LIH_CHK
    If(2)     within the interval: C_T_DYW_TPS_LIH_CHK                and
              |V_TPS_X -V_TPS_AD_LIH_x|  ≤ C_V_TPS_LIH_CHK_HYS       ( X = Poti 1, 2 )
    Then(2)       LV_TPS_AD_CUR_OFF = 0
                  INH_IV_TPS_AD =0
                  TPS_AD_STEP = AD_END
    Else (2)      go to If (1)
Else (1)          ERR_SYM_TPS_AD = SYM_2        "Spring test and CHK_LIH not passed"
                  LV_ERR_TPS_AD = 1
                  LV_TPS_AD_ACT = 0          (Stopping the TPS adaptation)
                  LV_TPS_AD_REQ = 1          (New TPS adaptation necessary)
                  LV_N_LIM_TPS_AD = 1        (ETC-LIMP-HOME-2)
                  INH_IV_TPS_AD = 0          (Activate ignition and injection)
Endif
```

Limp Home position:

TPS_LIH_1 = (V_TPS_AD_LIH_1 – V_TPS_AD_EL_BOL_1) * C_TPS_SLOP

TPS_LIH_2 = (V_TPS_AD_EL_BOL_2 – V_TPS_AD_LIH_2) * C_TPS_SLOP

0° ≤ TPS_LIH_x ≤ C_TPS_MAX

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_V_TPS_AD_BOL_INI_1	1	0...FFH	0...4.998	1.95E-02	[V]
Start value for the adaptation value lower start (Poti 1)					
C_V_TPS_AD_BOL_INI_2	1	0...FFH	0...4.998	1.95E-02	[V]
Start value for the adaptation value lower start (Poti 2)					
C_V_TPS_AD_WIN_LIH	1	0...FFH	0...2.49023	9.77E-03	[V]
Adaptation window size (+/-) for adaptation of the Limp Home position					
C_V_TPS_AD_HYS	1	0...FFH	0...1.24511	4.88E-03	[V]
Hysteresis band for adaptation value determination					
C_V_TPS_AD_WIN_BOL	1	0...FFH	0...2.49023	9.77E-03	[V]
Adaptation window size (+/-) for adaptation of lower stop					
C_V_TPS_AD_WIN_2_BOL	1	0...FFH	0...2.49023	9.77E-03	[V]
Adaptation window size (+/-) for continousliy mec BOL adaptation of lower stop					
C_V_TPS_SP_BOL_1	1	0...FFH	0...4.998	1.95E-02	[V]
lower limit for plausibility check of the adaptation value lower stop TPS_1					
C_V_TPS_SP_BOL_2	1	0...FFH	0...4.998	1.95E-02	[V]
upper limit for plausibility check of the adaptation value lower stop TPS_1					
C_V_TPS_SP_LIH_1	1	0...FFH	0...4.998	1.95E-02	[V]
Set point of the limp home position voltage value channel 1					
C_V_TPS_SP_LIH_2	1	0...FFH	0...4.998	1.95E-02	[V]
Set point of the limp home position voltage value channel 2					
C_V_TPS_LIH_CHK_HYS	1	0...FFH	0...1.24511	4.88E-03	[V]
Admissible tolerance on checking the limp home position					
C_T_DYW_TPS_LIH_CHK	1	0...FFH	0...1.275	0.005	[s]
Time period for Dynamic window around Limp Home					
C_T_MAX_TPS_LIH_CHK	1	0...FFH	0...1.275	0.005	[s]
Maximum time for Limp Home recognition					
C_T_TPS_AD_HYS	1	0...FFH	0...1.275	0.005	[s]
Time period for hysteresis band to adaptation					
C_T_TPS_ST_BOL_HYS	1	0...FFH	0...1.275	0.005	[s]
Time period for hysteresis band to adaptation of BOL position during Start Check					
C_T_MAX_TPS_AD	1	0...FFH	0...1.275	0.005	[s]
Maximum time for adaptation of the LIH and BOL position					
C_T_MAX_ST_BOL	1	0...FFH	0...1.275	0.005	[s]
Maximum time for adaptation of BOL position during Start Check					
C_T_MAX_TPS_SP_RST_CHK	1	0...FFH	0...1.275	0.005	[s]
Maximum time to reach the position C_TPS_SP_RST_CHK					
C_TPS_SP_RST_CHK_LGRD	1	1...FFH	7.29414E-03 ...1.86000	7.2941e-3	[°TPS/5ms]
Change limitation for stopping the position C_TPS_SP_RST_CHK					
C_TPS_SP_RST_CHK	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
TPS_SP for testing the retracting spring					
C_TPS_SP_RST_CHK_BAS	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
TPS_SP for testing the retracting spring					
C_TPS_SP_SPR_CHK_UP	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
TPS_SP for GO_LIH position					
C_TPS_SP_RST_CHK_HYS	1	1...FFH	7.29414E-03 ...1.86000	7.2941e-3	[°TPS]
Hysteresis of C_TPS_SP_RST_CHK					
C_TPS_SP_BOL_LGRD	1	1...FFH	7.29414E-03 ...1.86000	7.2941e-3	[°TPS/5ms]
Change limitation for stopping the lower position					
C_TPS_AD_SLOP_GAIN_INI_1	1	0...FFFFH	0...7.99987	1.22E-04	[-]
Nominal value of the amplification slope of amplified poti 1					
C_V_TPS_AD_OFS_GAIN_INI_1	1	FE00...200H	-2.5...2.5	4.88E-03	[V]
Nominal value of the amplification offset of amplified poti 1					
C_VB_MIN_TPS_AD	1	0...FFH	0...25.89843	0.1015625	[V]
VB threshold for ETC adaptation					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_TPS_AD	1	0...FFH	0...8160	32	[rpm]
Maximum engine speed for adaptation					
C_TIA_MIN_TPS_AD	1	0...FEH	-48...142.5	0.75	[°C]
TIA threshold for TPS adaptation					
C_TCO_MIN_TPS_AD	1	0...FEH	-48...142.5	0.75	[°C]
Min. TCO threshold for TPS adaptation					
C_TCO_MAX_TPS_AD	1	0...FEH	-48...142.5	0.75	[°C]
Max. TCO threshold for TPS adaptation					
C_TPS_AV_RST_CHK	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Threshold for activating the position controller					
C_T_TPS_AV_RST_CHK	1	0...FFH	0...1.275	0.005	[s]
Time condition for deactivating the position controller					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_VS_MAX_TPS_AD	1	0...FFH	0...255	1	[km/h]
Maximum vehicle speed treshold for adaptation					

61.10 TPS / ETC Limp-Home Manager

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_TPS	V/O	0...1H	0...1	1	[-]
Logical variable, set if one TPS channel is identified as faulty or a ETC-Limp-Home is active					
ERR_SYM_TPS	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom					
LV_CDN_DIAG_TPS	V	0...1H	0...1	1	[-]
Diagnosis condition					
LV_END_DIAG_TPS	V	0...1H	0...1	1	[-]
End of Diagnosis					
LV_MTC_LIH_CUR_OFF	V/O	0...1H	0...1	1	[-]
ETC-LIMP-HOME-1					
LV_N_LIM_ETC_LIH	V/O	0...1H	0...1	1	[-]
ETC-LIMP-HOME-2 (SAS)					
LV_N_LIM_ETC_LIH_REV	V/O	0...1H	0...1	1	[-]
ETC-LIMP-HOME-3 (Reversible)					
LV_TPS_MTC_N_LIM	V/O	0...1H	0...1	1	[-]
Logical variable, set if LIH engine speed limitation is required					
LV_MTC_CUR_OFF_REQ	V/O	0...1H	0...1	1	[-]
ETC without power request					
STATE_TPS_DIAG	O/V	0H 1H 2H 4H 8H	TPS_NO_ERROR TPS_LIH_1 TPS_LIH_2 TPS_LIH_MAX TPS_DBL_ERROR	1	-
Status TPS Diagnosis					
STATE_ETC_LIH	O/V	0H 1H 2H 4H 8H	ETC_NO_LIH ETC_LIH_1 ETC_LIH_2_REV ETC_LIH_2 ETC_LIH_3	1	-
Status ETC Limp-Home					

Input data:

LV_ERR_TPS_1	LV_ERR_TPS_2	LV_DET_ERR_TPS_RATIO	LV_ERR_TPS_MAF_1
LV_ERR_TPS_MAF_2	LV_ERR_MTC_CTL_1	LV_ERR_MTC_CTL_2	LV_ERR_MTC_CTL_3
LV_N_LIM_TPS_AD	LV_TPS_AD_DIAG_CUR_OFF	V_TPS_1	V_TPS_2
LV_ERR_MTC_DR	C_T_DYW_TPS_LIH_CHK	LV_IGK	LV_CT
C_V_TPS_LIH_CHK_HYS	C_T_MAX_TPS_LIH_CHK		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_TPS	1	0...FFH	0...255	1	[-]
Anti-bounce increment					
C_ABC_MAX_TPS	1	1...FFH	1...255	1	[-]
Anti-bounce maximum					

FUNCTION DESCRIPTION:

General information:

The following TPS/ETC-Limp Home system reactions are defined:

- **TPS-EMERGENCY-MAX**	*(STATE_TPS_DIAG = TPS_LIH_MAX)*
- **TPS-EMERGENCY-X**	*(STATE_TPS_DIAG = TPS_LIH_X)*
- **ETC-LIMP-HOME-1**	*(STATE_ETC_LIH = ETC_LIH_1)*
- **ETC-LIMP-HOME-2-REV**	*(STATE_ETC_LIH = ETC_LIH_2_REV)*
- **ETC-LIMP-HOME-2 (SAS)**	*(STATE_ETC_LIH = ETC_LIH_2)*
- **ETC-LIMP-HOME-3**	*(STATE_ETC_LIH = ETC_LIH_3)*

All **ETC-LIMP-HOME-..** have a higher priority then all **TPS-EMERGENCY-..**

Application conditions:

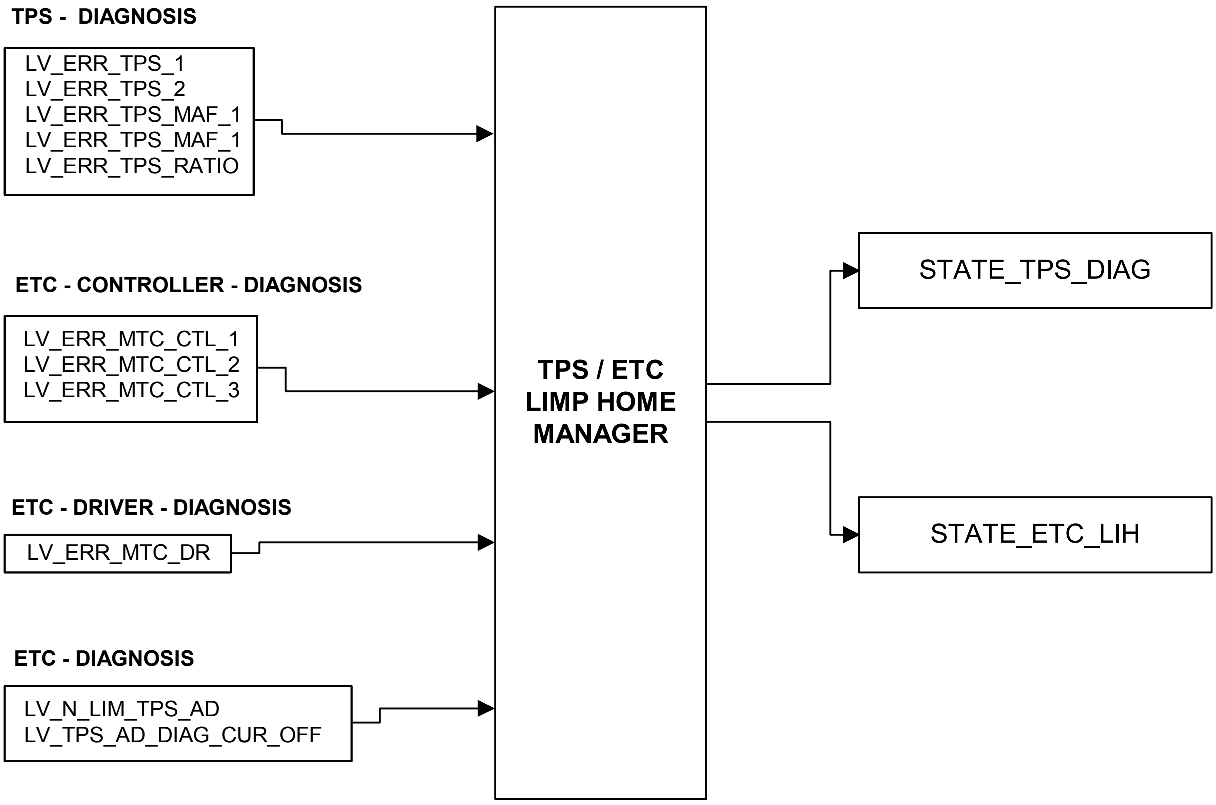
Initialization: all outputs = 0 at LV_IGK 0->1 or reset or at clearing error memory
LV_ERR_TPS according ABC type **STD_INI**

Recurrence: 5ms

Activation: LV_IGK = 1
Deactivation: LV_IGK = 0 (set LV_CDN_DIAG_TPS = 0)

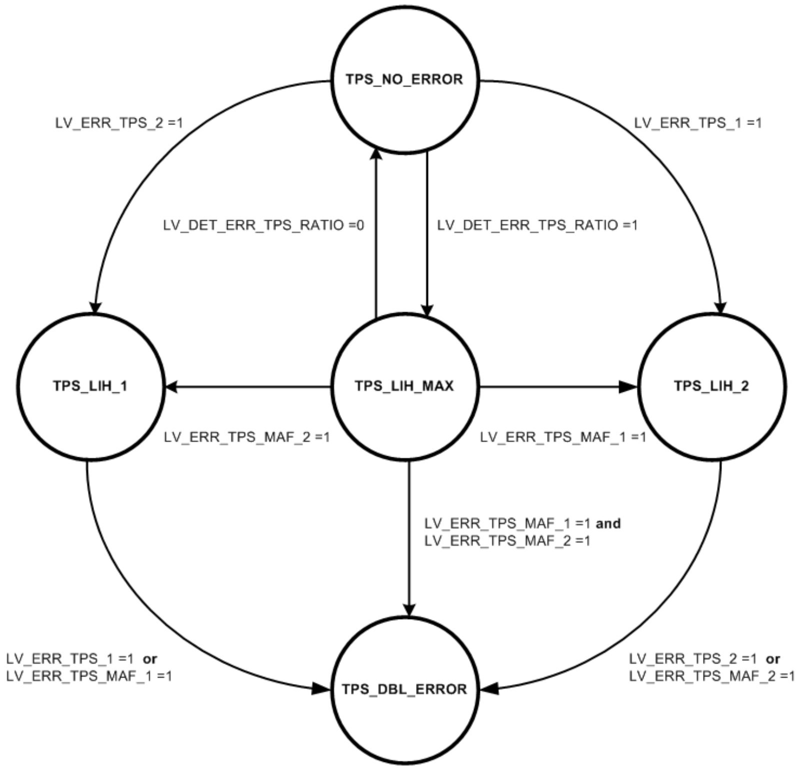
Signal flow diagram:

TPS / ETC Limp Home Manager



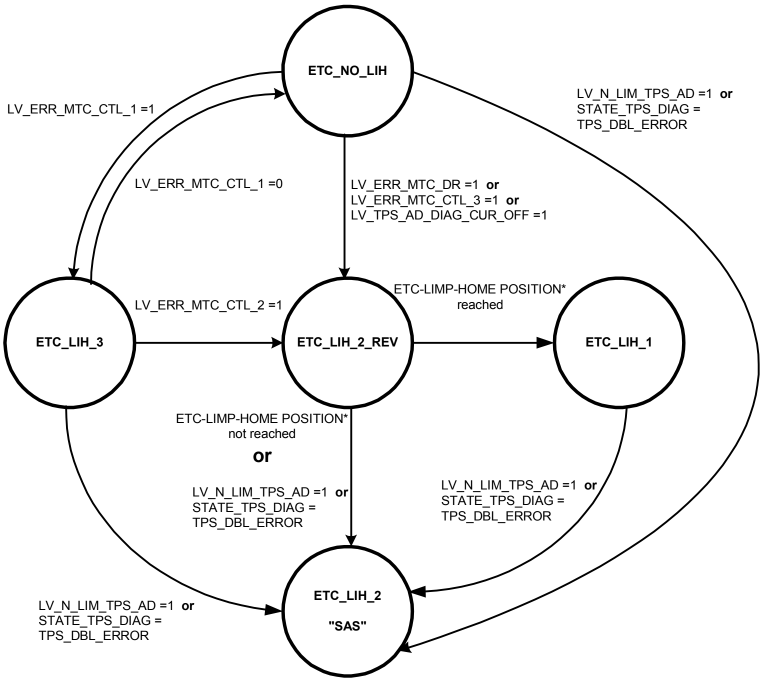
Signal flow diagram:

STATE_TPS_DIAG



Signal flow diagram:

STATE_ETC_LIH



* **ETC-LIMP-HOME POSITION**
reached =

I V_TPS_X - TPS_AD_LIH_X I < C_V_TPS_LIH_CHK_HYS
within C_T_DYW_TPS_LIH_CHK

(X = channel1 or 2 or 1/2 depending on LV_ERR_TPS_X)

61.10.1 TPS-EMERGENCY-MAX

(STATE_TPS_DIAG = TPS_LIH_MAX)

Description:

If TPS-Emergency MAX is activ, there is a maximum selection between the two channels (see “Determination of the Throttle Position”).

Formula section:

```
If      LV_DET_ERR_TPS_RATIO = 1
Then    STATE_TPS_DIAG = TPS_LIH_MAX
Endif
```

61.10.2 TPS-EMERGENCY-X

(STATE_TPS_DIAG = TPS_LIH_X)

Description:

If one TPS channel (=Y) is detected as faulty then TPS-EMERGENCY-X is active.
If TPS-EMERGENCY-X is active, the throttle position acqusition is done with a leading Poti-X
(see “Acqusition of the throttle position”).

Formula section:

```
If      LV_ERR_TPS_Y = 1                or
        LV_ERR_TPS_MAF_Y = 1
Then    STATE_TPS_DIAG = TPS_LIH_X
Endif
```

Idle speed demand:

```
If      STATE_TPS_DIAG = TPS_LIH_X            and
        LV_CT = 1
Then    LV_TPS_MTC_N_LIM = 1    (see “Engine speed limit coordination)
Endif
```

61.10.3 ETC-LIMP-HOME-1 / ETC-LIMP-HOME-2-REV

(STATE_ETC_LIH = ETC_LIH_1) / (STATE_ETC_LIH = ETC_LIH_2_REV)

Description:

ETC-LIMP-HOME-1 is demanded by the ETC diagnosis (see “Engine throttle diagnosis and adaptation”) and the ETC driver diagnosis.

During ETC-LIMP-HOME-1 the following limp home measures are defined:

- Throttle valve without power (hold by spring power in the Limp-Home position)
- Engine speed limitation in limp home mode (with considering the drivers request)

ETC-LIMP-HOME-2-REV is initiated reversible until the ETC-Limp Home throttle position (for calibration data and additional information see: “Check of the Limp-Home position” in chapter “Engine throttle diagnosis and adaptation”) has been safely reached.
If the Limp-Home Position is reached ETC-LIMP-HOME-1 is set, if not ETC-LIMP-HOME-2 is set.

During ETC-LIMP-HOME-2_REV the following limp home measures are defined:

- Throttle valve without power (hold by spring power in the Limp-Home position)
- Engine speed limitation in limp home mode (without considering the drivers request)

Formula section:

Activation condition of ETC-LIMP-HOME-1:

If(1) [STATE_ETC_LIH = ETC_NO_LIH **and**
LV_ERR_MTC_DR = 1 **or**
LV_ERR_MTC_CTL_3 = 1 **or**
LV_TPS_AD_DIAG_CUR_OFF = 1]

Or

[STATE_ETC_LIH = ETC_LIH_3 **and**
LV_ERR_MTC_CTL_2 = 1

Then(1)

Activation of ETC-LIMP-HOME-2-REV:

STATE_ETC_LIH = ETC_LIH_2_REV
LV_MTC_CUR_OFF_REQ = 1 (ETC without power request)
LV_N_LIM_ETC_LIH = 1
LV_TPS_MTC_N_LIM = 1 (see “Engine speed limit coordination)

Check of Limp Home position:

Start Time: C_T_MAX_TPS_LIH_CHK

IF(2) T < C_T_MAX_TPS_LIH_CHK **and**
I V_TPS_X – V_TPS_AD_LIH_X I < C_V_TPS_LIH_CHK_HYS
Within the interval: C_T_DYW_TPS_LIH_CHK

Activation of ETC-LIMP-HOME-1/Deactivation of ETC-LIMP-HOME-2-REV:

```
          Then(2)    "Check limp home ok"
                     STATE_ETC_LIH = ETC_LIH_1
                     LV_MTC_LIH_CUR_OFF = 1
                     LV_N_LIM_ETC_LIH = 0
                     LV_TPS_MTC_N_LIM = 0
```

Activation of ETC-LIMP-HOME-2:

```
          Else(2)    STATE_ETC_LIH = ETC_LIH_2
          Endif(2)
Else(1)   STATE_ETC_LIH = ETC_NO_LIH
Endif(1)
```

61.10.4 ETC-LIMP-HOME-2 (SAS)

(STATE_ETC_LIH = ETC_LIH_2)

Description:

ETC-LIMP-HOME-2 (German abbreviation: SAS = Sicherheitsschubabschalten) is demanded by the ETC diagnosis (see "Engine throttle diagnosis and adaptation") and/or the TPS diagnosis (if both TPS channels are faulty). ETC-LIMP-HOME-2 is set unreversible for the engine run.
During ETC-LIMP-HOME-2 the following limp home measures are defined:

- Throttle valve without power (hold by spring power in the Limp-Home position)
- Engine speed limitation in limp home mode (without considering the drivers request)

Formula section:

Activation of ETC-LIMP-HOME-2:

```
If        STATE_ETC_LIH = ETC_NO_LIH          or
          STATE_ETC_LIH = ETC_LIH_1           or
          STATE_ETC_LIH = ETC_LIH_2_REV       or
          STATE_ETC_LIH = ETC_LIH_3
    and
          LV_N_LIM_TPS_AD               or
          [LV_ERR_TPS_1 = 1             and
          LV_ERR_TPS_2 = 1]             or
          [LV_ERR_TPS_MAF_1 = 1         and
          LV_ERR_TPS_MAF_2 = 1]         or
          [LV_ERR_TPS_X = 1             and
          LV_ERR_TPS_MAF_Y = 1]
```

```
Then    STATE_ETC_LIH = ETC_LIH_2
        LV_N_LIM_ETC_LIH = 1
        LV_MTC_CUR_OFF_REQ = 1      (ETC without power request)
        LV_TPS_MTC_N_LIM = 1        (see “Engine speed limit coordination)
Endif
```

61.10.5 ETC-LIMP-HOME-3

(STATE_ETC_LIH = ETC_LIH_3)

Description:

ETC-LIMP-HOME-3 is demanded by the ETC position controller diagnosis (see “Engine throttle diagnosis and adaptation”). ETC-LIMP-HOME-3 is set reversible for the engine run. During ETC-LIMP-HOME-3 the following limp home measures are defined:

- Engine speed limitation in limp home mode (without considering the drivers request)

Formula section:

Activation of ETC-LIMP-HOME-3:

```
If      STATE_ETC_LIH = ETC_NO_LIH            and
        LV_ERR_MTC_CTL_1 = 1
Then    STATE_ETC_LIH = ETC_LIH_3
        LV_N_LIM_ETC_LIH_REV = 1
        LV_TPS_MTC_N_LIM = 1        (see “Engine speed limit coordination)
Else    STATE_ETC_LIH = ETC_NO_LIH
Endif
```

61.10.6 Calculation of LV_ERR_TPS

Description:

LV_ERR_TPS is used for MIL activation and inhibition of functions. The error is set depending the present status of ETC limp-home (combination of operation-mode and present TPS errors).

Formula section:

```
LV_CDN_DIAG_TPS = 1
If      STATE_ETC_LIH = ETC_LIH_1                 or
        STATE_ETC_LIH = ETC_LIH_2_REV             or
        STATE_ETC_LIH = ETC_LIH_2
Then    ERR_SYM_TPS = SYM_2
Else    ERR_SYM_TPS = NO_SYM
Endif
```

LV_ERR_TPS, LV_END_DIAG_TPS is calculated by error management.

61.11 Electrical diagnosis of throttle housing heater relay (LV_ERR_RLY_MTC_HEAT)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_RLY_MTC_HEAT	O/V	0...1H	0...1	1	[-]
Logical value for MTC_HEAT error of power stage					
LV_CDN_DIAG_RLY_MTC_HEAT	O/V	0...1H	0...1	1	[-]
Diagnosis condition MTC_HEAT of power stage					
ERR_SYM_RLY_MTC_HEAT	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom MTC_HEAT of power stage					
LV_END_DIAG_RLY_MTC_HEAT	O/V	0...1H	0...1	1	[-]
End of MTC_HEAT power stage diagnosis					
LV_INH_DIAG_RLY_MTC_HEAT	O/V	0...1H	0...1	1	[-]
Diagnosis inhibition RLY_MTC_HEAT diagnosis					
CDN_DIAG_RLY_MTC_HEAT	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of RLY_MTC_HEAT bit 0: diagnosis condition for symptom SCP (SYM_0)bit 1: diagnosis condition for symptom SCG (SYM_1)bit 2: diagnosis condition for symptom OC (SYM_2)					
ERR_DIAG_RLY_MTC_HEAT	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for RLY_MTC_HEAT (only parameter)					

Input data:

LV_IGK	LV_CDN_VB_MIN_DIAG	LV_ERR_SPI_MPS	

Import actions:

ACTION_ERRM_FilterMulticondition(IN< RLY_MTC_HEAT >, IN< CDN_DIAG_RLY_MTC_HEAT >, IN< ERR_DIAG_RLY_MTC_HEAT >, IN<C_ABC_INC_RLY_MTC_HEAT>, IN<C_ABC_MAX_RLY_MTC_HEAT>, OUT<LV_ERR_RLY_MTC_HEAT>)
ACTION_INFR_GetElDiagMTCHEAT(OUT< Cdn_diag_rly_mtc_heat>, OUT< Err_diag_rly_mtc_heat>)

Note : ACTION_INFR_GetElDiagMTCHEAT() is defined in the IRS (Infrastructure requirement specification AGGR THRO)

FUNCTION DESCRIPTION:

General information:

The signal from the driver which controls the throttle housing heater relay is diagnosed. The relay is driven by the ECU via an output driver. The failure detection is done by ECU Hardware and given to ASW. The purpose of the diagnosis is based on the detection of electrical faults as defined by OBD I requirements.

Description:

After activation conditions are met, the diagnosis activation is delayed for NC_PSD_DLY_RLY_MTC_HEAT executions, to avoid the usage of wrong infrastructure information. The error detection algorithm operates the electrical diagnosis of powerstage outputs of ATIC39.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> Short circuit to Vbatt(SCP)          (= SYM_0)
 │ │ └───> Short circuit to GND(SCG)            (= SYM_1)
 │ └─────> Open load(OC)                        (=SYM_2)
 └───────> -                                    (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

Set delay counter for NC_PSD_DLY_RLY_MTC_HEAT

Recurrence: 100 ms

Activation: LV_IGK = 1

Deactivation: LV_IGK = 0

Formula section:

If diagnosis is not inhibited (LV_INH_DIAG_RLY_MTC_HEAT = 0) the diagnosis conditions are set according infrastructure information: CDN_DIAG_RLY_MTC_HEAT.
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_RLY_MTC_HEAT.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_RLY_MTC_HEAT and ERR_DIAG_RLY_MTC_HEAT.

ACTION_ERRM_FilterMulticondition (IN<RLY_MTC_HEAT>, IN<CDN_DIAG_RLY_MTC_HEAT>, IN<ERR_DIAG_RLY_MTC_HEAT>, IN<C_ABC_INC_RLY_MTC_HEAT>, IN<C_ABC_MAX_RLY_MTC_HEAT>, OUT<LV_ERR_RLY_MTC_HEAT>)

This algorithm determines:

ERR_SYM_RLY_MTC_HEAT	(Detected error symptom)
LV_ERR_RLY_MTC_HEAT	(Error flag for debounced error)
LV_CDN_DIAG_RLY_MTC_HEAT	(Diagnosis condition information)
LV_END_DIAG_RLY_MTC_HEAT	(End of diagnosis information)

Diagnosis inhibition:

```
If      LV_IGK = 1                    and
        LV_CDN_VB_MIN_DIAG = 1        and
        LV_ERR_SPI_MPS = 0
Then    LV_INH_DIAG_RLY_MTC_HEAT = 0
Else    LV_INH_DIAG_RLY_MTC_HEAT = 1
Endif
```

```
If      LV_INH_DIAG_RLY_MTC_HEAT = 0

Then
```

Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_RLY_MTC_HEAT and basic diagnosis conditions CDN_DIAG_RLY_MTC_HEAT) received from the infrastructure:

ACTION_INFR_GetElDiagMTCHEAT (OUT<Cdn_diag_rly_mtc_heat>, OUT<Err_diag_rly_mtc_heat>)

Basic diagnosis conditions are set according infrastructure information: CDN_DIAG_RLY_MTC_HEAT
Failure symptoms (raw value) are set according infrastructure information: ERR_DIAG_RLY_MTC_HEAT

```
        If Activation conditions are met for the NC_PSD_DLY_RLY_MTC_HEAT
           recurrence of the diagnosis
        Then  [CDN_DIAG_RLY_MTC_HEAT valid from Infrastructure]
        Else  CDN_DIAG_RLY_MTC_HEAT = 0
        Endif
Else
        CDN_DIAG_RLY_MTC_HEAT = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_RLY_MTC_HEAT	1	0...FFH	0...255	1	[-]
Debounce counter increment MTC_HEAT					
C_ABC_MAX_RLY_MTC_HEAT	1	1...FFH	1...255	1	[-]
Debounce counter maximum value MTC_HEAT					

Configuration for diagnostic symptoms :

Diagnostic **RLY_MTC_HEAT**	**Symptom description**	**Symptom**	**Filter type**
Diagnostic description	*SCP*	*SYM_0*	*MPL_STD_INI*
	SCG	*SYM_1*	
	OC	*SYM_2*	

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_RLY_MTC_HEAT	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (= 2)					

61.12 ETC Jam detection

General information:

The described module is used for the detection or the removing of ice between throttle flap and body.

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_MTC_HEAT_ACT	V	0...FH	0...15	1	[-]
Counter active cycle throttle housing heater					
CTR_RLY_MTC_HEAT	V	0...FFFFH	0...6553.5	0.1	[s]
Counter throttle housing heater relay active					
CTR_TPS_JAM_DET_ACT	O/V/S	0...FFFFH	0...65535	1	[-]
The variable indicates the number of activations and is saved in the non-volatile memory					
CTR_TPS_JAM_INH	V	0...FFH	0...255	1	[-]
Inhibition counter locks the ice-breaking functionality for a defined time					
CTR_TPS_JAM_PER	V	0...FFH	0...255	1	[-]
Counter variable indicates the current number of rectangle pulses					
CTR_TPS_JAM_PER_MAX	V	0...FFH	0...255	1	[-]
The counter variable includes the half period of the rectangle pulse					
CTR_TPS_JAM_PLS	V	0...FFH	0...255	1	[-]
The counter variable is used for the calculation of the rectangle pulses					
ERR_SYM_TPS_JAM_DET	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
For each symptom : status of failure (set to 1 when failure symptom is detected)					
LV_CDN_DIAG_TPS_JAM_DET	V	0...1H	0...1	1	[-]
Diagnostic condition to start symptom detection (set to 1 when condition is fulfilled)					
LV_END_DIAG_TPS_JAM_DET	V	0...1H	0...1	1	[-]
Diagnostic done completely at least one time					
LV_ERR_TPS_JAM_DET	O/V	0...1H	0...1	1	[-]
Present safety critical failure without filtering of ETC jam diagnostic					
LV_MTC_HEAT_READY	O/V	0...1H	0...1	1	[-]
Throttle housing heater cycle finished (1:engine start enabled, 0: engine start disabled)					
LV_N_LIM_TPS_JAM_REQ	O/V	0...1H	0...1	1	[-]
Logical variable requests engine speed limitation from the module ETC limp-home management					
LV_RLY_MTC_HEAT	O/V	0...1H	0...1	1	[-]
Throttle housing heater relay (1:activated, 0:deactivated)					
LV_TPS_JAM_DET_DEAC	V	0...1H	0...1	1	[-]
Deactivation condition of the ETC jam detection					
LV_TPS_JAM_OPEN_DIR	O/V	0...1H	0...1	1	[-]
Logical variable indicates a disturbance of the throttle flap in opening direction					
LV_TPS_SP_JAM	O/V	0...1H	0...1	1	[-]
Logical variable indicates a throttle position setpoint request for ETC jam detection					
STATE_MTC_HEAT	O/V	0H 1H 2H 3H 4H	HEAT_OFF HEAT_OFF_DLY HEAT_ON_DLY HEAT_ON HEAT_EXT_ADJ	1	[-]
STATE throttle housing heater relay					
STATE_TPS_JAM_DET	O/V	0H 1H 2H	INIT ICE_BREAK SP_LIMIT	1	[-]
State variable indicates the current state of the ETC jam detection					
TPS_JAM_ACT	V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Throttle position indicates the location of the jammed position					
TPS_JAM_PLS	V	C001...3FFFH	-119.5...119.5	7.2941e-3	[°TPS]
Height of the rectangle pulse during active ice-breaking					
TPS_SP_JAM	O/V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Throttle position setpoint request in case of a jammed ETC actuator					
TPS_SP_JAM_TMP	-	C001...3FFFH	-119.5...119.5	7.2941e-3	[°TPS]
Throttle position setpoint request in case of a jammed ETC actuator					

Input data:

ECU_STATE	LV_ACT_RLY_MTC_HEAT_EXT_ADJ	LV_AST_END	LV_CTR_TPS_JAM_DET_ACT_EXT_ADJ
LV_ERR_RLY_MTC_HEAT	LV_IGK	LV_KEY_VLD	LV_RLY_MTC_HEAT_EXT_ADJ
LV_ST	LV_TPS_JAM_DET_ACT	LV_TPS_JAM_DET_DI	T_MTCPWM_PI_DIAG
TCO	TIA	TPS_AV	TPS_DIF
TPS_SP_MDL			

Import actions:

ACTION_INFR_SetMTCHEAT(IN <lv_rly_mtc_heat>**)**
This action sets the output pin for the "MTC_HEAT Switch"

Note: The imported actions are defined in the THRO – IRS (infrastructure requirement specification).

ACTION_ERRM_NoFilterSymptom(**IN**< *XX* >, **IN**< lv_cdn_diag_XX >, **IN**< err_sym_XX >, **IN**< lv_err_set_XX >, **IN**< lv_err_reset_XX >, **IN**< lv_end_diag_XX >, **OUT**< LV_ERR_XX >)

61.12.1 Throttle housing heater relay control

General information:

Under very cold conditions it is possible that the throttle valve ices due to the humidity in the blow by gas. Additionally after engine stop a accumulation of water is possible between flap and housing. When the dew point drops the water freezes and the detection of the ETC bottom limit is disturbed. The described module is used for the removing of ice between throttle flap and body.

To ensure the function of the ETC the housing is heated before engine start. The ECU controls a relay through the Pin A_S_BBH. This is done immediately after unlocking the vehicle and the wake up of the ECU. If the ECU is locked after powerlatch no control is done.

The heater relay is deactivated.

In order to take care of the battery capacity the heater is only activated in defined time intervals. An often re-activation shall be avoid.

Hint: The output of the air temperature sensor VP_TIA and cooling water sensor VP_TCO are calculated by using the action infrastructure ACTION_INFR_GetVpTia (VP_TIA) and ACTION_INFR_GetVpTco (VP_TCO). To ensure the availability of TIA and TCO in ECU_STATE ***WAKE_UP*** the action calls are active after ECU reset.

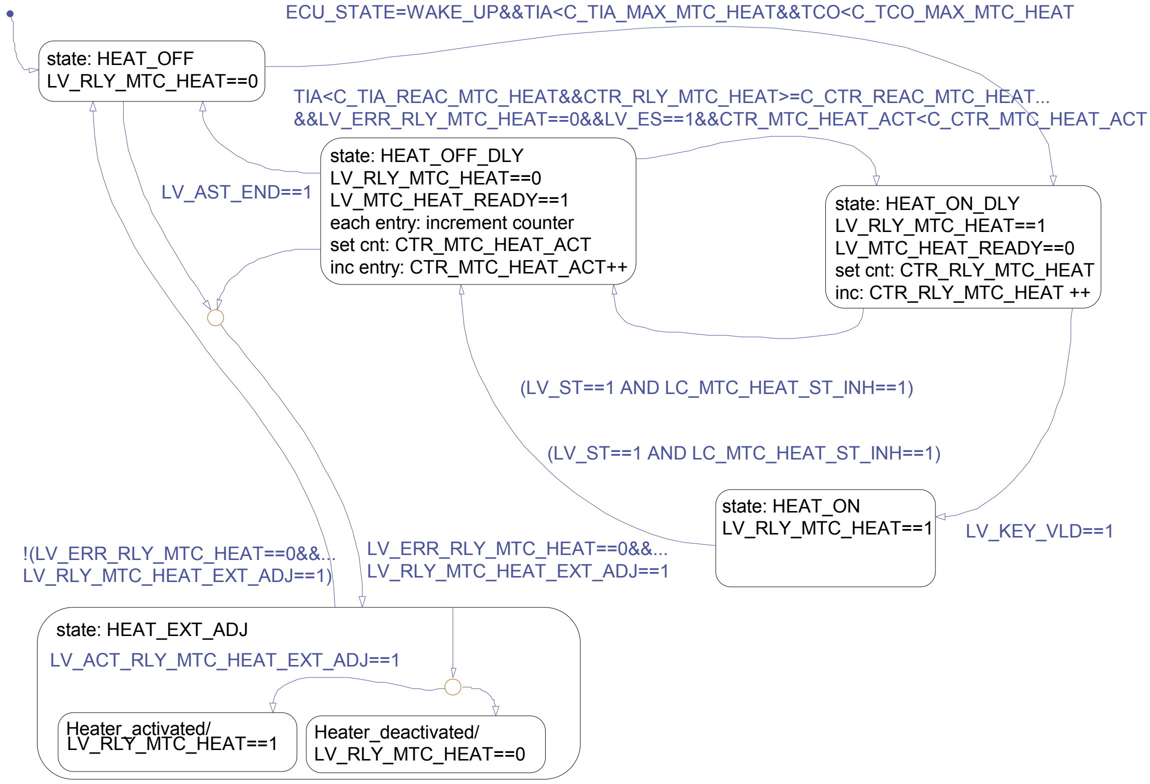
Application conditions:

Initialization:	at reset, all variables are initialized with 0,
	but LV_MTC_HEAT_READY = 1
	set delay counter for C_DLY_RLY_MTC_HEAT
Recurrence:	100 ms
Activation:	after reset, every engine operating state
	Activation conditions are met for the

C_DLY_RLY_MTC_HEAT recurrence

Deactivation:

Signal flow diagram:



Formula section:

see also signal flow diagram

Heater deactivated

STATE_MTC_HEAT = HEAT_OFF

LV_RLY_MTC_HEAT = 0

ACTION_INFR_SetMTCHEAT(IN < LV_RLY_MTC_HEAT >**)**

Exit to **HEAT_ON_DLY**:

(STATE_ECU = WAKE_UP **AND**
TCO < C_TCO_MAX_MTC_HEAT **AND**
TIA < C_TIA_MAX_MTC_HEAT)

Exit to **HEAT_EXT_ADJ**:

(LV_ERR_RLY_MTC_HEAT = 0 **AND**
LV_RLY_MTC_HEAT_EXT_ADJ = 1)

Heater previous activated

STATE_MTC_HEAT = HEAT_ON_DLY

LV_RLY_MTC_HEAT = 1

LV_MTC_HEAT_READY = 0

ACTION_INFR_SetMTCHEAT(IN < LV_RLY_MTC_HEAT >**)**

set timer: CTR_RLY_MTC_HEAT = 0

increment timer continuously: CTR_RLY_MTC_HEAT + 1

Hint: Timer is active as long as STATE_HEAT_OFF is reached

Exit to **HEAT_OFF_DLY**:

(LV_ST = 1 AND LC_MTC_HEAT_ST_INH = 1)

Exit to **HEAT_ON**:

LV_KEY_VLD = 1

Heater activated

STATE_MTC_HEAT = HEAT_ON

LV_RLY_MTC_HEAT = 1

ACTION_INFR_SetMTCHEAT(IN < LV_RLY_MTC_HEAT >**)**

Exit to **HEAT_OFF_DLY**:

(LV_ST = 1 AND LC_MTC_HEAT_ST_INH = 1)

Heater previous deactivated

STATE_MTC_HEAT = HEAT_OFF_DLY

LV_RLY_MTC_HEAT = 0

LV_MTC_HEAT_READY = 1

set counter: CTR_MTC_HEAT_ACT = 0

increment only each entry: CTR_MTC_HEAT_ACT + 1

ACTION_INFR_SetMTCHEAT(IN < LV_RLY_MTC_HEAT >**)**

Exit to **HEAT_OFF**:

LV_AST_END = 1

Exit to **HEAT_ON_DLY**:

(TIA < C_TIA_REAC_MTC_HEAT **AND**

CTR_RLY_MTC_HEAT > C_CTR_REAC_MTC_HEAT **AND**

LV_ERR_RLY_MTC_HEAT = 0 **AND**

LV_ES = 1 **AND**

CTR_MTC_HEAT_ACT < C_CTR_MTC_HEAT_ACT)

Exit to **HEAT_EXT_ADJ**:

(LV_ERR_RLY_MTC_HEAT = 0 **AND**

LV_RLY_MTC_HEAT_EXT_ADJ = 1)

External adjustment

STATE_MTC_HEAT = HEAT_EXT_ADJ

If LV_ACT_RLY_MTC_HEAT_EXT_ADJ = 1

Then LV_RLY_MTC_HEAT = 1

Else LV_RLY_MTC_HEAT = 0

Endif

ACTION_INFR_SetMTCHEAT(IN < LV_RLY_MTC_HEAT >**)**

Exit to **HEAT_OFF**:

(LV_ERR_RLY_MTC_HEAT = 1 **OR**

LV_RLY_MTC_HEAT_EXT_ADJ = 0)

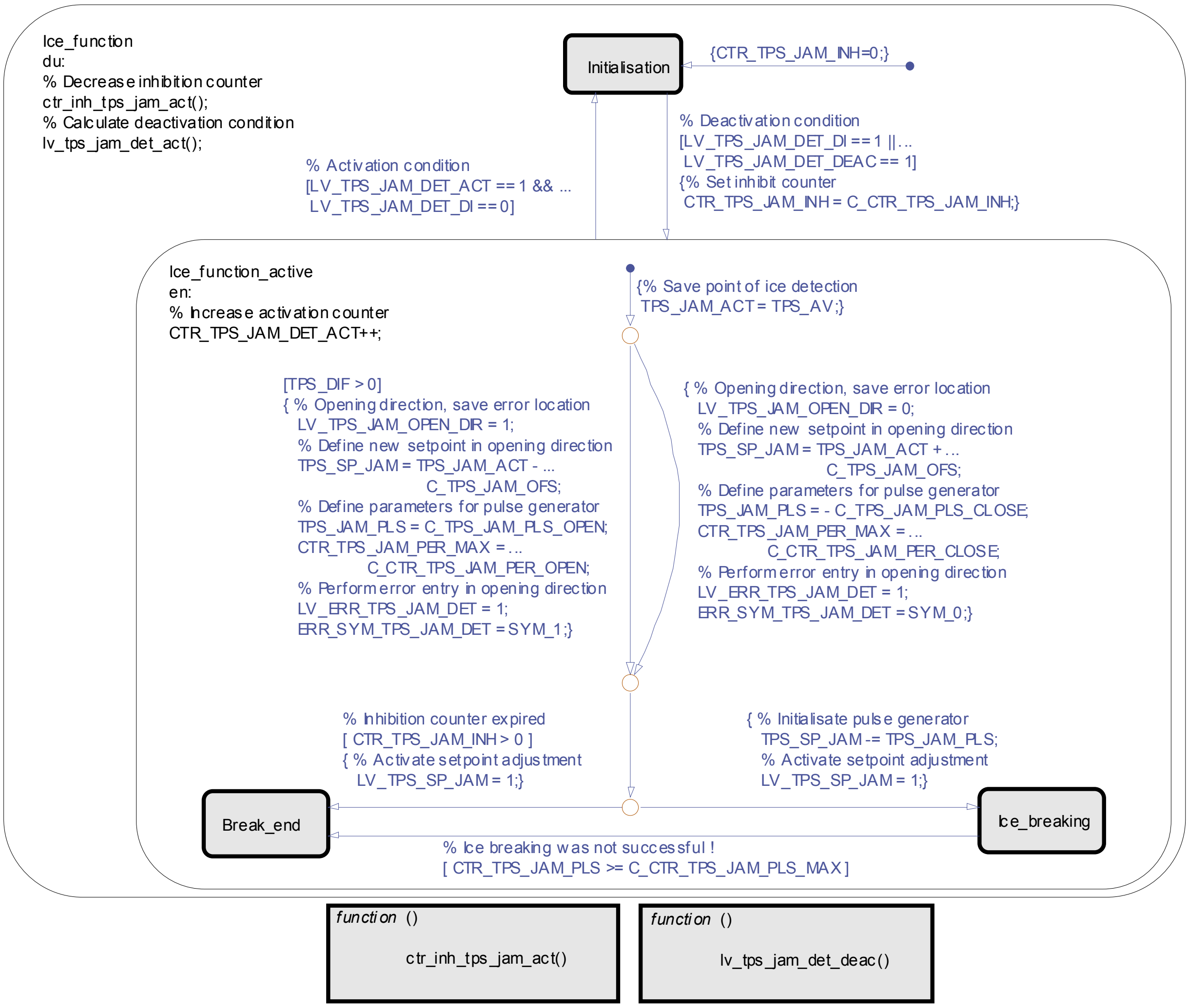
61.12.2 ETC ice breaking

FUNCTION DESCRIPTION:

General information:

In case of very low temperatures it is possible that the throttle actuator is jammed by ice. The described module is used for the detection and the removing of ice between throttle flap and body. The functionality is able to detect the location of the throttle flap disturbance. We can distinguish ice in closing and opening throttle flap direction.
The ice shall be removed with a variation of the setpoint for the digital position controller. CTR_TPS_JAM_DET_ACT is used to count these events. It needs to be saved reset resistant AND in NVMY !

Signal flow diagram:



```
Ice_function
du:
% Decrease inhibition counter
ctr_inh_tps_jam_act();
% Calculate deactivation condition
lv_tps_jam_det_act();

{CTR_TPS_JAM_INH=0;}

% Activation condition
[LV_TPS_JAM_DET_ACT == 1 && ...
 LV_TPS_JAM_DET_DI == 0]

% Deactivation condition
[LV_TPS_JAM_DET_DI == 1 || ...
 LV_TPS_JAM_DET_DEAC == 1]
{% Set inhibit counter
 CTR_TPS_JAM_INH = C_CTR_TPS_JAM_INH;}

Ice_function_active
en:
% Increase activation counter
CTR_TPS_JAM_DET_ACT++;

{% Save point of ice detection
 TPS_JAM_ACT = TPS_AV;}

[TPS_DIF > 0]
{ % Opening direction, save error location
  LV_TPS_JAM_OPEN_DIR = 1;
  % Define new setpoint in opening direction
  TPS_SP_JAM = TPS_JAM_ACT - ...
                    C_TPS_JAM_OFS;
  % Define parameters for pulse generator
  TPS_JAM_PLS = C_TPS_JAM_PLS_OPEN;
  CTR_TPS_JAM_PER_MAX = ...
               C_CTR_TPS_JAM_PER_OPEN;
  % Perform error entry in opening direction
  LV_ERR_TPS_JAM_DET = 1;
  ERR_SYM_TPS_JAM_DET = SYM_1;}

{ % Opening direction, save error location
  LV_TPS_JAM_OPEN_DIR = 0;
  % Define new setpoint in opening direction
  TPS_SP_JAM = TPS_JAM_ACT + ...
                    C_TPS_JAM_OFS;
  % Define parameters for pulse generator
  TPS_JAM_PLS = - C_TPS_JAM_PLS_CLOSE;
  CTR_TPS_JAM_PER_MAX = ...
            C_CTR_TPS_JAM_PER_CLOSE;
  % Perform error entry in opening direction
  LV_ERR_TPS_JAM_DET = 1;
  ERR_SYM_TPS_JAM_DET = SYM_0;}

% Inhibition counter expired
[ CTR_TPS_JAM_INH > 0 ]
{ % Activate setpoint adjustment
  LV_TPS_SP_JAM = 1;}

{ % Initialisate pulse generator
  TPS_SP_JAM -= TPS_JAM_PLS;
  % Activate setpoint adjustment
  LV_TPS_SP_JAM = 1;}

% Ice breaking was not successful !
[ CTR_TPS_JAM_PLS >= C_CTR_TPS_JAM_PLS_MAX ]
```

61.12.3 Diagnosis conditions and error symptom definition

Description:

The diagnostic conditions of the ETC jam detection are true as long as the general disable condition isn't set. The error symptoms are defined for this diagnosis as:

ERR_SYM_TPS_JAM_DET

```
  |_|_|_|_|
   |  |  |  └─> TPS jammed in closing direction          (= SYM_0)
   |  |  └────> TPS jammed in opening direction          (= SYM_1)
   |  └───────> -                                        (= SYM_2)
   └──────────> -                                        (= SYM_3)
```

Application conditions:

Initialisation: at IGKON or RST set all error variables ...TPS_JAM_DET to zero (done by the error – management function)

Recurrence: 20 ms

Activation: LV_IGK == 1

Deactivation: When the activation condition isn’t fulfilled !

Formula section:

```
if    LV_TPS_JAM_DET_DI == 0
then  { Set condition and end of diagnosis information }
      LV_CDN_DIAG_TPS_JAM_DET      = 1
      lv_end_diag_tps_jam_det      = 1
      ERR_SYM_TPS_JAM_DET( k )     = ERR_SYM_TPS_JAM_DET( k-1 )
else  { remove condition diagnosis information }
      LV_CDN_DIAG_TPS_JAM_DET      = 0
endif
```

61.12.4 Control condition for ETC jam detection

Description:

The following section describes the deactivation condition and the continuously part of the ETC jam detection. The aim is to control the state machine and the ice-breaking function.

Application conditions:

Initialisation: at IGKON or RST set all to zero

Recurrence: 20 ms

Activation: LV_IGK == 1

Deactivation: When the activation condition isn't fulfilled !

Formula section:

```
{ Calculate blocking time for active ice-breaking }
if      CTR_TPS_JAM_INH > 0
```

```
then    CTR_TPS_JAM_INH – = 1H        { counter is decremented }
endif

{ Calculate deactivation condition for ETC jam detection }
if      LV_TPS_JAM_OPEN_DIR == 1
then    { Deactivation in opening direction }
        LV_TPS_JAM_DET_DEAC = ( ...
                                TPS_SP_MDL  < TPS_JAM_ACT – C_TPS_JAM_HYS     or
                                TPS_AV      > TPS_JAM_ACT + C_TPS_JAM_HYS     )
else    { Deactivation in closing direction }
        LV_TPS_JAM_DET_DEAC = ( ...
                                TPS_SP_MDL  > TPS_JAM_ACT + C_TPS_JAM_HYS     or
                                TPS_AV      < TPS_JAM_ACT – C_TPS_JAM_HYS )
endif
```

61.12.5 State machine for ETC jam detection

Description:

This section describes the state machine of the ETC jam detection functionality. The state machine is used for the coordination of the different requirements during the detection of jammed ETC throttle actuator. Additionally the function manages all kind of requests to the remaining software system, that are engine speed limitation, throttle position setpoint adjustment and the inhibition of selected functions. The state machine distinguishes the following states:

STATE_TPS_JAM_DET	Note
INIT	The initialisation state initialises all important variables and checks the transition conditions to the other states.
ICE_BREAK	The ice-breaking state is used for an active breaking of ice in the throttle actuator. For this case the throttle position setpoint is varied by a rectangle function. If the ice-breaking is not successful after a defined number of rectangle pulses the function changes into the state throttle position setpoint limitation. During the active ice-breaking engine speed limitation is requested !
BREAK_END	The ice-breaking was not successful or the inhibition counter for ice-breaking isn't yet expired.

Application conditions:

Initialisation: at IGKON or RST set all to zero with exception of CTR_TPS_JAM_DET_ACT and STATE_TPS_JAM_DET = INIT

~~at first power up/on saved RAM lost (or reprogramming): CTR_TPS_JAM_ACT = 0~~

~~at reset: CTR_TPS_JAM_ACT = restored from NVMY~~

Recurrence: 20 ms

Activation: LV_IGK == 1

Deactivation: When the activation condition isn't fulfilled !

Formula section:

```
{ Reset of the activation counter in service }
if    LC_CTR_TPS_JAM_DET_ACT_CLR == 1 or
LV_CTR_TPS_JAM_DET_ACT_EXT_ADJ == 1
Then        { The throttle has been changed in service ! The activation counter is reset ! }
            CTR_TPS_JAM_DET_ACT = 0
EndIf

switch( STATE_TPS_JAM_DET )          { State machine of ETC jam detection ! }
   case   INIT:                      { ETC jam detection is in the initialisation state }
          { Remove engine speed limitation request }
          LV_N_LIM_TPS_JAM_REQ = 0
          { Remove throttle position setpoint request }
          LV_TPS_SP_JAM = 0
          { Remove error entry }
          lv_err_tps_jam_det = 0
          ERR_SYM_TPS_JAM_DET = NO_SYM
          { Initialisation of the counter variables }
          CTR_TPS_JAM_PLS = 0
          CTR_TPS_JAM_PER = 0
          { Calculate transitions to the setpoint limitation and ice-breaking function }
          if     LV_TPS_JAM_DET_ACT == 1      and
                 LV_TPS_JAM_DET_DI == 0
          then   { A jammed ETC actuator has been detected ! At first the throttle position of the
                  ice detection is saved for the further function ! }
                 TPS_JAM_ACT = TPS_AV
                 if      TPS_DIF > 0
                 then    { The throttle actuator is jammed in opening direction }
                         LV_TPS_JAM_OPEN_DIR = 1
                         { Define new setpoint in opening direction }
                         TPS_SP_JAM_TMP = TPS_JAM_ACT – C_TPS_JAM_OFS
                         { Define parameters for throttle position setpoint pulse generator }
                         TPS_JAM_PLS                = C_TPS_JAM_PLS_OPEN
                         CTR_TPS_JAM_PER_MAX = C_CTR_TPS_JAM_PER_OPEN
                         { Perform error entry in opening direction }
                         lv_err_tps_jam_det         = 1
                         ERR_SYM_TPS_JAM_DET = SYM_1
                 else    { The throttle actuator is jammed in closing direction }
                         LV_TPS_JAM_OPEN_DIR = 0
                         { Define new setpoint in closing direction }
                         TPS_SP_JAM_TMP = TPS_JAM_ACT + C_TPS_JAM_OFS
                         { Define parameters for throttle position setpoint pulse generator }
                         TPS_JAM_PLS                = - C_TPS_JAM_PLS_CLOSE
```

```
                        CTR_TPS_JAM_PER_MAX = C_CTR_TPS_JAM_PER_CLOSE
                        { Perform error entry in closing direction }
                        lv_err_tps_jam_det              = 1
                        ERR_SYM_TPS_JAM_DET = SYM_0
                endif
                { Increase activation counter }
                CTR_TPS_JAM_DET_ACT + = 1H
                { Selection between setpoint limitation and active ice-breaking depend on the
                  function inhibition counter ! }
                if      CTR_TPS_JAM_INH > 0
                then    { End of Ice breaking ! }
                        STATE_TPS_JAM_DET = BREAK_END
                else    { Initialisate pulse generator }
                        TPS_SP_JAM_TMP – = TPS_JAM_PLS
                        { Ice-breaking function will be started ! }
                        STATE_TPS_JAM_DET = ICE_BREAK
                endif
                { Activate throttle position setpoint request }
                LV_TPS_SP_JAM = 1
                ~~if LV_TPS_SP_JAM = 0 ›1~~
~~              then    { Counter of  jam detection requests ice-breaking ! }~~
~~                      CTR_TPS_JAM_ACT = CTR_TPS_JAM_ACT + 1~~
                ~~endif~~
        endif
        break
case    ICE_BREAK: { ETC jam detection requests ice-breaking ! }
        { Engine speed limitation is requested only in closing direction ! }
        if      LV_TPS_JAM_OPEN_DIR == 0
        then    LV_N_LIM_TPS_JAM_REQ = 1
        endif
        { Increase period counter of the rectangle pulse sequence }
        CTR_TPS_JAM_PER + = 1H;
        { Generate throttle position setpoint pulse sequence for the active ice-breaking ! }
        if      CTR_TPS_JAM_PER == CTR_TPS_JAM_PER_MAX
        then    { Half period of the rectangle pulse is finished, change throttle position setpoint }
                TPS_SP_JAM_TMP + = 2 * TPS_JAM_PLS
        endif
        if      CTR_TPS_JAM_PER >= 2 * CTR_TPS_JAM_PER_MAX
        then    { Increase pulse counter }
                CTR_TPS_JAM_PLS + = 1H;
                { Calculate transition to the setpoint limitation function }
                if      CTR_TPS_JAM_PLS >= C_CTR_TPS_JAM_PLS_MAX
                then    { The maximum pulse number has been performed, the function has to
                go in End of Ice breaking, ice-breaking was not successful ! }
                        STATE_TPS_JAM_DET = BREAK_END
                        { Set final throttle position setpoint }
                        TPS_SP_JAM_TMP – = TPS_JAM_PLS
```

```
                else    { Full period of the rectangle pulse is finished, change throttle position
                         setpoint }
                        TPS_SP_JAM_TMP – = 2 * TPS_JAM_PLS
                        { Reset period counter of the rectangle pulse sequence }
                        CTR_TPS_JAM_PER = 0H;
                endif
            endif
            { Calculate transition to the initialisation state }
            if      LV_TPS_JAM_DET_DI      == 1     or
                    LV_TPS_JAM_DET_DEAC    == 1
            then    { The ETC jam detection is deactivated !  }
                    STATE_TPS_JAM_DET = INIT
                    { Initialise inhibition counter for the next activation }
                    CTR_TPS_JAM_INH = C_CTR_TPS_JAM_INH
            endif
            break
            case    BREAK_END: { End of Ice breaking }
            { ETC jam detection is not requested ! }
            LV_TPS_SP_JAM = 0
            { Engine speed limitation is not requested ! }
            LV_N_LIM_TPS_JAM_REQ = 0
            { Calculate transition to the initialisation state }
            if      LV_TPS_JAM_DET_DI      == 1     or
                    LV_TPS_JAM_DET_DEAC    == 1
            then    { The ETC jam detection is deactivated !  }
                    STATE_TPS_JAM_DET = INIT
                    { Initialise inhibition counter for the next activation }
                    CTR_TPS_JAM_INH = C_CTR_TPS_JAM_INH
            endif
            break
    default
end switch
{ Finally copy throttle position setpoint ! }
TPS_SP_JAM = TPS_SP_JAM_TMP
```

For error management treatment the following action is called:

ACTION_ERRM_NoFilterSymptom(...
TPS_JAM_DET, LV_CDN_DIAG_TPS_JAM_DET, ERR_SYM_TPS_JAM_DET,
lv_err_tps_jam_det, **~lv_err_tps_jam_det**, **lv_end_diag_tps_jam_det**,
LV_ERR_TPS_JAM_DET)

This algorithm sets
LV_ERR_TPS_JAM_DET and LV_END_DIAG_TPS_JAM_DET
according the diagnostic result and delivers the result to Error Management.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TIA_MAX_MTC_HEAT	1	0...FEH	-48...142.5	0.75	[°C]
Maximal intake air temperature for MTC heater					
C_TIA_REAC_MTC_HEAT	1	0...FEH	-48...142.5	0.75	[°C]
Reactivation intake air temperature for MTC heater					
C_CTR_REAC_MTC_HEAT	1	0...FFFFH	0...6553.5	0.1	[s]
Reactivation time limit for MTC heater					
C_CTR_MTC_HEAT_ACT	1	0...FH	0...15	1	[-]
Number of MTC heater cycle					
LC_MTC_HEAT_ST_INH	1	0...1H	0...1	1	[-]
Logical switch MTC heater at start active (=0 heater active)					
C_TCO_MAX_MTC_HEAT	1	0...FEH	-48...142.5	0.75	[°C]
Maximal cooling water temperature for MTC heater					
C_TPS_JAM_HYS	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Throttle position hysteresis for the deactivation of the ice-function					
C_TPS_JAM_OFS	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Throttle position offset for the new throttle position setpoint definition					
C_TPS_JAM_PLS_OPEN	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Rectangle pulse height in throttle flap opening direction					
C_TPS_JAM_PLS_CLOSE	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Rectangle pulse height in throttle flap closing direction					
C_CTR_TPS_JAM_PER_OPEN	1	1...FFH	1...255	1	[-]
Half period of the rectangle pulse in throttle flap opening direction					
C_CTR_TPS_JAM_PER_CLOSE	1	1...FFH	1...255	1	[-]
Half period of the rectangle pulse in throttle flap closing direction					
C_CTR_TPS_JAM_INH	1	0...FFH	0...255	1	[-]
Inhibition counter for the locking of active ice-breaking					
C_CTR_TPS_JAM_PLS_MAX	1	0...FFH	0...255	1	[-]
Maximum number of possible rectangle pulse					
C_DLY_RLY_MTC_HEAT	1	0...FFH	0...255	1	[-]
Number of recurrences until activation conditions MTC Heater are met (default = 2)					
LC_CTR_TPS_JAM_DET_ACT_CLR	1	0...1H	0...1	1	[-]
Logical constant for initializing the activation counter for jam detection					

61.13 ETC jam detection (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TPS_JAM_DET_ACT	O/V	0...1H	0...1	1	[-]
Activation condition of the ETC jam detection					
LV_TPS_JAM_DET_DI	O/V	0...1H	0...1	1	[-]
Disable condition of the ETC jam detection					
LV_TPS_SP_JAM_OFF	V	0...1H	0...1	1	[-]
Irreversible lock condition after detection of abnormal engine operating state					
CTR_DLY_TPS_JAM	V	0...FFH	0...255	1	[-]
Time delay for the detection of abnormal engine operating state					

Input data:

LV_IGK	LV_IS	TIA_THR	TCO
T_MTCPWM_PI_DIAG	MTCPWM	LV_ERR_TIA_THR	LV_ERR_MAP
LV_ERR_MAF	LV_MTC_CUR_OFF	GEAR	N
VS	CTR_TPS_JAM_DET_ACT		

FUNCTION DESCRIPTION:

General information:

The following module described the application incidence of the ETC jam detection and should be used for the control of the ice-breaking algorithm.

61.13.1 Engine Speed Monitoring for TPS Jam Function being active

General information:

The following functionality is used for the detection of abnormal engine operating state, in this case the ETC jam detection functionality will be locked and the ETC actuator diagnosis must indicate an ETC actuator error.

Application conditions:

Initialisation: at RST set all to zero

Recurrence: 20 ms

Activation: All engine operating states

Deactivation: No deactivation

Formula section:

```
if      LV_IGK == 1
then    if    LV_IS == 1     and
          VS < C_VS_MAX_TPS_JAM     and
          ( N > C_N_MAX_NEUT_TPS_JAM      and     GEAR == 0     or
            N > C_N_MAX_GEAR_TPS_JAM      and     GEAR ~= 0     )
        then         CTR_DLY_TPS_JAM += 1H
        else         CTR_DLY_TPS_JAM = 0
```

```
        endif
        if      GEAR == 0
        then    if      CTR_DLY_TPS_JAM ≥ C_CTR_MAX_NEUT_TPS_JAM
                then    LV_TPS_SP_JAM_OFF = 1       { set for the driving cycle }
                endif
        else    if      CTR_DLY_TPS_JAM ≥ C_CTR_MAX_GEAR_TPS_JAM
                then    LV_TPS_SP_JAM_OFF = 1       { set for the driving cycle }
                endif
        endif
else    { Re-initialize function variables }
        LV_TPS_SP_JAM_OFF = 0; CTR_DLY_TPS_JAM = 0
endif
```

61.13.2 Control condition for ETC jam detection

Description:

The following section describes the control conditions of the ETC jam detection. The aim is to gather all activation and inhibition conditions for an efficient control of the algorithm.

Application conditions:

Initialisation: at RST set all to zero

Recurrence: 20 ms

Activation: All engine operating states

Deactivation: No deactivation

Formula section:

```
{ Calculate activation condition for ETC jam detection }
LV_TPS_JAM_DET_ACT = (  ...
                TIA_THR                 < IP_TIA_MAX_TPS_JAM      and
                | MTCPWM |              > C_PWM_ETC_TPS_JAM       and
                T_MTCPWM_PI_DIAG        > C_T_MIN_DIAG_TPS_JAM    and
                CTR_TPS_JAM_DET_ACT< C_CTR_TPS_JAM_DET_ACT_MAX)

{ Calculate disable condition for ETC jam detection }
LV_TPS_JAM_DET_DI = (  LV_IGK                 == 0      or
                       LV_ERR_TIA_THR         == 1      or
                       LV_ERR_MAF             == 1      or
                       LV_ERR_MAP             == 1      or
                       LV_TPS_SP_JAM_OFF      == 1      or
                       LV_MTC_CUR_OFF         == 1      )
```

Configuration for diagnostic symptoms:

Diagnostic TPS_JAM_DET	Symptom description	Symptom	Filter type
ETC jam detection	*TPS jammed in closing direction*	*SYM_0*	*NO*
	TPS jammed in opening direction	*SYM_1*	
	-	*SYM_2*	
	-	*SYM_3*	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TIA_MAX_TPS_JAM	4	0...FEH	-48...142.5	0.75	[°C]
LDP_TCO_IP_TIA_MAX_TPS_JAM	4	0...FEH	-48...142.5	0.75	[°C]
TIA activation threshold for ETC jam detection dependent on TCO					
C_T_MIN_DIAG_TPS_JAM	1	0...FFH	0...2.55	0.01	[s]
Debounce timer activation threshold for TPS jam detection function					
C_PWM_ETC_TPS_JAM	1	0...7FFFH	0...99.99694	3.0518e-3	[%]
PWM_ETC activation threshold for ETC jam detection function					
C_N_MAX_GEAR_TPS_JAM	1	0...1FE0H	0...8160	1	[rpm]
Maximum engine speed for TPS jam detection being active with gear engaged					
C_N_MAX_NEUT_TPS_JAM	1	0...1FE0H	0...8160	1	[rpm]
Maximum engine speed for TPS jam detection being active with gear not engaged					
C_CTR_MAX_NEUT_TPS_JAM	1	0...FFH	0...255	1	[-]
Counter maximum for TPS jam detection being active with gear not engaged and engine speed too high					
C_CTR_MAX_GEAR_TPS_JAM	1	0...FFH	0...255	1	[-]
Counter maximum for TPS jam detection being active with gear engaged and engine speed too high					
C_VS_MAX_TPS_JAM	1	0...FFH	0...255	1	[km/h]
Maximum vehicle speed for TPS jam detection being active					
C_CTR_TPS_JAM_DET_ACT_MAX	1	0...FFFFH	0...65535	1	[-]
Counter maximum for TPS jam detection being active to prevent ETC damage					

61.14 Throttle Position Setpoint Selection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TPS_SP	V/O	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Throttle position setpoint for MTC position controller					
TPS_SP_SEL	O/V	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Selected throttle position setpoint before limitation					
TPS_SP_LIM	V	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Limited throttle position setpoint					
TPS_SP_LIM_1	V	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Throttle position setpoint after absolute value limitation					
TPS_SP_LIM_2	V	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Throttle position setpoint after gradient limitation over whole operating range					

Input data:

TPS_AV	TPS_LIH	TPS_SP_AD	TPS_SP_ES
TPS_SP_EXT	TPS_SP_MDL	TPS_SP_PWL	LV_ES
LV_TPS_SP_LIH	LV_TPS_PWL	LV_TPS_AD_ACT	LV_TPS_AD_REQ
LV_TPS_SP_EXT	LV_TPS_SP_JAM	TPS_SP_JAM	

FUNCTION DESCRIPTION:

General information:

This module is describing the generation of the throttle position setpoint TPS_SP which is used as an input variable in the module ETC Position Controller.

The module consists of two blocks (see figure 1): In a first step the setpoint is selected depending on the engine state, in a second step the selected value is limited. This limitation acts as protection so that the throttle blade does not hit the mechanical stops with too high energy. Otherwise the throttle could be damaged.

The setpoint before limitation TPS_SP_SEL is selected from the following variables:

- Basic throttle position setpoint: TPS_SP_MDL (output of BMW OBJ EISY)
- Throttle position setpoint for engine stopped: TPS_SP_ES
- Throttle position setpoint for jam detection: TPS_SP_JAM
- Throttle position setpoint during adaptation: TPS_SP_AD
- Throttle position setpoint in case of external activation: TPS_SP_EXT
- Throttle position setpoint during the post operating phase: TPS_SP_PWL
- Throttle position setpoint when the position controller is switched off: TPS_LIH

From TPS_SP_SEL the final setpoint TPS_SP is generated by means of a limitation function which is including a limitation by absolute values and by gradients.

The setpoint is determined every 10 ms, except during adaptation (5ms).

Signal flow diagram:

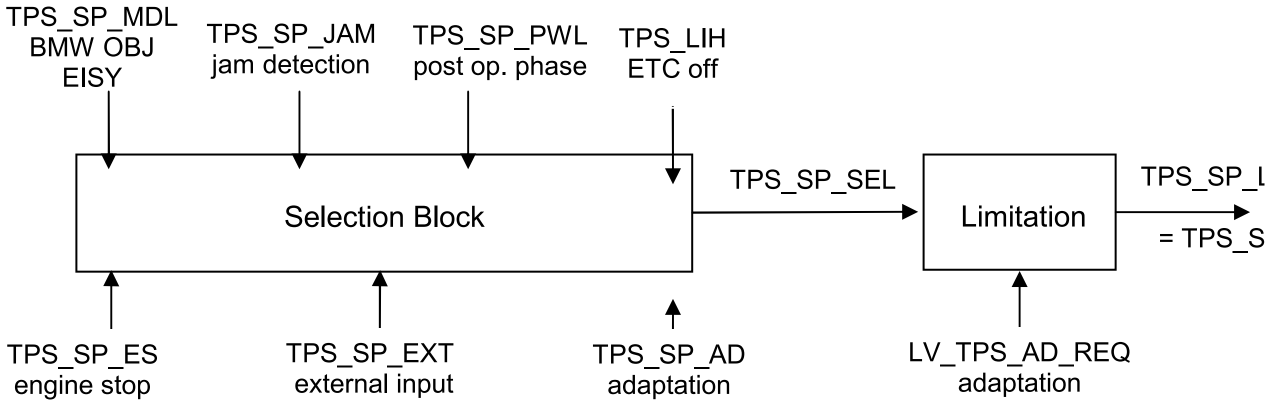


Figure 1: Throttle position setpoint selection and limitation.

Application conditions:

Activation: at every engine state

Deactivation: -

Initialisation: -

Update rate: 10 ms (5ms during adaptation)

61.14.1 Throttle position setpoint selection

For the selection logic see figure 2.

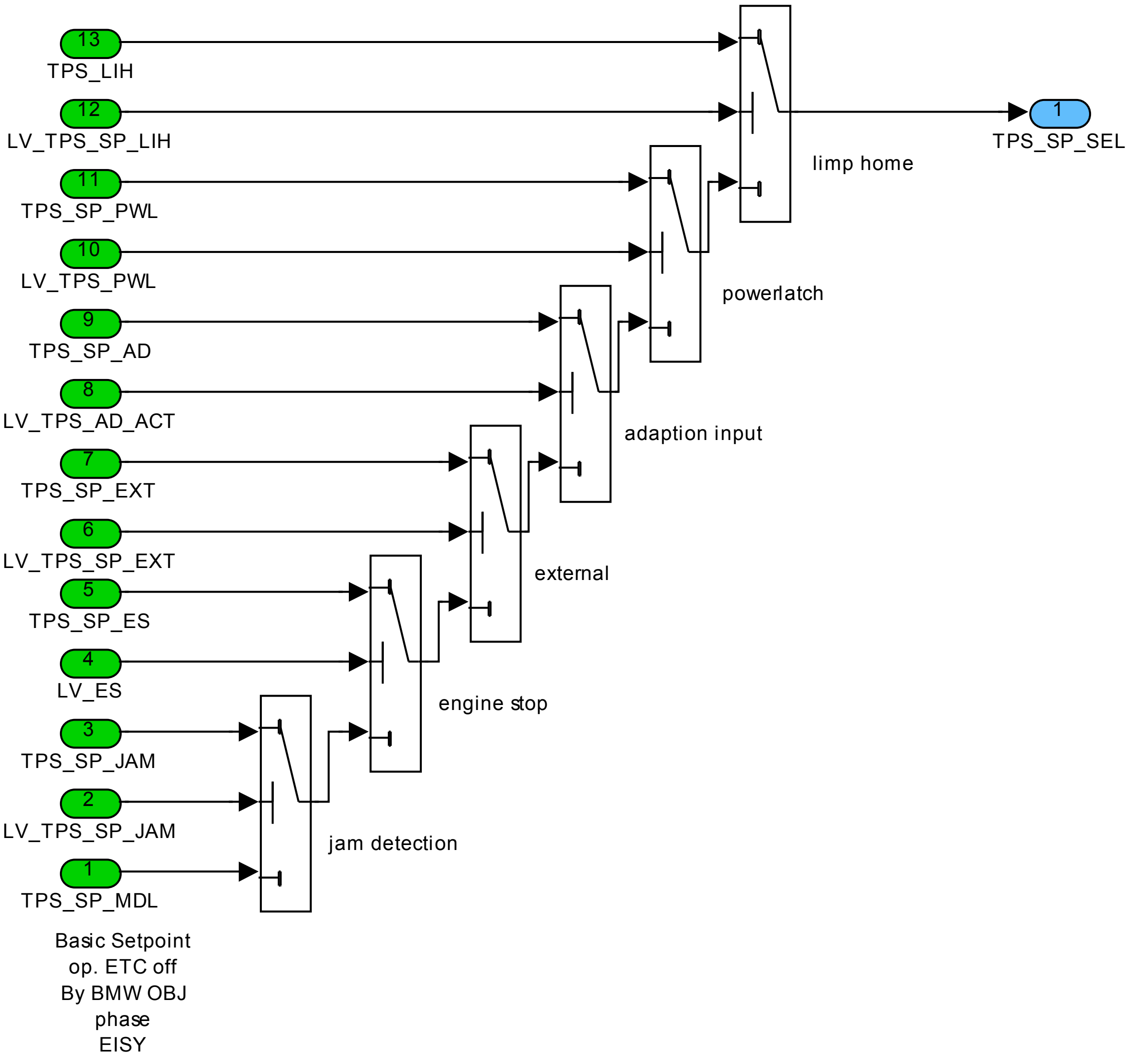


Figure 2: Throttle position setpoint selection block.

61.14.1.1 Throttle position setpoint for jam detection

During jam detection the throttle position setpoint is set from the jam detection module.

Formula section:

```
if      (LV_TPS_SP_JAM = 1          and
        LV_ES = 0                   and
        LV_TPS_SP_EXT = 0           and
        LV_TPS_AD_ACT= 0                    and
        LV_TPS_PWL = 0              and
        LV_TPS_SP_LIH = 0 )

then    TPS_SP_SEL = TPS_SP_JAM

else    TPS_SP_SEL = TPS_SP_MDL

endif
```

61.14.1.2 Throttle position setpoint for engine stopped

During the engine state “Engine stopped” (LV_ES = 1) the throttle position setpoint is determined by the special variable TPS_SP_ES.

Formula section:

```
if      (LV_ES = 1                  and
        LV_TPS_SP_EXT = 0           and
        LV_TPS_AD_ACT= 0                    and
        LV_TPS_PWL = 0              and
        LV_TPS_SP_LIH = 0 )

then    TPS_SP_SEL = TPS_SP_ES

endif
```

61.14.1.3 Throttle position setpoint by external device

In case of request by an external device the throttle position setpoint is set by the value TPS_SP_EXT.

Formula section:

```
if      ( LV_TPS_SP_EXT = 1         and
        LV_TPS_AD_ACT= 0                    and
        LV_TPS_PWL = 0              and
        LV_TPS_SP_LIH = 0)

then    TPS_SP_SEL = TPS_SP_EXT

endif
```

61.14.1.4 Throttle position setpoint during adaptation

During TPS adaptation the throttle position setpoint is set from the adaptation module.

Formula section:

```
if      (LV_TPS_AD_ACT= 1          and
        LV_TPS_PWL = 0             and
        LV_TPS_SP_LIH = 0 )

then    TPS_SP_SEL = TPS_SP_AD

endif
```

61.14.1.5 Throttle position setpoint during post operating phase

To avoid that for ‘Pin 15 Off’ the throttle blade is currentless and set to limp home position, a throttle position setpoint during the post operating phase is available.

Formula section:

```
if      ( LV_TPS_PWL = 1           and
        LV_TPS_SP_LIH = 0 )

then    TPS_SP_SEL =  TPS_SP_PWL

endif
```

61.14.1.6 Throttle position setpoint for ETC controller switched off

In case of the ETC controller switched off the TPS setpoint is set to the limp home position TPS_LIH.

<u>**Formula section:**</u>

```
if      LV_TPS_SP_LIH = 1

then    TPS_SP_SEL = TPS_LIH

endif
```

61.14.2 Throttle position setpoint limitation

The task of this function is a protection mechanism for the throttle blade. Therefore, it must be evaluated <u>after</u> the TPS setpoint selection.

For adjusting the setpoint to the physics of the throttle the variable TPS_SP_SEL is limited by

a) absolute values, for the non adapted throttle (LV_TPS_AD_REQ =1 remains set after adaptation) the values indicated by "(_TPS_AD_REQ)" are used,
b) a gradient valid for the whole operating range,
c) a gradient valid for the range of upper and lower stop.

The intermediate results of the limitation actions mentioned above are stored in the following way: a) b) c)
TPS_SP_SEL --> TPS_SP_LIM_1 --> TPS_SP_LIM_2 --> TPS_SP_LIM = TPS_SP

For illustration see figure 3.

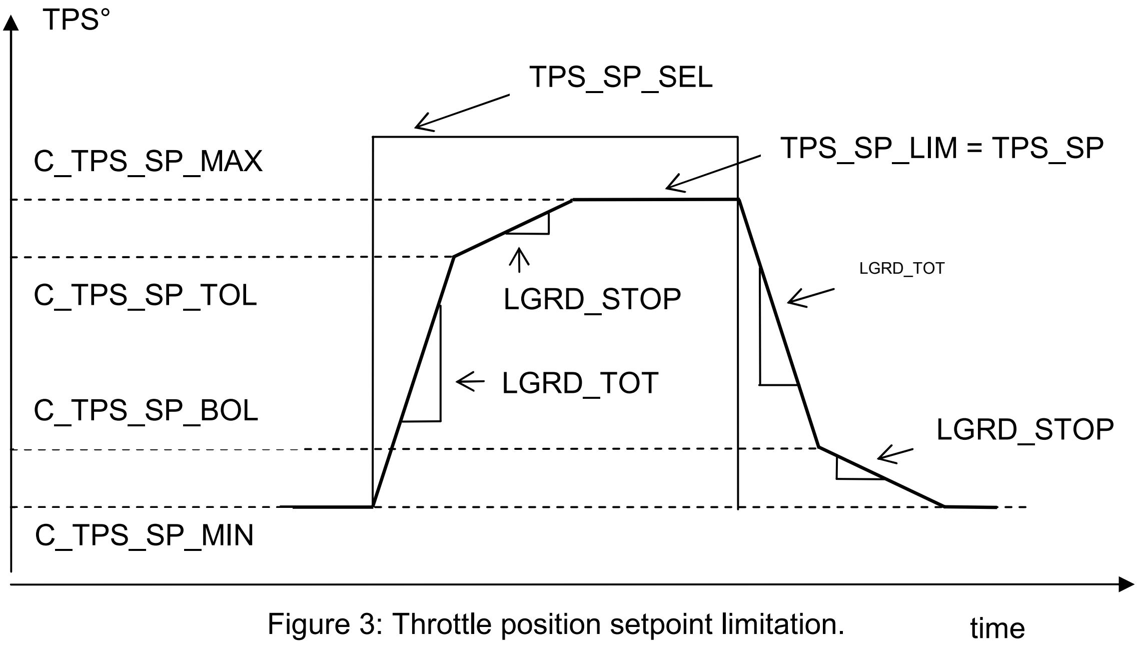


Figure 3: Throttle position setpoint limitation.

61.14.2.1 Limitation using absolute values

The throttle position setpoint may not fall below the value C_TPS_SP_MIN(_TPS_AD_REQ) or exceed C_TPS_SP_MAX(_TPS_AD_REQ).

Formula section:

```
if      ( TPS_SP_SEL  >  C_TPS_SP_MAX(_TPS_AD_REQ) )
then    TPS_SP_LIM_1 =  C_TPS_SP_MAX(_TPS_AD_REQ)
else    if      TPS_SP_SEL <  C_TPS_SP_MIN(_TPS_AD_REQ)
        then    TPS_SP_LIM_1 =  C_TPS_SP_MIN(_TPS_AD_REQ)
        else    TPS_SP_LIM_1 =  TPS_SP_SEL
        endif
endif
```

61.14.2.2 Gradient limitation over whole operating range

The gradient of the throttle position setpoint is limited throughout the whole operating range. A maximum change of C_TPS_SP_LGRD_TOT is permitted.

Formula section:

```
if      TPS_SP_LIM_1 > TPS_SP_LIM$_{i-1}$
then    TPS_SP_LIM_2 =
                MIN(TPS_SP_LIM_1, TPS_SP_LIM$_{i-1}$ + C_TPS_SP_LGRD_TOT)
else    TPS_SP_LIM_2 =
                MAX(TPS_SP_LIM_1, TPS_SP_LIM$_{i-1}$ - C_TPS_SP_LGRD_TOT)
endif
```

61.14.2.3 Gradient limitation near upper and lower position

The throttle position gradient is additionally limited as soon as the throttle position is near the upper or lower stop position: If TPS_SP_LIM_2 is close to the stop positions, for one step the setpoint TPS_SP_LIM is set to the value C_TPS_SP_TOL(_TPS_AD_REQ) or C_TPS_SP_BOL(_TPS_AD_REQ), respectively. When TPS_SP_LIM_2 is further moving towards the stops a gradient C_TPS_SP_LGRD_STOP < C_TPS_SP_LGRD_TOT is permitted. Thus a mechanical collision of blade and stops should be avoided (at least when TPS_AV is close toTPS_SP).

An additional protection mechanism near the mechanical stops is taking into account also the actual position TPS_AV of the throttle blade. If the difference between the desired TPS_SP and the actual TPS_AV is too high, the ETC controller will react aggressively. This reaction could lead to banging the throttle against the mechanical stops. To avoid such a behaviour the following algorithm is implemented: If TPS_AV is too far away from TPS_SP_LIM_2 (near the lower stop: TPS_AV > C_TPS_AV_BOL > C_TPS_SP_BOL whereas TPS_SP_LIM_2 ≈ C_TPS_SP_BOL; near the upper stop: TPS_AV < C_TPS_AV_TOL < C_TPS_SP_TOL whereas TPS_SP_LIM_2 ≈ C_TPS_SP_TOL), then TPS_SP_LIM is kept on the value C_TPS_SP_BOL (or C_TPS_SP_TOL, respectively) until TPS_AV is arriving below C_TPS_AV_BOL (or above C_TPS_AV_TOL) such that the difference between desired and actual position is small enough to have no strong ETC controller reactions. For illustration of the algorithm see figure 4.

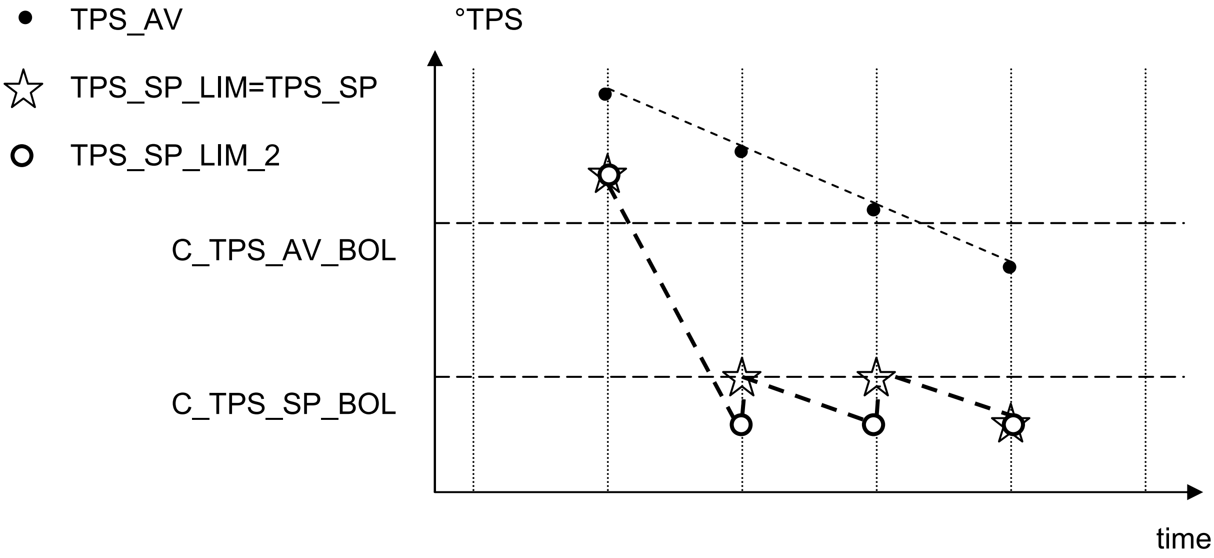


Figure 4: TPS_SP limitation mechanism at lower mechanical stop depending on TPS_AV.

Formula section:

```
if      TPS_SP_LIM_2  >  C_TPS_SP_TOL(_TPS_AD_REQ)          {above top limit}

then    if      ( TPS_SP_LIM$_{i-1}$  < C_TPS_SP_TOL(_TPS_AD_REQ)          or
                  TPS_AV  <  C_TPS_AV_TOL )

        then    TPS_SP_LIM = C_TPS_SP_TOL(_TPS_AD_REQ)

        else    TPS_SP_LIM = MIN(TPS_SP_LIM_2,
                                  TPS_SP_LIM$_{i-1}$+C_TPS_SP_LGRD_STOP)
        endif

else    if      TPS_SP_LIM_2 < C_TPS_SP_BOL(_TPS_AD_REQ)  {below bottom limit}

        then    if      ( TPS_SP_LIM$_{i-1}$ >C_TPS_SP_BOL(_TPS_AD_REQ) or
                        TPS_AV  >  C_TPS_AV_BOL )

                then    TPS_SP_LIM = C_TPS_SP_BOL(_TPS_AD_REQ)

                else    TPS_SP_LIM = MAX(TPS_SP_LIM_2,
                                          TPS_SP_LIM$_{i-1}$ -C_TPS_SP_LGRD_STOP)
                endif
```

```
        else    TPS_SP_LIM = TPS_SP_LIM_2

        endif

    endif
```

Remark: For correct working of the algorithm the following settings **must** be chosen:

- C_TPS_SP_BOL(_TPS_AD_REQ) < C_TPS_AV_BOL < TPS_LIH
- C_TPS_SP_TOL(_TPS_AD_REQ) > C_TPS_AV_TOL

61.14.2.4 Output of limited setpoint

Formula section:

TPS_SP = TPS_SP_LIM

Calibration data:

Name	Dim	Hex. Limit	Phys. Limit	Resol.	Unit
C_TPS_AV_BOL	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Lower limit (actual value) of the closing speed limitation of the throttle setpoint					
C_TPS_AV_TOL	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Upper limit (actual value) of the opening speed limitation of the throttle setpoint					
C_TPS_SP_MIN	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Lower limit of throttle position setpoint					
C_TPS_SP_MIN_TPS_AD_REQ	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Lower limit of throttle position setpoint without throttle adaptation					
C_TPS_SP_MAX	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Upper limit of throttle position setpoint					
C_TPS_SP_MAX_TPS_AD_REQ	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Upper limit of throttle position setpoint without throttle adaptation					
C_TPS_SP_TOL	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Upper limit of opening velocity limitation of throttle position setpoint					
C_TPS_SP_TOL_TPS_AD_REQ	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Upper limit of opening velocity limitation of throttle position setpoint without throttle adaptation					
C_TPS_SP_BOL	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Lower limit of opening velocity limitation of throttle position setpoint					
C_TPS_SP_BOL_TPS_AD_REQ	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Lower limit of opening velocity limitation of throttle position setpoint without throttle adaptation					
C_TPS_SP_LGRD_STOP	1	0...3FFFH	0...119.5	7.29E-03	[°TPS/10ms]
max. admissible gradient of throttle position setpoint for range of top and bottom stop					
C_TPS_SP_LGRD_TOT	1	0...3FFFH	0...119.5	7.29E-03	[°TPS/10ms]
max. admissible gradient of throttle position setpoint for whole operating range					

61.15 Throttle Position Setpoint Selection (Appl. Inc.)

61.15.1 Throttle position setpoint for engine stopped

Output data:

Name	Mode	Hex. limits	Phys. Limits	Resol.	Unit
TPS_SP_ES	V/O	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Throttle position setpoint for engine stopped					

Input data:

LV_ES	TCO		

FUNCTION DESCRIPTION:

General information:

For engine stopped TPS_SP_ES is generated from a TCO dependent map. The TPS_SP_ES value must be chosen with respect to the expected TPS_SP_ST. A sensible choice diminishes the transient time when the TPS_AV is approaching TPS_SP_ST.

Application conditions:

Initialisation: -

Recurrence: 10 ms

Activation: in every engine state

Formula section:

```
If    LV_ES = 1

      then    TPS_SP_ES  =  IP_TPS_SP_ES

endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TPS_SP_ES	8	0...3FFFH	0...119.5	7.29E-03	[°TPS]
LDPM_TCO_1_THRO	8	0...FEH	-48...142.5	0.75	[°C]
Throttle position setpoint for engine stopped					

61.15.2 Throttle position setpoint by external device

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TPS_SP_EXT	V/O	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Throttle position setpoint of external device					
LV_TPS_SP_EXT	V/O	0...1H	0...1	1	[-]
Bit active for external input of throttle position setpoint					

Input data:

LV_ES	LV_IGK	PV_AV	TPS_SP_EXT_ADJ
LV_TPS_SP_EXT_ADJ	C_TPS_SP_MAX		

Description:

For different tasks during calibration phase and for service procedures at the workshops, it is necessary to implement an external adjustment at throttle position.

Throttle position can be influenced via serial tester (KWP) or manual in the following way:

C_TPS_SP_EXT_ADJ_ENA = 0 : no manual adjustment

C_TPS_SP_EXT_ADJ_ENA = 1 : manual setpoint via application system

C_TPS_SP_EXT_ADJ_ENA = 2 : manual setpoint via PV_AV

C_TPS_SP_EXT_ADJ_ENA = 3 : manual setpoint via PV_AV and application system

Application conditions:

Initialisation: 0

Recurrence: 10ms

Activation: LV_IGK = 1

Formula section:

```
If(1)     LV_TPS_SP_EXT_ADJ = 1
Then(1)   LV_TPS_SP_EXT = 1
          TPS_SP_EXT = (TPS_SP_EXT_ADJ / 100%) * C_TPS_SP_MAX
                                              //to convert from % into °TPS
Else(1)   If(2)   C_TPS_SP_EXT_ADJ_ENA = 1     and
                  LV_ES = 1
          Then(2)     TPS_SP_EXT = C_TPS_SP_AS_MAN
                      LV_TPS_SP_EXT = 1
          Else(2)     If(3)   C_TPS_SP_EXT_ADJ_ENA = 2          and
                              PV_AV > C_PV_AV_EXT_ADJ_MIN       and
                              LV_ES = 1
```

```
                then(3) TPS_SP_EXT = IP_TPS_SP_EXT_ADJ
                        LV_TPS_SP_EXT = 1
                else(3)      If(4)  C_TPS_SP_EXT_ADJ_ENA = 3     and
                             PV_AV > C_PV_AV_EXT_ADJ_MIN
                        then(4) TPS_SP_EXT = C_TPS_SP_AS_MAN +
                                                      IP_TPS_SP_EXT_ADJ
                                LV_TPS_SP_EXT = 1
                        else(4) LV_TPS_SP_EXT = 0
                                TPS_SP_EXT = 0
                        endif(4)
                endif(3)
        endif(2)
endif(1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TPS_SP_AS_MAN	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Manual throttle setpoint					
IP_TPS_SP_EXT_ADJ	6	0...3FFFH	0...119.5	7.29E-03	[°TPS]
LDP_PV_AV__TPS_SP_EXT	6	0...FFH	0...99.60937	0.390625	[%]
External adjustment setpoint					
C_PV_AV_EXT_ADJ_MIN	1	0...FFH	0...99.60937	0.390625	[%]
Min. threshold for external ajustment about PV_AV					
C_TPS_SP_EXT_ADJ_ENA	1	0...3H	0...3	1	[-]
Variable enabled for throttle position setpoint manual or about PV_AV					

61.15.3 Throttle position setpoint during post operating phase

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
TPS_SP_PWL	V/O	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Throttle position setpoint during post operating phase					
LV_TPS_PWL	V/O	0...1H	0...1	1	[-]
Bit for throttle position setpoint in post operating phase active					
CTR_T_TPS_PWL	O	0...FFH	0...255	1	[s]
Timer after engine stop					

Input data:

LV_ES	LV_IGK	N_32	LV_MTC_CUR_OFF

General information:

To avoid that for ‚Power Down‘ the currentless throttle is set to the limp home position, for a tuneable time period a throttle position setpoint during the post operating phase is available.

A time counter is started on detection of ignition key OFF and N < 32 rpm (engine stopped). If the counter reaches its maximum C_T_MTC_ISA_OFF, the throttle is actuated to the setpoint C_TPS_SP_PWL_UP by the current TPS_SP and then de-energized. Thus, it is ensured that the throttle are closed until a defined time after ignition key OFF and engine stopped.

If the throttle is already deenergized when ignition key OFF is detected, it is not re-energized.

Formula section:

```
If (1)      LV_IGK = 0                                   and
            LV_MTC_CUR_OFF = 0; at LV_IGK = 1 to 0
Then(1)     LV_TPS_PWL =1
    If(2)       LV_ES = 0
    Then(2)         TPS_SP_PWL = IP_TPS_SP_PWL
    Else (2)    Timer started          (LV_ES = 1 to 0)
                if (3)      (CTR_T_TPS_PWL < C_T_MTC_ISA_OFF)
                Then(3)         TPS_SP_PWL = IP_TPS_SP_PWL
                Else (3)
                    Begin   $TPS\_SP\_PWL_n = TPS\_SP\_PWL_{n-1} + C\_TPS\_SP\_PWL\_LGRD$
                    Until   $TPS\_SP\_PWL \geq C\_TPS\_SP\_PWL\_UP$
                    Then    LV_TPS_PWL =0
Else(1)     LV_TPS_PWL =0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
C_T_MTC_ISA_OFF	1	0...FFH	0...255	1	[s]
Active time period for throttle position setpoint during post operating phase					
IP_TPS_SP_PWL	4	0...3FFFH	0...119.5	7.29E-03	[°TPS]
LDP_N_32_TPS_SP_PWL	4	0...FFH	0...8160	32	[rpm]
Throttle position setpoint during post operating phase					
C_TPS_SP_PWL_UP	1	0...3FFFH	0...119.5	7.29E-03	[°TPS]
Threshold opening throttele after power latch					
C_TPS_SP_PWL_LGRD	1	0...3FFFH	0...119.5	7.29E-03	[°TPS/10ms]
Gradient limitation TPS					

61.15.4 Throttle position setpoint and conditions for ETC switched off

Output data:

Name	Mode	Hex. Limits	Phys. limits	Resol.	Unit
LV_TPS_SP_LIH	V/O	0...1H	0...1	1	[-]
Bit collecting the requests of all 3 monitoring levels to switch off MTC					
LV_MTC_CUR_OFF	V/O	0...1H	0...1	1	[-]
The logical variable indicate the state of the ETC power stage					

Input data:

LV_ERR_MU_MC	LV_ERR_TMP_MU_MC	LV_MTC_CUR_OFF_REQ	LV_ES
LV_IGK	LV_TPS_PWL	LV_OFF_MTC_MON	LV_ERR_ECU_CKS
LV_ERR_ECU_RAM	LV_TPS_AD_CUR_OFF	LV_TPS_AD_ACT	LV_TPS_SP_EXT
LV_MC_SOPC_INH_DI			

Application conditions:

Initialisation: 1

Recurrence: 5 ms

Activation: in every engine state

Formula section:

```
If(1) LV_ES = 1                              and   //Engine stopped
      LV_IGK = 1                             and   //Key ON
      LV_TPS_AD_ACT = 0                      and   //No TPS adaptation active
      LV_TPS_SP_EXT = 0                      or    //No external adjustment TPS  active
      LV_TPS_AD_CUR_OFF = 1                  or    //Adaption-step with MTC off active
      LV_MTC_CUR_OFF_REQ = 1                 or    //Request from TPS/MTC diagnosis
      (LV_IGK = 0 and LV_TPS_PWL = 0)        or    //No activation in PWL
      LV_OFF_MTC_MON = 1                     or    //bit set by monitoring level 2
      LV_ERR_ECU_RAM = 1                     or    //ECU RAM error
      LV_ERR_ECU_CKS = 1                     or    //ECU CKS error
      LV_ERR_TMP_MU_MC =1                    or    //bits set by monitoring level 3
      LV_ERR_MU_MC= 1

then(1)     LV_MTC_CUR_OFF = 1
else(1)     LV_MTC_CUR_OFF = 0
endif(1)
```

LV_TPS_SP_LIH = LV_MTC_CUR_OFF

```
If(2)      LV_MC_SOPC_INH_DI = 1
then(2)    switch MTC HBR on, because MTC is disabled by MU
else(2)    switch MTC HBR according LV_MTC_CUR_OFF
endif(2)
```

61.16 Position Controller MTC

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
MTCPWM	V/O	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
duty cycle MTC					
LV_MTCPWM_CLOSE_ACT	V/O	0...1H	0...1	1	[-]
logical variable (is set, if closing duty cycle is active)					
MTCPWM_CTL	V	E0000H ... 1FFFFH	-400...400	0.00305	%
duty cycle of PID-control					
MTCPWM_P	V	8000...7FFFH	-400...399.98779	0.012207	[%]
duty cycle of P-share					
MTCPWM_I	V	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
duty cycle of I-share					
MTCPWM_D	V	8000...7FFFH	-400...399.98779	0.012207	[%]
duty cycle of D-share					
TPS_DIF	V/O	0...7FFEH	-119.5...119.5	7.2941e-3	[°TPS]
controller deviation					
TPS_DIF_GRD	V	0...7FFEH	-119.5...119.5	7.2941e-3	[°TPS]
gradient of controller deviation					
T_MTCPWM_CLOSE_ACT	V	0...3FEH	0...5110	5	[ms]
time counter for closing condition					
TPS_AV_LIH_DIF	V	0...7FFEH	-119.5...119.5	7.2941e-3	[°TPS]
difference between TPS_AV and TPS_LIH					
TPS_SP_LIH_DIF	V	0...7FFEH	-119.5...119.5	7.2941e-3	[°TPS]
difference between TPS_SP and TPS_LIH					
TPS_SP_CTL	V	C001...3FFFH	-119.5...119.5	7.2941e-3	[°TPS]
throttle position setpoint (+ or -)					
LV_TPS_SP_POS	V	0...1H	0...1	1	[-]
position detection of TPS-setpoint (0: below LIH; 1: above LIH)					
LV_TPS_CHG_POS	V	0...1H	0...1	1	[-]
detection of changed direction from LV_TPS_SP_POS					
MTCPWM_I_ADD	V	0...7FFFH	0...99.99694	3.0518e-3	[%]
additive friction compensation of I-share					
MTCPWM_ADD	V	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Tastverhältnis der Feed-Forward-Steuerung					
MTCPWM_VB_FAC	V	0...FFH	0...1.99218	0.0078125	[-]
battery voltage correction					

Input data:

TPS_AV	TPS_AV_CTL	TPS_LIH	TPS_SP
LV_MTC_CUR_OFF	VB	C_TPS_SP_LGRD_STOP	LV_IGK

FUNCTION DESCRIPTION:

General information:

The position controller is an adaptive PID-controller with a feed forward control.

Application conditions:

Initialisation: at deactiavtion **or** LV_IGK 0->1 **or** ECU reset

Recurrence: 5 ms

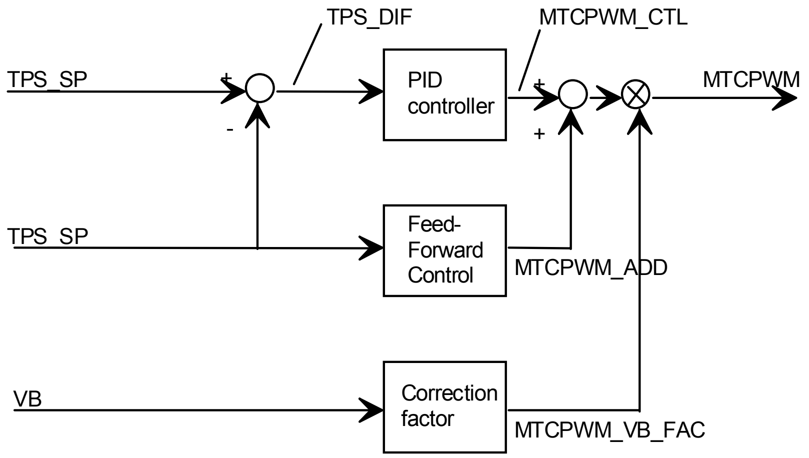
Activation: **If** LV_MTC_CUR_OFF = 0

Deactivation: **If** LV_MTC_CUR_OFF = 1

Then Initialization of variables:

[MTCPWM, MTCPWM_P, MTCPWM_I, MTCPWM_I_ADD
MTCPWM_D, MTCPWM_ADD, MTCPWM_CTL] with 0
TPS_SP_CTL = (S16) C_TPS_SP_STOP_HYS
T_MTCPWM_CLOSE_ACT = 0
LV_MTCPWM_CLOSE_ACT = 0
TPS_DIF = 0x3FFF
TPS_DIF_GRD = 0x3FFF

Signal flow diagram:



61.16.1 PID-Controller

Description:

The PID-controller is nonlinear. It consists of three parts: P-part, I-part and D-part P- and I-part depend on deviation. The D-part depends on deviation gradient.

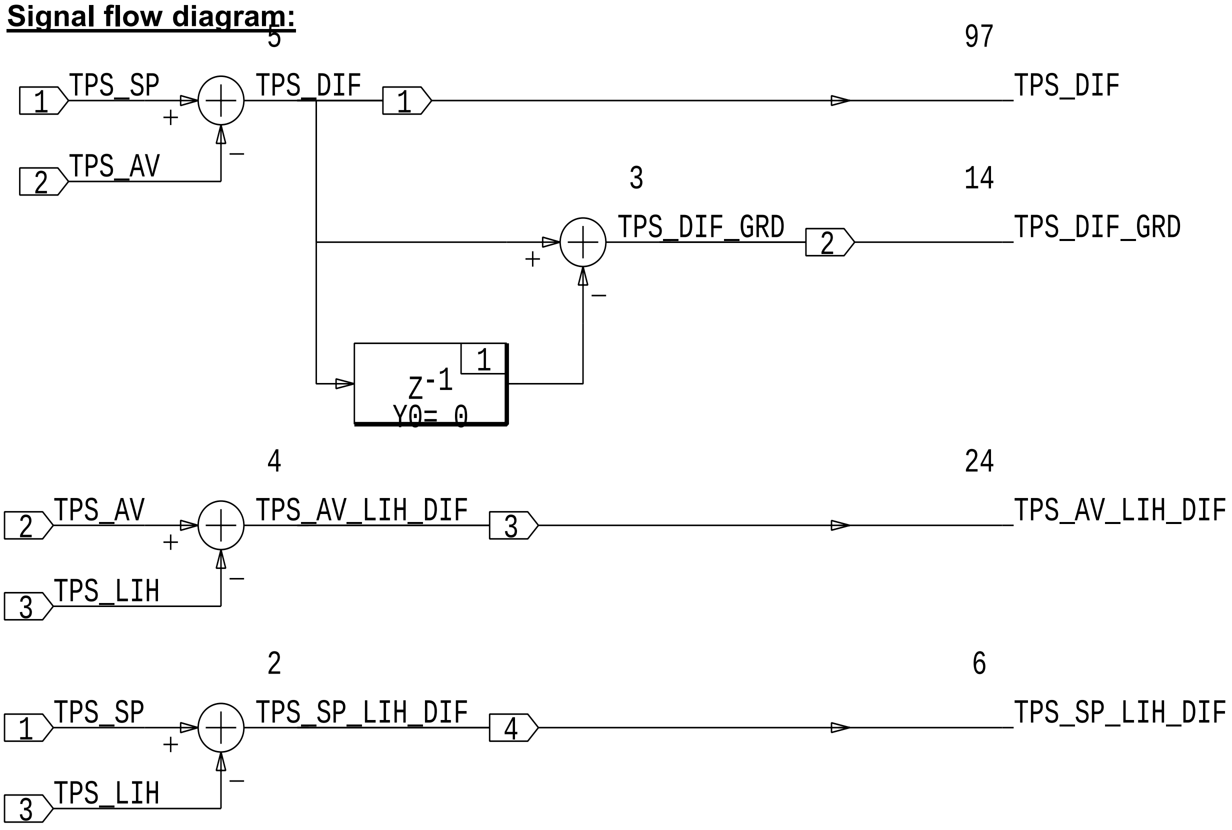
Near the mechanical bottom limit the controller deviation is calculated with an offset on the actual throttle position that is equal to the setpoint gradient limitation, even if setpoint value is at its lower limit. So the throttle is moving to the stop with a defined velocity. To calculate the controller deviation a value of the throttle position that can also be negativ is formed. The time where duty cycle is above the closing value is measured. If meanwhile the setpoint leaves the lower bottom limit, the time counter is set back to zero.

16-NOV-98

Discrete Procedure SuperBlock Deviation | Procedure Class Standard | Inputs 3 | Outputs 4

61.16.1.1 Calculation of Deviation

Signal flow diagram:



Formula section:

controller deviation:

```
If      TPS_SP > C_TPS_SP_STOP_HYS
then    TPS_DIF   =   TPS_SP - TPS_AV
        TPS_SP_CTL = C_TPS_SP_STOP_HYS
else    TPS_SP_CTL = TPS_SP_CTL - C_TPS_SP_LGRD_STOP
        TPS_DIF   =   MAX(TPS_SP_CTL;C_TPS_SP_CTL_MAX) - TPS_AV_CTL
endif
```

Remark: C_TPS_SP_STOP_HYS should be calibrated nearly to 0.

gradient of the controller deviation:

$$TPS\_DIF\_GRD = TPS\_DIF_n - TPS\_DIF_{n-1}$$

difference between the actual value and the limp home position:

TPS_AV_LIH_DIF = TPS_AV - TPS_LIH

TPS_SP_LIH_DIF = TPS_SP - TPS_LIH

61.16.1.2 Duty Cycle Supervision / Set Timer

Description:

Below the threshold C_TPS_SP_STOP_MAX is supervised if the throttle is at bottom mechanical limit, i.e. if he duty cycle has exceeded the threshold C_MTCPWM_CLOSE * MTCPWM_VB_FAC. In that case a time counter is started.

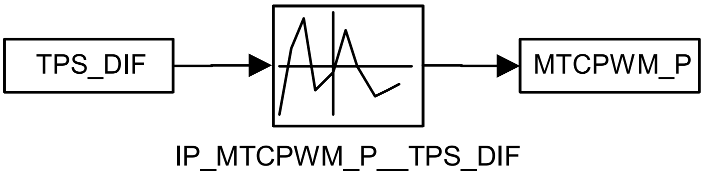
Formula section:

```
If (1)          TPS_SP ≤ C_TPS_SP_STOP_MAX
then (1)
    if (2)          MTCPWM <= C_MTCPWM_CLOSE * MTCPWM_VB_FAC +
                                        C_MTCPWM_HYS
    then (2)        T_MTCPWM_CLOSE_ACT = T_MTCPWM_CLOSE_ACT + 5ms
    else (2)        T_MTCPWM_CLOSE_ACT = 0
    endif (2)
else (1)        T_MTCPWM_CLOSE_ACT = 0
endif (1)
```

Remark: It´s not allowed to calibrate C_TPS_SP_STOP_MAX smaller than C_TPS_SP_STOP_HYS!

61.16.1.3 P-share:

Signal flow diagram:



Formula section:

MTCPWM_P = IP_MTCPWM_P__TPS_DIF

61.16.1.4 I- share:

Formula section:

detection of the TPS_SP direction :

```
if (1)          $TPS\_SP\_LIH\_DIF < - C\_TPS\_SP\_LIH\_DIF\_HYS$
then (1)        $LV\_TPS\_SP\_POS = 0$
else (1)
    if (2)      $TPS\_SP\_LIH\_DIF > C\_TPS\_SP\_LIH\_DIF\_HYS$
    then (2)    $LV\_TPS\_SP\_POS = 1$
    endif (2)
endif (1)


if (1)          $LV\_TPS\_SP\_POS_n \neq LV\_TPS\_SP\_POS_{n-1}$
then (1)        $LV\_TPS\_CHG\_POS = 1$
        if (2)      $LV\_TPS\_SP\_POS_n < LV\_TPS\_SP\_POS_{n-1}$
            and     $MTCPWM\_I \geq -C\_MTCPWM\_I\_ADD$
        then (2)    $MTCPWM\_I\_ADD = -C\_MTCPWM\_I\_ADD$
        else (2)
            if (3)      $LV\_TPS\_SP\_POS_n > LV\_TPS\_SP\_POS_{n-1}$
                and     $MTCPWM\_I \leq C\_MTCPWM\_I\_ADD$
            then (3)    $MTCPWM\_I\_ADD = C\_MTCPWM\_I\_ADD$
            endif (3)
        endif (2)
else (1)        $MTCPWM\_I\_ADD_n = MTCPWM\_I_{n-1}$
                $LV\_TPS\_CHG\_POS = 0$
endif (1)
MTCPWM_I   =    $MTCPWM\_I\_ADD + IP\_MTCPWM\_I\_\_TPS\_DIF$
                $* IP\_MTCPWM\_I\_FAC\_\_TPS\_AV\_LIH\_DIF$
```

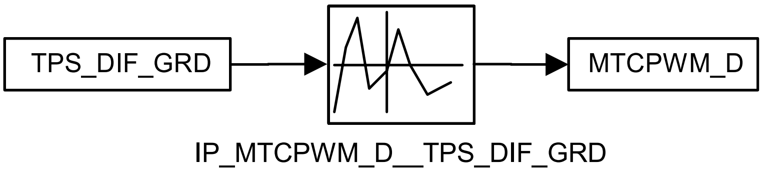
Limitation of I-share:

I-share is limited to a range C_MTCPWM_I_MAX_NEG, C_MTCPWM_I_MAX_POS.

```
if (1)          MTCPWM_I > C_MTCPWM_I_MAX_POS
then (1)        MTCPWM_I = C_MTCPWM_I_MAX_POS
else (1)
    if (2)      MTCPWM_I < C_MTCPWM_I_MAX_NEG
    then (2)    MTCPWM_I = C_MTCPWM_I_MAX_NEG
    else (2)    MTCPWM_I = MTCPWM_I
    endif (2)
endif (1)
```

61.16.1.5 D-share:

Signal flow diagram:



Formula section:

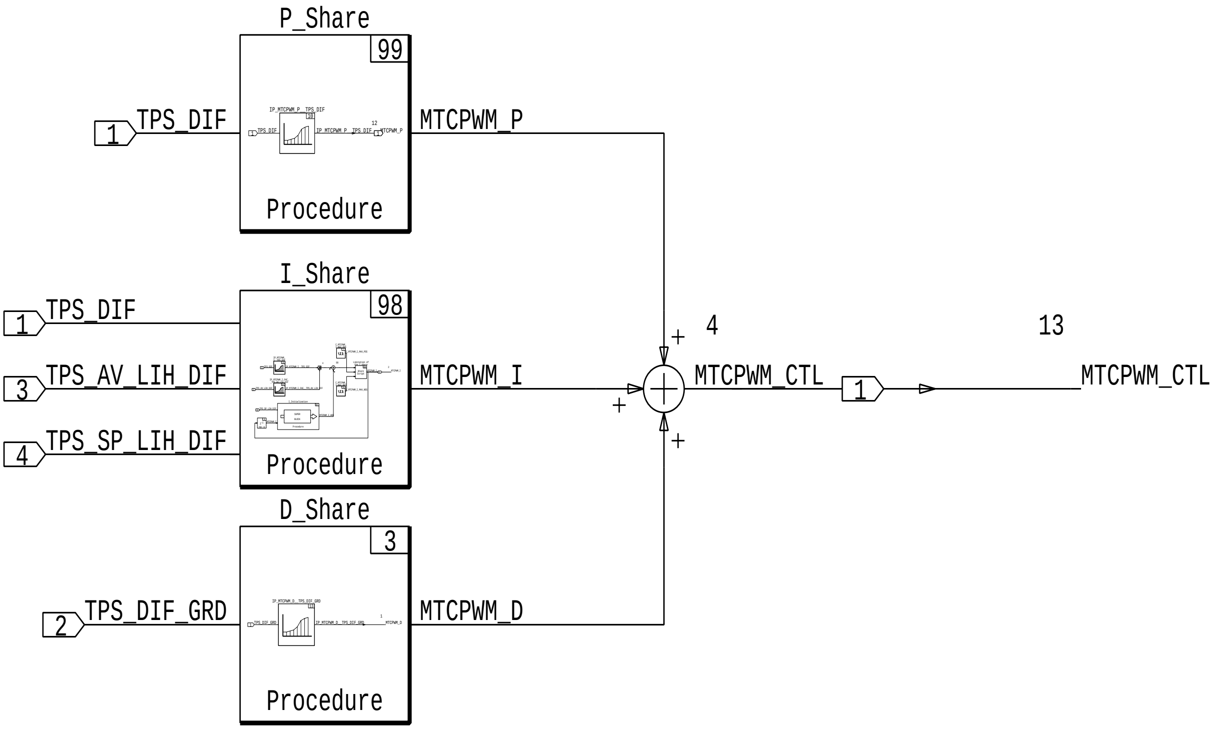
MTCPWM_D = IP_MTCPWM_D__TPS_DIF_GRD

61.16.1.6 Calculation of Controller Value:

Discrete Procedure SuperBlock	Procedure Class	Inputs	Outputs
PID_controller	Standard	4	1

16-NOV-98

Signal flow diagram:



Formula section:

MTCPWM_CTL = MTCPWM_P + MTCPWM_I + MTCPWM_D

61.16.1.7 Initialization of Position Controller after “mechanical limitation stop” respectively closing mode

If the throttle is opening beginning at bottom mechanical limit and exceeds the calibratable threshold C_TPS_SP_STOP_HYS the I-share is initialized with C_MTCPWM_I_INI, TPS_DIF_GRD is set to actual TPS_DIF and T_MTCPWM_CLOSE_ACT is set to zero.

```
if          TPS_SP > C_TPS_SP_STOP_HYS
      and   T_MTCPWM_CLOSE_ACT > 0
then        MTCPWM_I = C_MTCPWM_I_INI
            TPS_DIF_GRD = TPS_DIF
            T_MTCPWM_CLOSE_ACT = 0
endif
```

Calibration data:

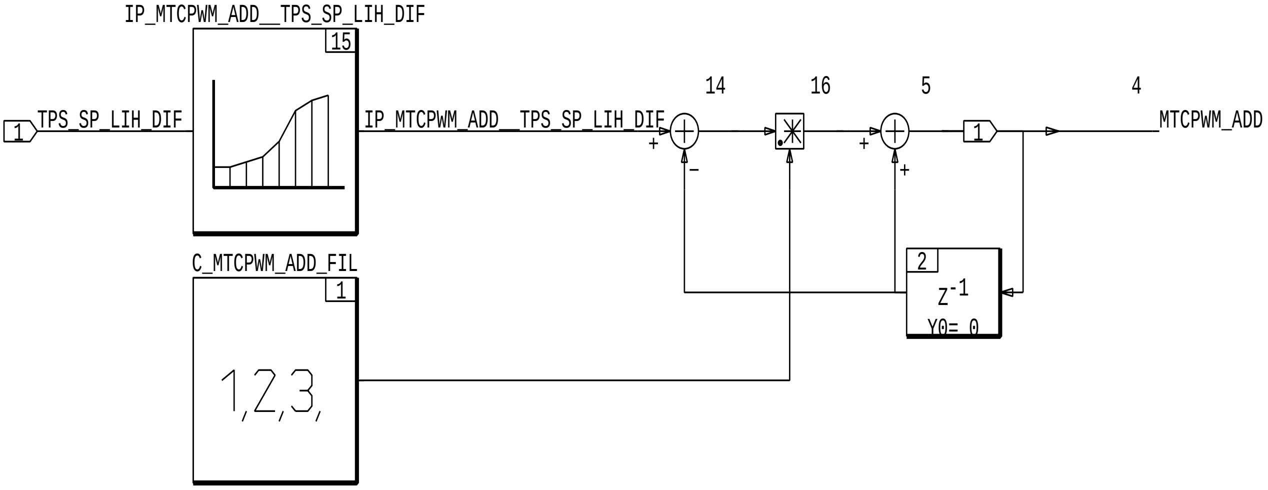
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_MTCPWM_P__TPS_DIF	8	0...FFFFH	-400...399.98779	0.012207	[%]
LDP_TPS_DIF__MTCPWM_P	8	0...7FFEH	-119.5...119.5	7.2941e-3	[°TPS]
position controller P-share					
IP_MTCPWM_I__TPS_DIF	8	0...FFFFH	-100...99.99694	3.0518e-3	[%]
LDP_TPS_DIF__MTCPWM_I	8	0...7FFEH	-119.5...119.5	7.2941e-3	[°TPS]
position controller I-share					
IP_MTCPWM_I_FAC__TPS_AV_LIH_DIF	1x8	0...FFH	0...7.97	0.03125	-
LDP_TPS_AV_LIH_DIF__MTCPWM_I	8	0...7FFEH	-119.5...119.5	0.0073	°TPS
LIH-correction of I-share					
IP_MTCPWM_D__TPS_DIF_GRD	8	0...FFFFH	-400...399.98779	0.012207	[%]
LDP_TPS_DIF_GRD__MTCPWM_D	8	0...7FFEH	-119.5...119.5	7.2941e-3	[°TPS]
position controller D-share					
C_MTCPWM_I_MAX_POS	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
limitation of I-share upwards					
C_MTCPWM_I_MAX_NEG	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
limitation of I-share downwards					
C_TPS_SP_STOP_HYS	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
setpoint hysteresis for closing throttle					
C_TPS_SP_STOP_MAX	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
threshold for testing duty cycle on value					
C_MTCPWM_I_INI	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
initialization of MTCPWM_I					
C_TPS_SP_CTL_MAX	1	FFFFC001...0H	-119.5...0	0.0073	°TPS
limitation of TPS_SP_CTL					
C_MTCPWM_HYS	1	0...7FFFH	0...99.99694	3.0518e-3	[%]
hysteresis to prevent MTCPWM-switching according to VB-changes					
C_TPS_SP_LIH_DIF_HYS	1	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
hysteresis at LIH-position for I-share initialization					
C_MTCPWM_I_ADD	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
friction compensation of I-share					

61.16.2 Feed - Forward Control of the MTC

Description:

The friction torque of the bearing and of the spring set a very strong non-linearization of the control course. To compensate this and to avoid instability a feed-forward-control is used dependent on the throttle setpoint deviation to LIH-position (TPS_SP_LIH_DIF). For to take into consideration the dynamic behavior a PT1-filter with the time constant C_MTCPWM_ADD_FIL is used.

Signal flow diagram:



Formula section:

$$\text{MTCPWM\_ADD}_n = \text{MTCPWM\_ADD}_{n-1} + (\text{IP\_MTCPWM\_ADD\_\_TPS\_SP\_LIH\_DIF}_n - \text{MTCPWM\_ADD}_{n-1}) * \text{C\_MTCPWM\_ADD\_FIL}$$

Calibration data:

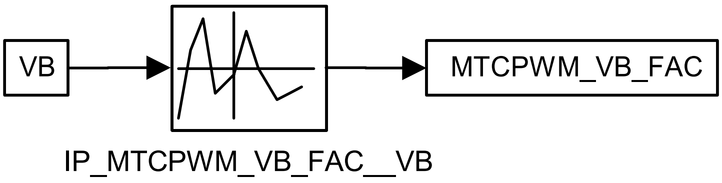
Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_MTCPWM_ADD__TPS_SP_LIH_DIF	8	0...FFFFH	-100...99.99694	3.0518e-3	[%]
LDPM_TPS_SP_LIH_DIF__MTCPWM_ADD	8	0...7FFEH	-119.5...119.5	7.2941e-3	[°TPS]
feed-forward-control at positive movving direction					
C_MTCPWM_ADD_FIL	1	0...FFH	0...0.99609	3.9063e-3	[-]
PT1-filtering constant					

61.16.3 Battery Voltage Correction

Description:

The control duty cycle will be corrected through the battery voltage VB.

Signal flow diagram:



Formula section:

MTCPWM_VB_FAC = IP_MTCPWM_VB_FAC__VB

Applikationsdaten:

Name	Dim	Hex. Limit	Phys. Limit	Auflös.	Einh.
IP_MTCPWM_VB_FAC__VB	8	0...FFH	0...1.99218	0.0078125	[-]
LDP_VB__MTCPWM_VB_FAC	8	0...FFH	0...25.89843	0.1015625	[V]
battery voltage correction					

61.16.4 Calculation of MTCPWM

Description:

The duty cycle that controls the main throttle is the sum of a feed-forward-controller value and the PID-controller value. This sum is corrected by a factor depending on battery voltage. Is the throttle position for sure at the lower mechanical stop, the duty cycle is limited to the allowed value. A rise of the duty cycle over the maximum value marks that the throttle position has reached the mechanical stop. The maximum duty cycle is allowed only for a defined time. Then the duty cycle is switched back to the closing level.

Formula section:

```
MTCPWM   =   (MTCPWM_CTL+ MTCPWM_ADD) * MTCPWM_VB_FAC
LV_MTCPWM_CLOSE_ACT = 0
    if (1)          T_MTCPWM_CLOSE_ACT > 0
    then (1)        MTCPWM =   MAX{(C_MTCPWM_CLOSE_MAX * MTCPWM_VB_FAC);
                                   MTCPWM}
            if (2)  T_MTCPWM_CLOSE_ACT > C_ T_MTCPWM_CLOSE_ACT_MAX
            then (2)        MTCPWM = C_MTCPWM_CLOSE * MTCPWM_VB_FAC
                            LV_MTCPWM_CLOSE_ACT = 1
            endif (2)
    endif (1)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MTCPWM_CLOSE_ACT_MAX	1	0...3E8H	0...5	0.005	[s]
timer-threshold for switching to closing duty cycle					
C_MTCPWM_CLOSE	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
closing duty cycle					
C_MTCPWM_CLOSE_MAX	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
duty cycle limitation at bottom mechanical limit					

61.17 Customer adaptation module THRO: Throttle

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TPS_SUB_DIAG	V/O	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Substitute TPS_AV value calculated out of IMM					
TPS_SP_MDL	V/O	0...3FFFH	0...119.5	7.2941e-3	[°TPS]
Desired throttle angle based on customer intake model					
Wdk_ist	V/O	8000...7FFFH	-800...799.97558	0.0244141	[%]
Actual throttle angle					
B_dkheiz_ende	V/O	0...1H	0...1	1	[-]
Bedingung Heizturnus Drosselklappe beendet					
B_dkerr_2pot	V/O	0...1H	0...1	1	[-]
Bedingung beide DK-Potis defekt					
B_dkratio_h	V/O	0...1H	0...1	1	[-]
Homogener Betrieb angefordert					

Input data:

Wdk_soll	C_TPS_SP_MAX	TPS_AV	Wdk_diag
STATE_TPS_DIAG	LV_MTC_HEAT_READY		

FUNCTION DESCRIPTION:

General information:

Adaption to BMW environment.

Remark: The really possible resolution and limits of the output are different to the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit power latch phase

0

except:

Wdk_ist = TPS_AV after first valid sampling
B_dkheiz_ende = 1

Recurrence: 10ms

Except:	5ms:	B_dkratio_h
	100ms:	B_dkheiz_ende
	segment:	Wdk_ist

Note: Wdk_ist has to be updated directly after TPS_AV has been calculated. Wdk_ist have to be hold on last calculated value during power latch phase.

general specification

Activation: at every engine stat

Deactivation: at power latch phase

Values at deactivation: B_dkratio_h: 0
B_dkerr_2pot, TPS_SUB_DIAG, TPS_SP_MDL, Wdk_ist,
B_dkheiz_ende : unchanged

Formula section:

Remark: all formulas are valid in a **physical** meaning

TPS_SP_MDL = (Wdk_soll/100%) • C_TPS_SP_MAX

Wdk_ist = (TPS_AV / C_TPS_SP_MAX) • 100%

```
If      STATE_TPS_DIAG = TPS_DBL_ERROR        //8H
Then    B_dkerr_2pot = 1
Else    B_dkerr_2pot = 0
Endif
If      STATE_TPS_DIAG = 1 or 2 or 4
Then    B_dkratio_h = 1
Else    B_dkratio_h = 0
Endif
```

TPS_SUB_DIAG = (Wdk_diag/100%) • C_TPS_SP_MAX

B_dkheiz_ende = LV_MTC_HEAT_READY

62 Torque determination and realization

Chapter Torque determination and realization	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 9189 of 9643

Table of Contents

Index

62.1 AGGR TQDR adaptation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_MAP_PUC_LIM_ACT_LSL_GAIN_AD	V/O	0...1H	0...1	1	-
Logical variable is set if MAP limitation is activated					
TQI_AV	V / O	8000 ... 7FFF H	-1024 ... 1023.97	0.03125	Nm
Actual indicated engine torque					

Input data:

TQI_AV_HOM	TQI_AV_S	LV_S_ACT	

FUNCTION DESCRIPTION:

Adaptation to aggregate environment.

Application conditions:

Initialisation: LV_MAP_PUC_LIM_ACT_LSL_GAIN_AD = 0 (never changed after initialisation)

Recurrence: 10 ms

Activation: -

Deactivation: -

Formula section:

If LV_S_ACT = 0

Then TQI_AV = TQI_AV_HOM

Else TQI_AV = TQI_AV_S

62.2 IGNITION ANGLE EFFICIENCY

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_EFF_IGA	O/V	0...FFH	0...11.22	0.044	-
Factor for basic ignition efficiency correction depening on engine operation mode					
EFF_IGA_MIN	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Ignition efficiency minimum ignition angle					
EFF_IGA_BAS_COR_CBK[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Ignition efficiency basic ignition angle (Bank selective)					
EFF_IGA_BAS_COR_KNK_FIL	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Ignition efficiency referenced to IGA_BAS_COR_KNK_FIL					
EFF_IGA_AV	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Ignition efficiency actual ignition angle bank selective					
EFF_IGA_AV_CBK[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Ignition efficiency actual ignition angle bank selective					
EFF_IGA_AV_CAN	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Ignition efficiency actual ignition angle for CAN					
EFF_IGA_MIN_TEG	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
possible ignition efficiency minimum according to the exhaust gas temperature					
EFF_IGA_SEG_AV	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Ignition efficiency actual ignition angle segment synchronous					
IGA_DIF_BAS_KNK_FIL	O/V	0...FFH	0...-95.625	-0.375	°CRK
Difference of IGA_BAS_COR_KNK_FIL to reference ignition angle					
FAC_EFF_IGA_CH_COR	O/V	0...FFH	0...11.22	0.044	-
Factor for basic ignition efficiency correction depening on engine operation mode					
EFF_IGA_BAS_COR	O/V	0...FFFFH	0...1.999969	3.05176E-5	-
Ignition efficiency basic ignition angle					
IGA_BAS_COR_KNK_ADD	V	0...FFH	-35.625...60	0.375	°CRK
Basic corrected ignition angle knock mean value IGA_MV_ADJ_KNK included					
IGA_DIF_BAS_CBK[NC_CBK_EX_NR]	V	0...FFH	0...-95.625	-0.375	°CRK
Difference of basic ignition angle to reference ignition angle bank selective					
IGA_DIF_AV_H_RNG	V	0...B40H	0...-180	-0.0625	°CRK
Difference of actual ignition angle to reference ignition angle					
IGA_DIF_SEG_AV_H_RNG	V	0...B40H	0...-180	-0.0625	°CRK
Difference of actual ignition angle to reference ignition angle segment synchronous					
IGA_DIF_AV_CAN_H_RNG	V	0...B40H	0...-180	-0.0625	°CRK
Difference of actual ignition angle to reference ignition angle for CAN					
IGA_DIF_AV_CBK_H_RNG[NC_CBK_EX_NR]	V	0...B40H	0...-180	-0.0625	°CRK
Difference of actual ignition angle to reference ignition angle bank selective					
IGA_MV_KNK_FIL	V	80...7FH	-48...47.625	0.375	°CRK
Filtered mean knock correction					
IGA_BAS_COR_KNK_FIL_CBK[NC_CBK_EX_NR]	V	0...FFH	-35.625...60	0.375	°CRK
Corrected basic ignition angle with mean knock correction bank selective					

Input data:

IGA_BAS_COR_CBK[NC_CBK_EX_NR]	IGA_MV_CBK_ADJ_KNK[NC_CBK_EX_NR]	IGA_REF_COR_CBK[NC_CBK_EX_NR]	IGA_AV_MV_H_RNG
IGA_AV_MV_CAN_H_RNG	IGA_AV_MV_CBK_H_RNG[NC_CBK_EX_NR]	IGA_DIF_MIN_H_RNG	IGA_DIF_MIN_TEG_H_RNG
N_32	FAC_MAF_REL	FAC_IGA_OPM_SEL	FAC_CAM_CH
LV_ST	FAC_EFF_IGA_CH	LV_IS	FAC_EFF_IGA_SECU
FAC_EFF_IGA_SCAV	LC_IGA_ST_ENA	NC_CBK_EX_NR	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_IGA_KNK	1	0...FFH	0...0.99609375	0.00390625	-
Correlation constant knock correction filter					
IP_EFF_IGA	16	0...4E20H	0...100	0.005	%
LDP_IGA_DIF_IP_EFF_IGA	16	0...B40H	0...-180	-0.0625	°CRK
Basic Ignition Angle Efficiency					
IP_FAC_EFF_IGA_CH_COLD_IS_OPM_1	8x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_8_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_5_TQDR	10	0...BB8H	0...300	0.1	%
Factor for catalyst heating cold engine during IS in normal mode					
IP_FAC_EFF_IGA_CH_COLD_IS_OPM_2	8x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_9_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_6_TQDR	10	0...BB8H	0...300	0.1	%
Factor for catalyst heating cold engine during IS in throttled mode					
IP_FAC_EFF_IGA_CH_COLD_OPM_1	12x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_13_TQDR	12	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_1_TQDR	10	0...BB8H	0...300	0.1	%
Factor for catalyst heating cold engine in normal mode					
IP_FAC_EFF_IGA_CH_COLD_OPM_2	12x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_16_TQDR	12	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_2_TQDR	10	0...BB8H	0...300	0.1	%
Factor for catalyst heating cold engine in throttled mode					
IP_FAC_EFF_IGA_CH_IS_OPM_1	8x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_8_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_5_TQDR	10	0...BB8H	0...300	0.1	%
Factor for catalyst heating warm engine during IS in normal mode					
IP_FAC_EFF_IGA_CH_IS_OPM_2	8x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_9_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_6_TQDR	10	0...BB8H	0...300	0.1	%
Factor for catalyst heating warm engine during IS in throttled mode					
IP_FAC_EFF_IGA_CH_OPM_1	12x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_2_TQDR	12	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_1_TQDR	10	0...BB8H	0...300	0.1	%
Factor for catalyst heating warm engine in normal mode					
IP_FAC_EFF_IGA_CH_OPM_2	12x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_11_TQDR	12	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_2_TQDR	10	0...BB8H	0...300	0.1	%
Factor for catalyst heating warm engine in throttled mode					
IP_FAC_EFF_IGA_COLD_IS_OPM_1	8x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_21_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_11_TQDR	10	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency depending on cold engine, IS, and engine operation mode 1					
IP_FAC_EFF_IGA_COLD_IS_OPM_2	8x8	0...FFH	0...11.22	0.044	-
LDPM_N_32_22_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_12_TQDR	8	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency depending on cold engine, IS, and engine operation mode 2					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_EFF_IGA_COLD_OPM_1	12x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_19_TQDR	12	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_9_TQDR	10	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency depending on cold engine and engine operation mode 1					
IP_FAC_EFF_IGA_COLD_OPM_2	12x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_20_TQDR	12	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_10_TQDR	10	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency depending on cold engine and engine operation mode 2					
IP_FAC_EFF_IGA_COLD_SCAV_OPM_1	8x8	0...FFH	0...11.22	0.044	-
LDPM_N_32_23_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_13_TQDR	8	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency for cold engine regarding scavenging operation (OPM 1)					
IP_FAC_EFF_IGA_COLD_SCAV_OPM_2	8x8	0...FFH	0...11.22	0.044	-
LDPM_N_32_24_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_14_TQDR	8	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency for cold engine regarding scavenging operation (OPM 2)					
IP_FAC_EFF_IGA_IS_OPM_1	8x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_17_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_7_TQDR	10	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency depending on engine in IS mode operation mode 1					
IP_FAC_EFF_IGA_IS_OPM_2	8x10	0...FFH	0...11.22	0.044	-
LDPM_N_32_18_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_8_TQDR	10	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency depending on engine in IS mode operation mode 2					
IP_FAC_EFF_IGA_OPM_1	20x16	0...FFH	0...11.22	0.044	-
LDPM_N_32_10_TQDR	20	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_3_TQDR	16	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency depending on engine operation mode 1					
IP_FAC_EFF_IGA_OPM_2	20x16	0...FFH	0...11.22	0.044	-
LDPM_N_32_12_TQDR	20	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_4_TQDR	16	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency depending on engine operation mode 2					
IP_FAC_EFF_IGA_SCAV_OPM_1	8x8	0...FFH	0...11.22	0.044	-
LDPM_N_32_23_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_13_TQDR	8	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency regarding scavenging operation (OPM 1)					
IP_FAC_EFF_IGA_SCAV_OPM_2	8x8	0...FFH	0...11.22	0.044	-
LDPM_N_32_24_TQDR	8	0...FFH	0...8.16E+3	32	rpm
LDPM_FAC_MAF_REL_14_TQDR	8	0...BB8H	0...300	0.1	%
Factor to correct the basic ignition efficiency regarding scavenging operation (OPM 2)					

62.2.1 Ignition angle efficiency

The ignition efficiency values EFF_IGA_AV (actual efficiency), EFF_IGA_BAS_COR (basic efficiency), EFF_IGA_BAS_COR_KNK_FIL (efficiency with mean knock correction) and EFF_IGA_MIN (minimum efficiency due to IGA_MIN) provide the link from an ignition angle to a torque efficiency. They are used in different torque modules to calculate the actual indicated engine torque, basic indicated engine torque values or the minimum indicated engine torque.

The efficiency calculation follows always the same strategy:

- calculation of all specific delta ignition angle IGA_DIF...
- calculation of efficiencies using engine mode depending function "EFF_IGA(IGA_DIF...)".

Application Condition

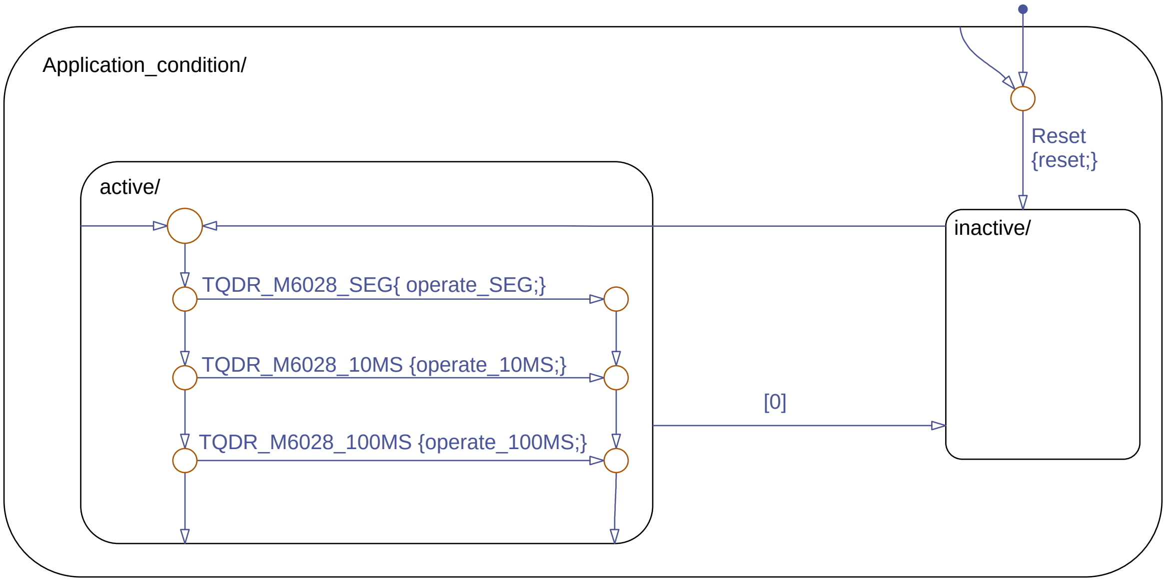


Figure 1 TQDR_M6028/ APP_CDN/ APPCND

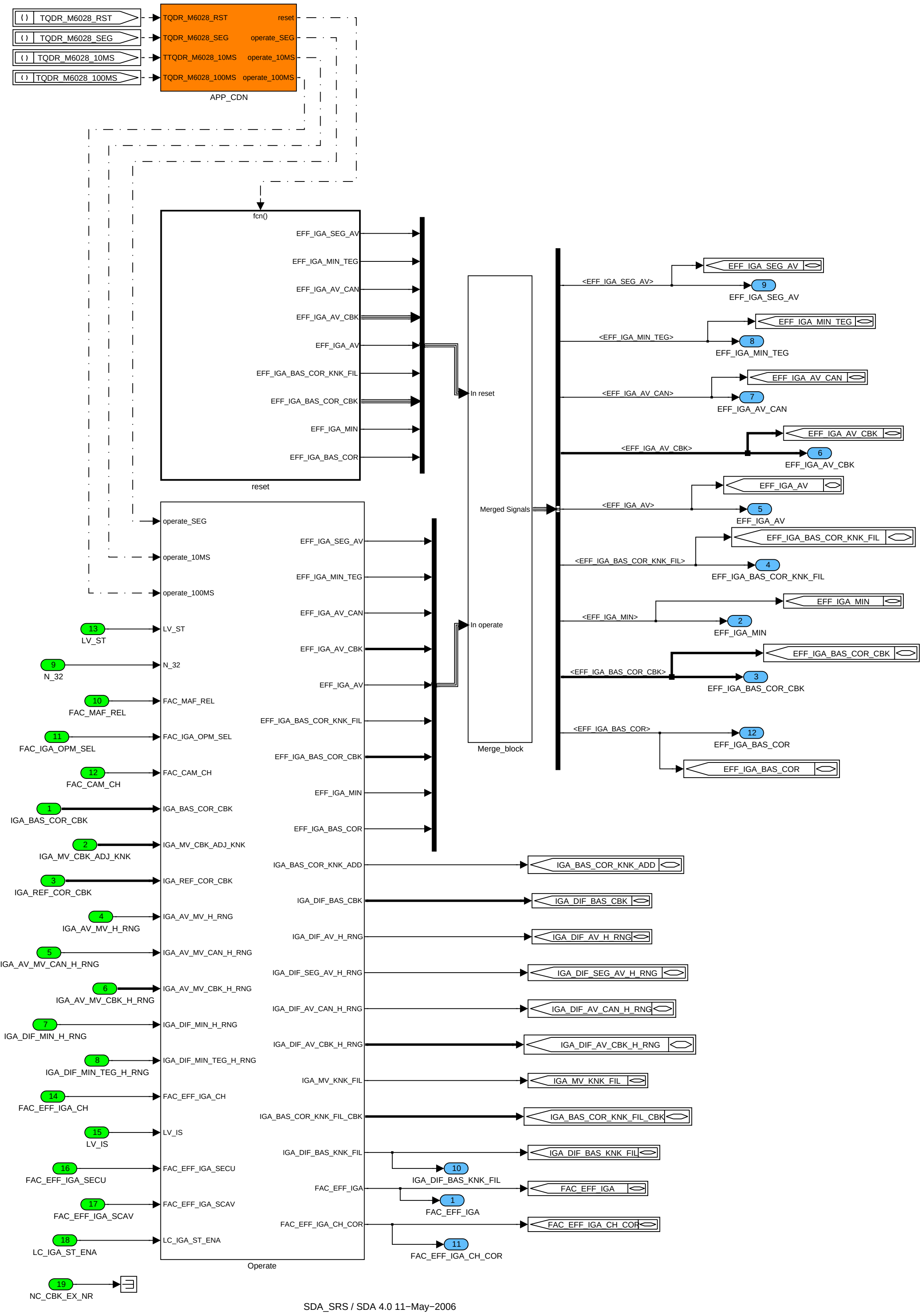


Figure 2 TQDR_M6028

62.2.1.1 RESET

Resetting efficiencies.

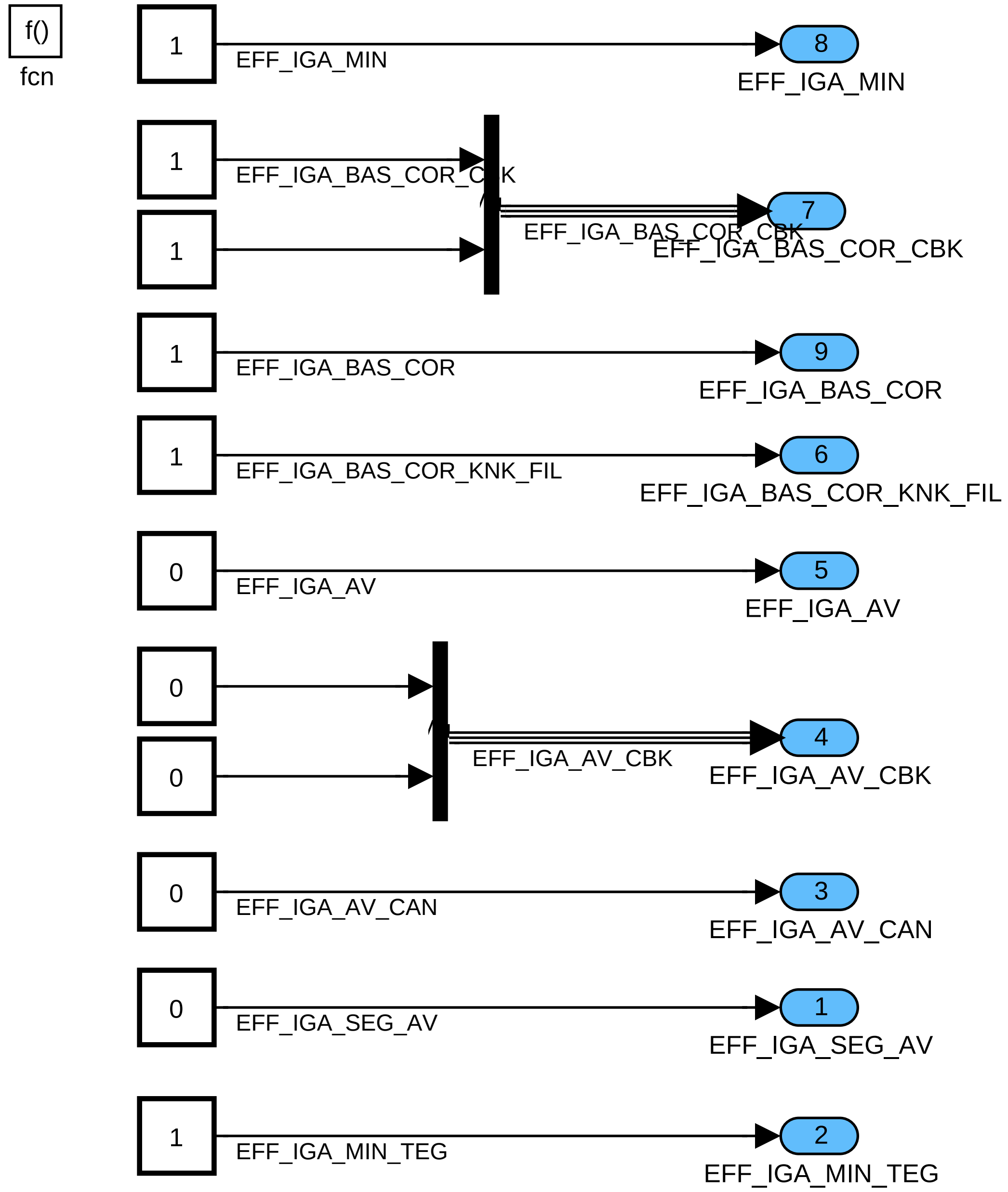


Figure 3 TQDR_M6028/ reset

62.2.1.2 OPERATE

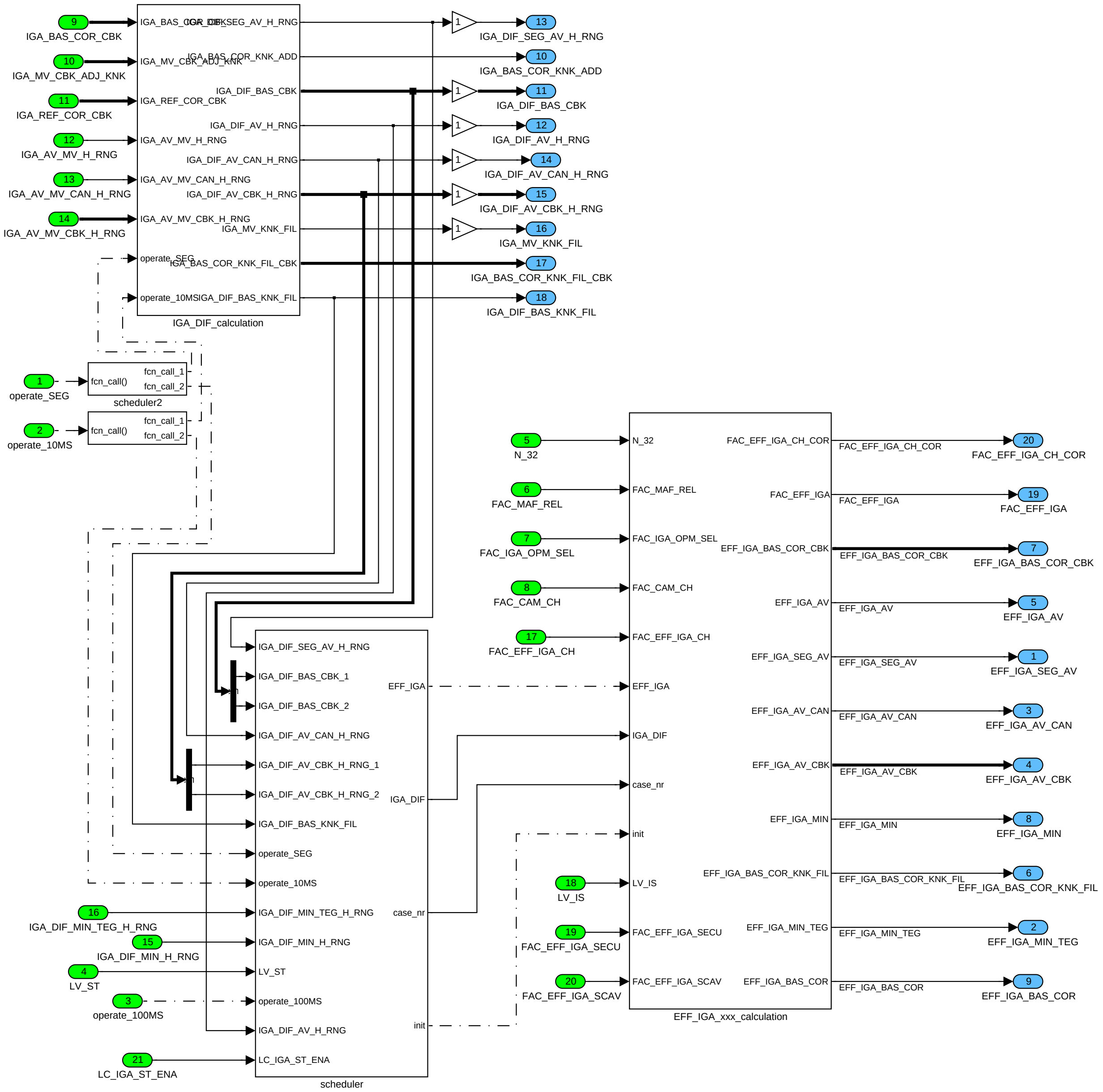


Figure 4 TQDR_M6028/ Operate

IGA_DIF_CALCULATION

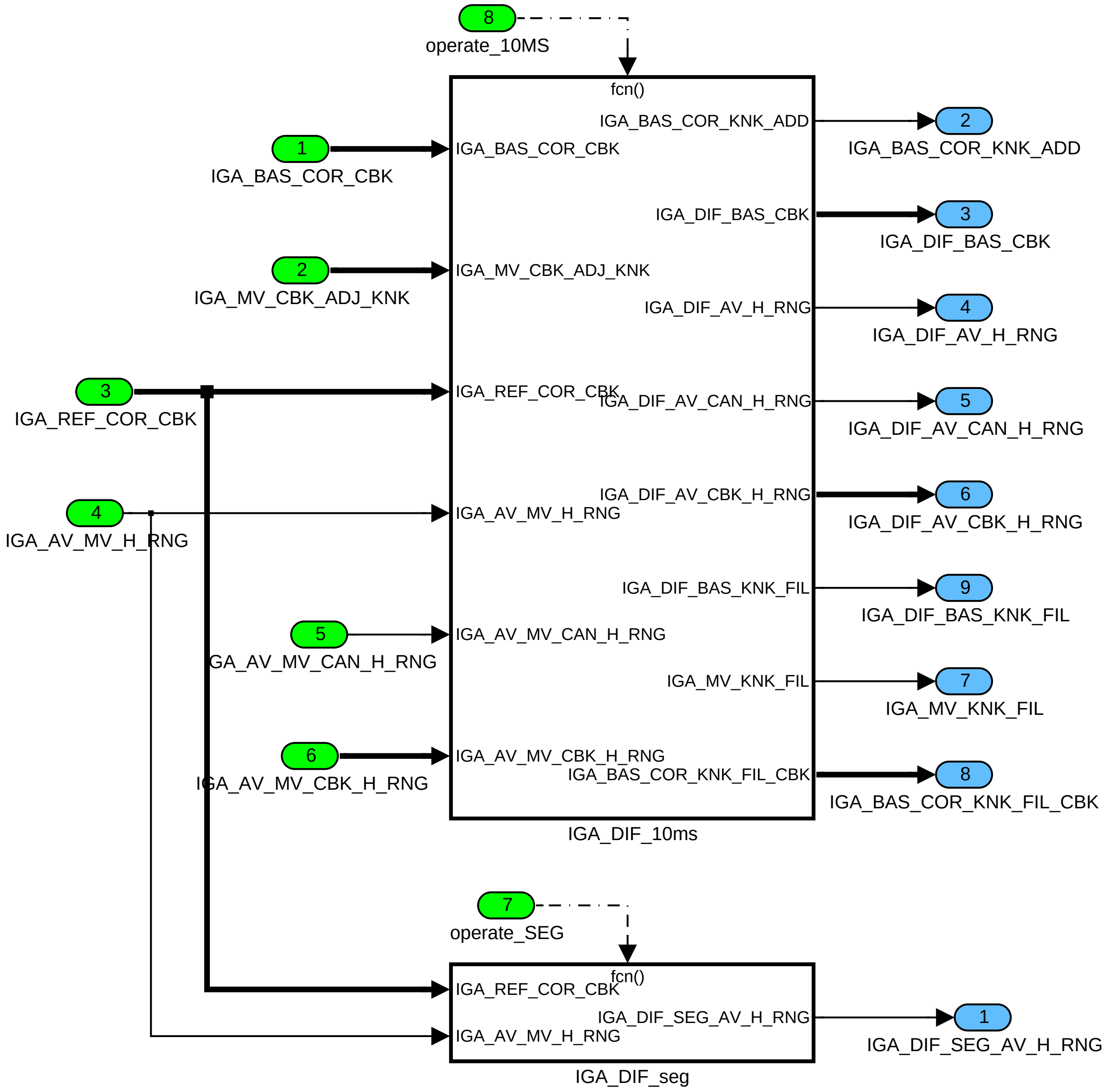


Figure 5 TQDR_M6028/ Operate/ IGA_DIF_calculation

IGA_DIF_10MS

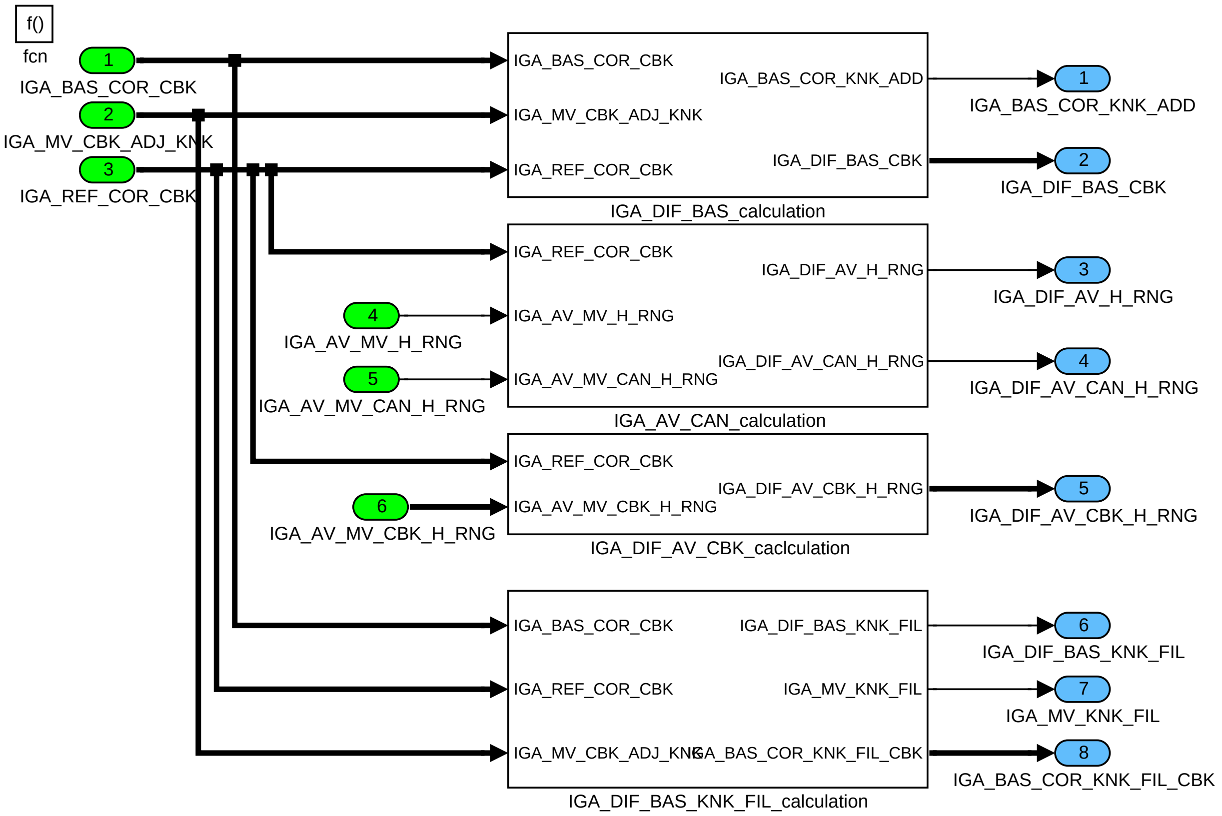


Figure 6 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms

IGA_DIF_BAS_CALCULATION

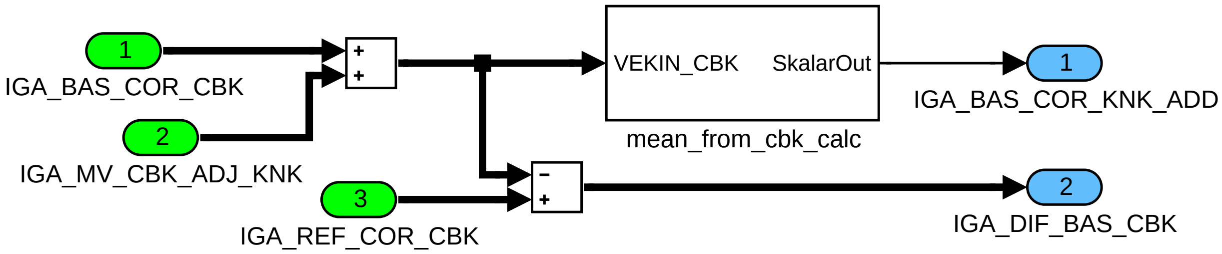


Figure 7 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_DIF_BAS_calculation

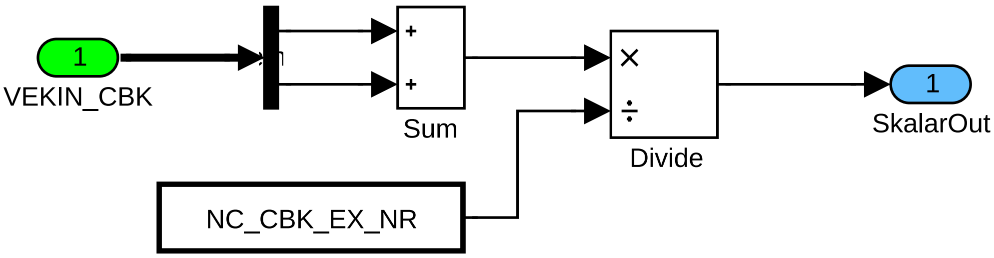


Figure 8 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_DIF_BAS_calculation/ mean_from_cbk_calc

IGA_AV_CAN_CALCULATION

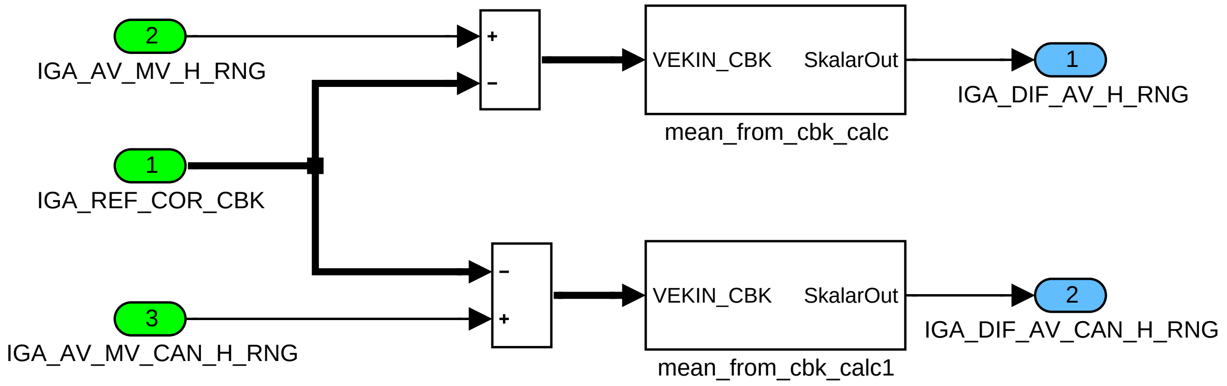


Figure 9 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_AV_CAN_calculation

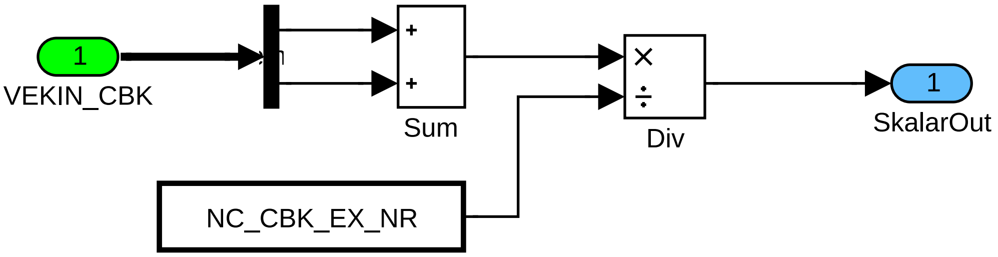


Figure 10 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_AV_CAN_calculation/ mean_from_cbk_calc

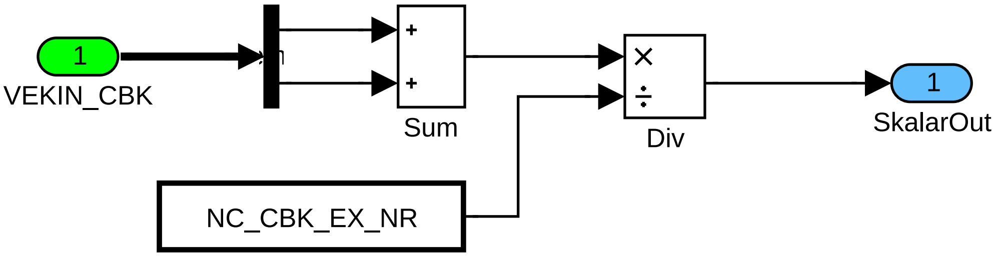


Figure 11 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_AV_CAN_calculation/ mean_from_cbk_calc1

IGA_DIF_AV_CBK_CACLCULATION

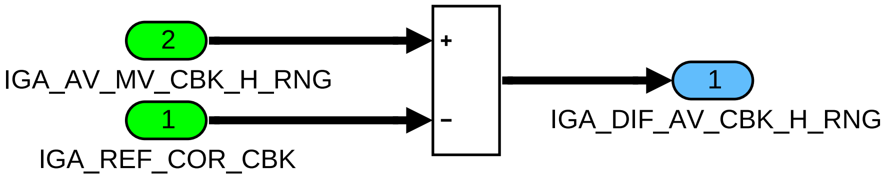


Figure 12 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_DIF_AV_CBK_caclculation

IGA_DIF_BAS_KNK_FIL_CALCULATION

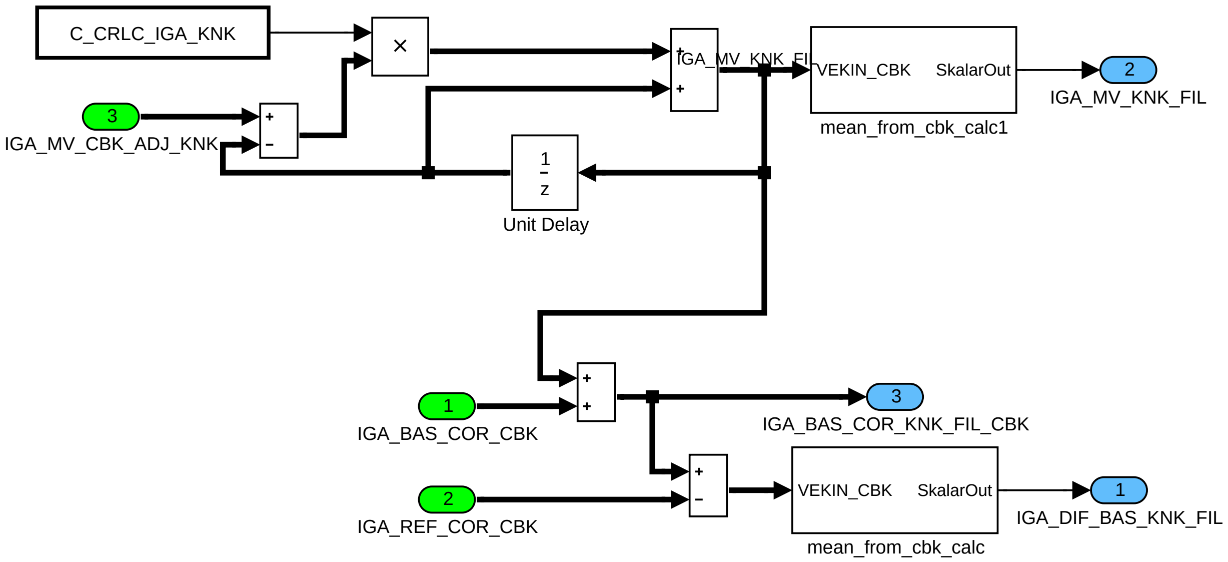


Figure 13 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_DIF_BAS_KNK_FIL_calculation

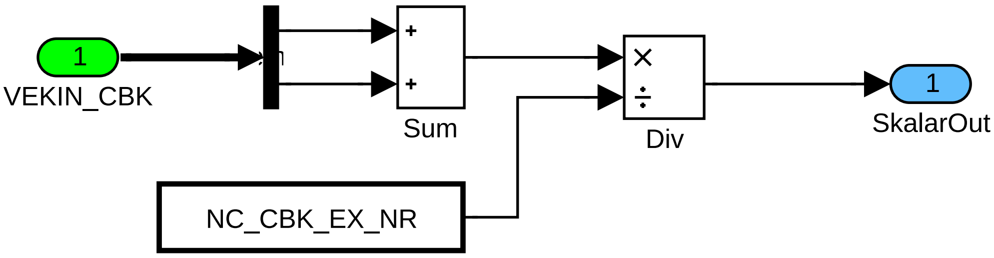


Figure 14 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_DIF_BAS_KNK_FIL_calculation/ mean_from_cbk_calc

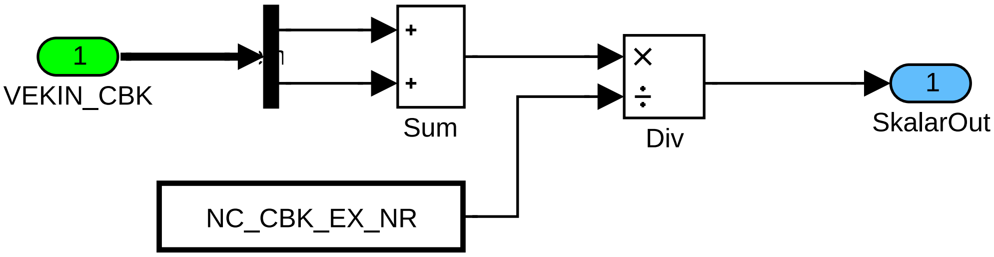


Figure 15 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_10ms/ IGA_DIF_BAS_KNK_FIL_calculation/ mean_from_cbk_calc1

IGA_DIF_SEG

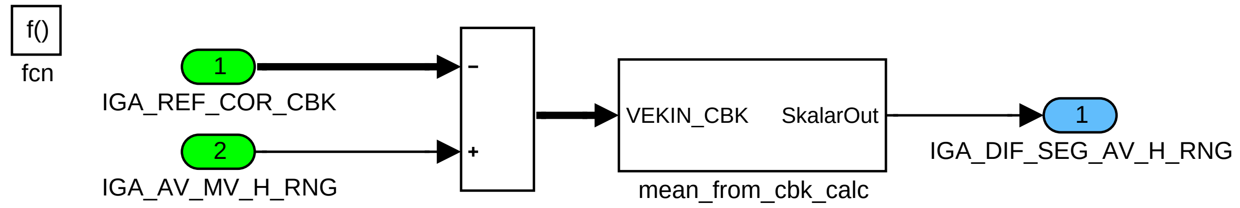


Figure 16 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_seg

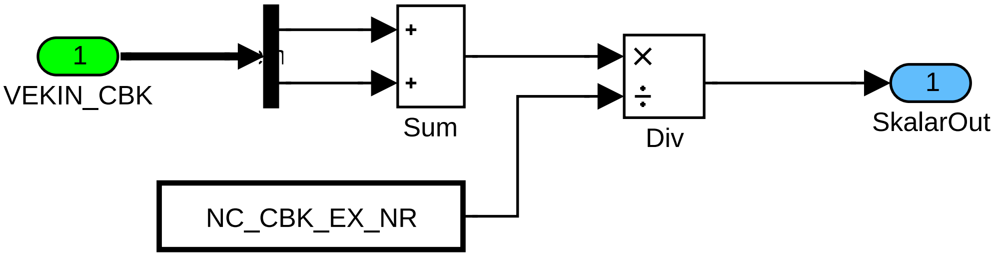


Figure 17 TQDR_M6028/ Operate/ IGA_DIF_calculation/ IGA_DIF_seg/ mean_from_cbk_calc

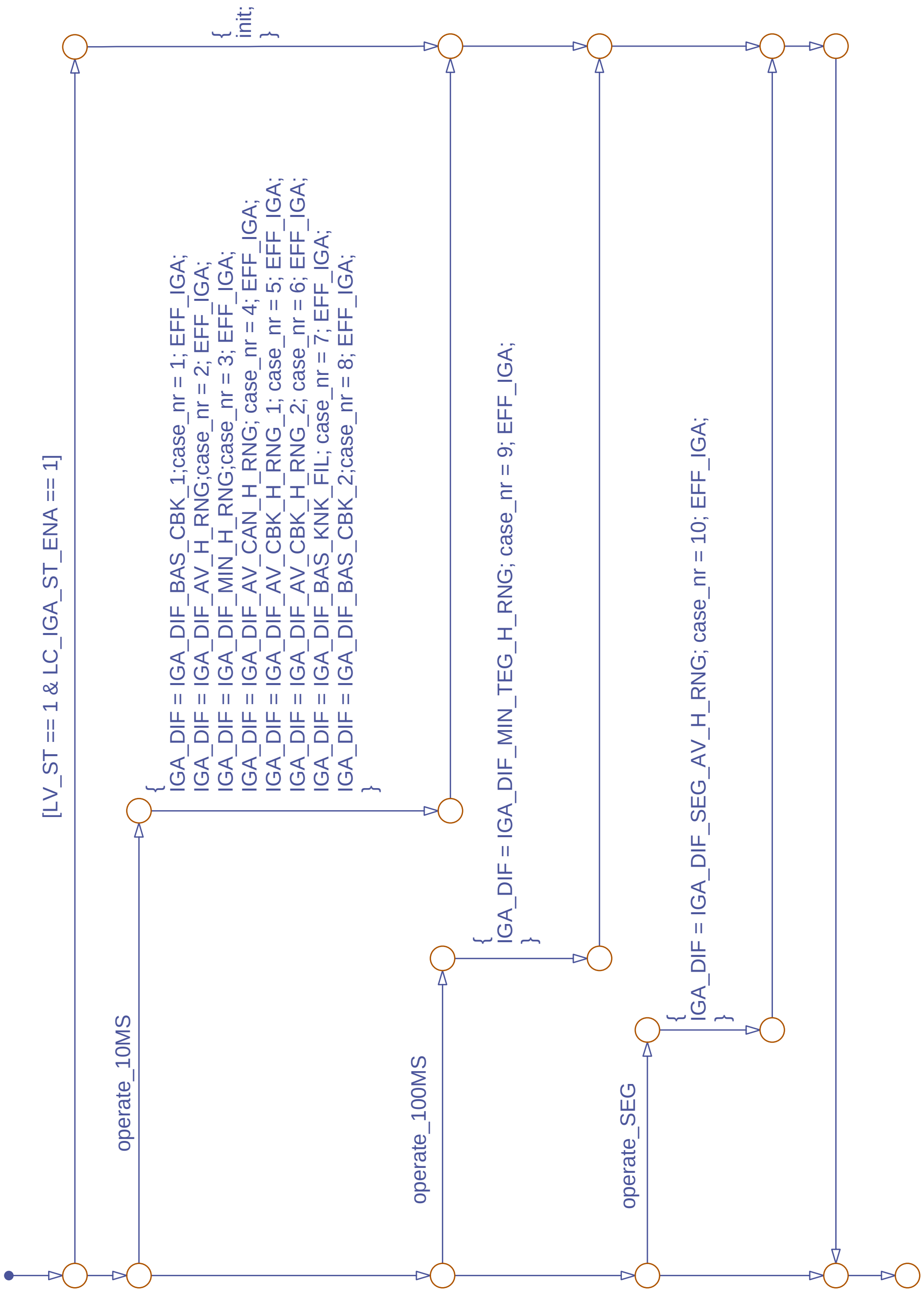


Figure 18 TQDR_M6028/ Operate/ scheduler/ scheduler

EFF_IGA_XXX_CALCULATION

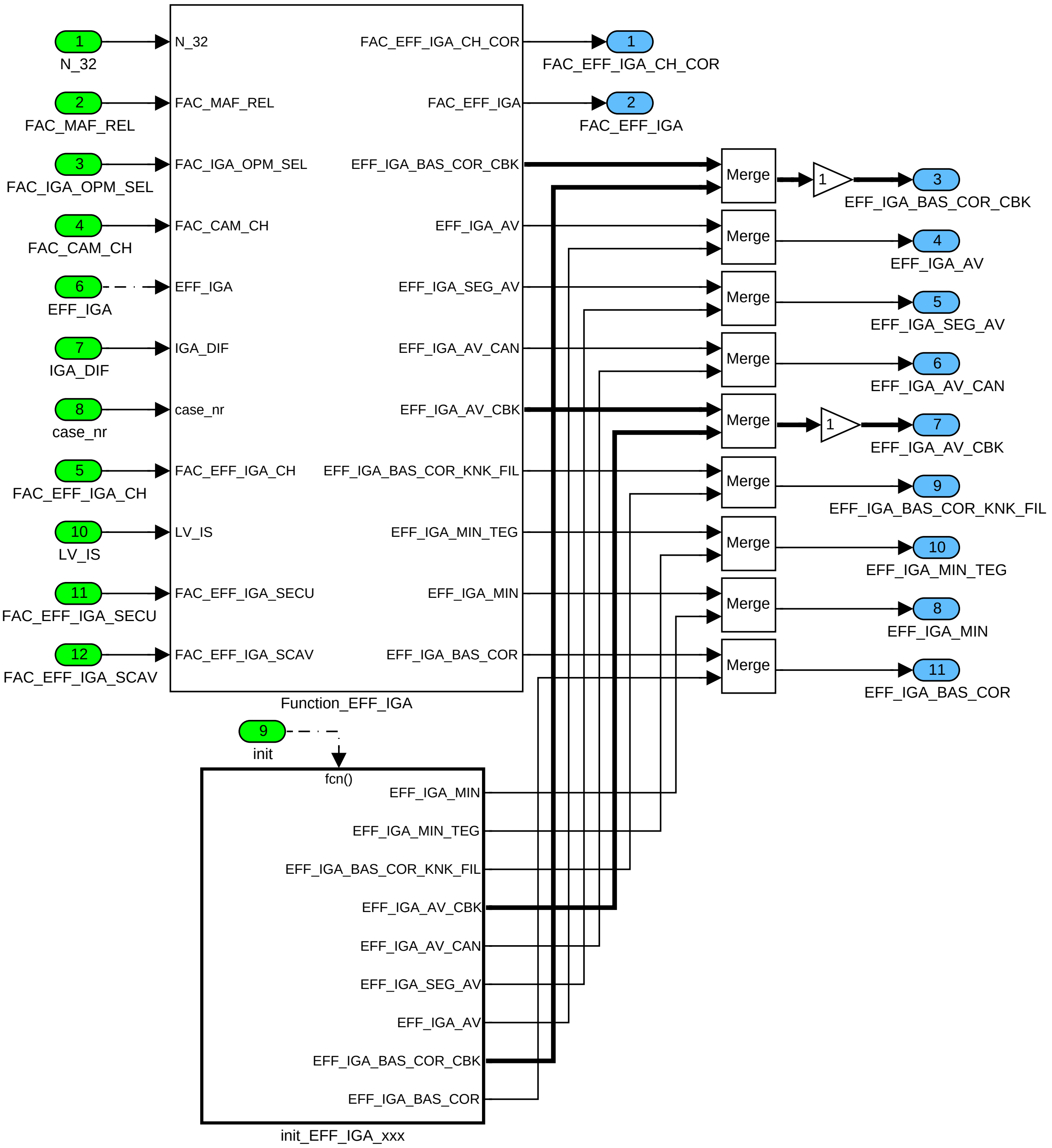


Figure 19 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation

INIT_EFF_IGA_XXX

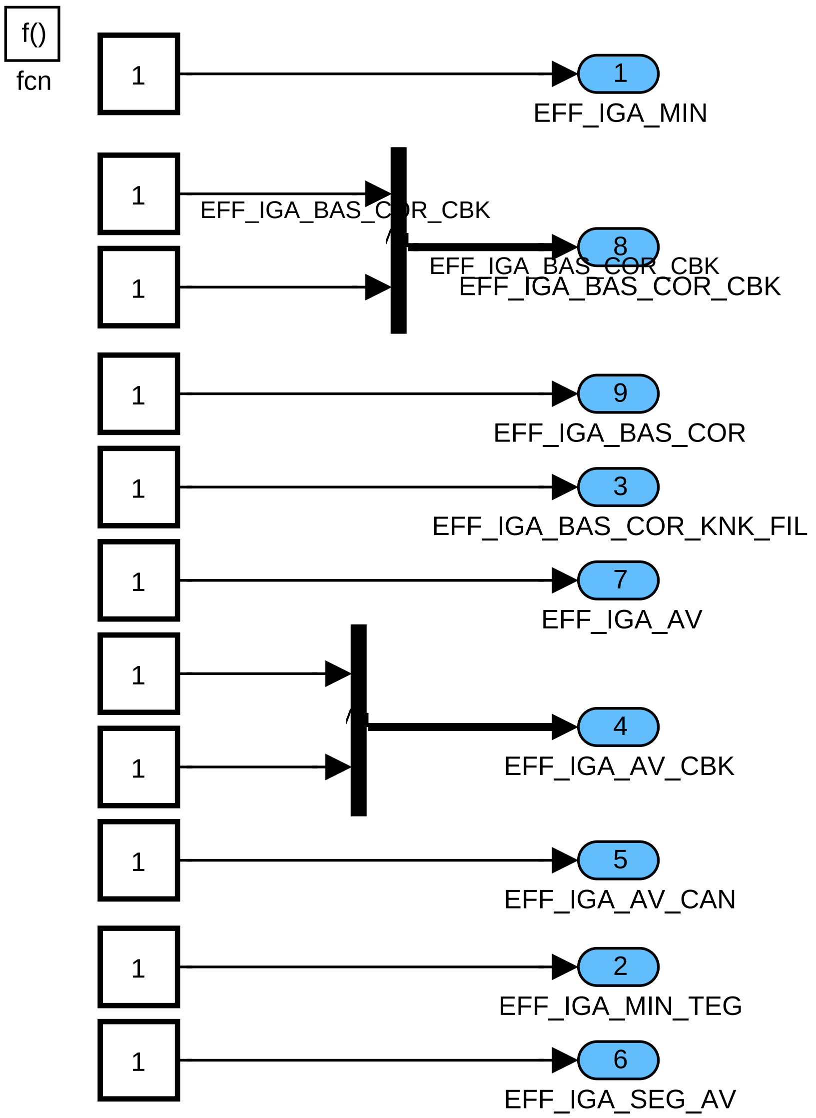


Figure 20 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ init_EFF_IGA_xxx

FUNCTION_EFF_IGA

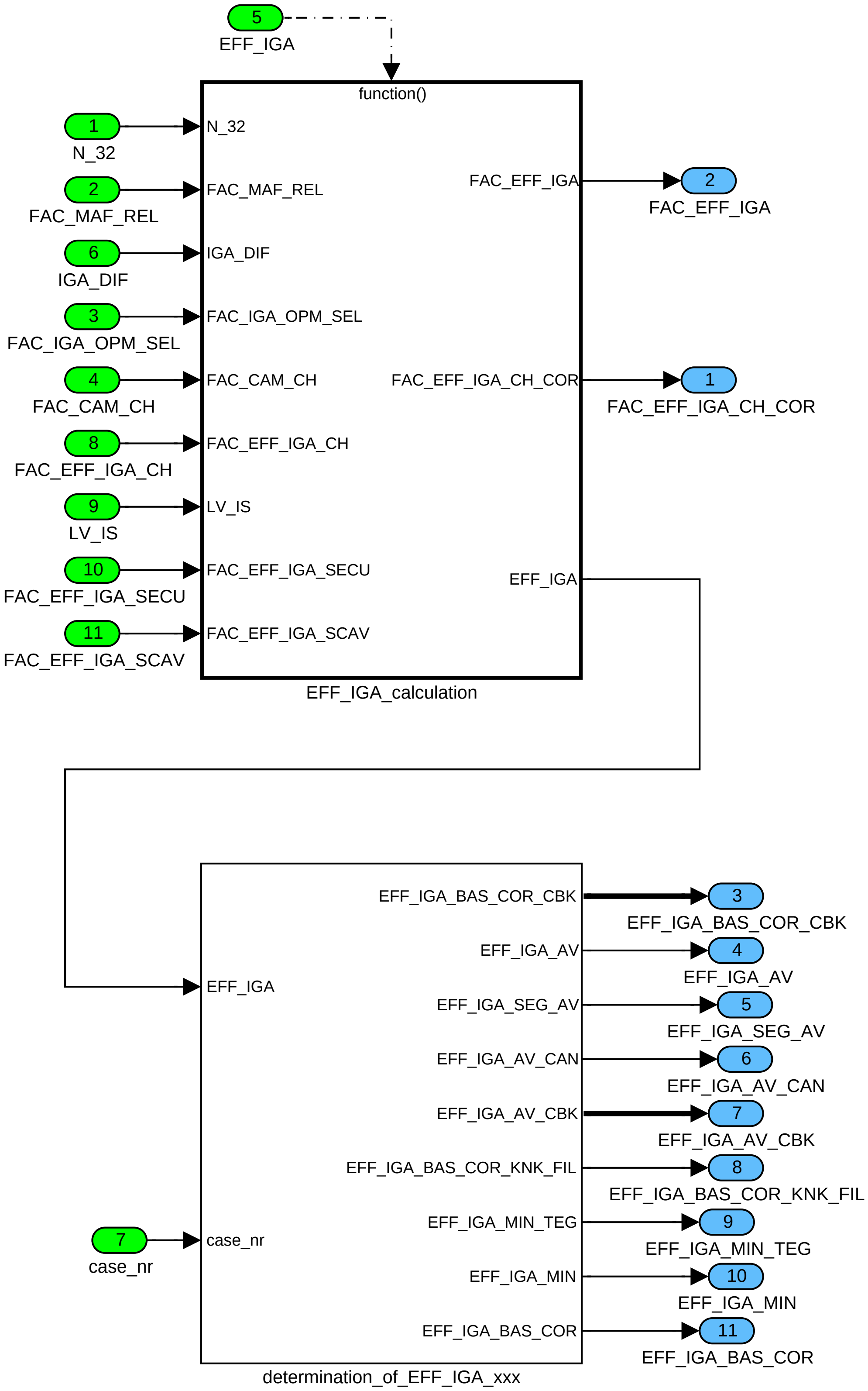


Figure 21 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA

EFF_IGA_CALCULATION

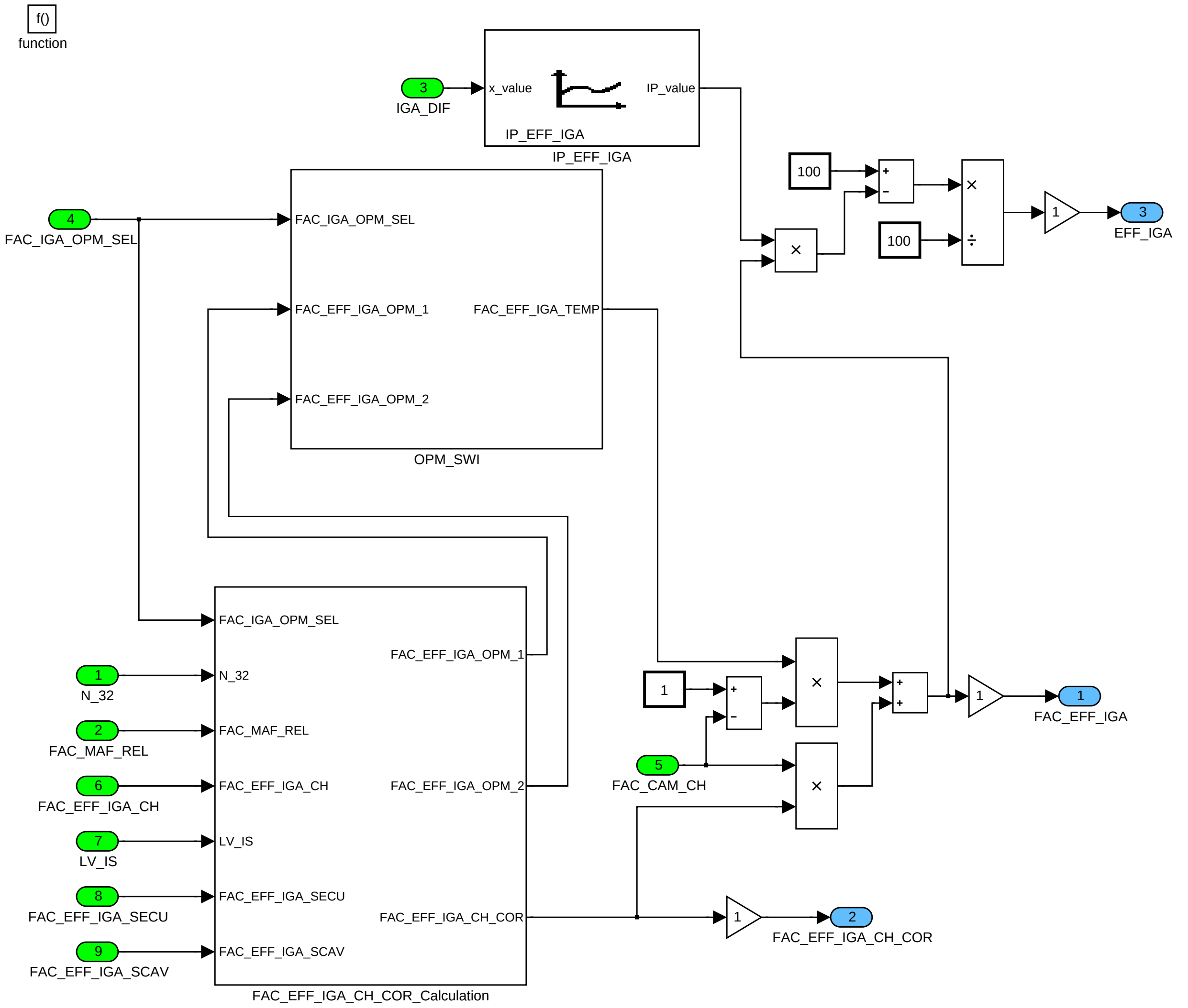


Figure 22 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation

FAC_EFF_IGA_CH_COR_CALCULATION

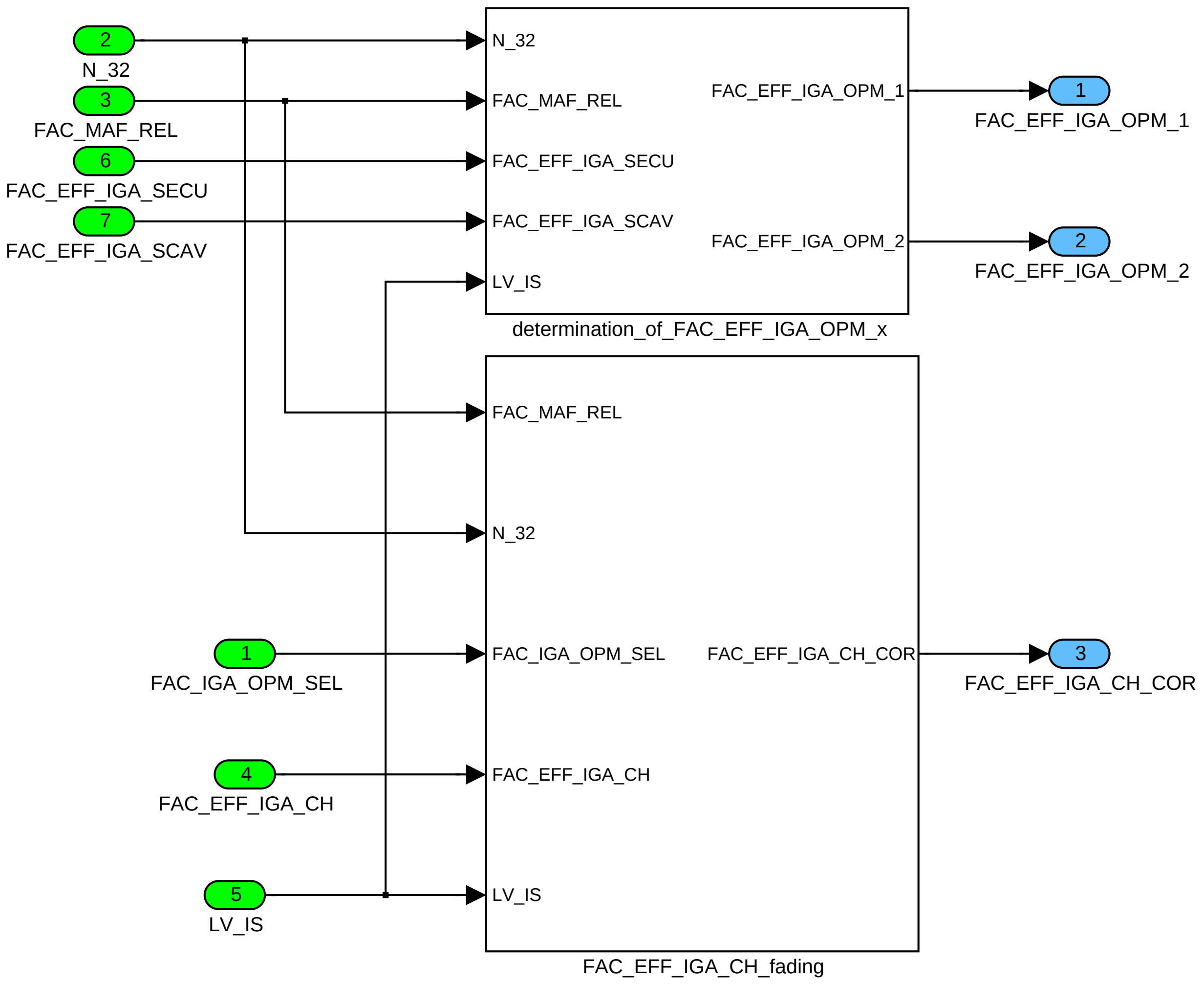


Figure 23 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation

DETERMINATION_OF_FAC_EFF_IGA_OPM_X

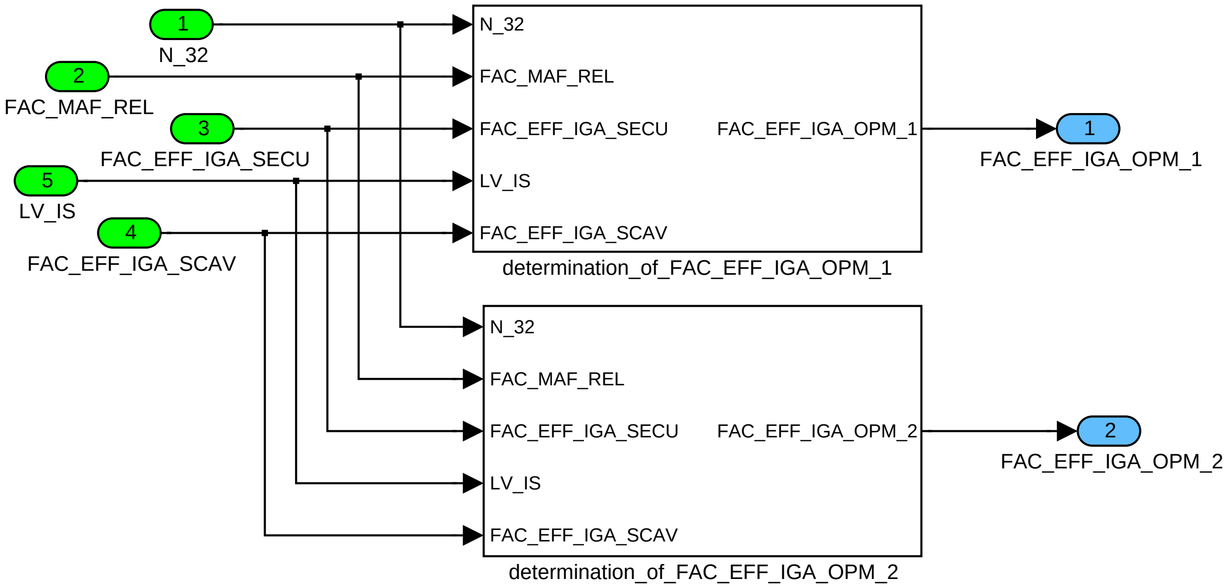


Figure 24 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation/ determination_of_FAC_EFF_IGA_OPM_x

DETERMINATION_OF_FAC_EFF_IGA_OPM_1

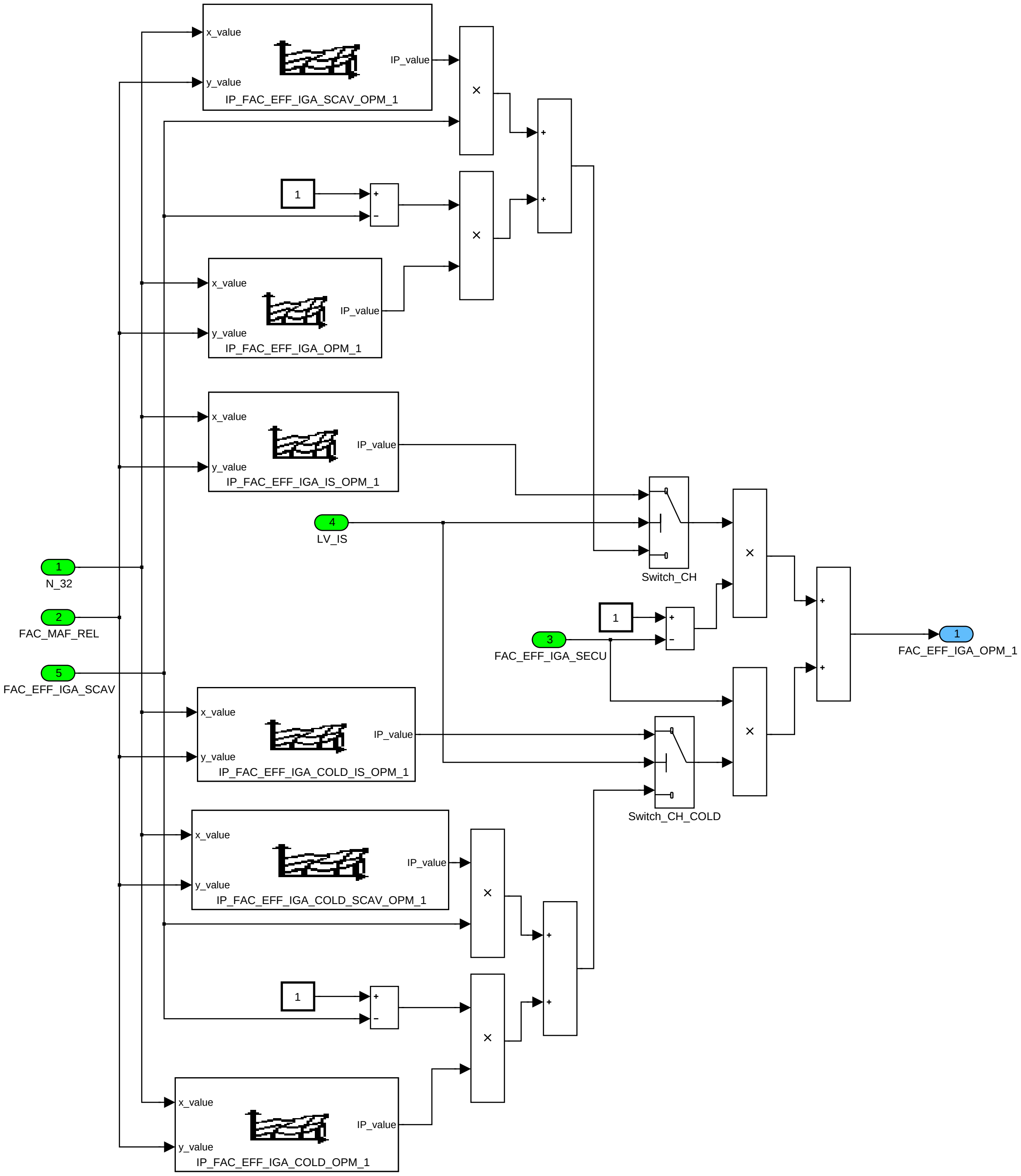


Figure 25 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation/ determination_of_FAC_EFF_IGA_OPM_x/ determination_of_FAC_EFF_IGA_OPM_1

DETERMINATION_OF_FAC_EFF_IGA_OPM_2

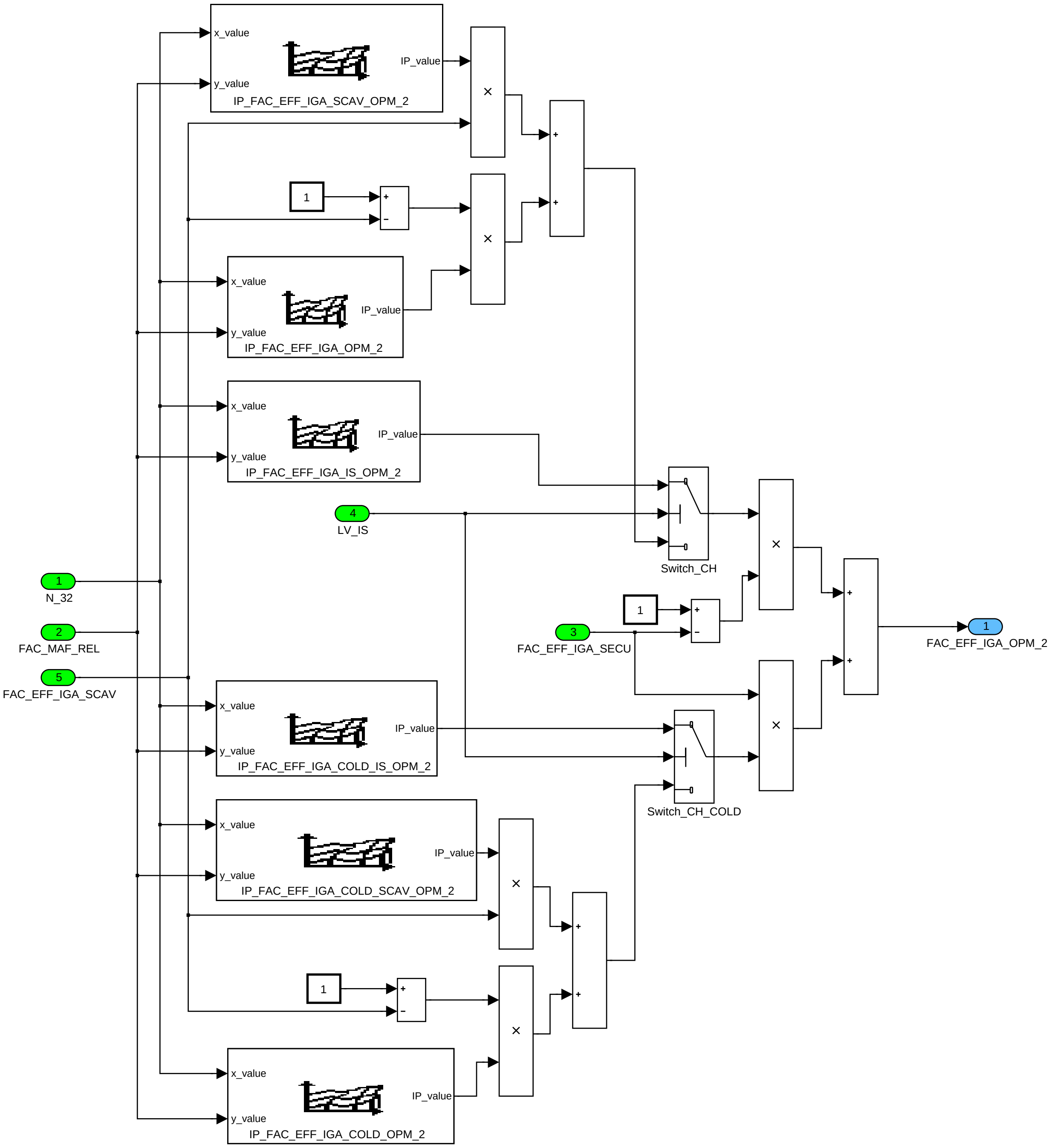


Figure 26 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation/ determination_of_FAC_EFF_IGA_OPM_x/ determination_of_FAC_EFF_IGA_OPM_2

FAC_EFF_IGA_CH_FADING

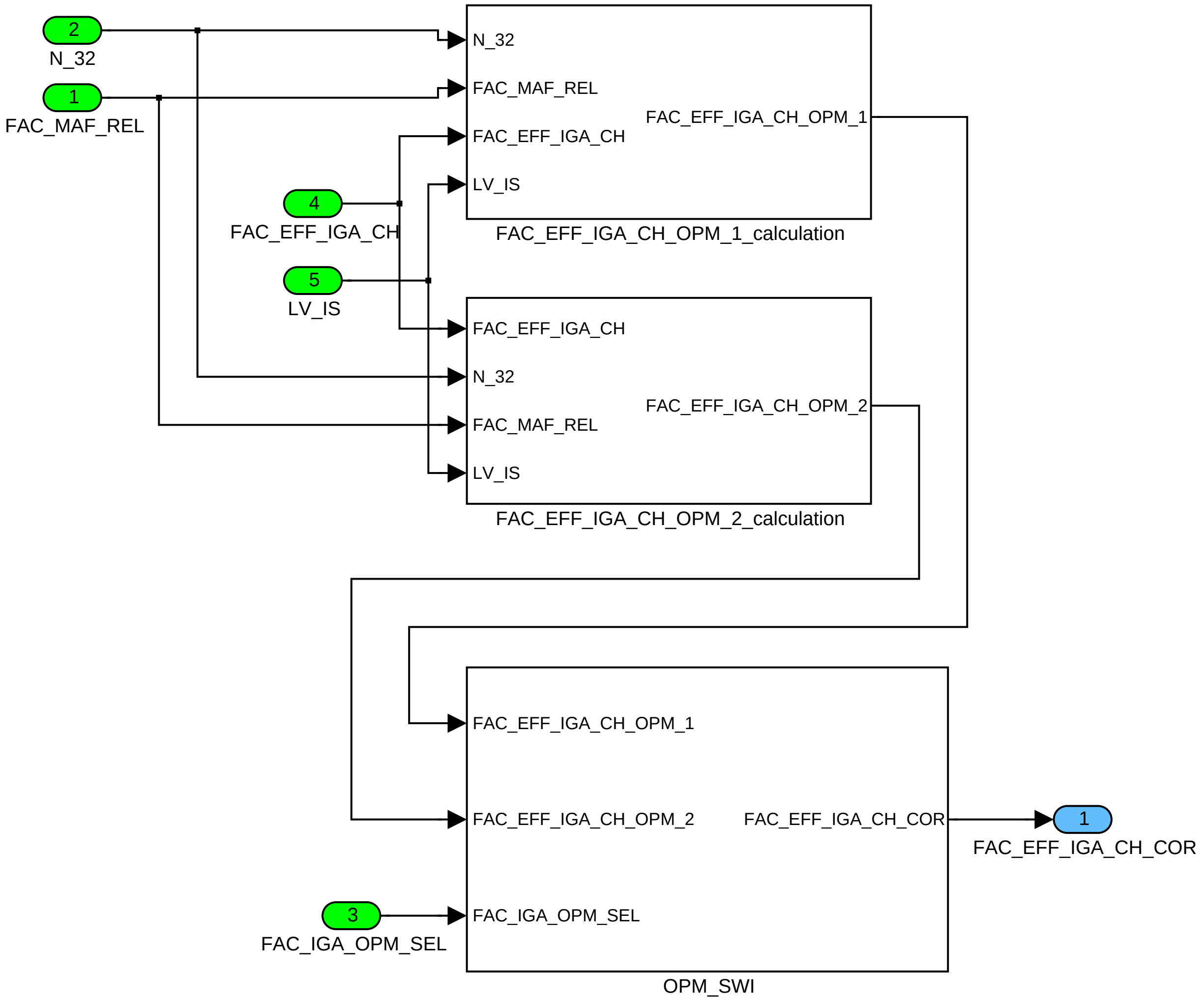


Figure 27 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation/ FAC_EFF_IGA_CH_fading

FAC_EFF_IGA_CH_OPM_1_CALCULATION

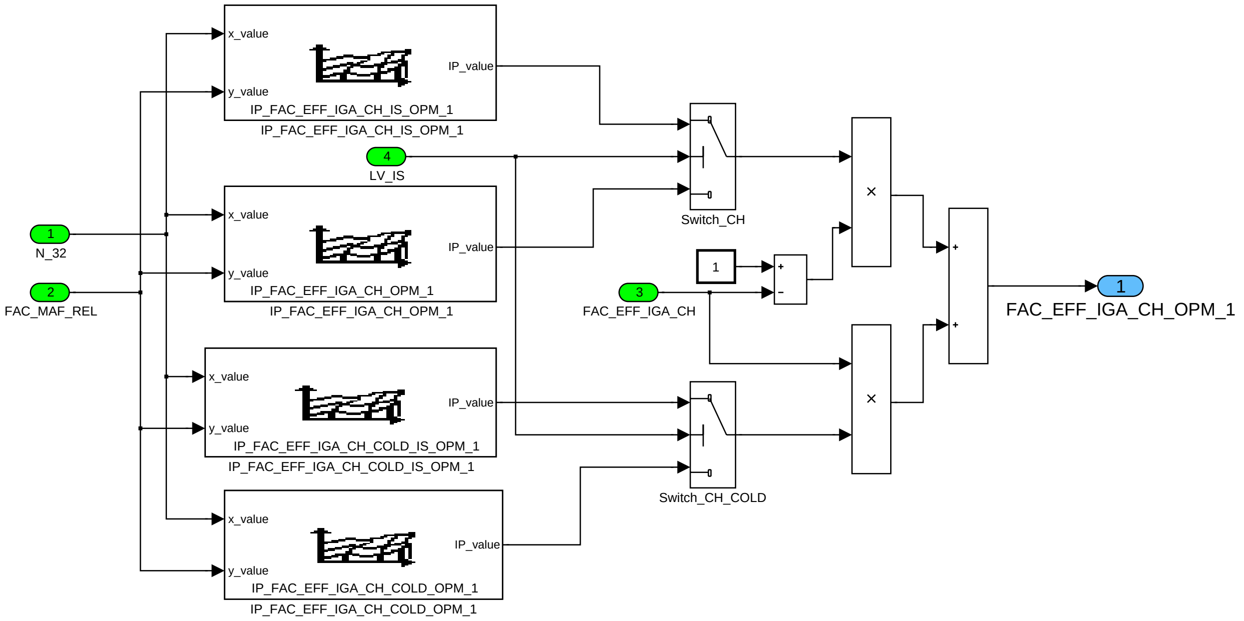


Figure 28 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation/ FAC_EFF_IGA_CH_fading/ FAC_EFF_IGA_CH_OPM_1_calculation

FAC_EFF_IGA_CH_OPM_2_CALCULATION

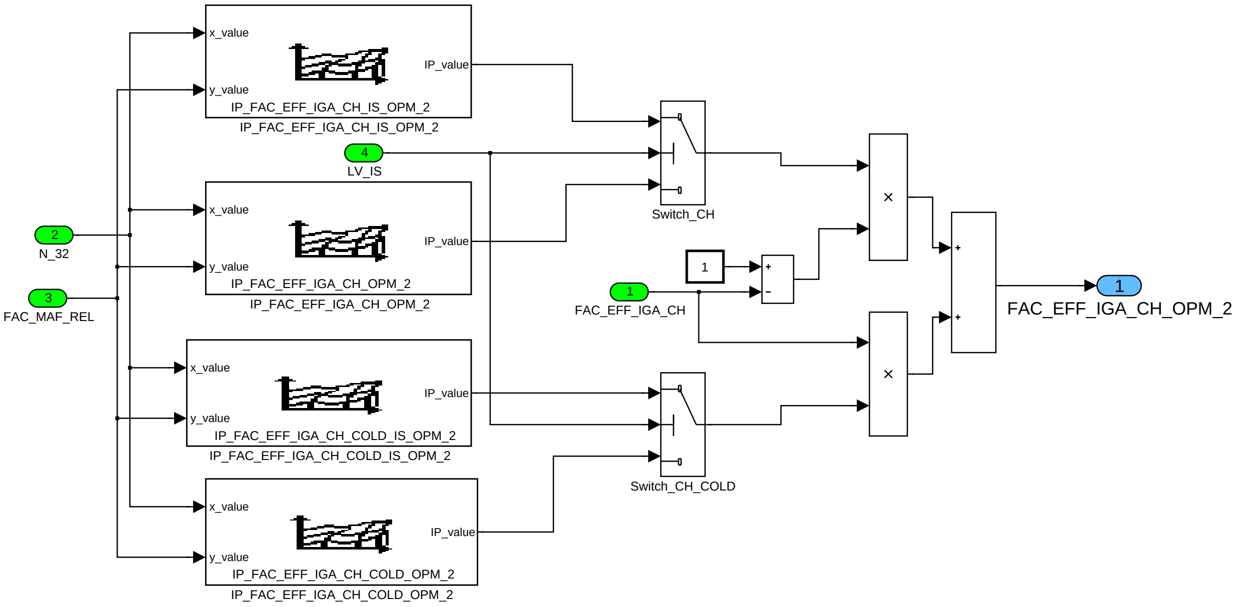


Figure 29 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation/ FAC_EFF_IGA_CH_fading/ FAC_EFF_IGA_CH_OPM_2_calculation

OPM_SWI

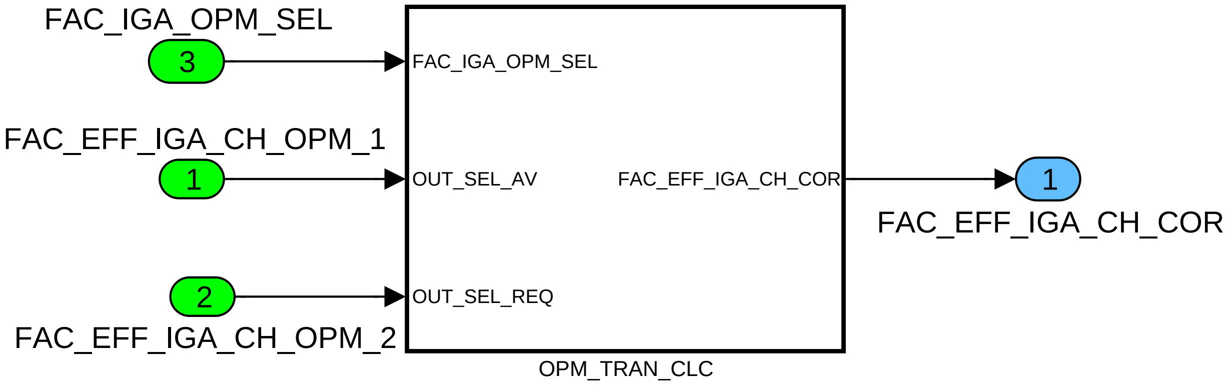


Figure 30 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation/ FAC_EFF_IGA_CH_fading/ OPM_SWI

OPM_TRAN_CLC

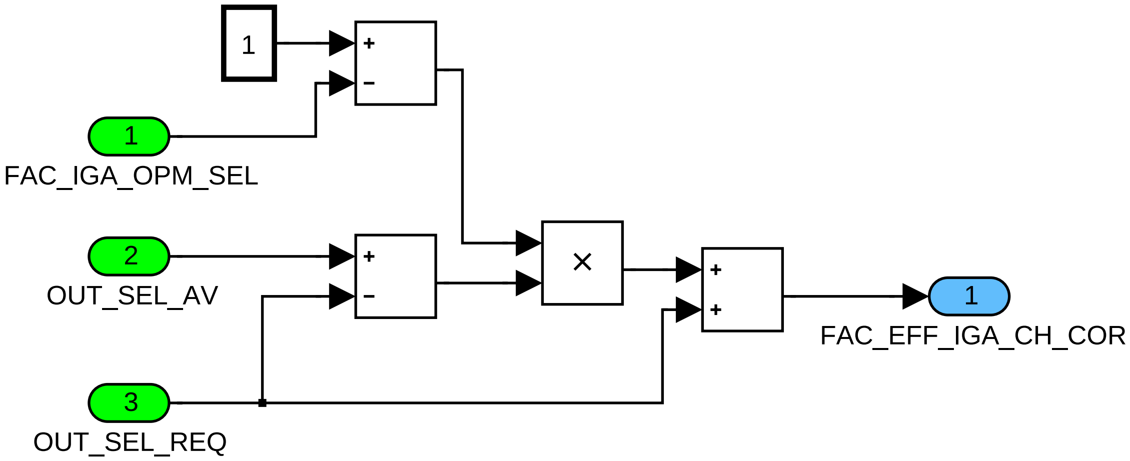


Figure 31 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ FAC_EFF_IGA_CH_COR_Calculation/ FAC_EFF_IGA_CH_fading/ OPM_SWI/ OPM_TRAN_CLC

OPM_SWI

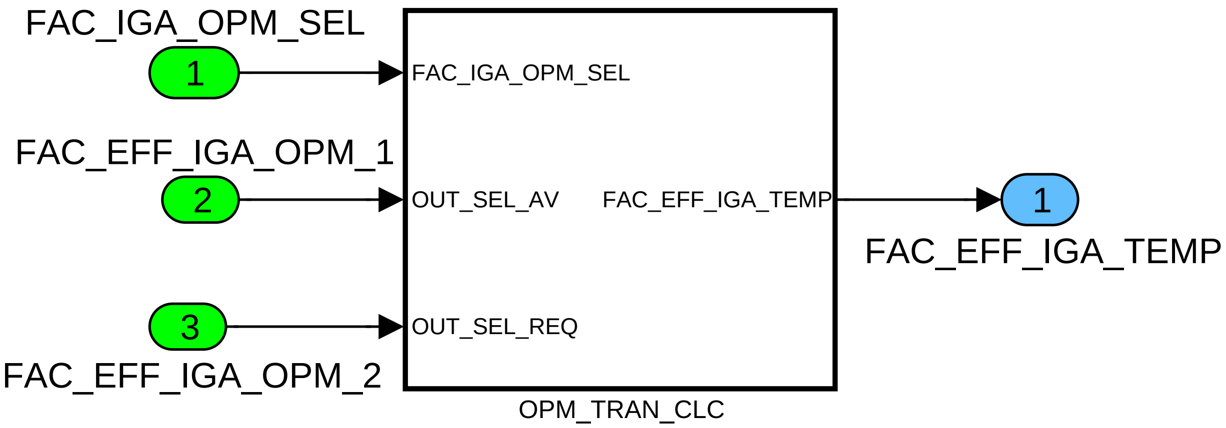


Figure 32 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ OPM_SWI

OPM_TRAN_CLC

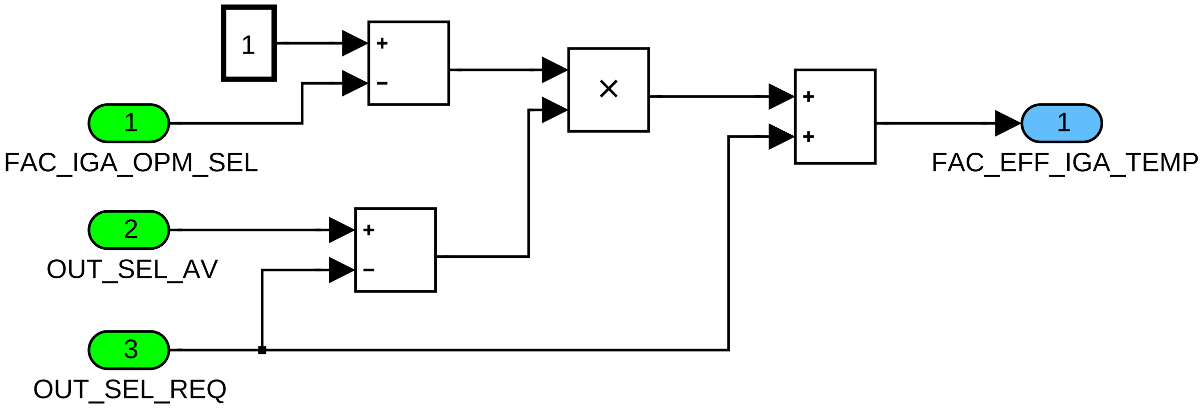
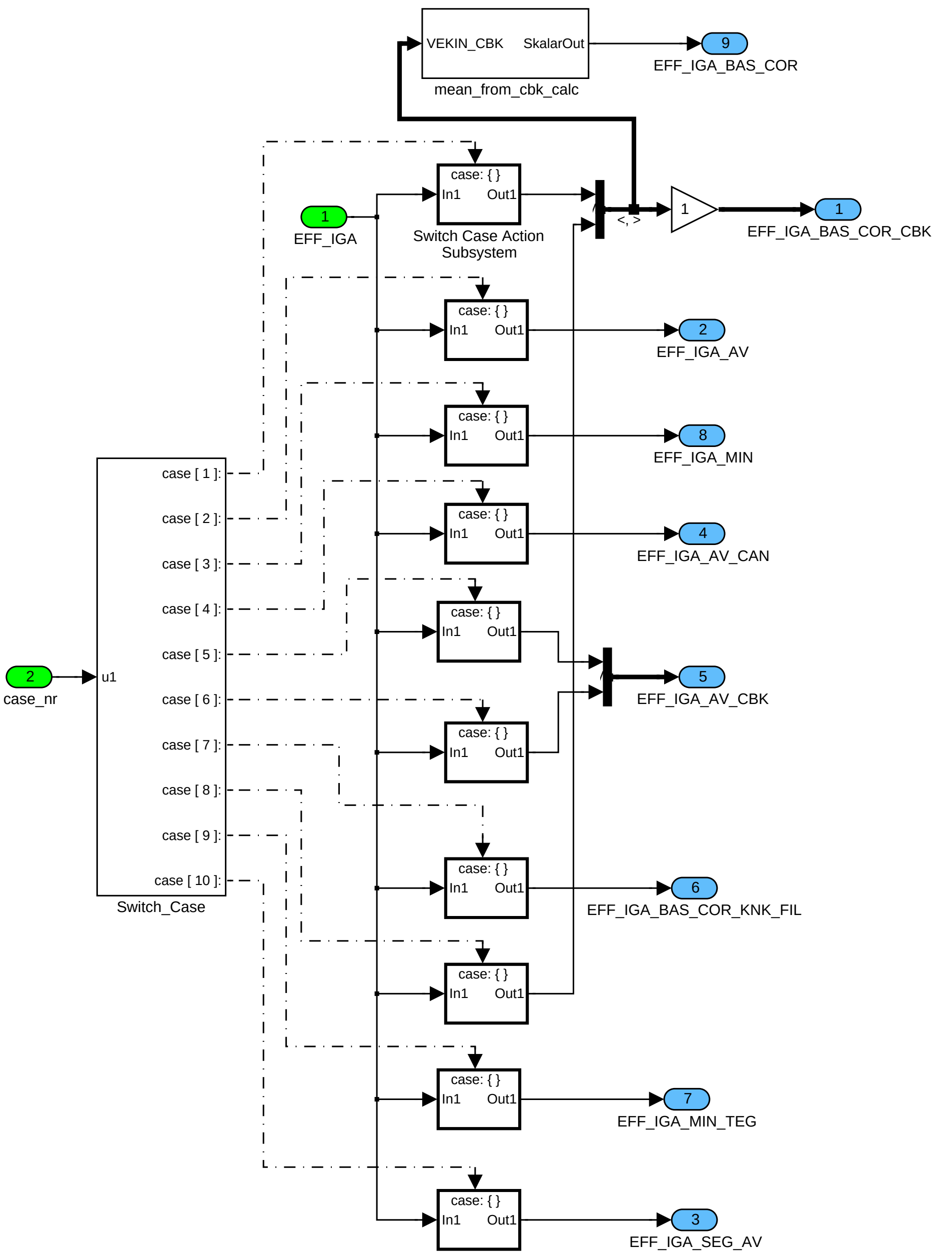


Figure 33 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ EFF_IGA_calculation/ OPM_SWI/ OPM_TRAN_CLC

FUNCTION_EFF_IGA/DETERMINATION_OF_EFF_IGA_XXX



Case 1,8: EFF_IGA_BAS_COR_CBK[i] = function call EFF_IGA (IGA_DIF_BAS_CBK[i])
Case 2: EFF_IGA_AV = function call EFF_IGA (IGA_DIF_AV_H_RNG)
Case 3: EFF_IGA_MIN = function call EFF_IGA (IGA_DIF_MIN_H_RNG)
Case 4: EFF_IGA_AV_CAN = function call EFF_IGA (IGA_DIF_AV_CAN_H_RNG)
Case 5,6:EFF_IGA_AV[i]= function call EFF_IGA (IGA_DIF_AV_H_RNG[i])
Case 7: EFF_IGA_BAS_COR_KNK_FIL = function call EFF_IGA (IGA_DIF_BAS_KNK_FIL_CBK)
Case 9: EFF_IGA_MIN_TEG = function call EFF_IGA (IGA_DIF_MIN_TEG_H_RNG)
Case 10: EFF_IGA_SEG_AV = function call EFF_IGA (IGA_DIF_SEG_AV)

Figure 34 TQDR_M6028/ Operate/ EFF_IGA_xxx_calculation/ Function_EFF_IGA/ determination_of_EFF_IGA_xxx

62.2.1.3 MERGE BLOCK

Signals merged from states reset and operate.

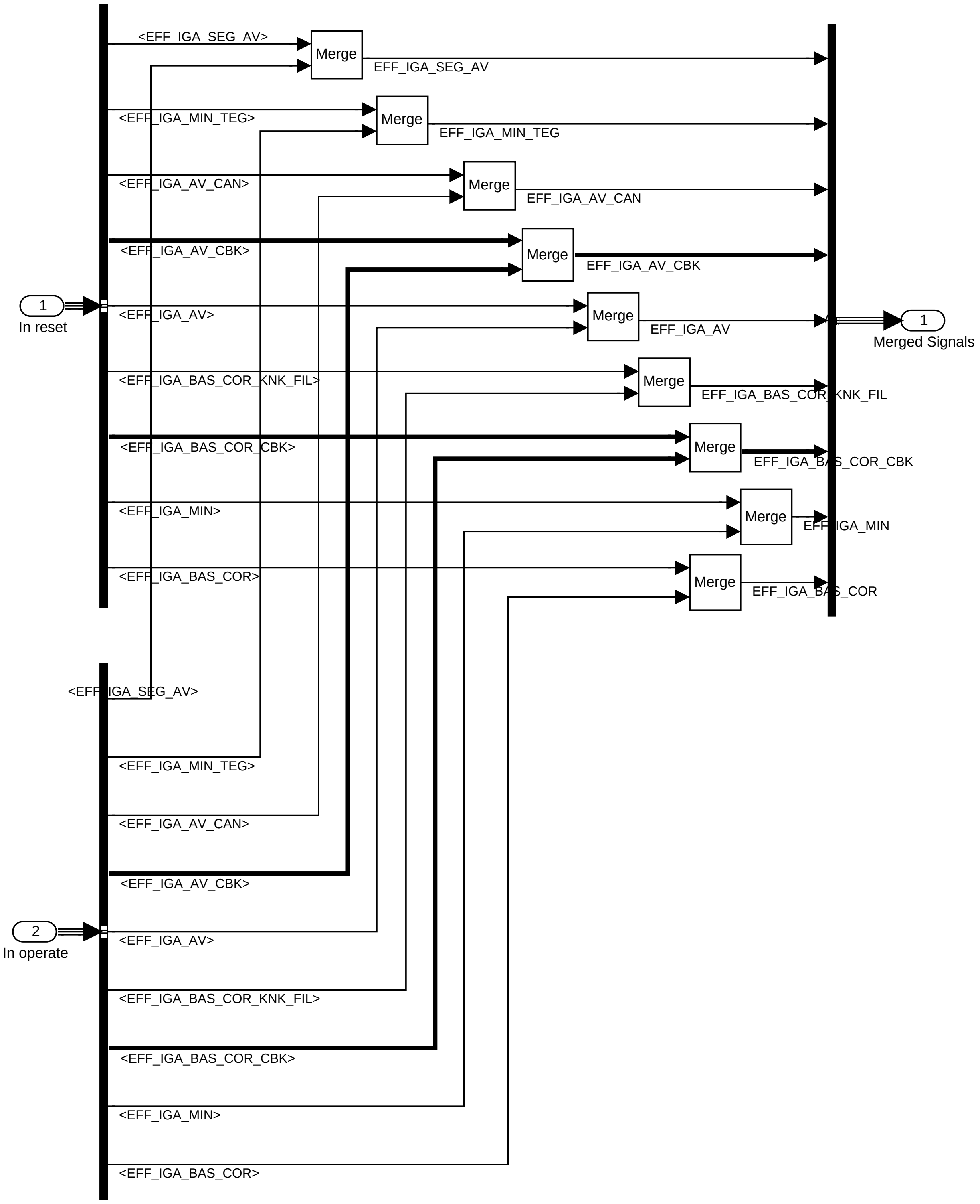


Figure 35 TQDR_M6028/ Merge_block

62.3 (TQM) Setpoint ignition angle

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EFF_IGA_INV_COR_CBK	V	0...4E20H	0...100	0.005	[%]
Setpoint ignition efficiency; inverted and corrected depending on engine state bank selective					
EFF_IGA_INV_COR_CBK_CAN	V	0...4E20H	0...100	0.005	[%]
Setpoint ignition efficiency CAN; inverted and corrected depending on engine state bank selective					
EFF_IGA_SP_CBK[NC_CBK_EX_NR]	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Setpoint ignition efficiency bank selective					
EFF_IGA_SP_CBK_1[NC_CBK_EX_NR]	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Setpoint ignition efficiency bank selective					
EFF_IGA_SP_CBK_1_CAN[NC_CBK_EX_NR]	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Setpoint ignition efficiency for CAN bank selective					
EFF_IGA_SP_CBK_CAN[NC_CBK_EX_NR]	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Setpoint ignition efficiency for CAN bank selective					
IGA_DIF_SP_CBK_CAN_H_RNG[NC_CBK_EX_NR]	O/V	0...B40H	0...-180	-0.0625	[°CRK]
Setpoint subtrahend ignition angle in wide range for CAN bank selective					
IGA_DIF_SP_CBK_H_RNG	O/V	0...B40H	0...-180	-0.0625	[°CRK]
Setpoint subtrahend ignition angle in wide range bank selective					
IGA_DIF_SP_H_RNG	O/V	0...B40H	0...-180	-0.0625	[°CRK]
Setpoint subtrahend ignition angle in wide range					
IGA_SP_CAN_H_RNG	O/V	0...B40H	0...-180	-0.0625	[°CRK]
Setpoint ignition angle from torque management for CAN					
IGA_SP_CBK_CAN_H_RNG[NC_CBK_EX_NR]	O/V	FA60...5A0H	-90...90	0.0625	[°CRK]
Setpoint ignition angle from torque management for CAN bank selective					
IGA_SP_CBK_H_RNG[NC_CBK_EX_NR]	O/V	FA60...5A0H	-90...90	0.0625	[°CRK]
Setpoint ignition angle from torque management bank selective					
IGA_SP_H_RNG	O/V	FA60...5A0H	-90...90	0.0625	[°CRK]
Setpoint ignition angle from torque management					
LV_TQ_IGA_ACT	O/V	0...1H	0...1	1	[-]
Logical variable torque intervention by spark retard is active					
LV_TQ_IGA_REQ	V	0...1H	0...1	1	[-]
Logical variable torque intervention by spark retard is request					

Input data:

EFF_IGA_BAS_COR_CBK[NC_CBK_EX_NR]	EFF_LAMB_AV_CBK[NC_CBK_EX_NR]	EFF_SCC_AV	FAC_EFF_IGA
IGA_REF_COR_CBK[NC_CBK_EX_NR]	LV_ACT_AJ	LV_HOM_RUN	LV_REQ_ISC
LV_TQ_IGA_ENA	TQI_REF	TQI_SP_CAN	TQI_SP_CBK[NC_CBK_EX_NR]

FUNCTION DESCRIPTION:

General information:

For homogeneous (*LV_HOM_RUN = 1*) the module ignition angle setpoint delivers IGA_DIF_SP_H_RNG as an output which depends on the torque setpoint TQI_SP_CBK[NC_CBK_EX_NR]. For the calculation of the setpoint of ignition efficiency EFF_IGA_SP the actual cylinder fuel cut-off efficiency EFF_SCC_AV, the Exhaust Gas Recirculation efficiency and the actual lambda efficiency EFF_LAMB_AV are considered which are multiplied with the reference torque value TQI_REF.

By evaluation of the standardised ignition characteristic the spark retard value IGA_DIF_SP_H_RNG is calculated (*a negative value*) and will be applied on the reference ignition angle to define the ignition angle setpoint for torque interventions. In a system with variable valve lift the standardised ignition characteristic depends on the actual engine mode. Two modes are distinguished: load control via inlet valve lift and load control via the main throttle. To pay this different load controls attention the basic standardised ignition characteristic given by IP_IGA_DIF_SP_H_RNG is corrected with engine mode depending factor FAC_IGA_DIF.

Hereafter, it is determined if a fast torque intervention request implies a spark retard for its application through the logical flag LV_TQ_IGA_REQ.
If EFF_IGA_SP >= EFF_IGA_BAS_COR then no torque intervention is necessary and LV_TQ_IGA_ACT is set to 0 since there is no spark retard requested by torque. In this case IGA_DIF_SP_H_RNG is set to the passive value which is the maximum value for difference ignition angle *(=0 °CRK*). Then IGA_DIF_SP_H_RNG is not a limitation for the basic ignition angle in any case.

If EFF_IGA_SP < (EFF_IGA_BAS_COR – Threshold) then a significant spark retard resulting from a torque request has to be applied. For this LV_TQ_IGA_REQ is set to 1.
The value of the Threshold used here depends on both if an anti-jerk intervention is activated and if the engine is in idle speed phase. In normal condition (*no anti-jerk nor idle speed*) the threshold is Threshold = C_EFF_IGA_THD_1. In case of Anti-Jerk intervention a specific threshold will be used (*Threshold_1 = C_EFF_IGA_THD_AJ*) ; in idle speed also another specific threshold (*Threshold_2 = C_EFF_IGA_THD_IS*) instead of the normal one. Finally, the minimal threshold between both preceding is retained to be *applied (Threshold = Min(Threshold_1,Threshold_2)*).
IGA_DIF_SP_H_RNG is deduced from EFF_IGA_SP and is added (*negative value*) to the reference ignition angle to get the ignition angle setpoint for torque intervention.

The main reason for the existence of these two cases is to reduce ignition angle jitters if torque values are jittering (TQI_SP_CBK[NC_CBK_EX_NR] *and TQI_REF*). If no significant torque intervention is recognised then IGA_DIF_SP_H_RNG is set to the passive value 0°CRK. In this case the torque controlled ignition angle setpoint has definitely no impact any more to the applied ignition angle which is then only determined from the basic ignition angle and knock control.

This functionality allows choosing what parameter (*Ignition, Injection, ETC, etc…*) will be impacted first by a fast torque intervention and how this torque reduction request will be handled.

IGA_SP_CAN_H_RNG is calculated like IGA_SP_H_RNG only with calculations depending on TQI_SP_CAN.

Application conditions:

Activation: LV_HOM_RUN = 1

Deactivation: LV_HOM_RUN = 0

Initialisation: IGA_SP_CBK_H_RNG[NC_CBK_EX_NR] = IGA_SP_H_RNG = 60°CRK **and**

IGA_SP_CBK_CAN_H_RNG[NC_CBK_EX_NR] = IGA_SP_CAN_H_RNG = 60°CRK **and**
LV_TQ_IGA_ACT = 0 at reset

Update Rate: 10 ms

Formula section:

```
IF      EFF_SCC_AV > 0
then    EFF_IGA_SP_CBK_1[NC_CBK_EX_NR] = TQI_SP_CBK[NC_CBK_EX_NR] / (TQI_REF * EFF_SCC_AV * EFF_LAMB_AV_CBK[NC_CBK_EX_NR] )
else    EFF_IGA_SP_CBK_1[NC_CBK_EX_NR]
    =TQI_SP_CBK[NC_CBK_EX_NR]/(TQI_REF* EFF_LAMB_AV_CBK[NC_CBK_EX_NR] )
endif
EFF_IGA_SP_CBK_1_CAN[NC_CBK_EX_NR] = TQI_SP_CAN / (TQI_REF * EFF_SCC_AV * EFF_LAMB_AV_CBK[NC_CBK_EX_NR] )
```

<u>note</u>: if division by 0 then set EFF_IGA_SP_CBK_1 to maximum Hexa value.

EFF_IGA_SP_CBK[NC_CBK_EX_NR]=Min(EFF_IGA_SP_CBK_1[NC_CBK_EX_NR] , 1.0)

EFF_IGA_SP_CBK_CAN[NC_CBK_EX_NR]=Min(EFF_IGA_SP_CBK_1_CAN[NC_CBK_EX_NR] , 1.0)

Those preliminary calculations have to be done all the time whatever should be the value of LV_TQ_IGA_ACT.

Fast torque intervention request determination

Preliminary calculations for threshold:

```
If      LV_ACT_AJ = 1
Then            Threshold_1 = C_EFF_IGA_THD_AJ
Else            Threshold_1 = C_EFF_IGA_THD_1
Endif
If      LV_REQ_ISC = 1
Then            Threshold_2 = C_EFF_IGA_THD_IS
Else            Threshold_2 = C_EFF_IGA_THD_1
Endif
And             Threshold = Min(Threshold_1,Threshold_2)
```

Threshold, Threshold_1, Threshold_2 here are only an intermediate variable to clarify those choices on the Torque intervention.

Fast Torque Intervention request determination upon Threshold
(Reduction to a global variable instead of cylinder bank selective approach):

```
If      (EFF_IGA_SP_CBK[CBK_1]<(EFF_IGA_BAS_COR_CBK[CBK_1] - Threshold)
        Or
        (EFF_IGA_SP_CBK[CBK_2]<(EFF_IGA_BAS_COR_CBK[CBK_2] - Threshold))
Then            LV_TQ_IGA_REQ = 1
Elseif  EFF_IGA_SP_CBK[CBK_1] >= EFF_IGA_BAS_COR_CBK[CBK_1]
        Or
        EFF_IGA_SP_CBK[CBK_2] >= EFF_IGA_BAS_COR_CBK[CBK_2]
        then            LV_TQ_IGA_REQ = 0
        else            LV_TQ_IGA_REQ $_{n}$ = LV_TQ_IGA_ACT $_{n-1}$
Endif
```

//LV_TQ_IGA_ACT determination:

```
If      C_TQ_IGA_INH = 0
```

```
Then    LV_TQ_IGA_ACT = LV_TQ_IGA_ENA
Else    If      C_TQ_IGA_INH = 1
        Then    LV_TQ_IGA_ACT = 0
        Else    If      LV_TQ_IGA_ENA = 1  and  LV_TQ_IGA_REQ = 1
                Then    LV_TQ_IGA_ACT = 1
                Else    LV_TQ_IGA_ACT = 0
                Endif
        Endif
Endif
```

Setpoint spark retard calculation (for visual purposes, [C] =[NC_EX_CBK_NR] in the following concerning _CBK_ variables):

```
If          LV_TQ_IGA_ACT = 1
Then    EFF_IGA_INV_COR_CBK[C] = (100 – EFF_IGA_SP_CBK[C]*100) / FAC_EFF_IGA
        EFF_IGA_INV_COR_CBK_CAN[C] = (100 - EFF_IGA_SP_CBK_CAN[C]*100) / FAC_EFF_IGA

        IGA_DIF_SP_CBK_H_RNG[C] = IP_IGA_DIF_SP_H_RNG(EFF_IGA_INV_COR_CBK[C])
        IGA_DIF_SP_CBK_CAN_H_RNG[C] = IP_IGA_DIF_SP_H_RNG (EFF_IGA_INV_COR_CBK_CAN[C])

        IGA_SP_CBK_H_RNG[C] = IGA_REF_COR_CBK[C] + IGA_DIF_SP_CBK_H_RNG[C]
        IGA_SP_CBK_CAN_H_RNG[C] = IGA_REF_COR_CBK[C] + IGA_DIF_SP_CBK_CAN_H_RNG[C]
Else    IGA_SP_CBK_H_RNG[C] = IGA_SP_CBK_CAN_H_RNG[C]   = 60°CRK
Endif
```

Calculation of Arithmetic Mean Values:

IGA_SP_H_RNG = mean(IGA_SP_CBK_H_RNG*[C]*)
IGA_DIF_SP_H_RNG = mean(IGA_DIF_SP_CBK_H_RNG*[C]*)
IGA_SP_CAN_H_RNG = mean(IGA_SP_CBK_CAN_H_RNG*[C]*)
IGA_DIF_SP_CAN_H_RNG = mean(IGA_DIF_SP_CBK_CAN_H_RNG*[C]*)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_EFF_IGA_THD_1	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Threshold ignition efficiency for deactivation spark retard					
C_EFF_IGA_THD_AJ	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Threshold on ignition efficiency for deactivation of spark retard in anti-jerk intervention					
C_EFF_IGA_THD_IS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Threshold on ignition efficiency for deactivation of spark retard in idle speed					
C_TQ_IGA_INH	1	0...2H	0...2	1	[-]
Constant inhibition TQ-intervention by spark retard (0..no inh, 1..Inh, 2..calculated old way)					
IP_IGA_DIF_SP_H_RNG	16	0...B40H	0...-180	-0.0625	[°CRK]
LDP_EFF_IGA_INV_COR_IP_IGA_DIF	16	0...4E20H	0...100	0.005	[%]
Setpoint spark retard for torque intervention					

62.4 Reference and Basic Torque

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQI_REF_IGA	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Indicated engine torque corrected by EFF_IGA_BAS_COR					
TQI_REF_IGA_SCC	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Indicated engine torque corrected by EFF_IGA_BAS_COR and EFF_SCC_BAS					
TQI_REF_LAMB	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Indicated engine torque corrected by EFF_LAMB_BAS					
TQI_REF_SCC	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Indicated engine torque corrected by EFF_SCC_BAS					
TQI_REF_IGA_LAMB	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Indicated engine torque corrected by EFF_IGA_BAS_COR and EFF_LAMB_BAS					
TQI_REF_LAMB_SCC	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Indicated engine torque corrected by EFF_LAMB_BAS and EFF_SCC_BAS					
TQI_REF_IGA_MIN_LAMB	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Indicated engine torque corrected by EFF_IGA_MIN and EFF_LAMB_BAS					
TQI_BAS	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Basic indicated engine torque					
EFF_TOT_BAS	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Total basic efficiency					
EFF_TOT_BAS_SLOW	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Total basic efficiency for feedback to slow torque path					

Input data:

EFF_IGA_BAS_COR	EFF_IGA_BAS_COR_KNK_FIL	EFF_IGA_MIN	EFF_SCC_AV
EFF_SCC_BAS	TQI_REF	EFF_LAMB_AV	EFF_LAMB_BAS_COR
EFF_LAMB_SP_BAS	LV_HOM_RUN		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_MOD_EFF_AV	1	0...1H	0...1	1	-
Logical constant to switch to EFF_LAMB_AV and EFF_SCC_AV calculation mode for TQI_BAS					

62.4.1 TQDR_MD001

General information:

This module delivers the basic indicated engine torque TQI_BAS and other basic torque values for the torque model in homogeneous combustion mode.

Application Condition

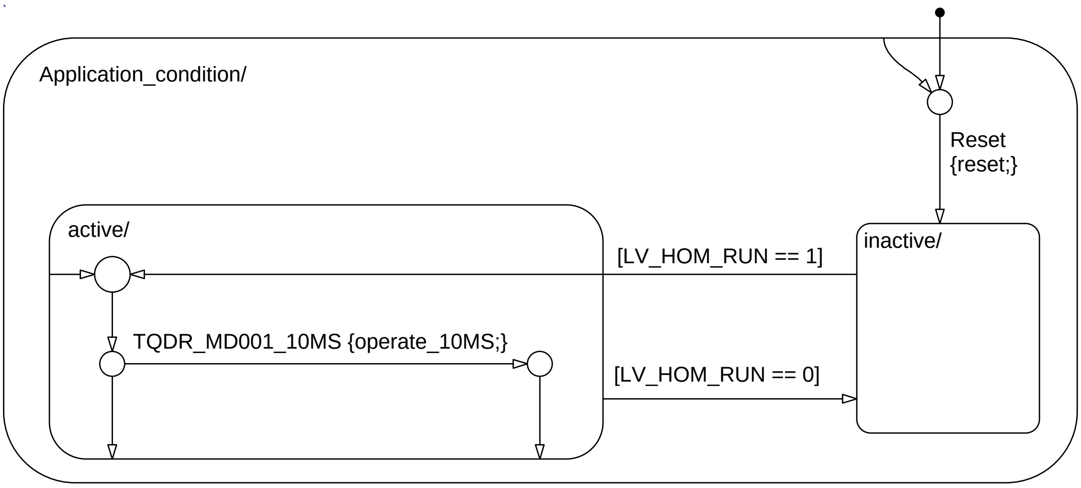
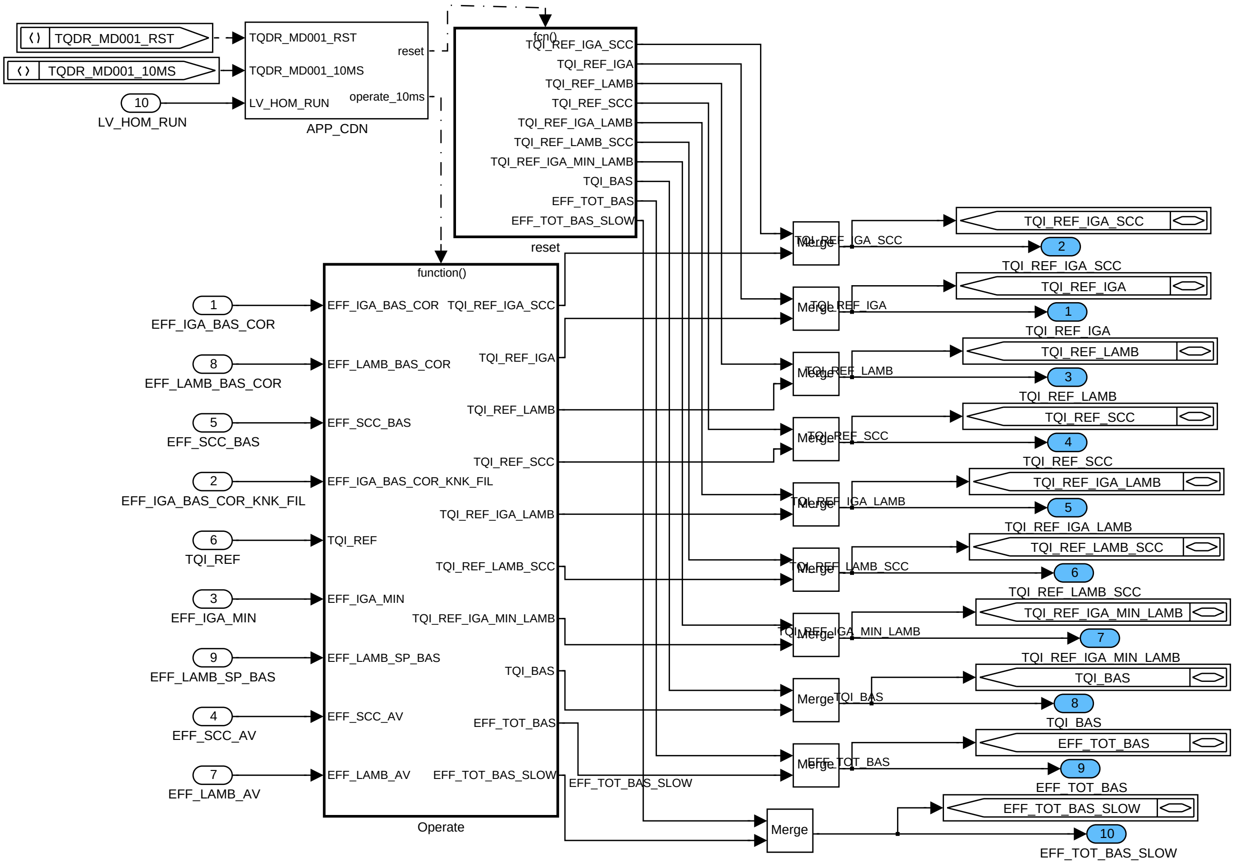


Figure 36 TQDR_MD001/ APP_CDN/ APPCND

Function Description



SDA_SRS / SDA 4.0 23−Jun−2004

Figure 37 TQDR_MD001

62.4.1.1 SUBFUNCTION: Operate

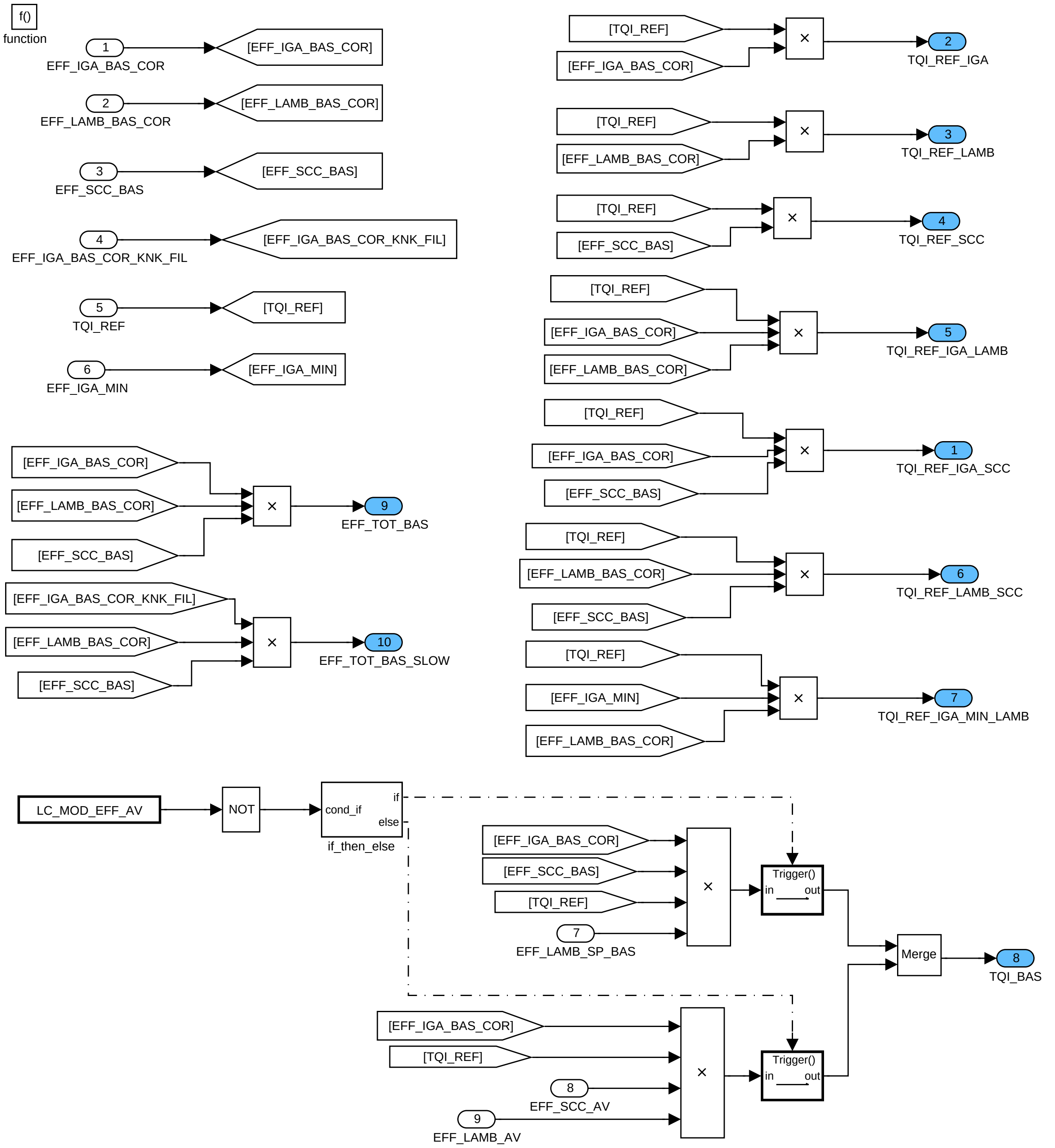


Figure 38 TQDR_MD001/ Operate

62.4.1.2 SUBFUNCTION: reset

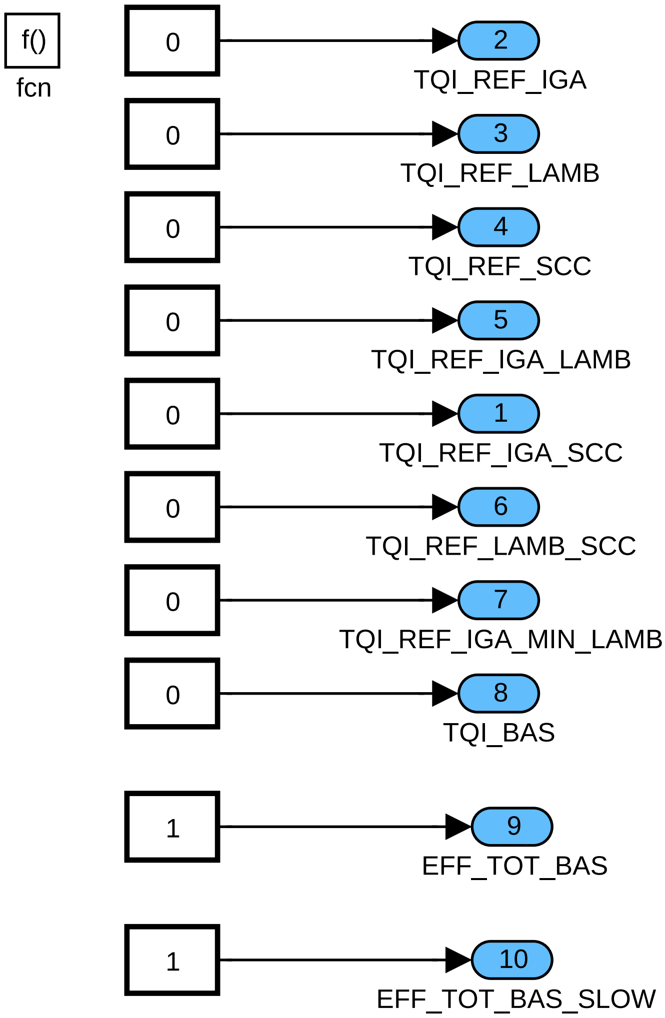


Figure 39 TQDR_MD001/ reset

62.5 Torque based pattern calculation

62.5.1 Pattern calculation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NR_PAT_SCC	V/O	0...FFH	0...255	1	[-]
Selected index of fuel cut-off pattern					
EFF_SCC_SP	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
Setpoint efficiency fuel cut-off pattern					
EFF_SCC_SP_HYS	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Setpoint efficiency fuel cut-off pattern with hysteresis					
LV_AUTH_TQ_PAT	V	0...1H	0...1	1	[-]
logical variable for authorizing fuel cut-off pattern					
FAC_IGA_MIN_TEG_IGA_BAS	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
scaling factor for efficiency calculation with EFF_IGA_MIN_TEG and EFF_IGA_BAS_COR					

Input data:

TQI_REQ_FAST	TQI_REF	EFF_IGA_MIN_TEG	EFF_IGA_BAS_COR
EFF_LAMB_BAS_COR	EFF_SCC_BAS	LV_TQ_GS_SCC	LV_GS_ACT
LV_ASR_ACT	LV_N_MAX	LV_FCUT_FAST	LV_TQ_SCC_INH
LV_TQ_SCC_REQ	LV_TQ_LIM_INTV	LV_TQ_GS_IGA	LV_AMT_ACT
LV_DCC_INC_ACT	LV_DCC_PUC_INH	LV_LDM_ACT	LV_LDM_PUC_INH
LV_INH_FCUT_AMT	LV_HOM_ACT	TQI_MIN_S	LV_GS
	LV_INH_FCUT_GS	LV_VAR_TCT	

FUNCTION DESCRIPTION:

General information:

The objective of this module is to generate a torque intervention by fuel cut-off pattern.
This torque intervention could be caused by several conditions which are summarized in the label LV_AUTH_TQ_PAT (see chapter 1.1.1 "Authorization for single cylinder cut-off"). The label LV_FCUT_FAST can be used for project specific fast fuel cut-off conditions (see module "Torque based pattern calculation (Appl. Inc.)").
Based on the TQI_REQ_FAST and the TQI_REF corrected with the basic a/f ratio efficiency and the ignition angle efficiency an efficiency setpoint for the single cylinder cut-off EFF_SCC_SP is calculated, which leads to the pattern index NR_PAT_SCC (see chapter 1.1.3 "Setpoint efficiency fuel cut-off").
In general, a torque reduction could be realized by spark retarding or single cylinder cut-off. With the scaling factor FAC_IGA_MIN_TEG_IGA_BAS a scaling between earliest cylinder cut-off (no spark retarding) and latest cylinder cut-off (full spark retarding up to IGA_MIN) can be realized (see chapter 1.1.2 "Determination of scaling factor").

Application conditions:

Activation:	at every engine state
Deactivation:	-
Initialization:	EFF_SCC_SP = 1 at reset EFF_SCC_SP_HYS = 1 at reset NR_PAT_SCC = 0 at reset
Update rate:	10 ms

62.5.1.1 Authorization for single cylinder cut-off

General information:

The authorization flag is set depending on:

- a manual enabling => LV_TQ_SCC_ENA_MAN
- a request from traction control (ASR) => LV_ASR_ACT
- a request from gear shift intervention (GS), if gear shift intervention is not explicitly inhibited because gear shift intervention should only work via spark retarding => LV_GS_ACT and LV_TQ_GS_IGA (see module "Torque request for gear shift intervention")
- a request from engine speed limitation (N_MAX) => LV_N_MAX
- other project specific requests for fast fuel cut-off => LV_FCUT_FAST (see module "Torque based pattern calculation (Appl. Inc.)")
- a request from torque management => LV_TQ_SCC_REQ and LV_TQ_SCC_INH
- a request from safety concept => LV_TQ_LIM_INTV (see module "Torque request for safety")

Due to logical calibration constants nearly all of these request can be inhibited, except of the requests from torque management and safety concept.

Formula section:

```
IF      LC_TQ_SCC_ENA_MAN == 1
        OR   LV_TQ_LIM_INTV == 1
        OR   (LV_TQ_SCC_INH == 0
                  AND   LC_TQ_SCC_INH_MAN == 0
                  AND   LV_TQ_SCC_REQ == 1)
        OR   (LV_ASR_ACT == 1 AND LC_INH_FCUT_ASR == 0)
        OR   (LV_N_MAX == 1 AND LC_INH_FCUT_N_MAX == 0)
        OR   [LV_AMT_ACT == 1
              AND (LV_INH_FCUT_AMT = 0 OR LC_INH_FCUT_AMT == 0)]
        OR   [LV_VAR_TCT == 1                                AND
             LV_GS = 1                                       AND
             (LV_INH_FCUT_GS = 0 OR LC_INH_FCUT_TCT == 0)]
        OR   (LV_DCC_INC_ACT == 1and LV_DCC_PUC_INH = 0
                                     AND LC_INH_FCUT_DCC == 0)
        OR   (LV_LDM_ACT == 1and LV_LDM_PUC_INH = 0
                                     AND LC_INH_FCUT_LDM == 0)
        OR   (LV_FCUT_FAST == 1 AND LC_INH_FCUT_FAST == 0)
        OR   (LV_GS_ACT == 1 AND LV_VAR_TCT == 0
             AND LC_INH_FCUT_GS == 0
             AND NOT(LV_TQ_GS_IGA == 1 AND LC_INH_FCUT_GS_IGA == 0))
THEN    LV_AUTH_TQ_PAT = 1
ELSE    LV_AUTH_TQ_PAT = 0
ENDIF
```

62.5.1.2 Determination of scaling factor

General information:

The factor FAC_IGA_MIN_TEG_IGA_BAS is used to scale between the efficiencies EFF_IGA_MIN_TEG (corrected basic minimum ignition angle including spark retard limitation due to exhaust gas temperature) and EFF_IGA_BAS_COR (corrected basic minimum ignition angle without spark retard limitation due to exhaust gas temperature). For a torque intervention due to gear shift, traction control, engine speed limitation and fast fuel cut-off the possibility of deactivating this scaling factor must be realized. So the scaling factor can be set equal to "1", if one of that interventions is active and if this method will be allowed (LC_ENA_FCUT_TOT_... = 1). It is also set equal to "1", if the basic efficiency for fuel cut-off is smaller than "1".

Formula section:

```
IF        EFF_SCC_BAS < 1.0
          OR    (LV_GS_ACT == 1
                AND LC_ENA_FCUT_TOT_GS == 1
                AND LV_TQ_GS_SCC == 1)
          OR    (LV_ASR_ACT == 1 AND LC_ENA_FCUT_TOT_ASR == 1)
          OR    (LV_N_MAX == 1 AND LC_ENA_FCUT_TOT_N_MAX == 1)
          OR    (LV_AMT_ACT== 1 AND LC_ENA_FCUT_TOT_AMT == 1)
          OR    (LV_DCC_INC_ACT== 1 and LV_DCC_PUC_INH ==0
                                        AND LC_ENA_FCUT_TOT_DCC == 1)
          OR    (LV_LDM_ACT== 1 and LV_LDM_PUC_INH ==0
                                        AND LC_ENA_FCUT_TOT_LDM == 1)
          OR    (LV_FCUT_FAST == 1 AND LC_ENA_FCUT_TOT_FAST == 1)
THEN      FAC_IGA_MIN_TEG_IGA_BAS = 1
ELSE      FAC_IGA_MIN_TEG_IGA_BAS = C_FAC_IGA_MIN_TEG_IGA_BAS
ENDIF
```

62.5.1.3 Setpoint efficiency fuel cut-off

62.5.1.3.1 Overview

Description:

Depending on the authorization flag LV_AUTH_TQ_PAT the efficiency EFF_SCC_SP is set to the calculated value EFF_SCC_SP_1 or is set equal to "1". Additionally, two application labels allow to manipulate the efficiency setpoint (LC_EFF_SCC_SP_MAN and C_EFF_SCC_SP_MAN). Depending on the efficiency setpoint a pattern is calculated. To avoid uncomfortable jumps of pattern, if the setpoint EFF_SCC_SP jitters, the calculation of the pattern index NR_PAT_SCC is done by applying a hysteresis C_EFF_SCC_SP_HYS on EFF_SCC_SP (see figure 1).

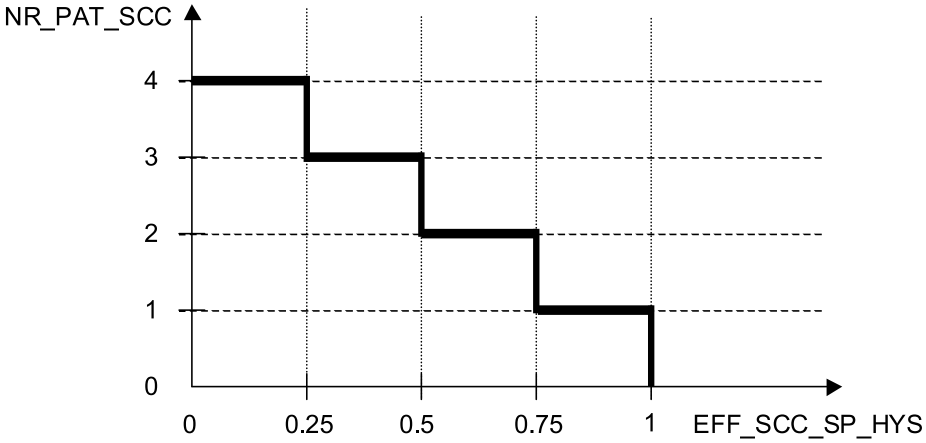


Figure 1: Example for a four cylinder engine

Formula section:

```
IF        LV_AUTH_TQ_PAT == 1
THEN      EFF_SCC_SP_2 = EFF_SCC_SP_1
ELSE      EFF_SCC_SP_2 = 1.0
ENDIF

IF        LC_EFF_SCC_SP_MAN == 1
THEN      EFF_SCC_SP = C_EFF_SCC_SP_MAN
ELSE      EFF_SCC_SP = EFF_SCC_SP_2
ENDIF

IF        (EFF_SCC_SP - EFF_SCC_SP_HYS  >  C_ EFF_SCC_SP_HYS)
          OR    (EFF_SCC_SP  ≥  1 - C_ EFF_SCC_SP_HYS)
          OR    (EFF_SCC_SP  <  EFF_SCC_SP_HYS)
THEN      EFF_SCC_SP_HYS  =  EFF_SCC_SP
```

```
ELSE     EFF_SCC_SP_HYS  =  EFF_SCC_SP_HYS
ENDIF

NR_PAT_SCC  =  ID_NR_PAT_SCC(EFF_SCC_SP_HYS)
```

62.5.1.3.2 Calculation of efficiency for single cylinder cut-off

Description:

The efficiency for single cylinder cut-off depends on the ratio between TQI_REQ_FAST and TQI_REF corrected with the basic efficiencies for a/f ratio and ignition angle. Generally, a scaling between EFF_IGA_MIN_TEG and EFF_IGA_BAS_COR is possible. The philosophy of this function is to give the customer the possibility to work with minimum ignition angle before fuel cut-off or to work with the basic ignition angle.

Formula section:

```
IF       LV_HOM_ACT = 1
THEN     EFF_SCC_SP_1 =  TQI_REQ_FAST /
                          (TQI_REF
                          * EFF_LAMB_BAS_COR
                          * (FAC_IGA_MIN_TEG_IGA_BAS
                          * EFF_IGA_BAS_COR
                          + (1.0 - FAC_IGA_MIN_TEG_IGA_BAS)
                          * EFF_IGA_MIN_TEG_LGRD))
ELSE     EFF_SCC_SP_1 =  TQI_REQ_FAST /
                          TQI_MIN_S
ENDIF


IF       EFF_SCC_SP_1 > 1.0
THEN     EFF_SCC_SP_1 = 1.0
ELSEIF   EFF_SCC_SP_1 < 0.0
THEN     EFF_SCC_SP_1 = 0.0
ENDIF
```

62.5.1.3.3 Rate limitations for negative gradients

Description:

Changes of the efficiency EFF_IGA_MIN_TEG has to be limited in falling direction to avoid high frequent jumps between pattern and spark retarding. Therefore a calibratable rate limit function is added. The principle application parameters for these rate limitations are C_MIN_DEC_EFF_IG_MIN_TEG_1, C_MIN_DEC_EFF_IG_MIN_TEG_2 and C_LGRD_EFF_IGA_MIN_TEG. The background is illustrated in figure 2.

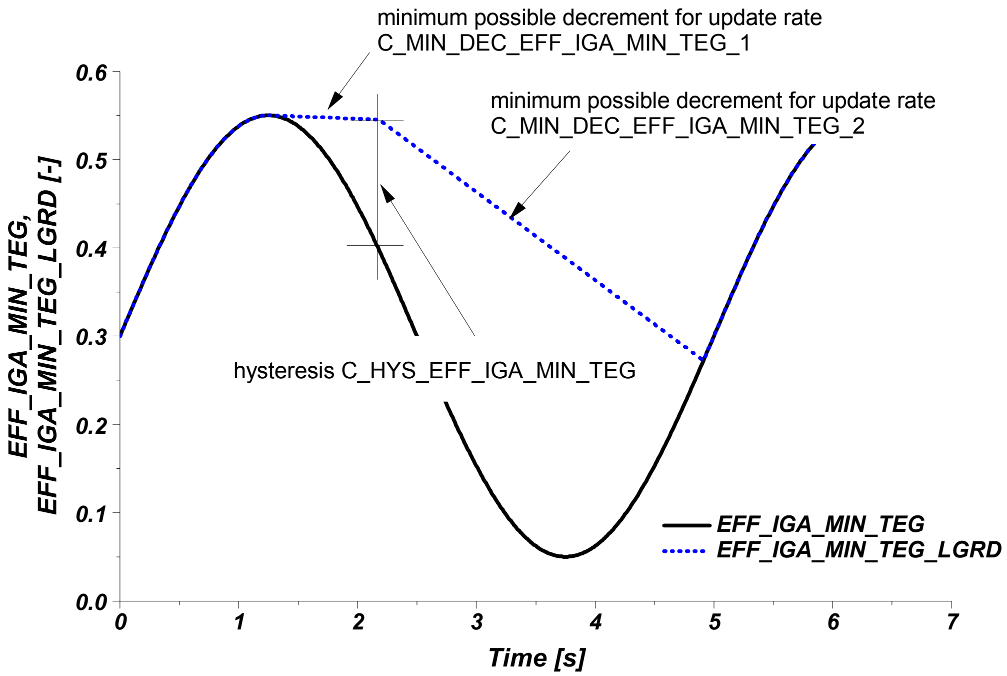


Figure 2: Rate limitations for EFF_IGA_MIN_TEG

Formula section:

```
IF      ($EFF\_IGA\_MIN\_TEG\_LGRD\_1_{k-1} > EFF\_IGA\_MIN\_TEG_k$)
THEN    IF      (($EFF\_IGA\_MIN\_TEG\_LGRD\_1_{k-1} - EFF\_IGA\_MIN\_TEG_k$)
                > $C\_MIN\_DEC\_EFF\_IGA\_MIN\_TEG\_1$)
        THEN    $EFF\_IGA\_MIN\_TEG\_LGRD\_1_k = EFF\_IGA\_MIN\_TEG\_LGRD\_1_{k-1}$
                                        $- C\_MIN\_DEC\_EFF\_IGA\_MIN\_TEG\_1$
        ELSE    $EFF\_IGA\_MIN\_TEG\_LGRD\_1_k = EFF\_IGA\_MIN\_TEG_k$
        ENDIF
ELSE    $EFF\_IGA\_MIN\_TEG\_LGRD\_1_k = EFF\_IGA\_MIN\_TEG_k$
ENDIF
```

```
IF      ((EFF_IGA_MIN_TEG$_{k-1}$ < EFF_IGA_MIN_TEG$_{k}$)
        OR
        (C_HYS_EFF_IGA_MIN_TEG
        <= (EFF_IGA_MIN_TEG_LGRD_1 – EFF_IGA_MIN_TEG)))
THEN    EFF_IGA_MIN_TEG_LGRD_2 = EFF_IGA_MIN_TEG
ELSE    EFF_IGA_MIN_TEG_LGRD_2 = EFF_IGA_MIN_TEG_LGRD_1
ENDIF

IF      (EFF_IGA_MIN_TEG_LGRD$_{k-1}$ > EFF_IGA_MIN_LGRD_2$_{k}$)
THEN    IF      ((EFF_IGA_MIN_TEG_LGRD$_{k-1}$ - EFF_IGA_MIN_LGRD_2$_{k}$)
                > C_MIN_DEC_EFF_IGA_MIN_TEG_2)
        THEN  EFF_IGA_MIN_TEG_LGRD$_{k}$ = EFF_IGA_MIN_TEG_LGRD$_{k-1}$
                                        - C_MIN_DEC_EFF_IGA_MIN_TEG_2
        ELSE  EFF_IGA_MIN_TEG_LGRD$_{k}$ = EFF_IGA_MIN_LGRD_2$_{k}$
        ENDIF
ELSE    EFF_IGA_MIN_TEG_LGRD$_{k}$ = EFF_IGA_MIN_LGRD_2$_{k}$
ENDIF
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_INH_FCUT_FAST	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due fast fuel cut-off					
LC_INH_FCUT_ASR	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due to ASR					
LC_INH_FCUT_N_MAX	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due to speed limitation					
LC_INH_FCUT_AMT	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due to AMT intervention					
LC_INH_FCUT_DCC	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due to DCC intervention					
LC_INH_FCUT_GS	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due to gear shift intervention					
LC_INH_FCUT_GS_IGA	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due to gear shift intervention with spark retarding					
LC_TQ_SCC_INH_MAN	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due to torque management					
LC_TQ_SCC_ENA_MAN	1	0...1H	0...1	1	[-]
flag for enabling SCC by application system					
LC_ENA_FCUT_TOT_FAST	1	0...1H	0...1	1	[-]
logical variable for setting scaling factor due to fast fuel cut-off					
LC_ENA_FCUT_TOT_ASR	1	0...1H	0...1	1	[-]
logical variable for setting scaling factor due to ASR					
LC_ENA_FCUT_TOT_N_MAX	1	0...1H	0...1	1	[-]
logical variable for setting scaling factor due to engine speed limitation					
LC_ENA_FCUT_TOT_AMT	1	0...1H	0...1	1	[-]
logical variable for setting scaling factor due to AMT					
LC_ENA_FCUT_TOT_DCC	1	0...1H	0...1	1	[-]
logical variable for setting scaling factor due to DCC					
LC_ENA_FCUT_TOT_GS	1	0...1H	0...1	1	[-]
logical variable for setting scaling factor due to gear shift intervention					
C_FAC_IGA_MIN_TEG_IGA_BAS	1	0...8000H	0...1	0.0305e-3	[-]
Manually adjusted scaling factor for efficiency calculation with EFF_IGA_MIN_TEG and EFF_IGA_BAS_COR					
C_EFF_SCC_SP_MAN	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
setpoint for efficiency of fuel cut-off set by application intervention					
LC_EFF_SCC_SP_MAN	1	0...1H	0...1	1	[-]
flag for setting EFF_SCC_SP by application system					
C_EFF_SCC_SP_HYS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
hysteresis applied on EFF_SCC_SP for NR_PAT_SCC calculation					
ID_NR_PAT_SCC	13	0...FFH	0...255	1	[-]
LDP_EFF_SCC_SP__NR_PAT_SCC	13	0...FFFFH	0...1.99996	0.0305e-3	[-]
SCC pattern depending on SCC efficiency					
C_HYS_EFF_IGA_MIN_TEG	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
hysteresis in falling direction for IGA_MIN_TEG correction					
C_MIN_DEC_EFF_IGA_MIN_TEG_1	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
minimum decrement for first decrease phase					
C_MIN_DEC_EFF_IGA_MIN_TEG_2	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
minimum decrement for second decrease phase					
LC_INH_FCUT_LDM	1	0...1H	0...1	1	[-]
inhibition flag for fuel cut-off due to LDM intervention					
LC_ENA_FCUT_TOT_LDM	1	0...1H	0...1	1	[-]
logical variable for setting scaling factor due to LDM					
LC_INH_FCUT_TCT	1	0...1H	0...1	1	[-]
Inhibition flag for fuel cut-off due to TCT intervention					

62.5.2 Basic efficiency fuel cut-off

Output data:

Name	V/S	Hex. limits	Phys. limits	Resol.	Unit
EFF_SCC_BAS	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
basic efficiency fuel cut-off					
SUM_INH_IV_BAS	V	0...8H	0...8	1	[-]
Sum of cylinders which are shut off due to malfunctions					

Input data:

INH_IV_MIS	INH_IV_DIAG_ERR	INH_IV_IGC	

General information:

EFF_SCC_BAS is the basic efficiency for single cylinder fuel cut-off. It is calculated by count of opened injectors divided by the number of cylinders (NC_CYL_NR), whereas only injectors which are shut off due to failures (e.g. failure on injector / ignition output stage) are considered.

Activation: at every engine state

Deactivation: -

Initialization: EFF_SCC_BAS = 1.0 at reset and LV_IGK = 0 -> 1

Update rate: 10 ms

Formula section:

SUM_INH_IV_BAS = bitwise summation (Bits 0 to (NC_CYL_NR-1)) of:

INH_IV_MIS **or** INH_IV_DIAG_ERR **or** INH_IV_IGC

Maximal possible value: NC_CYL_NR

```
If      LC_EFF_SCC_BAS_SWI_CLC = 1
Then    If      LC_ENA_EFF_SCC_BAS_MAN = 1
        Then    EFF_SCC_BAS = C_EFF_SCC_BAS_MAN
        Else    EFF_SCC_BAS = 1.0
        Endif
Else    EFF_SCC_BAS = 1- (SUM_INH_IV_BAS / NC_CYL_NR)
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_ENA_EFF_SCC_BAS_MAN	1	0...1H	0...1	1	[-]
logical variable for manually adjusting basic efficiency fuel cut-off					
LC_EFF_SCC_BAS_SWI_CLC	1	0...1H	0...1	1	[-]
logical variable for deactivation of basic efficiency fuel cut-off calculation through missfire pattern					
C_EFF_SCC_BAS_MAN	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
manually adjusted basic efficiency fuel cut-off					

62.5.3 Actual efficiency fuel cut-off

Output data:

Name	V/S	Hex. limits	Phys. limits	Resol.	Unit
EFF_SCC_AV	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
Actual efficiency fuel cut-off					
EFF_SCC_AV_CAN	V/O	0...FFFFH	0...1.99996	0.0305e-3	[-]
Actual efficiency fuel cut-off					
SUM_INH_IV_TOT	V	0...8H	0...8	1	[-]
Sum of cylinders which are shut off either by means of static or dynamic fuel cut-off pattern					
INH_IV_TOT	V	0...FFFFH	0...65535	1	[-]
Overall shut off pattern (static and dynamic)					

Input data:

LV_LDM_ACT	LV_GS_ACT	LV_AMT_ACT	LV_DCC_INC_ACT
LV_DCC_PUC_INH	LV_ASR_ACT	NC_CYL_NR	LV_LDM_PUC_INH
INH_IV	INH_SWI_IV		

FUNCTION DESCRIPTION:

General information:

EFF_SCC_AV is the actual efficiency from fuel cut-off interventions. The value is calculated by count of opened injectors divided by the number of cylinders (NC_CYL_NR). The input value INH_IV defines the status of each injector during the next injection cycle due to static fuel cut-off. The input value INH_SWI_IV defines the status of all injectors during the next injection cycle due to dynamic fuel cut-off.

Further information see in chapter “Basic SW inputs and Outputs” in subchapter “Injection information”.

Every static fuel cut-off request intervention deriving from pattern (engine speed limitation, ASR and GS), trailing throttle fuel cut-off, misfire, engine stop, etc. is considered by means of INH_IV.

Every dynamic fuel cut-off request intervention deriving from "torque based pattern"- and/or "sequential fuel cut off and restart fuel feed"-interventions is considered by means of INH_SWI_IV.

The pattern INH_IV_TOT is the bitwise OR of all pattern (static and dynamic) and SUM_INH_IV_TOT is the corresponding sum of cylinders which are shut off.

The EFF_SCC_AV_CAN is calculated like EFF_SCC_AV only at external intervention the efficiency is not regarded.

Activation:	at every engine state except engine stopped	
Deactivation:	at engine stopped	
Initialization:	LV_EFF_SCC_AV_AUTH = 0	at reset or at engine stop
Update rate:	10 ms	

Formula section:

```
If      INH_IV = 1              (bitwise)       or
        INH_SWI_IV = 1          (bitwise)
Then    INH_IV_TOT = 1          (bitwise)
Else    INH_IV_TOT = 0          (bitwise)
Endif
```

SUM_INH_IV_TOT = bitwise summation of INH_IV_TOT (Bits 0 to (NC_CYL_NR-1))

Maximal possible value: NC_CYL_NR

EFF_SCC_AV_CLC = 1- (SUM_INH_IV_TOT / NC_CYL_NR)

```
If      EFF_SCC_AV_CLC < 1              and
        LV_EFF_SCC_AV_AUTH = 0
Then    EFF_SCC_AV = 1
Else    EFF_SCC_AV = EFF_SCC_AV_CLC
        LV_EFF_SCC_AV_AUTH = 1
Endif
```

Calculation of EFF_SCC_AV_CAN for CAN Torque:

```
If      LV_GS_ACT = 1           or
        LV_AMT_ACT = 1          or
        LV_ASR_ACT = 1          or
        (LV_DCC_INC_ACT = 1     and
        LV_DCC_PUC_INH = 0)     or
        (LV_LDM_ACT = 1         and
        LV_LDM_PUC_INH = 0)
Then    EFF_SCC_AV_CAN = 1
Else    EFF_SCC_AV _CAN = EFF_SCC_AV
Endif
```

62.6 Application incidences for torque based pattern calculation

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_CS_CHG	V	0... 1H	0... 1	1	[-]
Logical variable for detection of clutch switch state change					
LV_CT_CHG	V	0... 1H	0... 1	1	[-]
Logical variable for detection of accelerator pedal released state change					
LV_FCUT_FAST	O/V	0... 1H	0... 1	1	[-]
Logical variable for authorizing fuel cut-off pattern					
T_MAX_FCUT_FAST	V	0... FFH	0... 2.55	0.01	[s]
Time counter for clutch switch observation fast fuel cut-off					
T_MAX_FCUT_FAST_TCHA	V	0... FFH	0... 2.55	0.01	[s]
Time counter for gear shift observation fast fuel cut-off turbo charger					

Input Data:

LV_CS	LV_CT	LV_PL	LV_PU
LV_VAR_AMT	MAF	MAF_MES	MAF_MMV
N	TCO	VS	LV_VAR_TCHA
LV_GS_ACT_FL	LV_VAR_TCT		

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_MAF_DIF_MIN_FCUT_FAST	1	0... FFFFH	0... 1389	0.0211948	[mg/stk]
min. load decrease fast fuel cut-off					
C_T_MAX_FCUT_FAST	1	0... FFH	0... 2.55	0.01	[s]
max. time clutch switch observation fast fuel cut-off					
C_T_MAX_FCUT_FAST_TCHA	1	0... FFH	0... 2.55	0.01	[s]
max. time gear shift observation fast fuel cut-off turbo charger					
C_TCO_MIN_FCUT_FAST	1	0... FEH	-48... 142.5	0.75	[°C]
min. coolant temperature fast fuel cut-off					
C_VS_MIN_FCUT_FAST	1	0... FFH	0... 255	1	[km/h]
min veh. speed fuel cut-off					
C_VS_MIN_FCUT_FAST_TCHA	1	0... FFH	0... 255	1	[km/h]
min veh. speed fuel cut-off turbo charger					
ID_MAF_N_MIN_FCUT_FAST	4*4	0... 1H	0... 1	1	[-]
LDP_MAF__MIN_FCUT_FAST	4	0... FFFFH	0... 1389	0.0211948	[mg/stk]
LDP_N__MIN_FCUT_FAST	4	0... 1FE0H	0... 8160	1	[rpm]
engine speed -load threshold for fast fuel cut-off					

General Information

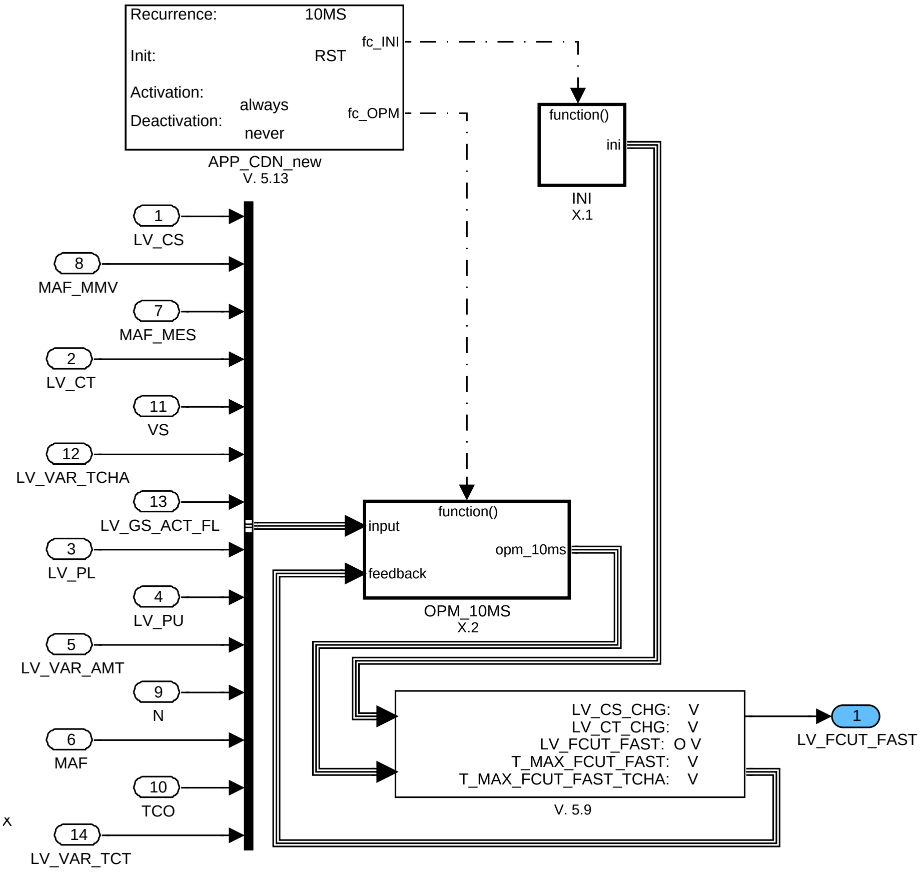
General information:

The objective of this module is to define the application incidences for the module "Torque based pattern calculation" including determination of Fast fuel cut off -condition .

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.2.b.1 29−Aug−2006

Figure 40:

62.6.1 Initialisation at reset

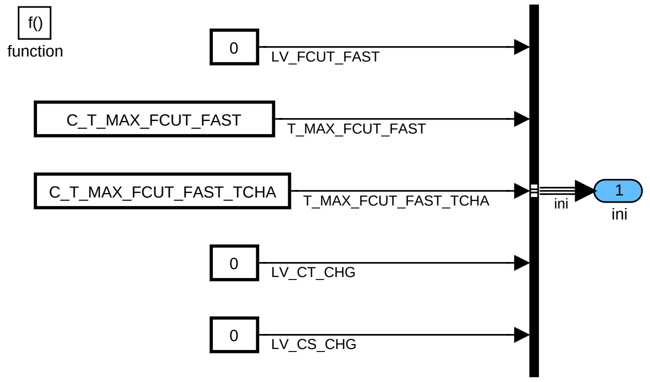


Figure 41:

62.6.2 Recurrence 10 ms

Determination of fast fuel cut-off conditions
General information:

The objective of this module is to define the condition for fast fuel cut-off. This function shall prevent engine speed overshooting during gear shift with manual transmission vehicles. If fast fuel cut-off is enabled, cylinder deactivation will be managed by the torque management.

Description:

The fast fuel cut-off will be enabled for the following situation:
the coolant temperature must exceed a calibratable threshold (see activation condidtions) and the load must decrease
the accelerator pedal must be released and the clutch has to be opened; these conditions have to be happen within a maximum time frame
if the clutch pedal has been pushed during this observation time and if it is still pushed the fast fuel cut-off is possible
the variant turbo regards the flag LV_GS_ACT_FL in order to set fast fuel cut-off

62.6.2.1 Activation conditions for calculation

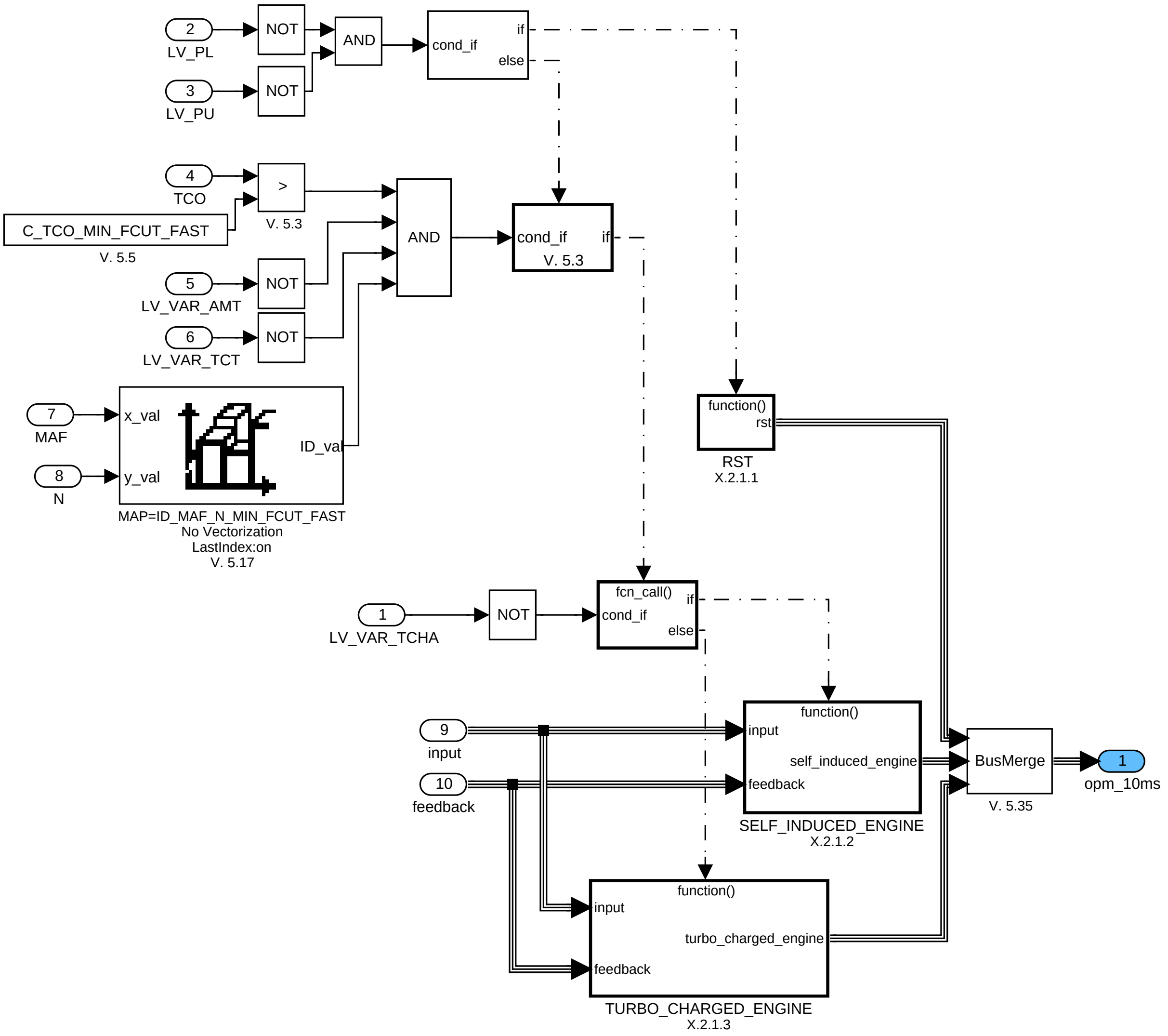


Figure 42:

62.6.2.1.1 Reset at LV_PL = 0 and LV_PU = 0

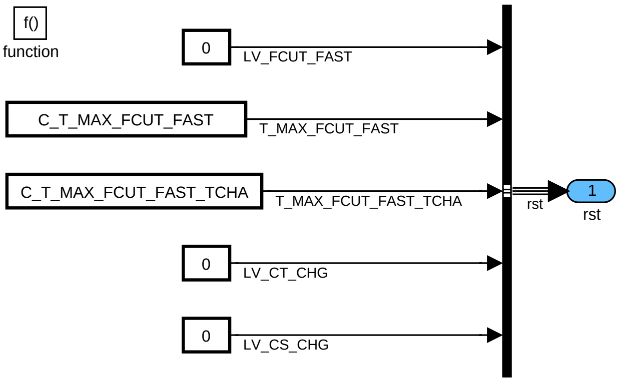


Figure 43:

62.6.2.1.2 Self induced engine

62.6.2.1.2.1 Calculation of self induced engine

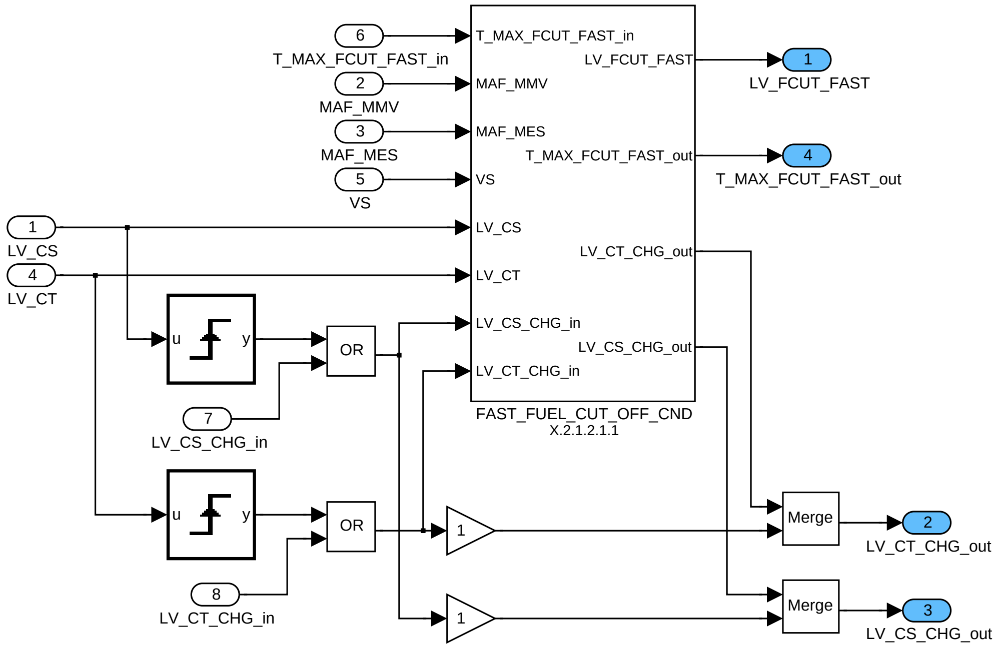


Figure 44:

62.6.2.1.2.1.1 Determination of fast fuel cut off conditions

62.6.2.1.2.1.1.1 Determination_of_fast_fuel_cut_off_conditions

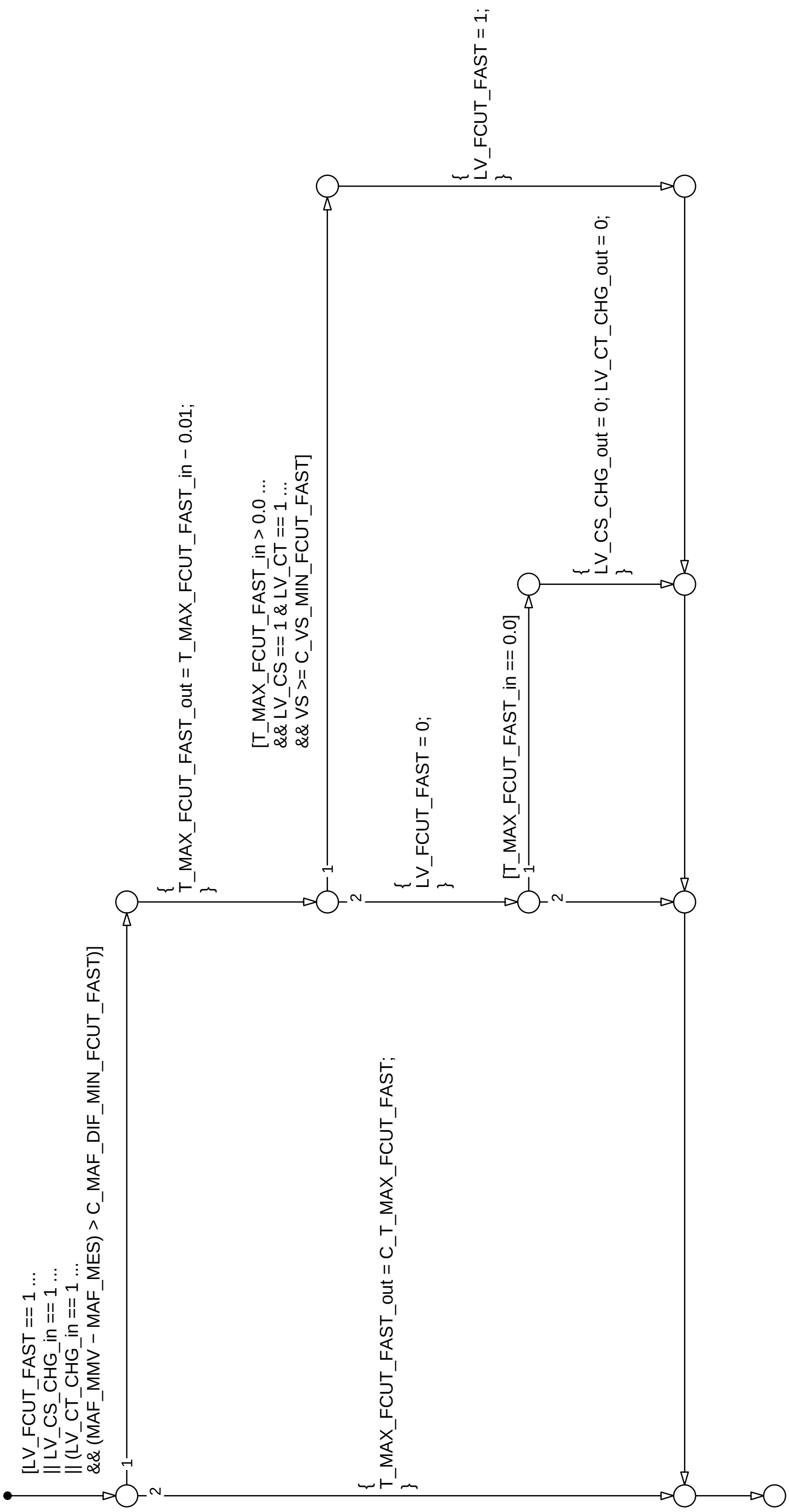


Figure 45:

62.6.2.1.3 Turbo charged engine

62.6.2.1.3.1 Calculation of turbo charged engine

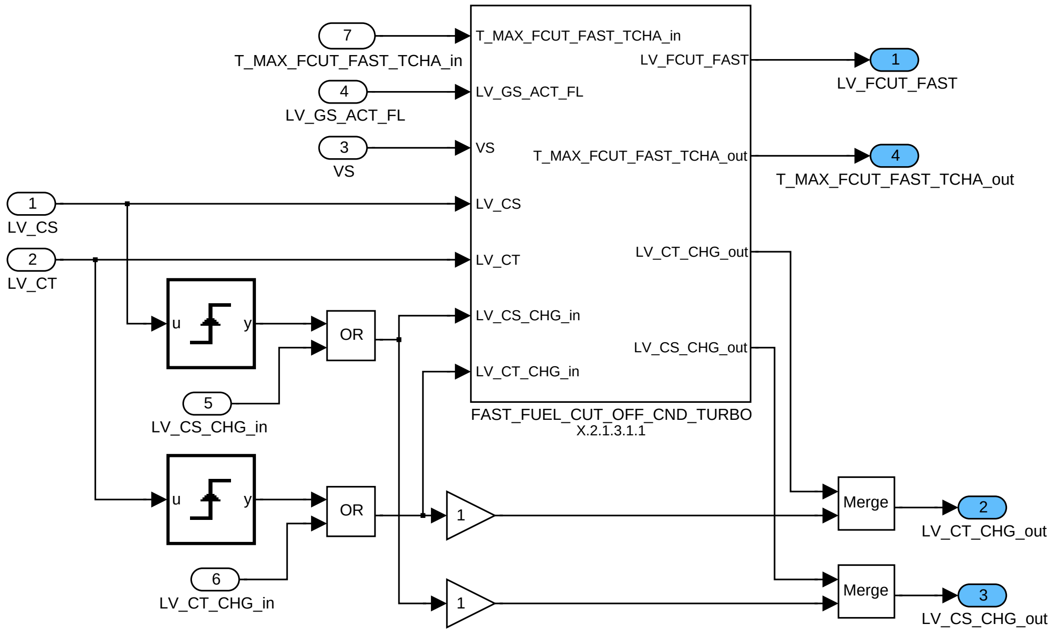


Figure 46:

62.6.2.1.3.1.1 Fast fuel cut off conditions turbo

62.6.2.1.3.1.1.1 fast_fuel_cut_off_conditions_turbo

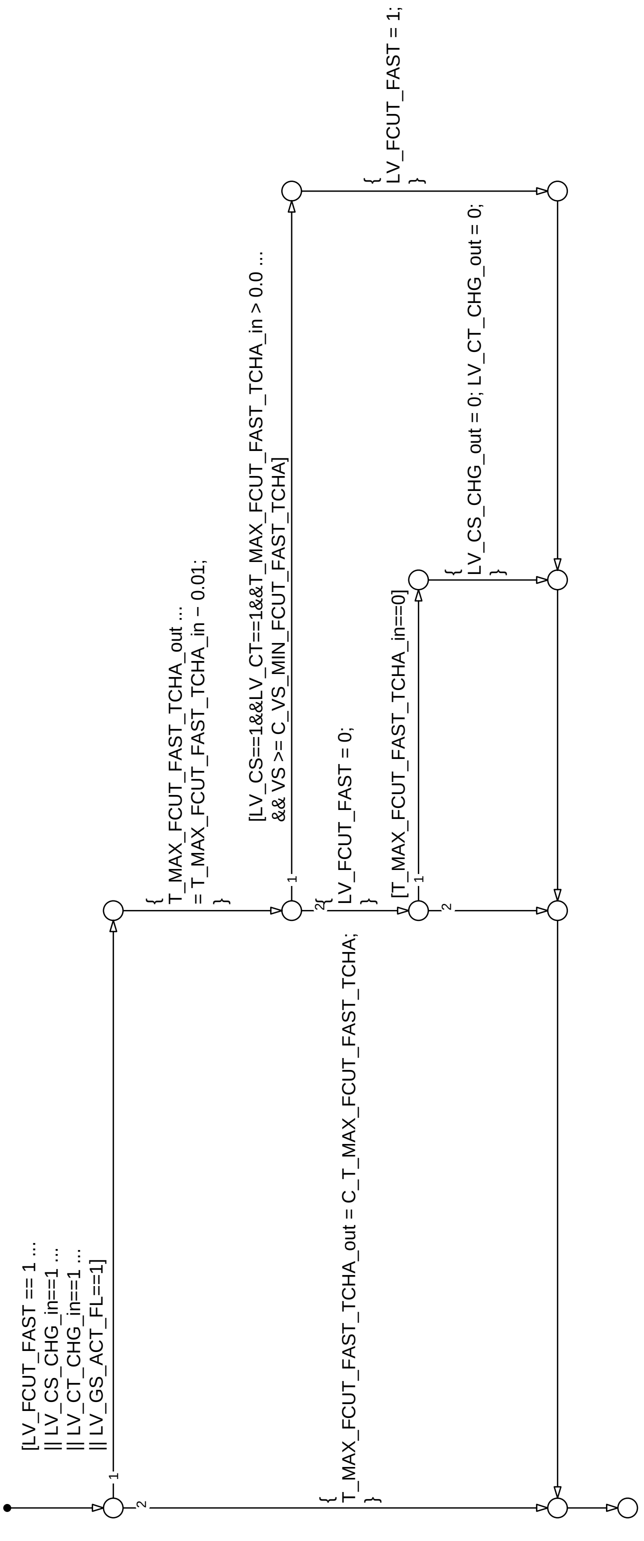


Figure 47:

62.7 Initialization Module

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>ACP</td><td>V/O</td><td>0 ... FFH</td><td>0...127.5</td><td>0.5</td><td>bar</td></tr>
<tr><td colspan="6">AC-fluid presssure</td></tr>
<tr><td>ACP_MMV</td><td>V/O</td><td>0 ... FFH</td><td>0...127.5</td><td>0.5</td><td>bar</td></tr>
<tr><td colspan="6">AC-fluid presssure</td></tr>
<tr><td>TACE</td><td>V/O</td><td>0 ... FEH</td><td>-48...142.5</td><td>0,75</td><td>°C</td></tr>
<tr><td colspan="6">ACC LOW pressure side temperature</td></tr>
<tr><td>EFF_EGR_S</td><td>V/O</td><td>0 ... FFFFH</td><td>0 ... 1.999969</td><td>30.517E-6</td><td>-</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>EFF_PHA_S</td><td>V/O</td><td>0 ... FFFFH</td><td>0 ... 1.999969</td><td>30.517E-6</td><td>-</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>LAMB_BAS_COR_MAX</td><td>V/O</td><td>0 ... 7FFFH</td><td>0 ... 31,9990234</td><td>0,00098</td><td>-</td></tr>
<tr><td colspan="6">Maximum value basic lambda value</td></tr>
<tr><td>LV_ACP_ORNG</td><td>V/O</td><td>0 ... 1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>LV_ERR_ACP</td><td>V/O</td><td>0 ... 1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>LV_CAN_GS_ACK</td><td>V/O</td><td>0 ... 1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>LV_TQ_SCC_INH</td><td>V/O</td><td>0 ... 1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>LV_TQ_SCC_REQ</td><td>V/O</td><td>0 ... 1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>TQ_AD_ADD_S_SWI</td><td>V/O</td><td>8000 ... 7FFFH</td><td>-1024 .. 1023,97</td><td>0.03125</td><td>Nm</td></tr>
<tr><td colspan="6">Adaptive engine torque value for switch between hom. and strat.</td></tr>
<tr><td>TQ_LOSS_SCHA</td><td>V/O</td><td>8000 ... 0H</td><td>-1024 .. 0</td><td>0.03125</td><td>Nm</td></tr>
<tr><td colspan="6">Torque losses super-charger</td></tr>
<tr><td>TQ_LOSS_THERMO</td><td>V/O</td><td>8000 ... 0H</td><td>-1024 .. 0</td><td>0.03125</td><td>Nm</td></tr>
<tr><td colspan="6">Engine torque losses thermostat</td></tr>
<tr><td>TQ_LOSS_WIN_HEAT</td><td>V/O</td><td>8000 ... 0H</td><td>-1024 .. 0</td><td>0.03125</td><td>Nm</td></tr>
<tr><td colspan="6">Engine torque losses rear window heater</td></tr>
<tr><td>TQ_LOSS_CWP_EL</td><td>V/O</td><td>8000 ... 0H</td><td>-1024 .. 0</td><td>0.03125</td><td>Nm</td></tr>
<tr><td colspan="6">Engine torque losses electric water pump</td></tr>
<tr><td>TQI_CAT_PROT</td><td>V/O</td><td>0 ... 7FFFH</td><td>0 .. 1023.97</td><td>0.03125</td><td>Nm</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>TQI_VS_MAX</td><td>V/O</td><td>0 ... 7FFFH</td><td>0 .. 1023.97</td><td>0.03125</td><td>Nm</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>TQ_ADD_REQ_ACT_FIL_CH</td><td>V/O</td><td>8000 ... 7FFFH</td><td>-1024 .. 1023,97</td><td>0.03125</td><td>Nm</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>LV_TQ_GS_IGA</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6">Torque intervention due gear shift about IGA</td></tr>
<tr><td>LV_TQ_GS_SCC</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6">Torque intervention due gear shift about SCC</td></tr>
<tr><td>LV_GS_ACT</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6">Gs intervention active</td></tr>
<tr><td>LV_AMT_ACT</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6">AMT intervention active</td></tr>
<tr><td>LV_IGA_CH_SO2P_STB</td><td>V/O</td><td>0...1H</td><td>0...1</td><td>1</td><td>-</td></tr>
<tr><td colspan="6"></td></tr>
</table>

Input data:

LV_S_RUN	LV_GS_DEC_ACT	LV_AMT_DEC_ACT	LV_AMT_INC_ACT

FUNCTION DESCRIPTION:

General information:

The initialization module is a temporary software module which is gradually reduced in that amount how functionallity of the MS45CP software is growing.
It generates output data (torque values, state bits and other values) as substitute input values for existing SW modules. The source modules will be realized in future.

Application conditions:

Recurrence:
Activation: at every engine operating state
Initialisation: at reset

ACP = ACP_MMV = 0 bar
TACE = 142,5 °C
EFF_EGR_S = 1
EFF_PHA_S = 1
LAMB_BAS_COR_MAX = 2
LV_ACP_ORNG = 1
LV_CAN_GS_ACK = 0
LV_TQ_FAST_ENA = 1
LV_TQ_IGA_ENA = 1
LV_TQ_SCC_INH = 0
LV_TQ_SCC_ENA = 0
LV_ERR_ACP = 0
TQ_AD_ADD_S_SWI = C_TQ_AD_ADD_S_SWI
TQ_LOSS_SCHA = 0 Nm
TQ_LOSS_THERMO = 0 Nm
TQ_LOSS_WIN_HEAT = 0 Nm
TQ_LOSS_CWP_EL = 0 Nm
TQI_CAT_PROT = 1024 Nm
TQI_VS_MAX = 1024 Nm
LV_TQ_GS_IGA = LV_TQ_GS_SCC = LV_GS_ACT
LV_GS_ACT = LV_GS_DEC_ACT
LV_AMT_ACT = LV_AMT_INC_ACT or LV_AMT_DEC_ACT
LV_IGA_CH_SO2P_STB =0

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TQ_AD_ADD_S_SWI	1	8000 ... 7FFFH	-1024 ... 1023,97	0,03125	Nm
Adaptive engine torque value for switch between hom. and strat.					

62.8 Actual Engine Torque

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
EFF_TOT_AV	V	0 ... FFFF H	0 ... 1.999969	30.518E-6	-
Total actual efficiency (in homogeneous mode)					
TQ_AV	V / O	8000 ... 7FFF H	-1024 ... 1023.97	0.03125	Nm
Actual engine torque static at clutch					
TQE	V / O	8000 ... 7FFF H	-1024 ... 1023.97	0.03125	Nm
Effective engine torque at clutch					
TQE_DIF	V	8000 ... 7FFF H	-1024 ... 1023.97	0.03125	Nm
Difference effective engine torque at clutch					

Input data:

TQI_REF	EFF_IGA_AV	LV_S_ACT	N_GRD
TQI_AV_S	EFF_SCC_AV	LV_HOM_RUN	
TQ_LOSS	EFF_LAMB_AV	TQI_AV	TQI_AV_HOM

FUNCTION DESCRIPTION:

General information:

TQI_AV is the actual indicated engine torque produces either in homogeneous or stratified combustion mode. The combustion mode is indicated by the logical variable LV_S_ACT (1 ... stratified combustion; 0 ... homogeneous combustion). TQ_AV is the actual static effective torque at clutch. TQE is the actual effective torque at clutch inclusive inertia.

These values are used for external units (e.g. traction control and gear box) as a feedback signal.

Application conditions:

Activation: at every engine state

Deactivation: -

Initialization at reset

TQ_AV = TQE = TQE_DIF = 0 Nm

Update rate: 10 ms

Formula section:

EFF_TOT_AV = EFF_IGA_AV * EFF_SCC_AV * EFF_LAMB_AV

TQ_AV = TQI_AV + TQ_LOSS

TQE_DIF = C_FAC_TQE * N_GRD

TQE = TQ_AV - TQE_DIF

The dynamic torque is derived as follows:

TQE_DIF = J (kg*m^2) * $d\omega/dt$ (1/s^2)

TQE_DIF = J (kg*m^2) * $\pi/30 *$ N_GRD (1/min/s)

TQE_DIF = C_FAC_TQE (kg*m^2*min) * N_GRD (1/min/s)

C_FAC_TQE = J (kg*m^2) * $\pi/30$

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_TQE	1	0...FF h	0...0.249	97.647E-5	kg*m^2/s
Effective engine torque at clutch					

62.9 Customer adaptation module: AGGR TQDR

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_safast	O/V	0...1H	0...1	1	[-]
Bedingung schnelles Schubabschalten					
EFF_LAMB_AV	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Actual lambda efficiency					
EFF_LAMB_AV_CBK[NC_CBK_EX_NR]	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Actual lambda efficiency, bank specific					
EFF_LAMB_BAS_COR	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Basic lambda efficiency					
EFF_LAMB_SP_BAS	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Basic lambda setpoint efficiency					
Eta_zw_ist1	O/V	0...7FFFH	0...99.99694	3.0518e-3	[%]
Ignition efficiency actual ignition angle bank selective					
Eta_zw_ist2	O/V	0...7FFFH	0...99.99694	3.0518e-3	[%]
Ignition efficiency actual ignition angle bank selective					
Eta_zw_min	O/V	0...7FFFH	0...99.99694	3.0518e-3	[%]
Ignition efficiency minimum ignition angle					
Eta_zw_mint	O/V	0...7FFFH	0...99.99694	3.0518e-3	[%]
Ignition efficiency					
Eta_zwve_m	O/V	0...7FFFH	0...99.99694	3.0518e-3	[%]
Mean value ignition angle efficiency					
Eta_zylausbas	O/V	0...FFFFH	0...199.99694	3.0518e-3	[%]
Basic efficiency fuel cut-off pattern					
F_wnwkhk_sw	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
Überblendfaktor (warm/kalt) für Spreizung und Zündhaken während Katheizen von Zulieferer					
FAC_CAM_CH	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for weighting FAC_EFF_IGA at Catalyst heating					
FAC_EFF_IGA_CH	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for fading ignition angle efficiency in case of catalyst heating					
FAC_EFF_IGA_SCAV	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for fading ignition angle efficiency in case of scavenging operation					
FAC_EFF_IGA_SECU	O/V	0...FFH	0...0.99609	3.9063e-3	[-]
Factor for fading ignition angle efficiency in case of cold/warm engine					
LV_GS_ACT_FL	O/V	0...1H	0...1	1	[-]
Logical bit which indicates gear shift at full load condition					
LV_TQ_IGA_ENA	O/V	0...1H	0...1	1	[-]
Torque intervention for ignition released					
LV_TQ_MIN_CLU	O/V	0...1H	0...1	1	[-]
Logical variable for coordination engine start torque and minimum torque and clutch					
LV_TQI_BOL_SET	O/V	0...1H	0...1	1	[-]
Logical variable it is set if TQI_SP falls below TQI_BOL					
Mdi_bas	O/V	D8F0...2710H	-1000...1000	0.1	[Nm]
Basic indicated engine torque					
Mdi_katschutz	O/V	D8F0...2710H	-1000...1000	0.1	[Nm]
torque limitation for catalyst protection					
Mdi_vmax	O/V	D8F0...2710H	-1000...1000	0.1	[Nm]
torque limitation for vehicle speed limitation					
Mdk_ist	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
minimum torque at clutch					
Mdk_ist_lm	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Ist-Moment aus Luftmasse ohne externe Eingriffe					
Mdk_max_gs	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Begrenzung maximales Kupplungsmoment durch EGS-Notlauf					
TQ_MAX_CLU	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Maximum torque at clutch					
TQ_MIN_CLU	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Minimum torque at clutch					
TQI_AV_HOM	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Actual indicated engine torque for homogeneous mode					
TQI_AV_S	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Actual indicated engine torque in stratified mode					
TQI_MAX	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Maximum indicated torque					
TQI_MIN_HOMS	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Min Reference indicated engine torque for HOMS					
TQI_MIN_S	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Min Reference indicated engine torque for stratified mode					
TQI_REF	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Reference indicated engine torque					
TQI_REF_MAX	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Maximum indicated torque from IP_TQI_REF at MAF_MAX_COR					
TQI_REF_S	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Indicated reference engine torque for stratified mode					
TQI_SP_MAF	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Indicated engine torque for determination of air mass flow setpoint in homogeneous mode					

Input data:

B_llrein	B_mdimin_sa	B_vlschalt	B_zw_dynman
ECU_STATE	EFF_IGA_AV_CBK[NC_CBK_EX_NR]	EFF_IGA_BAS_COR_KNK_FIL	EFF_IGA_MIN
EFF_IGA_MIN	EFF_IGA_MIN_TEG	EFF_SCC_BAS	Eta_labas
Eta_labas_sol	Eta_lambda_1	Eta_lambda_2	Eta_zw_minres
F_atldyn	F_sicher_zw	F_wnwkh	F_wnwkhk
FAC_EFF_IGA_CH_SP	LV_FCUT_FAST	LV_TQ_LIM_INTV	Mdi_ist_m
Mdi_max	Mdi_max_l1	Mdi_min_hs	Mdi_min_s
Mdi_ml_soll	Mdi_opt_l1	Mdi_opt_s	Mdi_s_istm
Mdk_max	Mdk_min	NC_CBK_EX_NR	STATE_TCT_INTV
TQ_AV	TQ_GS_SLOW_BN		
	TQ_MAF	TQI_BAS	TQI_CAT_PROT

62.9.1 Outputs for BMW functions which are defined as TQDR exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

Initialisation at reset or at exit power latch phase:

Mdi_bas, B_safast , F_wnwkhk_sw, Mdk_ist_lm = 0
Mdi_vmax = 1000 Nm
Eta_zylausbas,Eta_zw_min = 100%
Mdk_ist : first calculation
Mdk_max_gs = +3276.7 Nm

Recurrence:	10 ms:	Mdi_bas, Mdi_vmax, Mdk_ist, Mdk_ist_lm, Eta_zw_min, Eta_zylausbas, F_wnwkhk_sw, Mdk_max_gs
	20ms:	B_safast

Activation: every engine state

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

Mdi_vmax = TQI_VS_MAX

```
If(1)       ECU_STATE = "PWL"
Then(1)
    B_safast        = 0
    Eta_zw_min      = 0 %
    Mdi_bas         = 0
    Eta_zylaus_bas  = 100%
    Mdk_ist_lm      = 0 Nm
    Mdk_max_gs      = +3276.7 Nm

Elseif(1)
    B_safast        =   LV_FCUT_FAST
    Eta_zw_min      =   EFF_IGA_MIN * 100%
    Mdi_bas         =   TQI_BAS
    Mdk_ist         =   TQ_AV
    Eta_zylausbas   =   EFF_SCC_BAS * 100%
    F_wnwkhk_sw     =   FAC_EFF_IGA_CH_SP
    Mdk_ist_lm      =   TQ_MAF
    If (2)   STATE_TCT_INTV = "TQ_DEC"   /* BMW: ST_MOD_DKG */
    Then(2)         Mdk_max_gs = TQ_GS_SLOW_BN
                        /* CAN value TORQ_TAR_ADJR_POS_EGS */
    Else(2)
        If (3)      STATE_TCT_INTV = " NO_INTERVENTION"
```

```
                or      STATE_TCT_INTV = ″INVALID"
        Then(3)         Mdk_max_gs = +3276.7 Nm

        Endif(3)

    Endif(2)

Endif(1)
```

62.9.2 Outputs for BMW functions which are not defined as TQDR exported data

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or exit Power Latch Phase:

Eta_zw_ist1 = Eta_zw_ist2 =

Eta_zwve_m = Eta_zw_mint = 100 %

Mdi_katschutz = 1000 Nm

Recurrence : 10 ms , except Eta_zwve_m segment

Activation: every engine state

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

Eta_zw_ist1 = EFF_IGA_AV_CBK[1] * 100%
Eta_zw_ist2 = EFF_IGA_AV_CBK[2] * 100%
Eta_zw_mint = EFF_IGA_MIN_TEG * 100%

```
If    ECU_STATE = "PWL"
Then
      Eta_zwve_m        =     0%
      Mdi_katschutz     =     1000 Nm
Elseif
      Eta_zwve_m        =     EFF_IGA_BAS_COR_KNK_FIL * 100%
      Mdi_katschutz     =     TQI_CAT_PROT
Endif
```

62.9.3 Outputs for SV aggregates, BMW→SV

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset:

0.0: TQI_AV_HOM ,TQI_MIN_S , TQI_MIN_HOMS,TQI_REF_MAX ,TQI_SP_MAF, TQI_REF, TQI_AV, TQI_AV_S , TQI_REF_S , FAC_EFF_IGA_CH, TQ_MAX_CLU, FAC_EFF_IGA_SECU, TQ_MIN_CLU, FAC_EFF_IGA_SCAV

0: LV_TQ_IGA_ENA , LV_TQI_BOL_SET, LV_GS_ACT_FL , LV_TQ_MIN_CLU

1.0: EFF_LAMB_AV, EFF_LAMB_AV_CBK[NC_CBK_EX_NR], EFF_LAMB_BAS_COR , EFF_LAMB_SP_BAS

1023.96875 Nm: TQI_MAX

Recurrence : 10 ms TQI_REF, TQI_REF_MAX, TQI_SP_MAX, TQI_AV_HOM, TQI_REF_S, LV_TQ_IGA_ENA, TQI_MIN_HOMS EFF_LAMB_AV, EFF_LAMB_BAS_COR, TQI_AV_S, EFF_LAMB_SP_BAS, FAC_CAM_CH, TQ_MAX_CLU, TQI_MIN_S , EFF_LAMB_AV_CBK[NC_CBK_EX_NR], LV_TQI_BOL_SET, LV_GS_ACT_FL , FAC_EFF_IGA_SECU, LV_TQ_MIN_CLU, TQ_MIN_CLU, FAC_EFF_IGA_SCAV, TQI_MAX

bgn FAC_EFF_IGA_CH

Activation: at every engine state

Deactivation: -

Formula section:

Remark: all formulas are valid in a **physical** meaning

TQI_REF = Mdi_opt_l1

EFF_LAMB_AV_CBK[1] = Eta_lambda_1 /100%

EFF_LAMB_AV_CBK[2] = Eta_lambda_2 /100%

EFF_LAMB_AV = 0.5 * (EFF_LAMB_AV_CBK[1] + EFF_LAMB_AV_CBK[2])

EFF_LAMB_BAS_COR = Eta_labas / 100%

TQI_AV_S = Mdi_s_istm

TQI_AV_HOM = Mdi_ist_m

TQI_MAX = Mdi_max

TQI_REF_S = Mdi_opt_s

EFF_LAMB_SP_BAS = Eta_labas_sol / 100%

TQI_REF_MAX = Mdi_max_l1

TQI_SP_MAF = Mdi_ml_soll

TQI_MIN_S = Mdi_min_s

general specification

```
If    LV_TQ_LIM_INTV = 1
then        LV_TQ_IGA_ENA    =  1
else        LV_TQ_IGA_ENA    =  B_zw_dynman
endif
```

FAC_CAM_CH = F_wnwkh

FAC_EFF_IGA_CH = F_wnwkhk

TQ_MAX_CLU = Mdk_max

TQI_MIN_HOMS = Mdi_min_hs

LV_TQI_BOL_SET = B_mdimin_sa

LV_GS_ACT_FL = B_vlschalt

FAC_EFF_IGA_SECU = F_sicher_zw

LV_TQ_MIN_CLU = B_llrein

TQ_MIN_CLU = Mdk_min
FAC_EFF_IGA_SCAV = F_atldyn

63 Torque losses

Table of Contents

Index

63.1 Idle Speed Adaptation

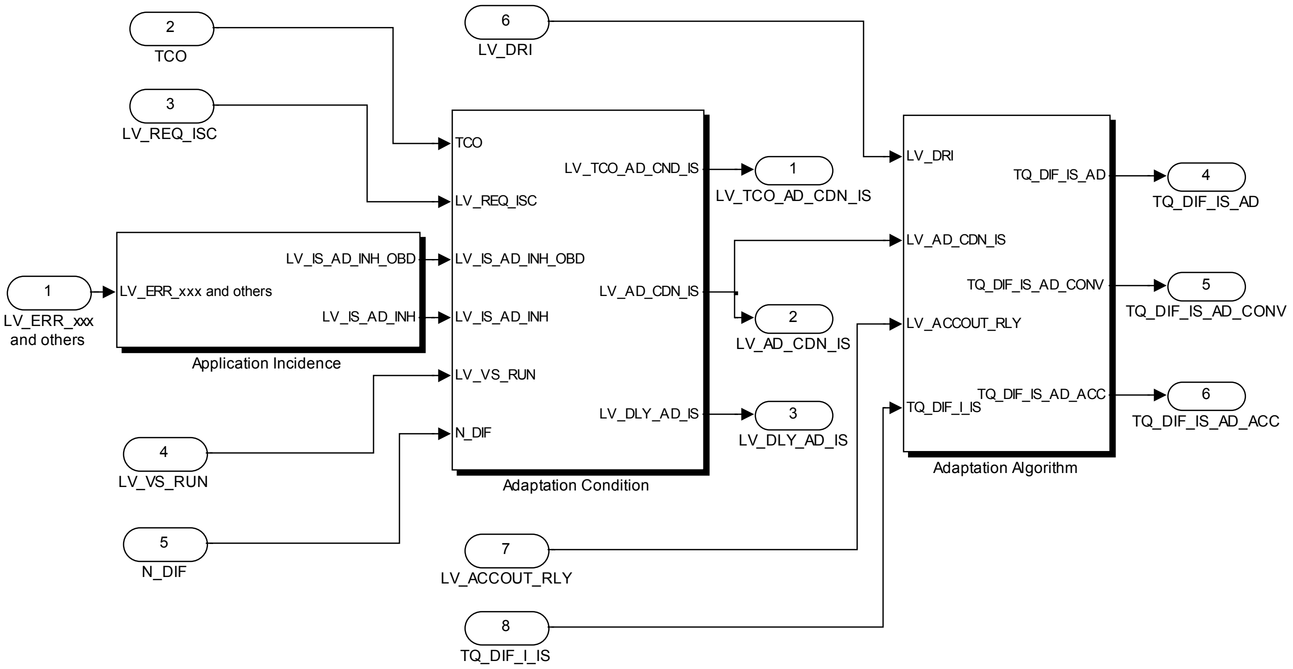


Figure 1: Idle Speed Adaptation, Overview

Functional Description:

The present adaptation is introduced to compensate spread and series variation of engine, air conditioning compressor and hydraulic converter torque losses. It is calculated during idle speed, using the integral part of the idle speed controller for the adaptation of three different parameters:

- TQ_DIF_IS_AD is the adaptation of the basic engine torque losses, without taking the drivetrain into account (disengaged gear),
- TQ_DIF_IS_AD_CONV is an adaptation of the hydraulic converter torque request at idle speed (for automatic transmission only),
- TQ_DIF_IS_AD_ACC is an adaptation value which consideres two adaptation cases depending on the actual situation:
  - air conditioning compressor and converter active at the same time
  - only air conditioning compressor active

The function consists of the following blocks:

- Application Incidence: Checks whether a relevant error has occurred.
- Adaptation Condition: Checks whether the adaptation conditions are fulfilled.
- Adaptation Algorithm: Calculation of the specific adaptation parameters.

These parameters are integrated directly in the specific functions, i.e. the adaptation of the converter torque TQ_DIF_IS_AD_CONV is used in the function "Converter torque" ...

Adaptation parameters for new torque consuming components should be integrated in the same way into the adaptation algorithm module.

63.2 Application Incidences for Idle speed adaptation

Output data:

Name	Mode	Hex. Limit	Phys. Limit	Resol.	Unit.
LV_IS_AD_INH_OBD	V/O	0...1H	0...1	1	[-]
Application incidence value					
LV_IS_AD_INH	V/O	0...1H	0...1	1	[-]
Application incidence value (additional adaptation inhibitation conditions, project specific)					

Input data:

LV_ERR_MAF	LV_ERR_PVS_DOUBLE	LV_ERR_LOAD_TPS_PLAUS	LV_ERR_BN_VS_TCS
LV_ERR_TCO	LV_ERR_AMP	LV_ERR_IGC	LV_ERR_AMP_PLAUS
STATE_ERR_IV	LV_ERR_TPS	LV_ERR_DIAGCPS	LV_ERR_EL_CPS
LV_MIS_STATE_A	LV_MTC_CUR_OFF	LV_ERR_VS	LV_ERR_CAN_BOFF
	LV_ERR_TOUT_ASR_1	LV_TQ_DROF_IS_AD_INH	
LV_TQ_LOSS_PSTE_IS_AD_INH	LV_TQ_LOSS_ARS_IS_AD_INH		LV_TQ_ADD_PSTE_IS_AD_INH
			LV_ST_END
TCO			LV_ERR_MAP_TPS_PLAUS

FUNCTION DESCRIPTION:

General information:

The idle speed adaptation is inhibited if on of the errors described in the formula section occur. Additionally it is possible to inihibit the idle speed adaptation for a certain time after start dependent on the start temperature.

Application conditions:

Initialization:	all 0 at LV_IGK 0 > 1 and reset
Recurrence:	100 ms
Activation:	at every engine state

Description:

If one of these error bits is set, LV_IS_AD_INH_OBD = 1 and LV_IS_AD_INH = 1.
Adaptation is not allowed see application conditions.

Formula section:

```
If      LV_ST_END = 0->1
Then    T_AST_IS_AD_INH = IP_T_AST_IS_AD_INH     /*LV_ST_END setting initiates timer*/
Else    T_AST_IS_AD_INH --                       /*timer is decremented*/
Endif


If      T_AST_IS_AD_INH > 0                      or
        LV_ERR_MAF = 1                           or
        LV_ERR_PVS_DOUBLE = 1                    or
        STATE_ERR_IV <> 0                        or
        LV_ERR_IGC                               or
        LV_ERR_LOAD_TPS_PLAUS = 1                or
        LV_ERR_TPS = 1                           or
        LV_ERR_EL_CPS = 1                        or
        LV_ERR_DIAGCPS = 1                       or
        LV_MIS_STATE_A = 1                       or
        LV_ERR_TCO = 1                           or
        LV_ERR_AMP = 1                           or
        LV_ERR_AMP_PLAUS = 1                     or
        LV_MTC_CUR_OFF = 1                       or
        LV_TQ_LOSS_PSTE_IS_AD_INH = 1            or
        LV_TQ_ADD_PSTE_IS_AD_INH = 1             or
        LV_TQ_LOSS_ARS_IS_AD_INH = 1             or
        LV_TQ_DROF_IS_AD_INH = 1                 or
        LV_ERR_VS = 1                            or
        LV_ERR_MAP_TPS_PLAUS                     or
Then    LV_IS_AD_INH = 1
        LV_IS_AD_INH_OBD = 1
Else    LV_IS_AD_INH = 0
        LV_IS_AD_INH_OBD = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_T_AST_IS_AD_INH	4	0...FFFFH	0...6553.5	0.1	[s]
LDP_TCO_IP_T_AST_IS_AD_INH	4	0...FEH	-48...142.5	0.75	[°C]
Time after start during which idle speed adaptation is disabled					

63.3 Adaptation Condition

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
LV_DLY_AD_IS	O/V	0...1H	0...1	1	-
Logical bit "Idle speed has settled for a given time delay"					
LV_TCO_AD_CDN_IS	O/V	0...1H	0...1	1	-
Logical bit "Coolant temperature in operating range"					
LV_AD_CDN_IS	O/V	0...1H	0...1	1	-
Adaptation condition					

Input data:

N_DIF	TCO	LV_REQ_ISC	LV_VS_RUN
LV_IS_AD_INH	LV_IS_AD_INH_OBD		

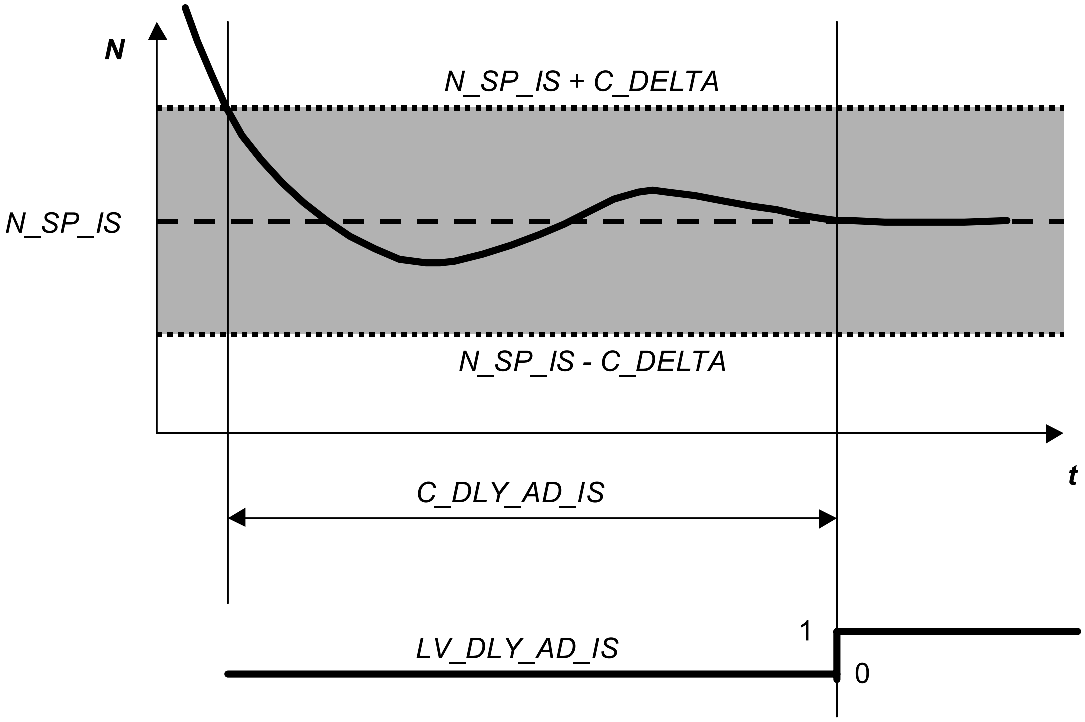
Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_DLY_AD_IS	1	0...FFH	0...25.5	0.1	s
time delay for ISC adaptation					
C_N_DELTA	1	0...1FE0H	0...8.16E+3	1	rpm
engine speed threshold for ISC adaptation					
C_TCO_AD_MAX_IS	1	0...FEH	-48...142.5	0.75	°C
temperature top limit for ISC adaptation					
C_TCO_AD_MIN_IS	1	0...FEH	-48...142.5	0.75	°C
temperature bottom limit for ISC adaptation					

63.3.1 Activation conditions for the Idle Speed Adaptation

The adaptation process is released when all of the following conditions are fulfilled (LV_AD_CDN_IS = 1):

- ♦ engine running in idle state (LV_REQ_ISC = 1)
- ♦ idle speed has settled for a given time delay (T_DLY_AD_IS is the timer value):



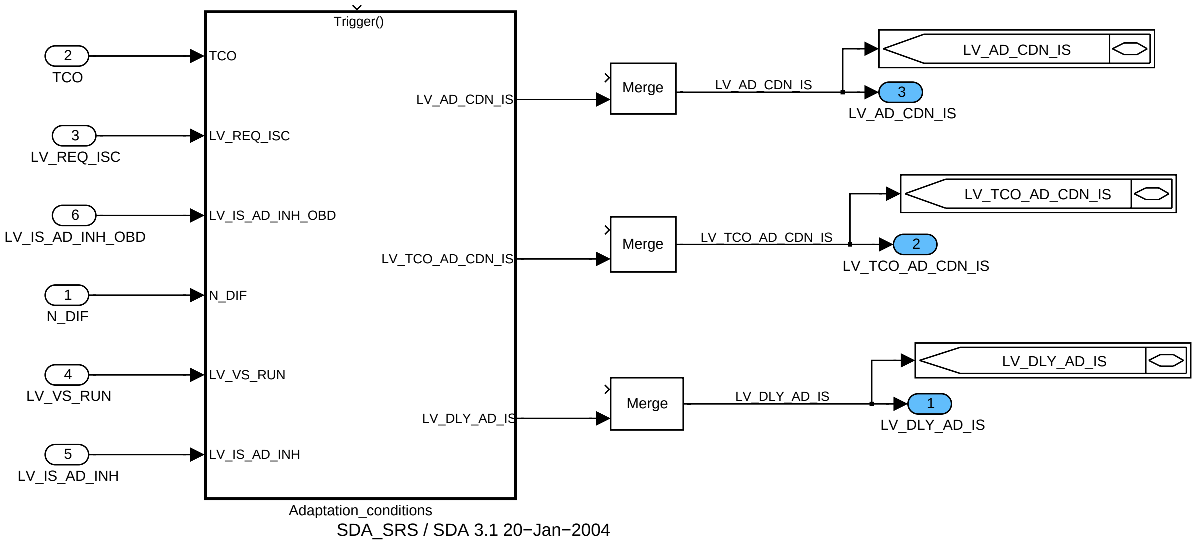
Delay timer for idle speed adaptation activation

- vehicle stopped (LV_VS_RUN = 0)
- coolant temperature in operating range:

LV_IS_AD_INH_OBD = 0 (see module "Application Incidence")

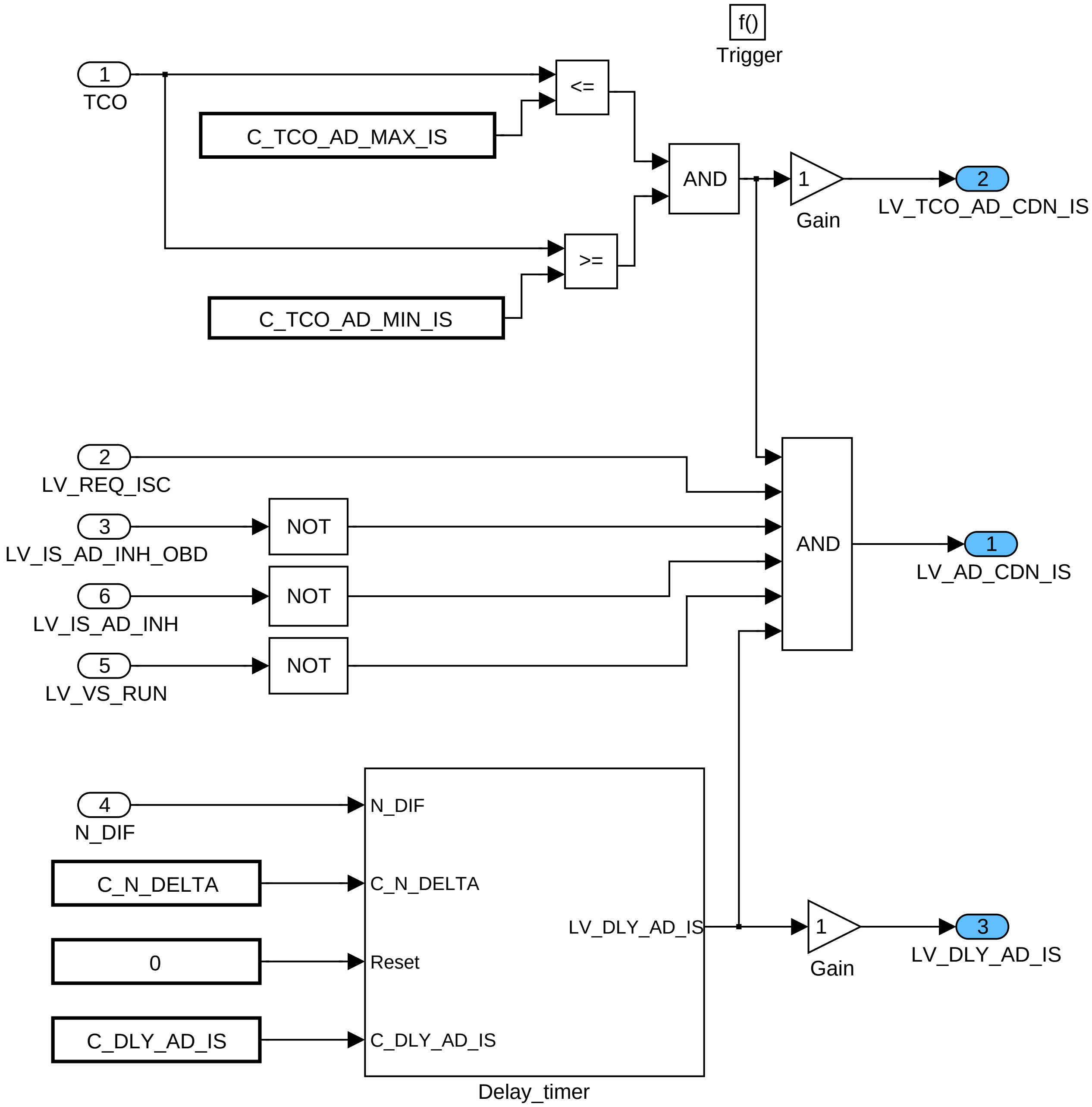
- LV_IS_AD_INH = 0 (see module "Application Incidence")

Function Description

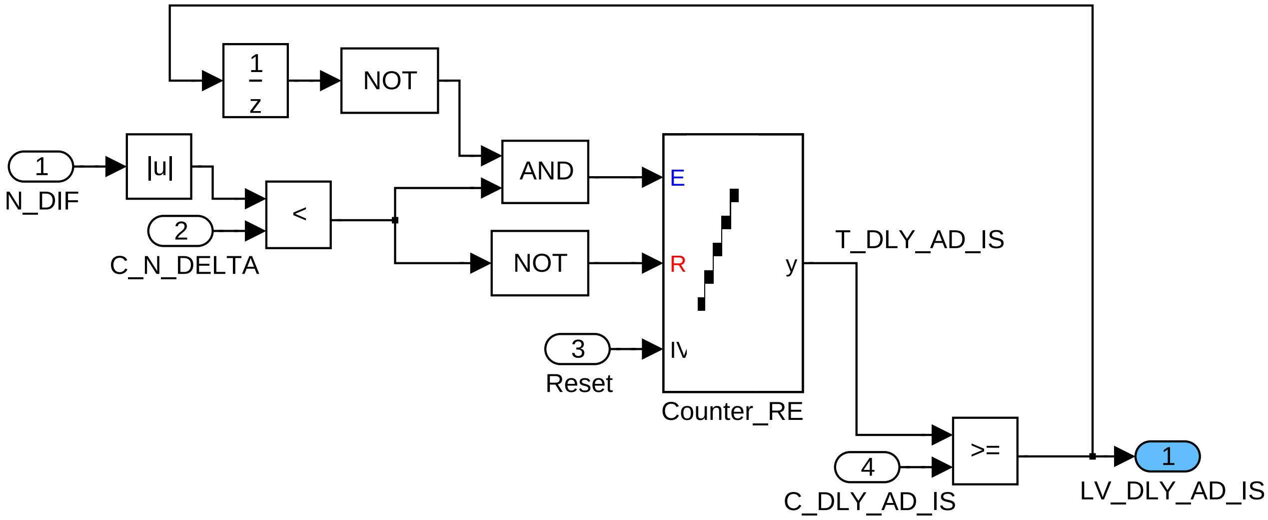


M800M

63.3.1.1 SUBFUNCTION: ADAPTATION_CONDITIONS



M800M/Adaptation_conditions



M800M/Adaptation_conditions/Delay_timer

63.4 Idle Speed Adaptation; Adaptation Algorithm

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_DIF_IS_AD	O/V/S	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Basic ISC adaptation					
TQ_DIF_IS_AD_CONV	O/V/S	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Hydraulic converter adaptation torque					
TQ_DIF_IS_AD_ACC	O/V/S	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Low-pass filtered ACC torque adaptation or ACC and Hydraulic converter torque adaptation					
TQ_DIF_IS_AD_OPM_1	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Basic ISC adaptation, intermediate value with high resolution (long signed word) OPM1					
TQ_DIF_IS_AD_OPM_2	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Basic ISC adaptation, intermediate value with high resolution (long signed word) OPM2					
TQ_DIF_IS_AD_ACC_1_OPM_1	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC torque adaptation, intermediate value with high resolution (long signed word) OPM1					
TQ_DIF_IS_AD_ACC_1_OPM_2	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC torque adaptation, intermediate value with high resolution (long signed word) OPM2					
TQ_DIF_IS_AD_ACC_CONV_OPM_1	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC and Hydraulic converter torque adaptation, intermediate value with high resolution (long signed word) OPM1					
TQ_DIF_IS_AD_ACC_CONV_OPM_2	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC and Hydraulic converter torque adaptation, intermediate value with high resolution (long signed word) OPM2					
TQ_DIF_IS_AD_CONV_NEUT	O/V/S	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Hydraulic converter adaptation torque with neutral control active					
TQ_DIF_IS_AD_CONV_1	O/V/S	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Hydraulic converter adaptation torque with neutral control passive					
TQ_DIF_IS_AD_ACC_NEUT	O/V/S	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
ACC and Hydraulic converter torque adaptation with neutral control active					
TQ_DIF_IS_AD_ACC_CONV	O/V/S	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
ACC and Hydraulic converter torque adaptation					
TQ_DIF_IS_AD_ACC_1	O/V/S	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
ACC torque adaptation					
TQ_DIF_IS_AD_CONV_1_OPM_1	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Hydraulic converter adaptation torque with neutral control passive, intermediate value with high resolution (long signed word) OPM1					
TQ_DIF_IS_AD_CONV_1_OPM_2	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Hydraulic converter adaptation torque with neutral control passive, intermediate value with high resolution (long signed word) OPM2					
TQ_DIF_IS_AD_ACC_NEUT_OPM_1	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC and Hydraulic converter torque adaptation with neutral control active, intermediate value with high resolution (long signed word) OPM1					

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_DIF_IS_AD_ACC_NEUT_OPM_2	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC and Hydraulic converter torque adaptation with neutral control active, intermediate value with high resolution (long signed word) OPM2					
TQ_DIF_IS_AD_CONV_NEUT_OPM_1	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Hydraulic converter adaptation torque with neutral control active, intermediate value with high resolution (long signed word) OPM1					
TQ_DIF_IS_AD_CONV_NEUT_OPM_2	O/V/S	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Hydraulic converter adaptation torque with neutral control active, intermediate value with high resolution (long signed word) OPM2					
FAC_TQ_DIF_IS_AD	V	0...FFFFH	0...0.99998474	1.52588E-5	-
ISC adaptation factor					
LV_OPM_SEL_TQ_IS_AD_1	V	0...1H	0...1	1	-
Last operation mode calculation active					
LV_OPM_SEL_TQ_IS_AD_2	V	0...1H	0...1	1	-
Requested value active					
LV_AT_NEUT_CTL	V	0...1H	0...1	1	-
Logical variable for neutral control active/passive (drive engaged)					
TQ_DIF_IS_AD_INTER	V	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Basic ISC adaptation, intermediate value with high resolution (long signed word)					
TQ_DIF_IS_AD_ACC_1_INTER	V	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC torque adaptation, intermediate value with high resolution (long signed word)					
TQ_DIF_IS_AD_ACC_NEUT_INTER	V	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC torque adaptation with neutral control active and drive engaged, intermediate value with high resolution (long signed word)					
TQ_DIF_IS_AD_ACC_CONV_INTER	V	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
ACC and Hydraulic converter torque adaptation, intermediate value with high resolution (long signed word)					
TQ_DIF_IS_AD_CONV_NEUT_INTER	V	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Hydraulic converter adaptation torque with neutral control active, intermediate value with high resolution (long signed word)					
TQ_DIF_IS_AD_CONV_1_INTER	V	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837E-7	Nm
Adaptation torque for hydraulic converter, intermediate value with high resolution (long signed word)					

Input data:

LV_ACCOUT_RLY	LV_DRI	TQ_DIF_I_IS	OPM_AV
OPM_REQ	FAC_TQ_LOSS_OPM_SEL	LV_AD_CDN_IS	STATE_MOD_GB
N_32	TQ_REQ_CLU	LV_TQ_LOSS_AD_CLR_EXT_ADJ	LC_AD_CLR

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_TQ_DIF_IS_AD_ACC	1	0...FFH	0...0.99609375	0.00390625	-
Filter constant for LOW_PASS_1 (smoothing of the ACC-adaptation values)					
C_CRLC_TQ_DIF_IS_AD_CONV	1	0...FFH	0...0.99609375	0.00390625	-
Filter constant for smoothing of the converter torque adaptation values					

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TQ_DIF_IS_AD_ACC_CONV_MAX	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Max. limitation of air conditioning compressor and converter torque adaptation					
C_TQ_DIF_IS_AD_ACC_CONV_MIN	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Min. limitation of air conditioning compressor and converter torque adaptation					
C_TQ_DIF_IS_AD_ACC_MAX	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Max. limitation for ACC torque adaptation					
C_TQ_DIF_IS_AD_ACC_MIN	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Min. limitation for ACC adaptation					
C_TQ_DIF_IS_AD_CONV_MAX	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Max. limitation for converter torque adaptation					
C_TQ_DIF_IS_AD_CONV_MIN	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Min. limitation of converter torque adaptation					
C_TQ_DIF_IS_AD_MAX	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Max. limitation of engine torque losses adaptation					
C_TQ_DIF_IS_AD_MIN	1	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Min. limitation of engine torque losses adaptation					
ID_TQ_DIF_IS_AD_OPM_AV	8	1...8H	1...8	1	-
LDP_OPM_AV_TQLO	8	0...8H	0...8	1	-
Table of index OPM_AV					
ID_TQ_DIF_IS_AD_OPM_REQ	8	1...8H	1...8	1	-
LDP_OPM_AV_TQLO	8	0...8H	0...8	1	-
Table of index OPM_REQ					
IP_FAC_TQ_DIF_IS_AD	4	0...FFFFH	0...0.99998474	1.52588E-5	-
LDP_TQ_DIF_I_IS_FAC_TQ_IS_AD	4	0...7FFFH	0...1.02397E+3	0.03125	Nm
ISC adaptation factor depend on absolut value of TQ_DIF_IS_I					
IP_FAC_TQ_DIF_IS_AD_OPM_MV	6x6	0...FFH	0...0.99609375	0.00390625	-
LDP_N_32_FAC_TQ_DIF_IS_AD	6	0...FFH	0...8.16E+3	32	rpm
LDP_TQ_REQ_CLU_FAC_TQ_DIF_IS_AD	6	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Factor for fading between TQ_DIF_IS_AD_INTER and the mean value of the adaptation values TQ_DIF_IS_AD_OPM_1/_2 in dependency of engine speed/ load					

Local actions:

ACTION_TQLO_OPM_M800N(IN <IN1>, IN <IN2>, OUT <OUT>, OUT <LV_OPM_SEL_TQ_IS_AD_1>, OUT <LV_OPM_SEL_TQ_IS_AD_2>)
OPM with I/O resolution 4.768E-7 and phys limits -1024...1023.9999[Nm]

63.4.1 General Information

Application conditions:

Initialization:

E2PROM Variables

At first initialization, after a checksum failure or at service tester request LV_TQ_LOSS_AD_CLR_EXT_ADJ or LC_AD_CLR 0 ->1 , all parameters are set to zero:

TQ_DIF_IS_AD = 0 Nm,

TQ_DIF_IS_AD_CONV_NEUT = 0 Nm,

TQ_DIF_IS_AD_CONV = 0 Nm,

TQ_DIF_IS_AD_CONV_1 = 0 Nm,

TQ_DIF_IS_AD_ACC_NEUT = 0 Nm,

TQ_DIF_IS_AD_ACC_CONV = 0 Nm,

TQ_DIF_IS_AD_ACC = 0 Nm,

TQ_DIF_IS_AD_ACC_1 = 0 Nm,

TQ_DIF_IS_AD_OPM_1 = 0 Nm,

TQ_DIF_IS_AD_OPM_2 = 0 Nm,

TQ_DIF_IS_AD_ACC_1_OPM_1 = 0 Nm,

TQ_DIF_IS_AD_ACC_1_OPM_2 = 0 Nm,

TQ_DIF_IS_AD_ACC_CONV_OPM_1 = 0 Nm,

TQ_DIF_IS_AD_ACC_CONV_OPM_2 = 0 Nm,

TQ_DIF_IS_AD_CONV_1_OPM_1 = 0 Nm,

TQ_DIF_IS_AD_CONV_1_OPM_2 = 0 Nm,

TQ_DIF_IS_AD_ACC_NEUT_OPM_1 = 0 Nm,

TQ_DIF_IS_AD_ACC_NEUT_OPM_2 = 0 Nm,

TQ_DIF_IS_AD_CONV_NEUT_OPM_1 = 0 Nm,

TQ_DIF_IS_AD_CONV_NEUT_OPM_2 = 0 Nm

Variables initialized after reset and after the E2PROM has been read

TQ_DIF_IS_AD_ACC_1_INTER = TQ_DIF_IS_AD_ACC_1,

TQ_DIF_IS_AD_ACC_NEUT_INTER = TQ_DIF_IS_AD_ACC_NEUT,

TQ_DIF_IS_AD_ACC_CONV_INTER = TQ_DIF_IS_AD_ACC_CONV,

TQ_DIF_IS_AD_CONV_NEUT_INTER = TQ_DIF_IS_AD_CONV_NEUT,

TQ_DIF_IS_AD_CONV_1_INTER = TQ_DIF_IS_AD_CONV_1,

LV_AT_NEUT_CTL = 0 [-]

Activation:

always

Deactivation:

-

Update rate:

100 ms

FUNCTION DESCRIPTION:

The complete function is calculated every 100ms independent of the status of the adaptation condition. If the adaptation conditions are fulfilled (LV_AD_CDN_IS = 1), the corresponding adaptation value is fed by the I-gain of the idle speed controller value TQ_DIF_I_IS :

- The adaptation of ACC and CONV (TQ_DIF_IS_AD_ACC_CONV) is calculated if the gear is engaged (LV_DRI = 1) and the air conditioning compressor is switched on (LV_ACCOUT_RLY = 1).
- The adaptation of ACC and CONV with neutral control active (TQ_DIF_IS_AD_ACC_NEUT) is calculated if the gear is engaged (LV_DRI = 1), neutral control active (LV_AT_NEUT_CTL = 1) and the air conditioning compressor is switched on (LV_ACCOUT_RLY = 1).
- If LV_ACCOUT_RLY = 1 and LV_DRI = 0, the adaptation of the air conditioning compressor is calculated (TQ_DIF_IS_AD_ACC_1).
- The Hydraulic converter torque adaptation with neutral control passive (TQ_DIF_IS_AD_CONV_1) is calculated if LV_DRI = 1, LV_AT_NEUT_CTL = 0 and LV_ACCOUT_RLY =0.
- The Hydraulic converter torque adaptation with neutral control active (TQ_DIF_IS_AD_CONV_NEUT) is calculated if LV_DRI = 1, LV_AT_NEUT_CTL = 1 and LV_ACCOUT_RLY =0.
- The basic adaptation of the idle speed controller (TQ_DIF_IS_AD) is calculated if the gear is disengaged (LV_DRI = 0) and the air conditioning compressor is switched off (LV_ACCOUT_RLY = 0).

Therefore, a fraction of the integral part of the idle speed controller (FAC_TQ_DIF_IS_AD * TQ_DIF_I_IS) is added at each cycle to the concerned adaptation parameter, whose value is limited through a min/max limiter. The adaptation parameters are stored, even after deactivation of the function. These values are used as start values after re-entering into the same adaptation path.

The outputs of the different adaptation paths are always covered with values of the last adaptation, even if the adaptation conditions are not fulfilled or the engine is not in idle. But these values will only be adapted (changed) if the certain adaptation path is activated and they are certainly only considered in the single modules (“Torque Loss and Lead for ACC”, “Converter Torque”) in conformance with the actual situation (ACC on, drivetrain engaged). Only the basic torque losses (TQ_DIF_IS_AD) are always considered in the module “Torque Losses”.

TQ_DIF_IS_AD_CONV is considered in the converter torque calculation module, TQ_DIF_IS_AD_ACC in the calculation of the air conditioning compressor losses. TQ_DIF_IS_AD is added in the torque losses.

The integration is done with high resolution values (long signed word).

At the end of the integration the conversion from high resolution (signed long word) to “normal” resolution (signed word) is done. Only the high word is taken from the signed long word value.

Application Condition

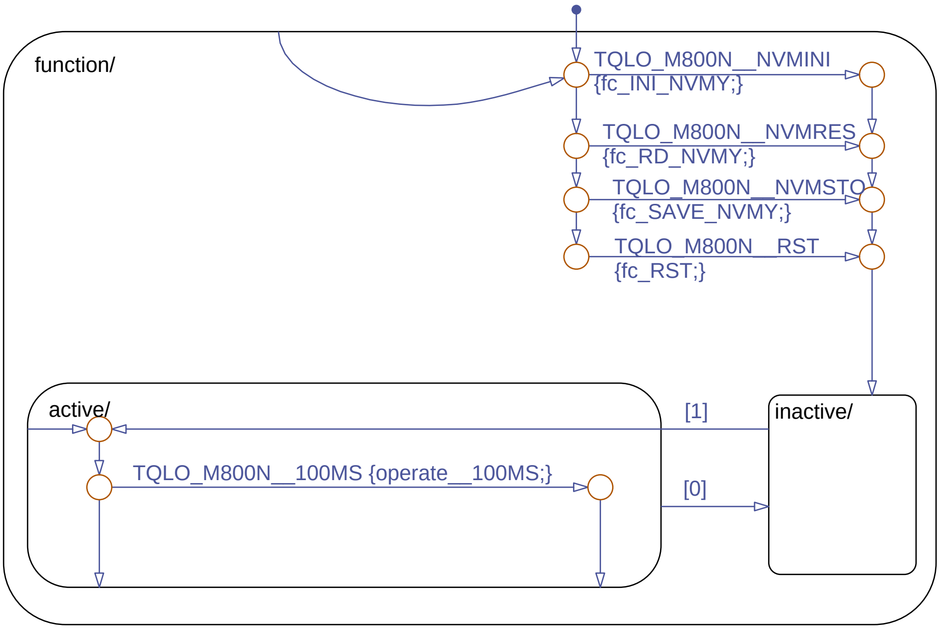
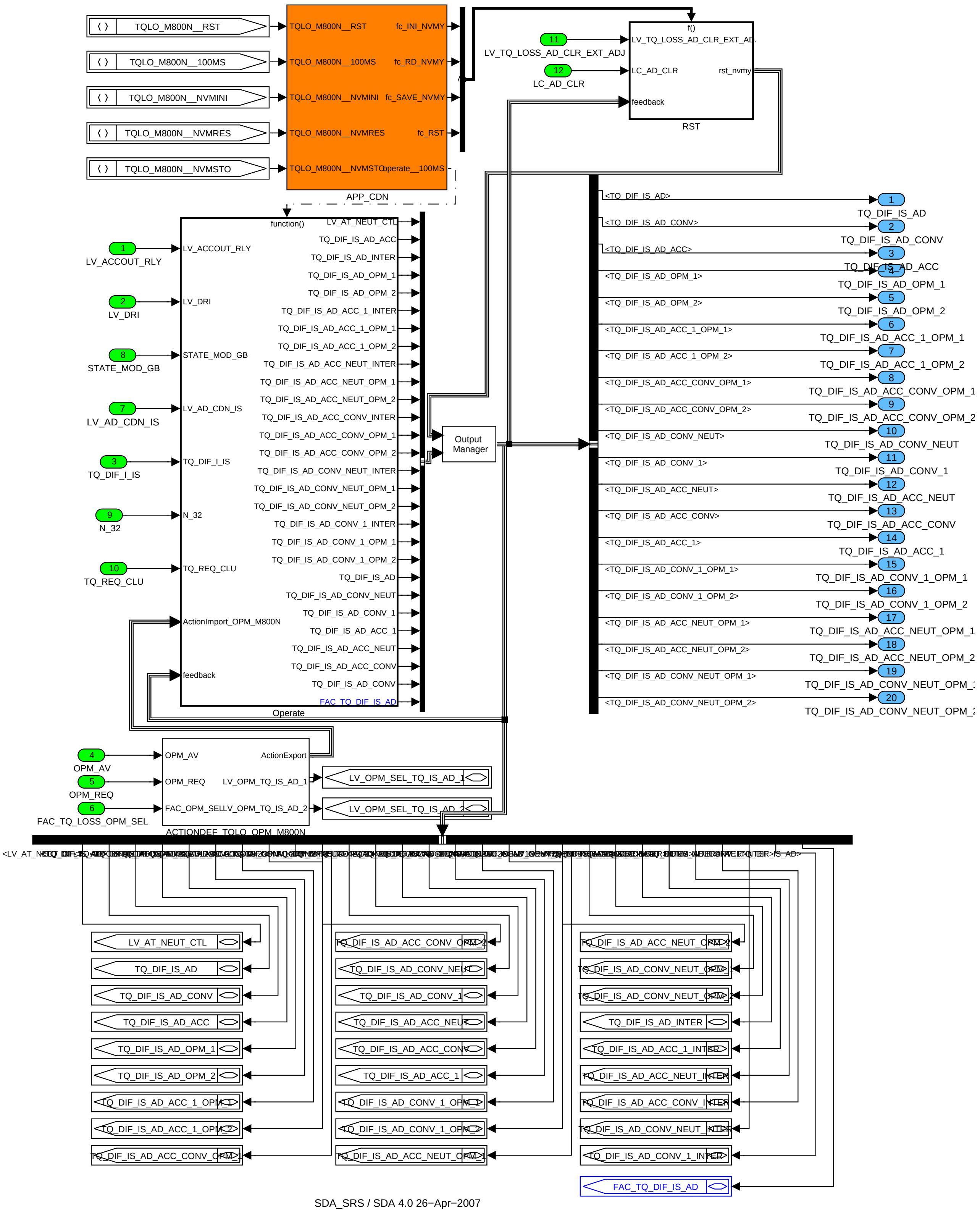


Figure 1 TQLO_M800N/ APP_CDN/ Chart1

Function Description



SDA_SRS / SDA 4.0 26−Apr−2007

Figure 2 TQLO_M800N

63.4.1.1 Action to smooth the transition between the operating modes

Description for ACTION_TQLO_OPM_M800N

ACTION_TQLO_OPM_M800N(IN <IN1>, IN <IN2>, OUT <OUT>, OUT <LV_OPM_SEL_TQ_IS_AD_1>, OUT <LV_OPM_SEL_TQ_IS_AD_2>)					
Action: OPM with I/O resolution 4.768E-7 and phys limits -1024...1023.9999[Nm]					
Parameter	**Type**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
IN1	IN	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837 E-7	Nm
Basic ISC adaptation, intermediate value with high resolution (long signed word)					
IN2	IN	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837 E-7	Nm
Basic ISC adaptation, intermediate value with high resolution (long signed word)					
OUT	OUT	80000000...7FFFFFFFH	-1.024E+3 ... 1.024E+3	4.76837 E-7	Nm
Basic ISC adaptation, intermediate value with high resolution (long signed word)					
LV_OPM_SEL_TQ_IS_AD_1	OUT	0...0H	0...0	0	-
UNDEFINED DATA - NO DATADEFINTION ENTERED YET!!!					
LV_OPM_SEL_TQ_IS_AD_2	OUT	0...0H	0...0	0	-
UNDEFINED DATA - NO DATADEFINTION ENTERED YET!!!					

The function controls the transition between the operating modes. As the operating mode changes the adaptation values change as well.

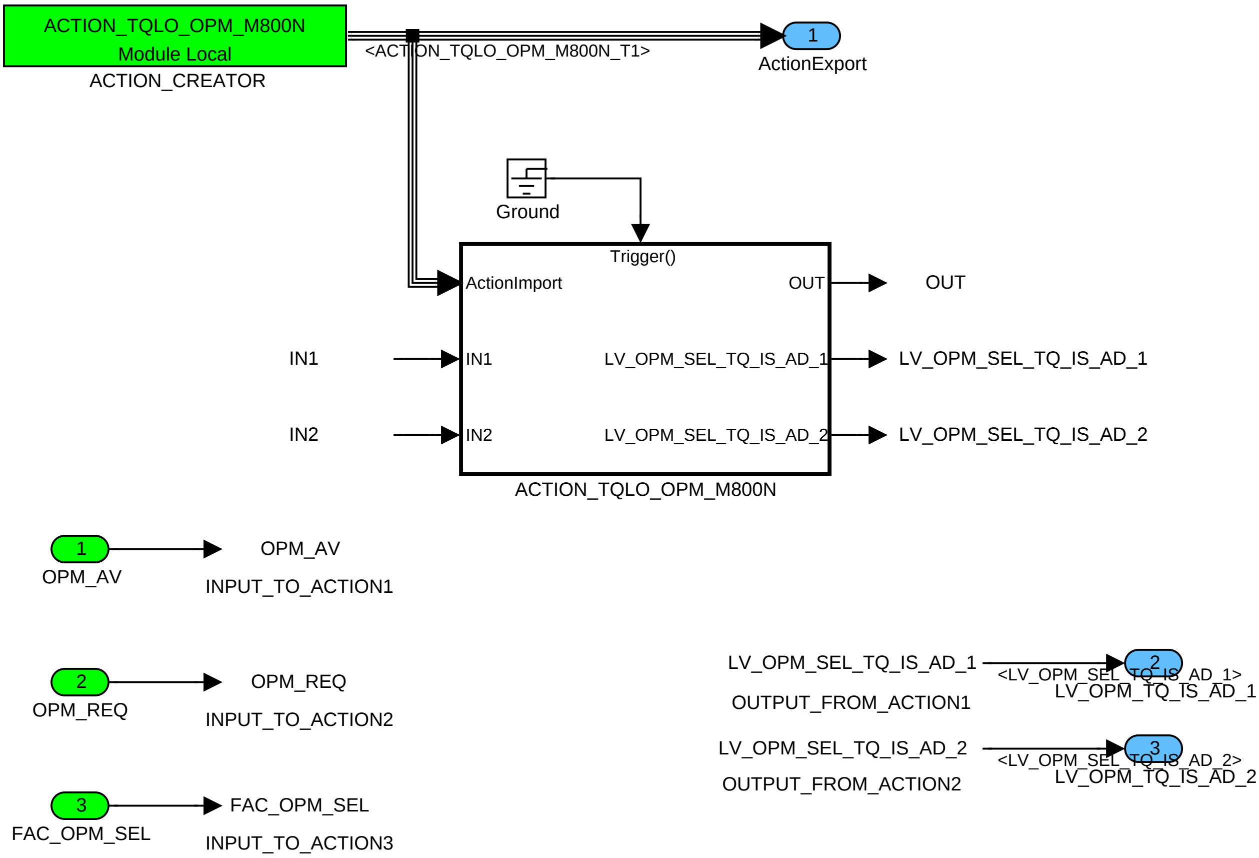


Figure 3 TQLO_M800N/ ACTIONDEF_TQLO_OPM_M800N

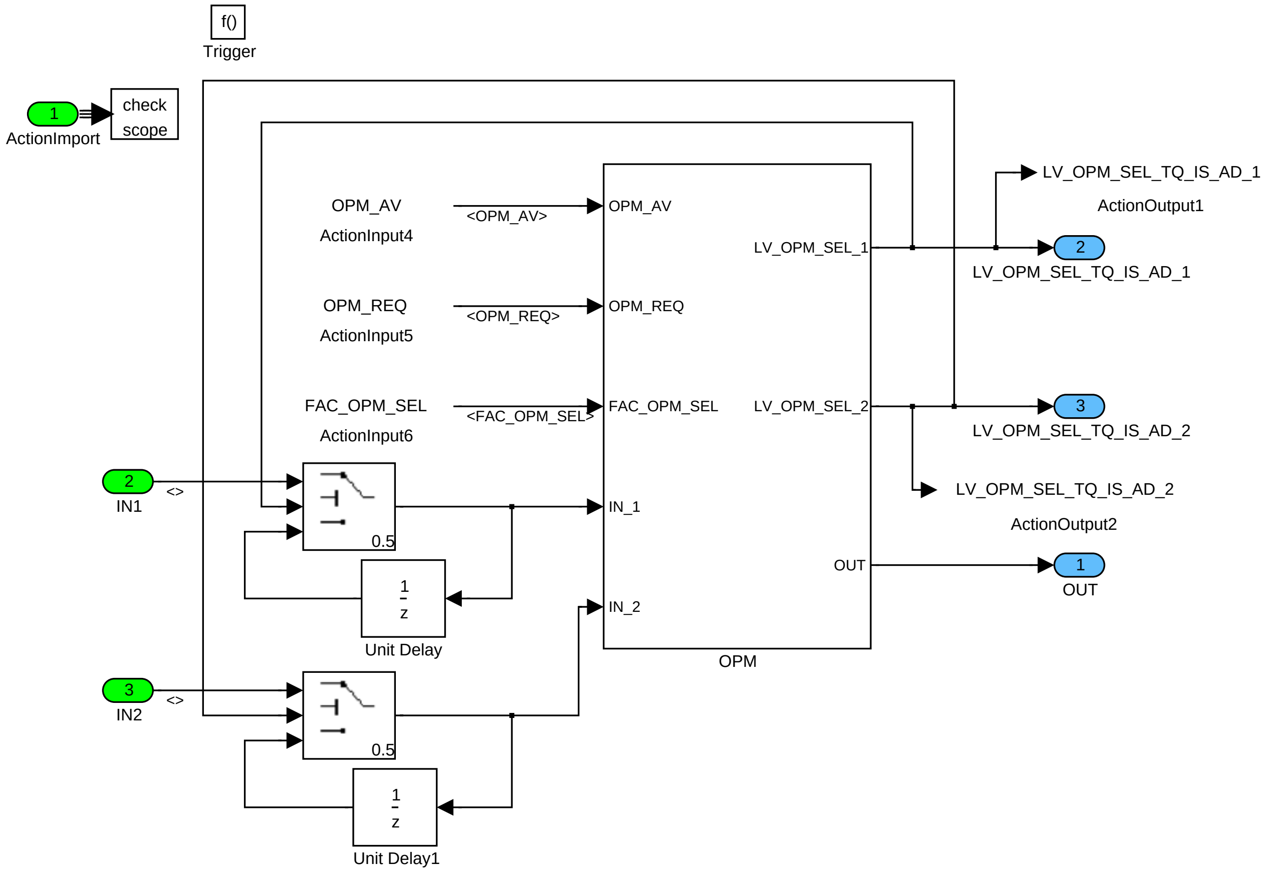


Figure 4 TQLO_M800N/ ACTIONDEF_TQLO_OPM_M800N/ ACTION_TQLO_OPM_M800N

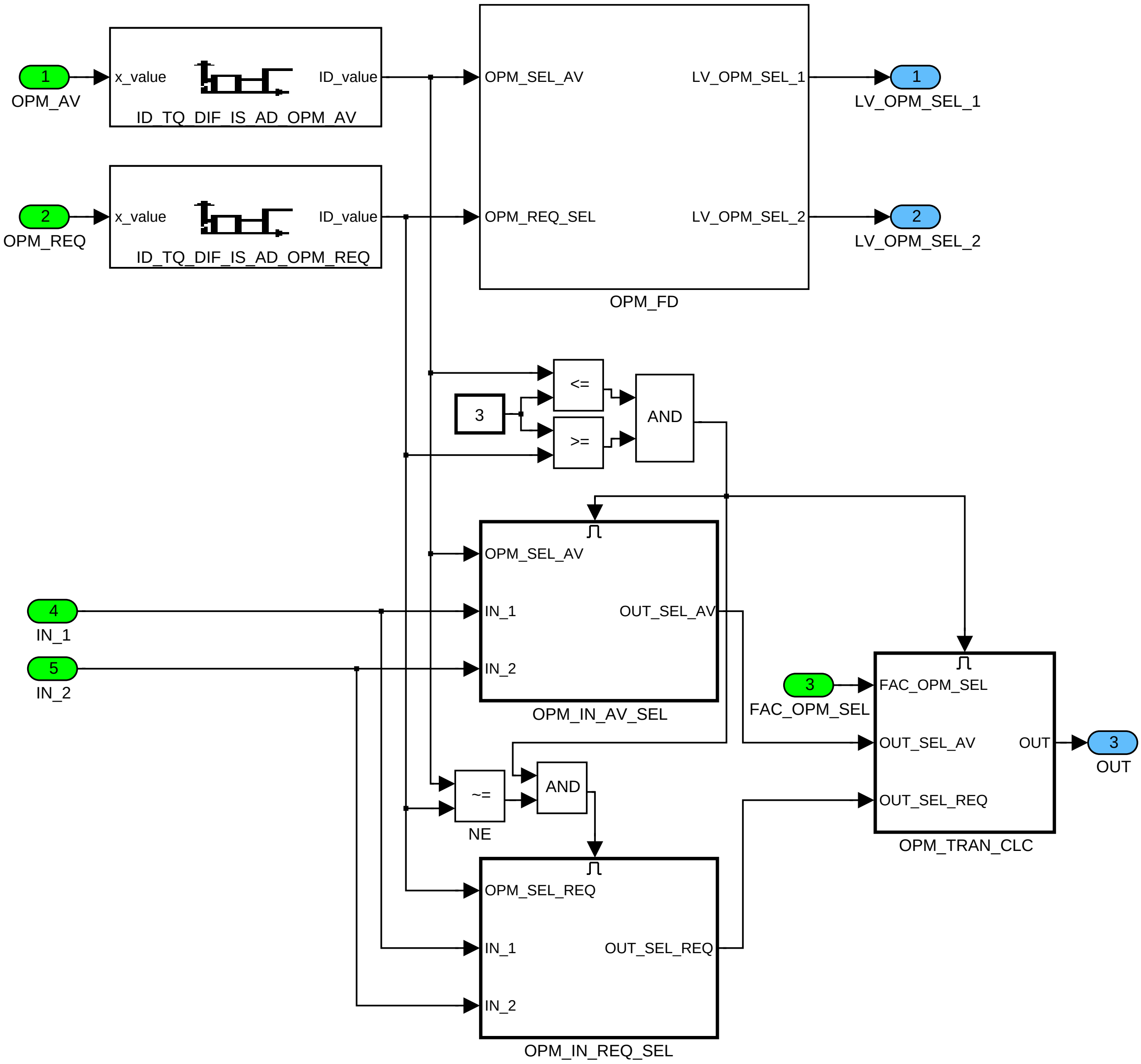


Figure 5 TQLO_M800N/ ACTIONDEF_TQLO_OPM_M800N/ ACTION_TQLO_OPM_M800N/ OPM

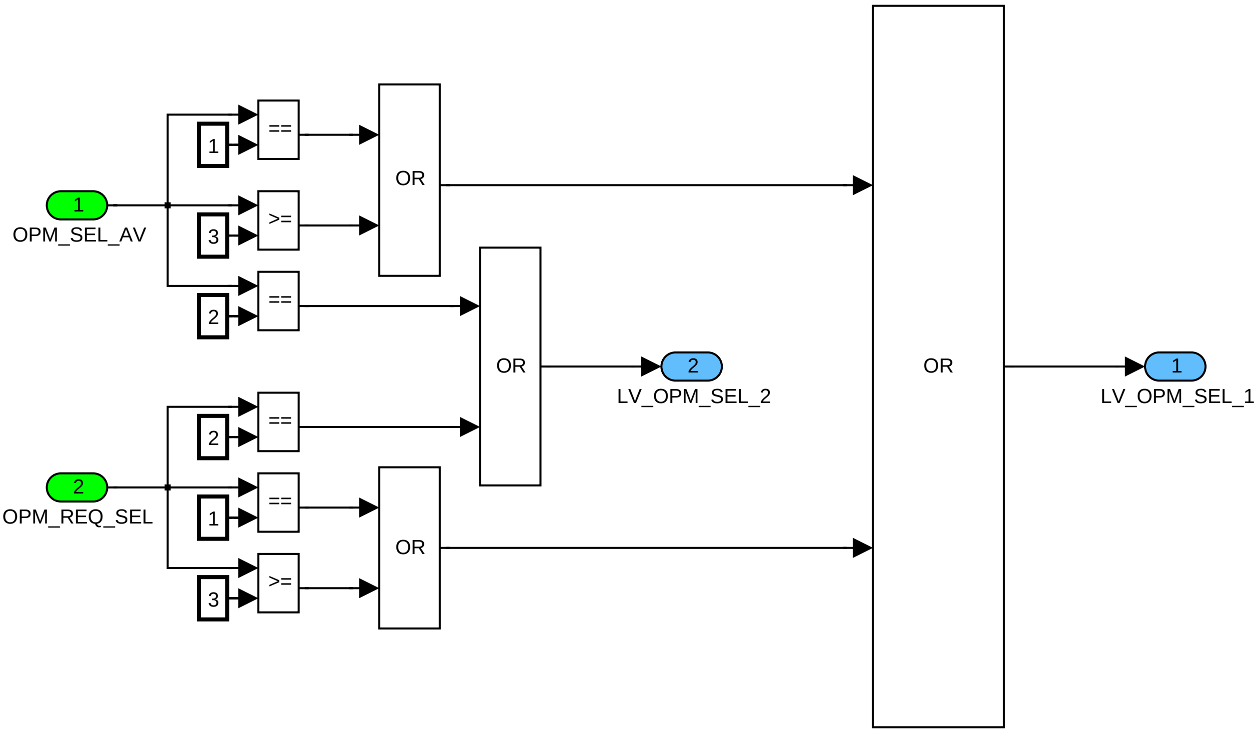


Figure 6 TQLO_M800N/ ACTIONDEF_TQLO_OPM_M800N/ ACTION_TQLO_OPM_M800N/ OPM/ OPM_FD

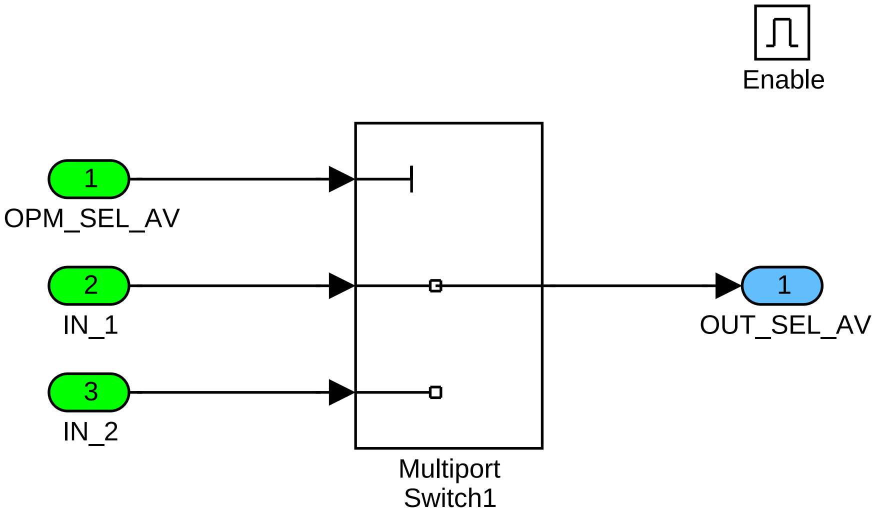


Figure 7 TQLO_M800N/ ACTIONDEF_TQLO_OPM_M800N/ ACTION_TQLO_OPM_M800N/ OPM/ OPM_IN_AV_SEL

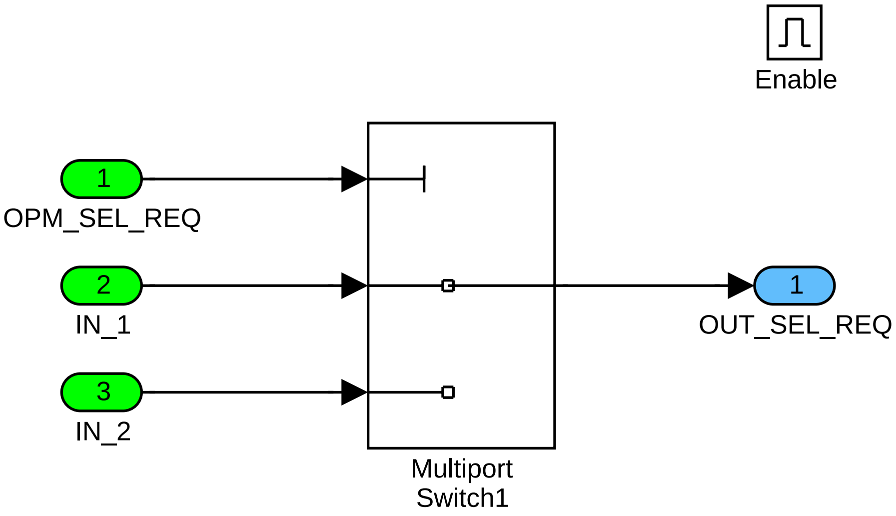


Figure 8 TQLO_M800N/ ACTIONDEF_TQLO_OPM_M800N/ ACTION_TQLO_OPM_M800N/ OPM/ OPM_IN_REQ_SEL

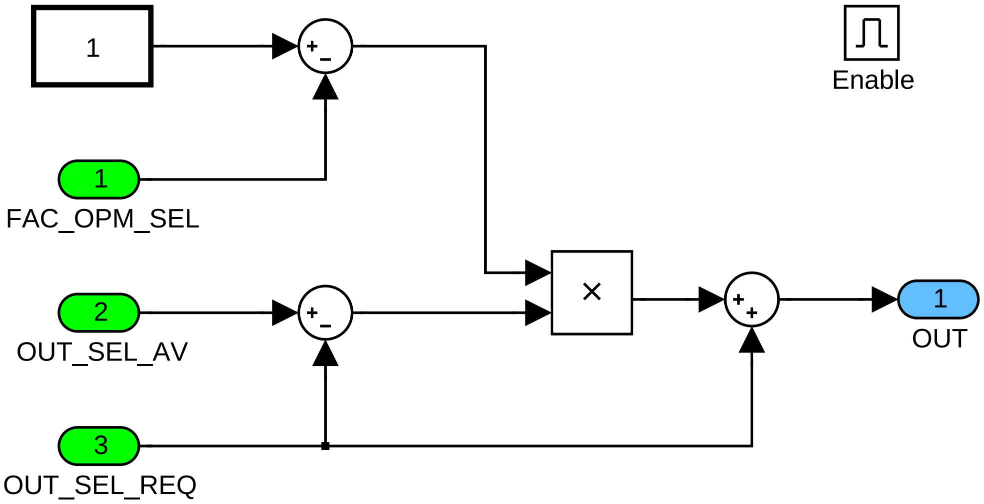


Figure 9 TQLO_M800N/ ACTIONDEF_TQLO_OPM_M800N/ ACTION_TQLO_OPM_M800N/ OPM/ OPM_TRAN_CLC

63.4.1.2 Overview Reset and NVMY handling

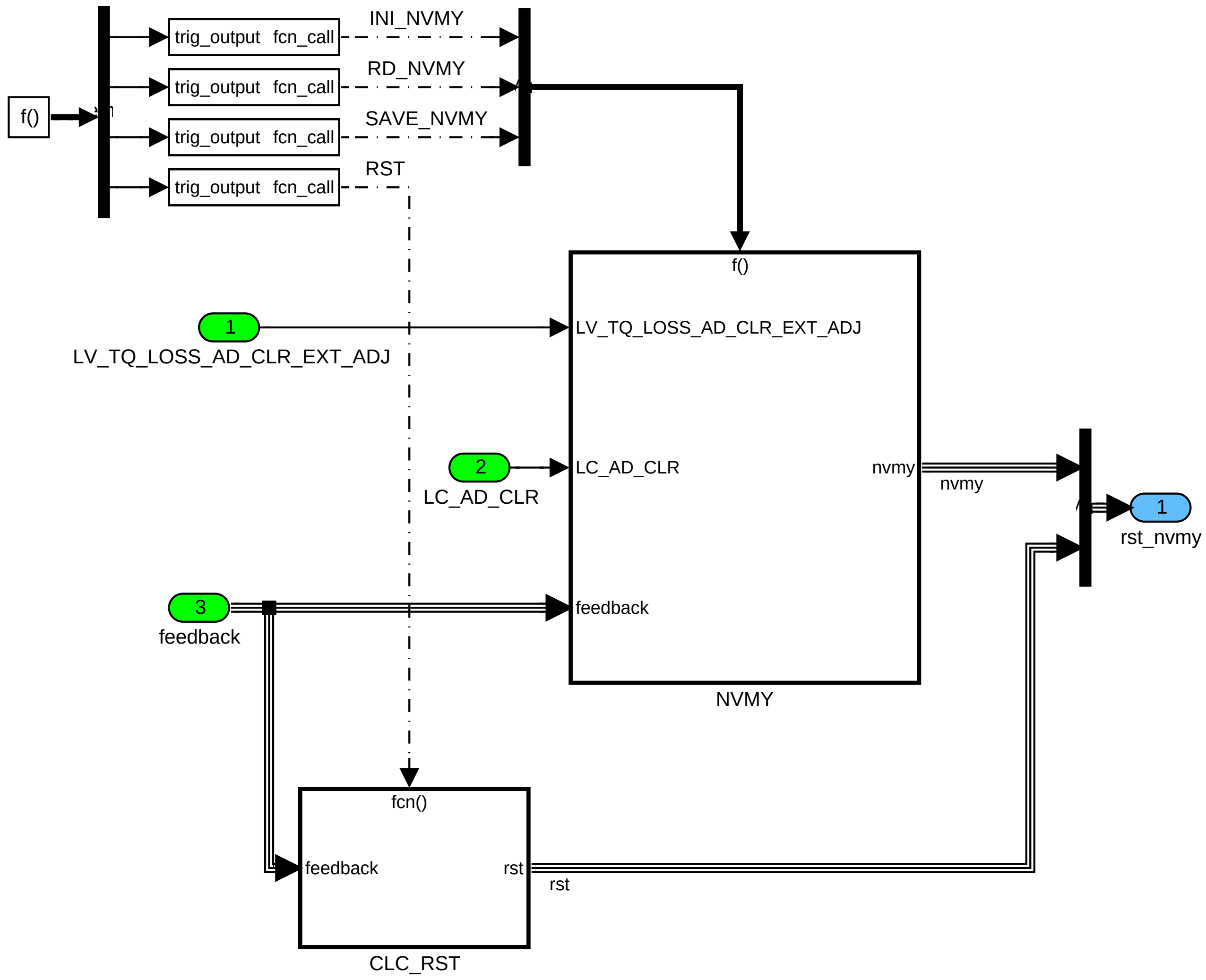
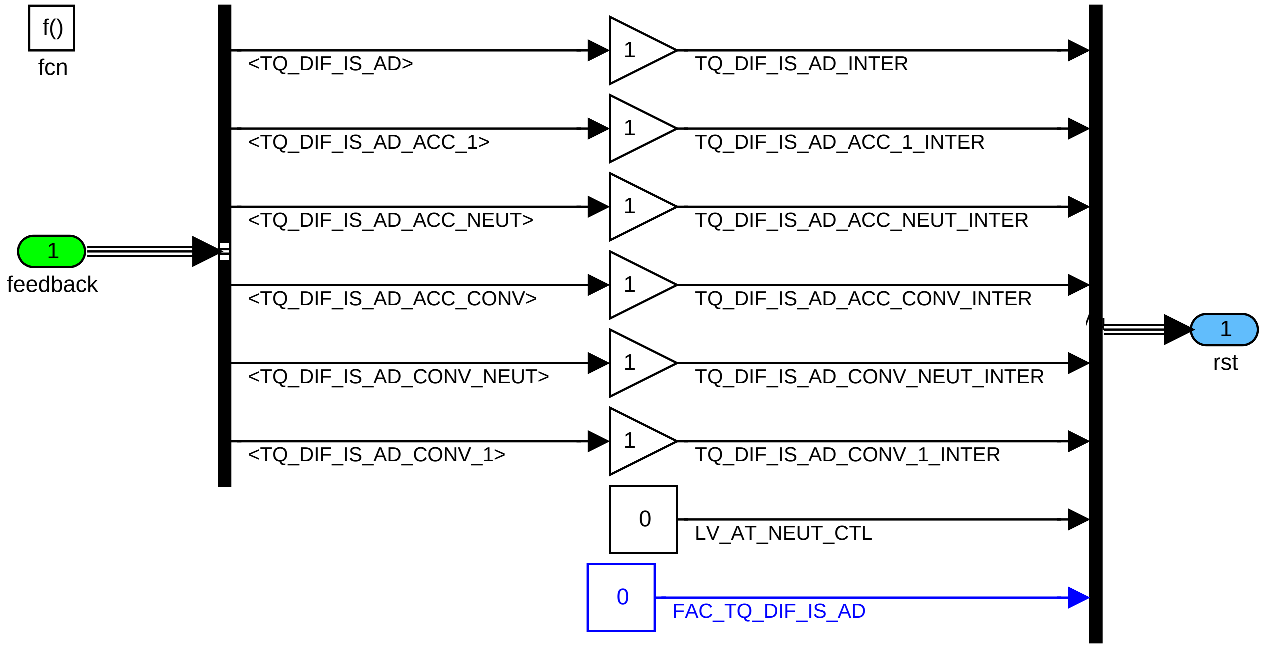


Figure 10 TQLO_M800N/ RST

Initialization after ECU Reset

The following variables are initialized at ECU Reset. Note that this has to be performed after the E2PROB has been read!



N.B.: The inititialization of the variables has to be performed after the E2PROM has been read!

Figure 11 TQLO_M800N/ RST/ CLC_RST

<u>Overview of NVMY handling</u>

Initialization, Saving and Reading adaptation values out of the E2PROM.

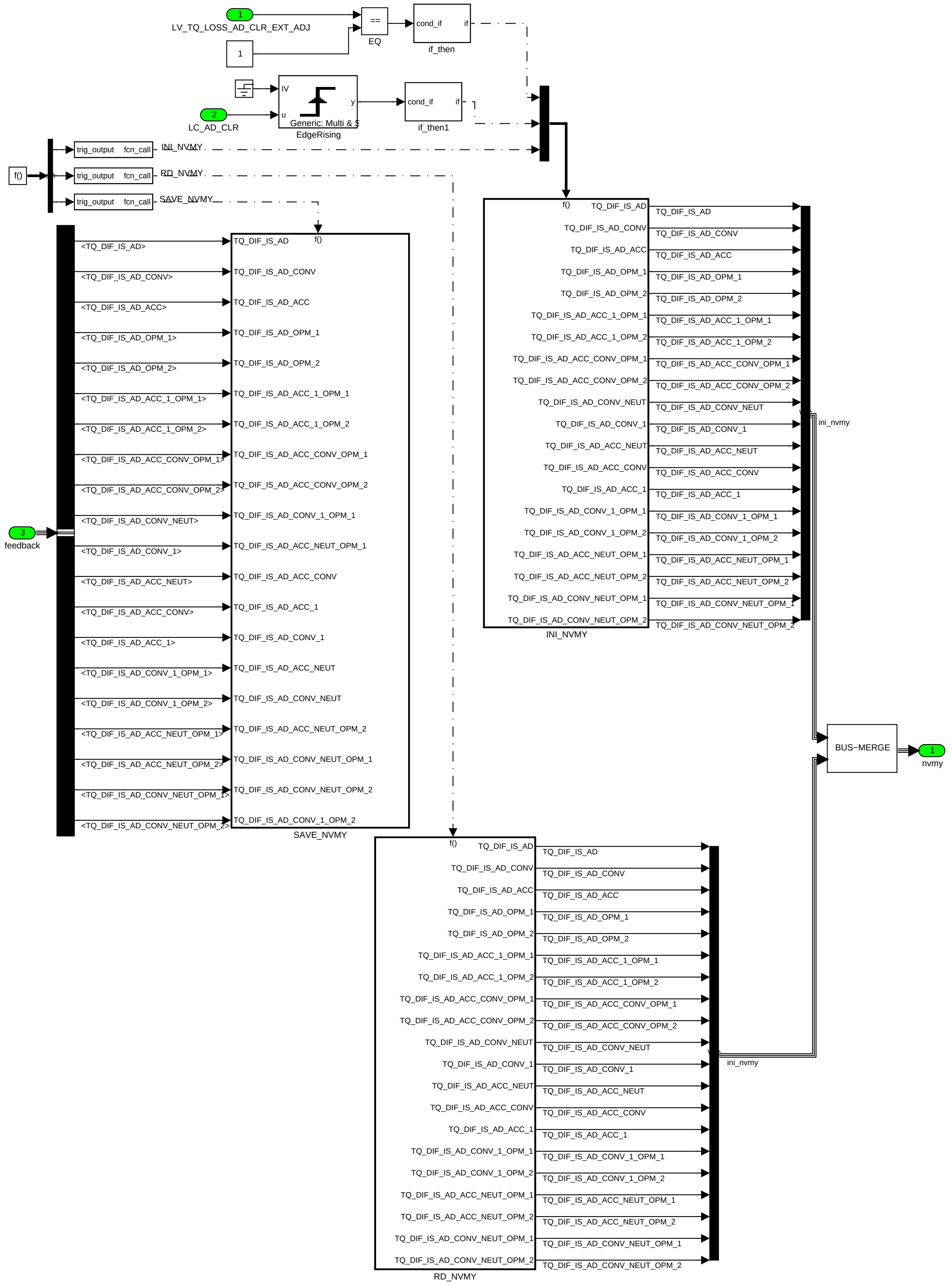


Figure 12 TQLO_M800N/ RST/ NVMY

Initialization of NVMY data

Initialization of the NVMY data in the E2PROM in case of:

- no adaptation values available
- checksum error in E2PROM sector
- cleared E2PROM sector by calibration system or service device

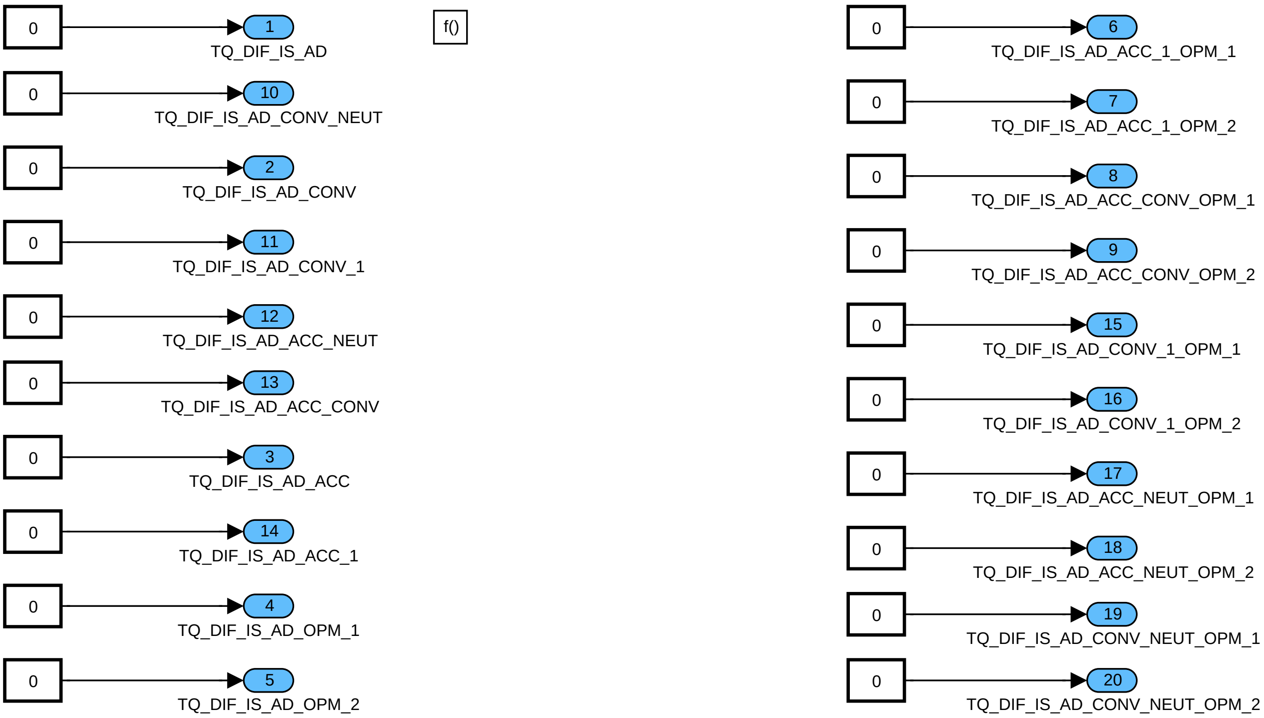


Figure 13 TQLO_M800N/ RST/ NVMY/ INI_NVMY

Reading of NVMY data

The following variables are read out of the NVMY after ECU Reset.

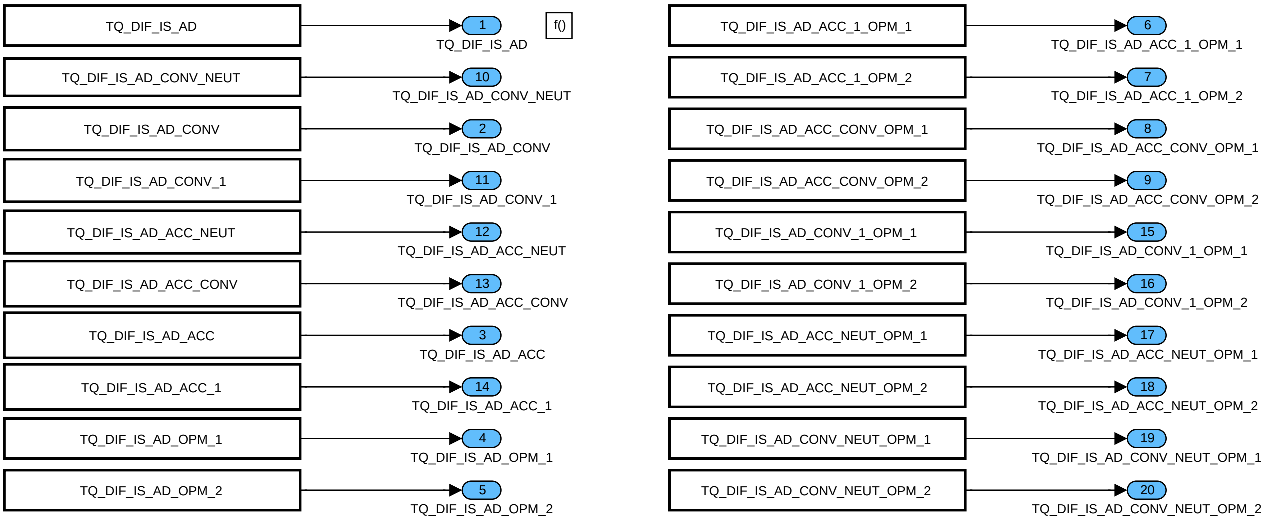


Figure 14 TQLO_M800N/ RST/ NVMY/ RD_NVMY

Saving of NVMY data

The following variables are stored in the NVMY.

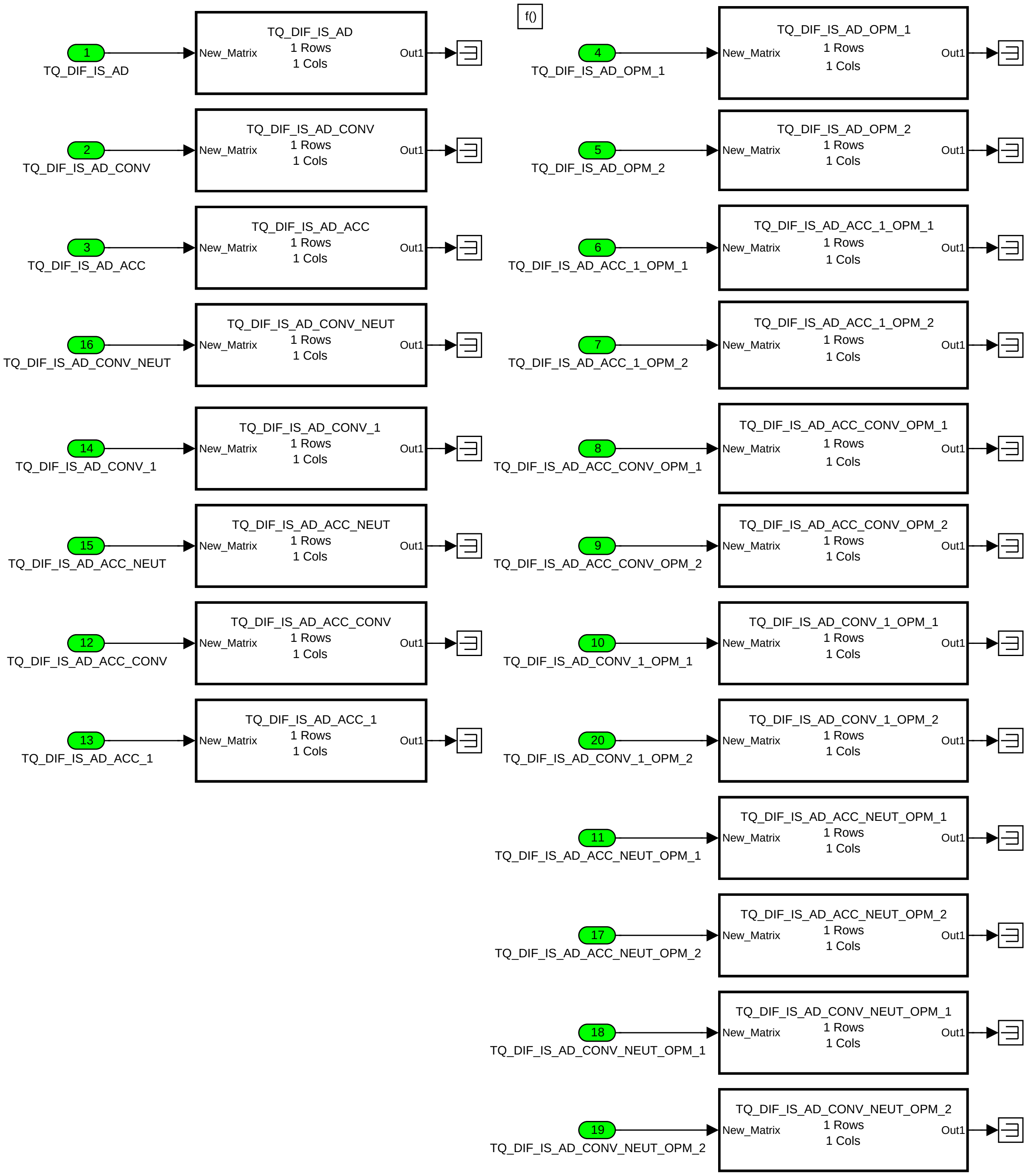


Figure 15 TQLO_M800N/ RST/ NVMY/ SAVE_NVMY

Bus Merge of the NVMY data

Either the data read out of the NVMY or the initialized values are used after ECU Reset.

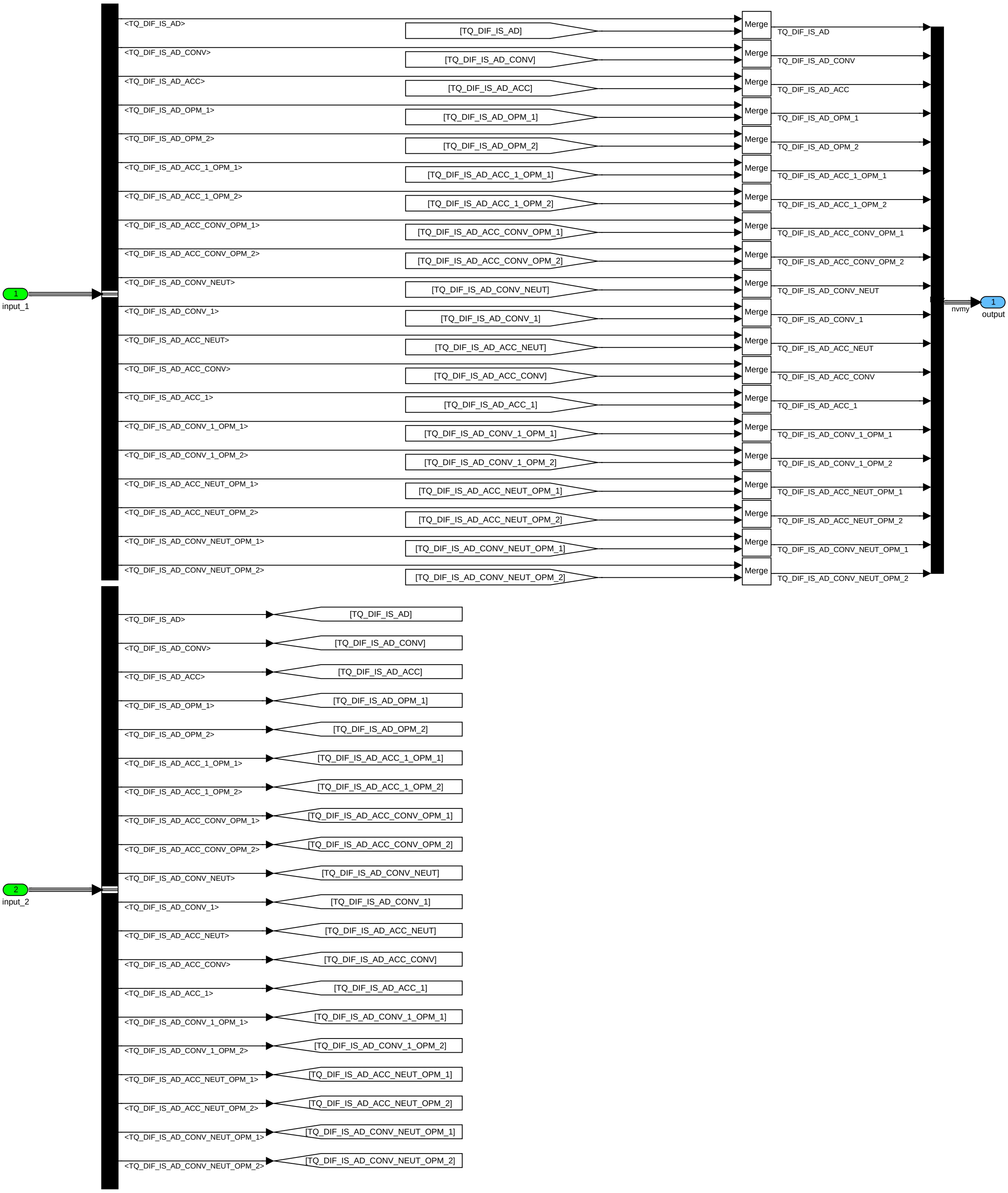


Figure 16 TQLO_M800N/ RST/ NVMY/ Bus_Merge

63.4.1.3 Functional overview of the adaptation algorithm

Depending of the status of several logical variables (LV_DRI, LV_ACCOUT_RLY, LV_AT_NEUT_CTL) and the automatic gearbox clutch information STATE_MOD_GB the different adaptation paths are chosen. With the help of the logical field STATE_MOD_GB the logical variable LV_AT_NEUT_CTL is created.

A filter first order is used to smooth the adaptation value changes in case of engaging/disengaging the drive w/o neutral control active and/or switching on/off the AC.

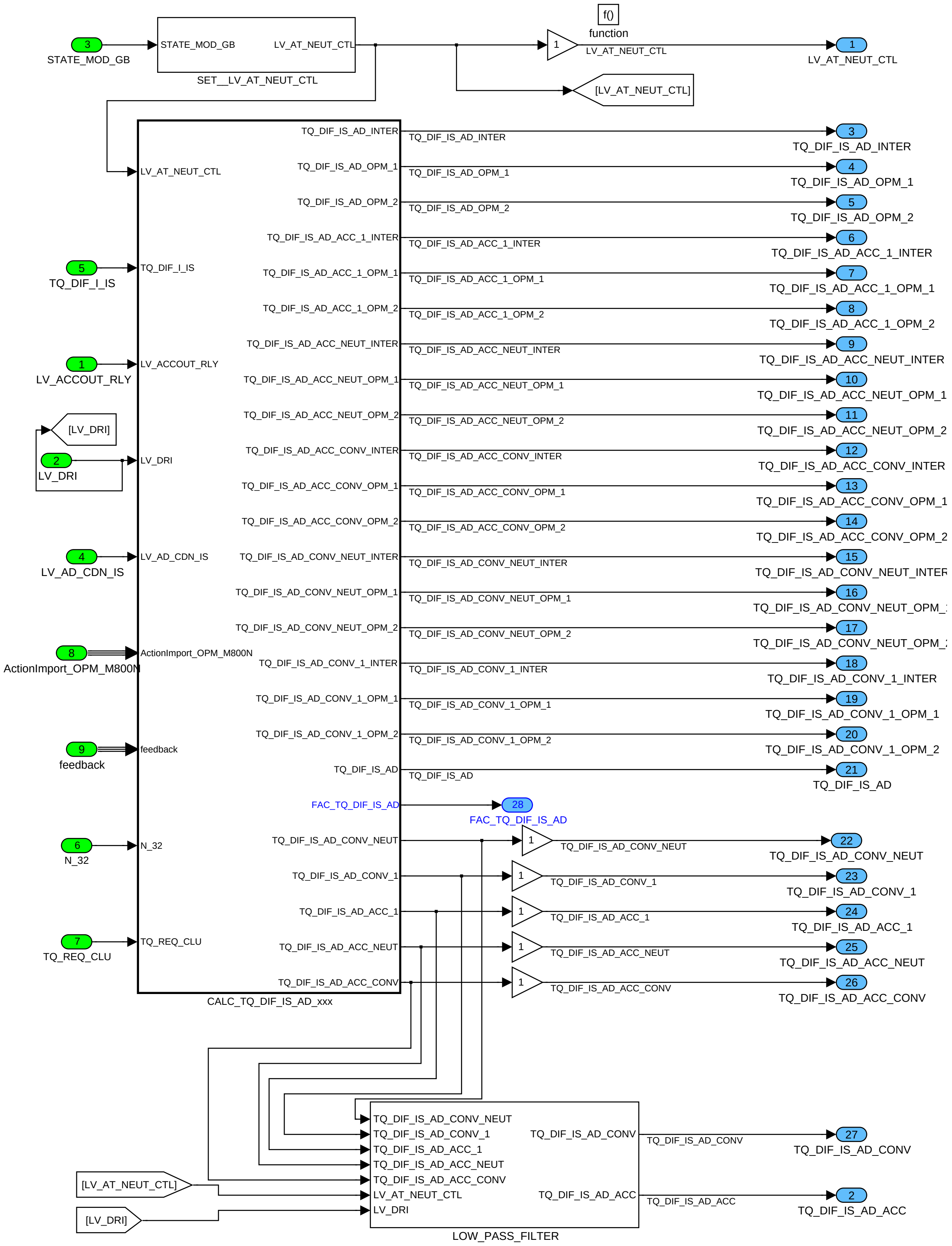


Figure 17 TQLO_M800N/ Operate

Setting of LV_AT_NEUT_CTL

With the help of the logical field STATE_MOD_GB the logical variable LV_AT_NEUT_CTL is created.

Depending on that information a separate set of adaptation values is chosen.

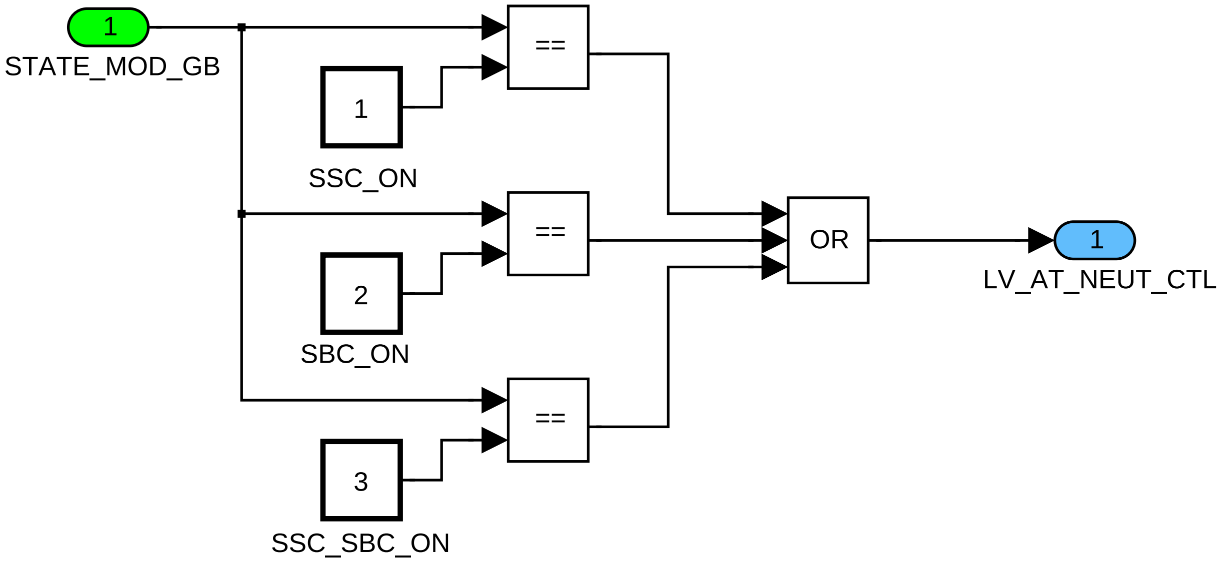


Figure 18 TQLO_M800N/ Operate/ SET__LV_AT_NEUT_CTL

Calculation of the different adaptation values TQ_DIF_IS_AD_XXX

Depending of the status of several logical variables (LV_DRI, LV_ACCOUT_RLY, LV_AT_NEUT_CTL) the different adaptation paths are chosen.

To ensure that the adaptation values don't change during the transition from OPM_1 to OPM_2 and vice versa the flags LV_OPM_SEL_TQ_IS_AD_1/_2 are taken into account. Only if LV_OPM_SEL_TQ_IS_AD_1 is set or LV_OPM_SEL_TQ_IS_AD_2 is set the corresponding integrator out value is routed through.

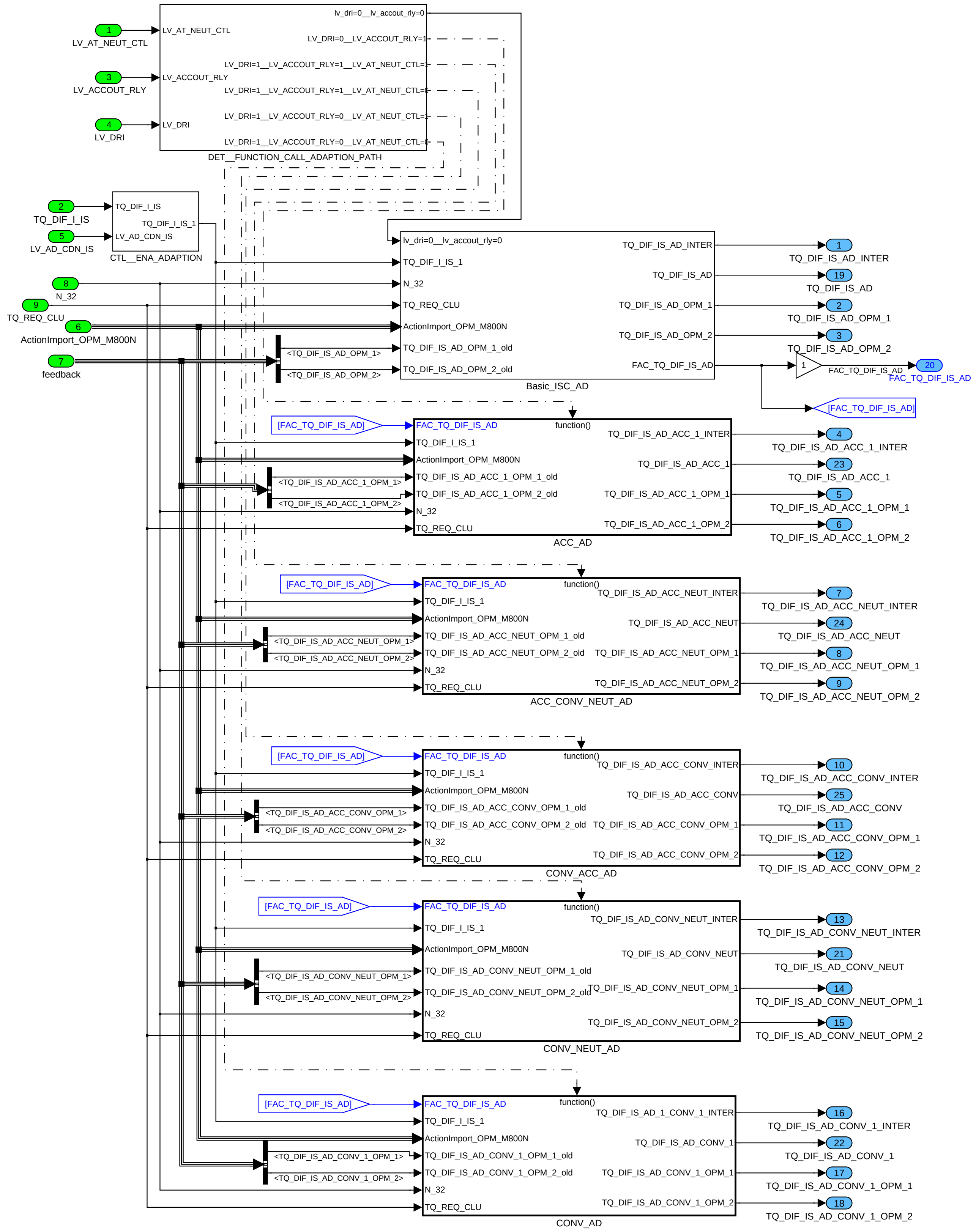


Figure 19 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx

Creation of the function calls for the several adaptation values

Depending on the logical status of the flags LV_DRI, LV_AT_NEUT_CTL and LV_ACCOUT_RLY the corresponding subsystem is triggered. So in total 6 different combinations are possible.

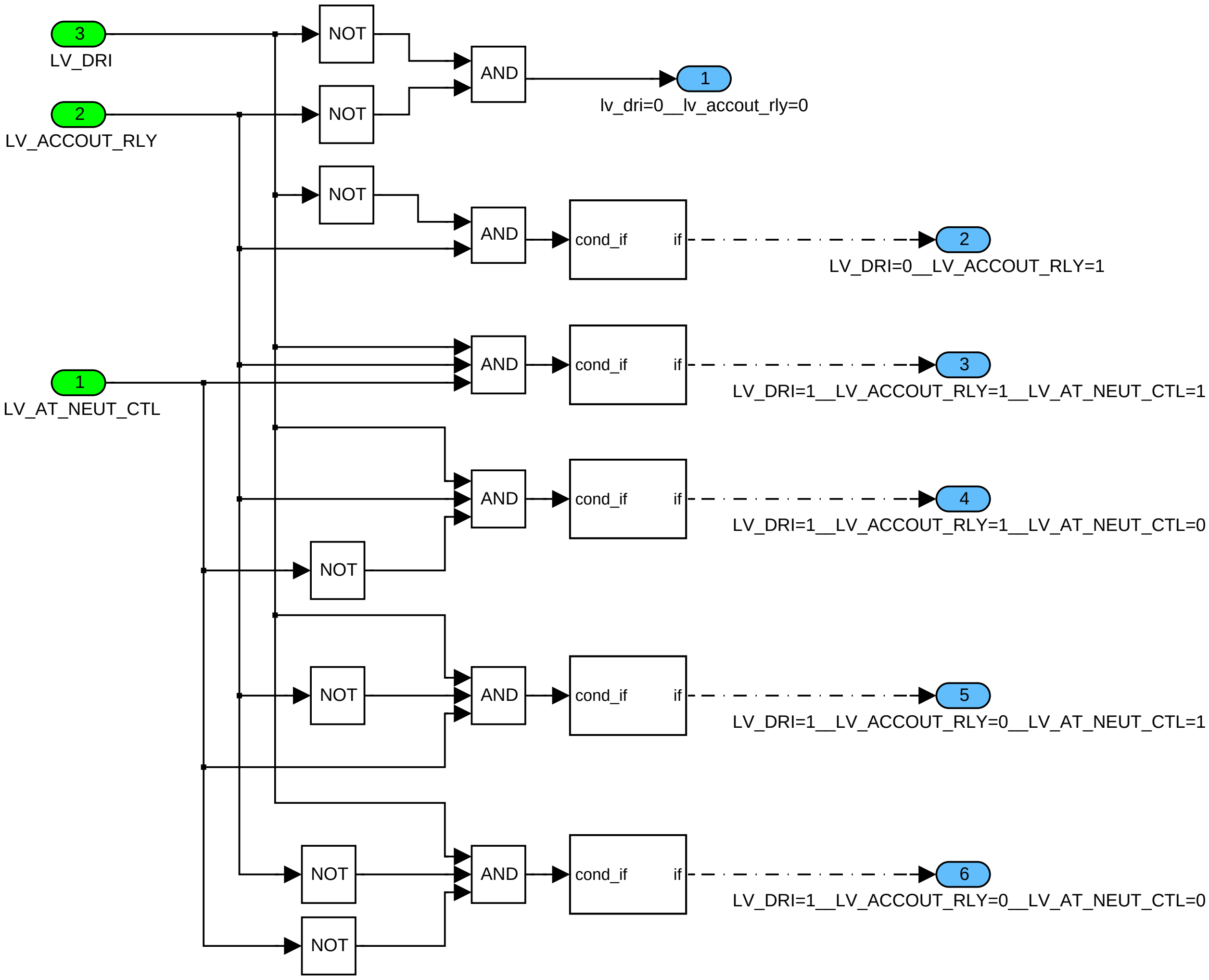
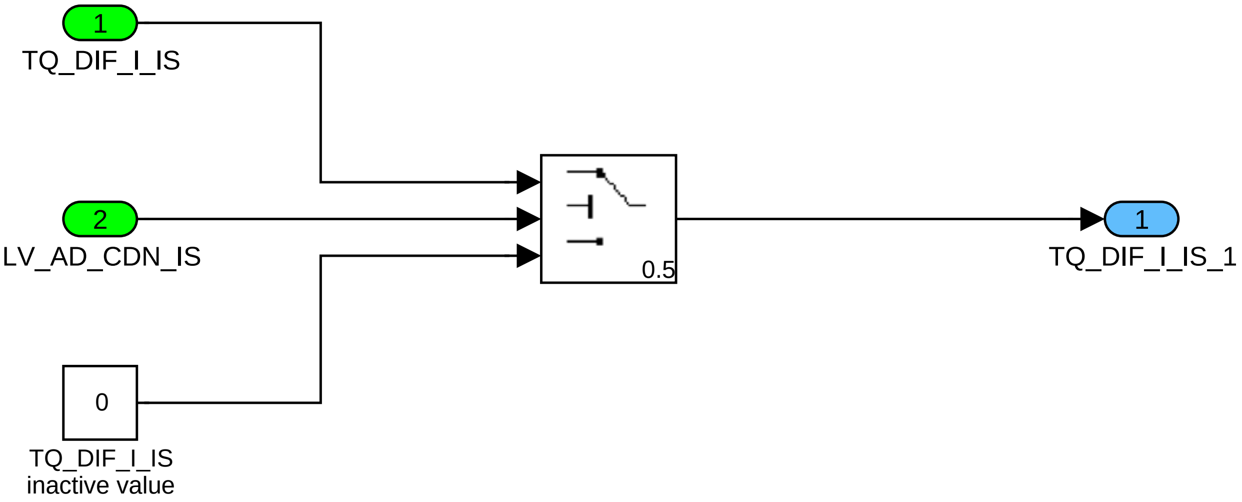


Figure 20 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx/ DET__FUNCTION_CALL_ADAPTION_PATH

Input for adaptation algorithm TQ_DIF_I_IS

The adaptation algorithm is fed by the I-gain of the idle speed controller. Due to the fact that the whole function is calculated continuously the input into the adaptation algorithm is controlled via the enable flag LV_AD_CDN_IS. If the adaptation condition is not fulfilled the passive value 0 Nm is routed into the adaptation algorithm. Thus no adaptation value change takes place.



This functions controls the input signal into the adatation algorithm
− if the adaptation conditions are not fullfilled the adaptation is fed by 0Nm
− if the adaptation gets active the adaptation is fed by the idle speed controller TQ_DIF_I_IS

Figure 21 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx/ CTL__ENA_ADAPTION

Basic adaptation value (TQ_DIF_IS_AD)

This adaptation value is applied for a MT or AT vehicle with drive not engaged.

The speed of the adaptation process is defined through the calibration parameter The speed of the adaptation process is defined through the parameter FAC_TQ_DIF_IS_AD and the function's update rate. This factor is calculated in the map IP_FAC_TQ_DIF_IS_AD, which is depend on the absolut value of TQ_DIF_I_IS. This provides a fast or a slow adaptation.

The sign of the input in the integrator function will be inverted because the output of the Subsystem is coordinated into TQ_LOSS [neg....0] Nm .

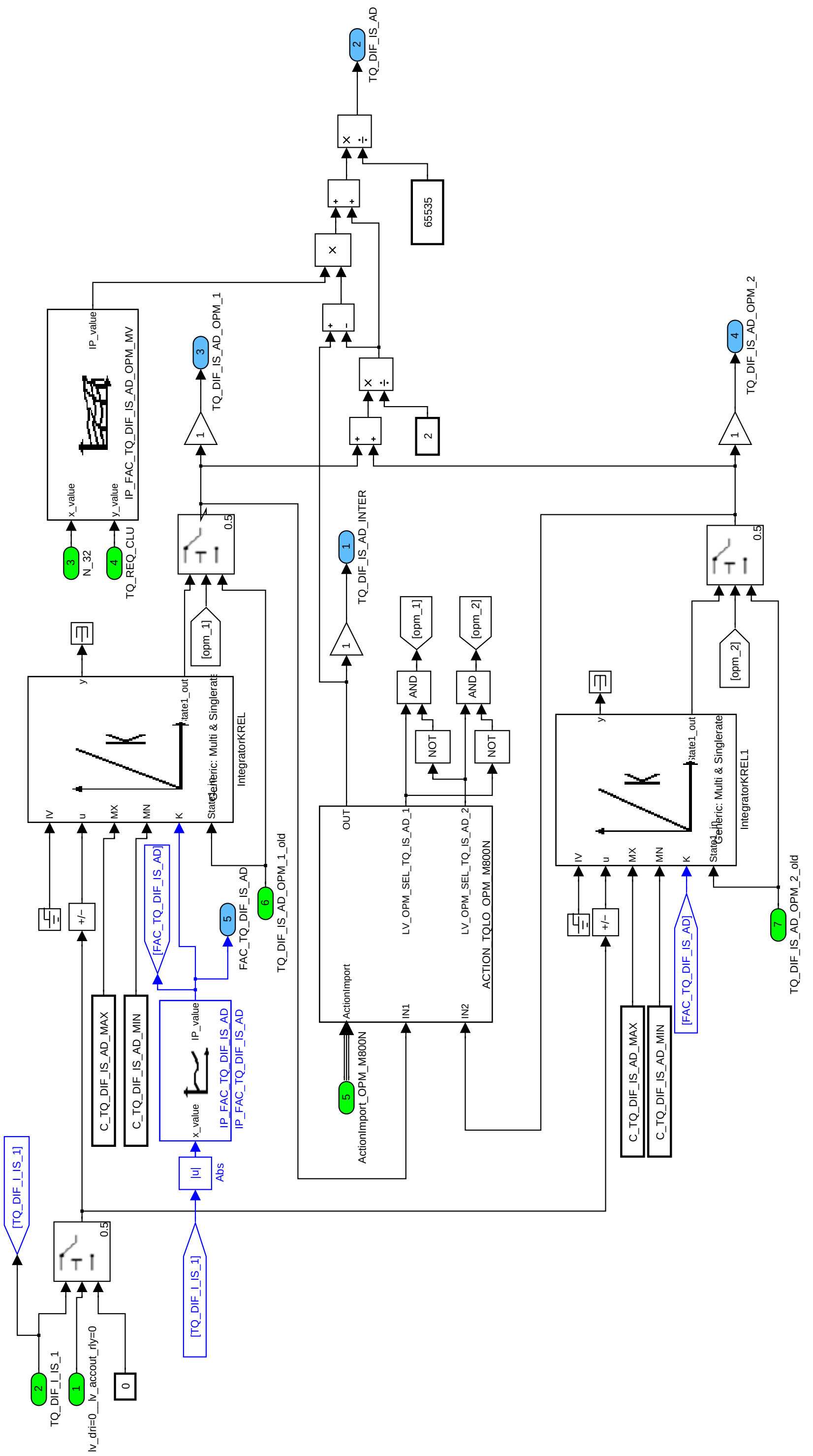


Figure 22 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx/ Basic_ISC_AD

<u>Adaptation value with AC on (TQ_DIF_IS_AD_ACC_1)</u>

This adaptation value is applied for a MT or AT vehicle with drive not engaged and AC ON.

The speed of the adaptation process is defined through the parameter FAC_TQ_DIF_IS_AD and the function’s update rate. This factor is calculated in the map IP_FAC_TQ_DIF_IS_AD, which is depend on the absolut value of TQ_DIF_I_IS. This provides a fast or a slow adaptation.

The sign of the input in the integrator function will be inverted because the output of the Subsystem is coordinated into TQ_LOSS_ACC [neg....0] Nm .

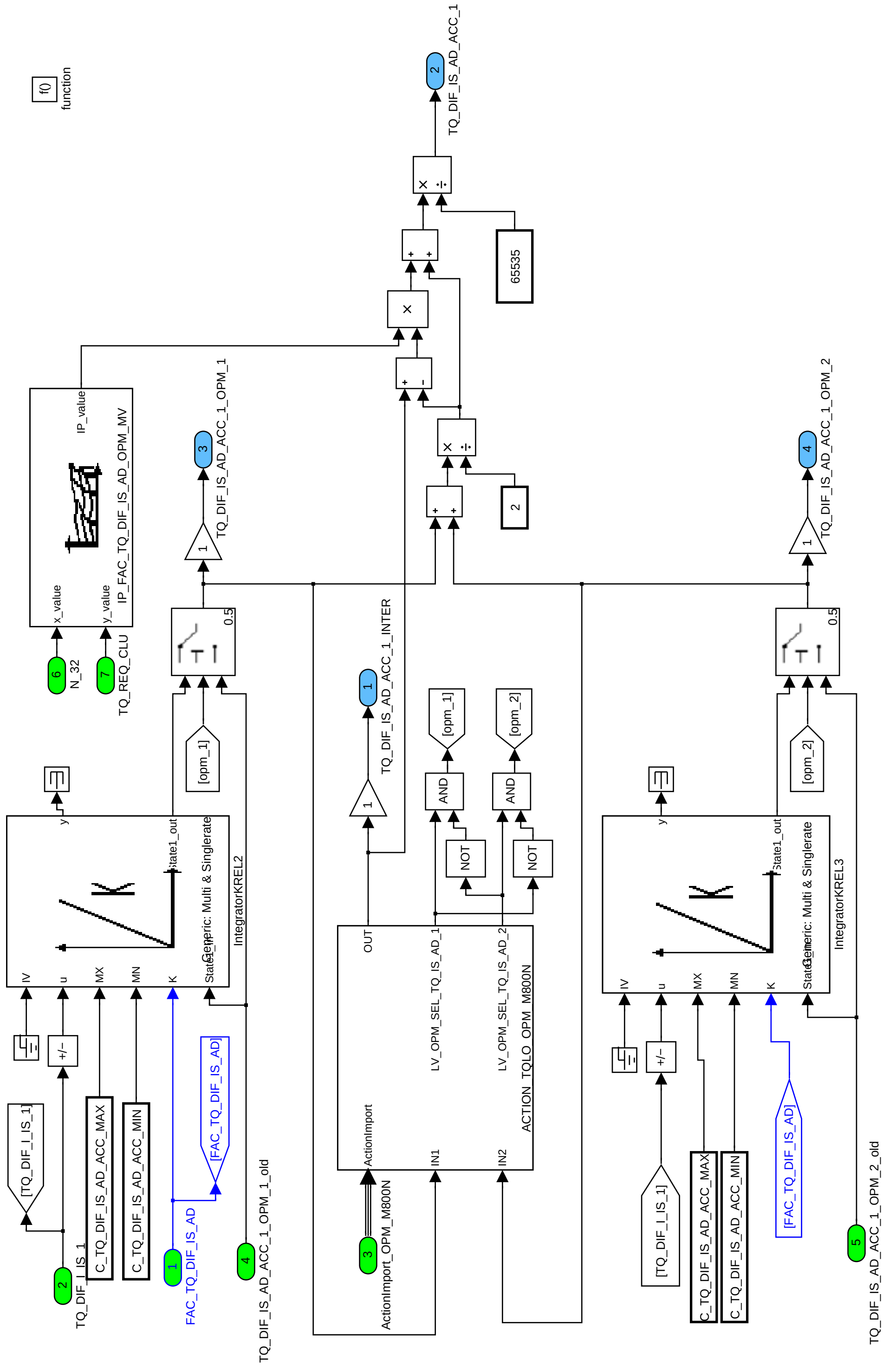


Figure 23 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx/ ACC_AD

Adaptation value with DRI engaged, Neutral control active and AC off (TQ_DIF_IS_AD_CONV_NEUT)

This adaptation value is applied for an AT vehicle with drive engaged, Neutral control active and AC OFF.

The speed of the adaptation process is defined through the parameter FAC_TQ_DIF_IS_AD and the function's update rate. This factor is calculated in the map IP_FAC_TQ_DIF_IS_AD, which is depend on the absolut value of TQ_DIF_I_IS. This provides a fast or a slow adaptation.

The sign of the input in the integrator function will not be changed because the output of the Subsystem is coordinated into TQ_CONV [0...pos.] Nm .

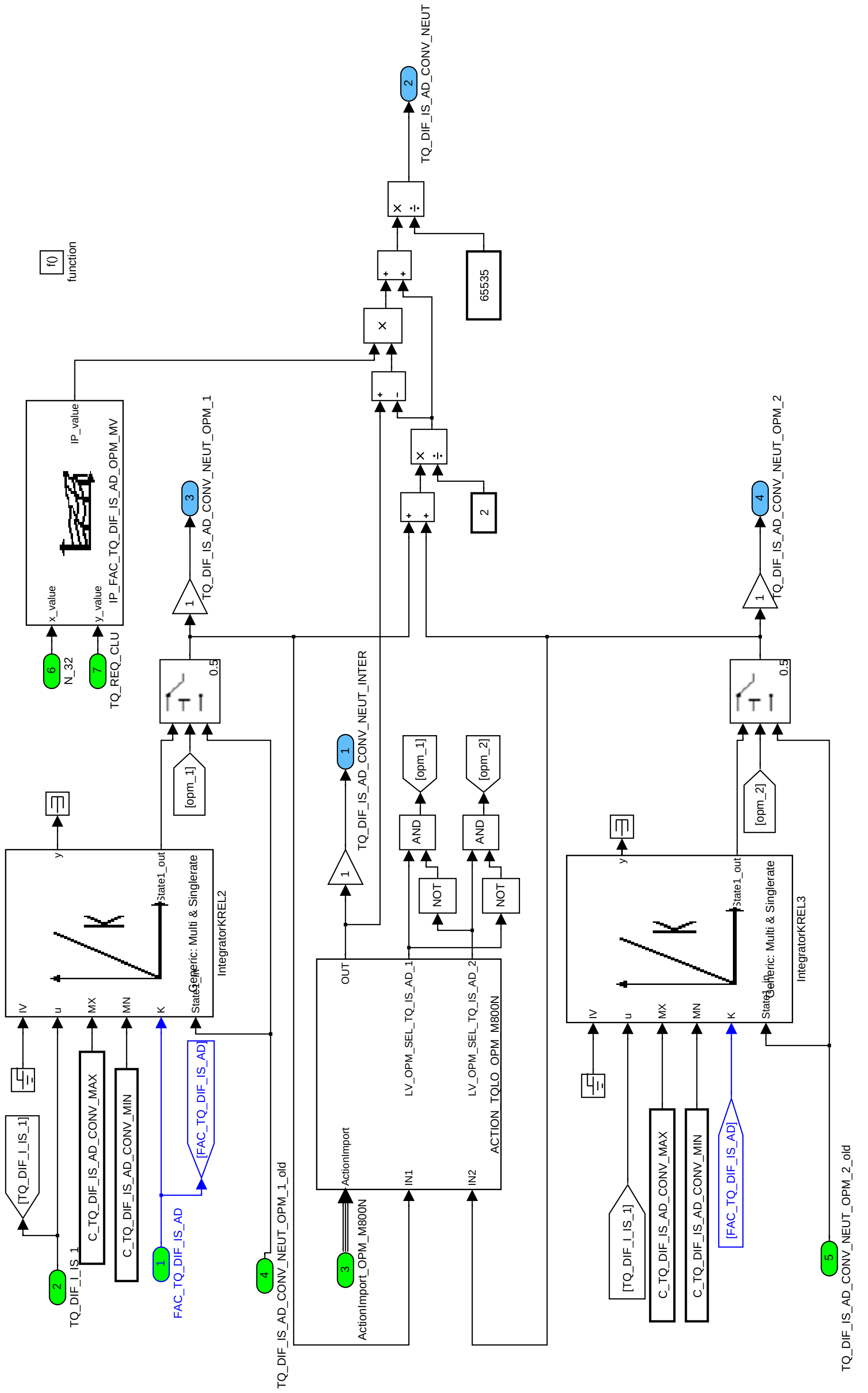


Figure 24 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx/ CONV_NEUT_AD

Adaptation value with DRI engaged, Neutral control passive and AC on (TQ_DIF_IS_AD_ACC_CONV)

This adaptation value is applied for an AT vehicle with drive engaged, Neutral control passive and AC ON.

The speed of the adaptation process is defined through the calibration parameter The speed of the adaptation process is defined through the parameter FAC_TQ_DIF_IS_AD and the function's update rate. This factor is calculated in the map IP_FAC_TQ_DIF_IS_AD, which is depend on the absolut value of TQ_DIF_I_IS. This provides a fast or a slow adaptation.

The sign of the input in the integrator function will be inverted because the output of the Subsystem is coordinated into TQ_LOSS_ACC [neg....0] Nm .

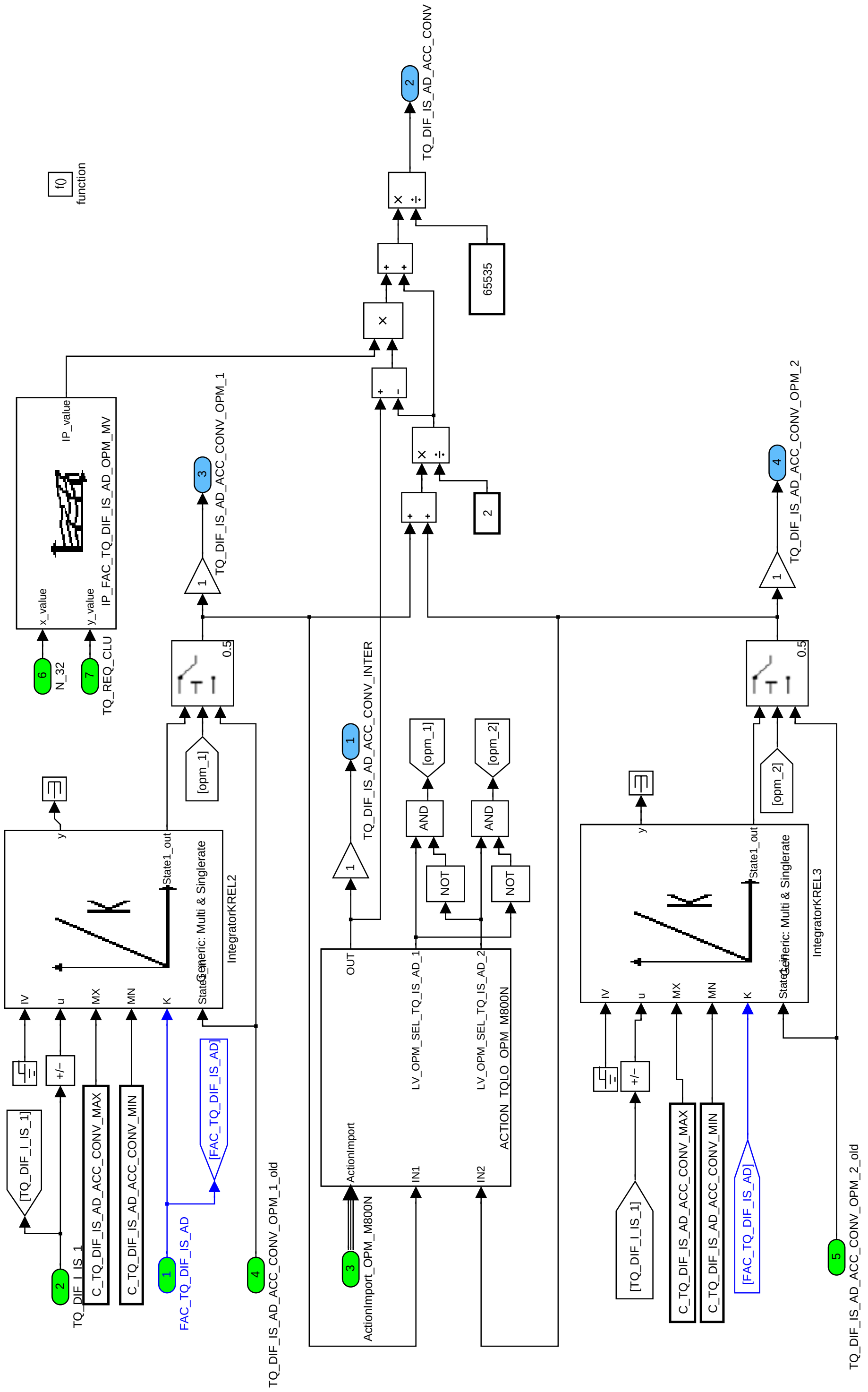


Figure 25 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx/ CONV_ACC_AD

Adaptation value with DRI engaged, Neutral control active and AC on (TQ_DIF_IS_AD_ACC_NEUT)

This adaptation value is applied for an AT vehicle with drive engaged, Neutral control active and AC ON.

The speed of the adaptation process is defined through the calibration parameter The speed of the adaptation process is defined through the parameter FAC_TQ_DIF_IS_AD and the function's update rate. This factor is calculated in the map IP_FAC_TQ_DIF_IS_AD, which is depend on the absolut value of TQ_DIF_I_IS. This provides a fast or a slow adaptation.

The sign of the input in the integrator function will be inverted because the output of the Subsystem is coordinated into TQ_LOSS_ACC [neg....0] Nm .

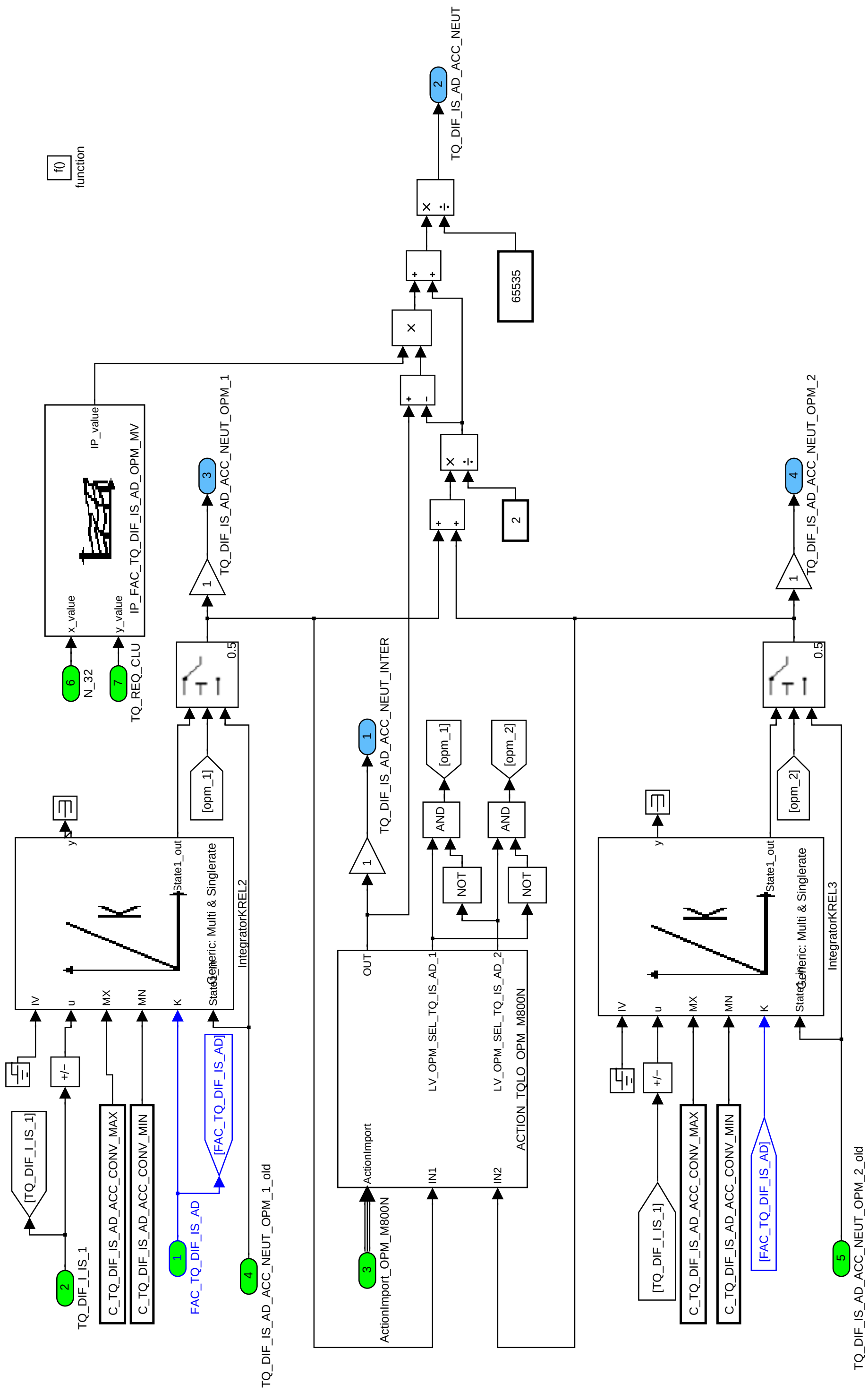


Figure 26 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx/ ACC_CONV_NEUT_AD

<u>Adaptation value with DRI engaged, Neutral control passive and AC off (TQ_DIF_IS_AD_CONV_1)</u>

This adaptation value is applied for an AT vehicle with drive engaged, Neutral control passive and AC OFF.

The speed of the adaptation process is defined through the parameter FAC_TQ_DIF_IS_AD and the function’s update rate. This factor is calculated in the map IP_FAC_TQ_DIF_IS_AD, which is depend on the absolut value of TQ_DIF_I_IS. This provides a fast or a slow adaptation.

The sign of the input in the integrator function will not be changed because the output of the Subsystem is coordinated into TQ_CONV [0...pos.] Nm .

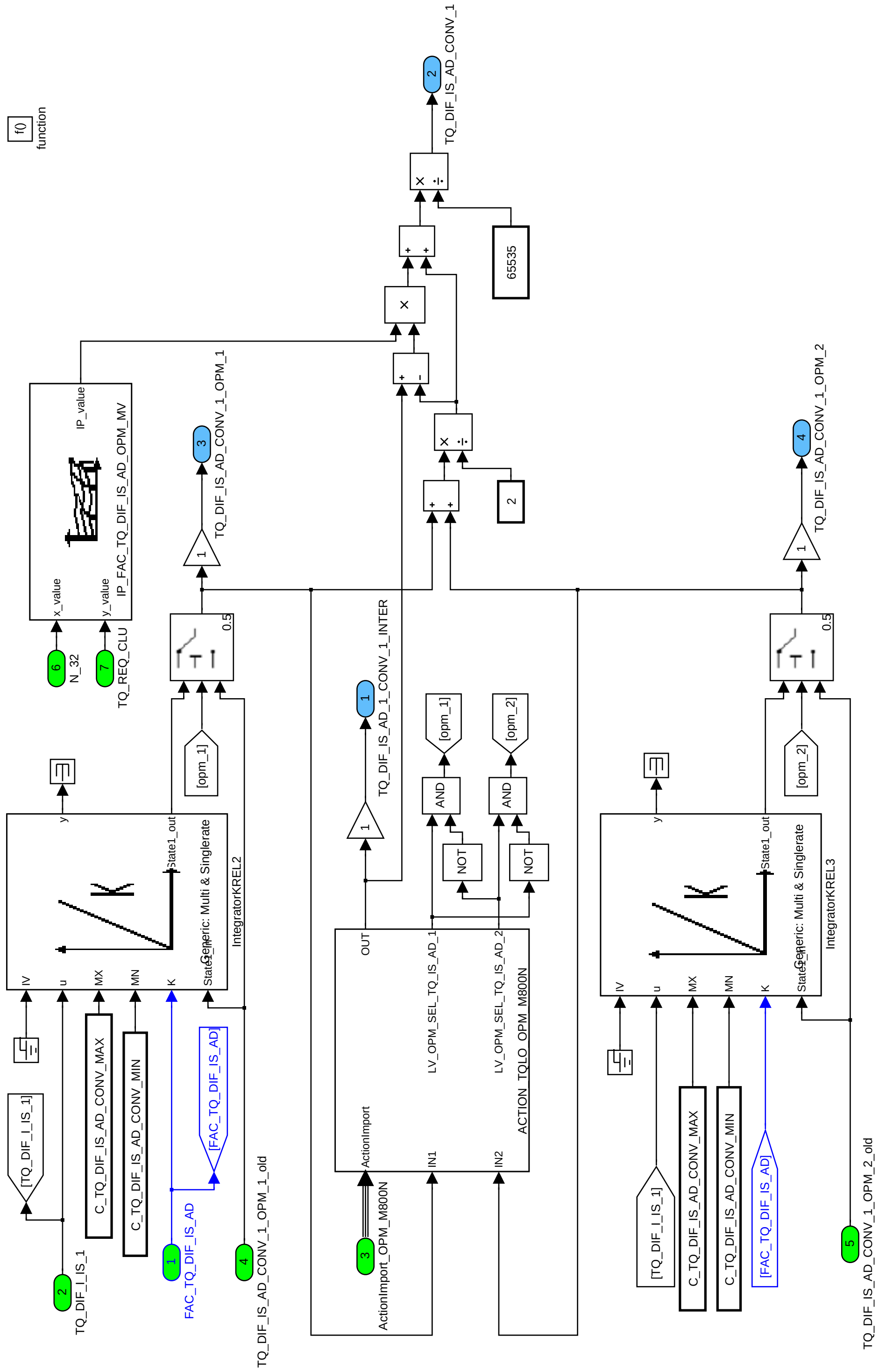
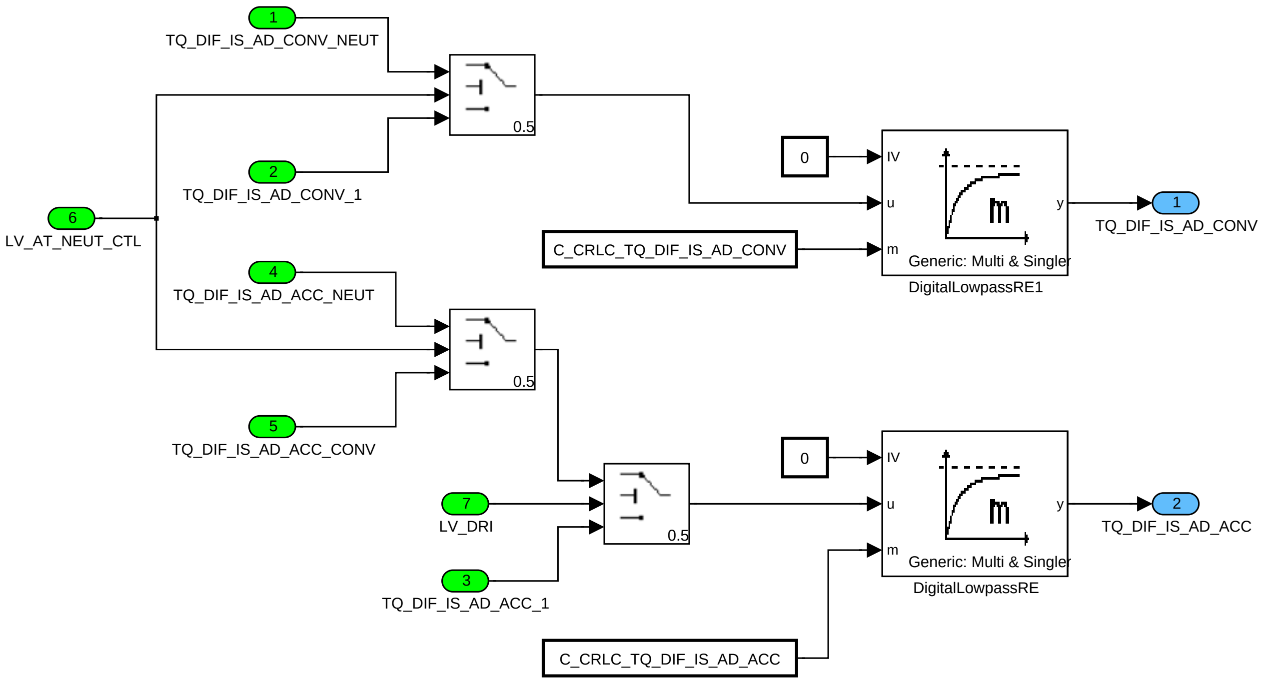


Figure 27 TQLO_M800N/ Operate/ CALC_TQ_DIF_IS_AD_xxx/ CONV_AD

Filtering of the adaptation values

In order to be able to use one input for the ACC torque losses module and one adaptation value for the converter torque calculation module, switches are introduced to change between TQ_DIF_IS_AD_ACC_1, TQ_DIF_IS_AD_ACC_CONV and TQ_DIF_IS_AD_ACC_NEUT in dependency of the status of the logical inputs. To prevent abrupt changes in TQ_DIF_IS_AD_ACC (in case of switching between TQ_DIF_AD_IS_ACC_CONV to TQ_DIF_AD_IS_ACC_NEUT or the TQ_DIF_IS_AD_ACC_1), the output value of the final switch is smoothed by a low pass filter. The same strategy is applied in case of a change of the status of the neutral control. The change between TQ_DIF_IS_AD_CONV_NEUT and TQ_DIF_IS_AD_CONV_1 is also routed via a low pass filter.

During the first ACC adaptations, TQ_DIF_IS_AD_ACC may be erroneous and should therefore be limited to small values.



N.B.: This Module has to be calculated even if LV_AD_CDN_IS == 0;
if the adapation is disabled but drive is engaged/disengaged //AC
switched on/off the calculation of the corresponding torque adaptation
value has to be performed!

Figure 28 TQLO_M800N/ Operate/ LOW_PASS_FILTER

63.4.1.4 Output Manager

The Output Manager merges the signals of the initialization and the normal operating system.

Figure 29 TQLO_M800N/ SubSystem

63.5 Cus adap module: TQLO

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TQ_LOSS_OPM_SEL	O/V	0...FFH	0...0.99609375	0.00390625	-
Torque loss interpolation factor for opeartion switch manager					
Md_na_defrost	O/V	8000...7FFFH	-3.2768E+3 ... 3.2767E+3	0.1	Nm
torque losses for defroster means					
Md_na_mlad	O/V	8000...7FFFH	-3.2768E+3 ... 3.2767E+3	0.1	Nm
torque losses for mechanical charger					
Md_na_thermo	O/V	8000...7FFFH	-3.2768E+3 ... 3.2767E+3	0.1	Nm
torque losses for thermostat					
TQ_LOSS_REQ_CLU	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Overall torque losses (filtered value)					
TQ_LOSS	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Overall torque losses (unfiltered value)					
TQ_LOSS_ADD	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
total torque losses for additive engine devices					

Input data:

TQ_LOSS_SCHA	TQ_LOSS_THERMO	TQ_LOSS_WIN_HEAT	Md_reib_sa
Mdi_reib	Md_na_ges		

63.5.1 General Information

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application Condition

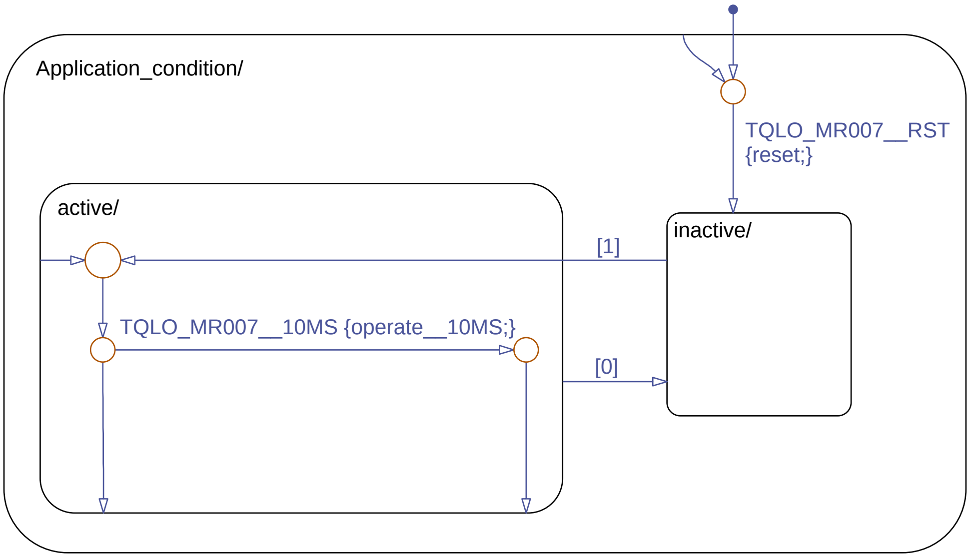


Figure 30 TQLO_MR007/ APP_CDN/ APPCND

Function Description

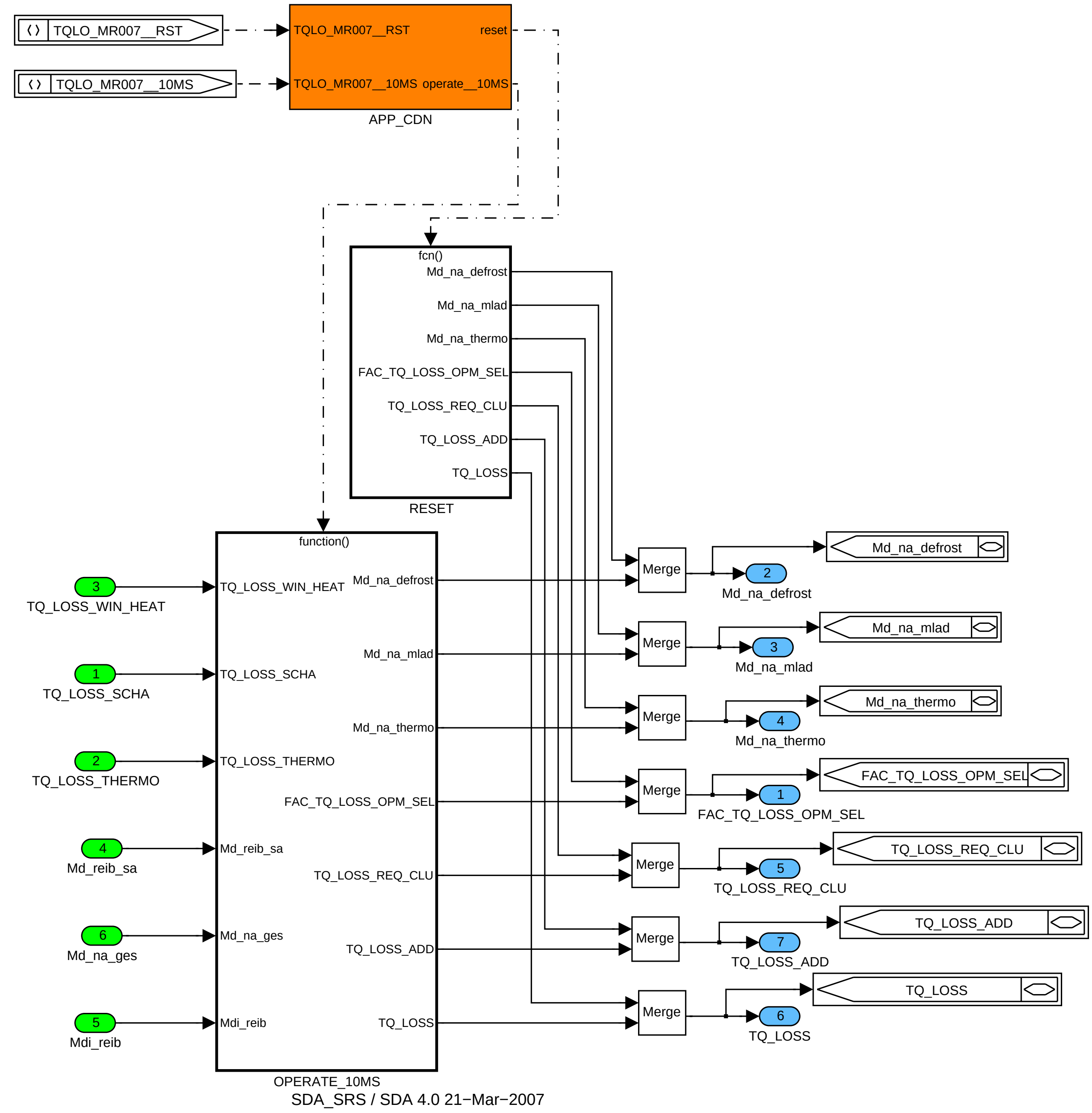


Figure 31 TQLO_MR007

63.5.1.1 Recurence: 10 ms

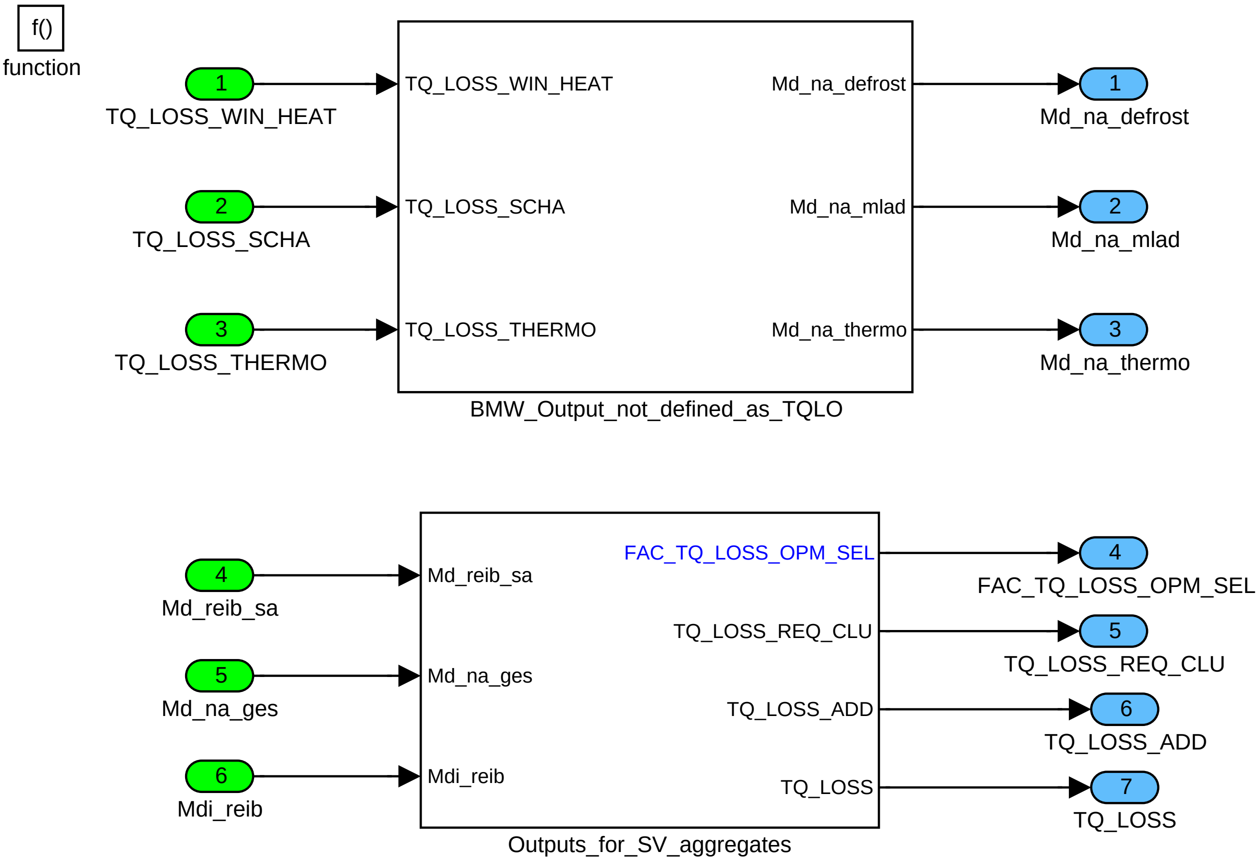


Figure 32 TQLO_MR007/ OPERATE_10MS

BMW output not defined as TQLO

Outputs for BMW functions which are not defined as TQLO exported data

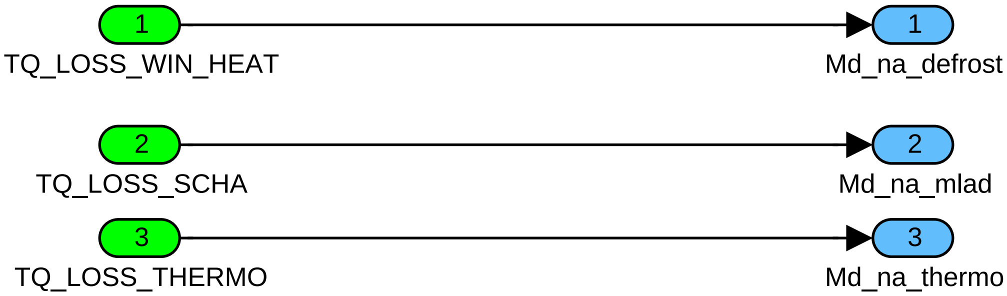


Figure 33 TQLO_MR007/ OPERATE_10MS/ BMW_Output_not_defined_as_TQLO

Outputs for SV aggregates

Remark: all formulas are valid in a physical meaning.

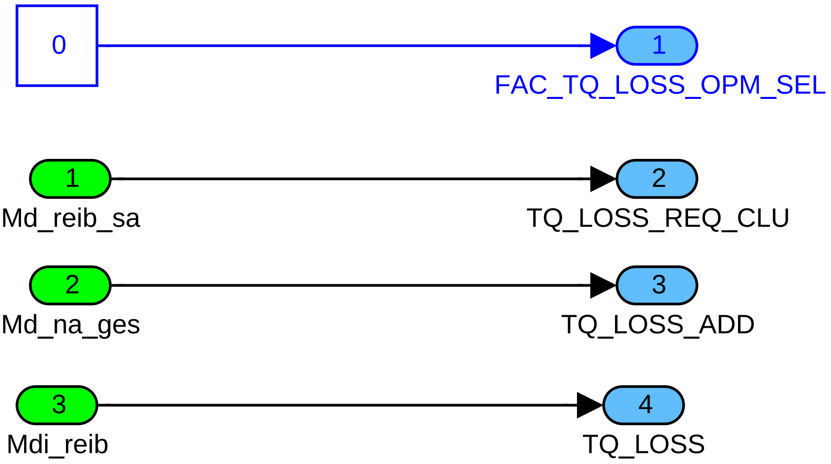


Figure 34 TQLO_MR007/ OPERATE_10MS/ Outputs_for_SV_aggregates

63.5.1.2 Reset

All outputs are set to zero.

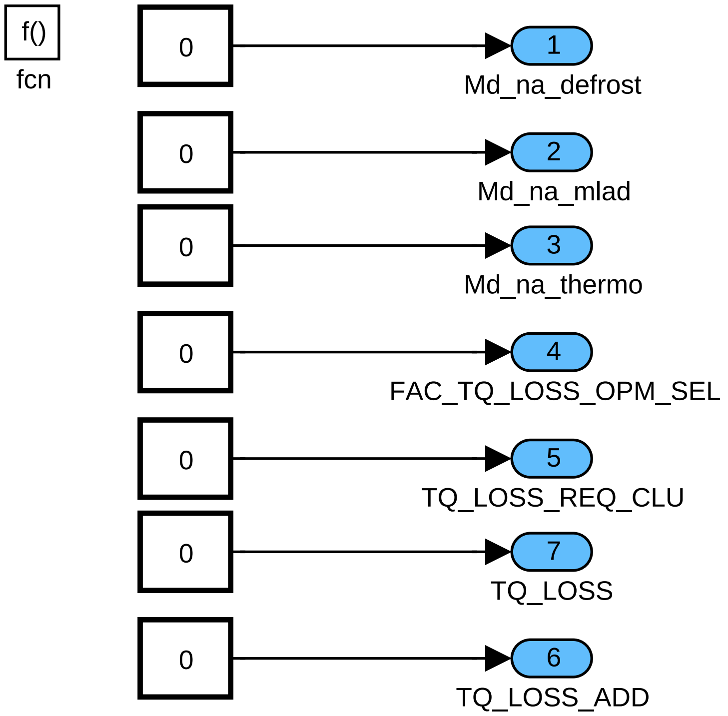


Figure 35 TQLO_MR007/ RESET

64 Torque setpoint

Table of Contents

Index

64.1 Detection of Driver or Cruise control passive (No TQ demand).

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CT	V/O	0...1H	0...1	1	[-]
Logical variable for detecting driver request passive					

Input data:

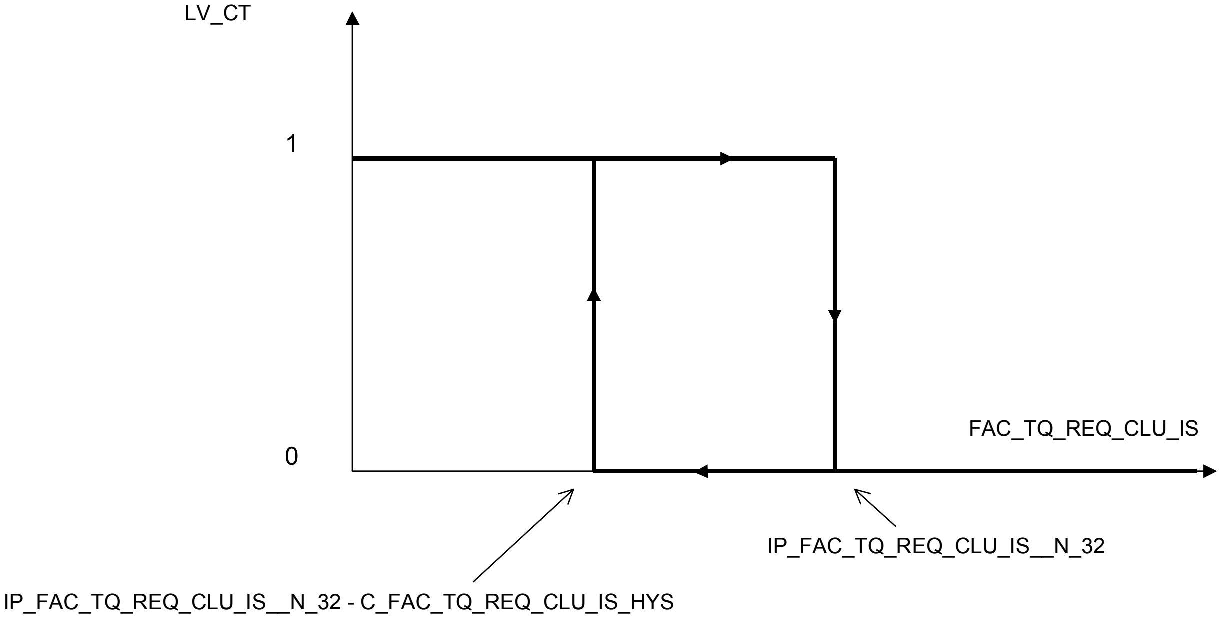
LV_MTC_CUR_OFF	FAC_TQ_REQ_CLU_LDM	N_32	

General information:

The detection of LV_CT = 0 (no torque request from driver or cruise control or LDM) is done by evaluating the torque scaling factor FAC_TQ_REQ_CLU_LDM (driver / cruise / LDM).
In case of LV_MTC_CUR_OFF = 1 (no more throttle control), the throttle will remain in the fixed limp home position, which is around 8° (depending on the used throttle); this is a part load position (LV_CT=0). All torque requests can only be realized by spark advance interaction or single cylinder cut off coordination. For this purpose the driver passive detection is deactivated (LV_CT=0).

Formula section:

```
If      LV_MTC_CUR_OFF = 1
Then    LV_CT = 0
Else    if     LV_CT = 1
        Then   if     FAC_TQ_REQ_CLU_LDM ≥ IP_FAC_TQ_REQ_CLU_IS__N_32
               Then   LV_CT = 0
               Endif
        Else   if     (FAC_TQ_REQ_CLU_LDM ≤ IP_FAC_TQ_REQ_CLU_IS__N_32
                                          - C_FAC_TQ_REQ_CLU_IS_HYS )
               Then   LV_CT = 1
               Endif
        Endif
Endif
```



Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_TQ_REQ_CLU_IS__N_32	0	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_N_32__FAC_TQ_REQ_CLU_IS	0	1...FFH	32...8160	32	[rpm]
Threshold on FAC_TQ_REQ_CLU for IS detection					
C_FAC_TQ_REQ_CLU_IS_HYS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
Hysteresis on threshold on FAC_TQ_REQ_CLU for IS detection					

64.2 Basic functions: FL, INF & REACX

64.2.1 Full Load: (LV_FL)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_FL	O/V	0...1H	0...1	1	-
Basic function FL					
LV_FL_RAW	O/V	0...1H	0...1	1	-
LV condition “pedal value” is fullfilled					

Input data:

FAC_TQ_REQ_DRIV	N_32	TCO	VS
N	LV_AT	T_AST	CL_MMV
LOAD_THD_FL	LV_CT	STATE_ETC_LIH	STATE_TPS_DIAG
LV_TI_LIM_ACT	LV_TQ_LIM_INTV		

FUNCTION DESCRIPTION:

General information:

Main parameter for full load is an engine speed dependent load threshold. Additional parameters to disable full load are several failure conditions, coolant temperature, engine speed and vehicle speed. After fullfillment of all other conditions, the state LV_FL can be suppressed for a certain time.

Application conditions:

Activation :	at every engine operating state
Deactivation :	-
Initialization :	LV_FL and LV_FL_RAW = 0 LV_FL_1 and LV_FL_2 and LV_FL_3 = 0
Update rate :	10 ms

Formula section:

Intermediary values:

check pedal value for full load request, including hysteresis:

```
if  FAC_TQ_REQ_DRIV  ≥  IP_FAC_TQ_REQ_DRIV_FL
  then   LV_FL_1  =  1
  else
    if  FAC_TQ_REQ_DRIV  <  IP_FAC_TQ_REQ_DRIV_FL - C_FAC_TQ_REQ_FL_HYS
      then   LV_FL_1  =  0
      else   LV_FL_1  remains unchanged
    endif
endif
```

check requested torque setpoint for full load request, including hysteresis:

```
if  LOAD_THD_FL  ≥  IP_LOAD_THD_FL
  then   LV_FL_2  =  1
  else
    if  LOAD_THD_FL  <  IP_LOAD_THD_FL - C_LOAD_THD_FL_HYS
      then   LV_FL_2  =  0
      else   LV_FL_2  remains unchanged
    endif
endif
```

check conditions that prevent full load:

```
if  LV_CT = 1                                                     or
    STATE_ETC_LIH ≠ ETC_NO_LIH  and  STATE_ETC_LIH ≠ ETC_LIH_1  or
    STATE_TPS_DIAG = TPS_LIH_MAX                                  or
    LV_TI_LIM_ACT = 1
    or
    LV_TQ_LIM_INTV = 1
  then   LV_FL_3  =  0
  else   LV_FL_3  =  1
endif
```

```
if       LV_FL_1 = 1   and   LV_FL_2 = 1   and   LV_FL_3 = 1 and
         TCO  ≥  C_TCO_MIN_FL
  then   LV_FL_RAW  =  1
  else   LV_FL_RAW  =  0
endif
```

Full load recognition:

```
if  LV_FL_RAW  =  1
 then
        check engine/ vehicle conditions for full load:
        if  ( CL_MMV ≤      C_CL_MMV_THD_FL  and
              N_32    ≥  C_N_MIN_FL              and
              VS      ≥  C_VS_MIN_FL )   at least one time since reset        or

            ( CL_MMV >      C_CL_MMV_THD_FL  and
              N_32    ≥  C_N_MIN_CP_FL           and
              VS      ≥  C_VS_MIN_FL )   at least one time since reset

        then  start timer T_TI_DLY_FL if not yet started
              check full load time delay:
              if  C_N_MIN_FL_DLY ≤    N_32    <  C_N_MAX_FL_DLY  and
                  C_T_MIN_FL_DLY ≤    T_AST   <  C_T_MAX_FL_DLY
              then

                    if      T_TI_DLY_FL  > C_T_TI_DLY_FL(_AT)_2     (LV_AT = 0/1)
                    then  LV_FL  =  1
                    endif
              else

                    if      T_TI_DLY_FL  > C_T_TI_DLY_FL(_AT)_1     (LV_AT = 0/1)
                    then  LV_FL  =  1
                    endif
              endif

              increment timer T_TI_DLY_FL

        else  reset timer T_TI_DLY_FL
              LV_FL  =  0
        endif

 else         LV_FL  =  0

endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CL_MMV_THD_FL	1	0...FFFFH	0...2	3.05e-5	-
threshold for high canister load					
C_FAC_TQ_REQ_FL_HYS	1	0...FFFFH	0...1.999969	3.0518E-5	-
Hysteresis for driver wish threshold for Full load recognition					
C_LOAD_THD_FL_HYS	1	0...FFFFH	0...3.99993897	0.000061	-
Hysteresis for load threshold for Full load recognition					
C_N_MAX_FL_DLY	1	0...FFH	0...8160	32	rpm
Maximum engine speed for Full load recognition					
C_N_MIN_CP_FL	1	0...FFH	0...8160	32	rpm
Minimum engine speed for FL at high canister purge load					
C_N_MIN_FL	1	0...FFH	0...8160	32	rpm
Minimum engine speed for FL					
C_N_MIN_FL_DLY	1	0...FFH	0...8160	32	rpm
Minimum engine speed for Full load recognition					
C_T_MAX_FL_DLY	1	0...FFFFH	0...6553,5	0,1	s
Time after Start- Threshold value for Full load recognition					
C_T_MIN_FL_DLY	1	0...FFFFH	0...6553,5	0,1	s
Time after Start- Threshold for Full load recognition					
C_T_TI_DLY_FL_1	1	0 ... FFFFH	0 ... 655,35	0,01	s
Delay time Full load					
C_T_TI_DLY_FL_2	1	0 ... FFFFH	0 ... 655,35	0,01	s
Delay time Full load					
C_T_TI_DLY_FL_AT_1	1	0 ... FFFFH	0 ... 655,35	0,01	s
Delay time Full load t AT					
C_T_TI_DLY_FL_AT_2	1	0 ... FFFFH	0 ... 655,35	0,01	s
Delay time Full load AT					
C_TCO_MIN_FL	1	0...FEH	-48...142.5	0.75	°C
Minimum temperature for FL					
C_VS_MIN_FL	1	0...FFH	0...255	1	km/ h
Minimum velocity for FL					
IP_FAC_TQ_REQ_DRIV_FL	1x16	0...FFFFH	0...1.999969	3.0518E-5	-
LDPM_N_2_3	16	0...1FE0H	0...8160	1	rpm
Driver wish threshold for Full load recognition					
IP_LOAD_THD_FL	1x16	0...FFFFH	0...3.99993897	0.000061	-
LDPM_N_2_3	16	0...1FE0H	0...8160	1	rpm
Load threshold for Full load recognition					

64.2.2 Intercept function : (LV_INF)

Output data:

Name	**Mode**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
LV_INF	O/V	0...1H	0...1	1	-
Basic function " Intercept "					

Input data:

LV_CT	N_32	N_GRD	N_SP_IS
T_MIN_D_ISA	LV_PUC	LV_PU	LV_IS

FUNCTION DESCRIPTION:

General information:

In the engine operating states trailing throttle (LV_PU) or trailing throttle fuel cut off (LV_PUC), the **intercept function (LV_INF)** can be enabled below an engine speed threshold IP_N_MAX_INF.

For this purpose, the engine speed gradient N_GRD is monitored.

If the negative engine speed gradient exceeds the adjustable constant C_N_GRD_MIN during trailing throttle fuel cut off the injection will be reactivated to counteract the high negative engine speed gradient.

Application conditions:

Activation : at every engine operating state

Initialization : LV_INF = 0

Update rate : every segment

Formula section:

```
If      LV_CT      =  1              and
        N_32       <  IP_N_MAX_INF   and
        N_GRD      <  C_N_GRD_MIN
then    LV_INF     =  1
else    LV_INF     =  0
```

Calibration data:

Name	**Dim**	**Hex. limits**	**Phys. limits**	**Resol.**	**Unit**
C_N_GRD_MIN	1	80...00H	-4096...0	32	min-1/s
engine speed gradient threshold for activating the intercept function					
IP_N_MAX_INF	1x6	0...FFH	0...8160	32	min-1
LDP_N_GRD_N_MAX_INF	6	80...00H	-4096...0	32	min-1/s
Maximum engine speed for intercept function					

64.3 Torque Request for Safety Diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CTR_ABC_TQI_REQ_LIM	V / O	0...FFFF H	0 ... 65535	1	-
counter for torque request for safety is active					
LV_ERR_TQI_LIM	V/O	0...1H	0...1	1	-
Error flag torque request for safety is active					

Input data:

LV_TQ_LIM_INTV	LV_MTC_CUR_OFF	LV_ES	LV_IGK

FUNCTION DESCRIPTION:

General information:

The objective of this function is to determine the number of intervention actions concerning the torque request for safety and to set the failure table entry depending on LV_ERR_TQI_LIM if the counter exceeds a calibratable threshold.

The monitoring level generates a desired indicated torque TQI_SP_MON. If the actual torque TQI_REF_IGA_LAMB exceeds this value for more than a calibratable threshold a torque demand for the slow and the fast path is generated. The flag LV_TQ_LIM_INTV shows whether this torque intervention is active or not (see chapter D – "Torque request for safety").

LV_ERR_TQI_LIM.

Description:

The counter is initialized with zero for ECU reset, ignition key off, engine starts and engine stops. Otherwize it is incremented, if LV_TQ_LIM_INTV is active.

Application conditions:

Activation: *LV_IGK = 1* ***and*** *LV_ES = 0* ***and*** *LV_MTC_CUR_OFF = 0*

Deactivation: LV_MTC_CUR_OFF == 1

Initialization: 0

by: ECU reset

OR LV_IGK == 0

OR LV_ES == 1

Recurrence: 100 ms

Formula section:

```
IF      LV_TQ_LIM_INTV == 1
THEN    LV_ERR_TQI_LIM = 1 after debounce
ELSE    the debounce counter (CTR_ABC_TQI_LIM) is set to 0.
ENDIF
```

Description:

Error-symptoms are defined to this diagnosis function as following:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 |  |  |  └─> -                                    (= SYM_0)
 |  |  └────> -                                    (= SYM_1)
 |  └───────> -                                    (= SYM_2)
 └──────────> -    Engine Torque implausible       (= SYM_3)
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_MAX_TQI_LIM	1	0 ... FFFF H	0 ... 65535	1	-
maximum counter for torque request for safety					
C_ABC_INC_TQI_LIM	1	0 ... FF H	0 ... 255	1	-
increment counter for torque request for safety					

64.4 Torque Coordination General

General information:

The torque structure introduces the engine torque as a fundamental quantity . Either torque at clutch or indicated engine torque is used depending on the needs for the coordination structure.

Every external or internal torque requests are coordinated in the torque coordination modules where several criteria such as emissions, fuel consumption, engine protection and driveabillity are considered. Then torque request is distributed to the different regulated quantities load, ignition, mixture and fuel cut-off.

In this way the complexity can be reduced by the modular approach of the torque structure.

Every subfunction is linked by a torque interface to the torque structure (e.g. idle speed control, anti jerk function, engine speed limitation or catalyst heating, etc.). So the mutual interference of the subfunctions disapear because there is no direct link anymore to the different regulated quantities which decrease the tuning effort.

For the following overview of the torque structure only the most important modules and interface variables are mentionend to show the main track through the structure.

Pedal value interpretation

The Driver requests his driving wish by manipulation of the drivng pedal. As a result a certain pedal value PV_AV is observed which is transformed by different functions into a scaling factor FAC_TQ_REQ_DRIV (0...1).

Coordination of scaling factors for requested torque

This module coordinates different "driver" whises comming from the manual driver, cruise control or variable vehicle speed limitation.

The result is a coordinated scaling factor FAC_TQ_REQ.

Torque request at clutch general

The structure "Torque request at clutch general" consits of the modules "Maximum torque at clutch" and "Minimum torque at clutch". By the scaling factor FAC_TQ_REQ the requested torque at clutch TQ_REQ_CLU is calculated.

Torque Transient

The driver torque request TQ_REQ_CLU is filtered to protect the powertrain from jerking. The step response of the torque transient function are linear ramps. The slope of the ramps are a function of gear ratio, engine speed, torque range and torque increase and decrease state.

The outputs are TQ_REQ_CLU_TRA and TQI_REQ_TRA (TQ_REQ_CLU_TRA – TQ_LOSS) which is the indicated torque representation.

Parallel a torque reserve can be activated by "Torque reserve for torque transient" withe the output variable TQ_ADD_TRA. This torque reserve minimizes the difference between torque setpoint TQI_REQ_TRA and the actual engine torque TQI_AV which leads to a better response behavior of the engine at positive torque jump requests.

Torque reserve coordination

The torque reserve builds up an additionbal load which can be used when a fast torque increase is necessary (e.g. idle speed control) or if the ignition efficiency has to be reduced like for catalyst heating. The torque reserve quantities TQ_ADD_XXX are added to the requested torque value inside the slow torque path. Automatically parallel to the load increase a spark retard is done which will compensate the torque. To build up a torque reserve is a torque neutral process.

Minimum / Maximum torque request selection

This module collects all torque requests to generate the torque setpoint for the slow path TQI_REQ_SLOW and TQI_REQ_FAST for the fast path.

External torque requests:	
- driver torque request	TQI_REQ_TRA , TQI_ADD_ACT
- traction control (torque reduction/ increase)	TQI_ASR/MSR_FAST/SLOW
- automatic transmission (gear support)	TQI_GS_FAST/SLOW_INC/DEC

Internal torque requests:	
- engine speed limitation	TQI_N_MAX
- vehicle speed limitation	TQI_VS_MAX
- cruise control (optional)	
- engine power limitation	TQI_P_MAX
- catalyst protection	TQI_CAT_PROT
- torque plausibility check in level 1	TQI_REQ_LIM_SLOW/FAST

Minimum indicated engine torque at trailing throttle

This module will provide a minimum indicated engine torque TQI_MIN_PU to limit the throttle opening to a minimum value to ensure a safe combustion and to limit the intake manifold pressure.

For the engine operation states the status bit LV_TQI_BOL_SET is delivered to prepare the trailing throttle fuel cut-off. It indicates that the torque setpoint TQI_SP is below the minimum torque TQI_BOL at minimum ignition efficiency.

Torque coordination

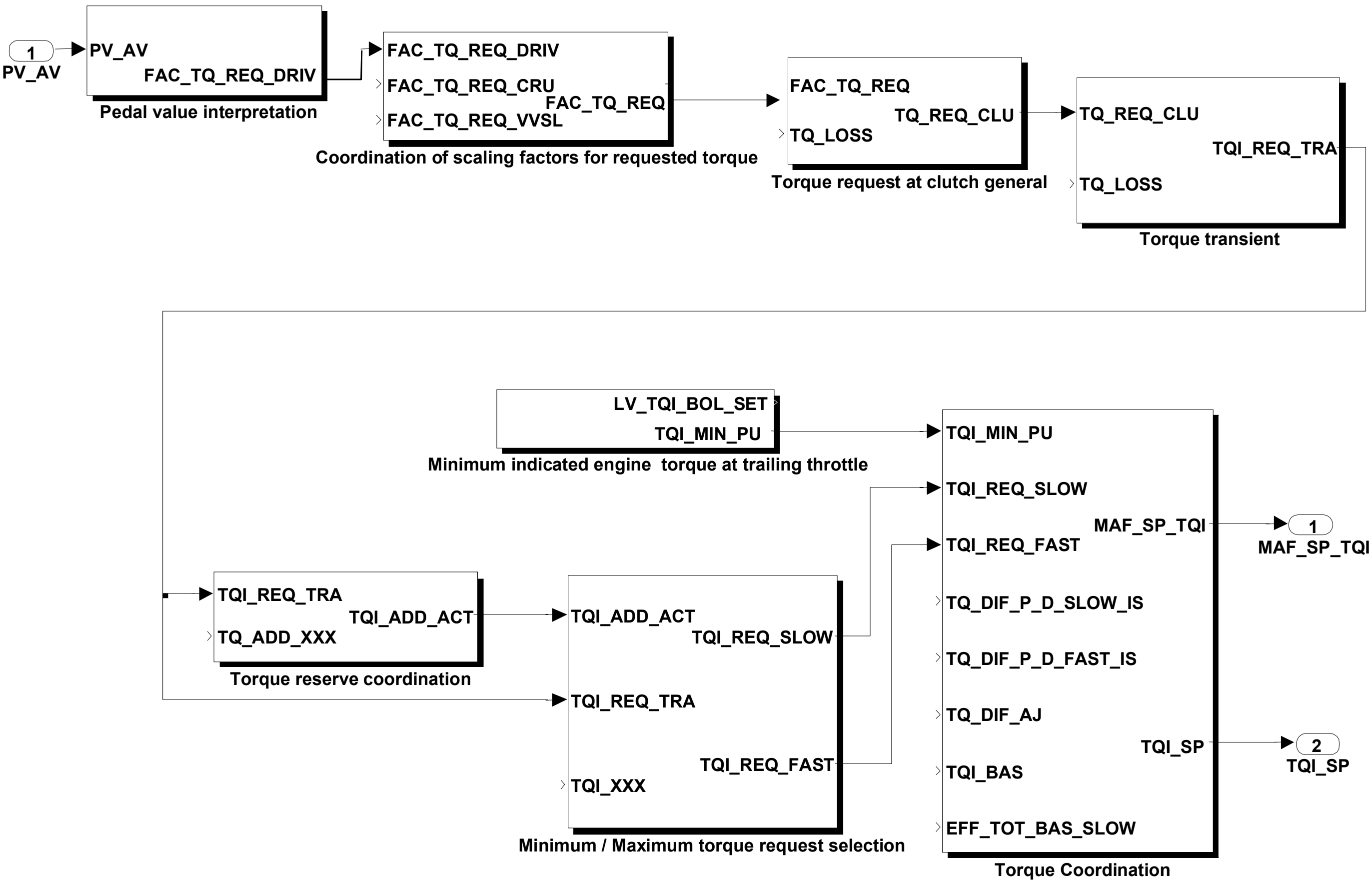
The coordinated torque values TQI_REQ_SLOW and TQI_REQ_FAST are entered into the slow and fast torque path where they are confronted with

- torque limitations TQI_MIN_PU, TQI_BAS
- idle speed control TQ_DIF_P_D_SLOW_IS, TQ_DIF_P_D_FAST_IS
- standardization to efficiency 1.00 by EFF_TOT_BAS_SLOW (for the air path torque TQI_REQ_SLOW)
- anti jerk correction TQ_DIF_AJ

The output from the slow path is MAF_SP_TQI which is an input into the inverse air path to determine the throttle opening.

TQI_SP the final target torque is the output from the fast torque path to be obtained in homogeneous mode. It serves as an input for the fast torque path like ignition angle, lambda adjustment (where it is necessary) and fuel cut-off pattern calculations.

Overview torque structure:



TQ_COORD_General_001.mdl

64.5 Torque Control at Engine Start

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_ST	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Engine Start Torque					
TQ_ST_1	V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Engine Start Torque raw value					
OPM_AV_SEL_TQ_ST	V	1...8H	1...8	1	-
Operation mode (selektiert)					

Input data:

LV_AT	LV_TQ_MIN_CLU	N_DIF	TCO
OPM_AV	LV_ST		

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_TQ_ST	1	0...FFH	0...0.99609375	0.00390625	-
Correlation constant for negative PT1 filtering					
ID_OPM_AV_TQ_ST	8	1...8H	1...8	1	-
LDPM_OPM_AV	8	0...8H	0...8	1	-
Operation mode matrix (actual)					
IP_TQ_ST_AT_OPM_1	8x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_IP_TQ_ST	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_TCO_IP_TQ_ST	8	0...FEH	-48...142.5	0.75	°C
Engine torque at start for automatic transmission in engine operation mode 1					
IP_TQ_ST_AT_OPM_2	8x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_IP_TQ_ST	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_TCO_IP_TQ_ST	8	0...FEH	-48...142.5	0.75	°C
Engine torque at start for automatic transmission in engine operation mode 2					
IP_TQ_ST_MT_OPM_1	8x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_IP_TQ_ST	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_TCO_IP_TQ_ST	8	0...FEH	-48...142.5	0.75	°C
Engine torque at start for manual transmission in engine operation mode 1					
IP_TQ_ST_MT_OPM_2	8x8	0...FFFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
LDPM_N_DIF_IP_TQ_ST	8	0...FFFFH	-3.2768E+4 ... 3.2767E+4	1	rpm
LDPM_TCO_IP_TQ_ST	8	0...FEH	-48...142.5	0.75	°C
Engine torque at start for manual transmission in engine operation mode 2					

64.5.1 TQSP_MD00S

General information:

The engine start torque TQ_ST influences the torque at clutch during start phase as a result it determines the engine speed transient response. It is calculated from a one dimensional map versus engine speed difference N_DIF.

TQ_ST is imported into module “Minimum Torque at Clutch” in order to increase engine speed during start phase. To bring the engine speed to its setpoint at idle the engine speed overshoot can be limitted supported by the tuning data IP_TQ_ST_MT(AT)_XY which depend on engine operation mode OPM_AV_SEL_TQ_ST.

Application conditions:

Activation:	LV_TQ_MIN_CLU = 0
Deactivation:	LV_TQ_MIN_CLU = 1
Initialization:	TQ_ST = 0 Nm at transition LV_ST = 0 to 1
Update rate:	10 ms

Application Condition

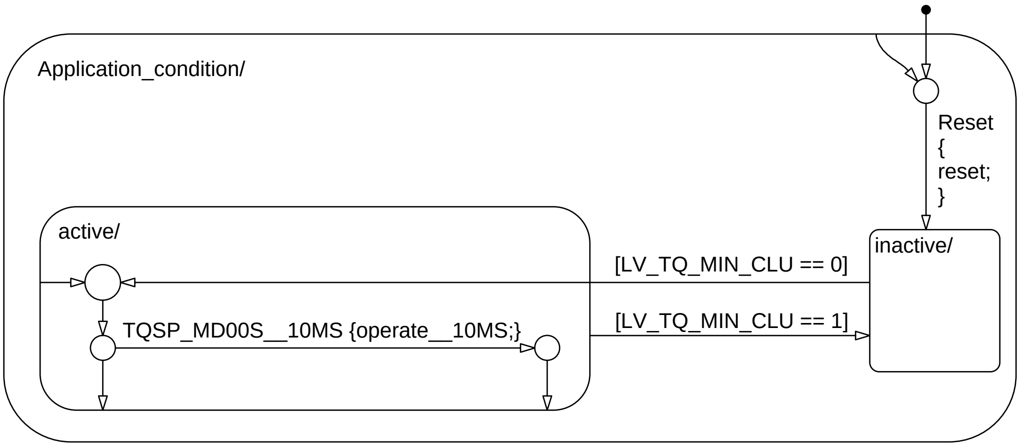
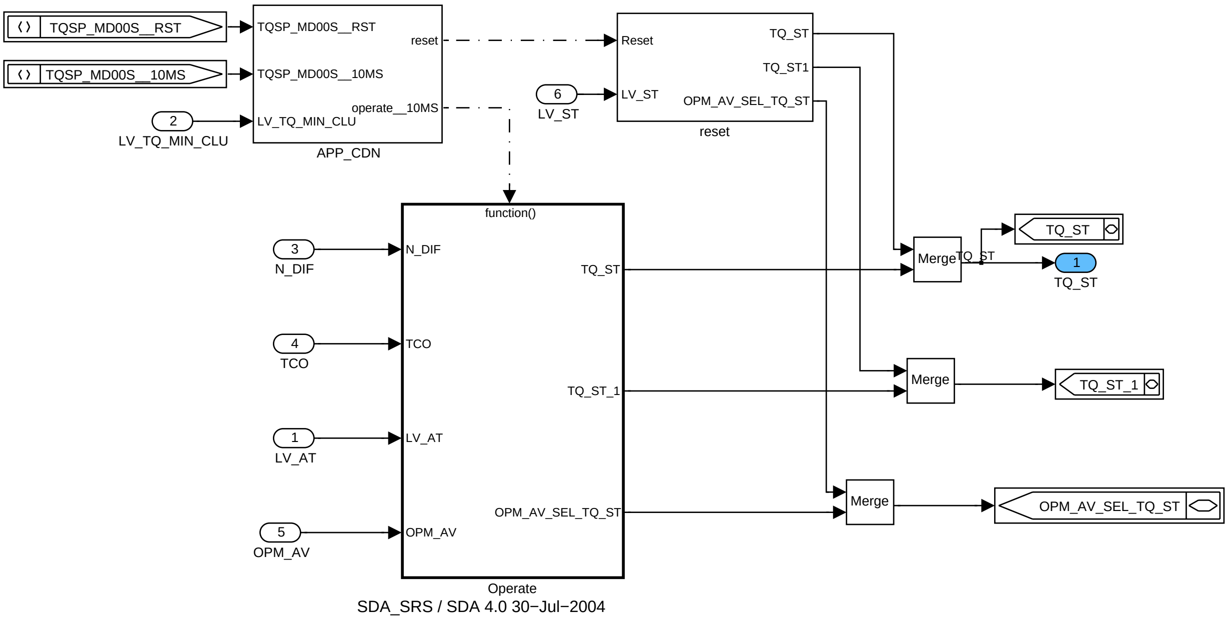


Figure 1 TQSP_MD00S/ APP_CDN/ APPCND

Function Description



SDA_SRS / SDA 4.0 30−Jul−2004

Figure 2 TQSP_MD00S

64.5.1.1 SUBFUNCTION: Operate

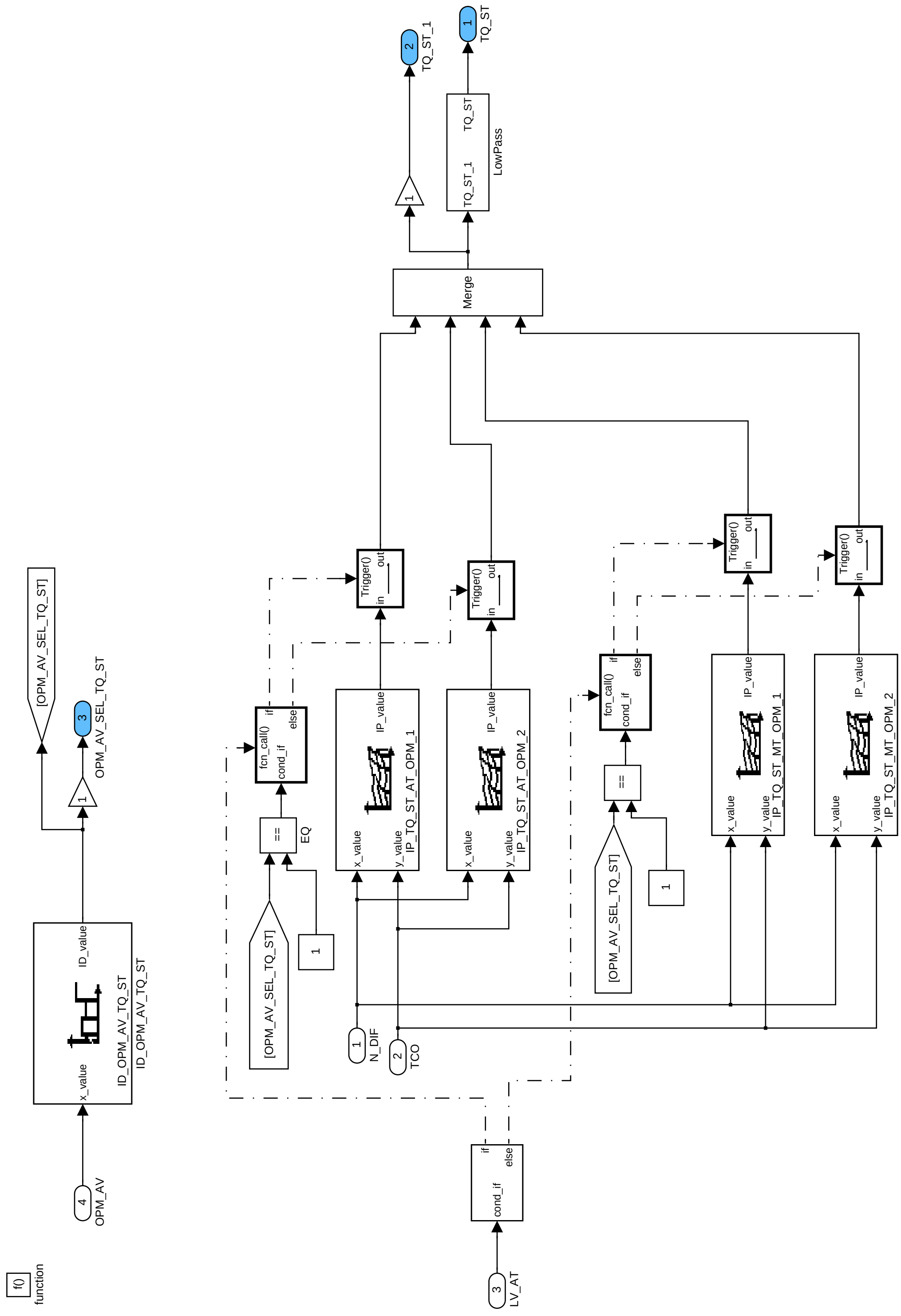


Figure 3 TQSP_MD00S/ Operate

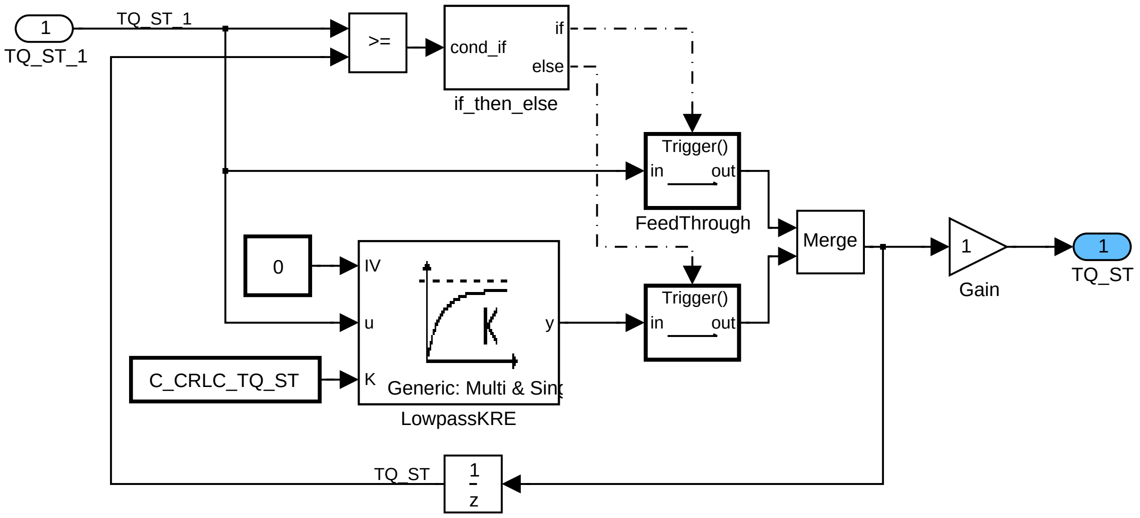


Figure 4 TQSP_MD00S/ Operate/ LowPass

64.5.1.2 SUBFUNCTION: reset

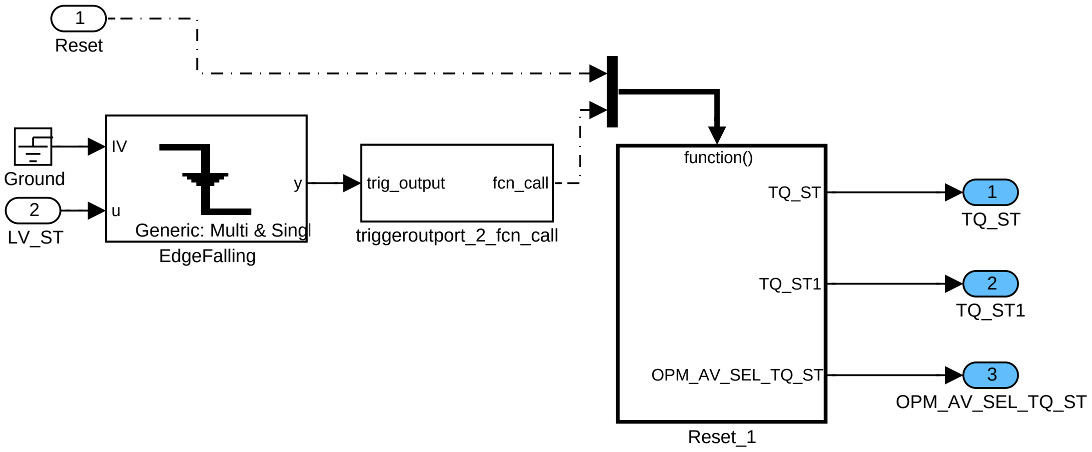


Figure 5 TQSP_MD00S/ reset

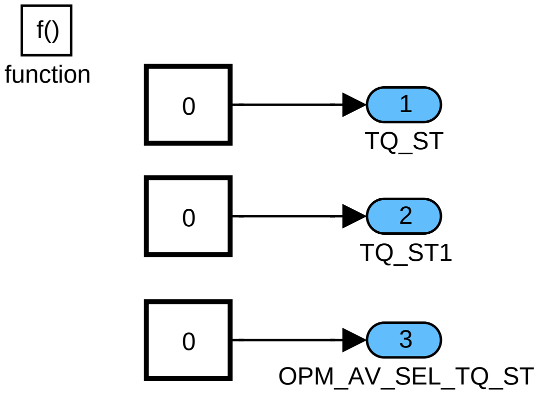


Figure 6 TQSP_MD00S/ reset/ Reset_1

64.6 Engine Torque Limitation

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_ACT_TQI_PBR_MAX	V	0... 1H	0... 1	1	[-]
Bit indicating conditions for torque limitation due to PBR fulfilled					
STATE_MOD_GB_WLC	O/V	0 1 2 3 7	SSC_SBC_OFF SSC_ON SBC_ON SSC_SBC_ON INVALID_SIGNAL	-	[-]
State mode gearbox for torque limitation due to WLC error (AT)					
TQ_LIM_AT	V	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Maximum allowed torque at clutch in case of TCU limp home (autarkic transmission protection for AT)					
TQ_LIM_TCT	V	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Maximum allowed torque at clutch in case of TCU limp home (autarkic transmission protection for TCT)					
TQ_LIM_WLC	V	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Maximum allowed torque at clutch in case of wandler launch control (WLC) error for AT					
TQI_P_MAX	O/V	0... 7FFFH	0... 1023.97	0.03125	[Nm]
maximum allowed indicated torque due to torque limitation					
TQI_PBR_MAX	O/V	0... 7FFFH	0... 1023.97	0.03125	[Nm]
maximum allowed indicated torque due to PBR					

Input Data:

TQ_LOSS	LV_AT	VS_FIL	LV_ETCU_DISABLE_CAN
STATE_ST_TQ_LIM_GS	GEAR	STATE_PBR	VS
LV_CFT_MOD_PBR	STATE_HLD_PBR	N_32	STATE_EGY_MIN_KWP
CTR_KM_CAN	C_CTR_KM_CAN_EGY_2_MAX	LV_VAR_TCT	STATE_PBR_ACT
STATE_TQ_WHEEL_PBR_QLY	STATE_PBR_ACT_QLY	STATE_MOD_GB	LV_TCS_DISABLE_CAN
ERR_SYM_BN_ETCU	ERR_SYM_BN_TQ_ETCU	ERR_SYM_BN_TQ_TCS	ERR_SYM_BN_VS_TCS
AC_WHEEL_PBR			

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_T_TQ_LIM_TOUT_AT	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for TCU limp - home (AT)					
C_T_TQ_LIM_TOUT_TCT	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for TCU limp - home (TCT)					
C_T_TQ_LIM_WLC	1	0... FFFFH	0... 655.35	0.01	[s]
Delay time for torque limitation due to WLC error (AT)					
C_TQ_LIM_AT_LGRD	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limitating gradient at end of AT torque limitation					
C_TQ_LIM_TCT_LGRD	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limitating gradient at end of TCT torque limitation					
C_TQ_LIM_TOUT_AT_LGRD	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limitation gradient for TCU limp - home (AT)					
C_TQ_LIM_TOUT_TCT_LGRD	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
Limitation gradient for TCU limp - home (TCT)					
C_TQ_LIM_WLC_LGRD	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]

Limitation gradient for WLC error (AT)					
C_TQI_PBR_MAX	1	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
maximum allowed TQI_ PBR_MAX					
C_VS_MAX_TQ_LIM_PBR	1	0... FFH	0... 255	1	[km/h]
Threshold of vehicle speed limitation for PBR					
ID_TQ_P_MAX_AT	13	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
LDPM_GEAR__TQ_P_MAX	13	0... CH	0... 12	1	[-]
maximum allowed torque at clutch due to torque limitation depending on gear ratio by AT					
ID_TQ_P_MAX_MT	13	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
LDPM_GEAR__TQ_P_MAX	13	0... CH	0... 12	1	[-]
maximum allowed torque at clutch due to torque limitation depending on gear ratio by MT					
IP_TQ_LIM_AT	8*8	0... 7FFFH	0... 1023.97	0.03125	[Nm]
LDPM_VS_FIL_IP_TQ_LIM_AT	8	0... FFFFH	0... 511.9921875	7.8125e-3	[km/h]
LDPM_N_32_IP_TQ_LIM_AT	8	0... FFH	0... 8160	32	[rpm]
Maximum allowed torque at clutch in case of TCU limp - home (autarkic transmission protection for AT)					
IP_TQ_LIM_TCT	8*8	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
LDPM_VS_FIL_IP_TQ_LIM_TCT	8	0... FFFFH	0... 511.9921875	7.8125e-3	[km/h]
LDPM_N_32_IP_TQ_LIM_TCT	8	0... FFH	0... 8160	32	[rpm]
Maximum allowed torque at clutch in case of TCU limp - home (autarkic transmission protection for TCT)					
IP_TQ_LIM_TOUT_AT	8*8	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
LDPM_VS_FIL_IP_TQ_LIM_AT	8	0... FFFFH	0... 511.9921875	7.8125e-3	[km/h]
LDPM_N_32_IP_TQ_LIM_AT	8	0... FFH	0... 8160	32	[rpm]
Maximum allowed torque at clutch before TCU limp - home (AT)					
IP_TQ_LIM_TOUT_TCT	8*8	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
LDPM_VS_FIL_IP_TQ_LIM_TCT	8	0... FFFFH	0... 511.9921875	7.8125e-3	[km/h]
LDPM_N_32_IP_TQ_LIM_TCT	8	0... FFH	0... 8160	32	[rpm]
Maximum allowed torque at clutch before TCU limp - home (TCT)					
IP_TQ_LIM_WLC	8	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_TQ_LIM_WLC	8	0... FFH	0... 8160	32	[rpm]
Maximum allowed torque at clutch in case of wandler launch control (WLC) error (AT)					
IP_TQ_P_MAX_EGY_2	8	0... 7FFFH	0... 1023.96875	0.03125	[Nm]
LDP_N_32_IP_TQ_P_MAX_EGY_2	8	0... FFH	0... 8160	32	[rpm]
Maximum allowed torque due to energy spare mode 2					
LC_TQI_P_MAX_INH_AS	1	0... 1H	0... 1	1	[-]
switch for torque limitation ON/OFF (0 = ON)					

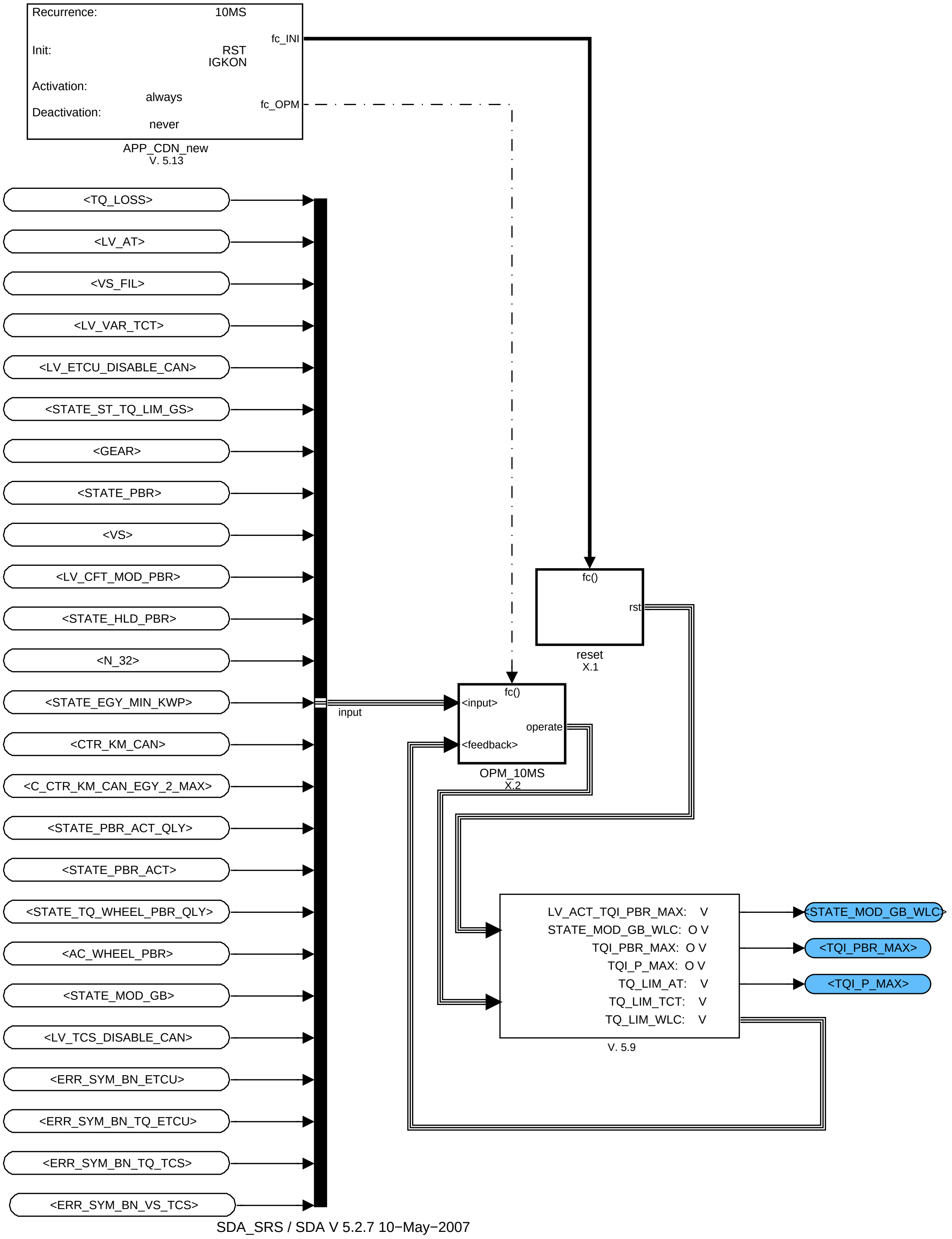
General Information

Due to customer- or hardware- (gear box) request the maximum torque at clutch depending on gear ratio and transmission type can be limited.

Application Conditions

Initialization: RST, IGKON
Recurrence: 10MS
Activation: always
Deactivation: never

Function description



SDA_SRS / SDA V 5.2.7 10−May−2007

Figure 7:

64.6.1 Initialisation at reset or LV_IGK = 0 to 1

TQI_P_MAX = 1023,97 Nm
TQ_PBR_MAX = 1023,97 Nm
TQ_LIM_AT = 1023,97 Nm
TQ_LIM_TCT = 1023,97 Nm
LV_ACT_TQI_PBR_MAX = 0

64.6.2 Recurrence:10 ms

64.6.2.1 Calculation of TQ_LIM_WLC

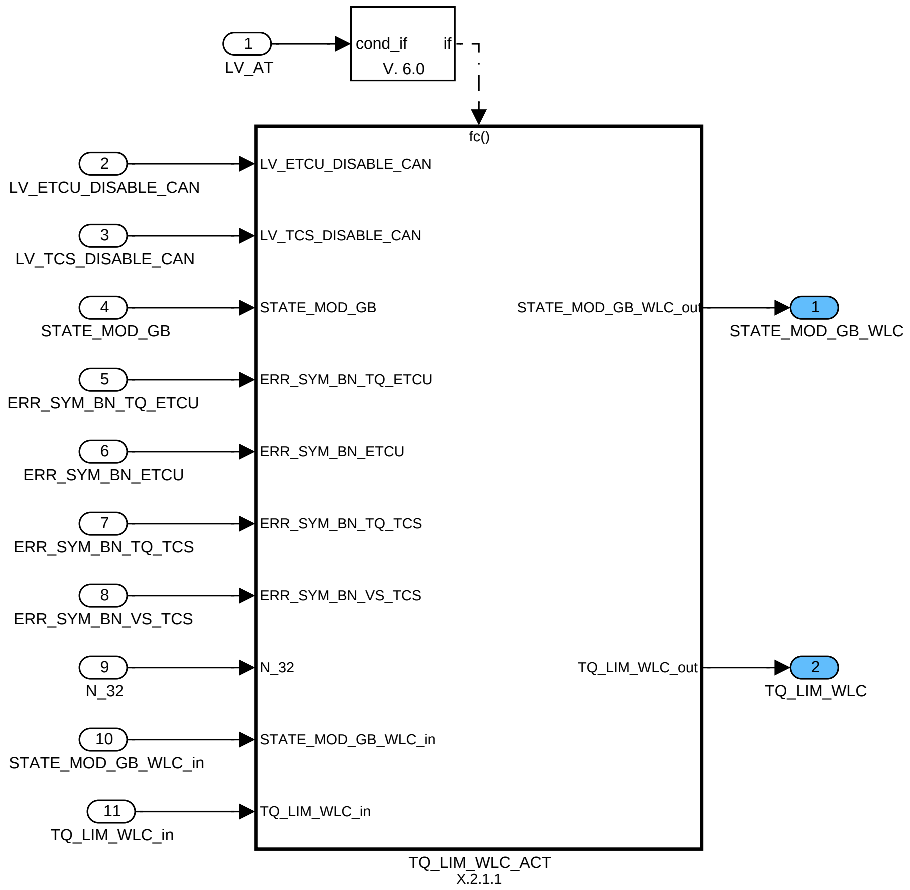


Figure 8:

64.6.2.1.1 Calculation of the torque limitation at clutch in case of wandler launch control error

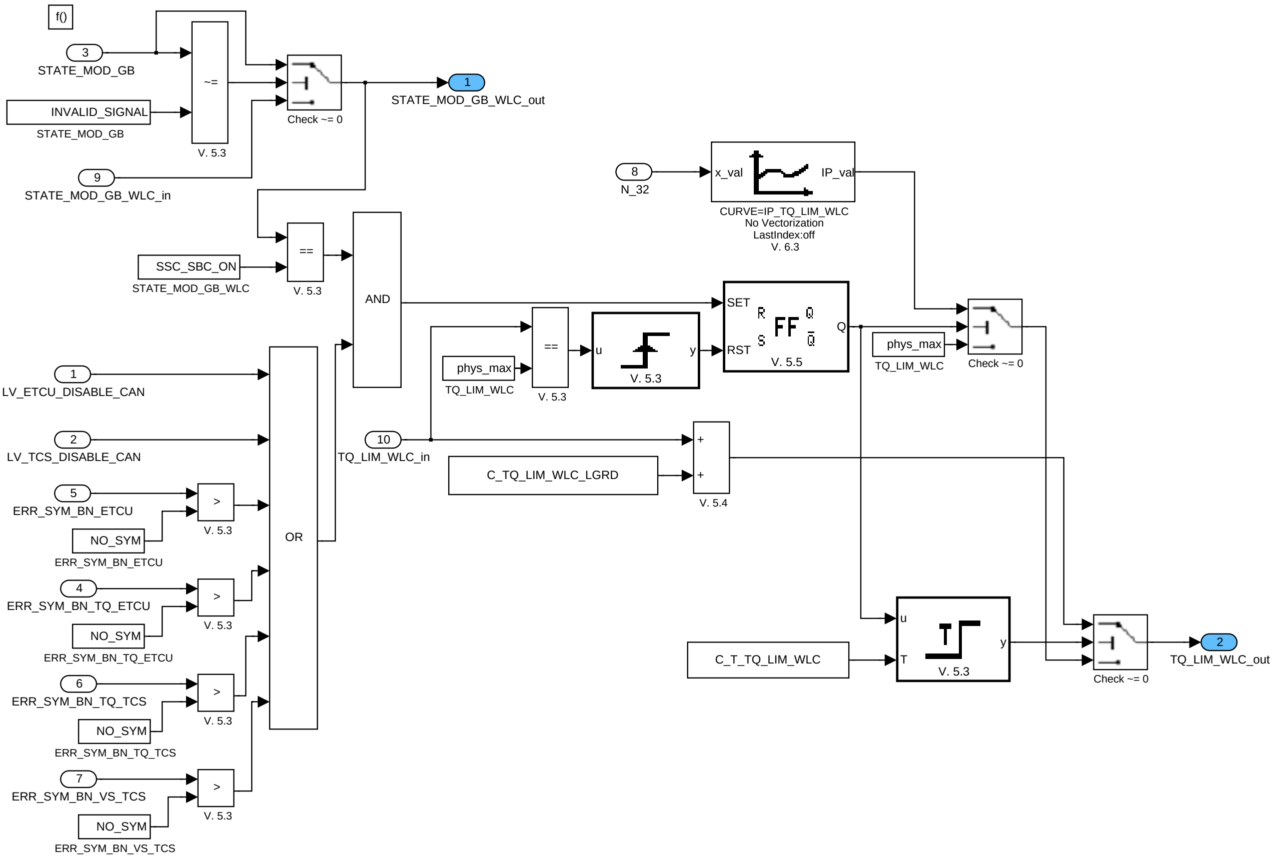


Figure 9:

64.6.2.2 Calculation of torque limitation for automatic shifted transmission

Maximum allowed torque at clutch in case of TCU limp home. Autarkic transmission protection for AT.

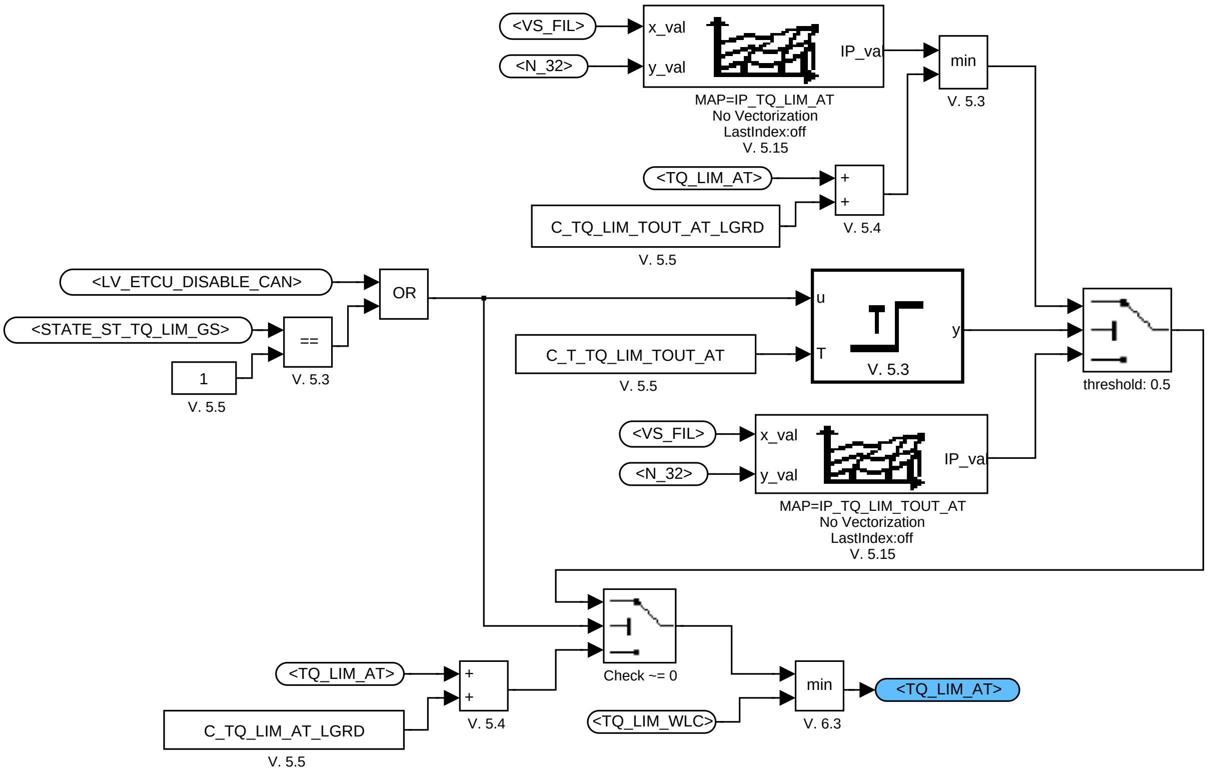


Figure 10:

64.6.2.3 Calculation of torque limitation for twin clutch transmission

Maximum allowed torque at clutch in case of TCU limp home. Autarkic transmission protection for TCT.

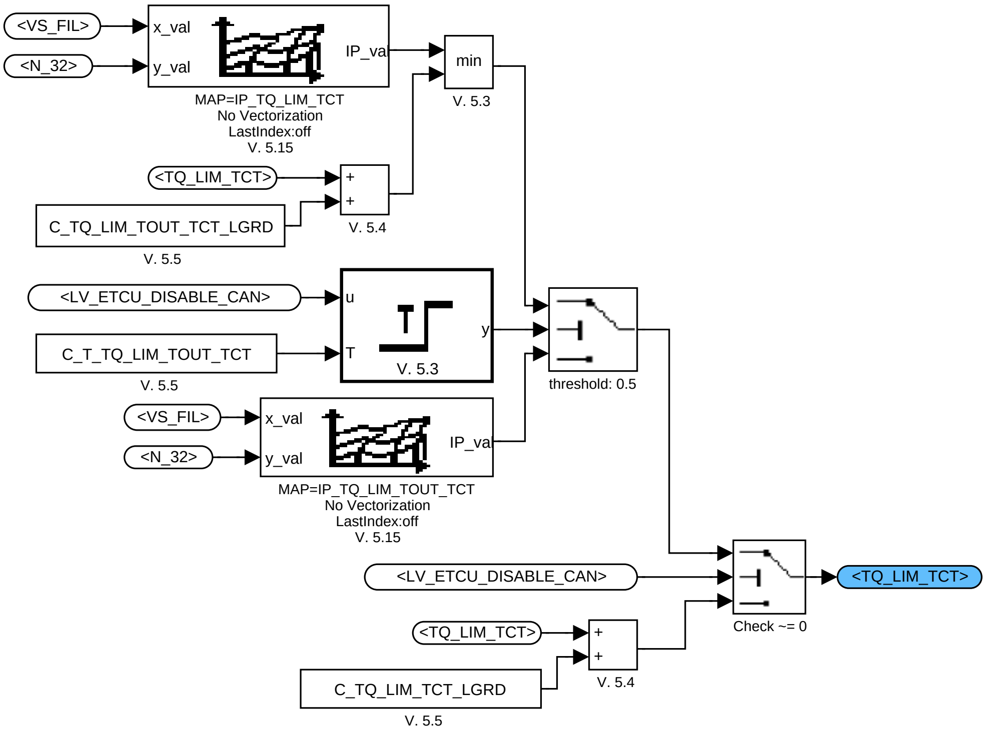


Figure 11:

64.6.2.4 Calculation of TQI_P_MAX_CLC

Calculation of the maximum indicated engine torque proportional part.

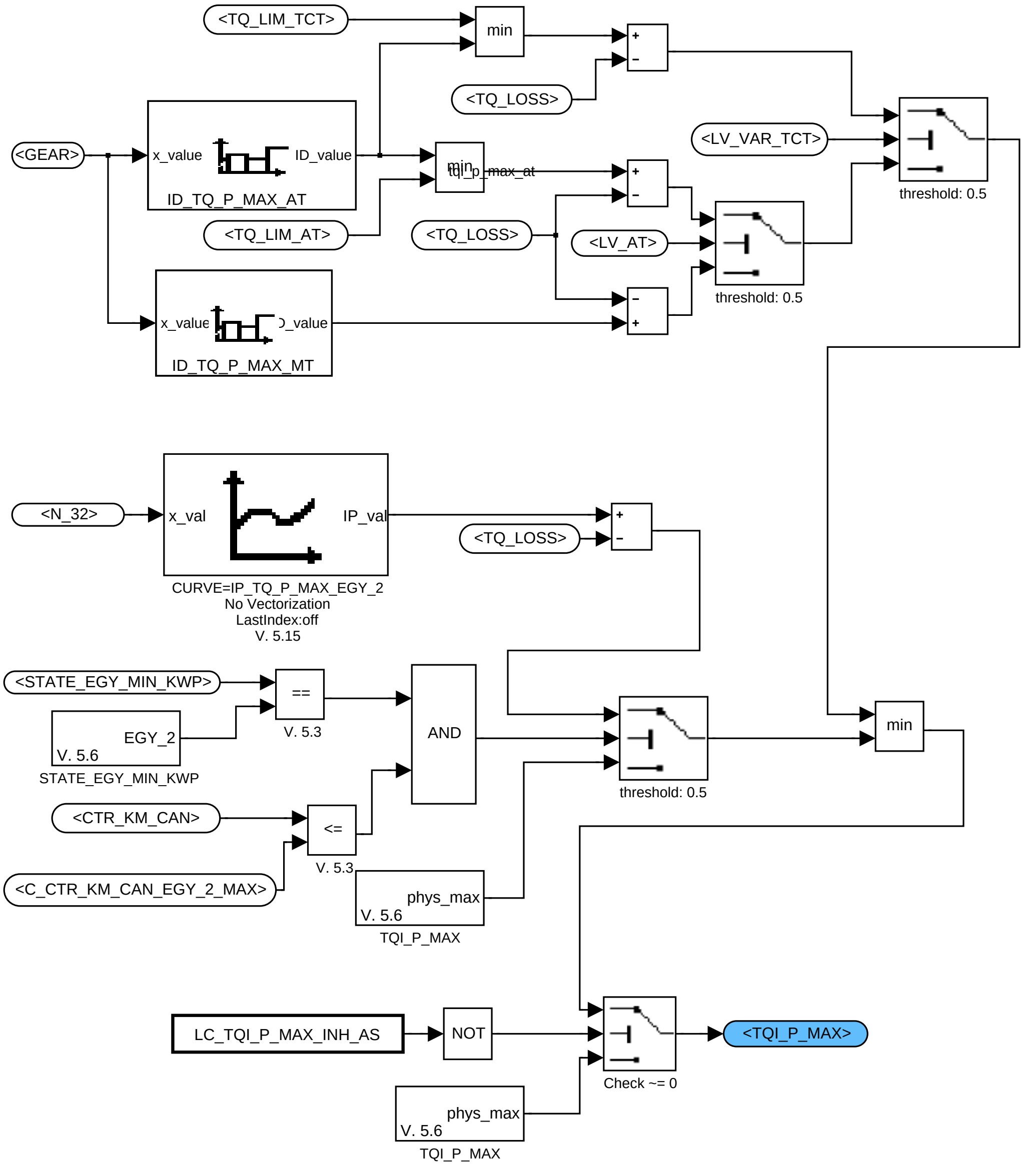


Figure 12:

64.6.2.5 Calculation of TQI_PBR_MAX

Calculation of the maximum indicated engine torque for PBR.

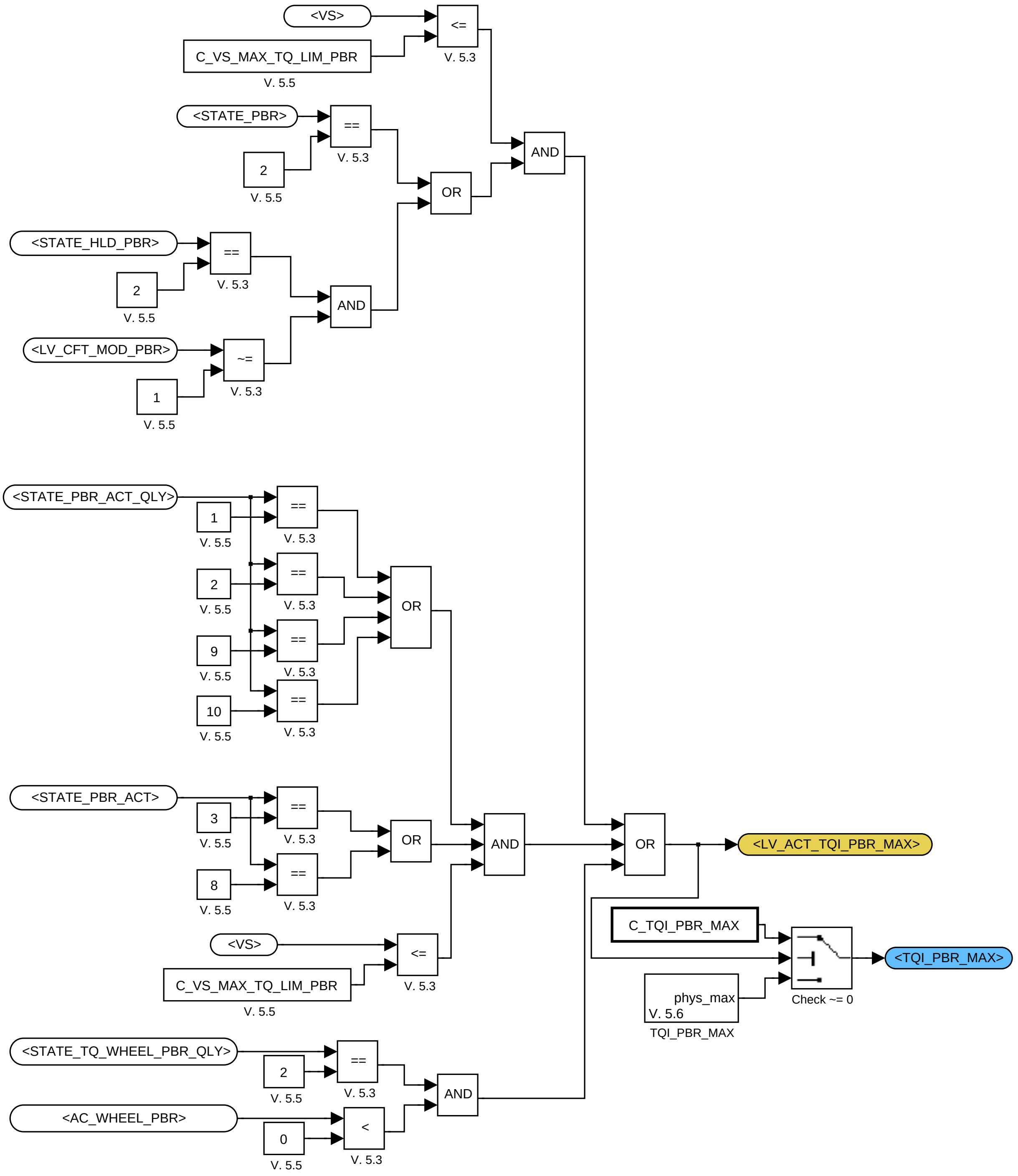

<VS>
C_VS_MAX_TQ_LIM_PBR
V. 5.5
<=
V. 5.3
<STATE_PBR>
2
V. 5.5
==
V. 5.3
AND
OR
<STATE_HLD_PBR>
2
V. 5.5
==
V. 5.3
AND
<LV_CFT_MOD_PBR>
1
V. 5.5
~=
V. 5.3
<STATE_PBR_ACT_QLY>
1
V. 5.5
==
V. 5.3
2
V. 5.5
==
V. 5.3
OR
9
V. 5.5
==
V. 5.3
10
V. 5.5
==
V. 5.3
<STATE_PBR_ACT>
3
V. 5.5
==
V. 5.3
OR
8
V. 5.5
==
V. 5.3
AND
OR
<LV_ACT_TQI_PBR_MAX>
C_TQI_PBR_MAX
<TQI_PBR_MAX>
Check ~= 0
phys_max
V. 5.6
TQI_PBR_MAX
<VS>
<=
V. 5.3
C_VS_MAX_TQ_LIM_PBR
V. 5.5
<STATE_TQ_WHEEL_PBR_QLY>
2
V. 5.5
==
V. 5.3
AND
<AC_WHEEL_PBR>
0
V. 5.5
<
V. 5.3

Figure 13:

64.7 Torque Request at Clutch General

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TQ_REQ	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Torque factor from pedal value interpretation and cruise control					
FAC_TQ_REQ_CLU	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Torque scaling factor for calculation requested torque at clutch					
LV_TQ_CRU_ACT	O/V	0...1H	0...1	1	-
Logical variable cruise control active					
FAC_TQ_REQ_CLU_LDM	O/V	0...FFFFH	0...1.99996948	3.05176E-5	-
Torque scaling factor for calculation requested torque at clutch for LDM					

Input data:

FAC_TQ_REQ_CRU	FAC_TQ_REQ_DRIV	LV_MAF_BLS_DIAG	LV_LDM_ENA
FAC_TQ_REQ_DCC			

64.7.1 TQSP_MD00L

General information:

The module "Torque Request at Clutch General" supplies the BMW torque structure with the torque scaling factor FAC_TQ_REQ_CLU from the driver or a cruise control.

FAC_TQ_REQ is exported and can be used from a transmission control unit for calculation of gear shifting points.

If a MAF error is detected and brake is active then the torque request is set to zero.

Application conditions:

Initialisation:	*all = 0 at reset*
Recurrence:	*10 ms*
Activation:	*at every engine state*
Deactivation:	-

Application Condition

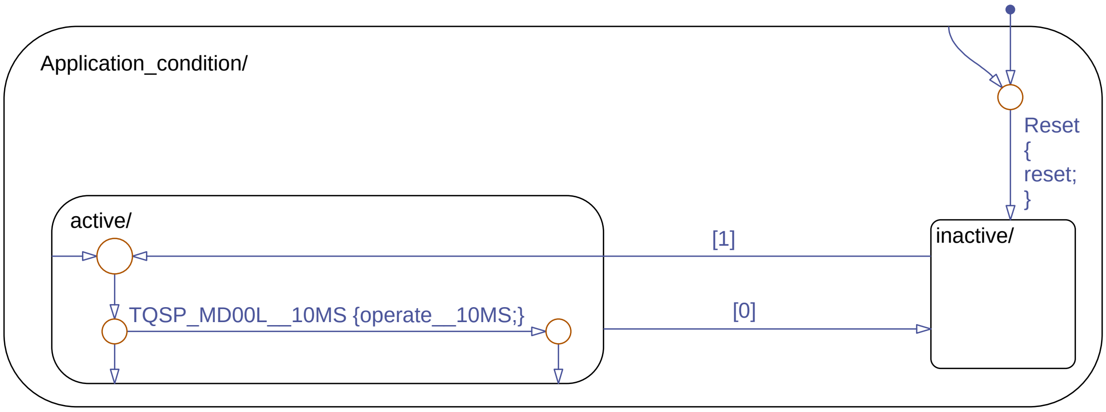


Figure 14 TQSP_MD00L/ APP_CDN/ APPCND

Function Description

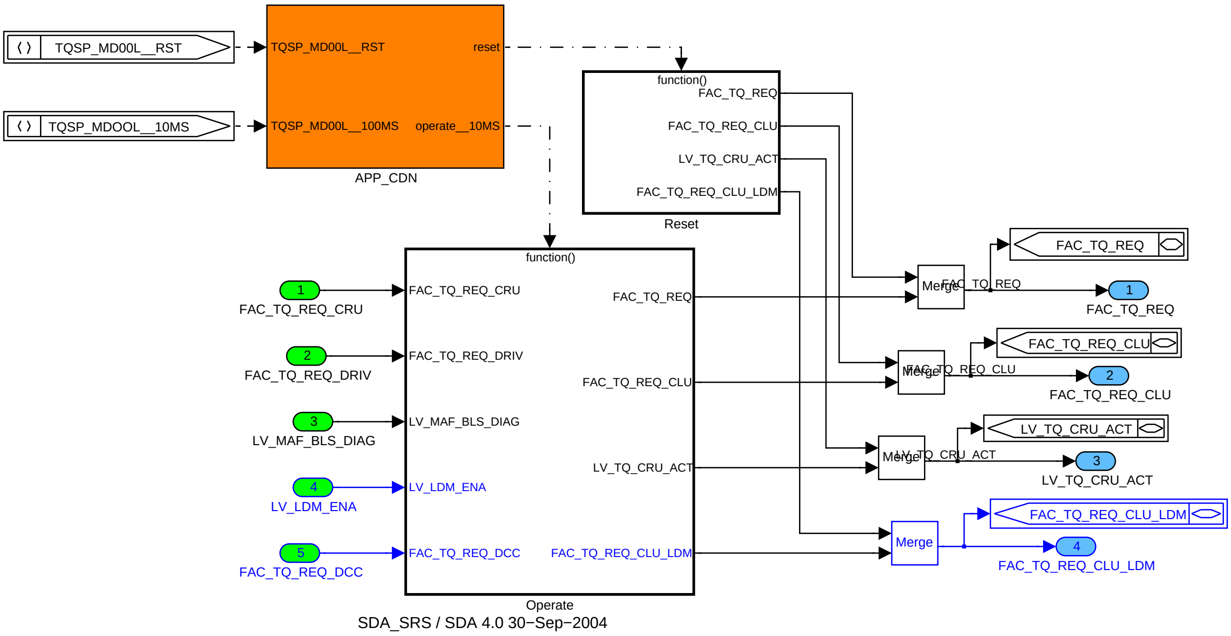


Figure 15 TQSP_MD00L

64.7.1.1 SUBFUNCTION: Operate

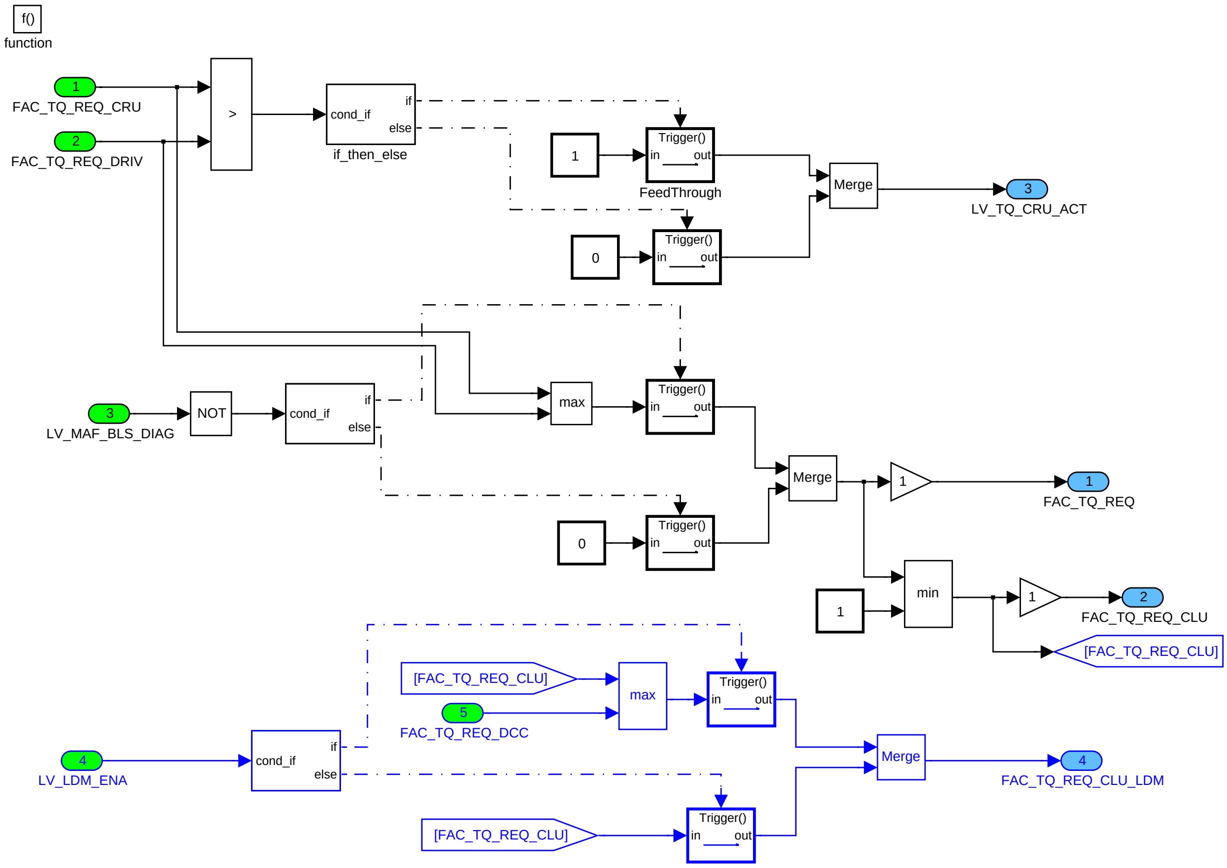


Figure 16 TQSP_MD00L/ Operate

64.7.1.2 SUBFUNCTION: Reset

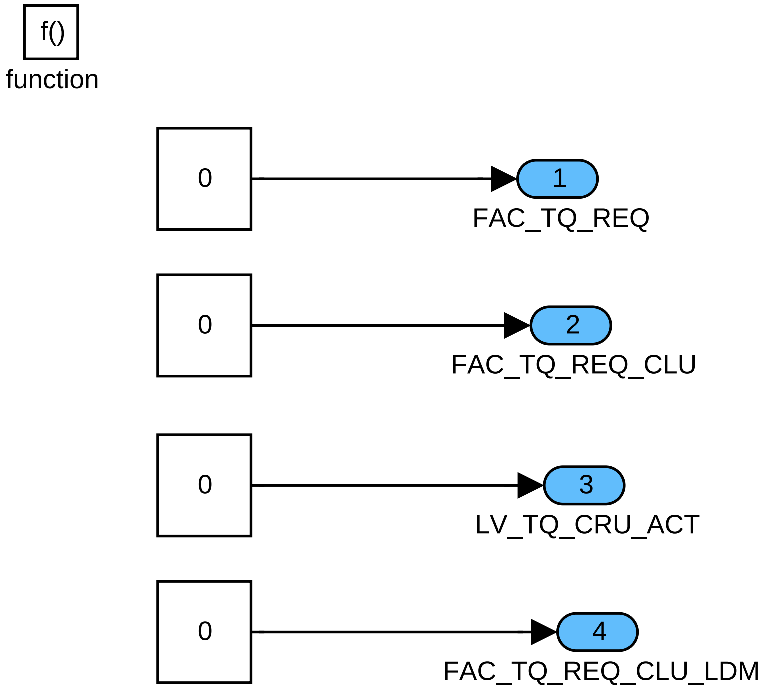


Figure 17 TQSP_MD00L/ Reset

64.8 General flag for inhibition of external torque requests

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TQI_REQ_CAN_INH	V/O	0...1H	0...1	1	[-]
general flag for inhibition of external torque requests					

Input data:

SF_TQD	LV_ERR_MTC_CTL_1		

FUNCTION DESCRIPTION:

General information:

LV_TQI_REQ_CAN_INH collects all states or flags within engine management which shall lead to ignoring external torque requests.

Application conditions:

Initialisation: LV_TQI_REQ_CAN_INH = 0 for LV_IGK 0 --> 1 or at reset

Recurrence: 10ms

Activation: at every engine state

Deactivation: otherwise

Formula section:

```
If      SF_TQD = 3                or          // --> see remark *
        LV_ERR_MTC_CTL_1 = 1

then    LV_TQI_REQ_CAN_INH = 1

else    LV_TQI_REQ_CAN_INH = 0

Endif
```

Remark:

For detailed calculation of SF_TQD see chapter "Torque request for traction control"

SF_TQD = 3 H is set if

STATE_ETC_LIH = ETC_LIH_1 ***or***

STATE_ETC_LIH = ETC_LIH_2_REV ***or***

STATE_ETC_LIH = ETC_LIH_2 ***or***

LV_OFF_MTC_MON = 1 ***or***

LV_SF_TQD_MON = 1

(see Chapter: "Applc. Inc of process monitoring" and "Fault reaction of Process monitoring")

64.9 Torque Request for Safety

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQI_REQ_LIM_FAST	V/O	8000...7FFF H	-1024...1023.97	0.03125	Nm
Requested torque for safety for fast path					
TQI_REQ_LIM_SLOW	V/O	8000...7FFF H	-1024...1023.97	0.03125	Nm
Requested torque for safety for slow path					
LV_TQ_LIM_INTV	V/O	0...1	0...1	1	-
Bit for torque intervention for safety active					
SUM_TQI_REQ_LIM	S/V/O	0...FFFF H	0...65535	1	-
Event counter of active torque limitations (stored in non volatile memory)					
TQI_DIF_LIM	V	8000...7FFF H	-1024...1023.97	0.03125	Nm
Torque offset for recognition torque intervention active					
TQI_REQ_LIM_1	V	8000...7FFF H	-1024...1023.97	0.03125	Nm
Intermediate value 1 for requested torque for safety					
TQI_REQ_LIM_2	V	8000...7FFF H	-1024...1023.97	0.03125	Nm
Intermediate value 2 for requested torque for safety					
TQI_REQ_LIM_3	V	8000...7FFF H	-1024...1023.97	0.03125	Nm
Intermediate value 3 for requested torque for safety					
TQI_REQ_LIM_4	V	8000...7FFF H	-1024...1023.97	0.03125	Nm
Intermediate value 4 for requested torque for safety					
LV_TQ_LIM_INTV_1	V	0...1	0...1	1	-
Result for TQI_AV > TQI_SP_MON					
LV_TQ_LIM_INTV_2	V	0...1	0...1	1	-
ON-Delayed (40ms) result for TQI_AV > TQI_SP_MON					
T_TQ_LIM_INTV	V	0...FFFFH	0...2621.4	0.04	s
Decremented time counter after transition LV_TQ_LIM_INTV_2 1 --> 0 to set LV_TQ_LIM_INTV = 0					
T_TQI_REQ_LIM	V	0...FFFFH	0...2621.4	0.04	s
Incremented time counter after transition LV_TQ_LIM_INTV_2 0 -->1 to set LV_TQ_LIM_INTV = 0					
T_TQI_REQ_LIM_MAX	V/O/S	0...FFFFH	0...2621.4	0.04	s
max time counter after transition LV_TQ_LIM_INTV_2 0 -->1 to set LV_TQ_LIM_INTV = 0					
TQI_REQ_LIM_DIF	V	8000...7FFF H	-1024...1023.97	0.03125	Nm
difference of torque request for safety					
TQI_REQ_LIM_DIF_MAX	V/O/S	8000...7FFF H	-1024...1023.97	0.03125	Nm
Max difference of torque request for safety					
TQI_REQ_LIM_DIF_OFS	-	0 ... 1FE0 H	0 ... 255	0.03125	Nm
Permitted deviation offset					

Input data:

TQI_SP_MON	TQI_AV	N_32	LV_MTC_CUR_OFF
LV_IGK	LV_TQ_LIM_EXT_ADJ	LV_ES	LC_AD_CLR_TQ

FUNCTION DESCRIPTION:

General information:

The ETC safety concept monitors the torque generation of the engine by comparing the actual indicated engine torque TQI_AV to the desired indicated engine torque TQI_SP_MON. In order to avoid a false diagnosis decision caused by tolerances and disturbances this torque intervention function is introduced.

The module '*Torque Request for Safety*' delivers two torque request values TQI_REQ_LIM_FAST for the fast torque path (ignition and fuel cut-off) and TQI_REQ_LIM_SLOW for the slow torque path (air path) which leads to a torque intervention if the actual reference torque TQI_AV exceeds the setpoint TQI_SP_MON by more than a specified threshold.

This state is indicated by the bit LV_TQ_LIM_INTV_1. It is set if TQI_AV is bigger than the setpoint TQI_SP_MON plus the offset TQI_DIF_LIM.

The bit LV_TQ_LIM_INTV is set to 1 if LV_TQ_LIM_INTV_2 = 1 and set to 0 C_T_TQ_LIM_INTV seconds after an transition of LV_TQ_LIM_INTV_2 to 0 to prevent LV_TQ_LIM_INTV from to fast jittering if the torque limitation goes on off on ... and is used for diagnosis. LV_TQ_LIM_INTV_2 is used to activate ignition or fuel cut of intervention.

TQI_AV is calculated from the reference torque value TQI_REF which includes the actual engine speed N_32 and MAF and actual efficencies of lambda ignition and cylinder shut off.

TQI_DIF_LIM is the safety offset which is the sum of TQI_SP_MON and IP_TQI_DIF_LIM which depends on N_32 and TQI_SP_MON itself.

If the actual torque TQI_AV is inside the permitted range (LV_TQ_LIM_INTV_1 = 0) then TQI_REQ_LIM_2 is set to the calibratible value C_TQI_REQ_LIM_PAS (e.g. max. engine torque + 10%).

The torque increase of TQI_REQ_LIM_3 is realized by a change limitation (C_TQ_LIM_INC) to avoid positive torque jumps if the limitation condition disappears.

TQI_REQ_LIM_SLOW is also limited by a rate limiter to avoid negative jumps of the throttle.

By the tuneable logical constants LC_TQ_LIM_INH_FAST and LC_TQ_LIM_INH_SLOW either the fast torque output or the slow torque output or both can be inhibited (set to passive values) for further investigations.

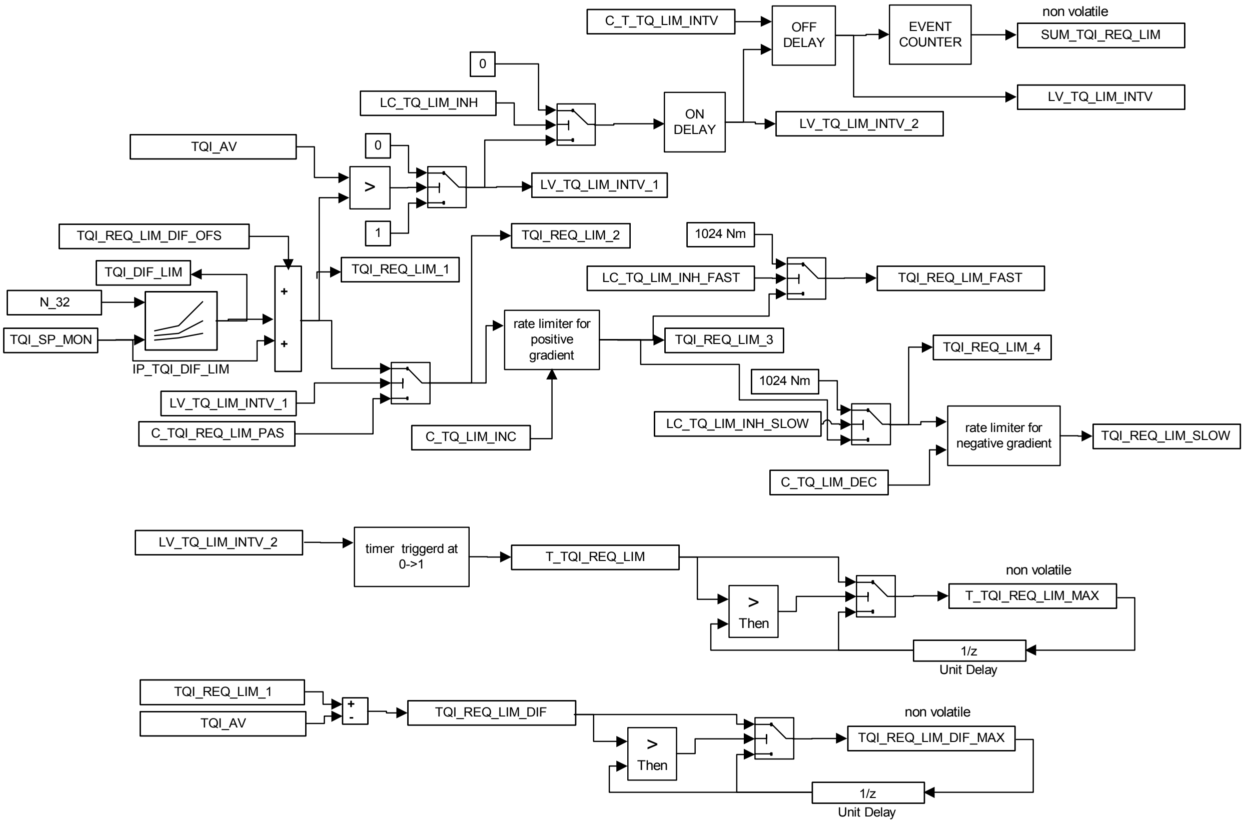
ON-Delay:

The ON-Delay is necessary because of this module is not synchronized to the module ‘Desired indicated engine torque’ of level 2. This could cause inaccurate torque limitation in case of a high positive torque gradient in level 1 while TQI_SP_MON is not updated yet.

Drag Pointer functions:

Within the tuning phase is reliable a statistical evaluation of interventions. Therefore some drag pointers information are available.

Signal flow diagram:



Application conditions:

Activation: LV_IGK = 1 **and** LV_ES = 0 **and** LV_MTC_CUR_OFF = 0

Deactivation: otherwise

Initialization: at reset **or** LV_IGK = 0 **or**

LV_ES = 1 **or** LV_MTC_CUR_OFF = 1

then LV_TQ_LIM_INTV = 0

LV_TQ_LIM_INTV_1 = 0

LV_TQ_LIM_INTV_2 = 0

TQI_REQ_LIM_FAST = C_TQI_REQ_LIM_PAS

TQI_REQ_LIM_SLOW = C_TQI_REQ_LIM_PAS

TQI_REQ_LIM_3 = C_TQI_REQ_LIM_PAS

T_TQ_LIM_INTV = 0 s

T_TQI_REQ_LIM = 0 s

TQI_REQ_LIM_DIF = 0 Nm

TQI_REQ_LIM_DIF_OFS = C_TQI_DIF_LIM_MAX_OFS

Update-Rate: *40 ms*

Formula section:

TQI_REQ_LIM_DIF_OFS$_n$ =

```
MAX (TQI_REQ_LIM_DIF_OFS$_{(n-1)}$ - C_DEC_TQI_LIM_DIF_MAX; 0Nm)
```

Request-Calculation (I):

```
TQI_REQ_LIM_1 = TQI_SP_MON + IP_TQI_DIF_LIM (N_32 ; TQI_SP_MON) +
                TQI_REQ_LIM_DIF_OFS
TQI_REQ_LIM_DIF = TQI_REQ_LIM_1 - TQI_AV

IF      LC_TQ_LIM_INH == 1
THEN    LV_TQ_LIM_INTV_1$_k$ = 0
ELSE    IF      TQI_AV > TQI_REQ_LIM_1
        THEN    LV_TQ_LIM_INTV_1$_k$ = 1
        ELSE    LV_TQ_LIM_INTV_1$_k$ = 0
        ENDIF
ENDIF
```

ON-Delay:

```
IF      [(LV_TQ_LIM_INTV_1$_{k-2}$ == 1) AND (LV_TQ_LIM_INTV_1$_{k-1}$ == 1) AND
        (LV_TQ_LIM_INTV_1$_k$ == 1)]
        {delay by 2 recurrencies of the module}
THEN    LV_TQ_LIM_INTV_2$_k$ = 1
ELSE    LV_TQ_LIM_INTV_2$_k$ = 0
ENDIF
```

Off-Delay:

```
IF      LV_TQ_LIM_INTV_2$_k$ == 1
THEN    (LV_TQ_LIM_INTV = 1
        T_TQ_LIM_INTV = C_T_TQ_LIM_INTV)
ELSE    (T_TQ_LIM_INTV = T_TQ_LIM_INTV – [update rate]            {limited to 0}
        IF      T_TQ_LIM_INTV == 0
        THEN    LV_TQ_LIM_INTV = 0
        ENDIF)
ENDIF
```

Request-Calculation (II):

```
IF      LV_TQ_LIM_INTV_2$_k$ == 1
THEN    TQI_REQ_LIM_2 = TQI_REQ_LIM_1
ELSE    TQI_REQ_LIM_2 = C_TQI_REQ_LIM_PAS
ENDIF
```

Rate limiter for positive gradient:

```
IF      (TQI_REQ_LIM_2 > (TQI_REQ_LIM_3$_{k-1}$ + C_TQ_LIM_INC))
THEN    TQI_REQ_LIM_3$_k$ = TQI_REQ_LIM_3$_{k-1}$ + C_TQ_LIM_INC
ELSE    TQI_REQ_LIM_3$_k$ = TQI_REQ_LIM_2
ENDIF
```

TQI_REQ_LIM_FAST calculation:

```
IF      LC_TQ_LIM_INH_FAST == 1             {fast torque request path disabled}
THEN    TQI_REQ_LIM_FAST = 1023.97          {upper limit}
ELSE    TQI_REQ_LIM_FAST = TQI_REQ_LIM_3$_k$
ENDIF
```

TQI_REQ_LIM_4 calculation:

```
IF      LC_TQ_LIM_INH_SLOW == 1             {slow torque request path disabled}
THEN    TQI_REQ_LIM_4 = 1023.97             {upper limit}
ELSE    TQI_REQ_LIM_4 = TQI_REQ_LIM_3$_k$
ENDIF
```

Rate limiter for negative gradient:

```
IF      (TQI_REQ_LIM_4 < (TQI_REQ_LIM_ SLOW$_{k-1}$ – C_TQ_LIM_DEC))
THEN    TQI_REQ_LIM_SLOW$_k$ = TQI_REQ_LIM_SLOW$_{k-1}$ - C_TQ_LIM_DEC
ELSE    TQI_REQ_LIM_SLOW$_k$ = TQI_REQ_LIM_4
ENDIF
```

Event counter active torque limitation: SUM_TQI_REQ_LIM

```
At transition LV_TQ_LIM_INTV =  0 -> 1
THEN    increment SUM_TQI_REQ_LIM by 1
ELSE    SUM_TQI_REQ_LIM unchanged
ENDIF
```

The counter SUM_TQI_REQ_LIM is stored in non volatile memory till error memory is erased or is reseted at external adjustment or LC_AD_CLR_TQ

Drag pointer of torque intervention:

```
IF LV_TQ_LIM_INTV_2 = 1 and TQI_REQ_LIM_DIF < TQI_REQ_LIM_DIF_MAX
Then    TQI_REQ_LIM_DIF_MAX = TQI_REQ_LIM_DIF
else    TQI_REQ_LIM_DIF_MAX unchanged
endif
```

```
IF(1) LV_TQ_LIM_INTV_2$_{n}$ = 1 and LV_TQ_LIM_INTV_2$_{n-1}$ = 0
Then(1)   T_TQI_REQ_LIM = 0s        ("reset at activation")
Else(1) IF(2)         LV_TQ_LIM_INTV_2$_{n}$ = 1
    Then (2) T_TQI_REQ_LIM = T_TQI_REQ_LIM + Update rate
            IF(3) T_TQI_REQ_LIM > T_TQI_REQ_LIM_MAX
            Then(3) T_TQI_REQ_LIM_MAX = T_TQI_REQ_LIM
            Else(3)       T_TQI_REQ_LIM_MAX unchanged
    Else(2)       T_TQI_REQ_LIM unchanged
Endif
```

The drag pointers TQI_REQ_LIM_DIF_MAX and T_TQI_REQ_LIM_MAX are stored in non volatile memory till error memory is erased or a reset is done at external adjustment or LC_AD_CLR_TQ .

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_TQ_LIM_INH	1	0 ... 1 H	0 ... 1	1	-
Logical constant inhibition torque intervention for safety in general					
LC_TQ_LIM_INH_FAST	1	0 ... 1 H	0 ... 1	1	-
Logical constant inhibition torque intervention for safety for fast path					
LC_TQ_LIM_INH_SLOW	1	0 ... 1 H	0 ... 1	1	-
Logical constant inhibition torque intervention for safety for slow path					
C_TQI_REQ_LIM_PAS	1	0 ... 7FFF H	0 ... 1023.97	0.03125	Nm
Passive torque request for no torque intervention					
C_TQ_LIM_INC	1	0 ... 7FFF H	0 ... 1023.97	0.03125	Nm
Maximum positve gradient of the requested torque TQI_REQ_LIM					
IP_TQI_DIF_LIM	4*4	0 ... FFFF H	-1024 ... 1023.97	0.03125	Nm
LDP_N_32__TQI_DIF_TQI_LIM	4	0 ... FF H	0 ... 8160	32	rpm
LDP_TQI_SP_MON__TQI_DIF_TQI_LIM	4	0 ... FF H	0 ... 510	2	Nm
Torque threshold depending on TQI_SP_MON					
C_TQ_LIM_DEC	1	0 ... 7FFF H	0 ... 1023.97	0.03125	Nm
Maximum negative gradient of the requested torque TQI_REQ_LIM_SLOW					
C_T_TQ_LIM_INTV	1	0...FFFFH	0...2621.4	0.04	s
Time after active torque intervention to reset LV_TQ_LIM_INTV					
C_DEC_TQI_LIM_DIF_MAX	1	20 ... 1FE0H	1 ... 255	0,03125	Nm
decrement after start condition					
C_TQI_DIF_LIM_MAX_OFS	1	0 ... 1FE0H	0 ... 255	0.03125	Nm
Offset after start condition					

64.10 Cus adap module: TQSP

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_fgr_akt	O/V	0...1H	0...1	1	[-]
cruise control active					
B_ll_roh	O/V	0...1H	0...1	1	[-]
Condition idle speed (raw value)					
B_vl	O/V	0...1H	0...1	1	[-]
condition full load					
EFF_IGA_MIN_ADD_MAX	O/V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Minimum possible ignition efficiency for catalyst heating					
LOAD_THD_FL	O/V	0...FFFFH	0...3.99993	0.061e-3	[-]
load threshold for Full load recognition					
MAF_ADD_MAX	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
Maximum possible engine load for catalyst heating - mg/Stk					
MAF_KGH_ADD_MAX	O/V	0...FFFFH	0...2047.96875	0.03125	[kg/h]
Maximum possible engine load for catalyst heating - kg/h					
MAF_SP_TQI	O/V	0...FFFFH	0...1389	0.0211948	[mg/stk]
mass air flow setpoint for torque intervention					
Mdi_fsb	O/V	D8F0...2710H	-1000...1000	0.1	[Nm]
Momentenbegrenzung bei aktivierter Feststellbremse					
Mdi_pmax	O/V	D8F0...2710H	-1000...1000	0.1	[Nm]
indicated engine torque for power limitation					
Mdi_res_st	O/V	D8F0...2710H	-1000...1000	0.1	[Nm]
torque reserve at engine start					
Mdi_siko_l	O/V	D8F0...2710H	-1000...1000	0.1	[Nm]
torque limitation monitoring, slow path					
Mdi_siko_s	O/V	D8F0...2710H	-1000...1000	0.1	[Nm]
torque limitation monitoring, fast path					
Mdr_wunsch	O/V	0...FFFFH	0...255.99609	3.9063e-3	[%]
scaling factor requested torque at clutch					
TQ_ADD_MAX	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
maximum torque reserve at clutch					
TQ_REQ_CLU	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Driver requested torque at clutch after IS-PL transition					
TQ_REQ_IPM	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Torque request for IPM					
TQI_ADD_MAX	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Maximum possible indicated engine torque inclusive torque reserve at IGA_MIN and ISC reserve limitation					
TQI_ADD_MAX_TOL	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Maximum possible indicated engine torque inclusive torque reserve at IGA_MIN					
TQI_EMS	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Requested indicated engine torque without external interventions					
TQI_REQ_FAST	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
limited torque request for fast path					
TQI_REQ_FAST_SEL	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
fast indicated engine torque request selected					
TQI_REQ_SLOW	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
limited torque request for slow path					
TQI_REQ_SLOW_SEL	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]

slow indicated engine torque request selected					
TQI_REQ_TRA_FAST	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
requested indicated torque for fast path					
TQI_SP	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque setpoint fast path					
TQI_SP_CAN	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
setpoint indicated engine torque for CAN calculations					
TQI_SP_CBK[NC_CBK_EX_NR]	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque setpoint for ignition path (bank selective)					
TQI_SP_S	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque setpoint for stratified combustion mode without interventions of ISC and AJC					

Input data:

C_MAF_REF	ECU_STATE	Eta_zw_minres	F_vl_aus
FAC_TQ_REQ_CLU	LV_CT	LV_FL_RAW	LV_TQ_CRU_ACT
Mdi_fzdyn	Mdi_res_max	Mdi_res_maxll	Mdi_soll
Mdi_soll_k	Mdi_soll_l	Mdi_soll_lk	Mdi_wunsch
Mdi_zw_soll	Mdi_zw_soll_i[NC_CBK_EX_NR]	Mdi_zw_sollcan	Mdk_wunsch
Mdk_wunsch_ipm	Ms_res_max	N_32	Rf_res_max
Rf_soll	TQ_ADD_ST	TQI_P_MAX	TQI_PBR_MAX
TQI_REQ_LIM_FAST	TQI_REQ_LIM_SLOW		

Application conditions:

Initialisation at reset:: Mdi_fsb = 1000 Nm
Mdi_pmax = 1000 Nm
All other = 0h

Recurrence : 10 ms

Activation: every engine state

64.10.1 Outputs for BMW functions

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
if      ECU_STATE = "PWL"
then

        B_ll_roh =  0
        Mdi_pmax =   1000 Nm
        Mdi_fsb    =    1000 Nm

else
```

```
    B_ll_roh =  LV_CT
    Mdi_pmax =  TQI_P_MAX
    Mdi_fsb     =  TQI_PBR_MAX
endif
```

Mdr_wunsch = FAC_TQ_REQ_CLU *100%

B_fgr_akt = LV_TQ_CRU_ACT

B_vl = LV_FL_RAW

Mdi_res_st = TQ_ADD_ST

Mdi_siko_l = TQI_REQ_LIM_SLOW

Mdi_siko_s = TQI_REQ_LIM_FAST

64.10.2 Outputs for SV aggregates

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs can be different from the specified values due to the input data attributes.

Formula section:

Remark: all formulas are valid in a **physical** meaning

EFF_IGA_MIN_ADD_MAX = Eta_zw_minres / 100 %

LOAD_THD_FL = F_vl_aus

MAF_ADD_MAX = Rf_res_max * C_MAF_REF

MAF_SP_TQI = Rf_soll * C_MAF_REF

TQ_ADD_MAX = Md_res_max

TQ_REQ_CLU = Mdk_wunsch

TQI_ADD_MAX = Mdi_res_maxll

TQI_ADD_MAX_TOL = Mdi_res_max

TQI_EMS = Mdi_wunsch

TQI_REQ_FAST = Mdi_soll

TQI_REQ_FAST_SEL = Mdi_soll_k

TQI_REQ_SLOW_SEL = Mdi_soll_lk

TQI_REQ_TRA_FAST = Mdi_fzdyn

TQI_SP = Mdi_zw_soll

TQI_SP_CBK[NC_CBK_EX_NR] = Mdi_zw_soll_i[NC_CBK_EX_NR]

TQI_SP_CAN = Mdi_zw_sollcan

```
TQI_REQ_SLOW        = Mdi_soll_l
MAF_KGH_ADD_MAX     = Ms_res_max
If   ECU_STATE = "PWL"
Then
     TQI_SP_S = 0 Nm
     TQ_REQ_IPM = 0 Nm
Else
     TQI_SP_S      =  Mdi_zw_soll
     TQ_REQ_IPM =  Mdk_wunsch_ipm
Endif
```

65 Transmission

Table of Contents

Index

65.1 Gear shift signal

65.1.1 Plausibility of gear shift setpoint displayed on CAN

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
GS_IDC_GEAR	O/V	0...FFH	0...255	1	[-]
Gear shift signal CAN - setpoint for gear					
GS_IDC_DISP	O/V	0...FFH	0...255	1	[-]
Gear shift signal CAN - display type					
LV_GS_DOWN_IDC_OFF	V	0...1H	0...1	1	[-]
Gear shift down inhibited - CAN display off					
LV_GS_DOWN_IDC_OFF_1	V	0...1H	0...1	1	[-]
Gear shift down 1 inhibited - CAN display off					
LV_GS_DOWN_IDC_OFF_2	V	0...1H	0...1	1	[-]
Gear shift down 2 inhibited - CAN display off					
LV_GS_DOWN_IDC_OFF_3	V	0...1H	0...1	1	[-]
Gear shift down 3 inhibited - CAN display off					
LV_GS_IDC_OFF	V	0...1H	0...1	1	[-]
SPA not coded					
CTR_INH_GS_IDC	V	0...FFH	0...255	1	[-]
Counter for inhibiting gear-shift signal display					
LV_INH_GS_IDC_1	V	0...1H	0...1	1	[-]
Gear shift signal display 1 on CAN inhibited					
LV_INH_GS_IDC_2	V	0...1H	0...1	1	[-]
Gear shift signal display 2 on CAN inhibited					

Input data:

GS_IDC_GEAR_RAW	GS_IDC_DISP_RAW	VS	LV_ERR_VS_CAN
GEAR	N	LV_ERR_CAN_BOFF	LV_ERR_VS_PLAUS
LV_ERR_BN_VS_TCS	LV_ERR_CRK	LV_MTC_CUR_OFF	LV_AT
LV_VAR_AMT	LV_VAR_TCT	LV_IGK	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CTR_INH_GS_IDC_MAX	1	1...FFH	1...255	1	[-]
Max. counter for inhibiting gear-shift signal display					
C_CTR_INH_GS_IDC_INC	1	0...FFH	0...255	1	[-]
Increment for inhibiting gear-shift signal display					
C_CTR_INH_GS_IDC_DEC	1	0...FFH	0...255	1	[-]
Decrement for inhibiting gear-shift signal display					
C_GS_IDC_GEAR_SUB	1	0...FFH	0...255	1	[-]
Default-value for gear display					
C_GS_IDC_DISP_SUB	1	0...FFH	0...255	1	[-]
Default-value for display type					
C_GS_IDC_GEAR_MAX	1	0...FFH	0...255	1	[-]
Max-value for gear display					
C_GS_IDC_DISP_MAX	1	0...FFH	0...255	1	[-]
Max-value for display type					
LC_GS_IDC_TCT_ENA	1	0...1H	0...1	1	[-]
Enabling plausibility also for TCT					
C_GS_DISP	1	0...FFH	0...255	1	[-]
Constant for calibrating gear shift condition to be checked (up/down, etc.)					
ID_GS_IDC_GEAR_MIN_1	8	0...FFH	0...255	1	[-]
LDP_VS_ID_GS_IDC_GEAR_MIN_1	8	0...FFH	0...255	1	[km/h]
Min. gear 1 depending on VS					
ID_GS_IDC_GEAR_MIN_2	8	0...FFH	0...255	1	[-]
LDP_VS_ID_GS_IDC_GEAR_MIN_2	8	0...FFH	0...255	1	[km/h]
Min. gear 2 depending on VS					

FUNCTION DESCRIPTION:

General information:

The calculation of gear shift signal setpoint (output BMW-layer) is based on VS and N. Therefore the the gear shift signal setpoint should not be transmitted on CAN in case of VS-error, unplausible N or limp home mode.

Application conditions:

Initialisation: all 0 at reset

Recurrence: 100 ms

Activation: LV_IGK = 1 **and**
LV_AT= 0 **and**
LV_VAR_AMT = 0 **and**
(LV_VAR_TCT = 0 **or**
LC_GS_IDC_TCT_ENA = 1)

Formula section:

Inhibition of gear shift down:

```
if      GS_IDC_GEAR_RAW < = GEAR               and
        GS_IDC_DISP_RAW  = C_GS_DISP           and
        GS_IDC_GEAR_RAW  != C_GS_IDC_GEAR_SUB
Then    LV_GS_DOWN_IDC_OFF_1 = 1
Else    LV_GS_DOWN_IDC_OFF_1 = 0
Endif
```

```
If        GS_IDC_GEAR_RAW <  ID_GS_IDC_GEAR_MIN_2 and
          GS_IDC_GEAR_RAW  != C_GS_IDC_GEAR_SUB
Then      LV_GS_DOWN_IDC_OFF_2 = 1
Else      LV_GS_DOWN_IDC_OFF_2 = 0
Endif

If        GS_IDC_GEAR_RAW <  ID_GS_IDC_GEAR_MIN_1 and
          GS_IDC_DISP_RAW ! = C_GS_DISP                and
          GS_IDC_GEAR_RAW  != C_GS_IDC_GEAR_SUB
Then      LV_GS_DOWN_IDC_OFF_3 = 1
Else      LV_GS_DOWN_IDC_OFF_3 = 0
Endif

If        LV_GS_DOWN_IDC_OFF_1 = 1    or
          LV_GS_DOWN_IDC_OFF_2 = 1    or
          LV_GS_DOWN_IDC_OFF_3 = 1
Then      LV_GS_DOWN_IDC_OFF = 1
Else      LV_GS_DOWN_IDC_OFF = 0
Endif

If        [B_spa_cist = 0        and                          // no SPA coded,
          (GS_IDC_GEAR_RAW != C_GS_IDC_GEAR_SUB or     // Spa_gang > 0
          GS_IDC_DISP_RAW ! = C_GS_IDC_DISP_SUB)]   or     // Spa_art > 0
          [GS_IDC_GEAR_RAW > C_GS_IDC_GEAR_MAX or      // Spa_gang > max.
          GS_IDC_DISP_RAW  > C_GS_IDC_DISP_MAX]           // Spa_art > max.
Then      LV_GS_IDC_OFF = 1
Else      LV_GS_IDC_OFF = 0
Endif
```

General suppressing of gear shift setpoint:

```
If      (LV_GS_DOWN_IDC_OFF = 1                 OR
        LV_ERR_VS_CAN     = 1                   OR     // VS-CAN-signal not valid,
                                                       // no debounced error, only
                                                       flag received via CAN
        LV_GS_IDC_OFF = 1)
Then    If      CTR_INH_GS_IDC < C_CTR_INH_GS_IDC_MAX
        Then    CTR_INH_GS_IDC = CTR_INH_GS_IDC + C_CTR_INH_GS_IDC_INC
        Else    LV_INH_GS_IDC_1 = 1
        Endif
Else    If      CTR_INH_GS_IDC > 0
        Then    CTR_INH_GS_IDC = CTR_INH_GS_IDC – C_CTR_INH_GS_IDC_DEC
        Else    LV_INH_GS_IDC_1 = 0
        Endif
Endif

If      LV_ERR_CRK = 1                          OR     // also transmitted on CAN for
                                                       // N-signal invalid
        LV_MTC_CUR_OFF = 1                      OR     // limp-home
        LV_ERR_VS_PLAUS = 1                     OR     // VS-error (PIN)
        LV_ERR_BN_VS_TCS = 1                    OR     // no VS-CAN-signal CAN12H
        LV_ERR_CAN_BOFF = 1                            // CAN-bus-off
Then    LV_INH_GS_IDC_2 = 1
Else    LV_INH_GS_IDC_2 = 0
Endif


If      LV_INH_GS_IDC_1 = 1        or
        LV_INH_GS_IDC_2 = 1
Then    GS_IDC_GEAR = C_GS_IDC_GEAR_SUB
        GS_IDC_DISP = C_GS_IDC_DISP_SUB
Else    GS_IDC_GEAR = GS_IDC_GEAR_RAW
        GS_IDC_DISP = GS_IDC_DISP_RAW
Endif
```

65.2 Gear ratio detection

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
GR_MT	V/O	0...FFH	0...255	1	[-]
Manual transmission gear					
GR_AT	V/O	0...FFH	0...255	1	[-]
Automatic transmission gear					
LV_RNG_L	V/O	0...1H	0...1	1	[-]
Logical variable for engaged low range					
LV_RNG_L_AT	V/O	0...1H	0...1	1	[-]
Logical variable for engaged low range AT					
N_VS_RATIO	O/V	0...FFH	0...255	1	[rpm/(km/h)]
engine speed / vehicle speed ratio, corrected by adaptation factor					
LV_DRI	V/O	0...1H	0...1	1	[-]
Logical variable for engaged drive					
FAC_GR	V	0...FFFFH	0...1.99996	0.0305e-3	[-]
Factor for setpoint/actual reduction ratio of rear axle					
GR_RAX_AV	V	0...FFFFH	0...255.99609	3.9063e-3	[-]
Current rear axle ratio					
FAC_GR_MMV_AV	V/O/S	0...FFFFH	0...1.99996	0.0305e-3	[-]
Rear axle ratio adaptation value					
GEAR	V/O	0...FFH	0...255	1	[-]
Common gear, independent from AT or MT					
GEAR_EF	V/O	0...FFH	0...255	1	[-]
Gear used for exhaust flap functionality					
STATE_GEAR_REV_AT_AMT	V/O	0...1H	0...1	1	[-]
AT/AMT reverse gear (activ = 1, inactive = 0)					

Input data:

VS	N	LV_ST	LV_ES
LV_ERR_VS	LV_CS	GR_SUB_MT	LV_AT
GEAR_INFO		LV_SOF_SWI_REQ	LV_PUC
LV_ERR_CS	LV_IM_BLS	LV_ERR_BLS_PLAUS	LC_AD_CLR_VAR
LV_ERR_CAN_BOFF	LV_ERR_BN_VS_TCS	LV_ERR_TOUT_ASR_1	STATE_ETCU_PROG_INFO
LV_CITY	LV_VAR_TCT	LV_VAR_ETCU_SPT	LV_ETCU_SPT_SWI

FUNCTION DESCRIPTION:

General information:

Gear ratio calculation is depending on transmission type (manual or automatic). When manual transmission is detected and (LV_AT == 0 && LV_VAR_TCT == 0), gear ratio is detected by using engine speed / vehicle speed ratio. In case of automatic transmission, gear ratio is detected by using the CAN-information of EGS. If LV_VAR_TCT = 1 then gear ratio is calculated versus CAN informations coming from DKG.

Application conditions:

Activation: at all engine operating states except LV_ES and LV_ST .

Deactivation: at engine operating states LV_ES and LV_ST .

Recurrence: 10 ms **except** subchapter 65.2.1.1: 100 ms

65.2.1 Manual transmission (LV_AT = 0 and LV_VAR_TCT = 0)

Application conditions:

Initialisation: GR_MT = 0 **at** [LV_ES = 1 **or** LV_ST = 1 **or** LV_CS = 1]

Formula section:

Executed only if LV_AT = 0 and LV_VAR_TCT = 0

```
if(1) LV_CS = 0
then(1)  if(2)     [ LV_ERR_VS = 1  and  LV_ERR_CAN_BOFF = 1 ]    or
                   [ LV_ERR_VS = 1  and  LV_ERR_BN_VS_TCS = 1 ]   or
                   [ LV_ERR_VS = 1  and  LV_ERR_TOUT_ASR_1 = 1 ]
         then(2)   N_VS_RATIO = 0
                   GR_MT = GR_SUB_MT
         else(2)   if(3)      VS > C_MIN_VS_RATIO
                   then(3)    N_VS_RATIO = $\frac{N}{VS}$
                   else(3)    N_VS_RATIO = 255 rpm/(km/h)
                   endif(3)

                                     GR_MT = ID_GR_MT__N_VS_RATIO
                             **)

         endif(2)
  else(1)  N_VS_RATIO = 0
           GR_MT = 0
  endif(1)
```

****)** $$\text{LDP\_N\_VS\_RATIO\_GR\_MT} = \frac{\text{N\_VS\_RATIO}}{\text{FAC\_GR\_MMV\_AV}}$$

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
ID_GR_MT__N_VS_RATIO	13	0...FFH	0...255	1	[-]
LDP_N_VS_RATIO_GR_MT	13	0...FFH	0...255	1	[rpm/(km/h)]
Index table for Gear Ratio detection					
C_MIN_VS_RATIO	1	0...FFH	0...255	1	[km/h]
Minimum engine speed for engine speed / vehicle speed ratio calculation					

65.2.1.1 Adaptation of different rear axle transmission ratios (only manual transmission)

Different rear axle ratios are used for the versions for different countries within the various vehicle series. The function aims at learning the rear axle ratio and to use it for correction for gear detection.
The gear detection is applied with the standard rear axle. Moreover, the standard reduction ratio is applied as fixed value. Moreover, the n/vs ratio used by the vehicle for driving in 2nd gear and trailing throttle fuel cut-off is applied.

Learning the new rear axle ratio
The adaptation of the rear axle ratio is only effected in 2nd gear and at trailing throttle fuel cut-off PUC. The quotient of the actual value and the setpoint of the n/vs ratio is used for calculation of the new rear axle ratio and the factor between the actual and setpoint reduction ratio.
The rear axle ratio is adapted via an adaptive correlation constant C_CRLC_FAC_GR.
Moreover, FAC_GR is biased with FAC_GR_MMV_AV.

Application conditions:

Initialisation: FAC_GR_MMV_AV = 1.0
when ECU is new or
when adaptation is reset by LC_AD_CLR_VAR or
when values of group VAR could not be read out from the non volatile memory;
else: FAC_GR_MMV_AV is read back from non volatile memory

N_VS_RATIO = 255 rpm/(km/h)

Formula section:

Executed only if (LV_AT = 0 and LV_VAR_TCT = 0)

if GEAR = 2 **and**
LV_PUC = 1 **and**
LV_CS = 0 **and**
LV_ERR_CS = 0 **and**
LV_IM_BLS = 0 **and**
LV_ERR_BLS_PLAUS = 0 **and**
$|\text{N\_VS\_RATIO}_{n-1} - \text{N\_VS\_RATIO}_{n}| < \text{C\_N\_VS\_CRU\_GRD\_FAC\_GR\_MAX}$

then *Calculation of the rear axle ratio "ACTUAL to SETPOINT values - rear axle"*

$$\text{FAC\_GR} = \frac{\text{N\_VS\_RATIO}}{\text{C\_N\_VS\_CRU\_GEAR2\_PUC\_SP}}$$

Adaptation of the rear axle ratio:

$$\text{FAC\_GR\_MMV\_AV}_{n} = \text{FAC\_GR\_MMV\_AV}_{n-1} + \text{C\_CRLC\_FAC\_GR} * (\text{FAC\_GR} - \text{FAC\_GR\_MMV\_AV}_{n-1})$$

Reduction ratio of the currently installed rear axle:

$$\text{GR\_RAX\_AV} = \text{C\_GR\_RAX\_SP} * \text{FAC\_GR\_MMV\_AV}$$

The calculated N_VS_RATIO ratio is continually divided by the current value of FAC_GR_MMV_AV and the current multiplication result is used to obtain the correct gear from the value applied map ID_GR_MT__N_VS_RATIO.

The adaptation value FAC_GR_MMV_AV can be reset to 1 via the KWP command "delete variant".

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_GR_RAX_SP	1	0...FFFFH	0...255.99609	3.91E-03	[-]
Rear axle ratio of standard rear axle					
C_CRLC_FAC_GR	1	0...FFH	0...0.99609	3.91E-03	[-]
Correlation constant for adaptation of the rear axle					
C_N_VS_CRU_GEAR2_PUC_SP	1	0...FFH	0...255	1	[rpm/(km/h)]
N/VS ratio in case of PUC in 2nd gear with standard rear axle					
C_N_VS_CRU_GRD_FAC_GR_MAX	1	0...FFH	0...255	1	[rpm/(km/h)]
Gradient of N/VS ratio below which FAC_GR is adapted					

65.2.2 Twin Clutch Transmission (LV_VAR_TCT = 1)

Gear ratio and LV_DRI are calculated versus CAN informations coming from DKG.

Application conditions:

Initialisation: GR_MT, LV_DRI = 0 **at** [LV_ES = 1 **or** LV_ST = 1]

Formula section:

Executed only if LV_VAR_TCT = 1

GR_MT, LV_DRI = f(GEAR_INFO), see list below:

GEAR_INFO	Corresponding gear	GR_MT	LV_DRI
0H	neutral or park	0	0
1H	1st gear	1	1
2H	2nd gear	2	1
3H	3rd gear	3	1
4H	4th gear	4	1
5H	5th gear	5	1
6H	6th gear	6	1
7H	7th gear	7	1
AH	reverse gear	0	1

65.2.3 Automatic transmission (LV_AT = 1)

Gear ratio and LV_DRI are calculated versus CAN informations coming from EGS.

Application conditions:

Initialisation: GR_AT, LV_DRI = 0 **at** [LV_ES = 1 **or** LV_ST = 1]

Formula section:

Executed only if LV_AT = 1

GR_AT, LV_DRI = f(GEAR_INFO), see list below:

GEAR_INFO	Corresponding gear	GR_AT	LV_DRI
0H	neutral or park	0	0
1H	1st gear	1	1
2H	2nd gear	2	1
3H	3rd gear	3	1
4H	4th gear	4	1
5H	5th gear	5	1
6H	6th gear	6	1
7H	7th gear	7	1
AH	reverse gear	0	1

65.2.4 Information if the reverse gear (AT or TCT) is activ/inactiv :

Formula section:

```
If      (LV_AT = 1 or LV_VAR_TCT = 1)
        and   GEAR_INFO = AH
then    STATE_GEAR_REV_AT_AMT = 1
else    STATE_GEAR_REV_AT_AMT = 0
endif
```

65.2.5 Low range detection (LV_RNG_L, LV_RNG_L_AT)

Depending on LC_RNG_L, the sport-switch and the LV_RNG_L_AT-request from EGS, the bit LV_RNG_L will be activated.

Meaning of C_STATE_ETCU_PROG_INFO:

Bit	LV_RNG_L_AT will be set at condition:
0	*AT / TCT in "S" / "DS" ("comfort transmission"; LV_VAR_ETCU_SPT=0)*
1	*AT / TCT in manual mode "M1 ... 6" ("comfort transmission"; LV_VAR_ETCU_SPT=0)*
2	*AT- / TCT-sport-switch in "S" / "DS" ("sport transmission"; LV_VAR_ETCU_SPT=1)*
3	*AT- / TCT-sport-switch in manual mode "M1 ... 6" ("sport transmission"; LV_VAR_ETCU_SPT=1)*
4	*Gearbox-sportswitch active and AT / TCT ("comfort transmission"; LV_VAR_ETCU_SPT=0)*
5	*Gearbox-sportswitch active and AT / TCT ("sport transmission"; LV_VAR_ETCU_SPT=1)*
6	*always with MT*
7	*not used*

Formula section:

If

AT / TCT "comfort" in "S-mode"

[STATE_ETCU_PROG_INFO = 1H **and**

LV_VAR_ETCU_SPT = 0 **and**

Bit 0 of C_STATE_ETCU_PROG_INFO is set**]** **or**

AT / TCT "comfort" in "M-mode"

[STATE_ETCU_PROG_INFO = 2H **and**

LV_VAR_ETCU_SPT = 0 **and**

Bit 1 of C_STATE_ETCU_PROG_INFO is set**]** **or**

AT/TCT "sport" in "S-mode"

[STATE_ETCU_PROG_INFO = 1H **and**

LV_VAR_ETCU_SPT = 1 **and**

Bit 2 of C_STATE_ETCU_PROG_INFO is set**]** **or**

AT/TCT "sport" in "M-mode"

[STATE_ETCU_PROG_INFO = 2H **and**

LV_VAR_ETCU_SPT = 1 **and**

Bit 3 of C_STATE_ETCU_PROG_INFO is set**]** **or**

AT / TCT "comfort" in "sport-mode-gearbox"

```
        [LV_VAR_ETCU_SPT = 0                           and
        LV_ETCU_SPT_SWI = 1                            and
        Bit 4 of C_STATE_ETCU_PROG_INFO is set]                 or
```

AT / TCT "sport" in "sport-mode-gearbox"

```
        [LV_VAR_ETCU_SPT = 1                           and
        LV_ETCU_SPT_SWI = 1                            and
        Bit 5 of C_STATE_ETCU_PROG_INFO is set]                 or
```

MT continious in "sport"-mode

```
        [LV_AT = 0 and LV_VAR_TCT = 0 and
        Bit 6 of C_STATE_ETCU_PROG_INFO is set]


Then    LV_RNG_L_AT = 1
Else    LV_RNG_L_AT = 0
Endif


if      LC_RNG_L = 1            or    // request from application system
        LV_SOF_SWI_REQ = 1      or    // request from sport switch
        LV_RNG_L_AT = 1               // request from gearbox
  then  LV_RNG_L = 1
  else  LV_RNG_L = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_RNG_L	1	0...1H	0...1	1	[-]
Manual switch for low range detection					
C_STATE_ETCU_PROG_INFO	1	0...FFH	0...255	1	[-]
Manual switch for activating low range AT detection					

65.2.6 Common Gear Ratio Independent from AT or MT (GEAR)

Description:

At not stopped or starting engine, GEAR displays the actual gear ratio independent from transmission type.

Application conditions:

Initialisation: GEAR = 0

Formula section:

```
If (1)          LV_ES = 0
    and         LV_ST = 0
  then (1)
    if (2)      LV_AT = 1
      then (2)  GEAR = GR_AT
      else (2)  GEAR = GR_MT
    endif (2)
  else (1)      GEAR = 0
endif (1)
```

65.2.7 Gear Ratio for exhaust flap functionality

Description:

For the exhaust flap function the detected gear shall not change during gear shift. For this purpose the variable GEAR_EF is generated from GEAR.

Application conditions:

Initialisation: GEAR_EF = 0

Formula section:

If LV_CS = 0

```
    then    GEAR_EF  =  GEAR        no gear shift because clutch is closed
    else    GEAR_EF  =  GEAR_EF     gear shift possible because open clutch, GEAR_EF remains
                                    unchanged
  endif
```

65.3 Drive Train Engaged

Output Data:

Name	Mode	Hex. Limits	Phys. Limits	Resol.	Unit
LV_DT	O/V	0... 1H	0... 1	1	[-]
Logical variable drive train engaged					

Input Data:

GR_MT	LV_AT	LV_CS	LV_DRI
LV_ES	VS	LV_VAR_TCT	

Calibration Data:

Name	Dim	Hex.Limits	Phys.Limits	Resol.	Unit
C_VS_MIN_DT	1	0... FFH	0... 255	1	[km/h]
Threshold for vehicle running and drive-train closed detection					
LC_MOD_DT_TCT	1	0... 1H	0... 1	1	[-]
Mode selection of drive train engaged detection for variant twin clutch transmission (0 = like MT; 1 = like AT)					

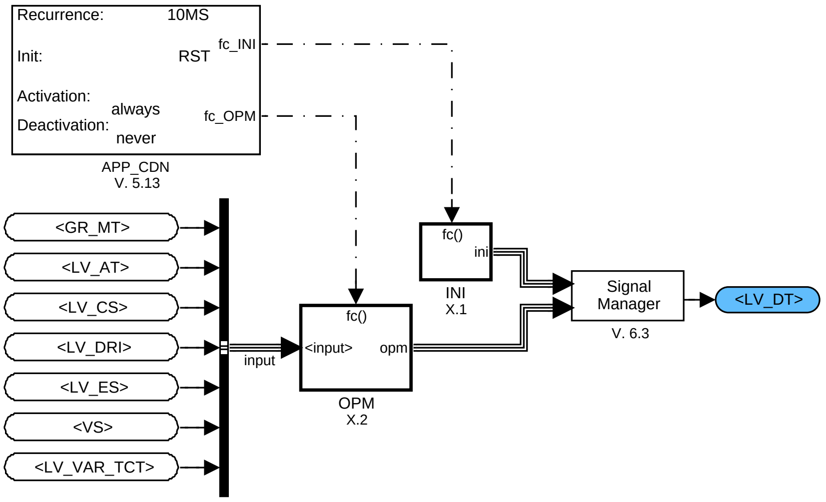
General Information

General information:
Drive train engaged active (LV_DT=1) means drive train is closed, engine torque is transmitted to the wheels and the vehicle is running (VS >= C_VS_MIN_DT).
The depedabillity to the vehicle-speed has different reason for AT and MT/AMT vehicle.
AT:
With engaged gear it´s normally not necessary to have an agressive idle-speed control. The only exception is a sudden breaking at slowly moving vehicle with engaged gear. In this case it´s necessary to switch to the strong P_D - parameter map of the idle-speed control to avoid idle-speed break-down.
MT:
At start of the vehicle without pushing the accelerator pedal it´s necessary to switch as fast as possible to the smooth P_D idle-speed control map to avoid vehicle jerking. Depending on slip of the clutch it may last too long until a gear-position determination is possible. Therefore in this case the switching from agressive to the smooth P_D map can happen even at slow vehicle speed.
Nevertheless the actualisation of LV_DT is strongly depending on the update-rate of the input signal VS .

Application Conditions

Initialization: RST
Recurrence: 10MS
Activation: always
Deactivation: never

Function description



x SDA_SRS / SDA V 5.2 30−Oct−2006

Figure 1:
Path: TRSM_M400T

65.3.1 Initialisation at Reset

LV_DT = 0

65.3.2 Formula section

65.3.2.1 Calculation of LV_DT

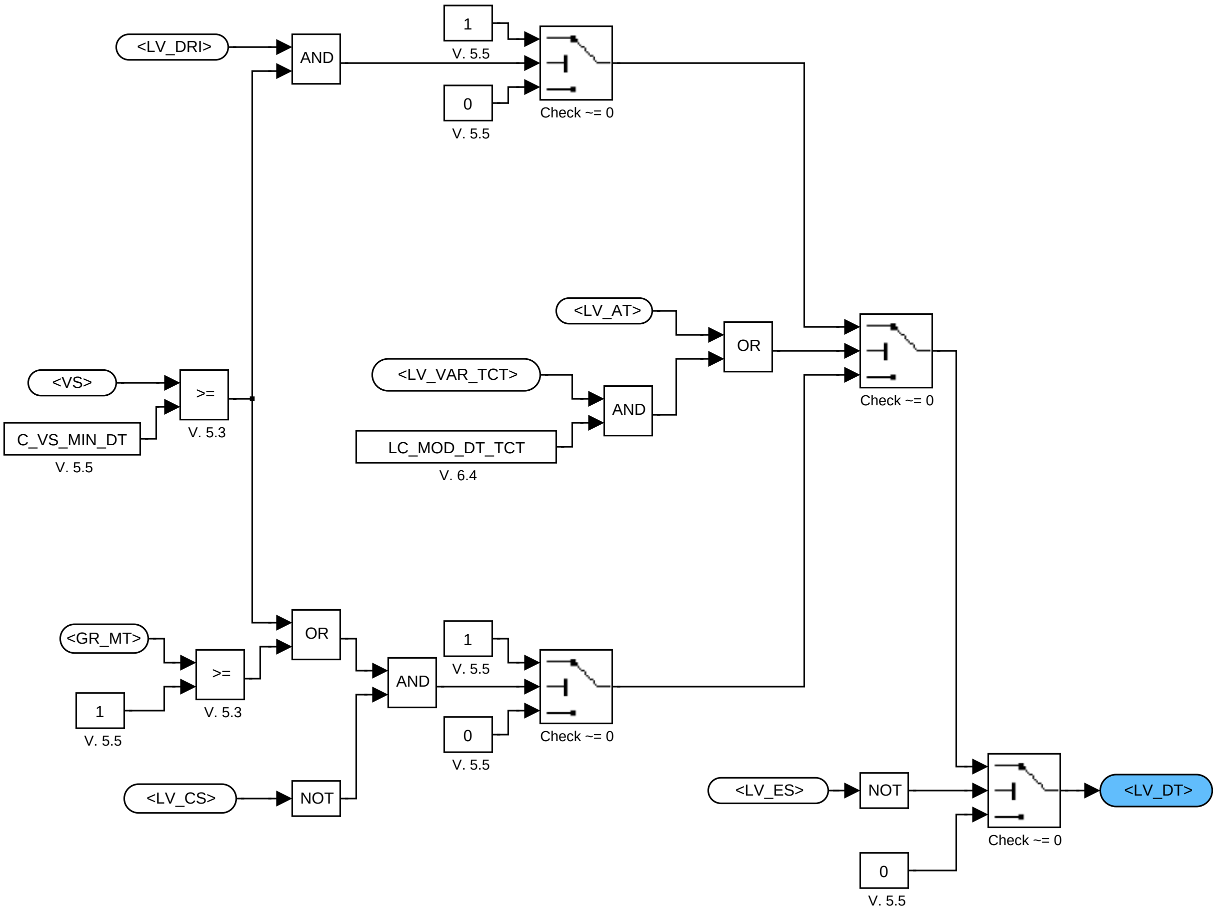


Figure 2:
Path: TRSM_M400T/OPM/CLC_LV_DT

65.4 Clutch Switch detection

Output data:

Name	V/S	Hex. limits	Phys. limits	Resol.	Unit
LV_CS_CUS	O/V	0...1H	0...1	1	[-]
Boolean for clutch switch detection					

Input data:

LV_IM_CS_PN	LV_AT		
STATE_ETCU_CLU	LV_VAR_TCT		

General information:

Clutch switch detection is required for function management at transient conditions. It´s only necessary for cars with manual transmission.
At determination of the ´real´ clutch-switch closing (LV_IM_CS_PN=1) LV_CS_CUS resets at change of LV_IM_CS_PN from 1 → 0 or STATE_ETCU_CLU 03H/4H/5H.

Application conditions:

Activation: LV_AT = 0

Deactivation : LV_AT = 1

Initialisation: **if** LV_AT=0 and LV_VAR_TCT = 0
then LV_CS_CUS = LV_IM_CS_PN

time recurrency : 10ms

Formula section:

```
if    LV_VAR_TCT = 0
  then      LV_CS_CUS  =  LV_IM_CS_PN
  elseif    STATE_ETCU_CLU = 3H or 4H or 5H
        then      LV_CS_CUS  =  0
        else      LV_CS_CUS  =  1
      endif
endif
```

65.5 AT/ TCT diagnosis (LV_ERR_GS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_GS	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on GS					
LV_CDN_DIAG_GS	V/O	0...1H	0...1	1	[-]
Diagnosis condition GS					
ERR_SYM_GS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom GS					
LV_END_DIAG_GS	V/O	0...1H	0...1	1	[-]
End of diagnosis cycle GS diagnosis					
STATE_DIAG_GS	V/O	0...FFH	0...255	1	[-]
Satus of GS diagnosis for feedback to gearbox					

Input data:

LV_IGK	LV_DC	LV_WUP_CYC	STATE_ETCU_OBD
	LV_VAR_BN		LV_AT
LV_ERR_MEM_XX	LV_ERR_CFM[NC_NR_ERR_DYN]	WAL_CONF_XX	LV_DC_RBM
LV_CDN_VB_OBD1	LV_VAR_TCT	LV_MIL_REQ_ETCU	

Import actions:

ACTION_ERRM_ReadFrfByDtc

FUNCTION DESCRIPTION:

General information:

The diagnosis is performed for AT or TCT vehicles in order to manage the OBD requirement which is defined by law (MIL activation and Freeze – frame storage).
Depending of CAN 12H / 11H and TCT/ AT and LC_ETCU_FRF the diagnosis is performed in a different way.

Description:

- Request of Gearbox to ECU: STATE_ETCU_OBD
- Response from ECU to Gearbox: STATE_DIAG_GS

Calculation of STATE_DIAG_GS (only first 3 bits, only valid for LC_ETCU_FRF = 1):

	Case 1	**Case 2**	**Case 3**	**Case 4**
Init	No FRF **000**	FRF not from GS **001**	FRF from GS **010**	FRF from GS **010**
Event	GS error	GS error	OBD error	OBD error with higher prior (FSD / MSF)
Result	FRF from GS **010**	FRF not from GS **001**	FRF from GS **010**	FRF from OBD error **001** (next DC)

Application conditions:

```
Initialisation:   according ABC configuration "STD_INI" + NLC_ABC_CAL_DEC_GS = 1
                  STATE_DIAG_GS = 0 at LV_IGK 0->1 or reset or at clearing error memory

Recurrence:       10ms

Activation:       If      (LV_AT = 1 or LV_VAR_TCT = 1) and
                          LV_CDN_VB_OBD1 = 1
                  Then    LV_CDN_DIAG_GS = 1
                  Else    LV_CDN_DIAG_GS = 0
                  Endif
Dectivation:      If      LV_AT = 0 and LV_VAR_TCT = 0
                  Then    LV_END_DIAG_GS = 1        //always set for readiness CARB_CC
                           LV_CDN_DIAG_GS = 0
                  Endif
```

Formula section:

Calculation of STATE_DIAG_GS

```
If(1) LC_ETCU_FRF = 1
Then(1)
If    LV_ERR_MEM_GS = 0       and
      LV_ERR_GS = 0
ThenIf     at least one LV_ERR_CFM[NC_NR_ERR_DYN]= 1        and
           Bit 4 of WAL_CONF_XX of this error is calibrated
     Then  If      LV_VAR_BN = 0
           Then    STATE_DIAG_GS = set bit  ( 001 - - - )  //FRF stored for OBD error, 11H
           Else    STATE_DIAG_GS = set bit  ( 001 - - )  //FRF stored for OBD error, 12H
           Endif
     Else  If      LV_VAR_BN = 0
           Then    STATE_DIAG_GS = set bit  ( 000 - - - )  //no FRF stored, 11H
           Else    STATE_DIAG_GS = set bit  ( 000 - - )  //no FRF stored, 12H
           Endif
ElseIf     LV_ERR_GS  0->1    and              //request from EGS to store GS error
           LV_ERR_MEM_GS (n-1) = 0             //not stored yet
```

```
Then      If      at least one LV_ERR_CFM[NC_NR_ERR_DYN] (n-1) = 1   and
                  Bit 4 of WAL_CONF_XX of this error is calibrated
          Then If      LV_VAR_BN = 0
               Then    STATE_DIAG_GS = set bit  ( 001 - - - ) //FRF stored for OBD error, 11H
               Else    STATE_DIAG_GS = set bit  ( 001 - - ) //FRF stored for OBD error, 12H
               Endif
          Else If      LV_VAR_BN = 0
               Then    STATE_DIAG_GS = set bit   (010 - - - )  //FRF stored for GS, 11H
                       Call of error memory to store LV_ERR_GS
               Else    STATE_DIAG_GS = set bit   (010 - - )  //FRF stored for GS, 12H
                       Call of error memory to store LV_ERR_GS
               Endif
          Endif
ElseIf    LV_IGK 0->1 or reset
     Then ACTION_ERRM_ReadFrfByDtc
          If      Resp XX = GS          //FRF still stored for LV_ERR_GS
          Then If      LV_VAR_BN = 0
               Then    STATE_DIAG_GS = set bits ( 010 - - - )  //FRF stored for GS, 11H
               Else    STATE_DIAG_GS = set bits ( 010 - - )  //FRF stored for GS, 12H
               Endif
          Else If      LV_VAR_BN = 0
               Then    STATE_DIAG_GS = set bit  ( 001 - - - ) //FRF stored for OBD error, 11H
               Else    STATE_DIAG_GS = set bit  ( 001 - - )  //FRF stored for OBD error, 12H
               Endif
Else      no change of bits x in STATE_DIAG_GS (xxx - - )
Endif
Else(1)   If      LV_VAR_BN = 0
          Then    STATE_DIAG_GS = set bit  ( 000 - - - ) //no FRF at all
          Else    STATE_DIAG_GS = set bit  ( 000 - - )  //no FRF at all
          Endif
Endif(1)

If    LV_DC = 1
ThenIf    LV_VAR_BN = 0
     Then STATE_DIAG_GS = set bit  ( - - - 1 - - )  //Driving cycle active, 11H
     Else STATE_DIAG_GS = set bit  ( - - - 1 - )  //Driving cycle active, 12H
```

```
        Endif
    Endif


    If    LV_WUP_CYC = 1
    ThenIf       LV_VAR_BN = 0
        Then   STATE_DIAG_GS = set bit  (- - - - - 1 )  //Warm-up cycle active, 11H
        Else   STATE_DIAG_GS = set bit  (- - - - 1 )  //Warm-up cycle active, 12H
        Endif
    Endif


    If    LV_DC_RBM = 1
    ThenIf       LV_VAR_BN = 0
        Then   STATE_DIAG_GS = set bit  (- - - - 1-)  //RBM cycle active, 11H
        Endif
    Endif
```

// Remark: in case of LV_VAR_BN = 1 the information of LV_DC_RBM is transmitted via CAN in a separate signal, not within STATE_DIAG_GS; for detailled information see CAN-messages BN2000

Calculation of STATE_ETCU_OBD

```
If(0)        LV_VAR_TCT = 0
Then(0)
If(1a) LC_ETCU_FRF = 1
Then(1a)
If(1b) LC_VAR_GS_EOBD = 0
Then(1b)   If(2)LV_AT = 1
           Calculation is done for - EGS - Configuration US-market
           Then(2)If     STATE_ETCU_OBD = 0110 or 0100
                  Then   ERR_SYM_GS = SYM_3
                         Anti-bounce counter increments, set LV_ERR_GS after debounce
                  Elseif STATE_ETCU_OBD = 0000 or 0010
                  Then   ERR_SYM_GS = NO_SYM
                         Anti-bounce counter decrements with calibratable decrement,
                         reset LV_ERR_GS after debounce
                  Else   no change
                  Endif


           Endif(2)
```

```
    Else(1b) If(2a)  LV_AT = 1
             Calculation is done for – EGS – Configuration EOBD
             Then(2a)If   STATE_ETCU_OBD = 0010
                    Then    ERR_SYM_GS = SYM_3
                            Anti-bounce counter increments, set LV_ERR_GS after debounce
                    Elseif  STATE_ETCU_OBD = 0000
                    Then    ERR_SYM_GS = NO_SYM
                            Anti-bounce counter decrements with calibratable decrement,
                            reset LV_ERR_GS after debounce
                    Else    no change
                    Endif
             Endif(2a)
    Endif(1b)
  Else(1a)
             If     LV_AT = 1
             Calculation is done for – EGS
             Then   If      STATE_ETCU_OBD = 0100
                    Then    ERR_SYM_GS = SYM_3
                            Anti-bounce counter increments, set LV_ERR_GS after debounce
                    Elseif  STATE_ETCU_OBD = 0000
                    Then    ERR_SYM_GS = NO_SYM
                            Anti-bounce counter decrements with calibratable decrement,
                            reset LV_ERR_GS after debounce
                    Else    no change
                    Endif

             Endif
  Endif(1a)
  Else(0)
      If     LV_MIL_REQ_ETCU = 1
      Then   ERR_SYM_GS = SYM_3
             Anti-bounce counter increments,
             set LV_ERR_GS after debounce
      Else   ERR_SYM_GS = NO_SYM
             Anti-bounce counter decrements with calibratable decrement,
             reset LV_ERR_GS after debounce
```

Endif

Endif(0)

For the calculation of the LV_END_DIAG_GS, see "Anti-bounce-Algorithm, calculation of the end of diagnosis"

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_VAR_GS_EOBD	1	0...1H	0...1	1	[-]
Logical constant for variant EOBD (=1) or US (=0)					
C_ABC_INC_GS	1	0...FFH	0...255	1	[-]
Anti ¿bounce increment for GS diagnosis					
C_ABC_MAX_GS	1	1...FFH	1...255	1	[-]
Anti ¿bounce maximum for GS diagnosis					
C_ABC_DEC_GS	1	0...FFH	0...255	1	[-]
Anti ¿bounce decrement for GS diagnosis					
LC_ETCU_FRF	1	0...1H	0...1	1	[-]
Logical constant for storing freeze frame for ETCU (1 = FRF stored for ETCU)					

65.6 Clutch switch diagnosis (LV_ERR_CS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CDN_DIAG_CS	V/O	0...1H	0...1	1	[-]
Diagnosis condition CS					
ERR_SYM_CS	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Detected CS Symptom					
LV_ERR_CS	V/O	0...1H	0...1	1	[-]
Active CS error					
LV_END_DIAG_CS	V/O	0...1H	0...1	1	[-]
End of CS diagnosis					
LV_CS_DIAG	V	0...1H	0...1	1	[-]
Flag set when vehicle was stopped = pre-condition for diagnosis cycle					
LV_CS_SWI	V	0...1H	0...1	1	[-]
Flag set when CS change its state					
T_MIN_MAF_CS_DIAG	V	0...FFH	0...255	1	[s]
Minimum diagnosis time for MAF condition					

Input data:

LV_IGK	LV_IM_CS_PN	N_32	LV_AT
MAF	VS	C_VS_MIN_RUN	
LV_ERR_VS	LV_ST	LV_ES	LV_VAR_TCT

FUNCTION DESCRIPTION:

General information:

This module performs the diagnosis of the clutch switch. This is valid only in case of manual transmission.

The diagnosis is based on the verification of the presence of the expected clutch action to shift the gears necessary to bring the vehicle to a certain speed. It is highly improbable that the vehicle is accelerated several consecutive times to a certain speed with the clutch always depressed or never depressed.

Description:

If there’s a a change in LV_IM_CS_PN, the CS is diagnosed as error-free and the diagnosis is switched off if end of diagnosis is reached.

If there’s no change in LV_IM_CS_PN then each time the vehicle is stopped, the diagnosis is enabled. Until the vehicle gets to a certain speed, LV_IM_CS_PN is monitored for value change (clutch pedal actuation). If, for several times (Anti bounce is inrementing by one inrement per diagnosis cycle) the vehicle gets to speed and the clutch switch state has never changed (always 0 or 1) then a failure is recognized.

Depending of status LV_IM_CS_PN 0/1 at beginning of monitoring period there is the symtom OL/SCB or SCG.

Application conditions:

Initialisation: according filter type: ***"STD-INI",*** all other outputs 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 1s (10 ms for LV_IM_CS_PN detection)

Activation:

```
If      LV_IGK = 0                   or
        LV_ES = 1                    or
        LV_ST = 1                    or
        LV_AT = 1                    or
        LV_VAR_TCT= 1                or
        LV_END_DIAG_CS = 1           or
        LV_ERR_VS = 1                // VS error
Then    Function passive   LV_CDN_DIAG_CS is reset to 0
Else    Function active
Endif
```

Formula section:

LV_CDN_DIAG_CS = 0

Errorfree case: CS activitation detected:

```
If      LV_IM_CS_PN   switched once from 1 to 0 or 0 to 1 (managed each 10 ms)
Then    LV_CDN_DIAG_CS = 1
        ERR_SYM_CS = NO_SYM
        LV_CS_SWI = 1
        LV_END_DIAG_CS = 1           Calculated by error management
Else
```

Error case: High vahicle speed without CS activation

```
    If      VS ≤ C_VS_MIN_RUN            Init when vehicle is stopped
    Then    LV_CS_DIAG = 1
    Else If     VS > C_VS_CS_DIAG        and
                MAF > C_MAF_CS_DIAG      and   above zero no load line
                N_32 > C_N_32_CS_DIAG          out of idle speed
        Then    increment T_MIN_MAF_CS_DIAG
                If      T_MIN_MAF_CS_DIAG >= C_T_MIN_MAF_CS_DIAG      and
                        LV_CS_SWI = 0    and   the clutch did not switched
                        LV_CS_DIAG = 1         vehicle was stopped
                Then    LV_CDN_DIAG_CS = 1
                        If      LV_IM_CS_PN = 1
```

```
                Then  ERR_SYM_CS = SYM_0
                If    LV_IM_CS_PN = 0
                Then  ERR_SYM_CS = SYM_1
                Endif
                LV_CS_DIAG = 0
                // vehicle must stop again before next diagnosis CTR_ABC_CS$_n$ = CTR_ABC_CS$_{n-1}$ + C_ABC_INC_CS     only once
            Else  no Action
            Endif
        Else  T_MIN_MAF_CS_DIAG = 0     reset
        Endif
    End if
Endif
```

Calculation of LV_END_DIAG_CS and LV_ERR_CS

LV_ERR_CS	Calculated by error management : CTR_ABC_CS reach maximum
LV_END_DIAG_CS	Calculated by error management

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_CS_DIAG	1	0...FFH	0...255	1	[km/h]
Speed threshold for activation of diagnosis cycle					
C_T_MIN_MAF_CS_DIAG	1	0...FFH	0...255	1	[s]
Minimum diagnosis time for MAF condition					
C_MAF_CS_DIAG	1	0...FFFFH	0...1389	2.12E-02	[mg/stk]
MAF threshold for error detection					
C_N_32_CS_DIAG	1	0...FFH	0...8160	32	[rpm]
N32 threshold for error detection					
C_ABC_INC_CS	1	0...FFH	0...255	1	[-]
Anti bouce increment CS diagnosis					
C_ABC_MAX_CS	1	1...FFH	1...255	1	[-]
Anti bouce maximum CS diagnosis					

65.7 Clutch switch diagnosis (appl. inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_CS	V/O	0...7H	0...7	1	[-]
Interface of CS monitor with the Rate-Based Monitoring statisticsBit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1)Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1)Bit 2: individual RBM condition					

Input data:

LV_DC	CTR_ERR_DYN_NR	LV_END_DIAG_CS	LV_ERR_VS
LV_AT		LV_VAR_TCT	

Import actions:

ACTION_ERRM_CheckPendingStatus (IN <XX>, OUT<PendingStatus>)

FUNCTION DESCRIPTION:

General information:

With this module the interface between the CS monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_CS data.

Within STATE_RBM_CS, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)

  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)

  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

  (not valid for CS diagnosis)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_CS = 0

on failure memory reset :

bit 1 of STATE_RBM_CS = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

At LV_DC 0 → 1 transition

The pending status of the following failures has to be checked only once :

LV_ERR_VS			

```
If(1)      { CPU optimization at LV_DC 0 → 1 transition }
           CTR_ERR_DYN_NR <> 0    { the dynamic failure memory isn't empty }
Then(1)
           While bit 1 of STATE_RBM_CS = 0 do
                 with each XX failure of the above list :
                 ACTION_ERRM_CheckPendingStatus(IN<XX>, OUT<PendingStatus>,
                 SYNCHRONIZATION<CALL>)
                 If(2)        XX has a pending status
                 Then(2)
                              bit 1 of STATE_RBM_CS = 1
                 Endif(2)
           Endwhile
Else(1)
           { the dynamic failure memory is empty }
           No action
Endif(1)
```

Every 1 s :

```
If    bit 0 of STATE_RBM_CS = 0
Then       If    LV_END_DIAG_CS = 1     or
                 LV_AT = 1              or
                 LV_VAR_TCT = 1
           Then  bit 0 of STATE_RBM_CS = 1
           Endif
Endif
If    bit 1 of STATE_RBM_CS = 0
Then
           If    LV_ERR_VS = 1
           Then  bit 1 of STATE_RBM_CS = 1
           Endif
Endif
bit 2 of STATE_RBM_CS = 1
```

65.8 Torque request for gear shift intervention

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ETCU_DISABLE	O/V	0...1H	0...1	1	[-]
Logical variable for disabling GS intervention due to EMS error					
LV_ETCU_DISABLE_CAN	O/V	0...1H	0...1	1	[-]
Logical variable for disabling GS intervention due to CAN error					
LV_GS_ENA_DEC	O/V	0...1H	0...1	1	[-]
Logical variable for decreased torque intervention due to gear shift					
LV_GS_ENA_INC	O/V	0...1H	0...1	1	[-]
Logical variable for increased torque intervention due to gear shift					
LV_GS_ENA	V	0...1H	0...1	1	[-]
Logical variable for gear shift intervention enabling					
STATE_TQ_ETCU_PLAUS	O/V	0...FFH	0...255	1	[-]
Bitwise coded State for ETCU intervention state					
TQI_GS_FAST_DEC_REQ	V	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque decrement for fast torque intervention during gear shift					
TQI_GS_SLOW_DEC_REQ	V	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque decrement for slow torque intervention during gear shift					
TQI_GS_FAST_INC_REQ	V	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque increase for fast torque intervention during gear shift					
TQI_GS_SLOW_INC_REQ	O/V	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque increase for slow torque intervention during gear shift					
TQI_GS_FAST_DEC_REQ_1	O	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque decrement for fast torque intervention during gear shift					
TQI_GS_SLOW_DEC_REQ_1	O	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque decrement for slow torque intervention during gear shift					
TQI_GS_FAST_INC_REQ_1	O	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque increase for fast torque intervention during gear shift					
TQI_GS_SLOW_INC_REQ_1	O	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque increase for slow torque intervention during gear shift					
T_TQR_GS_INTV	V	0...FFFFH	0...655.35	0.01	[s]
Duration of gear shift torque intervention					
TQI_GS_FAST_DEC	O/V	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque decrement for fast torque intervention during gear shift					
TQI_GS_SLOW_DEC	O/V	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque decrement for slow torque intervention during gear shift					
LV_GS_DEC_ACT	O/V	0...1H	0...1	1	[-]
Logical variable for torque intervention due to gear shift					
LV_GS_DEC_LIH	V	0...1H	0...1	1	[-]
Logical variable for torque intervention LIH due to gear shift					
T_GS_DEC_LIH_CTR	V	0...FFH	0...2.55	0.01	[s]
Delay time for activation of GS emergency ramp function					
TQI_GS_FAST_INC	O/V	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque increase for fast torque intervention during gear shift					
TQI_GS_SLOW_INC	O/V	0...7FFFH	0...1023.97	0.03125	[Nm]
Torque increase for slow torque intervention during gear shift					
LV_GS_INC_ACT	O/V	0...1H	0...1	1	[-]
Togical variable for torque intervention due to gear shift					
LV_GS_INC_LIH	O/V	0...1H	0...1	1	[-]
Logical variable for torque intervention LIH due to gear shift					
T_GS_INC_LIH_CTR	V	0...FFH	0...2.55	0.01	[s]
Delay time for activation of GS emergency ramp function					
LV_INH_FCUT_GS	O/V	0...1H	0...1	1	[-]
Logical variable inhibiting FCUT during TCT-intervention					

Input data:

LV_TQI_REQ_CAN_INH	LV_IGK	LV_VAR_BN	TQ_GS_FAST_DEC_BN
LV_ERR_BN_TQ_ETCU	LV_ST	TQ_GS_FAST_INC_BN	TQ_GS_SLOW_DEC_BN
LV_ERR_BN_ETCU	LV_ES		TQ_LOSS
	LV_TQ_GS_INC_REQ	LV_TQ_GS_DEC_REQ	TQ_GS_SLOW_BN
LV_ERR_CAN_BOFF	TQI_GS_SLOW_REQ	TQI_GS_SLOW_REQ_CAN	TQ_GS_SLOW_INC_BN
LV_ERR_TOUT_ETCU_1	LV_AT	LV_ETCU_LIH_CAN	TQ_GS_FAST_BN
LV_ERR_TOUT_ETCU_2	LV_TQI_REQ_CAN_INH	TQI_GS_FAST_REQ	TQI_GS_FAST_REQ_CAN
LV_DI_TQ_REQ_CAN_MPI_GDI	LV_CDN_VB_CAN_TQ_DIAG	LV_ERR_BN_TQ_TCT	LV_VAR_TCT
LV_TCT_LIH_CAN			

FUNCTION DESCRIPTION:

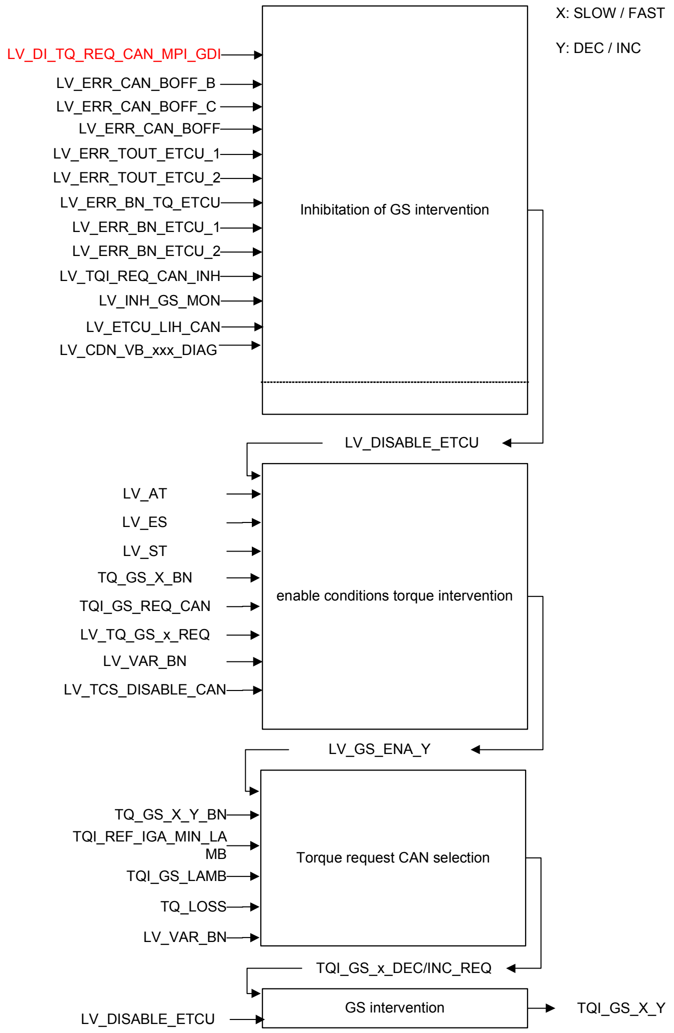
General information:

There are two different gearbox control units: the common electronic transmissiom control unit (LV_AT) and the twin clutch transmission control unit (LV_VAR_TCT). Depending which variant is active the torque request for gear shift is requested by AT or TCT.

During a gear shift operation of the automatic transmission a low engine torque helps the transmission to perform a smooth gear shift and avoids sudden changes of wheel slip that could cause problems on certain road surfaces, i.e. wet or icy roads. A transmission control unit (ETCU) torque reduction or increase demand via CAN is realized by spark retard (fast path) or charge intervention by ETC (slow path). A decision can be done which path is realized. Normally the intervention is done only by spark retard. By configuration of BN 2000 the Slow and fast path are own signals of CAN messages.

Some plausibilities are done for validation and limitation of torque requests.

Signal flow diagram:



65.8.1 Enable conditions

65.8.1.1 Inhibitation GS intervention

FUNCTION DESCRIPTION:

If the EMS detects a safety relevant error at throttle, no automatic transmission vehicle or pedal value sensor the logical variable LV_ETCU_DISABLE is set to 1 in order to indicate a static EMS error.

Gear shift intervention function is enabled if engine state is not “engine stopped”, “start” or “idle speed”, if LV_ETCU_DISABLE is not set, no CAN error is present and automatic transmission is detected. Additional a time condition during torque intervention is check which limits the duration of torque intervention.

Application conditions:

Initialisation: LV_ETCU_DISABLE /CAN = 0; **at reset**

Recurrence: 10 ms

Activation: (LV_AT = 1 **or** LV_VAR_TCT = 1) **and** LV_IGK = 1

Deactivation: -

Formula section:

CAN enviroment error GS inhibition:

```
if      LV_ERR_CAN_BOFF          = 1     or
        LV_ERR_TOUT_ETCU_1       = 1     or
        LV_ERR_TOUT_ETCU_2       = 1     or
        LV_ERR_BN_TQ_ETCU        = 1     or

        LV_ERR_BN_TQ_TCT         = 1     or
        LV_ERR_BN_ETCU           = 1     or
        LV_CDN_VB_CAN_TQ_DIAG    = 0


then          LV_ETCU_DISABLE_CAN        = 1
else          LV_ETCU_DISABLE_CAN        = 0
endif
```

Generell GS inhibition:

```
if            LV_TQI_REQ_CAN_INH         = 1   or
              LV_DI_TQ_REQ_CAN_MPI_GDI   = 1   or
              LV_ERR_GS_INH_MON          = 1   or
              LV_ETCU_DISABLE_ CAN  = 1   or
              LV_ETCU_LIH_CAN       = 1   or
              LV_TCT_LIH_CAN        = 1

then          LV_ETCU_DISABLE = 1
else          LV_ETCU_DISABLE = 0
endif
```

65.8.1.2 enable conditions torque intervention

FUNCTION DESCRIPTION:

Depending on the configuration and plausible torque interfaces an inc. or decreased torque intervention is possible.

The State STATE_TQ_ETCU_PLAUS indicates the interrupted condition which is responsible for enable Bit = 0. The [x] Value shows the HEX adress inside the Bitwise coded State.

Application conditions:

Initialisation: LV_GS_ENA/_INC/_DEC = 0 **at reset**

Recurrence: 10 ms

Activation: (LV_AT = 1 **or** LV_VAR_TCT = 1) **and**
LV_IGK = 1 and
C_GS_ENA_INC_DEC = not 0H

Formula section:

C_GS_ENA_INC_DEC =		
	No	no GS Torque intervention
	DEC	only decreasing torque intervention possible
	INC	only increasing torque intervention possible
	DEC_INC	both interventions possible

Enable conditions for GS intervention:

```
if    (LV_AT = 1 or LV_VAR_TCT = 1) and
      LV_ES = 0                     and
      LV_ST = 0                     and
      LV_ETCU_DISABLE = 0
then        LV_GS_ENA = 1
else        LV_GS_ENA = 0
endif
```

Torque decrease intervention:

```
IF(1)       (C_GS_ENA_INC_DEC = 1H or 3H )     and
            LV_GS_ENA =1                       and
            LV_TQ_GS_DEC_REQ = 1               and
            ((T_TQR_GS_INTV < C_T_MAX_TQR_GS_INTV) OR LV_AT = 0)
Then(1) IF(2)     LV_VAR_BN = 1
```

```
    Then(2) IF(3) TQ_GS_SLOW_BN   <> 8000H       and
    TQ_GS_FAST_BN    <> 8000H
        Then (3)       LV_GS_ENA_DEC = 1
        Else(3)        LV_GS_ENA_DEC = 0
        Endif (3)
    Else(2) IF(4) TQI_GS_SLOW_REQ_CAN ≠ FFH   and
                  TQI_GS_FAST_REQ_CAN ≠ FFH
        Then(4)        LV_GS_ENA_DEC = 1
        Else(4)        LV_GS_ENA_DEC = 0
        Endif(4)
    Endif(2)
Else(1)     LV_GS_ENA_DEC = 0
Endif
```

Torque increase intervention:

```
IF(1)        C_GS_ENA_INC_DEC = 2H or 3H                  and
             LV_GS_ENA  =1                                and
             LV_TQ_GS_INC_REQ = 1
Then(1) IF(2)     LV_VAR_BN = 1
    Then(2) IF(3) TQ_GS_SLOW_BN   <> 8000H       and
                  TQ_GS_FAST_BN   <> 8000H
        Then (3)       LV_GS_ENA_INC = 1
        Else(3)        LV_GS_ENA_INC = 0
        Endif(3)
    Else(2) IF(5) TQI_GS_SLOW_REQ_CAN ≠ FFH   and
                  TQI_GS_FAST_REQ_CAN ≠ FFH
        Then(5)        LV_GS_ENA_INC = 1
        Else(5)        LV_GS_ENA_INC = 0
        Endif(5)
    Endif(2)
Else(1)     LV_GS_ENA_INC = 0
Endif(1)
```

STATE_TQ_ETCU_PLAUS:

```
if   LV_GS_ENA = 0                                          BIT [0] is set
IF   LV_TQ_GS_DEC_REQ = 1              and
     T_TQR_GS_INTV > = C_T_MAX_TQR_GS_INTV                  BIT [2] is set
IF(1)        LV_TQ_GS_DEC_REQ = 1
Then(1) IF(2)      LV_VAR_BN = 1
     Then(2) IF(3) TQ_GS_SLOW_BN   == 8000H or              BIT [3]  is set
                   TQ_GS_FAST_BN   == 8000H
     Else(2) IF(4) TQI_GS_SLOW_REQ_CAN == FFH or            BIT [3] is set
                   TQI_GS_FAST_REQ_CAN ==FFH
     Endif(2)
Else(1)                                                     BIT [01] set
Endif
IF           LV_TQ_GS_INC_REQ = =1
Then IF(2) LV_VAR_BN = 1
     Then(2) IF(3) TQ_GS_SLOW_BN   == 8000H                 BIT [3] is set
                   TQ_GS_FAST_BN   == 8000H
     Else(2) IF(4) TQI_GS_SLOW_REQ_CAN == FFH or            BIT [3] is set
                   TQI_GS_FAST_REQ_CAN ==FFH
     Endif(2)
Else                                                        BIT [4] is set
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MAX_TQR_GS_INTV	1	0...FFFFH	0...655.35	0.01	[s]
Maximum duration of torque reduction due gear shift intervention					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_GS_ENA_INC_DEC	1	0H 1H 2H 3H	NO DEC INC DEC_INC	1	[-]
Enable switch GS intervention					

65.8.2 Torque request CAN selection

FUNCTION DESCRIPTION:

If one of the GS interventions enabled (LV_GS_ENA_DEC/INC = 1), the torque request is received from CAN and copied into TQI_GS_FAST_DEC/INC. At CAN 11H the requested torque is adjusted by ignition retardation. Only if ignition retardation is not sufficient for the requested torque reduction, TQI_GS_SLOW_DEC is set to TQI_GS_REQ. So the throttle is used too. At BN the slow and fast path are different messages.

After LV_GS_ACT_DEC/INC changes back to 0, both TQI_GS_FAST_DEC/INC and TQI_GS_SLOW_DEC/INC become their maximum/minimum value again.

Application conditions:

Initialisation: **at reset** TQI_GS_FAST_DEC_REQ = 1023,97Nm
TQI_GS_SLOW_DEC_REQ = 1023,97Nm
TQI_GS_FAST_INC_REQ = 0 Nm
TQI_GS_SLOW_INC_REQ = 0 Nm

Recurrence: 10 ms
Activation: (LV_AT = 1 **or** LV_VAR_TCT = 1) **and** at LV_IGK = 1
Deactivation: -

Formula section:

```
if      LV_VAR_BN = 1
then    TQI_GS_SLOW_DEC_REQ_1 = TQ_GS_SLOW_DEC_BN - TQ_LOSS
        TQI_GS_FAST_DEC_REQ_1 = TQ_GS_FAST_DEC_BN - TQ_LOSS
        TQI_GS_SLOW_INC_REQ_1 = TQ_GS_SLOW_INC_BN - TQ_LOSS
        TQI_GS_FAST_INC_REQ_1 = TQ_GS_FAST_INC_BN - TQ_LOSS
Else    TQI_GS_SLOW_INC_REQ_1 = TQI_GS_SLOW_REQ
        TQI_GS_FAST_INC_REQ_1 = TQI_GS_FAST_REQ
        TQI_GS_SLOW_DEC_REQ_1 = TQI_GS_SLOW_REQ
        TQI_GS_FAST_DEC_REQ_1 =TQI_GS_FAST_REQ
endif

IF      LV_GS_ENA_DEC = 1
Then    TQI_GS_SLOW_DEC_REQ = TQI_GS_SLOW_DEC_REQ_1
```

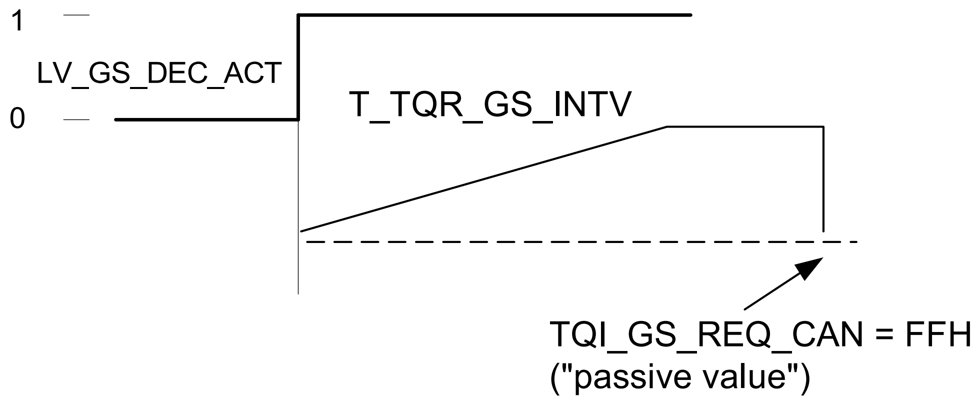
```
        TQI_GS_FAST_DEC_REQ = TQI_GS_FAST_DEC_REQ_1
Else    TQI_GS_SLOW_DEC_REQ = 7FFFH
        TQI_GS_FAST_DEC_REQ = 7FFFH
endif
IF      LV_GS_ENA_INC = 1
Then    TQI_GS_SLOW_INC_REQ = TQI_GS_SLOW_INC_REQ_1
        TQI_GS_FAST_INC_REQ = TQI_GS_FAST_INC_REQ_1
Else    TQI_GS_SLOW_INC_REQ = 0 Nm
        TQI_GS_FAST_INC_REQ = 0 Nm
endif
```

65.8.3 Determination of gear shift torque intervention duration (CAN11H only)

FUNCTION DESCRIPTION:

The duration of decreased gear shift torque intervention T_TQR_GS_INTV is measured beginning at transition of LV_GS_DEC_ACT from 0 to 1. T_TQR_GS_INTV is needed for limitation of gear shift intervention duration during enable conditions. T_TQR_GS_INTV is reseted to 0 if TQI_GS_SLOW_REQ_CAN or TQI_GS_FAST_REQ_CAN = FFH .

Signal flow diagram:



Application conditions:

Initialisation:	T_TQR_GS_INTV = 0 at reset **or** at transition LV_IGK 0 --> 1
Recurrence:	10 ms
Activation:	at LV_IGK = 1 and LV_VAR_BN = 0
Deactivation:	-

65.8.4 GS intervention

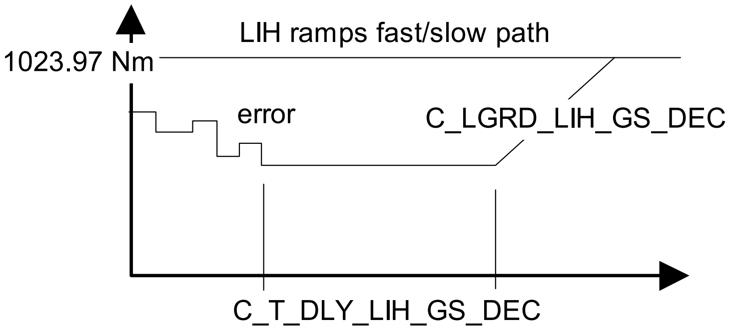
65.8.4.1 GS decreased intervention and emergency operation

FUNCTION DESCRIPTION:

If during an active GS intervention an GS error or an EMS error occurs, the current torque reduction is continued for the time C_T_DLY_LIH_GS_DEC and then subsequently increased to TQI_GS_FAST_DEC = TQI_GS_SLOW_DEC = 1023,97 Nm using C_LGRD_LIH_GS_DEC.

At transition from Limp home condition active to passive after torque inrease to TQI_GS_FAST_DEC = TQI_GS_SLOW_DEC = 1023,97 Nm at limp home operation the counter T_GS_LIH_CTR is stopped and set to 0.

Signal flow diagram:



Application conditions:

Initialisation: at reset **or** at transition LV_IGK 0 --> 1
LV_GS_DEC_LIH = 0
LV_GS_DEC_ACT = 0
TQI_GS_FAST_DEC = TQI_GS_SLOW_DEC = 1023.97 Nm
T_GS_DEC_LIH_CTR = 0

Recurrence: 10 ms

Activation: (LV_AT = 1 **or** LV_VAR_TCT = 1) **and** LV_IGK = 1

Deactivation:

Formula section:

GS_DEC intervention and detection of GS_DEC limp home operation :

```
If (1a)          LV_ETCU_DISABLE_CAN = 1
then (1a)
    if (1b)      TQI_GS_FAST_DEC  < 1023.97 Nm        or
                 TQI_GS_SLOW_DEC < 1023.97 Nm
    Then (1b)    LV_GS_DEC_LIH = 1
                 T_GS_DEC_LIH_CTR $_{N}$ = T_GS_DEC_LIH_CTR $_{N-1}$ + 10 ms
                              ( till T_DEC_GS_LIH_CTR = C_T_DLY_LIH_GS_DEC )
          if (2) T_GS_DEC_LIH_CTR       < = C_T_DLY_LIH_GS_DEC
                 (time delay)
          then (2)TQI_GS_FAST_DEC $_{N}$      = TQI_GS_FAST_DEC $_{N-1}$    (last valid value)
                 TQI_GS_SLOW_DEC $_{N}$      = TQI_GS_SLOW_DEC $_{N-1}$    (last valid value)
          else (2)TQI_GS_FAST_DEC $_{N}$      = TQI_GS_FAST_DEC $_{N-1}$ +
                                                C_LGRD_LIH_GS_DEC
                                    (till TQI_GS_FAST_DEC = 1023.97 Nm )
                 TQI_GS_SLOW_DEC $_{N}$      = TQI_GS_FAST_DEC $_{N}$
          Endif (2)
    Else (1b)    LV_GS_DEC_LIH     = 0
                 T_GS_LIH_CTR      = 0
    Endif (1b)
else (1a)        LV_GS_DEC_LIH     = 0
                 T_GS_LIH_CTR      = 0
                 TQI_GS_FAST_DEC        = TQI_GS_FAST_DEC_REQ
                 TQI_GS_SLOW_DEC        = TQI_GS_SLOW_DEC_REQ
Endif(1a)
```

GS intervention active (LV_GS_DEC_ACT) :

general specification

```
if      TQI_GS_FAST_DEC  < 1023.97 Nm
then    LV_GS_DEC_ACT    = 1
else    LV_GS_DEC_ACT    = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LGRD_LIH_GS_DEC	1	0...7FFFH	0...1023.96875	0.03125	[Nm/10ms]
Limiting gradient for GS emergency ramp function (default 0.6 Nm/10ms)					
C_T_DLY_LIH_GS_DEC	1	0...FFH	0...2.55	0.01	[s]
Delay time for activation of GS emergency ramp function (default 0s)					

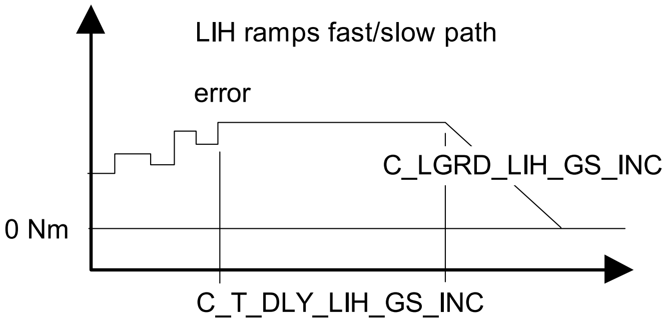
65.8.4.2 GS increased intervention and emergency operation

FUNCTION DESCRIPTION:

If during an active GS intervention an GS error or an EMS error occurs, the current torque offset is continued for the time C_T_DLY_LIH_GS_INC and then subsequently decreased to TQI_GS_FAST_INC = TQI_GS_SLOW_INC = 0 Nm using C_LGRD_LIH_GS_INC.

At transition from Limp home condition active to passive after torque decrease to TQI_GS_FAST_INC = TQI_GS_SLOW_INC = 0 Nm at limp home operation the counter T_GS_LIH_CTR is stopped and set to 0.

Signal flow diagram:



Application conditions:

Initialisation: at reset **or** at transition LV_IGK 0 --> 1

LV_GS_LIH_INC = 0

LV_GS_INC_ACT = 0

TQI_GS_FAST_INC = TQI_GS_SLOW_INC = 0 Nm

T_GS_INC_LIH_CTR = 0

Recurrence: 10 ms

Activation: (LV_AT = 1 **or** LV_VAR_TCT = 1) **and** LV_IGK = 1

Formula section:

GS_INC intervention and detection of GS_INC limp home operation :

If (1a) LV_ETCU_DISABLE_CAN = 1

```
then (1a)
    if(1b)  TQI_GS_FAST_INC > 0       or
            TQI_GS_SLOW_INC > 0
    Then(1b)     LV_GS_INC_LIH = 1
                 T_GS_INC_LIH_CTR$_{N}$ = T_GS_INC_LIH_CTR$_{N-1}$ + 10 ms
                                   ( till T_GS_INC_LIH_CTR = C_T_DLY_LIH_GS_INC )
            if (2)    T_GS_INC_LIH_CTR         < = C_T_DLY_LIH_GS_INC       (time delay)
            then (2)TQI_GS_FAST_INC$_{N}$       = TQI_GS_FAST_INC$_{N-1}$      (last valid value)
            and       TQI_GS_SLOW_INC$_{N}$       = TQI_GS_SLOW_INC$_{N-1}$      (last valid value)
            else (2)TQI_GS_FAST_INC$_{N}$       = TQI_GS_FAST_INC$_{N-1}$ -
                                                       C_LGRD_LIH_GS_INC
                                                  (till TQI_GS_FAST_INC = 0Nm )
                      TQI_GS_SLOW_INC$_{N}$       = TQI_GS_FAST_INC$_{N}$
            endif (2)
    Else(1b)     LV_GS_INC_LIH = 0
                 T_GS_LIH_CTR       = 0
    Endif(1b)
else (1a)        LV_GS_INC_LIH = 0
                 T_GS_LIH_CTR       = 0
                 TQI_GS_FAST_INC  = TQI_GS_FAST _INC_REQ
                 TQI_GS_SLOW_INC= TQI_GS_SLOW _INC_REQ
endif (1a)
```

GS intervention active (LV_GS_INC_ACT) :

```
if        TQI_GS_FAST_INC  > 0 Nm
then      LV_GS_INC_ACT     = 1
else      LV_GS_INC_ACT     = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LGRD_LIH_GS_INC	1	0...7FFFH	0...1023.96875	0.03125	[Nm/10ms]
Limiting gradient for GS emergency ramp function (default 0.6 Nm/10ms)					
C_T_DLY_LIH_GS_INC	1	0...FFH	0...2.55	0.01	[s]
delay time for activation of GS emergency ramp function (default 0s)					

65.8.4.3 Inhibiting fuel cut off during TCT-intervention

FUNCTION DESCRIPTION:

In order to avoid an increasing emission for very low emission vehicles the fuel cut off is inhibited. As result during TCT-intervention the SCC is suppressed in order to meet ULEV2 limits. In that case the reduction of the moment should only be done via spark retard.

Application conditions:

Initialisation: at reset **or** at transition LV_IGK 0 --> 1

LV_INH_FCUT_GS = 0

Recurrence: 10 ms

Activation: LV_IGK = 1 **and** LV_VAR_TCT = 1

Deactivation: -

FCUT inhibitation during TCT intervention active:

```
If   TQ_GS_FAST_BN >
     TQ_LOSS - C_TQ_TCT_FAST INH_FCUT          AND
     TQ_GS_SLOW_BN >
     TQ_LOSS + C_TQ_TCT_SLOW_INH_FCUT
     then    LV_INH_FCUT_GS   = 1
     else    LV_INH_FCUT_GS   = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_TQ_TCT_FAST_INH_FCUT	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Threshold for inhibiting FCUT during TCT intervention					
C_TQ_TCT_SLOW_INH_FCUT	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Threshold for inhibiting FCUT during TCT intervention					

65.9 Converter Torque

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_CONV	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Converter torque					
LV_DLY_N_SP_IS	O/V	0...1H	0...1	1	[-]
Logical variable idle speed setpoint change					
TQ_CONV_STN	V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Calculated converter torque					
TQ_CONV_CAN_1	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Converter torque of CAN					
FAC_TQ_DIF_IS_AD_CONV_LGRD	V	0...FFFFH	0...1	0.0153e-3	[-]
Convergence factor for ramping of TQ_DIF_IS_AD_CONV in case of LV_DRI=1					
LV_TQ_DIF_IS_AD_CONV_ACT	V	0...1H	0...1	1	[-]
Flag indicating IS adaptation value TQ_DIF_IS_AD_CONV is included in TQ_CONV					

Input data:

N_SP_IS	TOIL	LV_DRI	LV_AT
VS	GR_AT		TQ_CONV_CAN
LV_ERR_CAN_BOFF	LV_ERR_TOUT_ETCU_1	TQ_DIF_IS_AD_CONV	
LV_ERR_BN_TQ_ETCU	LV_VAR_TCT	STATE_TCT_INTV	TQ_TCT_CAN

FUNCTION DESCRIPTION:

General information:

TQ_CONV describes the hydraulic torque losses inside the converter of an automatic gearbox due to speed difference between converter turbine and pump. There are two kinds of transmission types with and w/o a relevant torque loss at P/N. So the calculation of TQ_CONV_STN at LV_DRI = 0 (= P/N) can be inhibited by LC_TQ_CONV_DRI_INH = 1. Additional to the calculated converter torque TQ_CONV_STN a converter torque transmitted via CAN TQ_CONV_CAN can be used by setting LC_TQ_CONV_CAN_INH to zero. In case of TQ_CONV_CAN is enabled (LC_TQ_CONV_CAN_INH = 0) and a timeout CAN-message ETCU1- or CAN bus off error is recogniced TQ_CONV = TQ_CONV_STN as calculated in ECU.

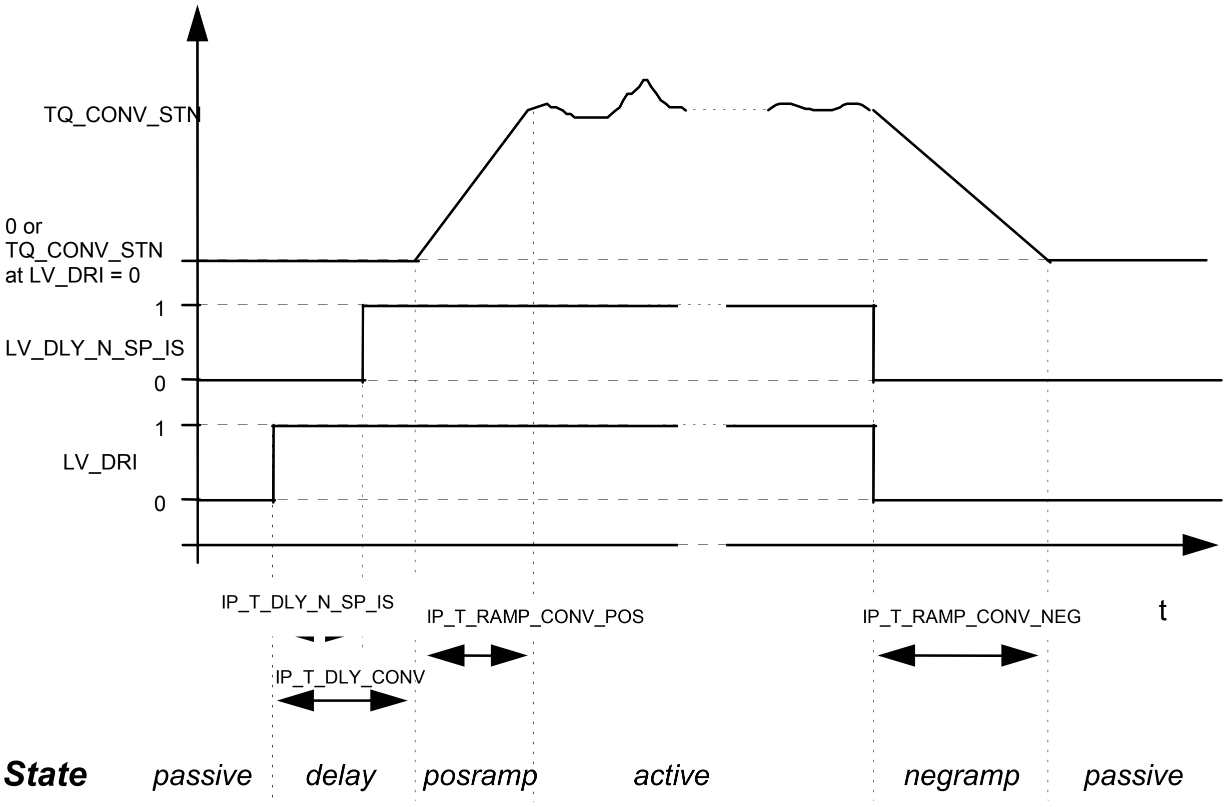
In order to assist the idle speed controller, TQ_CONV_STN is calculated for a reference condition with an engine speed set to N_SP_IS. As a result, TQ_CONV_STN is bigger than the real converter torque for N < N_SP_IS and smaller for N > N_SP_IS.

The converter torque adaptation TQ_DIF_IS_AD_CONV is calculated in the idle speed adaptation module.

After a delay of IP_T_DLY_CONV, TQ_CONV is reached following a ramp with the length IP_T_RAMP_CONV_POS (The calibration parameter IP_T_RAMP_CONV_POS is used to calculate the ramp gradient, which is kept constant for this ramp, even if TQ_CONV_STN changes during this time. Therefore, the observed ramp length may differ from IP_T_RAMP_CONV_POS). After disengaging the gear it is suppressed without any time delay, following a ramp with the length IP_T_RAMP_CONV_NEG. In case of LC_TQ_CONV_CAN_INH = 0 no ramps are calculated!

The change of the idle speed setpoint N_SP_IS during gearshift is synchronised with the converter torque request by application of a time delay T_DLY_N_SP_IS after engaging the gear.

The present model is valid for open converter clutch only. However, the latter normally can be closed or slips only in an operating range N >> N_SP_IS, in which TQ_CONV_STN -> 0 (or TQ_CONV_STN in case of LC_TQ_CONV_DRI_INH = 0).



Application conditions:

Activation: at every engine state (LV_AT = 1 or LV_VAR_TCT = 1)

Deactivation: -

Initialisation: at reset

TQ_CONV = 0 Nm

LV_DLY_N_SP_IS = 0

TQ_CONV_STN = 0 Nm

TQ_CONV_CAN_1 = 0 Nm

FAC_TQ_DIF_IS_AD_CONV_LGRD = 0

LV_TQ_DIF_IS_AD_CONV_ACT = 0

Update rate: 10 ms

Formula section:

selection of converter torque interface :

```
if      LV_VAR_TCT = 0
then    if      LV_ERR_TOUT_ETCU_1 = 0
                and  LV_ERR_BN_TQ_ETCU = 0
                and  LV_ERR_CAN_BOFF = 0
                and  LC_TQ_CONV_CAN_INH = 0
        then        if      LV_DRI = 1
                            and  TQ_CONV_CAN > C_TQ_CONV_CAN_THD
                    then            if      (1 - $FAC\_TQ\_DIF\_IS\_AD\_CONV\_LGRD_{n-1}$) >=
                                            C_FAC_TQ_DIF_IS_AD_CONV_LGRD
                                    then    $FAC\_TQ\_DIF\_IS\_AD\_CONV\_LGRD_{n}$ =
                                            $FAC\_TQ\_DIF\_IS\_AD\_CONV\_LGRD_{n-1}$
                                            + C_FAC_TQ_DIF_IS_AD_CONV_LGRD
                                            TQ_CONV =  TQ_CONV_CAN_1 +
                                                       (TQ_DIF_IS_AD_CONV
                                                         *
             $FAC\_TQ\_DIF\_IS\_AD\_CONV\_LGRD_{n}$)
                                    else    TQ_CONV =  TQ_CONV_CAN_1 +
                                                       TQ_DIF_IS_AD_CONV
                                            $FAC\_TQ\_DIF\_IS\_AD\_CONV\_LGRD_{n}$ = 1
                                    endif
                    else    TQ_CONV  =  TQ_CONV_CAN_1
                            $FAC\_TQ\_DIF\_IS\_AD\_CONV\_LGRD_{n}$ = 0
                    endif
        else  TQ_CONV  =  TQ_CONV_STN
        endif
else    if      STATE_TCT_INTV = 06h (N_REGULATION)
        then    $TQ\_CONV_{n}$ = $TQ\_CONV_{n-1}$
                $LV\_TQ\_DIF\_IS\_AD\_CONV\_ACT_{n}$ = $LV\_TQ\_DIF\_IS\_AD\_CONV\_ACT_{n-1}$
        else    if      LV_DRI = 1 and TQ_TCT_CAN <> 0
                then    $LV\_TQ\_DIF\_IS\_AD\_CONV\_ACT_{n}$ = 1
                endif
                if      LV_DRI = 0 and TQ_TCT_CAN <= C_TQ_TCT_CAN_THD_CONV
                then    $LV\_TQ\_DIF\_IS\_AD\_CONV\_ACT_{n}$ = 0
                endif
```

```
        if      $LV\_TQ\_DIF\_IS\_AD\_CONV\_ACT_n = 1$
        then    $TQ\_CONV_n = TQ\_TCT\_CAN + TQ\_DIF\_IS\_AD\_CONV$
        else    $TQ\_CONV_n = TQ\_TCT\_CAN$
        endif
endif
endif
```

TQ_CONV_CAN_1 = IP_TQ_CONV_CAN

```
if      LV_DRI = 0    and    LC_TQ_CONV_DRI_INH = 0
  then  $TQ\_CONV\_STN = IP\_FAC\_CONV * IP\_FAC\_TQ\_CONV * N\_SP\_IS^2 + TQ\_DIF\_IS\_AD\_CONV$
  else  $TQ\_CONV\_STN = IP\_FAC\_CONV\_DRI * IP\_FAC\_TQ\_CONV * N\_SP\_IS^2 + TQ\_DIF\_IS\_AD\_CONV$
endif
```

IP_FAC_TQ_CONV = f (N_RATIO_CONV)

N_RATIO_CONV = IP_N_TUR / N_SP_IS

IP_N_TUR = f (GR_AT,VS)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_CONV_DRI__TOIL	8	0...FFFFH	0...0.00048	7.4464e-9	[Nm/rpm²]
LDP_TOIL__FAC_CONV	8	0...C8H	-40...160	1	[°C]
Converter characteristic at engaged gear					
IP_FAC_CONV__TOIL	8	0...FFFFH	0...0.00048	7.4464e-9	[Nm/rpm²]
LDP_TOIL__FAC_CONV	8	0...C8H	-40...160	1	[°C]
Converter characteristic at P / N (LV_DRI = 0)					
IP_FAC_TQ_CONV__N_RATIO_CONV	8	0...FFFFH	0...1.99996	0.0305e-3	[-]
LDP_N_RATIO__CONV_FAC_TQ_CONV	8	0...FFH	0...0.99609	3.9063e-3	[-]
Torque request factor					
IP_T_DLY_CONV__TOIL	8	0...FFH	0...2550	10	[ms]
LDP_TOIL__T_DLY_CONV	8	0...C8H	-40...160	1	[°C]
time delay for TQ_CONV activation					
IP_T_DLY_N_SP_IS__TOIL	8	0...FFH	0...2550	10	[ms]
LDP_TOIL__T_DLY_N_SP_IS	8	0...C8H	-40...160	1	[°C]
time delay for change of idle speed setpoint					
IP_T_RAMP_CONV_POS__TOIL	8	0...FFH	0...2550	10	[ms]
LDP_TOIL__T_RAMP_CONV_POS	8	0...C8H	-40...160	1	[°C]
ramp after shift engaging					
IP_T_RAMP_CONV_NEG__TOIL	8	0...FFH	0...2550	10	[ms]
LDP_TOIL__T_RAMP_CONV_NEG	8	0...C8H	-40...160	1	[°C]
ramp after shift disengaging					
IP_N_TUR__GR_AT__VS	9*2	0...FFFFH	0...65535	1	[rpm]
LDP_GR_AT__N_TUR	9	0...8H	0...8	1	[-]
LDP_VS__N_TUR	2	0...FFH	0...255	1	[km/h]
turbine speed					
LC_TQ_CONV_DRI_INH	1	0...1H	0...1	1	[-]
Logical constant converter torque calculation at LV_DRI = 0 inhibit					
LC_TQ_CONV_CAN_INH	1	0...1H	0...1	1	[-]
Logical constant converter torque by CAN-interface inhibit					
IP_TQ_CONV_CAN	8*8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32__TQ_CONV	8	0...FFH	0...8160	32	[rpm]
LDP_TQ_CONV_CAN__TQ_CONV	8	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
Corrected Converter torque via N_32					
C_TQ_CONV_CAN_THD	1	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Threshold for TQ_CONV_CAN regarding the consideration of TQ_DIF_IS_AD_CONV in case of LV_DRI=1					
C_FAC_TQ_DIF_IS_AD_CONV_LGRD	1	0...FFFFH	0...1	0.0153e-3	[-]
Convergence factor (delta per sample) for ramping of TQ_DIF_IS_AD_CONV in case of LV_DRI=1					
C_TQ_TCT_CAN_THD_CONV	1	0...7FFFH	0...1023.96875	0.03125	[Nm]
Threshold for TQ_TCT_CAN to detect transmission not in gear					

65.10 Torque Reserve for Transmission

65.10.1 Determination of Torque reserve for AT Transmission

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_ADD_CONV	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for switching converter Torque					
LV_TOUT_CONV	V	0...1H	0...1	1	[-]
Logical variable to start deactivation of torque reserve and condition for switch off TQ_CONV calculation					
T_CONV	V	0...1FEH	0...5.1	0.01	[s]
Timer for duration of transient TQ_LOSS_CONV correction					
TQ_ADD_CONV_2	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for switching converter Torque					

Input data:

LV_DRI	N_32	LV_ST	LV_AT
TOIL	AC_CRU	VS	N_GRD

FUNCTION DESCRIPTION:

General information:

To guarantee idle quality during gear shift a positive torque reserve can be requested.

Application conditions:

Initialisation: at reset

TQ_ADD_CONV = 0 Nm

LV_TOUT_CONV = 0

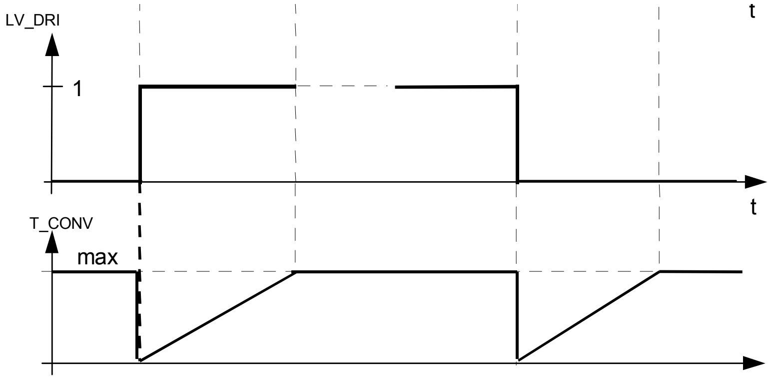
T_CONV = 0 s

TQ_ADD_CONV_2 = 0 Nm

Recurrence: 10 ms

Activation: LV_DRI= 1 **and** LV_ST = 0 and LV_AT = 1

Deactivation: **[** LV_DRI= 0 **and** TQ_ADD_CONV= 0 Nm **and** TQ_ADD_CONV_2 = 0Nm**]**


LV_DRI
t
1
T_CONV
t
max

Formula section:

```
IF T_CONV < 1FEH
Then    LV_TOUT_CONV = 0
Else    LV_TOUT_CONV = 1
endif

if      [ LV_DRI = 1 and LV_TOUT_CONV = 0 ]
then    TQ_ADD_CONV = max [ IP_TQ_ADD_CONV; 0 Nm ]
        TQ_ADD_CONV_2 = MAX ( IP_TQ_ADD_CONV_N_GRD;
                    IP_TQ_ADD_CONV_AC * IP_FAC_TQ_ADD_CONV_TOIL

else    $TQ\_ADD\_CONV_{N}$ = max [ ( $TQ\_ADD\_CONV_{N-1}$ - C_TQ_ADD_CONV_LGRD );
                            0 Nm ]
        $TQ\_ADD\_CONV\_2_{N}$ =
                    max [ ( $TQ\_ADD\_CONV\_2_{N-1}$ - C_TQ_ADD_CONV_LGRD );
                            0 Nm ]

endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQ_ADD_CONV	6*6	0...FFFFH	-1024...1023.96875	0.03125	[Nm]
LDP_N_32_IP_TQ_ADD_ACC	6	0...FFH	0...8160	32	[rpm]
LDPM_TOIL_TQ_ADD_CONV	6	0...C8H	-40...160	1	[°C]
Torque reserve for switch GS					
C_TQ_ADD_CONV_LGRD	1	0...7FFFH	0...1023.97	0.03125	[Nm]
Limitation gradient for TQ_ADD_CONV decrease after deactivation					
IP_TQ_ADD_CONV_N_GRD	6*6	0...FFFFH	-1024...1023.96875	0.03125	[Nm]
LDP_N_GRD_TQ_ADD_CONV	6	0...FFH	-4096...4064	32	[rpm/s]
LDPM_VS_TQ_ADD_CONV	6	0...FFH	0...255	1	[km/h]
Torque reserve for switch GS by N_GRD					
IP_TQ_ADD_CONV_AC	6*6	0...FFFFH	-1024...1023.96875	0.03125	[Nm]
LDP_AC_CRU_TQ_ADD_CONV	6	0...FFFFH	-15.72...15.71952	0.4797e-3	[m/s²]
LDPM_VS_TQ_ADD_CONV	6	0...FFH	0...255	1	[km/h]
Torque reserve for switch GS by AC_CRU					
IP_FAC_TQ_ADD_CONV_TOIL	6	0...FFFFH	0...0.99998	0.0153e-3	[-]
LDPM_TOIL_TQ_ADD_CONV	6	0...C8H	-40...160	1	[°C]
Torque reserve for switch GS					

65.10.2 Determination of Torque reserve for TCT

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_ADD_TCT	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for TCT					
LV_TOUT_TCT	V	0...1H	0...1	1	[-]
Logical variable to start deactivation of torque reserve and condition for switch off TCT calculation					
T_TCT	V	0...1FEH	0...5.1	0.01	[s]
Timer for duration of transient TQ_ADD_TCT correction					

Input data:

	N_32	LV_ST	
LV_VAR_TCT	STATE_ETCU_CLU		

FUNCTION DESCRIPTION:

General information:

To guarantee idle quality during gear shift a positive torque reserve can be requested.

Application conditions:

Initialisation: at reset

TQ_ADD_TCT = 0 Nm

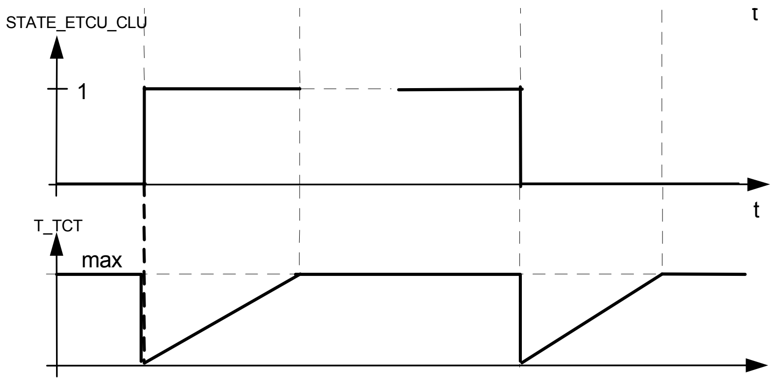
LV_TOUT_TCT = 0

T_TCT = 0 s

Recurrence: 10 ms

Activation: {**[**STATE_ETCU_CLU = 1 **and** LV_ST = 0**]**

and LV_VAR_TCT = 1}



Formula section:

```
IF T_TCT < 1FEH
Then    LV_TOUT_TCT = 0
Else    LV_TOUT_TCT = 1
endif

if      LV_TOUT_TCT = 0
then    TQ_ADD_TCT = max [ IP_TQ_ADD_TCT; 0 Nm ]
else    $TQ\_ADD\_TCT_{N}$ = max [ ( $TQ\_ADD\_TCT_{N-1}$ - C_TQ_ADD_TCT_LGRD );
                                        0 Nm ]
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_TQ_ADD_TCT	6	0...FFFFH	-1024... 1023.96875	0.03125	[Nm]
LDP_N_32__IP_TQ_ADD_ACC	6	0...FFH	0...8160	32	[rpm]
Torque reserve for TCT					
C_TQ_ADD_TCT_LGRD	1	0...7FFFH	0...1023.97	0.03125	[Nm]
Limitation gradient for TQ_ADD_TCT decrease after deactivation					

65.10.3 Determination of Torque reserve for GS Transmission

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_ADD_TRANS	O/V	8000...7FFFH	-1024... 1023.96875	0.03125	[Nm]
Torque reserve for Transmission					

Input data:

LV_AT			TQ_ADD_CONV
TQ_ADD_CONV_2	LV_VAR_TCT	TQ_ADD_TCT	

To serve the torque reserve interface it is switch between TCT and AT transmissions.

Initialisation: at reset

TQ_ADD_TRANS = 0 Nm

Recurrence: 10 ms

Activation: LV_AT = 1 or LV_VAR_TCT = 1

Formula section:

```
IF LV_VAR_TCT = 1
Then    TQ_ADD_TRANS = TQ_ADD_TCT
Else IF LV_AT = 1
    Then    TQ_ADD_TRANS = TQ_ADD_CONV + TQ_ADD_CONV_2
    Else    TQ_ADD_TRANS = 0Nm
endif
```

65.11 Customer adaptation module TRSM: Transmission

65.11.1 Outputs for BMW functions which are defined as TRSM exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
B_gang_rueck	V/O	0...1H	0...1	1	[-]
Logic varaiable for reverse gear					
B_fs	V/O	0...1H	0...1	1	[-]
Logic variable for gear engaged					
B_autget	V/O	0...1H	0...1	1	[-]
Automatic transmission rcognized					

Input data:

LV_DRI	LV_AT	STATE_GEAR_REV_CAN	ECU_STATE

FUNCTION DESCRIPTION:
Adaptation to BMW environment.

Application conditions:

Initialisation :	0
Recurrence :	10 ms
Activation :	every engine state

Formula section:

```
B_fs = LV_DRI
B_autget = LV_AT
If STATE_GEAR_REV_CAN = 0 or ECU_STATE = "PWL"
Then
    B_gang_rueck = 0
Else
    B_gang_rueck = 1
Endif
```

65.11.2 Outputs for BMW functions which are not defined as TRSM exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Md_getriebe	V/O	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
gear box torque					
Mdi_gsl_minus	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
GS torque intervention slow decrement					
Mdi_gsl_plus	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
GS torque intervention slow increment					
Mdi_gss_minus	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
GS torque intervention fast decrement					
Mdi_gss_plus	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
GS torque intervention fast increment					
Mdi_res_gs	V/O	D8F0...2710H	-1000...1000	0.1	[Nm]
GS torque intervention fast increment					
B_kupp_ext	V/O	0...1H	0...1	1	[-]
Clutch switch information from BMW					
Gangi	V/O	0...FFH	0...255	1	[-]
Actual gear					

Input data:

TQI_GS_SLOW_DEC	TQI_GS_SLOW_INC	TQI_GS_FAST_DEC	TQI_GS_FAST_INC
LV_CS_CUS	TQ_ADD_TRANS	GEAR	ECU_STATE
GR_AT	LV_DRI	TQ_CONV	LV_VAR_TCT
LV_AT			

DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit power latch phase:

Mdi_xxx_minus = 1000 Nm

Mdi_xxx_plus, Mdi_res_gs, Md_getriebe = 0 Nm

B_kupp_ext = first measured Value

Recurrence : 10 ms

Activation: every engine state

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

B_kupp_ext = LV_CS_CUS

```
    Md_getriebe         =  TQ_CONV


If ECU_STATE = "PWL"
  then

    Mdi_gsl_minus       =  1000  Nm
    Mdi_gsl_plus        =  0 Nm
    Mdi_gss_minus       =  1000  Nm
    Mdi_gss_plus        =  0 Nm
    Mdi_res_gs          =  0 Nm
elseif
    Mdi_gsl_minus       =  TQI_GS_SLOW_DEC
    Mdi_gsl_plus        =  TQI_GS_SLOW_INC
    Mdi_gss_minus       =  TQI_GS_FAST_DEC
    Mdi_gss_plus        =  TQI_GS_FAST_INC
    Mdi_res_gs          =  TQ_ADD_TRANS
    If (LV_AT = 1 && GR_AT = 0 && LV_DRI=1) or
      (LV_VAR_TCT = 1 && GR_MT = 0 && LV_DRI = 1)
      then

            Gangi = 255 {Rückwärtsgang}
    elseif
            Gangi = GEAR
Endif
```

65.11.3 Outputs for SV aggregates

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CS	V/O	0...1H	0...1	1	[-]
Clutch switch information from Siemens					

Input data:

B_kupp			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Application conditions:

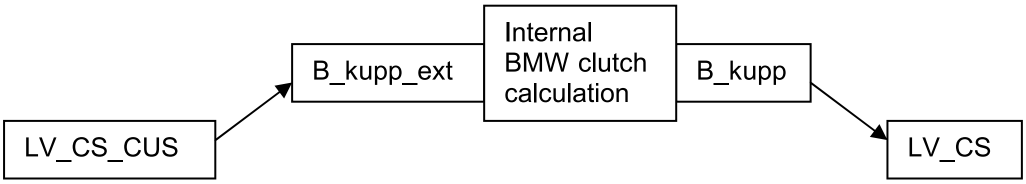
Initialisation : LV_CS = 0

Recurrence : 10 ms

Activation : every engine state

Formula section:

LV_CS = B_kupp



66 Vehicle motion determination

Table of Contents

Index

66.1 Acquisition of vehicle speed

Output data:

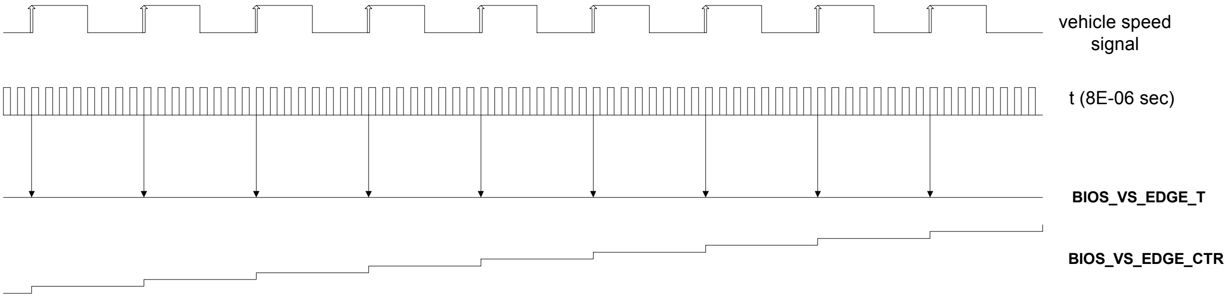
Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
BIOS_VS_EDGE_CTR	O	0 ... FFFFH	0 ... 65565	1	-
Vehicle speed sensor tooth counter (free running counter)					
BIOS_VS_EDGE_T	O	0 ... FFFFH	0 … 524280	8	µs
timestamp of last tooth (free running timer)					

FUNCTION DESCRIPTION:

General information:

The wheel sensor signal (high and low) is recorded by incrementation of the free running edge counter BIOS_VS_EDGE_CTR with one digit at every change from low to high. The value is stored as BIOS_VS_EDGE_CTR in a free running counter too.

Signal flow diagram:



66.2 VHMD - Requirements to infrastructure interface

Export actions:

ACTION_INFR_GetVsPulsStamp(OUT< vs_edge_ctr_av >, <vs_edge_t_av>, <vs_edge_t_av_max)
Reading of information about "vehicle speed"
ACTION_INFR_GetVsRrPulsStamp(OUT< vs_rr_edge_ctr_av >, <vs_rr_edge_t_av>, <vs_rr_edge_t_av_max)
Reading of information about vehicle speed on rough road

Description for actions:

ACTION_INFR_GetVsPulsStamp(OUT <vs_edge_ctr_av>, OUT<vs_edge_t_av>, OUT <vs_edge_t_av_max>)					
This action reads the actual value of vehicle speed tooth counter (free running counter) and the actual timestamp of last tooth (free running timer)					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
vs_edge_ctr_av	OUT	0 ... FFFFh	0 ... 65565	1	-
actual value of vehicle speed tooth counter (free running counter)					
vs_edge_t_av	OUT	0... FFFFFFFFh	0...34359738,37	0,008	ms
actual timestamp of last tooth (free running timer)					
vs_edge_t_av_max	OUT	0... FFFFFFFFh	0...34359738,37	0,008	ms
actual maximal timestamp of last tooth (free running timer)					

ACTION_INFR_GetVsRrPulsStamp(OUT< vs_rr_edge_ctr_av >, <vs_rr_edge_t_av>, <vs_rr_edge_t_av_max)					
This action reads the actual value of vehicle speed tooth counter (free running counter) on "rough road" and the actual timestamp of last tooth (free running timer)					
Name	Type	Hex. Limits	Phys. Limits	Resol.	Unit
vs_rr_edge_ctr_av	OUT	0 ... FFFFh	0 ... 65565	1	-
actual value of vehicle speed tooth counter (free running counter)					
vs_rr_edge_t_av	OUT	0... FFFFFFFFh	0...34359738,37	0,008	ms
actual timestamp of last tooth (free running timer)					
vs_rr_edge_t_av_max	OUT	0... FFFFFFFFh	0...34359738,37	0,008	ms

FUNCTION DESCRIPTION:

General information:

ACTION_INFR_GetVsPulsStamp()
This Action reads the actual value of vehicle speed tooth counter (free running counter) and the actual timestamp of last tooth (free running timer)

ACTION_INFR_GetVsRrPulsStamp()
This Action reads the actual value of vehicle speed tooth counter (free running counter) on rough road and the actual timestamp of last tooth (free running timer)

Requirements for ACTION_INFR_GetVsPulsStamp():

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
vs_edge_ctr_av	-	-	<bit coded>		
vs_edge_t_av	-	-			
vs_edge_t_av_max	-	-			

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

Requirements for ACTION_INFR_GetVsRrPulsStamp():

Data acquisition					
Parameter	**Absolute precision**	**Relative precision**	**Resolution**	**Coincidence with parameter**	**Comment**
vs_rr_edge_ctr_av	-	-	<bit coded>		
vs_rr_edge_t_av	-	-			
vs_rr_edge_t_av_max	-	-			

Diagnosis: no electrical diagnosis done here

Coincidence requirements: no coincidence requirements to other events

66.3 AGGR VHMD adaptation

66.3.1 Outputs for BMW which are not defined as VHMD exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
V	V/O	0...168H	0...360	1	[km/h]
Vehicle speed					

Input data:

VS			

FUNCTION DESCRIPTION:

Adaption to BMW environment.

Application conditions:

Initialisation: 0

Recurrence: 100ms

Activation: after reset (always)

Deactivation: -

Formula section:

Remark: all formulas are valid in a **physical** meaning

V = VS

66.4 Vehicle speed variables

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VS	V/O	0...FFH	0...255	1	[km/h]
Vehicle speed					
VS_H	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
vehicle speed high range					
LV_VS_RUN	V/O	0...1H	0...1	1	[-]
Boolean for running vehicle					
VS_SENS	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
vehicle speed calculated from vehicle speed sensor					
VS_SENS_GRD	V	8000...7FFFH	-256...255.99218	0.0078125	[(km/h)/10 ms]
Vehicle speed gradient calculated from VS_SENS					
VS_EDGE_CTR_AV	V	0...FFFFH	0...65535	1	[-]
actual value of vehicle speed tooth counter (free running counter)					
VS_EDGE_T_AV	V/O	0...FFFFFFFFH	0...34359738.36	0.008	[ms]
actual timestamp of last tooth (free running timer)					
LV_VST	V	0...1H	0...1	1	[-]
boolean for stopped vehicle					
VS_EDGE_T_AV_MAX	V	0...FFFFFFFFH	0...34359738.36	0.008	[ms]
actual maximum timestamp of last tooth (free running timer)					
VS_EDGE_T_AV_DIF	V	0...FFFFFFFFH	0...34359738.36	0.008	[ms]
actual timestamp difference					

Input data:

LV_ERR_VS_PLAUS	VS_FIL		

Import Actions:

ACTION_INFR_GetVsPulsStamp(**OUT**<vs_edge_ctr_av>, **OUT**<vs_edge_t_av>, **OUT**<vs_edge_t_av_max)
Reading of information about "vehicle speed"

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_MIN_RUN	1	0...FFH	0...255	1	[km/h]
vehicle speed threshold for running vehicle detection					
C_T_VST	1	1...14H	10...200	10	[ms]
Time delay for detection of vehicle stop					
C_VS_EDGE_SUM_MIN	1	1...FFH	1...255	1	[-]
minimum number of wheel segments for evaluation of vehicle speed					
C_VS_FAC	1	0...1000000H	0...0.29127	0.0174e-6	[m]
Vehicle distance per segment					

FUNCTION DESCRIPTION:

General information:

Running vehicle is detected if vehicle speed VS ≥ C_VS_MIN_RUN.

Application conditions:

Initialisation: VS = 0 ; VS_H = 0 ; LV_VS_RUN = 0

Recurrence: 10ms

Activation: at every engine operating state

Formula section:

```
If      LV_ERR_VS_PLAUS = 0          // no sensor-error
Then    VS = VS_SENS        VS_H = VS_SENS
Else    VS = VS_FIL         VS_H = VS_FIL
Endif
If      $VS \geq C\_VS\_MIN\_RUN$
Then    LV_VS_RUN = 1
Else    LV_VS_RUN = 0
Endif
```

66.4.1 Internal vehicle speed signal

FUNCTION DESCRIPTION:

General information:

A new edge is detected, if $VS\_EDGE\_CTR\_AV_{n} >= VS\_EDGE\_CTR\_AV_{n-10ms}$. If ($VS\_EDGE\_CTR\_AV_{n}$ - $VS\_EDGE\_CTR\_AV_{n-x}$) > C_VS_EDGE_SUM_MIN a new vehicle speed calculation is started. If during the time C_T_VST no changed edge is detected, the logical variable for stopped vehicle LV_VST is set to 1 and VS_SENS as well as VS_SENS_GRD are set to 0.

With the constant C_VS_FAC the adaption of the rolling circumference and the number of teeth per rotation is done.

Application conditions:

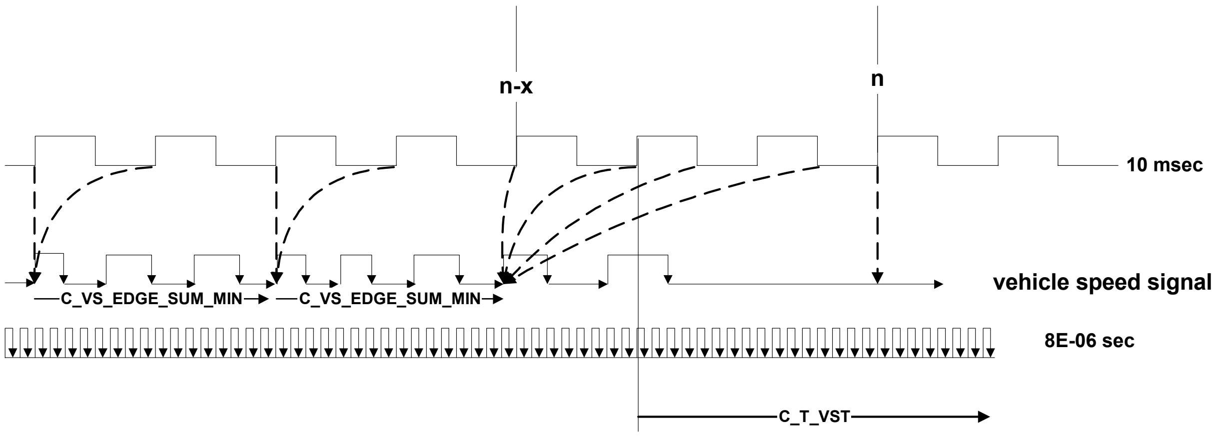
Initialization : VS_SENS = 0
VS_SENS_GRD = 0
LV_VST = 1

Activation : at every engine operating state

Deactivation : -

Update rate : 10ms

Signal flow diagram:



Formula section:

ACTION_INFR_GetVsPulsStamp(VS_EDGE_CTR_AV, VS_EDGE_T_AV, VS_EDGE_T_AV_MAX)

$VS\_EDGE\_T\_AV\_DIF = VS\_EDGE\_T\_AV_{n} - VS\_EDGE\_T\_AV_{n-x}$

```
IF    VS_EDGE_T_AV_DIF < 0   //   (Overflow)
THEN      VS_EDGE_T_AV_DIF = VS_EDGE_T_AV_DIF + VS_EDGE_T_AV_MAX
ELSE      VS_EDGE_T_AV_DIF = (VS_EDGE_T_AV_n - VS_EDGE_T_AV_n-x)
ENDIF
```

C_VS_FAC = (dynamic wheel circumference [m]) / (number of teeth [-])

$VS\_SENS = C\_VS\_FAC * (VS\_EDGE\_CTR\_AV_{n} - VS\_EDGE\_CTR\_AV_{n-x}) / VS\_EDGE\_T\_AV\_DIF$

$VS\_SENS\_GRD = (VS\_SENS_{n} - VS\_SENS_{n-x}) / VS\_EDGE\_T\_AV\_DIF$

66.5 Mileage counter

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
DIST	S/O/V	00000000H... FFFFFFFFH	0... 429496729500	100	m
Distance accumulation [100m]					
DIST_KWP	S/O	0...FFFFH	0...524280	8	Km
Distance accumulation [8km]					

Input data:

CTR_KM_CAN	VS_EDGE_CTR_AV	C_VS_FAC	LV_ES

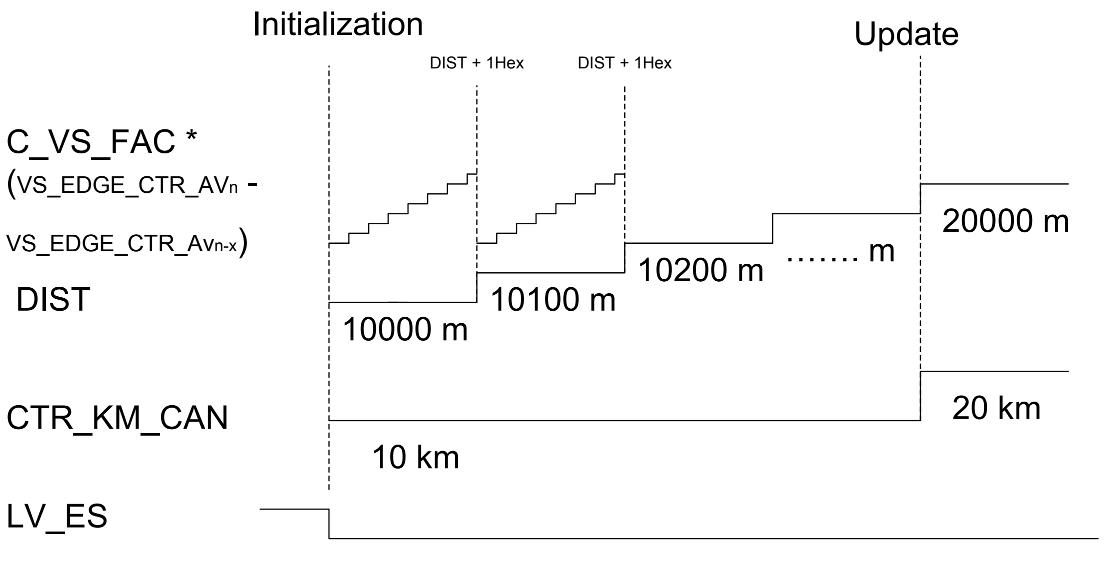
FUNCTION DESCRIPTION:

General information:

This function is to accumulate the covered distance since the first vehicle start in resolution of 100m for EOBD propose. The information for updating of the calculation (CTR_KM_CAN, VS_EDGE_CTR_AV) is sent by CAN.

Signal flow diagram:

DIST calculation



Application conditions:

Initialisation: restored out of NVMY, first value = 0

Recurrence: 200ms

Activation: LV_ES = 0

Description:

After activation DIST is restored out of NVMY and the calculation starts with (C_VS_FAC * $VS\_EDGE\_CTR\_AV_n$ - $VS\_EDGE\_CTR\_AV_{n-x}$) until DIST is incremented with 1Hex (100m).
This calculations is done until the next update of CTR_KM_CAN, then DIST is set to CTR_KM_CAN and the calculation starts from beginning.
If there is no update of CTR_KM_CAN (e.g. Timeout error), then the calculation remains active for this Driving cycle.

DIST_KWP is calculated from DIST but different resolution for BMW diagnosis tool.

Formula section:

DIST: See signal flow diagramm

DIST_KWP = DIST

66.6 Vehicle speed signal diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_VS_PLAUS	O/V	0...1H	0...1	1	[-]
VS plaus sensor error bit					
LV_CDN_DIAG_VS_PLAUS	O/V	0...1H	0...1	1	[-]
VS plaus Diagnosis active (=1)					
ERR_SYM_VS_PLAUS	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected Symptom VS plaus					
LV_END_DIAG_VS_PLAUS	O/V	0...1H	0...1	1	[-]
End of VS plaus Diagnosis					
T_VS_DIAG	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter for error detection					
T_VS_END_DIAG	V	0...FFFFH	0...6553.5	0.1	[s]
Time counter for end of diagnosis					
LV_ERR_VS	O/V	0...1H	0...1	1	[-]
VS sensor error bit					
LV_CDN_DIAG_VS	O/V	0...1H	0...1	1	[-]
VS Diagnosis active (=1)					
ERR_SYM_VS	O/V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected Symptom VS					
LV_END_DIAG_VS	O/V	0...1H	0...1	1	[-]
End of VS Diagnosis					

Input data:

VS	N	MAF	LV_IGK
LV_AT	LV_ERR_MAF	LV_ERR_LOAD_TPS_PLAUS	TQ_AV
LV_VAR_BN	LV_ERR_CAN_BOFF		LV_ERR_MAP_TPS_PLAUS
LV_ERR_BN_VS_TCS	LV_ERR_TOUT_ASR_1		LV_TCHA_DIAG_REQ

66.6.1 Plausibility check of electrical VS signal (PIN)

General information:

An vehicle speed signal error is detected if at calibratable engine speed-, mass air flow- and time thresholds the vehicle speed signal VS = 0. Depending if the vehicle is MT / AT different calibration data a applied (If LV_AT = 1 then _AT are used, else _MT are used).

Description:

If vehicle speed VS = 0 **and** engine speed N >= C_N_VS_MIN_DIAG_AT / _MT **and** (mass air flow MAF >= C_MAF_VS_MIN_DIAG_AT / _MT [for MSVx] **or** TQ_AV >= C_TQ_AV_VS_MIN_DIAG_AT / _MT [for MSDx]) the time counter T_VS_DIAG is started. The vehicle speed signal error bit LV_ERR_VS_PLAUS is set to 1 **if** T_VS_DIAG >= C_T_VS_MIN_DIAG.

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                              (= SYM_0)
 │ │ └───> -                              (= SYM_1)
 │ └─────> -                              (= SYM_2)
 └───────> Vehicle speed not plausible    (= SYM_3)
```

Application conditions:

Initialization: The configuration is **STD_INI**

*(=> all 0 at LV_IGK 0->1 **or** reset **or** clearing error memory)*

Recurrence: 100 ms

Activation:

```
If    LV_IGK = 1                      and
      LV_ERR_MAF = 0                  and
      LV_ERR_LOAD_TPS_PLAUS = 0       and
      LV_ERR_MAP_TPS_PLAUS = 0        and
      LV_TCHA_DIAG_REQ = 0            and
      LV_ERR_VS_PLAUS = 0
Then  LV_CDN_DIAG_VS_PLAUS = 1
Else  LV_CDN_DIAG_VS_PLAUS = 0
Endif
```

Formula section:

Timer start:

```
If(1)     VS       =    0                                   and
          N        >=   C_N_VS_MIN_DIAG_AT / _MT            and
          (MAF     >=   C_MAF_VS_MIN_DIAG_AT / _MT          or   [for MSVx]
          TQ_AV    >=   C_TQ_AV_VS_MIN_DIAG_AT / _MT)            [for MSDx]
Then(1)   T_VS_DIAG starts to count
          ERR_SYM_VS_PLAUS = SYM_3
Else(1)   T_VS_DIAG is set to 0
          ERR_SYM_VS_PLAUS = NO_SYM
Endif(1)
```

Error recognition:

```
If(1)      T_VS_DIAG   >=      C_T_VS_MIN_DIAG
Then(1)    LV_ERR_VS_PLAUS= 1      (set directly)
Endif(1)


If(1)      VS > 0
Then(1)    LV_ERR_VS_PLAUS= 0      (set directly)
Endif(1)
```

End of Diagnosis:

```
If (1)     N           >=      C_N_VS_MIN_DIAG_AT / _MT              and
           (MAF        >=      C_MAF_VS_MIN_DIAG_AT / _MT            or    [for MSVx]
           TQ_AV       >=      C_TQ_AV_VS_MIN_DIAG_AT / _MT)               [for MSDx]
Then(1)    T_VS_END_DIAG             starts to count
           If(2)       T_VS_END_DIAG     >=      C_T_VS_MIN_DIAG
           Then(2)     LV_END_DIAG_VS_PLAUS = 1
           Endif(2)
Endif(1)
```

66.6.2 Emission relevant VS double error (No PIN signal / No CAN signal)

General information:

LV_ERR_VS is set if neither a VS signal is available from ECU-PIN (LV_ERR_VS_PLAUS) nor a signal is received from CAN (11H / 12H).

Description:

Error symptoms are defined for this diagnosis function as:

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                              (= SYM_0)
 │ │ └───> -                              (= SYM_1)
 │ └─────> no vehicle speed signal        (= SYM_2)
 └───────> -                              (= SYM_3)
```

Application conditions:

Initialization: The configuration is **STD_INI**

(=> all 0 at LV_IGK 0->1 ***or*** *reset* ***or*** *clearing error memory)*

Recurrence: 100 ms (called just after VS_PLAUS diagnosis)

Activation:

```
If      LV_IGK = 1
        LV_END_DIAG_VS_PLAUS = 1
Then    LV_CDN_DIAG_VS = 1
Else    LV_CDN_DIAG_VS = 0
```

```
                Endif
```

Formula section:

```
If      LV_ERR_VS_PLAUS = 1
Then    If      LV_VAR_BN = 1
        Then    If      LV_ERR_BN_VS_TCS = 1
                Or      LV_ERR_CAN_BOFF = 1
                Then    ERR_SYM_VS = SYM_2          // ABC increments
                Else    ERR_SYM_VS = NO_SYM         // ABC decrements
                Endif
        Else    If      LV_ERR_TOUT_ASR_1 = 1
                Or      LV_ERR_CAN_BOFF = 1
                Then    ERR_SYM_VS = SYM_2          // ABC increments
                Else    ERR_SYM_VS = NO_SYM         // ABC decrements
                Endif
Else    ERR_SYM_VS = NO_SYM                         // ABC decrements
Endif
```

LV_ERR_VS and LV_END_DIAG_VS is calculated from error management.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_VS_MIN_DIAG_MT	1	0...1FE0H	0...8160	1	[rpm]
Engine speed condition - vehicle speed diagnosis, MT vehicle					
C_N_VS_MIN_DIAG_AT	1	0...1FE0H	0...8160	1	[rpm]
Engine speed condition - vehicle speed diagnosis, AT vehicle					
C_MAF_VS_MIN_DIAG_MT	1	0...FFH	0...1389	5.4470588	[mg/stk]
MAF condition - vehicle speed diagnosis, MT vehicle					
C_MAF_VS_MIN_DIAG_AT	1	0...FFH	0...1389	5.4470588	[mg/stk]
MAF condition - vehicle speed diagnosis, AT vehicle					
C_T_VS_MIN_DIAG	1	0...FFFFH	0...6553.5	0.1	[s]
Time condition - vehicle speed diagnosis					
C_TQ_AV_VS_MIN_DIAG_MT	1	0...FFH	0...510	2	[Nm]
TQ_AV condition - vehicle speed diagnosis, MT vehicle					
C_TQ_AV_VS_MIN_DIAG_AT	1	0...FFH	0...510	2	[Nm]
TQ_AV condition - vehicle speed diagnosis, AT vehicle					
C_ABC_INC_VS	1	0...FFH	0...255	1	[-]
ABC Increment value for VS diagnosis					
C_ABC_MAX_VS	1	1...FFH	1...255	1	[-]
ABC Maximum value for VS diagnosis					

Hint:

All conditions for setting LV_ERR_VS are already filtered, thus the debounce of this diagnosis should be very fast e.g. inc/max = 1/1.

66.7 Vehicle speed sensor diagnosis (Applic. Inc.)

66.7.1 Interface for Rate – Based – Monitoring

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_RBM_VS_PLAUS	V/O	0...7H	0...7	1	[-]
Interface of VS_PLAUS monitor with the Rate-Based Monitoring statistics Bit 0: conditions for monitoring are met long enough to detect malfunction (bit 0 = 1) Bit 1: inhibition of the monitor because of system failure(s) (bit 1 = 1) Bit 2: individual RBM conditions of the monitor were encountered within this DC (bit 2 = 1)					

Input data:

LV_DC	LV_END_DIAG_VS_PLAUS		

FUNCTION DESCRIPTION:

General information:

With this module the interface between the VS_PLAUS monitor and the Rate-Based Monitoring statistics is defined with STATE_RBM_VS_PLAUS data.

Within STATE_RBM_VS_PLAUS, three different information are defined:

- Conditions for monitoring are met long enough to detect malfunction (bit 0)

  (no intrusive operation, no short trip)
- Monitor disabled because of system malfunction (bit 1)

  (depending on failure status: pending)
- Monitor individual RBM conditions encountered within this DC (bit 2)

  (not valid for VS_PLAUS diagnosis)

Application conditions:

Initialisation :

at LV_DC 0 → 1 transition :

bit 0, bit 1 and bit 2 of STATE_RBM_VS_PLAUS = 0

on failure memory reset :

bit 1 of STATE_RBM_VS_PLAUS = 0

Recurrence: 1 s

Activation: LV_DC 0 → 1 transition **and** LV_DC = 1

Formula section:

Bit 0:

```
If    bit 0 of STATE_RBM_VS_PLAUS = 0
then
      if      LV_END_DIAG_VS_PLAUS = 1
      then    bit 0 of STATE_RBM_VS_PLAUS = 1
      endif
endif
```

Bit 1:

bit 1 of STATE_RBM_VS_PLAUS = 0

Bit 2:

bit 2 of STATE_RBM_VS_PLAUS = 1

67 Vehicle speed control

Chapter Vehicle speed control	Baseline 4DC3940S		
	Date	Department	Sign
Designed by	2008-07-01		
Released by	2008-07-01		
Continental	Designation Engine Management System MSD80 6 Cyl		
	Document Key E002-190.49.02 SPE 000 48.0		Pages 9447 of 9643

Table of Contents

Index

67.1 MSW interface

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
BIOS_CRU_DATA	V/O	FEH FCH DAH B6H 6EH	NONE DECELERATE OFF SET_ACCELERATE RESUME	0	[-]
Status off MSW module					
BIOS_CRU_TOG	O	0...1H	0...1	1	[-]
Togglebit for MSW Module					
BIOS_CRU_NEW_DATA	O	0...1H	0...1	1	[-]
Validation Bit for Status off MSW					

General information:

The multifunction steering wheel (MSW) with buttons the position of which is transferred digitally through a serial data line serves as interface between the driver and the CRU system. According to extension stage, a distinction is made between low and high version with 8-bit and 16-bit transmission protocol. Only the 8-bit data frame is relevant for the engine control. For plausibility check whether the MSW is still active, a toggle bit is appended to the 7-bit information. This bit changes its level cyclically.

Application conditions:

Recurrence: 10ms

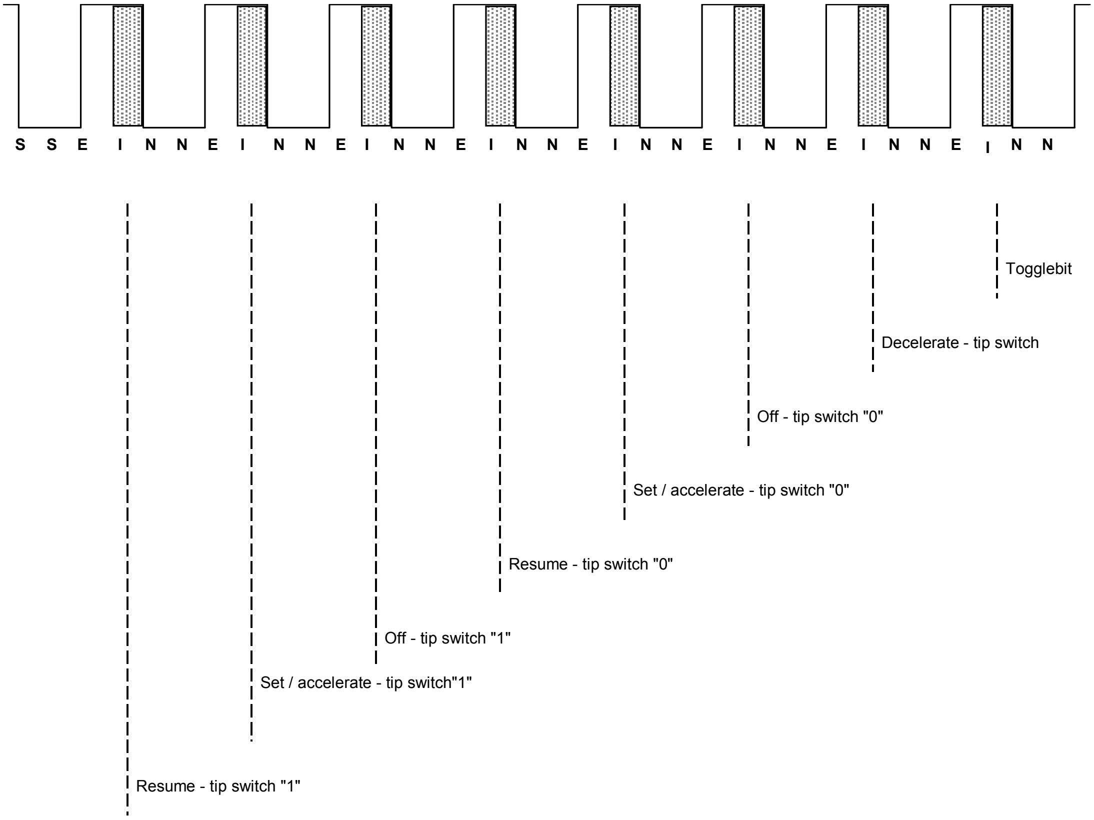
67.1.1 Serial data line from the MSW

General information:

The buttons of the multifunction steering wheel are not interlocked mechanically against one another, which means that a priority must be introduced for the switch position. The S/A and DEC functions are an exception. Here the buttons are formed as rocker in the MSW and thus cannot be actuated simultaneously. The position of the buttons is encoded digitally the MSW control unit and transferred through a serial data line. In the 8-bit data frame, which is transmitted by the MSW, the buttons are encoded as follows:

8-bit data frame of the transmission module

S = Start bit
N = Zero bit
E = One bit
I = Info bit



Resume, setting/acceleration as well as off buttons are encoded redundantly and numbered for differentiating the information (0;1). The encoding of the buttons is high active, i.e. for a pressed button a 1 is transmitted as useful information. Furthermore a toggle bit is transmitted in the bit position 0 of the 8-bit data frame, a special bit information which changes its level cyclical. The toggle bit can thus also be used for error detection, like the switch positions, which are encoded redundantly.

The following distinctions are made in the transmitted functions of the MSW:

1. Rest: SET/ACC and DEC and RES and OFF not pressed.
2. SET/ACC: SET/ACC and no DEC pressed
3. DEC: DEC and no SET/ACC pressed
4. RES: RES pressed.
5. OFF: OFF pressed.

67.2 Distance cruise control diagnosis (LV_ERR_DCC)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_DCC	V/O	0...1H	0...1	1	[-]
DCC error reversible/ irreversible					
LV_CDN_DIAG_DCC	V/O	0...1H	0...1	1	[-]
Status of diagnosis DCC error reversible/ irreversible					
ERR_SYM_DCC	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Detected symptoms DCC error reversible/ irreversible					
LV_END_DIAG_DCC	V/O	0...1H	0...1	1	[-]
End of diagnosis DCC error reversible/ irreversible					
LV_DCC_OFF_ECU	V/O	0...1H	0...1	1	[-]
Irreversible DCC error					
T_DCC_BRAKE_DET	V	0...FFH	0...2.55	0.01	[s]
Brake delay timer					

Input data:

LV_VAR_DCC	LV_IGK	LV_ERR_BLS_PLAUS	LV_ERR_CS
LV_ERR_BN_TQ_DCC	ERR_SYM_BN_TQ_DCC	LV_DCC_OFF_ACK	STATE_DCC_INTV
LV_BRAKE_DET	LV_DI_TQ_REQ_CAN_MPI_GDI	LV_ERR_CRK	LV_TQI_REQ_CAN_INH
LV_ERR_PVS			

FUNCTION DESCRIPTION:

General information:

At implausible DCC messages, e.g. timeout errors, or ECU internal errors a reversible or irreversible error is detected and reflect to the DCC Unit. This should be confirmed by the DCC switch-off flag LV_DCC_OFF_ACK. Only if the switch-off is confirmed by DCC the reversible DCC error is reset. This is necessary for a safety communication between ECU and DCC.

Description:

The following symptoms can be detected:

```
┌──┬──┬──┬──┐
└──┴──┴──┴──┘
 │  │  │  └─> DCC irreverible off        (= SYM_0)
 │  │  └────> DCC reversible off         (= SYM_1)
 │  └───────> -                          (= SYM_2)
 └──────────> -                          (= SYM_3)
```

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 **or** reset **or** at clearing error memory

Recurrence: 10 ms

Activation:

```
If      LV_VAR_DCC = 1          and
        LV_IGK= 1
Then    LV_CDN_DIAG_DCC = 1
```

Deactivation:

```
If      LV_VAR_DCC = 0          or
        LV_IGK = 0              or
        [ LV_ERR_DCC = 1        and
        ERR_SYM_DCC = SYM_0 ]
Then    LV_CDN_DIAG_DCC = 0
Endif
```

Formula section:

Calculation of error condition “brake detection” for setting irrversible error

```
If      LV_BRAKE_DET = 1
Then    T_DCC_BRAKE_DET$_n$ = T_DCC_BRAKE_DET$_{n-1}$ + 10 ms
Else    T_DCC_BRAKE_DET$_n$ =0 ms
Endif
```

Calculation of irrversible error

```
If      [ LV_ERR_BN_TQ_DCC  = 1 and          //ONLY ALIVE AND CHECKSUM ERROR
        ERR_SYM_BN_TQ_DCC  = SYM_1 or SYM_3 ]                    or
        LV_TQI_REQ_CAN_INH = 1                                   or
        LV_ERR_PVS = 1                                           or
         [ (T_DCC_BRAKE_DET > C_T_MAX_BRAKE_DET_DCC) and
        STATE_DCC_INTV == 01H ]
Then    ERR_SYM_DCC = SYM_0          //symptom irreversible DCC error
        // LV_ERR_DCC = 1 set after debounce by error management, set for this DC
```

Calculation of reversible error

```
ElseIf  ( LV_ERR_BN_TQ_DCC    = 1 and  ERR_SYM_BN_TQ_DCC   = SYM_2 ) or
                                                   //only timeout
        LV_ERR_BLS_PLAUS          = 1                    or
        LV_ERR_CRK                = 1                    or
        LV_ERR_CS                 = 1                    or
        LV_DI_TQ_REQ_CAN_MPI_GDI  = 1
Then    ERR_SYM_DCC = SYM_1            //symptom reversible DCC error
        // LV_ERR_DCC = 1 set after debounce by error management
Else    If     LV_ERR_DCC = 0          //not debounced yet
```

```
            Then   ERR_SYM_DCC = NO_SYM
                   debounce counter reset to 0 //reset CTR_ABC_DCC to 0
                                               //no rebouncing!
      Else  If     LV_DCC_OFF_ACK = 1
            Then   ERR_SYM_DCC = NO_SYM
                   LV_ERR_DCC = 0              //reset error and CTR_ABC_DCC to 0
                   debounce counter reset  to 0//no rebouncing!
            Else   ERR_SYM_DCC  = old value (SYM_0 or SYM_1)
            Endif
      Endif
Endif
```

Calculation of LV_DCC_OFF_ECU

```
If          LV_ERR_DCC = 1
Then        LV_DCC_OFF_ECU = 1
ElseIf      LV_DCC_OFF_ACK = 1
    Then    LV_DCC_OFF_ECU = 0
    Else    unchanged
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_DCC	1	0...FFH	0...255	1	[-]
Increment counter					
C_ABC_MAX_DCC	1	1...FFH	1...255	1	[-]
Maximum counter					
C_T_MAX_BRAKE_DET_DCC	1	0...FFH	0...2.55	0.01	[s]
Maximum Brake delay timer					

67.3 Multifunctional steering wheel diagnosis (MSW)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_MSW_2	V/O	0...1H	0...1	1	[-]
Redundant coding of the signal-error MSW_2					
LV_CDN_DIAG_MSW_2	V/O	0...1H	0...1	1	[-]
Diagnosis condition MSW_2					
ERR_SYM_MSW_2	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error					
LV_END_DIAG_MSW_2	V/O	0...1H	0...1	1	[-]
End of Diagnosis MSW_2					
LV_ERR_MSW_TOG	V/O	0...1H	0...1	1	[-]
Toggle bit monitoring-error TOG					
LV_CDN_DIAG_MSW_TOG	V/O	0...1H	0...1	1	[-]
Diagnosis condition TOG					
ERR_SYM_MSW_TOG	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error					
LV_END_DIAG_MSW_TOG	V/O	0...1H	0...1	1	[-]
End of Diagnosis TOG					
LV_ERR_MSW_3	V/O	0...1H	0...1	1	[-]
Toggle switch test-error MSW_3					
LV_CDN_DIAG_MSW_3	V/O	0...1H	0...1	1	[-]
Diagnosis condition MSW_3					
ERR_SYM_MSW_3	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	0	[-]
Detected error					
LV_END_DIAG_MSW_3	V/O	0...1H	0...1	1	[-]
End of Diagnosis MSW_3					
STATE_MSW_DATA	O	1H 3H 25H 49H 91H	NONE TIP_DOWN OFF TIP_UP RESUME	1	[-]
State MSW data					
LV_MSW_MSG_VLD	O	0...1H	0...1	1	[-]
Flag for MSW message is valid					

Input data:

LV_IGK	LV_VAR_MSW	C_T_READY_CRU	BIOS_CRU_TOG
LV_VAR_BN_MSW	BIOS_CRU_DATA	BIOS_LV_CRU_ERR	

FUNCTION DESCRIPTION:

General information:

In this function the diagnosis of the relevant signals from the MSW for the cruise control is made. The following error can be distinguished:

- Redundant coding of the signal
- Toggle switch test
- Toggle bit monitoring

Application conditions:

Initialisation: all 0 at LV_IGK 0->1 or reset

Recurrence: see diagnosis

Activation:

```
If      V_IGK  >  C_VB_MIN_MSW_DIAG
    and Time since  V_IGK > C_VB_MIN_MSW_DIAG  greater than  C_T_READY_CRU
    and LV_VAR_MSW = 1
    and LV_VAR_BN_MSW = 0
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VB_MIN_MSW_DIAG	1	0...FFH	0...25.89843	0.1015625	[V]
Min. battery voltage for MSW diagnosis					

67.3.1 Determination of STATE_MSW_DATA / LV_MSW_MSG_VLD

Description:

The datas are used for function cruise control.

Formula section:

Calculation of LV_MSW_MSG_VLD:

```
If      a single message is valid (no ERR_SYM_... is active, BIOS_LV_CRU_ERR = 0)
Then    LV_MSW_MSG_VLD = 1
Else    LV_MSW_MSG_VLD = 0
Endif
```

Calculation of STATE_MSW_DATA:

STATE_MSW_DATA = (range converted) BIOS_CRU_DATA

67.3.2 Redundant coding check of the signal (LV_ERR_MSW_2)

Application conditions:

Recurrence: 80ms
= every four equal messages plus delay time due to different recurrency
[4 x (15ms + Delay)]

Description:

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                              (= SYM_0)
 │ │ └───> -                              (= SYM_1)
 │ └─────> -                              (= SYM_2)
 └───────> Signal implausible             (= SYM_3)
```

Formula section:

Calculation of diagnosis condition:

```
If      at least one of the bits for WA, S/B or OFF is active in BIOS_CRU_DATA
Then    LV_CDN_DIAG_MSW_2 = 1
        Error detection and plausibility:
        If      Redundant coding is infringed (see "Supplementary functions" chapter, CRU )
        Then    ERR_SYM_MSW_2 = SYM_3
                // LV_ERR_MSW_2 = 1    after debounce by error management
        Else    ERR_SYM_MSW_2 = NO_SYM
                // LV_ERR_MSW_2 = 0    after rebounce by error management
        Endif
Else    LV_CDN_DIAG_MSW_2 = 0
Endif
```

Calculation end of diagnosis:

LV_END_DIAG_MSW_2 is calculated by error management if diagnosis condition is true.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_MSW_2	1	0...FFH	0...255	1	[-]
Debounce counter increment - MSW diagnosis					
C_ABC_MAX_MSW_2	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - MSW diagnosis					

67.3.3 Toggle switch test (LV_ERR_MSW_3)

Application conditions:

Recurrence: 80ms
= every four equal messages plus delay time due to different recurrency
[4 x (15ms + Delay)]

Description:

The toggle switch condition is checked for each new message (BIOS_CRU_DATA) in which S/B or VE are active. If this is infringed, the preceding message remains valid.

Error-symptoms are defined to this diagnosis function as following :

```
┌─┬─┬─┬─┐
└─┴─┴─┴─┘
 │ │ │ └─> -                               (= SYM_0)
 │ │ └───> -                               (= SYM_1)
 │ └─────> -                               (= SYM_2)
 └───────> Rocker switch defect            (= SYM_3)
```

Formula section:

Calculation of diagnosis condition:

```
If      S/B or VE is active in BIOS_CRU_DATA
Then    LV_CDN_DIAG_MSW_3 = 1
        Error detection and plausibility:
        If      S/B and VE are simultaneously active
        Then    ERR_SYM_MSW_3 = SYM_3
                // LV_ERR_MSW_3 = 1    after debounce by error management
        Else    ERR_SYM_MSW_3 = NO_SYM
                // LV_ERR_MSW_3 = 0    after rebounce by error management
        Endif
Else    LV_CDN_DIAG_MSW_3 = 0
Endif
```

Calculation end of diagnosis:

LV_END_DIAG_MSW_3 is calculated by error management if diagnosis condition is true.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_MSW_3	1	0...FFH	0...255	1	[-]
Debounce counter increment - MSW diagnosis					
C_ABC_MAX_MSW_3	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - MSW diagnosis					

67.3.4 Toggle bit monitoring (LV_ERR_MSW_TOG)

Application conditions:

Recurrence: each change of toggle bit (error free: *about 0.5s*)

Description:

The diagnosis condition is set latest if C_T_MAX_MSW_BIT_DIAG is reached, after this the diagnosis condition is on and T_TOG (time between two state changes of the toggle bit BIOS_CRU_TOG) is checked for MIN and MAX time at each new message.

If C_T_MAX_MSW_BIT_DIAG is reached without toggle bit change the error LV_ERR_MSW_TOG is set directly without debounce.

If toggle bit change is faster C_T_MIN_MSW_BIT_DIAG then the error symptom ERR_SYM_MSW_TOG is set and the anti-bounce is inrementing.

Error-symptoms are defined to this diagnosis function as following :

```
└─┴─┴─┴─┘
 │ │ │ └─> -                                   (= SYM_0)
 │ │ └───> -                                   (= SYM_1)
 │ └─────> -                                   (= SYM_2)
 └───────> Toggle-bit period is wrong          (= SYM_3)
```

Formula section:

Error detection and plausibility:

T_TOG is incrementing (10ms steps) and reseted at every toggle bit change

```
If          a full toggle bit cycle is received
Then        LV_CDN_DIAG_MSW_TOG = 1
    If          T_TOG ≤ C_T_MIN_MSW_BIT_DIAG
    Then        ERR_SYM_MSW_TOG = SYM_3
                // LV_ERR_MSW_TOG =  1 after debounce by error management
    Else        ERR_SYM_MSW_TOG = NO_SYM
                // LV_ERR_MSW_TOG =  0 after rebounce by error management
    Endif
ElseIf          T_TOG > C_T_MAX_MSW_BIT_DIAG
    Then        LV_CDN_DIAG_MSW_TOG = 1
                ERR_SYM_MSW_TOG = SYM_3
                LV_ERR_MSW_TOG = 1
                //set directly without debounce, Anti-bounce counter is set to max value
    Else        no change
    Endif
Endif
```

Calculation end of diagnosis:

LV_END_DIAG_MSW_TOG is calculated by error management if diagnosis condition is true.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MIN_MSW_BIT_DIAG	1	0...FFH	0...2550	10	[ms]
Min. diagnostic threshold - Toggle bit monitoring					
C_T_MAX_MSW_BIT_DIAG	1	0...FFH	0...2550	10	[ms]
Max. diagnostic threshold - Toggle bit monitoring					
C_ABC_INC_MSW_TOG	1	0...FFH	0...255	1	[-]
Debounce counter increment - Toggle bit monitoring					
C_ABC_MAX_MSW_TOG	1	1...FFH	1...255	1	[-]
Debounce counter maximum value - Toggle bit monitoring					

67.4 Torque request for distance cruise control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_DCC_DISABLE_CAN	V/O	0...1H	0...1	1	[-]
Logical variable for disabling DCC intervention due to CAN error					
LV_DCC_ENA_INC	V/O	0...1H	0...1	1	[-]
Logical variable for increased torque intervention due to DCC					
STATE_TQ_DCC_PLAUS	V/O	0...FFH	0...255	1	[-]
Bitwise coded State for DCC intervention state					
TQI_DCC_FAST_INC_REQ	V	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for fast torque intervention during DCC					
TQI_DCC_SLOW_INC_REQ	V	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for slow torque intervention during DCC					
TQI_DCC_FAST_INC_REQ_1	V	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for fast torque intervention during DCC					
TQI_DCC_SLOW_INC_REQ_1	V	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for slow torque intervention during DCC					
TQI_DCC_FAST_INC	V/O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for fast torque intervention during DCC					
TQI_DCC_SLOW_INC	V/O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for slow torque intervention during DCC					
LV_DCC_INC_ACT	V/O	0...1H	0...1	1	[-]
Logical variable for torque intervention due to DCC or LDM					
LV_DCC_INC_LIH	V/O	0...1H	0...1	1	[-]
Logical variable for torque intervention LIH due to DCC					
T_DCC_LIH_CTR	V	0...FFH	0...2.55	0.01	[s]
Time counter delay time					

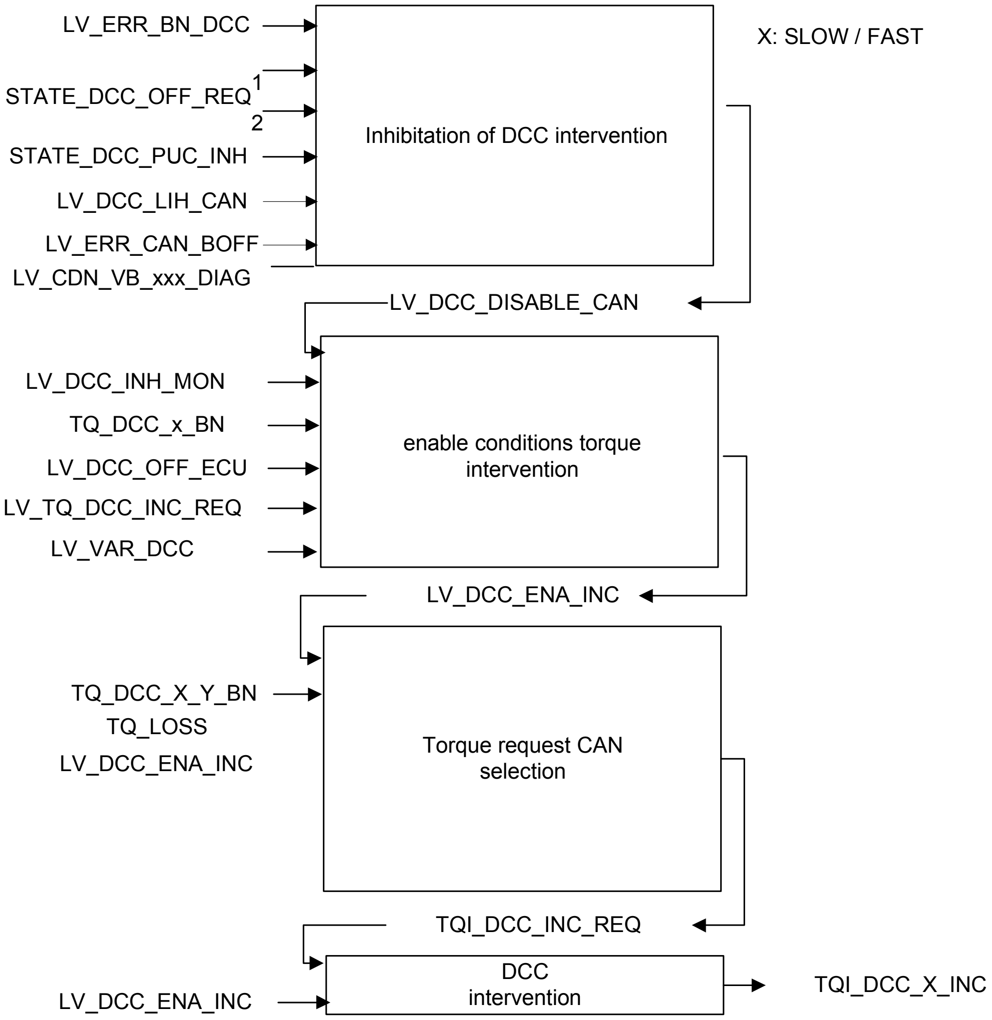
Input data:

LV_ERR_BN_TQ_DCC	LV_DCC_LIH_CAN	STATE_DCC_PUC_INH	LV_VAR_DCC
TQ_DCC_SLOW_BN	LV_IGK	STATE_DCC_OFF_REQ	LV_ERR_CAN_BOFF
TQ_LOSS	LV_DCC_OFF_ECU	LV_DCC_ENA_INC	
TQ_DCC_SLOW_INC_BN	TQ_DCC_FAST_BN	LV_ERR_DCC_INH_MON	LV_TQ_DCC_INC_REQ
TQ_DCC_FAST_INC_BN	LV_CDN_VB_CAN_TQ_DIAG		

General information:

The DCC unit delivers a torque request via BN this module serves the enable conditions and the torque request witch will used in the torque request selection.

Signal flow diagram:



67.4.1 Enable conditions

67.4.1.1 Inhibitation DCC intervention

FUNCTION DESCRIPTION:

If the EMS detects an error on the CAN messages, the logical variable LV_DCC_DISABLE_CAN is set to 1.

Application conditions:

Initialisation:	at reset and at transition LV_IGK off --> on
	LV_DCC_DISABLE_CAN = 0
Recurrence:	10 ms
Activation:	LV_IGK = 1 and LV_VAR_DCC = 1
Deactivation:	-

Formula section:

CAN enviroment DCC inhibition:

```
if      LV_ERR_BN_TQ_DCC          = 1      or
        LV_ERR_CAN_BOFF           = 1      or
        LV_DCC_LIH_CAN            = 1      or
        STATE_DCC_PUC_INH         = 03H    or
        STATE_DCC_OFF_REQ         ≠ 02H    or
        LV_CDN_VB_CAN_TQ_DIAG = 0

then        LV_DCC_DISABLE_CAN    = 1
else        LV_DCC_DISABLE_CAN    = 0
endif
```

67.4.1.2 enable conditions torque intervention

FUNCTION DESCRIPTION:

Depending on the configuration and plausible torque interfaces an increaced torque intervention is possible.

Application conditions:

Initialisation:	at reset and at transition LV_IGK off --> on
	LV_DCC_ENA_INC = 0

Recurrence: 10 ms

Activation: LV_IGK = 1 and LV_VAR_DCC =1

Formula section:

Torque increase intervention:

```
if      LV_DCC_OFF_ECU = 0                      and
        LV_TQ_DCC_INC_REQ = 1                   and
        LV_DCC_DISABLE_CAN =  0                 and
        TQ_DCC_SLOW_BN          <> 8000H        and
        TQ_DCC_FAST_BN          <> 8000H        and
        LV_ERR_DCC_INH_MON = 0
then    LV_DCC_ENA_INC   = 1
else    LV_DCC_ENA_INC   = 0
endif
```

STATE_TQ_DCC_PLAUS

```
if        LV_DCC_LIH_CAN          = 1      or    BIT [00] set
          STATE_DCC_PUC_INH       = 03H    or    BIT [01] set
          STATE_DCC_OFF_REQ       ≠ 02H    or    BIT [02] set
if  LV_DCC_OFF_ECU = 1                           BIT [03] set
if  LV_TQ_DCC_INC_REQ = 1
Then IF   TQ_DCC_SLOW_BN     == 8000H            BIT [05] set
          TQ_DCC_FAST_BN     == 8000H
else                                             BIT [04] set
endif
```

67.4.2 Torque request CAN selection

FUNCTION DESCRIPTION:

If the DCC interventions enabled (LV_DCC_ENA_INC = 1), the torque request is received from CAN and copied into TQI_DCC_FAST/SLOW_INC. The slow and fast path are different messages.
After LV_DCC_INC_ACT changes back to 0, both TQI_DCC_FAST_INC and TQI_DCC_SLOW_INC becomes their minimum values again.

Application conditions:

Initialisation: at reset and at transition LV_IGK off --> on

TQI_DCC_FAST_INC_REQ /_1 = 0 Nm
TQI_DCC_SLOW_INC_REQ /_1 = 0 Nm

Recurrence: 10 ms
Activation: at LV_IGK = 1 and LV_VAR_DCC = 1
Deactivation: -

Formula section:

```
TQI_DCC_SLOW_INC_REQ_1 = TQ_DCC_SLOW_INC_BN - TQ_LOSS
TQI_DCC_FAST_INC_REQ_1 = TQ_DCC_FAST_INC_BN - TQ_LOSS
IF        LV_DCC_ENA_INC = 1
Then      TQI_DCC_SLOW_INC_REQ = TQI_DCC_SLOW_INC_REQ_1
          TQI_DCC_FAST_INC_REQ = TQI_DCC_FAST_INC_REQ_1
Else      TQI_DCC_SLOW_INC_REQ = 0 Nm
          TQI_DCC_FAST_INC_REQ = 0 Nm
```

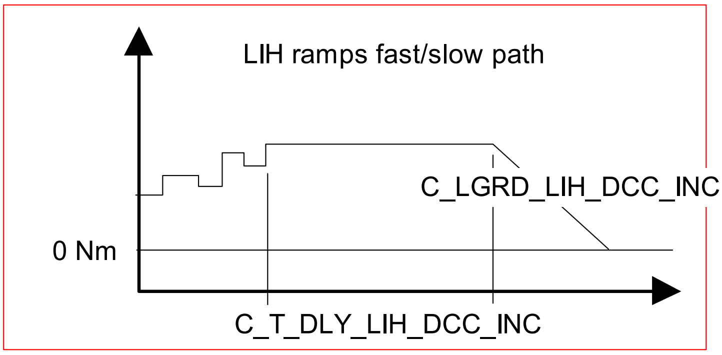
Endif

67.4.2.1 DCC increased intervention and emergency operation

FUNCTION DESCRIPTION:

If during an active DCC intervention an DCC or an EMS error occurs, the current torque offset is continued for the time C_T_DLY_LIH_DCC_INC and then subsequently decreased to TQI_DCC_FAST_INC = TQI_DCC_SLOW_INC = 0 Nm using C_LGRD_LIH_DCC_INC. The DCC unit itself controls the torque leading during active LV_TQ_DCC_INC_REQ. therefore you will haven't a plausible Torque request without LV_DCC_ENA_INC.

Signal flow diagram:



Application conditions:

Initialisation: at reset and at transition LV_IGK off --> on

LV_DCC_INC_LIH = 0

LV_DCC_INC_ACT = 0

TQI_DCC_FAST_INC = TQI_DCC_SLOW_INC = 0Nm

T_DCC_LIH_CTR = 0

Recurrence: 10 ms

Activation: LV_IGK = 1 **and** LV_VAR_DCC = 1

Formula section:

DCC_INC intervention and detection of DCC_INC limp home operation :

```
If(1)     TQI_DCC_FAST_INC_REQ_1 ≠ 0 Nm     and
          TQI_DCC_SLOW_INC_REQ_1 ≠ 0 Nm     and
          LV_DCC_ENA_INC = 0
then(1)
    if(2)      TQI_DCC_FAST_INC > 0 Nm   or   TQI_DCC_SLOW_INC > 0 Nm
    then(2)        LV_DCC_INC_LIH = 1
                   $T\_DCC\_LIH\_CTR_{N} = T\_DCC\_LIH\_CTR_{N-1} + 10\ ms$
```

```
                                            ( till T_DCC_LIH_CTR = C_T_DLY_LIH_DCC_INC )
    if(3)               $T\_DCC\_LIH\_CTR < C\_T\_DLY\_LIH\_DCC\_INC$
  (time delay)
    then(3)     $TQI\_DCC\_FAST\_INC_{N} = TQI\_DCC\_FAST\_INC_{N-1}$        (last valid value)
                $TQI\_DCC\_SLOW\_INC_{N} = TQI\_DCC\_SLOW\_INC_{N-1}$        (last valid value)
    else(3)     $TQI\_DCC\_FAST\_INC_{N} = TQI\_DCC\_FAST\_INC_{N-1} -$
                                                  $C\_LGRD\_LIH\_DCC\_INC$
                                            (till TQI_DCC_FAST_INC = 0Nm )
                $TQI\_DCC\_SLOW\_INC_{N} = TQI\_DCC\_FAST\_INC_{N}$
    endif(3)
```

```
        else(2)     LV_DCC_INC_LIH  =  0
                    T_DCC_LIH_CTR  =  0
        endif(2)
    else(1)         LV_DCC_INC_LIH  =  0
                    T_DCC_LIH_CTR  =  0
                    TQI_DCC_FAST_INC  =  TQI_DCC_FAST_INC_REQ
                    TQI_DCC_SLOW_INC  =  TQI_DCC_SLOW _INC_REQ
    endif(1)
```

DCCT intervention active (LV_DCC_ACT_INC) :

```
if      TQI_DCC_SLOW_INC > 0 Nm                    or
        TQI_DCC_FAST_INC  > 0 Nm
then    LV_DCC_INC_ACT   = 1
else    LV_DCC_INC_ACT   = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LGRD_LIH_DCC_INC	1	0...7FFFH	0...1023.96875	0.03125	[Nm/10ms]
Limiting gradient for DCC emergency ramp function (default 0.6 Nm/10ms)					
C_T_DLY_LIH_DCC_INC	1	0...FFH	0...2.55	0.01	[s]
Delay time for activation of DCC emergency ramp function (default 0s)					

67.5 Torque Request for Cruise Control

67.5.1 Torque request of DME internal Cruise control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_TQ_REQ_CRU	V/O	0...FFFFH	0...1.99996	3.05E-05	[-]
Scaling factor for requested torque at clutch from cruise control					

Input data:

FAC_TQ_CRU	LV_CRU_ACT	VS_FIL	VS_SP_DRIV_CRU
FAC_TQ_REQ_DRIV	FAC_TQ_CRU_INI	LV_VAR_DCC	LV_VAR_BN_LDM

General information:

Depending on the cruise controller manipulated variable FAC_TQ_CRU the interpretation of torque request is calculated. Regarding the different resolutions the output of cruise control is converted to resolutions at the torque structure interface.

Application conditions:

Activation: every engine state and LV_VAR_DCC = 0 and LV_VAR_BN_LDM = 0

Recurrence: 100 ms

Initialization: FAC_TQ_REQ_CRU = 0 at reset

Formula section:

General calculation after initialisation:

$$FAC\_TQ\_REQ\_CRU = \frac{FAC\_TQ\_CRU}{99{,}6\%}$$

At initialisation of Cruise control:

To get a smoothing transition into cruise controlled drive the initialisation ramp is depends on vehicle speed in relation the vehicle speed demand by driver:

```
This query once :
IF (1)       LV_CRU_ACT = 0 to 1
             FAC_TQ_CRU / 99,6% > FAC_TQ_REQ_DRIV
Then (1.1)   at the first time FAC_TQ_REQ_CRU = FAC_TQ_REQ_DRIV
    This query until the end condition (until (1)) is fulfilled:
    IF (2) VS_FIL ≤ VS_SP_DRIV_CRU
    Then (2)
        FAC_TQ_REQ_CRU = FAC_TQ_REQ_CRU + C_LGRD_TQ_CRU_INC_1
    Else (2)
        FAC_TQ_REQ_CRU = FAC_TQ_REQ_CRU + C_LGRD_TQ_CRU_INC_2
  Until (1.1) FAC_TQ_REQ_CRU ≥ FAC_TQ_CRU / 99,6%
Else (1)
             $FAC\_TQ\_REQ\_CRU = \frac{FAC\_TQ\_CRU}{99,6\%}$
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LGRD_TQ_CRU_INC_1	1	0...FFFFH	0...1.99996	3.05E-05	[-]
Change limitation for cruise control requesrt out of lower vehicle speed					
C_LGRD_TQ_CRU_INC_2	1	0...FFFFH	0...1.99996	3.05E-05	[-]
Change limitation for cruise control requesrt out of higher vehicle speed					

67.5.2 Torque request an STATE of DCC and STATE of LDM

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>STATE_TQ_DCC</td><td>V/O</td><td>0H
1H
8H
9H
AH
BH</td><td>0
1
2
3
4
5</td><td>0</td><td>[-]</td></tr>
<tr><td colspan="6">State of DCC torque intervention</td></tr>
<tr><td>STATE_TQ_LDM</td><td>V/O</td><td>0H
1H
8H
9H
AH
BH</td><td>0
1
2
3
4
5</td><td>0</td><td>[-]</td></tr>
<tr><td colspan="6">State of LDM torque intervention</td></tr>
<tr><td>LV_TQ_DCC_ACT</td><td>V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Torque structure leaded by driver but DCC intervention overtaking</td></tr>
<tr><td>FAC_TQ_REQ_DCC</td><td>V/O</td><td>0...FFFFH</td><td>0...1.99996</td><td>3.05E-05</td><td>[-]</td></tr>
<tr><td colspan="6">Scaling factor of inverse DCC model</td></tr>
<tr><td>TQ_DELTA_MAX_CLU</td><td>V</td><td>8000...7FFFH</td><td>-1024...1023.96875</td><td>0.03125</td><td>[Nm]</td></tr>
<tr><td colspan="6">Difference between TQ_MAX_CLU and TQ_MIN_CLU</td></tr>
</table>

Input data:

LV_VAR_DCC	PV_AV	TQI_REQ_TRA	TQI_DCC_FAST_INC
LV_IGK	LV_KD	LV_DCC_ENA_INC	TQ_MIN_CLU
TQ_MAX_CLU	TQ_LOSS	FAC_TQ_REQ_DRIV	TQ_WHEEL_LDM_BN
LV_LDM_ENA	GR_DT	LV_LDM_DRIV_ACT	LV_TQ_WHEEL_LDM_BN_ERR

FUNCTION DESCRIPTION:

General information:

The distance cruise control, the LDM and the transmission unit needs an information witch system (driver, DCC, LDM or CRU) leads the torque system. Therefore the compare between driver which and DCC/LDM is done.

Application conditions:

Initialisation: STATE_TQ_DCC = 08H
STATE_TQ_LDM = 08H
FAC_TQ_REQ_DCC = 0

Recurrence: 10 ms

Activation: LV_VAR_DCC = 1 **or** LV_VAR_BN_LDM = 1 (and LV_IGK = 1)

Formula section:

State of DCC/LDM in DME:

```
IF(1)       LV_VAR_DCC = 1
Then(1)     IF (2a) LV_KD = 0
            Then(2a)    IF(3) LV_DCC_ENA_INC = 1        // "active"
                        Then (3) IF (4) PV_AV > 0 %
                                 Then(4) IF(5) (TQI_REQ_TRA > TQI_DCC_FAST_INC )or
                                               (FAC_TQ_REQ_DRIV > C_FAC_DCC_FL)
                                         Then(5)    STATE_TQ_DCC = 0AH
                                                    " DCC active overtaking by driver"
                                         Else (5)   STATE_TQ_DCC = 01H
                                                    " DCC and Driver active"
                                         Endif(5)
                                 Else(4)            STATE_TQ_DCC = 00H
                                                    " driver passive DCC active"
                                 Endif(4)
                        Else (3) If(6)      PV_AV = 0 %
                                 Then(6)    STATE_TQ_DCC = 08H
                                            " driver DCC passive"
                                 Else(6)    STATE_TQ_DCC = 09H
                                            " active driver no DCC"
                                 Endif(6)
                        Endif(3)
            Else(2a)    STATE_TQ_DCC = 0BH "Kickdown active"
            Endif(2a)
            IF(2b)      STATE_TQ_DCC = 0AH
            Then(2b)    LV_TQ_DCC_ACT = 1
            Else(2b)    LV_TQ_DCC_ACT = 0
            Endif(2b)
Else (1)    if(7)   LV_VAR_BN_LDM = 1
            Then(7)     IF (8)  LV_KD = 0
                        Then(8) IF(9) LV_LDM_ENA = 1    // "active"
                                Then (9) IF (10) PV_AV > 0 %
                                         Then(10)   IF(11) LV_LDM_DRIV_ACT = 0
                                                    Then(11) STATE_TQ_LDM = 0AH
```

```
                                                          " LDM active overtaking by driver"
                                                          Else (11) STATE_TQ_LDM = 01H
                                                          " LDM and Driver active"
                                                          Endif(11)
                                              Else(10)    STATE_TQ_LDM = 00H
                                                    " driver passive LDM active"
                                              Endif(10)
                                        Else (9) If(12) IF PV_AV = 0 %
                                              Then(12) STATE_TQ_LDM = 08H
                                                    " driver LDM passive"
                                              Else(12) STATE_TQ_LDM = 09H
                                                    " active driver no LDM"
                                              Endif(12)
                                        Endif(9)
                                  Else(8) STATE_TQ_LDM = 0BH "Kickdown active"
                                  Endif(8)
                      Endif(7)
          Endif(1)
```

Inverse DCC Torque request model:

TQ_REQ_DCC = TQI_DCC_FAST_INC + TQ_LOSS

TQ_DELTA_MAX_CLU = TQ_MAX_CLU - TQ_MIN_CLU

TQ_LDM_BN_CLU = TQ_WHEEL_LDM_BN / GR_DT

```
If    (LV_VAR_BN_LDM = 1       and    LV_TQ_WHEEL_LDM_BN_ERR = 0)        or
      (LV_VAR_DCC = 1          and    LV_DCC_ENA_INC = 1)
Then        FAC_TQ_REQ_DCC =
            MIN(1; (MAX(TQ_REQ_DCC;TQ_LDM_BN_CLU) -TQ_MIN_CLU)  /
                                                   (TQ_DELTA_MAX_CLU))
Else        FAC_TQ_REQ_DCC = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_DCC_FL	1	0...8000H	0...1	3.05E-05	[-]
Full load detection of DCC					

67.6 Cruise control (CRU)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_CRU_MAIN_SWI	V/O	0...1H	0...1	1	[-]
Indicate CRU readiness					
REQ_MSW	V/O	0H 1H 2H 3H 4H 5H 6H 7H 8H 9H	HARD_OFF SOFT_OFF TIP_UP TIP_DOWN SET_ACCELLE RATE RESUME DECELERATE NONE MARK_HIGH MARK_DOWN	0	[-]
Evaluated Status of MSW request module					
STATE_MSW_CAN	V/O	0H 1H 2H 4H 7H	NONE SET__ACCELE RATE__TIP_UP DECELERATE_ _TIP_DOWN OFF ERROR	0	[-]
Status of MSW to the CAN Message					

Input data:

STATE_MSW_DATA	LV_CRU_ACT	LV_CRU_OFF_IRR	LV_VAR_BN_MSW
LV_IGK	LV_PROG_STEP_1	VS_CRU	VS_MIN_CRU
C_VS_MIN_PROG	LV_ERR_MSW_3	LV_CRU_MAIN_SWI	LV_ERR_MSW_2
LV_VAR_MSW	V_IGK	C_VB_MIN_MSW_DIAG	STATE_BN_MSW
LV_MSW_MSG_VLD	LV_ERR_MSW_TOG	LV_VAR_DCC	LV_ERR_BN_MSW
LV_ERR_CAN_BOFF	LV_VAR_BN_MSW	LV_VAR_BN_LDM	

67.6.1 Summary

General information:

The cruise control (CRU) controls the vehicle speed by a P/I controller to the required target speed. The CRU button in the multifunction steering wheel activates the control functions, if no cut-off conditions or switch-on prohibitions are present. The WHEEL signal of the ASC 1 massage is evaluated as controlled variable. Corresponding to the existing control deviation, the value PV_CRU is calculated for controlling the charge intervention.

Using the system, the driver can activate a control for maintaining the desired speed set by him. The desired speed can be approached comfortably in positive and negative direction by setting the instantaneous speed or by additional functions. Controlled approach to a stored desired speed is also possible. The driver then can store or change his desired speed using the following functions:

- Setting/acceleration	(SET/ACC)
- Deceleration	(DEC)
- Resume	(RES)

- Tip-up	(TP_UP)
- Tip-down	(TP_DOWN)
- OFF	(OFF)

67.6.1.1 Constant drive

General information:

After reaching the desired speed, the vehicle speed is kept constant independent of driving resistance changes which act on the vehicle, e.g. rising and falling gradient or headwind, if there are no abnormal termination conditions and the engine power or the braking effect is sufficient.

67.6.1.2 Setting/acceleration (SET/ACC)

General information:

By activating the SET/ACC function, the driver can store the instantaneous speed as desired speed and thus switch on the CRU system. Also the speed can be changed with a defined acceleration from the controlled constant drive. After reaching the desired speed, the constant drive function of the control is active.

67.6.1.3 Deceleration (DEC)

General information:

If the controlled constant drive function is active, the speed reduction takes place in the DEC function by the driver. In this case the desired speed is reduced. If the driver ends the function, the achieved instantaneous speed is stored as new desired speed and the controlled constant drive function is adopted. It is not possible to switch the system on through the recognised DEC function.

67.6.1.4 Resume (RES)

General information:

When the driver activates the RES function, the vehicle is accelerated from the instantaneous speed with a defined acceleration to the stored desired speed. This applies both for larger and for smaller desired speeds compared with the instantaneous speed.

67.6.1.5 Tip-up

General information:

Proceeding from controlled constant drive, the driver can increase his desired speed in steps of 1 k.p.h. by activating the TP_UP function. This function can be selected several times consecutively, thus the vehicle can be brought to a new higher desired speed with a defined acceleration using the TP_UP function.

67.6.1.6 Tip-down

General information:

Proceeding from controlled constant drive, the driver can reduce his desired speed in steps of 1 k.p.h. by activating the TP_DOWN function. This function can be selected several times consecutively, thus the vehicle can be brought to a new lower desired speed with a defined deceleration using the TP_DOWN function.

67.6.1.7 Off

General information:

To switch the system off, the driver can select the OFF function of the multifunction steering wheel. In this way the cruise control system is switched off softly, whereby the achieved desired speed remains stored for later activation by RES. If the engine running is stopped (LV_IGK = 0, ES), then the readiness of CRU and the speed-setpoint are reduced to their initial values.

67.6.2 Ready state and display

General information:

A prerequisite for activating the CRU functions is the CRU ready state. After ECU initialization, the CRU state is initially passive. C_T_READY_CRU after ignition key on LV_IGK = 1 can be transformed into the ready state by pressing the OFF button once. To indicate CRU readiness to the driver, the CAN signal LV_CRU_MAIN_SWI is set from 0 to 1. In this way the CRU ready lamp is activated in the CAN combination device.

The transition into the passive state occurs by pressing the OFF button once from the CRU ready state without controlled drive. The CRU ready lamp goes out.

For this functionality, the OFF button must be recognized for the minimum time of C_T_CRUS_ON on actuation; the OFF button must not be actuated for at least C_T_CRUS between switch on ready and switch off ready.
The ready state is then also ended (-> passive) if the MSW diagnosis is momentarily not possible or a MSW error is momentarily present, see "Diagnosis and limp home" chapter.
Depending on the vehicle configuration LC_CRU_REQ_SWI is switched between Multifunction steering wheel and Switch pack. The reason is, in a variant with switch pack is the main switch of cruise control on with LV_IGK = 1.

Application conditions:

Initialisation: LV_CRU_MAIN_SWI =0

Recurrence: 10ms

Formula section:

```
If (1)  LV_IGK = 1 and T> C_T_READY_CRU  and
        (LC_CRU_REQ_SWI =0 and LV_VAR_BN_MSW = 0) and LV_CRU_OFF_IRR = 0
Then(1) If(2)       LV_CRU _MAIN_SWI = 0
        Then(2) If(3)   REQ_MSW = SOFT_OFF    (for T> C_T_CRUS_ON)     and
                        Time between transition LV_CRU _MAIN_SWI = 0/1
                        (and vice versa)> C_T_CRUS
            Then(3)  LV_CRU _MAIN_SWI = 1
        Else (2) IF(4)  REQ_MSW = SOFT_OFF    (for T> C_T_CRUS_ON)     and
                        Time between transition LV_CRU _MAIN_SWI = 0/1
                        (and vice versa)> C_T_CRUS
            Then(4)IF(5)        LV_CRU_ACT = 0
                Then (5)        LV_CRU_MAIN_SWI = 0
                Else (5)        LV_CRU_MAIN_SWI = 1 (remains )
Else (1)    IF(6)(LC_CRU_REQ_SWI = 0 and LV_VAR_BN_MSW = 1 and
                LV_CRU_OFF_IRR = 0
            Then(6) IF(7) (VS_CRU ≥ VS_MIN_CRU   and
                        REQ_MSW = TIP_UP/_DOWN)            or
                        (VS_CRU > C_VS_MIN_PROG     and
                        REQ_MSW = MARK_UP/_DOWN)
                Then(7)    LV_CRU_MAIN_SWI = 1
                Else(7) IF  (REQ_MSW = SOFT_OFF (for T > C_T_CRUS_OFF) and
                                LC_CRUS_OFF_SWI = 0)
                        Then LV_CRU_MAIN_SWI = 0
                        Else LV_CRU_MAIN_SWI remains
            Else (6) IF     LC_CRU_REQ_SWI = 1
                then        LV_CRU_MAIN_SWI = 1
                else        "LV_CRU_OFF_IRR = =1"
                            LV_CRU_MAIN_SWI = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_READY_CRU	1	64...FFH	1000...2550	10	[ms]
Delay time after control unit initialization					
C_T_CRUS_ON	1	0...FFH	0...2550	10	[ms]
Minimum recognition time on actuating the OFF button					
C_T_CRUS	1	0...FFH	0...2550	10	[ms]
Minimum time between switch on ready and switch off ready					
C_T_CRUS_OFF	1	0...FFH	0...5100	20	[ms]
Minimum recognition time on actuating the OFF button to switch off the display at BN 2000					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
LC_CRU_REQ_SWI	1	0...1H	0...1	1	[-]
Configuration switch between MSW and steering lever					
LC_CRUS_OFF_SWI	1	0...1H	0...1	1	[-]
Configuration switch to switch off the cruise control via MSW					

67.6.3 Evaluation serial data line from the MSW

General information:

The buttons position of multifunction steering wheel (MSW) which is transferred digitally through a serial data line serves as interface between the driver and the CRU system (look at "Basic SW Inputs /Outputs"). The system interfaces are evaluated with a 10 ms time pattern.
A additional condition to the reception of a transfer pulse (see MSW interface in the Basic SW chapter) for reading the Output of the MSW-IC memory is further the MSW Diagnosis conditions.

Description:

The debounced and recognised function of the driver interface is evaluated only after the conditions for the CRU mode have been fulfilled. If no CRU mode is possible in the current driving cycle or at the moment, then "Hard cut-off" (FAC_TQ_CRU = 0) must be triggered.

According to the current status of the CRU and the recognised function of the driver interface, the functions OFF ("Soft cut-off"), resume, setting/acceleration, Tip-Up, Tip-Down, deceleration and the transition into the controlled constant drive can be triggered.

The function setting/acceleration (SET/ACC) is triggered if the CRU is ready, is in constant drive or another active function.

The deceleration function (DEC) is triggered accordingly, but requires that the CRU is in controlled drive, i.e. there is no setting mode in this case.

The tip functions Tip Up and Tip Down are triggered if the controlled constant drive or one of the two tip functions is active. Furthermore, a Tip-Up in CRU ready or with running resume triggers the setting function.

The Tip-Down function in active resume mode triggers only the abnormal termination of the running function and sets VS_SP_DRIV_CRU to the actual speed VS_FIL, but allows no setting in the above sense.

Precondition for a renewed activation of CRU-functions after cut-off or abnormal terminations are generally that the CRU mode is possible in principle and at least 'Rest' (None) was recognised once. As soon as 'None' or still no new function has be recognised after 'Setting/acceleration' or 'Deceleration', 'Transition into constant drive' is triggered. This applies especially for the resume function (RES) which is interlocked until rest is recognised again.

Resume is triggered only at CRU ready without controlled drive.

The following applies in an overtaking process: The triggering for hard and soft cut-off comes through. The same applies for functions which increase the value FAC_TQ_CRU, such as acceleration and TIP UP, in order to is able to "overdrive" in this way PV_AV with the CRU. For all other functions, especially for "Deceleration" and "TIP DOWN", which reduce the FAC_TQ_CRU, the trigger is prevented, since the instantaneous driver wish via pedal is viewed as higher priority. With recognised RES position, triggering would make no sense.
For a possibly multiple actuation of buttons and superimposed CRU functions, priority must be given to the function recognition according to the following significance:

Significance	Button/function
1	Brake light switch
1	Clutch switch
1	Transmission switch runup lock
2	OFF
3	Deceleration
4	Setting/acceleration
5	Resume

Significance 1 has highest priority

The function selection is made according to priority (see above). The tip functions are executed immediately after recognition. In this way the vehicle response should be improved for the driver, if he requests the SET/ACC or DEC function. In a pure tip request, rest (None) must be recognised between the individual tips.

Application conditions:

Initialization: *STATE_MSW_CAN = "None"* *at reset and deactivation*

REQ_MSW / = "Hard_off"

Recurrence: 10ms

Activation: LV_IGK = 1 and V_IGK > C_VB_MIN_MSW_DIAG

Time (since V_IGK > C_VB_MIN_MSW_DIAG) > C_T_READY_CRU

(Diagnosis conditions)

(LV_VAR_MSW = 1 **or** (LV_VAR_BN_MSW = 1**and** LV_VAR_DCC = 0 **and** LV_VAR_BN_LDM = 0))

Deactivation: otherwise

Formula section:

The evaluation of MSW is only done by CAN 11H variants. In BN 2000 the MSW requests are done by instument message.

IF LV_VAR_BN_MSW = 0

Then

The Signal of STATE_MSW_DATA has to be received multiple (LV_MSW_MSG_VLD = 1) before it is considered in the REQ_MSW evaluation. The following distinctions are made because of the evaluation of STATE_MSW_DATA and an additional time condition:

REQ_MSW	STATE_MSW_DATA
0. HARD OFF:	MSW not allowed, ready or available
1. SOFT OFF:	OFF pressed.
2. TP_UP:	SET/ACC and no DEC pressed for < C_T_MIN_MSW.
3. TP_DOWN:	DEC and no SET/ACC pressed for < C_T_MIN_MSW.
4. SET/ACC:	SET/ACC and no DEC pressed for ≥ C_T_MIN_MSW.
5. RES:	RES pressed.
6. DEC:	DEC and no SET/ACC pressed for ≥ C_T_MIN_MSW .
7. NONE:	SET/ACC and DEC and RES and OFF not pressed.

STATE_MSW_CAN	STATE_MSW_DATA
0. None:	
1. SET/ACC /TP_UP:	SET/ACC pressed.
2. DEC /TP_DOWN:	DEC pressed.
3. RES:	RES pressed.
4. OFF:	OFF pressed
5.	not used
6.	not used .
7. Error:	LV_ERR_MSW_2 = 1; LV_ERR_MSW_3 = 1 LV_ERR_MSW_TOG = 1

else

REQ_MSW	STATE_BN_MSW		
0. HARD OFF:	MSW message not valid (7FH or LV_ERR_BN_MSW ==1 or LV_ERR_CAN_BOFF == 1)		
1. SOFT OFF:	TIP UP or DOWN	"OFF pressed"	
2. TIP_UP:	TIP_PRE	pressed for (	≥ C_T_MIN_TIP_MSW. and < C_T_MIN_MSW.)
3. TIP_DOWN:	TIP_POST	pressed for (	≥ C_T_MIN_TIP_MSW. and < C_T_MIN_MSW.)
4. SET/ACC:	TIP_PRE	pressed for	≥ C_T_MIN_MSW.
5. RES:	TIP_AXIAL.		
6. DEC:	TIP_POST	pressed for	≥ C_T_MIN_MSW.
7. NONE:	NONE		
8. MARK_UP	PRESS_PRE		
9. MARK_DOWN	PRESS_POST		

The state REQ_MSW = TIP_UP/TIP_DOWN is held for minimum C_T_MIN_TIP_MSW.

endif

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_T_MIN_MSW	1	1...FFH	0.01...2.55	0.01	[s]
Time delay of REQ_MSW					
C_T_MIN_TIP_MSW	1	0...FFH	0...2.55	0.01	[s]
Time delay of REQ_MSW TIP detection					

67.6.4 Stage cruise control conditions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
NR_STEP_ON	V	0...FFH	0...255	1	[-]
Current vehicle speed step					
COUNT_VS_STEP	V	0...FFH	0...255	1	[-]
Number of used Vehicle speed steps					
VS_SP_DRIV_CRU_CAN	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Driver demand cruise control for INSTR					
VS_DIF_STEP_PROG	V	0...FFFFH	0...511.99218	0.0078125	[km/h]
Differences between steps					
NR_CRU_DIF_STEP	-	0...FFH	0...255	1	[-]
Number of steps between VS_SP_DRIV_CRU_CAN and VS_MIN_CRU					
LV_PROG_STEP_1	V/O	0...1H	0...1	1	[-]
Standing programming on					
LV_STEP_ON	V/O	0...1H	0...1	1	[-]
Step Display on					
LV_STEP_ON_ACT	O	0...1H	0...1	1	[-]
Step Display on activation of resume					
VS_SP_DRIV_STEP	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Active vehicle speed step for the controller					
NR_VS_CRU	V/O	0...FFH	0...255	1	[-]
Cycle numer of message IDX_NO					
VS_STEP_CAN_1	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed Step to the CAN i (1-3)					
VS_STEP_CAN_2	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed Step to the CAN i (1-3)					
VS_STEP_CAN_3	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed Step to the CAN i (1-3)					
VS_STEP_m	O/S	0...FFFFH	0...511.9922	0.0078	km/h
Array of the vehicle speed steps depending on VEH_KEY_NR and Step number					
CTR_CRU_FLASH_LIGHT	-	0...FH	0...15	1	[-]
Number of flash lights					
VS_IF_STEP_m	-	0...FFFFH	0...511.9922	0.0078	km/h
Array of the vehicle speed steps depending on VEH_KEY_NR and Step number					
LV_PROG_STEP_IF_ACT	V	0...1H	0...1	1	[-]
Step Display on					
LV_STEP_ON_ICL	V/O	0...1H	0...1	1	[-]
Step Display on for instrument cluster					
LV_CRU_DISP_HUD	V/O	0...1H	0...1	1	[-]
Display on for HUD					

Input data:

LV_IGK	LV_VAR_BN	LV_VAR_DCC	VS_CRU
STATE_VS_ICL_DISP	VS_SP_DRIV_CRU	LV_ST	STATE_CRU
REQ_MSW	VS_SP_CRU_MAX	LV_CRU_MAIN_SWI	C_T_CRUS_OFF
VS_MIN_CRU	LV_CRU_ACT	VEH_KEY_NR	NR_VS_CRU_IF
LV_MSG_PROG_STEP_CRU	VS_STEP_IF_1	VS_STEP_IF_2	VS_STEP_IF_3
LC_CRUS_OFF_SWI	LV_VAR_BN_LDM		

General information:

The stage cruise control is an add-on function of the cruise control. If is available then it's possible to program steps of vehicle speed witch are shown in the speedometer. The driver can used this function for jumps between fixed Steps during cruise control. The requirement is a steering lever with additional options MARK_UP and MARK_DOWN or the I drive functionality.
Basically there are two different ways of programming during drive or at standing vehicle.

The following assumptions of message from the CCC_MASK are done:

- receiving only at different data's
- the message to the instrument will sent after a complete receive of CCC_MASK message
- depending on the calibration of C_MAX_STEP the quantity of evaluated NR_VS_CRU_IF is limited

Application conditions:

Initialisation: NR_STEP_ON = 0; COUNT_VS_STEP out of NVMY

VS_SP_DRIV_CRU_CAN = VS_MIN_CRU

LV_PROG_STEP_1 = 0

LV_STEP_ON = 0

LV_CRU_DISP_HUD = 0

LV_STEP_ON_ICL = 0

CTR_T_REQ_MSW_CON_MSG = 0

VS_SP_DRIV_STEP = VS_MIN_CRU

VS_STEP_m out of NVMY sorted depending on VEH_KEY_NR (if key available)

T_PROG_STEP_IF_ACT = 0

The vehicle key number is only reed once at Initialisation, at the moment only one array for the first Key is available.

Recurrence: 10 ms

Recurrence: of NR_VS_CRU , VS_STEP_CAN_i 100 *ms to be Synchron with CAN Message*

Activation: LV_VAR_BN = 1 and LV_VAR_DCC = 0 and LV_VAR_BN_LDM = 0

Signal flow diagram:

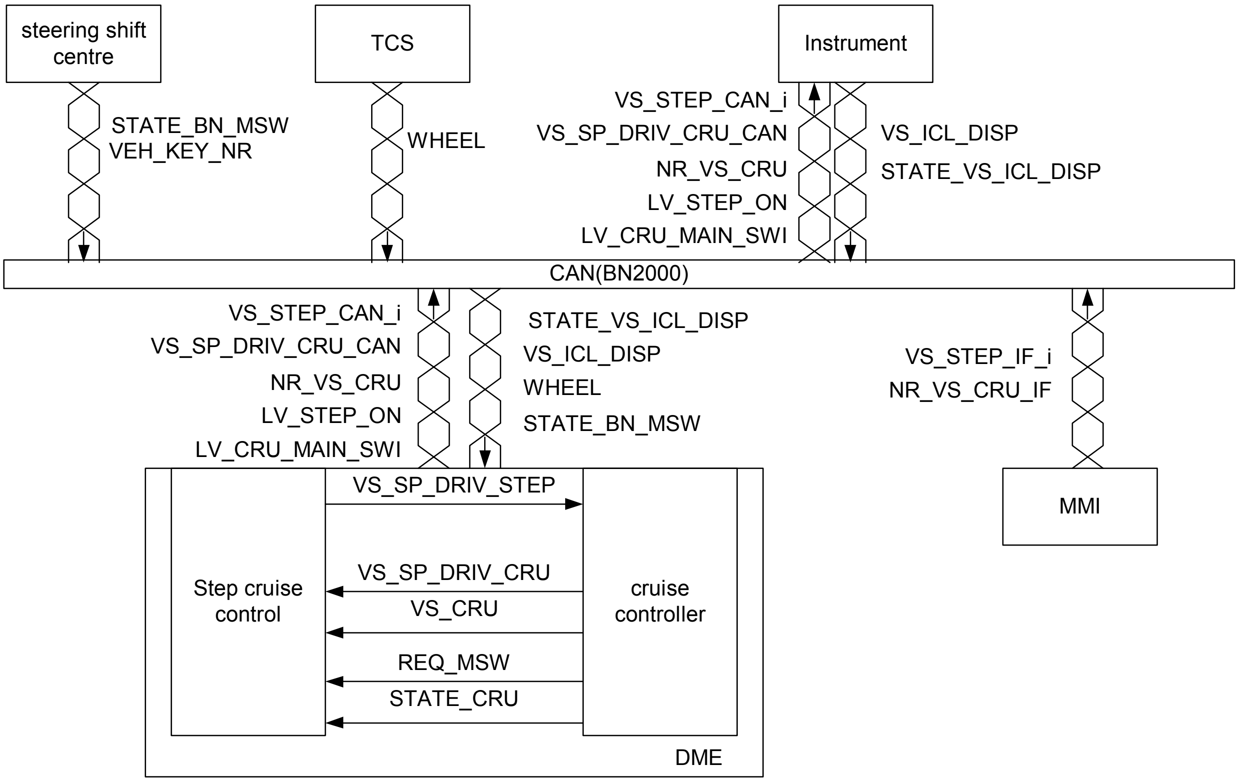


Figure 1: CAN structure for stage cruise control

FUNCTION DESCRIPTION:

Description:

The different states at the step control are triggered only at transitions of respective REQ_MSW state at ramp to active. After detection of an event the respective REQ_MSW State (TIP_UP, DOWN; MARK_UP, DOWN) is looked. MARK_UP, DOWN for C_T_MIN_CRU_CHG or until idle is detected again, the exception is LV_PROG_STEP =0 to 1

Formula section:

Counter:

```
IF      REQ_MSW = SOFT_OFF
THEN    increment CTR_T_REQ_MSW_CON_MSG
ELSE    CTR_T_REQ_MSW_CON_MSG = 0
ENDIF
```

Evaluation of CCC MASK massage information for programming:

Vehicle speed steps of CCC_MASK:

VS_IF_STEP_m: $m = \sum_{14}^{0}$

m: Step number (0...14)

```
IF  LV_MSG_PROG_STEP_CRU == 1 ( new data available)
    LV_MSG_PROG_STEP_CRU needs to be acknowledged after taking data over.

Then    LV_PROG_STEP_IF_ACT = 1
    If      LV_PROG_STEP_IF_ACT = 0 $\rightarrow$ 1
    Then    VS_IF_STEP_m(0...14) = 0
    Else
    Endif
    VS_IF_STEP_m(0...14) = VS_STEP_IF_1
            m = NR_VS_CRU_IF * 3     ;     positions(0,3,6, 9,12)
    VS_IF_STEP_m(0...14) = VS_STEP_IF_2
            m = NR_VS_CRU_IF * 3+1   ;     positions(1,4,7, 10,13)
    VS_IF_STEP_m(0...14) = VS_STEP_IF_3
            m = NR_VS_CRU_IF * 3+2   ;     positions(2,5,8, 11,14)
else VS_STEP_m calculation of
endif
IF  LV_PROG_STEP_IF_ACT = 1
Then    T_PROG_STEP_IF_ACT ++
```

```
Endif
If      receiving C_MAX_STEP / 3 blocks or
        T_PROG_STEP_IF_ACT $\geq$ C_T_PROG_STEP_IF_ACT
Then            LV_PROG_STEP_IF_ACT = 0
                T_PROG_STEP_IF_ACT = 0
                VS_STEP_m = VS_IF_STEP_m (update count_vs_step, note: receive data
                                          can be unsorted)
                [nr_step_on = 0]
Endif
```

Evaluation of key and stage information for programming:

Vevicle speed steps:

VS_STEP_m: $m = \sum_{14}^{0}$

m: Step number (0...14)

	NR_VS_CRU				
	0	1	2	3	4
VS_STEP_CAN_1 =	VS_STEP_0	VS_STEP_3	VS_STEP_6	VS_STEP_9	VS_STEP_12
VS_STEP_CAN_2 =	VS_STEP_1	VS_STEP_4	VS_STEP_7	VS_STEP_10	VS_STEP_13
VS_STEP_CAN_3 =	VS_STEP_2	VS_STEP_5	VS_STEP_8	VS_STEP_11	VS_STEP_14

VS_STEP_m; m 0-14

VS_STEP_CAN_i (i = 1,2,3) = VS_STEP_m

Programed vehicle speed steps:

VS_STEP_k: $k = \sum VS\_STEP\_m$; for VS_STEP > 0 km. p. h.

"0 = kmph means there is no programed VS_CRU at this n_m step"

COUNT_VS_STEP = number of VS_STEP_k

NR_VS_CRU : If a VS_STEP_m is changed the NR_VS_CRU is updated to the respective column number. In other cases and at first activation of Steps the Numbers is cycling updated. Additional the instrument unit itselfs saves the steps internal.

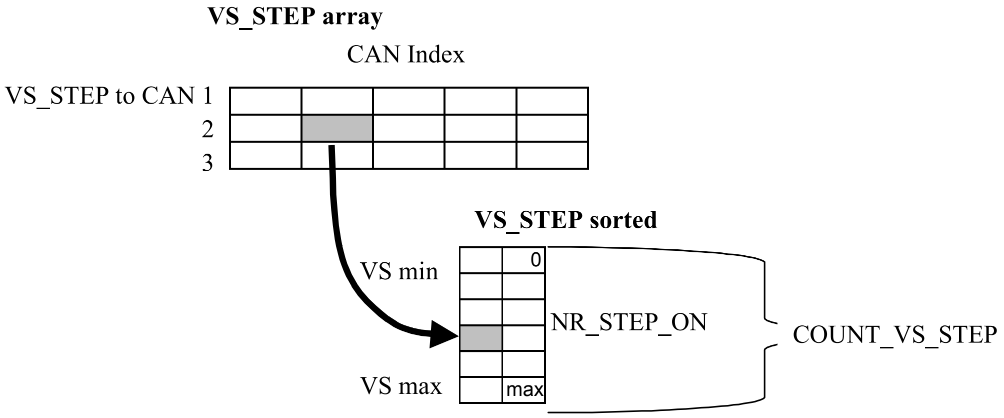
NR_STEP_ON : current step number of sorted Vehicle speed steps.

Concerning some programming rules of the steps, it is necessary to vary some variables in dependancy on the equipped ICL. The difference is, whether the vehicle speed is announced in miles or kilometers.

IF STATE_VS_ICL_DISP = 1 "km/h; mph"

```
Then    VS_DIF_STEP_MIN = C_VS_DIF_STEP_MIN_M
        VS_DIF_TIP_PROG = C_VS_DIF_TIP_PROG_M
Else    VS_DIF_STEP_MIN = C_VS_DIF_STEP_MIN_K
        VS_DIF_TIP_PROG = C_VS_DIF_TIP_PROG_K
Endif
```

Signal flow diagram:



REQ_MSW = MARK UP:

```
IF    STATE_VS_ICL_DISP = 1              "km/h; mph"
Then NR_CRU_DIF_STEP =
            round down( VS_SP_DRIV_CRU_CAN / C_VS_DIF_STEP_PROG_M)
            VS_DIF_STEP_PROG =      C_VS_DIF_STEP_PROG_M *
                                    (1 + NR_CRU_DIF_STEP)   -
                                    VS_SP_DRIV_CRU_CAN
VS_DIF_STEP_PROG  = MAX (VS_DIF_STEP_MIN; VS_CRU_DIF_STEP)
Else
NR_CRU_DIF_STEP =
            round down ( VS_SP_DRIV_CRU_CAN / C_VS_DIF_STEP_PROG_K)
            VS_DIF_STEP_PROG =      C_VS_DIF_STEP_PROG_K *
                                    (1 + NR_CRU_DIF_STEP)   -
                                    VS_SP_DRIV_CRU_CAN
VS_DIF_STEP_PROG  = MAX (VS_DIF_STEP_MIN; VS_CRU_DIF_STEP)
End
```

REQ_MSW = MARK DOWN:

```
IF   STATE_VS_ICL_DISP = 1           "km/h; mph"
Then  NR_CRU_DIF_STEP =
          round down( VS_SP_DRIV_CRU_CAN / C_VS_DIF_STEP_PROG_M)
          VS_ CRU_DIF_STEP =        -1*(C_VS_DIF_STEP_PROG_M *
                                    NR_CRU_DIF_STEP           -
                                    VS_SP_DRIV_CRU_CAN )
     IF       VS_ CRU_DIF_STEP = =0 km/h
     Then     VS_DIF_STEP_PROG  = C_VS_DIF_STEP_PROG_M
     else     VS_DIF_STEP_PROG  = MAX (VS_DIF_STEP_MIN; VS_CRU_DIF_STEP)

Else NR_CRU_DIF_STEP =
          round down( VS_SP_DRIV_CRU_CAN / C_VS_DIF_STEP_PROG_K)
          VS_ CRU_DIF_STEP =        -1*(C_VS_DIF_STEP_PROG_K *
                                    NR_CRU_DIF_STEP           -
                                    VS_SP_DRIV_CRU_CAN)
     IF       VS_ CRU_DIF_STEP = =0 km/h
     Then     VS_DIF_STEP_PROG  = C_VS_DIF_STEP_PROG_K
     Else     VS_DIF_STEP_PROG  = MAX (VS_DIF_STEP_MIN; VS_CRU_DIF_STEP)
End
```

Standing vehicle programming detection LV_PROG_STEP_1:

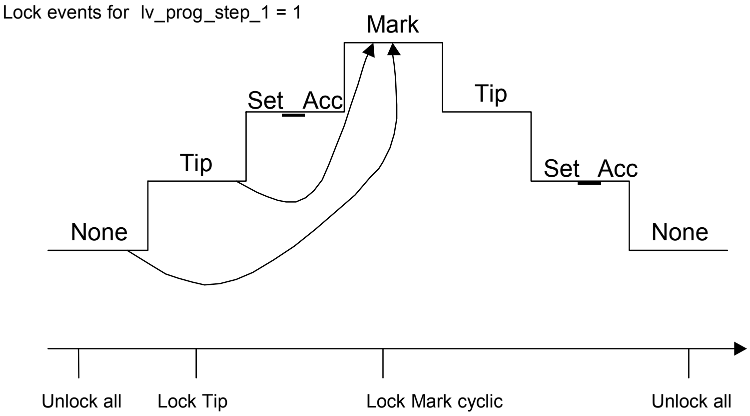
Hysteresis of PROG_STEP_1:

```
IF   VS_CRU  < C_VS_MIN_PROG                          and
(REQ_MSW = MARK_UP or DOWN)                           and
LV_IGK = 1                                            and
LV_PROG_STEP_1 = 0
Then       LV_PROG_STEP_1 = 1
           LV_STEP_ON_PREV = LV_STEP_ON
           After first cycle of NR_VS_CRU; then LV_STEP_ON = 1   "display on"
ElseIF     LV_PROG_STEP_1 = 1                  and
           (VS_CRU  > C_VS_MIN_PROG  or  LV_IGK = 0 or  LV_ST = 1 or
           (CTR_T_REQ_MSW_CON_MSG > C_T_CRUS_OFF and
           LC_CRUS_OFF_SWI = 0)
Then       LV_PROG_STEP_1 = 0
           LV_STEP_ON = LV_STEP_ON_PREV
           LV_STEP_ON_PREV = 0
Endif
```

67.6.4.1 Program conditions at Standing vehicle

Activation: LV_PROG_STEP_1 = 1

MSW detection:



VS_SP_DRIV_CRU_CAN:

First initialisation of VS_SP_DRIV_CRU_CAN:

IF LV_PROG_STEP_1_{n-1} = 0 LV_PROG_STEP_1_n = 1

Then VS_SP_DRIV_CRU_CAN = VS_MIN_CRU

Else "no action"

Endif

Calculation of VS_SP_DRIV_CRU_CAN:

```
REQ_MSW = MARK_UP
#----------------------------------------------------------------
No step available:
IF    COUNT_VS_STEP = 0
Then        VS_SP_DRIV_CRU_CAN = min((VS_SP_DRIV_CRU_CAN_(n-1) +
                                      VS_DIF_STEP_PROG); VS_SP_CRU_MAX )

----------------------------------------------------------------
step availible (goto next higher step):
Else IF VS_SP_DRIV_CRU_CAN < max (VS_STEP_m)
    then            VS_SP_DRIV_CRU_CAN     = VS_STEP_m
            [for m = $m=\sum_{k}^{0}$ ; min(abs(VS_SP_DRIV_CRU_CAN- VS_STEP_m)) and
                          VS_SP_DRIV_CRU_CAN < VS_STEP_m]
    Else    IF COUNT_VS_STEP < C_MAX_STEP
            then VS_SP_DRIV_CRU_CAN = min((VS_SP_DRIV_CRU_CAN_(n-1) +
                                      VS_DIF_STEP_PROG); VS_SP_CRU_MAX)

            Else    "NR_STEP_ON = C_MAX_STEP"
                    $VS\_SP\_DRIV\_CRU\_CAN_{n}$ = $VS\_SP\_DRIV\_CRU\_CAN_{(n-1)}$

Endif
----------------------------------------------------------------#
REQ_MSW = MARK_DOWN:

#----------------------------------------------------------------
No step availible:
IF    COUNT_VS_STEP = 0
Then  VS_SP_DRIV_CRU_CAN = max ((VS_SP_DRIV_CRU_CAN_(n-1) -
                                      VS_DIF_STEP_PROG); VS_MIN_CRU)
            ----------------------------------------------------------------
step availible (goto next lower step):
Else IF VS_SP_DRIV_CRU_CAN > min (VS_STEP_m; m= 0 to k)
    then                VS_SP_DRIV_CRU_CAN = VS_STEP_m
            [for m = $m=\sum_{k}^{0}$ ; min(abs(VS_SP_DRIV_CRU_CAN- VS_STEP_m)) and
```

```
                                        VS_SP_DRIV_CRU_CAN > VS_STEP_m]
    Else   IF COUNT_VS_STEP < C_MAX_STEP
           Then  VS_SP_DRIV_CRU_CAN = max ((VS_SP_DRIV_CRU_CAN$_{(n-1)}$ -
                                                   VS_DIF_STEP_PROG);
                                               VS_MIN_CRU)
           Else  VS_SP_DRIV_CRU_CAN $_{n}$ = VS_SP_DRIV_CRU_CAN $_{(n-1)}$
Endif
--------------------------------------------------------------------#
```

REQ_MSW = TIP_UP

```
#-------------------------------------------------------------------
No step availible:
IF   COUNT_VS_STEP = 0
Then       VS_SP_DRIV_CRU_CAN = (VS_SP_DRIV_CRU_CAN$_{(n-1)}$ +
                                    VS_DIF_TIP_PROG)
--------------------------------------------------------------------
"step availible ":
Else IF COUNT_VS_STEP < C_MAX_STEP
    Then  "tip up"
          VS_SP_DRIV_CRU_CAN = VS_SP_DRIV_CRU_CAN$_{(n-1)}$ +
                                         VS_DIF_TIP_PROG
          "Proofing of Step range catch function"
          IF     |VS_SP_DRIV_CRU_CAN - VS_STEP_m| < VS_DIF_STEP_MIN
                 [for m = $m = \sum_{k}^{0}$ ; min(abs(VS_SP_DRIV_CRU_CAN- VS_STEP_m)) ]

          Then   IF  VS_SP_DRIV_CRU_CAN < = VS_STEP_m
                 Then  VS_SP_DRIV_CRU_CAN = VS_STEP_m
                 else  VS_SP_DRIV_CRU_CAN =
                                 VS_STEP_m + VS_DIF_STEP_MIN
                     IF VS_SP_DRIV_CRU_CAN >
                                 VS_STEP_m (+1) - VS_DIF_STEP_MIN
                     Then VS_SP_DRIV_CRU_CAN = VS_STEP_m $_{(m+1)}$
                     else    remains
                     endif
```

```
        else "goto next higher step or stay on highest step"
                min (VS_SP_DRIV_CRU_CAN = VS_STEP_m ; max (VS_STEP_k))
                    [for m = $m = \sum_{k}^{0}$ ; min(abs(VS_SP_DRIV_CRU_CAN- VS_STEP_m)) and
                                VS_SP_DRIV_CRU_CAN < VS_STEP_m]
        endif
    endif
    VS_SP_DRIV_CRU_CAN = min (VS_SP_DRIV_CRU_CAN; VS_SP_CRU_MAX)
    -------------------------------------------------------------------#
```

REQ_MSW = SET/ACC:

Calculaion like "REQ_MSW = TIP_UP", but in a recurrency of the time delay C_T_DLY_CRU_TIP.

```
T= C_T_DLY_CRU_TIP
IF    T = 0 ms
Then        next cycle calculation
else        T = T -10 ms
endif
```

REQ_MSW = TIP_DOWN:

```
#----------------------------------------------------------------
No step availible:
IF   COUNT_VS_STEP = 0
Then   VS_SP_DRIV_CRU_CAN = (VS_SP_DRIV_CRU_CAN$_{(n-1)}$ - VS_DIF_TIP_PROG)
----------------------------------------------------------------
"step availible tip down":
Else IF COUNT_VS_STEP < C_MAX_STEP
     Then   "tip down"
            VS_SP_DRIV_CRU_CAN = VS_SP_DRIV_CRU_CAN$_{(n-1)}$ -
                                               VS_DIF_TIP_PROG
            "Proofing of Step range catch function"
            IF     |VS_SP_DRIV_CRU_CAN - VS_STEP_m|  <  VS_DIF_STEP_MIN
                   [for m =  $m=\sum_{k}^{0}$  ; min(abs(VS_SP_DRIV_CRU_CAN- VS_STEP_m)) ]
            Then   IF  VS_SP_DRIV_CRU_CAN => VS_STEP_m
                   Then  VS_SP_DRIV_CRU_CAN = VS_STEP_m
                   else  VS_SP_DRIV_CRU_CAN =
                                     VS_STEP_m - VS_DIF_STEP_MIN
                         IF VS_SP_DRIV_CRU_CAN < =
                                     VS_STEP_m (-1) + VS_DIF_STEP_MIN
                         Then VS_SP_DRIV_CRU_CAN = VS_STEP_m (m-1)
                         else    remains


            endif
     else "goto next lower step or stay on lowest step"
            max (VS_SP_DRIV_CRU_CAN = VS_STEP_m ; MIN(VS_STEP_k))
               [for m =  $m=\sum_{k}^{0}$  ; min(abs(VS_SP_DRIV_CRU_CAN- VS_STEP_m)) and
                           VS_SP_DRIV_CRU_CAN > VS_STEP_m]
     endif
endif
VS_SP_DRIV_CRU_CAN = max (VS_SP_DRIV_CRU_CAN ; VS_MIN_CRU)
----------------------------------------------------------------#
```

REQ_MSW = DECC:

Calculaion like "REQ_MSW = TIP_DOWN", but in a recurrency of the time delay C_T_DLY_CRU_TIP.

```
T= C_T_DLY_CRU_TIP
IF    T = 0 ms
Then        next cycle calculation
else        T = T -10 ms
endif
```

VS_SP_DRIV_CRU_CAN is allways limited in the range between VS_MIN_CRU and C_VS_SP_MAX_CRU.

Saving current Step VS_STEP_m:

(REQ_MSW = RES and LV_PROG_STEP_IF_ACT== 0) for T > C_T_MIN_PROG

```
--------------------------------------------------------------------------------
Set:
IF(3)     VS_SP_DRIV_CRU_CAN =><= VS_STEP_m ;$(m = \sum_{k}^{0})$ ± VS_DIF_STEP_MIN
Then(3)     VS_STEP_m; m=k+1= VS_SP_DRIV_CRU_CAN
                          For index m : VS_STEP_$m_{(n-1)}$ = 0 km.p.h.
--------------------------------------------------------------------------------
erase:
Else(3) VS_STEP_m; m=k  = 0 km.p.h (erase step)
Endif
```

67.6.4.2 Conditions at driving

Activation: LV_PROG_STEP_1 = 0:

Recurrence: 10 ms

Calculation of VS_SP_DRIV_CRU_CAN:

Depending on the State of cruise controller:

```
If      LV_CRU_MAIN_SWI = 1
Then If LV_CRU_MAIN_SWI = 0 → 1        or      STATE_CRU = Passiv → active
    Then    VS_SP_DRIV_CRU_CAN = VS_SP_DRIV_CRU
    Else if STATE_CRU = "SET_ACC / RETARD"
        Then if     REQ_MSW = ACC active
            Then  if    VS_CRU ≥ VS_SP_DRIV_CRU_CAN
                  Then  VS_SP_DRIV_CRU_CAN = MIN(VS_CRU;
                        VS_SP_CRU_MAX)
                  Endif
            Else  if    VS_CRU ≤ VS_SP_DRIV_CRU_CAN        // DEC active
                  Then  VS_SP_DRIV_CRU_CAN = MAX(VS_CRU;
                        VS_MIN_CRU)
                  Endif
            Endif
        Else  VS_SP_DRIV_CRU_CAN = VS_SP_DRIV_CRU
        Endif
    Endif
Endif
```

Recurrence: 10 ms:

Calculation of vehicle speed steps:

(REQ_MSW = RES and LV_PROG_STEP_IF_ACT== 0 and VS_CRU ≥ VS_MIN_CRU)
for T > C_T_MIN_PROG

Activation: LV_STEP_ON = 1 AND
LV_CRU_MAIN_SWI = 1 AND
NOT (LV_CRU_ACT = 0 -> 1)

Set:

```
IF VS_SP_DRIV_CRU_CAN >< VS_STEP_m ;$( m = \sum_{k}^{0} )$ ± VS_DIF_STEP_MIN

Then    IF      COUNT_VS_STEP < C_MAX_STEP
        Then    VS_STEP_m; m=k+1= VS_SP_DRIV_CRU_CAN
                        For index m : $VS\_STEP\_m_{(n-1)}$ = 0 km.p.h.
        Else
                repeat
                ignore of request, " number of steps are full"
                " flashlight message triggered for "
                LV_STEP_ON = 0 for C_T_CRU_FLASH_LIGHT  then
                LV_STEP_ON = 1 for C_T_CRU_FLASH_LIGHT  then
                CTR_CRU_FLASH_LIGHT + 1
                until   CTR_CRU_FLASH_LIGHT = C_CTR_CRU_FLASH_LIGHT
                        reset counter CTR_CRU_FLASH_LIGHT to 0

Else IF VS_SP_DRIV_CRU_CAN >< VS_STEP_m ;$( m = \sum_{k}^{0} )$ ± (VS_DIF_STEP_MIN -
                                                        C_VS_DIF_STEP_MIN_HYS)

    Then        erease: --------------------------------
                VS_STEP_m; m=k  = 0 km.p.h (errase step)
                --------------------------------
    else        repeat
                ignore of request, " flashlight message triggered for "
                LV_STEP_ON = 0 for C_T_CRU_FLASH_LIGHT  then
                LV_STEP_ON = 1 for C_T_CRU_FLASH_LIGHT  then
                CTR_CRU_FLASH_LIGHT + 1
                until   CTR_CRU_FLASH_LIGHT = C_CTR_CRU_FLASH_LIGHT
                        reset counter CTR_CRU_FLASH_LIGHT to 0
    endif
Endif
```

REQ_MSW = MARK_UP:

```
If    LV_CRU_MAIN_SWI = 0 → 1        or
      LV_CRU_ACT = 0 → 1
Then        lock key permanent
            If      LV_STEP_ON = 0     and
```

```
                        Flashlight is not active
                Then    LV_STEP_ON = 1
                Endif
  Else          lock key cyclic
        IF      LV_CRU_ACT = 1
        Then IF         COUNT_VS_STEP = 0
                Then    If      LV_STEP_ON = 0     and
                                Flashlight is not active
                        Then    LV_STEP_ON = 1
                        Endif
                        VS_SP_DRIV_STEP = min((VS_SP_DRIV_CRU_CAN +
                                                        VS_DIF_STEP_PROG);
                                                        VS_SP_CRU_MAX)
                        LV_STEP_ON_ACT = 1
                Else    If      LV_STEP_ON = 0     and
                                Flashlight is not active
                        Then    LV_STEP_ON = 1
                                Lock key permanent
                        Else    IF      VS_SP_DRIV_CRU_CAN < max (VS_STEP_m)
                                Then    VS_SP_DRIV_STEP = VS_STEP_m  "next higher step"
                     [for m =  $m=\sum_{k}^{0}$  ; min(abs(VS_SP_DRIV_CRU_CAN- VS_STEP_m)) and
                                        VS_SP_DRIV_CRU_CAN < VS_STEP_m]
                                LV_STEP_ON_ACT = 1
                                Else no reaction        LV_STEP_ON_ACT = 0
                                Endif
                        Endif
                Endif
        Else no reaction
        Endif
Endif
```

REQ_MSW = MARK_DOWN:

```
  If    LV_CRU_MAIN_SWI = 0 → 1          or
```

```
     LV_CRU_ACT = 0 → 1
Then         lock key permanent
             If      LV_STEP_ON = 0     and
                     Flashlight is not active
             Then    LV_STEP_ON = 1
             Endif
Else         lock key cyclic
     IF      LV_CRU_ACT = 1
     Then    IF      COUNT_VS_STEP = 0
             Then    If      LV_STEP_ON = 0     and
                             Flashlight is not active
                     Then    LV_STEP_ON = 1
                     Endif
                     VS_SP_DRIV_STEP = max((VS_SP_DRIV_CRU_CAN -
                                             VS_DIF_STEP_PROG); VS_MIN_CRU)
                     LV_STEP_ON_ACT = 1
             Else    If      LV_STEP_ON = 0     and
                             Flashlight is not active
                     Then    LV_STEP_ON = 1
                             Lock key permanent
                     Else    IF      VS_SP_DRIV_CRU_CAN > min (VS_STEP_m; m= 0 to k)
                             Then    VS_SP_DRIV_STEP = VS_STEP_m  "next lower step"
                [for m = $m=\sum_{k}^{0}$ ; min(abs(VS_SP_DRIV_CRU_CAN - VS_STEP_m)) and
                                     VS_SP_DRIV_CRU_CAN > VS_STEP_m]
                             LV_STEP_ON_ACT = 1
                             Else IF VS_SP_DRIV_CRU_CAN > VS_MIN_CRU
                                     Then     VS_SP_DRIV_STEP = VS_MIN_CRU
                                             LV_STEP_ON_ACT = 1
                                     else    no reaction     LV_STEP_ON_ACT = 0
                                     Endif
                             Endif
                     Endif
                Endif
           Else no reaction
           Endif
```

```
Endif
"regards phycial convertion of veh. Speeds"
```

reset of LV_STEP_ON during drive:

```
 IF LV_CRU_MAIN_SWI = 0 and LV_PROG_STEP_1 = 0
Then LV_STEP_ON = 0; LV_STEP_ON_ACT = 0; stop flashlight
Endif
```

reset of LV_STEP_ON_ACT during drive:

```
If    LV_STEP_ON_ACT = 1
Then        CTR_STEP_ON_ACT ++
Else        CTR_STEP_ON_ACT = 0
Endif
If    CTR_STEP_ON_ACT > 100 ms
Then        LV_STEP_ON_ACT = 0
            CTR_STEP_ON_ACT = 0
Endif
```

setting of LV_CRU_DISP_HUD and LV_STEP_ON_ICL :

```
If    LC_CRU_DISP_SWI  =  1
Then        If      CTR_T_REQ_MSW_CON_MSG >C_T_CRUS_OFF or
                    LV_CRU_OFF_IRR  =  1
            Then    LV_CRU_DISP_HUD=  0
                    LV_STEP_ON_ICL  =  0
            ElseIf  LV_CRU_ACT  =  1   or   LV_PROG_STEP_1  =  1  or  ( REQ_MSW  >  1H
and
                    REQ_MSW != 7H )
            Then    LV_CRU_DISP_HUD  =  1
                    LV_STEP_ON_ICL  =  LV_STEP_ON
            Endif
Else
            LV_STEP_ON_ICL  =  LV_STEP_ON
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_DIF_STEP_PROG_K	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
km.p.h step calibration					
C_VS_DIF_STEP_PROG_M	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
m.p.h step calibation					
C_VS_DIF_STEP_MIN_M	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Minimum between steps in case of mile ICL					
C_VS_DIF_STEP_MIN_K	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Minimum between steps in case of Km ICL					
C_VS_DIF_TIP_PROG_M	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Difference between tip steps in case of mile ICL					
C_VS_DIF_TIP_PROG_K	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Difference between tip steps of Km ICL					
C_VS_MIN_PROG	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
VS_CRU threshold stand programming detection					
C_MAX_STEP	1	0...FFH	0...255	1	[-]
Maximum number of steps					
C_T_DLY_CRU_TIP	1	0...FFH	0...2.55	0.01	[s]
Time delay for display vehicle speed pointer					
C_T_CRU_FLASH_LIGHT	1	0...FFH	0...2.55	0.01	[s]
Flash light duration time					
C_T_MIN_PROG	1	0...FFH	0...5.1	0.02	[s]
Time for programming detection					
C_CTR_CRU_FLASH_LIGHT	1	1...FFH	1...255	1	[-]
Counter flash lights					
C_T_MIN_CRU_CHG	1	0...FFH	0...2.55	0.01	[s]
Look time for events					
C_VS_DIF_STEP_MIN_HYS	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Minimum between steps hysteresis					
C_T_PROG_STEP_IF_ACT	1	0...FFH	0...2.55	0.01	[s]
Time out for programming mode					
LC_CRU_DISP_SWI	1	0...1H	0...1	1	[-]
Configuration switch to switch off CRU display for HUD and ICL due to driver request					

67.6.5 Construction and function of the cruise control

67.6.5.1 Terminology

CRU manipulated variable:
FAC_TQ_CRU

Reference input variable:
VS_SP_CRU

Target variable:
VS_SP_DRIV_CRU

Control deviation:
VS_DIF_CRU = VS_SP_CRU - VS

Reference input/target variable difference:
VS_SP_DIF_CRU = VS_SP_CRU - VS_SP_DRIV_CRU

67.6.5.2 Conditions for cruise control

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_CRU	V	0H 1H 3H 5H 7H 9H	PASSIVE CONST_DRIVE RESUME SET_ACC RETARD TIP	1	[-]
Status of current cruise control functionality					
STATE_CRU_CAN	O/V	1H 2H 3H 4H 7H	CONST_TIP - RES - DEC	1	[-]
Status of cruise control to the CAN- Message					
REQ_CRU	V	0H 1H 2H 3H 4H 5H 6H 7H	HARD_OFF SOFT_OFF TIP_UP TIP_DOWN SET_ACCELERATE RESUME DECELERATE NONE	1	[-]
Request of cruise control functionality					
LV_CRU_OFF_IRR	O/V	0...1H	0...1	1	[-]
Irreversible cut off conditions for cruise control					
LV_CRU_ACT	O/V	0...1H	0...1	1	[-]
Intervention of cruise control					
STATE_CRU_OFF_REV	O/V/S	0...FFFFH	0...65535	1	[-]
Status switch off condition reversible					
STATE_CRU_OFF_IRR	O/V/S	0...FFFFH	0...65535	1	[-]
Status switch off condition irreversible					
T_DLY_CRU	V	0...FFFFH	0...655.35	0.01	[s]
Time counter overtaking					
LV_CRU_OVER_ACT	V	0...1H	0...1	1	[-]
Overtaking function active					
VS_MIN_CRU	O/V	0...FFFFH	0...511.99218	0.0078125	[km/h]
Minimal Vehicle speed condition for VS_CRU					
LV_CRU_ACT_INH	O/V	0...1H	0...1	1	[-]
Intervention of cruise control inhibited due to environmental conditions					
LV_CRU_ACT_VLD	V	0...1H	0...1	1	[-]
Indicating if driver wanted to activate CRU in DC at least once (BN2000), in 11hex always =1					

Input data:

LV_ERR_VS	LV_VAR_AMT	LV_ERR_VS_CAN	LV_ERR_MSW_TOG
LV_ERR_MSW_2	LV_ERR_MSW_3	LV_ERR_BLS_PLAUS	LV_ERR_TOUT_ASR_1
LV_ERR_TOUT_ASR_3	LV_BRAKE_DET	LV_ERR_CS	LV_N_LIM_REQ_MON
LV_ERR_TQI_N_MAX_MON	LV_VS_DIAG_NOT_PLAUS	IP_FAC_TQ_REQ_CLU_IS__N_32	LV_CRU_OFF_BY_ASR_ESP_CTL
C_FAC_TQ_REQ_CLU_IS_HYS	T_ERR_VS_CAN	LV_TQI_REQ_CAN_INH	LV_ERR_TOUT_AMT_1
LV_VS_MAX_VSL	LV_IM_CS_PN	LV_ERR_CRU_INH_MON	LV_N_MAX
REQ_MSW	LV_CRU_MAIN_SWI	LV_ERR_TOUT_ETCU_1	LV_ES
GEAR_INFO	VS_SP_DRIV_CRU	LV_ERR_BN_TQ_TCS	VS_CRU
VS_SP_CRU	STATE_IF_ICL_BN_MSW	VS_SP_CRU_MAX	FAC_TQ_REQ_DRIV

FAC_TQ_REQ_CRU	AC_CRU	LV_ERR_BN_VS_TCS	STATE_TCS_DECE
VS_DIF_CRU	LV_ERR_PVS	LV_ERR_BN_ICL	LV_ERR_CAN_BOFF
LV_ERR_BN_TCS	LV_ERR_BN_ETCU	LV_ERR_BN_TQ_ETCU	LV_ERR_BN_MSW
LV_ACT_CRU_MEM_EXT_ADJ	LV_VAR_DCC	LV_VAR_BN_LDM	LV_CRU_OVER_ACT_ACK
LV_ERR_BN_TQ_AMT	LV_ERR_CRIT_OVL_ECU_VVL	LV_ERR_DR_SC_VVL	LV_ERR_OVL_ECU_VVL
LV_ERR_RLY_VVL	LV_ERR_VVL_ROT	LV_ERR_CUR_H_VVL	STATE_VS_ICL_DISP
LV_VAR_BN	LV_AT	N_VS_CRU	LV_ERR_TOUT_ETCU_2
LV_VAR_TCT	STATE_BN_MSW	STATE_DHL_CTL	LV_VS_MAX
LC_N_MAX_VS_MAX_ACT			

FUNCTION DESCRIPTION:

The State of cruise control is the evaluation of the request and the cruise conditions.
If one of the following conditions is detected, no controller can be activated or an activated controller function is switched off. In this case a distinction is made between hard (FAC_TQ_CRU = 0) and soft cut-off (FAC_TQ_CRU through ramp function to zero).

Application conditions:

Application conditions:

Initialisation: *STATE_CRU_OFF_IRR, STATE_CRU_OFF_REV out of NVMY*

LV_CRU_ACT_VLD = 0 at LV_IGK 0->1 ***or*** *at reset*

all other outputs = 0

Recurrence: 10 ms

Activation: LV_VAR_DCC = 0 and LV_VAR_BN_LDM = 0

Deactivation: LV_VAR_DCC = 1 or LV_VAR_BN_LDM = 1 then

Initialisation: STATE_CRU_OFF_IRR, STATE_CRU_OFF_REV =00H

Formula section:

STATE_CRU_CAN	STATE_CRU
0. None:	PASSIVE
1. CONST	CONST_DRIVE or TIP
2. not used	
3. RES:	RES
4. not used	
5. SET/ACC	SET/ACC
6. not used	
7. DEC	DEC

If FAC_TQ_REQ_DRIV > FAC_TQ_REQ_CRU and STATE_CRU not "passive"
Then STATE_CRU_CAN = "None"
Endif

67.6.5.2.1 Conditions for hard cut-off

The control functions of the cruise control can be activated only if:

- CRU ready state present LV_CRU_MAIN_SWI = 1

<u>Irreversible cut-off due to diagnosis failures</u>

With the following diagnosis errors the CRU is irreversible hard cut off for this engine run. The CRU-readiness and the CRU readiness light are reseted. The CRU function can be activated by LV_IGK=0/1 change, as long as no running diagnosis errors are present. A distinction is done between the irreversible failures because of an error output for CAN message.

Formula section:

```
If      STATE_VS_ICL_DISP = 1          "km/h; mph"
Then    VS_MIN_CRU = C_VS_MIN_CRU_M
Else    VS_MIN_CRU = C_VS_MIN_CRU_K
Endif
```

// Recognition if cruise control is really configuration of car (in Case of E90 LV_VAR_BN = 1 instead of no build in cruise control)

```
IF      LV_VAR_BN = 1
THEN    IF      STATE_BN_MSW != No_action (0h)        and
                STATE_BN_MSW != Invalid_signal (7h)
        THEN    //set bit for recognising desired usage of cruise control by driver for first
                time in driving cycle:
                LV_CRU_ACT_VLD = 1
        ENDIF
ELSE    in case of 11hex bit is always set:
        LV_CRU_ACT_VLD = 1
ENDIF
```

```
If    [7] LV_ERR_BLS_PLAUS = 1                or    /* Error in break light system */
      [12] LV_ERR_VS = 1                      or    /* Error in speed signal (sensor) */
      [11] LV_VS_DIAG_NOT_PLAUS = 1           or    /* Implausible speed signal VS_CRU */
      [10] LV_ERR_TQI_N_MAX_MON =1            or    /* Error Level 2 Monitoring */
      [10] LV_ERR_CRU_INH_MON =1              or    /* Error Level 2 Monitoring */
      [10] LV_N_LIM_REQ_MON =1                or    /* Error Level 2 Monitoring */
      [3] LV_ERR_CS = 1                       or    /* Error in clutch switch */
      [4] LV_ERR_MSW_2 = 1                    or    /* Multi-function steering wheel Error */
      [4] LV_ERR_MSW_3 = 1                    or    /* Multi-function steering wheel Error */
      [4] LV_ERR_MSW_TOG = 1                  or    /* Multi-function steering wheel Error */
      [4] LV_ERR_BN_MSW = 1                   or    /* Multi-function steering wheel Error */
      [9] LV_ERR_BN_ICL = 1                   or    /* CAN bus fault ICL */
      [8] LV_ERR_CAN_BOFF = 1                         or    /* CAN bus off */
      [13] LV_ERR_TOUT_ASR_1 = 1              or    /* CAN bus fault ASR */
      [13] LV_ERR_TOUT_ASR_3 = 1              or    /* CAN bus fault ASR */
      [14] LV_ERR_BN_TCS = 1                  or    /* CAN bus fault TCS */
      [14] LV_ERR_BN_TQ_TCS = 1               or    /* CAN bus fault TCS */
      [14] LV_ERR_BN_VS_TCS =1                or    /* CAN bus fault TCS */
      [5] LV_ERR_TOUT_AMT_1 = 1               or    /* CAN bus fault AMT */
      [5] LV_ERR_BN_TQ_AMT = 1                or    /* CAN bus fault AMT */
      [6] LV_ERR_TOUT_ETCU_1 = 1              or    /* CAN bus fault ETCU */
       [6] LV_ERR_TOUT_ETCU_2 = 1             or    /* CAN bus fault ETCU */
      [6] LV_ERR_BN_ETCU = 1                  or    /* CAN bus fault ETCU */
      [6] LV_ERR_BN_TQ_ETCU = 1               or    /* CAN bus fault ETCU */
      [2] LV_ERR_PVS = 1                      or    /* PVS-Error */
      [1] LV_ERR_CRIT_OVL_ECU_VVL = 1         or    /* critical VVL driver overload  */
      [1] LV_ERR_DR_SC_VVL = 1                or    /* short circuit on VVL driver */
      [1] LV_ERR_OVL_ECU_VVL = 1              or    /* VVL driver overload */
      [1] LV_ERR_RLY_VVL = 1                  or    /* VVL relay error */
      [1] LV_ERR_VVL_ROT = 1                  or    /* VVL rotation sense error */
      [1] LV_ERR_CUR_H_VVL = 1                      /* VVL motor current critical high */
Then          IF     LV_CRU_ACT_VLD = 1
              THEN   LV_CRU_OFF_IRR = 1
              ELSE   LV_CRU_OFF_IRR = 0
              ENDIF
              REQ_CRU = Hard off
```

(The cru readiness lamp is switched off.)

Endif

STATE_CRU_OFF_IRR :

High Byte								Low Byte							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
		13	12			11	10	9	8	7	5	4	3	2	1
		14									6				
------------------------	------------------------	ASR/TCS-Error	Error VS	------------------------	------------------------	VS plausibility	Monitoring level 2	error instrument message	CAN Bus off	Brake error	AMT/ETCU-Error	MSW error	Error clutch switch	PVS-Error	VVL-Error

External adjustment:

```
If      LV_ACT_CRU_MEM_EXT_ADJ = 1
Then    STATE_CRU_OFF_IRR  = 00H
        STATE_CRU_OFF_REV  = 00H
Endif


If      LC_N_MAX_VS_MAX_ACT = 1
Then    LV_VS_MAX_VSL_1   = LV_VS_MAX
Else    LV_VS_MAX_VSL_1   = LV_VS_MAX_VSL
Endif
```

If one of the following conditions is current in active CRU function, CRU is transferred by hard cut-off (FAC_TQ_CRU = 0) into the ready state without controlled drive:

```
If      LV_CRU_MAIN_SWI = 0                               or    /* main switch off */
[1]     VS_CRU <  C_VS_MIN_PLAUS_CRU                      or    /* speed under min. threshold  */
        LV_ES = 1                                         or    /* Engine stop */
[2]     LV_BRAKE_DET = 1                                  or    /* Brake switch */
[3]     LV_N_MAX = 1                                      or    /* active engine speed limitation */
[4]     (LV_AT = 0 and LV_VAR_AMT = 0 and LV_VAR_TCT = 0 and
        |(N_VS_CRU$_{n}$  -  N_VS_CRU$_{n-1}$) / N_VS_CRU$_{n-1}$| * 100  >  C_N_VS_MAX_CRU)
                                                          or    /* implausible N/VS-ratio with MT*/
[5]     (LV_CRU_OVER_ACT = 0 and |AC_CRU|  >  C_AC_MAX_CRU)

                                                          or    /* maximum acceleration exceeded
                                                                (not during overtaking function) */
        (LV_AT = 0 and LV_VAR_AMT = 0 and LV_VAR_TCT = 0) and
        LV_IM_CS_PN = 1)                                  or    /* clutch actuation with MT */
[6]     (LV_ERR_VS_CAN = 1 and
        T_ERR_VS_CAN = 0)                                 or    /* Error on CAN signal VS_CAN */
        ((LV_AT = 1  or  LV_VAR_AMT = 1 or LV_VAR_TCT = 1) and
        GEAR_INFO = 0 or AH)                              or/*illegal gear info with AT or AMT/TCT*/
[7]     REQ_MSW = "HARD_OFF"                              or    /* MSW-request */
[8]     (LV_CRU_OVER_ACT = 1 and LV_VS_MAX_VSL_1 = 1)

                                                          or

        (LV_VAR_BN = 1 and
        (STATE_TCS_DECE = 02H or
        STATE_IF_ICL_BN_MSW = 001))                       or
[15]     LV_TQI_REQ_CAN_INH = 1                           or    /* Error in TPS/ETC */
[16]    (STATE_DHL_CTL > 0H and STATE_DHL_CTL < FH)     /* hill descent control active */


Then    REQ_CRU = Hard off
Endif
```

67.6.5.2.2 Conditions for soft cut-off:

General information:

If one of the following conditions is detected, the controller function is switched off. In this FAC_TQ_CRU is set through ramp function to zero.

Formula section:

```
If
[9]     REQ_MSW = SOFT_OFF                                  or    /* MSW-request */
[10]    [LV_CRU_OVER_ACT = 0 and
        (VS_DIF_CRU > C_VS_SP_DIF_MAX_CRU or
        VS_DIF_CRU < C_VS_SP_DIF_MIN_CRU)]
                                                            or    /* control deviation too large */
[11]    [LV_CRU_OVER_ACT = 1 and
        ((T_DLY_CRU ≥ C_T_DLY_MAX_CRU) or
        (VS_SP_DRIV_CRU - VS_CRU ≥ C_VS_DIF_DLY_CRU_POS) or
        (VS_SP_DRIV_CRU - VS_CRU < C_VS_DIF_DLY_CRU_NEG))]
                                                            or
[12]    (VS_CRU ≥ VS_SP_CRU_MAX for longer than C_T_DLY_VS_SP_CRU_MAX)
        Remark: if this condition is interupted than time-counter is reset to 0
                                                            or
[13]    ASR- or ESP-control for minimum time active
        (LV_CRU_OFF_BY_ASR_ESP_CTL = 1)

Then     REQ_CRU = Soft off
Endif
```

STATE_CRU_OFF_REV :

High Byte								Low Byte							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
7	2	3	1	8	4	5	6		16	15	9	13	12	11	10
MSW request HARD_OFF	Brake detection	Engine speed limitation	VS_CRU to low	Overtaking and VS_MAX active	Runup lock	Acceleration monitoring	Error VS_CAN to long	------------------	Hill descent control (HDC) active	TPS/ETC-Error	MSW request "off"	External TQ intervention	VS_SP_MAX to long	Overtaking Funct. To long or to high VS_DIF	To high VS deviation

67.6.5.2.3 Overtaking function

Description:

If the throttle wish FAC_TQ_REQ_DRIV > FAC_TQ_REQ_CRU and ≥ IP_FAC_TQ_REQ_CLU_IS__N_32 , then the controlled detectes overtaking and the value VS_DIF_I_CRU remains stored as threshold value. The time for the overtaking function T_DLY_CRU is started. The adjustable time C_T_DLY_MAX_CRU states whether starting controlled constant drive is possible after ending the overtaking function.
During the overtaking function, 'Overdriving' FAC_TQ_REQ_DRIV by the acceleration (ACC) and Tip-up functions is possible.
If the time C_T_DLY_MAX_CRU or the max. Deviation from the stored target speed is exceeded, no automatic start of controlled constant drive is possible any more. The CRU mode is ended. The functions of the cruise control can be activated again exclusively by the multifunction steering wheel. At resume the after overtaking function is finished the resume function is activated with the initilisation calculation of VS_DIF_I_CRU.
During the overtaking function the calculation of FAC_TQ_CRU is calculated normal with freezed I - share and calculated P-Share.

Application conditions:

Initialisation: LV_CRU_OVER_ACT = 0

Formula section:

```
If    LV_CRU_OVER_ACT = 0
Then        If      (STATE_CRU = constant drive or tip)            and
                    REQ_MSW != TIP_UP                              and
                    REQ_MSW != SET_ACC                             and
                    (REQ_MSW != TIP_DOWN           and
                    REQ_MSW != DECE                             or
                    LV_VAR_BN = 0)                                 and
                    FAC_TQ_REQ_DRIV  >  FAC_TQ_REQ_CRU             and
                    FAC_TQ_REQ_DRIV >  IP_FAC_TQ_REQ_CLU_IS__N_32
            Then    $T\_DLY\_CRU_{n} = T\_DLY\_CRU_{n-1} + 10$ ms
                    LV_CRU_OVER_ACT = 1
                    // "VS_DIF_I_CRU stored see calculation of I-Share"
            Endif
Endif

If    LV_CRU_OVER_ACT = 1
Then        If (FAC_TQ_REQ_DRIV  <  FAC_TQ_REQ_CRU  -C_FAC_TQ_REQ_HYS_CRU
                                        or
                    FAC_TQ_REQ_DRIV < IP_FAC_TQ_REQ_CLU_IS__N_32 -
                    C_FAC_TQ_REQ_CLU_IS_HYS)
                                        and
                    VS_DIF_CRU > C_VS_DIF_MIN_CRU_OVER
            Then    LV_CRU_OVER_ACT = 0
            Else    If      REQ_MSW = TIP_UP                    or
                            (REQ_MSW = TIP_DOWN   and
                            LV_VAR_BN = 1)
                    Then If         LV_CRU_OVER_ACT_ACK = = 1
                            Then    LV_CRU_OVER_ACT = 0
                            Endif
                            REQ_CRU = REQ_MSW
                    Endif
            Endif
Endif
```

```
If      LV_CRU_ACT = 0 → 1        and     VS_CRU ≤ VS_SP_CRU_MAX
Then    If      LV_VAR_BN = 1               and
                (REQ_MSW = TIP_DOWN   or
                REQ_MSW = DECE              or
                REQ_MSW = MARK_UP        or
                REQ_MSW = MARK_DOWN)
        Then    REQ_CRU = SET_ACC
        Endif
Endif
```

```
If   LV_CRU_ACT = 1     and     LV_CRU_OVER_ACT = 0
Then    If      REQ_MSW = MARK_UP      or      REQ_MSW = MARK_DOWN
        Then    REQ_CRU = NONE
        Else    REQ_CRU = REQ_MSW
        Endif
Endif
```

67.6.5.2.4 Condition for cruise control intervention

Description:

The cruise control intervention is active if the vehicle is in operation depending of FAC_TQ_CRU. The Status is needed in the Cruise control monitoring of ETC Safety Unit.
If an MSW request is active but LV_CRU_ACT stays 0 due to environmental conditions (VS_CRU < VS_MIN_CRU, "Soft off", "Hard off" etc.) LV_CRU_ACT_INH is set.

Formula section:

```
If   LV_CRU_ACT = 0                                              and
     (LV_MSW_IDLE = 1            or
     (LV_CRU_MAIN_SWI = 0 → 1        and
     LV_VAR_BN = 1))                                             and
     VS_CRU >= VS_MIN_CRU                                        and
     ((REQ_MSW = TIP_UP  or  SET/ACC)        or
     (LV_VAR_BN = 1                and
     (REQ_MSW = TIP_DOWN or DECE or MARK_UP or MARK_DOWN)) or
      (REQ_MSW = RES     and
       Triggering the RES function for plausible VS_SP_DRIV_CRU
     (VS_MIN_CRU <= VS_SP_DRIV_CRU <= VS_SP_CRU_MAX)))
```

```
Then     LV_CRU_ACT = 1
Endif
```

LV_CRU_ACT is reseted to 0 if an off condition is present.

```
If       LV_CRU_ACT = 1   and   REQ_CRU =  "Soft off" or "Hard off"
Then     LV_CRU_ACT = 0
Endif


If       LV_CRU_ACT = 0  and  REQ_MSW > 1H  and  REQ_MSW != 7H
         and LV_CRU_OFF_IRR = 0
Then     LV_CRU_ACT_INH = 1
Else     LV_CRU_ACT_INH = 0
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_SP_DIF_MAX_CRU	1	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Max. vehicle speed deviation					
C_VS_SP_DIF_MIN_CRU	1	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Min. vehicle speed deviation					
C_T_DLY_VS_SP_CRU_MAX	1	0...FFH	0...255	1	[s]
Max .time for VS more than VS max at CRU					
C_VS_MIN_CRU_K	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed condition for VS_CRU in case of Km - ICL					
C_VS_MIN_CRU_M	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed condition for VS_CRU in case of mile - ICL					
C_VS_MIN_PLAUS_CRU	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Vehicle speed condition for VS_CRU					
C_N_VS_MAX_CRU	1	0...FFH	0...99.60937	0.390625	[%]
Threshold value for hard cut-off					
C_AC_MAX_CRU	1	0...7FFFH	0...15.71952	0.4797e-3	[m/s²]
Max. acceleration					
C_FAC_TQ_REQ_HYS_CRU	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
FAC_TQ_REQ hysteresis for start of the restart function					
C_T_DLY_MAX_CRU	1	0...FFFFH	0...655.35	0.01	[s]
Delay threshold for start of the restart function					
C_VS_DIF_DLY_CRU_NEG	1	0...7FFFH	0...255.99218	0.0078125	[km/h]
Max. difference from VS_SP_DRIV_CRU and C_VS_DIF_DLY_CRU					
C_VS_DIF_DLY_CRU_POS	1	8000...0H	-256...0	0.0078125	[km/h]
Max. difference from VS_SP_DRIV_CRU and C_VS_DIF_DLY_CRU					
C_VS_DIF_MIN_CRU_OVER	1	8000...0H	-256...0	0.0078125	[km/h]
VS_DIF_CRU min value for finish overtaking function					

67.6.6 Control basic function

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VS_DIF_CRU	V/O	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Control deviation					
VS_DIF_I_CRU	V	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Sum of the control deviations					
FAC_TQ_CRU	V/O	0...FFFFH	0...99.99847	1.53E-03	[%]
CRU control variable					
FAC_TQ_CRU_INI	V	0...FFFFH	0...99.99847	1.53E-03	[%]
CRU control variable initialisation value					
STATE_I_CRU	V	1H 2H 3H	LIM FREEZE NORM	1	[-]
Condition of I Share VS_DIF_I_CRU					
T_CTL_CRU	V	0...FFH	0...25.5	0.1	[s]
Time of the controlled acceleration					
VS_SP_CRU	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Reference input variable of the controller					
VS_SP_DIF_CRU	V	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Target variable/reference input variable difference					
FAC_VS_CTL_CRU	V	0...FFH	0...1.99218	0.0078125	[-]
Correction factor for actuating the reference input variable					
FAC_VS_LGRD_CRU	V	0...FFH	0...0.99609	3.91E-03	[-]
Factor for intercepting the reference input variable					
VS_SP_INC_CRU	V	0...FFFFH	0...511.99218	0.0078125	[km/h]
Increment for VS_SP_CRU					
VS_SP_DEC_CRU	V	0...FFFFH	0...511.99218	0.0078125	[km/h]
Decrement for VS_SP_CRU					
VS_SP_DRIV_CRU	V	0...FFFFH	0...511.99218	0.0078125	[km/h]
Driver wish target variable					
LV_CRU_OVER_ACT_ACK	V/O	0...1H	0...1	1	[-]
Hand shake overtaking function					
VS_SP_CRU_MAX	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
maximum limitation of cruise control					

Input data:

VS_CRU	STATE_VS_ICL_DISP	AC_CRU	FAC_TQ_REQ_DRIV_MMV
N_32	REQ_CRU	IP_FAC_TQ_REQ_CLU_IS__N_32	C_FAC_TQ_REQ_CLU_IS_HYS
LV_AT	LV_RNG_L_REQ	FAC_TQ_REQ_DRIV	LV_CRU_OVER_ACT
LV_VAR_BN	LV_CRU_MAIN_SWI	LV_STEP_ON_ACT	LV_IGK
N	VS_SP_DRIV_STEP	STATE_CC	T_CTL_CRU
REQ_MSW	FAC_TQ_REQ_CRU	STATE_CRU	VS_MIN_CRU
GEAR			

67.6.6.1 General calculation of cruise control and constant drive

Application conditions:

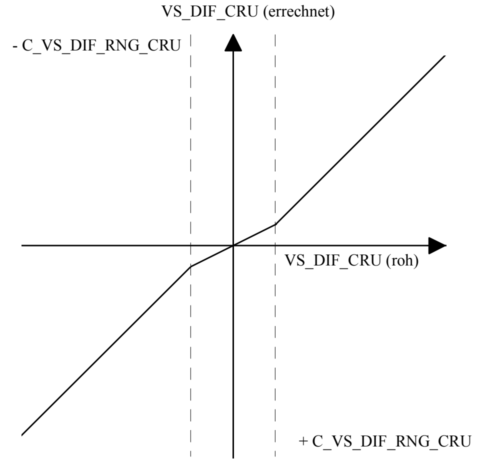
Recurrence: 100 ms

Initialisation: FAC_TQ_CRU = FAC_TQ_CRU_INI based on setting conditions

FUNCTION DESCRIPTION:

Description:

On setting in PU/PUC or with very small FAC_TQ_REQ_DRIV (driver gets off the pedal bevor setting Cruise Control), a very small I component setpoint (or 0) would be the results and it follows an undershooting below the required target speed. To prevent this, the I share set value is calculated depending on AC_CRU so that a minimum FAC_TQ_CRU results.



Formula section:

Basically calculation for all cruise states:

Total formula:

FAC_TQ_CRU = (VS_DIF_CRU + VS_DIF_I_CRU) * IP_FAC_PROP_CRU

Conditions for change limitations of FAC_TQ_CRU:

$|FAC\_TQ\_CRU_{n-1} - FAC\_TQ\_CRU_{n}| \leq C\_ FAC\_TQ\_CRU\_LGRD$

At driving by cruise control it should not be possible to request the maximum torque as well outside cruise control.

FAC_TQ_CRU <= C_ FAC_TQ_CRU_MAX

The non-linearized value of the control deviation from the reference input variable VS_SP_CRU and the control variable VS_CRU is calculated as

VS_DIF_CRU (rough)= VS_SP_CRU - VS_CRU

To achieve a quiet control in the range of the target variable, the internal control deviation VS_DIF_CRU can be reduced in the adjustable bandwidth | C_VS_DIF_RNG_CRU | by an applicable factor C_FAC_VS_DIF_RNG_CRU.

```
IF(1)            |VS_SP_CRU - VS_CRU | ≤ C_VS_DIF_RNG_CRU
Then(1)   VS_DIF_CRU = (VS_SP_CRU - VS_CRU ) * C_FAC_VS_DIF_RNG_CRU
Else(1) If(2)    (VS_SP_CRU - VS_CRU) > C_VS_DIF_RNG_CRU
    Then(2)  VS_DIF_CRU = (VS_SP_CRU - VS_CRU ) - C_VS_DIF_RNG_CRU
                        + C_VS_DIF_RNG_CRU * C_FAC_VS_DIF_RNG_CRU.
    Else(2)  VS_DIF_CRU = (VS_SP_CRU - VS_CRU )
                        + C_VS_DIF_RNG_CRU * C_FAC_VS_DIF_RNG_CRU.
    Endif (2)
Endif(1)
```

I share calculation:

Basically there are 3 states of STATE_I_CRU: NORM, LIM and FREEZE (for detailed information see formula section below)

If the FAC_TQ_CRU is on limitation (max, min value or freeze at PUC) then the I share is freezed **STATE_I_CRU = LIM** until a normal calculation of FAC_TQ_CRU is possible again. At constant driving or LV_CRU_OVER_ACT = 1 the I share is freezed **STATE_I_CRU = LIM** but in the calculation of FAC_TQ_CRU there is no interupt.

Calculation in case of controller active:

```
If      REQ_CRU = SOFT_OFF
Then    STATE_I_CRU = LIM
Else    If      [ STATE_CRU = DEC          AND
                (VS_DIF_I_CRU + C_VS_DIF_DELTA_I – IVS_DIF_CRUI >= 0)] OR
                [STATE_CRU = CONST_DRIVE     AND
                LV_CRU_OVER_ACT = 0     AND
                FAC_TQ_CRU > (IP_FAC_TQ_REQ_CLU_IS__N_32 –
                              C_FAC_TQ_REQ_CLU_IS_HYS) * 99,998%)
                              + C_FAC_TQ_CRU_OFS     AND
                 FAC_TQ_CRU < C_FAC_TQ_CRU_MAX]
        Then    // General calculation

                STATE_I_CRU = NORM
                VS_DIF_I_CRU $_{N}$ = VS_DIF_I_CRU $_{N-1}$ + VS_DIF_CRU *
                                                      IP_FAC_VS_DIF_I_CRU
                VS_SP_CRU$_{START}$ = VS_SP_CRU
                VS_DIF_I_CRU$_{START}$ = VS_DIF_I_CRU
        Else    If      STATE_CRU = SET_ACC    OR
                        STATE_CRU = RESUME
                Then    // Calculation during freeze of I share at SET/ACC, RES

                        STATE_I_CRU = FREEZE
                        VS_SP_CRU_STAT = VS_SP_CRU$_{n}$ - VS_SP_CRU$_{START}$
                        VS_DIF_I_CRU = VS_DIF_I_CRU$_{START}$ +
                                IP_FAC_VS_SP_DIF_I_CRU * VS_SP_CRU_STAT
                Else    // Calculation during limitation of I share

                        STATE_I_CRU = LIM
                        VS_SP_CRU$_{START}$ = VS_SP_CRU
                        VS_DIF_I_CRU$_{START}$ = VS_DIF_I_CRU
                Endif
        Endif
Endif
```

Calculation in case of controller passive:

Initialisation value calculated once on transition passive to active:

```
IF LV_AT = 0
Then    FAC_TQ_CRU_INI =    FAC_TQ_REQ_DRIV_MMV * 99,998 % +
        IP_FAC_TQ_CRU_INI_ADD_MT * AC_CRU
Else    FAC_TQ_CRU_INI =    FAC_TQ_REQ_DRIV_MMV * 99,998 % +
        IP_FAC_TQ_CRU_INI_ADD_AT * AC_CRU
endif
```

VS_DIF_I_CRU = FAC_TQ_CRU_INI / IP_FAC_PROP_CRU

$VS\_SP\_CRU_{START} = VS\_SP\_CRU$

$VS\_DIF\_I\_CRU_{START} = VS\_DIF\_I\_CRU$

Disable gradient limitation for FAC_TQ_CRU once

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_PROP_CRU	5*4	0...FFFFH	0...49.99923	7.63E-04	[%/(km/h)]
LDPM_VS_SP_CRU__1	5	0...FFFFH	0...511.99218	0.0078125	[km/h]
LDP_VS_DIF_CRU__FAC_PROP_CRU	4	0...FFFFH	-256...255.99218	0.0078125	[km/h]
Proportional coefficient of the PI controller					
IP_FAC_VS_DIF_I_CRU	5*4	0...FFFFH	0...0.99998	1.53E-05	[-]
LDPM_VS_SP_CRU__1	5	0...FFFFH	0...511.99218	0.0078125	[km/h]
LDP_VS_DIF_CRU__FAC_PROP_CRU	4	0...FFFFH	-256...255.99218	0.0078125	[km/h]
Proportional coefficient of the PI controller					
IP_FAC_TQ_CRU_INI_ADD_AT	6*12	0...FFFF	-200...199.99389	6.1035e-3	[%/(m/s²)]
LDP_AC_CRU__FAC_TQ_CRU	6	0...FFFF	-15.72...15.71952	0.4797e-3	[%]
LDP_GEAR__FAC_TQ_CRU	12	0...FF	0...255	1	[-]
Additive torque based on gear and acceleration					
C_FAC_TQ_CRU_LGRD	1	0...FFFFH	0...99.99847	1.53E-03	[%/100ms]
Control variable limitation gradient					
C_FAC_VS_DIF_RNG_CRU	1	0...FFH	0...0.99609	3.91E-03	[-]
Reduction factor for the control deviation VS_DIF_CRU					
C_VS_DIF_RNG_CRU	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Bandwidth for reducing the control deviation					
IP_FAC_VS_SP_DIF_I_CRU	5	0...FFFFH	0...1.99996	3.05E-05	[-]
LDPM_VS_SP_CRU__1	5	0...FFFFH	0...511.99218	0.0078125	[km/h]
Delta VS_SP_CRU correction					
C_FAC_TQ_CRU_MAX	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Maximum request of Cruise Control					
IP_FAC_TQ_CRU_INI_ADD_MT	6*12	0...FFFF	-200...199.99389	6.1035e-3	[%/(m/s²)]
LDP_AC_CRU__FAC_TQ_CRU	6	0...FFFF	-15.72...15.71952	0.4797e-3	[%]
LDP_GEAR__FAC_TQ_CRU	12	0...FF	0...255	1	[-]
Additive torque based on gear and acceleration					

67.6.6.1.1 Controlled constant drive (CONST_DRIVE)

For controlled constant drive, the reference input variable VS_SP_CRU corresponds to the last target speed value reached, i.e. VS_SP_DRIV_CRU. No control of the reference input variable is performed in this range.

With the basic function of control, the vehicle speed is controlled corresponding to the required reference input variable VS_SP_CRU. This occurs until a cut-off condition arises or a condition for cruise control is infringed. The resume function is locked in the controlled constant drive. On recognition of other CRU functions, the reference input variable is controlled. To be able to trigger trailing throttle cut-off expediently, the value FAC_TQ_CRU is initially limited downwards to the value ((IP_FAC_TQ_REQ_CLU_IS__N_32 - C_FAC_TQ_REQ_CLU_IS_HYS)*99,998%). Thus there can be no IS output. Only if the control deviation VS_DIF_CRU drops below the applicable value C_VS_DIF_MIN_IS_CRU is FAC_TQ_CRU set equal to zero. However, this zero setting takes place at the earliest C_T_MIN_PUC_CRU seconds after entry into controlled constant drive. The trailing throttle cut-off is retained until the control deviation exceeds the applicable value C_VS_DIF_MAX_IS_CRU.

Formula section:

FAC_TQ_CRU conditions:

```
IF   T_CONST_DRIVE < C_T_MIN_PUC_CRU
Then      If     FAC_TQ_CRU < ((IP_FAC_TQ_REQ_CLU_IS__N_32-
                          C_FAC_TQ_REQ_CLU_IS_HYS) * 99,998%)
                          + C_FAC_TQ_CRU_OFS

          then   FAC_TQ_CRU = ((IP_FAC_TQ_REQ_CLU_IS__N_32-
                          C_FAC_TQ_REQ_CLU_IS_HYS) * 99,998%)
                          + C_FAC_TQ_CRU_OFS
          endif
ElseIF    VS_DIF_CRU < C_VS_DIF_MIN_IS_CRU &&
          FAC_TQ_CRU < ((IP_FAC_TQ_REQ_CLU_IS__N_32-
                          C_FAC_TQ_REQ_CLU_IS_HYS) * 99,998%)
                          + C_FAC_TQ_CRU_OFS
    Then  FAC_TQ_CRU = 0
    Else  IF     VS_DIF_CRU > C_VS_DIF_MAX_IS_CRU
          Then   If     FAC_TQ_CRU < ((IP_FAC_TQ_REQ_CLU_IS__N_32-
                          C_FAC_TQ_REQ_CLU_IS_HYS) * 99,998%)
                          + C_FAC_TQ_CRU_OFS

                 Then   FAC_TQ_CRU = ((IP_FAC_TQ_REQ_CLU_IS__N_32-
                          C_FAC_TQ_REQ_CLU_IS_HYS) * 99,998%)
                          + C_FAC_TQ_CRU_OFS
                 Endif
          Else   calculation FAC_TQ_CRU_n = calculation FAC_TQ_CRU_n-1
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_DIF_MIN_IS_CRU	1	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Threshold of the control deviation for zero setting of FAC_TQ_CRU					
C_VS_DIF_MAX_IS_CRU	1	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Threshold of the control deviation for PUC end					
C_T_MIN_PUC_CRU	1	0...FFH	0...25.5	0.1	[s]
Time after entry into controlled constant drive					
C_FAC_TQ_CRU_OFS	1	0...FFFFH	0...99.99847	1.53E-03	[%]
Offset for cruise control variable					

67.6.6.1.2 Cut-off of the CRU function

General information:

Soft cut-off

In the soft cut-off of the cruise control, the cruise control wish FAC_TQ_REQ_CRU is reduced to the driver wish FAC_TQ_REQ_DRIV with the adjustable decrement IP_FAC_TQ_CRU_DEC __N_32 for FAC_TQ_REQ_DRIV < FAC_TQ_REQ_CRU until the pedal driver demand is reached. FAC_TQ_CRU is then set to zero.

Hard cut-off

In the hard cut-off mode of the cruise control, the charge wish FAC_TQ_CRU is set equal to zero. There is no limitation gradient. The CRU State is readiness.

Application conditions:

Activation: see Capter:Cut -off Conditions and switch- on prohibitions

Formula section:

Soft cut-off

For FAC_TQ_REQ_DRIV < FAC_TQ_REQ_CRU there results:

FAC_TQ_CRU = FAC_TQ_CRU $_{n-1}$ - IP_FAC_TQ_CRU_DEC __N_32 until

FAC_TQ_REQ_DRIV = FAC_TQ_REQ_CRU

Hard cut-off

FAC_TQ_CRU = 0

Then, the CRU ready status is reached. Only now can the CRU mode be activated again.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_TQ_CRU_DEC__N_32	5	0...FFFFH	0...99.99847	1.53E-03	[%]
LDP_N_32__FAC_TQ_CRU	5	0...FFH	0...8160	32	[rpm]
Decrement of the FAC_TQ_CRU					

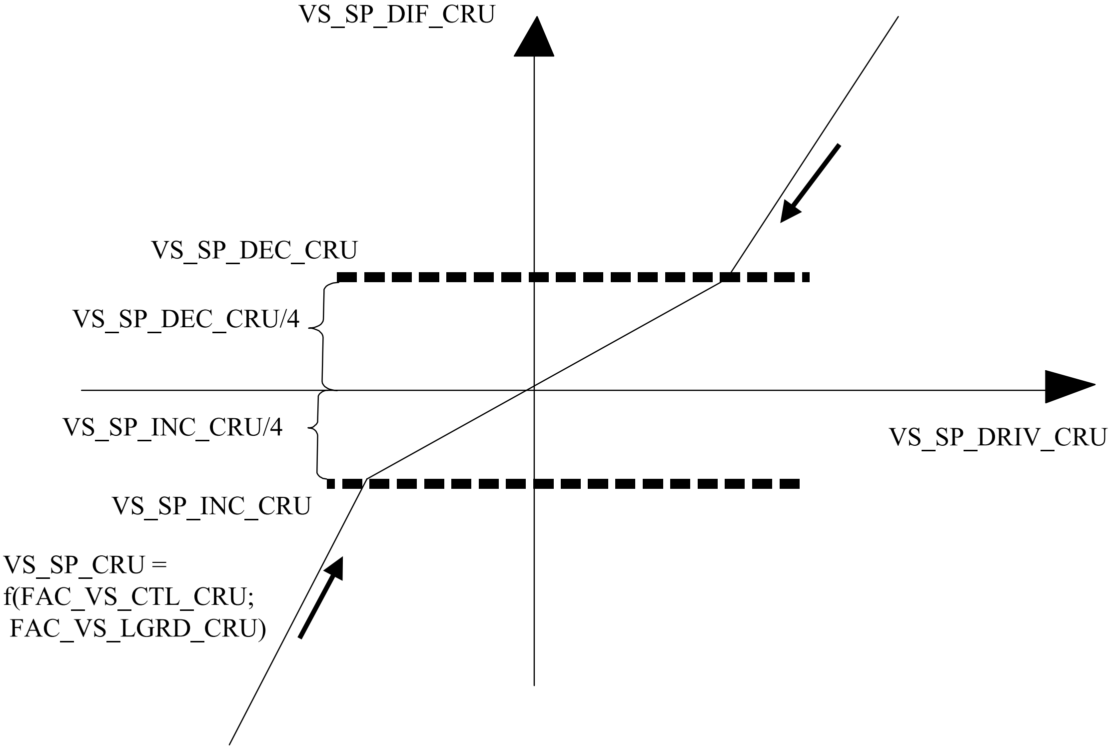
67.6.6.2 Controlled acceleration

FUNCTION DESCRIPTION:

Application conditions:

Initialisation: VS_SP_CRU = VS_CRU

Description:



The reference input variable VS_SP_CRU is set equal to the instantaneous speed of the vehicle VS_CRU at the start of the function.

The controlled acceleration of the vehicle is performed by controlled in/decrementation of the reference input variable of the cruise control. On start of the function, the time T_CTL_CRU is started and incremented in each computing cycle.
Is VS_SP_CRU is brought close to VS_SP_DRIV_CRU with fixed increment (VS_SP_DEC_CRU/4 or VS_SP_INC_CRU/4).
A distinction is made between positive and negative acceleration.
FAC_VS_CTL_CRU and FAC_VS_LGRD_CRU are set by the momentarily active function of the cruise control, see following subchapter.

Formula section:

T_CTL_CRUn = T_CTL_CRUn-1 + 100 ms

VS_SP_DIF_CRU = VS_SP_CRU - VS_SP_DRIV_CRU

IF(1) 0 < VS_SP_DIF_CRU ≤ VS_SP_DEC_CRU

Then(1) $VS\_SP\_CRU_{n}$ = $VS\_SP\_CRU_{n-1}$- VS_SP_DEC_CRU/4

Else(1)

IF(2) VS_SP_DIF_CRU > VS_SP_DEC_CRU > 0

```
Then(2)    VS_SP_CRU$_n$ =  VS_SP_CRU$_{n-1}$
               - VS_SP_DEC_CRU            :Decrement
               * FAC_VS_CTL_CRU           :Factor for controlling
               * FAC_VS_LGRD_CRU          :Factor for intercepting
Else(2)
If(3) (-1 * VS_SP_INC_CRU) $\leq$ VS_SP_DIF_CRU < 0
Then(3)    VS_SP_CRU$_n$ =  VS_SP_CRU$_{n-1}$+ VS_SP_INC_CRU/4
Else(3)
If(4)      VS_SP_DIF_CRU < (-1 * VS_SP_INC_CRU)
Then(4) VS_SP_CRU$_n$ =  VS_SP_CRU$_{n-1}$
               + VS_SP_INC_CRU            :Increment
               * FAC_VS_CTL_CRU           :Factor for controlling
               * FAC_VS_LGRD_CRU          :Factor for intercepting
Endif
Set of VS_SP_CRU in case of driver intervention:
If         (STATE_CRU = Set_acc or Retard)                        and
           LV_VAR_BN = 1                                          and
           FAC_TQ_REQ_DRIV > FAC_TQ_REQ_CRU                       and
           FAC_TQ_REQ_DRIV > IP_FAC_TQ_REQ_CLU_IS__N_32           and
           VS_CRU > VS_SP_CRU
Then       VS_SP_CRU = Min(VS_CRU, VS_SP_CRU_MAX)
Endif
```

Dec / Incrementation is interrupted if

VS_DIF_CRU > C_VS_DIF_MAX_AC_CRU **or**

VS_DIF_CRU < C_VS_DIF_MIN_AC_CRU

Dec / Incrementation is continued if

VS_DIF_CRU $\leq$ C_VS_DIF_MAX_AC_CRU **or**

VS_DIF_CRU $\geq$ C_VS_DIF_MIN_AC_CRU

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_DIF_MAX_AC_CRU	1	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Condition for interrupting the incrementation or decrementation					
C_VS_DIF_MIN_AC_CRU	1	8000...7FFFH	-256...255.99218	0.0078125	[km/h]
Condition for interrupting the incrementation or decrementation					

67.6.6.3 Actuating the reference input variable

General information:

The multiplicative correction factor FAC_VS_CTL_CRU for in/decrementation the reference input variable in the controlled acceleration is calculated in each case for the individual functions of the cruise control. Also for automatic transmission, a distinction is made partially whether the converter clutch is just open or closed. The correction factor depends upon the time T_CTL_CRU. While the actuation of this function is active, the calculation of the I component is interrupted. The sum of the I component thus remains constant.

Formula section:

FAC_VS_CTL_CRU = IP_FAC_XX_CRU_yy__T_CTL_CRU

A distinction is made between the following CRU functions:

Resume:	IP_FAC_CRU_REST__T_CTL_CRU
Setting/acceleration:	**IF** LV_AT = 0 **or** (LV_AT = 1 **and** STATE_CC != open)
Manual gear or automatic with closed converter clutch	**Then** IP_FAC_CRU_AC__T_CTL_CRU
Automatic with open converter	**Else** IP_FAC_CC_CRU_AC__T_CTL_CRU
Clutch	**Endif**
Deceleration:	**IF** LV_AT = 0 **or** (LV_AT = 1 **and** STATE_CC != open)
Manual gear or automatic with closed converter clutch	**Then** IP_FAC_CRU_DECE__T_CTL_CRU
Automatic with open converter	**Else** IP_FAC_CC_CRU_DECE__T_CTL_CRU
Clutch	**Endif**

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_FAC_CRU_REST__T_CTL_CRU	5	0...FFH	0...1.99218	0.0078125	[-]
LDPM_T_CTL_CRU_1	5	0...FFH	0...25.5	0.1	[s]
Correction factor of in/decrementing					
IP_FAC_CRU_AC__T_CTL_CRU	5	0...FFH	0...1.99218	0.0078125	[-]
LDPM_T_CTL_CRU_1	5	0...FFH	0...25.5	0.1	[s]
Correction factor of in/decrementing					
IP_FAC_CRU_DECE__T_CTL_CRU	5	0...FFH	0...1.99218	0.0078125	[-]
LDPM_T_CTL_CRU_1	5	0...FFH	0...25.5	0.1	[s]
Correction factor of in/decrementing					
IP_FAC_CC_CRU_AC__T_CTL_CRU	5	0...FFH	0...1.99218	0.0078125	[-]
LDPM_T_CTL_CRU_1	5	0...FFH	0...25.5	0.1	[s]
Correction factor of in/decrementing					
IP_FAC_CC_CRU_DECE__T_CTL_CRU	5	0...FFH	0...1.99218	0.0078125	[-]
LDPM_T_CTL_CRU_1	5	0...FFH	0...25.5	0.1	[s]
Correction factor of in/decrementing					

67.6.6.4 Intercepting the reference input variable target acceleration

General information:

To take account of the vehicle dynamics, the positive accelerations for high speeds as from approx. 80 k.p.h. must be reduced in the setting/acceleration and resume functions. For this purpose, the accelerations must be valued with a factor FAC_VS_LGRD_CRU for in/decrementing the reference input variable. The acceleration is valued with 1 below the threshold C_VS_AC_MIN_CRU. If VS_SP_CRU exceeds this threshold, then the factor FAC_VS_LGRD_CRU is reduced linearly by a ramp function up to the upper threshold C_VS_AC_MAX_CRU. The maximum reduction can be set through the value C_FAC_AC_POS_CRU.

To achieve comfortable interception of the reference input variable, the multiplicative correction factor, which is limited dependent on speed, must be calculated when in/decrementing the reference input variable in each case for the individual functions of the cruise control. While the interception of the reference input variable function is active, the calculation of the I component is interrupted. The sum of the I component remains constant. The correction factor depends upon the difference between reference input variable and target variable.

Formula section:

```
IF        VS_SP_CRU < C_VS_AC_MIN_CRU
Then          FAC_VS_LGRD_CRU = 1
Else If       VS_SP_CRU > C_VS_AC_MAX_CRU
    Then      IF LV_RNG_L_REQ = 1
              Then   FAC_VS_LGRD_CRU = C_FAC_AC_POS_CRU_RNG_L
              Else   FAC_VS_LGRD_CRU = C_FAC_AC_POS_CRU
Else          FAC_VS_LGRD_CRU =     FAC_VS_LGRD_CRU$_{vehicle-speed-dependent}$
                     * IP_FAC_LGRD_CRU_xx__VS_SP_DIF
```

During the acceleration IP_FAC_LGRD_CRU_xx__VS_SP_DIF = 1

(FAC_VS_LGRD_CRU = FAC_VS_LGRD_CRU$_{\text{vehicle-speed-dependent}}$)

- Resume (RES): IP_FAC_LGRD_CRU_REST__VS_SP_DIF
- Tip Up and Tip down: IP_FAC_LGRD_CRU_TIP__VS_SP_DIF (TP_UP, TP_DOWN)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VS_AC_MIN_CRU	1	0...FFH	0...255	1	[km/h]
Vehicle speed threshold for acceleration valuation					
C_VS_AC_MAX_CRU	1	0...FFH	0...255	1	[km/h]
Vehicle speed threshold for acceleration valuation					
C_FAC_AC_POS_CRU	1	0...FFH	0...0.99609	3.91E-03	[-]
Maximum reduction of FAC_VS_LGRD_CRU					
C_FAC_AC_POS_CRU_RNG_L	1	0...FFH	0...0.99609	3.91E-03	[-]
Maximum reduction of FAC_VS_LGRD_CRU at RNG_L					
IP_FAC_LGRD_CRU_TIP__VS_SP_DIF	5	0...FFH	0...0.99609	3.91E-03	[-]
LDPM_VS_SP_DIF_CRU_1	5	0...FFFFH	-256...255.99218	0.0078125	[km/h]
Interception factor of the in/decrementing (xx = REST, TIP)					
IP_FAC_LGRD_CRU_REST__VS_SP_DIF	5	0...FFH	0...0.99609	3.91E-03	[-]
LDPM_VS_SP_DIF_CRU_1	5	0...FFFFH	-256...255.99218	0.0078125	[km/h]
Interception factor of the in/decrementing (xx = REST, TIP)					

67.6.6.5 Interventions

Application conditions:

Initialisation: VS_SP_DRIV_CRU = VS_MIN_CRU at LV_IGK 0 to1

Formula section:

```
IF    STATE_VS_ICL_DISP = 1            "km/h; mph"
Then      VS_SP_CRU_MAX = C_VS_SP_CRU_MAX_M
Else      VS_SP_CRU_MAX = C_VS_SP_CRU_MAX_K
Endif
```

67.6.6.5.1 Resume (RES)

Description:

When the resume function is activated, the vehicle is brought to the last target speed with the adjustable positive or negative acceleration. In this case the control basic function is activated and the reference input variable is controlled. When controlling the reference input variable VS_SP_CRU, a distinction is made between a resume to high speeds and a resume to the low speeds.
In a resume to a target speed, the reference input variable is in / decremented per computing cycle (100 ms).

To achieve a comfortable transition from the CRU status of ready to CRU mode on start of the restart function, the I component of the P/I controller is preallocated depending upon the recognised acceleration of the vehicle. To achieve comfortable acceleration on activating the resume function and on reaching the stored target speed, the control and interception of the reference input variable functions are series connected with the controlled acceleration function. In these functions a multiplicative correction factor is calculated in each case for in/decrementing the reference input variable.
At resume the after overtaking function is finished the resume function is activated with the initilisation calculation of FAC_TQ_CRU.
The alghorithm for Step request is like the State resume. Only VS_SP_DRIV_CRU is set to the current select mark VS_SP_DRIV_STEP. Inside the controller a step request triggers the resume Function.

Formula section:

```
VS_SP_CRU = VS_CRU


IF      VS_CRU  <  VS_SP_DRIV_CRU

Then IF LV_RNG_L_REQ = 1
        Then  VS_SP_INC_CRU  =  3.6  *  ID_AC_POS_CRU_RESU_RNG_L  *  100 ms
        Else  VS_SP_INC_CRU  =  3.6  *  ID_AC_POS_CRU_RESU  *  100 ms
  VS_SP_CRU  = capter "calculation of controlled acceleration"
Else
If        VS_CRU  >  VS_SP_DRIV_CRU
Then   VS_SP_DEC_CRU  =  3.6  * IP_AC_NEG_CRU_RESU *  100 ms
     VS_SP_CRU  = capter "calculation of controlled acceleration"
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
IP_AC_NEG_CRU_RESU	3*3	0...5C28H	0...511.9789	2.17E-02	[m/s²]
LDPM_N_21	3	0...FFH	0...8160	32	[rpm]
LDPM_FAC_TQ_CRU_1	3	0...FFFFH	0...99.99847	1.53E-03	[%]
Negative acceleration at CRU resume					
ID_AC_POS_CRU_RESU	8	0...5C28H	0...511.9789	0.0217014	[m/s²]
LDPM_GEAR_CRU	8	0...FFH	0...255	1	[-]
Positive acceleration at CRU resume depending on GEAR					
ID_AC_POS_CRU_RESU_RNG_L	8	0...5C28H	0...511.9789	0.0217014	[m/s²]
LDPM_GEAR_CRU	8	0...FFH	0...255	1	[-]
Positive acceleration at CRU resume at RNG_L depending on GEAR					

67.6.6.5.2 CRU Step intervention

Formula section:

Normal operation

```
If    LV_STEP_ON_ACT = 1
Then if      STATE_CRU != Passiv
     Then    VS_SP_DRIV_CRU = VS_SP_DRIV_STEP
             If       VS_SP_DRIV_CRU > VS_SP_CRU
             Then  if       VS_SP_CRU > VS_CRU
                   Then   VS_SP_CRU remains unchanged
```

```
                Else    VS_SP_CRU = VS_CRU
                Endif
            Else    if      VS_SP_CRU > VS_CRU
                    Then    VS_SP_CRU = VS_CRU
                    Else    VS_SP_CRU remains unchanged
                    Endif
            Endif
            STATE_CRU = Resume
        Endif
    Endif
```

Additional for Overtaking

```
If      STATE_CRU = Resume                              and
        LV_VAR_BN                                       and
        VS_SP_DRIV_CRU > VS_SP_CRU                      and
        $VS\_SP\_CRU_{(n)} \geq VS\_SP\_CRU_{(n-1)}$    and
        FAC_TQ_REQ_DRIV > FAC_TQ_REQ_CRU                and
        FAC_TQ_REQ_DRIV > IP_FAC_TQ_REQ_CLU_IS__N_32    and
        VS_CRU > VS_SP_CRU
Then    VS_SP_CRU = VS_CRU
Endif
```

67.6.6.5.3 Setting/acceleration (SET/ACC)

Description:

On activating the setting/acceleration function from the CRU State of ready, the transition to ‘controlled constant drive’ takes place. Reference input variable VS_SP_CRU and target variable VS_SP_DRIV_CRU are set to the current speed VS_CRU.

On activating the setting/acceleration function from readiness, a timer T_CTL_CRU is started. As long as SET/ACC recognizes that this timer reaches the value C_T_MIN_CRU_CON, the CRU state changes from ‘controlled constant drive’ to ‘controlled acceleration’. The vehicle is brought to the required target speed with the adjustable positive ID_AC_POS_CRU_AC/_RNG_L or negative acceleration IP_AC_NEG_CRU_AC__N__FAC_TQ_CRU. In this case the control basic function is activated and the reference input variable VS_SP_CRU is controlled.

<u>As long as the setting/acceleration function is detected, the incrementation of the reference input variable takes place per computing cycle (100 ms) with the controlled acceleration function.</u>

On ending the setting/acceleration function, the target speed VS_SP_DRIV_CRU is set equal to the achieved vehicle speed VS_CRU. If the target speed VS_SP_DRIV_CRU stored in this way is greater than the instantaneous reference input variable VS_SP_CRU, then the value VS_SP_CRU is incremented further by the value VS_SP_INC_CRU through the controlled acceleration function. If the instantaneous reference input variable VS_SP_CRU is larger than the stored target speed VS_SP_DRIV_CRU, then the reference input variable is decremented with the controlled acceleration function.

<u>Formula section:</u>

```
REQ_MSW = ACC active:
IF        LV_RNG_L_REQ = 1
Then      VS_SP_INC_CRU  =  3.6  *  ID_AC_POS_CRU_AC_RNG_L  *  100 ms
Else      VS_SP_INC_CRU  =  3.6  *  ID_AC_POS_CRU_AC  *  100 ms
Endif
VS_SP_CRU$_{n}$  =   VS_SP_CRU$_{n-1}$
              +  VS_SP_INC_CRU            :Increment
              *  FAC_VS_CTL_CRU           :Factor for controlling
              *  FAC_VS_LGRD_CRU          :Factor for intercepting


REQ_MSW = ACC passive:
VS_SP_DRIV_CRU = Min(VS_CRU, VS_SP_CRU_MAX)
IF        T_CONST_DRIVE ≥ C_T_MIN_CRU_CON
Then IF   VS_SP_CRU <  VS_SP_DRIV_CRU
    Then IF LV_RNG_L_REQ = 1
        Then  VS_SP_INC_CRU  =  3.6  *  ID_AC_POS_CRU_AC_RNG_L  *  100 ms
          Else VS_SP_INC_CRU  =  3.6  *  ID_AC_POS_CRU_AC  *  100 ms
  VS_SP_CRU  = capter "calculation of controlled acceleration"
    Else
        If    VS_SP_CRU >  VS_SP_DRIV_CRU
        Then  VS_SP_DEC_CRU = 3.6 * IP_AC_NEG_CRU_AC__N__FAC_TQ_CRU* 100 ms
        VS_SP_CRU  = capter "calculation of controlled acceleration"
        Endif
    Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. Limits	Resol.	Unit
IP_AC_NEG_CRU_AC__N__FAC_TQ_CRU	3*3	0...5C28H	0...511.9789	2.17E-02	[m/s²]
LDPM_N_21	3	0...FFH	0...8160	32	[rpm]
LDPM_FAC_TQ_CRU_1	3	0...FFFFH	0...99.99847	1.53E-03	[%]
Negative acceleration in setting/acceleration					
C_T_MIN_CRU_CON	1	0...FFH	0...25.5	0.1	[s]
Time condition for controlled acceleration					
ID_AC_POS_CRU_AC	8	0...5C28H	0...511.9789	0.0217014	[m/s²]
LDPM_GEAR_CRU	8	0...FFH	0...255	1	[-]
Positive acceleration in setting/acceleration depending on GEAR					
ID_AC_POS_CRU_AC_RNG_L	8	0...5C28H	0...511.9789	0.0217014	[m/s²]
LDPM_GEAR_CRU	8	0...FFH	0...255	1	[-]
Positive acceleration in setting/acceleration at RNG_L depending on GEAR					

67.6.6.5.4 Tip-up

Description:

On activating the TIP_UP function, the vehicle is brought to the last target speed reached, i.e. VS_SP_DRIV_CRU plus C_VS_DIF_TIP_M/K [km/h] with the adjustable positive acceleration C_AC_POS_CRU_TIP. In this case the control basic function is activated and the reference input variable VS_SP_CRU is controlled. By each detection of the TIP_UP function, the target speed VS_SP_DRIV_CRU is increased by C_VS_DIF_TIP_M/K [km/h].
If the absolute amount of the difference VS_DIF_CRU is greater than or equal to C_VS_DIF_MAX_CRU, the reference input variable VS_SP_CRU is incremented per computing cycle (100 ms) with the controlled acceleration function by the value:

Formula section:

```
IF        LV_CRU_OVER_ACT = 1    and    REQ_CRU = "TIP_UP"
Then      VS_SP_CRU = VS_SP_DRIV_CRU = MIN(VS_CRU, VS_SP_CRU_MAX)
          LV_CRU_OVER_ACT_ACK = 1
else      LV_CRU_OVER_ACT_ACK = 0
endif

IF        VS_SP_DRIV_CRU ≥ VS_SP_CRU_MAX
Then      VS_SP_DRIV_CRU = VS_SP_CRU_MAX
Else if   STATE_VS_ICL_DISP = 1
    Then  VS_SP_DRIV_CRU = VS_SP_DRIV_CRU + C_VS_DIF_TIP_M [km/h]
    Else  VS_SP_DRIV_CRU = VS_SP_DRIV_CRU + C_VS_DIF_TIP_K [km/h]
    Endif
Endif
```

```
IF          $VS\_DIF\_CRU \geq C\_VS\_DIF\_MAX\_CRU$
Then        VS_SP_INC_CRU = 3.6 * C_AC_POS_CRU_TIP * 100 ms
            $VS\_SP\_CRU_{n}$ = $VS\_SP\_CRU_{n-1}$
               + VS_SP_INC_CRU             :Increment
                * FAC_VS_CTL_CRU           :Factor for controlling
                * FAC_VS_LGRD_CRU          :Factor for intercepting

Else if     STATE_VS_ICL_DISP = 1
    Then    VS_SP_CRU = VS_SP_CRU + C_VS_DIF_TIP_M [km/h]
    Else    VS_SP_CRU = VS_SP_CRU + C_VS_DIF_TIP_K [km/h]
    Endif
Endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_AC_POS_CRU_TIP	1	0...5C28H	0...511.9789	2.17E-02	[m/s²]
Positive acceleration for Tip-up					
C_VS_DIF_MAX_CRU	1	0...FFH	0...31.875	0.125	[km/h]
Control difference threshold for incrementing VS_SP_CRU					
C_VS_SP_CRU_MAX_K	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
maximum limitation of cruise control in case of KM - ICL					
C_VS_SP_CRU_MAX_M	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
maximum limitation of cruise control in case of Mile - ICL					
C_VS_DIF_TIP_M	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Km per tip in case of mile – icl					
C_VS_DIF_TIP_K	1	0...FFFFH	0...511.99218	0.0078125	[km/h]
Km per tip in case of km - icl					

67.6.6.5.5 Tip-down

Description:

On activating the TIP_DOWN function, the vehicle is brought to the last target speed reached, i.e. VS_SP_DRIV_CRU minus C_VS_DIF_TIP_M/K [km/h] by the adjustable negative acceleration C_AC_NEG_CRU_TIP. In this case the control basic function is activated and the reference input variable VS_SP_CRU is controlled. At each detection of the TIP_DOWN function, the target speed VS_SP_DRIV_CRU is reduced by C_VS_DIF_TIP_M/K [km/h].
VS_SP_CRU and VS_SP_DRIV_CRU cannot be less than VS_MIN_CRU.
If the absolute amount of the difference VS_DIF_CRU is greater than or equal to C_VS_DIF_MAX_CRU, the reference input variable VS_SP_CRU is decrements per computing cycle (100 ms) with the controlled acceleration function. The charge wish FAC_TQ_CRUis limited to the value (IP_FAC_TQ_REQ_CLU_IS__N_32*99,998%). so that no trailing throttle cut-off is triggered (except for: PUC was already active before TIP_DOWN).

Formula section:

```
IF          LV_CRU_OVER_ACT = 1    and    REQ_CRU = "TIP_DOWN"
Then        VS_SP_CRU = VS_SP_DRIV_CRU = MIN(VS_CRU; VS_SP_CRU_MAX)
            LV_CRU_OVER_ACT_ACK = 1
else        LV_CRU_OVER_ACT_ACK = 0
endif

IF          VS_SP_DRIV_CRU < VS_MIN_CRU
Then        VS_SP_DRIV_CRU = C_VS_SP_CRU_MIN
Else if     STATE_VS_ICL_DISP = 1
    Then    VS_SP_DRIV_CRU = VS_SP_DRIV_CRU - C_VS_DIF_TIP_M [km/h]
    Else    VS_SP_DRIV_CRU = VS_SP_DRIV_CRU - C_VS_DIF_TIP_K [km/h]
    Endif
Endif

IF          |VS_DIF_CRU| ≥ C_VS_DIF_MAX_CRU
Then        VS_SP_DEC_CRU  =  3.6  *  C_AC_NEG_CRU_TIP  *  100 ms
    $VS\_SP\_CRU_{n} = VS\_SP\_CRU_{n-1}$
                              -  VS_SP_DEC_CRU              :Decrement
                              *  FAC_VS_CTL_CRU             :Factor for controlling
                                   *  FAC_VS_LGRD_CRU            :Factor for intercepting
Else if     STATE_VS_ICL_DISP = 1
    Then    VS_SP_CRU = VS_SP_CRU - C_VS_DIF_TIP_M [km/h]
    Else    VS_SP_CRU = VS_SP_CRU - C_VS_DIF_TIP_K [km/h]
    Endif
Endif
```

$FAC\_TQ\_CRU \geq ((IP\_FAC\_TQ\_REQ\_CLU\_IS\_\_N\_32 - C\_FAC\_TQ\_REQ\_CLU\_IS\_HYS) * 99{,}998\%)$

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_AC_NEG_CRU_TIP	1	0...5C28H	0...511.9789	2.17E-02	[m/s²]
Positive acceleration for Tip-up					
C_VS_DIF_MAX_CRU	1	0...FFH	0...31.875	0.125	[km/h]
Control difference threshold for incrementing VS_SP_CRU					

67.6.6.5.6 Deceleration

Description:

On activating the deceleration function, the vehicle is brought to the required target speed with the adjustable negative C_AC_NEG_CRU_DEACC or positive acceleration C_AC_POS_CRU_DEACC (for the steady state). In this case the control basic function is activated and the reference input variable VS_SP_CRU is controlled.
As long as the deceleration function is detected, the decrementation of the reference input variable takes place per computing cycle (100 ms) with the controlled acceleration function

On ending the deceleration function, the target speed VS_SP_DRIV_CRU is set equal with the reached vehicle speed VS_CRU. If the target speed VS_SP_DRIV_CRU thus stored is smaller than the instantaneous reference input variable VS_SP_CRU, then the value VS_SP_CRU is decremented by the value VS_SP_DEC_CRU further via the controlled acceleration function. VS_SP_CRU and VS_SP_DRIV_CRU cannot be smaller than VS_MIN_CRU. If the instantaneous reference input variable VS_SP_CRU is smaller than the stored target speed VS_SP_DRIV_CRU, then the reference input variable is incremented with the controlled acceleration function.

Formula section:

REQ_MSW = DEC active:

VS_SP_DEC_CRU = 3.6 * C_AC_NEG_CRU_DEACC * 100 ms

$\mathrm{VS\_SP\_CRU}_n = \mathrm{VS\_SP\_CRU}_{n-1}$

- VS_SP_DEC_CRU : Decrement

* FAC_VS_CTL_CRU : Factor for controlling

* FAC_VS_LGRD_CRU : Factor for intercepting

REQ_MSW = DEC passive:

VS_SP_DRIV_CRU = Min(VS_CRU, VS_SP_CRU_MAX)

```
IF    VS_SP_DRIV_CRU <  VS_SP_CRU
Then      VS_SP_DEC_CRU  =  3.6  *  C_AC_NEG_CRU_DEACC  * 100 ms
          VS_SP_CRU  = capter "calculation of controlled acceleration"
Else      VS_SP_INC_CRU  =  3.6  *  C_AC_POS_CRU_DEACC  * 100 ms
          VS_SP_CRU  = capter "calculation of controlled acceleration"
Endif
```

As long as decrementing the reference input variable VS_SP_CRU is active, the calculation of the I component is released.

Calculation of VS_DIF_I_CRU until

$\mathrm{VS\_DIF\_I\_CRU} + \mathrm{C\_VS\_DIF\_DELTA\_I} < |\mathrm{VS\_DIF\_CRU}|$

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_AC_NEG_CRU_DEACC	1	0...5C28H	0...511.9789	2.17E-02	[m/s²]
Negative acceleration in deceleration					
C_AC_POS_CRU_DEACC	1	0...5C28H	0...511.9789	2.17E-02	[m/s²]
Positive acceleration					
C_VS_DIF_DELTA_I	1	0...FFH	0...63.75	0.25	[km/h]
Positive acceleration					

67.6.6.5.7 OFF

In the OFF function, the last setpoint VS_SP_DRIV_CRU remains stored and soft cut-off is activated. If the engine running is stopped (LV_IGK, LV_ES) the CRU - readiness and the last speed-setpoint are to reset to their initial values.

67.6.7 Additional functions

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
N_VS_CRU	V/O	0...1FE0H	0...16320	2	[rpm/(km/h)]
Ratio between engine speed and vehicle speed					
VS_FIL	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Filtered vehicle speed					
T_ERR_VS_CAN	V/O	0...FFFFH	0...655.35	0.01	[ms]
Time to freeze VS_FIL before VS_FIL = 0					
VS_CRU	V/O	0...FFFFH	0...511.99218	0.0078125	[km/h]
Filtered vehicle speed					
VS_DIF_ICL_TCS	V	0...1H	0...1	1	[-]
Difference of vehicle speed INSTR and TCS					
AC_CRU	V/O	8000...7FFFH	-15.72864... 15.72816	0.00048	[m/s²]
Vehicle acceleration in CRU					
FAC_TQ_REQ_DRIV_MMV	V/O	0...FFFFH	0...1.99996	3.05E-05	[-]
Mean moving value of driver torque demand					
LV_VS_DIAG_NOT_PLAUS	V/O	0...1H	0...1	1	[-]
VS_CRU not plausible					
LV_CRU_OFF_BY_ASR_ESP_CTL	V/O	0...1H	0...1	1	[-]
Soft off condition depending on ASC interventions					

Input data:

LV_ERR_CAN_BOFF	LV_VAR_BN_MSW	FAC_TQ_REQ_DRIV	LV_AT
LV_ERR_BN_ICL	LV_ERR_BN_VS_TCS	LV_ERR_CAN_BOFF	LV_ERR_VS
LV_ERR_VS_CAN	LV_PLAUS_ASR_CTL	LV_PLAUS_ESP_CTL	LV_TQ_ASR_REQ
LV_VAR_AMT	LV_VAR_ASR	N	VS
VS_FIL	VS_ICL_DISP	WHEEL	LV_VAR_TCT
LV_ERR_VS	LV_ERR_TOUT_ASR_1	VS_SENS	

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_VS_FIL	1	0...FFH	0...255	1	[-]
Filter factor					
C_T_ERR_VS_CAN	1	0...FFH	0...25.5	0.1	[s]
Maximum time after VS CAN error to set VS_FIL to 0					
C_CRLC_VS_CRU_MMV	1	0...FFFFH	0...0.99998	1.53E-05	[km/h]
Filter factor					
LC_VS_CRU_FIL	1	0...1H	0...1	1	[-]
Filter factor activation					
C_FAC_AC_CRU_MMV	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Filtering AC_CRU					
C_VS_MAX_DIAG	1	0...FFH	0...255	1	[km/h]
Max. allowed deviation between VS_CRU and VS					
C_T_VS_DIAG_NOT_PLAUS	1	0...FFH	0...25.5	0.1	[s]
debounce time for setting VS_CRU not plausible					
C_T_CRU_ESP_OFF	1	0...FFH	0...25.5	0.1	[s]
Minimum time before cut-off CRU in case of ESP intervention					
C_T_CRU_ASR_OFF	1	0...FFH	0...25.5	0.1	[s]
Minimum time before cut-off CRU in case of ASR intervention					
C_T_CRU_ESP_BRAKE_OFF	1	0...FFH	0...25.5	0.1	[s]

Minimum time before cut-off CRU in case of ESP intervention only with brake					
C_FAC_TQ_REQ_DRIV_MMV	1	0...FFFFH	0...0.99998	1.53E-05	[-]
Filtering of driver demand					

67.6.7.1 Runup lock

Application conditions:

Initialisation: 0 at reset

Activation: LV_AT = 0 **and** LV_VAR_AMT = 0 **and** LV_VAR_TCT = 0

Deactivation: LV_AT = 1 **or** LV_VAR_AMT = 1 **or** LV_VAR_TCT = 1

Recurrence: 100 ms

Description:

In vehicles with manual transmission, the ratio between engine speed and vehicle speed is monitored on activating a function of the cruise control. At the start of a function, the ratio N_VS_CRU = N / VS_FIL is stored.

The instantaneous ratio between engine speed and vehicle speed is formed in each cycle

$N\_VS\_CRU_n = N_n / VS\_FIL_n$

Remark:

If the percentage ratio exceeds the threshold value C_N_VS_MAX_CRU, hard cut-off has to ocurrs.

67.6.7.2 Filter of vehicle speed

Application conditions:

Initialization: 0 at reset

Activation: LV_VAR_ASR = 1

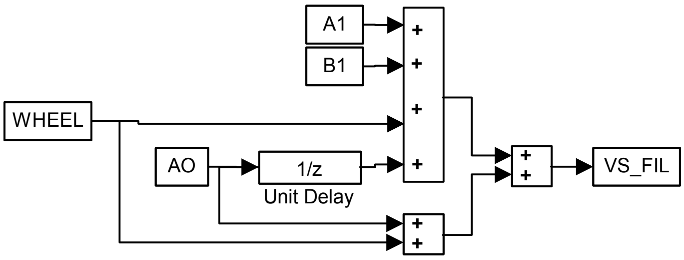
Recurrence: 10ms

General information:

For the CAN signal WHEEL, a 1st order IIR filter is used for evaluating the vehicle speed for the CRU and the speed limitation. The jitter error of the input signal WHEEL is smaller than 1%. A jitters error smaller than 0.2% must be achieved for the control variable VS_FIL with the filter. The frequency limit of the filter is approx. fg = 0.80 (Hz). Filtration of the speed signal is performed independently of the CRU mode with a 10ms time pattern.

Signal flow diagram:

The structure of the filter is represented as follows:



z^{-1} means a delay by one sampling time. The sampling time is T_{ab} = 10(ms) and thus f_{ab} = 100 (Hz). At a quality Q = 1, the filter factors are calculated as:

Formula section:

$$A1 = A0 = \frac{1}{1+W}; \qquad B1 = -\frac{1-W}{1+W}; \qquad \text{with } W = \frac{1}{\mathrm{Tan}\frac{\pi}{\Omega}}; \qquad W = C\_FAC\_VS\_FIL$$

```
If      LV_ERR_VS = 1                 // VS-double-error (PIN & CAN-signal)
Then    VS_FIL = 0
Else    If      LV_ERR_CAN_BOFF = 1          or
                LV_ERR_BN_VS_TCS = 1         or
                LV_ERR_TOUT_ASR_1 = 1
        Then    VS_FIL = VS_SENS (converted)
        ElseIf  LV_ERR_VS_CAN = 1
           Then     T_ERR_VS_CAN decrements (Start value C_T_ERR_VS_CAN )
                If      T_ERR_VS_CAN = 0
                Then    VS_FIL = 0
                Else    VS_FIL n = VS_FIL n-1
                Endif
```

```
            Else    VS_FIL is calculated
                    T_ERR_VS_CAN = 0
            Endif
        Endif
Endif
```

67.6.7.3 Filter of Vehicle Speed for Cruise control

Application conditions:

Initialisation: 0 at reset

Activation: LV_VAR_ASR = 1

Recurrence: 100ms

General information:

With integration of stage cruise control the vehicle speed of instrument message is used. There is a deviation of DSC vehicle speed to the vehicle Speed of ICL therefore the visible signal has to be used to supress a shown deviation to the driver.

Formula section:

```
If      LV_VAR_BN_MSW = 1       and
        LC_VS_CRU_FIL =1        and
        LV_ERR_CAN_BOFF = 0     and
        LV_ERR_BN_VS_TCS = 0    and
        LV_ERR_BN_ICL = 0
Then    $VS\_DIF\_ICL\_TCS = VS\_DIF\_ICL\_TCS_{n-1} - C\_CRLC\_VS\_CRU\_MMV$     *
        $(VS\_DIF\_ICL\_TCS_{(n-1)} - VS\_ICL\_DISP + VS\_FIL)$
        VS_CRU = VS_FIL + VS_DIF_ICL_TCS
        (a physical conversion VS_ICL_DISP has be done)
Else    VS_CRU = VS_FIL
Endif
```

67.6.8 Checking the conditions of the CRU mode

67.6.8.1 Acceleration monitoring

Application conditions:

Initialisation: 0 m/s² at reset

Recurrence: 10 ms

Description:

To monitor the vehicle dynamics in cruise control mode, the instantaneous acceleration of the vehicle is calculated in each computing cycle. The acceleration calculation is performed independently of the CRU mode with a 10 ms time pattern.

Formula section:

$$AC\_CRU = AC\_CRU + (VS\_FIL_{n} - VS\_FIL_{n-1}) / 10ms * C\_FAC\_AC\_CRU\_MMV$$

Remark:
If the instantaneous acceleration exceeds the adjustable value C_AC_MAX_CRU, hard cut-off is performed, see chapter hard off conditions.

67.6.8.2 Plausibility check of the speed signal VS_FIL

General information:

The vehicle speed is acquired through the CAN signal WHEEL. To detect possible errors - if WHEEL is currently received and identified as correct by the ASR control unit a plausibility check is performed.

The filtered external vehicle speed VS_CRU is compared with the vehicle speed sensor signal. If a maximum allowed absolute deviation C_VS_MAX_DIAG is exceeded, there is an error entry.

Application conditions:

Initialisation: 0 at reset or LV_IGK = 0 → 1

Recurrence: Checking every 100 ms:

Activation: LV_VAR_ASR = 1

Formula section:

```
IF   LV_ERR_VS              = 1         or
     LV_ERR_CAN_BOFF        = 1         or
     LV_ERR_BN_VS_TCS       = 1         or
     LV_ERR_BN_ICL          = 1         or
     LV_ERR_VS_CAN          = 1
THEN      LV_VS_DIAG_NOT_PLAUS = 0
          T_VS_DIAG_NOT_PLAUS = 0
ELSE      IF   |VS_CRU - VS|  ≥  C_VS_MAX_DIAG
```

```
        THEN T_VS_DIAG_NOT_PLAUS ++
             IF    T_VS_DIAG_NOT_PLAUS > C_T_VS_DIAG_NOT_PLAUS
             THEN  LV_VS_DIAG_NOT_PLAUS = 1
             ENDIF
        ELSE       LV_VS_DIAG_NOT_PLAUS = 0
                   T_VS_DIAG_NOT_PLAUS = 0
        ENDIF
ENDIF
```

67.6.8.3 Recognition ASR/ESP-control and valid minimum time

Application conditions:

Initialisation: 0 at reset

Recurrence: 100 ms

Description:

Recognition ASR/ESP-control and valid minimum time :

LV_CRU_OFF_BY_ASR_ESP_CTL = active:

Recognition due to the combination of the signals LV_PLAUS_ASR_CTL, LV_ASR_REQ and LV_PLAUS_ESP_CTL of the CAN-message ASC3

LV_PLAUS_ASR_CTL	LV_TQ_ASR_REQ	LV_PLAUS_ESP_CTL	function	minimum time before cut-off CRU
1	1	0	ASR-control	C_T_CRU_ASR_OFF
1	1	1	ESP-control	C_T_CRU_ESP_OFF
1	0	1	ESP-control only brake	C_T_CRU_ESP_BRAKE_OFF

67.6.8.4 Mean moving value calculation of Driver demand:

FUNCTION DESCRIPTION:

For initialisation of I share of the cruise control function it is nessessary to calulate a very slow driver demand

Application conditions:

Initialisation: 0 at reset

Recurrence: 10 ms

Formula section:

$$\mathrm{FAC\_TQ\_REQ\_DRIV\_MMV} = \mathrm{FAC\_TQ\_REQ\_DRIV\_MMV}_{n-1} + (\mathrm{FAC\_TQ\_REQ\_DRIV} - \mathrm{FAC\_TQ\_REQ\_DRIV\_MMV}_{n-1}) * \mathrm{C\_FAC\_TQ\_REQ\_DRIV\_MMV}$$

67.7 Cus adap module: VHSC

67.7.1 Outputs for BMW functions which are not defined as VHSC exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Mdr_fgr	O/V	0...FFFFH	0...255.99609	3.9063e-3	[%]
Relatives Wunschmoment vom Fahrgeschwindigkeitsregler					
Mdi_accl_plus	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
torque increase for slow torque intervention during DCC					
Mdi_accs_plus	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Torque increase for fast torque intervention during DCC					

Input data:

TQI_DCC_SLOW_INC	TQI_DCC_FAST_INC	FAC_TQ_REQ_CRU	

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: The really possible physical limits of the outputs are different to the specified values due to the input data attributes.

Application conditions:

Initialisation: 0 at reset

Recurrence : 10 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

Mdi_accl_plus = TQI_DCC_SLOW_INC

Mdi_accs_plus = TQI_DCC_FAST_INC

Mdr_fgr = FAC_TQ_REQ_CRU * 100 %

68 Vehicle speed limitation

Table of Contents

Index

68.1 VHSL General

General information:

The Aggregate VHSL replaces previous function which was engine speed based. Aim of function is to provide better control of vehicle speed, particularly in the avoidance of overshoot of the setpoint, and stability after gear changes (esp. in CVT vehicles).

The aggregate consits of the following main functions:

Vehicle speed limitation manager
Interface not used for the fix limitation ->VHSL_1.0.0. The state variable STATE_VSL is used only as aggregate internal interface in case of a variable vehicle speed limitation.

Vehicle speed limitation controller
This modules aims to limit the vehicle speed to C_VS_MAX_VSL. A PI controller is used to define the engine torque limitation. In order to minimise speed overshoot, a predicted vehicle speed is calculated and the I controller is initilised with a calculated torque reduction.

Vehicle speed limitation output
This functionality limits the engine torque output by TQI_VS_MAX. The value of TQI_VS_MAX is initialised with the current engine output torque requirement. This will be calculated as the maximum of TQI_AV and minimum of TQI_BAS_MIN, TQI_REQ_TRA. This starting value of TQI_VS_MAX is then stored.

Vehicle speed limit calculation
A filtered vehicle acceleration, AC_VEH_MMV is used to predict a future vehicle speed VS_PRED. The vehicle speed limit, VS_MAX is also determinated at this point. For the fix limitation this limit will be given by the constant C_VS_MAX_VSL.
Another part is the calculation of the vehicle speed deviation which is an essential variable for the vehicle speed controller.

Vehicle speed limitation manager (Appl. Inc.)
This module deals with the activation of the fix vehicle speed limitation functionality. The function is switched on, LV_VS_MAX_VSL is set, when the vehicle speed, VS_HIGH_RES exceeds the set point minus an offset, C_VS_MAX_VSL - C_VS_MAX_HYS_ON. On the other hand the function is reset, and turned off, when the vehicle speed, VS_HIGH_RES drops below the set point minus a hysterisis

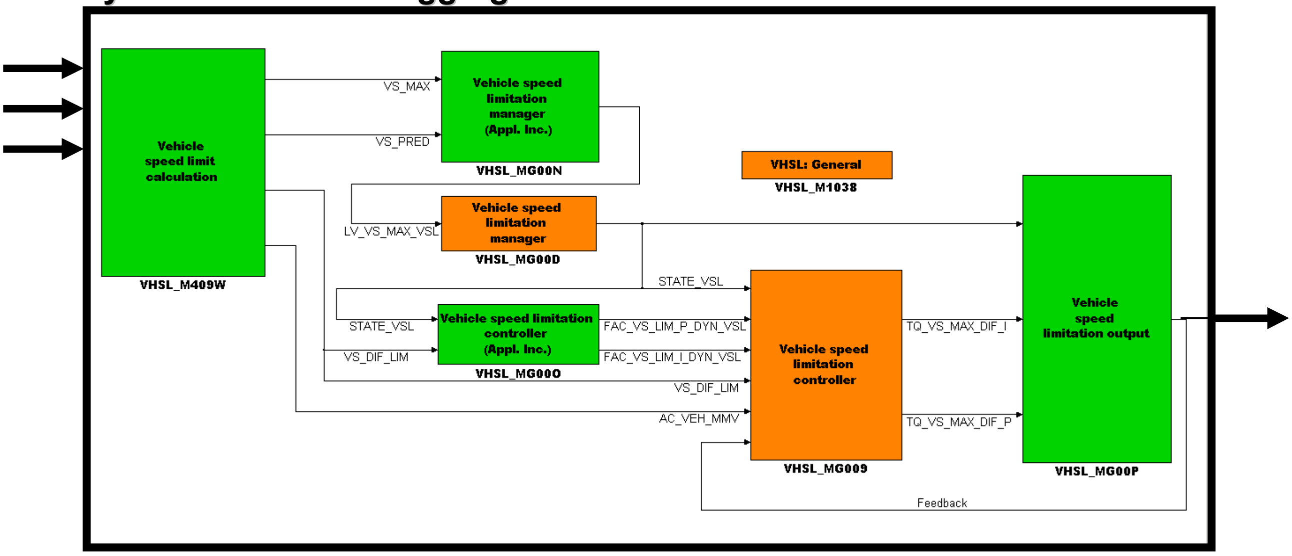
Vehicle speed limitation controller (Appl. Inc.)
The control dynamic is determined by LV_FAC_CTL_DYN_CHG_VSL. The I and PD control parameters are calculated based on the dynamic behaviour of the controller.

The aggregate is available in 2 variants:
- Variant 1 : Fix Vehicle Speed Limitation
- Variant 2 : Variable Vehicle Speed Limitation

Architecture Overview:

Variant 1:

System Arhitecture Aggregate VHSL Version 1



68.2 Vehicle speed limit calculation

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
VS_MAX	O/V	0...7FFFH	0...327.67	0.01	km/h
Maximum allowed vehicle speed					
VS_DIF_LIM	O/V	8000...7FFFH	-327.68...327.67	0.01	km/h
Vehicle speed deviation for speed limiter					
AC_VEH_MMV	O/V	8000...7FFFH	-27.77088 ... 27.770033	8.475E-4	m/s²
Current Vehicle Acceleration - Filtered					
VS_PRED	O/V	0...7FFFH	0...327.67	0.01	km/h
Predicted Vehicle Speed for Speed Limitation					
LV_VS_MAX_LIH_TCT_ACT	V	0...1H	0...1	1	-
Vehicle speed maximum limphome for TCT					

Input data:

AC_VEH	VS_HIGH_RES	LV_IGK	N_VS
VS_MAX_SEL_EXT_REQ	LV_GP	VS_MAX_LIH_TCT	PV_AV

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_CRLC_AC_VEH_VS_MAX	1	0...FFFFH	0...0.99998474	1.52588E-5	-
PT1 filter constant for low pass filter					
C_PV_MIN_LIH_TCT_ACT	1	0...FFH	0...99.609375	0.390625	%
Pedal value position minimum limphome for TCT					
C_VS_DIF_LIH_TCT_DEAC	1	0...FFH	0...255	1	km/h
Vehicle speed difference limphome for TCT					
C_VS_MAX_0_DFT	1	0...FFH	0...255	1	km/h
Vehicle speed limitation threshold 0 - default value					
C_VS_MAX_1	1	0...FFH	0...255	1	km/h
Vehicle speed limitation threshold 1					
C_VS_MAX_2	1	0...FFH	0...255	1	km/h
Vehicle speed limitation threshold 2					
C_VS_MAX_3	1	0...FFH	0...255	1	km/h
Vehicle speed limitation threshold 3					
C_VS_MAX_AMT	1	0...FFH	0...255	1	km/h
Vehicle speed limitation threshold - SSG					
C_VS_MAX_LIH_TCT_DEC	1	0...FFH	0...255	1	km/h
Vehicle speed maximum limphome for TCT decrement					
C_VS_MAX_LIH_TCT_INC	1	0...FFH	0...255	1	km/h
Vehicle speed maximum limphome for TCT increment					
ID_T_VS_MAX_PRED	8	0...FFH	0...7.96875	0.03125	s
LDPM_N_VS_VS_MAX	8	0...FFH	0...255	1	rpm/(km/h)
Prediction time for predicted engine speed for automatic transmission					

68.2.1 VHSL_M409W

A filtered vehicle acceleration, AC_VEH_MMV is used to predict a future vehicle speed VS_PRED. The vehicle speed limit, VS_MAX is also determinated at this point. For the fix limitation this limit will be given by the constant C_VS_MAX.

Another part is the calculation of the vehicle speed deviation which is an essential variable for the vehicle speed controller.

Application Condition

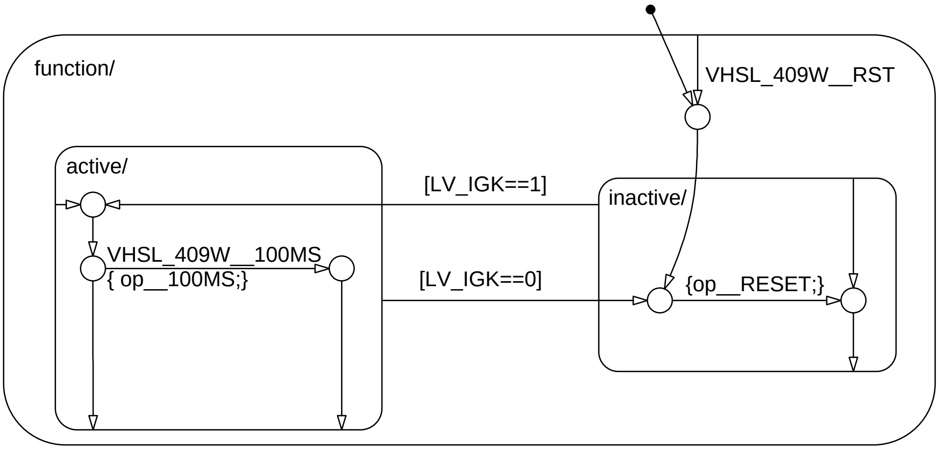
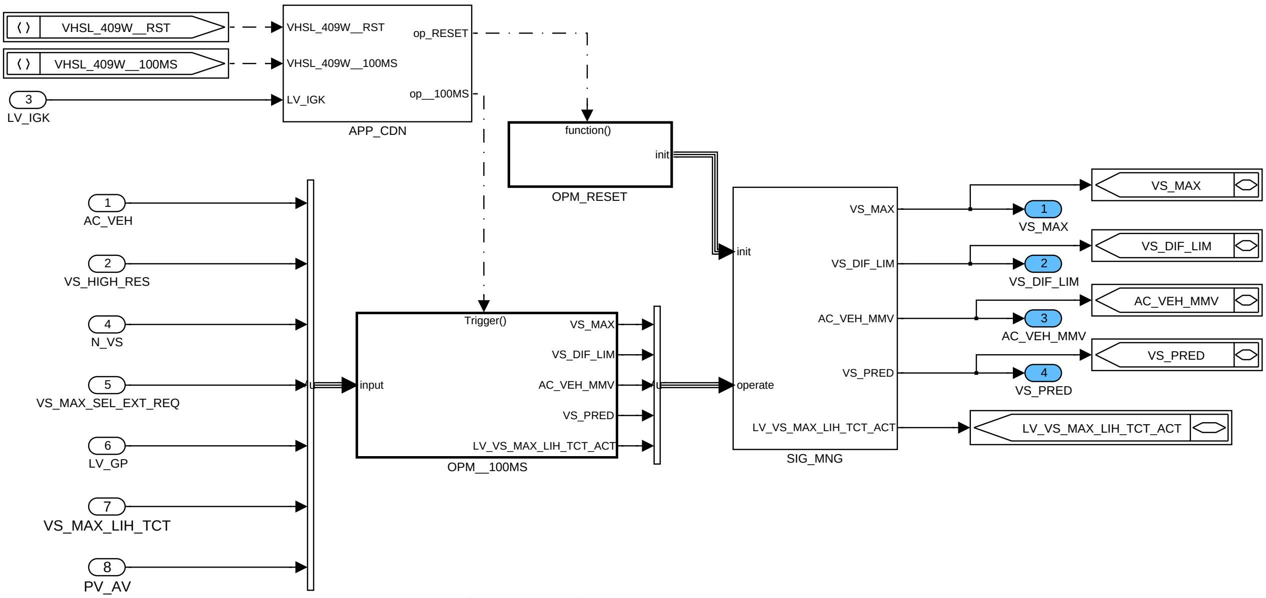


Figure 1 VHSL_M409W/ APP_CDN/ Chart

Function Description



SDA_SRS / SDA 4.0 21−Feb−2007

Figure 2 VHSL_M409W

68.2.1.1 SUBFUNCTION: OPM_RESET

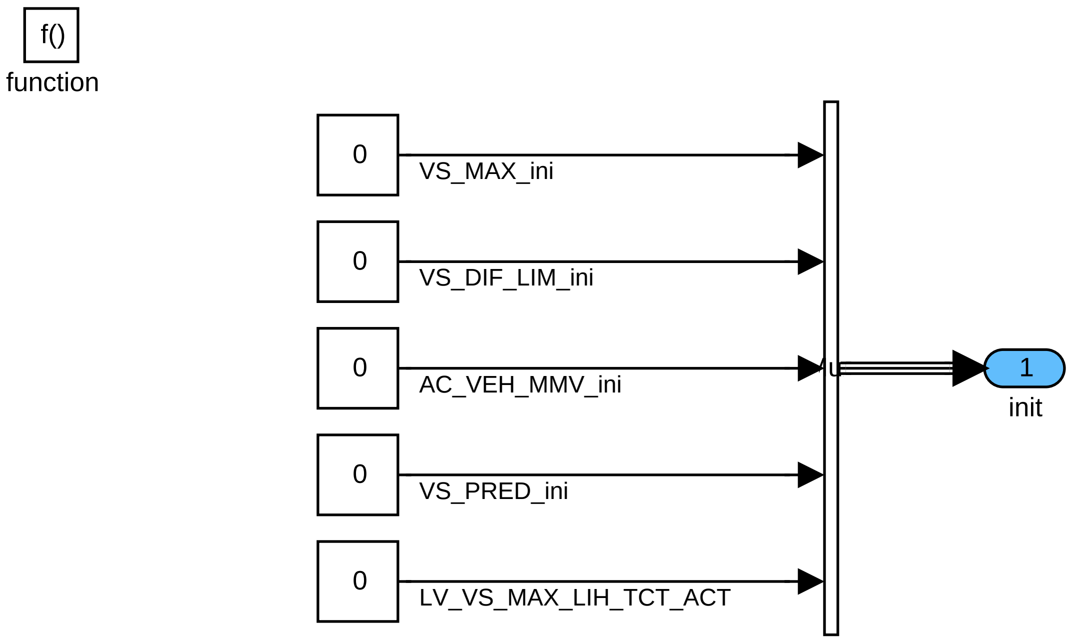


Figure 3 VHSL_M409W/ OPM_RESET

68.2.1.2 SUBFUNCTION: OPM__100MS

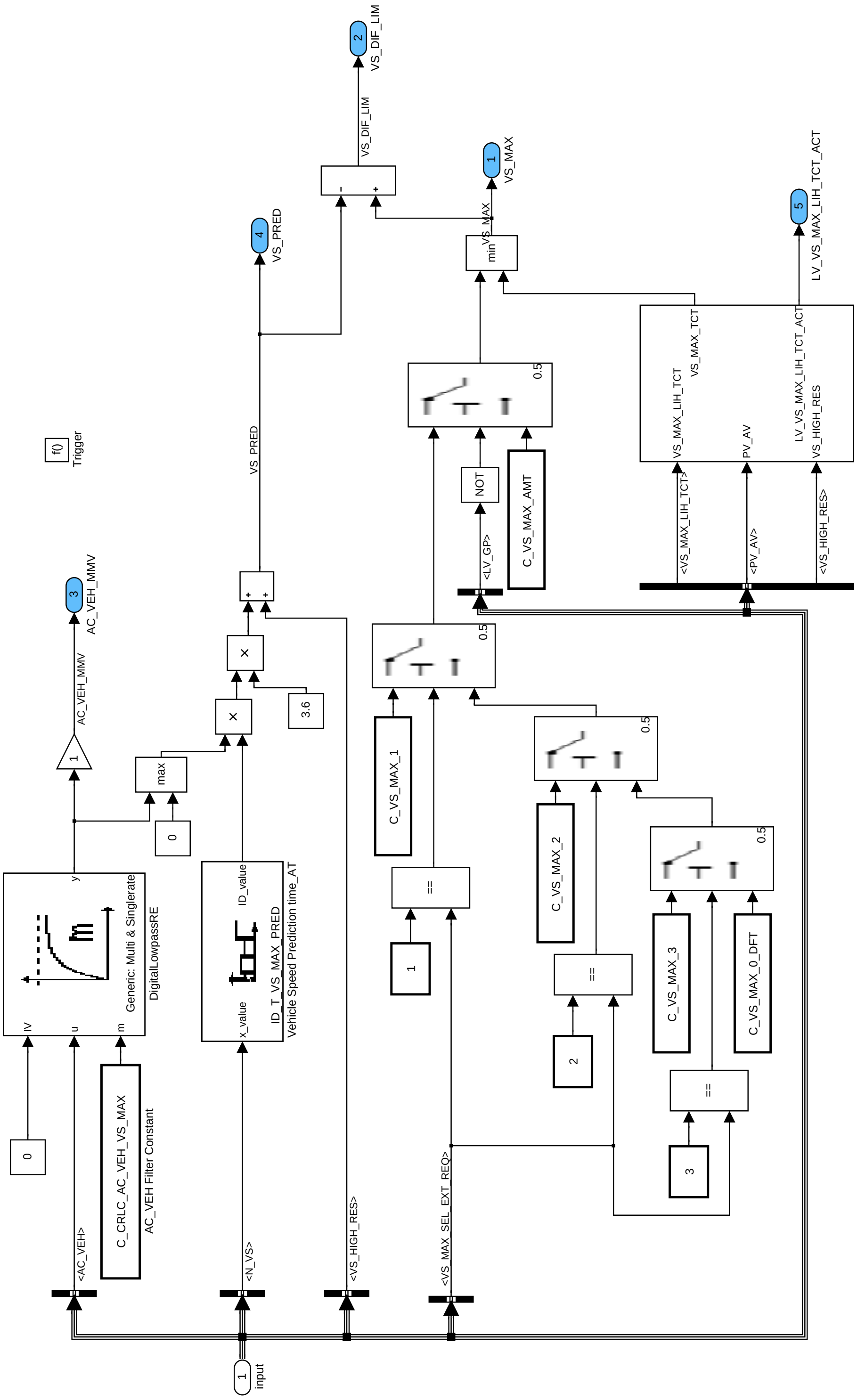


Figure 4 VHSL_M409W/ OPM__100MS

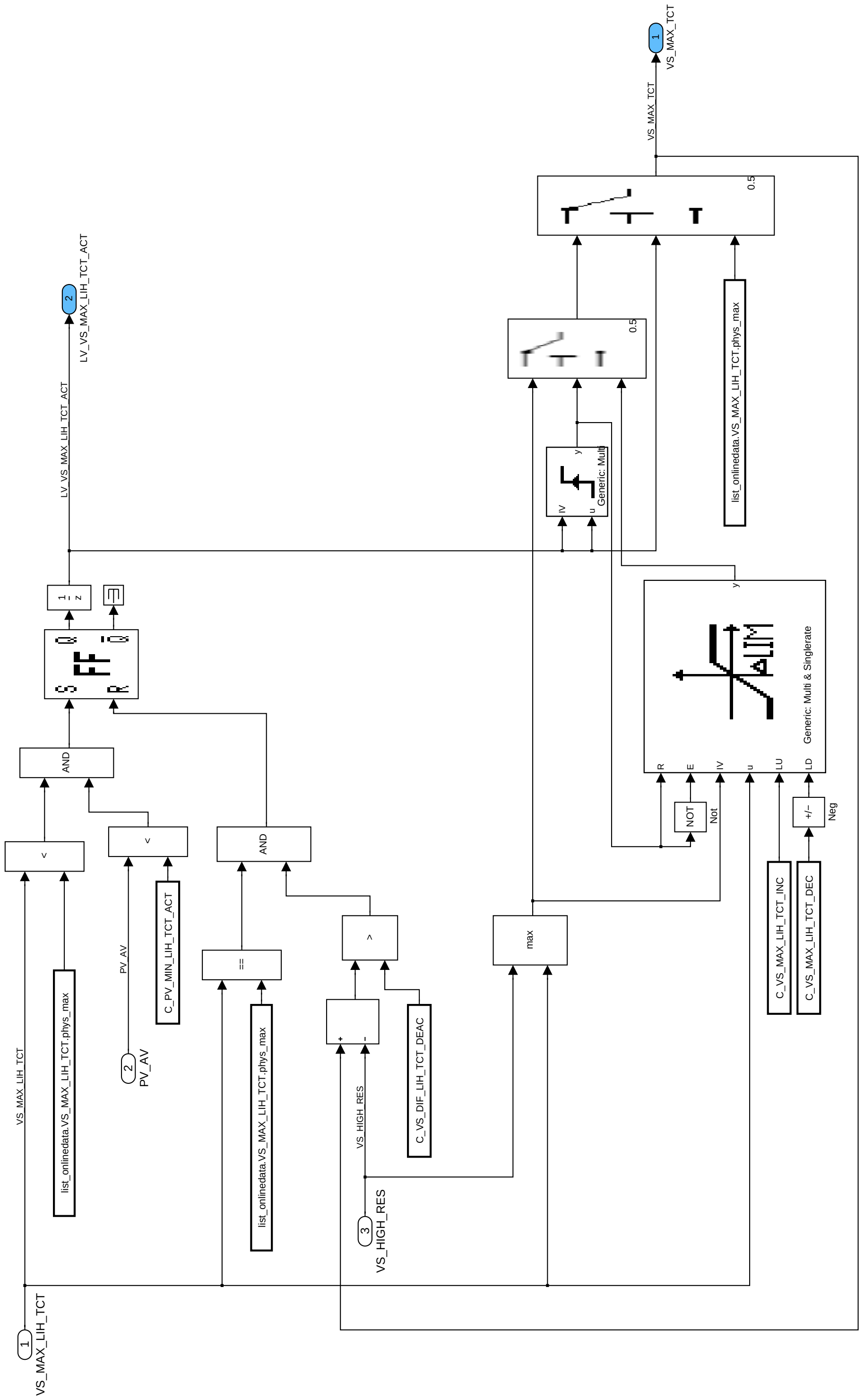


Figure 5 VHSL_M409W/ OPM__100MS/ CLC_VS_MAX_TCT

68.3 Vehicle speed limitation manager

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
STATE_VSL	O	0...4H	0...4	1	-
states of the vehicle limitation manager					

Input data:

LV_VS_MAX_VSL	LV_IGK		

68.3.1 FUNCTION PART: VHSL_MG00D

Application Condition

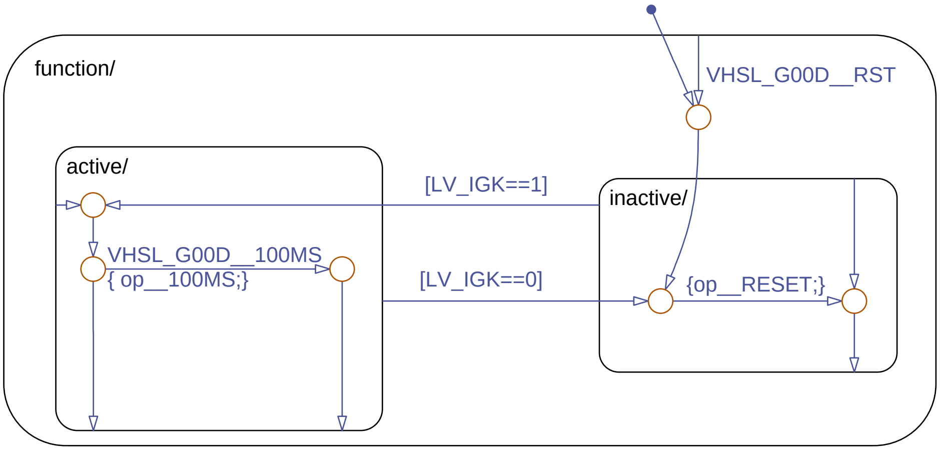


Figure 6 VHSL_MG00D/ APP_CDN/ Chart

Function Description

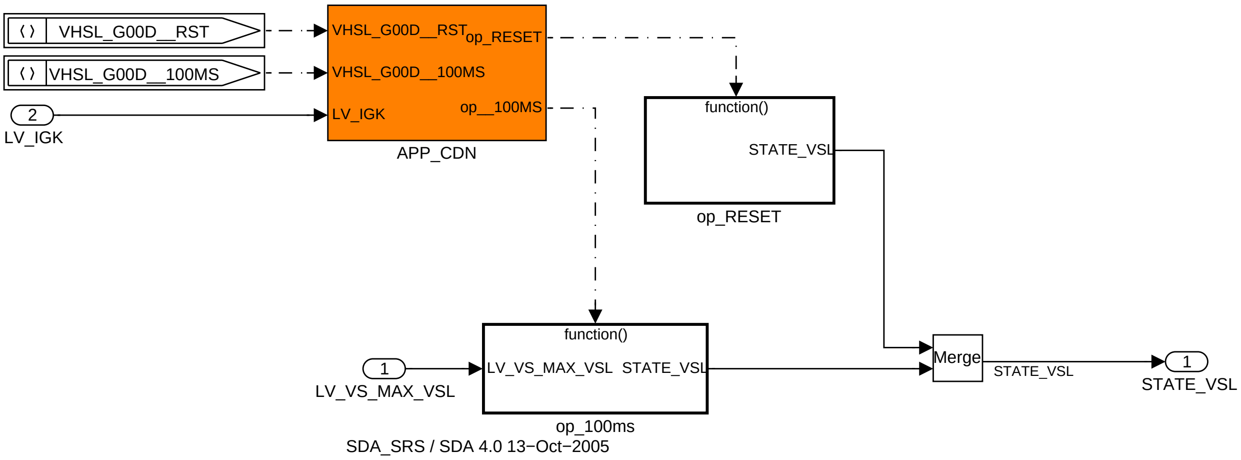


Figure 7 VHSL_MG00D

68.3.1.1 SUBFUNCTION: op_100ms

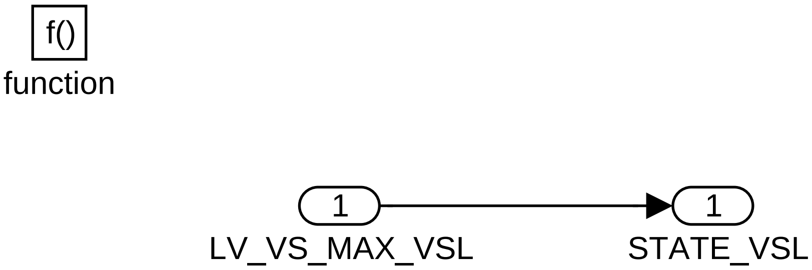


Figure 8 VHSL_MG00D/ OP_100MS

68.3.1.2 SUBFUNCTION: op_RESET

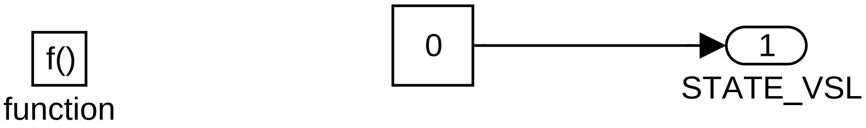


Figure 9 VHSL_MG00D/ OP_RESET

68.4 Vehicle speed limitation manager (Appl.Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_VS_MAX_VSL	O/V	0...1H	0...1	1	[-]
Boolean for vehicle speed limitation active					

Input data:

VS_MAX	VS_PRED	N_32	LV_IGK

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MIN_VS_MAX	1	0...FFH	0...8160	32	[rpm]
Minimum engine speed for VS_MAX function					
C_VS_MAX_HYS_OFF	1	0...7FFFH	0...327.67	0.01	[km/h]
Offset to turn off VS MAX limiter					
C_VS_MAX_HYS_ON	1	0...7FFFH	0...327.67	0.01	[km/h]
VS Offset to turn on VS MAX limiter					
LC_VSL_ACT_MAN	1	0...1H	0...1	1	[-]
Manual switch to enable vehicle speed limiter					
LC_VS_MAX_VSL_MAN	1	0...1H	0...1	1	[-]
Logical calibratable for the manually activation/deactivation of the vehicle speed limtation functionality					

68.4.1 General Information

This module deals with the activation of the fix vehicle speed limitation functionality. The function is switched on, LV_VS_MAX is set, when the predicted vehicle speed, VS_PRED exceeds the set point minus an offset, VS_MAX – C_VS_MAX_HYS_ON. On the other hand the function is reset, and turned off, when the predicted vehicle speed, VS_PRED drops below the set point minus a hysterisis VS_MAX – C_VS_MAX_HYS_OFF. An additional activation condition is that engine speed should be below of a threshold, C_N_MIN_VS_MAX. This threshold must be set to a low value away from the VS_MAX engine speed area. Application Condition

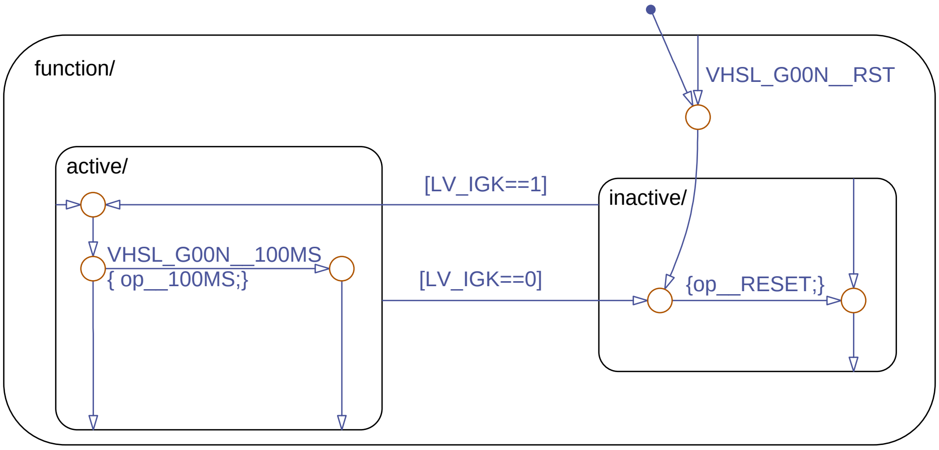


Figure 10 VHSL_MG00N/ APP_CDN/ Chart

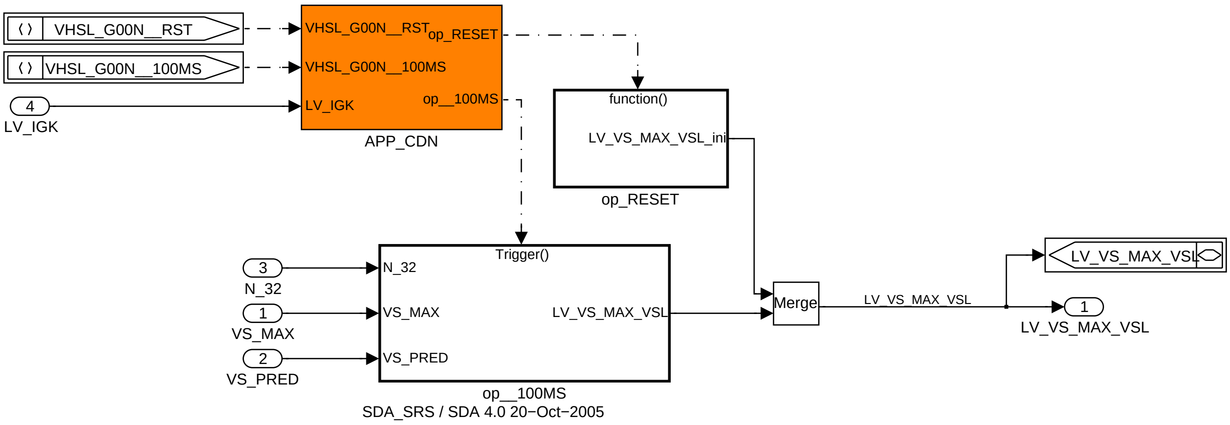


Figure 11 VHSL_MG00N

68.4.1.1 SUBFUNCTION: op_RESET

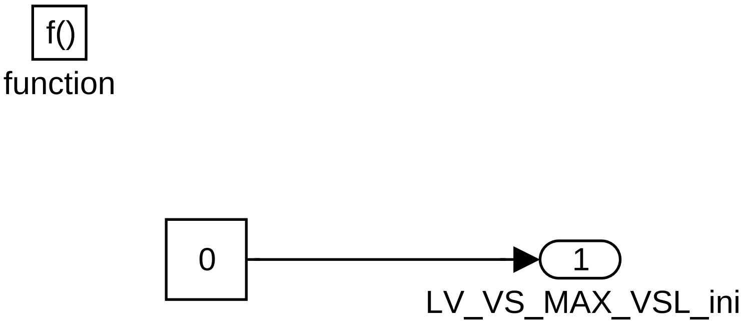


Figure 12 VHSL_MG00N/ op_RESET

68.4.1.2 VHSL_MG00N/OP__100MS

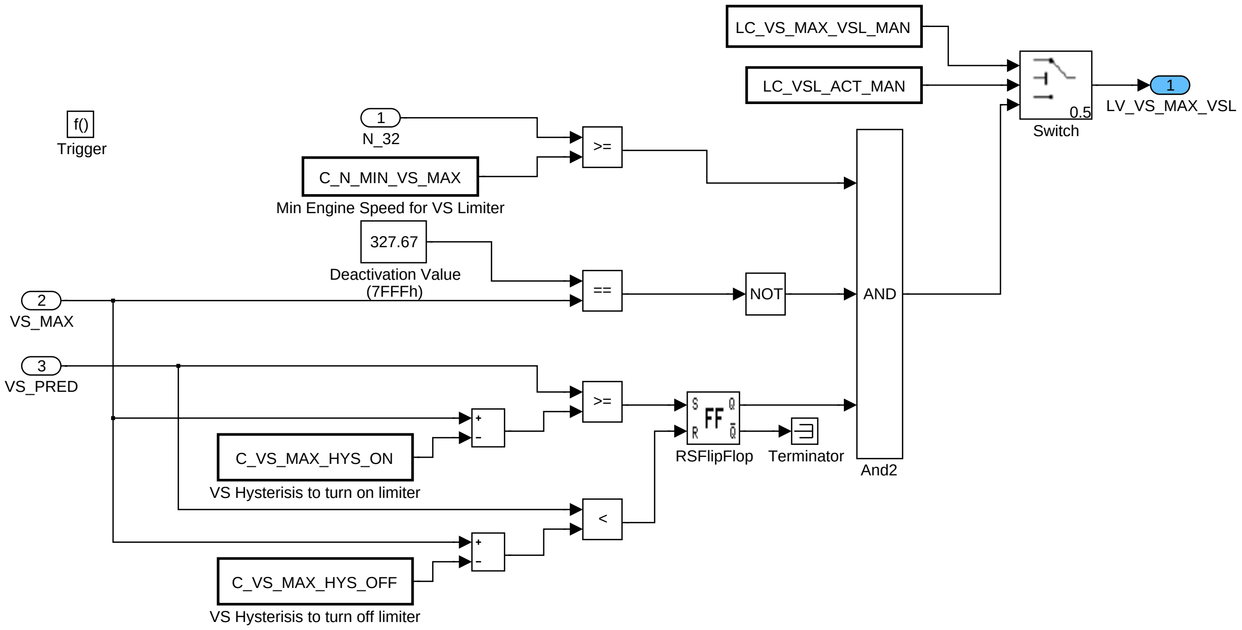


Figure 13 VHSL_MG00N/ op__100MS

68.5 Vehicle speed limitation controller

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQ_VS_MAX_DIF_I	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Torque output of VS_MAX-I-Controller					
TQ_VS_MAX_DIF_P	O/V	8000...7FFFH	-1.024E+3 ... 1.02397E+3	0.03125	Nm
Torque output of VS_MAX-P-Controller					

Input data:

AC_VEH_MMV	VS_DIF_LIM	FAC_VS_LIM_P_DYN_VSL	FAC_VS_LIM_I_DYN_VSL
LV_VS_MAX_VSL	N_VS	TQI_REQ_TRA	LV_IGK
TQI_VS_MAX			

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LGRD_TQ_VS_MAX_I	1	0...7FFFH	0...1.02397E+3	0.03125	Nm
Negative Gradient Limiter for VS_MAX I term					
C_MASS_VEH	1	0...FFH	0...2.55E+3	10	kg
Equivalent Vehicle Mass for acceleration calculation					

68.5.1 VHSL_MG009

This modules aims to limit the vehicle speed to C_VS_MAX. A PI controller is used to define the engine torque limitation. In order to minimise speed overshoot, a predicted vehicle speed is calculated and the I controller is initilised with a calculated torque reduction.

Application Condition

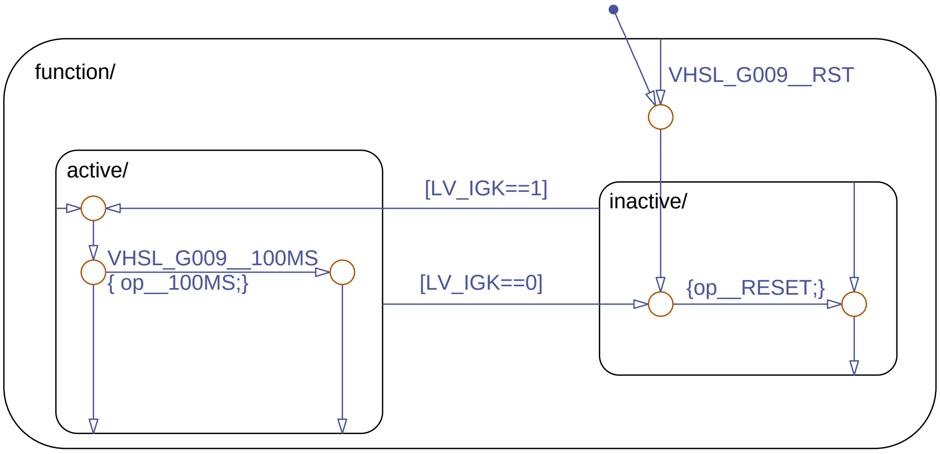


Figure 14 VHSL_MG009/ APP_CDN/ Chart

Function Description

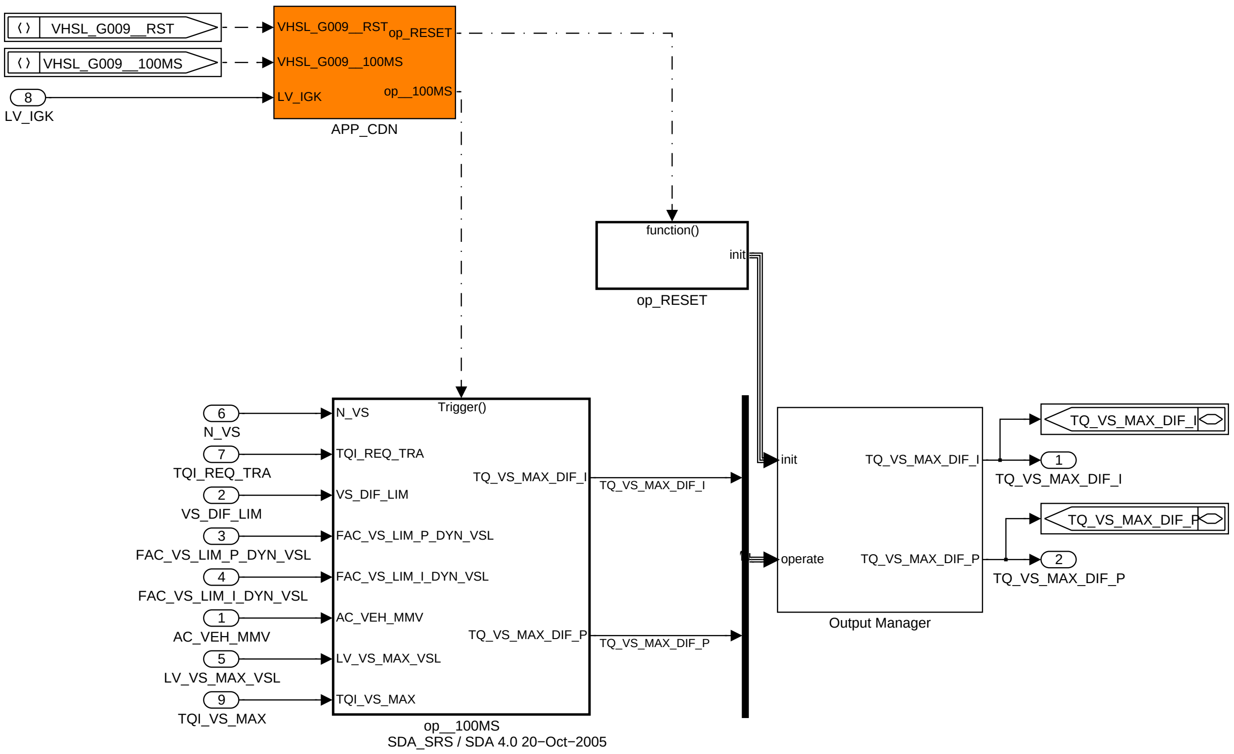


Figure 15 VHSL_MG009

68.5.1.1 SUBFUNCTION: Output Manager

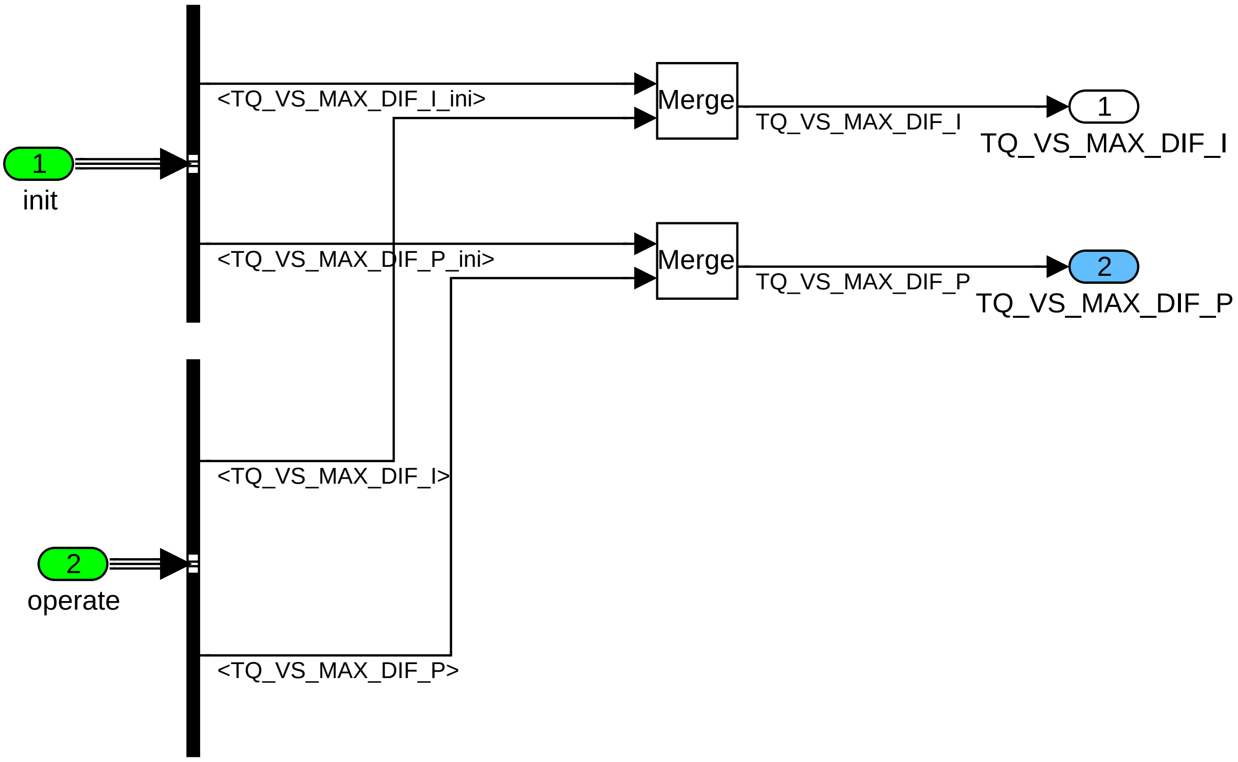


Figure 16 VHSL_MG009/ Output Manager

68.5.1.2 SUBFUNCTION: op_RESET

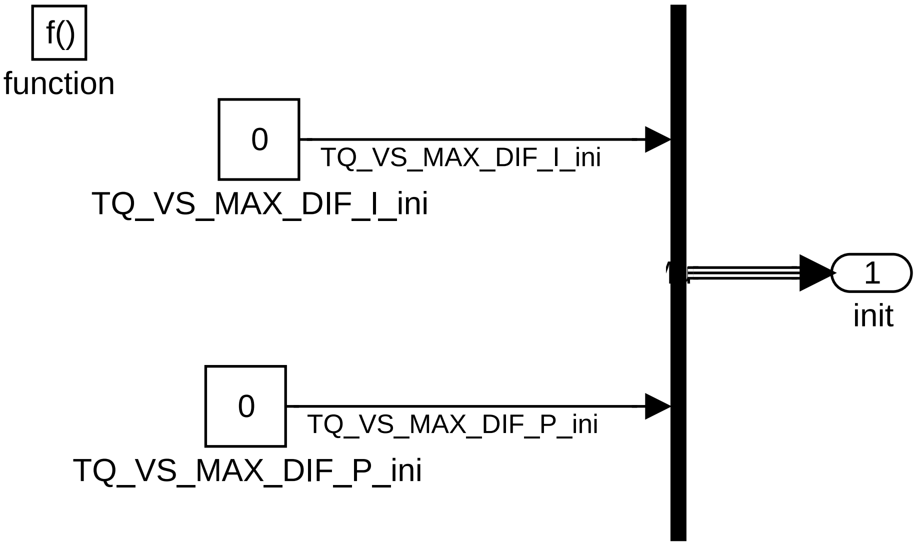


Figure 17 VHSL_MG009/ op_RESET

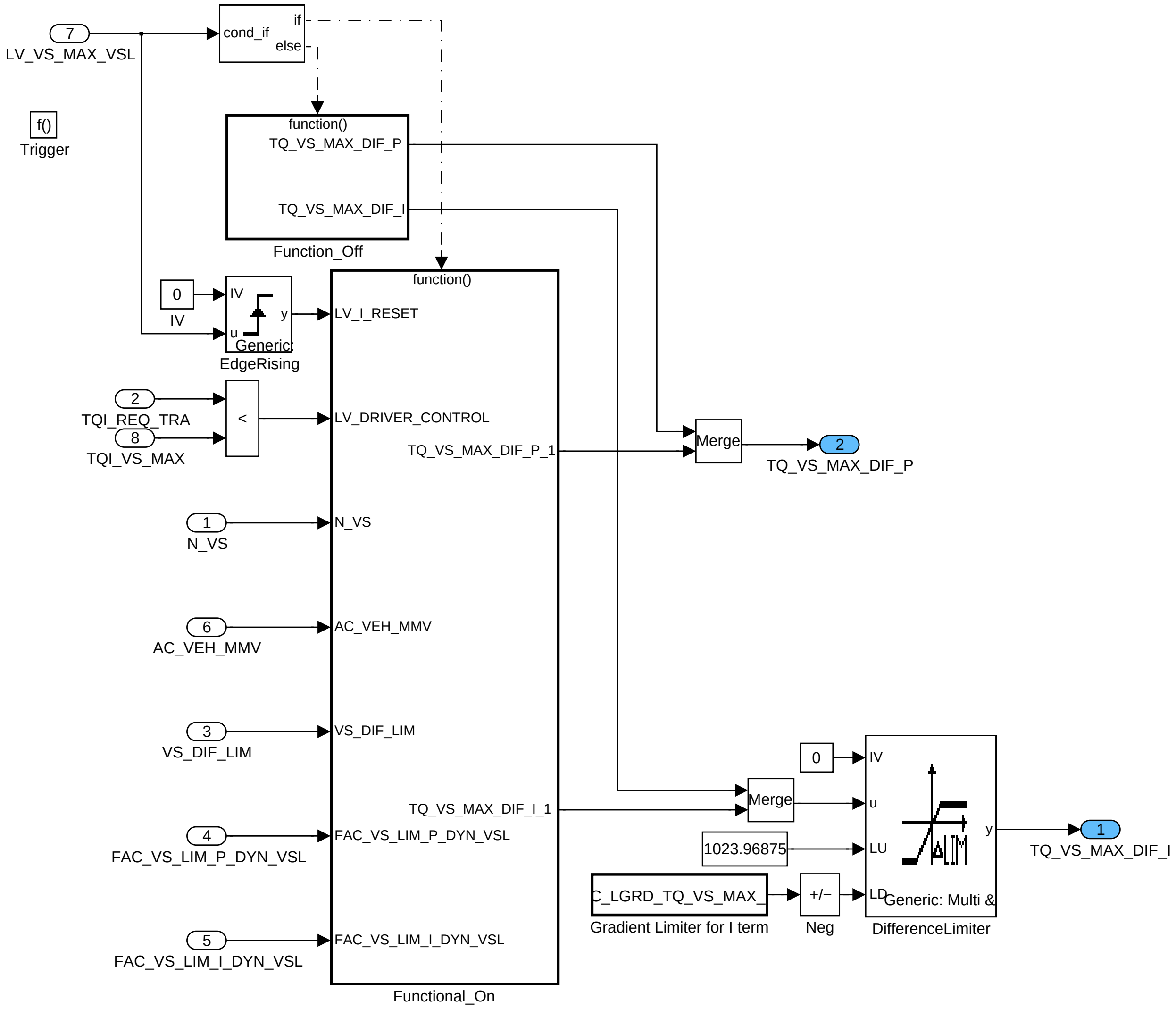


Figure 18 VHSL_MG009/ op__100MS

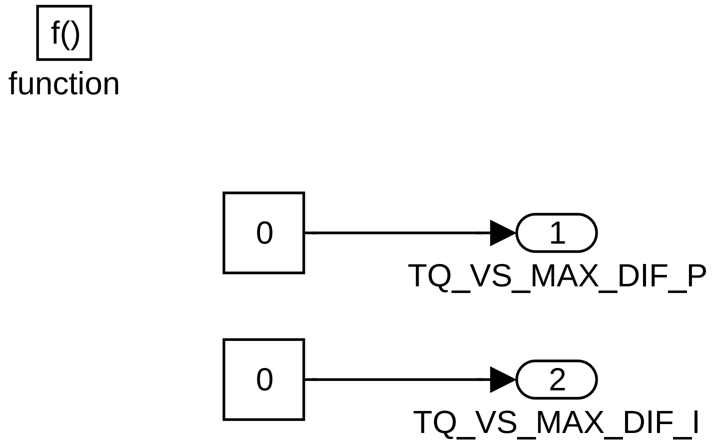


Figure 19 VHSL_MG009/ op__100MS/ Function_Off

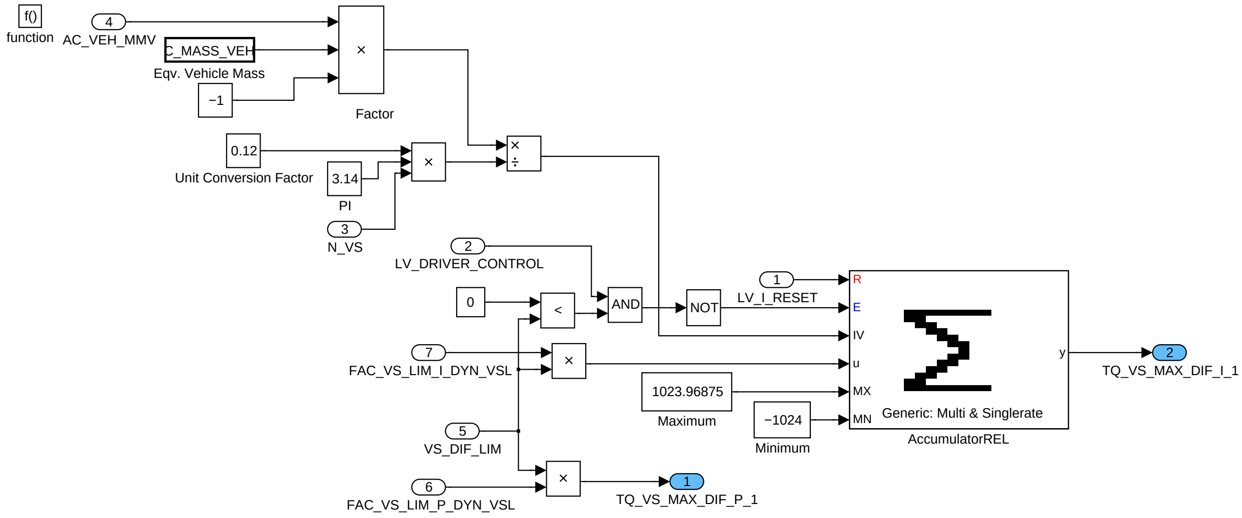


Figure 20 VHSL_MG009/ op__100MS/ Functional_On

68.6 Vehicle speed limitation controller (Appl. Inc.)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
FAC_VS_LIM_P_DYN_VSL	O/V	0...FFH	0...127.5	0.5	Nm/(km/h)
P-control parameter for VS limitation					
FAC_VS_LIM_I_DYN_VSL	O/V	0...7FFFH	0...255.992188	0.0078125	Nm/(km/h)
I-control parameter for VS limitation					
LV_FAC_CTL_DYN_CHG_VSL	V	0...1H	0...1	1	-
flag indicating a change in the control dynamics					

Input data:

LV_IGK	N_VS	VS_DIF_LIM	LV_VS_MAX_VSL

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_FAC_RAMP_VS_DEC_VSL	1	0...400H	0...1	9.76563E-4	-
ramp down constant (decrement)					
C_FAC_RAMP_VS_INC_VSL	1	0...400H	0...1	9.76563E-4	-
ramp up constant (increment)					
C_VS_THD_H_VSL	1	0...FFFFH	0...655.35	0.01	km/h
upper limit of hysteresis for the vehicle speed					
C_VS_THD_L_VSL	1	0...FFFFH	0...655.35	0.01	km/h
lower limit of hysteresis for the vehicle speed					
IP_FAC_VS_LIM_I_H_DYN_VSL	8	0...7FFFH	0...255.992188	0.0078125	Nm/(km/h)
LDPM_N_VS_LIM_VSL	8	0...FFH	0...255	1	rpm/(km/h)
Gain for the integral term in high dynamics					
IP_FAC_VS_LIM_I_L_DYN_VSL	8	0...7FFFH	0...255.992188	0.0078125	Nm/(km/h)
LDPM_N_VS_LIM_VSL	8	0...FFH	0...255	1	rpm/(km/h)
Gain for the integral term in low dynamics					
IP_FAC_VS_LIM_P_H_DYN_VSL	8	0...FFH	0...127.5	0.5	Nm/(km/h)
LDPM_N_VS_LIM_VSL	8	0...FFH	0...255	1	rpm/(km/h)
Gain for the proportional term in high dynamics					
IP_FAC_VS_LIM_P_L_DYN_VSL	8	0...FFH	0...127.5	0.5	Nm/(km/h)
LDPM_N_VS_LIM_VSL	8	0...FFH	0...255	1	rpm/(km/h)
Gain for the proportional term in low dynamics					

68.6.1 VHSL_MG00O

The control dynamic is determined by LV_FAC_CTL_DYN_CHG_VSL. The I and PD control parameters are calculated based on the dynamic behaviour of the controller.

Application Condition

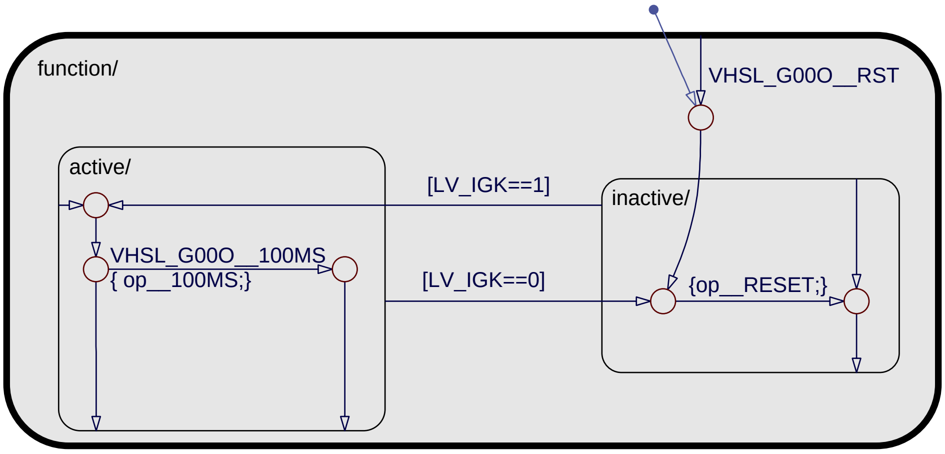


Figure 21 VHSL_MG00O/ APP_CDN/ Chart

Function Description

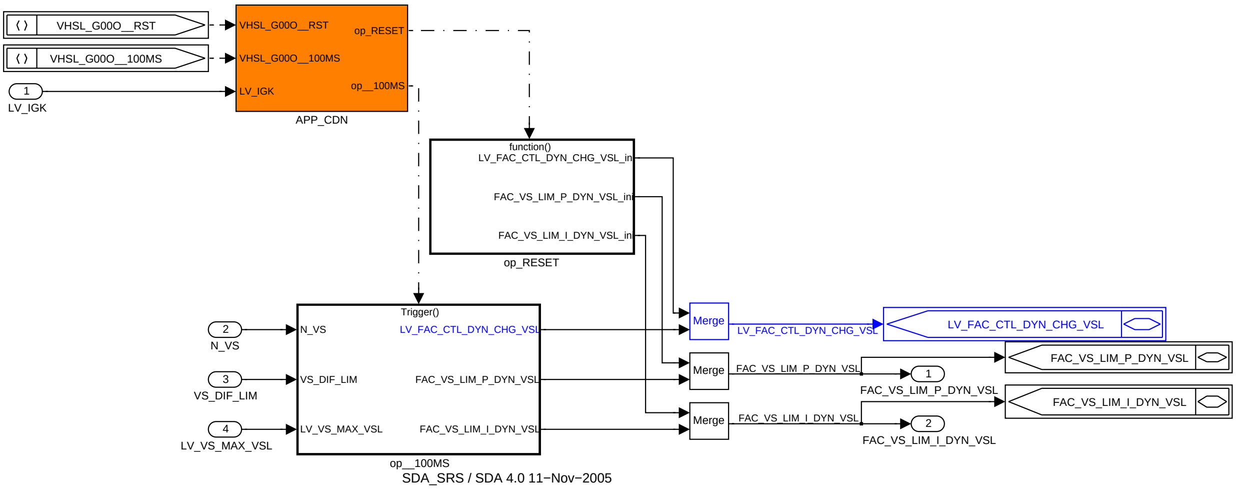


Figure 22 VHSL_MG00O

68.6.1.1 SUBFUNCTION: op_RESET

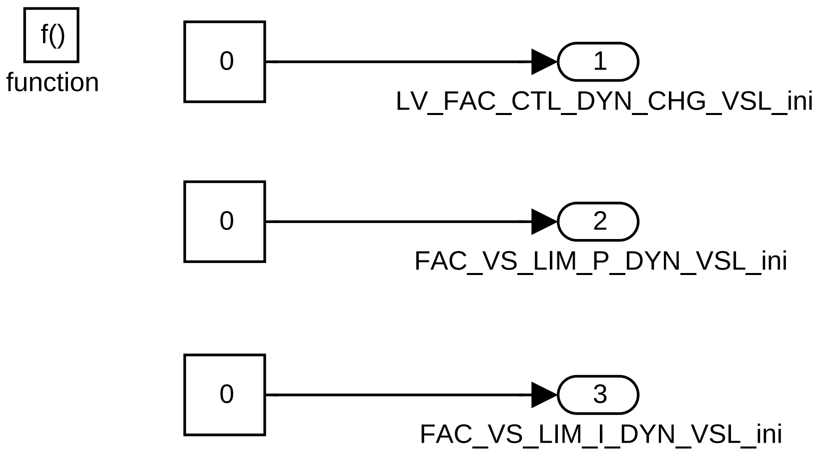


Figure 23 VHSL_MG00O/ op_RESET

68.6.1.2 SUBFUNCTION: op__100MS

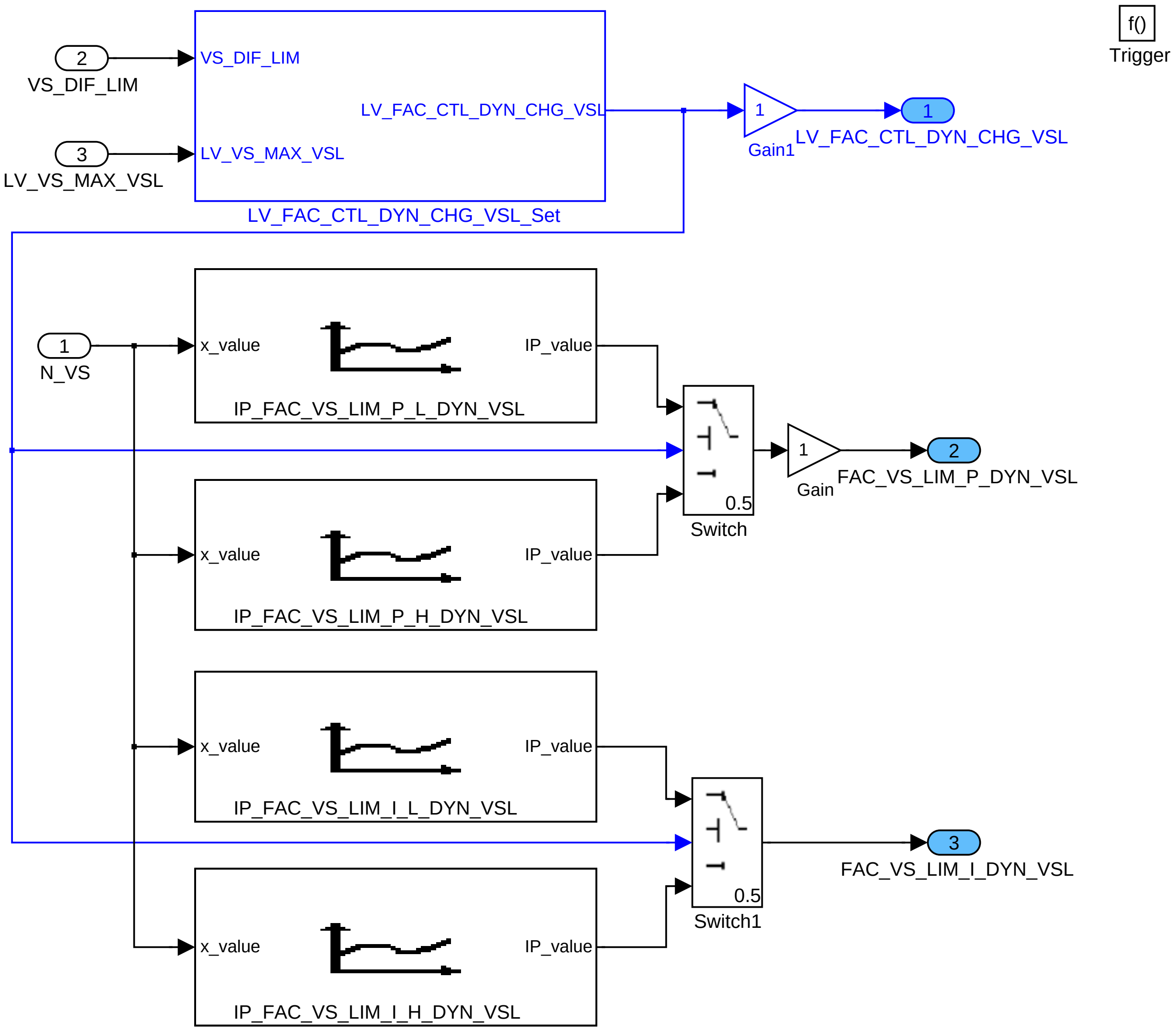


Figure 24 VHSL_MG00O/ op__100MS

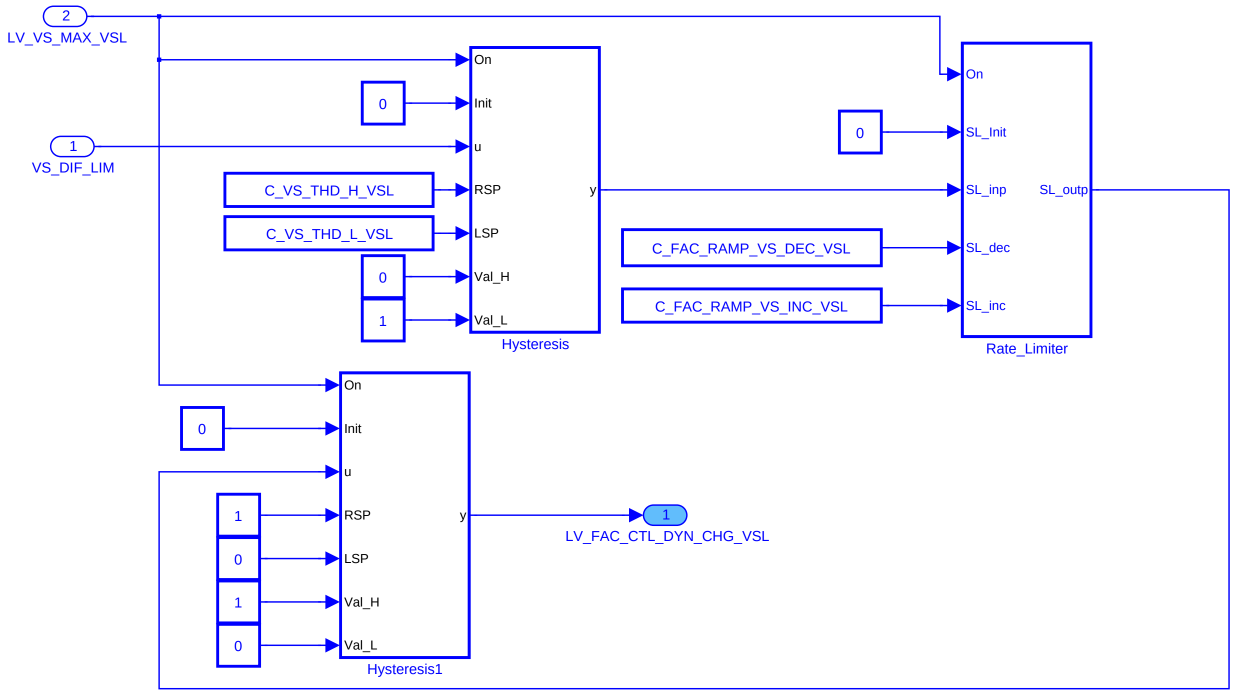


Figure 25 VHSL_MG00O/ op__100MS/ LV_FAC_CTL_DYN_CHG_VSL_Set

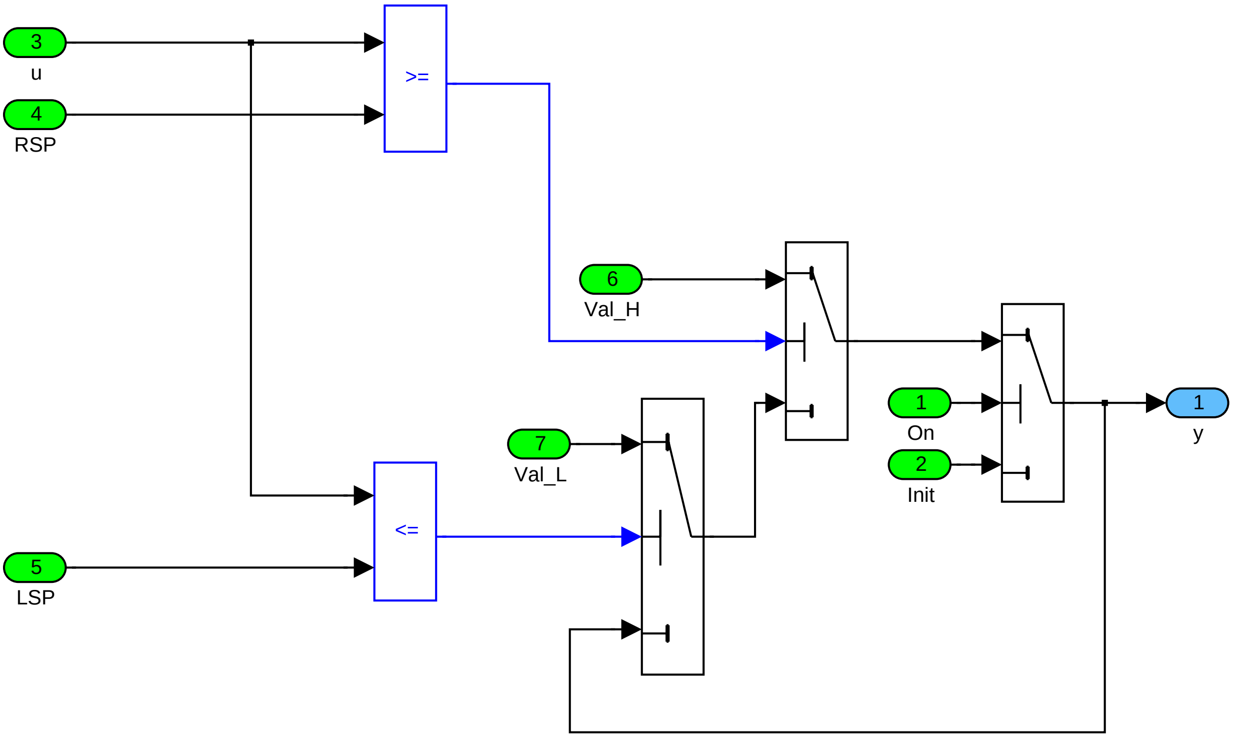


Figure 26 VHSL_MG00O/ op__100MS/ LV_FAC_CTL_DYN_CHG_VSL_Set/ Hysteresis

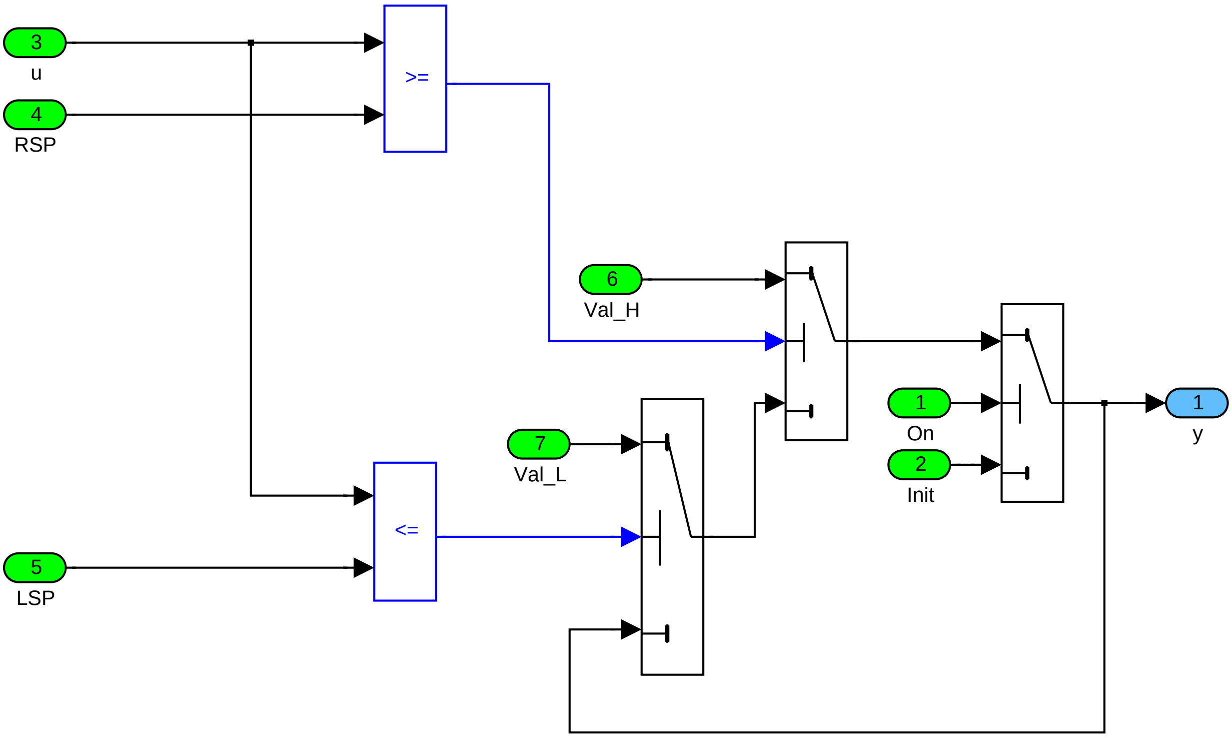


Figure 27 VHSL_MG00O/ op__100MS/ LV_FAC_CTL_DYN_CHG_VSL_Set/ Hysteresis1

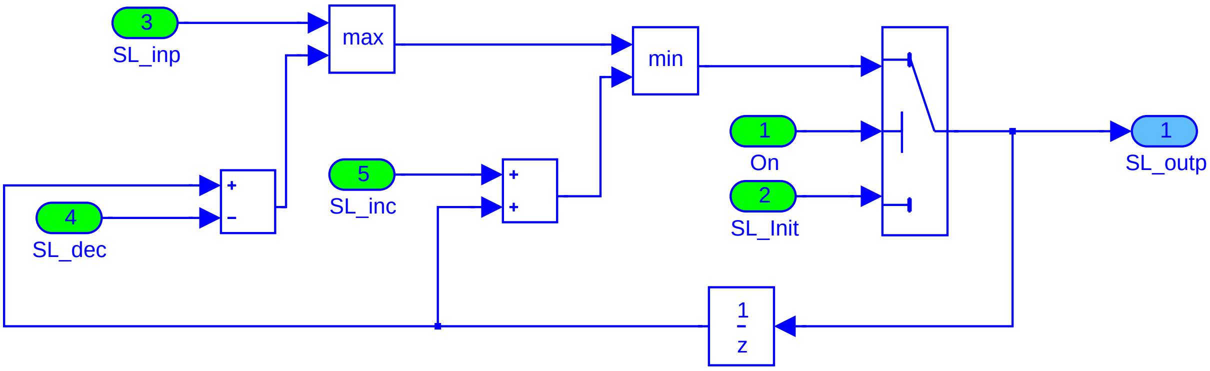


Figure 28 VHSL_MG00O/ op__100MS/ LV_FAC_CTL_DYN_CHG_VSL_Set/ Rate_Limiter

68.7 Vehicle speed limitation output

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
TQI_VS_MAX	O/V	0...7FFFH	0...1.02397E+3	0.03125004	Nm
Torque request for vehicle speed limitation					

Input data:

TQI_AV	TQI_REQ_TRA	TQI_BAS_MAX	LV_IGK
LV_VS_MAX_VSL	TQ_VS_MAX_DIF_I	TQ_VS_MAX_DIF_P	

68.7.1 VHSL_MG00P

This functionality limits the engine torque output by TQI_VS_MAX. The value of TQI_VS_MAX is initialised with the current engine output torque requirement. This will be calculated as the maximum of TQI_AV and minimum of TQI_BAS_MAX, TQI_REQ_TRA. This starting value of TQI_VS_MAX is then stored.

Application Condition

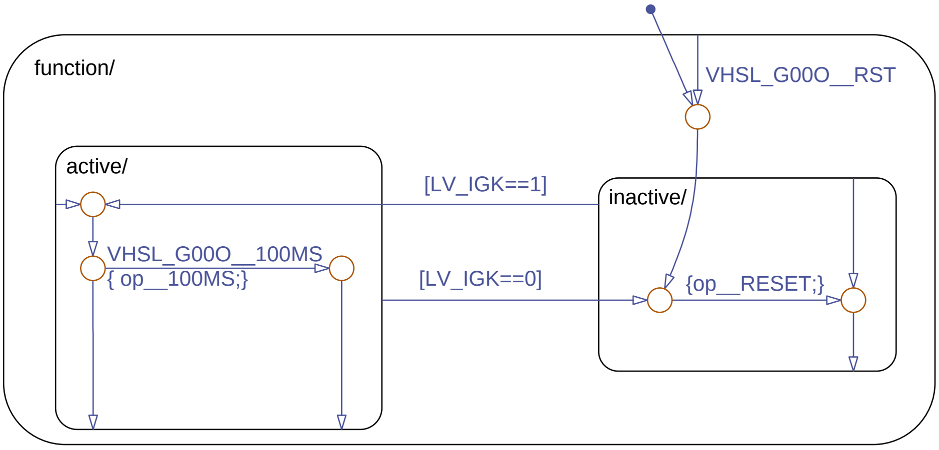


Figure 29 VHSL_MG00P/ APP_CDN/ Chart

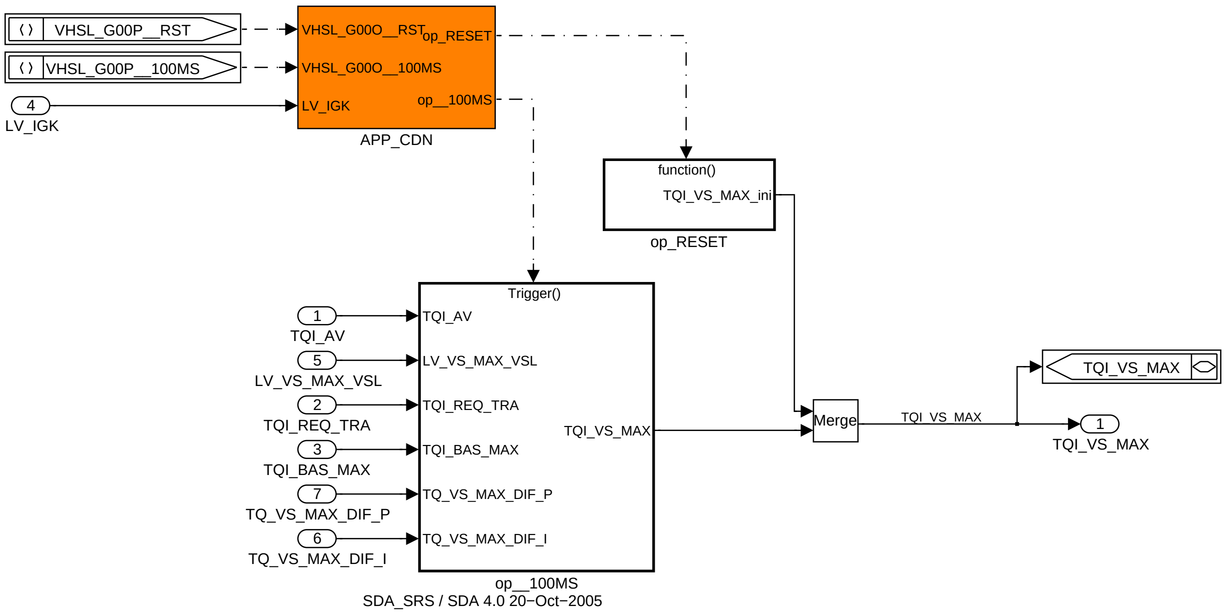


Figure 30 VHSL_MG00P

68.7.1.1 SUBFUNCTION: op_RESET

Figure 31 VHSL_MG00P/ op_RESET

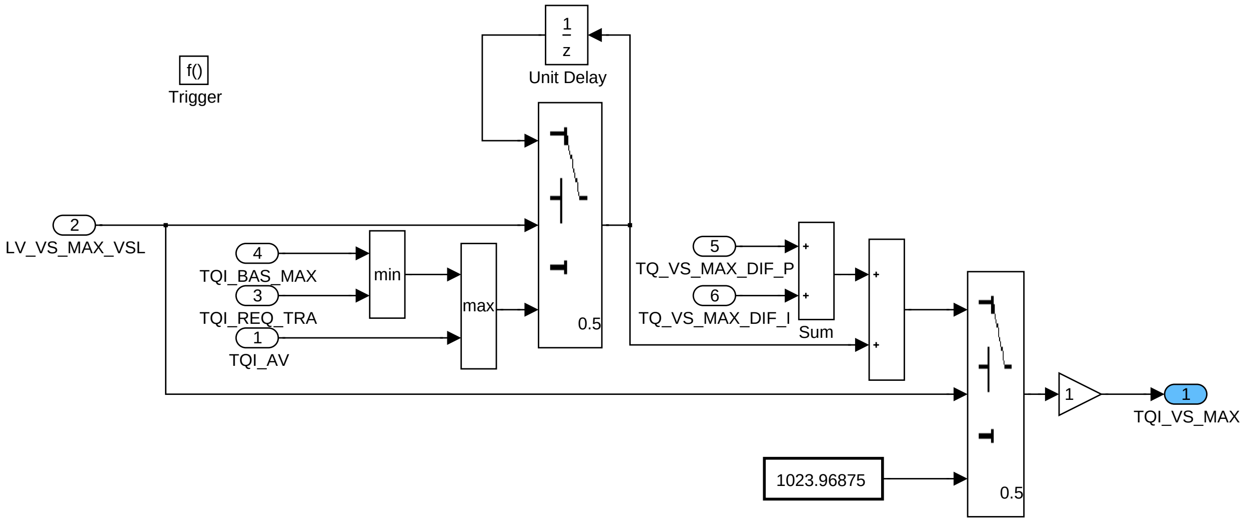


Figure 32 VHSL_MG00P/ op__100MS

68.8 Calibration hints for function Vehicle Speed Limitation (gen)

Calibration interfering functions

- torque model must be calibrated first
- of big importance is the calibration of *Torque Transient* function
- calibration of C_VS_FIL for the filtered vehicle speed of the function *Vehicle Motion Determination*
- Vehicle Motion Determination (VHMD), Gear Detection (TRSM) and Torque Transient (DRVB) should be calculated first

Calibration flowchart

The sequence of calibration is described by the order in *Calibrated methods*.

Calibration method

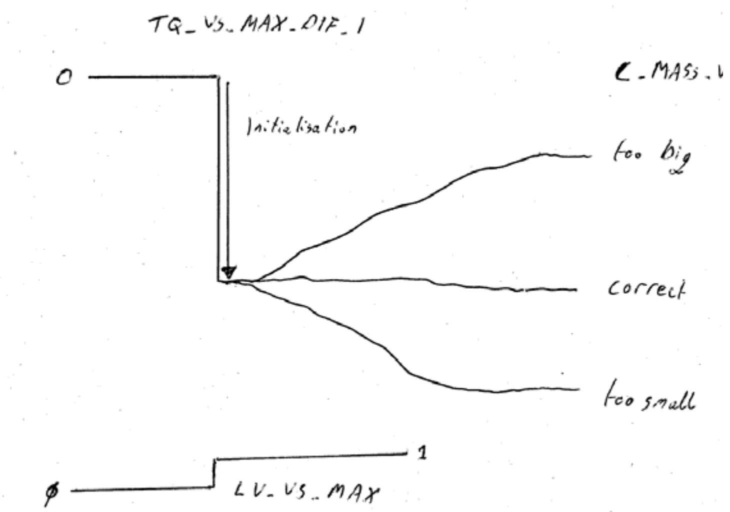
Vehicle speed limitation controller

C_LGRD_TQ_VS_MAX_I
C_MASS_VEH

68.8.1.1 Initialisation of I term

Initialisation of the I term is calculated from the vehicle acceleration, vehicle equivalent mass and gear ratio. The objective is to directly remove most of the vehicle acceleration force when the function is activated (LV_VS_MAX 0 -> 1).
Calibration procedure:

a) Set C_LGRD_TQ_VS_MAX_I to 0 (no limiting)
b) Accelerate to VS_MAX in various gears with constant pedal
c) Compare initialisation of I term with steady state value



When

steady state = initial value then vehicle mass is correct
steady state < initial value then vehicle mass is too small
steady state > initial value then vehicle mass is too high

If problems arise, check that AC_VEH_MMV =~ AC_VEH when function is initialised, if value is lagging then increase C_CRLC_AC_VEH_VS_MAX. Try to perform the test on a flat piece of road so that the road load is not changing. Start the acceleration well before VS_MAX, so that a quasi-steady state has been obtained.

Remember that C_MASS_VEH may be more than the actual vehicle mass due to the inertia of the engine and drivetrain. It may however be advantageous to have the initialisation value slightly small, so that a smooth entry to C_VS_MAX is obtained, i.e. acceleration is not completely stopped before C_VS_MAX is reached.

Reset C_LGRD_TQ_VS_MAX_I to 40 Nm, to avoid hard load changes when entering VS_MAX control.

Suggested initial data:
C_MASS_VEH = 1600 kg (use actual vehicle mass)
C_LGRD_TQ_VS_MAX_I = 40 Nm (/100ms)

68.8.1.2 Troubleshooting

1) Unstable at VS_MAX
    a. P / I terms too high (IP_FAC_VS_MAX_I / _P)
    b. ID_T_VS_MAX_PRED too large
    c. AC_VEH_MMV not filtered enough (C_CRLC_AC_VEH_VS_MAX too low)
2) Overshoot of VS_MAX
    a. P / I terms too small
    b. C_VS_MAX_HYS_ON too small
    c. Vehicle mass calibration (C_MASS_VEH) too low
    d. Prediction time (ID_T_VS_MAX_PRED) too small / short
    e. Difference limiter for I term too slow (increase C_LGRD_TQ_VS_MAX_I)
3) Toggling of LV_VS_MAX
    a. Hysterisis too small (increase C_VS_MAX_HYS_OFF)
4) Hard load change when function activated
    a. Difference limiter for I term too fast (decrease C_LGRD_TQ_VS_MAX_I)
    b. Vehicle mass calibration too high (C_MASS_VEH)
    c. ID_T_VS_MAX_PRED too large
    d. C_VS_MAX_HYS_ON too small

68.9 Calibration hints for function Vehicle Speed Limitation (cus)

Calibration interfering functions

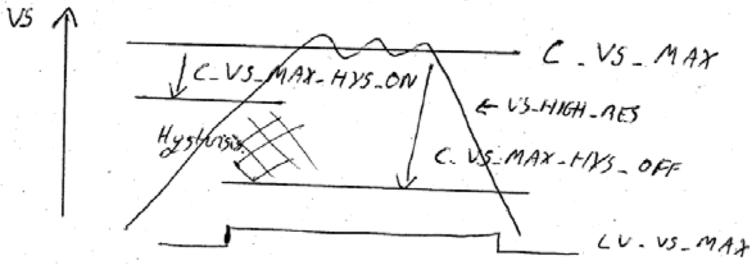
- torque model must be calibrated first
- of big importance is the calibration of *Torque Transient* function
- calibration of C_VS_FIL for the filtered vehicle speed of the function *Vehicle Motion Determination*
- Vehicle Motion Determination (VHMD), Gear Detection (TRSM) and Torque Transient (DRVB) should be calculated first

Calibration flowchart

The sequence of calibration is described by the order in *Calibrated methods*.

Calibration method

Vehicle speed limitation controller

C_VS_MAX_VSL	C_VS_MAX_VSL is the control setpoint. Suggested initial data: C_VS_MAX_VSL = 205 km/h for USA vehicles Note: The whole function can be deactivated by setting the setpoint (C_VS_MAX_VSL) to 327.67 km/h (0x7FFFh).
C_VS_MAX_HYS_ON C_VS_MAX_HYS_OFF	Set ON value to ensure function is active sufficiently early to avoid an overshoot. Set the OFF value sufficiently lower to avoid "chatter" – rapid on / off switching of the function.  Suggested initial data: C_VS_MAX_HYS_ON = 2 km/h (i.e. function on when VS_HIGH_PRED > 203 km/h) C_VS_MAX_HYS_OFF = 5 km/h (i.e function off when VS_HIGH_PRED < 200 km/h)
C_N_MIN_VS_MAX	Set C_N_MIN_VS_MAX to a low value away from the VS_MAX engine speed area. Suggested initial data: C_N_MIN_VS_MAX = 1500 rpm
C_CRLC_AC_VEH_VS_MAX	*Filtered Vehicle Speed Acceleration* AC_VEH ->1st order Filter -> AC_VEH_MMV Set the filter constant (C_CRLC_AC_VEH_VS_MAX) to give a smooth, but still fast reacting AC_VEH_MMV in the area of VS_MAX. Check with hard load changes (0 to 100% pedal etc). Suggested initial data: C_CRLC_AC_VEH_VS_MAX = 0.2

ID_T_VS_MAX_PRED	*Vehicle Speed Prediction* VS_PRED = (AC_VEH_MMV * T_VS_MAX_PRED) + VS_HIGH_RES Calibrate ID_T_VS_MAX_PRED to give sufficient time for vehicle speed to be stabilised after function turned on, without having an overshoot. Suggested initial data: LDP_N_VS_VS_MAX = [16 20 26 32 40 55 105 220]; ID_T_VS_MAX_PRED = 1 second (perhaps more for lower gears)
IP_FAC_VS_LIM_P_L_DYN_VSL IP_FAC_VS_LIM_P_H_DYN_VSL IP_FAC_VS_LIM_I_L_DYN_VSL IP_FAC_VS_LIM_I_H_DYN_VSL	*Calibration of P and I control factor* Calibrate IP_FAC_VS_LIM_P_L(H)_DYN_VSL and IP_FAC_VS_LIM_I_L(H)_DYN_VSL to give good stable control of the vehicle speed. Check correct control is obtained in the following situations: entry to VS_MAX (also in downhill conditions etc.) gear changes at VS_MAX load changes (via pedal) at VS_MAX road load changes at VS_MAX (e.g. hills) It may be worth applying a empirical tuning method (e.g. Ziegler-Nichols), or at the least, checking the stability limits of the system (by further increasing P & I term), before finalising these values. Suggested initial data: IP_FAC_VS_LIM_I_L(H)_DYN_VSL = 4 Nm (/ (km/h).100ms) IP_FAC_VS_LIM_P_L(H)_DYN_VSL = 40 Nm (/ (km/h))
LC_VSL_ACT_MAN	Logical calibratible for long time activation/deactivation of the vehicle speed limitation function.
LC_VS_MAX_VSL_MAN	Logical calibratible for long time activation/deactivation of the vehicle speed limitation function in case LC_VSL_ACT_MAN ==1. LC_VS_MAX_VSL_MAN =1 then limitation always active LC_VS_MAX_VSL_MAN =1 then limitation inhibited
C_VS_THD_H_VSL	
C_VS_THD_L_VSL	
C_FAC_RAMP_VS_DEC_VSL	
C_FAC_RAMP_VS_INC_VSL	

69 Vehicle stability and traction

Table of Contents

Index

69.1 AGGR VHST adaptation

69.1.1 Outputs for BMW functions which are not defined as VHST exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Md_na_ars	O/V	8000...7FFFH	-3276.8...3276.7	0.1	[Nm]
Torque losses active rolling stabilisation					
Mdi_asr_l	O/V	0...2710H	0...1000	0.1	[Nm]
Torque decrement for slow torque intervention during TCS					
Mdi_asr_s	O/V	0...2710H	0...1000	0.1	[Nm]
Torque decrement for fast torque intervention during TCS					
Mdi_msr_l	O/V	0...2710H	0...1000	0.1	[Nm]
Torque increase for slow torque intervention during MSR					
Mdi_msr_s	O/V	0...2710H	0...1000	0.1	[Nm]
Torque increase for fast torque intervention during MSR					

Input data:

TQI_ASR_SLOW	TQI_ASR_FAST	TQI_MSR_SLOW	TQI_MSR_FAST
TQ_LOSS_ARS			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: The really possible physical limits of the outputs are different to the specified values due to the input data attributes.

Application conditions:

Initialisation: 0

Recurrence : 10 ms

Activation: every engine state

Formula section:

Remark: all formulas are valid in a **physical** meaning

Md_na_ars = TQ_LOSS_ARS

Mdi_asr_l = TQI_ASR_SLOW

Mdi_asr_s = TQI_ASR_FAST

Mdi_msr_l = TQI_MSR_SLOW

Mdi_msr_s = TQI_MSR_FAST

69.2 Torque Request for Traction Control (TCS)

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_TCS_DISABLE	V/O	0...1H	0...1	1	[-]
Logical variable for disabling TCS intervention due to EMS error					
LV_TCS_DISABLE_CAN	V/O	0...1H	0...1	1	[-]
Logical variable for disabling TCS intervention due to EMS CAN error					
LV_ASR_ENA	V	0...1H	0...1	1	[-]
Logical variable for ASR intervention enabling					
LV_TCS_ENA	V	0...1H	0...1	1	[-]
Logical variable for basically TCS intervention enabling					
LV_MSR_ENA	V	0...1H	0...1	1	[-]
Logical variable for MSR intervention enabling					
STATE_TQ_TCS_PLAUS	V/O	0...FFH	0...255	1	[-]
Bitwise coded State for TCS intervention state					
TQI_ASR_FAST_REQ	V	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque decrement for fast torque intervention during TCS					
TQI_ASR_SLOW_REQ	V	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque decrement for slow torque intervention during TCS					
TQI_MSR_FAST_REQ	V	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for fast torque intervention during TCS					
TQI_MSR_SLOW_REQ	V	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for slow torque intervention during TCS					
TQI_ASR_FAST_REQ_1	O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque decrement for fast torque intervention during TCS					
TQI_ASR_SLOW_REQ_1	O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque decrement for slow torque intervention during TCS					
TQI_MSR_FAST_REQ_1	O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for fast torque intervention during TCS					
TQI_MSR_SLOW_REQ_1	O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for slow torque intervention during gear shift					
TQI_ASR_FAST	V/O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque decrement for fast torque intervention during TCS					
TQI_ASR_SLOW	V/O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque decrement for slow torque intervention during TCS					
LV_ASR_ACT	V/O	0...1H	0...1	1	[-]
Logical variable for torque intervention due to TCS					
LV_ASR_LIH	V	0...1H	0...1	1	[-]
Logical variable for torque intervention LIH due to TCS					
T_ASR_LIH_CTR	V	0...FFH	0...2.55	0.01	[s]
Delay time					
TQI_MSR_FAST	V/O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for fast torque intervention during MSR					
TQI_MSR_SLOW	V/O	0...7FFFH	0...1023.97	3.13E-02	[Nm]
Torque increase for slow torque intervention during MSR					
LV_MSR_ACT	V/O	0...1H	0...1	1	[-]
Logical variable for torque intervention due to MSR					
LV_MSR_LIH	V/O	0...1H	0...1	1	[-]
Logical variable for torque intervention LIH due to MSR					
T_MSR_LIH_CTR	V	0...FFH	0...2.55	0.01	[s]
Delay time					
SF_TQD	V/O	0...3H	0...3	1	[-]
Status flag driving torque at clutch					

Input data:

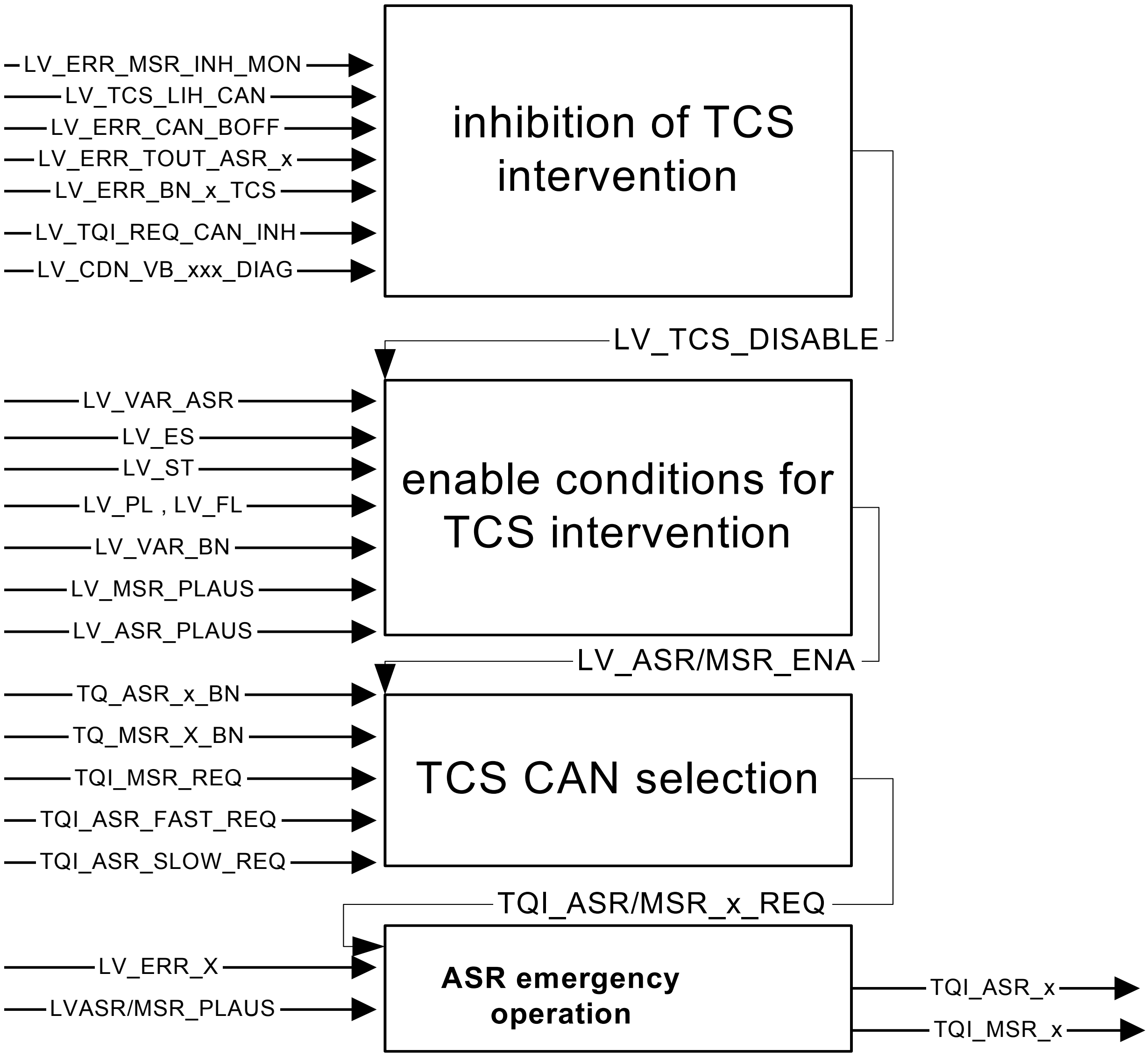
IGA_MIN_H_RNG	LV_VAR_ASR	LV_MTC_CUR_OFF	TQ_ASR_FAST_DEC_BN
IGA_SP_H_RNG	LV_ERR_MSR_INH_MON	LV_OFF_MTC_MON	TQ_ASR_SLOW_DEC_BN
LV_MSR_PLAUS	LV_VAR_BN	LV_PL	TQ_LOSS
LV_ASR_PLAUS	LV_ERR_TOUT_ASR_1	LV_ST	TQ_MSR_FAST_INC_BN
LV_DT	LV_ERR_TOUT_ASR_3	LV_TCS_LIH_CAN	TQ_MSR_SLOW_INC_BN
LV_ERR_BN_TCS	LV_ERR_CAN_BOFF	LV_TQ_ASR_REQ	TQ_TCS_FAST_BN
LV_ERR_BN_TQ_TCS	LV_ES	LV_TQ_LIM_INTV	TQ_TCS_SLOW_BN
LV_ERR_BN_VS_TCS	LV_PU	LV_TQ_MSR_REQ	TQI_ASR_FAST_REQ_CAN
LV_TQI_REQ_CAN_INH	LV_IGK	STATE_ETC_LIH	TQI_ASR_SLOW_REQ_CAN
LV_CDN_VB_CAN_TQ_DIAG	LV_SF_TQD_MON	LV_DI_TQ_REQ_CAN_MPI_GDI	TQI_MSR_REQ_CAN

FUNCTION DESCRIPTION:

Traction control function avoids wheel spinning on e.g. icy surfaces by reducing the engine torque in addition to brake the wheel. The ECU permanently checks, if a probable traction control torque intervention request can be executed, by checking, if an error at one or several of the relevant components for torque intervention occurred (LV_TCS_DISABLE). If no error is recognized and an actual traction control request is received, a check is done, if at actual engine operating condition a torque intervention allowed (LV_ASR_ENA or LV_MSR_ENA). Both torque request values, for fast and slow torque intervention are handed over to relevant torque coordination functions to adjust the engine torque.

MSR function avoids the vehicle from skidding on e.g. icy surfaces by reducing the drag torque of the engine at trailing throttle and trailing throttle fuel cut off. For to reach a higher lateral stability of the wheels the indicated engine torque is increased to reduce wheel slip.

Signal flow diagram:



69.2.1 inhibition TCS intervention

FUNCTION DESCRIPTION:

If the EMS detects a safety relevant error at throttle the logical variable LV_TCS_DISABLE is set to 1 in order to indicate a static EMS error.

Application conditions:

Initialisation:	LV_TCS_DISABLE = 0 ; LV_TCS_DISABLE_CAN = 0 at reset
Recurrence:	10 ms
Activation:	every engine operating state
Deactivation:	-

Formula section:

```
if      LV_ERR_CAN_BOFF        = 1
   or   LV_ERR_TOUT_ASR_1      = 1
   or   LV_ERR_TOUT_ASR_3      = 1
   or   LV_ERR_BN_TQ_TCS       = 1
   or   LV_ERR_BN_VS_TCS       = 1
   or   LV_ERR_BN_TCS          = 1
   or   LV_CDN_VB_CAN_TQ_DIAG = 0
then    LV_TCS_DISABLE_CAN     = 1
else    LV_TCS_DISABLE _CAN    = 0
endif

if      LV_TQI_REQ_CAN_INH     = 1
   or   LV_DI_TQ_REQ_CAN_MPI_GDI = 1
   or   LV_ERR_MSR_INH_MON    = 1
   or   LV_TCS_LIH_CAN         = 1
   or   LV_TCS_DISABLE_CAN     = 1
then    LV_TCS_DISABLE         = 1
else    LV_TCS_DISABLE         = 0
endif
```

69.2.2 Enable conditions for TCS torque intervention

FUNCTION DESCRIPTION:

General information:

Depending on the configuration and plausible torque interfaces an inc (MSR). or decreaced (ASR) torque intervention is possible.

The State STATE_TQ_TCS_PLAUS indicates the interrupted condition witch is responsible for enable Bit = 0. The [x] Value shows the HEX adress inside the Bitwise coded State.

Application conditions:

Initialisation:	LV_ASR/TCS/MSR_ENA = 0; *STATE_TQ_TCS_PLAUS = 0* at reset
Recurrence:	10 ms
Activation:	every engine operating state

Formula section:

```
If          LV_VAR_ASR = 1          and
            LV_TCS_DISABLE = 0      and
            LV_ES=0                 and
            LV_ST = 0               and
Then        LV_TCS_ENA = 1
Else        LV_TCS_ENA = 0
Endif

IF(1) LV_VAR_BN = 0

Then(1)if(2)    LC_ASR_DISABLE = 0
            and LV_TCS_ENA = 1
            and [ LV_PL = 1   or   LV_PU = 1]
            and LV_ASR_PLAUS = 1
            and LV_TQ_ASR_REQ = 1
    then(2)     LV_ASR_ENA = 1
    else(2)     LV_ASR_ENA = 0

else(1) if(3)   LC_ASR_DISABLE = 0
            and LV_TCS_ENA = 1
            and [ LV_PL = 1   or   LV_PU = 1]
            and TQ_TCS_FAST_BN <> 8000 H
            and TQ_TCS_SLOW_BN <>  8000 H
            and LV_TQ_ASR_REQ = 1
    then(3)     LV_ASR_ENA = 1
    else(3)     LV_ASR_ENA = 0
endif
```

```
IF(1) LV_VAR_BN = 0
Then(1) if  (2)     LC_MSR_DISABLE = 0
            and     LV_TCS_ENA = 1
            and     LV_MSR_PLAUS = 1
            and     LV_TQ_MSR_REQ = 1
         then (2)   LV_MSR_ENA = 1
         else (2)   LV_MSR_ENA = 0

else(1) if(3)       LC_MSR_DISABLE = 0
            and     LV_TCS_ENA = 1
      and   TQ_TCS_FAST_BN  <>      8000 H
      and   TQ_TCS_SLOW_BN <>       8000 H
            and     LV_TQ_MSR_REQ = 1
         then (3)   LV_MSR_ENA = 1
         else (3)   LV_MSR_ENA = 0
endif
-------------------------------------------------------
```

STATE_TQ_TCS_PLAUS:

```
If    LV_TCS_ENA = 0                                  BIT [00] set
IF[ LV_PL = 0   and  LV_PU = 0]                       BIT [01] set
IF LV_TQ_ASR_REQ = 1
Then IF   LV_VAR_BN = 0
            Then   IF LV_ASR_PLAUS = 0                BIT [02] set
            else  IF  TQ_TCS_FAST_BN ==   8000 H or   BIT [02] set
                      TQ_TCS_SLOW_BN ==  8000 H
else                                                  BIT [03] set
endif
If    LV_TCS_ENA = 0                                  BIT [00] set
IF[ LV_PL = 0   and  LV_PU = 0]                       BIT [01] set
IF LV_TQ_MSR_REQ = 1
Then IF   LV_VAR_BN = 0
            Then   IF LV_MSR_PLAUS = 0                BIT [02] set
            else  IF  TQ_TCS_FAST_BN ==   8000 H or   BIT [02] set
                      TQ_TCS_SLOW_BN ==  8000 H
else                                                  BIT [04] set
endif
```

Configuration data:

Name	Dim	Hex. Limits	Phys. Limits	Resol.	Unit
LC_ASR_DISABLE	1	0...1H	0...1	1	[-]
Locical constant to disable ASR interventions					
LC_MSR_DISABLE	1	0...1H	0...1	1	[-]
Locical constant to disable MSR interventions					

69.2.3 Torque request CAN selection

FUNCTION DESCRIPTION:

If one of the TCS interventions enabled (LV_ASR/MSR_ENA = 1), the torque request is received from CAN and copied into TQI_ASR/MSR_FAST. The slow and fast path are different messages.
After LV_ASR/MSR_ACT changes back to 0, both TQI_ASR/MSR_FAST and TQI_ASR/MSR_SLOW become their maximum/minimum value again.

Application conditions:

Initialisation: **at reset** TQI_ASR_FAST_REQ = 1023.97 Nm
TQI_ASR_SLOW_REQ = 1023.97 Nm
TQI_MSR_FAST_REQ = 0 Nm
TQI_MSR_SLOW_REQ = 0 Nm

Recurrence: 10 ms
Activation: at LV_IGK = 1
Deactivation: -

Formula section:

```
if      LV_VAR_BN = 1
then    TQI_ASR_SLOW_REQ_1 = TQ_ASR_SLOW_DEC_BN - TQ_LOSS
        TQI_ASR_FAST_REQ_1 = TQ_ASR_FAST_DEC_BN - TQ_LOSS
        TQI_MSR_SLOW_REQ_1 = TQ_MSR_SLOW_INC_BN - TQ_LOSS
        TQI_MSR_FAST_REQ_1 = TQ_MSR_FAST_INC_BN - TQ_LOSS
Else    TQI_MSR_SLOW_REQ_1 = TQI_MSR_REQ_CAN
        TQI_MSR_FAST_REQ_1 = TQI_MSR_REQ_CAN
        TQI_ASR_FAST_REQ_1 = TQI_ASR_FAST_REQ_CAN
        TQI_ASR_SLOW_REQ_1 =TQI_ASR_SLOW_REQ_CAN
endif
IF      LV_ASR_ENA= 1
Then    TQI_ASR_SLOW_REQ = TQI_ASR_SLOW_REQ_1
        TQI_ASR_FAST_REQ = TQI_ASR_FAST_REQ_1
Else    TQI_ASR_SLOW_REQ = 1023.97 Nm
        TQI_ASR_FAST_REQ = 1023.97 Nm
endif
IF      LV_MSR_ENA = 1
Then    TQI_MSR_SLOW_REQ = TQI_MSR_SLOW_REQ_1
        TQI_MSR_FAST_REQ = TQI_MSR_FAST_REQ_1
Else    TQI_MSR_SLOW_REQ = 0 Nm
        TQI_MSR_FAST_REQ = 0 Nm
endif
```

69.2.4 TCS intervention

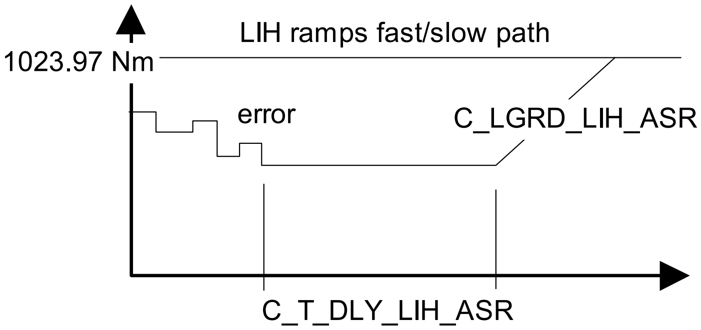
69.2.4.1 ASR intervention and emergency operation

FUNCTION DESCRIPTION:

If during an active TCS intervention an TCS error or an EMS error occurs, the current torque reduction is continued for the time C_T_DLY_LIH_ASR and then subsequently increased to TQI_ASR_FAST = TQI_ASR_SLOW = 1023,97 Nm using C_LGRD_LIH_ASR.

At transition from limp home condition active to passive after torque increase to TQI_ASR_FAST = TQI_ASR_SLOW = 1023,97 Nm at limp home operation the counter T_TCS_LIH_CTR is stopped and set to 0.

Signal flow diagram:



Application conditions:

Initialisation: at reset **or** at transition LV_IGK 0 --> 1
LV_ASR_LIH = 0
LV_ASR_ACT= 0
TQI_ASR_FAST = TQI_ASR_SLOW = 1023.97 Nm
T_ASR_LIH_CTR = 0

Recurrence: 10 ms

Activation: LV_IGK = 1

Formula section:

ASR intervention and detection of ASR limp home operation :

```
If (1)     LV_TCS_DISABLE_CAN = 1                                  or
           (LV_VAR_BN = 0 and LV_ASR_PLAUS = 0 and LV_TQ_ASR_REQ = 1)
   or
           (LV_VAR_BN = 1 and LV_TQ_ASR_REQ = 0 and LV_TQ_MSR_REQ = 0 and
           TQ_TCS_FAST_BN   <> 8000 H  and TQ_TCS_SLOW_BN <>  8000 H)

then (1)   if(1b)  TQI_ASR_FAST < 7FFFH    or
                   TQI_ASR_SLOW < 7FFFH
           Then(1b) LV_ASR_LIH = 1
                   If(1c)        T_ASR_LIH_CTR < C_T_DLY_LIH_ASR      (time delay)
                   Then(1c)      TQI_ASR_FAST $_{N}$    = TQI_ASR_FAST $_{N-1}$  (last valid value)
                                 TQI_ASR_SLOW $_{N}$    = TQI_ASR_SLOW $_{N-1}$(last valid value)
                                 T_ASR_LIH_CTR $_{N}$ = T_ASR_LIH_CTR $_{N-1}$ + 10 ms
                                          ( till T_TCS_LIH_CTR = C_T_DLY_LIH_ASR )
                   Else(1c)      TQI_ASR_FAST $_{N}$    = TQI_ASR_FAST $_{N-1}$ +
                                                     C_LGRD_LIH_ASR
                                          (till TQI_TCS_FAST = 1023.97 Nm )
                                 TQI_ASR_SLOW $_{N}$   = TQI_ASR_FAST $_{N}$
                   Endif(1c)
           else(1b) LV_ASR_LIH        = 0
                   T_TCS_LIH_CTR      = 0
           Endif(1b)
Else(1)            LV_ASR_LIH = 0
                   T_TCS_LIH_CTR = 0
                   TQI_ASR_FAST = TQI_ASR_FAST_REQ
                   TQI_ASR_SLOW  = TQI_ASR_SLOW_REQ
```

endif (1)

ASR intervention active (LV_ASR_ACT) :

```
if      TQI_ASR_SLOW < 1023.97 Nm      or
        TQI_ASR_FAST < 1023.97 Nm
then    LV_ASR_ACT= 1
else    LV_ASR_ACT= 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LGRD_LIH_ASR	1	0...7FFFH	0...1023.96875	0.03125	[Nm/10ms]
Limiting gradient for ASR emergency ramp function (default 0.6 Nm/10ms)					
C_T_DLY_LIH_ASR	1	0...FFH	0...2.55	0.01	[s]
Delay time for activation of ASR emergency ramp function (default 0s)					

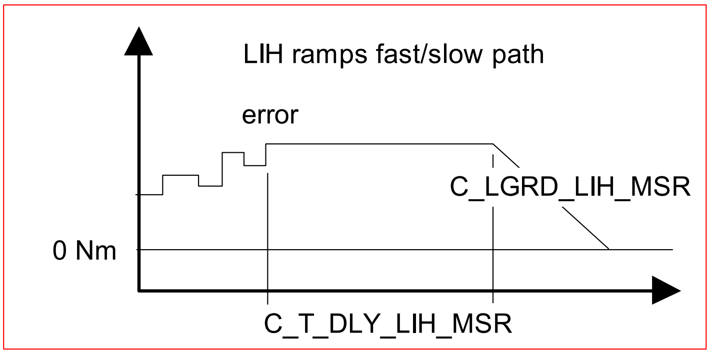
69.2.4.2 MSR intervention and emergency operation

FUNCTION DESCRIPTION:

If during an active MSR intervention an TCS error or an EMS error occurs, the current torque offset is continued for the time C_T_DLY_LIH_MSR and then subsequently decreased to TQI_MSR_FAST = TQI_MSR_SLOW = 0 Nm using C_LGRD_LIH_MSR.

At transition from limp home condition active to passive after torque decrease to TQI_MSR_FAST = TQI_MSR_SLOW = 0 Nm at limp home operation the counter T_TCS_LIH_CTR is stopped and set to 0.

Signal flow diagram:



Application conditions:

Initialisation: at reset **or** at transition LV_IGK 0 --> 1

LV_MSR_LIH = 0

LV_MSR_ACT = 0

TQI_MSR_FAST = TQI_MSR_SLOW = 0 Nm

T_MSR_LIH_CTR = 0

Recurrence: 10 ms

Activation: LV_IGK = 1

Formula section:

MSR intervention and detection of MSR limp home operation :

```
If (1)      LV_TCS_DISABLE_CAN = 1                                or
            (LV_VAR_BN = 0 and LV_MSR_PLAUS = 0 and LV_TQ_MSR_REQ = 1)
    or
            (LV_VAR_BN = 1 and LV_TQ_MSR_REQ = 0 and LV_TQ_ASR_REQ = 0 and
            TQ_TCS_FAST_BN   <> 8000 H  and TQ_TCS_SLOW_BN <>  8000 H)
then (1)
            if (1b)  TQI_MSR_FAST > 0            or
                     TQI_MSR_SLOW > 0
            Then (1b)     LV_MSR_LIH = 1
                          if (2)   T_MSR_LIH_CTR     < = C_T_DLY_LIH_MSR (time delay)
                          then (2) TQI_MSR_FAST $_N$   = TQI_MSR_FAST $_{N-1}$ (last valid value)
                                   TQI_MSR_SLOW $_N$   =TQI_MSR_SLOW $_{N-1}$(last valid value)
                                   T_MSR_LIH_CTR $_N$ = T_MSR_LIH_CTR $_{N-1}$ + 10 ms
                                           ( till T_MSR_LIH_CTR = C_T_DLY_LIH_MSR )
                          else (2) TQI_MSR_FAST $_N$   = TQI_MSR_FAST $_{N-1}$ -
                                                          C_LGRD_LIH_MSR
                                                     (till TQI_MSR_FAST = 0Nm )
                                   TQI_MSR_SLOW $_N$   = TQI_MSR_FAST $_N$
                          endif (2)
            Endif (1b)
else (1)         LV_MSR_LIH = 0
                 T_TCS_LIH_CTR      = 0
                 TQI_MSR_FAST       = min( (C_TQ_MAX_MSR -
                                                 TQ_LOSS); TQI_MSR_FAST_REQ)
                 TQI_MSR_SLOW       = min( (C_TQ_MAX_MSR -
                                                 TQ_LOSS); TQI_MSR_SLOW_REQ)
endif (1)
```

MSR intervention active (LV_MSR_ACT) :

```
if          TQI_MSR_SLOW > 0 Nm            or
            TQI_MSR_FAST  > 0 Nm
then        LV_MSR_ACT         = 1
else        LV_MSR_ACT         = 0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_LGRD_LIH_MSR	1	0...7FFFH	0...1023.96875	0.03125	[Nm/10ms]
Limiting gradient for MSR emergency ramp function (default 0.6 Nm/10ms)					
C_T_DLY_LIH_MSR	1	0...FFH	0...2.55	0.01	[s]
Delay time for activation of MSR emergency ramp function (default 0s)					
C_TQ_MAX_MSR	1	0...FFFFH	-1024...1023.96875	0.03125	[Nm]
Limiting torque for MSR					

69.2.5 Status flag driving torque at clutch SF_TQD

Application conditions:

Initialisation: SF_TQD = 0 **at reset**

Recurrence: 10 ms

Activation: at LV_IGK = 1

Deactivation: -

Formula section:

```
If (1)    STATE_ETC_LIH    = ETC_LIH_1        or
          STATE_ETC_LIH    = ETC_LIH_2_REV    or
          STATE_ETC_LIH    = ETC_LIH_2        or
          LV_OFF_MTC_MON         = 1
          (see Chapter: "Applc. Inc of process monitoring" and
          "Fault reaction of Process monitoring")     or
          LV_SF_TQD_MON = 1

then (1)  SF_TQD = 03H
else(1)If(2)    IGA_SP_H_RNG <= IGA_MIN_H_RNG       or
                LV_TQ_LIM_INTV = 1
    Then (2)    SF_TQD = 01H
    Else (2)    SF_TQD = 00H
    Endif(2)
Endif(1)
```

70 Variable Intake Manifold

Table of Contents

Index

70.1 Variable Intake Manifold

Output data:

<table>
<tr><th>Name</th><th>Mode</th><th>Hex. limits</th><th>Phys. limits</th><th>Resol.</th><th>Unit</th></tr>
<tr><td>VIMPWM_1</td><td>O/V</td><td>8000...7FFFH</td><td>-100...99.99694</td><td>3.0518e-3</td><td>[%]</td></tr>
<tr><td colspan="6">pulse width modulated signal for actuator DISA_1</td></tr>
<tr><td>VIMPWM_2</td><td>O/V</td><td>8000...7FFFH</td><td>-100...99.99694</td><td>3.0518e-3</td><td>[%]</td></tr>
<tr><td colspan="6">pulse width modulated signal for actuator DISA_2</td></tr>
<tr><td>VIM_AV</td><td>O/V</td><td>0...FFFFH</td><td>0...199.99694</td><td>3.0518e-3</td><td>[%]</td></tr>
<tr><td colspan="6">actual position variable intake manifold</td></tr>
<tr><td>LV_VIM</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Logical value VIM power stage (2-stage-DISA); 0 passive, 1 active</td></tr>
<tr><td>LV_VIM_RLS</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">conditions for release of VIM function are met</td></tr>
<tr><td>N_VIM</td><td>V</td><td>0...1FE0H</td><td>0...8160</td><td>1</td><td>[rpm]</td></tr>
<tr><td colspan="6">corrected engine speed for VIM application</td></tr>
<tr><td>LV_VIM_1</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">setpoint VIM flap 1, 0: closed, 1: opened</td></tr>
<tr><td>LV_VIM_2</td><td>O/V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">setpoint VIM flap 2, 0: closed, 1: opened</td></tr>
<tr><td>T_OPEN_VIM_DEAC</td><td>V</td><td>0...FFH</td><td>0...2.55</td><td>0.01</td><td>[s]</td></tr>
<tr><td colspan="6">timer to open the VIM flaps before deactivating the VIM function</td></tr>
<tr><td>LV_SWI_VIM_PAS</td><td>V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">Logical variable to set VIM in passive mode</td></tr>
<tr><td>LV_VIM_RLS_TMP</td><td>V</td><td>0...1H</td><td>0...1</td><td>1</td><td>[-]</td></tr>
<tr><td colspan="6">LV_VIM_RLS temporal value</td></tr>
<tr><td>T_VIM_LIH_MIN</td><td>V</td><td>0...FFH</td><td>0...2.55</td><td>0.01</td><td>[s]</td></tr>
<tr><td colspan="6">Limp-home timer for VIM function</td></tr>
<tr><td>T_DLY_VIM_OPEN_1</td><td>V</td><td>0...FFH</td><td>0...2.55</td><td>0.01</td><td>[s]</td></tr>
<tr><td colspan="6">Delay timer to open VIM flap 1</td></tr>
<tr><td>T_DLY_VIM_OPEN_2</td><td>V</td><td>0...FFH</td><td>0...2.55</td><td>0.01</td><td>[s]</td></tr>
<tr><td colspan="6">Delay timer to open VIM flap 2</td></tr>
<tr><td>T_VIM_CLOSE_MIN_1</td><td>V</td><td>0...FFH</td><td>0...2.55</td><td>0.01</td><td>[s]</td></tr>
<tr><td colspan="6">Timer to close VIM flap 1</td></tr>
<tr><td>T_VIM_CLOSE_MIN_2</td><td>V</td><td>0...FFH</td><td>0...2.55</td><td>0.01</td><td>[s]</td></tr>
<tr><td colspan="6">Timer to close VIM flap 2</td></tr>
</table>

Input data:

LV_ST_END	VB	LV_ERR_VIM_1	LV_ERR_VIM_2
LV_VIM_1_EXT_ADJ	LV_VIM_2_EXT_ADJ	LV_ACT_VIM_1_EXT_ADJ	LV_ACT_VIM_2_EXT_ADJ
N_32	TIA	TCO	FAC_MAF_MAX
LV_VIM_INH_CUS	LV_VIM_1_CUS	LV_VIM_2_CUS	LV_STST_STOP_CYC
LV_IGK	N	VS	

General information:

This function is used to control an open loop, 2- or 3-stage variable intake manifold. The output VIM_AV is generated to be used for the volumetric efficiency of the engine which is influenced by the VIM position. For a 2-stage variable intake manifold VIM_AV should be set to 0% / 100% in order to facilitate the application of the volumetric efficiency.
For test purpose at assembling line or workshop the VIM can be switched by a serial communication tool, if available (external adjustment).

Signal flow diagram:

Determination of the used range

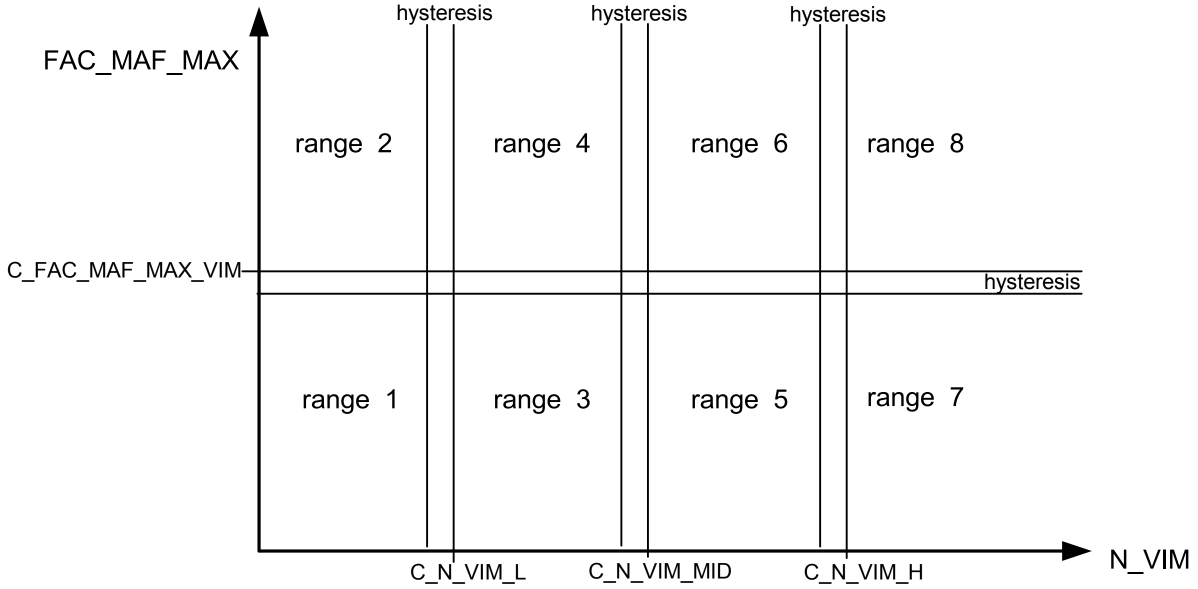


Diagram is valid if external adjustment is inactive.

Application conditions:

Initialisation: VIMPWM_1 = C_VIMPWM_PAS — at ECU initialisation
VIMPWM_2 = C_VIMPWM_PAS — and at function deactivation

VIM_AV = C_VIM_AV_1

T_OPEN_VIM_DEAC = C_T_OPEN_VIM_DEAC

all others: physical 0

at LV_IGK 0->1 or at reset : LV_VIM_RLS_TMP = 0

Recurrence: 10 ms

Activation: at every engine operating state

Deactivation: -

Formula section:

Condition for release of VIM function

```
If      LV_ST_END = 1                                and
        C_VB_MIN_VIM  ≤  VB  ≤  C_VB_MAX_VIM             and
        ID_STATE_RLS_VIM = 1                         and
        LV_ERR_VIM_1 = 0                             and
        LV_ERR_VIM_2 = 0
then    LV_VIM_RLS  =  1
else    LV_VIM_RLS  =  0
endif
```

Corrected engine speed for VIM function:

```
If    LV_VIM_RLS  =  1
then          N_VIM  =  N * IP_FAC_N_VIM * IP_FAC_TIA_N_VIM
endif
```

Hysteresis for switching thresholds:

Valid for all 4 thresholds, C_N_VIM_L, C_N_VIM_MID, C_N_VIM_H and C_FAC_MAF_MAX_VIM: when increasing the input value (N_VIM / FAC_MAF_MAX) then switch at
input value > threshold ,
when decreasing the input value then switch back at
input value < threshold - hysteresis .

threshold	C_N_VIM_L / -_MID / -_H	C_FAC_MAF_MAX_VIM
hysteresis	C_N_VIM_HYS	C_FAC_MAF_MAX_VIM_HYS

Determination of desired flap position

i = 1...2

```
if(1)   LC_PWM_VIM_MAN = 1
then(1)      LV_VIM_1  =  C_STATE_VIM_1_SP_MAN
             LV_VIM_2  =  C_STATE_VIM_2_SP_MAN
else(1)
  if(2)   LV_VIM_i_EXT_ADJ = 1
  then(2)   LV_VIM_i  =  LV_ACT_VIM_i_EXT_ADJ
  else(2)
    if(3)   LV_VIM_RLS = 1
    then(3)
        if(4)        LC_PWM_VIM_CUS = 1
        then(4)      LV_VIM_1 = LV_VIM_1_CUS
                     LV_VIM_2 = LV_VIM_2_CUS
        else(4)      LV_VIM_1  =  ID_STATE_VIM_1_SP      0: closed  or  1: opened, according
                     LV_VIM_2  =  ID_STATE_VIM_2_SP        the determined range, see above
        endif(4)                                           diagram
    else(3) LV_VIM_1  =  1                                 flaps opened before
            LV_VIM_2  =  1                                 switching off
    endif(3)
  endif(2)
endif(1)
```

Hint: the tables ID_STATE_VIM_1/2_SP contain 8 values which are divided in the 8 ranges, see above diagram. The lists of data points are fixed, not calibrateable. According the display conversion of the calibration system ‘INCA’, the table that can be seen on the INCA-screen is analogous the diagram above.

	LDPM_IDX_1_SP_VIM_1_VIMA			
	0	1	2	3
LDPM_IDX_2_SP_VIM_1_VIMA 0	flap setpoint range 2	flap setpoint range 4	flap setpoint range 6	flap setpoint range 8
1	flap setpoint range 1	flap setpoint range 3	flap setpoint range 5	flap setpoint range 7

Calculation of VIMPWM_1 and_2:

i = 1...2

```
if(1)   LC_PWM_VIM_MAN = 1
then(1)
  if(2a)   LV_VIM_i  =  1
  then(2a)          VIMPWM_i    =  C_PWM_VIM_OPEN
  else(2a)          VIMPWM_i    =  C_PWM_VIM_CLOSE
  endif(2a)
else(1)
   if(2b)    LV_VIM_i_EXT_ADJ = 1
   then(2b)
      if(3a)   LV_VIM_i  =  1
      then(3a)      VIMPWM_i    =  C_PWM_VIM_OPEN
      else(3a)      VIMPWM_i    =  C_PWM_VIM_CLOSE
      endif(3a)
    else(2b)
      if(3b)  LV_VIM_INH_CUS = 0
      then(3b)
              T_VIM_LIH_MIN = 0
         if(4a)       (LV_VIM_RLS = 1 and LV_IGK=1)
         then(4a)   T_OPEN_VIM_DEAC  =  0 s
                    LV_VIM_RLS_TMP = 1
            if(5a)     LV_VIM_i  =  1  /* output adjustment
            then(5a)          T_VIM_CLOSE_MIN_i = 0
                    If        T_DLY_VIM_OPEN_i < C_T_DLY_VIM_OPEN
                    then
                              T_DLY_VIM_OPEN_i = T_DLY_VIM_OPEN_i + 10ms
                              VIMPWM_i = C_PWM_VIM_CLOSE
                    else
                              VIMPWM_i = C_PWM_VIM_OPEN
                    Endif
             else(5a)         T_DLY_VIM_OPEN_i = 0



             /* torque adjustment
             if (6a) T_VIM_CLOSE_MIN_i < C_T_VIM_CLOSE_MIN
```

```
            then
                    T_VIM_CLOSE_MIN_i = T_VIM_CLOSE_MIN_i + 10ms
                    VIMPWM_i = C_PWM_VIM_CLOSE  /*Torque  adjustment for safe run up
            else (6a)
                    LV_SWI_VIM_PAS = ID_SWI_VIM_PAS_N_VS (N, VS)
                     /* see Calibration hints (1 , (2

                    if (7a)      LV_SWI_VIM_PAS = 1
                    then         VIMPWM_i = C_VIMPWM_PAS
                    else (7a)
                                 VIMPWM_i = C_PWM_VIM_CLOSE
                    endif (7a)
            endif (6a)
          endif(5a)
      else(4a)    T_OPEN_VIM_DEAC  =  T_OPEN_VIM_DEAC$_{n-1}$ + 10 ms
        if(5b)    T_OPEN_VIM_DEAC  <  C_T_OPEN_VIM_DEAC
        then(5b)  VIMPWM_i     =  C_PWM_VIM_DEAC
        else(5b)
            if (6b) T_OPEN_VIM_DEAC < C_T_DLY_PARK_PSN
            then   VIMPWM_i = C_VIMPWM_PAS
            else (6b)
                    if (7b)  LV_STST_STOP_CYC = 1     or
                             LV_ERR_VIM_1 = 1 or
                             LV_ERR_VIM_2 = 1 or
                             LV_VIM_RLS_TMP = 0
                    then     VIMPWM_i = C_VIMPWM_PRE_PRK_PSN
                    else (7b)
                             T_DLY_VIM_OPEN_i = 0
                             /* parking position,
                             VIMPWM_i = C_VIMPWM_PARK_PSN
                             /* see Calibration hints (3
                             T_VIM_CLOSE_MIN_i = 0 /* needed for restart from PWL phase
                    endif (7b)
            endif (6b)
          endif(5b)
      endif(4a)

    else(3b)
            T_DLY_VIM_OPEN_i = 0
            LV_SWI_VIM_PAS = 0
            /* Limp home active, therefore VIMPWM_i = C_PWM_VIM_LIH
            T_VIM_CLOSE_MIN_i = 0
            /* limp home reappears :
            if (4b) T_VIM_LIH_MIN < C_T_VIM_LIH_MIN
            then
                    T_VIM_LIH_MIN = T_VIM_LIH_MIN + 10ms
                    VIMPWM_i = C_PWM_VIM_LIH /*see Calibration hints (4
                    /*Desired adjustment for safe run up
            else (4b)
                    VIMPWM_1 = C_VIMPWM_PAS
                    VIMPWM_2 = C_VIMPWM_PAS
            endif(4b)
```

```
        endif(3b)
      endif(2b)

  endif(1)
```

Calculation of VIM_AV:

```
if(1)   VIMPWM_1 = VIMPWM_2 = C_VIMPWM_PAS
then(1)
      if(2a)    LV_SWI_VIM_PAS = 1
      then      VIM_AV = VIM_AV (n-1) /* VIM flap remains in the old position*/
      else(2a)  VIM_AV = C_VIM_AV_PAS
      endif(2a)
else(1)
      if(2)   VIMPWM_1 = VIMPWM_2 = C_PWM_VIM_DEAC
      then(2)                 VIM_AV  =  C_VIM_AV_DEAC
          if(3)         VIMPWM_1 = VIMPWM_2 = C_PWM_VIM_OPEN
          then(3)       VIM_AV  =  C_VIM_AV_1
          else(3)
                if(4)   VIMPWM_1 = VIMPWM_2 = C_PWM_VIM_CLOSE
                then(4)         VIM_AV  =  C_VIM_AV_2
                else(4)
                          if(5)   VIMPWM_1 = C_PWM_VIM_OPEN          and
                                  VIMPWM_2 = C_PWM_VIM_CLOSE
                          then(5)         VIM_AV  =  C_VIM_AV_3
                          else(5)         VIM_AV  =  C_VIM_AV_4
                          endif(5)
                endif(4)
             endif(3)
      endif(2)
endif(1)
```

In case of inhibition of VIM by customer:

```
If          LC_PWM_VIM_MAN = 0      and
            LV_VIM_i_EXT_ADJ = 0    and
            LV_VIM_INH_CUS = 1
then        VIM_AV  =  C_VIM_AV_1
endif
```

LV_VIM:

```
If    VIMPWM_1  ≥  C_PWM_VIM_CLOSE
  then      LV_VIM  =  1
  else      LV_VIM  =  0
endif
```

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_VB_MIN_VIM	1	0...FFH	0...25.89843	0.1015625	[V]
minimum battery voltage to release VIM function					
C_VB_MAX_VIM	1	0...FFH	0...25.89843	0.1015625	[V]
maximum battery voltage to release VIM function					
C_N_VIM_L	1	0...1FE0H	0...8160	1	[rpm]
lower engine speed threshold					
C_N_VIM_MID	1	0...1FE0H	0...8160	1	[rpm]
medium engine speed threshold					
C_N_VIM_H	1	0...1FE0H	0...8160	1	[rpm]
higher engine speed threshold					
C_N_VIM_HYS	1	0...1FE0H	0...8160	1	[rpm]
engine speed threshold hysteresis					
C_FAC_MAF_MAX_VIM	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
load threshold					
C_FAC_MAF_MAX_VIM_HYS	1	0...FFFFH	0...1.99996	0.0305e-3	[-]
load threshold hysteresis					
LC_PWM_VIM_MAN	1	0...1H	0...1	1	[-]
Logical constant to enable manual setting of VIMPWM_1/2 by calibration system					
C_STATE_VIM_1_SP_MAN	1	0...1H	0...1	1	[-]
manual setting of the setpoint VIM flap 1 by calibration system, 0: closed, 1: opened					
C_STATE_VIM_2_SP_MAN	1	0...1H	0...1	1	[-]
manual setting of the setpoint VIM flap 2 by calibration system, 0: closed, 1: opened					
C_VIMPWM_PAS	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
VIMPWM "passive"					
C_PWM_VIM_CLOSE	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
PWM to close VIM flap					
C_PWM_VIM_OPEN	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
PWM to open VIM flap					
C_T_OPEN_VIM_DEAC	1	0...FFH	0...2.55	0.01	[s]
time to open the VIM flaps before deactivating the VIM function					
ID_STATE_RLS_VIM	0	0...1H	0...1	1	[-]
temperature condition for VIM function is met (1) or not met (0)					
IP_FAC_N_VIM	8*6	0...FFH	0...1.99218	0.0078125	[-]
LDP_N_32_FAC_N_VIM	8	0...FFH	0...8160	32	[rpm]
LDP_FAC_MAF_MAX_IP_FAC_N_VIM	6	0...FFFFH	0...1.99996	0.0305e-3	[-]
N_32 correction for VIM switch thresholds					
ID_STATE_VIM_1_SP	0	0...1H	0...1	1	[-]
Table defining the setpoint VIM flap 1, 0: closed, 1: opened					
ID_STATE_VIM_2_SP	0	0...1H	0...1	1	[-]
Table defining the setpoint VIM flap 2, 0: closed, 1: opened					
C_VIM_AV_PAS	1	0...FFFFH	0...199.99694	3.0518e-3	[%]
VIM_AV during passive function					
C_VIM_AV_1	1	0...FFFFH	0...199.99694	3.0518e-3	[%]
VIM_AV for both flaps opened					
C_VIM_AV_2	1	0...FFFFH	0...199.99694	3.0518e-3	[%]
VIM_AV for both flaps closed					
C_VIM_AV_3	1	0...FFFFH	0...199.99694	3.0518e-3	[%]
VIM_AV for flap 1 opened, flap 2 closed					
C_VIM_AV_4	1	0...FFFFH	0...199.99694	3.0518e-3	[%]
VIM_AV for flap 1 closed, flap 2 opened					
IP_FAC_TIA_N_VIM	8	0...FFH	0...1.99218	0.0078125	[-]
LDP_TIA_FAC_TIA_N_VIM	8	0...FEH	-48...142.5	0.75	[°C]
TIA correction for VIM switch thresholds					
C_PWM_VIM_DEAC	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
PWM-signal to deactivate VIM flap					
C_VIM_AV_DEAC	1	0...FFFFH	0...199.99694	3.0518e-3	[%]

VIM_AV during deactivation					
LC_PWM_VIM_CUS	1	0...1H	0...1	1	[-]
Logical constant to enable setting of customer VIM setpoint					
C_T_VIM_CLOSE_MIN	1	0...FFH	0...2.55	0.01	[s]
minimum time to close VIM flaps					
ID_SWI_VIM_PAS_N_VS	6*6	0...1H	0...1	1	[-]
LDP_N_32_SWI_VIM_PAS	6	1...FFH	32...8160	32	[rpm]
LDP_VS_SWI_VIM_PAS	6	0...FFH	0...255	1	[km/h]
VIM Passive mode dependent on vehicle speed and motor speed					
C_T_DLY_PARK_PSN	1	0...FFH	0...2.55	0.01	[s]
delay timer to open the VIM flaps before deactivating the VIM function (park position)					
C_VIMPWM_PARK_PSN	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
PWM adjustment in case of park position					
C_T_VIM_LIH_MIN	1	0...FFH	0...2.55	0.01	[s]
minimum time to VIM limp home mode					
C_PWM_VIM_LIH	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
PWM for VIM limp home mode					
C_VIMPWM_PRE_PRK_PSN	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
PWM adjustment in case of pre-park position					
C_T_DLY_VIM_OPEN	1	0...FFH	0...2.55	0.01	[s]
delay timer to open the VIM flaps					

Calibration hints:

1) ID_SWI_VIM_PAS_N_VS out of 3000 rpm, ID_SWI_VIM_PAS_N_VS==0 (67% PWM isn´t allowed after 15% PWM)
2) ID_SWI_VIM_PAS_N_VS : Passive Mode for N< 3000U/min
3) C_VIMPWM_PARK_PSN: At parking position (15%-->67%): PWM= 67%
4) C_PWM_VIM_LIH = 33 %

70.2 VIMA actuator diagnosis

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
LV_ERR_VIM_1	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on VIM_1 (VIM_H) signal.					
LV_ERR_VIM_2	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on VIM_2 (VIM_L) signal.					
CDN_DIAG_VIM_1_BAS	O	0...7H	0...7	1	[-]
Primary diagnosis condition read from the infrastructure for each symptom of VIM bit 0: diagnosis condition for symptom SYM_0 (SCP) bit 1: diagnosis condition for symptom SYM_1 (SCG) bit 2: diagnosis condition for symptom SYM_2 (OC)					
CDN_DIAG_VIM_2_BAS	O	0...7H	0...7	1	[-]
Primary diagnosis condition read from the infrastructure for each symptom of VIM bit 0: diagnosis condition for symptom SYM_0 (SCP) bit 1: diagnosis condition for symptom SYM_1 (SCG) bit 2: diagnosis condition for symptom SYM_2 (OC)					
ERR_DIAG_VIM_1	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for VIM_1 (only parameter)					
ERR_DIAG_VIM_2	-	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Raw value of error symptom for VIM_2 (only parameter)					
ERR_VIMPWM_1_FB	V/O	0...FFH	0...255	1	[-]
00H: no failure feedback01H: failure 1 detected by control unitFFH: init value / feedback possibleThe values 01H are only valid, if LV_VIMPWM_1_FB_VLD = 1, else the failure is only supposed.					
ERR_VIMPWM_2_FB	V/O	0...FFH	0...255	1	[-]
00H: no failure feedback01H: failure 1 detected by control unitFFH: init value / feedback possibleThe values 01H are only valid, if LV_VIMPWM_2_FB_VLD = 1, else the failure is only supposed.					
T_VIMPWM_1_FB	V	0...FFH	0...25.5	0.1	[s]
Timer counting tlow for failure feedback recognition					
T_VIMPWM_2_FB	-	0...FFH	0...25.5	0.1	[s]
Timer counting tlow for failure feedback recognition					
LV_VIMPWM_1_FB_VLD	V/O	0...1H	0...1	1	[-]
If LV_VIMPWM_1_FB_VLD = 1 the failure number ERR_VIMPWM_1_FB (01H) is valid					
LV_VIMPWM_2_FB_VLD	V/O	0...1H	0...1	1	[-]
If LV_VIMPWM_2_FB_VLD = 1 the failure number ERR_VIMPWM_2_FB (01H) is valid					
LV_ERR_VIM_1_EL	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on VIM_1 (VIM_H) signal.					
LV_ERR_VIM_2_EL	V/O	0...1H	0...1	1	[-]
Boolean for error currently present on VIM_2 (VIM_L) signal.					
LV_CDN_DIAG_VIM_1_EL	V/O	0...1H	0...1	1	[-]
Diagnosis condition VIM_1 diagnosis					
LV_CDN_DIAG_VIM_2_EL	V/O	0...1H	0...1	1	[-]
Diagnosis condition VIM_2 diagnosis					
ERR_SYM_VIM_1_EL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Detected error symptom VIM_1 diagnosis					
ERR_SYM_VIM_2_EL	V/O	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]

Detected error symptom VIM_2 diagnosis					
LV_END_DIAG_VIM_1_EL	V/O	0...1H	0...1	1	[-]
End of diagnosis VIM_1 diagnosis					
LV_END_DIAG_VIM_2_EL	V/O	0...1H	0...1	1	[-]
End of diagnosis VIM_2 diagnosis					
LV_INH_DIAG_VIM_1_EL	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition VIM_1 diagnosis					
LV_INH_DIAG_VIM_2_EL	V/O	0...1H	0...1	1	[-]
Diagnosis inhibition VIM_2 diagnosis					
CDN_DIAG_VIM_1_EL	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of VIM_1bit 0: diagnosis condition for symptom SCP (SYM_0)bit 1: diagnosis condition for symptom SCG (SYM_1)bit 2: diagnosis condition for symptom OC (SYM_2)					
CDN_DIAG_VIM_2_EL	V	0...7H	0...7	1	[-]
Diagnosis condition for each symptom of VIM_2bit 0: diagnosis condition for symptom SCP (SYM_0)bit 1: diagnosis condition for symptom SCG (SYM_1)bit 2: diagnosis condition for symptom OC (SYM_2)					
LV_ERR_VIMPWM_1_FB	V/O	0...1H	0...1	1	[-]
Error currently present, indicated by failure feedback from DISA control unit.					
LV_ERR_VIMPWM_2_FB	V/O	0...1H	0...1	1	[-]
Error currently present, indicated by failure feedback from DISA control unit.					
LV_CDN_DIAG_VIMPWM_1_FB	V	0...1H	0...1	1	[-]
Diagnosis condition DISA failure feedback diagnosis.					
LV_CDN_DIAG_VIMPWM_2_FB	V	0...1H	0...1	1	[-]
Diagnosis condition DISA failure feedback diagnosis.					
ERR_SYM_VIMPWM_1_FB	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom DISA failure feedback diagnosis.					
ERR_SYM_VIMPWM_2_FB	V	0H 1H 2H 4H 8H	NO_SYM SYM_0 SYM_1 SYM_2 SYM_3	1	[-]
Error symptom DISA failure feedback diagnosis.					
LV_END_DIAG_VIMPWM_1_FB	V	0...1H	0...1	1	[-]
End of diagnosis DISA failure feedback diagnosis.					
LV_END_DIAG_VIMPWM_2_FB	V	0...1H	0...1	1	[-]
End of diagnosis DISA failure feedback diagnosis.					
ABC_VIMPWM_1_FB	-	0...FFH	0...255	1	[-]
Antibounce counter for DISA failure feedback diagnosis.					
ABC_VIMPWM_2_FB	-	0...FFH	0...255	1	[-]
Antibounce counter for DISA failure feedback diagnosis.					
T_ACT_VIMPWM_1_FB	-	0...FFFFH	0...6553.5	0.1	[s]
Time with VIMPWM_1 > C_VIMPWM_MIN_FB necessary to have failure feedback from DISA control unit					
T_ACT_VIMPWM_2_FB	-	0...FFFFH	0...6553.5	0.1	[s]
Time with VIMPWM_2 > C_VIMPWM_MIN_FB necessary to have failure feedback from DISA control unit					
LV_EDGE_VIM_1	V	0...1H	0...1	1	[-]
DISA switch (Flap 1)					
LV_EDGE_VIM_2	V	0...1H	0...1	1	[-]
DISA switch (Flap 2)					
T_DIAG_VIMPWM_1_FB	V	0...FFFFH	0...6553.5	0.1	[s]
feedback timer to allow decrement after switch (Flap 1)					
T_DIAG_VIMPWM_2_FB	V	0...FFFFH	0...6553.5	0.1	[s]
feedback timer to allow decrement after switch (Flap 2)					
LV_EDGE_DIAG_VIM_1	V	0...1H	0...1	1	[-]
DISA switch ready (Flap 1)					
LV_EDGE_DIAG_VIM_2	V	0...1H	0...1	1	[-]
DISA switch ready (Flap 2)					
CTR_VIMPWM_1_EDGE	V/O/S	0...FFFFFFH	0...16777215	1	[-]
counter detection switch flap 1					
CTR_VIMPWM_2_EDGE	V/O/S	0...FFFFFFH	0...16777215	1	[-]

counter detection switch flap 2

Input data:

N_32	VIMPWM_1	VIMPWM_2	LV_CDN_VB_OBD1
LV_ERR_SPI_MPS	LV_IGK		LV_CDN_VB_MIN_DIAG
LV_VIM_1	LV_VIM_2		

70.2.1 Summary of all VIM diagnosis

FUNCTION DESCRIPTION:

This function shall summarize all VIM failures in one flag only for each VIM.

Description:

Application conditions:

Initialisation: reset **or** FMY reset **or** LV_IGK = 0 -> 1

LV_ERR_VIM_1 = 0

LV_ERR_VIM_2 = 0

Recurrence: 100ms

Activation: every ECU state

Formula section:

```
If LV_ERR_VIM_1_EL = 1 or LV_ERR_VIMPWM_1_FB = 1
Then LV_ERR_VIM_1 = 1
Else  LV_ERR_VIM_1 = 0
Endif

If LV_ERR_VIM_2_EL = 1 or LV_ERR_VIMPWM_2_FB = 1
Then LV_ERR_VIM_2 = 1
Else  LV_ERR_VIM_2 = 0
Endif
```

70.2.2 Diagnostic information from infrastructure

FUNCTION DESCRIPTION:

Description:

The DISA control unit is controlled by a PWM signal line driven by the ATIC 39. The driver can distinguish between three symptoms: ‘Short circuit to Vbatt’, ‘Short circuit to GND’ and ‘Open Load’.
This function reads the failure information (ERR_DIAG_VIM_1/2) from the infrastructure.

Application conditions:

Initialisation: reset **or** reset of failure memory **or** LV_IGK = 0 -> 1

CDN_DIAG_VIM_1_BAS = 0
CDN_DIAG_VIM_2_BAS = 0
ERR_DIAG_VIM_1 = 0
ERR_DIAG_VIM_2 = 0

Recurrence: 100ms

Activation: every ECU state

Formula section:

Usage of the diagnosis information (failure symptoms (raw value) ERR_DIAG_VIM_1/2 and diagnosis conditions CDN_DIAG_VIM_1_BAS/ 2_BAS received from the infrastructure:

Diagnosis conditions are set according infrastructure info.: CDN_DIAG_VIM_1_BAS/ 2_BAS
Failure symptoms are set according infrastructure info.: ERR_DIAG_VIM_1/2

70.2.3 Failure feedback recognition of VIM

FUNCTION DESCRIPTION:

If a failure is detected from the DISA control unit, a failure feedback is send to the ECU by pulling the signal line (interface ECU to the DISA control unit) to ground in a defined pattern. The time t_{low} is an identification for the failure.

Description:

For the recognition of the failure feedback the information, if a failure “short circuit to ground” is detected on the signal line between ECU and DISA control unit is used.

Application conditions:

Initialisation: at reset **or** FMY reset **or** LV_IGK = 0 -> 1

ERR_VIMPWM_1_FB = FFH
ERR_VIMPWM_2_FB = FFH

T_VIMPWM_1_FB = 0
T_VIMPWM_2_FB = 0

LV_VIMPWM_1_FB_VLD = 0
LV_VIMPWM_2_FB_VLD = 0

Recurrence: 100ms

*FB-Diagnosis has to be calculated **after** VIM-Powerstage-Diagnosis*

Activation:
LV_IGK = 1 **and**
LV_CDN_VB_MIN_DIAG = 1 **and**
N_32 ≤ C_N_MAX_ERR_VIM_FB **and**
LV_ERR_SPI_MPS = 0

Deactivation: not activation
at deactivation function is initialized one time

Formula section:

```
LV_VIMPWM_1_FB_VLD = 0

If      ERR_DIAG_VIM_1 = 2                  (SCG detected by Powerstage-diagnosis)
Then    T_VIMPWM_1_FB  =  T_VIMPWM_1_FB + 100ms
        ERR_VIMPWM_1_FB  =  FFH             (failure feedback possible)
    If  T_VIMPWM_1_FB > C_T_VIMPWM_1_FB_MAX
                                            (T_VIMPWM_1_FB out of range for FB)
    Then    ERR_VIMPWM_1_FB  =  00H
    Endif
Else
    If  C_T_VIMPWM_1_FB_MIN ≤ T_VIMPWM_1_FB ≤ C_T_VIMPWM_1_FB_MAX
    Then    failure feedback detected
            ERR_VIMPWM_1_FB  =  01H
            LV_VIMPWM_1_FB_VLD  =  1
    Else    no failure feedback possible
            ERR_VIMPWM_1_FB  =  00H
    Endif
    T_VIMPWM_1_FB  =  0
Endif
```

The same calculation is done for all ...VIMPWM_2...-outputs.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_N_MAX_ERR_VIM_FB	1	0...FFH	0...8160	32	[rpm]
Engine speed threshold, no VIM feedback error above					
C_T_VIMPWM_1_FB_MIN	1	0...FFH	0...25.5	0.1	[s]
Time range necessary for identification of failure feedback minimal value					
C_T_VIMPWM_2_FB_MIN	1	0...FFH	0...25.5	0.1	[s]
Time range necessary for identification of failure feedback minimal value					
C_T_VIMPWM_1_FB_MAX	1	0...FFH	0...25.5	0.1	[s]
Time range necessary for identification of failure feedback maximal value					
C_T_VIMPWM_2_FB_MAX	1	0...FFH	0...25.5	0.1	[s]
Time range necessary for identification of failure feedback maximal value					

70.2.4 Electrical diagnosis of VIM (LV_ERR_VIM_1 / 2_EL)

Import actions:

ACTION_ERRM_FilterMulticondition(IN< *XX* >, IN< CDN_DIAG_XX >, IN< ERR_DIAG_XX >, IN<C_ABC_INC_XX>, IN<C_ABC_MAX_XX>,OUT<LV_ERR_XX>)
This action compute the elementary antibounce filter for one failure treatment and return filter result

FUNCTION DESCRIPTION:

General information:

The purpose is to diagnose the electrical signal from the two drivers which controls the VIM.

Description:

The driver can distinguish between three symptoms: ‘Short circuit to Vbatt’, ‘Short circuit to GND’ and ‘Open Load’.
The error detection algorithm in this function for “short circuit to GND” is respecting the status of the failure feedback recognition (ERR_VIMPWM_1_FB and ERR_VIMPWM_2_FB). The failure, detected by the HW, is delivered from the infrastructure ERR_DIAG_VIM_1 and ERR_DIAG_VIM_2.

Error-symptoms: are defined to this diagnosis function as following :

```
☐☐☐☐
 | | | └-> Short circuit to Vbatt          (= SYM_0)
 | | └---> Short circuit to GND            (= SYM_1)
 | └-----> Open load                        (=SYM_2)
 └-------> -                                (=SYM_3)
```

Application conditions:

Initialisation: according Filter-type **MPL_STD_INI**

(reset of variables at LV_IGK 0 -> 1 or ECU reset)

LV_ERR_VIM_1_EL = 0

LV_ERR_VIM_2_EL = 0

CDN_DIAG_VIM_1_EL = 0

CDN_DIAG_VIM_2_EL = 0

ERR_SYM_VIM_1_EL = 0

ERR_SYM_VIM_2_EL = 0

LV_INH_DIAG_VIM_1_EL = 0

LV_INH_DIAG_VIM_2_EL = 0

Recurrence: 100ms

Activation: at every engine operating state

Formula section:

Diagnosis inhibition:

```
If      LV_IGK = 1                  and
        LV_CDN_VB_OBD1 = 1          and
        LV_ERR_SPI_MPS = 0
Then    LV_INH_DIAG_VIM_1_EL = 0
Else    LV_INH_DIAG_VIM_1_EL = 1
Endif


If      LV_INH_DIAG_VIM_1_EL = 0
Then
    Note: The failure symptoms ERR_SYM_VIM_1_EL and the diagnosis conditions
    CDN_DIAG_VIM_1_EL (raw values) shall be set according infrastructure information
    ERR_DIAG_VIM_1 and CDN_DIAG_VIM_1_BAS.

    If      Activation conditions are met for the NC_PSD_DLY_VIM_EL recurrence
    And     ERR_VIMPWM_1_FB = 00h
    Then

        { Additional diagnosis conditions }

        If  VIMPWM_1 < C_VIM_PWM_DIAG_MIN_SCP
                { condition that disables the SYM_0 detection }
        Then
              bit 0 of CDN_DIAG_VIM_1_EL  = 0     { Diagnosis of SCP is not possible }
        Endif
        If  VIMPWM_1 > C_VIM_PWM_DIAG_MAX_SCG
                { condition that disables the SYM_1 detection }
        Then
              bit 1 of CDN_DIAG_VIM_1_EL = 0      { Diagnosis of SCG is not possible }
        Endif
        If  VIMPWM_1 < C_VIM_PWM_DIAG_MIN_OC
            or VIMPWM_1 > C_VIM_PWM_DIAG_MAX_SCG
                { conditions that disable the SYM_2 detection }
        Then
              bit 2 of CDN_DIAG_VIM_1_EL  = 0     { Diagnosis of OC is not possible }
        Endif
    Else
        CDN_DIAG_VIM_1_EL = 0
    Endif
Else
        CDN_DIAG_VIM_1_EL = 0
```

Endif

The same calculation is done for all ...VIMPWM_2...-outputs.

Failure debouncing and error management treatment:

For failure and error management treatment the anti-bounce mechanism is called with the parameters CDN_DIAG_VIM_1/2_EL and ERR_DIAG_VIM_1/2.

```
ACTION_ERRM_FilterMulticondition    (IN<VIMPWM_1/2_EL>,
                                     IN<CDN_DIAG_VIM_1/2_EL>,
                                     IN<ERR_DIAG_VIM_1/2>,
                                     IN<C_ABC_INC_VIM_1/2_EL>,
                                     IN<C_ABC_MAX_VIM_1/2_EL>,
                                     OUT< LV_ERR_VIM_1/2_EL>,
                                     SYNCRONIZATION<CALL>)
```

This algorithm determines:

ERR_SYM_VIM_1/2_EL	(detected error symptom)
LV_ERR_VIM_1/2_EL	(Error flag for debounced error)
LV_CDN_DIAG_VIM_1/2_EL	(Diagnosis condition information)
LV_END_DIAG_VIM_1/2_EL	(End of diagnosis information)

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_VIM_1_EL	1	0...FFH	0...255	1	[-]
Debounce counter increment VIM_1 diagnosis					
C_ABC_INC_VIM_2_EL	1	0...FFH	0...255	1	[-]
Debounce counter increment VIM_2 diagnosis					
C_ABC_MAX_VIM_1_EL	1	1...FFH	1...255	1	[-]
Debounce counter maximum value VIM_1 diagnosis					
C_ABC_MAX_VIM_2_EL	1	1...FFH	1...255	1	[-]
Debounce counter maximum value VIM_2 diagnosis					
C_VIM_PWM_DIAG_MIN_SCP	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Minimum threshold for SCB diagnosis window					
C_VIM_PWM_DIAG_MAX_SCG	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Maximum threshold for SCG diagnosis window					
C_VIM_PWM_DIAG_MIN_OC	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Minimum threshold for OC diagnosis window					

Configuration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
NC_PSD_DLY_VIM_EL	1	1...FH	1...15	1	[-]
Number of recurrences with diagnosis deactivation after diagnosis activation conditions are met (typical 2)					

70.2.5 Feedback diagnosis of VIM

FUNCTION DESCRIPTION:

General information:

If a failure is detected by the DISA control unit, a failure feedback is send to the ECU.

Description:

According the possible failure feedback from the DISA control unit (defined by ERR_VIMPWM_1/2_FB and LV_VIMPWM_1/2_FB_VLD), the diagnosis is managing the following failure: LV_ERR_VIMPWM_1/2_FB.

Error-symptoms of failure LV_ERR_VIMPWM_1/2_FB:

```
□□□□
│ │ │ └─> not used                              (= SYM_0)
│ │ └───> not used                              (= SYM_1)
│ └─────> not used                              (= SYM_2)
└───────> problem on DISA control unit detected (= SYM_3)
```

Application conditions:

Initialization: at reset **or** FMY reset **or** LV_IGK = 0 -> 1
at NVRAM error CTR_VIMPWM_1_EDGE and CTR_VIMPWM_2_EDGE are set to 0

Remark:

CTR_VIMPWM_1_EDGE and CTR_VIMPWM_2_EDGE are lifetime counter, the are not initialized at SW version change!

according filter-type NO_FIL
T_ACT_VIMPWM_1_FB = 0
T_ACT_VIMPWM_2_FB = 0
T_DIAG_VIMPWM_1_FB = 0
T_DIAG_VIMPWM_2_FB = 0
ABC_VIMPWM_1_FB = 0
ABC_VIMPWM_2_FB = 0
LV_EDGE_VIM_1 = 0
LV_EDGE_VIM_2 = 0
LV_EDGE_DIAG_VIM_1 = 0
LV_EDGE_DIAG_VIM_2 = 0
LV_ERR_VIMPWM_1_FB = 0
LV_ERR_VIMPWM_2_FB = 0
LV_CDN_DIAG_VIMPWM_1_FB = 0
LV_CDN_DIAG_VIMPWM_2_FB = 0
ERR_SYM_VIMPWM_1_FB = 0
ERR_SYM_VIMPWM_1_FB = 0

Recurrence: 100ms

Activation: LV_CDN_VB_MIN_DIAG = 1 **and** LV_ERR_SPI_MPS = 0

Deactivation: LV_CDN_VB_MIN_DIAG = 0 **or** LV_ERR_SPI_MPS = 1
at deactivation: LV_CDN_DIAG_VIMPWM_1/2_FB = 0

Formula section:

```
LV_CDN_DIAG_VIMPWM_1_FB  =  1

If    VIMPWM_1 > C_VIMPWM_1_MIN_FB
Then        T_ACT_VIMPWM_1_FB  =  T_ACT_VIMPWM_1_FB +100ms
            T_ACT_VIMPWM_1_FB is limited to 6553.5s
Endif

If(1) LV_VIM_1  0→1 or 1→0                          // Disa switchs
Then(1)     LV_EDGE_VIM_1 = 1                       // Switch detection
            T_DIAG_VIMPWM_1_FB = 0                  // Timer feedback re-set
            If(2) C_ABC_INC_VIMPWM_1_FB > 0         // diagnosis activ / Disa available
            Then(2) CTR_VIMPWM_1_EDGE + 1           // counter increments irreversible
            Endif(2)                                //(reso/range FFFFFFh, 16777215 dec.)
Else(1)     If(3) LV_EDGE_VIM_1 = 1 and ERR_VIMPWM_1_FB = 0
                                                    // switch without failure feedback
            Then(3) T_DIAG_VIMPWM_1_FB = T_DIAG_VIMPWM_1_FB +100ms
                                                    // start feedback timer
            Else(3) T_DIAG_VIMPWM_1_FB = 0          // reset feedbacktimer
```

```
          Endif(3)
Endif(1)

If(4) T_DIAG_VIMPWM_1_FB >= C_T_DIAG_VIMPWM_1_FB// Timer limit (ca. 6 sek)
Then(4)   LV_EDGE_DIAG_VIM_1 = 1                   // Switch ready
          LV_EDGE_VIM_1 = 0                        // re-set switch-detection
Else(4)   LV_EDGE_DIAG_VIM_1 = 0
Endif(4)

If    LV_ERR_VIMPWM_1_FB = 0
Then      If LV_VIMPWM_1_FB_VLD = 1 (valid failure feedback information available)
          Then  ERR_SYM_VIMPWM_1_FB  =  SYM_3
                LV_EDGE_VIM_1 = 0
                ABC_VIMPWM_1_FB = ABC_VIMPWM_1_FB
                                           + C_ABC_INC_VIMPWM_1_FB
                If     ABC_VIMPWM_1_FB ≥ C_ABC_MAX_VIMPWM_1_FB
                Then   LV_ERR_VIMPWM_1_FB = 1
                       LV_END_DIAG_VIMPWM_1_FB = 1
                Endif
          Else  If     ERR_VIMPWM_1_FB = 0     (no failure feedback detected)
                Then   ERR_SYM_VIMPWM_1_FB  =  0
                       If LV_EDGE_DIAG_VIM_1 = 1
                       Then ABC_VIMPWM_1_FB = ABC_VIMPWM_1_FB
                                                - C_ABC_DEC_VIMPWM_1_FB
                       Endif
                       If     T_ACT_VIMPWM_1_FB ≥ C_T_ACT_VIMPWM_1_FB
                       Then   LV_END_DIAG_VIMPWM_1_FB  =  1
                       Endif
                Endif
          Endif
Endif
```

The same calculations are done for all ...VIMPWM_2...-outputs.

Calibration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
C_ABC_INC_VIMPWM_1_FB	1	0...FFH	0...255	1	[-]
ABC Increment for VIMPWM_1_FB Diagnosis					
C_ABC_INC_VIMPWM_2_FB	1	0...FFH	0...255	1	[-]
ABC Increment for VIMPWM_2_FB Diagnosis					
C_ABC_MAX_VIMPWM_1_FB	1	1...FFH	1...255	1	[-]
ABC Maximum for VIMPWM_1_FB Diagnosis					
C_ABC_MAX_VIMPWM_2_FB	1	1...FFH	1...255	1	[-]
ABC Maximum for VIMPWM_2_FB Diagnosis					
C_VIMPWM_1_MIN_FB	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Minimum duty cycle of VIMPWM_1 to have a failure feedback from DISA control unit					
C_VIMPWM_2_MIN_FB	1	8000...7FFFH	-100...99.99694	3.0518e-3	[%]
Minimum duty cycle of VIMPWM_2 to have a failure feedback from DISA control unit					
C_T_ACT_VIMPWM_1_FB	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time necessary to have failure feedback from DISA control unit					
C_T_ACT_VIMPWM_2_FB	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time necessary to have failure feedback from DISA control unit					
C_ABC_DEC_VIMPWM_1_FB	1	0...FFH	0...255	1	[-]
ABC Decrement for VIMPWM_1_FB Diagnosis					
C_ABC_DEC_VIMPWM_2_FB	1	0...FFH	0...255	1	[-]
ABC Decrement for VIMPWM_2_FB Diagnosis					
C_T_DIAG_VIMPWM_1_FB	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time necessary after switch to allow decrement (Flap 1)					
C_T_DIAG_VIMPWM_2_FB	1	0...FFFFH	0...6553.5	0.1	[s]
Minimum time necessary after switch to allow decrement (Flap 2)					

Diagnostic symptoms:

Diagnostic **VIMPWM_1/2_ERR_FB**	**Symptom description**	**Symptom**	**Filter type**
failure feedback diagnosis of DISA control unit	see description	*SYM_0*	*NO*
		SYM_1	
		SYM_2	
		SYM_3	

70.3 Customer adaptation modul: AGGR VIMA

70.3.1 Outputs for BMW functions which are not defined as VIMA exported data

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
Disa_ist	V/O	0...FFFFH	0...99.99847	1.5259e-3	[%]
actual position variable intake manifold					

Input data:

VIM_AV			

FUNCTION DESCRIPTION:

Adaptation to BMW environment.

Remark: the really possible resolution and limits of the outputs are different from the specified values due to the input data attributes.

Application conditions:

Initialisation at reset or at exit power latch phase: 0 %

Recurrence : 10 ms

Activation: every engine state,

Deactivation: ---

Formula section:

Remark: all formulas are valid in a **physical** meaning

Disa_ist = VIM_AV

71 Variable Valve Timing

Table of Contents

Index

71.1 Customer adaptation module: AGGR VVTI

71.1.1 Interfaces served by customer OBJ Files

Output data:

Name	Mode	Hex. limits	Phys. limits	Resol.	Unit
CAM_EX[NC_NR_CBK_IVVT]	V/O	0...FF	-40...-135.625	-0.375	[°CRK]
Actual position exhaust CAM VVTI					
CAM_IN[NC_NR_CBK_IVVT]	V/O	0...FF	60...155.625	0.375	[°CRK]
Actual position inlet CAM VVTI					
CAM_SP_IVVT_EX	O	0...FFH	-40...-135.625	-0.375	[°CRK]
Exhaust CAM setpoint for VVTI					
CAM_SP_IVVT_IN	O	0...FFH	60...155.625	0.375	[°CRK]
Inlet CAM setpoint for VVTI					
CAM_SP_REF_EX	O	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Exhaus camshaft target spread at reference conditions (warm engine)					
CAM_SP_REF_IN	O	8000...7FFFH	-768...767.97656	0.0234375	[°CRK]
Inlet camshaft target spread at reference conditions (warm engine)					
Nwa1_pos	O	0...FFFFH	0...6553.5	0.1	[°CRK]
New exhaust camshaft edge (el. falling)					
Nwe1_pos	O	0...FFFFH	0...6553.5	0.1	[°CRK]
New inlet camshaft edge (el. falling)					
Nwa1_time	V/O	0...7FFFFFFFH	0...2147483647	1	[ms]
Time stamp at new exhaust camshaft edge (el. falling)					
Nwe1_time	V/O	0...7FFFFFFFH	0...2147483647	1	[ms]
Time stamp at new inlet camshaft edge (el. falling)					
Vse_sprs_kh	O	0...FFFFH	0...6553.5	0.1	[°CRK]
Cam setpoint for inlet cam via catalyst heating					
Vsa_sprs_kh	O	0...FFFFH	0...6553.5	0.1	[°CRK]
Cam setpoint for exhaust cam via catalyst heating					
LV_DI_AD_REF_CAM_IVVT_IN	V/O	0...1H	0...1	1	[-]
Adaptation of inlet camshaft reference position: 0 = enabled, 1 = disabled					
LV_DI_AD_REF_CAM_IVVT_EX	V/O	0...1H	0...1	1	[-]
Adaptation of exhaust camshaft reference position: 0 = enabled, 1 = disabled					
LV_CAM_LOCK_IVVT_EX[NC_NR_CBK_IVVT]	V/O	0...1H	0...1	1	[-]
VVT on exhaust camshaft in locked position at engine start					
LV_CAM_LOCK_IVVT_IN[NC_NR_CBK_IVVT]	V/O	0...1H	0...1	1	[-]
VVT on intake camshaft in locked position at engine start					

Input data:

Aspr	Espr	NC_NR_CBK_IVVT	NC_NR_CAM_CBK
Vsa_spri	Vse_spri	LV_ES	Vse_sprn
Vsa_sprn	NC_PSN_EDGE_CAM_EX[NC_NR_EDGE_CAM_EX][NC_NR_CAM_CBK]	NC_PSN_EDGE_CAM_IN[NC_NR_EDGE_CAM_IN][NC_NR_CAM_CBK]	PSN_CAM_IN[NC_NR_CAM_CBK]
PSN_CAM_EX[NC_NR_CAM_CBK]	NC_NR_EDGE_CAM_EX	NC_NR_EDGE_CAM_IN	IDX_EDGE_CAM_IN[NC_NR_CAM_CBK]
IDX_EDGE_CAM_EX[NC_NR_CAM_CBK]	LV_SYN_CAM_IN[NC_NR_CAM_CBK]	LV_SYN_CAM_EX[NC_NR_CAM_CBK]	LV_SYN_VLD
CAM_SP_CH_EX	CAM_SP_CH_IN	B_vseadp	B_vsaadp
LV_VLD_PSN_CAM_IN[NC_NR_CAM_CBK]	LV_VLD_PSN_CAM_EX[NC_NR_CAM_CBK]	B_vsa_an	B_vse_an

FUNCTION DESCRIPTION:

Adaption to BMW environment.

Remark: The really possible resolution and limits of the output are different to the specified values due to the input data attributes.

NC_NR_CBK_IVVT indicates the number of camshaft cylinder banks and it is set to 1 in this project.

NC_NR_CAM_CBK indicates whether camshaft sensor(s) is present in one or both cylinder banks.

Application conditions:

Initialisation: at reset

CAM_EX [1] = C_CAM_INI_EX
CAM_IN [1] = C_CAM_INI_IN
CAM_SP_IVVT_EX = C_CAM_INI_EX
CAM_SP_IVVT_IN = C_CAM_INI_IN
CAM_SP_REF_EX = C_CAM_INI_EX
CAM_SP_REF_IN = C_CAM_INI_IN
LV_DI_AD_REF_CAM_IVVT_IN = 1
LV_DI_AD_REF_CAM_IVVT_EX = 1
Vsa_sprs_kh = - C_CAM_INI_EX; Vse_sprs_kh = C_CAM_INI_IN
LV_CAM_LOCK_IVVT_EX[i] = B_vsa_an (out of NVMY)
LV_CAM_LOCK_IVVT_INEX[i] = B_vse_an (out of NVMY)
NLC_IVVT_IN = 1
NLC_IVVT_EX = 1

Recurrence: segment synchronous

Except :

10ms: LV_DI_AD_REF_CAM_IVVT_IN,
LV_DI_AD_REF_CAM_IVVT_EX,
LV_CAM_LOCK_IVVT_EX[NC_NR_CBK_IVVT],
LV_CAM_LOCK_IVVT_IN[NC_NR_CBK_IVVT],

Activation:

LV_ES = 1 → 0

Deactivation:

LV_ES = 0 → 1

Formula section:

Remark: all formulas are valid in a **physical** meaning

```
CAM_EX[1]  = - Vsa_spri
CAM_IN[1]   = Vse_spri
CAM_SP_IVVT_EX = - Aspr
CAM_SP_IVVT_IN  = Espr
CAM_SP_REF_EX = - Vsa_sprn
CAM_SP_REF_IN = Vse_sprn
```

With i = 1...NC_NR_CAM_CBK:

```
If          LV_SYN_CAM_IN[i] = 1
    And     LV_SYN_VLD = 1
    And     IDX_EDGE_CAM_IN[i] = 2 / 3 / 5 / 6 /*relevant edge on intake camshaft ?*/
Then

            Nwe1_pos = NC_PSN_EDGE_2 / 3 / 5 / 6_CAM_IN[i] + PSN_CAM_IN[i]
            Nwe1_time is updated by calling the Makro “getSystemTime1ms()”
Else
            do nothing
Endif

If          LV_SYN_CAM_EX[i] = 1
    And     LV_SYN_VLD = 1
    And     IDX_EDGE_CAM_EX[i] = 2 / 3 / 5 / 6 /* relevant edge on exhaust camshaft ?*/
Then

            Nwa1_pos = NC_PSN_EDGE_2 / 3 / 5 / 6_CAM_EX[i] + PSN_CAM_EX[i]
            Nwa1_time is updated by calling the Makro “getSystemTime1ms()”
Else
            do nothing
Endif
```

Remark: Starting with the SW releases 4DB7100S only engines with a 3-teeth-target-wheel are supported ! Older engines with a 4-teeth-target-wheel have to use older SW !

```
Vsa_sprs_kh = - CAM_SP_CH_EX
Vse_sprs_kh = CAM_SP_CH_IN
```

```
If      LV_VLD_PSN_CAM_IN[i] = 1        and     /*sensor self adaptation ended*/
        B_vseadp = 1                            /*adaptation enabled by BMW*/
Then    LV_DI_AD_REF_CAM_IVVT_IN = 0            /*inlet ref.-pos.-adap. enabled*/
Else    LV_DI_AD_REF_CAM_IVVT_IN = 1            /*inlet ref.-pos.-adap. disabled*/
Endif


If      LV_VLD_PSN_CAM_EX[i] = 1        and     /*sensor self adaptation ended*/
        B_vsaadp = 1                            /*adaptation enabled by BMW*/
Then    LV_DI_AD_REF_CAM_IVVT_EX = 0            /*inlet ref.-pos.-adap. enabled*/
```

```
Else    LV_DI_AD_REF_CAM_IVVT_EX = 1        /*inlet ref.-pos.-adap. disabled*/
Endif
LV_CAM_LOCK_IVVT_EX[i] = B_vsa_an
LV_CAM_LOCK_IVVT_IN[i] = B_vse_an
```

Calibration data:

Name	Dim	Hex. Limits	Phys. limits	Resol.	Unit
C_CAM_INI_IN	1	0...FFH	60...155.625	0.375	[°CRK]
Initialization value for inlet camshaft					
C_CAM_INI_EX	1	0...FFH	-40...-135.625	-0.375	[°CRK]
Initialization value for exhaust camshaft					
LC_NOT_ADJ_CAM_IVVT_EX[NC_NR_CBK_IVVT]	1	0...1H	0...1	1	[-]
Camshaft_EX_i: 0 = with adjustment, 1 = without adjustment					
LC_NOT_ADJ_CAM_IVVT_IN[NC_NR_CBK_IVVT]	1	0...1H	0...1	1	[-]
Camshaft_IN_i: 0 = with adjustment, 1 = without adjustment					

Configuration data:

Name	Dim	Hex. limits	Phys. limits	Resol.	Unit
NLC_IVVT_IN	-	0...1H	0...1	1	[-]
Inlet side configuration: 0 = without actuator, 1 = with actuator					
NLC_IVVT_EX	-	0...1H	0...1	1	[-]
Exhaust side configuration: 0 = without actuator, 1 = with actuator					